

TDC_Práctica4. Diseño de memorias en VHDL

4.1. Diseño VHDL de una Memoria de datos (RAM)

Diseño de RAM en VHDL

1. Crear el **Proyecto24** que incluya la descripción de una RAM 16x4 con modo de escritura Write-First. Comprueba su funcionamiento en Basys2.

Nombre del Proyecto: **Proyecto24**

Nombre del Módulo: **RAM_16x4**

Nombre de la arquitectura: **Behavioral**

Nombre E/S y Recursos en placa:

- Sw3,Sw2,Sw1,Sw0 como dato de entrada: **DataIn(3:0)**
- Pulsador BTN3 como entrada de control **WE**
- Sw7,Sw6,Sw5,Sw4 como dirección de entrada: **Address(3:0)**
- Leds 3 a 0 como dato de salida: **DataOut(3:0)**
- Reloj

Nota: durante la fase de prueba, deberá almacenar algunos datos en la RAM para posteriormente leerlos.

Diseño de RAM en VHDL

2. Diseña una RAM cuyo modo de escritura sea READ-FIRST.

Nombre del Proyecto: **Proyecto25**

Nombre del Módulo: **RAM_16x4_RF**

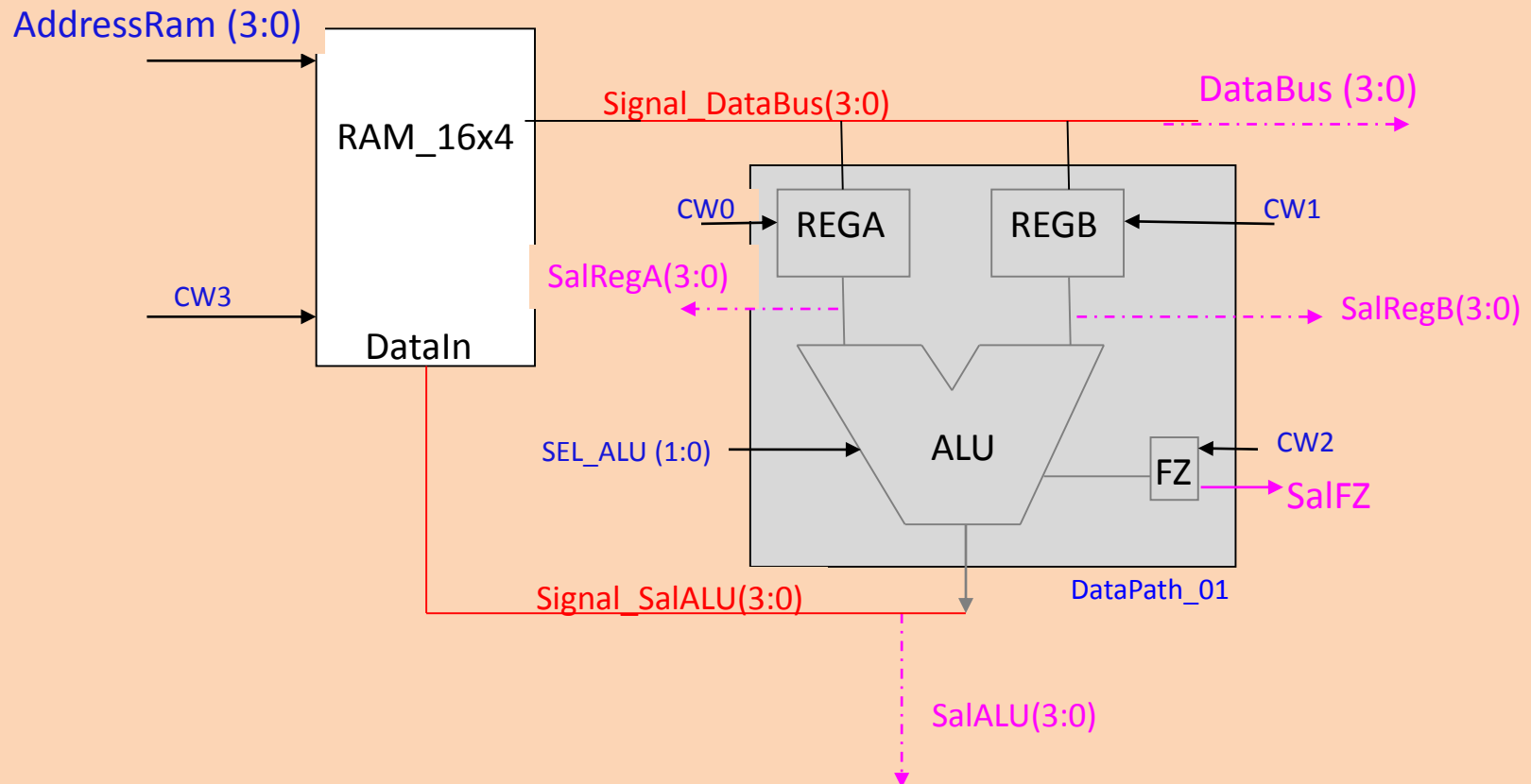
Nombre de la arquitectura: **Behavioral**

