

Bruno Biagi - 43553

9,7

D S T Q Q S S  
D L M M J V S

PC = Endereço inicial // contador de programas

Byte de execução = True

while (Byte de execução):

R = Memória [PC]

PC = PC + 1

Determina tipo de instrução (IR, Tipo de instrução)

Endereço (Tipo de instrução, IR, Localização de dados, Endereço de destino)

If (Endereço)

Endereço = Memória [Endereço de destino]

Executa (Tipo de instrução, dados, memória, AC, BH de instrução)

② 0705<sub>16</sub> C IP

③

ⓐ EF

ⓑ CDEF

ⓒ 89ABCDEF C IP

ⓓ 0123456789ABCDEF

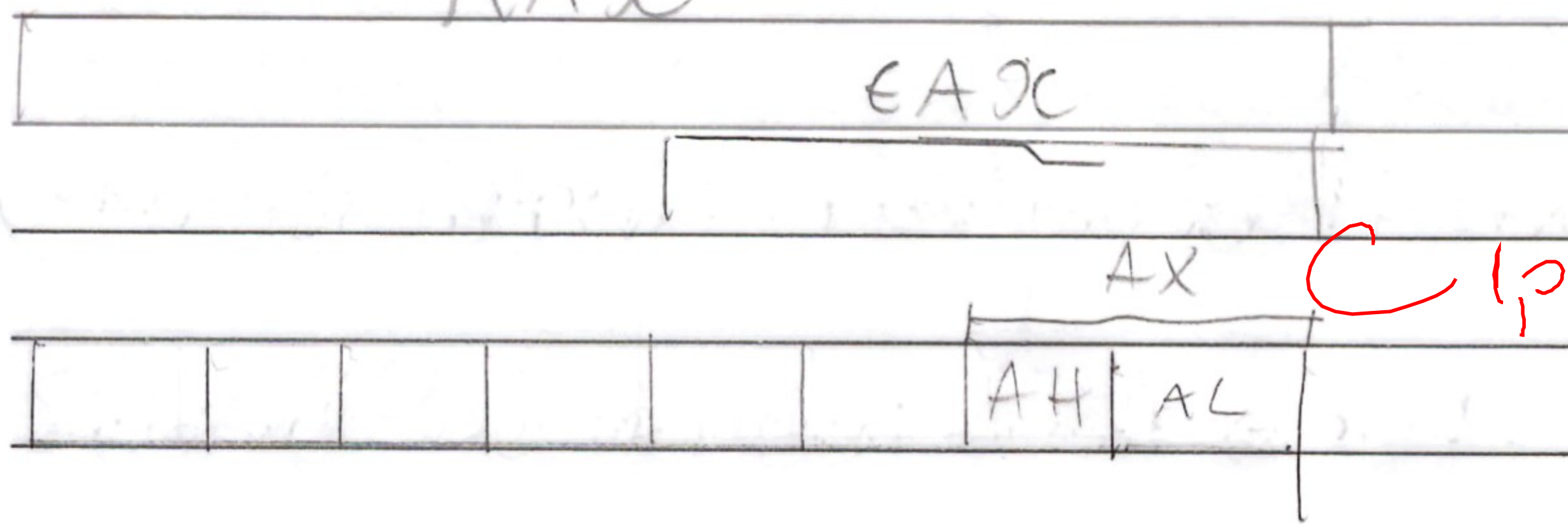
④ Na máxima, cada registrador alça de 1 byte ou 8 bits a 8 bytes ou 64 bits por endereço. Depende do tipo do variável ou registro C IP

500

20

5

RAX



608

64 supondo que isso seja 4

CIP

16

32

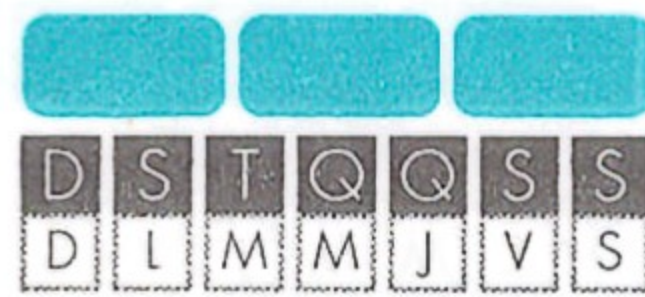
8

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 o suffixo, LEA R, [Var] faz o mesmo que MOV R, Var. Contudo, a instrução MOV R, [Var + deslocamento] não é aceita. Para isso temos que usar LEA R, [Var + deslocamento]

U

- 
- 
-



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

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 de programa, guarda o endereço do linha de instrução

Rsi: registrador de índice e origem

Rdi: registrador de índice e destino

Rsp: registrador que aponta para o topo do pilha de memória

Rbp: registrador de base do pilha

flags

OF: flag overflow flag

DF: flag de manipulação de dados

IF: flag de interrupção

TF: flag de trap

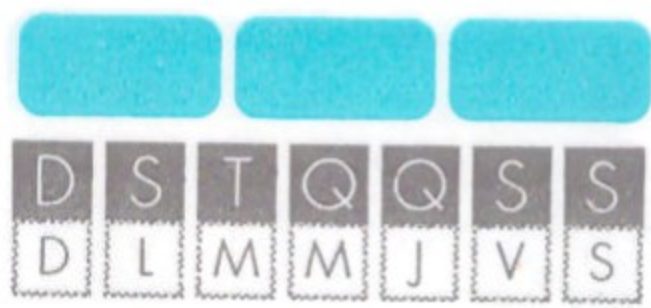
ZF: flag para resultado zero

PF: flag de resultado par

CF: flag de carry

Rip





9) @ 0000 0000 0111 1011, ~~000538~~ 007B<sub>16</sub>

Ⓐ 0000 0000 1011 1111<sub>2</sub> 191<sub>10</sub> 00BF<sub>16</sub>

Ⓑ 437<sub>8</sub> 95<sub>10</sub> 005F<sub>16</sub> C 0.7

Ⓐ 1010 1111 1010 1101<sub>2</sub> 127655<sub>8</sub> 420563<sub>10</sub>

10)

Ⓐ 0000 0000 1101 1110 0000<sub>2</sub>

Ⓑ 0000 0000 0000 0001 0001<sub>2</sub>

Ⓐ 0000 0000 1001 1111 0100<sub>2</sub>

Ⓐ 0000 0000 0001 0011 1111

