**4.4.14 Регистр состояния жесткого неисправности (HFSR)**

Смещение Адрес: 0x2c

Значение сброса: 0x0000 0000

Требуется привилегия: Привилегированный

HFSR дает информацию о событиях, которые активирует обработчик жесткий неисправности. Этот регистр читается, напишите очистить. Это означает, что биты в регистре чтения обычно, но писать 1 любому немного очищает этот бит в 0.

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16

|  |  |  |
| --- | --- | --- |
| DEBU  G\_VT | FORC  ED | Зарезервированный |
| rc\_w1 | rc\_w1 |

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

|  |  |  |
| --- | --- | --- |
| Зарезервированный | VECT  TBL | Res. |
| rc\_w |

Бит 31 DEBUG\_VT: Зарезервировано для отладки использования. При записи в регистр необходимо записать 0 в этот бит, в противном случае поведение непредсказуемо.

Бит 30 вынужденным: принудительные неисправность. Указывает на принудительные вину, порожденный эскалации вине с настраиваемым приоритетом, которые не могут быть ручки, либо из-за приоритета или потому, что это отключено.

Когда этот бит установлен в 1, обработчик трудно вина должны прочитать другой статус вина регистрирует найти причиной неисправности.

0: Нет принудительные вина

1: принудительные вина.

Биты 29: 2 зарезервировано, должно быть очищены

Бит 1 VECTTBL: таблица векторов трудно вина. Указывает на неисправность Шинаа на вектор читаемой таблицы, во время

обработка исключений. Эта ошибка всегда обрабатываются обработчиком жесткого отказа. Когда этот бит установлен в 1, значение PC сложены для возвращенных исключением точек с инструкцией

который был вытеснен исключения.

0: нет Шинаа неисправности таблицы векторов не читать

1: Шина неисправности таблицы векторов читать.

Бит 0 зарезервирован должны храниться очищается

**4.4.15 Управление памятью адрес ошибки регистр (MMFAR)**

Смещение Адрес: 0x34

Значение сброса не определено

Требуется привилегия: Привилегированный

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| MMFAR[31:16] | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| MMFAR[15:0] | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

Биты 31: 0 **MMFAR** : Память адрес ошибки управления.

Когда MMARVALID бит MMFSR установлен в 1, это поле содержит адрес

место, которое породило ошибка в управлении памятью.

Когда выровненным неисправностей доступа, адрес фактический адрес, нарушенными. Потому что Одноместный чтения или команда записи может быть разделен на несколько выравнивается обращений, адрес ошибки

может быть любой адрес в диапазоне от запрашиваемого размера доступа.

Флаги в регистре MMFSR указать причину неисправности, и будет ли значение в MMFAR действовать. См [*регистр состояния Настраиваемые ошибки (CFSR; UFSR + BFSR + MMFSR) на*](https://translate.googleusercontent.com/translate_f#221)

[*страница 221*](https://translate.googleusercontent.com/translate_f#221)

**4.4.16 Шина адрес ошибки регистр (BFAR)**

Смещение Адрес: 0x38

Значение сброса не определено

Требуется привилегия: Привилегированный

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| BFAR [31:16 | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| BFAR[15:0] | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

Биты 31: 0 **BFAR** : адрес шины вина

Когда BFARVALID бит BFSR установлен в 1, это поле содержит адрес ячейки

что генерируется неисправность Шины.

Когда невыровненный ошибки доступа адрес в BFAR является одним просьбе

инструкция, даже если это не адрес вине.

Флаги в регистре BFSR указать причину неисправности, и будет ли значение в BFAR

действительна. См [*Настраиваемый регистр состояния неисправности (CFSR; UFSR + BFSR + MMFSR) на стр 221*](https://translate.googleusercontent.com/translate_f#221)

**4.4.17 Дополнительный регистр состояния неисправности (AFSR)**

Смещение Адрес: 0x3C

Значение сброса не определено

Требуется привилегия: Привилегированный

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| IMPDEF[31:16] | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| IMPDEF[15:0] | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

Биты 31: 0 **IMPDEF** : Реализация определены. AFSR содержит дополнительную информацию о неисправности системы. Биты карту к входным AUXFAULT сигналов.

