

Esercitazione 12 - Gruppo AD

Flip-Flop e contatori

Federica Maria Surace

20 aprile 2015

1 Flip-Flop D-Latch

Nella prima parte dell'esercitazione si è montato un flip-flop D-Latch utilizzando le porte NAND di due integrati SN74LS00, come in figura 1. Trattandosi di un D-Latch, il NOT U2a ha qui la funzione di evitare stati proibiti che si ottengono quando gli ingressi Set e Reset sono entrambi 1. Nel corso dell'esercitazione si è usata una tensione di alimentazione di 4.96 ± 0.6 V.

Il corretto funzionamento del circuito è stato verificato mandando in ingresso l'onda quadra fornita da Arduino e collegando all'enable uno switch (a cui è stata collegata anche una resistenza di pull-up). Attivando l'enable (valore alto) si ha che, come atteso, l'uscita del flip-flop segue il segnale in ingresso. (figura 2).

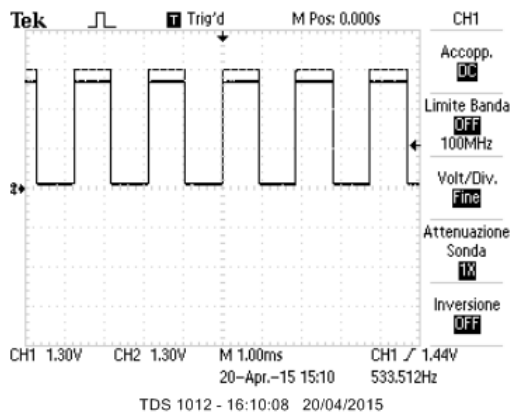


Figura 2: Flip-Flop D-Latch con Enable=1

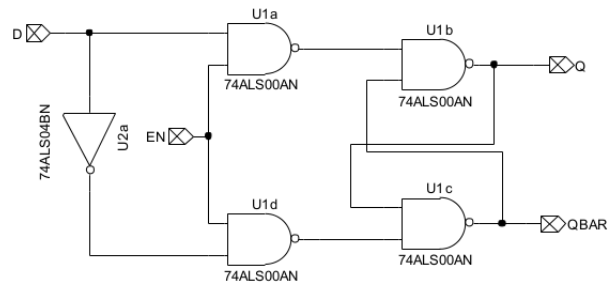
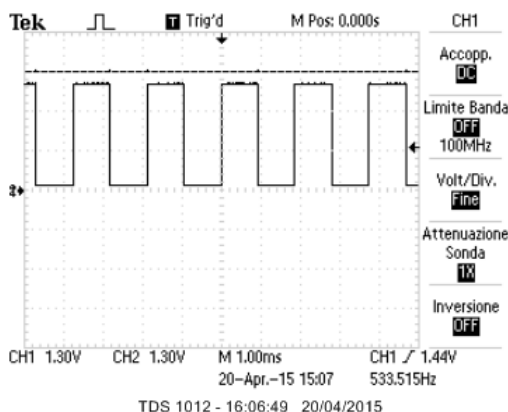


Figura 1: Flip-Flop D-Latch

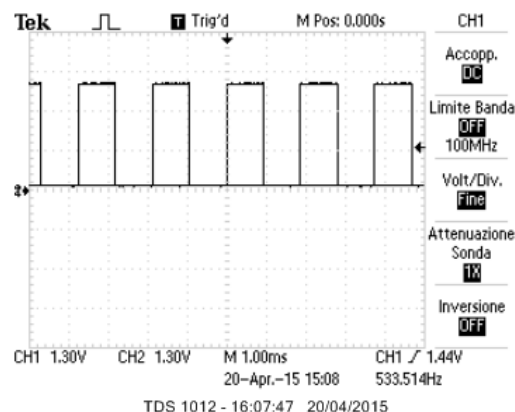
Quando, invece, l'enable diventa basso l'uscita del flip-flop non ha più transizioni da un livello all'altro e rimane nello stato precedente, alto o basso (figura 3). Infatti, quando l'enable è 0, entrambe le uscite dei NAND U1a e U1d sono alte, qualunque sia il valore dell'ingresso Data. Di conseguenza si ha $Q(t+\Delta t) = \overline{\overline{Q(t)}} = Q(t)$ e $\overline{Q(t+\Delta t)} = \overline{\overline{Q(t)}} = Q(t)$ e l'uscita resta costante.

