

19 ОТЛАДКА ПРОЦЕССОРА BLACKFIN

Функции отладки процессора Blackfin используются для отладки программ. Они также дополняют некоторые службы, характерные для ядра операционной системы. Функции отладки реализованы в процессоре аппаратно и сгруппированы в несколько уровней.

В таблице 19-1 перечислены доступные функции отладки.

Таблица 19-1. Функции отладки процессора Blackfin

Функция отладки	Описание
Точки останова	Определяют диапазон адресов и условия, по совпадению которых выполняется останов процессора
История трассировки	Последние 16 значений счётчика команд до переходов сохраняются во внутреннем буфере трассировки
Счётчик тактов	Обеспечивает поддержку любых функций профилирования (profiling)
Мониторинг выполнения	Позволяет выполнять мониторинг и измерение загрузки внутренних ресурсов процессора без вмешательства в их работу

Блок точек останова

Блок точек останова обеспечивает несколько механизмов исследования поведения программы, выполняя мониторинг адресов на шинах данных и команд. После нескольких совпадений с заданным адресом блок генерирует событие.

Информация, обеспечиваемая блоком точек останова, помогает оптимизировать код. При помощи этого блока также упрощается процедура корректирования кода (patching) исполняемых файлов.

Блок точек останова содержит следующие регистры, отображённые в карте памяти, которые доступны в режимах Супервизора и Эмуляции:

- регистр состояния точек останова (WPSTAT),
- шесть регистров адресов точек останова по командам (WPIA[5:0]),
- шесть регистров счётчиков точек останова по командам (WPIACNT[5:0]),
- регистр управления точками останова по командам (WPIACTL),
- два регистра адресов точек останова по данным (WPDA[1:0]),
- два регистра счётчиков точек останова по данным (WPDACNT[1:0]),
- регистр управления точками останова по данным (WPDACL).

19 Отладка процессора Blackfin

Точки останова по командам реализуются при помощи двух операций:

- Значения, содержащиеся в шести регистрах адресов точек останова по командам ($WPIA[5:0]$), сравниваются с адресами на шине команд.
- При каждом совпадении декрементируются соответствующие значения счётчиков в регистрах счётчиков точек останова по командам ($WPIACNT[5:0]$).

Шесть регистров адресов точек останова по командам могут группироваться, формируя три точки останова по диапазонам адресов команд. Эти диапазоны задаются значениями пар регистров $WPIA0$ и $WPIA1$, $WPIA2$ и $WPIA3$, $WPIA4$ и $WPIA5$.



Значения границ диапазонов, задаваемых регистрами $WPIA0$, $WPIA1$, $WPIA2$, $WPIA3$, $WPIA4$ и $WPIA5$, должны удовлетворять следующим условиям:

$$WPIA0 \leq WPIA1$$

$$WPIA2 \leq WPIA3$$

$$WPIA4 \leq WPIA5$$

Точки останова по данным реализуются при помощи двух операций:

- Значения, содержащиеся в двух регистрах адресов точек останова по данным ($WPDA[1:0]$), сравниваются с адресами на шине данных.
- При каждом совпадении декрементируются соответствующие значения счётчиков в регистрах счётчиков точек останова по данным ($WPDACNT[1:0]$).

Два регистра адресов точек останова по данным могут быть сгруппированы в одну точку останова по диапазону адресов данных, $WPDA[1:0]$.

В регистры счётчиков данных и команд должно загружаться значение, на единицу меньшее количества совпадений точки останова. После того, как значение счётчика достигает нуля, следующее совпадение вызывает исключение или эмуляцию.



Следует отметить, что после вызова события необходимо заново выполнить инициализацию счётчика.

События также могут вызываться комбинацией точек останова по данным и командам. Если в регистре $WPIACTL$ установлен бит $WPAND$, событие вызывается только при совпадении и точки останова по адресу команды, и точки останова по адресу данных. Если бит $WPAND$ равен нулю, событие вызывается при совпадении любой из разрешённых точек останова по адресу или диапазону.

Для активизации блока точек останова необходимо установить бит $WPPWR$ в регистре $WPIACTL$. Если $WPPWR = 1$, возможна активизация отдельных точек останова по адресам или диапазонам адресов при помощи битов активизации в регистрах $WPIACTL$ и $WPDACTL$. Если $WPPWR = 0$, все точки останова неактивны.

19 Отладка процессора Blackfin

Точки останова по командам

Как показано в таблице 19-2, каждая точка останова по командам управляется тремя битами регистра WPIACTL.

Таблица 19-2. Биты управления регистра WPIACTL

Название бита	Описание
EMUSW _x	Определяет, тип события, вызываемого совпадением адреса команды (эмуляция или исключение)
WPICNTEN _x	Активизирует 16-разрядный счётчик, считающий число совпадений адреса. Если счётчик отключён, событие вызывается по каждому совпадению.
WPIAEN _x	Активизирует функцию точки останова по адресу

Когда две точки останова формируют диапазон адресов, используются два дополнительных бита, указанные в таблице 19-3.

Таблица 19-3. Биты управления точками останова по диапазону адресов в регистре WPIACTL

Название бита	Описание
WPIREN _{xy}	Указывает две точки останова, формирующие диапазон адресов.
WPIRINV _{xy}	Определяет, вызывается ли событие при попадании адреса в заданный диапазон или за его пределы.

Функция корректировки кода позволяет заменять секции существующего кода новым кодом. Регистры точек останова используются для вызова исключения по начальным адресам предыдущей версии кода. После этого выполняется переход по вектору программы обработки исключения к ячейке памяти, содержащей новый код.

Функция корректировки кода в процессоре может реализовываться записью начального адреса предыдущей версии кода в один из регистров WPIAn и настройкой соответствующего бита EMUSW_x на вызов исключения. В программе обслуживания исключения выполняется чтение регистра WPSTAT для определения точки останова, вызвавшей исключение. Затем в регистр RETX записывается начальный адрес нового кода, и при выходе из программы обслуживания исключения начинает выполняться новый код. Так как для корректировки кода используется механизм исключений, невозможно исправление программ обслуживания событий одинакового или более высокого приоритета (исключения, NMI и сброса).

При записи в регистр WPSTAT, отображённый в карте памяти, сбрасываются все залипающие биты состояния. Записываемое значение игнорируется.

