

ВИКИПЕДИЯ

ARM (архитектура)

Материал из Википедии — свободной энциклопедии

Архитектура ARM (от англ. *Advanced RISC Machine* — усовершенствованная RISC-машина; иногда — *Acorn RISC Machine*) — система команд и семейство описаний и готовых топологий 32-битных и 64-битных микропроцессорных/микроконтроллерных ядер, разрабатываемых компанией ARM Limited^[1].

Среди лицензиатов готовых топологий ядер ARM — компании AMD, Apple, Analog Devices, Atmel, Xilinx, Cirrus Logic, Intel (до 27 июня 2006 года), Marvell, NXP, STMicroelectronics, Samsung, LG, MediaTek, Qualcomm, Sony, Texas Instruments, Nvidia, Freescale, Миландр, ЭЛВИС^[2], HiSilicon, Байкал электроникс.

Значимые семейства процессоров: ARM7, ARM9, ARM11 и Cortex^{[3][4]}.

Многие лицензиаты проектируют собственные топологии ядер на базе системы команд ARM: DEC StrongARM, Freescale i.MX, Intel XScale, NVIDIA Tegra, ST-Ericsson Nomadik, Krait и Kryo в Qualcomm Snapdragon, Texas Instruments OMAP, Samsung Hummingbird, LG H13, Apple A6 и HiSilicon K3.

Содержание

Популярность

История

Acorn RISC Machine: ARM2

Apple, DEC, Intel: ARM6, StrongARM, XScale

Влияние ARM-технологии на рынок

Процессоры ARM

Архитектура

Режимы

Набор команд

Набор команд ARM

Набор команд Thumb



Разработчик	ARM Limited
Разрядность	ARMv7 64/32 бит, до ARMv8 только 64 бит
Представлена	1985
Архитектура	RISC
Тип	Регистр-регистр
Кодирование СК	ARMv8: фиксированное (64 бит), кроме режима совместимости с ARMv7 кодом ARMv7/6: смешанное (32 и 16 бит выровнено) ARMv3/2: фиксированное (32 бит)
Реализация переходов	по флагам условий
Порядок байтов	Переключаемый (big-endian) , обычно — от младшего к старшему (little-endian)
Расширения	NEON, Thumb-2 (обязательно начиная с ARMv7), Jazelle, VFPv4-D16, VFPv4 (все обязательны в ARMv8) В микроконтроллерах: FPv4-SP



Медиафайлы на Викискладе

[Набор команд Thumb-2](#)

[Набор команд Jazelle](#)

[ARMv8 и набор команд ARM 64 бита](#)

[Функции RISC](#)

[Условное исполнение](#)

[Другие особенности](#)

[Конвейер и другие аспекты реализации](#)

[Сопроцессоры](#)

[Усовершенствованный SIMD \(NEON\)](#)

[VFP](#)

[Расширения безопасности](#)

[Отладка](#)

[Регистры](#)

[Работа с памятью](#)

[Поддерживаемые системы ввода-вывода](#)

[Процесс запуска ОС на ARM-машинах](#)

[ОС, поддерживающие ARM](#)

[См. также](#)

[Примечания](#)

[Ссылки](#)



ARM-процессор производства [Conexant](#), устанавливается в основном в [маршрутизаторах](#)

Популярность

В 2006 году около 98 % из более чем миллиарда мобильных телефонов, продававшихся ежегодно, были оснащены, по крайней мере, одним процессором ARM^[5]. По состоянию на 2009 на процессоры ARM приходилось до 90 % всех встроенных 32-разрядных процессоров^[6]. Процессоры ARM широко используются в потребительской электронике — в том числе смартфонах, мобильных телефонах и плеерах, портативных игровых консолях, калькуляторах, умных часах и компьютерных периферийных устройствах, таких, как жесткие диски или маршрутизаторы.

Эти процессоры имеют низкое энергопотребление, поэтому находят широкое применение во [встраиваемых системах](#) и преобладают на рынке мобильных устройств, для которых данный фактор критически важен.

История

После достижения некоторых успехов с компьютером [BBC Micro](#) британская компания [Acorn Computers](#) задумалась над переходом от относительно слабых процессоров [MOS Technology 6502](#) к более производительным решениям и выходом на рынок бизнес-компьютеров с той же платформой [BBC Micro](#). Такие процессоры, как [Motorola 68000](#) и [32016](#) от [National Semiconductor](#), были для этого непригодны, а [6502](#) был недостаточно мощным, чтобы поддерживать [графический пользовательский интерфейс](#)^[7].

Компании была нужна совершенно новая архитектура после того, как она протестировала все доступные ей процессоры и сочла их неэффективными. Acorn серьёзно настроился на разработку собственного процессора, и их инженеры начали изучать документацию проекта [RISC](#), разработанного в [Университете Калифорнии в Беркли](#). Они подумали, что раз уж группе студентов удалось создать вполне конкурентоспособный процессор, то их инженерам

это будет несложно. Поездка в Western Design Center (Аризона) показала инженерам Стиву Ферберу и Софи Уилсон (на тот момент известной под именем Роджер^[8]), что им не потребуются невероятные ресурсы для осуществления этого плана.

Уилсон приступила к разработке системы команд, создавая симулятор нового процессора на компьютере BBC Micro. Её успехи в этом убедили инженеров Acorn, что они на верном пути. Но все же перед тем, как идти дальше, им требовалось больше ресурсов, настало время для Уилсон идти к директору Acorn Герману Хаузеру и объяснить, в чём же дело. После того, как он дал добро, собралась небольшая команда для реализации модели Уилсон на аппаратном уровне.

Acorn RISC Machine: ARM2

Официальный проект Acorn RISC Machine был начат в октябре 1983 года. VLSI Technology была выбрана в качестве поставщика кремниевых компонентов, так как она уже снабжала Acorn микросхемами ПЗУ и некоторыми нестандартными интегральными схемами. Разработку возглавили Уилсон и Фербер. Их основной целью было достижение низкой латентности обработки прерывания, как у MOS Technology 6502. Архитектура доступа к памяти, взятая от 6502, позволила разработчикам достичь хорошей производительности без использования дорогостоящего в реализации модуля DMA. Первый процессор был произведен VLSI 26 апреля 1985 года — именно тогда он впервые заработал и был назван ARM1. Первые серийные процессоры под названием ARM2 стали доступны в следующем году.

Его первое применение было в качестве второго процессора в BBC Micro, где он был использован при разработке программного обеспечения для моделирования, что позволило завершить работу над вспомогательными микросхемами компьютера, а также ускорить работу программного обеспечения CAD, используемого при разработке ARM2. Уилсон оптимизировала набор инструкций ARM для исполнения BBC BASIC. Изначальная цель компьютера, полностью построенного на базе ARM, была достигнута в 1987 году с выходом Acorn Archimedes.

Атмосфера вокруг проекта ARM была настолько секретна, что когда компания Olivetti вела переговоры о покупке контрольного пакета акций Acorn в 1985 году, они не стали рассказывать о развитии проекта до конца переговоров. В 1992 году Acorn ещё раз выиграл Премию королевы для предприятий за ARM.

В ARM2 была 32-разрядная шина данных, 26-битное адресное пространство и 16 32-разрядных регистров. Программный код должен был лежать в первых 64 мегабайтах памяти, а программный счётчик был ограничен 26 битами, так как верхние 4 и нижние 2 бита 32-битного регистра служили флагами. ARM2 стал, возможно, самым простым из популярных 32-битных процессоров в мире, имея всего лишь 30 тысяч транзисторов (для сравнения, в сделанном на 6 лет раньше процессоре Motorola 68000 было 68 тысяч транзисторов). Многие из этой простоты обусловлено отсутствием микрокода (который в процессоре 68000 занимает от одной четверти до одной трети площади кристалла) и отсутствием кэша, как и во многих процессорах того времени. Эта простота привела к низким затратам энергии, в то время как ARM был гораздо более производителен, чем Intel 80286^[9]. У его преемника — процессора ARM3 — уже был кэш 4 кб, что ещё больше увеличило производительность.

