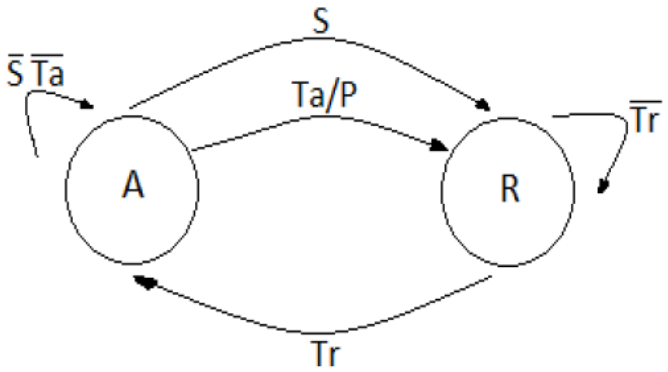
EXAMEN FEB2011 (usando T80)

Marcapasos sencillo



Estados:

R: Resposo

A: Alerta

Entradas

S: Sensa el latido

Salidas:

P: Estimula el corazón

Otras Entradas:

Port_Ta

Port_Tb

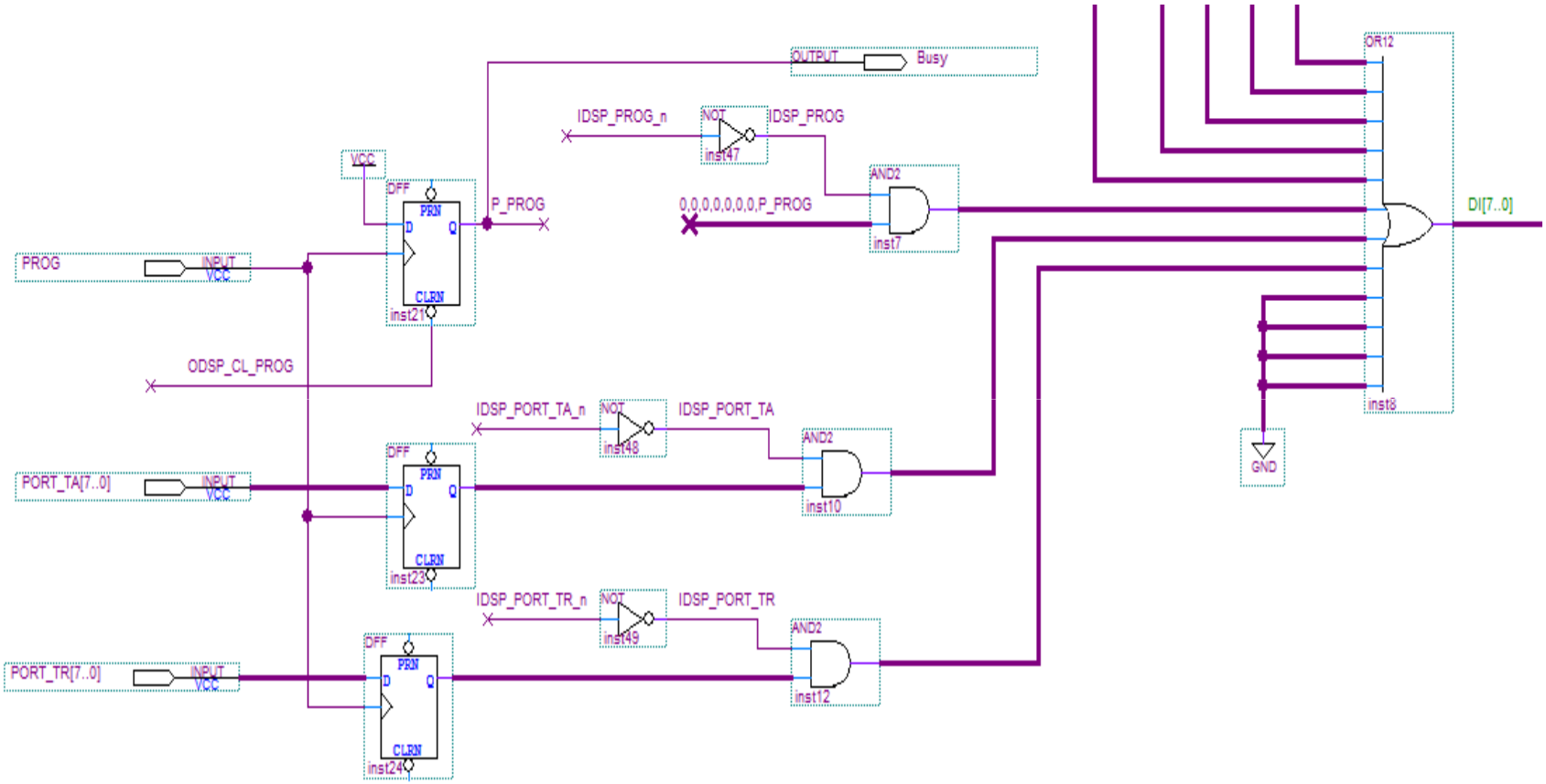
Prog

Otras Salidas:

Busy

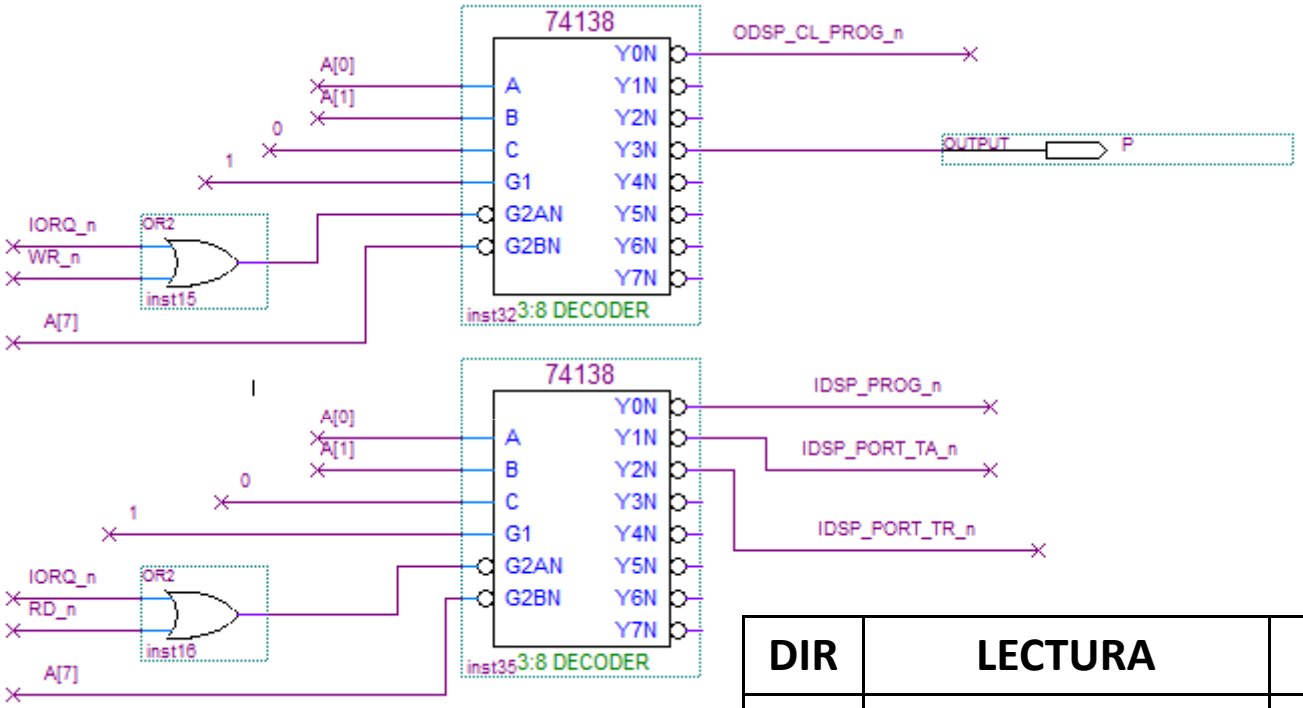
EXAMEN FEB2011 (usando T80)

