

Межпроцессорное взаимодействие (IPC) в многоядерных микроконтроллерах.

Часть 4. Модуль управления блокировками в i.MX6SX

Андрей САМОДЕЛОВ

Продолжаем цикл публикаций, посвященных способам и технике межпроцессорного обмена (IPC) в многоядерных гетерогенных СнК, одним из ядер которых является процессор ARM Cortex-M3/M4. В предлагаемой статье подробно рассмотрена структура и функционирование модуля управления блокировками (Semaphores module, SEMA4) в СнК i.MX6SX и аналогичных по архитектуре. Кроме того, показаны дополнительные функции оповещения с помощью прерываний, сообщающих о состоянии элементов блокировки (gate), это позволяет осуществившему неудачную блокировку процессору выполнять другие действия до поступления запроса на прерывание при разблокировании требуемого элемента блокировки (gate).

Введение

Модуль управления блокировками¹ (Semaphores module) блока межпроцессорного обмена IPS_Semaphores представляет собой аппаратную платформу, состоящую из ведомых (slave) устройств межпроцессорного обмена (IPS), в которой реализовано 16 аппаратных элементов блокировки² (gate) со следующими функциями:

- архитектура модуля поддерживает 16 аппаратных элементов блокировки (gates) в двухпроцессорной конфигурации, где cp0 — это ядро процессора Processor 0, а cp1 — ядро процессора Processor 1;
- аппаратные элементы блокировки (gates) представляют собой 16-элементные массивы байтового размера с доступом на чтение и запись;
- процессор блокирует элементы блокировки (gates) записью значения «номер_процессора+1» (processor_number+1) в соответствующий элемент блокировки (gates) и должен прочитать значение элемента блокировки (gates), чтобы определить его состояние и проверить, что операция блокировки прошла успешно;

- после того как элемент блокировки (gates) заблокирован, его можно разблокировать записью нулей при помощи блокирующего процессора;
- дополнительное оповещение с помощью прерываний после неудачной записи блокировки предоставляет механизм, указывающий на то, что элемент блокировки (gate) разблокирован;
- механизм защищенного сброса (secure reset) поддерживается для очистки содержимого отдельных элементов блокировки (gates) модуля блокировок (Semaphores module) или логики оповещения, а также очистки всех элементов блокировки (gates) модуля блокировок (clear_all);
- модель программирования выделяет область памяти для поддержки до 8 процессоров и 64 элементов блокировки (gates). Упрощенная блок-схема модуля управления блокировками (Semaphores module) показана на рис. 1. На этой схеме блок регистров, называемых gate0, gate1, ..., gate 15 и содержащих конечный автомат (finite state machines), реализует элементы блокировки (gate), а также логику оповещения с помощью прерываний.

Функциональные особенности

Модуль управления блокировками (Semaphores module) реализован в виде аппаратных элементов блокировки, отображенных на карту распределения памяти IPS в качестве ведомых (slave) периферийных устройств. Модуль поддерживает следующий набор функций:

- Поддержка до 16 аппаратных элементов блокировки (gate) в двухпроцессорной конфигурации.
- Каждый аппаратный элемент блокировки (gate) включает в свой состав 2-битный конечный автомат (finite state machine) на три состояния. Все 16 элементов блокировки (gate) отображаются в адресное пространство как битовый массив.
- Реализация трех состояний:
 - если значение элемента блокировки (gate) = 0b00, то его состояние = «разблокировано» (unlocked);
 - если значение элемента блокировки (gate) = 0b01, то его состояние = «заблокировано процессором processor 0» (locked by processor 0);
 - если значение элемента блокировки (gate) = 0b10, то его состояние = «заблокировано процессором processor 1» (locked by processor 1).
- Для проверки достоверности всех операций записи используется номер шинного ведущего (bus master) в качестве справочного атрибута и заданные шаблоны данных.

¹ Буквальный перевод термина Semaphores module — «модуль семафоров» — не совсем точно отражает функциональное назначение данного аппаратного блока, поэтому в статье будет использоваться термин «модуль управления блокировками», который более полно отражает его функциональное назначение.

² Термин gate в связи Semaphores gate соответствует понятию «крыло семафора» в железнодорожной терминологии и достаточно хорошо функционально согласуется с термином «элемент блокировки», который и будет использоваться далее в данной статье.

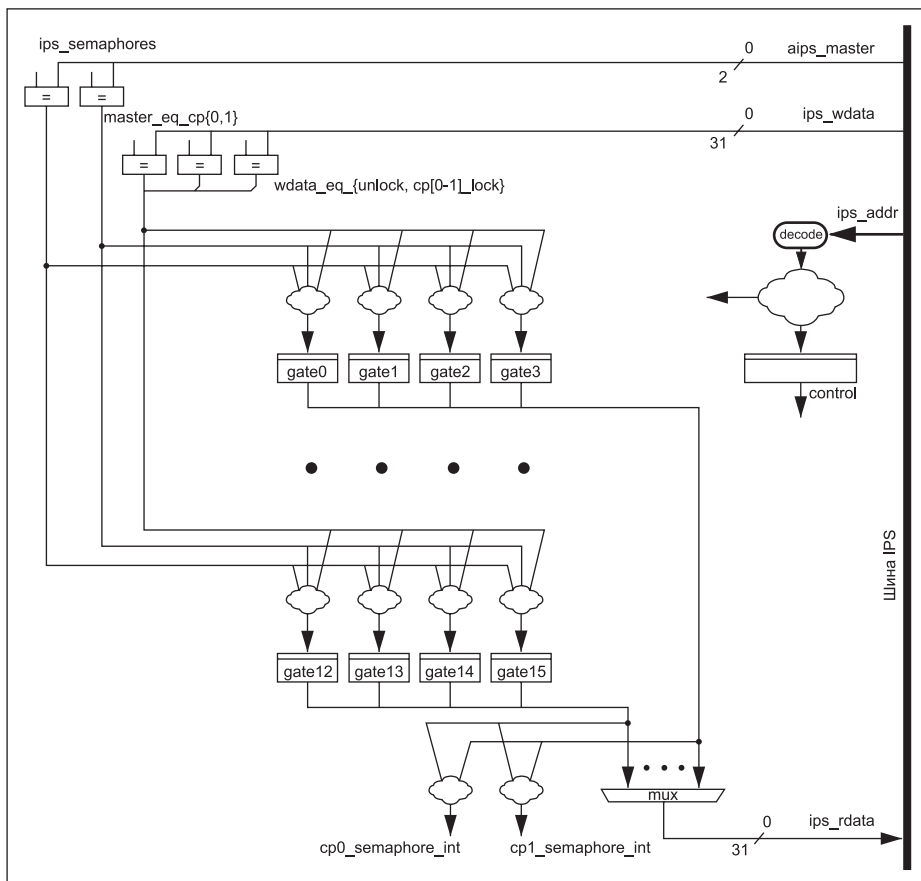


Рис. 1. Блок-схема модуля IPS_Semaphores

- После установки блокировки элемент блокировки (gate) может (и должен) быть разблокирован записью нулей с помощью блокирующего процессора.
- Дополнительное оповещение с использованием прерываний после неудачной попытки записи блокирующего значения обеспечивает механизм индикации того, что элемент блокировки (gate) разблокирован.
- Для очистки содержимого отдельных элементов блокировки (gate) или логики оповещения (notification logic), а также функции clear_all поддерживается механизм защищенного сброса (secure reset).
- Отображенный на системную память ведомый (slave) периферийный модуль IPS.
- Интерфейс к шине IPS для доступа в соответствии с моделью программирования.
- Два выхода (по одному на каждый процессор) для оповещения с помощью прерывания о неудачной попытке записи блокировки (failed lock writes).

