

A.A. 2015-2016

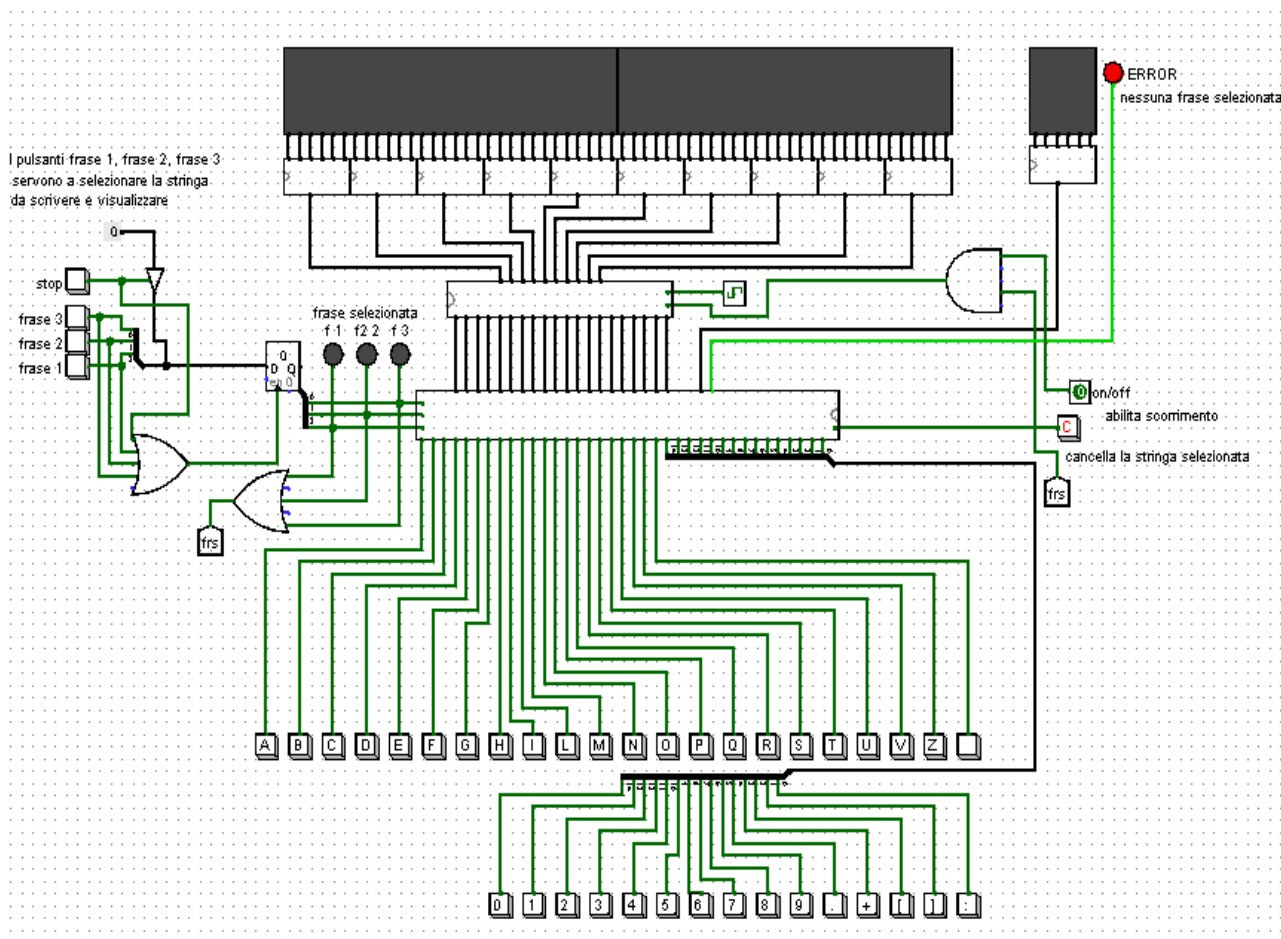
## Laboratorio architettura degli elaboratori I

### Progetto Logisim

#### Nome circuito: TESTO SCORREVOLE

Il progetto permette la memorizzazione di 3 diverse stringhe (max 20 caratteri) e la relativa visualizzazione su un display scorrevole da 10 caratteri.

