

EsD2: Costruzione di D-Latch, contatori e shift-register

Gruppo 1.AC
Matteo Rossi, Bernardo Tomelleri

12 aprile 2022

Misura componenti dei circuiti

Resistenze [Ω]	R	σR
R_{pot1}	9.53 k	0.08 k
R_{pot2}	9.78 k	0.08 k

Tabella 1: Valori di resistenza misurati per i componenti passivi dei circuiti studiati.

Riportiamo per completezza anche il valore della tensione continua di alimentazione per i circuiti integrati misurata con il multimetro

$$V_{CC} = 4.99 \pm 0.03V$$

Nota sul metodo di fit

Per determinare i parametri ottimali e le rispettive covarianze si è implementato in `Python` un algoritmo di fit basato sui minimi quadrati mediante la funzione `curve_fit` della libreria `SciPy`.

1 D-Latch con Enable

Si vuole studiare il funzionamento di un D-Latch con bit di Enable costruito utilizzando solo porte NAND appartenenti all'integrato SN74LS00, utilizzando come riferimento lo schema in fig. 1. Il circuito si ottiene a partire da un SR-Latch con Enable costruito tramite l'uso di porte NAND, andando poi a inviare all'ingresso R il segnale negato di S; la porta S sarà quindi rinominata in D.

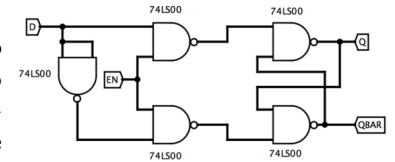


Figura 1

1.a Descrizione del funzionamento

Partendo da un SR-Latch, inviamo all'ingresso R il segnale negato di S: in questo modo i due bit non avranno mai lo stesso valore, andando così a prevenire lo stato proibito tipico del Latch e diminuendo effettivamente il numero di ingressi al circuito. Il nuovo circuito avrà quindi solo due ingressi, uno di enable che mi permette di sovrascrivere il segnale in uscita dal Latch e l'altro di Dato; sarà quindi chiamato D-Latch. Si può quindi costruire una tabella di verità per i D-Latch a partire da quella dei Latch SR.

E	D	Q'	$\overline{Q'}$
0	0	Q	\overline{Q}
0	1	Q	\overline{Q}
1	0	0	1
1	1	1	0

Tabella 2: Tabella di verità del circuito D-Latch

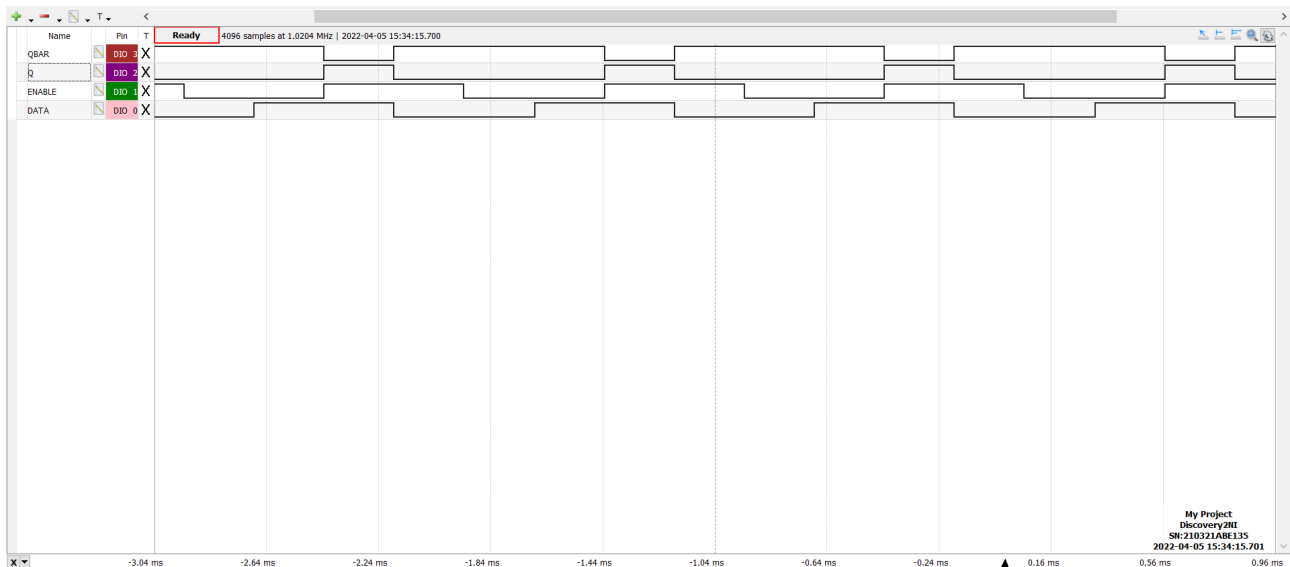


Figura 2: Acquisizione temporale con Logic dei segnali in entrata e in uscita da un D-Latch; tra i due segnali D ed E c'è uno sfasamento di 90°

Finché il bit di enable è 0, si mantiene lo stato di Hold, e in uscita da Q avrò sempre lo stesso valore. Quando invece il bit di enable vale 1, l'uscita Q assume il valore di D. Il circuito ha quindi il compito di memorizzare un bit: mentre E è 1 viene memorizzato un valore all'interno del Latch tramite D; quando poi E torna a essere 0 il circuito "ricorderà" il valore di D nell'istante immediatamente precedente al cambiamento di E.