Регистры WPIAn

Когда блок точек останова активизирован, значения, содержащиеся в регистрах адресов точек останова по командам (WPIAn), сравниваются с адресом на шине

19 Отладка процессора Blackfin

команд. При каждом совпадении декрементируется значение счётчика в соответствующем регистре счётчика точки останова по командам (WPIACNTn).

Регистры точек останова по командам, WPIA [5:0], показаны на рис. 19-1.



Рис. 19-1. Регистры точек останова по командам

Таблица 19-4. Адреса регистров точек останова по командам в карте памяти

Название регистра	Адрес в карте памяти
WPIA0	0xFFE0 7040
WPIA1	0xFFE0 7044
WPIA2	0xFFE0 7048
WPIA3	0xFFE0 704C
WPIA4	0xFFE0 7050
WPIA5	0xFFE0 7054

Регистры WPIACNTn

Когда блок точек останова активизирован, значения счётчиков в регистрах счётчиков точек останова по командам (WPIACNT [5:0]) декрементируются при каждом совпадении адреса на шине адреса со значением регистра WPIAn. В регистр WPIACNTn загружается значение, на единицу меньшее числа совпадений точек останова, после которого вызывается событие (см. рис. 19-2).



Рис. 19-2. Регистры счётчиков точек останова по командам

19 Отладка процессора Blackfin

Таблица. 19-5. Адреса регистров счётчиков точек останова по командам в карте памяти

Название регистра	Адрес в карте памяти
WPIACNT0	0xFFE0 7080
WPIACNT1	0xFFE0 7084
WPIACNT2	0xFFE0 7088
WPIACNT3	0xFFE0 708C
WPIACNT4	0xFFE0 7090
WPIACNT5	0xFFE0 7094

Регистр WPIACTL

Каждая точка останова по командам управляется тремя битами в регистре управления точками останова по командам (WPIACTL). На рис. 19-3 и 19-4 показаны старшая и младшая половины этого регистра, соответственно. Дополнительную информацию о битах этого регистра см. в разделе “Точки останова по командам”.

i Состояние битов регистра WPIACTL не имеет значения, пока не установлен бит WPPWR.

Регистр управления точками останова по командам (WPIACTL)

При сравнении с диапазоном адресов IA - адрес команды

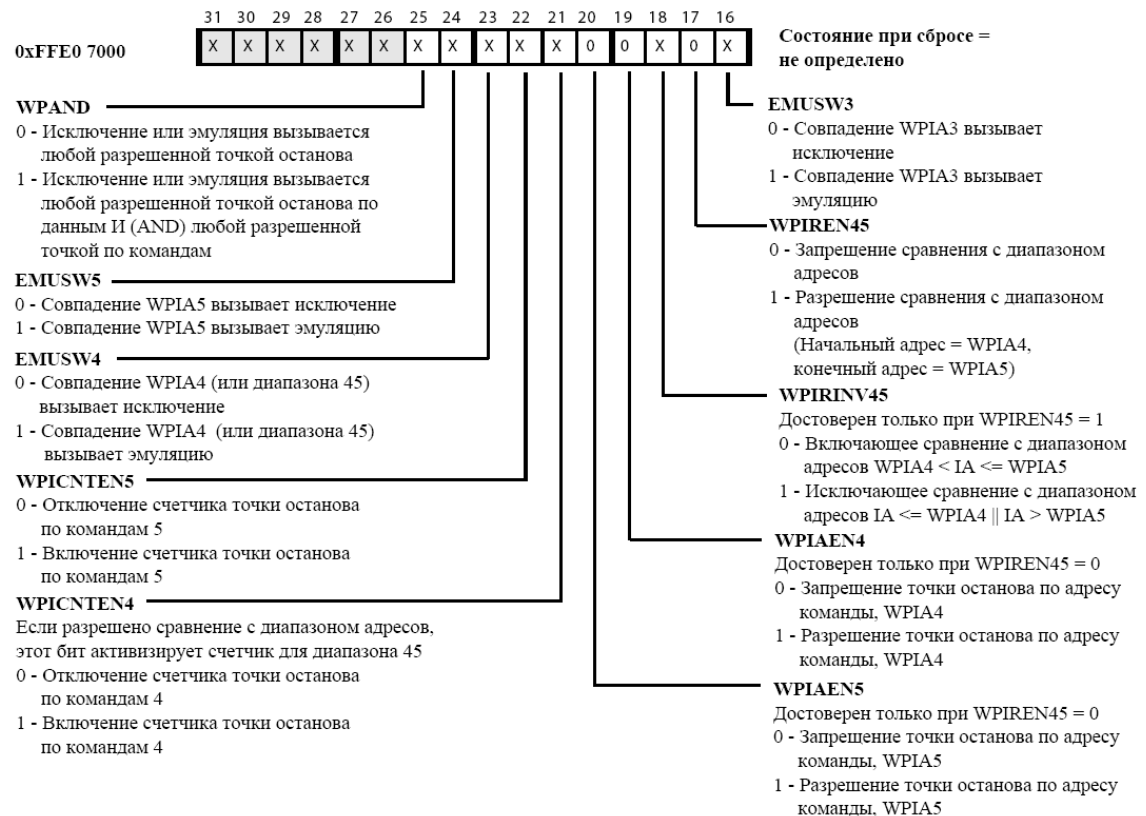


Рис. 19-3. Регистр управления точками останова по командам (WPIACTL) [31:16]

19 Отладка процессора Blackfin

Регистр управления точками останова по командам (WPIACTL)

