

# Juego de instrucciones del Computador Teórico

\*\*No se debe escribir nada en esta hoja ya que se deberá utilizar en los exámenes de la asignatura

## Nomenclatura utilizada:

**Rd:** Tres bits que codifican el registro destino de una operación.

**Rs:** Tres bits que codifican el registro fuente de una operación.

**Rs1:** Tres bits que codifican el registro fuente1 de una operación.

**Rs2:** Tres bits que codifican el registro fuente2 de una operación.

**Ri:** Tres bits que codifican el registro índice para direccionamiento indirecto.

**Rd/s:** Tres bits que codifican un registro que es a la vez fuente y destino de una operación.

**Rx:** Tres bits que codifican el registro que contiene la dirección de destino para saltos indirectos.

**Inm8:** Valor numérico de 8 bits.

## Instrucciones de movimiento

Código instrucción	Descripción	Mnemónico	Operación
00000 00000000000	Instrucción nula.	NOP	—
00001 Rd Rs 00000	Transferencia entre registros.	MOV Rd, Rs	$Rd \leftarrow Rs$
00010 Rd Ri 00000	Copiar el contenido de la posición de memoria cuya dirección está en Ri en Rd.	MOV Rd, [Ri]	$Rd \leftarrow [Ri]$
00011 Ri Rs 00000	Copiar el contenido del registro Rs en la posición de memoria cuya dirección está en Ri.	MOV [Ri], Rs	$[Ri] \leftarrow Rs$
00100 Rd Inm8	Copia en los 8 bits menos significativo de Rd el dato codificado en los 8 bits del campo Inm8.	MOVL Rd, Inm8	$Rd_{bajo} \leftarrow Inm8$
00101 Rd Inm8	Copia en los 8 bits más significativo de Rd el dato codificado en los 8 bits del campo Inm8.	MOVH Rd, Inm8	$Rd_{alto} \leftarrow Inm8$
00110 Rs 00000000	Apila el contenido del registro Rs.	PUSH Rs	$Pila \leftarrow Rs$
00111 Rd 00000000	Desapila un valor en el registro Rd.	POP Rd	$Rd \leftarrow Pila$

## Instrucciones Aritmético-Lógicas

### De tres operandos

Código instrucción	Descripción	Mnemónico	Operación
01000 Rd Rs1 Rs2 00	Suma el contenido de los registros Rs1 y Rs2 y almacena el resultado en Rd.	ADD Rd, Rs1, Rs2	$Rd \leftarrow Rs1 + Rs2$
01001 Rd Rs1 Rs2 00	Resta el contenido del registro Rs2 al registro Rs1 y almacena el resultado en Rd.	SUB Rd, Rs1, Rs2	$Rd \leftarrow Rs1 - Rs2$
01010 Rd Rs1 Rs2 00	Realiza la operación OR con el contenido de los registros Rs1 y Rs2 y almacena el resultado en Rd.	OR Rd, Rs1, Rs2	$Rd \leftarrow Rs1 \text{ OR } Rs2$
01011 Rd Rs1 Rs2 00	Realiza la operación AND con el contenido de los registros Rs1 y Rs2 y almacena el resultado en Rd.	AND Rd, Rs1, Rs2	$Rd \leftarrow Rs1 \text{ AND } Rs2$
01100 Rd Rs1 Rs2 00	Realiza la operación XOR con el contenido de los registros Rs1 y Rs2 y almacena el resultado en Rd.	XOR Rd, Rs1, Rs2	$Rd \leftarrow Rs1 \text{ XOR } Rs2$

### De dos operandos

Código instrucción	Descripción	Mnemónico	Operación
01101 Rs1 Rs2 000 00	Resta el contenido del registro Rs2 al registro Rs1.	CMP Rs1, Rs2	$Rs1 - Rs2$

### De un operando

Código instrucción	Descripción	Mnemónico	Operación
10000 Rd/s 000000000	Realiza la operación lógica NOT con los bits del registro Rd/s.	NOT Rd/s	$Rd/s \leftarrow \neg Rd/s$
10001 Rd/s 000000000	Incrementa el contenido del registro Rd/s en una unidad.	INC Rd/s	$Rd/s \leftarrow Rd/s + 1$
10010 Rd/s 000000000	Decrementa el contenido del registro Rd/s en una unidad.	DEC Rd/s	$Rd/s \leftarrow Rd/s - 1$
10011 Rd/s 000000000	Cambia de signo (complemento a 2) el contenido del registro Rd/s.	NEG Rd/s	$Rd/s \leftarrow \neg Rd/s + 1$

## Interrupciones

Código instrucción	Descripción	Mnemónico	Operación
10100 000000000000	Pone a cero el flag de máscara de interrupción	CLI	$IF \leftarrow 0$
10101 000000000000	Pone a uno el flag de máscara de interrupción.	STI	$IF \leftarrow 1$
10110 000 Inm8	Genera la interrupción software con vector de interrupción Inm8.	INT Inm8	$Pila \leftarrow SR$ $Pila \leftarrow PC$ $PC \leftarrow [Inm8]$
10111 000000000000	Retorna de una interrupción (bien sea hardware o software).	IRET	$PC \leftarrow Pila$ $SR \leftarrow Pila$

## Instrucciones de control de flujo

### Salto incondicionales

Código instrucción	Descripción	Mnemónico	Operación
11000 000 Inm8	Realiza un salto relativo.	JMP Inm8	$PC \leftarrow PC + Ex16(Inm8)$
11001 Rx 000000000	Realiza un salto indirecto absoluto a la posición de memoria contenida en el registro Rx.	JMP Rx	$PC \leftarrow Rx$

### Llamadas a procedimientos

Código instrucción	Descripción	Mnemónico	Operación
11010 000 Inm8	Realiza un salto relativo a un procedimiento.	CALL Inm8	$Pila \leftarrow PC$ $PC \leftarrow PC + Ext16(Inm8)$
11011 Rx 000000000	Realiza un salto indirecto absoluto a un procedimiento.	CALL Rx	$Pila \leftarrow PC$ $PC \leftarrow Rx$

### Retorno de procedimientos

Código instrucción	Descripción	Mnemónico	Operación
11100 000000000000	Retorna desde un procedimiento (desapila PC).	RET	$PC \leftarrow Pila$

### Salto condicionales

Código instrucción	Descripción	Mnemónico	Operación
11110 <i>Cond</i> Inm8	El desplazamiento con el signo extendido se le suma a PC siempre y cuando la condición codificada en el campo Cond sea cierta.	<i>BRCond</i> Inm8	Si condición cierta: $PC \leftarrow PC + Ext16(Inm8)$
<i>Cond:</i> Tres bits que codifican la condición de salto de la siguiente forma:	000 $\rightarrow$ Salta si C = 1 $\rightarrow$ BRC 001 $\rightarrow$ Salta si C = 0 $\rightarrow$ BRNC 010 $\rightarrow$ Salta si O = 1 $\rightarrow$ BRO 011 $\rightarrow$ Salta si O = 0 $\rightarrow$ BRNO	100 $\rightarrow$ Salta si Z = 1 $\rightarrow$ BRZ 101 $\rightarrow$ Salta si Z = 0 $\rightarrow$ BRNZ 110 $\rightarrow$ Salta si S = 1 $\rightarrow$ BRS 111 $\rightarrow$ Salta si S = 0 $\rightarrow$ BRNS	

