Laboratorio di Architetture degli Elaboratori I
Corso di Laurea in Informatica, A.A. 2019-2020
Università degli Studi di Milano

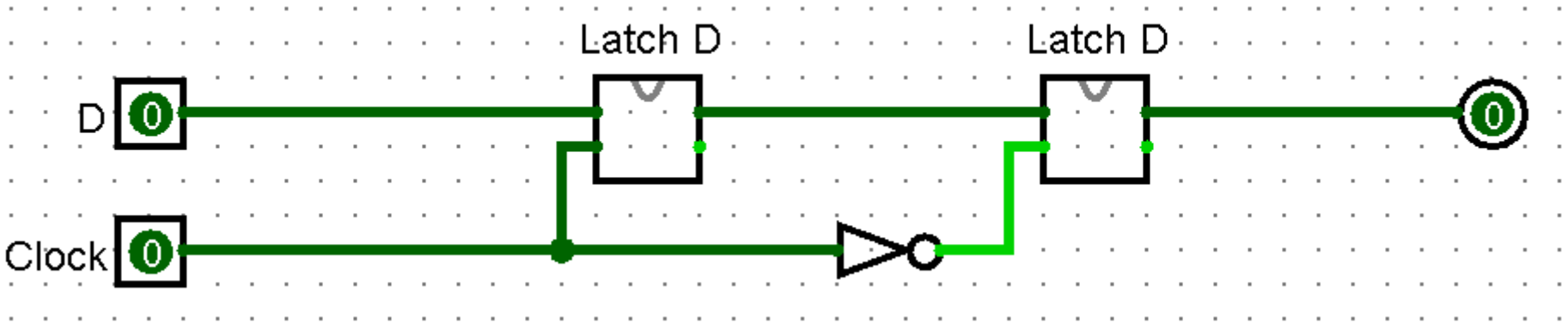
Memorie

Esercizio 0

- Si realizzi un flip-flop a partire da due latch D.

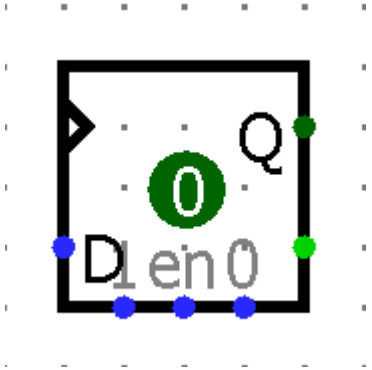
Esercizio 0

- Si realizzi un flip-flop a partire da due latch D.



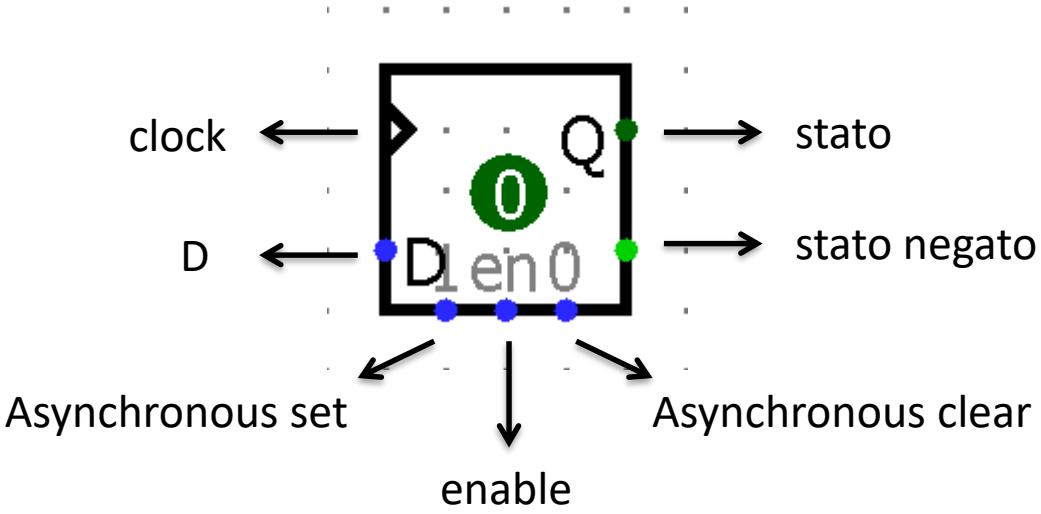
Esercizio 0

- Componenente logisim



Esercizio 0

- Componenente logisim



- **Clock**: sul *rising edge* (quando il clock passa da 0 a 1), lo stato viene aggiornato
- **Enabled**: se posto a 0 il segnale di clock viene ignorato
- **Asynchronous set**: se posto a 1, lo stato viene settato a 1 in modo asincrono (ignorando il clock)
- **Asynchronous clear**: se posto a 1, lo stato viene resettato a 0 in modo asincrono

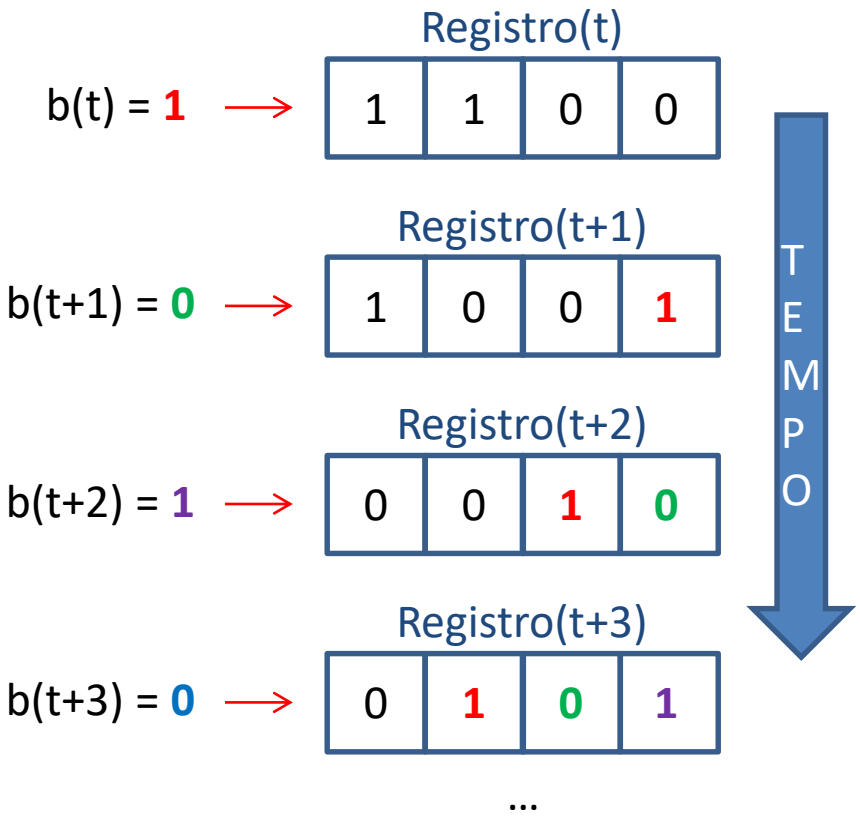
Esercizio 1

- Si realizzi un **registro a scorrimento** (verso sinistra) con input seriale e output parallelo.

