

Universidade Tecnológica Federal do Paraná (UTFPR)
Departamento Acadêmico de Eletrônica (DAELN)

SISTEMAS EMBARCADOS

Arquitetura ARM Cortex-M3

Prof. André Schneider de Oliveira

andreoliveira@utfpr.edu.br

Paradigma CISC

- Complex Instruction Set Computer
 - Conjunto de instruções inicialmente simples
 - Avanços tecnológicos permitiram a fabricação de computadores com mais transistores e menor custo
 - Projetistas optaram por conjuntos de instruções cada vez mais complexos
 - Intenção: reduzir a distância semântica entre Assembly e linguagens de alto nível

Paradigma CISC

- Instruções com elevado grau semântico
- Elevado número de modos de endereçamento (ex: endereçamento indireto em memória)
- Elevado número de ciclos de clock por instrução
→ redução da frequência de clock
- Menor número de instruções por programa
→ menor uso de memória de código
- Decodificação através de microcódigo
→ dificulta/impossibilita o uso de pipeline

Paradigma RISC

- Reduced Instruction Set Computer
 - Instruções simples que executam rápido
 - Elevado número de registradores de uso geral
 - Decodificação de instruções com lógica combinacional (tabela)
 - Execução utilizando pipeline
 - um ciclo de clock por instrução

Paradigma RISC

- Regularidade de tempo de execução
- Regularidade de tamanho de instrução
- Redução da área de silício e tempo de projeto
- Efeito final: melhor desempenho, apesar do número de instruções por programa ser maior

Resumo: RISC x CISC

RISC

Conjunto de instruções reduzido

Instruções semanticamente simples

Instruções de tamanho fixo

Decodificação simplificada
(tabela)

Execução regular

Instruções requerem o mesmo
número de ciclos de clock para
executar

Possibilita o uso de pipeline

CISC

Conjunto de instruções extenso

Instruções semanticamente
complexas

Instruções de tamanho variável

Decodificação complexa
(microcódigo)

Cada instrução executa à sua
maneira

Grande variação no número de
ciclos de clock por instrução

Extremamente difícil/impossível o
uso de pipeline

Pipeline (3 Estágios)

1. Busca (Fetch)

- Busca da instrução na memória

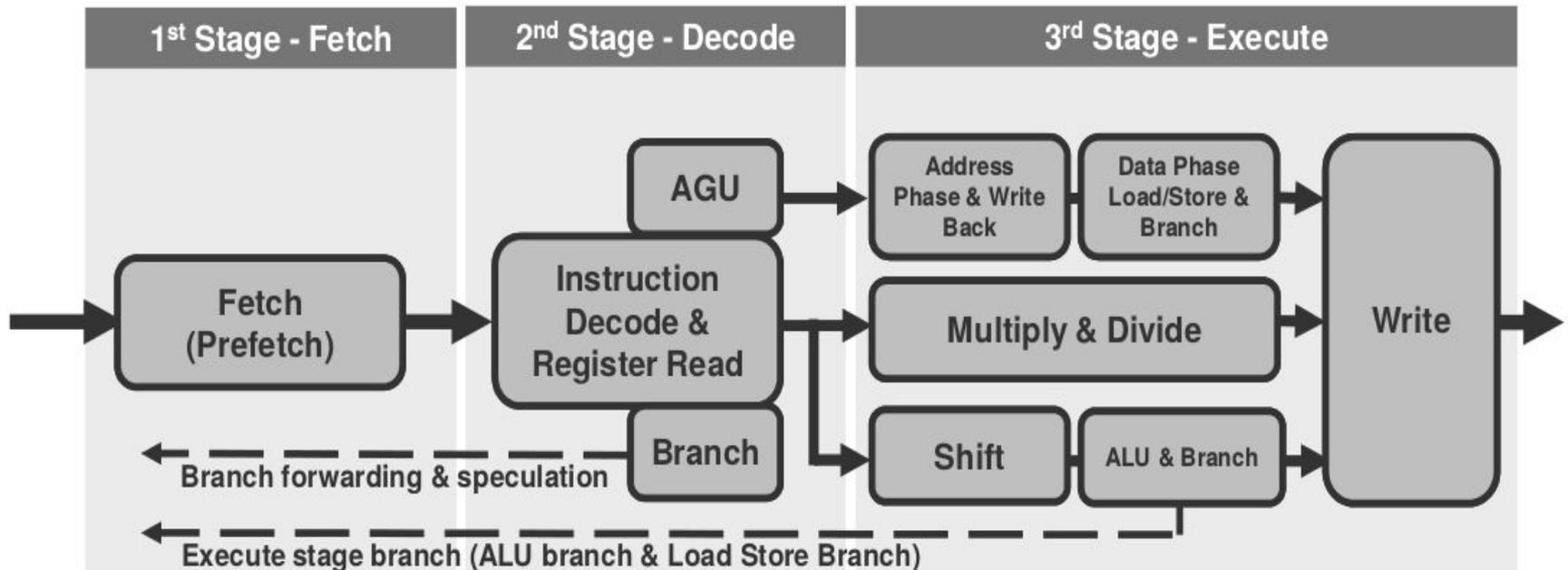
2. Decodificação (Decode)

- Decodificação dos registradores usados na instrução

3. Execução (Execute)

- Leitura de registradores
- Operações lógicas, aritméticas e de deslocamento;
- Escrita em registradores

