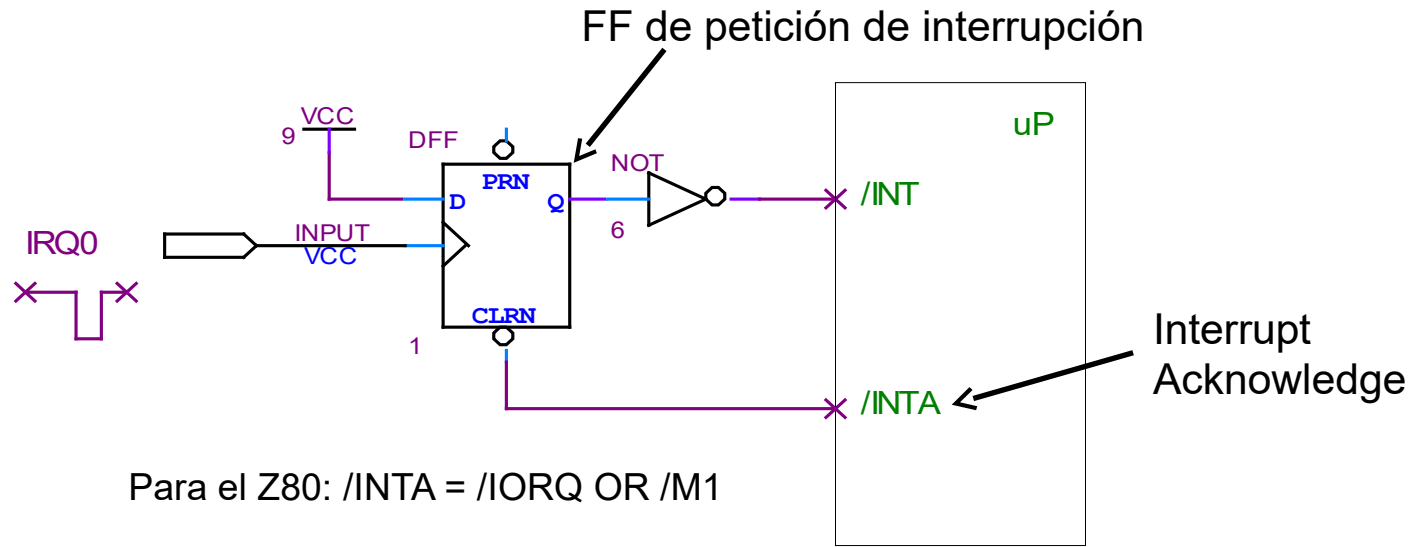


# Interrupciones Z80

Introducción a los microprocesadores  
2016

# Interrupciones

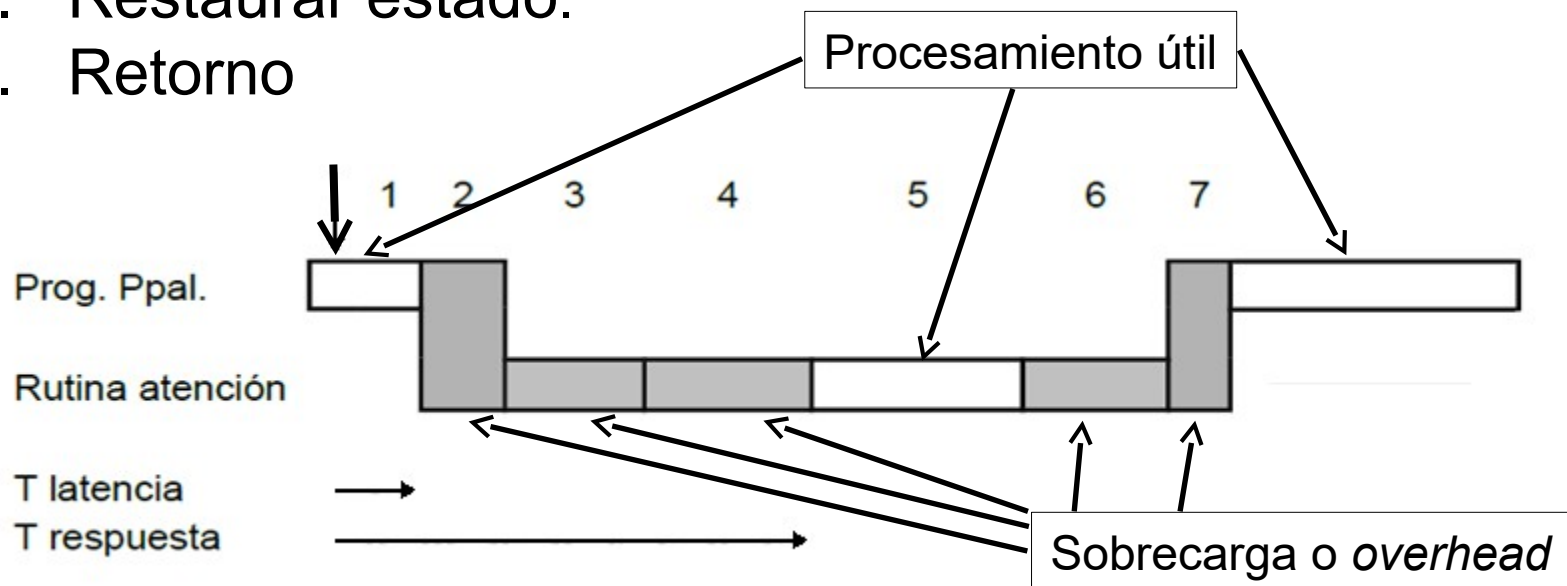
- Bandera de estado se convierte en FF de petición.



