MODELO DE COMPUTADOR

PRINCIPAIS ELEMENTOS:

CPU

BUS (canal de comunicação)

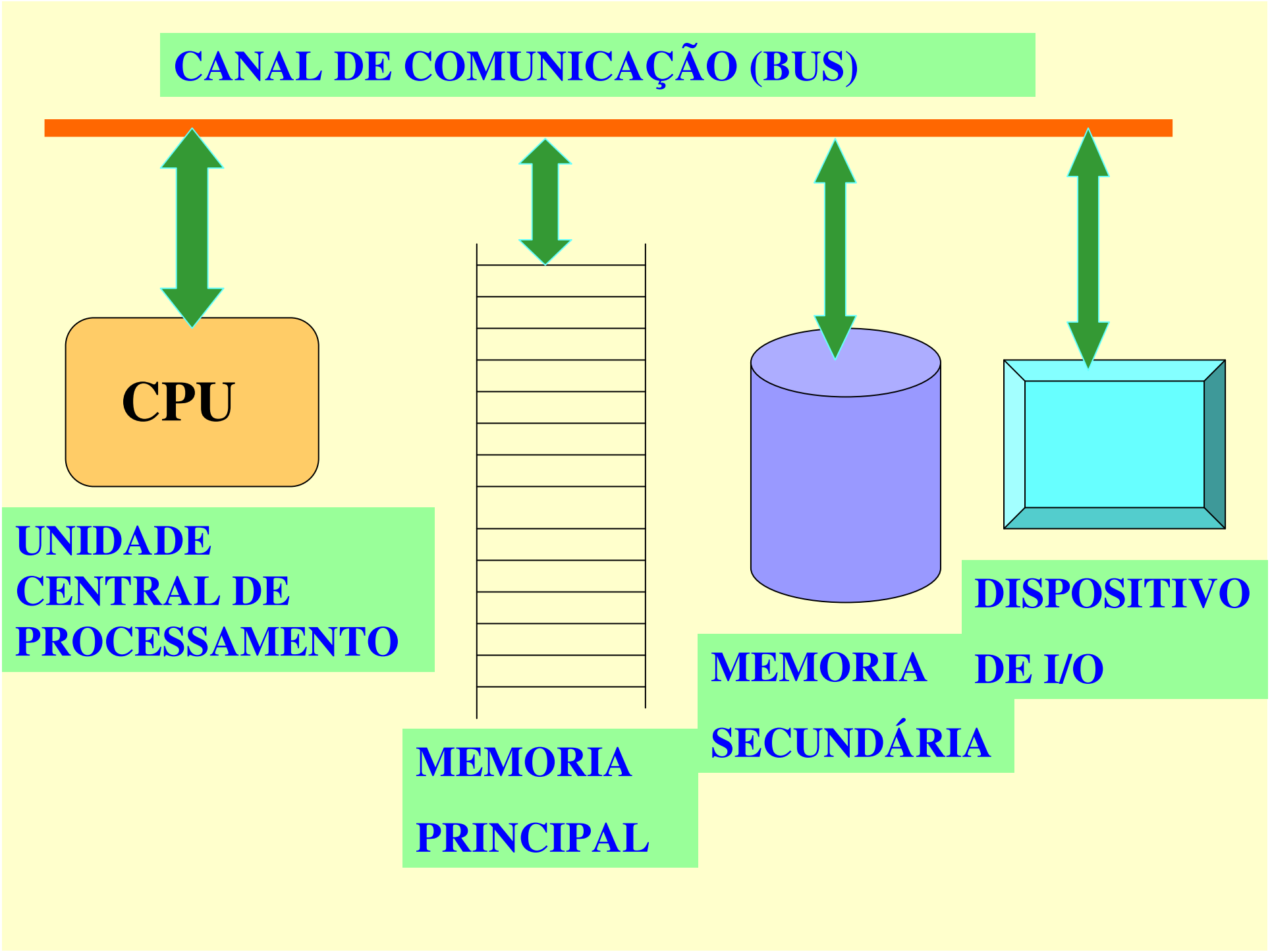
MEMÓRIA PRINCIPAL

(acesso randômico e armazenamento volátil)

MEMÓRIA SECUNDÁRIA

(meio magnético: HD, não volátil, custo e velocidade de acesso baixos)

DISPOSITIVOS DE ENTRADA E SAÍDA


CANAL DE COMUNICAÇÃO (BUS)
CPU
UNIDADE CENTRAL DE PROCESSAMENTO
MEMORIA PRINCIPAL
MEMORIA SECUNDÁRIA
DISPOSITIVO DE I/O

UM COMPUTADOR HIPOTÉTICO

CPU POSSUI UMA ÁREA DE MEMORIA INTERNA CAPAZ DE ARMAZENAR, TEMPORARIAMENTE, O RESULTADO DE UMA OPERAÇÃO (REGISTRADOR)

CADA POSIÇÃO DE MEMÓRIA PODE ARMAZENAR VALORES NUMÉRICOS E É ROTULADA POR UM NÚMERO INTEIRO (0,1, ,N)

PROGRAMA: SEQUÊNCIA DE INTRUÇÕES ARMAZENADA NA MEMORIA

INSTRUÇÕES DISPONÍVEIS

(MNEMÔNICOS, ASSEMBLY):

READ pos : captura o valor fornecido via teclado e o armazena na posição “pos”;

WRITE pos : escreve o valor armazenado na posição de memória rotulada por “pos” na tela do computador;

STORECONST num pos :armazena o valor da constante numérica especificada por “num” na posição de memória rotulada por “pos”;

ADD pos1 pos2 : calcula a soma dos dois operandos que são os valores numéricos armazenados nas posições “pos1” e “pos2”.