Apple, DEC, Intel: ARM6, StrongARM, XScale

В конце 1980-х годов Apple Computer и VLSI Technology начали работать с Acorn Computers над новыми версиями ядра ARM. Работа была настолько важна, что Acorn преобразовала команду разработчиков в 1990 году в новую компанию под названием Advanced RISC Machines. По этой причине ARM иногда расшифровывают как *Advanced RISC Machines* вместо *Acorn RISC Machine*. Advanced RISC Machines стала ARM, когда её родительская компания ARM Holdings вышла на Лондонскую фондовую биржу и NASDAQ в 1998 году.

Новая работа Apple-ARM в конечном итоге превратилась в ARM6, впервые выпущенный в 1992 году. Apple использовала основанный на базе ARM6 процессор ARM610 в качестве основы для своего продукта Apple Newton PDA. В 1994 году Acorn стала использовать ARM610 как главный процессор в своих компьютерах RISC PC. Компания DEC также купила лицензию на архитектуру ARM6 (чем вызвала небольшую путаницу, поскольку они также производили процессоры Alpha) и начала производить StrongARM. На 233 МГц этот процессор требовал всего 1 Вт мощности (более поздние версии требовали гораздо меньше). Позднее Intel получил права на эту работу в результате судебного процесса. Intel воспользовалась возможностью дополнить свою устаревшую линейку I960 процессором StrongARM и позднее разработала свою версию ядра под торговой маркой XScale, которую они впоследствии продали компании Marvell.

Ядро ARM сохранило все тот же размер после всех этих изменений. У ARM2 было 30 тысяч транзисторов.

Влияние ARM-технологии на рынок

В основном процессоры семейства завоевали сегмент массовых мобильных продуктов (сотовые телефоны, карманные компьютеры) и встраиваемых систем средней и высокой производительности (от сетевых маршрутизаторов и точек доступа до телевизоров). Отдельные компании заявляют о разработках эффективных серверов на базе кластеров ARM-процессоров^[10], но пока это только экспериментальные проекты с 32-битной архитектурой^[11].

Предыдущий лидер списка — суперкомпьютер Summit (OLCF-4), разработанный совместно компаниями IBM (серверные узлы), Mellanox (межсоединение) и Nvidia (графические ускорители), и установленный в Ок-Риджской национальной лаборатории — стал вторым, уступив почетное первое место новой японской топ-системе Fugaku (рус. *Фугаку*), которая показала результат High Performance/Highly Parallel Linpack (HPL), равный 415,5 петафлопс. Данный показатель превосходит возможности Summit в 2,8 раза. Fugaku оснащен 48-ядерным процессором A64FX SoC фирмы Fujitsu, таким образом, японская разработка стала первой в истории системой № 1 в списке TOP500, при этом оснащенной процессорами ARM. При одинарной или более низкой точности, которая часто используется для задач машинного обучения и искусственного интеллекта, пиковая производительность Fugaku составляет более 1000 петафлопс (1 эксафлопс). Новая система установлена в Центре вычислительных наук RIKEN (R-CCS) в Кобе, Япония.

Процессоры ARM

В настоящее время значимыми являются несколько семейств процессоров ARM:

- ARM7 (с тактовой частотой до 60-72 МГц), предназначенные, например, для недорогих мобильных телефонов и встраиваемых решений средней производительности. В настоящее время активно вытесняется новым семейством Cortex.
- ARM9, ARM11 (с частотами до 1 ГГц) для более мощных телефонов, карманных компьютеров и встраиваемых решений высокой производительности.

- Cortex A — новое семейство процессоров на смену [ARM9](#) и ARM11.
- Cortex M (<http://www.arm.com/products/processors/cortex-m>) — новое семейство процессоров на смену ARM7, также призванное занять новую для ARM нишу встраиваемых решений низкой производительности. В семействе присутствуют четыре значимых ядра:
 - Cortex-M0, Cortex-M0+ (более энергоэффективное) и Cortex-M1 (оптимизировано для применения в ПЛИС) с архитектурой ARMv6-M;
 - Cortex-M3 с архитектурой ARMv7-M;
 - Cortex-M4 (добавлены SIMD-инструкции, опционально FPU) и Cortex-M7 (<https://www.arm.com/products/processors/cortex-m/cortex-m7-processor.php>) (FPU с поддержкой чисел одинарной и двойной точности) с архитектурой ARMv7E-M;
 - Cortex-M23 (<https://www.arm.com/products/processors/cortex-m/cortex-m23-processor.php>) и Cortex-M33 (<https://www.arm.com/products/processors/cortex-m/cortex-m33-processor.php>) с архитектурой ARMv8-M (<https://www.arm.com/products/processors/instruction-set-architectures/armv8-m-architecture.php>) ARMv8-M .

В 2010 году производитель анонсировал процессоры Cortex-A15 под кодовым названием Eagle, ARM утверждает, что ядро Cortex A15 на 40 процентов производительнее на той же частоте, чем ядро Cortex-A9 при одинаковом числе ядер на чипе^[12]. Изделие, изготовленное по 28-нанометровому техпроцессу, имеет 4 ядра, может функционировать на частоте до 2,5 ГГц и будет поддерживаться многими современными операционными системами^{[13][14]}.

Популярное семейство микропроцессоров [xScale](#) фирмы Marvell (до 27 июня 2007 года — Intel^[15]) в действительности является расширением архитектуры [ARM9](#), дополненной набором инструкций [Wireless MMX](#), специально разработанных фирмой Intel для поддержки мультимедийных приложений.

Версии ядра ARM (таблица)

[\[показать\]](#)

Архитектура

Уже давно существует справочное руководство по архитектуре ARM, которое разграничивает все типы интерфейсов, которые поддерживает ARM, так как детали реализации каждого типа процессора могут различаться. Архитектура развивалась с течением времени и, начиная с ARMv7, были определены 3 профиля:

- A (application) — для устройств, требующих высокой производительности (смартфоны, планшеты);
- R (real time) — для приложений, работающих в реальном времени;
- M (microcontroller) — для микроконтроллеров и недорогих встраиваемых устройств^[3].

Профили могут поддерживать меньшее количество команд (команды определенного типа).

Режимы

Процессор может находиться в одном из следующих операционных режимов:

- User mode — обычный режим выполнения программ. В этом режиме выполняется большинство программ.
- Fast Interrupt (FIQ) — режим быстрого прерывания (меньшее время срабатывания).
- Interrupt (IRQ) — основной режим прерывания.

- **System mode** — защищённый режим для использования операционной системой.
- **Abort mode** — режим, в который процессор переходит при возникновении ошибки доступа к памяти (доступ к данным или к инструкции на этапе prefetch конвейера).
- **Supervisor mode** — привилегированный пользовательский режим.
- **Undefined mode** — режим, в который процессор входит при попытке выполнить неизвестную ему инструкцию.

Переключение режима процессора происходит при возникновении соответствующего исключения или же модификацией регистра статуса.

Набор команд

Чтобы сохранить устройство чистым, простым и быстрым, оригинальное изготовление ARM было исполнено без микрокода, как и более простой 8-разрядный процессор 6502, используемый в предыдущих микрокомпьютерах от Acorn Computers.

Набор команд ARM

Режим, в котором исполняется 32-битный набор команд.

