

## **JUEGO DE INSTRUCCIONES DE LOS PROCESADORES INTEL**

### **1.- INSTRUCCIONES DE TRANSFERENCIA DE DATOS.**

<b>MOV</b>	<b>destino, fuente</b>	<b>; transfiere dato de fuente a destino.</b>
<b>XCHG</b>	<b>destino, fuente</b>	<b>; intercambia contenido en1re fuente y destino.</b>
<b>PUSH</b>	<b>fuentes</b>	<b>; introduce en la pila el dato fuente.</b>
<b>POP</b>	<b>destino</b>	<b>; extrae de la pila una palabra y la deja en ; destino.</b>
<b>IN</b>	<b>fuentes</b>	<b>; carga AX con el dato entrante del puerto fuente.</b>
<b>INB</b>	<b>fuentes</b>	<b>; carga AL con puerto fuente.</b>
<b>IN</b>		<b>; carga AX con puerto apuntado por DX</b>
<b>INB</b>		<b>; carga AL con puerto apuntado por DX</b>
<b>OUT</b>	<b>destino</b>	<b>; escribe AX en puerto destino.</b>
<b>OUTB</b>	<b>destino</b>	<b>; escribe AL en puerto destino.</b>
<b>OUT</b>		<b>; escribe AX en puerto apuntado por DX</b>
<b>OUTB</b>		<b>; escribe AL en puerto apuntado por DX</b>

### **2.- ARITMETICA BINARIA.**

<b>NEG</b>	<b>destino</b>	<b>; hace negativo a destino (complemento de 2).</b>
<b>ADD</b>	<b>destino, fuente</b>	<b>; suma fuente y destino.</b>
<b>ADC</b>	<b>destino, fuente</b>	<b>; suma fuente, destino y acarreo.</b>
<b>SUB</b>	<b>destino, fuente</b>	<b>; resta fuente de destino.</b>
<b>SBB</b>	<b>destino, fuente</b>	<b>; resta fuente + acarreo de destino.</b>

## JUEGO DE INSTRUCCIONES DE LOS PROCESADORES INTEL

<b>MUL</b>	<b>fuente</b>	<b>; multiplica fuente con el acumulador. Si fuente es ; de 8 bits, se multiplica con AL y el resultado se ; almacena en AX. Si fuente es de 16 bits, la ; multiplicación se realiza con AX y el resultado se ; deja en DX:AX. DX tendrá la parte mas ; significativa. MUL supone que los operándos ; son positivos.</b>
<b>IMUL</b>	<b>fuente</b>	<b>; igual que MUL, pero se considera que los ; números son con signo.</b>
<b>DIV</b>	<b>fuente</b>	<b>; división de números no signados. El acumulador ; contendrá el dividendo y fuente el divisor. ; Existen dos modalidades. En la división de 8 ; bits, el dividendo es un número de 16 bits ; almacenado en AX. El cociente se deja en AL y el ; resto en AH. Si la división es de 16 bits, el ; dividendo es un numero de 32 bitB, almacenado ; en DX:AX ( DX mas Significativo ), el cociente ; se dejará en AX y el resto en DX.</b>
<b>IDIV</b>	<b>fuente</b>	<b>; igual que DIV, pero los argumentos se ; consideran signados.</b>
<b>INC</b>	<b>destino</b>	<b>; incrementa destino.</b>
<b>DEC</b>	<b>destino</b>	<b>; decrementa destino.</b>

### **3.- OPERACIONES LOGICAS.**

<b>NOT</b>	<b>destino</b>	<b>; complemento de destino.</b>
<b>AND</b>	<b>destino, fuente</b>	<b>; operación AND.</b>
<b>OR</b>	<b>destino, fuente</b>	<b>; operación OR.</b>
<b>XOR</b>	<b>destino, fuente</b>	<b>; operación OR exclusivo.</b>

#### **4.- GESTION DE CADENAS.**

<b>REP</b>		; repite la siguiente instrucción (de gestión de cadenas) CX veces.
<b>MOVSB</b>		; mueve el octeto [DI] ← [SI].
<b>MOVSW</b>		; mueve la palabra. [DI] ← [SI].
<b>CMPSB</b> <b>CMPW</b>		; compara los octetos apuntados por DI y SI. ; compara las palabras apuntadas por DI y SI. En el caso de CMPC y CMPW se normalmente se utilizar la opción REPZ o REPE, que repetirán la instrucción hasta que CX sea cero o que se detecte una igualdad. Con REPNZ y REPNE, se repite hasta que CX=0 o que se detecte una desigualdad, lo que ocurra primero.
<b>SCASB</b>		; busca el carácter contenido en AL en la cadena apuntada por DI.
<b>SCASW</b>		; busca la palabra contenida en AX en la cadena apuntada por DI.

#### **5.- ROTACIONES Y DESPLAZAMIENTOS.**

<b>SHL</b>	destino	; desplazamiento lógico a la izquierda.
<b>SHL</b>	destino, CL	; desplazamiento lógico a la izq. CL posiciones.
<b>SHR</b>	destino	; desplazamiento lógico a la derecha.
<b>SUR</b>	destino, CL	; desplazamiento lógico a la derecha, CL posiciones.
<b>SAL</b>	destino	; desplazamiento aritmético a la izquierda.

