



Esercitazione 09: Porte logiche, tempo di propagazione di un segnale e flip-flop RS

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 implementare l'algebra booleana in elettronica, mediante lo sviluppo di porte logiche tramite transistor. Si andrà poi a verificare il teorema di Morgan, costruendo una porta XOR a partire da sole porte NAND. Successivamente sarà analizzato il tempo di ritardo di un segnale che attraversa delle porte logiche. Infine si studierà il funzionamento del circuito *flip-flop* RS come elemento di memoria.

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 transistor 2N2222;
- 1 integrato 74xx00, 2 integrati 74xx04;
- 3 LED (*Light Emitting Diode*).

1.2 Procedure di misura

Sia gli integrati 74xx00 - 74xx04 che i transistor vengono alimentati con una tensione di 5V ai rispettivi pin, tramite il supporto di un condensatore di 100 nF, che funge da stabilizzatore. La corrente viene limitata a 40 mA in ciascun ramo. I vari circuiti con i relativi schemi sono riportati e discussi nella sezione analisi.

1.3 Compendio di algebra booleana

Riportiamo di seguito le tabelle di verità che descrivono le principali operazioni booleane tra 2 stati (*low* e *high*). Le operazioni logiche e i relativi simboli sono riportati in figura (fig: 2).

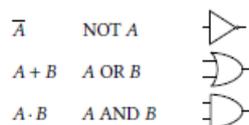


Figure 1: Operazioni booleane.

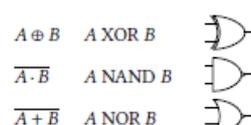


Figure 2: Operazioni derivate booleane.

Tabelle di verità:

| A | B | NOT A | A OR B | A AND B | A XOR B | A NOR B | A NAND B |
|---|---|-------|--------|---------|---------|---------|----------|
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |

2 Descrizione e discussione dei risultati

L'analisi dei dati è suddivisa in quattro sotto sezioni. Nella prima viene implementata la porta logica NOR e ne viene verificato il funzionamento, nella seconda viene fatta la stessa cosa con una porta XOR sviluppata a partire da 4 NAND. La terza sezione presenta la misura del ritardo del segnale acquisito attraversando porte logiche, nella fattispecie viene utilizzata la porta NOT. Infine si verifica il funzionamento di un circuito *flip-flop* RS.

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

2.1 Porte logiche

In elettronica possiamo costruire l'algebra booleana prendendo come stato *low*, l'uscita a $0V$ o *ground*, mentre per lo stato *high*, l'uscita a $5V$.

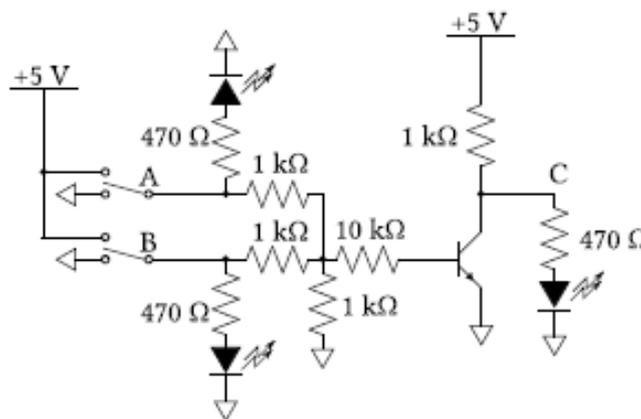


Figure 3: Porta logica NOR realizzata con un transistor.

Il circuito, in figura (fig: 3), implementa la porta logica NOR. Infatti se consideriamo le entrate A e B , entrambe a massa (*stato* 0), non arriva la tensione V_{BE} necessaria per attivare il transistor, quindi è come se avessimo un circuito aperto tra il collettore e l'emettitore. In uscita sul collettore C si avrà lo *stato* 1, e il LED risulterà acceso. Nel caso in cui si avesse lo *stato* 1, per una delle due entrate A o B o per entrambe invece, il transistor viene mandato in saturazione, cortocircuitando così il collettore e l'emettitore. Come risultato si ha che la corrente passa per quella maglia invece che accendere il LED (*stato* 0). Il comportamento appena descritto, se si consultano le tabelle di verità, corrisponde alla porta logica NOR.