ARM Base Instruction Set:^[49]

ADC, ADD, AND, B/BL, BIC, CMN, CMP, EOR, LDM, LDR/LDRB, MLA, MOV, MUL, MVN, ORR, RSB, RSC, SBC, STM, STR/STRB, SUB, SWI, SWP, TEQ, TST

Набор команд Thumb

Для улучшения плотности кода процессоры, начиная с ARM7TDMI, снабжены режимом «thumb». В этом режиме процессор выполняет альтернативный набор 16-битных команд. Большинство из этих 16-разрядных команд переводится в нормальные команды ARM. Уменьшение длины команды достигается за счёт сокрытия некоторых операндов и ограничения возможностей адресации по сравнению с режимом полного набора команд ARM.

В режиме Thumb меньшие коды операций обладают меньшей функциональностью. Например, только ветвления могут быть условными, и многие коды операций имеют ограничение в виде доступа только к половине главных регистров процессора. Более короткие коды операций в целом дают большую плотность кода, хотя некоторые операции требуют дополнительных команд. В ситуациях, когда порт памяти или ширина шины ограничены 16 битами, более короткие коды операций режима Thumb становятся гораздо производительнее по сравнению с обычным 32-битным ARM-кодом, так как меньший программный код придется загружать в процессор при ограниченной пропускной способности памяти.

Аппаратные средства типа Game Boy Advance, как правило, имеют небольшой объём оперативной памяти, доступной с полным 32-битным информационным каналом. Но большинство операций выполняется через 16-битный или более узкий информационный канал. В этом случае имеет смысл использовать Thumb-код и вручную оптимизировать некоторые тяжелые участки кода, используя переключение в режим полных 32-битных инструкций ARM.

Первым процессором с декодером Thumb-команд был ARM7TDMI. Все процессоры семейства ARM9, а также XScale, имели встроенный декодер Thumb-команд.

Набор команд Thumb-2

Thumb-2 — технология, появившаяся в ARM1156 core, анонсированном в 2003 году. Он расширяет ограниченный 16-битный набор команд Thumb дополнительными 32-битными командами, чтобы задать набору команд дополнительную ширину. Цель Thumb-2 — достичь плотности кода, как у Thumb, и производительности, как у набора команд ARM на 32 битах. Можно сказать, что в ARMv7 эта цель была достигнута.

Thumb-2 расширяет как команды ARM, так и команды Thumb ещё большим количеством команд, включая управление битовым полем, табличное ветвление, условное исполнение. Новый язык «Unified Assembly Language» (UAL) поддерживает создание команд, как для ARM, так и для Thumb из одного и того же исходного кода. Версии Thumb на ARMv7 выглядят, как код ARM. Это требует осторожности и использования новой команды if-then, которая поддерживает исполнение до 4 последовательных команд испытываемого состояния. Во время компиляции в ARM-код она игнорируется, но во время компиляции в код Thumb-2 генерирует команды. Например:

```

; if (r0 == r1)
CMP r0, r1
ITE EQ ; ARM: no code ... Thumb: IT instruction
; then r0 = r2;
MOVEQ r0, r2 ; ARM: conditional; Thumb: condition via ITE 'T' (then)
; else r0 = r3;
MOVNE r0, r3 ; ARM: conditional; Thumb: condition via ITE 'E' (else)
; recall that the Thumb MOV instruction has no bits to encode "EQ" or "NE"

```

Все кристаллы ARMv7 поддерживают набор команд Thumb-2, а некоторые кристаллы, вроде Cortex-m3, поддерживают только Thumb-2. Остальные кристаллы Cortex и ARM11 поддерживают наборы команд как Thumb-2, так и ARM.

Набор команд Jazelle

Jazelle — это технология, которая позволяет байткоду Java исполняться прямо в архитектуре ARM в качестве 3-го состояния исполнения (и набора команд) наряду с обычными командами ARM и режимом Thumb. Поддержка технологии Jazelle обозначается буквой «J» в названии процессора — например, ARMv5TEJ. Данная технология поддерживается начиная с архитектуры ARMv6, хотя новые ядра содержат лишь ограниченные реализации, которые не поддерживают аппаратного ускорения.

ARMv8 и набор команд ARM 64 бита

В конце 2011 года была опубликована новая версия архитектуры, ARMv8. В ней появилось определение архитектуры AArch64, в которой исполняется 64-битный набор команд A64. Поддержка 32-битных команд получила название A32 и исполняется на архитектурах AArch32. Инструкции Thumb поддерживаются в режиме T32, только при использовании 32-битных архитектур. Допускается исполнение 32-битных приложений в 64-битной ОС, и запуск виртуализованной 32-битной ОС при помощи 64-битного гипервизора^[50]. Applied Micro, AMD, Broadcom, Calxeda, HiSilicon, Samsung, STM и другие заявили о планах по использованию ARMv8. Ядра Cortex-A53 и Cortex-A57, поддерживающие ARMv8, были представлены компанией ARM 30 октября 2012 года^[51].

Как AArch32, так и AArch64, поддерживают VFPv3, VFPv4 и *advanced SIMD* (NEON). Также добавлены криптографические инструкции для работы с AES, SHA-1 и SHA-256.

Особенности AArch64:

- Новый набор команд A64
- 31 регистр общего назначения, каждый длиной 64 бит
- Отдельные регистры SP и PC
- Инструкции имеют размер 32 бит и многие совпадают с командами A32
- Большинство инструкций работает как с 32-, так и с 64-битными аргументами
- Адреса имеют размер 64 бит
- Улучшения Advanced SIMD (NEON) enhanced
- С 16 до 32 увеличено количество 128-битных регистров, доступных через NEON, VFPv4, криптоинструкции AES, SHA
- Поддерживает вычисления с числами с плавающей запятой двойной точности (64-бит double)
- Полная совместимость с IEEE 754
- Новая система исключений
- Трансляция виртуальных адресов из 48-битного формата работает с помощью существующих механизмов LPAE

Функции RISC

Архитектура ARM обладает следующими особенностями RISC:

- Архитектура загрузки/хранения
- Нет поддержки нелинейного (не выровненного по словам) доступа к памяти (теперь поддерживается в процессорах ARMv6, за некоторыми исключениями, и полностью в ARMv7)
- Равномерный 16x32-битный регистровый файл
- Фиксированная длина команд (32 бита) для упрощения декодирования за счет снижения плотности кода. Позднее режим Thumb повысил плотность кода.
- Однотактное исполнение

Чтобы компенсировать простой дизайн, в сравнении с современными процессорами вроде Intel 80286 или Motorola 68020 были использованы некоторые особенности дизайна:

- Арифметические инструкции заменяют условные коды, только когда это необходимо
- 32-битное многорегистровое циклическое сдвиговое устройство, которое может быть использовано без потерь производительности в большинстве арифметических инструкций и адресных расчетов.
- Мощные индексированные адресные режимы
- Регистр ссылок для быстрого вызова функций листов
- Простые, но быстрые, с двумя уровнями приоритетов подсистемы прерываний с включенными банками регистров.

Условное исполнение

Одним из существенных отличий архитектуры ARM (изначальная архитектура) от других архитектур ЦПУ является так называемая *предикация* — возможность условного исполнения команд. Под «условным исполнением» здесь понимается то, что команда будет выполнена или проигнорирована в зависимости от текущего состояния флагов состояния процессора. В Thumb и Arm 64 предикация не используется — в первом режиме для неё нет места в команде (всего 16 бит), а во втором — предикация бессмысленна и сложна для реализации на суперскалярных архитектурах.

В то время как для других архитектур таким свойством, как правило, обладают только команды условных переходов, в архитектуру ARM была заложена возможность условного исполнения практически любой команды. Это было достигнуто добавлением в коды их инструкций особого 4-битового поля (**предиката**). Одно из его значений зарезервировано на то, что инструкция должна быть выполнена безусловно, а остальные кодируют то или иное сочетание условий (флагов). С одной стороны, с учётом ограниченности общей длины инструкции, это сократило число битов, доступных для кодирования смещения в командах обращения к памяти, но с другой — позволило избавляться от инструкций ветвления при генерации кода для небольших if-блоков.

