Laboratorio di Architetture degli Elaboratori I
Corso di Laurea in Informatica, A.A. 2024-2025
Università degli Studi di Milano

Contatori e generatore random

Esercizio 1

- Si realizzi un contatore ciclico in modulo n=8:
  - Il contatore va da 0 a n-1 incrementando il suo valore di 1 ad ogni ciclo di clock
  - Il valore successivo a n-1 nella sequenza è 0

Esercizio 1

- Sintetizziamo la rete combinatoria “add mod 8” che, dato lo stato del contatore al tempo t

$$s(t) = c$$

determini lo stato del contatore al tempo t+1:

$$s(t+1) = c + 1 \pmod 8$$

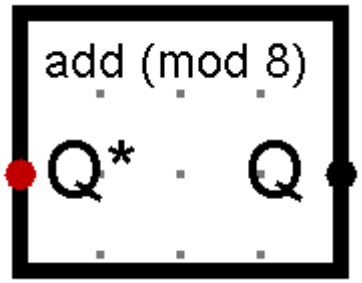
s(t+1) verrà scritto nel contatore in **retroazione**

Esercizio 1

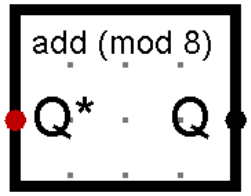
- I tre bit dello stato codificano il valore corrente del contatore
- Ad ogni ciclo di clock il valore (stato) passa al valore successivo, tranne il valore 7 che passerà a 0

Q_2	Q_1	Q_0	Q_2^*	Q_1^*	Q_0^*
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	0

$$Q_0^* = \neg Q_0$$
$$Q_1^* = Q_0 \text{ XOR } Q_1$$
$$Q_2^* = (Q_0 Q_1) \text{ XOR } Q_2$$