2.2 Porta logica XOR

La porta XOR (Exclusive OR) è una porta logica che riceve in ingresso due valori A e B , e restituisce *stato* 1 in uscita se, e solo se, vi è almeno un ingresso che differisce dagli altri. Questa porta logica si può costruire partendo da 4 porte logiche NAND collegate

opportunamente. In figura (fig: 4) è riportata una possibile implementazione della porta logica XOR. Variando gli stati in entrata e consultando le tabelle di verità, notiamo che l'uscita corrisponde esattamente a quanto richiesto. Utilizzando questo schema è stato

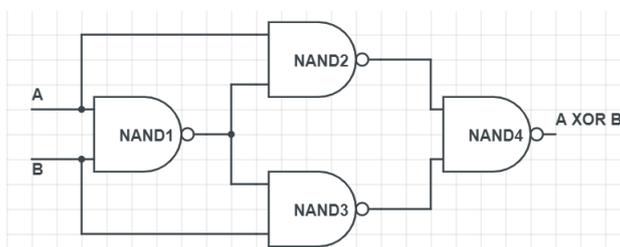


Figure 4: Porta logica XOR formata da 4 NAND.

costruita una porta logica XOR funzionante grazie all'integrato 74xx00.

2.3 Misura del tempo di propagazione di un segnale attraverso porte logiche

Collegando in serie n porte logiche NOT si è calcolato il ritardo tra l'onda in entrata (onda quadra 0 – 5 V) e l'onda in uscita in funzione di n . Rispetto alla teoria, va considerato infatti che le porte logiche in elettronica sono composte da transistor, resistenze e capacità (soprattutto parassite). Queste ultime sono le maggiori responsabili del ritardo del segnale in uscita rispetto a quello in entrata. Sicuramente la dimensione del componente elettronico 74XX04 ha la sua rilevanza in materia, infatti maggiore è l'estensione spaziale del componente, maggiori sono le capacità parassite e dunque anche i ritardi. Per misurare il ritardo tra il segnale in entrata e in uscita sono state acquisite le forme d'onda dell'oscilloscopio del segnale all'inizio della catena di NOT e quello alla fine del n -esimo NOT. In prossimità del cambio di stato tra 0 e 1, si osserva che il segnale in uscita è ritardato rispetto a quello in entrata. Per fare una misura di questo ritardo abbiamo considerato il valore a metà tra 0 – 5 V, dei due segnali, e abbiamo poi fatto la differenza tra i due tempi relativi. Infine abbiamo ripetuto questa procedura per $n = 1, \dots, 12$. Per calcolare il ritardo medio per porta, abbiamo fatto una regressione lineare per i ritardi di tempo in funzione del numero di porte logiche NOT utilizzate. Il grafico con i dati e la rispettiva regressione lineare sono riportati in figura (fig 5).

