



PROGETTO 012 MOLTIPLICATORE



OBIETTIVO

Consegna:

realizzare un circuito moltiplicatore, con ingressi a 4 bit e uscita ad 8 bit, simulare il circuito tramite Logisim.

DESCRIZIONE

Il sistema si compone di un paio di ingressi a e b a 4 bit ed un uscita mul ad 8 bit. Il circuito scelto è quello ottimizzato in quanto componenti ma avendo un tempo di calcolo di n cicli di clock. Il sistema si avvale infatti di un solo registro 8 bit per la memorizzazione del conteggio attuale, un'ALU a 4 bit e un circuito di controllo per sincronizzare le operazioni tra i diversi stage del moltiplicatore.

Il circuito moltiplicatore per eseguire il calcolo della moltiplicazione prende ispirazione dalla tradizionale moltiplicazione in colonna dove ogni volta il moltiplicando a viene riportato se la cifra n-esima del moltiplicatore b è settata 1, viceversa viene riportato uno 0. Ogni volta viene considerata una cifra successiva verso destra del moltiplicatore b e quindi, aggiungendo uno 0 sulla riga del prodotto n+1-esimo viene rieseguito il ciclo riportando il moltiplicando a oppure 0. Questo algoritmo viene ripetuto aggiungendo tanti 0 quanti sono i cicli passati e termina quando è stato scorso l'intero moltiplicatore b. Al termine tutti gli step, gli n risultati delle varie moltiplicazioni in colonna sono sommate ed il risultato ottenuto è difatti la moltiplicazione tra i due ingressi. L'operazione risulta corretta avendo in ingresso operandi con o senza segno, quindi complementati.

Tramite Logisim sarà poi simulato il circuito su diverse istanze di input, il calcolo sarà sviluppato sulla base del clock il quale scandirà le diverse operazioni interne del circuito, al raggiungimento del risultato corretto l'uscita sarà aggiornata e gli ingressi campioneranno i nuovi dati su cui basare la successiva moltiplicazione.

$$\begin{array}{r} 1\ 0\ 1\ 0 \\ 0\ 1\ 0\ 1 \end{array} \times =$$

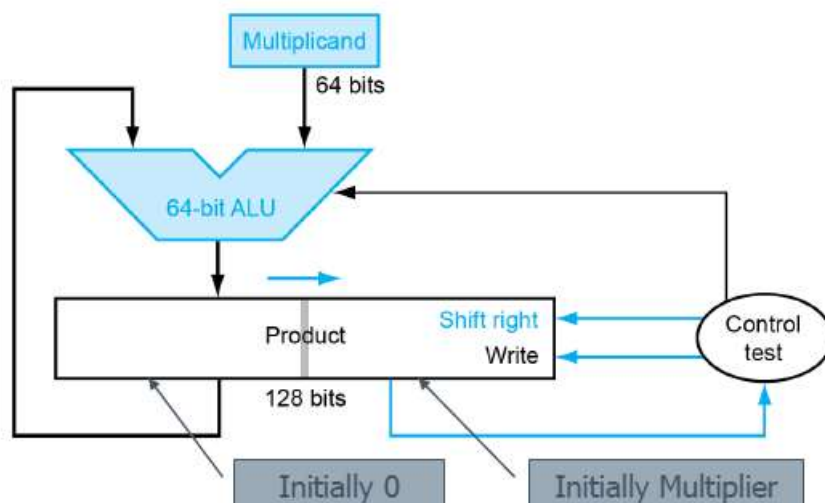
$$\begin{array}{r} 1\ 0\ 1\ 0\ 0 \\ 0\ 0\ 0\ 0\ 0 \\ 1\ 0\ 1\ 0\ 0\ 0 \end{array} + =$$

$$1\ 1\ 0\ 0\ 1\ 0$$

SVILUPPO

Sequenze di calcolo:

CICLO	1	2	3	4
OPERAZIONE	aggiornamento ingressi	calcolo parziale	calcolo parziale	aggiornamento uscite



Sincronizzazione:

il sistema per riconoscere il raggiungimento dell'esatto valore calcolato e riuscire dunque a portarlo in uscita aggiornando gli ingressi per la prossima operazione, fa riferimento ad un registro in grado di contare i cicli e scandire per cui le diverse operazioni da sviluppare.