## JUEGO DE INSTRUCCIONES DE LOS PROCESADORES INTEL

<b>SAL</b>	<b>destino, CL</b>	<b>; desplz. aritmético. a la izq. CL posiciones.</b>
<b>SAR</b>	<b>destino</b>	<b>; desplz. aritmético a la derecha.</b>
<b>SAR</b>	<b>destino, CL</b>	<b>; desplz. aritmético a la derecha, CL posiciones.</b>
<b>ROL</b>	<b>destino</b>	<b>; rotación a la izquierda.</b>
<b>ROL</b>	<b>destino, CL</b>	<b>; rotación izquierda CL posiciones.</b>
<b>ROR</b>	<b>destino</b>	<b>; rotación a la derecha.</b>
<b>ROR</b>	<b>destino, CL</b>	<b>; rotación a la derecha CL posiciones.</b>
<b>RCL</b>	<b>destino</b>	<b>; rotación izquierda con acarreo incluido.</b>
<b>RCL</b>	<b>destino, CL</b>	<b>; rotación izq. con acarreo CL posiciones.</b>
<b>RCR</b>	<b>destino</b>	<b>; rotación a la derecha con acarreo incluido.</b>
<b>RCR</b>	<b>destino, CL</b>	<b>; rotación derecha con acarreo CL posiciones.</b>

### **6.- CONTROL DEL PROGRAMA.**

#### **SALTOS INCONDICIONALES:**

<b>JMP</b>	<b>objetivo</b>	<b>; salto directo (en el mismo segmento). El objetivo ; puede ser de 16 bits (salto largo), de 8 bits (salto ; corto) o indirecto (a través de los registros de ; base e índices).</b>
<b>JMP</b>	<b>objetivo, segmento</b>	<b>; salto directo a otro segmento. Tanto el objetivo ; como el segmento deben ser de 16 bits.</b>

## JUEGO DE INSTRUCCIONES DE LOS PROCESADORES INTEL

**SALTOS CONDICIONALES:** en todas estas instrucciones, el objetivo es de 8 bits (solo saltos cortos).

<b>JE</b>	<b>objetivo</b>	<b>; salta si es igual.</b>
<b>JZ</b>	<b>objetivo</b>	<b>; salta si es cero.</b>
<b>JNE</b>	<b>objetivo</b>	<b>; salta si no es igual.</b>
<b>JNZ</b>	<b>objetivo</b>	<b>; salta si no es cero.</b>
<b>JS</b>	<b>objetivo</b>	<b>; salta si es negativo.</b>
<b>JNS</b>	<b>objetivo</b>	<b>; salta si no es negativo.</b>
<b>JP</b>	<b>objetivo</b>	<b>; salta si es positivo.</b>
<b>JNP</b>	<b>objetivo</b>	<b>; salta si no es positivo.</b>
<b>JPE</b>	<b>objetivo</b>	<b>; salta si hay paridad par.</b>
<b>JPO</b>	<b>objetivo</b>	<b>; salta si hay paridad impar.</b>
<b>JL</b>	<b>objetivo</b>	<b>; salta si es menor.</b>
<b>JNGE</b>	<b>objetivo</b>	<b>; salta si no es mayor o igual.</b>
<b>JNL</b>	<b>objetivo</b>	<b>; salta si no es menor.</b>
<b>JGE</b>	<b>objetivo</b>	<b>; salta si es mayor o igual.</b>
<b>JLE</b>	<b>objetivo</b>	<b>; salta si es menor o igual.</b>
<b>JNG</b>	<b>objetivo</b>	<b>; salta si no es mayor.</b>
<b>JNLE</b>	<b>objetivo</b>	<b>; salta si no es menor o igual.</b>
<b>JG</b>	<b>objetivo</b>	<b>; salta si es mayor.</b>

JUEGO DE INSTRUCCIONES DE LOS PROCESADORES INTEL

<b>JB</b>	<b>objetivo</b>	<b>; salta si esta por debajo.</b>
<b>JNAE</b>	<b>objetivo</b>	<b>; salta si no esta por arriba o es igual.</b>
<b>JNB</b>	<b>objetivo</b>	<b>; salta si no esta por debajo.</b>
<b>JAЕ</b>	<b>objetivo</b>	<b>; salta si esta por arriba o es igual.</b>
<b>JBE</b>	<b>objetivo</b>	<b>; salta si esta por debajo o es igual.</b>
<b>JNA</b>	<b>objetivo</b>	<b>; salta si no esta por arriba.</b>
<b>JNBE</b>	<b>objetivo</b>	<b>; salta si no esta por debajo o es igual.</b>
<b>TEST</b>	<b>destino, fuente</b>	<b>; verifica. Realiza una AND que no deja en ningún ; lado. Solo afecta las banderas.</b>
<b>CMP</b>	<b>destino fuente</b>	<b>; compara. Realiza una resta que solo afecta las ; banderas.</b>
<b>LOOP</b>	<b>objetivo</b>	<b>; realiza un bucle CX veces. Objetivo es de 8 bits. ; su uso es:</b>  <b>;</b> <b>MOV CX, número_de_ciclos</b> <b>;</b> <b>objetivo:</b> <b>;</b> <b>(instrucciones a repetir )</b> <b>;</b> <b>- - -</b> <b>;</b> <b>LOOP objetivo</b>  <b>; también se tienen bucles condicionales:</b> <b>; LOOPZ, LOOPNZ, LOOPE y LOOPNE.</b>
<b>JCXZ</b>	<b>objetivo</b>	<b>; salta si CX = 0.</b>
<b>CALL</b>	<b>objetivo</b>	<b>; llama subrutina en el mismo segmento. El salto</b>

