Laboratorio di Architetture degli Elaboratori I
Corso di Laurea in Informatica, A.A. 2018-2019
Università degli Studi di Milano

Esempio di progetto

Nicola Basilico
Dipartimento di Informatica
Via Comelico 39/41 - 20135 Milano (MI)
Ufficio S242
nicola.basilico@unimi.it
+39 02.503.16294

Esercizio 1

- Esempio di progetto base: realizziamo un circuito che riproduca il gioco «acchiappa la talpa»

- Come funziona? Ci sono n buche da cui, in modo imprevedibile, possono comparire talpe che restano visibili per un certo lasso di tempo. Lo scopo del gioco è colpire una talpa nel lasso di tempo in cui è visibile. Per ogni talpa colpita si ottiene un punto.

- Iniziamo interpretare questa specifica nei termini di un circuito logico e, soprattutto, considerando i componenti Logisim che conosciamo.

Esercizio 1

- Buca = led, la presenza/assenza della talpa è modellata con lo stato del led (acceso/spento)

Esercizio 1

- Buca = led, la presenza/assenza della talpa è modellata con lo stato del led (acceso/spento)

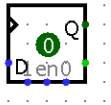
- Colpire la talpa = pressione di un bottone (input del circuito, uno per ogni buca)

Esercizio 1

- Buca = led, la presenza/assenza della talpa è modellata con lo stato del led (acceso/spento)

- Colpire la talpa = pressione di un bottone (input del circuito, uno per ogni buca)

- Stato del board di gioco = memoria da 1 bit (flip flop) per ogni buca, memorizzo un 1 se la talpa è visibile 0 altrimenti

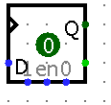


Esercizio 1

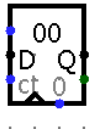
- Buca = led, la presenza/assenza della talpa è modellata con lo stato del led (acceso/spento)

- Colpire la talpa = pressione di un bottone (input del circuito, uno per ogni buca)

- Stato del board di gioco = memoria da 1 bit (flip flop) per ogni buca, memorizzo un 1 se la talpa è visibile 0 altrimenti



- Se premo il bottone di buca quando la talpa è presente guadagno un punto: stato del board e input determinano il punteggio (servirà contatore per punteggio)

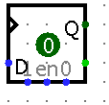


Esercizio 1

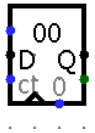
- Buca = led, la presenza/assenza della talpa è modellata con lo stato del led (acceso/spento)

- Colpire la talpa = pressione di un bottone (input del circuito, uno per ogni buca)

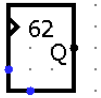
- Stato del board di gioco = memoria da 1 bit (flip flop) per ogni buca,
  memorizzo un 1 se la talpa è visibile 0 altrimenti



- Se premo il bottone di buca quando la talpa è presente guadagno un punto:
  stato del board e input determinano il punteggio (servirà contatore per punteggio)



- Le talpe compaiono in modo random, servirà un *random number generator*

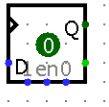


Esercizio 1

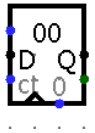
- Buca = led, la presenza/assenza della talpa è modellata con lo stato del led (acceso/spento)

- Colpire la talpa = pressione di un bottone (input del circuito, uno per ogni buca)

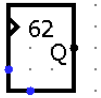
- Stato del board di gioco = memoria da 1 bit (flip flop) per ogni buca, memorizzo un 1 se la talpa è visibile 0 altrimenti



- Se premo il bottone di buca quando la talpa è presente guadagno un punto: stato del board e input determinano il punteggio (servirà contatore per punteggio)



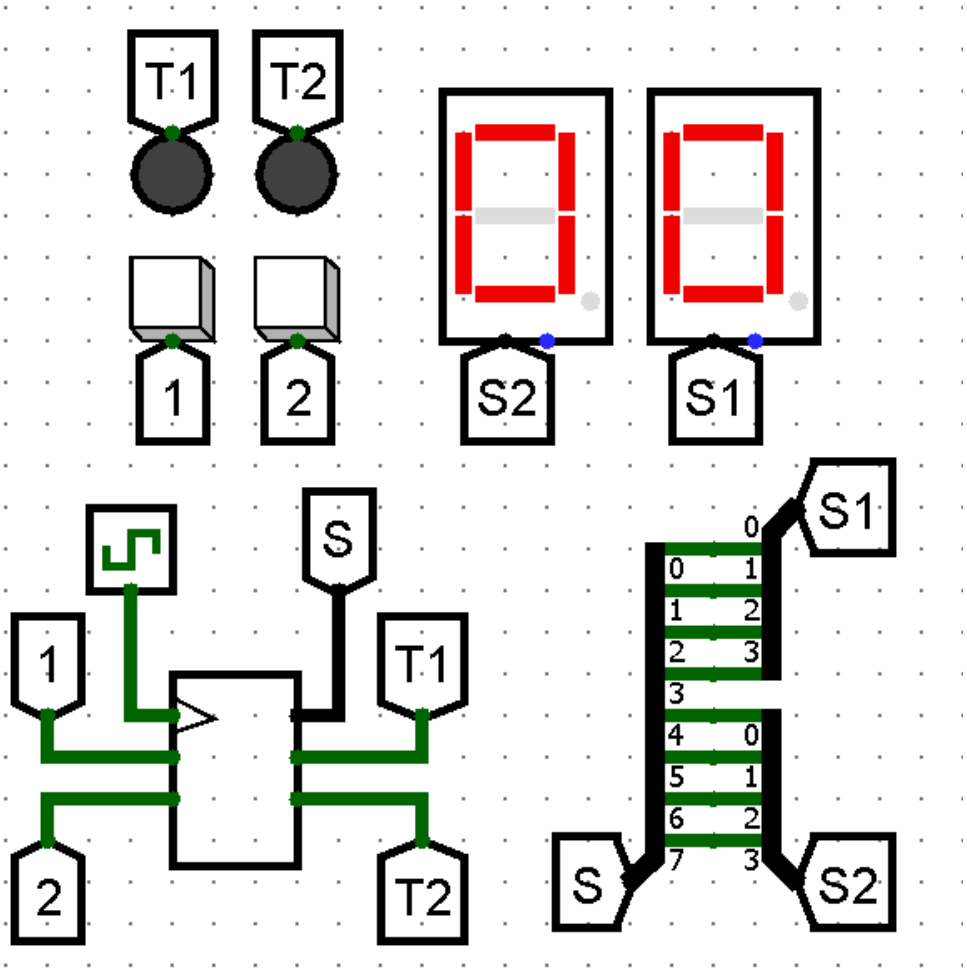
- Le talpe compaiono in modo random, servirà un *random number generator*



