

Scheda di laboratorio n.08

Dall'analogico al digitale

1 Introduzione all'elettronica digitale

Come visto nelle scorse esperienze di laboratorio, l'elettronica analogica permette di amplificare, filtrare, trattare segnali *continui* come per esempio la fotocorrente generata da un rivelatore di luce o un segnale radio captato da una antenna. Diversamente, i circuiti digitali sono disegnati in maniera da segmentare il continuo dei voltaggi (che è sempre possibile avere, come in un qualsiasi circuito) su un insieme di *livelli discreti* corrispondenti a specifiche bande di voltaggio ammesse. Nel caso più tipico i livelli sono due e vengono convenzionalmente chiamati “0” e “1” oppure, con un ovvio richiamo alla logica binaria *falso/vero* dell'algebra di Boole.

Il vantaggio chiave offerto da questo approccio è che un segnale digitale è largamente immune al rumore e rende possibile implementare varie tecniche per mandare quasi a zero la probabilità di perdere informazione nel trattamento e/o trasmissione di un segnale¹. Questo, unito al fatto che l'interdipendenza fra le diverse parti di un circuito digitale può essere minimizzata in maniera più semplice e quindi il disegno di complessi circuiti digitali risulta molto più facile da standardizzare², ha determinato l'indiscusso successo dell'approccio digitale rispetto a quello analogico in una vasta gamma di applicazioni. Lo scopo delle prossime esperienze sarà di prendere confidenza con il funzionamento dei circuiti elettronici digitali e per farlo ci porremo come obiettivo finale **la realizzazione e lo studio di un circuito in grado di generare una sequenza di bit “pseudocasuale”**, come illustrato in Fig.1.1. Il motivo per il prefisso “pseudo” è che sebbene la sequenza dei bit ha molte delle caratteristiche di un segnale binario stocastico, di fatto la sua evoluzione è perfettamente deterministica.

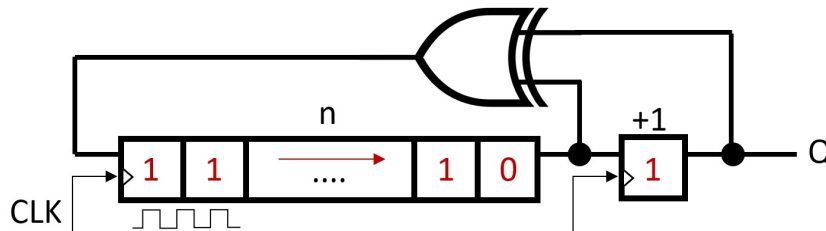


Figura 1.1: Generatore di numeri pseudocasuali realizzato con un registro di shift con feedback lineare.

Come vedremo a breve, la costruzione del circuito ci permetterà di toccare molti degli aspetti fondamentali dell'elettronica digitale quali: lo studio della risposta “analogica” di una singola porta logica elementare; la realizzazione di un circuito che implementi una data tabella di verità, usando delle porte universali NAND; la realizzazione e l'utilizzo di circuiti di logica sequenziale, in cui la presenza di uno o più feedback permette di avere degli effetti di memoria nel circuito che possono portare a bistabilità, instabilità e altro.

!!! ATTENZIONE !!! Dato che la complessità dei circuiti tenderà ad aumentare, con diversi integrati con una miriade di connessioni di configurazione, sarà (1) **obbligatorio usare i rail per le alimentazioni** e (2) **calda-mente consigliato pianificare** accuratamente il montaggio. La pulizia e razionalità del circuito saranno un **elemento primario per valutare l'esecuzione dell'esperienza**.

¹ Senza questa immunità al rumore, cose oggi banali come fare una copia esatta di un file di 1 GB sarebbe quasi impossibile!

² Il processore WSE2 disegnato dalla *Cerebras* e fabbricato nel 2021 dalla Taiwan Semiconductor Manufacturing Company (TSMC), con un così detto processo a 7 nm, contiene ben 2600 miliardi di transistor. Se ogni transistor richiedesse anche solo 1 secondo di attenzione da parte dei designer servirebbero circa 80 mila anni per dare una “occhiata” a questo processore. È evidente che macchine del genere possono solo esistere grazie ad un'alta standardizzazione, che è praticamente impossibile nei circuiti analogici.

2 Logica combinatoria

Iniziamo con qualche esempio di logica combinatoria, in cui diverse porte logiche vengono concatenate senza alcuna retroazione. In questo caso, le uscite del circuito sono completamente determinate dagli ingressi e si possono descrivere con una **tabella di verità**.