Puertos



EXAMEN FEB2011 (usando T80)

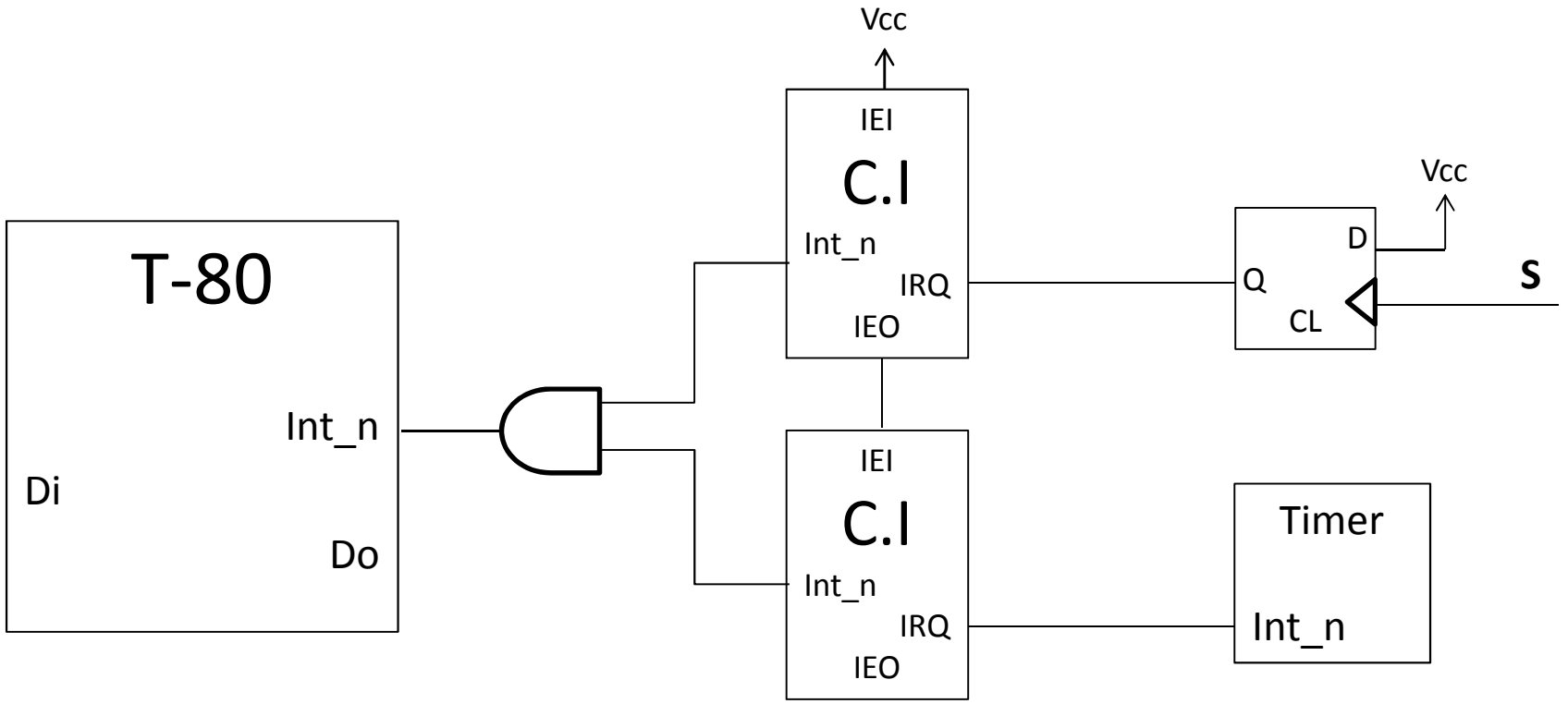
Decodificación de Puertos



DIR	LECTURA	ESCRITURA
00	PROG	CL_PROG
01	PORT_TA	
02	PORT_TR	
03		P

EXAMEN FEB2011 (usando T80)

