

Цифровые вычислительные устройства и микропроцессоры приборных комплексов

Микропроцессоры ARM

Соловьёв Сергей Юрьевич, канд. техн. наук, доцент кафедры 303

1. Общие сведения о микропроцессорах ARM

- **ARM Ltd.** (название происходит от **Advanced RISC Machines**) — британская корпорация, являющаяся одним из крупнейших разработчиков и лиценздателей современной архитектуры 32-разрядных RISC-процессоров, специально ориентированных для использования в портативных, мобильных устройствах (таких, как мобильные телефоны, персональные органайзеры и т.п.).
- Технология **ARM** оказалась весьма успешной и в настоящее время является доминирующей микропроцессорной архитектурой для портативных цифровых устройств.
- В настоящее время наибольшее распространение получили три семейства микропроцессоров **ARM**:
 - **ARM7** (обычно с тактовой частотой около 50 МГц, предназначенные, например, для недорогих мобильных телефонов);
 - **ARM9** (с частотами порядка 200 МГц для «продвинутых» телефонов и карманных компьютеров);
 - **ARM11** (новейшее семейство с расширенными мультимедийными возможностями, тактовая частота до 1 ГГц).

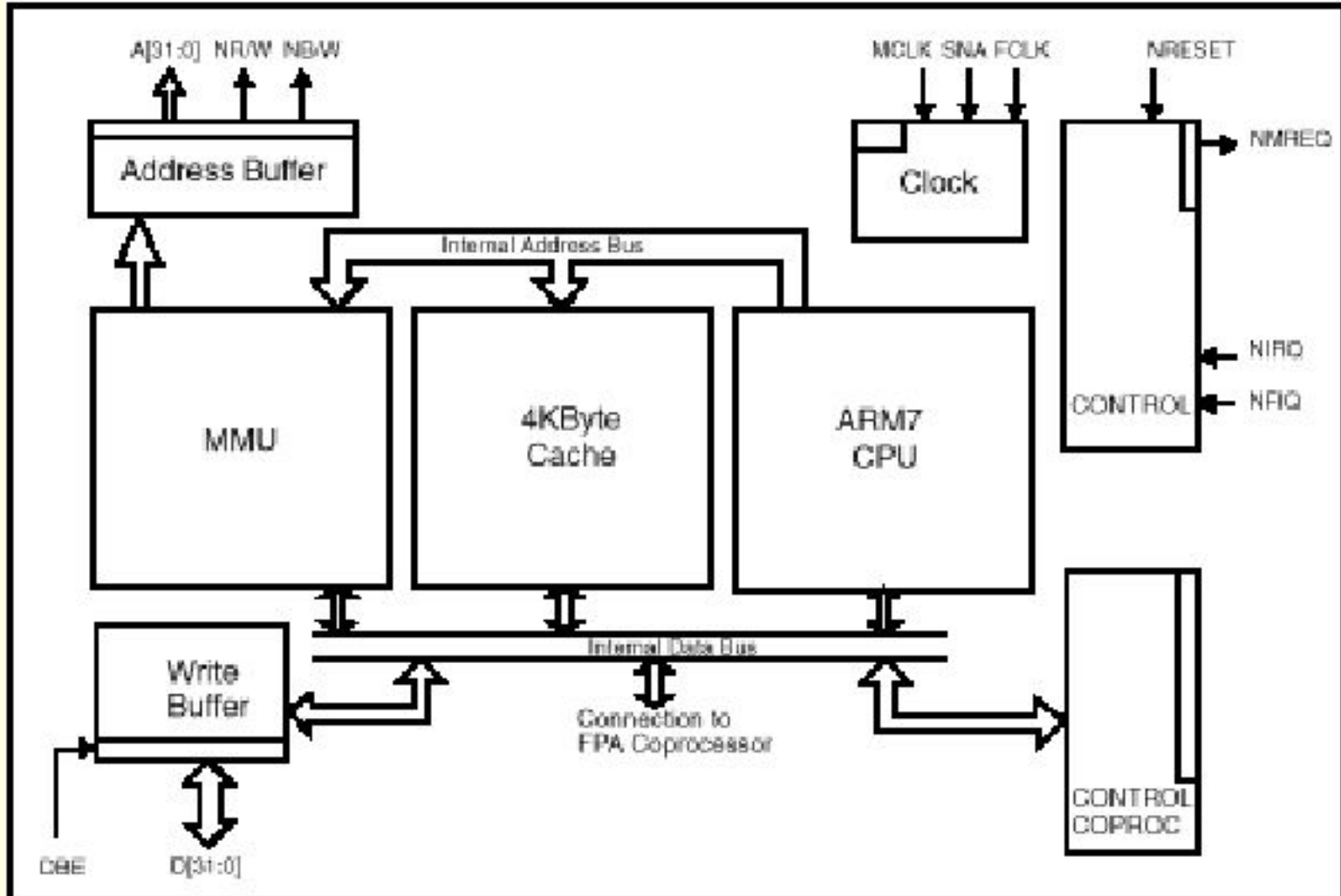
ARM 7

- Фирмой разработан целый ряд 32-разрядных RISC процессоров с различными возможностями и различной производительности, а её процессор ARM7, разработанный ещё в 1994 году, используется до настоящего времени.
- Сама фирма определяет процессор ARM7 как универсальное, с малым потреблением, ядро 32-разрядного RISC микропроцессора, предназначенное для использования в различных заказных и специальных ИС. Малые размеры RISC ядра позволяют успешно интегрировать его в большие заказные схемы, которые могут содержать RAM, ROM, DSP, дополнительную логику и другие элементы.

Основные характеристики ядра ARM7

- 32-разрядный RISC процессор (32-разрядные шины данных и адреса) с производительностью 17 MIPS при тактовой частоте 25 МГц (пиковая производительность 25 MIPS)
- 32-разрядная адресация - линейное адресное пространство в 4 Гбайта - исключает потребность в сегментированной, разделенной на банки или оверлейной памяти
- Тридцать один 32-разрядный регистр общего назначения и шесть регистров состояния
- Регистры адресов, записи и конвейера
- Циклическое сдвиговое устройство и перемножитель
- Трехуровневый конвейер (выборка команды, ее декодирование и выполнение)
- Рабочие режимы Big Endian и Little Endian
- Напряжение питания 3,3 и 5 В
- Полностью статическая работа, позволяющая дополнительно снижать потребление за счет уменьшения тактовой частоты, что идеально для критичных к потреблению применений
- Быстрый отклик на прерывания применений реального масштаба времени
- Поддержка систем виртуальной памяти
- Простая, но мощная система команд

Блок-схема ядра ARM7



Типы команд

- 32-разрядная система команд ядра ARM7 содержит одиннадцать базовых типов команд:
 - Два типа используют встроенное арифметико-логическое устройство, циклическое сдвиговое устройство и умножитель при операциях над данными в банке из 31 регистра, форматом по 32 разряда каждый;
 - Три класса команд управления перемещением данных между памятью и регистрами, один оптимизированный на обеспечение гибкости адресации, другой под быстрое контекстное переключение и третий под подкачку данных;
 - Три команды управляют потоком и уровнем привилегии выполнения;
 - Три типа предназначены для управления внешними сопроцессорами, что позволяет расширить функциональные возможности системы команд за пределами ядра.

Микроконтроллеры семейства ARM7

- Система команд ARM хорошо обрабатывается компиляторами языков высокого уровня. В отличие от некоторых RISC процессоров, процессор ARM7, при возникновении необходимости в некотором уменьшении объема кодов, допускает программирование и на ассемблере.
- Предоставляя, на лицензионной основе, ядро ARM7 своим кремниевым партнерам фирма ARM разработала микроконтроллеры ARM7100, ARM7500 и ARM7500FE.
- Микроконтроллер ARM7100 можно назвать микроконтроллером широкого применения, т.к. он ориентирован на использование в таких устройствах как: персональные информационные устройства (PDA), мобильные телефоны, карманные измерительные устройства и системы сбора данных.
- Два других микроконтроллера ARM7500 и ARM7500FE являются однокристальными микрокомпьютерами, ориентированными реализацию мультимедиа устройств, портативных и настольных компьютеров, карманных вычислительных и измерительных устройств.
- Общим для всех трех микропроцессоров является использование ядра ARM7, встроенного единого кэш команд и данных емкостью 8 Кбайт (ARM7100) и 4 Кбайт (ARM7500 и ARM7500FE), MMU, буфера записи, наличие режимов энергосбережения.

