

E/S controlada por programa

- Métodos de E/S
- Puertos
- Transferencia de datos
 - Incondicional
 - Condicional (*Handshaking*)
- *Polling*

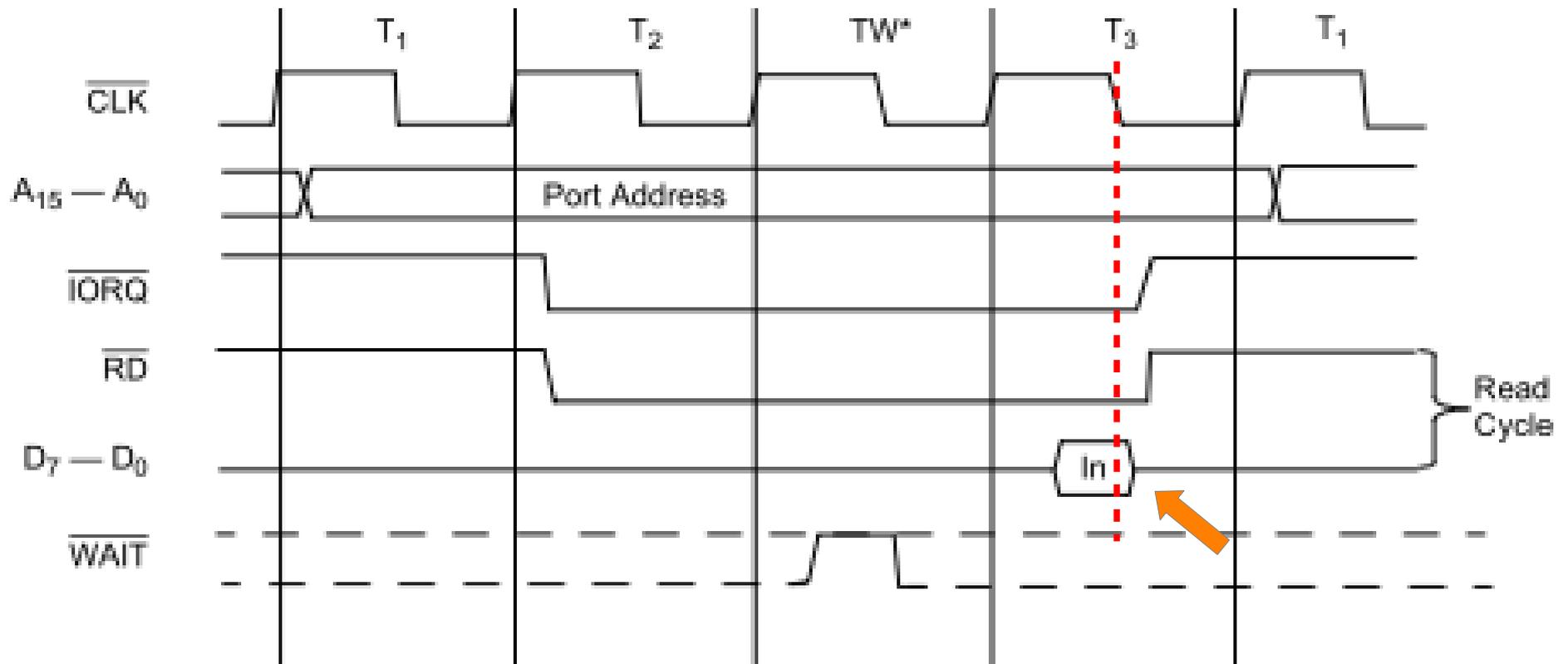
Métodos de E/S

- Controlada por programa
 - El programa decide cuándo
 - El programa realiza la transferencia de datos
- por Interrupciones
 - Iniciada por el hardware externo sin intervención del programa
 - Realizada por programa (rutina de servicio a la interrupción).
- Controlada por hardware (DMA)
 - Iniciada y realizada por el hardware externo.
 - El programa inicializa dir en memoria, cant de bytes, etc.²

