



## Esercitazione 10: Elettronica digitale sequenziale

Brognoli Roberto, Garbi Luca, Libardi Gabriele

*Issue:* 1

*Date:* September 11, 2020

University of Trento  
Department of Physics  
Via Sommarive 14, 38123  
Povo (TN), Italy

---

## Introduzione

L'obiettivo di questa esperienza è quello di realizzare e studiare alcuni circuiti digitali basati su flip-flop (FF), come vari divisori di frequenza e contatori. Nell'ambito dell'elettronica digitale sequenziale viene implementato un registro a scorrimento e un automa a stati finiti che simula il comportamento di un semaforo.

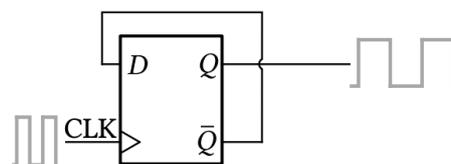
### 1 Setup

#### 1.1 Materiali

- Generatore di funzione d'onda *Rigol DG1022* (impedenza nominale in uscita  $R_s = 50 \Omega$ );
- multimetro digitale (DMM) *Agilent 34410A*;
- oscilloscopio *Agilent DSOX2002A*;
- cavi coassiali BNC-banana e BNC-BNC;
- resistori e capacitori vari;
- 1 integrato *74xx00*, 1 *74xx86*, 2 *74xx74* e 1 *74xx175*;
- 3 LED: verde, giallo, rosso e 1 display numerico.

#### 1.2 Procedure di misura

Tutti gli integrati *74xx00* - *74xx86* - *74xx74* - *74xx175* vengono alimentati con una tensione di  $5V$  ai rispettivi pin, tramite il supporto di un condensatore di  $100nF$ , che funge da stabilizzatore. Nella prima parte il circuito utilizzato è il seguente (Fig.1), dove come CLK si utilizza un'onda quadra TTL di frequenza  $f_{CLK} = 1 \text{ kHz}$ . Successivamente fino a 4



**Figure 1:** Divisore di frequenza per 2.

di questi elementi circuitali vengono utilizzati in cascata come spiegato nella sezione di descrizione dei risultati.

Nella seconda parte viene utilizzato il circuito seguente (Fig.2) come registro a scorrimento a 4 bit. In questo caso per il CLK si utilizza un'onda quadra TTL di frequenza  $f_{CLK} = 1 \text{ Hz}$ . Si connettono inoltre le uscite del registro ai 4 least-significant-bits del display numerico, opportunamente alimentato.

L'implementazione del circuito della sezione di automi a stati finiti viene infine descritta nella parte di descrizione dei risultati corrispondente.

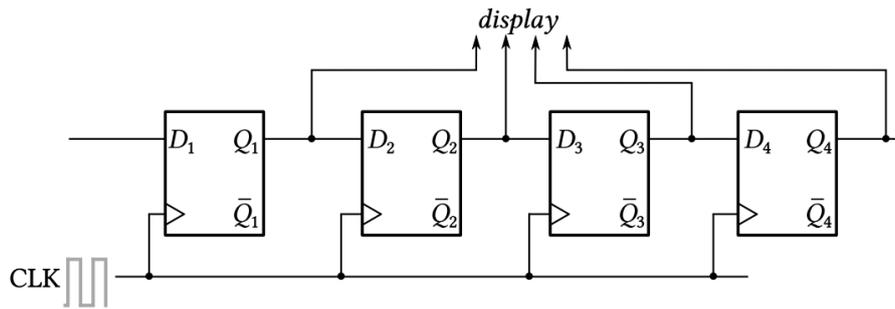


Figure 2: Registro a scorrimento a 4 bit.

## 2 Descrizione e discussione dei risultati

L'analisi dei dati è suddivisa in tre sotto sezioni: nella prima vengono esposti i risultati ottenuti implementando degli integrati *flip-flop* come divisori di frequenza, nella seconda è presentato un registro a scorrimento e nell'ultima viene implementato un automa a stati finiti come semaforo.

Non viene riportata una sezione per la conclusione in quanto i risultati vengono già esaminati e commentati in questa sezione.

