

TDC_4. Diseño de un procesador (VHDL)

Repaso de:

- Elementos constructivos de un computador simple.
- Repaso del diseño de un repertorio de instrucciones (ISA)
- Implementación del ISA: Diferentes microarquitecturas

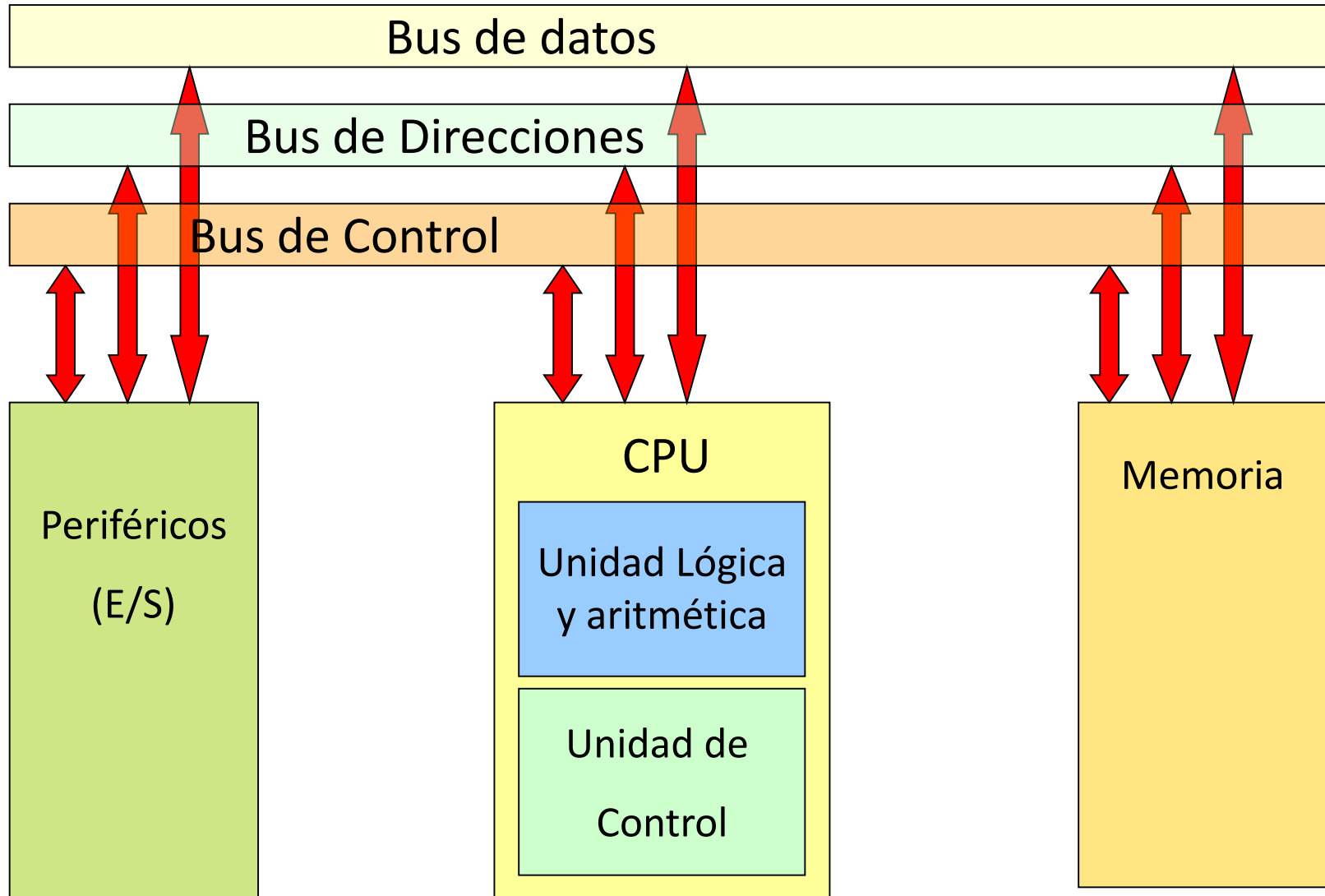
Objetivos:

- Descripción VHDL de los principales elementos de un computador.
- Diseño de un procesador simple mediante VHDL
- Verificación del diseño en placa desarrollo