In seguito, è stato misurato il ritardo tra il segnale in ingresso e quello in uscita del flip-flop sia in salita che in discesa (figura 4). I valori sono riportati in tabella 1.

I tempi di ritardo misurati sono in accordo con le attese: infatti, secondo il data sheet le porte NAND



(a) Flip-flop “congelato” nello stato alto



(b) Flip-flop “congelato” nello stato basso

Figura 3: Flip-Flop D-Latch con Enable=0

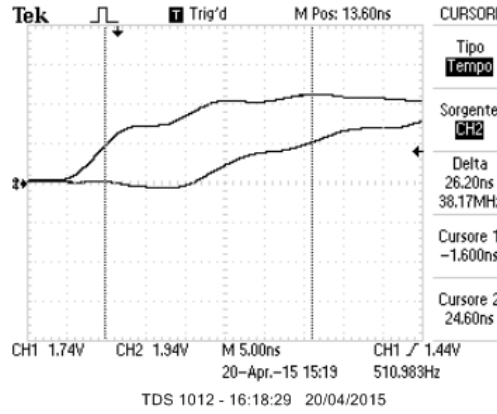


Figura 4: Misura del ritardo del flip-flop in salita

Tabella 1: Tempi di ritardo

$t_{PLH}[ns]$	$t_{PHL}[ns]$
26.2 ± 0.4	36.2 ± 0.4

hanno ritardi tipici $t_{PLH} = 9ns$ (max $15ns$) e $t_{PHL} = 10ns$ (max $15ns$); in salita il tempo di ritardo del flip-flop è quello dovuto alla commutazione di 2 porte in serie (U1a e U1b) da cui $t_{PLH} = 19ns$ (max $30ns$), mentre in discesa bisogna sommare i tempi di 4 porte (U2a, U1d, U1c, U1b)¹ da cui $t_{PHL} = 38ns$ (max $60ns$).

2 Divisori di frequenza

Si è utilizzato l'integrato 74LS93 per realizzare un contatore a 4 bit. Per fare questo si è connessa l'uscita del primo flip flop JK al clock del secondo, mentre i restanti flip flop sono interconnessi già all'interno dell'integrato. Si è inserito un diodo LED tra l'uscita di ogni flip flop e la terra, con una resistenza di limitazione, in modo da visualizzare se l'uscita è alta (LED acceso) o bassa (LED spento). Inoltre si è collegato a terra uno degli ingressi di reset in modo da mettere il circuito in funzione. Il circuito realizzato è mostrato in figura 5.

