Universidad Rey Juan Carlos

Grado en Ingeniería de Computadores

Tecnología de Computadores

Práctica 10

Implementación de un microprocesador sencillo en VHDL

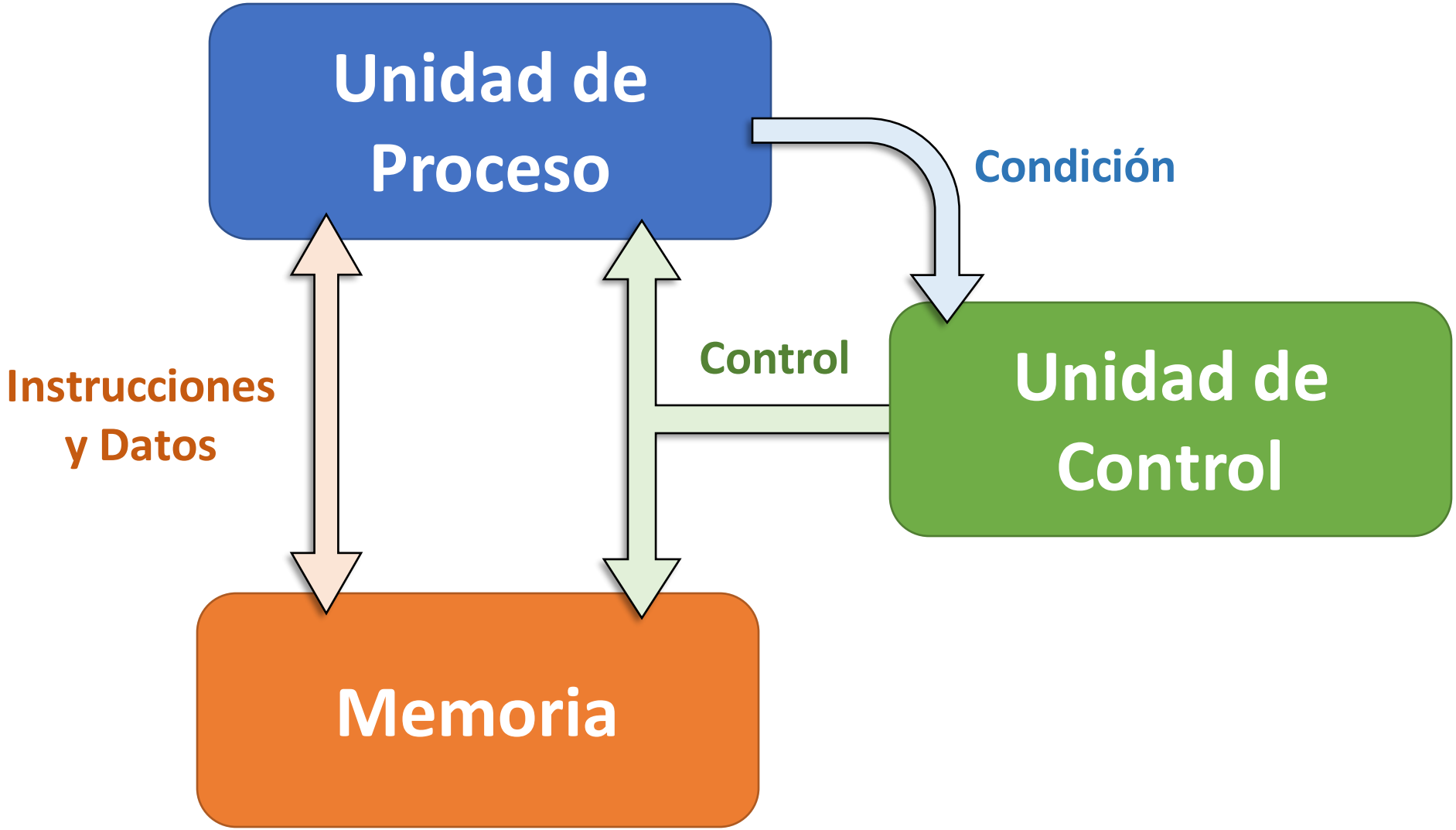
(Ruta de datos y unidad de control)

Agenda

Universidad Rey Juan Carlos

1. Arquitectura del microprocesador
   - Estructura
   - Memoria
   - Repertorio de Instrucciones
2. Ruta de datos del microprocesador
3. Unidad de control del microprocesador
4. Síntesis y simulación del microprocesador en VHDL

1. Arquitectura del microprocesador



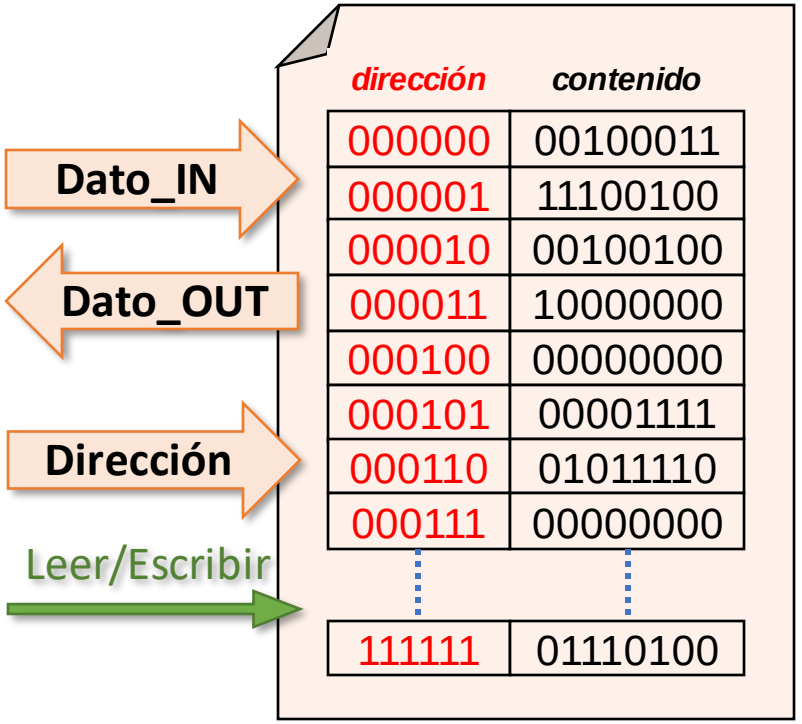
1. Arquitectura del microprocesador

Memoria

- 64 BYTES
  - 6 bits de dirección → 64 palabras de memoria
  - 8 bits (1 byte) para cada palabra de memoria
- Para **LEER**:
  - Dar una dirección
  - Dar la orden de leer (Leer/Escribir = 0)

  Y tendremos el dato en Dato_OUT
- Para **ESCRIBIR**:
  - Dar una dirección
  - Poner un dato en Dato_IN
  - Dar la orden de escribir (Leer/Escribir = 1)

  Y tendremos el dato se grabará en la dirección



1. Arquitectura del microprocesador

- 1 palabra de memoria puede ser:
  - una instrucción máquina
  - un dato

- Un programa = secuencia de instrucciones máquina.

