



Andrey Tatarnikov wrote:

Логика выражений типа  $M[0] :: M[1] = R[4]$  некорректна. Она должна означать:

$M[0] = R[4];$   
 $M[1] = R[4]$

Запись  $x1::x2 = value$  нужно интерпретировать следующим образом. Если суммарная разрядность  $x1$  и  $x2$  равна разрядности  $value$ . Часть значения записывается в  $x1$ , а часть - в  $x2$ . Если разрядности не совпадают, нужно начинать с младших бит, а дальше - либо обрезать, либо заполнять нулями.

Все конструкции с недоопределенной семантикой нужно документировать в руководстве по языку (нужно в wiki завести соответствующий раздел).

### #3 - 03/17/2013 09:30 PM - Andrei Tatarnikov

Сейчас так и делается, как ты написал. В модели ARM есть вот такие конструкции:

```
carry::ALU_OUT = GPR[src2] + src3;
```

Нужно подумать как их правильно интерпретировать. Левая часть заведомо больше на carry (некий флаг).

### #4 - 04/03/2014 07:57 AM - Alexander Kamkin

- Due date set to 04/30/2014

- Target version set to 2.0

Что с этой задачей?

### #5 - 11/12/2014 06:13 PM - Andrei Tatarnikov

Сейчас для присваивания используется BitVector.assign. Он копирует все данные. Если остаётся место - заполняет его нулями, если что-то не вмещается - данные обрезаются.

Вообще этот код нуждается в Review и я над ним сейчас работаю.

### #6 - 11/12/2014 06:20 PM - Andrei Tatarnikov

Тут же ещё и sign extension возможен.

### #7 - 11/13/2014 12:49 PM - Alexander Kamkin

- Target version changed from 2.0 to 2.1

### #8 - 03/12/2015 08:52 AM - Alexander Kamkin

- Subject changed from [translator] Concatenation in left hand side expressions to Concatenation in left hand side expressions

- Category set to 40

### #9 - 03/25/2015 06:31 AM - Alexander Kamkin

- Target version changed from 2.1 to 2.2

### #10 - 06/03/2015 09:15 AM - Alexander Kamkin

Has the issue been resolved?

### #11 - 04/19/2016 12:52 PM - Andrei Tatarnikov

- Status changed from New to Resolved

- % Done changed from 0 to 100

In the current implementation, both sides of an assignment expression must have the same size. Otherwise, it will cause a type error.

So, in  $M[0]:M[1]=R[4]$ , data one half of  $R[4]$  is copied into  $M[0]$  and another is copied into  $M[1]$ . If sizes of left and right sides mismatch, a translation error will occur.

### #12 - 04/19/2016 01:50 PM - Andrei Tatarnikov

- Status changed from Resolved to Closed

- Published in build set to 2.3.31