Il circuito in linea di massima esegue sequenzialmente le seguenti operazioni:

- ricomponimento del dato da memorizzare sul registro 8 bit, in base al ciclo in cui il sistema si trova;
- shift right di una posizione del dato ricomposto, per simulare l'inserimento di uno 0 all'interno della somma parziale di una moltiplicazione sviluppata in colonna,
- esecuzione della somma tra la somma parziale ed il moltiplicando;
- ricomponimento del dato shiftato con MSB settato sul cout dell'operazione di somma svolta dalla ALU;
- memorizzazione sul fronte positivo del clock del dato ricevuto, la scrittura è sempre attiva per permettere allo step successivo di basare le proprie operazioni sul dato aggiornato;
- set segnali di controllo utili al prossimo ciclo di clock.

START	LSB		OUT
0	0		Mltplr 8 bit
0	1		Res + Mltplr
1	0		Prdct
1	1		Res + App

Ciclo 1:

Mltplcnd = 1010

Mltplr = 0101

LSB = 1

Data = 1010 0101

SR Data = 0101 0010

Register = 0101 0010

al primo ciclo il contatore è settato a 0 dunque il bit di start è ad 1. In questa condizione il dato ricomposto è formato dal solo moltiplicatore b su 8 bit oppure dal moltiplicatore b e dal risultato della prima somma parziale con il moltiplicando a. Con lo start LSB è letto sul moltiplicatore piuttosto che dal registro poiché il registro sarebbe a 0 o sporco da un'operazione precedente.

Distinguendo dunque il dato da selezionare e portare in uscita dal multiplexer si compie lo shift right e si aggiorna dunque lo stato del registro.

Ciclo 2:

Mltplcnd = 1010

Mltplr = 0101

LSB = 0

Data = 0101 0010

SR Data = 0010 1001

Register = 0010 1001

al secondo ciclo il contatore, con valore 1, resetta il bit di start. In questa condizione il dato ricomposto è formato dal prodotto parziale (valore registro) se LSB si trova a 0, pertanto al dato non viene aggiunto il moltiplicando a e viene solo shiftato. Con LSB ad 1, invece il dato è composto dalla somma parziale con il moltiplicando a ed i 4 bit inferiori del registro.

Ciclo 3:

Mltplcnd = 1010

Mltplr = 0101

LSB = 1

Data = 1100 1001

SR Data = 0110 0100

Register = 0110 0100

al terzo ciclo, il contatore, con valore 1, resetta il bit di start. In questa condizione il dato ricomposto è formato dal prodotto parziale (valore registro) se LSB si trova a 0, pertanto al dato non viene aggiunto il moltiplicando a e viene solo shiftato. Con LSB ad 1, invece il dato è composto dalla somma parziale con il moltiplicando a ed i 4 bit inferiori del registro.

Ciclo 4:

Mltplcnd = 1010

Mltplr = 0101

LSB = 0

Data = 0110 0100

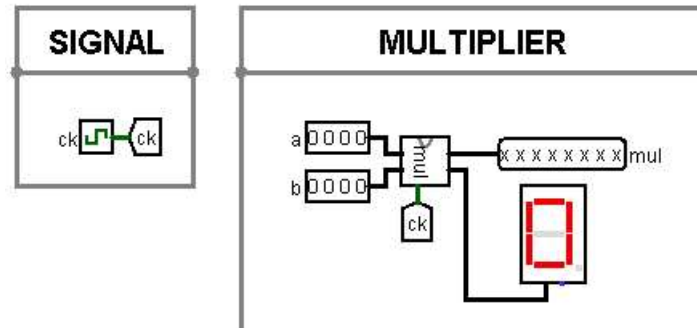
SR Data = 0011 0010

Register = 0011 0010

al quarto ed ultimo ciclo il contatore, con valore 1, resetta il bit di start. In questa condizione il dato ricomposto è formato dal prodotto parziale (valore registro) se LSB si trova a 0, pertanto al dato non viene aggiunto il moltiplicando a e viene solo shiftato. Con LSB ad 1, invece il dato è composto dalla somma parziale con il moltiplicando a ed i 4 bit inferiori del registro. Il contatore con valore 4 setta il bit di end il quale azzerava il contatore ed abilita la scrittura dell'uscita verso l'esterno. Il registro del prodotto parziale non necessita di un clear poiché al primo ciclo della moltiplicazione successiva non verrà accesso in lettura ma solamente in scrittura, non comportando per cui motivo di errore.