Микроконтроллеры семейства ARM7 (2)

- Эти два микроконтроллера отличаются друг от друга наличием в приборе ARM7500FE ускорителя операций с плавающей точкой (FPA) и, соответственно, его более высокой производительностью. Они также реализованы по модульному принципу и объединяют ядро ARM7 с самодостаточными макроячейками видео, звука, FPA (ARM7500FE) и стандартных библиотечных ячеек периферии.
- Рассмотрим в качестве примера микроконтроллер ARM7100.

Основные характеристики ARM7100

- Производительность 18,4 MIPS при тактовой частоте 18,4 МГц и напряжении питания 3,3 В
- Потребление 66 мВт при напряжении питания 3,3 В
- Встроенный единый кэш команд и данных емкостью 8 Кбайт
- Интерфейс ROM и расширения (сегменты 8x256 Мбайт 8-, 16- и 32-разрядные)
- Контроллер DRAM с поддержкой быстрого страничного режима (8-, 16- и 32-разрядных)
- Адресуемое пространство в 3072 Мбайт
- Порты I/O (4x8 + 1x4)
- Телефонный CODEC интерфейс с FIFO на 16 байт
- Программируемый контроллер LCD (halfVGA - 640x240) с поддержкой DMA
- Полнодуплексный UART с двумя 16-разрядными FIFO и логикой протокола IrDA
- Синхронный последовательный интерфейс
- Два 16-разрядных таймера/счетчика и сторожевой таймер
- Часы реального времени с компаратором
- Два интерфейса с DC-DC преобразователями
- Корпус PQFP с 208 выводами

Проблема большого размера кода

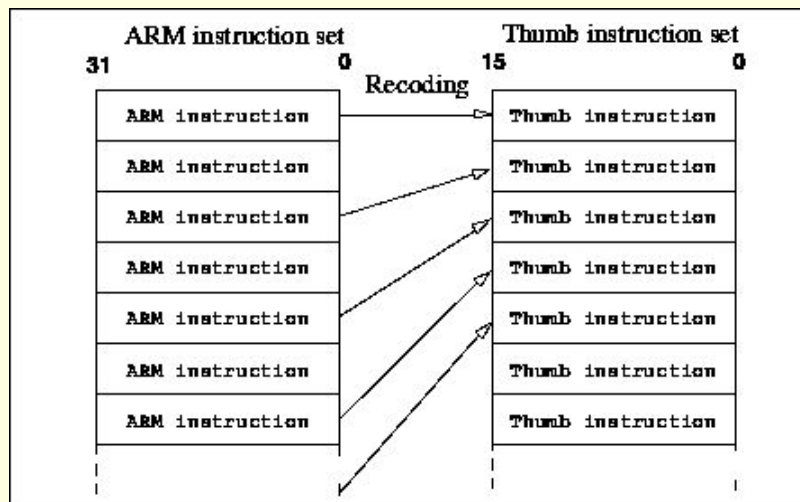
- Как видно из выше перечисленного, на основе ядер ARM7 реализуются достаточно мощные и сложные приборы, по своей производительности приближающиеся к производительности рабочих станций недавнего прошлого, что обеспечивается высокопроизводительным RISC ядром и мощной 32-разрядной ARM системой команд.
- Необходимо отметить, что программы, подготовленные даже для довольно эффективной 32-разрядной ARM системы команд, требуют памяти значительного объема, что в свою очередь приводит к росту общей стоимости системы.
- Специалисты фирмы ARM предложили решение этой проблемы, разработав и внедрив **технология Thumb**, позволяющую существенно сократить объем кодов, необходимых для реализации той же программы, что выполняется на 32-разрядной ARM системе команд. До настоящего времени эта технология считается лучшей из технологий, использующих сжатые системы команд.

Решение проблемы большого размера кода

- Существует несколько подходов, решающих проблему размера кода:
 - **Написание кода вручную на ассемблере.** Для получения минимального размера кода программист может писать коды вручную - на ассемблере. Однако, этот процесс может занять непрактично большое время и в результате может быть получен код, который трудно поддерживать и всего лишь на 10-20% более компактный чем код, полученный при использовании хорошего компилятора. Корень проблемы, неэффективность кода, все еще не решается.
 - **Использование улучшенного компилятора.** Технология компилирования может улучшить код, но опять таки меньшим размер кода будет при ручном кодировании на ассемблере.
 - **Использование компрессированного кода.** Одним из вариантов может быть использование некоторой формы сжатого кода, который разворачивается во время выполнения. Однако, быстрая декомпрессия, которая не будет снижать производительность процессора при выполнении этого кода, достаточно сложна и требует использования дополнительных ресурсов системы.

Система команд Thumb. Краткое описание

- Технология Thumb - дополнительное расширение к архитектуре ARM. Система команд Thumb содержит 36 команд, производных от стандартной 32-разрядной системы команд ARM, перекодированных в 16-разрядные коды. Такой подход обеспечил очень высокую плотность кода, поскольку команды Thumb составляют половину ширины формата команд ARM. В процессе выполнения эти новые 16-разрядные Thumb коды декомпрессируются процессором в соответствующие эквивалентные команды ARM, которые затем и выполняются ядром ARM обычным способом.

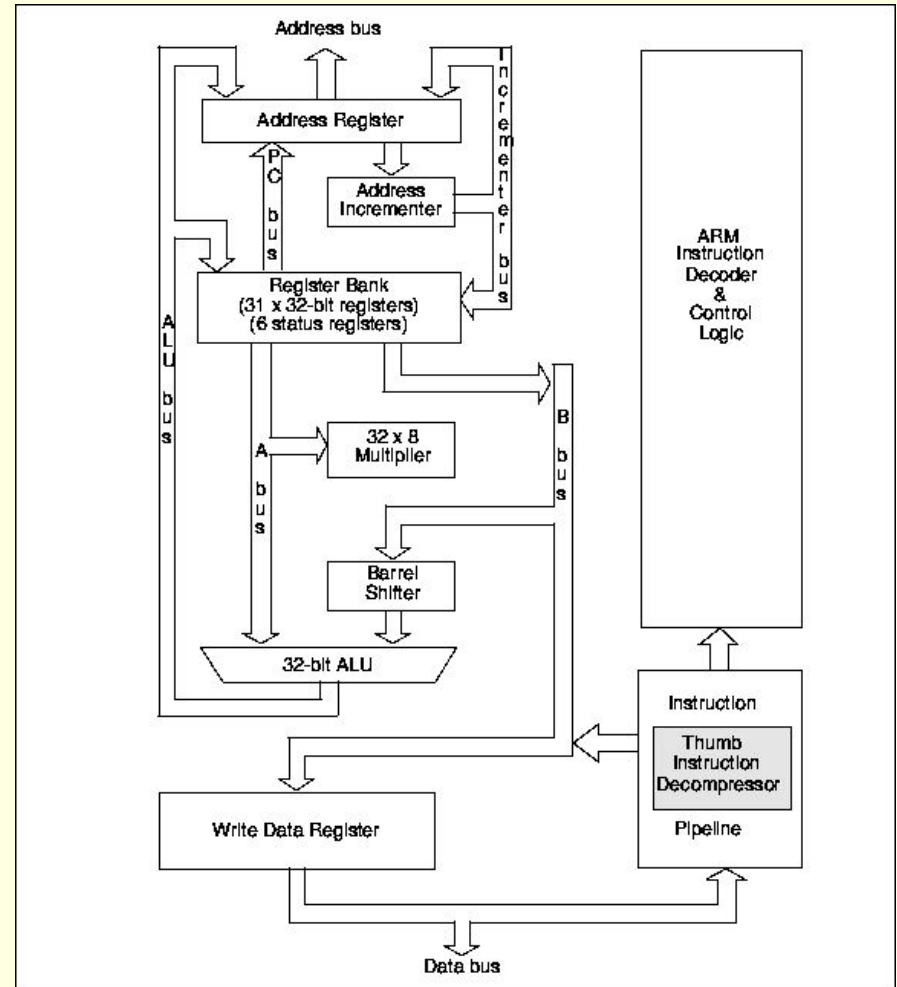


Аспекты концепции Thumb (1)

