

# Latch, contatori e shift-register

Gruppo 1G.BM

Gabriele Astorino, Stefano Romboni, Matteo Morresi

7 Aprile 2022

## Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
<b>2</b>	<b>D-Latch con Enable</b>	<b>1</b>
2.1	Effetti del tempo di propagazione delle singole porte sui segnali . . . . .	2
<b>3</b>	<b>Shift register con Edge-Triggered D-Flip Flop</b>	<b>5</b>
3.1	Twisted-ring Johnson counter . . . . .	6
<b>4</b>	<b>Divisore di frequenza</b>	<b>7</b>
4.1	Verifica della divisione di frequenza . . . . .	7
4.2	Comportamento sincrono del contatore . . . . .	7
4.3	Contatore a 10 stati . . . . .	8
4.3.1	Tramite Clear . . . . .	8
4.3.2	Tramite Load . . . . .	9

## 1 Introduzione

In questa esperienza si andranno a costruire e studiare alcuni circuiti elettronici di complessità sempre maggiore. Si parte da un circuito latch di tipo D che serve a memorizzare un singolo bit, per poi costruire uno Shift register con edge-triggered D-Flip Flop per memorizzare una stringa di 4 bit ed infine si utilizzerà un contatore a 4 bit sincrono (74LS163) per produrre un segnale di frequenza 1/10 rispetto a quella di clock inviata direttamente al contatore.

## 2 D-Latch con Enable

Questo circuito (Figure 1) ha lo scopo di memorizzare il valore di D in uscita (Q) quando enable (E) è abilitato (alto) e di tenere la memorizzazione (quindi lo stato Q) quando Enable viene disabilitato per qualsiasi valore di D venga inviato in ingresso.

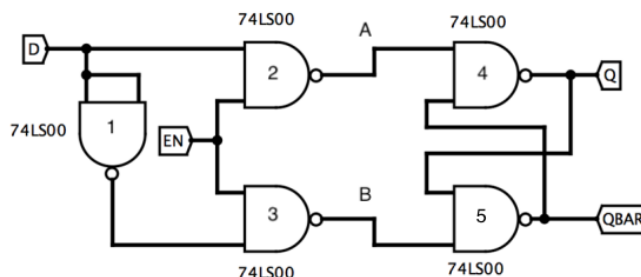


Figura 1: Circuito D-latch con Enable

E	D	$Q^+$	$\overline{Q}^+$
0	X	Q	$\overline{Q}$
1	0	0	1
1	1	1	0

Tabella 1: Tabella di verità circuito Figura(1)

Inviando tramite Patterns due segnali di clock sfasati di mezzo periodo all'ingresso D ed E, e misurando il valore degli stati tramite Logic, si vede come effettivamente il circuito montato segua la tabella di verità (1). Quando E è nello stato basso, indipendentemente dal valore assunto da D, le uscite Q e  $\overline{Q}$  rimangono immutate, mentre invece quando E è nello stato alto, abbiamo che Q e  $\overline{Q}$  sono in funzione di D.

