

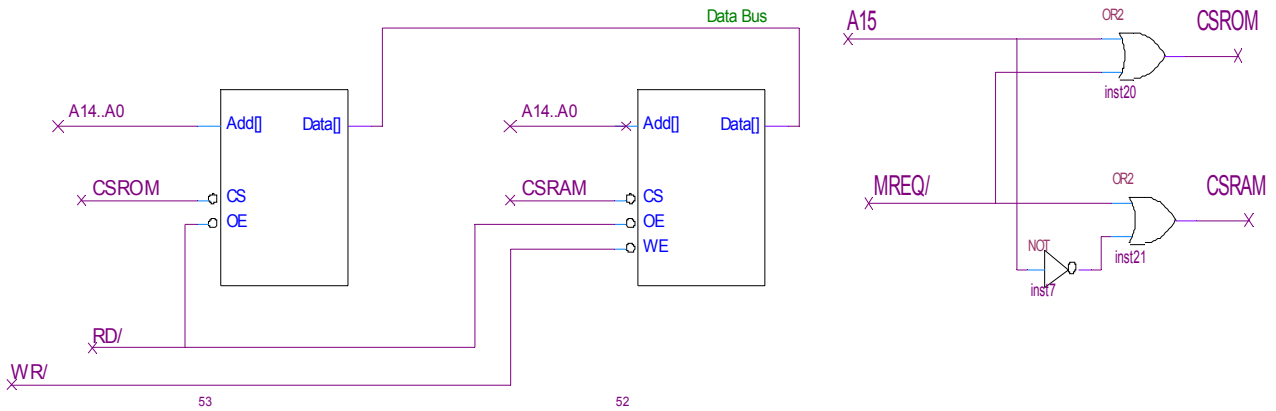
# SOLUCIONES

## PROBLEMA 1

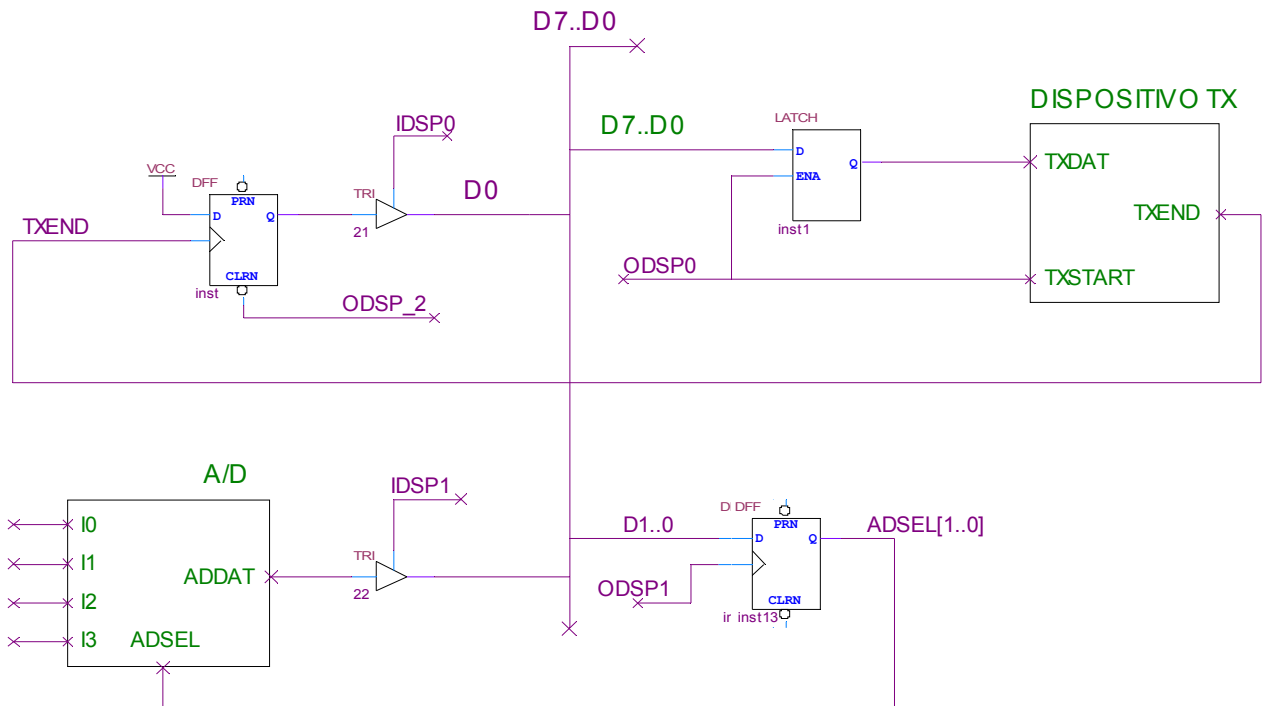
Aclaración letra en pizarrón: todas las tareas del programa principal insumen menos de 1 minuto.