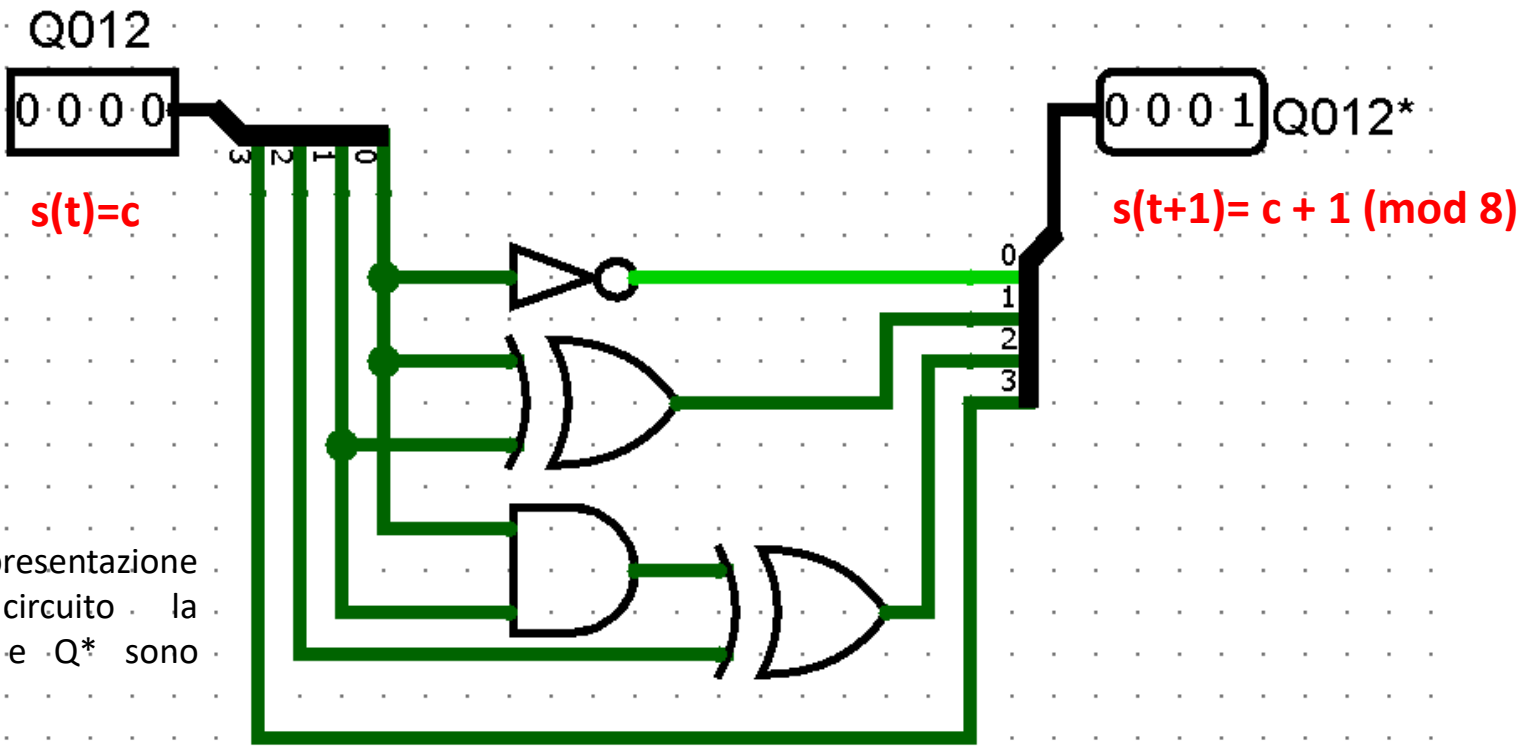
Esercizio 1



$$Q_0^* = \neg Q_0$$

$$Q_1^* = Q_0 \text{ XOR } Q_1$$

$$Q_2^* = (Q_0 Q_1) \text{ XOR } Q_2$$



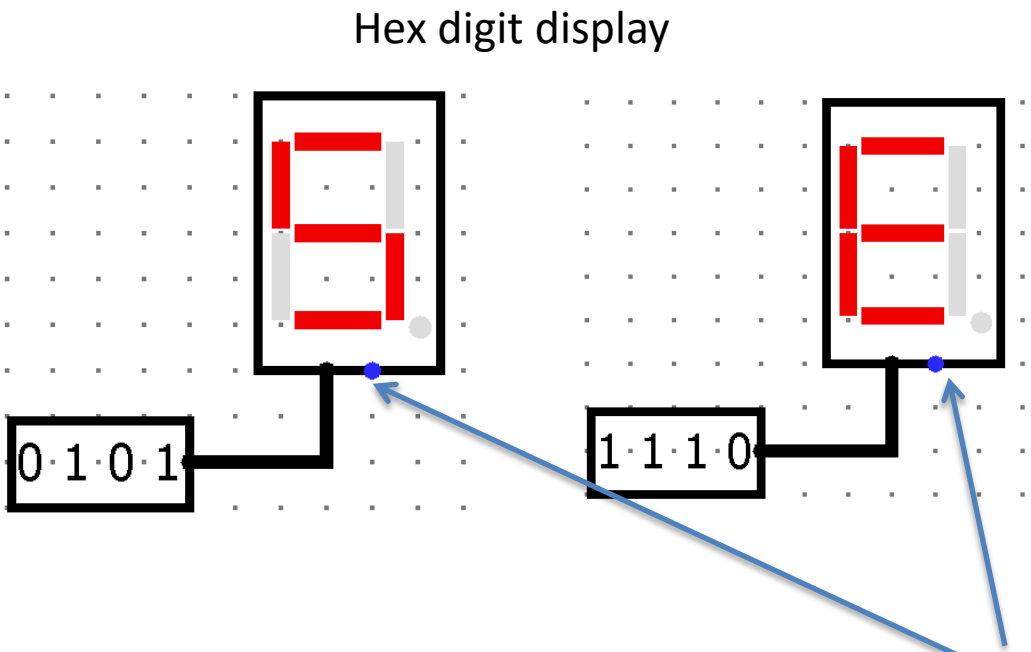
s(t)=c

s(t+1)= c + 1 (mod 8)

NOTA: nella rappresentazione esterna del circuito la posizione di Q e Q* sono scambiate di lato