Mapeo de Puertos

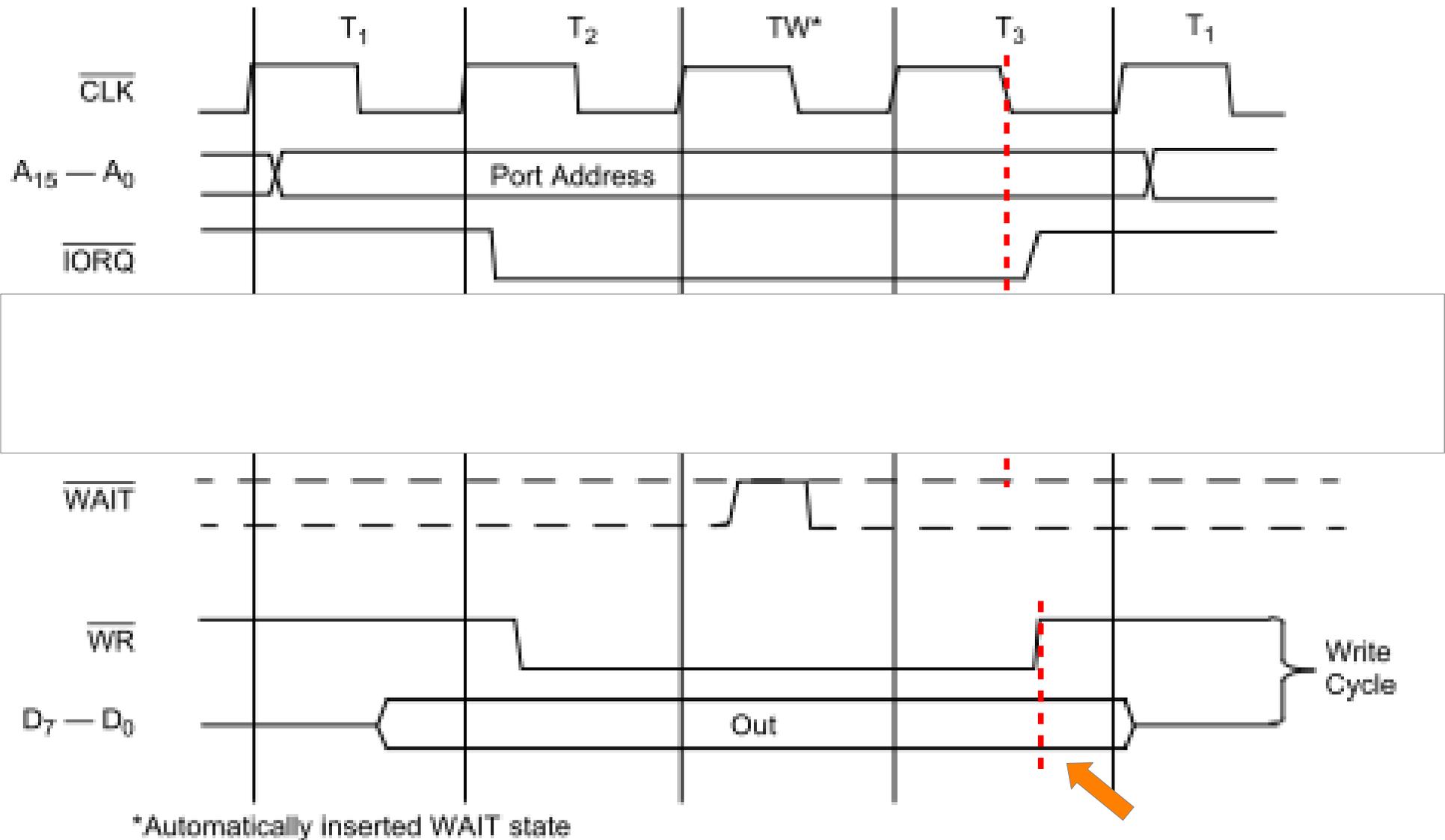
- E/S aislada:
 - espacio de direcciones separado
 - En Z80
 - Se diferencia con /IORQ
 - 256 puertos entrada y 256 puertos salida
 - Instrucciones específicas (IN y OUT)
- E/S mapeada en memoria
 - Se decodifican direcciones de memoria
 - Instrucciones de acceso a memoria
- Hay procesadores que no tienen espacio separado
 - Si lo tiene, entonces puedo elegir

