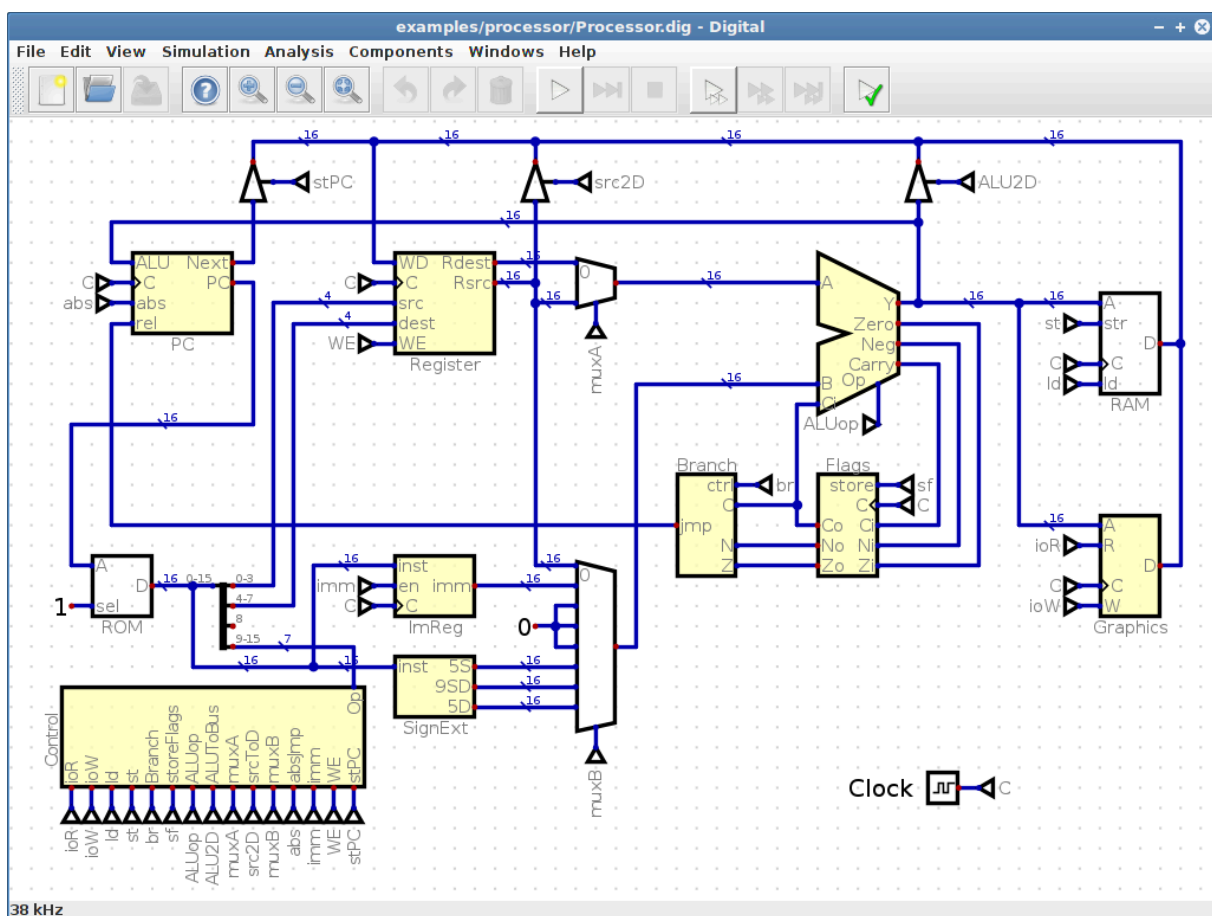


# Digital



Revisione: v0.31  
 Data: 2024-09-03 12:57

# Indice

## A Generale

1. Digital .....	6
1.1. Introduzione .....	6
1.2. Primi Passi .....	6
1.3. Cavi .....	14
1.4. Progettazione Gerarchica .....	14
2. Simulazione .....	18
2.1. Ritardo di Propagazione .....	18
3. Analisi .....	18
3.1. Analisi e Sintesi dei Circuiti .....	18
3.2. Espressione .....	18
3.3. Grafici di Stato .....	19
4. Hardware .....	19
4.1. GAL16v8 e GAL22v10 .....	19
4.2. ATF150xAS .....	19
4.3. Esportare a VHDL o Verilog .....	19
5. Figure Personalizzate .....	20
6. Circuiti Generici .....	21
7. Verifica Controllata da Script .....	22
8. FAQ .....	22
9. Comandi da Tastiera .....	24

## B Impostazioni

## C Interfaccia a Linea di Comando

## D Componenti

1. Logica	
1.1. And .....	31
1.2. NAnd .....	31
1.3. Or .....	32
1.4. NOR .....	33
1.5. XOr .....	33
1.6. XNOr .....	34
1.7. Not .....	35
1.8. Tabella di Lookup .....	35
2. IO	
2.1. Uscita .....	36
2.2. LED .....	37
2.3. Ingresso .....	37
2.4. Ingresso Clock .....	38
2.5. Bottone .....	39
2.6. Interruttore DIP .....	39
2.7. Sonda .....	40
2.8. Grafico Dati .....	40
2.9. Grafico Dati ad Attivazione .....	41
3. IO - Schermi	
3.1. LED RGB .....	41

3.2. LED con due connessioni. ....	42
3.3. Bottone con LED ....	42
3.4. Display a Sette Segmenti ....	43
3.5. Display a Sette Segmenti Esadecimale ....	44
3.6. Display a 16 Segmenti ....	44
3.7. Lampada ....	45
3.8. Matrice di LED ....	45
4. IO - Meccanici	
4.1. Trasduttore di Posizione a Rotella ....	46
4.2. Motore Passo-Passo Unipolare ....	46
4.3. Motore Passo-Passo Bipolare ....	47
5. IO - Periferiche	
5.1. Tastiera ....	48
5.2. Terminale ....	49
5.3. Telnet ....	49
5.4. Monitor VGA ....	50
5.5. MIDI ....	51
6. Cavi	
6.1. Massa ....	51
6.2. Tensione di alimentazione ....	52
6.3. Valore costante ....	52
6.4. Tunnel ....	53
6.5. Scissione/Divisione ....	53
6.6. Driver ....	54
6.7. Driver, selezione invertita ....	55
6.8. Ritardo. ....	55
6.9. Restitutore di Pull-Up ....	56
6.10. Resistore di Pull-Down ....	56
6.11. Non Connesso ....	57
7. Plexer	
7.1. Multiplexer ....	57
7.2. Demultiplexer ....	58
7.3. Decodificatore ....	58
7.4. Selettore del Bit ....	59
7.5. Codificatore a Priorità ....	59
8. Flip-Flop	
8.1. Flip-flop RS ....	60
8.2. RS-Flip-flop con Clock ....	61
8.3. Flip-flop JK ....	62
8.4. Flip-flop D ....	63
8.5. Flip-flop D ....	63
8.6. JK-Flip-flop, asincrono ....	64
8.7. D-Flip-flop, asincrono ....	65
8.8. Monoflop ....	66
9. Memoria - RAM	
9.1. RAM, porte separate ....	67
9.2. Block-RAM, porte separate ....	68
9.3. RAM, porte bidirezionali ....	69
9.4. RAM, Chip Select ....	70
9.5. File Registro ....	71

9.6. RAM, Doppia Porta .....	72
9.7. RAM, asincrona .....	73
9.8. RAM Grafica .....	74
10. Memoria - EEPROM .....	
10.1. EEPROM .....	75
10.2. EEPROM, porte separate .....	76
11. Memoria .....	
11.1. Registro .....	77
11.2. ROM .....	78
11.3. ROM a doppia porta .....	79
11.4. Contatore .....	80
11.5. Contatore con preset .....	81
11.6. Generatore di Numeri Casuali .....	82
12. Aritmetica .....	
12.1. Sommatore .....	83
12.2. Sottrattore .....	83
12.3. Moltiplicatore .....	84
12.4. Divisore .....	85
12.5. Barrel shifter .....	85
12.6. Comparatore .....	86
12.7. Negazione .....	87
12.8. Estenditore del segno .....	87
12.9. Contatore di bit .....	88
13. Interruttori .....	
13.1. Interruttore .....	88
13.2. Interruttore a Doppio Contatto .....	89
13.3. Relay .....	89
13.4. Relay a Doppio Contatto .....	90
13.5. FET a Canale P .....	91
13.6. FET a Canale N .....	92
13.7. Fusibile .....	92
13.8. Diodo a VDD .....	93
13.9. Diodo a Massa .....	93
13.10. FET a Canale P con Floating Gate .....	94
13.11. FET a Canale N con Floating Gate .....	95
13.12. Transmission gate .....	95
14. Altro .....	
14.1. Test case .....	96
15. Altro - Decorazioni .....	
15.1. Testo .....	96
15.2. Rettangolo .....	97
16. Altro - Generici .....	
16.1. Inizializzazione Generica .....	97
16.2. Codice .....	98
17. Altro - VHDL/Verilog .....	
17.1. Esterno .....	98
17.2. File Esterno .....	99
17.3. Controllo del Pin .....	100
18. Altro .....	
18.1. Alimentazione .....	101

18.2. Splitter Bidirezionale ..... 101

18.3. Reset ..... 102

18.4. Interruzione ..... 102

18.5. Fermata ..... 103

18.6. Temporizzazione Asincrona ..... 103

**E Libreria**

## A Generale

### 1. Digital

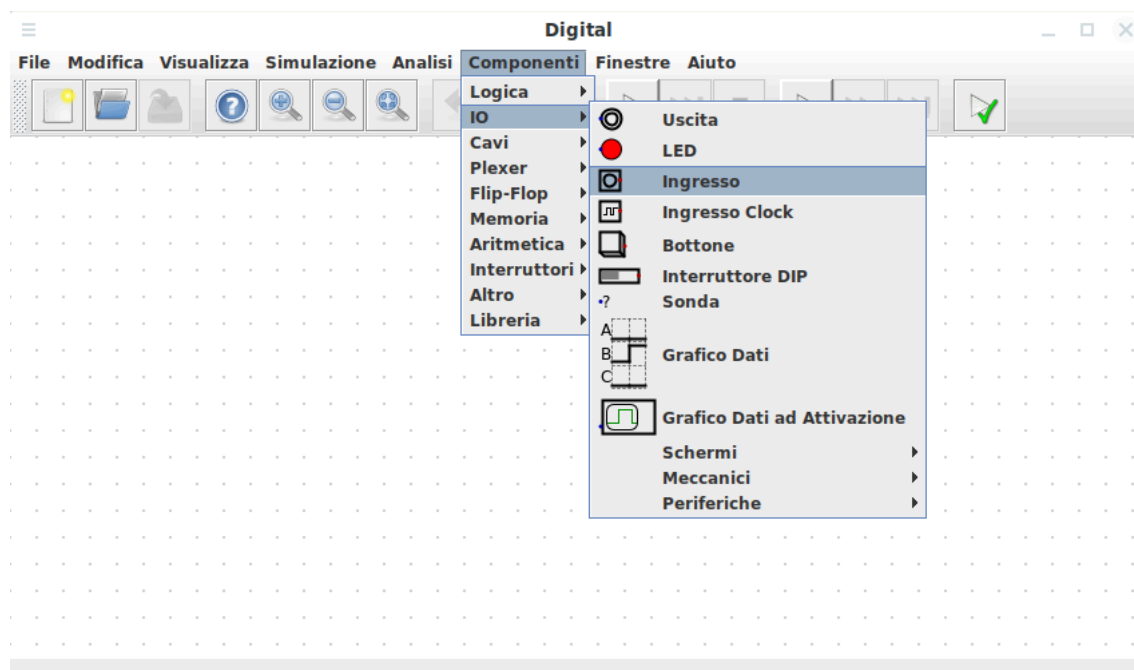
#### 1.1. Introduzione

Digital è un semplice simulatore utilizzato per simulare circuiti digitali. Le porte logiche sono connesse tra di loro da cavi e il comportamento complessivo del circuito può essere simulato. L'utente può interagire con la simulazione o premendo i bottoni o impostando i valori degli ingressi del circuito.

