Open Source Used In AMP for Endpoints Connector (Windows) 7.5.1

Cisco Systems, Inc.
www.cisco.com

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco website at www.cisco.com/go/offices.

Text Part Number: 78EE117C99-1228196852

This document contains licenses and notices for open source software used in this product. With respect to the free/open source software listed in this document, if you have any questions or wish to receive a copy of any source code to which you may be entitled under the applicable free/open source license(s) (such as the GNU Lesser/General Public License), please submit this form.

In your requests please include the following reference number 78EE117C99-1228196852

Contents

1.1 openssl 1.0.2k

1.1.1 Notifications :

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/)
This product includes cryptographic software written by Eric Young (eay@cryptsoft.com).

This product includes software written by Tim Hudson (tjh@cryptsoft.com).

1.1.2 Available under license :

Copyright (C) 1995-1997 Eric Young (eay@cryptsoft.com)
All rights reserved.

This package is an DES implementation written by Eric Young (eay@cryptsoft.com).
The implementation was written so as to conform with MIT's libdes.

This library is free for commercial and non-commercial use as long as
the following conditions are aheared to. The following conditions
apply to all code found in this distribution.

Copyright remains Eric Young's, and as such any Copyright notices in
the code are not to be removed.
If this package is used in a product, Eric Young should be given attribution
as the author of that the SSL library. This can be in the form of a textual
message at program startup or in documentation (online or textual) provided
with the package.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the copyright
notice, this list of conditions and
the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software
must display the following acknowledgement:
This product includes software developed by Eric Young (eay@cryptsoft.com)

THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

The license and distribution terms for any publically available version or
derivative of this code cannot be changed. i.e. this code cannot simply be
copied and put under another distrubution license

[including the GNU Public License.]

The reason behind this being stated in this direct manner is past
experience in code simply being copied and the attribution removed
from it and then being distributed as part of other packages. This
implementation was a non-trivial and unpaid effort.
Copyright (C) 1995-1997 Eric Young (eay@cryptsoft.com)
All rights reserved.

This package is an Blowfish implementation written
by Eric Young (eay@cryptsoft.com).

This library is free for commercial and non-commercial use as long as
the following conditions are aheared to. The following conditions
apply to all code found in this distribution.

Copyright remains Eric Young's, and as such any Copyright notices in
the code are not to be removed.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the copyright
notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software
must display the following acknowledgement:
This product includes software developed by Eric Young (eay@cryptsoft.com)

THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

The license and distribution terms for any publically available version or
derivative of this code cannot be changed. i.e. this code cannot simply be
copied
and put under another distrubution license
[including the GNU Public License.]

The reason behind this being stated in this direct manner is past
experience in code simply being copied and the attribution removed
from it and then being distributed as part of other packages. This
implementation was a non-trivial and unpaid effort.

LICENSE ISSUES
==========

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of
the OpenSSL License and the original SSLeay license apply to the toolkit.
See below for the actual license texts. Actually both licenses are BSD-style
Open Source licenses. In case of any license issues related to OpenSSL
please contact openssl-core@openssl.org.

OpenSSL License

/* ==
* Copyright (c) 1998-2016 The OpenSSL Project. All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
*
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer
in
*   the documentation and/or other materials provided with the
*   distribution.
*
* 3. All advertising materials mentioning features or use of this
*   software must display the following acknowledgment:
*   "This product includes software developed by the OpenSSL Project
*   for use in the OpenSSL Toolkit. (http://www.openssl.org/)"
*
* 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
*   endorse or promote products derived from this software without
*   prior written permission. For written permission, please contact
*   openssl-core@openssl.org.
*
* 5. Products derived from this software may not be called "OpenSSL"
*   nor may "OpenSSL" appear in their names without prior written
*   permission of the OpenSSL Project.
*

* 6. Redistributions of any form whatsoever must retain the following
*   acknowledgment:
*   "This product includes software developed by the OpenSSL Project
*   for use in the OpenSSL Toolkit (http://www.openssl.org/)"
 *
* THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY
* EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
* PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR
* ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
* STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
* OF THE POSSIBILITY OF SUCH DAMAGE.
* ==
*
* This product includes cryptographic software written by Eric Young
* (eay@cryptsoft.com). This product includes software
 written by Tim
* Hudson (tjh@cryptsoft.com).
*
*/

Original SSLeay License

/* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
* All rights reserved.
*
* This package is an SSL implementation written
* by Eric Young (eay@cryptsoft.com).
* The implementation was written so as to conform with Netscapes SSL.
*
* This library is free for commercial and non-commercial use as long as
* the following conditions are aheared to. The following conditions
* apply to all code found in this distribution, be it the RC4, RSA,
* lhash, DES, etc., code; not just the SSL code. The SSL documentation
* included with this distribution is covered by the same copyright terms
* except that the holder is Tim Hudson (tjh@cryptsoft.com).
*
* Copyright remains Eric Young's, and as such any Copyright notices in
* the code are not to be removed.
* If this package is used in a product, Eric Young should be given attribution
* as the author of the parts of the

```
 library used.
* This can be in the form of a textual message at program startup or
* in documentation (online or textual) provided with the package.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the copyright
*    notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the distribution.
* 3. All advertising materials mentioning features or use of this software
*    must display the following acknowledgement:
*    "This product includes cryptographic software written by
*     Eric Young (eay@cryptsoft.com)"
*    The word 'cryptographic' can be left out if the rouines from the library
*    being used are not cryptographic
 related :-).
* 4. If you include any Windows specific code (or a derivative thereof) from
*    the apps directory (application code) you must include an acknowledgement:
*    "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
*
* THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED.  IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
 OF
* SUCH DAMAGE.
*
* The licence and distribution terms for any publically available version or
* derivative of this code cannot be changed.  i.e. this code cannot simply be
* copied and put under another distribution licence
* [including the GNU Public Licence.]
*/
```

1.2 protobuf 3.5.1

1.2.1 Available under license :

Copyright 2008, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
This license applies to all parts of Protocol Buffers except the following:

- Atomicops support for generic gcc, located in
  src/google/protobuf/stubs/atomicops_internals_generic_gcc.h.
  This file is copyrighted by Red Hat Inc.

- Atomicops support for AIX/POWER, located in
  src/google/protobuf/stubs/atomicops_internals_power.h.
  This file is copyrighted by Bloomberg Finance LP.

Copyright 2014, Google Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
* Neither the name of Google Inc. nor the names of its
contributors
may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Code generated by the Protocol Buffer compiler is owned by the owner
of the input file used when generating it. This code is
not
standalone and requires a support library to be linked with it. This
support library is itself covered by the above license.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership

of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial
revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,

worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or

documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct,

indirect, special,
incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [2007] Neal Norwitz
Portions Copyright [2007] Google Inc.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the
License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

This file contains a list of people who've made non-trivial
contribution to the Google C++ Mocking Framework project. People
who commit code to the project are encouraged to add their names
here. Please keep the list sorted by first names.

Benoit Sigoure <tsuna@google.com>
Bogdan Piloca <boo@google.com>
Chandler Carruth <chandlerc@google.com>
Dave MacLachlan <dmaclach@gmail.com>
David Anderson <danderson@google.com>
Dean Sturtevant
Gene Volovich <gv@cite.com>
Hal Burch <gmock@hburch.com>
Jeffrey Yasskin <jyasskin@google.com>
Jim Keller <jimkeller@google.com>
Joe Walnes <joe@truemesh.com>
Jon Wray <jwray@google.com>
Keir Mierle <mierle@gmail.com>
Keith Ray <keith.ray@gmail.com>
Kostya Serebryany <kcc@google.com>
Lev Makhlis
Manuel Klimek <klimek@google.com>
Mario Tanev <radix@google.com>
Mark Paskin
Markus Heule <markus.heule@gmail.com>
Matthew Simmons <simmonmt@acm.org>
Mike Bland <mbland@google.com>
Neal Norwitz <nnorwitz@gmail.com>
Nermin Ozkiranartli <nermin@google.com>
Owen
 Carlsen <ocarlsen@google.com>
Paneendra Ba <paneendra@google.com>
Paul Menage <menage@google.com>
Piotr Kaminski <piotrk@google.com>
Russ Rufer <russ@pentad.com>
Sverre Sundsdal <sundsdal@gmail.com>
Takeshi Yoshino <tyoshino@google.com>
Vadim Berman <vadimb@google.com>
Vlad Losev <vladl@google.com>
Wolfgang Klier <wklier@google.com>
Zhanyong Wan <wan@google.com>
This file contains a list of people who have made large contributions
to the public version of Protocol Buffers.

Original Protocol Buffers design and implementation:
 Sanjay Ghemawat <sanjay@google.com>
 Jeff Dean <jeff@google.com>
 Daniel Dulitz <daniel@google.com>

Craig Silverstein
Paul Haahr <haahr@google.com>
Corey Anderson <corin@google.com>
(and many others)

Proto2 C++ and Java primary author:
Kenton Varda <kenton@google.com>

Proto2 Python primary authors:
Will Robinson <robinson@google.com>
Petar Petrov <petar@google.com>

Java Nano primary authors:
Brian Duff <bduff@google.com>
Tom Chao <chaot@google.com>
Max Cai <maxtroy@google.com>
Ulas Kirazci <ulas@google.com>

Large code contributions:
Jason Hsueh <jasonh@google.com>
Joseph Schorr <jschorr@google.com>
Wenbo Zhu <wenboz@google.com>

Large quantity of code reviews:
Scott Bruce <sbruce@google.com>
Frank Yellin
Neal Norwitz <nnorwitz@google.com>
Jeffrey Yasskin <jyasskin@google.com>
Ambrose
Feinstein <ambrose@google.com>

Documentation:
Lisa Carey <lcarey@google.com>

Maven packaging:
Gregory Kick <gak@google.com>

Patch contributors:
Kevin Ko <kevin.s.ko@gmail.com>
  * Small patch to handle trailing slashes in --proto_path flag.
Johan Euphrosine <proppy@aminche.com>
  * Small patch to fix Python CallMethod().
Ulrich Kunitz <kune@deine-taler.de>
  * Small optimizations to Python serialization.
Leandro Lucarella <llucax@gmail.com>
  * VI syntax highlighting tweaks.
  * Fix compiler to not make output executable.
Dilip Joseph <dilip.antony.joseph@gmail.com>

  * Heuristic detection of sub-messages when printing unknown fields in
    text format.
Brian Atkinson <nairb774@gmail.com>
  * Added @Override annotation to generated Java code where appropriate.
Vincent Choinire <Choiniere.Vincent@hydro.qc.ca>
  * Tru64 support.
Monty Taylor <monty.taylor@gmail.com>
  * Solaris 10 + Sun Studio fixes.
Alek Storm <alek.storm@gmail.com>

  * Slicing support for repeated scalar fields for the Python API.
Oleg Smolsky <oleg.smolsky@gmail.com>
  * MS Visual Studio error format option.
  * Detect unordered_map in stl_hash.m4.
Brian Olson <brianolson@google.com>
  * gzip/zlib I/O support.
Michael Poole <mdpoole@troilus.org>
  * Fixed warnings about generated constructors not explicitly initializing
    all fields (only present with certain compiler settings).
  * Added generation of field number constants.
Wink Saville <wink@google.com>
  * Fixed initialization ordering problem in logging code.
Will Pierce <willp@nuclei.com>
  * Small patch improving performance of in Python serialization.
Alexandre Vassalotti <alexandre@peadrop.com>
  * Emacs mode for Protocol Buffers (editors/protobuf-mode.el).
Scott Stafford <scott.stafford@gmail.com>
  * Added Swap(), SwapElements(), and RemoveLast() to Reflection interface.
Alexander Melnikov <alm@sibmail.ru>
  * HPUX support.
Oliver Jowett <oliver.jowett@gmail.com>
   * Detect whether zlib is new enough in configure script.
  * Fixes for Solaris 10 32/64-bit confusion.
Evan Jones <evanj@mit.edu>
  * Optimize Java serialization code when writing a small message to a stream.
  * Optimize Java serialization of strings so that UTF-8 encoding happens only
    once per string per serialization call.
  * Clean up some Java warnings.
  * Fix bug with permanent callbacks that delete themselves when run.
Michael Kucharski <m.kucharski@gmail.com>
  * Added CodedInputStream.getTotalBytesRead().
Kacper Kowalik <xarthisius.kk@gmail.com>
  * Fixed m4/acx_pthread.m4 problem for some Linux distributions.
William Orr <will@worrbase.com>
  * Fixed detection of sched_yield on Solaris.
  * Added atomicops for Solaris
Andrew Paprocki <andrew@ishiboo.com>
  * Fixed minor IBM xlC compiler build issues

* Added atomicops for AIX (POWER)
# This file contains a list of people who've made non-trivial
# contribution to the Google C++ Testing Framework project. People
# who commit code to the project are encouraged to add their names
# here. Please keep the list sorted by first names.

Ajay Joshi <jaj@google.com>
Balzs Dn <balazs.dan@gmail.com>
Bharat Mediratta <bharat@menalto.com>
Chandler Carruth <chandlerc@google.com>
Chris Prince <cprince@google.com>
Chris Taylor <taylorc@google.com>
Dan Egnor <egnor@google.com>
Eric Roman <eroman@chromium.org>
Hady Zalek <hady.zalek@gmail.com>
Jeffrey Yasskin <jyasskin@google.com>
Ji Sigursson <joi@google.com>
Keir Mierle <mierle@gmail.com>
Keith Ray <keith.ray@gmail.com>
Kenton Varda <kenton@google.com>
Manuel Klimek <klimek@google.com>
Markus Heule <markus.heule@gmail.com>
Mika Raento <mikie@iki.fi>
Mikls Fazekas <mfazekas@szemafor.com>
Pasi Valminen <pasi.valminen@gmail.com>
Patrick Hanna <phanna@google.com>
Patrick Riley <pfr@google.com>
Peter Kaminski <piotrk@google.com>
Preston
 Jackson <preston.a.jackson@gmail.com>
Rainer Klaffenboeck <rainer.klaffenboeck@dynatrace.com>
Russ Cox <rsc@google.com>
Russ Rufer <russ@pentad.com>
Sean Mcafee <eefacm@gmail.com>
Sigurur sgeirsson <siggi@google.com>
Tracy Bialik <tracy@pentad.com>
Vadim Berman <vadimb@google.com>
Vlad Losev <vladl@google.com>
Zhanyong Wan <wan@google.com>

1.3 bzip2 1.0.8

1.4 zlib 1.2.8

1.4.1 Available under license :

/* zlib.h -- interface of the 'zlib' general purpose compression library
version 1.2.11, January 15th, 2017

Copyright (C) 1995-2017 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied
warranty. In no event will the authors be held liable for any damages
arising from the use of this software.

Permission is granted to anyone to use this software for any purpose,
including commercial applications, and to alter it and redistribute it
freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not
claim that you wrote the original software. If you use this software
in a product, an acknowledgment in the product documentation would be
appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be
misrepresented as being the original software.
3. This notice may not be removed or altered from any source
distribution.

Jean-loup Gailly Mark Adler
jloup@gzip.org madler@alumni.caltech.edu

*/

Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization
obtaining a copy of the software and accompanying documentation covered by
this license (the "Software") to use, reproduce, display, distribute,
execute, and transmit the Software, and to prepare derivative works of the
Software, and to permit third-parties to whom the Software is furnished to
do so, all subject to the following:

The copyright notices in the Software and this entire statement, including
the above license grant, this restriction and the following disclaimer,
must be included in all copies of the Software, in whole or in part, and
all derivative works of the Software, unless such copies or derivative
works are solely in the form of machine-executable object code generated by
a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES

OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.5 zlib 1.2.7

1.5.1 Available under license :

Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization
obtaining a copy of the software and accompanying documentation covered by
this license (the "Software") to use, reproduce, display, distribute,
execute, and transmit the Software, and to prepare derivative works of the
Software, and to permit third-parties to whom the Software is furnished to
do so, all subject to the following:

The copyright notices in the Software and this entire statement, including
the above license grant, this restriction and the following disclaimer,
must be included in all copies of the Software, in whole or in part, and
all derivative works of the Software, unless such copies or derivative
works are solely in the form of machine-executable object code generated by
a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

This software is provided 'as-is', without any express or implied
warranty. In no event will the authors be held liable for any damages
arising from the use of this software.

Permission is granted to anyone to use this software for any purpose,
including commercial applications, and to alter it and redistribute it
freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not
   claim that you wrote the original software. If you use this software
   in a product, an acknowledgment in the product documentation would be
   appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be

misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

1.6 jansson 2.9

1.6.1 Available under license :

Copyright (c) 2009-2018 Petri Lehtinen <petri@digip.org>

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR
IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.7 libxml 2.9.0

1.7.1 Available under license :

Except where otherwise noted in the source code (e.g. the files hash.c,
list.c and the trio files, which are covered by a similar licence but
with different Copyright notices) all the files are:

1.8 sqlite 3.36.0

1.8.1 Available under license :

1.9 pcre 10.33

1.9.1 Available under license :

Release 8 of PCRE is distributed under the terms of the "BSD" licence, as specified below. The documentation for PCRE, supplied in the "doc" directory, is distributed under the same terms as the software itself. The data in the testdata directory is not copyrighted and is in the public domain.

The basic library functions are written in C and are freestanding. Also included in the distribution is a set of C++ wrapper functions, and a just-in-time compiler that can be used to optimize pattern matching. These are both optional features that can be omitted when the library is built.

THE BASIC LIBRARY FUNCTIONS

Written by: Philip Hazel
Email local part: ph10
Email domain: cam.ac.uk

University of Cambridge Computing Service,
Cambridge, England.

PCRE JUST-IN-TIME COMPILATION SUPPORT

Written by: Zoltan Herczeg
Email local part: hzmester
Email domain: freemail.hu

STACK-LESS JUST-IN-TIME COMPILER

Written by: Zoltan Herczeg
Email local part: hzmester
Email domain: freemail.hu

THE C++ WRAPPER FUNCTIONS

Contributed by: Google Inc.

THE "BSD" LICENCE

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice,
this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce
the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

* Neither the name of the University of Cambridge nor the name of Google
Inc. nor the names of their contributors may be used to endorse or
promote products derived from this software without specific prior
written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT,
STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
POSSIBILITY OF SUCH DAMAGE.

End
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the copyright
notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT
OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.10 llvm 8.0.1

1.10.1 Available under license :

ARM Limited

Software Grant License Agreement ("Agreement")

Except for the license granted herein to you, ARM Limited ("ARM") reserves all right, title, and interest in and to the Software (defined below).

Definition

"Software" means the code and documentation as well as any original work of authorship, including any modifications or additions to an existing work, that is intentionally submitted by ARM to llvm.org (http://llvm.org) ("LLVM") for inclusion in, or documentation of, any of the products owned or managed by LLVM (the "Work"). For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to LLVM or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, LLVM for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked otherwise.

1. Grant of
Copyright License. Subject to the terms and conditions of this
Agreement, ARM hereby grants to you and to recipients of the Software
distributed by LLVM a perpetual, worldwide, non-exclusive, no-charge,
royalty-free, irrevocable copyright license to reproduce, prepare derivative
works of, publicly display, publicly perform, sublicense, and distribute the

Software and such derivative works.

2. Grant of Patent License. Subject to the terms and conditions of this
Agreement, ARM hereby grants you and to recipients of the Software
distributed by LLVM a perpetual, worldwide, non-exclusive, no-charge,
royalty-free, irrevocable (except as stated in this section) patent license
to make, have made, use, offer to sell, sell, import, and otherwise transfer
the Work, where such license applies only to those patent claims licensable
by ARM that are necessarily infringed by ARM's Software alone or by
combination of the Software with the Work to which such Software
was
submitted. If any entity institutes patent litigation against ARM or any
other entity (including a cross-claim or counterclaim in a lawsuit) alleging
that ARM's Software, or the Work to which ARM has contributed constitutes
direct or contributory patent infringement, then any patent licenses granted
to that entity under this Agreement for the Software or Work shall terminate
as of the date such litigation is filed.

Unless required by applicable law or agreed to in writing, the software is
provided on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND,
either express or implied, including, without limitation, any warranties or
conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE.
; RUN: llc < %s -mtriple=s390x-linux-gnu -mcpu=zEC12 -verify-machineinstrs | FileCheck %s
;
; Test that early if conversion produces LOCR with operands of the right
; register classes.

define void @autogen_SD4739(i8*) {
; CHECK-NOT: Expected a GR32Bit register, but got a GRX32Bit register
BB:
 %L34 = load i8, i8* %0
 %Cmp56 = icmp sgt i8 undef, %L34
 br label %CF246

CF246:   ; preds = %CF246, %BB
 %Sl163 = select i1 %Cmp56, i8 %L34, i8 undef
 br i1 undef, label %CF246, label %CF248

CF248:   ; preds = %CF248, %CF246
 store i8 %Sl163, i8* %0
 br label %CF248
}
; NOTE: Assertions have been autogenerated by utils/update_llc_test_checks.py
; RUN: llc < %s -mtriple=x86_64-unknown-unknown -mattr=avx512vl,avx512bw,avx512dq,prefer-256-bit |
FileCheck %s

; This file primarily contains tests for specific places in X86ISelLowering.cpp that needed be made aware of the

legalizer not allowing 512-bit vectors due to prefer-256-bit even though AVX512 is enabled.

```
define void @add256(<16 x i32>* %a, <16 x i32>* %b, <16 x i32>* %c) "min-legal-vector-width"="256" {
; CHECK-LABEL: add256:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vmovdqa (%rdi), %ymm0
; CHECK-NEXT:    vmovdqa 32(%rdi), %ymm1
; CHECK-NEXT:    vpaddd (%rsi), %ymm0, %ymm0
; CHECK-NEXT:    vpaddd 32(%rsi), %ymm1, %ymm1
; CHECK-NEXT:    vmovdqa %ymm1, 32(%rdx)
; CHECK-NEXT:    vmovdqa %ymm0, (%rdx)
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
 %d = load <16 x i32>, <16 x i32>* %a
 %e = load <16 x i32>, <16 x i32>* %b
 %f = add <16 x i32> %d, %e
 store <16 x i32> %f, <16 x i32>* %c
 ret void
}

define
 void @add512(<16 x i32>* %a, <16 x i32>* %b, <16 x i32>* %c) "min-legal-vector-width"="512" {
; CHECK-LABEL: add512:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vmovdqa64 (%rdi), %zmm0
; CHECK-NEXT:    vpaddd (%rsi), %zmm0, %zmm0
; CHECK-NEXT:    vmovdqa64 %zmm0, (%rdx)
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
 %d = load <16 x i32>, <16 x i32>* %a
 %e = load <16 x i32>, <16 x i32>* %b
 %f = add <16 x i32> %d, %e
 store <16 x i32> %f, <16 x i32>* %c
 ret void
}

define void @avg_v64i8_256(<64 x i8>* %a, <64 x i8>* %b) "min-legal-vector-width"="256" {
; CHECK-LABEL: avg_v64i8_256:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vmovdqa (%rsi), %ymm0
; CHECK-NEXT:    vmovdqa 32(%rsi), %ymm1
; CHECK-NEXT:    vpavgb (%rdi), %ymm0, %ymm0
; CHECK-NEXT:    vpavgb 32(%rdi), %ymm1, %ymm1
; CHECK-NEXT:    vmovdqu %ymm1, (%rax)
; CHECK-NEXT:    vmovdqu %ymm0, (%rax)
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
 %1 = load <64 x i8>, <64 x i8>* %a
```

```
 %2 = load <64 x i8>, <64 x i8>*
 %b
 %3 = zext <64 x i8> %1 to <64 x i32>
 %4 = zext <64 x i8> %2 to <64 x i32>
 %5 = add nuw nsw <64 x i32> %3, <i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1,
i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32
1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1,
i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1>
 %6 = add nuw nsw <64 x i32> %5, %4
 %7 = lshr <64 x i32> %6, <i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1,
i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32
1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1,
i32 1, i32
 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1>
 %8 = trunc <64 x i32> %7 to <64 x i8>
 store <64 x i8> %8, <64 x i8>* undef, align 4
 ret void
}


define void @avg_v64i8_512(<64 x i8>* %a, <64 x i8>* %b) "min-legal-vector-width"="512" {
; CHECK-LABEL: avg_v64i8_512:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vmovdqa64 (%rsi), %zmm0
; CHECK-NEXT:    vpavgb (%rdi), %zmm0, %zmm0
; CHECK-NEXT:    vmovdqu64 %zmm0, (%rax)
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
 %1 = load <64 x i8>, <64 x i8>* %a
 %2 = load <64 x i8>, <64 x i8>* %b
 %3 = zext <64 x i8> %1 to <64 x i32>
 %4 = zext <64 x i8> %2 to <64 x i32>
 %5 = add nuw nsw <64 x i32> %3, <i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1,
i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32
1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1,
 i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1,
i32 1, i32 1>
 %6 = add nuw nsw <64 x i32> %5, %4
 %7 = lshr <64 x i32> %6, <i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1,
i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32
1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1,
i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1>
 %8 = trunc <64 x i32> %7 to <64 x i8>
 store <64 x i8> %8, <64 x i8>* undef, align 4
 ret void
}

define void @pmaddwd_32_256(<32 x i16>* %APtr, <32 x i16>* %BPtr, <16 x i32>* %CPtr) "min-legal-vector-
```

```
width"="256" {
; CHECK-LABEL: pmaddwd_32_256:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vmovdqa (%rdi), %ymm0
; CHECK-NEXT:    vmovdqa 32(%rdi),
 %ymm1
; CHECK-NEXT:    vpmaddwd (%rsi), %ymm0, %ymm0
; CHECK-NEXT:    vpmaddwd 32(%rsi), %ymm1, %ymm1
; CHECK-NEXT:    vmovdqa %ymm1, 32(%rdx)
; CHECK-NEXT:    vmovdqa %ymm0, (%rdx)
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
  %A = load <32 x i16>, <32 x i16>* %APtr
  %B = load <32 x i16>, <32 x i16>* %BPtr
  %a = sext <32 x i16> %A to <32 x i32>
  %b = sext <32 x i16> %B to <32 x i32>
  %m = mul nsw <32 x i32> %a, %b
  %odd = shufflevector <32 x i32> %m, <32 x i32> undef, <16 x i32> <i32 0, i32 2, i32 4, i32 6, i32 8, i32 10, i32
12, i32 14, i32 16, i32 18, i32 20, i32 22, i32 24, i32 26, i32 28, i32 30>
  %even = shufflevector <32 x i32> %m, <32 x i32> undef, <16 x i32> <i32 1, i32 3, i32 5, i32 7, i32 9, i32 11, i32
13, i32 15, i32 17, i32 19, i32 21, i32 23, i32 25, i32 27, i32 29, i32 31>
  %ret = add <16 x i32> %odd, %even
  store <16 x i32> %ret, <16 x i32>* %CPtr
  ret void
}

define void @pmaddwd_32_512(<32 x i16>* %APtr, <32 x i16>* %BPtr, <16 x i32>* %CPtr)
 "min-legal-vector-width"="512" {
; CHECK-LABEL: pmaddwd_32_512:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vmovdqa64 (%rdi), %zmm0
; CHECK-NEXT:    vpmaddwd (%rsi), %zmm0, %zmm0
; CHECK-NEXT:    vmovdqa64 %zmm0, (%rdx)
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
  %A = load <32 x i16>, <32 x i16>* %APtr
  %B = load <32 x i16>, <32 x i16>* %BPtr
  %a = sext <32 x i16> %A to <32 x i32>
  %b = sext <32 x i16> %B to <32 x i32>
  %m = mul nsw <32 x i32> %a, %b
  %odd = shufflevector <32 x i32> %m, <32 x i32> undef, <16 x i32> <i32 0, i32 2, i32 4, i32 6, i32 8, i32 10, i32
12, i32 14, i32 16, i32 18, i32 20, i32 22, i32 24, i32 26, i32 28, i32 30>
  %even = shufflevector <32 x i32> %m, <32 x i32> undef, <16 x i32> <i32 1, i32 3, i32 5, i32 7, i32 9, i32 11, i32
13, i32 15, i32 17, i32 19, i32 21, i32 23, i32 25, i32 27, i32 29, i32 31>
  %ret = add <16 x i32> %odd, %even
  store <16 x i32> %ret, <16 x i32>* %CPtr
  ret void
}
```

```
define void @psubus_64i8_max_256(<64 x i8>* %xptr,
 <64 x i8>* %yptr, <64 x i8>* %zptr) "min-legal-vector-width"="256" {
; CHECK-LABEL: psubus_64i8_max_256:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vmovdqa (%rdi), %ymm0
; CHECK-NEXT:    vmovdqa 32(%rdi), %ymm1
; CHECK-NEXT:    vpsubusb (%rsi), %ymm0, %ymm0
; CHECK-NEXT:    vpsubusb 32(%rsi), %ymm1, %ymm1
; CHECK-NEXT:    vmovdqa %ymm1, 32(%rdx)
; CHECK-NEXT:    vmovdqa %ymm0, (%rdx)
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
 %x = load <64 x i8>, <64 x i8>* %xptr
 %y = load <64 x i8>, <64 x i8>* %yptr
 %cmp = icmp ult <64 x i8> %x, %y
 %max = select <64 x i1> %cmp, <64 x i8> %y, <64 x i8> %x
 %res = sub <64 x i8> %max, %y
 store <64 x i8> %res, <64 x i8>* %zptr
 ret void
}

define void @psubus_64i8_max_512(<64 x i8>* %xptr, <64 x i8>* %yptr, <64 x i8>* %zptr) "min-legal-vector-
width"="512" {
; CHECK-LABEL: psubus_64i8_max_512:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vmovdqa64 (%rdi), %zmm0
; CHECK-NEXT:    vpsubusb (%rsi), %zmm0, %zmm0
; CHECK-NEXT:    vmovdqa64 %zmm0,
 (%rdx)
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
 %x = load <64 x i8>, <64 x i8>* %xptr
 %y = load <64 x i8>, <64 x i8>* %yptr
 %cmp = icmp ult <64 x i8> %x, %y
 %max = select <64 x i1> %cmp, <64 x i8> %y, <64 x i8> %x
 %res = sub <64 x i8> %max, %y
 store <64 x i8> %res, <64 x i8>* %zptr
 ret void
}

define i32 @_Z9test_charPcS_i_256(i8* nocapture readonly, i8* nocapture readonly, i32) "min-legal-vector-
width"="256" {
; CHECK-LABEL: _Z9test_charPcS_i_256:
; CHECK:       # %bb.0: # %entry
; CHECK-NEXT:    movl %edx, %eax
; CHECK-NEXT:    vpxor %xmm0, %xmm0, %xmm0
; CHECK-NEXT:    xorl %ecx, %ecx
```

```
; CHECK-NEXT:    vpxor %xmm1, %xmm1, %xmm1
; CHECK-NEXT:    vpxor %xmm2, %xmm2, %xmm2
; CHECK-NEXT:    .p2align 4, 0x90
; CHECK-NEXT:  .LBB8_1: # %vector.body
; CHECK-NEXT:    # =>This Inner Loop Header: Depth=1
; CHECK-NEXT:    vpmovsxbw (%rdi,%rcx), %ymm3
; CHECK-NEXT:    vpmovsxbw 16(%rdi,%rcx), %ymm4
; CHECK-NEXT:    vpmovsxbw (%rsi,%rcx), %ymm5
; CHECK-NEXT:    vpmaddwd %ymm3,
 %ymm5, %ymm3
; CHECK-NEXT:    vpaddd %ymm1, %ymm3, %ymm1
; CHECK-NEXT:    vpmovsxbw 16(%rsi,%rcx), %ymm3
; CHECK-NEXT:    vpmaddwd %ymm4, %ymm3, %ymm3
; CHECK-NEXT:    vpaddd %ymm2, %ymm3, %ymm2
; CHECK-NEXT:    addq $32, %rcx
; CHECK-NEXT:    cmpq %rcx, %rax
; CHECK-NEXT:    jne .LBB8_1
; CHECK-NEXT:  # %bb.2: # %middle.block
; CHECK-NEXT:    vpaddd %ymm0, %ymm1, %ymm1
; CHECK-NEXT:    vpaddd %ymm0, %ymm2, %ymm0
; CHECK-NEXT:    vpaddd %ymm0, %ymm1, %ymm0
; CHECK-NEXT:    vextracti128 $1, %ymm0, %xmm1
; CHECK-NEXT:    vpaddd %ymm1, %ymm0, %ymm0
; CHECK-NEXT:    vpshufd {{.*#+}} xmm1 = xmm0[2,3,0,1]
; CHECK-NEXT:    vpaddd %ymm1, %ymm0, %ymm0
; CHECK-NEXT:    vpshufd {{.*#+}} xmm1 = xmm0[1,1,2,3]
; CHECK-NEXT:    vpaddd %xmm1, %xmm0, %xmm0
; CHECK-NEXT:    vmovd %xmm0, %eax
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
entry:
 %3 = zext i32 %2 to i64
 br label %vector.body

vector.body:
 %index = phi i64 [ %index.next, %vector.body ], [ 0, %entry ]
 %vec.phi = phi <32 x i32>
 [ %11, %vector.body ], [ zeroinitializer, %entry ]
 %4 = getelementptr inbounds i8, i8* %0, i64 %index
 %5 = bitcast i8* %4 to <32 x i8>*
 %wide.load = load <32 x i8>, <32 x i8>* %5, align 1
 %6 = sext <32 x i8> %wide.load to <32 x i32>
 %7 = getelementptr inbounds i8, i8* %1, i64 %index
 %8 = bitcast i8* %7 to <32 x i8>*
 %wide.load14 = load <32 x i8>, <32 x i8>* %8, align 1
 %9 = sext <32 x i8> %wide.load14 to <32 x i32>
 %10 = mul nsw <32 x i32> %9, %6
 %11 = add nsw <32 x i32> %10, %vec.phi
```

```
 %index.next = add i64 %index, 32
 %12 = icmp eq i64 %index.next, %3
 br i1 %12, label %middle.block, label %vector.body

middle.block:
 %rdx.shuf1 = shufflevector <32 x i32> %11, <32 x i32> undef, <32 x i32> <i32 16, i32 17, i32 18, i32 19, i32 20, i32 21, i32 22, i32 23, i32 24, i32 25, i32 26, i32 27, i32 28, i32 29, i32 30, i32 31, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
 %bin.rdx1 = add <32 x i32> %11, %rdx.shuf1
 %rdx.shuf = shufflevector <32 x i32> %bin.rdx1, <32 x i32> undef, <32 x i32> <i32 8, i32 9, i32 10, i32 11, i32 12, i32 13, i32 14, i32 15, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
 %bin.rdx = add <32 x i32> %bin.rdx1, %rdx.shuf
 %rdx.shuf15 = shufflevector <32 x i32> %bin.rdx, <32 x i32> undef, <32 x i32> <i32 4, i32 5, i32 6, i32 7, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
 %bin.rdx32 = add <32 x i32> %bin.rdx, %rdx.shuf15
  %rdx.shuf17 = shufflevector <32 x i32> %bin.rdx32, <32 x i32> undef, <32 x i32> <i32 2, i32 3, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
 %bin.rdx18 = add <32 x i32> %bin.rdx32, %rdx.shuf17
 %rdx.shuf19 = shufflevector <32 x i32> %bin.rdx18, <32 x i32> undef, <32 x i32> <i32 1, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
 %bin.rdx20 = add <32 x i32> %bin.rdx18, %rdx.shuf19
 %13 = extractelement <32 x i32> %bin.rdx20,
 i32 0
 ret i32 %13
}

define i32 @_Z9test_charPcS_i_512(i8* nocapture readonly, i8* nocapture readonly, i32) "min-legal-vector-width"="512" {
; CHECK-LABEL: _Z9test_charPcS_i_512:
; CHECK:       # %bb.0: # %entry
; CHECK-NEXT:    movl %edx, %eax
; CHECK-NEXT:    vpxor %xmm0, %xmm0, %xmm0
; CHECK-NEXT:    xorl %ecx, %ecx
; CHECK-NEXT:    vpxor %xmm1, %xmm1, %xmm1
; CHECK-NEXT:    .p2align 4, 0x90
; CHECK-NEXT:  .LBB9_1: # %vector.body
; CHECK-NEXT:    # =>This Inner Loop Header: Depth=1
; CHECK-NEXT:    vpmovsxbw (%rdi,%rcx), %zmm2
; CHECK-NEXT:    vpmovsxbw (%rsi,%rcx), %zmm3
```

```
; CHECK-NEXT:    vpmaddwd %zmm2, %zmm3, %zmm2
; CHECK-NEXT:    vpaddd %zmm1, %zmm2, %zmm1
; CHECK-NEXT:    addq $32, %rcx
; CHECK-NEXT:    cmpq %rcx, %rax
; CHECK-NEXT:    jne .LBB9_1
; CHECK-NEXT:  # %bb.2: # %middle.block
; CHECK-NEXT:    vpaddd %zmm0, %zmm1, %zmm0
; CHECK-NEXT:    vextracti64x4 $1, %zmm0, %ymm1
; CHECK-NEXT:    vpaddd %zmm1, %zmm0, %zmm0
; CHECK-NEXT:    vextracti128 $1, %ymm0, %xmm1
;
 CHECK-NEXT:    vpaddd %zmm1, %zmm0, %zmm0
; CHECK-NEXT:    vpshufd {{.*#+}} xmm1 = xmm0[2,3,0,1]
; CHECK-NEXT:    vpaddd %zmm1, %zmm0, %zmm0
; CHECK-NEXT:    vpshufd {{.*#+}} xmm1 = xmm0[1,1,2,3]
; CHECK-NEXT:    vpaddd %xmm1, %xmm0, %xmm0
; CHECK-NEXT:    vmovd %xmm0, %eax
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
entry:
 %3 = zext i32 %2 to i64
 br label %vector.body

vector.body:
 %index = phi i64 [ %index.next, %vector.body ], [ 0, %entry ]
 %vec.phi = phi <32 x i32> [ %11, %vector.body ], [ zeroinitializer, %entry ]
 %4 = getelementptr inbounds i8, i8* %0, i64 %index
 %5 = bitcast i8* %4 to <32 x i8>*
 %wide.load = load <32 x i8>, <32 x i8>* %5, align 1
 %6 = sext <32 x i8> %wide.load to <32 x i32>
 %7 = getelementptr inbounds i8, i8* %1, i64 %index
 %8 = bitcast i8* %7 to <32 x i8>*
 %wide.load14 = load <32 x i8>, <32 x i8>* %8, align 1
 %9 = sext <32 x i8> %wide.load14 to <32 x i32>
 %10 = mul nsw <32 x i32> %9, %6
 %11 = add nsw <32 x i32> %10, %vec.phi
  %index.next = add i64 %index, 32
 %12 = icmp eq i64 %index.next, %3
 br i1 %12, label %middle.block, label %vector.body

middle.block:
 %rdx.shuf1 = shufflevector <32 x i32> %11, <32 x i32> undef, <32 x i32> <i32 16, i32 17, i32 18, i32 19, i32 20,
i32 21, i32 22, i32 23, i32 24, i32 25, i32 26, i32 27, i32 28, i32 29, i32 30, i32 31, i32 undef, i32 undef, i32 undef,
i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32
undef, i32 undef, i32 undef>
 %bin.rdx1 = add <32 x i32> %11, %rdx.shuf1
 %rdx.shuf = shufflevector <32 x i32> %bin.rdx1, <32 x i32> undef, <32 x i32> <i32 8, i32 9, i32 10, i32 11, i32
12, i32 13, i32 14, i32 15, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32
```

```
undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef,
i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
 %bin.rdx = add <32 x i32> %bin.rdx1,
 %rdx.shuf
 %rdx.shuf15 = shufflevector <32 x i32> %bin.rdx, <32 x i32> undef, <32 x i32> <i32 4, i32 5, i32 6, i32 7, i32
undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef,
i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32
undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
 %bin.rdx32 = add <32 x i32> %bin.rdx, %rdx.shuf15
 %rdx.shuf17 = shufflevector <32 x i32> %bin.rdx32, <32 x i32> undef, <32 x i32> <i32 2, i32 3, i32 undef, i32
undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef,
i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32
undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
 %bin.rdx18 = add <32 x i32> %bin.rdx32, %rdx.shuf17
 %rdx.shuf19 = shufflevector
 <32 x i32> %bin.rdx18, <32 x i32> undef, <32 x i32> <i32 1, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef,
i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32
undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef,
i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
 %bin.rdx20 = add <32 x i32> %bin.rdx18, %rdx.shuf19
 %13 = extractelement <32 x i32> %bin.rdx20, i32 0
 ret i32 %13
}

@a = global [1024 x i8] zeroinitializer, align 16
@b = global [1024 x i8] zeroinitializer, align 16

define i32 @sad_16i8_256() "min-legal-vector-width"="256" {
; CHECK-LABEL: sad_16i8_256:
; CHECK:       # %bb.0: # %entry
; CHECK-NEXT:    vpxor %xmm0, %xmm0, %xmm0
; CHECK-NEXT:    movq $-1024, %rax # imm = 0xFC00
; CHECK-NEXT:    vpxor %xmm1, %xmm1, %xmm1
; CHECK-NEXT:    .p2align 4, 0x90
; CHECK-NEXT:  .LBB10_1: # %vector.body
; CHECK-NEXT:    # =>This Inner Loop Header:
 Depth=1
; CHECK-NEXT:    vmovdqu a+1024(%rax), %xmm2
; CHECK-NEXT:    vpsadbw b+1024(%rax), %xmm2, %xmm2
; CHECK-NEXT:    vpaddd %ymm1, %ymm2, %ymm1
; CHECK-NEXT:    addq $4, %rax
; CHECK-NEXT:    jne .LBB10_1
; CHECK-NEXT:  # %bb.2: # %middle.block
; CHECK-NEXT:    vpaddd %ymm0, %ymm1, %ymm0
; CHECK-NEXT:    vextracti128 $1, %ymm0, %xmm1
; CHECK-NEXT:    vpaddd %ymm1, %ymm0, %ymm0
; CHECK-NEXT:    vpshufd {{.*#+}} xmm1 = xmm0[2,3,0,1]
; CHECK-NEXT:    vpaddd %ymm1, %ymm0, %ymm0
```

```
; CHECK-NEXT:    vpshufd {{.*#+}} xmm1 = xmm0[1,1,2,3]
; CHECK-NEXT:    vpaddd %xmm1, %xmm0, %xmm0
; CHECK-NEXT:    vmovd %xmm0, %eax
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
entry:
 br label %vector.body

vector.body:
 %index = phi i64 [ 0, %entry ], [ %index.next, %vector.body ]
 %vec.phi = phi <16 x i32> [ zeroinitializer, %entry ], [ %10, %vector.body ]
 %0 = getelementptr inbounds [1024 x i8], [1024 x i8]* @a, i64 0, i64 %index
 %1 = bitcast i8* %0 to <16 x i8>*
 %wide.load = load <16 x i8>,
 <16 x i8>* %1, align 4
 %2 = zext <16 x i8> %wide.load to <16 x i32>
 %3 = getelementptr inbounds [1024 x i8], [1024 x i8]* @b, i64 0, i64 %index
 %4 = bitcast i8* %3 to <16 x i8>*
 %wide.load1 = load <16 x i8>, <16 x i8>* %4, align 4
 %5 = zext <16 x i8> %wide.load1 to <16 x i32>
 %6 = sub nsw <16 x i32> %2, %5
 %7 = icmp sgt <16 x i32> %6, <i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1>
 %8 = sub nsw <16 x i32> zeroinitializer, %6
 %9 = select <16 x i1> %7, <16 x i32> %6, <16 x i32> %8
 %10 = add nsw <16 x i32> %9, %vec.phi
 %index.next = add i64 %index, 4
 %11 = icmp eq i64 %index.next, 1024
 br i1 %11, label %middle.block, label %vector.body

middle.block:
 %.lcssa = phi <16 x i32> [ %10, %vector.body ]
 %rdx.shuf = shufflevector <16 x i32> %.lcssa, <16 x i32> undef, <16 x i32> <i32 8, i32 9, i32 10, i32 11, i32 12, i32 13, i32 14, i32 15, i32 undef, i32 undef, i32 undef, i32 undef, i32
 undef, i32 undef, i32 undef, i32 undef>
 %bin.rdx = add <16 x i32> %.lcssa, %rdx.shuf
 %rdx.shuf2 = shufflevector <16 x i32> %bin.rdx, <16 x i32> undef, <16 x i32> <i32 4, i32 5, i32 6, i32 7, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
 %bin.rdx2 = add <16 x i32> %bin.rdx, %rdx.shuf2
 %rdx.shuf3 = shufflevector <16 x i32> %bin.rdx2, <16 x i32> undef, <16 x i32> <i32 2, i32 3, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
 %bin.rdx3 = add <16 x i32> %bin.rdx2, %rdx.shuf3
 %rdx.shuf4 = shufflevector <16 x i32> %bin.rdx3, <16 x i32> undef, <16 x i32> <i32 1, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
 %bin.rdx4 = add <16 x i32> %bin.rdx3, %rdx.shuf4
```

```
 %12 = extractelement
 <16 x i32> %bin.rdx4, i32 0
 ret i32 %12
}

define i32 @sad_16i8_512() "min-legal-vector-width"="512" {
; CHECK-LABEL: sad_16i8_512:
; CHECK:       # %bb.0: # %entry
; CHECK-NEXT:    vpxor %xmm0, %xmm0, %xmm0
; CHECK-NEXT:    movq $-1024, %rax # imm = 0xFC00
; CHECK-NEXT:    .p2align 4, 0x90
; CHECK-NEXT:  .LBB11_1: # %vector.body
; CHECK-NEXT:    # =>This Inner Loop Header: Depth=1
; CHECK-NEXT:    vmovdqu a+1024(%rax), %xmm1
; CHECK-NEXT:    vpsadbw b+1024(%rax), %xmm1, %xmm1
; CHECK-NEXT:    vpaddd %zmm0, %zmm1, %zmm0
; CHECK-NEXT:    addq $4, %rax
; CHECK-NEXT:    jne .LBB11_1
; CHECK-NEXT:  # %bb.2: # %middle.block
; CHECK-NEXT:    vextracti64x4 $1, %zmm0, %ymm1
; CHECK-NEXT:    vpaddd %zmm1, %zmm0, %zmm0
; CHECK-NEXT:    vextracti128 $1, %ymm0, %xmm1
; CHECK-NEXT:    vpaddd %zmm1, %zmm0, %zmm0
; CHECK-NEXT:    vpshufd {{.*#+}} xmm1 = xmm0[2,3,0,1]
; CHECK-NEXT:    vpaddd %zmm1, %zmm0, %zmm0
; CHECK-NEXT:    vpshufd {{.*#+}} xmm1 = xmm0[1,1,2,3]
; CHECK-NEXT:    vpaddd %xmm1,
 %xmm0, %xmm0
; CHECK-NEXT:    vmovd %xmm0, %eax
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
entry:
 br label %vector.body

vector.body:
 %index = phi i64 [ 0, %entry ], [ %index.next, %vector.body ]
 %vec.phi = phi <16 x i32> [ zeroinitializer, %entry ], [ %10, %vector.body ]
 %0 = getelementptr inbounds [1024 x i8], [1024 x i8]* @a, i64 0, i64 %index
 %1 = bitcast i8* %0 to <16 x i8>*
 %wide.load = load <16 x i8>, <16 x i8>* %1, align 4
 %2 = zext <16 x i8> %wide.load to <16 x i32>
 %3 = getelementptr inbounds [1024 x i8], [1024 x i8]* @b, i64 0, i64 %index
 %4 = bitcast i8* %3 to <16 x i8>*
 %wide.load1 = load <16 x i8>, <16 x i8>* %4, align 4
 %5 = zext <16 x i8> %wide.load1 to <16 x i32>
 %6 = sub nsw <16 x i32> %2, %5
 %7 = icmp sgt <16 x i32> %6, <i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1>
```

```
 %8 = sub nsw <16 x i32> zeroinitializer, %6
 %9 = select <16 x i1> %7, <16 x i32>
 %6, <16 x i32> %8
 %10 = add nsw <16 x i32> %9, %vec.phi
 %index.next = add i64 %index, 4
 %11 = icmp eq i64 %index.next, 1024
 br i1 %11, label %middle.block, label %vector.body

middle.block:
 %.lcssa = phi <16 x i32> [ %10, %vector.body ]
 %rdx.shuf = shufflevector <16 x i32> %.lcssa, <16 x i32> undef, <16 x i32> <i32 8, i32 9, i32 10, i32 11, i32 12,
i32 13, i32 14, i32 15, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
 %bin.rdx = add <16 x i32> %.lcssa, %rdx.shuf
 %rdx.shuf2 = shufflevector <16 x i32> %bin.rdx, <16 x i32> undef, <16 x i32> <i32 4, i32 5, i32 6, i32 7, i32
undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef,
i32 undef>
 %bin.rdx2 = add <16 x i32> %bin.rdx, %rdx.shuf2
 %rdx.shuf3 = shufflevector <16 x i32> %bin.rdx2, <16 x i32> undef, <16 x i32> <i32 2, i32 3, i32 undef, i32
undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef,
 i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
 %bin.rdx3 = add <16 x i32> %bin.rdx2, %rdx.shuf3
 %rdx.shuf4 = shufflevector <16 x i32> %bin.rdx3, <16 x i32> undef, <16 x i32> <i32 1, i32 undef, i32 undef, i32
undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef,
i32 undef, i32 undef>
 %bin.rdx4 = add <16 x i32> %bin.rdx3, %rdx.shuf4
 %12 = extractelement <16 x i32> %bin.rdx4, i32 0
 ret i32 %12
}

define void @sbto16f32_256(<16 x i16> %a, <16 x float>* %res) "min-legal-vector-width"="256" {
; CHECK-LABEL: sbto16f32_256:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vpmovw2m %ymm0, %k0
; CHECK-NEXT:    kshiftrw $8, %k0, %k1
; CHECK-NEXT:    vpmovm2d %k1, %ymm0
; CHECK-NEXT:    vcvtdq2ps %ymm0, %ymm0
; CHECK-NEXT:    vpmovm2d %k0, %ymm1
; CHECK-NEXT:    vcvtdq2ps %ymm1, %ymm1
; CHECK-NEXT:    vmovaps %ymm1, (%rdi)
; CHECK-NEXT:    vmovaps %ymm0, 32(%rdi)
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
 %mask
 = icmp slt <16 x i16> %a, zeroinitializer
 %1 = sitofp <16 x i1> %mask to <16 x float>
 store <16 x float> %1, <16 x float>* %res
 ret void
}
```

```
define void @sbto16f32_512(<16 x i16> %a, <16 x float>* %res) "min-legal-vector-width"="512" {
; CHECK-LABEL: sbto16f32_512:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vpmovw2m %ymm0, %k0
; CHECK-NEXT:    vpmovm2d %k0, %zmm0
; CHECK-NEXT:    vcvtdq2ps %zmm0, %zmm0
; CHECK-NEXT:    vmovaps %zmm0, (%rdi)
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
 %mask = icmp slt <16 x i16> %a, zeroinitializer
 %1 = sitofp <16 x i1> %mask to <16 x float>
 store <16 x float> %1, <16 x float>* %res
 ret void
}

define void @sbto16f64_256(<16 x i16> %a, <16 x double>* %res)  "min-legal-vector-width"="256" {
; CHECK-LABEL: sbto16f64_256:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vpmovw2m %ymm0, %k0
; CHECK-NEXT:    kshiftrw $8, %k0, %k1
; CHECK-NEXT:    vpmovm2d %k1, %ymm0
; CHECK-NEXT:    vcvtdq2pd %xmm0, %ymm1
; CHECK-NEXT:    vextracti128 $1, %ymm0, %xmm0
;
 CHECK-NEXT:    vcvtdq2pd %xmm0, %ymm0
; CHECK-NEXT:    vpmovm2d %k0, %ymm2
; CHECK-NEXT:    vcvtdq2pd %xmm2, %ymm3
; CHECK-NEXT:    vextracti128 $1, %ymm2, %xmm2
; CHECK-NEXT:    vcvtdq2pd %xmm2, %ymm2
; CHECK-NEXT:    vmovaps %ymm2, 32(%rdi)
; CHECK-NEXT:    vmovaps %ymm3, (%rdi)
; CHECK-NEXT:    vmovaps %ymm0, 96(%rdi)
; CHECK-NEXT:    vmovaps %ymm1, 64(%rdi)
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
 %mask = icmp slt <16 x i16> %a, zeroinitializer
 %1 = sitofp <16 x i1> %mask to <16 x double>
 store <16 x double> %1, <16 x double>* %res
 ret void
}

define void @sbto16f64_512(<16 x i16> %a, <16 x double>* %res)  "min-legal-vector-width"="512" {
; CHECK-LABEL: sbto16f64_512:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vpmovw2m %ymm0, %k0
; CHECK-NEXT:    vpmovm2d %k0, %zmm0
; CHECK-NEXT:    vcvtdq2pd %ymm0, %zmm1
```

```
; CHECK-NEXT:    vextracti64x4 $1, %zmm0, %ymm0
; CHECK-NEXT:    vcvtdq2pd %ymm0, %zmm0
; CHECK-NEXT:    vmovaps %zmm0, 64(%rdi)
; CHECK-NEXT:    vmovaps %zmm1,
 (%rdi)
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
 %mask = icmp slt <16 x i16> %a, zeroinitializer
 %1 = sitofp <16 x i1> %mask to <16 x double>
 store <16 x double> %1, <16 x double>* %res
 ret void
}

define void @ubto16f32_256(<16 x i16> %a, <16 x float>* %res) "min-legal-vector-width"="256" {
; CHECK-LABEL: ubto16f32_256:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vpmovw2m %ymm0, %k0
; CHECK-NEXT:    kshiftrw $8, %k0, %k1
; CHECK-NEXT:    vpmovm2d %k1, %ymm0
; CHECK-NEXT:    vpsrld $31, %ymm0, %ymm0
; CHECK-NEXT:    vcvtdq2ps %ymm0, %ymm0
; CHECK-NEXT:    vpmovm2d %k0, %ymm1
; CHECK-NEXT:    vpsrld $31, %ymm1, %ymm1
; CHECK-NEXT:    vcvtdq2ps %ymm1, %ymm1
; CHECK-NEXT:    vmovaps %ymm1, (%rdi)
; CHECK-NEXT:    vmovaps %ymm0, 32(%rdi)
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
 %mask = icmp slt <16 x i16> %a, zeroinitializer
 %1 = uitofp <16 x i1> %mask to <16 x float>
 store <16 x float> %1, <16 x float>* %res
 ret void
}

define void @ubto16f32_512(<16 x i16>
 %a, <16 x float>* %res) "min-legal-vector-width"="512" {
; CHECK-LABEL: ubto16f32_512:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vpmovw2m %ymm0, %k0
; CHECK-NEXT:    vpmovm2d %k0, %zmm0
; CHECK-NEXT:    vpsrld $31, %zmm0, %zmm0
; CHECK-NEXT:    vcvtdq2ps %zmm0, %zmm0
; CHECK-NEXT:    vmovaps %zmm0, (%rdi)
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
 %mask = icmp slt <16 x i16> %a, zeroinitializer
 %1 = uitofp <16 x i1> %mask to <16 x float>
 store <16 x float> %1, <16 x float>* %res
```

```
 ret void
}

define void @ubto16f64_256(<16 x i16> %a, <16 x double>* %res) "min-legal-vector-width"="256" {
; CHECK-LABEL: ubto16f64_256:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vpmovw2m %ymm0, %k0
; CHECK-NEXT:    kshiftrw $8, %k0, %k1
; CHECK-NEXT:    vpmovm2d %k1, %ymm0
; CHECK-NEXT:    vpsrld $31, %ymm0, %ymm0
; CHECK-NEXT:    vcvtdq2pd %xmm0, %ymm1
; CHECK-NEXT:    vextracti128 $1, %ymm0, %xmm0
; CHECK-NEXT:    vcvtdq2pd %xmm0, %ymm0
; CHECK-NEXT:    vpmovm2d %k0, %ymm2
; CHECK-NEXT:    vpsrld
 $31, %ymm2, %ymm2
; CHECK-NEXT:    vcvtdq2pd %xmm2, %ymm3
; CHECK-NEXT:    vextracti128 $1, %ymm2, %xmm2
; CHECK-NEXT:    vcvtdq2pd %xmm2, %ymm2
; CHECK-NEXT:    vmovaps %ymm2, 32(%rdi)
; CHECK-NEXT:    vmovaps %ymm3, (%rdi)
; CHECK-NEXT:    vmovaps %ymm0, 96(%rdi)
; CHECK-NEXT:    vmovaps %ymm1, 64(%rdi)
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
 %mask = icmp slt <16 x i16> %a, zeroinitializer
 %1 = uitofp <16 x i1> %mask to <16 x double>
 store <16 x double> %1, <16 x double>* %res
 ret void
}

define void @ubto16f64_512(<16 x i16> %a, <16 x double>* %res) "min-legal-vector-width"="512" {
; CHECK-LABEL: ubto16f64_512:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vpmovw2m %ymm0, %k0
; CHECK-NEXT:    vpmovm2d %k0, %zmm0
; CHECK-NEXT:    vpsrld $31, %zmm0, %zmm0
; CHECK-NEXT:    vcvtdq2pd %ymm0, %zmm1
; CHECK-NEXT:    vextracti64x4 $1, %zmm0, %ymm0
; CHECK-NEXT:    vcvtdq2pd %ymm0, %zmm0
; CHECK-NEXT:    vmovaps %zmm0, 64(%rdi)
; CHECK-NEXT:    vmovaps %zmm1, (%rdi)
; CHECK-NEXT:
   vzeroupper
; CHECK-NEXT:    retq
 %mask = icmp slt <16 x i16> %a, zeroinitializer
 %1 = uitofp <16 x i1> %mask to <16 x double>
 store <16 x double> %1, <16 x double>* %res
```

```
 ret void
}

define <16 x i16> @test_16f32toub_256(<16 x float>* %ptr, <16 x i16> %passthru) "min-legal-vector-
width"="256" {
; CHECK-LABEL: test_16f32toub_256:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vcvttps2dq (%rdi), %ymm1
; CHECK-NEXT:    vpslld $31, %ymm1, %ymm1
; CHECK-NEXT:    vpmovd2m %ymm1, %k0
; CHECK-NEXT:    vcvttps2dq 32(%rdi), %ymm1
; CHECK-NEXT:    vpslld $31, %ymm1, %ymm1
; CHECK-NEXT:    vpmovd2m %ymm1, %k1
; CHECK-NEXT:    kunpckbw %k0, %k1, %k1
; CHECK-NEXT:    vmovdqu16 %ymm0, %ymm0 {%k1} {z}
; CHECK-NEXT:    retq
 %a = load <16 x float>, <16 x float>* %ptr
 %mask = fptoui <16 x float> %a to <16 x i1>
 %select = select <16 x i1> %mask, <16 x i16> %passthru, <16 x i16> zeroinitializer
 ret <16 x i16> %select
}

define <16 x i16> @test_16f32toub_512(<16 x float>* %ptr, <16 x i16>
 %passthru) "min-legal-vector-width"="512" {
; CHECK-LABEL: test_16f32toub_512:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vcvttps2dq (%rdi), %zmm1
; CHECK-NEXT:    vpslld $31, %zmm1, %zmm1
; CHECK-NEXT:    vpmovd2m %zmm1, %k1
; CHECK-NEXT:    vmovdqu16 %ymm0, %ymm0 {%k1} {z}
; CHECK-NEXT:    retq
 %a = load <16 x float>, <16 x float>* %ptr
 %mask = fptoui <16 x float> %a to <16 x i1>
 %select = select <16 x i1> %mask, <16 x i16> %passthru, <16 x i16> zeroinitializer
 ret <16 x i16> %select
}

define <16 x i16> @test_16f32tosb_256(<16 x float>* %ptr, <16 x i16> %passthru) "min-legal-vector-
width"="256" {
; CHECK-LABEL: test_16f32tosb_256:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vcvttps2dq (%rdi), %ymm1
; CHECK-NEXT:    vpmovd2m %ymm1, %k0
; CHECK-NEXT:    vcvttps2dq 32(%rdi), %ymm1
; CHECK-NEXT:    vpmovd2m %ymm1, %k1
; CHECK-NEXT:    kunpckbw %k0, %k1, %k1
; CHECK-NEXT:    vmovdqu16 %ymm0, %ymm0 {%k1} {z}
; CHECK-NEXT:    retq
```

```
 %a = load <16 x float>, <16 x float>* %ptr
 %mask =
 fptosi <16 x float> %a to <16 x i1>
 %select = select <16 x i1> %mask, <16 x i16> %passthru, <16 x i16> zeroinitializer
 ret <16 x i16> %select
}

define <16 x i16> @test_16f32tosb_512(<16 x float>* %ptr, <16 x i16> %passthru) "min-legal-vector-
width"="512" {
; CHECK-LABEL: test_16f32tosb_512:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vcvttps2dq (%rdi), %zmm1
; CHECK-NEXT:    vpmovd2m %zmm1, %k1
; CHECK-NEXT:    vmovdqu16 %ymm0, %ymm0 {%k1} {z}
; CHECK-NEXT:    retq
 %a = load <16 x float>, <16 x float>* %ptr
 %mask = fptosi <16 x float> %a to <16 x i1>
 %select = select <16 x i1> %mask, <16 x i16> %passthru, <16 x i16> zeroinitializer
 ret <16 x i16> %select
}

define void @mul256(<64 x i8>* %a, <64 x i8>* %b, <64 x i8>* %c) "min-legal-vector-width"="256" {
; CHECK-LABEL: mul256:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vmovdqa (%rdi), %ymm0
; CHECK-NEXT:    vmovdqa 32(%rdi), %ymm1
; CHECK-NEXT:    vmovdqa (%rsi), %ymm2
; CHECK-NEXT:    vmovdqa 32(%rsi), %ymm3
; CHECK-NEXT:    vpunpckhbw
 {{.*#+}} ymm4 =
ymm2[8],ymm0[8],ymm2[9],ymm0[9],ymm2[10],ymm0[10],ymm2[11],ymm0[11],ymm2[12],ymm0[12],ymm2[13
],ymm0[13],ymm2[14],ymm0[14],ymm2[15],ymm0[15],ymm2[24],ymm0[24],ymm2[25],ymm0[25],ymm2[26],ym
m0[26],ymm2[27],ymm0[27],ymm2[28],ymm0[28],ymm2[29],ymm0[29],ymm2[30],ymm0[30],ymm2[31],ymm0[
31]
; CHECK-NEXT:    vpunpckhbw {{.*#+}} ymm5 =
ymm0[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31]
; CHECK-NEXT:    vpmullw %ymm4, %ymm5, %ymm4
; CHECK-NEXT:    vmovdqa {{.*#+}} ymm5 =
[255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255]
; CHECK-NEXT:    vpand %ymm5, %ymm4, %ymm4
; CHECK-NEXT:    vpunpcklbw {{.*#+}} ymm2 =
ymm2[0],ymm0[0],ymm2[1],ymm0[1],ymm2[2],ymm0[2],ymm2[3],ymm0[3],ymm2[4],ymm0[4],ymm2[5],ymm0[
5],ymm2[6],ymm0[6],ymm2[7],ymm0[7],ymm2[16],ymm0[16],ymm2[17],ymm0[17],ymm2[18],ymm0[18],ymm2[
19],ymm0[19],ymm2[20],ymm0[20],ymm2[21],ymm0[21],ymm2[22],ymm0[22],ymm2[23],ymm0[23]
; CHECK-NEXT:    vpunpcklbw {{.*#+}} ymm0
 = ymm0[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23]
; CHECK-NEXT:    vpmullw %ymm2, %ymm0, %ymm0
; CHECK-NEXT:    vpand %ymm5, %ymm0, %ymm0
```

```
; CHECK-NEXT:    vpackuswb %ymm4, %ymm0, %ymm0
; CHECK-NEXT:    vpunpckhbw {{.*#+}} ymm2 =
ymm3[8],ymm0[8],ymm3[9],ymm0[9],ymm3[10],ymm0[10],ymm3[11],ymm0[11],ymm3[12],ymm0[12],ymm3[13
],ymm0[13],ymm3[14],ymm0[14],ymm3[15],ymm0[15],ymm3[24],ymm0[24],ymm3[25],ymm0[25],ymm3[26],ym
m0[26],ymm3[27],ymm0[27],ymm3[28],ymm0[28],ymm3[29],ymm0[29],ymm3[30],ymm0[30],ymm3[31],ymm0[
31]
; CHECK-NEXT:    vpunpckhbw {{.*#+}} ymm4 =
ymm1[8],ymm0[8],ymm1[9],ymm0[9],ymm1[10],ymm0[10],ymm1[11],ymm0[11],ymm1[12],ymm0[12],ymm1[13
],ymm0[13],ymm1[14],ymm0[14],ymm1[15],ymm0[15],ymm1[24],ymm0[24],ymm1[25],ymm0[25],ymm1[26],ym
m0[26],ymm1[27],ymm0[27],ymm1[28],ymm0[28],ymm1[29],ymm0[29],ymm1[30],ymm0[30],ymm1[31],ymm0[
31]
; CHECK-NEXT:    vpmullw %ymm2, %ymm4, %ymm2
; CHECK-NEXT:    vpand %ymm5, %ymm2, %ymm2
; CHECK-NEXT:    vpunpcklbw {{.*#+}}
 ymm3 =
ymm3[0],ymm0[0],ymm3[1],ymm0[1],ymm3[2],ymm0[2],ymm3[3],ymm0[3],ymm3[4],ymm0[4],ymm3[5],ymm0[
5],ymm3[6],ymm0[6],ymm3[7],ymm0[7],ymm3[16],ymm0[16],ymm3[17],ymm0[17],ymm3[18],ymm0[18],ymm3[
19],ymm0[19],ymm3[20],ymm0[20],ymm3[21],ymm0[21],ymm3[22],ymm0[22],ymm3[23],ymm0[23]
; CHECK-NEXT:    vpunpcklbw {{.*#+}} ymm1 =
ymm1[0],ymm0[0],ymm1[1],ymm0[1],ymm1[2],ymm0[2],ymm1[3],ymm0[3],ymm1[4],ymm0[4],ymm1[5],ymm0[
5],ymm1[6],ymm0[6],ymm1[7],ymm0[7],ymm1[16],ymm0[16],ymm1[17],ymm0[17],ymm1[18],ymm0[18],ymm1[
19],ymm0[19],ymm1[20],ymm0[20],ymm1[21],ymm0[21],ymm1[22],ymm0[22],ymm1[23],ymm0[23]
; CHECK-NEXT:    vpmullw %ymm3, %ymm1, %ymm1
; CHECK-NEXT:    vpand %ymm5, %ymm1, %ymm1
; CHECK-NEXT:    vpackuswb %ymm2, %ymm1, %ymm1
; CHECK-NEXT:    vmovdqa %ymm1, 32(%rdx)
; CHECK-NEXT:    vmovdqa %ymm0, (%rdx)
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
 %d = load <64 x i8>, <64 x i8>* %a
 %e = load <64 x i8>, <64 x i8>* %b
 %f = mul <64 x i8> %d, %e
 store <64 x i8> %f, <64 x i8>*
 %c
 ret void
}

define void @mul512(<64 x i8>* %a, <64 x i8>* %b, <64 x i8>* %c) "min-legal-vector-width"="512" {
; CHECK-LABEL: mul512:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vmovdqa64 (%rdi), %zmm0
; CHECK-NEXT:    vmovdqa64 (%rsi), %zmm1
; CHECK-NEXT:    vpunpckhbw {{.*#+}} zmm2 =
zmm1[8],zmm0[8],zmm1[9],zmm0[9],zmm1[10],zmm0[10],zmm1[11],zmm0[11],zmm1[12],zmm0[12],zmm1[13],
zmm0[13],zmm1[14],zmm0[14],zmm1[15],zmm0[15],zmm1[24],zmm0[24],zmm1[25],zmm0[25],zmm1[26],zmm0
[26],zmm1[27],zmm0[27],zmm1[28],zmm0[28],zmm1[29],zmm0[29],zmm1[30],zmm0[30],zmm1[31],zmm0[31],z
mm1[40],zmm0[40],zmm1[41],zmm0[41],zmm1[42],zmm0[42],zmm1[43],zmm0[43],zmm1[44],zmm0[44],zmm1[
45],zmm0[45],zmm1[46],zmm0[46],zmm1[47],zmm0[47],zmm1[56],zmm0[56],zmm1[57],zmm0[57],zmm1[58],z
```

mm0[58],zmm1[59],zmm0[59],zmm1[60],zmm0[60],zmm1[61],zmm0[61],zmm1[62],zmm0[62],zmm1[63],zmm0[63]
; CHECK-NEXT: vpunpckhbw {{.*#+}} zmm3 = zmm0[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31,40,40,41,41,42,42,43,43,44,44,45,45,46,46,47,47,56,56,57,57,58,58,59,59,60,60,61,61,62,62,63,63]
;
 CHECK-NEXT: vpmullw %zmm2, %zmm3, %zmm2
; CHECK-NEXT: vmovdqa64 {{.*#+}} zmm3 = [255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255]
; CHECK-NEXT: vpandq %zmm3, %zmm2, %zmm2
; CHECK-NEXT: vpunpcklbw {{.*#+}} zmm1 = zmm1[0],zmm0[0],zmm1[1],zmm0[1],zmm1[2],zmm0[2],zmm1[3],zmm0[3],zmm1[4],zmm0[4],zmm1[5],zmm0[5],zmm1[6],zmm0[6],zmm1[7],zmm0[7],zmm1[16],zmm0[16],zmm1[17],zmm0[17],zmm1[18],zmm0[18],zmm1[19],zmm0[19],zmm1[20],zmm0[20],zmm1[21],zmm0[21],zmm1[22],zmm0[22],zmm1[23],zmm0[23],zmm1[32],zmm0[32],zmm1[33],zmm0[33],zmm1[34],zmm0[34],zmm1[35],zmm0[35],zmm1[36],zmm0[36],zmm1[37],zmm0[37],zmm1[38],zmm0[38],zmm1[39],zmm0[39],zmm1[48],zmm0[48],zmm1[49],zmm0[49],zmm1[50],zmm0[50],zmm1[51],zmm0[51],zmm1[52],zmm0[52],zmm1[53],zmm0[53],zmm1[54],zmm0[54],zmm1[55],zmm0[55]
; CHECK-NEXT: vpunpcklbw {{.*#+}}
 zmm0 = zmm0[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23,32,32,33,33,34,34,35,35,36,36,37,37,38,38,39,39,48,48,49,49,50,50,51,51,52,52,53,53,54,54,55,55]
; CHECK-NEXT: vpmullw %zmm1, %zmm0, %zmm0
; CHECK-NEXT: vpandq %zmm3, %zmm0, %zmm0
; CHECK-NEXT: vpackuswb %zmm2, %zmm0, %zmm0
; CHECK-NEXT: vmovdqa64 %zmm0, (%rdx)
; CHECK-NEXT: vzeroupper
; CHECK-NEXT: retq
 %d = load <64 x i8>, <64 x i8>* %a
 %e = load <64 x i8>, <64 x i8>* %b
 %f = mul <64 x i8> %d, %e
 store <64 x i8> %f, <64 x i8>* %c
 ret void
}
====================
LLVM Release License
====================
University of Illinois/NCSA
Open Source License

Developed by:

  LLVM Team

  University of Illinois at Urbana-Champaign

http://llvm.org

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal with the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimers.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimers
in the documentation and/or other materials provided with the distribution.