Ciclos M. Entrada y Salida



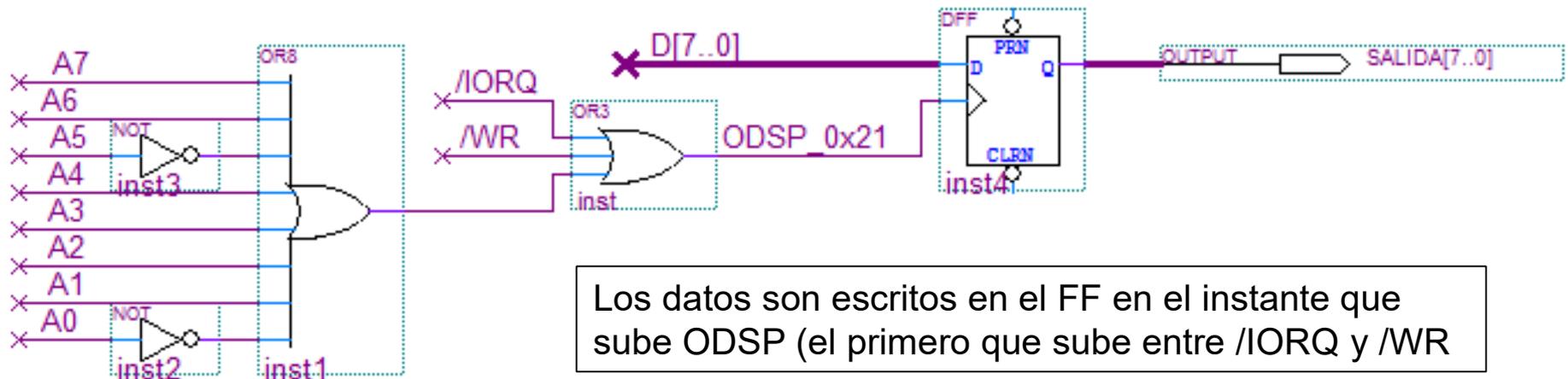
*Automatically inserted WAIT state

Ciclos M. Entrada y Salida



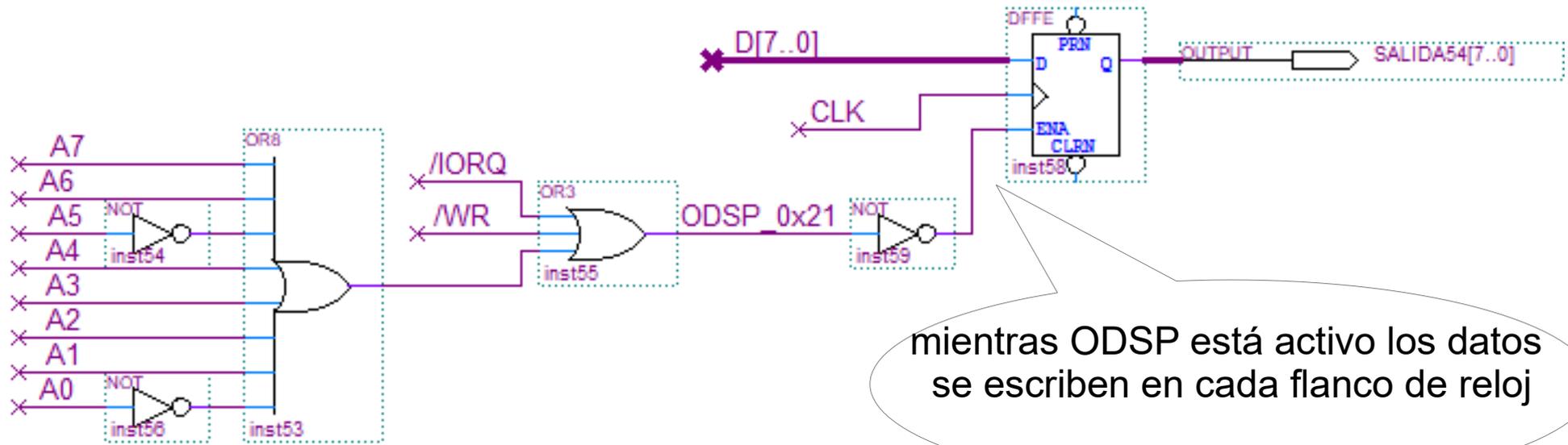
Puerto salida = REGISTRO

- **Elemento de memoria** que capture contenido del bus de datos en el instante en que hay datos presentes.
- Pulso **/ODSP_xx** a partir de /IORQ, /WR y A7..0
- **/ODSP**: **Output Device Select Pulse** (activo por nivel bajo)



Puerto salida = REGISTRO

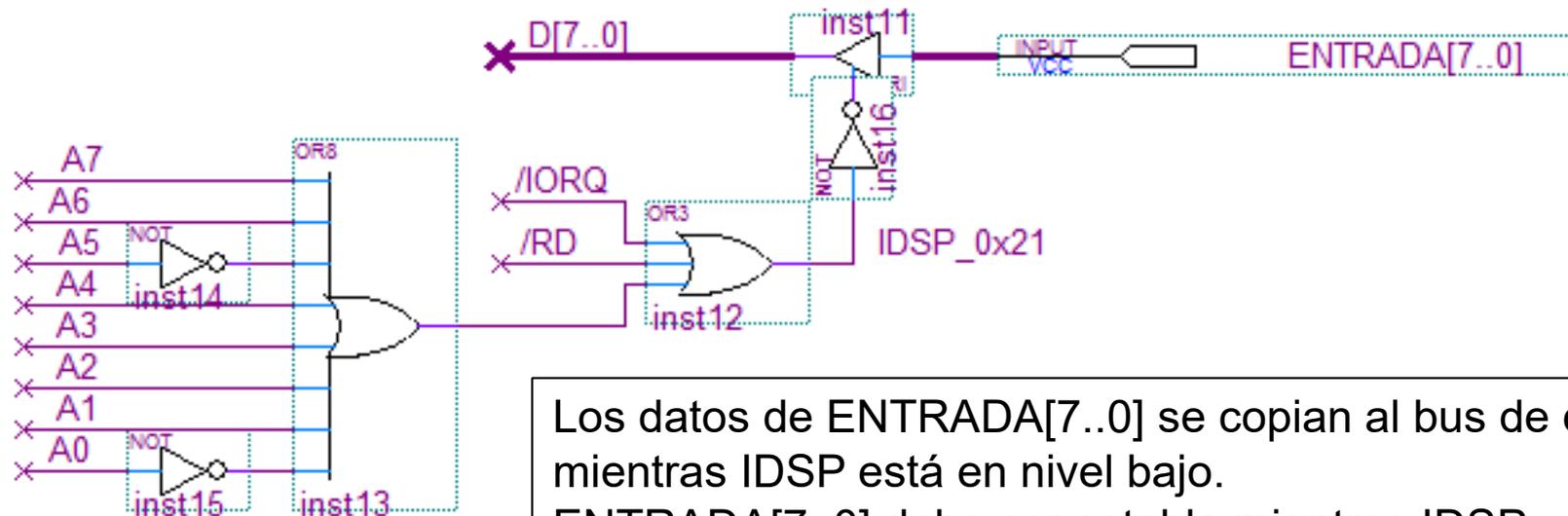
- Caso FF D con Enable
 - ODSP activo en 1 habilita
 - CLK manejado por reloj del sistema