- Технология Thumb - это не только смешанная система команд. **Thumb-ориентированные ядра имеют две отдельные системы команд** – уникальное достоинство, позволяющее разработчику использовать всю мощность 32-разрядной системы команд ARM при использовании преимуществ малого размера кода системы команд Thumb. Тот факт, что две системы команд являются совершенно отдельными, говорит о том, что средства декодирующей логики также чрезвычайно просты, что в свою очередь, сохраняет малым размер кристалла и сохраняет лучшее в отрасли соотношение производительность/потребление.
- Так как **Thumb-ориентированные ядра способны выполнять и стандартную ARM систему команд и новые команды Thumb**, разработчик, при переходе от подпрограммы к подпрограмме, может находить **компромисс между размером кода и производительностью**, подготавливая критичные к размеру подпрограммы в коде Thumb и критичные к производительности подпрограммы в кодах ARM.
- Thumb-ориентированные ядра типа ARM7TDMI имеют полную 32-разрядную архитектуру ARM, так что разработчик сохраняет 32-разрядную производительность RISC архитектуры. Комбинация двух систем команд, выполняющихся на 32-разрядном Thumb -ориентированном ядре, обеспечивает эффективное решение проблемы больших размеров кода и проблемы невысокой производительности 16-разрядных систем.

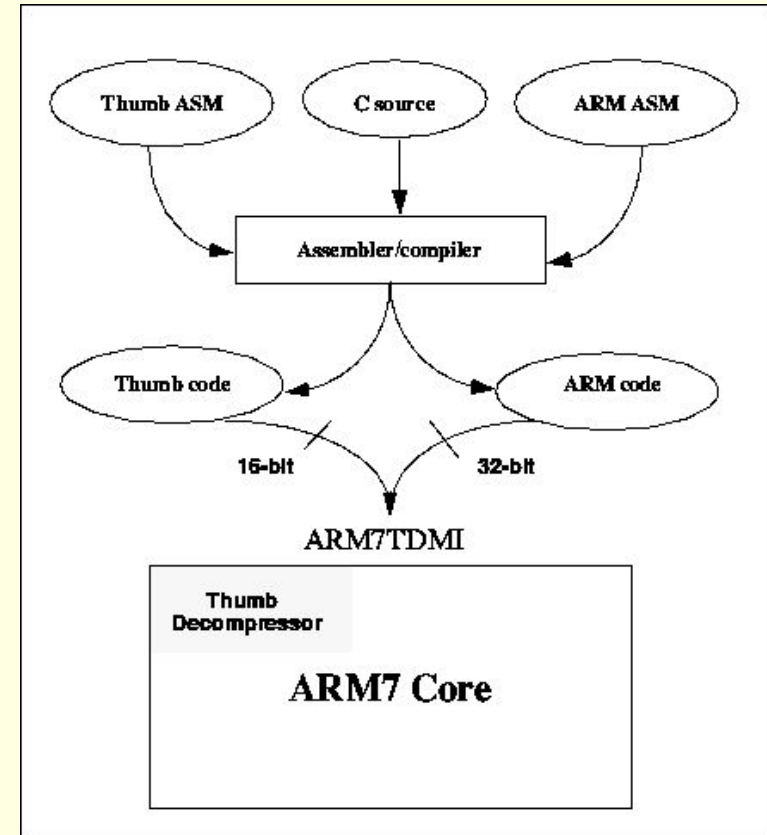
Аспекты концепции Thumb (2)

- Полученные к настоящему времени результаты показали **улучшение плотности кода на 30%, по сравнению с кодом ARM**, что позволяет считать Thumb - ориентированные процессоры лучшими по плотности кода в сравнении и с традиционными CISC процессорами.



Аспекты концепции Thumb (3)

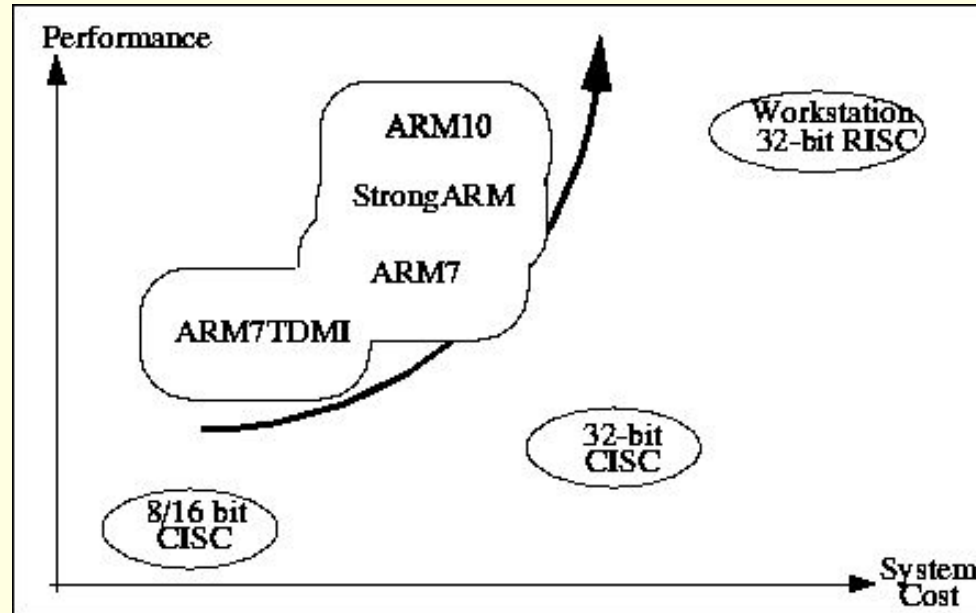
- Кроме введения новых Thumb команд, фирма ARM добавила к системам команд и ARM и Thumb поддержку формата полуслов (16-разрядных данных). Следовательно архитектура ARM теперь полностью поддерживает 8, 16 и 32-разрядные данные. Были добавлены и для Thumb и ARM ядер операции со знаками для поддержки ими 8 и 16-разрядных операций с данными со знаками.
- Комплект средств разработки программного обеспечения ARM также был расширен, чтобы обеспечить поддержку разработки кодов Thumb. Комплект позволяет программисту писать и размещать в памяти системы коды ARM, коды Thumb или оба вместе.



Последовательность этапов проектирования ПО для Thumb-ориентированного ядра

Производительность Thumb-ориентированных ядер

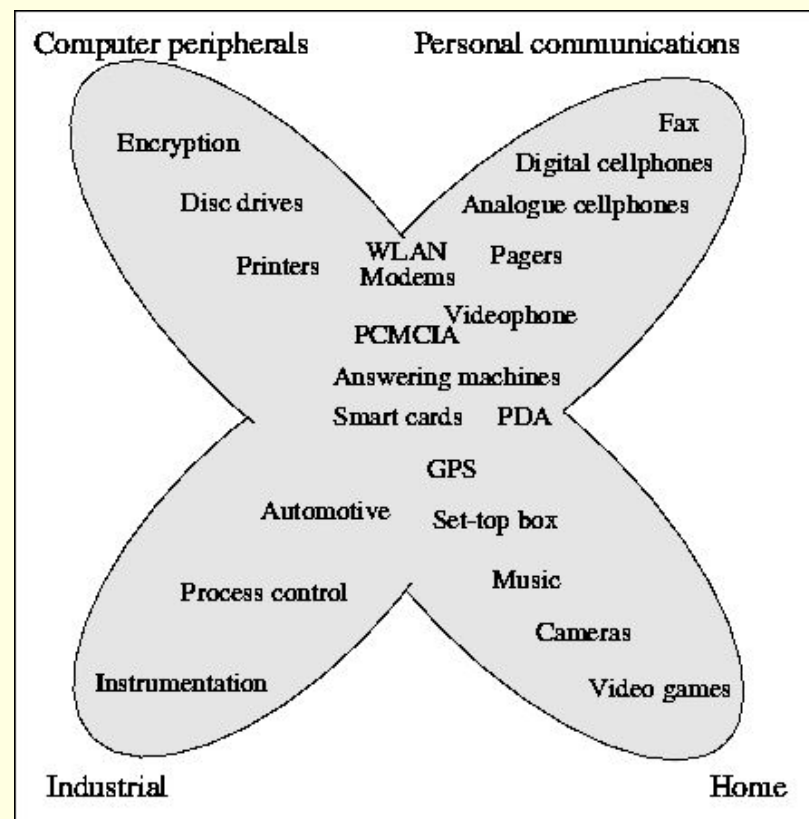
- Фирма ARM считает, что Thumb-ориентированные ядра будут особенно успешно использоваться в применениях с относительно ограниченными возможностями, т.е. в тех, в которых сегодня используют 8-разрядные и 16-разрядные контроллеры и которым необходима большая производительность без дополнительных затрат.