3. Crear un fichero de test para comprobar el funcionamiento de la RAM en el simulador (**RAM_16x4_RF_tb**).

4.2. Memoria de datos (RAM) + DataPath

RAM más DataPath_01

4. Crea un nuevo proyecto, **Proyecto26**, y añade un módulo VHDL, **RAM_DataPath_01**, donde interconectes el camino de datos del **Proyecto23**, DataPath_01, con la memoria RAM Write-First del **Proyecto24**.



RAM más DataPath_01

5. Para comprobar su funcionamiento añada al mismo **Proyecto26** un nuevo módulo, **TOP02**, que una RAM_DataPath_01 con Disp7seg_4ON.

Nombre del Proyecto: **Proyecto26**

Nombre del Módulo: **TOP02**

Nombre E/S y Recursos en placa:

- Sw7-4 → Address RAM (3:0)
- Sw3-2 → SelALU(1:0)
- Sw0 → CW3
- BTN3-0 → CW2,CW1,CW0,Reset
- CLK → CLK

- Display3 → DataBus (3:0)
- Display2 → SalRegA(3:0)
- Display1 → SalRegB(3:0)
- Display0 → SalALU(3:0)
- Led0 → SalFZ

NOTA: Modifica el código de RAM_16x4, solo en el proyecto 26, para que la RAM incluya los siguientes datos al arrancar. (Ver tabla).

RAM más DataPath_01

Dirección HEX	Dirección (4 bits)	Dato (4 bits)
0	0000	0101
1	0001	1111
2	0010	1110
3	0011	1110
4	0100	1110
5	0101	0000
6	0110	0001
7	0111	0010
8	1000	0101
9	1001	1111
A	1010	0000
B	1011	0000
C	1100	0000
D	1101	0000
E	1110	0000
F	1111	0000

Valor inicial de RAM

1. Dando valor en cada posición de cada elemento.

Elemento 15 del array

```
signal RAM : ram_type := ("0000", "0001", "0010", "0011", "0000"  
, "1111", "0000", "0000", "0000"  
, "1111", "1100", "0011", "0010"  
, "0001", "0000", "0000");
```

Elemento 0 del array

Valor inicial de RAM

2. En el caso de que tomen valores diferentes se puede realizar de la siguiente forma:

```
signal RAM: ram_type :=  
( 0 => "0001",  
  1 => "1111",  
  2 => "0101",  
  others => "0000");
```

Formato binario

Indice de cada
elemento de array

```
signal RAM: ram_type :=  
( 0 => X"1",  
  1 => X"F",  
  2 => X"5",  
  others => "0000");
```

Formato hexadecimal

Valor inicial de RAM

Con la palabra clave “*others*” se facilita asignar todo a ‘0’.
A su vez permite dos opciones:

1.1. Establecer el valor de cada elemento del array

```
RAM<=(others => "0000");
```

1.2. Establecer el valor de cada bit del array

```
RAM<=(others => (others => '0'));
```

4.3. Diseño VHDL de una Memoria de programa (ROM)

6. Crea el **Proyecto27**. Y crea un módulo de memoria ROM denominado **ROM_8x10**. Toma como referencia el código visto en clase.

