

Esercitazione 12 - Gruppo AC

Flip-flop e contatori

Marco Cilibrasi

Tommaso Pajero

22 aprile 2015

L'esercitazione consiste nella realizzazione e nella verifica del corretto funzionamento di alcuni semplici circuiti logici sequenziali (*flip-flop D-latch*, divisori di frequenza, *shift register*, generatore di sequenze pseudo-casuali).

1.a-b Flip-Flop D-Latch

Abbiamo montato il circuito in figura 1 usando le porte logiche NAND dei due IC SN74LS00 in dotazione¹, collegando l'ingresso **DATA** all'uscita Y2 del circuito impulsatore basato su Arduino e l'ingresso **ENABLE** ad uno switch la cui seconda estremità è stata messa a terra².

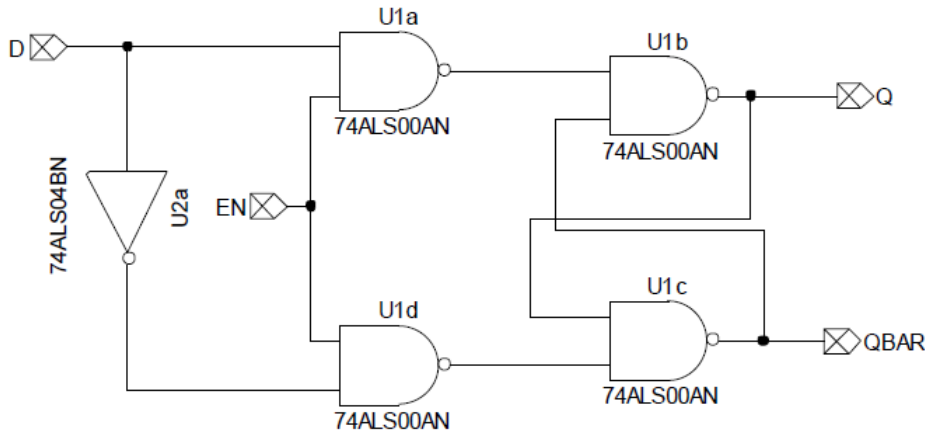


Figura 1: *Flip-flop D-Latch* montato al punto 1.

Per verificare il corretto funzionamento del circuito abbiamo inviato un'onda quadra all'ingresso D. Mantenendo l'ENABLE basso abbiamo ottenuto alle uscite Q e \bar{Q} i segnali mostrati in figura 2. Le forme d'onda confermano ciò che prevede una veloce analisi del circuito. Infatti, se EN è basso le uscite di U1a e U1d sono alte indipendentemente dal valore di D, per cui le uscite di U1b e U1c rimangono invariate nel tempo³. Mantenendo l'ENABLE alto abbiamo invece ottenuto alle uscite Q e \bar{Q} i segnali mostrati in figura 3, in accordo con la previsione teorica ($Q = D$, $\bar{Q} = \bar{D}$). Infatti, se EN è alto all'uscita di U1a troviamo il segnale \bar{D} e all'uscita di U1d il segnale D . Considerando i due casi $D = 0, 1$, si ottiene facilmente la predizione citata.

Abbiamo quindi misurato il ritardo tra il segnale in ingresso D e il segnale in uscita Q , sia in salita che in discesa⁴, ottenendo i seguenti risultati:

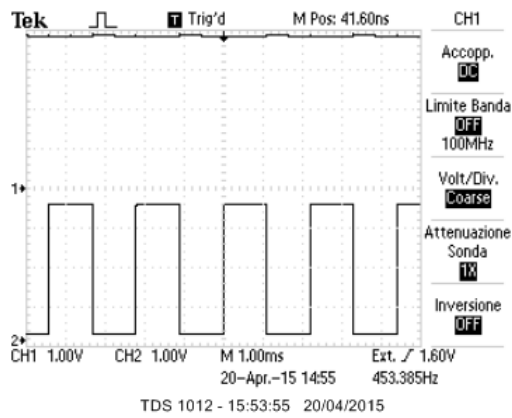
- ritardo per il segnale in salita (osservabile in figura 4): $t_{LH} = 29.2 \pm 0.6$ ns
- ritardo per il segnale in discesa: $t_{HL} = 36.0 \pm 0.6$ ns

¹La porta NOT è stata implementata mandando lo stesso ingresso alle due entrate di una porta NAND.

