МИРОВАЯ ЭЛЕКТРОНИКА

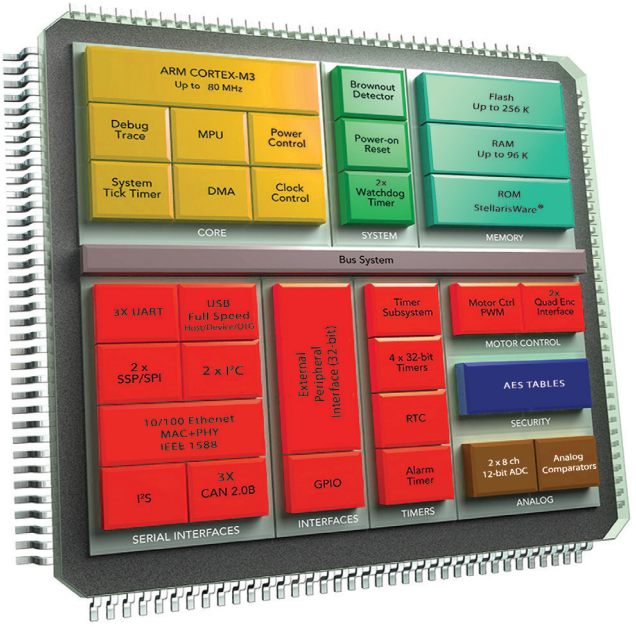
СЕРИЯ

Texas Instruments

Джозеф Ю

ЯДРО CORTEX-M3 КОМПАНИИ ARM

Полное руководство



- **Основы архитектуры ядра Cortex-M3, аппаратные функции, особенности отладки**
- **Набор инструкций из руководства пользователя на Cortex-M3**
- **Большое количество примеров приложений на Си и ассемблере, особенности отладки, диагностика и устранение неполадок, а также демонстрационный пример встроенного веб-сервера**
- **Новое приложение: семейство микроконтроллеров Stellaris от Texas Instruments, включая Cortex-M4F** NEW

ОДЭКА

ДМК ПРЕСС ИЗДАТЕЛЬСТВО

Newnes

УДК 004.31(035.3)
ББК 32.973-04я81
Ю11

Данное издание подготовлено при участии компании «Компэл» и российского представительства компании Texas Instruments. На сайтах www.compel.ru и www.ti.com/ru вы можете получить консультацию, а также заказать бесплатные образцы микросхем.

Телефон горячей линии технической поддержки TI +7-495-981-07-01.

Ю, Джозеф.

Ю11 Ядро Cortex-M3 компании ARM. Полное руководство / Джозеф Ю ; пер. с англ. А. В. Евстифеева. — М. : Додэка-XXI, 2012. — 552 с. : ил. — (Мировая электроника). — Доп. тит. л. англ. — ISBN 978-5-97060-307-9.

Настоящая книга представляет собой исчерпывающее руководство по новому 32-битному процессору компании ARM — Cortex-M3. В данном руководстве подробно описана архитектура процессорного ядра Cortex-M3 и его подсистемы памяти. Также подробно рассмотрены остальные узлы процессора, в том числе контроллер векторных прерываний NVIC, модуль защиты памяти MMU и разнообразные компоненты отладки. Приводится детальное описание новой системы команд Thumb-2, поддерживаемой данным процессором.

Книга содержит большое число примеров программного кода как на языке Си, так и на ассемблере.

Это руководство должно присутствовать на столе любого разработчика, использующего в своей работе микроконтроллеры с ядром Cortex-M3. Полнота и ясность изложения материала книги также позволяет рекомендовать её студентам соответствующих специальностей и подготовленным радиолюбителям.

УДК 004.31(035.3)
ББК 32.973-04я81

Книга «Ядро Cortex-M3 компании ARM. Полное руководство» Дж. Ю подготовлена и издана по договору с Elsevier Inc., 30 Corporste Drive, 4th Floor, Burlington, MA 01803, USA.

ISBN 978-1-85617-963-8 (англ.)
ISBN 978-5-94120-243-0 (Додэка)
ISBN 978-5-97060-307-9 (ДМК Пресс)

СОДЕРЖАНИЕ

ГЛАВА 2
ОБЗОР CORTEX-M3

2.1. Основные сведения

Процессор Cortex™-M3 представляет собой 32-битный микропроцессор. Он имеет 32-битную шину данных, 32-битный банк регистров и 32-битные интерфейсы памяти (**Рис. 2.1**).

