

CESAR

(As bases da civilização atual)

Cesar - características

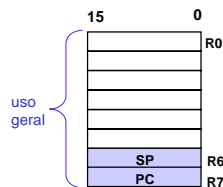
- largura de dados e endereços de 16 bits
- dados representados em complemento de dois
- 8 registradores de uso geral de 16 bits
 - R0 - R7
- 1 registrador de estado
 - 4 códigos de condição
 - negativo (N), zero (Z), carry (C) e overflow (V)
- 8 modos de endereçamento
- instruções com 2, 1 e zero operandos
- suporte para pilha (stack)



2

Registradores

- 8 registradores (R0 - R7)
 - R7 = program counter (PC)
 - R6 = stack pointer (SP)
 - R7 e R6 tem funções específicas,
 - mas também são de uso geral
- três bits para seleção de registrador na instrução



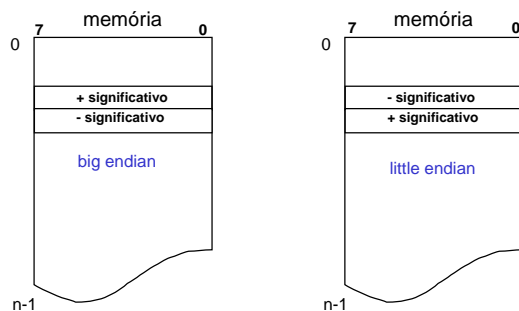
3

Endereçamento de memória

- arquitetura de 16 bits
- memória organizada em bytes (8 bits)
 - Big endian - byte mais significativo armazenado no endereço menor
 - PDP, Motorola, Cesar
 - Little endian - byte menos significativo armazenado no endereço menor (Intel)
- alinhamento do operando
 - Cesar - sem restrições no endereçamento
 - pode começar em endereço par ou ímpar

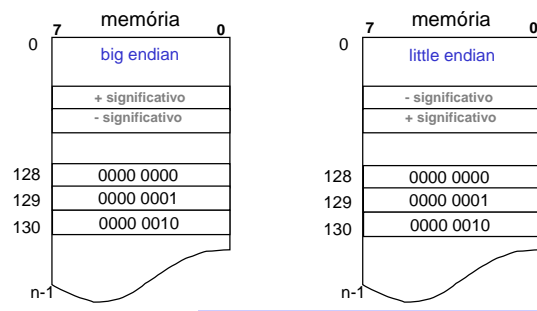
4

Big versus little endian



5

Exemplo



mem(128) = 0; mem(129) = 1; mem(130) = 2

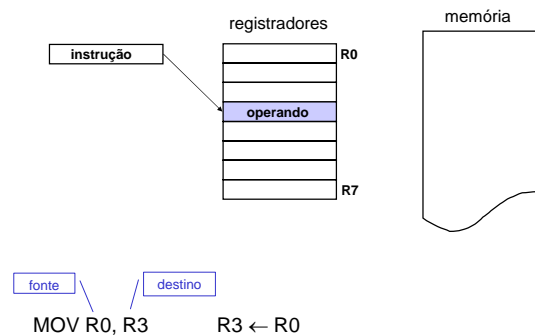
6

Modos de endereçamento

- 8 modos de endereçamento
 - 000 = registrador
 - 001 = registrador pós-incrementado
 - 010 = registrador pré-decrementado
 - 011 = indexado
 - 100 = registrador indireto
 - 101 = pós-incrementado indireto
 - 110 = pré-decrementado indireto
 - 111 = indexado indireto