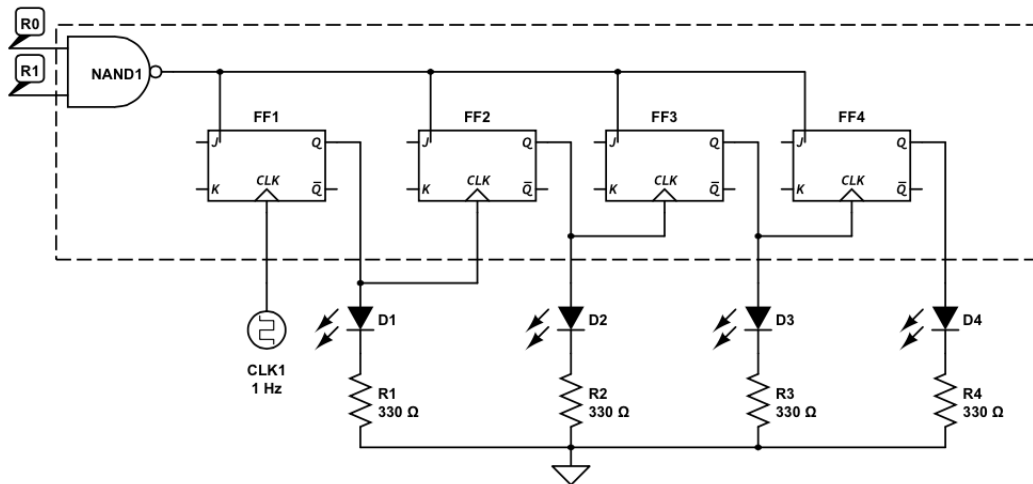


Figura 5: Circuito divisore di frequenza realizzato usando un integrato SN7400N.

Si è usato come clock per il primo flip flop un'onda quadra di frequenza di circa 1 Hz generata con Arduino in modo da poter vedere ad occhio nudo i diversi stati del circuito osservando l'accensione dei LED.

Il funzionamento del circuito è basato sul fatto che i flip flop JK sono utilizzati nello stato di Toggle in quanto gli ingressi J e K sono tutti collegati all'uscita del NAND1 che è alta in quanto uno degli ingressi di reset è

¹Infatti, affinché si abbia commutazione in salita è sufficiente che uno dei due ingressi del NAND U1b sia 0, mentre per la commutazione in discesa è necessario che entrambi gli ingressi del NAND U1b siano 1.

messo a terra. Pertanto ciascun FF oscilla tra lo stato 0 e lo stato 1 in modo sincrono al proprio clock (edge triggered). Dal momento che la transizione avviene solo su un fronte del clock, utilizzando l'uscita di ciascun FF come clock per il successivo si ottiene un divisore di frequenza per potenze di due. Poiché il circuito considerato ha quattro flip flop, si ottengono i divisori per 2, 4, 8 e 16.

Inoltre, interpretando l'uscita del primo FF come bit meno significativo di un numero binario a 4 bit e crescendo in significatività fino all'uscita del quarto FF, si ha un contatore binario. Poiché la transizione è sul fronte in discesa, il conteggio è in ordine crescente, come verificato sperimentalmente.

Successivamente si è usato come clock per il primo flip flop un'onda quadra di frequenza 97.8 kHz. I segnali in uscita dai quattro flip flop hanno frequenza pari a circa 1/2, 1/4, 1/8 e 1/16 della frequenza del clock in ingresso come si può vedere dalla figura 6. Le misure delle frequenze e dei tempi di ritardo tra la transizione del clock e quelle dei FF sono riportate in tabella 2. I risultati ottenuti per i tempi di propagazione indicano che i tempi di propagazione crescono linearmente all'aumentare del numero di porte. Inoltre l'ordine di grandezza dei tempi ottenuti è comparibile con quello misurato al punto 1.

Tabella 2: Divisore di frequenza.

Segnale	t_{PLH} [ns]	t_{PHL} [ns]	f [kHz]
Q_A	56.4 ± 0.8	52.4 ± 0.8	48.9 ± 0.5
Q_B	64.4 ± 0.8	62.0 ± 0.8	24.5 ± 0.2
Q_C	78.0 ± 0.8	72.4 ± 0.8	12.2 ± 0.1
Q_D	84.0 ± 0.8	82.4 ± 0.8	6.11 ± 0.06

Infine si è progettato un circuito che resettì il contatore una volta arrivato a 10, in modo da avere un segnale di frequenza 1/10 della frequenza di clock.