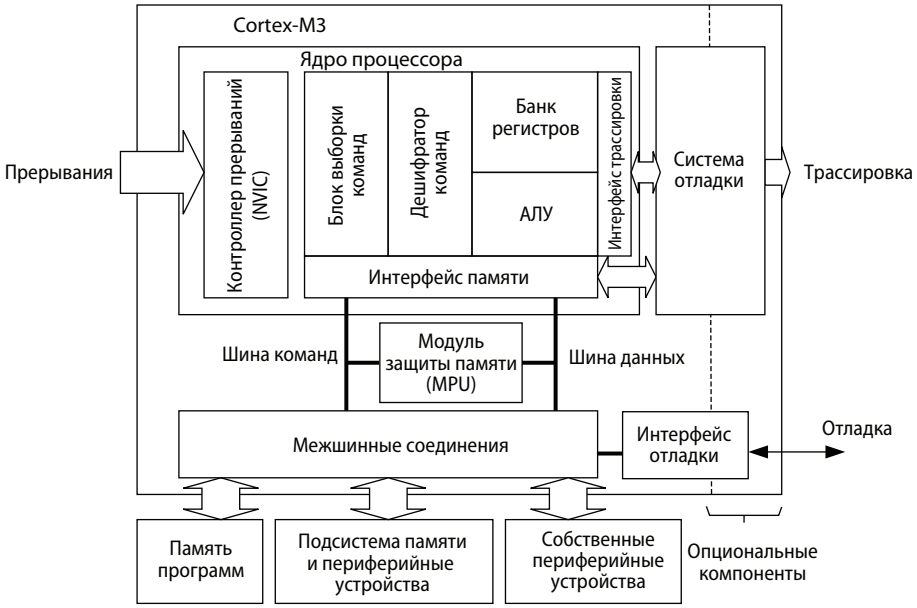


Рис. 2.1. *Упрощённая блок-схема Cortex-M3.*

Процессор выполнен по Гарвардской архитектуре, т.е. имеет раздельные шины команд и данных. Это позволяет осуществлять выборку команд одновременно с обращением к данным. В результате увеличивается производительность процессора, поскольку операции доступа к данным никак не влияют на конвейер команд. Это позволило реализовать в процессоре Cortex-M3 несколько шинных интерфейсов, каждый из которых оптимизирован для выполнения определённых функций и, в то же время, может использоваться одновременно с другими

интерфейсами. При этом шины команд и данных разделяют одно и то же адресное пространство (единая система памяти). Другими словами, наличие раздельных шинных интерфейсов вовсе не означает, что вы сможете использовать память размером 8 Гбайт.

Для поддержки сложных приложений, требующих более развитой системы памяти, в процессоре Cortex-M3 предусмотрены опциональный модуль защиты памяти (Memory Protection Unit — MPU) и возможность использования внешней кэш-памяти. Поддерживаются системы памяти, использующие как прямой (little endian), так и обратный (big endian) порядок байтов.

В составе процессора Cortex-M3 имеется ряд встроенных компонентов отладки. Эти компоненты осуществляют поддержку основных отладочных операций и возможностей, таких как точки останова (breakpoints) и точки наблюдений (watchpoints).

Предусмотрены и опциональные компоненты, поддерживающие расширенные возможности отладки, в частности трассировку команд, а также различные типы отладочных интерфейсов.

2.2. Регистры

Процессор Cortex-M3 имеет 16 регистров с R0 по R15 (**Рис. 2.2**). При этом регистр R13 (указатель стека) реализован в виде банка из двух регистров (в каждый момент времени доступен только один из регистров).

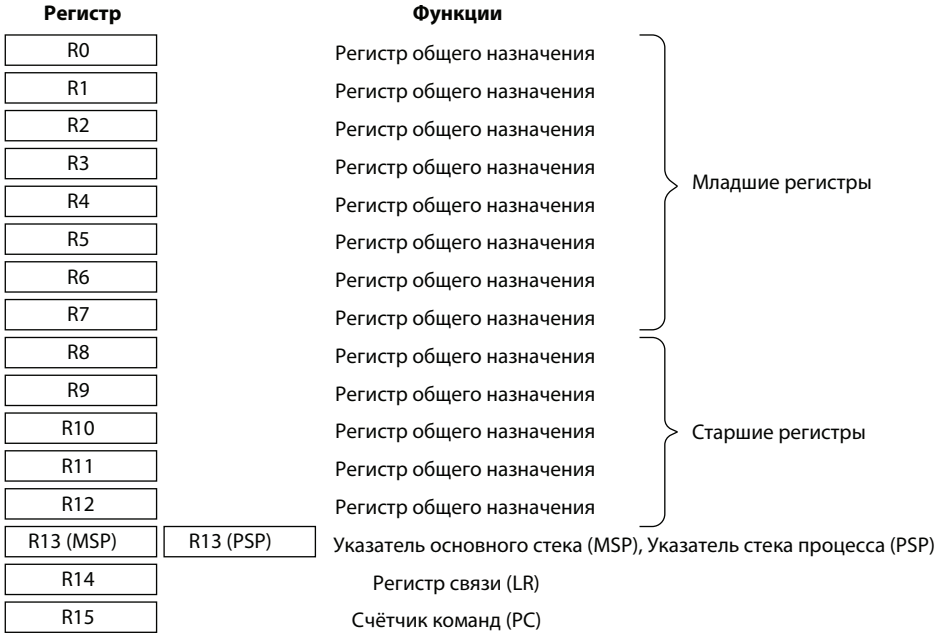


Рис. 2.2. *Регистры процессора Cortex-M3.*

2.2.1. R0…R12 — регистры общего назначения

Регистры с R0 по R12 являются 32-битными регистрами общего назначения, предназначенными для хранения обрабатываемых данных. Некоторые 16-битные команды Thumb® могут обращаться только к младшим регистрам (R0…R7).

