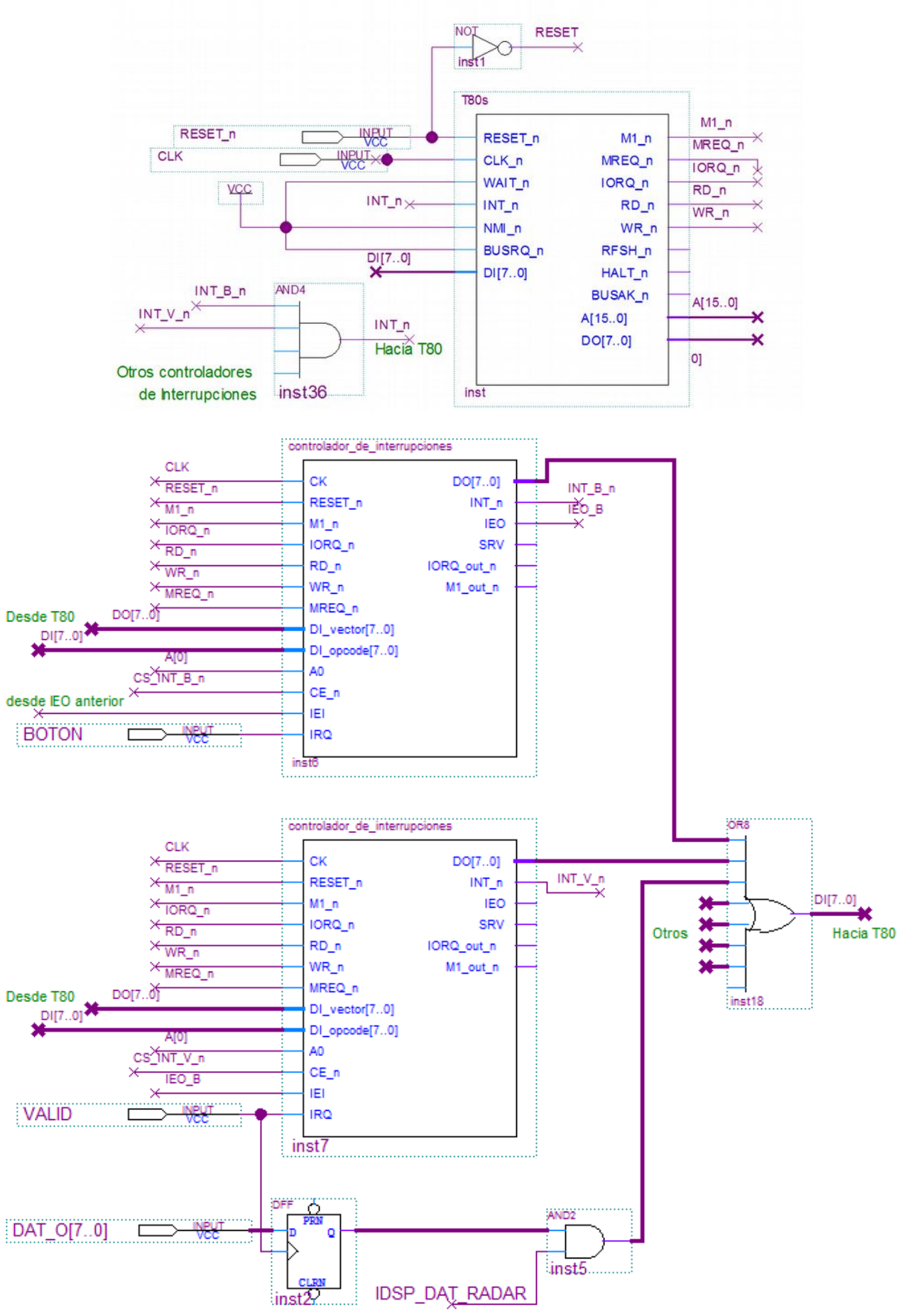
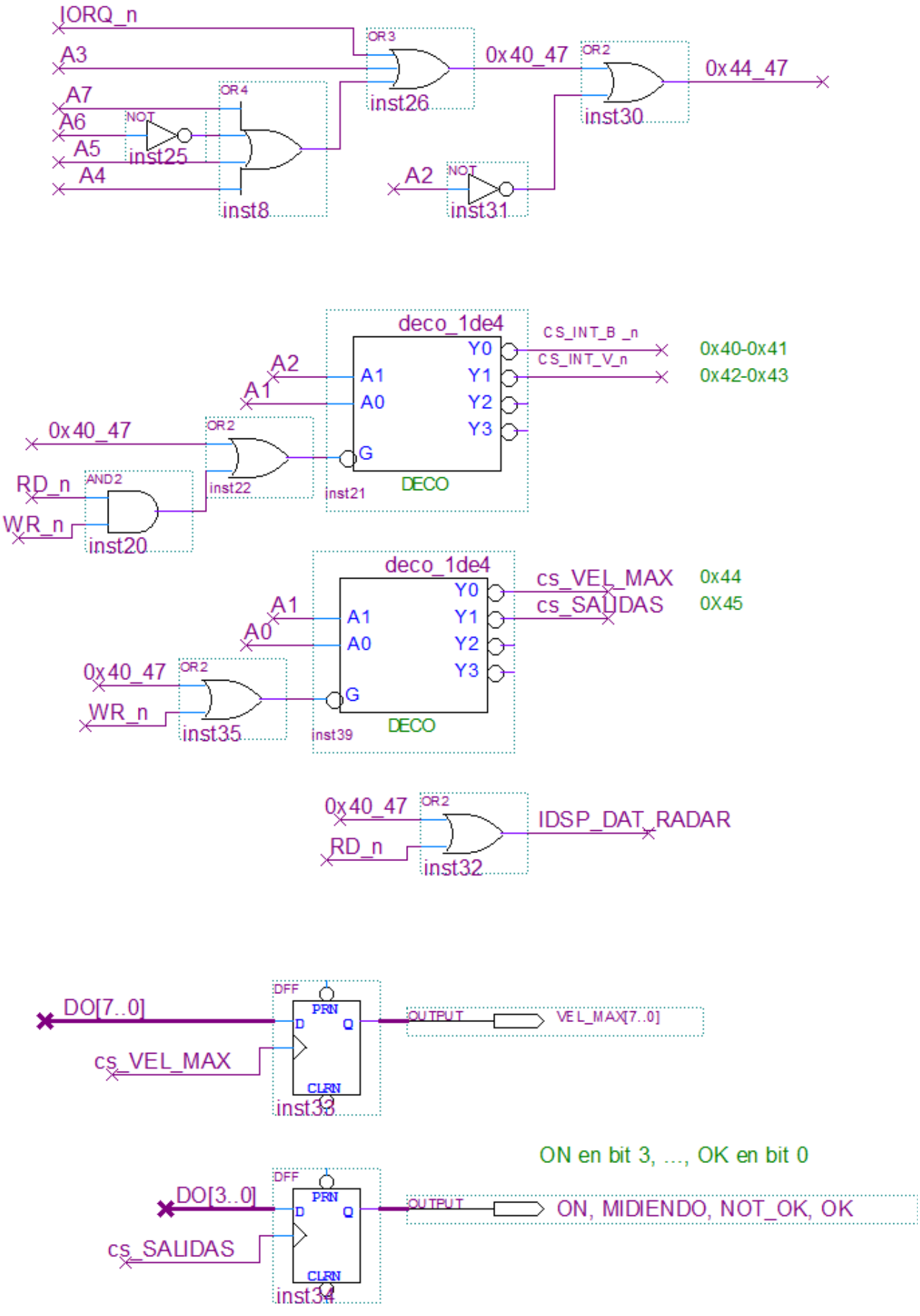
PROBLEMA 1 - a) Hardware



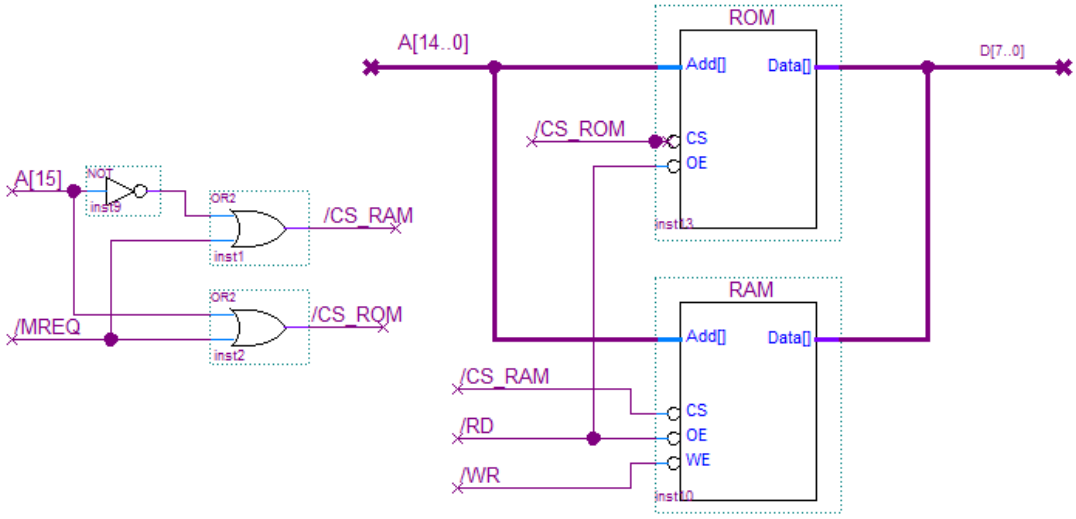

IORQ_n
A3
A7
A6
A5
A4
NOT
inst25
OR4
inst8
OR3
inst26
0x40_47
OR2
inst30
0x44_47
A2
NOT
inst31
deco_1de4
A1
A0
G
Y0
Y1
Y2
Y3
DECO
inst21
CS_INT_B _n
CS_INT_V_n
0x40-0x41
0x42-0x43
0x40_47
OR2
inst22
RD_n
AND2
WR_n
inst20
deco_1de4
cs_VEL_MAX
cs_SALIDAS
0x44
0X45
0x40_47
OR2
WR_n
inst35
inst39
0x40_47
OR2
RD_n
inst32
IDSP_DAT_RADAR
DO[7..0]
cs_VEL_MAX
DFF
PRN
D
Q
CLRN
inst33
OUTPUT
VEL_MAX[7..0]
ON en bit 3, ..., OK en bit 0
DO[3..0]
cs_SALIDAS
DFF
PRN
D
Q
CLRN
inst34
OUTPUT
ON, MIDIENDO, NOT_OK, OK

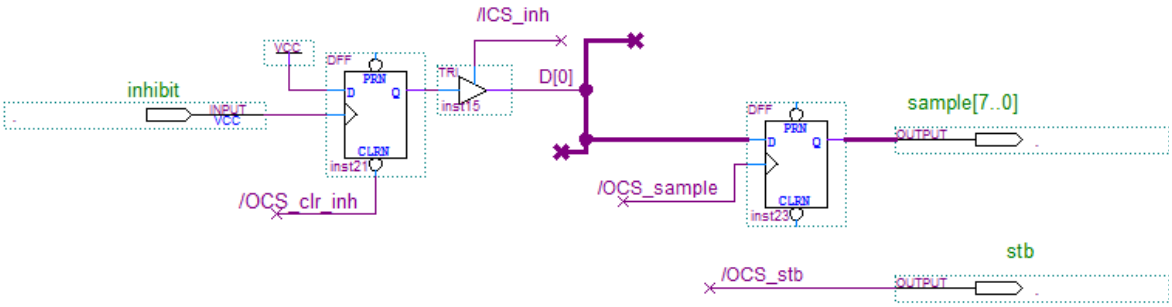
```
VEC_BOTON equ 0x0a
VEC_VALID equ 0x0c

ic_BOTON  equ 0x40
ic_VALID  equ 0x42

vel_max     equ 0x44
salidas     equ 0x45
dat_o_radar equ 0x44

bit_ok       equ 1
bit_not_ok   equ 2
bit_midiendo equ 3
bit_on       equ 4

E_IDLE    equ 0