Режимы работы

Модуль управления блокировками (Semaphores module) не поддерживает никаких специальных режимов работы. Как ведомое (slave), отображенное на карту системной памяти (memory-mapped) периферийное устройство, расположенное на ведомой шине (slave bus) блока IPS платформы, оно отве-

чает, основываясь строго на адресах памяти, выведенных на шину. Шина IPS используется для предоставления доступа в соответствии с моделью программирования элементов блокировки.

Функциональное описание

Перейдем к функциональному описанию работы модуля управления блокировками (Semaphores module). Особое внимание уделим конечным автоматам (finite state machine, FSM) регистров SEMA4_GATEn и SEMA4_CPnNTF.

Принципы функционирования регистра SEMA4_GATEn

Напомним, что каждый из регистров SEMA4_GATEn реализован как 2-битный конечный автомат на три состояния (3-state machine). Диаграмма перехода между состояниями для каждого из элементов блокировки (gate) изображена на рис. 2.

Номер ведущего на шине (bus master number) используется для идентификации ядра процессора Processor 0 (cp0) или ядра процессора Processor 1 (cp1). Платформа Standard (или Reduced) Product Platform передает номер ведущего на шине АНВ (АНВ bus master number) hmaster[2:0] через контроллер AIPS (или AIPS-Lite) и управляет выходами aips_master[2:0] к модулю управления блокировками (Semaphores module) в качестве дополнительного сигнала IPS.

Описание переходов между состояниями SEMA4_GATEn приведено в таблице 1.

Принципы функционирования регистра SEMA4_CPnNTF

Запрос на прерывание по уведомлению о сбое при записи блокировки реализован как 3-битный конечный автомат с пятью состояниями (5-state machine), который записывает информацию о сбоях при попытке записи блокировки и переходит из состоя-

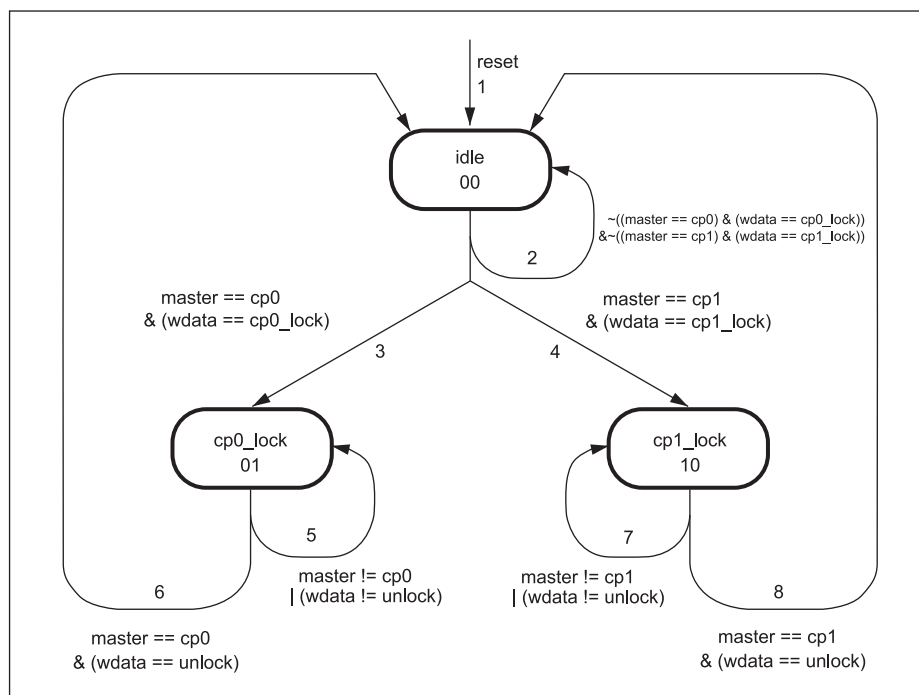


Рис. 2. Конечный автомат (State Machine) регистра SEMA4_GATEn

Таблица 1. Переходы между состояниями SEMA4_GATEn

Текущее состояние	Следующее состояние	Переход	Описание
—	idle	1	Любой тип сброса, либо системный сброс (system reset), либо сброс отдельного элемента блокировки (gate) безусловно переводит элемент блокировки (gate) в состояние ожидания (idle state).
idle	idle	2	До тех пор пока соответствующим процессором не будет произведена запись необходимого значения блокировки (lock value), элемент блокировки (gate) остается в состоянии ожидания (idle state).
idle	cp0_lock	3	При записи значения «cp0_lock», инициализированной процессором Processor 0, элемент блокировки (gate) переходит в состояние cp0_lock.
idle	cp1_lock	4	При записи значения «cp1_lock», инициализированной процессором Processor 1, элемент блокировки (gate) переходит в состояние cp1_lock.
cp0_lock	cp0_lock	5	В этом состоянии элемент блокировки (gate) остается при любых попытках записи, за исключением случая, когда процессор cp0 записывает значение разблокировки.
cp0_lock	idle	6	Элемент блокировки (gate) остается в состоянии (разблокированном) ожидания (idle state) (unlocked) до записи процессором cp0 значения разблокировки.
cp1_lock	cp1_lock	7	В данном состоянии элемент блокировки (gate) остается при любых попытках записи, за исключением случая, когда процессор cp1 записывает значение разблокировки.
cp1_lock	idle	8	Элемент блокировки (gate) остается в состоянии (разблокированном) ожидания (idle state) (unlocked) до записи процессором cp1 значения разблокировки.

ния в состоянии, основываясь на блокировке и разблокировке элемента блокировки (gate). Два специфических состояния закодированы (и видятся со стороны программного обеспечения) как SEMA4_CP0NTF[GNn] и SEMA4_CP1NTF[GNn]. Подробности можно увидеть на рис. 3.

Переходы между состояниями функции оповещения определены следующим образом.

Специфические состояния конечного автомата со стороны программного обеспечения представляются как регистры SEMA4_CPnNTF. В частности, два состояния видятся программным обеспечением, как показано в листинге 1.

```
if state == wait4_cp0_lock (0b110) then
// генерирует прерывание cp0_semaphore_int, если оно разрешено
SEMA4_CP0NTF[GNn] = 1;
else
SEMA4_CP0NTF[GNn] = 0
if state == wait4_cp1_lock (0b101) then
// генерирует прерывание cp1_semaphore_int, если оно разрешено
SEMA4_CP1NTF[GNn] = 1;
else
SEMA4_CP1NTF[GNn] = 0
```

Листинг 1. Специфические состояния конечного автомата элемента блокировки (gate)

В таблице 2 приведено описание условий переходов между состояниями конечного автомата IRQ-оповещения.

Модуль управления блокировками (Semaphores module) генерирует два выходных сигнала запроса на прерывание, по одному на каждый процессор, объединяя регистры SEMA4_CPnINE и SEMA4_CPnNTF в соответствии с булевым выражением, приведенным в листинге 2.

```
cp0_semaphore_int
= sema4_cp0ine[ine0] & sema4_cp0ntf[gn0]
| sema4_cp0ine[ine1] & sema4_cp0ntf[gn1]
| sema4_cp0ine[ine2] & sema4_cp0ntf[gn2]
...
| sema4_cp0ine[ine15] & sema4_cp0ntf[gn15]
cp1_semaphore_int
= sema4_cp1ine[ine0] & sema4_cp1ntf[gn0]
| sema4_cp1ine[ine1] & sema4_cp1ntf[gn1]
| sema4_cp1ine[ine2] & sema4_cp1ntf[gn2]
...
| sema4_cp1ine[ine15] & sema4_cp1ntf[gn15]
```

Листинг 2. Булева функция объединения регистров SEMA4_CPnINE и SEMA4_CPnNTF

Инициализация модуля управления блокировками IPS_Semaphores

Наличие у модуля управления блокировками IPS_Semaphores состояния сброса позволяет ему начинать работу сразу, без какой-либо дополнительной инициализации. Все внутренние конечные автоматы (finite state machine) очищаются по любому событию сброса, позволяя модулю немедленно начинать работу.