2.2.2. R13 — указатели стека

Процессор Cortex-M3 содержит два указателя стека (R13). Они объединены в банк, поэтому в каждый момент времени виден только один из них:

- *Основной указатель стека* (Main Stack Pointer — MSP) — указатель стека, используемый ядром операционной системы и обработчиками исключительных ситуаций.
- *Указатель стека процесса* (Process Stack Pointer — PSP) — указатель стека, используемый прикладной программой.

Два младших бита указателей стека всегда сброшены в 0, т.е. эти указатели всегда выровнены на границу 32-битного слова.

2.2.3. R14 — регистр связи

В этом регистре при вызове подпрограммы запоминается адрес возврата.

2.2.4. R15 — счётчик команд

Счётчик команд (Program Counter — PC) содержит адрес выполняемой в данный момент команды. Этот регистр может быть изменён для управления ходом выполнения программы.

2.2.5. Регистры специального назначения

В процессоре Cortex-M3 также имеется несколько регистров специального назначения (**Рис. 2.3**):

- регистры состояния программы (xPSR);
- регистры маскирования прерываний (PRIMASK, FAULTMASK и BASEPRI);
- регистр управления (CONTROL).

Эти регистры выполняют специальные функции и для обращения к ним необходимо использовать особые команды. Указанные регистры не могут задействоваться для обработки и хранения обычных данных (**Табл. 2.1**).

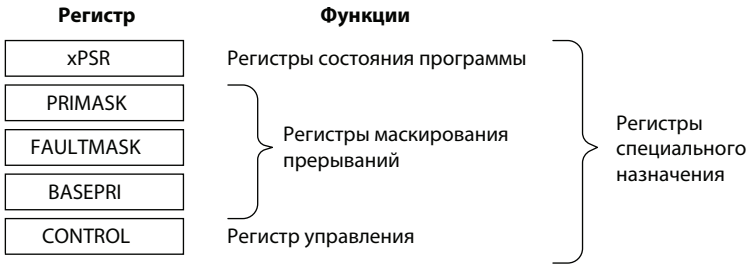


Рис. 2.3. *Регистры специального назначения процессора Cortex-M3.*

Таблица 2.1. **Регистры специального назначения и их функции**

Регистр	Назначение
xPSR	Содержат флаги результатов выполнения арифметических и логических операций (флаг нуля и флаг переноса), состояние выполнения программы и номер обрабатываемого в данный момент прерывания
PRIMASK	Запрещает все прерывания, за исключением немаскируемого прерывания (NMI) и исключения Hard Fault
FAULTMASK	Запрещает все прерывания, за исключением NMI
BASEPRI	Запрещает все прерывания, имеющие уровень приоритета, равный или меньший заданного
CONTROL	Определяет уровень доступа и используемый указатель стека

Примечание. Более подробно эти регистры рассмотрены в Главе 3.

2.3. Режимы работы

Процессор Cortex-M3 имеет два режима работы и поддерживает два уровня доступа к коду программы. Режимы работы Thread (режим потока) и Handler (режим обработчика) определяют, какой код выполняет процессор в данный момент времени — код обычной программы или же код обработчика исключительной ситуации, такой как прерывание или системное исключение (**Рис. 2.4**). Два уровня доступа к коду (привилегированный и пользовательский) обеспечивают безопасное обращение к критическим областям памяти, а также реализуют базовую модель механизма защиты.

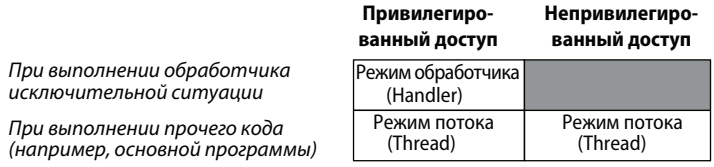


Рис. 2.4. *Режимы работы и уровни доступа к коду процессора Cortex-M3.*

При работе процессора в режиме потока допускается как привилегированное, так и непривилегированное выполнение программы, тогда как обработка исключительных ситуаций всегда осуществляется на привилегированном уровне. После сброса процессор находится в режиме потока с правами привилегированного доступа к коду. В состоянии привилегированного доступа программа может обращаться к любым областям памяти (за исключением тех, доступ к которым запрещён настройками модуля MPU) и использовать все поддерживаемые команды.

Программное обеспечение, выполняющееся на привилегированном уровне, может переключиться на пользовательский уровень, используя регистр управления. В случае возникновения исключительной ситуации процессор автоматически переключится на привилегированный уровень, а при выходе из обработчика — вернётся на исходный. Пользовательская программа не может самостоятельно переключиться на привилегированный уровень посредством записи в регистр управления (**Рис. 2.5**). Для этого необходим обработчик исключительной ситуации, который загрузит в регистр управления такое значение, чтобы при

возврате в режим потока процессор переключился на привилегированный уровень.

