Ошибки в микросхемах MDR12065

Настоящий документ содержит описание всех ошибок, выявленных в микросхемах на момент создания данной версии документа.

Статус документа

Настоящий документ является НЕКОНФИДЕНЦИАЛЬНЫМ.

Адрес в сети Интернет

http://www.milandr.ru

Обратная связь по продукту

Если у Вас есть какие-либо комментарии или предложения по данному продукту, свяжитесь с Вашим поставщиком, указав:

– название продукта;
– комментарии, либо краткое описание Ваших предложений;
– предпочтительный способ связи с Вами и контакты (организация, электронная почта, номер телефона).

Обратная связь по документу

Если у Вас есть какие-либо комментарии или предложения по данному документу, пожалуйста, пришлите их на электронную почту support@milandr.ru, указав:

– название документа;
– номер и/или дата документа;
– номер страницы;
– комментарии, либо краткое описание Ваших предложений;
– предпочтительный способ связи с Вами и контакты (организация, электронная почта, номер телефона).

Оглавление

Обзор 4
Категории ошибок 4
Сводная таблица ошибок 5
Ошибки категории 3 7
0001 Пропуск канала SAR АЦП при последовательном преобразовании нескольких каналов после выключения АЦП 7
0002 Не запуск вспомогательного LDO батарейного домена 8
0004 Команда dret не декодируется как illegal instruction при исполнении не в Debug-режиме 8
0005 Некорректная работа аппаратной точки останова на коде в Debug ROM 9
0006 Ошибка в tinfo-регистре модуля отладки 9
0007 Команда c.lwsp (если rd == x0) не декодируется как illegal instruction 10
0008 Чтение CSR-регистров командами csrrc, csrrsi и csrrci вызывает запись 10
0009 Некорректное значение по сбросу поля sbaccess регистра sbcs 11
0010 Остановка ядра при обработке невекторных прерываний по уровню в случае, если уровень изменяется в неактивный до завершения взятия прерывания 11
0011 Остановка счета IWDG при отсутствии частоты PCLK в процессе перезагрузки значения таймера или обновления значения предделителя 12
0012 Состояние вывода JTAG TDO (PA7) не определено после подачи питания 13
0013 Пропуск импульса SEC_CLK при использовании калибровки RTC_CLK 14
0014 "Залипание" флага HSE_RDY после пропадания частоты на OSC_IN и сброса не по питанию 16
0015 Контроллер интерфейса I2C не отслеживает clock stretching при формировании START и DIV=1-2 17
0016 Останов канала DMA при повторном запросе от АЦП во время обработки текущей транзакции 18
0017 Старший бит сдвигового регистра отладочного JTAG сбрасывается при переходе из состояния Exit2 в Shift 20
0018 Сдвиговый регистр отладочного JTAG не сбрасывается при выходе из состояния Capture-DR при выбранном регистре BYPASS 21
Лист регистрации изменений 22

Обзор

Настоящий документ содержит описание ошибок в продукте с указанием категории критичности. Каждое описание содержит:

- уникальный идентификатор ошибки;
- текущий статус ошибки;
- где существует отклонение от спецификации и условия, при которых возникает ошибка;
- последствия возникновения ошибки в типичных применениях;
- ограничения, рекомендации и способы обхода ошибки, где это возможно.

Категории ошибок

Ошибки разделены на три категории критичности:

Категория 1.

Ошибочное поведение, которое невозможно обойти. Ошибки данной категории серьезно ограничивают использование продукта во всех или в большинстве приложений, что делает устройство непригодным для использования.

Категория 2.

Ошибочное поведение, которое противоречит требуемому поведению. Ошибки данной категории могут ограничивать или серьезно ухудшать целевое использование указанных функций, но не делают продукт непригодным для использования во всех или в большинстве приложений.

Категория 3.

Ошибочное поведение, которое не было изначально определено, но не вызывает проблем в приложениях при соблюдении рекомендаций.

