

# RELAZIONE PROGETTO “SOMMA LE DITA”

## SPECIFICHE

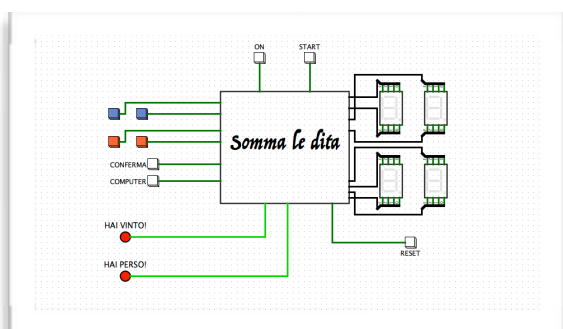
### *Regole del gioco.*

Nel “mondo reale” due giocatori partono con un dito della mano aperto per ogni mano. Si parte con due mani a testa, quindi ognuno avrà due dita aperte, una sulla mano destra e una sulla sinistra. Ogni giocatore dispone di una sola mossa per turno. Il giocatore attivo durante il turno corrente tocca una mano dell'avversario con una mano a scelta tra le proprie. A questo punto l'avversario aprirà nella mano che gli è stata toccata un numero di dita pari alla somma delle dita che aveva in precedenza più quelle che ha la mano con cui è stato toccato. Qualora la somma risultasse superiore a 5, il valore risultante sarebbe pari alla differenza tra quel numero e 5. Una mano che arriva al valore 5 (esatto) viene eliminata. Perde chi non ha più mani con cui giocare. Nel mio progetto, il primo giocatore è rappresentato dall'utente del circuito, il comportamento del secondo giocatore è invece simulato dal circuito. Le mani sono rappresentate simbolicamente dal valore che rappresentano. Se una delle due raggiunge valore 0 non sarà più in grado di ricevere input fino al refresh del gioco. La scelta è veicolata attraverso pulsanti di selezione.

### *Interfaccia, funzionamento e utilizzo.*

Il valore di ciascuna mano è trasmesso da un display: di colore blu per le mani del giocatore e di colore rosso per le mani del computer. Quattro pulsanti, del medesimo colore dei rispettivi display, permettono all'utente di scegliere una tra le sue mani e una tra le mani del computer.

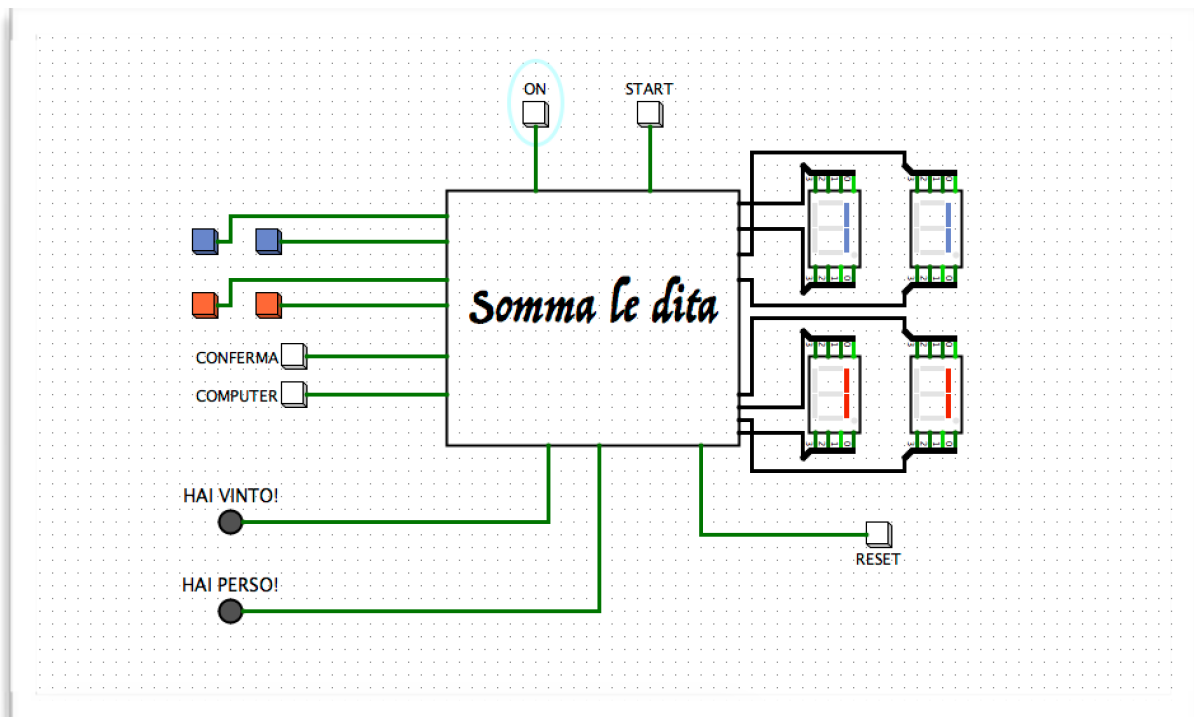
L'utente ha inoltre a disposizione un tasto CONFERMA che rende effettiva la propria scelta, e un tasto COMPUTER attraverso cui passare la parola al computer e visualizzare la sua mossa. Si verrà informati della vittoria o della sconfitta attraverso due LED: uno associato alla frase “HAI VINTO!”, l'altro associato alla frase “HAI PERSO!”.