Использование модуля управления блокировками IPS_Semaphores

В операционной многоядерной системе большинство взаимодействий, связанных с модулем управления блокировками (Semaphores module), состоит из операций чтения и записи в регистры SEMA4_GATEn для реализации программных функций элементов блокировки (gate) с аппаратной поддержкой. Типичные сегменты кода для функций элементов блокировки (gate) выполняют несколько операций.

Для того чтобы заблокировать (закрыть) элемент блокировки (gate):

- процессор выполняет запись байта “logical_processor_number + 1” в элемент блокировки gate[i];
- процессор считывает значение из элемента блокировки gate[i] и проверяет на совпадение со значением “logical_processor_number + 1”.

Если в результате сравнения оказывается, что прочитанное значение совпадает с ожидаемым значением, то элемент блокировки (gate) блокируется и осуществляется переход к защищенному сегменту кода. Если в результате сравнения оказывается, что прочитанное

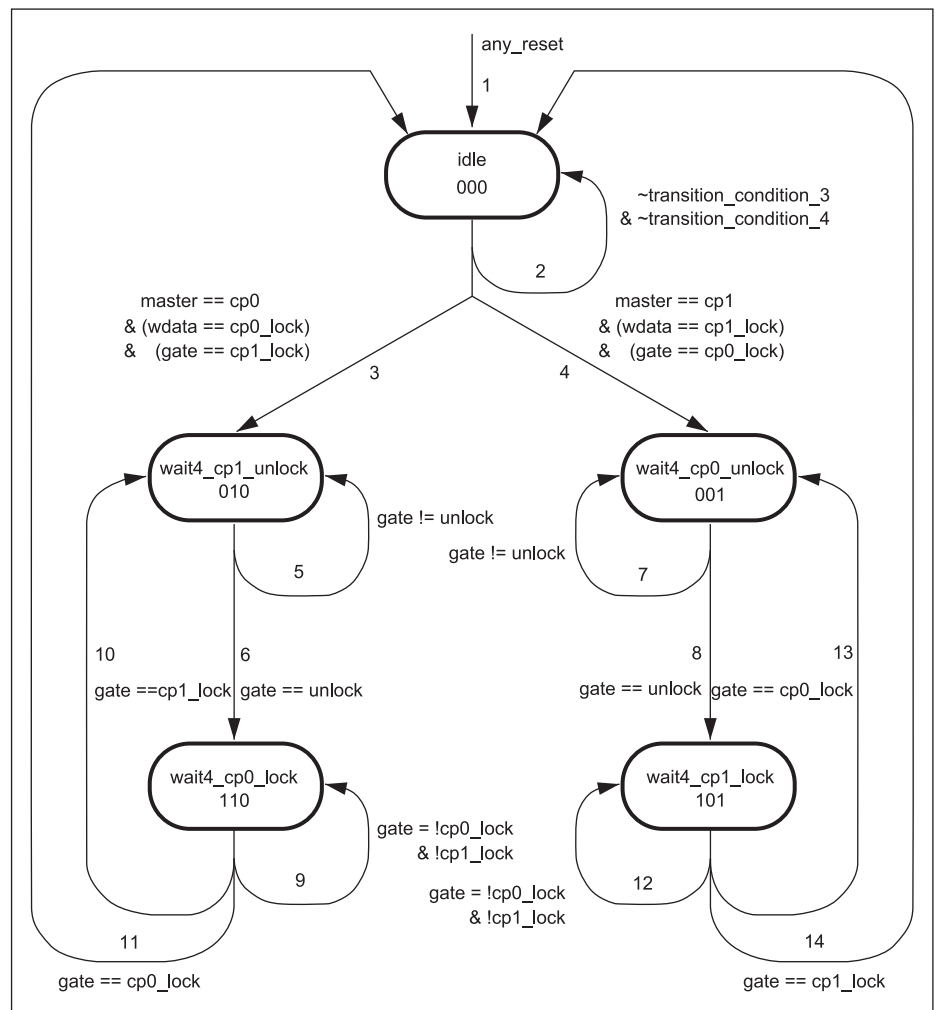


Рис. 3. Конечный автомат оповещения с помощью IRQ

значение не совпадает с ожидаемым значением, то операция блокировки завершается неудачно. Повторная попытка начинается с записи байта в элемент блокировки gate[i] в цикле ожидания, или осуществляется переход к другому пути выполнения операции с ожиданием оповещения прерыванием о неудачной блокировке (failed lock interrupt notification). В листинге 3 приведен простейший пример функции gateLock на языке C.

Эта функция соответствует примеру Хеннеси/Паттерсона (Hennessy/Patterson), описанному в документе “Multi-Core Programming 101: Software Gates”.

```
#define UNLOCK 0
#define CP0_LOCK 1
#define CP2_LOCK 2
void gateLock (n)
int n;
{
    /* Номер блокируемого элемента блокировки */
    int i;
    int current_value;
    int locked_value;
    i = processor_number(); /* Получение логического номера процессора */
    if (i == 0)
        locked_value = CP0_LOCK;
    else
        locked_value = CP1_LOCK;
    /* Чтение текущего значения элемента блокировки (gate) и ожидание до тех пор, пока состояние не станет gate == UNLOCK (разблокировано) */
    do {
        current_value = gate[n];
    } while (current_value != UNLOCK);
    /* Текущее значение состояния элемента блокировки gate == UNLOCK (разблокировано). Попытка заблокировать элемент блокировки (gate) данным процессором. Бесконечный цикл до тех пор, пока не будет получено монопольное владение элементом блокировки (gate) */
    do {
        gate[n] = locked_value; /* Запись в элемент блокировки (gate) процессором с номером processor_number + 1 */
        current_value = gate[n]; /* Чтение состояния элемента блокировки (gate) для проверки, было ли предоставлено право монопольного доступа */
    } while (current_value != locked_value);
}
```

Листинг 3. Пример реализации функции gateLock

Для того чтобы разблокировать (открыть) элемент блокировки (gate), после завершения выполнения защищенного сегмента кода блокирующий процессор выполняет запись нулевого байта в элемент блокировки gate[i].

Несколько комментариев относительно логического номера CPU. В приведенном примере для получения этого значения аппаратной конфигурации используется ссылка на processor_number(). Как правило, логические номера процессоров задаются аппаратными входными векторами к отдельным ядрам. Конкретный метод доступа к логическому номеру процессора зависит от архитектуры. Для ядер PowerPC имеется регистр идентификатора процессора (processor ID register, PIR) который называется SPR 286 и содержит требуемое значение. Для перемещения содержимого PIR в регистр общего назначения можно использовать одиночную команду:

```
mfspr rx, 286
```

