

# N.D02: Latch, contatori e shift-register

Gruppo 23

Alessandro Costanzo Ciano, Luca Palumbo

16 aprile 2024

## 1 D-Latch con Enable

### 1.1 Montaggio

Il circuito è stato montato come da schema, utilizzando le porte NAND come indicato in figura. Gli ingressi DATA (D) ed ENABLE (E) sono stati collegati ad altrettanti pattern di tipo “Clock” dell’AD2 sincroni (cioè di frequenza di circa 1 kHz) e sfasati di 90 gradi.

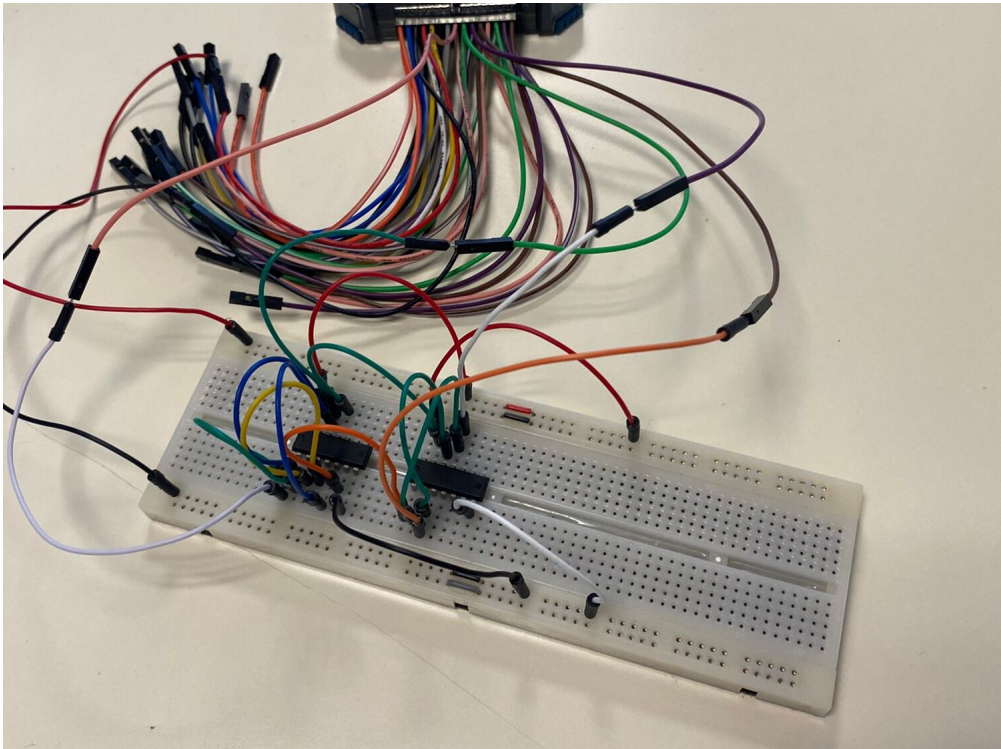


Figura 1: Foto del circuito D-Latch con enable.

### 1.2 Funzionamento

Il circuito è un D-Latch con enable. Il ruolo degli ingressi D ed E è il seguente: l’ingresso D è l’ingresso di dati, mentre l’ingresso E è l’ingresso di abilitazione. Se E è alto, il latch è abilitato e il valore in D viene trasmesso in uscita. Se E è basso, il latch è disabilitato e il valore in uscita rimane invariato.

### 1.3 Verifica

Il funzionamento del Latch è stato verificato 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 (per  $E = 0$  Q rimane costante e immune a variazioni di D, viceversa per  $E = 1$ ). Gli screen-shot relativi sono riportati in figura 2.

