

\_\_\_\_\_

Il progetto propone una versione in solitario del popolare gioco ormai vintage. L'input avviene attraverso il joystick, l'output attraverso matrici led e display 7 segmenti, più una piccola tastiera.

Una volta attivato il clock (ctrl+k) compare un pallino che rimbalza da una parte all'altra dello schermo, se una volta raggiunto il limite tocca una delle sponde mosse dal joystick la macchina a stati finiti (un contatore da 0 a 9) incrementa il proprio stato, facendo visualizzare sul display a 7 segmenti il punteggio effettivo.

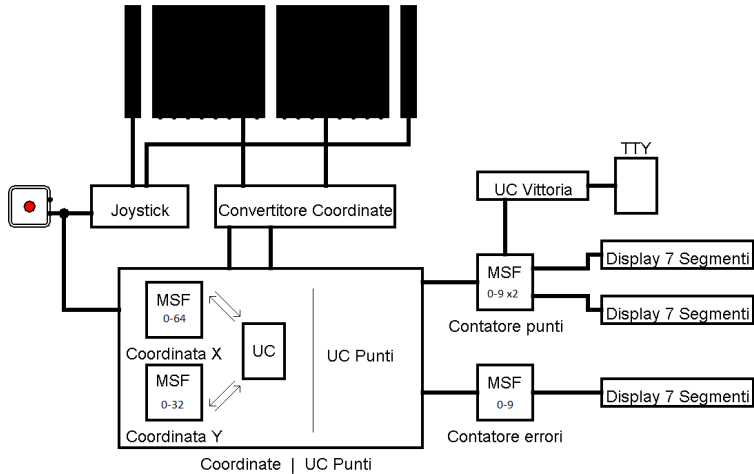
Qualora il giocatore mancasse il colpo verrà incrementato un altro contatore, quello degli errori.

Il giocatore vince raggiungendo il punteggio di 20, con 10 errori tuttavia il punteggio verrà resettato e il gioco dovrà ripartire daccapo.

Raggiunto il punteggio di 20 sulla tastiera apparirà la scritta "Hai vinto!"

Oltre al contatore da 0 a 10 vi è un'altra macchina a stati finiti, un contatore su 6 bit che controlla la posizione del pallino.

Qui lo schema a blocchi del circuito, con le descrizioni dei blocchi:



## Joystick:

Blocco che converte l'input del joystick in due sponde mobili sincronizzate.

L'input è a 5 bit, ed è la posizione della manopola, un numero da 1 a 32 (salta lo zero per avere un numero dispari di possibilità e quindi una posizione centrale).

Il Decoder collegato allo splitter trasforma l'input in una serie di zeri ed un unico uno, quello della posizione della monopola.

I blocchi successivi servono a trasformarlo in una sequenza di 5 "uni", il centrale dei quali è l'originale.

## 7Segmenti:

La funzione che associa un numero alla cifra sul display 7 segmenti è codificata attraverso 7 circuiti:

$$A = \neg n_3 \neg n_2 + \neg n_3 \neg n_1 \neg n_0 + \neg n_2 \neg n_1 + \neg n_3 n_1 n_0$$

