Grado en
Ingeniería Informática

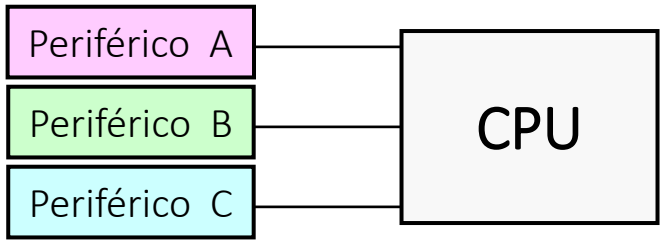
Tema 4:
Subsistema de E/S

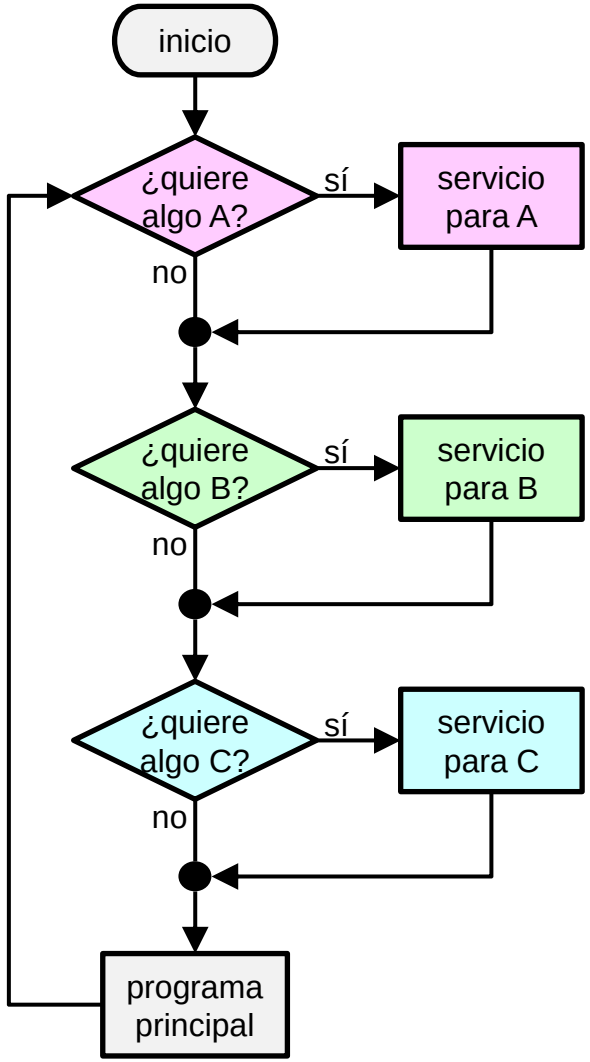
Unidad 3:
Sincronización por consulta de estado / interrupciones

Rafael Casado González
Rosa María García Muñoz
María Teresa López Bonal
Universidad de Castilla–La Mancha

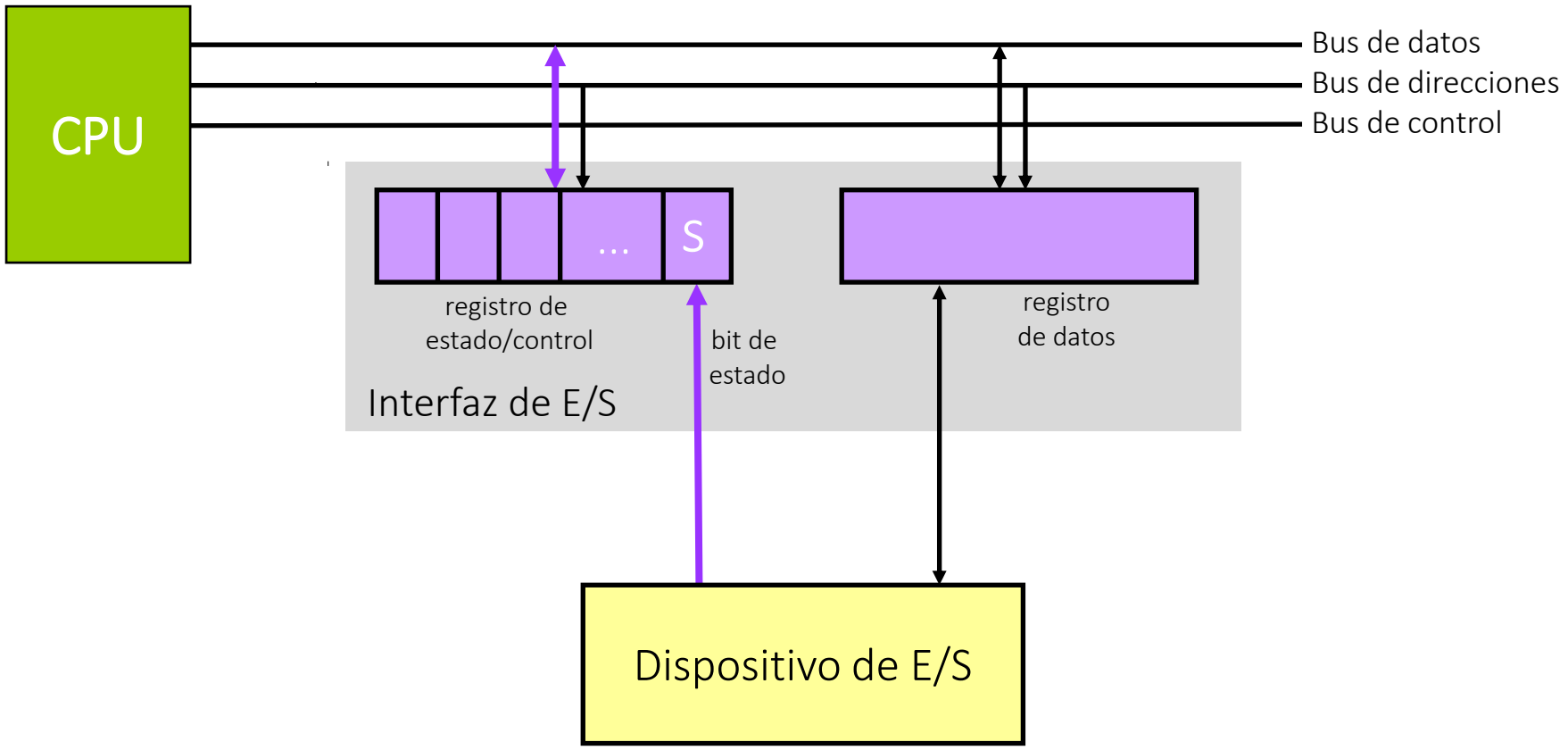
E/S por consulta de estado
(o polling)

- Los interfaces de E/S tienen un bit de estado
  - que indica si el periférico está preparado para la transferencia
- La CPU debe
  - consultar periódicamente ese bit
  - transmitir cuando esté activo





E/S por consulta de estado



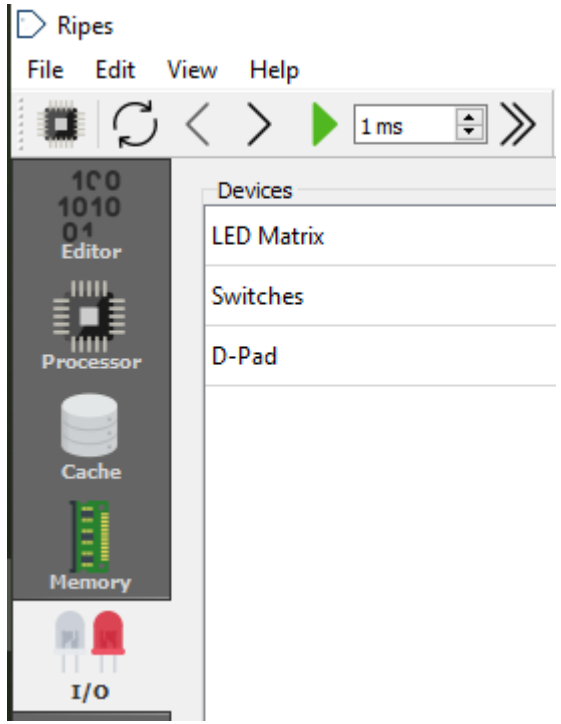
E/S por consulta de estado
Inconvenientes

