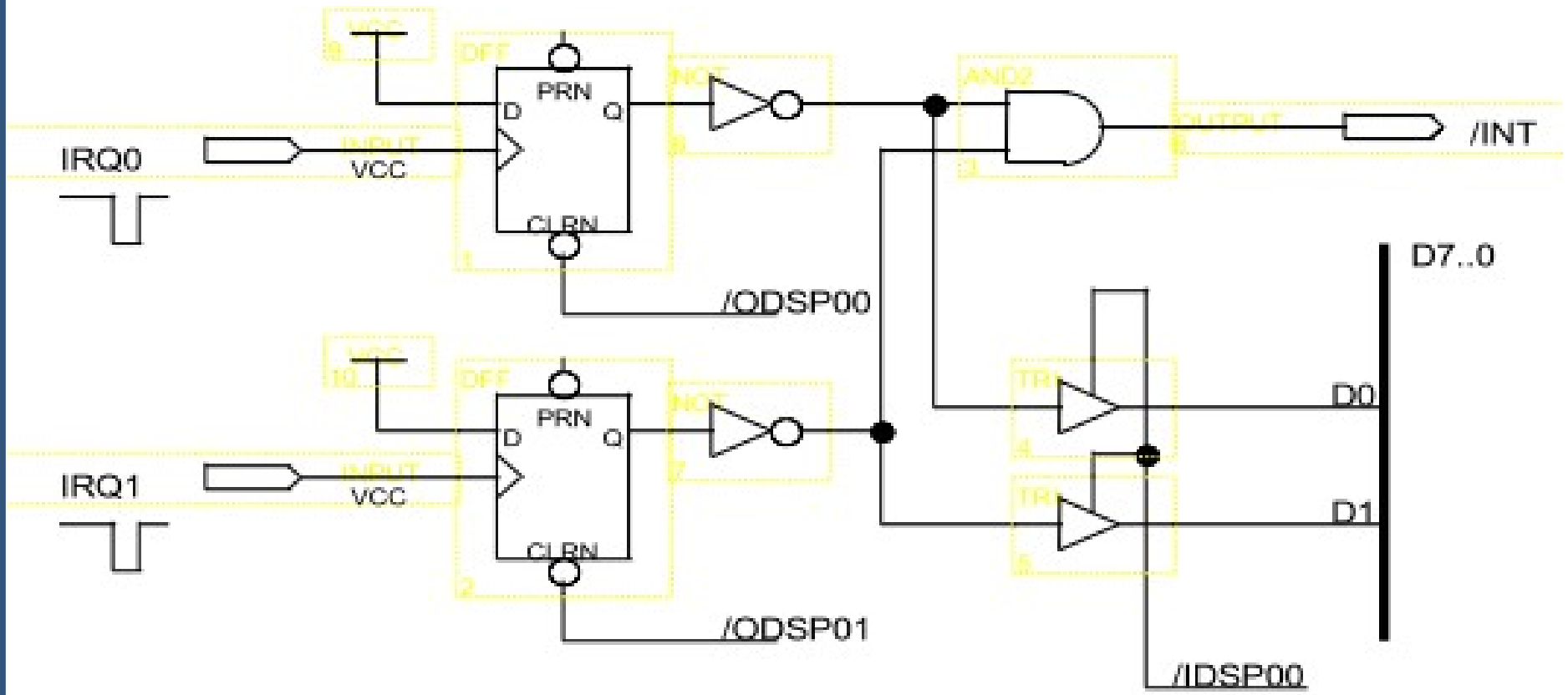
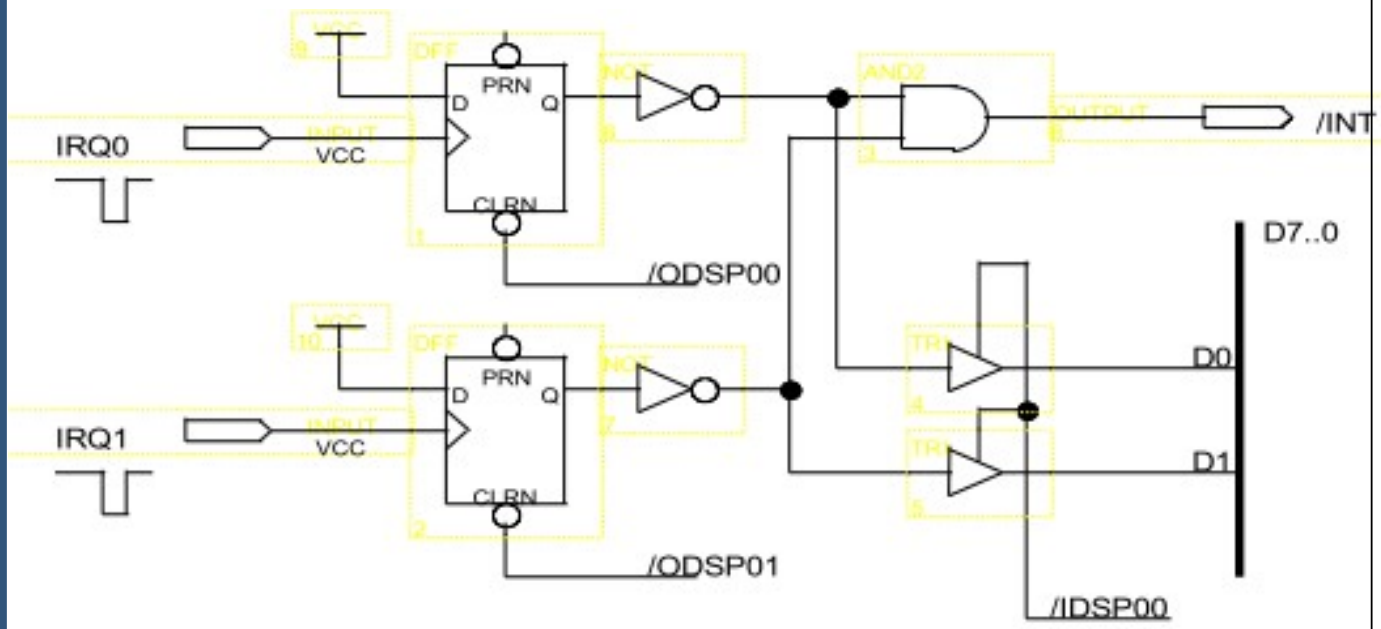
Interrupciones
Manejo de Prioridades