Modifica el valor de la constante para almacenar los datos de la tabla presente en la transparencia siguiente.

Comprueba su funcionamiento en Basys2. (Usa los 8 LEDs de la placa como salida, y dos externos para completar los 10).

Leer las especificaciones del manual de Basys2 sobre como usar los puertos JA,JB,JC y JD.

4.4. ROM + RAM + DataPath

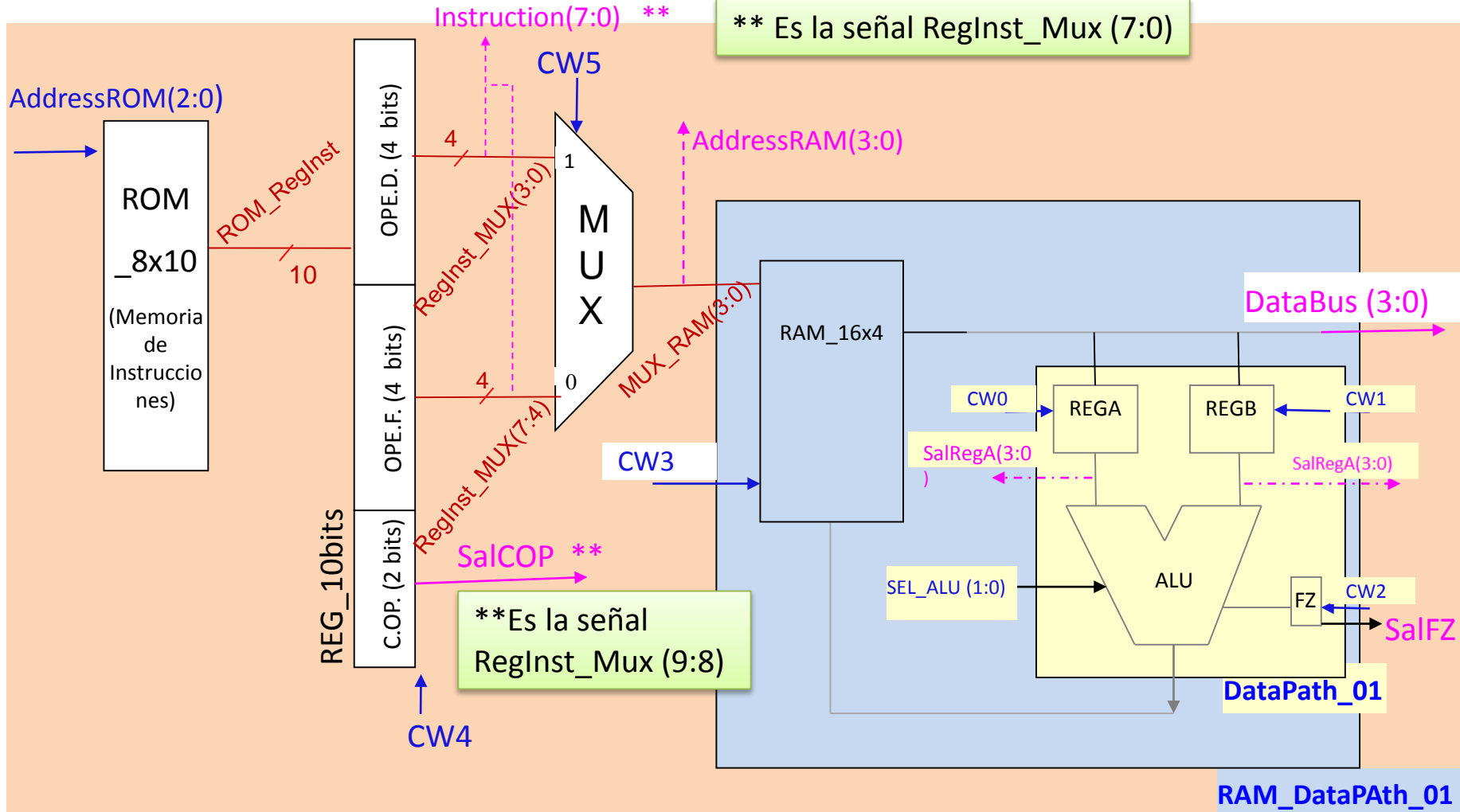
7. Crea el **Proyecto28** y añade el módulo **RAM_DataPath_01** del **Proyecto26** y el módulo **ROM_8x10** del **Proyecto27**. A continuación crea un nuevo módulo VHDL denominado **ROM_RAM_DataPath_01** que interconecte los anteriores como se aprecia en la figura. Añade los módulos prediseñados adicionales que se ven en el esquema siguiente.