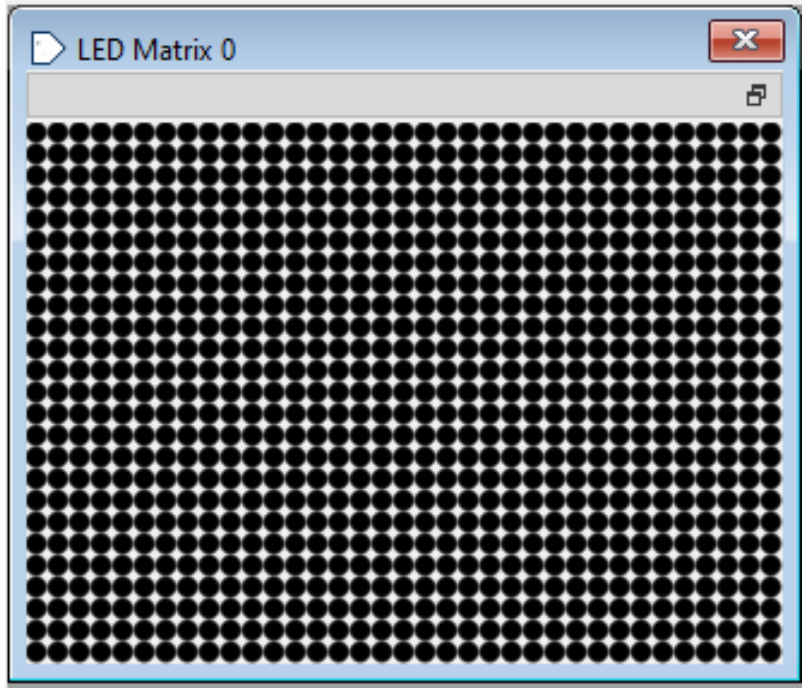
- La CPU no sabe cuándo el periférico estará preparado para enviar o recibir datos
  - Lo consulta continuamente
  - Mientras realiza consultas, la CPU no realiza otro trabajo

- El periférico puede estar preparado, y sin embargo debe esperar al próximo período de consulta

E/S por consulta de estado
Simulador Ripes

Máximo de
256x256 píxeles





LED Matrix 0

Each LED maps to a 24-bit register storing an RGB color value, with B stored in the least significant byte.
The byte offset of the LED at coordinates (x, y) is:
offset = (y + x*N_LEDS_ROW) * 4

Parameters

Name	Value
Height	25
Width	35
LED size	8

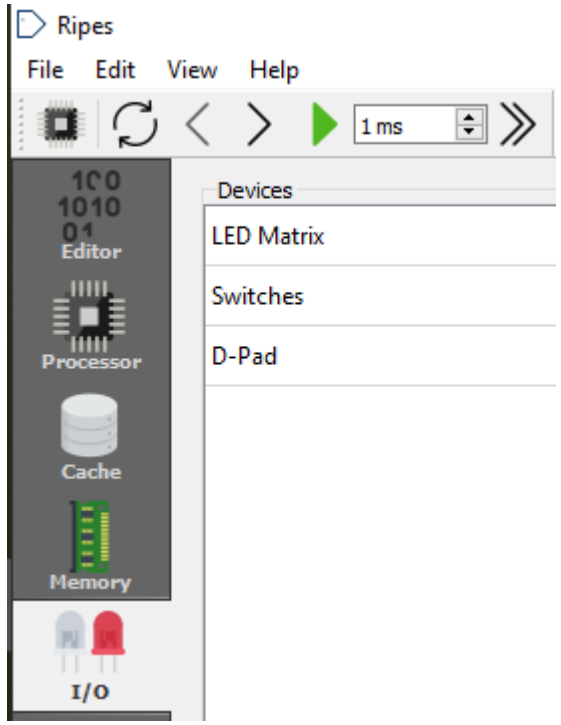
Register map

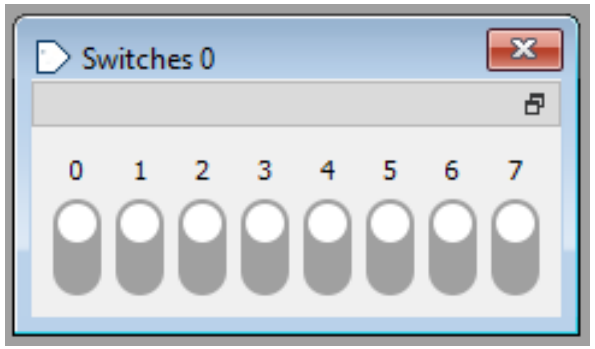
Name	Address	R/W?	Size
LED_0	0x0	R/W	24
LED_1	0x4	R/W	24
LED_2	0x8	R/W	24
LED_3	0xc	R/W	24

Exports

```
#define LED_MATRIX_0_BASE (0xf0000000)
#define LED_MATRIX_0_SIZE (0xdac)
#define LED_MATRIX_0_WIDTH (0x23)
#define LED_MATRIX_0_HEIGHT (0x19)
```

E/S por consulta de estado
Simulador Ripes





Máximo de
32 conmutadores

Switches 0

Each switch maps to a bit in the memory-mapped register of the peripheral.
switch n = bit n

Parameters

Name	Value
# Switches	8

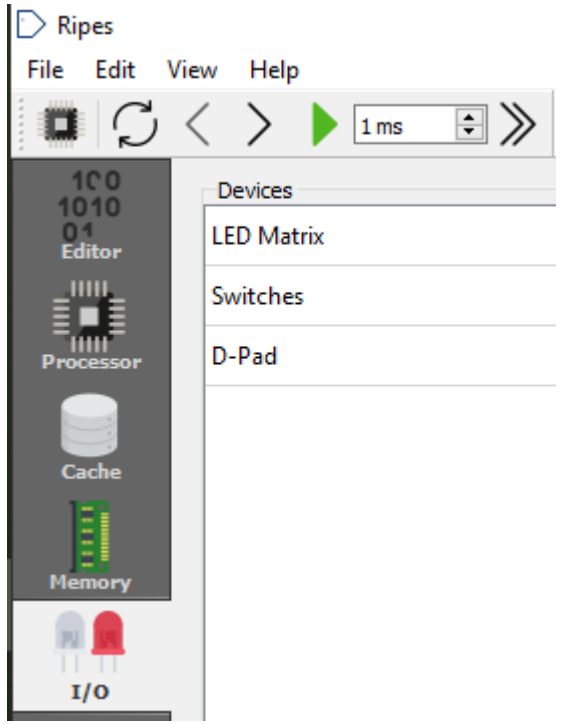
Register map

Name	Address	R/W?	Size
Switches	0x0	R	8

Exports

```
#define SWITCHES_0_BASE
(0xf0000000)
#define SWITCHES_0_SIZE (0x4)
#define SWITCHES_0_N (0x8)
```

E/S por consulta de estado
Simulador Ripes



D-Pad 0

Each button maps to a 32-bit register, with the least-significant bit indicating the state of the button.

If the D-pad window is in focus, the buttons may be pressed using the "WASD" keys of the keyboard.

