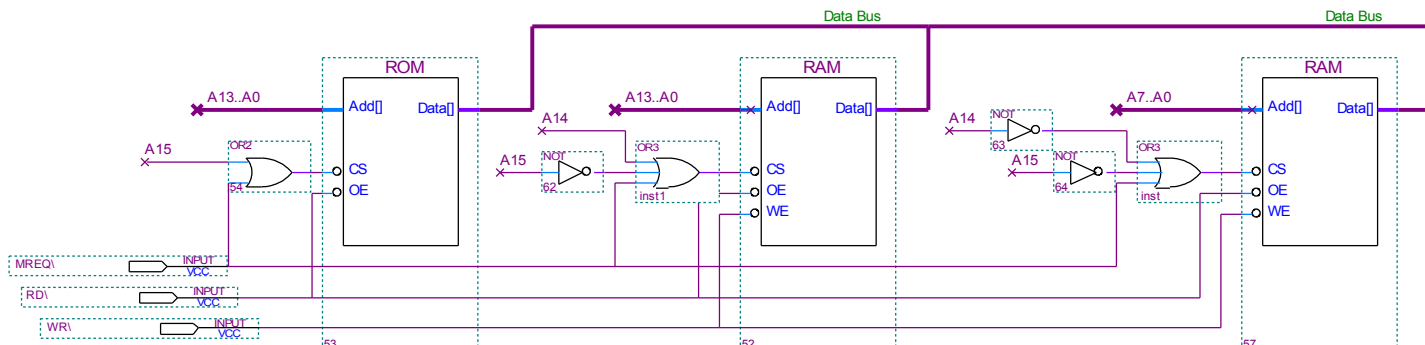


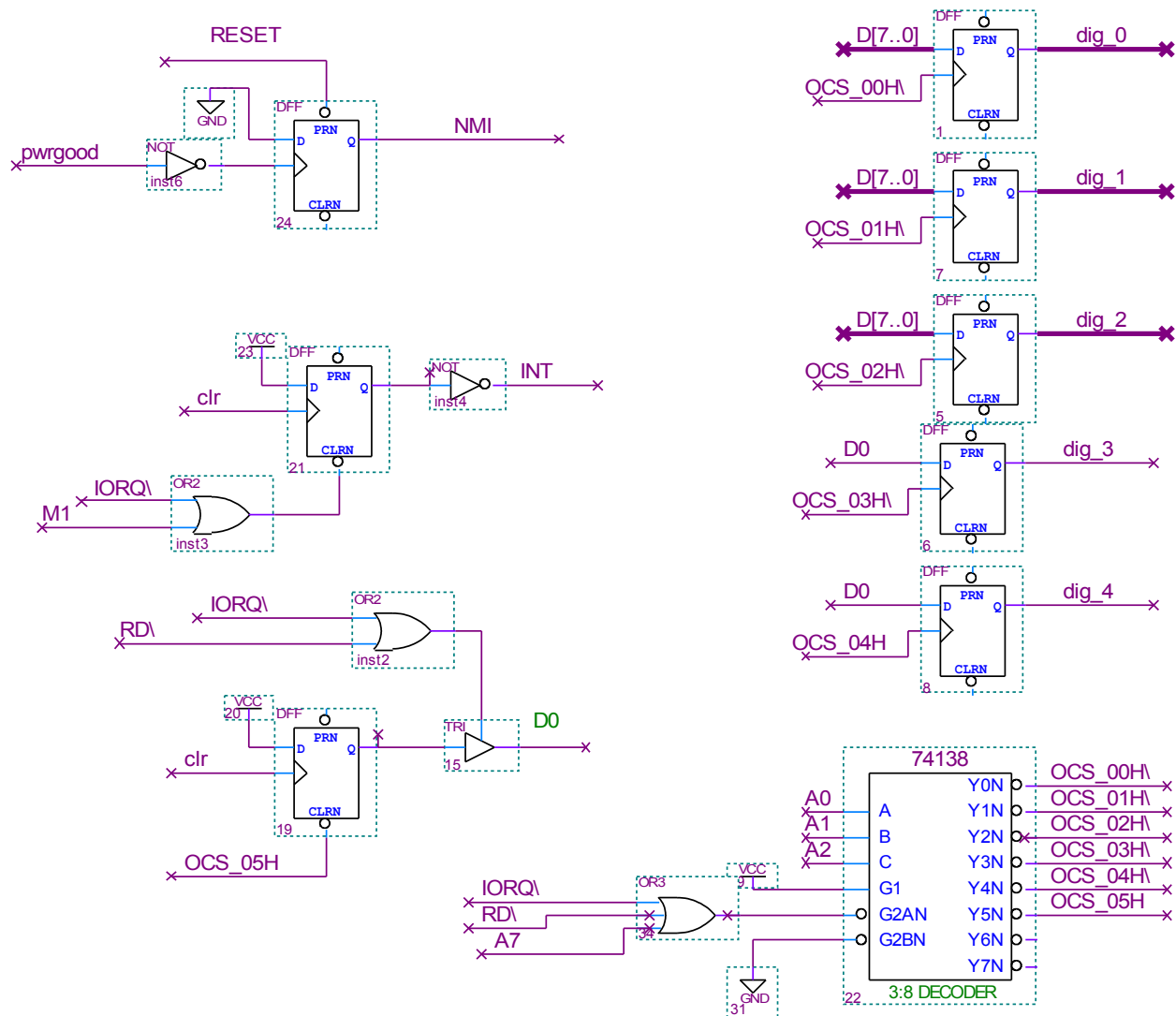
Problema 1 (a)