- Per semplicità assumiamo che ogni talpa una volta comparsa, resti visibile per un tempo fissato (esempio: 8 cicli di clock)

Esercizio 1

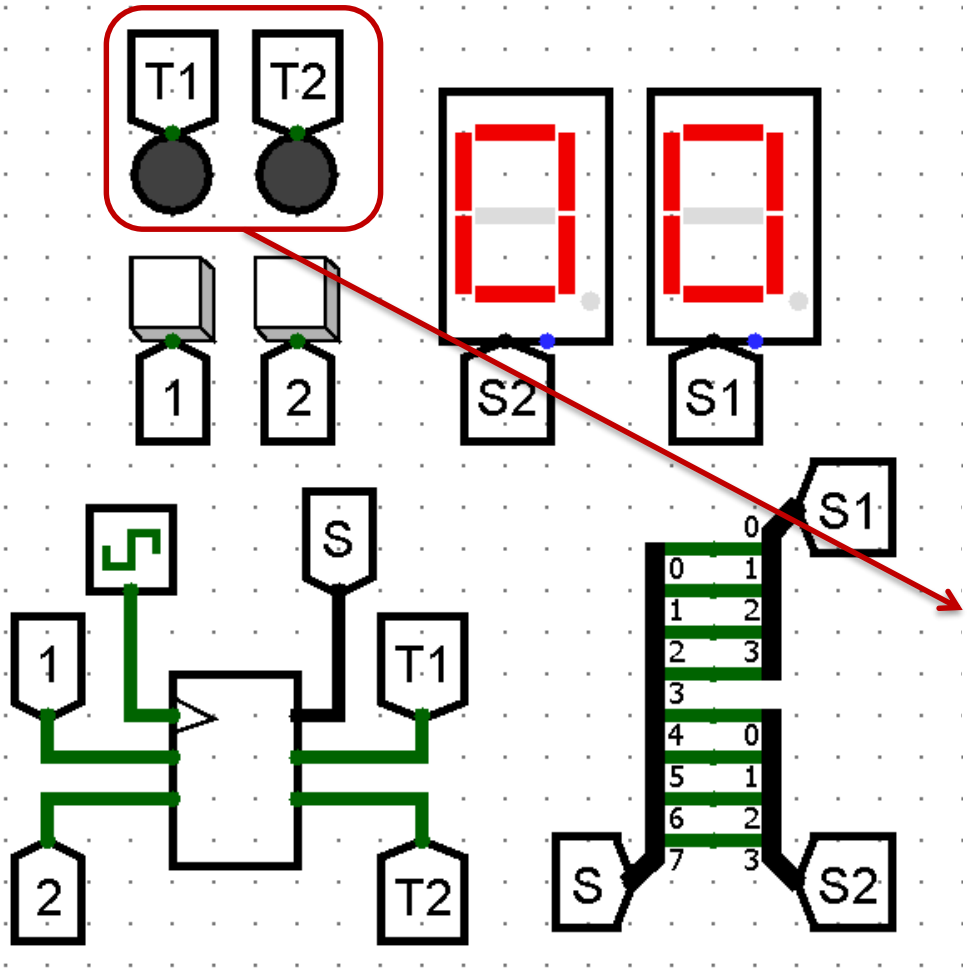
- Il circuito di «acchiappa la talpa», visuale esterna



Nomenclatura:
T1: talpa 1
T2: talpa 2
1: colpire buca 1
2: colpire buca 2
S: score su 8 bit
S1: 4 LSD dello score
S2: 4 MSD dello score

