PROGETTO ARCHITETTURA I

FLAPPY BIRD

INTRODUZIONE

Questo progetto prende ispirazione dall'applicazione che ha riscosso un buon successo nel mondo della telefonia mobile: Flappy Bird. Gioco che è stato accusato di aver (come tanti altri) plagiato, in modo particolare nello sfondo, Super Mario. Il classico per eccellenza.
Il gioco consiste nel controllare un uccello in caduta libera (con un solo tasto) facendo in modo che esso schivi gli ostacoli (tubi di Super Mario) che, in modo casuale, gli si presentano dinnanzi.
L'obiettivo è superare il maggior numero di ostacoli.
Questo progetto parte dall'idea di base di questo gioco e si sviluppa diversamente in qualche piccolo aspetto, per una questione puramente soggettiva.

DIFFERENZE DAL GIOCO DI BASE:

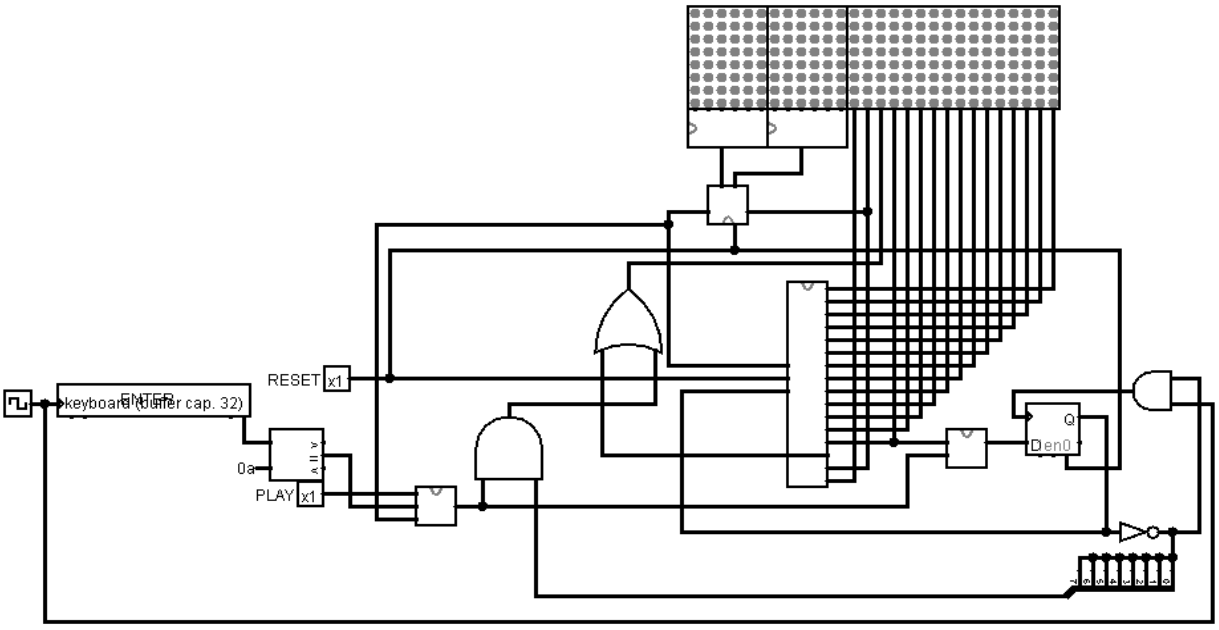
- Ostacoli di diversa posizione e dimensione.
- Possibilità di spostarsi dal punto più alto al punto più basso e viceversa con una sola mossa.
- L'obiettivo del gioco non è quello di siglare un nuovo record ma il gioco termina con il raggiungimento del decimo ostacolo.

MANUALE D'USO

Il gioco funziona in modo molto semplice:

- attivare il pulsante PLAY, che attiva la caduta libera dell'uccello.
- Posizionarsi sul pulsante TASTIERA e utilizzare CTRL+K per attivare il clock (in modo che si abbia il tempo di posizionarsi sulla tastiera)
- Utilizzare ENTER per controllare l'uccello e schivare gli ostacoli.

CIRCUITO MAIN



INPUT: clock, reset, play, tastiera.