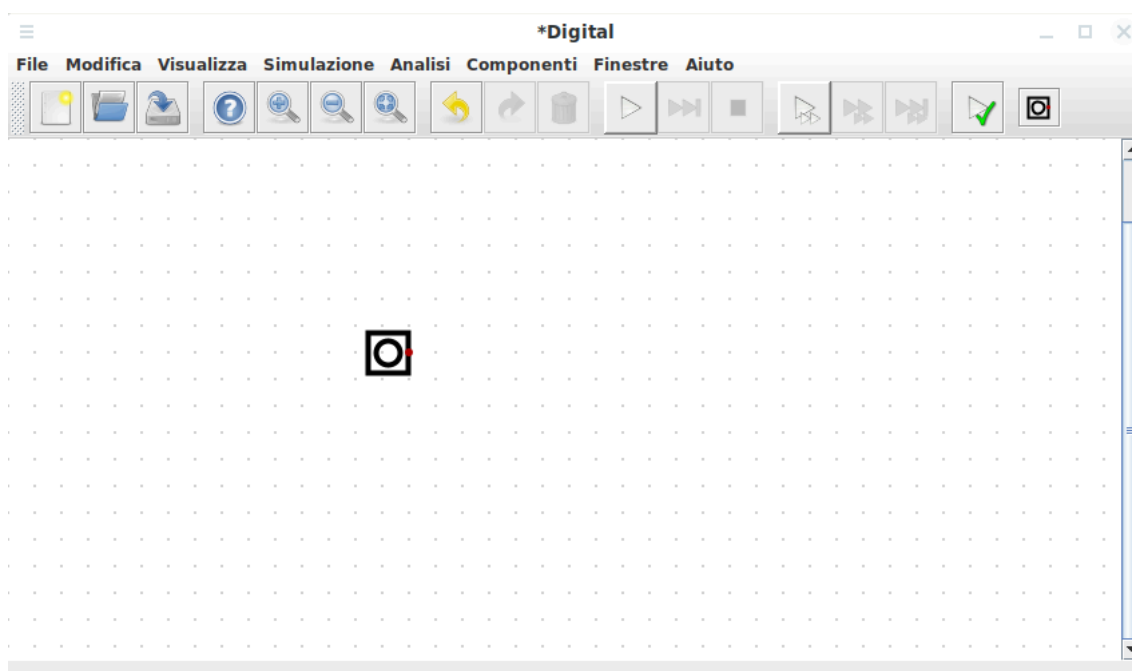
In questo modo, la maggior parte dei circuiti di base utilizzati nell'elettronica digitale può essere costruita e simulata. Nella cartella *examples*, gli utenti possono sfogliare gli esempi che includono anche un processore Harvard funzionale a ciclo singolo a 16 bit.

Il simulatore ha due modalità di funzionamento: modifica e simulazione. Nella modalità di modifica, si può apportare modifiche al circuito. Gli utenti possono aggiungere o connettere componenti. In questa modalità, la simulazione è disabilitata. La modalità simulazione invece è attivata premendo il tasto *Avvia* nella barra degli strumenti. Se ci sono errori nel circuito, un messaggio appropriato è mostrato e i componenti o cavi affetti da problemi sono evidenziati. Se il circuito non ha errori, la simulazione è abilitata. Ora si può interagire con l'esecuzione della simulazione. Nella modalità simulazione non è possibile modificare il circuito. Per far ciò, bisogna attivare la modalità di modifica ancora, fermando prima la simulazione.

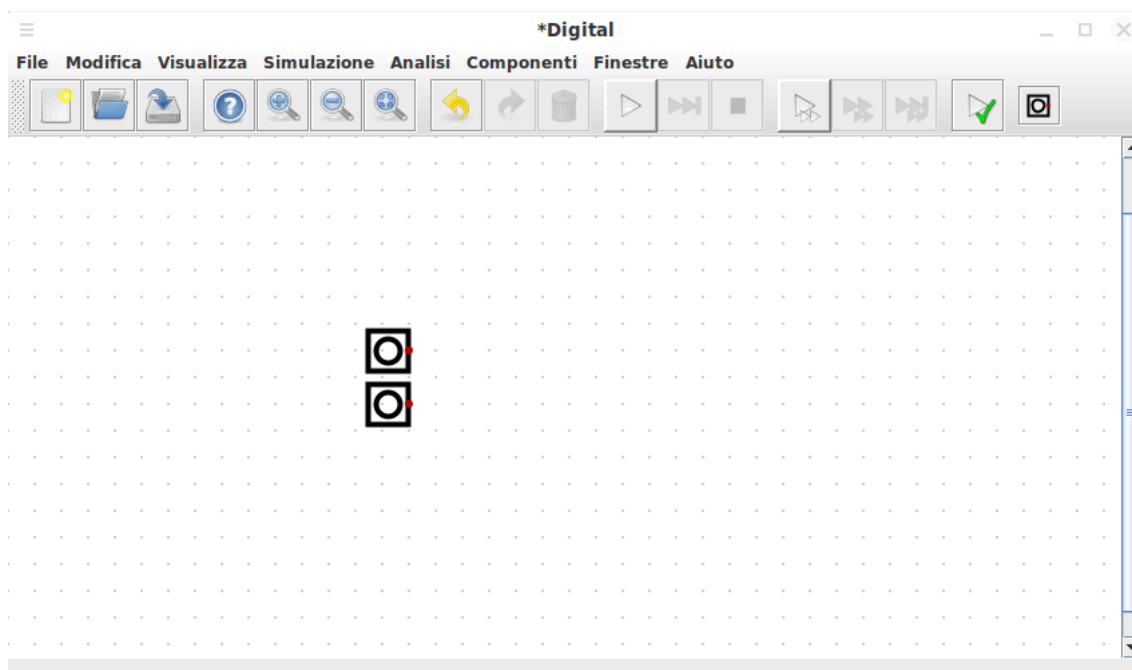
#### 1.2. Primi Passi



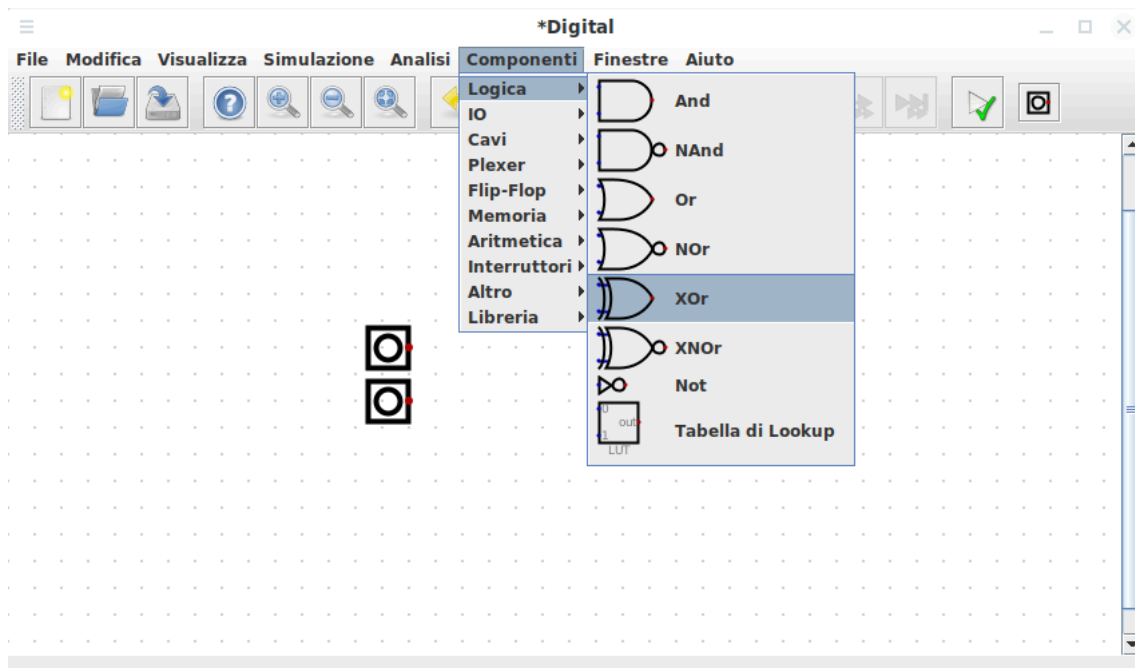
Come primo esempio, un circuito è costruito con una porta XOR. Dalla finestra principale, il menù *Componenti* consente di selezionarne vari. Possono essere poi piazzati nel pannello di disegno/progettazione. Questo processo può essere cancellato premendo il tasto ESC in qualsiasi momento. Inizia selezionando un componente di ingresso. Questo può essere controllato interattivamente più tardi utilizzando il mouse.



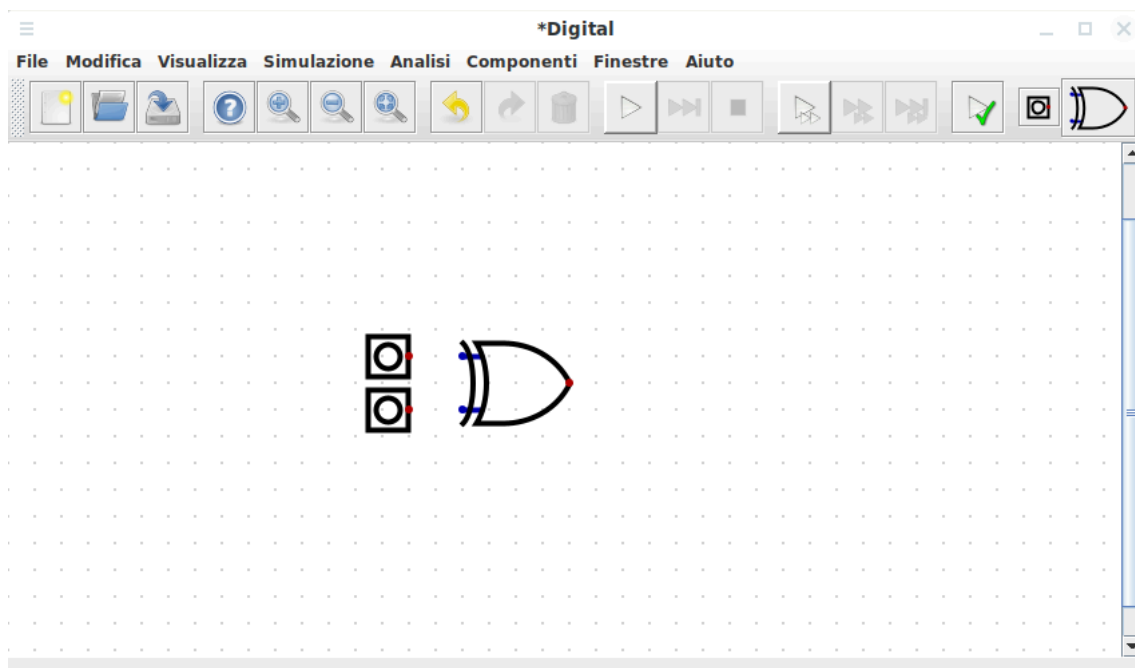
Dopo la selezione, il primo ingresso può essere posizionato nel pannello di progettazione. Il pallino rosso sul simbolo del componente di ingresso è un punto di connessione tra il componente ed il cavo, che sarà connesso più avanti. Il colore rosso indica un'uscita. Questo significa che la porta definisce un valore di segnale in uscita che può pilotare un cavo.