При сравнении с диапазоном адресов IA - адрес команды

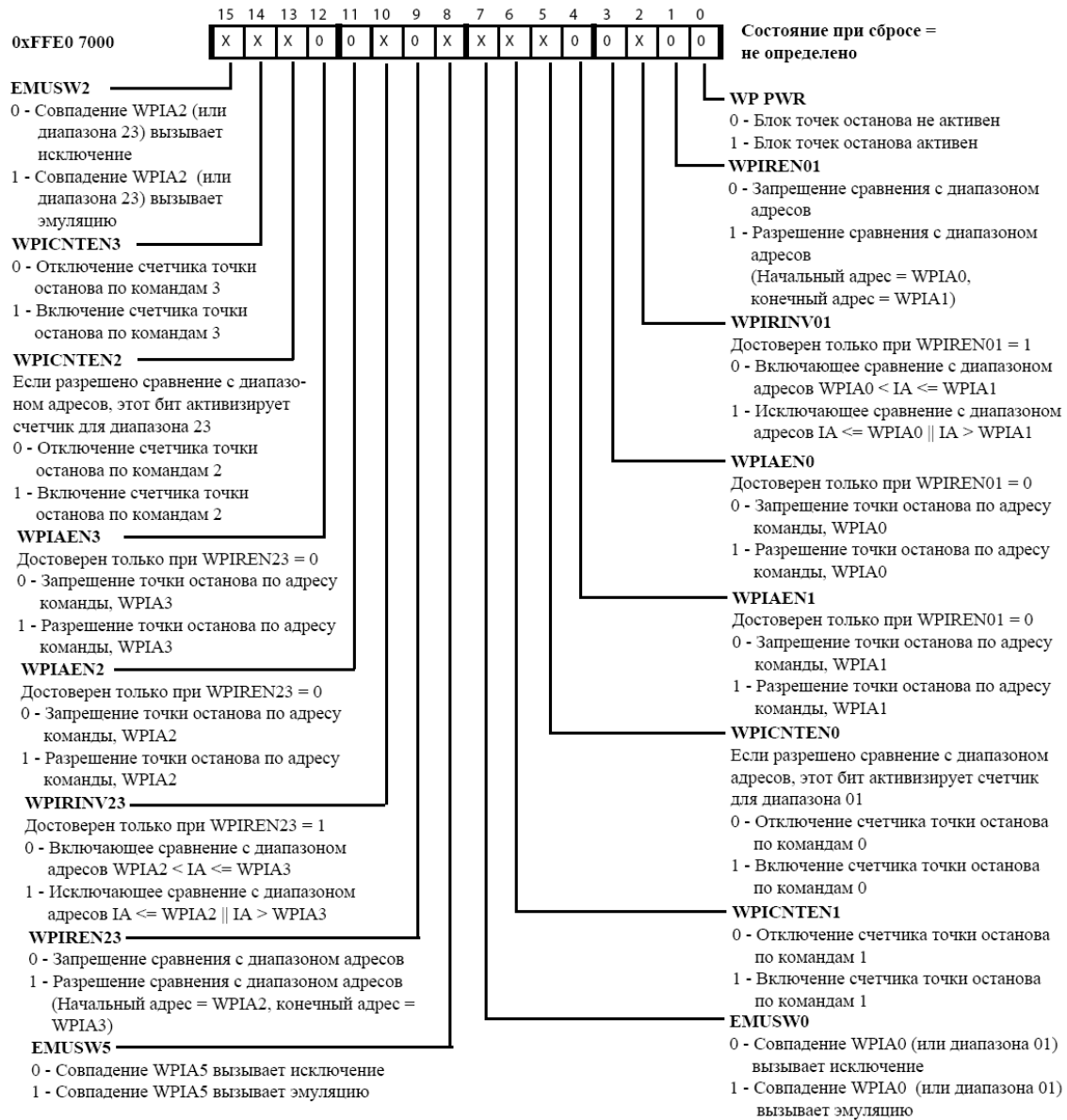


Рис. 19-4. Регистр управления точками останова по командам (WPIACTL) [15:0]

Точки останова по адресам данных

Как показано в таблице 19-6, каждая точка останова по данным управляется четырьмя битами регистра WPDCTL.

19 Отладка процессора Blackfin

Таблица 19-6. Точки останова по адресам данных

Название бита	Описание
WPDACCn	Определяет тип обращения (чтение или запись), по которому фиксируется совпадение.
WPDSRCn	Определяет DAG, мониторинг которого выполняет блок точек останова.
WPDACNTENn	Активизирует 16-разрядный счётчик, считающий число совпадений адреса. Если счётчик отключён, событие вызывается по каждому совпадению.
WPDAENn	Активизирует функцию точки останова по данным.

Когда две точки останова задают диапазон адресов, используются два дополнительных бита (см. таблицу 19-7).

Таблица 19-7. Биты управления точками останова в регистре WPDACCTL

Название бита	Описание
WPDREN01	Указывает две точки останова, формирующие диапазон адресов.
WPDRAINV01	Определяет, вызывается ли событие при попадании адреса в заданный диапазон или за его пределы.



Следует отметить, что точки останова по адресам данных всегда вызывают исключения.

Регистры WPDA_n

Когда блок точек останова активизирован, значения, содержащиеся в регистрах адресов точек останова по данным (WPDA_n), сравниваются с адресом на шине данных. При каждом совпадении декрементируется значение счётчика в соответствующем регистре счётчика точки останова по данным (WPDACNT_n).

Регистры адресов точек останова по данным, WPDA [1 : 0], показаны на рис. 19-5.

Регистр адресов точек останова по данным (WPDA_n)

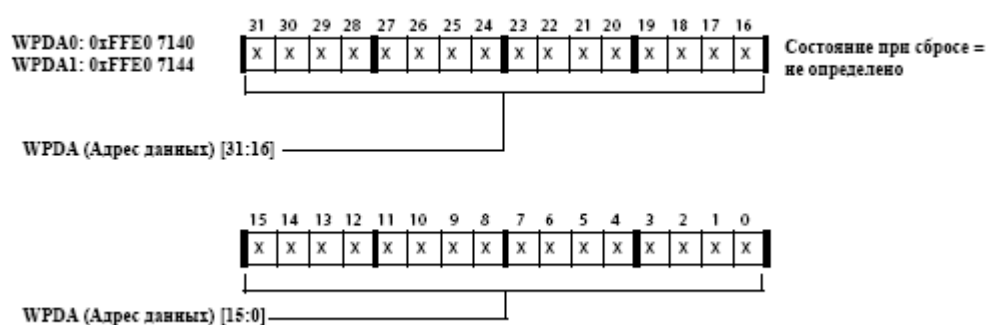


Рис. 19-5. Регистры адресов точек останова по данным

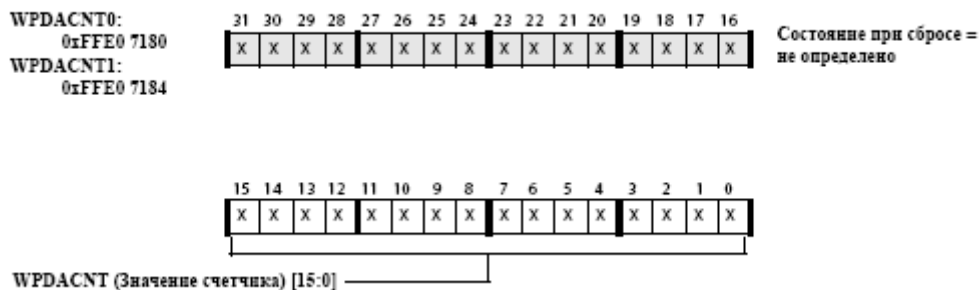
Регистры WPDACNT_n

Когда блок точек останова активизирован, значения счётчиков в регистрах счётчиков точек останова по данным (WPDACNT_n) декрементируются при каждом