1.b Verifica del funzionamento

Vogliamo ora verificarne il funzionamento; dopo aver montato il circuito utilizziamo la funzione Pattern per inviare due segnali di clock della stessa frequenza sfasati di $\pi/2$ a D e E. Utilizziamo poi Logic per acquisire i valori di D, E e Q, \bar{Q} in funzione del tempo. Dalle sezione 1.b e sezione 1.b possiamo notare che i valori di Q e \bar{Q} sono sempre uno il negato dell'altro, il valore di Q equivale a D nel caso in cui $E=1$, e che il valore di Q viene salvato quando E scende a 0; inoltre mentre $E=0$ il valore di D non va ad influire sul valore di Q. Da questo possiamo dedurre che il circuito ha il comportamento atteso.

1.c Ritardi dell'uscita Q

2 Shift Register con edge-triggered D-Flip Flop

Si vuole ora costruire uno Shift Register a 4 bit a partire dagli integrati della serie SN74LS74, secondo lo schema in fig. 4 e verificarne il funzionamento.

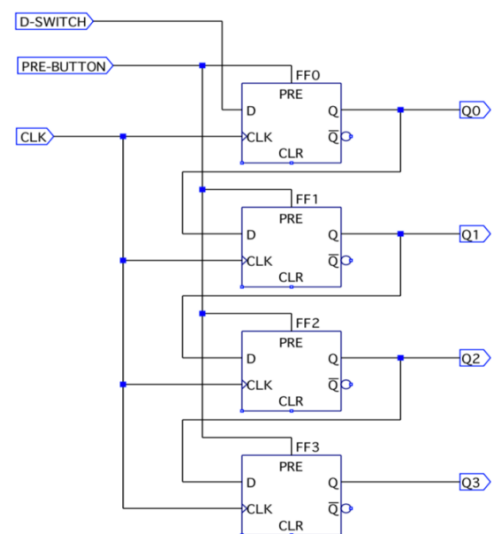


Figura 4

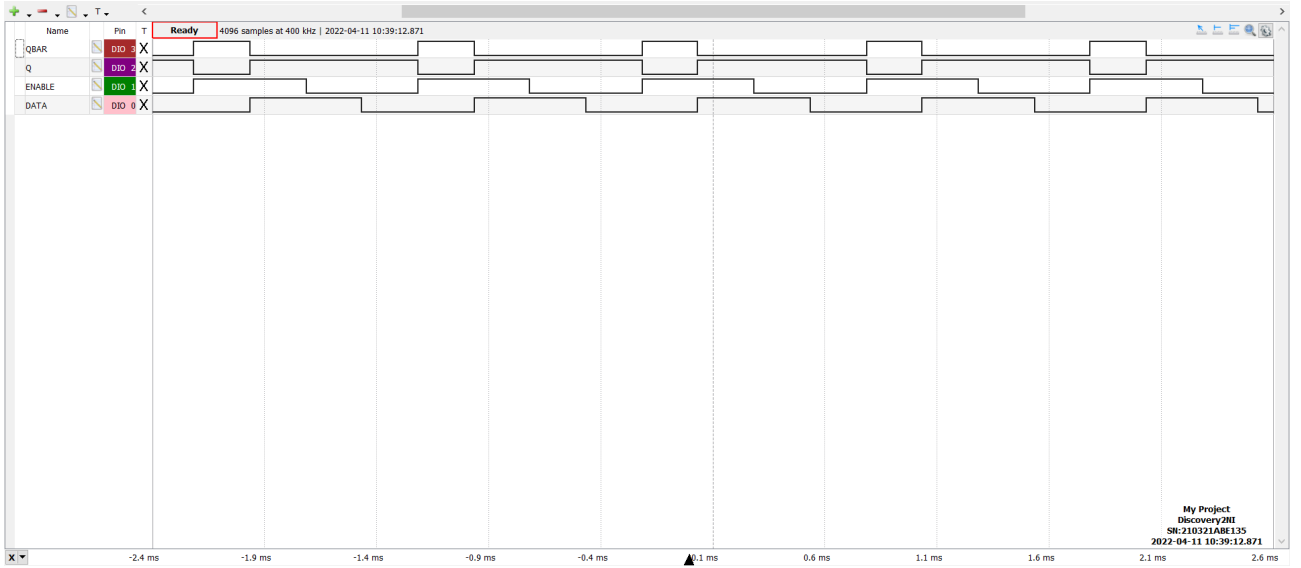


Figura 3: Acquisizione temporale con Logic dei segnali in entrata e in uscita da un D-Latch; tra i due segnali D ed E c'è uno sfasamento di -90°

2.a Funzionamento tramite Preset

Per prima cosa dopo aver montato il circuito verifichiamo la sincronicità delle commutazioni delle uscite. Dopo aver controllato che le uscite Q0, Q1, Q2 e Q4 fossero nello stato 0000, mantenendo il segnale di clock e il D-Switch scollegati abbiamo utilizzato la funzione StaticIO di wavegen per pilotare il pin PRE-BUTTON di fig. 4 tramite un button di tipo pressed=0 e released=1 e Logic per programmare un trigger che facesse partire l'acquisizione alla pressione del button, osservando le tracce prodotte dai 4 segnali in uscita. Dalla figura sezione 2.a vediamo che le commutazioni delle uscite avvengono in maniera sincrona, a un $\Delta T = 40\text{ns}$ a partire dalla pressione del pulsante di preset; successivamente le 4 uscite hanno raggiunto lo stato 1111.