### 2.1 Divisore di frequenza e contatore

In questa parte di esperienza viene utilizzato il circuito in figura (1). Se inizialmente  $CLK=0$  e  $Q = 0$ , avremo  $\bar{Q} = D = 1$ . Nel momento in cui  $CLK = 1$ , dopo un ritardo causato dalla presenza stessa del FF, si avrà  $Q = 1$  e  $\bar{Q} = D = 0$ . Successivamente lo stato di  $Q$  e  $\bar{Q}$  viene mantenuto quando ritorna  $CLK = 0$ , da qui rinizia il ciclo. In questo modo l'uscita presa su  $Q$  è un segnale TTL con periodo doppio rispetto a quello in ingresso ( $CLK$ ), ossia con frequenza dimezzata. Nel grafico (3) sono riportati il segnale  $V_{in}$  e il segnale con frequenza dimezzata, preso su  $Q$ . In figura (4) invece è presente l'uscita di una porta logica XOR al cui ingresso ci sono il  $CLOCK$  e  $Q$ . Osservando il picco che si

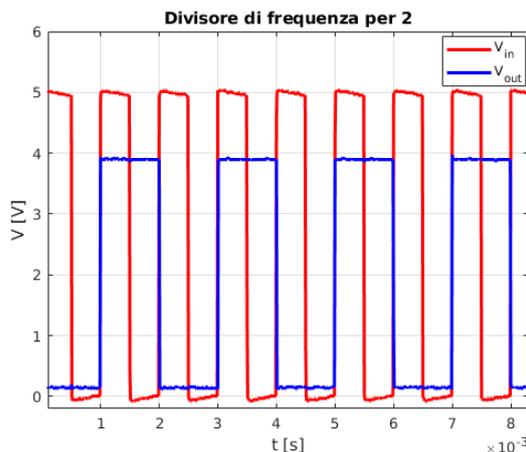


Figure 3: Divisore di frequenza per 2.

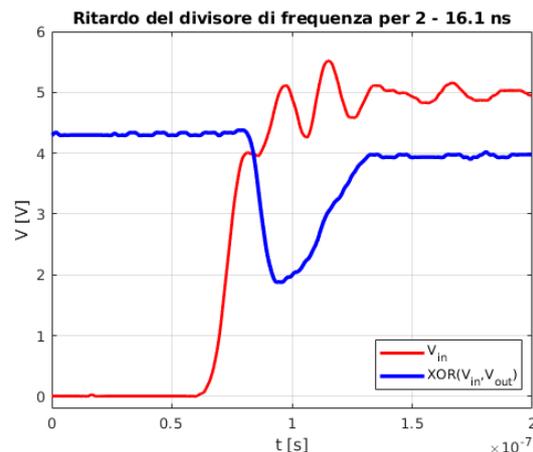


Figure 4: Porta XOR su CLK e Q.

crea riusciamo ad apprezzare meglio il ritardo del segnale in uscita rispetto ai fronti del segnale in entrata. Questo ritardo, causato dalla presenza stessa del *flip-flop* e misurato con i cursori dell'oscilloscopio, risulta  $\Delta t = (16.1 \pm 0.5) \text{ ns}$ .

Collegando l'uscita  $Q$  del primo *flip-flop* al  $CLOCK$  di un secondo, abbiamo come segnale in uscita da  $Q_2$  un segnale con frequenza dimezzata rispetto al segnale in uscita dal primo ( $Q$ ),

ovvero divisa per 4 rispetto a  $V_{in}$ . Realizziamo allora in questo modo divisori di frequenza per 4, 8 e 16 collegando rispettivamente 2, 3 e 4 *flip-flop*. Per fare questo ci serviamo di un ulteriore integrato 74xx74. A titolo di esempio i risultati del divisore per 4 e 16 sono riportati nella coppia di grafici seguente (5 e 6). Si può apprezzare come effettivamente

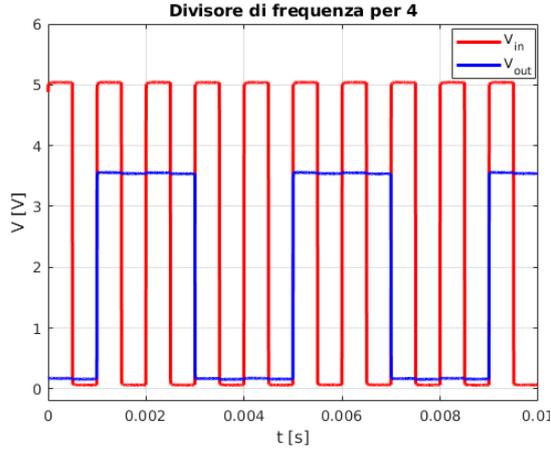


Figure 5: Divisore di frequenza per 4.