Pipeline (3 Estágios)



Similar ao pipeline do ARM7TDMI, porém com mais funcionalidades em cada estágio, o que resulta em maior desempenho geral

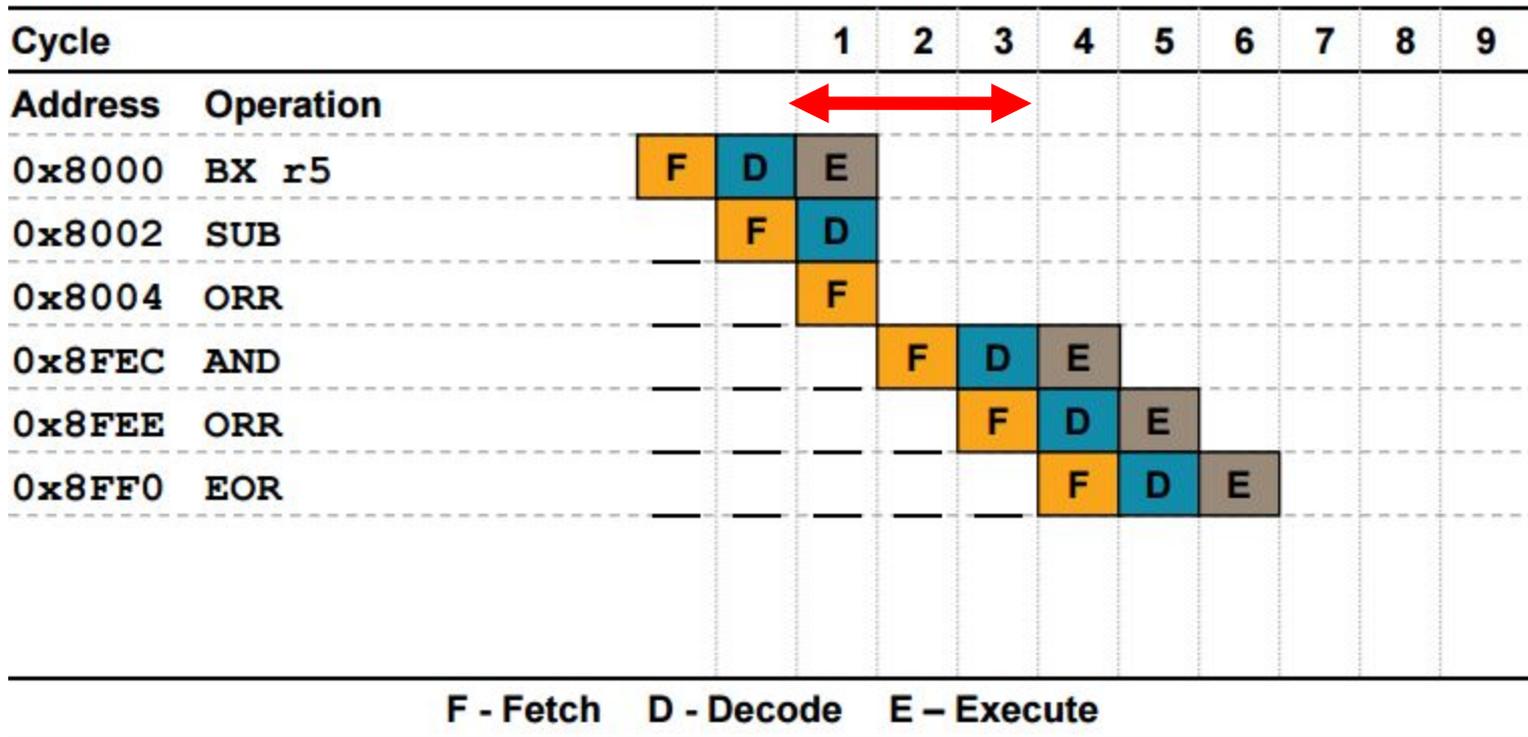
Pipeline: Situação Ideal

Cycle		1	2	3	4	5	6	7	8	9
Operation										
ADD		F	D	E						
SUB			F	D	E					
ORR				F	D	E				
AND					F	D	E			
ORR						F	D	E		
EOR							F	D	E	

F - Fetch D - Decode E - Execute

- Todas as operações realizadas em registradores → 6 instruções em 6 ciclos de clock (ARM Cortex-M3)

Pipeline: Efeito de Saltos



- Pior caso: salto indireto, 3 ciclos de clock para completar o salto (ARM Cortex-M3)

Arquitetura x Organização

- Arquitetura = documento de especificação
 - Instruções
 - Exceções
 - Registradores
 - Memória
- Ex: ARMv4, ARMv7, etc.
- Não tem custo, pode ser obtido diretamente do website da ARM