Пример, обычно рассматриваемый для иллюстрации — основанный на вычитании алгоритм Евклида. В языке C он выглядит так:

```
while (i != j) {
    if (i > j) {
        i -= j;
    } else {
        j -= i;
    }
}
```

А на ассемблере ARM — так:

```
loop CMP Ri, Rj      ; set condition «NE» if (i != j),
                    ;           "GT" if (i > j),
                    ;           or "LT" if (i < j)
SUBGT Ri, Ri, Rj    ; if "GT" (greater than), i = i-j;
SUBLT Rj, Rj, Ri    ; if "LT" (less than), j = j-i;
BNE  loop          ; if "NE" (not equal), then loop
```

Из кода видно, что использование предикации позволило полностью избежать ветвления в операторах else и then. Заметим, что если Ri и Rj равны, то ни одна из SUB-инструкций не будет выполнена, полностью убирая необходимость в ветке, реализующей проверку while при каждом начале цикла, что могло быть реализовано, например, при помощи инструкции SUBLE (меньше либо равно).

Один из способов, которым уплотнённый (Thumb) код достигает большей экономии объёма — это именно удаление 4-битового предиката из всех инструкций, кроме ветвлений.

Другие особенности

Другая особенность набора команд — это возможность соединять сдвиги и вращения в инструкции «обработки информации» (арифметическую, логическую, движение регистр-регистр) так, что, например, выражение C:

```
a += (j << 2);
```

может быть преобразовано в команду из одного слова и одного цикла в ARM:

```
ADD Ra, Ra, Rj, LSL #2
```

Это приводит к тому, что типичные программы ARM становятся плотнее, чем обычно, с меньшим доступом к памяти. Таким образом, конвейер используется гораздо более эффективно. Даже несмотря на то, что ARM работает на скоростях, которые многие бы сочли низкими, он довольно-таки легко конкурирует с многими более сложными архитектурами ЦПУ.

ARM-процессор также имеет некоторые особенности, редко встречающиеся в других архитектурах RISC — такие, как адресация относительно счетчика команд (на самом деле счетчик команд ARM является одним из 16 регистров), а также пре- и постинкрементные режимы адресации.

Другая особенность, которую стоит отметить, — это то, что некоторые ранние ARM-процессоры (до ARM7TDMI), например, не имеют команд для хранения 2-байтных чисел. Таким образом, строго говоря, для них невозможно сгенерировать эффективный код, который бы вел себя так, как ожидается от объектов C типа «volatile int16_t».

Конвейер и другие аспекты реализации

ARM7 и более ранние версии имеют трехступенчатый конвейер. Это ступени переноса, декодирования и исполнения. Более производительные архитектуры, типа ARM9, имеют более сложные конвейеры. Cortex-a8 имеет 13-ступенчатый конвейер.

Сопроцессоры

Архитектура предоставляет способ расширения набора команд, используя сопроцессоры, которые могут быть адресованы, используя MCR, MRC, MRRC, MCRR и похожие команды. Пространство сопроцессора логически разбито на 16 сопроцессоров с номерами от 0 до 15, причем 15-й зарезервирован для некоторых типичных функций управления, типа управления кэш-памятью и операции блока управления памятью (на процессорах, в которых они есть).

В машинах на основе ARM периферийные устройства обычно подсоединяются к процессору путём сопоставления их физических регистров в памяти ARM или в памяти сопроцессора, или путём присоединения к шинам, которые, в свою очередь, подсоединяются к процессору. Доступ к сопроцессорам имеет большее время ожидания, поэтому некоторые периферийные устройства проектируются для доступа в обоих направлениях. В остальных случаях разработчики чипов лишь пользуются механизмом интеграции сопроцессора. Например, движок обработки изображений должен состоять из малого ядра ARM7TDMI, совмещенного с сопроцессором, который поддерживает примитивные операции по обработке элементарных кодировок HDTV.

Усовершенствованный SIMD (NEON)

Расширение усовершенствованного SIMD, также называемое технологией NEON — это комбинированный 64- и 128-битный набор команд SIMD (single instruction multiple data), который обеспечивает стандартизованное ускорение для медиаприложений и приложений обработки сигнала. NEON может выполнять декодирование аудиоформата mp3 на частоте процессора в 10 МГц, и может работать с речевым кодеком GSM AMR (adaptive multi-rate) на

частоте не более 13 МГц. Он обладает внушительным набором команд, отдельными регистровыми файлами, и независимой системой исполнения на аппаратном уровне. NEON поддерживает 8-, 16-, 32-, 64-битную информацию целого типа, одинарной точности и с плавающей запятой, и работает в операциях SIMD по обработке аудио и видео (графика и игры). В NEON SIMD поддерживает до 16 операций одновременно.

Одним из недостатков (или, скажем, особенностью) усовершенствованного SIMD является то, что сопроцессор выполняет команды усовершенствованного SIMD с достаточно значительной задержкой относительно кода основного процессора, задержка достигает двух десятков тактов и более (зависит от архитектуры и конкретных условий). По этой причине при попытке основного процессора воспользоваться результатами вычисления сопроцессора исполнение будет заморожено на значительное время.

VFP

Технология VFP (Vector Floating Point, вектора чисел с плавающей запятой) — расширение сопроцессора в архитектуре ARM. Она производит низкозатратные вычисления над числами с плавающей запятой одинарной/двойной точности, в полной мере соответствующие стандарту ANSI/IEEE Std 754—1985 Standard for Binary Floating-Point Arithmetic. VFP производит вычисления с плавающей запятой, подходящие для широкого спектра приложений — например, для КПК, смартфонов, сжатие звука, трёхмерной графики и цифрового звука, а также принтеров и телеприставок. Архитектура VFP также поддерживает исполнение коротких векторных команд. Но, поскольку процессор выполняет операции последовательно над каждым элементом вектора, то VFP нельзя назвать истинным SIMD-набором инструкций. Этот режим может быть полезен в графике и приложениях обработки сигнала, так как он позволяет уменьшить размер кода и выработку команд.

Другие сопроцессоры с плавающей запятой и/или SIMD, находящиеся в ARM-процессорах, включают в себя FPA, FPE, iwMMXt. Они обеспечивают ту же функциональность, что и VFP, но не совместимы с ним на уровне опкодов.

Расширения безопасности

Расширения безопасности, позиционируемые как TrustZone Technology, находятся в ARMv6KZ и других, более поздних, профилированных на приложениях архитектурах. Оно обеспечивает низкозатратную альтернативу добавлению специального ядра безопасности, обеспечивая 2 виртуальных процессора, поддерживаемых аппаратным контролем доступа. Это позволяет ядру приложения переключаться между двумя состояниями, называемыми «миры» (чтобы избежать путаницы с названиями возможных доменов), чтобы не допустить утечку информации из более важного мира в менее важный. Этот переключатель миров обычно ортогонален всем другим возможностям процессора. Таким образом, каждый мир может работать независимо от других миров, используя одно и то же ядро. Память и периферия соответственно изготавливаются с учетом особенностей мира ядра, и могут использовать это, чтобы получить контроль доступа к секретам и кодам ядра. Типичные приложения TrustZone Technology должны запускать полноценную операционную систему в менее важном мире, и компактный, специализированный на безопасности, код в более важном мире, позволяя Digital Rights Management'у намного точнее контролировать использование медиа на устройствах на базе ARM, и предотвращая несанкционированный доступ к устройству.