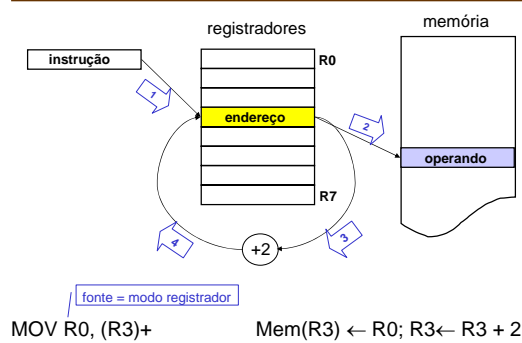
7

Modo Registrador



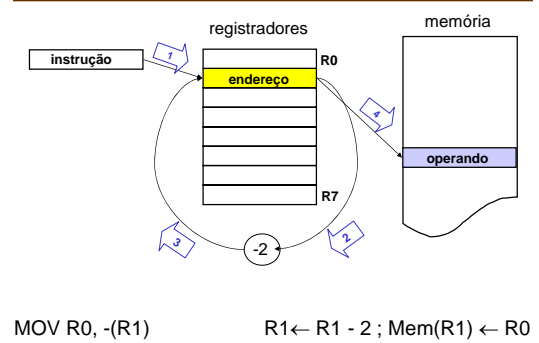
8

Pós-incrementado



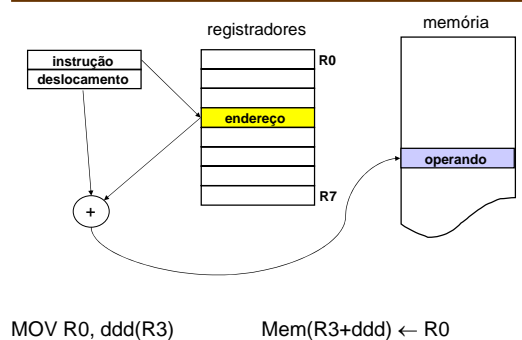
9

Pré-decrementado



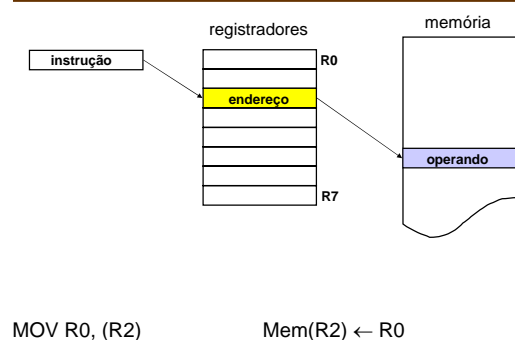
10

Indexado



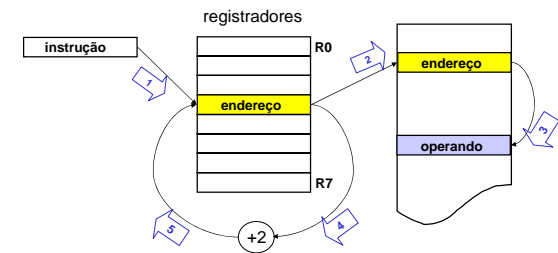
11

Registrador indireto



12

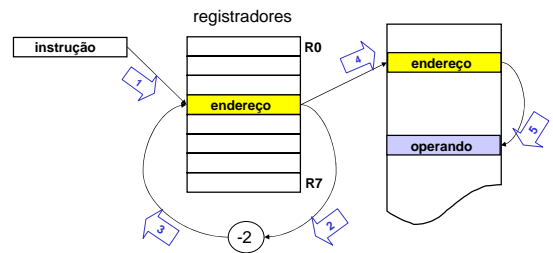
Pós-incrementado indireto



MOV R0, ((R3)+) $\text{Mem}(\text{Mem}(\text{R3})) \leftarrow \text{R0}; \text{R3} \leftarrow \text{R3} + 2$

13

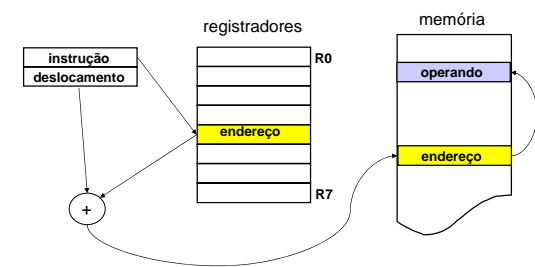
Pré-decrementado indireto



MOV R0, (-(R3)) $\text{R3} \leftarrow \text{R3} - 2; \text{Mem}(\text{Mem}(\text{R3})) \leftarrow \text{R0}$

14

Indexado indireto



MOV R0, (ddd(R4)) $\text{Mem}(\text{Mem}(\text{R4} + \text{ddd})) \leftarrow \text{R0}$

