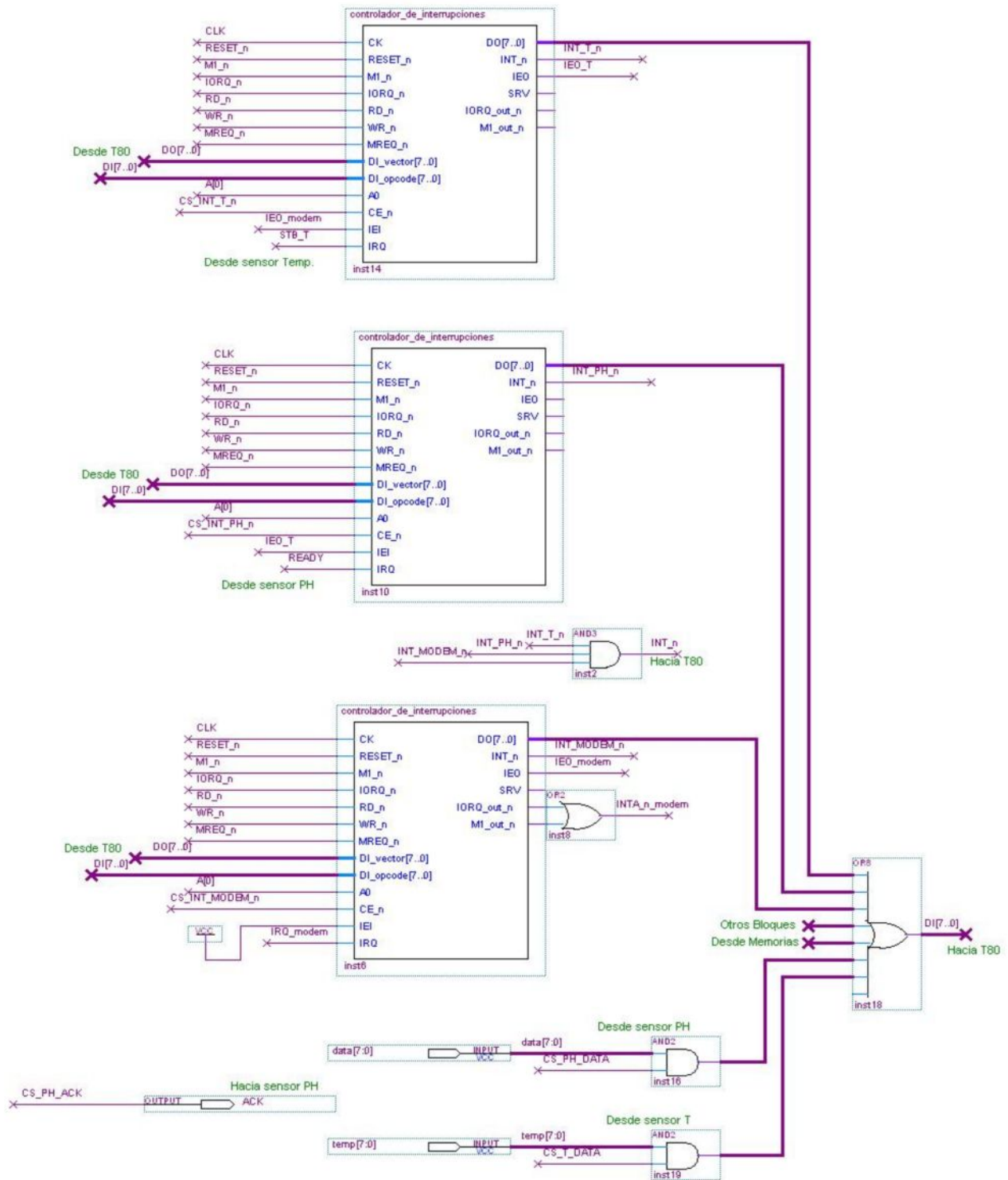
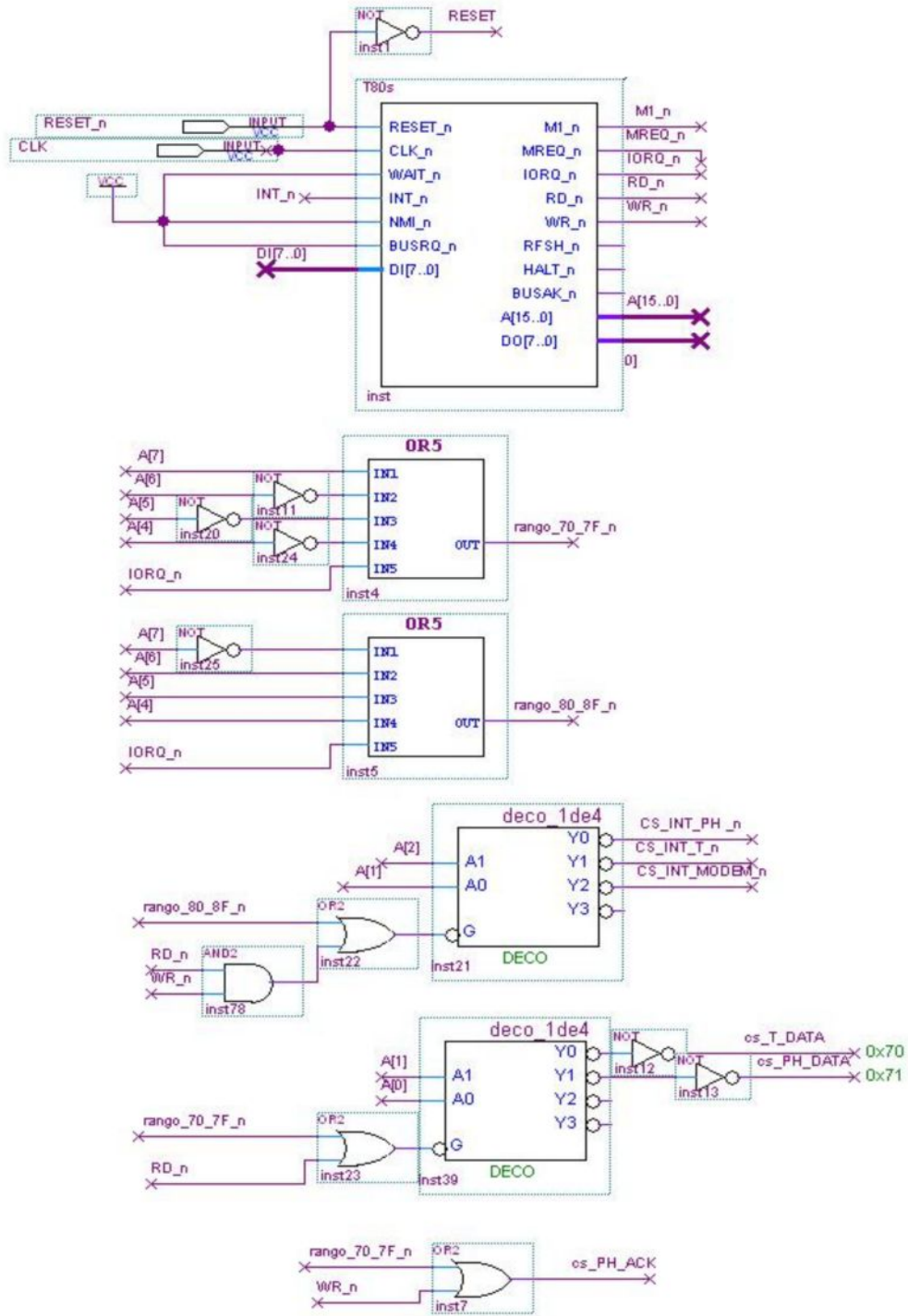


PROBLEMA 1 - a) Hardware





```

b) inicialización y reserva memoria
CI_PH_L equ 0x80
CI_PH_H equ 0x81
CI_T_L equ 0x82
CI_T_H equ 0x83
CI_M_L equ 0x84
CI_M_H equ 0x85
T_data equ 0x70
PH_data equ 0x71
PH_ACK equ 0x70

VI_M equ 0x00
VI_T equ 0x02
VI_PH equ 0x04

org 0x0000
ld SP, 0000h
im2
ld A, T_INT / 256
ld I,A

; VI a CI M
ld A,VI_M
out (CI_M_L), A
out (CI_M_H), A ; borro pendientes
; VI a CI temperatura
ld A,VI_T
out (CI_T_L), A ; borro pendientes
out (CI_T_H), A
ld A,VI_PH
; VI a CI PH
out (CI_PH_L), A
out (CI_PH_H), A ; borro pendientes

ld A,0
ld (puntero_T), A ; inicio tabla Temp.
