



UNIVERSIDAD
DE LA REPUBLICA
URUGUAY

16. – Periféricos

Introducción a los microprocesadores
2015

Periféricos

- Muchas veces los fabricantes de μ Ps también ofrecen periféricos con aplicaciones establecidas que facilitan el hardware del sistema.
- En general son programables, lo que les permite diferentes modos de funcionamiento.
- En general la interfaz hacia el microprocesador incluye:
 - Bus de datos
 - Bus de direcciones
 - Pocos bits, los suficientes para direccionar registros internos (ej: A0 y A1)
 - Bus de control
 - $/\text{CE}$, para decodificarlo. En general se mapea en E/S
 - Para programar o enviar comandos ($/\text{IORQ}$, $/\text{WR}$)
 - Para leer estado ($/\text{IORQ}$, $/\text{RD}$)
 - Para interrupciones ($/\text{INT}$, $/\text{M1}$, $/\text{IORQ}$, IEI , IEO)
 - Borrar banderas ($/\text{IORQ}$, $/\text{RD}$ o $/\text{WR}$)
 - Inicialización ($/\text{RESET}$)

De lo que disponga cada periférico dependerá de su funcionalidad y como fue diseñado.

Periféricos

- Ejemplos:

- Controlador de Interrupciones
- Temporizador
- Contador
- Controlador de acceso directo a memoria
- Puerto paralelo
- Puerto serial
- Salida PWM
- Controlador USB
- Conversor A/D
- Conversor D/A
-

Periféricos – Z80

- Algunos de los periféricos para Z80:
 - PIO (Parallel Input Output).
 - Dos puertos de 8 bits con su lógica de handshake, que pueden configurarse en varios modos (entrada, salida, bidireccional)
 - SIO (Serial Input Output).
 - Implementa un puerto serial.
 - CTC (Counter Timer Clock).
 - Cuatro contadores de 8 bits programables que pueden utilizarse como temporizadores o como contadores de eventos externos.
 - DMA Controller.
 - Controlador de acceso directo a memoria.

Periféricos – T80

- Periféricos para T80:
 - Controlador de Interrupciones
 - Controlador que implementa el protocolo Daisy Chain.
 - Counter
 - Cuenta flancos en una entrada.
 - Timer
 - Cuenta períodos de reloj o múltiplos de períodos de reloj.

En el curso se van a estudiar únicamente estos 3 periféricos

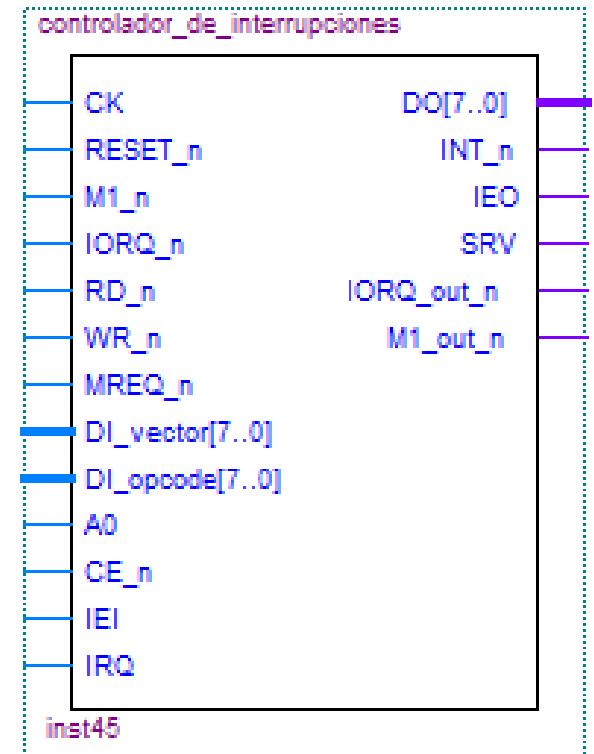
Controlador de Interrupciones

- Permite gestionar ante el T80 el requerimiento de una interrupción (por su entrada IRQ).
- Utilizando varios controladores permite utilizar prioridades con el protocolo Daisy Chain.
- Se pueden realizar por software las siguientes operaciones:
 - Escritura/lectura del vector de interrupciones.
 - Lectura del estado del controlador codificado en un byte.
 - Borrado de peticiones pendientes (mediante escritura a una dirección de E/S).