Il circuito si presenta come un display composto da 2 matrici di led 30x7, una tastiera per l'input dei caratteri, 3 pulsanti per la selezione della stringa da scrivere/visualizzare, un interruttore per l'attivazione o disattivazione dello scorrimento mentre il tasto **C** serve a cancellare la stringa selezionata.



## STATO INIZIALE

Allo stato iniziale il circuito si presenta con le memorie relative alle tre possibili "frasi" vuote, in questo stadio il clock per lo scorrimento dei caratteri risulta spento. Il circuito è completamente a riposo in attesa che venga selezionata una stringa e la conseguente immissione dei caratteri tramite i tasti.

Appena avviata la simulazione gli schermi scorrevoli saranno spenti al contrario del LED di errore che segnala che non è stata selezionata alcuna frase da scrivere. Il display separato dallo schermo mostra l'ultimo carattere inserito. L'interruttore *on/off* non ha alcun effetto se non è stata scelta una stringa.

## STADIO FINALE

Il circuito non ha uno stadio finale definito, lo scorrimento della frase selezionata, se attivo, proseguirà all'infinito.

Il circuito principale contiene diversi sottocircuiti:

- 1) memoria
- 2) shiftlettera
- 3) bottoni
- 4) caratteri
- 5) comparatore7bit

### 1) MEMORIA

Il circuito Memoria (vedi fig. 1) permette la memorizzazione di una stringa di 20 caratteri caricabili in modo sequenziale. Ogni carattere è memorizzato in un Registro a 7bit.

|        |               |      |   |
|--------|---------------|------|---|
| Input: | <i>On/Off</i> | 1bit | se 1 permette inserimento nuovi caratteri                 |
|        | <i>Button</i> | 1bit | quando passa da 0 ad 1 attiva il registro successivo      |
|        | <i>Letter</i> | 7bit | codifica del carattere da memorizzare nel registro attivo |
|        | <i>Canc</i>   | 1bit | azzerà il contenuto della memoria (20 registri)           |

Output: 20 porte 7bit numerati da 1 a 20 contenenti i 20 caratteri in memoria

Il circuito memoria riceve in *Letter* il carattere da memorizzare nel registro attivo. Ad ogni passaggio dello stato di *Button* da 0 a 1, il carattere presente in *Letter* viene immagazzinato nel registro attivo, e contestualmente il counter si sposta sul registro successivo.

Il counter è realizzato mediante uno Shift register unitamente ad una porta XNOR che restituisce 1 solo quando tutti gli output sono a 0.

Superata la lunghezza massima di 20 caratteri i successivi sostituiranno i primi nei registri.