Thumb-ориентированные ядра и занимаемая ими область производительности

Совместимость исходного текста

- Так как Thumb-ориентированные ядра – просто расширение архитектуры ARM, разработчик может компилировать коды Thumb, коды ARM или смесь обоих. Эта совместимость исходного текста между Thumb-ориентированными ядрами и ядрами ARM, обеспечивает беспрепятственный путь к будущим обновлениям до 32-разрядных уже находящихся в эксплуатации систем, что делает Thumb-ориентированные ядра гарантированным вложением капитала в будущее.
- Кроме того, простота реализации технологии Thumb гарантирует, что перспективные ядра ARM с еще более высокой производительностью также будут располагать Thumb-ориентированными возможностями.



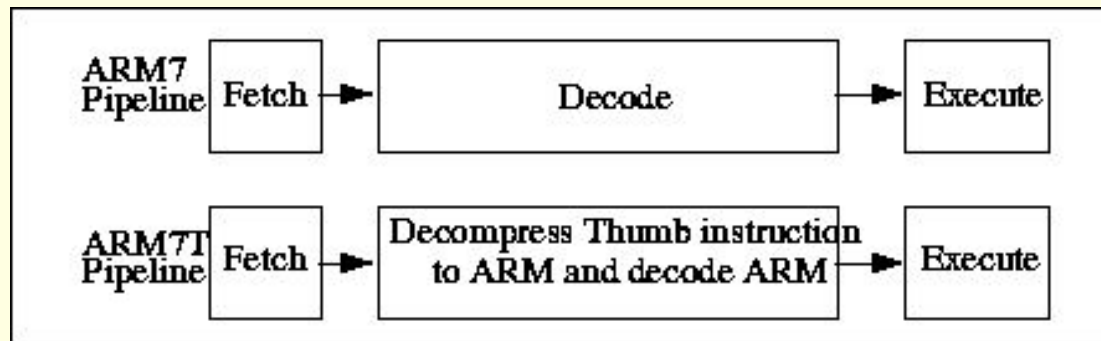
Прикладные области для Thumb-ориентированного ядра

ARM7TDMI. Краткое описание

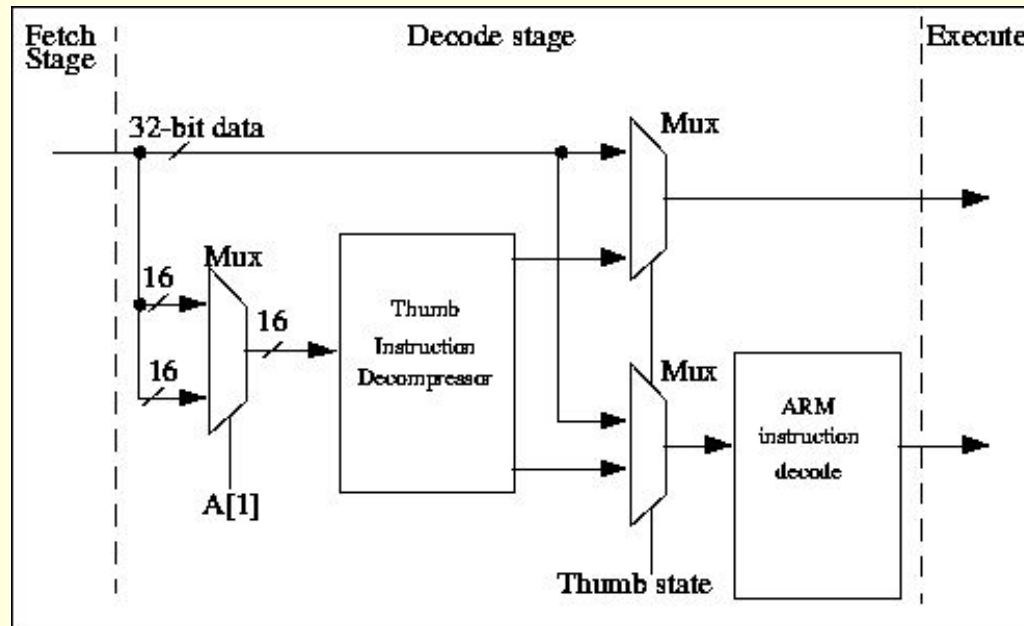
- Первым Thumb-ориентированным ядром стало ядро ARM7TDMI. Это ядро семейства ARM7 располагающее:
 - Встроенной макроячейкой EmbeddedICE™, поддерживающей отладку встроенного ядра;
 - 32-разрядным аппаратным умножителем;
 - Декомпрессором Thumb;
 - 32-разрядной производительностью в 8- и 16-разрядных управляющих применениях.
- Ядро ARM7TDMI пополнило стандартный ряд 32-разрядных ядер ARM, обеспечив возможность выхода на рынок встраиваемого управления, принося 32-разрядную производительность в 8 и 16-разрядные применения управления. Первый Thumb-ориентированный прибор в кремнии был выпущен во второй половине 1995.
- Ядро ARM7TDMI используется как лицензионная макроячейка ASIC ARM, предназначенная использования при создании стандартных приборов специального назначения.

Декодирование и выполнение в одном цикле (1)

- И ARM7 и ARM7T ядра в одном тактовом цикле, используют 3-уровневый конвейер с фазами выборки, декодирования и выполнения. Поток команд через каждый уровень конвейера управляется высокими и низкими фазами тактового сигнала. **В ядре ARM7TDMI неиспользуемая фаза тактового сигнала используется для декомпрессирования команд Thumb в каскаде декодирования.** Следовательно, в одном тактовом цикле производится и декодирование и выполнение команды, **не требуется никаких дополнительных непроизводительных затрат синхронизации.**



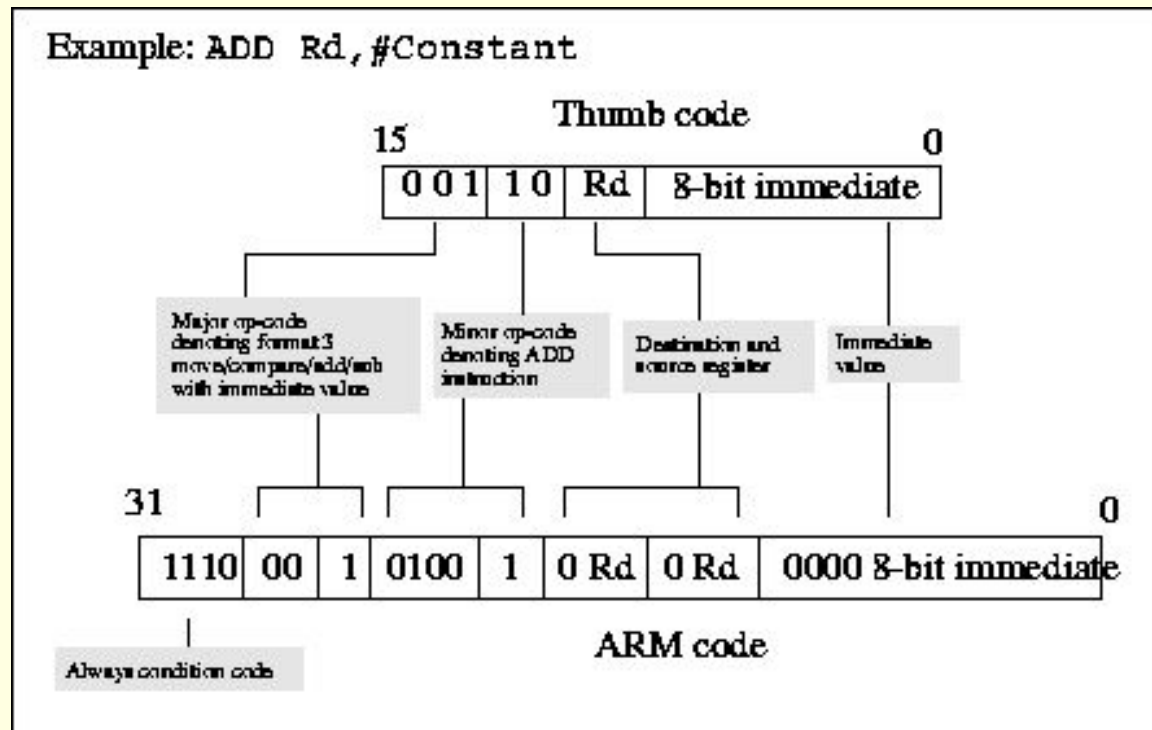
Декодирование и выполнение в одном цикле (2)