Nella stessa maniera, un secondo ingresso è aggiunto. Si ritiene buona pratica posizionarlo direttamente sotto il primo ingresso.

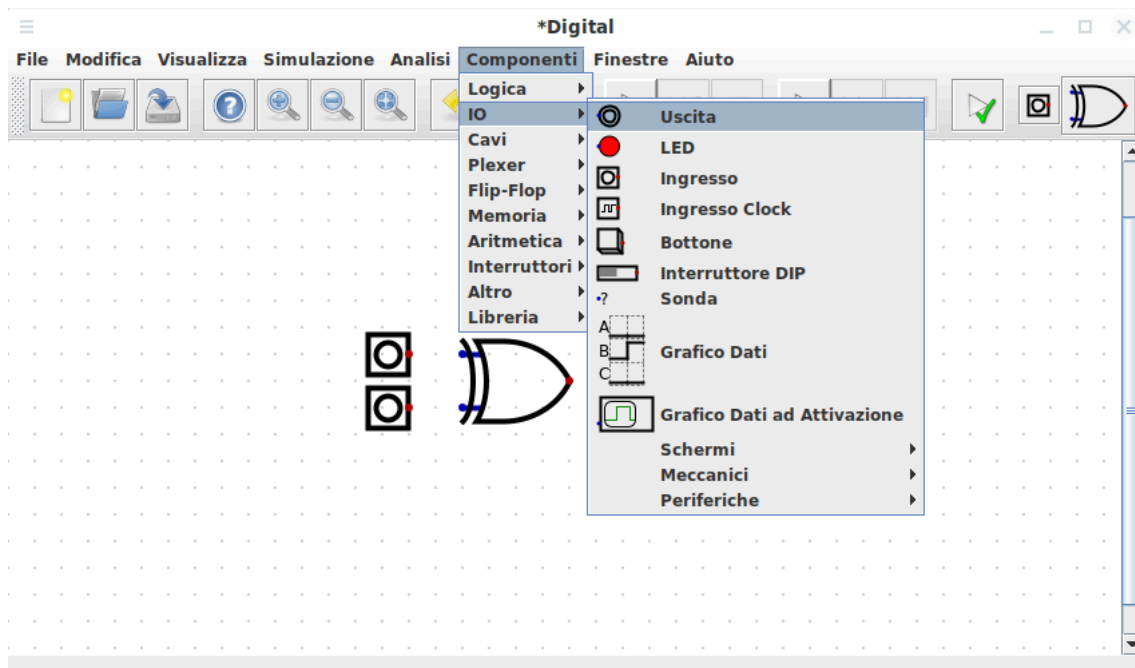


Dopo aver aggiunto tutti gli ingressi, la porta XOR è selezionata. Questa porta rappresenta la funzione logica.

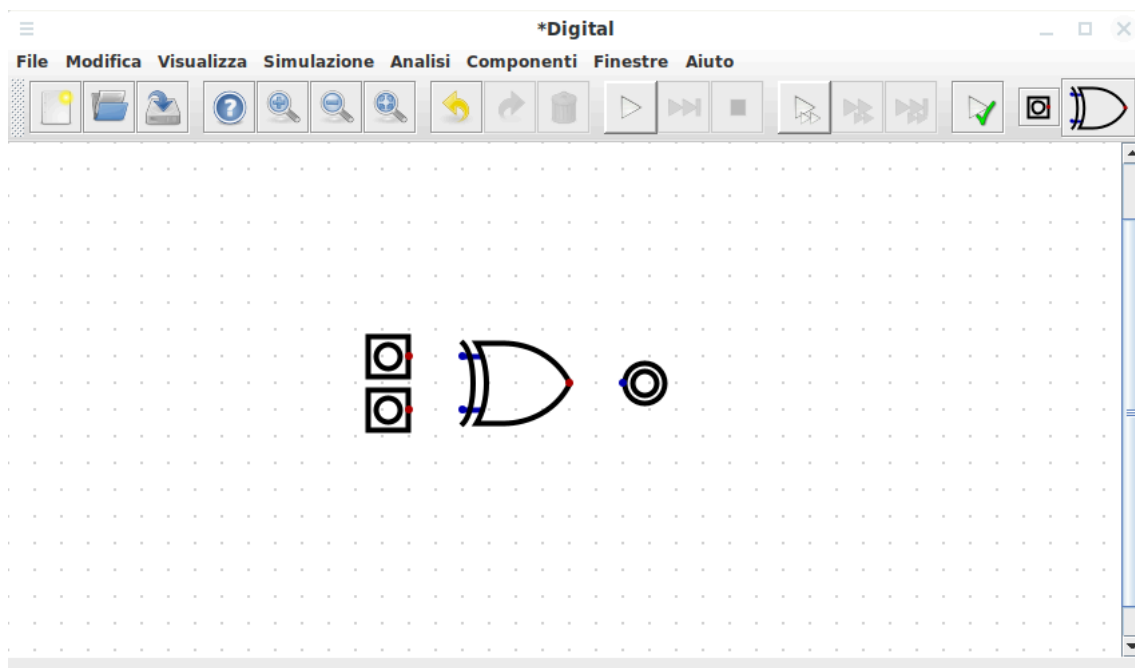


Anche questa porta può essere aggiunta al circuito. Si ritiene buona pratica posizionarla in maniera tale che tutto il cablaggio successivo è reso più semplice possibile. I pallini blu indicano i terminali di ingresso della porta.

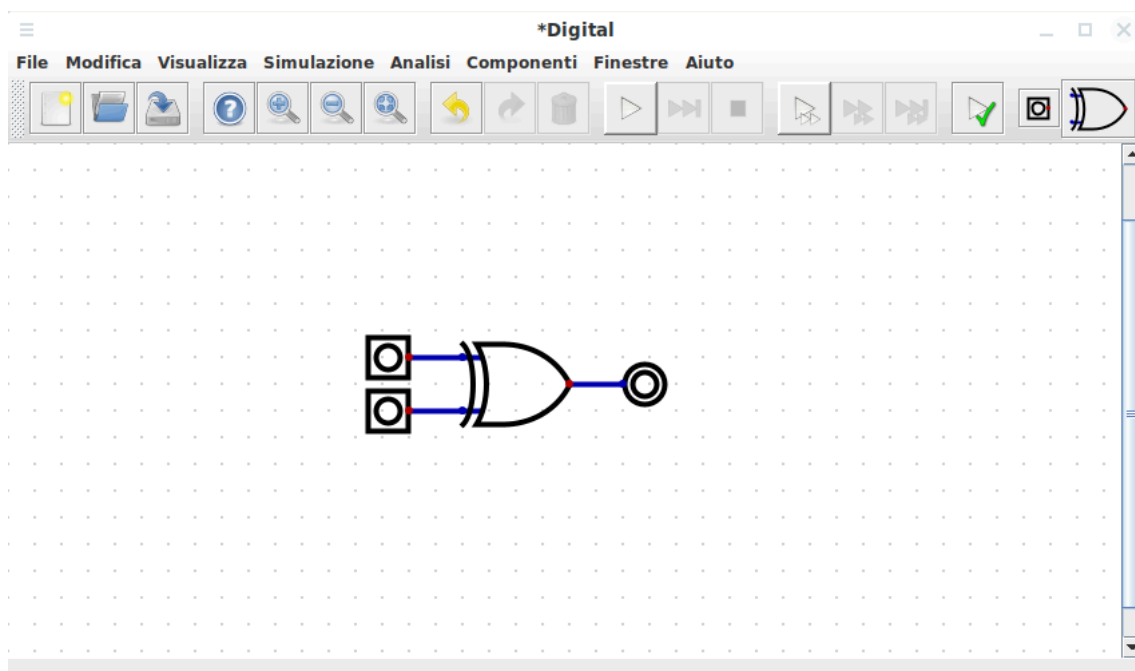




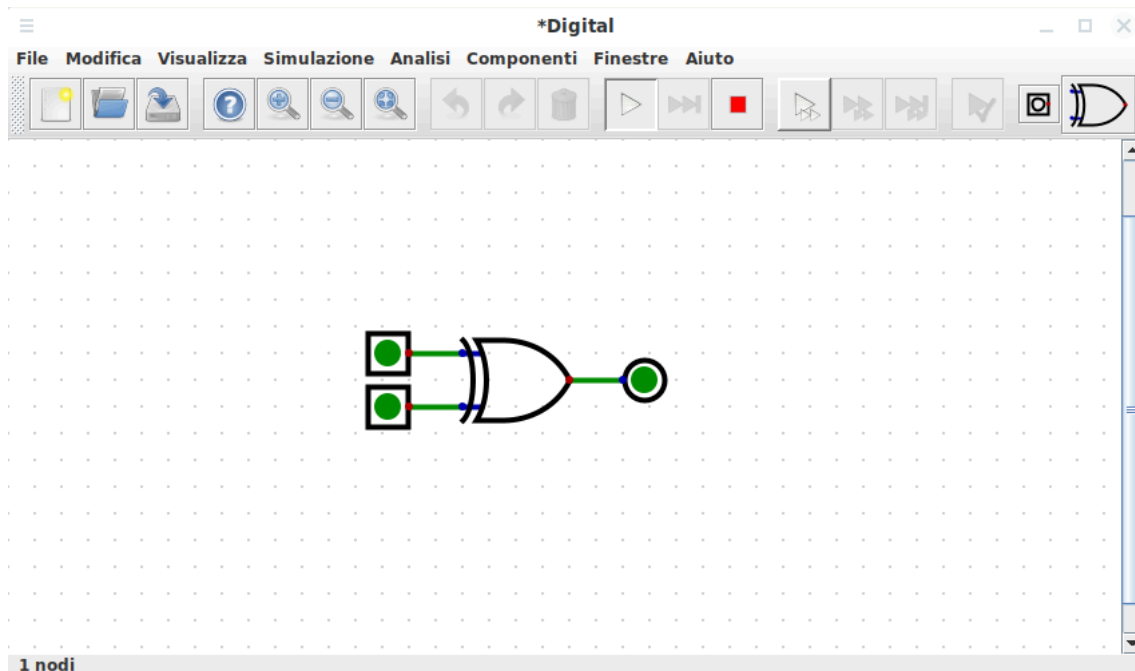
Ora, seleziona un'uscita che può essere utilizzata per visualizzare lo stato del segnale o per successivamente passare i segnali ad un circuito incorporato.