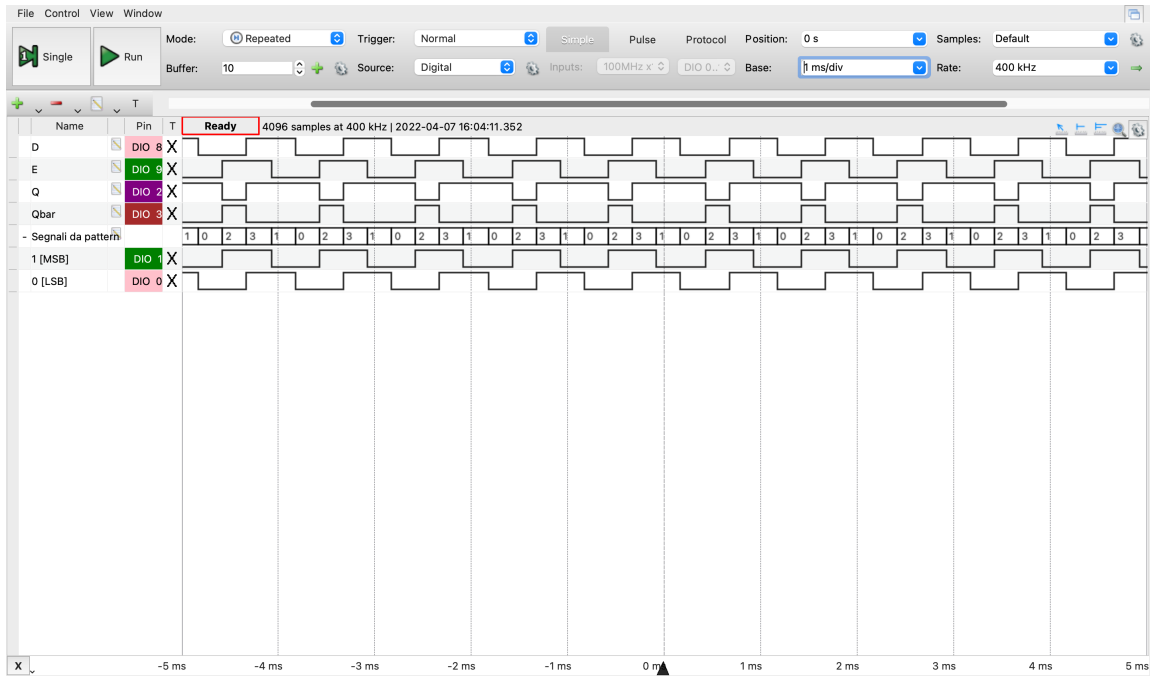


Figura 2: Verifica tabella di verità 1

## 2.1 Effetti del tempo di propagazione delle singole porte sui segnali

Ogni porta logica è caratterizzata da un proprio ritardo di propagazione del segnale. Non sono infatti strumenti ideali, per cui è necessario un lasso di tempo finito  $\Delta T$  per tradurre i segnali di ingresso in quello di uscita. Questo causa successivi ritardi del segnale che possono diventare anche significativamente importanti, qualora il numero di porte dovesse diventare sufficientemente elevato. Questo effetto diventa visibile andando a studiare nel dettaglio alcune transizioni dei segnali di ingresso e impostando una scala adeguata su WaveGen, nel nostro caso la minima possibile, ovvero 10 [ns] (la banda passante dell'AD2 è di 100 [MHz]).

**Transizione D  $1 \rightarrow 0$ , E=1** In questo caso, la presenza della porta NOT (1) introduce un ritardo in una parte di circuito che causa la presenza di uno stato basso in ingresso alle porte (2) e (3) del segnale D per un lasso di tempo limitato. Non abbiamo più quindi un circuito D-Latch, ma è come se avessimo un semplice Latch-RS con in ingresso S=0 e R=0. In sintesi, la transizione di D  $1 \rightarrow 0$  viene vista dalle porte (2) e (3) come una transizione  $1_2 0_3 \rightarrow 0_2 0_3 \rightarrow 0_2 1_3$  rispettivamente. I segnali in uscita Q e  $\overline{Q}$  rimangono inalterati nonostante il segnale D sia cambiato. Successivamente, in ingresso alle porte (2) e (3) abbiamo il corretto segnale D e  $\overline{D}$  rispettivamente, ma ora il segnale B ha un ritardo di un  $\Delta T$  in più rispetto al segnale di A. Quindi abbiamo che i segnali AB passano da

11 a 10, conseguentemente i segnali  $Q$  e  $\overline{Q}$ , che prima della transizione erano 1 e 0, passano ad 11, per poi cambiare ancora e andare in 01. In totale, come si vede anche dalla Figura (3), abbiamo un ritardo di circa 30 [ns] dalla transizione del segnale D all'effettiva risposta in uscita del circuito con i segnali  $Q$  e  $\overline{Q}$ .