Controlador de Interrupciones

❖ PINES:

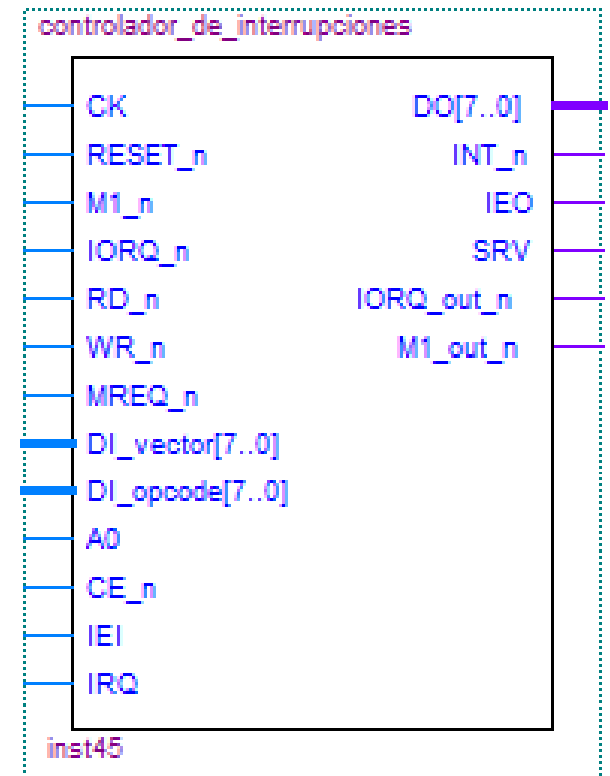
- Generales
 - CK, RESET_n.
- Daisy chain y solicitud al procesador
 - IEI, IEO, INT_n
- Bus de control
 - M1_n, IORQ_n, RD_n, WR_n, MREQ_n
- Bus de datos
 - DI_vector:
 - Escritura del vector de interrupciones.
 - Se conecta a DO del Z80
 - DO.
 - Lectura de vector de interrupciones y estado.
 - Ciclo INTA, vector de interrupciones.
 - Se conecta al OR que entra al DI del Z80.
 - DI_opcode:
 - Para husmear el bus de datos y esperar RETI.



Controlador de Interrupciones

❖ PINES:

- Bus de direcciones
 - A0
 - 2 puertos de entrada y 2 de salida.
- Decodificación
 - CE_n
- Conexión al periférico
 - IRQ:
 - Flanco creciente activa petición
 - SRV:
 - Igual a 1 mientras solicitud “en servicio”
 - IORQ_{out_n}, M1_{out_n}:
 - Se activan en ciclo INTA
 - Se utilizan para conectar a las entradas del mismo nombre de un dispositivo modo 1.