Memoria:



Puertos:

- 5 puertos de salida de 7 bits (dig_0 a dig_4)
- 1 puerto entrada (clr) con ff para detectar flancos
- 1 pulso para borrar ff de clr



```
b) software:

clr          equ    0
borro_clr   equ    05h

          org 8000h
result:     db
           db
           db
           db
cuenta:     dw

          org 0FFFEh
cuentabk:   dw

; inicializacion

          org 0
          ld sp, 0
          ld bc, (cuentabk)
          ld (cuenta), bc
          im 1
          ei
          call display
          jp ppal

          org 100h

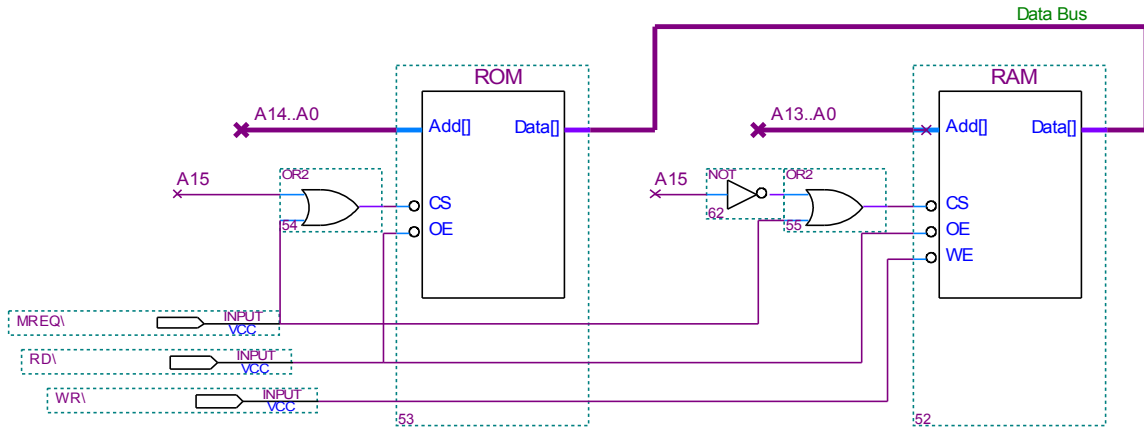
ppal: in a, (clr)
      bit 0, a
      jr nz, ppal
      out (borro_clr), a
      ld hl, 0
      ld (cuenta), hl
      call display
      jr ppal