Этот регистр читать, писать, чтобы очистить. Это означает, что биты в регистре чтения обычно, но писать 1 к любому немного очищает этот бит в 0.

Каждый AFSR немного карты непосредственно к AUXFAULT входом процессора, и один-цикла ВЫСОКОЙ Сигнал на входе устанавливает соответствующий бит в AFSR один. Он остается в 1, пока вы не напишете 1

на некоторое время, чтобы очистить его к нулю.

Когда AFSR бит зафиксирован, как один, исключение не возникает. Используйте прерывание, если Исключение необходимости.

**4.4.18 Управление системой блочной конструкции советы и подсказки**

Убедитесь, программное обеспечение использует унифицированных доступов правильный размер, чтобы доступ к управлению системы

блок регистров:

• для CFSR и SHPR1-SHPR3 исключением, он должен использовать выровнены слово доступ

• для CFSR и SHPR1-SHPR3 он может использовать байт или выровненный полуслова или слова доступ.

Процессор не поддерживает невыровненные доступы к системе управления блока регистров.

В обработчик ошибок. определить истинный адрес Ошибка в:

1.Читайте и сохранить значение MMFAR или BFAR.

2.Читайте немного MMARVALID в MMFSR, или немного BFARVALID в BFSR.

MMFAR или BFAR адрес будет использоваться только если этот бит равен 1.

Программное обеспечение должно следовать этой последовательности, потому что другой более высокий приоритет, исключение может измениться

Значение MMFAR или BFAR. Например, если выше, обработчик приоритетом вытесняет ток вина обработчик, другая неисправность может изменить значение MMFAR или BFAR.

**4.4.19 SCB Регистрация Карта**

В таблице приведены показывает управления системного блока карту регистров и сброс значений. База адрес SCB регистра блока 0xE000 ED00 для реестра, описанного в [*таблице 53*](https://translate.googleusercontent.com/translate_f#228)

**Таблица 53. SCB Регистрация Карта и сбросить значения**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Смещение | Регистрация | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0x00 | CPUID  Значение сброса | Исполнитель | | | | | | | | Вариант | | | | Постоянная | | | | PartNo | | | | | | | | | | | | Пересмотр | | | | |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 0x04 | ICSR  Значение сброса |  |  | |  |  |  |  |  | |  | VECTPENDING[9:0] | | | | | | | | | |  |  | | VECTACTIVE[8:0] | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**4,5 SysTick таймера (СТК)**

Процессор имеет 24-битный системный таймер, SysTick, который отсчитывает от стоимости перезагрузки в нуле, перезагрузки (обертывания с) значение в регистре STK\_LOAD на следующей тактового импульса, то отсчет на последующих часов.

Когда процессор останавливается для отладки счетчик не ДЕКРЕМЕНТ.

**Таблица 54. Сведения о системе Регистры таймера**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Адрес | Имя | Тип | Требуется привилегия | Значение сброса | Описание |
| 0xE000E010 | STK\_CTRL | RW | Привилегированный | 0x00000000 | Sys Tick управление и состояние регистра (STK­\_CTRL) на стр 231 |
| 0xE000E014 | STK\_LOAD | RW | Привилегированный | Неизвестный | SysTick перезагрузка значения регистра (STK\_LOAD) на стр 232 |
| 0xE000E018 | STK\_VAL | RW | Привилегированный | Неизвестный | SysTick текущее значение регистра (STK\_VAL) на стр 233 |
| 0xE000E01C | STK\_CALIB | RO | Привилегированный | 0xC0000000 | SysTick калибровки значения регистра (STK\_CALIB) на стр 234 |

**4.5.1 SysTick управления и состояния регистра (STK\_CTRL)**

Смещение Адрес: 0x00

Значение сброса: 0x0000 0000

Требуется привилегия: Привилегированный

Регистрация SysTick CTRL позволяет функции SysTick.