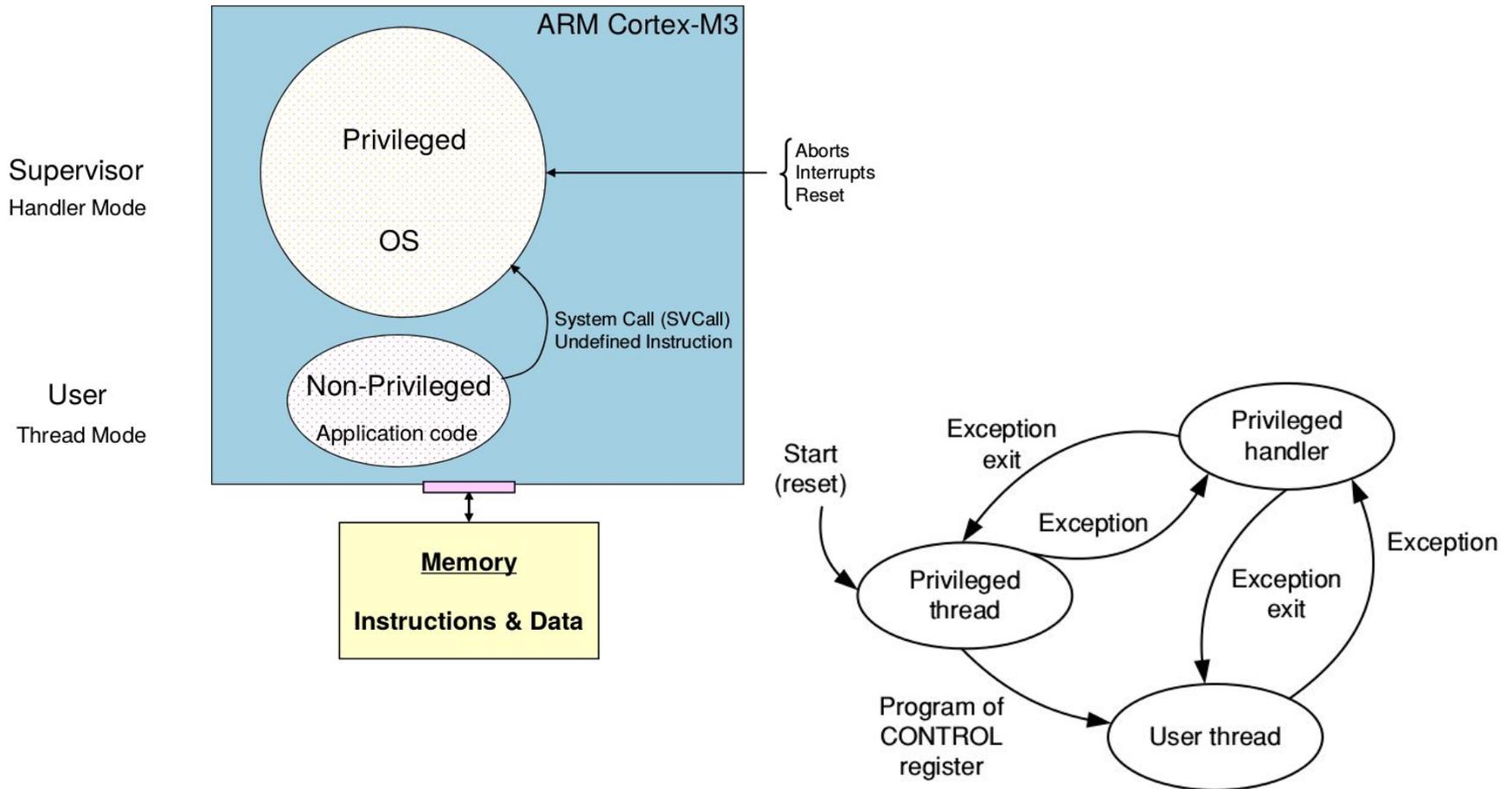
Arquitetura x Organização

- Organização = implementação física (silício)
 - Ex: ARM7TDMI, ARM Cortex-M3, etc.
- ARM vende a implementação de núcleos em VHDL ou máscara de difusão para empresas licenciadas

ARM Cortex-M3

- 18 registradores de 32 bits
- Tratamento muito eficiente de interrupções
- Gerenciamento de consumo de energia
- Projetado para ser programado em C (completamente, até mesmo tratamento de reset, interrupções e exceções)
- Permite uso de sistemas operacionais (RTOS)
 - Modelo
 - Usuário : sem privilégio de execução)
 - Supervisor : com privilégios (processador inicia, reinicia, executa chamada de sistema)