Сводная таблица ошибок

В таблице указано, в каких версиях микросхем присутствует ошибка. Наличие ошибки обозначено символом «X».
Версия микросхем определяется датой изготовления, указанной на корпусе микросхемы в формате ГГНН, где ГГ – год изготовления, НН – неделя изготовления.

ID	Описание	Микросхемы, изготавливаемые с даты			
		2605			
Категория 1					
Категория 2					
Категория 3					
0001	Пропуск канала SAR АЦП при последовательном преобразовании нескольких каналов после выключения АЦП	X			
0002	Не запуск вспомогательного LDO батарейного домена	X			
0004	Команда dret не декодируется как illegal instruction при исполнении не в Debug-режиме	X			
0005	Некорректная работа аппаратной точки останова на коде в Debug ROM	X			
0006	Ошибка в tinfo регистре модуля отладки	X			
0007	Команда c.lwsp (если rd == x0) не декодируется как illegal instruction	X			
0008	Чтение CSR-регистров командами csrrc, csrrsi и csrrci вызывает запись	X			
0009	Некорректное значение по сбросу поля sbaccess регистра sbcs	X			
0010	Остановка ядра при обработке невекторных прерываний по уровню в случае, если уровень изменяется в неактивный до завершения взятия прерывания	X			
0011	Остановка счета IWDG при отсутствии частоты PCLK в процессе перезагрузки значения таймера или обновления значения предделителя	X			
0012	Состояние вывода JTAG TDO (PA7) не определено после подачи питания	X			
0013	Пропуск импульса SEC_CLK при использовании калибровки RTC_CLK	X			

ID	Описание	Микросхемы, изготавливаемые с даты			
		2605			
0014	"Залипание" флага HSE_RDY после пропадания частоты на OSC_IN и сброса не по питанию	X			
0015	Контроллер интерфейса I2C не отслеживает clock stretching при формировании START и DIV=1-2	X			
0016	Останов канала DMA при повторном запросе от АЦП во время обработки текущей транзакции	X			
0017	Старший бит сдвигового регистра отладочного JTAG сбрасывается при переходе из состояния Exit2 в Shift	X			
0018	Сдвиговый регистр отладочного JTAG не сбрасывается при выходе из состояния Capture-DR при выбранном регистре BYPASS	X			

Ошибки категории 3

0001 Пропуск канала SAR АЦП при последовательном преобразовании нескольких каналов после выключения АЦП

Статус
Исследование.

Описание
В режиме последовательного преобразования нескольких каналов после отключения АЦП (бит Gfg_REG_ADON) и при последующем его включении однократно пропускается канал, на котором остановилось преобразование в момент отключения АЦП. Происходит преобразование следующего канала, участвующего в последовательном преобразовании.

Условия
Включение АЦП после его отключения в режиме последовательного преобразования нескольких каналов.

Последствия
Пропуск преобразования канала АЦП, на котором остановилось преобразование при отключении.

Рекомендации и способы обхода
После отключения АЦП при использовании последовательного преобразования нескольких каналов:

1 Отключить переключение каналов (бит Cfg_REG_CHCH).
2 Включить переключение каналов только для канала, на котором остановилось преобразование при отключении, и участвующих в преобразовании каналов с бóльшими номерами (регистр ADC1_CHSEL).

После включения АЦП:

3 Включить переключение для всех требуемых каналов.

0002 Не запуск вспомогательного LDO батарейного домена

Статус
Исследование.

Описание
Вспомогательный LDO батарейного домена может не запустится при несоответствии фронта питания V_{CCB} требованиям спецификации.

Условия
Фронт питания V_{CCB} более 1 мс или менее 10 мкс.

Последствия
Батарейный домен будет работать, используя основное питание и основной LDO. Но при отключении основного питания (V_{CCA} и V_{CC}) информация из батарейного домена может исчезнуть, несмотря на наличие батарейного питания.

Рекомендации и способы обхода
После подачи напряжения питания на вывод VCCB необходимо однократно прописать в бит BLDO_BOOST регистра BKP_LDO значение «1» и через задержку не менее 20 мкс – значение «0».

0004 Команда dret не декодируется как illegal instruction при исполнении не в Debug-режиме

Статус
Исследование.