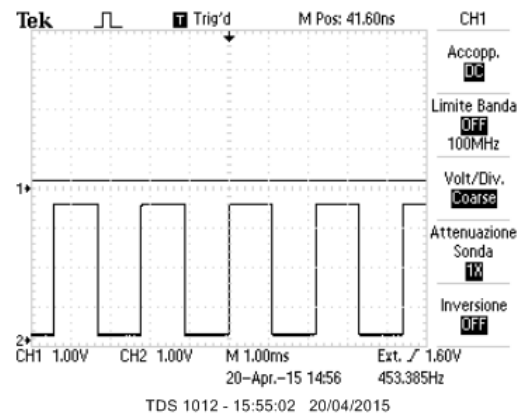
²Quando lo switch è ON allora EN è a terra quindi FALSE, quando lo switch è OFF allora EN è scollegato, quindi TRUE (stiamo usando porte TTL).

³Mantengono il valore che hanno al momento in cui ENABLE commuta da alto a basso. Ciò è stato verificato direttamente mandando impulsi a una frequenza vicina all'Hz, in modo da poter determinare il valore di D al momento della commutazione di ENABLE.

⁴Abbiamo usato la convenzione secondo cui il ritardo è l'intervallo di tempo tra il momento in cui il segnale in ingresso raggiunge la metà tra il suo valore HIGH e il suo valore LOW e il momento in cui il segnale in uscita raggiunge la metà tra il suo valore HIGH e il suo valore LOW.

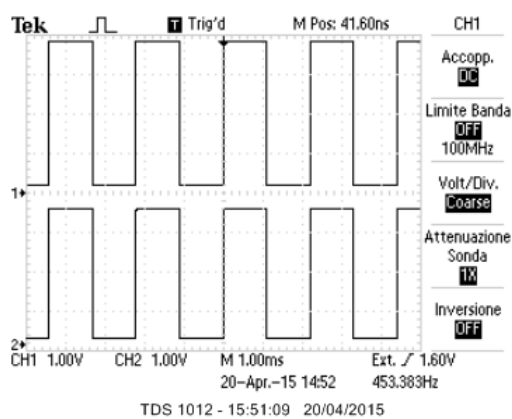


(a) Q .

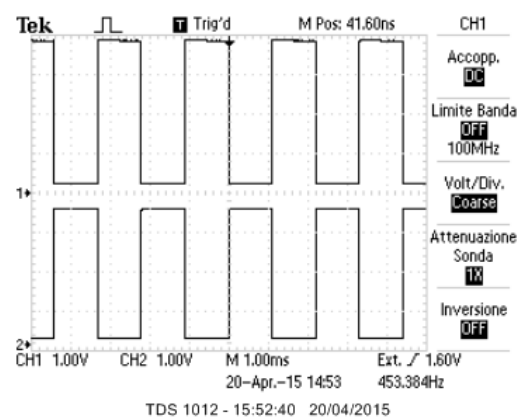


(b) \bar{Q} .

Figura 2: Segnale alle uscite Q e \bar{Q} (CH1) e segnale D in ingresso (CH2) con ENABLE basso.



(a) Q .



(b) \bar{Q} .

Figura 3: Segnale alle uscite Q e \bar{Q} (CH1) e segnale D in ingresso (CH2) con ENABLE alto.