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16

|  |  |
| --- | --- |
| Зарезервированный | COUNT  FLAG |
| rw |

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

|  |  |  |  |
| --- | --- | --- | --- |
| Зарезервированный | CLKSO  URCE | TICK  INT | EN  ABLE |
| rw | rw | rw |

Биты 31:17 защищены, должны быть всегда очищены.

Бит 16 COUNTFLAG:

**Возвращает 1, если таймер досчитал до 0, так как в последний раз это было прочитать.**

Биты 15: 3 резерв, должны быть очищены.

Бит 2 **CLKSOURCE:** Часы Выбор источника

Выбор источника синхронизации.

0: АОБ / 8

1: Тактовая частота процессора (АОБ)

Бит 1 **TICKINT** : Запрос SysTick исключение включить

0: Обратный отсчет до нуля не утверждает запрос об исключении SysTick

1: обратный отсчет до утверждает запрос об исключении SysTick.

*Примечание: Программное обеспечение можно использовать COUNTFLAG, чтобы определить, SysTick когда-либо досчитал до нуля.*

Бит 0 **ВКЛ** : Counter включить

Включает счетчик. Когда Включить установлен в 1, счетчик загружается значение перезагрузки с НАГРУЗКИ зарегистрироваться, а затем в обратном порядке. При достижении 0, он устанавливает COUNTFLAG 1 и необязательно утверждает SysTick в зависимости от значения TICKINT. Затем она загружает перезагрузки цените снова, и начинает считать.

0: Counter отключена

Включен Счетчик: 1

**4.5.2 SysTick перезагрузки значение регистра (STK\_LOAD)**

Смещение Адрес: 0x04

Значение сброса: 0x0000 0000

Требуется привилегия: Привилегированный

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Зазервированнный | | | | | | | | RELOAD[23:16] | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | Rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RELOAD[15:0] | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

Биты 31:24 защищены, должны быть всегда очищены.

Биты 23: 0 **Обновить:** значение перезагрузки

Регистрация Load Определяет начальное значение для загрузки в регистр STK\_VAL, когда счетчик включен и когда она достигает 0.

Расчет Значение перезагрузки

Значение перезагрузки может быть любое значение в диапазоне 0x00000001-0x00FFFFFF. Начальное значение 0 можно, но не имеет никакого эффекта, потому что запрос об исключении SysTick и COUNTFLAG являются

активируется при подсчете от 1 до 0.

Значение перезагрузки рассчитывается в соответствии с его использованием:

L Для создания мульти-выстрел таймер с периодом N циклов тактовой используйте перезагрузки

Значение N-1. Например, если прерывание SysTick требуется каждые 100 тактовых импульсов, устанавливается

Перезагрузите страницу, чтобы 99.

L Для доставки единый SysTick прерывание с задержкой N циклов тактовой используйте

Перезагрузка значение N. Например, если прерывание SysTick требуется после 100 часов

Импульсы, установите RELOAD до 99.

**4.5.3 SysTick текущее значение регистра (STK\_VAL)**

Смещение Адрес: 0x08

Значение сброса: 0x0000 0000

Требуется привилегия: Привилегированный

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Зазервированнный | | | | | | | | CURRENT[23:16] | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | Rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CURRENT[15:0] | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

Биты 31:24 защищены, должны быть всегда очищены.

Биты 23: 0 **ток:** Текущее значение счетчика

Регистрация VAL содержит текущее значение счетчика SysTick.

Читает возвращает текущее значение счетчика SysTick.