display:
      push hl
      push af
      ld hl, (cuenta)
      call convert
      ld a, (resultado)
      out (dig_0), a
      ld a, (resultado+1)
      out (dig_1), a
      ld a, (resultado+2)
      out (dig_2), a
      ld a, (resultado+3)
      out (dig_3), a
      ld a, (resultado+4)
      out (dig_4), a
      pop af
      pop hl
      ret

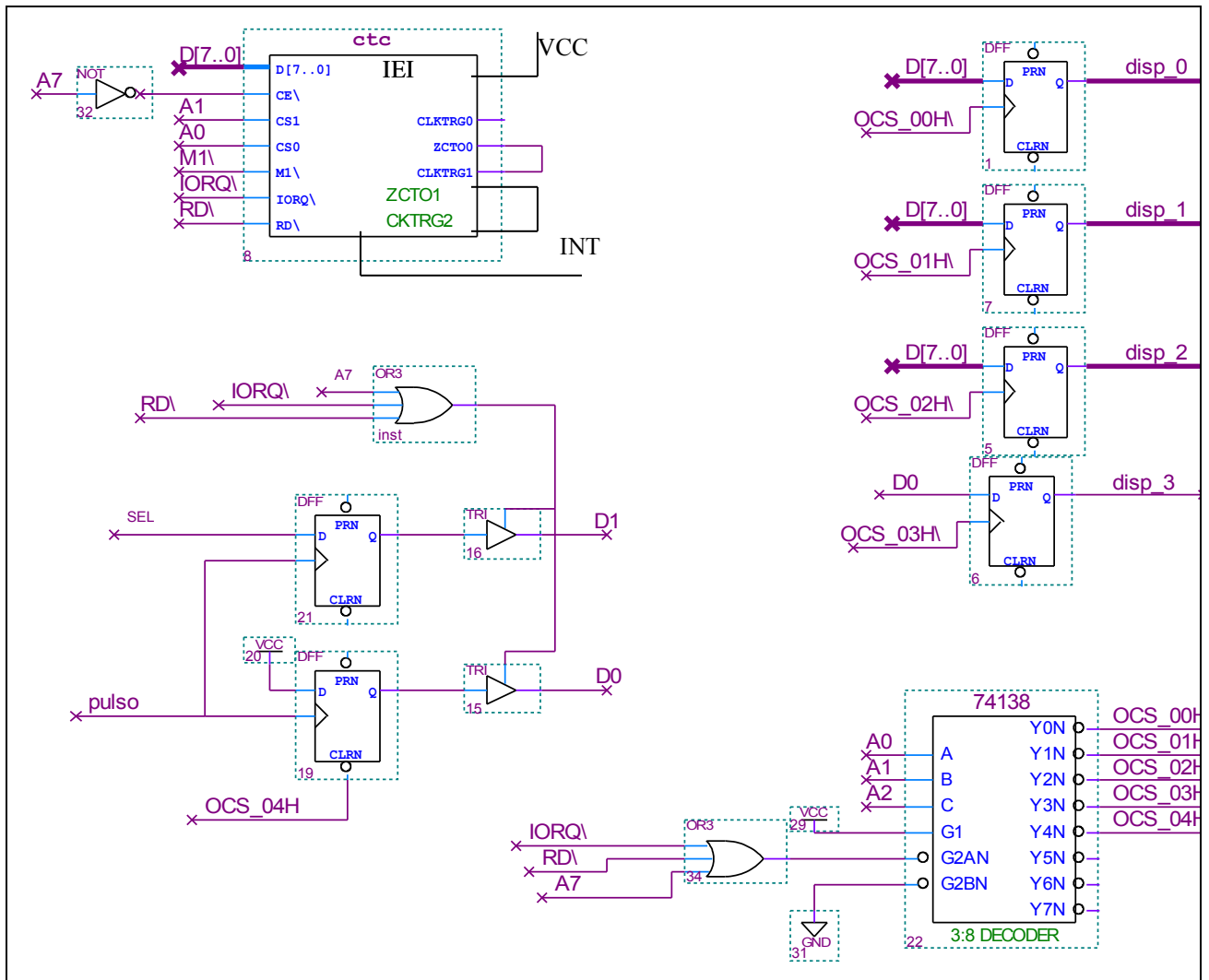
; -- rutina atencion a int
          org 38h
rutint:   push af
          push hl
          ld hl, (cuenta)
          inc hl
          ld (cuenta), hl
          call display
          pop hl
          pop af
          ei
          ret

; -- rutina nmi
          org 66h
rutnmi:   ld hl, (cuenta)
          ld (cuentabk), hl
loop:    jp loop
```