mientras ODSP está activo los datos se escriben en cada flanco de reloj

Puerto entrada = Buffer tristate

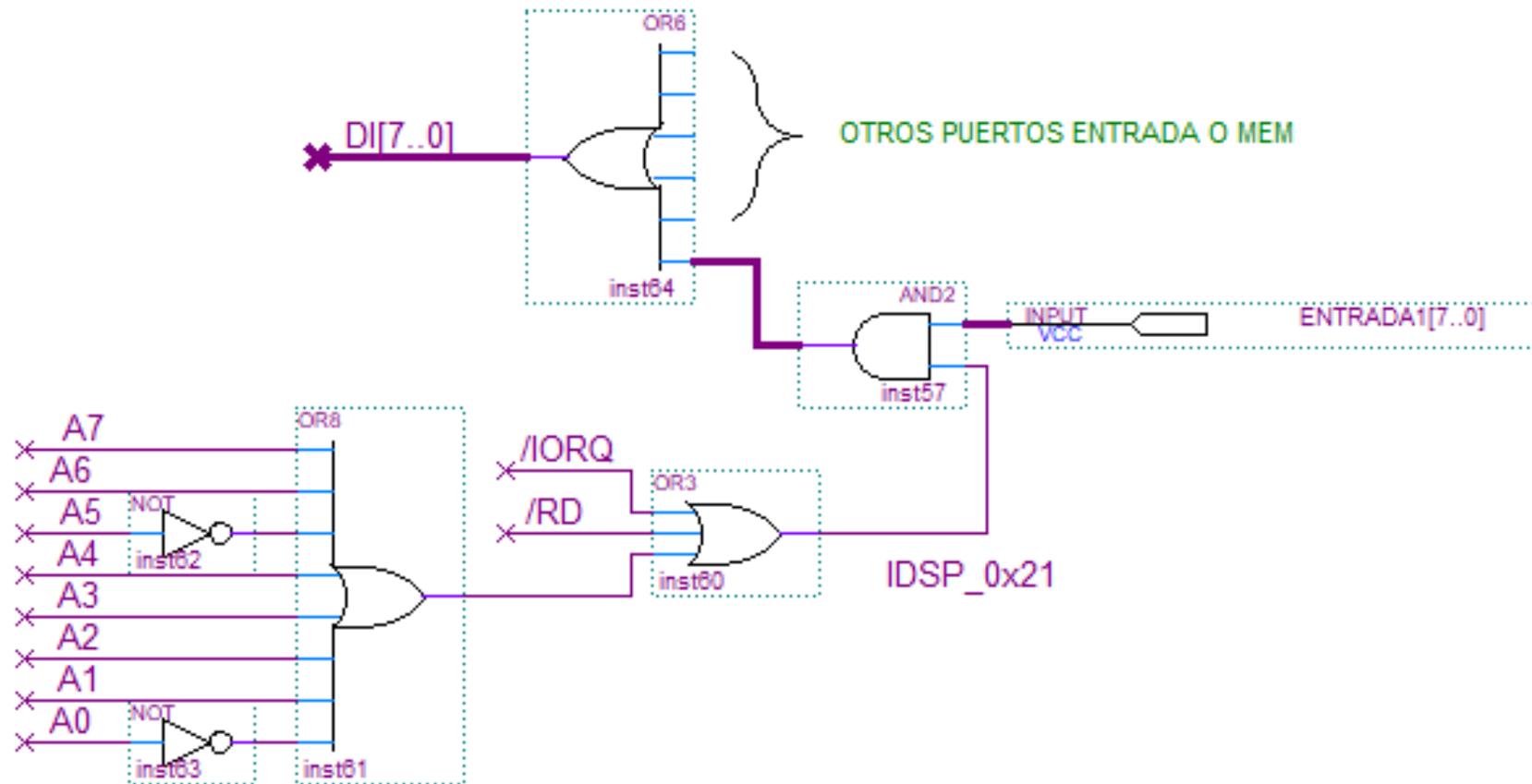
- Buffer triestado en buses triestado.
- Compuertas AND/OR en buses multiplexados
- Pulso IDSP_xx a partir de /IORQ, /RD y A7..0
- /IDSP: Input Device Select Pulse



Los datos de `ENTRADA[7..0]` se copian al bus de datos mientras `IDSP` está en nivel bajo.
`ENTRADA[7..0]` debe ser estable mientras `IDSP = 0`.

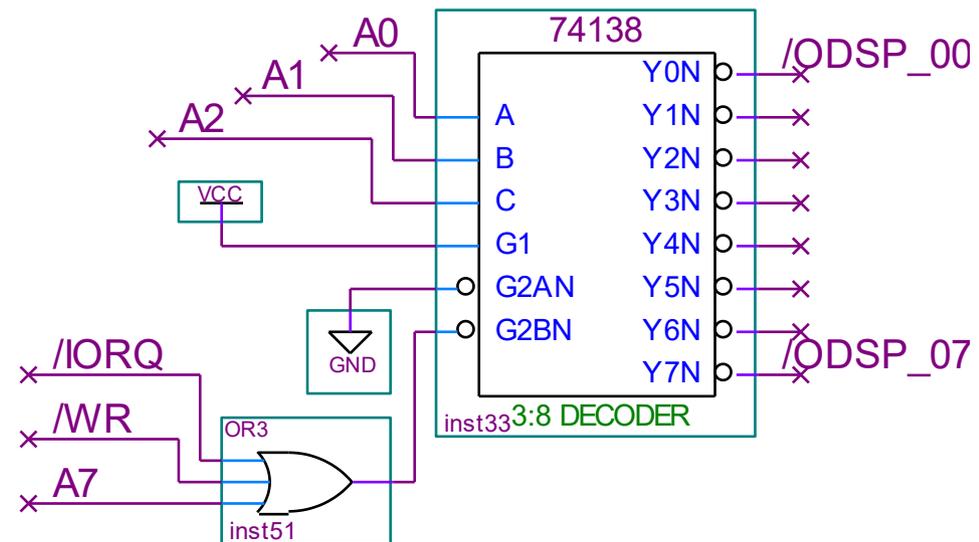
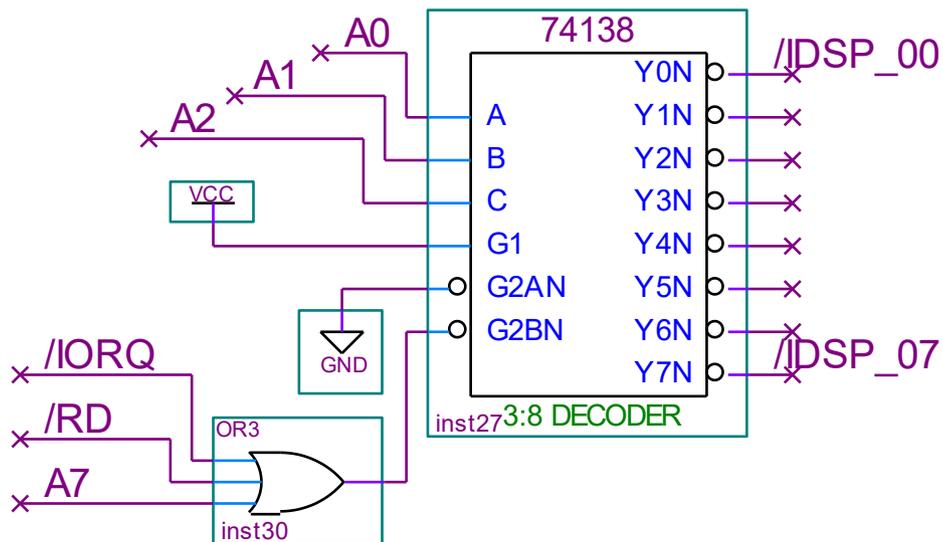
Puerto entrada