- El  $\mu P$  consulta la entrada  $/INT$  al final de cada instrucción.
  - Si esa entrada está activa, comienza secuencia de atención

# Secuencia de atención

1. Se completa la instrucción en curso
2. Ciclo de reconocimiento.
3. Preservar estado
4. [ Identificar dispositivo ]
5. Atención al dispositivo
6. Restaurar estado.
7. Retorno



# Interrupciones

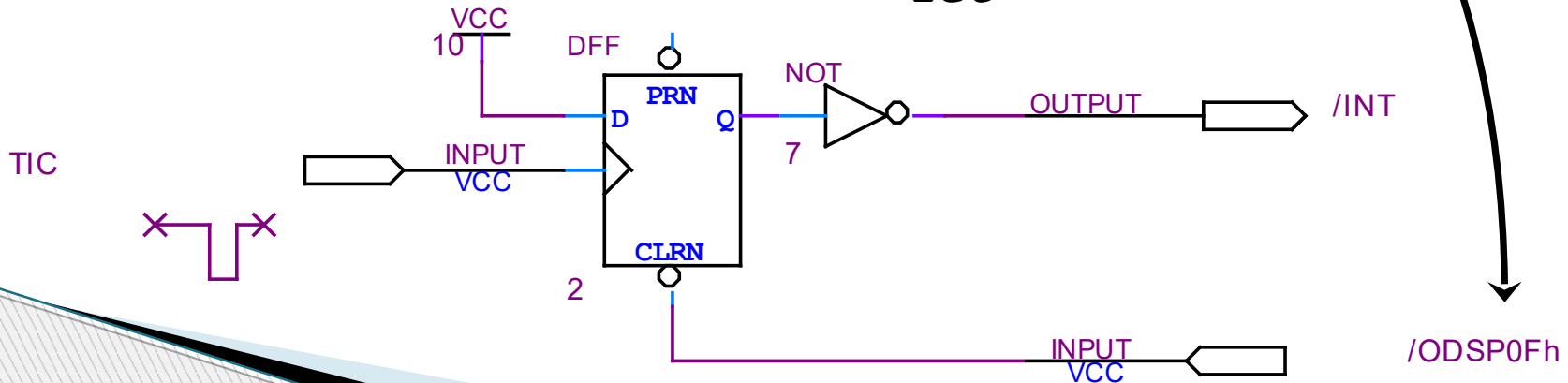
- Una **interrupción** es un mecanismo por el cual se invoca a una “subrutina” en respuesta a un evento de petición iniciado por un dispositivo hardware externo.
  - Mejora el “*throughput*” del sistema (cantidad total de información útil procesada).
  - Reduce la complejidad del programa.
  - Es el dispositivo externo quien inicia la transferencia de datos al  $\mu\text{P}$ .
  - La “subrutina” se le llama “rutina de atención o rutina de servicio a la interrupción o ISR (Interrupt Service Routine)”.
  - La ejecución de la subrutina queda intercalada entre 2 instrucciones consecutivas del programa en curso.
  - La petición de interrupción es asíncrona (ocurre en cualquier instante), por lo tanto, se DEBE salvar SIEMPRE el estado del  $\mu\text{P}$ .

# Ejemplo: Interrupción periódica

- Señal TIC periódica
- Aplicaciones
  - Reloj
  - Conversor A/D muestreo periódico