Durante tutta l'esperienza, quantomeno per quel che riguarda le porte elementari, ci baseremo sull'integrato CD4011 che contiene 4 porte NAND di tipo CMOS. Prima di passare a questa classe di integrati e dimenticarcelo completamente, è utile ricordare che le porte digitali sono semplicemente dei circuiti analogici disegnati per tendere a "proiettare" i voltaggi su degli intervalli standard corrispondenti ai livelli digitali. Questo significa che *a volte* anche in un circuito digitale è utile ricordare che i voltaggi possono essere delle quantità continue. Questo è utile sia perché viene sfruttato in specifici circuiti (come per esempio i multivibratori che vedremo in seguito), sia perché fa capire meglio i limiti di operazione dei circuiti logici.

2.1 Costruzione di una porta NOT

Il circuito che realizzeremo è illustrato in Fig.2.1: una porta NOT realizzata con un transistor MOSFET Q_1 ad arricchimento a canale n -type, o NMOS. Il funzionamento è banale: quando il segnale di ingresso in A, ossia il voltaggio di gate V_{GS} , è vicino a zero Q_1 è sostanzialmente isolante e il voltaggio di uscita sarà portato in alto dalla resistenza R_1 ; viceversa, se V_{GS} viene portato oltre la *threshold* Q_1 diventerà molto conduttivo e sostanzialmente cortocircuiterà l'uscita a terra. Entro certi limiti, il MOSFET Q_1 qui si comporta sostanzialmente come un interruttore controllato da V_{GS} . Questo schema non è molto diverso da una porta NOT "vera" costruita con tecnologia CMOS, dove semplicemente al posto di R_1 viene usato un transistor PMOS che diventa isolante o conduttivo in maniera "complementare" a Q_1 . Questo comporta vari vantaggi³ ma concettualmente non cambia la sostanza del nostro primo esperimento.

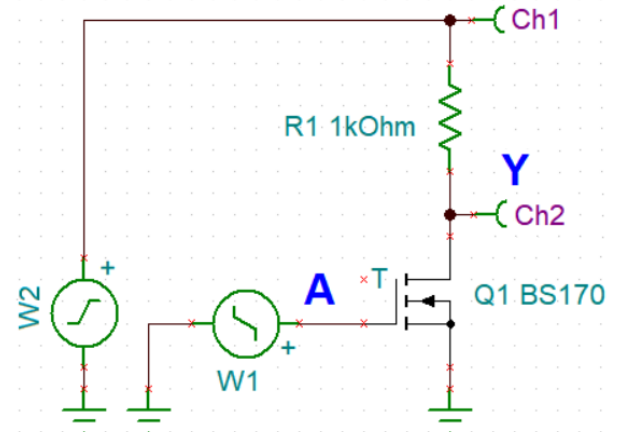


Figura 2.1: Porta NOT con tecnologia NMOS.

Task 1 Gli esperimenti verranno realizzati usando uno fra due MOSFET simili, con numero di parte BS170 e BS107. Annotare le caratteristiche del componente, in particolare la piedinatura e la *threshold* o *pinch-off* V_P .

In linea di principio il circuito di Fig.2.1 può operare con diversi voltaggi di alimentazione V_{DD} , anche se esistono chiaramente delle limitazioni date dalla *threshold*⁴ del MOSFET. Nel nostro caso useremo $V_{DD} = 5V$, che verrà impostato usando l'uscita W2 della scheda Analog Discovery 2, connessa al resistore R_1 . Per documentare sia il voltaggio di uscita che la corrente che scorre in Q_1 acquisiremo i due voltaggi indicati con gli ingressi Ch1 e Ch2. Come già fatto, assumendo che gli input analogici assorbano una corrente trascurabile, le caratteristiche di Q_1 si possono calcolare semplicemente come

$$V_{DS} = V_{Ch2} \quad (2.1)$$

$$I_D = (V_{Ch1} - V_{Ch2}) / R_1, \quad (2.2)$$

che possono essere usate per costruire un grafico parametrico di $I_D(V_{DS})$ per diversi V_{GS} . Come nel caso del LED considerate che è possibile usare Ch1 in modalità differenziale piuttosto che riferita a terra, e misurare direttamente la caduta di potenziale sulla resistenza.