Questo circuito è il circuito principale dove effettivamente si gioca.
Sono presenti:

- 3 display uniti, 2 utilizzati per segnare il punteggio, 1 per il gioco in sé.
- Il riferimento alla tastiera dato dal pulsante Enter.
- Un tasto per il reset.
- Un tasto Play per attivare l'uccello.
- I blocchi componenti che verranno analizzati in seguito.

La parte sinistra del circuito si occupa di controllare i movimenti dell'uccello e di visualizzare correttamente sul display le varie possibilità che possono presentarsi. In più risiede il comando di Reset che serve per reiniziare il gioco e il Blocco Flappy che serve per determinare l'andamento dell'uccello.

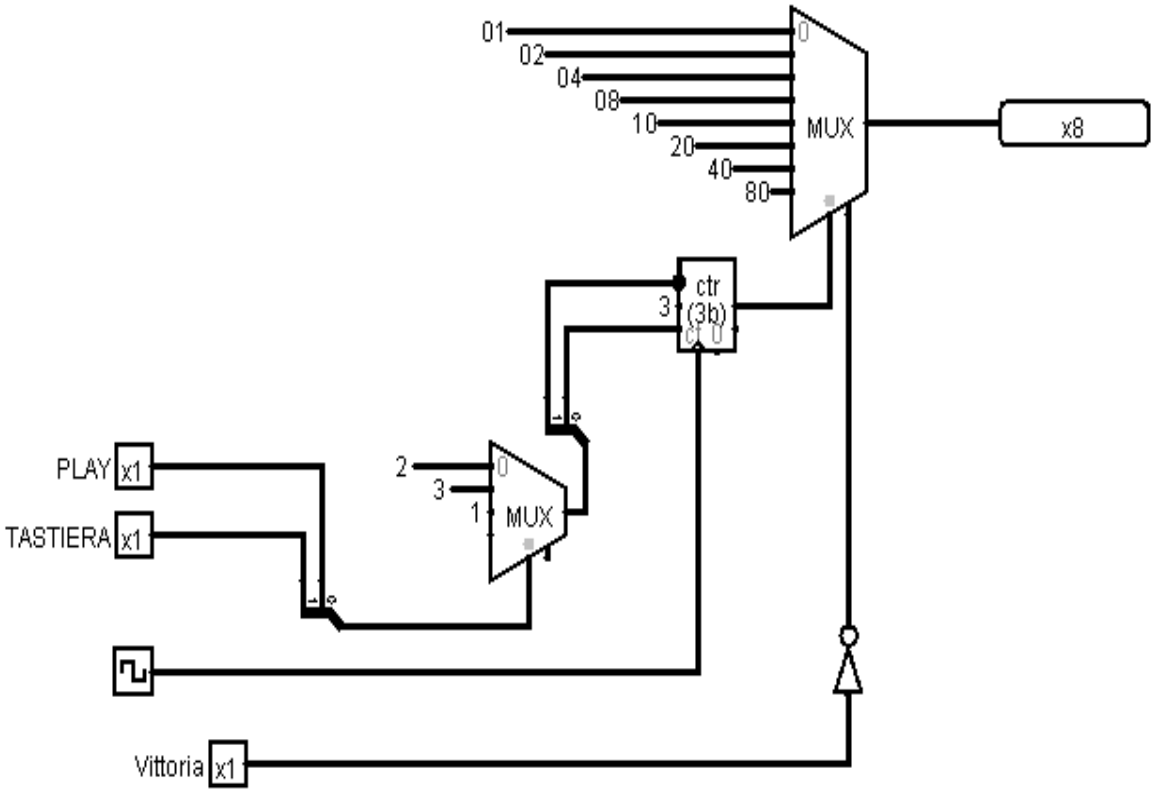
Nella *parte destra* invece è presente il Blocco Multiplexers (legato al display), nel quale viene deciso cosa visualizzare a video: se il simbolo di vittoria, se la scritta di game over oppure lo sfondo di gioco. Risiede poi il Blocco Game Over che ha il compito di determinare la sconfitta nel caso in cui l'uccello non eviti l'ostacolo. Collegata a questo Blocco c' è una sezione che si occupa di salvare lo stato game over nel tempo (finchè non verrà premuto il pulsante di Reset).

Nella *parte centrale* del circuito sono presenti due grandi porte logiche: AND e OR, che hanno il compito di far visualizzare correttamente (a seconda dei casi) la matrice colonna corrispondente alla locazione dell'uccello. Questa matrice infatti deve visualizzare o solo l'uccello o contemporaneamente l'uccello e l'ostacolo.