2.b Funzionamento tramite Clock

A questo punto si vuole verificare il funzionamento del registro a scorrimento tramite un segnale di clock. Possiamo quindi costruire una tabella di verità in funzione del tempo del registro a partire dal periodo T del clock inviato: Come strategia per la verifica, utilizzeremo inizialmente il pulsante di preset per inizializzare

$t = t'$	$t = t' + \Delta T$	
Q_0	Q'_0	Q'_0
Q_1	$Q_0(t=t')$	$Q_0(t=t')$
Q_2	$Q_1(t=t')$	$Q_0(t=t'-\Delta T)$
Q_3	$Q_2(t=t')$	$Q_0(t=t'-2\Delta T)$

Tabella 3: Tabella "di verità" di un registro a scorrimento, Q'_0 è il valore che viene inviato durante l'impulso di clock tramite il D-SWITCH

tutte le uscite a 1, successivamente imposteremo il D-Switch tramite StaticIO come switch di tipo Push-Pull impostandolo a 0 e invieremo un segnale di clock dell'ordine di 1 Hz. Utilizzeremo quindi Logic per acquisire gli andamenti nel tempo delle uscite. Dunque ci si aspetta che dopo 3 periodi di clock (3 secondi) a partire da quando l'uscita Q0 diventa 0, tutte le uscite diventino 0, e che aspettando ulteriormente questi valori non cambino. Dalla sezione 2.b si verifica quanto detto prima, e le 4 uscite diventano (e si mantengono nel tempo visto che il D-Switch resta fisso a 0) tutte quante 0. Da questo si intuisce che collegando l'uscita Q_3 all'entrata del D-switch, possiamo generare una sequenza periodica. Dopo aver impostato tutte le uscite a 0, si collega l'uscita $\overline{Q_3}$ all'entrata D del primo Flip-Flop e si invia un clock di frequenza pari a 1 kHz al circuito. Ci si aspetta che, prendendo un'uscita a caso, si osservi un segnale di clock di frequenza un quarto di quella di clock: questo effetto è dovuto a come la sequenza viene caricata nel registro a partire dall'uscita $\overline{Q_3}$. In generale supponendo che nel circuito ci siano in cascata n Flip-Flop, e che tutte le uscite siano inizializzate a 0, in una qualsiasi delle uscite otterrò un clock di frequenza pari a $\frac{f_{clock}}{n}$.

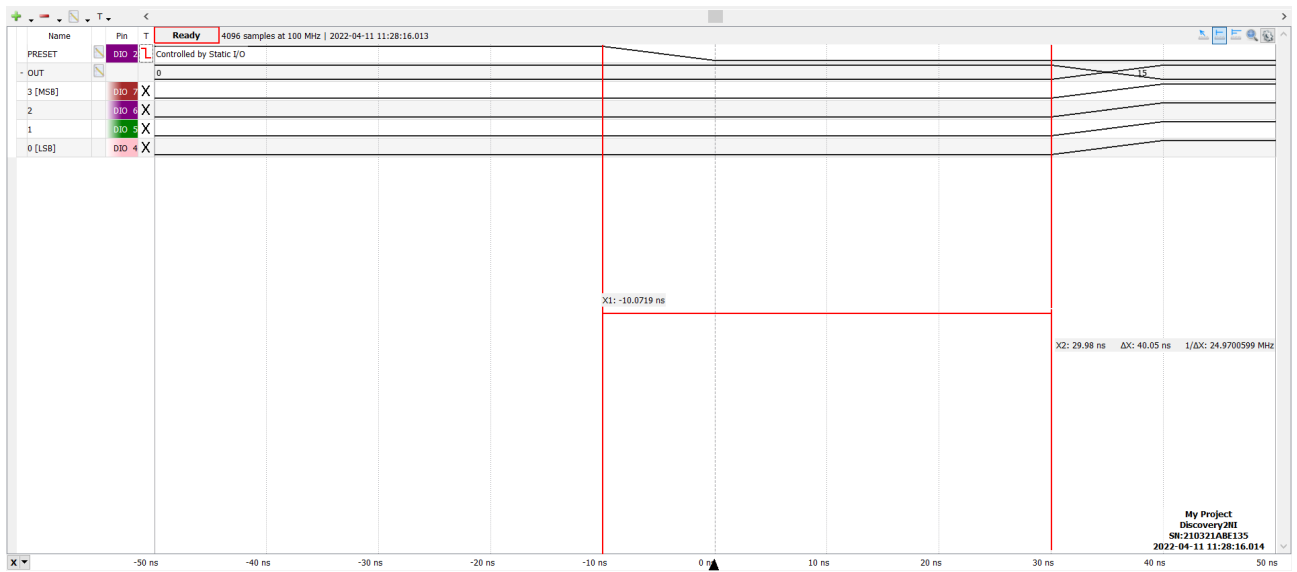


Figura 5: Acquisizione temporale con Logic dei segnali di PRE-BUTTON e i 4 segnali in uscita da un registro a scorrimento di 4 bit come illustrato in fig. 4