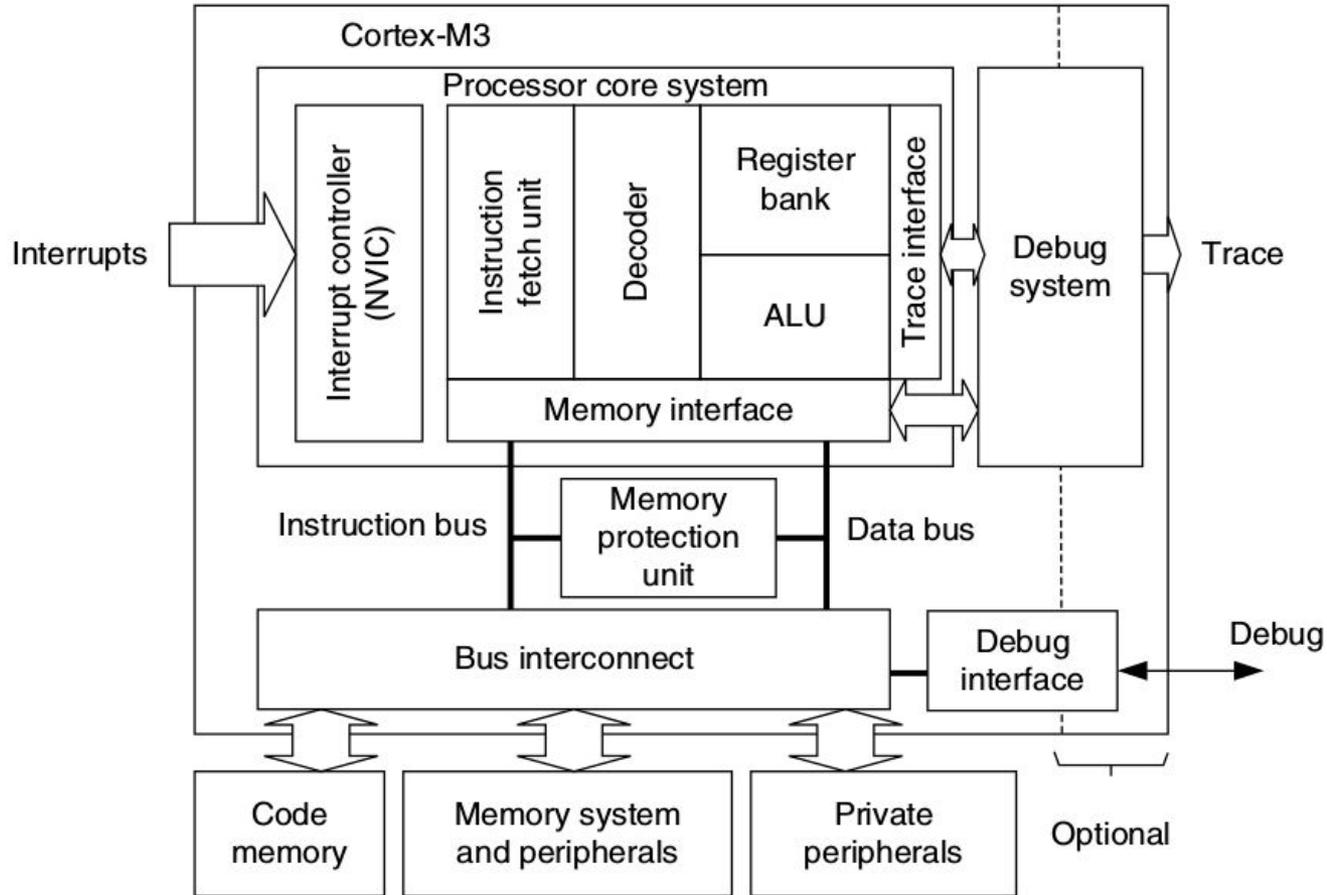
ARM Cortex-M3



ARM Cortex-M3

- Arquitetura ARMv7M
- Sem memória cache ou Unidade de Gerenciamento de Memória (MMU)
- Instrução DIV (divisão)
- Interrupções salvam e recuperam automaticamente o estado do processador

Visão simplificada do Cortex-M3



Unidade de Proteção de Memória (MPU)

- A MPU provê controle de acesso a várias regiões de memória
- Proteção de memória com latência zero
 - 8 regiões em registradores
 - As mesmas regiões são usadas por instruções e dados
 - Tamanho: mínimo de 32 bytes, máximo de 4GB
 - Não há paginação de endereços
- Configurado através de registradores de controle mapeados em memória

Gerenciamento de Consumo de Energia

- Vários modos de baixo consumo (*sleep modes*)
 - Sleep Now
 - Instruções Wait for Interrupt/Event
 - Sleep On Exit
 - Imediatamente após o retorno da última interrupção
 - Deep Sleep
 - Longa duração, PLL desligado
 - Sinal externo SLEEPDEEP
- Controlados pelo NVIC (interrupt controller)

ARM Cortex-M3

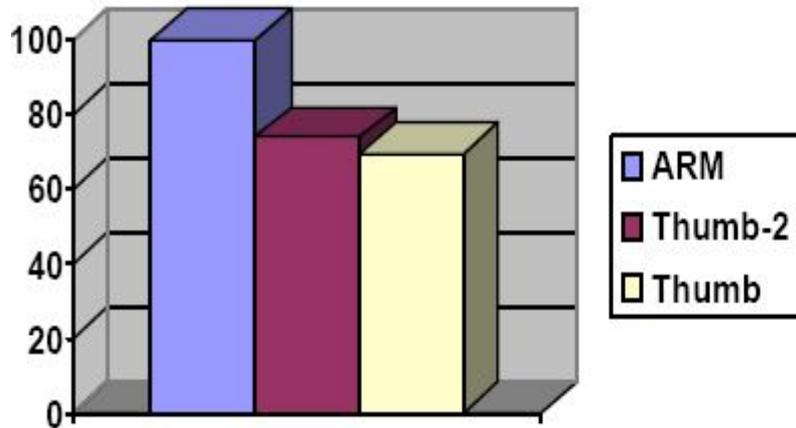
- Controlador de Interrupções é parte da macrocélula Cortex-M3
- Mapa de memória fixo
- Registrador de estado do processador único
- Núcleo de processamento Thumb-2
 - Mistura de instruções de 16 e 32 bit (alta densidade de código), mas não requer alinhamento para instruções de 32 bits