4.1. Elementos de un computador simple

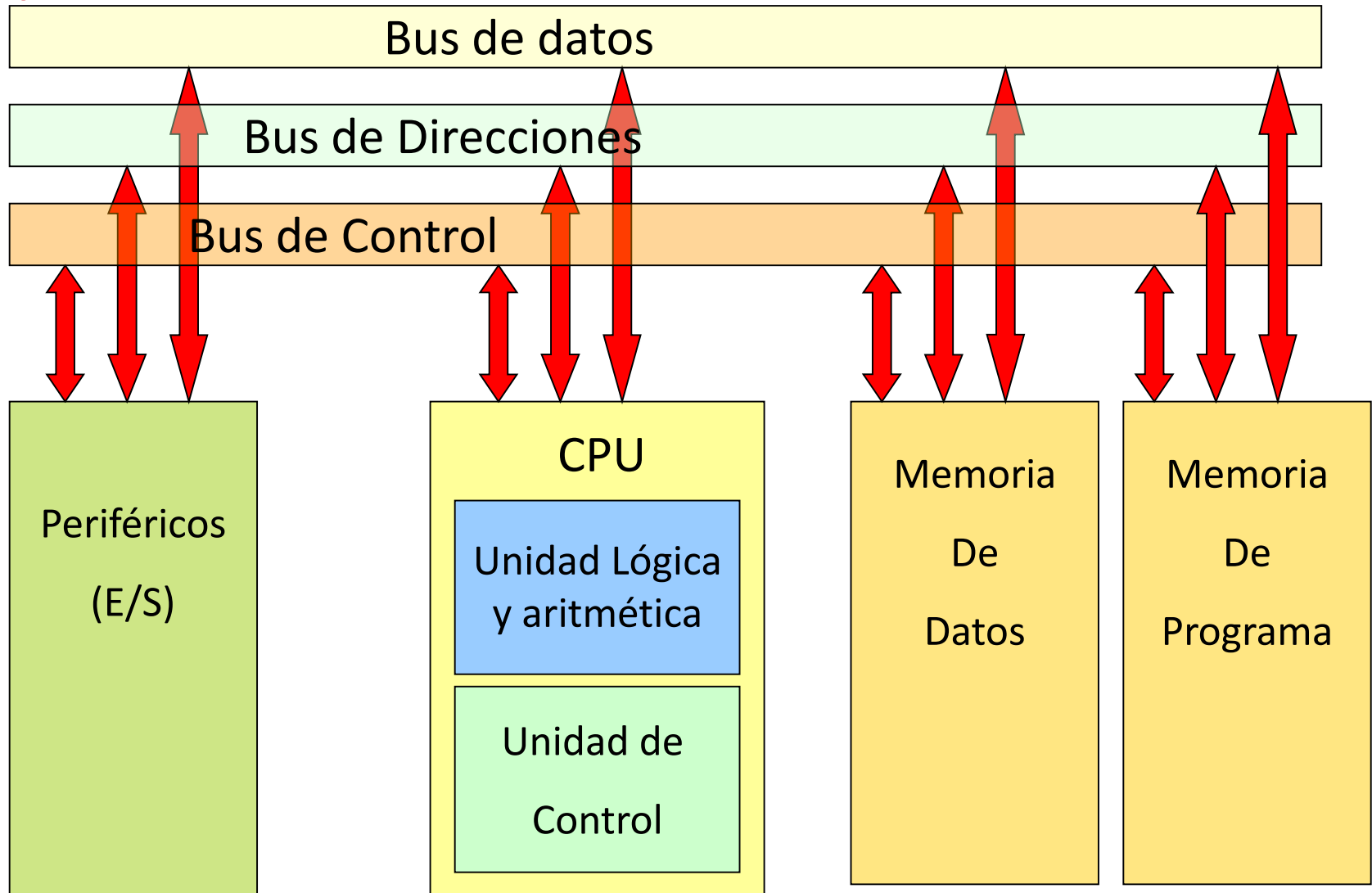
Elementos de un computador simple

Arquitectura Von Neumann



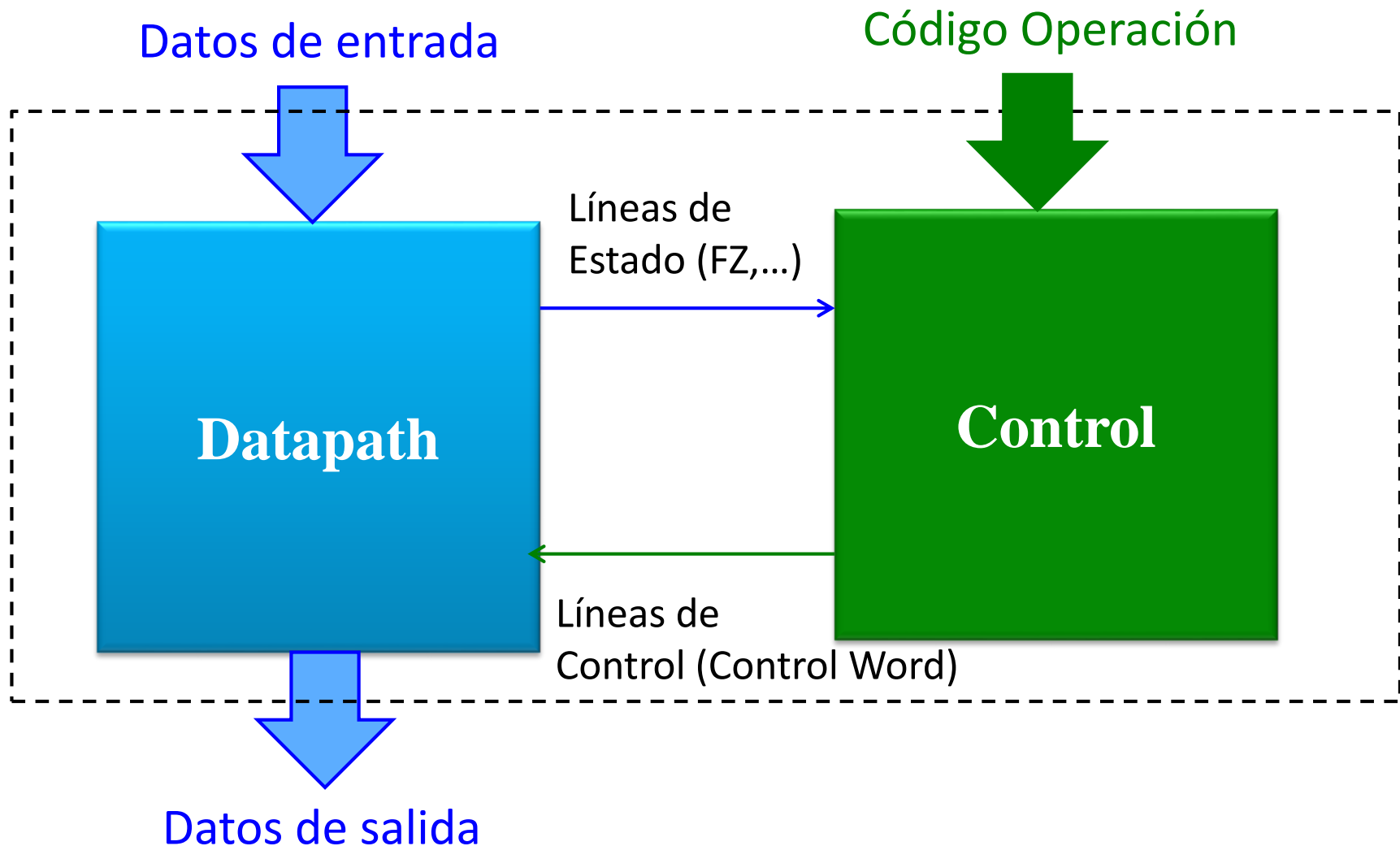
Elementos de un computador simple

Arquitectura Harvard



Elementos de un computador simple

Unidad central de proceso (CPU)



4.2. Camino de datos (DataPath)

Camino de datos (Datapath)

- Elementos encargados de manipular los datos (información)
- Consta de tres tipos de elementos:
 - (1) Unidades funcionales: ALU, desplazadores, etc
 - (2) Elementos de almacenamiento: registros
 - (3) Buses y multiplexores
- Su funcionamiento es gobernado por las líneas de control procedentes de la **unidad de control**.
- Avisa de sucesos especiales sucedidos durante las operaciones (cero, signo, desbordamiento, etc).
- Implementa el ISA establecido (varias uArquitecturas).
- **Tipos :**
 - ✓ **General**
 - ✓ **Dedicado (p.e.: solo suma)**