Nota: Modifica el código de la ROM para escribirle los siguientes datos. Modifica la ROM solo en el proyecto28.

ROM+RAM+DataPath01

Dir. Hex	Instrucción (10bits)	Operación de la inst.
0	10 1000 0011	RAM(8)+RAM(3)→RAM(3)
1	00 0001 0100	RAM(1) → RAM(4)
2	10 0000 1001	RAM(0)+RAM(9)→RAM(9)
3	11 0001 0001	RAM(1)-RAM(1) → RAM(1)
4	01 0011 0111	Si FZ=1, DirSalto→PC
5	00 0100 0000	
6	00 0000 0000	
7	00 0000 0000	

ROM+RAM+DataPath01



8. Añade al **Proyecto28** el módulo **Disp7Seg_4ON**. Luego añade un nuevo módulo denominado TOP03. Interconecta **ROM_RAM_DataPath_01** con **Disp7Seg_4ON**.
9. Comprueba que el diseño funciona usando Basys2.

Asignación de recursos:

- Sw7-5 → AddressROM(2:0)
- Sw4-3 → Sel_ALU(1:0)
- Sw2,1,0 → CW5, CW4, CW3
- BTN3-0 → CW2,CW1,CW0,Reset

- Led7-0 → Instruction(7:0)
- Display3 → AddressRAM (3:0)
- Display2 → DataBus(3:0)
- Display1 → SalRegA(3:0)
- Display0 → SalRegB(3:0)
- Led Externo (B5,B7) → SalCOP (1:0)
- Led Externo (D12) → SalFZ