Запись любого значения очищает поле 0, а также очищает бит в COUNTFLAG

STK\_CTRL зарегистрируйтесь, чтобы получить 0

**4.5.4 SysTick калибровки значение регистра (STK\_CALIB)**

Смещение Адрес: 0x0C

Значение сброса: 0x0000000

Требуется привилегия: Привилегированный

Регистрация CALIB указывает свойства калибровки SysTick.

Бит 31 **NOREF:** NOREF флаг. Изложить в ноль. Указывает, что отдельный опорный тактовый предоставляется.

Частота этих часов является HCLK / 8.

Бит 30 **SKEW:** SKEW флаг: Указывает, является ли значение TENMS точно. Читает, как один. Калибровка Значение для 1 мс неточного времени не известно, так как TENMS не известно. Это может повлиять на

Пригодность SysTick в качестве программной часов реального времени.

Биты 29:24 защищены, должны быть всегда очищены.

Биты 23: 0 **TENMS [23: 0]:** Значение калибровки. Указывает значение калибровки, когда счетчик SysTick работает на HCLK макс / 8 в качестве внешнего круглосуточно. Значение зависит от продукта, пожалуйста, обратитесь к Справочное руководство товаров, в разделе SysTick калибровочного коэффициента. Когда HCLK запрограммирован на

Максимальная частота, период SysTick является 1 мс.

Если калибровка информация не известна, вычислить значение калибровки требуется от Частота счетчика процессора или внешним тактовым сигналом.

**4.5.5 SysTick дизайн советы и подсказки**

SysTick счетчик работает по часам процессора. Если это тактового сигнала остановки для низкого Режим питания, счетчик SysTick останавливается.

Убедитесь, программное обеспечение использует выровненный слово обращается к доступа к регистрам SysTick.

SysTick счетчик перезагружается, и текущее значение определено при сбросе, правильно

Последовательность инициализации для счетчика SysTick является:

1.Значение перезагрузки программы.

2.Ясно текущее значение.

3.Программа контроля и регистра состояния.

**4.5.6**

**SysTick Регистрация Карта**

Таблица, приведенная показывает SysTick зарегистрируйтесь карту и обновления значений. Базовый адрес

блок регистров SysTick является 0xE000 E010.

**Таблица 55. SysTick Регистрация Карта и сбросить значения**

**4,6 Плавающей запятой (FPU)**

Cortex-M4F FPU выполняет расширение по FPv4-SP с плавающей точкой.

FPU полностью поддерживает одинарной точностью складывать, вычитать, умножать, делить, умножать и накапливаются, и квадратный корень операции. Она также обеспечивает преобразование между фиксированной xxxxx- точку и форматы данных с плавающей точкой, а константа с плавающей точкой инструкции.

FPU обеспечивает плавающей точкой функциональность вычислений, совместимый с ANSI / IEEE стандарт 754-2008, IEEE стандарт двоичной арифметики с плавающей точкой, называется в качестве стандарта IEEE 754.

FPU содержит 32 одинарной точностью регистры расширение, которое вы также можете получить доступ как 16 двойное регистры для загрузки, хранения и перемещения операций.

[*Таблица 56*](https://translate.googleusercontent.com/translate_f#236) показывает, система регистрирует с плавающей точкой в блоке управления системой Cortex-M4F

(SCB). Базовый адрес дополнительных регистров для расширения FP является 0xE000 ED00.

**Таблица 56. Cortex-M4F регистры с плавающей точкой система**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Адрес | Имя | Тип | Сброс | Описание |
| 0xE000ED88 | CPACR | RW | 0x00000000 | Раздел 4.6.1.: сопроцессора регистр управления доступом (CPARC) на стр 237 |
| 0xE000EF34 | FPCCR | RW | 0x00000000 | Раздел 4.6.2.: с плавающей точкой регистр управления контекст (FPCCR) на стр 237 |
| 0xE000EF38 | FPCAR | RW | - | Раздел 4.6.3.: с плавающейточкой контекст адрес регистра (FPCAR) на стр 239 |
| 0xE000EF3C | FPDSCR | RW | 0x00000000 | Раздел 4.6.5.: с плавающей точкой регистр управления статуса по умолчанию (FPDSCR) на стр 341 |
| - | FPSCR | RW | - | Раздец 4.6.4.:с плавающей точкой регистр управления состояния (FPSCR) на стр 239 |

В следующих разделах описаны регистры с плавающей точкой внедрения системы которых является специфичные для этого процессора.

