

Laboratorio di Fisica 3 AVANZATO

Prof. D.Nicolò e Prof.ssa C.Roda

Esercitazione N. D2 Latch, contatori e shift-register

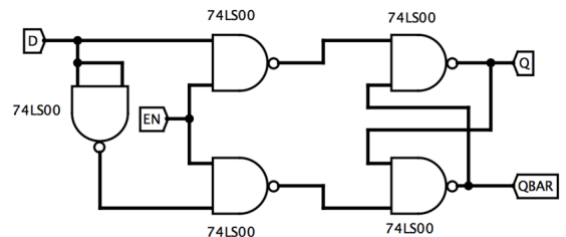
Questa esercitazione ha lo scopo di costruire alcuni circuiti logici sequenziali, progressivamente più complessi. Si raccomanda di eseguire un montaggio ordinato e pianificare lo spazio sulla basetta. Utilizzare fili di opportuna lunghezza e cercando di rispettare un codice colori consistente (ad es. fili di colore rosso e nero per l'alimentazione e la massa, un unico colore per la distribuzione del clock, etc.).

1) Materiale a disposizione. Consultare i data-sheet per le piedinature e le caratteristiche degli integrati:

- SN74LS00 Quad NAND Gate
- SN74LS163 4-bit synchronous binary counter with synchronous clear/load
- SN74LS74 Dual D-Latch
- SN74LS86 Quad XOR Gate

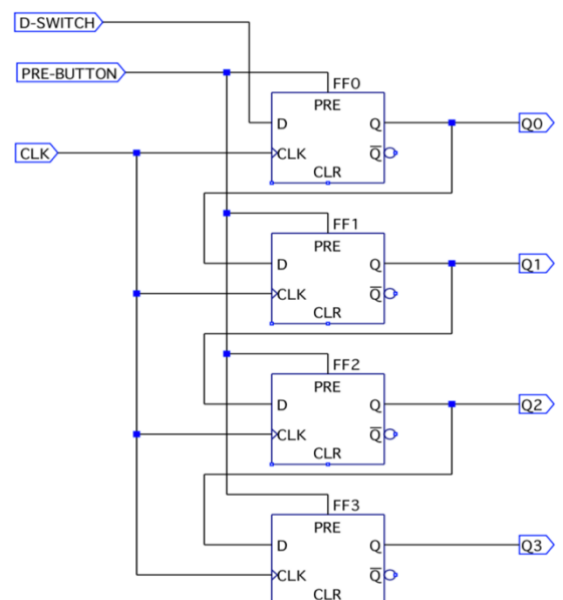
2) D-Latch con Enable

- Montare un D-Latch con enable utilizzando le porte NAND come indicato in figura. Collegare gli ingressi **DATA (D)** ed **ENABLE (E)** ad altrettanti pattern di tipo "Clock" dell'AD2 sincroni (cioè di frequenza di circa 1 kHz) e sfasati di 90 gradi.
- Spiegare brevemente il funzionamento del circuito indicando il ruolo degli ingressi D ed EN.
- Verificare il funzionamento del Latch osservando l'andamento dei segnali D, EN e Q (inviati ad altrettante linee di Logic), confrontando la successione osservata degli stati con quanto previsto dalla tabella delle verità ed allegando gli screen-shot necessari.
- Supponendo che ogni porta introduca lo stesso tempo di propagazione (indipendentemente dal segno della transizione), discutere il ritardo dell'uscita per ogni riga della precedente tabella di verità ed individuare per quale transizione dei segnali di ingresso il ritardo indotto sia massimo. Aggiustate eventualmente lo shift tra D ed E per visualizzare tutte le transizioni.



3) Shift register con edge-triggered D-Flip Flop

- Si costruisca uno shift register a 4 bit utilizzando 2 integrati 74LS74 (ciascuno con due FF di tipo D) seguendo lo schema in figura. Collegare:
 - gli ingressi di preset di tutti i FF ad uno StaticIO di tipo "Button" e polarità tale che l'uscita sia 1=released, 0=pressed;
 - l'ingresso D del FF0 ad uno staticIO di tipo Switch in modalità Push-Pull.
 - Pilotare il Clock con un segnale di tipo clock di Pattern.
 - Ogni uscita dei FF a canali dello StaticIO di tipo LED-software.