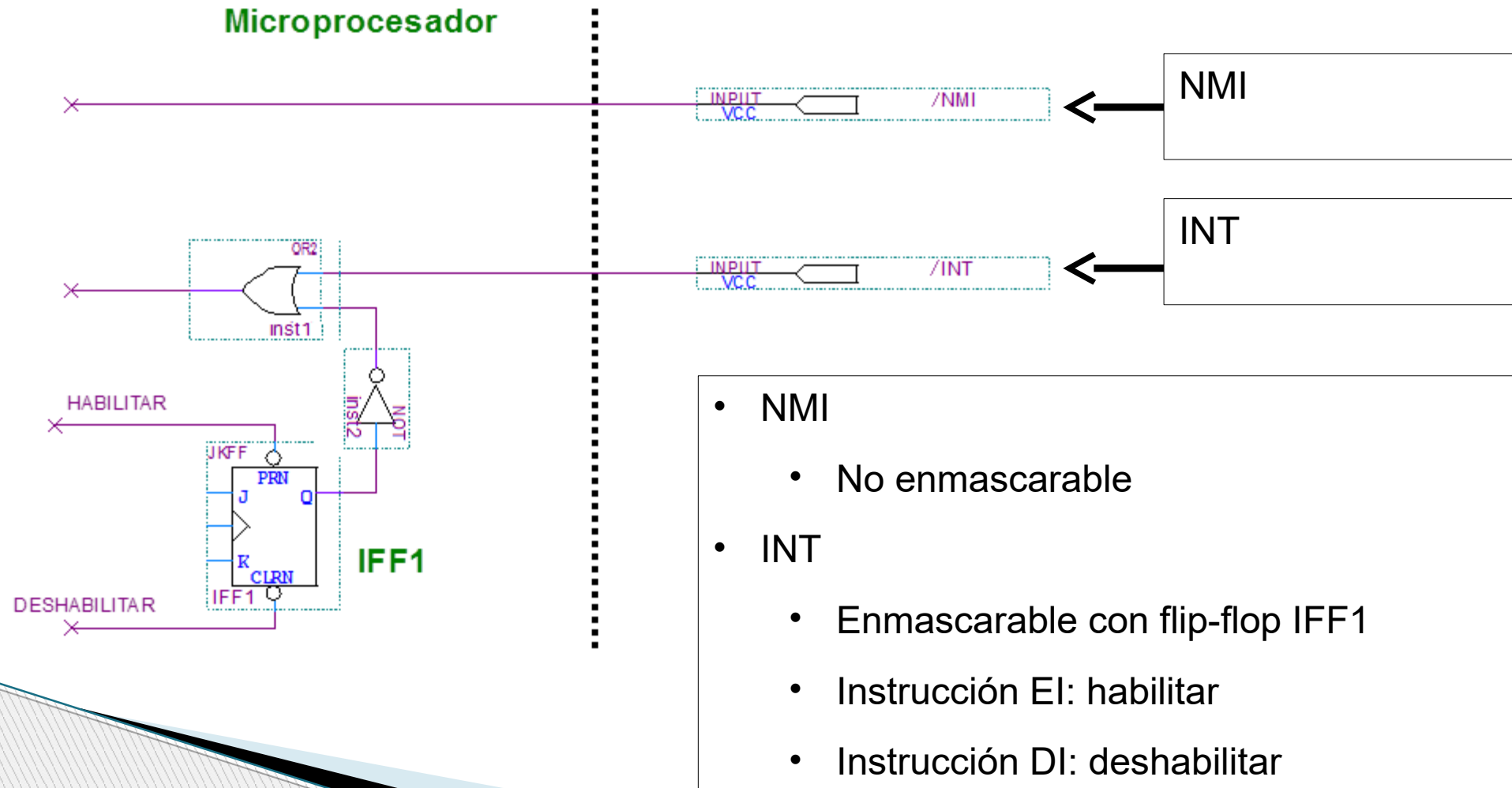
rutint:

```
;preservo estado
push af
;incremento variable
ld a, (VARIABLE)
inc a
ld (VARIABLE), a
; borro FF de petición
out (0Fh), a
; restaura estado y ret
pop af
ei
ret
```



```
rutint:
    ;preservo estado
push af
    ;prendo leds
...
    ;incremento variable
...
    ;espero SW[0] == 1
...
    ;apago leds
...
    ; borro FF de petición
...
    ; restauro estado y ret
pop af
ei
ret
```

# Z80: entradas de Interrupciones



- NMI
  - No enmascarable
- INT
  - Enmascarable con flip-flop IFF1
  - Instrucción EI: habilitar
  - Instrucción DI: deshabilitar

# Z80: Estructura de interrupciones

- *INT: Interrupciones enmascarables*
  - Se pueden deshabilitar
    - DI deshabilita ( $IFF1 \leftarrow 0$ )
    - EI habilita ( $IFF1 \leftarrow 1$ )
  - Se chequea el último T de cada instrucción si  $IFF1 = 1$ .
  - Menor prioridad que NMI
  - Existen 3 modos de interrupciones
    - Modo 0 (modo por defecto)
    - Modo 1
    - Modo 2
  - Tres instrucciones IM 0 / IM 1 / IM 2
    - Se elije en inicialización y no se modifica más.