Описание
Неверное декодирование инструкции (команды) dret не в Debug-режиме.

Условия
Исполнение ассемблерной инструкции (команды) dret не в режиме Debug-режиме.

Последствия
Не декодирование как illegal instruction.

Рекомендации и способы обхода
Не использовать dret в пользовательском коде программы.

0005 Некорректная работа аппаратной точки останова на коде в Debug ROM

Статус
Исследование.

Описание
Останов на коде в Debug ROM не выполняется.

Условия
Установка точки останова на коде в Debug ROM.

Последствия
Возникновение ошибки.

Рекомендации и способы обхода
Не использовать отладку в Debug ROM.

0006 Ошибка в tinfo-регистре модуля отладки

Статус
Исследование.

Описание
Ошибка в tinfo-регистре, приводящая к проблеме механизма Discovery.

Условия
Применение OpenOCD с включённым механизмом Discovery.

Последствия
Возникновение ошибки.

Рекомендации и способы обхода
I. Использовать OpenOCD из основной ветки https://github.com/openocd-org/openocd до версии 0.12.0 включительно.
II. 1. Использовать OpenOCD для RISCV https://github.com/riscv/riscv-openocd до версии 2018.12.0;
2. Использовать patch* для OpenOCD для RISCV версии 0.12.0+ dev-03629-g87331a82a. Patch заменяет значение, прочитанное из регистра tinfo, на верное.

* https://support.milandr.ru/upload/iblock/366/1bla1mied7u7vkvie2nuenks9d440kw0/tinfo-patch.zip

0007 Команда c.lwsp (если rd == x0) не декодируется как illegal instruction

Статус
Исследование.

Описание
Команда (инструкция) c.lwsp должна декодироваться как illegal instruction.

Условия
Исполнение команды (инструкции) c.lwsp (если rd == x0).

Последствия
Не декодирование illegal instruction.

Рекомендации и способы обхода
Компилятор “C” такие команды не генерирует, поэтому не использовать ассемблерные вставки в “C”-код с такими командами.

0008 Чтение CSR-регистров командами csrrc, csrrsi и csrrci вызывает запись

Статус
Исследование.

Описание
Команды (инструкции) csrrs, csrrc, csrrsi и csrrci могут быть использованы для выполнения операции чтения CSR-регистров без выполнения операции записи при rs1 = x0 (csrrs, csrrc) или uimm = 0 (csrrsi, csrrci):

```
- csrrs  rd, csr, x0;
- csrrc  rd, csr, x0;
- csrrsi rd, csr, 0;
- csrrci rd, csr, 0.
```

Однако команды csrrc, csrrsi и csrrci при выполнении операции чтения CSR-регистров также ошибочно выполняют операцию записи, что вызывает побочные эффекты записи (например, для регистра xNXTI) или вызывает исключение illegal-instruction при обращении к CSR-регистру, доступному только для чтения.

Условия
Выполнение чтения CSR-регистра командой csrrc, csrrsi и csrrci при rs1 = x0 (csrrc) и uimm = 0 (csrrsi, csrrci).

Последствия
Помимо операции чтения также выполняется операция записи, которая не приводит к изменению содержимого регистра (т.к. маска установки/сброса бит равна 0), но вызывает побочные эффекты записи (например, для регистра xNXTI) или вызывает исключение illegal-instruction при обращении к CSR-регистру, доступному только для чтения.

Рекомендации и способы обхода
Для выполнения операции чтения CSR-регистров без выполнения операции записи использовать команду csrrs с rs1 = x0 или псевдокоманду csrr (кодируется как csrrs с rs1 = x0):

```
- csrrs rd, csr, x0;
- csrr  rd, csr.
```

0009 Некорректное значение по сбросу поля sbaccess регистра sbcs

Статус
Исследование.

Описание
Согласно спецификации RISC-V External Debug Support Version 0.13 значение поля sbaccess, которое задает разрядность данных для доступа к памяти с помощью System Bus Access, должно быть установлено по сбросу в значение 2, что соответствует размеру транзакции в 32 бита. Фактически значение по сбросу поля sbaccess равно 0, что соответствует размеру транзакции в 8 бит.

