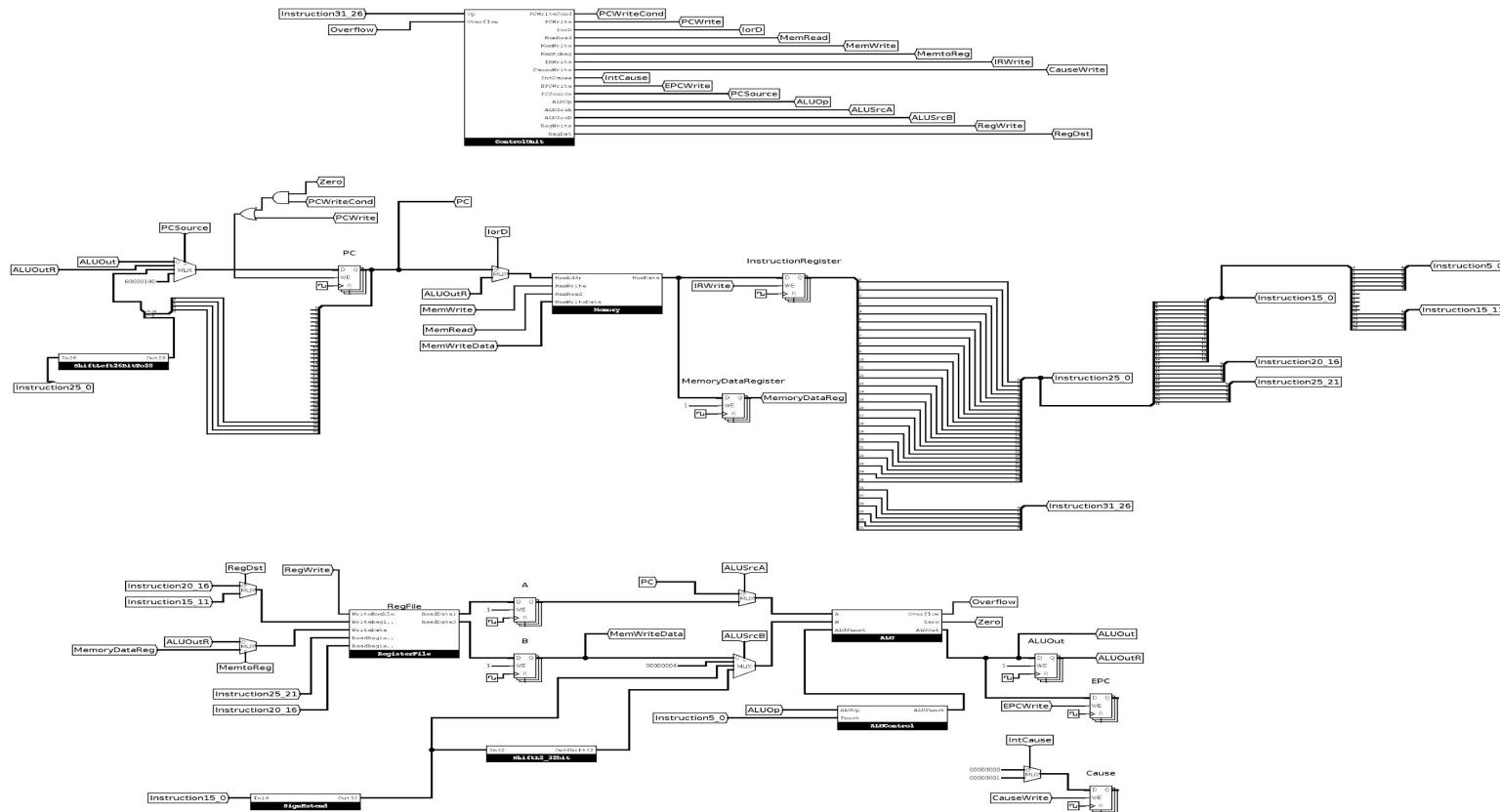


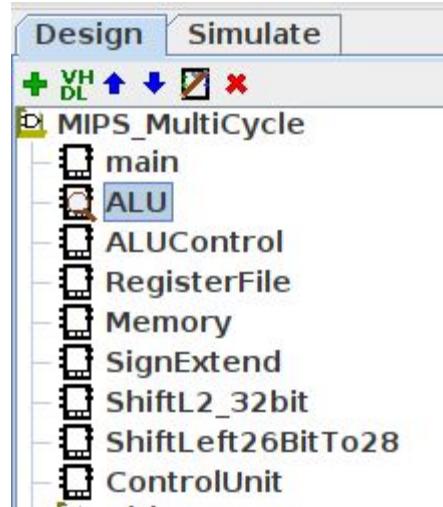
Datapath multiciclo in Logisim

Struttura generale

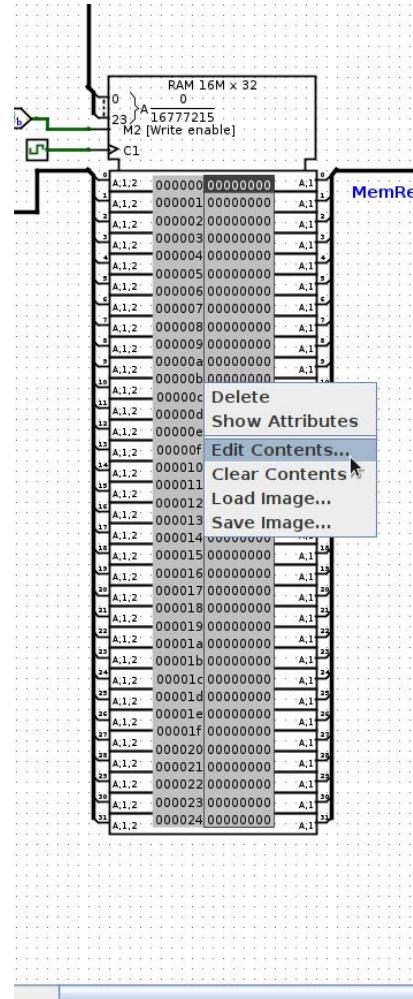
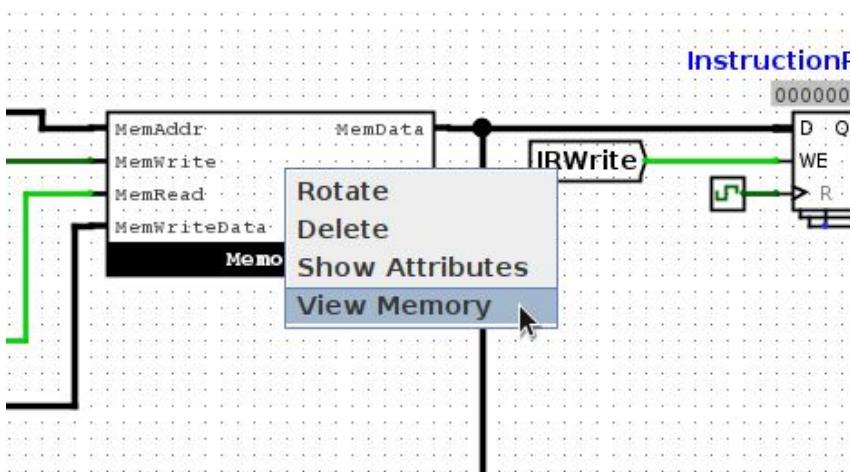


Sotto-componenti

Potete visualizzare i vari componenti del datapath utilizzando il menù a sinistra



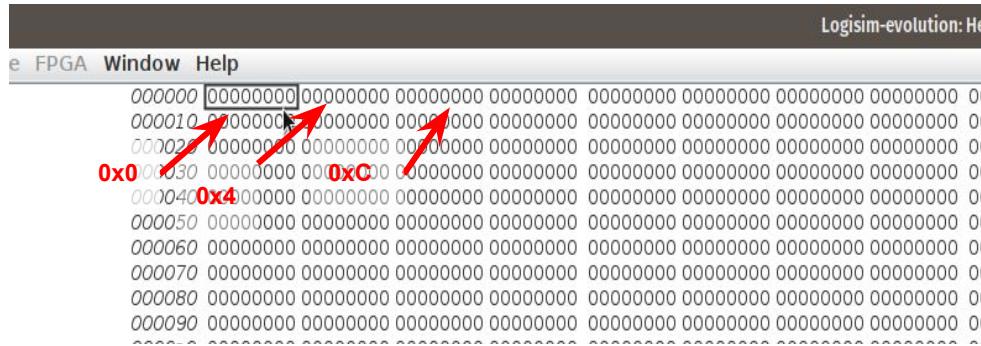
Caricare un programma (1)



Caricare un programma (2)

Inserire in formato esadecimale le istruzioni e i dati.

ATTENZIONE: gli indirizzi di questa memoria sono allineati a word, non è possibile l'indirizzamento al byte



Eseguire un programma

Verificate che l'opzione “Run Simulator” sia attiva.

Cliccando su “Tick Full Cycle” (o premendo F2) verrà eseguito un ciclo di clock.