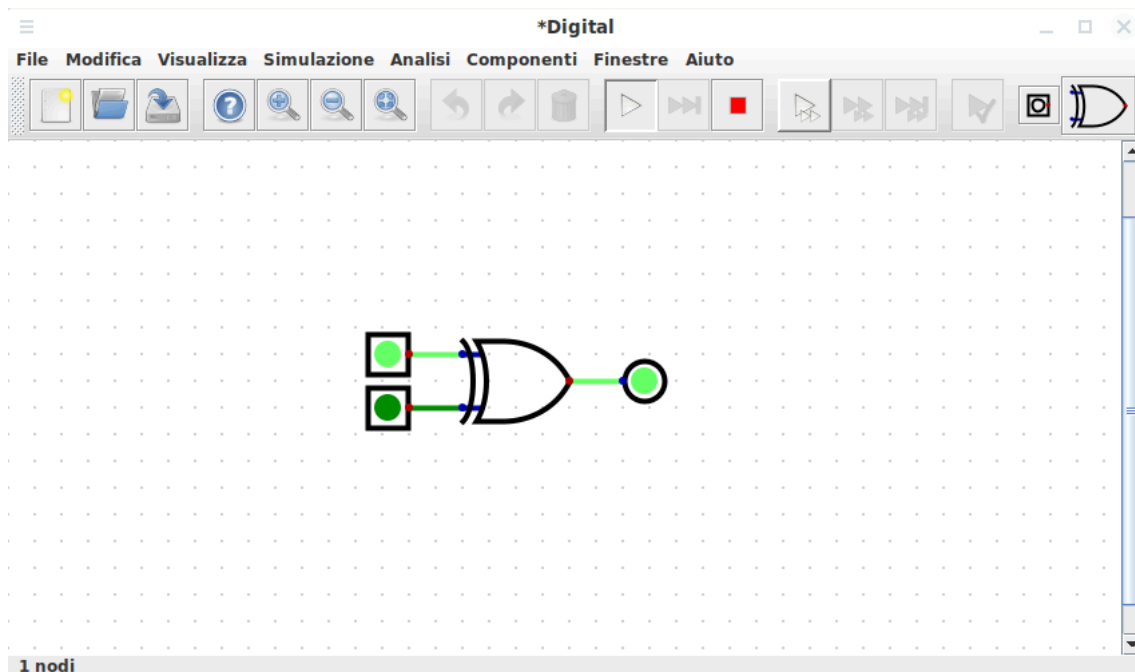
Questo è posizionato in un modo tale da essere cablato con facilità. L'uscita ha un pallino blu, che indica un terminale di ingresso. Qui puoi fornire il valore che è poi esportato.



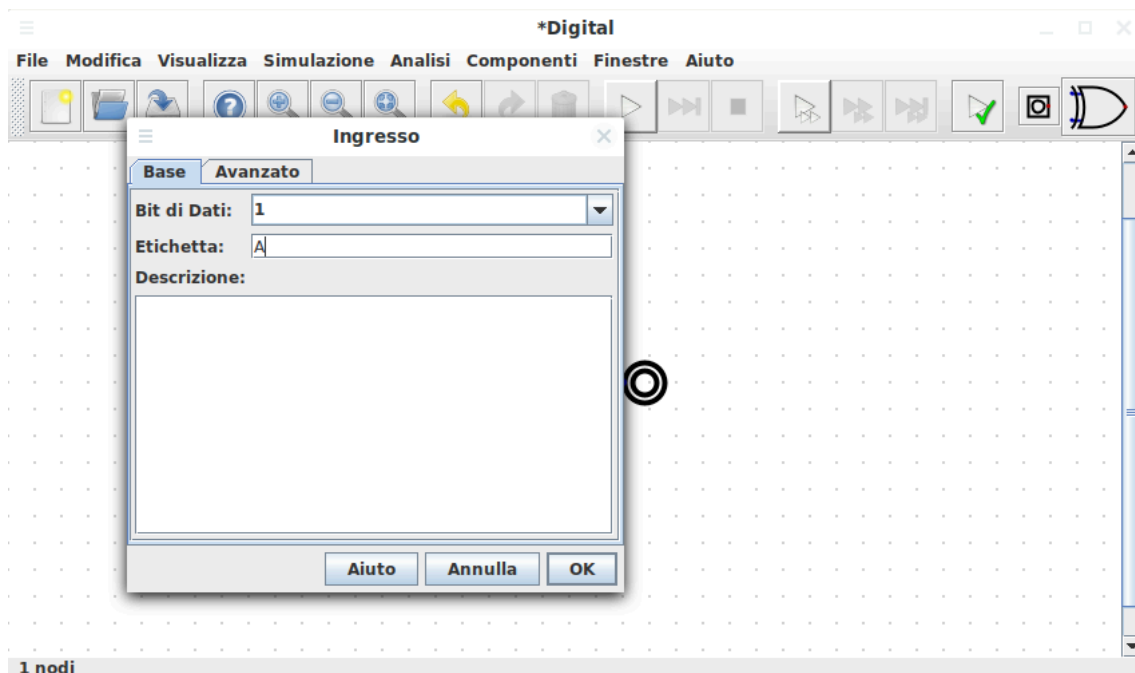
Dopo che tutti i componenti sono stati selezionati e posizionati, utilizza il mouse per creare una connessione tra i pallini blu e rosso. Assicurati che esattamente un pallino rosso è connesso ad un certo numero di pallini blu. Solo l'uso di uscite a tre stati rende possibile deviare da questa regola e connettere più pallini rossi. Se tutti i cavi sono stati posizionati, il circuito è completo.



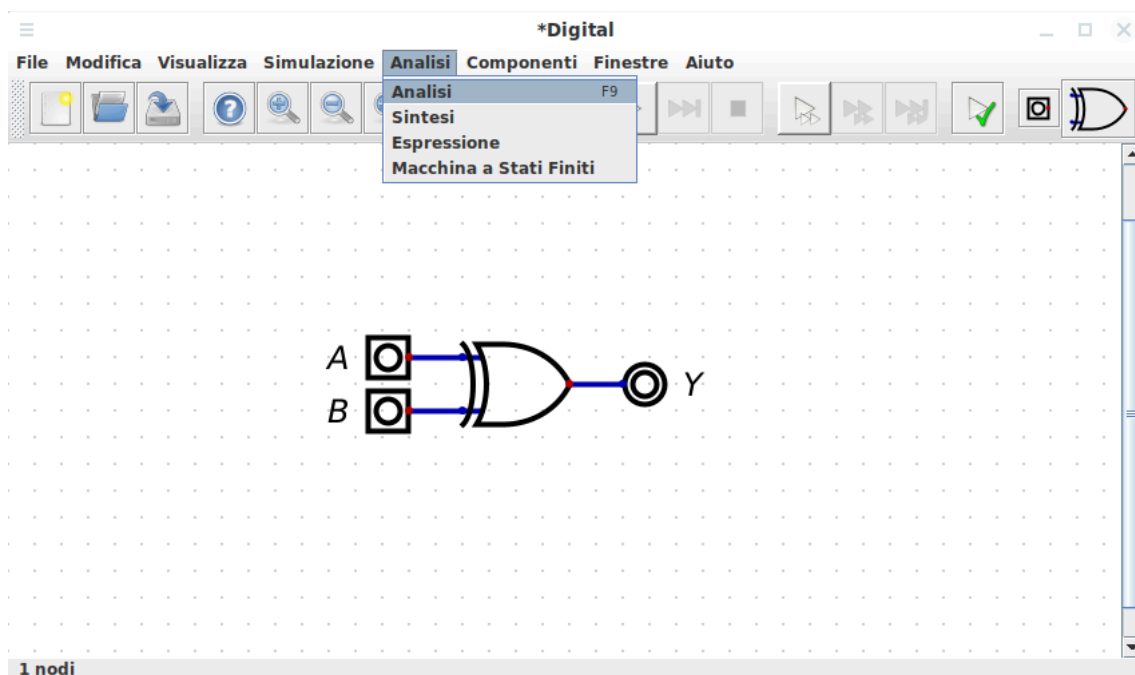
L'interazione con il circuito è possibile quando la simulazione è stata avviata. Questo si effettua premendo il bottone di avvio localizzato nella barra degli strumenti. Dopo aver avviato la simulazione, il colore dei cavi cambia e gli ingressi ed uscite sono ora calcolati. Verde chiaro indica un valore logico "1", mentre invece verde scuro indica un valore logico "0". Nella figura di sopra, tutti i cavi hanno un valore "0".



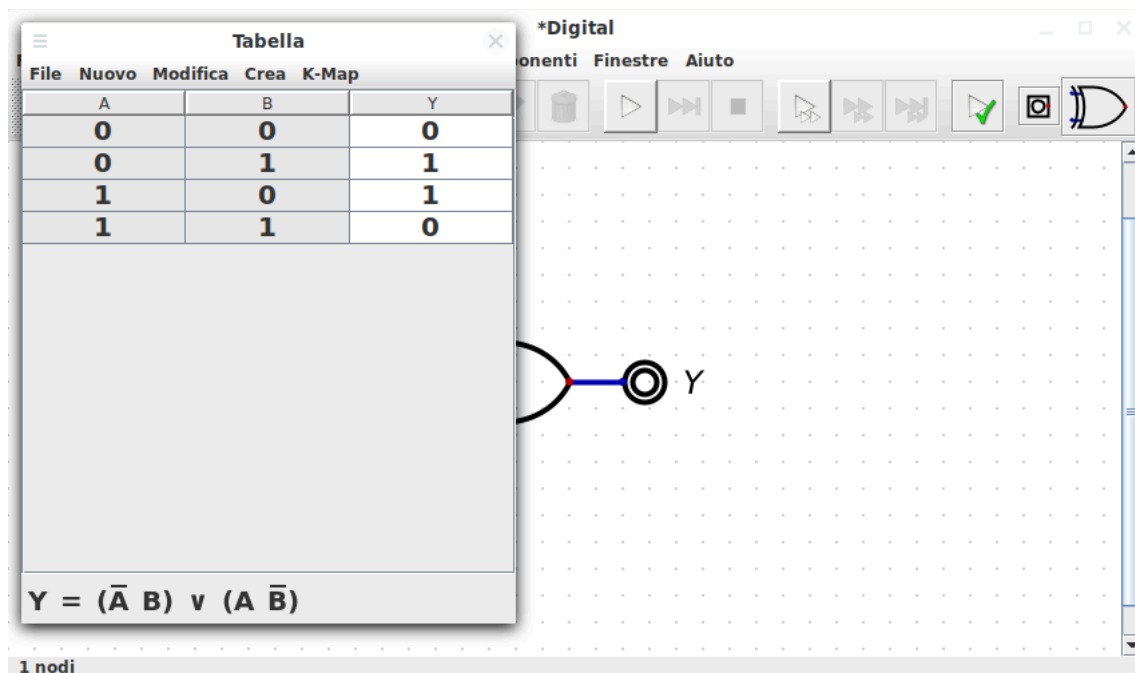
Premendo il bottone del mouse, il valore degli ingressi può essere invertito. Visto che la simulazione è ora attiva, l'uscita cambia a seconda degli stati attuali di ingresso. Il circuito si comporta come una porta XOR come ci si aspettava.