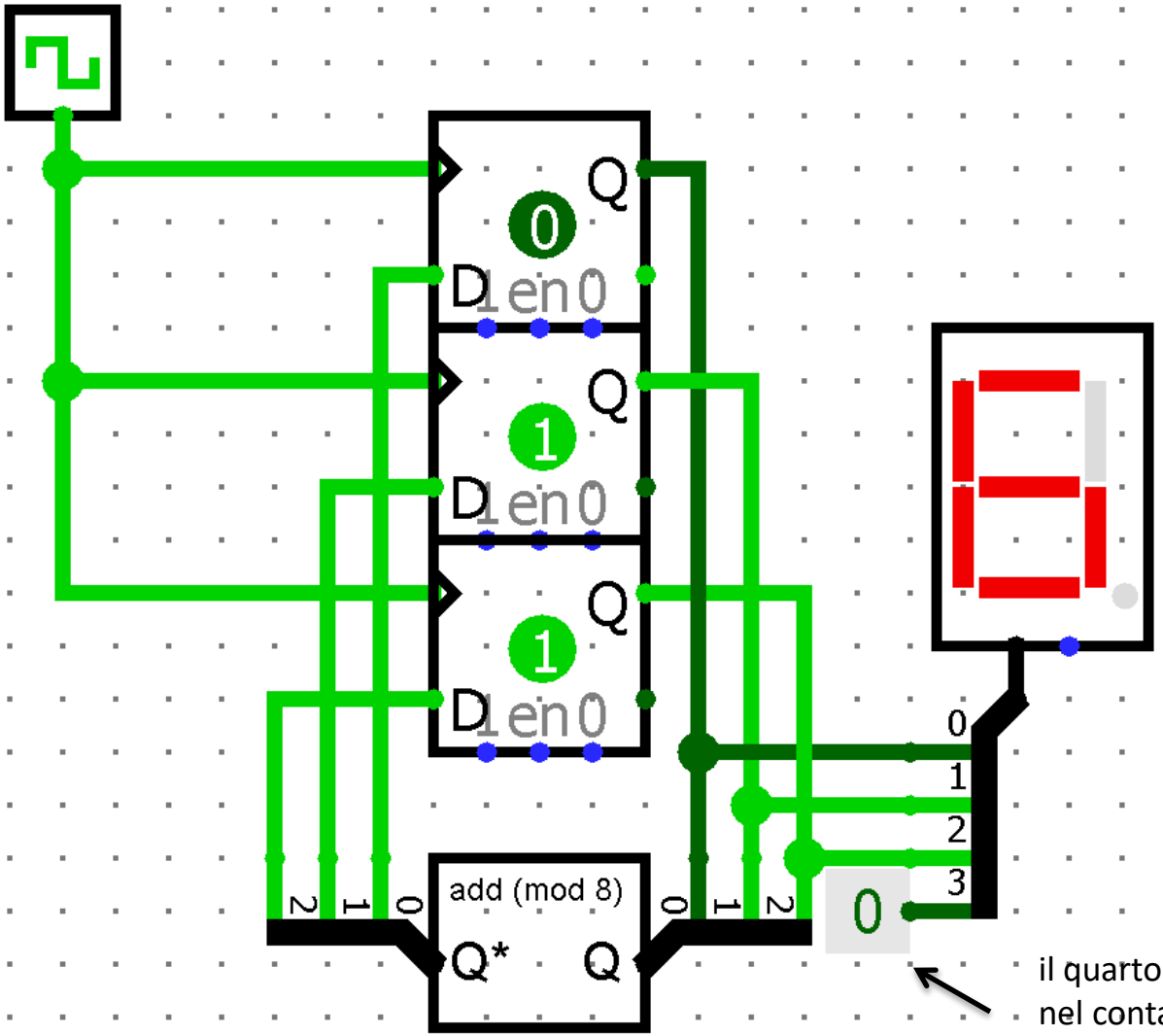
Esercizio 1

- Utilizziamo questo componente per visualizzare in modo “human-friendly” il valore corrente del registro

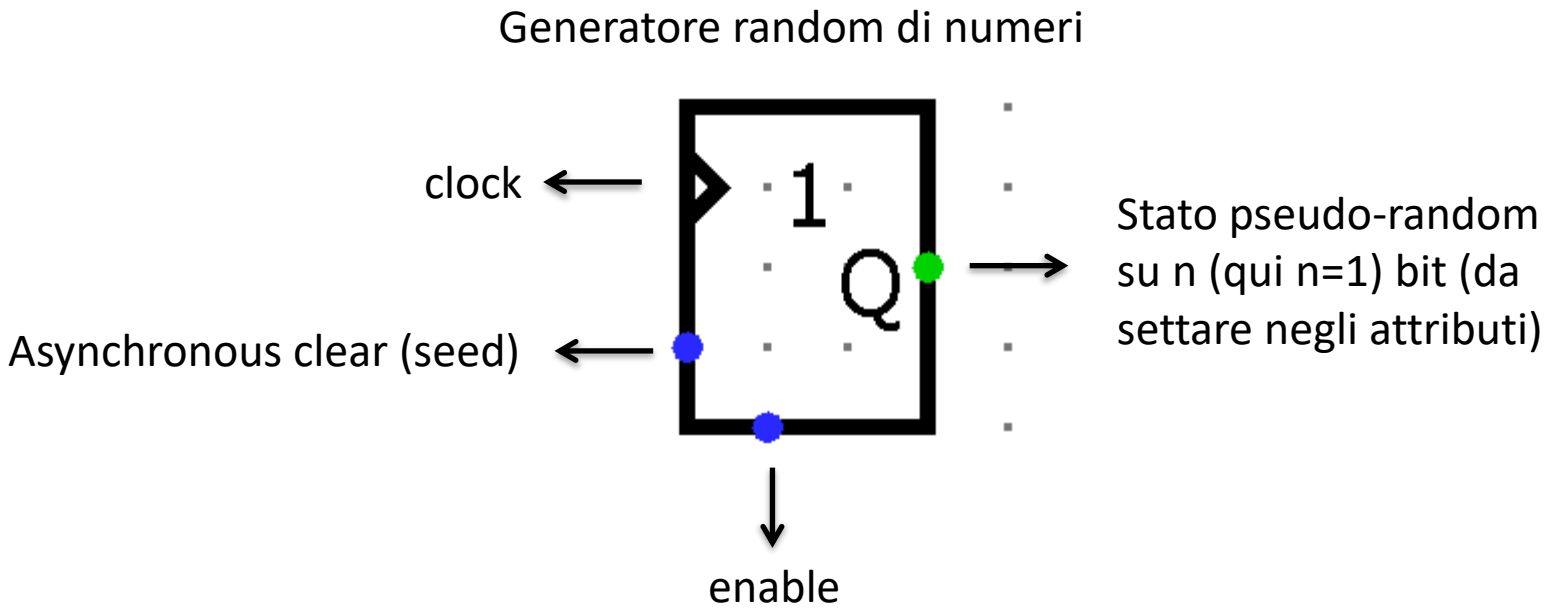


- Visualizza in base 16 un valore binario su 4 bit

Esercizio 1


Q
0
D
en
Q
1
D
en
Q
1
D
en
add (mod 8)
Q*
Q
0
1
2
3
0
il quarto bit non mi serve nel contatore, lo pongo a 0 nel display

Generatore Random



- Stato è su **n** bit
- A ogni ciclo di clock effettua una transizione verso uno dei possibili $\mathbf{2^n}$ stati
- La sequenza seguita è *pseudorandom*: significa che nella realtà è deterministica (ciclica e fissata da un valore iniziale che chiamiamo **seed**), ma che agli occhi di un osservatore attento è difficilmente prevedibile (quindi *sembra* random)

Generatore Random

- Come si realizza un generatore random di numeri?