Tipos de Camino de datos

➤ Clasificación según forma de almacenar datos establecido en ISA

☐ **Acumulador**

☐ **Memoria-Memoria**

☐ **Memoria-Registro**

☐ **Registro-Registro (Load-Store)**

☐ **Pila**

$$C = A + B$$

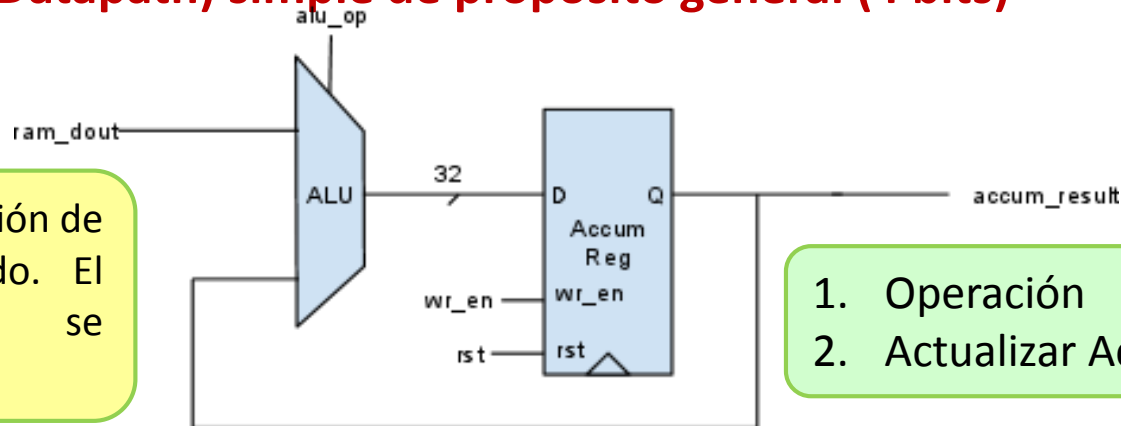
Acumulador	Memoria-Memoria	Registro-Memoria	Pila	Registro (load-store)
			Push A	Load r1,A
Load A	Add C,B,A	Load r1,A	Push B	Load r2,B
Add B		Add r1,B	Add	Add r3,r1,r2
Store C		Store C,r1	Pop C	Store C,r3

Tipos de Camino de datos

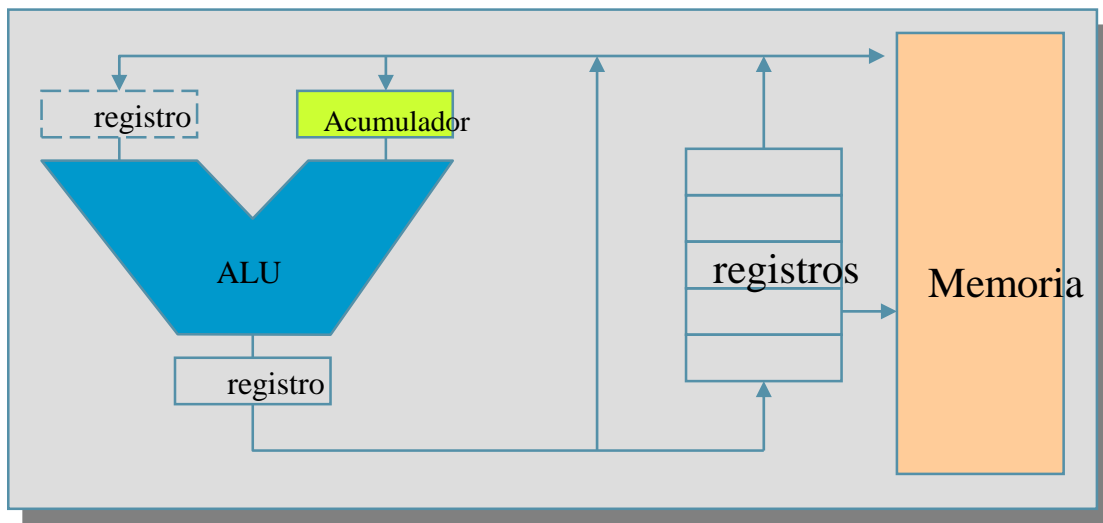
Camino de datos (Datapath) simple de propósito general (4 bits)

Acumulador

Hay que indicar ubicación de un segundo operando. El resultado también se almacena en el Acc.



1. Operación
2. Actualizar Acc/FZ

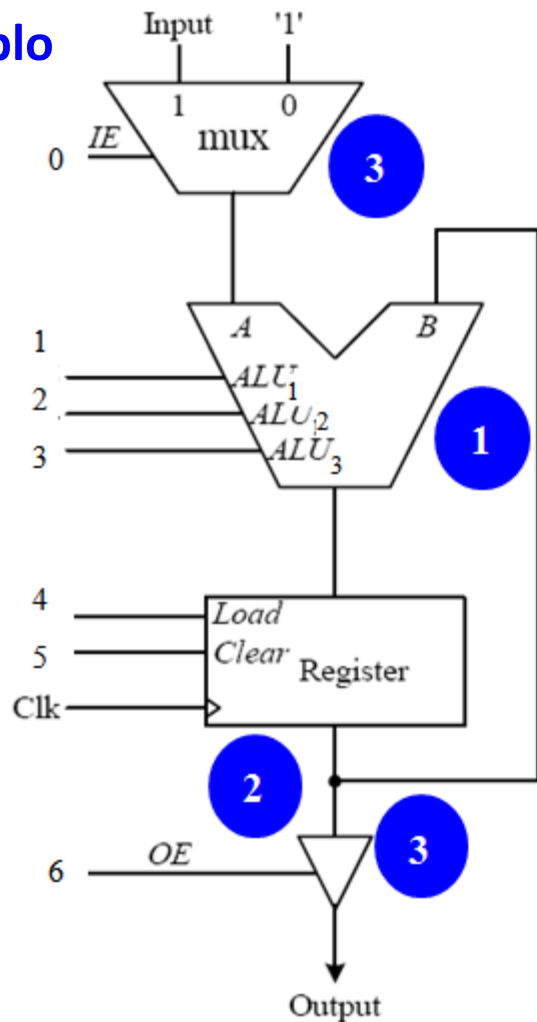


1. Operación
2. Actualiza Reg
3. Actualizar Acc/FZ

Tipos de Camino de datos

Camino de datos (Datapath) simple de propósito general (4 bits)

Ejemplo



ALU_3	ALU_2	ALU_1	Operation
0	0	0	Pass through A
0	0	1	$A \text{ AND } B$
0	1	0	$A \text{ OR } B$
0	1	1	$\text{NOT } A$
1	0	0	$A + B$
1	0	1	$A - B$
1	1	0	$A + 1$
1	1	1	$A - 1$