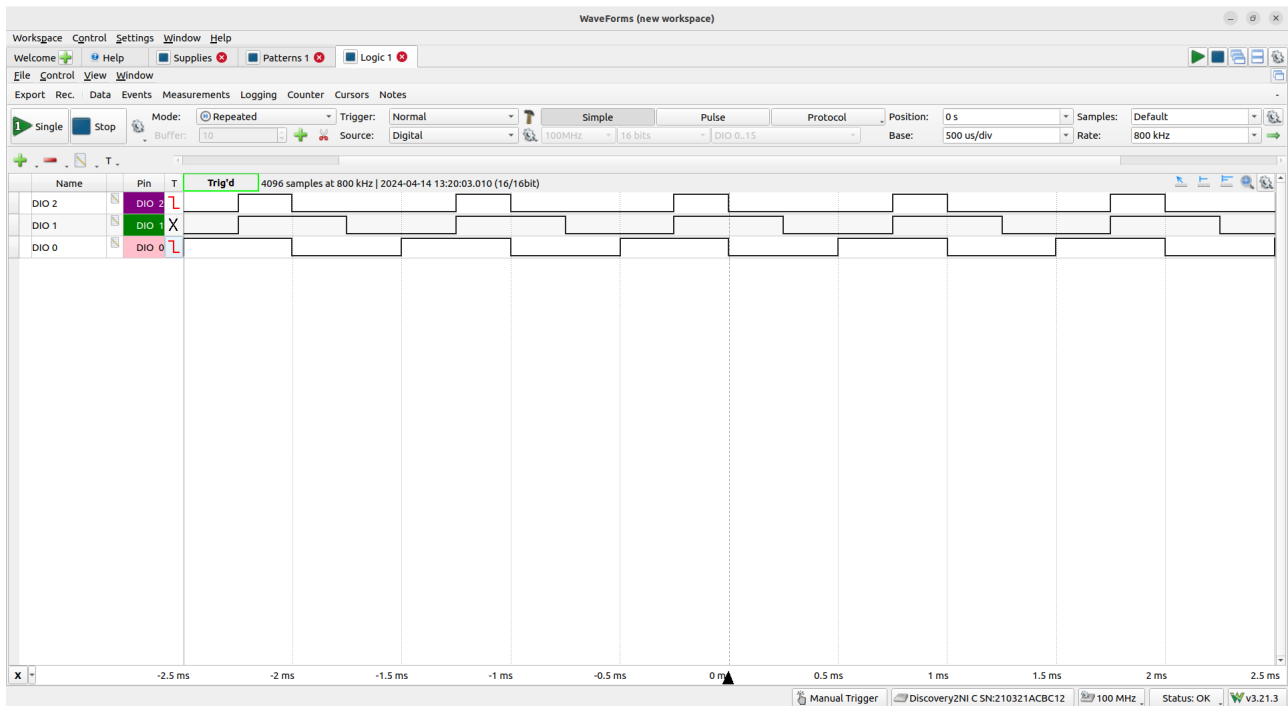


Figura 2: Screen-shot relativi al funzionamento del D-Latch con enable. DIO 0 = D, DIO 1 = E , DIO 2 = Q.

## 2 Shift register con edge-triggered D-Flip Flop

### 2.1 Montaggio

Il circuito è stato montato come da schema, utilizzando 2 integrati 74LS74 (ciascuno con due FF di tipo D) seguendo lo schema in figura. Gli ingressi di preset di tutti i FF sono stati collegati ad uno StaticIO di tipo “Button” e polarità tale che l’uscita sia 1=released, 0=pressed. L’ingresso D del FF0 è stato collegato ad uno staticIO di tipo Switch in modalità Push-Pull. Il Clock è stato pilotato con un segnale di tipo Clock di Pattern. Ogni uscita dei FF è stata collegata a canali dello StaticIO di tipo LED-software.

### 2.2 Funzionamento e verifica

Il circuito è uno shift register a 4 bit. Il pulsante collegato al preset porta lo shift register nello stato 1111. La commutazione delle uscite è asincrona, ossia avviene indipendentemente dal Clock. Disabilitato il preset, è stato inviato un Clock di bassa frequenza ( 1 Hz). Il funzionamento del circuito è stato verificato controllando l’accensione/spengimento dei LED-software in corrispondenza a successive commutazioni del D-switch: lasciando i preset alti le uscite cambiano progressivamente stato (a partire da  $Q_0$ , verso  $Q_3$ ) ad ogni fronte di salita del Clock in base all’ingresso B. Usando sia il D-switch che il preset, è stato verificato che l’ingresso che guida le uscite è il preset (agendo indipendentemente dal Clock). Collegando l’uscita NOT( $Q_3$ ) all’ingresso D del primo FF (al posto del Switch) e utilizzando un clock ad alta frequenza ( 1 kHz), sui piedini di uscita  $Q[3..0]$  si osservano le forme d’onda in figura 5. Il funzionamento del circuito è il seguente: il clock ad alta frequenza fa sì che tutte uscite cambino stato ad ogni fronte di salita, trasmettendo il valore di NOT( $Q_3$ ) al primo FF, che lo trasmette al secondo FF e così via, ottenendo uno shift register: partendo da una configurazione 1111 è chiaro che si passerà a 0111 e 0011 negli stati successivi, e così via fino ad arrivare a 0000, da cui si riparte ciclicamente.

## 3 Utilizzo di un contatore come divisore di frequenza

### 3.1 Montaggio

Il circuito è stato montato come da schema, utilizzando il contatore a 4 bit sincrono 74LS163. Le 4 uscite sono state collegate ad un bus con altrettanti bit di Logic. Il clock è stato inviato ad una frequenza di circa 10 kHz.