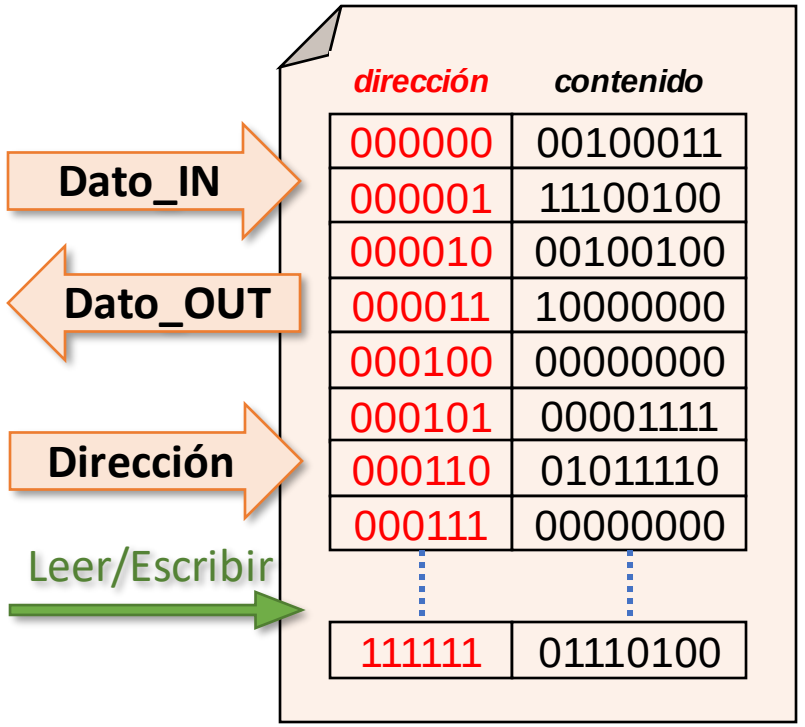
  → **Hay que diseñar un conjunto de instrucciones**

- Las instrucciones se ejecutan una a una en la unidad de proceso (**U.P.**)

  → **Hay que diseñar una U.P.**

- La unidad de control (**U.C.**) es la encargada de orquestar el flujo de los bits a través de todo el sistema para que cada instrucción se ejecute bien.

  → **Hay que diseñar una U.C.**



1. Arquitectura del microprocesador

- **MCPU**: minimal 8-bit CPU diseñada por Tim Böscke
- Se crea un **repertorio de instrucciones** (ISA*) sencillo y reducido pero que sea funcional
- Utiliza un registro acumulador de 8 bits (+1 adicional para el acarreo)

Instrucción (*8* bits)	**Opcode** (*2* bits)	**Operación**	**Descripción**
00AAAAAA	00	**NOR**	Acc ← Acc NOR mem[AAAAAA]
01AAAAAA	01	**ADD**	Acc ← Acc + mem[AAAAAA] Actualizo el valor del carry
10AAAAAA	10	**STA** (store)	Mem[AAAAAA] ← Acc
11DDDDDD	11	**JCC** (salto condicional)	PC ←DDDDDD si carry = '0' Si carry = '1' sólo lo pongo a '0'

Utiliza los dos bits más significativos de la instrucción para el código de operación

* *ISA = Instruction Set Architecture*

1. Arquitectura del microprocesador

- Con este reducido conjunto de instrucciones se pueden crear otras más complejas denominadas macros. Por ejemplo:

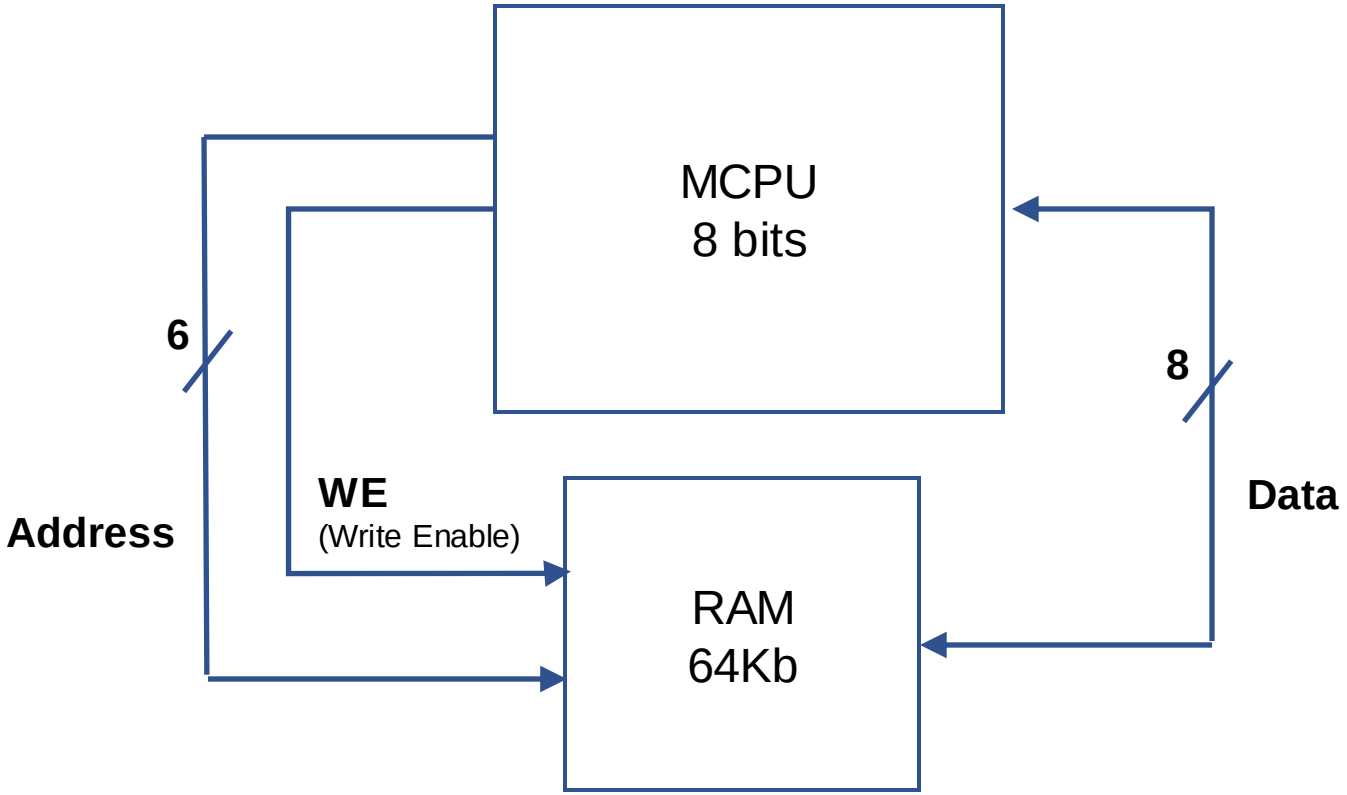
Macro	Código ensamblador	Descripción
CLR	NOR 0xFF	Borra el contenido del acumulador
NOT	NOR 0x00	Invierte el valor del acumulador
JMP	JCC dst, JCC dst	Salto incondicional a dst
…	…	…