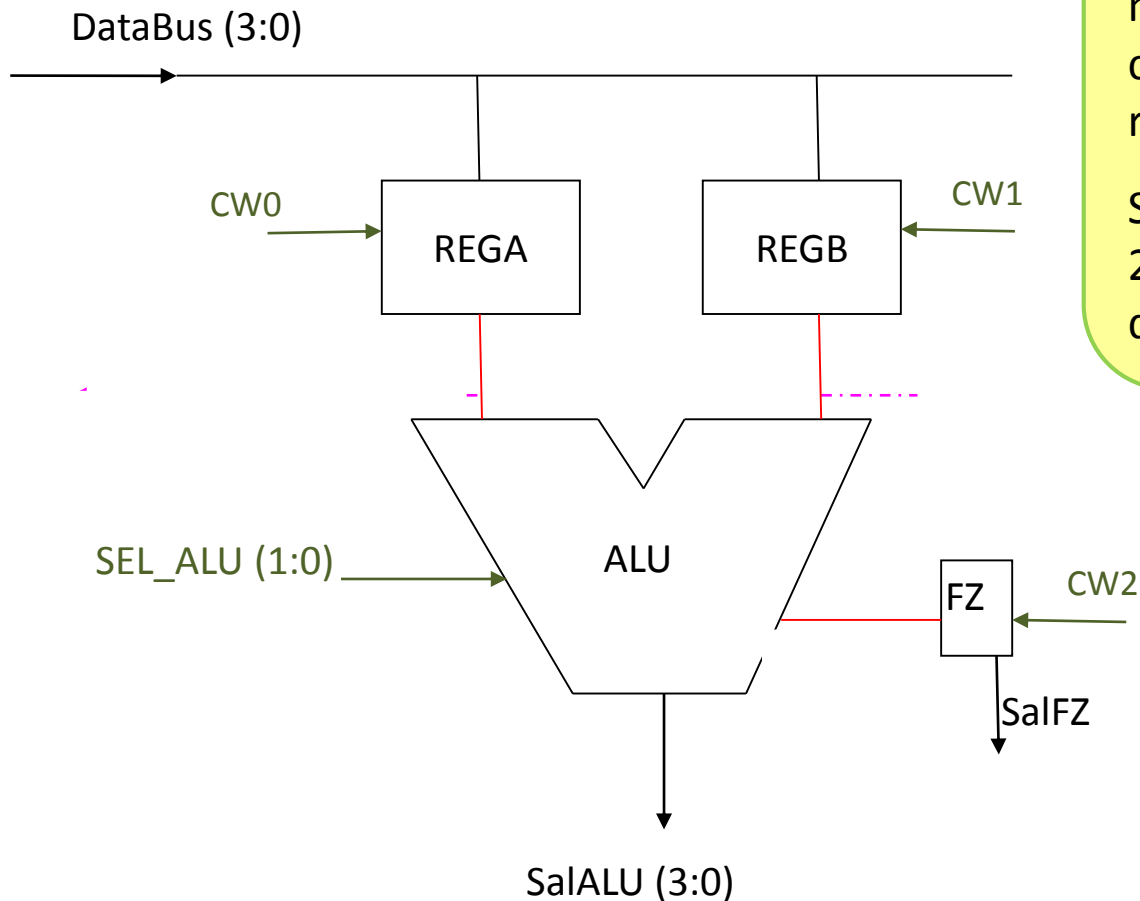
(b)

Fuente: Microprocessor Design: Principles and Practices With VHDL. Enoch O. Hwang

Tipos de Camino de datos

Camino de datos (Datapath) simple de propósito general (4 bits)

Memoria-memoria



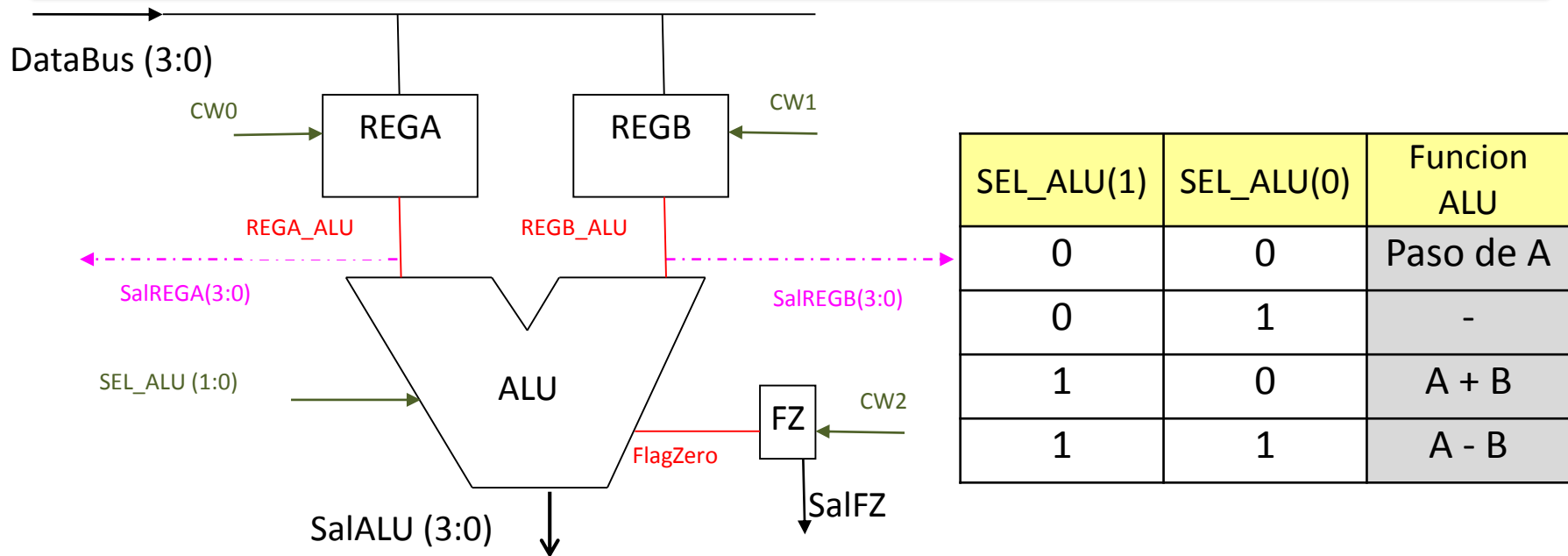
Hay que indicar dirección de memoria de A y de B, así como dónde almacenar el resultado.

Se puede hacer coincidir el 2º operando con dirección del resultado.