- Команды ARM, поступающие из каскада выборки конвейера, проходят через декодер ARM и активируют сигналы управления старшими и младшими битами операционного кода. Старшие биты операционного кода описывают тип выполняемой команды, тогда как младшие биты определяют детали команды типа определения регистров или операнда.

Трансляция команд Thumb в команды ARM (1)

- В Thumb состоянии мультиплексоры направляют Thumb команды через логику декомпрессии Thumb, разворачивающую команду Thumb в ее эквивалент ARM команды. Затем команда ARM выполняется в нормальном режиме. Это легко понять на примере:



Трансляция команды ADD Thumb в команду ADD ARM.

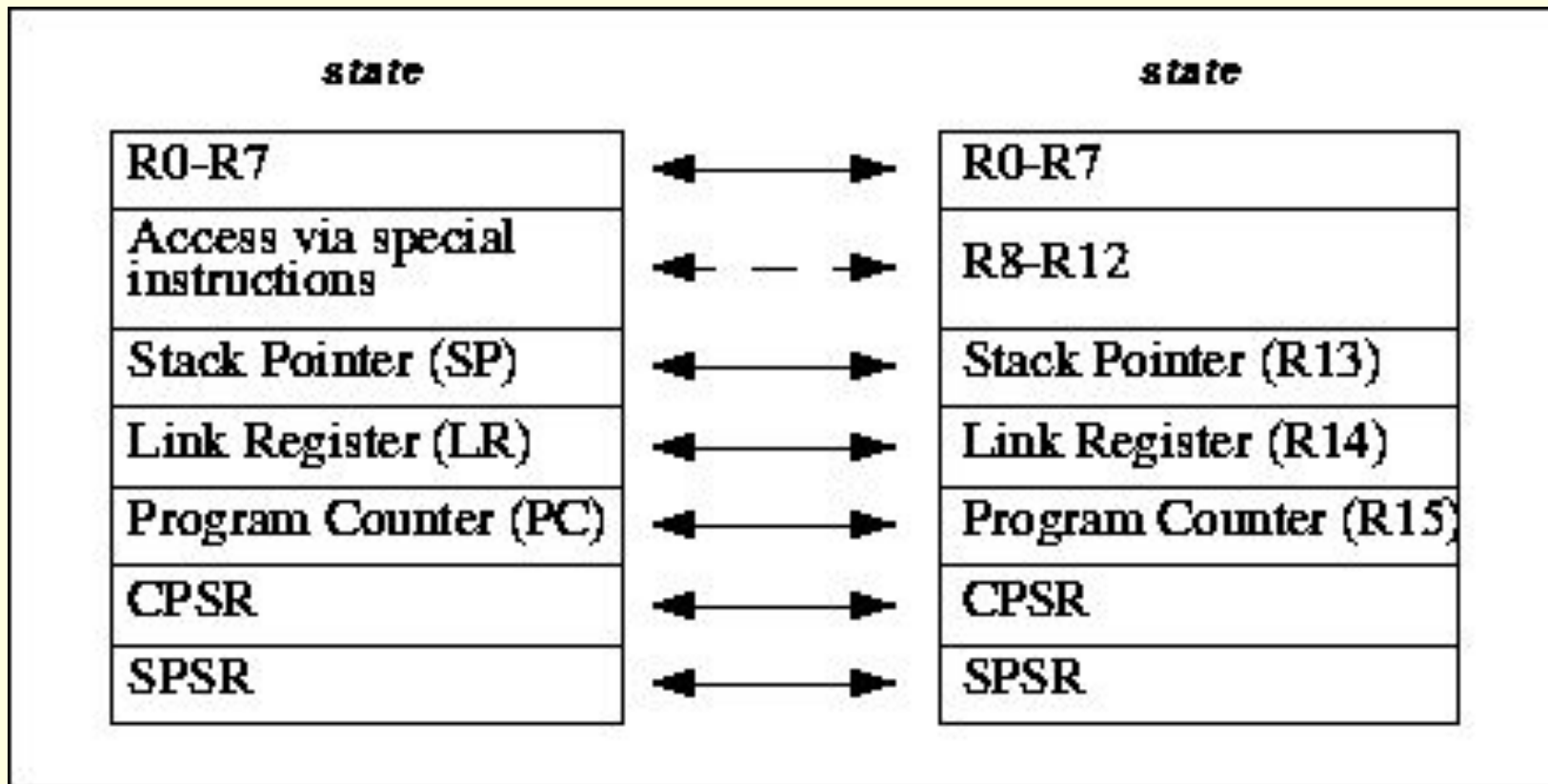
Трансляция команд Thumb в команды ARM (2)

- Как видно из рисунка, старший операционный код команды Thumb помещается в команду ARM, а младший транслируется через справочную таблицу.
- В команде ARM всегда присутствует код условия, определяемый старшим операционным кодом.
- Старший операционный код определяет маршрут операнда от операционного кода Thumb до операционного кода ARM. Определители регистров расширяются дополнительным нулем от операционного кода Thumb (3 бита) до 4 битов, поскольку эта команда Thumb обращается только к регистрам ARM R0-R7. Значение константы также расширяется нулями, определяя 8-разрядную константу в операционном коде ARM.
- Это изящное решение применимо на любом ядре ARM и будет использоваться в перспективных ARM архитектурах.

Система команд Thumb

- Для системы команд Thumb выбраны те команды ARM, которые, в среднем, не обеспечивают преимуществ полноформатных стандартных 32-разрядных операционных кодов ARM, или те, в отношении которых опыт показал, что они используются наиболее часто и, следовательно, являются наиболее важными, или те которые необходимы компилятору, для того чтобы обеспечить наилучшую возможную плотность кода.
- Процесс формирования команд Thumb заключался в сокращении полей всех наиболее часто используемых команд ARM до тех пор, пока они не стали 16-разрядными. Полученная в результате система команд затем обрабатывалась последовательностью итераций, обеспечивавшей наибольшее количество битов опкодам наиболее употребительных команд (например, вызовам подпрограмм, которые составляют 1/16 системы команд) и удаление битов из менее критических команд до тех пор, пока не было получено наилучшее соотношение производительность/плотность кода. Система команд была ориентирована и на перспективу, в операционных кодах оставлено пространство для новых команд, которые будут вводиться разработчиками программного обеспечения и которые должны будут поддерживаться новыми версиями компилятора.

- При создании 16-разрядных операционных кодов, в систему команд Thumb были введены некоторые ограничения . Наиболее очевидные: уменьшенное число регистров, доступных при выполнении кода Thumb. Вместо пятнадцати 32-разрядных регистров общего назначения (**GPR**) плюс **PC** процессора ARM, программист имеет доступ к восьми **GPR** регистрам, указателю стека, регистру связей (**Link Register**) и **PC**. В системе команд Thumb, восемь **GPR** регистров (R0-R7) называются **Lo набором регистров**. Другие регистры ARM (R8-R15) обозначаются как **Hi регистры**. Программист в Thumb режиме ограничен доступом к **Hi регистрам** через команды перемещения, сравнения и команду **ADD** , предоставляющие ему некоторую локальную быструю временную среду хранения. Переключения между ARM и Thumb кодами не влияют на содержимое **GPR** регистров.



