Computador Neander

Prof. Alexandre Beletti

Cap. 4 – Raul Weber

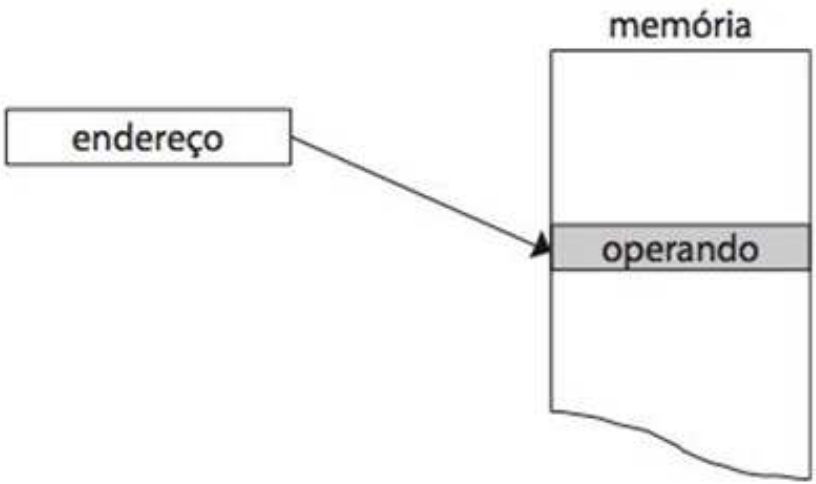
Características

- Largura de dados e endereços de 8 bits
- Dados representados em complemento dois
- Um acumulador de 8 bits (AC)
- Um apontador de programa de 8 bits (PC)
- Um registrador de estado com 2 códigos de condição: negativo (N) e zero (Z)

Endereçamento

- Possui somente um modo: direto ou absoluto
- A palavra que segue o código da instrução contém, nas instruções de manipulação de dados, o endereço da memória do operando
- Nas instruções de desvio, o endereço contido na instrução corresponde à posição de memória onde está uma instrução a ser executada

Modo de endereçamento direto



Conjunto (SET) de Instruções

código	instrução		comentário
0000	NOP		nenhuma operação
0001	STA	end	armazena acumulador – (store)
0010	LDA	end	carrega acumulador – (load)
0011	ADD	end	soma
0100	OR	end	"ou" lógico
0101	AND	end	"e" lógico
0110	NOT		inverte (complementa) acumulador
1000	JMP	end	desvio incondicional – (jump)
1001	JN	end	desvio condicional – (jump on negative)
1010	JZ	end	desvio condicional – (jump on zero)
1111	HLT		término de execução – (halt)

Operandos - END

- End: significa endereço direto
- Nas instruções STA, LDA, ADD, OR e AND corresponde ao endereço do operando
- Nas instruções JMP, JN e JZ corresponde ao endereço de desvio

Ações Executadas

instrução		comentário
NOP		nenhuma operação
STA	end	MEM(end) ← AC
LDA	end	AC← MEM(end)
ADD	end	AC← MEM(end) + AC
OR	end	AC← MEM(end) OR AC
AND	end	AC← MEM(end) AND AC
NOT		AC← NOT AC
JMP	end	PC← end
JN	end	IF N=1 THEN PC ← end
JZ	end	IF Z=1 THEN PC ← end

Explicações Gerais

- AC é o acumulador
- MEM (end) significa o conteúdo da posição "end" de memória
- N e Z são códigos de condição
- "<-" significa atribuição

Códigos de Condição

- N(negativo): sinal do resultado
  - 1 – resultado negativo
  - 0 – resultado positivo (ou não negativo, pois consideramos 0 como sendo positivo)
- Z(zero): indica resultado igual a zero
  - 1 – resultado é igual a zero
  - 0 – resultado é diferente de zero

Instruções que afetam flags N e Z

- ADD
- NOT
- AND
- OR
- LDA (instrução de transferência)

Formato das Instruções

- Formadas por 1 ou 2 bytes

Instruções de 1 e 2 bytes

- Instruções de 1 byte: os 4 bits mais significativos contém o código da instrução
- Instruções de 2 bytes: o primeiro byte contém o código (também nos 4 bits mais significativos) e o segundo byte contém um endereço. São as instruções que fazem referência a memória.

Exemplo 1

- Soma de três posições consecutivas de memória
- Armazena o resultado em uma quarta posição
- Escolha a área de alocação de valores e a área do programa na RAM

Área de Programa e Dados

área de programa		
início do programa	posição 0	(0H)
área de dados		
primeira parcela	posição 128	(80H)
segunda parcela	posição 129	(81H)
terceira parcela	posição 130	(82H)
resultado	posição 131	(83H)

Exemplo - Assembly

- LDA 128 (acum. recebe conteúdo da posição 128)
- ADD 129 (soma 129 ao conteúdo do acumulador)
- ADD 130 (soma 130 ao conteúdo do acumulador)
- STA 31 (conteúdo de acum. copiado para end. 31)
- HLT (processador para)

Exemplo - Opcode

INSTRUÇÃO	OPCODES	
LDA 128	20	80
ADD 129	30	81
ADD 130	30	82
STA 131	30	83
HLT	F0	

Exercícios

- Exercício 1
- Exercício 2
- Exercício 3
- Exercício 4