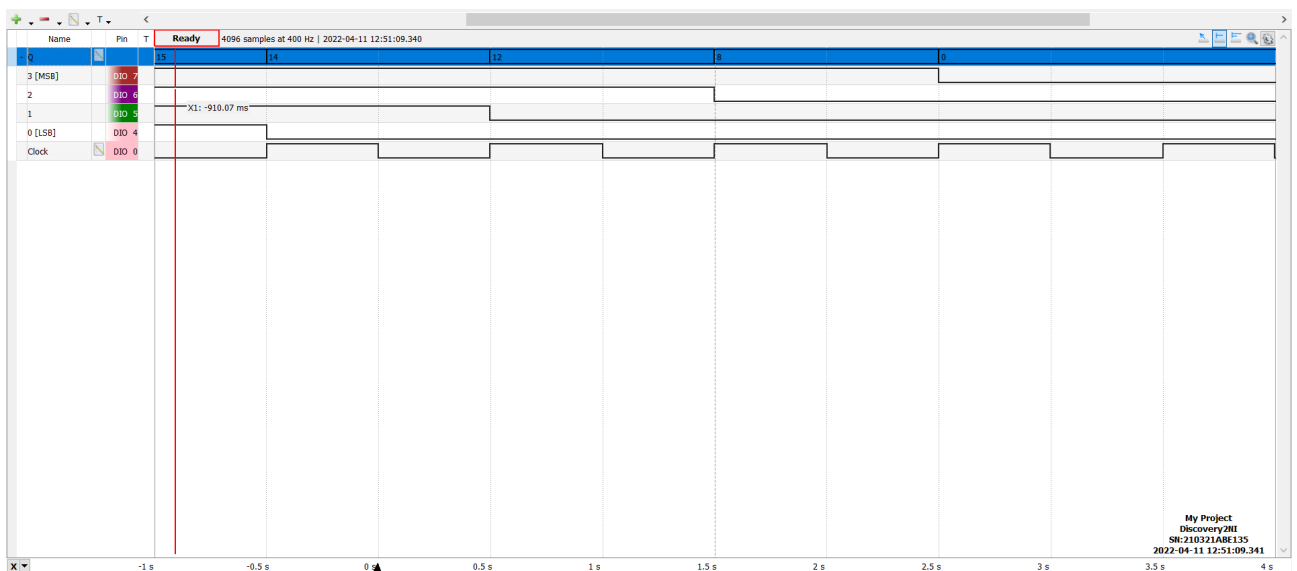


Figura 6: Acquisizione temporale con Logic dei segnali in uscita da un registro a scorrimento di 4 bit come spiegato in sezione 2.b

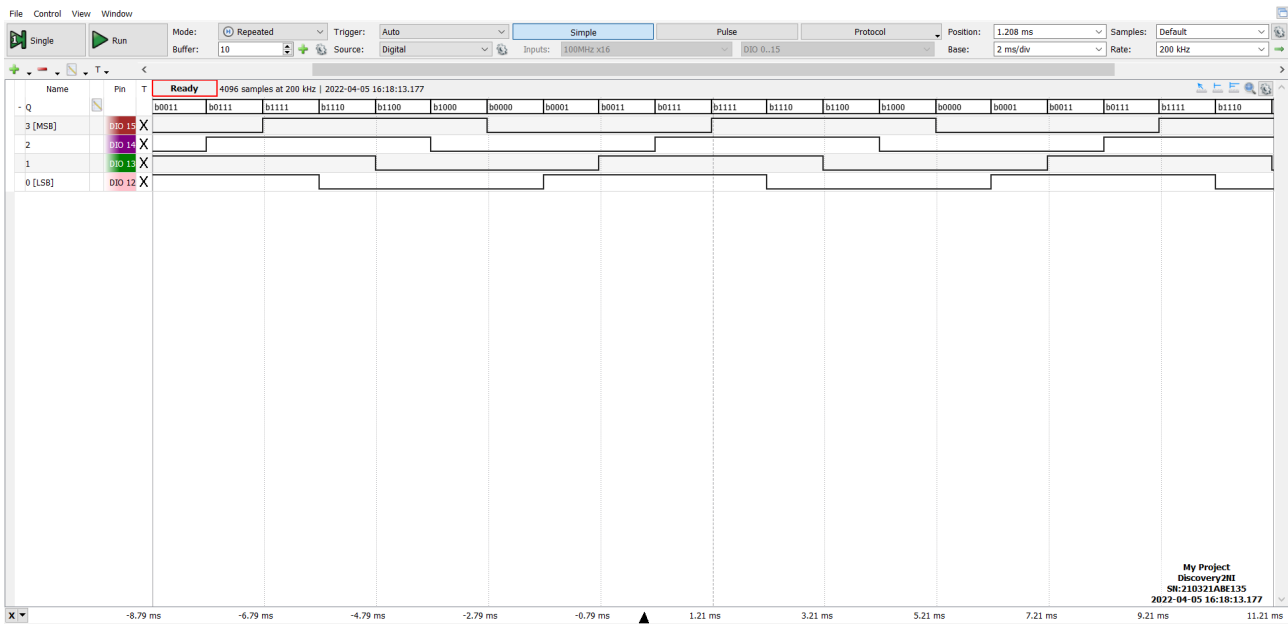


Figura 7: Acquisizione temporale con Logic dei segnali in uscita da un registro a scorrimento di 4 bit con l'ultima uscita negata collegata all'entrata del primo flip flop

3 Generatore di sequenze pseudo casuali

Si vuole ora costruire un generatore di sequenze pseudo-casuali a 4 bit utilizzando lo shift register costruito in precedenza e una porta XOR; la schematica del circuito che utilizzeremo è riportato in figura fig. 8.

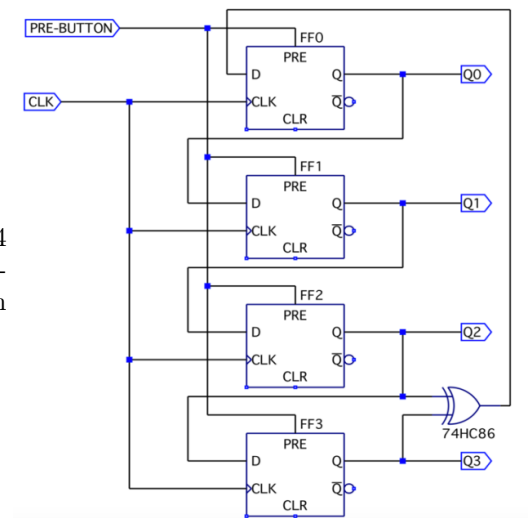


Figura 8

3.a Funzionamento e scelta dei TAP

Dopo aver montato il circuito si inizializzano tutti Flip-Flop a 1 e si invia un segnale di clock a 10 kHz per verificare il funzionamento: essendo il registro di lunghezza pari a 4 bit, dalla teoria ci aspettiamo che la sequenza si ripeta dopo $2^4 = 16$ eventi al massimo, condizione che si ottiene utilizzando come TAP (segnali in ingresso alla porta XOR, la cui uscita sarà inviata all'entrata D del primo Flip-Flop) le uscite Q_2 e Q_3 .