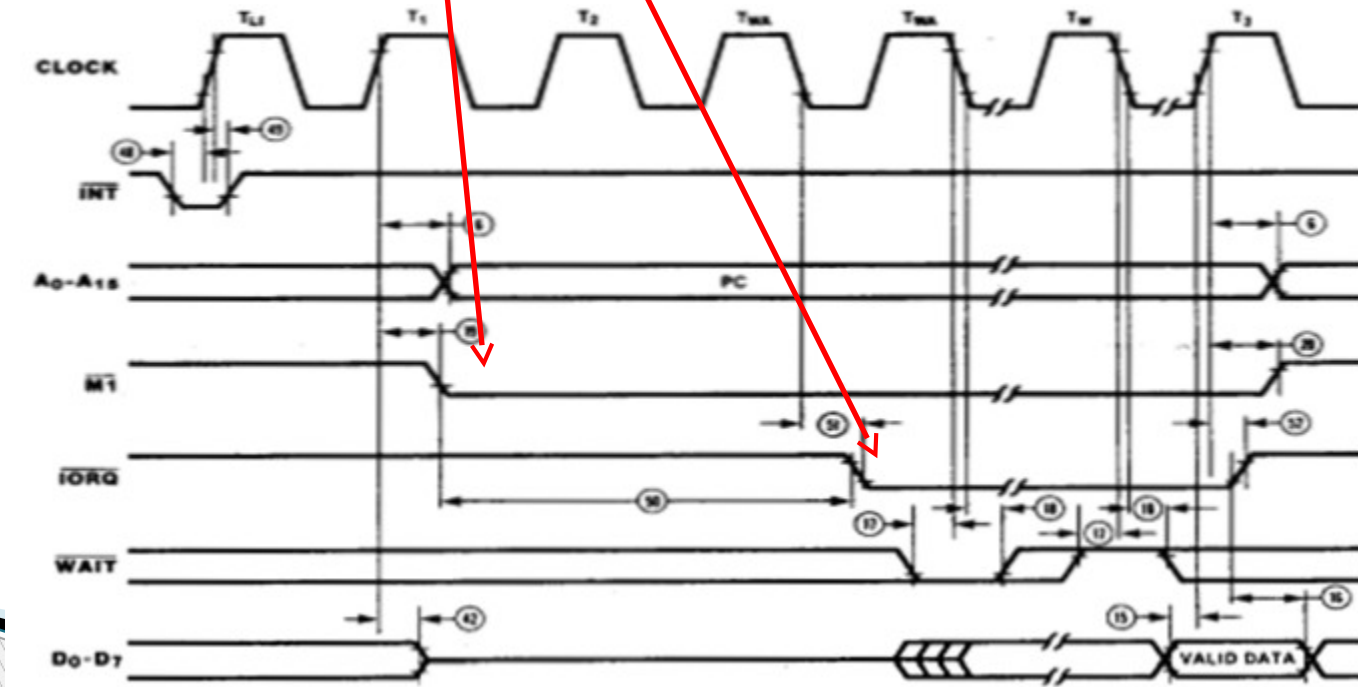


# Z80: Estructura de interrupciones

- INT: Interrupciones enmascarables

- Ciclo de reconocimiento

- El mismo para los 3 modos de interrupciones.
- $IFF1 \leftarrow 0$  (se deshabilitan interrupciones)
- Señal  $/INTA = /M1$  or  $/IORQ$
- Lo que sucede en este ciclo (y después) depende del modo.



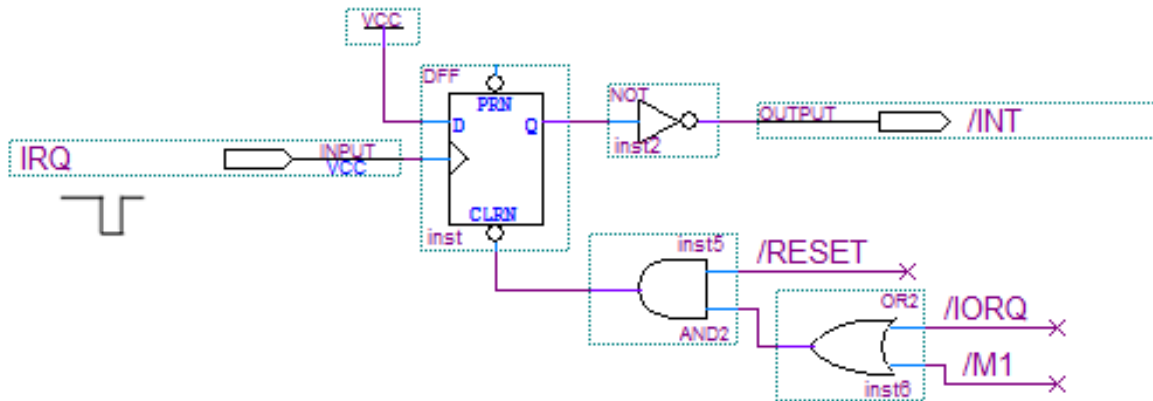
# Z80: Interrupciones enmascarables

- MODO 1

- Se selecciona con **IM1** en la inicialización.
- Rutina de atención en **dirección fija 0x0038**
- Pensada para **un solo dispositivo** que interrumpe.
- Secuencia
  1. Ciclo de reconocimiento (5T);  $IFF1 \leftarrow 0$  (deshabilita int.)
  2.  $Stack \leftarrow$  Dirección retorno (6T);  $PC \leftarrow 0x0038$
  3. ----- se ejecuta la rutina -----
  4. RET ( $PC \leftarrow Stack$ ) (6T)

# Z80: Estructura de interrupciones

- Ejemplo: 1 dispositivo que interrumpe en MODO 1



```
org 38H
rutint:  push ...
.....
.....
.....
.....
.....
pop ...
ei
ret
```

- En el circuito, el FF de petición se borra:
  - Luego de un Reset
  - En el ciclo de reconocimiento (/IORQ + /M1)
- Alternativa:
  - Usar ODSP de un puerto como vimos en ejemplo contador

# Z80: Interrupciones enmascarables

- MODO 0

- Permite **varios periféricos**.
- Modo por defecto luego de un Reset.
- Compatibilidad con 8080 de Intel
- El Z80 durante el ciclo de INTA:
  - lee el bus de datos, lo interpreta como un OPPOSITE.
  - Ejecuta la instrucción
- En general el OPPOSITE es una **instrucción RST**
  - Hasta 7 periféricos
- Con instrucción CALL
  - Más periféricos
  - Z80 genera 3 ciclos INTA para completar la instrucción.

# Z80: Interrupciones enmascarables

## • Modo 2. Tabla de interrupciones

- Ocupa 256 bytes (128 direcciones)
- Comienza en la dir dada por el **registro I** multiplicado por 0x0100
  - Ej I = 0x23 → La tabla comienza en 0x2300
- Durante el ciclo INTA el Z80 lee el **vector de interrupciones**.
- Este vector es el desplazamiento en la tabla, donde se encuentra la dirección de comienzo de la rutina de atención a la interrupción.

### Ejemplo:

Sea I = 0x23 y la tabla.

