

Tabela Resumida de Instruções do Assembly do 8086

Instruções para transferência de dados:

MOV	copia um byte ou word de uma fonte especificada para um destino especificado.	MOV destino, fonte
PUSH	copia word especificada para o topo da pilha.	
POP	copia word do topo da pilha para local especificado.	
PUSHA	copia todos os registradores para a pilha (80186 em diante).	
POPA	copia words da pilha para todos os registradores.	
XCHG	Exchange bytes ou words.	

Instruções para entrada/saída de dados:

IN	copia um byte ou word de uma porta especificada para o acumulador (a porta é sempre uma cte ou DX).	IN AL, OC8H IN AX, DX
OUT	copia um byte ou word do acumulador para uma porta especificada (a porta é sempre uma cte ou DX).	OUT OC8H, AL OUT DX, AX

Instruções para transferência de endereço especial:

LEA	carrega em um registrador de 16 bits, o endereço (offset) de uma variável.	LEA BX, [BP]DELTA
LDS	carrega DS e um reg de 16 bits com um endereço FAR (32 bits).	LDS SI, SEGMENTO[DI]
LES	carrega DS e um reg de 16 bits com um endereço FAR (32 bits).	LES DI, [BX]BUFFER

Instruções para adição:

ADD	adição: ADD destino, fonte	ADD AL, 74H ADD DX, BX
ADC	adição com carry	ADC DX, BX ADC AL, 45H
INC	incrementa um byte ou word especificada	INC BL INC VARIÁVEL

Instruções para subtração:

SUB	subtração	SUB AL, VARIÁVEL SUB DX, BX
SBB	subtração com carry.	SUB AL, 74H SUB BX, [BX]
DEC	decrementa um byte ou word especificada	DEC CL DEC VARIÁVEL
NEG	inverte o sinal de um número.(complemento de 2)	
CMP	comparação: CMP destino, fonte	CMP AL, 01H CMP BH, CL CMP CX, VARIÁVEL_16BITS CMP VARIÁVEL_8BITS, BH

Instruções para multiplicação:

MUL	multiplicação não sinalizada byte x byte ou word x word	MUL BH ; AL x BH => result em AX MUL CX ; AX x CX => result em DX e AX
IMUL	multiplicação sinalizada	

Instruções para divisão:

DIV	divisão sem sinal:DIV fonte AL=quociente e AH=resto	DIV BH => AX/BH DIV CX => DXeAX/CX
IDIV	divisão com sinal:DIV fonte AL=quociente e AH=resto	DIV BL => AX/BL DIV CX => DXeAX/CX

Instruções lógicas:

NOT	inverte todos os bits do <i>destino</i>	NOT AX NOT [SI+06H]
AND	AND destino, fonte	AND BX, 00FFH AND CX, [SI]
OR	OR destino, fonte	OR CL, BH OR AH, 0H
XOR	XOR destino, fonte	XOR BX, FFH XOR AL, AL
TEST	AND fonte, fonte resultado somente nos flags.	TEST DX, AX TEST VARIÁVEL, 0FH

Instruções para deslocamento:

SHL	deslocamento à esquerda SHL destino, contagem	SHL DX, 1 SHL AX, CL
SHR	deslocamento à direita SHR destino, contagem	SHR BX, 1 SHR [BX], CL
SAL	deslocamento à esquerda SAL destino, contagem	SAL [SI+BX], 1 SAL AX, CL
SAR	deslocamento à direita SAR destino, contagem	SAR SI, 1 SAR DX, CL

Instruções para rotação:

ROL	rotação à esquerda ROL destino, contagem	ROL BH, 1 ROL BX, CL
ROR	rotação à direita ROR destino, contagem	ROR CX, 1 ROR DX, CL
RCL	rotação à esquerda com flag carry RCL destino, contagem	RCL [DI+BX], 1 RCL AX, CL
RCR	rotação à direita com flag carry RCR destino, contagem	RCR BX, 1 RCR [BX], CL

Instruções para string:

REP	repete operação_string o número de vezes contido em CX.
REPE/REPZ REPNE/REPNZ	repete operação_string enquanto condição for verdadeira ou enquanto CX maior que zero.
MOVS/MOVS/MOVSW	transfere string ([ES:DI] <= [DS:SI]) em blocos de byte (MOVS), palavra (MOVSW) ou pela definição destino/fonte.
CMPS/CMPSB/CMPSW	comparação de strings
INS/INSB/INSW	entrada de string por uma porta. A partir do 80186
OUTS/OUTSB/OUTSW	
SCAS/SCASB/SCASW	varre o string apontado por ES:DI procurando o conteúdo de AL (SCASB), AX (SCASW).
LODS/LODSB/LODSW	carrega AL ou AX com o byte (LODSB), a palavra (LODSW) ou o objeto designado por fonte (LODS) localizado em bloco de memória apontado por DS:SI.
STOS/STOSB/STOSW	carrega o bloco de memória apontado por DS:DI com AL, AX.