- Uno degli approcci più usati e anche facili da implementare è l'uso di un tipo particolare di registri chiamati **Linear Feedback Shift Register** (LFSR)

LFSR

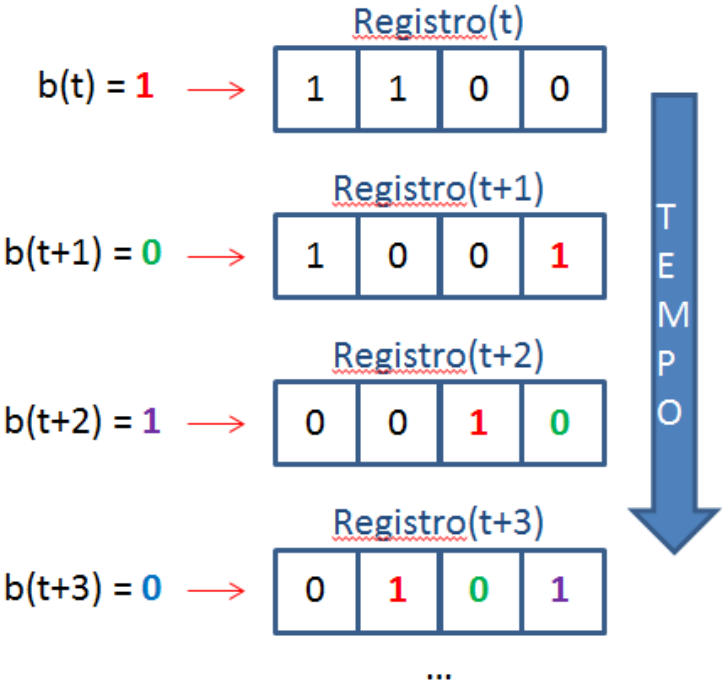
Linear Feedback Shift Register

- E' un registro a scorrimento, un componente che già conosciamo

Ad ogni istante di tempo diamo in input un singolo bit **b(t)** (input seriale)

Il registro shifta a sinistra tutto il suo contenuto per fare posto a b(t) e lo memorizza nel bit più a destra

I quattro bit del registro possono essere letti contemporaneamente ad ogni t (output parallelo)



LFSR

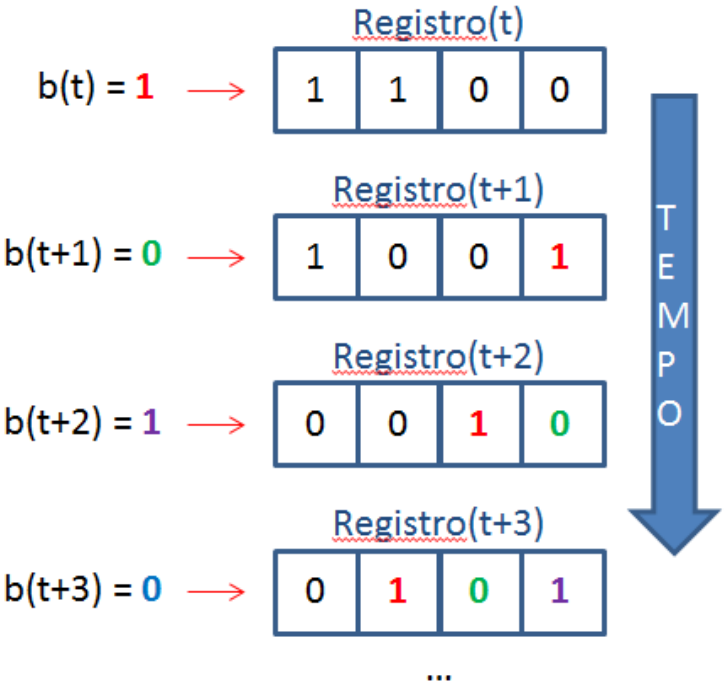
Linear Feedback Shift Register

- E' un registro a scorrimento, un componente che già conosciamo

Ad ogni istante di tempo diamo in input un singolo bit **b(t)** (input seriale)

Il registro shifta a sinistra tutto il suo contenuto per fare posto a b(t) e lo memorizza nel bit più a destra

I quattro bit del registro possono essere letti contemporaneamente ad ogni t (output parallelo)



- Come si determina il bit **b(t)** che entra in input ad ogni ciclo di clock? La spiegazione sta nella dicitura "Linear Feedback"

LFSR

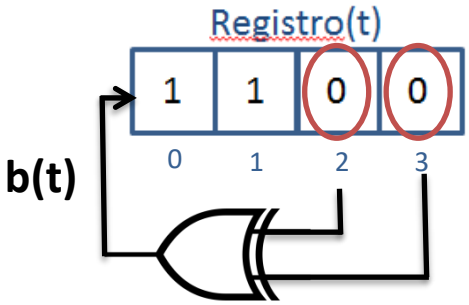
Linear Feedback Shift Register

- Linear feedback: il bit **b(t)** è una funzione lineare di un sottoinsieme di bit (detti taps) del registro al tempo **t**

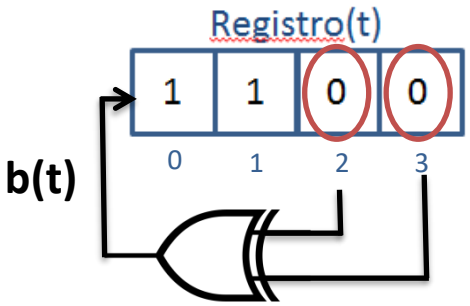
- Esempio: registro da 4 bit, scelgo taps: 2,3 (nota, qui metto LSD a destra e MSD a sinistra)

- Calcolo una funzione lineare dei taps, nel nostro caso sarà sempre lo XOR (è anche il caso più comune)

Retroazione (feedback)



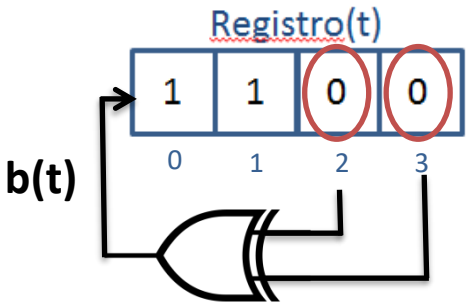
LFSR



- Lo stato prossimo dipende solo dallo stato corrente
- La sequenza seguita dipende da:
  - I taps scelti
  - La funzione lineare di feedback (nel nostro caso sempre XOR)
  - Il valore iniziale del registro, detto anche **seed**
- La sequenza si ripete dopo un certo numero di stati che costituisce il periodo della sequenza

- **Domanda**: quale è il periodo massimo di una sequenza generata da un LFSR di n bit?

