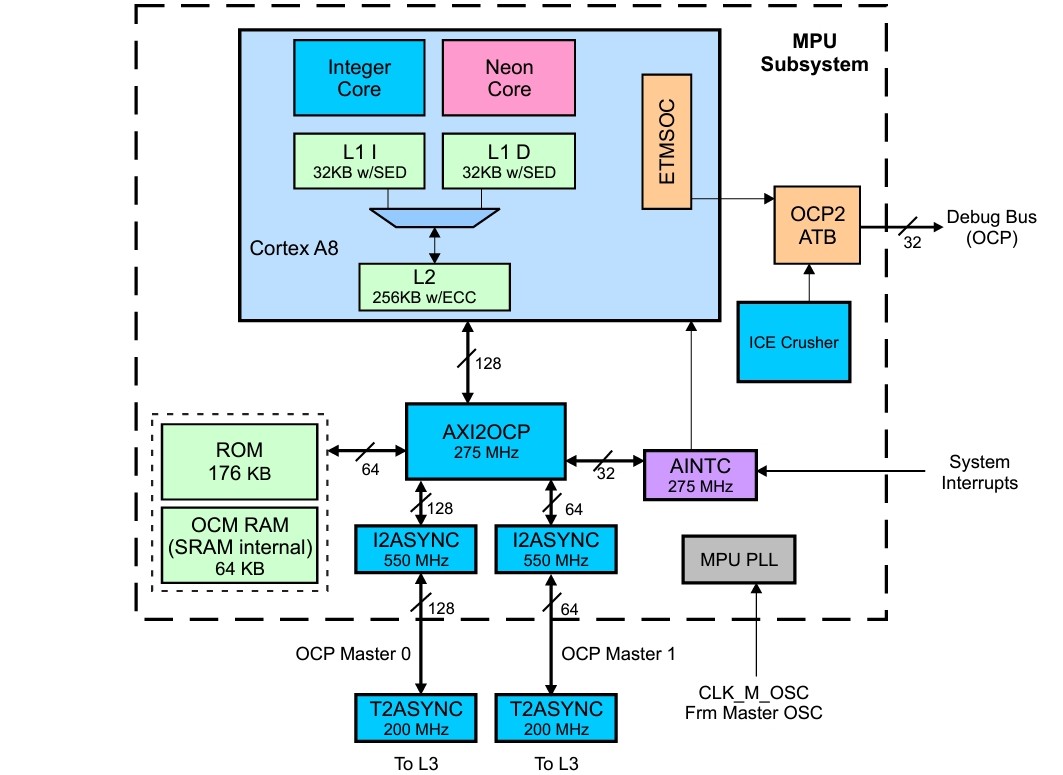
**3.1 ARMCortex-A8 подсистема MPU**

Подсистема микропроцессорного блока (MPU) устройства обрабатывает транзакции между ядром ARM (процессор ARM® Cortex™ -A8), соединением L3 и контроллером прерываний (INTC). Подсистема MPU представляет собой жесткий макрос, который интегрирует процессор ARM® Cortex™ -A8 с дополнительной логикой для преобразования протокола, эмуляции, обработки прерываний и улучшения отладки. Cortex™ -A8 - это совместимый с ARMv7, двойной по порядку механизм выполнения с интегрированными кэшами L1 и L2 с NEON™ SIMD Media Processing Unit. Контроллер прерываний включен в подсистему MPU для обработки запросов прерываний хоста в системе. Подсистема MPU включает логику, совместимую с CoreSight, чтобы разрешить подсистеме отладки доступ к CortexA8 ресурсам отладки и эмуляции, включая встроенный Macrocell трассировки. Подсистема MPU имеет три функциональных домена синхронизации, включая высокочастотный домен синхронизации, используемый Cortex™ -A8. Высокочастотная область изолирована от остальной системы асинхронными мостами. На рисунке 3-1 показана высокоуровневая блок-схема подсистемы MPU.

Рисунок 3-1. Подсистема микропроцессорного блока (МПУ)

**3.1.1 Характеристики**

В данном разделе описываются основные характеристики подсистемы MPU:

• ARM Microprocessor

- Cortex-A8

- ARM Architecture version 7 ISA.

- 2-issue, конвейер поточного выполнения.

- L1 кэш инструкций и данных 32KB, 4-way, 16-word line с 128-битным интерфейсом.