*Примечание:*

*Для получения более подробной информации о стандарте IEEE и арифметики с плавающей точкой (IEEE 754), обратитесь к*

*AN4044 Указания по применению. Доступна с сайта* [*www.st.com*](http://www.st.com)*.*

**4.6.1 Регистр управления доступом сопроцессора (CPACR)**

Адрес смещения (от SCB): 0x88

Значение сброса: 0x0000000

Требуется привилегия: Привилегированный

Регистрация CPACR определяет права доступа для сопроцессоров.

Биты 31:24 защищены. Читайте как ноль, напишите Игнорировать.

Биты 23:20 **СРП:** [2n + 1: 2n] для значений п 10 и 11. привилегий доступа для сопроцессора п. Можно

Значения каждого поля:

0b00: Отказано в доступе. Любая попытка доступа формирует NOCP UsageFault.

0b01: Привилегированный доступ только. Непривилегированных доступа генерирует ошибку NOCP.

0b10: Reserved. Результат любого доступа непредсказуемо.

0b11: Полный доступ.

Биты 19: 0 Reserved. Читайте как ноль, напишите Игнорировать.

**4.6.2 С плавающей точкой регистр управления контекст (FPCCR)**

Смещение Адрес: 0x04

Сброс значения: 0xC000000

Требуется привилегия: Привилегированный

В FPCCR наборов регистров или возвращает данные управления FPU.

Бит 31**ASPEN**: позволяет управлять <2> настройка на выполнение команды с плавающей точкой. Это Результаты в автоматическом сохранении состояния оборудования и реставрации, ибо контексте с плавающей точкой, на

вход исключение и выхода.

0: отключает контроль <2> настройка на выполнение команды с плавающей точкой.

1 Включение управления <2> настройка на выполнение инструкции с плавающей точкой.

Бит 30 **LSPEN** :

0: Отключить автоматическое ленивый государственной сохранности для связи с плавающей точкой.

1: Включить автоматическое ленивый государственной сохранности для связи с плавающей точкой.

Биты 29: 9 Reserved.

Бит 8 **MONRDY** :

0: DebugMonitor отключен или приоритет не позволяют настройки MON\_PEND, когда с плавающей

было выделено точка фрейм стека.

1: DebugMonitor включен и приоритетные разрешения установки MON\_PEND, когда с плавающей точкой было выделено кадр стека.

Бит 7 Зарезервировано.

Бит 6 **BFRDY** :

0: BusFault отключен или приоритет не позволяют Установка обработчика BusFault с ожиданием

состояние, когда было выделено плавающей точкой фрейм стека.

1: BusFault включен и приоритет разрешается Установка обработчика BusFault в состоянии ожидания когда было выделено плавающей точкой фрейм стека.

Бит 5 **MMRDY** :

0: MemManage отключен или приоритет не позволяют Установка обработчика MemManage к до состояния, когда было выделено плавающей точкой фрейм стека.

1: MemManage включен и приоритет разрешается Установка обработчика MemManage к до состояния, когда было выделено плавающей точкой фрейм стека.

Бит 4 **HFRDY** :

0: Приоритет не позволяют Установка обработчика HardFault в состоянии ожидания, когда с плавающей было выделено точка фрейм стека.

1: Приоритет разрешается Установка обработчика HardFault в состоянии ожидания, когда с плавающей точкой

было выделено кадр стека.

