

DSP-ядро ELcore-30M

ПЕРЕЧЕНЬ ВЫЯВЛЕННЫХ ОГРАНИЧЕНИЙ

ОГЛАВЛЕНИЕ

1. О документе	1-3
2. Выявленные ограничения	2-4
2.1 Прерывание от таймера блокируется командами <code>move reg, pc</code> и <code>move reg, ss</code>	2-4
2.1.1 Описание	2-4
2.1.2 Решение	2-4
2.2 Пропадание прерываний на <code>move reg, pc</code> и <code>move reg, ss</code>	2-4
2.2.1 Описание	2-4
2.2.2 Решение	2-4
2.3 Некорректная работа команд <code>DCR</code> , <code>DCRb</code>	2-5
2.3.1 Описание	2-5
2.3.2 Решение	2-5
2.4 Не срабатывает точка останова на первой команде программы	2-5
2.4.1 Описание	2-5
2.4.2 Решение	2-5
2.5 Автоматическая блокировка конвейера не отслеживает зависимостей по данным, хранящимся в регистрах-аккумуляторах	2-5
2.5.1 Описание	2-5
2.5.2 Решение	2-5
2.6 Некорректное чтение из XYRAM по адресу $(An+\#const)$ сразу после записи в In ($n=0,1,\dots,7$) (<code>rf#8063</code>)	2-5
2.6.1 Описание	2-5
2.6.2 Решение	2-6
3. Перечень изменений	3-7
12.02.2018	3-7

1.0 ДОКУМЕНТЕ

Данный документ описывает известные ограничения в работе DSP-ядра ELcore-30M.

2.ВЫЯВЛЕННЫЕ ОГРАНИЧЕНИЯ

2.1 Прерывание от таймера блокируется командами `move reg, pc` и `move reg, ss`

2.1.1 Описание

Прерывание от таймера может не выполняться при наличии в коде команд `move reg, pc` и `move reg, ss`.

2.1.2 Решение

При наличии разрешенных прерываний DSP-ядра необходимо непосредственно перед пересылками `move reg, pc` и `move reg, ss` добавить команду `nop.cc`.

2.2 Пропадание прерываний на `move reg, pc` и `move reg, ss`

2.2.1 Описание

```
01: bd 0x4
02: move (a7+0x8), r0.l
03: nop
04: nop
05: move r0.l, (a7+0xb)
06: move (a7+0xb), r0.l
```

`r0.l` в результате `06` ожидается равным выгруженному из памяти в `bd`-слоте `02`,

`05` загружает правильное значение (которое выгрузилось в `02`), а `06` выгружает предыдущее (`02`) значение регистра `r0.l`.

2.2.2 Решение

1. Не использовать в слоте переходов `bd, jd` команд чтения данных из памяти;
2. Не использовать в слоте переходов `bd, jd` команд чтения данных из памяти с предмодификацией адреса, и при этом, за счет соответствующего размещения данных в памяти, гарантировать отсутствие блокировок арбитража при обращениях к памяти;
3. Непосредственно после переходов `bd, jd`, в слоте которых выполнялась команда чтения данных из памяти в регистр, содержимое этого регистра не должно использоваться в командах пересылок - в течение 2 тактов и в вычислительных командах - в течение 1 такта.

2.3 Некорректная работа команд DCR, DCRb

2.3.1 Описание

Команды DCR, DCRb выполняются с ошибкой

2.3.2 Решение

Для реализации декодера Витерби/Турбо рекомендуется использовать команду CS2 и другие возможности системы инструкций.

2.4 Не срабатывает точка останова на первой команде программы

2.4.1 Описание

Не срабатывает точка останова на первой команде программы (при любом значении начального адреса программы)

2.4.2 Решение

Не устанавливать точку останова на первой команде программы.

2.5 Автоматическая блокировка конвейера не отслеживает зависимости по данным, хранящимся в регистрах-аккумуляторах

2.5.1 Описание

После выполнения вычислительной команды с участием аккумулятора правильный результат можно считать из регистра-аккумулятора не на следующем такте, а лишь спустя один такт.

2.5.2 Решение

После выполнения вычислительной команды с участием аккумулятора результат команды следует считывать из регистра-аккумулятора не на следующем такте, а спустя, по крайней мере, один такт.

2.6 Некорректное чтение из XYRAM по адресу (An+#const) сразу после записи в In (n=0,1,...,7) (rf#8063)

2.6.1 Описание

У пользователя неверно сработал код:

```
add #2,R10,R0
cvfi R2,R2      move R0.s,I0
move (A0+1),R12.1 ;читает ноль ??????????????????????
inc R10,R10     move (A0+I0),R0.1
```

а именно, прочитались неверные данные при обращении по адресу (A0+1).

При анализе обнаружилась схемотехническая ошибка – недостаточно детально описан сигнал BLA_WR в блоке AGU. Это приводит к тому, что если сразу после записи в регистр In ($n=0,1,\dots,7$) идет команда обращения к памяти по адресу An с непосредственным смещением ($An+\#const$), причем адресный регистр An относится к тому адресному триплету (т.е. у An и In один и тот же номер n), то этой командой формируется неправильный адрес обращения к памяти.

2.6.2 Решение

Последовательность команд:

```
move Ri.s, In  
move (An+#const), Rj.l
```

где $i,j=0,1,\dots,31$ – номер регистра RF; $n=0,1,\dots,7$ – номер адресного регистра.

следует считать запрещенной.

В рассматриваемом случае достаточно поменять указанные две команды местами, и код сработает правильно.

3.ПЕРЕЧЕНЬ ИЗМЕНЕНИЙ

12.02.2018

Добавлено граничение «2.6 Некорректное чтение из XYRAM по адресу ($A_n + \#const$) сразу после записи в In ($n=0,1,\dots,7$) (rf#8063)».