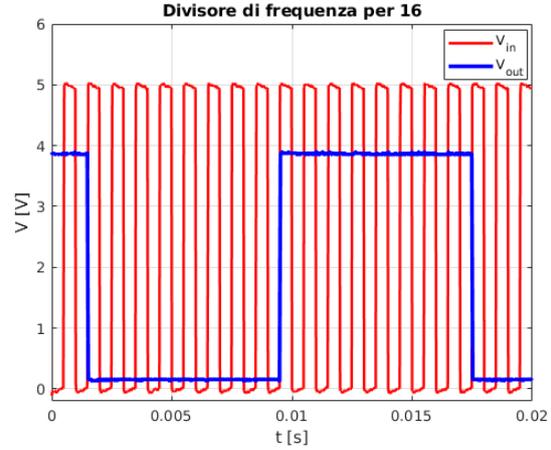


Figure 6: Divisore di frequenza per 16.

la frequenza del segnale in uscita sia rispettivamente 250 Hz e 62.5 Hz, ovvero un quarto e un sedicesimo di  $f_{CLK} = 1$  kHz. Anche per questi divisori sono stati misurati i ritardi, che sono riportati nella tabella seguente. L'uscita della porta XOR applicata al *CLOCK*

Divisore di $f$	$\Delta t$ [ns]
4	$30.3 \pm 0.5$
8	$41.8 \pm 0.5$
16	$59.1 \pm 0.5$

iniziale e all'uscita della cascata di *flip-flop* per i divisori a 8 e 16 è riportata di seguito (Fig.7 e 8).

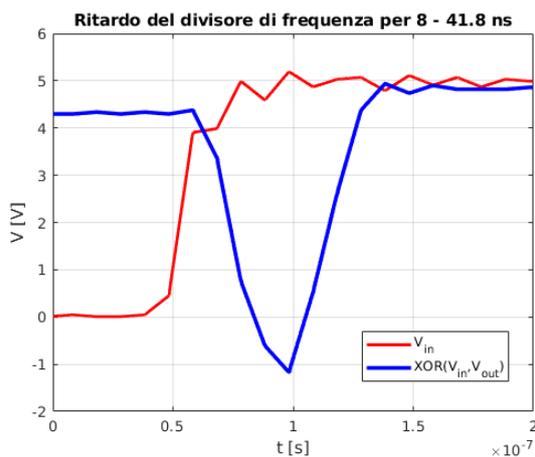


Figure 7: Porta XOR su CLK e  $Q_3$ .

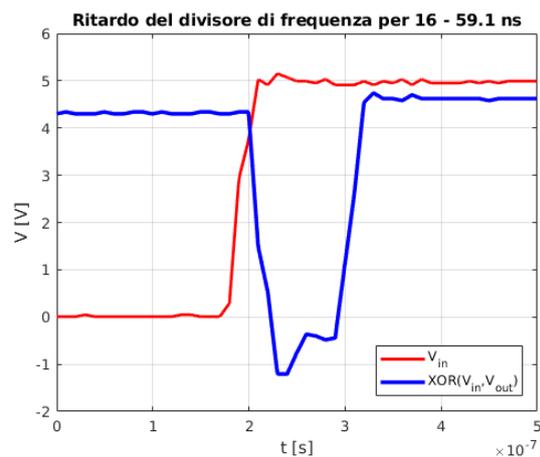


Figure 8: Porta XOR su CLK e  $Q_4$ .

Notiamo che, come ci aspettiamo, il ritardo è cumulativo e nonostante non sembri essere un tempo uguale per ogni porta, in media risulta di 15 ns per ogni *flip-flop* aggiunto. Per quanto riguarda la sincronia di questi 3 divisori osserviamo che poiché i *flip-flop* utilizzati non si servono del medesimo *CLOCK*, ma il *CLK* del FF n-esimo è dato da  $Q_{n-1}$ , i

circuiti sono tutti asincroni. Inoltre l'asincronia si manifesta anche con il fatto che il ritardo del segnale in uscita rispetto al segnale in ingresso cresce con l'aumentare del numero di *flip-flop* utilizzati.