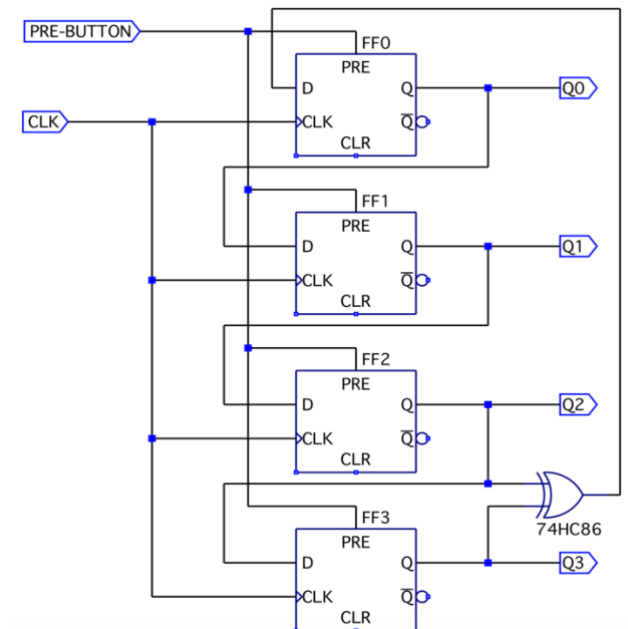


- b. Utilizzare il pulsante collegato al preset per portare lo shift register nello stato 1111. La commutazione delle uscite è sincrona o asincrona?
- c. Disabilitato il preset, inviare un clock di bassa frequenza (~1 Hz). Verificare controllando l'accensione/spengimento dei LED-software il funzionamento del circuito in corrispondenza a successive commutazioni del D-switch. Provare anche inizializzazioni diverse da 1111.
- d. Verificate quale ingresso guida le uscite se utilizzate sia il D-switch che il preset.
- e. si colleghi adesso l'uscita NOT(Q3) all'ingresso D del primo FF (al posto del Switch). Utilizzando un clock ad alta frequenza (~1 kHz) si descrivano le forme d'onde e le frequenze osservate sui piedini di uscita Q[3..0] collegate ad un bus di Logic in ordine di significatività. Spiegare il funzionamento del circuito.

NB: in teoria gli ingressi clear e preset possono essere lasciati disconnessi, ma se si dovessero osservare malfunzionamenti, potrebbe essere opportuno collegarli ai 5V attraverso una resistenza di pull-up da 1k Ω .

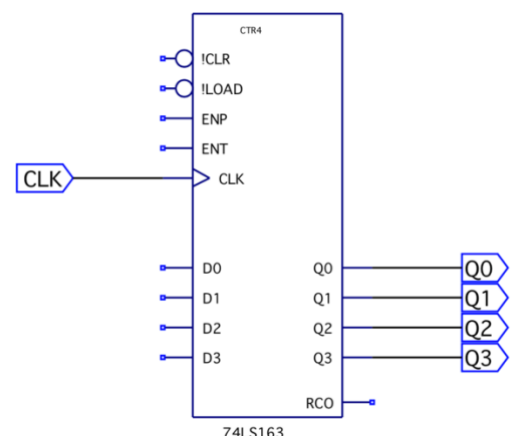
4) Generatore di sequenze pseudo-casuali

- a. Costruire un generatore di sequenze pseudo-casuali a 4 bit modificando il precedente circuito secondo lo schema riportato in figura, che utilizza i due bit più significativi (Q2 e Q3) come "tap", collegandoli ad un XOR la cui uscita è collegata all'ingresso Data del primo FF.
- b. Inviando un clock a 10 kHz verificare la sequenza e riportarla in una tabella facendo vedere che è completa. Registrare il bus di uscita della macchina in una finestra temporale di opportuna durata (tale cioè che se ne vedano tutti gli stati).
- c. Discutere quali altri "tap" si possono utilizzare per ottenere una sequenza completa, aiutandovi eventualmente con la documentazione presente in rete.