Conjunto de Instruções Thumb-2

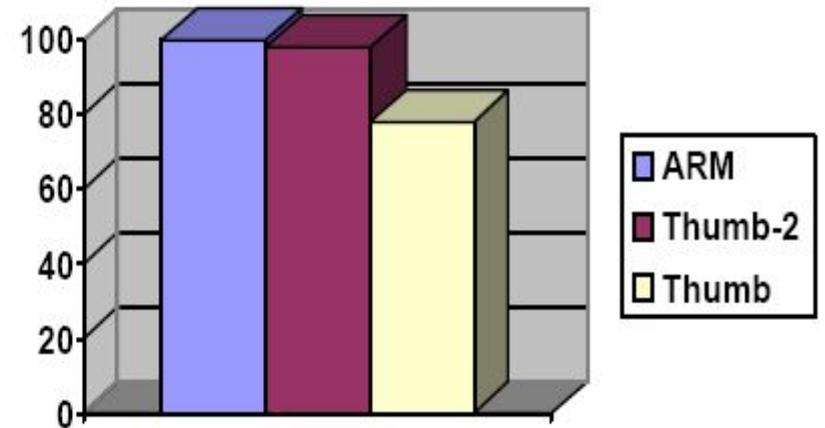
- Comprimento de instruções:
 - Instruções ARM = 32 bits (fixo)
 - Instruções Thumb = 16 bits (fixo)
 - Instruções **Thumb-2** = 16 ou 32 bits (variável)
- Melhor de dois mundos:
 - Aproximadamente 26% de melhora em densidade de código em relação a instruções ARM
 - Aproximadamente 25% de melhora em desempenho em relação a instruções Thumb

Conjunto de Instruções Thumb-2

Code Size



Performance



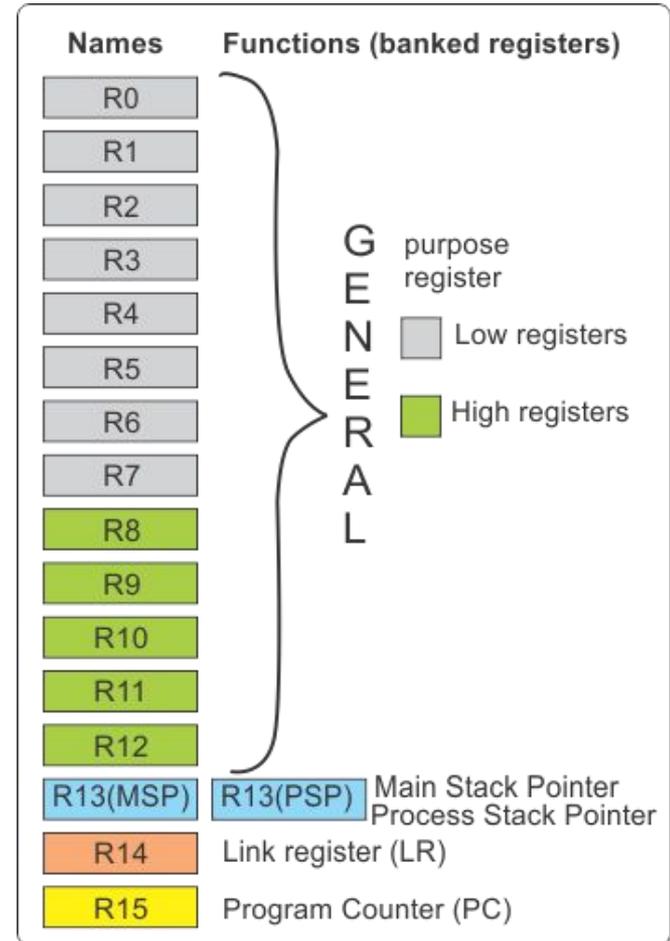
Fonte: ARM

Arquitetura Load/Store

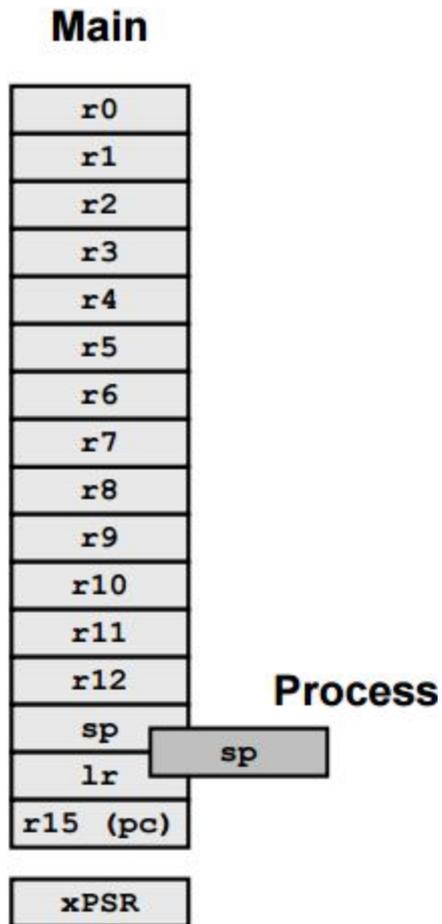
- Acesso à memória:
 - Somente instruções LD leem dados da memória
 - Somente instruções ST escrevem dados na memória
 - Instruções de processamento de dados **não acessam a memória**
- Operações sobre dados em memória requerem:
 1. Leitura em registrador
 2. Operação
 3. Escrita em memória

Registradores (32 bits)

- **13 registradores de propósito geral**
 - R0 a R7 (low registers) - **Thumb & Thumb-2**
 - R8 a R12 (high registers) - **Thumb-2 32bit**
- **3 registradores de uso/significado especial**
 - R13 = Stack Pointer (SP) - **Main + Process**
 - R14 = Link Register (LR) - **PC de retorno**
 - R15 = Program Counter (PC)
- **1 registrador de propósito especial**
 - xPSR = Program Status Register