* Neither the names of the LLVM Team, University of Illinois at Urbana-Champaign, nor the names of its contributors may be used to endorse or promote products derived from this Software without specific prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

==========
The LLVM software contains code written by third parties. Such software will have its own individual LICENSE.TXT file in the directory in which it appears. This file will describe the copyrights, license, and restrictions which apply
to that code.

The disclaimer of warranty in the University of Illinois Open Source License applies to all code in the LLVM Distribution, and nothing in any of the other licenses gives permission to use the names of the LLVM Team or the University of Illinois to endorse or promote products derived from this Software.

The following pieces of software have additional or alternate copyrights, licenses, and/or restrictions:

Program Directory
------- ---------
<none yet>
; NOTE: Assertions have been autogenerated by utils/update_analyze_test_checks.py
; RUN: opt < %s -cost-model -analyze -mtriple=x86_64-apple-macosx10.8.0 -mattr=+avx2 | FileCheck %s --check-prefixes=CHECK,VEC256,AVX
; RUN: opt < %s -cost-model -analyze -mtriple=x86_64-apple-macosx10.8.0 -mattr=+avx512vl,+avx512bw,+avx512dq,+prefer-256-bit | FileCheck %s --check-prefixes=CHECK,VEC256,SKX256
; RUN: opt < %s -cost-model -analyze -mtriple=x86_64-apple-macosx10.8.0 -

```
mattr=+avx512vl,+avx512bw,+avx512dq,-prefer-256-bit | FileCheck %s --check-prefixes=CHECK,VEC512

define void @zext256() "min-legal-vector-width"="256" {
; VEC256-LABEL: 'zext256'
; VEC256-NEXT:  Cost Model: Found an estimated cost of 7 for instruction: %A = zext <8 x i16> undef to <8 x
i64>
; VEC256-NEXT:  Cost Model: Found an estimated cost of 3 for instruction: %B = zext <8 x i32> undef to <8 x
i64>
; VEC256-NEXT:  Cost Model: Found an estimated cost of 4 for instruction: %C = zext <16 x i8> undef to <16 x
i32>
; VEC256-NEXT:
  Cost Model: Found an estimated cost of 4 for instruction: %D = zext <16 x i16> undef to <16 x i32>
; VEC256-NEXT:  Cost Model: Found an estimated cost of 3 for instruction: %E = zext <32 x i8> undef to <32 x
i16>
; VEC256-NEXT:  Cost Model: Found an estimated cost of 0 for instruction: ret void
;
; VEC512-LABEL: 'zext256'
; VEC512-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %A = zext <8 x i16> undef to <8 x
i64>
; VEC512-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %B = zext <8 x i32> undef to <8 x
i64>
; VEC512-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %C = zext <16 x i8> undef to <16 x
i32>
; VEC512-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %D = zext <16 x i16> undef to <16 x
i32>
; VEC512-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %E = zext <32 x i8> undef to <32 x
i16>
; VEC512-NEXT:  Cost Model: Found an estimated cost of 0 for instruction: ret void
;
 %A = zext <8 x i16>
 undef to <8 x i64>
 %B = zext <8 x i32> undef to <8 x i64>
 %C = zext <16 x i8> undef to <16 x i32>
 %D = zext <16 x i16> undef to <16 x i32>
 %E = zext <32 x i8> undef to <32 x i16>
 ret void
}

define void @zext512() "min-legal-vector-width"="512" {
; AVX-LABEL: 'zext512'
; AVX-NEXT:  Cost Model: Found an estimated cost of 7 for instruction: %A = zext <8 x i16> undef to <8 x i64>
; AVX-NEXT:  Cost Model: Found an estimated cost of 3 for instruction: %B = zext <8 x i32> undef to <8 x i64>
; AVX-NEXT:  Cost Model: Found an estimated cost of 4 for instruction: %C = zext <16 x i8> undef to <16 x i32>
; AVX-NEXT:  Cost Model: Found an estimated cost of 4 for instruction: %D = zext <16 x i16> undef to <16 x
i32>
; AVX-NEXT:  Cost Model: Found an estimated cost of 3 for instruction: %E = zext <32 x i8> undef to <32 x i16>
; AVX-NEXT:  Cost Model: Found an estimated cost of 0 for instruction: ret void
;
```

```
; SKX256-LABEL: 'zext512'
; SKX256-NEXT:  Cost Model: Found an estimated cost of
 1 for instruction: %A = zext <8 x i16> undef to <8 x i64>
; SKX256-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %B = zext <8 x i32> undef to <8 x
i64>
; SKX256-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %C = zext <16 x i8> undef to <16 x
i32>
; SKX256-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %D = zext <16 x i16> undef to <16 x
i32>
; SKX256-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %E = zext <32 x i8> undef to <32 x
i16>
; SKX256-NEXT:  Cost Model: Found an estimated cost of 0 for instruction: ret void
;
; VEC512-LABEL: 'zext512'
; VEC512-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %A = zext <8 x i16> undef to <8 x
i64>
; VEC512-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %B = zext <8 x i32> undef to <8 x
i64>
; VEC512-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %C = zext <16 x i8> undef to <16 x
i32>
; VEC512-NEXT:  Cost Model: Found
 an estimated cost of 1 for instruction: %D = zext <16 x i16> undef to <16 x i32>
; VEC512-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %E = zext <32 x i8> undef to <32 x
i16>
; VEC512-NEXT:  Cost Model: Found an estimated cost of 0 for instruction: ret void
;
 %A = zext <8 x i16> undef to <8 x i64>
 %B = zext <8 x i32> undef to <8 x i64>
 %C = zext <16 x i8> undef to <16 x i32>
 %D = zext <16 x i16> undef to <16 x i32>
 %E = zext <32 x i8> undef to <32 x i16>
 ret void
}

define void @sext256() "min-legal-vector-width"="256" {
; VEC256-LABEL: 'sext256'
; VEC256-NEXT:  Cost Model: Found an estimated cost of 7 for instruction: %A = sext <8 x i8> undef to <8 x i64>
; VEC256-NEXT:  Cost Model: Found an estimated cost of 7 for instruction: %B = sext <8 x i16> undef to <8 x
i64>
; VEC256-NEXT:  Cost Model: Found an estimated cost of 3 for instruction: %C = sext <8 x i32> undef to <8 x
i64>
; VEC256-NEXT:  Cost Model: Found an estimated cost of 4 for instruction: %D
 = sext <16 x i8> undef to <16 x i32>
; VEC256-NEXT:  Cost Model: Found an estimated cost of 4 for instruction: %E = sext <16 x i16> undef to <16 x
i32>
; VEC256-NEXT:  Cost Model: Found an estimated cost of 3 for instruction: %F = sext <32 x i8> undef to <32 x
i16>
; VEC256-NEXT:  Cost Model: Found an estimated cost of 0 for instruction: ret void
```

```
;
; VEC512-LABEL: 'sext256'
; VEC512-NEXT:  Cost Model: Found an estimated cost of 24 for instruction: %A = sext <8 x i8> undef to <8 x
i64>
; VEC512-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %B = sext <8 x i16> undef to <8 x
i64>
; VEC512-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %C = sext <8 x i32> undef to <8 x
i64>
; VEC512-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %D = sext <16 x i8> undef to <16 x
i32>
; VEC512-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %E = sext <16 x i16> undef to <16 x
i32>
; VEC512-NEXT:  Cost Model: Found an estimated cost of
 1 for instruction: %F = sext <32 x i8> undef to <32 x i16>
; VEC512-NEXT:  Cost Model: Found an estimated cost of 0 for instruction: ret void
;
 %A = sext <8 x i8> undef to <8 x i64>
 %B = sext <8 x i16> undef to <8 x i64>
 %C = sext <8 x i32> undef to <8 x i64>
 %D = sext <16 x i8> undef to <16 x i32>
 %E = sext <16 x i16> undef to <16 x i32>
 %F = sext <32 x i8> undef to <32 x i16>
 ret void
}

define void @sext512() "min-legal-vector-width"="512" {
; AVX-LABEL: 'sext512'
; AVX-NEXT:  Cost Model: Found an estimated cost of 7 for instruction: %A = sext <8 x i8> undef to <8 x i64>
; AVX-NEXT:  Cost Model: Found an estimated cost of 7 for instruction: %B = sext <8 x i16> undef to <8 x i64>
; AVX-NEXT:  Cost Model: Found an estimated cost of 3 for instruction: %C = sext <8 x i32> undef to <8 x i64>
; AVX-NEXT:  Cost Model: Found an estimated cost of 4 for instruction: %D = sext <16 x i8> undef to <16 x i32>
; AVX-NEXT:  Cost Model: Found an estimated cost of 4 for instruction: %E
 = sext <16 x i16> undef to <16 x i32>
; AVX-NEXT:  Cost Model: Found an estimated cost of 3 for instruction: %F = sext <32 x i8> undef to <32 x i16>
; AVX-NEXT:  Cost Model: Found an estimated cost of 0 for instruction: ret void
;
; SKX256-LABEL: 'sext512'
; SKX256-NEXT:  Cost Model: Found an estimated cost of 24 for instruction: %A = sext <8 x i8> undef to <8 x
i64>
; SKX256-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %B = sext <8 x i16> undef to <8 x
i64>
; SKX256-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %C = sext <8 x i32> undef to <8 x
i64>
; SKX256-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %D = sext <16 x i8> undef to <16 x
i32>
; SKX256-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %E = sext <16 x i16> undef to <16 x
i32>
; SKX256-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %F = sext <32 x i8> undef to <32 x
```

i16>
; SKX256-NEXT: Cost Model: Found an estimated cost of 0 for
 instruction: ret void
;
; VEC512-LABEL: 'sext512'
; VEC512-NEXT: Cost Model: Found an estimated cost of 24 for instruction: %A = sext <8 x i8> undef to <8 x
i64>
; VEC512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %B = sext <8 x i16> undef to <8 x
i64>
; VEC512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %C = sext <8 x i32> undef to <8 x
i64>
; VEC512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %D = sext <16 x i8> undef to <16 x
i32>
; VEC512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %E = sext <16 x i16> undef to <16 x
i32>
; VEC512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %F = sext <32 x i8> undef to <32 x
i16>
; VEC512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
 %A = sext <8 x i8> undef to <8 x i64>
 %B = sext <8 x i16> undef to <8 x i64>
 %C = sext <8 x i32> undef to <8 x i64>
 %D = sext <16 x i8> undef to <16 x i32>
 %E = sext <16 x i16>
 undef to <16 x i32>
 %F = sext <32 x i8> undef to <32 x i16>
 ret void
}

$OpenBSD: COPYRIGHT,v 1.3 2003/06/02 20:18:36 millert Exp $

=-=

```
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*
* @(#)COPYRIGHT 8.1 (Berkeley) 3/16/94
*/
; RUN: llc -march=hexagon < %s
; REQUIRES: asserts

; The two loads based on %struct.0, loading two different data types
; cause LSR to assume type "void" for the memory type. This would then
; cause an assert in isLegalAddressingMode. Make sure we no longer crash.

target triple = "hexagon"

%struct.0 = type { i8*, i8, %union.anon.0 }
%union.anon.0 = type { i8* }

define hidden fastcc void @fred() unnamed_addr #0 {
entry:
 br i1 undef, label %while.end, label %while.body.lr.ph

while.body.lr.ph:                                 ; preds = %entry
 br label %while.body

while.body:                                       ; preds = %exit.2, %while.body.lr.ph
 %lsr.iv = phi %struct.0* [ %cgep22, %exit.2 ], [ undef, %while.body.lr.ph ]
 switch i32 undef, label %exit [
   i32 1, label %sw.bb.i
   i32 2, label %sw.bb3.i
 ]

sw.bb.i:                                          ; preds = %while.body
 unreachable

sw.bb3.i:                                         ; preds = %while.body
 unreachable

exit:
                                                  ; preds = %while.body
 switch i32 undef, label %exit.2 [
   i32 1, label %sw.bb.i17
   i32 2, label %sw.bb3.i20
 ]

sw.bb.i17:                                        ; preds = %.exit
```

```
 %0 = bitcast %struct.0* %lsr.iv to i32*
 %1 = load i32, i32* %0, align 4
 unreachable

sw.bb3.i20:                                       ; preds = %exit
 %2 = bitcast %struct.0* %lsr.iv to i8**
 %3 = load i8*, i8** %2, align 4
 unreachable

exit.2:                                           ; preds = %exit
 %cgep22 = getelementptr %struct.0, %struct.0* %lsr.iv, i32 1
 br label %while.body

while.end:                                        ; preds = %entry
 ret void
}

attributes #0 = { nounwind optsize "target-cpu"="hexagonv55" }
; RUN: opt %s -inline -S | FileCheck %s

define internal void @innerSmall() "min-legal-vector-width"="128" {
 ret void
}

define internal void @innerLarge() "min-legal-vector-width"="512" {
 ret void
}

define internal void @innerNoAttribute() {
 ret void
}

; We should not add an attribute during inlining. No attribute means unknown.
; Inlining doesn't change the fact that we don't know anything about this
; function.
define void @outerNoAttribute() {
 call void @innerLarge()
 ret void
}

define void @outerConflictingAttributeSmall() "min-legal-vector-width"="128" {
 call void @innerLarge()
 ret void
}

define void @outerConflictingAttributeLarge() "min-legal-vector-width"="512" {
 call void @innerSmall()
 ret void
```

```
}

; We should remove the attribute after inlining since the callee's
; vector width requirements are unknown.
define void @outerAttribute() "min-legal-vector-width"="128" {
 call void @innerNoAttribute()
 ret void
}

; CHECK: define void @outerNoAttribute() {
; CHECK:
 define void @outerConflictingAttributeSmall() #0
; CHECK: define void @outerConflictingAttributeLarge() #0
; CHECK: define void @outerAttribute() {
; CHECK: attributes #0 = { "min-legal-vector-width"="512" }
; RUN: llc -mtriple=aarch64-apple-ios %s -o - | FileCheck %s

define <16 x double> @test_sitofp_fixed(<16 x i32> %in) {
; CHECK-LABEL: test_sitofp_fixed:

 ; First, extend each i32 to i64
; CHECK-DAG: sshll2.2d [[BLOCK0_HI:v[0-9]+]], v0, #0
; CHECK-DAG: sshll2.2d [[BLOCK1_HI:v[0-9]+]], v1, #0
; CHECK-DAG: sshll2.2d [[BLOCK2_HI:v[0-9]+]], v2, #0
; CHECK-DAG: sshll2.2d [[BLOCK3_HI:v[0-9]+]], v3, #0
; CHECK-DAG: sshll.2d [[BLOCK0_LO:v[0-9]+]], v0, #0
; CHECK-DAG: sshll.2d [[BLOCK1_LO:v[0-9]+]], v1, #0
; CHECK-DAG: sshll.2d [[BLOCK2_LO:v[0-9]+]], v2, #0
; CHECK-DAG: sshll.2d [[BLOCK3_LO:v[0-9]+]], v3, #0

 ; Next, convert each to double.
; CHECK-DAG: scvtf.2d v0, [[BLOCK0_LO]]
; CHECK-DAG: scvtf.2d v1, [[BLOCK0_HI]]
; CHECK-DAG: scvtf.2d v2, [[BLOCK1_LO]]
; CHECK-DAG: scvtf.2d v3, [[BLOCK1_HI]]
; CHECK-DAG: scvtf.2d v4, [[BLOCK2_LO]]
; CHECK-DAG: scvtf.2d v5, [[BLOCK2_HI]]
; CHECK-DAG: scvtf.2d v6, [[BLOCK3_LO]]
; CHECK-DAG: scvtf.2d v7, [[BLOCK3_HI]]

; CHECK: ret
 %flt = sitofp <16 x i32>
 %in to <16 x double>
 %res = fdiv <16 x double> %flt, <double 64.0, double 64.0, double 64.0, double 64.0, double 64.0, double 64.0,
double 64.0, double 64.0, double 64.0, double 64.0, double 64.0, double 64.0, double 64.0, double 64.0, double 64.0,
double 64.0>
 ret <16 x double> %res
}
```

; This one is small enough to satisfy isSimple, but still illegally large.
define <4 x double> @test_sitofp_fixed_shortish(<4 x i64> %in) {
; CHECK-LABEL: test_sitofp_fixed_shortish:

; CHECK-DAG: scvtf.2d v0, v0
; CHECK-DAG: scvtf.2d v1, v1

; CHECK: ret
 %flt = sitofp <4 x i64> %in to <4 x double>
 %res = fdiv <4 x double> %flt, <double 64.0, double 64.0, double 64.0, double 64.0>
 ret <4 x double> %res
}
# People who have agreed to one of the CLAs and can contribute patches.
# The AUTHORS file lists the copyright holders; this file
# lists people. For example, Google employees are listed here
# but not in AUTHORS, because Google holds the copyright.
#
Names should be added to this file only after verifying that
the individual or the individual's organization has agreed to
the appropriate Contributor License Agreement, found here:
#
https://developers.google.com/open-source/cla/individual
https://developers.google.com/open-source/cla/corporate
#
The agreement for individuals can be filled out on the web.
#
When adding J Random Contributor's name to this file,
either J's name or J's organization's name should be
added to the AUTHORS file, depending on whether the
individual or corporate CLA was used.
#
Names should be added to this file as:
Name <email address>
#
Please keep the list sorted.

Albert Pretorius <pretoalb@gmail.com>
Arne Beer <arne@twobeer.de>
Billy
 Robert O'Neal III <billy.oneal@gmail.com> <bion@microsoft.com>
Chris Kennelly <ckennelly@google.com> <ckennelly@ckennelly.com>
Christopher Seymour <chris.j.seymour@hotmail.com>
David Coeurjolly <david.coeurjolly@liris.cnrs.fr>
Deniz Evrenci <denizevrenci@gmail.com>
Dominic Hamon <dma@stripysock.com> <dominic@google.com>
Dominik Czarnota <dominik.b.czarnota@gmail.com>
Eric Fiselier <eric@efcs.ca>

Eugene Zhuk <eugene.zhuk@gmail.com>
Evgeny Safronov <division494@gmail.com>
Felix Homann <linuxaudio@showlabor.de>
Ismael Jimenez Martinez <ismael.jimenez.martinez@gmail.com>
Jern-Kuan Leong <jernkuan@gmail.com>
JianXiong Zhou <zhoujianxiong2@gmail.com>
Joao Paulo Magalhaes <joaoppmagalhaes@gmail.com>
John Millikin <jmillikin@stripe.com>
Jussi Knuuttila <jussi.knuuttila@gmail.com>
Kai Wolf <kai.wolf@gmail.com>
Kishan Kumar <kumar.kishan@outlook.com>
Kaito Udagawa <umireon@gmail.com>
Lei Xu <eddyxu@gmail.com>
Matt Clarkson <mattyclarkson@gmail.com>
Maxim Vafin <maxvafin@gmail.com>
Nick Hutchinson
<nshutchinson@gmail.com>
Oleksandr Sochka <sasha.sochka@gmail.com>
Pascal Leroy <phl@google.com>
Paul Redmond <paul.redmond@gmail.com>
Pierre Phaneuf <pphaneuf@google.com>
Radoslav Yovchev <radoslav.tm@gmail.com>
Raul Marin <rmrodriguez@cartodb.com>
Ray Glover <ray.glover@uk.ibm.com>
Robert Guo <robert.guo@mongodb.com>
Roman Lebedev <lebedev.ri@gmail.com>
Shuo Chen <chenshuo@chenshuo.com>
Tobias Ulvgrd <tobias.ulvgard@dirac.se>
Tom Madams <tom.ej.madams@gmail.com> <tmadams@google.com>
Yixuan Qiu <yixuanq@gmail.com>
Yusuke Suzuki <utatane.tea@gmail.com>
Zbigniew Skowron <zbychs@gmail.com>
; RUN: opt -consthoist -S -o - %s | FileCheck %s
target datalayout = "e-m:e-p:32:32-i64:64-v128:64:128-a:0:32-n32-S64"
target triple = "thumbv6m-none--musleabi"

; Check that for i8 type, the maximum legal offset is 31.
; Also check that an constant used as value to be stored rather than
; pointer in a store instruction is hoisted.
; CHECK: foo_i8
; CHECK-DAG: %[[C1:const[0-9]?]] = bitcast i32 805874720 to i32
; CHECK-DAG: %[[C2:const[0-9]?]] = bitcast i32 805874688 to i32
; CHECK-DAG: %[[C3:const[0-9]?]] = bitcast i32 805873720 to i32
; CHECK-DAG: %[[C4:const[0-9]?]] = bitcast i32 805873688 to i32
; CHECK: %0 = inttoptr i32 %[[C2]] to i8*
; CHECK-NEXT: %1 = load volatile i8, i8* %0
; CHECK-NEXT: %[[M1:const_mat[0-9]?]] = add i32 %[[C2]], 4
; CHECK-NEXT: %2 = inttoptr i32 %[[M1]] to i8*

```
; CHECK-NEXT:  %3 = load volatile i8, i8* %2
; CHECK-NEXT:  %[[M2:const_mat[0-9]?]] = add i32 %[[C2]], 31
; CHECK-NEXT:  %4 = inttoptr i32 %[[M2]] to i8*
; CHECK-NEXT:  %5 = load volatile i8, i8*
 %4
; CHECK-NEXT:  %6 = inttoptr i32 %[[C1]] to i8*
; CHECK-NEXT:  %7 = load volatile i8, i8* %6
; CHECK-NEXT:  %[[M3:const_mat[0-9]?]] = add i32 %[[C1]], 7
; CHECK-NEXT:  %8 = inttoptr i32 %[[M3]] to i8*
; CHECK-NEXT:  %9 = load volatile i8, i8* %8
; CHECK-NEXT:  %10 = inttoptr i32 %[[C4]] to i8*
; CHECK-NEXT:  store i8 %9, i8* %10
; CHECK-NEXT:  %[[M4:const_mat[0-9]?]] = add i32 %[[C4]], 31
; CHECK-NEXT:  %11 = inttoptr i32 %[[M4]] to i8*
; CHECK-NEXT:  store i8 %7, i8* %11
; CHECK-NEXT:  %12 = inttoptr i32 %[[C3]] to i8*
; CHECK-NEXT:  store i8 %5, i8* %12
; CHECK-NEXT:  %[[M5:const_mat[0-9]?]] = add i32 %[[C3]], 7
; CHECK-NEXT:  %13 = inttoptr i32 %[[M5]] to i8*
; CHECK-NEXT:  store i8 %3, i8* %13
; CHECK-NEXT:  %[[M6:const_mat[0-9]?]] = add i32 %[[C1]], 80
; CHECK-NEXT:  %14 = inttoptr i32 %[[M6]] to i8*
; CHECK-NEXT:  store i8* %14, i8** @goo

@goo = global i8* undef

define void @foo_i8() {
entry:
 %0 = load volatile i8, i8* inttoptr (i32 805874688 to i8*)
 %1 = load volatile
 i8, i8* inttoptr (i32 805874692 to i8*)
 %2 = load volatile i8, i8* inttoptr (i32 805874719 to i8*)
 %3 = load volatile i8, i8* inttoptr (i32 805874720 to i8*)
 %4 = load volatile i8, i8* inttoptr (i32 805874727 to i8*)
 store i8 %4, i8* inttoptr(i32 805873688 to i8*)
 store i8 %3, i8* inttoptr(i32 805873719 to i8*)
 store i8 %2, i8* inttoptr(i32 805873720 to i8*)
 store i8 %1, i8* inttoptr(i32 805873727 to i8*)
 store i8* inttoptr(i32 805874800 to i8*), i8** @goo
 ret void
}

; Check that for i16 type, the maximum legal offset is 62.
; CHECK: foo_i16
; CHECK-DAG: %[[C1:const[0-9]?]] = bitcast i32 805874752 to i32
; CHECK-DAG: %[[C2:const[0-9]?]] = bitcast i32 805874688 to i32
; CHECK: %0 = inttoptr i32 %[[C2]] to i16*
; CHECK-NEXT: %1 = load volatile i16, i16* %0, align 2
```

```
; CHECK-NEXT: %[[M1:const_mat[0-9]?]] = add i32 %[[C2]], 4
; CHECK-NEXT: %2 = inttoptr i32 %[[M1]] to i16*
; CHECK-NEXT: %3 = load volatile i16, i16* %2, align 2
; CHECK-NEXT: %[[M2:const_mat[0-9]?]] = add i32
 %[[C2]], 32
; CHECK-NEXT: %4 = inttoptr i32 %[[M2]] to i16*
; CHECK-NEXT: %5 = load volatile i16, i16* %4, align 2
; CHECK-NEXT: %[[M3:const_mat[0-9]?]] = add i32 %[[C2]], 62
; CHECK-NEXT: %6 = inttoptr i32 %[[M3]] to i16*
; CHECK-NEXT: %7 = load volatile i16, i16* %6, align 2
; CHECK-NEXT: %8 = inttoptr i32 %[[C1]] to i16*
; CHECK-NEXT: %9 = load volatile i16, i16* %8, align 2
; CHECK-NEXT: %[[M4:const_mat[0-9]?]] = add i32 %[[C1]], 22
; CHECK-NEXT: %10 = inttoptr i32 %[[M4]] to i16*
; CHECK-NEXT: %11 = load volatile i16, i16* %10, align 2

define void @foo_i16() {
entry:
 %0 = load volatile i16, i16* inttoptr (i32 805874688 to i16*), align 2
 %1 = load volatile i16, i16* inttoptr (i32 805874692 to i16*), align 2
 %2 = load volatile i16, i16* inttoptr (i32 805874720 to i16*), align 2
 %3 = load volatile i16, i16* inttoptr (i32 805874750 to i16*), align 2
 %4 = load volatile i16, i16* inttoptr (i32 805874752 to i16*), align 2
 %5 = load volatile i16, i16* inttoptr (i32 805874774
 to i16*), align 2
 ret void
}

; Check that for i32 type, the maximum legal offset is 124.
; CHECK: foo_i32
; CHECK-DAG:  %[[C1:const[0-9]?]] = bitcast i32 805874816 to i32
; CHECK-DAG:  %[[C2:const[0-9]?]] = bitcast i32 805874688 to i32
; CHECK:  %0 = inttoptr i32 %[[C2]] to i32*
; CHECK-NEXT:  %1 = load volatile i32, i32* %0, align 4
; CHECK-NEXT:  %[[M1:const_mat[0-9]?]] = add i32 %[[C2]], 4
; CHECK-NEXT:  %2 = inttoptr i32 %[[M1]] to i32*
; CHECK-NEXT:  %3 = load volatile i32, i32* %2, align 4
; CHECK-NEXT:  %[[M2:const_mat[0-9]?]] = add i32 %[[C2]], 124
; CHECK-NEXT:  %4 = inttoptr i32 %[[M2]] to i32*
; CHECK-NEXT:  %5 = load volatile i32, i32* %4, align 4
; CHECK-NEXT:  %6 = inttoptr i32 %[[C1]] to i32*
; CHECK-NEXT:  %7 = load volatile i32, i32* %6, align 4
; CHECK-NEXT:  %[[M3:const_mat[0-9]?]] = add i32 %[[C1]], 8
; CHECK-NEXT:  %8 = inttoptr i32 %[[M3]] to i32*
; CHECK-NEXT:  %9 = load volatile i32, i32* %8, align 4
; CHECK-NEXT:  %[[M4:const_mat[0-9]?]] = add i32 %[[C1]],
 12
; CHECK-NEXT:  %10 = inttoptr i32 %[[M4]] to i32*
```

; CHECK-NEXT: %11 = load volatile i32, i32* %10, align 4

define void @foo_i32() {
entry:
 %0 = load volatile i32, i32* inttoptr (i32 805874688 to i32*), align 4
 %1 = load volatile i32, i32* inttoptr (i32 805874692 to i32*), align 4
 %2 = load volatile i32, i32* inttoptr (i32 805874812 to i32*), align 4
 %3 = load volatile i32, i32* inttoptr (i32 805874816 to i32*), align 4
 %4 = load volatile i32, i32* inttoptr (i32 805874824 to i32*), align 4
 %5 = load volatile i32, i32* inttoptr (i32 805874828 to i32*), align 4
 ret void
}

All rights reserved.

Developed by:

LLVM Team

University of Illinois at Urbana-Champaign

http://llvm.org

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal with the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimers.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimers in the documentation and/or other materials provided with the distribution.

* Neither the names of the LLVM Team, University of Illinois at Urbana-Champaign, nor the names of its contributors may be used to endorse or promote products derived from this Software without specific prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

==

Copyrights and Licenses for Third Party Software Distributed with LLVM:

==

The LLVM software contains code written by third parties. Such software will have its own individual LICENSE.TXT file in the directory in which it appears. This file will describe the copyrights, license, and restrictions which apply to that code.

The disclaimer of warranty in the University of Illinois Open Source License

applies to all code in the LLVM Distribution, and nothing in any of the
other licenses gives permission to use the names of the LLVM Team or the
University of Illinois to endorse or promote products derived from this
Software.

The following pieces of software have additional or alternate copyrights,
licenses, and/or restrictions:

Program	Directory
Google Test	llvm/utils/unittest/googletest
OpenBSD regex	llvm/lib/Support/{reg*, COPYRIGHT.regex}
pyyaml tests	llvm/test/YAMLParser/{*.data, LICENSE.TXT}
ARM contributions	llvm/lib/Target/ARM/LICENSE.TXT
md5 contributions	llvm/lib/Support/MD5.cpp llvm/include/llvm/Support/MD5.h

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical

transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made,

use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,

or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

LLVM System Interface Library

The LLVM System Interface Library is licensed under the Illinois Open Source
License and has the following additional copyright:

Copyright (C) 2004 eXtensible Systems, Inc.

```
; RUN: llc < %s -mtriple=x86_64-linux-android -mattr=+mmx -enable-legalize-types-checking | FileCheck %s
;
; D31946
; Check that we dont end up with the ""LLVM ERROR: Cannot select" error.
; Additionally ensure that the output code actually put fp128 values in SSE registers.

declare fp128 @llvm.fabs.f128(fp128)
declare fp128 @llvm.copysign.f128(fp128, fp128)
```

```
define fp128 @TestSelect(fp128 %a, fp128 %b) {
 %cmp = fcmp ogt fp128 %a, %b
 %sub = fsub fp128 %a, %b
 %res = select i1 %cmp, fp128 %sub, fp128 0xL00000000000000000000000000000000
 ret fp128 %res
; CHECK-LABEL: TestSelect:
; CHECK        movaps 16(%rsp), %xmm1
; CHECK-NEXT   callq __subtf3
; CHECK-NEXT   testl %ebx, %ebx
; CHECK-NEXT   jg .LBB0_2
; CHECK-NEXT # %bb.1:
; CHECK-NEXT   movaps .LCPI0_0(%rip), %xmm0
; CHECK-NEXT .LBB0_2:
; CHECK-NEXT   addq $32, %rsp
; CHECK-NEXT   popq %rbx
; CHECK-NEXT   retq
}

define fp128 @TestFabs(fp128 %a) {
 %res = call fp128 @llvm.fabs.f128(fp128 %a)
 ret fp128 %res
; CHECK-LABEL: TestFabs:
;
 CHECK      andps .LCPI1_0(%rip), %xmm0
; CHECK-NEXT retq
}

define fp128 @TestCopysign(fp128 %a, fp128 %b) {
 %res = call fp128 @llvm.copysign.f128(fp128 %a, fp128 %b)
 ret fp128 %res
; CHECK-LABEL: TestCopysign:
; CHECK      andps .LCPI2_1(%rip), %xmm0
; CHECK-NEXT orps %xmm1, %xmm0
; CHECK-NEXT retq
}

define fp128 @TestFneg(fp128 %a) {
 %mul = fmul fp128 %a, %a
 %res = fsub fp128 0xL00000000000000008000000000000000, %mul
 ret fp128 %res
; CHECK-LABEL: TestFneg:
; CHECK      movaps %xmm0, %xmm1
; CHECK-NEXT callq __multf3
; CHECK-NEXT xorps .LCPI3_0(%rip), %xmm0
; CHECK-NEXT popq %rax
; CHECK-NEXT retq
}
```

1.11 json-c 1.2.11

1.11.1 Available under license :

Copyright (c) 2009-2012 Eric Haszlakiewicz

Permission is hereby granted, free of charge, to any person obtaining a
copy of this software and associated documentation files (the "Software"),
to deal in the Software without restriction, including without limitation
the rights to use, copy, modify, merge, publish, distribute, sublicense,
and/or sell copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included
in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

Copyright (c) 2004, 2005 Metaparadigm Pte Ltd

Permission is hereby granted, free of charge, to any person obtaining a
copy of this software and associated documentation files (the "Software"),
to deal in the Software without restriction, including without limitation
the rights to use, copy, modify, merge, publish, distribute, sublicense,
and/or sell copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included
in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE
FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE

SOFTWARE.

1.12 nghttp2 1.40.0

1.12.1 Available under license :

The MIT License

Copyright (c) 2012, 2014, 2015, 2016 Tatsuhiro Tsujikawa
Copyright (c) 2012, 2014, 2015, 2016 nghttp2 contributors

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
 WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
See COPYING
Copyright Joyent, Inc. and other Node contributors.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to
deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or
sell copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING

FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS
IN THE SOFTWARE.
This software is licensed under the MIT License.

Copyright Fedor Indutny, 2018.

Permission is hereby granted, free of charge, to any person obtaining a
copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, subject to the
following conditions:

The above copyright notice and this permission notice shall be included
in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN
NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM,
DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR
OTHERWISE,
ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE
USE OR OTHER DEALINGS IN THE SOFTWARE.
If not otherwise noted, the extensions in this package are licensed
under the following license.

Copyright (c) 2010 by the contributors (see AUTHORS file).
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,

SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.13 json-cpp 1.9.3

1.13.1 Available under license :

The JsonCpp library's source code, including accompanying documentation,
tests and demonstration applications, are licensed under the following
conditions...

Baptiste Lepilleur and The JsonCpp Authors explicitly disclaim copyright in all
jurisdictions which recognize such a disclaimer. In such jurisdictions,
this software is released into the Public Domain.

In jurisdictions which do not recognize Public Domain property (e.g. Germany as of
2010), this software is Copyright (c) 2007-2010 by Baptiste Lepilleur and
The JsonCpp Authors, and is released under the terms of the MIT License (see below).

In jurisdictions which recognize Public Domain property, the user of this
software may choose to accept it either as 1) Public Domain, 2) under the
conditions of the MIT License (see below), or 3) under the terms of dual
Public Domain/MIT License conditions described here, as they choose.

The MIT License is about as close to Public Domain as a license can get, and is
described in clear,
concise terms at:

http://en.wikipedia.org/wiki/MIT_License

The full text of the MIT License follows:

==========

Copyright (c) 2007-2010 Baptiste Lepilleur and The JsonCpp Authors

Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without
restriction, including without limitation the rights to use, copy,
modify, merge, publish, distribute, sublicense, and/or sell copies
of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS
BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN
ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
==========
(END LICENSE TEXT)

The MIT license is compatible with both the GPL and commercial
software, affording one all of the rights of Public Domain with the
minor nuisance of being required to keep the above copyright notice
and license text in the source code. Note also that by accepting the
Public Domain "license" you can re-license your copy using whatever
license you like.

1.14 sqlite 3.33.0

1.14.1 Available under license :

The author disclaims copyright to this source code. In place of
a legal notice, here is a blessing:

May you do good and not evil.
May you find forgiveness for yourself and forgive others.
May you share freely, never taking more than you give.

1.15 libevent 2.1.7-beta

1.15.1 Available under license :

Libevent is available for use under the following license, commonly known
as the 3-clause (or "modified") BSD license:

==========
Copyright (c) 2000-2007 Niels Provos <provos@citi.umich.edu>
Copyright (c) 2007-2012 Niels Provos and Nick Mathewson

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright

notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING,
BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
==========

Portions of Libevent are based on works by others, also made available by
them under the three-clause BSD license above. The copyright notices are
available in the corresponding source files; the license is as above. Here's
a list:

log.c:
Copyright (c) 2000 Dug Song <dugsong@monkey.org>
Copyright (c) 1993 The Regents of the University of California.

strlcpy.c:
Copyright (c) 1998 Todd C. Miller <Todd.Miller@courtesan.com>

win32select.c:
Copyright (c) 2003 Michael A. Davis <mike@datanerds.net>

evport.c:
Copyright (c) 2007 Sun Microsystems

ht-internal.h:
Copyright (c) 2002 Christopher Clark

minheap-internal.h:
Copyright (c) 2006 Maxim Yegorushkin <maxim.yegorushkin@gmail.com>

==========

The arc4module is available under the following, sometimes called the
"OpenBSD" license:

Copyright (c) 1996, David Mazieres <dm@uun.org>
Copyright (c) 2008, Damien Miller <djm@openbsd.org>

Permission to use, copy, modify, and distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR
BE LIABLE FOR
ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

==========

The Windows timer code is based on code from libutp, which is
distributed under this license, sometimes called the "MIT" license.

Copyright (c) 2010 BitTorrent, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be
included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
CMake - Cross Platform Makefile Generator
Copyright 2000-2013 Kitware, Inc.
Copyright 2000-2011 Insight Software Consortium
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

* Redistributions of source code must retain the above copyright
 notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright
 notice, this list of conditions and the following disclaimer in the
 documentation and/or other materials provided with the distribution.

* Neither the names of Kitware, Inc., the Insight Software Consortium,
 nor the names of their contributors may be used to endorse or promote
 products derived from this software without specific prior written
 permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO,
 THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The above copyright and license notice applies to distributions of
CMake in source and binary form. Some source files contain additional
notices of original copyright by their contributors; see each source
for details. Third-party software packages supplied with CMake under
compatible licenses provide their
 own copyright notices documented in
corresponding subdirectories.

CMake was initially developed by Kitware with the following sponsorship:

* National Library of Medicine at the National Institutes of Health
  as part of the Insight Segmentation and Registration Toolkit (ITK).

* US National Labs (Los Alamos, Livermore, Sandia) ASC Parallel
  Visualization Initiative.

* National Alliance for Medical Image Computing (NAMIC) is funded by the
  National Institutes of Health through the NIH Roadmap for Medical Research,
  Grant U54 EB005149.

* Kitware, Inc.
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the copyright
  notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the copyright
  notice, this list of conditions and the following disclaimer in the
  documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products
  derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT
 OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.16 expat 2.2.0

1.16.1 Available under license :

Copyright (c) 1998-2000 Thai Open Source Software Center Ltd and Clark Cooper
Copyright (c) 2001-2016 Expat maintainers

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be included
in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN
ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

1.17 openssl 1.1.1k

1.17.1 Notifications :

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit
(http://www.openssl.org/)
This product includes cryptographic software written by Eric Young (eay@cryptsoft.com).
This product includes software written by Tim Hudson (tjh@cryptsoft.com).

1.17.2 Available under license :

LICENSE ISSUES
==========

The OpenSSL toolkit stays under a double license, i.e. both the conditions of
the OpenSSL License and the original SSLeay license apply to the toolkit.
See below for the actual license texts.

OpenSSL License

/* ==
* Copyright (c) 1998-2019 The OpenSSL Project. All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
*
* 1. Redistributions of source code must retain the above copyright
*   notice, this list of conditions and the following disclaimer.
*
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in
*   the documentation and/or other materials provided with the
*   distribution.
*
* 3. All advertising materials mentioning features or use
of this
*   software must display the following acknowledgment:
*   "This product includes software developed by the OpenSSL Project

```
*    for use in the OpenSSL Toolkit. (http://www.openssl.org/)"
*
* 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
*    endorse or promote products derived from this software without
*    prior written permission. For written permission, please contact
*    openssl-core@openssl.org.
*
* 5. Products derived from this software may not be called "OpenSSL"
*    nor may "OpenSSL" appear in their names without prior written
*    permission of the OpenSSL Project.
*
* 6. Redistributions of any form whatsoever must retain the following
*    acknowledgment:
*    "This product includes software developed by the OpenSSL Project
*    for use in the OpenSSL Toolkit (http://www.openssl.org/)"
*
* THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY
* EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED
 TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
* PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL THE OpenSSL PROJECT OR
* ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
* NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
* LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
* STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
* OF THE POSSIBILITY OF SUCH DAMAGE.
* ====================================================================
*
* This product includes cryptographic software written by Eric Young
* (eay@cryptsoft.com).  This product includes software written by Tim
* Hudson (tjh@cryptsoft.com).
*
*/
```

Original SSLeay License

```
/* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
 * All rights reserved.
*
* This package is an SSL implementation written
* by Eric Young (eay@cryptsoft.com).
* The implementation was written so as to conform with Netscapes SSL.
*
* This library is free for commercial and non-commercial use as long as
```

* the following conditions are aheared to. The following conditions
* apply to all code found in this distribution, be it the RC4, RSA,
* lhash, DES, etc., code; not just the SSL code. The SSL documentation
* included with this distribution is covered by the same copyright terms
* except that the holder is Tim Hudson (tjh@cryptsoft.com).
*
* Copyright remains Eric Young's, and as such any Copyright notices in
* the code are not to be removed.
* If this package is used in a product, Eric Young should be given attribution
* as the author of the parts of the library used.
* This can be in the form of a textual message at program startup or
* in documentation (online or textual) provided with the package.
 *
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the copyright
*   notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
*   notice, this list of conditions and the following disclaimer in the
*   documentation and/or other materials provided with the distribution.
* 3. All advertising materials mentioning features or use of this software
*   must display the following acknowledgement:
*   "This product includes cryptographic software written by
*   Eric Young (eay@cryptsoft.com)"
*   The word 'cryptographic' can be left out if the rouines from the library
*   being used are not cryptographic related :-).
* 4. If you include any Windows specific code (or a derivative thereof) from
*   the apps directory (application code) you must include
 an acknowledgement:
*   "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
*
* THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*
* The licence and distribution terms for any publically available version or
* derivative of this code cannot be changed. i.e. this
 code cannot simply be
* copied and put under another distribution licence

* [including the GNU Public Licence.]
*/

1.18 mbed-tls 2.23.133

1.18.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct

or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.19 curl 7.77.0

1.19.1 Available under license :

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1996 - 2021, Daniel Stenberg, <daniel@haxx.se>, and many
contributors, see the THANKS file.

All rights reserved.

Permission to use, copy, modify, and distribute this software for any purpose
with or without fee is hereby granted, provided that the above copyright
notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE

OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not
be used in advertising or otherwise to promote the sale, use or other
 dealings
in this Software without prior written authorization of the copyright holder.

1.20 curl 7.78.0

1.20.1 Available under license :

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1996 - 2021, Daniel Stenberg, <daniel@haxx.se>, and many
contributors, see the THANKS file.

All rights reserved.

Permission to use, copy, modify, and distribute this software for any purpose
with or without fee is hereby granted, provided that the above copyright
notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN
NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM,
DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR
OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE
OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not
be used in advertising or otherwise to promote the sale, use or other
 dealings
in this Software without prior written authorization of the copyright holder.

1.21 libmspack 0.10.1alpha

1.21.1 Available under license :

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts

as the successor of the GNU Library Public License, version 2, hence
the version number 2.1.]

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
Licenses are intended to guarantee your freedom to share and change
free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some
specially designated software packages--typically libraries--of the
Free Software Foundation and other authors who decide to use it. You
can use it too, but we suggest
you first think carefully about whether
this license or the ordinary General Public License is the better
strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use,
not price. Our General Public Licenses are designed to make sure that
you have the freedom to distribute copies of free software (and charge
for this service if you wish); that you receive source code or can get
it if you want it; that you can change the software and use pieces of
it in new free programs; and that you are informed that you can do
these things.

To protect your rights, we need to make restrictions that forbid
distributors to deny you these rights or to ask you to surrender these
rights. These restrictions translate to certain responsibilities for
you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis
or for a fee, you must give the recipients
all the rights that we gave
you. You must make sure that they, too, receive or can get the source
code. If you link other code with the library, you must provide
complete object files to the recipients, so that they can relink them
with the library after making changes to the library and recompiling
it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the
library, and (2) we offer you this license, which gives you legal
permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that
there is no warranty for the free library. Also, if the library is
modified by someone else and passed on, the recipients should know
that what they have is not the original version, so that the original

author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU

operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without
limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does

and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves,

then this License, and its terms, do not apply to those
sections when you distribute them as separate works. But when you
distribute the same sections as part of a whole which is a work based
on the Library, the distribution of the whole must be on the terms of
this License, whose permissions for other licensees extend to the
entire whole, and thus to each and every part regardless of who wrote
it.

Thus, it is not the intent of this section to claim rights or contest
your rights to work written entirely by you; rather, the intent is to
exercise the right to control the distribution of derivative or
collective works based on the Library.

In addition, mere aggregation of another work not based on the Library
with the Library (or with a work based on the Library) on a volume of
a storage or distribution medium does not bring the other work under
the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public
License instead of this License to
a given copy of the Library. To do
this, you must alter all the notices that refer to this License, so
that they refer to the ordinary GNU General Public License, version 2,
instead of to this License. (If a newer version than version 2 of the
ordinary GNU General Public License has appeared, then you can specify
that version instead if you wish.) Do not make any other change in
these notices.

Once this change is made in a given copy, it is irreversible for
that copy, so the ordinary GNU General Public License applies to all
subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of
the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or
derivative of it, under Section 2) in object code or executable form
under the terms of Sections 1 and 2 above provided that you accompany
it with the complete corresponding machine-readable source code, which
must be distributed
under the terms of Sections 1 and 2 above on a
medium customarily used for software interchange.

If distribution of object code is made by offering access to copy
from a designated place, then offering equivalent access to copy the
source code from the same place satisfies the requirement to
distribute the source code, even though third parties are not
compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License.
Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot

use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues),

conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute
so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people
have made
generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License
from time to time.
Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and

conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of
all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL
DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is
safest to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least the
"copyright" line and
a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Lesser General Public
License as published by the Free Software Foundation; either
version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public
License along with this library; if not, write to the Free Software
Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

You should also
get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the library, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the
library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!

1.22 httpparser 2.9.4

1.22.1 Available under license :

Copyright Joyent, Inc. and other Node contributors.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to
deal in the Software without restriction, including without limitation the

rights to use, copy, modify, merge, publish, distribute, sublicense, and/or
sell copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS
IN THE SOFTWARE.

1.23 capnproto 0.7.0

1.23.1 Available under license :

Copyright (c) 2013-2017 Sandstorm Development Group, Inc.; Cloudflare, Inc.;
and other contributors. Each commit is copyright by its respective author or
author's employer.

Licensed under the MIT License:

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS
OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

1.24 clamav 0.103.2.19

1.24.1 Available under license :

Found license 'General Public License 2.0' in 'GNU Libltdl is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version. As a special exception to the GNU Lesser General Public License, GNU Libltdl is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. You should have received a copy of the GNU Lesser General Public License along with GNU Libltdl; see the file COPYING.LIB. If not, a copy can be downloaded from http://www.gnu.org/licenses/lgpl.html,'

Found license 'General Public License 2.0' in '* Copyright (C) 2020-2021 Cisco Systems, Inc. and/or its affiliates. All rights reserved. * This program is free software; you can redistribute it and/or modify * it under the terms of the GNU General Public License version 2 as * published by the Free Software Foundation. * This program is distributed in the hope that it will be useful, * but WITHOUT ANY WARRANTY; without even the implied warranty of * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the * GNU General Public License for more details. * You should have received a copy of the GNU General Public License * along with this program; if not, write to the Free Software * MA 02110-1301, USA.'

Found license 'General Public License 2.0' in '* Copyright (C) 2014 Cisco and/or its affiliates. All rights reserved. * This program is free software; you can redistribute it and/or modify * it under the terms of the GNU General Public License version 2 as * published by the Free Software Foundation. * This program is distributed in the hope that it will be useful, * but WITHOUT ANY WARRANTY; without even the implied warranty of * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the * GNU General Public License for more details. * You should have received a copy of the GNU General Public License * along with this program; if not, write to the Free Software * MA 02110-1301, USA. * In addition, as a special exception, the copyright holders give * You must obey the GNU General Public License in all respects'

Found license 'General Public License 2.0' in '* Copyright (C) 2014-2021 Cisco Systems, Inc. and/or its affiliates. All rights reserved. * This program is free software; you can redistribute it and/or modify it under * the terms of the GNU General Public License version 2 as published by the * This program is distributed in the hope that it will be useful, but WITHOUT * ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or * FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for * more details. * You should have received a copy of the GNU General Public License along with Copyright (c) 2007-2013. The YARA Authors. All Rights Reserved. Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0 distributed under the License is distributed on an "AS IS" BASIS,'

Found license 'General Public License 2.0' in '* libmspack is free software; you can redistribute it and/or modify it under * the terms of the GNU Lesser General Public License (LGPL) version 2.1 * For further details, see the file COPYING.LIB distributed with libmspack'

Found license 'General Public License 2.0' in '** Copyright (C) 2013-2021 Cisco Systems, Inc. and/or its affiliates. All rights reserved. ** This program is free software; you can redistribute it and/or modify ** it under the terms of the GNU General Public License Version 2 as ** published by the Free Software Foundation. You may not use, modify or ** distribute this program under any other version of the GNU General ** Public License. ** This program is distributed in the hope that it will be useful, ** but WITHOUT ANY WARRANTY; without even the implied warranty of ** MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the ** GNU General Public License for more details. ** You should have received a copy of the GNU General Public License ** along with this program; if not, write to the Free Software'

Found license 'General Public License 2.0' in '* Copyright (C) 2019-2021 Cisco Systems, Inc. and/or its affiliates. All rights reserved. * This program is free software; you can redistribute it and/or modify * it under the terms of the GNU General Public License version 2 as * published by the Free Software Foundation. * This program is distributed in the hope that it will be useful, * but WITHOUT ANY WARRANTY; without even the implied warranty of * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the * GNU General Public License for more details. * You should have received a copy of the GNU General Public License * along with this program; if not, write to the Free Software * MA 02110-1301, USA.'
Found license 'General Public License 2.0' in '* Copyright (C) 2013-2021 Cisco Systems, Inc. and/or its affiliates. All rights reserved. * This program is free software; you can redistribute it and/or modify * it under the terms of the GNU General Public License as published by * the Free Software Foundation; either version 2 of the License, or * (at your option) any later version. * This program is distributed in the hope that it will be useful, * but WITHOUT ANY WARRANTY; without even the implied warranty of * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the * GNU General Public License for more details. * You should have received a copy of the GNU General Public License * along with this program; if not, write to the Free Software * MA 02110-1301, USA.'
Found license 'GNU Lesser General Public License' in 'The GNU C Library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version. The GNU C Library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. You should have received a copy of the GNU Lesser General Public License along with the GNU C Library; if not, see <http://www.gnu.org/licenses/>. */ * The Regents of the University of California. All rights reserved. * Redistribution and use in source and binary forms, with or without * modification, are permitted provided that the following conditions * are met: * 1. Redistributions of source code must retain the above copyright *   notice,
 this list of conditions and the following disclaimer. * 2. Redistributions in binary form must reproduce the above copyright *   notice, this list of conditions and the following disclaimer in the *   documentation and/or other materials provided with the distribution. * 4. Neither the name of the University nor the names of its contributors * without specific prior written permission.'
Found license 'General Public License 2.0' in '* Copyright (C) 2016-2017 Cisco and/or its affiliates. All rights reserved. * This program is free software; you can redistribute it and/or modify * it under the terms of the GNU General Public License version 2 as * published by the Free Software Foundation. * This program is distributed in the hope that it will be useful, * but WITHOUT ANY WARRANTY; without even the implied warranty of * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the * GNU General Public License for more details. * You should have received a copy of the GNU General Public License * along with this program; if not, write to the Free Software * MA 02110-1301, USA. * In addition, as a special exception, the copyright holders give * You must obey the GNU General Public License in all respects'
Found license 'General Public License 2.0' in '* This program is free software; you can redistribute it and/or modify * it under the terms of the GNU General Public License version 2 as * published by the Free Software Foundation. * This program is distributed in the hope that it will be useful, * but WITHOUT ANY WARRANTY; without even the implied warranty of * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the * GNU General Public License for more details. * You should have received a copy of the GNU General Public License * along with this program; if not, write to the Free Software * MA 02110-1301, USA.'
Found license 'GNU Lesser General Public License' in '* libmspack is free software; you can redistribute it and/or modify it under * the terms of the GNU Lesser General Public License (LGPL) version 2.1 * This program is distributed in the hope that it will be useful, * but WITHOUT ANY WARRANTY; without even the implied warranty of * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the * GNU Lesser General Public License for more details. * You should have received a copy of the GNU Lesser General Public

License * along with this program; if not, write to the Free Software'
Found license 'GNU Lesser General Public License' in 'GNU Libltdl is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version. As a special exception to the GNU Lesser General Public License, GNU Libltdl is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. You should have received a copy of the GNU Lesser General Public License along with GNU Libltdl; see the file COPYING.LIB. If not, a copy can be downloaded from http://www.gnu.org/licenses/lgpl.html,'
Found license 'General Public License 2.0' in '* Copyright (C) 2013-2019 Cisco Systems, Inc. and/or its affiliates. All rights reserved. * This program is free software; you can redistribute it and/or modify * it under the terms of the GNU General Public License version 2 as * published by the Free Software Foundation. * This program is distributed in the hope that it will be useful, * but WITHOUT ANY WARRANTY; without even the implied warranty of * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the * GNU General Public License for more details. * You should have received a copy of the GNU General Public License * along with this program; if not, write to the Free Software * MA 02110-1301, USA.'
Found license 'GNU Lesser General Public License' in '* libmspack is free software; you can redistribute it and/or modify it under * the terms of the GNU Lesser General Public License (LGPL) version 2.1 * For further details, see the file COPYING.LIB distributed with libmspack'
Found license 'GNU Lesser General Public License' in 'This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version. As a special exception to the GNU Lesser General Public License, GNU Libltdl is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. You should have received a copy of the GNU Lesser General Public License along with GNU Libltdl; see the file COPYING.LIB. If not, a copy con be downloaded from http://www.gnu.org/licenses/lgpl.html,'
Found license 'General Public License 2.0' in 'This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>. */ /* As a special exception, you may create a larger work that contains part or all of the Bison parser skeleton and distribute that work special exception, which will cause the skeleton and the resulting Bison output files to be licensed under the GNU General Public License without this special exception. This special exception was added by the Free Software Foundation in'
Found license 'General Public License 2.0' in '* Copyright (C) 2014-2021 Cisco Systems, Inc. and/or its affiliates. All rights reserved. * This program is free software; you can redistribute it and/or modify it under * the terms of the GNU General Public License version 2 as published by the * This program is distributed in the hope that it will be useful, but WITHOUT * ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or * FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for * more details. * You should have received a copy of the GNU General Public License along with'
Found license 'General Public License 2.0' in '* Copyright (C) 2015-2021 Cisco Systems, Inc. and/or its affiliates. All rights reserved. * This program is free software; you can redistribute it and/or modify it under * the terms of the GNU General Public License version 2 as published by the * This program is distributed in the hope that it will be useful, but WITHOUT * ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or * FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for * more details. * You

should have received a copy of the GNU General Public License along with'
Found license 'General Public License 2.0' in '* Copyright (C) 2015 Cisco Systems, Inc. and/or its affiliates. All rights reserved. * This program is free software; you can redistribute it and/or modify * it under the terms of the GNU General Public License version 2 as * published by the Free Software Foundation. * This program is distributed in the hope that it will be useful, * but WITHOUT ANY WARRANTY; without even the implied warranty of * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the * GNU General Public License for more details. * You should have received a copy of the GNU General Public License * along with this program; if not, write to the Free Software * MA 02110-1301, USA.'
Found license 'General Public License 2.0' in '* Copyright (C) 2014-2021 Cisco Systems, Inc. and/or its affiliates. All rights reserved. * This program is free software; you can redistribute it and/or modify * it under the terms of the GNU General Public License version 2 as * published by the Free Software Foundation. * This program is distributed in the hope that it will be useful, * but WITHOUT ANY WARRANTY; without even the implied warranty of * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the * GNU General Public License for more details. * You should have received a copy of the GNU General Public License * along with this program; if not, write to the Free Software * MA 02110-1301, USA.'
Found license 'General Public License 2.0' in 'This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2, or (at your option) any later version. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation,'
Found license 'General Public License 2.0' in '* Copyright (C) 2013-2021 Cisco Systems, Inc. and/or its affiliates. All rights reserved. * This program is free software; you can redistribute it and/or modify * it under the terms of the GNU General Public License version 2 as * published by the Free Software Foundation. * This program is distributed in the hope that it will be useful, * but WITHOUT ANY WARRANTY; without even the implied warranty of * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the * GNU General Public License for more details. * You should have received a copy of the GNU General Public License * along with this program; if not, write to the Free Software * MA 02110-1301, USA.'
Found license 'General Public License 2.0' in '* Copyright (C) 2013-2021 Cisco Systems, Inc. and/or its affiliates. All rights reserved. * Acknowledgements: cli_strcasestr() contains a public domain code from: * This program is free software; you can redistribute it and/or modify * it under the terms of the GNU General Public License version 2 as * published by the Free Software Foundation. * This program is distributed in the hope that it will be useful, * but WITHOUT ANY WARRANTY; without even the implied warranty of * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the * GNU General Public License for more details. * You should have received a copy of the GNU General Public License * along with this program; if not, write to the Free Software * MA 02110-1301, USA.'
Found license 'General Public License 2.0' in '* Copyright (C) 2015-2018 Cisco Systems, Inc. and/or its affiliates. All rights reserved. * This program is free software; you can redistribute it and/or modify * it under the terms of the GNU General Public License version 2 as * published by the Free Software Foundation. * This program is distributed in the hope that it will be useful, * but WITHOUT ANY WARRANTY; without even the implied warranty of * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the * GNU General Public License for more details. * You should have received a copy of the GNU General Public License * along with this program; if not, write to the Free Software * MA 02110-1301, USA.'
Found license 'General Public License 2.0' in '* Copyright (C) 2016-2021 Cisco Systems, Inc. and/or its affiliates. All rights reserved. * This program is free software; you can redistribute it and/or modify * it under the terms of the GNU General Public License version 2 as * published by the Free Software Foundation. * This program is distributed in the hope that it will be useful, * but WITHOUT ANY WARRANTY; without even the implied warranty of * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the * GNU General

Public License for more details. * You should have received a copy of the GNU General Public License * along with this program; if not, write to the Free Software * MA 02110-1301, USA.'
Found license 'General Public License 2.0' in 'GNU Libltdl is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version. As a special exception to the GNU Lesser General Public License, GNU Libltdl is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. You should have received a copy of the GNU Lesser General Public License along with GNU Libltdl; see the file COPYING.LIB. If not, a copy can be downloaded from http://www.gnu.org/licenses/lgpl.html,'
Found license 'General Public License 2.0' in '* Copyright (C) 2018-2021 Cisco Systems, Inc. and/or its affiliates. All rights reserved. * This program is free software; you can redistribute it and/or modify * it under the terms of the GNU General Public License version 2 as * published by the Free Software Foundation. * This program is distributed in the hope that it will be useful, * but WITHOUT ANY WARRANTY; without even the implied warranty of * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the * GNU General Public License for more details. * You should have received a copy of the GNU General Public License * along with this program; if not, write to the Free Software * MA 02110-1301, USA.'
Found license 'General Public License 2.0' in '* Copyright (C) 2013-2021 Cisco Systems, Inc. and/or its affiliates. All rights reserved. * Acknowledgements: ClamAV untar code is based on a public domain minitar utility * This program is free software; you can redistribute it and/or modify * it under the terms of the GNU General Public License version 2 as * published by the Free Software Foundation. * This program is distributed in the hope that it will be useful, * but WITHOUT ANY WARRANTY; without even the implied warranty of * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the * GNU General Public License for more details. * You should have received a copy of the GNU General Public License * along with this program; if not, write to the Free Software * MA 02110-1301, USA.'
Found license 'General Public License 2.0' in '* Copyright (C) 2013-2021 Cisco Systems, Inc. and/or its affiliates. All rights reserved. * This program is free software; you can redistribute it and/or modify it under * the terms of the GNU General Public License version 2 as published by the * This program is distributed in the hope that it will be useful, but WITHOUT * ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or * FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for * more details. * You should have received a copy of the GNU General Public License along with'
Found license 'General Public License 2.0' in '* Copyright (C) 2015-2021 Cisco Systems, Inc. and/or its affiliates. All rights reserved. * This program is free software; you can redistribute it and/or modify * it under the terms of the GNU General Public License version 2 as * published by the Free Software Foundation. * This program is distributed in the hope that it will be useful, * but WITHOUT ANY WARRANTY; without even the implied warranty of * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the * GNU General Public License for more details. * You should have received a copy of the GNU General Public License * along with this program; if not, write to the Free Software * MA 02110-1301, USA.'
Found license 'General Public License 2.0' in '* Copyright (C) 2014 Cisco and/or its affiliates. All rights reserved. * This program is free software; you can redistribute it and/or modify * it under the terms of the GNU General Public License version 2 as * published by the Free Software Foundation. * This program is distributed in the hope that it will be useful, * but WITHOUT ANY WARRANTY; without even the implied warranty of * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the * GNU General Public License for more details. * You should have received a copy of the GNU General Public License * along with this program; if not, write to the Free Software * MA 02110-1301, USA.'
Found license 'General Public License 2.0' in '* This program is free software; you can redistribute it and/or modify * it under the terms of the GNU General Public License as published by * the Free Software Foundation; either version 2 of the License, or * (at your option) any later version. * This program is distributed in the hope that it

will be useful, * but WITHOUT ANY WARRANTY; without even the implied warranty of * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the * GNU General Public License for more details. * You should have received a copy of the GNU General Public License * along with this program; if not, write to the Free Software * MA 02110-1301, USA.'
Found license 'General Public License 2.0' in 'This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version. As a special exception to the GNU Lesser General Public License, GNU Libltdl is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. You should have received a copy of the GNU Lesser General Public License along with GNU Libltdl; see the file COPYING.LIB. If not, a copy con be downloaded from http://www.gnu.org/licenses/lgpl.html,'
Found license 'General Public License 2.0' in 'The GNU C Library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version. The GNU C Library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. You should have received a copy of the GNU Lesser General Public License along with the GNU C Library; if not, see <http://www.gnu.org/licenses/>. */ * The Regents of the University of California. All rights reserved. * Redistribution and use in source and binary forms, with or without * modification, are permitted provided that the following conditions * are met: * 1. Redistributions of source code must retain the above copyright * notice, this
list of conditions and the following disclaimer. * 2. Redistributions in binary form must reproduce the above copyright * notice, this list of conditions and the following disclaimer in the * documentation and/or other materials provided with the distribution. * 4. Neither the name of the University nor the names of its contributors * without specific prior written permission.'
Found license 'General Public License 2.0' in '* libmspack is free software; you can redistribute it and/or modify it under * the terms of the GNU Lesser General Public License (LGPL) version 2.1 * This program is distributed in the hope that it will be useful, * but WITHOUT ANY WARRANTY; without even the implied warranty of * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the * GNU Lesser General Public License for more details. * You should have received a copy of the GNU Lesser General Public License * along with this program; if not, write to the Free Software'
Found license 'GNU Lesser General Public License' in 'GNU Libltdl is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version. As a special exception to the GNU Lesser General Public License, GNU Libltdl is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. You should have received a copy of the GNU Lesser General Public License along with GNU Libltdl; see the file COPYING.LIB. If not, a copy can be downloaded from http://www.gnu.org/licenses/lgpl.html,'
Found license 'General Public License 2.0' in '* Copyright (C) 2014-2020 Cisco and/or its affiliates. All rights reserved. * This program is free software; you can redistribute it and/or modify * it under the terms of the GNU General Public License version 2 as * published by the Free Software Foundation. * This program is distributed in the hope that it will be useful, * but WITHOUT ANY WARRANTY; without even the implied warranty of * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the * GNU General Public License for more details. * You should have received a copy of the GNU General Public License * along with this program; if not, write to the Free Software * MA 02110-1301, USA.'