1. Guardar en REGA
2. Guardar en REGB
3. Seleccionar operación /Actualizar FlagZero

4.2.1. Camino de datos de DidaComp (DataPath_01)

Camino de datos de Didacomp



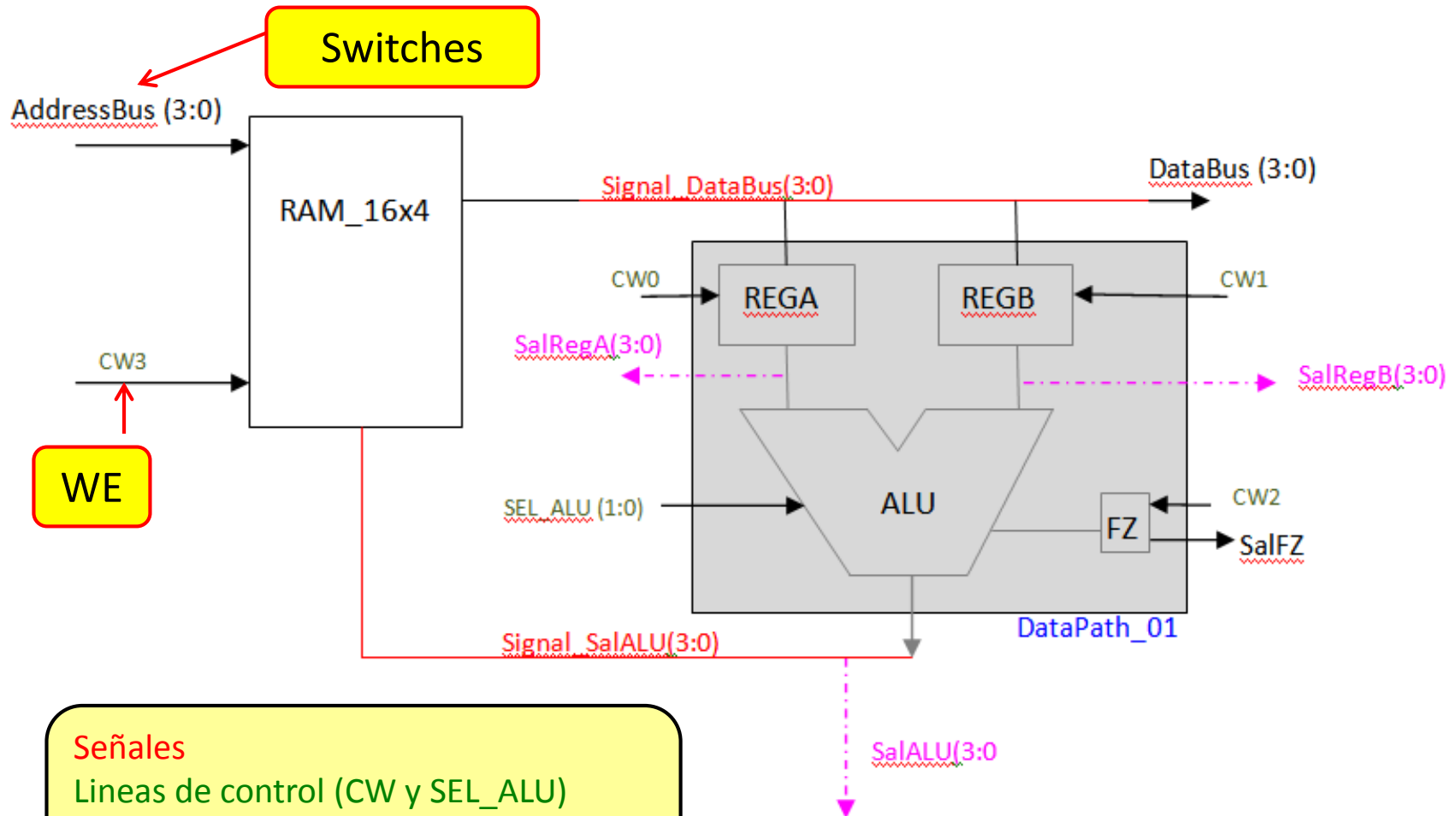
Bus de control (CW)

Acción	SEL_ALU 1-0	CW2	CW1	CW0
	Selec. Operación	Carga FZ	Carga REGB	Carga REGA
Carga OPE1 en REGA	XX	0	0	1
Carga OPE2 en REGB	XX	0	1	0
Seleccionar suma en ALU/ Actualiza FZ	10	1	0	0

4.4. Añadiendo RAM a DidaComp (RAM_DataPath_01)

Añadiendo RAM a DidaComp

Origen de los datos, memoria RAM



Añadiendo RAM a DidaComp

Ejecución de operaciones: Líneas de control (Control Word o CW)

Para ejecutar la instrucción “Suma dos datos **ADD A,B**” se necesitan tres microinstrucciones.

Bus de control					
Acción	SEL_ALU 1-0	CW3	CW2	CW1	CW0
	Selección de operación	L(bajo)/E en RAM	Carga FZ	Carga REGB	Carga REGA
Carga OPE1 desde RAM al registro A	XX	0	0	0	1
Carga OPE2 desde RAM al registro B	XX	0	0	1	0
Seleccionar suma en ALU	10	1	1	0	0