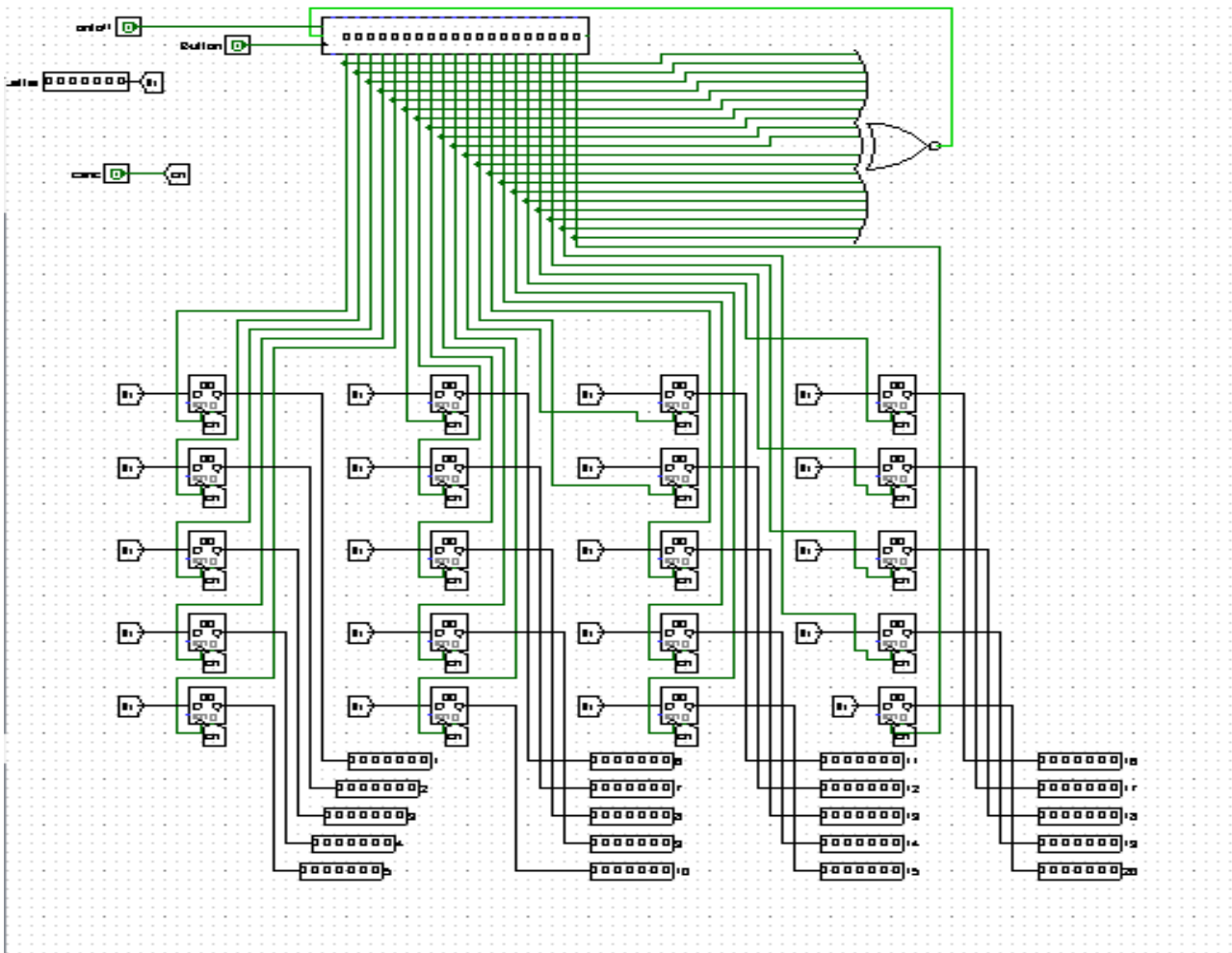


Figura 1 - Sottocircuito memoria

## 2) SHIFTLITTERA

Il circuito *Shiftlettera* (vedi fig. 2) consente di far scorrere i caratteri salvati in Memoria, 10 alla volta.

Input:        *On/Off*        1bit    se 1 permette lo scorrimento dei caratteri  
               *Clock*        quando passa da 0 ad 1 attiva il buffer successivo  
               20 ingressi da 7bit contenenti le codifiche dei caratteri

Output:        10 porte da 7 bit aggiornate ad ogni colpo di clock

Il circuito riceve in input 20 codifiche dei caratteri. Ad ogni colpo di *Clock* viene attivato il buffer corrispondente.

I counter sono realizzati con due shift register: uno da 1bit e 20 stadi collegato ai buffer e uno da 7bit e 10 stadi collegato alle porte di output.