Found license 'General Public License 2.0' in '* Copyright (C) 2016 Cisco and/or its affiliates. All rights reserved. * This program is free software; you can redistribute it and/or modify * it under the terms of the GNU General Public License version 2 as * published by the Free Software Foundation. * This program is distributed in the hope that it will be useful, * but WITHOUT ANY WARRANTY; without even the implied warranty of * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the * GNU General Public License for more details. * You should have received a copy of the GNU General Public License * along with this program; if not, write to the Free Software * MA 02110-1301, USA. * In addition, as a special exception, the copyright holders give * You must obey the GNU General Public License in all respects'

1.25 mbed-tls 3.0.0

1.25.1 Available under license :

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their

Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute
copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding
those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf

of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

1.26 yara 4.1.3

1.26.1 Available under license :

```
# This is the official list of people who can contribute
# (and typically have contributed) code to the YARA repository.
# The AUTHORS file lists the copyright holders; this file
# lists people.  For example, Google employees are listed here
# but not in AUTHORS, because Google holds the copyright.
#
# The submission process automatically checks to make sure
# that people submitting code are listed in this file (by email address).
#
# Names should be added to this file only after verifying that
# the individual or the individual's organization has agreed to
# the appropriate Contributor License Agreement, found here:
```

#
http://code.google.com/legal/individual-cla-v1.0.html
http://code.google.com/legal/corporate-cla-v1.0.html
#
The agreement for individuals can be filled out on the web.
#
When adding J Random Contributor's name to this file,
either J's name or J's organization's name should be
added to the AUTHORS file, depending on whether the
individual or corporate
 CLA was used.

Names should be added to this file like so:
Name <email address>

Please keep the list sorted.

Anthony Desnos <adesnos@google.com>
Antonio Vargas Gonzalez <winden@google.com>
Christian Blichmann <cblichmann@google.com>
Hilko Bengen <bengen@hilluzination.de>
Joachim Metz <joachim.metz@gmail.com>
Karl Hiramoto <karl.hiramoto@virustotal.com>
Mike Wiacek <mjwiacek@google.com>
Shane Huntley <shuntley@google.com>
Stefan Buehlmann <stefan.buehlmann@joebox.org>
Victor M. Alvarez <plusvic@gmail.com>;<vmalvarez@virustotal.com>
Wesley Shields <wxs@atarininja.org>

HOLDER OR CONTRIBUTORS BE LIABLE FOR
ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON
ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.27 httpparser 2.7.1

1.27.1 Available under license :

http_parser.c is based on src/http/ngx_http_parse.c from NGINX copyright
Igor Sysoev.

Additional changes are licensed under the same terms as NGINX and
copyright Joyent, Inc. and other Node contributors. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to
deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or
sell copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT
SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS
IN THE SOFTWARE.

1.28 llvm 3.6.0

1.28.1 Available under license :

; NOTE: Assertions have been autogenerated by utils/update_llc_test_checks.py
; RUN: llc < %s -mtriple=aarch64-- | FileCheck %s

; A shuffle mask with all undef elements is always legal.

define <4 x i32> @PR41535(<2 x i32> %p1, <2 x i32> %p2) {
; CHECK-LABEL: PR41535:

```
; CHECK:       // %bb.0:
; CHECK-NEXT:    ext v0.8b, v0.8b, v1.8b, #4
; CHECK-NEXT:    mov v0.d[1], v0.d[0]
; CHECK-NEXT:    ret
 %cat1 = shufflevector <2 x i32> %p1, <2 x i32> undef, <4 x i32> <i32 undef, i32 1, i32 undef, i32 undef>
 %cat2 = shufflevector <2 x i32> %p2, <2 x i32> undef, <4 x i32> <i32 0, i32 undef, i32 undef, i32 undef>
 %r = shufflevector <4 x i32> %cat1, <4 x i32> %cat2, <4 x i32> <i32 undef, i32 undef, i32 1, i32 4>
 ret <4 x i32> %r
}
# People who have agreed to one of the CLAs and can contribute patches.
# The AUTHORS file lists the copyright holders; this file
# lists people.  For example, Google employees are listed here
# but not in AUTHORS, because Google holds the copyright.
#
# Names should be added to this file only after verifying that
# the individual or the individual's organization has agreed to
# the appropriate Contributor License Agreement, found here:
#
# https://developers.google.com/open-source/cla/individual
# https://developers.google.com/open-source/cla/corporate
#
# The agreement for individuals can be filled out on the web.
#
# When adding J Random Contributor's name to this file,
# either J's name or J's organization's name should be
# added to the AUTHORS file, depending on whether the
# individual or corporate CLA was used.
#
# Names should be added to this file as:
#     Name <email address>
#
# Please keep the list sorted.

Abhina Sreeskantharajan <abhina.sreeskantharajan@ibm.com>
Albert Pretorius
 <pretoalb@gmail.com>
Alex Steele <steelal123@gmail.com>
Andriy Berestovskyy <berestovskyy@gmail.com>
Arne Beer <arne@twobeer.de>
Billy Robert O'Neal III <billy.oneal@gmail.com> <bion@microsoft.com>
Chris Kennelly <ckennelly@google.com> <ckennelly@ckennelly.com>
Christian Wassermann <christian_wassermann@web.de>
Christopher Seymour <chris.j.seymour@hotmail.com>
Colin Braley <braley.colin@gmail.com>
Cyrille Faucheux <cyrille.faucheux@gmail.com>
Daniel Harvey <danielharvey458@gmail.com>
David Coeurjolly <david.coeurjolly@liris.cnrs.fr>
Deniz Evrenci <denizevrenci@gmail.com>
```

Dominic Hamon <dma@stripysock.com> <dominic@google.com>
Dominik Czarnota <dominik.b.czarnota@gmail.com>
Eric Backus <eric_backus@alum.mit.edu>
Eric Fiselier <eric@efcs.ca>
Eugene Zhuk <eugene.zhuk@gmail.com>
Evgeny Safronov <division494@gmail.com>
Fanbo Meng <fanbo.meng@ibm.com>
Federico Ficarelli <federico.ficarelli@gmail.com>
Felix Homann <linuxaudio@showlabor.de>
Geoffrey Martin-Noble <gcmn@google.com> <gmngeoffrey@gmail.com>
Gerg
Szitr <szitar.gergo@gmail.com>
Hannes Hauswedell <h2@fsfe.org>
Ismael Jimenez Martinez <ismael.jimenez.martinez@gmail.com>
Jern-Kuan Leong <jernkuan@gmail.com>
JianXiong Zhou <zhoujianxiong2@gmail.com>
Joao Paulo Magalhaes <joaoppmagalhaes@gmail.com>
John Millikin <jmillikin@stripe.com>
Jordan Williams <jwillikers@protonmail.com>
Jussi Knuuttila <jussi.knuuttila@gmail.com>
Kai Wolf <kai.wolf@gmail.com>
Kaito Udagawa <umireon@gmail.com>
Kishan Kumar <kumar.kishan@outlook.com>
Lei Xu <eddyxu@gmail.com>
Matt Clarkson <mattyclarkson@gmail.com>
Maxim Vafin <maxvafin@gmail.com>
Nick Hutchinson <nshutchinson@gmail.com>
Norman Heino <norman.heino@gmail.com>
Oleksandr Sochka <sasha.sochka@gmail.com>
Ori Livneh <ori.livneh@gmail.com>
Pascal Leroy <phl@google.com>
Paul Redmond <paul.redmond@gmail.com>
Pierre Phaneuf <pphaneuf@google.com>
Radoslav Yovchev <radoslav.tm@gmail.com>
Raul Marin <rmrodriguez@cartodb.com>
Ray Glover <ray.glover@uk.ibm.com>
Robert Guo <robert.guo@mongodb.com>
Roman
Lebedev <lebedev.ri@gmail.com>
Sayan Bhattacharjee <aero.sayan@gmail.com>
Shuo Chen <chenshuo@chenshuo.com>
Steven Wan <wan.yu@ibm.com>
Tobias Schmidt <tobias.schmidt@in.tum.de>
Tobias Ulvgrd <tobias.ulvgard@dirac.se>
Tom Madams <tom.ej.madams@gmail.com> <tmadams@google.com>
Yixuan Qiu <yixuanq@gmail.com>
Yusuke Suzuki <utatane.tea@gmail.com>
Zbigniew Skowron <zbychs@gmail.com>

Min-Yih Hsu <yihshyng223@gmail.com>
; RUN: llc < %s -mtriple=s390x-linux-gnu -mcpu=zEC12 -verify-machineinstrs | FileCheck %s
;
; Test that early if conversion produces LOCR with operands of the right
; register classes.

define void @autogen_SD4739(i8*) {
; CHECK-NOT: Expected a GR32Bit register, but got a GRX32Bit register
BB:
 %L34 = load i8, i8* %0
 %Cmp56 = icmp sgt i8 undef, %L34
 br label %CF246

CF246:   ; preds = %CF246, %BB
 %Sl163 = select i1 %Cmp56, i8 %L34, i8 undef
 br i1 undef, label %CF246, label %CF248

CF248:   ; preds = %CF248, %CF246
 store i8 %Sl163, i8* %0
 br label %CF248
}
Ptyprocess is under the ISC license, as code derived from Pexpect.
   http://opensource.org/licenses/ISC

Copyright (c) 2013-2014, Pexpect development team
Copyright (c) 2012, Noah Spurrier <noah@noah.org>

PERMISSION TO USE, COPY, MODIFY, AND/OR DISTRIBUTE THIS SOFTWARE FOR ANY PURPOSE
WITH OR WITHOUT FEE IS HEREBY GRANTED, PROVIDED THAT THE ABOVE COPYRIGHT NOTICE
AND THIS PERMISSION NOTICE APPEAR IN ALL COPIES. THE SOFTWARE IS PROVIDED
"AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE
INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT
SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL
DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS,
WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING
OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
==
The LLVM Project is under the Apache License v2.0 with LLVM Exceptions:
==

                                 Apache License
                           Version 2.0, January 2004
                        http://www.apache.org/licenses/

   TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

   1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the
power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached
to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition,

"submitted"
means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work
includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional
attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License
does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each

Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including
negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such
obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

---- LLVM Exceptions to the Apache 2.0 License ----

As an exception, if, as a result of your compiling your source code, portions
of this Software are embedded into an Object form of such source code, you
may redistribute such embedded portions in such Object form without complying
with the conditions of Sections 4(a), 4(b) and 4(d) of the License.

In addition, if you combine or link compiled
forms of this Software with
software that is licensed under the GPLv2 ("Combined Software") and if a
court of competent jurisdiction determines that the patent provision (Section
3), the indemnity provision (Section 9) or other Section of the License
conflicts with the conditions of the GPLv2, you may retroactively and
prospectively choose to deem waived or otherwise exclude such Section(s) of
the License, but only in their entirety and only with respect to the Combined
Software.

==
Software from third parties included in the LLVM Project:
==
The LLVM Project contains third party software which is under different license
terms. All such code will be identified clearly using at least one of two
mechanisms:
1) It will be in a separate directory tree with its own `LICENSE.txt` or
`LICENSE` file at the top containing the specific
license and restrictions
which apply to that software, or
2) It will contain specific license and restriction terms at the top of every
file.
; RUN: opt -mtriple=aarch64-linux-gnu -mattr=+sve -scalarize-masked-mem-intrin -S < %s | FileCheck %s

; Testing that masked gathers operating on scalable vectors that are
; packed in SVE registers are not scalarized.

; CHECK-LABEL: @masked_gather_nxv4i32(

```
; CHECK: call <vscale x 4 x i32> @llvm.masked.gather.nxv4i32
define <vscale x 4 x i32> @masked_gather_nxv4i32(<vscale x 4 x i32*> %ld, <vscale x 4 x i1> %masks, <vscale x 4 x i32> %passthru) {
 %res = call <vscale x 4 x i32> @llvm.masked.gather.nxv4i32(<vscale x 4 x i32*> %ld, i32 0, <vscale x 4 x i1> %masks, <vscale x 4 x i32> %passthru)
 ret <vscale x 4 x i32> %res
}

; Testing that masked gathers operating on scalable vectors of FP data
; that is packed in SVE registers are not scalarized.

; CHECK-LABEL: @masked_gather_nxv2f64(
; CHECK: call <vscale x 2 x double> @llvm.masked.gather.nxv2f64
define <vscale x 2 x double> @masked_gather_nxv2f64(<vscale x 2 x double*> %ld, <vscale x 2 x i1> %masks, <vscale x 2 x double> %passthru)
 {
 %res = call <vscale x 2 x double> @llvm.masked.gather.nxv2f64(<vscale x 2 x double*> %ld, i32 0, <vscale x 2 x i1> %masks, <vscale x 2 x double> %passthru)
 ret <vscale x 2 x double> %res
}

; Testing that masked gathers operating on scalable vectors of FP data
; that is unpacked in SVE registers are not scalarized.

; CHECK-LABEL: @masked_gather_nxv2f16(
; CHECK: call <vscale x 2 x half> @llvm.masked.gather.nxv2f16
define <vscale x 2 x half> @masked_gather_nxv2f16(<vscale x 2 x half*> %ld, <vscale x 2 x i1> %masks, <vscale x 2 x half> %passthru) {
 %res = call <vscale x 2 x half> @llvm.masked.gather.nxv2f16(<vscale x 2 x half*> %ld, i32 0, <vscale x 2 x i1> %masks, <vscale x 2 x half> %passthru)
 ret <vscale x 2 x half> %res
}

; Testing that masked gathers operating on 64-bit fixed vectors are
; scalarized because NEON doesn't have support for masked gather
; instructions.

; CHECK-LABEL: @masked_gather_v2f32(
; CHECK-NOT: @llvm.masked.gather.v2f32(
define <2 x float> @masked_gather_v2f32(<2
 x float*> %ld, <2 x i1> %masks, <2 x float> %passthru) {
 %res = call <2 x float> @llvm.masked.gather.v2f32(<2 x float*> %ld, i32 0, <2 x i1> %masks, <2 x float> %passthru)
 ret <2 x float> %res
}

; Testing that masked gathers operating on 128-bit fixed vectors are
; scalarized because NEON doesn't have support for masked gather
```

```
; instructions and because we are not targeting fixed width SVE.

; CHECK-LABEL: @masked_gather_v4i32(
; CHECK-NOT: @llvm.masked.gather.v4i32(
define <4 x i32> @masked_gather_v4i32(<4 x i32*> %ld, <4 x i1> %masks, <4 x i32> %passthru) {
 %res = call <4 x i32> @llvm.masked.gather.v4i32(<4 x i32*> %ld, i32 0, <4 x i1> %masks, <4 x i32> %passthru)
 ret <4 x i32> %res
}

declare <vscale x 4 x i32> @llvm.masked.gather.nxv4i32(<vscale x 4 x i32*> %ptrs, i32 %align, <vscale x 4 x i1>
%masks, <vscale x 4 x i32> %passthru)
declare <vscale x 2 x double> @llvm.masked.gather.nxv2f64(<vscale x 2 x double*> %ptrs, i32 %align, <vscale x
2 x i1> %masks, <vscale
 x 2 x double> %passthru)
declare <vscale x 2 x half> @llvm.masked.gather.nxv2f16(<vscale x 2 x half*> %ptrs, i32 %align, <vscale x 2 x
i1> %masks, <vscale x 2 x half> %passthru)
declare <2 x float> @llvm.masked.gather.v2f32(<2 x float*> %ptrs, i32 %align, <2 x i1> %masks, <2 x float>
%passthru)
declare <4 x i32> @llvm.masked.gather.v4i32(<4 x i32*> %ptrs, i32 %align, <4 x i1> %masks, <4 x i32>
%passthru)
; NOTE: Assertions have been autogenerated by utils/update_test_checks.py UTC_ARGS: --function-signature --
check-attributes --check-globals
; RUN: opt -attributor -enable-new-pm=0 -attributor-manifest-internal  -attributor-max-iterations-verify -attributor-
annotate-decl-cs -attributor-max-iterations=6 -S < %s | FileCheck %s --check-
prefixes=CHECK,NOT_CGSCC_NPM,NOT_CGSCC_OPM,NOT_TUNIT_NPM,IS__TUNIT____,IS________OP
M,IS__TUNIT_OPM
; RUN: opt -aa-pipeline=basic-aa -passes=attributor -attributor-manifest-internal  -attributor-max-iterations-verify -
attributor-annotate-decl-cs -attributor-max-iterations=6 -S < %s | FileCheck %s --check-
prefixes=CHECK,NOT_CGSCC_OPM,NOT_CGSCC_NPM,NOT_TUNIT_OPM,IS__TUNIT____,IS________NP
M,IS__TUNIT_NPM
; RUN: opt -attributor-cgscc -enable-new-pm=0 -attributor-manifest-internal  -attributor-annotate-decl-cs -S < %s |
FileCheck %s --check-
prefixes=CHECK,NOT_TUNIT_NPM,NOT_TUNIT_OPM,NOT_CGSCC_NPM,IS__CGSCC____,IS________OP
M,IS__CGSCC_OPM
; RUN: opt -aa-pipeline=basic-aa
 -passes=attributor-cgscc -attributor-manifest-internal  -attributor-annotate-decl-cs -S < %s | FileCheck %s --check-
prefixes=CHECK,NOT_TUNIT_NPM,NOT_TUNIT_OPM,NOT_CGSCC_OPM,IS__CGSCC____,IS________NP
M,IS__CGSCC_NPM
; Test that we only promote arguments when the caller/callee have compatible
; function attrubtes.

target triple = "x86_64-unknown-linux-gnu"

; This should promote
define internal fastcc void @callee_avx512_legal512_prefer512_call_avx512_legal512_prefer512(<8 x i64>* %arg,
<8 x i64>* readonly %arg1) #0 {
;
; IS________OPM: Function Attrs: argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn
```

```
; IS_____OPM-LABEL: define
{{[^@]+}}@callee_avx512_legal512_prefer512_call_avx512_legal512_prefer512
; IS_____OPM-SAME: (<8 x i64>* nocapture nofree noundef nonnull writeonly align 64 dereferenceable(64)
[[ARG:%.*]], <8 x i64>* nocapture nofree noundef nonnull readonly align 64 dereferenceable(64) [[ARG1:%.*]])
#[[ATTR0:[0-9]+]] {
; IS_____OPM-NEXT:
 bb:
; IS_____OPM-NEXT:    [[TMP:%.*]] = load <8 x i64>, <8 x i64>* [[ARG1]], align 64
; IS_____OPM-NEXT:    store <8 x i64> [[TMP]], <8 x i64>* [[ARG]], align 64
; IS_____OPM-NEXT:    ret void
;
; IS__TUNIT_NPM: Function Attrs: argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn
; IS__TUNIT_NPM-LABEL: define
{{[^@]+}}@callee_avx512_legal512_prefer512_call_avx512_legal512_prefer512
; IS__TUNIT_NPM-SAME: (<8 x i64>* noalias nocapture nofree noundef nonnull writeonly align 64
dereferenceable(64) [[ARG:%.*]], <8 x i64> [[TMP0:%.*]]) #[[ATTR0:[0-9]+]] {
; IS__TUNIT_NPM-NEXT:  bb:
; IS__TUNIT_NPM-NEXT:    [[ARG1_PRIV:%.*]] = alloca <8 x i64>, align 64
; IS__TUNIT_NPM-NEXT:    store <8 x i64> [[TMP0]], <8 x i64>* [[ARG1_PRIV]], align 64
; IS__TUNIT_NPM-NEXT:    [[TMP:%.*]] = load <8 x i64>, <8 x i64>* [[ARG1_PRIV]], align 64
; IS__TUNIT_NPM-NEXT:    store <8 x i64> [[TMP]], <8 x i64>* [[ARG]], align 64
; IS__TUNIT_NPM-NEXT:    ret void
;
; IS__CGSCC_NPM: Function
 Attrs: argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn
; IS__CGSCC_NPM-LABEL: define
{{[^@]+}}@callee_avx512_legal512_prefer512_call_avx512_legal512_prefer512
; IS__CGSCC_NPM-SAME: (<8 x i64>* noalias nocapture nofree noundef nonnull writeonly align 64
dereferenceable(64) [[ARG:%.*]], <8 x i64> [[TMP0:%.*]]) #[[ATTR0:[0-9]+]] {
; IS__CGSCC_NPM-NEXT:  bb:
; IS__CGSCC_NPM-NEXT:    [[ARG1_PRIV:%.*]] = alloca <8 x i64>, align 64
; IS__CGSCC_NPM-NEXT:    store <8 x i64> [[TMP0]], <8 x i64>* [[ARG1_PRIV]], align 64
; IS__CGSCC_NPM-NEXT:    [[TMP:%.*]] = load <8 x i64>, <8 x i64>* [[ARG1_PRIV]], align 64
; IS__CGSCC_NPM-NEXT:    store <8 x i64> [[TMP0]], <8 x i64>* [[ARG]], align 64
; IS__CGSCC_NPM-NEXT:    ret void
;
bb:
 %tmp = load <8 x i64>, <8 x i64>* %arg1
 store <8 x i64> %tmp, <8 x i64>* %arg
 ret void
}

define void @avx512_legal512_prefer512_call_avx512_legal512_prefer512(<8 x i64>* %arg) #0 {
;
; IS__TUNIT_OPM: Function Attrs: argmemonly inlinehint nofree
 norecurse nosync nounwind uwtable willreturn
; IS__TUNIT_OPM-LABEL: define {{[^@]+}}@avx512_legal512_prefer512_call_avx512_legal512_prefer512
; IS__TUNIT_OPM-SAME: (<8 x i64>* nocapture nofree writeonly [[ARG:%.*]]) #[[ATTR0]] {
```

```
; IS__TUNIT_OPM-NEXT:  bb:
; IS__TUNIT_OPM-NEXT:    [[TMP:%.*]] = alloca <8 x i64>, align 32
; IS__TUNIT_OPM-NEXT:    [[TMP2:%.*]] = alloca <8 x i64>, align 32
; IS__TUNIT_OPM-NEXT:    [[TMP3:%.*]] = bitcast <8 x i64>* [[TMP]] to i8*
; IS__TUNIT_OPM-NEXT:    call void @llvm.memset.p0i8.i64(i8* nocapture nofree noundef nonnull writeonly
align 64 dereferenceable(64) [[TMP3]], i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR6:[0-9]+]]
; IS__TUNIT_OPM-NEXT:    call fastcc void
@callee_avx512_legal512_prefer512_call_avx512_legal512_prefer512(<8 x i64>* nocapture nofree noundef
nonnull writeonly align 64 dereferenceable(64) [[TMP2]], <8 x i64>* nocapture nofree noundef nonnull readonly
align 64 dereferenceable(64) [[TMP]]) #[[ATTR7:[0-9]+]]
; IS__TUNIT_OPM-NEXT:
   [[TMP4:%.*]] = load <8 x i64>, <8 x i64>* [[TMP2]], align 64
; IS__TUNIT_OPM-NEXT:    store <8 x i64> [[TMP4]], <8 x i64>* [[ARG]], align 2
; IS__TUNIT_OPM-NEXT:    ret void
;
; IS__TUNIT_NPM: Function Attrs: argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn
; IS__TUNIT_NPM-LABEL: define {{[^@]+}}@avx512_legal512_prefer512_call_avx512_legal512_prefer512
; IS__TUNIT_NPM-SAME: (<8 x i64>* nocapture nofree writeonly [[ARG:%.*]]) #[[ATTR0]] {
; IS__TUNIT_NPM-NEXT:  bb:
; IS__TUNIT_NPM-NEXT:    [[TMP:%.*]] = alloca <8 x i64>, align 32
; IS__TUNIT_NPM-NEXT:    [[TMP2:%.*]] = alloca <8 x i64>, align 32
; IS__TUNIT_NPM-NEXT:    [[TMP3:%.*]] = bitcast <8 x i64>* [[TMP]] to i8*
; IS__TUNIT_NPM-NEXT:    call void @llvm.memset.p0i8.i64(i8* nocapture nofree noundef nonnull writeonly
align 64 dereferenceable(64) [[TMP3]], i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR6:[0-9]+]]
; IS__TUNIT_NPM-NEXT:    [[TMP0:%.*]] = load <8 x i64>, <8 x i64>* [[TMP]], align
 64
; IS__TUNIT_NPM-NEXT:    call fastcc void
@callee_avx512_legal512_prefer512_call_avx512_legal512_prefer512(<8 x i64>* noalias nocapture nofree
noundef nonnull writeonly align 64 dereferenceable(64) [[TMP2]], <8 x i64> [[TMP0]]) #[[ATTR7:[0-9]+]]
; IS__TUNIT_NPM-NEXT:    [[TMP4:%.*]] = load <8 x i64>, <8 x i64>* [[TMP2]], align 64
; IS__TUNIT_NPM-NEXT:    store <8 x i64> [[TMP4]], <8 x i64>* [[ARG]], align 2
; IS__TUNIT_NPM-NEXT:    ret void
;
; IS__CGSCC_OPM: Function Attrs: argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn
; IS__CGSCC_OPM-LABEL: define {{[^@]+}}@avx512_legal512_prefer512_call_avx512_legal512_prefer512
; IS__CGSCC_OPM-SAME: (<8 x i64>* nocapture nofree noundef nonnull writeonly align 2 dereferenceable(64)
[[ARG:%.*]]) #[[ATTR0]] {
; IS__CGSCC_OPM-NEXT:  bb:
; IS__CGSCC_OPM-NEXT:    [[TMP:%.*]] = alloca <8 x i64>, align 32
; IS__CGSCC_OPM-NEXT:    [[TMP2:%.*]] = alloca <8 x i64>, align 32
; IS__CGSCC_OPM-NEXT:    [[TMP3:%.*]] = bitcast <8 x
 i64>* [[TMP]] to i8*
; IS__CGSCC_OPM-NEXT:    call void @llvm.memset.p0i8.i64(i8* nocapture nofree noundef nonnull writeonly
align 64 dereferenceable(64) [[TMP3]], i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR6:[0-9]+]]
; IS__CGSCC_OPM-NEXT:    call fastcc void
@callee_avx512_legal512_prefer512_call_avx512_legal512_prefer512(<8 x i64>* nocapture nofree noundef
nonnull writeonly align 64 dereferenceable(64) [[TMP2]], <8 x i64>* nocapture nofree noundef nonnull readonly
align 64 dereferenceable(64) [[TMP]]) #[[ATTR7:[0-9]+]]
```

```
; IS__CGSCC_OPM-NEXT:    [[TMP4:%.*]] = load <8 x i64>, <8 x i64>* [[TMP2]], align 64
; IS__CGSCC_OPM-NEXT:    store <8 x i64> [[TMP4]], <8 x i64>* [[ARG]], align 2
; IS__CGSCC_OPM-NEXT:    ret void
;
; IS__CGSCC_NPM: Function Attrs: argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn
; IS__CGSCC_NPM-LABEL: define {{[^@]+}}@avx512_legal512_prefer512_call_avx512_legal512_prefer512
; IS__CGSCC_NPM-SAME: (<8 x i64>* nocapture nofree noundef
 nonnull writeonly align 2 dereferenceable(64) [[ARG:%.*]]) #[[ATTR0]] {
; IS__CGSCC_NPM-NEXT:  bb:
; IS__CGSCC_NPM-NEXT:    [[TMP:%.*]] = alloca <8 x i64>, align 32
; IS__CGSCC_NPM-NEXT:    [[TMP2:%.*]] = alloca <8 x i64>, align 32
; IS__CGSCC_NPM-NEXT:    [[TMP3:%.*]] = bitcast <8 x i64>* [[TMP]] to i8*
; IS__CGSCC_NPM-NEXT:    call void @llvm.memset.p0i8.i64(i8* nocapture nofree noundef nonnull writeonly
align 64 dereferenceable(64) [[TMP3]], i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR6:[0-9]+]]
; IS__CGSCC_NPM-NEXT:    [[TMP0:%.*]] = load <8 x i64>, <8 x i64>* [[TMP]], align 64
; IS__CGSCC_NPM-NEXT:    call fastcc void
@callee_avx512_legal512_prefer512_call_avx512_legal512_prefer512(<8 x i64>* noalias nocapture nofree
noundef nonnull writeonly align 64 dereferenceable(64) [[TMP2]], <8 x i64> [[TMP0]]) #[[ATTR7:[0-9]+]]
; IS__CGSCC_NPM-NEXT:    [[TMP4:%.*]] = load <8 x i64>, <8 x i64>* [[TMP2]], align 64
; IS__CGSCC_NPM-NEXT:    store <8 x i64> [[TMP4]], <8 x i64>* [[ARG]],
 align 2
; IS__CGSCC_NPM-NEXT:    ret void
;
bb:
 %tmp = alloca <8 x i64>, align 32
 %tmp2 = alloca <8 x i64>, align 32
 %tmp3 = bitcast <8 x i64>* %tmp to i8*
 call void @llvm.memset.p0i8.i64(i8* align 32 %tmp3, i8 0, i64 32, i1 false)
 call fastcc void @callee_avx512_legal512_prefer512_call_avx512_legal512_prefer512(<8 x i64>* %tmp2, <8 x
i64>* %tmp)
 %tmp4 = load <8 x i64>, <8 x i64>* %tmp2, align 32
 store <8 x i64> %tmp4, <8 x i64>* %arg, align 2
 ret void
}

; This should promote
define internal fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal512_prefer256(<8 x i64>* %arg,
<8 x i64>* readonly %arg1) #1 {
;
; IS________OPM: Function Attrs: argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn
; IS________OPM-LABEL: define
{{[^@]+}}@callee_avx512_legal512_prefer256_call_avx512_legal512_prefer256
; IS________OPM-SAME: (<8 x i64>* nocapture nofree noundef nonnull writeonly align 64 dereferenceable(64)
[[ARG:%.*]], <8 x i64>* nocapture
 nofree noundef nonnull readonly align 64 dereferenceable(64) [[ARG1:%.*]]) #[[ATTR1:[0-9]+]] {
; IS________OPM-NEXT:  bb:
; IS________OPM-NEXT:    [[TMP:%.*]] = load <8 x i64>, <8 x i64>* [[ARG1]], align 64
; IS________OPM-NEXT:    store <8 x i64> [[TMP]], <8 x i64>* [[ARG]], align 64
```

```
; IS_____OPM-NEXT:    ret void
;
; IS__TUNIT_NPM: Function Attrs: argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn
; IS__TUNIT_NPM-LABEL: define
{{[^@]+}}@callee_avx512_legal512_prefer256_call_avx512_legal512_prefer256
; IS__TUNIT_NPM-SAME: (<8 x i64>* noalias nocapture nofree noundef nonnull writeonly align 64
dereferenceable(64) [[ARG:%.*]], <8 x i64> [[TMP0:%.*]]) #[[ATTR1:[0-9]+]] {
; IS__TUNIT_NPM-NEXT:  bb:
; IS__TUNIT_NPM-NEXT:    [[ARG1_PRIV:%.*]] = alloca <8 x i64>, align 64
; IS__TUNIT_NPM-NEXT:    store <8 x i64> [[TMP0]], <8 x i64>* [[ARG1_PRIV]], align 64
; IS__TUNIT_NPM-NEXT:    [[TMP:%.*]] = load <8 x i64>, <8 x i64>* [[ARG1_PRIV]], align 64
; IS__TUNIT_NPM-NEXT:
   store <8 x i64> [[TMP]], <8 x i64>* [[ARG]], align 64
; IS__TUNIT_NPM-NEXT:    ret void
;
; IS__CGSCC_NPM: Function Attrs: argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn
; IS__CGSCC_NPM-LABEL: define
{{[^@]+}}@callee_avx512_legal512_prefer256_call_avx512_legal512_prefer256
; IS__CGSCC_NPM-SAME: (<8 x i64>* noalias nocapture nofree noundef nonnull writeonly align 64
dereferenceable(64) [[ARG:%.*]], <8 x i64> [[TMP0:%.*]]) #[[ATTR1:[0-9]+]] {
; IS__CGSCC_NPM-NEXT:  bb:
; IS__CGSCC_NPM-NEXT:    [[ARG1_PRIV:%.*]] = alloca <8 x i64>, align 64
; IS__CGSCC_NPM-NEXT:    store <8 x i64> [[TMP0]], <8 x i64>* [[ARG1_PRIV]], align 64
; IS__CGSCC_NPM-NEXT:    [[TMP:%.*]] = load <8 x i64>, <8 x i64>* [[ARG1_PRIV]], align 64
; IS__CGSCC_NPM-NEXT:    store <8 x i64> [[TMP0]], <8 x i64>* [[ARG]], align 64
; IS__CGSCC_NPM-NEXT:    ret void
;
bb:
 %tmp = load <8 x i64>, <8 x i64>* %arg1
 store <8 x i64> %tmp, <8 x i64>* %arg
 ret void
}

define void @avx512_legal512_prefer256_call_avx512_legal512_prefer256(<8
 x i64>* %arg) #1 {
;
; IS__TUNIT_OPM: Function Attrs: argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn
; IS__TUNIT_OPM-LABEL: define {{[^@]+}}@avx512_legal512_prefer256_call_avx512_legal512_prefer256
; IS__TUNIT_OPM-SAME: (<8 x i64>* nocapture nofree writeonly [[ARG:%.*]]) #[[ATTR1]] {
; IS__TUNIT_OPM-NEXT:  bb:
; IS__TUNIT_OPM-NEXT:    [[TMP:%.*]] = alloca <8 x i64>, align 32
; IS__TUNIT_OPM-NEXT:    [[TMP2:%.*]] = alloca <8 x i64>, align 32
; IS__TUNIT_OPM-NEXT:    [[TMP3:%.*]] = bitcast <8 x i64>* [[TMP]] to i8*
; IS__TUNIT_OPM-NEXT:    call void @llvm.memset.p0i8.i64(i8* nocapture nofree noundef nonnull writeonly
align 64 dereferenceable(64) [[TMP3]], i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR6]]
; IS__TUNIT_OPM-NEXT:    call fastcc void
@callee_avx512_legal512_prefer256_call_avx512_legal512_prefer256(<8 x i64>* nocapture nofree noundef
nonnull writeonly align 64 dereferenceable(64) [[TMP2]], <8 x i64>* nocapture
```

```
 nofree noundef nonnull readonly align 64 dereferenceable(64) [[TMP]]) #[[ATTR7]]
; IS__TUNIT_OPM-NEXT:    [[TMP4:%.*]] = load <8 x i64>, <8 x i64>* [[TMP2]], align 64
; IS__TUNIT_OPM-NEXT:    store <8 x i64> [[TMP4]], <8 x i64>* [[ARG]], align 2
; IS__TUNIT_OPM-NEXT:    ret void
;
; IS__TUNIT_NPM: Function Attrs: argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn
; IS__TUNIT_NPM-LABEL: define {{[^@]+}}@avx512_legal512_prefer256_call_avx512_legal512_prefer256
; IS__TUNIT_NPM-SAME: (<8 x i64>* nocapture nofree writeonly [[ARG:%.*]]) #[[ATTR1]] {
; IS__TUNIT_NPM-NEXT:  bb:
; IS__TUNIT_NPM-NEXT:    [[TMP:%.*]] = alloca <8 x i64>, align 32
; IS__TUNIT_NPM-NEXT:    [[TMP2:%.*]] = alloca <8 x i64>, align 32
; IS__TUNIT_NPM-NEXT:    [[TMP3:%.*]] = bitcast <8 x i64>* [[TMP]] to i8*
; IS__TUNIT_NPM-NEXT:    call void @llvm.memset.p0i8.i64(i8* nocapture nofree noundef nonnull writeonly
align 64 dereferenceable(64) [[TMP3]], i8 noundef 0, i64 noundef 32, i1 noundef
 false) #[[ATTR6]]
; IS__TUNIT_NPM-NEXT:    [[TMP0:%.*]] = load <8 x i64>, <8 x i64>* [[TMP]], align 64
; IS__TUNIT_NPM-NEXT:    call fastcc void
@callee_avx512_legal512_prefer256_call_avx512_legal512_prefer256(<8 x i64>* noalias nocapture nofree
noundef nonnull writeonly align 64 dereferenceable(64) [[TMP2]], <8 x i64> [[TMP0]]) #[[ATTR7]]
; IS__TUNIT_NPM-NEXT:    [[TMP4:%.*]] = load <8 x i64>, <8 x i64>* [[TMP2]], align 64
; IS__TUNIT_NPM-NEXT:    store <8 x i64> [[TMP4]], <8 x i64>* [[ARG]], align 2
; IS__TUNIT_NPM-NEXT:    ret void
;
; IS__CGSCC_OPM: Function Attrs: argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn
; IS__CGSCC_OPM-LABEL: define {{[^@]+}}@avx512_legal512_prefer256_call_avx512_legal512_prefer256
; IS__CGSCC_OPM-SAME: (<8 x i64>* nocapture nofree noundef nonnull writeonly align 2 dereferenceable(64)
[[ARG:%.*]]) #[[ATTR1]] {
; IS__CGSCC_OPM-NEXT:  bb:
; IS__CGSCC_OPM-NEXT:    [[TMP:%.*]] = alloca <8 x i64>, align 32
; IS__CGSCC_OPM-NEXT:    [[TMP2:%.*]]
 = alloca <8 x i64>, align 32
; IS__CGSCC_OPM-NEXT:    [[TMP3:%.*]] = bitcast <8 x i64>* [[TMP]] to i8*
; IS__CGSCC_OPM-NEXT:    call void @llvm.memset.p0i8.i64(i8* nocapture nofree noundef nonnull writeonly
align 64 dereferenceable(64) [[TMP3]], i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR6]]
; IS__CGSCC_OPM-NEXT:    call fastcc void
@callee_avx512_legal512_prefer256_call_avx512_legal512_prefer256(<8 x i64>* nocapture nofree noundef
nonnull writeonly align 64 dereferenceable(64) [[TMP2]], <8 x i64>* nocapture nofree noundef nonnull readonly
align 64 dereferenceable(64) [[TMP]]) #[[ATTR7]]
; IS__CGSCC_OPM-NEXT:    [[TMP4:%.*]] = load <8 x i64>, <8 x i64>* [[TMP2]], align 64
; IS__CGSCC_OPM-NEXT:    store <8 x i64> [[TMP4]], <8 x i64>* [[ARG]], align 2
; IS__CGSCC_OPM-NEXT:    ret void
;
; IS__CGSCC_NPM: Function Attrs: argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn
; IS__CGSCC_NPM-LABEL: define {{[^@]+}}@avx512_legal512_prefer256_call_avx512_legal512_prefer256
;
 IS__CGSCC_NPM-SAME: (<8 x i64>* nocapture nofree noundef nonnull writeonly align 2 dereferenceable(64)
[[ARG:%.*]]) #[[ATTR1]] {
; IS__CGSCC_NPM-NEXT:  bb:
```

```
; IS__CGSCC_NPM-NEXT:    [[TMP:%.*]] = alloca <8 x i64>, align 32
; IS__CGSCC_NPM-NEXT:    [[TMP2:%.*]] = alloca <8 x i64>, align 32
; IS__CGSCC_NPM-NEXT:    [[TMP3:%.*]] = bitcast <8 x i64>* [[TMP]] to i8*
; IS__CGSCC_NPM-NEXT:    call void @llvm.memset.p0i8.i64(i8* nocapture nofree noundef nonnull writeonly
align 64 dereferenceable(64) [[TMP3]], i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR6]]
; IS__CGSCC_NPM-NEXT:    [[TMP0:%.*]] = load <8 x i64>, <8 x i64>* [[TMP]], align 64
; IS__CGSCC_NPM-NEXT:    call fastcc void
@callee_avx512_legal512_prefer256_call_avx512_legal512_prefer256(<8 x i64>* noalias nocapture nofree
noundef nonnull writeonly align 64 dereferenceable(64) [[TMP2]], <8 x i64> [[TMP0]]) #[[ATTR7]]
; IS__CGSCC_NPM-NEXT:    [[TMP4:%.*]] = load <8 x i64>, <8 x i64>* [[TMP2]], align 64
;
 IS__CGSCC_NPM-NEXT:    store <8 x i64> [[TMP4]], <8 x i64>* [[ARG]], align 2
; IS__CGSCC_NPM-NEXT:    ret void
;
bb:
 %tmp = alloca <8 x i64>, align 32
 %tmp2 = alloca <8 x i64>, align 32
 %tmp3 = bitcast <8 x i64>* %tmp to i8*
 call void @llvm.memset.p0i8.i64(i8* align 32 %tmp3, i8 0, i64 32, i1 false)
 call fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal512_prefer256(<8 x i64>* %tmp2, <8 x
i64>* %tmp)
 %tmp4 = load <8 x i64>, <8 x i64>* %tmp2, align 32
 store <8 x i64> %tmp4, <8 x i64>* %arg, align 2
 ret void
}

; This should promote
define internal fastcc void @callee_avx512_legal512_prefer512_call_avx512_legal512_prefer256(<8 x i64>* %arg,
<8 x i64>* readonly %arg1) #1 {
;
; IS________OPM: Function Attrs: argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn
; IS________OPM-LABEL: define
{{[^@]+}}@callee_avx512_legal512_prefer512_call_avx512_legal512_prefer256
; IS________OPM-SAME: (<8 x i64>* nocapture nofree noundef nonnull writeonly align
 64 dereferenceable(64) [[ARG:%.*]], <8 x i64>* nocapture nofree noundef nonnull readonly align 64
dereferenceable(64) [[ARG1:%.*]]) #[[ATTR1]] {
; IS________OPM-NEXT:  bb:
; IS________OPM-NEXT:    [[TMP:%.*]] = load <8 x i64>, <8 x i64>* [[ARG1]], align 64
; IS________OPM-NEXT:    store <8 x i64> [[TMP]], <8 x i64>* [[ARG]], align 64
; IS________OPM-NEXT:    ret void
;
; IS__TUNIT_NPM: Function Attrs: argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn
; IS__TUNIT_NPM-LABEL: define
{{[^@]+}}@callee_avx512_legal512_prefer512_call_avx512_legal512_prefer256
; IS__TUNIT_NPM-SAME: (<8 x i64>* noalias nocapture nofree noundef nonnull writeonly align 64
dereferenceable(64) [[ARG:%.*]], <8 x i64> [[TMP0:%.*]]) #[[ATTR1]] {
; IS__TUNIT_NPM-NEXT:  bb:
; IS__TUNIT_NPM-NEXT:    [[ARG1_PRIV:%.*]] = alloca <8 x i64>, align 64
```

```
; IS__TUNIT_NPM-NEXT:    store <8 x i64> [[TMP0]], <8 x i64>* [[ARG1_PRIV]], align 64
; IS__TUNIT_NPM-NEXT:    [[TMP:%.*]] = load <8 x i64>, <8 x i64>*
 [[ARG1_PRIV]], align 64
; IS__TUNIT_NPM-NEXT:    store <8 x i64> [[TMP]], <8 x i64>* [[ARG]], align 64
; IS__TUNIT_NPM-NEXT:    ret void
;
; IS__CGSCC_NPM: Function Attrs: argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn
; IS__CGSCC_NPM-LABEL: define
{{[^@]+}}@callee_avx512_legal512_prefer512_call_avx512_legal512_prefer256
; IS__CGSCC_NPM-SAME: (<8 x i64>* noalias nocapture nofree noundef nonnull writeonly align 64
dereferenceable(64) [[ARG:%.*]], <8 x i64> [[TMP0:%.*]]) #[[ATTR1]] {
; IS__CGSCC_NPM-NEXT:  bb:
; IS__CGSCC_NPM-NEXT:    [[ARG1_PRIV:%.*]] = alloca <8 x i64>, align 64
; IS__CGSCC_NPM-NEXT:    store <8 x i64> [[TMP0]], <8 x i64>* [[ARG1_PRIV]], align 64
; IS__CGSCC_NPM-NEXT:    [[TMP:%.*]] = load <8 x i64>, <8 x i64>* [[ARG1_PRIV]], align 64
; IS__CGSCC_NPM-NEXT:    store <8 x i64> [[TMP0]], <8 x i64>* [[ARG]], align 64
; IS__CGSCC_NPM-NEXT:    ret void
;
bb:
 %tmp = load <8 x i64>, <8 x i64>* %arg1
 store <8 x i64> %tmp, <8 x i64>* %arg
 ret
 void
}

define void @avx512_legal512_prefer512_call_avx512_legal512_prefer256(<8 x i64>* %arg) #0 {
;
; IS__TUNIT_OPM: Function Attrs: argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn
; IS__TUNIT_OPM-LABEL: define {{[^@]+}}@avx512_legal512_prefer512_call_avx512_legal512_prefer256
; IS__TUNIT_OPM-SAME: (<8 x i64>* nocapture nofree writeonly [[ARG:%.*]]) #[[ATTR0]] {
; IS__TUNIT_OPM-NEXT:  bb:
; IS__TUNIT_OPM-NEXT:    [[TMP:%.*]] = alloca <8 x i64>, align 32
; IS__TUNIT_OPM-NEXT:    [[TMP2:%.*]] = alloca <8 x i64>, align 32
; IS__TUNIT_OPM-NEXT:    [[TMP3:%.*]] = bitcast <8 x i64>* [[TMP]] to i8*
; IS__TUNIT_OPM-NEXT:    call void @llvm.memset.p0i8.i64(i8* nocapture nofree noundef nonnull writeonly
align 64 dereferenceable(64) [[TMP3]], i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR6]]
; IS__TUNIT_OPM-NEXT:    call fastcc void
@callee_avx512_legal512_prefer512_call_avx512_legal512_prefer256(<8 x i64>* nocapture nofree noundef
nonnull writeonly align 64 dereferenceable(64)
 [[TMP2]], <8 x i64>* nocapture nofree noundef nonnull readonly align 64 dereferenceable(64) [[TMP]])
#[[ATTR7]]
; IS__TUNIT_OPM-NEXT:    [[TMP4:%.*]] = load <8 x i64>, <8 x i64>* [[TMP2]], align 64
; IS__TUNIT_OPM-NEXT:    store <8 x i64> [[TMP4]], <8 x i64>* [[ARG]], align 2
; IS__TUNIT_OPM-NEXT:    ret void
;
; IS__TUNIT_NPM: Function Attrs: argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn
; IS__TUNIT_NPM-LABEL: define {{[^@]+}}@avx512_legal512_prefer512_call_avx512_legal512_prefer256
; IS__TUNIT_NPM-SAME: (<8 x i64>* nocapture nofree writeonly [[ARG:%.*]]) #[[ATTR0]] {
```

```
; IS__TUNIT_NPM-NEXT:  bb:
; IS__TUNIT_NPM-NEXT:    [[TMP:%.*]] = alloca <8 x i64>, align 32
; IS__TUNIT_NPM-NEXT:    [[TMP2:%.*]] = alloca <8 x i64>, align 32
; IS__TUNIT_NPM-NEXT:    [[TMP3:%.*]] = bitcast <8 x i64>* [[TMP]] to i8*
; IS__TUNIT_NPM-NEXT:    call void @llvm.memset.p0i8.i64(i8* nocapture nofree noundef nonnull writeonly
align 64 dereferenceable(64) [[TMP3]],
 i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR6]]
; IS__TUNIT_NPM-NEXT:    [[TMP0:%.*]] = load <8 x i64>, <8 x i64>* [[TMP]], align 64
; IS__TUNIT_NPM-NEXT:    call fastcc void
@callee_avx512_legal512_prefer512_call_avx512_legal512_prefer256(<8 x i64>* noalias nocapture nofree
noundef nonnull writeonly align 64 dereferenceable(64) [[TMP2]], <8 x i64> [[TMP0]]) #[[ATTR7]]
; IS__TUNIT_NPM-NEXT:    [[TMP4:%.*]] = load <8 x i64>, <8 x i64>* [[TMP2]], align 64
; IS__TUNIT_NPM-NEXT:    store <8 x i64> [[TMP4]], <8 x i64>* [[ARG]], align 2
; IS__TUNIT_NPM-NEXT:    ret void
;
; IS__CGSCC_OPM: Function Attrs: argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn
; IS__CGSCC_OPM-LABEL: define {{[^@]+}}@avx512_legal512_prefer512_call_avx512_legal512_prefer256
; IS__CGSCC_OPM-SAME: (<8 x i64>* nocapture nofree noundef nonnull writeonly align 2 dereferenceable(64)
[[ARG:%.*]]) #[[ATTR0]] {
; IS__CGSCC_OPM-NEXT:  bb:
; IS__CGSCC_OPM-NEXT:    [[TMP:%.*]] = alloca <8 x i64>,
 align 32
; IS__CGSCC_OPM-NEXT:    [[TMP2:%.*]] = alloca <8 x i64>, align 32
; IS__CGSCC_OPM-NEXT:    [[TMP3:%.*]] = bitcast <8 x i64>* [[TMP]] to i8*
; IS__CGSCC_OPM-NEXT:    call void @llvm.memset.p0i8.i64(i8* nocapture nofree noundef nonnull writeonly
align 64 dereferenceable(64) [[TMP3]], i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR6]]
; IS__CGSCC_OPM-NEXT:    call fastcc void
@callee_avx512_legal512_prefer512_call_avx512_legal512_prefer256(<8 x i64>* nocapture nofree noundef
nonnull writeonly align 64 dereferenceable(64) [[TMP2]], <8 x i64>* nocapture nofree noundef nonnull readonly
align 64 dereferenceable(64) [[TMP]]) #[[ATTR7]]
; IS__CGSCC_OPM-NEXT:    [[TMP4:%.*]] = load <8 x i64>, <8 x i64>* [[TMP2]], align 64
; IS__CGSCC_OPM-NEXT:    store <8 x i64> [[TMP4]], <8 x i64>* [[ARG]], align 2
; IS__CGSCC_OPM-NEXT:    ret void
;
; IS__CGSCC_NPM: Function Attrs: argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn
; IS__CGSCC_NPM-LABEL: define {{[^@]+}}@avx512_legal512_prefer512_call_avx512_legal512_prefer256
;
 IS__CGSCC_NPM-SAME: (<8 x i64>* nocapture nofree noundef nonnull writeonly align 2 dereferenceable(64)
[[ARG:%.*]]) #[[ATTR0]] {
; IS__CGSCC_NPM-NEXT:  bb:
; IS__CGSCC_NPM-NEXT:    [[TMP:%.*]] = alloca <8 x i64>, align 32
; IS__CGSCC_NPM-NEXT:    [[TMP2:%.*]] = alloca <8 x i64>, align 32
; IS__CGSCC_NPM-NEXT:    [[TMP3:%.*]] = bitcast <8 x i64>* [[TMP]] to i8*
; IS__CGSCC_NPM-NEXT:    call void @llvm.memset.p0i8.i64(i8* nocapture nofree noundef nonnull writeonly
align 64 dereferenceable(64) [[TMP3]], i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR6]]
; IS__CGSCC_NPM-NEXT:    [[TMP0:%.*]] = load <8 x i64>, <8 x i64>* [[TMP]], align 64
; IS__CGSCC_NPM-NEXT:    call fastcc void
@callee_avx512_legal512_prefer512_call_avx512_legal512_prefer256(<8 x i64>* noalias nocapture nofree
```

```
noundef nonnull writeonly align 64 dereferenceable(64) [[TMP2]], <8 x i64> [[TMP0]]) #[[ATTR7]]
; IS__CGSCC_NPM-NEXT:    [[TMP4:%.*]] = load <8
 x i64>, <8 x i64>* [[TMP2]], align 64
; IS__CGSCC_NPM-NEXT:    store <8 x i64> [[TMP4]], <8 x i64>* [[ARG]], align 2
; IS__CGSCC_NPM-NEXT:    ret void
;
bb:
 %tmp = alloca <8 x i64>, align 32
 %tmp2 = alloca <8 x i64>, align 32
 %tmp3 = bitcast <8 x i64>* %tmp to i8*
 call void @llvm.memset.p0i8.i64(i8* align 32 %tmp3, i8 0, i64 32, i1 false)
 call fastcc void @callee_avx512_legal512_prefer512_call_avx512_legal512_prefer256(<8 x i64>* %tmp2, <8 x
i64>* %tmp)
 %tmp4 = load <8 x i64>, <8 x i64>* %tmp2, align 32
 store <8 x i64> %tmp4, <8 x i64>* %arg, align 2
 ret void
}

; This should promote
define internal fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal512_prefer512(<8 x i64>* %arg,
<8 x i64>* readonly %arg1) #0 {
;
; IS________OPM: Function Attrs: argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn
; IS________OPM-LABEL: define
{{[^@]+}}@callee_avx512_legal512_prefer256_call_avx512_legal512_prefer512
; IS________OPM-SAME: (<8 x i64>* nocapture
 nofree noundef nonnull writeonly align 64 dereferenceable(64) [[ARG:%.*]], <8 x i64>* nocapture nofree noundef
nonnull readonly align 64 dereferenceable(64) [[ARG1:%.*]]) #[[ATTR0]] {
; IS________OPM-NEXT:  bb:
; IS________OPM-NEXT:    [[TMP:%.*]] = load <8 x i64>, <8 x i64>* [[ARG1]], align 64
; IS________OPM-NEXT:    store <8 x i64> [[TMP]], <8 x i64>* [[ARG]], align 64
; IS________OPM-NEXT:    ret void
;
; IS__TUNIT_NPM: Function Attrs: argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn
; IS__TUNIT_NPM-LABEL: define
{{[^@]+}}@callee_avx512_legal512_prefer256_call_avx512_legal512_prefer512
; IS__TUNIT_NPM-SAME: (<8 x i64>* noalias nocapture nofree noundef nonnull writeonly align 64
dereferenceable(64) [[ARG:%.*]], <8 x i64> [[TMP0:%.*]]) #[[ATTR0]] {
; IS__TUNIT_NPM-NEXT:  bb:
; IS__TUNIT_NPM-NEXT:    [[ARG1_PRIV:%.*]] = alloca <8 x i64>, align 64
; IS__TUNIT_NPM-NEXT:    store <8 x i64> [[TMP0]], <8 x i64>* [[ARG1_PRIV]], align 64
; IS__TUNIT_NPM-NEXT:
   [[TMP:%.*]] = load <8 x i64>, <8 x i64>* [[ARG1_PRIV]], align 64
; IS__TUNIT_NPM-NEXT:    store <8 x i64> [[TMP]], <8 x i64>* [[ARG]], align 64
; IS__TUNIT_NPM-NEXT:    ret void
;
; IS__CGSCC_NPM: Function Attrs: argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn
; IS__CGSCC_NPM-LABEL: define
```

```
{{[^@]+}}@callee_avx512_legal512_prefer256_call_avx512_legal512_prefer512
; IS__CGSCC_NPM-SAME: (<8 x i64>* noalias nocapture nofree noundef nonnull writeonly align 64
dereferenceable(64) [[ARG:%.*]], <8 x i64> [[TMP0:%.*]]) #[[ATTR0]] {
; IS__CGSCC_NPM-NEXT:  bb:
; IS__CGSCC_NPM-NEXT:    [[ARG1_PRIV:%.*]] = alloca <8 x i64>, align 64
; IS__CGSCC_NPM-NEXT:    store <8 x i64> [[TMP0]], <8 x i64>* [[ARG1_PRIV]], align 64
; IS__CGSCC_NPM-NEXT:    [[TMP:%.*]] = load <8 x i64>, <8 x i64>* [[ARG1_PRIV]], align 64
; IS__CGSCC_NPM-NEXT:    store <8 x i64> [[TMP0]], <8 x i64>* [[ARG]], align 64
; IS__CGSCC_NPM-NEXT:    ret void
;
bb:
 %tmp = load <8 x i64>, <8 x i64>* %arg1
 store
 <8 x i64> %tmp, <8 x i64>* %arg
 ret void
}

define void @avx512_legal512_prefer256_call_avx512_legal512_prefer512(<8 x i64>* %arg) #1 {
;
; IS__TUNIT_OPM: Function Attrs: argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn
; IS__TUNIT_OPM-LABEL: define {{[^@]+}}@avx512_legal512_prefer256_call_avx512_legal512_prefer512
; IS__TUNIT_OPM-SAME: (<8 x i64>* nocapture nofree writeonly [[ARG:%.*]]) #[[ATTR1]] {
; IS__TUNIT_OPM-NEXT:  bb:
; IS__TUNIT_OPM-NEXT:    [[TMP:%.*]] = alloca <8 x i64>, align 32
; IS__TUNIT_OPM-NEXT:    [[TMP2:%.*]] = alloca <8 x i64>, align 32
; IS__TUNIT_OPM-NEXT:    [[TMP3:%.*]] = bitcast <8 x i64>* [[TMP]] to i8*
; IS__TUNIT_OPM-NEXT:    call void @llvm.memset.p0i8.i64(i8* nocapture nofree noundef nonnull writeonly
align 64 dereferenceable(64) [[TMP3]], i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR6]]
; IS__TUNIT_OPM-NEXT:    call fastcc void
@callee_avx512_legal512_prefer256_call_avx512_legal512_prefer512(<8 x i64>* nocapture nofree
 noundef nonnull writeonly align 64 dereferenceable(64) [[TMP2]], <8 x i64>* nocapture nofree noundef nonnull
readonly align 64 dereferenceable(64) [[TMP]]) #[[ATTR7]]
; IS__TUNIT_OPM-NEXT:    [[TMP4:%.*]] = load <8 x i64>, <8 x i64>* [[TMP2]], align 64
; IS__TUNIT_OPM-NEXT:    store <8 x i64> [[TMP4]], <8 x i64>* [[ARG]], align 2
; IS__TUNIT_OPM-NEXT:    ret void
;
; IS__TUNIT_NPM: Function Attrs: argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn
; IS__TUNIT_NPM-LABEL: define {{[^@]+}}@avx512_legal512_prefer256_call_avx512_legal512_prefer512
; IS__TUNIT_NPM-SAME: (<8 x i64>* nocapture nofree writeonly [[ARG:%.*]]) #[[ATTR1]] {
; IS__TUNIT_NPM-NEXT:  bb:
; IS__TUNIT_NPM-NEXT:    [[TMP:%.*]] = alloca <8 x i64>, align 32
; IS__TUNIT_NPM-NEXT:    [[TMP2:%.*]] = alloca <8 x i64>, align 32
; IS__TUNIT_NPM-NEXT:    [[TMP3:%.*]] = bitcast <8 x i64>* [[TMP]] to i8*
; IS__TUNIT_NPM-NEXT:    call void @llvm.memset.p0i8.i64(i8* nocapture nofree noundef nonnull writeonly
 align 64 dereferenceable(64) [[TMP3]], i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR6]]
; IS__TUNIT_NPM-NEXT:    [[TMP0:%.*]] = load <8 x i64>, <8 x i64>* [[TMP]], align 64
; IS__TUNIT_NPM-NEXT:    call fastcc void
@callee_avx512_legal512_prefer256_call_avx512_legal512_prefer512(<8 x i64>* noalias nocapture nofree
```

```
noundef nonnull writeonly align 64 dereferenceable(64) [[TMP2]], <8 x i64> [[TMP0]]) #[[ATTR7]]
; IS__TUNIT_NPM-NEXT:    [[TMP4:%.*]] = load <8 x i64>, <8 x i64>* [[TMP2]], align 64
; IS__TUNIT_NPM-NEXT:    store <8 x i64> [[TMP4]], <8 x i64>* [[ARG]], align 2
; IS__TUNIT_NPM-NEXT:    ret void
;
; IS__CGSCC_OPM: Function Attrs: argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn
; IS__CGSCC_OPM-LABEL: define {{[^@]+}}@avx512_legal512_prefer256_call_avx512_legal512_prefer512
; IS__CGSCC_OPM-SAME: (<8 x i64>* nocapture nofree noundef nonnull writeonly align 2 dereferenceable(64)
[[ARG:%.*]]) #[[ATTR1]] {
; IS__CGSCC_OPM-NEXT:  bb:
; IS__CGSCC_OPM-NEXT:
   [[TMP:%.*]] = alloca <8 x i64>, align 32
; IS__CGSCC_OPM-NEXT:    [[TMP2:%.*]] = alloca <8 x i64>, align 32
; IS__CGSCC_OPM-NEXT:    [[TMP3:%.*]] = bitcast <8 x i64>* [[TMP]] to i8*
; IS__CGSCC_OPM-NEXT:    call void @llvm.memset.p0i8.i64(i8* nocapture nofree noundef nonnull writeonly
align 64 dereferenceable(64) [[TMP3]], i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR6]]
; IS__CGSCC_OPM-NEXT:    call fastcc void
@callee_avx512_legal512_prefer256_call_avx512_legal512_prefer512(<8 x i64>* nocapture nofree noundef
nonnull writeonly align 64 dereferenceable(64) [[TMP2]], <8 x i64>* nocapture nofree noundef nonnull readonly
align 64 dereferenceable(64) [[TMP]]) #[[ATTR7]]
; IS__CGSCC_OPM-NEXT:    [[TMP4:%.*]] = load <8 x i64>, <8 x i64>* [[TMP2]], align 64
; IS__CGSCC_OPM-NEXT:    store <8 x i64> [[TMP4]], <8 x i64>* [[ARG]], align 2
; IS__CGSCC_OPM-NEXT:    ret void
;
; IS__CGSCC_NPM: Function Attrs: argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn
;
 IS__CGSCC_NPM-LABEL: define {{[^@]+}}@avx512_legal512_prefer256_call_avx512_legal512_prefer512
; IS__CGSCC_NPM-SAME: (<8 x i64>* nocapture nofree noundef nonnull writeonly align 2 dereferenceable(64)
[[ARG:%.*]]) #[[ATTR1]] {
; IS__CGSCC_NPM-NEXT:  bb:
; IS__CGSCC_NPM-NEXT:    [[TMP:%.*]] = alloca <8 x i64>, align 32
; IS__CGSCC_NPM-NEXT:    [[TMP2:%.*]] = alloca <8 x i64>, align 32
; IS__CGSCC_NPM-NEXT:    [[TMP3:%.*]] = bitcast <8 x i64>* [[TMP]] to i8*
; IS__CGSCC_NPM-NEXT:    call void @llvm.memset.p0i8.i64(i8* nocapture nofree noundef nonnull writeonly
align 64 dereferenceable(64) [[TMP3]], i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR6]]
; IS__CGSCC_NPM-NEXT:    [[TMP0:%.*]] = load <8 x i64>, <8 x i64>* [[TMP]], align 64
; IS__CGSCC_NPM-NEXT:    call fastcc void
@callee_avx512_legal512_prefer256_call_avx512_legal512_prefer512(<8 x i64>* noalias nocapture nofree
noundef nonnull writeonly align 64 dereferenceable(64) [[TMP2]], <8 x i64> [[TMP0]]) #[[ATTR7]]
; IS__CGSCC_NPM-NEXT:
   [[TMP4:%.*]] = load <8 x i64>, <8 x i64>* [[TMP2]], align 64
; IS__CGSCC_NPM-NEXT:    store <8 x i64> [[TMP4]], <8 x i64>* [[ARG]], align 2
; IS__CGSCC_NPM-NEXT:    ret void
;
bb:
 %tmp = alloca <8 x i64>, align 32
 %tmp2 = alloca <8 x i64>, align 32
 %tmp3 = bitcast <8 x i64>* %tmp to i8*
```

```
 call void @llvm.memset.p0i8.i64(i8* align 32 %tmp3, i8 0, i64 32, i1 false)
 call fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal512_prefer512(<8 x i64>* %tmp2, <8 x
i64>* %tmp)
 %tmp4 = load <8 x i64>, <8 x i64>* %tmp2, align 32
 store <8 x i64> %tmp4, <8 x i64>* %arg, align 2
 ret void
}

; This should not promote
define internal fastcc void @callee_avx512_legal256_prefer256_call_avx512_legal512_prefer256(<8 x i64>* %arg,
<8 x i64>* readonly %arg1) #1 {
;
; IS________OPM: Function Attrs: argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn
; IS________OPM-LABEL: define
{{[^@]+}}@callee_avx512_legal256_prefer256_call_avx512_legal512_prefer256
;
 IS________OPM-SAME: (<8 x i64>* nocapture nofree noundef nonnull writeonly align 64 dereferenceable(64)
[[ARG:%.*]], <8 x i64>* nocapture nofree noundef nonnull readonly align 64 dereferenceable(64) [[ARG1:%.*]])
#[[ATTR1]] {
; IS________OPM-NEXT:  bb:
; IS________OPM-NEXT:    [[TMP:%.*]] = load <8 x i64>, <8 x i64>* [[ARG1]], align 64
; IS________OPM-NEXT:    store <8 x i64> [[TMP]], <8 x i64>* [[ARG]], align 64
; IS________OPM-NEXT:    ret void
;
; IS________NPM: Function Attrs: argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn
; IS________NPM-LABEL: define
{{[^@]+}}@callee_avx512_legal256_prefer256_call_avx512_legal512_prefer256
; IS________NPM-SAME: (<8 x i64>* noalias nocapture nofree noundef nonnull writeonly align 64
dereferenceable(64) [[ARG:%.*]], <8 x i64>* noalias nocapture nofree noundef nonnull readonly align 64
dereferenceable(64) [[ARG1:%.*]]) #[[ATTR1:[0-9]+]] {
; IS________NPM-NEXT:  bb:
; IS________NPM-NEXT:    [[TMP:%.*]] = load <8 x
 i64>, <8 x i64>* [[ARG1]], align 64
; IS________NPM-NEXT:    store <8 x i64> [[TMP]], <8 x i64>* [[ARG]], align 64
; IS________NPM-NEXT:    ret void
;
bb:
 %tmp = load <8 x i64>, <8 x i64>* %arg1
 store <8 x i64> %tmp, <8 x i64>* %arg
 ret void
}

define void @avx512_legal256_prefer256_call_avx512_legal512_prefer256(<8 x i64>* %arg) #2 {
;
; IS__TUNIT_OPM: Function Attrs: argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn
; IS__TUNIT_OPM-LABEL: define {{[^@]+}}@avx512_legal256_prefer256_call_avx512_legal512_prefer256
; IS__TUNIT_OPM-SAME: (<8 x i64>* nocapture nofree writeonly [[ARG:%.*]]) #[[ATTR2:[0-9]+]] {
; IS__TUNIT_OPM-NEXT:  bb:
```