- Modo 2 rudimentario
- Manejo de Prioridades en interrupciones
  - Más de una petición pendiente, ¿a quién atiendo primero?
  - ¿Quién puede interrumpir a quién?
- Controlador de interrupciones
- Políticas de prioridad
  - Fully Nested
  - Prioridad rotativa
- Daisy chain

Ejemplo: 2 dispositivos que interrumpen en MODO 1



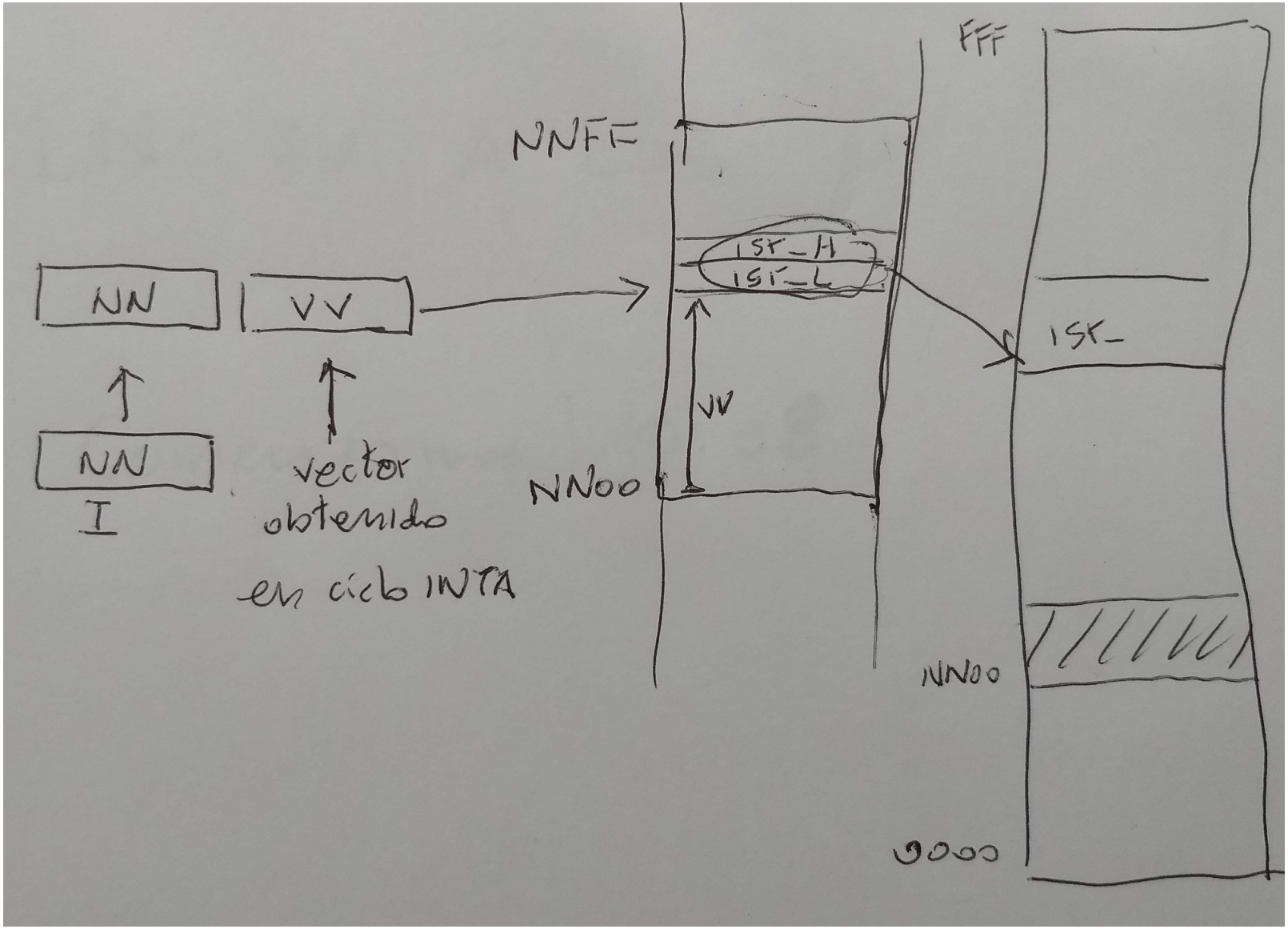
Ejemplo: 2 dispositivos que interrumpen en MODO 1