4.4. Diseño del juego de instrucciones de DidaComp ()

Juego de instrucciones de DidaComp

Formato de Instrucciones

Código de operación	Dirección OPE1	Dirección OPE2/RES
?	b7 b6 b5 b4	b3 b2 b1 b0

Instrucciones

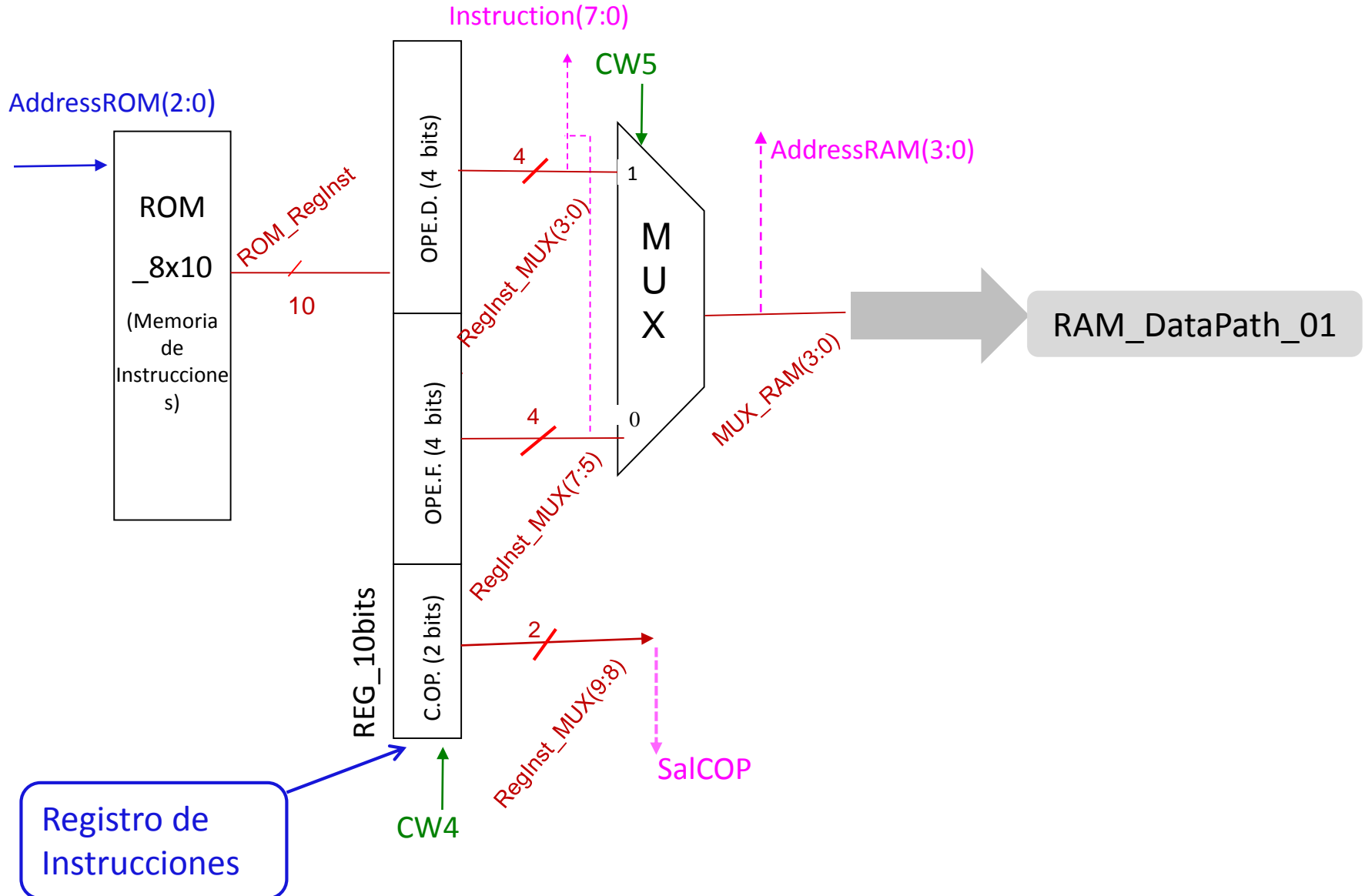
SEL_ALU(1)	SEL_ALU(0)	Funcion
0	0	Paso de A
0	1	-
1	0	A + B
1	1	A - B



Instrucción	COP	Función de la instrucción
MOV dir	00	A → dir
ADD A,B	10	A + B → B
SUB A,B	11	A-B → B

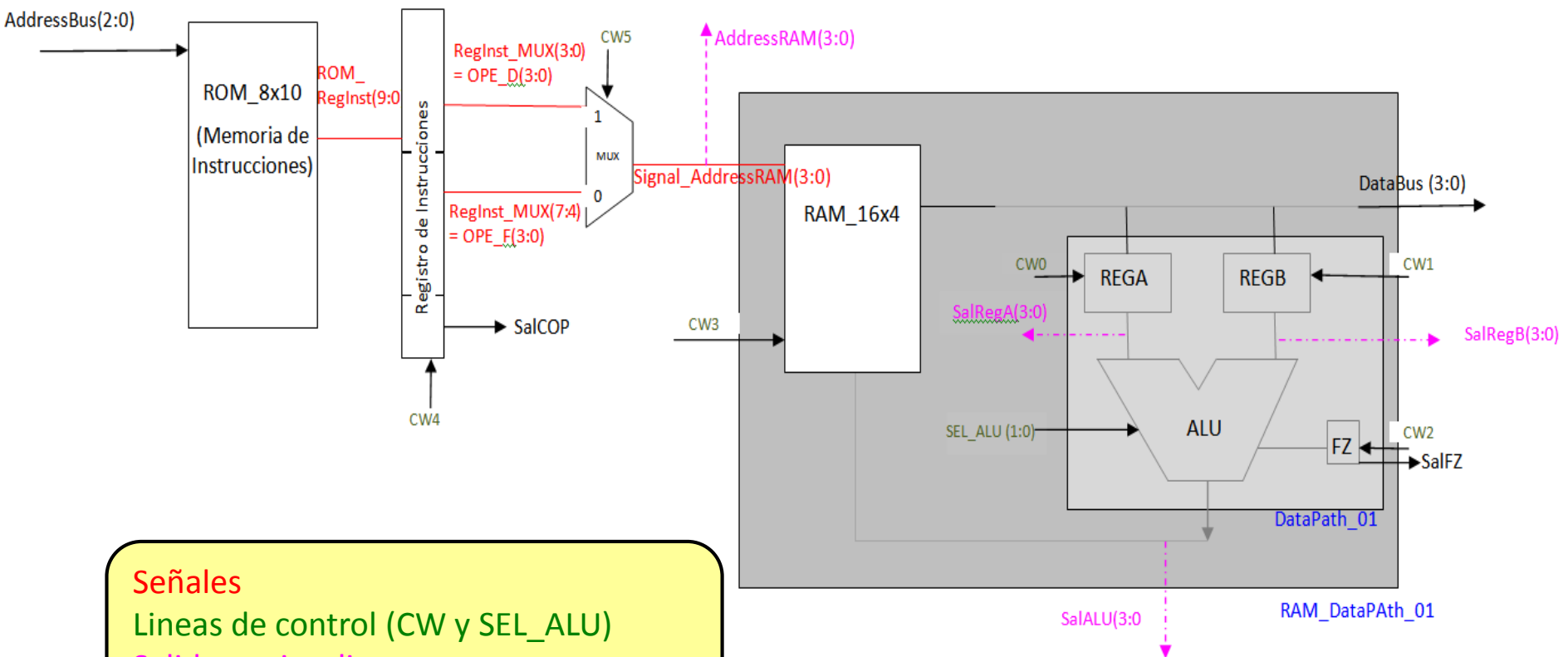
4.5. Añadiendo la memoria ROM de programa a DidaComp (ROM_RAM_DataPath_01)

Memoria de programa (ROM) y Registro de instrucciones



Memoria de programa ROM de DidaComp

ROM + Registro de instrucciones



Memoria de programa ROM de DidaComp

Ejecución de operaciones: Líneas de control (Control Word o CW)