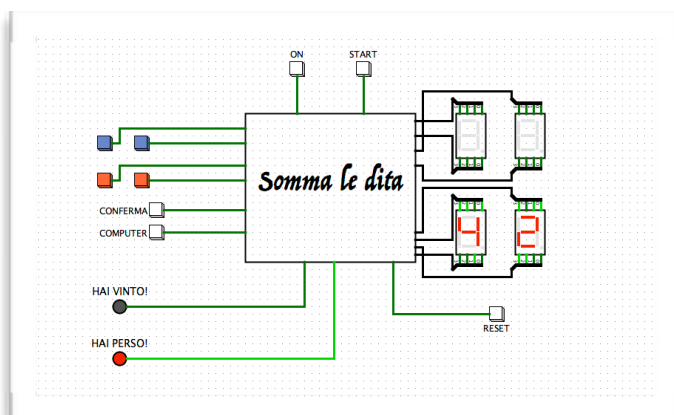
Sono presenti i tasti ON, START e RESET attraverso cui è possibile avviare le sessioni di gioco.

Una volta avviata la simulazione, i display risultano spenti, di conseguenza entrambi i LED saranno accesi.

È necessario premere il tasto ON per far sì che i display si accendano. Una volta accessi conterranno automaticamente il valore 1, corrispondente allo stato iniziale di gioco.



Il tasto START indica il vero e proprio inizio della sessione di gioco: una volta avviata questa, è possibile procedere con la propria selezione e con la conseguente conferma. Durante il turno dell'utente, questo ha 4 alternative: o incrementa il contatore della prima mano dell'avversario (in due modi: o aggiungendo il valore della sua prima mano o aggiungendo il valore della sua seconda mano) oppure incrementa il contatore della seconda mano dell'avversario (in due modi: come sopra). Se il valore S della somma supera 4 (massimo valore del contatore), allora il valore risultante del contatore sarà  $S-5$  ( $<5$ ).



Quando uno dei due giocatori (utente o computer) resta con entrambi i contatori a 0 (display spenti), il gioco termina segnalando come vincitore l'avversario.

Dato che il gioco prevede una mossa alternata tra utente e computer, alla conferma del giocatore dovrà seguire la pressione del tasto COMPUTER.

Quando uno dei due giocatori avrà vinto, quindi quando entrambi i display dell'avversario risulteranno spenti, s'illuminerà uno dei due LED.

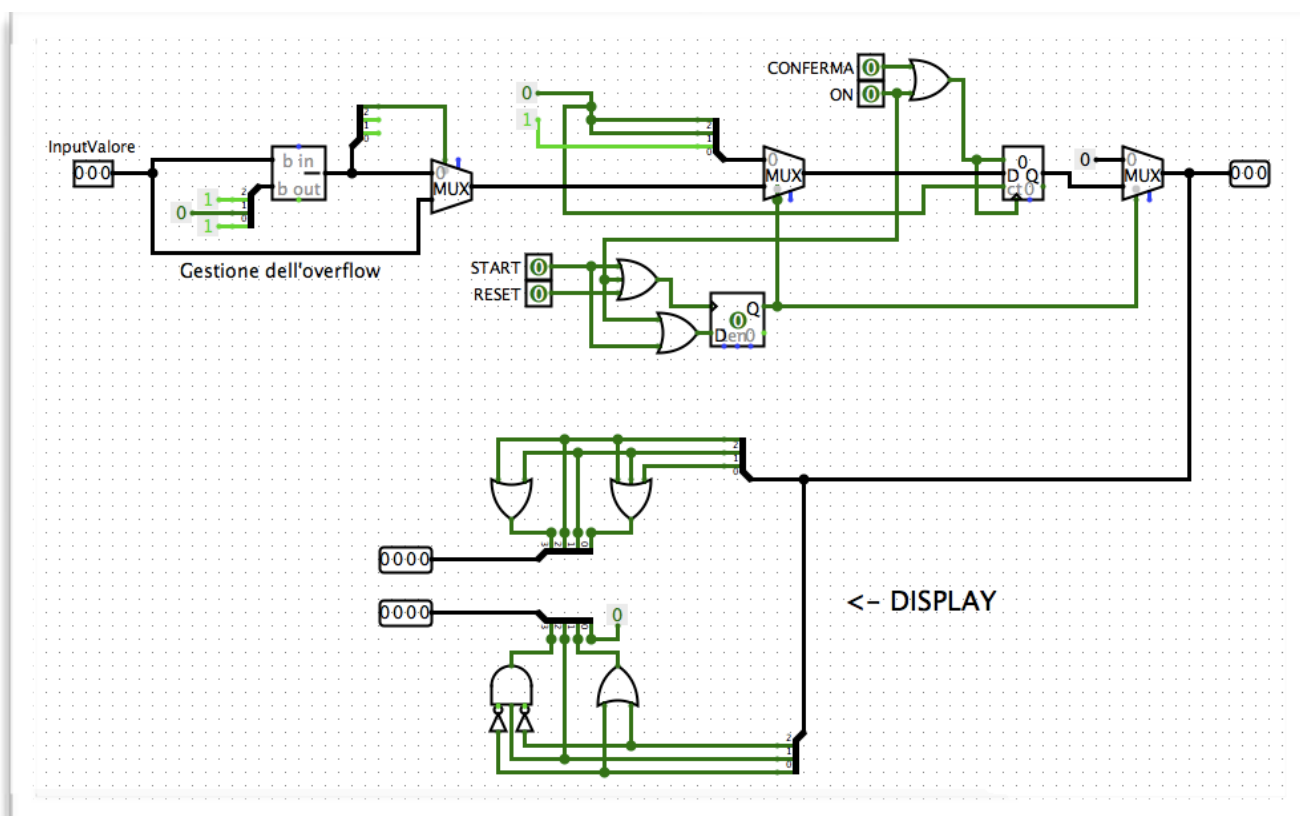
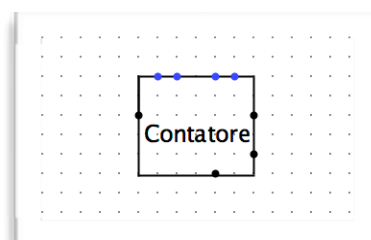
In qualsiasi momento del gioco sarà possibile ricominciare dallo stato iniziale con la pressione del tasto RESET. A questo punto, si potrà ricominciare, a partire dal tasto ON seguito dal tasto START con una nuova sessione di gioco.

