

# Relazione del Progetto di Laboratorio Architettura degli Elaboratori I

Matteo Limoncini\*

matricola: 910311, Turno: B

matteo.limoncini@studenti.unimi.it

## 1 Test sui numeri

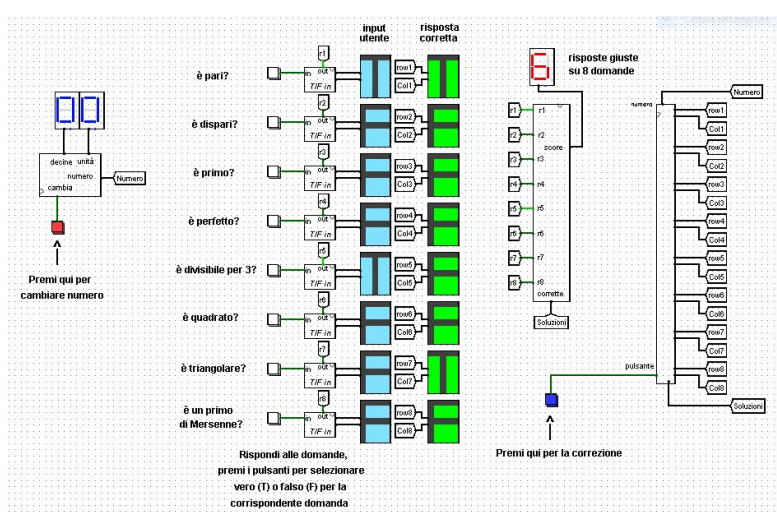


Figura 1: Interfaccia del progetto

Il mio progetto in Logisim consiste nella realizzazione di un circuito che simula un test in cui l'utente deve rispondere a diverse domande riguardanti le proprietà dei numeri naturali tra 0 e 15.

L'input avviene mediante dei pulsanti. Un primo pulsante permette di generare un numero a caso tra 0 e 15, 8 pulsanti permettono di rispondere vero o falso alla domanda corrispondente e un pulsante abilita la correzione e calcola il punteggio ottenuto. L'output è costituito da un display che mostra il numero generato in base 10. 8 display rappresentano la risposta data dall'utente (T o F) e 8 display mostrano la risposta corretta (T o F). Inoltre è presente un display che raffigura il punteggio dell'utente. All'inizio della simulazione i display con le risposte dell'utente sono settati tutti a F e il display rappresenta un numero generato casualmente.

\*Progetto approvato il giorno 15/07/2019, consegnato il 17/07/2019

## 2 Circuiti principali

### 2.1 Generatore numero casuale

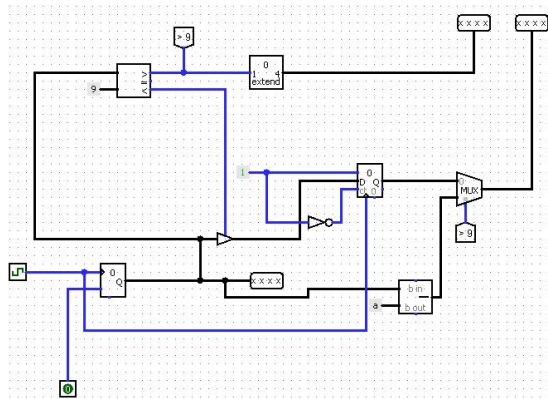


Figura 2: Circuito generatore del numero casuale

Il numero casuale è generato dall’RNG predisposto da Logisim. L’output, in base 10, è gestito nell’interfaccia utente tramite due display esadecimali affiancati, uno per le decine e uno per le unità. Viene gestito in modo diverso il caso in cui il numero sia minore strettamente minore di 10. In questo caso viene usato solo il display delle unità, altrimenti il display delle decine rappresenta il valore 1 e quello delle unità il numero generato a cui è stato sottratto 10.

