




## Aula 8

**Engenharia de Sistemas Embarcados**  
 Prof. Abel Guilhermino  
 Tópico: Arquitetura ARM




### Repertório de Instruções


- O conjunto de instruções possui uma largura fixa de 32 bits
- É o conjunto padrão de instruções quando a CPU se encontra no estado ARM (T=0 no CPSR)
- Instruções
  - Instruções de Manipulação de Dados
  - Instruções Lógicas e Aritméticas
  - Instruções de Teste e Desvio
  - Instruções de Controle da CPU
  - Instruções de comunicação e controle de co-processadores



Grupo de Engenharia da Computação - CIn / UFPE

### Instruções de Manipulação de Dados ARM


- Instruções de Manipulação de Dados
  - **LDR**: carrega um registrador com valor lido da memória
  - **STR**: armazena o conteúdo de um registrador em um endereço de memória
  - **LDM**: carrega uma lista de registradores com o conteúdo localizado a partir do endereço especificado pelo registrador base.
  - **STM**: armazena uma lista de registradores nas posições de memória a iniciar pelo endereço especificado pelo registrador base.
  - **SWP**: troca o conteúdo do registrador com o do endereço de memória especificado
  - **MOV**: move uma constante ou o valor de um registrador para outro
  - **MVN**: o registrador destino recebe o registrador fonte com os bits invertidos.



Grupo de Engenharia da Computação - CIn / UFPE

### Instruções de Lógicas e Aritméticas ARM


- Instruções Lógicas e Aritméticas
  - **ADC**: adição com transporte (carry)
  - **SBC**: subtração com empréstimo (carry)
  - **ADD**: adição
  - **SUB**: subtração
  - **RSB**: subtração reversa
  - **RSC**: subtração reversa com empréstimo (carry)
  - **MUL**: multiplicação
  - **MLA**: multiplicação com acúmulo
  - **AND**: operação lógica AND bit a bit
  - **ORR**: operação lógica OR bit a bit
  - **EOR**: operação lógica EOR bit a bit
  - **BIC**: apaga bits do registrador



Grupo de Engenharia da Computação - CIn / UFPE

### Instruções de Teste e Desvio ARM


- Instruções Teste e Desvio
  - **B**: desvio
  - **BL**: chamada de sub-rotina
  - **BX**: desvio com mudança de estado (ARM/Thumb)
  - **CMN**: compara (soma) registradores
  - **CMP**: compara (subtrai) registradores
  - **TEQ**: testa igualdade (operação EOU)
  - **TST**: testa os bits (operação E)



Grupo de Engenharia da Computação - CIn / UFPE

### Instruções Controle da CPU e funções Co-Processador

- Instruções Controle da CPU
  - **MRS**: copia o CPSR para o registrador especificado
  - **MSR**: copia o registrador para o CPSR
  - **SWI**: interrupção por software
- Instruções Co-Processador
  - **CDP**: inicia uma determinada operação no co-processador especificado
  - **LDC**: carrega um registrador do co-processador indicado com um valor lido da memória
  - **STC**: armazena o conteúdo de um registrador do co-processador especificado em um endereço de memória
  - **MCR**: move um registrador da CPU para um registrador do co-processador
  - **MRC**: move um registrador do co-processador para um registrador da CPU



Grupo de Engenharia da Computação - CIn / UFPE

### Formato de Instruções

- Formato

function	operand 1 address	operand 2 address	operand 3 address
----------	-------------------	-------------------	-------------------

### Instruções de Manipulação de Dados e Lógicas e Aritméticas

#### Operações Aritméticas

ADD r0, r1, r2	r0 := r1 + r2
ADC r0, r1, r2	r0 := r1 + r2 + C
SUB r0, r1, r2	r0 := r1 - r2
SBC r0, r1, r2	r0 := r1 - r2 + C - 1
RSB r0, r1, r2	r0 := r2 - r1
RSC r0, r1, r2	r0 := r2 - r1 + C - 1