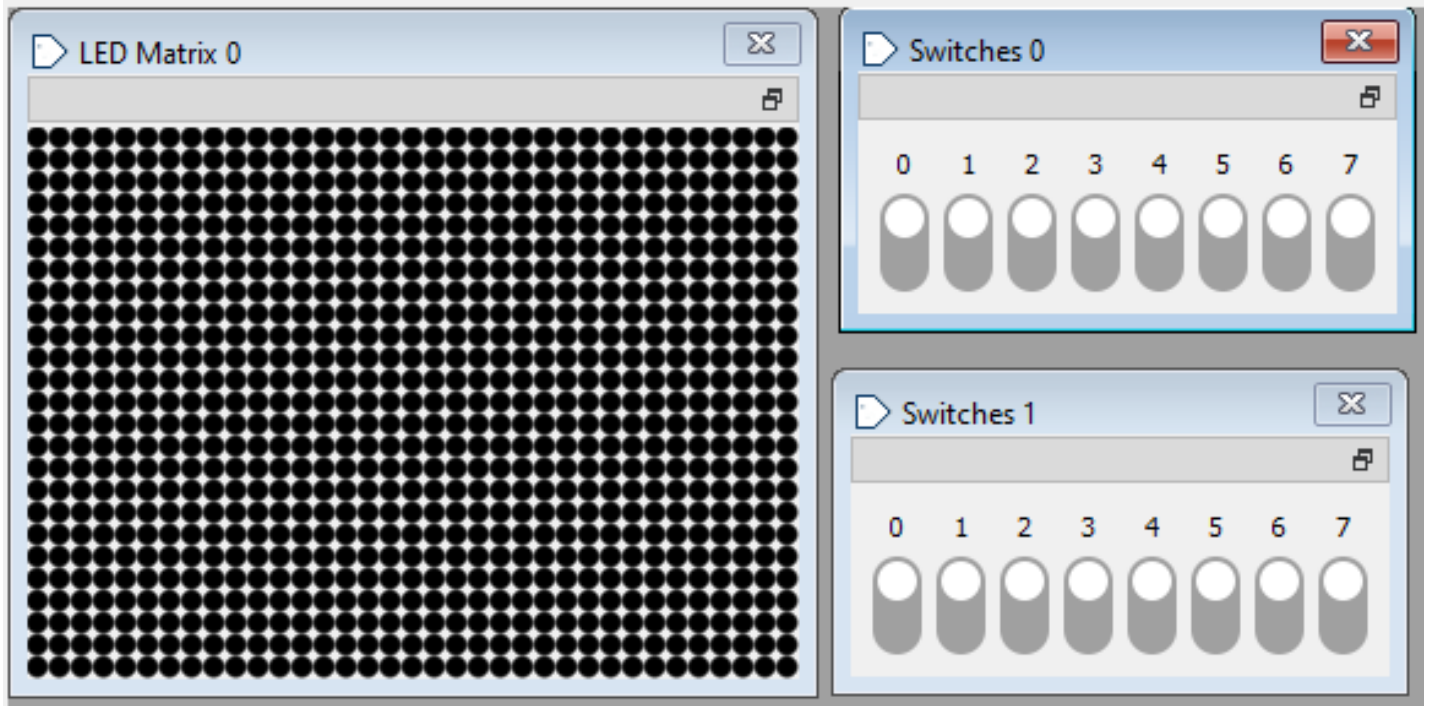
Register map

Name	Address	R/W?	Size
UP	0x0	R	1
DOWN	0x4	R	1
LEFT	0x8	R	1
RIGHT	0xc	R	1

Exports

```
#define D_PAD_0_BASE (0xf0000000)
#define D_PAD_0_SIZE (0x10)
#define D_PAD_0_UP_OFFSET (0x0)
#define D_PAD_0_UP (0xf0000000)
#define D_PAD_0_DOWN_OFFSET (0x4)
#define D_PAD_0_DOWN (0xf0000004)
#define D_PAD_0_LEFT_OFFSET (0x8)
#define D_PAD_0_LEFT (0xf0000008)
#define D_PAD_0_RIGHT_OFFSET (0xc)
#define D_PAD_0_RIGHT (0xf000000c)
```

E/S por consulta de estado
Simulador Ripes

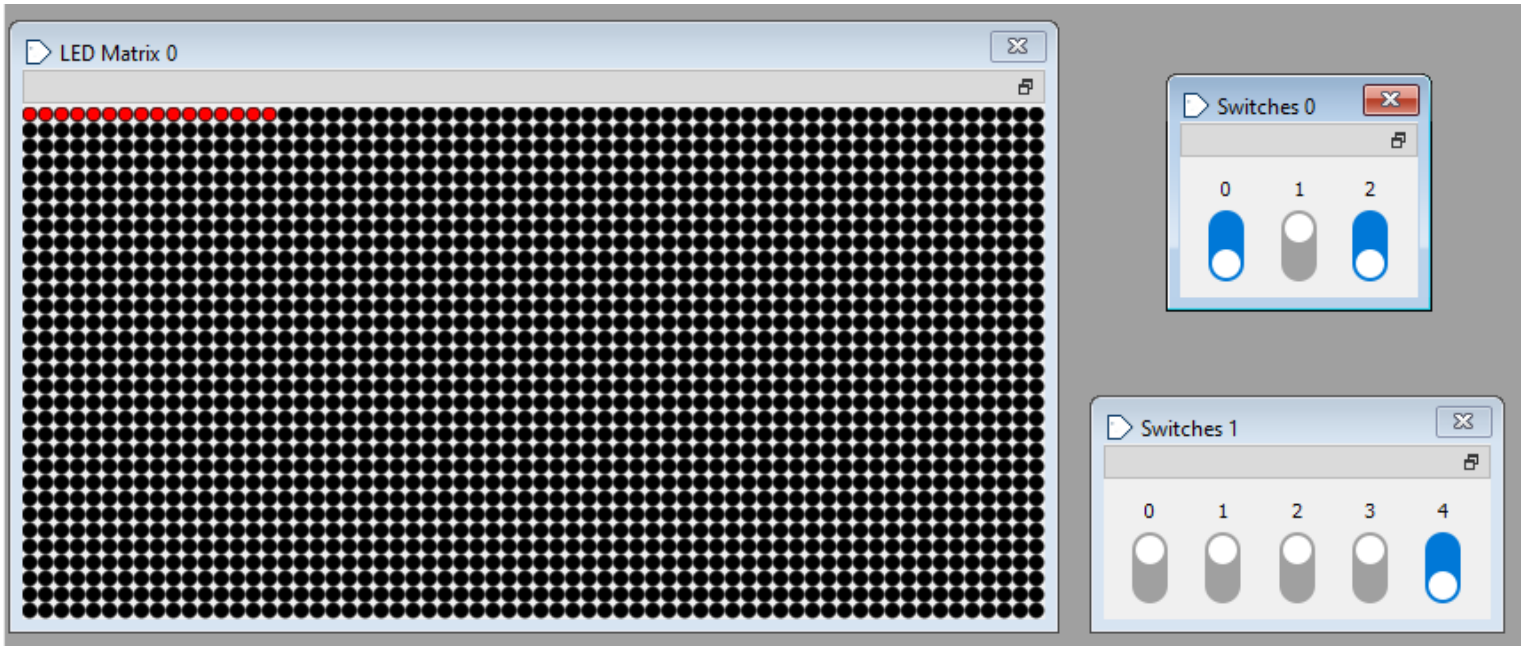


I/O exports

```
#ifndef RIPES_IO_HEADER
#define RIPES_IO_HEADER
// ****************************************************************************
// * LED_MATRIX_0
// ****************************************************************************
#define LED_MATRIX_0_BASE          (0xf0000000)
#define LED_MATRIX_0_SIZE          (0xdac)
#define LED_MATRIX_0_WIDTH         (0x23)
#define LED_MATRIX_0_HEIGHT        (0x19)


// ****************************************************************************
// * SWITCHES_1
// ****************************************************************************
#define SWITCHES_1_BASE            (0xf0000db0)
#define SWITCHES_1_SIZE            (0x4)
#define SWITCHES_1_N    (0x8)


// ****************************************************************************
// * SWITCHES_0
// ****************************************************************************
#define SWITCHES_0_BASE            (0xf0000dac)
#define SWITCHES_0_SIZE            (0x4)
#define SWITCHES_0_N    (0x8)


#endif // RIPES_IO_HEADER
```

E/S por consulta de estado
Ejemplo

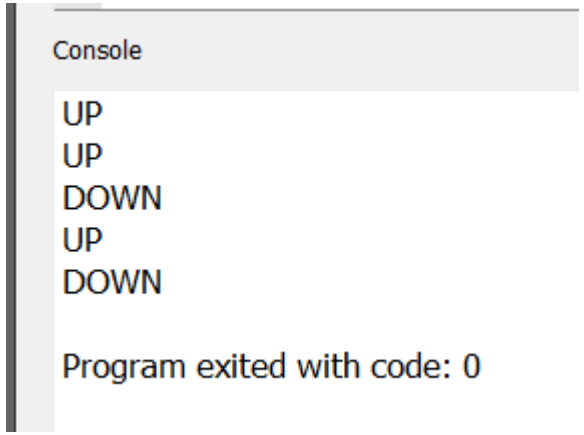
- Escribir un programa que
  - Espere a que el bit 2 del **switch 0** valga 1
  - Entonces imprima en pantalla tantos píxeles rojos como indique el **switch 1**




```
1     .text
2     li   a0,LED_MATRIX_0_BASE
3     li   s0,0xff0000    # rojo en RGB
4
5  sondeo:
6     lw   s1,SWITCHES_0_BASE
7     andi s1,s1,0x4
8     beqz s1,sondeo
9
10    lw   s2,SWITCHES_1_BASE
11 pinto:
12    beqz s2,fin
13    sw   s0,0(a0)
14    addi a0,a0,4
15    addi s2,s2,-1
16    j    pinto
17
18 fin:
19    li   a7,10
20    ecall
21
```

E/S por consulta de estado
Ejemplo

- Escribir un programa que
  - Mientras un conmutador esté desactivado
  - Compruebe e imprima en pantalla el estado del cursor