## SOTTOCIRCUITI

### Contatori.

Per memorizzare il valore di ciascuna mano ho utilizzato dei contatori fatti *ad hoc* che sfruttano il componente *counter* già disponibile in logisim.

Ciascun contatore può contenere un valore numerico compreso tra 0 e 4 (nel caso in cui il valore della somma tra le due mani risultasse 5 al contatore dovrà arrivare il valore 0, corrispondente al caso "mano eliminata").



Il contatore è influenzato da 5 input:

- il valore in ingresso della somma delle due mani
- il segnale di accensione (ON)
- il segnale di inizio gioco (START)
- il segnale di CONFERMA
- il segnale di RESET

Gli ultimi 4 input intervengono all'interno del circuito modificando l'output risultante (rappresentato dal valore stesso e dalla sua rappresentazione per il display) sulla base delle varie fasi di gioco.

I 3 bit del valore in ingresso entrano in un sistema di verifica che, se il numero in ingresso è minore di 5, manda in uscita il numero stesso, viceversa il valore in uscita del multiplexer sarà dato dal risultato della sottrazione: il bit di selezione del valore del multiplexer su cui si basa il sistema di verifica è infatti il bit di segno del risultato della sottrazione tra il numero in ingresso e la costante 5.

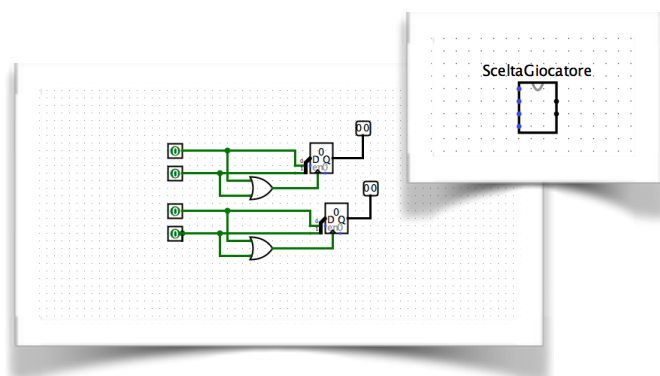
Il segnale adesso entra all'interno di un altro multiplexer: nel primo stadio della simulazione circuitale, ovvero quando l'utente preme il pulsante ON, questo multiplexer seleziona come valore da mandare in output la costante 1. Invece, con la pressione del tasto corrispondente al segnale START, il multiplexer selezionerà il valore in uscita dal primo multiplexer sopra descritto.

Affinché il valore esca dal sottocircuito come output è necessaria la pressione del tasto connesso all'input CONFERMA (il cui relativo tasto è presente nella versione *black-box* del circuito principale), questo azionerà un registro che manderà in uscita il valore presente in entrata.

Nel caso in cui si azioni il tasto RESET un multiplexer manderà in uscita la costante 0, allo scopo di azzerare il valore in uscita del contatore.