#### Movimentação de Registradores

MOV r0, r2	r0 := r2
MVN r0, r2	r0 := not r2

#### Operações Lógicas com Bits

AND r0, r1, r2	r0 := r1 and r2
ORR r0, r1, r2	r0 := r1 or r2
EOR r0, r1, r2	r0 := r1 xor r2
BIC r0, r1, r2	r0 := r1 and (not) r2

#### Operações de Comparação

CMP r1, r2	set cc on r1 - r2
CMN r1, r2	set cc on r1 + r2
TST r1, r2	set cc on r1 and r2
TEQ r1, r2	set cc on r1 xor r2

### Instruções de Manipulação de Dados

- LDR
  - Realizam carga de um registrador (LDR) com valores lidos da memória
  - LDR R1, [R2]
    - Copia a word do endereço indicado por R2 para o registrador R1
  - LDR R0, [R2, R3]
    - Copia a word do endereço resultante da soma de R2 mais R3 para o registrador R0. Os registradores R2 e R3 não são alterados.
- STR
  - Permitem escrever dados de um registrador (STR) na memória.
  - STR R3, [R5, #4]
    - Copia a word do registrador R3 para o endereço indicado por R5 mais um offset igual a 4.

### Instruções de Transferência de Dados

- LDM
  - Realizam carga de múltiplos registradores (LDM) com valores lidos da memória
  - LDM Rbase, [lista\_registradores]
    - Transfere o conteúdo dos registradores listados entre chaves para o endereço especificado pelo registrador Rbase.
- STM
  - Permitem escrever dados de múltiplos registradores (STM) na memória.
  - STM Rbase, [lista\_Registradores]
    - Transfere o conteúdo dos registradores listados entre chaves para o endereço da memória especificado pelo registrador Rbase.

LDM r1, (r0, r2, r5)	r0 := mem <sub>32</sub> [r1] r2 := mem <sub>32</sub> [r1 + 4] r5 := mem <sub>32</sub> [r1 + 8]
----------------------	--

### Operações de deslocamento do ARM

- O ARM não possui instruções de deslocamento
- É permitido o deslocamento de um operando de instrução aritmética
- A operação de deslocamento de n bits é feita pelo barrel shifter, e esta não insere atrasos na execução da instrução.
- Barrel shifter: circuito lógico combinacional digital capaz de realizar tarefas de deslocamento de múltiplos bits à direita ou à esquerda em uma única operação.
  - Barrel Shifter - Deslocamento para Esquerda
    - LSL #5 => multiplica por 2<sup>5</sup> => multiplica por 32
  - Barrel Shifter - Deslocamento para Direita
    - LSR #5 => divide por 2<sup>5</sup> => divide por 32
    - Add r3, r2, r1, LSL #3 ;r3=r2+8 x r1

Mnemonic	Operação
ASL, LSL	Deslocamento lógico à esquerda (L a 32 bits): o conteúdo do registrador é deslocado "n" bits à esquerda, os bits mais significativos são preenchidos com 0 e o bit excedente é armazenado no carry (C do CPSR). O C guarda o último bit deslocado para fora do registrador.
LSR	Deslocamento lógico à direita (R a 32 bits): o conteúdo do registrador é deslocado "n" bits à direita, os bits mais significativos são preenchidos com 0 e o bit excedente é armazenado no carry (C do CPSR). O C guarda o último bit deslocado para fora do registrador.
ASR	Deslocamento aritmético à direita (R a 32 bits): o conteúdo do registrador é deslocado "n" bits à direita, os bits mais significativos são mantidos com 0 ou 1 de acordo com o bit excedente é armazenado no carry (C do CPSR). O C guarda o último bit deslocado para fora do registrador. Desta forma, preserva-se o sinal do número deslocado.
ROR	Rotação lógica à direita (R a 32 bits): o conteúdo do registrador é deslocado "n" bits à direita, os bits mais significativos são mantidos com os bits excedentes (de qual uma estrutura circular). O carry (C do CPSR) guarda o último bit deslocado para fora do registrador.
RRX	Rotação lógica à direita estendida (R a 32 bits): o conteúdo do registrador é deslocado um bit à direita. O conteúdo previo do C é inserido no bit mais significativo e em seguida recebe o bit menos significativo do registrador.

