

Caso di Studio: un Timer

Corrado Santoro

Dipartimento di Matematica e Informatica
santoro@dmi.unict.it

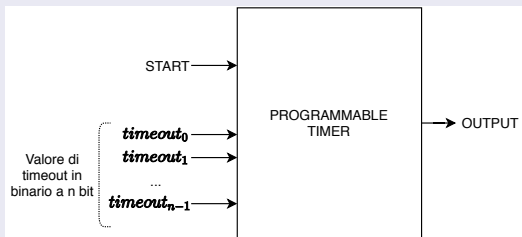


Corso di Architettura degli Elaboratori

Timer: Specifica del problema

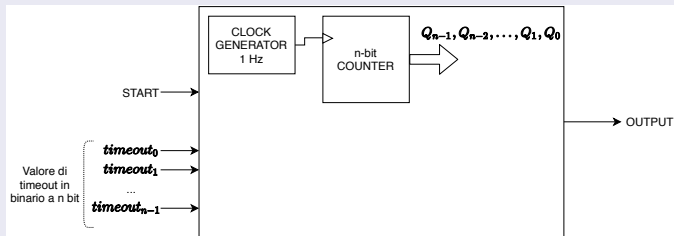
- Desideriamo realizzare un tipico “interruttore temporizzato” attraverso un circuito logico
- Esso dovrà produrre in output 1 alla pressione di un pulsante
- Dopo un periodo di tempo **programmabile**, dovrà produrre in in output 0
- Se, durante tale periodo, viene premuto nuovamente il pulsante, il timer dovrà ripartire

Interfaccia Componente



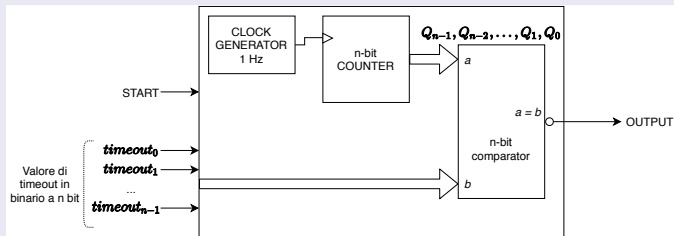
- Possiamo pensare a un componente con:
- Un ingresso **START** che permette di “armare” il timer
- N ingressi di **timeout** che permettono di specificare il valore numerico del tempo a N bit
- L'**OUTPUT**

Consapevolezza del Tempo



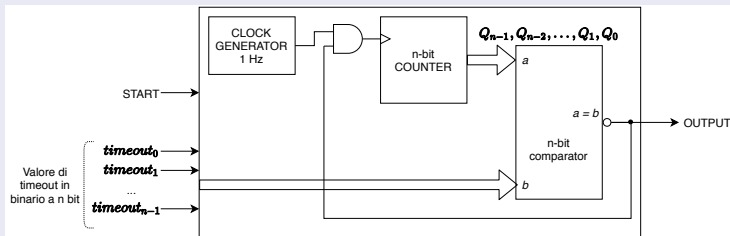
- Abbiamo la necessità di implementare qualcosa che rappresenti lo “scorrere del tempo”
- Includiamo un circuito **generatore di clock** in grado di generare continuamente degli impulsi la cui durata equivale alla risoluzione di tempo che desideriamo (esempio 1 sec)
- Colleghiamo il clock ad un **contatore binario a N bit** che ci permetterà di “contare il tempo”

Raggiungimento del Timeout



- Per implementare un'azione al raggiungimento del timeout, includiamo un circuito **comparatore a N bit**
- Colleghiamo al comparatore l'uscita del contatore (**a**) e gli ingressi di timeout (**b**)
- Il circuito si comporterà nel modo:
 $a \neq b$ Output $\rightarrow$ 1
 $a = b$ Output $\rightarrow$ 0

Stop del Conteggio

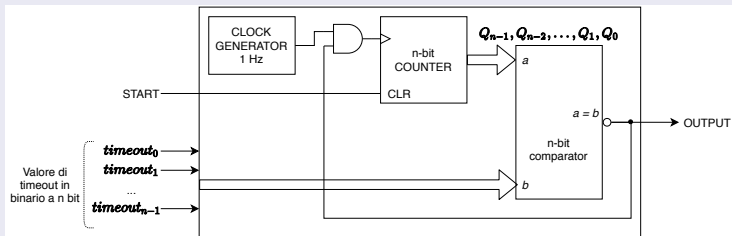


- Quando il timeout è stato raggiunto dobbiamo **fermare il contatore**
- Possiamo dunque utilizzare una porta AND sul clock pilotata dall'uscita del comparatore:

$a \neq b$ Output $\rightarrow$ 1 Il clock non è bloccato

$a = b$ Output $\rightarrow$ 0 Il clock è **bloccato**

Avviamento del Conteggio



- Per avviare il conteggio è sufficiente **resettare il contatore**
- Utilizziamo un contatore con ingresso di **CLR** che collegheremo al nostro ingresso di **START**
- Non appena **START = 1**, il contatore si resetta, il comparatore genera un'uscita pari a 1 e il clock può fluire all'interno del contatore

Implementazione del comparatore a n bit

Comparatore a n bit

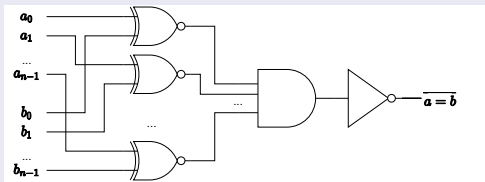
- Per quanto si possa procedere nel modo “classico”, ossia attraverso tabelle della verità e mappe di Karnaugh, in questo caso è più opportuno usare un approccio diverso
- Pensiamo al comparatore a n come a un “AND” di n comparatori a 1 bit con uscita pari a 1 quando $a_i = b_i$:

a_i	b_i	$a_i = b_i$
0	0	1
0	1	0
1	0	0
1	1	1

$\overline{a_i \oplus b_i}$

- La tabella corrisponde ad un EX-NOR

Comparatore a n bit



- Mettiamo in “AND” tutte le uscite delle EX-NOR
- Aggiungiamo un “NOT” finale in modo da far sì che l’uscita sia:

1 se $a \neq b$

0 se $a = b$

Caso di Studio: un Timer

Corrado Santoro

Dipartimento di Matematica e Informatica
santoro@dmi.unict.it



Corso di Architettura degli Elaboratori