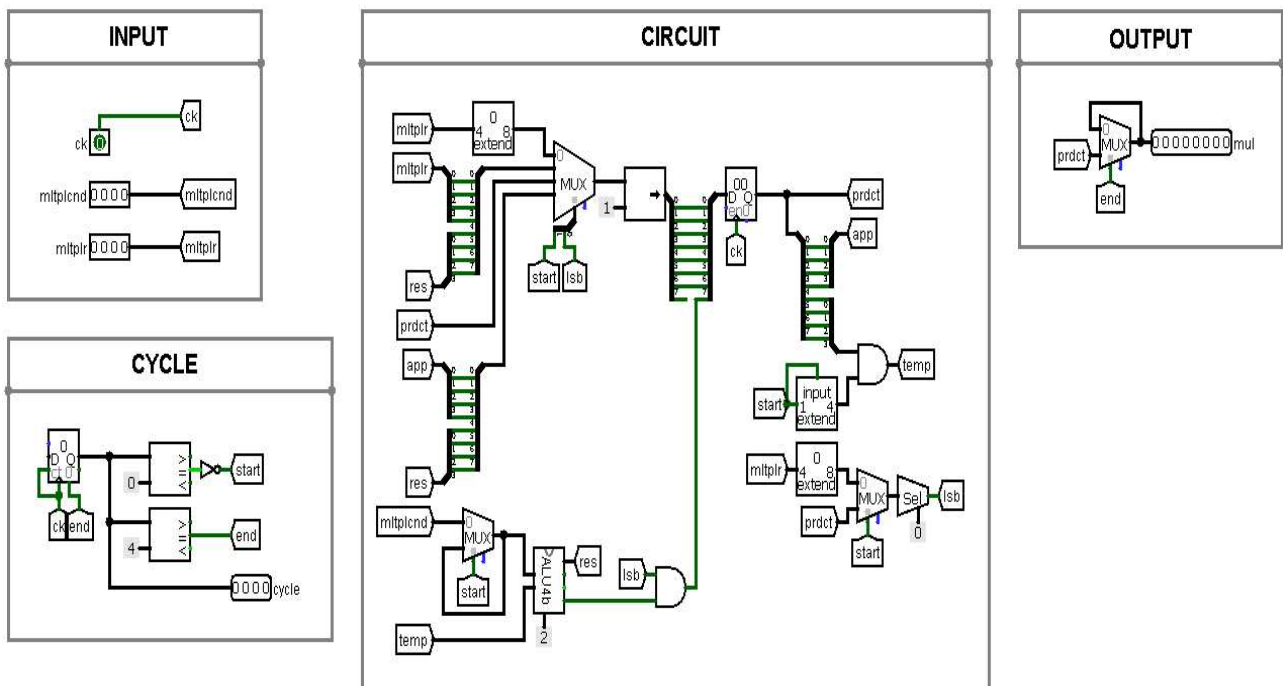
REALIZZAZIONE

Il circuito realizzato attraverso Logisim si compone di 3 circuiti principali: main, mul, ALU4b. Mentre il main rimane il circuito madre, gli altri sono tutti sottocircuiti sviluppati per rendere maggiormente ordinato e chiaro il funzionamento del sistema.