- Интегрированный кэш второго уровня 256 КБ, 8-way, 16-word line, 128-интерфейс к L1 вместе с EC Поддержка C/четности.

- Включает сопроцессор Neon Media (NEON™), который реализует расширенную архитектуру обработки мультимедиа SIMD.

- Включает сопроцессор VFP, который реализует архитектуру VFPv3 и полностью соответствует стандарту IEEE 754.

- Внешний интерфейс использует протокол AXI, настроенный на ширину 128-битных данных.

- Включает встроенный Macrocell трассировки. (ETM) поддержка неинвазивной отладки.

- Реализует отладку ARMv7 с регистрами контрольных точек и точек останова и 32-разрядной расширенной периферийной шиной. (APB) подчиненный интерфейс к системам отладки CoreSight. • AXI2OCP Bridge- Support OCP2.2.

- Single Request Multiple Data Protocol на двух портах.

- Несколько целей, включая три порта OCP. (128-разрядный, 64-разрядный и 32-разрядный).

• Контроллер прерываний

- Поддержка до 128 запросов прерываний

• Эмуляция/отладка - Совместимость с архитектурой CoreSight.

• Генерация тактовых импульсов - через PRCM

**3.1.2 Интеграция подсистемы МПУ**

Подсистема MPU объединяет следующую группу подмодулей:

**Процессор ARM® Cortex-A8** обеспечивает высокую производительность, включая технологию NEON для мобильного мультимедийного ускорения. ARM взаимодействует через шину AXI с мостом AXI2OCP и принимает прерывания от контроллера прерываний подсистемы MPU (MPU INTC).

**Контроллер прерываний:** Обрабатывает прерывания модуля (подробнее см. Главу 6, Прерывания).

**AXI2OCP мост:** Обеспечивает связь между ARM (AXI), INTC (OCP) и модулями(OCP L3).

**I2Async мост:** Это асинхронный мостовой интерфейс, обеспечивающий асинхронный OCP к OCP интерфейсу. Этот интерфейс находится между мостом AXI2OCP в подсистеме MPU и T2Async внешним мостом к подсистеме MPU.

**Генератор тактовых импульсов:** Обеспечивает необходимое разделенное тактирования для внутренних модулей подсистемы MPU и имеет тактовый вход от MPU\_CLK, который питается от управления питанием, сбросом и тактовым сигналом (PRCM) модуля устройства.

**Внутрисхемный эмулятор:** полностью совместим с архитектурой CoreSight и предоставляет возможности отладки.

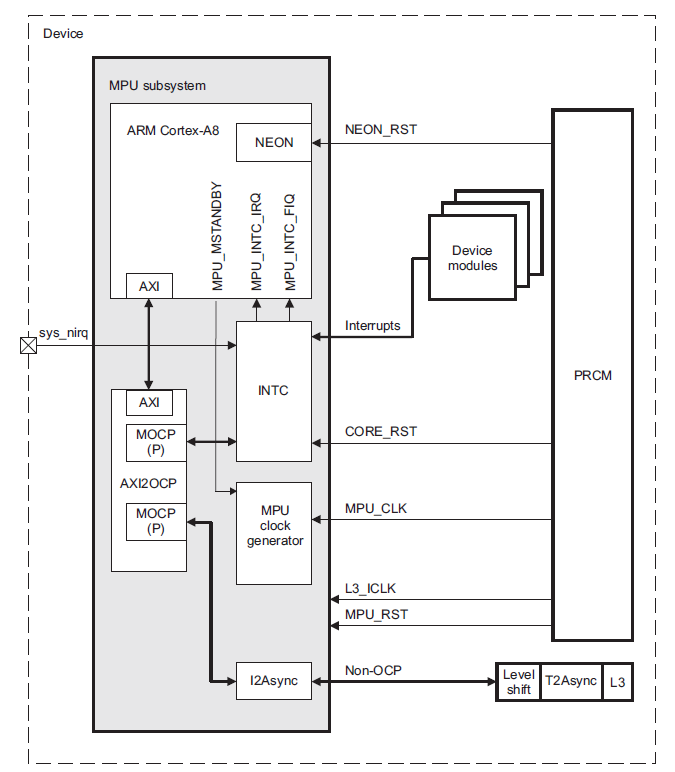


Рисунок 3-2. Сигнальный интерфейс подсистемы микропроцессорного блока (MPU)

**3.1.3 Распределение тактовых сигналов и сброса подсистемы MPU**

**3.1.3.1 Распределение синхросигналов**

Подсистема MPU включает в себя встроенную DPLL, которая обеспечивает синхронизацию для ARM Cortex-A8 процессор. Делитель синхросигнала в подсистеме используется для получения синхросигналов для других внутренних модулей.