Infine nella *parte superiore* del circuito sono presenti i display di gioco e il Blocco Contatori che controlla il punteggio della partita e lo visualizza a video.

BLOCCHI COMPONENTI

Flappy

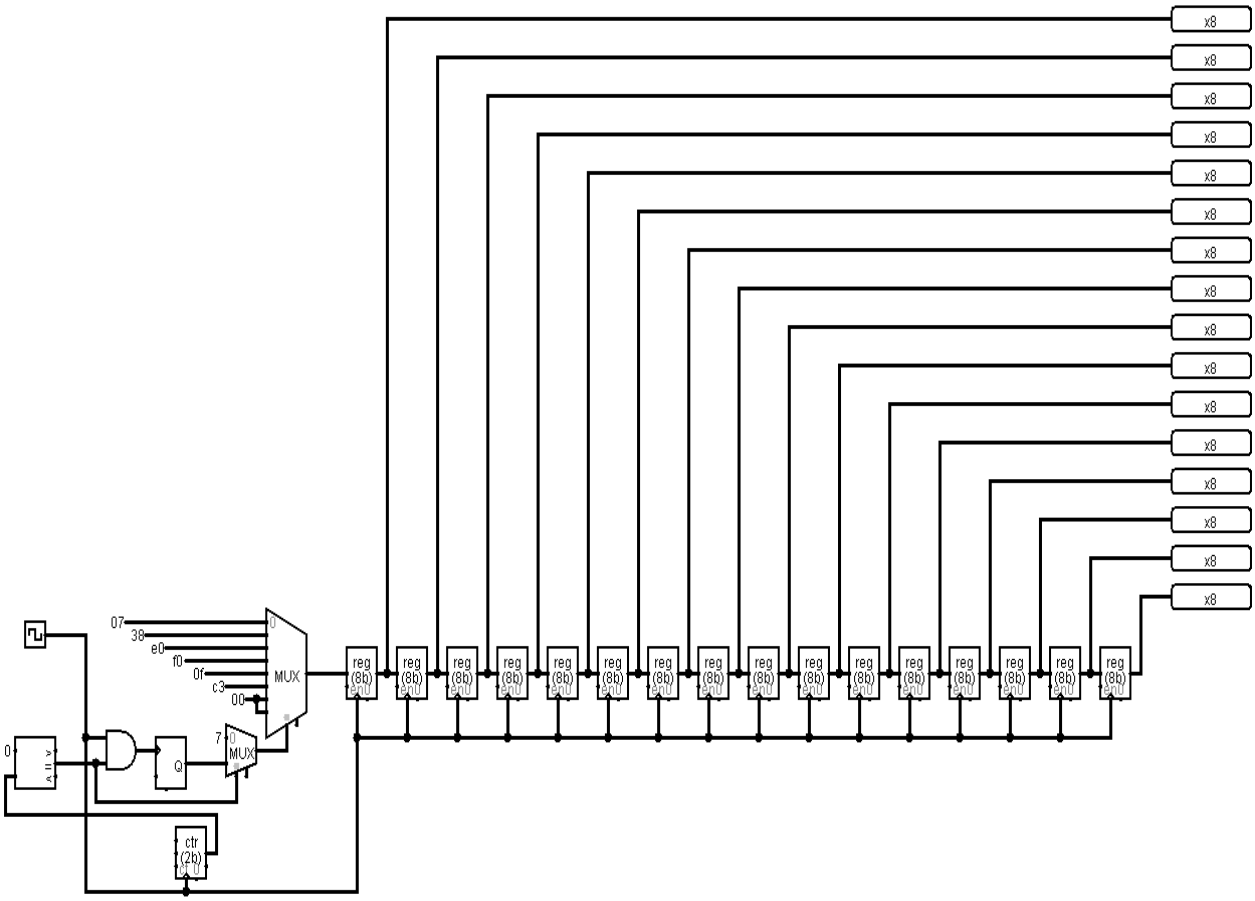


INPUT: Play, Tastiera, vittoria, clock

OUTPUT: 1 uscita a 8 bit

Questo circuito controlla il movimento dell'uccello verso il basso o verso l'alto a seconda che si prema Enter o meno.
I due Input Play e Tastiera agiscono da selettore in un multiplexer nel quale vengono scelti tre possibili valori che a loro volta controllano un contatore.
L'uscita del contatore diviene infine il segnale di selezione di un altro multiplexer che determina la salita o la discesa dell'uccello.
L'input vittoria invece, quando è attivo, blocca il movimento dell'uccello.