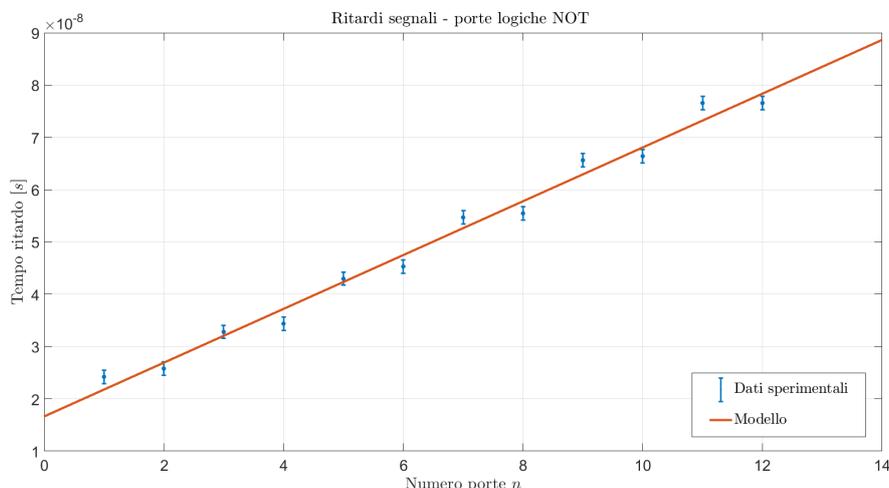


Figure 5: Dati ritardi tempi in funzione del numero di porte NOT.

Dal grafico possiamo notare un comportamento sistematico tra i ritardi misurati per gli n pari e per gli n dispari. Il motivo di ciò sta nella differenza del segnale in uscita nel caso di un utilizzo di un numero di porte pari o dispari. Infatti nel caso di un n pari (fig: 6), il segnale in uscita è uguale a quello in entrata (a meno di dissipazioni), in virtù del fatto che una doppia negazione è una affermazione. Nel caso di n dispari (fig: 7) invece il segnale in uscita è opposto a quello in entrata, a causa della negazione. Per questo motivo, la misura del ritardo, eseguita come spiegato precedentemente, porta degli errori sistematici sia nel caso di n pari che dispari, e questo spiega l'andamento 'a gradini' che si osserva nel grafico.

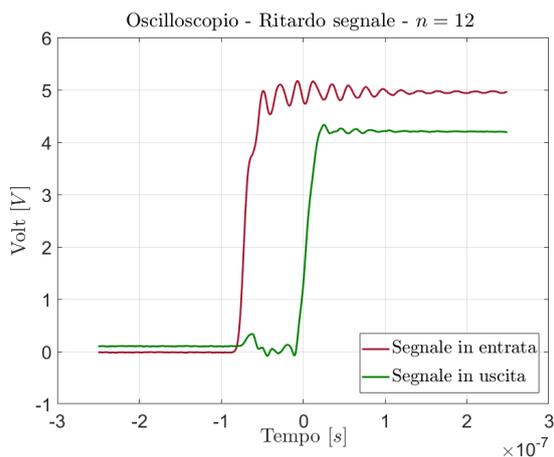


Figure 6: Ritardo segnale caso $n = 12$ (pari).

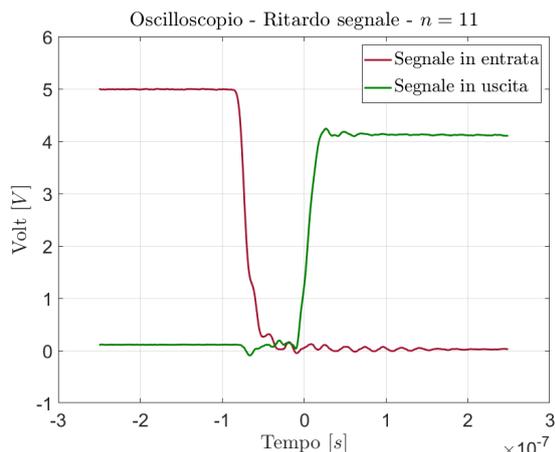


Figure 7: Ritardo segnale caso $n = 11$ (dispari).