E_WUMBRAL equ 1
E_WFIN    equ 2

UMBRAL    equ 50

; b) Inicializacion

   org 0
   ld sp, 0
   im 2
     ;; tabla de int en rom
   ld hl, tabla
   ld a, h
   ld i, a
     ;; controladores int
   ld a, VEC_BOTON
   out (ic_BOTON), a
   out (ic_BOTON+1), a
   ld a, VEC_VALID
   out (ic_VALID), a
   out (ic_VALID+1), a
     ;; salidas apagadas
   ld a, 0
   out (salidas), a
     ;; variables
   ld a, E_IDLE
   ld (estado), a
     ;; otras inicializaciones
   call init_otros
   ei
   jp ppal

;; c) isrs
;; c.1) boton
;; if ( estado == idle ) {/* arranco */
;;    prendo radar
;;    actualizo salidas
;;       (midiendo=1, el resto a 0)
;;    max=0
;;    estado = espero_umbral
;; }else{              /* cancelo */
;;    apago radar
;;    actualizo salidas
;;       (not_ok=1, desactivo el resto)
;;    estado = idle
;; }

isr_boton:
   ei
   push af
   ld a, (estado)
   cp E_IDLE
   jr nz, else_boton
     ;; arranco secuencia de medida
     ;; salidas
   ld a, bit_on + bit_midiendo
   out (salidas), a
     ;; max=0
   ld a, 0
   ld (max), a
     ;; estado = espero_umbral
   ld a, E_WUMBRAL
   ld (estado), a
   jr fin_boton
else_boton:
     ;; salidas
   ld a, bit_not_ok
   out (salidas), a
   ld a, 0
   out (vel_max), a
     ;; estado = idle
   ld a, E_IDLE
   ld (estado), a
fin_boton:
   pop af
   reti

;; c.2) valid
;;    leo medida
;;    si ( medida > max ) {
;;       max = medida
;;    }
;;    si ( medida > umbral ){
;;       si ( estado == espero_umbral ){
;;          estado = espero_fin
;;       }
;;    }else si ( medida < umbral ){
;;    /* si son iguales no hago nada */
;;       si ( estado == espero_fin ){
;;        /* ya habia superado umbral*/
;;        /* termino medida exitosa */
;;          detengo timer
;;          apago radar
;;          actualizo salidas
;;           (ok=1, desactivo el resto
;;          vel_max = max
