PD32

Esercitazione sull'interfacciamento con un dispositivo di IO

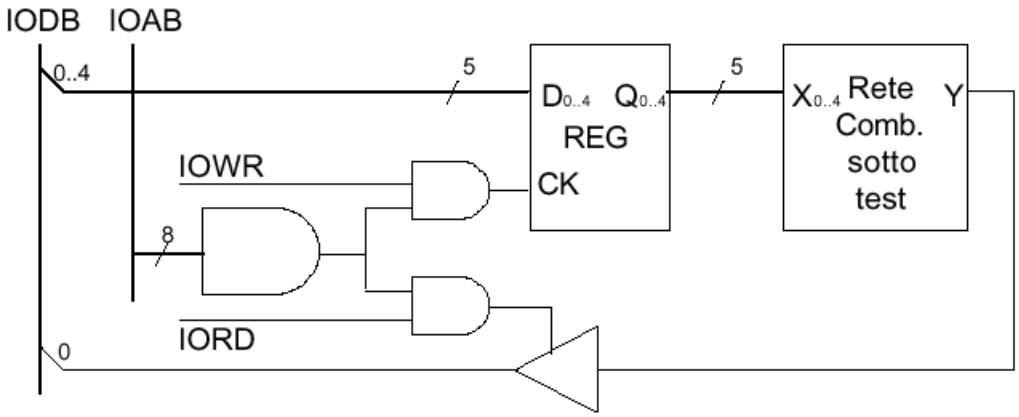
Domanda #5 – 14/09/2000

Si dispone di un PD32 per effettuare il collaudo di un circuito integrato combinatorio con 5 ingressi e una uscita, sospettato di essere guasto.Il collaudo consiste nella verifica del comportamento del circuito combinatorio secondo la tavola di verità memorizzata all'indirizzo TRUTAB della memoria del PD32. L'esito (sano/guasto) del test va registrato in una variabile all'indirizzo TEST. Disegnare l'interfaccia di I/O del processore e scrivere la routine di test.

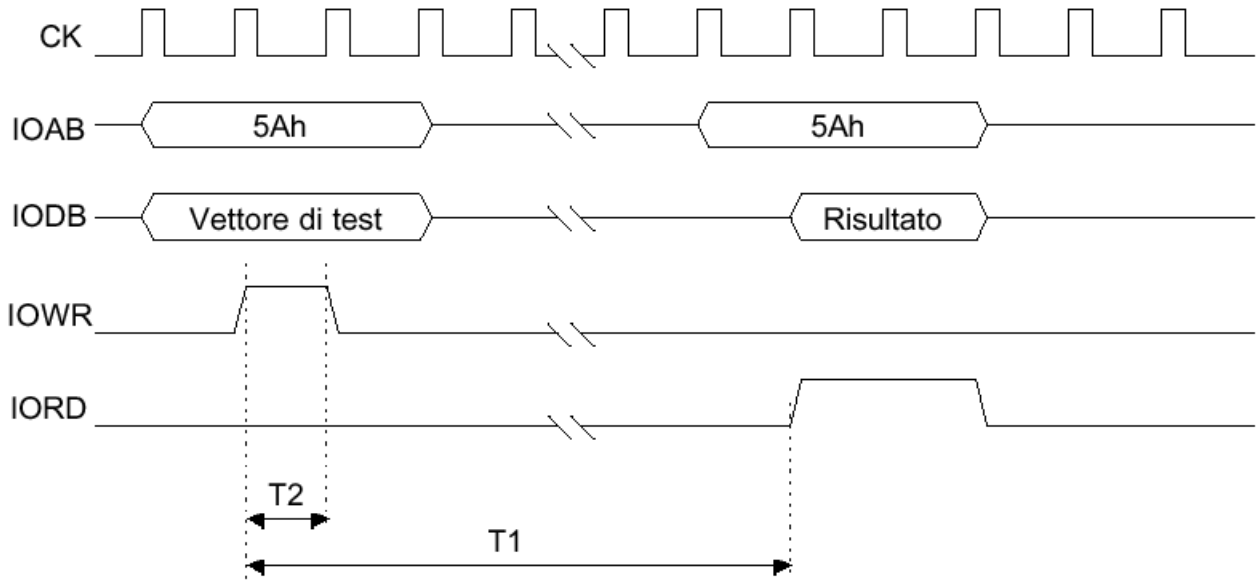
Impostazione del sistema:

- **HW**
  - Registro per memorizzare ognuna delle 2^5=32 configurazioni d'ingresso (uscita del PD32)
  - Buffer 3-state d'ingresso (al PD32) per leggere l'uscita della rete
- **SW:**
  - Prevediamo ciclo per generare ognuna delle configurazioni
  - Non prevediamo attesa tra OUT "configurazione" e "IN" uscita r.c., ipotizziamo che delay r.c. < tempo intercorso per l'esecuzione delle 2 istruzioni
  - Memorizziamo tutte le uscite in una LW e facciamo confronto con la tavola in TRUTAB

Interfaccia

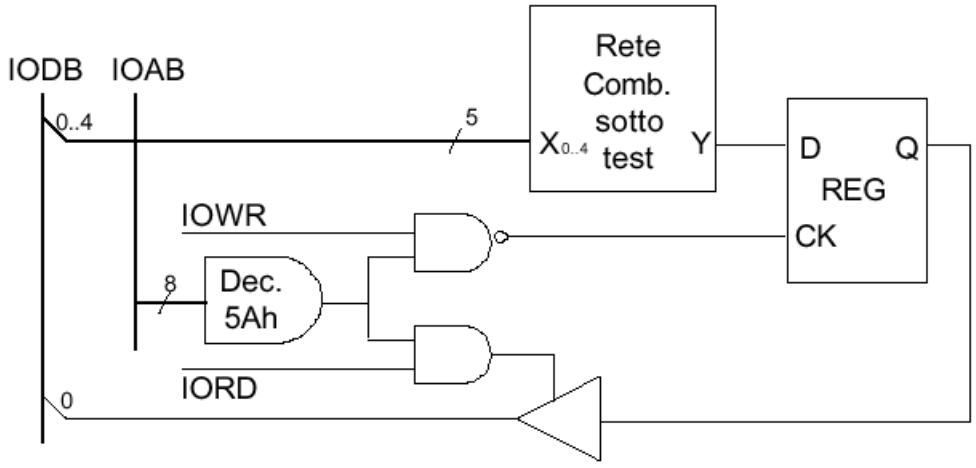


Temporizzazioni



- T1: delay ammissibile per la rete combinatoria per la produzione del dato
- T2: delay ammissibile per la seconda versione

Interfaccia #2



- Memorizziamo uscita della r.c. invece che l'ingresso
- L'uscita della r.c. viene campionata sul fronte di discesa del segnale IOWR
- Assumiamo ipotesi ragionevole che, dato il delay d della r.c., sia d < 1 ciclo di CK del PD32

Subroutine tester

```
                org 400h
        TESTIN equ 05Ah          ;indirizzi diversi perchè
        TESTOUT equ 05Bh         ;richiesto dal simulator

        TRUTAB dl 0A51E780Fh     ; tavola della verità
        TEST db 00h              ; inizializzo risultato
        code
main:           seti
                jsr tester
                halt
tester:         < push r0, r1, r2 >
                movb #0,TEST     ; 0:OK (1:KO)
                xorl r0,r0
next:           outb r0, TESTOUT  ; ciclo di test
                inb TESTIN, r1
                andl #01h, r1    ;forzo bit 1..7 a 0
                asll #1,r2 ;in r2 accumulo risult.parz.
                orl r1,r2   ;scalo r2, bit0 di r1→ bit0 di r2
                addb #1,r0
                cmpb #32,r0
                jnz next
                movl TRUTAB,r1
                cmpl r2,r1
                jz exit
                movb #1,TEST
exit:           < pop r2,r1,r0 >
                ret
        end
```

