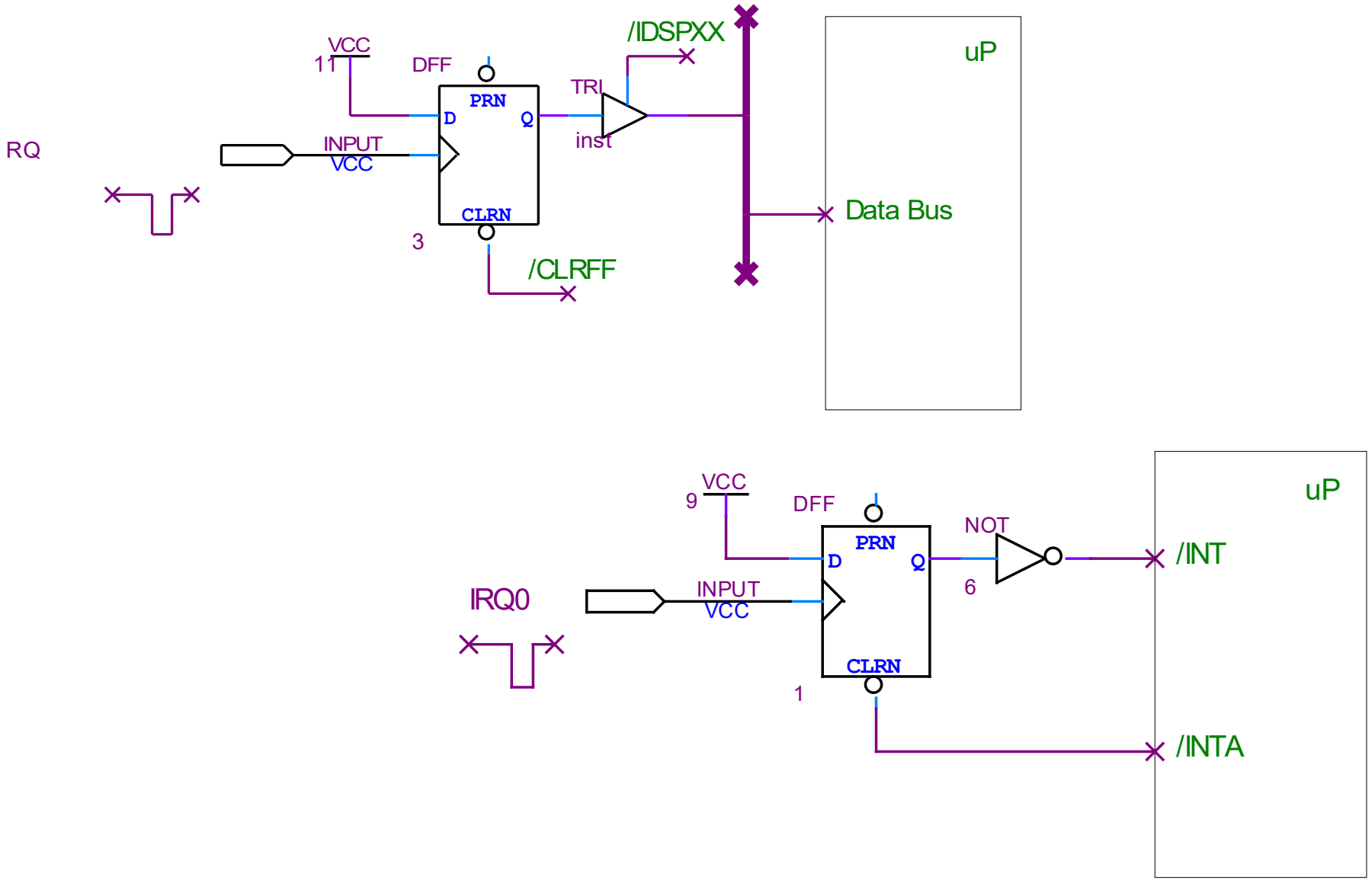
Interrupciones

- Notas en el EVA:
  - Material
    - Notas sobre temas del curso
      - interrupciones-v1.pdf

Interrupciones



Interrupciones

- Secuencia de atención
  1. Se completa instrucción.
     - Se termina de ejecutar la instrucción en curso.

