

Guia de trabajos prácticos N° 3**Microprogramación**

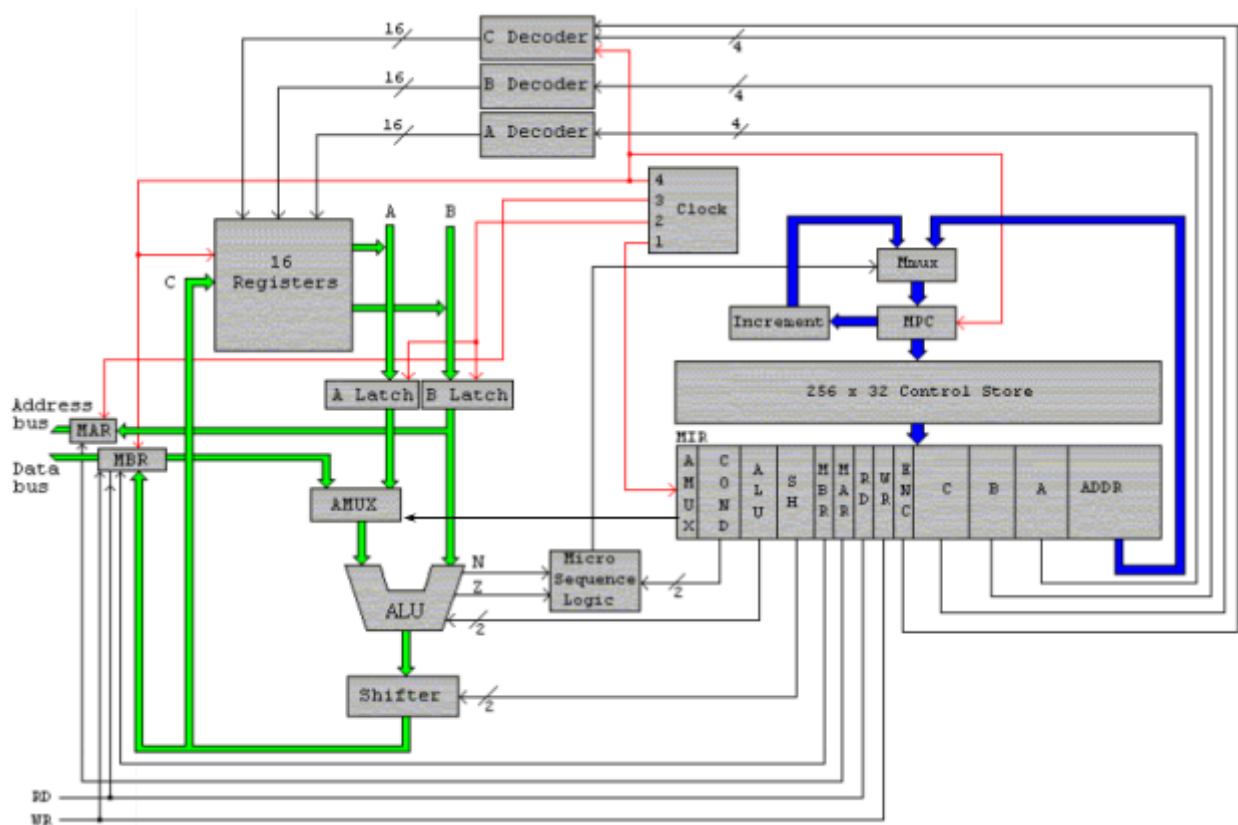
1. Construir en **Arcosim** módulos específicos para resolver los siguientes problemas:

- a. Crear un componente que incremente en 1 un valor con cada ciclo del clock (usando una única ALU).
- b. Crear un componente que decremente en 1 un valor con cada ciclo de clock hasta que el resultado sea cero.
- c. Hacer un componente que reciba en 2 buses 2 valores de 32 bits y devuelva en un bus el menor de los dos.
- d. Crear un componente que reciba una entrada de 32 bits donde, los 4 primeros bits sean un código de operación, los siguientes 14 bits sean el primer operando, y los siguientes 14 bits el segundo operando. El componente debe desensamblar la instrucción realizar la operación entre ambos operando y devolver el resultado en 14 bits por un display.

| Operaciones | Significado |
|-------------|-------------|
| 0000 | A + B |
| 0001 | A - B |
| 0010 | A * B |
| 0011 | A / B |