### 2.2 Correzioni

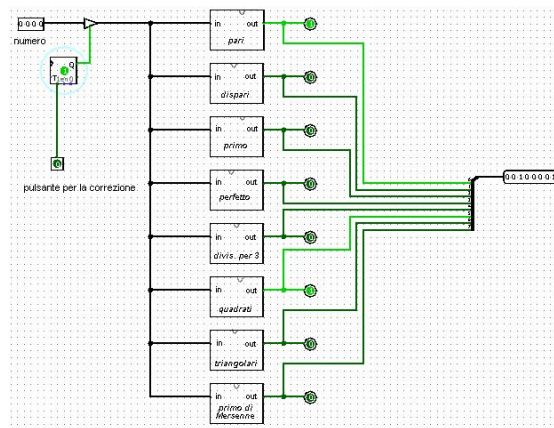


Figura 3: Circuito che genera le soluzioni corrette

Riceve il numero prodotto dal generatore casuale e viene attivato dall’utente tramite un pulsante. In output restituisce 8 bit che corrispondono alle soluzioni delle domande. Ogni bit avrà valore 0 se la risposta corretta è falso (F), 1 se la risposta corretta è vero (T). Al suo interno sono presenti 8 riconoscitori di proprietà necessari per la correzione.

## 3 Circuiti secondari

### 3.1 Output della correzione

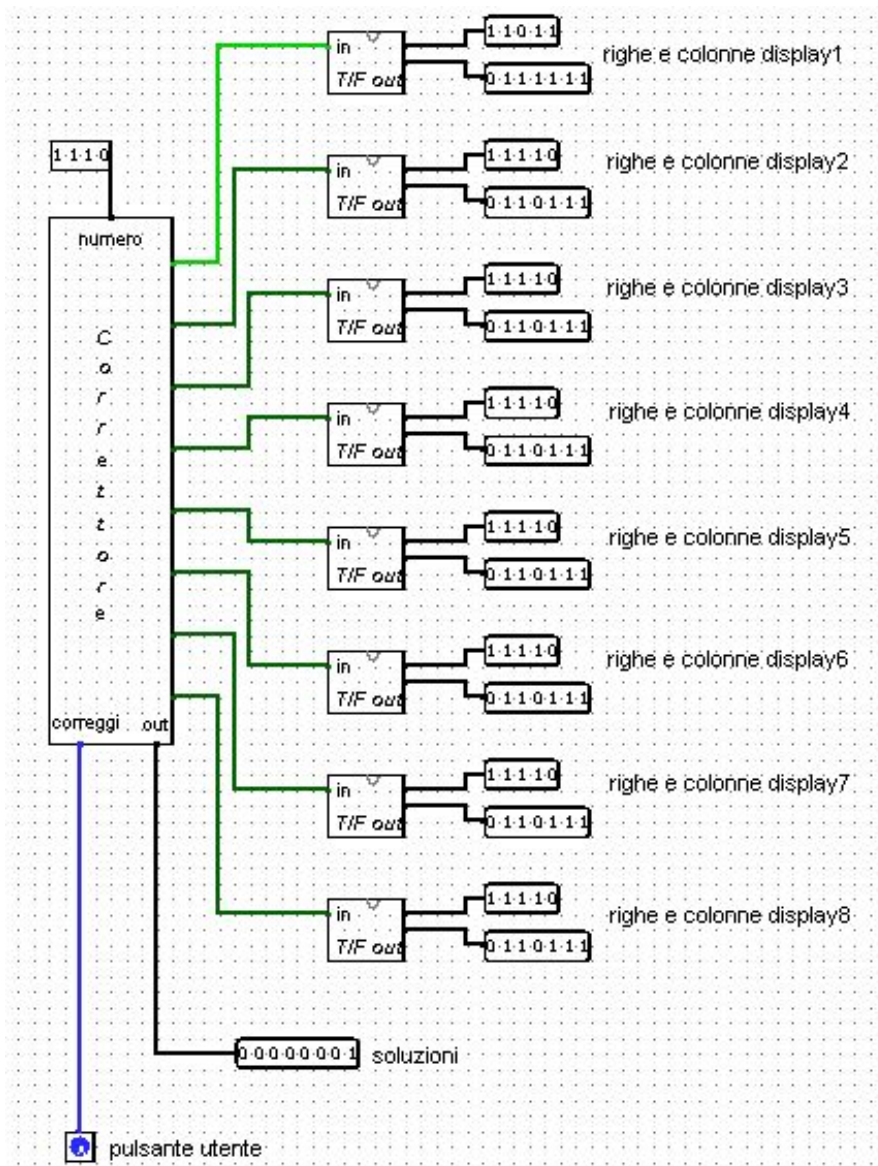


Figura 4: Circuito di output delle correzioni

Il circuito comprende il modulo per generare le soluzioni corrette e si occupa di gestire il suo output utilizzando 8 sottocircuiti T/F Display Output. Riceve in input il numero generato casualmente e l'input dell'utente. L'output è diviso per ogni display in 5 bit dedicate alla selezione delle righe e 7 bit alla selezione delle colonne. Inoltre restituisce 8 bit contenenti le soluzioni. Questo circuito è particolarmente utile per semplificare l'interfaccia utente.