### Sufixos nas instruções do ARM

- Transferência com outros tipos de dados com uso dos seguintes sufixos:
  - B: transferência de bytes (LDRB, STRB)
  - SB: transferência de bytes sinalizados (LDRSB/STRSB)
  - H: transferência de half words (valores de 16bits) (LDRH/STRH)
  - SH: transferência de half words sinalizados (LDRSH/STRSH)
  - D: transferência de double words (valores de 64 bits) (LDRD/STRD) - versões 5e da CPU ARM e posteriores

## Instruções de Processamento de Dados

- SWP
  - SWP Ra, Rb, [Rc]
    - Ra é carregado com o conteúdo da posição de memória especificada por Rc e em seguida essa posição de memória recebe o valor armazenado no registrador Rb.
    - Obs: esta instrução realiza duas operações (leitura da memória e em seguida uma escrita) de forma atômica. Útil para uso em semáforos de software para sinalização de processos.



Grupo de Engenharia da Computação - CIn / UFPE



## Instruções de Processamento de Dados

- MUL Ra, Rb, Rc
  - Implementa a multiplicação
  - Elas suportam multiplicação de operandos de 32 bits com ou sem sinal com um resultado de 32 bits.
  - Multiplica o conteúdo do registrador Rb pelo conteúdo do registrador Rc e armazena o resultado no registrador Ra
- MLA Ra, Rb, Rc, Rd
  - Implementa a multiplicação e acúmulo
  - Elas suportam multiplicação de operandos de 32 bits com ou sem sinal com um resultado de 32 bits.
  - Multiplica o conteúdo de Rb pelo de Rc e soma o resultados ao Rd. O resultado final é armazenado em Ra.
- Obs: Nas versões ARM7TDMI, há suporte para multiplicações longas (32x32=64bits) bem como multiplicações e acúmulos.



Grupo de Engenharia da Computação - CIn / UFPE



## Instruções de Processamento de Dados

- B teste
  - Desvio incondicional para teste
- BL calcula
  - Chama a sub-rotina calcula
  - BL = *Branch and link*, opera de forma semelhante a B, porém antes de realizar o desvio, o valor prévio ao PC é salvo em LR(R14). Desta forma ela é usada para chamada de subrotinas.
- BX R1
  - Desvia para o endereço especificado por R1, e se o bit 0 do registrador R1 for igual a 1, passa a operar no estado Thumb (bit T do CPSR é setado). Caso contrário, modo ARM (bit T do CPSR é apagado)
  - BX = *branch and exchange*.



Grupo de Engenharia da Computação - CIn / UFPE



## Instruções de Transferência de Registradores de Controle

- MRS
  - Permite a transferência da palavra de estado do processador (CPSR ou SPSR) para um registrador qualquer da CPU.
  - Exemplo:
    - MRS Ra, <CPSR/SPSR>
      - Transfere o CPSR ou SPSR ao registrador Ra da CPU.
- MSR
  - Permite a transferência do conteúdo de um registrador qualquer da CPU ou um operando imediato, para a palavra de estado (CPSR ou SPSR).
  - Exemplo:
    - MSR <CPSR/SPSR>, Ra
    - MSR <CPSR/SPSR>, <#constante>



Grupo de Engenharia da Computação - CIn / UFPE



## Interrupção por Software

- SWI
  - A instrução SWI efetua a chamada interrupção por software que, nas CPUs ARM, provoca a entrada no modo supervisor (svc).
  - Esta é a única forma que um programa rodando em modo de usuário (não privilegiado) possui para entrar em modo privilegiado.
  - Ela causa uma mudança para o modo supervisor e a rotina de tratamento é executada
  - A rotina de tratamento depende do conteúdo do campo comentário. O campo comentário (24 bits) é ignorado pelo CPU, mas que pode ser usado pelo programa para diferenciar a razão da entrada no supervisor.
  - sintaxe:
    - SWI campo\_comentário
      - O parâmetro do campo\_comentário funciona como uma espécie de parâmetro permitindo que S.O. implemente diversas funções que são acessadas pelo programa rodando em modo usuário através da SWI