Successivamente si connettono le uscite  $Q_n$  del divisore di frequenza ai 4 *least-significant-bits* del display numerico opportunamente alimentato che mostra la sequenza di numeri esadecimali decrescenti F, E,... 1, 0, uno al secondo, per poi ripetere il conteggio. Questo fenomeno si spiega sapendo che la cifra binaria n-esima varia con una frequenza pari a  $f_{CLK}/2n$ , ovvero quella data dai  $Q_n$  rispettivi.

Infine viene invertita la direzione di conteggio collegando al display le uscite negate  $\bar{Q}$ ,  $\bar{Q}_2$ ,  $\bar{Q}_3$ ,  $\bar{Q}_4$ . Passando un segnale opposto al precedente, ma con la medesima periodicità il conteggio avviene in modo crescente, da 0 a F. Infatti rispetto al caso precedente tutti i numeri binari passati al display sono negati.

## 2.2 Registro a scorrimento

In questa parte di esperienza viene utilizzato il circuito in figura (2), che rappresenta un registro a scorrimento a 4 bit. L'entrata  $D_n$  del *flip-flop* n-esimo è data dall'uscita  $Q_{n-1}$  del FF (n-1)-esimo. Inoltre le porte sono tutte connesse in modo da avere lo stesso *CLOCK*, il registro è dunque sincrono. Utilizziamo come *CLK* un segnale TTL ad 1 Hz e colleghiamo come nella sezione precedente le uscite dei FF ai 4 LSB del display. Connettiamo con un filo  $D_1$  a 5 V, partendo inizialmente con tutte le uscite a 0, dopo un periodo di *CLK*, otteniamo  $Q_1 = 1$  e pertanto il LSB passa da 0 ad 1. Ad ogni *CLOCK* successivo, il bit seguente passa da 0 ad 1, mentre quelli già ad 1 mantengono il loro stato. Questo succede finché tutte e quattro i bit non diventano 1.

Se invece colleghiamo l'uscita  $Q_4$  all'entrata  $D_1$  otteniamo un registro a scorrimento ad anello chiuso. Ovvero se colleghiamo  $D_1$  a 5 V per una durata inferiore ad un periodo di *CLOCK* il LSB passerà da 0 a 1, mentre al secondo periodo questo bit tornerà a 0 e sarà il secondo bit meno significativo a passare da 0 a 1. Questo avviene per tutti i bit, tornando eventualmente allo stato iniziale. Questo comportamento avviene per qualsiasi tipo di configurazione iniziale di bit.

Collegando invece l'uscita negata  $\bar{Q}_4$  a  $D_1$  verifichiamo di avere un registro a scorrimento ad anello chiuso con entrata negata. Quello che accade è che, al contrario del caso precedente, una volta che abbiamo il secondo segnale di *CLOCK* il LSB non torna da 0 a 1 ma mantiene il suo stato iniziale e così via per i bit successivi ad ogni periodo di *CLOCK*. Una volta che tutti i bits sono a 1 il ciclo rinizia portandoli a 0 a partire da quello meno significativo.

## 2.3 Automi a stati finti

Quando è stato costruito un divisore in frequenza con i flip-flop RS, si stava operando con 4 stati, dove ad ogni *CLOCK* il sistema passava da uno a quello successivo. Ad ogni stato vengono preparate le condizioni tali per cui al *CLOCK* successivo il sistema passa alla configurazione preparata nello stato precedente (in altre parole ad ogni *CLOCK* i  $D_i$  diventano i  $Q_i$  successivi). Nel caso in cui si ha a che fare con due flip-flop ( $Q_0, D_0$  e  $Q_1, D_1$ ) i possibili stati sono chiaramente  $2^2$ . Come si è visto nella sezione precedente, il circuito implementato fa sì che ci sia un ciclo ricorsivo in cui l'ultimo stato si lega al primo in modo tale che il ciclo possa ripetersi. Nel caso in cui da questa configurazione si voglia progettare un sistema che lavori con tre soli stati, bisogna trovare il modo di far convogliare uno stato in un altro, per permettere al sistema di entrare in un loop di soli tre stati. Di seguito abbiamo progettato un possibile automa a tre stati (fig: 9), in cui è possibile l'utilizzo di sole porte *NAND*. Questo circuito è stato pensato per fare in modo che lo stato