где rx задает регистр назначения (GPRn). Другие архитектуры могут поддерживать

Таблица 2. Переход между состояниями IRQ-оповещения

Текущее состояние	Следующее состояние	Переход	Описание
—	idle	1	Любой сброс, включая системный сброс (system reset), или отдельное событие оповещения, или защищенный сброс элемента блокировки (secure gate reset), безусловно переводит конечный автомат в состояние ожидания (idle state).
idle	idle	2	До тех пор пока в уже заблокированный элемент блокировки (gate) не будет записано необходимое значение блокировки (lock value) от соответствующего процессора, конечный автомат находится в состоянии ожидания (idle state).
idle	wait4_cp1_unlock	3	При попытке записи значения “cp0_lock”, инициализированной процессором Processor 0 в случае, когда элемент блокировки (gate) уже заблокирован процессором cp1, конечный автомат переходит в это состояние и ожидает, пока процессор cp1 разблокирует элемент блокировки (gate).
idle	wait4_cp0_unlock	4	При попытке записи значения “cp1_lock”, инициализированной процессором Processor 1 в случае, когда элемент блокировки (gate) уже заблокирован процессором cp0, конечный автомат переходит в это состояние и ожидает, пока процессор cp0 разблокирует элемент блокировки (gate).
wait4_cp1_unlock	wait4_cp1_unlock	5	После перехода в это состояние конечный автомат остается в нем до тех пор, пока элемент блокировки (gate) не будет разблокирован.
wait4_cp1_unlock	wait4_cp0_lock	6	Из этого состояния конечный автомат переходит в следующее состояние, ожидая, пока процессор cp0 заблокирует элемент блокировки (gate), после того как он был разблокирован.
wait4_cp0_unlock	wait4_cp0_unlock	7	После перехода в это состояние конечный автомат остается в нем до тех пор, пока элемент блокировки (gate) не будет разблокирован.
wait4_cp0_unlock	wait4_cp1_lock	8	Из этого состояния конечный автомат переходит в следующее состояние, ожидая, пока процессор cp1 заблокирует элемент блокировки (gate), после того как он был разблокирован.
wait4_cp0_lock	wait4_cp0_lock	9	В этом состоянии конечный автомат генерирует прерывание оповещения (notification interrupt), если оно должным образом разрешено, и остается в нем до тех пор, пока элемент блокировки (gate) не будет заблокирован процессором Processor 0 или повторно заблокирован процессором Processor 1.
wait4_cp0_lock	wait4_cp1_unlock	10	В этом состоянии конечный автомат генерирует прерывание оповещения (notification interrupt), если оно должным образом разрешено, и осуществляет переход, если элемент блокировки (gate) повторно блокируется процессором Processor 1. При этом переходе запрос на прерывание оповещения (notification interrupt request) отвергается.
wait4_cp0_lock	idle	11	В этом состоянии конечный автомат генерирует прерывание оповещения (notification interrupt), если оно должным образом разрешено, и осуществляет переход, если элемент блокировки (gate) окончательно заблокирован процессором Processor 0. При этом переходе прерывание оповещения отвергается.
wait4_cp1_lock	wait4_cp1_lock	12	В этом состоянии конечный автомат генерирует прерывание оповещения (notification interrupt), если оно должным образом разрешено, и остается в нем до тех пор, пока элемент блокировки (gate) не будет заблокирован процессором Processor 1 или повторно заблокирован процессором Processor 0.
wait4_cp1_lock	wait4_cp0_unlock	13	В этом состоянии конечный автомат генерирует прерывание оповещения (notification interrupt), если оно должным образом разрешено, и осуществляет переход, если элемент блокировки (gate) оказывается повторно заблокирован процессором Processor 0. При этом переходе запрос на прерывание оповещения (notification interrupt) отвергается.
wait4_cp1_lock	idle	14	В этом состоянии конечный автомат генерирует прерывание оповещения (notification interrupt), если оно должным образом разрешено, и осуществляет переход, если элемент блокировки (gate) окончательно блокируется процессором Processor 1. При этом переходе запрос на прерывание оповещения (notification interrupt) отвергается.

собственные инструкции для перемещения логического номера процессора в регистр общего назначения, например:

```
rdcpn rx
```

для инструкции «прочитать номер CPU» (“read CPU number”).

Если используются дополнительные механизмы оповещения прерываниями о неудачной блокировке (failed lock IRQ notification mechanism), то необходим доступ к соответствующим регистрам (SEMA4_CPnINE, SEMA4_CPnNTF). Следует отметить, что не требуется отклонять прерывание оповещения о сбое при записи блокировки (failed lock write notification interrupt), поскольку запрос автоматически отклоняется модулем управления блокировками (Semaphores module), когда элемент блокировки (gate) полностью заблокирован «отказавшим» процессором.

Итак, в случае если состояние системы требует программно управляемого сброса элемента блокировки (gate) или регистров IRQ-оповещения, необходим доступ к регистрам управления защищенным сбросом (SEMA4_RSTGT, SEMA4_RSTNTF). В таком случае рекомендуется, чтобы соответствующие биты разрешения IRQ-оповещения (SEMA4_CPnINE) были отключены **перед**

инициализацией последовательности защищенного сброса с двойной записью для предотвращения любых условий возникновения ложных запросов на прерывания оповещения (notification interrupt requests).

Перейдем к описанию карты распределения памяти и краткому описанию регистров модуля управления блокировками.

Карта распределения памяти и описание регистров модуля управления блокировками

Модуль управления блокировками (Semaphores module) предлагает модель программирования IPS, отображенную на вунтриплатформенное адресное пространство размером 16 кбайт в стандарте SPP. Далее будет описана двухъядерная конфигурация с 16 элементами блокировки (gate). Все имена регистров имеют префиксы Sema4 в качестве сокращения от полного названия модуля.

Модель программирования включает 8-, 16- и 32-битный доступ. При операции чтения можно использовать любой допустимый размер данных, в то время как операция записи в общем случае ограничена размером регистра.

Исключения из ограничений на размер записываемых данных подробно изложены в описании отдельных регистров. Попытки

Таблица 3. Карта распределения памяти модуля SEMA4

Абсолютный адрес, hex	Наименование регистра	Размер, бит	Тип доступа	Значение после сброса
229_0000	Semaphores Gate 0 Register (SEMA4_Gate00)	8	R/W	00h
229_0001	Semaphores Gate 1 Register (SEMA4_Gate01)	8	R/W	00h
229_0002	Semaphores Gate 2 Register (SEMA4_Gate02)	8	R/W	00h
229_0003	Semaphores Gate 3 Register (SEMA4_Gate03)	8	R/W	00h
229_0004	Semaphores Gate 4 Register (SEMA4_Gate04)	8	R/W	00h
229_0005	Semaphores Gate 5 Register (SEMA4_Gate05)	8	R/W	00h
229_0006	Semaphores Gate 6 Register (SEMA4_Gate06)	8	R/W	00h
229_0007	Semaphores Gate 7 Register (SEMA4_Gate07)	8	R/W	00h
229_0008	Semaphores Gate 8 Register (SEMA4_Gate08)	8	R/W	00h
229_0009	Semaphores Gate 9 Register (SEMA4_Gate09)	8	R/W	00h
229_000A	Semaphores Gate 10 Register (SEMA4_Gate10)	8	R/W	00h
229_000B	Semaphores Gate 11 Register (SEMA4_Gate11)	8	R/W	00h
229_000C	Semaphores Gate 12 Register (SEMA4_Gate12)	8	R/W	00h
229_000D	Semaphores Gate 13 Register (SEMA4_Gate13)	8	R/W	00h
229_000E	Semaphores Gate 14 Register (SEMA4_Gate14)	8	R/W	00h
229_000F	Semaphores Gate 15 Register (SEMA4_Gate15)	8	R/W	00h
229_0040	Semaphores Processor n IRQ Notification Enable (SEMA4_CP0INE)	16	R/W	0000h
229_0048	Semaphores Processor n IRQ Notification Enable (SEMA4_CP1INE)	16	R/W	0000h
229_0080	Semaphores Processor n IRQ Notification (SEMA4_CP0NTF)	16	R	0000h
229_0088	Semaphores Processor n IRQ Notification (SEMA4_CP1NTF)	16	R	0000h
229_0100	Semaphores (Secure) Reset Gate n (SEMA4_RSTGT)	16	R/W	0000h
229_0104	Semaphores (Secure) Reset IRQ Notification (SEMA4_RSTNTF)	16	R/W	0000h

доступа с использованием непредусмотренных размеров данных, по неопределенным (зарезервированным) адресам или с неподдерживаемым типом доступа (например, запись в регистры, доступные только для чтения, — read-only register) приводят к завершению операции с ошибкой IPS.