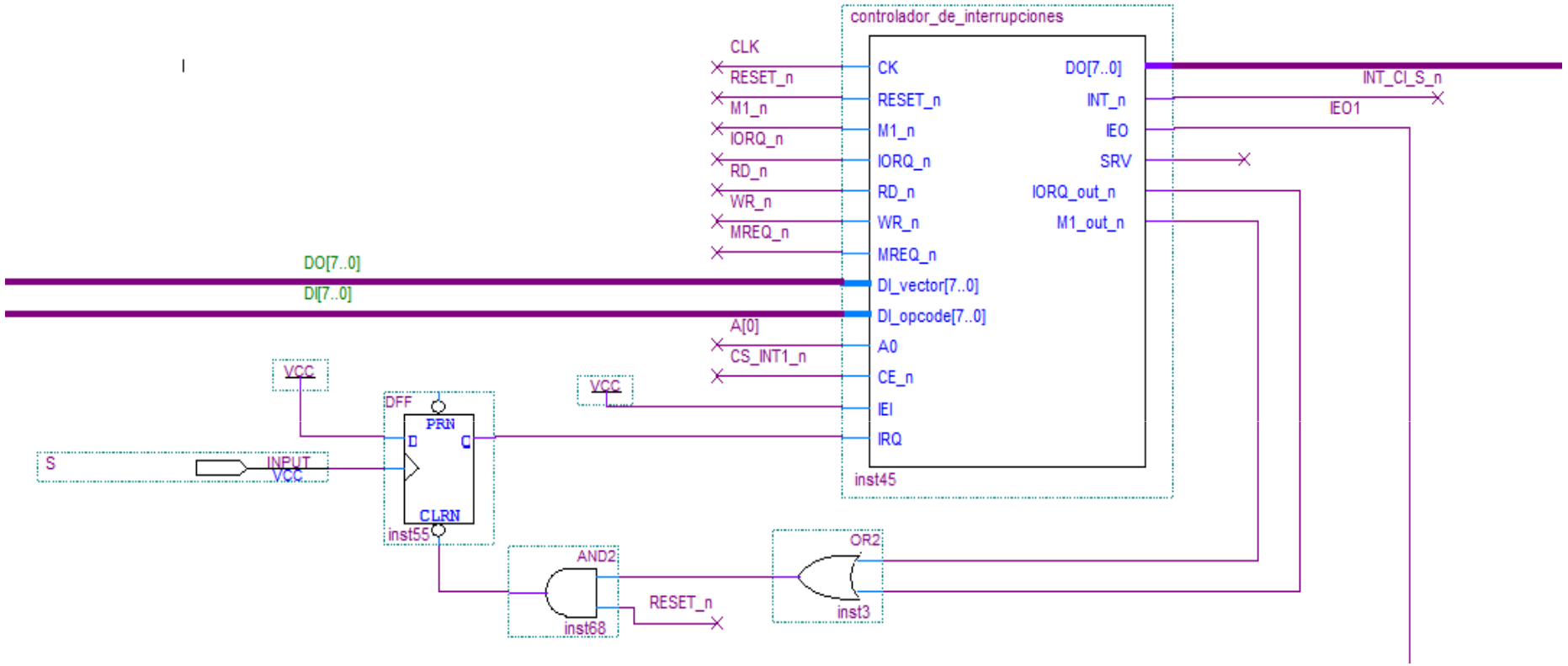
Controladores de Interrupciones y Timer



- Interrupción de **S** tiene mayor prioridad que la interrupción del Timer