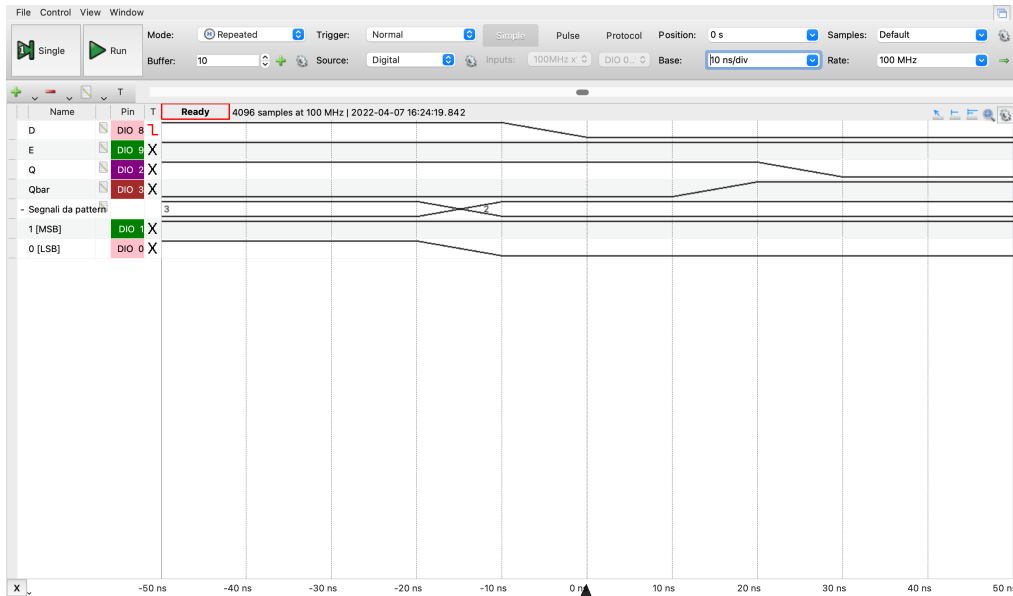


Figura 3: Acquisizione transizione  $D \ 1 \rightarrow 0$ ,  $E=1$

**Transizione  $D \ 0 \rightarrow 1$ ,  $E=1$**  In questo caso abbiamo la transizione di D da  $0 \rightarrow 1$ . L'analisi è uguale alla precedente, cambia solo che ora si ha in ingresso alle porte (2) e (3) uno stato alto per entrambe invece che uno basso. In pratica è come se avessimo in un Latch-RS il segnale S e R entrambi alti. In sintesi, la transizione di  $D \ 0 \rightarrow 1$  viene vista dalle porte (2) e (3) come una transizione  $0_21_3 \rightarrow 1_21_3 \rightarrow 1_20_3$  rispettivamente. Questo porta i segnali A e B ad essere entrambi bassi e quindi ad avere  $Q$  e  $\overline{Q}$  alti. Completata la transizione, si alza B, e dato lo stato in cui si trovavano  $Q$  e  $\overline{Q}$  (entrambi alti), si abbassa il segnale di  $Q$  e si stabilizza il valore in uscita. Anche qui, si vede dalla Figura (4) come il segnale di uscita sia in ritardo di circa 40 [ns] rispetto alla transizione del segnale di entrata.

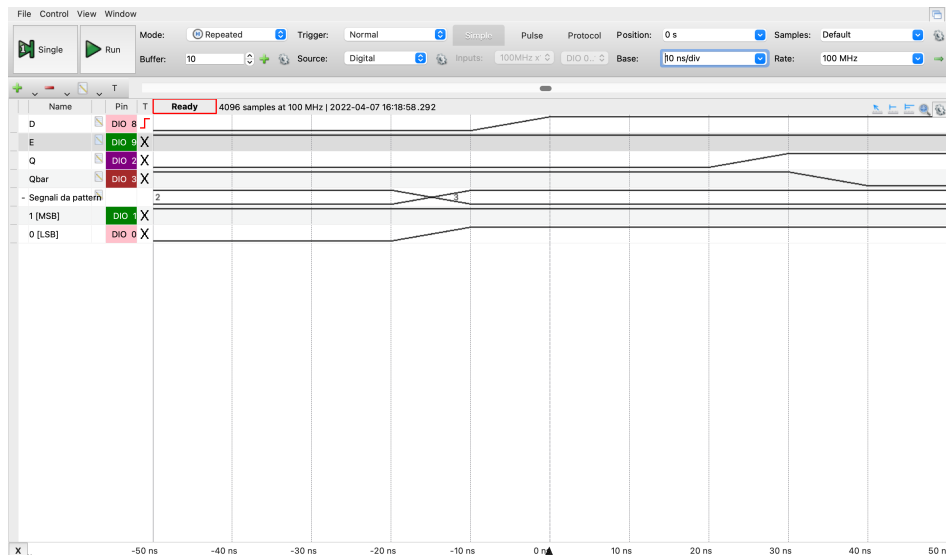


Figura 4: Acquisizione transizione  $D \ 0 \rightarrow 1$ ,  $E=1$

**Transizione D=1, E  $0 \rightarrow 1$**  In questo caso, il circuito di memorizzazione viene attivato quando Enable completa la transizione  $0 \rightarrow 1$ . In questo modo, in A e B abbiamo stati alti, che producono come risposta in output la transizione da basso a alto per Q, e la permanenza dello stato  $\bar{Q}$ . In questa nuova "configurazione", allo stadio successivo si ha l'abbassamento di  $\bar{Q}$ , e la stabilizzazione del segnale. Dalla Figura (5) si vede come il delay temporale totale sia di circa 20 [ns].