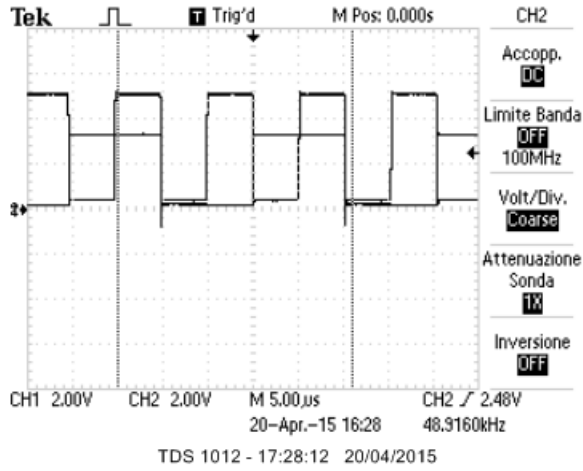
Per sincronizzare il reset si è deciso di utilizzare il D-Latch realizzato nella parte precedente dell'esperienza. Si è preso il NAND dei segnali Q_2 e Q_4 , che dà un segnale basso non appena il contatore arriva a 10. Si è inviato tale segnale in ingresso al D-Latch e si è presa l'uscita negata di quest'ultimo. Tale uscita diviene alta all'arrivo del clock e inviando questo segnale in uno degli ingressi di reset - e tenendo l'altro alto tramite una resistenza di pull-up - si ottiene un reset sincrono. Affinché questo funzioni bisogna porre attenzione al fatto che il contatore è edge-triggered sul fronte di discesa del clock, mentre il D-Latch costruito in precedenza è attivo quando il clock è alto. Pertanto, per avere un reset sincrono a 10 è necessario usare come segnale di clock per il D-Latch il negato del clock del contatore².

In figura 7 sono riportati i segnali Q_A , Q_B , Q_C e Q_D per il circuito realizzato. Come si vede chiaramente dall'immagine (d), il segnale Q_D ha frequenza 1/10 della frequenza di clock. Misurando la frequenza si ha $f = 9.78$ kHz. Anche i segnali Q_B e Q_C hanno tale frequenza, mentre il segnale Q_A ha frequenza 1/2 di quella di clock. I segnali ottenuti sono in perfetto accordo con le sequenze attese, riportate in tabella 3.

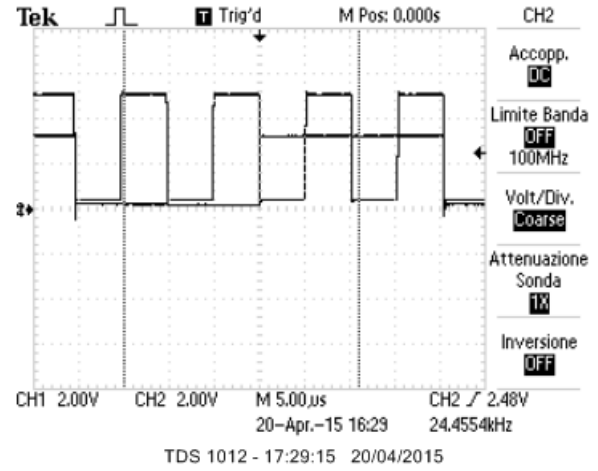
Tabella 3: Sequenza del contatore decadico

A	B	C	D
0	0	0	0
1	0	0	0
0	1	0	0
1	1	0	0
0	0	1	0
1	0	1	0
0	1	1	0
1	1	1	0
0	0	0	1
1	0	0	1

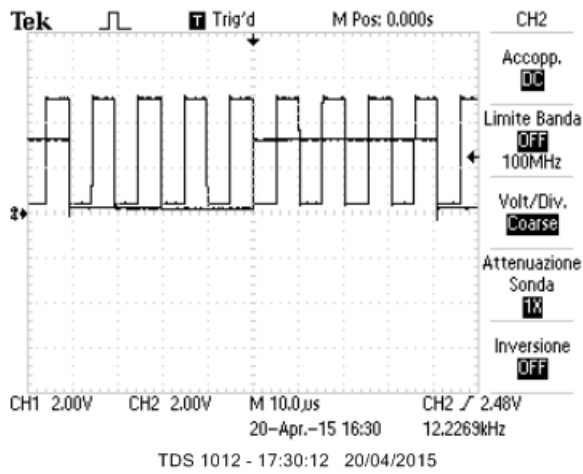
²Altrimenti si ha un errore pari a un semiperiodo del clock.