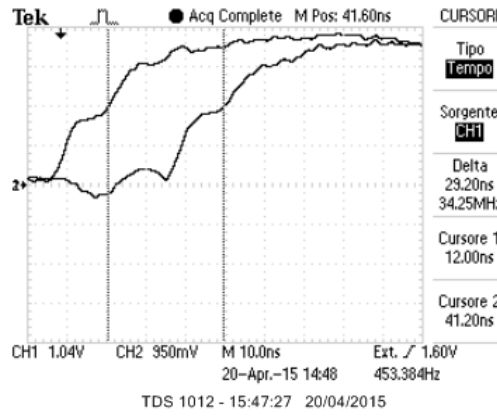


Figura 4: Misura del ritardo tra il segnale D e il segnale \bar{Q} in corrispondenza della transizione LOW-HIGH.

1.c Commenti

Come già accennato, l'ingresso è attivo se ENABLE è alto. La presenza della porta NOT nel circuito fa sì che le uscite di U1a e U1d non siano mai contemporaneamente basse, il che corrisponderebbe a uno stato in cui il *latch* non ha uscita ben determinata⁵.

2. Divisori di frequenza

2.a-b-c Divisore di frequenza per i valori 1/2, 1/4, 1/8, 1/16

Si è montato il contatore binario a quattro *bit* in figura 5, utilizzando i *FF JK* contenuti nell'IC 74LS93 in dotazione. L'ingresso R0 è stato messo a terra, lasciando quello R1 aperto e utilizzando come *clock* la solita uscita del circuito impulsatore Arduino.

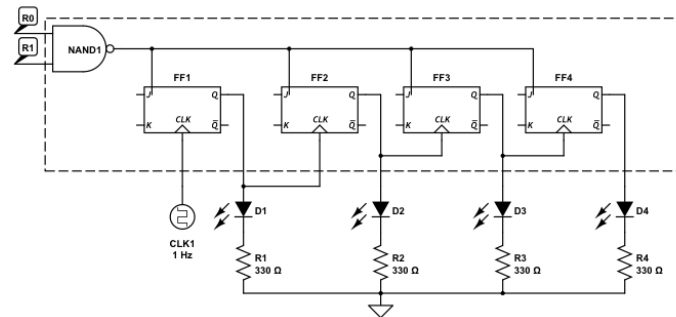


Figura 5: Divisore di frequenza montato al punto 2.

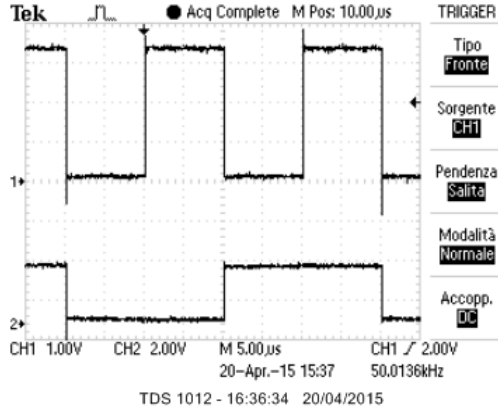
Mandando in ingresso un *clock* di frequenza 1.02 ± 0.01 Hz, si è verificato che sui LED apparisse la codifica binaria dei numeri da 0 a 15, in sequenza crescente e con un avanzamento ogni secondo. In particolare, il *LSB* corrispondeva al diodo D_1 , il *MSB* a D_4 , secondo le attese.

Quindi, si è aumentata la frequenza del *clock* a 50.0 ± 0.05 kHz e si sono osservati all'oscilloscopio i segnali in uscita ai quattro *flip-flop*, riportati in figura 6. Come atteso, ai piedini Q_1 , Q_2 , Q_3 e Q_4 si sono visualizzate onde quadre di frequenza rispettivamente 1/2, 1/4, 1/8 e 1/16 di quella del *clock*. Infine, si è misurato il ritardo delle transizioni dei segnali su tali piedini rispetto a quelle del *clock*, ottenendo i risultati riportati in tabella 1⁶.