Grupo de Engenharia da Computação - CIn / UFPE



## Instruções de Comunicação e Controle de Co-processadores

- LDC, STC, CDP, MCR, MRC
  - Instruções usadas para comunicação e controle dos co-processadores internos.
  - Os co-processadores são dispositivos de hardware capazes de controlar operações externas a CPU, como: CP15 para gerenciamento de memória MMUs, controle do sistema, co-processadores matemáticos, co-processadores gráficos, etc.
  - A instrução LDC: é utilizada para carregar dados da memória para um registrador de um dos co-processadores.
  - A instrução STC para armazenar na memória um dado proveniente de um registrador de um co-processador.
  - A instrução CDP envia comandos a um co-processador.
  - A instrução MCR é utilizada para transferir um registrador da CPU diretamente para um registrador de um co-processador.
  - A instrução MRC é utilizada para transferir um registrador de um co-processador para um registrador da CPU.





Grupo de Engenharia da Computação - CIn / UFPE





### Instruções de Teste e Desvio

Branch	Interpretation	Normal uses
B	Unconditional	Always take this branch
BAL	Always	Always take this branch
BEQ	Equal	Comparison equal or zero result
BNE	Not equal	Comparison not equal or non-zero result
BPL	Plus	Result positive or zero
BMI	Minus	Result minus or negative
BCC	Carry clear	Arithmetic operation did not give carry-out
BLO	Lower	Unsigned comparison gave lower
BCS	Carry set	Arithmetic operation gave carry-out
BHS	Higher or same	Unsigned comparison gave higher or same
BVC	Overflow clear	Signed integer operation: no overflow occurred
BVS	Overflow set	Signed integer operation: overflow occurred
BGT	Greater than	Signed integer comparison gave greater than
BGE	Greater or equal	Signed integer comparison gave greater or equal
BLT	Less than	Signed integer comparison gave less than
BLE	Less or equal	Signed integer comparison gave less than or equal
BHI	Higher	Unsigned comparison gave higher
BLS	Lower or same	Unsigned comparison gave lower or same


 Grupo de Engenharia da Computação - CIn / UFPE
 



### Instruções Thumb

- Esse conjunto de instruções alternativo de 16 bits é selecionado quando a CPU executa uma instrução BX(*branch and exchange*) e o registrador com o endereço de destino possui o seu bit menos significativo em nível lógico '1' o que provoca a transição para o estado Thumb (bit T = 1 no CPSR).
- O conjunto de instruções Thumb é baseado no conjunto de instruções ARM, porém comprimido para uma largura de 16 bits.
- As instruções Thumb permitem realizar praticamente as mesmas operações disponíveis no estado ARM, porém ocupam metade do espaço em memória, o que resulta uma maior densidade de código.
- A contrapartida da sua utilização é que, em alguns casos, um programa Thumb pode ser mais lento que o mesmo programa codificado com instruções ARM.


 Grupo de Engenharia da Computação - CIn / UFPE
 



### Instruções Thumb

- As instruções Thumb classificam-se conforme as seguintes categorias:
  - Manipulação de Dados
  - Lógicas e Aritméticas
  - Teste e Desvio
  - Controle da CPU


 Grupo de Engenharia da Computação - CIn / UFPE
 



### Instruções Thumb (Manipulação de Dados)

- Manipulação de Dados
  - LDR: carrega um registrador com um valor lido da memória
  - STR: armazena o conteúdo de um registrador em um endereço de memória
  - LDMIA: carrega uma lista de registradores com o conteúdo localizado a partir do endereço especificado pelo registrador base e o pós-incrementa
  - STMIA: armazena uma lista de registradores nas posições de memória iniciando pelo endereço especificado pelo registrador base e o pós-incrementa.
  - MOV: move uma constante ou o valor de um registrador para outro.
  - MVN: complementa e move um valor de um registrador
  - PUSH: armazena um ou mais registradores na pilha
  - POP: lê um ou mais registradores na pilha