Potete verificare sia il funzionamento del datapath (main) che dei sotto-componenti cliccando su essi nel menù “Simulate” a fianco a “Design”



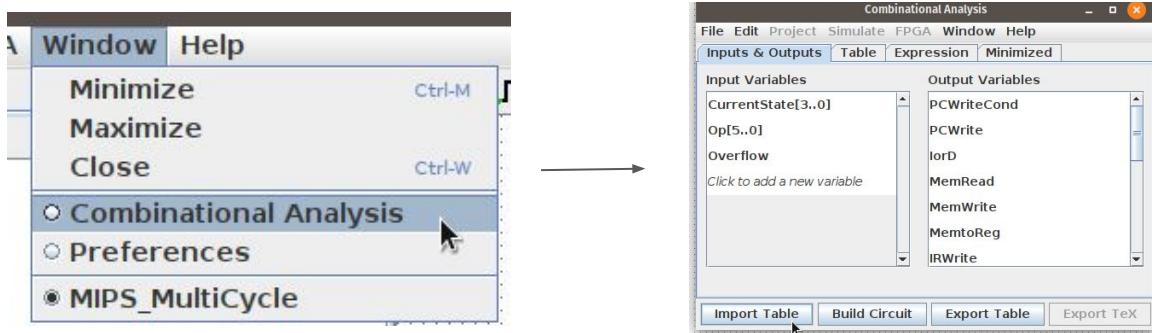
Tips & Tricks per modificare il datapath

Control Unit (1)

La control unit è generata grazie ad un tool di Logisim Evolution che permette di creare un circuito combinatorio a partire dalla sua tabella di verità.

Potete importare la tabella di verità dal file .txt fornito (lo stesso vale per l'ALU Control).

Essendo la Control Unit una FSM abbiamo bisogno, oltre agli input e output "standard" che potete vedere dallo schema del libro, anche un input "CurrentState" e in output un "NextState".



Control Unit (2)

Nella tab “Table” è presente la tabella di verità della Control Unit.

I bit indicati con “-” sono bit da cui non dipende l'output. Ad esempio se ci troviamo nello stato 0, a prescindere dall'opcode e dal segnale di overflow in input, si avranno sempre gli stessi segnali in uscita.

CurrentState[3..0]	Op[5..0]	Overflow
0 0 0 0	- - - - -	-

Control Unit (3)

Sappiamo che per tutte le istruzioni le prime due fasi sono uguali.

Quando ci troviamo nel secondo stato (0001) dobbiamo passare ad un nuovo stato in base all'opcode letto.

Per fare questo andiamo in corrispondenza degli input:

CurrentState -> 0001

Op -> l'opcode interessato

E andiamo ad inserire in NextState il numero dello stato in cui vogliamo andare.

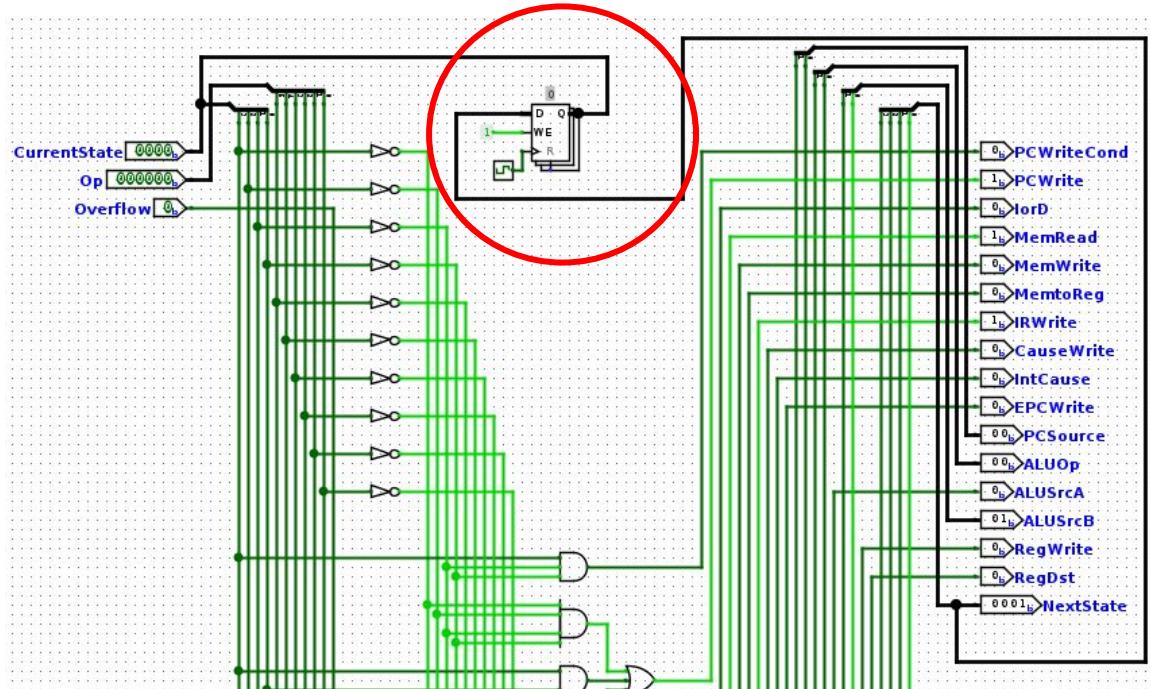
Ricordiamo che gli stati utilizzati dal datapath originale vanno da 0000 a 1011, quindi possiamo sfruttare i numeri successivi per creare nuovi stati.