¿Cómo se podría implementar una operación de resta (SUB)?

Acc – mem → NOR 0x00, ADD mem, ADD 0x01

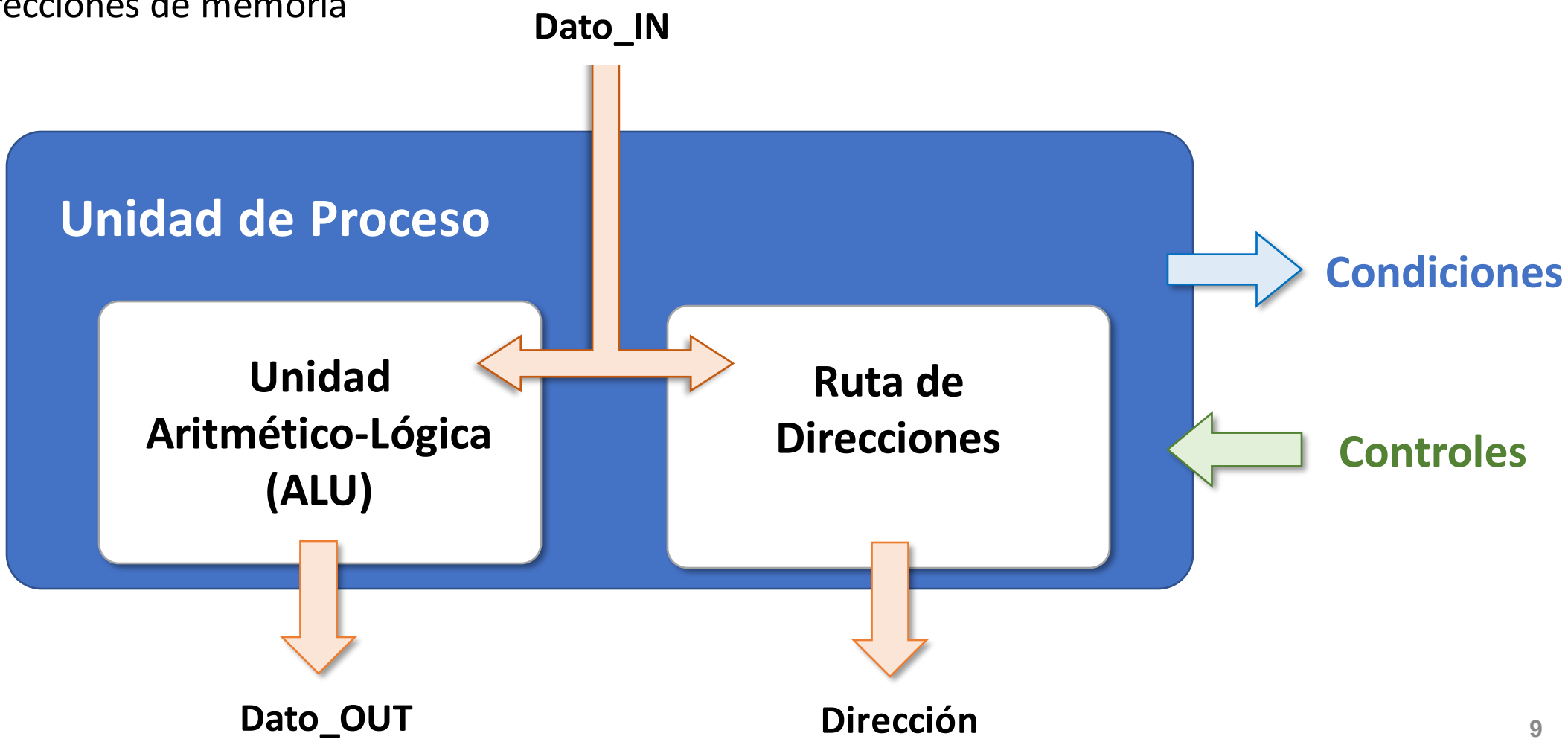
1. Arquitectura del microprocesador

- El **contador de programa** (PC) será de 6 bits porque 2 de los 8 bits son para el opcode, por lo que se podrán direccionar $2^6 = 64$ palabras de memoria de 8 bits cada una
- La **memoria RAM** del micro será de 64Kb y estará compartida entre datos e instrucciones