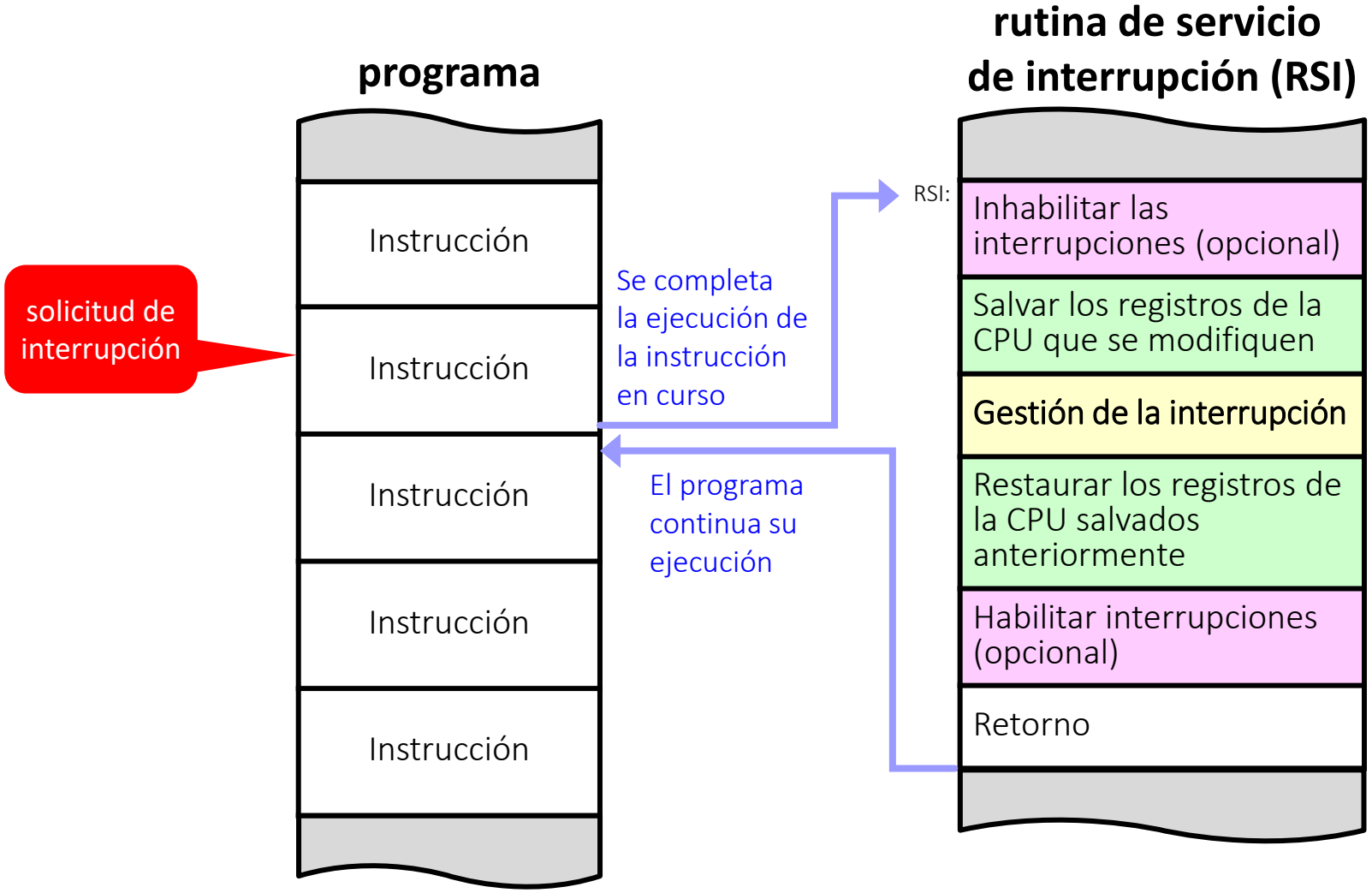



```
1  .data
2  txtUP:   .string "UP\n"
3  txtDOWN: .string "DOWN\n"
4
5  .text
6  .equ OSExit 10
7  .equ OSPrintString 4
8
9  repeat:
10     lw t0 D_PAD_0_UP
11     beqz t0 noUP
12     la a0 txtUP
13     li a7 OSPrintString
14     ecall
15
16 noUP:
17     lw t0 D_PAD_0_DOWN
18     beqz t0 noDOWN
19     la a0 txtDOWN
20     li a7 OSPrintString
21     ecall
22
23 noDOWN:
24     lw t0 SWITCHES_0_BASE
25     beqz t0 repeat
26
27 fin:
28     li a7 OSExit
29     ecall
30
```

E/S por interrupciones
Concepto

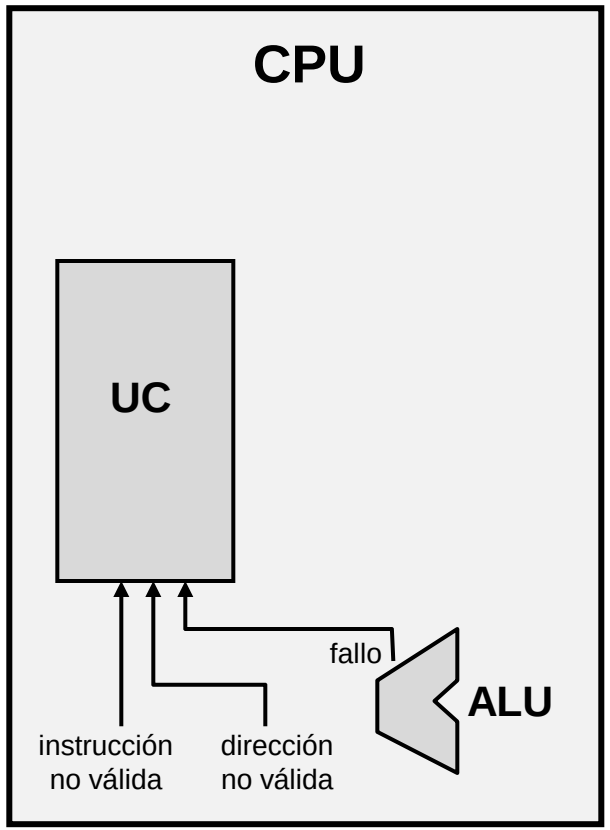
- Una interrupción es la suspensión temporal de la ejecución de un proceso
  - Para pasar a ejecutar una subrutina de servicio de interrupción
- Una vez finalizada dicha subrutina, se reanuda la ejecución del proceso en el punto en el que se quedó
  - El proceso no se percata de la interrupción

E/S por interrupciones
Concepto



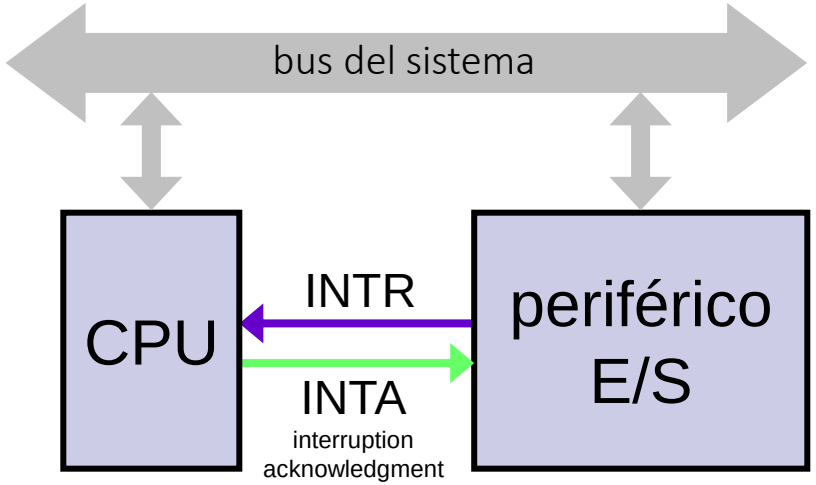
E/S por interrupciones
Interrupciones internas (software)

- Se generan a consecuencia de alguna circunstancia interna de la CPU
  - Código de operación no valido
  - Acceso a memoria no válido
  - Fallo en la operación ALU
    - Desbordamiento del resultado
    - División por cero
  - Interrupción de rastreo
    - En depuración de programas