Dalla regressione lineare otteniamo un coefficiente angolare di $m = (5.14 \pm 0.11) \times 10^{-9} s/n$, che rappresenta il ritardo medio per porta, e un'intercetta di $b = (1.66 \pm 0.08) \times 10^{-8} s$. Quest'ultimo risultato non è compatibile con zero, perché va considerata la presenza di fili coassiali di collegamento tra il circuito, il generatore e l'oscilloscopio. Infatti questi presentano una capacità non nulla, che causa il ritardo 'di offset', che si può misurare anche se non vi è collegata nessuna porta logica. Il χ^2 ridotto della regressione lineare risulta $\chi_r^2 = 3.44$, e una possibile spiegazione di questo valore relativamente elevato la si può dare tenendo presente quanto già accennato riguardo al caso di n pari e dispari.

Nel caso in cui si collegano un numero n dispari di porte logiche NOT in modo da formare un loop, ovvero collegando l'output dell'ultima porta all'input della prima, si genera un'oscillazione. Se si usasse un numero pari di porte non si avrebbe nessuna oscillazione, infatti se si nega un numero pari di volte un segnale, si torna ad ottenere lo stesso segnale originale non negato, così in un loop pari di NOT, a fine ciclo si torna ad ottenere lo stesso segnale di partenza e per questo non si ha alcuna oscillazione. Nel caso di n dispari invece, a fine di ogni giro, all'entrata del loop, si avrà il segnale negato di quello precedente, e questo causa il propagarsi di un'oscillazione, permessa grazie al fatto che i chip sono alimentati esternamente. Non ci si aspetta che le oscillazioni siano perfettamente sinusoidali, però ci aspetta che il tempo che intercorre tra un massimo ed un minimo sia il tempo di ritardo che impiega il segnale a fare un loop, e quindi a cambiare stato (rispettivamente nel punto in cui è stata posta la sonda dell'oscilloscopio). Quindi si può concludere che il periodo dell'oscillazione, cioè la differenza di tempo tra due massimi, è uguale al ritardo che il segnale subisce, percorrendo $2n$ porte logiche NOT. Poiché il loop viene chiuso tra la porta iniziale e finale, non va considerato il contributo dato dai cavi coassiali del generatore, quindi verrà tenuto conto solo del coefficiente angolare calcolato sopra, e non l'intercetta. Sotto (fig: 8) è riportata l'immagine dell'oscilloscopio per l'oscillazione che

abbiamo ottenuto usando $n = 5$.

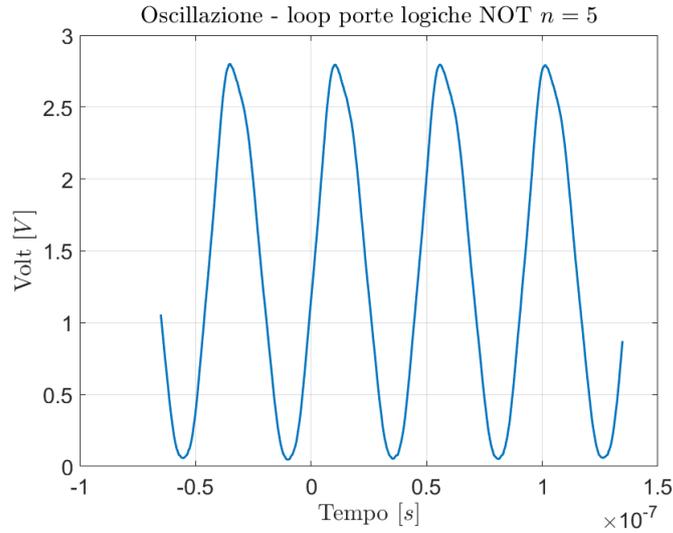


Figure 8: Oscillazione loop $n = 5$ porte logiche NOT.