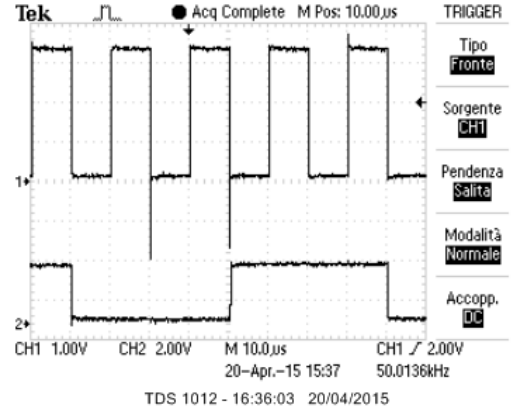
Come è logico aspettarsi, i segnali in uscita dagli *slave* dei *flip-flop* sono in ritardo rispetto al *clock*. In ogni *flip-flop*, tale ritardo è dovuto al tempo finito di commutazione delle porte integrate in esso; poiché l'uscita di

⁵L'uscita dipenderebbe dalla transizione dallo stato precedente a quello in cui U1a e U1d sono entrambi bassi (che tra l'altro, nel caso '00-11', è non deterministica).

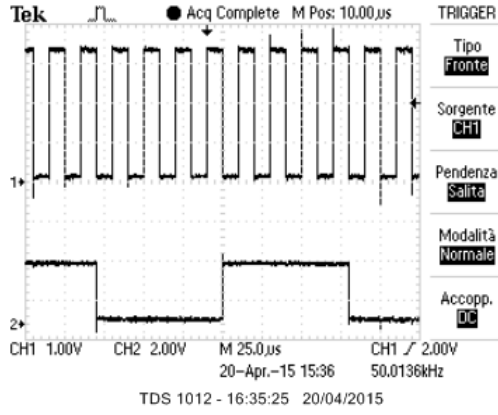
⁶Le misure sono state effettuate in maniera analoga a quanto fatto al punto 1.



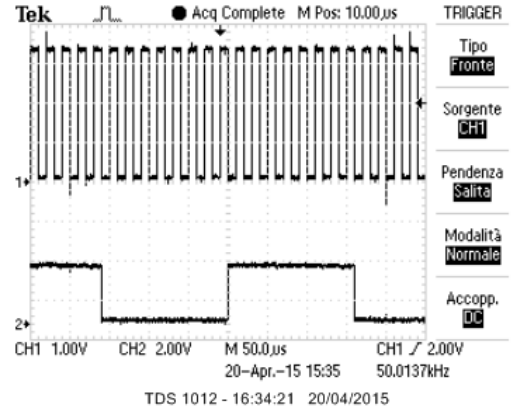
(a) Q_1 : divisore di frequenza di fattore 1/2.



(b) Q_2 : divisore di frequenza di fattore 1/4.



(c) Q_3 : divisore di frequenza di fattore 1/8.



(d) Q_4 : divisore di frequenza di fattore 1/16.

Figura 6: Segnali in uscita ai quattro *flip-flop* (Ch2) e *clock* (Ch1) per il circuito in figura 5.

Tabella 1: Ritardo rispetto al *clock* dei segnali in uscita ai quattro *FF JK* del circuito in figura 5, per le transizioni *low-high* e *high-low*.

uscita	t_{LH} [ns]	$\sigma_{t_{LH}}$ [ns]	t_{HL} [ns]	$\sigma_{t_{HL}}$ [ns]
Q_1	20.8	0.4	14.2	0.4
Q_2	31.2	0.4	23.8	0.4
Q_3	38.6	0.4	35.0	0.4
Q_4	50.4	0.6	47.2	0.6

un *flip-flop* funge da *clock* per il *flip-flop* successivo, i ritardi si sommano e dunque $t_{LH}^{Q_i} > t_{LH}^{Q_j}$ se $i > j$ ⁷. Si nota per inciso che, come dichiarato dal *datasheet*, i t_{LH} sono minori dei t_{HL} .