 Grupo de Engenharia da Computação - CIn / UFPE
 



### Instruções Thumb (Lógica e Aritmética)

- Lógica e Aritmética
  - ADC: adição com transporte (carry)
  - SBC: subtração com empréstimo (carry)
  - ADD: adição
  - SUB subtração
  - MUL: multiplicação
  - NEG: inverte o sinal do conteúdo do registrador
  - AND: operação lógica AND bit a bit
  - ORR: operação lógica OR bit a bit
  - EOR: operação lógica EOR bit a bit
  - BIC: apaga bits do registrador
  - ASR: deslocamento aritmético à direita
  - LSL: deslocamento lógico à esquerda
  - LSR: deslocamento lógico à direita
  - ROR: rotação à direita


 Grupo de Engenharia da Computação - CIn / UFPE
 



### Instruções Thumb (Teste e Desvio, Controle da CPU)

- Teste e Desvio
  - B: desvio
  - Bxx: desvio condicional
  - BL: chamada de sub-rotina
  - BX: desvio com mudança de estado (ARM/Thumb)
  - CMN: compara (soma) registradores
  - CMP: compara (subtrai) registradores
  - TST: testa bits (operação E)
- Controle da CPU
  - SWI: interrupção por software


 Grupo de Engenharia da Computação - CIn / UFPE
 

### Instruções Thumb



- As instruções Thumb não admitem prefixos e sufixos como aqueles presentes nas instruções ARM.
- A alteração do estado dos flags do CPSR não é opcional.

Grupo de Engenharia da Computação - CIn / UFPE

### Instruções Thumb

- LDR e STR
  - São similares as instruções ARM. Destinam-se à operação de cara (LDR) ou armazenamento (STR) de um registrador.
    - LDR Ra, [Rb]
    - STR Ra, [Rb]
  - Na instrução LDR o conteúdo de 32 bits da posição de memória apontada por Rb será lido e armazenado em Ra.
  - Na instrução STR, a operação é inversa, o conteúdo do registrador Ra é armazenado no endereço indicado por Rb.
  - Tais instruções somente podem acessar os registradores baixos (R0 a R7).

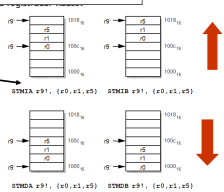
Grupo de Engenharia da Computação - CIn / UFPE



### Instruções Thumb

- LDMIA
  - Realizam carga de múltiplos registradores (LDM) com valores lidos da memória
    - LDMIA Rbase, {lista\_registradores}
      - Transfere o conteúdo dos registradores listados entre chaves para o endereço especificado pelo registrador Rbase.
  - STMIA
    - Permitem escrever dados de múltiplos registradores (STM) na memória.

Tais instruções somente podem acessar os registradores baixos (R0 a R7).

LDMIA r1, {r0, r2, r5}    r0 := mem<sub>32</sub>[r1]  
 r2 := mem<sub>32</sub>[r1 + 4]  
 r5 := mem<sub>32</sub>[r1 + 8]





Grupo de Engenharia da Computação - CIn / UFPE

### Instruções Thumb



- PUSH e POP
  - Permitem a manipulação da pilha de memória apontada pelo registrador SP (R13 no estado ARM).
  - A instrução PUSH permite armazenar o conteúdo de um ou mais registradores baixos (R0 a R7) na pilha.
  - Adicionalmente também é possível armazenar o LR (R14 no estado ARM).
  - Sintaxe:
    - PUSH {lista\_registradores}
  - A instrução POP carrega um ou mais registradores baixos (R0 a R7) a partir da pilha.
  - Adicionalmente também é possível efetuar o carregamento do PC.
    - POP {lista\_registradores}