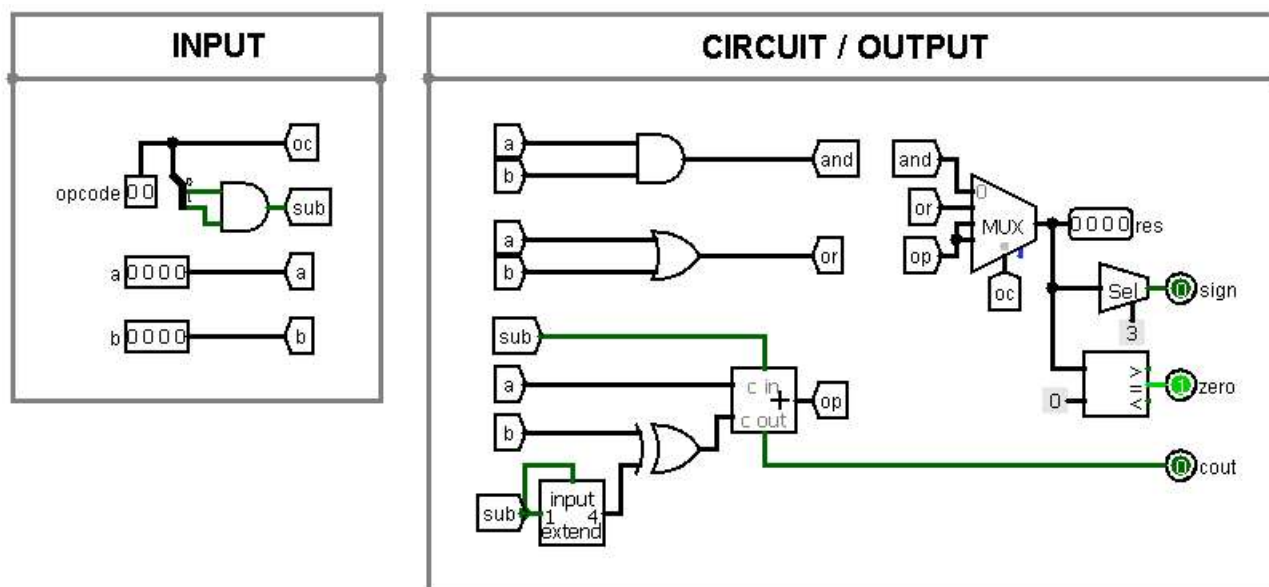
Main:

identificato come il circuito principale di ogni progetto Logisim, il main si occupa di gestire l'interfacciamento tra i segnali di input generali come il clock (per sincronizzare tutti gli elementi del sistema). Il sottocircuito mul esegue solamente l'operazione di moltiplicazione riportando in uscita lo stato attuale del sistema e quindi lo step attuale che, collegato ad un display 7 segmenti, permette di visualizzarne il valore in modo chiaro.



mul:

identificato come il circuito di controllo del sistema, questo sottocircuito implementa l'intera operazione di moltiplicazione tra i due operandi a e b 4 bit. Il multiplexer tramite i bit di start e LSB implementa la tabella di verità per la composizione del dato, il quale, passato ad uno shifter viene shiftato verso destra. MSB del dato viene rimpiazzato con il risultato della ALU, questa essendo a 4 bit potrebbe generare un overflow che setterebbe il bit ci cout, quest'ultimo collegato al MSB ricompone il giusto risultato dell'operazione su 5 bit. Di seguito sono poi riportati i vari segnali di controllo presi dal prodotto parziale.