Instruções para set/reset de flags:

STC	set carry flag	CF = 1
CLC	clear carry flag	CF = 0
CMC	complement the carry flag	inverte CF
STD	set direction flag	DF = 1 (decrementa apontador de string)
CLD	clear direction flag	DF = 0 (incrementa apontador de string)
STI	set interrupt flag	IF = 1 (habilita interrupções)
CLI	clear interrupt flag	IF = 0 (desabilita interrupções)

CALL	chamada de subrotina	- sub-rotinas NEAR (perto): altera somente IP CALL LABEL => IP = LABEL CALL BX => IP = BX CALL WORD PTR [BX] => IP = [BX+3] e [BX+2] - sub-rotinas FAR (distante): altera CS e IP CALL SUBROT => SUBROT PROC FAR CALL DWORD PTR [BX] => CS = [BX+1] e [BX] IP = [BX+3] e [BX+2]
RET	Retorno de subrotina	
JMP	Jump incondicional	JMP LABEL JMP BX JMP WORD PTR [BX] - near jump JMP DWORD PTR [SI] - far jump

JA	Jump if above	salta se superior	CF=0 e ZF=0
JNBE	Jump if not below or equal	salta se não inferior ou igual	
JAЕ	Jump if above or equal	salta se superior ou igual	CF=0
JNB	Jump if not below	salta se não inferior	
JB	Jump if below	salta se inferior	CF=1
JNAЕ	Jump if not above or equal	salta se não superior ou igual	
JBE	Jump if below or equal	salta se inferior ou igual	CF=1 ou ZF=1
JNA	Jump if not above	salta se não superior	
JC	Jump if carry	salta se ocorrer carry	CF=1
JE	Jump if equal	salta se igual	ZF=1
JZ	Jump if zero	salta se resultado zero	
JG	Jump if greater	salta se maior (>)	ZF=0 e CF = OF
JNLE	Jump if not less or equal	salta se não menor ou igual	
JGE	Jump if greater or equal	salta se maior ou igual (>=)	SF = OF
JNL	Jump if not less	salta se não menor	
JL	Jump if less	salta se menor (<)	SF ≠ OF
JNGE	Jump if not greater or equal	salta se não maior ou igual	
JLE	Jump if less or equal	salta se menor ou igual (<=)	ZF=1 ou SF ≠ OF
JNG	Jump if not greater	salta se não maior	
JNC	Jump if no carry	salta se não ocorreu carry	CF=0
JNE	Jump if not equal	salta se não igual	ZF=0
JNZ	Jump if not zero	salta se não ocorreu zero	
JNO	Jump if no overflow	salta se não ocorreu overflow	OF=0
JNP	Jump if no parity	salta se não ocorreu paridade	PF=0
JPO	Jump if parity odd	salta se paridade ímpar	
JNS	Jump if not signed	salta se positivo	SF=0
JO	Jump if overflow	salta se ocorreu overflow	OF=1
JP	Jump if parity	salta se ocorreu paridade	PF=1
JPE	Jump if parity even	salta se paridade par	
JS	Jump if signed (if negative)	salta se negativo	SF=1

LOOP	dec CX e salta enquanto CX \neq 0
JCXZ	salta se CX = 0h

INT	executa uma rotina de interrupção.	INT n°
INTO	interrupção quando ocorrer overflow (OF = 1).	INTO ==> executa a interrupção 4 se OF=1
IRET	retorno de interrupção	

NOP	nenhuma operação
-----	------------------

Tabela Resumida de Diretivas do Assembly do 8086

Declaração de Dados

DB	Reserva para variáveis espaços na memória múltiplos de bytes	Nome DB Valor/?
DW	Reserva para variáveis espaços na memória múltiplos de 2 bytes (words)	Nome DW Valor/?
EQU	Atribui um Nome a um valor constante	Nome EQU Valor
DUP	Cria um Array com tantas posições quanto o número declarado antes da diretiva	10 DUP (?) – não inicializa os 10 elementos do <i>array</i> 20 DUP (Valor) – inicializa os 20 elementos do <i>array</i> com Valor
OFFSET	Obtem o <i>offset</i> do endereço de uma posição de memória dada por um Nome	MOV BX, OFFSET VAR – carrega BX com o <i>offset</i> do endereço de VAR

Estrutura do Programa

.MODEL	Indica o modelo de memória
.DATA	Indica ao montador o início do segmento de dados
.CODE	Indica ao montador o início do segmento de código
.STACK	Declara o número de posições de memória consecutivas reservadas para armazenar a pilha
.EXIT	Retorna ao Sistema Operacional o controle do processador
.STARTUP	Indica o início do programa
ORG	Estabelece o <i>offset</i> do endereço inicial
END	Indica o final do programa
PROC	Indica o início de um procedimento (subrotina)
ENDP	Indica o final de um procedimento (subrotina)
MACRO	Indica o início da sequência de instruções de uma macro
ENDM	Indica o final da sequência de instruções de uma macro