Все основные модули внутри подсистемы MPU синхронизируются на половине частоты ядра ARM. Делитель выходного синхросигнала может быть запрограммирован с помощью CM\_DIV\_M2\_DPLL\_MPU.DPLL\_CLKOUT\_DIV поля регистров, частота относительно ядра ARM. Для получения дополнительной информации см. Главу 8, Питание, Сброс и Управление синхросигналами (PRCM).

Генератор синхросигналов генерирует следующие функциональные синхросигналы:

**ARM (ARM\_FCLK):** Это тактовые импульсы ядра. Это базовый быстрый синхросигнал, который направляется внутри в ARM логическая и внутренняя оперативная память, включая NEON, кэш-память второго уровня, ядро ETM (эмуляция) и ядро ARM.

**AXI2OCP Clock (AXI\_FCLK):** Эта частота вдвое меньше частоты ARM clock (ARM\_FCLK). Интерфейс OCP таким образом выполняется на половине частоты ARM.

**Функциональные тактовые импульсы контроллера прерываний (MPU\_INTC\_FCLK):** Эти такты, которые являются частью INTC , составляет половину частоты тактового сигнала ARM (ARM\_FCLK).

**Функциональные такты ICE-Crusher (ICECRUSHER\_FCLK):** ICE-Crusher clocking работает на APB интерфейсе с использованием синхронизации ядра ARM. Этот тактовый сигнал вдвое меньше частоты тактового сигнала ARM (ARM\_FCLK).

**I2Async Clock (I2ASYNC\_FCLK):** Эта частота вдвое меньше частоты ARM clock (ARM\_FCLK). Это соответствует интерфейсу OCP моста AXI2OCP.

ПРИМЕЧАНИЕ: Вторая половина асинхронного моста (T2ASYNC) синхронизируется непосредственно PRCM с тактами ядра. T2ASYNC не является частью подсистемы MPU.

**Тактовые импульсы эмуляции:** распределены модулем PRCM и асинхронны тактовому сигналу ядра ARM (ARM\_FCLK) и может работать максимум на 1/3 такта ядра ARM.

Таблица 3-1 суммирует синхросигналы, генерируемые в подсистеме MPU генератором синхросигналов MPU.

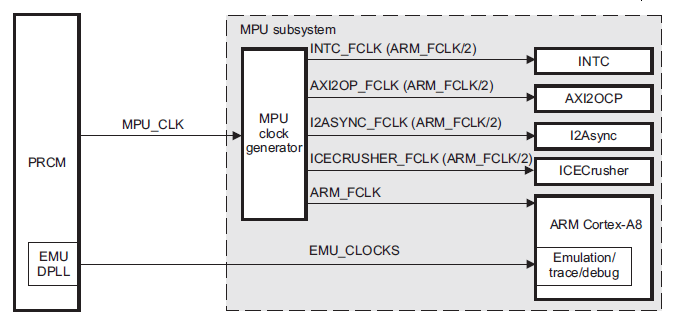


Рисунок 3-3. Схема синхронизации подсистемы MPU

Таблица 3-1. Тактовые частоты подсистемы MPU

|  |  |
| --- | --- |
| Тактовый сигнал | частота |
| Cortex A8 Core Functional Clock | MPU\_CLK |
| AXI2OCP Bridge Functional Clock | MPU\_CLK/2 |
| Device Clock | MPU\_CLK/2 |
| I2Async Bridge Functional Clock | MPU\_CLK/2 |

**3.1.3.2 Распределение сброса**

Сбросы подсистемы MPU обеспечивается PRCM и управляется модулем генератора тактовых сигналов.