Esercizio 1

- Il circuito di «acchiappa la talpa», visuale esterna

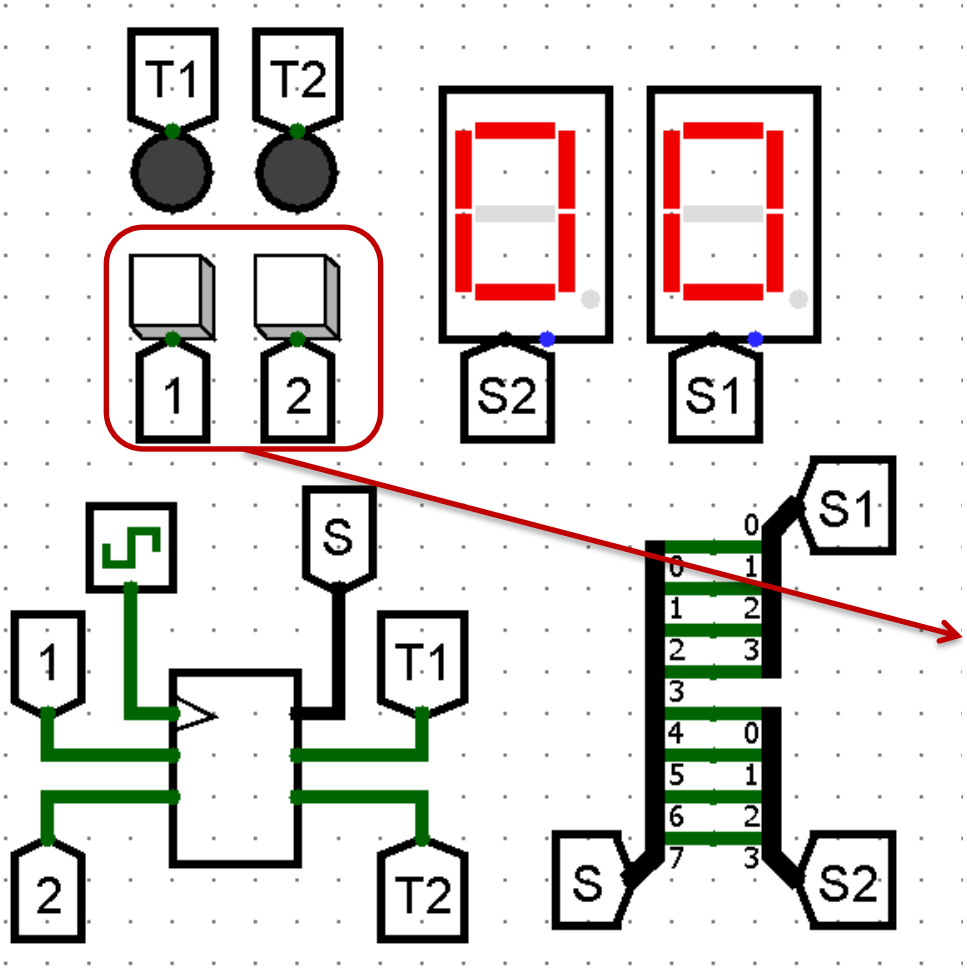


Nomenclatura:
T1: talpa 1
T2: talpa 2
1: colpire buca 1
2: colpire buca 2
S: score su 8 bit
S1: 4 LSD dello score
S2: 4 MSD dello score

I due led rappresentano le due buche: led acceso significa talpa visibile, led spendo significa talpa non visibile