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

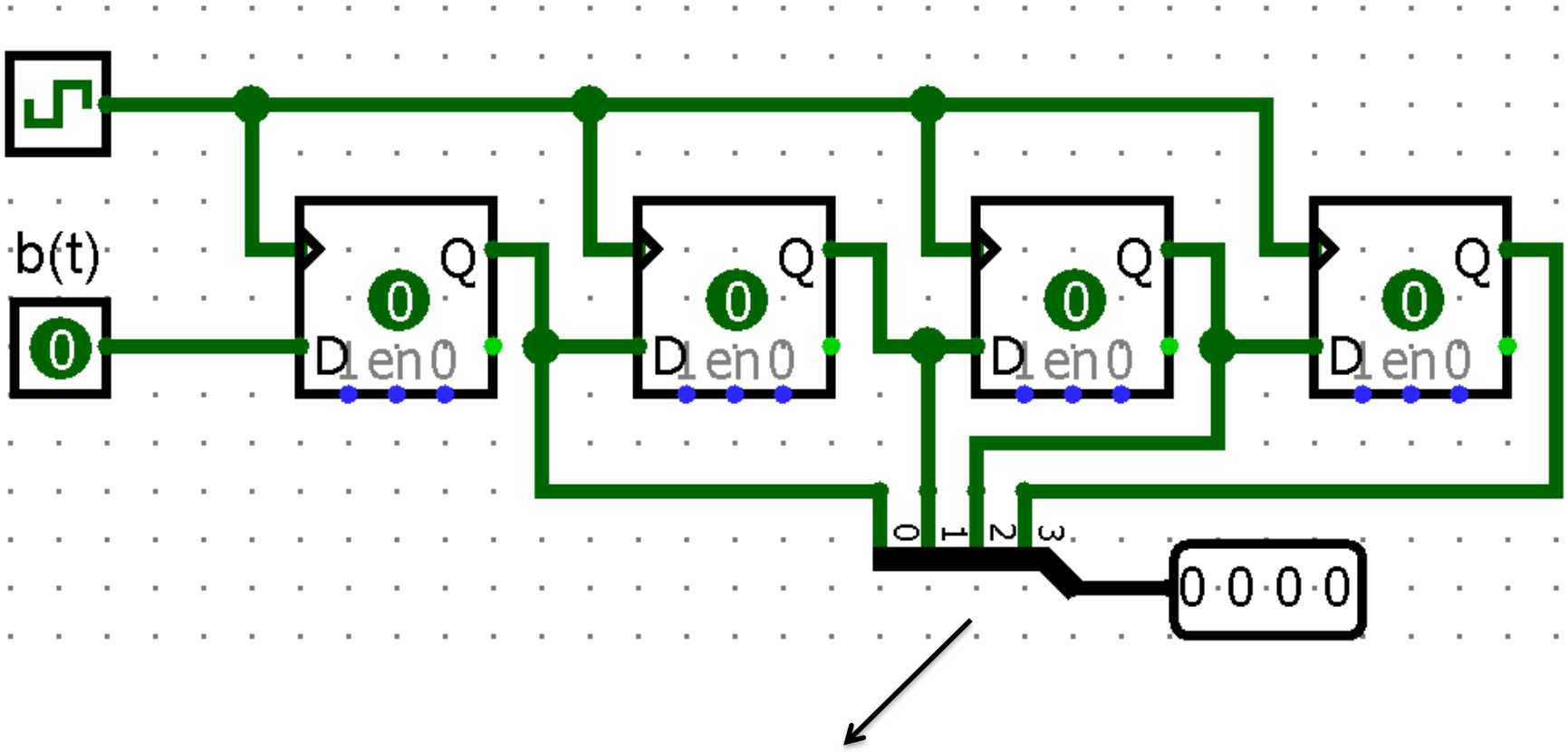
Il registro shifta a sinistra tutto il suo contenuto per fare posto a b(t) e lo memorizza nel bit più a destra

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



Esercizio 1

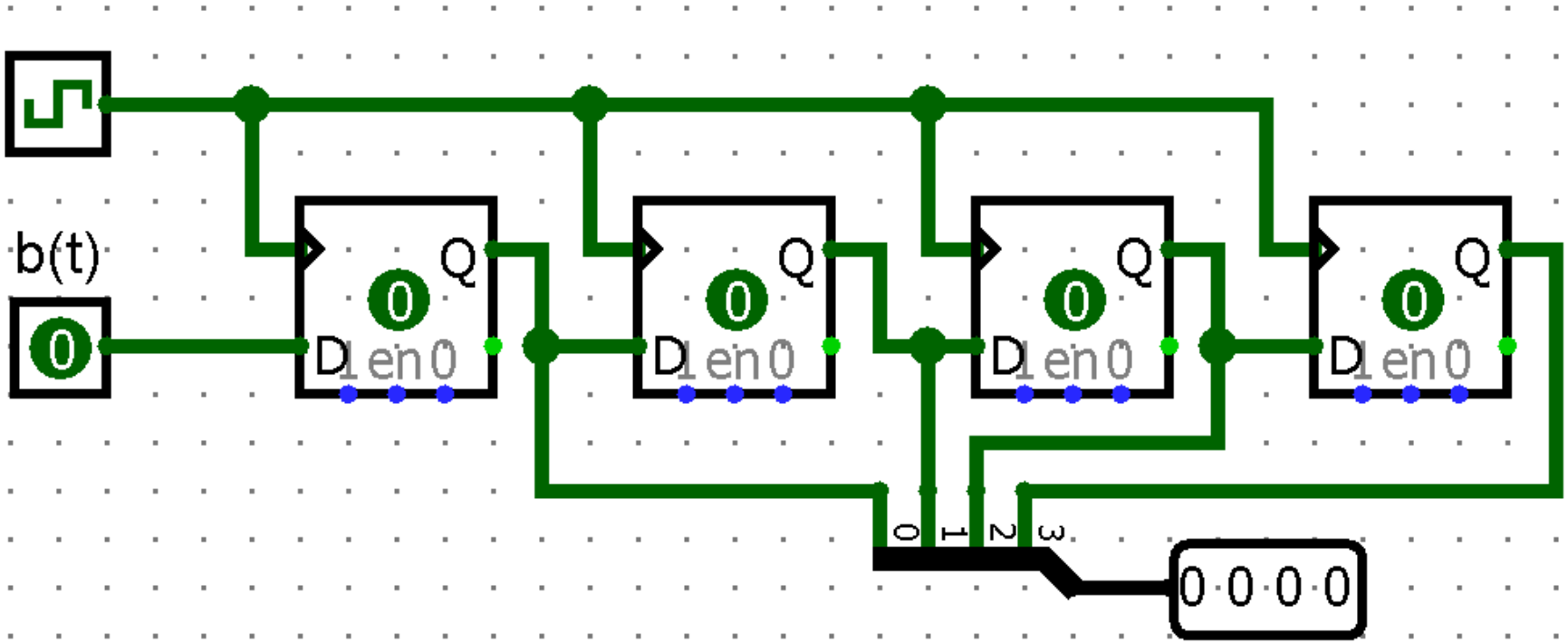
- 4 Flip Flop



Invertiamo l'ordine dei bit nello splitter per ottenere uno shift da destra a sinistra

Esercizio 1

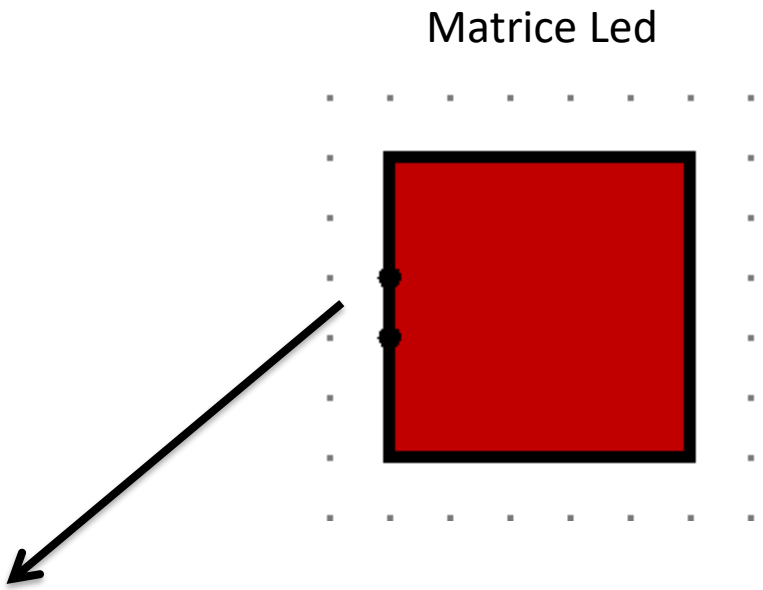
- 4 Flip Flop



- Avremmo potuto ottenere lo stesso risultato utilizzando 4 latch D?

Esercizio 1

- Proviamo a rendere questo circuito graficamente più «interessante»

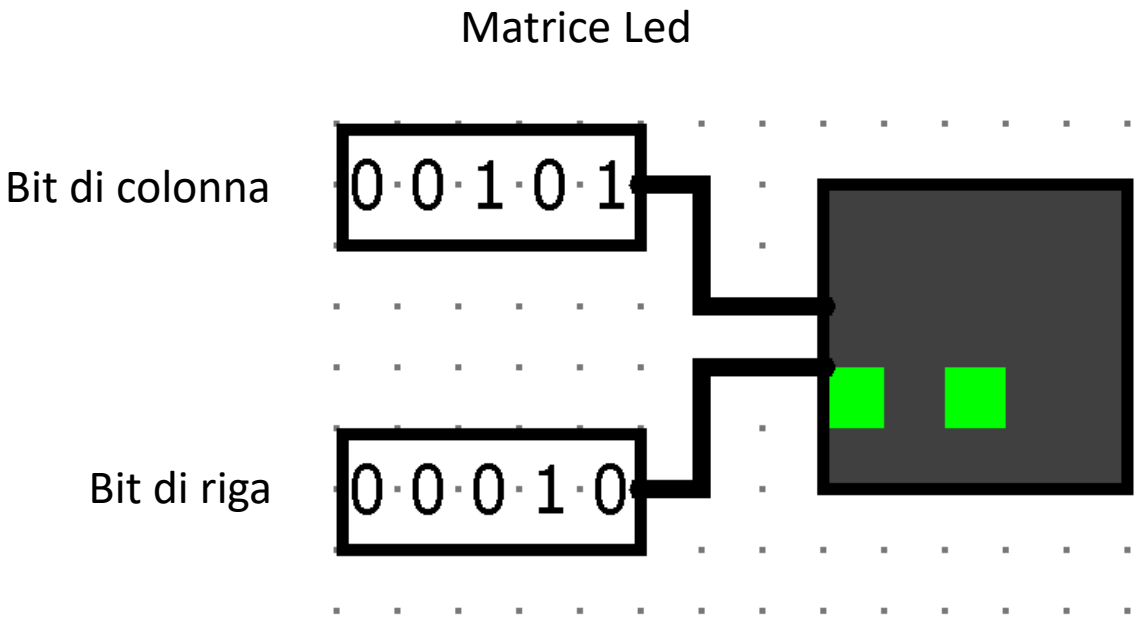


Ha numero di righe e numero di colonne (settabili in attributi), ogni elemento è un led. Qui abbiamo 4x4 led