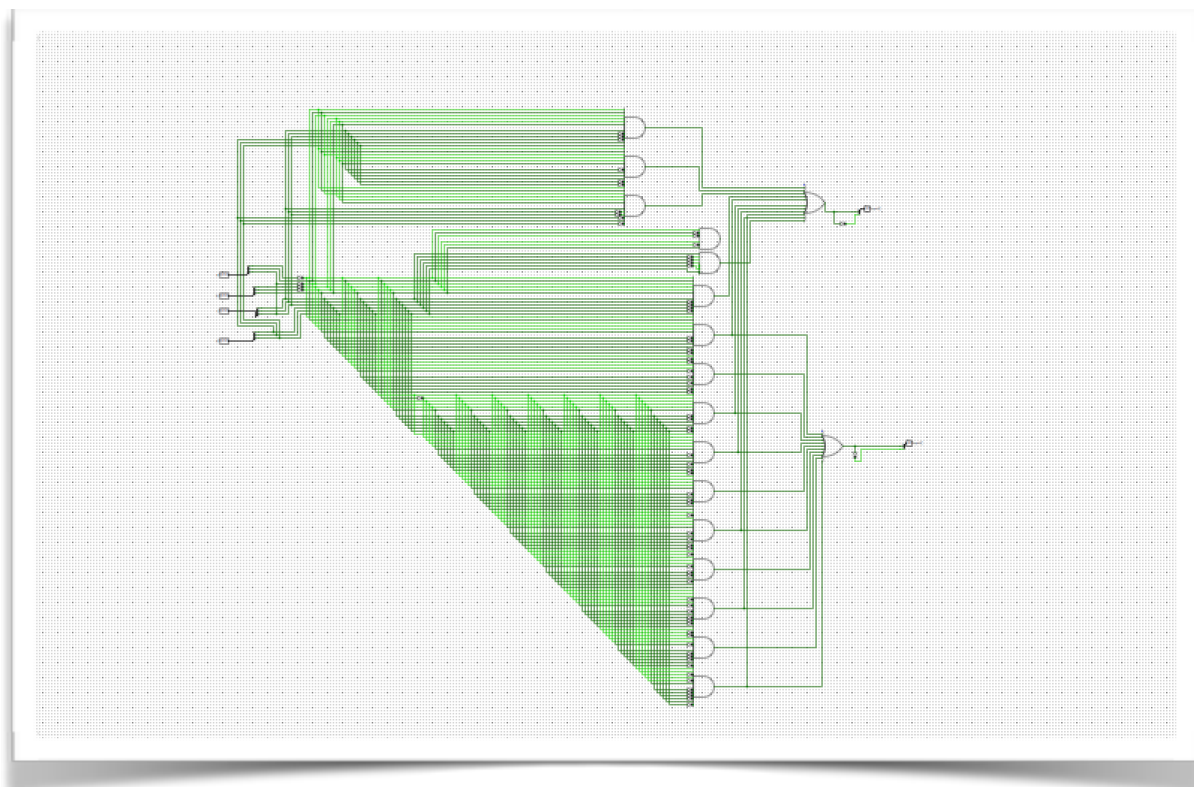
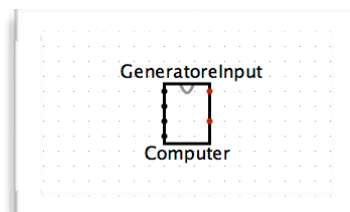
### ***SceltaGiocatore.***

Questo semplicissimo sottocircuito ha il compito di trasformare la scelta compiuta dall'utente tramite i pulsanti colorati in due coppie di bit di scelta delle due mani: 01 nel caso della mano 1, 10 nel caso della mano 2. I due output saranno indicati nel seguito come 1sceltaGIOCATORE e 2sceltaGIOCATORE.



## ***GeneratoreInputComputer.***

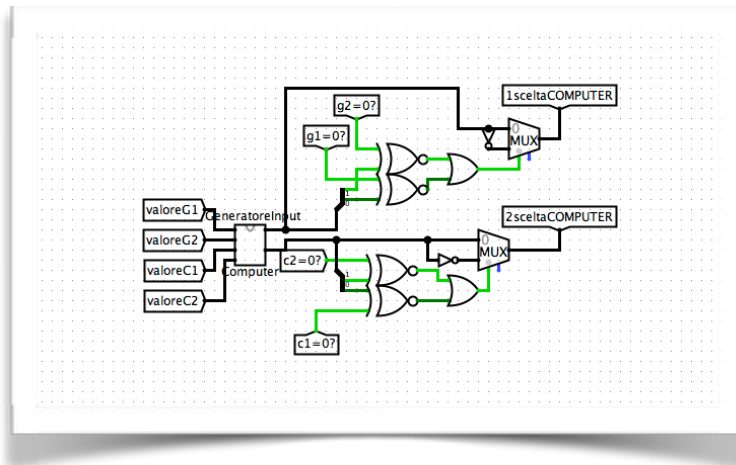
Questo sottocircuito ha l'importante compito di gestire il comportamento del computer sulla base dello stato della partita. Riceve in ingresso i valori contenuti nei 4 contatori (3x4 bit) e restituisce in uscita due coppie di bit che codificano la "scelta" del computer: i primi due codificano su quale delle due mani del giocatore il computer vuole agire (01 per la prima mano, 10 per la seconda) e i restanti codificano la scelta tra le mani del computer. Questa scelta non è, infatti, casuale, ma influenzata dal valore *in quel momento di gioco* delle mani in campo.



Il circuito è stato realizzato in maniera tale da far evitare al computer errori grossolani, tuttavia, in alcune situazioni, non conferisce alla strategia di gioco del computer particolare finezza.