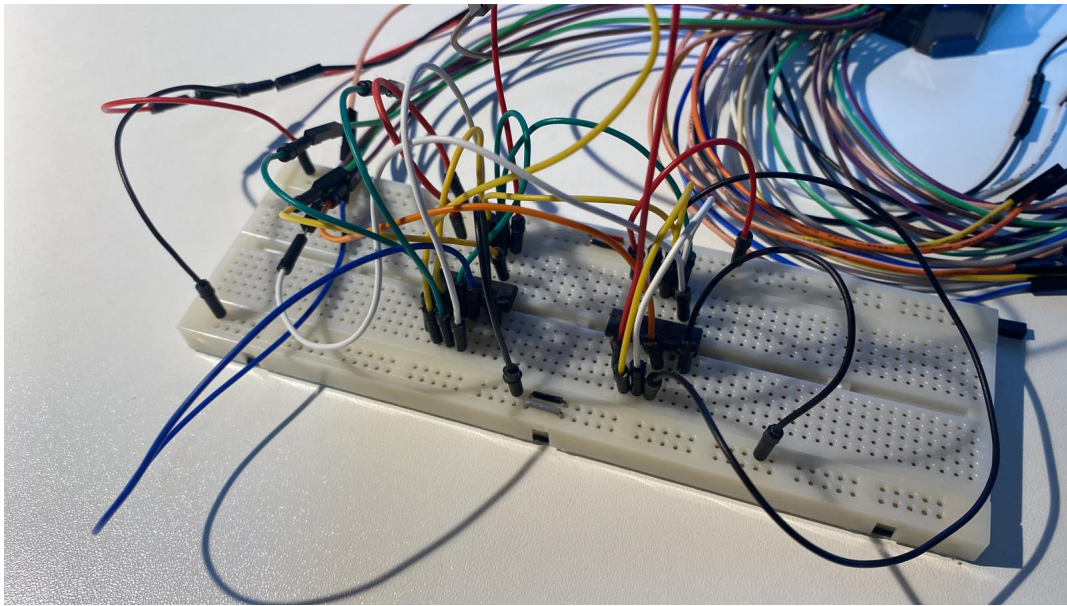


Figura 3: Prima foto del circuito shift register.

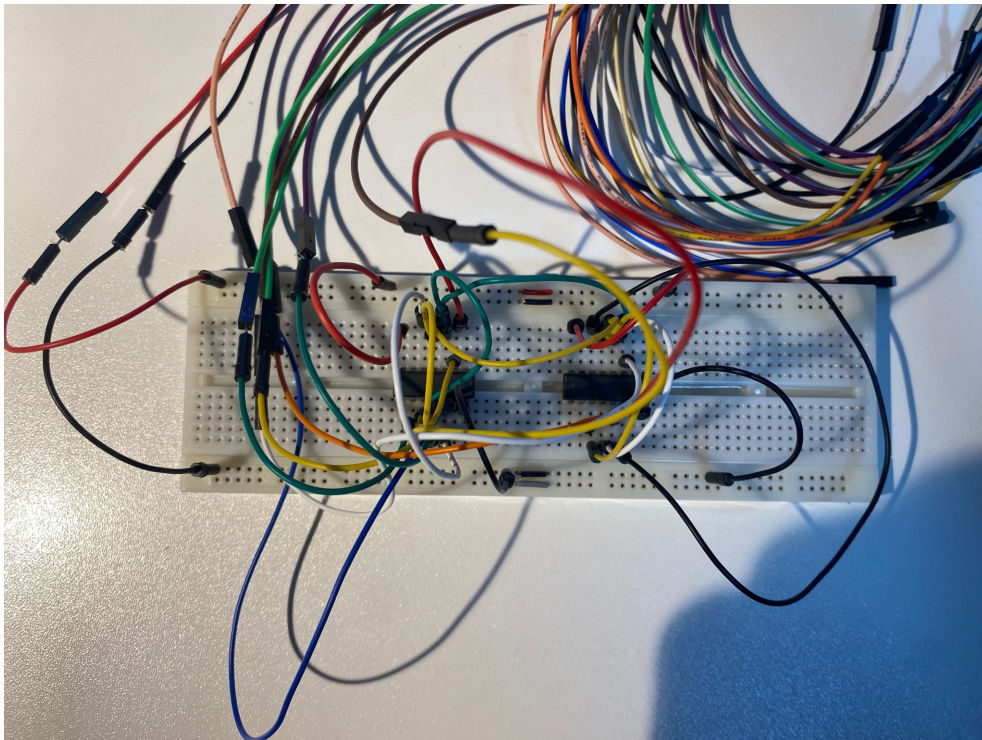


Figura 4: Seconda foto del circuito shift register.