Si vector de interrupciones = A0 → la dirección de comienzo de la rutina de atención a la interrupción es 0x**BC9A**, almacenada en la dirección 0x23A0 y siguiente

0x23A1	<b>BC</b>
0x23A0	<b>9A</b>
-----	--
-----	--
0x2300	

# Z80: Interrupciones enmascarables

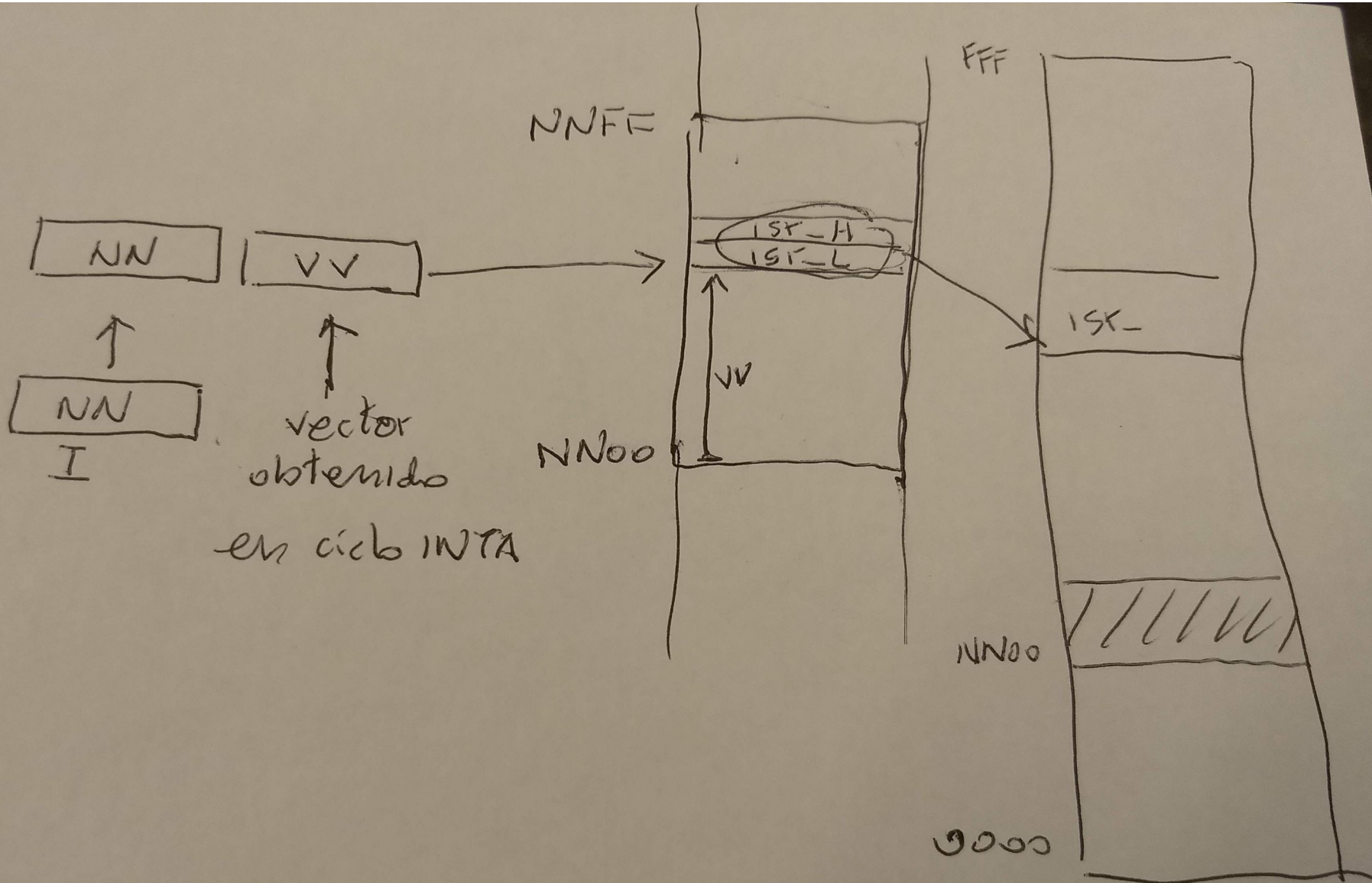
- MODO 2

- Debe ejecutarse la instrucción IM2 en la inicialización.
- Permite hasta 128 interrupciones diferentes.
- Utiliza la **tabla de interrupciones** definida anteriormente (**registro I**)
- El Z80 lee el bus de datos durante el ciclo de INTA y lo interpreta como un **vector de interrupciones**.