Esercizio 1

- Il circuito di «acchiappa la talpa», visuale esterna

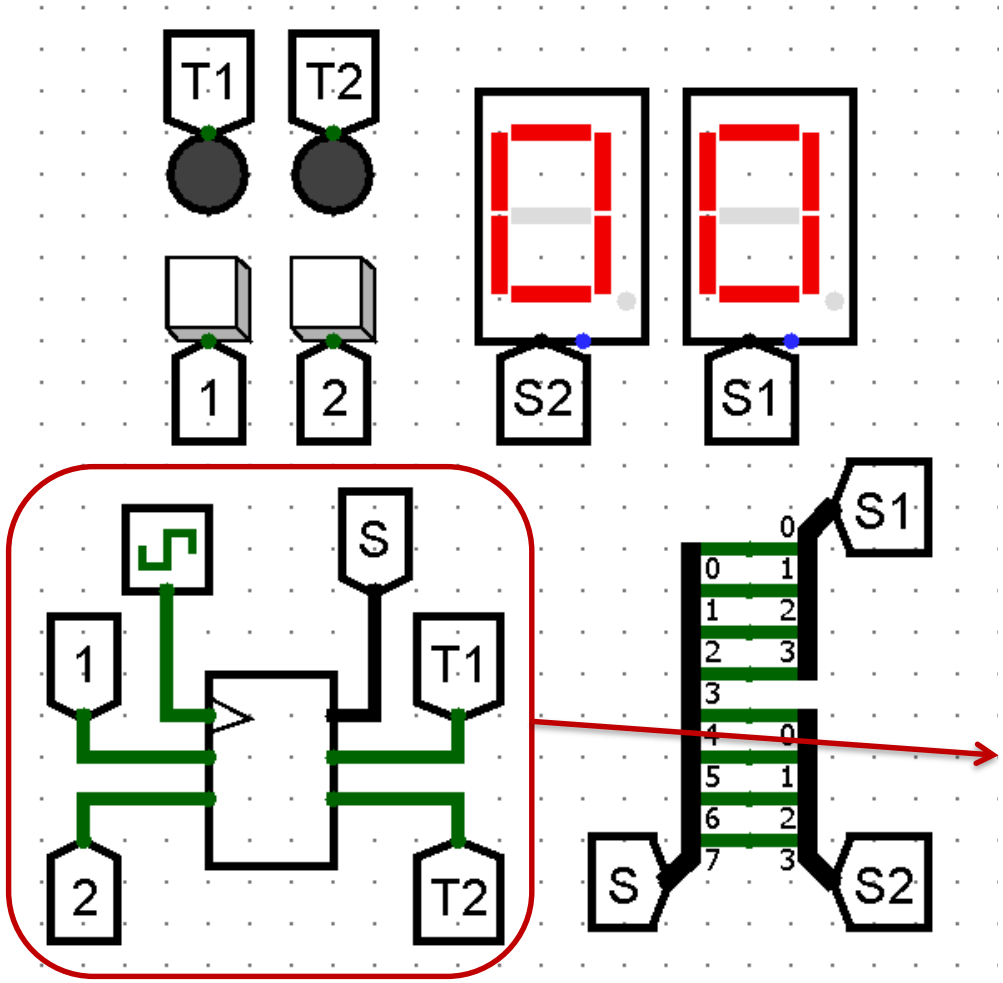


Nomenclatura:
T1: talpa 1
T2: talpa 2
1: colpire buca 1
2: colpire buca 2
S: score su 8 bit
S1: 4 LSD dello score
S2: 4 MSD dello score

I due bottoni verranno utilizzati per colpire le talpe: premere il bottone 1 significa colpire nella buca 1, premere il bottone 2 significa colpire nella buca 2

Esercizio 1

- Il circuito di «acchiappa la talpa», visuale esterna



Nomenclatura:
T1: talpa 1
T2: talpa 2
1: colpire buca 1
2: colpire buca 2
S: score su 8 bit
S1: 4 LSD dello score
S2: 4 MSD dello score

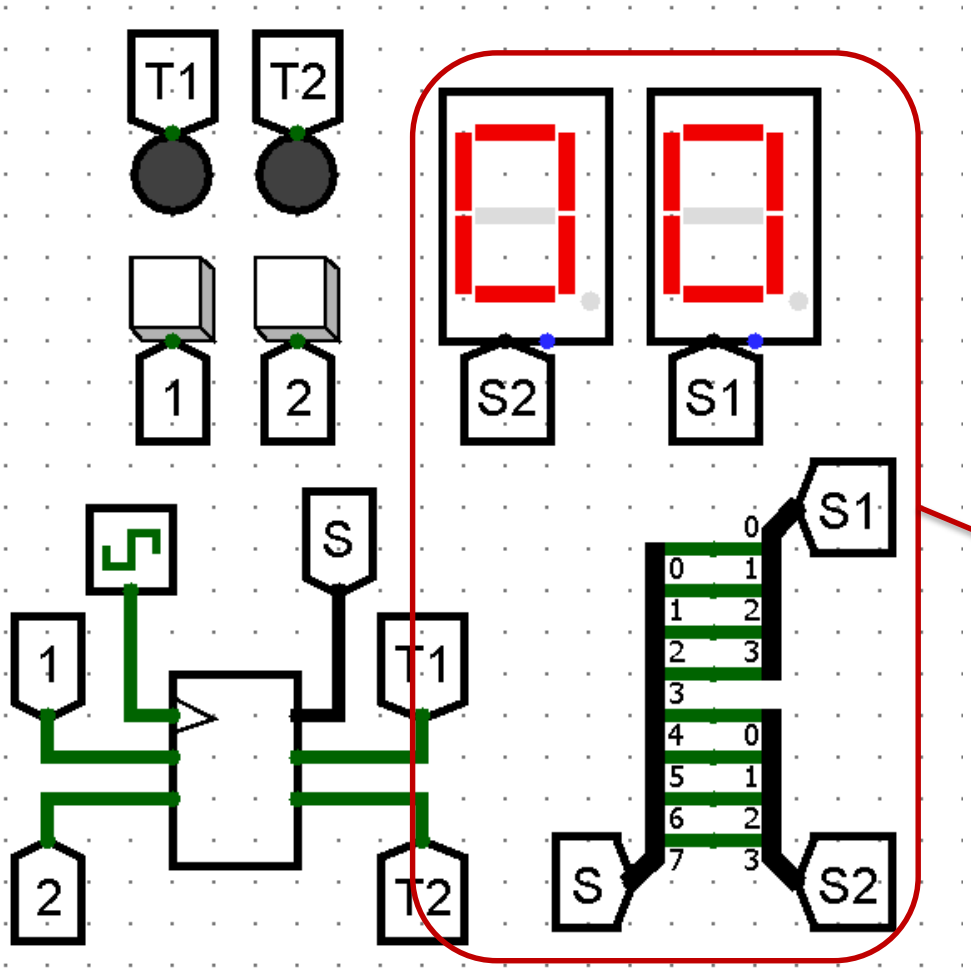
Il gioco è regolato da questa macchina dove:

- input: bottone 1 e 2
- stato: visibilità/non visibilità della talpa in buca 1 e 2, score corrente
- uscita: stato

Quale parte dello stato dipende dagli input e quale invece no?