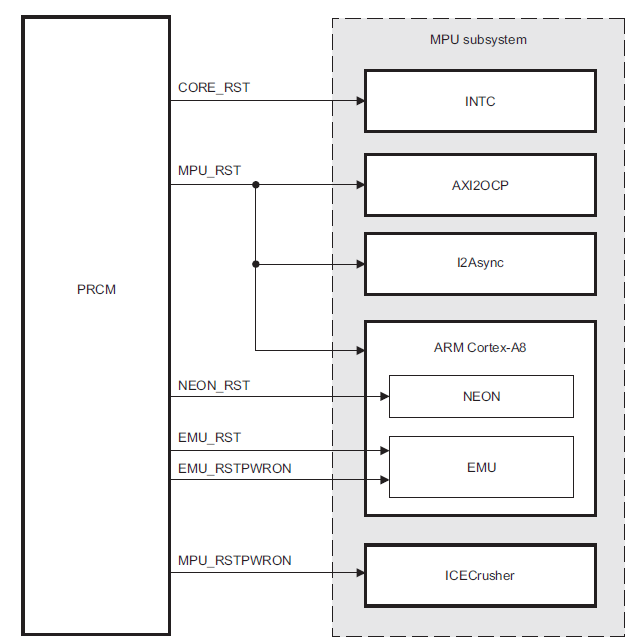


Рис. 3-4. Схема сброса подсистемы MPU

Таблица 3-2. Схема сброса подсистемы MPU

|  |  |  |
| --- | --- | --- |
| **Signal Name** | **I/O** | **Interface** |
| MPU\_RST | I | PRCM |
| NEON\_RST | I | PRCM |
| CORE\_RST | I | PRCM |
| MPU\_RSTPWRON | I | PRCM |
| EMU\_RST | I | PRCM |
| EMU\_RSTPWRON | I | PRCM |

**3.1.4 Подчип ARM**

**3.1.4.1 Обзор ARM**

Процессор ARM Cortex-A8 включает в себя технологии, доступные в архитектуре ARM7™. Эти технологии включают в себя NEON™ для обработки медиа и сигналов и Jazelle™ RCT для ускорения компиляторов реального времени, технология Thumb№ -2 для плотности кода и VFPv3 архитектура с плавающей запятой.

**3.1.4.2 Описание ARM**

**3.1.4.2.1 Инструкция ARM® Cortex™ -A8, данные и частный периферийный порт**

Интерфейс шины AXI является основным интерфейсом системной шины ARM. Он выполняет заполнение кэша второго уровня и не кэшируемых доступов как для инструкций, так и для данных. Интерфейс AXI поддерживает 128-разрядную и 64-разрядную разрядность входных и выходных данных шины. Он поддерживает несколько невыполненных запросов на шине AXI и широкий спектр отношений тактовых импульсов шины к тактовым сигналам ядра. Тактовый сигнал шины синхронизирован с тактовым сигналом ядра.

Полную модель программирования системы ARM ® Cortex™ -A8 можно найти в техническом справочном руководстве правил транзакций (заказ, проводка и синхронизация конвейера), которые применяются в зависимости от атрибутов области памяти, связанный с адресом назначения транзакции.

**3.1.4.2.2 Защищенные контрольные вызовы для доступа к регистрам CP15**

Устройство поддерживает специальные функции безопасного мониторинга, что позволяет получить доступ к определенным регистрам ядра ARM в привилегированном режиме. Функции записи в регистр вспомогательного управления, регистр незащищенного управления доступом, и вспомогательный регистр управления кэш-памятью второго уровня (см. Техническое справочное руководство по ARM для описание этих регистров).

|  |  |
| --- | --- |
| **Service ID (R12)** | **Description** |
| 0x100 | Write value in R0 to Auxiliary Control Register |
| 0x101 | Write value in R0 to Non Secure Access Control Register |
| 0x102 | Write value in R0 to L2 Cache Auxiliary Control Register |

В общем случае процедура использования этого защищенного вызова монитора следующая:

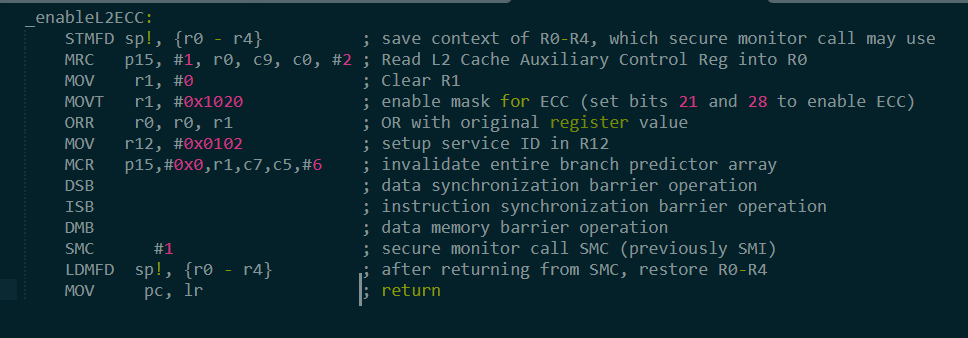
• Запишите соответствующий идентификатор услуги в R12.