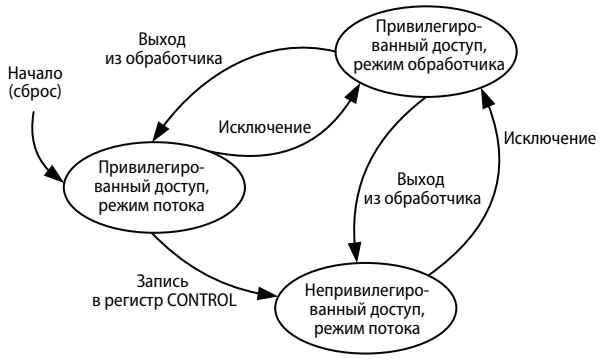


Рис. 2.5. *Диаграмма допустимых переходов между режимами работы процессора Cortex-M3.*

Наличие двух уровней доступа увеличивает надёжность системы, запрещая непроверенному коду доступ к регистрам, определяющим конфигурацию системы. При наличии модуля MPU он может использоваться совместно с привилегированным уровнем доступа для защиты критических секций памяти, например содержащих исполняемый код и данные ядра ОС.

Так, на привилегированном уровне доступа, обычно используемом ядром операционной системы, допускается обращение к любым областям памяти, не заблокированным настройками модуля MPU. При запуске системой пользовательского приложения, оно, в большинстве случаев, выполняется на непривилегированном (пользовательском) уровне, что позволяет защитить систему от возникновения сбоев из-за некорректного функционирования пользовательских программ.

2.4. Встроенный контроллер вложенных векторных прерываний

В составе процессора Cortex-M3 имеется контроллер вложенных векторных прерываний (Nested Vectored Interrupt Controller — NVIC). Он тесно связан с ядром процессора и выполняет следующие функции:

- поддержка вложенных прерываний;
- поддержка векторных прерываний;
- поддержка динамического изменения приоритетов;
- уменьшение задержки обработки прерывания;
- маскирование прерываний.

2.4.1. Поддержка вложенных прерываний

Контроллер NVIC обеспечивает поддержку вложенных прерываний. Всем внешним прерываниям и большинству системных исключений могут быть назначены различные уровни приоритета. При возникновении прерывания контроллер сравнивает его приоритет с приоритетом прерывания, обрабатываемого

в данный момент. Если новое прерывание имеет более высокий приоритет, то обработка текущего прерывания приостанавливается и запускается обработчик нового прерывания.

2.4.2. Поддержка векторных прерываний

В процессоре Cortex-M3 реализована поддержка векторных прерываний. В случае возникновения разрешённого прерывания стартовый адрес соответствующей процедуры обработки прерывания (Interrupt Service Routine — ISR) берётся из таблицы векторов, расположенной в памяти. Причём, определение стартового адреса ISR и переход на него осуществляется полностью аппаратно, что ускоряет обслуживание запроса прерывания.

2.4.3. Поддержка динамического изменения приоритетов

Уровни приоритета прерываний могут изменяться программно. При этом активация обслуживаемых в данный момент прерываний блокируется до выхода из процедуры обработки прерывания, что исключает нежелательный повторный вызов обработчиков прерываний при изменении приоритетов.

2.4.4. Уменьшение времени реакции на прерывание

В процессоре Cortex-M3 также реализованы определённые решения, позволяющие уменьшить задержку обработки прерываний. Это автоматическое сохранение и восстановление содержимого некоторых регистров, уменьшение задержки при переходе от одной процедуры обработки прерывания к другой, а также обработка «опоздавших» прерываний. Более подробно эти возможности процессора рассматриваются в Главе 9.

2.4.5. Маскирование прерываний

Прерывания и системные исключения могут быть маскированы в соответствии с их уровнями приоритета или же полностью при помощи регистров маскирования BASEPRI, PRIMASK и FAULTMASK. Эти регистры могут использоваться для того, чтобы исключить прерывание строго ограниченных во времени задач, тем самым гарантируя их своевременное завершение.

2.5. Карта памяти

В процессоре Cortex-M3 используется фиксированное распределение адресного пространства. Это позволяет обращаться к встроенным периферийным устройствам, таким как контроллер прерываний и компоненты системы отладки, посредством обычных команд доступа к памяти. То есть большинство системных функций можно использовать напрямую из программ, написанных на языке Си. Предопределённая карта памяти также обеспечивает чрезвычайно высокую скорость работы процессора и облегчает его интеграцию в системы на кристалле (System on a Chip — SoC).

Распределение общего адресного пространства размером 4 Гбайт показано на **Рис. 2.6**.