Controlador de Interrupciones

ESPACIO E/S

A0	WR	RD
0	VECTOR INT.	VECTOR INT.
1	BORRA PETICIONES*	ESTADO

* no importa que se escriba

PALABRA DE ESTADO

BITS[1:0]	ESTADO
00	RESERVADO
01	SIN SOLICITUDES PENDIENTES
10	CON SOLICITUD PENDIENTE
11	EN SERVICIO

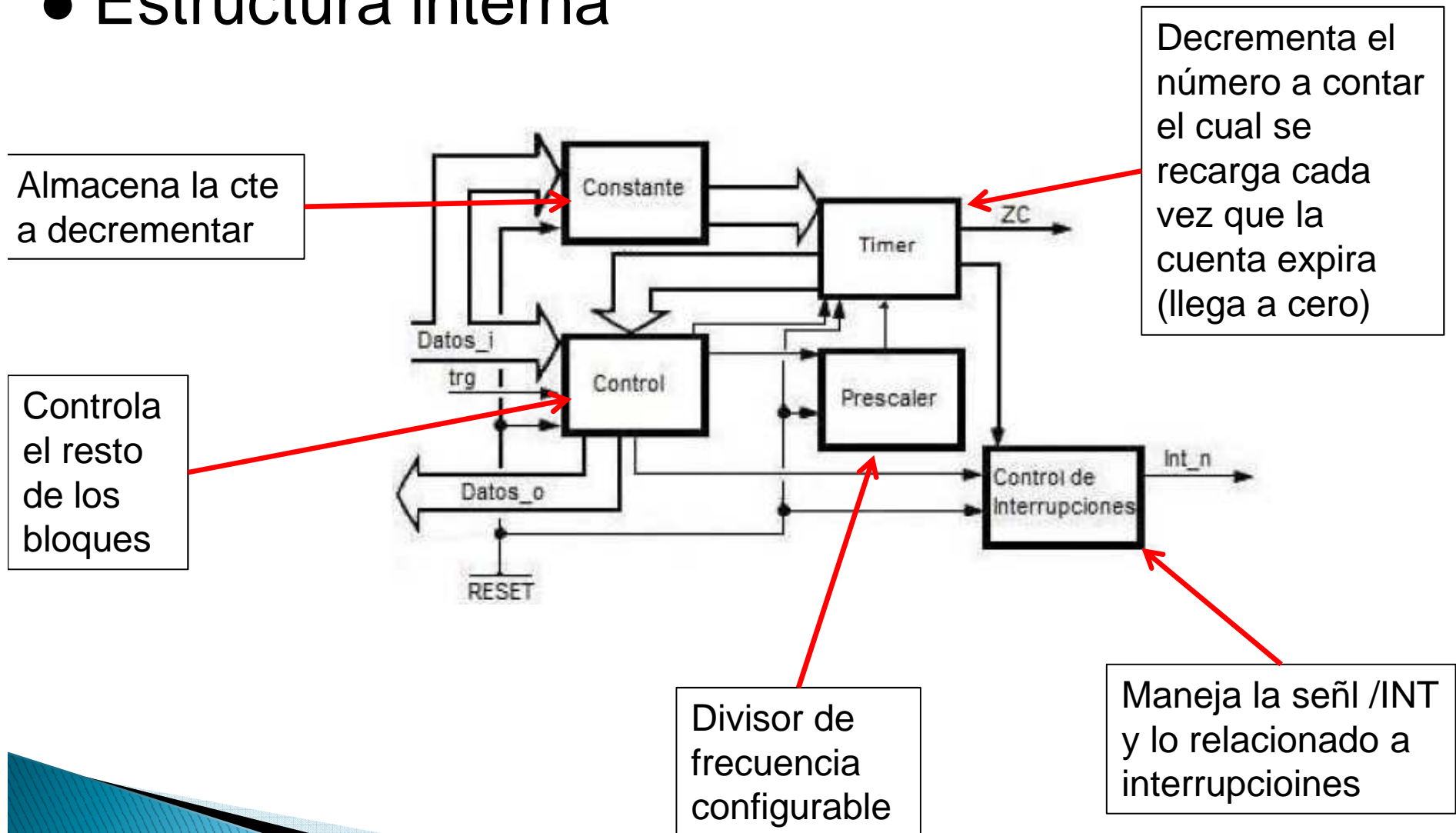
- El valor de los bits 7 a 2 de la palabra de estado es siempre 0 (cero).

Timer

- Mide intervalos de tiempo contando períodos de reloj o múltiplos de él (configurable).
- Realiza una cuenta regresiva de un valor configurable.
- Cada vez que la cuenta llega a 0 (cero) da un pulso en su salida Z_c y puede generar una interrupción (si está programado para ello).
- La medida puede iniciarse en forma automática o por una señal externa.
- Se puede consultar mediante:
 - Polling
 - Interrupciones modo 1
 - Interrupciones modo 2 (utilizando un controlador de interrupciones).

Timer

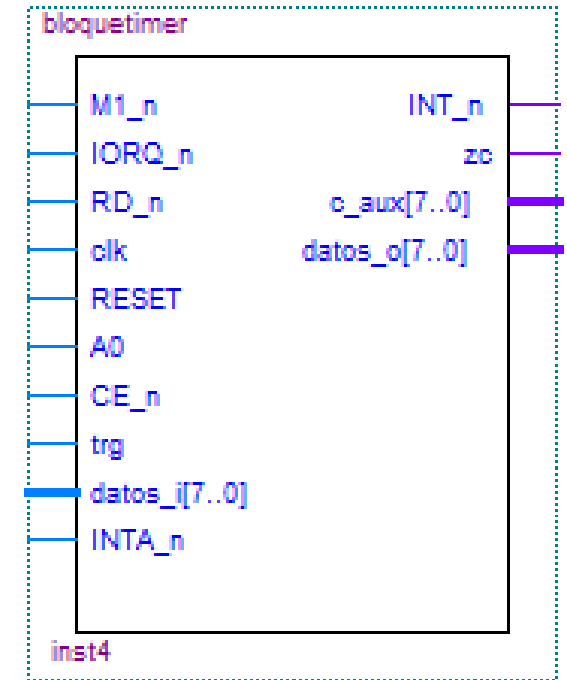
- Estructura interna



Timer

❖ PINES:

- Generales
 - CK, RESET_n.
- Bus de control
 - M1_n, IORQ_n, RD_n
 - No posee /WR, se deduce de las demás.
- Bus de datos
 - Datos_i:
 - Se conecta a DO del Z80
 - Datos_o.
 - Se conecta al OR que entra al DI del Z80 habilitándose en forma adecuada.
- Bus de direcciones
 - A0
 - 2 puertos de entrada y 2 de salida.
- Decodificación
 - CE_n



Timer

❖ PINES:

● Sincronización

- trg:
 - Se puede configurar para que arranque a contar con un flanco y éste se puede además configurar creciente o decreciente.
- Zc
 - Da un pulso a 1 cuando expira la cuenta
- C_aux
 - Muestra el valor interno de la cuenta.

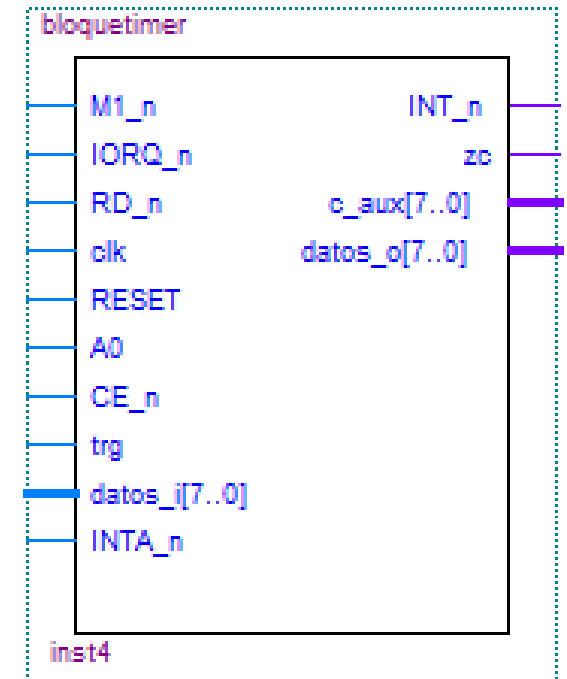
● Interrupciones

MODO 1:

- INT_n:
 - Conectado a INT_n del Z80
- INTA_n
 - Conectado a ODSP o (IORQ + M1_n)

MODO 2:

- INT_n o Zc
 - Conectado al controlador de Interrupciones.



Timer

ESPACIO E/S

A0	WR	RD
0	CONSTANTE	FLAG ZC*
1	CONTROL	CUENTA

Se borra flag al leerla.
Se utiliza en bit D0

PALABRA CONTROL

BITS	
7	INT (1=habilitada)
6	TRG_CONF (1=subida)
5	SW_RESET (1=reinicia)
4	TRG_START(1=espera f anco)
3:0	PRESCALER