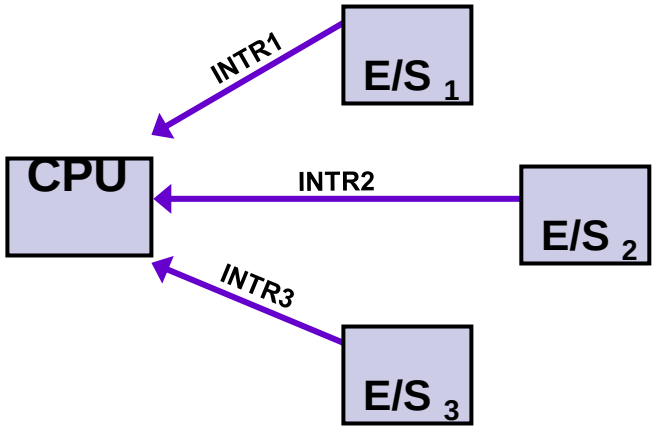
E/S por interrupciones
Interrupciones externas

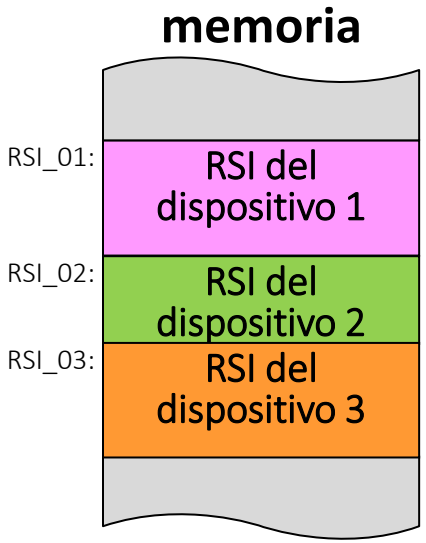
- Un periférico se encuentra conectado a una patilla de interrupciones del procesador
- La interrupción es asíncrona con respecto a la ejecución del programa
  - No está asociada a ninguna instrucción
  - El periférico lleva la iniciativa



E/S por interrupciones
Interrupciones externas múltiples

- Conexión individual (en estrella)
  - Consume muchas patillas del procesador
  - Identificación directa del dispositivo
    - Para ejecutar la RSI asociada
  - Gestión inmediata de las interrupciones

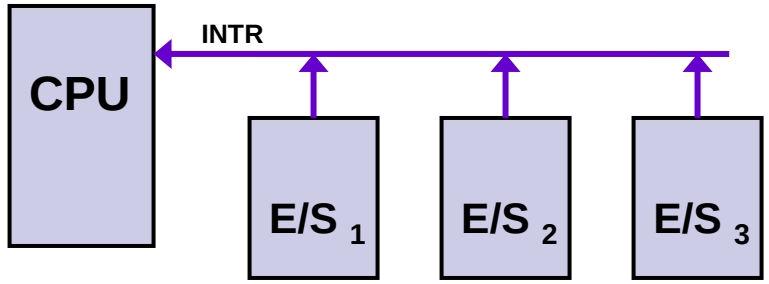


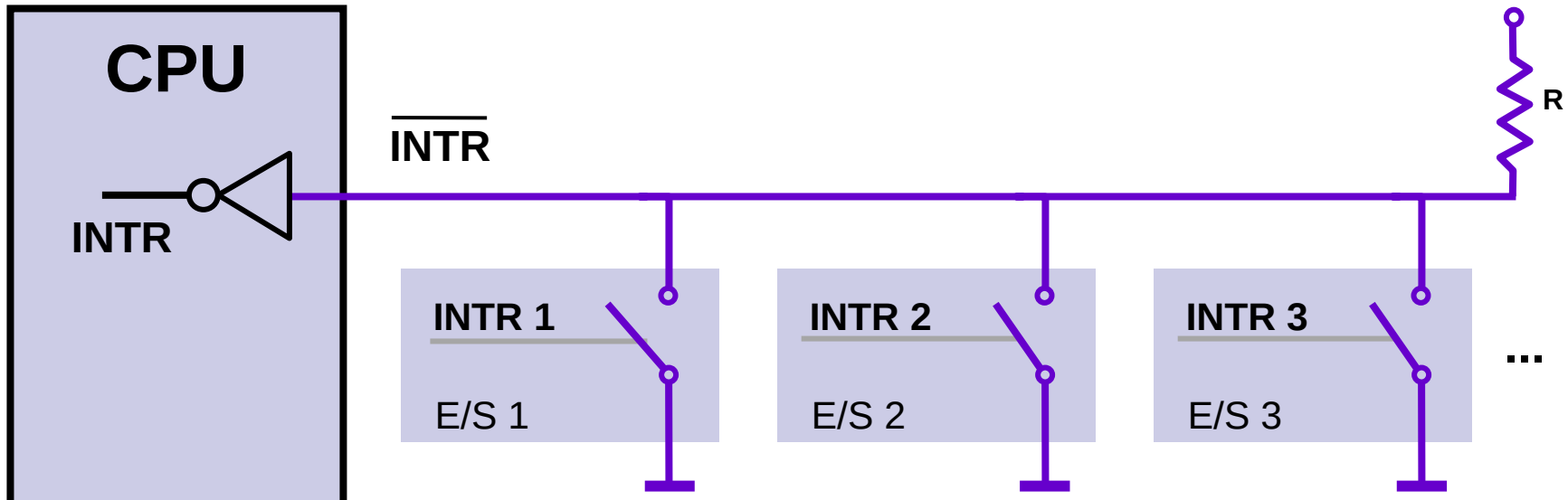


E/S por interrupciones

Interrupciones externas múltiples

- Conexión en cadena

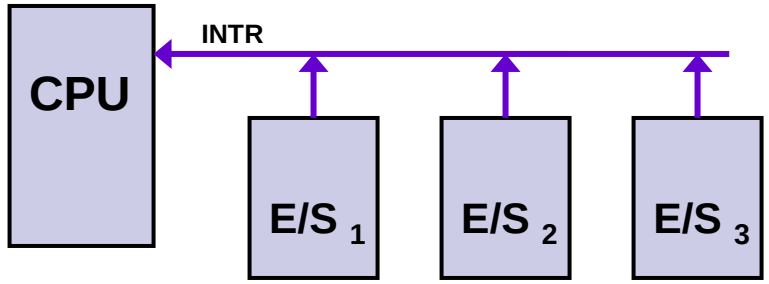




$$INTR = INTR_1 + INTR_2 + INTR_3 + ... + INTR_n$$

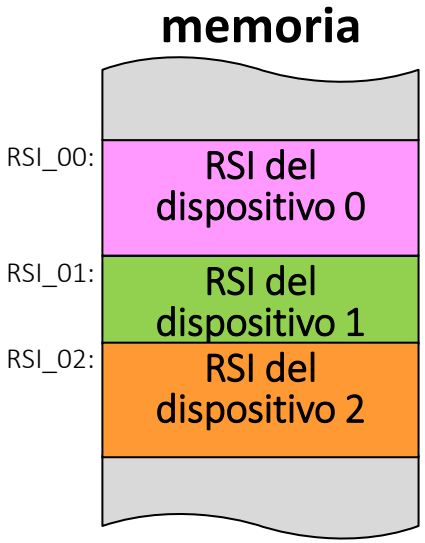
E/S por interrupciones
Interrupciones externas múltiples

- Conexión en cadena



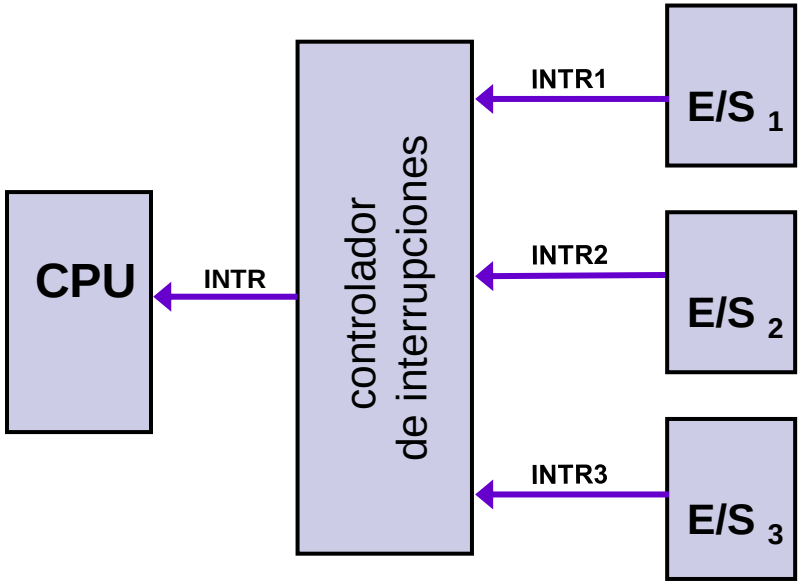
  - Consume pocas patillas del procesador