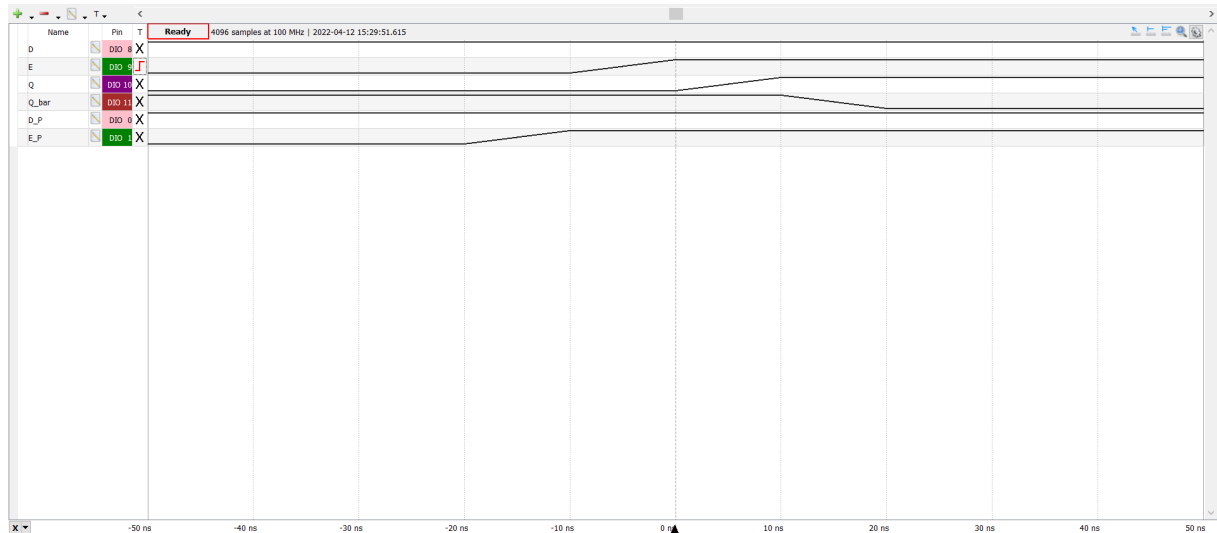


Figura 5: Acquisizione transizione E  $0 \rightarrow 1$ , D=1

**Transizione D=0, E  $0 \rightarrow 1$**  Caso analogo al precedente con un ritardo totale di 20 [ns].

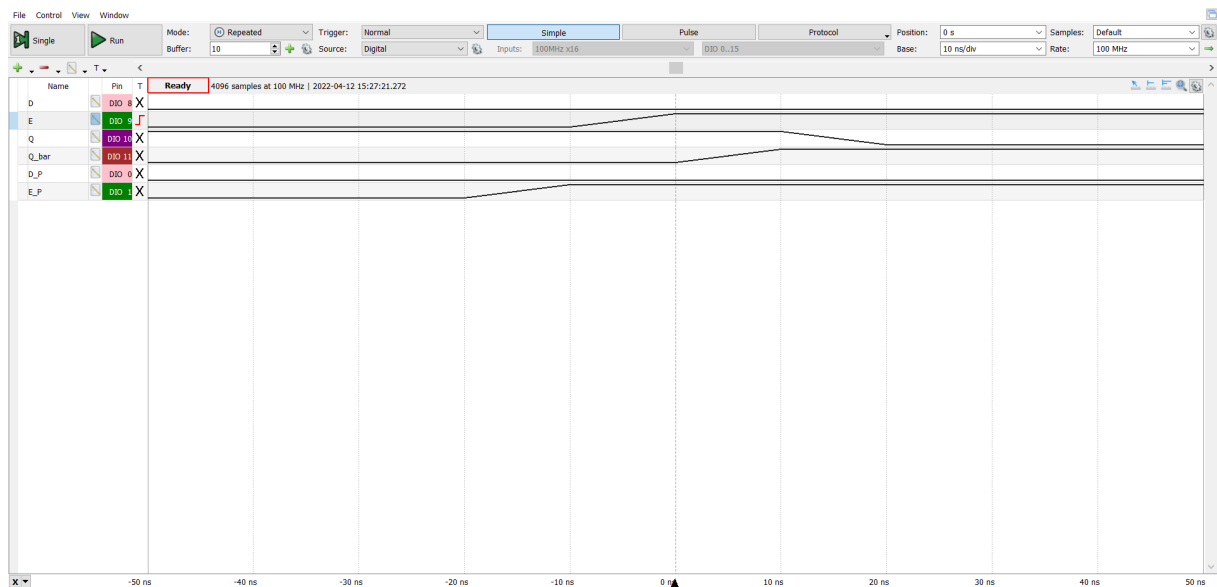


Figura 6: Acquisizione transizione E  $0 \rightarrow 1$ , D=0

**Altre transizioni** In tutte le altre transizioni in cui  $E=0$ , oppure in cui si aveva  $E 1 \rightarrow 0$  non si osserva alcun cambiamento nelle uscite come è atteso dallo studio delle tabelle di verità.

**Osservazioni** Vogliamo sottolineare comunque che i tempi di ritardo in questo caso sono da considerare più come stime che come delle misure vere e proprie. Questo perchè la strumentazione utilizzata non permette di avere una risoluzione temporale tale da poter eseguire una misura ragionevole del delay temporale delle porte. Da datasheet risulta infatti che il ritardo delle porte tipicamente è proprio di 10 [ns] (al massimo di 15 [ns]). Le misure effettuate danno comunque una conferma sul fatto che le porte utilizzate rientrano tutte nelle specifiche del datasheet. Un'altra cosa da sottolineare è lo sfasamento tra i segnali generati da Patterns e i segnali misurati da Logic. Come si vede dalle immagini, i segnali misurati tramite Logic risultano caratterizzati da uno sfasamento di circa 10 [ns] rispetto al rispettivo segnale inviato da Patterns. Questo è sempre un effetto della banda passante di 100 [MHz] dell'AD2.

### 3 Shift register con Edge-Triggered D-Flip Flop

Si realizza ora uno shift register a 4 bit (Figura 7) utilizzando due integrati 74LS74 (ciascuno con due positive-edge-triggered D-Flip Flop). Ricordiamo che per ciascun Flip Flop vale la tabella di verità 2.

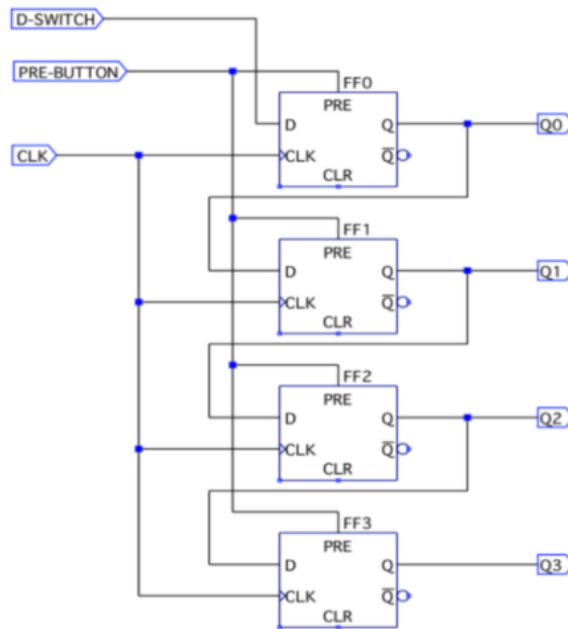


Figura 7: Schema dei collegamenti tra D-Flip Flop per realizzare uno shift register a 4 bit.