### a) Hardware

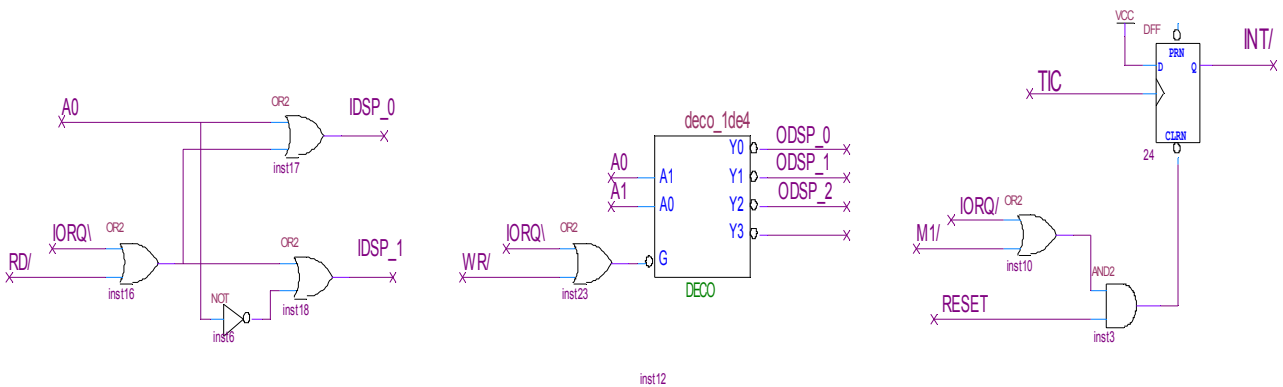
-- Memorias:



-- Puertos:



-- Decodificación e Interrupciones.



## SOLUCIONES

```

b) Rutina interrupcion
;rutint(){
;   preservar estado
;   tic++
;   si tic=100 entonces{
;       tic = 0
;       seg++
;       si seg == 60 entonces{
;           seg = 0
;           FIN_ADQUI = TRUE
;       }
;   }
;   si seg == 59 entonces{
;       tabla100[tic] = lectura puerto
;   }
;   restaurar estado
;   ret
;}
;
;   org 38
rutint:
push af
push hl
ld a, (TIC) ;tic++
inc a
ld (TIC), a
;   ;si tic=100 entonces{
cp 100
jr nz, finsil100
ld a, 0 ; tic = 0
ld (TIC), a
ld a, (SEG) ; seg++
inc a
ld (SEG), a
;   ; si seg == 60
cp 60
jr nz finsi60
ld a, 0
ld (SEG), a ;seg = 0
ld a, TRUE
ld (FIN_ADQUI), a ;FIN_ADQUI = TRUE
finsi60:
finsi100:
;   ;si seg == 59
ld a, (SEG)
cp 59
jr nz, finsi59
ld hl, tabla100
;   ;tabla100[tic] = addat
ld a, (TIC)
ld l, a ; supongo tabla100
;   ; alineado a página
in a, (ADDAT)
ld (hl), a
finsi59:
pop hl
pop af
ei
ret

c) Programa principal
;ppal(){
;   forever{
;       espero FIN_ADQUI == TRUE
;       FIN_ADQUI = FALSE
;       resultado = procesar datos
;       transmitir(minuto)
;       transmitir(resultadoHi)
;       transmitir(resultadoLo)
;       minuto++
;       puerto adsel = minuto
;   }
;}
;
;   org 100h
ppal:
;   ;   espero FIN_ADQUI == TRUE
ld a, (FIN_ADQUI)
or a
jr z, ppal
;   ;   FIN_ADQUI = FALSE
ld a, 0
ld (FIN_ADQUI), a
;   ;   resultado = procesar datos
ld hl, tabla100
call PROCESAR
push DE ; preservo resultado
;   ;   transmitir(minuto)
loop1:
in a, (TXRDY)
bit 0, a
jr z, loop1
ld a, (MIN)
out (TXDAT), a
out (TXENDclr), a
;   ;   transmitir(resultadoHi)
pop DE
loop2:
in a, (TXRDY)
bit 0, a
jr z, loop2
ld a, D
out (TXDAT), a
out (TXENDclr), a
;   ;   transmitir(resultadoLo)
loop3:
in a, (TXRDY)
bit 0, a
jr z, loop3
ld a, E
out (TXDAT), a
out (TXENDclr), a
;   ;   minuto++
ld a, (MIN)
inc a
ld (MIN), a
;   ;   puerto adsel = minuto
out (ADSEL), a
jr ppal