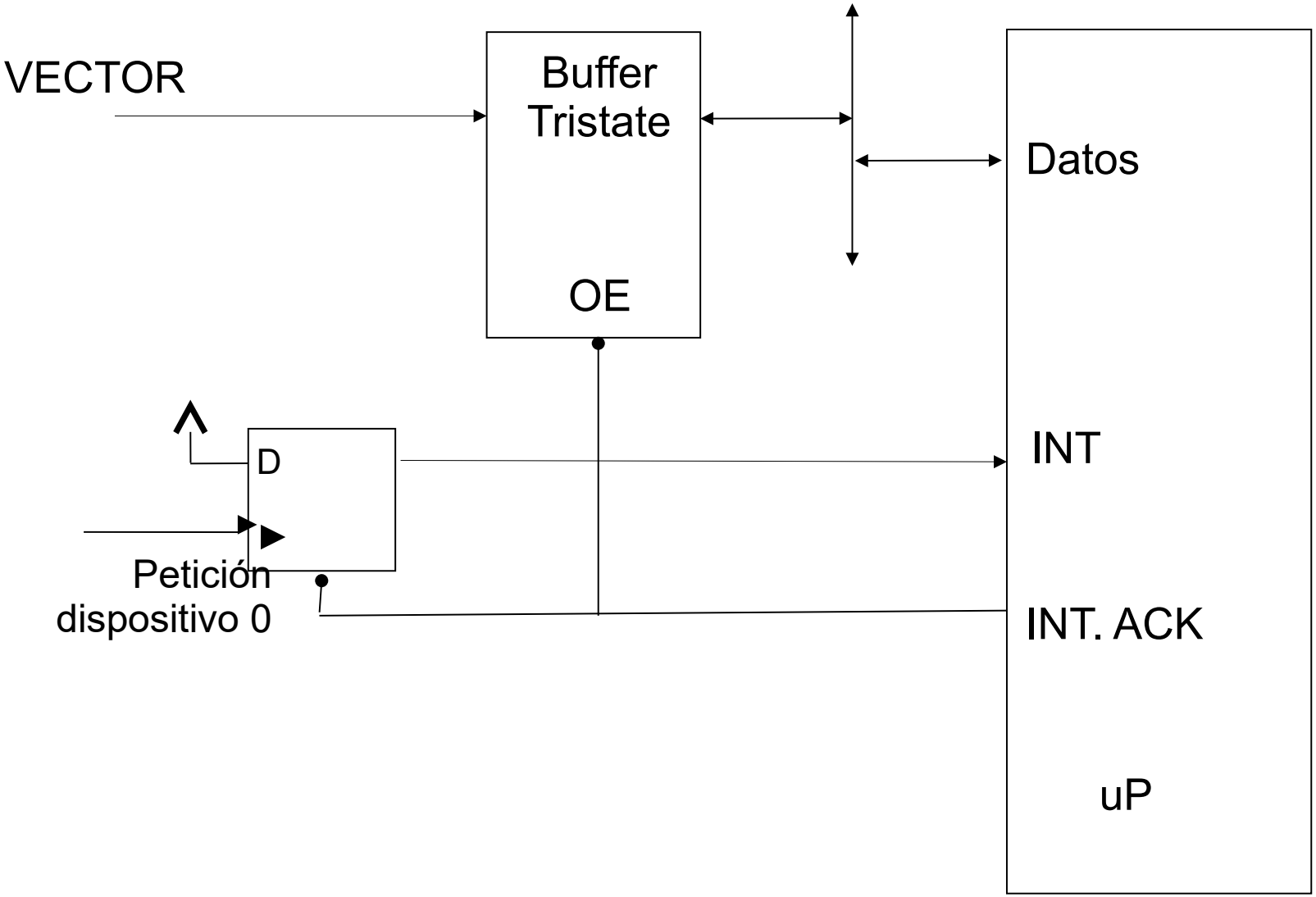

```
pendientes EQU 0
borro0     EQU 0
borro1     EQU 1

           org 38H
rutint:    push af
           in a, (pendientes)
           bit 0, a
           jr z, atiendo0
atiendo1:
           ei
           out (borro1), a
           call isr1
           pop af
           ret
atiendo0:
           out (borro0), a
           call isr0
           pop af
           ei
           ret
```

- Los FF de petición se borran por SW.
- IRQ0 puede interrumpir a IRQ1 pero no viceversa (ei)