ld (puntero_PH), A ; inicio tabla PH
EI
jp ppal

ORG 4000h
T_INT:
DW 0x3500 ; rutint modem
DW RUTINT_T
DW RUTINT_PH

org 8000h
tabla_T: DS 128
puntero_T: DB

org 8100h
tabla_PH: DS 128
puntero_PH: DB

c) RUTINT_PH
org 5000h
RUTINT_PH:
EI
push AF
push HL

; almaceno valor en tabla
ld HL, tabla_PH

ld A, (puntero_PH)
ld L,A
in A, (PH_data)
;;; pulso de ACK, no importa que tenga A
out (PH_ACK),A
ld (HL),A
inc L
in A, (PH_data)
ld (HL),A
inc L

;verifico si ya van 128 valores
ld A,L
cp A, 128
jp Z, envio_PH

ld (puntero_PH),A
jp fin_PH

envio_PH:
ld A,0
ld (puntero_PH),A
ld HL,tabla_PH
call Transmision

fin:
pop HL
pop AF
reti

d) RUTINT_T
org 6000h
RUTINT_T:
ei
push AF
push HL

; almaceno valor en tabla
ld HL, tabla_T
ld A, (puntero_T)
ld L,A
in A, (T_data)
ld (HL),A
inc L

;verifico si ya van 128 valores
ld A,L
cp A, 128
jp Z, envio_T

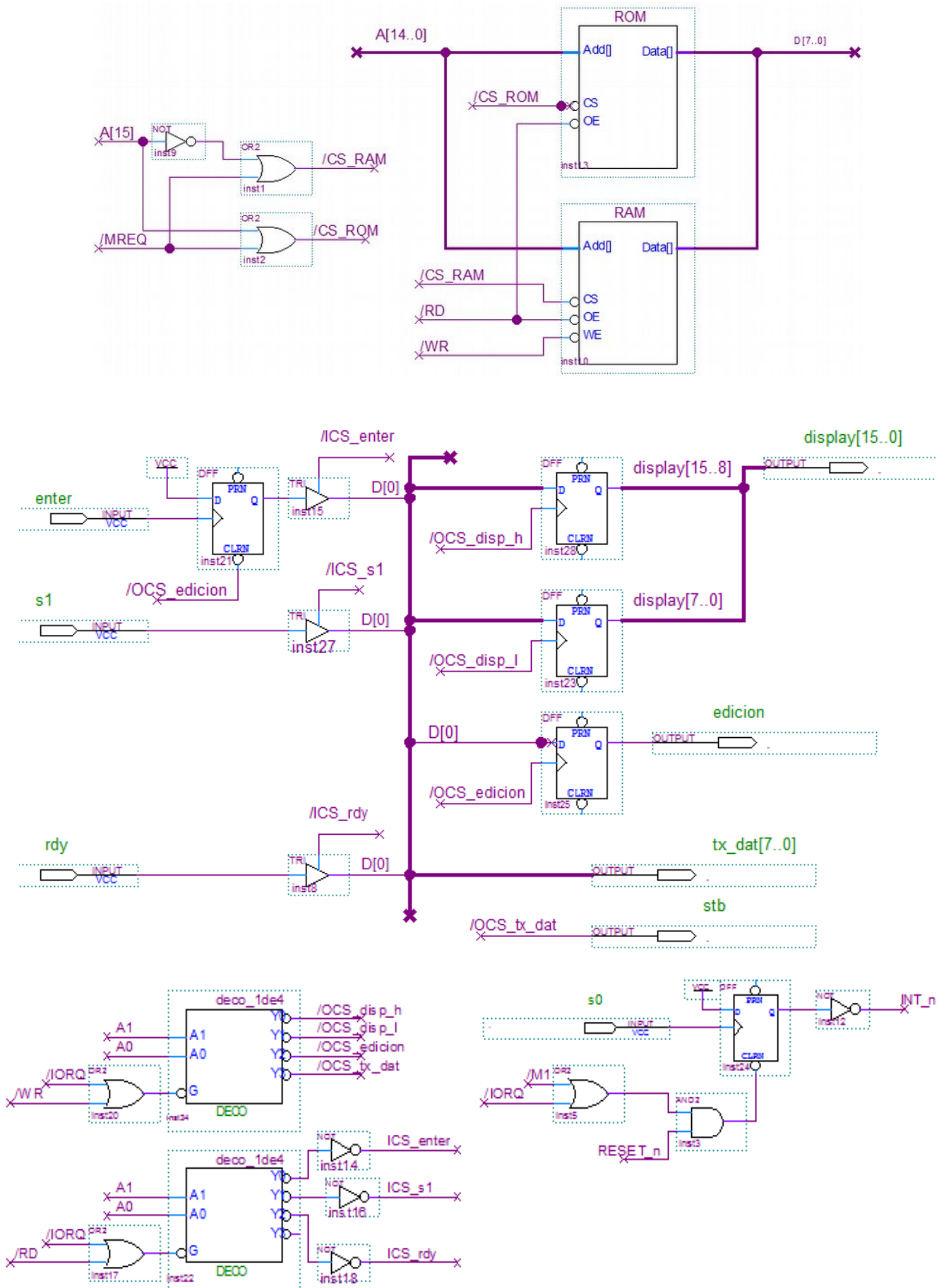
ld (puntero_T),A
jp fin_T

envio_T:
ld A,0
ld (puntero_T),A
ld HL,tabla_T
call Transmision

fin:
pop HL
pop AF
reti

```

PROBLEMA 2 - a) Hardware



```

b.1) lazo infinito prog. principal      ;; fin
;;; Puertos
enter      equ 0
s1         equ 1
rdy        equ 2

disp_h     equ 0
disp_l     equ 1
edicion    equ 2
tx_dat     equ 3

;;; Constantes
m_EDICION  equ 1
m_REPOSO   equ 0

; --- lazo infinito prog. Principal
forever:
    ;; espero enter para ir a edicion
while_enter:
    in a, (enter)
    bit 0, a
    jr z, while_enter
    ld a, m_EDICION
    ld (modo), a
    ;; prendo edicion y borro ff
    out (edicion), a
    ;; espero enter para fin de edicion