Formato di input: 4 bit per righe, 4 bit per colonne.
Esempio: se terzo bit delle righe e secondo bit delle colonne sono a 1, allora il led in posizione (3,2) è acceso

Esercizio 1

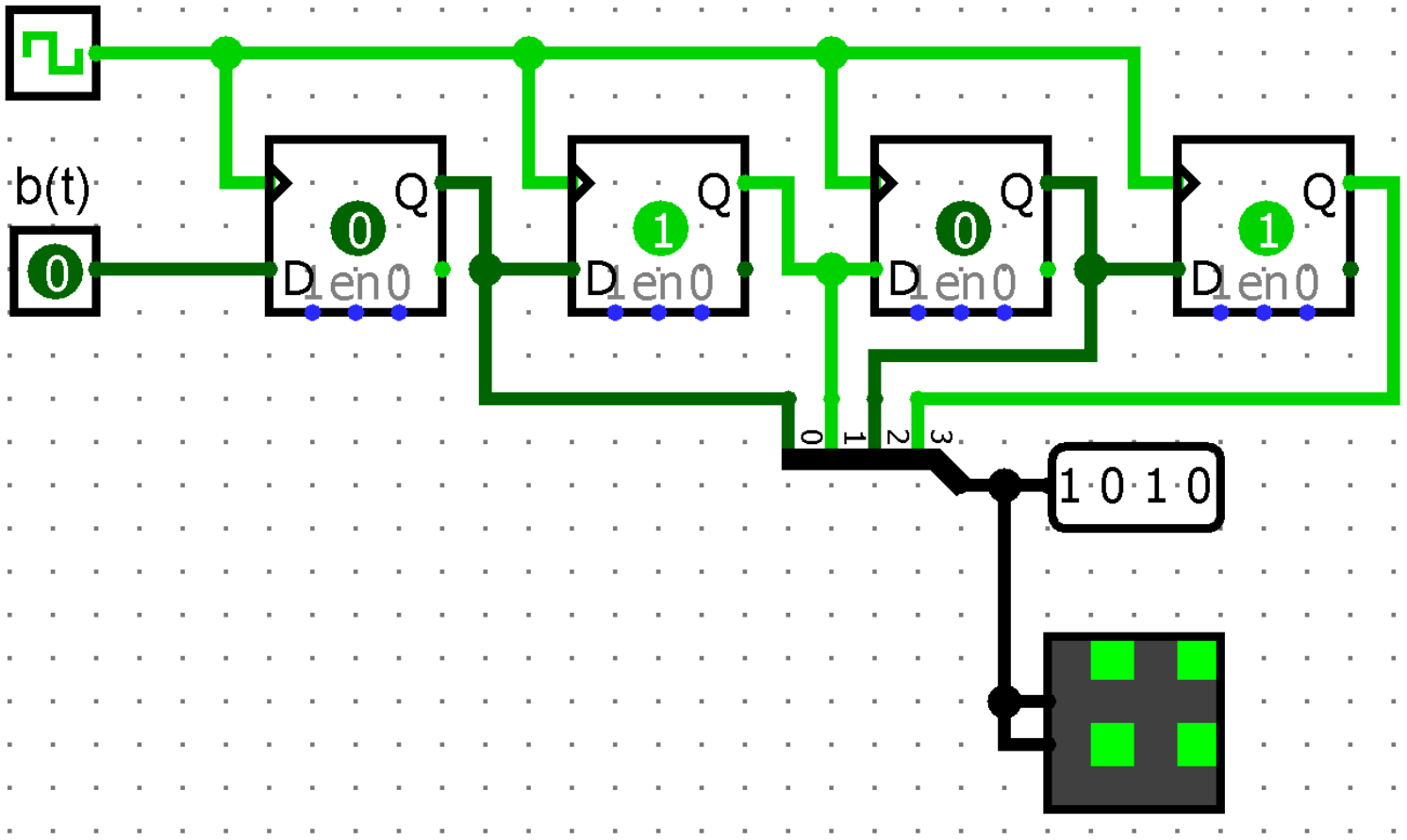
- Proviamo a rendere questo circuito graficamente più «interessante»



Esempio:
Riga 1, colonne 0 e 2: accendo led in posizione (1,0) e (1,2)

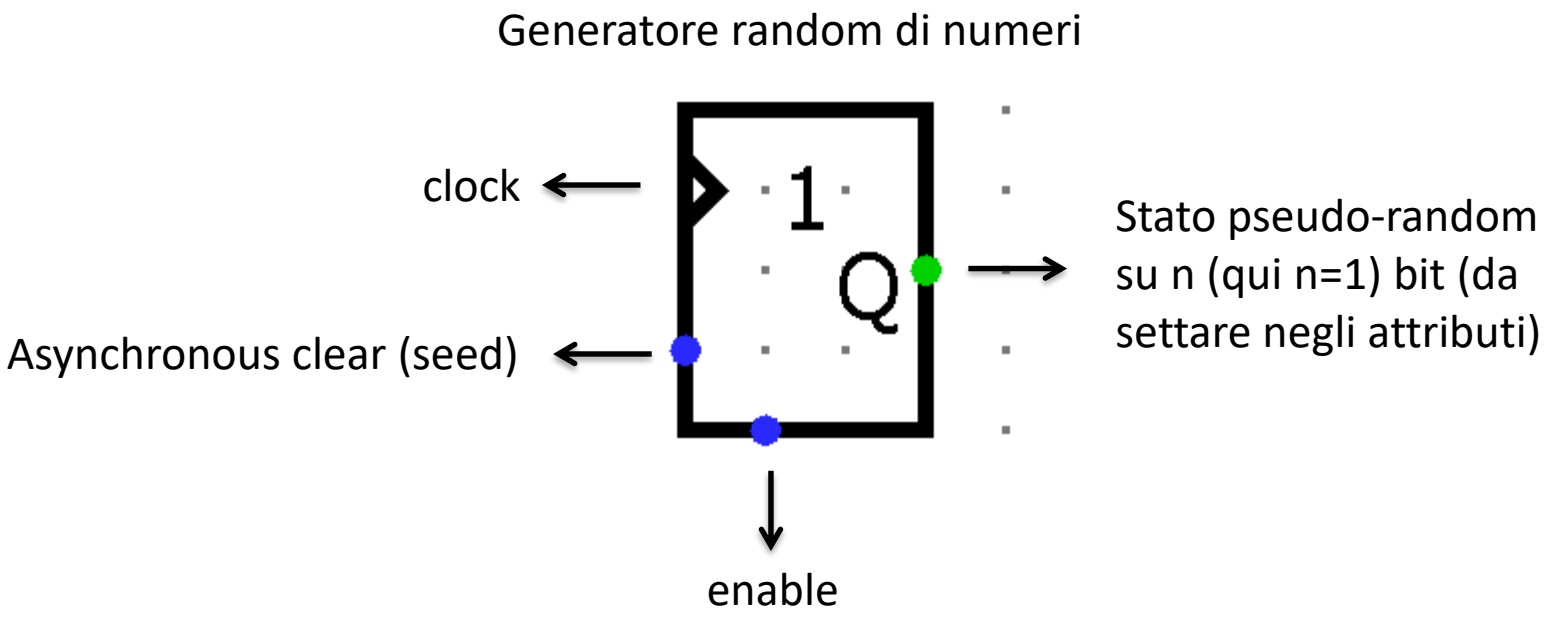
Esercizio 1

- Colleghiamo la matrice led 4x4 al nostro registro a scorrimento replicando i 4 bit del registro sia per le 4 righe che per le 4 colonne:



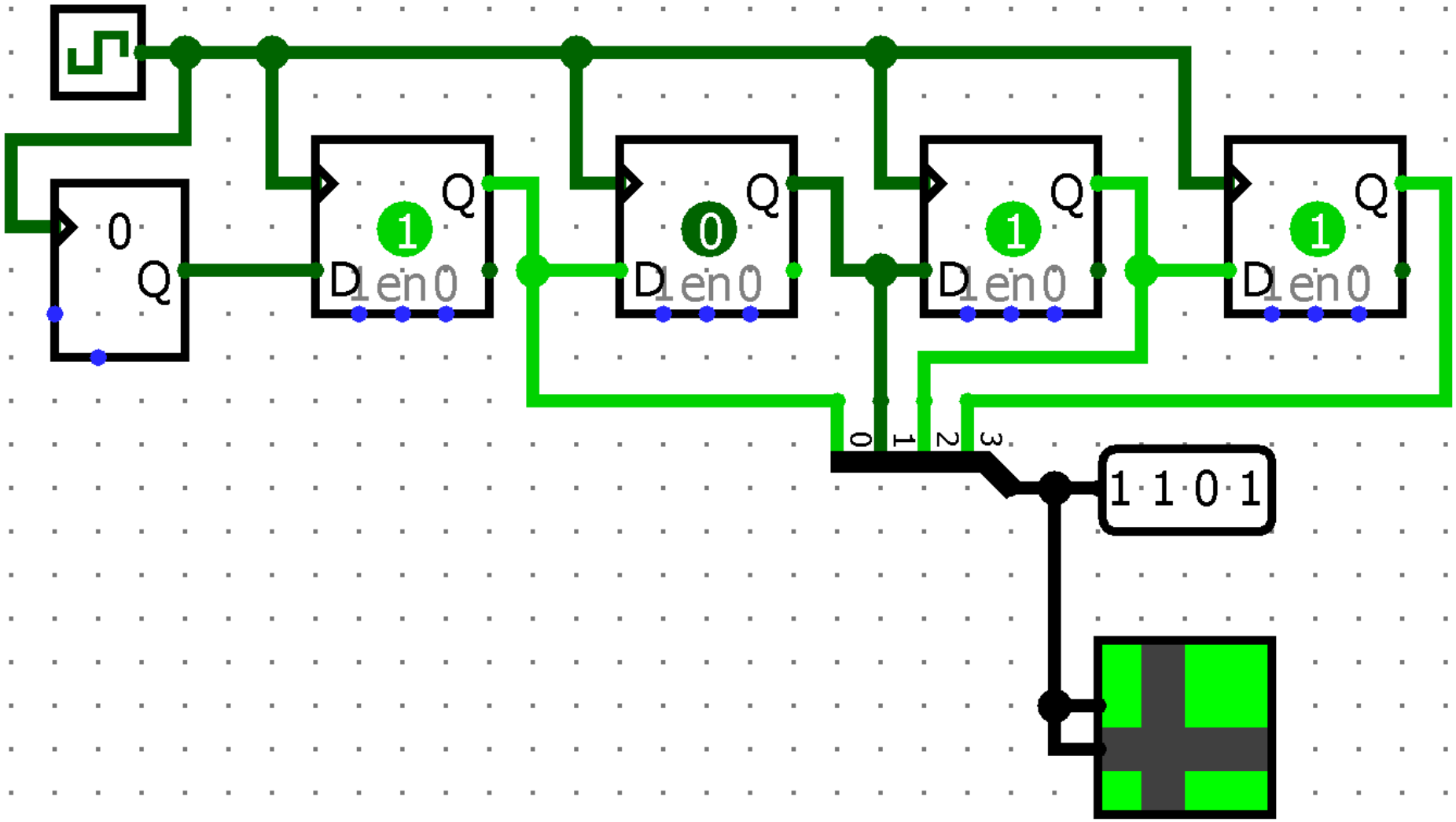
Esercizio 1

- Proviamo a rendere questo circuito graficamente più «interessante»



- Ad ogni rising edge (o falling edge se specificato negli attributi) cambia il suo stato in modo random. Proviamo a collegarlo al nostro registro come input b(t) e osserviamo che succede

Esercizio 1

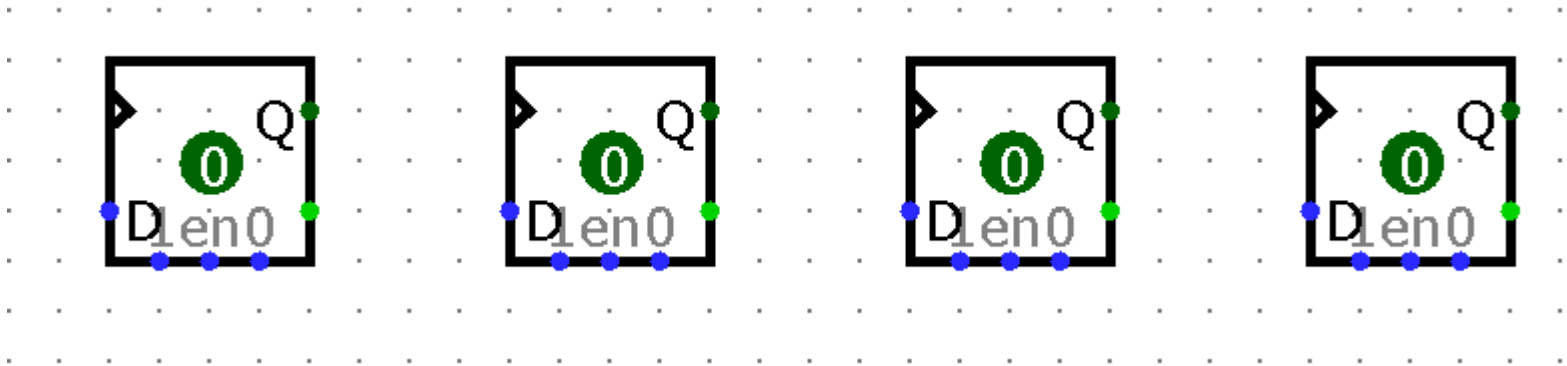


Esercizio 2

- Si realizzi un banco di memoria da 4 bit, formato da 4 FF di tipo D che consenta di effettuare, previa selezione, una delle seguenti funzioni:
  - Scrittura parallela di 4 bit (input di 4 bit e scrittura di ciascun bit in un FF);
  - Reset asincrono dei 4 FF;
  - Inserimento di un bit a sinistra (con shift della parola correntemente memorizzata nei FF verso destra).

Esercizio 2

- Suggerimenti:



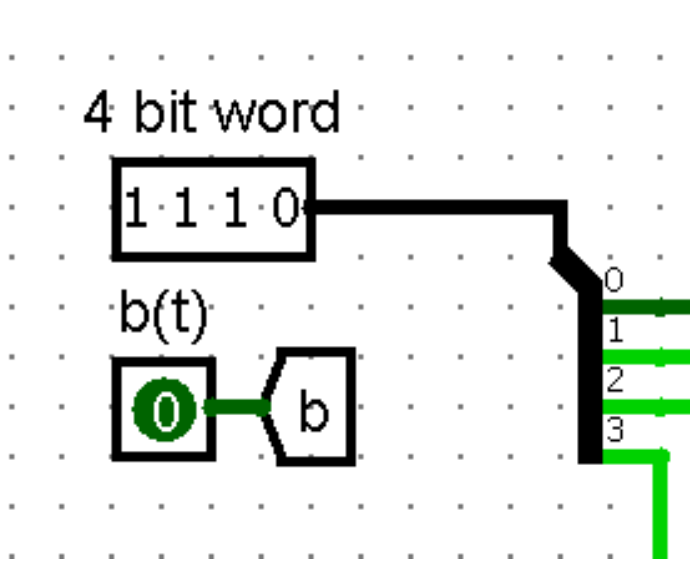
Il banco di memoria è formato da 4 FF di tipo D:

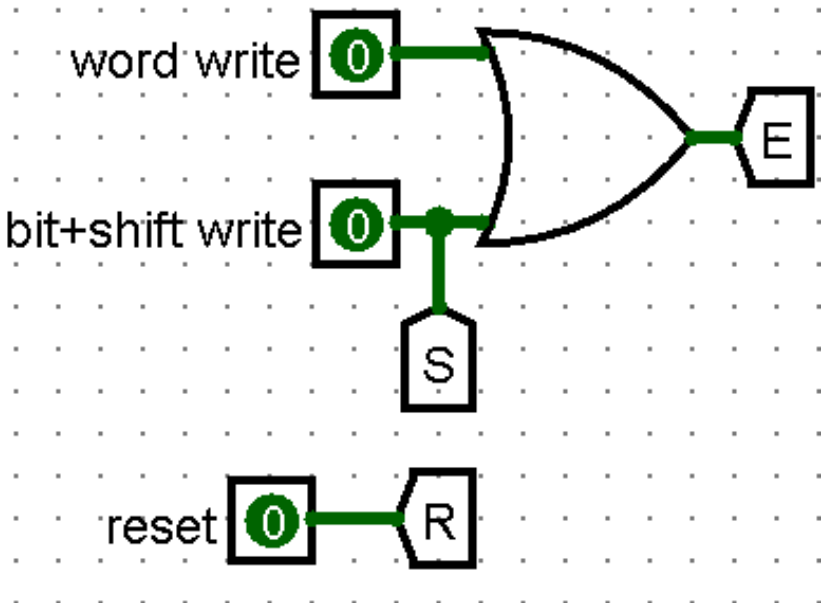
- Segnale **Word write**: vale 1 se in modalità parallel input
- Segnale **S**: vale 1 se in modalità serial input con shift
- Segnale **E** (enable): vale 1 se voglio scrivere in parallelo i 4 bit **oppure** inserire un bit con shift
- Segnale **R** (reset): se R=1, clear asincrono di ciascun FF

- Cosa deve entrare nei FF? Due possibilità: i 4 bit da scrivere in parallelo **oppure** lo shift con inserimento a sinistra (come nell'esercizio precedente)

Esercizio 2

- Soluzione (iniziamo dagli input):