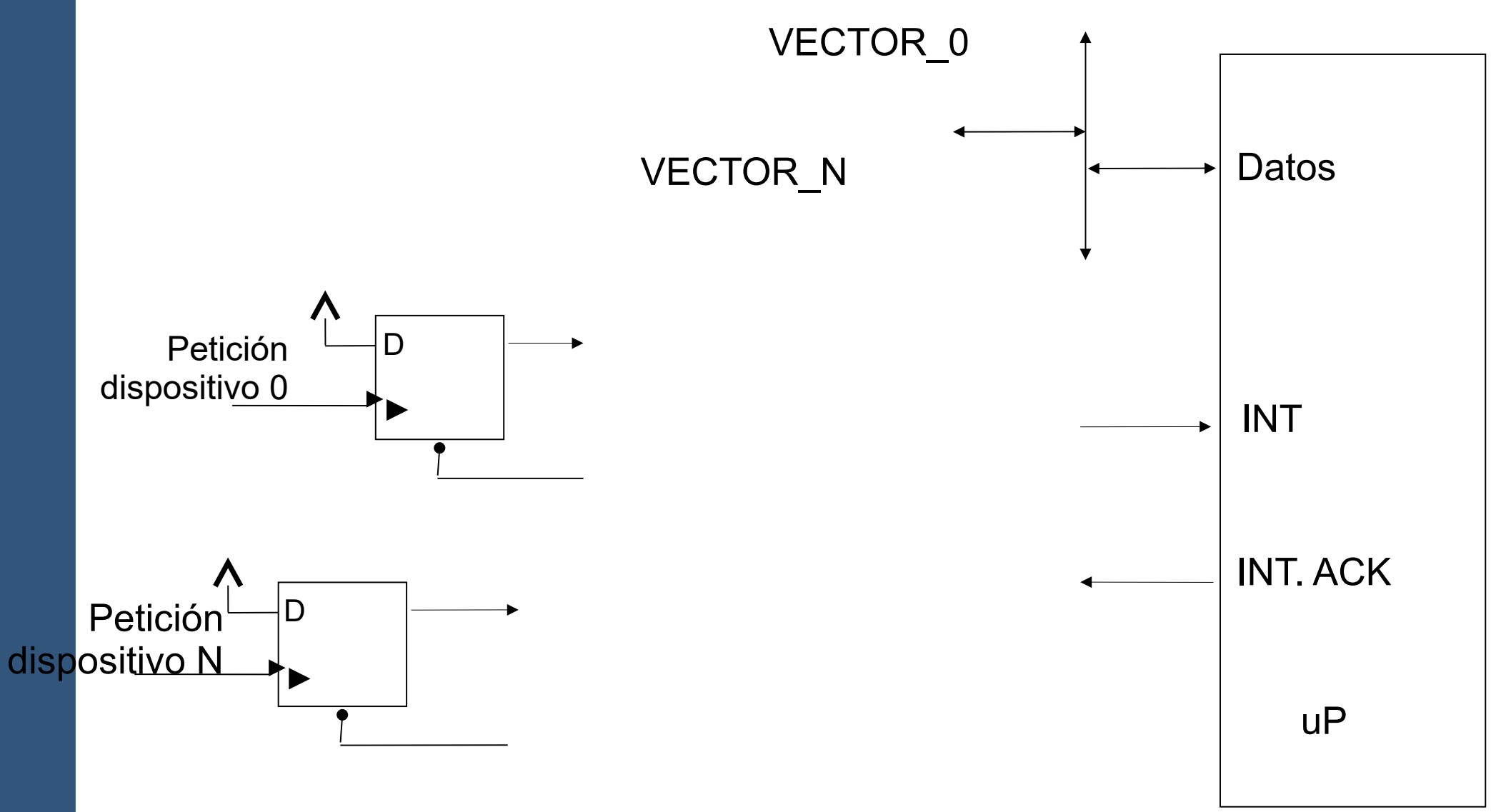
Modo 2
Primer intento



Modo 2
Primer intento



Interrupciones
Prioridades

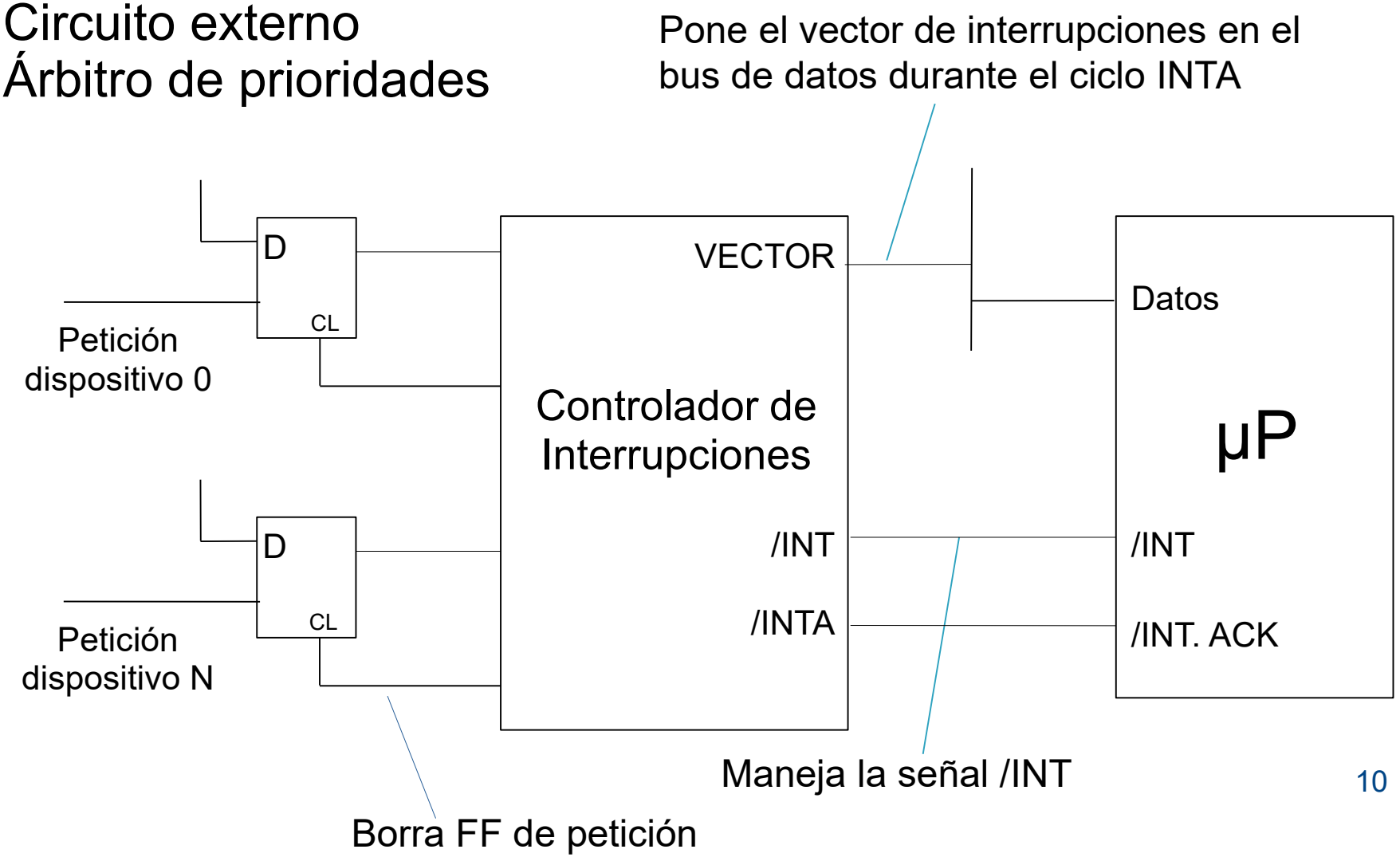


Interrupciones Prioridades

- Orden de prioridad para:
  - Más de una petición pendiente en ciclo INTA
    - ¿Cuál debe ser atendida?
    - En ejemplo modo 1: solución software
      - ¿Cuál flag miro primero?
    - En modo 2: solución Hardware
      - ¿Quién pone su vector sobre el bus?
  - Periférico importante interrumpe a otro menos importante pero no a la inversa.
    - En ejemplo modo 1: jugando con ubicación de EI.
      - Imposible con más de dos periféricos.
    - En modo 2. Solución Hardware
      - Dejar pasar o no la solicitud del periférico hasta entrada /INT

Controlador interrupciones

- Solución por hardware:
  - Circuito externo
  - Árbitro de prioridades



Controlador Interrupciones

- Cuando llega ciclo INTA
  - Pone en el bus el vector del dispositivo de más prioridad de los pendientes de ser atendidos.
- Cuando llega una solicitud
  - La pasa al procesador (activando INT) sólo si el dispositivo es de más prioridad que el dispositivo que está siendo atendido.
  - Caso contrario espera a que terminen todos los de mayor prioridad
- Chip 8259 de Intel
  - Utilizado en computadores personales

Interrupciones *"Fully Nested"*

- Orden fijo de prioridades
- *Politica*
  - *Siempre se está atendiendo al de mayor prioridad*
- Asignación prioridades a los dispositivos
  - Problema complejo
  - Varias características
    - Requerimientos en tiempo respuesta.
    - Qué tan catastrófico es no atenderlo a tiempo.
    - Periodicidad
    - Duración de rutina de atención a interrupción
  - Mapeadas a único orden de prioridades.
- Objetivo:
  - Garantizar que se respetan todos los *deadlines*