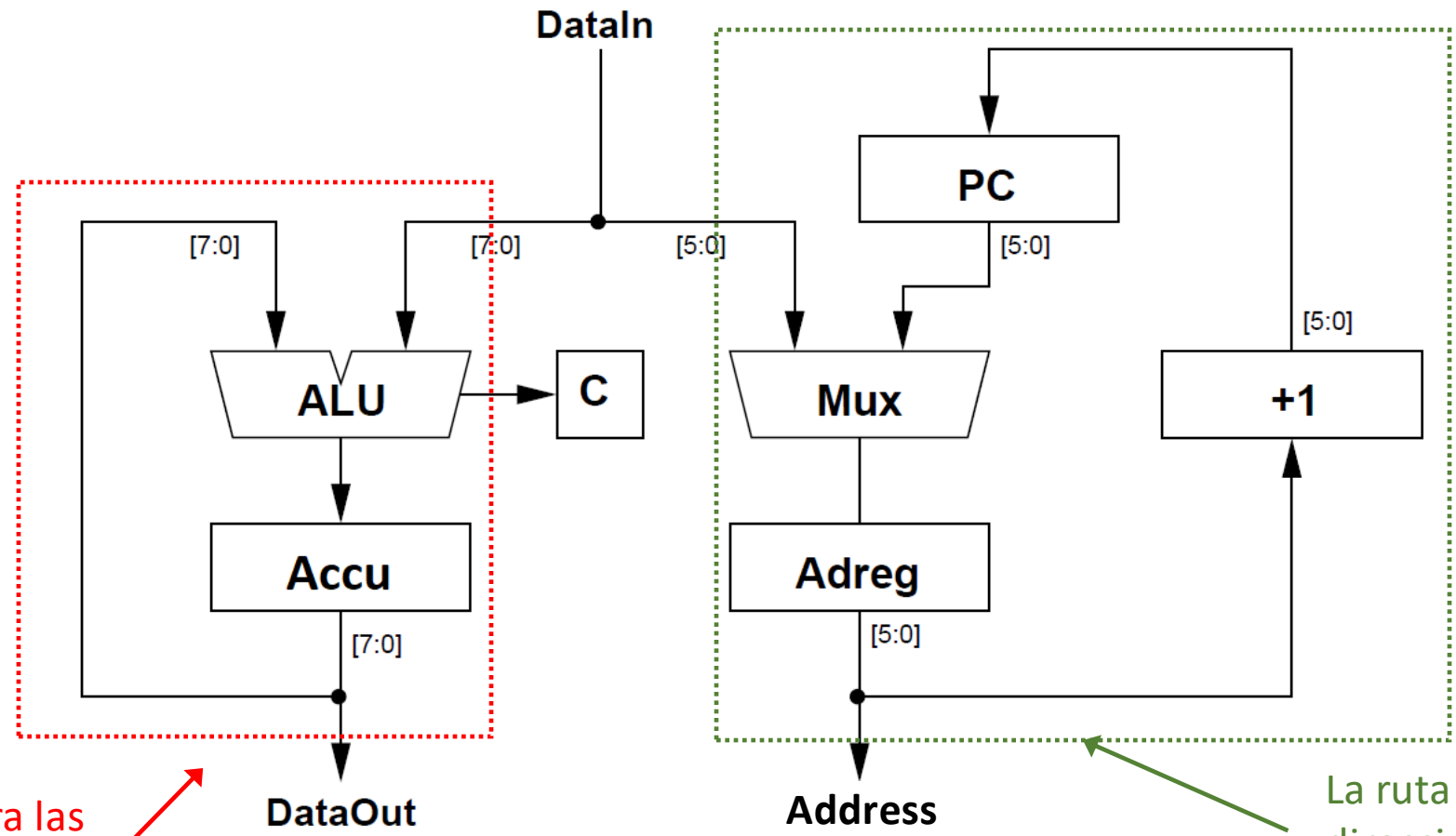


2. Ruta de datos microprocesador

- La ruta de datos está dividida en dos partes:
  - Unidad aritmético-lógica (ALU)
  - Direcciones de memoria



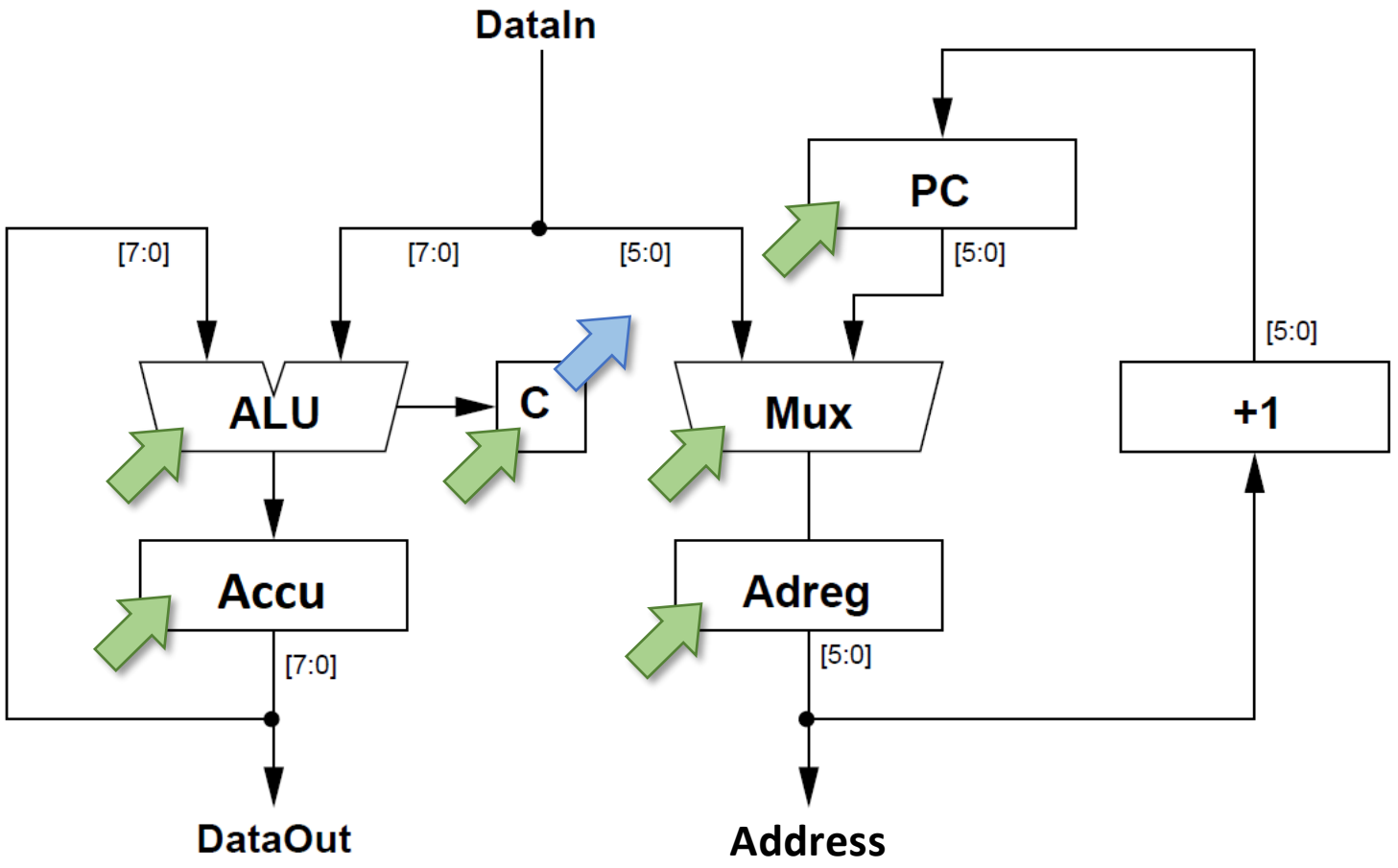
2. Ruta de datos microprocesador



La ALU no genera las direcciones de memoria.