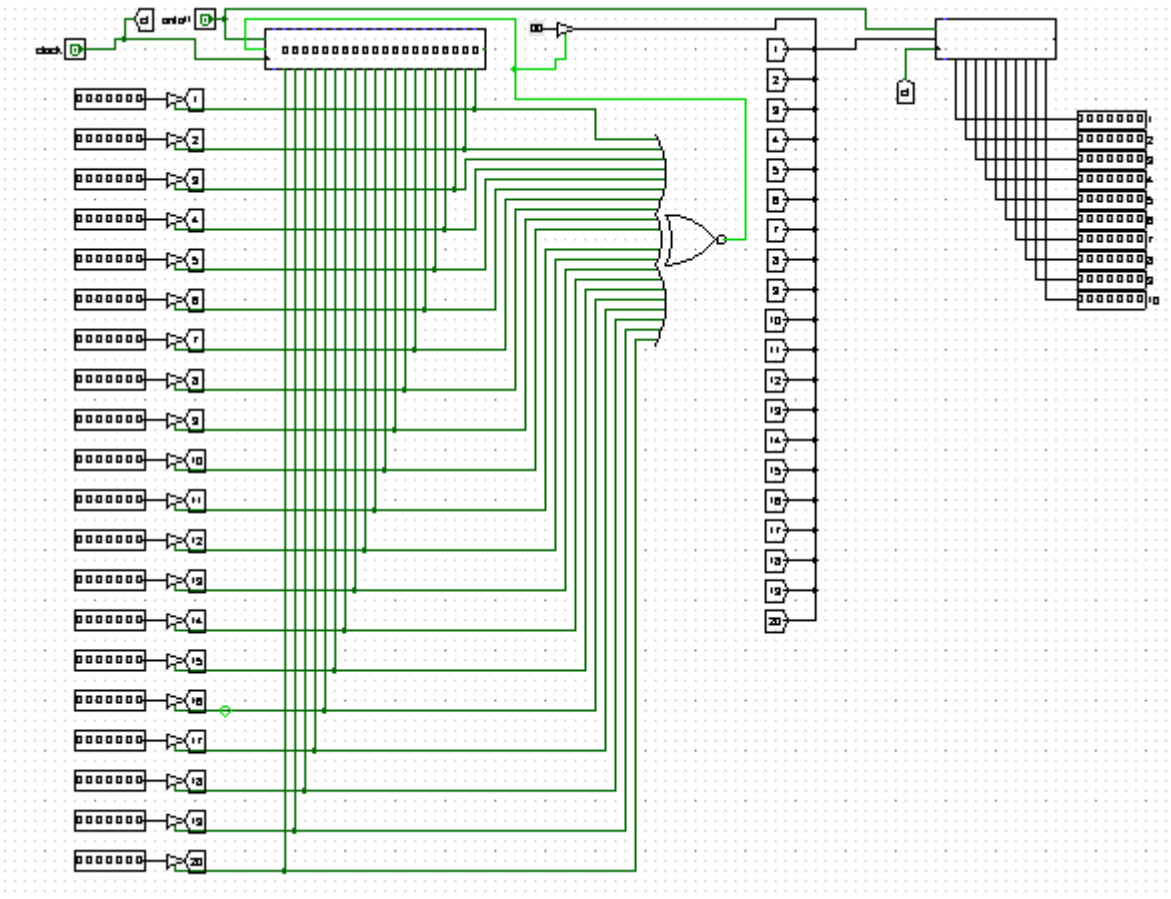


Figura 2 - Sottocircuito shiftlettera

### 3) BOTTONI

Il sottocircuito *Bottoni* (vedi fig. 3) associa ad ogni input di 1bit la codifica del relativo carattere e lo salva in *Memoria*.

Input: 37 input da 1bit, ognuno relativo ad un carattere.  
*Canc* 1bit se 1 cancella il contenuto della Memoria collegata  
*f1,f2,f3* 1bit attiva la relativa *Memoria*

Output: 20 output da 7bit delle codifiche dei caratteri  
*preview* 7bit codifica dell'ultimo carattere inserito  
*error* 1bit se 1 indica che non è stata selezionata alcuna stringa

Ogni input associato ad un carattere attiva un buffer da 7bit che associa la codifica relativa da una costante che viene salvata nel circuito *Memoria*.

Una porta XNOR restituisce l'*error* se *f1,f2,f3* sono 0.

Il tasto *canc* svuota il circuito *Memoria* attivo grazie a delle porte AND a cui sono collegati gli input *f1,f2,f3*.

L'ultima codifica viene salvata in un registro che ne fa anche l'output per la *preview*.