Dalla fig. 9 si verificano le aspettative per cui la sequenza generata a una qualsiasi uscita si ripete ogni 16 periodi di clock (essendo uno shift register la sequenza nelle altre uscite sarà la medesima, solo che saranno sfasate lungo l'asse temporale le une con le altre). Si provano quindi altre combinazioni di TAP, per verificare che la scelta di utilizzare l'uscita Q_2 e Q_3 produce una sequenza più lunga rispetto a qualunque altra configurazione

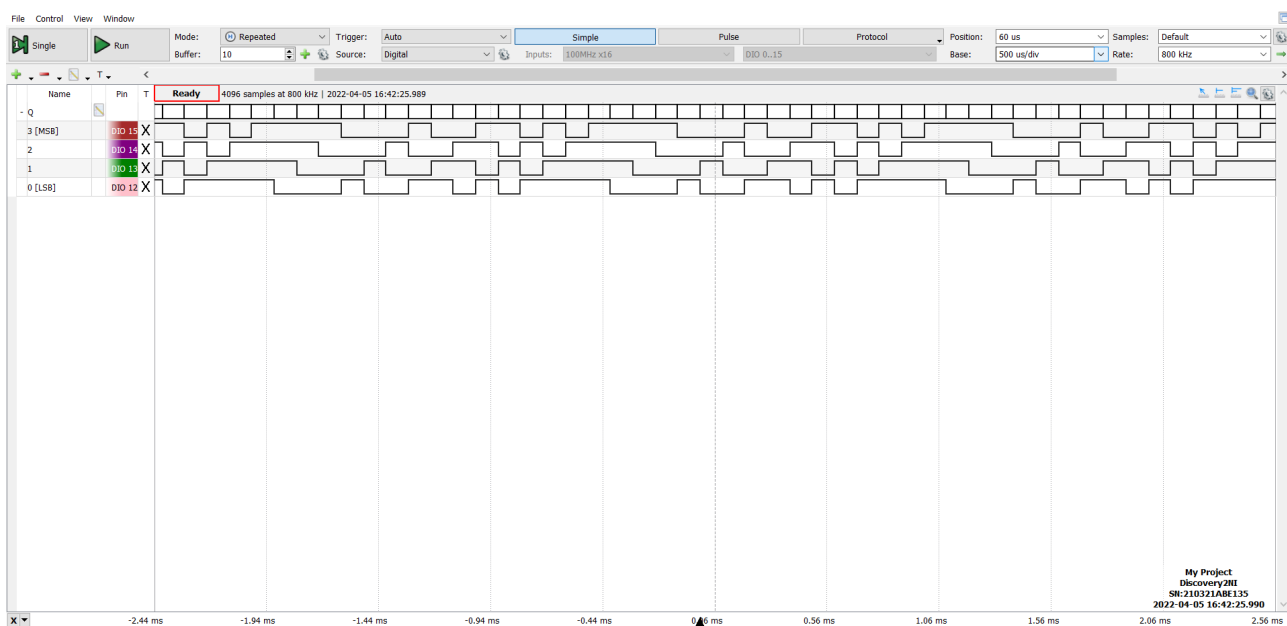


Figura 9: Acquisizione temporale con Logic del bus in uscita dal generatore di sequenze pseudo-casuale descritto in fig. 8