; IS__TUNIT_OPM-NEXT: [[TMP:%.*]] = alloca <8 x i64>, align 32
; IS__TUNIT_OPM-NEXT: [[TMP2:%.*]] = alloca <8 x i64>, align 32
; IS__TUNIT_OPM-NEXT: [[TMP3:%.*]] = bitcast <8 x i64>* [[TMP]] to i8*
; IS__TUNIT_OPM-NEXT: call void @llvm.memset.p0i8.i64(i8* nocapture nofree noundef nonnull writeonly align 64 dereferenceable(64)
[[TMP3]], i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR6]]
; IS__TUNIT_OPM-NEXT: call fastcc void @callee_avx512_legal256_prefer256_call_avx512_legal512_prefer256(<8 x i64>* nocapture nofree noundef nonnull writeonly align 64 dereferenceable(64) [[TMP2]], <8 x i64>* nocapture nofree noundef nonnull readonly align 64 dereferenceable(64) [[TMP]]) #[[ATTR7]]
; IS__TUNIT_OPM-NEXT: [[TMP4:%.*]] = load <8 x i64>, <8 x i64>* [[TMP2]], align 64
; IS__TUNIT_OPM-NEXT: store <8 x i64> [[TMP4]], <8 x i64>* [[ARG]], align 2
; IS__TUNIT_OPM-NEXT: ret void
;
; IS__TUNIT_NPM: Function Attrs: argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn
; IS__TUNIT_NPM-LABEL: define {{[^@]+}}@avx512_legal256_prefer256_call_avx512_legal512_prefer256
; IS__TUNIT_NPM-SAME: (<8 x i64>* nocapture nofree writeonly [[ARG:%.*]]) #[[ATTR2:[0-9]+]] {
; IS__TUNIT_NPM-NEXT: bb:
; IS__TUNIT_NPM-NEXT: [[TMP:%.*]] = alloca <8 x i64>, align 32
; IS__TUNIT_NPM-NEXT: [[TMP2:%.*]]
= alloca <8 x i64>, align 32
; IS__TUNIT_NPM-NEXT: [[TMP3:%.*]] = bitcast <8 x i64>* [[TMP]] to i8*
; IS__TUNIT_NPM-NEXT: call void @llvm.memset.p0i8.i64(i8* nocapture nofree noundef nonnull writeonly align 64 dereferenceable(64) [[TMP3]], i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR6]]
; IS__TUNIT_NPM-NEXT: call fastcc void @callee_avx512_legal256_prefer256_call_avx512_legal512_prefer256(<8 x i64>* noalias nocapture nofree noundef nonnull writeonly align 64 dereferenceable(64) [[TMP2]], <8 x i64>* noalias nocapture nofree noundef nonnull readonly align 64 dereferenceable(64) [[TMP]]) #[[ATTR7]]
; IS__TUNIT_NPM-NEXT: [[TMP4:%.*]] = load <8 x i64>, <8 x i64>* [[TMP2]], align 64
; IS__TUNIT_NPM-NEXT: store <8 x i64> [[TMP4]], <8 x i64>* [[ARG]], align 2
; IS__TUNIT_NPM-NEXT: ret void
;
; IS__CGSCC_OPM: Function Attrs: argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn
; IS__CGSCC_OPM-LABEL: define {{[^@]+}}@avx512_legal256_prefer256_call_avx512_legal512_prefer256
;
IS__CGSCC_OPM-SAME: (<8 x i64>* nocapture nofree noundef nonnull writeonly align 2 dereferenceable(64) [[ARG:%.*]]) #[[ATTR2:[0-9]+]] {
; IS__CGSCC_OPM-NEXT: bb:
; IS__CGSCC_OPM-NEXT: [[TMP:%.*]] = alloca <8 x i64>, align 32
; IS__CGSCC_OPM-NEXT: [[TMP2:%.*]] = alloca <8 x i64>, align 32
; IS__CGSCC_OPM-NEXT: [[TMP3:%.*]] = bitcast <8 x i64>* [[TMP]] to i8*
; IS__CGSCC_OPM-NEXT: call void @llvm.memset.p0i8.i64(i8* nocapture nofree noundef nonnull writeonly align 64 dereferenceable(64) [[TMP3]], i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR6]]
; IS__CGSCC_OPM-NEXT: call fastcc void @callee_avx512_legal256_prefer256_call_avx512_legal512_prefer256(<8 x i64>* nocapture nofree noundef nonnull writeonly align 64 dereferenceable(64) [[TMP2]], <8 x i64>* nocapture nofree noundef nonnull readonly align 64 dereferenceable(64) [[TMP]]) #[[ATTR7]]
; IS__CGSCC_OPM-NEXT: [[TMP4:%.*]] = load <8 x i64>, <8 x i64>* [[TMP2]], align 64

```
;
 IS__CGSCC_OPM-NEXT:    store <8 x i64> [[TMP4]], <8 x i64>* [[ARG]], align 2
; IS__CGSCC_OPM-NEXT:    ret void
;
; IS__CGSCC_NPM: Function Attrs: argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn
; IS__CGSCC_NPM-LABEL: define {{[^@]+}}@avx512_legal256_prefer256_call_avx512_legal512_prefer256
; IS__CGSCC_NPM-SAME: (<8 x i64>* nocapture nofree noundef nonnull writeonly align 2 dereferenceable(64)
[[ARG:%.*]]) #[[ATTR2:[0-9]+]] {
; IS__CGSCC_NPM-NEXT:  bb:
; IS__CGSCC_NPM-NEXT:    [[TMP:%.*]] = alloca <8 x i64>, align 32
; IS__CGSCC_NPM-NEXT:    [[TMP2:%.*]] = alloca <8 x i64>, align 32
; IS__CGSCC_NPM-NEXT:    [[TMP3:%.*]] = bitcast <8 x i64>* [[TMP]] to i8*
; IS__CGSCC_NPM-NEXT:    call void @llvm.memset.p0i8.i64(i8* nocapture nofree noundef nonnull writeonly
align 64 dereferenceable(64) [[TMP3]], i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR6]]
; IS__CGSCC_NPM-NEXT:    call fastcc void
@callee_avx512_legal256_prefer256_call_avx512_legal512_prefer256(<8
 x i64>* noalias nocapture nofree noundef nonnull writeonly align 64 dereferenceable(64) [[TMP2]], <8 x i64>*
noalias nocapture nofree noundef nonnull readonly align 64 dereferenceable(64) [[TMP]]) #[[ATTR7]]
; IS__CGSCC_NPM-NEXT:    [[TMP4:%.*]] = load <8 x i64>, <8 x i64>* [[TMP2]], align 64
; IS__CGSCC_NPM-NEXT:    store <8 x i64> [[TMP4]], <8 x i64>* [[ARG]], align 2
; IS__CGSCC_NPM-NEXT:    ret void
;
bb:
 %tmp = alloca <8 x i64>, align 32
 %tmp2 = alloca <8 x i64>, align 32
 %tmp3 = bitcast <8 x i64>* %tmp to i8*
 call void @llvm.memset.p0i8.i64(i8* align 32 %tmp3, i8 0, i64 32, i1 false)
 call fastcc void @callee_avx512_legal256_prefer256_call_avx512_legal512_prefer256(<8 x i64>* %tmp2, <8 x
i64>* %tmp)
 %tmp4 = load <8 x i64>, <8 x i64>* %tmp2, align 32
 store <8 x i64> %tmp4, <8 x i64>* %arg, align 2
 ret void
}

; This should not promote
define internal fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal256_prefer256(<8 x i64>* %arg,
<8 x i64>* readonly
 %arg1) #2 {
;
; IS________OPM: Function Attrs: argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn
; IS________OPM-LABEL: define
{{[^@]+}}@callee_avx512_legal512_prefer256_call_avx512_legal256_prefer256
; IS________OPM-SAME: (<8 x i64>* nocapture nofree noundef nonnull writeonly align 64 dereferenceable(64)
[[ARG:%.*]], <8 x i64>* nocapture nofree noundef nonnull readonly align 64 dereferenceable(64) [[ARG1:%.*]])
#[[ATTR2:[0-9]+]] {
; IS________OPM-NEXT:  bb:
; IS________OPM-NEXT:    [[TMP:%.*]] = load <8 x i64>, <8 x i64>* [[ARG1]], align 64
; IS________OPM-NEXT:    store <8 x i64> [[TMP]], <8 x i64>* [[ARG]], align 64
```

```
; IS________OPM-NEXT:    ret void
;
; IS________NPM: Function Attrs: argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn
; IS________NPM-LABEL: define
{{[^@]+}}@callee_avx512_legal512_prefer256_call_avx512_legal256_prefer256
; IS________NPM-SAME: (<8 x i64>* noalias nocapture nofree noundef nonnull writeonly align 64
dereferenceable(64)
 [[ARG:%.*]], <8 x i64>* noalias nocapture nofree noundef nonnull readonly align 64 dereferenceable(64)
[[ARG1:%.*]]) #[[ATTR2:[0-9]+]] {
; IS________NPM-NEXT:  bb:
; IS________NPM-NEXT:    [[TMP:%.*]] = load <8 x i64>, <8 x i64>* [[ARG1]], align 64
; IS________NPM-NEXT:    store <8 x i64> [[TMP]], <8 x i64>* [[ARG]], align 64
; IS________NPM-NEXT:    ret void
;
bb:
 %tmp = load <8 x i64>, <8 x i64>* %arg1
 store <8 x i64> %tmp, <8 x i64>* %arg
 ret void
}

define void @avx512_legal512_prefer256_call_avx512_legal256_prefer256(<8 x i64>* %arg) #1 {
;
; IS__TUNIT_OPM: Function Attrs: argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn
; IS__TUNIT_OPM-LABEL: define {{[^@]+}}@avx512_legal512_prefer256_call_avx512_legal256_prefer256
; IS__TUNIT_OPM-SAME: (<8 x i64>* nocapture nofree writeonly [[ARG:%.*]]) #[[ATTR1]] {
; IS__TUNIT_OPM-NEXT:  bb:
; IS__TUNIT_OPM-NEXT:    [[TMP:%.*]] = alloca <8 x i64>, align 32
; IS__TUNIT_OPM-NEXT:    [[TMP2:%.*]] = alloca
 <8 x i64>, align 32
; IS__TUNIT_OPM-NEXT:    [[TMP3:%.*]] = bitcast <8 x i64>* [[TMP]] to i8*
; IS__TUNIT_OPM-NEXT:    call void @llvm.memset.p0i8.i64(i8* nocapture nofree noundef nonnull writeonly
align 64 dereferenceable(64) [[TMP3]], i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR6]]
; IS__TUNIT_OPM-NEXT:    call fastcc void
@callee_avx512_legal512_prefer256_call_avx512_legal256_prefer256(<8 x i64>* nocapture nofree noundef
nonnull writeonly align 64 dereferenceable(64) [[TMP2]], <8 x i64>* nocapture nofree noundef nonnull readonly
align 64 dereferenceable(64) [[TMP]]) #[[ATTR7]]
; IS__TUNIT_OPM-NEXT:    [[TMP4:%.*]] = load <8 x i64>, <8 x i64>* [[TMP2]], align 64
; IS__TUNIT_OPM-NEXT:    store <8 x i64> [[TMP4]], <8 x i64>* [[ARG]], align 2
; IS__TUNIT_OPM-NEXT:    ret void
;
; IS__TUNIT_NPM: Function Attrs: argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn
; IS__TUNIT_NPM-LABEL: define {{[^@]+}}@avx512_legal512_prefer256_call_avx512_legal256_prefer256
;
 IS__TUNIT_NPM-SAME: (<8 x i64>* nocapture nofree writeonly [[ARG:%.*]]) #[[ATTR1]] {
; IS__TUNIT_NPM-NEXT:  bb:
; IS__TUNIT_NPM-NEXT:    [[TMP:%.*]] = alloca <8 x i64>, align 32
; IS__TUNIT_NPM-NEXT:    [[TMP2:%.*]] = alloca <8 x i64>, align 32
; IS__TUNIT_NPM-NEXT:    [[TMP3:%.*]] = bitcast <8 x i64>* [[TMP]] to i8*
```

```
; IS__TUNIT_NPM-NEXT:    call void @llvm.memset.p0i8.i64(i8* nocapture nofree noundef nonnull writeonly
align 64 dereferenceable(64) [[TMP3]], i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR6]]
; IS__TUNIT_NPM-NEXT:    call fastcc void
@callee_avx512_legal512_prefer256_call_avx512_legal256_prefer256(<8 x i64>* noalias nocapture nofree
noundef nonnull writeonly align 64 dereferenceable(64) [[TMP2]], <8 x i64>* noalias nocapture nofree noundef
nonnull readonly align 64 dereferenceable(64) [[TMP]]) #[[ATTR7]]
; IS__TUNIT_NPM-NEXT:    [[TMP4:%.*]] = load <8 x i64>, <8 x i64>* [[TMP2]], align 64
; IS__TUNIT_NPM-NEXT:    store <8 x i64> [[TMP4]], <8 x i64>* [[ARG]],
 align 2
; IS__TUNIT_NPM-NEXT:    ret void
;
; IS__CGSCC_OPM: Function Attrs: argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn
; IS__CGSCC_OPM-LABEL: define {{[^@]+}}@avx512_legal512_prefer256_call_avx512_legal256_prefer256
; IS__CGSCC_OPM-SAME: (<8 x i64>* nocapture nofree noundef nonnull writeonly align 2 dereferenceable(64)
[[ARG:%.*]]) #[[ATTR1]] {
; IS__CGSCC_OPM-NEXT:  bb:
; IS__CGSCC_OPM-NEXT:    [[TMP:%.*]] = alloca <8 x i64>, align 32
; IS__CGSCC_OPM-NEXT:    [[TMP2:%.*]] = alloca <8 x i64>, align 32
; IS__CGSCC_OPM-NEXT:    [[TMP3:%.*]] = bitcast <8 x i64>* [[TMP]] to i8*
; IS__CGSCC_OPM-NEXT:    call void @llvm.memset.p0i8.i64(i8* nocapture nofree noundef nonnull writeonly
align 64 dereferenceable(64) [[TMP3]], i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR6]]
; IS__CGSCC_OPM-NEXT:    call fastcc void
@callee_avx512_legal512_prefer256_call_avx512_legal256_prefer256(<8 x i64>* nocapture nofree noundef
nonnull writeonly align 64 dereferenceable(64)
 [[TMP2]], <8 x i64>* nocapture nofree noundef nonnull readonly align 64 dereferenceable(64) [[TMP]])
#[[ATTR7]]
; IS__CGSCC_OPM-NEXT:    [[TMP4:%.*]] = load <8 x i64>, <8 x i64>* [[TMP2]], align 64
; IS__CGSCC_OPM-NEXT:    store <8 x i64> [[TMP4]], <8 x i64>* [[ARG]], align 2
; IS__CGSCC_OPM-NEXT:    ret void
;
; IS__CGSCC_NPM: Function Attrs: argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn
; IS__CGSCC_NPM-LABEL: define {{[^@]+}}@avx512_legal512_prefer256_call_avx512_legal256_prefer256
; IS__CGSCC_NPM-SAME: (<8 x i64>* nocapture nofree noundef nonnull writeonly align 2 dereferenceable(64)
[[ARG:%.*]]) #[[ATTR1]] {
; IS__CGSCC_NPM-NEXT:  bb:
; IS__CGSCC_NPM-NEXT:    [[TMP:%.*]] = alloca <8 x i64>, align 32
; IS__CGSCC_NPM-NEXT:    [[TMP2:%.*]] = alloca <8 x i64>, align 32
; IS__CGSCC_NPM-NEXT:    [[TMP3:%.*]] = bitcast <8 x i64>* [[TMP]] to i8*
; IS__CGSCC_NPM-NEXT:    call void @llvm.memset.p0i8.i64(i8* nocapture nofree noundef nonnull writeonly
align
 64 dereferenceable(64) [[TMP3]], i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR6]]
; IS__CGSCC_NPM-NEXT:    call fastcc void
@callee_avx512_legal512_prefer256_call_avx512_legal256_prefer256(<8 x i64>* noalias nocapture nofree
noundef nonnull writeonly align 64 dereferenceable(64) [[TMP2]], <8 x i64>* noalias nocapture nofree noundef
nonnull readonly align 64 dereferenceable(64) [[TMP]]) #[[ATTR7]]
; IS__CGSCC_NPM-NEXT:    [[TMP4:%.*]] = load <8 x i64>, <8 x i64>* [[TMP2]], align 64
; IS__CGSCC_NPM-NEXT:    store <8 x i64> [[TMP4]], <8 x i64>* [[ARG]], align 2
; IS__CGSCC_NPM-NEXT:    ret void
```

```
;
bb:
 %tmp = alloca <8 x i64>, align 32
 %tmp2 = alloca <8 x i64>, align 32
 %tmp3 = bitcast <8 x i64>* %tmp to i8*
 call void @llvm.memset.p0i8.i64(i8* align 32 %tmp3, i8 0, i64 32, i1 false)
 call fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal256_prefer256(<8 x i64>* %tmp2, <8 x
i64>* %tmp)
 %tmp4 = load <8 x i64>, <8 x i64>* %tmp2, align 32
 store <8 x i64> %tmp4,
 <8 x i64>* %arg, align 2
 ret void
}

; This should promote
define internal fastcc void @callee_avx2_legal256_prefer256_call_avx2_legal512_prefer256(<8 x i64>* %arg, <8
x i64>* readonly %arg1) #3 {
;
; IS________OPM: Function Attrs: argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn
; IS________OPM-LABEL: define {{[^@]+}}@callee_avx2_legal256_prefer256_call_avx2_legal512_prefer256
; IS________OPM-SAME: (<8 x i64>* nocapture nofree noundef nonnull writeonly align 64 dereferenceable(64)
[[ARG:%.*]], <8 x i64>* nocapture nofree noundef nonnull readonly align 64 dereferenceable(64) [[ARG1:%.*]])
#[[ATTR3:[0-9]+]] {
; IS________OPM-NEXT:  bb:
; IS________OPM-NEXT:    [[TMP:%.*]] = load <8 x i64>, <8 x i64>* [[ARG1]], align 64
; IS________OPM-NEXT:    store <8 x i64> [[TMP]], <8 x i64>* [[ARG]], align 64
; IS________OPM-NEXT:    ret void
;
; IS__TUNIT_NPM: Function Attrs: argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn
; IS__TUNIT_NPM-LABEL:
 define {{[^@]+}}@callee_avx2_legal256_prefer256_call_avx2_legal512_prefer256
; IS__TUNIT_NPM-SAME: (<8 x i64>* noalias nocapture nofree noundef nonnull writeonly align 64
dereferenceable(64) [[ARG:%.*]], <8 x i64> [[TMP0:%.*]]) #[[ATTR3:[0-9]+]] {
; IS__TUNIT_NPM-NEXT:  bb:
; IS__TUNIT_NPM-NEXT:    [[ARG1_PRIV:%.*]] = alloca <8 x i64>, align 64
; IS__TUNIT_NPM-NEXT:    store <8 x i64> [[TMP0]], <8 x i64>* [[ARG1_PRIV]], align 64
; IS__TUNIT_NPM-NEXT:    [[TMP:%.*]] = load <8 x i64>, <8 x i64>* [[ARG1_PRIV]], align 64
; IS__TUNIT_NPM-NEXT:    store <8 x i64> [[TMP]], <8 x i64>* [[ARG]], align 64
; IS__TUNIT_NPM-NEXT:    ret void
;
; IS__CGSCC_NPM: Function Attrs: argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn
; IS__CGSCC_NPM-LABEL: define {{[^@]+}}@callee_avx2_legal256_prefer256_call_avx2_legal512_prefer256
; IS__CGSCC_NPM-SAME: (<8 x i64>* noalias nocapture nofree noundef nonnull writeonly align 64
dereferenceable(64) [[ARG:%.*]], <8 x i64> [[TMP0:%.*]])
 #[[ATTR3:[0-9]+]] {
; IS__CGSCC_NPM-NEXT:  bb:
; IS__CGSCC_NPM-NEXT:    [[ARG1_PRIV:%.*]] = alloca <8 x i64>, align 64
; IS__CGSCC_NPM-NEXT:    store <8 x i64> [[TMP0]], <8 x i64>* [[ARG1_PRIV]], align 64
```

```
; IS__CGSCC_NPM-NEXT:    [[TMP:%.*]] = load <8 x i64>, <8 x i64>* [[ARG1_PRIV]], align 64
; IS__CGSCC_NPM-NEXT:    store <8 x i64> [[TMP0]], <8 x i64>* [[ARG]], align 64
; IS__CGSCC_NPM-NEXT:    ret void
;
bb:
 %tmp = load <8 x i64>, <8 x i64>* %arg1
 store <8 x i64> %tmp, <8 x i64>* %arg
 ret void
}

define void @avx2_legal256_prefer256_call_avx2_legal512_prefer256(<8 x i64>* %arg) #4 {
;
; IS__TUNIT_OPM: Function Attrs: argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn
; IS__TUNIT_OPM-LABEL: define {{[^@]+}}@avx2_legal256_prefer256_call_avx2_legal512_prefer256
; IS__TUNIT_OPM-SAME: (<8 x i64>* nocapture nofree writeonly [[ARG:%.*]]) #[[ATTR4:[0-9]+]] {
; IS__TUNIT_OPM-NEXT:  bb:
; IS__TUNIT_OPM-NEXT:    [[TMP:%.*]] = alloca <8 x i64>, align 32
; IS__TUNIT_OPM-NEXT:
   [[TMP2:%.*]] = alloca <8 x i64>, align 32
; IS__TUNIT_OPM-NEXT:    [[TMP3:%.*]] = bitcast <8 x i64>* [[TMP]] to i8*
; IS__TUNIT_OPM-NEXT:    call void @llvm.memset.p0i8.i64(i8* nocapture nofree noundef nonnull writeonly
align 64 dereferenceable(64) [[TMP3]], i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR6]]
; IS__TUNIT_OPM-NEXT:    call fastcc void
@callee_avx2_legal256_prefer256_call_avx2_legal512_prefer256(<8 x i64>* nocapture nofree noundef nonnull
writeonly align 64 dereferenceable(64) [[TMP2]], <8 x i64>* nocapture nofree noundef nonnull readonly align 64
dereferenceable(64) [[TMP]]) #[[ATTR7]]
; IS__TUNIT_OPM-NEXT:    [[TMP4:%.*]] = load <8 x i64>, <8 x i64>* [[TMP2]], align 64
; IS__TUNIT_OPM-NEXT:    store <8 x i64> [[TMP4]], <8 x i64>* [[ARG]], align 2
; IS__TUNIT_OPM-NEXT:    ret void
;
; IS__TUNIT_NPM: Function Attrs: argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn
; IS__TUNIT_NPM-LABEL: define {{[^@]+}}@avx2_legal256_prefer256_call_avx2_legal512_prefer256
;
 IS__TUNIT_NPM-SAME: (<8 x i64>* nocapture nofree writeonly [[ARG:%.*]]) #[[ATTR4:[0-9]+]] {
; IS__TUNIT_NPM-NEXT:  bb:
; IS__TUNIT_NPM-NEXT:    [[TMP:%.*]] = alloca <8 x i64>, align 32
; IS__TUNIT_NPM-NEXT:    [[TMP2:%.*]] = alloca <8 x i64>, align 32
; IS__TUNIT_NPM-NEXT:    [[TMP3:%.*]] = bitcast <8 x i64>* [[TMP]] to i8*
; IS__TUNIT_NPM-NEXT:    call void @llvm.memset.p0i8.i64(i8* nocapture nofree noundef nonnull writeonly
align 64 dereferenceable(64) [[TMP3]], i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR6]]
; IS__TUNIT_NPM-NEXT:    [[TMP0:%.*]] = load <8 x i64>, <8 x i64>* [[TMP]], align 64
; IS__TUNIT_NPM-NEXT:    call fastcc void
@callee_avx2_legal256_prefer256_call_avx2_legal512_prefer256(<8 x i64>* noalias nocapture nofree noundef
nonnull writeonly align 64 dereferenceable(64) [[TMP2]], <8 x i64> [[TMP0]]) #[[ATTR7]]
; IS__TUNIT_NPM-NEXT:    [[TMP4:%.*]] = load <8 x i64>, <8 x i64>* [[TMP2]], align 64
; IS__TUNIT_NPM-NEXT:    store <8
 x i64> [[TMP4]], <8 x i64>* [[ARG]], align 2
; IS__TUNIT_NPM-NEXT:    ret void
```

```
;
; IS__CGSCC_OPM: Function Attrs: argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn
; IS__CGSCC_OPM-LABEL: define {{[^@]+}}@avx2_legal256_prefer256_call_avx2_legal512_prefer256
; IS__CGSCC_OPM-SAME: (<8 x i64>* nocapture nofree noundef nonnull writeonly align 2 dereferenceable(64)
[[ARG:%.*]]) #[[ATTR4:[0-9]+]] {
; IS__CGSCC_OPM-NEXT:  bb:
; IS__CGSCC_OPM-NEXT:    [[TMP:%.*]] = alloca <8 x i64>, align 32
; IS__CGSCC_OPM-NEXT:    [[TMP2:%.*]] = alloca <8 x i64>, align 32
; IS__CGSCC_OPM-NEXT:    [[TMP3:%.*]] = bitcast <8 x i64>* [[TMP]] to i8*
; IS__CGSCC_OPM-NEXT:    call void @llvm.memset.p0i8.i64(i8* nocapture nofree noundef nonnull writeonly
align 64 dereferenceable(64) [[TMP3]], i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR6]]
; IS__CGSCC_OPM-NEXT:    call fastcc void
@callee_avx2_legal256_prefer256_call_avx2_legal512_prefer256(<8 x i64>* nocapture nofree noundef nonnull
 writeonly align 64 dereferenceable(64) [[TMP2]], <8 x i64>* nocapture nofree noundef nonnull readonly align 64
dereferenceable(64) [[TMP]]) #[[ATTR7]]
; IS__CGSCC_OPM-NEXT:    [[TMP4:%.*]] = load <8 x i64>, <8 x i64>* [[TMP2]], align 64
; IS__CGSCC_OPM-NEXT:    store <8 x i64> [[TMP4]], <8 x i64>* [[ARG]], align 2
; IS__CGSCC_OPM-NEXT:    ret void
;
; IS__CGSCC_NPM: Function Attrs: argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn
; IS__CGSCC_NPM-LABEL: define {{[^@]+}}@avx2_legal256_prefer256_call_avx2_legal512_prefer256
; IS__CGSCC_NPM-SAME: (<8 x i64>* nocapture nofree noundef nonnull writeonly align 2 dereferenceable(64)
[[ARG:%.*]]) #[[ATTR4:[0-9]+]] {
; IS__CGSCC_NPM-NEXT:  bb:
; IS__CGSCC_NPM-NEXT:    [[TMP:%.*]] = alloca <8 x i64>, align 32
; IS__CGSCC_NPM-NEXT:    [[TMP2:%.*]] = alloca <8 x i64>, align 32
; IS__CGSCC_NPM-NEXT:    [[TMP3:%.*]] = bitcast <8 x i64>* [[TMP]] to i8*
; IS__CGSCC_NPM-NEXT:    call void @llvm.memset.p0i8.i64(i8* nocapture nofree
 noundef nonnull writeonly align 64 dereferenceable(64) [[TMP3]], i8 noundef 0, i64 noundef 32, i1 noundef false)
#[[ATTR6]]
; IS__CGSCC_NPM-NEXT:    [[TMP0:%.*]] = load <8 x i64>, <8 x i64>* [[TMP]], align 64
; IS__CGSCC_NPM-NEXT:    call fastcc void
@callee_avx2_legal256_prefer256_call_avx2_legal512_prefer256(<8 x i64>* noalias nocapture nofree noundef
nonnull writeonly align 64 dereferenceable(64) [[TMP2]], <8 x i64> [[TMP0]]) #[[ATTR7]]
; IS__CGSCC_NPM-NEXT:    [[TMP4:%.*]] = load <8 x i64>, <8 x i64>* [[TMP2]], align 64
; IS__CGSCC_NPM-NEXT:    store <8 x i64> [[TMP4]], <8 x i64>* [[ARG]], align 2
; IS__CGSCC_NPM-NEXT:    ret void
;
bb:
 %tmp = alloca <8 x i64>, align 32
 %tmp2 = alloca <8 x i64>, align 32
 %tmp3 = bitcast <8 x i64>* %tmp to i8*
 call void @llvm.memset.p0i8.i64(i8* align 32 %tmp3, i8 0, i64 32, i1 false)
 call fastcc void @callee_avx2_legal256_prefer256_call_avx2_legal512_prefer256(<8 x i64>* %tmp2, <8 x i64>*
%tmp)
 %tmp4 = load <8 x i64>, <8 x i64>* %tmp2,
 align 32
 store <8 x i64> %tmp4, <8 x i64>* %arg, align 2
```

```
 ret void
}

; This should promote
define internal fastcc void @callee_avx2_legal512_prefer256_call_avx2_legal256_prefer256(<8 x i64>* %arg, <8
x i64>* readonly %arg1) #4 {
;
; IS________OPM: Function Attrs: argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn
; IS________OPM-LABEL: define {{[^@]+}}@callee_avx2_legal512_prefer256_call_avx2_legal256_prefer256
; IS________OPM-SAME: (<8 x i64>* nocapture nofree noundef nonnull writeonly align 64 dereferenceable(64)
[[ARG:%.*]], <8 x i64>* nocapture nofree noundef nonnull readonly align 64 dereferenceable(64) [[ARG1:%.*]])
#[[ATTR4:[0-9]+]] {
; IS________OPM-NEXT:  bb:
; IS________OPM-NEXT:    [[TMP:%.*]] = load <8 x i64>, <8 x i64>* [[ARG1]], align 64
; IS________OPM-NEXT:    store <8 x i64> [[TMP]], <8 x i64>* [[ARG]], align 64
; IS________OPM-NEXT:    ret void
;
; IS__TUNIT_NPM: Function Attrs: argmemonly inlinehint nofree norecurse nosync nounwind uwtable
 willreturn
; IS__TUNIT_NPM-LABEL: define {{[^@]+}}@callee_avx2_legal512_prefer256_call_avx2_legal256_prefer256
; IS__TUNIT_NPM-SAME: (<8 x i64>* noalias nocapture nofree noundef nonnull writeonly align 64
dereferenceable(64) [[ARG:%.*]], <8 x i64> [[TMP0:%.*]]) #[[ATTR4]] {
; IS__TUNIT_NPM-NEXT:  bb:
; IS__TUNIT_NPM-NEXT:    [[ARG1_PRIV:%.*]] = alloca <8 x i64>, align 64
; IS__TUNIT_NPM-NEXT:    store <8 x i64> [[TMP0]], <8 x i64>* [[ARG1_PRIV]], align 64
; IS__TUNIT_NPM-NEXT:    [[TMP:%.*]] = load <8 x i64>, <8 x i64>* [[ARG1_PRIV]], align 64
; IS__TUNIT_NPM-NEXT:    store <8 x i64> [[TMP]], <8 x i64>* [[ARG]], align 64
; IS__TUNIT_NPM-NEXT:    ret void
;
; IS__CGSCC_NPM: Function Attrs: argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn
; IS__CGSCC_NPM-LABEL: define {{[^@]+}}@callee_avx2_legal512_prefer256_call_avx2_legal256_prefer256
; IS__CGSCC_NPM-SAME: (<8 x i64>* noalias nocapture nofree noundef nonnull writeonly align 64
dereferenceable(64) [[ARG:%.*]],
 <8 x i64> [[TMP0:%.*]]) #[[ATTR4]] {
; IS__CGSCC_NPM-NEXT:  bb:
; IS__CGSCC_NPM-NEXT:    [[ARG1_PRIV:%.*]] = alloca <8 x i64>, align 64
; IS__CGSCC_NPM-NEXT:    store <8 x i64> [[TMP0]], <8 x i64>* [[ARG1_PRIV]], align 64
; IS__CGSCC_NPM-NEXT:    [[TMP:%.*]] = load <8 x i64>, <8 x i64>* [[ARG1_PRIV]], align 64
; IS__CGSCC_NPM-NEXT:    store <8 x i64> [[TMP0]], <8 x i64>* [[ARG]], align 64
; IS__CGSCC_NPM-NEXT:    ret void
;
bb:
 %tmp = load <8 x i64>, <8 x i64>* %arg1
 store <8 x i64> %tmp, <8 x i64>* %arg
 ret void
}

define void @avx2_legal512_prefer256_call_avx2_legal256_prefer256(<8 x i64>* %arg) #3 {
```

```
;
; IS__TUNIT_OPM: Function Attrs: argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn
; IS__TUNIT_OPM-LABEL: define {{[^@]+}}@avx2_legal512_prefer256_call_avx2_legal256_prefer256
; IS__TUNIT_OPM-SAME: (<8 x i64>* nocapture nofree writeonly [[ARG:%.*]]) #[[ATTR3]] {
; IS__TUNIT_OPM-NEXT:  bb:
; IS__TUNIT_OPM-NEXT:    [[TMP:%.*]] = alloca <8 x i64>, align 32
;
 IS__TUNIT_OPM-NEXT:    [[TMP2:%.*]] = alloca <8 x i64>, align 32
; IS__TUNIT_OPM-NEXT:    [[TMP3:%.*]] = bitcast <8 x i64>* [[TMP]] to i8*
; IS__TUNIT_OPM-NEXT:    call void @llvm.memset.p0i8.i64(i8* nocapture nofree noundef nonnull writeonly
align 64 dereferenceable(64) [[TMP3]], i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR6]]
; IS__TUNIT_OPM-NEXT:    call fastcc void
@callee_avx2_legal512_prefer256_call_avx2_legal256_prefer256(<8 x i64>* nocapture nofree noundef nonnull
writeonly align 64 dereferenceable(64) [[TMP2]], <8 x i64>* nocapture nofree noundef nonnull readonly align 64
dereferenceable(64) [[TMP]]) #[[ATTR7]]
; IS__TUNIT_OPM-NEXT:    [[TMP4:%.*]] = load <8 x i64>, <8 x i64>* [[TMP2]], align 64
; IS__TUNIT_OPM-NEXT:    store <8 x i64> [[TMP4]], <8 x i64>* [[ARG]], align 2
; IS__TUNIT_OPM-NEXT:    ret void
;
; IS__TUNIT_NPM: Function Attrs: argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn
; IS__TUNIT_NPM-LABEL: define {{[^@]+}}@avx2_legal512_prefer256_call_avx2_legal256_prefer256
;
 IS__TUNIT_NPM-SAME: (<8 x i64>* nocapture nofree writeonly [[ARG:%.*]]) #[[ATTR3]] {
; IS__TUNIT_NPM-NEXT:  bb:
; IS__TUNIT_NPM-NEXT:    [[TMP:%.*]] = alloca <8 x i64>, align 32
; IS__TUNIT_NPM-NEXT:    [[TMP2:%.*]] = alloca <8 x i64>, align 32
; IS__TUNIT_NPM-NEXT:    [[TMP3:%.*]] = bitcast <8 x i64>* [[TMP]] to i8*
; IS__TUNIT_NPM-NEXT:    call void @llvm.memset.p0i8.i64(i8* nocapture nofree noundef nonnull writeonly
align 64 dereferenceable(64) [[TMP3]], i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR6]]
; IS__TUNIT_NPM-NEXT:    [[TMP0:%.*]] = load <8 x i64>, <8 x i64>* [[TMP]], align 64
; IS__TUNIT_NPM-NEXT:    call fastcc void
@callee_avx2_legal512_prefer256_call_avx2_legal256_prefer256(<8 x i64>* noalias nocapture nofree noundef
nonnull writeonly align 64 dereferenceable(64) [[TMP2]], <8 x i64> [[TMP0]]) #[[ATTR7]]
; IS__TUNIT_NPM-NEXT:    [[TMP4:%.*]] = load <8 x i64>, <8 x i64>* [[TMP2]], align 64
; IS__TUNIT_NPM-NEXT:    store
 <8 x i64> [[TMP4]], <8 x i64>* [[ARG]], align 2
; IS__TUNIT_NPM-NEXT:    ret void
;
; IS__CGSCC_OPM: Function Attrs: argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn
; IS__CGSCC_OPM-LABEL: define {{[^@]+}}@avx2_legal512_prefer256_call_avx2_legal256_prefer256
; IS__CGSCC_OPM-SAME: (<8 x i64>* nocapture nofree noundef nonnull writeonly align 2 dereferenceable(64)
[[ARG:%.*]]) #[[ATTR3]] {
; IS__CGSCC_OPM-NEXT:  bb:
; IS__CGSCC_OPM-NEXT:    [[TMP:%.*]] = alloca <8 x i64>, align 32
; IS__CGSCC_OPM-NEXT:    [[TMP2:%.*]] = alloca <8 x i64>, align 32
; IS__CGSCC_OPM-NEXT:    [[TMP3:%.*]] = bitcast <8 x i64>* [[TMP]] to i8*
; IS__CGSCC_OPM-NEXT:    call void @llvm.memset.p0i8.i64(i8* nocapture nofree noundef nonnull writeonly
align 64 dereferenceable(64) [[TMP3]], i8 noundef 0, i64 noundef 32, i1 noundef false) #[[ATTR6]]
```

```
; IS__CGSCC_OPM-NEXT:    call fastcc void
@callee_avx2_legal512_prefer256_call_avx2_legal256_prefer256(<8 x i64>* nocapture nofree noundef nonnull
 writeonly align 64 dereferenceable(64) [[TMP2]], <8 x i64>* nocapture nofree noundef nonnull readonly align 64
dereferenceable(64) [[TMP]]) #[[ATTR7]]
; IS__CGSCC_OPM-NEXT:    [[TMP4:%.*]] = load <8 x i64>, <8 x i64>* [[TMP2]], align 64
; IS__CGSCC_OPM-NEXT:    store <8 x i64> [[TMP4]], <8 x i64>* [[ARG]], align 2
; IS__CGSCC_OPM-NEXT:    ret void
;
; IS__CGSCC_NPM: Function Attrs: argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn
; IS__CGSCC_NPM-LABEL: define {{[^@]+}}@avx2_legal512_prefer256_call_avx2_legal256_prefer256
; IS__CGSCC_NPM-SAME: (<8 x i64>* nocapture nofree noundef nonnull writeonly align 2 dereferenceable(64)
[[ARG:%.*]]) #[[ATTR3]] {
; IS__CGSCC_NPM-NEXT:  bb:
; IS__CGSCC_NPM-NEXT:    [[TMP:%.*]] = alloca <8 x i64>, align 32
; IS__CGSCC_NPM-NEXT:    [[TMP2:%.*]] = alloca <8 x i64>, align 32
; IS__CGSCC_NPM-NEXT:    [[TMP3:%.*]] = bitcast <8 x i64>* [[TMP]] to i8*
; IS__CGSCC_NPM-NEXT:    call void @llvm.memset.p0i8.i64(i8* nocapture nofree noundef
 nonnull writeonly align 64 dereferenceable(64) [[TMP3]], i8 noundef 0, i64 noundef 32, i1 noundef false)
#[[ATTR6]]
; IS__CGSCC_NPM-NEXT:    [[TMP0:%.*]] = load <8 x i64>, <8 x i64>* [[TMP]], align 64
; IS__CGSCC_NPM-NEXT:    call fastcc void
@callee_avx2_legal512_prefer256_call_avx2_legal256_prefer256(<8 x i64>* noalias nocapture nofree noundef
nonnull writeonly align 64 dereferenceable(64) [[TMP2]], <8 x i64> [[TMP0]]) #[[ATTR7]]
; IS__CGSCC_NPM-NEXT:    [[TMP4:%.*]] = load <8 x i64>, <8 x i64>* [[TMP2]], align 64
; IS__CGSCC_NPM-NEXT:    store <8 x i64> [[TMP4]], <8 x i64>* [[ARG]], align 2
; IS__CGSCC_NPM-NEXT:    ret void
;
bb:
 %tmp = alloca <8 x i64>, align 32
 %tmp2 = alloca <8 x i64>, align 32
 %tmp3 = bitcast <8 x i64>* %tmp to i8*
 call void @llvm.memset.p0i8.i64(i8* align 32 %tmp3, i8 0, i64 32, i1 false)
 call fastcc void @callee_avx2_legal512_prefer256_call_avx2_legal256_prefer256(<8 x i64>* %tmp2, <8 x i64>*
%tmp)
 %tmp4 = load <8 x i64>, <8 x i64>* %tmp2, align
 32
 store <8 x i64> %tmp4, <8 x i64>* %arg, align 2
 ret void
}

; Function Attrs: argmemonly nounwind
declare void @llvm.memset.p0i8.i64(i8* nocapture writeonly, i8, i64, i1) #5

attributes #0 = { inlinehint norecurse nounwind uwtable "target-features"="+avx512vl" "min-legal-vector-
width"="512" "prefer-vector-width"="512" }
attributes #1 = { inlinehint norecurse nounwind uwtable "target-features"="+avx512vl" "min-legal-vector-
width"="512" "prefer-vector-width"="256" }
attributes #2 = { inlinehint norecurse nounwind uwtable "target-features"="+avx512vl" "min-legal-vector-
```

width"="256" "prefer-vector-width"="256" }
attributes #3 = { inlinehint norecurse nounwind uwtable "target-features"="+avx2" "min-legal-vector-width"="512" "prefer-vector-width"="256" }
attributes #4 = { inlinehint norecurse nounwind uwtable "target-features"="+avx2" "min-legal-vector-width"="256" "prefer-vector-width"="256" }
attributes #5 = { argmemonly nounwind }
;.
; IS__TUNIT____: attributes #[[ATTR0:[0-9]+]]
 = { argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn "min-legal-vector-width"="512" "prefer-vector-width"="512" "target-features"="+avx512vl" }
; IS__TUNIT____: attributes #[[ATTR1:[0-9]+]] = { argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn "min-legal-vector-width"="512" "prefer-vector-width"="256" "target-features"="+avx512vl" }
; IS__TUNIT____: attributes #[[ATTR2:[0-9]+]] = { argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn "min-legal-vector-width"="256" "prefer-vector-width"="256" "target-features"="+avx512vl" }
; IS__TUNIT____: attributes #[[ATTR3:[0-9]+]] = { argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn "min-legal-vector-width"="512" "prefer-vector-width"="256" "target-features"="+avx2" }
; IS__TUNIT____: attributes #[[ATTR4:[0-9]+]] = { argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn "min-legal-vector-width"="256" "prefer-vector-width"="256" "target-features"="+avx2"
 }
; IS__TUNIT____: attributes #[[ATTR5:[0-9]+]] = { argmemonly nofree nounwind willreturn writeonly }
; IS__TUNIT____: attributes #[[ATTR6:[0-9]+]] = { willreturn writeonly }
; IS__TUNIT____: attributes #[[ATTR7:[0-9]+]] = { nofree nosync nounwind willreturn }
;.
; IS__CGSCC____: attributes #[[ATTR0:[0-9]+]] = { argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn "min-legal-vector-width"="512" "prefer-vector-width"="512" "target-features"="+avx512vl" }
; IS__CGSCC____: attributes #[[ATTR1:[0-9]+]] = { argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn "min-legal-vector-width"="512" "prefer-vector-width"="256" "target-features"="+avx512vl" }
; IS__CGSCC____: attributes #[[ATTR2:[0-9]+]] = { argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn "min-legal-vector-width"="256" "prefer-vector-width"="256" "target-features"="+avx512vl" }
; IS__CGSCC____: attributes #[[ATTR3:[0-9]+]] = { argmemonly inlinehint
 nofree norecurse nosync nounwind uwtable willreturn "min-legal-vector-width"="512" "prefer-vector-width"="256" "target-features"="+avx2" }
; IS__CGSCC____: attributes #[[ATTR4:[0-9]+]] = { argmemonly inlinehint nofree norecurse nosync nounwind uwtable willreturn "min-legal-vector-width"="256" "prefer-vector-width"="256" "target-features"="+avx2" }
; IS__CGSCC____: attributes #[[ATTR5:[0-9]+]] = { argmemonly nofree nounwind willreturn writeonly }
; IS__CGSCC____: attributes #[[ATTR6:[0-9]+]] = { willreturn writeonly }
; IS__CGSCC____: attributes #[[ATTR7:[0-9]+]] = { nounwind willreturn }
;.
==========

The LLVM Project is under the Apache License v2.0 with LLVM Exceptions:

==========

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the
power, direct or indirect, to cause the
direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached
to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of

the copyright owner. For the purposes of this definition,
"submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative
Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work
includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional
attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License
does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or

agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including
negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such
obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

---- LLVM Exceptions to the Apache 2.0 License ----

As an exception, if, as a result of your compiling your source code, portions
of this Software are embedded into an Object form of such source code, you
may redistribute such embedded portions in such Object form without complying
with the conditions of Sections 4(a), 4(b) and 4(d) of the License.

In addition, if you combine or link compiled
forms of this Software with
software that is licensed under the GPLv2 ("Combined Software") and if a
court of competent jurisdiction determines that the patent provision (Section
3), the indemnity provision (Section 9) or other Section of the License
conflicts with the conditions of the GPLv2, you may retroactively and
prospectively choose to deem waived or otherwise exclude such Section(s) of
the License, but only in their entirety and only with respect to the Combined
Software.

==
Software from third parties included in the LLVM Project:
==
The LLVM Project contains third party software which is under different license
terms. All such code will be identified clearly using at least one of two
mechanisms:
1) It will be in a separate directory tree with its own `LICENSE.txt` or
`LICENSE` file at the top containing the specific
license and restrictions
which apply to that software, or
2) It will contain specific license and restriction terms at the top of every
file.

==
Legacy LLVM License (https://llvm.org/docs/DeveloperPolicy.html#legacy):
==

; NOTE: Assertions have been autogenerated by utils/update_llc_test_checks.py
; RUN: llc < %s -mtriple=x86_64-linux-android -mattr=+mmx -enable-legalize-types-checking | FileCheck %s
;
; D31946
; Check that we dont end up with the ""LLVM ERROR: Cannot select" error.
; Additionally ensure that the output code actually put fp128 values in SSE registers.

declare fp128 @llvm.fabs.f128(fp128)
declare fp128 @llvm.copysign.f128(fp128, fp128)

define fp128 @TestSelect(fp128 %a, fp128 %b) {

```
; CHECK-LABEL: TestSelect:
; CHECK:       # %bb.0:
; CHECK-NEXT:    pushq %rbx
; CHECK-NEXT:    .cfi_def_cfa_offset 16
; CHECK-NEXT:    subq $32, %rsp
; CHECK-NEXT:    .cfi_def_cfa_offset 48
; CHECK-NEXT:    .cfi_offset %rbx, -16
; CHECK-NEXT:    movaps %xmm1, {{[-0-9]+}}(%r{{[sb]}}p) # 16-byte Spill
; CHECK-NEXT:    movaps %xmm0, (%rsp) # 16-byte Spill
; CHECK-NEXT:    callq __gttf2@PLT
; CHECK-NEXT:    movl %eax, %ebx
; CHECK-NEXT:    movaps (%rsp), %xmm0 # 16-byte Reload
; CHECK-NEXT:    movaps {{[-0-9]+}}(%r{{[sb]}}p),
 %xmm1 # 16-byte Reload
; CHECK-NEXT:    callq __subtf3@PLT
; CHECK-NEXT:    testl %ebx, %ebx
; CHECK-NEXT:    jg .LBB0_2
; CHECK-NEXT:  # %bb.1:
; CHECK-NEXT:    xorps %xmm0, %xmm0
; CHECK-NEXT:  .LBB0_2:
; CHECK-NEXT:    addq $32, %rsp
; CHECK-NEXT:    .cfi_def_cfa_offset 16
; CHECK-NEXT:    popq %rbx
; CHECK-NEXT:    .cfi_def_cfa_offset 8
; CHECK-NEXT:    retq
 %cmp = fcmp ogt fp128 %a, %b
 %sub = fsub fp128 %a, %b
 %res = select i1 %cmp, fp128 %sub, fp128 0xL00000000000000000000000000000000
 ret fp128 %res
}

define fp128 @TestFabs(fp128 %a) {
; CHECK-LABEL: TestFabs:
; CHECK:       # %bb.0:
; CHECK-NEXT:    andps {{\.?LCPI[0-9]+_[0-9]+}}(%rip), %xmm0
; CHECK-NEXT:    retq
 %res = call fp128 @llvm.fabs.f128(fp128 %a)
 ret fp128 %res
}

define fp128 @TestCopysign(fp128 %a, fp128 %b) {
; CHECK-LABEL: TestCopysign:
; CHECK:       # %bb.0:
; CHECK-NEXT:    andps {{\.?LCPI[0-9]+_[0-9]+}}(%rip), %xmm1
; CHECK-NEXT:    andps {{\.?LCPI[0-9]+_[0-9]+}}(%rip), %xmm0
; CHECK-NEXT:
   orps %xmm1, %xmm0
; CHECK-NEXT:    retq
```

```
 %res = call fp128 @llvm.copysign.f128(fp128 %a, fp128 %b)
 ret fp128 %res
}

define fp128 @TestFneg(fp128 %a) {
; CHECK-LABEL: TestFneg:
; CHECK:       # %bb.0:
; CHECK-NEXT:    pushq %rax
; CHECK-NEXT:    .cfi_def_cfa_offset 16
; CHECK-NEXT:    movaps %xmm0, %xmm1
; CHECK-NEXT:    callq __multf3@PLT
; CHECK-NEXT:    xorps {{\.?LCPI[0-9]+_[0-9]+}}(%rip), %xmm0
; CHECK-NEXT:    popq %rax
; CHECK-NEXT:    .cfi_def_cfa_offset 8
; CHECK-NEXT:    retq
 %mul = fmul fp128 %a, %a
 %res = fsub fp128 0xL00000000000000008000000000000000, %mul
 ret fp128 %res
}
```

```
; RUN: opt %s -inline -S | FileCheck %s

define internal void @innerSmall() "min-legal-vector-width"="128" {
 ret void
}

define internal void @innerLarge() "min-legal-vector-width"="512" {
 ret void
}

define internal void @innerNoAttribute() {
 ret void
}

; We should not add an attribute during inlining. No attribute means unknown.
; Inlining doesn't change the fact that we don't know anything about this
; function.
define void @outerNoAttribute() {
 call void @innerLarge()
 ret void
}

define void @outerConflictingAttributeSmall() "min-legal-vector-width"="128" {
 call void @innerLarge()
 ret void
}

define void @outerConflictingAttributeLarge() "min-legal-vector-width"="512" {
 call void @innerSmall()
 ret void
}

; We should remove the attribute after inlining since the callee's
; vector width requirements are unknown.
define void @outerAttribute() "min-legal-vector-width"="128" {
 call void @innerNoAttribute()
 ret void
}

; CHECK: define void @outerNoAttribute() {
; CHECK:
 define void @outerConflictingAttributeSmall() #0
; CHECK: define void @outerConflictingAttributeLarge() #0
; CHECK: define void @outerAttribute() {
; CHECK: attributes #0 = { "min-legal-vector-width"="512" }
```

---

clang-tidy CERT Files

All clang-tidy files are licensed under the same terms as the rest of the LLVM
project with the following additions:

Any file referencing a CERT Secure Coding guideline:
Please allow this letter to serve as confirmation that open source projects on
http://llvm.org are permitted to link via hypertext to the CERT(R) secure coding
guidelines available at https://www.securecoding.cert.org.

The foregoing is permitted by the Terms of Use as follows:
"Linking to the Service
Because we update many of our Web documents regularly, we would prefer that you
link to our Web pages whenever possible rather than reproduce them. It is not
necessary to request permission to make referential hypertext links to The
Service."
http://www.sei.cmu.edu/legal/ip/index.cfm.

Please allow this letter to also confirm that no formal permission
 is required
to reproduce the title of the content being linked to, nor to reproduce any
de Minimis description of such content.
; RUN: llc -march=hexagon < %s
; REQUIRES: asserts

; The two loads based on %struct.0, loading two different data types
; cause LSR to assume type "void" for the memory type. This would then
; cause an assert in isLegalAddressingMode. Make sure we no longer crash.

target triple = "hexagon"

%struct.0 = type { i8*, i8, %union.anon.0 }
%union.anon.0 = type { i8* }

define hidden fastcc void @fred() unnamed_addr #0 {
entry:
 br i1 undef, label %while.end, label %while.body.lr.ph

while.body.lr.ph: ; preds = %entry
 br label %while.body

while.body: ; preds = %exit.2, %while.body.lr.ph
 %lsr.iv = phi %struct.0* [%cgep22, %exit.2], [undef, %while.body.lr.ph]
 switch i32 undef, label %exit [
   i32 1, label %sw.bb.i
   i32 2, label %sw.bb3.i
 ]

sw.bb.i: ; preds = %while.body
 unreachable

```
sw.bb3.i:                                         ; preds = %while.body
 unreachable

exit:
                                                  ; preds = %while.body
 switch i32 undef, label %exit.2 [
   i32 1, label %sw.bb.i17
   i32 2, label %sw.bb3.i20
 ]

sw.bb.i17:                                        ; preds = %.exit
 %0 = bitcast %struct.0* %lsr.iv to i32*
 %1 = load i32, i32* %0, align 4
 unreachable

sw.bb3.i20:                                       ; preds = %exit
 %2 = bitcast %struct.0* %lsr.iv to i8**
 %3 = load i8*, i8** %2, align 4
 unreachable

exit.2:                                           ; preds = %exit
 %cgep22 = getelementptr %struct.0, %struct.0* %lsr.iv, i32 1
 br label %while.body

while.end:                                        ; preds = %entry
 ret void
}

attributes #0 = { nounwind optsize "target-cpu"="hexagonv55" }
```

==========

The LLVM Project is under the Apache License v2.0 with LLVM Exceptions:

==========

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the
power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached
to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition,
"submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise

designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work

includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional
attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License
does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including
negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such
obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

---- LLVM Exceptions to the Apache 2.0 License ----

As an exception, if, as a result of your compiling your source code, portions
of this Software are embedded into an Object form of such source code, you
may redistribute such embedded portions in such Object form without complying
with the conditions of Sections 4(a), 4(b) and 4(d) of the License.

In addition, if you combine or link compiled
forms of this Software with
software that is licensed under the GPLv2 ("Combined Software") and if a
court of competent jurisdiction determines that the patent provision (Section
3), the indemnity provision (Section 9) or other Section of the License
conflicts with the conditions of the GPLv2, you may retroactively and
prospectively choose to deem waived or otherwise exclude such Section(s) of
the License, but only in their entirety and only with respect to the Combined
Software.

==
Software from third parties included in the LLVM Project:
==
The LLVM Project contains third party software which is under different license
terms. All such code will be identified clearly using at least one of two
mechanisms:
1) It will be in a separate directory tree with its own `LICENSE.txt` or
`LICENSE` file at the top containing the specific
license and restrictions
which apply to that software, or
2) It will contain specific license and restriction terms at the top of every
file.

==
Legacy LLVM License (https://llvm.org/docs/DeveloperPolicy.html#legacy):
==

The libclc library is dual licensed under both the University of Illinois
"BSD-Like" license and the MIT license. As a user of this code you may choose
to use it under either license. As a contributor, you agree to allow your code
to be used under both.

Full text of the relevant licenses is included below.

==

Copyright (c) 2011-2019 by the contributors listed in CREDITS.TXT

All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"),
 to deal with
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies
of the Software, and to permit persons to whom the Software is furnished to do
so, subject to the following conditions:

  * Redistributions of source code must retain the above copyright notice,
    this list of conditions and the following disclaimers.

  * Redistributions in binary form must reproduce the above copyright notice,
    this list of conditions and the following disclaimers in the
    documentation and/or other materials provided with the distribution.

  * The names of the contributors may not be used to endorse or promote
    products derived from this Software without specific prior written
    permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR
 PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE
SOFTWARE.

==========

Copyright (c) 2011-2014 by the contributors listed in CREDITS.TXT

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE

SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

```
; RUN: opt -consthoist -S -o - %s | FileCheck %s
target datalayout = "e-m:e-p:32:32-i64:64-v128:64:128-a:0:32-n32-S64"
target triple = "thumbv6m-none--musleabi"

; Check that for i8 type, the maximum legal offset is 31.
; Also check that an constant used as value to be stored rather than
; pointer in a store instruction is hoisted.
; CHECK: foo_i8
; CHECK-DAG:  %[[C1:const[0-9]?]] = bitcast i32 805874720 to i32
; CHECK-DAG:  %[[C2:const[0-9]?]] = bitcast i32 805874688 to i32
; CHECK-DAG:  %[[C3:const[0-9]?]] = bitcast i32 805873720 to i32
; CHECK-DAG:  %[[C4:const[0-9]?]] = bitcast i32 805873688 to i32
; CHECK:  %0 = inttoptr i32 %[[C2]] to i8*
; CHECK-NEXT:  %1 = load volatile i8, i8* %0
; CHECK-NEXT:  %[[M1:const_mat[0-9]?]] = add i32 %[[C2]], 4
; CHECK-NEXT:  %2 = inttoptr i32 %[[M1]] to i8*
; CHECK-NEXT:  %3 = load volatile i8, i8* %2
; CHECK-NEXT:  %[[M2:const_mat[0-9]?]] = add i32 %[[C2]], 31
; CHECK-NEXT:  %4 = inttoptr i32 %[[M2]] to i8*
; CHECK-NEXT:  %5 = load volatile i8, i8*
 %4
; CHECK-NEXT:  %6 = inttoptr i32 %[[C1]] to i8*
; CHECK-NEXT:  %7 = load volatile i8, i8* %6
; CHECK-NEXT:  %[[M3:const_mat[0-9]?]] = add i32 %[[C1]], 7
; CHECK-NEXT:  %8 = inttoptr i32 %[[M3]] to i8*
; CHECK-NEXT:  %9 = load volatile i8, i8* %8
; CHECK-NEXT:  %10 = inttoptr i32 %[[C4]] to i8*
; CHECK-NEXT:  store i8 %9, i8* %10
; CHECK-NEXT:  %[[M4:const_mat[0-9]?]] = add i32 %[[C4]], 31
; CHECK-NEXT:  %11 = inttoptr i32 %[[M4]] to i8*
; CHECK-NEXT:  store i8 %7, i8* %11
; CHECK-NEXT:  %12 = inttoptr i32 %[[C3]] to i8*
; CHECK-NEXT:  store i8 %5, i8* %12
; CHECK-NEXT:  %[[M5:const_mat[0-9]?]] = add i32 %[[C3]], 7
; CHECK-NEXT:  %13 = inttoptr i32 %[[M5]] to i8*
; CHECK-NEXT:  store i8 %3, i8* %13
; CHECK-NEXT:  %[[M6:const_mat[0-9]?]] = add i32 %[[C1]], 80
; CHECK-NEXT:  %14 = inttoptr i32 %[[M6]] to i8*
; CHECK-NEXT:  store i8* %14, i8** @goo

@goo = global i8* undef
```

```
define void @foo_i8() {
entry:
 %0 = load volatile i8, i8* inttoptr (i32 805874688 to i8*)
 %1 = load volatile
 i8, i8* inttoptr (i32 805874692 to i8*)
 %2 = load volatile i8, i8* inttoptr (i32 805874719 to i8*)
 %3 = load volatile i8, i8* inttoptr (i32 805874720 to i8*)
 %4 = load volatile i8, i8* inttoptr (i32 805874727 to i8*)
 store i8 %4, i8* inttoptr(i32 805873688 to i8*)
 store i8 %3, i8* inttoptr(i32 805873719 to i8*)
 store i8 %2, i8* inttoptr(i32 805873720 to i8*)
 store i8 %1, i8* inttoptr(i32 805873727 to i8*)
 store i8* inttoptr(i32 805874800 to i8*), i8** @goo
 ret void
}

; Check that for i16 type, the maximum legal offset is 62.
; CHECK: foo_i16
; CHECK-DAG: %[[C1:const[0-9]?]] = bitcast i32 805874752 to i32
; CHECK-DAG: %[[C2:const[0-9]?]] = bitcast i32 805874688 to i32
; CHECK: %0 = inttoptr i32 %[[C2]] to i16*
; CHECK-NEXT: %1 = load volatile i16, i16* %0, align 2
; CHECK-NEXT: %[[M1:const_mat[0-9]?]] = add i32 %[[C2]], 4
; CHECK-NEXT: %2 = inttoptr i32 %[[M1]] to i16*
; CHECK-NEXT: %3 = load volatile i16, i16* %2, align 2
; CHECK-NEXT: %[[M2:const_mat[0-9]?]] = add i32
 %[[C2]], 32
; CHECK-NEXT: %4 = inttoptr i32 %[[M2]] to i16*
; CHECK-NEXT: %5 = load volatile i16, i16* %4, align 2
; CHECK-NEXT: %[[M3:const_mat[0-9]?]] = add i32 %[[C2]], 62
; CHECK-NEXT: %6 = inttoptr i32 %[[M3]] to i16*
; CHECK-NEXT: %7 = load volatile i16, i16* %6, align 2
; CHECK-NEXT: %8 = inttoptr i32 %[[C1]] to i16*
; CHECK-NEXT: %9 = load volatile i16, i16* %8, align 2
; CHECK-NEXT: %[[M4:const_mat[0-9]?]] = add i32 %[[C1]], 22
; CHECK-NEXT: %10 = inttoptr i32 %[[M4]] to i16*
; CHECK-NEXT: %11 = load volatile i16, i16* %10, align 2

define void @foo_i16() {
entry:
 %0 = load volatile i16, i16* inttoptr (i32 805874688 to i16*), align 2
 %1 = load volatile i16, i16* inttoptr (i32 805874692 to i16*), align 2
 %2 = load volatile i16, i16* inttoptr (i32 805874720 to i16*), align 2
 %3 = load volatile i16, i16* inttoptr (i32 805874750 to i16*), align 2
 %4 = load volatile i16, i16* inttoptr (i32 805874752 to i16*), align 2
 %5 = load volatile i16, i16* inttoptr (i32 805874774
```

```
 to i16*), align 2
 ret void
}

; Check that for i32 type, the maximum legal offset is 124.
; CHECK: foo_i32
; CHECK-DAG:  %[[C1:const[0-9]?]] = bitcast i32 805874816 to i32
; CHECK-DAG:  %[[C2:const[0-9]?]] = bitcast i32 805874688 to i32
; CHECK:  %0 = inttoptr i32 %[[C2]] to i32*
; CHECK-NEXT:  %1 = load volatile i32, i32* %0, align 4
; CHECK-NEXT:  %[[M1:const_mat[0-9]?]] = add i32 %[[C2]], 4
; CHECK-NEXT:  %2 = inttoptr i32 %[[M1]] to i32*
; CHECK-NEXT:  %3 = load volatile i32, i32* %2, align 4
; CHECK-NEXT:  %[[M2:const_mat[0-9]?]] = add i32 %[[C2]], 124
; CHECK-NEXT:  %4 = inttoptr i32 %[[M2]] to i32*
; CHECK-NEXT:  %5 = load volatile i32, i32* %4, align 4
; CHECK-NEXT:  %6 = inttoptr i32 %[[C1]] to i32*
; CHECK-NEXT:  %7 = load volatile i32, i32* %6, align 4
; CHECK-NEXT:  %[[M3:const_mat[0-9]?]] = add i32 %[[C1]], 8
; CHECK-NEXT:  %8 = inttoptr i32 %[[M3]] to i32*
; CHECK-NEXT:  %9 = load volatile i32, i32* %8, align 4
; CHECK-NEXT:  %[[M4:const_mat[0-9]?]] = add i32 %[[C1]],
 12
; CHECK-NEXT:  %10 = inttoptr i32 %[[M4]] to i32*
; CHECK-NEXT:  %11 = load volatile i32, i32* %10, align 4

define void @foo_i32() {
entry:
 %0 = load volatile i32, i32* inttoptr (i32 805874688 to i32*), align 4
 %1 = load volatile i32, i32* inttoptr (i32 805874692 to i32*), align 4
 %2 = load volatile i32, i32* inttoptr (i32 805874812 to i32*), align 4
 %3 = load volatile i32, i32* inttoptr (i32 805874816 to i32*), align 4
 %4 = load volatile i32, i32* inttoptr (i32 805874824 to i32*), align 4
 %5 = load volatile i32, i32* inttoptr (i32 805874828 to i32*), align 4
 ret void
}
```

distribution.
  * Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR
 CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
==========
The LLVM Project is under the Apache License v2.0 with LLVM Exceptions:
==========

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the
power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,

including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside

or as an addendum to the NOTICE text from the Work, provided
that such additional
attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License
does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including
negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such
obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

---- LLVM Exceptions to the Apache 2.0 License ----

As an exception, if, as a result of your compiling your source code, portions
of this Software are embedded into an Object form of such source code, you
may redistribute such embedded portions in such Object form without complying
with the conditions of Sections 4(a), 4(b) and 4(d) of the License.

In addition, if you combine or link compiled
forms of this Software with
software that is licensed under the GPLv2 ("Combined Software") and if a
court of competent jurisdiction determines that the patent provision (Section
3), the indemnity provision (Section 9) or other Section of the License
conflicts with the conditions of the GPLv2, you may retroactively and
prospectively choose to deem waived or otherwise exclude such Section(s) of
the License, but only in their entirety and only with respect to the Combined
Software.

==
Software from third parties included in the LLVM Project:
==
The LLVM Project contains third party software which is under different license
terms. All such code will be identified clearly using at least one of two
mechanisms:
1) It will be in a separate directory tree with its own `LICENSE.txt` or
`LICENSE` file at the top containing the specific
license and restrictions
which apply to that software, or
2) It will contain specific license and restriction terms at the top of every
file.

==
Legacy LLVM License (https://llvm.org/docs/DeveloperPolicy.html#legacy):
==

The GPURuntime library is dual licensed under both the University of Illinois
"BSD-Like" license and the MIT license. As a user of this code you may choose
to use it under either license. As a contributor, you agree to allow your code
to be used under both.

Full text of the relevant licenses is included below.

==

University of Illinois/NCSA
Open Source License

Copyright (c) 2009-2019 by the contributors listed in CREDITS.TXT

All rights reserved.

Developed by:

Polly Team

http://polly.llvm.org

Permission is hereby granted,
free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal with
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies
of the Software, and to permit persons to whom the Software is furnished to do
so, subject to the following conditions:

* Redistributions of source code must retain the above copyright notice,
this list of conditions and the following disclaimers.

* Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimers in the
documentation and/or other materials provided with the distribution.