O resultado é armazenado no REGISTRADOR.

SUB pos1 pos2 : calcula a diferença entre o primeiro e segundo operando, o resultado é armazenado no REGISTRADOR.

MULT pos1 pos2 : multiplica o primeiro e segundo operando, o resultado é armazenado no REGISTRADOR.

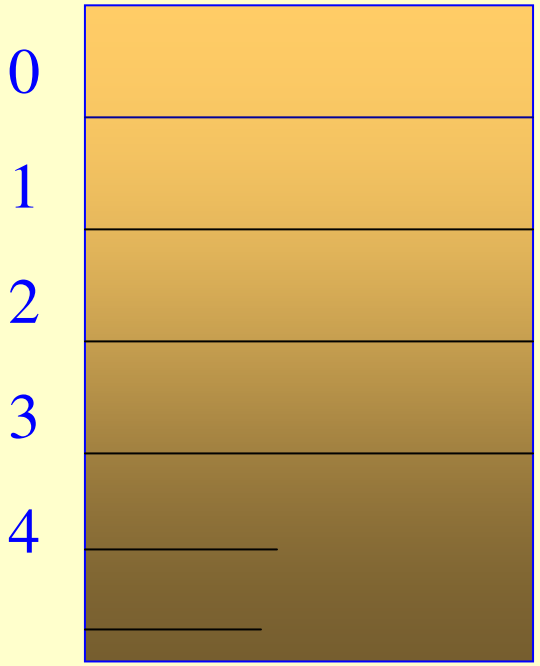
DIV pos1 pos2 : divide o primeiro pelo segundo operando, o resultado é armazenado no REGISTRADOR.

STORE pos : armazena o valor do registrador na posição de memória rotulada por “pos”.

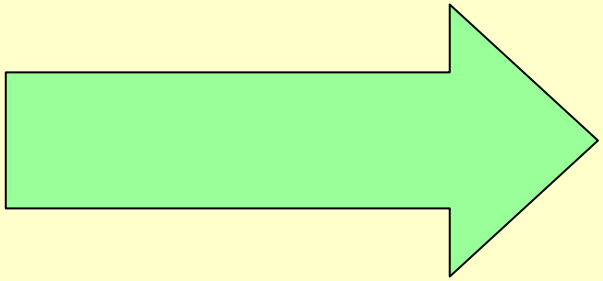
PROGRAMA EM LINGUAGEM DE

ALTO NÍVEL:

```
INT   X ;

READ ( X );

X = X + 1;

WRITE ( X );
```



COMPILADOR



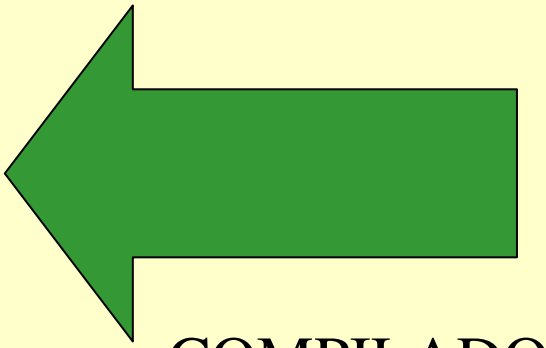
VAR	POS
X	0
C01	1

PROGRAMA EM LINGUAGEM ASSEMBLY

GERADO PELO COMPILADOR

ASSEMBLY É UMA LINGUAGEM DE PROGRAMAÇÃO?

```
INT   X ;
READ ( X );
X = X + 1;
WRITE ( X );
```



COMPILADOR

```
READ X
STORECONST  1 C01
ADD X C01
STORE X
WRITE X
```

VAR	POS
X	0
C01	1

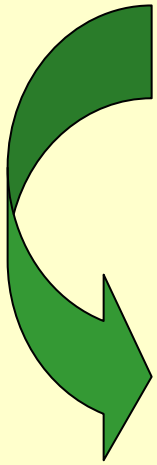
GERAÇÃO DE CÓDIGO:

- **MNEMÔNICOS SUBSTITUIDOS PELOS CÓDIGOS DAS RESPECTIVAS INSTRUÇÕES**
- **NOME DAS VARIÁVEIS E CONSTANTES SUBSTITUIDOS PELOS RESPECTIVOS ENDEREÇOS DE MEMÓRIA**

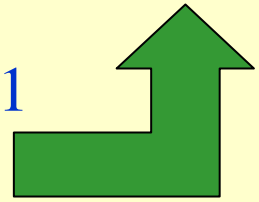
NO NOSSO COMPUTADOR HIPOTÉTICO VAMOS SUPOR OS SEGUINTES CÓDIGOS :

INSTRUÇÃO	CÓDIGO
READ	01
WRITE	02
STORECONST	03
ADD	04
SUB	05
MUL	06
DIV	07
STORE	08

```
INT   X ;
READ ( X );
X = X + 1;
WRITE ( X );
```



```
READ X
STORECONST  1
C01
ADD X C01
STORE X
WRITE X
```



```
01 0
03 1 1
04 0 1
08 0
02 0
```

VAR	POS
X	0
C01	1

VOLTEMOS AOS MNEMÔNICOS MAS COM POSIÇÕES REAIS DE MEMÓRIA:

```
READ   0
STORECONST  35.5  1
ADD   0  1
STORE   2
WRITE   2
```

QUAL O VALOR ESCRITO SE O VALOR FORNECIDO FOR 20?

```
READ   0
STORECONST  35.5  1
ADD   0  1
STORE   0
WRITE   0
```

```
READ   0
STORECONST  35.5  1
ADD   0  1
STORE   0
WRITE   2
```

```
READ   0
ADD   0  1
STORE   2
WRITE   2
```