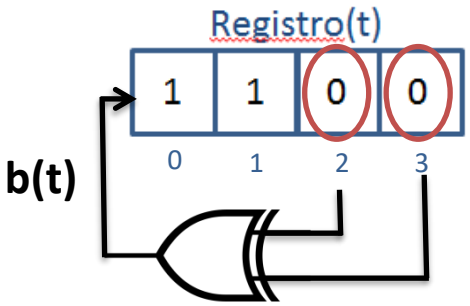
LFSR



- Lo stato prossimo dipende solo dallo stato corrente
- La sequenza seguita dipende da:
  - I taps scelti
  - La funzione lineare di feedback (nel nostro caso sempre XOR)
  - Il valore iniziale del registro, detto anche **seed**
- La sequenza si ripete dopo un certo numero di stati che costituisce il periodo della sequenza

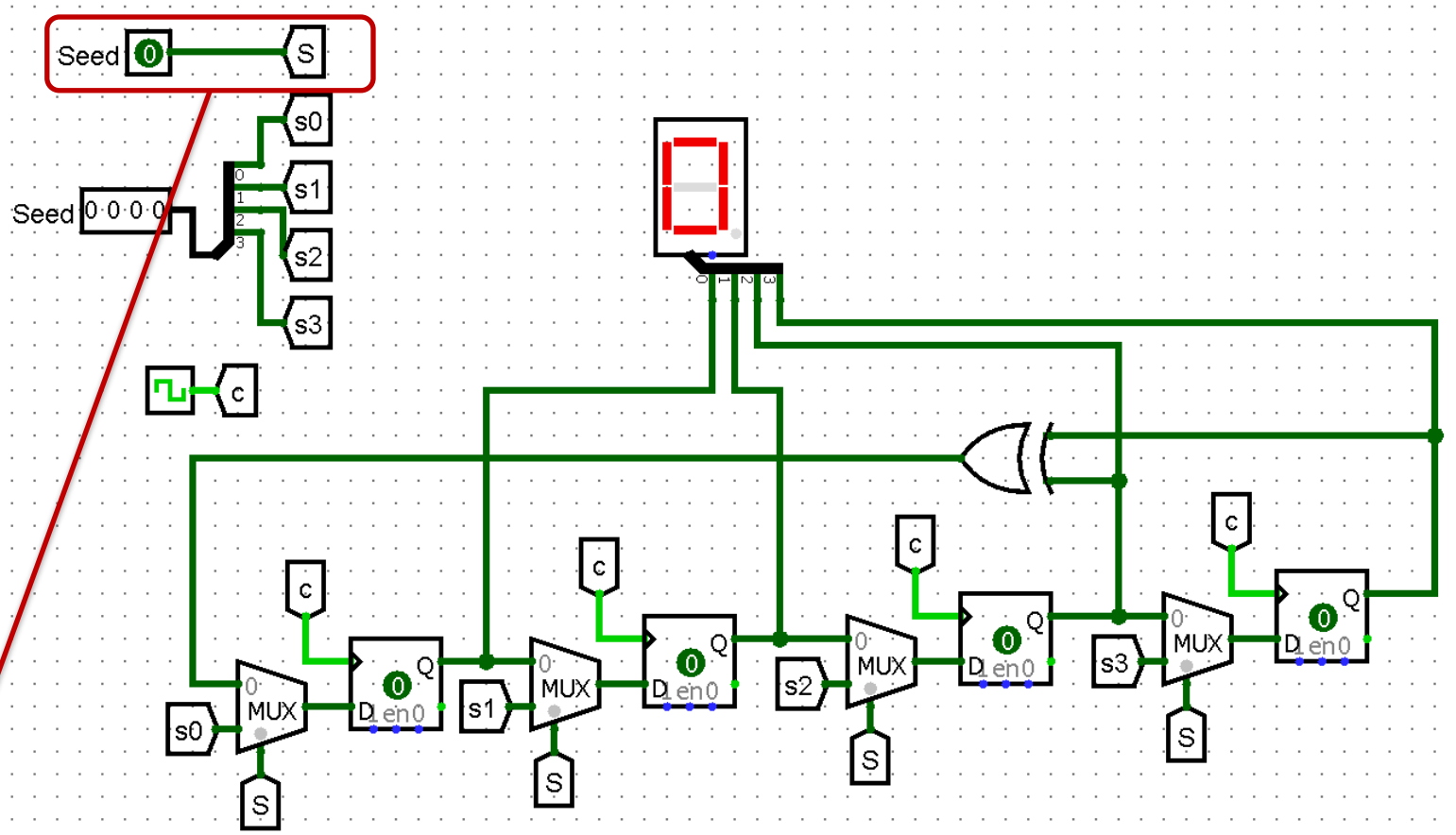
- **Domanda**: quale è il periodo massimo di una sequenza generata da un LFSR di n bit?
- **Risposta**: $2^n - 1$, tutti i possibili stati meno lo stato di soli zeri. Cosa succede in questo stato?