Рис. 2.6. *Карта памяти процессора Cortex-M3.*

Процессор Cortex-M3 имеет внутреннюю шинную инфраструктуру, которая оптимизирована для использования памяти, распределённой в соответствии с **Рис. 2.6**. Кроме того, конструкция процессора позволяет задействовать указанные области различным образом. Так, память данных может быть размещена в секции кода, а код программы может запускаться из секции внешнего ОЗУ.

В системных областях памяти располагаются контроллер прерываний и компоненты отладки. Эти устройства имеют фиксированные адреса, полная информация о которых приведена в Главе 5. Размещение этих периферийных устройств по фиксированным адресам значительно облегчает перенос приложений между микроконтроллерами различных производителей.

2.6. Интерфейсы шин

В процессоре Cortex-M3 реализовано несколько шин, что позволяет ему осуществлять выборку команд одновременно с обращением к данным. Можно выделить следующие основные интерфейсы шин:

- шины памяти кода;
- системная шина;
- шина собственных периферийных устройств.

Шины памяти кода предназначены для обеспечения доступа к одноимённой области памяти и физически реализованы в виде двух шин, называемых I-Code и D-Code. Такое решение позволило уменьшить время выборки команд, увеличив тем самым быстродействие процессора.

Системная шина используется для доступа к памяти и периферийным устройствам. С её помощью производятся обращения к статическому ОЗУ (СОЗУ), периферии, внешнему ОЗУ, внешним устройствам, а также к некоторым системным областям памяти.

Шина собственных периферийных устройств обеспечивает доступ к определённой части памяти, зарезервированной для использования встроенными периферийными устройствами процессора, такими как компоненты отладки.

2.7. Модуль защиты памяти MPU

В процессоре Cortex-M3 предусмотрен опциональный модуль защиты памяти MPU. Этот модуль позволяет задавать правила доступа к памяти на привилегированном и пользовательском уровнях. При нарушении правил доступа генерируется исключение отказа, в обработчике которого можно проанализировать проблему и, по возможности, скорректировать её.

Модуль MPU можно задействовать различным образом. В общем случае ОС может настроить MPU для защиты данных, используемых ядром ОС и другими привилегированными процессами, от недостаточно надёжных пользовательских программ. Помимо этого, модуль MPU может применяться для перевода определённых областей памяти в режим «только для чтения», позволяя тем самым предотвратить случайное повреждение данных или же изолировать области памяти различных задач в многозадачной системе. В общем и целом, модуль MPU помогает повысить надёжность и отказоустойчивость встроенных систем.

Модуль MPU является необязательным компонентом процессора, и решение о его использовании принимается на этапе реализации микроконтроллера или системы на кристалле. Более подробно о модуле MPU рассказывается в Главе 13.

2.8. Набор команд

Процессор Cortex-M3 поддерживает набор команд Thumb-2. Это одна из наиболее важных особенностей процессора, поскольку позволяет совместно использовать 16- и 32-битные команды, обеспечивая одновременно как высокую эффективность, так и высокую плотность кода. Набор Thumb-2 — гибкий, мощный и при этом простой в использовании набор команд.

В предыдущих процессорах компании ARM центральный процессор (ЦПУ) мог находиться в одном из двух состояний: 32-битном состоянии ARM и 16-битном состоянии Thumb. В состоянии ARM все команды являются 32-битными, что обеспечивает очень высокую производительность. В состоянии Thumb команды являются 16-битными, позволяя получить более высокую плотность кода. Однако эти команды имеют ограниченную функциональность по сравнению с командами ARM, поэтому для выполнения определённых операций может потребоваться большее число команд.

Для использования преимуществ обоих состояний код большинства приложений состоит из смеси команд ARM и Thumb. Однако такой подход не всегда себя оправдывает. На переключение процессора между состояниями расходуется как время, так и место в памяти (**Рис. 2.7**). К тому же наличие двух наборов команд может потребовать разбиения исходного кода на отдельные файлы, каждый из которых будет скомпилирован с использованием соответствующего набора. Это усложняет разработку программного обеспечения и уменьшает максимальную производительность ядра ЦПУ.

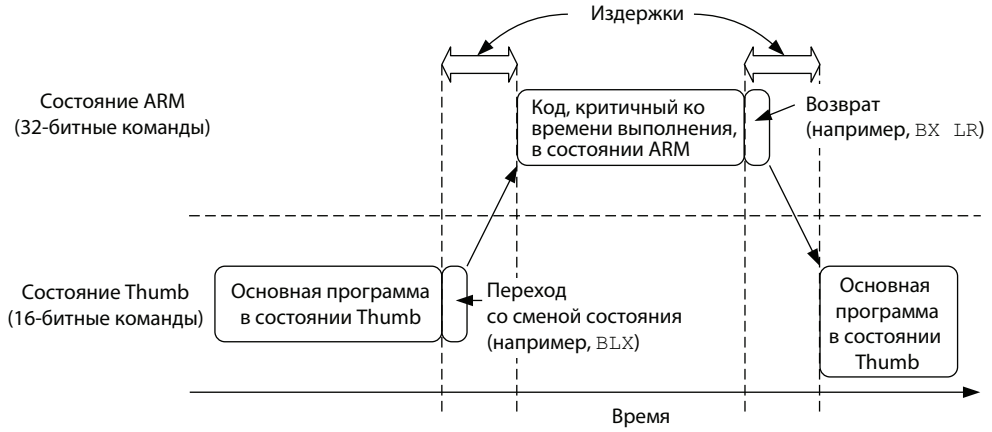


Рис. 2.7. *Переключение между состояниями ARM и Thumb в классическом процессоре ARM, таком как ARM7.*