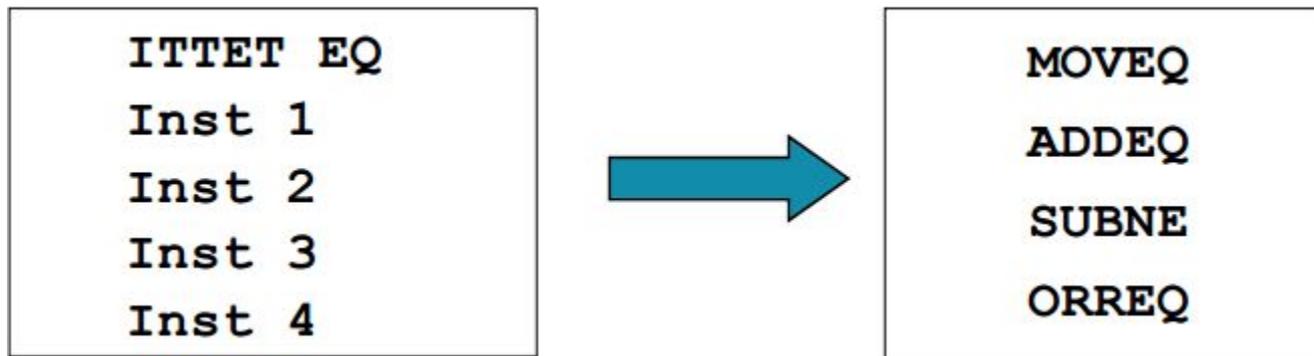
Registradores (32 bits)



- Registradores **xPSR, PC, LR, R12, R3, R2, R1 e R0** são todos armazenados automaticamente na pilha quando uma interrupção ocorre.

Execução Condicional

- Bloco If-Then (IT)
 - Até 3 instruções condicionais “then” (T) ou “else” (E) podem ser adicionadas
 - Transforma até 4 instruções consecutivas em condicionais



Execução Condicional

- Qualquer código de condição ARM pode ser utilizado (ver tabela a seguir)
- Instruções de 16 bits dentro do bloco não afetam flags, exceto a instrução de comparação
- Instruções de 32 bits podem ou não afetar flags (conforme o uso ou não do sufixo S)
- Estado do bloco If-Then é armazenado no CPSR
 - Bloco If-Then pode ser interrompido de forma segura
 - Saltar de dentro ou para dentro de um bloco If-Then **NÃO** é permitido