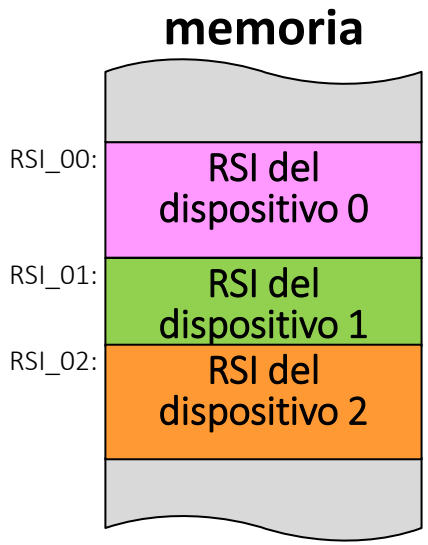
  - Mecanismo adicional para identificar al dispositivo
    - Y ejecutar la RSI asociada
  - Gestión lenta de las interrupciones



E/S por interrupciones
Interrupciones externas múltiples

- Conexión mediante controlador de E/S
  - Consume pocas patillas del procesador
  - Hardware adicional
  - Mecanismo adicional para identificar al dispositivo
    - Y ejecutar la RSI asociada
  - Gestión rápida de las interrupciones

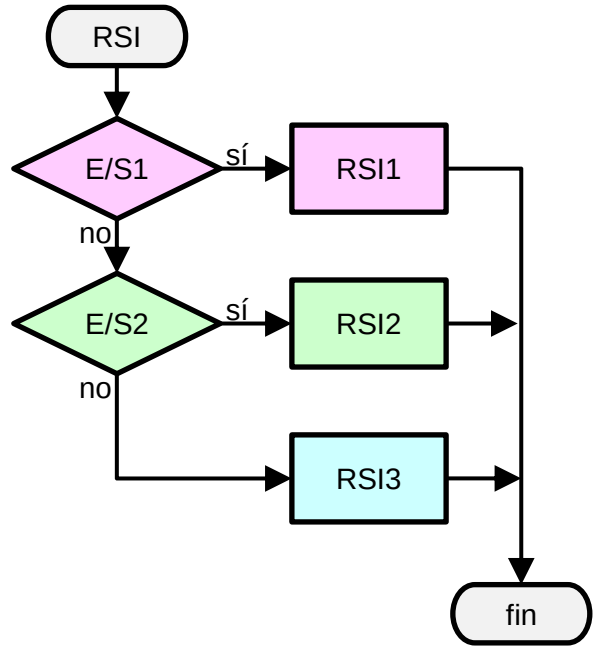


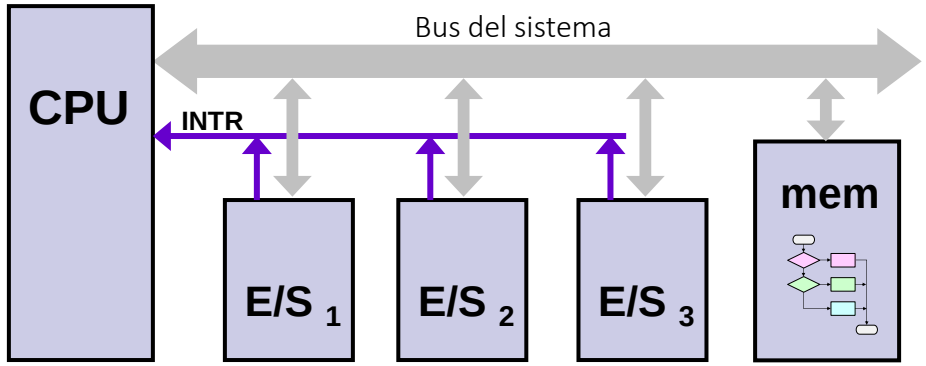


Identificación del dispositivo

Interrupciones prefijadas

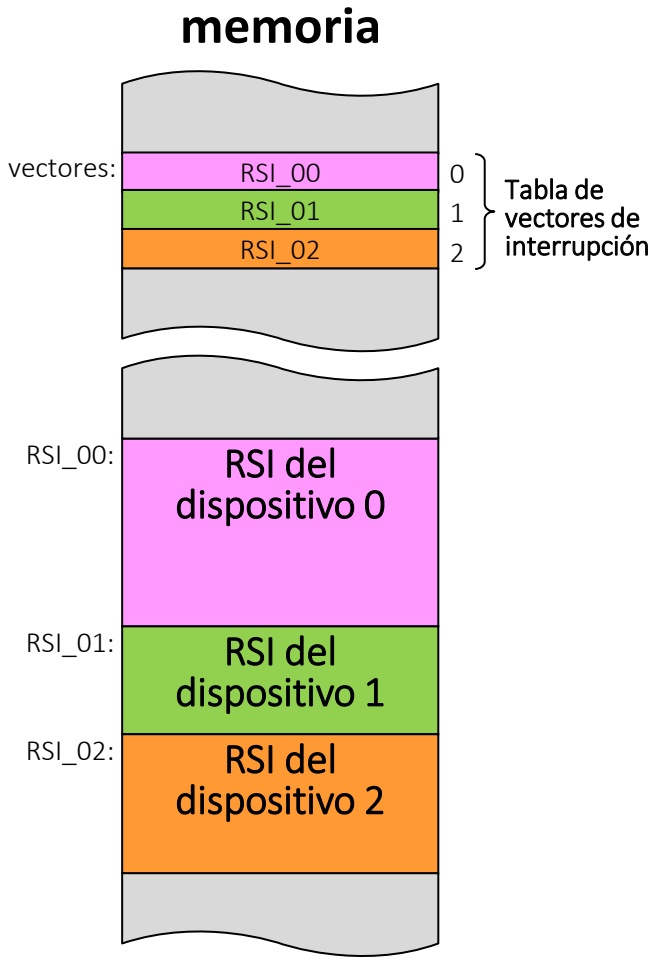
- Todas las interrupciones están centralizadas en una única RSI
- que sondea los dispositivos, y determina cuál solicitó la interrupción
  - También denominadas interrupciones por escrutinio, sondeo o polling
- Lento
  - Implementación software