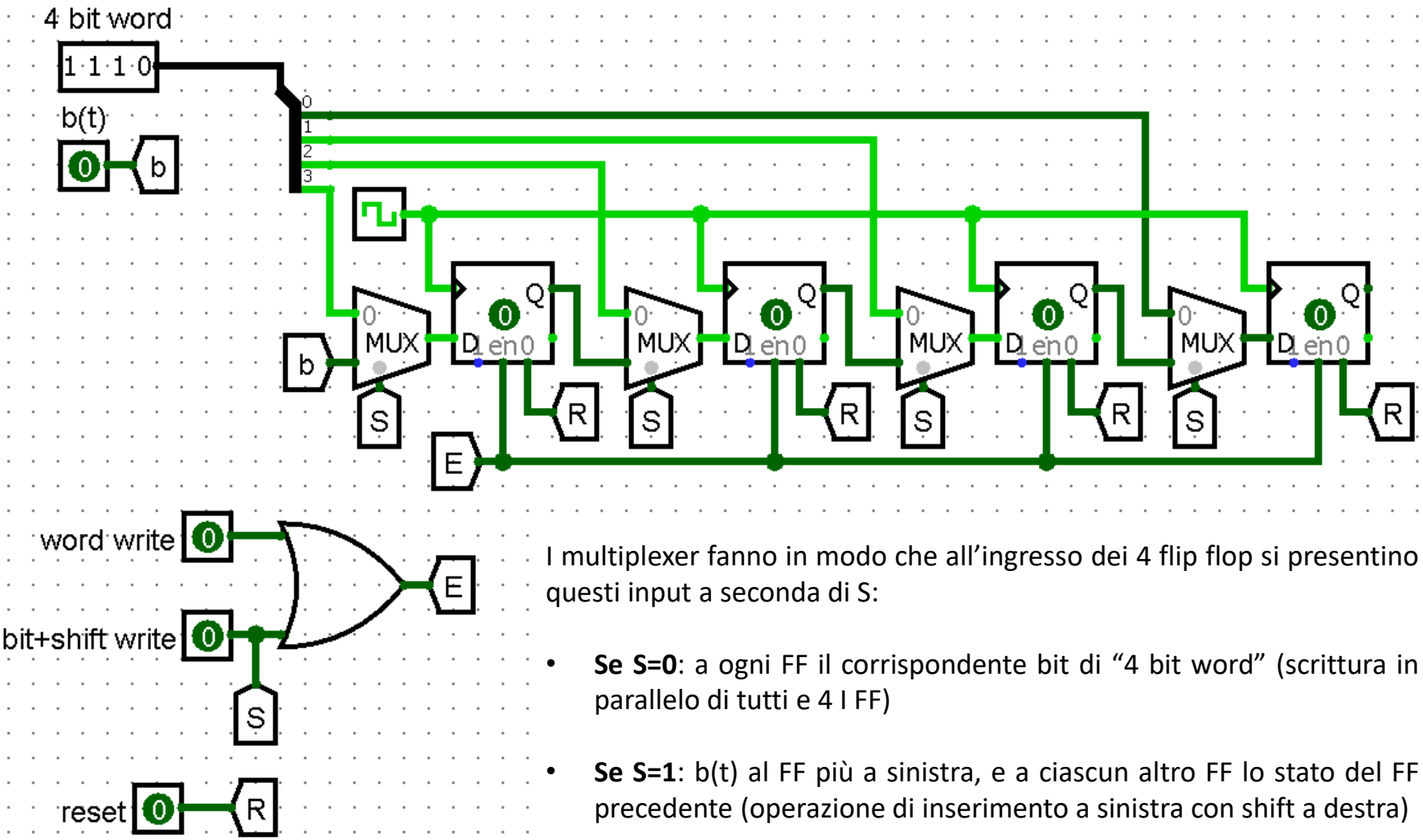




- **4 bit word:** la parola da scrivere in modalità parallel input
- **b(t):** il bit da inserire a sinistra quando in modalità serial input (con shift)
- **Word write**: vale 1 se in modalità parallel input
- **S** (Bit+shift write): vale 1 se in modalità serial input con shift
- **E** (enable): vale 1 se voglio scrivere in parallelo i 4 bit o inserire un bit con shift
- **R** (reset): setta la memoria a 0 0 0 0 in modo asincrono

Esercizio 2

- Soluzione (circuito completo):



I multiplexer fanno in modo che all'ingresso dei 4 flip flop si presentino questi input a seconda di S:

- **Se S=0**: a ogni FF il corrispondente bit di "4 bit word" (scrittura in parallelo di tutti e 4 I FF)
- **Se S=1**: b(t) al FF più a sinistra, e a ciascun altro FF lo stato del FF precedente (operazione di inserimento a sinistra con shift a destra)

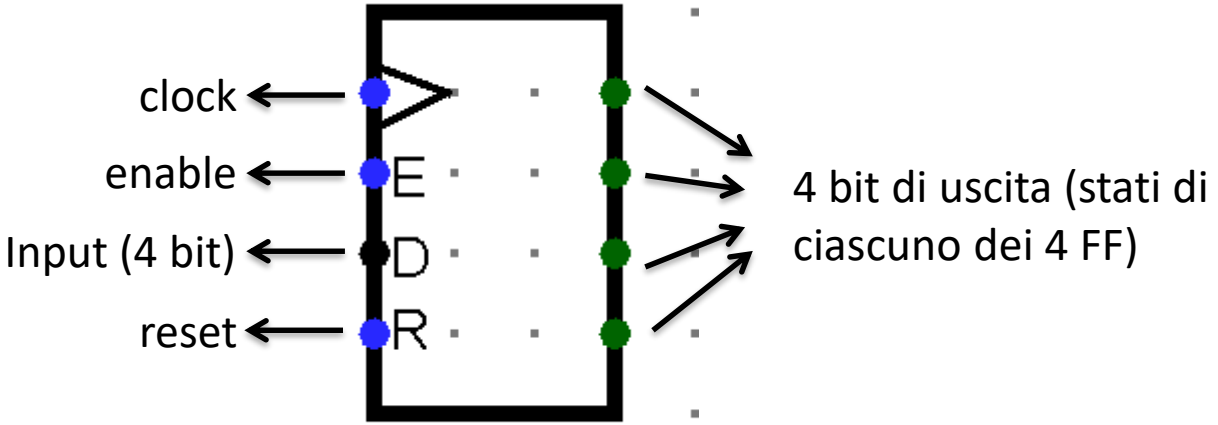
Esercizio 3

- Si realizzi un registro di memoria a 4 bit (4 FF di tipo D)
- Si crei una banco di memoria composto da 4 di tali registri
- Si implementi la funziona di scrittura parallela di 4 bit con indirizzamento del registro.

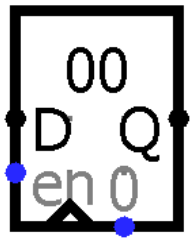
- Segnali in input:
  - **RESET**: se 1, setta in modo asincrono tutti i 16 bit del banco di memoria
  - **WRITE**: se 1, permette la scrittura
  - **WORD**: i 4 bit da scrivere
  - **ADDRESS**: 2 bit che identificano in quale dei 4 registri scrivere

Esercizio 3

- Registro da 4 bit (realizzato da noi), visuale esterna

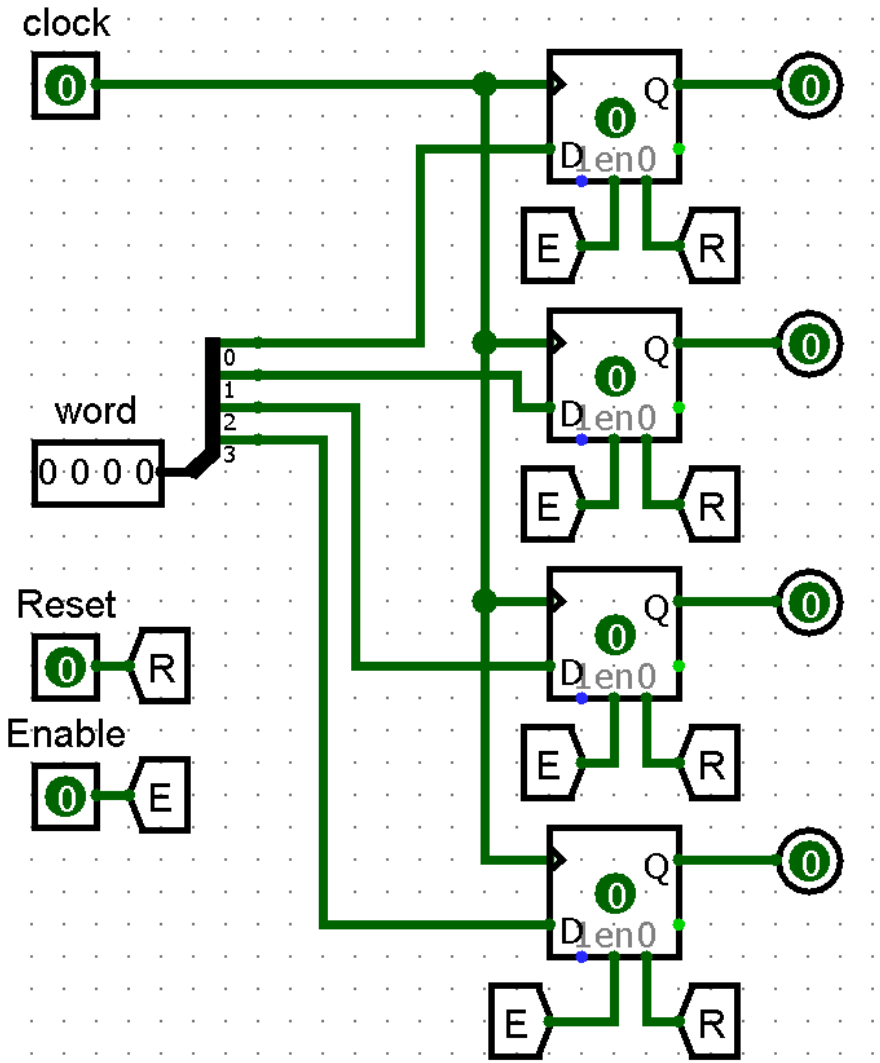


- C'è anche un componente logisim (register):