Отображение Thumb регистров состояния на регистры состояния ARM

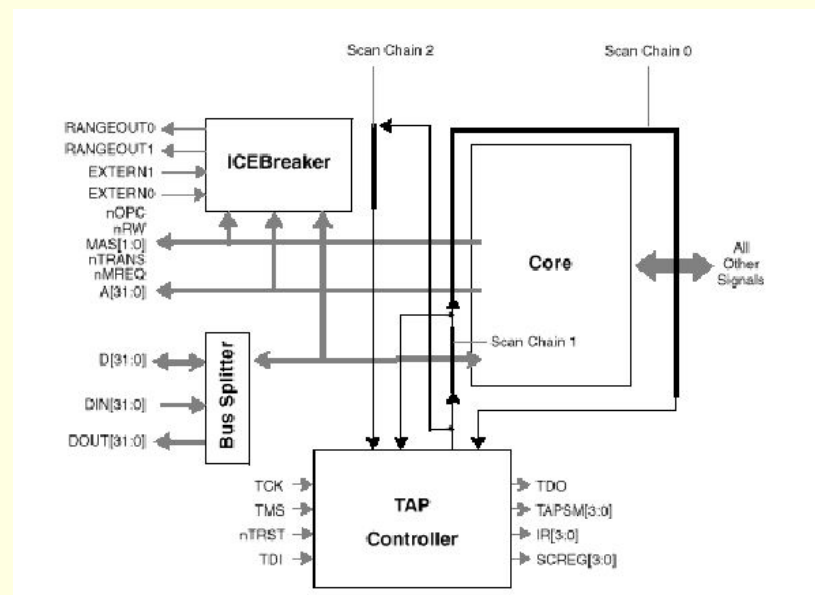
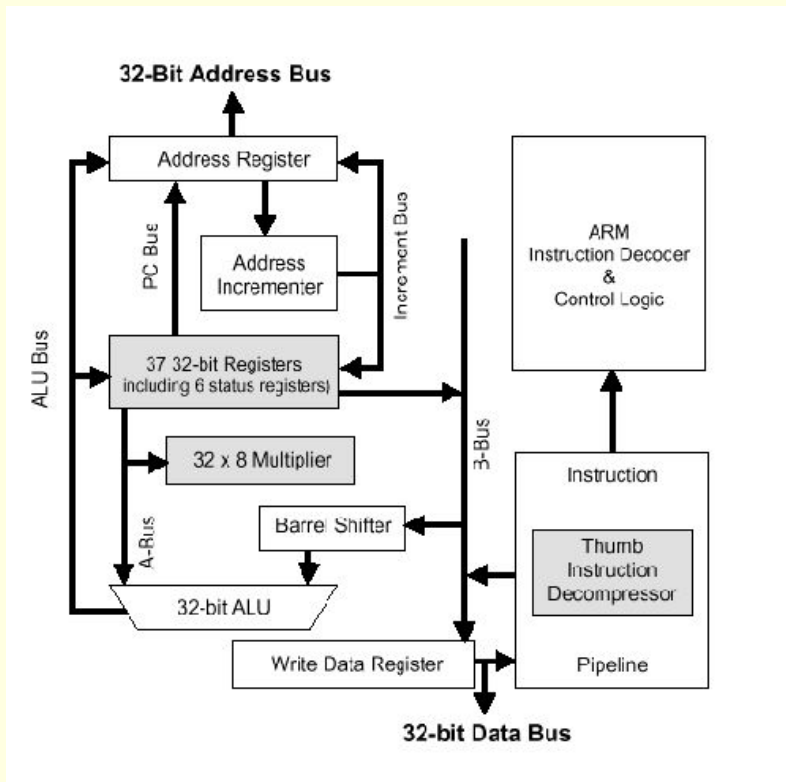
Интерфейс памяти ARM7TDMI

- При разработке интерфейса памяти разработчики ориентировались на обеспечение реализации потенциально возможной производительности без повышения стоимости самой памяти. Критичные к быстродействию сигналы управления также конвейеризованы с тем, чтобы реализовать функции управления системой памяти в стандартной логике с малым потреблением, и эти сигналы управления упрощают использование быстрых локальных режимов обращения, необходимых для стандартных динамических RAM. Кроме того, интерфейс памяти идеально подходит для организации взаимодействия как со встроенной на кристалл памятью, так и с внешней памятью, с блоками Flash памяти, что позволяет реализовать внутрисистемное программирование, защиту памяти, снизить время выхода на рынок, сократить общую стоимость системы.

Интерфейс памяти ARM7TDMI

- Интерфейс с памятью у процессора ARM7TDMI организуется следующими основными элементами:
 - 32-разрядной шиной адреса, определяющей адрес ячейки памяти, которую необходимо использовать.
 - 32-разрядной двунаправленной шиной передачи данных $D[31:0]$, плюс двумя отдельными однонаправленными шинами данных $DIN[31:0]$ и $DOUT[31:0]$, через которые перемещаются команды и данные. Данные могут иметь формат слова, полуслова или байта.
 - Сигналами управления, определяющими, например, формат перемещаемых данных и направление их передачи и, кроме того, уровень приоритета (этот набор сигналов позволяет ARM7TDMI достаточно просто организовывать интерфейс с DRAM, SRAM и ROM).

Блок-схема ядра ARM7TDMI

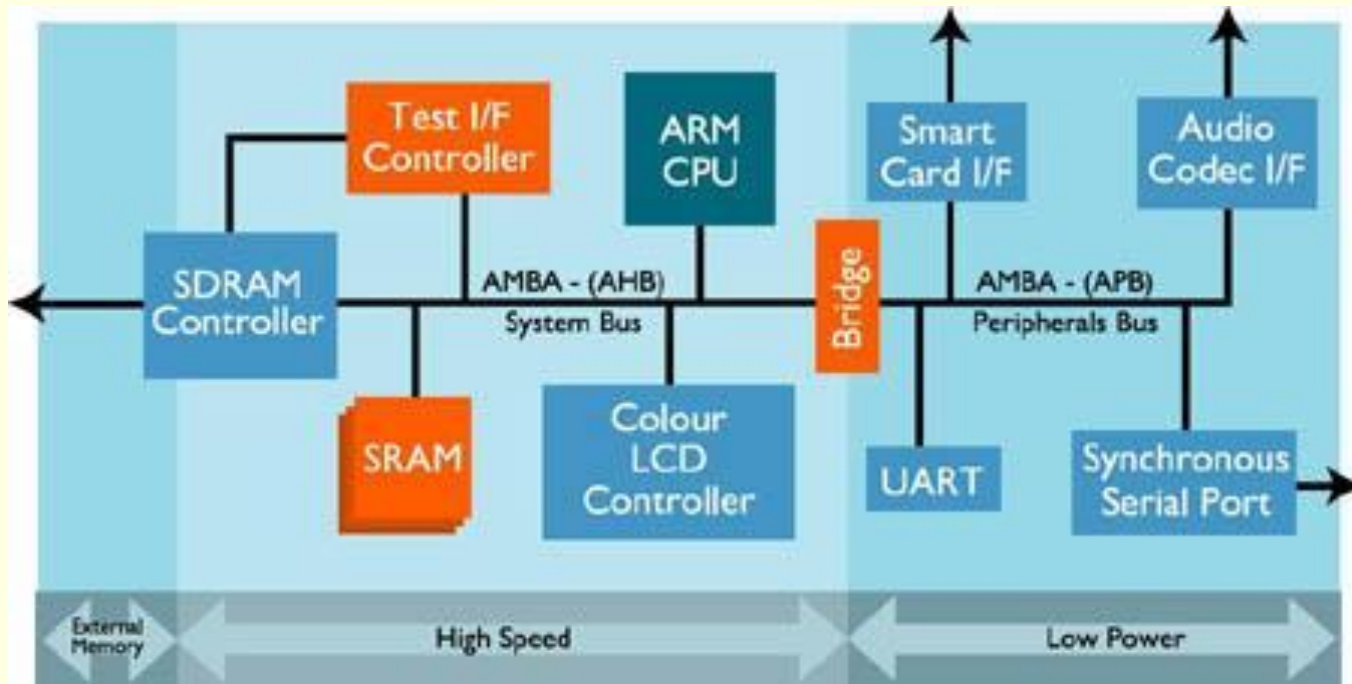


Архитектура ARM7TDMI

- Процессор ARM7TDMI - 32-разрядный RISC процессор с 3-уровневым конвейером, сформированный вокруг банка из 37 32-разрядных регистров, в который входят 6 регистров состояния. Процессор оснащен встроенным умножителем 32x8 и 32-разрядным многорегистровым циклическим устройством сдвига. Пять независимых встроенных шин (PC шина, шина инкремента, шина ALU и A- и B-шины) обеспечивают, при выполнении команд высокую степень параллелизма.
- Как видно из сравнения блок-схем процессоров ARM7 и ARM7TDMI (См. Рис. 1) в блоке конвейера процессора добавился декомпрессор команд Thumb.
- Как видно из Рис. 2, ядро ARM7TDMI состоит из ядра собственно процессора, показанного на Рис.1, и расширений отладки: контроллера сканирования TAP (boundary scan) и внутрисхемного эмулятора (ICEBreaker).