```

**SOLUCIONES**

```

d) Inicialización
;inicializacion{
;   stack
;   modo 1 interrupciones
;   tic = 0
;   seg = 0
;   minuto = 0
;   FIN_ADQUI = FALSE
;   puerto adsel = 0
;   ei
;   jp ppal
;}

; definicion constantes
TRUE equ 0ffh
ADDAT equ 1
ADSEL equ 1
TXDAT equ 0
TXEND equ 0
TXENDclr equ 2

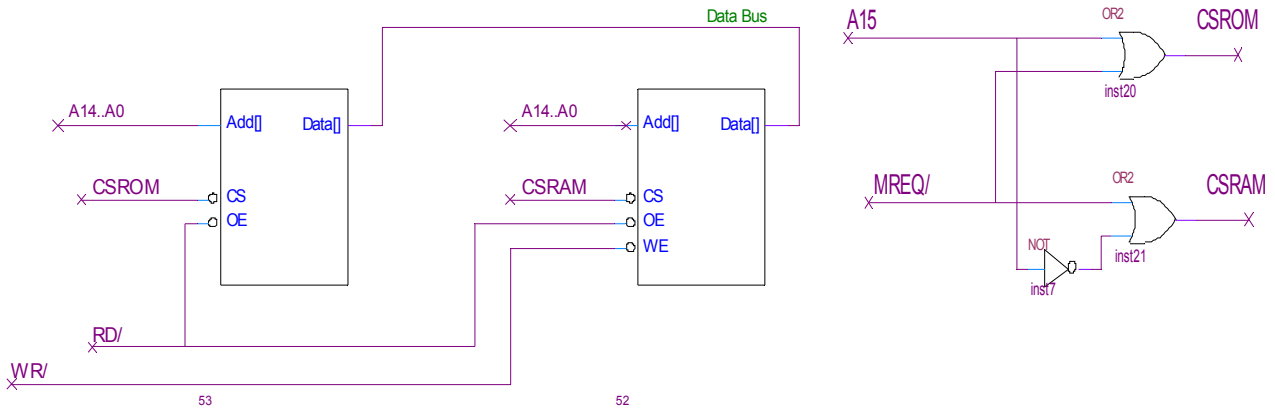
org 8000h
tabla100: DS 100
TIC: DB
SEG: DB
MIN: DB
FIN_ADQI: DB

org 0
ld sp, 0
im 1
ld a, 0
ld (TIC), a
ld (SEG), a
ld (MIN), a
ld (FIN_ADQUI), a
out (ADSEL), a
    
```