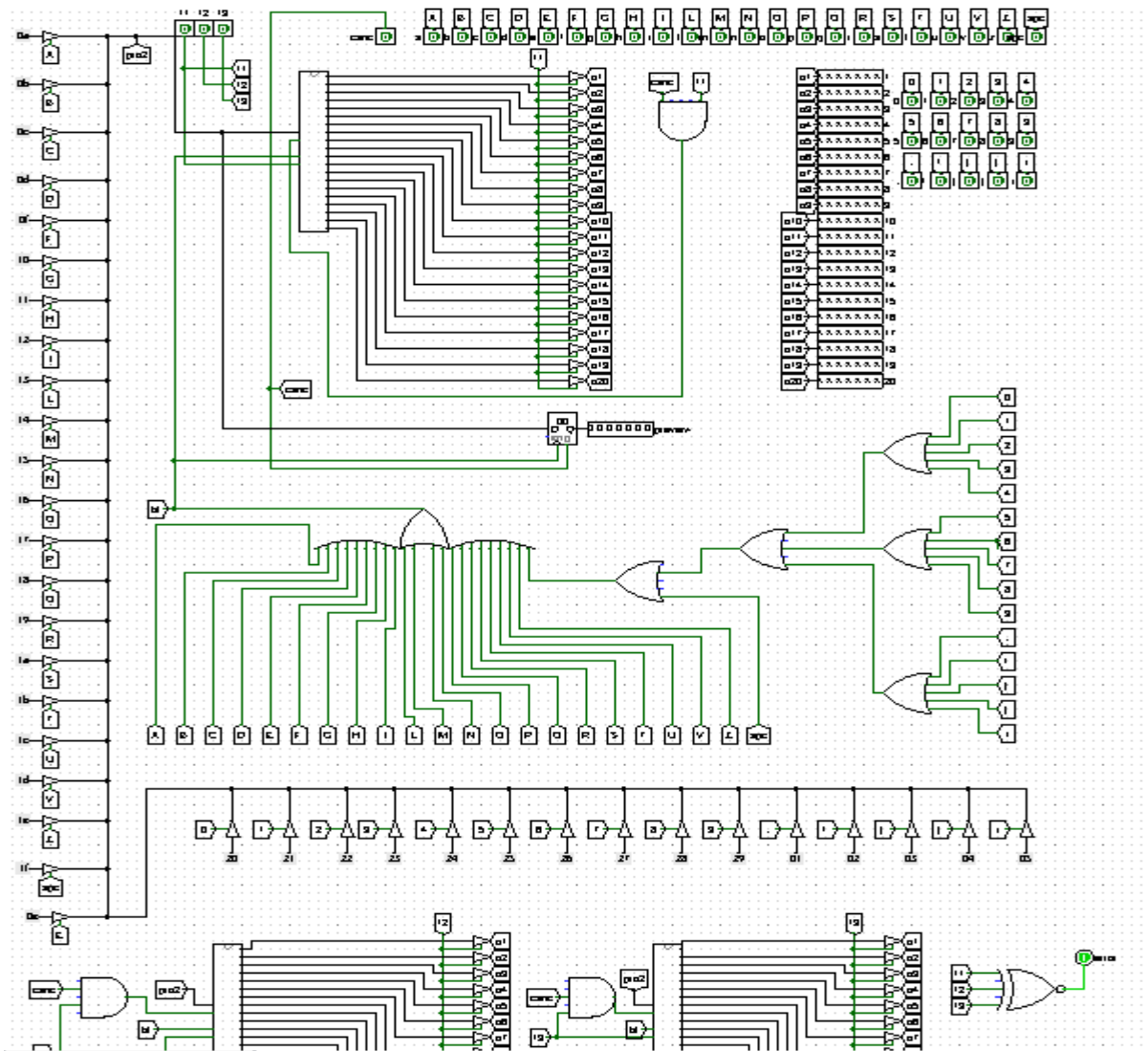


Figura 3 - Sottocircuito bottoni

#### 4) CARATTERI

Ad ogni costante ne vengono associate 5 da 7bit le quali rappresentano la bitmap del carattere (vedi fig. 4).

Input: *carattere* 7bit codifica del carattere

Output: 1 7bit costante a 0  
5 porte da da 7bit per la bitmap del carattere.

Il primo output è costante a 0 per rappresentare sulla matrice di led lo spazio tra i caratteri.

Un comparatore verifica che l'input sia uguale ad una costante e se tale condizione è vera attiva i buffer che mandano in output le costanti del bitmap.