19 Отладка процессора Blackfin

совпадении адреса на шине адреса со значением регистра $WPDAn$. В регистр $WPDACNTn$ загружается значение, на единицу меньшее числа совпадений точек останова, после которого вызывается событие.. Регистры счётчиков точек

Регистры счётчиков точек останова по данным ($WPDACNTn$)



останова по данным, $WPDACNT [1:0]$, показаны на рис. 19-6.

Рис. 19-6. Регистры счётчиков точек останова по данным

Регистр $WPDACTL$

Дополнительную информацию о битах регистра управления точками останова по данным ($WPDACTL$) см. в разделе “Точки останова по адресам данных”.

19 Отладка процессора Blackfin

Регистр управления точками останова по данным (WPDACTL)

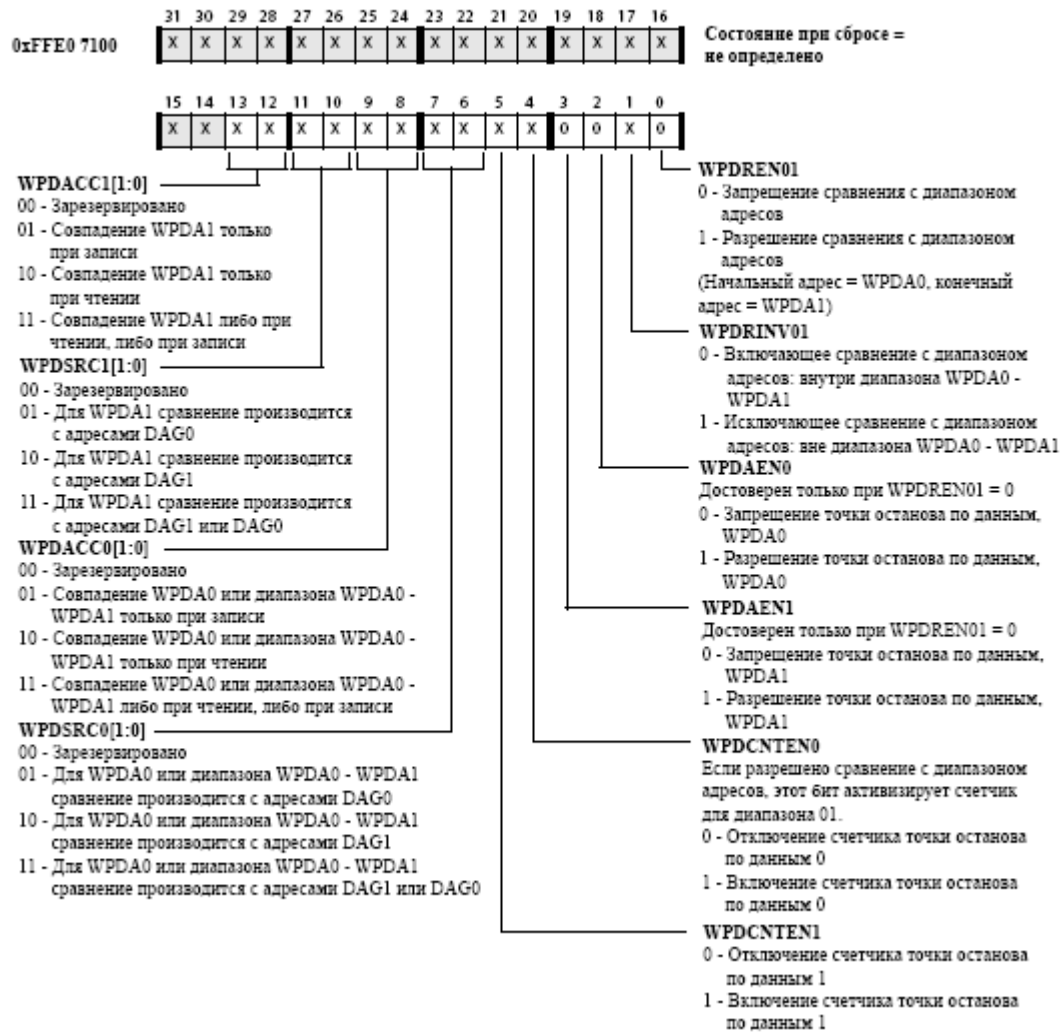


Рис. 19-7. Регистр управления точками останова по данным

Регистр WPSTAT

В регистре состояния точек останова (WPSTAT) указывается состояние точек останова. Чтение и запись этого регистра возможны только в режимах Супервизора и Эмуляции. Когда происходит совпадение значения точки останова или точки останова по диапазону, в этом регистре устанавливается соответствующий бит. Биты состояния в регистре WPSTAT являются залипающими. Они сбрасываются при записи любого значения в регистр.

Регистр состояния точек останова показан на рис. 19-8.