Il circuito è completato da un sistema di controllo (immagine alla pagina seguente) posizionato, per comodità, all'interno della sezione principale del progetto che, qualora una delle mani scelte dal computer sia in quel momento a "0", inverte i bit di

scelta per far sì che l'output finale corrisponda all'altra mano (infatti l'output di scelta può essere di due tipi: 10 o 01). Mettere questo controllo a valle risultava progettualmente più conveniente. Agli output del circuito ho associato le etichette 1sceltaCOMPUTER e 2sceltaCOMPUTER, rispettivamente per la scelta della mano del giocatore e per quella del computer.

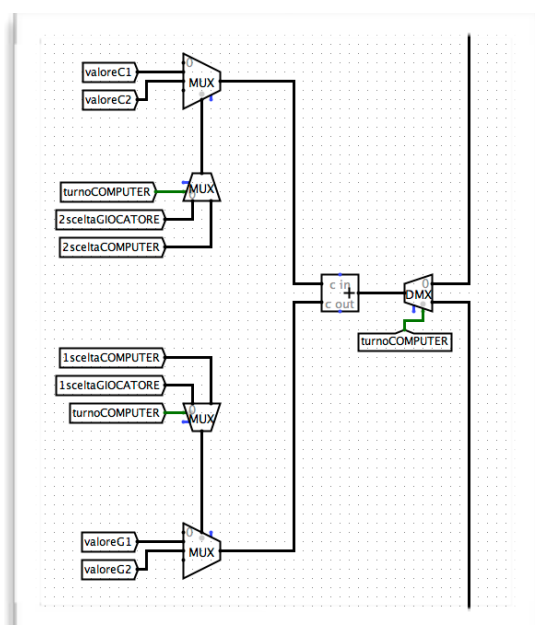


## CIRCUITO PRINCIPALE E INTERAZIONE FRA LE PARTI

Il circuito principale si occupa di coordinare i segnali provenienti dai sottocircuiti ed è incaricato di calcolare la somma tra i valori selezionati.

Per questo motivo, la parte centrale del circuito è costituita da un adder che riceve come addendi i segnali corrispondenti al valore di una delle due mani del giocatore e di una delle due mani del computer. Quindi, i 4 valori che gestisce in input altro non sono che i valori in uscita dei contatori (a cui ho associato le etichette valoreC1, valore C2, valoreG1, valoreG2).

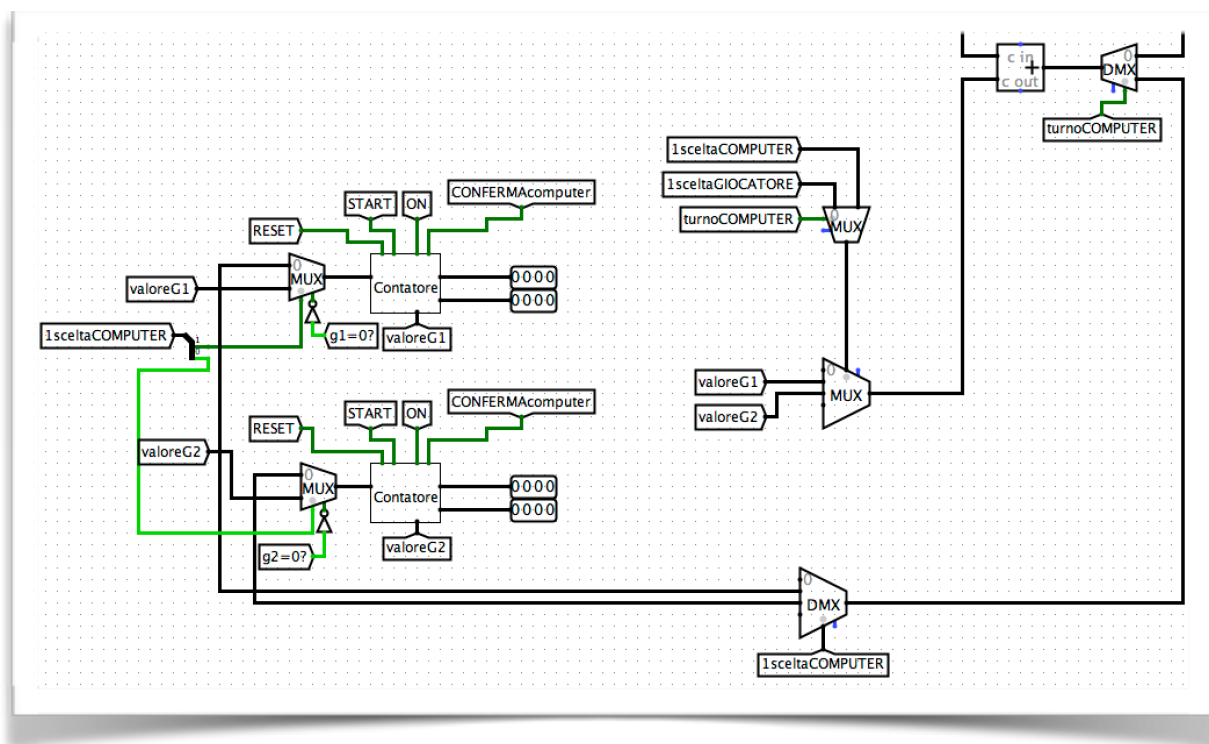
La scelta degli addendi da mandare in entrata all'adder è ovviamente influenzata dalla scelta compiuta dal giocatore o dal computer, ovvero gli output generati dai sottocircuiti GeneratoreInputComputer e SceltaGiocatore (indicati con le etichette: 1sceltaCOMPUTER, 2sceltaCOMPUTER, 1sceltaGIOCATORE, 2sceltaGIOCATORE)