Problema 2 (a)
Memoria:



Puertos e interrupciones:



Problema 2 (b y c)

```

;----- CONSTANTES -----
display EQU 00h
botones EQU 04h
cl_botones EQU 04h
pulso EQU 0d
sel EQU 1d
CTC_C0 EQU 80h
CTC_C1 EQU 81h
CTC_C2 EQU 82h
CW0 EQU 011X0111
CT0 EQU 250
CW1 EQU 01X1X111
CT1 EQU 250
CW2 EQU 11X1X111
CT2 EQU 15

;----- RESERVA DE MEMORIA -----
ORG 8000h
tiempo DS 4
v_hora DB
v_minuto DB

;----- RUT. INTERRUPCIONES -----
ORG 0038h
PUSH AF
PUSH BC
PUSH HL ; preservo registros

LD A, (v_minuto)
INC A
CP A, 60d ; incremento minutos y
JP Z, inc_hora ; chequeo desborde

LD (v_minuto), A ; actualizo v_minutos
JP fin ; si no hay desborde

inc_hora:
LD A, 00d ; actualizo v_minutos
LD (v_minuto), A ; si hubo desborde

LD A, (v_hora)
INC A
CP A, 24d ; incremento hora y
JP Z, reset_hora ; chequeo desborde

LD (v_hora), A ; actualizo v_hora
JP fin ; si no hay desborde

reset_hora:
LD A, 00d ; actualizo v_hora
LD (v_hora), A ; si hubo desborde

fin:
LD A, (v_minuto)
LD C, A
LD A, (v_hora)

LD B, A ; cargo en BC tiempo
LD HL, tiempo ; HL = comienzo RAM
reservada
CALL dispdisplay ; convierto tiempo a
7segmentos

LD C, display ; C=dir. primer display
LD B, 4 ; B=cantidad de display
act_display:
LD A, (HL)
OUT (C), A
INC HL
INC C
DJNZ act_display ; actualizo los display

POP HL
POP BC
POP AF ; restauro registros
EI ; habilito interrupciones
RETI ; retorno

;----- INICIALIZACIONES -----
ORG 0000h
LD SP, C000H ; inicializo SP

LD A, 0
LD (v_hora), A
LD (v_minuto), A ; v_hora=0, v_minuto=0

LD BC, 0d ; cargo en BC tiempo
00:00hs
LD HL, tiempo ; HL = comienzo RAM
reservada
CALL dispdisplay ; convierto tiempo a
7segmentos

LD C, display ; C=dir. primer display
LD B, 4 ; B=cantidad de display
act_display:
LD A, (HL)
OUT (C), A
INC HL
INC C
DJNZ act_display ; actualizo los display

LD A, CW0 ; Canal 0 de CTC
OUT (CTC_C0), A ; Timer s/ interrupciones
LD A, CT0 ; prescaler = 256
OUT (CTC_C0), A ; cte = 250

LD A, CW1 ; Canal 1 de CTC
OUT (CTC_C1), A ; Contador
s/interrupciones
LD A, CT1 ; prescaler = 256
OUT (CTC_C1), A ; cte = 256

LD A, CW2 ; Canal 2 de CTC
OUT (CTC_C2), A ; Contador
c/interrupciones

```

```
LD A, CT2          ; prescaler = 256
OUT (CTC_C2), A    ; cte = 15

EI                 ; habilito interrpciones

;----- PRINCIPAL -----
main:
  IN A, (botones)
  BIT pulso, A
  JP Z, main        ; Polling de PULSO

  OUT (cl_botones), A
  BIT sel, A
  JP Z, inc_hora    ; estado de sel?

inc_minuto:
  LD A, (v_minuto) ; incremento minutos y

INC A              ; chequeo desborde
CP A, 60d
JP NZ, save_minuto
LD A, 0            ; si hubo desborde
                  minuto=00

save_minuto:
  LD (v_minuto), A ; actualizo v_minutos
  JP main

inc_hora:
  LD A, (v_hora)   ; incremento hora y
  INC A            ; chequeo desborde
  CP A, 24d
  JP NZ, save_hora
  LD A, 0          ; si hubo desborde hora=00

save_hora:
  LD (v_hora), A  ; actualizo v_hora
  JP main
```