;;          estado = idle
;;       }
;;    }

isr_valid:
   ei
   push af
   push bc
   ld a, (max)
   ld b, a
   in a, (dat_o_radar)
   cp b
   jp M, no_mayor_max
```

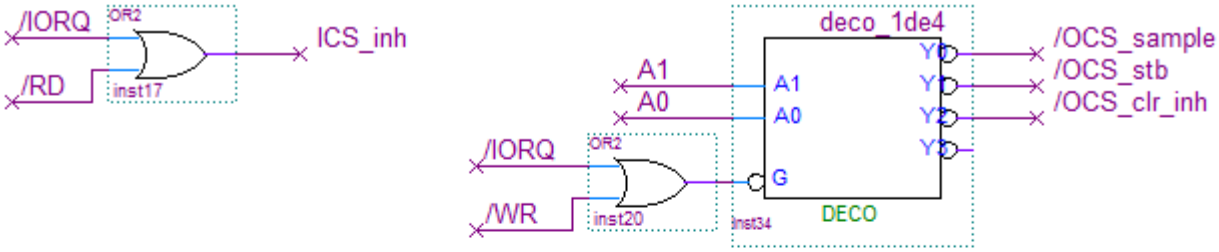
```
   jr Z, no_mayor_max
     ;; si medida > max
   ld (max), a
no_mayor_max:
   cp UMBRAL
   jp M, else_umbral
   jr Z, fin_isr_valid
     ;; medida es mayor que umbral
   ld a, (estado)
   cp E_WUMBRAL
   jr nz, fin_isr_valid
     ;; todavía no había superado umbral
     ;; estado = espero_fin
   ld a, E_WFIN
   ld (estado), a
   jr fin_isr_valid

else_umbral:
     ;; medida es menor que umbral
   ld a, (estado)
   cp E_WFIN
   jr nz, fin_isr_valid
      /* ya habia superado umbral*/
      /* termino medida exitosa */
     ;; salidas
   ld a, bit_ok
   out (salidas), a
     ;;
   ld a, (max)
   out (vel_max), a
     ;; estado = idle
   ld a, E_IDLE
   ld (estado), a

fin_isr_valid:
   pop bc
   pop af
   reti

ppal:
   ...