Grupo de Engenharia da Computação - CIn / UFPE

### Instruções Thumb



- MOV
  - A instrução MOV permite a movimentação de dados de um registrador para outro.
  - Esta é uma das poucas instruções Thumb que permitem acessar tanto registradores baixos (R0 a R7) quanto altos (R8 a R15).
  - Exemplo:
    - MOV R3, R1 ; copia R1 para R3 (afeta os flags N e Z)
    - MOV R2, #50 ; copia a constante 50 para R2 (N=0, Z=0)
    - MOV R0, R9 ; copia R9 para R0
    - MOV R15, R1 ; copia R1 para R15(PC), provoca desvio
    - MOV R15, R14 ; carrega R14(LR) no R15(PC), provoca retorno de uma sub-rotina

Grupo de Engenharia da Computação - CIn / UFPE

### Instruções Thumb

- MVN
  - A instrução MVN, tal qual sua equivalente ARM, inverte e copia o conteúdo de um registrador para outro.
    - MVN Ra, Rb
  - A instrução faz a leitura de Rb, complementa o seu conteúdo e armazena o resultado em Ra.
  - Ambos devem ser registradores baixos (R0 a R7)
  - Alguns flags do CPSR são alterados de acordo com o resultado da operação.

Grupo de Engenharia da Computação - CIn / UFPE

## Instruções Thumb

- ADD e ADC
  - Realizam a operação de adição aritmética de dois registradores.
  - A instrução ADD pode operar com registradores altos e baixos, enquanto que a ADC que realiza a adição dos registradores mais o carry (C), somente pode operar com registradores baixos (R0 a R7).
  - Exemplos
    - ADD R2, R8 ; R2 = R2 + R8
    - ADC R0, R1 ; R0 = R0 + R1 + C



## Instruções Thumb

- SUB e SBC
  - Realizam a operação de subtração aritmética de dois registradores.
  - Essas instruções apenas operam com os registradores baixos (R0 a R7).
  - Exemplos
    - SUB Ra, Rb, Rc ; Ra = Rb - Rc
    - SBC Ra, Rb ; Ra = Ra - Rb - Não(C)
    - SUB R0, #100 ; subtrai 100 do registrador R0



## Instruções Thumb

- NEG
  - Inverte o sinal matemático do registrador (equivalente a subtrair o registrador de 0, ou realizar o complemento de dois).
  - Essa instrução somente opera com os registradores baixos (R0 a R7).
  - Exemplos
    - NEG Ra, Rb ; Ra = - Rb
  - OBS: os flags de CPSR são alterados de acordo com o resultado da operação.



## Instruções Thumb

- MUL
  - Multiplica dois registradores.
  - Essa instrução somente opera com os registradores baixos (R0 a R7).
  - Exemplos
    - MUL Ra, Rb ; Ra = Ra \* Rb
  - OBS: os flags N, Z, C, V do CPSR são alterados de acordo com o resultado da operação.



## Instruções Thumb

- AND, ORR e EOR
  - Operações lógicas bit a bit.
  - Sintaxe:
    - AND Ra, Rb ; Ra = Ra E Rb
    - ORR Ra, Rb ; Ra = Ra OU Rb
    - EOR Ra, Rb ; Ra = Ra EOU Rb
  - Essas instruções somente operam com os registradores baixos (R0 a R7). Os flags N e Z do CPSR são alterados de acordo com o resultado da operação.



## Instruções Thumb

- BIC
  - Apaga os bits especificados do registrador.
  - Essas instruções somente operam com os registradores baixos (R0 a R7).
  - Sintaxe:
    - BIC Ra, Rb ; Ra = Ra E (Não Rb)
  - Os flags N e Z do CPSR são alterados de acordo com o resultado da operação.