Task 2 Collegare Ch1 e Ch2 per controllare V_{DD} e V_{GS} , rispettivamente. Verificare la corrente massima attesa quando $V_{DD} = 5V$ e se la scheda sia in grado di erogarla. Impostare W2 a 5V e poi misurare la curva risposta del circuito $V_{DS}(V_{GS})$ usando `sweepbias.vi` per variare W1 sull'intervallo che va da 0 a 5V. Quantificare dei possibili valori di V_{OH} , V_{OL} , V_{IH} e V_{IL} sulla base della pendenza della curva di risposta (trascurare la possibile variabilità di V_P da un dispositivo all'altro, che ovviamente è cruciale nella definizione di questi voltaggio e del margine di rumore.).

³In particolare, a voltaggi V_{GS} bassi l'uscita sarà connessa a V_{DD} con una impedenza $\ll R_1$; inoltre, mentre nel nostro circuito quando il MOSFET è conduttivo passa una corrente non trascurabile, il transistor PMOS annulla, o quasi, la dissipazione.

⁴Banalmente, se $V_{DD} < V_P$ il circuito diventa poco utile dato che Q_1 resterà sempre spento.

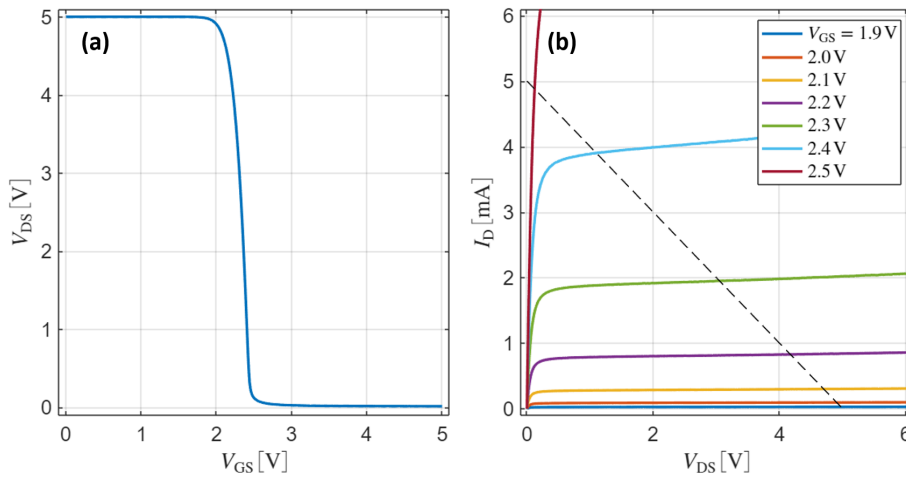


Figura 2.2: Caratteristiche della porta: (a) funzione di risposta; (b) caratteristiche di uscita.

A questo punto ci aspettiamo di osservare qualcosa di simile a quanto riportato in Fig.2.2a. È utile a questo punto riconoscere come questa curva non sia altro che una intersezione delle caratteristiche $I_D(V_{DS}; V_{GS})$ del MOSFET con la retta di carico $V_{DS} = V_{DD} - R_1 I_D$, come illustrato in Fig.2.2b.

Task 3 Ricostruire le caratteristiche di uscita $I_D(V_{DS})$ del MOSFET al variare del parametro di controllo del dispositivo, ossia del voltaggio di gate V_{GS} . Per fare questo scambiate il ruolo di W1 e W2, e quindi usare `sweepbias.vi` per impostare W1 e poi eseguire una spazzata in $W2 = V_{DD}$. Plottare le curve per una sequenza di $W1 = V_{GS}$ che copra la regione in cui il circuito passa da output $\approx 5V$ a $\approx 0V$. Generare un grafico riassuntivo dei risultati dell'intero esperimento con il transistor NMOS e salvarlo in `PortaNOT.pdf`. Si noti che non ci aspetteremo che il modello riportato a lezione, secondo cui la corrente di saturazione segue la legge

$$I_D \propto (V_{GS} - V_P - V_{DS}/2) V_{DS} \quad (2.3)$$

riproduca gli andamenti sperimentali al di là del livello qualitativo. Può essere interessante tentare dei fit ma più che altro per rendersi conto della differenza fra il dispositivo reale e i modelli discussi.

