

EXEMPLO INSTRUÇÃO ALTO NÍVEL PARA BAIXO NÍVEL

Utilizaremos uma (3 Endereços, Registrador-para-Registrador), o Modo de Endereçamento Direto para as instruções de acesso à memória e palavra de 32 bits (4 bytes).

Definições para o Exemplo:

Elemento	Símbolo	Função	Formato
Registradores	R1, R2, R3...	Memória interna de alta velocidade do processador.	
Memória RAM	[Endereço]	Posição de memória (ex: [1000])	
Instruções	LOAD	Copia dado da RAM para um registrador.	LOAD Destino, Endereço
	STORE	Copia dado de um registrador para a RAM.	STORE Fonte, Endereço
	ADD	Soma de registradores.	ADD Destino, Fonte1, Fonte2
	MUL	Multiplicação de registradores.	MUL Destino, Fonte1, Fonte2
	BNE	Desvio Se Não Igual (Branch Not Equal).	BNE Reg1, Reg2, Rótulo
	JMP	Salto incondicional.	JMP Rótulo

EXEMPLO 01: Soma de Três Números ($A = B + C + D$)

Endereçamento Direto: Os valores de B, C e D são carregados da memória. Assumimos: B está em [1000], C em [1004], D em [1008]. O resultado A será armazenado em [1012].

Linha	Código Assembly (3 Operandos)	Operação na ULA / Unidade de Controle
1	LOAD R1, 1000	Busca/Decodificação: UC reconhece LOAD (End. Direto). Execução: UC envia 1000 ao barramento de endereço, solicita leitura da RAM. Dado de [1000] (Valor B) volta e é armazenado em R1.
2	LOAD R2, 1004	Execução: UC carrega dado de [1004] (Valor C) em R2.
3	LOAD R3, 1008	Execução: UC carrega dado de [1008] (Valor D) em R3.
4	ADD R4, R1, R2	Decodificação: UC reconhece ADD. Execução: UC envia R1 e R2 para a ULA, que executa $R1 + R2$. Resultado (B+C) é armazenado em R4.
5	ADD R4, R4, R3	Execução: UC envia R4 e R3 para a ULA, que executa $R4 + R3$. Resultado final (B+C+D) é armazenado em R4.
6	STORE R4, 1012	Execução: UC envia R4 (Valor A) ao barramento de dados e 1012 ao barramento de endereço. Solicita escrita na RAM.

EXEMPLO 02: Multiplicação de Dois Números ($Z = X * Y$)

Endereçamento Direto: X em [2000], Y em [2004]. Resultado Z em [2008].

Linha	Código Assembly (3 Operandos)	Operação na ULA / Unidade de Controle
1	LOAD R1, 2000	Execução: $R1 \leftarrow$ Valor de X na RAM.
2	LOAD R2, 2004	Execução: $R2 \leftarrow$ Valor de Y na RAM.
3	MUL R3, R1, R2	Decodificação: UC reconhece MUL. Execução: UC envia R1 e R2 para a ULA. A ULA ativa o circuito multiplicador: $R1 * R2$. Resultado ($X * Y$) armazenado em R3.
4	STORE R3, 2008	Execução: UC armazena R3 (Valor Z) na RAM, endereço [2008].

EXEMPLO 03: Estrutura de Decisão (IF-ELSE)

Linha	Rótulo	Código Assembly (3 Operandos)	Ação da Unidade de Controle (UC)
1		BNE R1, 10, ELSE	Decodificação: UC reconhece BNE. Execução: UC/ULA compara <i>R1</i> com o valor imediato <i>10</i> . Se NÃO forem iguais (condição FALSA), a UC muda o PC (Contador de Programa) para o endereço de ELSE (salta a próxima instrução). Se for igual, avança.
2		ADD R3, R2, R0	Execução (Bloco IF): Se $R1=10$, $R3 \leftarrow R2 + 0$ (Copia R2 para R3).
3		JMP FIM	Decodificação: UC reconhece JMP. Execução: UC muda o PC para o endereço de FIM (salta o bloco ELSE).
4	ELSE	ADD R3, R0, R0	Execução (Bloco ELSE): Se $R1 \neq 10$, $R3 \leftarrow 0 + 0$ (Zera R3).
5	FIM	... próxima instrução ...	Sequência normal de execução.

Código Fonte: IF ($R1 == 10$) THEN ($R3 = R2$) ELSE ($R3 = 0$)

EXEMPLO 04: Salto Incondicional (GOTO)

Código Fonte: ... Linhas ... GOTO LOOP ... Linhas ...

Linha	Rótulo	Código Assembly (3 Operandos)	Ação da Unidade de Controle (UC)
1		...	UC executa normalmente.
2		JMP LOOP	Decodificação: UC reconhece JMP. Execução: UC ignora as instruções restantes e atualiza o PC (Contador de Programa) para o endereço da instrução marcada por LOOP .
3		... (Instruções ignoradas) ...	
4	LOOP	ADD R1, R2, R3	O processador continua executando a partir daqui.

EXEMPLO 05: Laço de Repetição (FOR)

Código Fonte: FOR (i=0; i < 10; i++) { R1 = R1 + R2; }

Assumimos que *R9* é a variável de contador *i*.

Linha	Rótulo	Código Assembly (3 Operandos)	Ação da Unidade de Controle (UC)
1		ADD R9, R0, R0	Inicialização: $R9 \leftarrow 0 + 0$ (Inicializa $i=0$).
2	LOOP	BNE R9, 10, FIM	Teste de Condição: UC/ULA compara <i>R9</i> com 10. Se NÃO for igual (ou seja, $i < 10$ é FALSO, ou $i=10$), UC muda o PC para FIM (sai do FOR). Se for diferente, avança.
3		ADD R1, R1, R2	Corpo do Laço (Execução): UC/ULA executa a soma $R1 \leftarrow R1 + R2$.
4		ADD R9, R9, 1	Incremento: UC/ULA incrementa o contador $R9 \leftarrow R9 + 1$.
5		JMP LOOP	Salto: UC muda o PC de volta para o endereço de LOOP (repetição do ciclo).
6	FIM	... próxima instrução ...	O processador continua a execução após o laço.