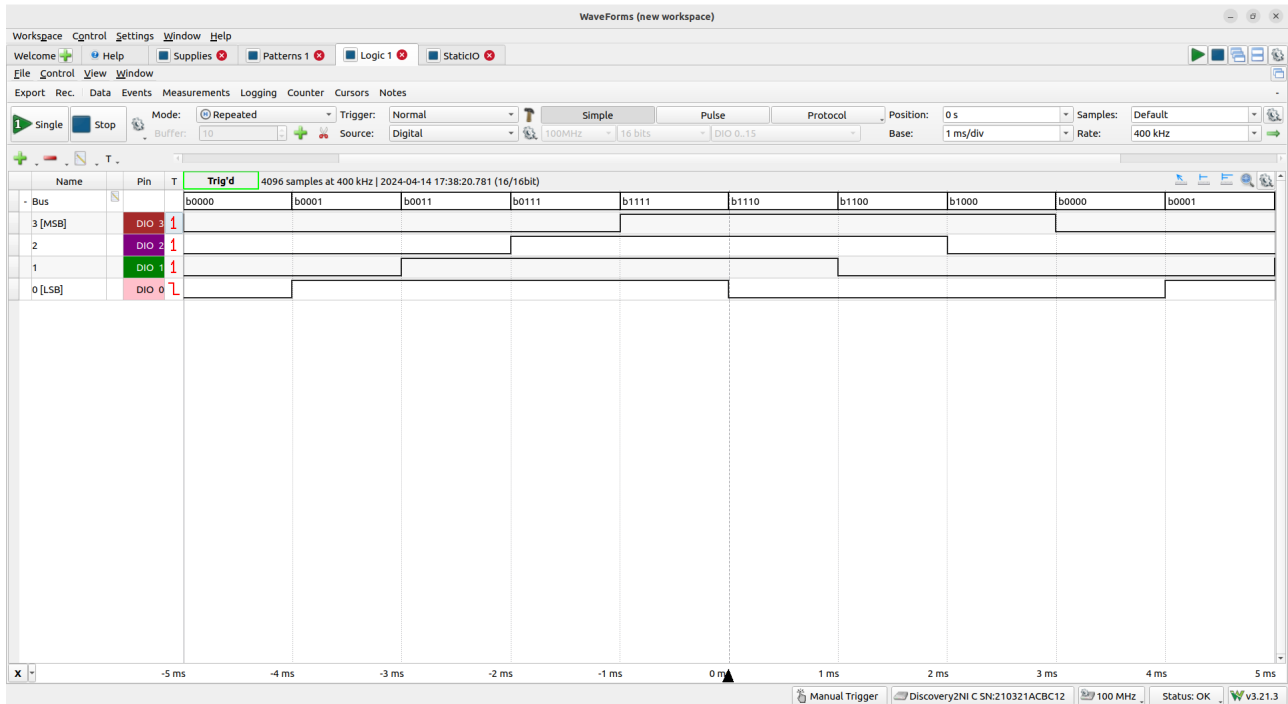


Figura 5: Forme d'onda osservate sui piedini di uscita Q[3..0] del circuito shift register.