## Instruções Thumb

- ASR, LSL, LSR e ROR
  - Essas instruções permitem realizar operações de rotação e deslocamento de bits.
  - Essas instruções somente operam com os registradores baixos (R0 a R7).
  - Sintaxe:
    - ASR Ra, Rb ; Ra = Ra ASR Rb => desloc. arit. à direita
    - LSL Ra, Rb ; Ra = Ra << Rb => desloc. log. à esquerda
    - LSR Ra, Rb ; Ra = Ra >> Rb => desloc. log. à direita
    - ROR Ra, Rb ; Ra = Ra ROR Rb => rotação a direita
  - Os flags N, Z e C do CPSR são alterados de acordo com o resultado da operação.



Grupo de Engenharia da Computação - CIn / UFPE



## Instruções Thumb

- B
  - Permite realizar um desvio relativo incondicional.
  - Sintaxe:
    - B inicio ; desvia para inicio
- Bxx
  - As instruções de desvio condicional Thumb são:
    - BEQ offset9 ; desvia se igual (Z=1)
    - BNE offset9 ; desvia se diferente (Z=0)
    - BCS offset9 ; desvia se C=1
    - BCC offset9 ; desvia se C=0
    - BMI offset9 ; desvia se negativo (N=1)
    - BPL offset9 ; desvia se positivo ou zero (N=0)
    - BVS offset9 ; desvia se overflow (V=1)
    - BVC offset9 ; desvia se não overflow (V=0)
    - BHI offset9 ; desvia se maior (C=1 e Z=0)
    - BLT offset9 ; desvia se menor (N!>V)



Grupo de Engenharia da Computação - CIn / UFPE



## Instruções Thumb

- BL
  - Realiza o desvio longo, permitindo que o endereço de destino esteja 4Mbytes à frente ou atrás do endereço atual.
  - Sintaxe:
    - B offset12 ; além de realizar o desvio, essa instrução também armazena o endereço da próxima instrução no registrador LR.



Grupo de Engenharia da Computação - CIn / UFPE



## Instruções Thumb

- BX
  - Permite realizar um desvio para um endereço especificado por um registrador qualquer (R0 a R14).
  - Adicionalmente, o estado do bit 0 do registrador define se a CPU decodifica instruções ARM (bit0 = 0) ou Thumb (bit0 = 1).



Grupo de Engenharia da Computação - CIn / UFPE



## Instruções Thumb

- CMP
  - Permite realizar comparação entre dois registradores ou entre um registrador e um operando imediato.
  - Sintaxe:
    - CMP RaL, #operando8
    - CMP RaL, RbL
    - CMP RaL, RbH
    - CMP RaH, RbL
    - CMP RaH, RbH
  - Essa instrução permite operar tanto com registradores baixos quanto com os altos.
  - A comparação é realizada através de uma subtração e somente os flags N,Z,C e V do CPSR são alterados, de forma a indicar o resultado da operação.



Grupo de Engenharia da Computação - CIn / UFPE



## Instruções Thumb

- CMN
  - Também realiza comparação, mas neste caso, somente entre dois registradores baixos (R0 a R7).
  - Sintaxe:
    - CMN Ra, Rb
  - A comparação é realizada através da soma de Ra com Rb e os flags N,Z,C e V do CPSR são alterados de acordo com resultado da operação.



Grupo de Engenharia da Computação - CIn / UFPE



## Instruções Thumb

- TST
  - Permite realizar um teste de bits por meio de uma operação lógica E entre eles.
  - Sintaxe:
    - TST Ra, Rb
  - A instrução irá realizar o E lógico entre os registradores Ra e Rb, setando os flags N, Z, C e V do CPSR de acordo com resultado da operação.



## Instruções Thumb

- SWI
  - A interrupção por software também está disponível no estado Thumb, porém, neste caso, o campo comentário possui um tamanho de somente 8 bits
  - Sintaxe:
    - SWI comentário
  - O campo comentário não é interpretado pela CPU, mas pode ser lido e interpretado pelo software, de forma que, no modo Thumb, até 256 chamadas diferentes da SWI são possíveis.
  - A instrução SWI provoca o desvio para o vetor de modo supervisor (0x00000008), a entrada no modo svc e a mudança para o estado ARM.