operazione	PRE	CLR	D	Q	$\overline{Q}$
preset	L	H	X	H	L
clear	H	L	X	L	H
indeterminata	L	L	X	H	H
Set	H	H	H	H	L
Reset	H	H	L	L	H

Tabella 2: Tabella di verità dei D-Flip Flop utilizzati. H = stato alto, L = stato basso, X = Don't care. Lo stato in D viene passato alle uscite Q,  $\overline{Q}$  da una transizione positiva del clock.

Si collegano gli ingressi di preset di tutti i Flip Flop utilizzati ad uno StaticIO di tipo “Button” con polarità tale che l’uscita sia 1 = released, 0 = pressed. Si collega l’ingresso D del FF0 ad uno staticIO di tipo Switch in modalità Push-Pull. Si pilota il Clock con un segnale di tipo clock di Patterns. Si collega infine ogni uscita dei Flip Flop a canali dello StaticIO di tipo LED-software.

Premendo il pulsante di StaticIO che controlla gli ingressi di preset si mandano le uscite dello Shift register allo stato 1111. Si osserva che la commutazione delle uscite è asincrona, in quanto non dipende dal clock, ma avviene non appena si attiva il comando di Preset. Le uscite però cambiano il proprio stato tutte contemporaneamente. Cambiando lo stato di inizializzazione del Preset, avremmo cambiato il valore ottenuto alle uscite una volta premuto (esempio, impostando 1010, si ottiene in uscita 1010). Si disabilita ora il preset, e si invia un clock di frequenza 1 [Hz]. Si verifica controllando l’accensione dei LED-software che il circuito si comporti come atteso in risposta a successive commutazioni del D-Switch. Se si manda all’ingresso  $D_0$  del Flip Flop 0 un bit  $b_0$ , una transizione positiva del clock lo trasferisce in  $Q_0 = D_1$ ; il successivo ciclo di clock porta il bit in  $Q_1 = D_2$  e così via fino ad avere dopo quattro cicli di clock il bit  $b_0$  all’uscita  $Q_3$ . Tenendo lo Switch alto per un tempo corrispondente a 4 cicli di clock si ottiene in uscita la sequenza 1111, che appare (se si parte da uno stato con uscite 0000) come un’accensione successiva di tutte le uscite da  $Q_0$  a  $Q_3$ . Commutando lo Switch si possono inserire sequenze diverse, anche se a tale scopo risulta più conveniente utilizzare la funzionalità Patterns, che permette di regolare meglio sulla frequenza del clock le sequenze in ingresso al D-Switch.

Si osserva infine in accordo con la tabella di verità 2 che l’ingresso di preset ha la priorità rispetto al D-switch nel determinare le uscite: quando il preset è basso tutte le uscite sono alte (oppure seguono l’inizializzazione scelta) indipendentemente da cosa si ha in ingresso al D-switch, quando invece è alto permette al D-switch di pilotare il registro.

### 3.1 Twisted-ring Johnson counter

Si collega ora l’uscita  $Q_3$ , tramite una porta NOT realizzata con una porta NAND dell’integrato 74LS00, all’ingresso  $D_0$  del primo Flip Flop (al posto dello Switch). Si invia un clock di 1 [kHz] e si acquisiscono le uscite  $Q_0, Q_1, Q_2, Q_3$  con un bus di Logic in ordine di significatività. Riportiamo in figura 8 quanto osservato. L’unica peculiarità