Esercizio 1

- Il circuito di «acchiappa la talpa», visuale esterna



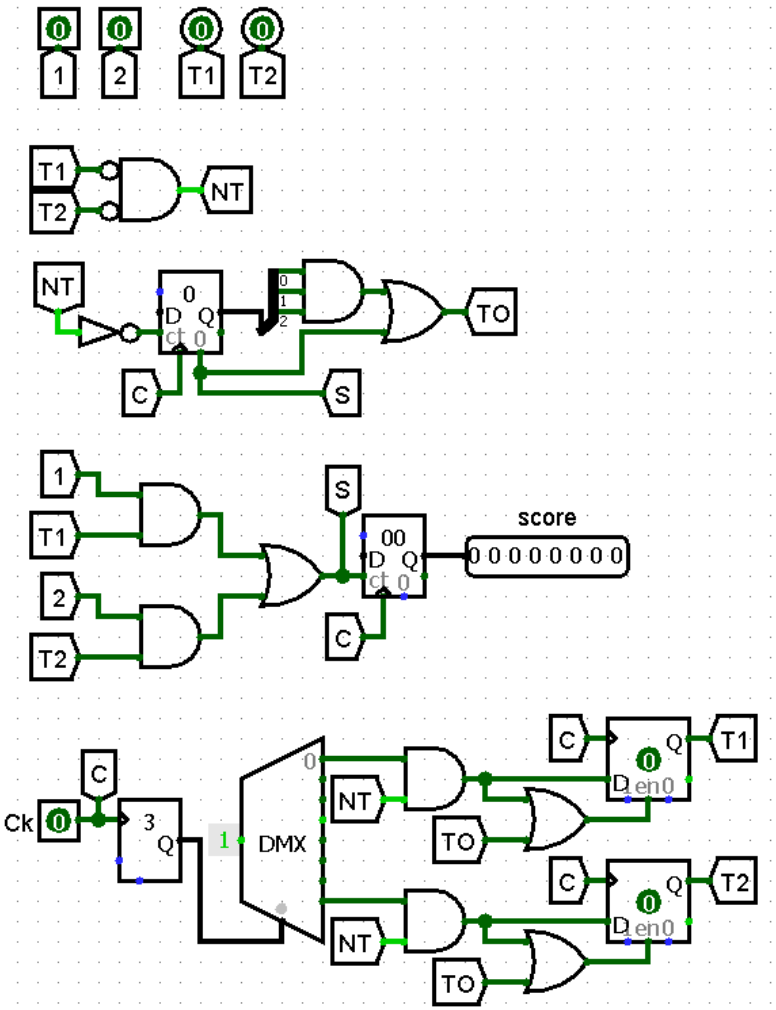
Nomenclatura:
T1: talpa 1
T2: talpa 2
1: colpire buca 1
2: colpire buca 2
S: score su 8 bit
S1: 4 LSD dello score
S2: 4 MSD dello score

Lo score S è su 16 bit, li splitto in due gruppi da 4 (S1, S2) per poterli visualizzare sui due Hex Digit Display

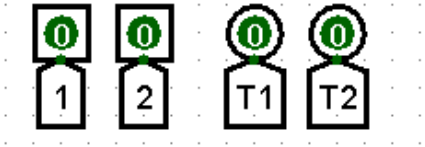
Esercizio 1

Soluzione

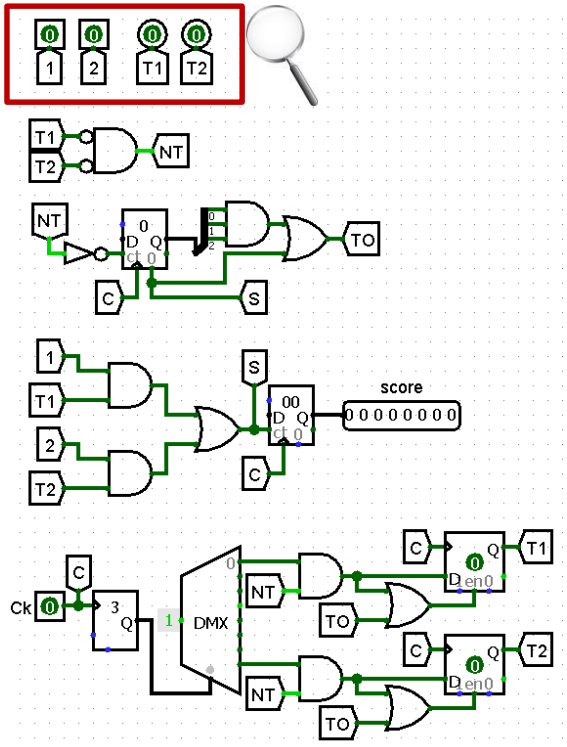
Analizziamola un sottocircuito alla volta



Esercizio 1

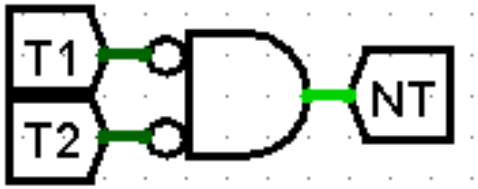


- **1** e **2** sono i pin di input per i due bottoni
- **T1** e **T2** sono i pin di output a cui collegheremo i led