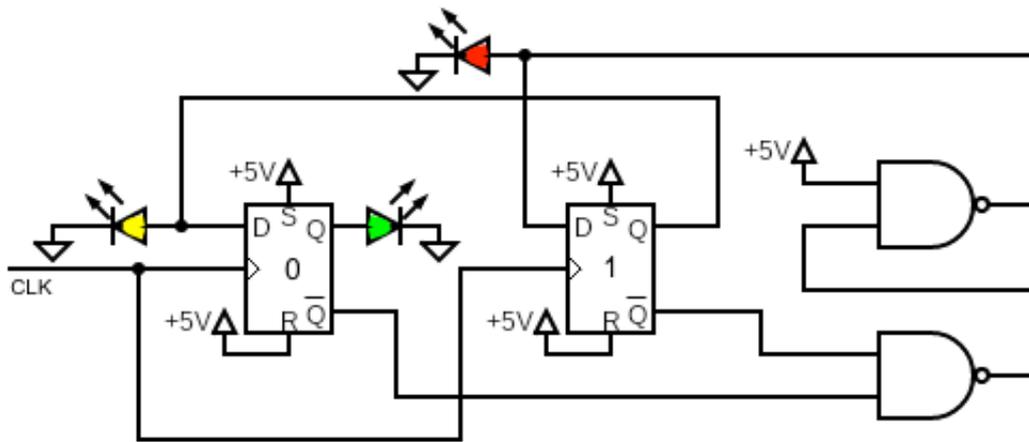


Figure 9: Automa a stati finiti.

dato da  $(Q_1, Q_0) = (1, 1)$  converga nello stato  $(D_1, D_0) = (0, 1)$ . Questo è essenziale nel funzionamento del circuito, infatti nel caso in cui il circuito all'inizio si dovesse trovare in quello stato, sarebbe subito (dopo il primo *CLOCK*) portato allo stato successivo e non verrebbe più ripetuto nel loop. Quindi, una volta che ci siamo assicurati che questo stato converge in uno stato del loop, andiamo a vedere cosa succede agli altri. Consideriamo il caso in cui  $(Q_1, Q_0) = (0, 0)$ , analizzando il circuito si conclude che  $(D_1, D_0) = (1, 0)$ , ed in questo caso l'unico LED che si accende è quello rosso in figura. Dopo un *CLOCK*, siamo nella situazione in cui  $(Q_1, Q_0) = (1, 0)$ , dunque  $(D_1, D_0) = (0, 1)$ , e questa volta da accendersi è il LED giallo. Infine dopo un altro *CLOCK*, gli stati saranno  $(Q_1, Q_0) = (0, 1)$ , e di conseguenza  $(D_1, D_0) = (0, 0)$ , e si accenderà il LED verde. Al *CLOCK* successivo il sistema tornerà nello stato  $(Q_1, Q_0) = (0, 0)$ , e il ciclo continuerà solo tra questi tre stati. Quindi, come già scritto sopra, la cosa essenziale qui, è stata quella di creare un loop a tre stati e di far convergere lo stato che rimane in uno dei tre, in modo da assicurare l'esistenza del ciclo anche nel caso in cui ci sia quello stato all'inizio. Di seguito riportiamo la tabella di verità del circuito:

$Q_1$	$Q_0$	$D_1$	$D_2$
0	0	1	0
1	0	0	1
0	1	0	0
1	1	0	1

Quello realizzato è dunque un'implementazione di un semaforo in cui i tre LED restano tutti accessi rispettivamente per la durata del periodo del *CLOCK*. Se ci poniamo all'uscita di uno dei LED, ci si accorge che quello che abbiamo ottenuto è un divisore in frequenza per 3 sincrono. Infatti basta pensare che mentre il *CLOCK* è ad una certa frequenza, l'uscita che abbiamo scelto sarà accesa solo ogni 3 *CLOCK*, quindi abbiamo di fatto ridotto di 3 volte la frequenza di *CLOCK*. Il circuito costruito è sincrono in quanto i *CLOCK* di entrambi i Flip-Flop sono in comune, dunque gli eventuali ritardi nasceranno dalle porte logiche *NAND* utilizzate. In ogni caso si parla di ritardi trascurabili per il funzionamento del semaforo testato ad  $f_{CLK} = 1$  Hz in laboratorio.