Di seguito sono riportati i periodi di oscillazione e i rispettivi valori aspettati dati dalla moltiplicazione tra il coefficiente angolare e $2n$.

| Numero porte n | Periodo T [ns] | Valore aspettato [ns] |
|------------------|-----------------|-----------------------|
| 5 | 45.4 ± 0.2 | 51.4 ± 1.1 |
| 7 | 88.8 ± 3.2 | 71.96 ± 1.54 |
| 9 | 99.8 ± 0.4 | 92.52 ± 1.98 |
| 11 | 114.8 ± 3.2 | 113.08 ± 2.42 |

I due risultati appaiono in prima analisi non compatibili tra loro, infatti abbiamo trascurato la presenza del cavo coassiale che collega il circuito all'oscilloscopio. Il contributo di questo infatti inizia a pesare quando si tratta di scala di tempi del ns . Il valore trovato precedentemente del coefficiente angolare è inoltre approssimativo, infatti la modalità con cui è stato calcolato non è priva di errori sistematici, come abbiamo già osservato. Per concludere osserviamo che il ritardo di questi componenti elettronici non è così trascurabile quando si considerano frequenze alte dell'ordine dei GHz (nei computer). Ad un certo punto è necessaria quindi un'importante riduzione delle dimensioni dei componenti, per ridurre il più possibile le capacità parassite.

2.4 Circuito Flip-Flop RS

In figura (fig: 9) è riportata una possibile implementazione del circuito flip-flop RS, utilizzando porte logiche NAND.

Questo circuito è un elemento di memoria, in grado di mantenere il proprio stato indefinitamente nel tempo. I due ingressi \bar{S} e \bar{R} , sono rispettivamente detti *set* e *reset*. Nel caso in figura queste due entrate sono sempre collegate a $5V$ tramite una resistenza, quindi nel caso di S e R flottanti, gli stati di *default* sono *high*, pertanto si dice che la configurazione è *pull-up*. Per studiare la tabella di verità del circuito assumiamo inizialmente di non avere una configurazione nè *pull-up*, nè *pull-down*. Nel caso in cui lo stato di \bar{S} è *low*, mentre quello di \bar{R} è *high*, avremo sicuramente lo stato di Q *high* (infatti se una porta NAND ha come ingresso un *low*, restituisce sempre *high*), mentre il complementare \bar{Q} sarà sicuramente *low*. Nel caso in cui \bar{S} è *high* e \bar{R} è *low*, per le stesse ragioni si avrà \bar{Q} *high* e

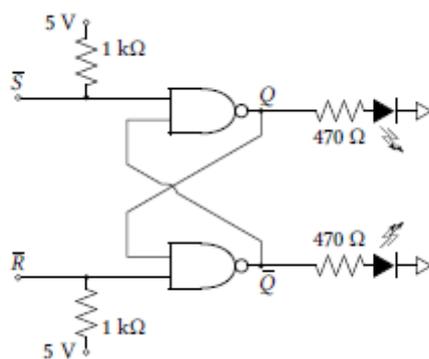


Figure 9: Circuito flip-flop RS con porte logiche NAND.

di conseguenza Q *low*. Nel caso in cui entrambi \bar{R} e \bar{S} sono *high*, quello che succede non è nessun cambio di *stato*, infatti l'ingresso *high* in una porta NAND non porta a nessuna certezza, a meno che non si conosca l'altra entrata. Per questo motivo la situazione che si ottiene è quella di temporanea memoria dello stato immediatamente precedente a quello in cui le entrate sono entrambe *high*. Il caso in cui entrambe le entrate sono *low*, non va considerato, perché si avrebbe una situazione in cui sia Q che \bar{Q} sono *high*, quindi non ha alcun interesse analizzarlo. Nonostante ciò va però considerato questo caso, e va anche evitato, in quanto potrebbe portare problemi nel prevedere lo stato del circuito appena questo viene acceso. Per evitare tutto ciò si opera con la configurazione *pull-up* in figura, in modo che nel caso di fili flottanti, comunque si mantiene lo *stato di memoria* (\bar{R} e \bar{S} *high*). Per riassumere riportiamo sotto la tabella di verità:

| \bar{S} | \bar{R} | \bar{Q} |
|-----------|-----------|-------------|
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | memory |
| 0 | 0 | not allowed |