15

codificação simbólica

	Modo	Símbolo
000	registrador	Rx
001	registrador pós-incrementado	(Rx)+
010	registrador pré-decrementado	-(Rx)
011	indexado	ddd(Rx)
100	registrador indireto	(Rx)
101	pós-incrementado indireto	((Rx)+)
110	pré-decrementado indireto	(-(Rx))
111	indexado indireto	(ddd(Rx))

16

Exemplos

Modo registrador apenas um operando

CLR R0 $\text{R0} \leftarrow 0$

Modo registrador indireto

CLR (R0) $\text{Mem}(\text{R0}) \leftarrow 0$

Modo indexado

CLR ddd(R0) $\text{Mem}(\text{R0} + \text{ddd}) \leftarrow 0;$

Modo pós-incrementado

CLR (R0)+ $\text{Mem}(\text{R0}) \leftarrow 0; \text{R0} \leftarrow \text{R0} + 2$

17

Exemplos

Modo pré-decrementado apenas um operando

CLR -(R0) $\text{R0} \leftarrow \text{R0} - 2; \text{Mem}(\text{R0}) \leftarrow 0$

Modo indexado indireto

CLR (ddd(R0)) $\text{Mem}(\text{Mem}(\text{R0} + \text{ddd})) \leftarrow 0$

Modo pós-incrementado indireto

CLR ((R0)+) $\text{Mem}(\text{Mem}(\text{R0})) \leftarrow 0; \text{R0} \leftarrow \text{R0} + 2$

Modo pré-decrementado indireto

CLR (-(R0)) $\text{R0} \leftarrow \text{R0} - 2; \text{Mem}(\text{Mem}(\text{R0})) \leftarrow 0$

18

Instruções com dois operandos

- Cesar permite qualquer modo de endereçamento para cada um dos 2 operandos
 - primeiro** é calculado o modo de endereçamento do fonte;
 - depois**, o modo de endereçamento do destino

MOV R0, R5	$R5 \leftarrow R0$
MOV (R0), 47(R5)	$Mem(R5+47) \leftarrow Mem(R0)$
MOV (R0)+, (R0)	temp $\leftarrow Mem(R0)$; $R0 \leftarrow R0 + 2$ $Mem(R0) \leftarrow temp$

19

Endereçamento com PC (R7)

modo registrador

MOV R0, R7 $R7 \leftarrow R0$
 move um novo valor para o PC;
 equivale a um jump incondicional

MOV R7, R0 $R0 \leftarrow R7$
 move para R0 o valor do PC

20

Endereçamento com PC (R7)

modo pós-incrementado

MOV R0, (R7)+ $Mem(R7) \leftarrow R0$; $R7 \leftarrow R7 + 2$
 move conteúdo do R0 para próxima posição
 no programa e pula essa posição

MOV (R7)+, R0 $R0 \leftarrow Mem(R7)$; $R7 \leftarrow R7 + 2$
 modo imediato

21

Endereçamento com PC (R7)

modo pré-decrementado

MOV R0, -(R7) $R7 \leftarrow R7 - 2$; $Mem(R7) \leftarrow R0$
 move conteúdo do R0 para instrução anterior
 e pula para instrução anterior

MOV -(R7), R0 $R7 \leftarrow R7 - 2$; $R0 \leftarrow Mem(R7)$
 move instrução anterior para R0 e pula para
 instrução anterior

22

Endereçamento com PC (R7)

modo indexado

MOV R0, ddd(R7) $Mem(R7+ddd) \leftarrow R0$
 move conteúdo do R0 para área de
 memória apontada pelo PC mais
 deslocamento (relativo ao PC)

MOV ddd(R7), R0 $R0 \leftarrow Mem(R7+ddd)$
 move para R0 conteúdo da área de
 memória apontada pelo PC mais
 deslocamento (relativo ao PC)

23

Endereçamento com PC (R7)

modo registrador indireto

MOV R0, (R7) $Mem(R7) \leftarrow R0$
 move conteúdo do R0 para área de memória
 apontada pelo PC (conteúdo do R0 vai ser a
 próxima instrução a ser executada)

MOV (R7), R0 $R0 \leftarrow Mem(R7)$
 move para R0 o conteúdo da área de
 memória apontada pelo PC
 (move próxima instrução para o R0)