- Buffer triestado en buses separados.
- Compuertas AND/OR en buses multiplexados



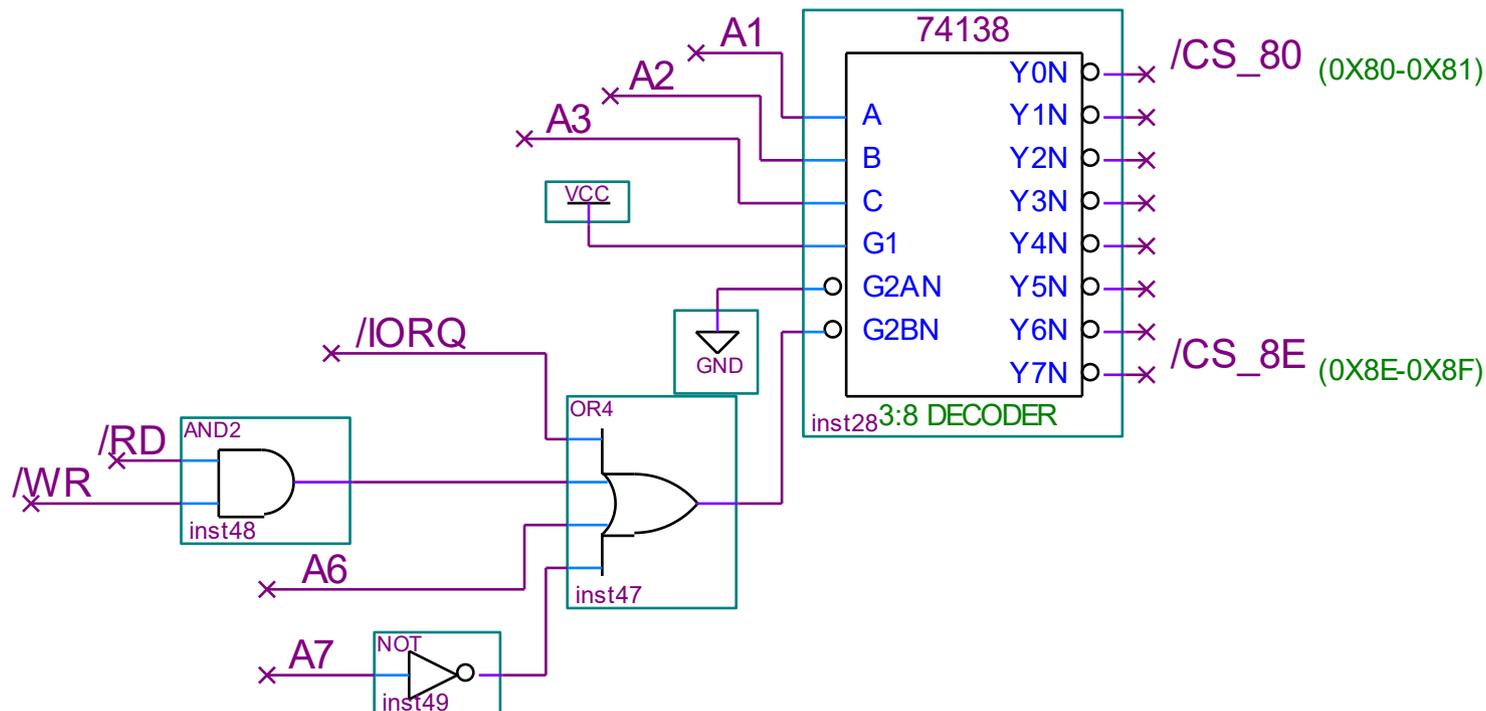
Decodificación y Mapas I/O

- Intervienen **direcciones** y señales de control
 - Dispositivos solo entrada: **/IORQ** con **/RD**
 - Dispositivos solo salida: **/IORQ** con **/WR**



Decodificación y Mapas I/O

- Dispositivos de entrada y salida
 - p. ej. una pequeña memoria mapeada en E/S
 - Evitar habilitar en ciclos rec. de interrupción
 - /IORQ con /RD o con /WR



Mapas I/O



- Similar a lo visto para mapa de memoria:
 - Se indica dirección o rango en que se accede cada dispositivo
 - También repeticiones fantasma

E/S controlada por programa

- ✓ Métodos de E/S
- ✓ Puertos
- Transferencia de datos
 - Incondicional
 - Condicional (*Handshaking*)
- *Polling*