С появлением набора команд Thumb-2 стало возможным выполнять все необходимые операции, находясь в одном состоянии, — необходимость в переключении между двумя рабочими состояниями полностью исчезла. Следует отметить, что процессор Cortex-M3 вообще не поддерживает набор команд ARM. Даже прерывания теперь обрабатываются в состоянии Thumb (ранее при входе в процедуры обработки прерываний процессор переключался в состояние ARM). Поскольку процессору Cortex-M3 не требуется переключаться между рабочими состояниями, он имеет ряд преимуществ по сравнению с традиционными процессорами ARM:

- отсутствуют накладные расходы на переключения между состояниями, в результате чего уменьшается время выполнения и размер кода программы;
- отсутствует необходимость разделения исходных файлов на файлы с кодом ARM и файлы с кодом Thumb, что облегчает разработку программ и их дальнейшее сопровождение;
- упрощается достижение максимальной эффективности и производительности, что, в свою очередь, облегчает разработку программного обеспечения (не требуется перепрыгивать с одного набора команд на другой, пытаясь достичь лучшего соотношения между размером кода и производительностью).

Процессор Cortex-M3 поддерживает ряд интересных и мощных команд. Вот только некоторые из них:

- `UBFX`, `BFI` и `BFC` — извлечение, вставка и очистка битового поля;

- `UDIV` и `SDIV` — беззнаковое и знаковое деление;
- `WFE`, `WFI` и `SEV` — ожидание события, ожидание прерывания и генерация события; эти команды используются для перевода процессора в режим пониженного энергопотребления и для поддержки синхронизации между задачами в многопроцессорных системах;
- `MSR` и `MRS` — пересылка данных между регистрами общего назначения и регистрами специального назначения.

Поскольку процессор Cortex-M3 поддерживает только набор команд Thumb-2, то для использования существующего кода, предназначенного для других процессоров ARM, требуется его перенос на новую архитектуру. В большинстве случаев процедура переноса заключается в перекомпиляции исходных текстов на языке Си с применением нового компилятора, поддерживающего процессор Cortex-M3. Некоторые фрагменты программы, написанные на ассемблере, придётся переписать с учётом новой архитектуры и нового унифицированного языка ассемблера.

Обратите внимание, что в процессоре Cortex-M3 реализованы не все команды из набора Thumb-2. В соответствии с [2] обязательным к реализации является только определённое подмножество команд Thumb-2. В частности, в процессоре Cortex-M3 отсутствует поддержка команд сопроцессора (могут быть подключены внешние устройства обработки данных), а также не реализованы SIMD-команды (один поток команд — несколько потоков данных). Кроме того, не поддерживаются некоторые команды Thumb, такие как команда перехода `BLX` (использовалась для переключения процессора из состояния Thumb в состояние ARM), ряд команд изменения состояния процесса (`CPS`), а также команды изменения представления многобайтных чисел (`SETEND`), появившиеся в архитектуре v6. Полный список поддерживаемых команд приведён в Приложении А.

2.9. Прерывания и исключения

В процессоре Cortex-M3 реализована новая модель исключений, разработанная для архитектуры ARMv7-M. Данная модель отличается от классической модели исключений ARM и обеспечивает очень эффективную поддержку исключительных ситуаций. В этой модели предусмотрено несколько системных исключений плюс некоторое количество внешних запросов прерываний (входы внешних прерываний). В процессоре Cortex-M3 отсутствует быстрое прерывание FIQ, имеющееся в процессорах ARM7/ARM9/ARM10/ARM11. С другой стороны, он поддерживает приоритеты прерываний, а также вложенные прерывания. Поэтому не составляет никакого труда реализовать систему с поддержкой вложенных прерываний (прерывание с более высоким приоритетом может приостановить выполнение обработчика прерывания с более низким приоритетом), которые будут вести себя аналогично прерыванию FIQ в предыдущих процессорах ARM.

За поддержку прерываний в процессоре Cortex-M3 отвечает контроллер прерываний NVIC. Помимо внешних прерываний, процессор также поддерживает несколько внутренних источников исключений, предназначенных, в частности, для обработки системных отказов. Соответственно, в процессоре имеется несколько предопределённых типов исключений (**Табл. 2.2**).

