

MC404: Organização de Computadores e Linguagem de Montagem

Lista de Instruções do processador ARM

Prof. Edson Borin

MNE	Operação	Descrição
Transferência de Dados		<MNE>{cond}{S} Rd, <Op2>
MOV	Rd := Op2	Movimentação de registradores
MVN	Rd := (NOT) Op2	Movimentação negada
LDR	Rd := IMEDIADO	Carregamento de um imediato muito grande ou símbolo. LDR Rd, =IMEDIATO
Operações Lógicas e Aritméticas		<MNE>{cond}{S} Rd, Rn, <Op2>
ADD	Rd := Rn + Op2	Adição sem Carry
ADC	Rd := Rn + Op2 + C	Adição com Carry
SUB	Rd := Rn - Op2	Subtração sem Carry
SBC	Rd := Rn - Op2 - (NOT) C	Subtração com Carry
RSB	Rd := Op2 - Rn	Subtração Reversa sem Carry
RSC	Rd := Op2 - Rn - (NOT) C	Subtração Reversa com Carry
AND	Rd := Rn AND Op2	E lógico
EOR	Rd := Rn XOR Op2	OU exclusivo lógico
ORR	Rd := Rn OR Op2	OU lógico
BIC	Rd := Rn AND (NOT)Op2	Bitwise Clear
Multiplicação		<MNE>{cond}{S} Rd, Rm, Rs {, Rn}
MUL	Rd := (Rm x Rs)	Multiplicação
MLA	Rd := Rn + (Rm x Rs)	Multiplicação com acumulação
Comparação		<MNE>{cond} Rn, <Op2>
CMP	Rn - Op2, atualiza <i>flags</i>	Comparação
CMN	Rn + Op2, atualiza <i>flags</i>	Comparação Negativa
TST	Rn AND Op2, atualiza <i>flags</i>	Test (AND)
TEQ	Rn EOR Op2, atualiza <i>flags</i>	Test Equivalence (EOR)
Saltos		<MNE>{cond} <address>
B	PC := <address>	Salta
BL	LR := PC+4, PC := <address>	Salta com <i>link</i>
BX	PC := Rn	Salta para conteúdo do registrador
BLX	LR := PC+4, PC := Rn	Salta para conteúdo do registrador com <i>link</i>
Transferência Entre Memória e Registradores		<MNE>{cond}{size} Rd, <address>
LDR	Rd := Mem[<address>]	Load
STR	Mem[<address>] := Rd	Store
Transferência de Múltiplos Registradores		<MNE>{cond}<mode> Rn{!}, <reglist>
LDM	<reglist> := Mem[<address>]	Load múltiplo
STM	Mem[<address>] := <reglist>	Store múltiplo

Operando {mode}

IA	<i>Increment After</i> : incrementa depois
IB	<i>Increment Before</i> : incrementa antes
DA	<i>Decrement After</i> : decrementa depois
DB	<i>Decrement Before</i> : decrementa antes

