ARM (Advanced RISC Machine)

Архитектура ARM (Advanced RISC Machine, Acorn RISC Machine — продвинутая ЭВМ с упрощённым набором команд, ПВМ / усовершенствованная RISC-машина) — семейство лицензируемых 32-битных и 64-битных микропроцессорных ядер, разрабатываемых компанией ARM Limited.

Среди лицензиатов: AMD, Apple, Analog Devices, Atmel, Xilinx, Altera, Cirrus Logic, Intel (до 27 июня 2006 года), Marvell, NXP, STMicroelectronics, Samsung, LG, MediaTek, MStar, Qualcomm, Sony, Texas Instruments, nVidia, Freescale, Миландр, HiSilicon.

Известные семейства процессоров: ARM7, ARM9, ARM11 и Cortex.

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

Процессоры ARM

Профили

Архитектура развивалась с течением времени, и, начиная с ARMv7, были определены три профиля:

- ‘П’ (Прикладной) / англ.‘A’ (application) — для устройств, требующих высокой производительности (смартфоны, планшеты)
- ‘В’ (реального Времени) / англ.‘R’ (real time) — для приложений, работающих в реальном времени,
- ’M’ (Микроконтроллер) / англ.’M’ (microcontroller) — для микроконтроллеров и недорогих встраиваемых устройств.

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

Примеры семейств процессоров ARM:

«Классические»

- 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

- Cortex A57 – для высокопроизводительных мобильных устройств и микросерверов
- Cortex-A76AE - с повышенной отказоустойчивостью и высокой производительнстью

Cortex R

- Cortex-R52 — производительный процессор для систем реального времени

Cortex M

- Cortex M35P — надежный процессор для встраиваемых embedded систем.

Neoverse

- Neoverse E1 - Для серверов и датацентров, ориентирован на сетевые и облачные решения, системы хранения данных
- Neoverse N1 - Для серверов и датацентров, обработка больших массивов данных.

Режимы

Процессор может находиться в одном из следующих операционных режимов:

- Пользовательский режим / англ. User mode — обычный режим выполнения программ. В этом режиме выполняется большинство программ.
- Режим быстрого прерывания / англ. Fast Interrupt (FIQ) — режим, где время срабатывания меньше.
- Основной режим прерывания / англ. Interrupt (IRQ).
- Системный режим / англ. System mode — защищённый режим для использования операционной системой.
- Режим аварийного отказа / англ. Abort mode — режим, куда переходит процессор, когда возникает ошибка доступа к памяти (доступ к данным или к команде на этапе prefetch-конвейера).
- Привилегированный пользовательский режим / англ. Supervisor mode.
- Неопределённый режим / англ. Undefined mode — режим, куда процессор входит при попытке выполнить неизвестную ему команду.

Режим процессора переключается, когда возникает соответствующее исключение, или же при модификации регистра статуса.

Особенности архитектуры

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 бит) для упрощения декодирования за счет снижения плотности кода. Позднее режим Thumb повысил плотность кода.
- Одноцикловое 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 листьев
- Простые, но быстрые, с двумя уровнями приоритетов подсистемы прерываний с включенными банками регистров.
-

Условное исполнение

Одно из существенных отличий архитектуры ARM (изначальная архитектура) от других архитектур ЦПУ — так называемая предикация, то есть 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 Arm64 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.

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 счётчика команд** (на самом деле

счётчик команд ARM — это один из 16 регистров), а также пре- и пост-инкрементные режимы адресации.

Другая особенность, что 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.

В то же время стоит отметить, что в **архитектурах Cortex A65AE и Cortex A76AE конвейер отсутствует (out-of-order)**, но при этом ядро остается суперскалярным

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 более 13МГц. 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 единовременно.

Один из недостатков (или же особенностью) усовершенствованного SIMD — 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 включают аппаратные средства отладки, так как без них отладчики ПО не смогли бы выполнить самые базовые операции типа остановки, 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 — остановки 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 определённому 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

Архитектура ARM использует единое адресное пространство.

На практике такая схема означает, что адрес может указывать на оперативную память, ПЗУ или порты ввода-вывода, в противовес традиционной схеме, при которой содержимое ПЗУ при запуске копируется в оперативную память, а порты ввода-вывода имеют собственное адресное пространство.

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.

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 переводятся 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, стартовавшая с ARM 1156 core, анонсированного 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.

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 бит

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.

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 работают 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

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: хотя современные версии стандарта UEFI предусматривают возможность реализации этого стандарта для архитектуры ARM, но на практике UEFI используется редко — большинство ОС используют собственные загрузчики, или реализуют совместимость с одним из загрузчиков других систем.

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), Firefox OS.

Также на платформе запускаются отдельные варианты семейства Windows: Windows CE, Windows Phone, Windows RT, Windows 10 (только на Raspberry Pi)

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

Источники

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. ↑ 2,0 2,1 Электроника НТБ [Электронный ресурс]: Д. Козлов-Кононов. Процессорные ядра семейства Cortex. Сочетание высокой производительности и низкого энергопотребления / журнал Электроника, вып. #8/2010. — Дата обращения: 13.11.2016. Режим доступа: http://www.electronics.ru/journal/article/135.