19 Отладка процессора Blackfin

Регистр состояния точек останова (WPSTAT)

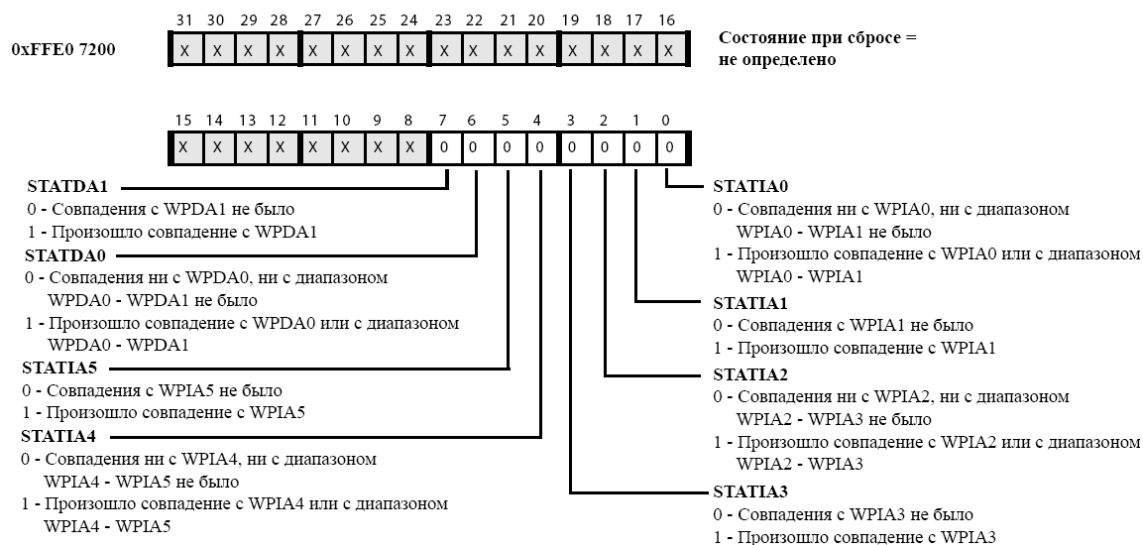


Рис. 19-8. Регистр состояния точек останова

Блок трассировки

Блок трассировки сохраняет историю последних 16 изменений программного потока, выполненных программным автоматом. Это позволяет пользователю воссоздать процесс работы программного автомата.

Возможно разрешение генерации исключения по заполнению буфера трассировки. В соответствующей программе обслуживания исключения выполняется сохранение элементов буфера трассировки в память. Таким образом, возможно воссоздание полного пути программного автомата с момента активизации блока трассировки.

Изменения программного потока, вызванные циклами с нулевыми непроизводительными затратами, не сохраняются в буфер трассировки. При отладке кода, “зависающего” при выполнении цикла с нулевыми непроизводительными затратами, можно отслеживать значения регистров счётчиков циклов, LC0 и LC1.



Буфер трассировки можно сконфигурировать на пропуск записи изменений программного потока, совпадающих с последним или одним из двух последних элементов, содержащихся в нём. Пропуск записи значений, совпадающих с этими элементами, предотвращает переполнение буфера, вызываемое выполнением циклов в программе. Так как изменения, вызываемые циклами с нулевыми непроизводительными затратами, не сохраняются в буфер трассировки, эта опция может использоваться для предотвращения переполнения, возникающего при выполнении до четырёх вложенных циклов.

При чтении регистра буфера трассировки (TBUF) возвращается значение вершины стека блока трассировки, содержащего 16 элементов. Каждый элемент содержит

19 Отладка процессора Blackfin

пару адресов: адрес команды перехода (branch source) и адрес перехода (адрес, по которому производится переход, branch target). При чтении регистра TBUF сначала возвращается последний сохранённый элемент, начиная с адреса перехода. При следующем чтении возвращается адрес команды перехода.

Значение количества достоверных элементов в TBUF содержится в поле TBUFCNT регистра TBUFSTAT. Значение поля TBUFCNT инкрементируется при каждом втором чтении TBUF. Так как каждый элемент состоит из двух слов данных, регистр TBUF опустошается после выполнения $2 \times TBUFCNT$ операций чтения.

-  Изменения последовательности выполнения программы, совпадающие с любым из двух последних элементов буфера трассировки, не записываются.
-  Так как операция чтения буфера трассировки является деструктивной, рекомендуется выполнять её в части программы, в которой запрещены прерывания.

Следует отметить, что, если выполняется одноуровневая компрессия, устанавливается младший бит адреса перехода. Если выполняется двухуровневая компрессия, устанавливается младший бит адреса команды перехода.

Регистр TBUFCTL

Работа блока трассировки разрешается двумя битами управления в регистре управления буфером трассировки (TBUFCTL). Буфер трассировки должен активизироваться установкой бита TBUFPWR. Если TBUFPWR = 1, установка бита TBUFEN в единицу разрешает работу блока трассировки.

Регистр управления буфером трассировки (TBUFCTL) показан на рис. 19-9. Если TBUFOVF = 1, блок трассировки не производит запись изменений процесса выполнения программы во время обслуживания исключений, NMI и сброса.

19 Отладка процессора Blackfin

Регистр управления буфером трассировки (TBUFCTL)

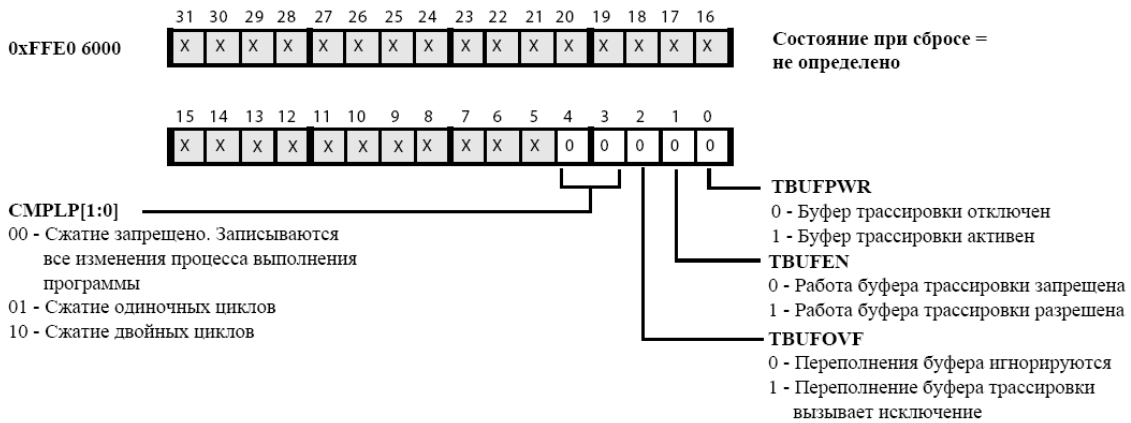


Рис. 19-9. Регистр управления буфером трассировки

Регистр TBUFSTAT

На рис. 19-10 показан регистр состояния буфера трассировки (TBUFSTAT). Значение TBUFCNT декрементируется на единицу при каждом втором чтении регистра TBUF.