dovranno costituire i bit selezionandi del multiplexer più prossimo all'adder. Ricordiamo però che i nostri sottocircuiti, GeneratoreInputComputer e SceltaGiocatore, hanno prodotto ben 4 diverse scelte, 2 a carico del computer e 2 a carico del giocatore. Dunque, si è rivelato necessario introdurre nel circuito un ulteriore multiplexer che selezioni a sua volta la coppia di bit selezionandi (rispetto al primo multiplexer) selezionati a loro volta sulla base del turno in corso, quindi in funzione del segnale chiamato "turnoCOMPUTER" che, gestito dal pulsante "COMPUTER", va a 1 solo durante il turno del computer e torna a 0 durante il turno del giocatore. Questo meccanismo, ovviamente, è replicato per entrambe le entrate dell'addizionatore.

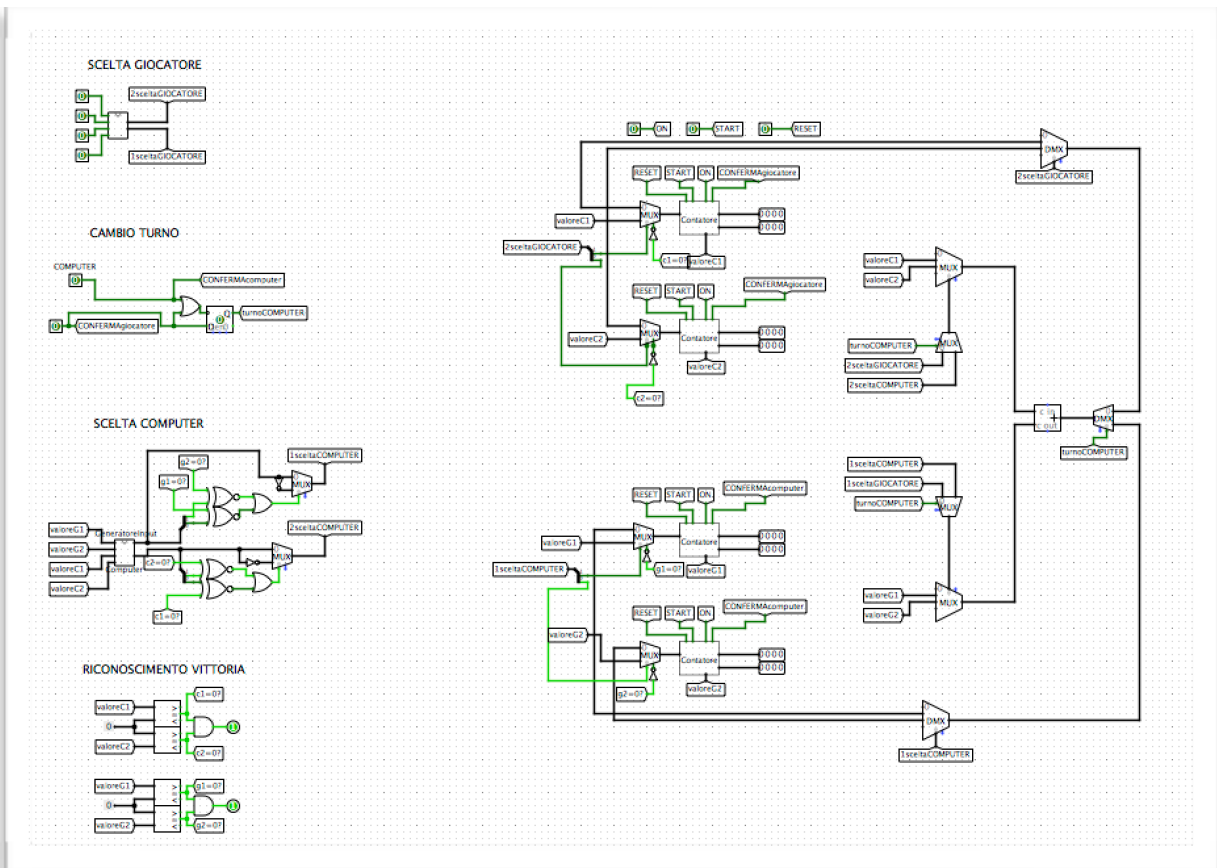
A questo punto il segnale di somma, appena uscito dall'addizionatore, troverà davanti a sé 4 possibilità che incontrerà sotto forma di due bivi, uno dopo l'altro, che guideranno il risultato verso il giusto contatore.