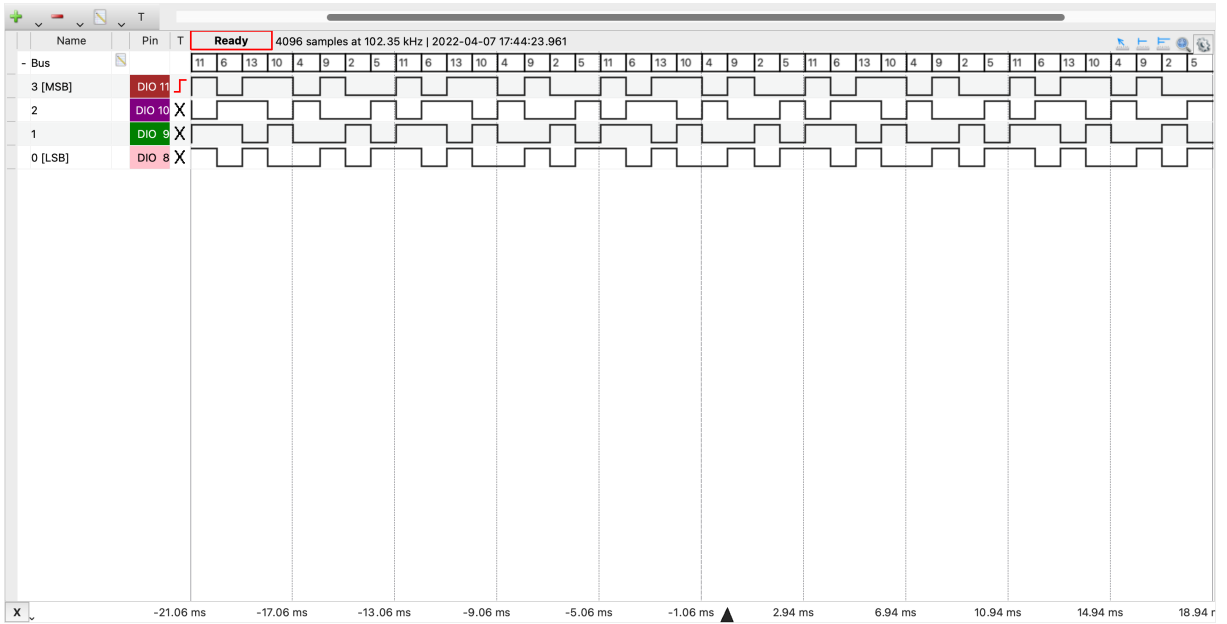


Figura 8: Bus delle uscite con indicizzazione corrispondente a quella delle uscite  $Q_i$ ,  $i = 0,1,2,3$ .

di questo circuito rispetto allo shift register discusso precedentemente è il principio di regolazione dell’ingresso  $D_0$  del Flip Flop 0: si ha che se all’uscita  $Q_3$  si ha il bit  $b_0$ , allora  $\overline{b_0}$  viene mandato all’ingresso  $D_0$  e quindi ad ogni transizione positiva di clock successiva viene propagato in ordine alle uscite  $Q_0, Q_1, Q_2$ , fino ad arrivare all’uscita dell’ultimo Flip Flop. Si ha poi che il ciclo si ripete, questa volta (data la doppia negazione) con il bit  $b_0$  in ingresso

al primo Flip Flop. Si ottiene allora su ciascuna delle quattro uscite lo stesso segnale periodico di periodo  $8T_{CLK}$  (con  $T_{CLK}$  periodo del clock), con sfasamento relativo tra le uscite determinato dalla frequenza di clock scelta e costituito da un succedersi della sequenza di 4 bit che si aveva memorizzata nelle quattro uscite prima di collegare  $\overline{Q_3}$  a  $D_0$  e dalla sua negazione. Si ha quindi che un circuito di questo tipo è anche un divisore di frequenza che, se  $n$  è il numero di Flip Flop che costituiscono il registro, divide per  $2n$ .

## 4 Divisore di frequenza

### 4.1 Verifica della divisione di frequenza

In questa sezione si vuole verificare che il contatore utilizzato come divisore di frequenza funzioni correttamente: in particolare che, visualizzando con la funzione Logic dell'Ad2 i segnali alle uscite  $Q_0, Q_1, Q_2, Q_3$ , abbiano frequenza rispettivamente  $1/2, 1/4, 1/8$  e  $1/16$  rispetto a quella del segnale di Clock mandato in ingresso,  $f = 10$  [kHz].

A tale scopo si visualizzano 20 periodi del segnale di clock e le uscite per  $Q_0, Q_1, Q_2, Q_3$ , rispettivamente con DIO 9, DIO 10, DIO 11, DIO 12. Riportiamo ora l'immagine dell'acquisizione in Figura (9).

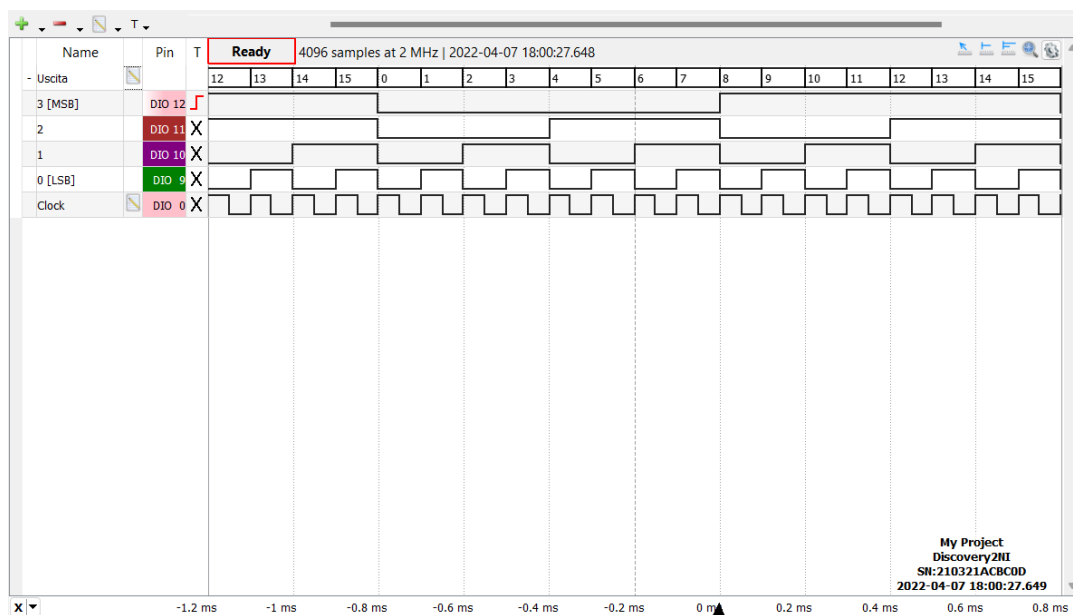


Figura 9: Immagine con i segnali acquisiti per il funzionamento del contatore come divisore di frequenza.