Esercizio 3

- Registro da 4 bit (realizzato da noi), visuale interna

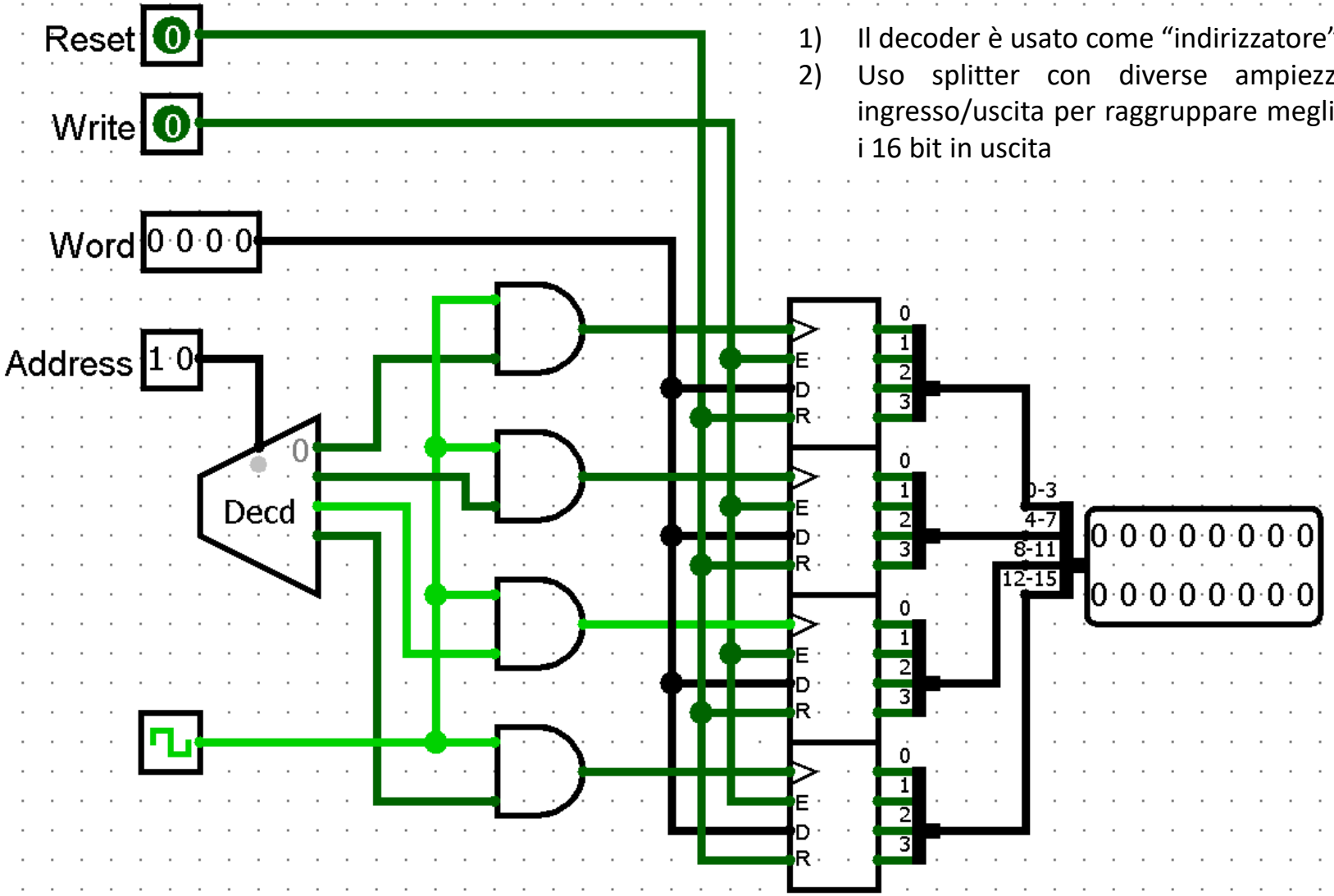


NOTE:

1) Non uso una sorgente di clock ma un normale pin di ingresso: la sorgente di clock sarà nel circuito più esterno che conterrà questo
2) Uso 4 pin di uscita per avere quattro diverse uscite nella rappresentazione esterna del circuito

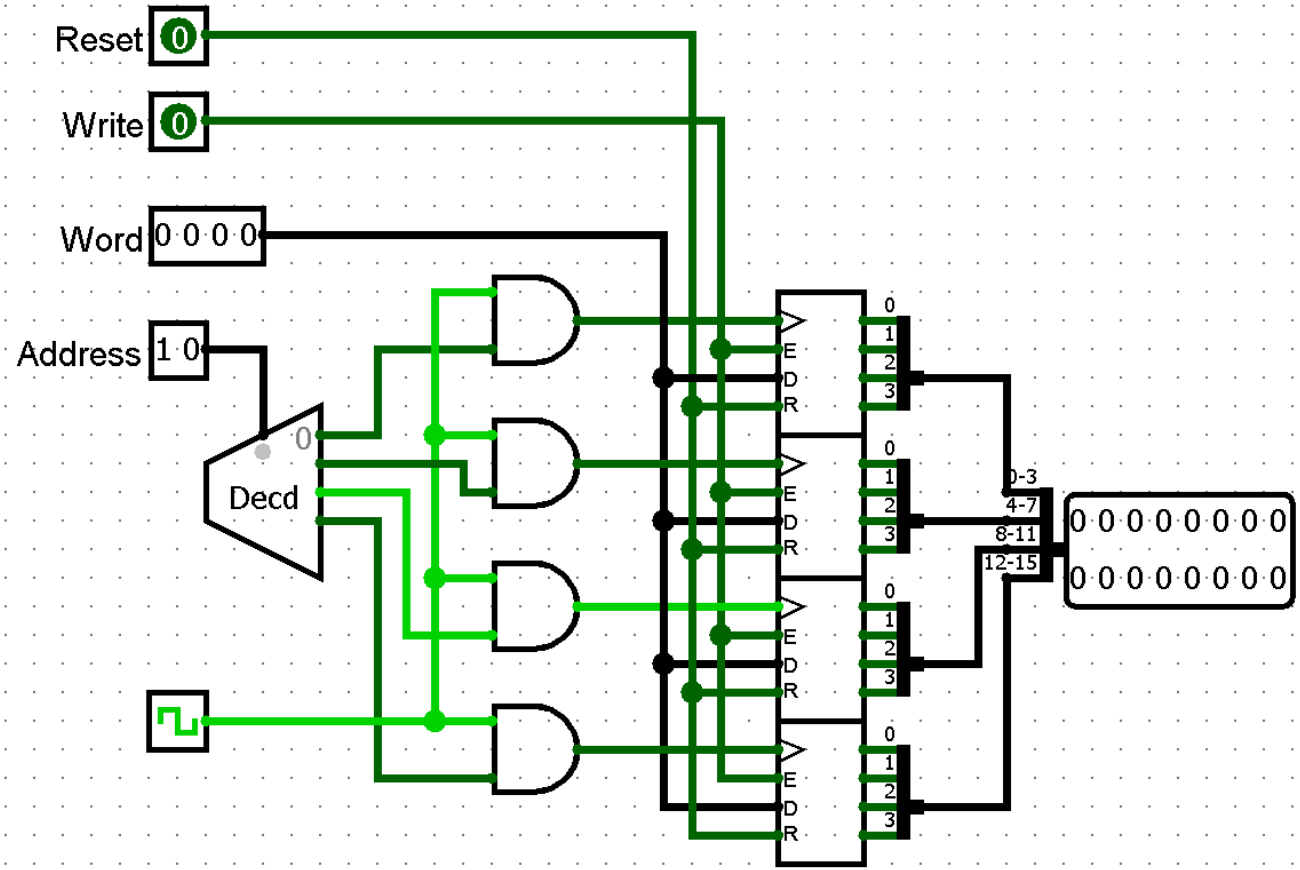
Esercizio 3

- Soluzione (circuito completo):