Таблица 2.2. **Типы исключений Cortex-M3**

Номер исключения	Тип исключения	Приоритет*	Описание
0	—	—	Исключение отсутствует
1	Reset	–3 (Наивысший)	Сброс
2	NMI	–2	Немаскируемое прерывание (вход внешнего немаскируемого прерывания)
3	Hard Fault	–1	Любой отказ, если соответствующий обработчик не разрешён
4	MemManage Fault	Программируемый	Отказ системы управления памятью; нарушение правил доступа, заданных модулем MPU, или обращение по некорректному адресу
5	Bus Fault	Программируемый	Отказ шины (отказ предвыборки или отказ данных)
6	Usage Fault	Программируемый	Отказ программы
7…10	Зарезервировано	—	Зарезервировано
11	SVCall	Программируемый	Вызов супервизора
12	Debug monitor	Программируемый	Исключение монитора отладки (точки останова, точки наблюдения или внешняя команда отладки)
13	Зарезервировано	—	Зарезервировано
14	PendSV	Программируемый	Запрос системной службы
15	SYSTICK	Программируемый	Системный таймер
16	IRQ #0	Программируемый	Внешнее прерывание №0
17	IRQ #1	Программируемый	Внешнее прерывание №1
…	…	…	…
255	IRQ #239	Программируемый	Внешнее прерывание №239
* Если допускает программирование, то по умолчанию равен 0.			

Примечание. Количество входов внешних прерываний определяется изготовителями микросхем (поддерживается до 240 входов). Кроме того, Cortex-M3 имеет вход немаскируемого прерывания NMI, при активации которого в обязательном порядке запускается обработчик немаскируемого прерывания.

2.9.1. Низкое энергопотребление и высокая энергоэффективность

В процессоре Cortex-M3 применён ряд решений, позволяющих разработчикам создавать экономичные и энергоэффективные изделия. Прежде всего, это наличие двух режимов пониженного энергопотребления, позволяющих использовать различные стратегии для уменьшения потребления во время простоя.

Во-вторых, снижению потребляемой мощности способствует относительно небольшое число логических вентилей, образующих процессор, а также определённые схемотехнические решения, позволяющие уменьшить активность отдельных его узлов. Кроме того, процессор Cortex-M3 обеспечивает высокую плотность кода, что снижает требования к объёму памяти программ. В то же время все эти особенности позволяют ускорить выполнение различных задач обработ-

ки данных и, соответственно, возврат процессора в спящий режим для сокращения энергопотребления. В результате энергоэффективность процессора Cortex-M3 оказывается лучшей, нежели у большинства 8- и 16-битных микроконтроллеров.

Во второй ревизии процессора Cortex-M3 появился новый модуль — контроллер «пробуждающих» прерываний (Wakeup Interrupt Controller — WIC). Этот модуль позволяет отключать питание процессорного ядра с сохранением состояния процессора, а также обеспечивает практически мгновенный возврат процессора в активное состояние при возникновении прерывания. Подобная возможность позволяет использовать процессор Cortex-M3 во многих приложениях со сверхнизким потреблением, которые прежде могли быть выполнены только на 8- или 16-битных микроконтроллерах.

2.10. Возможности отладки

Процессор Cortex-M3 поддерживает различные функции отладки, такие как управление процессом выполнения программы, включая останов и пошаговое исполнение, точки останова и точки наблюдения данных, обращение к регистрам процессора и к памяти «на лету», профилирование и трассировку.

Аппаратные средства отладки процессора Cortex-M3 базируются на архитектуре CoreSight™. В отличие от традиционных процессоров ARM, в самом ядре процессора интерфейс JTAG отсутствует. Вместо этого интерфейс отладки реализован в виде отдельного модуля, для связи с которым в процессоре предусмотрен специальный интерфейс, называемый *портом доступа к средствам отладки* (Debug Access Port — DAP). С помощью указанного интерфейса внешние отладчики могут обращаться как к регистрам управления аппаратных средств отладки, так и к системе памяти даже во время выполнения процессором программы. Управление данным интерфейсом осуществляется через внешний *порт отладки* (Debug Port — DP). В настоящее время реализованы следующие порты: Serial-Wire JTAG Debug Port (SWJ-DP), поддерживающий как традиционный протокол JTAG, так и протокол Serial-Wire, и SW-DP (поддерживает только протокол Serial-Wire). Также можно использовать модуль JTAG-DP из семейства продукции CoreSight™ от ARM. Производители микроконтроллеров могут реализовать интерфейс отладки на базе любого из указанных модулей.

Для обеспечения возможности трассировки команд производители микросхем могут также включать в свои изделия *модуль встроенной макроячейки трассировки* (Embedded Trace Macrocell — ETM). Трассировочная информация выводится через *модуль интерфейса порта трассировки* (Trace Port Interface Unit — TPIU). Далее информация о выполненных командах с помощью внешнего аппаратного трассировщика передаётся хосту отладки, в качестве которого обычно применяется персональный компьютер.

В самом процессоре для запуска отладочных действий могут использоваться различные события. Источниками этих событий могут служить точки останова, точки наблюдения, отказы или сигналы от внешнего отладчика. При возникновении любого из указанных событий процессор может либо перейти в режим останова, либо запустить обработчик исключения монитора отладки.