5) Utilizzo di un contatore come divisore di frequenza

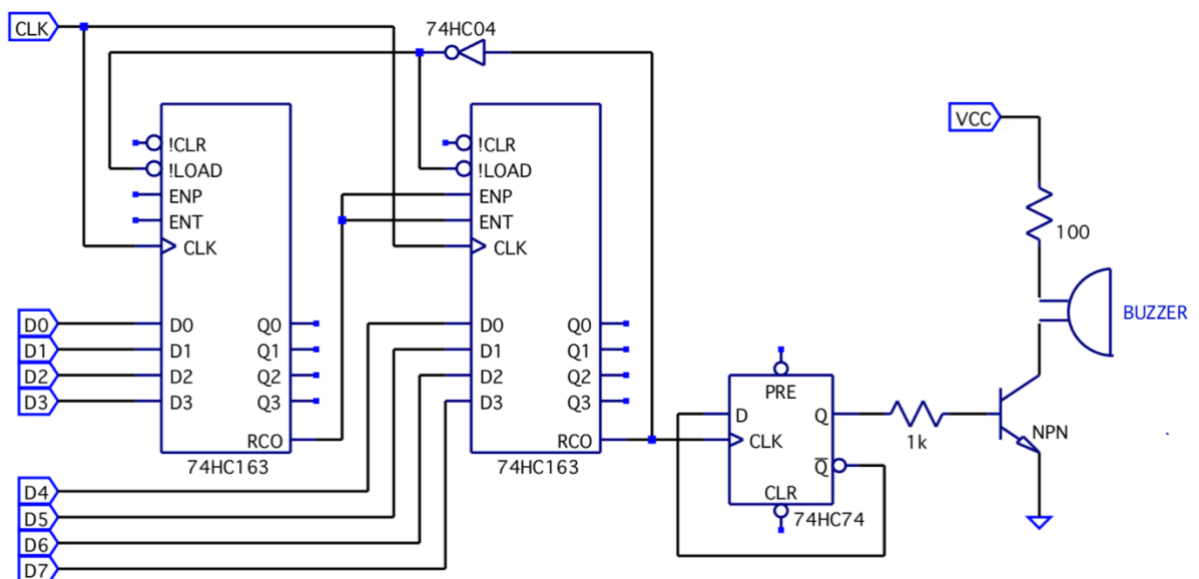
- a. Si vuole costruire un divisore di frequenza binario (x2, x4, x8, x16) utilizzando il contatore a 4 bit sincrono 74LS163. Montare il circuito come in figura, collegando le 4 uscite ad un bus con altrettanti bit di Logic, anche in questo caso rispettando l'ordine di significatività.
- b. Inviare un clock di frequenza circa 10 kHz e verificare che il circuito agisca come contatore visualizzando il valore del bus su una finestra temporale di durata pari a 20 periodi del Clock. Per una corretta visualizzazione dell'intero ciclo del contatore, potrebbe risultare conveniente impostare in Logic un trigger di tipo "Protocol" con la richiesta bus=0.
- c. Osservare individualmente i segnali Q0, Q1, Q2 e Q3 e verificare che la loro frequenza risulti pari rispettivamente ad 1/2, 1/4, 1/8, 1/16 della frequenza di clock.