SW_RESET: Fuerza la inicialización del Timer

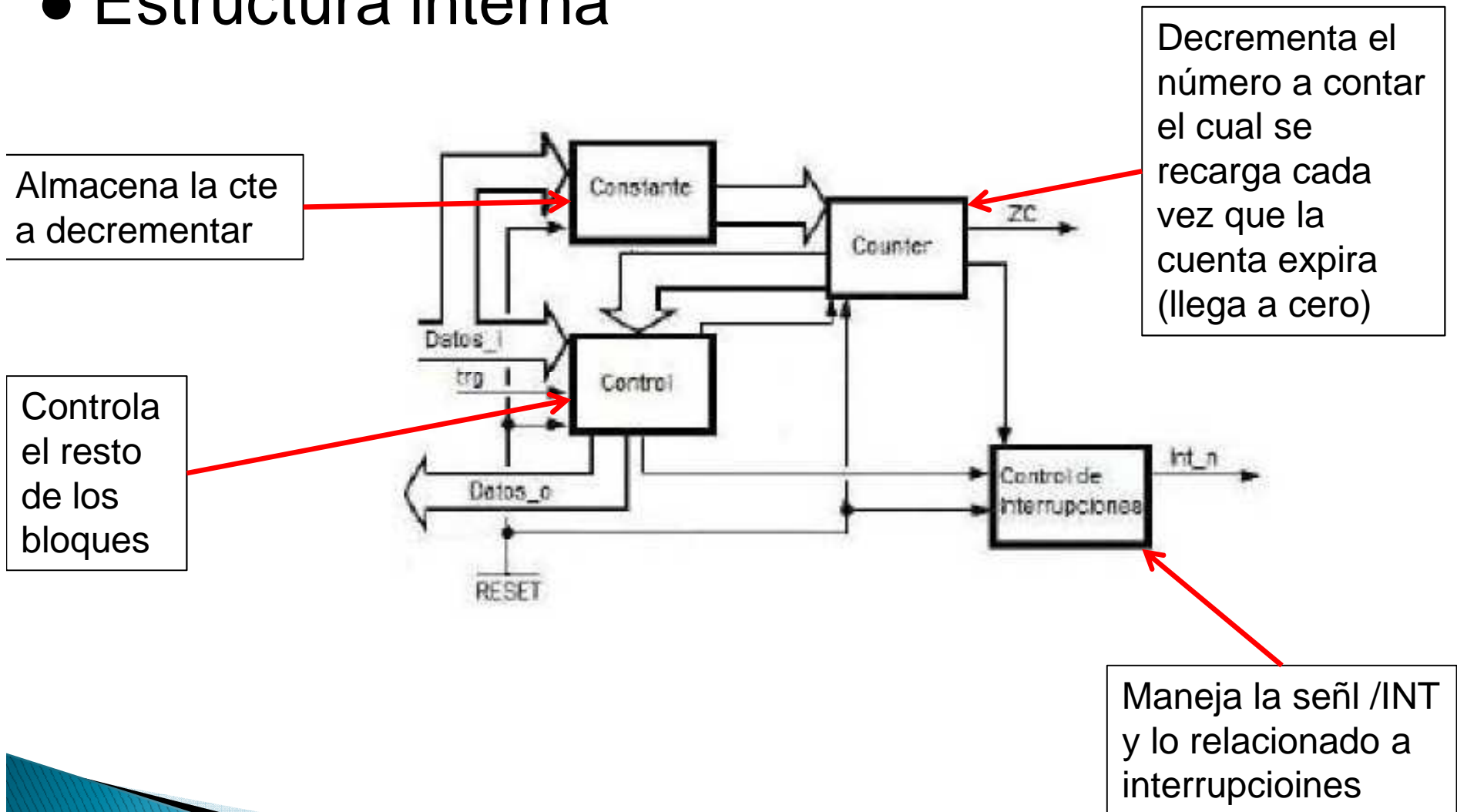
PRESCALAR: Divide el CK por $2^{\text{PRESCALAR}}$ (el valor 0000 lo toma como 2^{16})

Counter

- Cuenta flancos en su entrada **trg**
- Realiza una cuenta regresiva de un valor configurable.
- Cada vez que la cuenta llega a 0 (cero) da un pulso en su salida **Zc** y puede generar una interrupción (si está programado para ello).
- Se puede consultar mediante:
 - Polling
 - Interrupciones modo 1
 - Interrupciones modo 2 (utilizando un controlador de interrupciones).

Counter

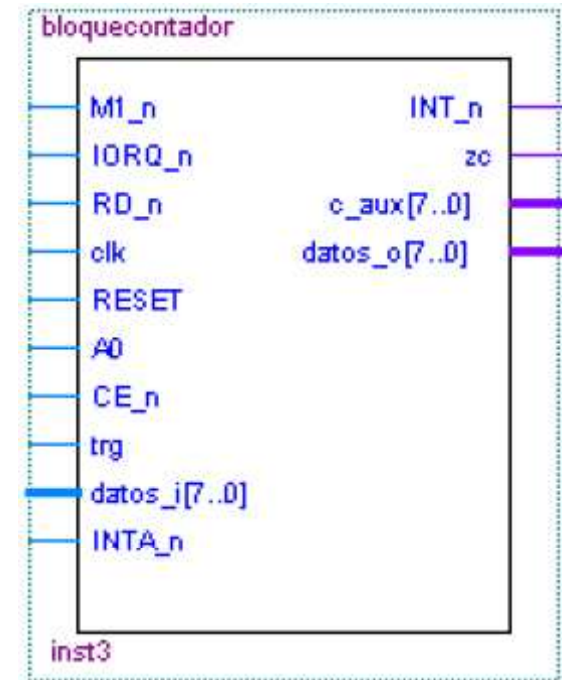
- Estructura interna



Counter

❖ PINES:

- Generales
 - CK, RESET_n.
- Bus de control
 - M1_n, IORQ_n, RD_n
 - No posee /WR, se deduce de las demás.
- Bus de datos
 - Datos_i:
 - Se conecta a DO del Z80
 - Datos_o..
 - Se conecta al OR que entra al DI del Z80 habilitándose en forma adecuada.
- Bus de direcciones
 - A0
 - 2 puertos de entrada y 2 de salida.
- Decodificación
 - CE_n



Counter

❖ PINES:

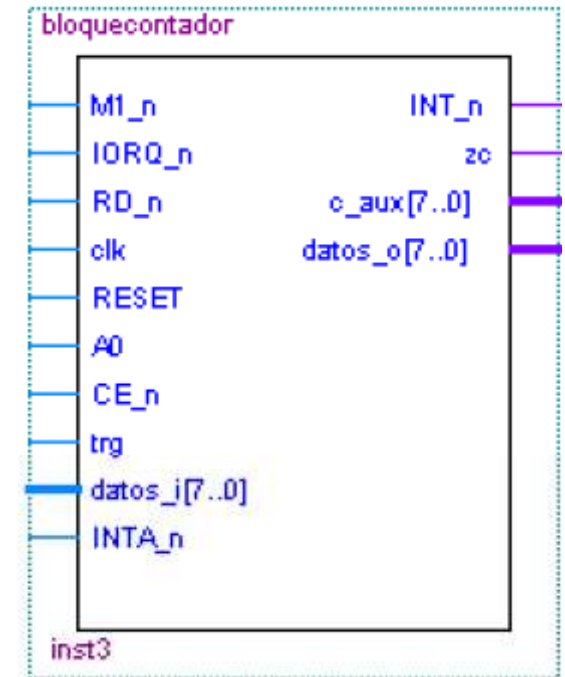
- Sincronización
 - **trg:**
 - **Cuenta flancos en esta señal.**
 - Zc
 - Da un pulso a 1 cuando expira la cuenta
 - C_aux
 - Muestra el valor interno de la cuenta.
- Interrupciones

MODO 1:

 - INT_n:
 - Conectado a INT_n del Z80
 - INTA_n
 - Conectado a ODSP o (IORQ + M1_n)

MODO 2:

 - INT_n o Zc
 - Conectado al controlador de Interrupciones.