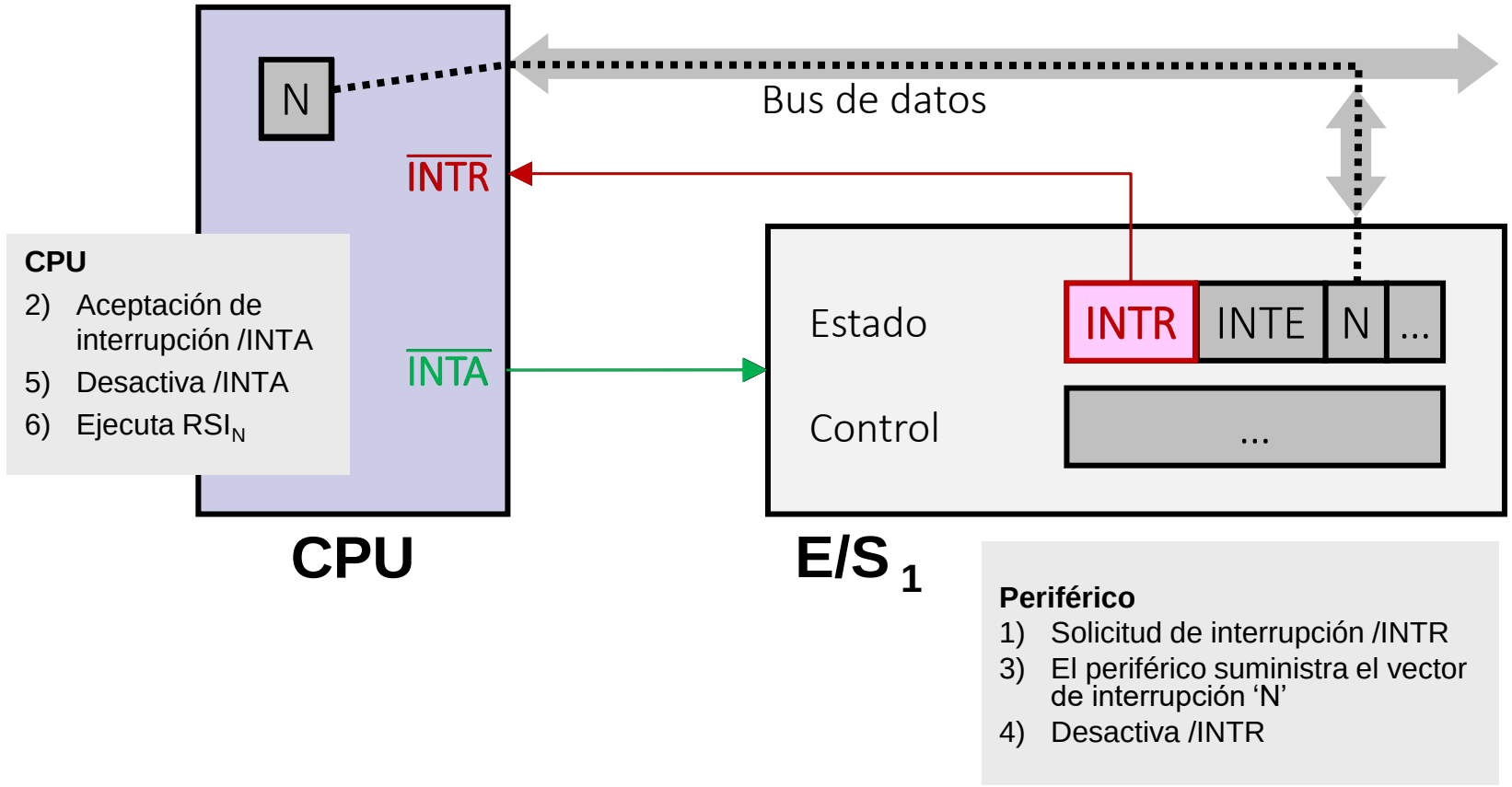


Identificación del dispositivo
Interrupciones vectorizadas

- El dispositivo proporciona
  - La dirección de la RSI
  - Parte de la dirección de la RSI
  - Un índice de una tabla de "vectores de interrupción"
- Rápido

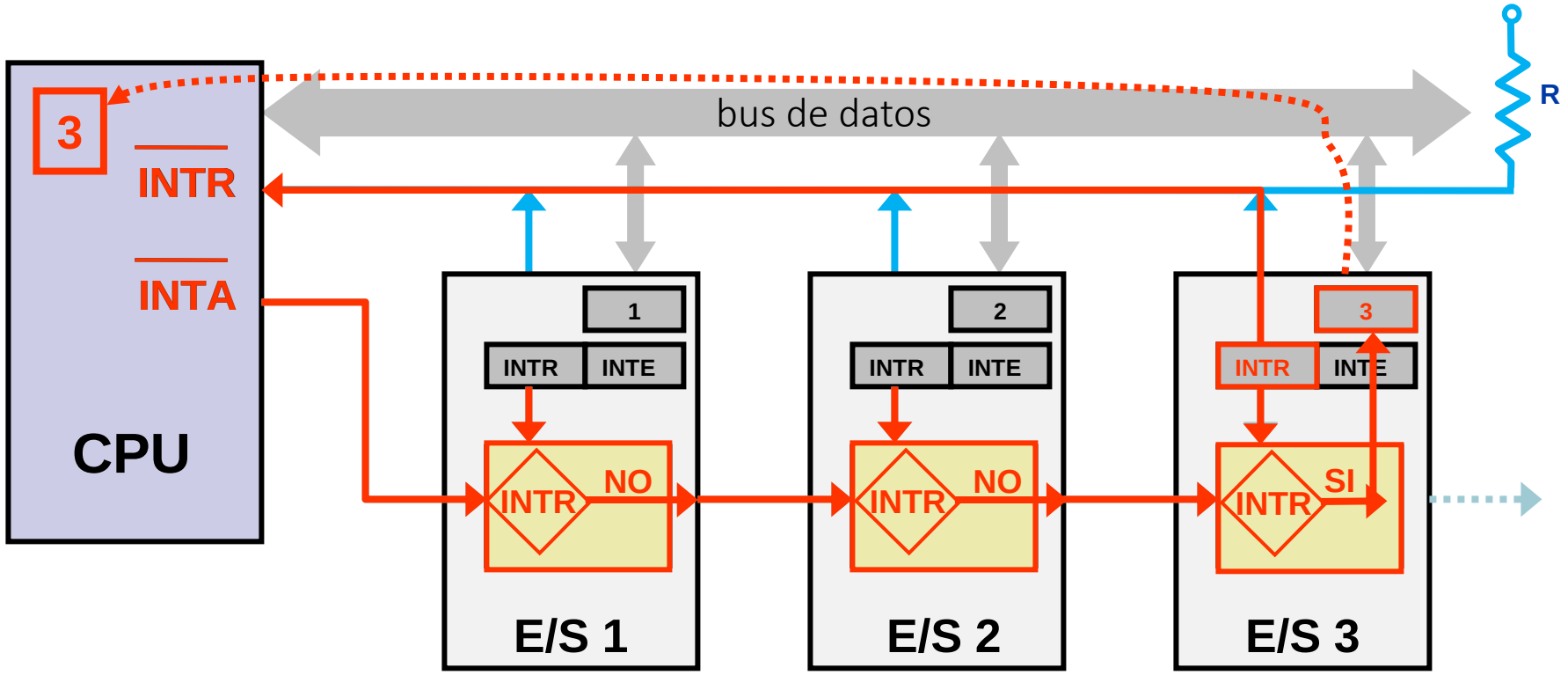


Identificación del dispositivo
Interrupciones vectorizadas



Identificación del dispositivo
Interrupciones vectorizadas

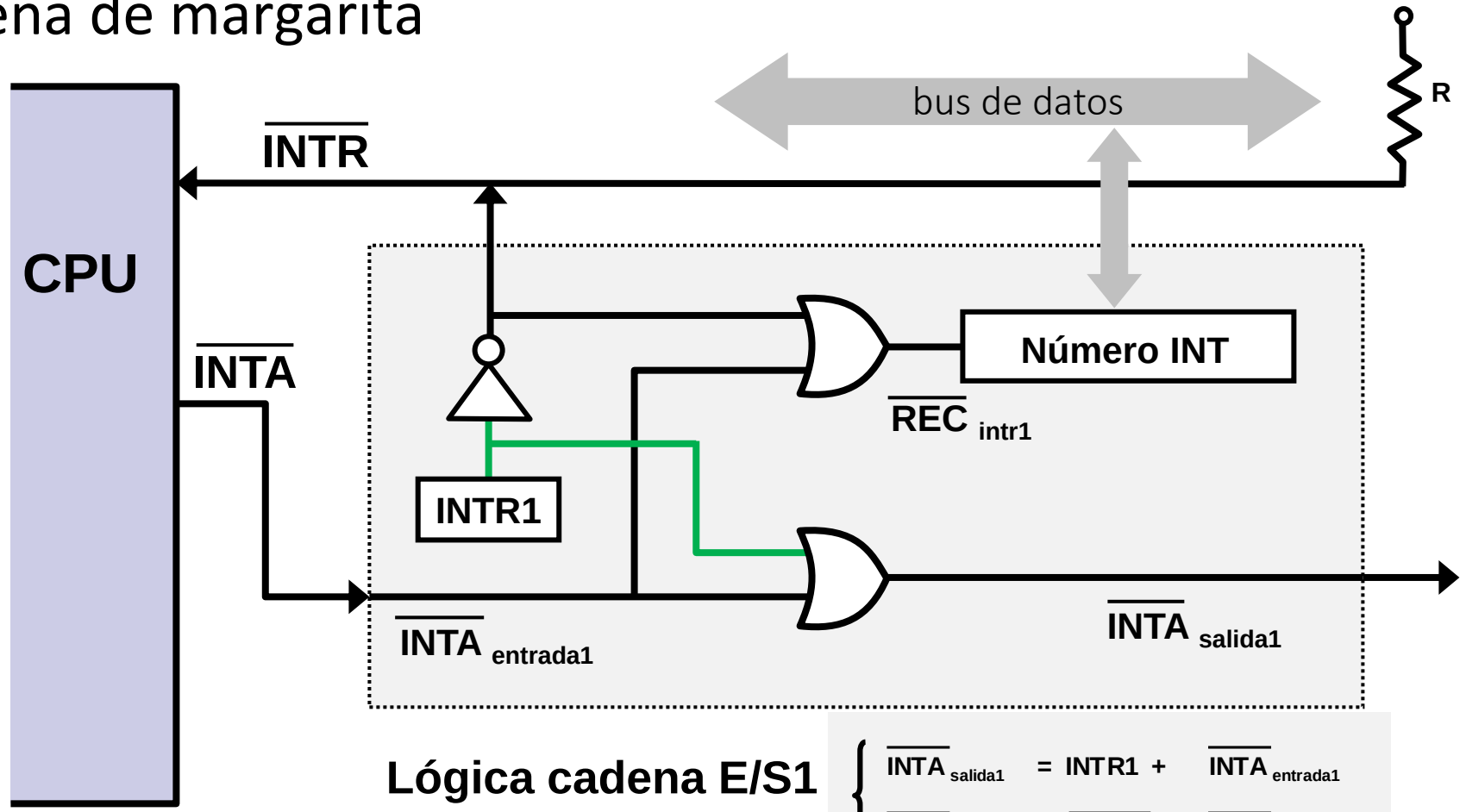
- Conexión en cadena de margarita (daisy chain)