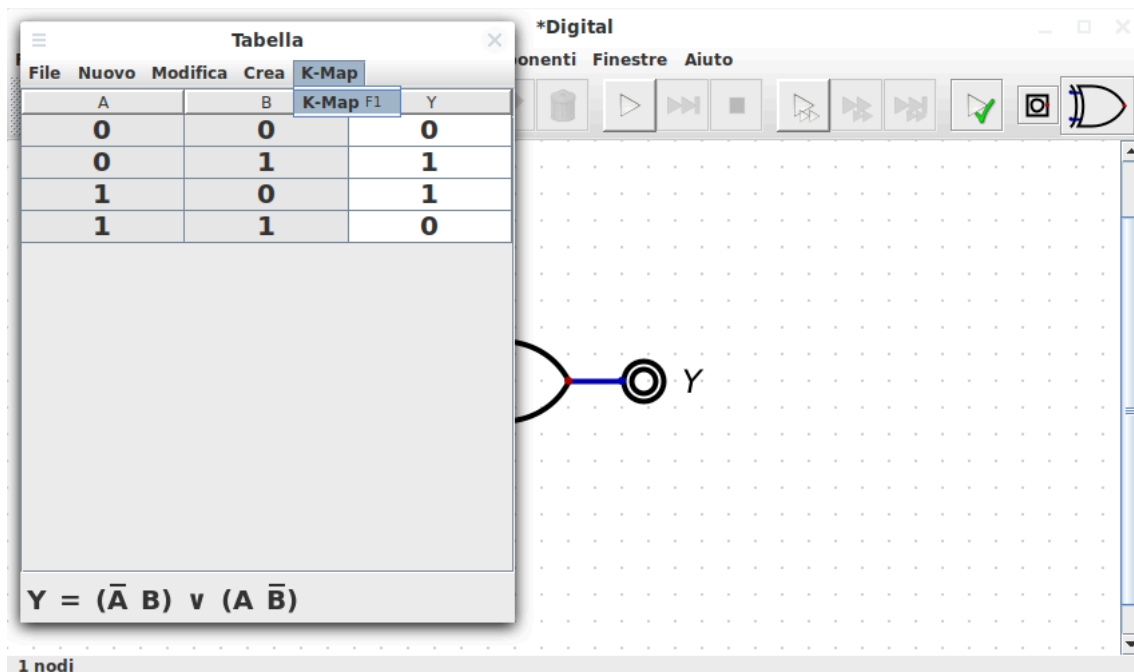
Per un ulteriore cambiamento del circuito, la simulazione va prima fermata. Il modo più facile di fare ciò è con il bottone di stop nella barra degli strumenti. Premendo il bottone del mouse su un componente con il tasto destro (control-click su macOS) apre una finestra di dialogo che mostra le proprietà del componente. L'etichetta "A" può essere definita per il primo ingresso tramite questo dialogo.



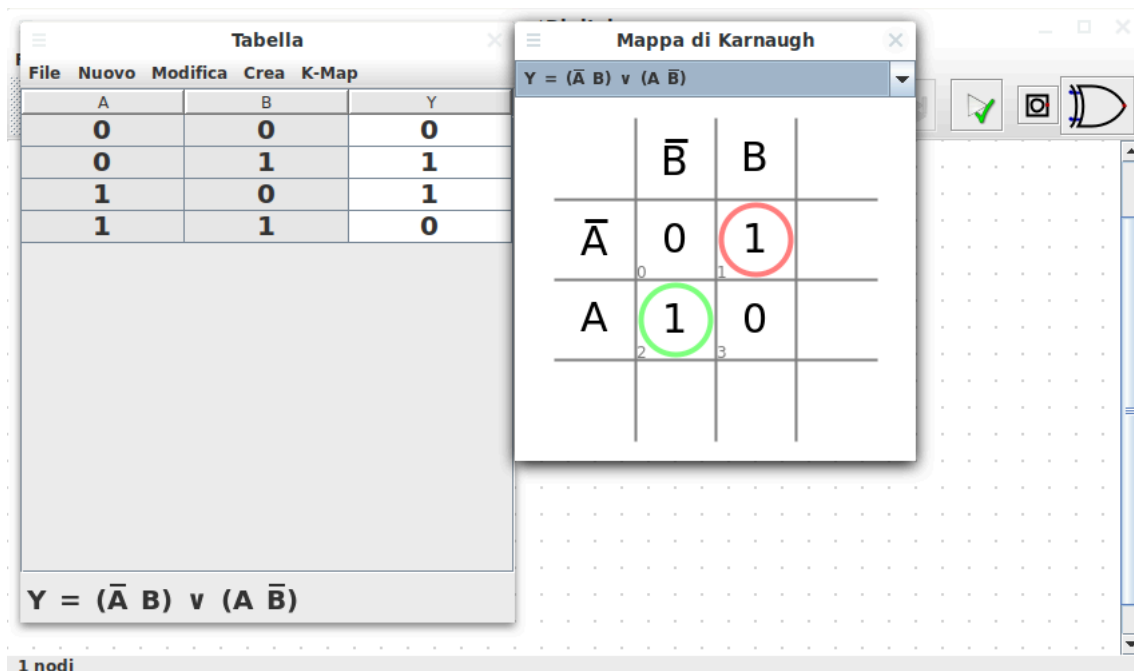
Nello stesso modo, le etichette per i rimanenti ingressi ed uscite possono essere definite. L'opzione nel menù *Analisi* contiene una funzionalità di *Analisi*. Questa funzione effettua un'analisi del circuito attuale. Questo rimane comunque possibile se tutti gli ingressi ed uscite sono stati etichettati appropriatamente.



La tabella di verità del circuito simulato appare in una nuova finestra. Sotto la tabella si possono trovare le espressioni algebriche associate al circuito. Se ci sono più possibili espressioni algebriche, una finestra separata si aprirà, mostrando tutte le possibili espressioni.



La finestra di dialogo contenente la tabella ha una opzione nel menù principale chiamata *K-Map*. Questa consente la visualizzazione della tabella di verità con una mappa di Karnaugh.



Nella parte superiore di questa finestra di dialogo c'è una lista che consente la selezione dell'espressione desiderata da visualizzare nella mappa di Karnaugh. In questo modo è possibile, per esempio, illustrare come alcune espressioni algebriche equivalenti possono risultare. In questo esempio però c'è una sola espressione minimizzata. La tabella di verità può anche essere modificata cliccando sulla mappa di Karnaugh.

### 1.3. Cavi

Tutti i componenti devono essere connessi tramite cavi. Non è possibile connettere due componenti posizionandoli direttamente l'uno vicino all'altro.

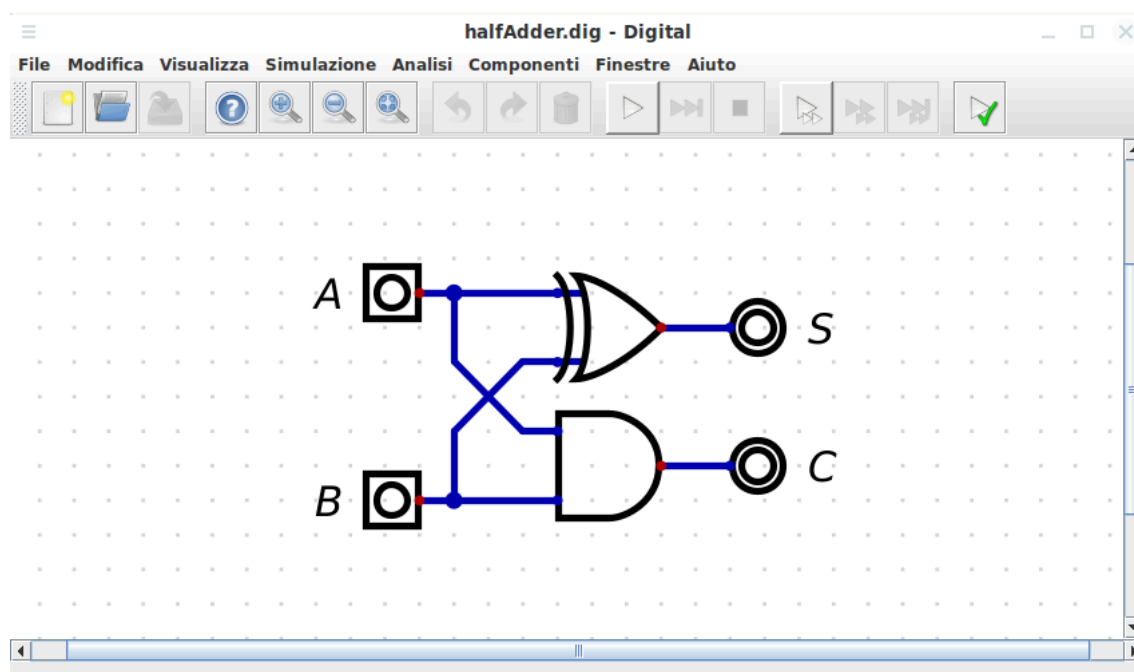
In aggiunta, ci sono soltanto connessioni tra un punto terminale di un cavo ed un componente. Se un pin di un componente è posizionato nel mezzo di un cavo, nessuna connessione è creata tra il componente ed il cavo. Il cavo quindi deve terminare su ogni pin al quale lo si vuole connettere. Anche se il componente tunnel è utilizzato, ci deve essere un cavo tra il pin ed il componente tunnel.

Il componente deve essere selezionato usando lo strumento di selezione rettangolare per spostarlo assieme ai cavi collegati. Per spostare un componente senza i cavi collegati, seleziona il componente con un click del mouse.

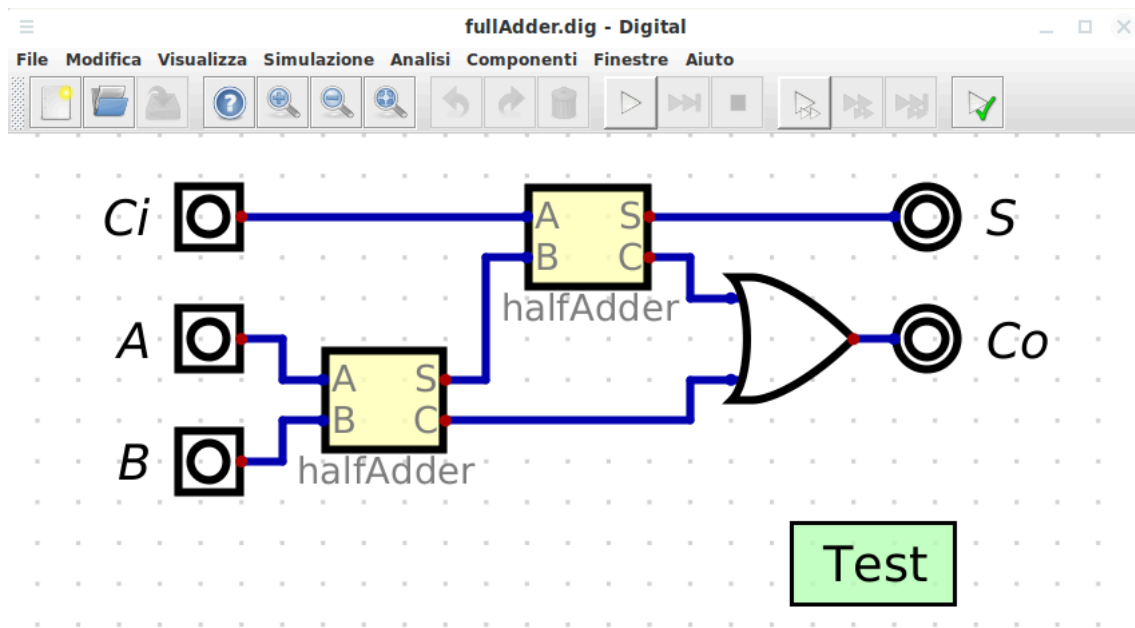
Con ctrl+click un singolo cavo può essere selezionato per spostarlo o eliminarlo. Se il tasto D è premuto mentre si disegna un cavo, un cavo diagonale può essere posizionato. Il tasto S invece consente la divisione di un segmento in due segmenti rispettivi.