Interrupciones

- Secuencia de atención
  1. Se completa instrucción.
  2. Ciclo de reconocimiento.
     - Señal hardware de reconocimiento para avisar al periférico que va a ser atendido.
     - Dirección de retorno al stack para poder retornar.
     - [En algunos casos se identifica cuál dispositivo pidió interrupción para determinar cuál rutina de atención.]
     - Se salta a dirección de comienzo de subrutina de atención.
     - [En algunos casos se borra el FF de petición.]

Interrupciones

- Secuencia de atención
  1. Se completa instrucción.
  2. Ciclo de reconocimiento.
  3. Preservar estado
     - Registros y banderas, usualmente al stack.
     - No se puede saber para qué los estaba usando el programa principal.
     - Responsabilidad del programador

Interrupciones

- Secuencia de atención
  1. Se completa ejecución de instrucción.
  2. Ciclo de reconocimiento.
  3. Preservar estado
  4. [Identificar dispositivo]
     - Si no se hizo por hardware en (2), el software debe identificar cuál de los dispositivos interrumpió.

Interrupciones

- Secuencia de atención
  1. Se completa ejecución de instrucción.
  2. Ciclo de reconocimiento.
  3. Preservar estado
  4. [Identificar dispositivo]
  5. Atención al dispositivo
     - Finalmente podemos atender la solicitud del dispositivo.
     - En algunos casos debemos además borrar el FF de petición (si no se hizo en paso 2).

Interrupciones

- Secuencia de atención
  1. Se completa ejecución de instrucción.
  2. Ciclo de reconocimiento.
  3. Preservar estado
  4. [Identificar dispositivo]
  5. Atención al dispositivo
  6. Restaurar estado
     - Se debe dejar registros y banderas como los tenía el programa principal cuando fue interrumpido.
     - En general además EI
     - Responsabilidad del programador