Условия
Всегда.

Последствия
Доступ к памяти в отладочном режиме с помощью System Bus Access будет осуществляться к данным с размером, отличным от ожидаемого.

Рекомендации и способы обхода
Устанавливать значение поля sbaccess в требуемое значение перед осуществлением любых операций с помощью System Bus Access.

0010 Остановка ядра при обработке невекторных прерываний по уровню в случае, если уровень изменяется в неактивный до завершения взятия прерывания

Статус
Исследование.

Описание
Зависание (счетчик PC останавливается, на шине AHB нет полноценного обмена) процессорного ядра может произойти в случае обработки невекторных прерываний, настроенных на взятие по уровню. Такое событие может произойти, если после начала обработки прерывания его уровень изменяется в неактивный до завершения взятия прерывания.

Условия
1 Прерывание настроено на срабатывание по уровню.
2 Прерывание невекторное.
3 Уровень запроса прерывания изменяется в неактивный до завершения взятия прерывания.

Последствия
Остановка ядра микросхемы.

Рекомендации и способы обхода
В контроллере CLIC настраивать используемые прерывания в один из описанных ниже режимов:
1 В векторный режим работы на срабатывание по уровню или фронту;
2 В невекторный режим работы на срабатывание по фронту.

0011 Остановка счета IWDG при отсутствии частоты PCLK в процессе перезагрузки значения таймера или обновления значения предделителя

Статус
Исследование.

Описание
В процессе работы IWDG остановка частоты PCLK блока IWDG после вызова запроса перезагрузки значения таймера через регистр IWDG_KEY (записью значения 0xAAAA) и перед фактическим обновлением значения таймера или после вызова запроса обновления предделителя (запись в регистр IWDG_PR) и перед фактическим обновлением значение предделителя блокирует сигнал запроса на обновление значения таймера в активном состоянии. В результате чего до повторного появления частоты PCLK или любого сброса сторожевой таймер IWDG не осуществляет счет и не формирует сброс. Так как в случае запроса перезагрузки IWDG непрерывно перезагружается значением перезагрузки, а в случае запроса обновления предделителя – непрерывно осуществляет обновление предделителя. После появления частоты PCLK и фактического обновления соответствующих значений или после любого сброса IWDG восстанавливает процесс счета.

Условия
Остановка частоты PCLK блока IWDG:
1 После вызова запроса перезагрузки значения таймера через регистр IWDG_KEY (записью значения 0xAAAA) и перед фактическим обновлением значения таймера в процессе работы IWDG;
2 После вызова запроса обновления предделителя таймера через регистр IWDG_PR и перед фактическим обновлением предделителя в процессе работы IWDG.

Последствия
Отсутствие счета таймера IWDG и формирования им сброса до появления частоты PCLK или любого сброса.

Рекомендации и способы обхода
Реализовать в разрабатываемой системе один или несколько из предложенных способов:
1 Перед вызовом запроса перезагрузки значения таймера и на время сброса флага RVU, а также перед вызовом запроса обновления предделителя и на время сброса флага PVU, переводить тактирование PCLK (соответствует частоте HCLK) на частоту, которая гарантируется в разрабатываемой системе – например, LSI (поскольку наличие частоты LSI обязательно для работы IWDG, в случае её отсутствия IWDG не сбросит систему при любых обстоятельствах);
2 Использовать сторожевые таймеры IWDG и WWDG совместно, при этом время между их перезагрузкой должно быть не менее одного периода частоты LSI и WWDG должен быть запущен до настройки IWDG;
3 Использовать внешний сторожевой таймер.

0012 Состояние вывода JTAG TDO (PA7) не определено после подачи питания

Статус
Исследование.