Регистр состояния буфера трассировки (TBUFSTAT)

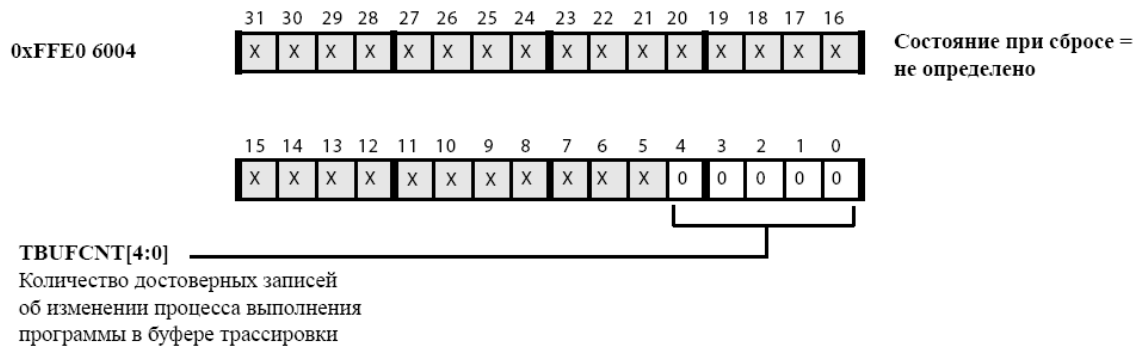


Рис. 19-10. Регистр состояния буфера трассировки

Регистр TBUF

На рис. 19-11 показан регистр буфера трассировки (TBUF). При первом чтении регистра TBUF возвращается последний адрес перехода. При втором чтении возвращается последний адрес команды перехода.

Регистр буфера трассировки (TBUF)

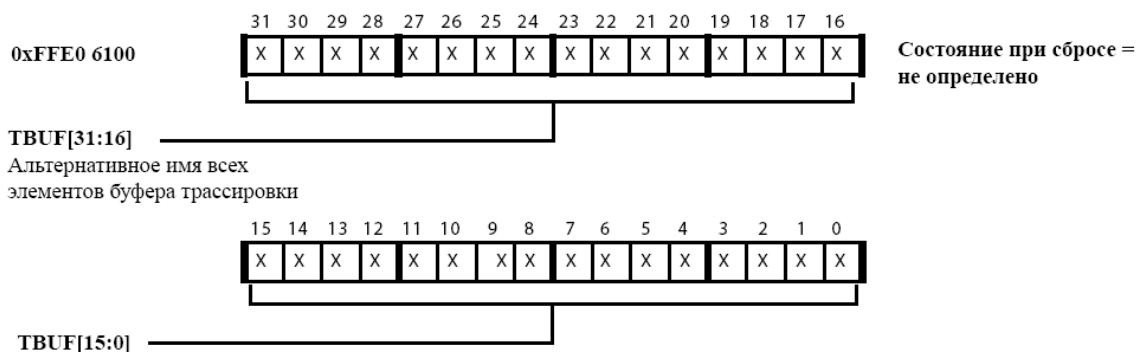


Рис. 19-11. Регистр буфера трассировки

Блок трассировки не производит запись изменений процесса выполнения программы:

- в режиме эмуляции,
- в программах обслуживания исключений или прерываний более высокого приоритета (если TBUFOVF = 1).

Изменения процесса выполнения программы могут быть считаны из регистра TBUF в программе обслуживания исключения и сохранены при помощи кода, приведенного в листинге 19-1.

i На время чтения регистра TBUF следует запретить запись в буфер трассировки новых изменений процесса выполнения программы.

Программа, воссоздающая в памяти след выполнения программы

В листинге 19-1 приведен код, воссоздающий в памяти полный след выполнения программы (execution trace).

Листинг 19-1. Воссоздание в памяти следа выполнения программы

```
[--sp] = (r7:7, p5:2); /* Сохранение регистров,
используемых в этой программе */
p5 = 32; /* Для опустошения TBUF необходимо выполнить 32
операции чтения */
p2.l = buf; /*Указатель на заголовок программного буфера
трассировки */
p2.h = buf; /*Заголовок содержит указатель на первую
доступную ячейку буфера buf*/
```

19 Отладка процессора Blackfin

```
p4 = [p2++]; /*Считывается значение заголовка буфера buf
(указателя на первую доступную пустую ячейку buf)*/
p3.l = TBUF & 0xffff; /* Младшие 16 битов TBUF */
p3.h = TBUF >> 16; /* Старшие 16 битов TBUF */

lsetup(loop1_start, loop1_end) lc0 = p5;
loop1_start: r7 = [p3]; /*Чтение TBUF */
loop1_end: [p4++] = r7; /* Запись в память и инкремент */
[p2] = p4; /* В заголовке буфера buf сохраняется указатель
на следующую доступную ячейку */
(r7:7, p5:3) = [sp++]; /* Извлечение сохраненных регистров
из стека */
```

Блок мониторинга выполнения

Два 32-разрядных счетчика (регистры счетчиков монитора выполнения, PFCNTR[1:0]) считают число появлений события ядра процессора в течение периода мониторинга. Эти регистры обеспечивают способ отображения баланса загрузки различных ресурсов кристалла. С их помощью можно производить анализ и сравнение ожидаемого и действительного коэффициентов использования. Кроме того, они также позволяют отслеживать такие события, как неправильное предсказание переходов и циклы останова.

Регистры PFCNTRn

На рис. 19-2 показаны регистры счетчиков монитора выполнения PFCNTR[1:0]. Регистр PFCNTR0 содержит значение счетчика выполнения 0. Регистр PFCNTR1 содержит значение счетчика выполнения 1.