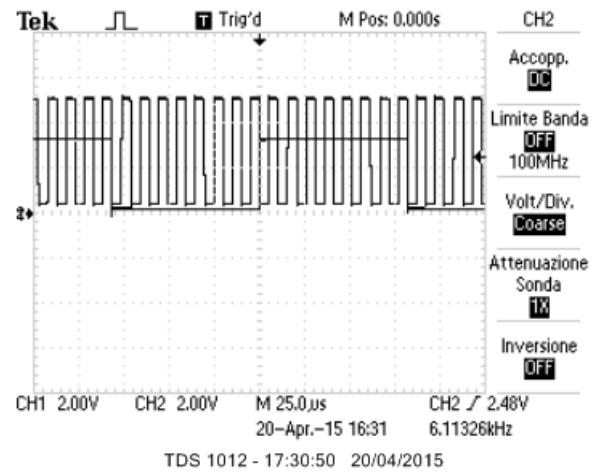
(a) Segnale Q_A



(b) Segnale Q_B

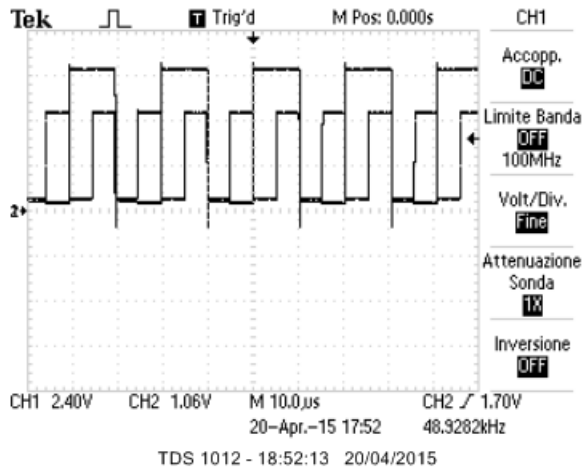


(c) Segnale Q_C

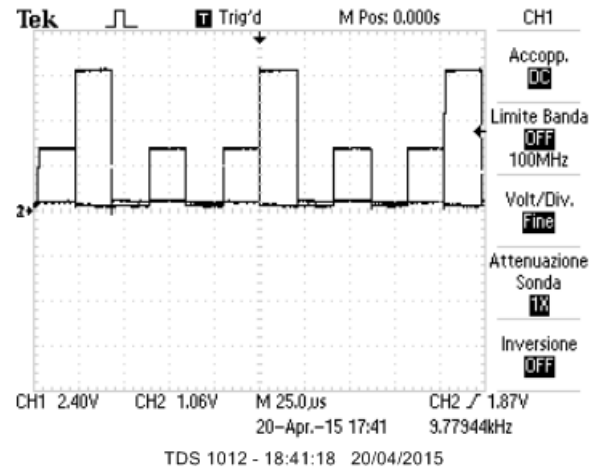


(d) Segnale Q_D

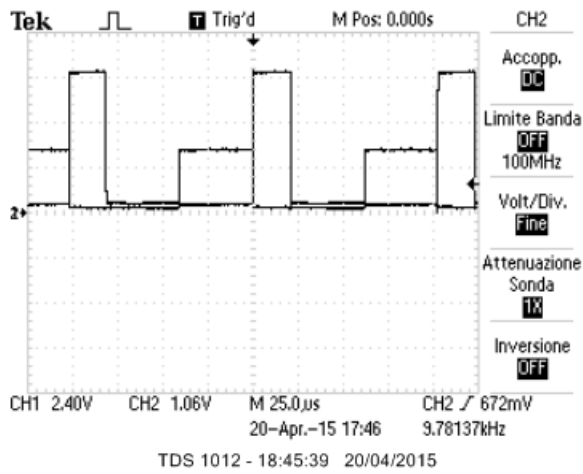
Figura 6: Circuito divisore di frequenza (potenze di 2) realizzato usando un integratore SN7400N. In figura sono rappresentati i segnali Q_A , Q_B , Q_C e Q_D in uscita dai quattro flip flop e il clock inviato in ingresso al primo FF.