$$B = \neg n_2 \neg n_1 + \neg n_3 n_0 + \neg n_3 n_2$$

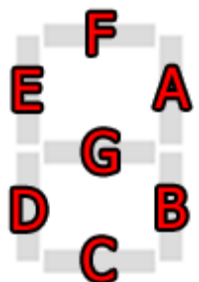
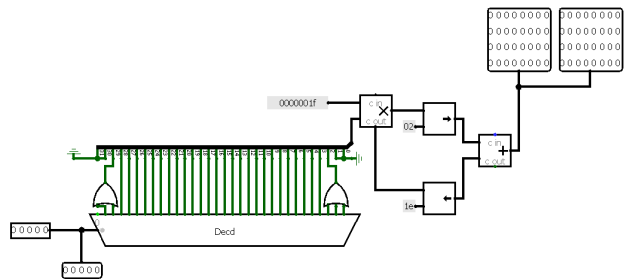
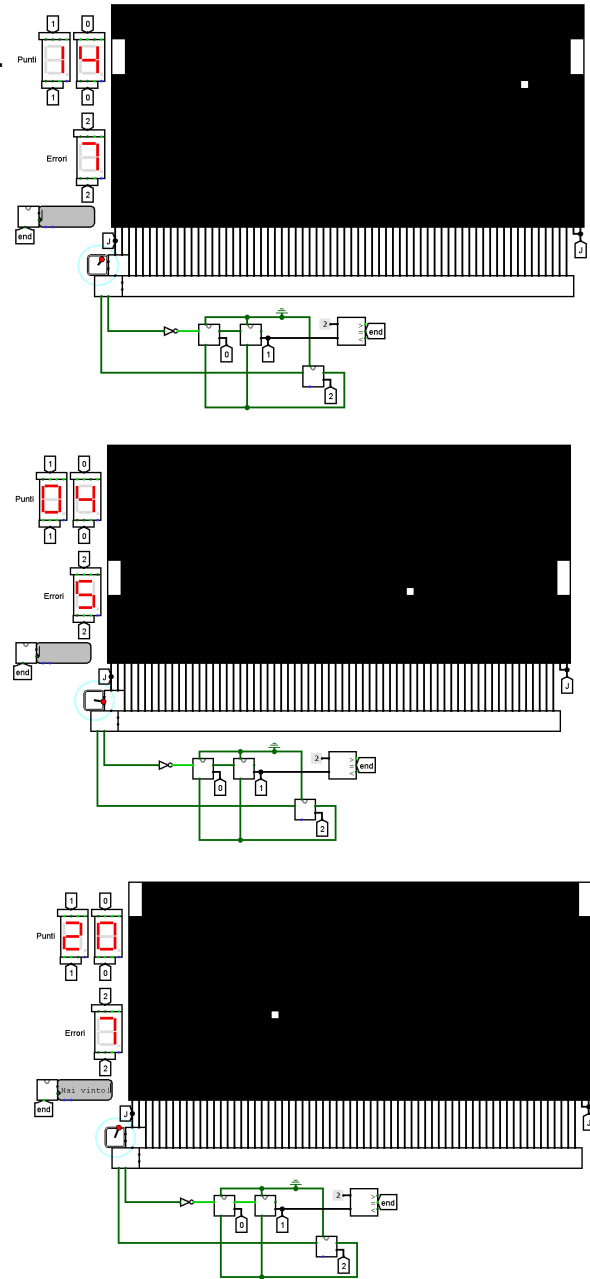
$$C = \neg n_3 \neg n_2 \neg n_0 + \neg n_3 \neg n_2 n_1 + \neg n_3 n_1 \neg n_0 + \neg n_3 n_2 \neg n_1 n_0 + n_3 \neg n_2 \neg n_1$$

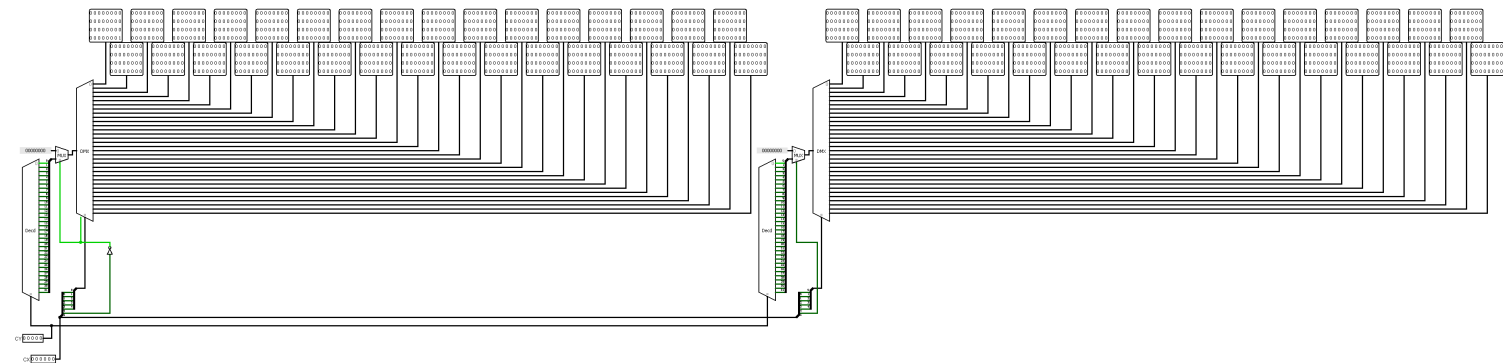
$$D = \neg n_2 \neg n_1 \neg n_0 + \neg n_3 n_1 \neg n_0$$

$$E = \neg n_3 \neg n_1 \neg n_0 + \neg n_3 n_2 \neg n_1 + \neg n_3 n_2 \neg n_0 + n_3 \neg n_2 \neg n_1$$

$$F = \neg n_3 \neg n_2 \neg n_0 + \neg n_3 n_1 + \neg n_3 n_2 n_0 + n_3 \neg n_2 \neg n_1$$

$$G = \neg n_3 \neg n_2 n_1 + \neg n_3 n_2 \neg n_1 + \neg n_3 n_2 \neg n_0 + n_3 \neg n_2 \neg n_1 + n_3 \neg n_2 \neg n_0$$





### Convertitore coordinate:

Un blocco che converte i numeri a 6 bit delle coordinate in input che la matrice possa visualizzare.

