

Архитектура процессоров ARM

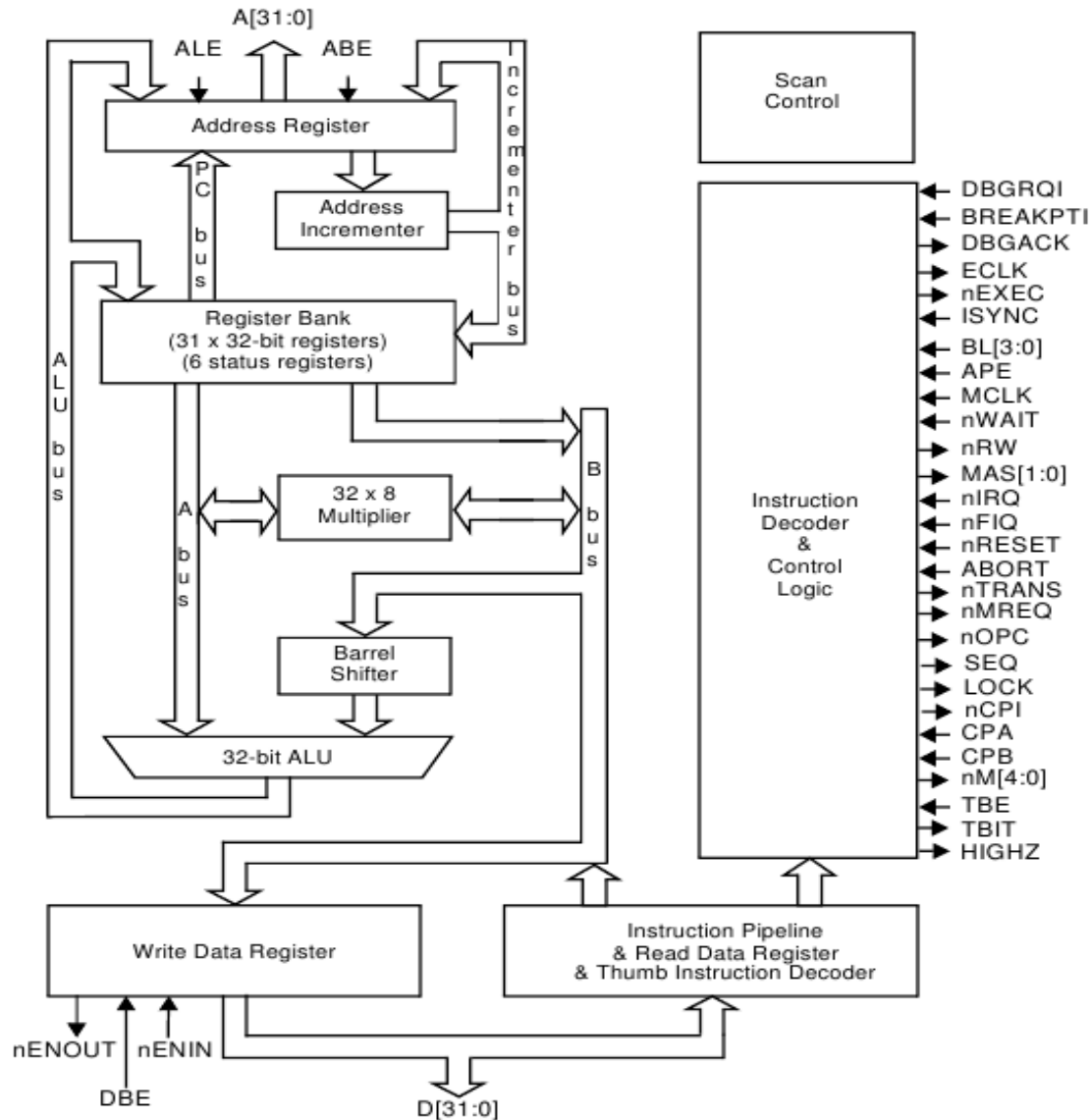
Архитектура ARM (ранее Advanced RISC Machine — усовершенствованная RISC-машина, предшественник — Acorn RISC Machine) — 32-битная архитектура набора команд с сокращённым набором команд, разрабатываемая ARM Limited.

