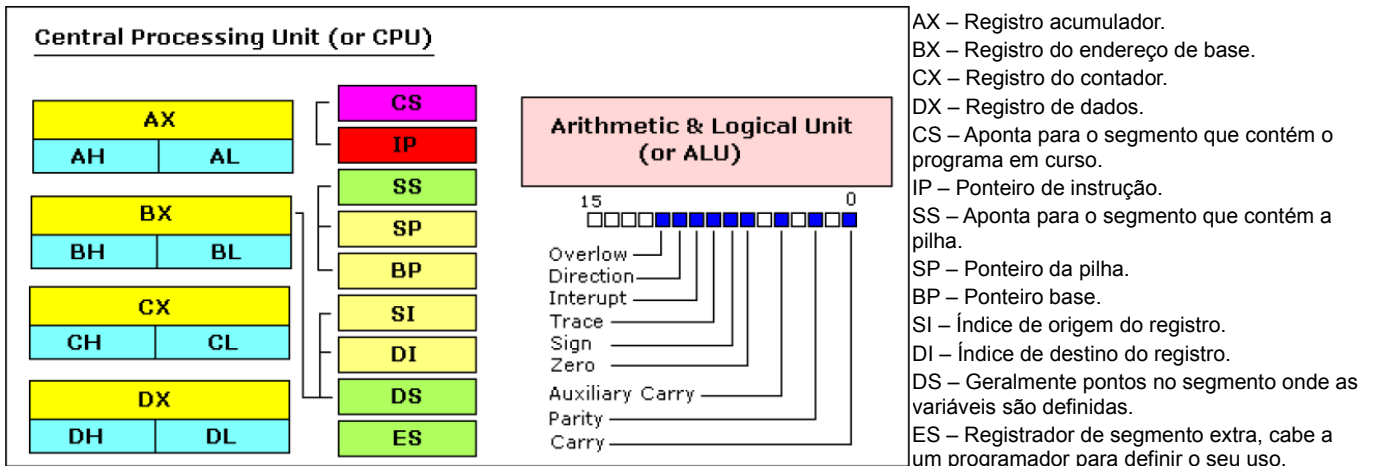


Assembly – Auxílio Rápido

Fernando Anselmo © Versão 1.0



Declaração de Variável

variável DB valor variável DW valor

DB - Define Byte.
DW - Define Word.

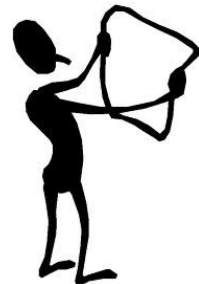


Arrays:

a DB 48h, 65h, 6Ch, 6Ch, 6Fh, 00h
b DB 'Ola', 0
c DB 5 DUP(9) = c DB 9, 9, 9, 9, 9

Comandos

- MOV destino, origem – mover um valor origem para o destino
- XCHG registro1, registro2 – intercambiar os dados entre dois registros
- LEA destino, origem – obter o valor OFFSET de uma variável de origem
- INT valor – o valor pode ser entre 0 e 255 (ou 0 a 0FFh), interrupções
- variável EQU expressão – criar uma variável constante
- RET – retornar ao sistema operacional
- ADD operador1, operador2 – adicionar o operador2 ao operador1
- SUB operador1, operador2 – subtrair o operador2 do operador1
- CMP operador1, operador2 – subtrair o operador2 do operador1 para flags
- AND operador1, operador2 – E Lógico entre dois operadores
- TEST operador1, operador2 – E Lógico entre dois operadores para flags
- OR operador1, operador2 – OU Lógico entre dois operadores
- XOR operador1, operador2 – OU EXCLUSIVO Lógico entre dois operadores
- MUL operador – multiplicar sem sinal (AX = AL * operador)
- IMUL operador – multiplicar com sinal (AX = AL * operador)
- DIV operador – dividir sem sinal (AX = AX / operador), se operador é BYTE resto em AH, se WORD resto em DX
- IDIV operador – dividir com sinal (AX = AX / operador), se operador é BYTE resto em AH, se WORD resto em DX
- INC – Incrementar em 1 byte. Afeta somente as flags: ZF, SF, OF, PF, AF
- DEC – Decrementar em 1 byte. Afeta somente as flags: ZF, SF, OF, PF, AF
- NOT – Reverter cada bit do operador.
- NEG – Fazer o operador como negativo. Afeta somente as flags: CF, ZF, SF, OF, PF, AF.