Random Mondo

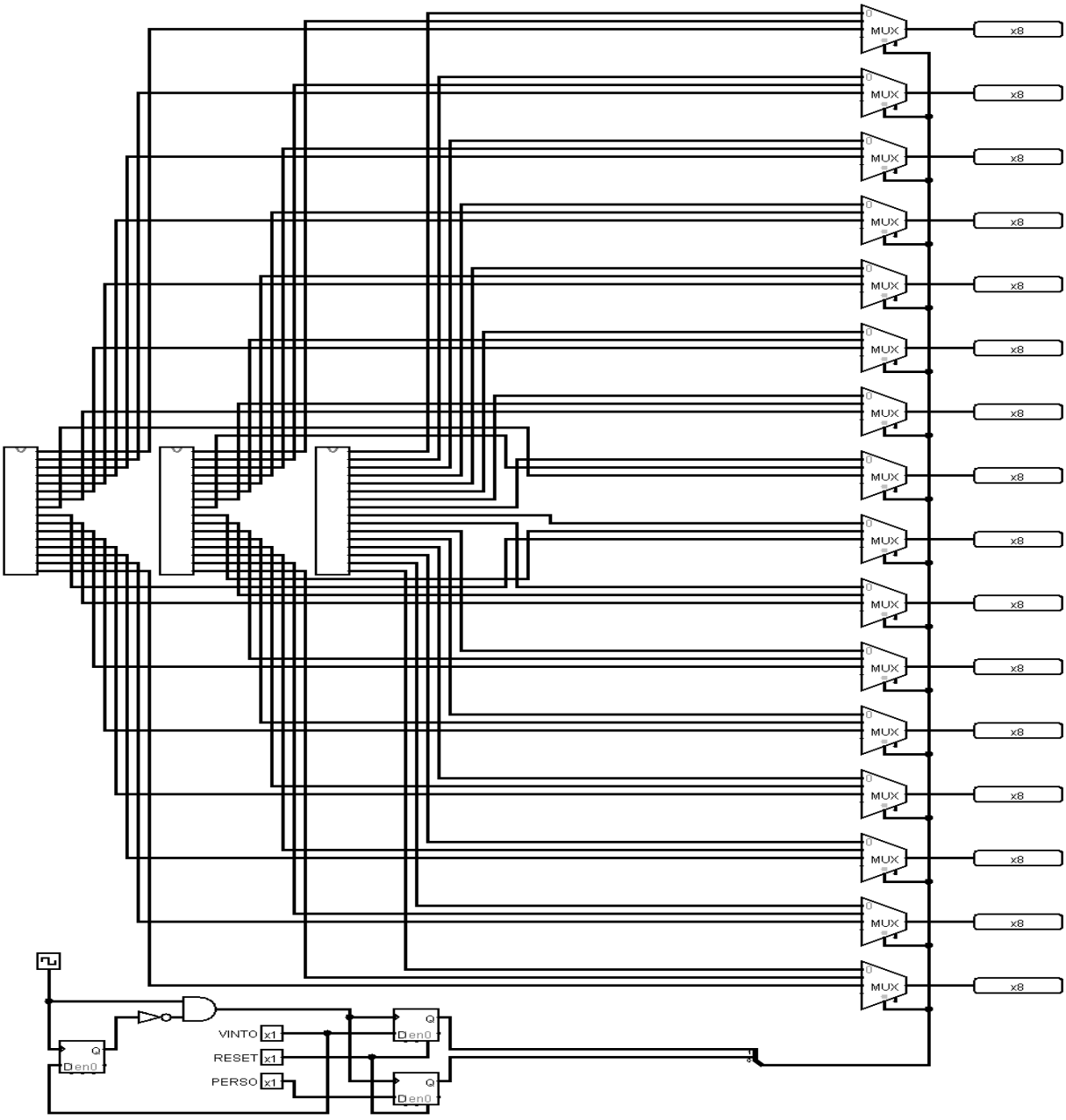


INPUT: clock

OUTPUT: 16 uscite da 8 bit

Questo circuito ha come input il solo clock.
Attraverso l'uso di diversi contatori e di un comparatore si determina il segnale che entrando nel multiplexer seleziona un ostacolo (in modo casuale) distanziandolo dal successivo con tre cicli di clock vuoti.
I bit che rappresentano l'ostacolo vengono inseriti in un banco di registri a scorrimento che ne salvano lo stato e determinano l'appunto scorrimento dell'ostacolo nel mondo.
Le 16 uscite ad 8 bit vanno pensate come le 16 matrici colonne che formano il display di gioco.