2.2 La porta universale NAND

In questa parte dell'esperienza prenderemo confidenza con l'integrato CD4011 di tipo CMOS, che contiene quattro porte NAND e che useremo ampiamente nelle prossime esperienze. In questa fase continueremo ad usare delle tecniche di controllo e di acquisizione sostanzialmente *analogiche*. Sebbene questo non sia molto standard, ci permetterà di apprezzare meglio il funzionamento dei vari elementi circuitali, almeno durante i nostri primi passi con questa nuova classe di dispositivi.

Il primo circuito da montare è quello in Fig.2.3: una semplice porta NAND con un ingresso (B) collegato alla sorgente analogica e l'altro (A) collegabile a terra o al voltaggio V_{DD} . In questi casi si usa spesso un resistore di *pull-up* R_1 , che è sufficientemente conduttivo a impostare A a V_{DD} , ma che permette all'occorrenza di portarlo a terra quando l'interruttore⁵ viene chiuso. Un vantaggio in questo caso è che l'ingresso A non può mai essere *flottante*, condizione che è meglio evitare con un ingresso ad alta impedenza come nel caso di una porta CMOS.

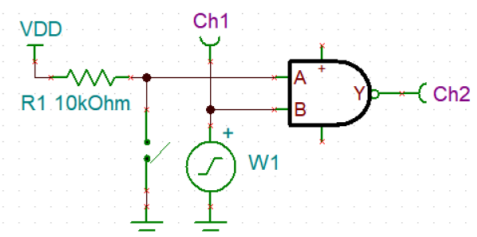


Figura 2.3: Test di porta NAND.

Task 4 Consultare il *datasheet* di CD4011 e riportare la piedinatura nel *logbook*. Verificare che l'integrato sia compatibile con le specifiche di Analog Discovery 2, in particolare il voltaggio di alimentazione.

⁵Nel vostro caso sarà un banale cavetto che potrete connettere al *rail* di GND a vostro piacimento.

Task 5 Verificare rapidamente il comportamento di una delle porte NAND contenute nel chip usando la strategia illustrata in Fig.2.3. Connettere uno degli ingressi a W1 e a Ch1 e l'uscita a Ch2. Non è necessario salvare o plottare le misure, verificate semplicemente che la porta funzioni. Fare i seguenti test:

- lasciare l'interruttore aperto portando l'ingresso A a 5V, verificare che la porta fa una operazione NOT, infatti $Y = \overline{A \cdot B} = \overline{1 \cdot B} = \overline{B}$;
- chiudere l'interruttore portando l'ingresso A a terra, verificando che l'uscita rimane alta a prescindere dal valore dell'ingresso B , infatti $Y = \overline{A \cdot B} = \overline{0 \cdot B} = \overline{0} = 1$;
- prendere nota del voltaggio di soglia a cui la porta cambia stato.

Passiamo a studiare una prima rete logica combinatoria: nel circuito finale avremo bisogno di una porta XOR; sebbene esistano dei chip dedicati che la realizzano, noi la creeremo con delle porte NAND. Ricordiamo infatti che le porte NAND sono universali e in grado di riprodurre qualsiasi funzione booleana, inclusa l'operazione XOR.

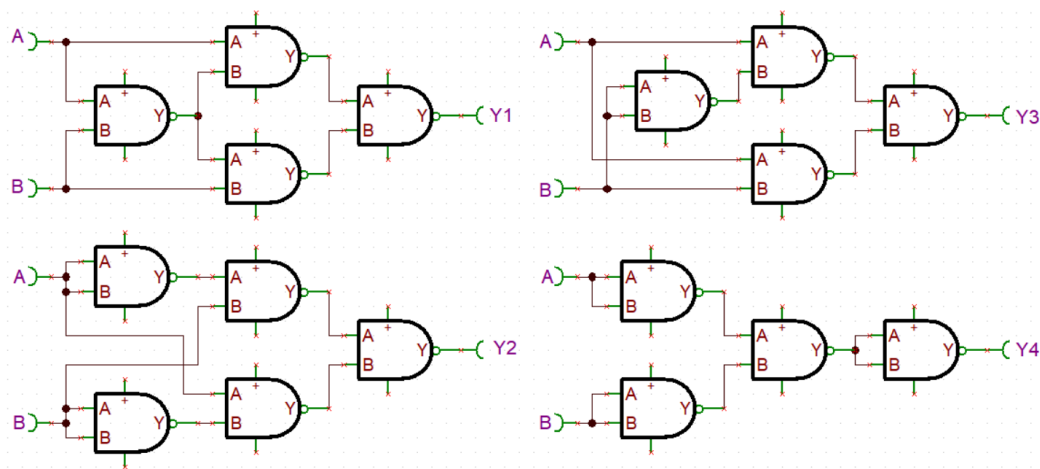


Figura 2.4: Esempi di reti logiche combinatorie costruite con porte NAND.