Итак, выделенного для модели программирования адресного пространства достаточно для определения до 64 элементов блокировки (gate) и до 8 процессорных ядер, даже несмотря на то, что такое определение значительно больше, чем любая планируемая в настоящее время реализация модуля. Количество элементов блокировки (gate) и поддерживаемых процессорных ядер являются величинами независимыми. Между этими двумя системными значениями не существует никакой связи. Карта распределе-

ния 16 кбайт памяти для модели программирования модуля управления блокировками приведена в таблице 3.

Перейдем к описанию каждой из категорий регистров, приведенных в таблице 3.

Регистры элементов блокировки Gate n (SEMA4_GATEn, n = 0...15)

В процессоре имеется 16 элементов блокировки (SEMA4_Gate0... SEMA4_Gate15). Каждый элемент блокировки (gate) реализован в виде 2-битного конечного автомата (finite state machine) с выравниванием вправо в структуре байта данных. Для проверки допустимости всех попыток записи аппаратное обеспечение использует номер ведущего на шине (bus master) совместно с шаблоном данных. Только ведущие на шине (bus master) процессора могут изменять состоя-

ние регистров элементов блокировки (gate registers). После блокировки элемент блокировки (gate) может (и должен) быть открыт (разблокирован) блокировавшим его ранее процессорным ядром.

Значения нескольких элементов блокировки (gate) могут быть прочитаны в едином сеансе доступа, но в процессе выполнения одной операции записи может быть обновлено состояние только единственного элемента блокировки (gate). Допустима 16- и 32-битная запись в несколько элементов блокировки (gate), но операнд записываемых данных должен обновлять состояние только одного элемента блокировки (gate). Значение 0x03 байта записываемых данных определено как «нет операции» и не влияет на состояние соответствующего регистра элемента блокировки (gate). Попытка записи значений для нескольких элементов блокировки (gate) в едином цикле доступа с размером данных, превышающим 8 бит (1 байт), приводит к завершению операции по ошибке и не позволяет измениться состоянию ни одного из элементов блокировки (gate). Структура регистра SEMA4_GATEn изображена на рис. 4, а описание полей регистра SEMA4_GATEn приведено в таблице 4.

Регистры разрешения IRQ-оповещения процессора Processor n от элементов блокировки (SEMA4_CPnINE, n = 0...1)

Применение аппаратного модуля элементов блокировки (Semaphores module) предоставляет возможность реализации полезных функций системного уровня. В качестве примера приведем дополнительный механизм генерации прерываний процессора после неудачной попытки блокировки. Вспомним, что традиционные функции поддержки программного элемента блокировки (gate) содержат бесконечный цикл ожидания изменения его состояния. При использовании рассматриваемого модуля процессор, который произвел неудачную попытку блокировки, может продолжать выполнять другие задачи, а оповещение должным образом сконфигурированным прерыванием вернет его к повторной попытке блокировки.

Дополнительная функция оповещения прерыванием включает в свою реализацию по два регистра для каждого процессора: регистр разрешения оповещения прерыванием (interrupt notification enable register, SEMA4_CPnINE) и регистр запросов на прерывание (interrupt request register, SEMA4_CPnNTF). Для поддержки реализации более чем 16 элементов

Таблица 4. Описание полей регистра SEMA4_GATEn

Поле	Описание
0–5 Reserved	Зарезервировано. Это доступное только для чтения (read-only) поле зарезервировано и при чтении всегда возвращает значение 0.
6–7 GTFSM	Конечный автомат элемента блокировки (Gate Finite State Machine). Аппаратный элемент блокировки (gate) поддерживается реализацией конечного автомата с тремя состояниями*: 1. Разблокирован (unlocked). 2. Заблокирован процессором Processor 0. 3. Заблокирован процессором Processor 1. Возможные значения поля: 00 Элемент блокировки (gate) разблокирован (unlocked), то есть свободен. 01 Элемент блокировки (gate) заблокирован процессором Processor 0. 10 Элемент блокировки (gate) заблокирован процессором Processor 1. 11 Состояние не используется и поэтому зарезервировано. Попытка записи значения 0x03 рассматривается как «нет операции» и не влияет на конечный автомат (finite state machine) элемента блокировки (gate).

Примечание. * Состояние элемента блокировки (gate) отражает, какой из процессоров последним его заблокировал, что может оказаться полезным при отладке системы.

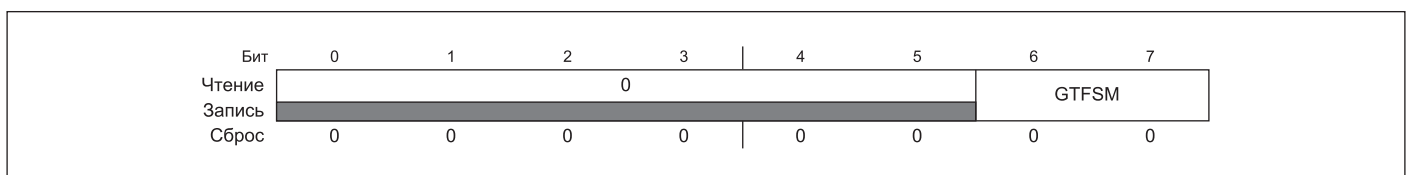


Рис. 4. Структура регистра SEMA4_GATEn

Бит	15	14	13	12	11	10	9	8
Чтение	INE8	INE9	INE10	INE11	INE12	INE13	INE14	INE15
Запись								
Сброс	0	0	0	0	0	0	0	0
Бит	7	6	5	4	3	2	1	0
Чтение	INE0	INE1	INE2	INE3	INE4	INE5	INE6	INE7
Запись								
Сброс	0	0	0	0	0	0	0	0

Рис. 5. Структура регистра SEMA4_CPnINE

блокировки (gate) эти регистры могут быть с 16- или 32-битным доступом. Для регистров SEMA4_CPnINE при чтении неиспользуемых битов возвращается нулевое значение,

а операции записи в них игнорируются. Структура регистра SEMA4_CPnINE изображена на рис. 5, а описание полей регистра SEMA4_CPnINE приведено в таблице 5.

