

MicroTESK - Bug #6287

Variable dependencies causes false positives in path feasibility test

09/22/2015 01:41 PM - Artem Kotsynyak

Status:	Closed	Start date:	09/22/2015
Priority:	Normal	Due date:	
Assignee:	Alexander Kamkin	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:	2.3	Spent time:	0.00 hour
Detected in build:	svn	Published in build:	20150925
Platform:			

Description

For attached code six (6) code paths are produced:

- 1: L1.Event=MISS, L2.Event=MISS, pmem.condition=0, pmem.dependent!=1, pmem.STOP
- 2: L1.Event=MISS, L2.Event=MISS, pmem.condition=0, pmem.dependent=1, pmem.STOP
- 3: L1.Event=MISS, L2.Event=MISS, pmem.condition!=0, pmem.dependent!=1, pmem.STOP
- 4: L1.Event=MISS, L2.Event=MISS, pmem.condition!=0, pmem.dependent=1, pmem.STOP
- 5: L1.Event=MISS, L2.Event=HIT, pmem.STOP
- 6: L1.Event=HIT, pmem.STOP

None of them being pruned with isFeasible() check while paths 2 and 3 are infeasible.

```
read = {
  l1Tag = va.vaddress<47..12>;
  l2Tag = va.vaddress<47..17>;

  if L1(va).hit then
    l1Entry = L1(va);
    line = l1Entry.DATA;
  else
    if L2(va).hit then
      l2Entry = L2(va);
      line = l2Entry.DATA;

      // Fill L1.
      l1Entry.TAG = l1Tag;
      l1Entry.DATA = line;
      L1(va) = l1Entry;
    else
      pa.addrdesc.paddress.physicaladdress<1..0> = 0b00;
      tempValue = MEM(pa);
      line<(64 * 0 + 63)..(64 * 0)> = tempValue;
      pa.addrdesc.paddress.physicaladdress<1..0> = 0b01;
      tempValue = MEM(pa);
      line<(64 * 1 + 63)..(64 * 1)> = tempValue;
      pa.addrdesc.paddress.physicaladdress<1..0> = 0b10;
      tempValue = MEM(pa);
      line<(64 * 2 + 63)..(64 * 2)> = tempValue;
      pa.addrdesc.paddress.physicaladdress<1..0> = 0b11;
      tempValue = MEM(pa);
      line<(64 * 3 + 63)..(64 * 3)> = tempValue;

      // Fill L2.
      l2Entry.TAG = l2Tag;
      l2Entry.DATA = line;
      L2(va) = l2Entry;

      // Fill L1.
      l1Entry.TAG = l1Tag;
      l1Entry.DATA = line;
      L1(va) = l1Entry;
```

```

    if condition != 0 then
        dependent = 1;
    else
        dependent = 0;
    endif;

    if dependent == 1 then
        temp.value = 0;
    endif;
endif;
endif;

data = line<(64 * va.vaddress<4..3> + 63)..(64 * va.vaddress<4..3>)>;
}

```

History

#1 - 09/22/2015 02:29 PM - Alexander Kamkin

- Assignee changed from Andrei Tatarnikov to Alexander Kamkin
- Target version set to 2.3

#2 - 09/22/2015 02:41 PM - Alexander Kamkin

- Description updated

#3 - 09/22/2015 02:47 PM - Artem Kotsynyak

- File *mmu.debug* added

#4 - 09/22/2015 02:52 PM - Alexander Kamkin

```

[pmem.dependent == 0:1]
[pmem.dependent == 1]

```

It seems that the solver does not care about variables with fixed values ([pmem.dependent = 0:1] is interpreted as [pmem.dependent = x], where x is some variable).
To be fixed soon.

#5 - 09/22/2015 04:25 PM - Artem Kotsynyak

- Status changed from *New* to *Resolved*
- % Done changed from 0 to 100

#6 - 09/29/2015 11:03 AM - Alexander Kamkin

- Status changed from *Resolved* to *Closed*
- Published in build set to 20150925

Files

mmu.debug	5.68 KB	09/22/2015	Artem Kotsynyak
-----------	---------	------------	-----------------