### 1.4. Progettazione Gerarchica

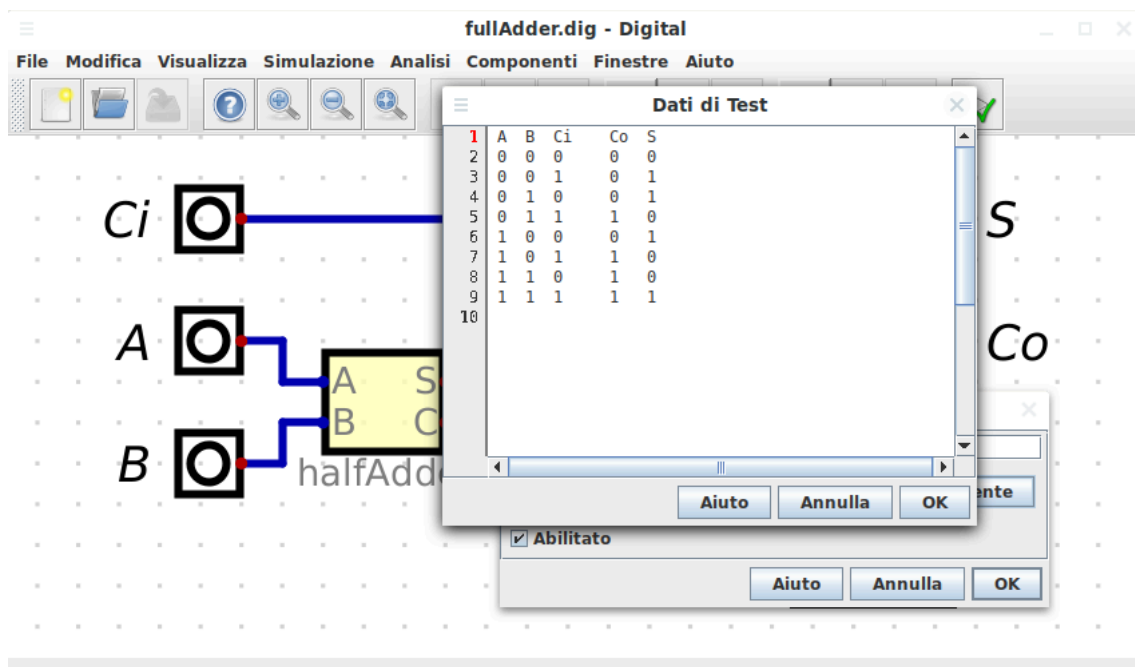
Se un circuito complesso viene costruito, questo può velocemente causare confusione. Per tenerne traccia, le differenti parti del circuito possono essere memorizzate in file separati. Questo meccanismo rende possibile utilizzare quella sezione separata come un sottocircuito, che è stato creato una sola volta ma può essere riutilizzato numerose volte in altri circuiti. Questo approccio offre anche il vantaggio che i file possono essere memorizzati indipendentemente l'uno rispetto all'altro in un sistema di controllo delle revisioni e quindi i cambiamenti possono essere tracciati.



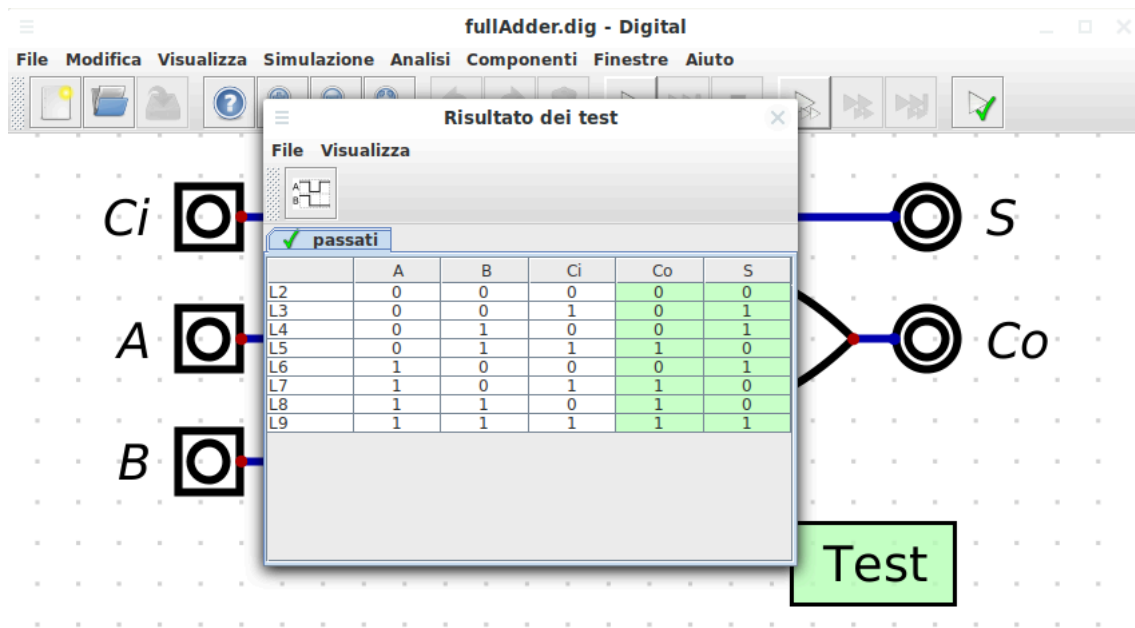
A titolo di esempio, si consideri un sommatore a 4 bit: prima si costruisce un semplice half-adder. Questo consiste in una porta XOR ed una AND. La somma dei due bit "A" e "B" è data nelle uscite "S" e "C". Il circuito è memorizzato nel file *halfAdder.dig*.



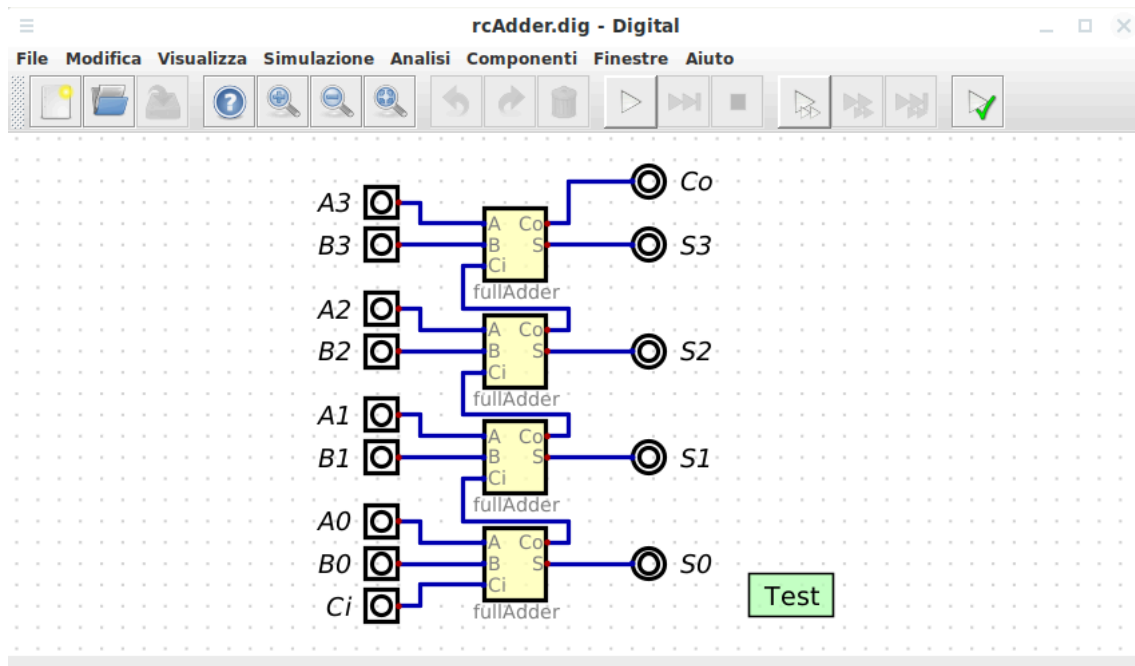
Da due half-adder, un full-adder può essere ora costruito. Per fare ciò, si crea un nuovo file e lo si salva come *fullAdder.dig* nella stessa cartella dove si trova l'*half-adder*. Successivamente, l'*half-adder* può essere aggiunto nel circuito tramite il menù *Componenti* → *Personalizzati*. L'ordine dei pin nel package dell'*half-adder* può essere cambiato dall'*half-adder* nel menù *Modifica* → *Ordina ingressi* oppure *Modifica* → *Ordina uscite*. Il sommatore aggiunge i tre bit "A", "B" e "Ci" e restituisce la somma nelle uscite "S" e "Co".



Per verificare la corretta funzionalità del sommatore, un test case dovrebbe essere aggiunto. In un test case, la tabella di verità viene specificata, che dovrebbe rispecchiare quella del circuito. In questo modo è automaticamente possibile verificare la corrispondenza.

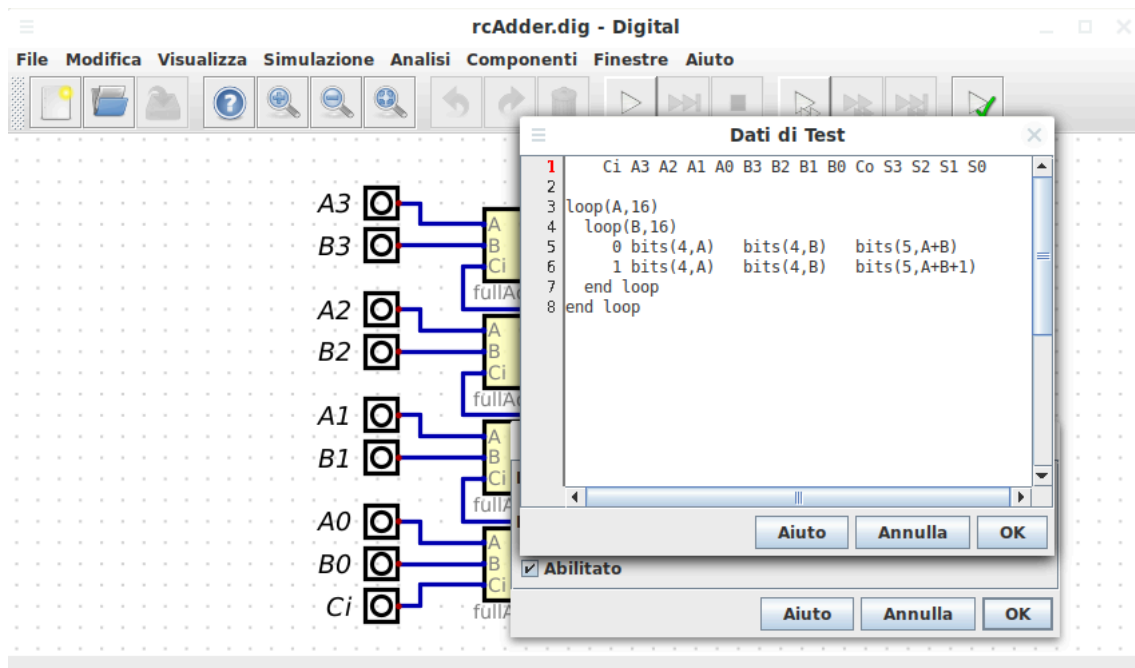


I test possono essere eseguiti nella schermata di modifica del test case oppure con il rispettivo bottone nella barra degli strumenti. Le celle della tabella evidenziate in verde indicano che le uscite del circuito corrispondono alla tabella di verità costruita dal circuito.