Регистры счетчиков монитора выполнения (PFCNTRn)

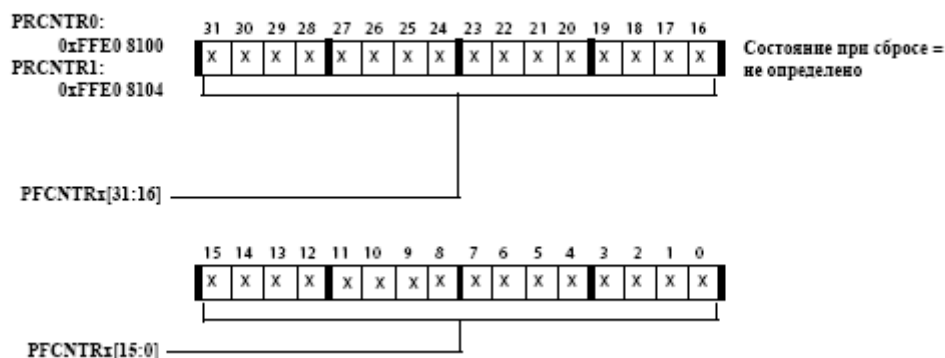


Рис. 19-12. Регистры счетчиков монитора выполнения

Регистр PFCTL

Для активизации блока мониторинга выполнения необходимо установить бит PFPWR в регистре управления монитором выполнения (PFCTL), показанном на рис. 19-13. После активизации блока вступают в силу значения отдельных битов активизации счетчиков (PFCENn). Биты PFCENx используются для включения или отключения мониторов выполнения в Пользовательском режиме, режиме Супервизора или обоих режимах. Биты PEMUSWx используются для выбора вызываемого события.

Регистр управления монитором выполнения (PFCTL)

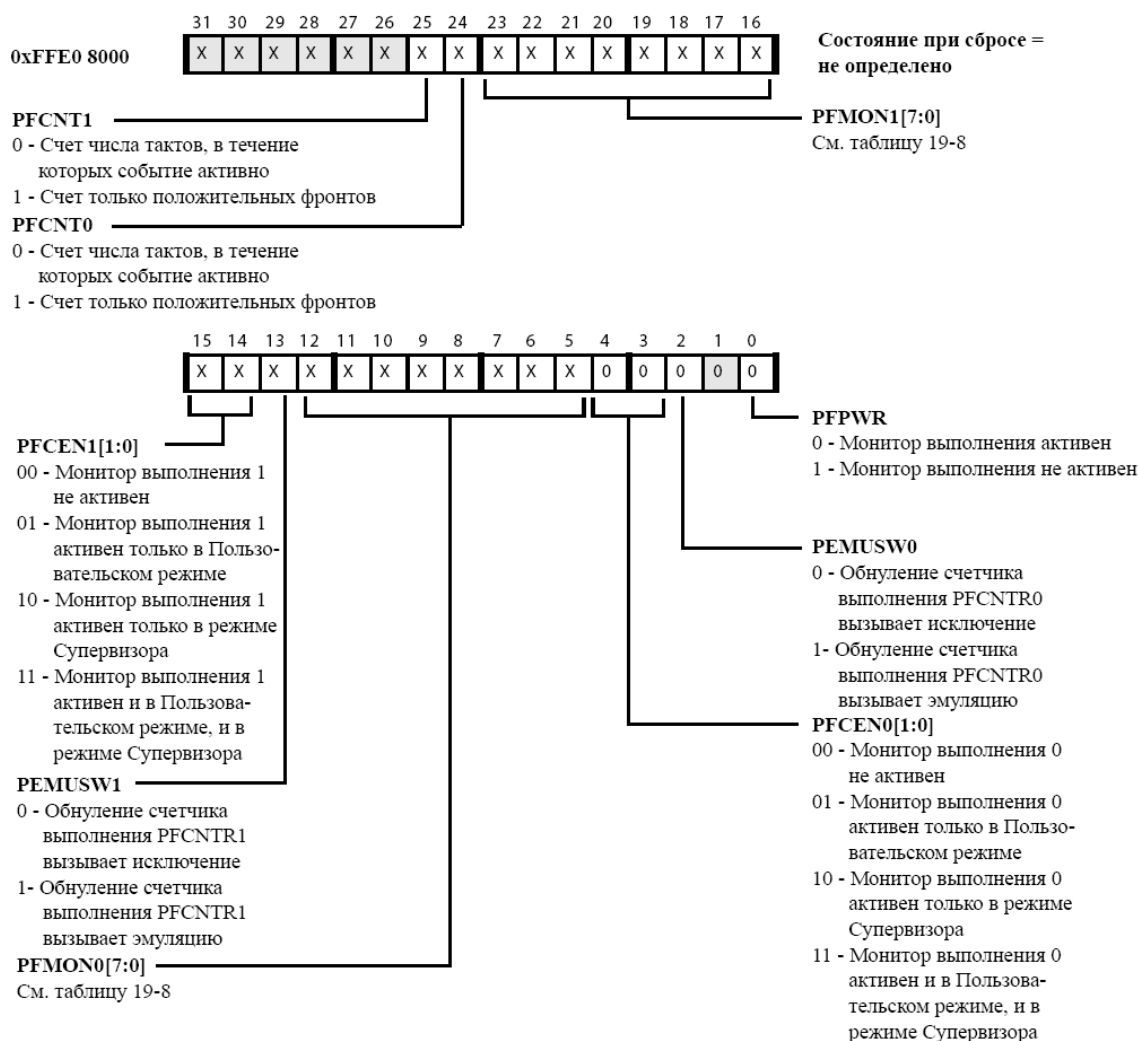


Рис. 19-13. Регистр управления монитором выполнения

Таблица событий монитора

В таблице 19-8 указаны события, вызывающие инкремент регистров счётчиков монитора выполнения (PFMON0 или PFMON1).