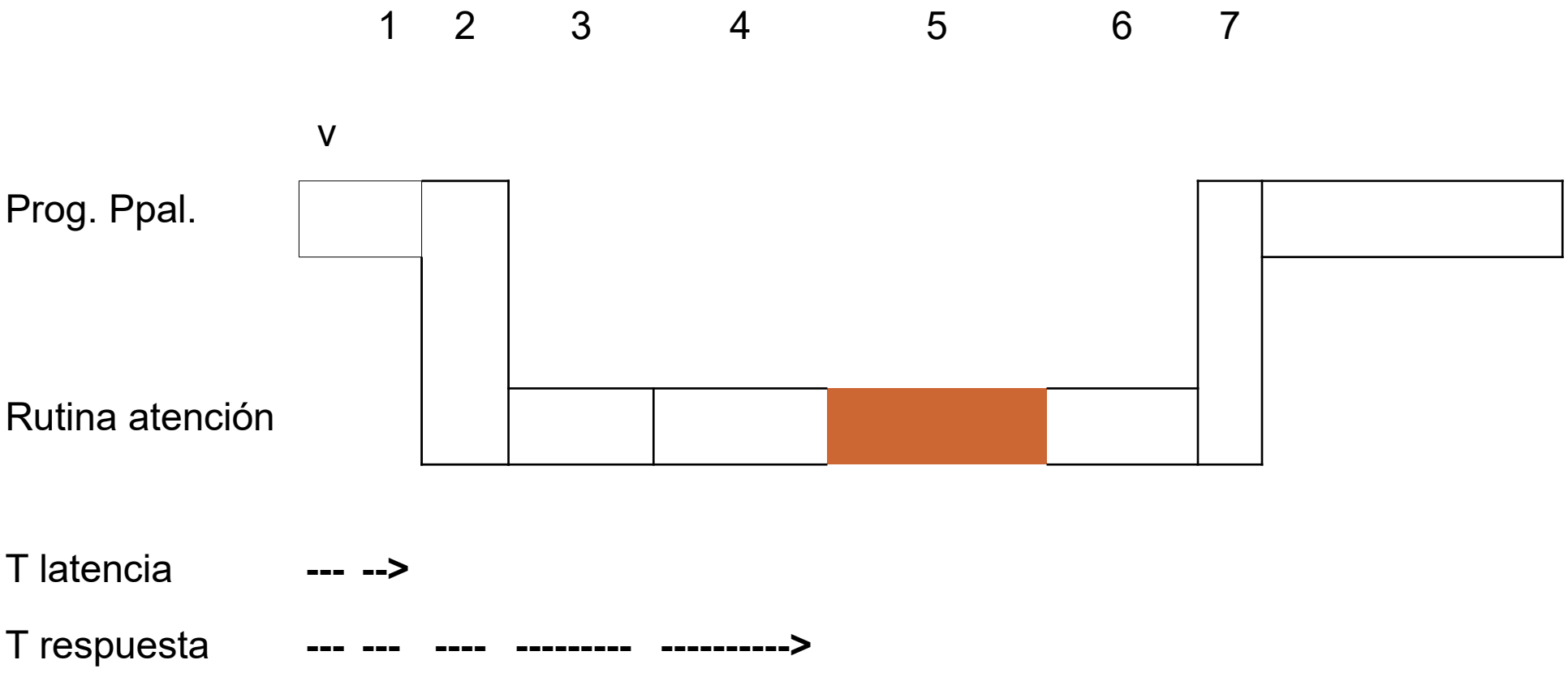
Interrupciones

- Secuencia de atención
  1. Se completa ejecución de instrucción.
  2. Ciclo de reconocimiento.
  3. Preservar estado
  4. [Identificar dispositivo]
  5. Atención al dispositivo
  6. Restaurar estado
  7. Retorno
     - Con alguna variante de instrucción RET

Interrupciones

- Tiempo de latencia:
  - desde solicitud hasta que procesador la reconoce.
- Tiempo de respuesta:
  - desde solicitud hasta que es efectivamente atendido.
- Deadline
  - Instante a partir del cual si la petición no fue aún atendida se degrada el funcionamiento del sistema.
- Ejemplo
  - Reloj implementado con interrupción periódica.
  - Si no se atiende antes de la siguiente solicitud se pierde un período y el reloj atrasa.

