



---

IIC2343 - Arquitectura de Computadores (II/2025)

## Ayudantía 4

Ayudantes: Daniela Ríos (danielaarp@uc.cl), Alberto Maturana (alberto.maturana@uc.cl), Tomás López Massaro (tomas.lopezm20@uc.cl)

### Pregunta 1: Preguntas Conceptuales

- (a) Un *assembler* es una herramienta de software que traduce un programa escrito en *assembly* a su representación equivalente en código máquina (binario), que puede ser ejecutado directamente por el procesador. Describa en detalle los pasos que debe realizar un *assembler* para procesar código *assembly* del computador básico.
- (b) Un *disassembler* es un programa que transforma código binario ejecutable en *assembly*. Describa cómo funcionaría un *disassembler* para el computador básico. ¿Es posible obtener el *assembly* original a partir de un programa ubicado en la memoria de instrucciones?

### Pregunta 2: Programabilidad

- (a) Describa qué hace el siguiente fragmento de código, escrito en el *assembly* del computador básico del curso. Indique los valores de los registros A y B, y el de las variables al finalizar la ejecución del código. Recuerde que el computador básico del curso realiza operaciones con 8 bits.

**Bonus:** Adicionalmente, indique los valores finales de las *flags* C, Z y N.

```
DATA:
    var1 4
    var2 8
    var3 0

CODE:
    MOV A, (var1)
    SHR B,A
    ADD B,A
    MOV (var1), B
    MOV A, (var2)
    SHL B,A
    ADD B,A
    MOV (var2), B
    MOV A, (var1)
    AND A, B
    NOT A, A
    MOV (var3), A
    INC (var3)
```

- (b) Programe en *assembly* un código que calcule el perímetro de un rectángulo. Para ello, se proporciona la sección **DATA**, la cual contiene las coordenadas de los vértices del rectángulo almacenadas en dos *arrays* de forma ordenada. Cada índice  $i$  de estos arreglos corresponde al punto  $(x_i, y_i)$ . Utilice **direccionamiento indirecto** y guarde el resultado final en la dirección **p**.

**Propuesto:** Resuelva este mismo problema calculando el largo de cada lado en cualquier orden. *Hint:* Utilice saltos.

```
DATA:
    p 0
    x 2
        6
        6
        2
    y 1
        1
        4
        4
CODE:
```

### Pregunta 3: Modificación del computador básico

Modifique la microarquitectura de la CPU Básica para poder realizar lo siguiente:

- Direccionamiento indirecto por registro base + *offset*. Por ejemplo, `MOV A, (B + offset)`.
- Además del direccionamiento anterior, agregar direccionamiento indirecto por registro base + registro índice. Por ejemplo, `MOV A, (B + A)`.
- Agregar instrucción `MUL A, B`, la cual multiplica las salidas del mux A y mux B, para luego usar el resultado como salida de la ALU. Se puede utilizar un bloque como multiplicador.
- Condition code* I, que indica cuando un número es impar. Indique como se debe procesar el *output* de la ALU para obtener este *condition code*. Este *condition code* debe incluirse en el *Status Register*.

Para cada uno de estos casos se deben señalar los bits de control, buses, componentes digitales y cualquier otra modificación que estime pertinente.