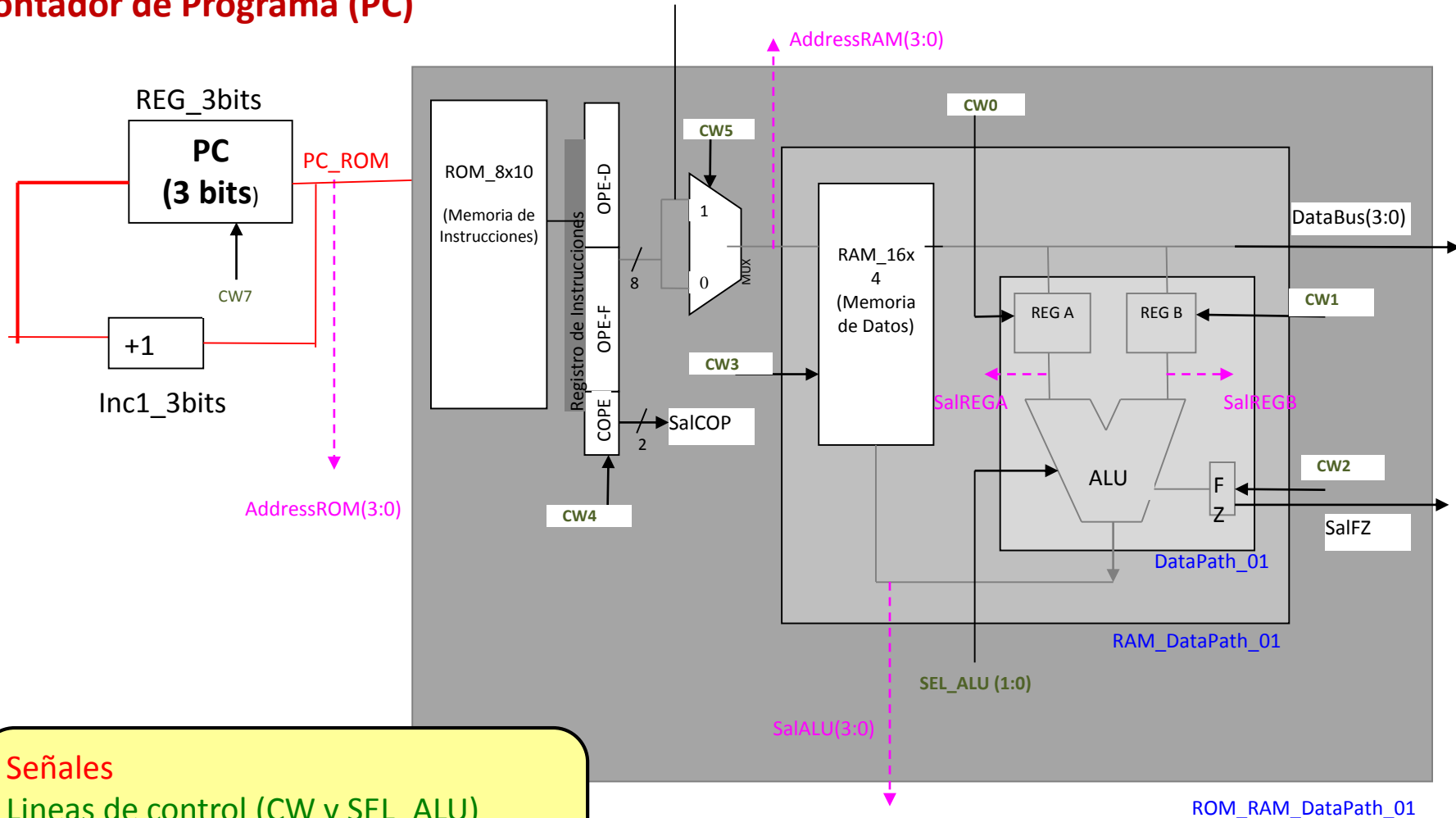
Para ejecutar la operación de la instrucción “Suma dos datos **ADD A,B**” se necesitan tres microinstrucciones.

Bus de control							
Acción	SEL_ALU 1-0	CW5	CW4	CW3	CW2	CW1	CW0
	Selección de operación	Selección de dirección para RAM	Carga de I en RegInst	Escritura en RAM	Carga FZ	Carga REGB	Carga REGA
Leer instrucción	XX	0	1	0	0	0	0
Observar COP (Decodificación)							
Carga OPE1 desde RAM al registro A	XX	0	0	0	0	0	1
Carga OPE2 desde RAM al registro B	XX	1	0	0	0	1	0
Seleccionar suma en ALU	10	1	0	1	1	0	0

4.6. Contador de programa de DidaComp

Contador de programa de DidaComp

Contador de Programa (PC)



Señales

Lineas de control (CW y SEL_ALU)

Salidas a visualizar

Puertos E/S

4.7. Estructura de salto de DidaComp

Estructura de salto de DidaComp

Formato de Instrucciones: Estructura de salto (BEQ)

Código de operación	Dirección OPE1	Dirección OPE2/RES
b9 b8	b7 b6 b5 b4	b3 b2 b1 b0

Código de operación		Dirección ROM de salto
01	X X X X	X b2 b1 b0

Instrucciones

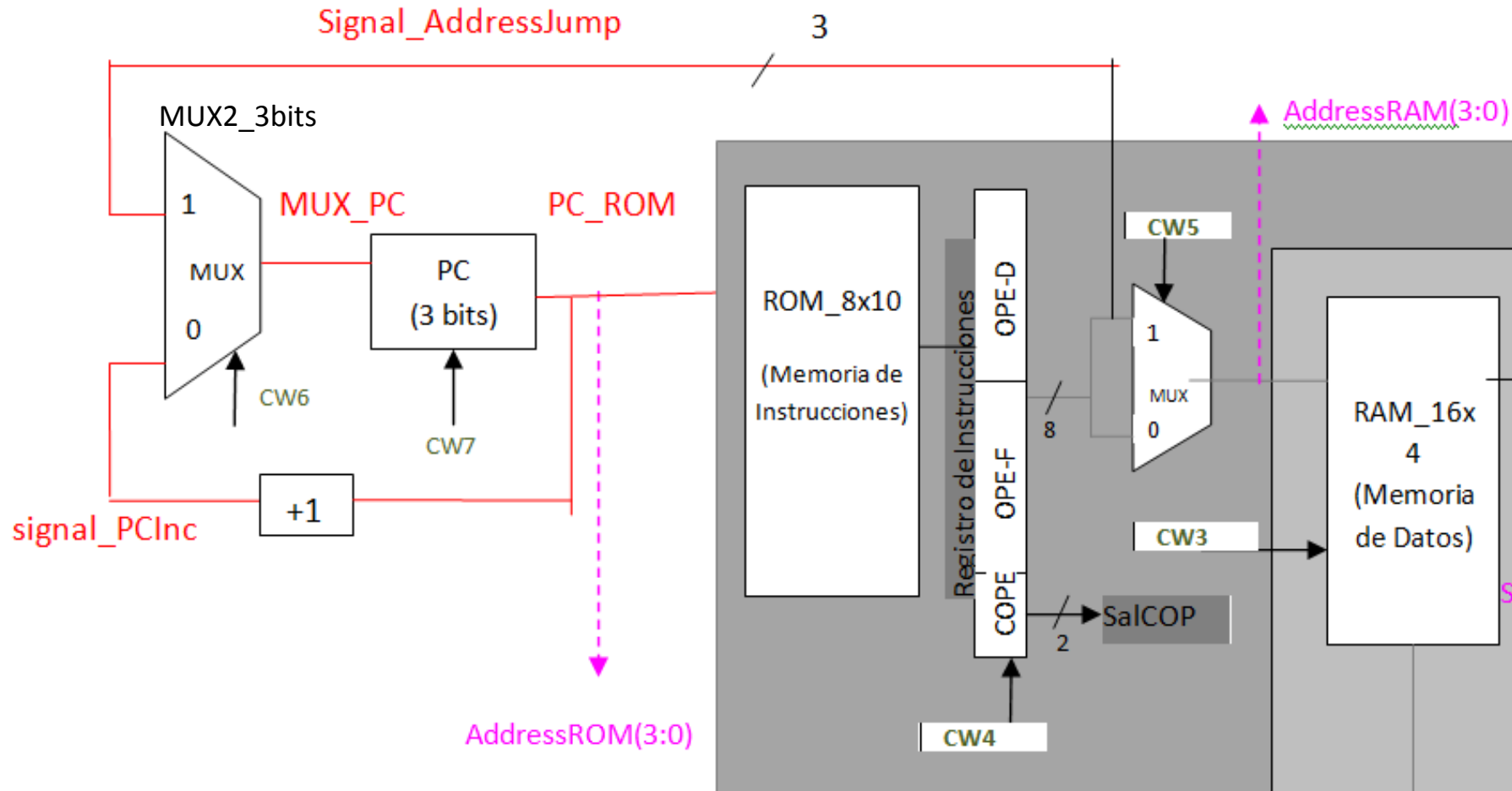
SEL_ALU(1)	SEL_ALU(0)	Funcion
0	0	Paso de A
0	1	-
1	0	A + B
1	1	A - B



