**INSTRUCCIONES DE TRANSFERENCIA DE DATOS:**

|  |  |
| --- | --- |
| MOV AX,BX | Almacena el contenido de BX en AX |
| XCHG AX,BX | Intercambia el contenido de AX y BX |
| XLAT | Almacena en AL el contenido de la direccion [BX+AL] |
| LEA AX,[BX] | Almacena en AX la direccion de memoria fisica o efectiva de BX |
| LDS/LES AX,[1000h] | Carga el contenido de una memoria de 32bits; parte baja en AX y el resto en DS o ES |
| PUSH AX | Inserta AX en la pila |
| POP DS | Carga el primer elemento de la pila en DS |
| PUSHF AX | Apila el registro AX |
| POPF AX | Desapila el registro AX |
| LAHF AX | Carga la parte baja del registro en AH |
| SAHF AX | Carga la parte alta del registro en AL |

**INSTRUCCIONES ARITMETICAS:**

|  |  |
| --- | --- |
| ADD/ADC AX, BX | Suma BX Y AX y almacena en AX. ADC aplica acarreos. |
| DAA AX,BX  \*Si AF=1 o valor de 4bits menos significativos > 9, suma a AL el valor 06h  \*Si CF=1 o valor de 4bits mas significativos > 9, suma a AL el valor 60h  \*Afecta a OF,SF,ZF,AF,PF | Realiza una suma en AL con la correccion BCD empaquetado. |
| SUB/SBB AX, BX | Resta AX con BX y lo almacena en AX. SBB aplica acarreos. |
| DAS AX,BX | Realiza una resta en AL con correccion BCD empaquetado. |
| NEG AX | Cambia el signo de AX |
| MUL/IMUL BX  \*Almacena en AX para operaciones de 8 bits  \*Almacena en DX:AX para operaciones de 16 bits | Realiza una multiplicacion. IMUL aplica signos. |
| DIV/IDIV BX  \*Almacena el cociente en AL y el resto en AH para operaciones de 8 bits  \*Almacena en DX(RESTO):AX(COCIENTE) para operaciones de 16 bits | Realiza una division. IDIV aplica signos. |
| CBW/CWD  \*Actua sobre AX y DX:AX | Realiza extension del bit de signo de byte a WORD. CWD lo mismo pero de WORD a DWORD |
| INC AX | Incrementa +1 |
| DEC AX | Decrementa -1 |

**INSTRUCCIONES LOGICAS:**

|  |  |
| --- | --- |
| OR AX,BX | Operacion logica or |
| AND AX,BX | Operacion logica and |
| XOR AX,BX | Operacion logica xor |

**INSTRUCCIONES DE COMPARACION:**

|  |  |
| --- | --- |
| TEST AX,BX  \*No afecta a ningun registro (AX NI BX)  \*Solo afecta al registro de estado | Operacion logica and |
| CMP AX,BX  \*No afecta a ningun registro (AX NI BX)  \*Solo afecta al registro de estado | Operacion de resta |

**INSTRUCCIONES DE DESPLAZAMIENTO Y ROTACIONES:**

|  |  |
| --- | --- |
| SAL/SHL AX,1 o CL  \*SHL Introduce un 0 y guarda el bit que sale en el bit CF del registro de estado | Desplaza a izquierda de AX una cantidad de X bits |
| SAR/SHR AX,1 o CL  \*SHR Repite el bit de signo y guarda el resultado en el bit CF del registro de estado | Desplaza a derecha de AX una cantidad de X bits |
| RCL AX,1 o CL | Realiza la rotacion a izquierda de los bits del operando a traves del bit CF del registro de estado |
| RCR AX,1 o CL | Realiza la rotacion a la derecha de los bits del operando a traves del bit CF del registro de estado |
| ROL AX,1 o CL  \*El bit que rota es almacenado en CF | Realiza la rotacion a izquierda de los bits del operando, ignorando el bit CF del registro de estado |
| ROR AX,1 o CL | Realiza la rotacion a derecha de los bits del operando, ignorando el bit CF del registro de estado |

**INSTRUCCIONES DE ENTRADA/SALIDA:**

|  |  |
| --- | --- |
| IN AX,X o DX | Realiza lectura de un puerto (solo si la direccion es < 255) |
| OUT AX,X o DX | Realiza escritura en un puerto (solo si la direccion es < 255) |