La ruta de direcciones utiliza los bits [5:0] del dato/instrucción

2. Ruta de datos microprocesador



Condiciones

Controles

3. Unidad de control del microprocesador

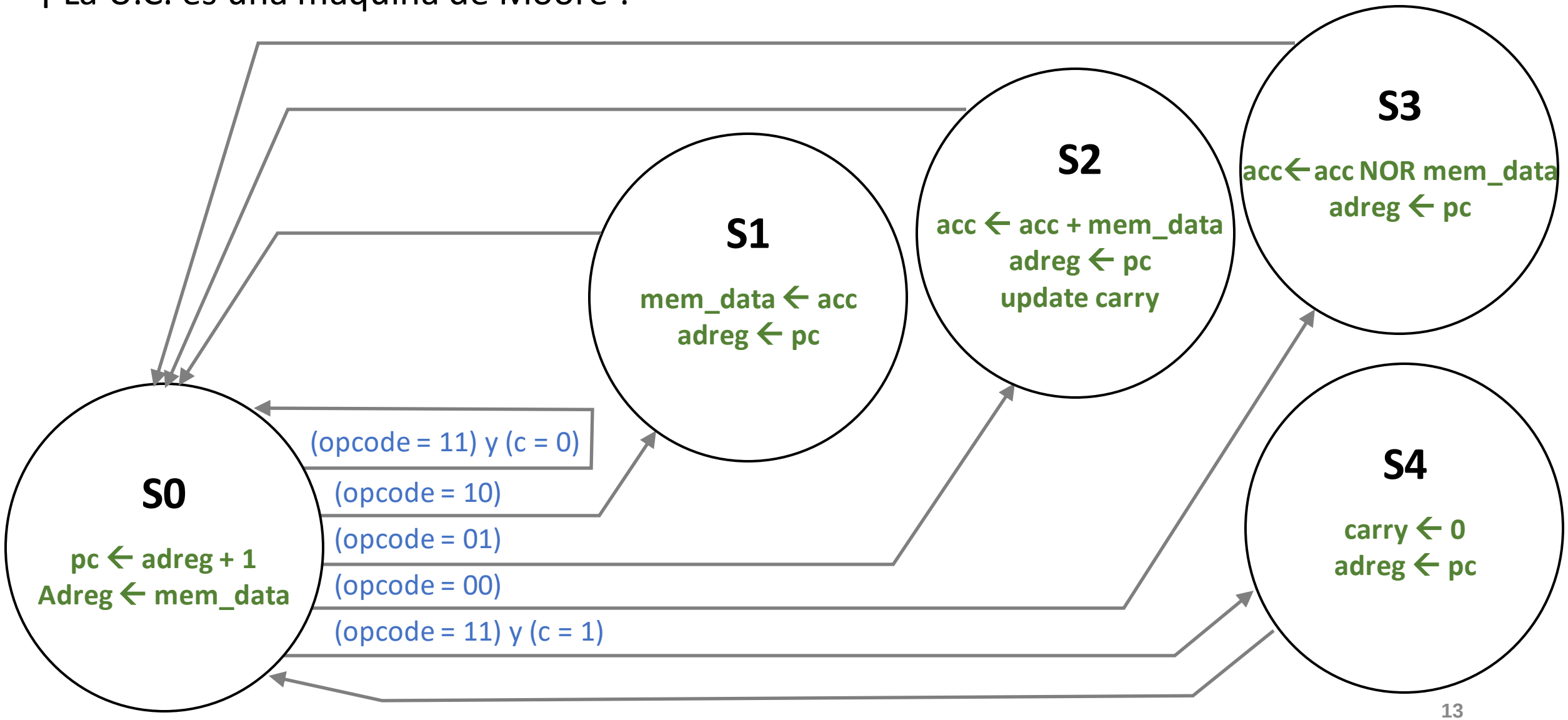
Universidad Rey Juan Carlos

Se utilizan **5 estados** para controlar el micro y la escritura/lectura de memoria

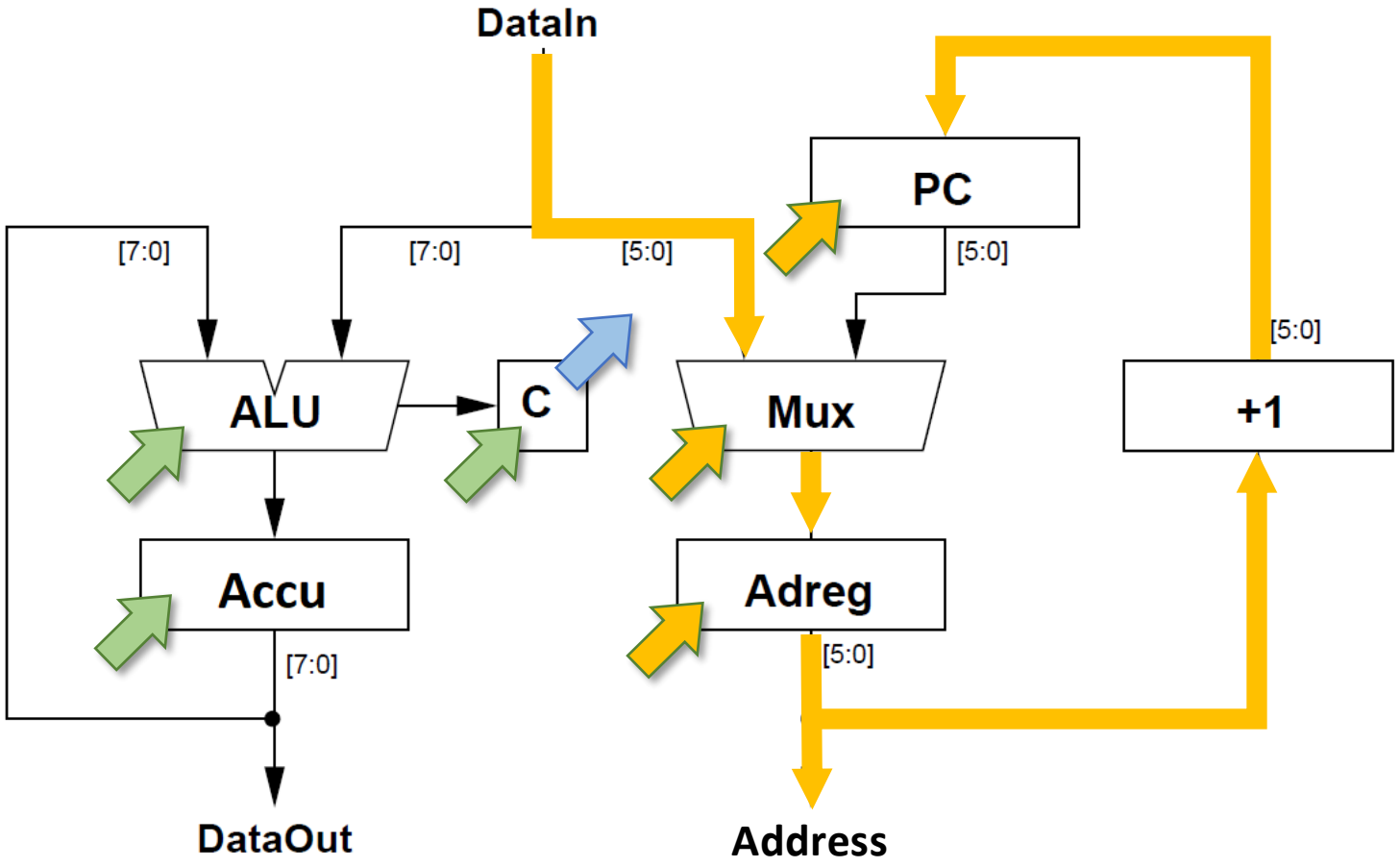
Estado	Descripción	Siguiente estado	Salidas
S0 - Fetch	Recibo y decodifico una nueva instrucción	**S0** si opcode = 11 y c = 0 **S1** si opcode = 10 **S2** si opcode = 01 **S3** si opcode = 00 **S4** si opcode = 11 y c = 1	pc ← adreg + 1 adreg ← mem_data
S1 - Write	Escribo en memoria el valor del acumulador	**S0**	mem_data ← acc adreg ← pc
S2 - ADD	Sumo y actualizo el carry	**S0**	acc ← acc + mem_data adreg ← pc update carry
S3 - NOR	Operación de NOR	**S0**	acc ← acc NOR mem_data adreg ← pc
S4 - Clear carry	Borro el carry	**S0**	carry ← 0 adreg ← pc