Instrucción	COP	Función de la instrucción
MOV dir	00	A → dir
BEQ dir	01	Si FZ=1 Dir → PC Si FZ=0 PC+1→PC
ADD A,B	10	A + B → B
SUB A,B	11	A-B → B

Estructura de salto de DidaComp

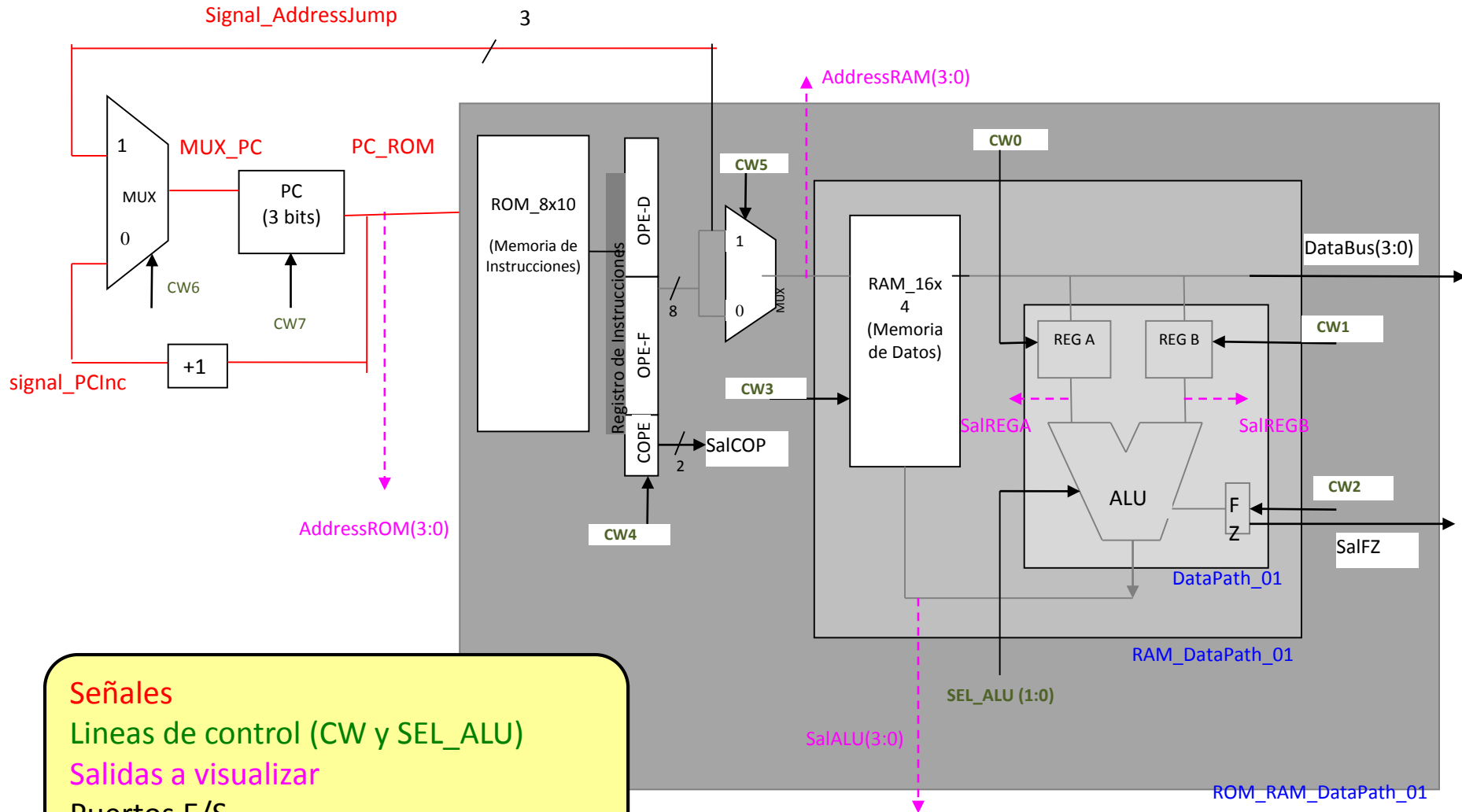
Estructura de salto



4.8. Microarquitectura de DidaComp

Microarquitectura de DidaComp

Estructura de salto+PC+RI+ROM+RAM+DataPath_01



Microarquitectura de DidaComp

Ejecución de operaciones: Líneas de control (Control Word o CW)

Suma dos datos ADD A,B

Bus de control									
Acción	SEL_ALU 1-0	CW7	CW6	CW5	CW4	CW3	CW2	CW1	CW0
ESTADO	Selección de operación	Actualiza PC	Origen PC	Dir RAM	Carga RI	Escribe RAM	Carga FZ	Carga REGB	Carga REGA
Leer instrucción Incrementar PC	XX	1	0	0	1	0	0	0	0
Observar COP (Decodificación)									
Carga OPE1 desde RAM al registro A	XX	0	0	0	0	0	0	0	1
Carga OPE2 desde RAM al registro B	XX	0	0	1	0	0	0	1	0
Seleccionar suma en ALU y actualizar FZ	10	0	0	1	0	1	1	0	0