Описание
Отладочный TAP-контроллер интерфейса JTAG при подаче питания сбрасывается по сигналу POR (сброс по питанию) и переходит в состояние Test-Logic-Reset. Если работа отладочного интерфейса JTAG разрешена, то выводы PA[9:6] переходят под управление отладочного TAP-контроллера.
Согласно стандарту JTAG IEEE 1149.1, TAP-контроллер в состоянии Test-Logic-Reset должен перевести вывод TDO в высокоимпедансное состояние. Однако из-за ошибки состояние вывода TDO (PA7) после сброса по питанию не определено – вывод может находиться в одном из следующих состояний: высокоимпедансное, высокий логический уровень, низкий логический уровень. Гарантированный переход вывода TDO (PA7) в высокоимпедансное состояние после сброса по питанию выполняется по спадающему фронту сигнала на входе TCK (PA6).

Условия
Отладочный TAP-контроллер интерфейса JTAG после сброса по питанию, работа отладочного интерфейса JTAG разрешена.

Последствия
Состояние вывода TDO (PA7) не определено: высокоимпедансное, высокий логический уровень, низкий логический уровень.

Рекомендации и способы обхода
Учитывать при разработке.
Для управления выводом TDO (PA7) с помощью контроллера портов ввода-вывода необходимо запретить работу интерфейса JTAG одним из следующих способов:

- сброс бита JTAG_ON в регистре BKP_LDO;
- установка защиты адресного пространства Flash-памяти;
- блокировка интерфейса JTAG в регистре CHIP_ID_CTRL путем программирования памяти OTP.

0013 Пропуск импульса SEC_CLK при использовании калибровки RTC_CLK

Статус
Исследование.

Описание
В блоке RTC для формирования частоты SEC_CLK из частоты RTC_CLK используется делитель, выполненный на счетчике BKP_RTC_PREDIV_S с основанием счета BKP_RTC_PRL. Для калибровки (замедления) RTC_CLK используется счетчик BKP_RTC_PREDIV_A, который на время BKP_RTC_PREDIV_A < RTC_CAL останавливает счет BKP_RTC_PREDIV_S.
Во время работы RTC с калибровкой частоты RTC_CLK (поле RTC_CAL[7:0] != 0 в регистре BKP_RTC_CR) при определенных значениях RTC_CAL и BKP_RTC_PRL периодически происходит одновременное выполнение событий BKP_RTC_PREDIV_S == BKP_RTC_PRL и BKP_RTC_PREDIV_A == 0, которое приводит к ошибочному сбросу счетчика BKP_RTC_PREDIV_S и пропуску импульса SEC_CLK.
При пропуске импульса SEC_CLK счетчик/календарь (регистры BKP_RTC_TR/DR) и сторожевой таймер BKP_RTC_WUT (при тактировании от SEC_CLK: бит WUCK_SEL[2] == 1 в регистре BKP_RTC_CR) не изменяются и начинают отставать на 1 секунду. Соответствующие флаги в регистре BKP_RTC_CS при пропуске импульса SEC_CLK также не устанавливаются: SECF, ALRAF, ALRBF, WUTF и OWF.

Условия
Установлена калибровка частоты RTC_CLK (поле RTC_CAL[7:0] != 0) и заданы определенные значения RTC_CAL и BKP_RTC_PRL, при которых периодически происходит одновременное выполнение событий BKP_RTC_PREDIV_S == BKP_RTC_PRL и BKP_RTC_PREDIV_A == 0.

Последствия
Счетчик BKP_RTC_PREDIV_S ошибочно сбрасывается, при этом импульс SEC_CLK не формируется. При пропуске импульса SEC_CLK счетчик/календарь и сторожевой таймер BKP_RTC_WUT (при тактировании от SEC_CLK) не изменяются, соответствующие флаги также не устанавливаются: SECF, ALRAF, ALRBF, WUTF и OWF.

Рекомендации и способы обхода
1 Температурная компенсация частоты SEC_CLK используется.
Не использовать калибровку частоты RTC_CLK (поле RTC_CAL[7:0] == 0), подстройку хода RTC выполнять с помощью настройки CONSTx и BKP_RTC_PRL с учетом следующих особенностей (значения для кварцевого резонатора с номинальной частотой 32768 Гц):
- RTC_CAL выполняет замедление RTC с шагом ~0,95 ppm;
- CONSTx выполняет ускорение RTC с шагом ~7,63 ppm;
- BKP_RTC_PRL выполняет замедление/ускорение RTC с шагом ~30,52 ppm.