24

Endereçamento com PC (R7)

pós-incrementado indireto

MOV R0, ((R7)+) $\text{Mem}(\text{Mem}(\text{R7})) \leftarrow \text{R0}; \text{R7} \leftarrow \text{R7} + 2$

posição que segue instrução contém endereço de operando destino (modo direto)

MOV ((R7)+), R0 $\text{R0} \leftarrow \text{Mem}(\text{Mem}(\text{R7})); \text{R7} \leftarrow \text{R7} + 2$

posição que segue instrução contém endereço de operando fonte (modo direto)

25

Endereçamento com PC (R7)

pré-decrementado indireto

MOV R0, -(R7) $\text{R7} \leftarrow \text{R7} - 2; \text{Mem}(\text{Mem}(\text{R7})) \leftarrow \text{R0}$

retorna para instrução anterior e usa instrução como endereço do operando destino

MOV -(R7), R0 $\text{R7} \leftarrow \text{R7} - 2; \text{R0} \leftarrow \text{Mem}(\text{Mem}(\text{R7}))$

retorna para instrução anterior e usa instrução como endereço do operando fonte

26

Endereçamento com PC (R7)

indexado indireto

MOV R0, (ddd(R7)) $\text{Mem}(\text{Mem}(\text{R7}+\text{ddd})) \leftarrow \text{R0}$

move conteúdo do R0 para posição de memória apontada por endereço contido em posição de memória apontada pelo PC mais deslocamento (relativo indireto)

MOV (ddd(R7)), R0 $\text{R0} \leftarrow \text{Mem}(\text{Mem}(\text{R7}+\text{ddd}))$

move para R0 conteúdo da posição de memória apontada por endereço contido em posição de memória apontada pelo PC mais deslocamento (relativo indireto)

27

Codificação simbólica com R7

Modo	Nome	Símbolos
001	imediato	#nnn ou (R7)+ nnn
101	absoluto	nnn ou ((R7)+) nnn
011	relativo	ddd(R7)

28

Endereçamento com R7

MOV R0, R7	$\text{R7} \leftarrow \text{R0}$	Útil Cuidado ! Não usar !!
MOV R7, R0	$\text{R0} \leftarrow \text{R7}$	
MOV R0, (R7)+	$\text{Mem}(\text{R7}) \leftarrow \text{R0}; \text{R7} \leftarrow \text{R7} + 2$	
MOV (R7)+, R0	$\text{R0} \leftarrow \text{Mem}(\text{R7}); \text{R7} \leftarrow \text{R7} + 2$	imediato
MOV R0, -(R7)	$\text{R7} \leftarrow \text{R7} - 2; \text{Mem}(\text{R7}) \leftarrow \text{R0}$	
MOV -(R7), R0	$\text{R7} \leftarrow \text{R7} - 2; \text{R0} \leftarrow \text{Mem}(\text{R7})$	
MOV R0, ddd(R7)	$\text{Mem}(\text{R7}+\text{ddd}) \leftarrow \text{R0}$	
MOV ddd(R7), R0	$\text{R0} \leftarrow \text{Mem}(\text{R7}+\text{ddd})$	relativo
MOV R0, (R7)	$\text{Mem}(\text{R7}) \leftarrow \text{R0}$	
MOV (R7), R0	$\text{R0} \leftarrow \text{Mem}(\text{R7})$	
MOV R0, ((R7)+)	$\text{Mem}(\text{Mem}(\text{R7})) \leftarrow \text{R0}; \text{R7} \leftarrow \text{R7} + 2$	
MOV ((R7)+), R0	$\text{R0} \leftarrow \text{Mem}(\text{Mem}(\text{R7})); \text{R7} \leftarrow \text{R7} + 2$	direto
MOV R0, -(R7))	$\text{R7} \leftarrow \text{R7} - 2; \text{Mem}(\text{Mem}(\text{R7})) \leftarrow \text{R0}$	
MOV -(R7)), R0	$\text{R7} \leftarrow \text{R7} - 2; \text{R0} \leftarrow \text{Mem}(\text{Mem}(\text{R7}))$	
MOV R0, ddd((R7))	$\text{Mem}(\text{Mem}(\text{R7}+\text{ddd})) \leftarrow \text{R0}$	
MOV ddd((R7)), R0	$\text{R0} \leftarrow \text{Mem}(\text{Mem}(\text{R7}+\text{ddd}))$	relativo ind.

29