19 Отладка процессора Blackfin


Таблица 19-8. Таблица событий монитора

Значение поля PFMONx	События, вызывающие инкремент значения счетчика
0x00	Итерация цикла 0
0x01	Итерация цикла 1
0x02	Буфер цикла 0 не оптимизирован
0x03	Буфер цикла 1 не оптимизирован
0x04	Переходы, инвариантные к значению PC (требует разрешения записи в буфер трассировки, см. раздел “Регистр TBUFCTL”)
0x06	Условные переходы
0x09	Любые переходы (вызовы, возвраты, переходы) за исключением переходов, вызванных прерыванием (требует разрешения записи в буфер трассировки, см. раздел “Регистр TBUFCTL”)
0x0A	Остановы, вызванные командами CSYNC, SSYNC
0x0B	Команды EXCPT
0x0C	Команды CSYNC, SSYNC
0x0D	Подтвержденные команды
0x0E	Воспринятые прерывания
0x0F	Исключения, вызванные неверным выравниванием адреса
0x10	Такты останова, вызванные чтением после записи регистров DAG
0x13	Такты останова, вызванные чтением данных после записи в вычислительных устройствах
0x80	Выборки из памяти команд, отложенные из-за конфликтов DMA (при возникновении события счётчик инкрементируется, по меньшей мере на 2)
0x81	Остановы памяти команд (вызванные промахами в кэше или операциями FlushI, при операциях FlushI счётчик инкрементируется на 3). Следует отметить, что остановки памяти команд вызывают остановки процессора, только если FIFO блока сборки команд становится пуст
0x82	Остановы, вызванные заполнением памяти команд (кэшируемым или некэшируемым). Следует отметить, что остановки памяти команд вызывают остановки процессора, только если FIFO блока сборки команд становится пуст
0x83	Поступление в блок сборки команд процессора 64-разрядного слова из памяти команд
0x90	Останов памяти
0x91	Останов памяти данных, не скрытый остановом процессора
0x92	Останов памяти данных, вызванный заполнением буфера сохранения
0x93	Останов памяти данных, вызванный заполнением буфера записи из-за перехода от обслуживания прерывания с высоким приоритетом к обслуживанию прерывания с низким приоритетом
0x94	Остановы пересылки данных из буфера записи в память данных, вызванные нехваткой данных, подтвержденных процессором.
0x95	Останов буфера заполнения памяти данных
0x96	Остановы памяти данных, вызванные конфликтом массива или тега (DAG – DAG или DMA – DAG)
0x97	Остановы памяти данных, вызванные конфликтом массива (DAG – DAG или DMA – DAG)
0x98	Остановы памяти данных
0x99	Остановы памяти данных, вызывающие останов процессора
0x9A	Завершение заполнения строки банка А кэша данных
0x9B	Завершение заполнения строки банка В кэша данных
0x9C	Поступление жертв кэша данных из банка А
0x9D	Поступление жертв кэша данных из банка В
0x9E	Запрос заполнения строки кэша данных с высоким приоритетом
0x9F	Запрос заполнения строки кэша данных с низким приоритетом

Счетчик тактов

Счетчик тактов считает такты CCLK во время выполнения программы. Когда процессор находится в режиме Супервизора или Пользовательском режиме производится счет всех тактов, включая такты, затрачиваемые на выполнение команд, состояния ожидания, обработку прерываний и событий. В режиме Эмуляции работа счетчика тактов останавливается.

Счетчик тактов является 64-разрядным и инкрементируется на каждом такте. Значение счетчика хранится в двух 32-разрядных регистрах, CYCLES и CYCLES2. Регистр CYCLES содержит младшие 32 бита, регистр CYCLES2 – старшие 32 бита.

 Для получения достоверного значения количества тактов чтение регистра CYCLES должно производиться до чтения регистра CYCLES2.


В Пользовательском режиме эти два регистра доступны только для чтения. В режимах Супервизора и Эмуляции эти регистры доступны и для чтения, и для записи.

Для включения счетчика тактов необходимо установить бит CCEN в регистре SYSCFG. Ниже приводится пример, демонстрирующий использование счетчика тактов:

```
R2 = 0;
CYCLES = R2;
CYCLES2 = R2;
R2 = SYSCFG;
BITSET(R2, 1);
SYSCFG = R2;
/* Вставьте сюда тестируемый код */
R2 = SYSCFG;
BITCLR(R2, 1);
SYSCFG = R2;
```

Регистры CYCLES и CYCLES2

Регистры счетчика тактов выполнения (CYCLES и CYCLES2) показаны на рис. 19-14. Этот 64-разрядный счетчик инкрементируется на каждом такте CCLK. Регистр CYCLES содержит 32 младших бита 64-разрядного значения счетчика тактов. Регистр CYCLES2 содержит 32 старших бита.

 Регистры CYCLES и CYCLES2 являются регистрами системы, но не отображены в карте памяти. Полный список регистров системы приведен в таблице 4-1.

19 Отладка процессора Blackfin

Регистры счетчика тактов выполнения (CYCLES и CYCLES2)

В Пользовательском режиме доступны только для чтения. В режимах Эмуляции и Супервизора доступны и для чтения, и для записи.

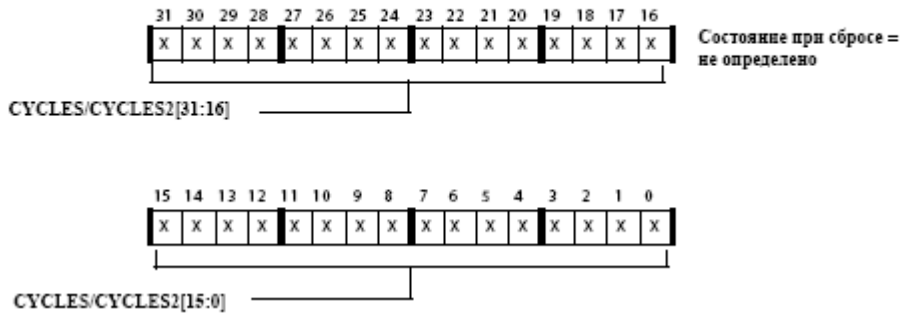


Рис. 19-14. Регистры счетчика тактов выполнения

Регистр идентификации продукта

32-разрядный регистр идентификации ЦСП (DSPID) – это регистр ядра, отображенный в карте памяти, который содержит поля версии и идентификации ядра.

Регистр DSPID

Регистр идентификации ЦСП (DSPID), показанный на рис. 19-15, является частью ядра и доступен только для чтения.

Регистр идентификации ЦСП (DSPID)

Доступен только для чтения

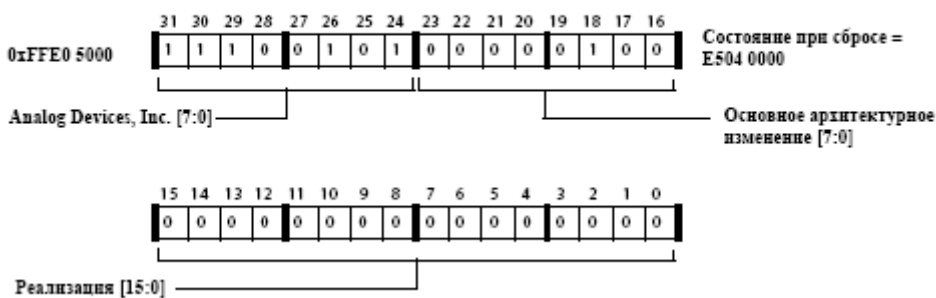


Рис. 19-15. Регистр идентификации ЦСП