Multiplexers



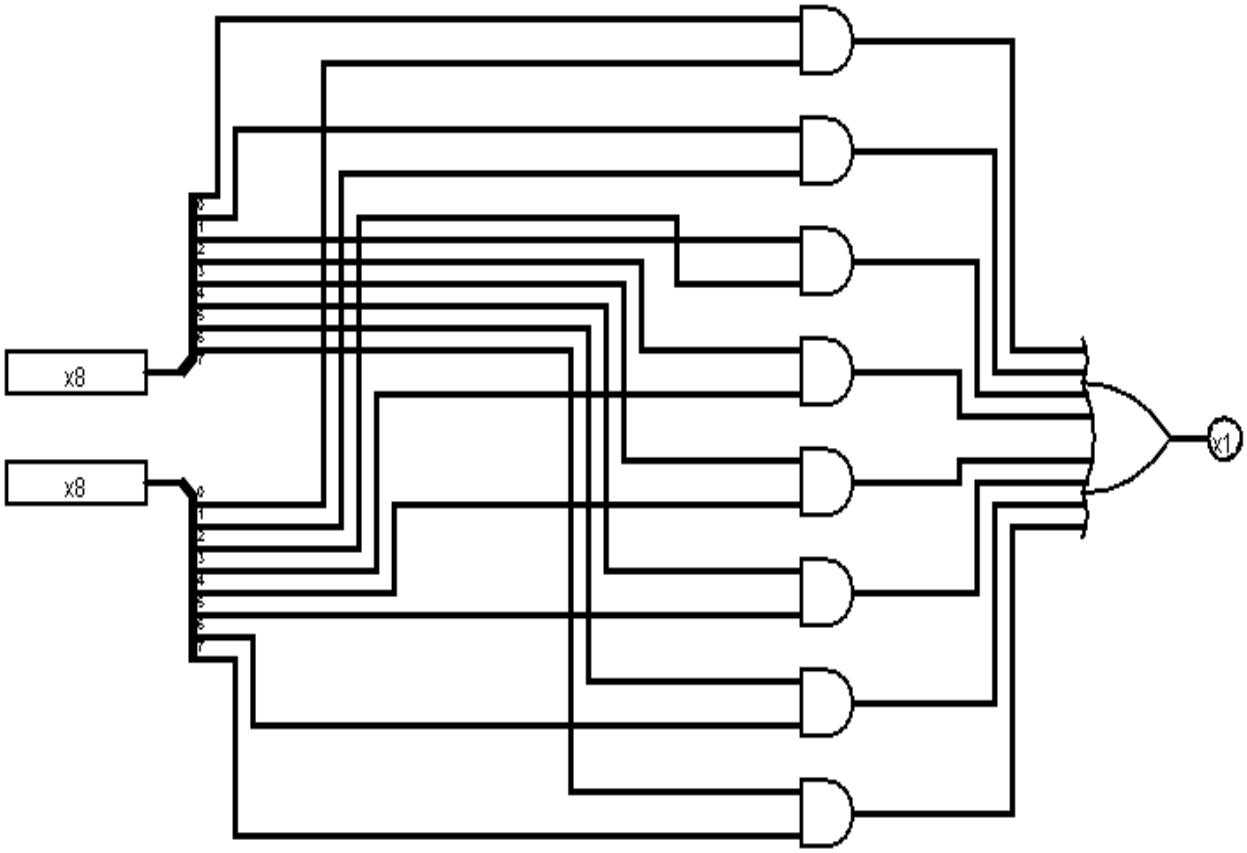
INPUT: clock, Blocco Fin, Blocco vittoria, Blocco Random Mondo, Vinto, Perso, Reset.

OUTPUT: 16 uscite da 8 bit

In questo circuito il clock salva o il segnale di Vinto o il segnale di Reset oppure quello di Perso in due flip flop di tipo D. I quali fungono da unico segnale nei 16 multiplexer che scelgono lo stato del mondo: Blocco Fine, Blocco Vittoria, Blocco Mondo.