Task 6 Considerare le reti logiche riportate in figura 2.4. Costruire una tabella di verità contenente A , B , $Y1$, $Y2$, $Y3$ e $Y4$ e verificare se qualcuno dei circuiti riproduce la funzione XOR, ossia $A \oplus B$. Scrivete una formula booleana $Y = \dots$ anche per le eventuali reti che non svolgono la funzione XOR. Verificare sperimentalmente se la rete da voi scelta funziona effettivamente come uno XOR, generando con W1 e W2 due ingressi logici dipendenti dal tempo, e verificando che l'uscita riproduce effettivamente la funzione booleana XOR. Salvare il risultato in un grafico *CircuitoXOR.pdf*. Organizzatevi per tenere il circuito montato per il futuro, dato che sarà utile nell'esperienza successiva.

3 Primi circuiti sequenziali

In un circuito logico sequenziale, il flusso dei segnali digitali contiene anche dei percorsi di retroazione, che possono generare diverse configurazioni stabili e quindi un *effetto memoria*. Di conseguenza, i valori dei segnali di uscita dipendono non solo dal valore degli ingressi ma anche dalla *storia* della rete logica. Come nei circuiti analogici, i feedback possono portare a vari effetti quali oscillazioni o bistabilità. Inizieremo dal caso più semplice possibile dato da un circuito bistabile creato con due porte NAND connesse in un loop di retroazione reciproca, che realizzano un così detto *latch SR*.

3.1 Latch SR

Per questo montaggio useremo un nuovo chip CD4011 e lasceremo la rete XOR montata (servirà la prossima settimana!). Posizionate il nuovo chip sulla basetta tenendo conto che in seguito ne dovrete montare altri due di dimensioni identiche. Il circuito è visibile in Fig.3.1 e include due LED, per avere una risposta visiva immediata. Come illustrato a lezione, quando i due ingressi \bar{S} e \bar{R} sono ad un livello alto, il circuito può permanere per un tempo indefinito in

una fra due configurazioni diverse. Portando temporaneamente uno o l'altro a GND, ossia a un valore logico basso, è possibile transire da uno stato stabile all'altro.

Task 7 Montate il circuito in Fig.3.1 e mostrate che è possibile farlo transire fra i due stati stabili agendo sugli ingressi \bar{S} e \bar{R} , connettendo momentaneamente uno o l'altro a GND. Quando il circuito è pronto dimostrate il funzionamento ai supervisori del laboratorio.

3.2 Approfondimenti

Il *latch* SR può essere visto come un primo passo verso la realizzazione di un *flip-flop* di tipo D, che sarà un elemento chiave del circuito studiato nella prossima esperienza. Per muoversi ulteriormente in questa direzione un possibile approfondimento consiste nel realizzare il circuito logico illustrato a lezione per l'implementazione di un *gated latch* di tipo D, che contiene un ingresso *enable* E e uno di *data* D. Come discusso, semplicemente quando $E = 0$ il circuito è "disabilitato" e si comporta come una memoria, diversamente mentre quando $E = 1$ il circuito registra continuamente il valore di D proiettandolo su Q (e, logicamente negato, su \bar{Q}).

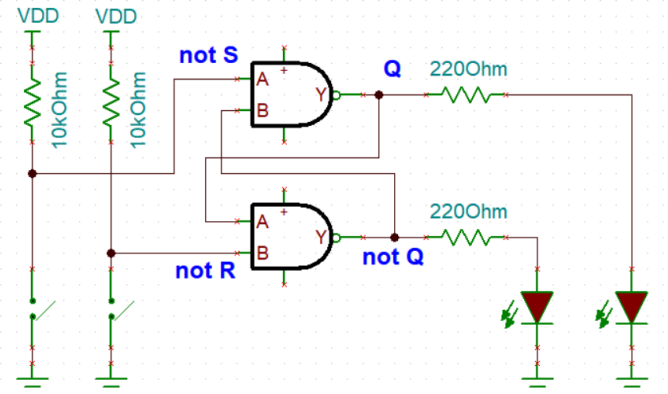


Figura 3.1: Circuito bistabile di tipo SR.