3. Unidad de control del microprocesador

¡ La U.C. es una maquina de Moore !



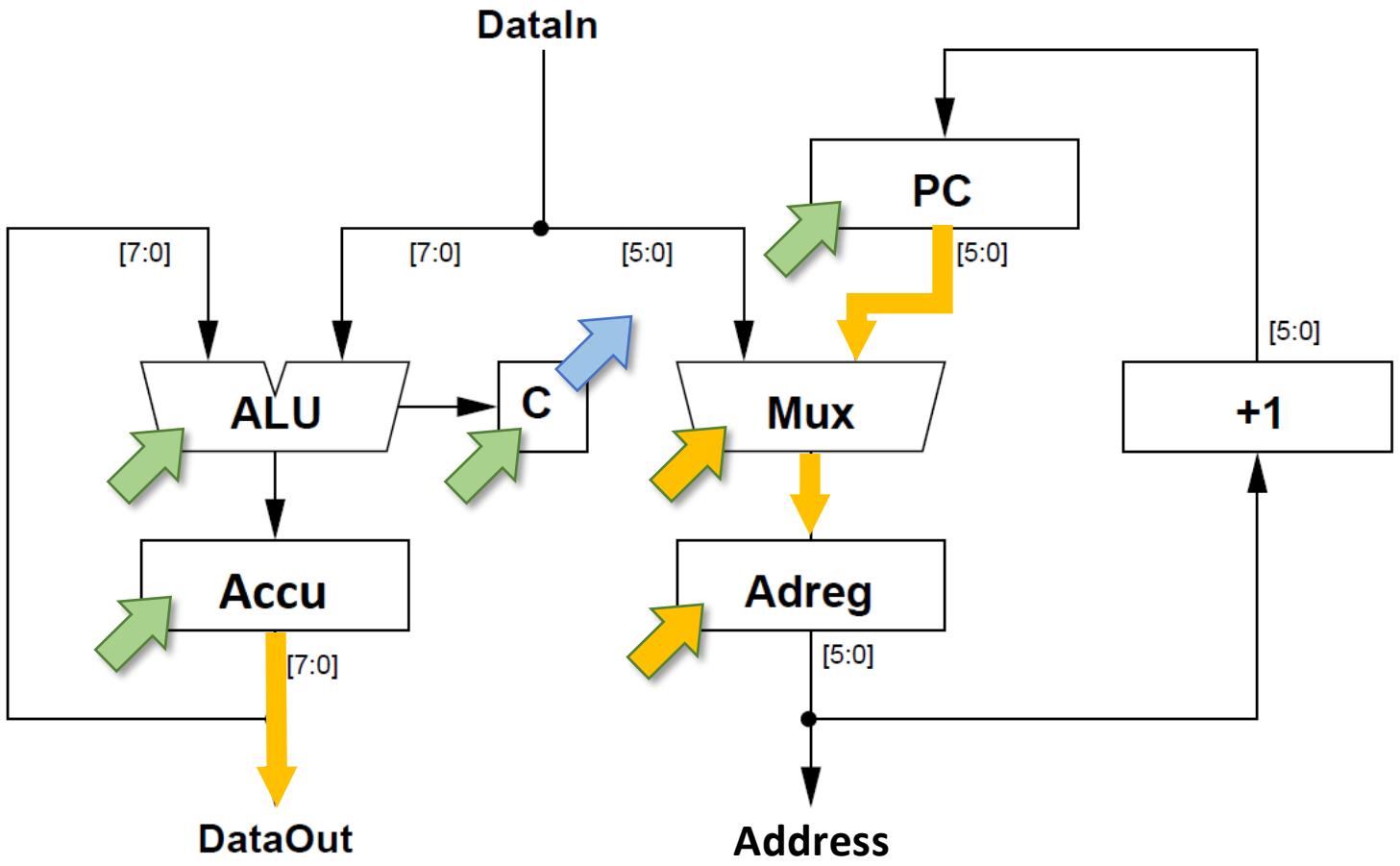
3. Unidad de control del microprocesador