* Neither the names of the LLVM Team, University of Illinois at
Urbana-Champaign, nor the names of its contributors may be used to
endorse or promote products derived from this Software without specific
prior
written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

==========

Copyright (c) 2009-2016 by the contributors listed in CREDITS.TXT

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom
the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE

```
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
; NOTE: Assertions have been autogenerated by utils/update_test_checks.py UTC_ARGS: --function-signature --
scrub-attributes
; RUN: opt -S -argpromotion < %s | FileCheck %s
; RUN: opt -S -passes=argpromotion < %s | FileCheck %s
; Test that we only promote arguments when the caller/callee have compatible
; function attrubtes.

target triple = "x86_64-unknown-linux-gnu"

; This should promote
define internal fastcc void @callee_avx512_legal512_prefer512_call_avx512_legal512_prefer512(<8 x i64>* %arg,
<8 x i64>* readonly %arg1) #0 {
; CHECK-LABEL: define {{[^@]+}}@callee_avx512_legal512_prefer512_call_avx512_legal512_prefer512
; CHECK-SAME: (<8 x i64>* [[ARG:%.*]], <8 x i64> [[ARG1_VAL:%.*]])
; CHECK-NEXT:  bb:
; CHECK-NEXT:    store <8 x i64> [[ARG1_VAL]], <8 x i64>* [[ARG]]
; CHECK-NEXT:    ret void
;
bb:
 %tmp = load <8 x i64>, <8 x i64>* %arg1
 store <8 x i64> %tmp, <8 x i64>* %arg
 ret void
}

define void @avx512_legal512_prefer512_call_avx512_legal512_prefer512(<8 x i64>* %arg) #0
 {
; CHECK-LABEL: define {{[^@]+}}@avx512_legal512_prefer512_call_avx512_legal512_prefer512
; CHECK-SAME: (<8 x i64>* [[ARG:%.*]])
; CHECK-NEXT:  bb:
; CHECK-NEXT:    [[TMP:%.*]] = alloca <8 x i64>, align 32
; CHECK-NEXT:    [[TMP2:%.*]] = alloca <8 x i64>, align 32
; CHECK-NEXT:    [[TMP3:%.*]] = bitcast <8 x i64>* [[TMP]] to i8*
; CHECK-NEXT:    call void @llvm.memset.p0i8.i64(i8* align 32 [[TMP3]], i8 0, i64 32, i1 false)
; CHECK-NEXT:    [[TMP_VAL:%.*]] = load <8 x i64>, <8 x i64>* [[TMP]]
; CHECK-NEXT:    call fastcc void @callee_avx512_legal512_prefer512_call_avx512_legal512_prefer512(<8 x
i64>* [[TMP2]], <8 x i64> [[TMP_VAL]])
; CHECK-NEXT:    [[TMP4:%.*]] = load <8 x i64>, <8 x i64>* [[TMP2]], align 32
; CHECK-NEXT:    store <8 x i64> [[TMP4]], <8 x i64>* [[ARG]], align 2
; CHECK-NEXT:    ret void
;
bb:
 %tmp = alloca <8 x i64>, align 32
 %tmp2 = alloca <8 x i64>, align 32
 %tmp3 = bitcast <8 x i64>* %tmp to i8*
```

```
 call void @llvm.memset.p0i8.i64(i8* align 32 %tmp3, i8 0, i64
 32, i1 false)
 call fastcc void @callee_avx512_legal512_prefer512_call_avx512_legal512_prefer512(<8 x i64>* %tmp2, <8 x
i64>* %tmp)
 %tmp4 = load <8 x i64>, <8 x i64>* %tmp2, align 32
 store <8 x i64> %tmp4, <8 x i64>* %arg, align 2
 ret void
}

; This should promote
define internal fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal512_prefer256(<8 x i64>* %arg,
<8 x i64>* readonly %arg1) #1 {
; CHECK-LABEL: define {{[^@]+}}@callee_avx512_legal512_prefer256_call_avx512_legal512_prefer256
; CHECK-SAME: (<8 x i64>* [[ARG:%.*]], <8 x i64> [[ARG1_VAL:%.*]])
; CHECK-NEXT:  bb:
; CHECK-NEXT:    store <8 x i64> [[ARG1_VAL]], <8 x i64>* [[ARG]]
; CHECK-NEXT:    ret void
;
bb:
 %tmp = load <8 x i64>, <8 x i64>* %arg1
 store <8 x i64> %tmp, <8 x i64>* %arg
 ret void
}

define void @avx512_legal512_prefer256_call_avx512_legal512_prefer256(<8 x i64>* %arg) #1 {
; CHECK-LABEL: define {{[^@]+}}@avx512_legal512_prefer256_call_avx512_legal512_prefer256
; CHECK-SAME: (<8 x i64>* [[ARG:%.*]])
;
 CHECK-NEXT:  bb:
; CHECK-NEXT:    [[TMP:%.*]] = alloca <8 x i64>, align 32
; CHECK-NEXT:    [[TMP2:%.*]] = alloca <8 x i64>, align 32
; CHECK-NEXT:    [[TMP3:%.*]] = bitcast <8 x i64>* [[TMP]] to i8*
; CHECK-NEXT:    call void @llvm.memset.p0i8.i64(i8* align 32 [[TMP3]], i8 0, i64 32, i1 false)
; CHECK-NEXT:    [[TMP_VAL:%.*]] = load <8 x i64>, <8 x i64>* [[TMP]]
; CHECK-NEXT:    call fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal512_prefer256(<8 x
i64>* [[TMP2]], <8 x i64> [[TMP_VAL]])
; CHECK-NEXT:    [[TMP4:%.*]] = load <8 x i64>, <8 x i64>* [[TMP2]], align 32
; CHECK-NEXT:    store <8 x i64> [[TMP4]], <8 x i64>* [[ARG]], align 2
; CHECK-NEXT:    ret void
;
bb:
 %tmp = alloca <8 x i64>, align 32
 %tmp2 = alloca <8 x i64>, align 32
 %tmp3 = bitcast <8 x i64>* %tmp to i8*
 call void @llvm.memset.p0i8.i64(i8* align 32 %tmp3, i8 0, i64 32, i1 false)
 call fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal512_prefer256(<8 x i64>* %tmp2, <8 x
 i64>* %tmp)
 %tmp4 = load <8 x i64>, <8 x i64>* %tmp2, align 32
```

```
 store <8 x i64> %tmp4, <8 x i64>* %arg, align 2
 ret void
}

; This should promote
define internal fastcc void @callee_avx512_legal512_prefer512_call_avx512_legal512_prefer256(<8 x i64>* %arg,
<8 x i64>* readonly %arg1) #1 {
; CHECK-LABEL: define {{[^@]+}}@callee_avx512_legal512_prefer512_call_avx512_legal512_prefer256
; CHECK-SAME: (<8 x i64>* [[ARG:%.*]], <8 x i64> [[ARG1_VAL:%.*]])
; CHECK-NEXT:  bb:
; CHECK-NEXT:    store <8 x i64> [[ARG1_VAL]], <8 x i64>* [[ARG]]
; CHECK-NEXT:    ret void
;
bb:
 %tmp = load <8 x i64>, <8 x i64>* %arg1
 store <8 x i64> %tmp, <8 x i64>* %arg
 ret void
}

define void @avx512_legal512_prefer512_call_avx512_legal512_prefer256(<8 x i64>* %arg) #0 {
; CHECK-LABEL: define {{[^@]+}}@avx512_legal512_prefer512_call_avx512_legal512_prefer256
; CHECK-SAME: (<8 x i64>* [[ARG:%.*]])
; CHECK-NEXT:  bb:
; CHECK-NEXT:    [[TMP:%.*]] = alloca <8 x i64>, align 32
; CHECK-NEXT:    [[TMP2:%.*]] = alloca
 <8 x i64>, align 32
; CHECK-NEXT:    [[TMP3:%.*]] = bitcast <8 x i64>* [[TMP]] to i8*
; CHECK-NEXT:    call void @llvm.memset.p0i8.i64(i8* align 32 [[TMP3]], i8 0, i64 32, i1 false)
; CHECK-NEXT:    [[TMP_VAL:%.*]] = load <8 x i64>, <8 x i64>* [[TMP]]
; CHECK-NEXT:    call fastcc void @callee_avx512_legal512_prefer512_call_avx512_legal512_prefer256(<8 x
i64>* [[TMP2]], <8 x i64> [[TMP_VAL]])
; CHECK-NEXT:    [[TMP4:%.*]] = load <8 x i64>, <8 x i64>* [[TMP2]], align 32
; CHECK-NEXT:    store <8 x i64> [[TMP4]], <8 x i64>* [[ARG]], align 2
; CHECK-NEXT:    ret void
;
bb:
 %tmp = alloca <8 x i64>, align 32
 %tmp2 = alloca <8 x i64>, align 32
 %tmp3 = bitcast <8 x i64>* %tmp to i8*
 call void @llvm.memset.p0i8.i64(i8* align 32 %tmp3, i8 0, i64 32, i1 false)
 call fastcc void @callee_avx512_legal512_prefer512_call_avx512_legal512_prefer256(<8 x i64>* %tmp2, <8 x
i64>* %tmp)
 %tmp4 = load <8 x i64>, <8 x i64>* %tmp2, align 32
 store <8 x i64> %tmp4, <8 x i64>* %arg, align 2
 ret
 void
}
```

```
; This should promote
define internal fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal512_prefer512(<8 x i64>* %arg,
<8 x i64>* readonly %arg1) #0 {
; CHECK-LABEL: define {{[^@]+}}@callee_avx512_legal512_prefer256_call_avx512_legal512_prefer512
; CHECK-SAME: (<8 x i64>* [[ARG:%.*]], <8 x i64> [[ARG1_VAL:%.*]])
; CHECK-NEXT:  bb:
; CHECK-NEXT:    store <8 x i64> [[ARG1_VAL]], <8 x i64>* [[ARG]]
; CHECK-NEXT:    ret void
;
bb:
 %tmp = load <8 x i64>, <8 x i64>* %arg1
 store <8 x i64> %tmp, <8 x i64>* %arg
 ret void
}

define void @avx512_legal512_prefer256_call_avx512_legal512_prefer512(<8 x i64>* %arg) #1 {
; CHECK-LABEL: define {{[^@]+}}@avx512_legal512_prefer256_call_avx512_legal512_prefer512
; CHECK-SAME: (<8 x i64>* [[ARG:%.*]])
; CHECK-NEXT:  bb:
; CHECK-NEXT:    [[TMP:%.*]] = alloca <8 x i64>, align 32
; CHECK-NEXT:    [[TMP2:%.*]] = alloca <8 x i64>, align 32
; CHECK-NEXT:    [[TMP3:%.*]] = bitcast <8 x i64>* [[TMP]] to i8*
; CHECK-NEXT:    call void @llvm.memset.p0i8.i64(i8*
 align 32 [[TMP3]], i8 0, i64 32, i1 false)
; CHECK-NEXT:    [[TMP_VAL:%.*]] = load <8 x i64>, <8 x i64>* [[TMP]]
; CHECK-NEXT:    call fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal512_prefer512(<8 x
i64>* [[TMP2]], <8 x i64> [[TMP_VAL]])
; CHECK-NEXT:    [[TMP4:%.*]] = load <8 x i64>, <8 x i64>* [[TMP2]], align 32
; CHECK-NEXT:    store <8 x i64> [[TMP4]], <8 x i64>* [[ARG]], align 2
; CHECK-NEXT:    ret void
;
bb:
 %tmp = alloca <8 x i64>, align 32
 %tmp2 = alloca <8 x i64>, align 32
 %tmp3 = bitcast <8 x i64>* %tmp to i8*
 call void @llvm.memset.p0i8.i64(i8* align 32 %tmp3, i8 0, i64 32, i1 false)
 call fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal512_prefer512(<8 x i64>* %tmp2, <8 x
i64>* %tmp)
 %tmp4 = load <8 x i64>, <8 x i64>* %tmp2, align 32
 store <8 x i64> %tmp4, <8 x i64>* %arg, align 2
 ret void
}

; This should not promote
define internal fastcc void @callee_avx512_legal256_prefer256_call_avx512_legal512_prefer256(<8
 x i64>* %arg, <8 x i64>* readonly %arg1) #1 {
; CHECK-LABEL: define {{[^@]+}}@callee_avx512_legal256_prefer256_call_avx512_legal512_prefer256
; CHECK-SAME: (<8 x i64>* [[ARG:%.*]], <8 x i64>* readonly [[ARG1:%.*]])
```

```
; CHECK-NEXT:  bb:
; CHECK-NEXT:    [[TMP:%.*]] = load <8 x i64>, <8 x i64>* [[ARG1]]
; CHECK-NEXT:    store <8 x i64> [[TMP]], <8 x i64>* [[ARG]]
; CHECK-NEXT:    ret void
;
bb:
 %tmp = load <8 x i64>, <8 x i64>* %arg1
 store <8 x i64> %tmp, <8 x i64>* %arg
 ret void
}

define void @avx512_legal256_prefer256_call_avx512_legal512_prefer256(<8 x i64>* %arg) #2 {
; CHECK-LABEL: define {{[^@]+}}@avx512_legal256_prefer256_call_avx512_legal512_prefer256
; CHECK-SAME: (<8 x i64>* [[ARG:%.*]])
; CHECK-NEXT:  bb:
; CHECK-NEXT:    [[TMP:%.*]] = alloca <8 x i64>, align 32
; CHECK-NEXT:    [[TMP2:%.*]] = alloca <8 x i64>, align 32
; CHECK-NEXT:    [[TMP3:%.*]] = bitcast <8 x i64>* [[TMP]] to i8*
; CHECK-NEXT:    call void @llvm.memset.p0i8.i64(i8* align 32 [[TMP3]], i8 0,
 i64 32, i1 false)
; CHECK-NEXT:    call fastcc void @callee_avx512_legal256_prefer256_call_avx512_legal512_prefer256(<8 x
i64>* [[TMP2]], <8 x i64>* [[TMP]])
; CHECK-NEXT:    [[TMP4:%.*]] = load <8 x i64>, <8 x i64>* [[TMP2]], align 32
; CHECK-NEXT:    store <8 x i64> [[TMP4]], <8 x i64>* [[ARG]], align 2
; CHECK-NEXT:    ret void
;
bb:
 %tmp = alloca <8 x i64>, align 32
 %tmp2 = alloca <8 x i64>, align 32
 %tmp3 = bitcast <8 x i64>* %tmp to i8*
 call void @llvm.memset.p0i8.i64(i8* align 32 %tmp3, i8 0, i64 32, i1 false)
 call fastcc void @callee_avx512_legal256_prefer256_call_avx512_legal512_prefer256(<8 x i64>* %tmp2, <8 x
i64>* %tmp)
 %tmp4 = load <8 x i64>, <8 x i64>* %tmp2, align 32
 store <8 x i64> %tmp4, <8 x i64>* %arg, align 2
 ret void
}

; This should not promote
define internal fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal256_prefer256(<8 x i64>* %arg,
<8 x i64>* readonly %arg1) #2 {
; CHECK-LABEL: define {{[^@]+}}@callee_avx512_legal512_prefer256_call_avx512_legal256_prefer256
;
 CHECK-SAME: (<8 x i64>* [[ARG:%.*]], <8 x i64>* readonly [[ARG1:%.*]])
; CHECK-NEXT:  bb:
; CHECK-NEXT:    [[TMP:%.*]] = load <8 x i64>, <8 x i64>* [[ARG1]]
; CHECK-NEXT:    store <8 x i64> [[TMP]], <8 x i64>* [[ARG]]
; CHECK-NEXT:    ret void
```

```
;
bb:
 %tmp = load <8 x i64>, <8 x i64>* %arg1
 store <8 x i64> %tmp, <8 x i64>* %arg
 ret void
}

define void @avx512_legal512_prefer256_call_avx512_legal256_prefer256(<8 x i64>* %arg) #1 {
; CHECK-LABEL: define {{[^@]+}}@avx512_legal512_prefer256_call_avx512_legal256_prefer256
; CHECK-SAME: (<8 x i64>* [[ARG:%.*]])
; CHECK-NEXT:  bb:
; CHECK-NEXT:    [[TMP:%.*]] = alloca <8 x i64>, align 32
; CHECK-NEXT:    [[TMP2:%.*]] = alloca <8 x i64>, align 32
; CHECK-NEXT:    [[TMP3:%.*]] = bitcast <8 x i64>* [[TMP]] to i8*
; CHECK-NEXT:    call void @llvm.memset.p0i8.i64(i8* align 32 [[TMP3]], i8 0, i64 32, i1 false)
; CHECK-NEXT:    call fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal256_prefer256(<8
 x i64>* [[TMP2]], <8 x i64>* [[TMP]])
; CHECK-NEXT:    [[TMP4:%.*]] = load <8 x i64>, <8 x i64>* [[TMP2]], align 32
; CHECK-NEXT:    store <8 x i64> [[TMP4]], <8 x i64>* [[ARG]], align 2
; CHECK-NEXT:    ret void
;
bb:
 %tmp = alloca <8 x i64>, align 32
 %tmp2 = alloca <8 x i64>, align 32
 %tmp3 = bitcast <8 x i64>* %tmp to i8*
 call void @llvm.memset.p0i8.i64(i8* align 32 %tmp3, i8 0, i64 32, i1 false)
 call fastcc void @callee_avx512_legal512_prefer256_call_avx512_legal256_prefer256(<8 x i64>* %tmp2, <8 x
i64>* %tmp)
 %tmp4 = load <8 x i64>, <8 x i64>* %tmp2, align 32
 store <8 x i64> %tmp4, <8 x i64>* %arg, align 2
 ret void
}

; This should promote
define internal fastcc void @callee_avx2_legal256_prefer256_call_avx2_legal512_prefer256(<8 x i64>* %arg, <8
x i64>* readonly %arg1) #3 {
; CHECK-LABEL: define {{[^@]+}}@callee_avx2_legal256_prefer256_call_avx2_legal512_prefer256
; CHECK-SAME: (<8 x i64>* [[ARG:%.*]], <8 x i64> [[ARG1_VAL:%.*]])
; CHECK-NEXT:  bb:
; CHECK-NEXT:
    store <8 x i64> [[ARG1_VAL]], <8 x i64>* [[ARG]]
; CHECK-NEXT:    ret void
;
bb:
 %tmp = load <8 x i64>, <8 x i64>* %arg1
 store <8 x i64> %tmp, <8 x i64>* %arg
 ret void
}
```

```
define void @avx2_legal256_prefer256_call_avx2_legal512_prefer256(<8 x i64>* %arg) #4 {
; CHECK-LABEL: define {{[^@]+}}@avx2_legal256_prefer256_call_avx2_legal512_prefer256
; CHECK-SAME: (<8 x i64>* [[ARG:%.*]])
; CHECK-NEXT:  bb:
; CHECK-NEXT:    [[TMP:%.*]] = alloca <8 x i64>, align 32
; CHECK-NEXT:    [[TMP2:%.*]] = alloca <8 x i64>, align 32
; CHECK-NEXT:    [[TMP3:%.*]] = bitcast <8 x i64>* [[TMP]] to i8*
; CHECK-NEXT:    call void @llvm.memset.p0i8.i64(i8* align 32 [[TMP3]], i8 0, i64 32, i1 false)
; CHECK-NEXT:    [[TMP_VAL:%.*]] = load <8 x i64>, <8 x i64>* [[TMP]]
; CHECK-NEXT:    call fastcc void @callee_avx2_legal256_prefer256_call_avx2_legal512_prefer256(<8 x i64>*
[[TMP2]], <8 x i64> [[TMP_VAL]])
; CHECK-NEXT:    [[TMP4:%.*]] = load <8 x i64>, <8 x i64>* [[TMP2]], align 32
; CHECK-NEXT:    store
 <8 x i64> [[TMP4]], <8 x i64>* [[ARG]], align 2
; CHECK-NEXT:    ret void
;
bb:
 %tmp = alloca <8 x i64>, align 32
 %tmp2 = alloca <8 x i64>, align 32
 %tmp3 = bitcast <8 x i64>* %tmp to i8*
 call void @llvm.memset.p0i8.i64(i8* align 32 %tmp3, i8 0, i64 32, i1 false)
 call fastcc void @callee_avx2_legal256_prefer256_call_avx2_legal512_prefer256(<8 x i64>* %tmp2, <8 x i64>*
%tmp)
 %tmp4 = load <8 x i64>, <8 x i64>* %tmp2, align 32
 store <8 x i64> %tmp4, <8 x i64>* %arg, align 2
 ret void
}

; This should promote
define internal fastcc void @callee_avx2_legal512_prefer256_call_avx2_legal256_prefer256(<8 x i64>* %arg, <8
x i64>* readonly %arg1) #4 {
; CHECK-LABEL: define {{[^@]+}}@callee_avx2_legal512_prefer256_call_avx2_legal256_prefer256
; CHECK-SAME: (<8 x i64>* [[ARG:%.*]], <8 x i64> [[ARG1_VAL:%.*]])
; CHECK-NEXT:  bb:
; CHECK-NEXT:    store <8 x i64> [[ARG1_VAL]], <8 x i64>* [[ARG]]
; CHECK-NEXT:    ret void
;
bb:
 %tmp = load <8 x i64>, <8 x i64>* %arg1
 store <8 x i64>
 %tmp, <8 x i64>* %arg
 ret void
}

define void @avx2_legal512_prefer256_call_avx2_legal256_prefer256(<8 x i64>* %arg) #3 {
; CHECK-LABEL: define {{[^@]+}}@avx2_legal512_prefer256_call_avx2_legal256_prefer256
; CHECK-SAME: (<8 x i64>* [[ARG:%.*]])
```

```
; CHECK-NEXT:  bb:
; CHECK-NEXT:    [[TMP:%.*]] = alloca <8 x i64>, align 32
; CHECK-NEXT:    [[TMP2:%.*]] = alloca <8 x i64>, align 32
; CHECK-NEXT:    [[TMP3:%.*]] = bitcast <8 x i64>* [[TMP]] to i8*
; CHECK-NEXT:    call void @llvm.memset.p0i8.i64(i8* align 32 [[TMP3]], i8 0, i64 32, i1 false)
; CHECK-NEXT:    [[TMP_VAL:%.*]] = load <8 x i64>, <8 x i64>* [[TMP]]
; CHECK-NEXT:    call fastcc void @callee_avx2_legal512_prefer256_call_avx2_legal256_prefer256(<8 x i64>*
[[TMP2]], <8 x i64> [[TMP_VAL]])
; CHECK-NEXT:    [[TMP4:%.*]] = load <8 x i64>, <8 x i64>* [[TMP2]], align 32
; CHECK-NEXT:    store <8 x i64> [[TMP4]], <8 x i64>* [[ARG]], align 2
; CHECK-NEXT:    ret void
;
bb:
 %tmp = alloca <8 x i64>, align 32
 %tmp2 = alloca <8 x i64>,
 align 32
 %tmp3 = bitcast <8 x i64>* %tmp to i8*
 call void @llvm.memset.p0i8.i64(i8* align 32 %tmp3, i8 0, i64 32, i1 false)
 call fastcc void @callee_avx2_legal512_prefer256_call_avx2_legal256_prefer256(<8 x i64>* %tmp2, <8 x i64>*
%tmp)
 %tmp4 = load <8 x i64>, <8 x i64>* %tmp2, align 32
 store <8 x i64> %tmp4, <8 x i64>* %arg, align 2
 ret void
}

; If the arguments are scalar, its ok to promote.
define internal i32 @scalar_callee_avx512_legal256_prefer256_call_avx512_legal512_prefer256(i32* %X, i32*
%Y) #2 {
; CHECK-LABEL: define
{{[^@]+}}@scalar_callee_avx512_legal256_prefer256_call_avx512_legal512_prefer256
; CHECK-SAME: (i32 [[X_VAL:%.*]], i32 [[Y_VAL:%.*]])
; CHECK-NEXT:    [[C:%.*]] = add i32 [[X_VAL]], [[Y_VAL]]
; CHECK-NEXT:    ret i32 [[C]]
;
 %A = load i32, i32* %X
 %B = load i32, i32* %Y
 %C = add i32 %A, %B
 ret i32 %C
}

define i32 @scalar_avx512_legal256_prefer256_call_avx512_legal512_prefer256(i32* %B) #2 {
; CHECK-LABEL: define {{[^@]+}}@scalar_avx512_legal256_prefer256_call_avx512_legal512_prefer256
;
 CHECK-SAME: (i32* [[B:%.*]])
; CHECK-NEXT:    [[A:%.*]] = alloca i32
; CHECK-NEXT:    store i32 1, i32* [[A]]
; CHECK-NEXT:    [[A_VAL:%.*]] = load i32, i32* [[A]]
; CHECK-NEXT:    [[B_VAL:%.*]] = load i32, i32* [[B]]
```

```
; CHECK-NEXT:    [[C:%.*]] = call i32
@scalar_callee_avx512_legal256_prefer256_call_avx512_legal512_prefer256(i32 [[A_VAL]], i32 [[B_VAL]])
; CHECK-NEXT:    ret i32 [[C]]
;
 %A = alloca i32
 store i32 1, i32* %A
 %C = call i32 @scalar_callee_avx512_legal256_prefer256_call_avx512_legal512_prefer256(i32* %A, i32* %B)
 ret i32 %C
}

; If the arguments are scalar, its ok to promote.
define internal i32 @scalar_callee_avx512_legal512_prefer256_call_avx512_legal256_prefer256(i32* %X, i32*
%Y) #2 {
; CHECK-LABEL: define
{{[^@]+}}@scalar_callee_avx512_legal512_prefer256_call_avx512_legal256_prefer256
; CHECK-SAME: (i32 [[X_VAL:%.*]], i32 [[Y_VAL:%.*]])
; CHECK-NEXT:    [[C:%.*]] = add i32 [[X_VAL]], [[Y_VAL]]
; CHECK-NEXT:    ret
 i32 [[C]]
;
 %A = load i32, i32* %X
 %B = load i32, i32* %Y
 %C = add i32 %A, %B
 ret i32 %C
}

define i32 @scalar_avx512_legal512_prefer256_call_avx512_legal256_prefer256(i32* %B) #2 {
; CHECK-LABEL: define {{[^@]+}}@scalar_avx512_legal512_prefer256_call_avx512_legal256_prefer256
; CHECK-SAME: (i32* [[B:%.*]])
; CHECK-NEXT:    [[A:%.*]] = alloca i32
; CHECK-NEXT:    store i32 1, i32* [[A]]
; CHECK-NEXT:    [[A_VAL:%.*]] = load i32, i32* [[A]]
; CHECK-NEXT:    [[B_VAL:%.*]] = load i32, i32* [[B]]
; CHECK-NEXT:    [[C:%.*]] = call i32
@scalar_callee_avx512_legal512_prefer256_call_avx512_legal256_prefer256(i32 [[A_VAL]], i32 [[B_VAL]])
; CHECK-NEXT:    ret i32 [[C]]
;
 %A = alloca i32
 store i32 1, i32* %A
 %C = call i32 @scalar_callee_avx512_legal512_prefer256_call_avx512_legal256_prefer256(i32* %A, i32* %B)
 ret i32 %C
}

; Function Attrs: argmemonly nounwind
declare void @llvm.memset.p0i8.i64(i8* nocapture writeonly, i8, i64, i1) #5

attributes #0 = { inlinehint norecurse nounwind
```

uwtable "target-features"="+avx512vl" "min-legal-vector-width"="512" "prefer-vector-width"="512" }
attributes #1 = { inlinehint norecurse nounwind uwtable "target-features"="+avx512vl" "min-legal-vector-width"="512" "prefer-vector-width"="256" }
attributes #2 = { inlinehint norecurse nounwind uwtable "target-features"="+avx512vl" "min-legal-vector-width"="256" "prefer-vector-width"="256" }
attributes #3 = { inlinehint norecurse nounwind uwtable "target-features"="+avx2" "min-legal-vector-width"="512" "prefer-vector-width"="256" }
attributes #4 = { inlinehint norecurse nounwind uwtable "target-features"="+avx2" "min-legal-vector-width"="256" "prefer-vector-width"="256" }
attributes #5 = { argmemonly nounwind }
Copyright (c) 2010-2015 Benjamin Peterson

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
==
The LLVM Project is under the Apache License v2.0 with LLVM Exceptions:
==

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all

other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the
power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached
to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative
Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work

includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional
attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License
does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including
negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such
obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

---- LLVM Exceptions to the Apache 2.0 License ----

As an exception, if, as a result of your compiling your source code, portions
of this Software are embedded into an Object form of such source code, you
may redistribute such embedded portions in such Object form without complying
with the conditions of Sections 4(a), 4(b) and 4(d) of the License.

In addition, if you combine or link compiled
forms of this Software with
software that is licensed under the GPLv2 ("Combined Software") and if a
court of competent jurisdiction determines that the patent provision (Section
3), the indemnity provision (Section 9) or other Section of the License
conflicts with the conditions of the GPLv2, you may retroactively and
prospectively choose to deem waived or otherwise exclude such Section(s) of
the License, but only in their entirety and only with respect to the Combined
Software.

==
Software from third parties included in the LLVM Project:
==
The LLVM Project contains third party software which is under different license
terms. All such code will be identified clearly using at least one of two
mechanisms:
1) It will be in a separate directory tree with its own `LICENSE.txt` or
`LICENSE` file at the top containing the specific
license and restrictions
which apply to that software, or
2) It will contain specific license and restriction terms at the top of every
file.

==
Legacy LLVM License (https://llvm.org/docs/DeveloperPolicy.html#legacy):
==
University of Illinois/NCSA
Open Source License

Copyright (c) 2003-2019 University of Illinois at Urbana-Champaign.
All rights reserved.

Developed by:

LLVM Team

University of Illinois at Urbana-Champaign

http://llvm.org

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal with
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies
of the Software, and to permit persons to whom the Software is furnished
to do
so, subject to the following conditions:

* Redistributions of source code must retain the above copyright notice,
this list of conditions and the following disclaimers.

* Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimers in the
documentation and/or other materials provided with the distribution.

* Neither the names of the LLVM Team, University of Illinois at
Urbana-Champaign, nor the names of its contributors may be used to
endorse or promote products derived from this Software without specific
prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE
SOFTWARE.
==
The LLVM Project is under the Apache License v2.0 with LLVM Exceptions:
==

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the
power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached
to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition,
"submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,

and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of

the Derivative Works; and

(d) If the Work
includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional
attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License
does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A

PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including
negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such
obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

---- LLVM Exceptions to the Apache 2.0 License ----

As an exception, if, as a result of your compiling your source code, portions
of this Software are embedded into an Object form of such source code, you
may redistribute such embedded portions in such Object form without complying
with the conditions of Sections 4(a), 4(b) and 4(d) of the License.

In addition, if you combine or link compiled
forms of this Software with
software that is licensed under the GPLv2 ("Combined Software") and if a
court of competent jurisdiction determines that the patent provision (Section
3), the indemnity provision (Section 9) or other Section of the License
conflicts with the conditions of the GPLv2, you may retroactively and
prospectively choose to deem waived or otherwise exclude such Section(s) of
the License, but only in their entirety and only with respect to the Combined
Software.

==
Software from third parties included in the LLVM Project:
==
The LLVM Project contains third party software which is under different license
terms. All such code will be identified clearly using at least one of two
mechanisms:
1) It will be in a separate directory tree with its own `LICENSE.txt` or
`LICENSE` file at the top containing the specific
license and restrictions
which apply to that software, or
2) It will contain specific license and restriction terms at the top of every
file.

==
Legacy LLVM License (https://llvm.org/docs/DeveloperPolicy.html#legacy):
==
University of Illinois/NCSA
Open Source License

Copyright (c) 2010 Apple Inc.
All rights reserved.

Developed by:

LLDB Team

http://lldb.llvm.org/

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal with
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies
of the Software, and to permit persons to whom the Software is furnished to do
so, subject to the following conditions:

* Redistributions of
source code must retain the above copyright notice,
this list of conditions and the following disclaimers.

* Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimers in the
documentation and/or other materials provided with the distribution.

* Neither the names of the LLDB Team, copyright holders, nor the names of
its contributors may be used to endorse or promote products derived from
this Software without specific prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
DEALINGS WITH THE
SOFTWARE.
ISC LICENSE

This license is approved by the OSI and FSF as GPL-compatible.
http://opensource.org/licenses/isc-license.txt

Copyright (c) 2013-2014, Pexpect development team
Copyright (c) 2012, Noah Spurrier <noah@noah.org>

Permission to use, copy, modify, and/or distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR

ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

---

clang-tidy High-Integrity C++ Files

All clang-tidy files are licensed under the same terms as the rest of the LLVM project with the following additions:

Any file referencing a High-Integrity C++ Coding guideline:

HIC++ Coding Standard as created by PRQA.

Please see http://www.codingstandard.com/section/conditions-of-use/ for more information.

MIT License (MIT)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE
USE OR OTHER DEALINGS IN THE
SOFTWARE.

==========

The LLVM Project is under the Apache License v2.0 with LLVM Exceptions:

==========

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the
power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached
to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition,

"submitted"
means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work
includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional
attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License
does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each

Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including
negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such
obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

---- LLVM Exceptions to the Apache 2.0 License ----

As an exception, if, as a result of your compiling your source code, portions
of this Software are embedded into an Object form of such source code, you
may redistribute such embedded portions in such Object form without complying
with the conditions of Sections 4(a), 4(b) and 4(d) of the License.

In addition, if you combine or link compiled
forms of this Software with
software that is licensed under the GPLv2 ("Combined Software") and if a
court of competent jurisdiction determines that the patent provision (Section
3), the indemnity provision (Section 9) or other Section of the License
conflicts with the conditions of the GPLv2, you may retroactively and
prospectively choose to deem waived or otherwise exclude such Section(s) of
the License, but only in their entirety and only with respect to the Combined
Software.

==
Software from third parties included in the LLVM Project:
==
The LLVM Project contains third party software which is under different license
terms. All such code will be identified clearly using at least one of two
mechanisms:
1) It will be in a separate directory tree with its own `LICENSE.txt` or
`LICENSE` file at the top containing the specific
license and restrictions
which apply to that software, or
2) It will contain specific license and restriction terms at the top of every
file.

==
Legacy LLVM License (https://llvm.org/docs/DeveloperPolicy.html#legacy):
==

The libc++ library is dual licensed under both the University of Illinois

"BSD-Like" license and the MIT license. As a user of this code you may choose
to use it under either license. As a contributor, you agree to allow your code
to be used under both.

Full text of the relevant licenses is included below.

==

University of Illinois/NCSA
Open Source License

Copyright (c) 2009-2019 by the contributors listed in CREDITS.TXT

All rights reserved.

Developed by:

  LLVM Team

  University of Illinois at Urbana-Champaign

  http://llvm.org

Permission
 is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal with
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies
of the Software, and to permit persons to whom the Software is furnished to do
so, subject to the following conditions:

  * Redistributions of source code must retain the above copyright notice,
    this list of conditions and the following disclaimers.

  * Redistributions in binary form must reproduce the above copyright notice,
    this list of conditions and the following disclaimers in the
    documentation and/or other materials provided with the distribution.

  * Neither the names of the LLVM Team, University of Illinois at
    Urbana-Champaign, nor the names of its contributors may be used to
    endorse or promote products derived from this
 Software without specific
    prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER

LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

==========

Copyright (c) 2009-2014 by the contributors listed in CREDITS.TXT

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software,
and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
; NOTE: Assertions have been autogenerated by utils/update_test_checks.py UTC_ARGS: --include-generated-funcs
; RUN: opt -S -verify -iroutliner -ir-outlining-no-cost < %s | FileCheck %s

; This test checks that debug info is recognized as able to be extracted along
; with the other instructions, but is not included in the consolidated function.

```
define void @function1() !dbg !6 {
entry:
 %a = alloca i32, align 4, !dbg !17
 call void @llvm.dbg.value(metadata i32* %a, metadata !9, metadata !DIExpression()), !dbg !17
 %b = alloca i32, align 4, !dbg !18
 call void @llvm.dbg.value(metadata i32* %b, metadata !11, metadata !DIExpression()), !dbg !18
 %c = alloca i32, align 4, !dbg !19
 call void @llvm.dbg.value(metadata i32* %c, metadata !12, metadata !DIExpression()), !dbg !19
 store i32 2, i32* %a, align 4, !dbg !20
 store i32 3, i32* %b, align 4, !dbg !21
 store i32 4, i32* %c, align 4, !dbg !22
 %al = load i32, i32* %a, align 4, !dbg !23
 call void @llvm.dbg.value(metadata i32 %al,
 metadata !13, metadata !DIExpression()), !dbg !23
 %bl = load i32, i32* %b, align 4, !dbg !24
```

```
 call void @llvm.dbg.value(metadata i32 %bl, metadata !15, metadata !DIExpression()), !dbg !24
 %cl = load i32, i32* %c, align 4, !dbg !25
 call void @llvm.dbg.value(metadata i32 %cl, metadata !16, metadata !DIExpression()), !dbg !25
 ret void, !dbg !26
}

define void @function2() !dbg !27 {
entry:
 %a = alloca i32, align 4, !dbg !35
 call void @llvm.dbg.value(metadata i32* %a, metadata !29, metadata !DIExpression()), !dbg !35
 %b = alloca i32, align 4, !dbg !36
 call void @llvm.dbg.value(metadata i32* %b, metadata !30, metadata !DIExpression()), !dbg !36
 %c = alloca i32, align 4, !dbg !37
 call void @llvm.dbg.value(metadata i32* %c, metadata !31, metadata !DIExpression()), !dbg !37
 store i32 2, i32* %a, align 4, !dbg !38
 store i32 3, i32* %b, align 4, !dbg !39
 store i32 4, i32* %c, align 4, !dbg !40
 %al = load i32, i32* %a, align 4, !dbg !41
 call void @llvm.dbg.value(metadata
 i32 %al, metadata !32, metadata !DIExpression()), !dbg !41
 %bl = load i32, i32* %b, align 4, !dbg !42
 call void @llvm.dbg.value(metadata i32 %bl, metadata !33, metadata !DIExpression()), !dbg !42
 %cl = load i32, i32* %c, align 4, !dbg !43
 call void @llvm.dbg.value(metadata i32 %cl, metadata !34, metadata !DIExpression()), !dbg !43
 ret void, !dbg !44
}


; Function Attrs: nounwind readnone speculatable willreturn
declare void @llvm.dbg.value(metadata, metadata, metadata) #0

attributes #0 = { nounwind readnone speculatable willreturn }

!llvm.dbg.cu = !{!0}
!llvm.debugify = !{!3, !4}
!llvm.module.flags = !{!5}

!0 = distinct !DICompileUnit(language: DW_LANG_C, file: !1, producer: "debugify", isOptimized: true,
runtimeVersion: 0, emissionKind: FullDebug, enums: !2)
!1 = !DIFile(filename: "legal-debug.ll", directory: "/")
!2 = !{}
!3 = !{i32 20}
!4 = !{i32 12}
!5 = !{i32 2, !"Debug Info Version", i32 3}
!6 = distinct !DISubprogram(name: "function1", linkageName: "function1",
 scope: null, file: !1, line: 1, type: !7, scopeLine: 1, spFlags: DISPFlagDefinition | DISPFlagOptimized, unit: !0,
retainedNodes: !8)
!7 = !DISubroutineType(types: !2)
```

```
!8 = !{!9, !11, !12, !13, !15, !16}
!9 = !DILocalVariable(name: "1", scope: !6, file: !1, line: 1, type: !10)
!10 = !DIBasicType(name: "ty64", size: 64, encoding: DW_ATE_unsigned)
!11 = !DILocalVariable(name: "2", scope: !6, file: !1, line: 2, type: !10)
!12 = !DILocalVariable(name: "3", scope: !6, file: !1, line: 3, type: !10)
!13 = !DILocalVariable(name: "4", scope: !6, file: !1, line: 7, type: !14)
!14 = !DIBasicType(name: "ty32", size: 32, encoding: DW_ATE_unsigned)
!15 = !DILocalVariable(name: "5", scope: !6, file: !1, line: 8, type: !14)
!16 = !DILocalVariable(name: "6", scope: !6, file: !1, line: 9, type: !14)
!17 = !DILocation(line: 1, column: 1, scope: !6)
!18 = !DILocation(line: 2, column: 1, scope: !6)
!19 = !DILocation(line: 3, column: 1, scope: !6)
!20 = !DILocation(line: 4, column: 1, scope: !6)
!21
 = !DILocation(line: 5, column: 1, scope: !6)
!22 = !DILocation(line: 6, column: 1, scope: !6)
!23 = !DILocation(line: 7, column: 1, scope: !6)
!24 = !DILocation(line: 8, column: 1, scope: !6)
!25 = !DILocation(line: 9, column: 1, scope: !6)
!26 = !DILocation(line: 10, column: 1, scope: !6)
!27 = distinct !DISubprogram(name: "function2", linkageName: "function2", scope: null, file: !1, line: 11, type: !7,
scopeLine: 11, spFlags: DISPFlagDefinition | DISPFlagOptimized, unit: !0, retainedNodes: !28)
!28 = !{!29, !30, !31, !32, !33, !34}
!29 = !DILocalVariable(name: "7", scope: !27, file: !1, line: 11, type: !10)
!30 = !DILocalVariable(name: "8", scope: !27, file: !1, line: 12, type: !10)
!31 = !DILocalVariable(name: "9", scope: !27, file: !1, line: 13, type: !10)
!32 = !DILocalVariable(name: "10", scope: !27, file: !1, line: 17, type: !14)
!33 = !DILocalVariable(name: "11", scope: !27, file: !1, line: 18, type: !14)
!34 = !DILocalVariable(name: "12", scope: !27, file: !1, line: 19, type:
 !14)
!35 = !DILocation(line: 11, column: 1, scope: !27)
!36 = !DILocation(line: 12, column: 1, scope: !27)
!37 = !DILocation(line: 13, column: 1, scope: !27)
!38 = !DILocation(line: 14, column: 1, scope: !27)
!39 = !DILocation(line: 15, column: 1, scope: !27)
!40 = !DILocation(line: 16, column: 1, scope: !27)
!41 = !DILocation(line: 17, column: 1, scope: !27)
!42 = !DILocation(line: 18, column: 1, scope: !27)
!43 = !DILocation(line: 19, column: 1, scope: !27)
!44 = !DILocation(line: 20, column: 1, scope: !27)
; CHECK-LABEL: @function1(
; CHECK-NEXT:  entry:
; CHECK-NEXT:    [[A:%.*]] = alloca i32, align 4, !dbg [[DBG17:![0-9]+]]
; CHECK-NEXT:    call void @llvm.dbg.value(metadata i32* [[A]], metadata [[META9:![0-9]+]], metadata
!DIExpression()), !dbg [[DBG17]]
; CHECK-NEXT:    [[B:%.*]] = alloca i32, align 4, !dbg [[DBG18:![0-9]+]]
; CHECK-NEXT:    call void @llvm.dbg.value(metadata i32* [[B]], metadata [[META11:![0-9]+]], metadata
!DIExpression()), !dbg [[DBG18]]
```

```
; CHECK-NEXT:    [[C:%.*]]
 = alloca i32, align 4, !dbg [[DBG19:![0-9]+]]
; CHECK-NEXT:    call void @llvm.dbg.value(metadata i32* [[C]], metadata [[META12:![0-9]+]], metadata
!DIExpression()), !dbg [[DBG19]]
; CHECK-NEXT:    call void @outlined_ir_func_0(i32* [[A]], i32* [[B]], i32* [[C]]), !dbg [[DBG20:![0-9]+]]
; CHECK-NEXT:    ret void, !dbg [[DBG21:![0-9]+]]
;
;
; CHECK-LABEL: @function2(
; CHECK-NEXT:  entry:
; CHECK-NEXT:    [[A:%.*]] = alloca i32, align 4, !dbg [[DBG30:![0-9]+]]
; CHECK-NEXT:    call void @llvm.dbg.value(metadata i32* [[A]], metadata [[META24:![0-9]+]], metadata
!DIExpression()), !dbg [[DBG30]]
; CHECK-NEXT:    [[B:%.*]] = alloca i32, align 4, !dbg [[DBG31:![0-9]+]]
; CHECK-NEXT:    call void @llvm.dbg.value(metadata i32* [[B]], metadata [[META25:![0-9]+]], metadata
!DIExpression()), !dbg [[DBG31]]
; CHECK-NEXT:    [[C:%.*]] = alloca i32, align 4, !dbg [[DBG32:![0-9]+]]
; CHECK-NEXT:    call void @llvm.dbg.value(metadata i32* [[C]], metadata [[META26:![0-9]+]], metadata
!DIExpression()),
 !dbg [[DBG32]]
; CHECK-NEXT:    call void @outlined_ir_func_0(i32* [[A]], i32* [[B]], i32* [[C]]), !dbg [[DBG33:![0-9]+]]
; CHECK-NEXT:    ret void, !dbg [[DBG34:![0-9]+]]
;
;
; CHECK: @outlined_ir_func_0(i32* [[TMP0:%.*]], i32* [[TMP1:%.*]], i32*  [[TMP2:%.*]])
; CHECK:       entry_to_outline:
; CHECK-NEXT:    store i32 2, i32* [[TMP0]], align 4
; CHECK-NEXT:    store i32 3, i32* [[TMP1]], align 4
; CHECK-NEXT:    store i32 4, i32* [[TMP2]], align 4
; CHECK-NEXT:    [[AL:%.*]] = load i32, i32* [[TMP0]], align 4
; CHECK-NEXT:    [[BL:%.*]] = load i32, i32* [[TMP1]], align 4
; CHECK-NEXT:    [[CL:%.*]] = load i32, i32* [[TMP2]], align 4
; CHECK-NEXT:    br label [[ENTRY_AFTER_OUTLINE_EXITSTUB:%.*]]
;
; RUN: llc -O3 -mtriple=powerpc-unknown-linux-gnu -mcpu=e500 -mattr=spe < %s | FileCheck %s

; PowerPC SPE is a rare in-tree target that has the FP_TO_SINT node marked
; as Legal.

; Verify that fptosi(42.1) isn't simplified when the rounding mode is
; unknown.
; Verify that no gross errors happen.
; CHECK-LABEL: @f20
; COMMON: cfdctsiz
define i32 @f20(double %a) strictfp {
entry:
 %result = call i32 @llvm.experimental.constrained.fptosi.i32.f64(double 42.1,
                                               metadata !"fpexcept.strict")
```

strictfp
ret i32 %result
}

@llvm.fp.env = thread_local global i8 zeroinitializer, section "llvm.metadata"
declare i32 @llvm.experimental.constrained.fptosi.i32.f64(double, metadata)
==========
The LLVM Project is under the Apache License v2.0 with LLVM Exceptions:
==========

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the
power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached

to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative
Works of,
publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a

cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work
includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional
attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License
does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including
negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such
obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason

of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

---- LLVM Exceptions to the Apache 2.0 License ----

As an exception, if, as a result of your compiling your source code, portions
of this Software are embedded into an Object form of such source code, you
may redistribute such embedded portions in such Object form without complying
with the conditions of Sections 4(a), 4(b) and 4(d) of the License.

In addition, if you combine or link compiled
forms of this Software with
software that is licensed under the GPLv2 ("Combined Software") and if a
court of competent jurisdiction determines that the patent provision (Section
3), the indemnity provision (Section 9) or other Section of the License
conflicts with the conditions of the GPLv2, you may retroactively and
prospectively choose to deem waived or otherwise exclude such Section(s) of
the License, but only in their entirety and only with respect to the Combined
Software.

==========

Software from third parties included in the LLVM Project:
==
The LLVM Project contains third party software which is under different license
terms. All such code will be identified clearly using at least one of two
mechanisms:
1) It will be in a separate directory tree with its own `LICENSE.txt` or
  `LICENSE` file at the top containing the specific
 license and restrictions
  which apply to that software, or
2) It will contain specific license and restriction terms at the top of every
  file.

==
Legacy LLVM License (https://llvm.org/docs/DeveloperPolicy.html#legacy):
==

The software contained in this directory tree is dual licensed under both the
University of Illinois "BSD-Like" license and the MIT license. As a user of
this code you may choose to use it under either license. As a contributor,
you agree to allow your code to be used under both.

Full text of the relevant licenses is included below.

==

University of Illinois/NCSA
Open Source License

Copyright (c) 2017-2019 by the contributors listed in CREDITS.TXT

All rights reserved.

Developed by:
  Threading Runtimes Team
  Intel Corporation
   http://www.intel.com

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal with
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies
of the Software, and to permit persons to whom the Software is furnished to do
so, subject to the following conditions:

  * Redistributions of source code must retain the above copyright notice,
    this list of conditions and the following disclaimers.

  * Redistributions in binary form must reproduce the above copyright notice,

this list of conditions and the following disclaimers in the
documentation and/or other materials provided with the distribution.

* Neither the names of Intel Corporation Threading Runtimes Team nor the
names of its contributors may be used to endorse or promote products
derived from
this Software without specific prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

==========

Copyright (c) 2017-2019 by the contributors listed in CREDITS.TXT

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software,
and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
Copyright (c) 2019 Intel Corporation. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote

products derived from this software without specific prior written permission.
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
IMath is Copyright 2002-2009 Michael J. Fromberger
You may use it subject to the following Licensing Terms:

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies
of the Software, and to permit persons to whom the Software is furnished to do
so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF
CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
==========
The LLVM Project is under the Apache License v2.0 with LLVM Exceptions:
==========

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the
power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached
to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition,
"submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,

and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of

the Derivative Works; and

(d) If the Work
includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional
attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License
does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A

PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including
negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such
obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

---- LLVM Exceptions to the Apache 2.0 License ----

As an exception, if, as a result of your compiling your source code, portions
of this Software are embedded into an Object form of such source code, you
may redistribute such embedded portions in such Object form without complying
with the conditions of Sections 4(a), 4(b) and 4(d) of the License.

In addition, if you combine or link compiled
forms of this Software with
software that is licensed under the GPLv2 ("Combined Software") and if a
court of competent jurisdiction determines that the patent provision (Section
3), the indemnity provision (Section 9) or other Section of the License
conflicts with the conditions of the GPLv2, you may retroactively and
prospectively choose to deem waived or otherwise exclude such Section(s) of
the License, but only in their entirety and only with respect to the Combined
Software.

==
Software from third parties included in the LLVM Project:
==
The LLVM Project contains third party software which is under different license
terms. All such code will be identified clearly using at least one of two
mechanisms:
1) It will be in a separate directory tree with its own `LICENSE.txt` or
`LICENSE` file at the top containing the specific
license and restrictions
which apply to that software, or
2) It will contain specific license and restriction terms at the top of every
file.

==
Legacy LLVM License (https://llvm.org/docs/DeveloperPolicy.html#legacy):
==

The libc++abi library is dual licensed under both the University of Illinois
"BSD-Like" license and the MIT license. As a user of this code you may choose
to use it under either license. As a contributor, you agree to allow your code
to be used under both.

Full text of the relevant licenses is included below.

==

University of Illinois/NCSA
Open Source License

Copyright (c) 2009-2019 by the contributors listed in CREDITS.TXT

All rights reserved.

Developed by:

LLVM Team

University of Illinois at Urbana-Champaign

http://llvm.org

Permission
is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal with
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies
of the Software, and to permit persons to whom the Software is furnished to do
so, subject to the following conditions:

* Redistributions of source code must retain the above copyright notice,
this list of conditions and the following disclaimers.

* Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimers in the
documentation and/or other materials provided with the distribution.

* Neither the names of the LLVM Team, University of Illinois at
Urbana-Champaign, nor the names of its contributors may be used to
endorse or promote products derived from
this Software without specific
prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

==

Copyright (c) 2009-2014 by the contributors listed in CREDITS.TXT

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the
 Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
add_lldb_library(lldbPluginObjectContainerBSDArchive PLUGIN
 ObjectContainerBSDArchive.cpp

 LINK_LIBS
  lldbCore
  lldbHost
  lldbSymbol
 LINK_COMPONENTS
  Support
)
==
The LLVM Project is under the Apache License v2.0 with LLVM Exceptions:
==

                                 Apache License
                           Version 2.0, January 2004
                        http://www.apache.org/licenses/

   TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

   1. Definitions.

     "License" shall mean the terms and conditions for use, reproduction,
     and distribution as defined by Sections 1 through 9 of this document.

     "Licensor" shall mean the copyright owner or entity authorized by

the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the
power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached
to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition,
"submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the

Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the

appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including
negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such
obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

---- LLVM Exceptions to the Apache 2.0 License ----

As an exception, if, as a result of your compiling your source code, portions
of this Software are embedded into an Object form of such source code, you
may redistribute such embedded portions in such Object form without complying
with the conditions of Sections 4(a), 4(b) and 4(d) of the License.

In addition, if you combine or link compiled
forms of this Software with
software that is licensed under the GPLv2 ("Combined Software") and if a
court of competent jurisdiction determines that the patent provision (Section
3), the indemnity provision (Section 9) or other Section of the License
conflicts with the conditions of the GPLv2, you may retroactively and
prospectively choose to deem waived or otherwise exclude such Section(s) of
the License, but only in their entirety and only with respect to the Combined
Software.

==
Software from third parties included in the LLVM Project:
==
The LLVM Project contains third party software which is under different license
terms. All such code will be identified clearly using at least one of two
mechanisms:
1) It will be in a separate directory tree with its own `LICENSE.txt` or
`LICENSE` file at the top containing the specific
license and restrictions
which apply to that software, or
2) It will contain specific license and restriction terms at the top of every
file.

==
Legacy LLVM License (https://llvm.org/docs/DeveloperPolicy.html#legacy):
==
University of Illinois/NCSA
Open Source License

Copyright (c) 2007-2019 University of Illinois at Urbana-Champaign.
All rights reserved.

Developed by:

LLVM Team

University of Illinois at Urbana-Champaign

http://llvm.org

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal with
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies
of the Software, and to permit persons to whom the Software is furnished
 to do
so, subject to the following conditions:

* Redistributions of source code must retain the above copyright notice,
  this list of conditions and the following disclaimers.

* Redistributions in binary form must reproduce the above copyright notice,
  this list of conditions and the following disclaimers in the
  documentation and/or other materials provided with the distribution.

* Neither the names of the LLVM Team, University of Illinois at
  Urbana-Champaign, nor the names of its contributors may be used to
  endorse or promote products derived from this Software without specific
  prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION
 OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE
SOFTWARE.
; RUN: llc -march=hexagon -hexagon-hvx-widen=32 < %s | FileCheck %s

; Truncating a type-to-be-widenened to a legal type (v8i8).
; Check that this compiles successfully.
; CHECK-LABEL: f0:
; CHECK: dealloc_return

target datalayout = "e-m:e-p:32:32:32-a:0-n16:32-i64:64:64-i32:32:32-i16:16:16-i1:8:8-f32:32:32-f64:64:64-
v32:32:32-v64:64:64-v512:512:512-v1024:1024:1024-v2048:2048:2048"
target triple = "hexagon"

define dllexport void @f0(i8* %a0) local_unnamed_addr #0 {
b0:

```
 %v0 = load i8, i8* undef, align 1
 %v1 = zext i8 %v0 to i16
 %v2 = add i16 0, %v1
 %v3 = icmp sgt i16 %v2, 1
 %v4 = select i1 %v3, i16 %v2, i16 1
 %v5 = udiv i16 -32768, %v4
 %v6 = zext i16 %v5 to i32
 %v7 = insertelement <8 x i32> undef, i32 %v6, i32 0
 %v8 = shufflevector <8 x i32> %v7, <8 x i32> undef, <8 x i32> zeroinitializer
 %v9 = load <8 x i16>, <8 x i16>* undef, align 2
 %v10 = sext <8 x i16> %v9 to <8 x i32>
 %v11 = mul nsw <8 x i32> %v8, %v10
 %v12 = add nsw <8 x i32> %v11, <i32 16384, i32 16384,
 i32 16384, i32 16384, i32 16384, i32 16384, i32 16384, i32 16384>
 %v13 = lshr <8 x i32> %v12, <i32 15, i32 15, i32 15, i32 15, i32 15, i32 15, i32 15, i32 15>
 %v14 = trunc <8 x i32> %v13 to <8 x i8>
 %v15 = getelementptr inbounds i8, i8* %a0, i32 undef
 %v16 = bitcast i8* %v15 to <8 x i8>*
 store <8 x i8> %v14, <8 x i8>* %v16, align 1
 ret void
}

attributes #0 = { "target-features"="+hvx,+hvx-length128b" }
; NOTE: Assertions have been autogenerated by utils/update_analyze_test_checks.py
; RUN: opt < %s  -cost-model -analyze -mtriple=x86_64-apple-macosx10.8.0 -mattr=+avx2 | FileCheck %s --check-
prefixes=VEC256,AVX
; RUN: opt < %s  -cost-model -analyze -mtriple=x86_64-apple-macosx10.8.0 -mattr=+avx512vl,+prefer-256-bit |
FileCheck %s --check-prefixes=VEC256,AVX512VL256
; RUN: opt < %s  -cost-model -analyze -mtriple=x86_64-apple-macosx10.8.0 -mattr=+avx512vl,-prefer-256-bit |
FileCheck %s --check-prefixes=AVX512VL512
; RUN: opt < %s  -cost-model -analyze -mtriple=x86_64-apple-macosx10.8.0 -
mattr=+avx512vl,+avx512bw,+avx512dq,+prefer-256-bit | FileCheck %s --check-prefixes=VEC256,SKX256
; RUN: opt < %s  -cost-model -analyze -mtriple=x86_64-apple-macosx10.8.0 -
mattr=+avx512vl,+avx512bw,+avx512dq,-prefer-256-bit | FileCheck %s --check-prefixes=SKX512

define void @zext256() "min-legal-vector-width"="256" {
; AVX-LABEL: 'zext256'
; AVX-NEXT:  Cost Model: Found an estimated cost of 4 for instruction:
 %A = zext <8 x i16> undef to <8 x i64>
; AVX-NEXT:  Cost Model: Found an estimated cost of 5 for instruction: %B = zext <8 x i32> undef to <8 x i64>
; AVX-NEXT:  Cost Model: Found an estimated cost of 4 for instruction: %C = zext <16 x i8> undef to <16 x i32>
; AVX-NEXT:  Cost Model: Found an estimated cost of 3 for instruction: %D = zext <16 x i16> undef to <16 x
i32>
; AVX-NEXT:  Cost Model: Found an estimated cost of 5 for instruction: %E = zext <32 x i8> undef to <32 x i16>
; AVX-NEXT:  Cost Model: Found an estimated cost of 0 for instruction: ret void
;
; AVX512VL256-LABEL: 'zext256'
; AVX512VL256-NEXT:  Cost Model: Found an estimated cost of 2 for instruction: %A = zext <8 x i16> undef to
```

<8 x i64>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %B = zext <8 x i32> undef to <8 x i64>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %C = zext <16 x i8> undef to <16 x i32>
; AVX512VL256-NEXT: Cost Model: Found an estimated
 cost of 3 for instruction: %D = zext <16 x i16> undef to <16 x i32>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %E = zext <32 x i8> undef to <32 x i16>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
; AVX512VL512-LABEL: 'zext256'
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %A = zext <8 x i16> undef to <8 x i64>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %B = zext <8 x i32> undef to <8 x i64>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %C = zext <16 x i8> undef to <16 x i32>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %D = zext <16 x i16> undef to <16 x i32>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %E = zext <32 x i8> undef to <32 x i16>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
;
 SKX256-LABEL: 'zext256'
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %A = zext <8 x i16> undef to <8 x i64>
; SKX256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %B = zext <8 x i32> undef to <8 x i64>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %C = zext <16 x i8> undef to <16 x i32>
; SKX256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %D = zext <16 x i16> undef to <16 x i32>
; SKX256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %E = zext <32 x i8> undef to <32 x i16>
; SKX256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
; SKX512-LABEL: 'zext256'
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %A = zext <8 x i16> undef to <8 x i64>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %B = zext <8 x i32> undef to <8 x i64>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction:
 %C = zext <16 x i8> undef to <16 x i32>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %D = zext <16 x i16> undef to <16 x i32>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %E = zext <32 x i8> undef to <32 x

```
i16>
; SKX512-NEXT:  Cost Model: Found an estimated cost of 0 for instruction: ret void
;
 %A = zext <8 x i16> undef to <8 x i64>
 %B = zext <8 x i32> undef to <8 x i64>
 %C = zext <16 x i8> undef to <16 x i32>
 %D = zext <16 x i16> undef to <16 x i32>
 %E = zext <32 x i8> undef to <32 x i16>
 ret void
}

define void @zext512() "min-legal-vector-width"="512" {
; AVX-LABEL: 'zext512'
; AVX-NEXT:  Cost Model: Found an estimated cost of 4 for instruction: %A = zext <8 x i16> undef to <8 x i64>
; AVX-NEXT:  Cost Model: Found an estimated cost of 5 for instruction: %B = zext <8 x i32> undef to <8 x i64>
; AVX-NEXT:  Cost Model: Found an estimated cost of 4 for instruction: %C = zext <16 x i8> undef to <16 x i32>
; AVX-NEXT:
  Cost Model: Found an estimated cost of 3 for instruction: %D = zext <16 x i16> undef to <16 x i32>
; AVX-NEXT:  Cost Model: Found an estimated cost of 5 for instruction: %E = zext <32 x i8> undef to <32 x i16>
; AVX-NEXT:  Cost Model: Found an estimated cost of 0 for instruction: ret void
;
; AVX512VL256-LABEL: 'zext512'
; AVX512VL256-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %A = zext <8 x i16> undef to
<8 x i64>
; AVX512VL256-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %B = zext <8 x i32> undef to
<8 x i64>
; AVX512VL256-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %C = zext <16 x i8> undef to
<16 x i32>
; AVX512VL256-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %D = zext <16 x i16> undef to
<16 x i32>
; AVX512VL256-NEXT:  Cost Model: Found an estimated cost of 3 for instruction: %E = zext <32 x i8> undef to
<32 x i16>
; AVX512VL256-NEXT:  Cost Model: Found an estimated cost of 0 for instruction:
 ret void
;
; AVX512VL512-LABEL: 'zext512'
; AVX512VL512-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %A = zext <8 x i16> undef to
<8 x i64>
; AVX512VL512-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %B = zext <8 x i32> undef to
<8 x i64>
; AVX512VL512-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %C = zext <16 x i8> undef to
<16 x i32>
; AVX512VL512-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %D = zext <16 x i16> undef to
<16 x i32>
; AVX512VL512-NEXT:  Cost Model: Found an estimated cost of 3 for instruction: %E = zext <32 x i8> undef to
<32 x i16>
; AVX512VL512-NEXT:  Cost Model: Found an estimated cost of 0 for instruction: ret void
;
```

```
; SKX256-LABEL: 'zext512'
; SKX256-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %A = zext <8 x i16> undef to <8 x i64>
; SKX256-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %B = zext <8 x i32> undef to <8 x i64>
; SKX256-NEXT:  Cost
 Model: Found an estimated cost of 1 for instruction: %C = zext <16 x i8> undef to <16 x i32>
; SKX256-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %D = zext <16 x i16> undef to <16 x i32>
; SKX256-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %E = zext <32 x i8> undef to <32 x i16>
; SKX256-NEXT:  Cost Model: Found an estimated cost of 0 for instruction: ret void
;
; SKX512-LABEL: 'zext512'
; SKX512-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %A = zext <8 x i16> undef to <8 x i64>
; SKX512-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %B = zext <8 x i32> undef to <8 x i64>
; SKX512-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %C = zext <16 x i8> undef to <16 x i32>
; SKX512-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %D = zext <16 x i16> undef to <16 x i32>
; SKX512-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %E = zext <32 x i8> undef to <32 x i16>
;
 SKX512-NEXT:  Cost Model: Found an estimated cost of 0 for instruction: ret void
;
 %A = zext <8 x i16> undef to <8 x i64>
 %B = zext <8 x i32> undef to <8 x i64>
 %C = zext <16 x i8> undef to <16 x i32>
 %D = zext <16 x i16> undef to <16 x i32>
 %E = zext <32 x i8> undef to <32 x i16>
 ret void
}

define void @sext256() "min-legal-vector-width"="256" {
; AVX-LABEL: 'sext256'
; AVX-NEXT:  Cost Model: Found an estimated cost of 4 for instruction: %A = sext <8 x i8> undef to <8 x i64>
; AVX-NEXT:  Cost Model: Found an estimated cost of 4 for instruction: %B = sext <8 x i16> undef to <8 x i64>
; AVX-NEXT:  Cost Model: Found an estimated cost of 5 for instruction: %C = sext <8 x i32> undef to <8 x i64>
; AVX-NEXT:  Cost Model: Found an estimated cost of 4 for instruction: %D = sext <16 x i8> undef to <16 x i32>
; AVX-NEXT:  Cost Model: Found an estimated cost of 3 for instruction: %E = sext <16 x i16> undef to <16 x i32>
; AVX-NEXT:  Cost Model: Found an estimated cost of 5 for
 instruction: %F = sext <32 x i8> undef to <32 x i16>
; AVX-NEXT:  Cost Model: Found an estimated cost of 0 for instruction: ret void
;
; AVX512VL256-LABEL: 'sext256'
; AVX512VL256-NEXT:  Cost Model: Found an estimated cost of 2 for instruction: %A = sext <8 x i8> undef to <8
```

x i64>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %B = sext <8 x i16> undef to <8 x i64>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %C = sext <8 x i32> undef to <8 x i64>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %D = sext <16 x i8> undef to <16 x i32>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %E = sext <16 x i16> undef to <16 x i32>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %F = sext <32 x i8> undef to <32 x i16>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
; AVX512VL512-LABEL: 'sext256'
;
 AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %A = sext <8 x i8> undef to <8 x i64>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %B = sext <8 x i16> undef to <8 x i64>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %C = sext <8 x i32> undef to <8 x i64>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %D = sext <16 x i8> undef to <16 x i32>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %E = sext <16 x i16> undef to <16 x i32>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %F = sext <32 x i8> undef to <32 x i16>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
; SKX256-LABEL: 'sext256'
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %A = sext <8 x i8> undef to <8 x i64>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction:
 %B = sext <8 x i16> undef to <8 x i64>
; SKX256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %C = sext <8 x i32> undef to <8 x i64>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %D = sext <16 x i8> undef to <16 x i32>
; SKX256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %E = sext <16 x i16> undef to <16 x i32>
; SKX256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %F = sext <32 x i8> undef to <32 x i16>
; SKX256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
; SKX512-LABEL: 'sext256'
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %A = sext <8 x i8> undef to <8 x i64>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %B = sext <8 x i16> undef to <8 x i64>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %C = sext <8 x i32> undef to <8 x

```
i64>
; SKX512-NEXT:  Cost Model: Found an estimated
 cost of 1 for instruction: %D = sext <16 x i8> undef to <16 x i32>
; SKX512-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %E = sext <16 x i16> undef to <16 x
i32>
; SKX512-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %F = sext <32 x i8> undef to <32 x
i16>
; SKX512-NEXT:  Cost Model: Found an estimated cost of 0 for instruction: ret void
;
 %A = sext <8 x i8> undef to <8 x i64>
 %B = sext <8 x i16> undef to <8 x i64>
 %C = sext <8 x i32> undef to <8 x i64>
 %D = sext <16 x i8> undef to <16 x i32>
 %E = sext <16 x i16> undef to <16 x i32>
 %F = sext <32 x i8> undef to <32 x i16>
 ret void
}

define void @sext512() "min-legal-vector-width"="512" {
; AVX-LABEL: 'sext512'
; AVX-NEXT:  Cost Model: Found an estimated cost of 4 for instruction: %A = sext <8 x i8> undef to <8 x i64>
; AVX-NEXT:  Cost Model: Found an estimated cost of 4 for instruction: %B = sext <8 x i16> undef to <8 x i64>
; AVX-NEXT:  Cost Model: Found an estimated cost of 5 for
 instruction: %C = sext <8 x i32> undef to <8 x i64>
; AVX-NEXT:  Cost Model: Found an estimated cost of 4 for instruction: %D = sext <16 x i8> undef to <16 x i32>
; AVX-NEXT:  Cost Model: Found an estimated cost of 3 for instruction: %E = sext <16 x i16> undef to <16 x i32>
; AVX-NEXT:  Cost Model: Found an estimated cost of 5 for instruction: %F = sext <32 x i8> undef to <32 x i16>
; AVX-NEXT:  Cost Model: Found an estimated cost of 0 for instruction: ret void
;
; AVX512VL256-LABEL: 'sext512'
; AVX512VL256-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %A = sext <8 x i8> undef to <8
x i64>
; AVX512VL256-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %B = sext <8 x i16> undef to
<8 x i64>
; AVX512VL256-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %C = sext <8 x i32> undef to
<8 x i64>
; AVX512VL256-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %D = sext <16 x i8> undef to
<16 x i32>
; AVX512VL256-NEXT:  Cost Model:
 Found an estimated cost of 1 for instruction: %E = sext <16 x i16> undef to <16 x i32>
; AVX512VL256-NEXT:  Cost Model: Found an estimated cost of 3 for instruction: %F = sext <32 x i8> undef to
<32 x i16>
; AVX512VL256-NEXT:  Cost Model: Found an estimated cost of 0 for instruction: ret void
;
; AVX512VL512-LABEL: 'sext512'
; AVX512VL512-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %A = sext <8 x i8> undef to <8
x i64>
; AVX512VL512-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %B = sext <8 x i16> undef to
```