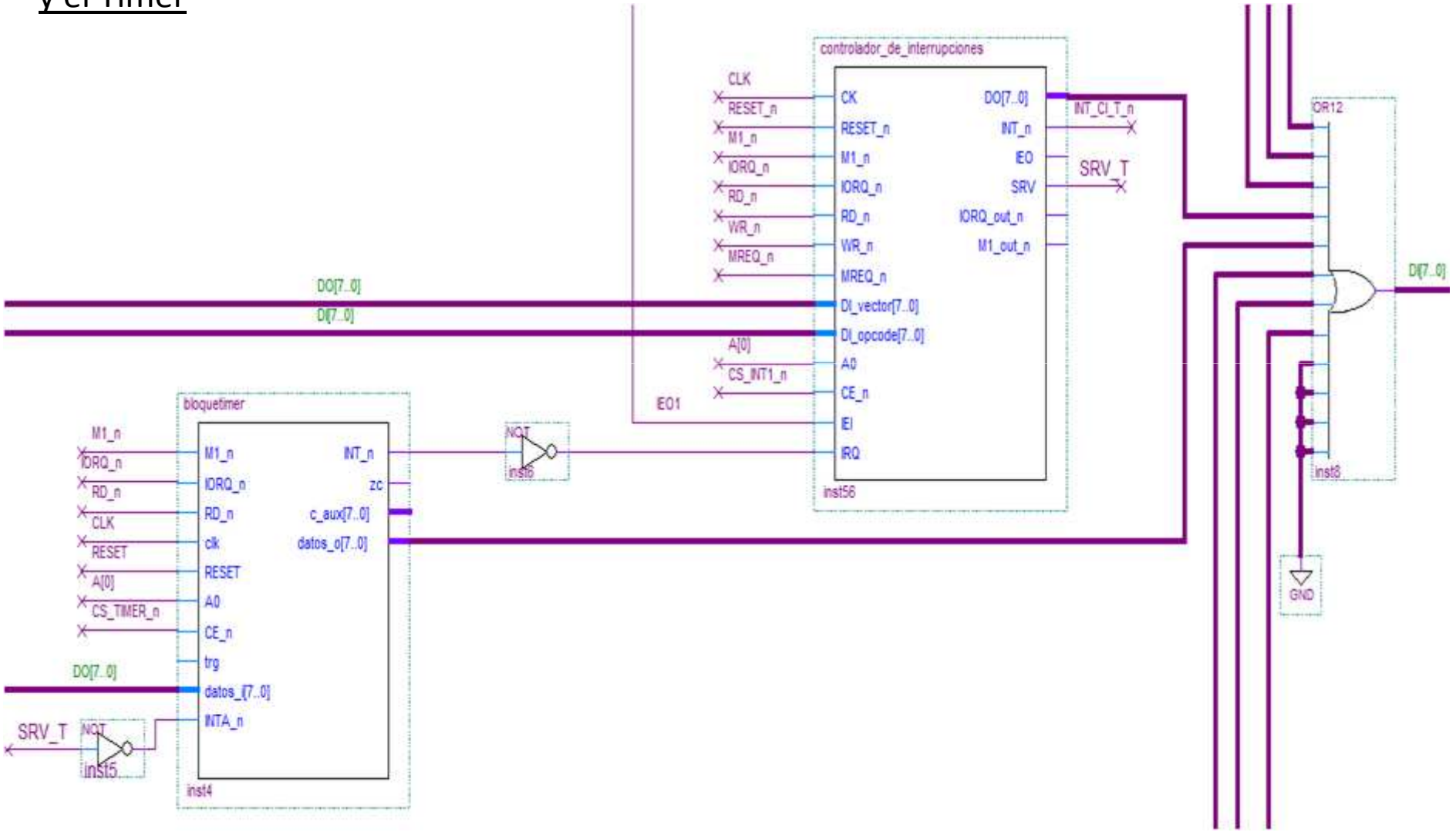
EXAMEN FEB2011 (usando T80)

Controlador de Interrupciones de **S**



EXAMEN FEB2011 (usando T80)