PD32

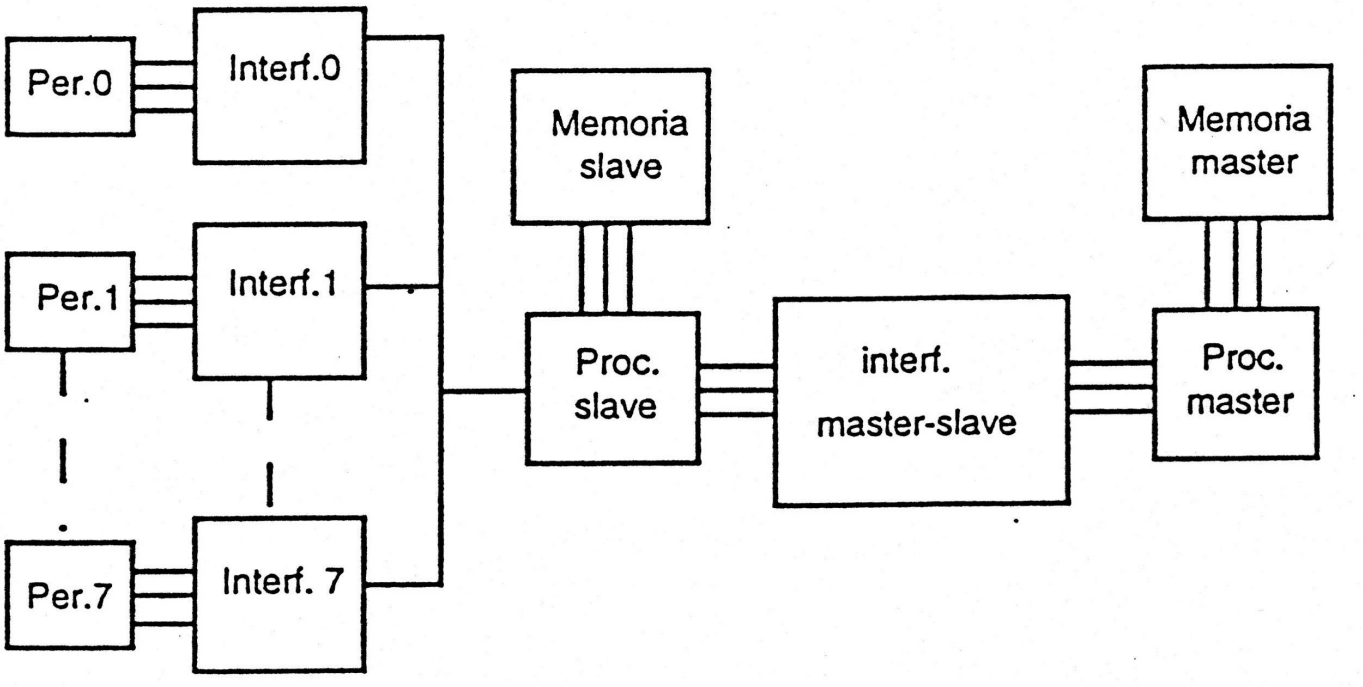
Esercitazioni sull'interfacciamento e la gestione delle interruzioni

Progetto di un sistema Master-Slave

- Specifiche:
  - un processore slave acquisisce dati da 8 perifer.
  - dati prodotti dalle periferiche in formato 8 byte
  - lo slave acquisce, ogni 10 s, il dato presente da ogni periferica e ne produce la media
  - Un processore master può fare richiesta asincrona allo slave dell'ultimo set di dati acquisito
  - se valore media > valore soglia (noto allo slave) lo slave deve trasmettere immediatamente l'ultimo set di dati al master
- Due possibili soluzioni:
  - Sistema cablato (Ipotizziamo processori generici a 32 bit con istruzioni per trattamento busy-waiting o interruzioni)
  - Specializzazione al caso del processore PD32

Fase di Analisi

- In entrambi i casi occorre progettare i componenti seguenti:
  - Interfaccia periferica-processore slave
  - Software slave (calcolo della media, scambio dati con il master)
  - Interfaccia master/slave
  - Software master
- Schema del sistema:



Soluzione con sistema cablato

Interfaccia periferica-processore slave

- Periferica produce dati da 64 bit → associamo ad ogni periferica 2 registri da 32 bit indirizzabili in modo indipendente (2 indirizzi)
- Non prevediamo busy-waiting (acquisizione fatta sui dati presenti nei registri)
- Lo slave assume che le periferiche abbiano indirizzi compresi tra 0 e 15 (2 per periferica)
- Specifiche richiedono acquisizione ultimo set di dati da parte del master in qualsiasi momento:
  - Dati vengono caricati in memoria a partire dall'indirizzo AD1
  - Finito ciclo di acquisizione, il set viene spostato a partire dall'indirizzo AD2

Software processore slave (sistema cablato)

- Acquisizione dati da periferiche
- Spostamento set di dati da AD1 a AD2
- Calcolo media del set di dati
- Controllo se media è superiore a valore soglia
  - Scambio dati con il master (interruzione al master)