<8 x i64>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %C = sext <8 x i32> undef to <8 x i64>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %D = sext <16 x i8> undef to <16 x i32>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %E = sext <16 x i16> undef to <16 x i32>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction:
 %F = sext <32 x i8> undef to <32 x i16>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
; SKX256-LABEL: 'sext512'
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %A = sext <8 x i8> undef to <8 x i64>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %B = sext <8 x i16> undef to <8 x i64>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %C = sext <8 x i32> undef to <8 x i64>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %D = sext <16 x i8> undef to <16 x i32>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %E = sext <16 x i16> undef to <16 x i32>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %F = sext <32 x i8> undef to <32 x i16>
; SKX256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
; SKX512-LABEL: 'sext512'
; SKX512-NEXT: Cost Model: Found an estimated cost
 of 1 for instruction: %A = sext <8 x i8> undef to <8 x i64>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %B = sext <8 x i16> undef to <8 x i64>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %C = sext <8 x i32> undef to <8 x i64>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %D = sext <16 x i8> undef to <16 x i32>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %E = sext <16 x i16> undef to <16 x i32>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %F = sext <32 x i8> undef to <32 x i16>
; SKX512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret void
;
 %A = sext <8 x i8> undef to <8 x i64>
 %B = sext <8 x i16> undef to <8 x i64>
 %C = sext <8 x i32> undef to <8 x i64>
 %D = sext <16 x i8> undef to <16 x i32>
 %E = sext <16 x i16> undef to <16 x i32>
 %F = sext <32 x i8> undef to <32 x i16>
 ret void
}

```
define void @trunc256()
 "min-legal-vector-width"="256" {
; VEC256-LABEL: 'trunc256'
; VEC256-NEXT:  Cost Model: Found an estimated cost of 3 for instruction: %A = trunc <8 x i64> undef to <8 x
i32>
; VEC256-NEXT:  Cost Model: Found an estimated cost of 10 for instruction: %B = trunc <8 x i64> undef to <8 x
i16>
; VEC256-NEXT:  Cost Model: Found an estimated cost of 8 for instruction: %C = trunc <8 x i64> undef to <8 x
i8>
; VEC256-NEXT:  Cost Model: Found an estimated cost of 4 for instruction: %D = trunc <16 x i32> undef to <16 x
i16>
; VEC256-NEXT:  Cost Model: Found an estimated cost of 4 for instruction: %E = trunc <16 x i32> undef to <16 x
i8>
; VEC256-NEXT:  Cost Model: Found an estimated cost of 5 for instruction: %F = trunc <32 x i16> undef to <32 x
i8>
; VEC256-NEXT:  Cost Model: Found an estimated cost of 0 for instruction: ret void
;
; AVX512VL512-LABEL: 'trunc256'
; AVX512VL512-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %A = trunc <8 x i64> undef to
<8 x i32>
; AVX512VL512-NEXT:
  Cost Model: Found an estimated cost of 2 for instruction: %B = trunc <8 x i64> undef to <8 x i16>
; AVX512VL512-NEXT:  Cost Model: Found an estimated cost of 2 for instruction: %C = trunc <8 x i64> undef to
<8 x i8>
; AVX512VL512-NEXT:  Cost Model: Found an estimated cost of 2 for instruction: %D = trunc <16 x i32> undef
to <16 x i16>
; AVX512VL512-NEXT:  Cost Model: Found an estimated cost of 2 for instruction: %E = trunc <16 x i32> undef to
<16 x i8>
; AVX512VL512-NEXT:  Cost Model: Found an estimated cost of 8 for instruction: %F = trunc <32 x i16> undef to
<32 x i8>
; AVX512VL512-NEXT:  Cost Model: Found an estimated cost of 0 for instruction: ret void
;
; SKX512-LABEL: 'trunc256'
; SKX512-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %A = trunc <8 x i64> undef to <8 x
i32>
; SKX512-NEXT:  Cost Model: Found an estimated cost of 2 for instruction: %B = trunc <8 x i64> undef to <8 x
i16>
; SKX512-NEXT:  Cost Model: Found an estimated cost of 2 for instruction:
 %C = trunc <8 x i64> undef to <8 x i8>
; SKX512-NEXT:  Cost Model: Found an estimated cost of 2 for instruction: %D = trunc <16 x i32> undef to <16 x
i16>
; SKX512-NEXT:  Cost Model: Found an estimated cost of 2 for instruction: %E = trunc <16 x i32> undef to <16 x
i8>
; SKX512-NEXT:  Cost Model: Found an estimated cost of 2 for instruction: %F = trunc <32 x i16> undef to <32 x
i8>
; SKX512-NEXT:  Cost Model: Found an estimated cost of 0 for instruction: ret void
;
```

```
 %A = trunc <8 x i64> undef to <8 x i32>
 %B = trunc <8 x i64> undef to <8 x i16>
 %C = trunc <8 x i64> undef to <8 x i8>
 %D = trunc <16 x i32> undef to <16 x i16>
 %E = trunc <16 x i32> undef to <16 x i8>
 %F = trunc <32 x i16> undef to <32 x i8>
 ret void
}

define i32 @zext256_vXi1() "min-legal-vector-width"="256" {
; AVX-LABEL: 'zext256_vXi1'
; AVX-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %V2i64 = zext <2 x i1> undef to <2 x
i64>
; AVX-NEXT:  Cost Model: Found an estimated cost of 3
 for instruction: %V4i64 = zext <4 x i1> undef to <4 x i64>
; AVX-NEXT:  Cost Model: Found an estimated cost of 4 for instruction: %V8i64 = zext <8 x i1> undef to <8 x
i64>
; AVX-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %V2i32 = zext <2 x i1> undef to <2 x
i32>
; AVX-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %V4i32 = zext <4 x i1> undef to <4 x
i32>
; AVX-NEXT:  Cost Model: Found an estimated cost of 3 for instruction: %V8i32 = zext <8 x i1> undef to <8 x
i32>
; AVX-NEXT:  Cost Model: Found an estimated cost of 4 for instruction: %V16i32 = zext <16 x i1> undef to <16 x
i32>
; AVX-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %V2i16 = zext <2 x i1> undef to <2 x
i16>
; AVX-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %V4i16 = zext <4 x i1> undef to <4 x
i16>
; AVX-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %V8i16 = zext <8 x i1> undef to <8 x
i16>
; AVX-NEXT:  Cost Model: Found an
 estimated cost of 1 for instruction: %V16i16 = zext <16 x i1> undef to <16 x i16>
; AVX-NEXT:  Cost Model: Found an estimated cost of 3 for instruction: %V32i16 = zext <32 x i1> undef to <32 x
i16>
; AVX-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %V2i8 = zext <2 x i1> undef to <2 x i8>
; AVX-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %V4i8 = zext <4 x i1> undef to <4 x i8>
; AVX-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %V8i8 = zext <8 x i1> undef to <8 x i8>
; AVX-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %V16i8 = zext <16 x i1> undef to <16 x
i8>
; AVX-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %V32i8 = zext <32 x i1> undef to <32 x
i8>
; AVX-NEXT:  Cost Model: Found an estimated cost of 2 for instruction: %V64i8 = zext <64 x i1> undef to <64 x
i8>
; AVX-NEXT:  Cost Model: Found an estimated cost of 0 for instruction: ret i32 undef
;
; AVX512VL256-LABEL: 'zext256_vXi1'
; AVX512VL256-NEXT:
```

Cost Model: Found an estimated cost of 2 for instruction: %V2i64 = zext <2 x i1> undef to <2 x i64>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i64 = zext <4 x i1> undef to <4 x i64>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 5 for instruction: %V8i64 = zext <8 x i1> undef to <8 x i64>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i32 = zext <2 x i1> undef to <2 x i32>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i32 = zext <4 x i1> undef to <4 x i32>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i32 = zext <8 x i1> undef to <8 x i32>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 5 for instruction: %V16i32 = zext <16 x i1> undef to <16 x i32>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 5 for instruction: %V2i16 = zext <2 x i1> undef to <2 x i16>
; AVX512VL256-NEXT: Cost Model:
Found an estimated cost of 5 for instruction: %V4i16 = zext <4 x i1> undef to <4 x i16>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 5 for instruction: %V8i16 = zext <8 x i1> undef to <8 x i16>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 12 for instruction: %V16i16 = zext <16 x i1> undef to <16 x i16>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 24 for instruction: %V32i16 = zext <32 x i1> undef to <32 x i16>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 6 for instruction: %V2i8 = zext <2 x i1> undef to <2 x i8>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 6 for instruction: %V4i8 = zext <4 x i1> undef to <4 x i8>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 6 for instruction: %V8i8 = zext <8 x i1> undef to <8 x i8>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 12 for instruction: %V16i8 = zext <16 x i1> undef to <16 x i8>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 25
for instruction: %V32i8 = zext <32 x i1> undef to <32 x i8>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 50 for instruction: %V64i8 = zext <64 x i1> undef to <64 x i8>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret i32 undef
;
; AVX512VL512-LABEL: 'zext256_vXi1'
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i64 = zext <2 x i1> undef to <2 x i64>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i64 = zext <4 x i1> undef to <4 x i64>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i64 = zext <8 x i1> undef to <8 x i64>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i32 = zext <2 x i1> undef to <2 x i32>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i32 = zext <4 x i1> undef to <4 x i32>

; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 2 for instruction:
 %V8i32 = zext <8 x i1> undef to <8 x i32>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V16i32 = zext <16 x i1> undef to <16 x i32>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V2i16 = zext <2 x i1> undef to <2 x i16>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V4i16 = zext <4 x i1> undef to <4 x i16>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V8i16 = zext <8 x i1> undef to <8 x i16>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V16i16 = zext <16 x i1> undef to <16 x i16>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 9 for instruction: %V32i16 = zext <32 x i1> undef to <32 x i16>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V2i8 = zext <2 x i1> undef to <2 x i8>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V4i8 = zext <4 x i1> undef to <4 x i8>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V8i8 = zext <8 x i1> undef to <8 x i8>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V16i8 = zext <16 x i1> undef to <16 x i8>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 9 for instruction: %V32i8 = zext <32 x i1> undef to <32 x i8>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 19 for instruction: %V64i8 = zext <64 x i1> undef to <64 x i8>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret i32 undef
;
; SKX256-LABEL: 'zext256_vXi1'
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i64 = zext <2 x i1> undef to <2 x i64>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i64 = zext <4 x i1> undef to <4 x i64>
; SKX256-NEXT: Cost Model: Found an estimated cost of 5 for instruction: %V8i64 = zext <8 x i1> undef to <8 x i64>
; SKX256-NEXT: Cost Model:
 Found an estimated cost of 2 for instruction: %V2i32 = zext <2 x i1> undef to <2 x i32>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i32 = zext <4 x i1> undef to <4 x i32>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i32 = zext <8 x i1> undef to <8 x i32>
; SKX256-NEXT: Cost Model: Found an estimated cost of 5 for instruction: %V16i32 = zext <16 x i1> undef to <16 x i32>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i16 = zext <2 x i1> undef to <2 x i16>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i16 = zext <4 x i1> undef to <4 x i16>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i16 = zext <8 x i1> undef to <8 x

i16>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V16i16 = zext <16 x i1> undef to <16 x i16>
; SKX256-NEXT: Cost Model: Found an estimated cost of 5 for instruction: %V32i16 = zext <32 x i1> undef to <32 x i16>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i8 = zext <2 x i1> undef to <2 x i8>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i8 = zext <4 x i1> undef to <4 x i8>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i8 = zext <8 x i1> undef to <8 x i8>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V16i8 = zext <16 x i1> undef to <16 x i8>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V32i8 = zext <32 x i1> undef to <32 x i8>
; SKX256-NEXT: Cost Model: Found an estimated cost of 5 for instruction: %V64i8 = zext <64 x i1> undef to <64 x i8>
; SKX256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret i32 undef
;
; SKX512-LABEL: 'zext256_vXi1'
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i64 = zext <2 x i1> undef to <2 x i64>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i64 = zext <4 x i1> undef to <4 x i64>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i64 = zext <8 x i1> undef to <8 x i64>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i32 = zext <2 x i1> undef to <2 x i32>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i32 = zext <4 x i1> undef to <4 x i32>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i32 = zext <8 x i1> undef to <8 x i32>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V16i32 = zext <16 x i1> undef to <16 x i32>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i16 = zext <2 x i1> undef to <2 x i16>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i16 = zext <4 x i1> undef to <4 x i16>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i16 = zext <8 x i1> undef to <8 x i16>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V16i16 = zext <16 x i1> undef to <16 x i16>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V32i16 = zext <32 x i1> undef to <32 x i16>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i8 = zext <2 x i1> undef to <2 x i8>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i8 = zext <4 x i1> undef to <4 x i8>

; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i8 = zext <8 x i1> undef to <8 x i8>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V16i8 = zext <16 x i1> undef to <16 x i8>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V32i8 = zext <32 x i1> undef to <32 x i8>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V64i8 = zext <64 x i1> undef to <64 x i8>
; SKX512-NEXT: Cost Model: Found an estimated cost of 0 for instruction:
 ret i32 undef
;
 %V2i64 = zext <2 x i1> undef to <2 x i64>
 %V4i64 = zext <4 x i1> undef to <4 x i64>
 %V8i64 = zext <8 x i1> undef to <8 x i64>

 %V2i32 = zext <2 x i1> undef to <2 x i32>
 %V4i32 = zext <4 x i1> undef to <4 x i32>
 %V8i32 = zext <8 x i1> undef to <8 x i32>
 %V16i32 = zext <16 x i1> undef to <16 x i32>

 %V2i16 = zext <2 x i1> undef to <2 x i16>
 %V4i16 = zext <4 x i1> undef to <4 x i16>
 %V8i16 = zext <8 x i1> undef to <8 x i16>
 %V16i16 = zext <16 x i1> undef to <16 x i16>
 %V32i16 = zext <32 x i1> undef to <32 x i16>

 %V2i8 = zext <2 x i1> undef to <2 x i8>
 %V4i8 = zext <4 x i1> undef to <4 x i8>
 %V8i8 = zext <8 x i1> undef to <8 x i8>
 %V16i8 = zext <16 x i1> undef to <16 x i8>
 %V32i8 = zext <32 x i1> undef to <32 x i8>
 %V64i8 = zext <64 x i1> undef to <64 x i8>

 ret i32 undef
}

define i32 @sext256_vXi1() "min-legal-vector-width"="256" {
; AVX-LABEL: 'sext256_vXi1'
; AVX-NEXT: Cost Model: Found an estimated cost of 1 for instruction:
 %I64 = sext i1 undef to i64
; AVX-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i64 = sext <2 x i1> undef to <2 x i64>
; AVX-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V4i64 = sext <4 x i1> undef to <4 x i64>
; AVX-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V8i64 = sext <8 x i1> undef to <8 x i64>
; AVX-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %I32 = sext i1 undef to i32
; AVX-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i32 = sext <2 x i1> undef to <2 x

i32>
; AVX-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i32 = sext <4 x i1> undef to <4 x i32>
; AVX-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V8i32 = sext <8 x i1> undef to <8 x i32>
; AVX-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V16i32 = sext <16 x i1> undef to <16 x i32>
; AVX-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %I16
 = sext i1 undef to i16
; AVX-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i16 = sext <2 x i1> undef to <2 x i16>
; AVX-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i16 = sext <4 x i1> undef to <4 x i16>
; AVX-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i16 = sext <8 x i1> undef to <8 x i16>
; AVX-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V16i16 = sext <16 x i1> undef to <16 x i16>
; AVX-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V32i16 = sext <32 x i1> undef to <32 x i16>
; AVX-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %I8 = sext i1 undef to i8
; AVX-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i8 = sext <2 x i1> undef to <2 x i8>
; AVX-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i8 = sext <4 x i1> undef to <4 x i8>
; AVX-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i8 = sext
 <8 x i1> undef to <8 x i8>
; AVX-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V16i8 = sext <16 x i1> undef to <16 x i8>
; AVX-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V32i8 = sext <32 x i1> undef to <32 x i8>
; AVX-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V64i8 = sext <64 x i1> undef to <64 x i8>
; AVX-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret i32 undef
;
; AVX512VL256-LABEL: 'sext256_vXi1'
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %I64 = sext i1 undef to i64
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V2i64 = sext <2 x i1> undef to <2 x i64>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V4i64 = sext <4 x i1> undef to <4 x i64>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V8i64 = sext <8 x i1> undef to <8 x i64>
; AVX512VL256-NEXT: Cost Model: Found
 an estimated cost of 1 for instruction: %I32 = sext i1 undef to i32
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V2i32 = sext <2 x i1> undef to <2 x i32>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V4i32 = sext <4 x i1> undef to <4 x i32>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V8i32 = sext <8 x i1> undef to <8 x i32>

; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V16i32 = sext <16 x i1> undef to <16 x i32>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %I16 = sext i1 undef to i16
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V2i16 = sext <2 x i1> undef to <2 x i16>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V4i16 = sext <4 x i1> undef to <4 x i16>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V8i16 = sext <8 x i1> undef to <8 x i16>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 10 for instruction: %V16i16 = sext <16 x i1> undef to <16 x i16>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 20 for instruction: %V32i16 = sext <32 x i1> undef to <32 x i16>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %I8 = sext i1 undef to i8
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 5 for instruction: %V2i8 = sext <2 x i1> undef to <2 x i8>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 5 for instruction: %V4i8 = sext <4 x i1> undef to <4 x i8>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 5 for instruction: %V8i8 = sext <8 x i1> undef to <8 x i8>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 10 for instruction: %V16i8 = sext <16 x i1> undef to <16 x i8>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 21 for instruction: %V32i8 = sext <32 x i1> undef to <32 x i8>
; AVX512VL256-NEXT:
  Cost Model: Found an estimated cost of 42 for instruction: %V64i8 = sext <64 x i1> undef to <64 x i8>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret i32 undef
;
; AVX512VL512-LABEL: 'sext256_vXi1'
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %I64 = sext i1 undef to i64
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V2i64 = sext <2 x i1> undef to <2 x i64>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V4i64 = sext <4 x i1> undef to <4 x i64>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V8i64 = sext <8 x i1> undef to <8 x i64>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %I32 = sext i1 undef to i32
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V2i32 = sext <2 x i1> undef to <2 x i32>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V4i32 = sext <4 x i1> undef to <4 x i32>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V8i32 = sext <8 x i1> undef to <8 x i32>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V16i32 = sext <16 x i1> undef to <16 x i32>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %I16 = sext i1 undef to i16
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V2i16 = sext <2 x i1> undef to <2 x i16>

; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V4i16 = sext <4 x i1> undef to <4 x i16>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V8i16 = sext <8 x i1> undef to <8 x i16>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V16i16 = sext <16 x i1> undef to <16 x i16>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 7 for instruction: %V32i16 = sext <32 x i1> undef to <32 x i16>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %I8 = sext i1 undef to i8
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V2i8 = sext <2 x i1> undef to <2 x i8>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V4i8 = sext <4 x i1> undef to <4 x i8>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V8i8 = sext <8 x i1> undef to <8 x i8>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V16i8 = sext <16 x i1> undef to <16 x i8>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 7 for instruction: %V32i8 = sext <32 x i1> undef to <32 x i8>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 15 for instruction: %V64i8 = sext <64 x i1> undef to <64 x i8>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret i32 undef
;
; SKX256-LABEL: 'sext256_vXi1'
; SKX256-NEXT: Cost Model:
 Found an estimated cost of 1 for instruction: %I64 = sext i1 undef to i64
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V2i64 = sext <2 x i1> undef to <2 x i64>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V4i64 = sext <4 x i1> undef to <4 x i64>
; SKX256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V8i64 = sext <8 x i1> undef to <8 x i64>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %I32 = sext i1 undef to i32
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V2i32 = sext <2 x i1> undef to <2 x i32>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V4i32 = sext <4 x i1> undef to <4 x i32>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V8i32 = sext <8 x i1> undef to <8 x i32>
; SKX256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V16i32 = sext <16 x i1> undef to <16 x i32>
; SKX256-NEXT:
  Cost Model: Found an estimated cost of 1 for instruction: %I16 = sext i1 undef to i16
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V2i16 = sext <2 x i1> undef to <2 x i16>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V4i16 = sext <4 x i1> undef to <4 x i16>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V8i16 = sext <8 x i1> undef to <8 x

i16>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V16i16 = sext <16 x i1> undef to <16 x i16>
; SKX256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V32i16 = sext <32 x i1> undef to <32 x i16>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %I8 = sext i1 undef to i8
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V2i8 = sext <2 x i1> undef to <2 x i8>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V4i8 = sext <4 x i1> undef to <4 x i8>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V8i8 = sext <8 x i1> undef to <8 x i8>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V16i8 = sext <16 x i1> undef to <16 x i8>
; SKX256-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V32i8 = sext <32 x i1> undef to <32 x i8>
; SKX256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V64i8 = sext <64 x i1> undef to <64 x i8>
; SKX256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret i32 undef
;
; SKX512-LABEL: 'sext256_vXi1'
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %I64 = sext i1 undef to i64
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V2i64 = sext <2 x i1> undef to <2 x i64>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V4i64 = sext <4 x i1> undef to <4 x i64>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V8i64 = sext <8 x i1> undef to <8 x i64>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %I32 = sext i1 undef to i32
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V2i32 = sext <2 x i1> undef to <2 x i32>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V4i32 = sext <4 x i1> undef to <4 x i32>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V8i32 = sext <8 x i1> undef to <8 x i32>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V16i32 = sext <16 x i1> undef to <16 x i32>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %I16 = sext i1 undef to i16
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V2i16 = sext <2 x i1> undef to <2 x i16>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V4i16 = sext <4 x i1> undef to <4 x i16>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V8i16 = sext <8 x i1> undef to <8 x i16>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V16i16 = sext <16 x i1> undef to <16 x i16>
; SKX512-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V32i16 = sext <32 x i1> undef to <32 x i16>

```
; SKX512-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %I8 = sext i1 undef to i8
; SKX512-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %V2i8 = sext <2 x i1> undef to <2 x
i8>
; SKX512-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %V4i8 = sext <4 x i1> undef to <4 x
i8>
; SKX512-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %V8i8 = sext <8 x i1> undef to <8 x
i8>
; SKX512-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %V16i8 = sext <16 x i1> undef to <16
x i8>
; SKX512-NEXT:  Cost Model: Found an estimated cost of 1 for instruction: %V32i8 = sext <32 x i1> undef to <32
x i8>
; SKX512-NEXT:  Cost
 Model: Found an estimated cost of 1 for instruction: %V64i8 = sext <64 x i1> undef to <64 x i8>
; SKX512-NEXT:  Cost Model: Found an estimated cost of 0 for instruction: ret i32 undef
;
 %I64 = sext i1 undef to i64
 %V2i64 = sext <2 x i1> undef to <2 x i64>
 %V4i64 = sext <4 x i1> undef to <4 x i64>
 %V8i64 = sext <8 x i1> undef to <8 x i64>

 %I32 = sext i1 undef to i32
 %V2i32 = sext <2 x i1> undef to <2 x i32>
 %V4i32 = sext <4 x i1> undef to <4 x i32>
 %V8i32 = sext <8 x i1> undef to <8 x i32>
 %V16i32 = sext <16 x i1> undef to <16 x i32>

 %I16 = sext i1 undef to i16
 %V2i16 = sext <2 x i1> undef to <2 x i16>
 %V4i16 = sext <4 x i1> undef to <4 x i16>
 %V8i16 = sext <8 x i1> undef to <8 x i16>
 %V16i16 = sext <16 x i1> undef to <16 x i16>
 %V32i16 = sext <32 x i1> undef to <32 x i16>

 %I8 = sext i1 undef to i8
 %V2i8 = sext <2 x i1> undef to <2 x i8>
 %V4i8 = sext <4 x i1> undef to <4 x i8>
 %V8i8 = sext <8 x i1> undef to <8 x i8>
 %V16i8 = sext <16 x i1> undef
 to <16 x i8>
 %V32i8 = sext <32 x i1> undef to <32 x i8>
 %V64i8 = sext <64 x i1> undef to <64 x i8>

 ret i32 undef
}

define i32 @trunc_vXi1() "min-legal-vector-width"="256" {
; AVX-LABEL: 'trunc_vXi1'
; AVX-NEXT:  Cost Model: Found an estimated cost of 0 for instruction: %V2i64 = trunc <2 x i64> undef to <2 x
```

i1>
; AVX-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V4i64 = trunc <4 x i64> undef to <4 x i1>
; AVX-NEXT: Cost Model: Found an estimated cost of 9 for instruction: %V8i64 = trunc <8 x i64> undef to <8 x i1>
; AVX-NEXT: Cost Model: Found an estimated cost of 11 for instruction: %V16i64 = trunc <16 x i64> undef to <16 x i1>
; AVX-NEXT: Cost Model: Found an estimated cost of 23 for instruction: %V32i64 = trunc <32 x i64> undef to <32 x i1>
; AVX-NEXT: Cost Model: Found an estimated cost of 46 for instruction: %V64i64 = trunc <64 x i64> undef to <64 x i1>
; AVX-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V2i32 = trunc <2 x i32>
 undef to <2 x i1>
; AVX-NEXT: Cost Model: Found an estimated cost of 0 for instruction: %V4i32 = trunc <4 x i32> undef to <4 x i1>
; AVX-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i32 = trunc <8 x i32> undef to <8 x i1>
; AVX-NEXT: Cost Model: Found an estimated cost of 8 for instruction: %V16i32 = trunc <16 x i32> undef to <16 x i1>
; AVX-NEXT: Cost Model: Found an estimated cost of 17 for instruction: %V32i32 = trunc <32 x i32> undef to <32 x i1>
; AVX-NEXT: Cost Model: Found an estimated cost of 34 for instruction: %V64i32 = trunc <64 x i32> undef to <64 x i1>
; AVX-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V2i16 = trunc <2 x i16> undef to <2 x i1>
; AVX-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V4i16 = trunc <4 x i16> undef to <4 x i1>
; AVX-NEXT: Cost Model: Found an estimated cost of 0 for instruction: %V8i16 = trunc <8 x i16> undef to <8 x i1>
; AVX-NEXT: Cost Model: Found an estimated cost of 4 for
 instruction: %V16i16 = trunc <16 x i16> undef to <16 x i1>
; AVX-NEXT: Cost Model: Found an estimated cost of 9 for instruction: %V32i16 = trunc <32 x i16> undef to <32 x i1>
; AVX-NEXT: Cost Model: Found an estimated cost of 18 for instruction: %V64i16 = trunc <64 x i16> undef to <64 x i1>
; AVX-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V2i8 = trunc <2 x i8> undef to <2 x i1>
; AVX-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V4i8 = trunc <4 x i8> undef to <4 x i1>
; AVX-NEXT: Cost Model: Found an estimated cost of 1 for instruction: %V8i8 = trunc <8 x i8> undef to <8 x i1>
; AVX-NEXT: Cost Model: Found an estimated cost of 0 for instruction: %V16i8 = trunc <16 x i8> undef to <16 x i1>
; AVX-NEXT: Cost Model: Found an estimated cost of 0 for instruction: %V32i8 = trunc <32 x i8> undef to <32 x i1>
; AVX-NEXT: Cost Model: Found an estimated cost of 0 for instruction: %V64i8 = trunc <64 x i8> undef to <64 x i1>
; AVX-NEXT: Cost Model:
 Found an estimated cost of 0 for instruction: ret i32 undef
;
; AVX512VL256-LABEL: 'trunc_vXi1'

; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i64 = trunc <2 x i64> undef to <2 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i64 = trunc <4 x i64> undef to <4 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 9 for instruction: %V8i64 = trunc <8 x i64> undef to <8 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 11 for instruction: %V16i64 = trunc <16 x i64> undef to <16 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 22 for instruction: %V32i64 = trunc <32 x i64> undef to <32 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 44 for instruction: %V64i64 = trunc <64 x i64> undef to <64 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i32 = trunc <2 x i32> undef to <2 x i1>
; AVX512VL256-NEXT: Cost Model: Found
 an estimated cost of 2 for instruction: %V4i32 = trunc <4 x i32> undef to <4 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i32 = trunc <8 x i32> undef to <8 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 5 for instruction: %V16i32 = trunc <16 x i32> undef to <16 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 10 for instruction: %V32i32 = trunc <32 x i32> undef to <32 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 20 for instruction: %V64i32 = trunc <64 x i32> undef to <64 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V2i16 = trunc <2 x i16> undef to <2 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V4i16 = trunc <4 x i16> undef to <4 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V8i16 = trunc <8 x i16> undef to <8 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated
 cost of 8 for instruction: %V16i16 = trunc <16 x i16> undef to <16 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 16 for instruction: %V32i16 = trunc <32 x i16> undef to <32 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 32 for instruction: %V64i16 = trunc <64 x i16> undef to <64 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V2i8 = trunc <2 x i8> undef to <2 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V4i8 = trunc <4 x i8> undef to <4 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V8i8 = trunc <8 x i8> undef to <8 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 8 for instruction: %V16i8 = trunc <16 x i8> undef to <16 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 17 for instruction: %V32i8 = trunc <32 x i8> undef to <32 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 34 for

instruction: %V64i8 = trunc <64 x i8> undef to <64 x i1>
; AVX512VL256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret i32 undef
;
; AVX512VL512-LABEL: 'trunc_vXi1'
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i64 = trunc <2 x i64> undef to <2 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i64 = trunc <4 x i64> undef to <4 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i64 = trunc <8 x i64> undef to <8 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 11 for instruction: %V16i64 = trunc <16 x i64> undef to <16 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 22 for instruction: %V32i64 = trunc <32 x i64> undef to <32 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 44 for instruction: %V64i64 = trunc <64 x i64> undef to <64 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 2 for instruction:
 %V2i32 = trunc <2 x i32> undef to <2 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i32 = trunc <4 x i32> undef to <4 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i32 = trunc <8 x i32> undef to <8 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V16i32 = trunc <16 x i32> undef to <16 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 4 for instruction: %V32i32 = trunc <32 x i32> undef to <32 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 8 for instruction: %V64i32 = trunc <64 x i32> undef to <64 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V2i16 = trunc <2 x i16> undef to <2 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V4i16 = trunc <4 x i16> undef to <4 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V8i16
 = trunc <8 x i16> undef to <8 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V16i16 = trunc <16 x i16> undef to <16 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 7 for instruction: %V32i16 = trunc <32 x i16> undef to <32 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 14 for instruction: %V64i16 = trunc <64 x i16> undef to <64 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V2i8 = trunc <2 x i8> undef to <2 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V4i8 = trunc <4 x i8> undef to <4 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V8i8 = trunc <8 x i8> undef to <8 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 3 for instruction: %V16i8 = trunc <16 x i8> undef to <16 x i1>

; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 7 for instruction: %V32i8 = trunc <32 x i8>
 undef to <32 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 15 for instruction: %V64i8 = trunc <64 x i8>
undef to <64 x i1>
; AVX512VL512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret i32 undef
;
; SKX256-LABEL: 'trunc_vXi1'
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i64 = trunc <2 x i64> undef to <2
x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i64 = trunc <4 x i64> undef to <4
x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 9 for instruction: %V8i64 = trunc <8 x i64> undef to <8
x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 11 for instruction: %V16i64 = trunc <16 x i64> undef to
<16 x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 23 for instruction: %V32i64 = trunc <32 x i64> undef to
<32 x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 47 for instruction: %V64i64 = trunc <64 x i64> undef to
<64 x i1>
; SKX256-NEXT: Cost
 Model: Found an estimated cost of 2 for instruction: %V2i32 = trunc <2 x i32> undef to <2 x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i32 = trunc <4 x i32> undef to <4
x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i32 = trunc <8 x i32> undef to <8
x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 5 for instruction: %V16i32 = trunc <16 x i32> undef to
<16 x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 11 for instruction: %V32i32 = trunc <32 x i32> undef to
<32 x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 23 for instruction: %V64i32 = trunc <64 x i32> undef to
<64 x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i16 = trunc <2 x i16> undef to <2
x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i16 = trunc <4 x i16> undef to <4
x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction:
 %V8i16 = trunc <8 x i16> undef to <8 x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V16i16 = trunc <16 x i16> undef to
<16 x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 5 for instruction: %V32i16 = trunc <32 x i16> undef to
<32 x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 11 for instruction: %V64i16 = trunc <64 x i16> undef to
<64 x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i8 = trunc <2 x i8> undef to <2 x
i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i8 = trunc <4 x i8> undef to <4 x
i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i8 = trunc <8 x i8> undef to <8 x

i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V16i8 = trunc <16 x i8> undef to <16 x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V32i8 = trunc <32 x i8> undef to <32 x i1>
; SKX256-NEXT:
  Cost Model: Found an estimated cost of 5 for instruction: %V64i8 = trunc <64 x i8> undef to <64 x i1>
; SKX256-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret i32 undef
;
; SKX512-LABEL: 'trunc_vXi1'
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i64 = trunc <2 x i64> undef to <2 x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i64 = trunc <4 x i64> undef to <4 x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i64 = trunc <8 x i64> undef to <8 x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 11 for instruction: %V16i64 = trunc <16 x i64> undef to <16 x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 23 for instruction: %V32i64 = trunc <32 x i64> undef to <32 x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 47 for instruction: %V64i64 = trunc <64 x i64> undef to <64 x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction:
 %V2i32 = trunc <2 x i32> undef to <2 x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i32 = trunc <4 x i32> undef to <4 x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i32 = trunc <8 x i32> undef to <8 x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V16i32 = trunc <16 x i32> undef to <16 x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 5 for instruction: %V32i32 = trunc <32 x i32> undef to <32 x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 11 for instruction: %V64i32 = trunc <64 x i32> undef to <64 x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i16 = trunc <2 x i16> undef to <2 x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i16 = trunc <4 x i16> undef to <4 x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i16 = trunc <8 x i16> undef to <8 x i1>
;
 SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V16i16 = trunc <16 x i16> undef to <16 x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V32i16 = trunc <32 x i16> undef to <32 x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 5 for instruction: %V64i16 = trunc <64 x i16> undef to <64 x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V2i8 = trunc <2 x i8> undef to <2 x

i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V4i8 = trunc <4 x i8> undef to <4 x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V8i8 = trunc <8 x i8> undef to <8 x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V16i8 = trunc <16 x i8> undef to <16 x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for instruction: %V32i8 = trunc <32 x i8> undef to <32 x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 2 for
 instruction: %V64i8 = trunc <64 x i8> undef to <64 x i1>
; SKX512-NEXT: Cost Model: Found an estimated cost of 0 for instruction: ret i32 undef
;
 %V2i64 = trunc <2 x i64> undef to <2 x i1>
 %V4i64 = trunc <4 x i64> undef to <4 x i1>
 %V8i64 = trunc <8 x i64> undef to <8 x i1>
 %V16i64 = trunc <16 x i64> undef to <16 x i1>
 %V32i64 = trunc <32 x i64> undef to <32 x i1>
 %V64i64 = trunc <64 x i64> undef to <64 x i1>

 %V2i32 = trunc <2 x i32> undef to <2 x i1>
 %V4i32 = trunc <4 x i32> undef to <4 x i1>
 %V8i32 = trunc <8 x i32> undef to <8 x i1>
 %V16i32 = trunc <16 x i32> undef to <16 x i1>
 %V32i32 = trunc <32 x i32> undef to <32 x i1>
 %V64i32 = trunc <64 x i32> undef to <64 x i1>

 %V2i16 = trunc <2 x i16> undef to <2 x i1>
 %V4i16 = trunc <4 x i16> undef to <4 x i1>
 %V8i16 = trunc <8 x i16> undef to <8 x i1>
 %V16i16 = trunc <16 x i16> undef to <16 x i1>
 %V32i16 = trunc <32 x i16> undef to <32 x i1>
 %V64i16 = trunc <64 x i16> undef to <64 x i1>

 %V2i8 =
 trunc <2 x i8> undef to <2 x i1>
 %V4i8 = trunc <4 x i8> undef to <4 x i1>
 %V8i8 = trunc <8 x i8> undef to <8 x i1>
 %V16i8 = trunc <16 x i8> undef to <16 x i1>
 %V32i8 = trunc <32 x i8> undef to <32 x i1>
 %V64i8 = trunc <64 x i8> undef to <64 x i1>

 ret i32 undef
}
# People who have agreed to one of the CLAs and can contribute patches.
# The AUTHORS file lists the copyright holders; this file
# lists people. For example, Google employees are listed here
# but not in AUTHORS, because Google holds the copyright.

#
Names should be added to this file only after verifying that
the individual or the individual's organization has agreed to
the appropriate Contributor License Agreement, found here:
#
https://developers.google.com/open-source/cla/individual
https://developers.google.com/open-source/cla/corporate
#
The agreement for individuals can be filled out on the web.
#
When adding J Random Contributor's name to this file,
either J's name or J's organization's name should be
added to the AUTHORS file, depending on whether the
individual or corporate CLA was used.
#
Names should be added to this file as:
Name <email address>
#
Please keep the list sorted.

Albert Pretorius <pretoalb@gmail.com>
Arne Beer <arne@twobeer.de>
Billy
 Robert O'Neal III <billy.oneal@gmail.com> <bion@microsoft.com>
Chris Kennelly <ckennelly@google.com> <ckennelly@ckennelly.com>
Christopher Seymour <chris.j.seymour@hotmail.com>
David Coeurjolly <david.coeurjolly@liris.cnrs.fr>
Deniz Evrenci <denizevrenci@gmail.com>
Dominic Hamon <dma@stripysock.com> <dominic@google.com>
Dominik Czarnota <dominik.b.czarnota@gmail.com>
Eric Fiselier <eric@efcs.ca>
Eugene Zhuk <eugene.zhuk@gmail.com>
Evgeny Safronov <division494@gmail.com>
Felix Homann <linuxaudio@showlabor.de>
Ismael Jimenez Martinez <ismael.jimenez.martinez@gmail.com>
Jern-Kuan Leong <jernkuan@gmail.com>
JianXiong Zhou <zhoujianxiong2@gmail.com>
Joao Paulo Magalhaes <joaoppmagalhaes@gmail.com>
John Millikin <jmillikin@stripe.com>
Jussi Knuuttila <jussi.knuuttila@gmail.com>
Kai Wolf <kai.wolf@gmail.com>
Kishan Kumar <kumar.kishan@outlook.com>
Kaito Udagawa <umireon@gmail.com>
Lei Xu <eddyxu@gmail.com>
Matt Clarkson <mattyclarkson@gmail.com>
Maxim Vafin <maxvafin@gmail.com>
Nick Hutchinson

<nshutchinson@gmail.com>
Oleksandr Sochka <sasha.sochka@gmail.com>
Pascal Leroy <phl@google.com>
Paul Redmond <paul.redmond@gmail.com>
Pierre Phaneuf <pphaneuf@google.com>
Radoslav Yovchev <radoslav.tm@gmail.com>
Raul Marin <rmrodriguez@cartodb.com>
Ray Glover <ray.glover@uk.ibm.com>
Robert Guo <robert.guo@mongodb.com>
Roman Lebedev <lebedev.ri@gmail.com>
Shuo Chen <chenshuo@chenshuo.com>
Steven Wan <wan.yu@ibm.com>
Tobias Ulvgrd <tobias.ulvgard@dirac.se>
Tom Madams <tom.ej.madams@gmail.com> <tmadams@google.com>
Yixuan Qiu <yixuanq@gmail.com>
Yusuke Suzuki <utatane.tea@gmail.com>
Zbigniew Skowron <zbychs@gmail.com>
Min-Yih Hsu <yihshyng223@gmail.com>
; NOTE: Assertions have been autogenerated by utils/update_llc_test_checks.py
; RUN: llc -mtriple=aarch64-apple-ios %s -o - | FileCheck %s

define <16 x double> @test_sitofp_fixed(<16 x i32> %in) {
; CHECK-LABEL: test_sitofp_fixed:
; CHECK: ; %bb.0:
; CHECK-NEXT: sshll2.2d v4, v2, #0
; CHECK-NEXT: sshll.2d v16, v1, #0
; CHECK-NEXT: sshll2.2d v5, v0, #0
; CHECK-NEXT: sshll2.2d v6, v1, #0
; CHECK-NEXT: sshll2.2d v7, v3, #0
; CHECK-NEXT: sshll.2d v0, v0, #0
; CHECK-NEXT: sshll.2d v17, v2, #0
; CHECK-NEXT: sshll.2d v18, v3, #0
; CHECK-NEXT: scvtf.2d v1, v5, #6
; CHECK-NEXT: scvtf.2d v3, v6, #6
; CHECK-NEXT: scvtf.2d v2, v16, #6
; CHECK-NEXT: scvtf.2d v5, v4, #6
; CHECK-NEXT: scvtf.2d v0, v0, #6
; CHECK-NEXT: scvtf.2d v7, v7, #6
; CHECK-NEXT: scvtf.2d v4, v17, #6
; CHECK-NEXT: scvtf.2d v6, v18, #6
; CHECK-NEXT: ret

 %flt = sitofp <16 x i32> %in to <16 x double>
 %res = fdiv <16 x double> %flt, <double 64.0, double 64.0, double 64.0,
 double 64.0, double 64.0, double 64.0, double 64.0, double 64.0, double 64.0, double 64.0, double
64.0, double 64.0, double 64.0, double 64.0, double 64.0>
 ret <16 x double> %res
}

```
; This one is small enough to satisfy isSimple, but still illegally large.
define <4 x double> @test_sitofp_fixed_shortish(<4 x i64> %in) {
; CHECK-LABEL: test_sitofp_fixed_shortish:
; CHECK:       ; %bb.0:
; CHECK-NEXT:    scvtf.2d v0, v0, #6
; CHECK-NEXT:    scvtf.2d v1, v1, #6
; CHECK-NEXT:    ret


 %flt = sitofp <4 x i64> %in to <4 x double>
 %res = fdiv <4 x double> %flt, <double 64.0, double 64.0, double 64.0, double 64.0>
 ret <4 x double> %res
}
LLVM System Interface Library
-------------------------------------------------------------------------------

Part of the LLVM Project, under the Apache License v2.0 with LLVM Exceptions.
See https://llvm.org/LICENSE.txt for license information.
SPDX-License-Identifier: Apache-2.0 WITH LLVM-exception
==============================================================================
The LLVM Project is under the Apache License v2.0 with LLVM Exceptions:
==============================================================================

                                 Apache License
                           Version 2.0, January 2004
                        http://www.apache.org/licenses/

   TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

   1. Definitions.

      "License" shall mean the terms and conditions for use, reproduction,
      and distribution as defined by Sections 1 through 9 of this document.

      "Licensor" shall mean the copyright owner or entity authorized by
      the copyright owner that is granting the License.

      "Legal Entity" shall mean the union of the acting entity and all
      other entities that control, are controlled by, or are under common
      control with that entity. For the purposes of this definition,
      "control" means (i) the
    power, direct or indirect, to cause the
      direction or management of such entity, whether by contract or
      otherwise, or (ii) ownership of fifty percent (50%) or more of the
      outstanding shares, or (iii) beneficial ownership of such entity.

      "You" (or "Your") shall mean an individual or Legal Entity
```

exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached
to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition,
"submitted"
means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative

Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work
includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents

of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional
attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License
does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including
negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,

work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

---- LLVM Exceptions to the Apache 2.0 License ----

As an exception, if, as a result of your compiling your source code, portions
of this Software are embedded into an Object form of such source code, you

may redistribute such embedded portions in such Object form without complying
with the conditions of Sections 4(a), 4(b) and 4(d) of the License.

In addition, if you combine or link compiled
 forms of this Software with
software that is licensed under the GPLv2 ("Combined Software") and if a
court of competent jurisdiction determines that the patent provision (Section
3), the indemnity provision (Section 9) or other Section of the License
conflicts with the conditions of the GPLv2, you may retroactively and
prospectively choose to deem waived or otherwise exclude such Section(s) of
the License, but only in their entirety and only with respect to the Combined
Software.

==
Software from third parties included in the LLVM Project:
==
The LLVM Project contains third party software which is under different license
terms. All such code will be identified clearly using at least one of two
mechanisms:
1) It will be in a separate directory tree with its own `LICENSE.txt` or
  `LICENSE` file at the top containing the specific
 license and restrictions
  which apply to that software, or
2) It will contain specific license and restriction terms at the top of every
  file.

==
Legacy LLVM License (https://llvm.org/docs/DeveloperPolicy.html#legacy):
==
University of Illinois/NCSA
Open Source License

Copyright (c) 2011-2019 by the contributors listed in CREDITS.TXT
All rights reserved.

Developed by:

  LLVM Team

  University of Illinois at Urbana-Champaign

  http://llvm.org

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal with
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies
of the Software, and to permit persons to whom the Software is furnished

to do
so, subject to the following conditions:

* Redistributions of source code must retain the above copyright notice,
this list of conditions and the following disclaimers.

* Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimers in the
documentation and/or other materials provided with the distribution.

* Neither the names of the LLVM Team, University of Illinois at
Urbana-Champaign, nor the names of its contributors may be used to
endorse or promote products derived from this Software without specific
prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

==========

The LLVM Project is under the Apache License v2.0 with LLVM Exceptions:

==========

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the
power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the

outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached
to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition,
"submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of

this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or

documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special,

incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

---- LLVM Exceptions to the Apache 2.0 License ----

As an exception, if, as a result of your compiling your source code, portions
of this Software are embedded into an Object form of such source code, you
may redistribute such embedded portions in such Object form without complying
with the conditions of Sections 4(a), 4(b) and 4(d) of the License.

In addition, if you combine or link compiled
 forms of this Software with
software that is licensed under the GPLv2 ("Combined Software") and if a
court of competent jurisdiction determines that the patent provision (Section
3), the indemnity provision (Section 9) or other Section of the License
conflicts with the conditions of the GPLv2, you may retroactively and
prospectively choose to deem waived or otherwise exclude such Section(s) of
the License, but only in their entirety and only with respect to the Combined
Software.

==
Software from third parties included in the LLVM Project:
==
The LLVM Project contains third party software which is under different license
terms. All such code will be identified clearly using at least one of two
mechanisms:
1) It will be in a separate directory tree with its own `LICENSE.txt` or
  `LICENSE` file at the top containing the specific
 license and restrictions
  which apply to that software, or
2) It will contain specific license and restriction terms at the top of every
  file.

==
Legacy LLVM License (https://llvm.org/docs/DeveloperPolicy.html#legacy):
==

The software contained in this directory tree is dual licensed under both the
University of Illinois "BSD-Like" license and the MIT license. As a user of
this code you may choose to use it under either license. As a contributor,
you agree to allow your code to be used under both. The full text of the
relevant licenses is included below.

In addition, a license agreement from the copyright/patent holders of the
software contained in this directory tree is included below.

==

University of Illinois/NCSA
Open Source License

Copyright

(c) 1997-2019 Intel Corporation

All rights reserved.

Developed by:
OpenMP Runtime Team
Intel Corporation
http://www.openmprtl.org

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal with the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimers.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimers in the documentation and/or other materials provided with the distribution.

* Neither the names of Intel Corporation OpenMP Runtime Team nor the names of its contributors may be used to endorse or promote products derived from this Software without specific prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

==========

Copyright (c) 1997-2019 Intel Corporation

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

==========

Intel Corporation

Software Grant License Agreement ("Agreement")

Except for the license granted herein to you, Intel Corporation ("Intel") reserves
all right, title, and interest in and to the Software (defined below).

Definition

"Software" means the code and documentation as well as any original work of authorship, including any modifications or additions to an existing work, that is intentionally submitted by Intel to llvm.org (http://llvm.org) ("LLVM") for inclusion in, or documentation of, any of the products owned or managed by LLVM (the "Work"). For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to LLVM or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, LLVM for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked otherwise.

1. Grant of Copyright License. Subject to the terms and conditions of this Agreement, Intel hereby grants to you and to recipients of the Software
distributed by LLVM a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, sublicense, and distribute the Software and such derivative works.

2. Grant of Patent License. Subject to the terms and conditions of this Agreement, Intel hereby grants you and to recipients of the Software distributed by LLVM a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable

by Intel that are necessarily infringed by Intel's Software alone or by
combination of the Software with the Work to which such Software was
submitted. If any entity institutes patent litigation against Intel or any
other entity (including
a cross-claim or counterclaim in a lawsuit) alleging
that Intel's Software, or the Work to which Intel has contributed constitutes
direct or contributory patent infringement, then any patent licenses granted
to that entity under this Agreement for the Software or Work shall terminate
as of the date such litigation is filed.

Unless required by applicable law or agreed to in writing, the software is
provided on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND,
either express or implied, including, without limitation, any warranties or
conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE.

==========

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership
of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial
revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this
License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable

(except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the
content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct,
indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,