Esercizio 2



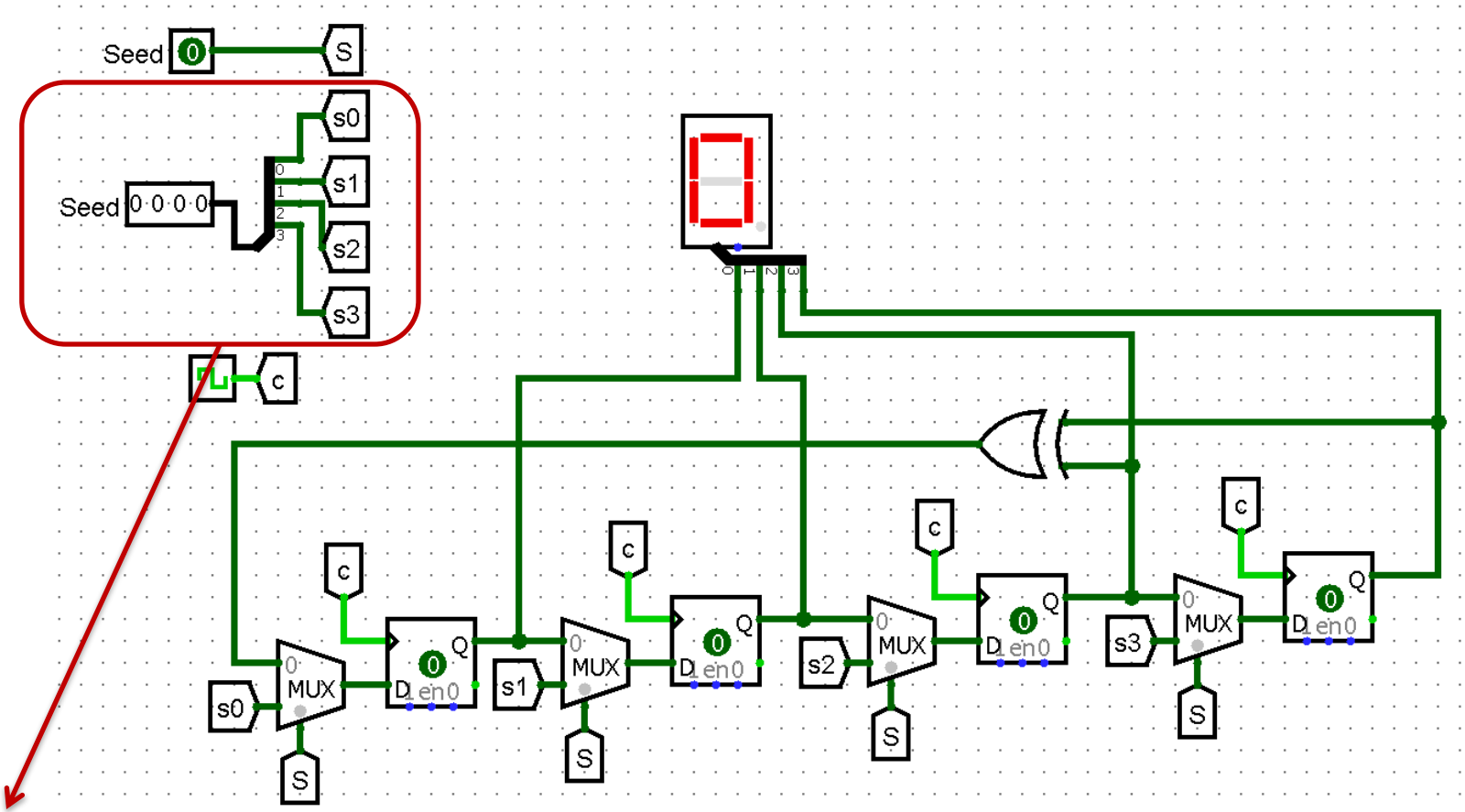
- Esercizio: implementiamo questo generatore e verifichiamo se produce una sequenza di periodo massimo
- Diamo anche la possibilità di operare in modalità scrittura (sincrona) del seed