• Загрузите R0 со значением для записи в основной регистр ARM.

• Выполнение заградительных работ для обеспечения надлежащего выполнения.

• Используйте SMC # 1 (или SMI # 1) для безопасного вызова монитора

Инструкции барьера также необходимы для обеспечения чистого состояния перед входом в безопасный монитор. Относиться к следующему примеру, который обеспечивает правильную кодовую последовательность. Этот код является примером включение ECC в кэше второго уровня. *Примечание.Эта функция должна выполняться в привилегированном режиме ARM.*



**3.1.4.2.3 Основные поддерживаемые функции ARM**

В таблице 3-3 приведен список основных функций ядра Cortex™ -A8, поддерживаемых подсистемой MPU.

Таблица 3-3. Поддерживаемые функции ARM Core

|  |  |
| --- | --- |
| **Features** | **Comments** |
| ARM version 7 ISA | Standard ARM instruction set + Thumb2™, JazelleX™ Java  accelerator, and Media extensions. |
| Backward compatible with previous ARM ISA versions. |
| L1 Icache and Dcache | 32 KB , 4-way, 16 word line, 128 bit interface. |
| L2 Cache | 256 KB, 8-way, 16 word line, 128 bit interface to L1, ECC/Parity is supported. The L2 cache and cache controller are embedded within the ARM core. |
| L2 valid bits cleared by software loop or by hardware. |
| TLB | Fully associative and separate ITLB with 32 entries and DTLB  with 32 entries. |
| CoreSight ETM | The CoreSight ETM is embedded with the ARM core. The 32KB buffer (ETB) exists at the Chip Level (DebugSS) |
| Branch Target Address Cache | 512 entries |
| Enhanced Memory Management Unit | Mapping sizes are 4KB, 64KB, 1MB, and 16MB. (ARM MMU  adds extended physical address ranges) |
| Neon | Gives greatly enhanced throughput for media workloads and  VFP-Lite support. |
| Flat Memories | 176 Kbytes of ROM |
| 64 Kbytes of RAM |
| Buses | 128 bit AXI internal bus from CortexA8 routed by an AXI2OCP  bridge to the interrupt controller, ROM, RAM, and 3  asynchronous OCP bridges (128 bits, and 64 bits) |
| Low interrupt latency | Closely coupled INTC to the ARM core with 128 interrupt lines |
| Vectored Interrupt Controller Port | Present. |
| JTAG based debug | Supported via DAP |
| Trace support | Supported via TPIU |
| External Coprocessor | Not supported |

**3.1.5 Контроллер прерываний**

Контроллер прерываний Host ARM (AINTC) отвечает за приоритизацию всех запросов на обслуживание nIRQ или nFIQ от периферийные устройства системы и генерирование для хоста. Тип прерывания (nIRQ или nFIQ) и приоритет входов прерываний являются программируемыми. Интерфейсы AINTC с ARM процессора через порт AXI через мост AXI2OCP и работает с половинной скоростью процессора. Он имеет возможность обработки до 128 запросов, которые могут быть направлены/приоритизированы как запросы на прерывание A8 nFIQ или nIRQ.

Общие характеристики AINTC:

• До 128 чувствительных к уровню прерываний на входах

• Индивидуальный приоритет для каждого входа прерывания

• Каждое прерывание может быть направлено к nFIQ или nIRQ

• Независимая сортировка приоритетов для nFIQ и nIRQ

**3.1.6 Управление питанием**

**3.1.6.1 Домены питания**

Подсистема MPU разделена на четыре области питания, управляемые PRCM, как показано на рис. 3-5.