Семейство	Название ядра/процессора	Размер кэш памяти кода/данных, байт	Блок управления памятью (MMU, MPU)	Thumb®	DSP	Jazelle™
Cortex	ARM Cortex-A8	переменный	MMU	есть	есть	есть
ARM10E™	ARM1020E	32k/32k	MMU	есть	есть	нет
	ARM1022E	16k/16k	MMU	есть	есть	нет
	ARM1026EJ-S	переменный	MMU или MPU	есть	есть	есть
ARM11™	ARM11MPCore	переменный	MMU	есть	есть	есть
	ARM1136J(F)-S	переменный	MMU	есть	есть	есть
	ARM1176JZ(F)-S	переменный	MMU	есть	есть	есть
ARM7™	ARM720T	8k	MMU	есть	нет	нет
ARM9™	ARM920T	16k/16k	MMU	есть	нет	нет
	ARM922T	32k/8k	MMU	есть	нет	нет
ARM9E™	ARM926EJ-S	переменный	MMU	есть	есть	есть

Особенности архитектуры ARMv4T

- Нет поддержки нелинейного (не выровненного по словам) доступа к памяти
 - Равномерный 16x32-битный файл регистра
 - Фиксированная длина команд 32 бита снижения плотности кода. Все арифметические команды — условные.
 - Режим Thumb (16 бит) — повышенная плотность кода.
 - Наличие двух систем команд:
 - ARM** - 32-битная система команд (полный доступ к регистрам).
 - THUMB** - 16-битная система команд (экономия памяти).
 - Одноцикловое исполнение.
 - Отдельные банки регистров для обработчиков прерываний.
-

Ядро ARMv4T



Ядро — не выпускается в виде отдельной микросхемы, а является составной частью различных микросхем.

Режимы работы процессора

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 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.

Теневые регистры — регистры, замещаемые на дополнительные.

Регистры

System & User	FIQ	Supervisor	Abort	IRQ	Undefined
R0	R0	R0	R0	R0	R0
R1	R1	R1	R1	R1	R1
R2	R2	R2	R2	R2	R2
R3	R3	R3	R3	R3	R3
R4	R4	R4	R4	R4	R4
R5	R5	R5	R5	R5	R5
R6	R6	R6	R6	R6	R6
R7	R7	R7	R7	R7	R7
R8	 R8_fiq	R8	R8	R8	R8
R9	 R9_fiq	R9	R9	R9	R9
R10	 R10_fiq	R10	R10	R10	R10
R11	 R11_fiq	R11	R11	R11	R11
R12	 R12_fiq	R12	R12	R12	R12
R13	 R13_fiq	 R13_svc	 R13_abt	 R13_irq	 R13_und
R14	 R14_fiq	 R14_svc	 R14_abt	 R14_irq	 R14_und
R15 (PC)	R15 (PC)	R15 (PC)	R15 (PC)	R15 (PC)	R15 (PC)

ARM State Program Status Registers

CPSR	CPSR  SPSR_fiq	CPSR  SPSR_svc	CPSR  SPSR_abt	CPSR  SPSR_irq	CPSR  SPSR_und
------	--	--	--	--	--

 — теневой (shadow, banked) регистр. R13 = SP R14 = LR

Особенности системы команд.

Предикация

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 режиме **ARM** — все команды содержат предикат.

В режиме **THUMB** — предикат только у команд условного перехода.

<i>while</i> (i != j)	loop: <i>CMP</i> Ri, Rj ; Проверка условия if (i != j),
{	; Признак GT, если i>j
<i>if</i> (i > j)	; Признак LT, если i<j
i -= j;	<i>SUBGT</i> Ri, Ri, Rj ; if "GT" (greater than), i = i-j;
<i>else</i>	
j -= i;	<i>SUBLT</i> Rj, Rj, Ri ; if "LT" (less than), j = j-i;
}	
	<i>BNE</i> loop ; if "NE" (not equal), then loop

Особенности системы команд.

Команды сдвига

Команды сдвига и арифметических действий — выполняются параллельно.

ARM:

ADD Ra, Rb, Rj, LSL #2

$a = b + (j \ll 2);$

THUMB:

ADD Rb, Rj, LSL #2

ADD Ra, Rb

Режим *ARM* — трёхоперандные команды.

Режим *THUMB* — двухоперандные команды.

Операционные системы для ARM-процессоров

Linux, BSD, ReactOS,

Plan 9, Inferno, OpenSolaris,

FreeRTOS, Nucleus, Symbian OS,

Windows CE, Android, QNX,

Apple iOS, RISC OS, eCos

Стандарт POSIX

POSIX® (**P**ortable **O**perating **S**ystem **I**nterface for **U**ni**X** —
Переносимый интерфейс операционных систем Unix)

Задачи, которые должен разрешить стандарт POSIX

- содействовать облегчению переноса кода прикладных программ на иные платформы;
 - способствовать определению и унификации интерфейсов заранее при проектировании, а не в процессе их реализации;
 - сохранить по возможности и учитывать все главные, созданные ранее и используемые прикладные программы;
 - определять необходимый минимум интерфейсов прикладных программ, для ускорения создания, одобрения и утверждения документов;
 - развивать стандарты в направлении обеспечения коммуникационных сетей, распределенной обработки данных и защиты информации;
 - рекомендовать ограничивать использование бинарного (объектного) кода для приложений в простых системах.
-