Control Unit (4)

Dopo aver opportunamente modificato la tabella di verità possiamo generare il circuito cliccando su “Build Circuit”.

Il circuito generato sarà un circuito combinatorio. Essendo il datapath un circuito sequenziale, dobbiamo aggiungere un registro per conservare lo stato corrente della FSM, che andrà in input al “CurrentState” e verrà aggiornato dal “NextState” ad ogni ciclo di clock.

Possiamo poi cancellare i “tunnel” CurrentState e NextState.



Modificare il numero di bit di un segnale (1)

Può essere utile cambiare il numero di bit dei segnali di controllo.

Per farlo andare nella tab “Inputs & Outputs”, individuare l’output e fare doppio click. Selezionare poi il numero di bit.

Se si vanno ad aggiungere dei bit a dei segnali di controllo ricordarsi di inserire 0 nella tabella di verità in corrispondenza dei nuovi bit segnati come “-”



ALUSrcA[1..0]

Modificare il numero di bit di un segnale (2)

Quando si cambia la risoluzione dei segnali nella Control Unit vanno adeguati anche i componenti (ad esempio multiplexer e tunnel) che usano questi segnali.

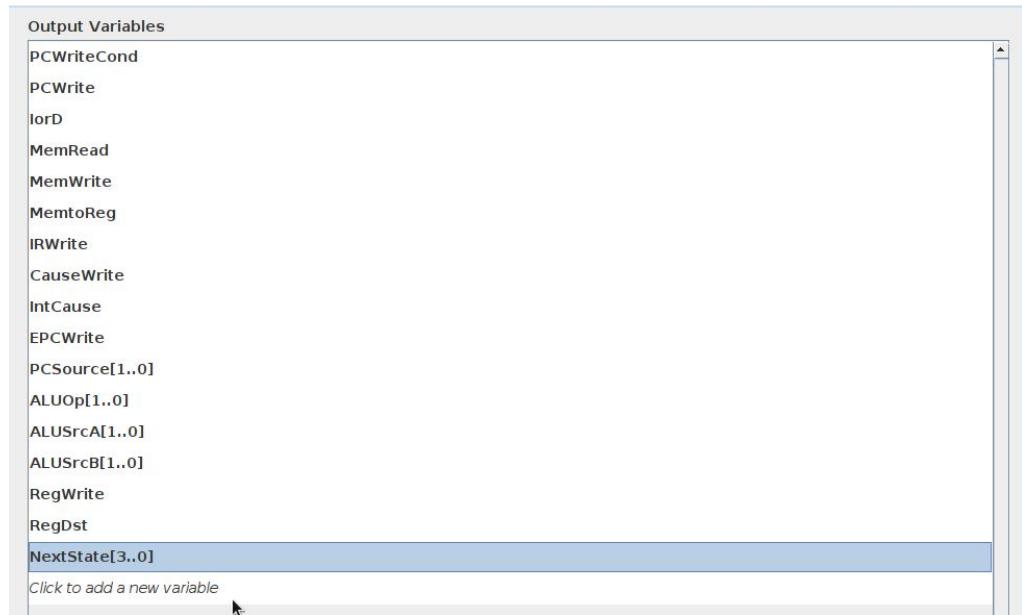
Selection: Multiplexer	
VHDL	Verilog
Facing	East
Gate Size	Wide
Select Location	Top/Right
Select Bits	2
Data Bits	32
Disabled Output	Zero
Include Enable?	No

Selection: Tunnel "ALUSrcA"	
VHDL	Verilog
Facing	South
Data Bits	2
Label	ALUSrcA
Label Font	SansSerif Bold 16

Aggiungere segnali di controllo (1)

Per aggiungere un segnale di controllo andare nella tab “Inputs & Outputs”, clickare “Add a new variable” , scrivere il nome del segnale e selezionare il numero di bit voluto.

Nella tabella di verità inserire i valori del segnale, anche in corrispondenza delle righe già esistenti.



Aggiungere segnali di controllo (2)

Dopo aver sistemato la ControlUnit
possiamo trovare il nuovo segnale in
output.

Aggiungiamo un componente di tipo
“Tunnel” collegato all’output della CU
per poter usare il segnale come input
all’interno del datapath.

Il segnale sarà ora accessibile
ovunque utilizzando un altro
componente di tipo “tunnel” con la
stessa label.