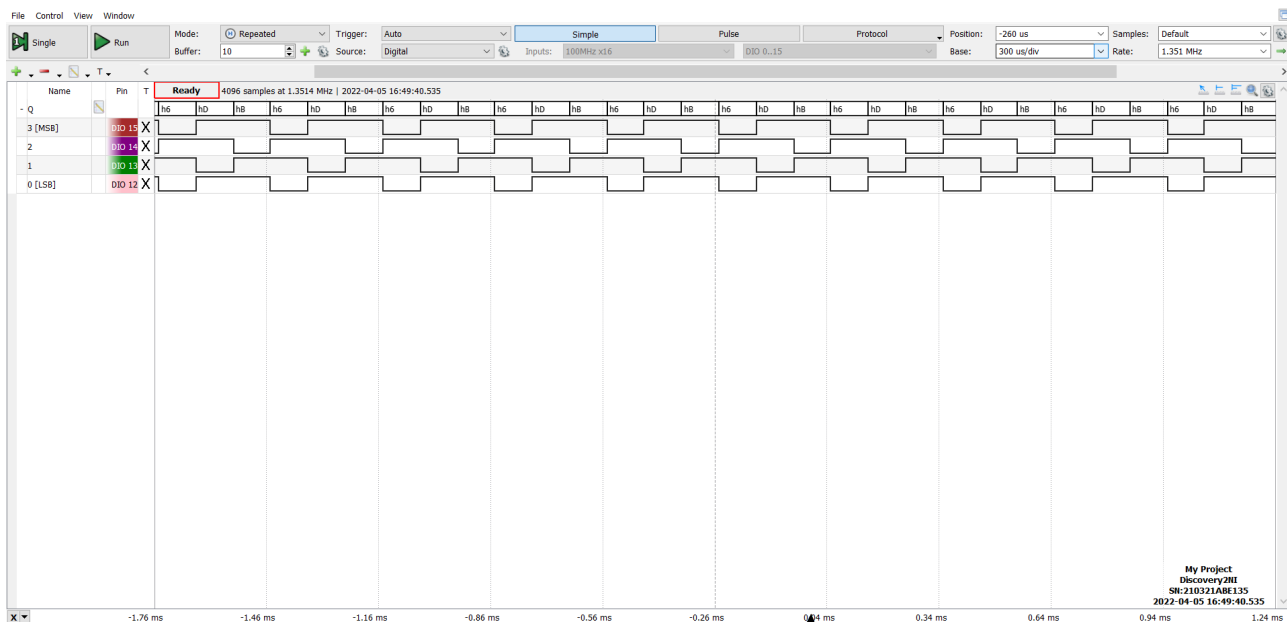


Figura 10: Acquisizione temporale con Logic del bus in uscita dal generatore di sequenze pseudo-casuale con TAP sulle uscite Q_0 e Q_1 , la sequenza si ripete ogni 4 eventi

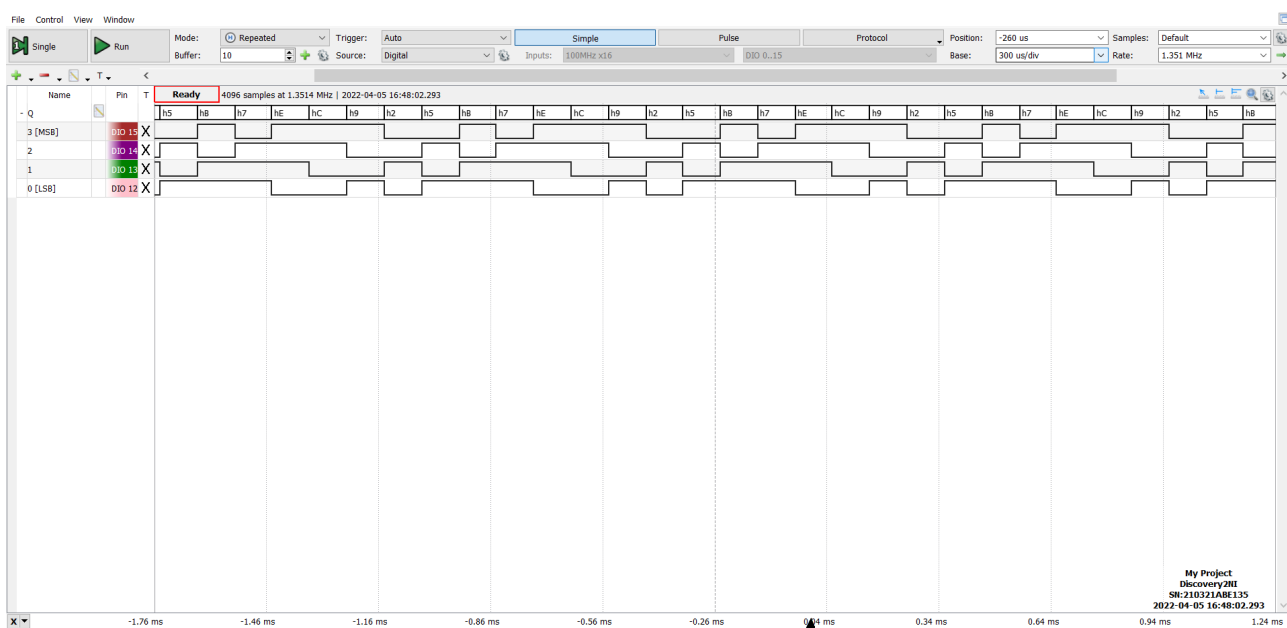


Figura 11: Acquisizione temporale con Logic del bus in uscita dal generatore di sequenze psudo-casuale con TAP sulle uscite Q_2 e Q_1 , la sequenza si ripete ogni 8 eventi