На практике же, так как конкретные детали реализации TrustZone остаются собственностью компании и не разглашаются, остается неясным, какой уровень безопасности гарантируется для этой модели угроз.

Отладка

Все современные процессоры ARM имеют аппаратные средства отладки, так как без них отладчики ПО не смогли бы выполнить самые базовые операции типа останова, отступа, установки контрольных точек после перезагрузки.

Архитектура ARMv7 определяет базовые средства отладки на архитектурном уровне. К ним относятся точки останова, точки просмотра и выполнение команд в режиме отладки. Такие средства были также доступны с модулем отладки EmbeddedICE. Поддерживаются оба режима — останова и обзора. Реальный транспортный механизм, который используется для доступа к средствам отладки, не специфицирован архитектурно, но реализация, как правило, включает поддержку JTAG.

Существует отдельная архитектура отладки «с обзором ядра», которая не требуется архитектурно процессорами ARMv7.

Регистры

ARM предоставляет 31 регистр общего назначения разрядностью 32 бит. В зависимости от режима и состояния процессора пользователь имеет доступ только к строго определённым набору регистров. В ARM state разработчику постоянно доступны 17 регистров:

- 13 регистров общего назначения (r0..r12).
- Stack Pointer (r13) — содержит указатель стека выполняемой программы.
- Link register (r14) — содержит адрес возврата в инструкциях ветвления.
- Program Counter (r15) — биты [31:1] содержат адрес выполняемой инструкции.
- Current Program Status Register (CPSR) — содержит флаги, описывающие текущее состояние процессора. Модифицируется при выполнении многих инструкций: логических, арифметических, и др.

Во всех режимах, кроме User mode и System mode, доступен также Saved Program Status Register (SPSR). После возникновения исключения регистр CPSR сохраняется в SPSR. Тем самым фиксируется состояние процессора (режим, состояние; флаги арифметических, логических операций, разрешения прерываний) на момент непосредственно перед прерыванием^[52].

usr	sys	svc	abt	und	irq	fiq
R0						
R1						
R2						
R3						
R4						
R5						
R6						
R7						
R8						R8_fiq
R9						R9_fiq
R10						R10_fiq
R11						R11_fiq
R12						R12_fiq
R13	R13_svc	R13_abt	R13_und	R13_irq	R13_fiq	
R14	R14_svc	R14_abt	R14_und	R14_irq	R14_fiq	
R15						
CPSR						
SPSR_svc		SPSR_abt	SPSR_und	SPSR_irq	SPSR_fiq	

Псевдонимы регистров:

рег.	функц.	args	vars	GCC	Comment
R0		a1			aN - аргумент функции
R1		a2			
R2		a3			
R3		a4			
R4			v1		vN - регистр-переменная
R5			v2		
R6			v3		
R7			v4		
R8			v5		
R9	SB		v6		SB - Статический базовый регистр
R10			v7	SL	SL - ?
R11			v8	FP	FP - Адрес <u>стекового кадра</u> ?
R12	IP				IP - Intra-procedure call scratch register
R13	SP				Stack pointer
R14	LR				Link register
R15	PC				Program counter

Работа с памятью

Архитектура ARM использует единое адресное пространство^[53]. На практике такая схема означает, что адрес может указывать на память (RAM или ROM) или порты ввода-вывода, в противовес схеме, при которой порты ввода-вывода имеют собственное адресное пространство.

Поддерживаемые системы ввода-вывода

Большинство существующих продуктов с использованием архитектуры ARM представляет собой законченные системы на кристалле, имеющие возможность работы с внешней динамической оперативной памятью (DRAM) и содержащие в себе контроллеры множества периферийных шин, в частности USB, ИИС (I2C), звуковые устройства, контроллеры для работы с флэш-носителями стандартов SD и MMC, контроллеры дисплеев и цифровых камер (MIPI).

Все процессоры имеют линии ввода-вывода общего назначения (GPIO). В потребительских устройствах к ним могут быть подключены кнопки «быстрого запуска» и регуляции громкости, сигнальные светодиоды, клавиатура и т. п.

Процесс запуска ОС на ARM-машинах

После включения системы на базе ARM-процессора из ROM-памяти загружается начальный загрузчик и адрес его точки входа. Начальный загрузчик проводит предварительную инициализацию системы, исполняя тем самым ту же роль, которую исполняет BIOS на системах x86, после чего может загрузить либо системный загрузчик, либо напрямую ОС.

Единого стандарта на начальный загрузчик не существует. Может применяться загрузчик U-Boot, а для 64-разрядной ARMv8 часто используется UEFI. Во многих применениях используются собственные загрузчики.

ОС, поддерживающие ARM

Архитектура ARM поддерживается множеством операционных систем. Наиболее широко используемые: Linux (в том числе Android), iOS, Windows Phone.

Работать на системах с ARM-процессором могут различные Unix и Unix-подобные ОС: Linux (многие дистрибутивы), iOS, Android, BSD (FreeBSD, NetBSD, OpenBSD), QNX, Plan 9, Inferno, OpenSolaris (2008—2009^[54]), Firefox OS, macOS Big Sur, MINIX 3.

Также на платформе запускаются отдельные варианты семейства Windows: Windows CE, Windows Phone, Windows RT, Windows 10, Windows 11.

Кроме того, ARM поддерживают: A2, FreeRTOS, Nucleus, Symbian OS, RISC OS, RISC iX.

См. также

- OpenRISC — архитектура 2000 года с GPL-реализацией or1k
- LEON — свободные реализации (GPL, LGPL) архитектуры SPARC V8, появившиеся в 1997 году
- OpenSPARC — свободная (GPL) реализация архитектуры SPARC V9 от 2005 года
- OpenPOWER — архитектура IBM Power, основанная в 2013 году IBM, Google, Mellanox, NVIDIA
- RISC-V — свободная и открытая архитектура и система команд для микропроцессоров и микроконтроллеров, созданная в 2010 году
- MIPS — наборы команд и архитектура MIPS

Примечания

1. Компания ARM Limited занимается исключительно разработкой ядер и инструментов для них (компиляторы, средства отладки и т. п.), зарабатывая на лицензировании архитектуры сторонним производителям.
2. Система на кристалле 1892BM14Я (<http://multicore.ru/index.php?id=1335>). multicore.ru. Дата обращения: 28 декабря 2018. Архивировано (<https://web.archive.org/web/20181229032714/http://multicore.ru/index.php?id=1335>) 29 декабря 2018 года.
3. Д. Козлов-Кононов. Процессорные ядра семейства Cortex. Сочетание высокой производительности и низкого энергопотребления (<http://www.electronics.ru/journal/article/135>) Архивная копия (<https://web.archive.org/web/20131124083110/http://www.electronics.ru/journal/article/135>) от 24 ноября 2013 на Wayback Machine, журнал Электроника, вып. #8/2010
4. Ознакомительное руководство по ARM-микроконтроллерам Cortex-M3 (http://www.gaw.ru/html.cgi/txt/doc/micros/arm/cortex_arh/index.htm). Дата обращения: 29 сентября 2013. Архивировано (https://web.archive.org/web/20130927031809/http://www.gaw.ru/html.cgi/txt/doc/micros/arm/cortex_arh/index.htm) 27 сентября 2013 года.
5. «ARMed for the living room» (http://news.cnet.com/ARMed-for-the-living-room/2100-1006_3-6056729.html) Архивная копия (https://web.archive.org/web/20121109215746/http://news.cnet.com/ARMed-for-the-living-room/2100-1006_3-6056729.html) от 9 ноября 2012 на Wayback Machine.

