

1. // Passos para que um computador execute as instruções

PC = Endereço Inicial // contador de programa (RIP)

Byte De Execução = Teme

while (Byte De Execução) :

 R = Memória [PC]

 PC = PC + 1

 DeterminaTipoInstrução (iR, TipoDaInstrução).

 AchaDados (TipoDaInstrução, iR, LocalizaçãoDados, DadosRequerido)

 If (DadosRequerido) :

 dados = Memória [LocalizaçãoDados]

 Execute (TipoDeInstrução, dados, memória, AC, PC, Bit De Instrução)

9,7

2. 0705₁₆

3. a) EF

 b) CDEF

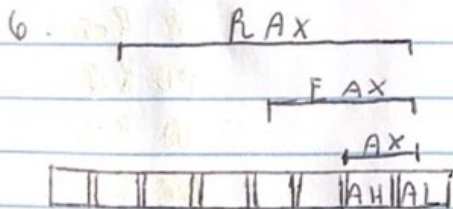
 c) 89ABCDEF

 d) 0123456789ABCDEF

4. O limite máximo, cada registrador tem um alcance de 1 byte / 8 bits a 8 bytes / 64 bits por endereço. Dependendo de como é o tipo da variável ou registro.

5. a) 0

 b) 0



7. a) 8 c) 16

 b) 64 d) 32

8. a) MOV R, Var carrega o endereço de Var em R ou o 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 intuição MOV R, Var + deslocamento não é aceita. Para isso temos 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

c) RAX: registrador conhecido como acumulador, geralmente os resultados são para RAX

R BX: registrador que pode ser usado índice de base

R CX: registrador geralmente usado para contagem

R DX: registrador que serve de auxiliar em algumas operações de multiplicação/divisão.

RIP: registrador contador de programa, guarda o endereço da linha de instrução.

R SI: registrador de índice e origem

R DI: registrador de índice e destino

R SP: registrador que aponta para o topo da pilha de memória

R BP: registrador de base da pilha

OF: flag overflow flag

DF: flag de manipulação de dados

IF: flag de interrupção

TF: flag de burca (trap)

ZF: flag para resultado zero

PF: flag de resultado par

CF: flag de vai em.

d) RIP

9. a) 0000 0000 0111 1011₂ = 00073₈ = 007B₁₆

b) 0000 0000 1011 1111₂ = 191₁₀ = 00BF₁₆

c) 137₈ = 95₁₀ = 005F₁₆

d) 1010 1111 1010 1101₂ = 127655₈ = 2050B₁₆

10. a) 110 111 100 000

b) 001 000 000 010 001

c) 100 111 110 100

d) 100 111 111