1) Il decoder è usato come “indirizzatore”
2) Uso splitter con diverse ampiezze ingresso/uscita per raggruppare meglio i 16 bit in uscita

Esercizio 4

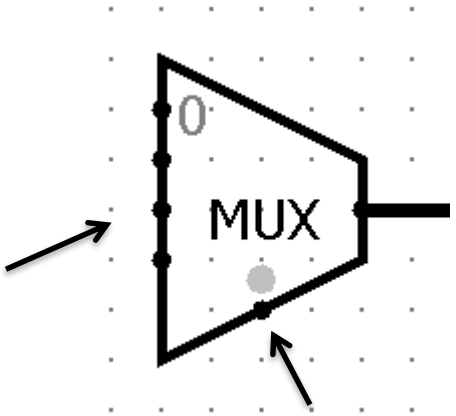


- Si estenda questo circuito in modo tale da poter effettuare la lettura indirizzata di uno dei quattro registri

Esercizio 4

- Dovremo aggiungere:

con un MUX a 4 vie selezionare quale dei 4 registri (4 bit) far uscire

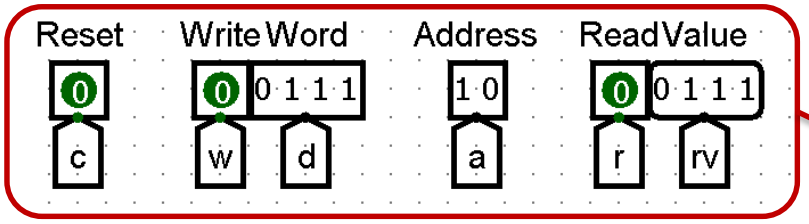


pin di output per il valore (4 bit) del registro letto

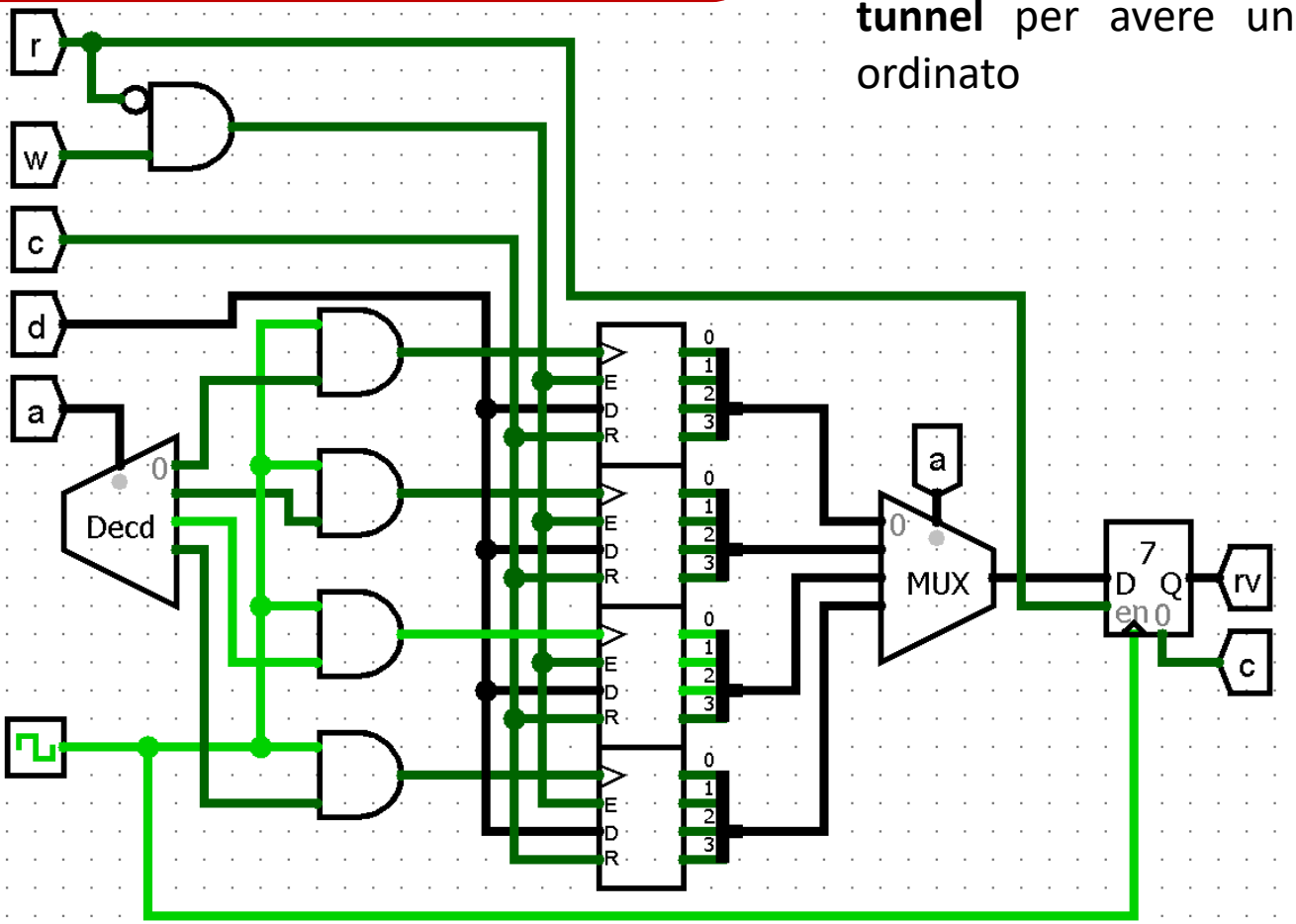
I bit di selezione del MUX coincideranno con l'indirizzo del registro che voglio leggere

bit di read: ponendolo a 1 attivo la modalità lettura e inibisco la scrittura

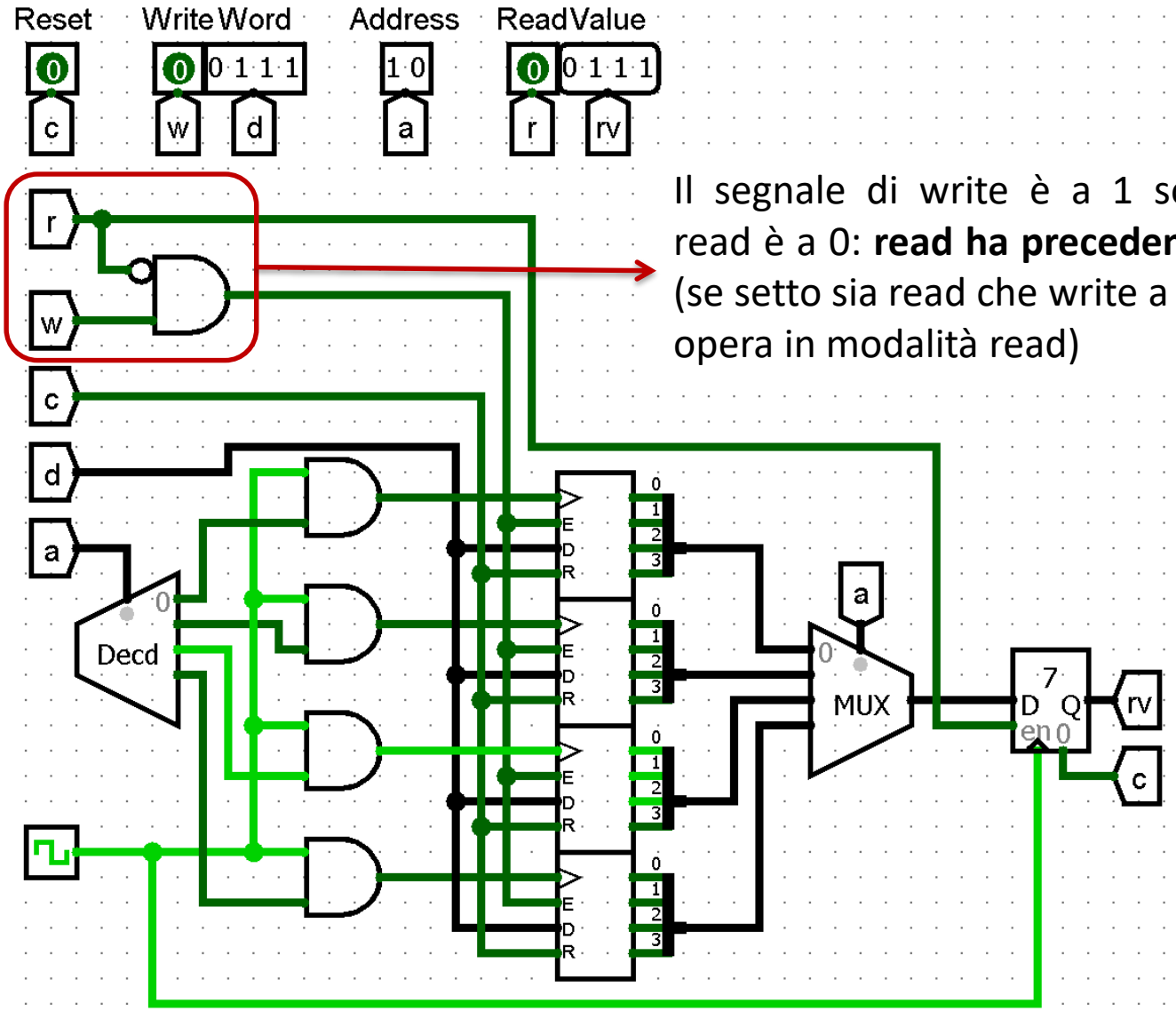
Esercizio 4



Raggruppo l'interfaccia utente del circuito in un unico punto distribuendo i segnali con l'uso di **tunnel** per avere un layout più ordinato