Blocco Fine: racchiude le costanti che visualizzano sul display il simbolo di sconfitta.
Blocco Vittoria: racchiude le costanti che visualizzano sul display il simbolo di vittoria.
Blocco Mondo: è il Blocco Random Mondo già analizzato.

Game Over



INPUT: 2 entrate da 8 bit

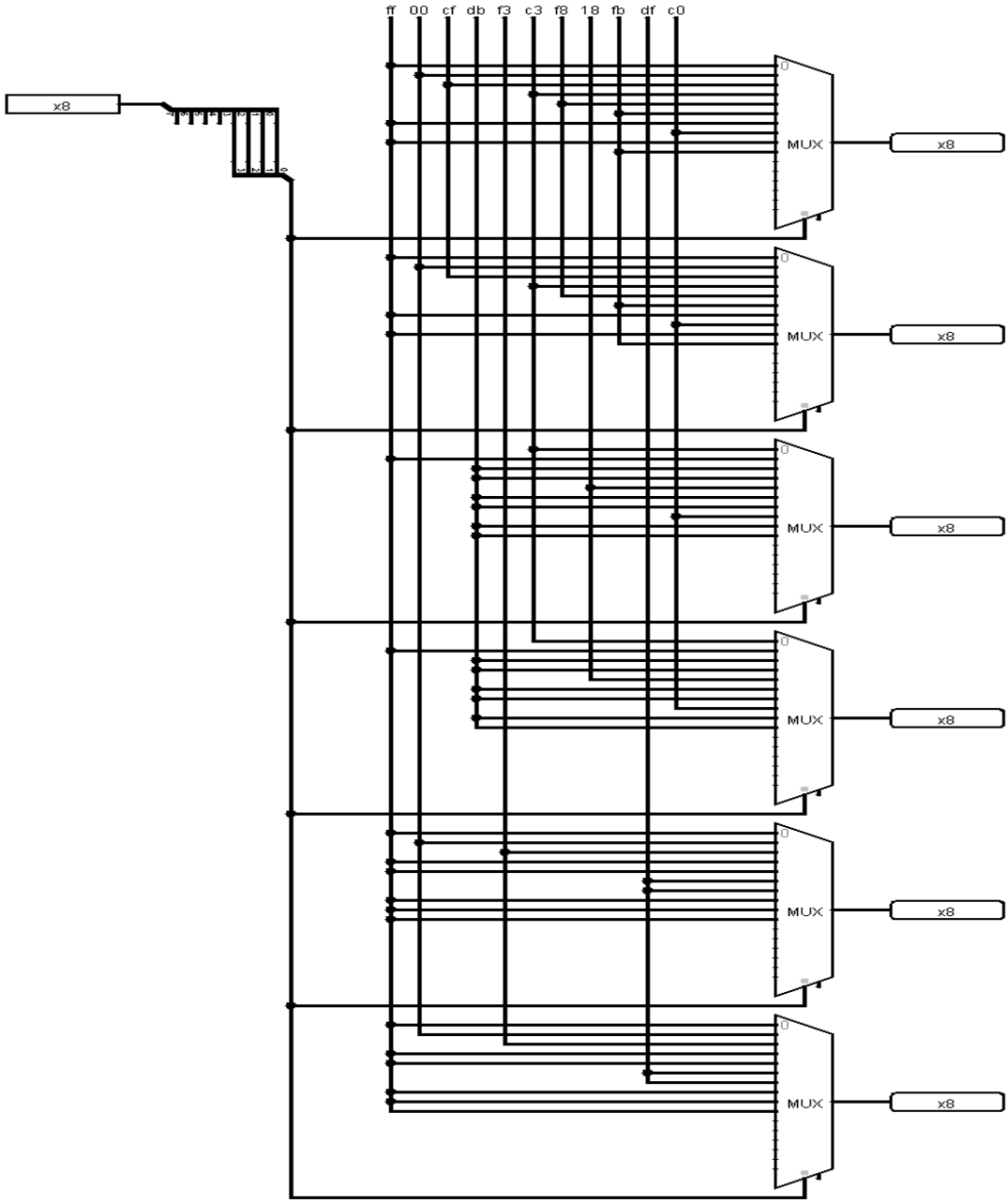
OUTPUT: 1 uscita da 1 bit

In questo circuito un'entrata rappresenta la posizione dell'uccello, l'altra rappresenta la forma dell'ostacolo che si trova nella stessa matrice colonna della posizione dell'uccello.
Attraverso porte AND si possono verificare 2 casi:
il caso in cui l'uccello tocca l'ostacolo con la conseguente attivazione dell'output, oppure il caso in cui l'uccello eviti l'ostacolo.