Secuencia de atención



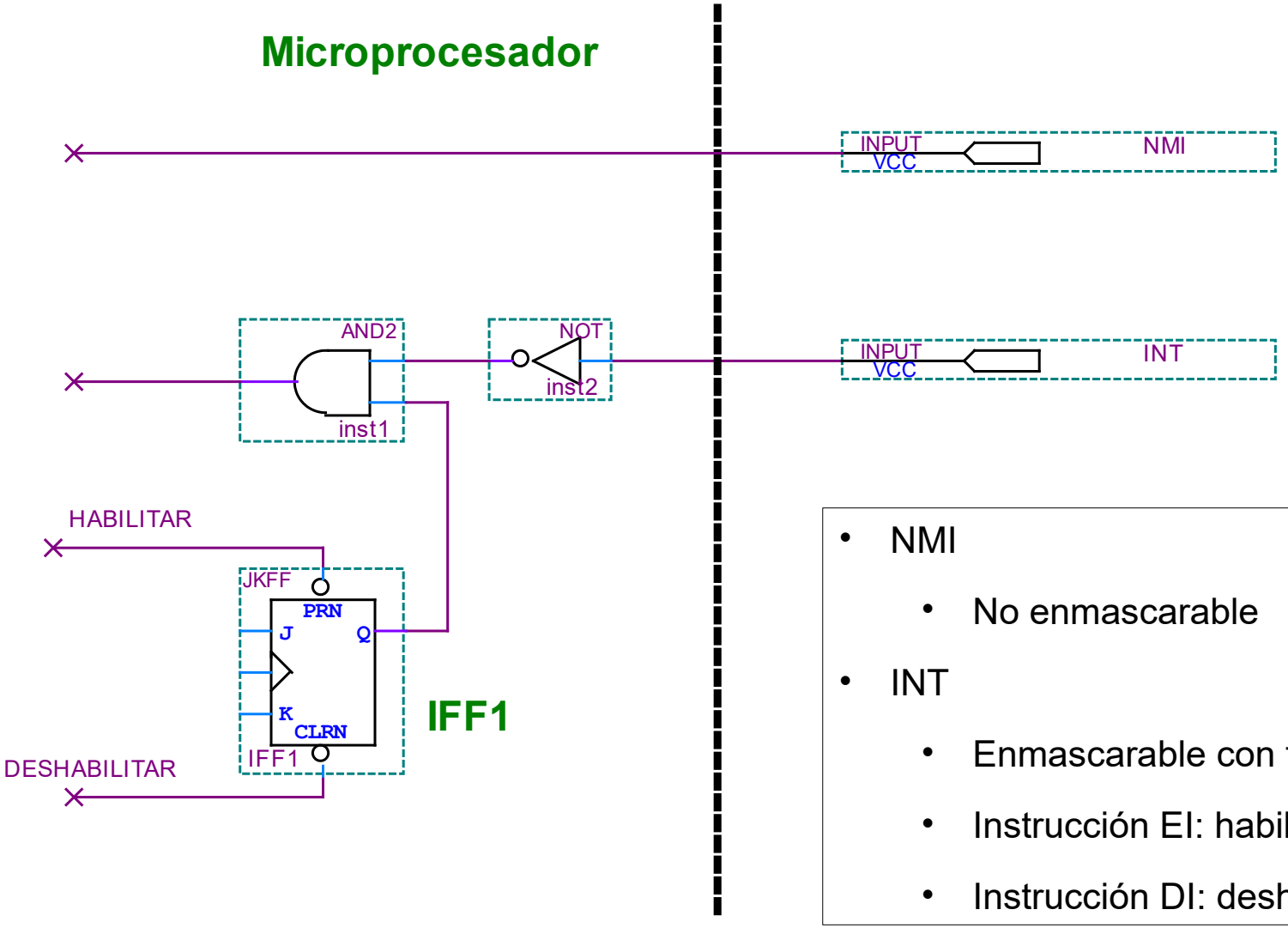
Interrupciones

- 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 µP.
  - La “subrutina” se le llama rutina “de atención” o “de servicio” a la interrupción” o ISR (Interrupt Service Rutine).
  - 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 µP.

Interrupciones enmascarables

- A veces necesito no atender interrupciones
- Para inicializaciones varias
  - Stack
  - Hardware externo
  - Microprocesador
  - Variables
- Para proteger una sección de programa y evitar que sea interrumpida (sección crítica)
  - Código que se debe ejecutar lo más rápido posible.
  - Código que debe demorar un tiempo determinado.
  - Código que accede a datos que también son manipulados por la interrupción.

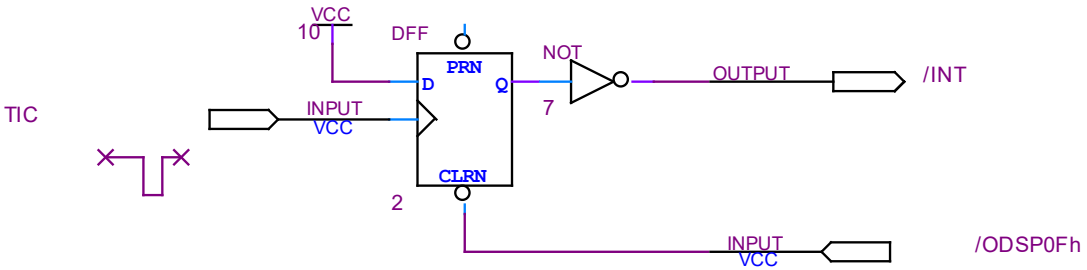
Interrupciones enmascarables



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

Ejemplo: Interrupción periódica

- Señal tic periódica
  - Reloj tiempo real
  - Muestreo en conv. A/D




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

Interrupciones

- En el ejemplo anterior:
  - FF de petición se borra dentro de la subrutina
  - Por tanto, /INT activa y se sigue pidiendo interrupción al ejecutar PUSH AF
  - ¿Por qué no se vuelve a interrumpir al final de PUSH AF?
- Durante el ciclo de reconocimiento se deshabilitan interrupciones.
- Se debe volver a habilitarlas antes de retornar
  - La instrucción EI habilita las interrupciones.