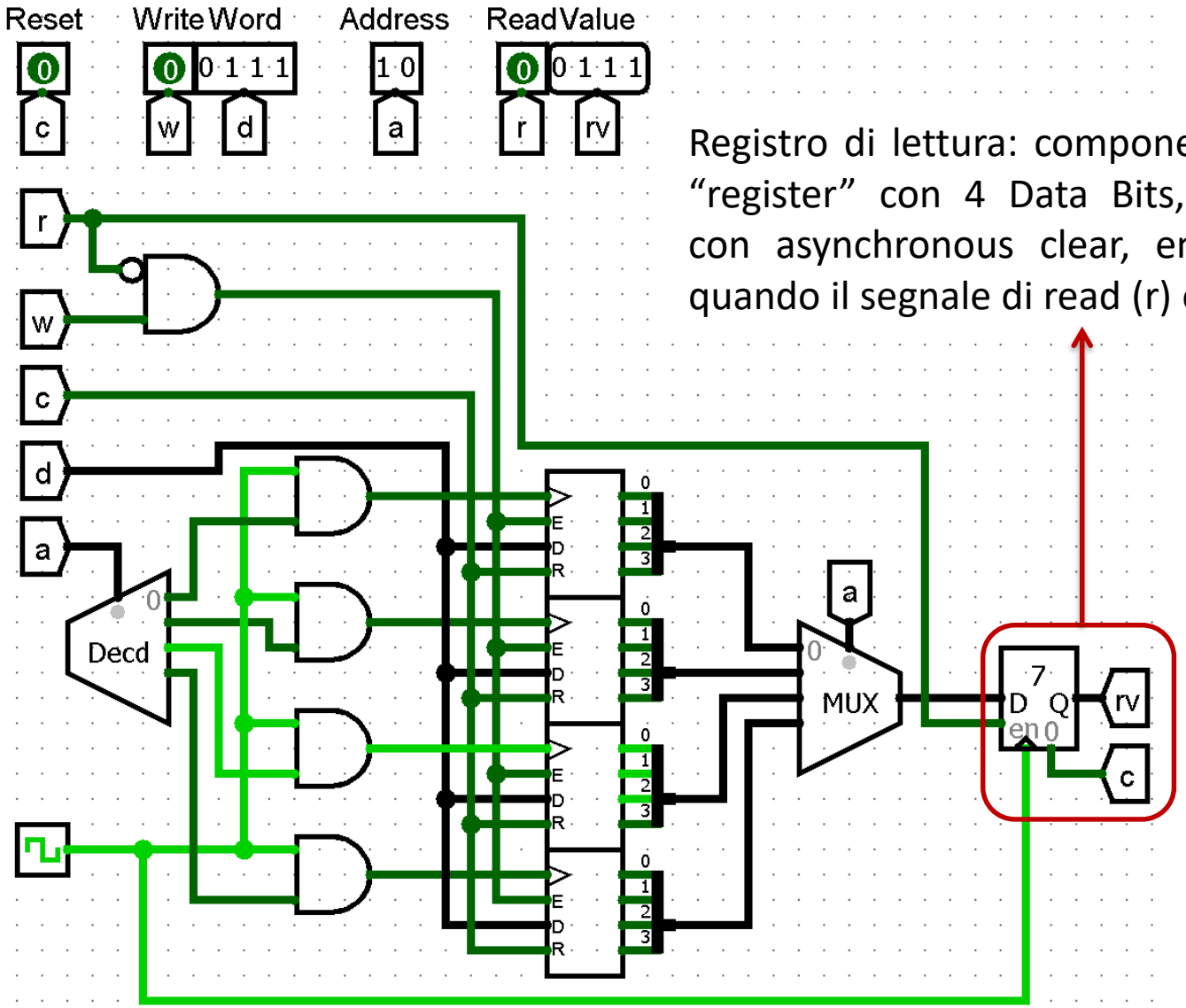


Esercizio 4



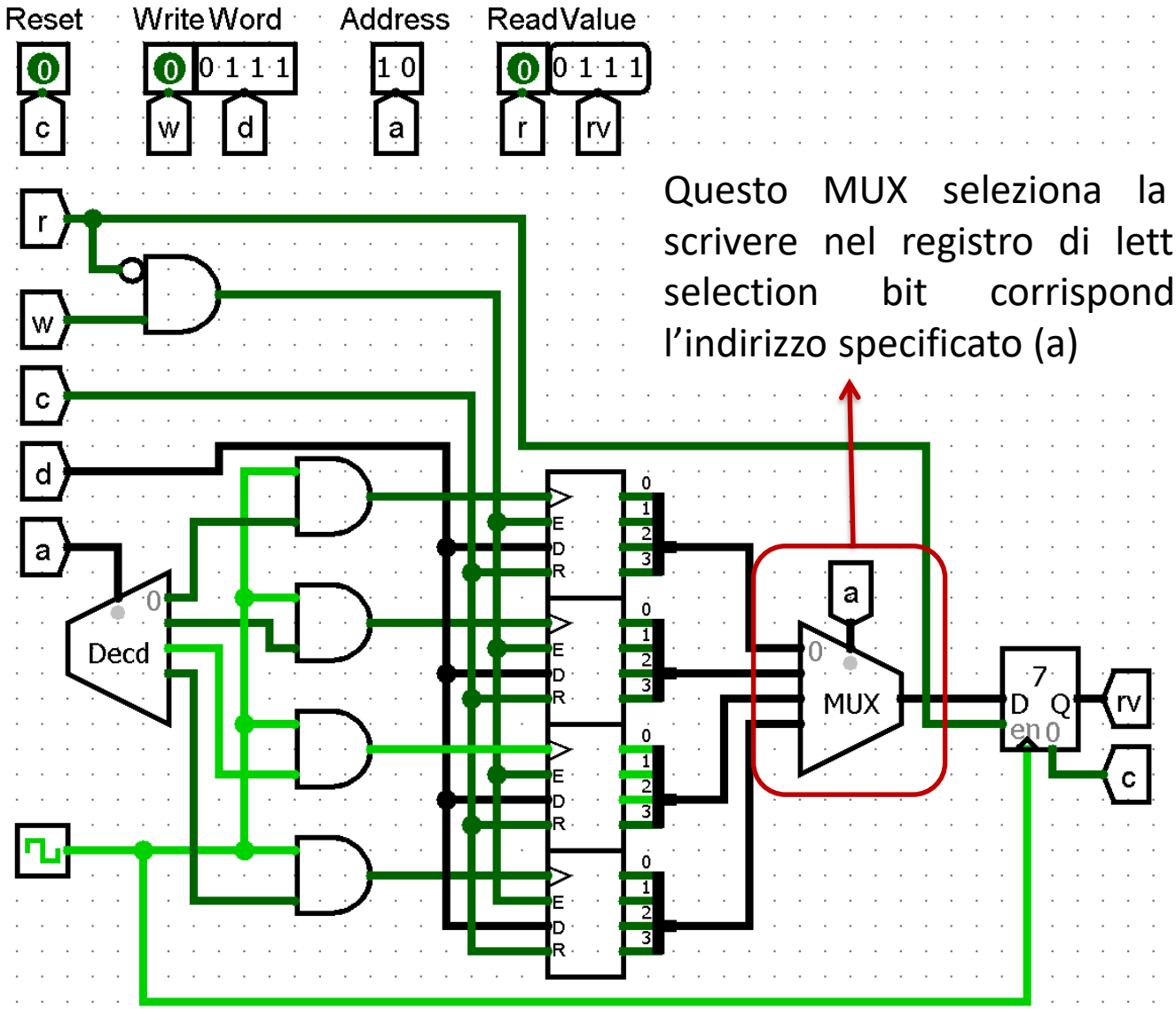
Il segnale di write è a 1 solo quando read è a 0: **read ha precedenza su write** (se setto sia read che write a 1, il circuito opera in modalità read)

Esercizio 4



Registro di lettura: componente logisim "register" con 4 Data Bits, resettabile con asynchronous clear, enabled solo quando il segnale di read (r) è 1

Esercizio 4



Questo MUX seleziona la word da scrivere nel registro di lettura. I due selection bit corrispondono con l'indirizzo specificato (a)

Laboratorio di Architetture degli Elaboratori I
Corso di Laurea in Informatica, A.A. 2019-2020
Università degli Studi di Milano