### 3.2 T/F Display Output

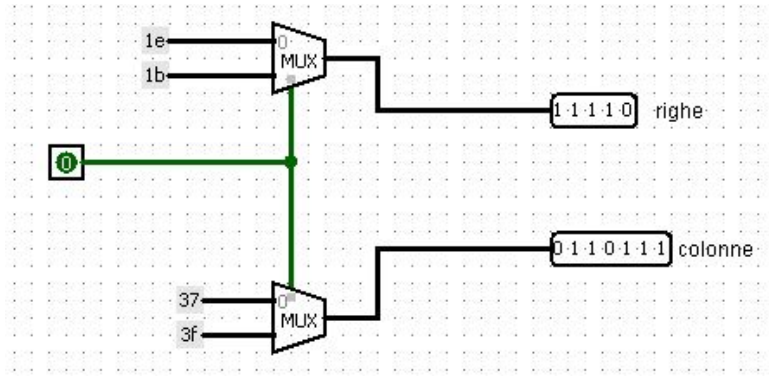


Figura 5: T/F Display Output

Permette una visualizzazione delle risposte corrette tramite delle matrici LED. Questo circuito è collegato alle soluzioni generate dal correttore. Vengono utilizzati due MUX binari per permettere la visualizzazione di due lettere distinte. I valori in uscita rappresentano la selezione delle righe e delle colonne. Le costanti in ingresso ai MUX permettono l'accensione di tutti i led eccetto quelli raffiguranti una T o una F.

### 3.3 T/F Display input

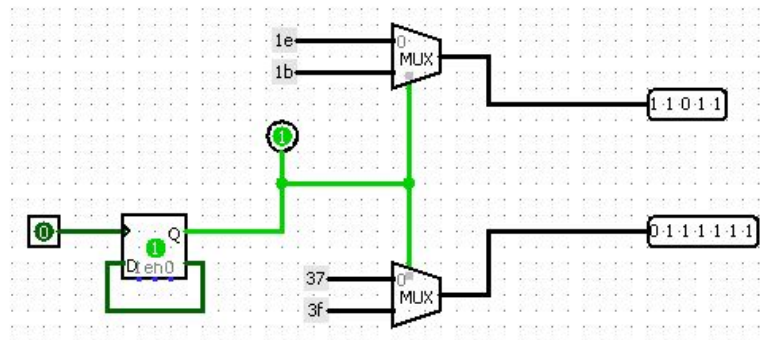


Figura 6: T/F display input

Permette all'utente, attraverso la pressione di un pulsante, di poter rispondere alla relativa domanda (Vero o Falso), visualizzando la scelta su una matrice LED che può indicare T o F. Si basa sul display di output con l'aggiunta di un Flip Flop di tipo D necessario per la memorizzazione dell'input dato dall'utente. In output avrò anche un bit che mi indica la risposta dell'utente (vero o falso).