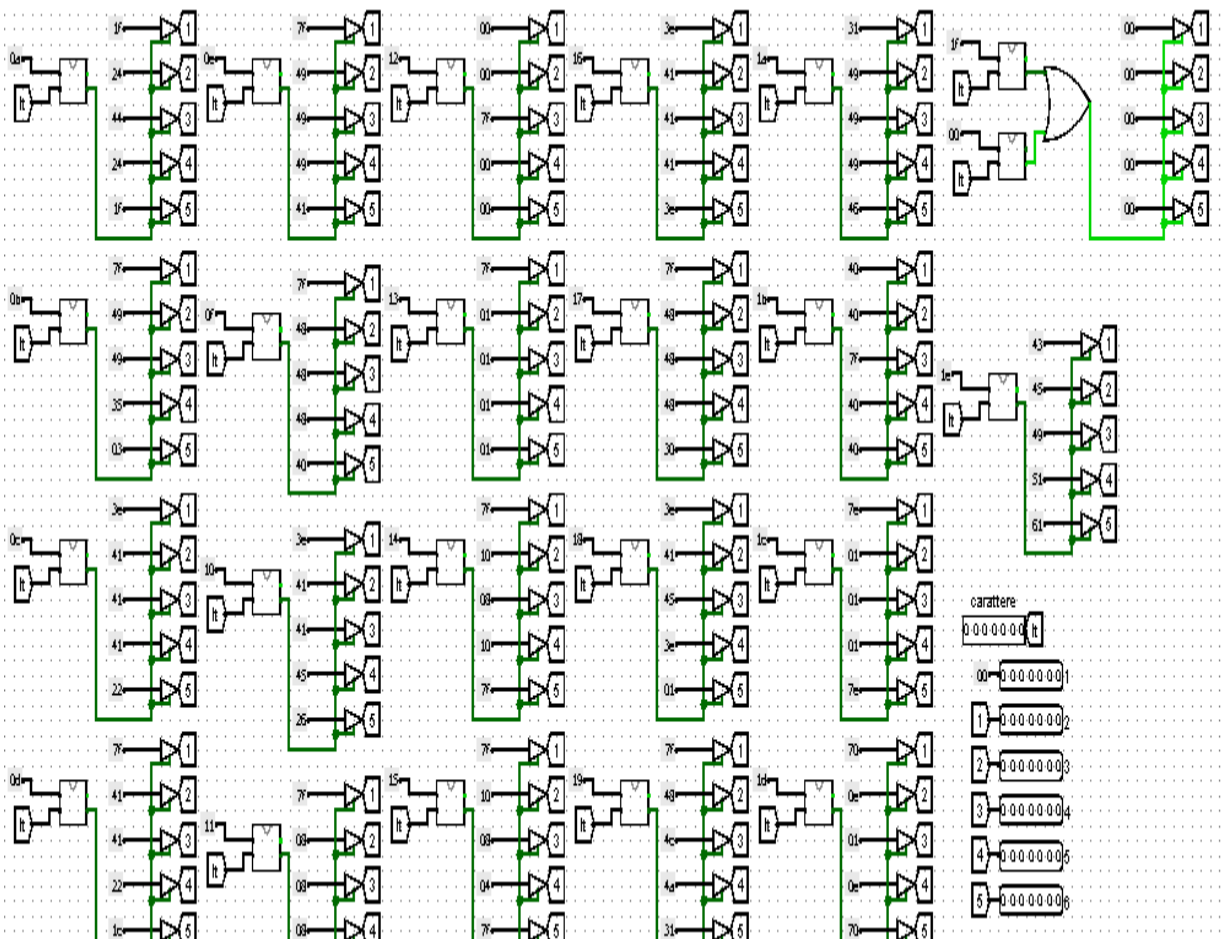


Figura 4 - Sottocircuito caratteri

### 5) COMPARATORE7BIT

Il sottocircuito comparatore7bit (vedi fig. 5) verifica che una sequenza di 7 bit sia uguale ad un'altra.

Input:        A        7bit  
              B        7bit

Output        =        1bit    se 1 allora A e B sono uguali  
              !=       1bit    se 1 allora A è diverso da B