Interrupciones
Jerarquía fija de prioridades

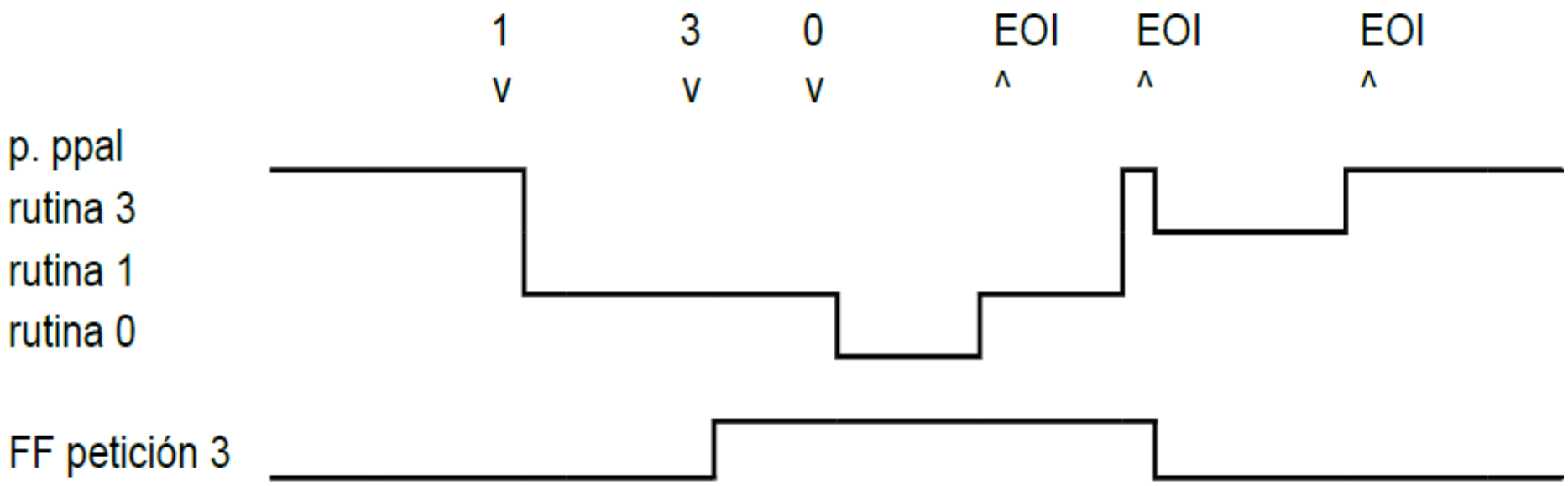
- Periféricos p0, p1, p2, p3
  - Prioridades fijas (p0 mayor prioridad)
  - Inicialmente ejecutando programa principal
  - Interrumpe p1, luego p3 y finalmente p0
  - ¿Qué sucede?

		1				3		0												
		v				v		v												
p.ppal.	x	x																		
rutina3																				
rutina2																				
rutina1			x	x																
rutina0																				

Interrupciones
Jerarquía fija de prioridades

- Periféricos p0, p1, p2, p3
  - Prioridades fijas (p0 mayor prioridad)
  - Inicialmente ejecutando programa principal
  - Interrumpe p1 → pasa a ejecutar rutina 1
  - Pide interrupción p3
    - Menor prioridad que p1, debe esperar
    - ¿Cómo?
      - Controlador de interrupciones debe postergar solicitud
  - Pide atención p0
    - Mayor prioridad → pasa a ejecutar rutina 0
  - p3 recién es atendido después de terminar p0 y p1

Interrupciones
Jerarquía fija de prioridades



Interrupciones
Jerarquía fija de prioridades

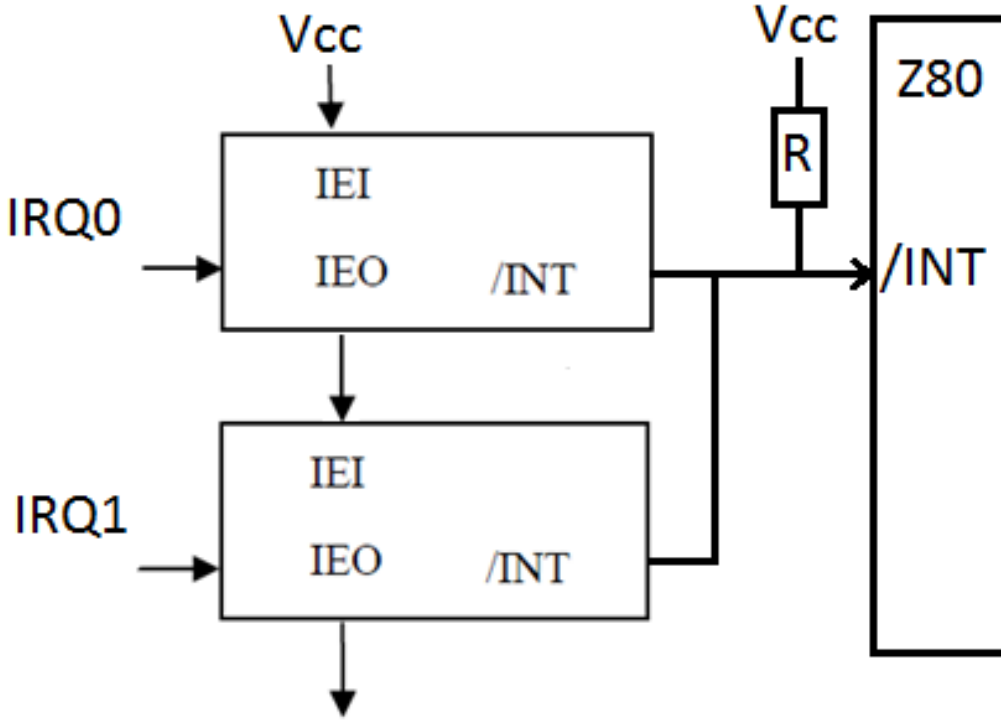
- Controlador necesita saber quién está ejecutando en cada momento.
- Sabe cuándo comienza cada rutina
- Necesita mecanismo para saber cuándo termina.
- Solución 8259:
  - Comando “Fin de Interrupción” (EOI) dado por el procesador escribiendo en un puerto.
- Solución Zilog
  - En modo 2 se retorna con instrucción RETI
    - Ídem RET
    - Opcode diferente (dos bytes ED 4D)
    - Controlador observa bus hasta ver pasar opcode de RETI