Counter

ESPACIO E/S

A0	WR	RD
0	CONSTANTE	FLAG ZC*
1	CONTROL	CUENTA

Se borra flag al leerla.
Se utiliza en bit D0

PALABRA CONTROL

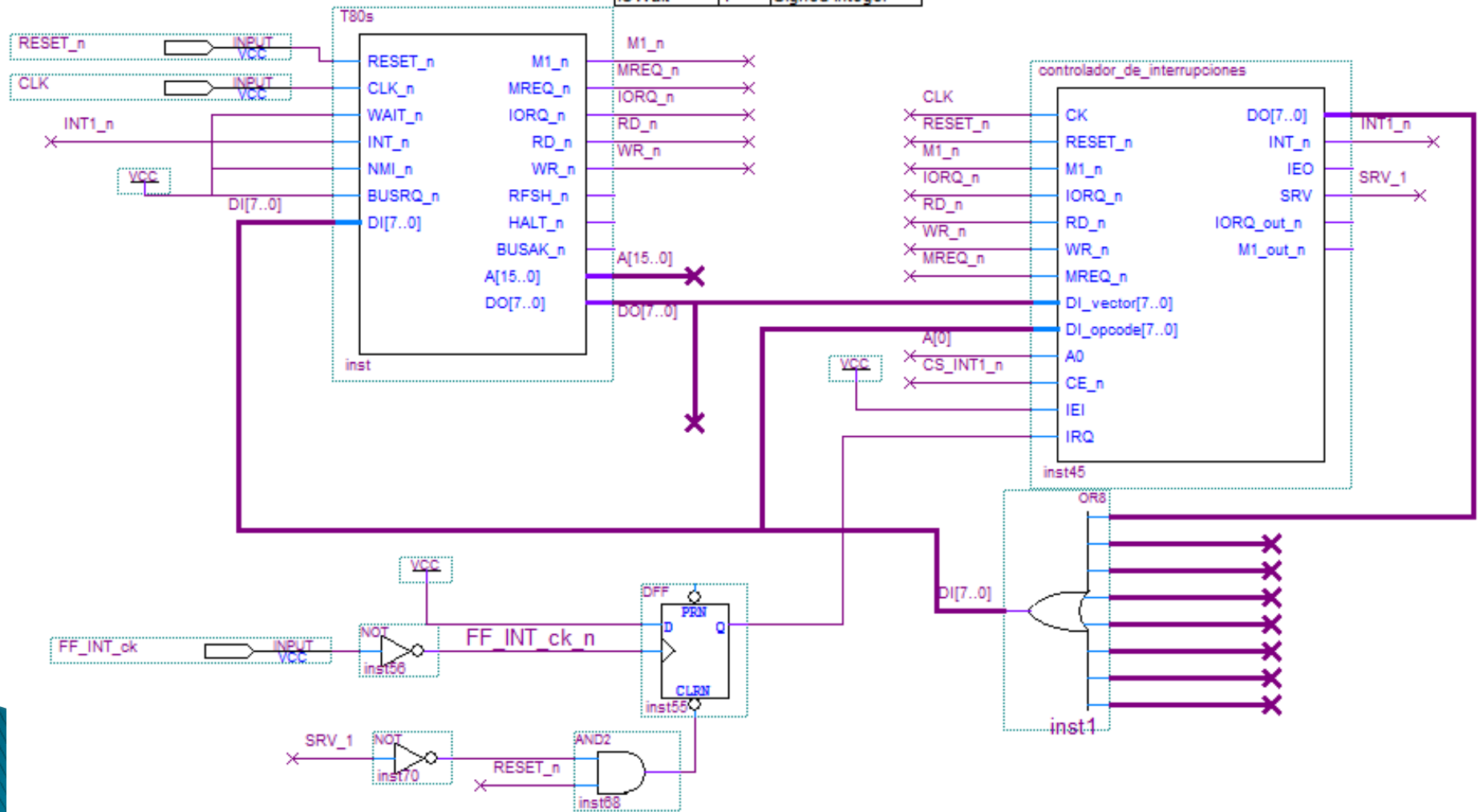
BITS	
7	INT (1=habilitada)
6	TRG_CONF (1=subida)
5	SW_RESET (1=reinicia)
4	X
3:0	X X X X

SW_RESET: Fuerza la inicialización del Timer

En este caso los bits 4 al 0 de la palabra de control son Don't care

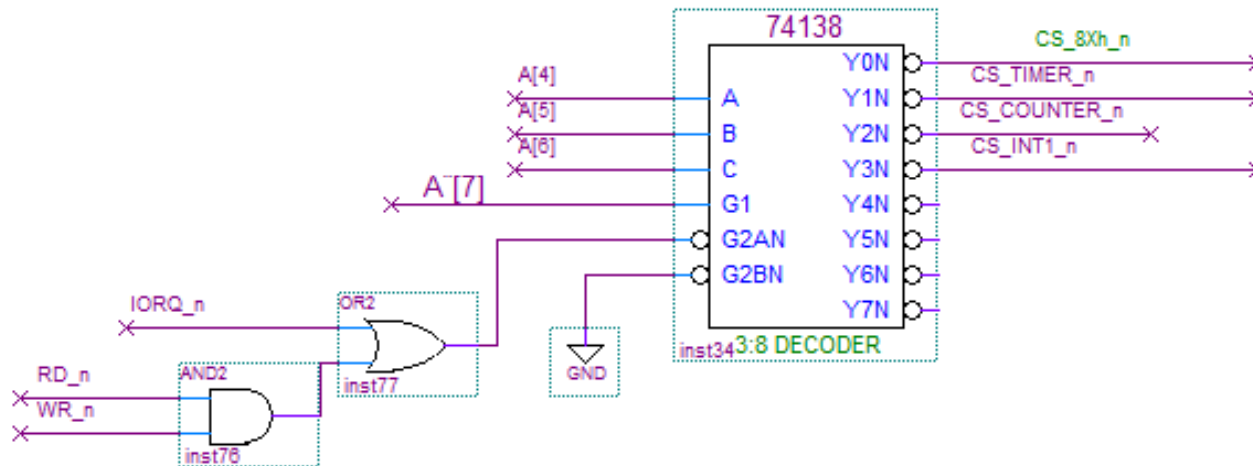
Ejemplos

Parameter	Value	Type
Mode	0	Signed Integer
T2Write	0	Signed Integer
IOWait	1	Signed Integer



