

PC = Endereço inicial / contador (RIP)

ByteDeExecução:

Cauc¹ - 43558

9.7

while (ByteDeExecução):

R = Memória [PC]

PC = PC + 1

DeterminaTipoInstrução (R, TipoDaInstrução)

C 1P
1

if (DadoRequerido):

dado = Memória [LocalizaçãoDado]

Execute (TipoDeInstrução, dado, memória, AC, PC, BitDaInstrução)

2) 0705₁₆

C 1P

3) a) EF

C

b) CDEF

c) 89ABCDEF

d) 0123456789ABCDEF

4) No máximo, cada registrando alongo de 1 byte a 8 bits a 8 bytes a 64 bits em endereço, dependendo do tipo do variável ou registro.

C P

~~4) Faça um desenho do layout~~

5) Dado o hexadecimal de 32 bits 004C4B4016, qual o σ :

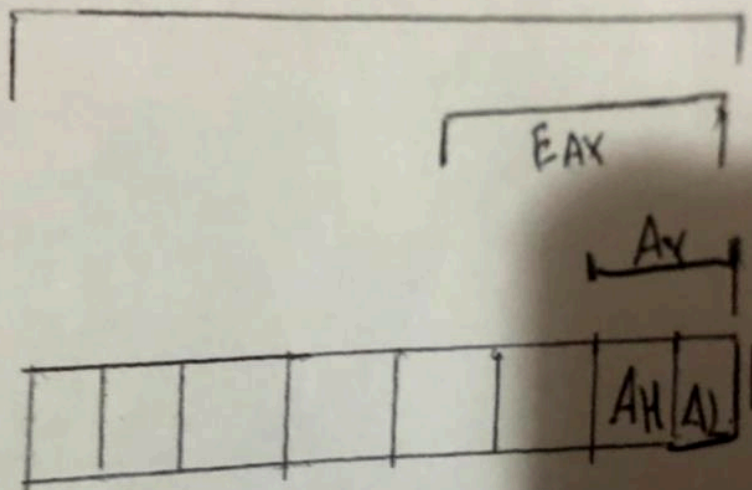
a) 0 .

b) 0 .

C 1P

5) Faça um desenho do layout do registrador acumulador.

RAX



6)

a) al \rightarrow 8

b) ecx \rightarrow 64

c) ebx \rightarrow 16

d) edx \rightarrow 32

CIP

8)

a) MOV R, Var carrega o endereço de Var em R ou conteúdo de Var no caso de MOV R, [Var]

LEA - Carrega o endereço efetivo + deslocamento, se o deslocamento for 0, não é necessário adicionar.

Assim, LEA R, [VAR] faz o mesmo que MOV R, Var. Contudo, a instrução MOV R, Var + deslocamento não é aceita pois isso tem que usar LEA R, [Var + deslocamento]

b)

section .text : seção de código, onde o programa é feito

section .data : seção onde se declaram as variáveis inicializadas

section .bss : seção de variáveis não inicializadas

variáveis não inicializadas

- c) RAX : Registrador conhecido como acumulador geralmente os resultados vão para RAX
- RBX : Registrador que pode ser usado índice de base
- RCX : registrador geralmente usado para contagem
- RDX : Registrador que serve de auxiliar em algumas operações de multiplicação/divisão
- RIP : registrador contendo o endereço de programa, guarda o endereço do próximo de instruções
- RSI : registrador de índice e origem
- RDI : registrador de índice e destino
- RSI : registrador que aponta para o topo da pilha de memória

RSI : registrador de índice e origem

RDI : registrador de índice e destino

RSP : registrador que aponta para o topo da pilha de memória

RBP : registrador de base da pilha

OF : flag overflow flag

DF : flag de manipulação de dados

IF : flag de interrupção

CF : flag de carry (borrow)

ZF : flag para resultado zero

PF : flag de resultado par

SF : flag de signo

d) EIP

9) a) $0000 \quad 0000 \quad 0111 \quad 1011_2 = 000153_8 = 007B_{16}$

b) $0000 \quad 0000 \quad 1011 \quad 1111_2 = 191_{10} = 00BF_{16}$

c) $137_8 = 95_{10} = 005F_{16}$ C. 017

d) $10101111_2 \quad 1010 \quad 1101_2 = 127655_8 = -20563_{10}$

10)

$$a) = 100 \quad 111 \quad 111$$

$$b) = 001 \quad 000 \quad 000 \quad 010 \quad 001$$

$$c) = 100 \quad 111 \quad 110 \quad 100$$

$$d) = 100 \quad 111 \quad 111$$

C 19