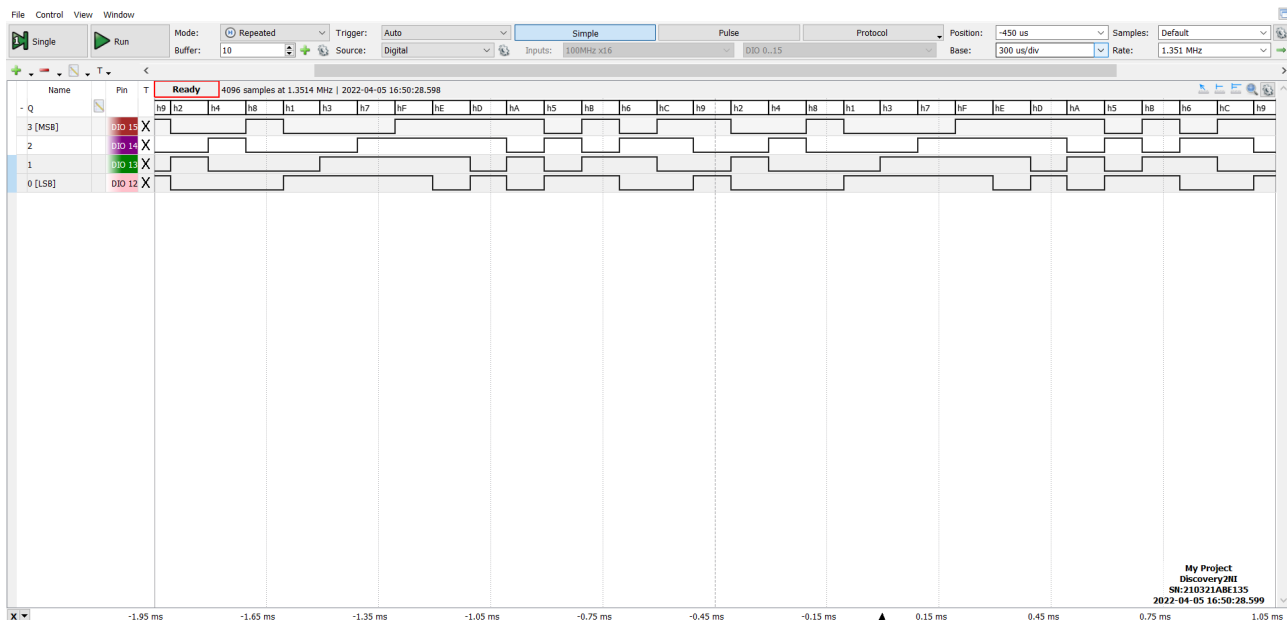


Figura 12: Acquisizione temporale con Logic del bus in uscita dal generatore di sequenze psudo-casuale con TAP sulle uscite Q_0 e Q_3 , la sequenza si ripete ogni 16 eventi

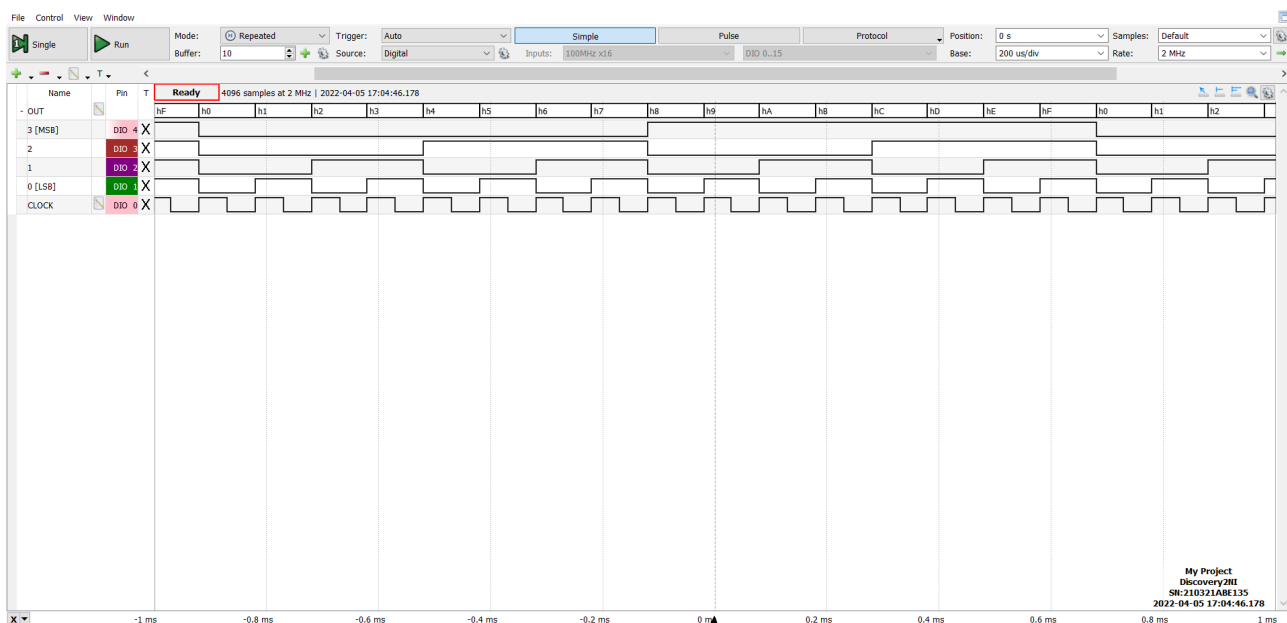


Figura 14: Acquisizione temporale con Logic del bus in uscita dal contatore, con frequenza di clock pari a 10 kHz

4 Divisore di frequenza

Si vuole infine costruire un divisore di frequenza a partire da un contatore binario a 4 bit, utilizzando l'integrato SN74LS163, presente in fig. 13. Si vuole innanzitutto verificarne il funzionamento.

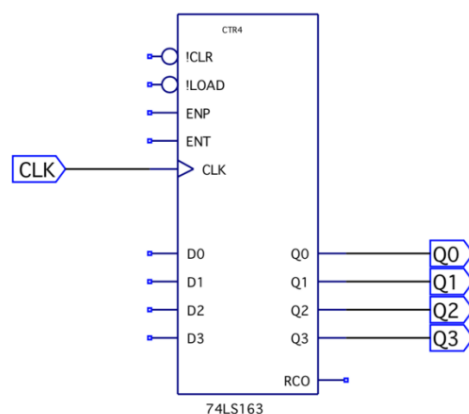


Figura 13

Conclusioni e commenti finali

Si è riusciti a verificare il corretto comportamento delle porte TTL studiate caratterizzandone le tensioni, correnti di operazione e tempi caratteristici di circuiti integrati come il SN7404. Inoltre, è stato possibile verificare il funzionamento di circuiti logici di diversa complessità costruiti con porte NAND, XOR, e OR e si è riusciti ad apprezzare l'effetto dei tempi di propagazione delle porte nella conversione dalla codifica Gray al binario.

Dichiarazione

I firmatari di questa relazione dichiarano che il contenuto della relazione è originale, con misure effettuate dai membri del gruppo, e che tutti i firmatari hanno contribuito alla elaborazione della relazione stessa.