Шина AMBA (1)

- Шина AMBA - стандартная встроенная ASIC шина, разработанная фирмой ARM для организации эффективного взаимодействия компонентов приборов, построенных на базе ядер фирмы и обеспечивающая быстрое модульное проектирование систем при упрощении многократного использования схемотехники и тестов.



Блок-схема прибора класса «система-на-кристалле» с шиной AMBA

Шина AMBA (2)

- Типовая шина AMBA содержит системную шину (AHB или ASB) и шину периферии (APB).
- Системная шина соединяет встраиваемые процессоры, такие как ARM ядра, с высокопроизводительной периферией, контроллерами DMA, встроенными памятью и интерфейсами. Это высокоскоростная, с широкой полосой пропускания шина, поддерживающая, для обеспечения максимальной производительности, управление с большим количеством ведущих устройств.
- Шина периферии работает с упрощенным протоколом и разработана для организации интерфейса с периферийными устройствами общего назначения или дополнительными периферийными устройствами. С системной шиной она соединяется через мост (bridge), способствующий снижению потребления системы.

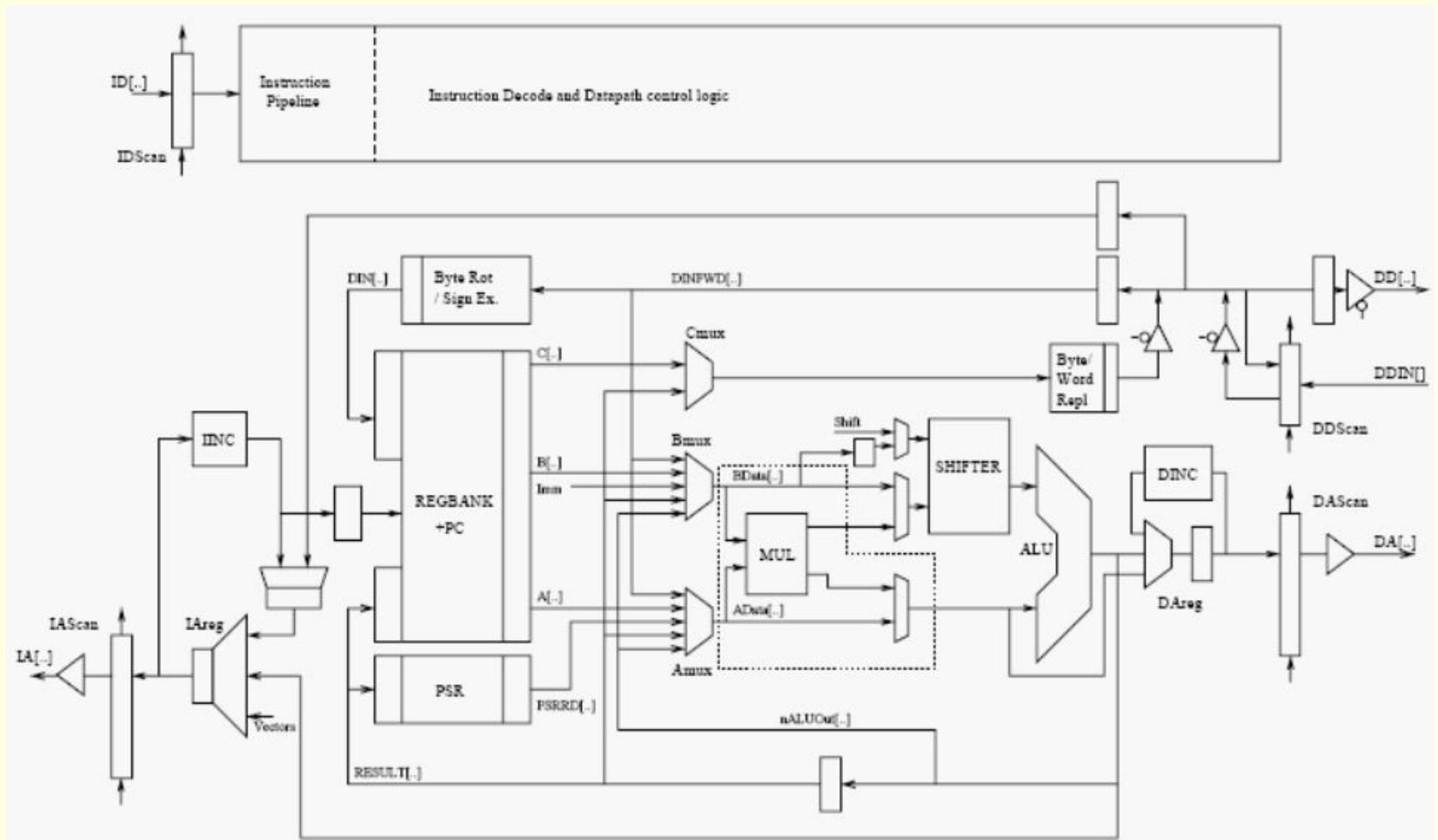
Семейство ARM9 Thumb

- Первые приборы семейства ARM9 Thumb появились в октябре 1997 года.
- Семейство 32-разрядных ARM RISC процессоров ARM9TDMI, сохранившее основные преимущества ARM7TDMI, ставших промышленным стандартом, обеспечивает двукратное увеличение производительности, при изготовлении по эквивалентной технологии.
- Производительность ARM9TDMI составляет 133 MIPS при 120 МГц и технологии CMOS с топологическими нормами 0,35 мкм. При топологии 0,25 мкм и 0,18 мкм рабочая частота составит свыше 200 МГц и производительность свыше 220 MIPS.
- В семейство ARM9TDMI, в настоящее время, входит процессорное ядро ARM9TDMI, и оснащенные кэш процессорные макроядра ARM940T и ARM920T.

Возможности процессоров семейства ARM9TDMI

- Общие для, всех процессоров семейства ARM9TDMI возможности:
 - Высокопроизводительный 32-разрядный ARM RISC механизм
 - Гарвардская архитектура с отдельными шинами команд и данных
 - Пятиуровневый конвейер
 - Модуль, выполняющий операцию перемножения/аккумуляции 16x32 за один цикл
 - Гибкая синхронизация CPU и шины, включая асинхронную, синхронную и одноктактовую конфигурации
 - Thumb 16-разрядная система команд, обеспечивающая лучшую в отрасли, плотность кода
 - Встроенные возможности EmbeddedICE JTAG отладки программного обеспечения
 - Возможность адаптации к перспективным CMOS технологиям с меньшими топологическими нормами
 - Совместимость с низковольтными CMOS технологиями
 - 100% совместимость двоичных кодов пользователя с ARM7TDMI
 - Возможность интеграции класса "система-на-кристалле" со встроенным тестированием в процессе производства