and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
; NOTE: Assertions have been autogenerated by utils/update_llc_test_checks.py
; RUN: llc < %s -mtriple=x86_64-unknown-unknown -mcpu=skylake-avx512 -mattr=prefer-256-bit | FileCheck %s
--check-prefixes=CHECK,CHECK-AVX512
; RUN: llc < %s -mtriple=x86_64-unknown-unknown -mcpu=skylake-avx512 -mattr=prefer-256-bit,avx512vbmi |
FileCheck %s --check-prefixes=CHECK,CHECK-VBMI
; Make sure CPUs default to prefer-256-bit. avx512vnni isn't interesting as it just adds an isel peephole for
vpmaddwd+vpaddd
; RUN: llc < %s -mtriple=x86_64-unknown-unknown -mcpu=skylake-avx512 | FileCheck %s --check-
prefixes=CHECK,CHECK-AVX512
; RUN: llc < %s -mtriple=x86_64-unknown-unknown -mattr=-avx512vnni -mcpu=cascadelake | FileCheck %s --
check-prefixes=CHECK,CHECK-AVX512
; RUN: llc < %s -mtriple=x86_64-unknown-unknown -mattr=-avx512vnni -mcpu=cooperlake | FileCheck %s --
check-prefixes=CHECK,CHECK-AVX512

```
; RUN: llc < %s -mtriple=x86_64-unknown-unknown -mcpu=cannonlake | FileCheck %s --check-prefixes=CHECK,CHECK-VBMI
;
 RUN: llc < %s -mtriple=x86_64-unknown-unknown -mattr=-avx512vnni -mcpu=icelake-client | FileCheck %s --check-prefixes=CHECK,CHECK-VBMI
; RUN: llc < %s -mtriple=x86_64-unknown-unknown -mattr=-avx512vnni -mcpu=icelake-server | FileCheck %s --check-prefixes=CHECK,CHECK-VBMI
; RUN: llc < %s -mtriple=x86_64-unknown-unknown -mattr=-avx512vnni -mcpu=tigerlake | FileCheck %s --check-prefixes=CHECK,CHECK-VBMI

; This file primarily contains tests for specific places in X86ISelLowering.cpp that needed be made aware of the legalizer not allowing 512-bit vectors due to prefer-256-bit even though AVX512 is enabled.

define dso_local void @add256(<16 x i32>* %a, <16 x i32>* %b, <16 x i32>* %c) "min-legal-vector-width"="256" {
; CHECK-LABEL: add256:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vmovdqa (%rdi), %ymm0
; CHECK-NEXT:    vmovdqa 32(%rdi), %ymm1
; CHECK-NEXT:    vpaddd 32(%rsi), %ymm1, %ymm1
; CHECK-NEXT:    vpaddd (%rsi), %ymm0, %ymm0
; CHECK-NEXT:    vmovdqa %ymm0, (%rdx)
; CHECK-NEXT:
 vmovdqa %ymm1, 32(%rdx)
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
 %d = load <16 x i32>, <16 x i32>* %a
 %e = load <16 x i32>, <16 x i32>* %b
 %f = add <16 x i32> %d, %e
 store <16 x i32> %f, <16 x i32>* %c
 ret void
}

define dso_local void @add512(<16 x i32>* %a, <16 x i32>* %b, <16 x i32>* %c) "min-legal-vector-width"="512" {
; CHECK-LABEL: add512:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vmovdqa64 (%rdi), %zmm0
; CHECK-NEXT:    vpaddd (%rsi), %zmm0, %zmm0
; CHECK-NEXT:    vmovdqa64 %zmm0, (%rdx)
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
 %d = load <16 x i32>, <16 x i32>* %a
 %e = load <16 x i32>, <16 x i32>* %b
 %f = add <16 x i32> %d, %e
 store <16 x i32> %f, <16 x i32>* %c
 ret void
}
```

```
define dso_local void @avg_v64i8_256(<64 x i8>* %a, <64 x i8>* %b) "min-legal-vector-width"="256" {
; CHECK-LABEL: avg_v64i8_256:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vmovdqa (%rsi), %ymm0
; CHECK-NEXT:    vmovdqa 32(%rsi), %ymm1
; CHECK-NEXT:    vpavgb (%rdi), %ymm0,
 %ymm0
; CHECK-NEXT:    vpavgb 32(%rdi), %ymm1, %ymm1
; CHECK-NEXT:    vmovdqu %ymm1, (%rax)
; CHECK-NEXT:    vmovdqu %ymm0, (%rax)
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
 %1 = load <64 x i8>, <64 x i8>* %a
 %2 = load <64 x i8>, <64 x i8>* %b
 %3 = zext <64 x i8> %1 to <64 x i32>
 %4 = zext <64 x i8> %2 to <64 x i32>
 %5 = add nuw nsw <64 x i32> %3, <i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1,
i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32
1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1,
i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1>
 %6 = add nuw nsw <64 x i32> %5, %4
 %7 = lshr <64 x i32> %6, <i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1,
i32 1, i32 1, i32 1,
 i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1,
i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32
1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1>
 %8 = trunc <64 x i32> %7 to <64 x i8>
 store <64 x i8> %8, <64 x i8>* undef, align 4
 ret void
}


define dso_local void @avg_v64i8_512(<64 x i8>* %a, <64 x i8>* %b) "min-legal-vector-width"="512" {
; CHECK-LABEL: avg_v64i8_512:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vmovdqa64 (%rsi), %zmm0
; CHECK-NEXT:    vpavgb (%rdi), %zmm0, %zmm0
; CHECK-NEXT:    vmovdqu64 %zmm0, (%rax)
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
 %1 = load <64 x i8>, <64 x i8>* %a
 %2 = load <64 x i8>, <64 x i8>* %b
 %3 = zext <64 x i8> %1 to <64 x i32>
 %4 = zext <64 x i8> %2 to <64 x i32>
 %5 = add nuw nsw <64 x i32> %3, <i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32
 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1,
i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32
```

```
1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1,
i32 1>
 %6 = add nuw nsw <64 x i32> %5, %4
 %7 = lshr <64 x i32> %6, <i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1,
i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32
1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1,
i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1, i32 1>
 %8 = trunc <64 x i32> %7 to <64 x i8>
 store <64 x i8> %8, <64 x i8>* undef, align 4
  ret void
}

define dso_local void @pmaddwd_32_256(<32 x i16>* %APtr, <32 x i16>* %BPtr, <16 x i32>* %CPtr) "min-
legal-vector-width"="256" {
; CHECK-LABEL: pmaddwd_32_256:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vmovdqa (%rdi), %ymm0
; CHECK-NEXT:    vmovdqa 32(%rdi), %ymm1
; CHECK-NEXT:    vpmaddwd 32(%rsi), %ymm1, %ymm1
; CHECK-NEXT:    vpmaddwd (%rsi), %ymm0, %ymm0
; CHECK-NEXT:    vmovdqa %ymm0, (%rdx)
; CHECK-NEXT:    vmovdqa %ymm1, 32(%rdx)
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
  %A = load <32 x i16>, <32 x i16>* %APtr
  %B = load <32 x i16>, <32 x i16>* %BPtr
  %a = sext <32 x i16> %A to <32 x i32>
  %b = sext <32 x i16> %B to <32 x i32>
  %m = mul nsw <32 x i32> %a, %b
  %odd = shufflevector <32 x i32> %m, <32 x i32> undef, <16 x i32> <i32 0, i32 2, i32 4, i32 6, i32 8, i32 10, i32
12, i32 14, i32 16, i32 18, i32 20, i32 22, i32 24, i32 26, i32 28, i32 30>
  %even = shufflevector <32 x i32> %m, <32 x i32> undef, <16 x i32> <i32 1, i32 3, i32 5, i32 7, i32 9,
 i32 11, i32 13, i32 15, i32 17, i32 19, i32 21, i32 23, i32 25, i32 27, i32 29, i32 31>
  %ret = add <16 x i32> %odd, %even
  store <16 x i32> %ret, <16 x i32>* %CPtr
  ret void
}

define dso_local void @pmaddwd_32_512(<32 x i16>* %APtr, <32 x i16>* %BPtr, <16 x i32>* %CPtr) "min-
legal-vector-width"="512" {
; CHECK-LABEL: pmaddwd_32_512:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vmovdqa64 (%rdi), %zmm0
; CHECK-NEXT:    vpmaddwd (%rsi), %zmm0, %zmm0
; CHECK-NEXT:    vmovdqa64 %zmm0, (%rdx)
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
  %A = load <32 x i16>, <32 x i16>* %APtr
```

```
 %B = load <32 x i16>, <32 x i16>* %BPtr
 %a = sext <32 x i16> %A to <32 x i32>
 %b = sext <32 x i16> %B to <32 x i32>
 %m = mul nsw <32 x i32> %a, %b
 %odd = shufflevector <32 x i32> %m, <32 x i32> undef, <16 x i32> <i32 0, i32 2, i32 4, i32 6, i32 8, i32 10, i32
12, i32 14, i32 16, i32 18, i32 20, i32 22, i32 24, i32 26, i32 28, i32 30>
 %even = shufflevector <32 x i32> %m, <32 x i32> undef, <16
 x i32> <i32 1, i32 3, i32 5, i32 7, i32 9, i32 11, i32 13, i32 15, i32 17, i32 19, i32 21, i32 23, i32 25, i32 27, i32 29,
i32 31>
 %ret = add <16 x i32> %odd, %even
 store <16 x i32> %ret, <16 x i32>* %CPtr
 ret void
}

define dso_local void @psubus_64i8_max_256(<64 x i8>* %xptr, <64 x i8>* %yptr, <64 x i8>* %zptr) "min-legal-
vector-width"="256" {
; CHECK-LABEL: psubus_64i8_max_256:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vmovdqa (%rdi), %ymm0
; CHECK-NEXT:    vmovdqa 32(%rdi), %ymm1
; CHECK-NEXT:    vpsubusb 32(%rsi), %ymm1, %ymm1
; CHECK-NEXT:    vpsubusb (%rsi), %ymm0, %ymm0
; CHECK-NEXT:    vmovdqa %ymm0, (%rdx)
; CHECK-NEXT:    vmovdqa %ymm1, 32(%rdx)
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
 %x = load <64 x i8>, <64 x i8>* %xptr
 %y = load <64 x i8>, <64 x i8>* %yptr
 %cmp = icmp ult <64 x i8> %x, %y
 %max = select <64 x i1> %cmp, <64 x i8> %y, <64 x i8> %x
 %res = sub <64 x i8> %max, %y
 store <64 x i8> %res, <64 x i8>* %zptr
 ret void
}

define dso_local
 void @psubus_64i8_max_512(<64 x i8>* %xptr, <64 x i8>* %yptr, <64 x i8>* %zptr) "min-legal-vector-
width"="512" {
; CHECK-LABEL: psubus_64i8_max_512:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vmovdqa64 (%rdi), %zmm0
; CHECK-NEXT:    vpsubusb (%rsi), %zmm0, %zmm0
; CHECK-NEXT:    vmovdqa64 %zmm0, (%rdx)
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
 %x = load <64 x i8>, <64 x i8>* %xptr
 %y = load <64 x i8>, <64 x i8>* %yptr
 %cmp = icmp ult <64 x i8> %x, %y
```

```
 %max = select <64 x i1> %cmp, <64 x i8> %y, <64 x i8> %x
 %res = sub <64 x i8> %max, %y
 store <64 x i8> %res, <64 x i8>* %zptr
 ret void
}

define dso_local i32 @_Z9test_charPcS_i_256(i8* nocapture readonly, i8* nocapture readonly, i32) "min-legal-vector-width"="256" {
; CHECK-LABEL: _Z9test_charPcS_i_256:
; CHECK:       # %bb.0: # %entry
; CHECK-NEXT:    movl %edx, %eax
; CHECK-NEXT:    vpxor %xmm0, %xmm0, %xmm0
; CHECK-NEXT:    xorl %ecx, %ecx
; CHECK-NEXT:    vpxor %xmm1, %xmm1, %xmm1
; CHECK-NEXT:    vpxor %xmm2,
 %xmm2, %xmm2
; CHECK-NEXT:    .p2align 4, 0x90
; CHECK-NEXT:  .LBB8_1: # %vector.body
; CHECK-NEXT:    # =>This Inner Loop Header: Depth=1
; CHECK-NEXT:    vpmovsxbw 16(%rdi,%rcx), %ymm3
; CHECK-NEXT:    vpmovsxbw (%rdi,%rcx), %ymm4
; CHECK-NEXT:    vpmovsxbw 16(%rsi,%rcx), %ymm5
; CHECK-NEXT:    vpmaddwd %ymm3, %ymm5, %ymm3
; CHECK-NEXT:    vpaddd %ymm2, %ymm3, %ymm2
; CHECK-NEXT:    vpmovsxbw (%rsi,%rcx), %ymm3
; CHECK-NEXT:    vpmaddwd %ymm4, %ymm3, %ymm3
; CHECK-NEXT:    vpaddd %ymm1, %ymm3, %ymm1
; CHECK-NEXT:    addq $32, %rcx
; CHECK-NEXT:    cmpq %rcx, %rax
; CHECK-NEXT:    jne .LBB8_1
; CHECK-NEXT:  # %bb.2: # %middle.block
; CHECK-NEXT:    vpaddd %ymm0, %ymm1, %ymm1
; CHECK-NEXT:    vpaddd %ymm0, %ymm2, %ymm0
; CHECK-NEXT:    vpaddd %ymm0, %ymm1, %ymm0
; CHECK-NEXT:    vextracti128 $1, %ymm0, %xmm1
; CHECK-NEXT:    vpaddd %xmm1, %xmm0, %xmm0
; CHECK-NEXT:    vpshufd {{.*#+}} xmm1 = xmm0[2,3,2,3]
; CHECK-NEXT:    vpaddd %xmm1, %xmm0, %xmm0
; CHECK-NEXT:    vpshufd {{.*#+}} xmm1
 = xmm0[1,1,1,1]
; CHECK-NEXT:    vpaddd %xmm1, %xmm0, %xmm0
; CHECK-NEXT:    vmovd %xmm0, %eax
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
entry:
 %3 = zext i32 %2 to i64
 br label %vector.body
```

```
vector.body:
 %index = phi i64 [ %index.next, %vector.body ], [ 0, %entry ]
 %vec.phi = phi <32 x i32> [ %11, %vector.body ], [ zeroinitializer, %entry ]
 %4 = getelementptr inbounds i8, i8* %0, i64 %index
 %5 = bitcast i8* %4 to <32 x i8>*
 %wide.load = load <32 x i8>, <32 x i8>* %5, align 1
 %6 = sext <32 x i8> %wide.load to <32 x i32>
 %7 = getelementptr inbounds i8, i8* %1, i64 %index
 %8 = bitcast i8* %7 to <32 x i8>*
 %wide.load14 = load <32 x i8>, <32 x i8>* %8, align 1
 %9 = sext <32 x i8> %wide.load14 to <32 x i32>
 %10 = mul nsw <32 x i32> %9, %6
 %11 = add nsw <32 x i32> %10, %vec.phi
 %index.next = add i64 %index, 32
 %12 = icmp eq i64 %index.next, %3
 br i1 %12, label %middle.block, label %vector.body

middle.block:
 %rdx.shuf1 = shufflevector <32 x i32> %11,
 <32 x i32> undef, <32 x i32> <i32 16, i32 17, i32 18, i32 19, i32 20, i32 21, i32 22, i32 23, i32 24, i32 25, i32 26,
i32 27, i32 28, i32 29, i32 30, i32 31, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32
undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
 %bin.rdx1 = add <32 x i32> %11, %rdx.shuf1
 %rdx.shuf = shufflevector <32 x i32> %bin.rdx1, <32 x i32> undef, <32 x i32> <i32 8, i32 9, i32 10, i32 11, i32
12, i32 13, i32 14, i32 15, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32
undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef,
i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
 %bin.rdx = add <32 x i32> %bin.rdx1, %rdx.shuf
 %rdx.shuf15 = shufflevector <32 x i32> %bin.rdx, <32 x i32> undef, <32 x i32> <i32 4, i32 5, i32 6, i32 7, i32
undef, i32 undef, i32 undef, i32 undef, i32 undef,
 i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32
undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef,
i32 undef, i32 undef>
 %bin.rdx32 = add <32 x i32> %bin.rdx, %rdx.shuf15
 %rdx.shuf17 = shufflevector <32 x i32> %bin.rdx32, <32 x i32> undef, <32 x i32> <i32 2, i32 3, i32 undef, i32
undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef,
i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32
undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
 %bin.rdx18 = add <32 x i32> %bin.rdx32, %rdx.shuf17
 %rdx.shuf19 = shufflevector <32 x i32> %bin.rdx18, <32 x i32> undef, <32 x i32> <i32 1, i32 undef, i32 undef,
i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32
 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef,
i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
 %bin.rdx20 = add <32 x i32> %bin.rdx18, %rdx.shuf19
 %13 = extractelement <32 x i32> %bin.rdx20, i32 0
 ret i32 %13
}
```

```
define dso_local i32 @_Z9test_charPcS_i_512(i8* nocapture readonly, i8* nocapture readonly, i32) "min-legal-vector-width"="512" {
; CHECK-LABEL: _Z9test_charPcS_i_512:
; CHECK:       # %bb.0: # %entry
; CHECK-NEXT:    movl %edx, %eax
; CHECK-NEXT:    vpxor %xmm0, %xmm0, %xmm0
; CHECK-NEXT:    xorl %ecx, %ecx
; CHECK-NEXT:    vpxor %xmm1, %xmm1, %xmm1
; CHECK-NEXT:    .p2align 4, 0x90
; CHECK-NEXT:  .LBB9_1: # %vector.body
; CHECK-NEXT:    # =>This Inner Loop Header: Depth=1
; CHECK-NEXT:    vpmovsxbw (%rdi,%rcx), %zmm2
; CHECK-NEXT:    vpmovsxbw (%rsi,%rcx), %zmm3
; CHECK-NEXT:    vpmaddwd %zmm2, %zmm3, %zmm2
; CHECK-NEXT:    vpaddd %zmm1, %zmm2, %zmm1
;
 CHECK-NEXT:    addq $32, %rcx
; CHECK-NEXT:    cmpq %rcx, %rax
; CHECK-NEXT:    jne .LBB9_1
; CHECK-NEXT:  # %bb.2: # %middle.block
; CHECK-NEXT:    vpaddd %zmm0, %zmm1, %zmm0
; CHECK-NEXT:    vextracti64x4 $1, %zmm0, %ymm1
; CHECK-NEXT:    vpaddd %zmm1, %zmm0, %zmm0
; CHECK-NEXT:    vextracti128 $1, %ymm0, %xmm1
; CHECK-NEXT:    vpaddd %xmm1, %xmm0, %xmm0
; CHECK-NEXT:    vpshufd {{.*#+}} xmm1 = xmm0[2,3,2,3]
; CHECK-NEXT:    vpaddd %xmm1, %xmm0, %xmm0
; CHECK-NEXT:    vpshufd {{.*#+}} xmm1 = xmm0[1,1,1,1]
; CHECK-NEXT:    vpaddd %xmm1, %xmm0, %xmm0
; CHECK-NEXT:    vmovd %xmm0, %eax
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
entry:
 %3 = zext i32 %2 to i64
 br label %vector.body

vector.body:
 %index = phi i64 [ %index.next, %vector.body ], [ 0, %entry ]
 %vec.phi = phi <32 x i32> [ %11, %vector.body ], [ zeroinitializer, %entry ]
 %4 = getelementptr inbounds i8, i8* %0, i64 %index
 %5 = bitcast i8* %4 to <32 x i8>*
 %wide.load = load <32 x i8>, <32 x i8>* %5, align
 1
 %6 = sext <32 x i8> %wide.load to <32 x i32>
 %7 = getelementptr inbounds i8, i8* %1, i64 %index
 %8 = bitcast i8* %7 to <32 x i8>*
 %wide.load14 = load <32 x i8>, <32 x i8>* %8, align 1
 %9 = sext <32 x i8> %wide.load14 to <32 x i32>
```

```
 %10 = mul nsw <32 x i32> %9, %6
 %11 = add nsw <32 x i32> %10, %vec.phi
 %index.next = add i64 %index, 32
 %12 = icmp eq i64 %index.next, %3
 br i1 %12, label %middle.block, label %vector.body

middle.block:
 %rdx.shuf1 = shufflevector <32 x i32> %11, <32 x i32> undef, <32 x i32> <i32 16, i32 17, i32 18, i32 19, i32 20,
i32 21, i32 22, i32 23, i32 24, i32 25, i32 26, i32 27, i32 28, i32 29, i32 30, i32 31, i32 undef, i32 undef, i32 undef,
i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32
undef, i32 undef, i32 undef>
 %bin.rdx1 = add <32 x i32> %11, %rdx.shuf1
 %rdx.shuf = shufflevector <32 x i32> %bin.rdx1, <32 x i32> undef, <32 x i32> <i32 8, i32 9, i32 10, i32 11, i32
12,
 i32 13, i32 14, i32 15, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32
undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef,
i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
 %bin.rdx = add <32 x i32> %bin.rdx1, %rdx.shuf
 %rdx.shuf15 = shufflevector <32 x i32> %bin.rdx, <32 x i32> undef, <32 x i32> <i32 4, i32 5, i32 6, i32 7, i32
undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef,
i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32
undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
 %bin.rdx32 = add <32 x i32> %bin.rdx, %rdx.shuf15
 %rdx.shuf17 = shufflevector <32 x i32> %bin.rdx32, <32 x i32> undef, <32 x i32> <i32 2, i32 3, i32 undef, i32
undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef,
 i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32
undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
 %bin.rdx18 = add <32 x i32> %bin.rdx32, %rdx.shuf17
 %rdx.shuf19 = shufflevector <32 x i32> %bin.rdx18, <32 x i32> undef, <32 x i32> <i32 1, i32 undef, i32 undef,
i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32
undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef,
i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
 %bin.rdx20 = add <32 x i32> %bin.rdx18, %rdx.shuf19
 %13 = extractelement <32 x i32> %bin.rdx20, i32 0
 ret i32 %13
}

@a = dso_local global [1024 x i8] zeroinitializer, align 16
@b = dso_local global [1024 x i8] zeroinitializer, align 16

define dso_local i32 @sad_16i8_256()
 "min-legal-vector-width"="256" {
; CHECK-LABEL: sad_16i8_256:
; CHECK:       # %bb.0: # %entry
; CHECK-NEXT:    vpxor %xmm0, %xmm0, %xmm0
; CHECK-NEXT:    movq $-1024, %rax # imm = 0xFC00
; CHECK-NEXT:    vpxor %xmm1, %xmm1, %xmm1
; CHECK-NEXT:    .p2align 4, 0x90
```

```
; CHECK-NEXT:  .LBB10_1: # %vector.body
; CHECK-NEXT:    # =>This Inner Loop Header: Depth=1
; CHECK-NEXT:    vmovdqu a+1024(%rax), %xmm2
; CHECK-NEXT:    vpsadbw b+1024(%rax), %xmm2, %xmm2
; CHECK-NEXT:    vpaddd %ymm1, %ymm2, %ymm1
; CHECK-NEXT:    addq $4, %rax
; CHECK-NEXT:    jne .LBB10_1
; CHECK-NEXT:  # %bb.2: # %middle.block
; CHECK-NEXT:    vpaddd %ymm0, %ymm1, %ymm0
; CHECK-NEXT:    vextracti128 $1, %ymm0, %xmm1
; CHECK-NEXT:    vpaddd %xmm1, %xmm0, %xmm0
; CHECK-NEXT:    vpshufd {{.*#+}} xmm1 = xmm0[2,3,2,3]
; CHECK-NEXT:    vpaddd %xmm1, %xmm0, %xmm0
; CHECK-NEXT:    vpshufd {{.*#+}} xmm1 = xmm0[1,1,1,1]
; CHECK-NEXT:    vpaddd %xmm1, %xmm0, %xmm0
; CHECK-NEXT:    vmovd %xmm0, %eax
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:
   retq
entry:
 br label %vector.body

vector.body:
 %index = phi i64 [ 0, %entry ], [ %index.next, %vector.body ]
 %vec.phi = phi <16 x i32> [ zeroinitializer, %entry ], [ %10, %vector.body ]
 %0 = getelementptr inbounds [1024 x i8], [1024 x i8]* @a, i64 0, i64 %index
 %1 = bitcast i8* %0 to <16 x i8>*
 %wide.load = load <16 x i8>, <16 x i8>* %1, align 4
 %2 = zext <16 x i8> %wide.load to <16 x i32>
 %3 = getelementptr inbounds [1024 x i8], [1024 x i8]* @b, i64 0, i64 %index
 %4 = bitcast i8* %3 to <16 x i8>*
 %wide.load1 = load <16 x i8>, <16 x i8>* %4, align 4
 %5 = zext <16 x i8> %wide.load1 to <16 x i32>
 %6 = sub nsw <16 x i32> %2, %5
 %7 = icmp sgt <16 x i32> %6, <i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1>
 %8 = sub nsw <16 x i32> zeroinitializer, %6
 %9 = select <16 x i1> %7, <16 x i32> %6, <16 x i32> %8
 %10 = add nsw <16 x i32> %9, %vec.phi
 %index.next = add i64 %index,
 4
 %11 = icmp eq i64 %index.next, 1024
 br i1 %11, label %middle.block, label %vector.body

middle.block:
 %rdx.shuf = shufflevector <16 x i32> %10, <16 x i32> undef, <16 x i32> <i32 8, i32 9, i32 10, i32 11, i32 12, i32 13, i32 14, i32 15, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
 %bin.rdx = add <16 x i32> %10, %rdx.shuf
```

```
 %rdx.shuf2 = shufflevector <16 x i32> %bin.rdx, <16 x i32> undef, <16 x i32> <i32 4, i32 5, i32 6, i32 7, i32
undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef,
i32 undef>
 %bin.rdx2 = add <16 x i32> %bin.rdx, %rdx.shuf2
 %rdx.shuf3 = shufflevector <16 x i32> %bin.rdx2, <16 x i32> undef, <16 x i32> <i32 2, i32 3, i32 undef, i32
undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef,
i32 undef, i32 undef>
 %bin.rdx3 = add <16 x i32> %bin.rdx2, %rdx.shuf3
 %rdx.shuf4 = shufflevector
 <16 x i32> %bin.rdx3, <16 x i32> undef, <16 x i32> <i32 1, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef,
i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
 %bin.rdx4 = add <16 x i32> %bin.rdx3, %rdx.shuf4
 %12 = extractelement <16 x i32> %bin.rdx4, i32 0
 ret i32 %12
}

define dso_local i32 @sad_16i8_512() "min-legal-vector-width"="512" {
; CHECK-LABEL: sad_16i8_512:
; CHECK:       # %bb.0: # %entry
; CHECK-NEXT:    vpxor %xmm0, %xmm0, %xmm0
; CHECK-NEXT:    movq $-1024, %rax # imm = 0xFC00
; CHECK-NEXT:    .p2align 4, 0x90
; CHECK-NEXT:  .LBB11_1: # %vector.body
; CHECK-NEXT:    # =>This Inner Loop Header: Depth=1
; CHECK-NEXT:    vmovdqu a+1024(%rax), %xmm1
; CHECK-NEXT:    vpsadbw b+1024(%rax), %xmm1, %xmm1
; CHECK-NEXT:    vpaddd %zmm0, %zmm1, %zmm0
; CHECK-NEXT:    addq $4, %rax
; CHECK-NEXT:    jne .LBB11_1
; CHECK-NEXT:  # %bb.2: # %middle.block
; CHECK-NEXT:    vextracti64x4 $1, %zmm0, %ymm1
; CHECK-NEXT:
   vpaddd %zmm1, %zmm0, %zmm0
; CHECK-NEXT:    vextracti128 $1, %ymm0, %xmm1
; CHECK-NEXT:    vpaddd %xmm1, %xmm0, %xmm0
; CHECK-NEXT:    vpshufd {{.*#+}} xmm1 = xmm0[2,3,2,3]
; CHECK-NEXT:    vpaddd %xmm1, %xmm0, %xmm0
; CHECK-NEXT:    vpshufd {{.*#+}} xmm1 = xmm0[1,1,1,1]
; CHECK-NEXT:    vpaddd %xmm1, %xmm0, %xmm0
; CHECK-NEXT:    vmovd %xmm0, %eax
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
entry:
 br label %vector.body

vector.body:
 %index = phi i64 [ 0, %entry ], [ %index.next, %vector.body ]
 %vec.phi = phi <16 x i32> [ zeroinitializer, %entry ], [ %10, %vector.body ]
```

```
 %0 = getelementptr inbounds [1024 x i8], [1024 x i8]* @a, i64 0, i64 %index
 %1 = bitcast i8* %0 to <16 x i8>*
 %wide.load = load <16 x i8>, <16 x i8>* %1, align 4
 %2 = zext <16 x i8> %wide.load to <16 x i32>
 %3 = getelementptr inbounds [1024 x i8], [1024 x i8]* @b, i64 0, i64 %index
 %4 = bitcast i8* %3 to <16 x i8>*
 %wide.load1 = load <16 x i8>, <16 x i8>* %4, align 4
 %5 = zext <16 x i8> %wide.load1
 to <16 x i32>
 %6 = sub nsw <16 x i32> %2, %5
 %7 = icmp sgt <16 x i32> %6, <i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1, i32 -1>
 %8 = sub nsw <16 x i32> zeroinitializer, %6
 %9 = select <16 x i1> %7, <16 x i32> %6, <16 x i32> %8
 %10 = add nsw <16 x i32> %9, %vec.phi
 %index.next = add i64 %index, 4
 %11 = icmp eq i64 %index.next, 1024
 br i1 %11, label %middle.block, label %vector.body

middle.block:
 %rdx.shuf = shufflevector <16 x i32> %10, <16 x i32> undef, <16 x i32> <i32 8, i32 9, i32 10, i32 11, i32 12, i32 13, i32 14, i32 15, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
 %bin.rdx = add <16 x i32> %10, %rdx.shuf
 %rdx.shuf2 = shufflevector <16 x i32> %bin.rdx, <16 x i32> undef, <16 x i32> <i32 4, i32 5, i32 6, i32 7, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
  %bin.rdx2 = add <16 x i32> %bin.rdx, %rdx.shuf2
 %rdx.shuf3 = shufflevector <16 x i32> %bin.rdx2, <16 x i32> undef, <16 x i32> <i32 2, i32 3, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
 %bin.rdx3 = add <16 x i32> %bin.rdx2, %rdx.shuf3
 %rdx.shuf4 = shufflevector <16 x i32> %bin.rdx3, <16 x i32> undef, <16 x i32> <i32 1, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef, i32 undef>
 %bin.rdx4 = add <16 x i32> %bin.rdx3, %rdx.shuf4
 %12 = extractelement <16 x i32> %bin.rdx4, i32 0
 ret i32 %12
}

define dso_local void @sbto16f32_256(<16 x i16> %a, <16 x float>* %res) "min-legal-vector-width"="256" {
; CHECK-LABEL: sbto16f32_256:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vpmovw2m %ymm0, %k0
; CHECK-NEXT:    kshiftrw $8, %k0, %k1
; CHECK-NEXT:    vpmovm2d %k1, %ymm0
;
 CHECK-NEXT:    vcvtdq2ps %ymm0, %ymm0
; CHECK-NEXT:    vpmovm2d %k0, %ymm1
```

```
; CHECK-NEXT:    vcvtdq2ps %ymm1, %ymm1
; CHECK-NEXT:    vmovaps %ymm1, (%rdi)
; CHECK-NEXT:    vmovaps %ymm0, 32(%rdi)
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
 %mask = icmp slt <16 x i16> %a, zeroinitializer
 %1 = sitofp <16 x i1> %mask to <16 x float>
 store <16 x float> %1, <16 x float>* %res
 ret void
}

define dso_local void @sbto16f32_512(<16 x i16> %a, <16 x float>* %res) "min-legal-vector-width"="512" {
; CHECK-LABEL: sbto16f32_512:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vpmovw2m %ymm0, %k0
; CHECK-NEXT:    vpmovm2d %k0, %zmm0
; CHECK-NEXT:    vcvtdq2ps %zmm0, %zmm0
; CHECK-NEXT:    vmovaps %zmm0, (%rdi)
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
 %mask = icmp slt <16 x i16> %a, zeroinitializer
 %1 = sitofp <16 x i1> %mask to <16 x float>
 store <16 x float> %1, <16 x float>* %res
 ret void
}

define dso_local void @sbto16f64_256(<16 x i16> %a, <16 x double>* %res)  "min-legal-vector-width"="256"
 {
; CHECK-LABEL: sbto16f64_256:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vpmovw2m %ymm0, %k0
; CHECK-NEXT:    kshiftrw $8, %k0, %k1
; CHECK-NEXT:    vpmovm2d %k1, %ymm0
; CHECK-NEXT:    vcvtdq2pd %xmm0, %ymm1
; CHECK-NEXT:    vextracti128 $1, %ymm0, %xmm0
; CHECK-NEXT:    vcvtdq2pd %xmm0, %ymm0
; CHECK-NEXT:    vpmovm2d %k0, %ymm2
; CHECK-NEXT:    vcvtdq2pd %xmm2, %ymm3
; CHECK-NEXT:    vextracti128 $1, %ymm2, %xmm2
; CHECK-NEXT:    vcvtdq2pd %xmm2, %ymm2
; CHECK-NEXT:    vmovaps %ymm2, 32(%rdi)
; CHECK-NEXT:    vmovaps %ymm3, (%rdi)
; CHECK-NEXT:    vmovaps %ymm0, 96(%rdi)
; CHECK-NEXT:    vmovaps %ymm1, 64(%rdi)
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
 %mask = icmp slt <16 x i16> %a, zeroinitializer
 %1 = sitofp <16 x i1> %mask to <16 x double>
```

```
 store <16 x double> %1, <16 x double>* %res
 ret void
}

define dso_local void @sbto16f64_512(<16 x i16> %a, <16 x double>* %res)  "min-legal-vector-width"="512" {
; CHECK-LABEL: sbto16f64_512:
; CHECK:
   # %bb.0:
; CHECK-NEXT:    vpmovw2m %ymm0, %k0
; CHECK-NEXT:    vpmovm2d %k0, %zmm0
; CHECK-NEXT:    vcvtdq2pd %ymm0, %zmm1
; CHECK-NEXT:    vextracti64x4 $1, %zmm0, %ymm0
; CHECK-NEXT:    vcvtdq2pd %ymm0, %zmm0
; CHECK-NEXT:    vmovaps %zmm0, 64(%rdi)
; CHECK-NEXT:    vmovaps %zmm1, (%rdi)
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
 %mask = icmp slt <16 x i16> %a, zeroinitializer
 %1 = sitofp <16 x i1> %mask to <16 x double>
 store <16 x double> %1, <16 x double>* %res
 ret void
}

define dso_local void @ubto16f32_256(<16 x i16> %a, <16 x float>* %res) "min-legal-vector-width"="256" {
; CHECK-LABEL: ubto16f32_256:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vpmovw2m %ymm0, %k0
; CHECK-NEXT:    kshiftrw $8, %k0, %k1
; CHECK-NEXT:    vpmovm2d %k1, %ymm0
; CHECK-NEXT:    vpsrld $31, %ymm0, %ymm0
; CHECK-NEXT:    vcvtdq2ps %ymm0, %ymm0
; CHECK-NEXT:    vpmovm2d %k0, %ymm1
; CHECK-NEXT:    vpsrld $31, %ymm1, %ymm1
; CHECK-NEXT:    vcvtdq2ps %ymm1, %ymm1
; CHECK-NEXT:    vmovaps %ymm1,
 (%rdi)
; CHECK-NEXT:    vmovaps %ymm0, 32(%rdi)
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
 %mask = icmp slt <16 x i16> %a, zeroinitializer
 %1 = uitofp <16 x i1> %mask to <16 x float>
 store <16 x float> %1, <16 x float>* %res
 ret void
}

define dso_local void @ubto16f32_512(<16 x i16> %a, <16 x float>* %res) "min-legal-vector-width"="512" {
; CHECK-LABEL: ubto16f32_512:
; CHECK:       # %bb.0:
```

```
; CHECK-NEXT:    vpmovw2m %ymm0, %k0
; CHECK-NEXT:    vpmovm2d %k0, %zmm0
; CHECK-NEXT:    vpsrld $31, %zmm0, %zmm0
; CHECK-NEXT:    vcvtdq2ps %zmm0, %zmm0
; CHECK-NEXT:    vmovaps %zmm0, (%rdi)
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
 %mask = icmp slt <16 x i16> %a, zeroinitializer
 %1 = uitofp <16 x i1> %mask to <16 x float>
 store <16 x float> %1, <16 x float>* %res
 ret void
}

define dso_local void @ubto16f64_256(<16 x i16> %a, <16 x double>* %res) "min-legal-vector-width"="256" {
; CHECK-LABEL: ubto16f64_256:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vpmovw2m
 %ymm0, %k0
; CHECK-NEXT:    kshiftrw $8, %k0, %k1
; CHECK-NEXT:    vpmovm2d %k1, %ymm0
; CHECK-NEXT:    vpsrld $31, %ymm0, %ymm0
; CHECK-NEXT:    vcvtdq2pd %xmm0, %ymm1
; CHECK-NEXT:    vextracti128 $1, %ymm0, %xmm0
; CHECK-NEXT:    vcvtdq2pd %xmm0, %ymm0
; CHECK-NEXT:    vpmovm2d %k0, %ymm2
; CHECK-NEXT:    vpsrld $31, %ymm2, %ymm2
; CHECK-NEXT:    vcvtdq2pd %xmm2, %ymm3
; CHECK-NEXT:    vextracti128 $1, %ymm2, %xmm2
; CHECK-NEXT:    vcvtdq2pd %xmm2, %ymm2
; CHECK-NEXT:    vmovaps %ymm2, 32(%rdi)
; CHECK-NEXT:    vmovaps %ymm3, (%rdi)
; CHECK-NEXT:    vmovaps %ymm0, 96(%rdi)
; CHECK-NEXT:    vmovaps %ymm1, 64(%rdi)
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
 %mask = icmp slt <16 x i16> %a, zeroinitializer
 %1 = uitofp <16 x i1> %mask to <16 x double>
 store <16 x double> %1, <16 x double>* %res
 ret void
}

define dso_local void @ubto16f64_512(<16 x i16> %a, <16 x double>* %res) "min-legal-vector-width"="512" {
; CHECK-LABEL: ubto16f64_512:
; CHECK:       # %bb.0:
; CHECK-NEXT:
   vpmovw2m %ymm0, %k0
; CHECK-NEXT:    vpmovm2d %k0, %zmm0
; CHECK-NEXT:    vpsrld $31, %zmm0, %zmm0
```

```
; CHECK-NEXT:    vcvtdq2pd %ymm0, %zmm1
; CHECK-NEXT:    vextracti64x4 $1, %zmm0, %ymm0
; CHECK-NEXT:    vcvtdq2pd %ymm0, %zmm0
; CHECK-NEXT:    vmovaps %zmm0, 64(%rdi)
; CHECK-NEXT:    vmovaps %zmm1, (%rdi)
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
 %mask = icmp slt <16 x i16> %a, zeroinitializer
 %1 = uitofp <16 x i1> %mask to <16 x double>
 store <16 x double> %1, <16 x double>* %res
 ret void
}

define <16 x i16> @test_16f32toub_256(<16 x float>* %ptr, <16 x i16> %passthru) "min-legal-vector-
width"="256" {
; CHECK-LABEL: test_16f32toub_256:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vcvttps2dq (%rdi), %ymm1
; CHECK-NEXT:    vpslld $31, %ymm1, %ymm1
; CHECK-NEXT:    vpmovd2m %ymm1, %k0
; CHECK-NEXT:    vcvttps2dq 32(%rdi), %ymm1
; CHECK-NEXT:    vpslld $31, %ymm1, %ymm1
; CHECK-NEXT:    vpmovd2m %ymm1, %k1
; CHECK-NEXT:    kunpckbw %k0, %k1, %k1
; CHECK-NEXT:    vmovdqu16
 %ymm0, %ymm0 {%k1} {z}
; CHECK-NEXT:    retq
 %a = load <16 x float>, <16 x float>* %ptr
 %mask = fptoui <16 x float> %a to <16 x i1>
 %select = select <16 x i1> %mask, <16 x i16> %passthru, <16 x i16> zeroinitializer
 ret <16 x i16> %select
}

define <16 x i16> @test_16f32toub_512(<16 x float>* %ptr, <16 x i16> %passthru) "min-legal-vector-
width"="512" {
; CHECK-LABEL: test_16f32toub_512:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vcvttps2dq (%rdi), %zmm1
; CHECK-NEXT:    vpslld $31, %zmm1, %zmm1
; CHECK-NEXT:    vpmovd2m %zmm1, %k1
; CHECK-NEXT:    vmovdqu16 %ymm0, %ymm0 {%k1} {z}
; CHECK-NEXT:    retq
 %a = load <16 x float>, <16 x float>* %ptr
 %mask = fptoui <16 x float> %a to <16 x i1>
 %select = select <16 x i1> %mask, <16 x i16> %passthru, <16 x i16> zeroinitializer
 ret <16 x i16> %select
}
```

```
define <16 x i16> @test_16f32tosb_256(<16 x float>* %ptr, <16 x i16> %passthru) "min-legal-vector-width"="256" {
; CHECK-LABEL: test_16f32tosb_256:
; CHECK:       # %bb.0:
; CHECK-NEXT:
   vcvttps2dq (%rdi), %ymm1
; CHECK-NEXT:    vpmovd2m %ymm1, %k0
; CHECK-NEXT:    vcvttps2dq 32(%rdi), %ymm1
; CHECK-NEXT:    vpmovd2m %ymm1, %k1
; CHECK-NEXT:    kunpckbw %k0, %k1, %k1
; CHECK-NEXT:    vmovdqu16 %ymm0, %ymm0 {%k1} {z}
; CHECK-NEXT:    retq
 %a = load <16 x float>, <16 x float>* %ptr
 %mask = fptosi <16 x float> %a to <16 x i1>
 %select = select <16 x i1> %mask, <16 x i16> %passthru, <16 x i16> zeroinitializer
 ret <16 x i16> %select
}

define <16 x i16> @test_16f32tosb_512(<16 x float>* %ptr, <16 x i16> %passthru) "min-legal-vector-width"="512" {
; CHECK-LABEL: test_16f32tosb_512:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vcvttps2dq (%rdi), %zmm1
; CHECK-NEXT:    vpmovd2m %zmm1, %k1
; CHECK-NEXT:    vmovdqu16 %ymm0, %ymm0 {%k1} {z}
; CHECK-NEXT:    retq
 %a = load <16 x float>, <16 x float>* %ptr
 %mask = fptosi <16 x float> %a to <16 x i1>
 %select = select <16 x i1> %mask, <16 x i16> %passthru, <16 x i16> zeroinitializer
 ret <16 x i16> %select
}

define
 dso_local void @mul256(<64 x i8>* %a, <64 x i8>* %b, <64 x i8>* %c) "min-legal-vector-width"="256" {
; CHECK-AVX512-LABEL: mul256:
; CHECK-AVX512:       # %bb.0:
; CHECK-AVX512-NEXT:    vmovdqa (%rdi), %ymm0
; CHECK-AVX512-NEXT:    vmovdqa 32(%rdi), %ymm1
; CHECK-AVX512-NEXT:    vmovdqa (%rsi), %ymm2
; CHECK-AVX512-NEXT:    vmovdqa 32(%rsi), %ymm3
; CHECK-AVX512-NEXT:    vpunpckhbw {{.*#+}} ymm4 =
ymm3[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31]
; CHECK-AVX512-NEXT:    vpunpckhbw {{.*#+}} ymm5 =
ymm1[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31]
; CHECK-AVX512-NEXT:    vpmullw %ymm4, %ymm5, %ymm4
; CHECK-AVX512-NEXT:    vmovdqa {{.*#+}} ymm5 =
[255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255]
; CHECK-AVX512-NEXT:    vpand %ymm5, %ymm4, %ymm4
```

```
; CHECK-AVX512-NEXT:    vpunpcklbw {{.*#+}} ymm3 =
ymm3[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23]
;
 CHECK-AVX512-NEXT:    vpunpcklbw {{.*#+}} ymm1 =
ymm1[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23]
; CHECK-AVX512-NEXT:    vpmullw %ymm3, %ymm1, %ymm1
; CHECK-AVX512-NEXT:    vpand %ymm5, %ymm1, %ymm1
; CHECK-AVX512-NEXT:    vpackuswb %ymm4, %ymm1, %ymm1
; CHECK-AVX512-NEXT:    vpunpckhbw {{.*#+}} ymm3 =
ymm2[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31]
; CHECK-AVX512-NEXT:    vpunpckhbw {{.*#+}} ymm4 =
ymm0[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31]
; CHECK-AVX512-NEXT:    vpmullw %ymm3, %ymm4, %ymm3
; CHECK-AVX512-NEXT:    vpand %ymm5, %ymm3, %ymm3
; CHECK-AVX512-NEXT:    vpunpcklbw {{.*#+}} ymm2 =
ymm2[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23]
; CHECK-AVX512-NEXT:    vpunpcklbw {{.*#+}} ymm0 =
ymm0[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23]
; CHECK-AVX512-NEXT:    vpmullw
 %ymm2, %ymm0, %ymm0
; CHECK-AVX512-NEXT:    vpand %ymm5, %ymm0, %ymm0
; CHECK-AVX512-NEXT:    vpackuswb %ymm3, %ymm0, %ymm0
; CHECK-AVX512-NEXT:    vmovdqa %ymm0, (%rdx)
; CHECK-AVX512-NEXT:    vmovdqa %ymm1, 32(%rdx)
; CHECK-AVX512-NEXT:    vzeroupper
; CHECK-AVX512-NEXT:    retq
;
; CHECK-VBMI-LABEL: mul256:
; CHECK-VBMI:       # %bb.0:
; CHECK-VBMI-NEXT:    vmovdqa (%rdi), %ymm0
; CHECK-VBMI-NEXT:    vmovdqa 32(%rdi), %ymm1
; CHECK-VBMI-NEXT:    vmovdqa (%rsi), %ymm2
; CHECK-VBMI-NEXT:    vmovdqa 32(%rsi), %ymm3
; CHECK-VBMI-NEXT:    vpunpckhbw {{.*#+}} ymm4 =
ymm3[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31]
; CHECK-VBMI-NEXT:    vpunpckhbw {{.*#+}} ymm5 =
ymm1[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31]
; CHECK-VBMI-NEXT:    vpmullw %ymm4, %ymm5, %ymm4
; CHECK-VBMI-NEXT:    vpunpcklbw {{.*#+}} ymm3 =
ymm3[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23]
;
 CHECK-VBMI-NEXT:    vpunpcklbw {{.*#+}} ymm1 =
ymm1[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23]
; CHECK-VBMI-NEXT:    vpmullw %ymm3, %ymm1, %ymm1
; CHECK-VBMI-NEXT:    vmovdqa {{.*#+}} ymm3 =
[0,2,4,6,8,10,12,14,32,34,36,38,40,42,44,46,16,18,20,22,24,26,28,30,48,50,52,54,56,58,60,62]
; CHECK-VBMI-NEXT:    vpermt2b %ymm4, %ymm3, %ymm1
; CHECK-VBMI-NEXT:    vpunpckhbw {{.*#+}} ymm4 =
```

```
ymm2[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31]
; CHECK-VBMI-NEXT:    vpunpckhbw {{.*#+}} ymm5 =
ymm0[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31]
; CHECK-VBMI-NEXT:    vpmullw %ymm4, %ymm5, %ymm4
; CHECK-VBMI-NEXT:    vpunpcklbw {{.*#+}} ymm2 =
ymm2[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23]
; CHECK-VBMI-NEXT:    vpunpcklbw {{.*#+}} ymm0 =
ymm0[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23]
; CHECK-VBMI-NEXT:
   vpmullw %ymm2, %ymm0, %ymm0
; CHECK-VBMI-NEXT:    vpermt2b %ymm4, %ymm3, %ymm0
; CHECK-VBMI-NEXT:    vmovdqa %ymm0, (%rdx)
; CHECK-VBMI-NEXT:    vmovdqa %ymm1, 32(%rdx)
; CHECK-VBMI-NEXT:    vzeroupper
; CHECK-VBMI-NEXT:    retq
 %d = load <64 x i8>, <64 x i8>* %a
 %e = load <64 x i8>, <64 x i8>* %b
 %f = mul <64 x i8> %d, %e
 store <64 x i8> %f, <64 x i8>* %c
 ret void
}

define dso_local void @mul512(<64 x i8>* %a, <64 x i8>* %b, <64 x i8>* %c) "min-legal-vector-width"="512" {
; CHECK-AVX512-LABEL: mul512:
; CHECK-AVX512:       # %bb.0:
; CHECK-AVX512-NEXT:    vmovdqa64 (%rdi), %zmm0
; CHECK-AVX512-NEXT:    vmovdqa64 (%rsi), %zmm1
; CHECK-AVX512-NEXT:    vpunpckhbw {{.*#+}} zmm2 =
zmm1[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31,40,40,41,41,4
2,42,43,43,44,44,45,45,46,46,47,47,56,56,57,57,58,58,59,59,60,60,61,61,62,62,63,63]
; CHECK-AVX512-NEXT:    vpunpckhbw {{.*#+}} zmm3 =
zmm0[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31,40,40,41,41,4
2,42,43,43,44,44,45,45,46,46,47,47,56,56,57,57,58,58,59,59,60,60,61,61,62,62,63,63]
;
 CHECK-AVX512-NEXT:    vpmullw %zmm2, %zmm3, %zmm2
; CHECK-AVX512-NEXT:    vmovdqa64 {{.*#+}} zmm3 =
[255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,2
55,255,255,255,255,255,255]
; CHECK-AVX512-NEXT:    vpandq %zmm3, %zmm2, %zmm2
; CHECK-AVX512-NEXT:    vpunpcklbw {{.*#+}} zmm1 =
zmm1[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23,32,32,33,33,34,34,35,35,36,3
6,37,37,38,38,39,39,48,48,49,49,50,50,51,51,52,52,53,53,54,54,55,55]
; CHECK-AVX512-NEXT:    vpunpcklbw {{.*#+}} zmm0 =
zmm0[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23,32,32,33,33,34,34,35,35,36,3
6,37,37,38,38,39,39,48,48,49,49,50,50,51,51,52,52,53,53,54,54,55,55]
; CHECK-AVX512-NEXT:    vpmullw %zmm1, %zmm0, %zmm0
; CHECK-AVX512-NEXT:    vpandq %zmm3, %zmm0,
```

```
 %zmm0
; CHECK-AVX512-NEXT:    vpackuswb %zmm2, %zmm0, %zmm0
; CHECK-AVX512-NEXT:    vmovdqa64 %zmm0, (%rdx)
; CHECK-AVX512-NEXT:    vzeroupper
; CHECK-AVX512-NEXT:    retq
;
; CHECK-VBMI-LABEL: mul512:
; CHECK-VBMI:       # %bb.0:
; CHECK-VBMI-NEXT:    vmovdqa64 (%rdi), %zmm0
; CHECK-VBMI-NEXT:    vmovdqa64 (%rsi), %zmm1
; CHECK-VBMI-NEXT:    vpunpckhbw {{.*#+}} zmm2 =
zmm1[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31,40,40,41,41,4
2,42,43,43,44,44,45,45,46,46,47,47,56,56,57,57,58,58,59,59,60,60,61,61,62,62,63,63]
; CHECK-VBMI-NEXT:    vpunpckhbw {{.*#+}} zmm3 =
zmm0[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31,40,40,41,41,4
2,42,43,43,44,44,45,45,46,46,47,47,56,56,57,57,58,58,59,59,60,60,61,61,62,62,63,63]
; CHECK-VBMI-NEXT:    vpmullw %zmm2, %zmm3, %zmm2
; CHECK-VBMI-NEXT:    vpunpcklbw {{.*#+}} zmm1 =
zmm1[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23,32,32,33,33,34,34,35,35,36,3
6,37,37,38,38,39,39,48,48,49,49,50,50,51,51,52,52,53,53,54,54,55,55]
;
 CHECK-VBMI-NEXT:    vpunpcklbw {{.*#+}} zmm0 =
zmm0[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23,32,32,33,33,34,34,35,35,36,3
6,37,37,38,38,39,39,48,48,49,49,50,50,51,51,52,52,53,53,54,54,55,55]
; CHECK-VBMI-NEXT:    vpmullw %zmm1, %zmm0, %zmm0
; CHECK-VBMI-NEXT:    vmovdqa64 {{.*#+}} zmm1 =
[0,2,4,6,8,10,12,14,64,66,68,70,72,74,76,78,16,18,20,22,24,26,28,30,80,82,84,86,88,90,92,94,32,34,36,38,40,42,44,
46,96,98,100,102,104,106,108,110,48,50,52,54,56,58,60,62,112,114,116,118,120,122,124,126]
; CHECK-VBMI-NEXT:    vpermi2b %zmm2, %zmm0, %zmm1
; CHECK-VBMI-NEXT:    vmovdqa64 %zmm1, (%rdx)
; CHECK-VBMI-NEXT:    vzeroupper
; CHECK-VBMI-NEXT:    retq
 %d = load <64 x i8>, <64 x i8>* %a
 %e = load <64 x i8>, <64 x i8>* %b
 %f = mul <64 x i8> %d, %e
 store <64 x i8> %f, <64 x i8>* %c
 ret void
}

; This threw an assertion at one point.
define <4 x i32> @mload_v4i32(<4
 x i32> %trigger, <4 x i32>* %addr, <4 x i32> %dst) "min-legal-vector-width"="256" {
; CHECK-LABEL: mload_v4i32:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vptestnmd %xmm0, %xmm0, %k1
; CHECK-NEXT:    vpblendmd (%rdi), %xmm1, %xmm0 {%k1}
; CHECK-NEXT:    retq
 %mask = icmp eq <4 x i32> %trigger, zeroinitializer
```

```
 %res = call <4 x i32> @llvm.masked.load.v4i32.p0v4i32(<4 x i32>* %addr, i32 4, <4 x i1> %mask, <4 x i32>
%dst)
 ret <4 x i32> %res
}
declare <4 x i32> @llvm.masked.load.v4i32.p0v4i32(<4 x i32>*, i32, <4 x i1>, <4 x i32>)

define <16 x i32> @trunc_v16i64_v16i32(<16 x i64>* %x) nounwind "min-legal-vector-width"="256" {
; CHECK-LABEL: trunc_v16i64_v16i32:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vmovdqa (%rdi), %ymm0
; CHECK-NEXT:    vmovdqa 32(%rdi), %ymm1
; CHECK-NEXT:    vmovdqa 64(%rdi), %ymm2
; CHECK-NEXT:    vmovdqa 96(%rdi), %ymm3
; CHECK-NEXT:    vpmovqd %ymm0, %xmm0
; CHECK-NEXT:    vpmovqd %ymm1, %xmm1
; CHECK-NEXT:    vinserti128 $1, %xmm1, %ymm0, %ymm0
; CHECK-NEXT:
   vpmovqd %ymm2, %xmm1
; CHECK-NEXT:    vpmovqd %ymm3, %xmm2
; CHECK-NEXT:    vinserti128 $1, %xmm2, %ymm1, %ymm1
; CHECK-NEXT:    retq
 %a = load <16 x i64>, <16 x i64>* %x
 %b = trunc <16 x i64> %a to <16 x i32>
 ret <16 x i32> %b
}

define <16 x i8> @trunc_v16i64_v16i8(<16 x i64>* %x) nounwind "min-legal-vector-width"="256" {
; CHECK-LABEL: trunc_v16i64_v16i8:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vmovdqa (%rdi), %ymm0
; CHECK-NEXT:    vmovdqa 32(%rdi), %ymm1
; CHECK-NEXT:    vmovdqa 64(%rdi), %ymm2
; CHECK-NEXT:    vmovdqa 96(%rdi), %ymm3
; CHECK-NEXT:    vpmovqb %ymm3, %xmm3
; CHECK-NEXT:    vpmovqb %ymm2, %xmm2
; CHECK-NEXT:    vpunpckldq {{.*#+}} xmm2 = xmm2[0],xmm3[0],xmm2[1],xmm3[1]
; CHECK-NEXT:    vpmovqb %ymm1, %xmm1
; CHECK-NEXT:    vpmovqb %ymm0, %xmm0
; CHECK-NEXT:    vpunpckldq {{.*#+}} xmm0 = xmm0[0],xmm1[0],xmm0[1],xmm1[1]
; CHECK-NEXT:    vpunpcklqdq {{.*#+}} xmm0 = xmm0[0],xmm2[0]
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
 %a = load <16 x i64>, <16
 x i64>* %x
 %b = trunc <16 x i64> %a to <16 x i8>
 ret <16 x i8> %b
}
```

```
define <16 x i8> @trunc_v16i32_v16i8(<16 x i32>* %x) nounwind "min-legal-vector-width"="256" {
; CHECK-LABEL: trunc_v16i32_v16i8:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vmovdqa (%rdi), %ymm0
; CHECK-NEXT:    vmovdqa 32(%rdi), %ymm1
; CHECK-NEXT:    vpmovdb %ymm1, %xmm1
; CHECK-NEXT:    vpmovdb %ymm0, %xmm0
; CHECK-NEXT:    vpunpcklqdq {{.*#+}} xmm0 = xmm0[0],xmm1[0]
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
 %a = load <16 x i32>, <16 x i32>* %x
 %b = trunc <16 x i32> %a to <16 x i8>
 ret <16 x i8> %b
}

define <8 x i8> @trunc_v8i64_v8i8(<8 x i64>* %x) nounwind "min-legal-vector-width"="256" {
; CHECK-LABEL: trunc_v8i64_v8i8:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vmovdqa (%rdi), %ymm0
; CHECK-NEXT:    vmovdqa 32(%rdi), %ymm1
; CHECK-NEXT:    vpmovqb %ymm1, %xmm1
; CHECK-NEXT:    vpmovqb %ymm0, %xmm0
; CHECK-NEXT:    vpunpckldq {{.*#+}} xmm0 = xmm0[0],xmm1[0],xmm0[1],xmm1[1]
; CHECK-NEXT:    vzeroupper
;
 CHECK-NEXT:    retq
 %a = load <8 x i64>, <8 x i64>* %x
 %b = trunc <8 x i64> %a to <8 x i8>
 ret <8 x i8> %b
}

define <8 x i16> @trunc_v8i64_v8i16(<8 x i64>* %x) nounwind "min-legal-vector-width"="256" {
; CHECK-LABEL: trunc_v8i64_v8i16:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vmovdqa (%rdi), %ymm0
; CHECK-NEXT:    vmovdqa 32(%rdi), %ymm1
; CHECK-NEXT:    vpmovqw %ymm1, %xmm1
; CHECK-NEXT:    vpmovqw %ymm0, %xmm0
; CHECK-NEXT:    vpunpcklqdq {{.*#+}} xmm0 = xmm0[0],xmm1[0]
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
 %a = load <8 x i64>, <8 x i64>* %x
 %b = trunc <8 x i64> %a to <8 x i16>
 ret <8 x i16> %b
}

define <8 x i32> @trunc_v8i64_v8i32_zeroes(<8 x i64>* %x) nounwind "min-legal-vector-width"="256" {
; CHECK-LABEL: trunc_v8i64_v8i32_zeroes:
```

```
; CHECK:       # %bb.0:
; CHECK-NEXT:    vpsrlq $48, 32(%rdi), %ymm0
; CHECK-NEXT:    vpsrlq $48, (%rdi), %ymm1
; CHECK-NEXT:    vpackusdw %ymm0, %ymm1, %ymm0
; CHECK-NEXT:    vpermq {{.*#+}} ymm0 = ymm0[0,2,1,3]
; CHECK-NEXT:
   retq
 %a = load <8 x i64>, <8 x i64>* %x
 %b = lshr <8 x i64> %a, <i64 48, i64 48, i64 48, i64 48, i64 48, i64 48, i64 48, i64 48>
 %c = trunc <8 x i64> %b to <8 x i32>
 ret <8 x i32> %c
}

define <16 x i16> @trunc_v16i32_v16i16_zeroes(<16 x i32>* %x) nounwind "min-legal-vector-width"="256" {
; CHECK-LABEL: trunc_v16i32_v16i16_zeroes:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vmovdqa (%rdi), %ymm1
; CHECK-NEXT:    vmovdqa {{.*#+}} ymm0 = [1,3,5,7,9,11,13,15,17,19,21,23,25,27,29,31]
; CHECK-NEXT:    vpermi2w 32(%rdi), %ymm1, %ymm0
; CHECK-NEXT:    retq
 %a = load <16 x i32>, <16 x i32>* %x
 %b = lshr <16 x i32> %a, <i32 16, i32 16, i32 16, i32 16, i32 16, i32 16, i32 16, i32 16, i32 16, i32 16, i32 16, i32
16, i32 16, i32 16, i32 16, i32 16>
 %c = trunc <16 x i32> %b to <16 x i16>
 ret <16 x i16> %c
}

define <32 x i8> @trunc_v32i16_v32i8_zeroes(<32 x i16>* %x) nounwind "min-legal-vector-width"="256" {
; CHECK-AVX512-LABEL: trunc_v32i16_v32i8_zeroes:
; CHECK-AVX512:
  # %bb.0:
; CHECK-AVX512-NEXT:    vpsrlw $8, 32(%rdi), %ymm0
; CHECK-AVX512-NEXT:    vpsrlw $8, (%rdi), %ymm1
; CHECK-AVX512-NEXT:    vpackuswb %ymm0, %ymm1, %ymm0
; CHECK-AVX512-NEXT:    vpermq {{.*#+}} ymm0 = ymm0[0,2,1,3]
; CHECK-AVX512-NEXT:    retq
;
; CHECK-VBMI-LABEL: trunc_v32i16_v32i8_zeroes:
; CHECK-VBMI:       # %bb.0:
; CHECK-VBMI-NEXT:    vmovdqa (%rdi), %ymm1
; CHECK-VBMI-NEXT:    vmovdqa {{.*#+}} ymm0 =
[1,3,5,7,9,11,13,15,17,19,21,23,25,27,29,31,33,35,37,39,41,43,45,47,49,51,53,55,57,59,61,63]
; CHECK-VBMI-NEXT:    vpermi2b 32(%rdi), %ymm1, %ymm0
; CHECK-VBMI-NEXT:    retq
 %a = load <32 x i16>, <32 x i16>* %x
 %b = lshr <32 x i16> %a, <i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8,
i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8>
 %c = trunc <32 x i16> %b to <32 x i8>
```

```
 ret <32 x i8> %c
}

define <8 x i32> @trunc_v8i64_v8i32_sign(<8 x i64>*
 %x) nounwind "min-legal-vector-width"="256" {
; CHECK-LABEL: trunc_v8i64_v8i32_sign:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vpsraq $48, 32(%rdi), %ymm0
; CHECK-NEXT:    vpsraq $48, (%rdi), %ymm1
; CHECK-NEXT:    vpmovqd %ymm1, %xmm1
; CHECK-NEXT:    vpmovqd %ymm0, %xmm0
; CHECK-NEXT:    vinserti128 $1, %xmm0, %ymm1, %ymm0
; CHECK-NEXT:    retq
 %a = load <8 x i64>, <8 x i64>* %x
 %b = ashr <8 x i64> %a, <i64 48, i64 48, i64 48, i64 48, i64 48, i64 48, i64 48, i64 48>
 %c = trunc <8 x i64> %b to <8 x i32>
 ret <8 x i32> %c
}

define <16 x i16> @trunc_v16i32_v16i16_sign(<16 x i32>* %x) nounwind "min-legal-vector-width"="256" {
; CHECK-LABEL: trunc_v16i32_v16i16_sign:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vmovdqa (%rdi), %ymm1
; CHECK-NEXT:    vmovdqa {{.*#+}} ymm0 = [1,3,5,7,9,11,13,15,17,19,21,23,25,27,29,31]
; CHECK-NEXT:    vpermi2w 32(%rdi), %ymm1, %ymm0
; CHECK-NEXT:    retq
 %a = load <16 x i32>, <16 x i32>* %x
 %b = ashr <16 x i32> %a, <i32 16, i32 16, i32 16, i32 16,
 i32 16, i32 16, i32 16, i32 16, i32 16, i32 16, i32 16, i32 16, i32 16, i32 16, i32 16, i32 16>
 %c = trunc <16 x i32> %b to <16 x i16>
 ret <16 x i16> %c
}

define <32 x i8> @trunc_v32i16_v32i8_sign(<32 x i16>* %x) nounwind "min-legal-vector-width"="256" {
; CHECK-AVX512-LABEL: trunc_v32i16_v32i8_sign:
; CHECK-AVX512:       # %bb.0:
; CHECK-AVX512-NEXT:    vpsrlw $8, 32(%rdi), %ymm0
; CHECK-AVX512-NEXT:    vpsrlw $8, (%rdi), %ymm1
; CHECK-AVX512-NEXT:    vpackuswb %ymm0, %ymm1, %ymm0
; CHECK-AVX512-NEXT:    vpermq {{.*#+}} ymm0 = ymm0[0,2,1,3]
; CHECK-AVX512-NEXT:    retq
;
; CHECK-VBMI-LABEL: trunc_v32i16_v32i8_sign:
; CHECK-VBMI:       # %bb.0:
; CHECK-VBMI-NEXT:    vmovdqa (%rdi), %ymm1
; CHECK-VBMI-NEXT:    vmovdqa {{.*#+}} ymm0 =
[1,3,5,7,9,11,13,15,17,19,21,23,25,27,29,31,33,35,37,39,41,43,45,47,49,51,53,55,57,59,61,63]
; CHECK-VBMI-NEXT:    vpermi2b 32(%rdi), %ymm1, %ymm0
```

```
; CHECK-VBMI-NEXT:    retq
 %a = load <32 x i16>, <32 x i16>* %x
 %b = ashr <32 x i16> %a, <i16 8, i16
 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8,
i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8, i16 8>
 %c = trunc <32 x i16> %b to <32 x i8>
 ret <32 x i8> %c
}

define dso_local void @zext_v16i8_v16i64(<16 x i8> %x, <16 x i64>* %y) nounwind "min-legal-vector-
width"="256" {
; CHECK-LABEL: zext_v16i8_v16i64:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vpmovzxbw {{.*#+}} ymm1 =
xmm0[0],zero,xmm0[1],zero,xmm0[2],zero,xmm0[3],zero,xmm0[4],zero,xmm0[5],zero,xmm0[6],zero,xmm0[7],zer
o,xmm0[8],zero,xmm0[9],zero,xmm0[10],zero,xmm0[11],zero,xmm0[12],zero,xmm0[13],zero,xmm0[14],zero,xm
m0[15],zero
; CHECK-NEXT:    vpshufd {{.*#+}} xmm2 = xmm1[2,3,2,3]
; CHECK-NEXT:    vpmovzxwq {{.*#+}} ymm2 =
xmm2[0],zero,zero,zero,xmm2[1],zero,zero,zero,xmm2[2],zero,zero,zero,xmm2[3],zero,zero,zero
; CHECK-NEXT:    vextracti128 $1, %ymm1, %xmm1
; CHECK-NEXT:    vpshufd {{.*#+}} xmm3 = xmm1[2,3,2,3]
; CHECK-NEXT:
   vpmovzxwq {{.*#+}} ymm3 =
xmm3[0],zero,zero,zero,xmm3[1],zero,zero,zero,xmm3[2],zero,zero,zero,xmm3[3],zero,zero,zero
; CHECK-NEXT:    vpmovzxwq {{.*#+}} ymm1 =
xmm1[0],zero,zero,zero,xmm1[1],zero,zero,zero,xmm1[2],zero,zero,zero,xmm1[3],zero,zero,zero
; CHECK-NEXT:    vpmovzxbq {{.*#+}} ymm0 =
xmm0[0],zero,zero,zero,zero,zero,zero,zero,zero,xmm0[1],zero,zero,zero,zero,zero,zero,zero,xmm0[2],zero,zero,zero,zer
o,zero,zero,zero,xmm0[3],zero,zero,zero,zero,zero,zero,zero
; CHECK-NEXT:    vmovdqa %ymm0, (%rdi)
; CHECK-NEXT:    vmovdqa %ymm1, 64(%rdi)
; CHECK-NEXT:    vmovdqa %ymm3, 96(%rdi)
; CHECK-NEXT:    vmovdqa %ymm2, 32(%rdi)
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
 %a = zext <16 x i8> %x to <16 x i64>
 store <16 x i64> %a, <16 x i64>* %y
 ret void
}

define dso_local void @sext_v16i8_v16i64(<16 x i8> %x, <16 x i64>* %y) nounwind "min-legal-vector-
width"="256" {
; CHECK-LABEL: sext_v16i8_v16i64:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vpmovsxbw %xmm0, %ymm1
; CHECK-NEXT:
```

```
    vpshufd {{.*#+}} xmm2 = xmm1[2,3,2,3]
; CHECK-NEXT:    vpmovsxwq %xmm2, %ymm2
; CHECK-NEXT:    vextracti128 $1, %ymm1, %xmm1
; CHECK-NEXT:    vpshufd {{.*#+}} xmm3 = xmm1[2,3,2,3]
; CHECK-NEXT:    vpmovsxwq %xmm3, %ymm3
; CHECK-NEXT:    vpmovsxwq %xmm1, %ymm1
; CHECK-NEXT:    vpmovsxbq %xmm0, %ymm0
; CHECK-NEXT:    vmovdqa %ymm0, (%rdi)
; CHECK-NEXT:    vmovdqa %ymm1, 64(%rdi)
; CHECK-NEXT:    vmovdqa %ymm3, 96(%rdi)
; CHECK-NEXT:    vmovdqa %ymm2, 32(%rdi)
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
 %a = sext <16 x i8> %x to <16 x i64>
 store <16 x i64> %a, <16 x i64>* %y
 ret void
}

define dso_local void @vselect_split_v8i16_setcc(<8 x i16> %s, <8 x i16> %t, <8 x i64>* %p, <8 x i64>* %q, <8
x i64>* %r) "min-legal-vector-width"="256" {
; CHECK-LABEL: vselect_split_v8i16_setcc:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vmovdqa (%rsi), %ymm2
; CHECK-NEXT:    vmovdqa 32(%rsi), %ymm3
; CHECK-NEXT:    vpcmpeqw %xmm1, %xmm0, %k1
; CHECK-NEXT:    kshiftrb $4, %k1, %k2
;
 CHECK-NEXT:    vmovdqa64 32(%rdi), %ymm3 {%k2}
; CHECK-NEXT:    vmovdqa64 (%rdi), %ymm2 {%k1}
; CHECK-NEXT:    vmovdqa %ymm2, (%rdx)
; CHECK-NEXT:    vmovdqa %ymm3, 32(%rdx)
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
 %x = load <8 x i64>, <8 x i64>* %p
 %y = load <8 x i64>, <8 x i64>* %q
 %a = icmp eq <8 x i16> %s, %t
 %b = select <8 x i1> %a, <8 x i64> %x, <8 x i64> %y
 store <8 x i64> %b, <8 x i64>* %r
 ret void
}

define dso_local void @vselect_split_v8i32_setcc(<8 x i32> %s, <8 x i32> %t, <8 x i64>* %p, <8 x i64>* %q, <8
x i64>* %r) "min-legal-vector-width"="256" {
; CHECK-LABEL: vselect_split_v8i32_setcc:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vmovdqa (%rsi), %ymm2
; CHECK-NEXT:    vmovdqa 32(%rsi), %ymm3
; CHECK-NEXT:    vpcmpeqd %ymm1, %ymm0, %k1
```

```
; CHECK-NEXT:    kshiftrb $4, %k1, %k2
; CHECK-NEXT:    vmovdqa64 32(%rdi), %ymm3 {%k2}
; CHECK-NEXT:    vmovdqa64 (%rdi), %ymm2 {%k1}
; CHECK-NEXT:    vmovdqa %ymm2, (%rdx)
; CHECK-NEXT:    vmovdqa %ymm3, 32(%rdx)
; CHECK-NEXT:
   vzeroupper
; CHECK-NEXT:    retq
 %x = load <8 x i64>, <8 x i64>* %p
 %y = load <8 x i64>, <8 x i64>* %q
 %a = icmp eq <8 x i32> %s, %t
 %b = select <8 x i1> %a, <8 x i64> %x, <8 x i64> %y
 store <8 x i64> %b, <8 x i64>* %r
 ret void
}

define dso_local void @vselect_split_v16i8_setcc(<16 x i8> %s, <16 x i8> %t, <16 x i32>* %p, <16 x i32>* %q,
<16 x i32>* %r) "min-legal-vector-width"="256" {
; CHECK-LABEL: vselect_split_v16i8_setcc:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vmovdqa (%rsi), %ymm2
; CHECK-NEXT:    vmovdqa 32(%rsi), %ymm3
; CHECK-NEXT:    vpcmpeqb %xmm1, %xmm0, %k1
; CHECK-NEXT:    kshiftrw $8, %k1, %k2
; CHECK-NEXT:    vmovdqa32 32(%rdi), %ymm3 {%k2}
; CHECK-NEXT:    vmovdqa32 (%rdi), %ymm2 {%k1}
; CHECK-NEXT:    vmovdqa %ymm2, (%rdx)
; CHECK-NEXT:    vmovdqa %ymm3, 32(%rdx)
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
 %x = load <16 x i32>, <16 x i32>* %p
 %y = load <16 x i32>, <16 x i32>* %q
 %a = icmp eq <16 x i8> %s, %t
 %b = select <16 x i1>
 %a, <16 x i32> %x, <16 x i32> %y
 store <16 x i32> %b, <16 x i32>* %r
 ret void
}

define dso_local void @vselect_split_v16i16_setcc(<16 x i16> %s, <16 x i16> %t, <16 x i32>* %p, <16 x i32>*
%q, <16 x i32>* %r) "min-legal-vector-width"="256" {
; CHECK-LABEL: vselect_split_v16i16_setcc:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vmovdqa (%rsi), %ymm2
; CHECK-NEXT:    vmovdqa 32(%rsi), %ymm3
; CHECK-NEXT:    vpcmpeqw %ymm1, %ymm0, %k1
; CHECK-NEXT:    kshiftrw $8, %k1, %k2
; CHECK-NEXT:    vmovdqa32 32(%rdi), %ymm3 {%k2}
```

```
; CHECK-NEXT:    vmovdqa32 (%rdi), %ymm2 {%k1}
; CHECK-NEXT:    vmovdqa %ymm2, (%rdx)
; CHECK-NEXT:    vmovdqa %ymm3, 32(%rdx)
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
 %x = load <16 x i32>, <16 x i32>* %p
 %y = load <16 x i32>, <16 x i32>* %q
 %a = icmp eq <16 x i16> %s, %t
 %b = select <16 x i1> %a, <16 x i32> %x, <16 x i32> %y
 store <16 x i32> %b, <16 x i32>* %r
 ret void
}

define <16 x i8> @trunc_packus_v16i32_v16i8(<16 x i32>* %p) "min-legal-vector-width"="256"
 {
; CHECK-LABEL: trunc_packus_v16i32_v16i8:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vmovdqa (%rdi), %ymm0
; CHECK-NEXT:    vpackusdw 32(%rdi), %ymm0, %ymm0
; CHECK-NEXT:    vpermq {{.*#+}} ymm0 = ymm0[0,2,1,3]
; CHECK-NEXT:    vpmovuswb %ymm0, %xmm0
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
 %a = load <16 x i32>, <16 x i32>* %p
 %b = icmp slt <16 x i32> %a, <i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32
255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255>
 %c = select <16 x i1> %b, <16 x i32> %a, <16 x i32> <i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32
255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255>
 %d = icmp sgt <16 x i32> %c, zeroinitializer
 %e = select <16 x i1> %d, <16 x i32> %c, <16 x i32> zeroinitializer
 %f = trunc <16 x i32> %e to <16 x i8>
 ret <16 x i8> %f
}

define dso_local void @trunc_packus_v16i32_v16i8_store(<16 x i32>* %p, <16 x i8>* %q) "min-legal-vector-
width"="256"
 {
; CHECK-LABEL: trunc_packus_v16i32_v16i8_store:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vmovdqa (%rdi), %ymm0
; CHECK-NEXT:    vpackusdw 32(%rdi), %ymm0, %ymm0
; CHECK-NEXT:    vpermq {{.*#+}} ymm0 = ymm0[0,2,1,3]
; CHECK-NEXT:    vpmovuswb %ymm0, (%rsi)
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
 %a = load <16 x i32>, <16 x i32>* %p
 %b = icmp slt <16 x i32> %a, <i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32
255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255>
```

```
 %c = select <16 x i1> %b, <16 x i32> %a, <16 x i32> <i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32
255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255, i32 255>
 %d = icmp sgt <16 x i32> %c, zeroinitializer
 %e = select <16 x i1> %d, <16 x i32> %c, <16 x i32> zeroinitializer
 %f = trunc <16 x i32> %e to <16 x i8>
 store <16 x i8> %f, <16 x i8>* %q
 ret void
}

define <64 x i1> @v64i1_argument_return(<64 x
 i1> %x) "min-legal-vector-width"="256" {
; CHECK-LABEL: v64i1_argument_return:
; CHECK:       # %bb.0:
; CHECK-NEXT:    retq
 ret <64 x i1> %x
}

define dso_local void @v64i1_shuffle(<64 x i8>* %x, <64 x i8>* %y) "min-legal-vector-width"="256" {
; CHECK-LABEL: v64i1_shuffle:
; CHECK:       # %bb.0: # %entry
; CHECK-NEXT:    vmovdqa (%rdi), %ymm1
; CHECK-NEXT:    vmovdqa 32(%rdi), %ymm0
; CHECK-NEXT:    vptestnmb %ymm1, %ymm1, %k0
; CHECK-NEXT:    kshiftrd $1, %k0, %k1
; CHECK-NEXT:    movq $-3, %rax
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k1, %k1
; CHECK-NEXT:    kshiftlq $63, %k0, %k2
; CHECK-NEXT:    kshiftrq $62, %k2, %k2
; CHECK-NEXT:    korq %k2, %k1, %k1
; CHECK-NEXT:    movq $-5, %rax
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k1, %k1
; CHECK-NEXT:    kshiftrd $3, %k0, %k2
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $61, %k2, %k2
; CHECK-NEXT:    korq %k2, %k1, %k1
; CHECK-NEXT:    movq $-9, %rax
; CHECK-NEXT:    kmovq
 %rax, %k2
; CHECK-NEXT:    kandq %k2, %k1, %k1
; CHECK-NEXT:    kshiftrd $2, %k0, %k2
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $60, %k2, %k2
; CHECK-NEXT:    korq %k2, %k1, %k1
; CHECK-NEXT:    movq $-17, %rax
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k1, %k1
```

```
; CHECK-NEXT:    kshiftrd $5, %k0, %k2
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $59, %k2, %k2
; CHECK-NEXT:    korq %k2, %k1, %k1
; CHECK-NEXT:    movq $-33, %rax
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k1, %k1
; CHECK-NEXT:    kshiftrd $4, %k0, %k2
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $58, %k2, %k2
; CHECK-NEXT:    korq %k2, %k1, %k1
; CHECK-NEXT:    movq $-65, %rax
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k1, %k1
; CHECK-NEXT:    kshiftrd $7, %k0, %k2
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $57, %k2, %k2
; CHECK-NEXT:    korq %k2, %k1, %k1
; CHECK-NEXT:    movq
 $-129, %rax
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k1, %k1
; CHECK-NEXT:    kshiftrd $6, %k0, %k2
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $56, %k2, %k2
; CHECK-NEXT:    korq %k2, %k1, %k1
; CHECK-NEXT:    movq $-257, %rax # imm = 0xFEFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k1, %k1
; CHECK-NEXT:    kshiftrd $9, %k0, %k2
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $55, %k2, %k2
; CHECK-NEXT:    korq %k2, %k1, %k1
; CHECK-NEXT:    movq $-513, %rax # imm = 0xFDFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k1, %k1
; CHECK-NEXT:    kshiftrd $8, %k0, %k2
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $54, %k2, %k2
; CHECK-NEXT:    korq %k2, %k1, %k1
; CHECK-NEXT:    movq $-1025, %rax # imm = 0xFBFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k1, %k1
; CHECK-NEXT:    kshiftrd $11, %k0, %k2
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:
   kshiftrq $53, %k2, %k2
; CHECK-NEXT:    korq %k2, %k1, %k1
```

```
; CHECK-NEXT:    movq $-2049, %rax # imm = 0xF7FF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k1, %k1
; CHECK-NEXT:    kshiftrd $10, %k0, %k2
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $52, %k2, %k2
; CHECK-NEXT:    korq %k2, %k1, %k1
; CHECK-NEXT:    movq $-4097, %rax # imm = 0xEFFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k1, %k1
; CHECK-NEXT:    kshiftrd $13, %k0, %k2
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $51, %k2, %k2
; CHECK-NEXT:    korq %k2, %k1, %k1
; CHECK-NEXT:    movq $-8193, %rax # imm = 0xDFFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k1, %k1
; CHECK-NEXT:    kshiftrd $12, %k0, %k2
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $50, %k2, %k2
; CHECK-NEXT:    korq %k2, %k1, %k1
; CHECK-NEXT:    movq $-16385, %rax # imm = 0xBFFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2,
 %k1, %k1
; CHECK-NEXT:    kshiftrd $15, %k0, %k2
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $49, %k2, %k2
; CHECK-NEXT:    korq %k2, %k1, %k1
; CHECK-NEXT:    movq $-32769, %rax # imm = 0xFFFF7FFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k1, %k1
; CHECK-NEXT:    kshiftrd $14, %k0, %k2
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $48, %k2, %k2
; CHECK-NEXT:    korq %k2, %k1, %k1
; CHECK-NEXT:    movq $-65537, %rax # imm = 0xFFFEFFFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k1, %k1
; CHECK-NEXT:    kshiftrd $17, %k0, %k2
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $47, %k2, %k2
; CHECK-NEXT:    korq %k2, %k1, %k1
; CHECK-NEXT:    movq $-131073, %rax # imm = 0xFFFDFFFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k1, %k1
; CHECK-NEXT:    kshiftrd $16, %k0, %k2
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
```

```
; CHECK-NEXT:    kshiftrq $46, %k2, %k2
; CHECK-NEXT:    korq %k2,
 %k1, %k1
; CHECK-NEXT:    movq $-262145, %rax # imm = 0xFFFBFFFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k1, %k1
; CHECK-NEXT:    kshiftrd $19, %k0, %k2
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $45, %k2, %k2
; CHECK-NEXT:    korq %k2, %k1, %k1
; CHECK-NEXT:    movq $-524289, %rax # imm = 0xFFF7FFFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k1, %k1
; CHECK-NEXT:    kshiftrd $18, %k0, %k2
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $44, %k2, %k2
; CHECK-NEXT:    korq %k2, %k1, %k1
; CHECK-NEXT:    movq $-1048577, %rax # imm = 0xFFEFFFFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k1, %k1
; CHECK-NEXT:    kshiftrd $21, %k0, %k2
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $43, %k2, %k2
; CHECK-NEXT:    korq %k2, %k1, %k1
; CHECK-NEXT:    movq $-2097153, %rax # imm = 0xFFDFFFFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k1, %k1
; CHECK-NEXT:
   kshiftrd $20, %k0, %k2
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $42, %k2, %k2
; CHECK-NEXT:    korq %k2, %k1, %k1
; CHECK-NEXT:    movq $-4194305, %rax # imm = 0xFFBFFFFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k1, %k1
; CHECK-NEXT:    kshiftrd $23, %k0, %k2
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $41, %k2, %k2
; CHECK-NEXT:    korq %k2, %k1, %k1
; CHECK-NEXT:    movq $-8388609, %rax # imm = 0xFF7FFFFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k1, %k1
; CHECK-NEXT:    kshiftrd $22, %k0, %k2
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $40, %k2, %k2
; CHECK-NEXT:    korq %k2, %k1, %k1
; CHECK-NEXT:    movq $-16777217, %rax # imm = 0xFEFFFFFF
; CHECK-NEXT:    kmovq %rax, %k2
```

```
; CHECK-NEXT:    kandq %k2, %k1, %k1
; CHECK-NEXT:    kshiftrd $25, %k0, %k2
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $39, %k2, %k2
; CHECK-NEXT:    korq %k2, %k1, %k1
; CHECK-NEXT:
   movq $-33554433, %rax # imm = 0xFDFFFFFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k1, %k1
; CHECK-NEXT:    kshiftrd $24, %k0, %k2
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $38, %k2, %k2
; CHECK-NEXT:    korq %k2, %k1, %k1
; CHECK-NEXT:    movq $-67108865, %rax # imm = 0xFBFFFFFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k1, %k1
; CHECK-NEXT:    kshiftrd $27, %k0, %k2
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $37, %k2, %k2
; CHECK-NEXT:    korq %k2, %k1, %k1
; CHECK-NEXT:    movq $-134217729, %rax # imm = 0xF7FFFFFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k1, %k1
; CHECK-NEXT:    kshiftrd $26, %k0, %k2
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $36, %k2, %k2
; CHECK-NEXT:    korq %k2, %k1, %k1
; CHECK-NEXT:    movq $-268435457, %rax # imm = 0xEFFFFFFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k1, %k1
; CHECK-NEXT:    kshiftrd
 $29, %k0, %k2
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $35, %k2, %k2
; CHECK-NEXT:    korq %k2, %k1, %k1
; CHECK-NEXT:    movq $-536870913, %rax # imm = 0xDFFFFFFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k1, %k1
; CHECK-NEXT:    kshiftrd $28, %k0, %k2
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $34, %k2, %k2
; CHECK-NEXT:    korq %k2, %k1, %k1
; CHECK-NEXT:    movq $-1073741825, %rax # imm = 0xBFFFFFFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k1, %k1
; CHECK-NEXT:    kshiftrd $31, %k0, %k2
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $33, %k2, %k2
```

```
; CHECK-NEXT:    korq %k2, %k1, %k1
; CHECK-NEXT:    movabsq $-2147483649, %rax # imm = 0xFFFFFFFF7FFFFFFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k1, %k2
; CHECK-NEXT:    vptestnmb %ymm0, %ymm0, %k1
; CHECK-NEXT:    kshiftrd $30, %k0, %k0
; CHECK-NEXT:    kshiftlq $63, %k0, %k0
; CHECK-NEXT:    kshiftrq $32, %k0,
 %k0
; CHECK-NEXT:    korq %k0, %k2, %k0
; CHECK-NEXT:    movabsq $-4294967297, %rax # imm = 0xFFFFFFFEFFFFFFFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k0, %k0
; CHECK-NEXT:    kshiftrd $1, %k1, %k2
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $31, %k2, %k2
; CHECK-NEXT:    korq %k2, %k0, %k0
; CHECK-NEXT:    movabsq $-8589934593, %rax # imm = 0xFFFFFFFDFFFFFFFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k0, %k0
; CHECK-NEXT:    kshiftlq $63, %k1, %k2
; CHECK-NEXT:    kshiftrq $30, %k2, %k2
; CHECK-NEXT:    korq %k2, %k0, %k0
; CHECK-NEXT:    movabsq $-17179869185, %rax # imm = 0xFFFFFFFBFFFFFFFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k0, %k0
; CHECK-NEXT:    kshiftrd $3, %k1, %k2
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $29, %k2, %k2
; CHECK-NEXT:    korq %k2, %k0, %k0
; CHECK-NEXT:    movabsq $-34359738369, %rax # imm = 0xFFFFFFF7FFFFFFFF
; CHECK-NEXT:    kmovq %rax, %k2
;
 CHECK-NEXT:    kandq %k2, %k0, %k0
; CHECK-NEXT:    kshiftrd $2, %k1, %k2
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $28, %k2, %k2
; CHECK-NEXT:    korq %k2, %k0, %k0
; CHECK-NEXT:    movabsq $-68719476737, %rax # imm = 0xFFFFFFEFFFFFFFFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k0, %k0
; CHECK-NEXT:    kshiftrd $5, %k1, %k2
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $27, %k2, %k2
; CHECK-NEXT:    korq %k2, %k0, %k0
; CHECK-NEXT:    movabsq $-137438953473, %rax # imm = 0xFFFFFFDFFFFFFFFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k0, %k0
```

```
; CHECK-NEXT:    kshiftrd $4, %k1, %k2
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $26, %k2, %k2
; CHECK-NEXT:    korq %k2, %k0, %k0
; CHECK-NEXT:    movabsq $-274877906945, %rax # imm = 0xFFFFFFBFFFFFFFFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k0, %k0
; CHECK-NEXT:    kshiftrd $7, %k1, %k2
; CHECK-NEXT:    kshiftlq $63, %k2,
 %k2
; CHECK-NEXT:    kshiftrq $25, %k2, %k2
; CHECK-NEXT:    korq %k2, %k0, %k0
; CHECK-NEXT:    movabsq $-549755813889, %rax # imm = 0xFFFFFF7FFFFFFFFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k0, %k0
; CHECK-NEXT:    kshiftrd $6, %k1, %k2
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $24, %k2, %k2
; CHECK-NEXT:    korq %k2, %k0, %k0
; CHECK-NEXT:    movabsq $-1099511627777, %rax # imm = 0xFFFFFEFFFFFFFFFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k0, %k0
; CHECK-NEXT:    kshiftrd $9, %k1, %k2
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $23, %k2, %k2
; CHECK-NEXT:    korq %k2, %k0, %k0
; CHECK-NEXT:    movabsq $-2199023255553, %rax # imm = 0xFFFFFDFFFFFFFFFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k0, %k0
; CHECK-NEXT:    kshiftrd $8, %k1, %k2
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $22, %k2, %k2
; CHECK-NEXT:    korq %k2, %k0, %k0
; CHECK-NEXT:    movabsq
 $-4398046511105, %rax # imm = 0xFFFFFBFFFFFFFFFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k0, %k0
; CHECK-NEXT:    kshiftrd $11, %k1, %k2
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $21, %k2, %k2
; CHECK-NEXT:    korq %k2, %k0, %k0
; CHECK-NEXT:    movabsq $-8796093022209, %rax # imm = 0xFFFFF7FFFFFFFFFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k0, %k0
; CHECK-NEXT:    kshiftrd $10, %k1, %k2
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $20, %k2, %k2
; CHECK-NEXT:    korq %k2, %k0, %k0
```

```
; CHECK-NEXT:    movabsq $-17592186044417, %rax # imm = 0xFFFFEFFFFFFFFFFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k0, %k0
; CHECK-NEXT:    kshiftrd $13, %k1, %k2
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $19, %k2, %k2
; CHECK-NEXT:    korq %k2, %k0, %k0
; CHECK-NEXT:    movabsq $-35184372088833, %rax # imm = 0xFFFFDFFFFFFFFFFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:
   kandq %k2, %k0, %k0
; CHECK-NEXT:    kshiftrd $12, %k1, %k2
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $18, %k2, %k2
; CHECK-NEXT:    korq %k2, %k0, %k0
; CHECK-NEXT:    movabsq $-70368744177665, %rax # imm = 0xFFFFBFFFFFFFFFFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k0, %k0
; CHECK-NEXT:    kshiftrd $15, %k1, %k2
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $17, %k2, %k2
; CHECK-NEXT:    korq %k2, %k0, %k0
; CHECK-NEXT:    movabsq $-140737488355329, %rax # imm = 0xFFFF7FFFFFFFFFFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k0, %k0
; CHECK-NEXT:    kshiftrd $14, %k1, %k2
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $16, %k2, %k2
; CHECK-NEXT:    korq %k2, %k0, %k0
; CHECK-NEXT:    movabsq $-281474976710657, %rax # imm = 0xFFFEFFFFFFFFFFFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k0, %k0
; CHECK-NEXT:    kshiftrd $17, %k1, %k2
; CHECK-NEXT:    kshiftlq $63,
 %k2, %k2
; CHECK-NEXT:    kshiftrq $15, %k2, %k2
; CHECK-NEXT:    korq %k2, %k0, %k0
; CHECK-NEXT:    movabsq $-562949953421313, %rax # imm = 0xFFFDFFFFFFFFFFFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k0, %k0
; CHECK-NEXT:    kshiftrd $16, %k1, %k2
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $14, %k2, %k2
; CHECK-NEXT:    korq %k2, %k0, %k0
; CHECK-NEXT:    movabsq $-1125899906842625, %rax # imm = 0xFFFBFFFFFFFFFFFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k0, %k0
; CHECK-NEXT:    kshiftrd $19, %k1, %k2
```

```
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $13, %k2, %k2
; CHECK-NEXT:    korq %k2, %k0, %k0
; CHECK-NEXT:    movabsq $-2251799813685249, %rax # imm = 0xFFF7FFFFFFFFFFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k0, %k0
; CHECK-NEXT:    kshiftrd $18, %k1, %k2
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $12, %k2, %k2
; CHECK-NEXT:    korq %k2, %k0, %k0
; CHECK-NEXT:
   movabsq $-4503599627370497, %rax # imm = 0xFFEFFFFFFFFFFFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k0, %k0
; CHECK-NEXT:    kshiftrd $21, %k1, %k2
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $11, %k2, %k2
; CHECK-NEXT:    korq %k2, %k0, %k0
; CHECK-NEXT:    movabsq $-9007199254740993, %rax # imm = 0xFFDFFFFFFFFFFFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k0, %k0
; CHECK-NEXT:    kshiftrd $20, %k1, %k2
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $10, %k2, %k2
; CHECK-NEXT:    korq %k2, %k0, %k0
; CHECK-NEXT:    movabsq $-18014398509481985, %rax # imm = 0xFFBFFFFFFFFFFFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k0, %k0
; CHECK-NEXT:    kshiftrd $23, %k1, %k2
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $9, %k2, %k2
; CHECK-NEXT:    korq %k2, %k0, %k0
; CHECK-NEXT:    movabsq $-36028797018963969, %rax # imm = 0xFF7FFFFFFFFFFFF
; CHECK-NEXT:    kmovq
 %rax, %k2
; CHECK-NEXT:    kandq %k2, %k0, %k0
; CHECK-NEXT:    kshiftrd $22, %k1, %k2
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $8, %k2, %k2
; CHECK-NEXT:    korq %k2, %k0, %k0
; CHECK-NEXT:    movabsq $-72057594037927937, %rax # imm = 0xFEFFFFFFFFFFFFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k0, %k0
; CHECK-NEXT:    kshiftrd $25, %k1, %k2
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $7, %k2, %k2
; CHECK-NEXT:    korq %k2, %k0, %k0
; CHECK-NEXT:    movabsq $-144115188075855873, %rax # imm = 0xFDFFFFFFFFFFFFF
```

```
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k0, %k0
; CHECK-NEXT:    kshiftrd $24, %k1, %k2
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $6, %k2, %k2
; CHECK-NEXT:    korq %k2, %k0, %k0
; CHECK-NEXT:    movabsq $-288230376151711745, %rax # imm = 0xFBFFFFFFFFFFFFFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k0, %k0
; CHECK-NEXT:    kshiftrd $27, %k1, %k2
;
 CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $5, %k2, %k2
; CHECK-NEXT:    korq %k2, %k0, %k0
; CHECK-NEXT:    movabsq $-576460752303423489, %rax # imm = 0xF7FFFFFFFFFFFFFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k0, %k0
; CHECK-NEXT:    kshiftrd $26, %k1, %k2
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $4, %k2, %k2
; CHECK-NEXT:    korq %k2, %k0, %k0
; CHECK-NEXT:    movabsq $-1152921504606846977, %rax # imm = 0xEFFFFFFFFFFFFFFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k0, %k0
; CHECK-NEXT:    kshiftrd $29, %k1, %k2
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $3, %k2, %k2
; CHECK-NEXT:    korq %k2, %k0, %k0
; CHECK-NEXT:    movabsq $-2305843009213693953, %rax # imm = 0xDFFFFFFFFFFFFFFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k0, %k0
; CHECK-NEXT:    kshiftrd $28, %k1, %k2
; CHECK-NEXT:    kshiftlq $63, %k2, %k2
; CHECK-NEXT:    kshiftrq $2, %k2, %k2
; CHECK-NEXT:
   korq %k2, %k0, %k0
; CHECK-NEXT:    movabsq $-4611686018427387905, %rax # imm = 0xBFFFFFFFFFFFFFFF
; CHECK-NEXT:    kmovq %rax, %k2
; CHECK-NEXT:    kandq %k2, %k0, %k0
; CHECK-NEXT:    kshiftrd $31, %k1, %k2
; CHECK-NEXT:    kshiftlq $62, %k2, %k2
; CHECK-NEXT:    korq %k2, %k0, %k0
; CHECK-NEXT:    kshiftrd $30, %k1, %k1
; CHECK-NEXT:    kshiftlq $1, %k0, %k0
; CHECK-NEXT:    kshiftrq $1, %k0, %k0
; CHECK-NEXT:    kshiftlq $63, %k1, %k1
; CHECK-NEXT:    korq %k1, %k0, %k1
; CHECK-NEXT:    vmovdqu8 %ymm1, (%rsi) {%k1}
```

```
; CHECK-NEXT:    kshiftrq $32, %k1, %k1
; CHECK-NEXT:    vmovdqu8 %ymm0, 32(%rsi) {%k1}
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
entry:
 %a = load <64 x i8>, <64 x i8>* %x
 %b = icmp eq <64 x i8> %a, zeroinitializer
 %shuf = shufflevector <64 x i1> %b, <64 x i1> undef, <64 x i32> <i32 1, i32 0, i32 3, i32 2, i32 5, i32 4, i32 7, i32
6, i32 9, i32 8, i32 11, i32 10, i32 13, i32 12, i32 15, i32 14, i32 17, i32 16, i32 19, i32 18, i32 21, i32 20, i32
 23, i32 22, i32 25, i32 24, i32 27, i32 26, i32 29, i32 28, i32 31, i32 30, i32 33, i32 32, i32 35, i32 34, i32 37, i32
36, i32 39, i32 38, i32 41, i32 40, i32 43, i32 42, i32 45, i32 44, i32 47, i32 46, i32 49, i32 48, i32 51, i32 50, i32 53,
i32 52, i32 55, i32 54, i32 57, i32 56, i32 59, i32 58, i32 61, i32 60, i32 63, i32 62>
 call void @llvm.masked.store.v64i8.p0v64i8(<64 x i8> %a, <64 x i8>* %y, i32 1, <64 x i1> %shuf)
 ret void
}
declare void @llvm.masked.store.v64i8.p0v64i8(<64 x i8>, <64 x i8>*, i32, <64 x i1>)

@mem64_dst = dso_local global i64 0, align 8
@mem64_src = dso_local global i64 0, align 8
define dso_local i32 @v64i1_inline_asm() "min-legal-vector-width"="256" {
; CHECK-LABEL: v64i1_inline_asm:
; CHECK:       # %bb.0:
; CHECK-NEXT:    kmovq mem64_src(%rip), %k0
; CHECK-NEXT:    #APP
; CHECK-NEXT:    #NO_APP
; CHECK-NEXT:    kmovq %k0, mem64_dst(%rip)
; CHECK-NEXT:    movl -{{[0-9]+}}(%rsp), %eax
; CHECK-NEXT:    retq
 %1 = alloca i32, align 4
 %2 = load i64, i64*
 @mem64_src, align 8
 %3 = call i64 asm "", "=k,k,~{dirflag},~{fpsr},~{flags}"(i64 %2)
 store i64 %3, i64* @mem64_dst, align 8
 %4 = load i32, i32* %1, align 4
 ret i32 %4
}

define dso_local void @cmp_v8i64_sext(<8 x i64>* %xptr, <8 x i64>* %yptr, <8 x i64>* %zptr) "min-legal-
vector-width"="256" {
; CHECK-LABEL: cmp_v8i64_sext:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vmovdqa (%rsi), %ymm0
; CHECK-NEXT:    vmovdqa 32(%rsi), %ymm1
; CHECK-NEXT:    vpcmpgtq 32(%rdi), %ymm1, %ymm1
; CHECK-NEXT:    vpcmpgtq (%rdi), %ymm0, %ymm0
; CHECK-NEXT:    vmovdqa %ymm0, (%rdx)
; CHECK-NEXT:    vmovdqa %ymm1, 32(%rdx)
; CHECK-NEXT:    vzeroupper
```

```
; CHECK-NEXT:    retq
 %x = load <8 x i64>, <8 x i64>* %xptr
 %y = load <8 x i64>, <8 x i64>* %yptr
 %cmp = icmp slt <8 x i64> %x, %y
 %ext = sext <8 x i1> %cmp to <8 x i64>
 store <8 x i64> %ext, <8 x i64>* %zptr
 ret void
}

define dso_local void @cmp_v8i64_zext(<8 x i64>* %xptr, <8 x i64>* %yptr, <8 x i64>* %zptr) "min-legal-vector-width"="256"
 {
; CHECK-LABEL: cmp_v8i64_zext:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vmovdqa (%rsi), %ymm0
; CHECK-NEXT:    vmovdqa 32(%rsi), %ymm1
; CHECK-NEXT:    vpcmpgtq 32(%rdi), %ymm1, %ymm1
; CHECK-NEXT:    vpcmpgtq (%rdi), %ymm0, %ymm0
; CHECK-NEXT:    vpsrlq $63, %ymm1, %ymm1
; CHECK-NEXT:    vpsrlq $63, %ymm0, %ymm0
; CHECK-NEXT:    vmovdqa %ymm0, (%rdx)
; CHECK-NEXT:    vmovdqa %ymm1, 32(%rdx)
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
 %x = load <8 x i64>, <8 x i64>* %xptr
 %y = load <8 x i64>, <8 x i64>* %yptr
 %cmp = icmp slt <8 x i64> %x, %y
 %ext = zext <8 x i1> %cmp to <8 x i64>
 store <8 x i64> %ext, <8 x i64>* %zptr
 ret void
}

define <16 x i8> @var_rotate_v16i8(<16 x i8> %a, <16 x i8> %b) nounwind "min-legal-vector-width"="256" {
; CHECK-LABEL: var_rotate_v16i8:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vmovdqa {{.*#+}} xmm2 = [8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8]
; CHECK-NEXT:    vpsubb %xmm1, %xmm2, %xmm2
; CHECK-NEXT:    vpmovzxbw {{.*#+}} ymm1 =
xmm1[0],zero,xmm1[1],zero,xmm1[2],zero,xmm1[3],zero,xmm1[4],zero,xmm1[5],zero,xmm1[6],zero,xmm1[7],zero,xmm1[8],zero,xmm1[9],zero,xmm1[10],zero,xmm1[11],zero,xmm1[12],zero,xmm1[13],zero,xmm1[14],zero,xmm1[15],zero
;
 CHECK-NEXT:    vpmovzxbw {{.*#+}} ymm0 =
xmm0[0],zero,xmm0[1],zero,xmm0[2],zero,xmm0[3],zero,xmm0[4],zero,xmm0[5],zero,xmm0[6],zero,xmm0[7],zero,xmm0[8],zero,xmm0[9],zero,xmm0[10],zero,xmm0[11],zero,xmm0[12],zero,xmm0[13],zero,xmm0[14],zero,xmm0[15],zero
; CHECK-NEXT:    vpsllvw %ymm1, %ymm0, %ymm1
; CHECK-NEXT:    vpmovzxbw {{.*#+}} ymm2 =
```

```
xmm2[0],zero,xmm2[1],zero,xmm2[2],zero,xmm2[3],zero,xmm2[4],zero,xmm2[5],zero,xmm2[6],zero,xmm2[7],zero,xmm2[8],zero,xmm2[9],zero,xmm2[10],zero,xmm2[11],zero,xmm2[12],zero,xmm2[13],zero,xmm2[14],zero,xmm2[15],zero
; CHECK-NEXT:    vpsrlvw %ymm2, %ymm0, %ymm0
; CHECK-NEXT:    vpor %ymm0, %ymm1, %ymm0
; CHECK-NEXT:    vpmovwb %ymm0, %xmm0
; CHECK-NEXT:    vzeroupper
; CHECK-NEXT:    retq
 %b8 = sub <16 x i8> <i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8>, %b
 %shl = shl <16 x i8> %a, %b
 %lshr = lshr <16 x i8> %a, %b8
 %or = or <16 x i8> %shl, %lshr
 ret <16 x i8> %or
}

define <32 x i8> @var_rotate_v32i8(<32 x i8> %a, <32 x i8> %b) nounwind "min-legal-vector-width"="256" {
; CHECK-LABEL: var_rotate_v32i8:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vpsllw $4, %ymm0, %ymm2
; CHECK-NEXT:    vpsrlw $4, %ymm0, %ymm3
; CHECK-NEXT:    vpternlogq $216, {{\.?LCPI[0-9]+_[0-9]+}}(%rip){1to4}, %ymm2, %ymm3
; CHECK-NEXT:    vpsllw $5, %ymm1, %ymm1
; CHECK-NEXT:    vpblendvb %ymm1, %ymm3, %ymm0, %ymm0
; CHECK-NEXT:    vpsllw $2, %ymm0, %ymm2
; CHECK-NEXT:    vpsrlw $6, %ymm0, %ymm3
; CHECK-NEXT:    vpternlogq $216, {{\.?LCPI[0-9]+_[0-9]+}}(%rip){1to4}, %ymm2, %ymm3
; CHECK-NEXT:    vpaddb %ymm1, %ymm1, %ymm1
; CHECK-NEXT:    vpblendvb %ymm1, %ymm3, %ymm0, %ymm0
; CHECK-NEXT:    vpsrlw $7, %ymm0, %ymm2
; CHECK-NEXT:    vpaddb %ymm0, %ymm0, %ymm3
; CHECK-NEXT:    vpternlogq $248, {{\.?LCPI[0-9]+_[0-9]+}}(%rip),
 %ymm2, %ymm3
; CHECK-NEXT:    vpaddb %ymm1, %ymm1, %ymm1
; CHECK-NEXT:    vpblendvb %ymm1, %ymm3, %ymm0, %ymm0
; CHECK-NEXT:    retq
 %b8 = sub <32 x i8> <i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8>, %b
 %shl = shl <32 x i8> %a, %b
 %lshr = lshr <32 x i8> %a, %b8
 %or = or <32 x i8> %shl, %lshr
 ret <32 x i8> %or
}

define <32 x i8> @splatvar_rotate_v32i8(<32 x i8> %a, <32 x i8> %b) nounwind "min-legal-vector-width"="256"
{
; CHECK-AVX512-LABEL: splatvar_rotate_v32i8:
; CHECK-AVX512:       # %bb.0:
```

```
; CHECK-AVX512-NEXT:    vpand {{\.?LCPI[0-9]+_[0-9]+}}(%rip), %xmm1, %xmm1
; CHECK-AVX512-NEXT:    vpmovzxbq {{.*#+}} xmm2 =
xmm1[0],zero,zero,zero,zero,zero,zero,zero,xmm1[1],zero,zero,zero,zero,zero,zero,zero
; CHECK-AVX512-NEXT:    vpsllw %xmm2, %ymm0, %ymm3
; CHECK-AVX512-NEXT:    vmovdqa {{.*#+}} xmm4 = [8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8]
;
 CHECK-AVX512-NEXT:    vpsubb %xmm1, %xmm4, %xmm1
; CHECK-AVX512-NEXT:    vpcmpeqd %xmm4, %xmm4, %xmm4
; CHECK-AVX512-NEXT:    vpsllw %xmm2, %xmm4, %xmm2
; CHECK-AVX512-NEXT:    vpbroadcastb %xmm2, %ymm2
; CHECK-AVX512-NEXT:    vpmovzxbq {{.*#+}} xmm1 =
xmm1[0],zero,zero,zero,zero,zero,zero,zero,xmm1[1],zero,zero,zero,zero,zero,zero,zero
; CHECK-AVX512-NEXT:    vpsrlw %xmm1, %ymm0, %ymm5
; CHECK-AVX512-NEXT:    vpand %ymm2, %ymm3, %ymm2
; CHECK-AVX512-NEXT:    vpsrlw %xmm1, %xmm4, %xmm0
; CHECK-AVX512-NEXT:    vpsrlw $8, %xmm0, %xmm0
; CHECK-AVX512-NEXT:    vpbroadcastb %xmm0, %ymm0
; CHECK-AVX512-NEXT:    vpternlogq $236, %ymm5, %ymm2, %ymm0
; CHECK-AVX512-NEXT:    retq
;
; CHECK-VBMI-LABEL: splatvar_rotate_v32i8:
; CHECK-VBMI:       # %bb.0:
; CHECK-VBMI-NEXT:    vpand {{\.?LCPI[0-9]+_[0-9]+}}(%rip), %xmm1, %xmm1
; CHECK-VBMI-NEXT:    vpmovzxbq {{.*#+}} xmm2 =
xmm1[0],zero,zero,zero,zero,zero,zero,zero,xmm1[1],zero,zero,zero,zero,zero,zero,zero
; CHECK-VBMI-NEXT:    vpsllw
 %xmm2, %ymm0, %ymm3
; CHECK-VBMI-NEXT:    vpcmpeqd %xmm4, %xmm4, %xmm4
; CHECK-VBMI-NEXT:    vpsllw %xmm2, %xmm4, %xmm2
; CHECK-VBMI-NEXT:    vpbroadcastb %xmm2, %ymm2
; CHECK-VBMI-NEXT:    vpand %ymm2, %ymm3, %ymm2
; CHECK-VBMI-NEXT:    vmovdqa {{.*#+}} xmm3 = [8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8]
; CHECK-VBMI-NEXT:    vpsubb %xmm1, %xmm3, %xmm1
; CHECK-VBMI-NEXT:    vpmovzxbq {{.*#+}} xmm1 =
xmm1[0],zero,zero,zero,zero,zero,zero,zero,xmm1[1],zero,zero,zero,zero,zero,zero,zero
; CHECK-VBMI-NEXT:    vpsrlw %xmm1, %ymm0, %ymm3
; CHECK-VBMI-NEXT:    vpsrlw %xmm1, %xmm4, %xmm0
; CHECK-VBMI-NEXT:    vmovdqa {{.*#+}} ymm1 =
[1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1]
; CHECK-VBMI-NEXT:    vpermb %ymm0, %ymm1, %ymm0
; CHECK-VBMI-NEXT:    vpternlogq $236, %ymm3, %ymm2, %ymm0
; CHECK-VBMI-NEXT:    retq
 %splat = shufflevector <32 x i8> %b, <32 x i8> undef, <32 x i32> zeroinitializer
 %splat8 = sub <32 x i8> <i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8,
 i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8, i8 8>,
%splat
 %shl = shl <32 x i8> %a, %splat
 %lshr = lshr <32 x i8> %a, %splat8
```

```
 %or = or <32 x i8> %shl, %lshr
 ret <32 x i8> %or
}

define <32 x i8> @constant_rotate_v32i8(<32 x i8> %a) nounwind "min-legal-vector-width"="256" {
; CHECK-AVX512-LABEL: constant_rotate_v32i8:
; CHECK-AVX512:       # %bb.0:
; CHECK-AVX512-NEXT:    vpxor %xmm1, %xmm1, %xmm1
; CHECK-AVX512-NEXT:    vpunpckhbw {{.*#+}} ymm2 =
ymm0[8],ymm1[8],ymm0[9],ymm1[9],ymm0[10],ymm1[10],ymm0[11],ymm1[11],ymm0[12],ymm1[12],ymm0[13
],ymm1[13],ymm0[14],ymm1[14],ymm0[15],ymm1[15],ymm0[24],ymm1[24],ymm0[25],ymm1[25],ymm0[26],ym
m1[26],ymm0[27],ymm1[27],ymm0[28],ymm1[28],ymm0[29],ymm1[29],ymm0[30],ymm1[30],ymm0[31],ymm1[
31]
; CHECK-AVX512-NEXT:    vpsllvw {{\.?LCPI[0-9]+_[0-9]+}}(%rip), %ymm2, %ymm2
; CHECK-AVX512-NEXT:    vpsrlw $8, %ymm2, %ymm2
; CHECK-AVX512-NEXT:    vpunpcklbw {{.*#+}} ymm1 =
ymm0[0],ymm1[0],ymm0[1],ymm1[1],ymm0[2],ymm1[2],ymm0[3],ymm1[3],ymm0[4],ymm1[4],ymm0[5],ymm1[
5],ymm0[6],ymm1[6],ymm0[7],ymm1[7],ymm0[16],ymm1[16],ymm0[17],ymm1[17],ymm0[18],ymm1[18],ymm0[
19],ymm1[19],ymm0[20],ymm1[20],ymm0[21],ymm1[21],ymm0[22],ymm1[22],ymm0[23],ymm1[23]
; CHECK-AVX512-NEXT:    vpsllvw {{\.?LCPI[0-9]+_[0-9]+}}(%rip), %ymm1, %ymm1
; CHECK-AVX512-NEXT:    vpsrlw $8, %ymm1, %ymm1
; CHECK-AVX512-NEXT:    vpackuswb %ymm2, %ymm1, %ymm1
; CHECK-AVX512-NEXT:    vpunpckhbw {{.*#+}} ymm2 =
ymm0[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31]
; CHECK-AVX512-NEXT:    vpsllvw {{\.?LCPI[0-9]+_[0-9]+}}(%rip), %ymm2, %ymm2
; CHECK-AVX512-NEXT:    vmovdqa {{.*#+}} ymm3 =
[255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255]
; CHECK-AVX512-NEXT:    vpand %ymm3, %ymm2, %ymm2
; CHECK-AVX512-NEXT:    vpunpcklbw {{.*#+}} ymm0 =
ymm0[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23]
; CHECK-AVX512-NEXT:
  vpsllvw {{\.?LCPI[0-9]+_[0-9]+}}(%rip), %ymm0, %ymm0
; CHECK-AVX512-NEXT:    vpand %ymm3, %ymm0, %ymm0
; CHECK-AVX512-NEXT:    vpackuswb %ymm2, %ymm0, %ymm0
; CHECK-AVX512-NEXT:    vpor %ymm1, %ymm0, %ymm0
; CHECK-AVX512-NEXT:    retq
;
; CHECK-VBMI-LABEL: constant_rotate_v32i8:
; CHECK-VBMI:       # %bb.0:
; CHECK-VBMI-NEXT:    vpunpckhbw {{.*#+}} ymm1 =
ymm0[8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,24,24,25,25,26,26,27,27,28,28,29,29,30,30,31,31]
; CHECK-VBMI-NEXT:    vpsllvw {{\.?LCPI[0-9]+_[0-9]+}}(%rip), %ymm1, %ymm1
; CHECK-VBMI-NEXT:    vpunpcklbw {{.*#+}} ymm2 =
ymm0[0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,16,16,17,17,18,18,19,19,20,20,21,21,22,22,23,23]
; CHECK-VBMI-NEXT:    vpsllvw {{\.?LCPI[0-9]+_[0-9]+}}(%rip), %ymm2, %ymm2
; CHECK-VBMI-NEXT:    vmovdqa {{.*#+}} ymm3 =
[0,2,4,6,8,10,12,14,32,34,36,38,40,42,44,46,16,18,20,22,24,26,28,30,48,50,52,54,56,58,60,62]
; CHECK-VBMI-NEXT:    vpermi2b %ymm1, %ymm2, %ymm3
```

```
; CHECK-VBMI-NEXT:    vpxor %xmm1, %xmm1, %xmm1
; CHECK-VBMI-NEXT:
   vpunpckhbw {{.*#+}} ymm2 =
ymm0[8],ymm1[8],ymm0[9],ymm1[9],ymm0[10],ymm1[10],ymm0[11],ymm1[11],ymm0[12],ymm1[12],ymm0[13
],ymm1[13],ymm0[14],ymm1[14],ymm0[15],ymm1[15],ymm0[24],ymm1[24],ymm0[25],ymm1[25],ymm0[26],ym
m1[26],ymm0[27],ymm1[27],ymm0[28],ymm1[28],ymm0[29],ymm1[29],ymm0[30],ymm1[30],ymm0[31],ymm1[
31]
; CHECK-VBMI-NEXT:    vpsllvw {{\.?LCPI[0-9]+_[0-9]+}}(%rip), %ymm2, %ymm2
; CHECK-VBMI-NEXT:    vpsrlw $8, %ymm2, %ymm2
; CHECK-VBMI-NEXT:    vpunpcklbw {{.*#+}} ymm0 =
ymm0[0],ymm1[0],ymm0[1],ymm1[1],ymm0[2],ymm1[2],ymm0[3],ymm1[3],ymm0[4],ymm1[4],ymm0[5],ymm1[
5],ymm0[6],ymm1[6],ymm0[7],ymm1[7],ymm0[16],ymm1[16],ymm0[17],ymm1[17],ymm0[18],ymm1[18],ymm0[
19],ymm1[19],ymm0[20],ymm1[20],ymm0[21],ymm1[21],ymm0[22],ymm1[22],ymm0[23],ymm1[23]
; CHECK-VBMI-NEXT:    vpsllvw {{\.?LCPI[0-9]+_[0-9]+}}(%rip), %ymm0, %ymm0
; CHECK-VBMI-NEXT:    vpsrlw $8, %ymm0, %ymm0
; CHECK-VBMI-NEXT:    vpackuswb %ymm2, %ymm0, %ymm0
; CHECK-VBMI-NEXT:    vpor %ymm0, %ymm3, %ymm0
; CHECK-VBMI-NEXT:    retq
  %shl = shl <32 x i8> %a, <i8 0, i8 1, i8 2, i8 3, i8 4, i8 5, i8 6, i8 7, i8 8, i8 7, i8 6, i8 5, i8 4, i8 3, i8 2, i8 1, i8 0, i8
1, i8 2, i8 3, i8 4, i8 5, i8 6, i8 7, i8 8, i8 7, i8 6, i8 5, i8 4, i8 3, i8 2, i8 1>
 %lshr = lshr <32 x i8> %a, <i8 8, i8 7, i8 6, i8 5, i8 4, i8 3, i8 2, i8 1, i8 0, i8 1, i8 2, i8 3, i8 4, i8 5, i8 6, i8 7, i8 8,
i8 7, i8 6, i8 5, i8 4, i8 3, i8 2, i8 1, i8 0, i8 1, i8 2, i8 3, i8 4, i8 5, i8 6, i8 7>
 %or = or <32 x i8> %shl, %lshr
 ret <32 x i8> %or
}

define <32 x i8> @splatconstant_rotate_v32i8(<32 x i8> %a) nounwind "min-legal-vector-width"="256" {
; CHECK-LABEL: splatconstant_rotate_v32i8:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vpsllw $4, %ymm0, %ymm1
; CHECK-NEXT:    vpsrlw $4, %ymm0, %ymm0
; CHECK-NEXT:    vpternlogq $216, {{\.?LCPI[0-9]+_[0-9]+}}(%rip){1to4}, %ymm1, %ymm0
; CHECK-NEXT:    retq
 %shl = shl <32 x i8> %a, <i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8
4, i8 4,
 i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4>
 %lshr = lshr <32 x i8> %a, <i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4,
i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4>
 %or = or <32 x i8> %shl, %lshr
 ret <32 x i8> %or
}

define <32 x i8> @splatconstant_rotate_mask_v32i8(<32 x i8> %a) nounwind "min-legal-vector-width"="256" {
; CHECK-LABEL: splatconstant_rotate_mask_v32i8:
; CHECK:       # %bb.0:
; CHECK-NEXT:    vpsllw $4, %ymm0, %ymm1
; CHECK-NEXT:    vpsrlw $4, %ymm0, %ymm0
; CHECK-NEXT:    vpternlogq $216, {{\.?LCPI[0-9]+_[0-9]+}}(%rip){1to4}, %ymm1, %ymm0
```

```
; CHECK-NEXT:    vpand {{\.?LCPI[0-9]+_[0-9]+}}(%rip), %ymm0, %ymm0
; CHECK-NEXT:    retq
 %shl = shl <32 x i8> %a, <i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8
 4, i8 4>
 %lshr = lshr <32 x i8> %a, <i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4, i8 4>
 %rmask = and <32 x i8> %lshr, <i8 55, i8 55, i8 55, i8 55, i8 55, i8 55, i8 55, i8 55, i8 55, i8 55, i8 55, i8 55, i8 55, i8 55, i8 55, i8 55, i8 55, i8 55, i8 55, i8 55, i8 55, i8 55, i8 55, i8 55, i8 55, i8 55, i8 55, i8 55, i8 55, i8 55, i8 55, i8 55>
 %lmask = and <32 x i8> %shl, <i8 33, i8 33, i8 33, i8 33, i8 33, i8 33, i8 33, i8 33, i8 33, i8 33, i8 33, i8 33, i8 33, i8 33, i8 33, i8 33, i8 33, i8 33, i8 33, i8 33, i8 33, i8 33, i8 33, i8 33, i8 33, i8 33, i8 33, i8 33, i8 33, i8 33, i8 33, i8 33>
 %or = or <32 x i8> %lmask, %rmask
 ret <32 x i8> %or
}
```