Ora i sommatore possono essere composti insieme per creare un ripple-adder. In questo caso, il riporto in uscita di una addizione è inviato come riporto in ingresso all'addizione successiva del bit un ordine più alto, come si usa fare nella somma su carta. Questo sommatore a 4 bit può essere testato. A questo scopo, un test case è stato inserito.





Questo test case effettua un test al 100%, che è possibile soltanto con circuiti relativamente semplici: tutte le 512 combinazioni degli ingressi sono applicate al circuito e il risultato viene controllato. La prima linea elenca i segnali di ingresso ed uscita. Sotto di essa, i valori di ingresso da applicare e i valori di uscita da controllare sono specificati in una riga, come in una tabella di verità. In questo esempio però, 512 righe sono richieste. Inserire queste righe sarebbe un processo tedioso e potrebbe portare ad errori. Risulta più facile ed affidabile generare automaticamente queste linee. A questo scopo, le variabili  $A$  e  $B$  sono entrambe fatte iterare da 0 a 15. I rispettivi valori sono poi assegnati agli ingressi " $A[n]$ " e " $B[n]$ ". Si controlla poi l'uscita del circuito corrispondente ad  $A+B$ . Si controlla anche se il bit di riporto è stato impostato, ovvero se  $A+B+1$  è risultato. I dettagli della sintassi del test case sono forniti dalla finestra di dialogo dell'aiuto.

Se un circuito è incorporato all'interno di un altro, soltanto il nome del file del sottocircuito è memorizzato nel circuito, non lo stesso circuito incorporato. Di conseguenza, durante la simulazione i file dei sottocircuiti devono essere localizzabili dal simulatore. Per evitare problemi di amministrazione dei percorsi importati, una insolita strategia è stata adottata.

Solo i nomi del file di un circuito incorporato sono memorizzati all'interno di un file di circuito, non il percorso completo. Se un file deve essere aperto, tutte le sottocartelle vengono esplorate alla ricerca di quel file. Se un file compatibile viene trovato, esso è importato. Questo processo dipende solamente dal nome del file da leggere, non dal suo percorso. Nella stessa maniera, un errore è generato se ci sono più file con lo stesso nome in diverse sottocartelle, visto che potrebbero nascere delle ambiguità.

Una struttura del progetto adeguata può essere quindi costruita così: il circuito principale è localizzato in una cartella separata. Tutti i circuiti da importare sono nella stessa cartella o sottocartelle. Tutti i circuiti devono avere nomi differenti, quindi non ci devono essere circuiti con lo stesso nome in sottocartelle diverse.

## 2. Simulazione

### 2.1. Ritardo di Propagazione

Durante la simulazione ogni porta logica ha un ritardo di propagazione. Ogni componente trovato nella libreria ha lo stesso ritardo di propagazione indipendentemente dalla sua complessità. La porta AND ha quindi lo stesso ritardo di un moltiplicatore. Le uniche eccezioni sono i diodi, interruttori e splitter che sono usati per creare bus dati. Questi componenti non hanno alcun ritardo di propagazione.

Se è necessario simulare una porta - per esempio il moltiplicatore - con un ritardo più lungo, una porta di ritardo deve essere inserita esattamente prima dell'uscita del moltiplicatore.

Se un circuito viene incluso in un altro circuito padre, il circuito incluso mantiene il suo comportamento temporale. Quindi se si include un circuito che ha un lungo tempo di propagazione perché i segnali di ingresso devono passare all'interno di molte porte prima di raggiungere l'uscita, questo comportamento è conservato dopo l'inclusione del circuito. Non ci sono ritardi aggiuntivi introdotti dall'inclusione di un circuito. Se non tutte le uscite del circuito hanno lo stesso ritardo di propagazione, allora questo lo si vede anche nel circuito incluso. In generale, includere un circuito in un altro si comporta nello stesso modo in cui si comporta un circuito con tutti gli stessi componenti interni aggiunti a mano.

## 3. Analisi

### 3.1. Analisi e Sintesi dei Circuiti

Un circuito può essere analizzato tramite la voce di menù *Analisi*. Una tabella di verità è generata per circuiti puramente combinatori. Questa tabella di verità può essere modificata a piacere. Un nuovo circuito può essere generato a partire da questa tabella di verità modificata.

In aggiunta a circuiti puramente combinatori, è anche possibile analizzare o generare circuiti sequenziali. Anziché una semplice tabella di verità, una tabella di transizione di stato è creata. Ogni flip-flop quindi appare nel lato degli ingressi e uscite della tabella di transizione degli stati. In questa tabella, nel lato destro si trovano gli stati successivi, a cui si effettuerà la transizione al prossimo segnale di clock. Lo stato successivo dipende dallo stato corrente del flip-flop, come specificato nel lato sinistro della tabella. Per rendere possibile l'analisi, ciascun flip-flop deve avere un nome.

La seguente convenzione di nomenclatura si applica: lo stato successivo di un bit nel lato destro della tabella è indicato come "n+1". Il corrispondente stato corrente è indicato con un suffisso "n". Se c'è una variabile di stato "A", "An" indica lo stato corrente e "An+1" indica lo stato successivo. Se, nella tabella di verità sul lato destro e sinistro, sono presenti dei segnali che corrispondono a questa descrizione, si assume che la tabella di verità sia una tabella di transizione e quindi un circuito sequenziale è generato al posto di uno combinatorio.

Si noti che il circuito da analizzare potrebbe contenere elementi puramente combinatori in aggiunta ai flip-flop JK e D. Se un flip-flop è, per esempio, fatto da porte NOR, questo circuito non è riconosciuto come un flip-flop e quindi non è possibile analizzarlo.

### 3.2. Espressione

Tramite la voce di menù *Espressione* è possibile inserire una funzione booleana dalla quale generare poi un circuito.

### 3.3. Grafici di Stato

Un editor per i grafici di stato è disponibile tramite la voce di menù *Macchina a Stati Finiti*. Consente la creazione grafica di macchine a stati disegnando stati e transizioni. Di conseguenza, uscite differenti possono essere impostate in stati differenti. Fornendo le transizioni con condizioni, i segnali di ingresso possono essere generati. Impostando i valori di uscita nelle transizioni, un automa di Mealy può essere definito.

La macchina a stati definita in questo modo può essere quindi automaticamente trasferita in una tabella di transizione dalla quale, in un passo successivo, un circuito che implementa la macchina a stati iniziale può essere generato. Se la simulazione di questo circuito è avviata, lo stato corrente può anche essere visualizzato dal grafico degli stati.

## 4. Hardware

### 4.1. GAL16v8 e GAL22v10

Nel menù di generazione del circuito nella tabella di verità ci sono anche funzioni per generare i cosiddetti file JEDEC. Questo è un formato speciale che descrive la mappa dei fusibili di un PLD. Questo file JEDEC può essere scritto in un corrispondente PLD utilizzando un programmatore speciale. Al momento, circuiti del tipo *GAL16v8* e *GAL22v10* o compatibili con dispositivi a mappa di fusibili sono supportati.

### 4.2. ATF150xAS

I circuiti nella famiglia *ATF150x* sono semplici CPLD con al massimo 128 macrocelle. Sono disponibili in un package PLCC, che li rende utili per esercizi di laboratorio: se un circuito integrato viene distrutto durante gli esercizi, può essere semplicemente sostituito. In aggiunta, con *ATDH1150USB* un programmatore a basso costo e semplice da utilizzare è disponibile. Questo programmatore è in grado di programmare i chip *ATF150x* nel sistema utilizzando una interfaccia JTAG. Una scheda di valutazione compatibile (*ATF15XX-DK3-U*) è disponibile. Il software *ATMISP*, che è disponibile nel sito di ATMEL/Microchip è richiesto per programmare i circuiti.

Sfortunatamente, i dettagli della mappa a fusibili non sono pubblicamente disponibili quindi nessun fitter per questo circuito può essere integrato in Digital, come è possibile con circuiti *GAL16v8* and *GAL22v10*.

Quindi, i fitter *fit150[x].exe* forniti da ATMEL devono essere utilizzati. Questi programmi creano un file *JEDEC* da un file *TT2* compatibile, che può essere quindi programmato sul chip. Digital avvia il fitter automaticamente ogni volta che un file *TT2* viene creato. A questo scopo, il percorso di *fit150[n].exe* deve essere specificato nelle impostazioni. Il file *JEDEC* creato può poi essere aperto e programmato direttamente con *ATMISP*.

Per motivi legali, il fitter *fit1502.exe* non può essere distribuito con Digital. Può però essere trovato nella cartella *WinCupl\Fitters* dopo aver installato *WinCupl*. *WinCupl* è disponibile dal sito ATMEL/Microchip. Sui sistemi Linux, i fitter possono essere eseguiti da Digital se *wine* è installato.

### 4.3. Esportare a VHDL o Verilog

Un circuito può essere esportato in VHDL o Verilog. Un file viene generato e contiene la descrizione completa del circuito. Il codice VHDL generato è stato testato con Xilinx Vivado e il

simulatore VHDL open-source ghdl. Il codice Verilog è invece stato testato con il simulatore Verilog Icarus Verilog.

Se un circuito contiene dei test case, i dati di test possono essere usati per generare una test bench HDL. Questa può essere usata per verificare la corretta funzionalità del circuito nella simulazione HDL.

File aggiuntivi che sono richiesti da schede specifiche possono essere creati. Attualmente, soltanto la scheda BASYS3 e le schede Mimas e Mimas V2 sono supportate. Un file di vincoli è creato, che contiene gli assegnamenti dei pin. La descrizione dei pin può essere trovata nel datasheet della scheda e deve essere inserito come un numero di pin per gli ingressi e le uscite.

Per una scheda BASYS3, se la frequenza di clock del circuito è bassa, un divisore di frequenza è integrato all'interno del codice HDL per dividere il clock della scheda correttamente. Se la frequenza selezionata supera i 4.7MHz, l'unità MMCM dell'Artix-7 è utilizzata per la generazione del clock. Questo fa sì che le risorse FPGA fornite per la distribuzione del clock siano utilizzate. Questo consente al processore di esempio incluso di eseguire a 20MHz; se si riesce a farlo senza il moltiplicatore si raggiungono anche i 30MHz.