- Secuencia

1. Ciclo de reconocimiento (5T);  $IFF1 \leftarrow 0$  (deshabilita int.); Z80 lee vector desde el bus
2.  $Stack \leftarrow$  Dirección retorno (6T);
3.  **$PC \leftarrow$  lee dir. de comienzo de la rutina de la tabla de interrupciones (6T).**
4. ----- se ejecuta la rutina -----
5. RETI ( $PC \leftarrow Stack$ ) (6T)
  - DEBE ser RETI, no puede ser RET

# Z80: MODO 2



# Z80: Estructura de interrupciones

- *NMI: Interrupciones NO enmascarables*
  - No se pueden deshabilitar.
  - Es chequeada en el último T de la instrucción en curso si ocurrió un flanco de bajada y se mantiene  $NMI = 0$ .
  - Una única interrupción
  - De mayor prioridad
  - La rutina de atención se ejecuta en la dir **0x0066**

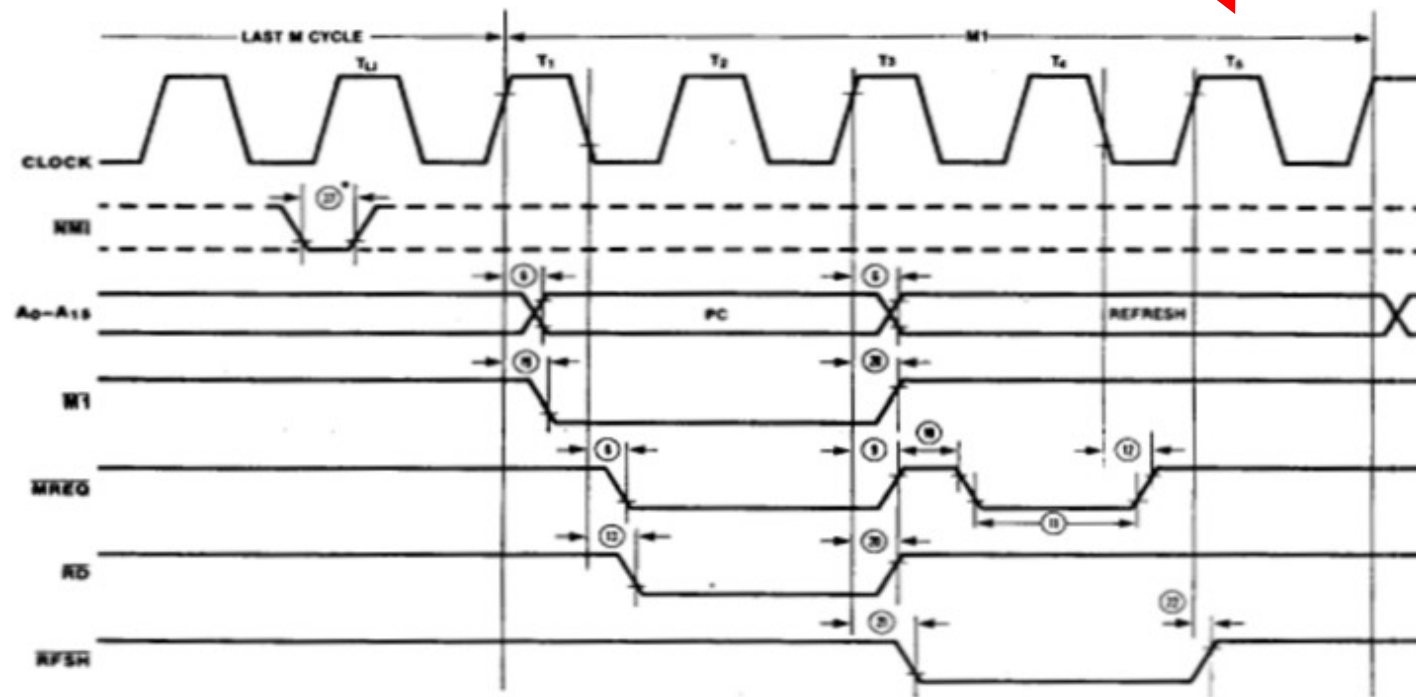


# Z80: Estructura de interrupciones

- NMI: Interrupciones NO enmascarables

- Ciclo de reconocimiento

- Es idéntico a M1 pero:
  - Se descarta el OP CODE
  - Agrega un T al final (demora 5T).



# Z80: Estructura de interrupciones

- *NMI: Interrupciones NO enmascarables*

- Secuencia

1. Ciclo de reconocimiento (5T)

IFF2  $\leftarrow$  IFF1

IFF1  $\leftarrow$  0 (deshabilita interrupciones enmascarables)