При подстройке хода RTC необходимо выполнить замедление/ускорение с помощью BKP_RTC_PRL (грубая подстройка), после чего выполнить ускорение с помощью CONSTx (точная подстройка), увеличив значения всех CONSTx на N, где N – значение от 0 до 3, дающее оптимальный результат.

Алгоритм расчета BKP_RTC_PRL и CONSTx при подстройке хода RTC на C ppm с использованием кварцевого резонатора с номинальной частотой 32768 Гц:

- настройка BKP_RTC_PRL:
  - K = C ppm / 30,52 ppm;
  - округление K к меньшему целому;
  - BKP_RTC_PRL = BKP_RTC_PRL(nominal) – K;
- настройка CONSTx:
  - N = (C – K · 30,52) ppm / 7,63 ppm;
  - округление N к ближайшему целому;
  - CONSTx = CONSTx(nominal) + N;
- ошибка подстройки в ppm:
  - E = K · 30,52 ppm + N · 7,63 ppm – C ppm.

Пример замедления хода RTC (номинальная частота 32768 Гц):

- исходные данные:
  - исходная частота RTC_CLK: 32773,3 Гц, т.е. ~ 32768 Гц +161,74 ppm;
  - требуемое изменение хода RTC: –161,74 ppm (замедление);
- настройка BKP_RTC_PRL:
  - K = –161,74 / 30,52 ≈ –5,3 → –6;
  - BKP_RTC_PRL = 32767 – (–6) = 32773;
- настройка CONSTx:
  - N = (–161,74 – (–6) · 30,52) / 7,63 = 21,38 / 7,63 ≈ 2,8 → 3;
  - CONSTx = CONSTx(nominal) + 3;
- ошибка подстройки:
  - E = –6 · 30,52 + 3 · 7,63 – (–161,74) = 1,51 ppm.

2 Температурная компенсация частоты SEC_CLK не используется.
Для калибровки частоты RTC_CLK использовать только ограниченный набор значений RTC_CAL, полученный с помощью скрипта* для заданного значения BKP_RTC_PRL. Перед настройкой блок RTC должен быть выключен и сброшен, при настройке регистр BKP_RTC_PREDIV_S не должен записываться значением, отличным от 0. После настройки и включения RTC значения RTC_CAL[7:0], BKP_RTC_PREDIV_S и BKP_RTC_PRL не должны перезаписываться. В этом случае ошибка проявляться не будет.

Пример инициализации RTC:

- включить источник тактирования RTC и выбрать его в поле RTC_SEL[1:0] регистра BKP_RTC_CR;
- дождаться формирования валидной частоты RTC_CLK согласно спецификации;
- выключить RTC путем сброса бита RTC_EN в регистре BKP_RTC_CR;
- выполнить сброс RTC путем последовательной установки и сброса бита RTC_RESET в регистре BKP_RTC_CR;
- записать требуемое значение в регистр BKP_RTC_PRL;
- записать допустимое значение в поле RTC_CAL[7:0] регистра BKP_RTC_CR;
- записать 0 в регистр BKP_RTC_PREDIV_S;
- дождаться окончания записи с помощью бита WEC в регистре BKP_RTC_CS;
- выполнить другие настройки RTC;
- включить RTC путем установки бита RTC_EN в регистре BKP_RTC_CR.

* https://support.milandr.ru/upload/iblock/285/9bliqywbkpifcoptft5dctv327ycqh9s/bkp_rtc_cal.py

0014 "Залипание" флага HSE_RDY после пропадания частоты на OSC_IN и сброса не по питанию

Статус
Исследование.

Описание
После включения HSE и его выхода в рабочий режим (HSE_RDY = 1), пропадания частоты HSE на OSC_IN и последующего сброса не по питанию (например, сброс от IWDG или nRESET), флаг HSE_RDY выставляется в "1" сразу после включения HSE (HSE_ON = 1) не смотря на отсутствие частоты на выводе OSC_IN.