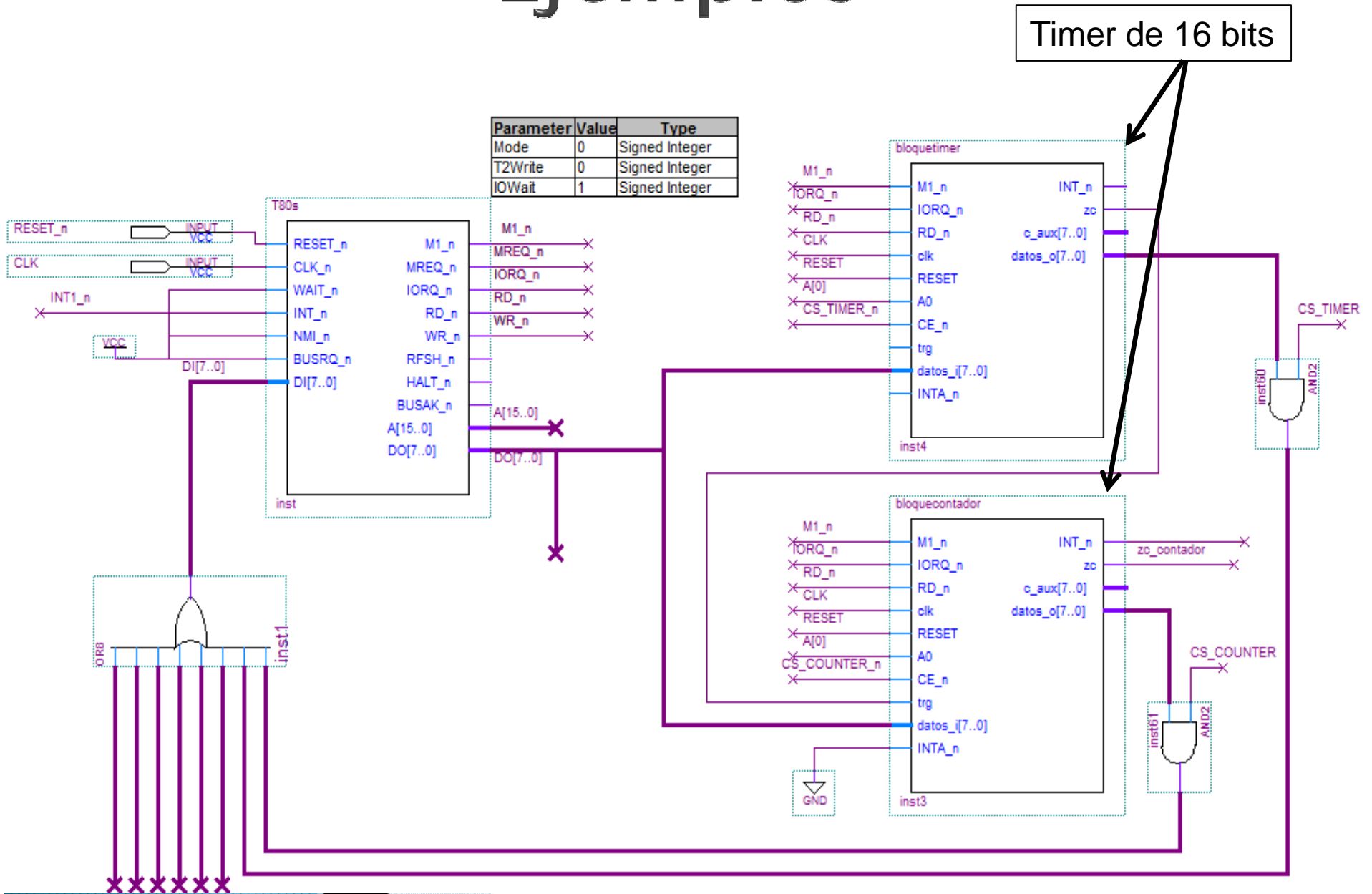
Ejemplos

- Único controlador (IEI = VCC y IEO no conectada).
- El FF evita peticiones de interrupción hasta tanto no finalice la rutina de atención. Observar que el FF se borra con SRV.
- Con la siguiente decodificación:



- ¿En que dirección se configura el vector de interrupciones?
- ¿En que dirección se lee el valor del vector de interrupciones?
- ¿En que dirección se borra el FF de petición de interrupciones?
- ¿En que dirección se lee el estado del controlador de interrupciones?

Ejemplos



Parameter	Value	Type
Mode	0	Signed Integer
T2Write	0	Signed Integer
IOWait	1	Signed Integer

Timer de 16 bits