Come si nota da quanto riportato, le uscite hanno la frequenza riscalata correttamente: per esempio DIO 9 oscilla a 5 [kHz], infatti si contano 10 periodi. Analogo per gli altri.

### 4.2 Comportamento sincrono del contatore

Si vuole ora verificare il comportamento sincrono del contatore. Per fare ciò si visualizza (sempre mediante Logic) contemporaneamente il fronte del clock e quello dei singoli bit di uscita alla transizione del contatore  $15 \rightarrow 0$  impostando un time division di 10 [ns]/div.

I segnali sono impostati sui DIO analoghi a quelli in Sezione 4.1.

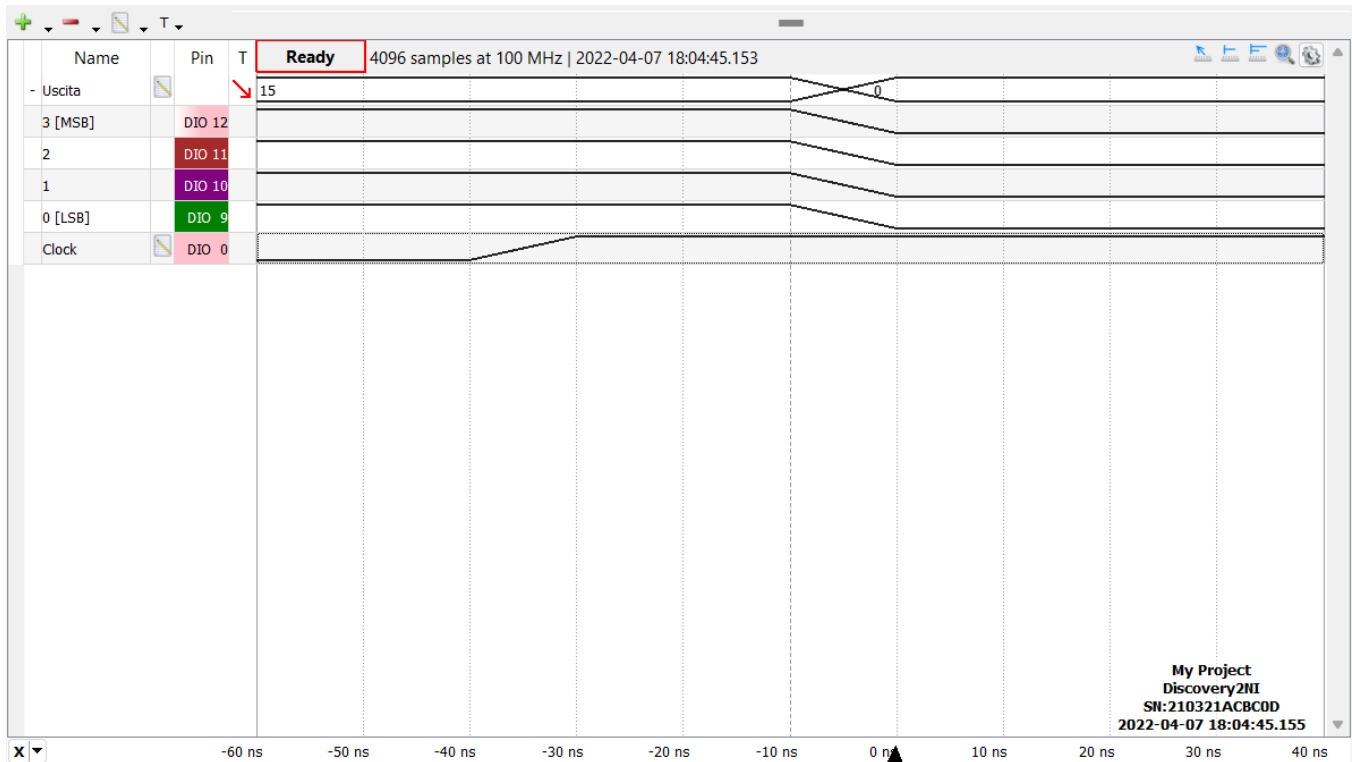


Figura 10: Verifica del comportamento sincrono del contatore visualizzando la transizione 15→0.

Come riportato in Figura 10 la transizione  $15 \rightarrow 0$  dei bit in uscita avviene in maniera sincrona. Per quanto riguarda invece il delay temporale di circa 20 [ns] (compatibile con le specifiche riportate da datasheet) tra lo switch basso-alto nel clock e l'inizio di  $15 \rightarrow 0$  in uscita, questo è dovuto al ritardo introdotto dalla somma dei ritardi delle singole porte presenti nel circuito.