Бит 3 **РЕЗЬБА** :

0: Режим не Автор режима, когда было выделено плавающей точкой фрейм стека.

1: Режим был Режим поток, когда было выделено плавающей точкой фрейм стека.

Бит 2 Зарезервировано.

Бит 1 **USER** :

0: уровень привилегий не пользователь, когда было выделено плавающей точкой фрейм стека.

1: Уровень привилегия пользователя при было выделено плавающей точкой фрейм стека.

Бит 1 **LSPACT** :

0: Ленивый сохранение состояние не является активным.

1: Ленивый сохранение состояние активно. было выделено плавающей точкой фрейм стека но экономия Государствоа к нему была отложена.

**4.6.3 С плавающей точкой контекст адрес регистра (FPCAR)**

Смещение Адрес: 0x08

Значение сброса: 0x0000000

Требуется привилегия: Привилегированный

Регистр FPCAR имеет расположение безлюдной с плавающей точкой регистров пространства выделено на кадр стека исключений.

Биты 31: 3 Адрес: расположение безлюдной с плавающей точкой регистров пространство, выделенное на исключение фрейм стека.

Биты 2: 0 защищены. Читайте как ноль, пишет Игнорируется.

**4.6.4 С плавающей точкой регистр управления состояния (FPSCR)**

Смещение Адрес: Не отображается

Значение сброса: 0x0000000

Требуется привилегия: Привилегированный

Регистрация FPSCR предоставляет всю необходимую контроль на уровне пользователя системы с плавающей точкой

Бит 31 **N** Код флаг негативном состоянии. Операции сравнения чисел с плавающей точкой обновлять эти флаги. Для

Подробнее о результате, см [*Таблица 57*](https://translate.googleusercontent.com/translate_f#240) .

0: Работа результат был положительным, ноль, больше или равно.

1: Операция результат был отрицательным или менее.

Бит 30 **Z** : Код флаг нуля состояние. Операции сравнения чисел с плавающей точкой обновлять эти флаги. Для более

Подробная информация о результате, обратитесь к [*таблице 57*](https://translate.googleusercontent.com/translate_f#240) ,

0: результат операции не равен нулю.

1: результат операции равен нулю.

Бит 29 **C** : Carry код состояния флажка. Операции сравнения чисел с плавающей точкой обновлять эти флаги. Для более

Подробная информация о результате, обратитесь к [*таблице 57*](https://translate.googleusercontent.com/translate_f#240) ,

0: Добавить работу не привести к бит переноса или вычесть результате операции заема бит.

1: Добавить работу в результате бит переноса или вычесть операции не привели к заема бит.

Бит 28 **V** : Код флаг переполнения состояние. Операции сравнения чисел с плавающей точкой обновить этот флаг. Для более подробной информация о результате, обратитесь к [*таблице 57*](https://translate.googleusercontent.com/translate_f#240) ,

0: операция не приведет к переполнению

1: Операция привела к переполнению.

Бит 27 Резерв.

Бит 26 **МАИ** : бит управления Альтернативная наполовину точность:

0: формат IEEE половина точности выбран.

1: альтернативный формат половина точности выбран.

Бит 25 **DN** : По умолчанию бит контроля режима NaN:

0: NaN операнды распространяются на выход операции с плавающей точкой.

1: Любая операция с участием одного или более пренебрежимо возвращает умолчанию NaN.

Бит 24 **ФЗ** : Флеш-к нулю управления бит режима:

0: Флеш-к нулю режим отключен. Поведение системы с плавающей точкой полностью соответствует

Стандарт IEEE 754.

1: Промойте к нулю режим включен.

Биты 23:22 **RMode** : Поле управления режим округления. Указанный режим округления используется почти во всех с плавающей точкой инструкции:

0b00: Округление до ближайшего режиме (RN)

0b01: Круглый к плюс бесконечности режим (RP)

0b10: Круглый сторону минус бесконечности режим (RM)

0b11: Круглый к нулю режиме (RZ).