Условия
Пропадание частоты HSE с последующим сбросом сброс не по питанию.

Последствия
Флаг HSE_RDY "залипает" в "1", нельзя использовать для определения готовности HSE.

Рекомендации и способы обхода
Использовать следующий алгоритм:
1. Стандартно включить HSE и дождаться готовности (HSE_RDY = 1).
2. Настроить PLL с тактированием от HSI (CPU_C1 = HSI, при этом HSI должен быть активен и готов), включить PLL и дождаться готовности PLL_RDY = 1, после чего выключить PLL.
3. Настроить PLL с тактированием от HSE, включить PLL и ждать готовности PLL_RDY = 1.
   - Если PLL_RDY = 0, то на выводе OSC_IN нет частоты, использовать HSE нельзя.
   - Если PLL_RDY = 1, то на выводе OSC_IN есть частота, HSE работает и готов для использования. Если использование PLL не требуется, PLL можно отключить.

0015 Контроллер интерфейса I2C не отслеживает clock stretching при формировании START и DIV=1-2

Статус
Исследование.

Описание
Контроллер интерфейса I2C при задании предделителя частоты DIV равного 1 или 2 не отслеживает растяжение тактовой частоты SCL (clock stretching) ведомым устройством при формировании сигнала START. В связи с этим отправка сигнала START (установка битов START и RD/WR в регистре CMD) при удерживании ведомым устройством линии SCL в состоянии низкого уровня будет выполнена без ожидания поднятия SCL в состояние высокого уровня, что приведет к некорректному формированию сигнала START на шине I2C (данная ситуация может наблюдаться при передаче повторного сигнала START).

Условия
Внутренний делитель частоты DIV равен 1 или 2, ведомое устройство удерживает линию SCL в состоянии низкого уровня, контроллер интерфейса I2C отправляет сигнал START (установка битов START и RD/WR в регистре CMD).

Последствия
Отправка сигнала START выполняется без ожидания поднятия SCL в состояние высокого уровня, что приводит к некорректному формированию сигнала START на шине I2C.

Рекомендации и способы обхода
Если ведомое устройство применяет растяжение тактовой частоты SCL (clock stretching), то перед формированием сигнала START необходимо выдержать время, за которое ведомое устройство гарантированно отпустит линию SCL, либо использовать предделитель частоты DIV равный 3 и более.

0016 Останов канала DMA при повторном запросе от АЦП во время обработки текущей транзакции

Статус
Исследование.

Описание
В контроллере DMA для всех каналов, обрабатывающих запросы периферийных блоков, соответствующий сигнал dma_waitonreq установлен в 1 – после завершения передачи данных контроллер DMA ожидает снятия запроса от периферийного блока.
Контроллер АЦП формирует запрос к DMA по флагу Flg_REG_EOCIF регистра ADC1_STATUS. Флаг устанавливаются при завершении преобразования (новый результат записан в регистр ADC1_RESULT) и сбрасываются при чтении регистра ADC1_RESULT.
При последовательных преобразованиях АЦП возможна следующая последовательность событий:
1. По завершении преобразования N формируется запрос к DMA.
2. Контроллер DMA начинает обработку транзакции (dma_active = 1) и выполняет чтение регистра ADC1_RESULT, запрос к DMA снимается.
3. До завершения обработки транзакции (dma_active = 1) завершается преобразование N+1, запрос к DMA устанавливаются повторно.
4. Контроллер DMA воспринимает повторно установленный запрос как продолжение предыдущего – новая передача не инициируется, контроллер ожидает снятия запроса.

Условия
Контроллер АЦП выполняет последовательные преобразования, канал DMA выполняет чтение результатов преобразования из регистра ADC1_RESULT по запросу от АЦП. При этом новое преобразование завершается после чтения регистра ADC1_RESULT контроллером DMA, но до завершения обработки текущей транзакции. Например, такая ситуация может произойти:
- при одновременной работе нескольких каналов DMA и низком приоритете канала, который обрабатывает запросы от АЦП;
- при длительной переинициализации структуры управляющих данных канала DMA.