Estado	Descripción
S0 - Fetch	Recibo y decodifico una nueva instrucción

Salidas
pc ← adreg + 1 adreg ← mem_data

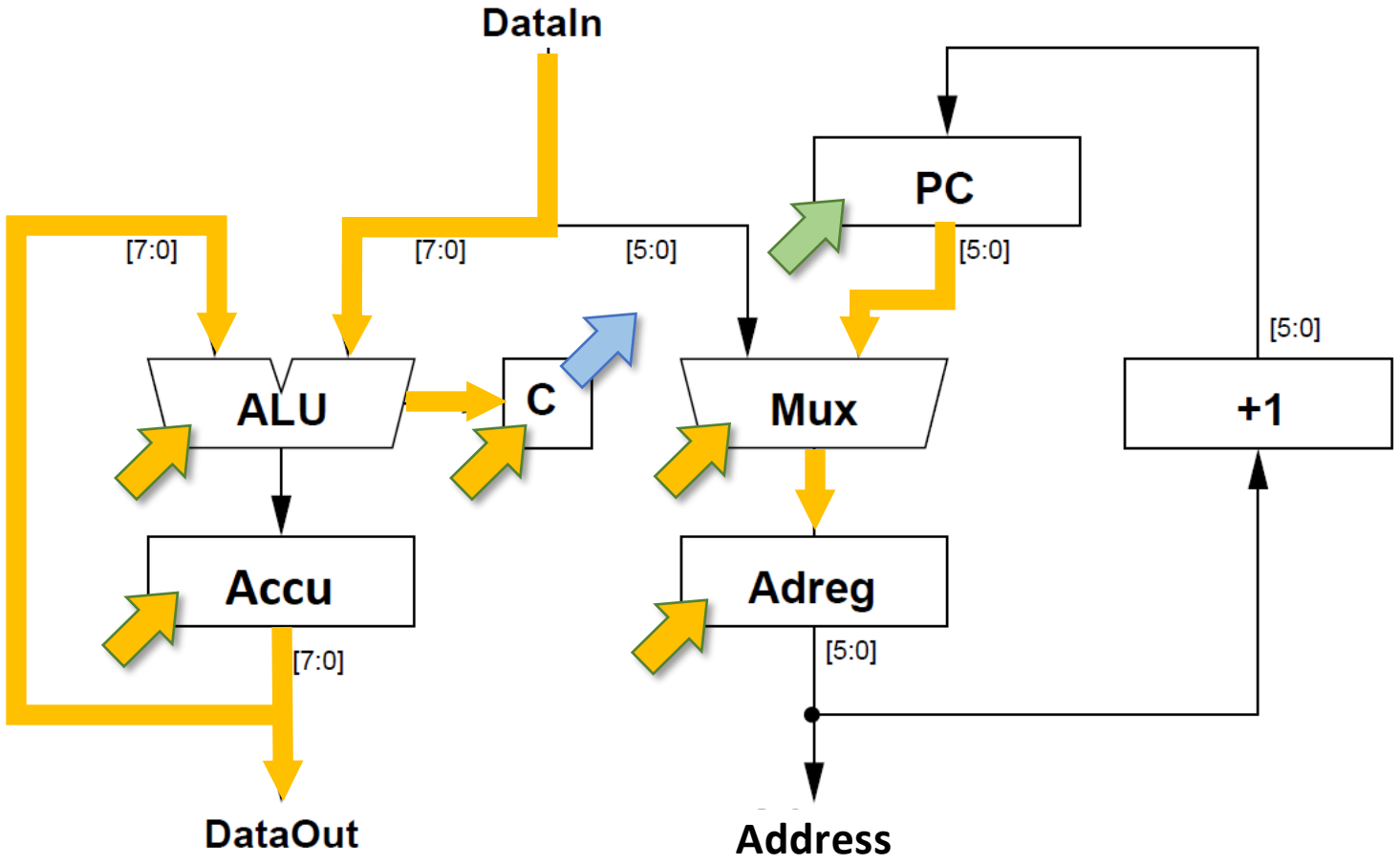
3. Unidad de control del microprocesador



Estado	Descripción
S1 - Write	Escribo en memoria el valor del acumulador

Salidas
mem_data ← acc adreg ← pc

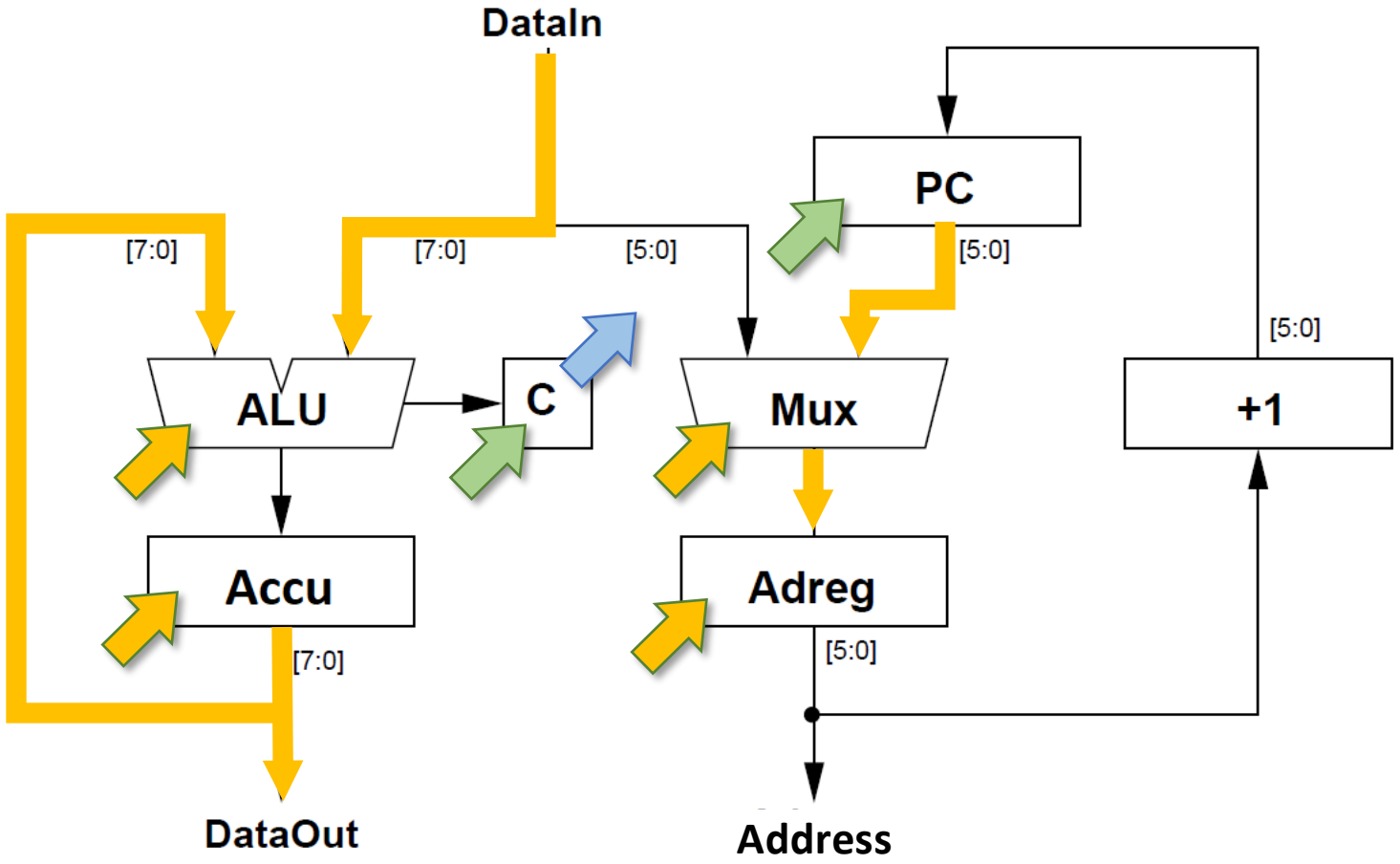
3. Unidad de control del microprocesador