while_enter2:
    in a, (enter)
    bit 0, a
    jr z, while_enter2
    ld a, m_REPOSO
    ld (modo), a
    ;; apago edicion y borro ff
    out (edicion), a
    ;; transmito
    ;; byte menos significativo
while_rdyn:
    in a, (rdy)
    bit 0, a
    jr z, while_rdyn
    ld a, (VALOR)
    out (tx_dat), a
    ;; byte MAS significativo
while_rdyn2:
    in a, (rdy)
    bit 0, a
    jr z, while_rdyn2
    ld a, (VALOR+1)
    out (tx_dat), a
    ;; vuelvo a esperar enter
    jr forever

b.2) Rutina atención a interrupción.
;;; preservó regs
;;; si modo edicion entonces
;;; si s1=0 entonces
;;; incremento valor
;;; else
;;; decremento valor
;;; fin_si s1
;;; actualizo display
;;; fin_si modo
;;; restaura registros

org 0x0038
push af
push hl
    ;; si modo edicion
ld a, m_EDICION
cp (modo)
jr nz, fin_isr
    ;; si s1=0
in a, (s1)
bit 0, a
jr nz, else_s1
    ;; incremento valor
ld hl, (valor)
inc hl
ld (valor), hl
jr fin_si_s1
else_s1:
    ;; else decremento
ld hl, (valor)
dec hl
ld (valor), hl
fin_si_s1:
    ;; actualizo display
ld a, l
out (disp_l), a
ld a, h
out (disp_h), a
fin_isr:
pop hl
pop af
ei
ret

b.iii) inicializacion
org 0
    ;; sp, modo int
ld sp, 0
im 1
    ;; variables (valor y modo)
ld hl, VAL_INI
ld (valor), hl
ld a, m_REPOSO
ld (modo), a
    ;; puertos
    ;; (display, edicion, ff enter)
ld hl, (valor)
ld a, l
out (disp_l), a
ld a, h
out (disp_h), a
ld a, 0
    ;; apago edicion y borro ff
out (edicion), a
    ;; ei y salto a loop principal
ei
jr forever

;;;;;;;;;; --- reserva memoria
ORG 0x8000
valor: dw
modo: db

```