- d. Verificare il comportamento sincrono del contatore visualizzando contemporaneamente il fronte del clock e quello dei singoli bit di uscita alla transizione del contatore $15 \rightarrow 0$. Impostare il trigger come al punto precedente e ridurre la larghezza della scala temporale fino a poche decine di ns.
- e. Progettare e costruire un circuito che conti 10 stati, in modo da avere segnale di frequenza $1/10$ della frequenza di clock utilizzando il clear sincrono o il load sincrono. Montare il circuito e discuterne il funzionamento indicando quale segnale abbia frequenza $1/10$ della frequenza di clock e con quale duty cycle.
- f. Progettare e costruire un circuito che realizzi un divisore di frequenza programmabile, utilizzando il "Ripple carry output" (RCO) per stimolare il "Load" (prestare attenzione al fatto che il Load del 74LS163 è "active low" ed occorre negare l'RCO prima di inviarlo). Caricare un valore iniziale nel contatore a seconda della divisione di frequenza che si vuole ottenere. Il valore iniziale si può definire mediante Pattern (bus a 4 bit di tipo PP costante) oppure StaticIO (4 switch oppure lo slider) collegando adeguatamente i canali DIO alle linee di ingresso D[3..0] del contatore (anche in questo caso prestare attenzione all'ordine dei bit).
- g. Osservare il ritardo (in termini di Clock ticks) tra il segnale di RCO e l'effettivo trasferimento sull'uscita delle linee di dato. Verificare che la funzione di Load sia effettivamente sincrona per questo integrato.
- h. Verificare la corrispondenza tra fattore di divisione in frequenza del RCO rispetto al Clock e valore iniziale del contatore e discutere il funzionamento del circuito.

6) Sintetizzatore musicale (facoltativo)

- a. Costruire un divisore di frequenza programmabile ad 8 bit secondo lo schema mostrato in figura, collegando in cascata due contatori 74LS163 (ovvero inviando l'uscita RCO del primo ad abilitare il Clock del secondo). Utilizzare per il segnale di Clock di entrambi l'uscita di un pattern regolato a 133 kHz. Collegare l'uscita RCO (negato) del secondo agli ingressi Load di entrambi i contatori ed all'ingresso di Clock di un FF di tipo D in modalità "toggle". Utilizzare infine l'uscita Q del FF per eccitare un buzzer. In analogia al punto precedente, è possibile programmare il fattore di divisione del Clock inviando un bus di pattern ad 8 bit agli ingressi D[7..0], così da regolare la frequenza di eccitazione del buzzer su una scala di circa 5 ottave (da circa 200 Hz fino ad alcune decine di kHz) in grado di coprire l'intervallo di sensibilità dell'orecchio umano.



- b. Per alcuni valori iniziali del contatore, verificare che il fattore di divisione in frequenza del RCO del secondo contatore sia compatibile con quanto atteso.

- c. Osservare il funzionamento in modalità toggle del FF, verificando che la sua uscita Q abbia frequenza dimezzata rispetto al suo Clock e duty-cycle prossimo al 50%.
- d. Provare ad eseguire una melodia a vostro piacimento fissando il fattore di divisione ad intervalli temporali regolari. Una possibile procedura da seguire potrebbe essere la seguente:
 - i. scrivere su un file di testo la successione delle note, riportando in ogni riga l'ottava, la nota ed il suo valore espresso in termini della figura di minima durata (ad es. in 64esimi se fosse una semibiscroma, ed in tal caso varrebbe 64 per una semibreve, ..., 8 per una croma, ..., 1 per una semibiscroma); le pause potrebbero essere trattate alla stregua di ultrasuoni;
 - ii. scrivere un programma (ad esempio una macro di Python) che importi quel file di testo per produrre in uscita un altro file testuale a singola colonna contenente il fattore della divisione in frequenza del Clock necessario a riprodurre quella nota (per la corrispondenza tra la nota ed il fattore di divisione si faccia riferimento alla tabella di corrispondenza allegata); ogni riga sia ripetuta un numero di volte pari al valore della nota stessa;
 - iii. importare il contenuto dell'ultimo file nel bus D[7..0], impostato in modalità Custom, e regolare la frequenza di scansione ("Parameter 1") sulla base dell'unità di tempo metronomica (tipicamente riportato sopra il primo pentagramma);
 - iv. far partire la riproduzione del brano e ... buon divertimento!