Бит 21: 8 защищены.

Бит 7 **IDC** : Введите denormal совокупный исключением немного. Накопительное исключение бит для чисел с плавающей точкой

исключение.

1: Указывает, что соответствующее исключение произошло с 0 последней записи в нем.

Бит 6: 5 зарезервировано

Бит 4 **IXC** : Неточное совокупный немного исключение. Накопительное исключение бит для чисел с плавающей точкой исключения.

1: Указывает, что соответствующее исключение произошло с 0 последней записи в нем.

Бит 3 **UFC** : Underflow совокупный немного исключение. Накопительное исключение бит для чисел с плавающей точкой исключения.

1: Указывает, что соответствующее исключение произошло с 0 последней записи в нем.

Бит 2 **OFC** : Переполнение совокупный немного исключение. Накопительное исключение бит для чисел с плавающей точкой исключения.

1: Указывает, что соответствующее исключение произошло с 0 последней записи в нем.

Бит 1 **DZC** : Деление на ноль совокупного немного исключений. Накопительное исключение бит для чисел с плавающей точкой

исключение. 1: Указывает, что соответствующее исключение произошло с 0 последней записи в нем.

Бит 0 **МОК** : Недопустимая операция совокупный исключением немного. Накопительное исключение бит для чисел с плавающей точкой

исключение. 1: Указывает, что соответствующее исключение произошло с 0 последней записи в нем.

**Таблица 57. Влияние сравнению с плавающей точкой на условии флагов**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Сравнение результатов | N | Z | C | V |
| Равным | 0 | 1 | 1 | 0 |
| Меньше чем | 1 | 0 | 0 | 0 |
| Более | 0 | 0 | 1 | 0 |
| Незаказанный | 0 | 0 | 1 | 1 |

**4.6.5 С плавающей точкой регистр управления статуса по умолчанию (FPDSCR)**

Смещение Адрес: 0x0C

Значение сброса: 0x0000000

Требуется привилегия: Привилегированный

Регистрация FPDSCR имеет значения по умолчанию для данных контроля состояния с плавающей точкой.

Биты 31:27 защищены, должны быть всегда очищены.

Бит 26 **МАИ:** Значение по умолчанию для FPSCR.AHP

Бит 25 **DN:** Значение по умолчанию для FPSCR.DN

Бит 24 **ФЗ:** Значение по умолчанию для FPSCR.FZ

Биты 23:22 **RMode:** Значение по умолчанию для FPSCR.RMode

Биты 21: 0 забронированных, должны быть очищены.

**4.6.6 Включение FPU**

FPU отключена от сброса. Вы должны включить его, прежде чем вы можете использовать любой плавающей точкой инструкции.

Пример показывает последовательность пример кода для включения FPU в обоих привилегированных и пользовательских режимов. Процессор должен быть в привилегированном режиме для чтения и записи CPACR.

**Пример**

; CPACR находится по адресу 0xE000ED88

LDR.W R0 = 0xE000ED88

; Читайте CPACR

LDR R1, [R0]

; Установить биты 20-23 для того, чтобы CP10 и CP11 сопроцессоры

ЧОО R1, R1, # (0xF << 20)

; Написать обратно измененное значение в CPACR

STR R1, [R0]; ждать магазин, чтобы завершить

DSB

; Трубопровод сброса теперь FPU включена

ISB

**4.6.7 Включение и очистки FPU прерывания исключений**

В FPU исключения флаги (IDC, UFC, OFC, DZC и кроме того, в устройствах STM32F3xxx,Флаг IXC) являются ORed и подключен к контроллеру прерываний.

В устройствах STM32F4xx там нет индивидуального маски и включить / отключить ПДП

прерывание производится на прерывания уровне контроллера.