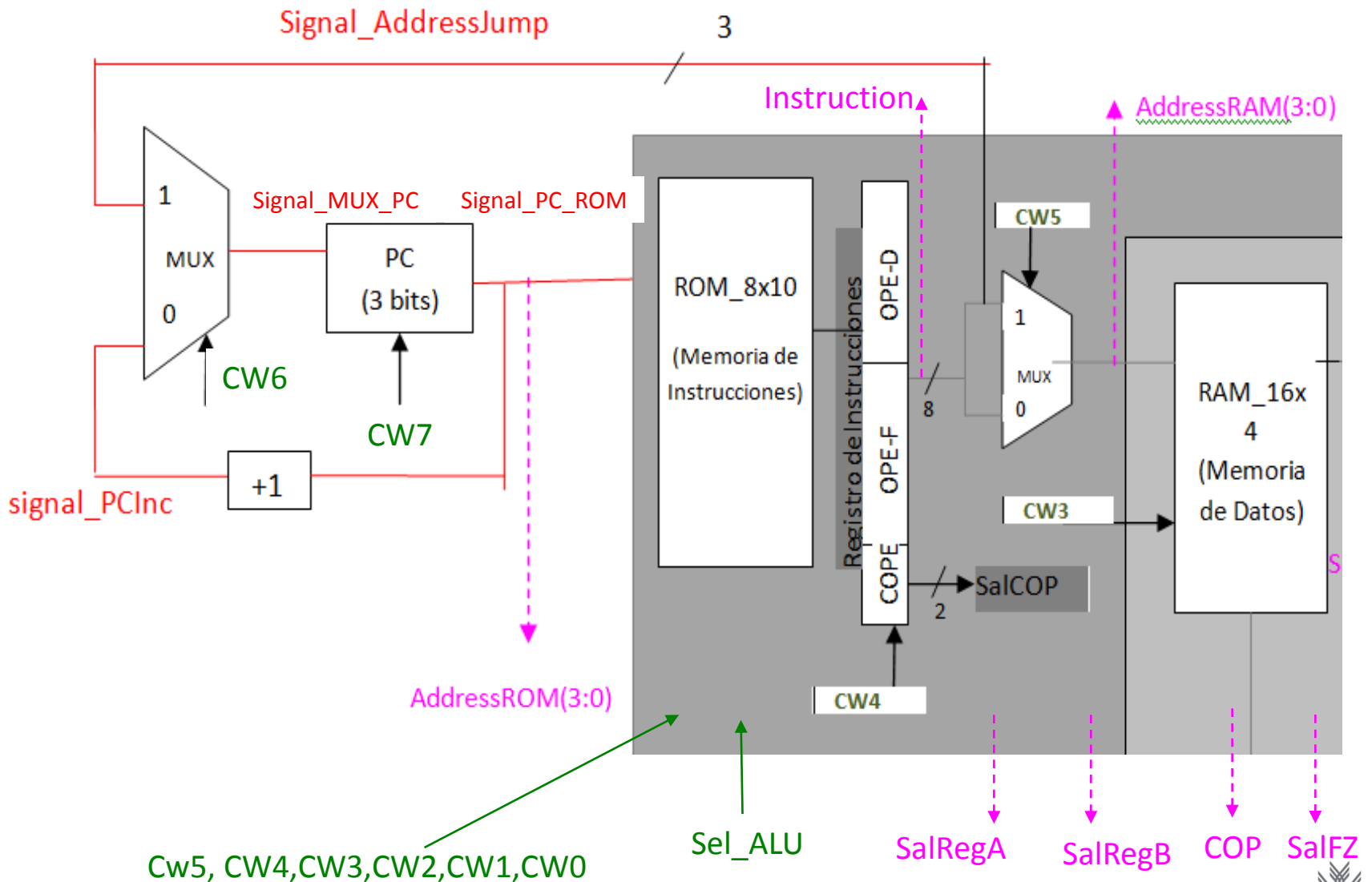
4.5. Microarquitectura completa de Didacomp

Microarquitectura de Didacomp

10. Crea el **Proyecto29** añade un módulo denominado **MicroArq_DidaComp**. Este módulo debe conectar todos los módulos necesarios para construir la microarquitectura de DidaComp (transparencia siguiente).
11. Añade un segundo módulo al proyecto denominado **TOP04** que interconecte **MicroArq_DidaComp** con **Disp7Seg_4ON**.
12. Comprueba que **TOP04** funciona usando Basys2. (Asignación de recursos en la transparencia siguiente)

Microarquitectura de Didacomp

MicroArq_Didacomp



Asignación de recursos:

- Sw6-5 → CW7,CW6
- Sw4-3 → SelALU(1:0)
- Sw2,1,0 → CW5, CW4, CW3
- BTN3-0 → CW2,CW1,CW0,Reset

- Led7-0 → RegInst(7:0)
- Display3 → AddressROM (3:0)
- Display2 → AddressRAM(3:0)
- Display1 → SalRegA(3:0)
- Display0 → SalRegB(3:0)
- Led Externo (B5,B7) → COP (1:0)
- Led Externo (D12) → SalFZ

Microarquitectura de Didacomp

13. Completa la tabla que recoge la función de cada línea CW, el bus de control.
14. Observa si hay alguna anomalía y trata de subsanar el problema añadiendo algún módulo prediseñado.

Bibliografía del tema

[1] Manual de referencia de la placa de evaluación Basys2

http://www.digilentinc.com/Data/Products/BASYS2/Basys2_rm.pdf.

[2] Datasheet de la FPGA Spartan3E de Xilinx

http://www.xilinx.com/support/documentation/data_sheets/ds312.pdf

[3] Diseño de circuitos digitales con VHDL

<http://eciencia.urjc.es/handle/10115/4045>