- e. Armar un componente con varias ALUs que reciba en un bus de 32 bits dos operandos (16 y 16) y realice las cuatro operaciones y las deje en 4 display diferentes.
- f. Armar un componente con una única ALU (con capacidad para realizar las 4 operaciones básicas) que reciba en un bus de 32 bits dos operandos (16 y 16) y realice las cuatro operaciones y las deje en 4 display diferentes.
- g. Armar un dispositivo que permita almacenar un valor en un registro, donde reciba en un bus el número binario del registro a configurar y en otro bus el valor del registro.
- h. Armar un dispositivo que permita copiar a la memoria todos los valores de los registros. Recibiendo en un bus la dirección inicial de la memoria.
- i. Armar un dispositivo con 16 registros, que reciba en el bus 1 un valor decimal y en el bus 2 otro valor decimal y copie automáticamente el valor del registro indicado en el bus 1 al registro indicado en el bus 2.
- j. Contar la cantidad de bits en 1 que se reciben por bus de 32 bits.

Según la siguiente micro-arquitectura horizontal:



Con registros:

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|----|----|----|----|-----|-------|-------|-------|--------|--------|----|----|----|----|----|----|
| PC | AC | SP | IR | TIR | 0 | +1 | -1 | AM ask | SM ask | Ax | Bx | Cx | Dx | Ex | Fx |
| ? | ? | ? | ? | ? | x0000 | x0001 | xFFFF | x0FFFF | x000F | ? | ? | ? | ? | ? | ? |

Estructura del MIR

| | | | | | | | | | | | | |
|---|---|-----|----|---|---|---|---|---|---|---|---|------|
| A | C | ALU | SH | M | M | R | W | E | C | B | A | ADDR |
| M | O | | | B | A | D | R | N | | | | |
| U | N | | | R | R | | | C | | | | |
| X | D | | | | | | | | | | | |

AMUX

0 = Buffer A
1 = MBR

COND

0 = No salta
1 = Si N==1
2 = Si Z==1
3 = Salta siempre

ALU

0 = A + B
1 = A & B
2 = A
3 = ~A

SH

0 = No desplace
1 = alu >> 1
2 = alu << 1
3 = (no usado)

MBR, MAR, RD, WR, ENC

0 = No
1 = Si

2. Explicar en lenguaje natural lo que hacen los siguientes grupos de micro instrucciones.

a.

| A M U X | C O N D | ALU | SH | M B R | M A R | R D | W R | E N C | C | B | A | ADDR |
|------------------|----------------------|---------|--------|-------------|-------------|--------|--------|-------------|-------|-------|-------|----------|
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 4 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |

b.

| A M U X | C O N D | ALU | SH | M B R | M A R | R D | W R | E N C | C | B | A | ADDR |
|------------------|----------------------|---------|--------|-------------|-------------|--------|--------|-------------|-------|-------|-------|----------|
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 4 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |

c.

| A M U X | C O N D | ALU | SH | M B R | M A R | R D | W R | E N C | C | B | A | ADDR |
|------------------|----------------------|---------|--------|-------------|-------------|--------|--------|-------------|-------|-------|-------|----------|
| 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 13 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 6 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 10 | (+3) |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 6 | 0 | 0 |
| 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | yy |
| 0 | 3 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 3 | 8 | xx |

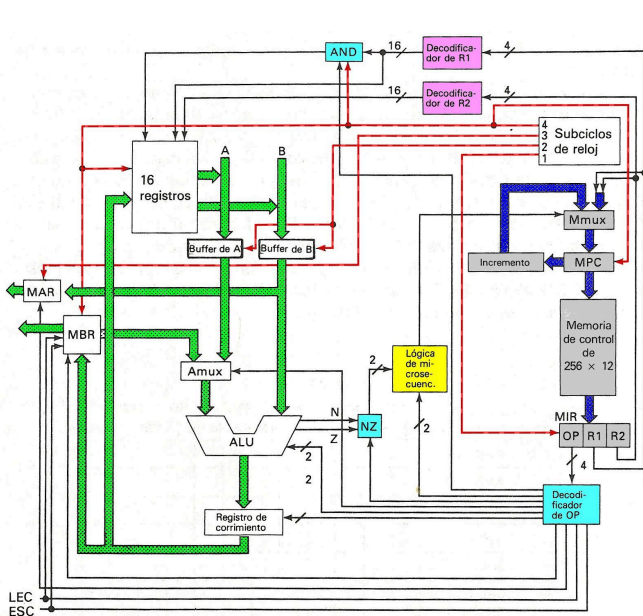
3. Codificar las siguientes micro-instrucciones:

- a. $Ax = (Ax + Bx) * 2$
- b. $Dx = (Fx + Ax) / 4$
- c. $[Fx] = [Cx - Dx]$
- d. $[(IR \text{ and } AMask)] = Ax$
- e. $Bx = Ex \text{ AND } [Fx]$
- f. $[SP+1] = [Dx * 2] + Cx$
- g. $Ax = Ax \text{ or } Bx$
- h. $PC = [Dx] + Cx$
- i. $[(IR \text{ and } AMask)] = (Ex * 2) + (Ex / 2)$
- j. $[Dx] = (Ax + Bx + Cx) * \frac{1}{2}$
- k. $Ax = Ax * Bx$

4. Resuelva las operaciones del ejercicio anterior usando las siguientes microinstrucciones de una arquitectura vertical:

| Binario | Mnemónico | Instrucción | Significado |
|---------|---------------|--------------------------|---|
| 0000 | ADD | Suma | $r1:=r1+r2$ |
| 0001 | AND | Y bit a bit | $r1:=r1\&r2$ |
| 0010 | MOVE | Mueve registro | $r1:=r2$ |
| 0011 | COMPL | Complemento | $r1:=\text{inv}(r2)$ |
| 0100 | LSHIFT | Desplaza a la Izquierda | $r1:=r2<<1$ |
| 0101 | RSHIFT | Desplaza a la Derecha | $r1:=r2>>1$ |
| 0110 | GETMBR | Almacena MBR en registro | $r1:=MBR$ |
| 0111 | TEST | Examina registro | $N:=(r1<0); Z:=(r1==0)$ |
| 1000 | BEGRD | Comienza la Lectura | $MAR:=r1; rd$ |
| 1001 | BEGWR | Comienza la Escritura | $MAR:=r1; MBR:=r2; wr$ |
| 1010 | CONRD | Continúa la Lectura | rd |
| 1011 | CONWR | Continúa la Escritura | wr |
| 1100 | | | |
| 1101 | NJUMP | Salta si N=true | $\text{if } N \text{ then goto } r1<<4 \mid r2$ |
| 1110 | ZJUMP | Salta si Z=true | $\text{if } Z \text{ then goto } r1<<4 \mid r2$ |
| 1111 | UJUMP | Salta siempre | $\text{goto } r1<<4 \mid r2$ |

5. Dada la arquitectura vertical propuesta por Tanenbaum y su matriz representativa del Decodificador de OP, se quieren implementar nuevas microinstrucciones, a saber:



| Código de operación de microinstrucción | | ALUL | | SHL | AMUX | | MAR | RD | MSLH |
|---|--------|------|-----|-----|------|-----|-----|------|------|
| | | ALUH | SHH | NZ | AND | MBR | WR | MSLL | |
| 0 | ADD | | | | + | + | | | |
| 1 | AND | | + | | + | + | | | |
| 2 | MOVE | + | | | + | + | | | |
| 3 | COMPL | + | + | | + | + | | | |
| 4 | LSHIFT | + | | + | + | + | | | |
| 5 | RSHIFT | + | | + | + | + | | | |
| 6 | GETMBR | + | | | + | + | + | | |
| 7 | TEST | + | | | + | | | | |
| 8 | BEGRD | + | | | | | + | + | |
| 9 | BEGWR | + | | | | | + | + | |
| 10 | CONRD | + | | | | | | + | |
| 11 | CONWR | + | | | | | | | + |
| 12 | | | | | | | | | |
| 13 | NJUMP | + | | | | | | | + |
| 14 | ZJUMP | + | | | | | | | + |
| 15 | UJUMP | + | | | | | | | + |
| | SetMBR | | | | | | | | |
| | AddMBR | | | | | | | | |
| | PJMBR | | | | | | | | |

- a. **SetMBR** carga el registro MBR con el contenido de un registro de uso general, $MBR:=r2$
SetMBR AC
- b. **AddMBR** acumula el contenido del registro MBR y de un registro de uso general, $r1:=MBR+r1$
AddMBR AC
- c. **ZJMBR** si el valor del registro MBR es cero, entonces salta a la microinstrucción cuya dirección se calcula como $(r1<<4) | r2$
PJMBR XXXX