### 3.4 Calcolo del punteggio

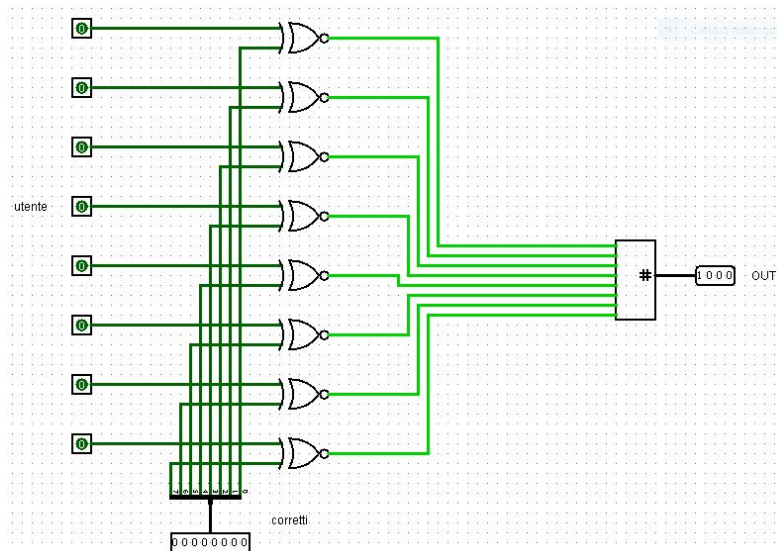


Figura 7: Circuito per il calcolo del punteggio

Ha la funzione di calcolare il numero di risposte corrette date dall'utente confrontandole con quelle generate dal correttore. L'output viene rappresentato con un display esadecimale. Il circuito confronta i bit di risposta dell'utente (1 se vero, 0 se falso) con i bit di risposta prodotti dal correttore e, se uguali, li somma producendo così il numero di risposte corrette.

## 4 Riconoscitori di proprietà

Ogni circuito riceve un numero di 4 bit, restituisce 1 solo se la il numero in input verifica la proprietà del corrispondente circuito. I circuiti sono stati realizzati partendo dalle tabelle di verità corrispondenti semplificandone il risultato.

### 4.1 Numeri pari

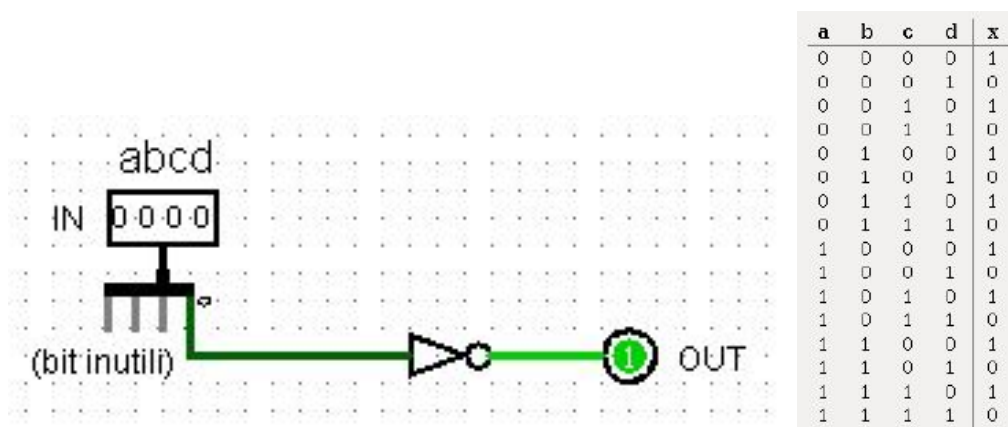


Figura 8: Circuito riconoscitore dei numeri pari e relativa tabella delle verità

formula logica:  $\neg d$



## 4.2 Numeri dispari

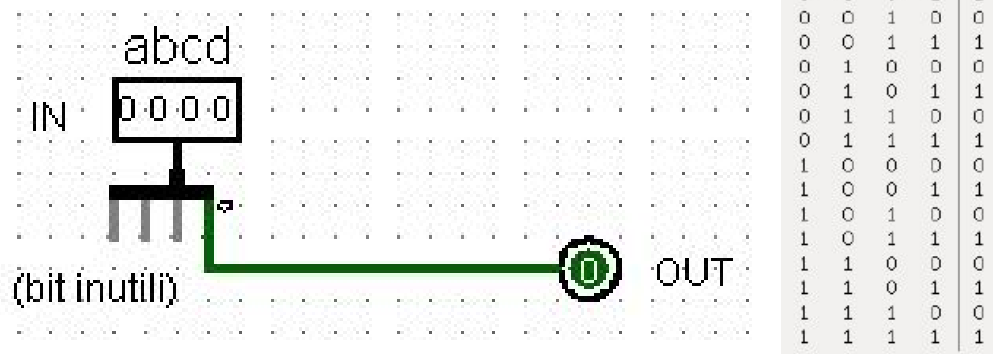


Figura 9: Circuito riconoscitore dei numeri dispari e relativa tabella delle verità