Salto

- JMP label – Transferir o controle para outro ponto do programa.
- LOOP label – Repete um laço com um determinado número (em CX) de vezes

Salto determinados

Instrução	Saltar se	Condição	Instrução Oposta	Exemplo
JZ, JE	Zero (igual)	ZF = 1	JNZ, JNE	include emu8086.inc
JNZ, JNE	Não zero (não igual)	ZF = 0	JZ, JE	
JC, JB, JNAE	Procede (baixo, não superior ao valor)	CF = 1	JNC, JNB, JAE	ORG 100h
JS	Sinal	SF = 1	JNS	MOV AL, 25
JO	Overflow	OF = 1	JNO	MOV BL, 10



JPE, JP	Pararidade par	PF = 1	JPO	
JG, JNLE JA*, JNBE*	Maior que (>)	ZF = 0 SF = OF	JNG, JLE JNA*, JBE*	
JL, JNGE JB*, JNAE*, JC*	Menor que (<)	SF <> OF	JNL, JGE JNB*, JAE*, JNC*	
JGE, JNE JAE*, JNB*, JNC*	Maior ou igual (>=)	SF = OF	JNGE, JL JNAE*, JB*	
JLE, JNG JBE*, JNA*	Menor ou igual (<=)	ZF = 1 SF <> OF	JNLE, JG JNBE*, JA*	

CMP AL, BL
JE equal
PUTC 'N'
JMP stop
equal:
PUTC 'Y'
stop:

* Valores sem sinal

Procedures, Pilhas e Macros

nome PROC

instruções;

RET

nome ENDP

PUSH – Guarda um valor de 16 bits na pilha.

POP – Obtém um valor de 16 bits na pilha.

PUSHF – Guarda na pilha

POPF – Obtém o valor guardado

CALL nomeprocedure

ORG 100h

MOV AX, 1234h

PUSH AX

MOV AX, 5678h

POP AX



nome MACRO parâmetros

instruções;

ENDM

minhaMacro MACRO p1, p2

MOV AX, p1

MOV BX, p2

ENDM

minhaMacro 4, 7

Biblioteca emu8086.inc

Macros:

PUTC char – macro com um parâmetro, imprimir um char em ASCII posição atual do cursor.

GOTOXY col, row – macro com dois parâmetros, define a posição do cursor.

PRINT string – macro com um parâmetro, imprimir uma string.

PRINTN string – macro com um parâmetro, imprimir uma string.

A mesma impressão, adiciona automaticamente o "retorno de carro" no final da cadeia.

CURSOROFF – desativar o cursor de texto.

CURSORON – ligar o cursor de texto.

Procedures:

PRINT_STRING – imprimir uma string terminada em null posição atual do cursor, recebe o endereço da string em DS:SI registo. Inserir DEFINE_PRINT_STRING.

PTTHIS – imprimir uma string terminada em null atual do cursor posição (tal como PRINT_STRING), mas recebe o endereço da string de pilha. O ZERO TERMINATED rescindido deve ser definida apenas após a instrução CALL.

GET_STRING – Obter uma string terminada em nulo a partir de um usuário, a sequência recebida é gravada no buffer DS:DI, o tamanho do buffer deve estar em DX. Procedimento interrompe quando o "Enter" é pressionado. Inserir DEFINE_GET_STRING antes directiva END.

CLEAR_SCREEN – Limpar a tela, (feito por rolagem janela de tela inteira), e definir a posição do cursor para cima dele. Inserir DEFINE_CLEAR_SCREEN antes directiva END.

SCAN_NUM – procedimento que recebe o número de vários dígitos SIGNED a partir do teclado e armazena o resultado no registo CX. Inserir DEFINE_SCAN_NUM.

PRINT_NUM – imprimir um número assinado em AX registo. Inserir DEFINE_PRINT_NUM e DEFINE_PRINT_NUM_UNNS.

PRINT_NUM_UNNS – imprimir um número sem sinal no registrador AX. Inserir DEFINE_PRINT_NUM_UNNS



```
include 'emu8086.inc'
ORG 100h
LEA SI, msg1
CALL print_string
CALL scan_num
MOV AX, CX
CALL pthis
DB 13, 10, 'Digitado foi: ', 0
CALL print_num
```

```
RET
msg1 DB 'Entre com o numero: ', 0
DEFINE_SCAN_NUM
DEFINE_PRINT_STRING
DEFINE_PRINT_NUM
DEFINE_PRINT_NUM_UNNS
DEFINE_PTHIS
END
```

Links Úteis

<http://www.emu8086.com/>

<http://www.lawebdelprogramador.com>