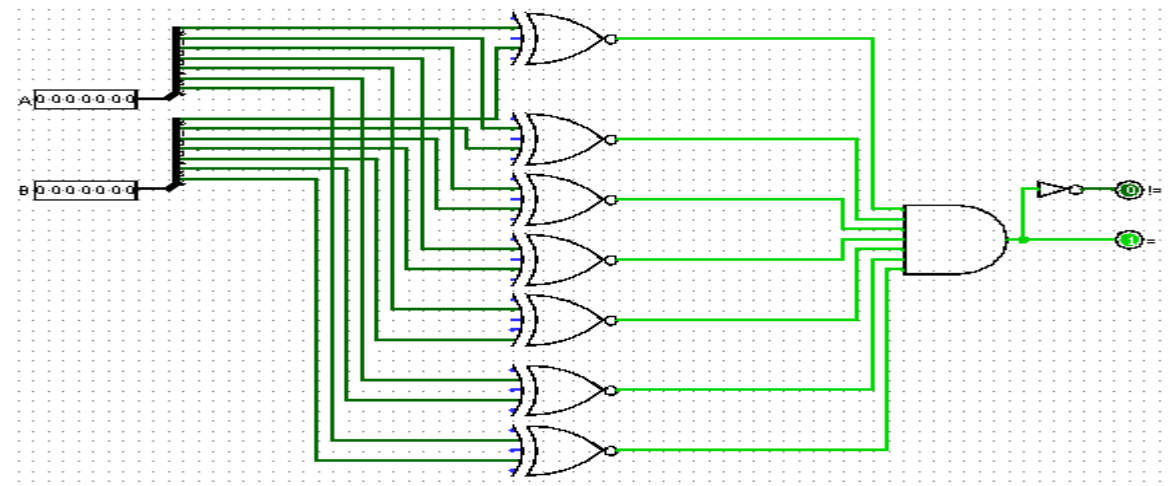


Figura 5 - Sottocircuito comparatore7bit

**TABELLA CODIFICA CARATTERI**

| <b>CHAR</b> | <b>BIN</b> | <b>HEX</b> |
|-------------|------------|------------|
| A           | 001010     | 0x0A       |
| B           | 001011     | 0x0B       |
| C           | 001100     | 0x0C       |
| D           | 001101     | 0x0D       |
| E           | 001110     | 0x0E       |
| F           | 001111     | 0x0F       |
| G           | 010000     | 0x10       |
| H           | 010001     | 0x11       |
| I           | 010010     | 0x12       |
| L           | 010011     | 0x13       |
| M           | 010100     | 0x14       |
| N           | 010101     | 0x15       |
| O           | 010110     | 0x16       |
| P           | 010111     | 0x17       |
| Q           | 011000     | 0x18       |
| R           | 011001     | 0x19       |
| S           | 011010     | 0x1A       |
| T           | 011011     | 0x1B       |
| U           | 011100     | 0x1C       |
| V           | 011101     | 0x1D       |
| Z           | 011110     | 0x1E       |
| 0           | 100000     | 0x20       |
| 1           | 100001     | 0x21       |
| 2           | 100010     | 0x22       |
| 3           | 100011     | 0x23       |
| 4           | 100100     | 0x24       |
| 5           | 100101     | 0x25       |
| 6           | 100110     | 0x26       |
| 7           | 100111     | 0x27       |
| 8           | 101000     | 0x28       |
| 9           | 101001     | 0x29       |
| .           | 000001     | 0x01       |
| +           | 000010     | 0x02       |
| [           | 000011     | 0x03       |
| ]           | 000100     | 0x04       |
| :           | 000101     | 0x05       |

## ITERAZIONE TRA SOTTOCIRCUITI

Tutti gli input del circuito principale ad eccezione dell' interruttore on/off sono collegati al sottocircuito "bottoni", questo ha lo scopo di associare ad ogni pulsante di ogni carattere una costante che codifica il carattere stesso, inoltre attiva una dei tre circuiti "memoria" integrati al suo interno, quest'ultimo sottocircuito ha la funzione di memorizzare le codifiche dei 20 caratteri della stringa. L'output della "memoria" attiva corrisponde con l'output del circuito "bottoni", le 20 sequenze di bit vengono mandate in input al circuito "shiftlettera" anch'esso integrato nel circuito principale. Questo sottocircuito, una volta ricevuti le stringe delle codifiche le fa scorrere 10 alla volta grazie a 2 componenti shift register. Le 10 sequenze di bit che vengono restituite come output da "shiftlettera" vengono codificate dal sottocircuito "caratteri" che restituisce per ogni sequenza di bit altre 5 sequenze da 7 bit che corrispondono alla bitmap del carattere. Le 6 uscite (una è costante 0) sono collegate direttamente con i display.

## COMPONENTI UTILIZZATI

- Pin
- LED matrix
- LED
- Pulsanti
- Costanti
- Buffer controllati
- Clock
- Splitter
- Porte OR
- Porte AND
- Porte XNOR
- Registri
- Shift register
- Tunnel

## PARTI DI SUPPORTO

- 3 pulsanti per la selezione della stringa da scrivere e visualizzare.
- Un pulsante stop per deselegione la stringa scelta.
- Un led rosso avvisa quando non è stata selezionata alcuna stringa.
- L'interruttore *on/off* attiva e disattiva lo scorrimento della stringa selezionata.

## POSSIBILI ESTENSIONI E MODIFICHE

- Una modifica potrebbe essere l'aggiunta del pulsante 'backspace' per cancellare l'ultimo carattere inserito.
- Potrebbe essere aggiunto un Random generator per selezionare una stringa a caso da far scorrere.
- La tastiera potrebbe essere ampliata con più caratteri.