В устройствах STM32F3xx в дополнение к ФПУ контроллер прерываний маски прерывания бит, существует индивидуальный бит маски для включения / выключения источников FPU прерываний (флаг IXC является

в масках по умолчанию в устройствах STM32F3xx).

Как это происходит очень часто, то флаг IXC исключение не подключен к прерыванию Контроллер в устройствах STM32F4xx, и не может генерировать прерывание. Если необходимо, он должен быть удалось путем опроса.

Очистка флаги исключений FPU зависит от контекста FPU сохранения / восстановления конфигурации:

• Нет плавающей точкой экономия Регистрация: когда с плавающей точкой регистр управления контекст (FPCCR)

Бит 30 LSPEN = 0 и бит 31 ОСИНЫ = 0.

Вы должны очистить источник прерывания в операциях с плавающей точкой управления и состояния регистра (FPSCR).

Пример:

зарегистрируйтесь uint32\_t fpscr\_val = 0;

fpscr\_val = \_\_get\_FPSCR ();

{Проверяем, флаги исключительных}

fpscr\_val и = (uint32\_t) 0x8F; // Очистить все флаги исключений

\_\_set\_FPSCR (fpscr\_val);

• Ленивый сохранения / восстановления: когда с плавающей точкой регистр управления контекст (FPCCR)

Бит 30 LSPEN = 1 и бит 31 ОСИНЫ = X.

В случае ленивым контексте с плавающей точкой сохранения / восстановления, фиктивная доступ для чтения должны быть

внесены плавающей точкой состояния и регистра управления (FPSCR), чтобы заставить государство

сохранение и FPSCR ясно.

Затем обрабатывать FPSCR в стеке.

Пример:

зарегистрируйтесь uint32\_t fpscr\_val = 0;

зарегистрируйтесь uint32\_t reg\_val = 0;

reg\_val = \_\_get\_FPSCR (); // Манекен доступ

fpscr\_val = \* (\_\_ IO uint32\_t \*) (FPU-> FPCAR + 0x40);

{Проверяем, флаги исключительных}

fpscr\_val и = (uint32\_t) 0x8F; // Очистить все флаги исключений

\* (\_\_ IO uint32\_t \*) (FPU-> FPCAR + 0x40) = fpscr\_val;

\_\_DMB ();

•Автоматические регистры с плавающей точкой сохранения / восстановления: когда с плавающей точкой контроль контекстное

зарегистрируйтесь (FPCCR)

Бит 30 LSPEN = 0 и бит 31 ОСИНЫ = 1.

В случае автоматического контексте с плавающей точкой сохранения / восстановления, доступ на чтение должно быть сделано

Плавающая-точка состояния и регистр управления (FPSCR), чтобы заставить ясно. Затем обрабатывать FPSCR в стеке.

Пример:

// FPU обработчик ошибок

недействительными FPU\_ExceptionHandler (uint32\_t LR, uint32\_t ИП)

{

зарегистрируйтесь uint32\_t fpscr\_val;

если (LR == 0xFFFFFFE9)

{

SP = SP + 0x60;

}

иначе, если (LR == 0xFFFFFFED)

{

SP = \_\_get\_PSP () + 0x60;

}

fpscr\_val = \* (uint32\_t \*) SP;

{Проверяем, флаги исключительных}

fpscr\_val и = (uint32\_t) 0x8F; // Очистить все флаги исключений

\* (Uint32\_t \*) SP = fpscr\_val;

\_\_DMB ();

}

// FPU IRQ обработчик

недействительными \_\_asm ​​FPU\_IRQHandler (недействительными)

{

ИМПОРТ FPU\_ExceptionHandler

MOV

R0, LR

// Перемещаем LR до R0

MOV

R1, ИП

// Сохранить SP с R1, чтобы избежать каких-либо изменений в

// Указатель стека из FPU\_ExceptionHandler

VMRS R2, FPSCR

// Манекен доступ на чтение, чтобы заставить ясно

В

FPU\_ExceptionHandler

BX

LR

}