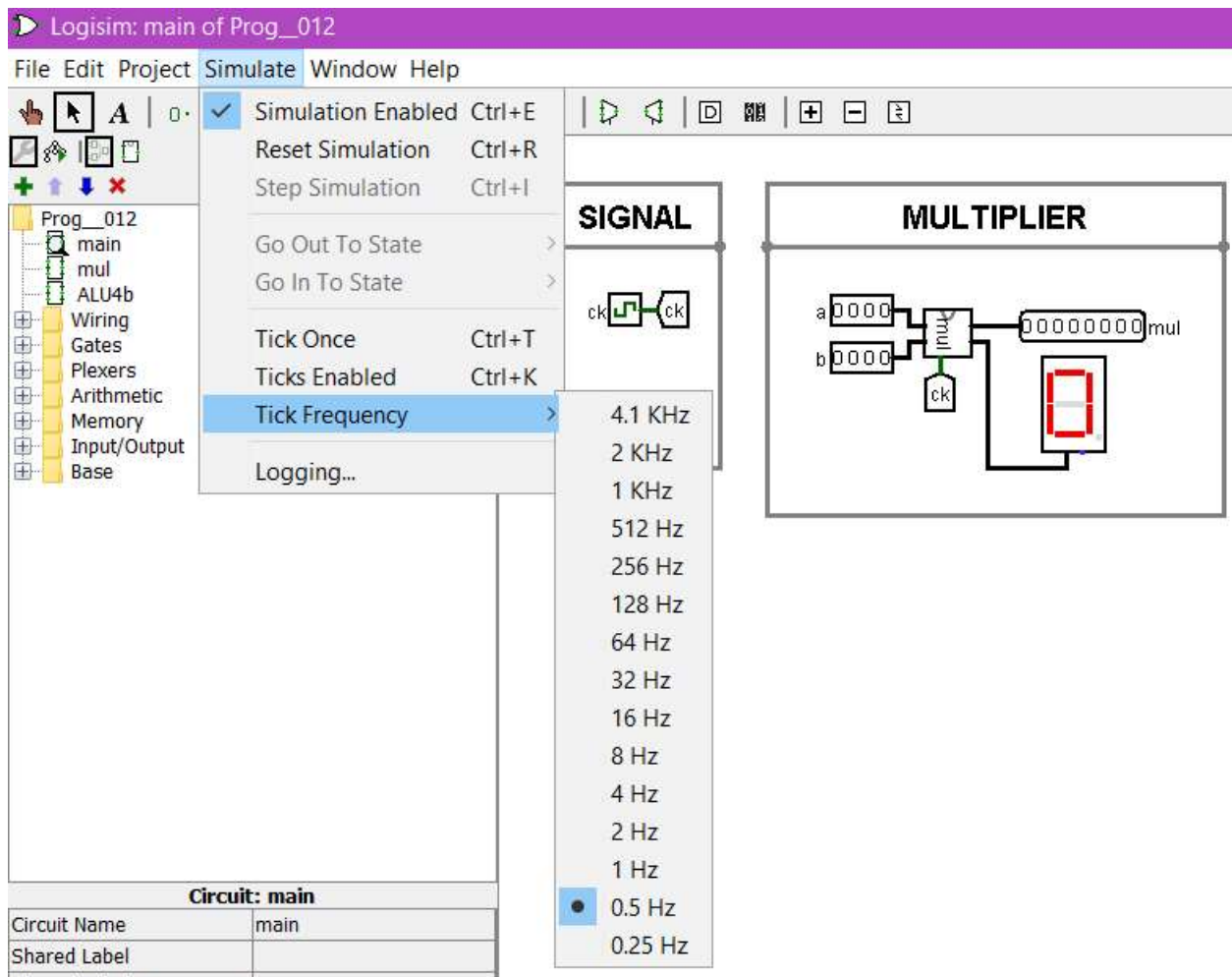


ALU4b:

identificato come il circuito base per la diverse operazioni fondamentali in ogni architettura, la ALU a 4 bit tramite un codice in ingresso presenta in uscita le operazioni di AND (codice 00), OR (codice 01), ADD (codice 10), SUB (codice 11). Le prime due sono semplicemente realizzate tramite porte logiche con fan-in e fan-out a 4 bit. La ADD e la SUB si appoggiano ad un circuito sommatore full-adder in cui il bit di cin è settato tramite il riconoscimento del codice 11 da parte di una porta AND. Con questo bit oltre a settare il cin del full-adder viene anche preso il dato b negato per eseguire il complemento a 2 del dato. $A - B \rightarrow a + C2(b) \rightarrow a + C1(b) + 1 \rightarrow a + \bar{b} + 1 \rightarrow a + \text{exor}(b, 1) + 1$.

In uscita dalla ALU oltre al risultato dell'operazione corrente sono inoltre riportati i bit di segno, zero (caso di risultato nullo dell'operazione), cout (overflow dell'operazione produce un riporto).