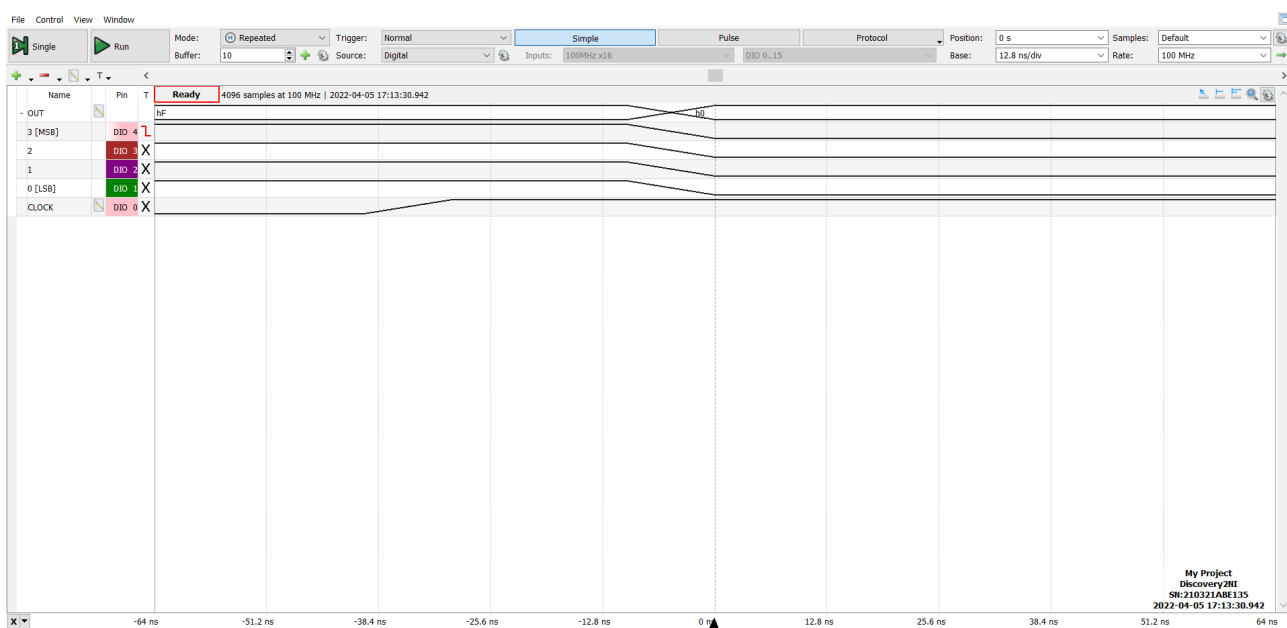


Figura 15: Acquisizione temporale con Logic del bus in uscita dal contatore durante la transizione 15->0; dall'immagine possiamo notare il comportamento sincrono della commutazione delle uscite del contatore

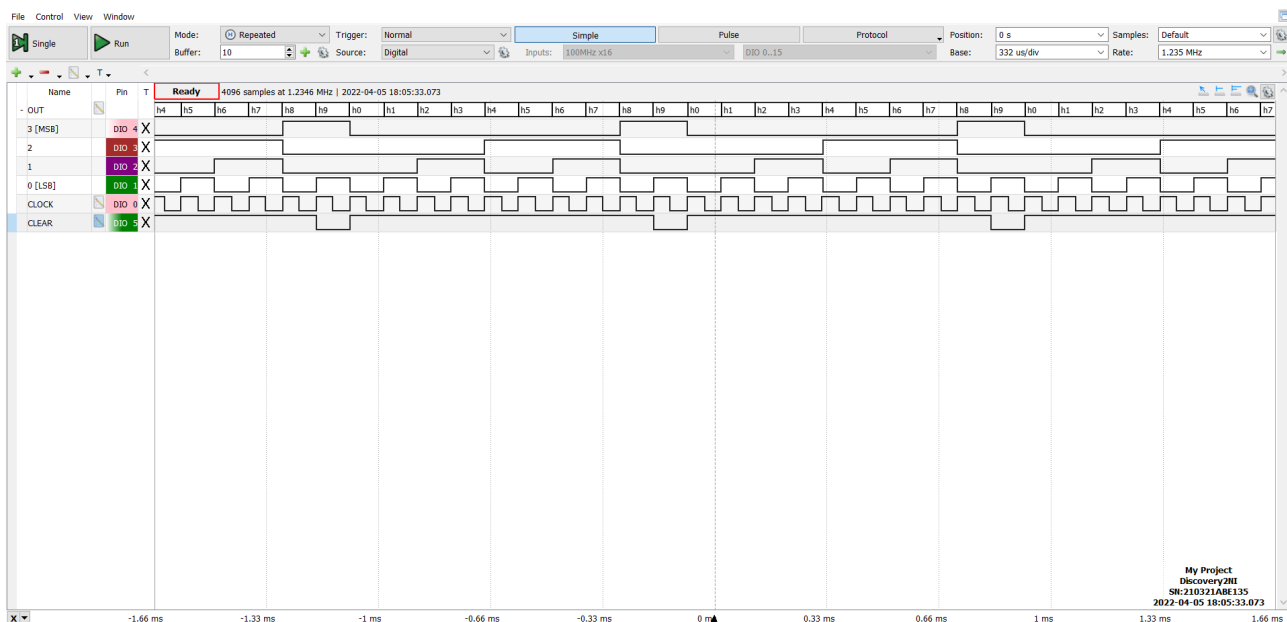


Figura 16: Acquisizione temporale con Logic del bus in uscita dal circuito che conta fino a 10; si può notare che il segnale in entrata nel Clock ha effettivamente periodo pari a 10 volte quello del clock