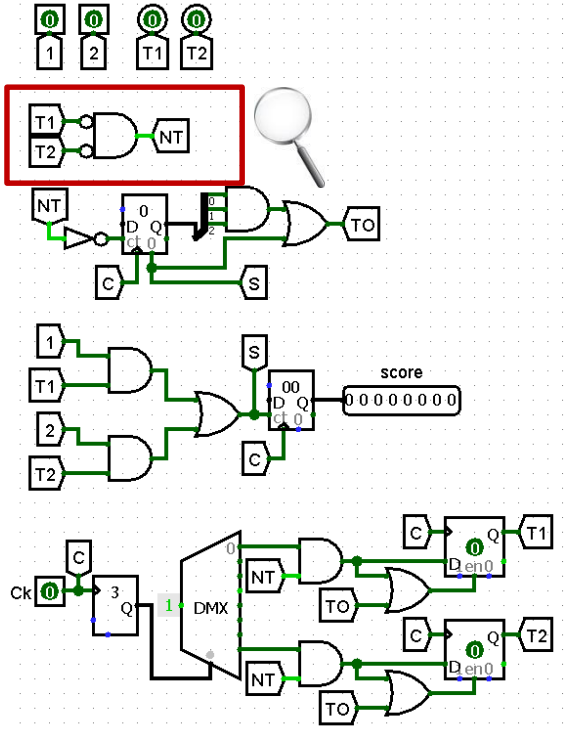


T1: talpa 1
T2: talpa 2
1: colpire buca 1
2: colpire buca 2
S: +1 score su 8 bit

Esercizio 1

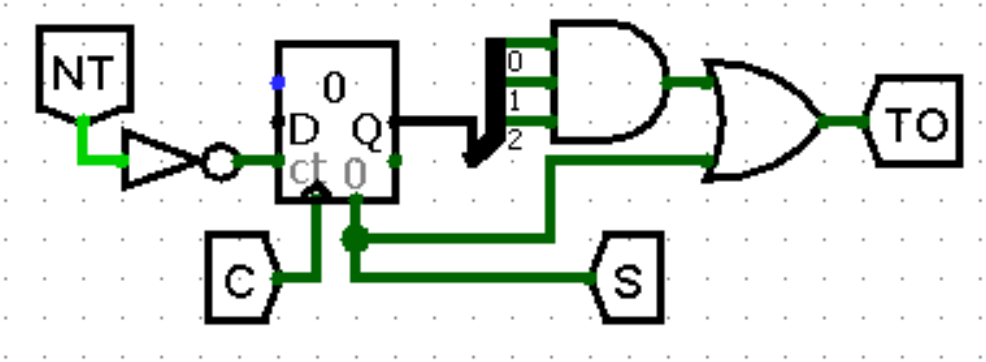


- Calcolo del Segnale **NT** = Nessuna Talpa visibile



T1: talpa 1
T2: talpa 2
1: colpire buca 1
2: colpire buca 2
NT: no talpe

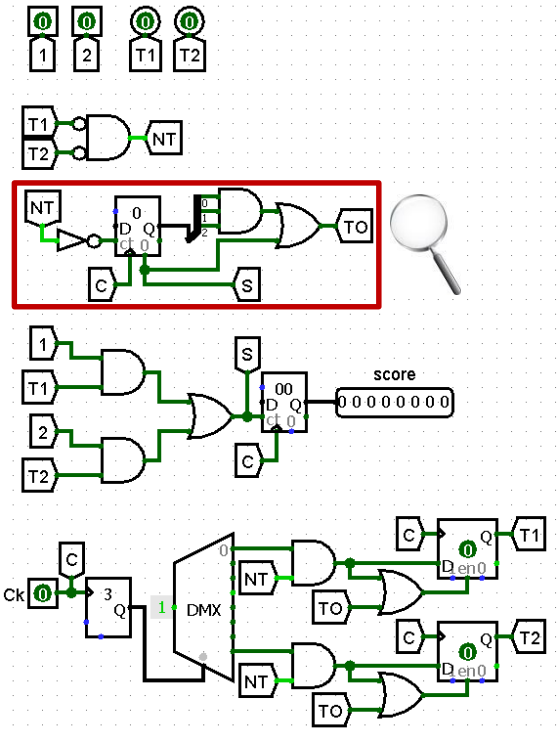
Esercizio 1



Gestione del tempo di visibilità della talpa

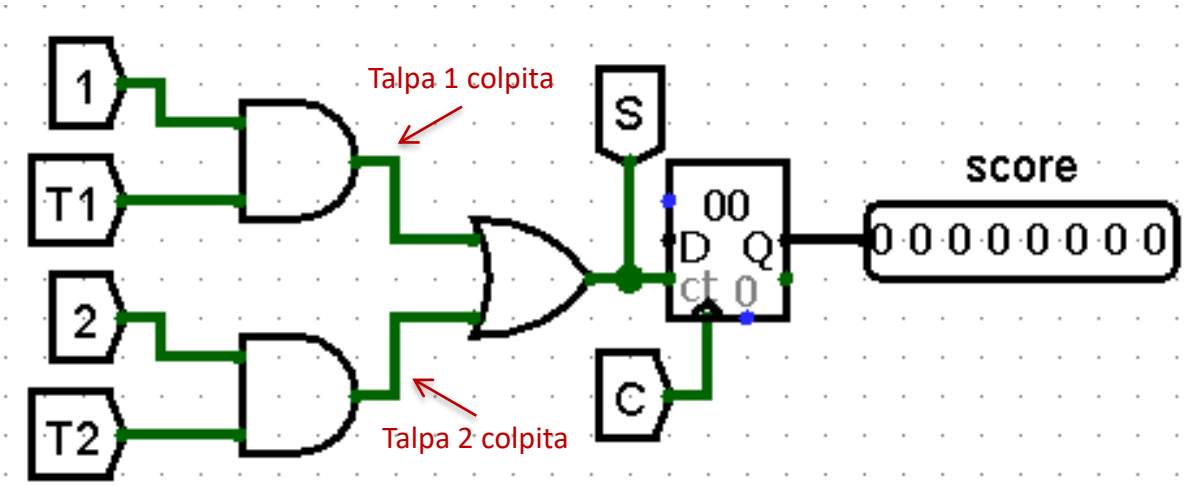
Counter:

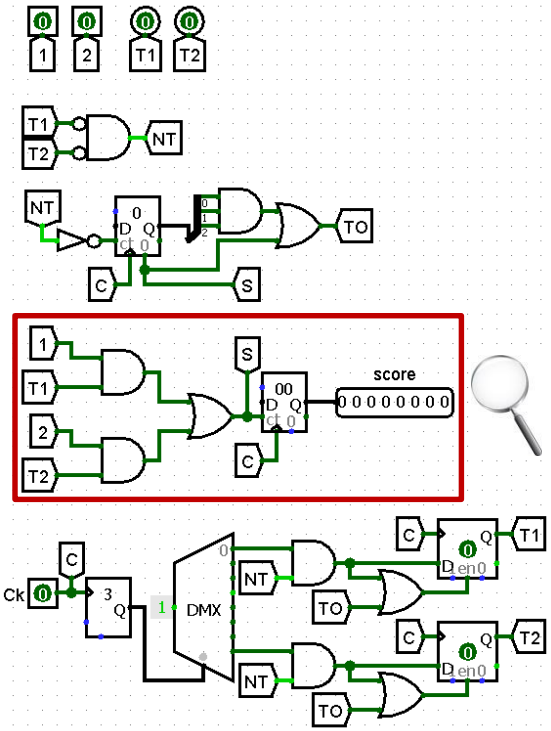
- conta il tempo di visibilità su 3 bit
- per ogni ciclo di clock in cui c'è almeno una talpa (**NT** = 0) viene incrementato di 1 **(nel circuito non ci potrà essere più di una talpa visibile, almeno una talpa = esattamente una talpa)**
- se guadagno un punto (colpisco la talpa, **S** = 1), resetto il counter
- se il counter raggiunge il valore massimo (**Q** = 111) **OR** colpisco la talpa (**S**=1) genero segnale **TO** (time out)



T1: talpa 1
T2: talpa 2
1: colpire buca 1
2: colpire buca 2
NT: no talpe
C: clock
TO: time out
S: +1 score su 8 bit

Esercizio 1



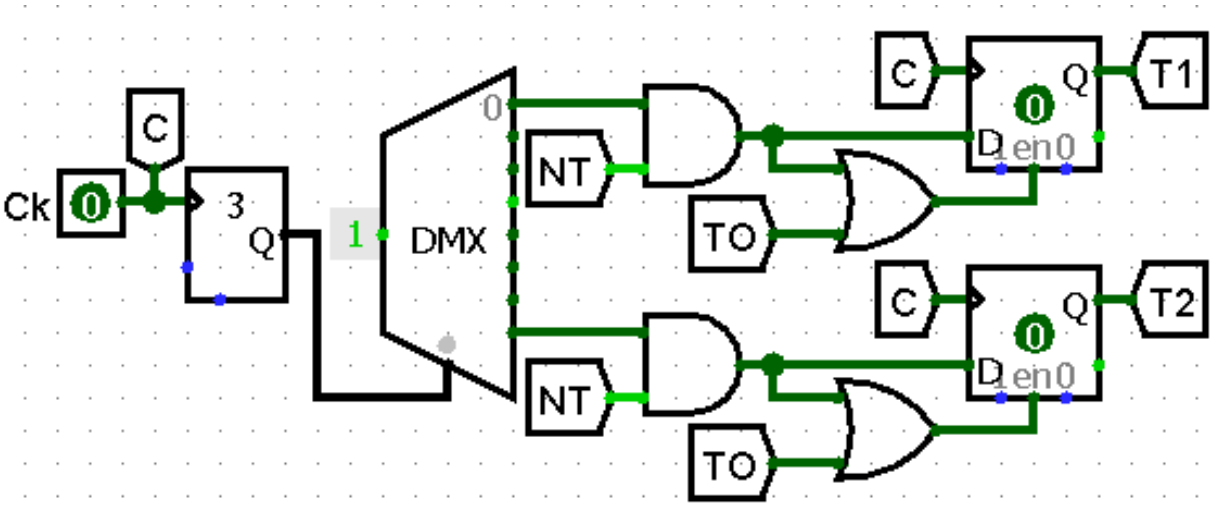


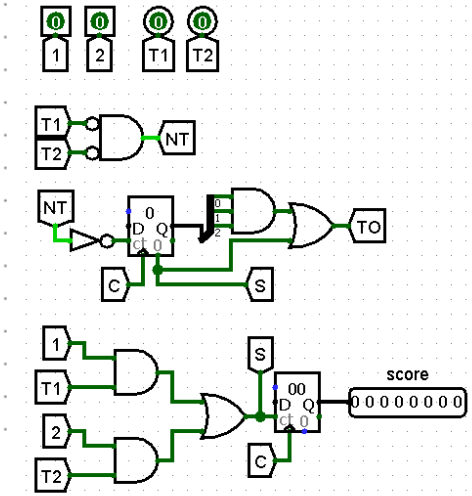
Gestione del punteggio:

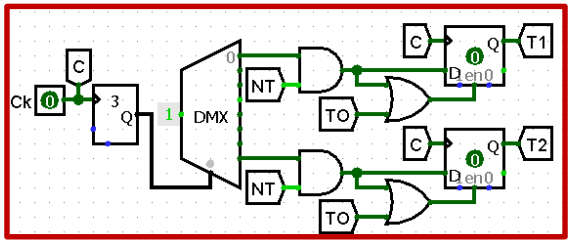
- colpisco la talpa 1: **1** = 1 **AND T1** = 1
- colpisco la talpa 2: **2** = 1 **AND T2** = 1
- se *(colpisco la talpa)* 1 **OR** *(colpisco la talpa 2)* allora **S**=1 e incremento di 1 il contatore dello score