Passato

Stessa cosa accade nel Blocco Passato, con l'unica differenza che come ingressi si hanno la matrice antecedente a quella della posizione dell'uccello e un'entrata ff.
In questo modo l'uscita si attiverà solo quando un ostacolo risulterà alla sinistra dell'uccello, ovvero passato.

Punti

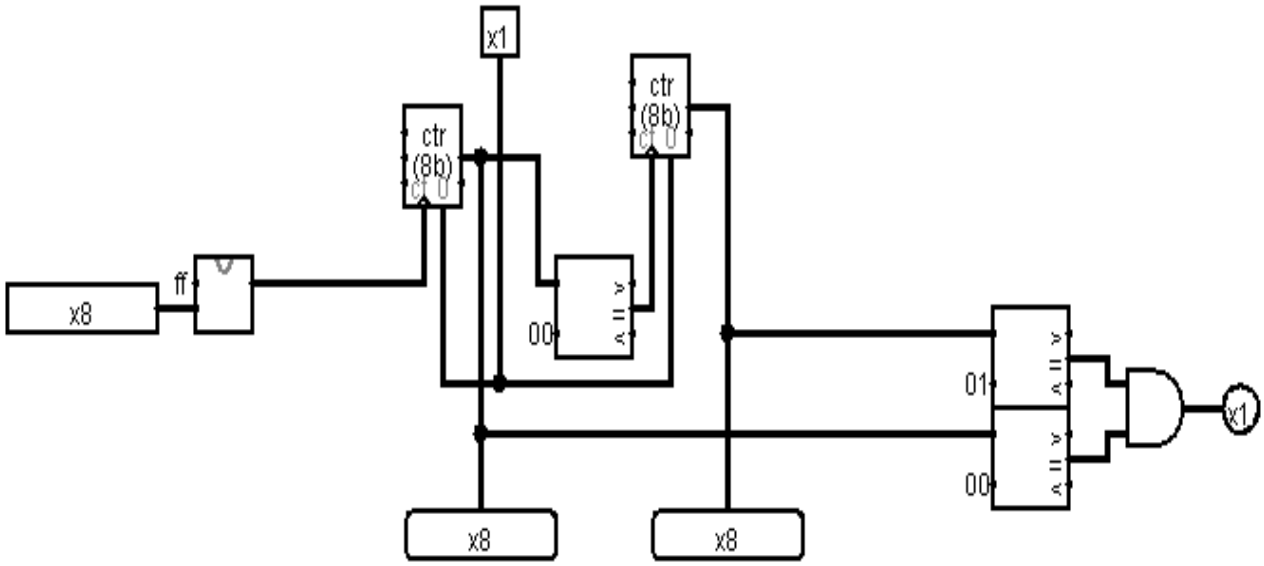


INPUT: 1 entrata da 8 bit

OUTPUT: 6 uscite da 8 bit

In questo circuito, attraverso costanti e multiplexer, sono rappresentate le cifre che appaiono nei due display del Main.

Contatori



INPUT: 1 entrata da 8 bit

OUTPUT: 2 uscite da 8 bit e 1 uscita da 1 bit

In questo circuito viene determinato il numero degli ostacoli superati.
L'uscita delle unità è determinata da un contatore che assume come valore massimo 9.
L'uscita delle decine è determinata da un contatore che si attiva solo quando il valore del contatore delle unità è uguale a 0.
Infine attraverso due comparatori si può impostare il numero di ostacoli da superare necessario per vincere.
Al raggiungimento di questo si attiva un'uscita da 1 bit, ovvero il segnale di Vittoria.

CONCLUSIONI

Ci sono diverse parti che potrebbero essere ottimizzate, ad esempio la parte relativa al punteggio la si potrebbe implementare utilizzando elementi già esistenti e non creandola da zero.
Oppure si potrebbero associare agli input Reset e Play dei pulsanti controllati da tastiera.
Sono comunque soddisfatto di questo progetto, nato dalla semplice idea di riuscire a spostare un segnale verso l'alto o verso il basso in una matrice colonna di led.