Таблица 5. Описание полей регистра SEMA4_CPnINE

Поле	Описание
15 INE8	Разрешить оповещение запросом на прерывание 8 (Interrupt Request Notification Enable 8). Это битовое поле разрешает генерацию оповещения прерыванием при неудачной попытке заблокировать элемент блокировки gate 8. Возможные значения: 0 Генерация оповещения прерыванием запрещена. 1 Генерация оповещения прерыванием.
14 INE9	Разрешить оповещение запросом на прерывание 9 (Interrupt Request Notification Enable 9). Это битовое поле разрешает генерацию оповещения прерыванием при неудачной попытке заблокировать элемент блокировки gate 9. Возможные значения: 0 Генерация оповещения прерыванием запрещена. 1 Генерация оповещения прерыванием.
13 INE10	Разрешить оповещение запросом на прерывание 10 (Interrupt Request Notification Enable 10). Это битовое поле разрешает генерацию оповещения прерыванием при неудачной попытке заблокировать элемент блокировки gate 10. Возможные значения: 0 Генерация оповещения прерыванием запрещена. 1 Генерация оповещения прерыванием.
12 INE11	Разрешить оповещение запросом на прерывание 11 (Interrupt Request Notification Enable 11). Это битовое поле разрешает генерацию оповещения прерыванием при неудачной попытке заблокировать элемент блокировки 11. Возможные значения: 0 Генерация оповещения прерыванием запрещена. 1 Генерация оповещения прерыванием.
11 INE12	Разрешить оповещение запросом на прерывание 12 (Interrupt Request Notification Enable 12). Это битовое поле разрешает генерацию оповещения прерыванием при неудачной попытке заблокировать элемент блокировки gate 12. Возможные значения: 0 Генерация оповещения прерыванием запрещена. 1 Генерация оповещения прерыванием.
10 INE13	Разрешить оповещение запросом на прерывание 13 (Interrupt Request Notification Enable 13). Это битовое поле разрешает генерацию оповещения прерыванием при неудачной попытке заблокировать элемент блокировки gate 13. Возможные значения: 0 Генерация оповещения прерыванием запрещена. 1 Генерация оповещения прерыванием.
9 INE14	Разрешить оповещение запросом на прерывание 14 (Interrupt Request Notification Enable 14). Это битовое поле разрешает генерацию оповещения прерыванием при неудачной попытке заблокировать элемент блокировки gate 14. Возможные значения: 0 Генерация оповещения прерыванием запрещена. 1 Генерация оповещения прерыванием.
8 INE15	Разрешить оповещение запросом на прерывание 15 (Interrupt Request Notification Enable 15). Это битовое поле разрешает генерацию оповещения прерыванием при неудачной попытке заблокировать элемент блокировки gate 15. Возможные значения: 0 Генерация оповещения прерыванием запрещена. 1 Генерация оповещения прерыванием.
7 INE0	Разрешить оповещение запросом на прерывание 0 (Interrupt Request Notification Enable 0). Это битовое поле разрешает генерацию оповещения прерыванием при неудачной попытке заблокировать элемент блокировки gate 0. Возможные значения: 0 Генерация оповещения прерыванием запрещена. 1 Генерация оповещения прерыванием.
6 INE1	Разрешить оповещение запросом на прерывание 1 (Interrupt Request Notification Enable 1). Это битовое поле разрешает генерацию оповещения прерыванием при неудачной попытке заблокировать элемент блокировки gate 1. Возможные значения: 0 Генерация оповещения прерыванием запрещена. 1 Генерация оповещения прерыванием.
5 INE2	Разрешить оповещение запросом на прерывание 2 (Interrupt Request Notification Enable 2). Это битовое поле разрешает генерацию оповещения прерыванием при неудачной попытке заблокировать элемент блокировки gate 2. Возможные значения: 0 Генерация оповещения прерыванием запрещена. 1 Генерация оповещения прерыванием.
4 INE3	Разрешить оповещение запросом на прерывание 3 (Interrupt Request Notification Enable 3). Это битовое поле разрешает генерацию оповещения прерыванием при неудачной попытке заблокировать элемент блокировки gate 3. Возможные значения: 0 Генерация оповещения прерыванием запрещена. 1 Генерация оповещения прерыванием.
3 INE4	Разрешить оповещение запросом на прерывание 4 (Interrupt Request Notification Enable 4). Это битовое поле разрешает генерацию оповещения прерыванием при неудачной попытке заблокировать элемент блокировки gate 4. Возможные значения: 0 Генерация оповещения прерыванием запрещена. 1 Генерация оповещения прерыванием.
2 INE5	Разрешить оповещение запросом на прерывание 5 (Interrupt Request Notification Enable 5). Это битовое поле разрешает генерацию оповещения прерыванием при неудачной попытке заблокировать элемент блокировки gate 5. Возможные значения: 0 Генерация оповещения прерыванием запрещена. 1 Генерация оповещения прерыванием.
1 INE6	Разрешить оповещение запросом на прерывание 6 (Interrupt Request Notification Enable 6). Это битовое поле разрешает генерацию оповещения прерыванием при неудачной попытке заблокировать элемент блокировки gate 6. Возможные значения: 0 Генерация оповещения прерыванием запрещена. 1 Генерация оповещения прерыванием.
0 INE7	Разрешить оповещение запросом на прерывание 7 (Interrupt Request Notification Enable 7). Это битовое поле разрешает генерацию оповещения прерыванием при неудачной попытке заблокировать элемент блокировки gate 7. Возможные значения: 0 Генерация оповещения прерыванием запрещена. 1 Генерация оповещения прерыванием.

Регистры IRQ-оповещения процессора Processor n от модуля управления блокировками (SEMA4_CPnNTF, n = 0...1)

Модуль управления блокировками (Semaphores module) дополнительно позволяет процессору, который произвел неудачную попытку блокировки, продолжить выполнение других задач и с помощью оповещения должным образом сконфигурированным прерыванием вернуть его к повторной попытке блокировки вместо простейшего ожидания в бесконечном цикле.

Дополнительный механизм оповещения прерываниями содержит два регистра для каждого из процессоров: регистр разрешения оповещения прерыванием (SEMA4_CPnINE) и доступный только для чтения (read-only) регистр запросов на прерывания (SEMA4_CPnNTF). Для реализации поддержки более чем 16 элементов блокировки (gate) эти регистры могут быть с 16- или 32-битным доступом. Для регистров SEMA4_CPnNTF при чтении неиспользуемых битов возвращаются нулевые значения.

Прерывания оповещения (notification interrupt) генерируются уникальными конечными автоматами (finite state machine), по одному на каждый аппаратный элемент блокировки (gate). Эти конечные автоматы функционируют следующим образом:

1. При неудачной попытке блокировки конечный автомат (FSM) переходит в первое состояние, где он ожидает, пока элемент блокировки (gate) не будет разблокирован.
2. После того как элемент блокировки (gate) окажется разблокированным, конечный автомат (FSM) переходит во второе состояние, в котором генерируется запрос на прерывание (IRQ) к процессору, ранее осуществившему неудачную попытку блокировки.
3. Когда предпринявший ранее неудачную попытку блокировки процессор добивается успеха в блокировке элемента блокировки (gate), запрос на прерывание (IRQ) автоматически снимается и конечный автомат (FSM) возвращается в исходное состояние ожидания (idle state). Однако если

Бит	15	14	13	12	11	10	9	8
Чтение	GN8	GN9	GN10	GN11	GN12	GN13	GN14	GN15
Запись								
Сброс	0	0	0	0	0	0	0	0
Бит	7	6	5	4	3	2	1	0
Чтение	GN0	GN1	GN2	GN3	GN4	GN5	GN6	GN7
Запись								
Сброс	0	0	0	0	0	0	0	0

Рис. 6. Структура регистра SEMA4_CPnNTF

в этот промежуток времени другой процессор заблокирует элемент блокировки (gate), то конечный автомат (FSM) вернется в первое состояние, снимет запрос на прерывание и будет ожидать, пока элемент блокировки (gate) не будет разблокирован (новью).

Запросы на прерывания оповещения (notification interrupt request) реализованы как 3-битный конечный автомат с пятью состояниями и для программного обеспечения видимы как SEMA4_CP0NTF[GNn] и SEMA4_CP1NTF[GNn]. Структура регистра SEMA4_CPnNTF изображена на рис. 6, а описание полей регистра SEMA4_CPnNTF приведено в таблице 6.

Регистр (защищенного) сброса элемента блокировки Gate n (SEMA4_RSTGT)

Хотя идея аппаратной реализации элемента блокировки (gate) определяет протокол, в котором блокирующий процессор должен разблокировать элемент блокировки (gate), известно, что для системных операций может потребоваться функция повторной инициализации одного или нескольких элементов блокировки (gates) без необходимости выполнения сброса системного уровня (system-level reset).