La coordinata Y viene convertita allo stesso modo che nel blocco Joystick, la coordinata X fa da selettore per un Demultiplexer che decide in quale posizione mandare la colonna.

Poiché in Logisim non si può creare un Demultiplexer a 6 bit ho collegato due Demultiplexer che lavorano con i primi 5 bit del valore, l'ultimo bit fa da enable per i due Demultiplexer.

$$X'_0 = \neg X_0$$

$$X'_1 = (\neg I \neg X_0 X_1) + (\neg I X_0 \neg X_1) + (I \neg X_0 \neg X_1) + (I X_0 X_1)$$

$$X'_2 = (\neg I \neg X_0 X_2) + (X_0 \neg X_1 X_2) + (\neg I X_0 X_1 \neg X_2) + I \neg X_0 \neg X_1 \neg X_2 + (I X_1 X_2)$$

$$X'_3 = (\neg I \neg X_0 X_3) + (\neg I \neg X_1 X_3) + (X_1 \neg X_2 X_3) + (X_0 \neg X_1 X_3) + (\neg I X_0 X_1 X_2 \neg X_3) + (I \neg X_0 \neg X_1 \neg X_2 \neg X_3) + (I X_2 X_3)$$

$$X'_4 = (\neg I \neg X_1 X_4) + (\neg I \neg X_3 X_4) + (\neg X_0 X_3 X_4) + (X_1 \neg X_2 X_4) + (X_0 \neg X_1 X_4) + (\neg I X_0 X_1 X_2 X_3 \neg X_4) + (I \neg X_0 \neg X_1 \neg X_2 \neg X_3 \neg X_4) + (I X_2 X_4)$$

$$X'_5 = (\neg I \neg X_0 X_5) + (\neg I \neg X_2 X_5) + (\neg I \neg X_3 X_5) + (\neg I \neg X_4 X_5) + (\neg X_0 X_4 X_5) + (\neg X_0 X_3 X_5) + (\neg X_0 X_2 X_5) + (X_0 \neg X_1 X_5) + (\neg I X_0 X_1 X_2 X_3 X_4 \neg X_5) + (I \neg X_0 \neg X_1 \neg X_2 \neg X_3 \neg X_4 \neg X_5) + (I X_1 X_5)$$

	S <sub>0</sub>				I				0				1			
	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	1	0
1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0
2	0	0	1	0	0	0	1	1	0	0	0	0	1	0	0	1
3	0	0	1	1	0	1	0	0	0	0	1	0	0	1	0	0
4	0	1	0	0	0	1	0	1	0	0	1	0	1	0	1	1
5	0	1	0	1	0	1	1	0	0	1	0	0	1	0	0	0
6	0	1	1	0	0	1	1	1	0	1	1	0	1	0	1	1
7	0	1	1	1	1	0	0	0	0	0	1	1	0	1	1	0
8	1	0	0	0	1	0	0	1	0	1	0	1	1	1	1	1
9	1	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0

$$X'_0 = \neg X_0$$

$$X'_1 = (\neg I \neg X_0 X_1 \neg X_3) + (\neg I X_0 \neg X_1 \neg X_3) + (I \neg X_0 \neg X_1 X_3) + (I \neg X_0 \neg X_1 X_2) + (I X_0 X_1)$$

$$X'_2 = (\neg I \neg X_0 X_2) + (X_0 \neg X_1 X_2) + (\neg I X_0 X_1 \neg X_2) + (I \neg X_0 \neg X_2 X_3) + (I X_1 X_2)$$

$$X'_3 = (\neg I \neg X_0 \neg X_1 \neg X_2 X_3) + (\neg I X_0 X_1 X_2 \neg X_3) + (I \neg X_0 \neg X_1 \neg X_2 \neg X_3) + (I X_0 \neg X_1 \neg X_2 X_3)$$