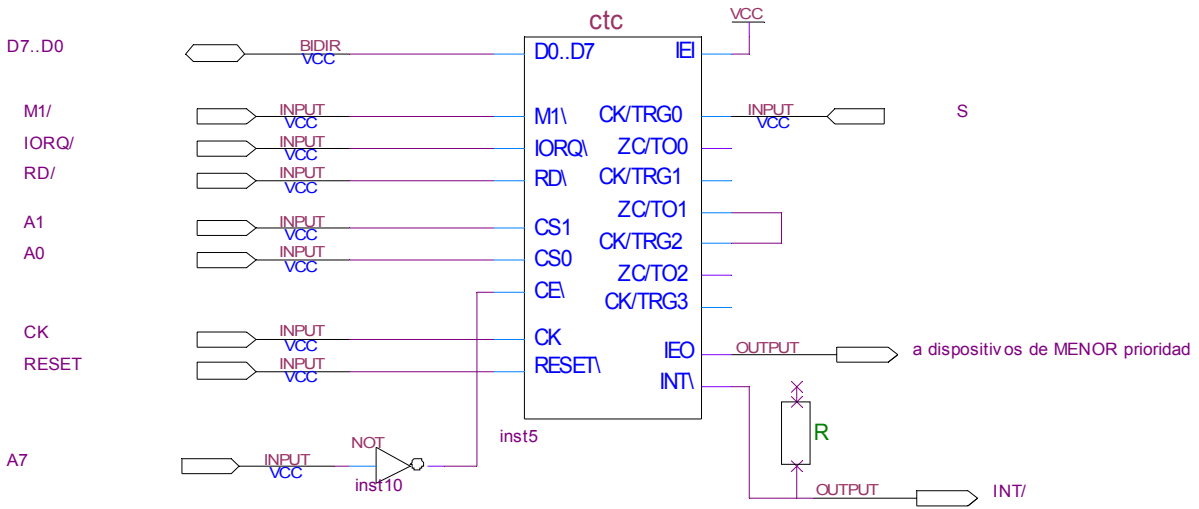
SOLUCIONES

Solución problema 2

a) Hardware. Memoria

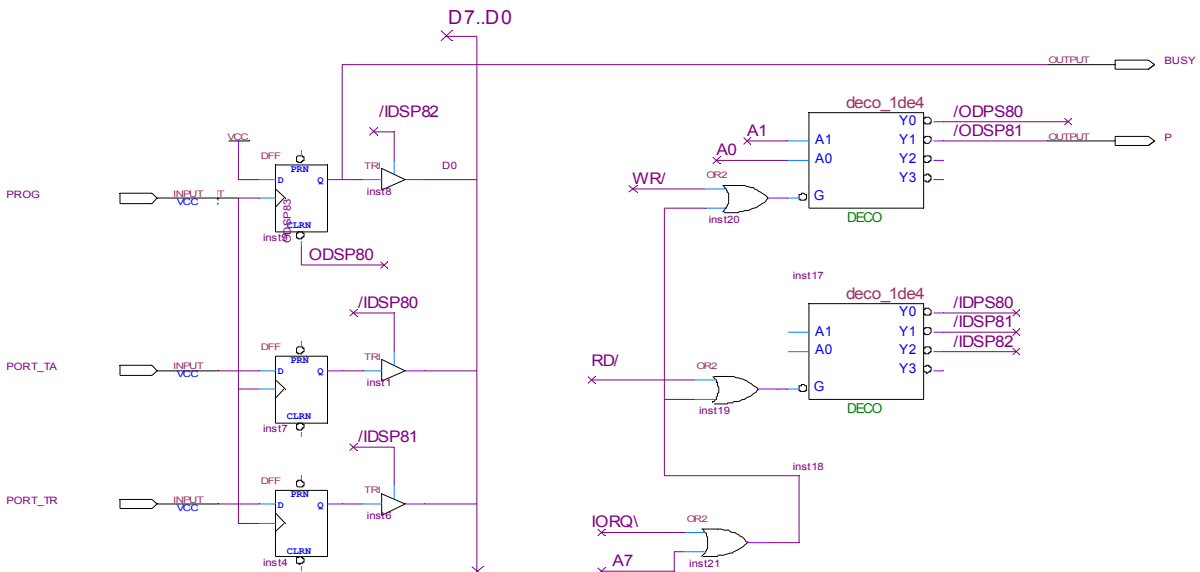


CTC



Puertos y decodificación E/S

Port\_Ta and port\_TR solo valen en el el flanco de subida de PROG, por lo que deben ser memorizados.



