Bug 93027 - ICE: in match_reload, at lra-constraints.c:1060

Status:	ASSIGNED	**Reported:**	2019-12-20 13:00 UTC by 吳建興
Alias:	None	**Modified:**	2021-12-03 03:36 UTC (History)
Product:	gcc	**CC List:**	3 users (show)
Component:	rtl-optimization (show other bugs)	**See Also:**	
Version:	10.0	**Host:**	
Importance:	P3 normal	**Target:**	
Target Milestone:	---	**Build:**	
Assignee:	Vladimir Makarov	**Known to work:**	
URL:		**Known to fail:**	
Keywords:	ice-on-valid-code, inline-asm, ra	**Last reconfirmed:**	2020-01-21 00:00:00
Depends on:			
Blocks:			

Attachments

Add an attachment (proposed patch, testcase, etc.)

Note

You need to log in before you can comment on or make changes to this bug.

吳建興 2019-12-20 13:00:50 UTC — Description

```
g++ ./hi.c

INPUT:

int main() {
  int f= 0, w;

  asm volatile(
    ""
    : "+m&l"(f)
    : "0a"(&w)
  );
}

OUTPUT:

during RTL pass: reload
./bug.cc: In function ‘int main()’:
./bug.cc:9:1: internal compiler error: in match_reload, at lra-constraints.c:1060
    9 | }
      | ^
0x2b55f43 match_reload
        ../../../gcc/lra-constraints.c:1058
0x2b9c03f curr_insn_transform
        ../../../gcc/lra-constraints.c:4306
0x2ba4656 lra_constraints(bool)
        ../../../gcc/lra-constraints.c:5009
0x2b102b4 lra(_IO_FILE*)
        ../../../gcc/lra.c:2437
0x28e6954 do_reload
        ../../../gcc/ira.c:5518
0x28e6954 execute
        ../../../gcc/ira.c:5704

It is probably the same bug in https://gcc.gnu.org/bugzilla/show_bug.cgi?id=84985
```

Vladimir Makarov 2020-01-08 21:35:33 UTC — Comment 1

```
Thank you for the report.  I've started working on it.  As changes in constraint
processing needs a lot of testing, I think the patch will be read on Friday or on
the next week.
```

Vladimir Makarov 2020-01-10 20:37:55 UTC — Comment 2

```
Sorry, I did a mistake in PR number and automatic commits reporting did not work.

Here are the patches fixing the PR:

https://gcc.gnu.org/viewcvs/gcc?view=revision&revision=280133
https://gcc.gnu.org/viewcvs/gcc?view=revision&revision=280135
https://gcc.gnu.org/viewcvs/gcc?view=revision&revision=280136
```

Vladimir Makarov 2020-01-10 20:45:58 UTC — Comment 3

```
Author: vmakarov
Date: Fri Jan 10 20:45:19 2020
New Revision: 280138

URL: https://gcc.gnu.org/viewcvs?rev=280138&root=gcc&view=rev
Log:
2020-01-10  Vladimir Makarov  <vmakarov@redhat.com>

        PR inline-asm/93027
        * gcc.target/i386/pr93027.c: Use the right PR number in the test.


Modified:
    trunk/gcc/testsuite/ChangeLog
    trunk/gcc/testsuite/gcc.target/i386/pr93027.c
```

CVS Commits 2020-01-23 21:52:33 UTC — Comment 4

```
The master branch has been updated by Jakub Jelinek <jakub@gcc.gnu.org>:

https://gcc.gnu.org/g:3a26c7b3a3569a5e1b6f0342e50aefe76ad0ec4d

commit r10-6190-g3a26c7b3a3569a5e1b6f0342e50aefe76ad0ec4d
Author: Jakub Jelinek <jakub@redhat.com>
Date:   Thu Jan 23 22:50:40 2020 +0100

    testsuite: Require lp64 target rather x86_64-*-* in pr93027.c. [PR93027]

    I've noticed this test failed on x86_64-linux with -m32 or -mx32 testing,
    the triplet doesn't really say which actual multilib it is, and the test
    really works with lp64.

    2020-01-23  Jakub Jelinek  <jakub@redhat.com>

        PR inline-asm/93027
        * gcc.target/i386/pr93027.c: Require lp64 target rather x86_64-*-*.
```

Jakub Jelinek 2020-02-13 12:19:18 UTC Comment 5

So fixed for trunk? GCC 9 seems to ICE on this too in the same spot (but with -fchecking only or --enable-checking=yes), GCC 8 in extract_constraint_insn (but again with checking only).

Alexandre Oliva 2021-12-03 03:36:12 UTC Comment 6

FWIW, with or without optimization, this testcase allocates inout f and in &w to the same register in the .reload dump. If the asm code used %0, expecting it to be initially 0 as it should, it would fail:

```
int main (void) {
  int f = 0, w, z;

  asm volatile(
    "movl\t%0, %1"
    : "+m&l"(f), "=g" (z)
    : "0a"(&w)
  );

  if (z)
    __builtin_abort ();

  return 0;
}
```