T1: talpa 1
T2: talpa 2
1: colpire buca 1
2: colpire buca 2
NT: no talpe
C: clock
S: +1 score su 8 bit

Esercizio 1





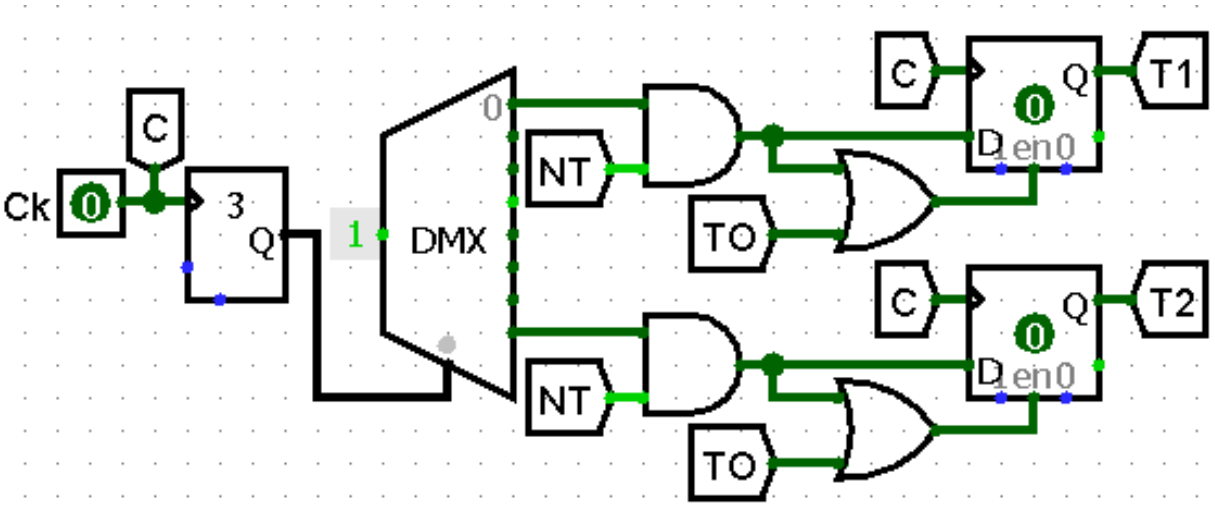


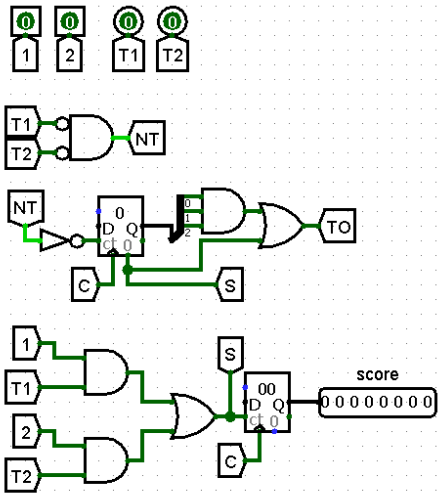
Gestione della comparsa random delle talpe:

- A ogni ciclo di clock genero un numero random tra 0 e 7, lo collego a un DEMUX ponendo a 1 una di otto vie in modo random; solo due vie possono essere accese = solo due talpe
- Se la linea del DEMUX è accesa **AND** non ci sono talpe, faccio *enable* del registro della buca corrispondente e scrivo 1
- Se c'è stato un time out allora per ciascun registro (buca):
  - se la linea del DEMUX è spenta, scrivo 0 nel registro (la talpa scompare)
  - se la linea del DEMUX è accesa, scrivo 1 nel registro (la talpa ricompare subito)

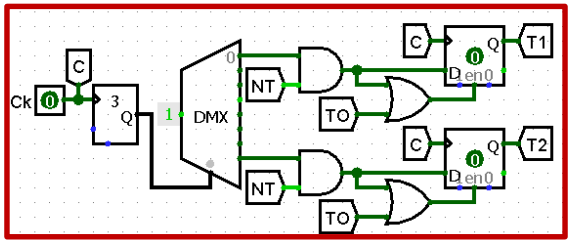
T1: talpa 1
T2: talpa 2
1: colpire buca 1
2: colpire buca 2
NT: no talpe
TO: time out
C: clock
S: +1 score su 8 bit

Esercizio 1





NOTA:
Dimensionando opportunamente il numero di data bits del random generator e il numero di vie del DEMUX possiamo controllare la probabilità di comparsa delle talpe



T1: talpa 1
T2: talpa 2
1: colpire buca 1
2: colpire buca 2
NT: no talpe
TO: time out
C: clock
S: +1 score su 8 bit

Laboratorio di Architetture degli Elaboratori I
Corso di Laurea in Informatica, A.A. 2018-2019
Università degli Studi di Milano

Nicola Basilico
Dipartimento di Informatica
Via Comelico 39/41 - 20135 Milano (MI)
Ufficio S242
nicola.basilico@unimi.it
+39 02.503.16294

Hanno contribuito alla realizzazione di queste slides:
Dr. Alberto Quattrini Li