2. Stack  $\leftarrow$  Dirección retorno (6T)

PC  $\leftarrow$  0x0066

3. ----- se ejecuta la rutina -----

4. Retorno: RETN (6T)

- PC  $\leftarrow$  Stack

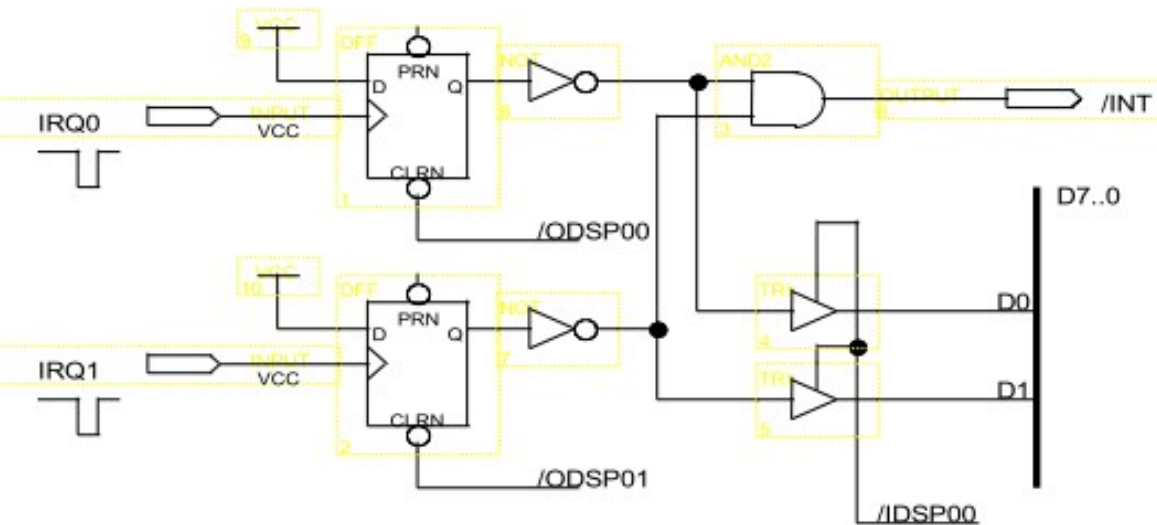
- IFF1  $\leftarrow$  IFF2

# Z80: Estructura de interrupciones

- *NMI: Interrupciones NO enmascarables*
  - Puedo querer dejar las interrupciones deshabilitadas
  - EI y DI afectan ambos IFF
    - DI
      - $IFF1 \leftarrow 0$
      - $IFF2 \leftarrow 0$
    - EI
      - $IFF1 \leftarrow 1$
      - $IFF2 \leftarrow 1$

# Z80: Estructura de interrupciones

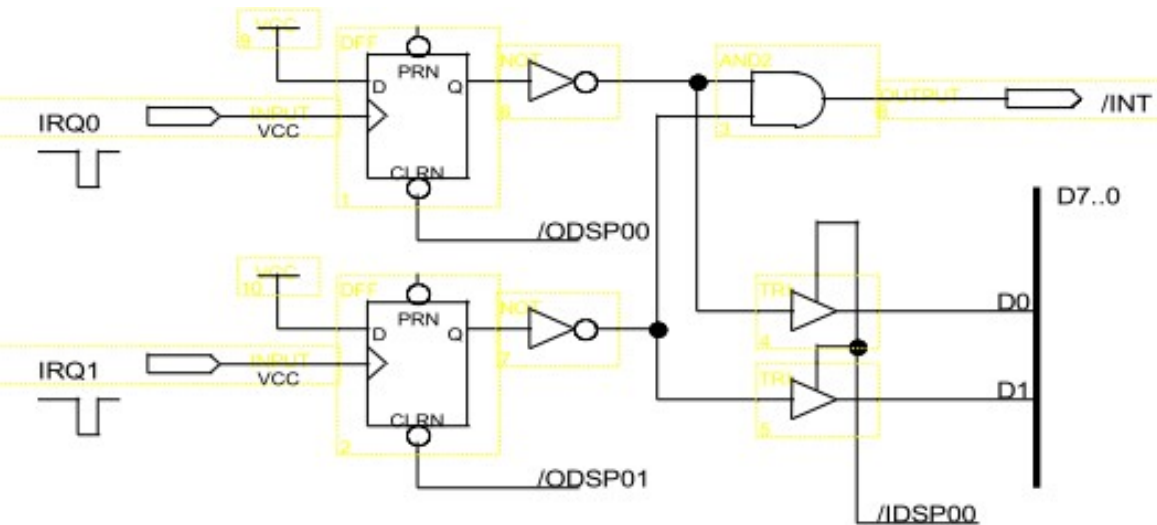
- Ejemplo: 2 dispositivos que interrumpen en MODO 1



- Los FF de petición se borran por SW.

# Z80: Estructura de interrupciones

- Ejemplo: 2 dispositivos que interrumpen en MODO 1



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

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

        org 38H
Rutint:
    push af
        in a,
        (pendientes)
        bit 0, a
        jr z, atiando0
atiendo1:
    ei
    out (borro1), a
    call isr1
    pop af
    ret
atiendo0:
    out (borro0), a
    call isr0
    pop af
    ei
    ret
```

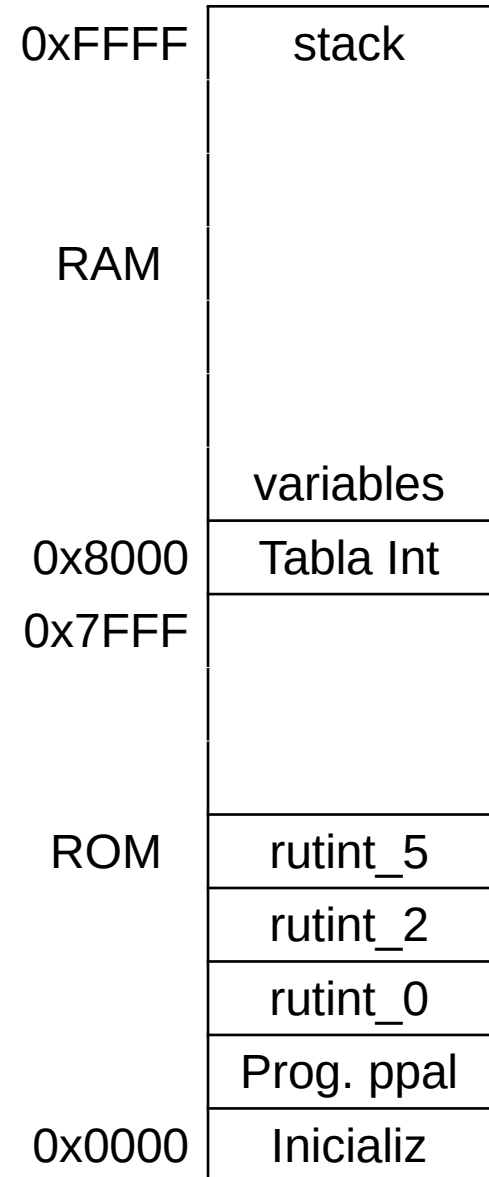
# Z80: Estructura de interrupciones

- Ejemplo:

## Inicialización en MODO 2

- Sistema con 32kRAM y 32kROM
- Sean 3 dispositivos cuyos vectores son el 0<sup>vo</sup>, el 2<sup>do</sup> y el 5<sup>to</sup>
- Las rutinas de atención son rutint\_0, rutint\_2 y rutint\_5

Solución con la tabla en RAM



# Z80: Estructura de interrupciones

- Ejemplo: Inicialización en MODO 2

- Sistema con 32kRAM y 32kROM
- Sean 3 dispositivos cuyos vectores son el 0<sup>vo</sup>, el 2<sup>do</sup> y el 5<sup>to</sup>
- Las rutinas de atención son rutint\_0, rutint\_2 y rutint\_5

## Solución con la tabla en RAM:

```
ORG 0x8000
TABLA: DS 256 ; Reservo memoria para la Tabla de Interr.

ORG 0x0000
LD SP, 0 ; inicializo SP en la parte sup de la RAM + 1
LD A, Tabla / 256
LD I, A ; tabla de interr. al comienzo de la RAM
LD BC, rutint_0
LD (TABLA), BC ; cargo rutint_0 en el lugar 0 de la tabla
LD BC, rutint_2
LD (TABLA+2*2), BC ; cargo rutint_2 en el lugar 2 de la tabla
LD BC, rutint_5
LD (TABLA+2*5), BC ; cargo rutint_5 en el lugar 5 de la tabla
IM 2 ; modo 2 de interrupciones
... ; inicialización de otros dispositivos
EI ; habilito interrupciones
```

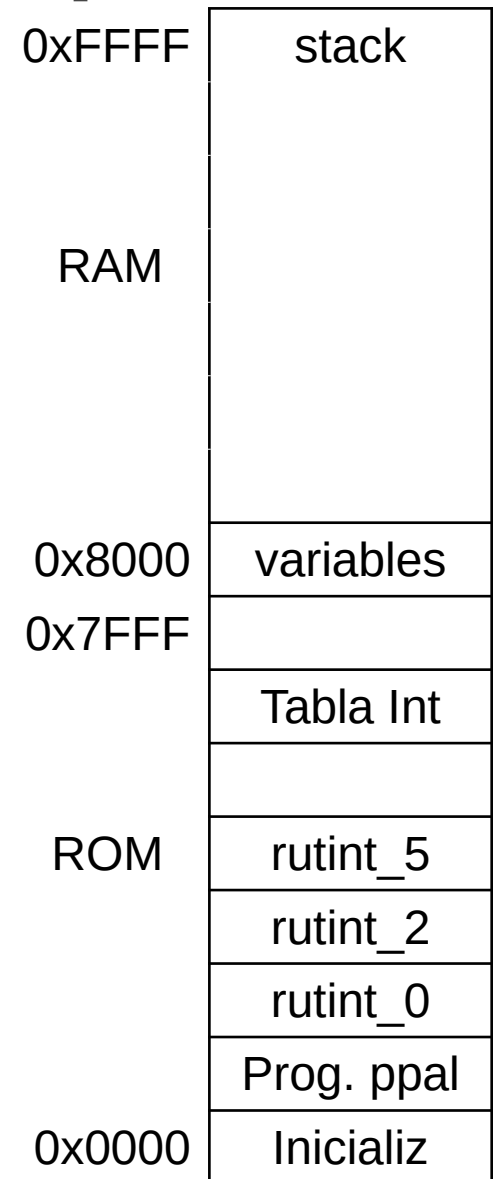
# Z80: Estructura de interrupciones

- Ejemplo:

## Inicialización en MODO 2

- Sistema con 32kRAM y 32kROM
- Sean 3 dispositivos cuyos vectores son el 0<sup>vo</sup>, el 2<sup>do</sup> y el 5<sup>to</sup>
- Las rutinas de atención son rutint\_0, rutint\_2 y rutint\_5

Solución con la tabla en ROM





# Z80: Estructura de interrupciones

- Ejemplo: Inicialización en MODO 2

- Sistema con 32kRAM y 32kROM
- Sean 3 dispositivos cuyos vectores son el 0<sup>vo</sup>, el 2<sup>do</sup> y el 5<sup>to</sup>
- Las rutinas de atención son rutint\_0, rutint\_2 y rutint\_5

Solución con la tabla en ROM:

```
ORG 0x0000
LD SP, 0x0000    ; inicializo SP al fin de RAM + 1
LD A, TABLA / 256
LD I, A ; tabla de interr. en dir 0x2000 en ROM
IM 2    ; modo 2 de interrupciones
... ; inicialización de otros dispositivos
EI    ; habilito interrupciones
...
ORG 0x2000    ; en ROM
TABLA: DW rutint_0 ; directivas para grabar la tabla en la ROM
        DW
        DW rutint_2
        DW
        DW
        DW rutint_5
```