6. «An interview with Steve Furber» (<http://cacm.acm.org/magazines/2011/5/107684-an-interview-with-steve-furber/fulltext>) Архивная копия (<https://web.archive.org/web/20110727233519/http://cacm.acm.org/magazines/2011/5/107684-an-interview-with-steve-furber/fulltext>) от 27 июля 2011 на Wayback Machine // Communications of the ACM, May 2011, Vol. 54 No. 5, Pages 34-39, doi:10.1145/1941487.1941501
7. *Manners, David*. ARM's way (<http://www.electronicweekly.com/Articles/29/04/1998/7242/ARM39s-way.htm>), *Electronics Weekly* (29 апреля 1998). Архивировано (<https://web.archive.org/web/20120729024818/http://www.electronicweekly.com/Articles/29/04/1998/7242/ARM39s-way.htm>) 29 июля 2012 года. Дата обращения: 26 октября 2012.
8. Sophie Wilson — CHM (<https://www.computerhistory.org/fellowawards/hall/sophie-wilson/>). Дата обращения: 4 апреля 2019. Архивировано (<https://web.archive.org/web/20190404145431/https://www.computerhistory.org/fellowawards/hall/sophie-wilson/>) 4 апреля 2019 года.
9. *Santanu Chattopadhyay*. *Embedded System Design* (<https://books.google.com/books?id=4Bir9Kw059gC&pg=PA9>) (неопр.). — PHI Learning Pvt. Ltd., 2010. — С. 9. — ISBN 978-81-203-4024-4. Архивировано (<https://web.archive.org/web/20170227090422/https://books.google.com/books?id=4Bir9Kw059gC&pg=PA9>) 27 февраля 2017 года.
10. Apache Benchmarks for Calxeda's 5-Watt Web Server — ARM Servers, Now! (<https://web.archive.org/web/20121030065429/http://armservers.com/2012/06/18/apache-benchmarks-for-calxeda-5-watt-web-server/>) Дата обращения: 26 ноября 2012. Архивировано из оригинала (<http://armservers.com/2012/06/18/apache-benchmarks-for-calxeda-5-watt-web-server/>) 30 октября 2012 года.
11. Cloudy with a chance of ARM. What the Microserver Market Means for Semiconductor Vendors (http://www.apm.com/global/x-gene/docs/2012_03_OPP%20Cloudy%20with%20a%20Chance%20of%20ARM.pdf) Архивная копия (https://web.archive.org/web/20121202024037/http://www.apm.com/global/x-gene/docs/2012_03_OPP%20Cloudy%20with%20a%20Chance%20of%20ARM.pdf) от 2 декабря 2012 на Wayback Machine // Oppenheimer & Co. Inc, March 30, 2012
12. Exclusive : ARM Cortex-A15 «40 Per Cent» Faster Than Cortex-A9 (<http://www.itproportal.com/2011/03/14/exclusive-arm-cortex-a15-40-cent-faster-cortex-a9/>). Дата обращения: 5 октября 2013. Архивировано (<https://web.archive.org/web/20110721081000/http://www.itproportal.com/2011/03/14/exclusive-arm-cortex-a15-40-cent-faster-cortex-a9/>) 21 июля 2011 года.
13. Cortex-A15 Processor (<http://www.arm.com/products/processors/cortex-a/cortex-a15.php>) (англ.). ARM. — Презентация процессоров семейства ARM CortexA15 MPCore на сайте производителя. Дата обращения: 20 апреля 2012. Архивировано (<https://www.webcitation.org/682zlNuG3?url=http://www.arm.com/products/processors/cortex-a/cortex-a15.php>) 30 мая 2012 года.
14. ARM Cortex-A15 — процессор с тактовой частотой до 2,5 ГГц, не только для смартфонов (<http://news.ferra.ru/hard/2010/09/10/102926/>). Ferra.ru (10 сентября 2010). Дата обращения: 20 апреля 2012. Архивировано (<https://web.archive.org/web/20120426112040/http://news.ferra.ru/hard/2010/09/10/102926/>) 26 апреля 2012 года.
15. Intel Newsroom | Top headlines, breaking news and current events from Intel (<https://web.archive.org/web/20090306235150/http://www.intel.com/pressroom/archive/releases/20060627corp.htm>). Дата обращения: 4 октября 2013. Архивировано из оригинала (<http://www.intel.com/pressroom/archive/releases/20060627corp.htm>) 6 марта 2009 года.
16. «ARM810 — Dancing to the Beat of a Different Drum» (http://www.hotchips.org/archives/hc8/2_Mon/HC8.S4/HC8.4.1.pdf) Архивная копия (https://web.archive.org/web/20080910025355/http://www.hotchips.org/archives/hc8/2_Mon/HC8.S4/HC8.4.1.pdf) от 10 сентября 2008 на Wayback Machine Архивировано (https://web.archive.org/web/20110723211338/http://www.hotchips.org/archives/hc8/2_Mon/HC8.S4/HC8.4.1.pdf) 23 июля 2011 года. ARM Holdings presentation at Hot Chips 1996-08-07.
17. Register 13, FCSE PID register (<http://infocenter.arm.com/help/topic/com.arm.doc.ddi0151c/I47491.html>) Архивная копия (<https://web.archive.org/web/20110707164527/http://infocenter.arm.com/help/topic/com.arm.doc.ddi0151c/I47491.html>) от 7 июля 2011 на Wayback Machine ARM920T Technical Reference Manual