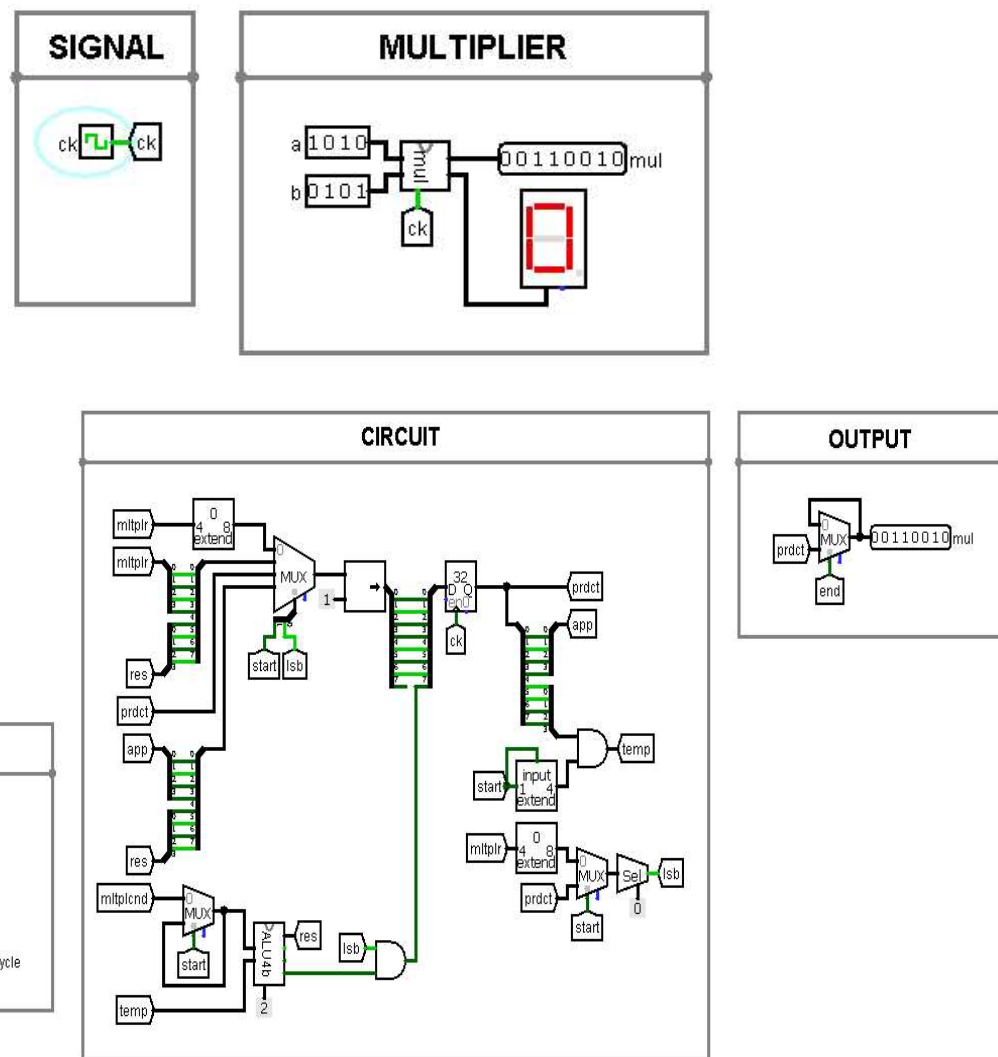
TEST



Impostazioni Logisim:

per realizzare un test pratico del circuito con Logisim bisogna portarsi in fase di simulazione tramite la voce Simulate → Simulation Enabled.

Potrebbe essere utile anche settare il clock sempre attivo in maniera di non doversi occupare anche del clock, Simulate → Ticks Enabled, e definire una frequenza, Simulate → Tick Frequency → xxx.x Hz. Nel caso specifico il clock per comodità sarà comandato dall'utente in maniera da scandire meglio e con tranquillità gli stati ed ingressi del sistema. Da notare che l'unico clock inserito all'interno del progetto è quello del circuito main, tutti gli altri sottocircuiti infatti, assumendo il clock come ingresso, presuppongono il fatto che questo sia comandato dall'esterno. Sarebbe considerato come errore inserire diversi clock nel progetto, solo uno comune e generale deve gestire e dunque coordinare tutti gli elementi del circuito Logisim.



Simulazione:

In fase di esecuzione il circuito risulterà circa in queste condizioni, le linee evidenziate indicano quali bit sono ad 1, viceversa, a 0 o anche per canali di n bit, saranno di colore verde scuro. Le varie label facilitano notevolmente la lettura e chiarezza del circuito, nonostante anche queste non assumano alcuna colorazione in base allo stato del dato. Il display visualizza lo stato del moltiplicatore mentre l'uscita è stata settata al valore degli ingressi letti in precedenza, nel caso specifico la sequenza quella presa come esempio.

CONCLUSIONI

Il progetto approfondito si dimostra essere, teoricamente, lievemente diverso da quello trattato in classe, il bit di write infatti non è stato utilizzato in quanto, altrimenti, le operazioni successive, non essendo aggiornato lo stato del registro con un LSB pari a 0, si sarebbero basate su un valore diverso. Il moltiplicando a inoltre non viene salvato in un registro di appoggio ma usato immediatamente come ingresso per ricomporre il dato da shiftare in seguito. Nonostante queste divergenze teoriche, il circuito svolge il suo compito nei n cicli di clock prestabiliti, elaborando correttamente moltiplicazioni con e senza segno.