Блок-схема процессорного ядра ARM9TDMI

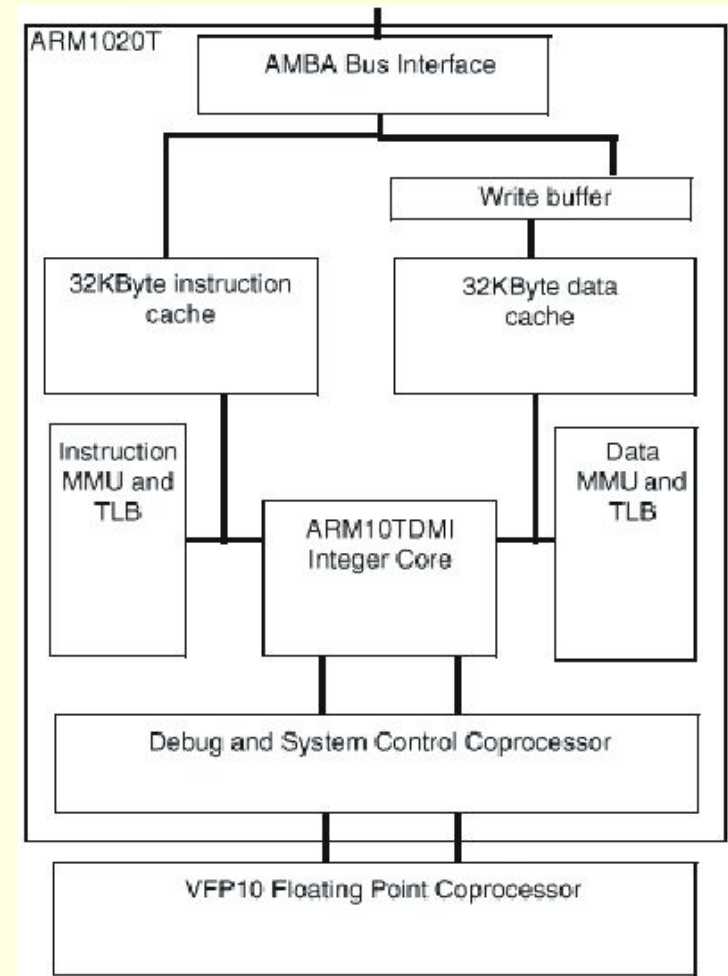


Семейство ARM10

- Всего через год после обнародования сведений о разработке процессоров семейства ARM9 Thumb были обнародованы (октябрь 1998 года) сведения о разработке процессоров нового еще более производительного семейства - семейства ARM10 Thumb, обеспечивающего производительность свыше 400 MIPS. Однако главной, пожалуй, особенностью нового семейства нужно считать наличие векторного сопроцессора вычислений с плавающей точкой, явного свидетельства современной тенденции к объединению в одном приборе управляющих целочисленных процессоров и процессоров цифровой обработки сигналов.
- В основе процессоров семейства ARM10 Thumb целочисленное ядро ARM10TDMI(TM), использующее ARM® 32-разрядную RISC систему команд, сжатую 16-разрядную систему команд Thumb и расширенные Multi-ICE средства отладки программного обеспечения. Ядро процессора ARM10TDMI - первая реализация ARMV5T Архитектуры Системы команд (*Instruction Set Architecture - ISA*).
- Ядро ARM10TDMI оснащено пятиуровневым конвейером, реализует параллельное выполнение команд, предсказание переходов и способно продолжать работу при неудачном обращении к кэш, обеспечивая высокую производительность в реальных применениях.

Семейство ARM10 (2)

- В настоящее время в семейство входит ARM1020T™ - кэшированное макроядро процессора, сформированное на основе ядра ARM10TDMI и оснащенное встроенными кэш команд и данных емкостью по 32 Кбайт, MMU, поддерживающим виртуальную память с подкачкой страниц по требованию, буфером записи, и широкополосным шинным интерфейсом AMBA, класса "система на кристалле".

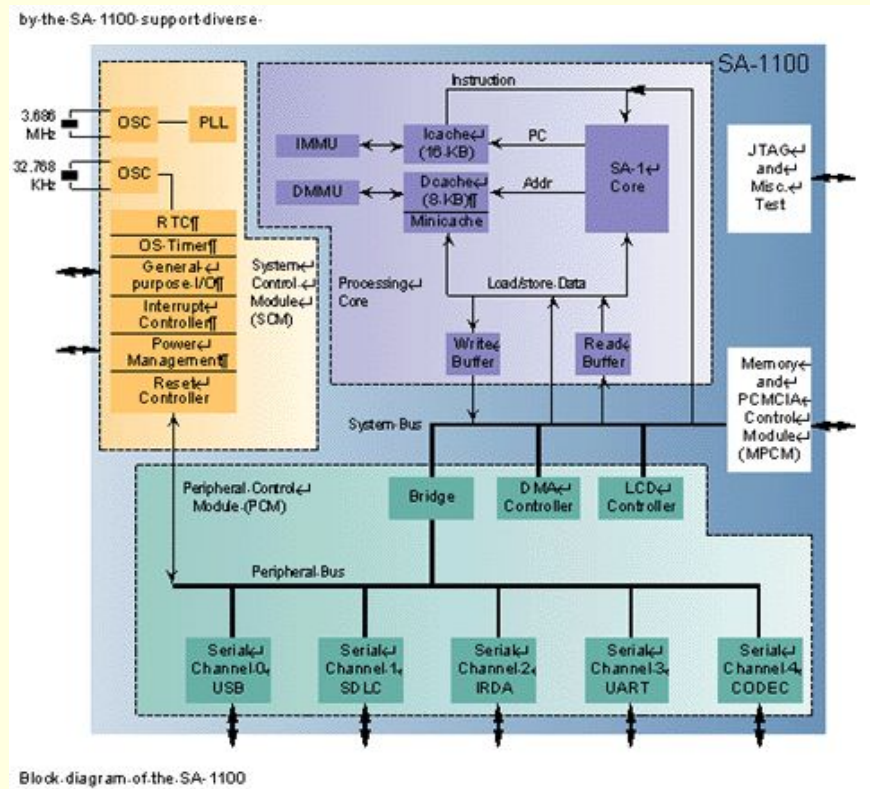


Семейство StrongARM (1)

- Первым членом семейства StrongARM, порожденного соглашением между фирмами ARM и Digital Semiconductor, стал ориентированный на встраиваемые применения микропроцессор SA-110 (StrongARM-110), выпущенный еще в сентябре 1996 года.
- Программируемый системный интерфейс микропроцессора SA-110 предоставляет пользователям высокую гибкость при разработке устройств, соответствующих специфическими требованиям применений. Интерфейс SA-110 может работать в синхронном или асинхронном режимах и программируется в диапазоне частот. Такая гибкость позволяет разработчикам организовывать интерфейс с различными типами памяти, ориентируясь или на обеспечение высокой производительности или на получение системы с невысокой стоимостью.
- Всего через год после появления микропроцессора SA-110 фирма Digital Equipment Corporation расширила семейство StrongARM, выпустив микропроцессор SA-1100. Ориентированный на дальнейшее расширение возможностей и технических характеристик карманных PC, суб-ноутбуков, "интеллектуальных" сотовых телефонов, Internet-совместимых "Web" телефонов и высокопроизводительных встраиваемых применений, микропроцессор StrongARM SA-1100, обеспечил в три-пять раз большую производительность, чем конкурирующие приборы - уже находившиеся в то время в производстве или только разрабатывавшиеся.

Семейство StrongARM (2)

- Микропроцессор SA-1100, с максимальной тактовой частотой 220 МГц - первый встраиваемый процессор, приближающийся к соотношению 1000 MIPS/Вт. При тактовой частоте 220 МГц и напряжении питания 2,0 В микропроцессор SA-1100, ориентированный на встраиваемые применения, обеспечивает производительность 250 Dhrystone V2.1 MIPS, потребляя менее 550 мВт, ориентированная на мобильные применения версия с тактовой частотой 133 МГц и напряжением питания 1,5 В обеспечивает производительность 150 Dhrystone V2.1 MIPS, потребляя менее 230 мВт.



Семейство StrongARM (3)

- Следующим пополнением семейства StrongARM, выпущенным в середине 1999 уже под маркой фирмы Intel года, стал микропроцессор SA-1110.
- Этот микропроцессор, ориентированный на использование в мобильной аппаратуре с батарейным питанием, во многом повторил возможности и характеристики микропроцессора SA-1100, но он располагает расширенными возможностями взаимодействия с памятью, расширенными возможностями I/O и конфигурирования.