2.c-bis Funzionamento del circuito

Il circuito sfrutta il fatto che i *FF* – *JK* contenuti nell'IC SN74LS93 hanno tutti gli ingressi J e K scollegati, dunque a valore alto (l'IC appartiene alla famiglia TTL). Questo fa sì che l'uscita di ciascun *flip-flop* cambi a ogni colpo di *clock*⁸. La configurazione particolare del circuito per cui l'uscita di ogni *flip-flop* fa da *clock* al successivo e l'analogia del suo funzionamento col codice binario di codifica dei numeri (per cui quando si conta una posizione passa dal valore '0' a '1' solo se quella che la precede passa dal valore '1' a '0'⁹), fa sì che esso possa fungere da contatore binario. Il fatto che esso funzioni pure come divisore di frequenza è una banale conseguenza della forma della codifica binaria dei numeri (per cui la posizione n-sima cambia di valore ogni 2^{n+1} numeri).

2.d Divisore di frequenza di valore 1/10

Si è collegato l'AND delle uscite Q_2 e Q_4 a $R0$ nel circuito in figura 5, dopo aver scollegato quest'ultimo dalla massa¹⁰. In tal modo, si è ottenuto un divisore di frequenza di valore 1/10 sul piedino Q_4 , nonché un contatore asincrono da 0 a 9 (cfr. figura 7). Infatti, il circuito funziona in maniera del tutto analoga al punto precedente, con l'unica eccezione che, non appena i piedini Q_2 e Q_4 assumono contemporaneamente il valore alto (configurazione corrispondente al numero 10), l'uscita del loro AND commuta. Poiché tale uscita è collegata a $R0$, in tal caso avviene quasi istantaneamente un *reset* del contatore e la conta ricomincia a partire da 0. Il piccolo ritardo nel passaggio dalla cifra 9 a 0 è dovuto a due fattori:

- il tempo finito di commutazione delle due porte NAND che implementano la funzione AND utilizzata;
- il tempo finito necessario per il *reset* del contatore.

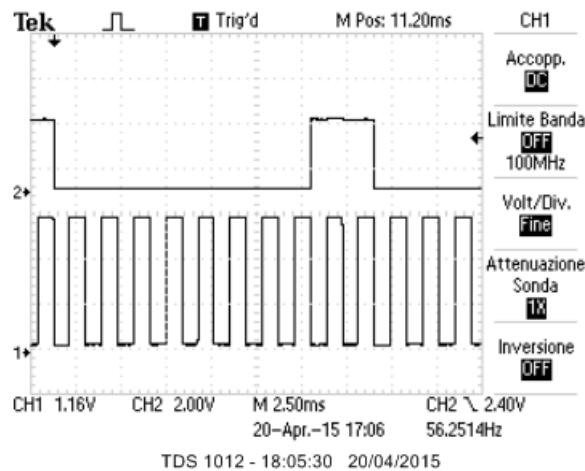


Figura 7: Segnale sul piedino Q_4 per il divisore di frequenza montato al punto 2.d.

3. Shift register con D-Latch

Abbiamo montato uno shift register a 4 bit utilizzando 2 integrati SN74LS74 (figura 8), collegando gli ingressi di Preset ad un pulsante verso massa e le 4 uscite ai LED, messi a terra attraverso resistenze da 330 Ω . L'ingresso Data del primo flip-flop è stato collegato a terra tramite DIP switch, mentre tutte le entrate scollegate (estremità non collegate a terra del pulsante e dello switch, entrate CLEAR) sono state collegate alla tensione di alimentazione tramite resistenze di PULL-UP da 1 k Ω per evitare malfunzionamenti. Abbiamo

⁷E analogamente per t_{HL} . Il fatto che non tutti i ritardi relativi tra due *flip-flop* consecutivi siano uguali dipende verosimilmente dai diversi tempi di ritardo delle porte integrate nei *flip-flop* (diversità questa dovuta all'imprecisione nella loro fabbricazione: il costruttore non garantisce tempi costanti). Non si è invece riusciti a spiegare perché il primo *flip-flop* accumuli un ritardo relativo allo stadio precedente ben maggiore rispetto agli altri.

⁸In particolare, poiché i *flip-flop* sono del tipo *master/slave*, sul relativo fronte in discesa.

⁹Fatto che corrisponde, nell'analogia col circuito, alla discesa del fronte del *clock* che è fornito dal *flip-flop* precedente.