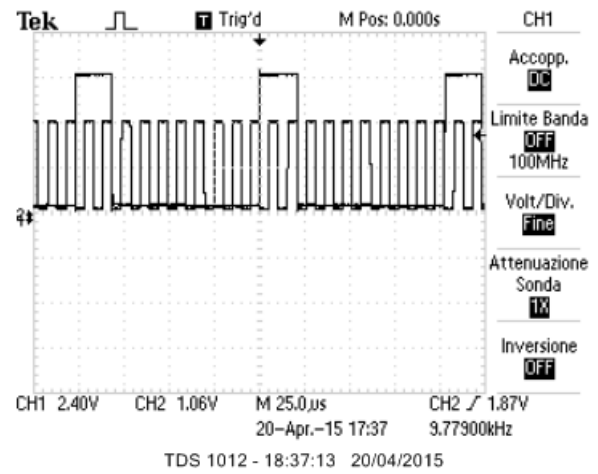
(a) Segnale Q_A e segnale di clock



(b) Segnale Q_B e segnale Q_D



(c) Segnale Q_C e segnale Q_D



(d) Segnale Q_D e segnale di clock

Figura 7: Circuito divisore di frequenza per 10 realizzato usando un integrato SN7400N. In figura sono rappresentati i segnali Q_A , Q_B , Q_C e Q_D in uscita dai quattro flip flop e il clock inviato in ingresso al primo FF.

3 Shift register con D-Latch

Il circuito in figura 8 è stato montato utilizzando 2 integrati 74LS74, con il data del primo flip-flop collegato al DIP switch e gli ingressi di preset collegati verso massa tramite un pulsante. Tutti gli ingressi di clear (negati) sono tenuti alti tramite resistenze di pull-up. Il funzionamento dello shift register è stato verificato tramite dei LED collegati alle uscite Q_0 , Q_1 , Q_2 , Q_3 e usando un clock a bassa frequenza.

Si nota che il DIP switch forza il primo flip-flop a essere alto oppure basso e che, a ogni colpo di clock, lo stato di ogni flip-flop viene trasmesso a quello successivo. Il preset, invece, porta tutte le uscite al valore alto.

4 Generatore di sequenze pseudo-casuali

Il circuito in figura 9 permette di generare sequenze pseudo-casuali a 4 bit, ovvero, partendo da uno stato qualsiasi (diverso da 0000) si ottengono in sequenza tutti gli altri 14 stati secondo la tabella 4.

Il funzionamento è stato verificato tramite i LED e inviando un'onda quadra a bassa frequenza come clock. Nell'immagine 9 si può vedere il funzionamento del circuito con un clock a frequenza più elevata.