### 4.3 Contatore a 10 stati

Si realizza ora un circuito che conti 10 stati, in modo da avere un segnale di frequenza  $1/10$  della frequenza di clock, utilizzando prima il clear sincrono e poi il load sincrono.

#### 4.3.1 Tramite Clear

Per realizzare un contatore a 10 stati, si vuole attivare il clear solo quando il contatore arriva a 9 (partendo da 0 sono in totale 10 stati) e che lo lasci disattivato, quindi alto, in tutti gli altri casi. Contando in binario si osserva come solamente quando il contatore arriva a 9 abbiamo entrambe le uscite  $Q_0$  e  $Q_3$  alte, in tutti gli altri casi sono basse. A tale scopo, presa la configurazione circuitale precedente si inviano le uscite  $Q_0$  e  $Q_3$  in ingresso ad una porta NAND dell'integrato 74LS00, e l'uscita di quest'ultima viene poi utilizzata per pilotare il clear. Il clear è attivo basso, si attiva solamente quando sia  $Q_0$  che  $Q_3$  sono alti, ovvero al numero 1001. Una volta attivo il clear, tutte le uscite vengono portate a zero alla successiva transizione ( $0 \rightarrow 1$ ) del clock. Il segnale ottenuto ha un duty cycle del 90%. I segnali sono graficati in Figura (11).



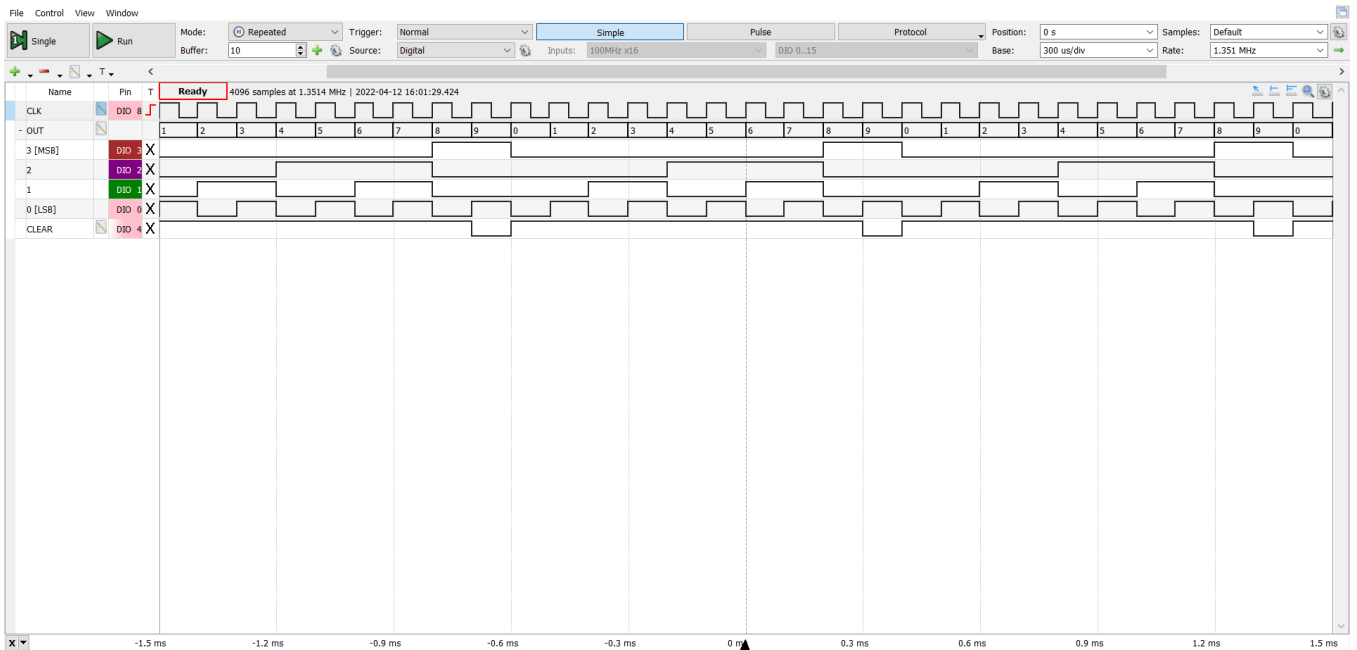


Figura 11: Misura dei segnali di clock, di uscita e del clear.

#### 4.3.2 Tramite Load

In alternativa al circuito precedente, per ottenere lo stesso segnale, si può mandare l'uscita della porta NAND all'ingresso del Load. In aggiunta, serve anche inviare agli ingressi  $D_0, D_2, D_2, D_3$  un segnale di 4 bit. Questo segnale, passato un ciclo di clock, sarà il nuovo punto di partenza del contatore. Nel nostro caso, volendo un contatore a 10 stati, è sufficiente inviare in ingresso il segnale 0000, così che dopo essere arrivato a 9, riparta nel conteggio da 0. Si noti come in questo caso, si poteva far ripartire il contatore anche da un numero diverso da 0, potendo ottenere quindi un segnale di frequenza pari a  $1/9$ , oppure  $1/5$  (in generale  $1/n$  con  $n \in [1, 10]$ ) di quello di clock.