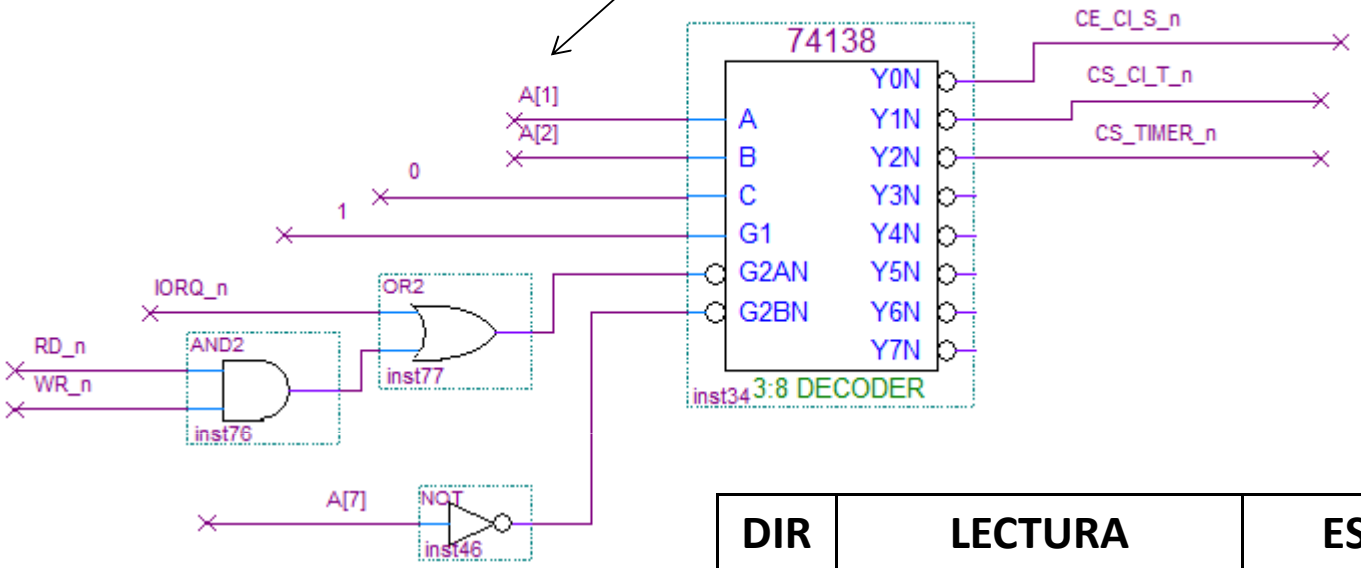
Controlador de Interrupciones y el Timer



EXAMEN FEB2011 (usando T80)

Observar que no tiene A0

Decodificación de Cis y Timer

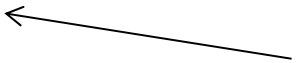


DIR	LECTURA	ESCRITURA
80	CI_S_VECTOR	CI_S_VECTOR
81	CI_S_CL_PETICIÓN	CI_S_ESTADO
82	CI_T_VECTOR	CI_T_VECTOR
83	CI_T_CL_PETICIÓN	CI_T_ESTADO
84	TMR_CTE	TMR_FLG_ZC
85	TMR_CTRL	TMR_CUENTA

EXAMEN FEB2011 (usando T80)

```
PROG equ 00
CLR_PROG equ 00
PORT_TA equ 01
PORT_TR equ 02
P equ 03

CI_S_VI equ 80
CI_S_CL_PETICIÓN equ 81

CI_T_VI equ 82
CI_T_CL_PETICIÓN equ 83

TMR_CTE equ 84
TMR_CTRL equ 85

VAL_TA_INI equ 70
VAL_TR_INI equ 30
```

DIR	LECTURA	ESCRITURA
00	PROG	CL_PROG
01	PORT_TA	
02	PORT_TR	
03		P
80	CI_S_VECTOR	CI_S_VECTOR
81	CI_S_CL_PETICIÓN	CI_S_ESTADO
82	CI_T_VECTOR	CI_T_VECTOR
83	CI_T_CL_PETICIÓN	CI_T_ESTADO
84	TMR_CTE	TMR_FLG_ZC
85	TMR_CTRL	TMR_CUENTA