Interrupciones
Prioridad rotativa

- Más “justo” con dispositivos de similar importancia
- Cada vez que se atiende a un dispositivo, éste pasa al último lugar en el orden de prioridad.
- Permite poner cota superior a tiempo de respuesta.
- Esto último no es posible en esquema *“Fully Nested”*

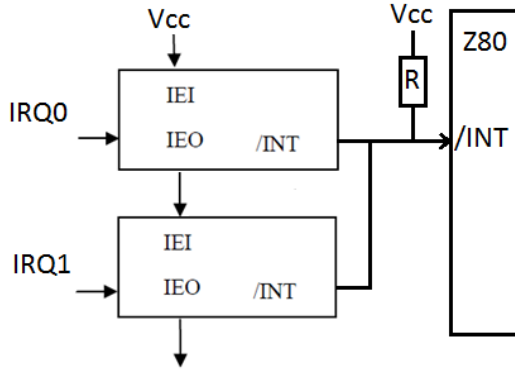
Interrupciones
"Daisy Chain"

- Mecanismo distribuido de arbitración de prioridades.
- Utilizado por Zilog.
- Esquema prioridad fija (*Fully nested*).
- Bloque para cada dispositivo
  - Entrada IEI informa si no se completó aún atención de alguien más arriba en la cadena.
  - Salida IEO conectada a entrada IEI del siguiente bloque.



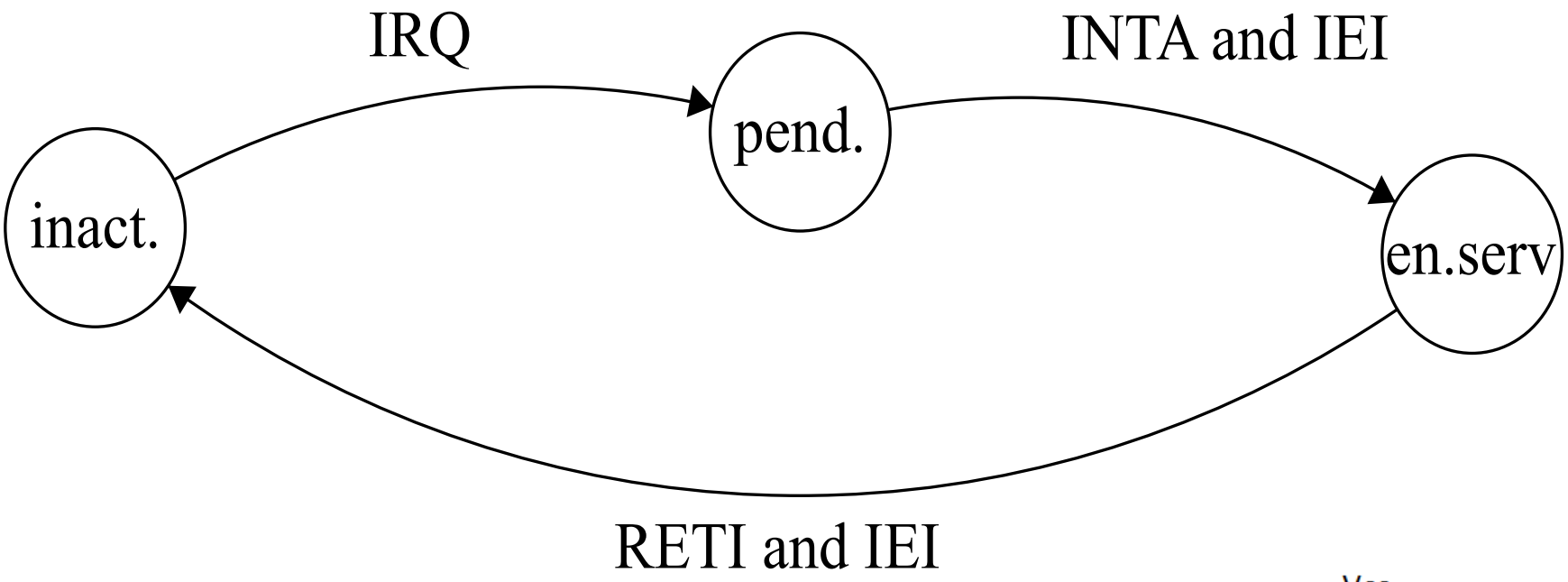
Interrupciones
"Daisy Chain"