Последствия
Повторно установленный запрос от АЦП воспринимается контроллером DMA как продолжение предыдущего – новая передача данных не выполняется, канал переходит в ожидание снятия запроса. Поскольку чтение регистра ADC1_RESULT не выполняется, то запрос остаётся установленным, передача данных контроллером DMA останавливается.

Рекомендации и способы обхода
Передача DMA должна выполняться с минимальной задержкой относительно момента формирования запроса от АЦП. Для этого рекомендуется:
- при одновременной работе нескольких каналов DMA:
    - установить для канала, который обрабатывает запросы от АЦП, высокий уровень приоритета в регистре CHNL_PRIORITY_SET;
    - уменьшить для используемых каналов период арбитража в поле R_power настроек канала channel_cfg;

- минимизировать время переинициализации структуры управляющих данных канала DMA, при этом, если переинициализация выполняется в обработчике прерывания DMA - назначить данному прерыванию высокий приоритет.

0017 Старший бит сдвигового регистра отладочного JTAG сбрасывается при переходе из состояния Exit2 в Shift

Статус
Исследование.

Описание
В TAP-контроллере отладочного интерфейса JTAG при переходе из состояния Exit2-IR в состояние Shift-IR либо из состояния Exit2-DR в состояние Shift-DR (для всех регистров данных, кроме BYPASS) происходит сброс старшего бита значения, ранее загруженного в сдвиговый регистр.

Условия
При работе с TAP-контроллером отладочного интерфейса JTAG выполняется переход из состояния Exit2-IR в состояние Shift-IR либо из состояния Exit2-DR в состояние Shift-DR (для всех регистров данных, кроме BYPASS).

Последствия
Старший бит сдвигового регистра сбрасывается в 0.

Рекомендации и способы обхода
Учитывать при разработке.
Для обхода проблемы воспользоваться одним из следующих способов:
- не выполнять переход из состояния Exit2-IR в состояние Shift-IR либо из состояния Exit2-DR в состояние Shift-DR (для всех регистров данных, кроме BYPASS), при этом загрузку сдвигового регистра выполнять целиком за один проход, находясь в состоянии Shift-IR либо Shift-DR;
- установить МК последним в JTAG-цепочке.

0018 Сдвиговый регистр отладочного JTAG не сбрасывается при выходе из состояния Capture-DR при выбранном регистре BYPASS

Статус
Исследование.

Описание
Согласно IEEE 1149.1, п. 10.1.1, если выбран регистр BYPASS, то в состоянии Capture-DR по переднему фронту TCK сдвиговый регистр должен быть сброшен в 0. Таким образом первый выдвигаемый бит в состоянии Shift-DR должен быть равен 0.
В TAP-контроллере отладочного интерфейса JTAG при выборе регистра BYPASS сброс сдвигового регистра в состоянии Capture-DR не реализован, в связи с чем сдвиговый регистр сохраняет состояние последнего загруженного бита в предыдущей последовательности DR-Scan.

Условия
Для TAP-контроллера отладочного интерфейса JTAG выбран регистр BYPASS, по переднему фронту TCK выполняется выход из состояния Capture-DR.

Последствия
Сдвиговый регистр не сбрасывается в 0, а сохраняет состояние последнего загруженного бита в предыдущей последовательности DR-Scan. Таким образом первый выдвигаемый бит в состоянии Shift-DR может быть равен 1.

Рекомендации и способы обхода
Если отсутствие сброса сдвигового регистра при выборе регистра BYPASS может влиять на последующие устройства в JTAG-цепочке, то рекомендуется установить МК в конец JTAG-цепочки, либо перед требуемыми операциями принудительно загружать в сдвиговый регистр TAP-контроллера МК последний бит равный 0.

Лист регистрации изменений

Дата	Страница	Статус	ID	Категория	Описание
10.02.26					Документ создан
04.08.26	18		0016	3	Добавлено описание ошибки
30.10.26	20 21		0017 0018	3 3	Добавлено описание ошибки; Добавлено описание ошибки