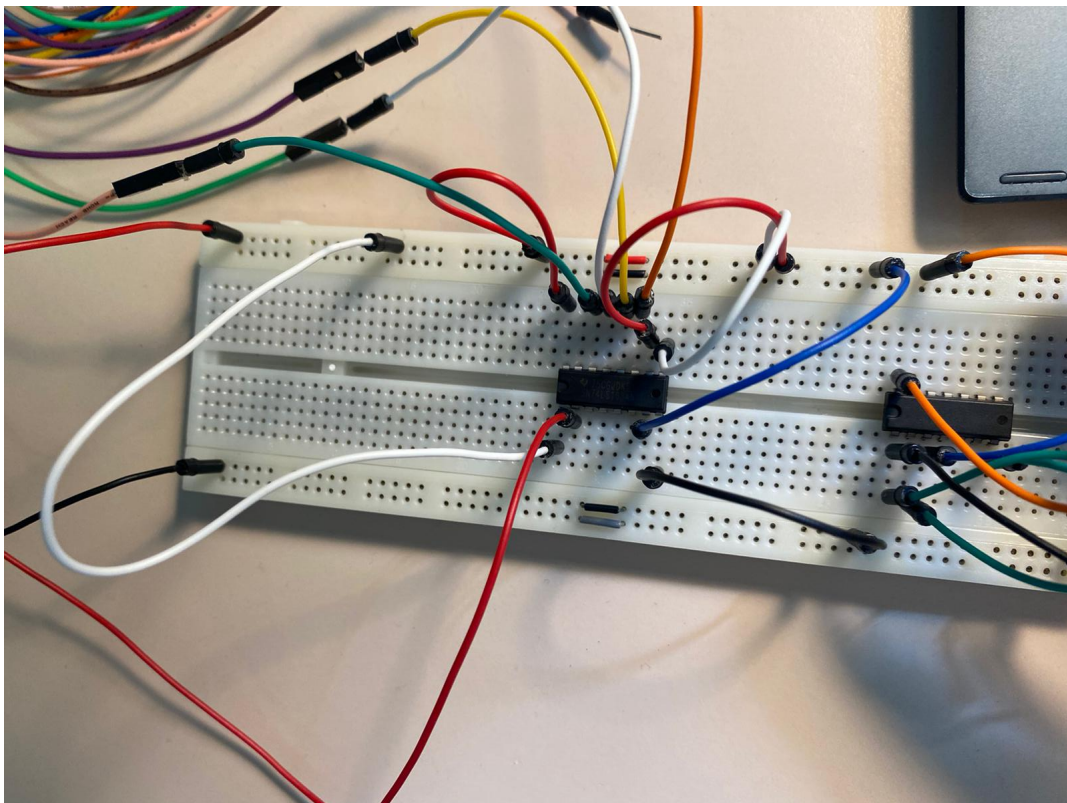


Figura 6: Foto del circuito contatore.

### 3.2 Funzionamento e verifica

Il circuito agisce come contatore: visualizzando il valore del bus si osserva che il contatore conta da 0 a 15 e riparte ciclicamente. Osservando individualmente i segnali Q0, Q1, Q2 e Q3 si verifica che la loro frequenza risulti pari rispettivamente ad  $1/2$ ,  $1/4$ ,  $1/8$ ,  $1/16$  della frequenza di clock. Il comportamento sincrono del contatore è stato verificato in figura 7. Per verificare il comportamento sincrono del contatore, si è visualizzato contemporaneamente il fronte del clock e quello dei singoli bit di uscita alla transizione del contatore  $15 \rightarrow 0$ . Riducendo la larghezza della scala temporale fino a poche decine di ns, si osserva che i bit di uscita cambiano stato in modo sincrono al fronte di discesa del clock (figura 8).