## SOLUCIONES

```

b) Programa principal
; forever{
;     mientras ( ff_prog = 0 ) espero ;
;     leo port_ta, escribo val_ta ;
;     leo port_tr, escribo val_tr ;
;     borro FF (y bajo busy) ;
; }
; }

ppal:
;     mientras ( ff_prog = 0 ) espero org 3000h
;     in a, (PROG) isr_timeout:
;     bit 0, a ei
;     jr 0, ppal push af
;     leo port_ta, escribo val_ta ; si (estado = A) {
;     leo port_tr, escribo val_tr ;     ld a, estadoA
;     in a, (port_TA) cp (ESTADO)
;     ld (valTA), a jr nz, else
;     in a, (port_TR) ;     saca pulso Pace
;     ld (valTR), a out (pulsoP), a
;     borro FF (y bajo busy) ;     deshabilito CTC sensado
;     out (borroPROG), a ld a, CWDISABLE
;     jr ppal out (ctc0), a
;                                     prog. canal timeout = Tr

c) Rutinas atención interrupción
; -- rutint Sensado{
;     preservio
;     si (estado = A) {
;         deshabilito CTC sensado ;
;         prog. canal timeout = Tr
;         estado = R
;     }
;     restaurio
;     reti
; }

;                                     -- estado es R
;                                     programo CTC sensado
;                                     ld a, cwCTC0
;                                     out (ctc0), a
;                                     ld a, cteCTC0
;                                     out (ctc2), a
;                                     prog. canal timeout = Ta

;                                     ld a, cwCTC2
;                                     out (ctc2), a
;                                     ld a, (valTA); timeout = Ta
;                                     out (ctc2), a
;                                     estado = A
;                                     ld a, estadoA
;                                     ld (ESTADO), a
;                                     finsi

;                                     else: ;
;                                     -- estado es R
;                                     programo CTC sensado
;                                     ld a, cwCTC0
;                                     out (ctc0), a
;                                     ld a, cteCTC0
;                                     out (ctc2), a
;                                     prog. canal timeout = Ta

;                                     ld a, cwCTC2
;                                     out (ctc2), a
;                                     ld a, (valTA); timeout = Ta
;                                     out (ctc2), a
;                                     estado = A
;                                     ld a, estadoA
;                                     ld (ESTADO), a
;                                     finsi:
;                                     pop af
;                                     reti

;                                     estado = R
;                                     ld a, estadoR
;                                     ld (ESTADO), a

finsiA:
;     pop af
;     reti

; -- rutint timeout{
;     preservio
;     si (estado = A) {
;         saca pulso Pace
;         deshabilito CTC sensado
;         prog. canal timeout = Tr
;         estado = R
;     }
; }

```

**SOLUCIONES**

```

d)  -- inicializacion                                ; constantes
org 0                                           PROG          equ          82h
ld sp, 0                                       ; stack      borroPROG     equ          80h
im 2                                           ; modo int   portTA       equ          80h
ld a, TABLAHI; tabla int en ROM              portTR       equ          81h
ld I, a                                       pulsoP       equ          81h
ld a, VAL_TA_INI   ; val_ta y val_tr         ctc0         equ          0
ld (valTA), a                                  ctc1         equ          1
ld a, VAL_TR_INI                                  ctc2         equ          2
ld (valTR), a                                  vecCTC0      equ          0

                                           ; borro ff_prog
out (borroPROG), a

                                           ; canal 0, entrada S
                                           ; ei|counter|sigue cte|reset
ld a, vecCTC0                                  cwCTC0       equ          11x0x111B
out (ctc10), a                                  ; disable canal 0: di|counter
                                           CWDISABLE   equ          01x0x001B
                                           ; (timer, pre = 16, N = 20)
                                           cteCTC0     equ          1
ld a, cwCTC1
out (ctc1), a                                  ; canal 1, timer
ld a, cteCTC1                                  ; di|timer|pre16|autotrg|sigue cte|reset
out (ctc1), a                                  cwCTC1       equ          000x0111B
                                           cteCTC1     equ          20
                                           ; sense (counter, N = 1)
ld a, cwCTC0
out (ctc0), a
ld a, cteCTC0
out (ctc0), a
                                           ; canal 2 cuenta salida canal 1
                                           ; ei|counter|sigue cte|reset
                                           cwCTC2       equ          11x0x111B
                                           estadoA      equ          0
                                           estadoR      equ          1
                                           VAL_TA_INI  equ          70
                                           VAL_TR_INI  equ          30
                                           ; ctc canal T2
                                           ; (counter, N = val_ta)
ld a, cwCTC2
out (ctc2), a
ld a, (valTA); timeout = Ta
out (ctc2), a

                                           ; reserva de memoria
                                           org 1000h
                                           ; tabla interrupciones en ROM
TABLA:
DW   isr_sens   ; canal0, S
DW
DW   isr_timer  ; canal2, ta y tr
DW

                                           org 8000h
                                           ; variables en RAM
valTA DB
valTR DB
ESTADO DB
    
```