## Diagrama Computador Básico

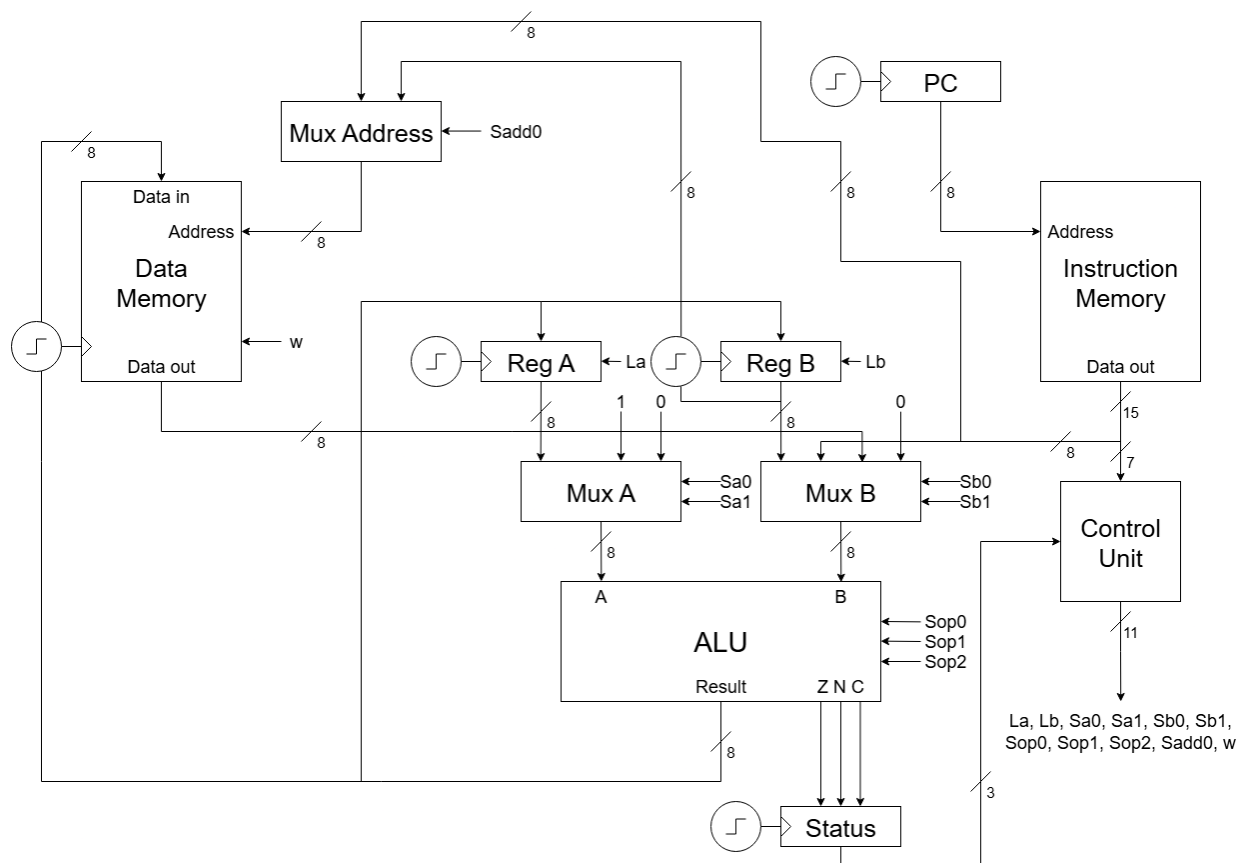


Figura 5: Computador Básico Sin Saltos.

## Tabla de instrucciones

Instrucción	Operandos	Opcode	Condición	L <sub>A</sub>	L <sub>B</sub>	S <sub>A</sub> 0,1	S <sub>B</sub> 0,1	S <sub>Op</sub> 0,1,2	S <sub>Add</sub> 0,1	W
MOV	A,B	0000000		1	0	ZERO	B	ADD	-	0
	B,A	0000001		0	1	A	ZERO	ADD	-	0
	A,Lit	0000010		1	0	ZERO	LIT	ADD	-	0
	B,Lit	0000011		0	1	ZERO	LIT	ADD	-	0
	A,(Dir)	0000100		1	0	ZERO	DOUT	ADD	LIT	0
	B,(Dir)	0000101		0	1	ZERO	DOUT	ADD	LIT	0
	(Dir),A	0000110		0	0	A	ZERO	ADD	LIT	1
	(Dir),B	0000111		0	0	ZERO	B	ADD	LIT	1
	A,(B)	0001000		1	0	ZERO	DOUT	ADD	B	0
ADD	B,(B)	0001001		0	1	ZERO	DOUT	ADD	B	0
	(B),A	0001010		1	0	A	ZERO	ADD	B	1
	A,B	0001011		1	0	A	B	ADD	-	0
	B,A	0001100		0	1	A	B	ADD	-	0
	A,Lit	0001101		1	0	A	LIT	ADD	-	0
	A,(Dir)	0001110		1	0	A	DOUT	ADD	LIT	0
	A,(B)	0001111		1	0	A	DOUT	ADD	B	0
	(Dir)	0010000		0	0	A	B	ADD	LIT	1
	(Dir),A	0010001		0	0	A	B	ADD	LIT	1
SUB	B,A	0010010		0	1	A	B	SUB	-	0
	A,Lit	0010011		1	0	A	LIT	SUB	-	0
	A,(Dir)	0010100		1	0	A	DOUT	SUB	LIT	0
	A,(B)	0010101		1	0	A	DOUT	SUB	B	0
	(Dir)	0010110		0	0	A	B	SUB	LIT	1
	(Dir),A	0010111		0	0	A	B	SUB	LIT	1
	A,B	0011000		1	0	A	B	AND	-	0
	B,A	0011001		0	1	A	B	AND	-	0
	A,Lit	0011010		1	0	A	LIT	AND	-	0
AND	A,(Dir)	0011011		1	0	A	DOUT	AND	LIT	0
	A,(B)	0011011		1	0	A	DOUT	AND	B	0
	(Dir)	0011100		0	0	A	B	AND	LIT	1
	(Dir),A	0011101		0	0	A	B	AND	LIT	1
	B,A	0011110		0	1	A	B	OR	-	0
	A,Lit	0011111		1	0	A	LIT	OR	-	0
	A,(Dir)	0100000		1	0	A	DOUT	OR	LIT	0
	A,(B)	0100001		1	0	A	DOUT	OR	B	0
	(Dir)	0100010		0	0	A	B	OR	LIT	1
OR	(Dir),A	0100011		0	0	A	B	OR	LIT	1
	A,A	0100011		1	0	A	-	NOT	-	0
	B,A	0100100		0	1	A	-	NOT	-	0
	(Dir)	0100101		0	0	A	B	NOT	LIT	1
	A,B	0100110		1	0	A	B	XOR	-	0
	B,A	0100111		0	1	A	B	XOR	-	0
	A,Lit	0101000		1	0	A	LIT	XOR	-	0
	A,(Dir)	0101001		1	0	A	DOUT	XOR	LIT	0
	A,(B)	0101010		1	0	A	DOUT	XOR	B	0
XOR	(Dir)	0101011		0	0	A	B	XOR	LIT	1
	A,A	0101100		1	0	A	-	SHL	-	0
	B,A	0101101		0	1	A	-	SHL	-	0
	(Dir)	0101110		0	0	A	B	SHL	LIT	1
	A,A	0101111		1	0	A	-	SHR	-	0
	B,A	0110000		0	1	A	-	SHR	-	0
	(Dir)	0110001		0	0	A	B	SHR	LIT	1
	A,A	0110010		0	1	ONE	B	ADD	-	0
	(B)	0110011		0	0	ONE	DOUT	ADD	B	1
INC	(Dir)	0110100		0	0	ONE	DOUT	ADD	LIT	1

Tabla 1: ISA del computador básico.

## Feedback ayudantía

Escanee el QR para entregar feedback sobre la ayudantía.