Esercizio 2



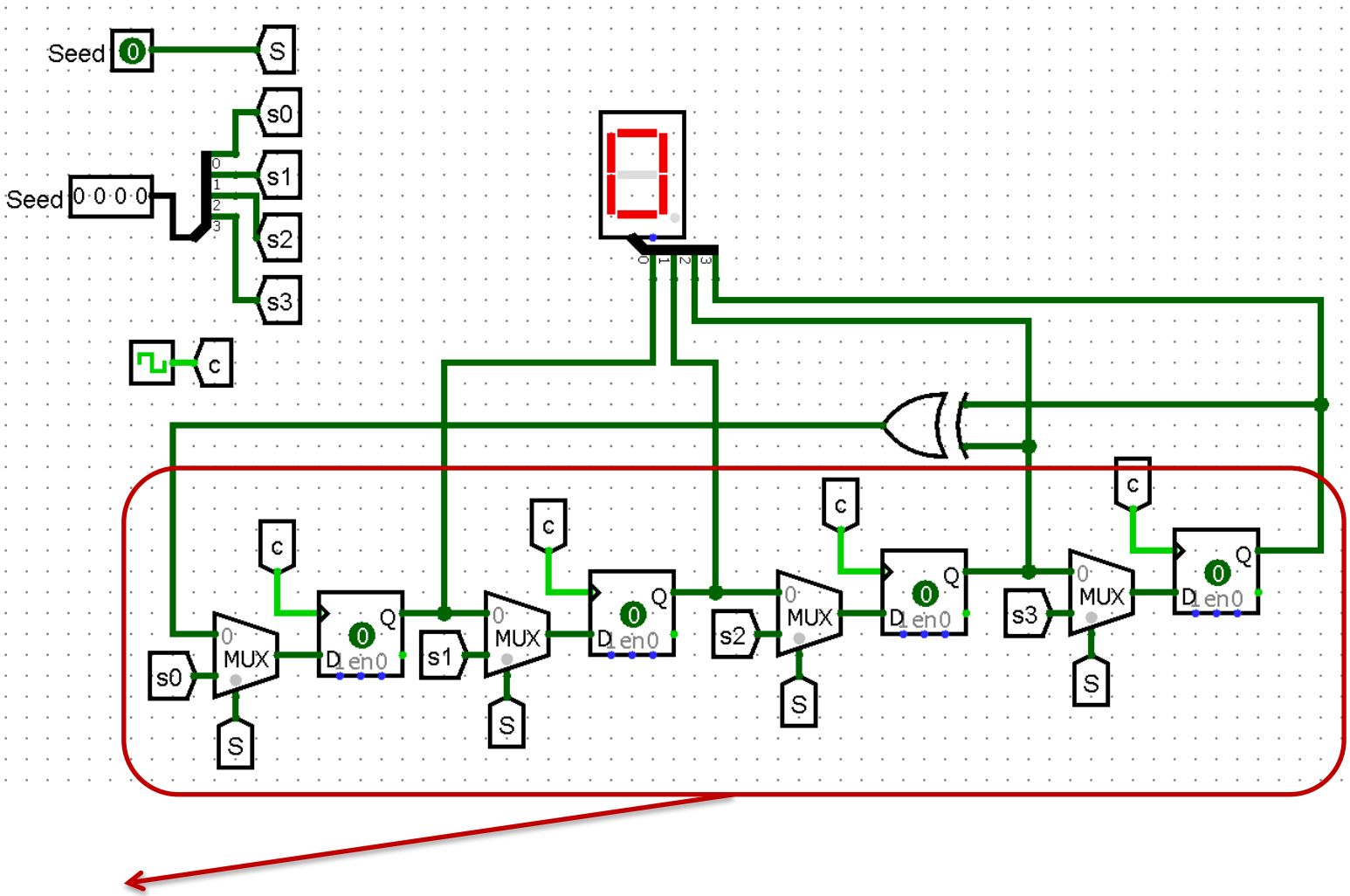
Bit per la modalità scrittura del seed: se posto a 1 scrivo in modo sincrono un seed nel registro