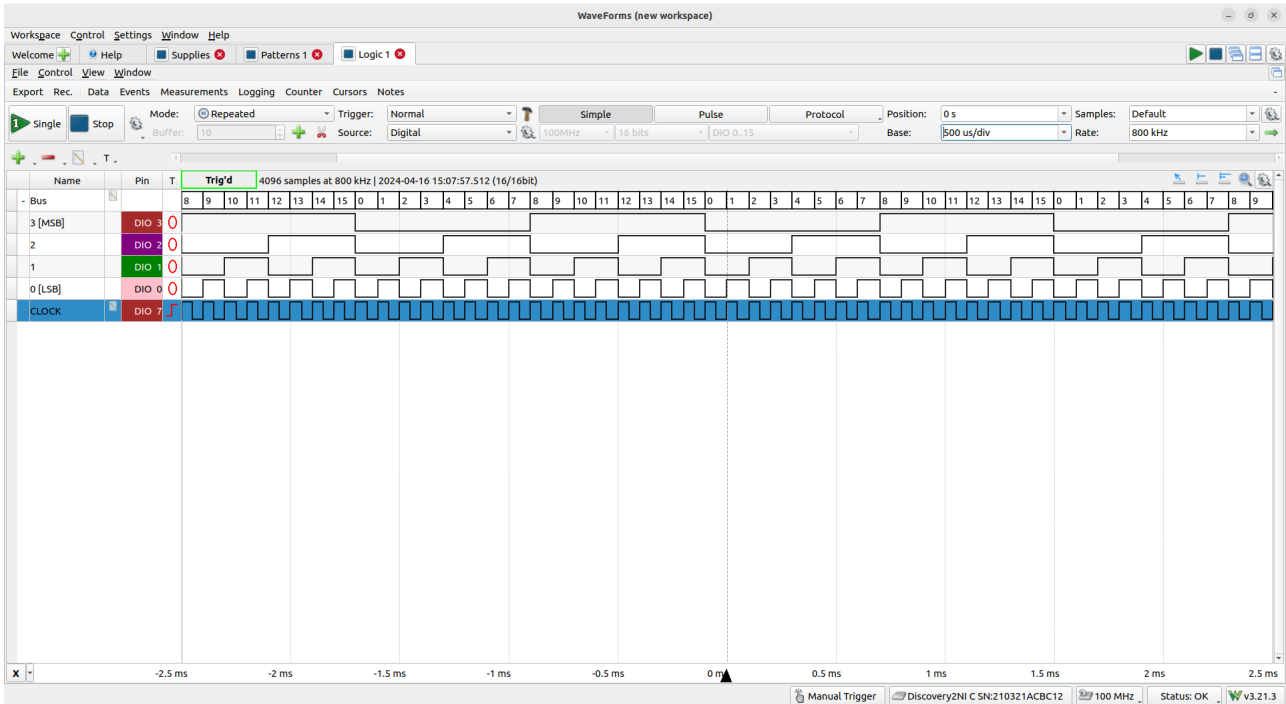


Figura 7: Comportamento sincrono del contatore.

### 3.3 Divisore di frequenza 1/10

Per costruire un divisore di frequenza  $1/10$  si è progettato un circuito che conta 10 stati, in modo da avere un segnale di frequenza  $1/10$  della frequenza di clock utilizzando il load sincrono. Il circuito è stato montato come da schema, utilizzando l'RCO per generare il segnale di Load. Il valore iniziale del contatore è stato caricato mediante Pattern (bus a 4 bit di tipo costante) collegando i canali DIO 8-11 alle linee di ingresso del contatore (figura 10). Il valore iniziale del contatore è stato scelto come 6. Il funzionamento del circuito è il seguente: il contatore conta da 6 a 15 e riparte ciclicamente, generando un segnale di Load che abilita il caricamento.

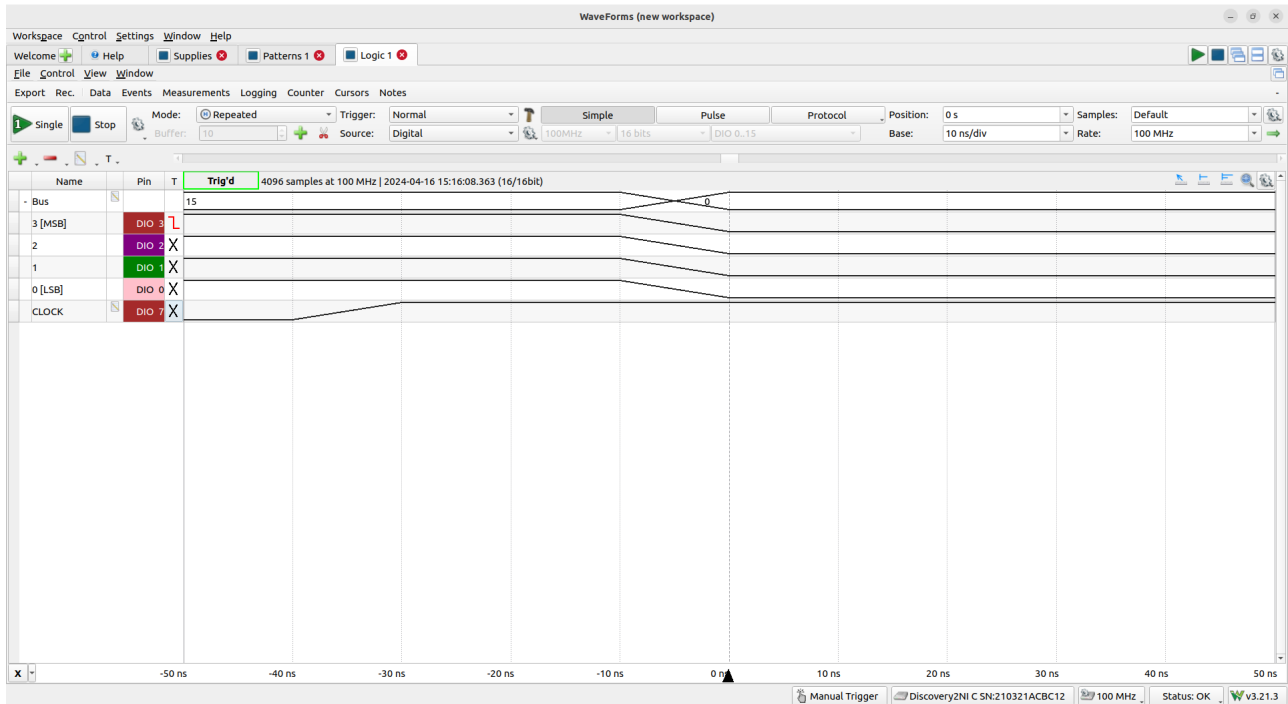


Figura 8: Comportamento sincrono del contatore a scala temporale ridotta.