==

The LLVM Project is under the Apache License v2.0 with LLVM Exceptions:

==

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation

source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work
includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided

that such additional
attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License
does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including
negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such
obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

---- LLVM Exceptions to the Apache 2.0 License ----

As an exception, if, as a result of your compiling your source code, portions
of this Software are embedded into an Object form of such source code, you
may redistribute such embedded portions in such Object form without complying
with the conditions of Sections 4(a), 4(b) and 4(d) of the License.

In addition, if you combine or link compiled

forms of this Software with
software that is licensed under the GPLv2 ("Combined Software") and if a
court of competent jurisdiction determines that the patent provision (Section
3), the indemnity provision (Section 9) or other Section of the License
conflicts with the conditions of the GPLv2, you may retroactively and
prospectively choose to deem waived or otherwise exclude such Section(s) of
the License, but only in their entirety and only with respect to the Combined
Software.

==
Software from third parties included in the LLVM Project:
==
The LLVM Project contains third party software which is under different license
terms. All such code will be identified clearly using at least one of two
mechanisms:
1) It will be in a separate directory tree with its own `LICENSE.txt` or
  `LICENSE` file at the top containing the specific
 license and restrictions
  which apply to that software, or
2) It will contain specific license and restriction terms at the top of every
  file.

==
Legacy LLVM License (https://llvm.org/docs/DeveloperPolicy.html#legacy):
==
University of Illinois/NCSA
Open Source License

Copyright (c) 2009-2019 Polly Team
All rights reserved.

Developed by:

  Polly Team

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal with
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies
of the Software, and to permit persons to whom the Software is furnished to do
so, subject to the following conditions:

  * Redistributions of source code must retain
 the above copyright notice,
    this list of conditions and the following disclaimers.

  * Redistributions in binary form must reproduce the above copyright notice,
    this list of conditions and the following disclaimers in the

documentation and/or other materials provided with the distribution.

* Neither the names of the Polly Team, copyright holders, nor the names of
its contributors may be used to endorse or promote products derived from
this Software without specific prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

==========

The LLVM Project is under the Apache License v2.0 with LLVM Exceptions:

==========

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the
power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical

transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached
to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition,
"submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative
Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable

(except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work
includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional
attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License
does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including
negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,

or other liability obligations and/or rights consistent with this
License. However, in accepting such
obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

---- LLVM Exceptions to the Apache 2.0 License ----

As an exception, if, as a result of your compiling your source code, portions
of this Software are embedded into an Object form of such source code, you
may redistribute such embedded portions in such Object form without complying
with the conditions of Sections 4(a), 4(b) and 4(d) of the License.

In addition, if you combine or link compiled
forms of this Software with
software that is licensed under the GPLv2 ("Combined Software") and if a
court of competent jurisdiction determines that the patent provision (Section

3), the indemnity provision (Section 9) or other Section of the License
conflicts with the conditions of the GPLv2, you may retroactively and
prospectively choose to deem waived or otherwise exclude such Section(s) of
the License, but only in their entirety and only with respect to the Combined
Software.

==
Software from third parties included in the LLVM Project:
==
The LLVM Project contains third party software which is under different license
terms. All such code will be identified clearly using at least one of two
mechanisms:
1) It will be in a separate directory tree with its own `LICENSE.txt` or
  `LICENSE` file at the top containing the specific
 license and restrictions
  which apply to that software, or
2) It will contain specific license and restriction terms at the top of every
  file.

==
Legacy LLVM License (https://llvm.org/docs/DeveloperPolicy.html#legacy):
==

The compiler_rt library is dual licensed under both the University of Illinois
"BSD-Like" license and the MIT license. As a user of this code you may choose
to use it under either license. As a contributor, you agree to allow your code
to be used under both.

Full text of the relevant licenses is included below.

==

University of Illinois/NCSA
Open Source License

Copyright (c) 2009-2019 by the contributors listed in CREDITS.TXT

All rights reserved.

Developed by:

  LLVM Team

  University of Illinois at Urbana-Champaign

 http://llvm.org

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal with the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimers.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimers in the documentation and/or other materials provided with the distribution.

* Neither the names of the LLVM Team, University of Illinois at Urbana-Champaign, nor the names of its contributors may be used to endorse or promote products derived from this Software without specific prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

==========

Copyright (c) 2009-2015 by the contributors listed in CREDITS.TXT

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,

```
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
; RUN: opt -mtriple=aarch64-linux-gnu -mattr=+sve -scalarize-masked-mem-intrin -S < %s | FileCheck %s

; Testing that masked scatters operating on scalable vectors that are
; packed in SVE registers are not scalarized.

; CHECK-LABEL: @masked_scatter_nxv4i32(
; CHECK: call void @llvm.masked.scatter.nxv4i32
define void @masked_scatter_nxv4i32(<vscale x 4 x i32> %data, <vscale x 4 x i32*> %ptrs, <vscale x 4 x i1>
%masks) {
 call void @llvm.masked.scatter.nxv4i32(<vscale x 4 x i32> %data, <vscale x 4 x i32*> %ptrs, i32 0, <vscale x 4 x
i1> %masks)
 ret void
}

; Testing that masked scatters operating on scalable vectors of FP
; data that is packed in SVE registers are not scalarized.

; CHECK-LABEL: @masked_scatter_nxv2f64(
; CHECK: call void @llvm.masked.scatter.nxv2f64
define void @masked_scatter_nxv2f64(<vscale x 2 x double> %data, <vscale x 2 x double*> %ptrs, <vscale x 2 x
i1> %masks) {
 call void @llvm.masked.scatter.nxv2f64(<vscale x 2 x double> %data, <vscale x 2 x double*> %ptrs,
 i32 0, <vscale x 2 x i1> %masks)
 ret void
}

; Testing that masked scatters operating on scalable vectors of FP
; data that is unpacked in SVE registers are not scalarized.

; CHECK-LABEL: @masked_scatter_nxv2f16(
; CHECK: call void @llvm.masked.scatter.nxv2f16
define void @masked_scatter_nxv2f16(<vscale x 2 x half> %data, <vscale x 2 x half*> %ptrs, <vscale x 2 x i1>
%masks) {
 call void @llvm.masked.scatter.nxv2f16(<vscale x 2 x half> %data, <vscale x 2 x half*> %ptrs, i32 0, <vscale x 2
x i1> %masks)
 ret void
}

; Testing that masked scatters operating on 64-bit fixed vectors are
; scalarized because NEON doesn't have support for masked scatter
; instructions.

; CHECK-LABEL: @masked_scatter_v2f32(
; CHECK-NOT: @llvm.masked.scatter.v2f32(
define void @masked_scatter_v2f32(<2 x float> %data, <2 x float*> %ptrs, <2 x i1> %masks) {
 call void @llvm.masked.scatter.v2f32(<2 x float> %data, <2 x float*> %ptrs, i32 0, <2 x i1> %masks)
```

```
 ret void
}

; Testing that masked scatters operating
 on 128-bit fixed vectors are
; scalarized because NEON doesn't have support for masked scatter
; instructions and because we are not targeting fixed width SVE.

; CHECK-LABEL: @masked_scatter_v4i32(
; CHECK-NOT: @llvm.masked.scatter.v4i32(
define void @masked_scatter_v4i32(<4 x i32> %data, <4 x i32*> %ptrs, <4 x i1> %masks) {
 call void @llvm.masked.scatter.v4i32(<4 x i32> %data, <4 x i32*> %ptrs, i32 0, <4 x i1> %masks)
 ret void
}

declare void @llvm.masked.scatter.nxv4i32(<vscale x 4 x i32> %data, <vscale x 4 x i32*> %ptrs, i32 %align,
<vscale x 4 x i1> %masks)
declare void @llvm.masked.scatter.nxv2f64(<vscale x 2 x double> %data, <vscale x 2 x double*> %ptrs, i32
%align, <vscale x 2 x i1> %masks)
declare void @llvm.masked.scatter.nxv2f16(<vscale x 2 x half> %data, <vscale x 2 x half*> %ptrs, i32 %align,
<vscale x 2 x i1> %masks)
declare void @llvm.masked.scatter.v2f32(<2 x float> %data, <2 x float*> %ptrs, i32 %align, <2 x i1> %masks)
declare void @llvm.masked.scatter.v4i32(<4
 x i32> %data, <4 x i32*> %ptrs, i32 %align, <4 x i1> %masks)
```

```
================================================================================
The LLVM Project is under the Apache License v2.0 with LLVM Exceptions:
================================================================================
```

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all
other entities that control, are controlled by, or are under common
control with that entity. For the purposes of this definition,
"control" means (i) the
power, direct or indirect, to cause the
direction or management of such entity, whether by contract or

otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached
to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition,
"submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed

as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional
attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License
does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including
negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be

liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such
obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

---- LLVM Exceptions to the Apache 2.0 License ----

As an exception, if, as a result of your compiling your source code, portions
of this Software are embedded into an Object form of such source code, you
may redistribute such embedded portions in such Object form without complying
with the conditions of Sections 4(a), 4(b) and 4(d) of the License.

In addition, if you combine or link compiled
 forms of this Software with
software that is licensed under the GPLv2 ("Combined Software") and if a
court of competent jurisdiction determines that the patent provision (Section
3), the indemnity provision (Section 9) or other Section of the License
conflicts with the conditions of the GPLv2, you may retroactively and
prospectively choose to deem waived or otherwise exclude such Section(s) of
the License, but only in their entirety and only with respect to the Combined
Software.

==
Software from third parties included in the LLVM Project:
==
The LLVM Project contains third party software which is under different license
terms. All such code will be identified clearly using at least one of two
mechanisms:
1) It will be in a separate directory tree with its own `LICENSE.txt` or
  `LICENSE` file at the top containing the specific
 license and restrictions
  which apply to that software, or
2) It will contain specific license and restriction terms at the top of every
  file.

==
Legacy LLVM License (https://llvm.org/docs/DeveloperPolicy.html#legacy):
==

The libunwind library is dual licensed under both the University of Illinois
"BSD-Like" license and the MIT license. As a user of this code you may choose
to use it under either license. As a contributor, you agree to allow your code
to be used under both.

Full text of the relevant licenses is included below.

==

University of Illinois/NCSA
Open Source License

Copyright (c) 2009-2019 by the contributors listed in CREDITS.TXT

All rights reserved.

Developed by:

LLVM Team

University of Illinois at Urbana-Champaign

http://llvm.org

Permission
is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal with
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies
of the Software, and to permit persons to whom the Software is furnished to do
so, subject to the following conditions:

* Redistributions of source code must retain the above copyright notice,
this list of conditions and the following disclaimers.

* Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimers in the
documentation and/or other materials provided with the distribution.

* Neither the names of the LLVM Team, University of Illinois at
Urbana-Champaign, nor the names of its contributors may be used to
endorse or promote products derived from
this Software without specific
prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

==

Copyright (c) 2009-2014 by the contributors listed in CREDITS.TXT

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the

Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

```
static_library("BSD-Archive") {
 output_name = "lldbPluginObjectContainerBSDArchive"
 configs += [ "//llvm/utils/gn/build:lldb_code" ]
 deps = [
   "//lldb/source/Core",
   "//lldb/source/Host",
   "//lldb/source/Symbol",
   "//llvm/lib/Support",
 ]
 sources = [ "ObjectContainerBSDArchive.cpp" ]
}
```

1.29 tiny-xml 2_6_2

1.29.1 Available under license :

No license file was found, but licenses were detected in source scan.

TinyXML is released under the ZLib license, so you can use it in open source or commercial code. The details of the license are at the top of every source file.
TinyXML is released under the zlib license:
Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:
1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Found in path(s):
* /opt/cola/permits/1222902257_1637000982.24/0/tinyxml-2-6-2-zip/tinyxml/docs/index.html
No license file was found, but licenses were detected in source scan.

```
/*
```

www.sourceforge.net/projects/tinyxml

This software is provided 'as-is', without any express or implied
warranty. In no event will the authors be held liable for any
damages arising from the use of this software.

Permission is granted to anyone to use this software for any
purpose, including commercial applications, and to alter it and
redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must
not claim that you wrote the original software. If you use this
software in a product, an acknowledgment in the product documentation
would be appreciated but is not required.

2. Altered source versions must be plainly marked as such, and
must not be misrepresented as being the original software.

3. This notice may not be removed or altered from any source
distribution.
*/

Found in path(s):
* /opt/cola/permits/1222902257_1637000982.24/0/tinyxml-2-6-2-zip/tinyxml/tinystr.cpp
*
 /opt/cola/permits/1222902257_1637000982.24/0/tinyxml-2-6-2-zip/tinyxml/tinystr.h
No license file was found, but licenses were detected in source scan.

/*
www.sourceforge.net/projects/tinyxml
Original code (2.0 and earlier)copyright (c) 2000-2006 Lee Thomason (www.grinninglizard.com)

This software is provided 'as-is', without any express or implied
warranty. In no event will the authors be held liable for any
damages arising from the use of this software.

Permission is granted to anyone to use this software for any
purpose, including commercial applications, and to alter it and
redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must
not claim that you wrote the original software. If you use this
software in a product, an acknowledgment in the product documentation
would be appreciated but is not required.

2. Altered source versions must be plainly marked as such, and
must not be misrepresented as being the original software.

3. This notice may not be removed or altered from any source
distribution.
*/

Found
in path(s):
* /opt/cola/permits/1222902257_1637000982.24/0/tinyxml-2-6-2-zip/tinyxml/tinyxmlerror.cpp
No license file was found, but licenses were detected in source scan.

/*
www.sourceforge.net/projects/tinyxml
Original code by Lee Thomason (www.grinninglizard.com)

This software is provided 'as-is', without any express or implied
warranty. In no event will the authors be held liable for any
damages arising from the use of this software.

Permission is granted to anyone to use this software for any
purpose, including commercial applications, and to alter it and
redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must
not claim that you wrote the original software. If you use this
software in a product, an acknowledgment in the product documentation
would be appreciated but is not required.

2. Altered source versions must be plainly marked as such, and
must not be misrepresented as being the original software.

3. This notice may not be removed or altered from any source
distribution.
*/

Found in path(s):
*
/opt/cola/permits/1222902257_1637000982.24/0/tinyxml-2-6-2-zip/tinyxml/tinyxmlparser.cpp
No license file was found, but licenses were detected in source scan.

00009 Permission is granted to anyone to use this software for any
00013 1. The origin of this software must not be misrepresented; you must
00018 2. Altered source versions must be plainly marked as such, and
00021 3. This notice may not be removed or altered from any source

Found in path(s):

* /opt/cola/permits/1222902257_1637000982.24/0/tinyxml-2-6-2-zip/tinyxml/docs/tinyxml_8h_source.html
No license file was found, but licenses were detected in source scan.

/** @mainpage

<h1> TinyXML </h1>

TinyXML is a simple, small, C++ XML parser that can be easily
integrated into other programs.

<h2> What it does. </h2>

In brief, TinyXML parses an XML document, and builds from that a
Document Object Model (DOM) that can be read, modified, and saved.

XML stands for "eXtensible Markup Language." It allows you to create
your own document markups. Where HTML does a very good job of marking
documents for browsers, XML allows you to define any kind of document
markup, for example a document that describes a "to do" list for an
organizer application. XML is a very structured and convenient format.
All those random file formats created to store application data can
all be replaced with XML. One parser for everything.

The best place for the complete, correct, and quite frankly hard to
read spec is at
http://www.w3.org/TR/2004/REC-xml-20040204/.
An intro to XML
(that I really like) can be found at
http://skew.org/xml/tutorial.

There are different ways to access and interact with XML data.
TinyXML uses a Document Object Model (DOM), meaning the XML data is parsed
into a C++ objects that can be browsed and manipulated, and then
written to disk or another output stream. You can also construct an XML document
from scratch with C++ objects and write this to disk or another output
stream.

TinyXML is designed to be easy and fast to learn. It is two headers
and four cpp files. Simply add these to your project and off you go.
There is an example file - xmltest.cpp - to get you started.

TinyXML is released under the ZLib license,
so you can use it in open source or commercial code. The details
of the license are at the top of every source file.

TinyXML attempts to be a flexible parser, but with truly correct and
compliant XML output. TinyXML should compile on

any reasonably C++
compliant system. It does not rely on exceptions or RTTI. It can be
compiled with or without STL support. TinyXML fully supports
the UTF-8 encoding, and the first 64k character entities.

<h2> What it doesn't do. </h2>

TinyXML doesn't parse or use DTDs (Document Type Definitions) or XSLs
(eXtensible Stylesheet Language.) There are other parsers out there
(check out www.sourceforge.org, search for XML) that are much more fully
featured. But they are also much bigger, take longer to set up in
your project, have a higher learning curve, and often have a more
restrictive license. If you are working with browsers or have more
complete XML needs, TinyXML is not the parser for you.

The following DTD syntax will not parse at this time in TinyXML:

@verbatim
<!DOCTYPE Archiv [
<!ELEMENT Comment (#PCDATA)>
]>
@endverbatim

because TinyXML sees this as a !DOCTYPE node with an illegally
embedded !ELEMENT node. This may be addressed in the future.

<h2> Tutorials. </h2>

For
the impatient, here is a tutorial to get you going. A great way to get started,
but it is worth your time to read this (very short) manual completely.

- @subpage tutorial0

<h2> Code Status. </h2>

TinyXML is mature, tested code. It is very stable. If you find
bugs, please file a bug report on the sourceforge web site
(www.sourceforge.net/projects/tinyxml). We'll get them straightened
out as soon as possible.

There are some areas of improvement; please check sourceforge if you are
interested in working on TinyXML.

<h2> Related Projects </h2>

TinyXML projects you may find useful! (Descriptions provided by the projects.)

 TinyXPath (http://tinyxpath.sourceforge.net). TinyXPath is a small footprint
   XPath syntax decoder, written in C++.
 TinyXML++ (http://code.google.com/p/ticpp/). TinyXML++ is a completely new
   interface to TinyXML that uses MANY of the C++ strengths. Templates,
  exceptions, and much better error handling.

<h2>
 Features </h2>

<h3> Using STL </h3>

TinyXML can be compiled to use or not use STL. When using STL, TinyXML
uses the std::string class, and fully supports std::istream, std::ostream,
operator<<, and operator>>. Many API methods have both 'const char*' and
'const std::string&' forms.

When STL support is compiled out, no STL files are included whatsoever. All
the string classes are implemented by TinyXML itself. API methods
all use the 'const char*' form for input.

Use the compile time #define:

 TIXML_USE_STL

to compile one version or the other. This can be passed by the compiler,
or set as the first line of "tinyxml.h".

Note: If compiling the test code in Linux, setting the environment
variable TINYXML_USE_STL=YES/NO will control STL compilation. In the
Windows project file, STL and non STL targets are provided. In your project,
It's probably easiest to add the line "#define TIXML_USE_STL" as the first
line of tinyxml.h.

<h3> UTF-8 </h3>

TinyXML supports UTF-8 allowing
 to manipulate XML files in any language. TinyXML
also supports "legacy mode" - the encoding used before UTF-8 support and
probably best described as "extended ascii".

Normally, TinyXML will try to detect the correct encoding and use it. However,
by setting the value of TIXML_DEFAULT_ENCODING in the header file, TinyXML
can be forced to always use one encoding.

TinyXML will assume Legacy Mode until one of the following occurs:

  If the non-standard but common "UTF-8 lead bytes" (0xef 0xbb 0xbf)
  begin the file or data stream, TinyXML will read it as UTF-8.
  If the declaration tag is read, and it has an encoding="UTF-8", then
  TinyXML will read it as UTF-8.
  If the declaration tag is read, and it has no encoding specified, then TinyXML will
  read it as UTF-8.
  If the declaration tag is read, and it has an encoding="something else", then TinyXML
  will read it as Legacy Mode. In legacy mode, TinyXML will work as it did before. It's

 not clear what that mode does exactly, but old content should keep working.
  Until one of the above criteria is met, TinyXML runs in Legacy Mode.

What happens if the encoding is incorrectly set or detected? TinyXML will try
to read and pass through text seen as improperly encoded. You may get some strange results or
mangled characters. You may want to force TinyXML to the correct mode.

You may force TinyXML to Legacy Mode by using LoadFile(TIXML_ENCODING_LEGACY) or
LoadFile(filename, TIXML_ENCODING_LEGACY). You may force it to use legacy mode all
the time by setting TIXML_DEFAULT_ENCODING = TIXML_ENCODING_LEGACY. Likewise, you may
force it to TIXML_ENCODING_UTF8 with the same technique.

For English users, using English XML, UTF-8 is the same as low-ASCII. You
don't need to be aware of UTF-8 or change your code in any way. You can think
of UTF-8 as a "superset" of ASCII.

UTF-8 is not a double byte format - but it is a standard encoding of Unicode!
TinyXML
 does not use or directly support wchar, TCHAR, or Microsoft's _UNICODE at this time.
It is common to see the term "Unicode" improperly refer to UTF-16, a wide byte encoding
of unicode. This is a source of confusion.

For "high-ascii" languages - everything not English, pretty much - TinyXML can
handle all languages, at the same time, as long as the XML is encoded
in UTF-8. That can be a little tricky, older programs and operating systems
tend to use the "default" or "traditional" code page. Many apps (and almost all
modern ones) can output UTF-8, but older or stubborn (or just broken) ones
still output text in the default code page.

For example, Japanese systems traditionally use SHIFT-JIS encoding.
Text encoded as SHIFT-JIS can not be read by TinyXML.
A good text editor can import SHIFT-JIS and then save as UTF-8.

The Skew.org link does a great
job covering the encoding issue.

The test file "utf8test.xml" is an XML containing English,

Spanish, Russian,
and Simplified Chinese. (Hopefully they are translated correctly). The file
"utf8test.gif" is a screen capture of the XML file, rendered in IE. Note that
if you don't have the correct fonts (Simplified Chinese or Russian) on your
system, you won't see output that matches the GIF file even if you can parse
it correctly. Also note that (at least on my Windows machine) console output
is in a Western code page, so that Print() or printf() cannot correctly display
the file. This is not a bug in TinyXML - just an OS issue. No data is lost or
destroyed by TinyXML. The console just doesn't render UTF-8.

<h3> Entities </h3>
TinyXML recognizes the pre-defined "character entities", meaning special
characters. Namely:

@verbatim
& &
< <
> >
" "
' '
@endverbatim

These are recognized when the XML document is read, and translated to there
UTF-8 equivalents. For instance, text with the XML of:

@verbatim
Far & Away
@endverbatim

will have the Value()
of "Far & Away" when queried from the TiXmlText object,
and will be written back to the XML stream/file as an ampersand. Older versions
of TinyXML "preserved" character entities, but the newer versions will translate
them into characters.

Additionally, any character can be specified by its Unicode code point:
The syntax " " or " " are both to the non-breaking space characher.

<h3> Printing </h3>
TinyXML can print output in several different ways that all have strengths and limitations.

- Print(FILE*). Output to a std-C stream, which includes all C files as well as stdout.
 - "Pretty prints", but you don't have control over printing options.
 - The output is streamed directly to the FILE object, so there is no memory overhead
   in the TinyXML code.
 - used by Print() and SaveFile()

- operator<<. Output to a c++ stream.
 - Integrates with standart C++ iostreams.
 - Outputs in "network printing" mode without line breaks. Good for network transmission
   and moving XML
 between C++ objects, but hard for a human to read.

- TiXmlPrinter. Output to a std::string or memory buffer.
 - API is less concise
 - Future printing options will be put here.
 - Printing may change slightly in future versions as it is refined and expanded.

<h3> Streams </h3>
With TIXML_USE_STL on TinyXML supports C++ streams (operator <<,>>) streams as well as C (FILE*) streams. There are some differences that you may need to be aware of.

C style output:
 - based on FILE*
 - the Print() and SaveFile() methods

 Generates formatted output, with plenty of white space, intended to be as
 human-readable as possible. They are very fast, and tolerant of ill formed
 XML documents. For example, an XML document that contains 2 root elements
 and 2 declarations, will still print.

C style input:
 - based on FILE*
 - the Parse() and LoadFile() methods

 A fast, tolerant read. Use whenever you don't need the C++ streams.

C++ style output:
 - based on std::ostream
 - operator<<

 Generates
 condensed output, intended for network transmission rather than
 readability. Depending on your system's implementation of the ostream class,
 these may be somewhat slower. (Or may not.) Not tolerant of ill formed XML:
 a document should contain the correct one root element. Additional root level
 elements will not be streamed out.

C++ style input:
 - based on std::istream
 - operator>>

 Reads XML from a stream, making it useful for network transmission. The tricky
 part is knowing when the XML document is complete, since there will almost
 certainly be other data in the stream. TinyXML will assume the XML data is

complete after it reads the root element. Put another way, documents that
are ill-constructed with more than one root element will not read correctly.
Also note that operator>> is somewhat slower than Parse, due to both
implementation of the STL and limitations of TinyXML.

<h3> White space </h3>
The world simply does not agree on whether white space should be kept,
or condensed.
For example, pretend the '_' is a space, and look at "Hello____world". HTML, and
at least some XML parsers, will interpret this as "Hello_world". They condense white
space. Some XML parsers do not, and will leave it as "Hello____world". (Remember
to keep pretending the _ is a space.) Others suggest that __Hello___world__ should become
Hello___world.

It's an issue that hasn't been resolved to my satisfaction. TinyXML supports the
first 2 approaches. Call TiXmlBase::SetCondenseWhiteSpace(bool) to set the desired behavior.
The default is to condense white space.

If you change the default, you should call TiXmlBase::SetCondenseWhiteSpace(bool)
before making any calls to Parse XML data, and I don't recommend changing it after
it has been set.

<h3> Handles </h3>

Where browsing an XML document in a robust way, it is important to check
for null returns from method calls. An error safe implementation can
generate a lot of code like:

@verbatim
TiXmlElement* root = document.FirstChildElement(
"Document");
if (root)
{
TiXmlElement* element = root->FirstChildElement("Element");
if (element)
{
TiXmlElement* child = element->FirstChildElement("Child");
if (child)
{
TiXmlElement* child2 = child->NextSiblingElement("Child");
if (child2)
{
// Finally do something useful.
@endverbatim

Handles have been introduced to clean this up. Using the TiXmlHandle class,
the previous code reduces to:

@verbatim
TiXmlHandle docHandle(&document);
TiXmlElement* child2 = docHandle.FirstChild("Document").FirstChild("Element").Child("Child", 1
).ToElement();
if (child2)
{
 // do something useful
@endverbatim

Which is much easier to deal with. See TiXmlHandle for more information.

<h3> Row and Column tracking </h3>
Being able to track nodes and attributes back to their origin location
in source files can be very important for some applications. Additionally,
knowing where parsing errors occured in the original source can be very
time
 saving.

TinyXML can tracks the row and column origin of all nodes and attributes
in a text file. The TiXmlBase::Row() and TiXmlBase::Column() methods return
the origin of the node in the source text. The correct tabs can be
configured in TiXmlDocument::SetTabSize().

<h2> Using and Installing </h2>

To Compile and Run xmltest:

A Linux Makefile and a Windows Visual C++ .dsw file is provided.
Simply compile and run. It will write the file demotest.xml to your
disk and generate output on the screen. It also tests walking the
DOM by printing out the number of nodes found using different
techniques.

The Linux makefile is very generic and runs on many systems - it
is currently tested on mingw and
MacOSX. You do not need to run 'make depend'. The dependecies have been
hard coded.

<h3>Windows project file for VC6</h3>

tinyxml: tinyxml library, non-STL
tinyxmlSTL: tinyxml library, STL
tinyXmlTest: test app, non-STL
tinyXmlTestSTL: test app,

STL

<h3>Makefile</h3>
At the top of the makefile you can set:

PROFILE, DEBUG, and TINYXML_USE_STL. Details (such that they are) are in
the makefile.

In the tinyxml directory, type "make clean" then "make". The executable
file 'xmltest' will be created.

<h3>To Use in an Application:</h3>

Add tinyxml.cpp, tinyxml.h, tinyxmlerror.cpp, tinyxmlparser.cpp, tinystr.cpp, and tinystr.h to your
project or make file. That's it! It should compile on any reasonably
compliant C++ system. You do not need to enable exceptions or
RTTI for TinyXML.

<h2> How TinyXML works. </h2>

An example is probably the best way to go. Take:
@verbatim
<?xml version="1.0" standalone=no>
<!-- Our to do list data -->
<ToDo>
<Item priority="1"> Go to the <bold>Toy store!</bold></Item>
<Item priority="2"> Do bills</Item>
</ToDo>
@endverbatim

Its not much of a To Do list, but it will do. To read this file
(say "demo.xml") you would create a document, and parse it in:
@verbatim
TiXmlDocument
doc("demo.xml");
doc.LoadFile();
@endverbatim

And its ready to go. Now lets look at some lines and how they
relate to the DOM.

@verbatim
<?xml version="1.0" standalone=no>
@endverbatim

The first line is a declaration, and gets turned into the
TiXmlDeclaration class. It will be the first child of the
document node.

This is the only directive/special tag parsed by TinyXML.
Generally directive tags are stored in TiXmlUnknown so the
commands wont be lost when it is saved back to disk.

@verbatim
<!-- Our to do list data -->
@endverbatim

A comment. Will become a TiXmlComment object.

@verbatim
<ToDo>
@endverbatim

The "ToDo" tag defines a TiXmlElement object. This one does not have
any attributes, but does contain 2 other elements.

@verbatim
<Item priority="1">
@endverbatim

Creates another TiXmlElement which is a child of the "ToDo" element.
This element has 1 attribute, with the name "priority" and the value
"1".

@verbatim
Go to the
@endverbatim

A TiXmlText.
This is a leaf node and cannot contain other nodes.
It is a child of the "Item" TiXmlElement.

@verbatim
<bold>
@endverbatim

Another TiXmlElement, this one a child of the "Item" element.

Etc.

Looking at the entire object tree, you end up with:

@verbatim
TiXmlDocument "demo.xml"
 TiXmlDeclaration "version='1.0'" "standalone=no"
 TiXmlComment " Our to do list data"
 TiXmlElement "ToDo"
  TiXmlElement "Item" Attribtutes: priority = 1
   TiXmlText "Go to the "
   TiXmlElement "bold"
    TiXmlText "Toy store!"
  TiXmlElement "Item" Attributes: priority=2
   TiXmlText "Do bills"
@endverbatim

<h2> Documentation </h2>

The documentation is build with Doxygen, using the 'dox'
configuration file.

<h2> License </h2>

TinyXML is released under the zlib license:

This software is provided 'as-is', without any express or implied
warranty. In no event will the authors be held liable for any
damages arising from the use of this software.

Permission is granted to
 anyone to use this software for any
purpose, including commercial applications, and to alter it and
redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must
not claim that you wrote the original software. If you use this
software in a product, an acknowledgment in the product documentation
would be appreciated but is not required.

2. Altered source versions must be plainly marked as such, and
must not be misrepresented as being the original software.

3. This notice may not be removed or altered from any source
distribution.

<h2> References </h2>

The World Wide Web Consortium is the definitive standard body for
XML, and their web pages contain huge amounts of information.

The definitive spec:

http://www.w3.org/TR/2004/REC-xml-20040204/

I also recommend "XML Pocket Reference" by Robert Eckstein and published by
OReilly...the book that got
the whole thing started.

<h2> Contributors, Contacts, and a Brief History </h2>

Thanks very much to everyone who sends suggestions, bugs, ideas, and
encouragement. It all helps, and makes this project fun. A special thanks
to the contributors on the web pages that keep it lively.

So many people have sent in bugs and ideas, that rather than list here
we try to give credit due in the "changes.txt" file.

TinyXML was originally written by Lee Thomason. (Often the "I" still
in the documentation.) Lee reviews changes and releases new versions,
with the help of Yves Berquin, Andrew Ellerton, and the tinyXml community.

We appreciate your suggestions, and would love to know if you
use TinyXML. Hopefully you will enjoy it and find it useful.
Please post questions, comments, file bugs, or contact us at:

www.sourceforge.net/projects/tinyxml

Lee Thomason, Yves Berquin, Andrew Ellerton
*/

Found in path(s):
* /opt/cola/permits/1222902257_1637000982.24/0/tinyxml-2-6-2-zip/tinyxml/readme.txt
No license file was found, but licenses were detected in source scan.

00008 Permission is granted to anyone to use this software for any
00012 1. The origin of this software must not be misrepresented; you must
00017 2. Altered source versions must be plainly marked as such, and
00020 3. This notice may not be removed or altered from any source

Found in path(s):
* /opt/cola/permits/1222902257_1637000982.24/0/tinyxml-2-6-2-zip/tinyxml/docs/tinystr_8h_source.html
No license file was found, but licenses were detected in source scan.

/*
www.sourceforge.net/projects/tinyxml
Original code by Lee Thomason (www.grinninglizard.com)

This software is provided 'as-is', without any express or implied
warranty. In no event will the authors be held liable for any
damages arising from the use of this software.

Permission is granted to anyone to use this software for any
purpose, including commercial applications, and to alter it and
redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must
not claim that you wrote the original software. If you use this
software in a product, an acknowledgment in the product documentation
would be appreciated but is not required.

2. Altered source versions must be plainly marked as such, and
must not be misrepresented as being the original software.

3. This notice may not be removed or altered from any source
distribution.
*/

Found in path(s):
* /opt/cola/permits/1222902257_1637000982.24/0/tinyxml-2-6-2-zip/tinyxml/tinyxml.cpp
*
/opt/cola/permits/1222902257_1637000982.24/0/tinyxml-2-6-2-zip/tinyxml/tinyxml.h