¹⁰La porta AND è stata realizzata tramite due porte NAND integrate nell'IC SN74LS00 in dotazione.

infine inviato un clock a bassa frequenza al circuito per mezzo dell'impulsatore Arduino e abbiamo verificato il corretto comportamento del circuito osservando le luci dei led prima e facendo delle misure all'oscilloscopio successivamente¹¹.

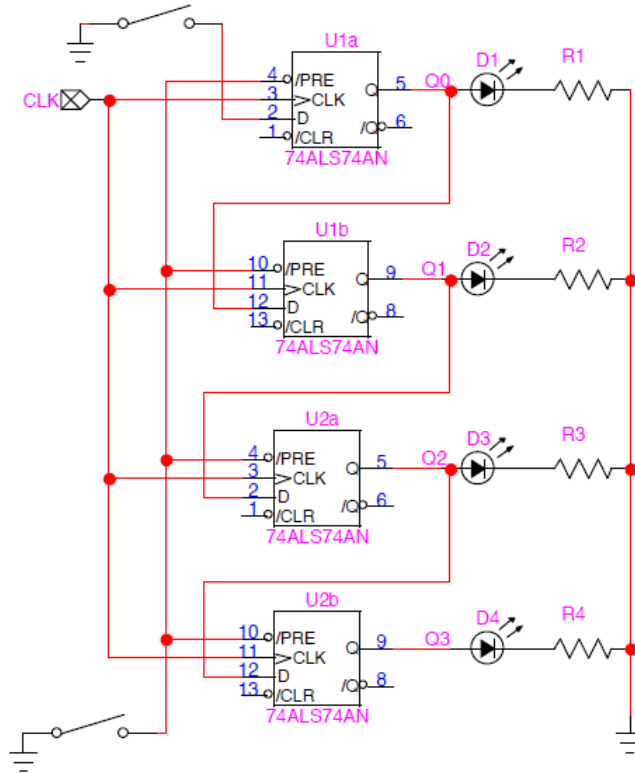


Figura 8: Shift register a 4 bit.

Inizialmente si è mantenuta l'entrata D a terra e si è verificato che, premendo il pulsante di PRESET (cioè mettendo il PRESET a terra, condizione che forza tutte le uscite ad essere alte, indipendentemente dallo stato del clock), i LED si accendessero tutti e iniziassero a scorrere non appena si cessava la pressione, da Q_0 verso Q_3 e con frequenza pari a quella del clock (transizioni poste in corrispondenza dei fronti in salita). Ciò è dovuto al fatto che, fintanto che l'uscita $Q_i = D_{i+1}$ si mantiene alta (si sta supponendo che l'impulso duri più di un periodo di clock, se no non è detto che l'equazione valga¹²), l'uscita Q_{i+1} resta alta a sua volta¹³. Quando l'uscita Q_i si abbassa, invece, ciò fa sì che anche il valore di Q_{i+1} diventi basso al successivo colpo di clock. Questo fenomeno è ben visibile in figura 9: l'abbassamento dei vari segnali avviene a intervalli di tempo regolari, pari ad un periodo del clock.

Successivamente abbiamo invertito per comodità lo SWITCH con il pulsante in modo da riuscire ad inviare brevi impulsi positivi all'ingresso del canale D del primo FF¹⁴ (lasciando il PRESET sempre alto). In particolare, si è osservato che:

- se in corrispondenza del fronte in salita del clock il segnale in ingresso era basso, allora non vi era alcuna transizione sul canale Q_0 (figura 10¹⁵);
- viceversa, se l'impulso inviato era da tale che il segnale D fosse alto in corrispondenza del fronte in salita del clock, vi era una transizione sul canale Q_0 , che durava fino al colpo di clock successivo alla discesa di D (figura 11); a quel punto, il bit iniziava a scorrere col meccanismo descritto prima.

In maniera più qualitativa, si è infine osservato che ogni volta che si pigiava il pulsante per un tempo sufficientemente lungo, il bit Q_0 si illuminava e quindi iniziava a scorrere, secondo le attese.