*ПРИМЕЧАНИЕ: Домен эмуляции и основной домен не полностью встроены в подсистему MPU.*

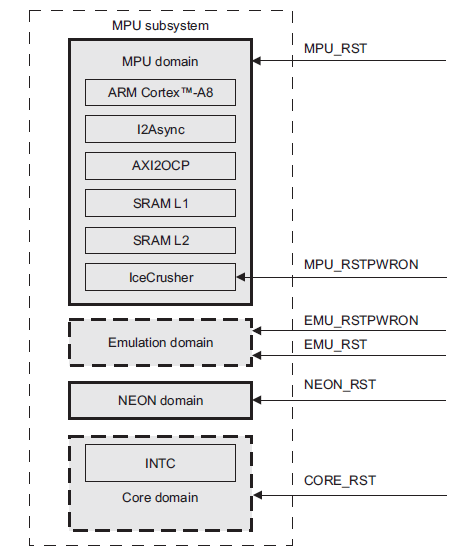


Рисунок 3-5. Обзор области питания подсистемы MPU

Требования к управлению питанием на уровне устройства определяют домены питания для подсистемы MPU. Домены питания уровня устройства непосредственно выровнены с доменами напряжения и, таким образом, могут быть представлены как перекрестная ссылка на различные домены напряжения.

В таблице 3-4 показаны различные домены питания подсистемы MPU и внутренних модулей.

Таблица 3-4. Обзор доменов питания подсистемы MPU

|  |  |
| --- | --- |
| **Functional Power Domain** | **Physical Power Domain per System/Module** |
| MPU subsystem domain | ARM, AXI2OCP, I2Asynch Bridge, ARM L1 and L2 periphery  logic and array, ICE-Crusher, ETM, APB modules |
| MPU NEON domain | ARM NEON accelerator |
| CORE domain | MPU interrupt controller |
| EMU domain | EMU (ETB,DAP) |

*ПРИМЕЧАНИЕ: Запоминающие устройства массивов L1 и L2 имеют отдельные сигналы управления в подсистеме MPU, таким образом непосредственно контролируется PRCM Для получения подробной информации о доменах физической мощности и напряжении , см. главу 8, «Управление питанием, сбросом и синхронизацией (PRCM)».*

**3.1.6.2 Состояния питания**

Каждый домен питания может управляться PRCM в 3 различных состояниях, в зависимости от функционального режима требуемого пользователем.

Для каждого домена питания PRCM управляет всеми переходами, управляя часами домена, сбросами домена, переключатели питания логики домена и переключатели питания памяти.

Таблица 3-5. Состояния питания MPU

|  |  |  |  |
| --- | --- | --- | --- |
| **Power State** | **Logic Power** | **Memory Power** | **Clocks** |
| Active | On | On or Off | On (at least one clock) |
| Inactive | On | On or Off | Off |
| Off | Off | Off | Off (all clocks) |

**3.1.6.3 Режимы питания**

Основная часть подсистемы MPU относится к домену питания MPU. Модули внутри силового домена могут быть выключены в то время, когда процессор ARM находится в выключенном или резервном режиме. Управление режимом ОЖИДАНИЯ/ПРОБУЖДЕНИЯ осуществляется блоком генератора синхросигналов, но инициируется модулем PRCM.

Состояние MPU Standby можно проверить с помощью CM\_MPU\_MPU\_CLKCTRL.STBYST. Для того, чтобы MPU был включен , питание ядра (называемого здесь ядром устройства) должно быть включено. Управление питанием устройства не разрешает INTC переходить в состояние OFF, когда домен MPU включен (активный или один из режимов хранения).

Ядро NEON имеет независимый режим выключения питания, когда не используется. Включение и выключение NEON может управляться программным обеспечением.

|  |
| --- |
| Внимание  Кэш-память первого уровня MPU не поддерживает режим хранения, и переключение ее массива управляется совместно с логикой MPU. Для соответствия требованиям L1  сигналы управления удержанием существуют на границе PRCM, но не используются.  ARM L2 может быть переведен в режим хранения независимо от других доменов. |

В таблице 3-6 приведены поддерживаемые режимы рабочего питания. Все остальные комбинации незаконны. ARM L2, NEON и ETM/Debug могут быть включены/выключены независимо. Столбец APB/ATB ETM/Debug относится ко всем трем функциям: эмуляция ARM, трассировка и отладка.