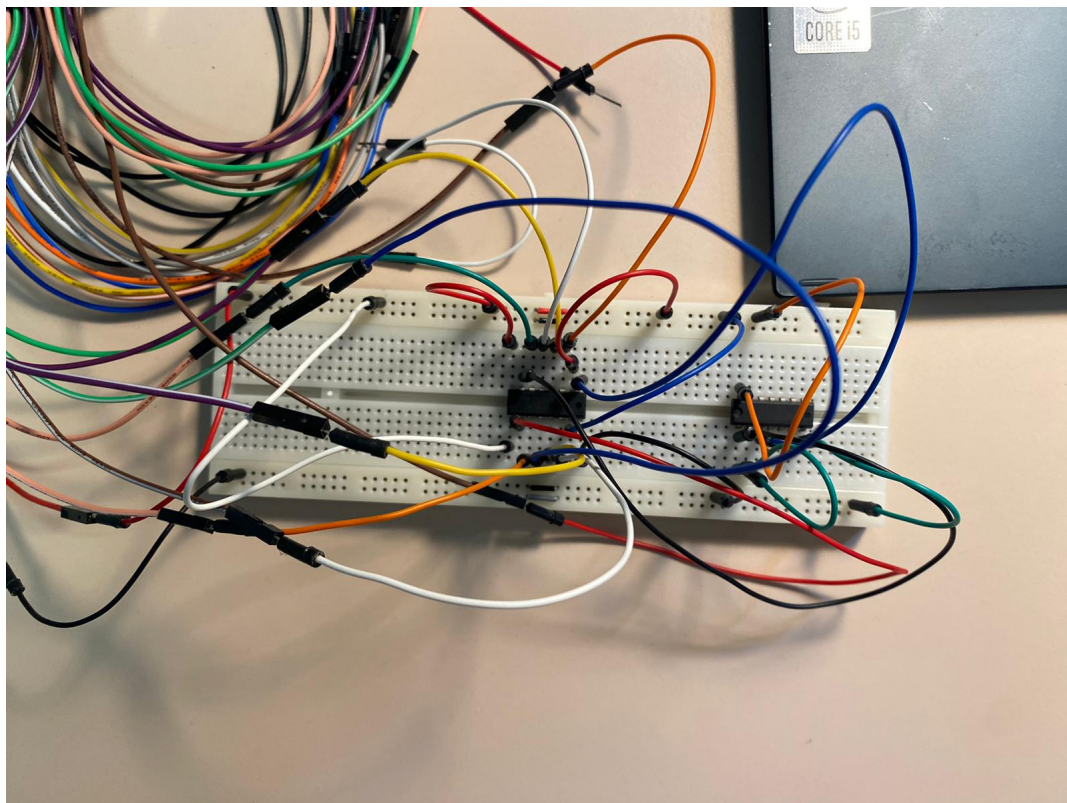


Figura 9: Foto del circuito divisore di frequenza 1/10.