;; tabla de interrupciones
   org 0x1000
tabla:
   dw isr0
   dw isr2
   dw isr4
   dw isr6
   dw isr8
   dw isr_boton
   dw isr_valid
   dw isr_timer

;; variables
   org 0x8000
estado: db 0
max:    db 0
```

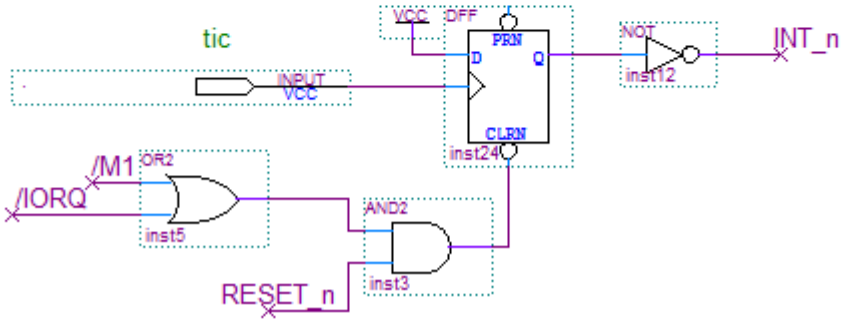
PROBLEMA 2 - a) Hardware

```
;; interrupción: con tic, FF borrado con /inta y reset
;; Puertos entrada: inhibit con FF de handshake
;; Puertos salida: sample[8]
;; Pulsos: stb, clr_inh
```









b)

```
   N equ 64

;; b) isr_tic

;; si ( indice == N)  {
;;    delta = -1
;;    inhibit = 0xFF
;; }elseif ( indice == 0 ) {
;;    delta = +1
;;    inhibit = 0xFF
;;    signo = -signo
;; }
;; indice = indice + delta
;;
;; tmp = tabla[ indice ]
;; si ( signo == negativo ) {
;;    tmp = -tmp
;; }
;; tmp = tmp and inhibit
;; sample = tmp
;; pulso en stb

   org 0x38
isr_tic:
   push af
   push hl
   ld a, (indice)
   cp N
   jr nz else_N
   ld a, -1
   ld (delta), a
   ld a, 0xFF
   ld (inhibit), a
   jr finsi_N
else_N:
   cp 0
   jr nz finsi_0
   ls a, 1
   ld (delta), a
   ld a, 0xFF
   ld (inhibit), a
   ld a, (signo)
   neg
   ld (signo), a
finsi_0:
finsi_N:
     ;; indice = indice + delta
   ld hl, indice
   ld a, (delta)
   add a, (hl)
   ld (hl), a
     ;; tmp = tabla[ indice ]
   ld hl, tabla
   ld b, 0
   ld a, (indice)
   ld c, a
   add hl, bc
   ld b, (hl)
     ;; si ( signo == negativo ) {
   ld a, (signo)
   cp (-1)
   jr nz, finsi_negativo
   ld a, b
     ;;    tmp = -tmp
   neg
   ld b, a
finsi_negativo:
     ;; tmp = tmp and inhibit
   ld a, (inhibit)
   and a, b
     ;; sample = tmp
   out (sample), a
     ;; pulso en stb
   out (stb), a
   pop hl
   pop af
   ei
   ret



;;;
;;; c) programa pincipal
;;;

main:
   in a, (inh)
   and 1
   jr z, main
   out (clr_inh),a  ; clear FF handshake
   ld a, 0xFF
   ld (inhibit),a   ; inhibit=true
   jr main
```

```
;;;
;;; d) init y reservas
;;;

inh     equ 0
sample  equ 0
stb     equ 1
clr_inh equ 2

    .org 0
    ld sp,0
    im 1

; ultima muestra del ultimo cuadrante
; para que la próxima sea la primer
; muestra del primer cuadrante.
; Ultima muestra ultmo cuadrante:
; signo=-1, delta=-1, indice=1

    ld a,-1
    ld (signo),a
    ld (delta),a
    ld a,1
    ld (indice),a
    ld a,0
    ld (inhibit),a
    out (clr_inh),a

    ei
    jp main

 ;; tabla en ROM con primer cuad sin(t)
tabla:
db 0
db 3
db 6
db 9
db 12
…
db 127

.org 0x8000
;; variables
signo: db
indice: db
delta: db
inhibit: db
```