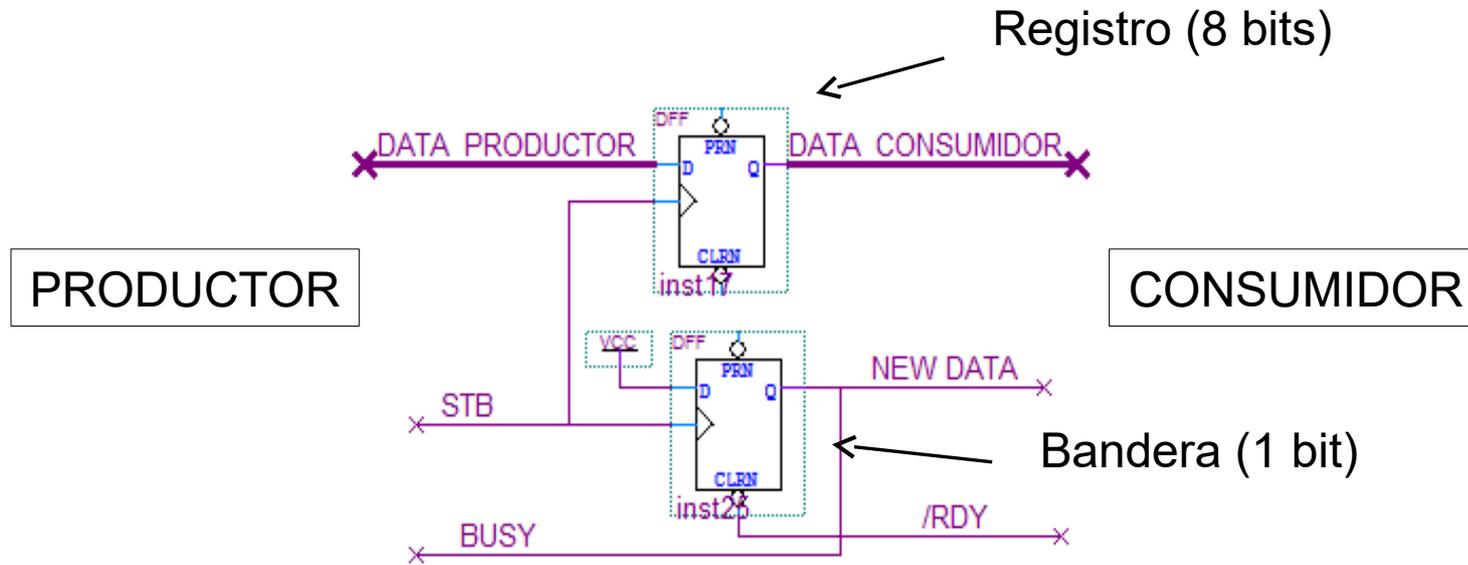
Transferencias de datos

- Transferencia incondicional
 - Se presupone que el dispositivo está listo
 - Cuando el programa quiere hacer una transferencia, la hace sin más
- Ejemplos salidas
 - Indicadores luminosos
 - Configuración a un periférico
- Ejemplos entradas
 - Llaves (SW en placa lab)
 - Status de un periférico

Transferencias de datos

- Transferencia condicional (**HANDSHAKING**)
 - Consulta previa al dispositivo para ver si está pronto.
- Sincronización productor-consumidor
 - Productor genera datos a su velocidad
 - Consumidor los procesa de a una palabra por vez, le lleva un tiempo cada palabra.
 - ¿Cómo sincronizarlos?
- Ej. productor puede ser un teclado
 - Si leo dos veces seguidas una letra “U”, cómo distingo el caso tecla apretada todo el tiempo del caso se presionó dos veces la misma tecla?