Lo pide la letra (valor inicial de Ta y Tr)

EXAMEN FEB2011 (usando T80)

b)

```
Programa PRINCIPAL
        org 100h
ppal:   in A,(PROG)          ; leo PROG
        BIT 0,A              ; Si es 0 sigo leyendo
        jp Z, ppal           ; sino, cargo tiempos
        in A, (PORT_TA)
        ld (VAL_TA), A
        in A, (PORT_TR)
        ld (VAL_TR),A
        out (CLR_PROG),A
        JP ppal
```

Definidas con los EQU

```
        org 8000h
VAL_TA: db
VAL_TR: db
ESTADO: db
```

Se definen las variables

EXAMEN FEB2011 (usando T80)

```
c) org 500h
rutint_s:
        push AF
        ld A, E_REPOSO
        cp (ESTADO)             ; me fijo si estoy en reposo
        jp Z, retorno_rutint_s
        ld (ESTADO), A          ; paso a estado reposo
        ld A, (VAL_TR)
        out (TMR_CTE), A        ; Cambio cte tiempo a VAL_TR
        ld A, T_CW
        out (TMR_CTRL), A       ; empiezo a contar TR
        out (CI_T_CL_PETICIÓN),A        ;Borro petición CI_T
                                        ;por si interrumpió mientras
                                        ;atendía a CI_S
retorno_rutint_s:
        pop AF
        ei              ; es indistinto donde se habilitan las
                        ; interrupciones
        reti


E_REPOSO equ 00 ; representa reposo en variable ESTADO
E_ALERTA equ 01 ; representa alerta en variable ESTADO

T_CW equ 1X101001 ; int habil, sw reset, trg c/cte,
                        prescaler=9  (2^9 = 512)
```

EXAMEN FEB2011 (usando T80)

Si está en ALERTA e interrumpe CI_S aqui, y se ejecuta rutint_s:, se cambia a REPOSO, Luego al retornar a rutint_t: se vuelve a cambia a ALERTA en forma erronea. Por eso “ei” debe ir luego de leer (ESTADO), puede ser al final.

```
rutint_t:
        push AF
        ld A, E_REPOSO
        cp (ESTADO)             ; me fijo si estoy en reposo
        ei
        jp Z, Estoy_en_reposo
Estoy_en_alerta:
        ld (ESTADO), A          ; paso a estado reposo
        ld A, (VAL_TR)
        out (TMR_CTE), A        ; Cambio cte tiempo a VAL_TR
        ld A, T_CW
        out (TMR_CTRL), A       ; empiezo a contar TR
        out (P), A              ; doy pulso P
        jp retorno_rutint_t
Estoy_en_reposo::
        ld A, E_ALERTA
        ld (ESTADO), A          ; paso a estado alerta
        ld A, (VAL_TA)
        out (TMR_CTE), A        ; Cambio cte tiempo a VAL_TA
        ld A, T_CW
        out (TMR_CTRL), A       ; empiezo a contar TA
retorno_rutint_t:
        pop AF
        reti
```

EXAMEN FEB2011 (usando T80)

```
d) org 2000h
tabla_int:
   DW: rutint_s
   DW: rutint_t

    org 0000h
    ld sp, 0000
    Im2
    ld A, tabla_int/256
    ld I,A

    ld A,E_ALERTA
    ld (ESTADO), A
    ld A, VAL_TA_INI
    ld (VAL_TA), A
    ld A, VAL_TR_INI
    ld (VAL_TR), A
```

```
ld A, 00        ; vector 0 para
                ; controlador de S
out (CI_S_VI), A

ld A, 02        ; vector 2 para
                ; controlador de T
out (CI_T_VI), A

ld A, VAL_TA_INI
out (TMR_CTE), A
ld A, T_CW
out (TMR_CTRL), A
                ; empiezo a contar TA

out (CLR_PROG), A
                ; Limpio FF PROG
ei
jp ppal
```