formula logica:  $d$

## 4.3 Numeri primi

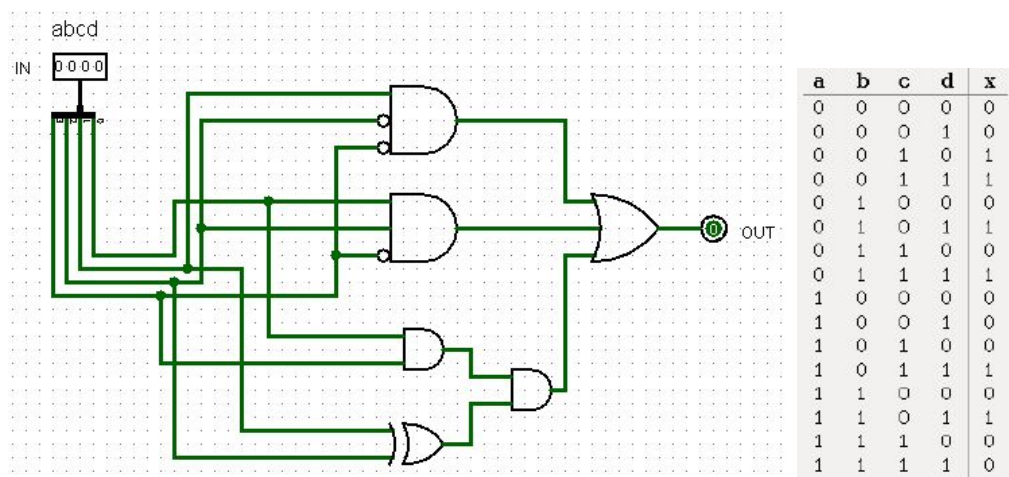


Figura 10: Circuito riconoscitore dei numeri primi e relativa tabella delle verità

formula logica:  $\neg a \neg bc + \neg abd + ad(b \oplus c)$

## 4.4 Numeri perfetti

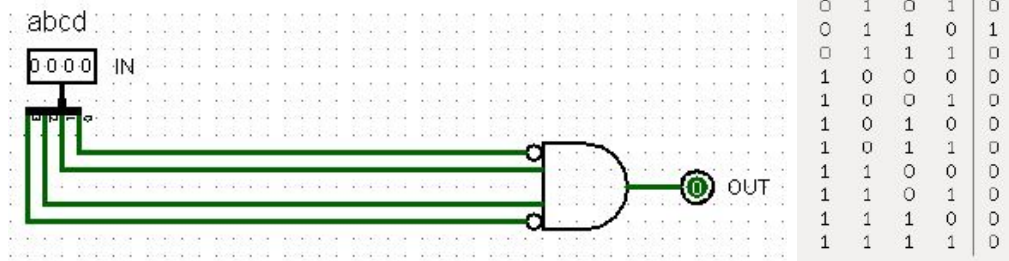


Figura 11: Circuito riconoscitore dei numeri perfetti e relativa tabella delle verità

formula logica:  $\neg abc \neg d$

## 4.5 Numeri divisibili per 3

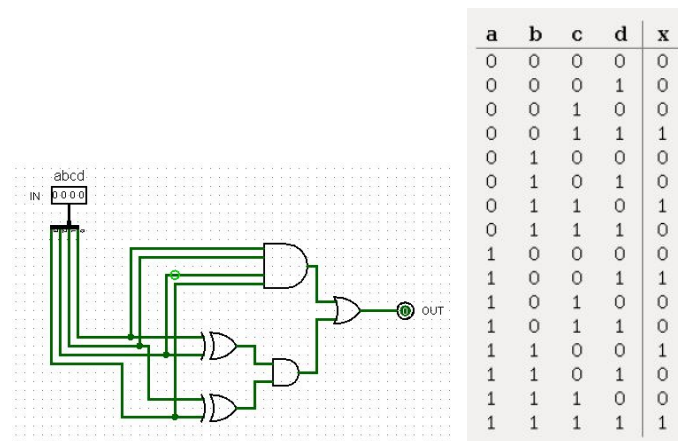


Figura 12: Circuito riconoscitore dei numeri divisibili per 3 e relativa tabella delle verità