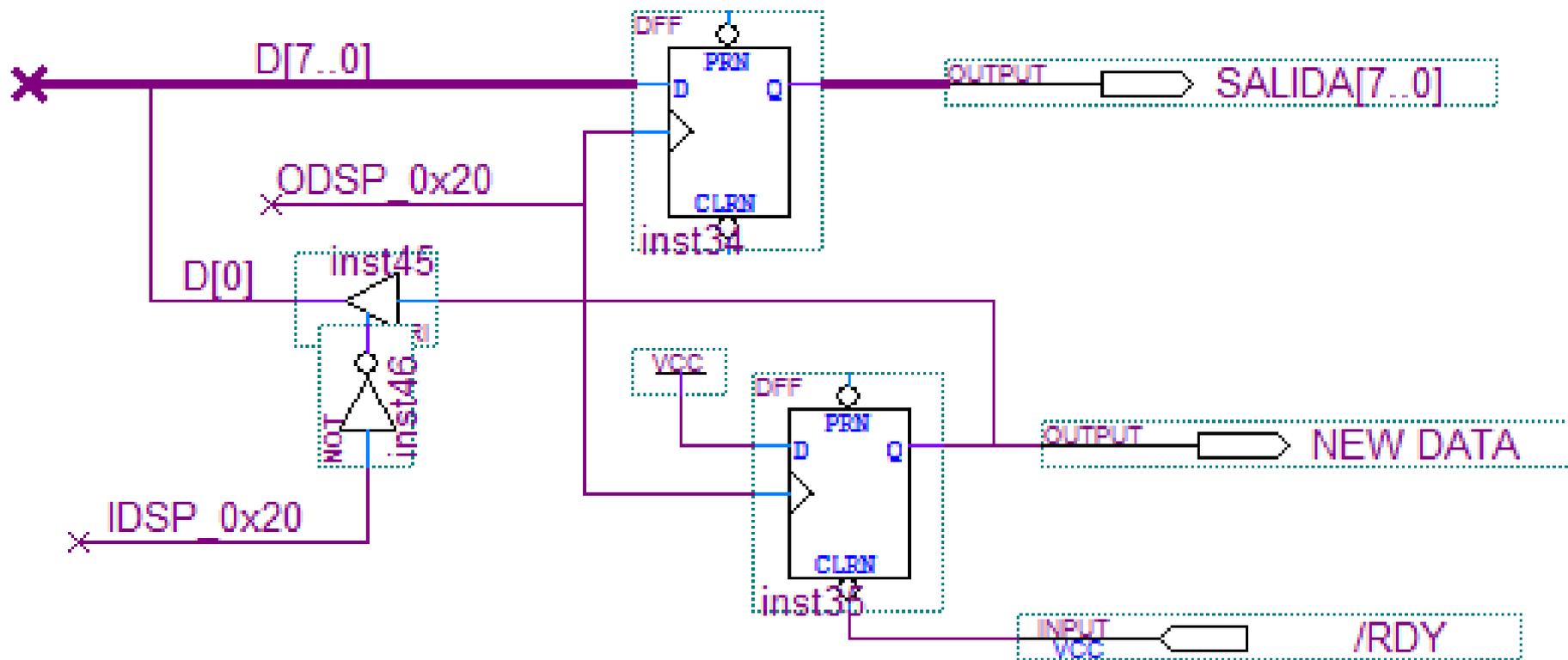
HANDSHAKING



- Registro de datos
 - El productor escribe, el consumidor lee
- Bandera de estado (es un FF)
 - Productor la prende cuando escribe nuevo dato
 - Consumidor la borra cuando lee
- Sincronización
 - Productor consulta la bandera para saber si dato anterior fue consumido
 - Consumidor consulta bandera para saber si hay un nuevo dato para consumir

HANDSHAKING

- Caso **puerto salida**
 - Puerto salida para datos
 - **Puerto entrada** para bandera
 - Borrado bandera con pulso puerto datos

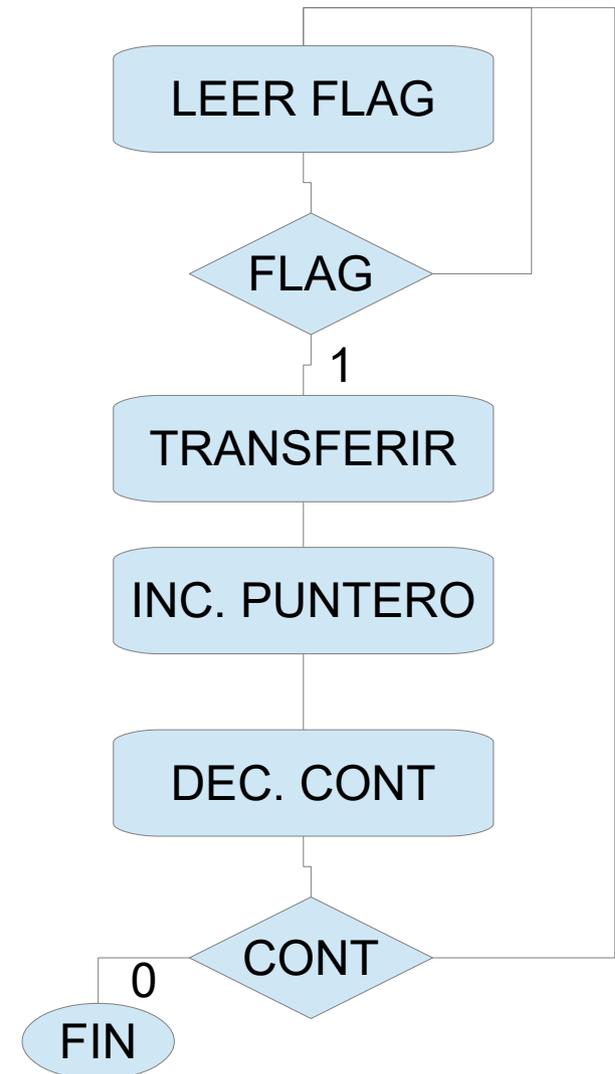


HANDSHAKING

ejemplo

- Subrutina
 - Transfiere N bytes a puerto salida en dirección 0x00
 - Handshake. Flag en bit menos significativo de puerto 0x00 de entrada
 - Parámetros
 - B: cant. bytes N
 - HL: direcc. memoria datos.
- Seudocódigo:

```
Para cont=N hasta 1 hacer
  Mientras flag=0 hacer nada
  Transferir
  Incrementar puntero a memoria
Fin_para
```

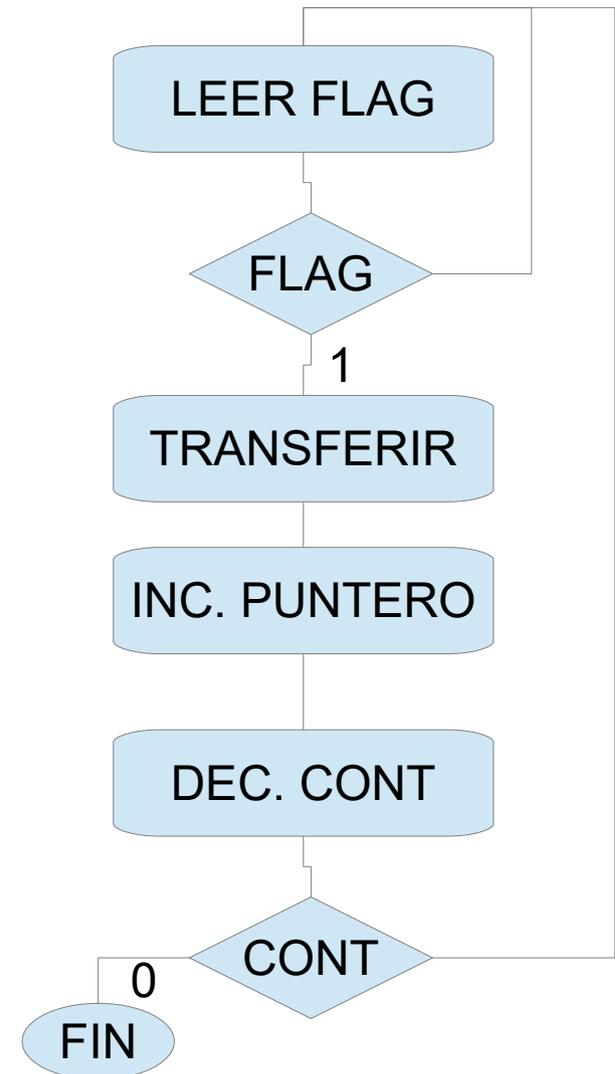


HANDSHAKING

ejemplo

```
; SUBROUTINA escribo_bytes
; escribe n-bytes de memoria a un
puerto
; ASUME: B: n-bytes
; HL: dir base en memoria
; DESTRUYE: A, B, HL
; DEVUELVE: nada
```

```
escribo_bytes:
    in A, (00H)
    bit 0, A
    jr nz, escribo_bytes
    ld a, (hl)
    out (00H), a
    inc hl
    djnz escribo_bytes
    ret
```