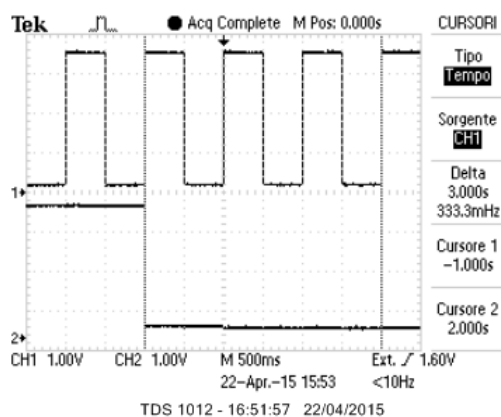
¹¹Sempre a bassa frequenza, con l'acquisizione di una singola schermata dell'oscilloscopio.

¹²Il FF-D-Latch, infatti, è sensibile solo al valore dell'ingresso D in corrispondenza del fronte in salita del clock.

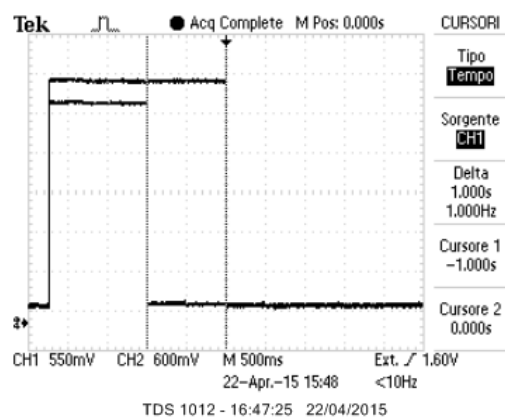
¹³Se invece Q_{i+1} in precedenza era bassa (come accade talvolta nel prosieguo dell'esercitazione), sale al colpo di clock (fronte in salita) successivo.

¹⁴Per fare ciò abbiamo usato i piedini del pulsante che aprono il circuito quando esso viene premuto.

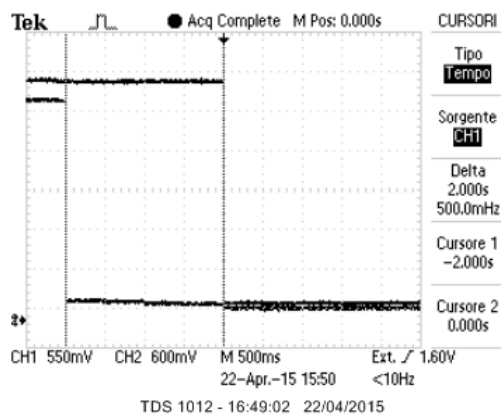
¹⁵Nell'immagine sono stati evidenziati 3 fronti in salita del clock tramite i due cursori e la freccia che indica l'origine dei tempi.



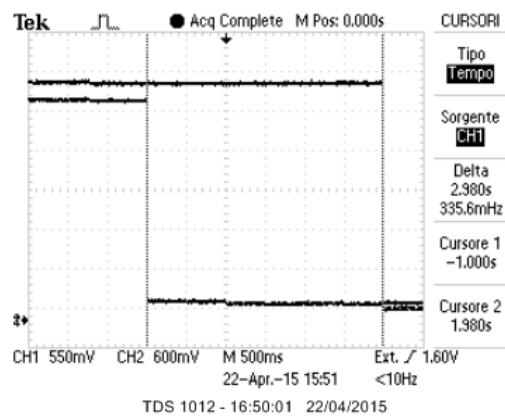
(a) Segnale del clock (CH1) e segnale in uscita da Q0 (CH2). I cursori misurano la durata di 3 periodi del clock.



(b) Segnale in uscita da Q1 (CH1) e segnale in uscita da Q0 (CH2). I cursori misurano l'intervallo di tempo tra la transizione di Q0 e la transizione di Q1.



(c) Segnale in uscita da Q2 (CH1) e segnale in uscita da Q0 (CH2). I cursori misurano l'intervallo di tempo tra la transizione di Q0 e la transizione di Q2.