Prima di tutto dovrà scegliere se dirigersi verso le mani del giocatore o verso le mani del computer. Questa scelta può essere fatta esclusivamente sulla base del turno in corso. Basterà un demultiplexer in grado di dirottare il risultato in una delle due coppie di contatori a seconda del valore turnoCOMPUTER. Durante il turno del giocatore la somma dovrà sostituirsi a uno dei valori delle mani del computer, viceversa, durante il turno del computer, la somma dovrà alterare una delle mani del giocatore.



Una volta presa una delle due strade il segnale sarà nuovamente dirottato attraverso un altro demultiplexer in una delle due mani, sta volta sulla base della scelta compiuta o dal computer o dal giocatore (questo in base alla strada imboccata dopo il primo demultiplexer, quindi in base al turno).

Questo ci garantisce che, nel rispetto della logica di gioco, il valore dei contatori relativi al computer cambi durante il turno del giocatore e il valore dei contatori relativi al giocatore cambi durante il turno del computer.



Solo ora il segnale potrà entrare all'interno del contatore e sostituirsi al valore attualmente presente al suo interno.

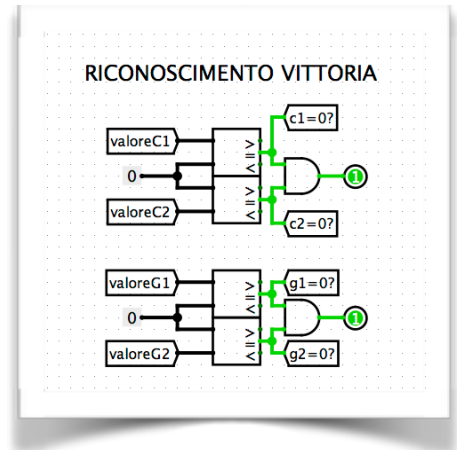
L'altra mano però non dovrà in nessun modo modificare il suo attuale valore. Affinché questo non accada l'ingresso di ogni contatore è gestito da un multiplexer incaricato di mantenere il valore corrente nel caso in cui il segnale di selezione indichi che non sia quella la mano prescelta. Questo multiplexer presiede a un'altra importante funzione: evita che quando il relativo contatore sia a 0 (mano eliminata) il valore possa in qualche modo subire modifiche, inibendo il multiplexer tramite l'apposito ingresso. Una mano eliminata non ha più nessun ruolo nel gioco.



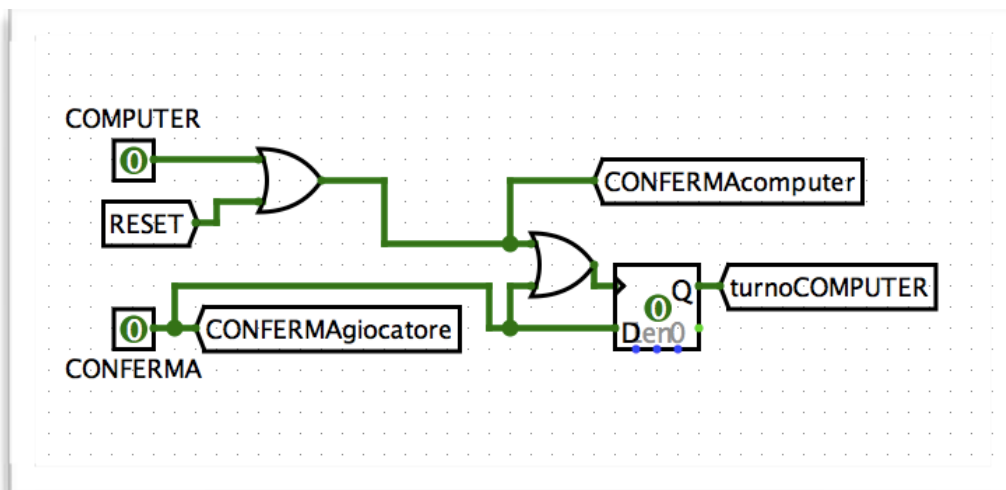
Ricordiamoci, inoltre, che il valore d'uscita del contatore è anche influenzato dai segnali ON, START e RESET precedentemente descritti e che oltre all'output contenente il valore, il contatore produce altri due output da 4 bit ciascuno avente a carico la gestione dei display.

Per comodità ho collocato all'interno del circuito principale un sottocircuito che si occupa della gestione dei LED che notificano all'utente la vittoria o la sconfitta.

Il suo funzionamento è molto semplice: il LED di vittoria è il risultato di un AND tra la comparazione di entrambe le mani del computer con la costante 0, il LED che notifica la sconfitta funziona allo stesso modo ma con riferimento alle mani del giocatore.



Un ulteriore sottocircuito presente nel circuito principale si occupa della gestione del segnale turnoCOMPUTER: quando il pulsante collegato all'input CONFERMA viene premuto il segnale sarà mandato a 1 (in quanto segnerà la fine del turno del giocatore); viceversa, questo andrà a 0 alla pressione del tasto COMPUTER. Il segnale turnoCOMPUTER sarà riportato a 0 anche qualora l'utente prema il pulsante relativo al RESET, così da riportare il circuito alla situazione iniziale.