Esercizio 2



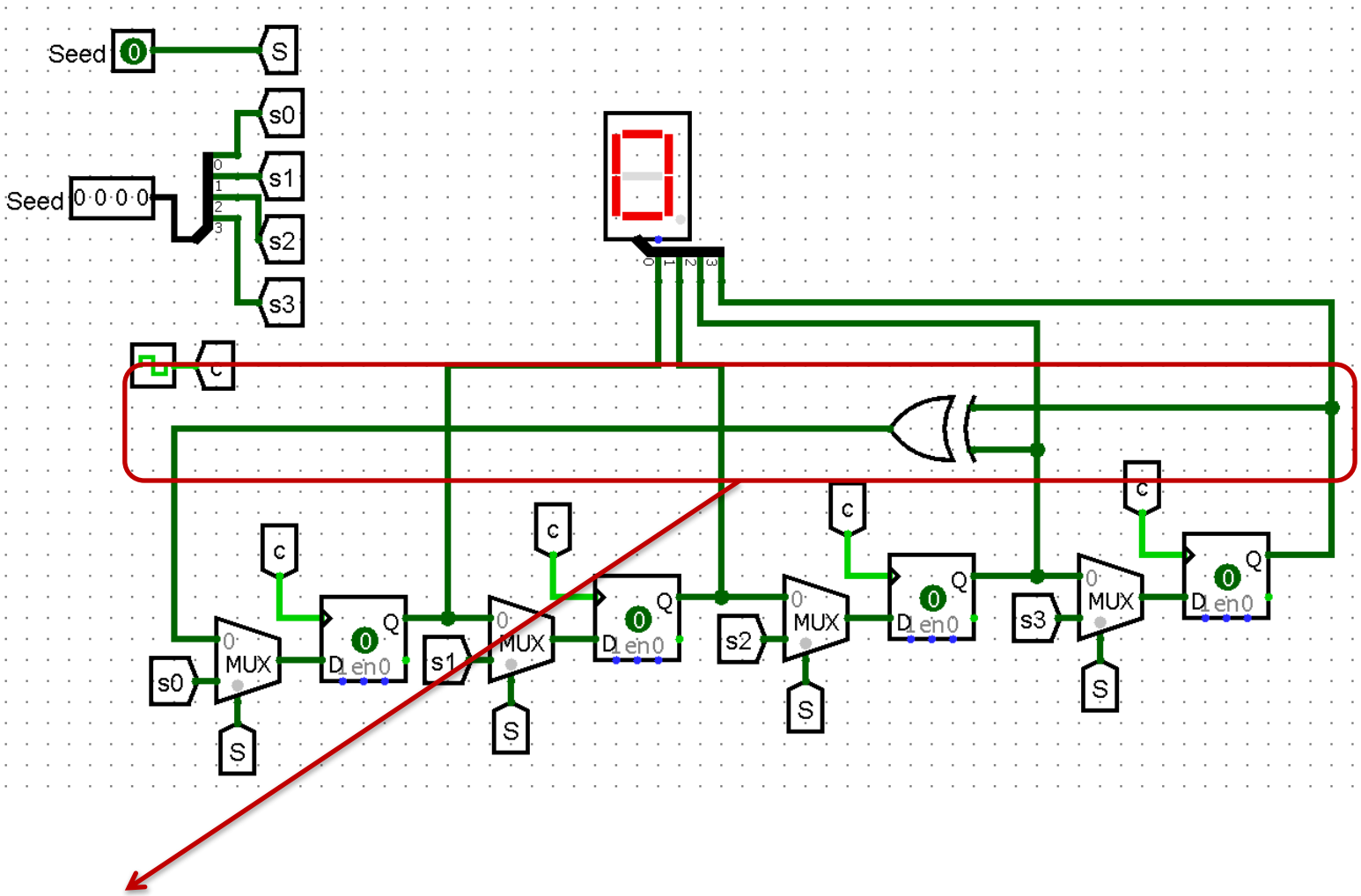
Seed da scrivere

Esercizio 2



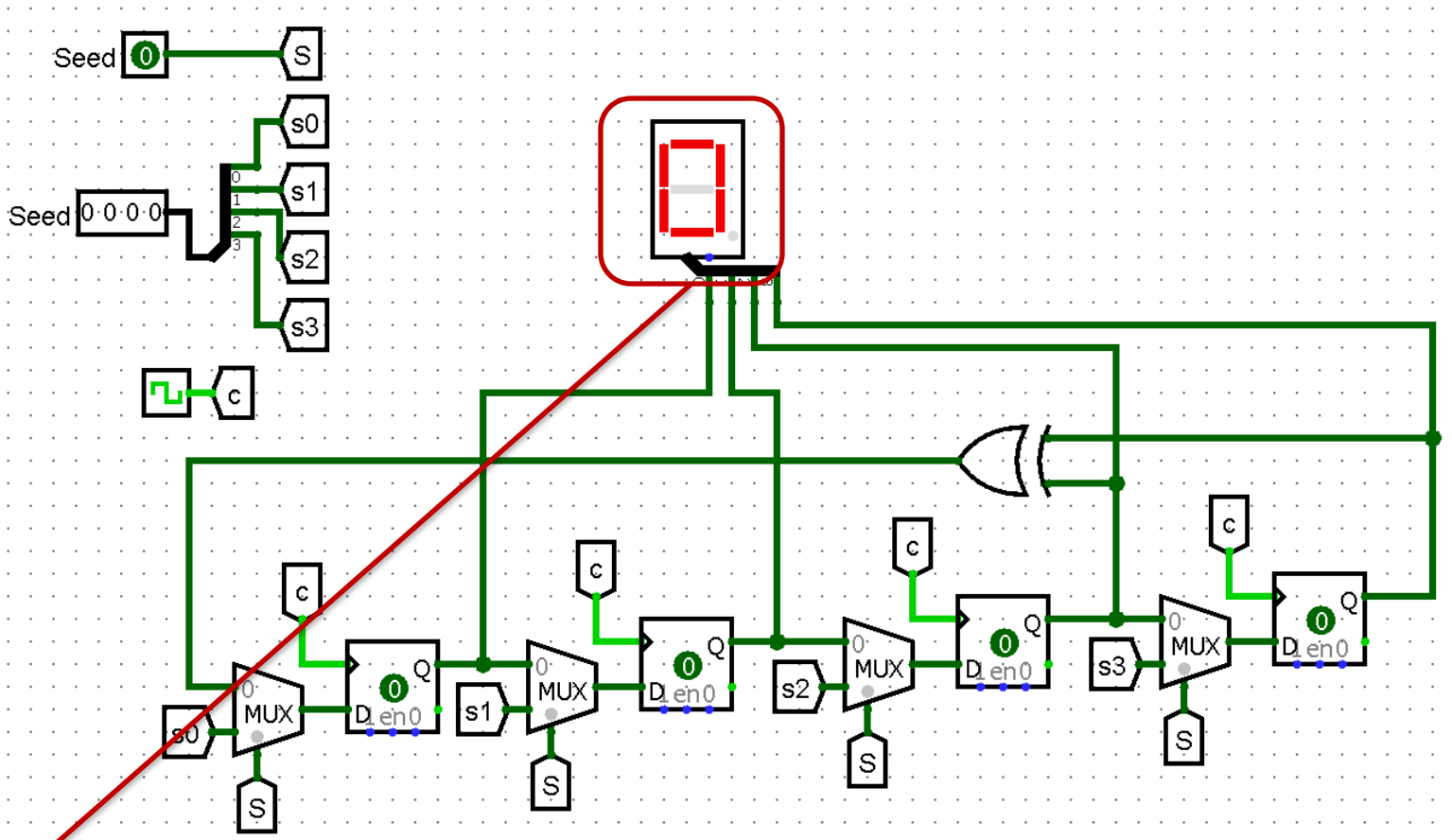
Registro a scorrimento classico; i multiplexer all'ingresso di ogni flip-flop permettono di selezionare tra lo scorrimento o la scrittura del seed (il bit di seed è il selezionatore)

Esercizio 2



Linear feedback: lo XOR dei taps 2 e 3 viene mandato in input al primo flip-flop

Esercizio 2



Sul display visualizzo il valore corrente della sequenza