formula logica:  $(a \oplus c)(b \oplus d) + abcd$

## 4.6 Numeri quadrati

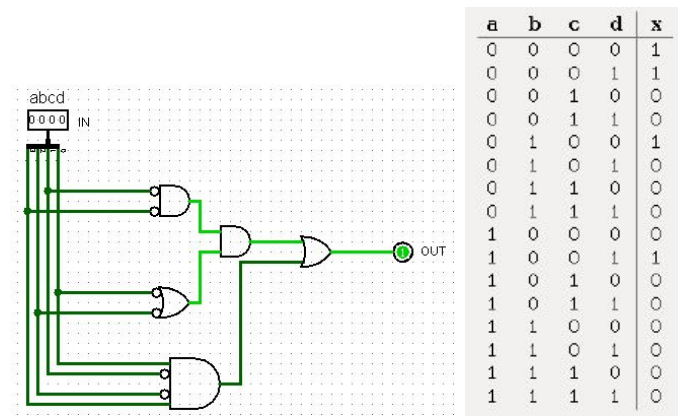


Figura 13: Circuito riconoscitore dei numeri quadrati e relativa tabella delle verità

formula logica:  $\neg a \neg c (\neg b + \neg d) + a \neg b \neg cd$

## 4.7 Numeri triangolari

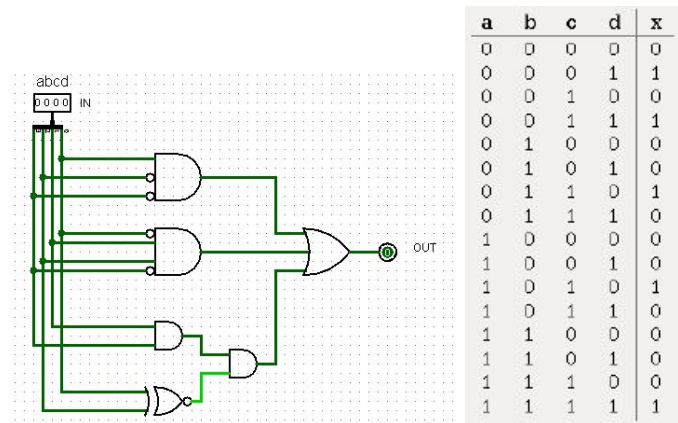


Figura 14: Circuito riconoscitore dei numeri triangolari e relativa tabella delle verità

formula logica:  $\neg a \neg bd + \neg abc \neg d + ac \neg (b \oplus d)$



## 4.8 Numeri primi di Mersenne

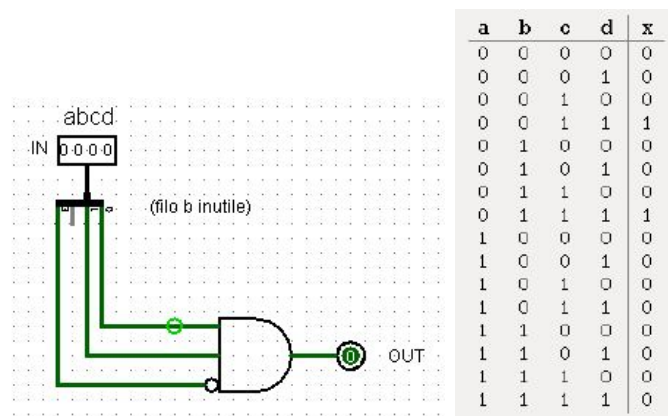


Figura 15: Circuito riconoscitore dei numeri primi di Mersenne e relativa tabella delle verità

formula logica:  $\neg acd$