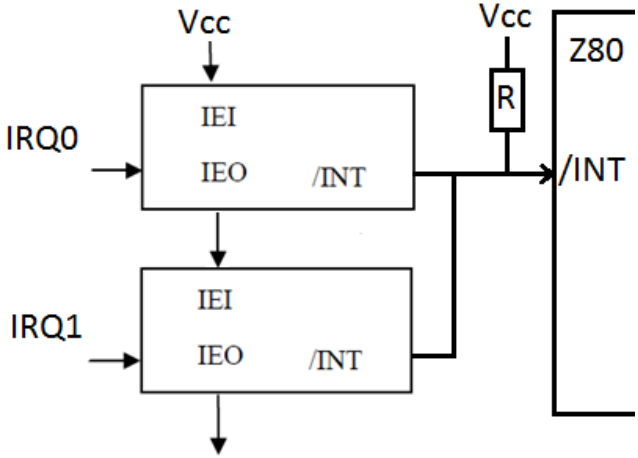
- IEI (Interrupt Enable Input):
  - inactiva (0) indica periférico de mayor prioridad no terminado de atender.
- IEO (Interrupt Enable Output):
  - conectada a IEI de siguiente periférico. Debe desactivarse cuando:
    - Este dispositivo tiene solicitud no completada, o
    - IEI inactiva.
- INT:
  - Entrada INT del microprocesador.
  - AND de las salidas INT de todos los bloques.
  - En general salida en *"colector abierto"*, unidas en conexión *"and cableado"*.



- El bloque debe detectar:
  - Ciclo INTA para poner, cuando le corresponda, su vector sobre el bus.
  - End of Interrupt. Observando buses y detectando ejecución de RETI.

Interrupciones
"Daisy Chain"



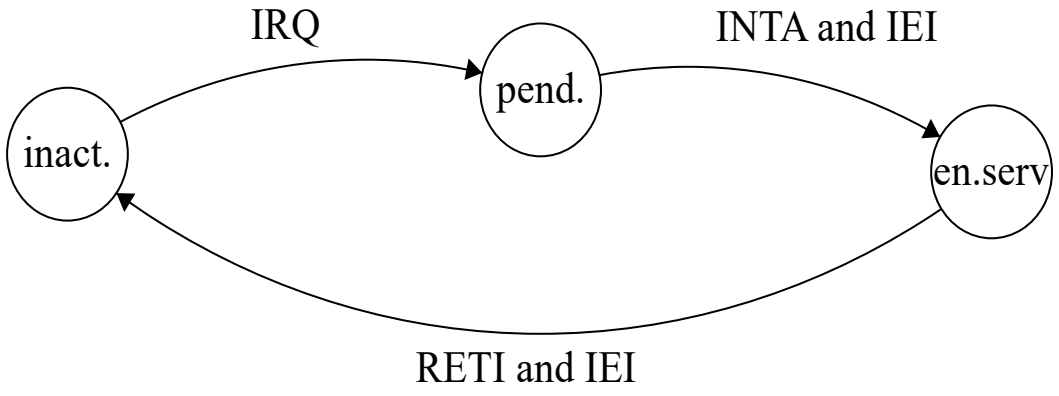


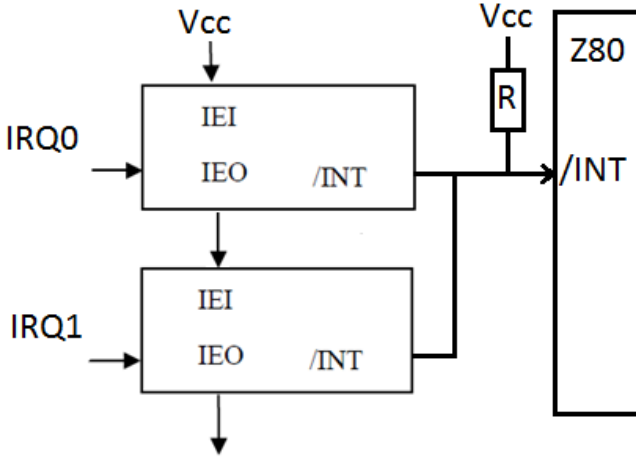
Interrupciones
"Daisy Chain"

- ¿Cómo detecta un dispositivo un ciclo INTA?
  - R: Tiene como entradas /M1 e /IORQ del Z80 (/M1 + /IORQ = /INTA)
- ¿Cómo sabe un dispositivo que debe poner SU vector de interr. en el bus de datos?
  - R: Detecta un ciclo INTA, tiene una petición pendiente e IEI = 1.
- ¿Como sabe un dispositivo cuando finaliza su ISR (Interrupt Service Rutine)?
  - R: Tiene una solicitud en servicio, observa el OPCODE de la instrucción RETI e IEI = 1, entonces ese RETI es de su ISR y lo reconoce como un EOI (End Of Interrupt).

Interrupciones
"Daisy Chain"

- IEO (Interrupt Enable Output):
  - IEO = IEI and (estado Inactivo).
- INT:
  - INT activa = (estado Pendiente) and IEI
- Vector
  - se pone sobre el bus si:
  - (ciclo INTA) and (estado Pendiente) and IEI





Interrupciones
Manejo de Prioridades

- Modo 2 rudimentario
- Manejo de Prioridades en interrupciones
  - Más de una petición pendiente, ¿a quién atiendo primero?
  - ¿Quién puede interrumpir a quién?
- Controlador de interrupciones
- Políticas de prioridad
  - Fully Nested
  - Prioridad rotativa
- Daisy chain