18. [Neo1973: GTA01Bv4 versus GTA02 comparison](http://wiki.openmoko.org/wiki/Neo1973:_GTA01Bv4_versus_GTA02_comparison) (http://wiki.openmoko.org/wiki/Neo1973:_GTA01Bv4_versus_GTA02_comparison). Дата обращения: 15 ноября 2007. Архивировано (https://www.webcitation.org/667ojWDRI?url=http://wiki.openmoko.org/wiki/Neo1973:_GTA01Bv4_versus_GTA02_comparison) 13 марта 2012 года.
19. [S3C2410](http://elinux.org/S3C2410) (<http://elinux.org/S3C2410>). Дата обращения: 13 января 2010. Архивировано (<https://www.webcitation.org/667окGpJS?url=http://elinux.org/S3C2410>) 13 марта 2012 года.
20. [Rockbox Samsung SA58xxx series](http://www.rockbox.org/twiki/bin/view/Main/SamsungSA58) (<http://www.rockbox.org/twiki/bin/view/Main/SamsungSA58>). Дата обращения: 22 февраля 2008. Архивировано (<https://www.webcitation.org/667окsgNi?url=http://www.rockbox.org/wiki/SamsungSA58>) 13 марта 2012 года.
21. [Rockbox Meizu M6 Port – Hardware Information](http://www.rockbox.org/twiki/bin/view/Main/MeizuM6Port) (<http://www.rockbox.org/twiki/bin/view/Main/MeizuM6Port>). Дата обращения: 22 февраля 2008. Архивировано (<https://www.webcitation.org/667olfDjt?url=http://www.rockbox.org/wiki/MeizuM6Port>) 13 марта 2012 года.
22. [Datasheets — Magic Lantern Firmware Wiki](http://magiclantern.wikia.com/wiki/Datasheets) (<http://magiclantern.wikia.com/wiki/Datasheets>). Дата обращения: 6 июня 2010. Архивировано (<https://web.archive.org/web/20110718045528/http://magiclantern.wikia.com/wiki/Datasheets>) 18 июля 2011 года.
23. [STR9 – STR912 – STR912FW44 microcontroller – documents and files download page](http://mcu.st.com/mcu/modules.php?name=mcu&file=devicedocs&DEV=STR912FW44&FAM=101) (<http://mcu.st.com/mcu/modules.php?name=mcu&file=devicedocs&DEV=STR912FW44&FAM=101>). Mcu.st.com. Дата обращения: 18 апреля 2009. Архивировано (<https://web.archive.org/web/20070210162600/http://mcu.st.com/mcu/modules.php?name=mcu&file=devicedocs&DEV=STR912FW44&FAM=101>) 10 февраля 2007 года.
24. [Starlet](http://wiibrew.org/wiki/Starlet) (<http://wiibrew.org/wiki/Starlet>) Архивная копия (<https://web.archive.org/web/20181010144616/http://wiibrew.org/wiki/Starlet>) от 10 октября 2018 на [Wayback Machine](http://www.waybackmachine.org/).
25. [Benchmarks – Albatross](http://www.albatross-uav.org/index.php/Benchmarks) (<http://www.albatross-uav.org/index.php/Benchmarks>). Albatross-uav.org (18 июня 2005). Дата обращения: 18 апреля 2009. Архивировано (<https://web.archive.org/web/20081016070519/http://www.albatross-uav.org/index.php/Benchmarks>) 16 октября 2008 года.
26. [ARM1136J\(F\)-S – ARM Processor](http://www.arm.com/products/CPUs/ARM1136JF-S.html) (<http://www.arm.com/products/CPUs/ARM1136JF-S.html>). Arm.com. Дата обращения: 18 апреля 2009. Архивировано (<https://www.webcitation.org/667omPSMh?url=http://www.arm.com/products/processors/classic/arm11/arm1136.php>) 13 марта 2012 года.
27. [Qualcomm chips kernel ARM - from phones to laptops](http://xi0.info/qualcomm-chips-kernel-arm-phones_7000.html) (http://xi0.info/qualcomm-chips-kernel-arm-phones_7000.html). xi0.info. Дата обращения: 8 мая 2010. Архивировано (https://www.webcitation.org/667onN6Aq?url=http://xi0.info/qualcomm-chips-kernel-arm-phones_7000.html) 13 марта 2012 года.
28. [Qualcomm MSM7227 RISC Chipset](http://pdadb.net/index.php?m=cpu&id=a7227&c=qualcomm_msm7227) (http://pdadb.net/index.php?m=cpu&id=a7227&c=qualcomm_msm7227). pdadb.net. Дата обращения: 8 мая 2010. Архивировано (<https://www.webcitation.org/667ооM6DX?url=http://pdadb.net/index.php?m=cpu>) 13 марта 2012 года.
29. [GoForce 6100](http://www.nvidia.com/page/goforce_6100.html) (http://www.nvidia.com/page/goforce_6100.html). Nvidia.com. Дата обращения: 18 апреля 2009. Архивировано (https://www.webcitation.org/667oprK0P?url=http://www.nvidia.com/page/goforce_6100.html) 13 марта 2012 года.
30. [Mediatek MT6573](http://www.mediatek.com/en/Products/featured_content.php?sn=2) (http://www.mediatek.com/en/Products/featured_content.php?sn=2). <http://www.mediatek.com>.+Дата обращения: 18 апреля 2009. Архивировано (https://www.webcitation.org/68DLO2pD5?url=http://www.mediatek.com/en/Products/featured_content.php?sn=2) 6 июня 2012 года.
31. [Samsung S3C6410 and S3C6430 Series ARM Processors](https://web.archive.org/web/20090901161759/http://www.samsung.com/global/business/semiconductor/productInfo.do?fmly_id=229&partnum=S3C6410) (https://web.archive.org/web/20090901161759/http://www.samsung.com/global/business/semiconductor/productInfo.do?fmly_id=229&partnum=S3C6410). Samsung. Дата обращения: 8 октября 2009. Архивировано из оригинала (http://www.samsung.com/global/business/semiconductor/productInfo.do?fmly_id=229&partnum=S3C6410) 1 сентября 2009 года., and the Qualcomm MSM7627 as seen in the [Palm Pixi](#) and [Motorola Calgary/Devour](#)
32. *Merrit, Rick* "ARM stretches out with A5 core, graphics, FPGAs" (<http://www.eetimes.com/showArticle.jhtml?articleID=220700447>). EE Times (21 октября 2009). Дата обращения: 28 октября 2009. Архивировано (<https://www.webcitation.org/667ov2zwD?url=http://eetimes.com/electronics-news/4085371/ARM-stretches-out-with-A5-core-graphics-FPGAs>) 13 марта 2012 года.

33. *Clarke, Peter* ARM tips plans for Swift and Sparrow processor cores (http://www.eetimes.com/news/semi/rss/showArticle.jhtml?articleID=213000665&cid=RSSfeed_eetimes_semiRSS). EE Times (3 февраля 2009). Дата обращения: 18 апреля 2009. Архивировано (<https://www.webcitation.org/667ovsxRU?url=http://eetimes.com/electronics-products/processors/4110314/ARM-tips-plans-for-Swift-and-Sparrow-processor-cores>) 13 марта 2012 года.
34. *Segan, Sascha* ARM's Multicore Chips Aim for Netbooks (<https://www.pcmag.com/article2/0,2817,2341032,00.asp>). PC Magazine (9 апреля 2009). Дата обращения: 18 апреля 2009. Архивировано (<https://www.webcitation.org/667owcBPe?url=http://www.pcmag.com/article2/0,2817,2341032,00.asp>) 13 марта 2012 года.
35. æ™æ°- (<http://pc.watch.impress.co.jp/video/pcw/docs/423/409/p1.pdf>) Архивная копия (<https://web.archive.org/web/20130729025904/http://pc.watch.impress.co.jp/video/pcw/docs/423/409/p1.pdf>) от 29 июля 2013 на Wayback Machine Архивировано (<https://web.archive.org/web/20130729025904/http://pc.watch.impress.co.jp/video/pcw/docs/423/409/p1.pdf>) 29 июля 2013 года.
36. Cortex-A15 Processor — ARM (<http://www.arm.com/products/processors/cortex-a/cortex-a15.php>). Дата обращения: 24 октября 2012. Архивировано (<https://www.webcitation.org/682zINuG3?url=http://www.arm.com/products/processors/cortex-a/cortex-a15.php>) 30 мая 2012 года.
37. Cortex-A7 Processor — ARM (<http://www.arm.com/products/processors/cortex-a/cortex-a7.php>). Дата обращения: 24 октября 2012. Архивировано (<https://web.archive.org/web/20160531034437/http://www.arm.com/products/processors/cortex-a/cortex-a7.php>) 31 мая 2016 года.
38. *Benz, Benjamin* Cortex Nachwuchs bei ARM (<http://www.heise.de/newsticker/meldung/Cortex-Nachwuchs-bei-ARM-936412.html>). Heise.de (2 февраля 2010). Дата обращения: 3 мая 2010. Архивировано (<https://www.webcitation.org/667охрYk?url=http://www.heise.de/newsticker/meldung/Cortex-Nachwuchs-bei-ARM-936412.html>) 13 марта 2012 года.
39. *Clarke, Peter* ARM preps tiny core for low-power microcontrollers (<http://www.eetimes.com/news/design/rss/showArticle.jhtml?articleID=214502333>). EE Times (23 февраля 2009). Дата обращения: 30 ноября 2009. Архивировано (<https://www.webcitation.org/667ozlX8G?url=http://eetimes.com/electronics-products/processors/4110516/ARM-preps-tiny-core-for-low-power-microcontrollers>) 13 марта 2012 года.
40. *Walko, John* NXP first to demo ARM Cortex-M0 silicon (<http://www.eetimes.com/news/design/rss/showArticle.jhtml?articleID=216200026>). EE Times (23 марта 2009). Дата обращения: 29 июня 2009. Архивировано (<https://www.webcitation.org/667p02jg6?url=http://eetimes.com/electronics-products/processors/4110888/NXP-first-to-demo-ARM-Cortex-M0-silicon>) 13 марта 2012 года.
41. ARM Powered VCAs " Triad Semiconductor (<http://www.triadsemi.com/services/arm-powered-vcas/>) Архивная копия (<https://web.archive.org/web/20110716112427/http://www.triadsemi.com/services/arm-powered-vcas/>) от 16 июля 2011 на Wayback Machine Архивировано (<https://web.archive.org/web/20110716112427/http://www.triadsemi.com/services/arm-powered-vcas/>) 16 июля 2011 года.
42. Cortex-M0 used in low power touch controller — 10/06/2009 — Electronics Weekly (<http://www.electronicsworld.com/Articles/2009/06/10/46252/cortex-m0-used-in-low-power-touch-controller.htm>). Дата обращения: 6 июня 2010. Архивировано (<https://web.archive.org/web/20110816075034/http://www.electronicsworld.com/Articles/2009/06/10/46252/cortex-m0-used-in-low-power-touch-controller.htm>) 16 августа 2011 года.
43. Chungbuk Technopark Chooses ARM Cortex-M0 Processor (<http://www.design-reuse.com/news/22137/arm-cortex-m0-processor.html>). Дата обращения: 6 июня 2010. Архивировано (<https://web.archive.org/web/20101007045935/http://www.design-reuse.com/news/22137/arm-cortex-m0-processor.html>) 7 октября 2010 года.
44. Google Translate (<http://translate.google.co.uk/translate?hl=en&sl=zh-TW&u=http%3A%2F%2Fwww.nuvoton.com%2Fhq%2Fenu%2FNewsAndEvents%2FNews%2FMediaCoverage%2F20091005.htm>). Дата обращения: 3 мая 2022. Архивировано (<https://web.archive.org/web/20200122191515/http://translate.google.co.uk/translate?hl=en&sl=zh-TW&u=http%3A%2F%2Fwww.nuvoton.com%2Fhq%2Fenu%2FNewsAndEvents%2FNews%2FMediaCoverage%2F20091005.htm>) 22 января 2020 года.
45. Austriamicrosystems Chooses ARM Cortex-M0 Processor For Mixed Signal Applications (http://www10.edacafe.com/nbc/articles/view_article.php?articleid=752000). Дата обращения: 6 июня 2010. Архивировано (https://web.archive.org/web/20110722230112/http://www10.edacafe.com/nbc/articles/view_article.php?articleid=752000) 22 июля 2011 года.