Identificación del dispositivo

Interrupciones vectorizadas

- Conexión en cadena de margarita (daisy chain)

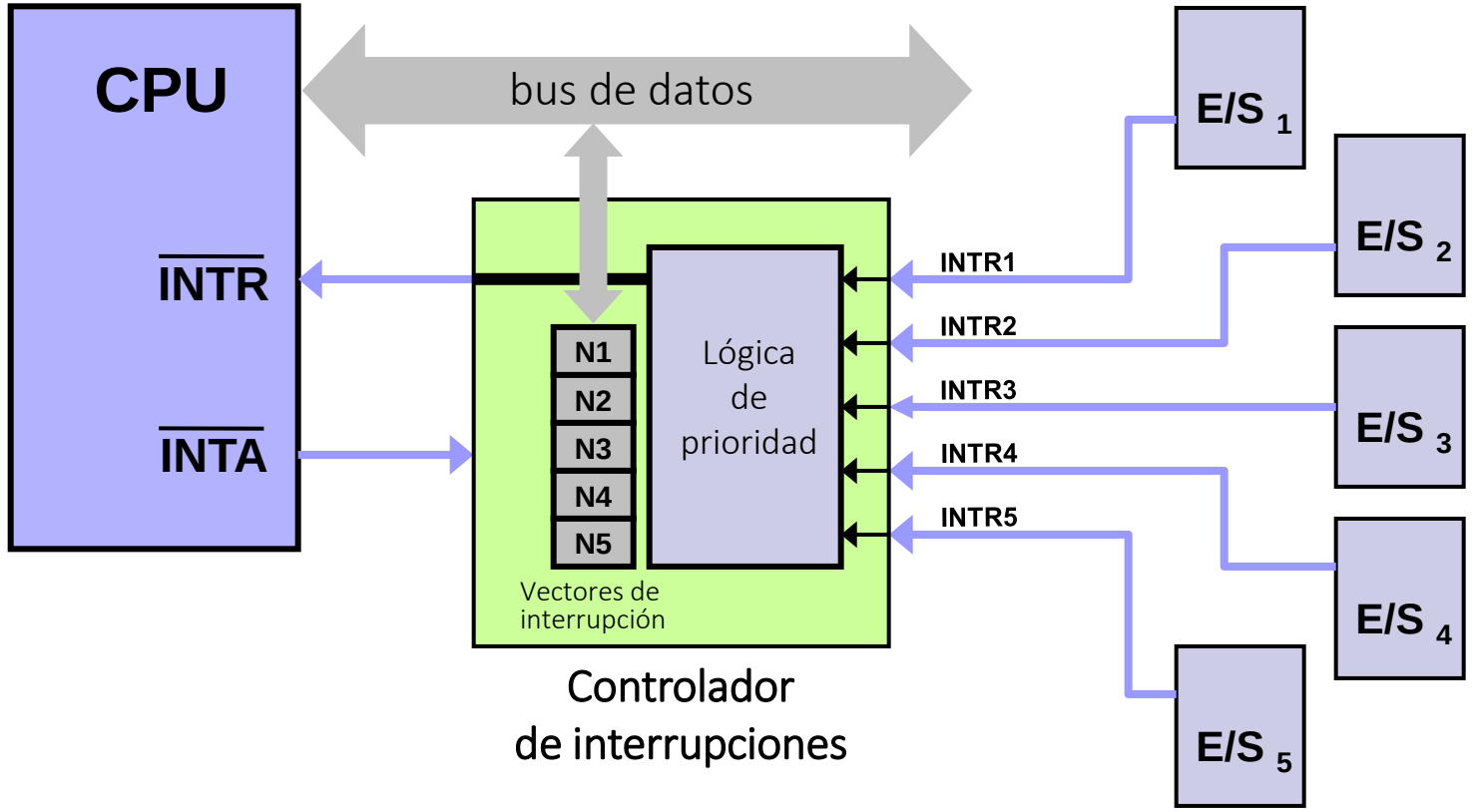


Lógica cadena E/S1

$$\begin{cases} \overline{INTA}_{salida1} = INTR1 + \overline{INTA}_{entrada1} \\ \overline{REC}_{intr1} = \overline{INTR1} + \overline{INTA}_{entrada1} \end{cases}$$

Identificación del dispositivo
Interrupciones vectorizadas

- Conexión con controlador de interrupciones



Interrupciones simultáneas

Interrupciones simultáneas

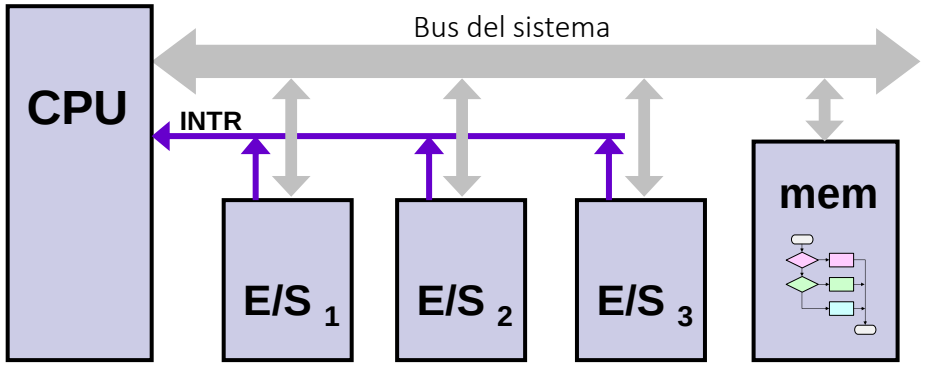
- ¿Qué ocurre cuando varios dispositivos solicitan simultáneamente una interrupción?

- Sólo es posible atender a un periférico cada vez

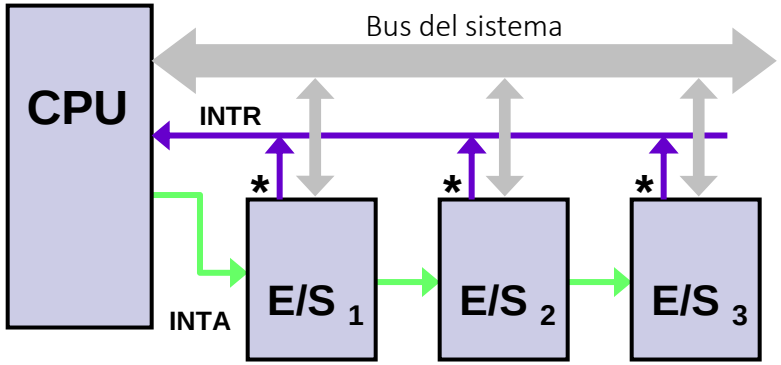
- Habrá que atenderlos en función de un esquema de prioridades

Interrupciones simultáneas
Resolución de prioridades

- Interrupciones prefijadas
  - El orden en la secuencia de escrutinio determina la prioridad



- Interrupciones vectorizadas
  - La posición en la cadena determina la prioridad



Interrupciones simultáneas
Anidamiento de interrupciones

- La CPU debe evitar el anidamiento descontrolado de interrupciones

- Para ello puede deshabilitarlas
  - La RSI comienza con la deshabilitación automática de interrupciones
  - Y termina con su habilitación (instrucción tipo IRET)
- La deshabilitación puede ser jerarquizada
  - Sólo se desactivan interrupciones de prioridad igual o menor a la atendida actualmente

Interrupciones simultáneas
Inhibición de interrupciones

- Gestión global mediante programa
  - Instrucciones tipo EI, DI
- Deshabilitación selectiva
  - Mediante registros especiales (máscaras) en los que cada bit identifica un nivel de interrupción
- Interrupciones no enmascarables