Apelidos para operação de pilha

FD	STMTD/LDMFD = STMDB/LDMIA
ED	STMED/LDMED = STMDA/LDMIB
FA	STMFA/LDMFA = STMIB/LMDA
EA	STMEA/LDMEA = STMIA/LMDMB

Operando 0p2

Imm	Constante de 8 <i>bits</i> ou constante de mais <i>bits</i> obtidos através do deslocamento de uma constante de 8 <i>bits</i> .																		
Rn	Valor armazenado no registrador Rn																		
Rn, <deslop> <quant>	Valor armazenado no registrador Rn deslocado <quant> vezes de acordo com o operador de deslocamento <deslop>. <table border="1"> <tr> <td><deslop></td><td>Operador de deslocamento</td></tr> <tr> <td>LSL</td><td>Deslocamento lógico para a esquerda</td></tr> <tr> <td>LSR</td><td>Deslocamento lógico para a direita</td></tr> <tr> <td>ASR</td><td>Deslocamento lógico para a direita</td></tr> <tr> <td>ROR</td><td>Rotação de <i>bits</i> para direita</td></tr> <tr> <td>RRX</td><td>Rotação de <i>bits</i> para direita estendido</td></tr> <tr> <td><quant></td><td>Quantidade de <i>bits</i> a ser deslocada</td></tr> <tr> <td>#Imm</td><td>Constante de 5 <i>bits</i></td></tr> <tr> <td>Rm</td><td>5 <i>bits</i> menos significativos do valor armazenado em Rm</td></tr> </table>	<deslop>	Operador de deslocamento	LSL	Deslocamento lógico para a esquerda	LSR	Deslocamento lógico para a direita	ASR	Deslocamento lógico para a direita	ROR	Rotação de <i>bits</i> para direita	RRX	Rotação de <i>bits</i> para direita estendido	<quant>	Quantidade de <i>bits</i> a ser deslocada	#Imm	Constante de 5 <i>bits</i>	Rm	5 <i>bits</i> menos significativos do valor armazenado em Rm
<deslop>	Operador de deslocamento																		
LSL	Deslocamento lógico para a esquerda																		
LSR	Deslocamento lógico para a direita																		
ASR	Deslocamento lógico para a direita																		
ROR	Rotação de <i>bits</i> para direita																		
RRX	Rotação de <i>bits</i> para direita estendido																		
<quant>	Quantidade de <i>bits</i> a ser deslocada																		
#Imm	Constante de 5 <i>bits</i>																		
Rm	5 <i>bits</i> menos significativos do valor armazenado em Rm																		

Operando <address>

Operando	Indexação	Cálculo do endereço
[Rn, constante]{!}	Pré-indexado	Valor em Rn + constante
[Rn, {-}Rm]{!}	Pré-indexado	Valor em Rn + {-} valor em Rm
[Rn, {-}Rm <deslop> <quant>]{!}	Pré-indexado	Valor em Rn + {-} valor em Rm deslocado <quant> vezes de acordo com o operador de deslocamento <deslop>
[Rn], constante	Pós-indexado	Valor em Rn + constante
[Rn], {-}Rm	Pós-indexado	Valor em Rn + {-} valor em Rm
[Rn], {-}Rm <deslop> <quant>	Pós-indexado	Valor em Rn + {-} valor em Rm deslocado <quant> vezes de acordo com o operador de deslocamento <deslop>

Sufixos {cond}

Sufixo	Condição	Flags
EQ	Igual	Z = 1
NE	Diferente	Z = 0
CS/HS	Carry setada/maior ou igual (sem sinal)	C = 1
CC/LO	Carry limpa/menor (sem sinal)	C = 0
MI	negativo	N = 1
PL	positivo ou zero	N = 0
VS	<i>overflow</i>	V = 1
VC	<i>no overflow</i>	V = 0
HI	maior (sem sinal)	(C = 1) e (Z = 0)
LS	menor ou igual (sem sinal)	(C = 0) ou (Z = 1)
GE	maior ou igual (com sinal)	N = V
LT	menor (com sinal)	N != V
GT	maior (com sinal)	(Z = 0) e (N = V)
LE	menor ou igual (com sinal)	(Z = 1) ou (N != V)
AL	sempre (padrão)	–

Operando {size}

B	<i>Byte</i> : Carrega um <i>byte</i> da memória em um registrador ou armazena o <i>byte</i> menos significativo do registrador na memória.
SB	<i>Signed Byte</i> : Carrega um <i>byte</i> da memória no <i>byte</i> menos significativo do registrador e ajusta os <i>bits</i> restantes de acordo com o sinal do número.
H	<i>Half-Word</i> : Carrega meia palavra (dois <i>bytes</i>) da memória em um registrador ou armazena os 2 <i>bytes</i> menos significativos do registrador na memória.
SH	<i>Signed Half-Word</i> : Carrega meia palavra (dois <i>bytes</i>) da memória no registrador e ajusta os <i>bits</i> restantes de acordo com o sinal do número.

Operando <reglist>

Lista de registradores ou faixa de registradores separados por vírgula.	
Exemplo	Registradores na lista
{R1,R2}	Lista com registradores R1 e R2
{R1-R4}	Lista com registradores R1, R2, R3 e R4
{R1-R3, R7, R11}	Lista com registradores R1, R2, R3, R7 e R11