3. ↑ Справочник по электронным компонентам [Электронный ресурс]: Ознакомительное руководство по ARM-микроконтроллерам Cortex-M3 / Дата

обращения: 13.11.2016. Режим доступа: http://www.gaw.ru/html.cgi/txt/doc/micros/arm/cortex_arh/index.htm.

4. ↑ ITProPortal [Электронный ресурс]: Exclusive : ARM Cortex-A15 «40 Per Cent» Faster Than Cortex-A9 / Дата обращения: 13.11.2016. Режим доступа: http://www.itproportal.com/2011/03/14/exclusive-arm-cortex-a15-40-cent-faster-cortex-a9/.

5. ↑ ARM [Электронный ресурс]: Презентация процессоров семейства ARM Cortex-A15 MPCore / Дата обращения: 13.11.2016. Режим доступа: http://www.arm.com/products/processors/cortex-a/cortex-a15.php.

6. ↑ Ferra.ru [Электронный ресурс]: ARM Cortex-A15 — процессор с тактовой частотой до 2,5 ГГц, не только для смартфонов / Дата обращения: 13.11.2016. Режим доступа: http://news.ferra.ru/hard/2010/09/10/102926/.

7. ↑ ARM [Электронный ресурс]: Презентация «ARMv8 Technology Preview» / Дата обращения: 13.11.2016. Режим доступа: http://www.arm.com/files/downloads/ARMv8_Architecture.pdf.

8. ↑ ARM [Электронный ресурс]: ARM Launches Cortex-A50 Series, the World’s Most Energy-Efficient 64-bit Processors / Дата обращения: 13.11.2016. Режим доступа: http://www.arm.com/about/newsroom/arm-launches-cortex-a50-series-the-worlds-most-energy-efficient-64-bit-processors.php.

9. ↑ ARM [Электронный ресурс]: ARM7TDMI (rev 3) Technical Reference Manual / Дата обращения: 13.11.2016. Режим доступа: http://infocenter.arm.com/help/topic/com.arm.doc.ddi0029g/DDI0029.pdf.

10.↑ Pete's Pages [Электронный ресурс]: ARM Assembly Language Programming — Chapter 2. Inside the ARM / Дата обращения: 13.11.2016. Режим доступа: http://www.peter-cockerell.net/aalp/html/ch-2.html. «Порты ввода-вывода имеют отображение в ОЗУ. Тут нет понятия раздельного адресного пространства ввода-вывода. Периферийные чипы считываются и записываются, как если бы они были областями памяти. То есть в практических ARM-системах памяти распределяется на три зоны: ОЗУ, ПЗУ и порты ввода-вывода (вероятно, в порядке убывания размера)».

Ссылки

- Справочник по электронным компонентам [Электронный ресурс]: Архитектура и система команд RISC-процессоров семейства ARM / Дата обращения: 13.11.2016. Режим доступа: http://www.gaw.ru/html.cgi/txt/doc/micros/arm/index.htm.

- Сахара [Электронный ресурс]: Форум по ARM-микроконтроллерам / Дата обращения: 13.11.2016. Режим доступа: http://caxapa.ru/arm.html.

- Журнал Phrack [Электронный ресурс]: Статья о WinCE и ARM / Дата обращения: 13.11.2016. Режим доступа: http://phrack.org/issues.html?issue=63&id=6#article.

- DroidDevice.ru [Электронный ресурс]: Рейтинг производительности ARM-процессоров / Дата обращения: 13.11.2016. Режим доступа: http://droiddevice.ru/arm-protsessory.
- MCU.by — Embedded software [Электронный ресурс]: Цикл статей по программированию ARM-микроконтроллеров / Дата обращения: 13.11.2016. Режим доступа: http://www.mcu.by/category/arm/.
- ARM [Электронный ресурс]: Technologies / Дата обращения 02.03.17. Режим доступа: https://www.arm.com/products/processors/technologies/vector-floating-point.php
- Debian Wiki [Электронный ресурс]: ArmHardFloatPort / Дата обращения 02.03.17. Режим доступа: https://wiki.debian.org/ArmHardFloatPort#Background_information
- Хабрахабр [Электронный ресурс]: Перевод / Дата обращения: 01.03.2017. Режим доступа: https://habrahabr.ru/post/197854/
- ARM Neoverse N1 и E1 новые процессоры для дата-центров и серверов, до 128 ядер [Электронный ресурс]/ Режим доступа: https://www.hardwareluxx.ru/index.php/news/hardware/prozessoren/46561-arm-neoverse-n1-e1-.html
- Cortex-A76AE [Электронный ресурс]/ Режим доступа: https://developer.arm.com/products/processors/cortex-a/cortex-a76ae
- Национальная библиотека им. Н. Э. Баумана. ARM (Advanced RISC Machine) [Электронный ресурс]/ Режим доступа: https://ru.bmstu.wiki/index.php?title=ARM_(Advanced_RISC_Machine)&mobileaction=toggle_view_mobile