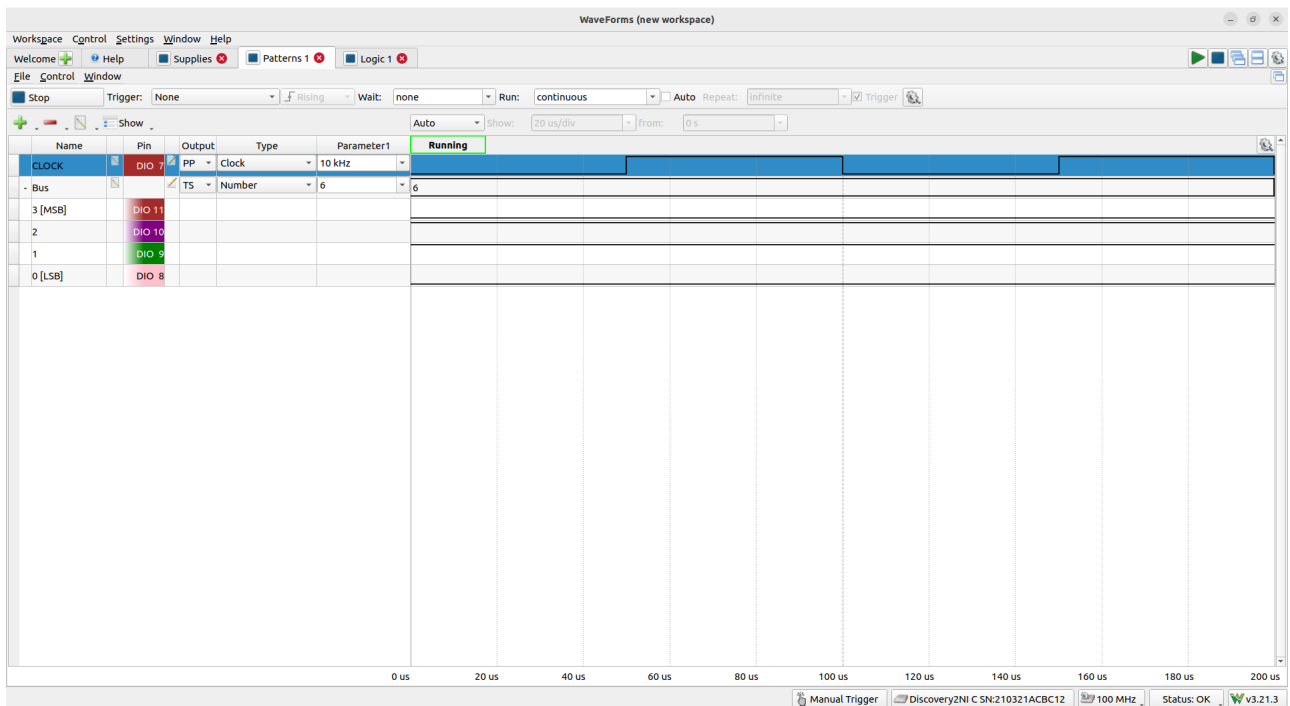


Figura 10: Valori iniziali di ingresso e Clock.

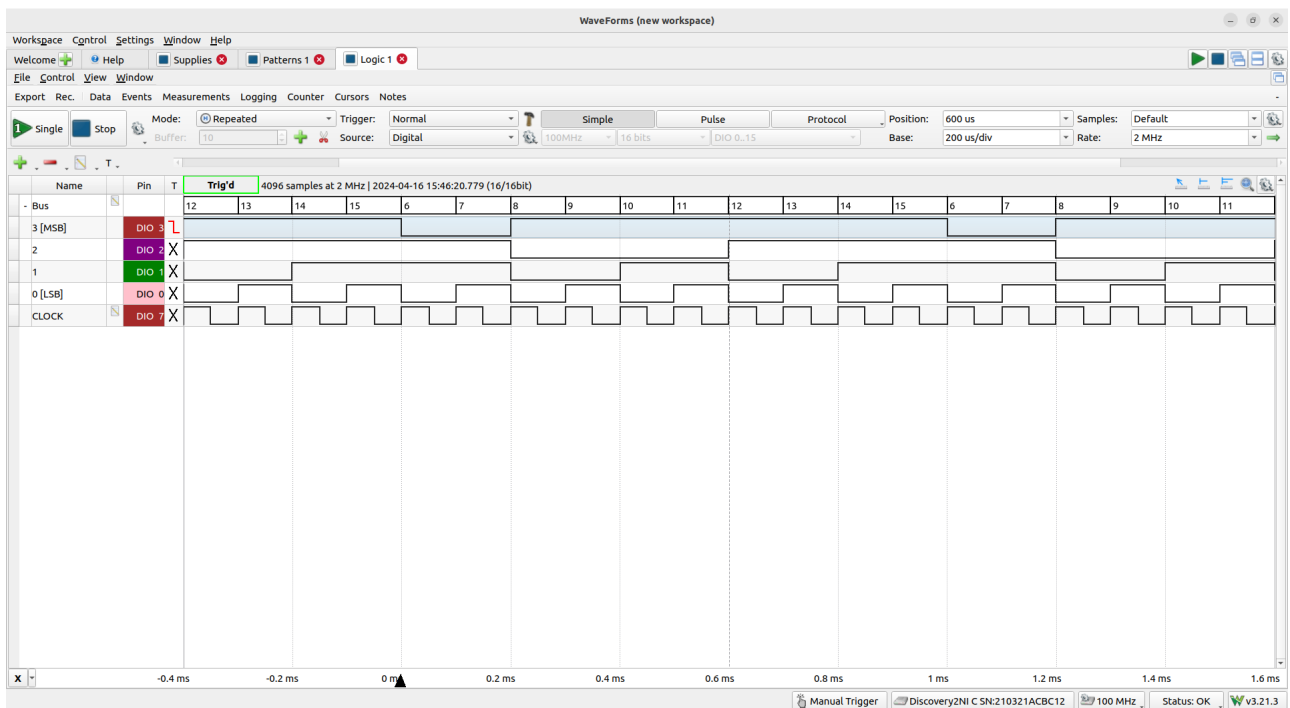


Figura 11: Comportamento del divisore di frequenza 1/10.