(d) Segnale in uscita da Q3 (CH1) e segnale in uscita da Q0 (CH2). I cursori misurano l'intervallo di tempo tra la transizione di Q0 e la transizione di Q3.

Figura 9: Misure sullo shift register.

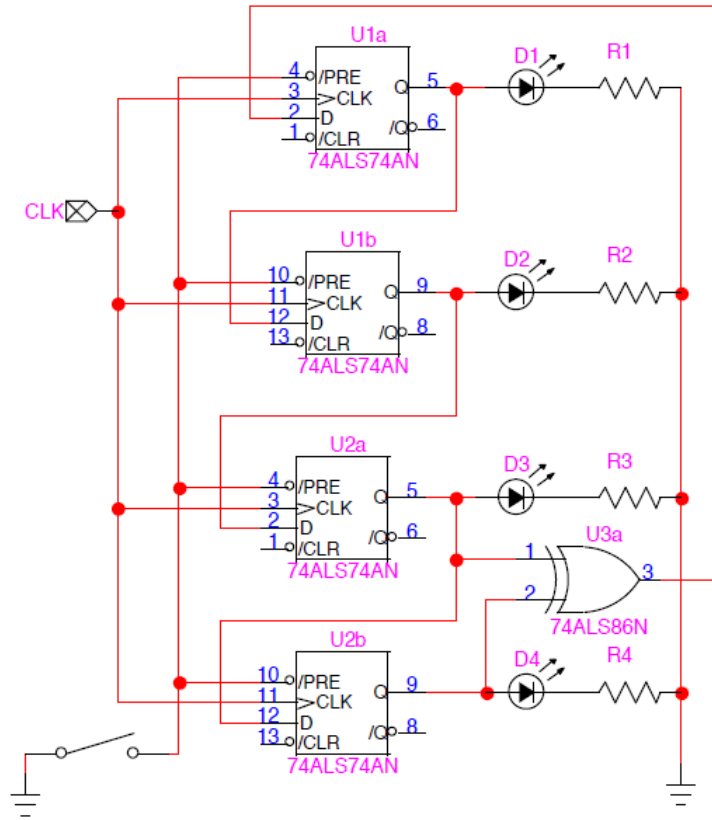


Figura 12: Generatore di sequenze pseudo-casuali.

Tabella 2: Sequenza di numeri pseudo-casuali generata dal tap su Q_2 e Q_3 .

Q_0	Q_1	Q_2	Q_3
1	1	1	1
0	1	1	1
0	0	1	1
0	0	0	1
1	0	0	0
0	1	0	0
0	0	1	0
1	0	0	1
1	1	0	0
0	1	1	0
1	0	1	1
0	1	0	1
1	0	1	0
1	1	0	1
1	1	1	0
1	1	1	1

Tabella 3: Periodo delle sequenze di numeri pseudo-casuali generate da tutti i tap possibili, partendo dalla configurazione '1111'.

Entrate XOR	Periodo della sequenza
Q0-Q1	3
Q0-Q2	7
Q0-Q3	15
Q1-Q2	7
Q1-Q3	6
Q2-Q3	15

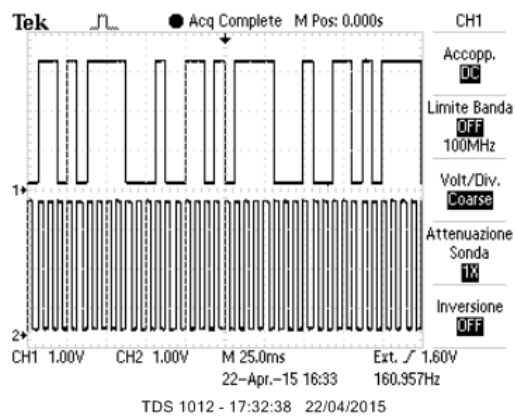


Figura 13: Verifica del comportamento del generatore di sequenze pseudo-casuali. Il clock è su CH2, mentre il segnale Q3 è su CH1.