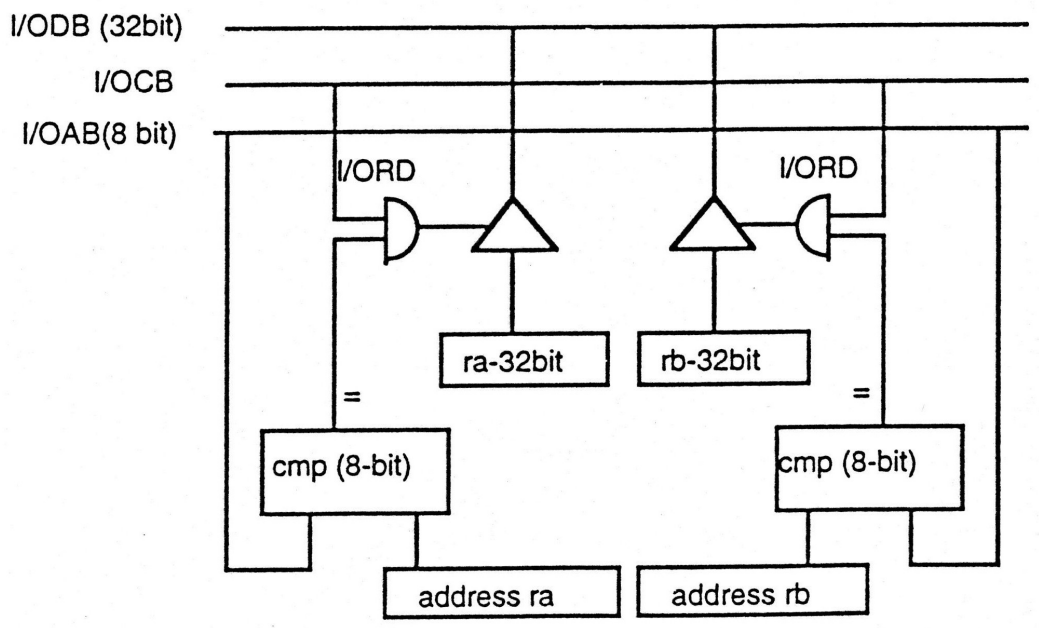
- Scambio dati con il master su interruzione dal master

Routine di acquisizione dati

Accediamo sequenzialmente ai 16 registri (8 coppie).

Per ogni IN viene eseguito un ciclo macchina.

```
            ; acquisizione dati
            ; ro: address periferica
            ; r2: address set AD1
ac-dati:    xorl r0,r0
            xorl r2,r2
ciclo1:     inl r0,r1           ;acquisisci dato in r1
            movl r1,AD1(r2)
            addl #4,r2
            addb #1,r0
            cmpb #16, r0        ; controllo se ho acquisito
            jnz ciclo1          ; tutti i dati
```



Routine di spostamento dati da AD1 a AD2

In questa fase il master non può interrompere lo slave per acquisire il set di dati.

```
        ; spostamento dati da AD1 a AD2
sp-dati: clri
        xorl r0,r0
ciclo2:  movl AD1(r0), AD2(r0)+
        cmpb #64, r0     ; controllo se ho spostato
        jnz ciclo2       ; tutti i dati
        seti
```

Routine per il calcolo della media

Sommiamo 8 elementi da 64 bit.

Somma finale di 96 bit contenuta in [R4, R6, R5]

Per la media eseguo lo shift a destra di 3 bit:

R4 → 3bit → R6 →3bit→ R5

R5: somma finale – R6: eventuale riporto

```
            ; ***** calcolo media *****
media:      xorl r0,r0
            xorl r6,r6
            movl AD2,r5
            ; sommo i 32 lsb, risultato in r5, riporto in r6
somma1:     addl #8,r0
            addl AD2(r0),r5
            jnc contin
            addl #1, r6
contin:     cmpb #56,r0
            jnz somma1
            ; sommo i 32 msb, risultato in r6,r5, riporto in r4
            xorl r4,r4
            movl #4,r0
somma2:     addl AD2(r0), r6
            jnc contin1
            addl #1,r4
contin1:    addl#8,r0
            cmpb #60,r0
            jnz somma2
            ; calcolo media di [R4, R6, R5] in [R6, R5]
            lsrl 3,r5
            movl r6,r0
            lsll 29,r0
            addl r0,r5
            lsrl 3,r6
            lsll 29,r4
            addl r4,r6
```

Routine per confronto con valore soglia

Ipotizziamo di avere memorizzato in AD3 il valore soglia nel formato AD3: [MSlw], AD3+4: [LSlw].

```
            ; confronto valore soglia
confr:      movl AD3,r0
            movl #4, r1
            subl r6,r0
            jn gest
            movl AD3(r1),r2
            subl r5,r2
            jn gest
            jnz fine
gest:       jsr scambio-dati
fine:       ::::::::::::::
```