Функция точек наблюдения данных реализуется *модулем просмотра и трассировки данных* (Data Watchpoint and Trace — DWT) процессора Cortex-M3. Этот модуль может быть задействован для останова процессора (или для запуска обработчика исключения монитора отладки) или же для генерации информации о трассировке данных. При осуществлении трассировки соответствующая информация выводится через модуль TPIU. Следует отметить, что в архитектуре CoreSight единственный порт трассировки может использоваться совместно несколькими устройствами трассировки.

Помимо описанных базовых средств отладки, в процессоре Cortex-M3 также имеется *модуль коррекции флэш-памяти и задания точки останова* (Flash Patch and Breakpoint — FPB). Этот модуль может использоваться для установки простых точек останова или же для переназначения адресов команд из области флэш-памяти в область ОЗУ.

Макроячейка инструментальной трассировки (Instrumentation Trace Macrocell — ITM) предоставляет разработчику новый канал для передачи данных в отладчик. Данные, записываемые в регистры модуля ITM, могут быть получены отладчиком по интерфейсу трассировки для их последующего отображения или обработки. Этот метод прост в использовании и более быстрый, нежели вывод по интерфейсу JTAG.

Управление всеми указанными компонентами отладки осуществляется по шине интерфейса DAP процессора Cortex-M3 или же программой, выполняемой процессором. Вся информация о процессе трассировки может быть получена из модуля TPIU.

2.11. Резюме

Почему процессор Cortex-M3 считается революционным устройством? Какие преимущества даёт его использование? Ответы на эти и аналогичные вопросы можно найти в данном разделе.

2.11.1. Высокая производительность

Процессор Cortex-M3 обеспечивает высокое быстродействие микроконтроллеров:

- Большинство команд, включая команды умножения, выполняются за один такт. По данному параметру Cortex-M3 опережает большинство популярных микроконтроллеров.
- Раздельные шины команд и данных позволяют одновременно выполнять операции выборки команд и обращения к данным.
- Набор команд Thumb-2 исключает непроизводительные издержки на переключение состояний процессора. Больше не нужно тратить время на переход между 32-битным состоянием ARM и 16-битным состоянием Thumb, что увеличивает скорость выполнения программы и уменьшает её размер. Это новшество также упрощает разработку программного обеспечения, что, в свою очередь, сокращает время выхода продукции на рынок и облегчает последующее сопровождение кода.

- Набор Thumb-2 является чрезвычайно гибким. Многие операции могут быть выполнены с использованием меньшего числа команд. Как следствие, Cortex-M3 обеспечивает бóльшую плотность кода и требует меньше памяти для хранения программ.
- Выборка команд осуществляется 32-битными словами, соответственно, за один такт может быть выбрано до двух команд. В результате обеспечивается более высокая пропускная способность для передачи данных.
- Конструкция процессора Cortex-M3 позволяет создавать микроконтроллеры, работающие на высокой частоте (при использовании современных техпроцессов — более 100 МГц). Но даже при работе на той же частоте, что и большинство других микроконтроллеров, Cortex-M3 имеет лучшее соотношение числа тактов на одну команду (CPI). Это позволяет выполнять большее число операций в пересчёте на мегагерц тактовой частоты или же даёт возможность снизить тактовую частоту для уменьшения энергопотребления.

2.11.2. Развитые средства поддержки прерываний

Средства поддержки прерываний в процессоре Cortex-M3 легки в использовании, обладают большой гибкостью и обеспечивают высокую производительность при обработке прерываний:

- Встроенный контроллер прерываний NVIC поддерживает до 240 входов внешних прерываний. Поддержка векторных прерываний значительно уменьшает задержку обработки прерываний, поскольку выбор необходимого обработчика осуществляется полностью аппаратно. Кроме того, не требуется прибегать к программным ухищрениям для обеспечения поддержки вложенных прерываний.
- Процессор Cortex-M3 при входе в процедуру обработки прерывания автоматически сохраняет в стеке регистры R0…R3, R12, LR, PSR и PC и извлекает их из стека при выходе из обработчика. Это уменьшает задержку обработки запроса прерывания и даёт возможность описывать обработчик прерывания в виде обычной функции на языке Си (см. Главу 8).
- Система прерываний является чрезвычайно гибкой, поскольку NVIC позволяет задавать приоритет индивидуально для каждого прерывания. Поддерживаются не менее 8 уровней приоритета, причём приоритет может изменяться динамически.
- Для уменьшения задержки обработки прерываний используются такие специальные методы, как принятие «опоздавших» прерываний и прямой переход (tail-chain) от одного обработчика прерывания к другому.
- Допускается прерывание некоторых операций, выполняемых за несколько тактов, в том числе операций загрузки/сохранения нескольких регистров (`LDM`/`STM`) и операций сохранения в стеке и извлечения из стека (`PUSH`/`POP`).

При появлении запроса немаскируемого прерывания гарантируется немедленный запуск обработчика этого прерывания, если только система не является полностью заблокированной. Наличие немаскируемого прерывания является