Для поддержки этих специальных требований сброса элемента блокировки (gate) в модуле управления блокировками (Semaphores module) реализован механизм защищенного сброса, который позволяет инициализировать аппаратный элемент блокировки (gate) или сразу все элементы блокировки с помощью специального шаблона доступа с двойной записью. С использованием технологии, аналогичной программной обработке событий от сторожевого таймера (watchdog), при защищенном сбросе элемента блокировки (secure reset) необходимы две последовательные записи одним и тем же процессором предопределенных шаблонов данных для принудительной очистки выбранных эле-

Таблица 6. Описание полей регистра SEMA4_CPnNTF

Поле	Описание
15 GN8	Оповещение от элемента блокировки 8 (Gate 8 Notification). Это доступное только для чтения (read-only) битовое поле является битовой картой оповещения запросом на прерывание (interrupt request notification) при неудачной попытке блокировки элемента блокировки gate 8.
14 GN9	Оповещение от элемента блокировки 9 (Gate 9 Notification). Это доступное только для чтения (read-only) битовое поле является битовой картой оповещения запросом на прерывание (interrupt request notification) при неудачной попытке блокировки элемента блокировки gate 9.
13 GN10	Оповещение от элемента блокировки 10 (Gate 10 Notification). Это доступное только для чтения (read-only) битовое поле является битовой картой оповещения запросом на прерывание (interrupt request notification) при неудачной попытке блокировки элемента блокировки gate 10.
12 GN11	Оповещение от элемента блокировки 11 (Gate 11 Notification). Это доступное только для чтения (read-only) битовое поле является битовой картой оповещения запросом на прерывание (interrupt request notification) при неудачной попытке блокировки элемента блокировки gate 11.
11 GN12	Оповещение от элемента блокировки 12 (Gate 12 Notification). Это доступное только для чтения (read-only) битовое поле является битовой картой оповещения запросом на прерывание (interrupt request notification) при неудачной попытке блокировки элемента блокировки gate 12.
10 GN13	Оповещение от элемента блокировки 13 (Gate 13 Notification). Это доступное только для чтения (read-only) битовое поле является битовой картой оповещения запросом на прерывание (interrupt request notification) при неудачной попытке блокировки элемента блокировки gate 13.
9 GN14	Оповещение от элемента блокировки 14 (Gate 14 Notification). Это доступное только для чтения (read-only) битовое поле является битовой картой оповещения запросом на прерывание (interrupt request notification) при неудачной попытке блокировки элемента блокировки gate 14.
8 GN15	Оповещение от элемента блокировки 15 (Gate 15 Notification). Это доступное только для чтения (read-only) битовое поле является битовой картой оповещения запросом на прерывание (interrupt request notification) при неудачной попытке блокировки элемента блокировки gate 15.
7 GN0	Оповещение от элемента блокировки 0 (Gate 0 Notification). Это доступное только для чтения (read-only) битовое поле является битовой картой оповещения запросом на прерывание (interrupt request notification) при неудачной попытке блокировки элемента блокировки gate 0.
6 GN1	Оповещение от элемента блокировки 1 (Gate 1 Notification). Это доступное только для чтения (read-only) битовое поле является битовой картой оповещения запросом на прерывание (interrupt request notification) при неудачной попытке блокировки элемента блокировки gate 1.
5 GN2	Оповещение от элемента блокировки 2 (Gate 2 Notification). Это доступное только для чтения (read-only) битовое поле является битовой картой оповещения запросом на прерывание (interrupt request notification) при неудачной попытке блокировки элемента блокировки gate 2.
4 GN3	Оповещение от элемента блокировки 3 (Gate 3 Notification). Это доступное только для чтения (read-only) битовое поле является битовой картой оповещения запросом на прерывание (interrupt request notification) при неудачной попытке блокировки элемента блокировки gate 3.
3 GN4	Оповещение от элемента блокировки 4 (Gate 4 Notification). Это доступное только для чтения (read-only) битовое поле является битовой картой оповещения запросом на прерывание (interrupt request notification) при неудачной попытке блокировки элемента блокировки gate 4.
2 GN5	Оповещение от элемента блокировки 5 (Gate 5 Notification). Это доступное только для чтения (read-only) битовое поле является битовой картой оповещения запросом на прерывание (interrupt request notification) при неудачной попытке блокировки элемента блокировки gate 5.
1 GN6	Оповещение от элемента блокировки 6 (Gate 6 Notification). Это доступное только для чтения (read-only) битовое поле является битовой картой оповещения запросом на прерывание (interrupt request notification) при неудачной попытке блокировки элемента блокировки gate 6.
0 GN7	Оповещение от элемента блокировки 7 (Gate 7 Notification). Это доступное только для чтения (read-only) битовое поле является битовой картой оповещения запросом на прерывание (interrupt request notification) при неудачной попытке блокировки элемента блокировки gate 7.

ментов блокировки (gate). Для выполнения операции необходимы следующие шаблоны:

1. Процессор выполняет запись 16-битных данных в область памяти SEMA4_RSTGT. Для этой операции значение старшего байта (SEMA4_RSTGT[RSTGDP]) должно быть равно 0xe2; а значение младшего байта — “don't care”.
2. Затем тот же самый процессор выполняет запись 16-битных данных в область памяти SEMA4_RSTGT. При этой записи значение верхнего байта (SEMA4_RSTGT[RSTGDP]) является логическим дополнением первого шаблона данных (0x1d), а значение нижнего байта (SEMA4_RSTGT[RSTGTN]) задает, какие из элементов блокировки (gate) будут сброшены. В этом поле элементов блокировки (gate field) можно задать очистку либо одиночного элемента блокировки

(gate), либо всех элементов блокировки (gate).

3. При чтении области памяти SEMA4_RSTGT возвращается информация о состоянии 2-битного конечного автомата (finite state machine) SEMA4_RSTGT[RSTGSM], который реализует данную функцию, логический номер ведущего на шине (bus master), выполняющего сброс, SEMA4_RSTGT[RSTGMS] и номера последних очищенных элементов блокировки (gate number), SEMA4_RSTGT[RSTGTN]. Чтение регистра SEMA4_RSTGT никоим образом не влияет на состояние конечного автомата для защищенного сброса (secure reset finite state machine).

Структура регистра SEMA4_RSTGT изображена на рис. 7, а описание полей регистра SEMA4_RSTGT приведено в таблице 7.

Бит	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Чтение	RSTGTN								RSTGSM_RSTGMS_RSTGDP							
Запись																
Сброс	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Рис. 7. Структура регистра SEMA4_RSTGT

Таблица 7. Описание полей регистра SEMA4_RSTGT

Поле	Тип доступа	Подполе	Описание
15–8 RSTGTN			Номер сбрасываемого элемента блокировки (Reset Gate Number). Это 8-битное поле определяет, какой из аппаратных элементов блокировки (gate) будет сброшен. Значение поля можно обновить повторной записью. Если RSTGTN < 64, то сброшен будет единственный элемент блокировки (gate), задаваемый значением RSTGTN, иначе будут сброшены все элементы блокировки (gate). Соответствующие защищенные конечные автоматы (finite state machine) уведомлений с помощью IRQ также сбрасываются.
RSTGSM_RSTGMS_RSTGDP*	Read-Only	7–6 Reserved	Зарезервировано. При считывании всегда возвращает 0.
		5–4 RSTGSM	Сброс конечного автомата элемента блокировки (Reset Gate Finite State Machine). При чтении регистра SEMA4_RSTGT возвращается закодированное значение состояния конечного автомата (state machine value). Конечный автомат сброса (reset state machine) реализован как 2-битный, с тремя активными состояниями, заданными как: 00 Ожидание (Idle). Ожидание записи первого шаблона данных. 01 Ожидание записи второго шаблона данных. 10 Последовательность из двух записей завершена. Генерируются сбросы заданных элементов блокировки (gate). После выполнения сброса конечный автомат возвращается в состояние idle (ожидая запись первого шаблона данных). Следует отметить, что состояние RSTGSM = 0b10 действительно только для одного такта конечного автомата, поэтому операция чтения не может вернуть это значение. 11 Это состояние не используется и поэтому зарезервировано.
		3 Reserved	Зарезервировано. При считывании всегда возвращает 0.
		2–0 RSTGMS	Ведущий на шине, выполнивший сброс элемента блокировки (Reset Gate Bus Master). В этом 3-битном, доступном только для чтения (read-only) поле записан логический номер ведущего на шине (bus master), выполнившего сброс элемента блокировки (gate reset). Функция сброса необходима для завершения операции, при которой одним и тем же ведущим на шине (bus master) инициализируются две последовательные записи в этот регистр. Поле обновляется всякий раз, когда осуществляется запись в регистр. Соответствие между номерами портов ведущего на системной шине (system bus master), связанного с ним ведущего устройства (bus master device) и логическим номером процессора зависит от типа ЧИК (SoC).
Write-Only	7–0 RSTGDP	Шаблон данных для сброса элемента блокировки (Reset Gate Data Pattern). К этому доступному только для записи (write-only) полю обращаются за две последовательные операции записи со специальными шаблонами данных для запуска механизма сброса элемента блокировки (gate). Вначале записывается RSTGDP = 0xe2, а затем — RSTGDP = 0x1d.	