Подсистема MPU должна находиться в режиме питания, в котором домен питания MPU, домен питания NEON, debug power domain и INTC power domain находятся в состоянии ожидания или в выключенном состоянии.

Таблица 3-6. Режимы работы и мощности подсистемы МПУ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Mode** | **MPU and ARM Core Logic** | **ARM L2 RAM** | **NEON INTC** | **Device Core and ETM** | **APB/ATB Debug** |
| 1 | Active | Active | Active | Active | Disabled or enabled |
| 2 | Active | Active | OFF | Active | Disabled or enabled |
| 3 | Active | RET | Active | Active | Disabled or enabled |
| 4 | Active | RET | OFF | Active | Disabled or enabled |
| 5 | Active | OFF | Active | Active | Disabled or enabled |
| 6 | Active | OFF | OFF | Active | Disabled or enabled |
| 7 | OFF | RET | OFF | OFF | Disabled or enabled |
| 8 | Standby | Active | Standby | Active | Disabled or enabled |
| 9 | Standby | Active | OFF | Active | Disabled or enabled |
| 10 | Standby | RET | Standby | Active | Disabled or enabled |
| 11 | Standby | RET | OFF | Active | Disabled or enabled |
| 12 | Standby | OFF | Standby | Active | Disabled or enabled |
| 13 | Standby | OFF | OFF | Active | Disabled or enabled |
| 14 | OFF | OFF | OFF | OFF | Disabled or enabled |

**3.1.7 Модель программирования ARM**

Подробное описание регистров, используемых для конфигурирования MPU, см. в главе 8, управление «Питанием, сбросом и тактовым сигналом» (PRCM).

**3.1.7.1 Управление синхросигналами**

Для получения информации о настройках синхронизации см. главу 8, Питание, сброс и управление синхронизацией (PRCM).

**3.1.7.2 Переходы в режим питания MPU**

Следующие подразделы описывают переходы различных режимов мощности для области мощности МПУ:

• Основное питание при сбросе

• MPU в режим ожидания

• MPU вне режима ожидания

• Включение питания MPU из выключенного состояния

**3.1.7.2.1 Базовый сброс при включении питания**

Сброс при включении питания следует следующей последовательности операций и применим к первоначальному включению питания и пробуждение из режима выключения устройства:

Сбросьте модули INTC (CORE\_RST) и подсистемы MPU (MPU\_RST). Тактовые сигналы должны быть

активны во время сброса MPU и CORE.

**3.1.7.2.2 MPU в режиме ожидания**

MPU в режиме ожидания следует следующей последовательности операций и применим для начального включения питания и пробуждения из режима выключения устройства.

1. Ядро ARM инициирует переход в режим ожидания только через программное обеспечение (CP15 - WFI).

2. Модули MPU запрашиваются внутри подсистемы MPU для перехода в режим ожидания после обнаружения режима ожидания ядра ARM.

3. MPU находится в резервном выходе, утвержденном для PRCM (все выходы гарантированно имеют значения сброса).

4. PRCM теперь может запросить INTC перейти в режим ожидания. Подтверждение от INTC поступает в PRCM.

ПРИМЕЧАНИЕ: Выход INTC SWAKEUP является чисто аппаратным сигналом к PRCM для состояния его IDLE

запрос и квитирование подтверждения IDLE.

ПРИМЕЧАНИЕ: В режиме отладки ICE-Crusher может предотвратить переход подсистемы MPU в режим IDLE.

**3.1.7.2.3 MPU вне режима ожидания**

MPU вне режима ожидания следует следующей последовательности действий и применим к начальному включению питания и выход из режима выключения устройства.

1. PRCM должен запускать тактирование через программирование DPLL.

2. Обнаружение активной синхронизации посредством вывода состояния DPLL.

3. Инициируйте прерывание через INTC для вывода ядра ARM из режима STANDBYWFI.

**3.1.7.2.4 Включение питания MPU из выключенного состояния**

1. MPU Power On, NEON Power On, Core Power On (INTC) должны следовать упорядоченной последовательности в соответствии с daisy chain последовательностью выключателя питания для минимизации пикового тока при включении питания.

*ПРИМЕЧАНИЕ: Основной домен должен быть включен и сброшен, прежде чем MPU может быть сброшен.*

2. Выполните последовательность сброса, описанную в разделе «Базовый сброс при включении питания».