46. «ARM Extends Cortex Family with First Processor Optimized for FPGA» (<http://www.arm.com/news/17017.html>) Архивная копия (<https://web.archive.org/web/20070505180243/http://www.arm.com/news/17017.html>) от 5 мая 2007 на [Wayback Machine](#), ARM press release, March 19, 2007. Retrieved April 11, 2007.
47. «ARM Cortex-M1» (http://www.arm.com/products/CPUs/ARM_Cortex-M1.html) Архивная копия (https://web.archive.org/web/20070401051142/http://www.arm.com/products/CPUs/ARM_Cortex-M1.html) от 1 апреля 2007 на [Wayback Machine](#), ARM product website. Retrieved April 11, 2007.
48. ARM Extends Cortex Family with First Processor Optimized for FPGA (<http://www.arm.com/news/17017.html>) Архивная копия (<https://web.archive.org/web/20070505180243/http://www.arm.com/news/17017.html>) от 5 мая 2007 на [Wayback Machine](#).
49. Category:Opcodes - ARMwiki (<http://www.heyrick.co.uk/armwiki/Category:Opcodes>) (англ.). www.heyrick.co.uk. Дата обращения: 6 августа 2018. Архивировано (<https://web.archive.org/web/20180806145431/http://www.heyrick.co.uk/armwiki/Category:Opcodes>) 6 августа 2018 года.
50. ARM (https://web.archive.org/web/20180610181021/https://www.arm.com/files/downloads/ARMv8_Architecture.pdf) . Дата обращения: 15 апреля 2013. Архивировано из оригинала (http://www.arm.com/files/downloads/ARMv8_Architecture.pdf) 10 июня 2018 года.
51. ARM Launches Cortex-A50 Series, the World’s Most… - ARM (<http://www.arm.com/about/newsroom/arm-launches-cortex-a50-series-the-worlds-most-energy-efficient-64-bit-processors.php>). Дата обращения: 17 апреля 2013. Архивировано (<https://www.webcitation.org/6G0m1ys2z?url=http://www.arm.com/about/newsroom/arm-launches-cortex-a50-series-the-worlds-most-energy-efficient-64-bit-processors.php>) 20 апреля 2013 года.
52. ARM7TDMI (rev 3) Technical Reference Manual (<http://infocenter.arm.com/help/topic/com.arm.doc.ddi0029g/DDI0029.pdf>) (англ.). Дата обращения: 22 августа 2009. Архивировано (<https://www.webcitation.org/667p0IzjD?url=http://infocenter.arm.com/help/topic/com.arm.doc.ddi0029g/DDI0029.pdf>) 13 марта 2012 года.
53. <http://www.peter-cockerell.net/aalp/html/ch-2.html> Архивная копия (<https://web.archive.org/web/20151208093837/http://www.peter-cockerell.net/aalp/html/ch-2.html>) от 8 декабря 2015 на [Wayback Machine](#) «Input and output devices are memory mapped. There is no concept of a separate I/O address space. Peripheral chips are read and written as if they were areas of memory. This means that in practical ARM systems, the memory map is divided into three areas: RAM, ROM, and input/output devices (probably in decreasing order of size).»
54. OpenSolaris Project: ARM Platform Port (<http://www.opensolaris.org/os/project/osarm/>). Sun Microsystems. Архивировано (<https://www.webcitation.org/667p9tu7w?url=http://hub.opensolaris.org/bin/view/Project+osarm/WebHome>) 13 марта 2012 года.

Ссылки

- [ARM Cortex — A9 white paper](https://web.archive.org/web/20141117060156/http://www.arm.com/files/pdf/ARMCortexA-9Processors.pdf) (<https://web.archive.org/web/20141117060156/http://www.arm.com/files/pdf/ARMCortexA-9Processors.pdf>) (англ.)
- [Архитектура и система команд RISC-процессоров семейства ARM](http://www.gaw.ru/html/cgi/txt/doc/micros/arm/index.htm) (<http://www.gaw.ru/html/cgi/txt/doc/micros/arm/index.htm>) (рус.)
- [Форум по ARM-микроконтроллерам](http://сахара.ru/arm.html) (<http://сахара.ru/arm.html>) (рус.)
- [Статья о WinCE и ARM](http://phrack.org/issues.html?issue=63&id=6#article) (<http://phrack.org/issues.html?issue=63&id=6#article>) в журнале [Phrack](#) (англ.)
- [Рейтинг производительности ARM процессоров](http://droiddevice.ru/arm-protsessory) (<http://droiddevice.ru/arm-protsessory>) (рус.)
- [Цикл статей по программированию ARM микроконтроллеров](http://www.mcu.by/category/arm/) (<http://www.mcu.by/category/arm/>) (рус.)
- [Реверс инжиниринг ARM1](https://habr.com/ru/post/411425/) (<https://habr.com/ru/post/411425/>)

Источник — [https://ru.wikipedia.org/w/index.php?title=ARM_\(архитектура\)&oldid=133936859](https://ru.wikipedia.org/w/index.php?title=ARM_(архитектура)&oldid=133936859)

Эта страница в последний раз была отредактирована 1 ноября 2023 в 18:49.

Текст доступен по лицензии Creative Commons «С указанием авторства — С сохранением условий» (CC BY-SA); в отдельных случаях могут действовать дополнительные условия.

Wikipedia® — зарегистрированный товарный знак некоммерческой организации Фонд Викимедиа (Wikimedia Foundation, Inc.)