HANDSHAKING

Timeout

- ¿Qué pasa si el periférico no consume el dato?
 - Queda trancado en loop infinito
- Solución: límite de tiempo (timeout)
 - Implementación
 - Limitar iteraciones esperando flag.
 - Temporizadores hardware

POLLING

- ¿Cómo atender a varios dispositivos?
 - Ciclo consultando por turno bandera de estado de cada uno.
 - Si es necesario se llama a subrutina que lo atiende y al retornar se sigue el ciclo.
 - A este proceso se le llama **POLLING**.

otrociclo:

```
IN A, (banderas)
```

```
RRA
```

```
JR NC, prueba1
```

```
PUSH AF
```

```
CALL (atiendo0)
```

```
POP AF
```

prueba1:

```
RRA
```

```
JR NC, prueba2
```

```
PUSH AF
```

```
CALL (atiendo1)
```

```
POP AF
```

Prueba2:

```
RRA
```

```
...
```

```
...
```

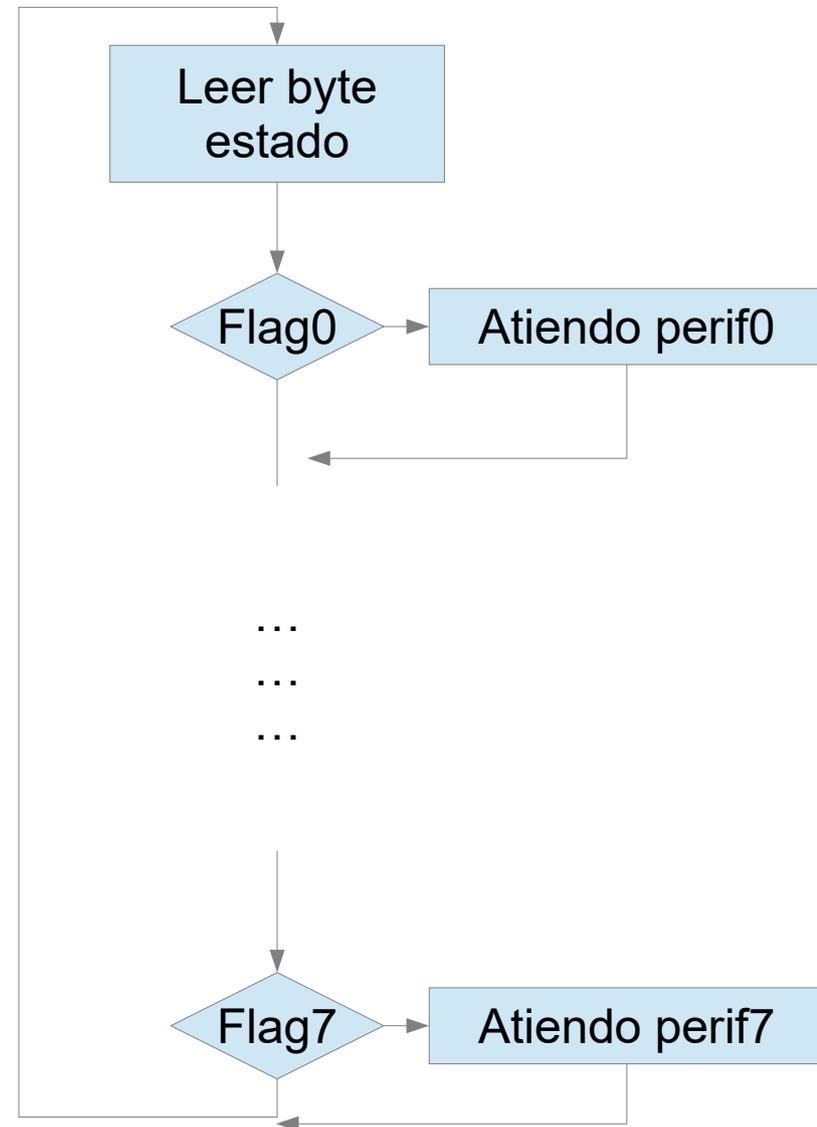
prueba7:

```
RRA
```

```
JR NC, fin
```

```
CALL (atiendo7)
```

fin: JP otrociclo



POLLING

- Tiempo de respuesta
 - Desde que el periférico prende la bandera hasta que es atendido.
 - Cotas en el caso del ejemplo
 - Para periférico 0
 - Para periférico 7
- Prioridades
- Sobrecarga (“overhead”)

otrociclo:

IN A, (banderas)

BIT 0, A

JR NC, prueba1

PUSH AF

CALL (atiendo0)

POP AF

prueba1:

IN A, (banderas)

BIT 1, A

JR NC, prueba2

PUSH AF

CALL (atiendo1)

POP AF

Prueba2:

...

...

prueba7:

IN A, (banderas)

BIT 7, A

JR NC, fin

CALL (atiendo7)

fin: JP otrociclo