Códigos de Condição ARM

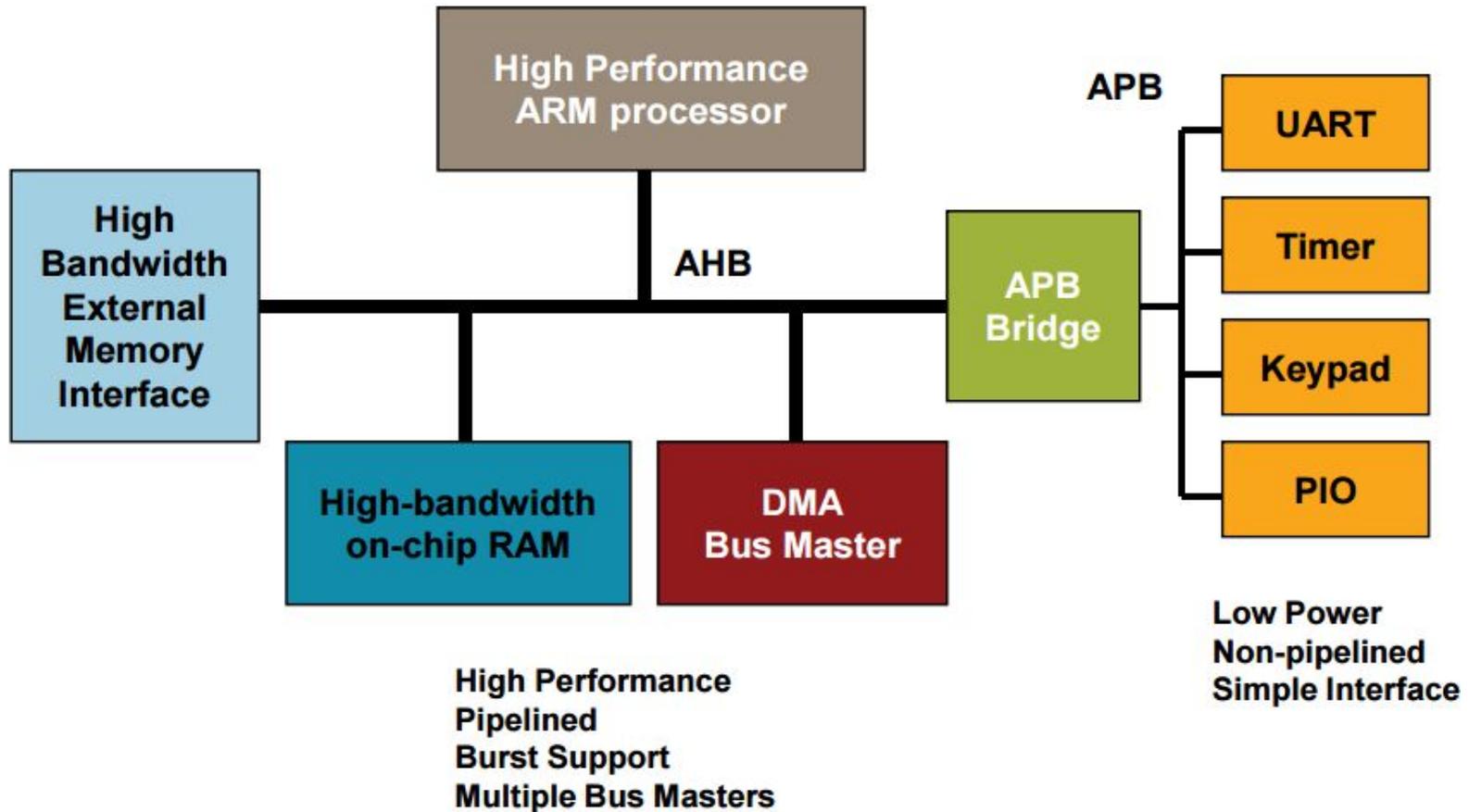
Mn	Descrição	Flags			Mn	Descrição	Flags						
EQ	equal	Mn	Descrição	Flags	Mn	Descrição	Flags	EQ	equal	Z	NE	not equal	Z
		EQ	equal	Z	NE	not equal	Z						
		CS	carry set	C	CC	carry clear	\bar{C}						
		HS	higher or same	C	LO	lower	\bar{C}						
		MI	minus/negative	N	PL	plus/positive	\bar{N}						
		VS	overflow	V	VC	no overflow	\bar{V}						
		HI	higher	ZC	LS	lower or same	Z + \bar{C}						
GE	greater or equal	NV + NP	LT	less than	NP + NV								
GT	greater than	NZV + NZP	LE	less or equal	Z + NP + NV								
CS HS	carry set higher or same	Mn	Descrição	Flags	Mn	Descrição	Flags	CC LO	carry clear lower	C	CC	carry clear	\bar{C}
		EQ	equal	Z	NE	not equal	\bar{Z}						
		CS	carry set	C	CC	carry clear	\bar{C}						
		HS	higher or same	C	LO	lower	\bar{C}						
		MI	minus/negative	N	PL	plus/positive	\bar{N}						
		VS	overflow	V	VC	no overflow	\bar{V}						
		HI	higher	ZC	LS	lower or same	Z + \bar{C}						
GE	greater or equal	NV + NP	LT	less than	NP + NV								
GT	greater than	NZV + NZP	LE	less or equal	Z + NP + NV								
MI	minus/negative	Mn	Descrição	Flags	Mn	Descrição	Flags	PL	plus/positive	N	PL	plus/positive	\bar{N}
		EQ	equal	Z	NE	not equal	Z						
		CS	carry set	C	CC	carry clear	\bar{C}						
		HS	higher or same	C	LO	lower	\bar{C}						
		MI	minus/negative	N	PL	plus/positive	\bar{N}						
		VS	overflow	V	VC	no overflow	\bar{V}						
		HI	higher	ZC	LS	lower or same	Z + \bar{C}						
GE	greater or equal	NV + NP	LT	less than	NP + NV								
GT	greater than	NZV + NZP	LE	less or equal	Z + NP + NV								
VS	overflow	Mn	Descrição	Flags	Mn	Descrição	Flags	VC	no overflow	V	VC	no overflow	\bar{V}
		EQ	equal	Z	NE	not equal	Z						
		CS	carry set	C	CC	carry clear	\bar{C}						
		HS	higher or same	C	LO	lower	\bar{C}						
		MI	minus/negative	N	PL	plus/positive	\bar{N}						
		VS	overflow	V	VC	no overflow	\bar{V}						
		HI	higher	ZC	LS	lower or same	Z + \bar{C}						
GE	greater or equal	NV + NP	LT	less than	NP + NV								
GT	greater than	NZV + NZP	LE	less or equal	Z + NP + NV								
HI	higher	Mn	Descrição	Flags	Mn	Descrição	Flags	LS	lower or same	ZC	LS	lower or same	Z + \bar{C}
		EQ	equal	Z	NE	not equal	Z						
		CS	carry set	C	CC	carry clear	\bar{C}						
		HS	higher or same	C	LO	lower	\bar{C}						
		MI	minus/negative	N	PL	plus/positive	\bar{N}						
		VS	overflow	V	VC	no overflow	\bar{V}						
		HI	higher	ZC	LS	lower or same	Z + \bar{C}						
GE	greater or equal	NV + NP	LT	less than	NP + NV								
GT	greater than	NZV + NZP	LE	less or equal	Z + NP + NV								
GE	greater or equal	Mn	Descrição	Flags	Mn	Descrição	Flags	LT	less than	NV + NP	LT	less than	NP + NV
		EQ	equal	Z	NE	not equal	Z						
		CS	carry set	C	CC	carry clear	\bar{C}						
		HS	higher or same	C	LO	lower	\bar{C}						
		MI	minus/negative	N	PL	plus/positive	\bar{N}						
		VS	overflow	V	VC	no overflow	\bar{V}						
		HI	higher	ZC	LS	lower or same	Z + \bar{C}						
GT	greater than	NZV + NZP	LE	less or equal	Z + NP + NV								
GT	greater than	Mn	Descrição	Flags	Mn	Descrição	Flags	LE	less or equal	NZV + NZP	LE	less or equal	Z + NP + NV
		EQ	equal	Z	NE	not equal	Z						
		CS	carry set	C	CC	carry clear	\bar{C}						
		HS	higher or same	C	LO	lower	\bar{C}						
		MI	minus/negative	N	PL	plus/positive	\bar{N}						
		VS	overflow	V	VC	no overflow	\bar{V}						
		HI	higher	ZC	LS	lower or same	Z + \bar{C}						
GT	greater than	NZV + NZP	LE	less or equal	Z + NP + NV								