Примечание. *Это поле содержит подполя, которые могут изменяться в зависимости от того, производится ли операция чтения или записи. Подполя, для которых указан доступ только на чтение (read-only), могут использоваться только в операциях чтения. Подполя, для которых указан доступ только на запись (write-only), могут использоваться лишь в операциях записи. Нумерация битов в описании начинается со старшего бита (most significant bit), который имеет номер 0.

Таблица 8. Описание полей регистра SEMA4_RSTNTF

Поле	Тип доступа	Подполе	Описание
15–8 RSTNTN			Номер сбрасываемого оповещения (Reset Notification Number). Это 8-битное поле задает конечный автомат (state machine) оповещения с помощью IRQ, который будет сброшен. Поле обновляется во время второй операции записи. Если RSTNTN < 64, то сбрасывается одиночный конечный автомат оповещения с помощью IRQ, заданный значением RSTNTN, в противном случае сбрасываются все оповещения.
RSTNSM_RSTNMS_RSTNDP*	Read-Only	7–6 Reserved	Зарезервировано. При чтении всегда возвращает 0.
		5–4 RSTNSM	Сброс конечного автомата оповещения (Reset Notification Finite State Machine). При чтении регистра SEMA4_RSTNTF возвращается закодированное значение состояния конечного автомата (state machine value). Конечный автомат сброса (reset state machine) реализован как 2-битный, с тремя активными состояниями, заданными как: 00 Ожидание (Idle). Ожидание записи первого шаблона данных. 01 Ожидание записи второго шаблона данных. 10 После выполнения сброса конечный автомат возвращается в состояние idle (ожидая запись первого шаблона данных). Следует отметить, что состояние RSTNSM = 10 действительно только для одного такта конечного автомата, поэтому операция чтения не может вернуть это значение. 11 Это состояние не используется и поэтому зарезервировано.
		3 Reserved	Зарезервировано. При чтении всегда возвращает 0.
		2–0 RSTNMS	Ведущий на шине, выполнивший сброс оповещения (Reset Notification Bus Master). В этом 3-битном, доступном только для чтения (read-only) поле записан логический номер ведущего на шине (bus master), выполнившего сброс оповещения. Функция сброса необходима для завершения операции, при которой одним и тем же ведущим на шине (bus master) инициализируются две последовательные записи в этот регистр. Поле обновляется всякий раз, когда осуществляется запись в регистр. Соответствие между номерами портов ведущего на системной шине (system bus master), связанного с ним ведущего устройства (bus master device) и логическим номером процессора зависит от типа ЧИК (SoC).
Write-Only	7–0 RSTNDP	Шаблон данных для сброса оповещения (Reset Notification Data Pattern). К этому доступному только для записи (write-only) полю обращаются за две последовательные операции записи со специальными шаблонами данных для запуска механизма сброса оповещения (notification). Вначале записывается RSTNDP = 0x47, а затем — RSTNDP = 0xb8.	

Примечание. *Это поле содержит подполя, которые могут изменяться в зависимости от того, производится ли операция чтения или записи. Подполя, для которых указан доступ только на чтение (read-only), могут использоваться только в операциях чтения. Подполя, для которых указан доступ только на запись (write-only), могут использоваться только в операциях записи. Нумерация битов в описании начинается со старшего бита (most significant bit), который имеет номер 0.

Регистр (защищенного) сброса IRQ-оповещения модуля управления блокировками (SEMA4_RSTNTF)

Как и в случае функции защищенного сброса (secure reset) аппаратных элементов блокировки (gate), имеется понимание того, что для работы системы может потребоваться функция сброса для перенициализации состояния логики IRQ-оповещения без необходимости выполнения сброса системного уровня.

Для поддержки этих требований к сбросу оповещения в модуле управления блокировками (Semaphores module) реализован механизм защищенного сброса, который позволяет инициализировать оповещение с помощью IRQ (или все оповещение) посредством двукратной записи определенного шаблона доступа. При удачном завершении сбрасываются выбранные конечные автоматы (finite state machine) оповещения с помощью IRQ. С использованием технологии, аналогичной той, которая требуется для обслуживания программного сторожевого таймера (watchdog), для запуска механизма защищенного сброса (secure reset) понадобится две последовательные записи предопределенных шаблонов данных от одного и того же процессора, чтобы осуществить принудительную очистку оповещения с помощью IRQ.

Используются следующие шаблоны доступа:

1. Процессор выполняет запись 16-битного значения в область памяти SEMA4_RSTNTF. Старший байт (SEMA4_RSTNTF[RSTNDP]) должен иметь значение 0x47; а младший байт — значение «не обслуживается» (“don’t care”).
2. Затем тот же самый процессор должен выполнить запись второго 16-битного значения в область памяти SEMA4_RSTNTF. При этой записи верхний байт (SEMA4_RSTNTF[RSTNDP]) является логическим дополнением первого шаблона данных (0xb8), а нижний байт (SEMA4_RSTNTF[RSTNTN]) определяет, какие из оповещений будут сброшены. В этом поле можно задать сброс одного конкретного или всех оповещений.
3. Операция чтения области памяти SEMA4_RSTNTF возвращает информацию о состоянии 2-битного конечного автомата (finite state machine) SEMA4_RSTNTF[RSTNSM], который реализует эту функцию. Ведущий на шине (bus master) выполняет сброс (SEMA4_RSTNTF[RSTNMS]) и очищает номера последних уведомлений (SEMA4_RSTNTF[RSTNTN]). Чтение регистра

Бит	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Чтение	RSTNTN								RSTNSM_RSTNMS_RSTNDP							
Запись																
Сброс	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Рис. 8. Структура регистра SEMA4_RSTNTF

SEMA4_RSTNTF никоим образом не влияет на состояние конечного автомата защищенного сброса (secure reset finite state machine). Структура регистра изображена на рис. 8, а описание полей регистра SEMA4_RSTNTF приведено в таблице 8.

Заключение

Данная статья завершает описание аппаратной архитектуры модулей межпроцессорного обмена в гетерогенных многоядерных СнК, одним из ядер которых является процессор Cortex-M3/M4.

В публикации рассмотрен модуль управления блокировками (Semaphores module, SEMA4) СнК i.MX6SX. Модуль состоит из элементов блокировки (gate), а также имеет дополнительные функции оповещения с помощью прерываний, сообщающих об их состоянии. Это позволяет осуществившему неудачную блокировку процессору выполнять другие действия до поступления запроса на прерыва-

ние при снятии блокировки требуемого элемента блокировки (gate). Наряду с рассмотренным в предыдущей статье модулем обмена сообщениями (Messaging Unit, MU) модуль SEMA4 образует систему межпроцессорного взаимодействия в i.MX6SX и аналогичных по архитектуре СнК.

В следующей статье будут описаны практические аспекты работы с модулем управления блокировками SEMA4. ■

Литература

1. i.MX 6SoloX Applications Processor Reference Manual.
www.nxp.com/docs/en/referencemanual/IMX6SXRm.pdf
2. i.MX 7Solo Applications Processor Reference Manual.
www.nxp.com/docs/en/referencemanual/IMX7SRM.pdf
3. i.MX 7Dual Applications Processor Reference Manual.
www.nxp.com/docs/en/referencemanual/IMX7DRM.pdf