## CONSIDERAZIONI FINALI

Il circuito presenta sicuramente margini di miglioramento. A mio avviso, la principale carenza riguarda la strategia di gioco del secondo giocatore. Il circuito `GeneratoreInputComputer` è stato progettato in maniera tale da non simulare un comportamento volto a una strategia a lungo termine. Il computer mira alla sua vittoria immediata, quando possibile; tuttavia, se non si presenta questa possibilità, può compiere scelte convenienti con la stessa probabilità con la quale compie scelte sconvenienti (dal suo punto di vista). Il risultato è un secondo giocatore non del tutto incapace di giocare, ma comunque inesperto.

In che modo rimediare a questa carenza? Dunque, il sottocircuito da me sviluppato si presenta sotto forma di una SOP basata su una tabella di verità che associa ai vari possibili “stati” di gioco (tutte le combinazioni possibili dei 3x4 bit rappresentanti il valore delle mani) quattro possibili scelte (1-1, 1-2, 2-1, 2-2). In fase di progettazione ho elaborato una tabella che esauriva tutti i casi possibili fornendo sempre l’output (la scelta) più efficace per avere la meglio sull’utente. Una volta preso atto dell’impossibilità di semplificazioni evidenti, a causa del dispendio di risorse che avrebbe significato la trascrizione circuitale, ho deciso di ridurre i casi possibili ai casi “evidenti”, quelli dove il computer proprio non poteva fallire per non sembrare totalmente estraneo alle regole del gioco, ottenendo una tabella lunga un terzo di quella di partenza. In altre parole, l’implementazione di una strategia lungimirante è del tutto possibile e non presenterebbe ostacoli in termini “logici”.

Aggiungo che, in fase di progettazione, sono partita da un grafo delle transizioni per avere un’idea schematica degli stati possibili (nel senso dei valori delle 4 mani) e quindi degli stati prossimi in funzione degli input. È proprio da qui che è scaturita la tabella a cui ho fatto riferimento in precedenza che, sebbene non nel modo atteso, è stata determinante nell’implementazione del circuito `GeneratoreInputComputer`.

L’idea originaria, infatti, prevedeva una vera e propria macchina a stati finiti per la gestione dell’intero gioco, quindi attiva sia nei turni del giocatore che in quella del computer, il cui stato prossimo fosse espresso in funzione dello stato attuale (valori dei contatori) e di 5 input: le quattro possibili coppie di scelta del giocatore e il segnale relativo al turno del computer). A causa della difficoltà nell’implementazione ho optato per la soluzione sopra descritta, quindi, sfruttando un unico circuito principale sia per i turni del giocatore che per il computer, influenzato dalle scelte di entrambi, si è reso necessario un sottocircuito che simulasse una scelta da parte del computer, così da poter sfruttare gli stessi sistemi di selezione (multiplexer) utilizzati per gestire il circuito principale.

Accantonando questo aspetto, la realizzazione del progetto è passata per un buon numero di ostacoli progettuali non preventivati, collezionando diversi tentativi fallimentari. Nel seguito riporto alcune delle difficoltà incontrate.

In un primo momento ero sicura di poter usare l'elemento *counter* di logisim, solo dopo ho compreso che, nonostante il settaggio della modalità *wrap around*, non era possibile ottenere l'effetto desiderato in quanto il mio scopo non era quello di incrementare un dato valore, ma di sostituirlo. Per questo ho dovuto creare un sistema che gestisse i numeri maggiori-uguali di 5 basandomi su un sottrattore.

Altre difficoltà sono sorte per la gestione, nel circuito principale, dei valori in entrata nei contatori. Accadeva spesso che, durante il turno del computer, mani fuori gioco resuscitassero improvvisamente. Ho risolto questo con il segnale "inibitorio" per i multiplexer più prossimi ai contatori, che blocca il segnale tutte le volte che quella mano risulta attualmente a 0.

La genesi di questi ultimi multiplexer è frutto della constatazione empirica che, una volta dirottato il segnale somma all'interno del contatore scelto, attraverso il secondo demultiplexer, l'altro assumeva valore 0 piuttosto che mantenere il proprio valore attuale.

Inoltre, la prima versione del circuito, presentava un'interfaccia di input abbastanza scomoda. Il turno del computer doveva essere selezionato mandando un bit a 1 e il turno del giocatore rimandando lo stesso a 0. Ho risolto grazie al sottocircuito presente all'interno del circuito principale descritto sopra.

Quando il circuito era praticamente ultimato ho pensato che fosse opportuno inserire un segnale di RESET che permettesse di avviare una nuova partita in qualsiasi momento senza dover intervenire dall'esterno attraverso i comandi messi a disposizione dall'ambiente logisim. Questo segnale era assente nella prima versione.

Altre migliorie apportabili, ma che personalmente ritengo marginali, possono riguardare l'aggiunta di maggiori restrizioni per l'utente. Attualmente l'utente può tranquillamente fare più di una mossa senza lasciare il turno al computer, così come può permettere al computer di fare più mosse. Questa possibilità potrebbe essere negata dal circuito se questo riuscisse a riconoscere le azioni non permesse e non alterasse il suo stato se non per azioni ammissibili.