$$\text{NEXT} = X_0 \neg X_1 \neg X_2 X_3$$

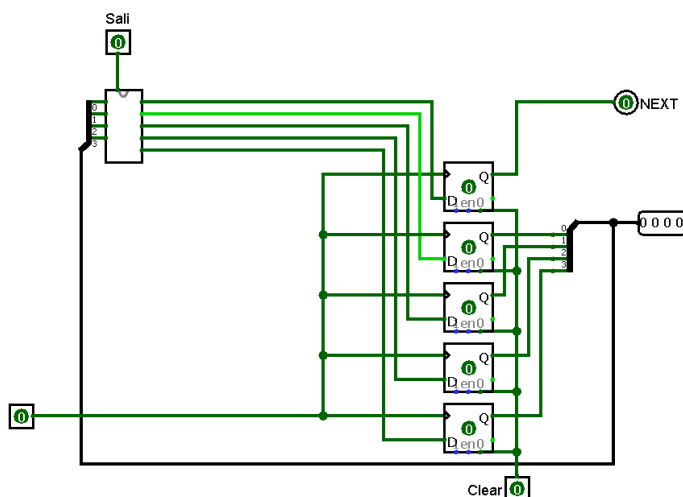
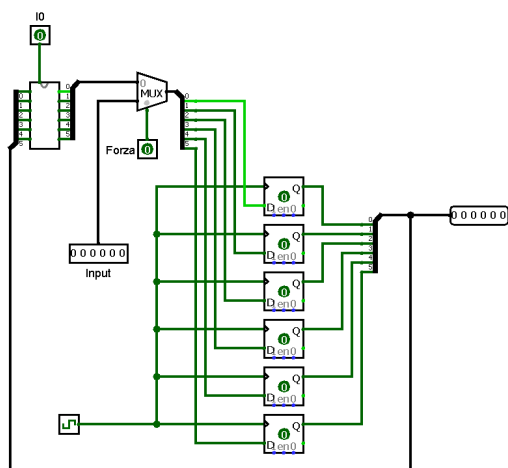
### Stato prossimo 64 e Stato prossimo 10:

I due blocchi Stato Prossimo definiscono le due funzioni contenute nelle due macchine a stati finiti.

In entrambi i casi lo stato prossimo è il numero successivo con input 0 e il numero precedente con input 1, in modulo 64 su 6 bit e 10 su 4 bit.

Qui sopra vi sono le funzioni stato prossimo della macchina su 6 bit,

per ovvie ragioni di spazio non ho messo la tabella della verità; a destra invece ci sono sia tabella che funzione logica degli stati prossimi più un output "Next" che si verifica quando si passa da 9 a 0, serve ad incrementare la seconda cifra del contatore dei punti.

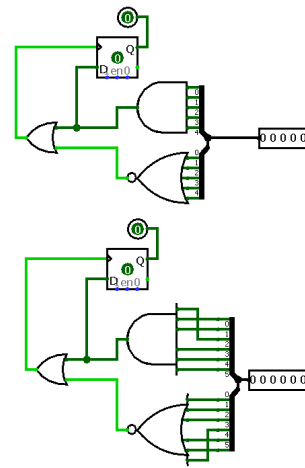


### MSF 64 e MSF 10:

Le Macchine sono molto simili, come ogni macchina di Huffman hanno un flip-flop D per ogni bit della funzione stato prossimo. Il contatore mod 64 ha il clock incorporato, a differenza di quello mod 10 che è esterno.

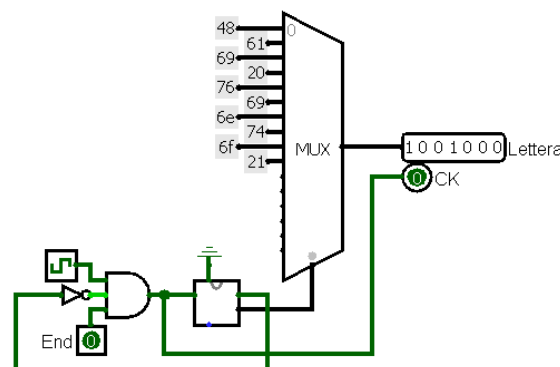
Il contatore mod 64 ha anche un bit di ingresso "Forza", che sostituisce lo stato prossimo con uno ricevuto come input.

Il contatore mod 10 ha un bit Clear che svuota il contenuto dei flip-flop riportando la macchina allo stato iniziale "0000".



L'output punto invece è uno quando il pallino è in prima o ultima colonna e la distanza è compresa tra 2 e -2 estremi inclusi. L'output "errore" non è solo portato esternamente per contare gli errori ma anche internamente per far ripartire il gioco riposizionando il pallino davanti alla sponda più vicina (non immediatamente davanti per non incrementare anche il punteggio ad ogni errore)

Lo stato del contatore fa da selezione di un multiplexer che prende come input, in ordine, le costanti su 7 bit che in ASCII rappresentano le lettere. Quando il giocatore raggiunge il punteggio 20 un comparatore dà segnale 1 all'input "End" del blocco, questo in 10 cicli di clock fa entrare nella tastiera, una alla volta, ciascuna lettera. Finite le lettere l'output "next" del contatore blocca definitivamente la porta And per non incorrere in errori (Quando si tenta di aggiungere una lettera ad una tastiera piena la simulazione viene interrotta).



In rari casi se la sponda è in movimento molto rapido può capitare di prendere il pallino all'ultimo momento, questo fa incrementare sia il contatore degli errori che quello del punteggio, sembrerebbe un problema irrisolvibile per sua stessa natura, visto che lo spostamento della sponda avviene a cavallo tra un clock e l'altro.