Mapa de Memória

0xFFFFFFFF	System level	Private peripherals including build-in interrupt controller (NVIC), MPU control registers, and debug components
0xE0000000		
0xDFFFFFFF	External device	Mainly used as external peripherals
0xA0000000		
0x9FFFFFFF	External RAM	Mainly used as external memory
0x60000000		
0x5FFFFFFF	Peripherals	Mainly used as peripherals
0x40000000		
0x3FFFFFFF	SRAM	Mainly used as static RAM
0x20000000		
0x1FFFFFFF	CODE	Mainly used for program code. Also provides exception vector table after power up
0x00000000		

ARM-AMBA

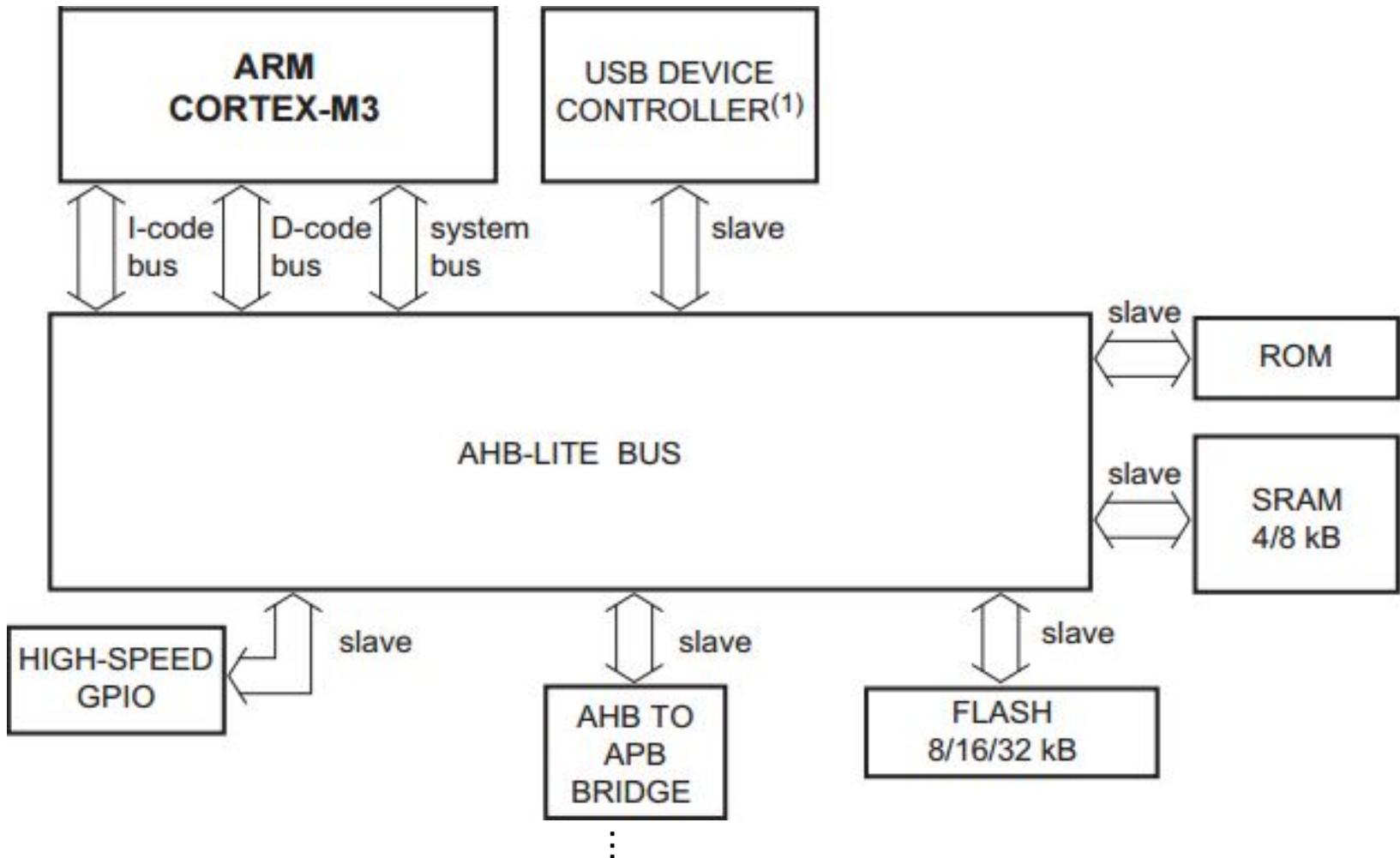
Advanced Microcontroller Bus Architecture



AHB - Advanced High-Performance Bus

APB - Advanced Peripheral Bus

Diagrama em Blocos – LPC1343



Mapa de Memória – LPC1343