**INSTRUCCIONES DE CONTROL DEL PROGRAMA:**

*INSTRUCCIONES DE SALTO*

\*Todas cambian el registro IP

\*Si es salto lejano tambien cambia el registro CS (que no este en el mismo segmento de codigo)

\*Saltos absolutos (direccion absoluta especificada. Admite a memoria o registro si es cercano. Admite un valor o una direccion de memoria si es lejano) o Saltos relativos (hacia atras o adelante desde el IP. 8 o 16 bits. No puede ser lejano)

|  |  |
| --- | --- |
| JPM | Realiza un salto incondicional a la direccion especificada |
| CALL  \*Si es cercano almacena IP en pila  \*Si es lejano almacena IP y CS en pila  \*Modifica el SP | Realiza una llamada a un procedimiento externo |
| RET X  \*Se recuperan los valores IP y CS almacenados en pila  \*Para dejar la pila como estaba se puede usar POP o RET n (n=numero de posiciones a descartar en la pila) | Realiza un devolucion al procedimiento del programa, return |

*INSTRUCCIONES DE BUCLE*

|  |  |
| --- | --- |
| LOOP | Realiza un salto n veces siempre y cuando CX sea distinto de 0 y cada vez este decrementa en 1 |
| LOOPNE/LOOPNZ | Realiza un salto n veces siempre y cuando CX sea distinto de 0 y ZF sea igual a 0 |
| LOOPE/LOOPZ | Realiza un salto n veces siempre y cuando CX sea distinto de 0 y ZF sea igual a 1 |
| JCXZ | Realiza un salto n veces siempre y cuando CX sea igual a 0 |

**INTRUCCIONES DE CADENA DE CARACTERES:**

|  |  |
| --- | --- |
| REP  \*Se puede usar con MOVS,MOVDSW o STOS | Repite la instruccion a la que acompaña siempre y cuando CX sea distinto de 0 |
| REPE/REPZ  \*Se puede usar con CMPS o SCAS | Repite la instruccion a la que acompaña siempre y cuando CX sea distinto de 0 y ZF sea igual a 1 |
| REPNE/REPNZ  \*Se puede usar con CMPS o SCAS | Repite la instruccion a la que acompaña siempre y cuando CX sea distinto de 0 y ZF sea igual a 0 |
| MOVS/MOVSW dest,fuent  \*El destino es ES:DI  \*La fuente es DS:SI (cargar los valores antes de utilizar la instruccion)  \*Pasos: colocar el bit DF al valor correcto - cargar en SI el desplazamiento de la fuente - cargar en DI el desplazamiento del destino - cargar en CX el numero de elementos a mover - ejecutar la instruccion MOVS/MOVSB con el prefijo REP | Copia un byte o un WORD de una parte de memoria a otra |
| CMPS dest,fuent  \*Mismos pasos que MOVS pero aplicando el modificador REPE/REPZ o REPNE/REPNZ  \*Afecta todos los bits del registro de estado | Realiza una comparacion de 2 cadenas, devuelve el resultado en el registro de estado |
| SCAS/SCASW dest  \*Es necesario cargar en DI el desplazamiento del primer elemento de la cadena  \*Afecta a todos los bits del registro de estado | Localiza el valor contenido en AL o AX en una cadena, si encuentra el elemento, devuelve en DI el desplazamiento del siguiente |
| LODS/LODSW fuent  \*La fuente debe estar cargada en SI | Transfiere un elemento de una cadena a AL o AX |
| STOS/STOSW dest  \*El desplazamiento de la cadena debe cargarse en DI | Transfiere contenido de AL o AX a una cadena |

**OTRAS INSTRUCCIONES:**

|  |  |
| --- | --- |
| HLT | Para el procesador, solo se puede salir reiniciando o con una interrupcion externa |
| LOCK instruccion | Bloquea el acceso al bus por parte de otro dispositivo mientras se ejecute la instruccion a la que acompaña |
| WAIT | Genera estados de espera en el procesador hasta que se active |
| CLC/STC | Pone en 0 o 1 el bit CF del registro de estado |
| CMC | Cambia el valor del bit CF del registro de estado |
| CLI/STI | Pone en 0 o 1 el bit IF del registro de estado |
| CLD/STD | Pone en 0 o 1 el bit DF del registro de estado |
| NOP | Hace que el procesador ejecute NADA |