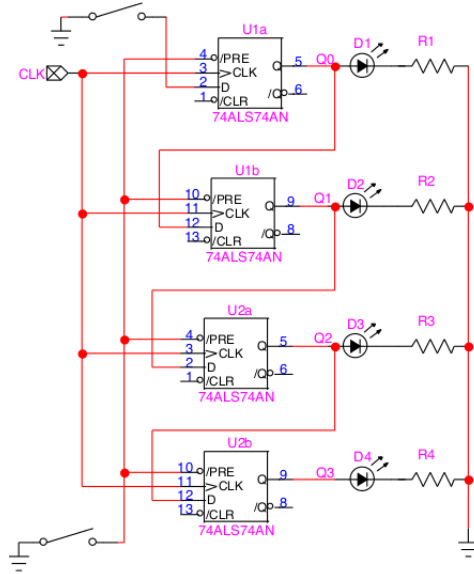
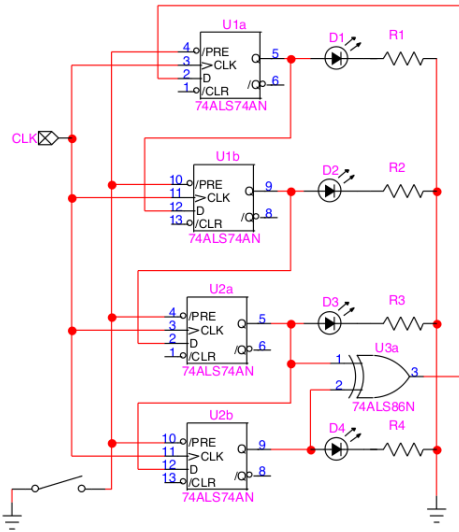
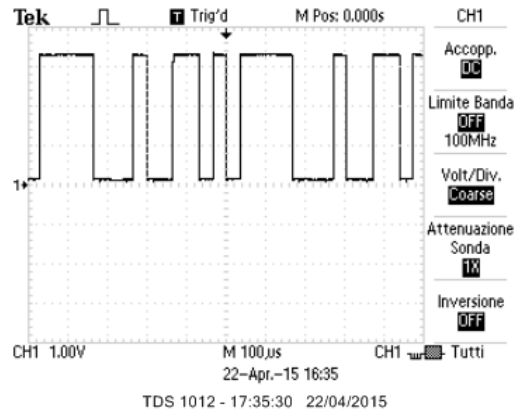


Figura 8: Shift register con D-Latch



(a) Schema circuitale



(b) Uscita Q_0

Figura 9: Generatore di sequenze pseudo-casuali: XOR di Q_2 e Q_3

Provando a utilizzare come tap un'altra coppia di uscite dello shift register si ottengono le sequenze riportate nelle tabelle 5, 6, 7, 8 e 9. Si osserva che solo considerando la coppia Q_0 , Q_3 si ottengono tutti gli stati a 4 bit. Il funzionamento in questo è stato verificato con i LED. Negli altri casi, partendo dallo stato 1111, si ottengono sequenze di periodi rispettivamente 6, 3, 7, 7.

Tabella 4: Sequenza pseudo-casuale: XOR di Q_2 , Q_3

1	1	1	1	0
0	1	1	1	0
0	0	1	1	0
0	0	0	1	1
1	0	0	0	0
0	1	0	0	0
0	0	1	0	1
1	0	0	1	1
1	1	0	0	0
0	1	1	0	1
1	0	1	1	0
0	1	0	1	1
1	0	1	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Tabella 5: Sequenza pseudo-casuale: XOR di Q_0 e Q_3

1	1	1	1	0
0	1	1	1	1
1	0	1	1	0
0	1	0	1	1
1	0	1	0	1
1	1	0	1	0
0	1	1	0	0
0	0	1	1	1
1	0	0	1	0
0	1	0	0	0
0	0	1	0	0
0	0	0	1	1
1	0	0	0	1
1	1	0	0	1
1	1	1	0	1
1	1	1	1	0

Tabella 6: Sequenza pseudo-casuale: XOR di Q_1 e Q_3

1	1	1	1	0
0	1	1	1	0
0	0	1	1	1
1	0	0	1	1
1	1	0	0	1
1	1	1	0	1
1	1	1	1	0

Tabella 7: Sequenza pseudo-casuale: XOR di Q_0 e Q_1

1	1	1	1	0
0	1	1	1	1
1	0	1	1	1
1	1	0	1	0
0	1	1	0	1
1	0	1	1	1
1	1	0	1	0

Tabella 8: Sequenza pseudo-casuale: XOR di Q_0 e Q_2

1	1	1	1	0
0	1	1	1	1
1	0	1	1	0
0	1	0	1	0
0	0	1	0	1
1	0	0	1	1
1	1	0	0	1
1	1	1	0	0
0	1	1	1	1

Tabella 9: Sequenza pseudo-casuale: XOR di Q_1 e Q_2

1	1	1	1	0
0	1	1	1	0
0	0	1	1	1
1	0	0	1	0
0	1	0	0	1
1	0	1	0	1
1	1	0	1	1
1	1	1	0	0
0	1	1	1	0