

# JUEGO DE INSTRUCCIONES DE LA CPU ELEMENTAL

## 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.

**Inm\_8:** Valor numérico de 8 bits.

## Instrucciones de movimiento

Código instrucción	Descripción	Mnemónico	Operación
00 000 0000000000	Instrucción nula.	NOP	—
00 001 Rd Rs 00000	Transferencia entre registros.	MOV Rd, Rs	Rd ← Rs
00 010 Rd Ri 00000	Copiar el contenido de la posición de memoria cuya dirección está en Ri en Rd.	MOV Rd, [Ri]	Rd ← [Ri]
00 011 Ri Rs 00000	Copiar el contenido del registro Rs en la posición de memoria cuya dirección está en Ri.	MOV [Ri], Rs	[Ri] ← Rs
00 100 Rd Inm_8	Copia en los 8 bits menos significativo de Rd el dato codificado en los 8 bits del campo Inm_8.	MOVL Rd, Inm_8	Rdbajo ← Inm_8
00 101 Rd Inm_8	Copia en los 8 bits más significativo de Rd el dato codificado en los 8 bits del campo Inm_8.	MOVH Rd, Inm_8	Rdalto ← Inm_8
00 110 Rs 00000000	Apila el contenido del registro Rs.	PUSH Rs	Pila ← Rs
00 111 Rd 00000000	Desapila un valor en el registro Rd.	POP Rd	Rd ← Pila

## Instrucciones Aritmético-Lógicas

### De tres operandos

Código instrucción	Descripción	Mnemónico	Operación
01 000 Rd Rs1 Rs2 00	Suma el contenido de los registros Rs1 y Rs2 y almacena el resultado en Rd.	ADD Rd, Rs1, Rs2	Rd ← Rs1 + Rs2
01 001 Rd Rs1 Rs2 00	Resta el contenido del registro Rs2 al registro Rs1 y almacena el resultado en Rd.	SUB Rd, Rs1, Rs2	Rd ← Rs1 - Rs2
01 010 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 ← Rs1 OR Rs2
01 011 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 ← Rs1 AND Rs2
01 100 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 ← Rs1 XOR Rs2

### De dos operandos

Código instrucción	Descripción	Mnemónico	Operación
01 101 Rs1 Rs2 000 00	Resta el contenido del registro Rs2 al registro Rs1.	COMP Rs1, Rs2	Rs1 - Rs2

## De un operando (100)

Código instrucción	Descripción	Mnemónico	Operación
100 00 Rd/s 00000000	Realiza la operación lógica NOT con los bits del registro Rd/s.	NOT Rd/s	Rd/s ← ~Rd/s
100 01 Rd/s 00000000	Incrementa el contenido del registro Rd/s en una unidad.	INC Rd/s	Rd/s ← Rd/s + 1
100 10 Rd/s 00000000	Decrementa el contenido del registro Rd/s en una unidad.	DEC Rd/s	Rd/s ← Rd/s - 1
100 11 Rd/s 00000000	Cambia de signo (complemento a 2) el contenido del registro Rd/s.	NEG Rd/s	Rd/s ← ~Rd/s + 1

## Interrupciones

Código instrucción	Descripción	Mnemónico	Operación
101 00 000000000000	Pone a cero el flag de máscara de interrupción.	CLI	IF ← 0
101 01 000000000000	Pone a uno el flag de máscara de interrupción.	STI	IF ← 1
101 10 000 Inm_8	Genera la interrupción software con vector de interrupción Inm_8.	INT Inm_8	Pila ← SR Pila ← PC PC ← [Inm_8]
101 11 000000000000	Retorna de una interrupción (bien sea hardware o software).	IRET	PC ← Pila SR ← Pila

## Instrucciones de control de flujo

### Salto incondicionales

Código instrucción	Descripción	Mnemónico	Operación
11 00 0 000 Inm_8	Realiza un salto relativo.	JMP Inm_8	PC ← PC + Ex_16(Inm_8)
11 00 1 Rx 00000000	Realiza un salto indirecto absoluto a la posición de memoria contenida en el registro Rx.	JMP Rx	PC ← Rx

### Llamadas a procedimientos

Código instrucción	Descripción	Mnemónico	Operación
11 01 0 000 Inm_8	Realiza un salto relativo a un procedimiento.	CALL Inm_8	Pila ← PC PC ← PC + Ex_16(Inm_8)
11 01 1 Rx 00000000	Realiza un salto indirecto absoluto a un procedimiento.	CALL Rx	Pila ← PC PC ← Rx

### Retorno de procedimientos

Código instrucción	Descripción	Mnemónico	Operación
11 10 0 000000000000	Retorna desde un procedimiento (desapila PC).	RET	PC ← Pila

### Salto condicionales

Código instrucción	Descripción	Mnemónico	Operación
11 11 0 Cond Inm_8	El desplazamiento con el signo extendido se le suma a PC siempre y cuando la condición codificada en el campo Cond sea cierta.	BRFLAG Inm_8	Si condición cierta: PC ← PC + Ex_16(Inm_8)
<i>Cond:</i> Tres bits que codifican la condición de salto de la siguiente forma:		000 → Salta si C = 1 → BRC 001 → Salta si C = 0 → BRNC 010 → Salta si O = 1 → BRO 011 → Salta si O = 0 → BRNO	100 → Salta si Z = 1 → BRZ 101 → Salta si Z = 0 → BRNZ 110 → Salta si S = 1 → BRS 111 → Salta si S = 0 → BRNS

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