Estado	Descripción
S2 - ADD	Sumo y actualizo el carry

Salidas
accu ← accu + mem_data adreg ← pc, update carry

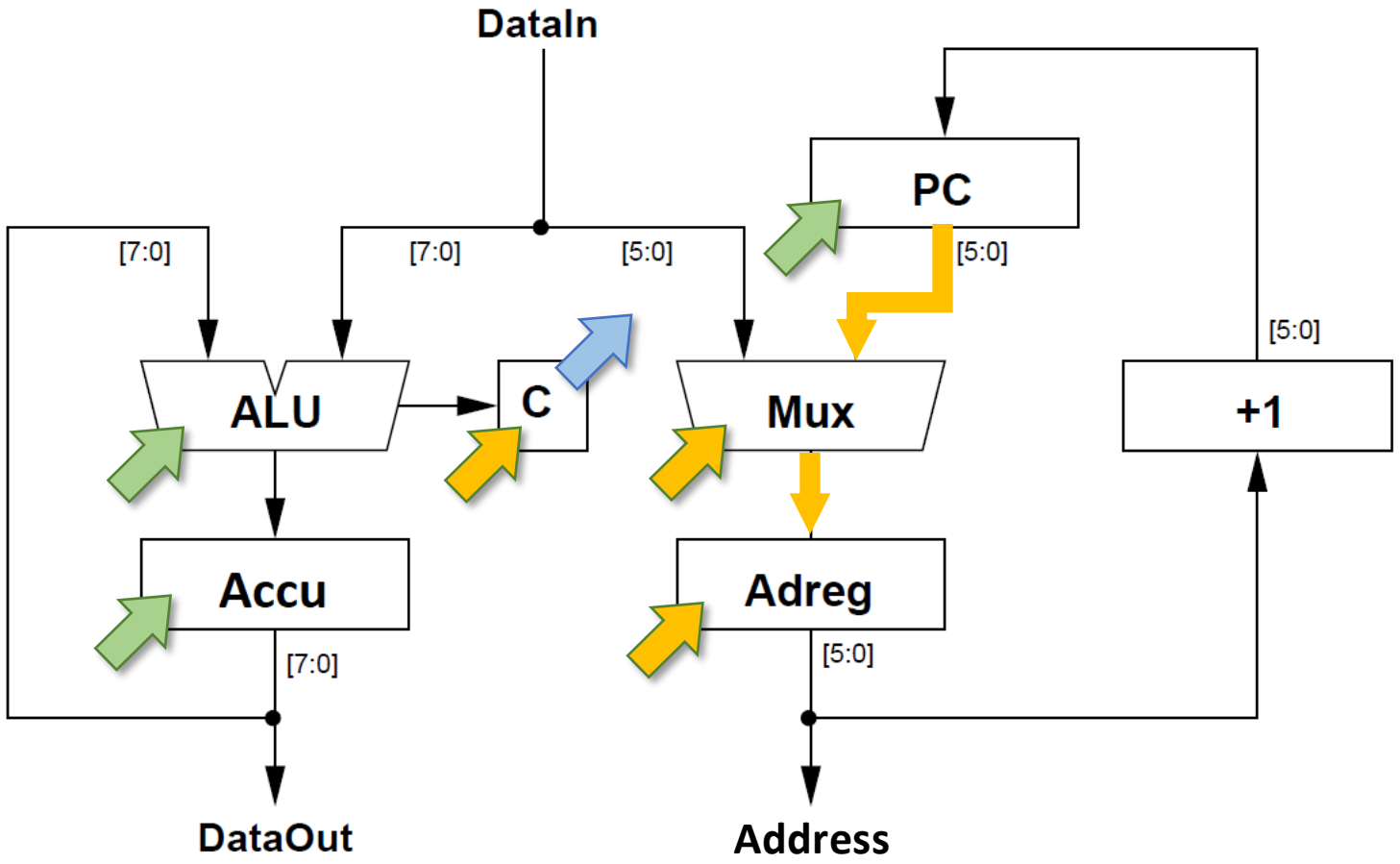
3. Unidad de control del microprocesador



Estado	Descripción
S3 - NOR	Operación de NOR

Salidas
accu ← accu NOR mem_data adreg ← pc

3. Unidad de control del microprocesador



Estado	Descripción
S4 - Clear carry	Borro el carry
Salidas	
carry ← 0 adreg ← pc	

4. Síntesis y simulación del microprocesador en VHDL

- La entidad del microprocesador es:

```
entity mcpu is
 port (
  clk   : in std_logic;                          -- reloj
  rst   : in std_logic;                          -- reset
  data  : inout std_logic_vector (7 downto 0); -- bus de datos de entrada y salida
  addr  : out std_logic_vector (5 downto 0);    -- dirección de memoria
  oe    : out std_logic;                        -- habilitación de salida
  we    : out std_logic                         -- habilitación de escritura
 );
end mcpu;
```

Ejercicios

1. Completar el código mcpu.vhd con la información faltante de la ruta de datos

2. Completar el código mcpu.vhd con la información faltante de la unidad de control

3. Crear una entidad denominada top.vhd que instancie mcpu.vhd y sram.vhd

4. Crear un testbench para simular top.vhd. Observe el funcionamiento del microprocesador

+ Compilar y simular el resto de códigos ensamblador que se proporcionan en la práctica.

Pregunta:

¿Podríamos ejecutar en el microprocesador programas que utilicen otras instrucciones de ensamblador distintas a las 4 iniciales? ¿por qué?