Se un circuito viene fatto eseguire su una scheda BASYS3, un nuovo progetto può essere creato in Vivado. Il file VHDL generato e i file dei vincoli devono essere aggiunti al progetto. Una volta che il progetto è stato creato, il bitstream può essere generato e l'hardware manager può essere utilizzato per programmare la scheda.

Per poter creare il file dei vincoli richiesto in aggiunta al file HDL, la scheda corrispondente deve essere configurata nelle impostazioni. Nel campo "Configurazione della Toolchain", il file XML corrispondente può essere selezionato. Le configurazioni disponibili possono essere trovate nella cartella *examples/hdl* e hanno l'estensione file *.config*. Se la configurazione viene integrata con successo, un menù aggiuntivo compare, che rende disponibili le funzionalità specifiche della scheda.

## 5. Figure Personalizzate

Anche se Digital ha alcune opzioni che modificano l'aspetto di un circuito quando viene incorporato in un altro, in alcuni casi è utile utilizzare una figura specifica per un particolare sottocircuito. Un esempio è la rappresentazione dell'ALU nel processore incluso negli esempi. Questo capitolo spiega come definire queste figure speciali per un circuito.

Digital non fornisce un editor per creare una figura. Anzi, una piccola digressione è richiesta per capire come creare le figure del circuito: prima di tutto, il circuito che deve essere rappresentato da una figura particolare deve essere aperto. Successivamente un template SVG viene creato per questo circuito. Nel template, il circuito è rappresentato da un semplice rettangolo. Questo contiene anche tutti i pin del circuito, rappresentati dai pallini blu (ingressi) e da quelli rossi (uscite). Per vedere quale pallino appartiene a quale pin, bisogna guardare all'ID del pallino nelle proprietà dell'oggetto. Questo ID ha la forma *pin:[nome]* o *pin+:[nome]*. Nella seconda variante, il pin è fornito con un'etichetta se reimportato all'interno di Digital. Se non si vuole tale etichetta, il segno + può essere rimosso.

Il file SVG può ora essere modificato. Il programma open-source più utile e compatibile è Inkscape che è disponibile in forma gratuita. I pin possono essere spostati liberamente, ma sono mossi nel punto della griglia successivo durante la reimportazione.

Se un file SVG preesistente deve essere utilizzato, è più facile aprire il template creato e incollare la grafica esistente all'interno del template con copia-incolla.

Se il file è stato salvato, può essere importato con Digital. Il file viene letto e tutte le informazioni necessarie sono estratte e memorizzate nel circuito. Per un uso successivo del circuito, il file SVG non è più richiesto.

Una importante nota: SVG è un format molto potente e flessibile. Questo può essere utilizzato per descrivere grafiche estremamente complesse. L'importazione in Digital non sarà in grado di riconoscere tutti i tipi di grafica SVG senza errori. Se un file non può essere importato, è utile fare un po' di sperimentazione per capire quale componente grafico possa causare dei problemi.

## 6. Circuiti Generici

Si consideri un circuito che vada incorporato all'interno di un altro ed utilizzato in vari modi. Per esempio, si può immaginare un particolare contatore che è necessario abbia differente numero di bit. Se uno deve creare un circuito per 4, 5 e 6 bit ciascuno, la manutenzione del circuito diventa molto difficile nel futuro mano a mano che nuove configurazioni sono aggiunte.

Per prevenire ciò, un sottocircuito generico può essere parametrizzato e creato. A questo scopo, la spunta "Il circuito è generico" deve essere impostata nelle impostazioni specifiche del circuito stesso. Come conseguenza, ogni finestra di dialogo per le proprietà di un componente del circuito conterrà il campo aggiuntivo "parametrizzazione generica". In questo campo si può inserire del codice che cambia i parametri del componente. Ogni parametro ha un nome e può essere modificato come un attributo del campo *this*. Ogni parametro può essere individuato nella finestra di dialogo di aiuto per quanto riguarda il componente. Se si vuole cambiare la larghezza di bit di un sommatore, per esempio, la linea *this.Bits=1*; può essere usata.

In questo modo, comunque, non è ancora possibile creare un circuito che può essere parametrizzato. Bisogna necessariamente accedere ai parametri che sono impostati quando il circuito viene usato. Questo è fatto tramite il campo "args". Se si vuole impostare la larghezza di bit dall'esterno, si può scrivere: *this.Bits=args.bitWidth*;. Il nome dell'argomento - qui *bitWidth* - è arbitrario. Se un sottocircuito è utilizzato, questo argomento va impostato.

Se il circuito è utilizzato e il dialogo dei parametri del circuito incorporato viene aperto, questo ha anche il campo "parametrizzazione generica". Qui la larghezza di bit da utilizzare può essere impostata con l'istruzione *bitWidth:=5*;

Non è possibile avviare direttamente un circuito generico visto che gli argomenti richiesti sono mancanti; devono essere specificati nel circuito incorporante. Questi argomenti mancanti porterebbero ad errori. Per semplificare la verifica del circuito quindi, il componente *Inizializzazione Generica* può essere aggiunto al circuito. In questo componente si possono impostare gli argomenti generici che dovrebbero provenire dal circuito incorporante. In questo modo, un circuito generico può essere simulato direttamente. Se il circuito è incorporato, il componente è ignorato. Questo è necessario soltanto per l'avvio diretto della simulazione.

Sotto determinate circostanze, potrebbe essere utile non solo cambiare gli attributi dei componenti di un circuito ma anche aggiungere componenti totalmente nuovi e cavi a seconda degli argomenti ricevuti. Il componente *Codice* può essere usato per questo scopo. Se aggiunto al circuito, il codice contenuto sarà eseguito durante l'avvio della simulazione. Qui, un cavo può essere aggiunto con la funzione *addWire([x1],[y1],[x2],[y2])*; con *addComponent([nome],[x],[y])* invece si aggiunge un nuovo componente *[nome]* nella posizione specificata. Il valore di ritorno di questa ultima funzione consente di impostare i parametri del componente.

Il circuito di esempio *examples/generic/modify/Conway/GenericConway.dig* mostra come un circuito complesso possa essere assemblato in questo modo.

Un altro modo di creare un circuito è la ricorsione: è possibile, a seconda degli argomenti, rimpiazzare un circuito con un altro. A questo scopo la funzione `setCircuit([nome])` è disponibile. Se chiamata nella parte di definizione di un sottocircuito, il circuito da inserire può essere sostituito con un altro circuito. Questo consente la definizione ricorsiva di un circuito. Come negli altri linguaggi di programmazione, una condizione di fine della ricorsione deve essere specificata.

La cartella *examples/generic* contiene un esempio di un contatore Gray code dove il numero di bit può essere configurato. Qui questo contatore è istruito all'aggiunta ricorsiva di più bit ad un circuito iniziale fino a quando il numero richiesto di bit del contatore è raggiunto.

## 7. Verifica Controllata da Script

Se gli studenti devono completare esercizi con Digital, potrebbe essere utile verificare i circuiti consegnati dagli studenti tramite un processo automatico. Per effettuare questa verifica, Digital può essere avviato dalla linea di comando. La chiamata funziona in questo modo:

```
java -cp Digital.jar CLI test [file da verificare] [-tests [file op-  
zionale con i test case]]
```

Se solo il file con il test è specificato, i test case contenuti in esso sono eseguiti. In questo modo, i test case creati dagli studenti stessi possono essere verificati.

Se un secondo file è specificato, i test case sono presi da questo secondo file e vengono controllati rispetto ai test case del file del circuito. Il secondo file contiene quindi di solito una soluzione di esempio i cui test case sono completi e corretti. Il circuito contenuto nel secondo file è ignorato. Soltanto i test case vengono presi da esso.

Per verificare un circuito con una soluzione di esempio, il nome dei segnali di ingresso e uscita devono corrispondere in entrambi i circuiti.

## 8. FAQ

### Come spostare un cavo?

Seleziona uno dei terminali con la selezione rettangolare. Sposta poi il punto utilizzando il mouse. Puoi anche selezionare un cavo con `ctrl+click`.

### Come eliminare un cavo?

Seleziona uno dei terminali e premi `DEL` o clicca sul cestino. Puoi anche selezionare un cavo con `ctrl+click`.

### Come spostare un componente inclusi tutti i cavi connessi?

Seleziona il componente con la selezione rettangolare. Questa selezione deve includere l'intero componente. Sposta ora il componente con il mouse e i cavi saranno anch'essi spostati.

### C'è un componente non connesso da un cavo, anche se i pin sono sul cavo.

Un pin è connesso al cavo solo se il cavo ha un terminale al pin.

### Se il nome dei pin in un circuito sono lunghi, i nomi non sono più leggibili quando il circuito diventa incorporato. Cosa posso fare?

La larghezza del blocco può essere aumentata con la voce del menù *Modifica* → *Impostazioni specifiche del circuito*.

**I pin nel circuito non hanno un ordine ottimale. Come posso cambiarlo?**

La sequenza può essere cambiata utilizzando la voce del menù *Modifica* → *Ordina ingressi* o *Modifica* → *Ordina uscite*.

**Quando la simulazione è avviata, un cavo diventa grigio. Che cosa significa?**

I colori verde chiaro e scuro indicano gli stati alto e basso. Il cavo grigio indica alta impedenza.

**Ho una tabella di verità. Come calcolo le equazioni booleane minimizzate?**

Nel menù *Analisi* seleziona la voce *Sintesi*. Inserisci quindi la tabella di verità. Nella parte in basso della finestra puoi trovare le equazioni booleane. Se inserisci più di una variabile dipendente, una nuova finestra si apre per mostrare tutte le equazioni.

**Ho inserito una tabella di verità, ma ci sono più equazioni mostrate. Quale di esse è corretta?**

Minimizzare un'equazione booleana può risultare in molte equazioni che descrivono la stessa funzione. Digital le mostra tutte visto che creano la stessa tabella di verità. Ci potrebbero essere delle differenze a seconda dei "don't care" nella tabella di verità.

**Ho una tabella di verità. Come creo il circuito corrispondente?**

Nel menù *Analisi* seleziona la voce *Sintesi*. Inserisci quindi la tabella di verità. Puoi modificare la tabella usando i menù *Nuovo* o *Modifica*. Nel menù *Crea*, puoi creare un circuito utilizzando la voce *Circuito*.

**Come modificare un nome di un segnale nella tabella di verità?**

Click destro nell'intestazione della tabella per modificare il nome.

**Ho un'equazione booleana. Come creo il circuito corrispondente?**

Nel menù *Analisi* seleziona la voce *Espressione*. Inserisci poi l'equazione.

**Come creare una tabella di verità da un'espressione booleana?**

Nel menù *Analisi* seleziona la voce *Espressioni*. Inserisci poi l'espressione. Crea quindi il circuito e nel menù *Analisi* usa la voce *Analisi* per creare la tabella di verità.

**Come creare un file JEDEC dal circuito dato?**

Nel menù *Analisi* seleziona la voce *Analisi*. Successivamente nel menù *Crea* nella nuova finestra seleziona il dispositivo corretto nel sottomenù *Dispositivo*.

**Quando creo un file JEDEC, come faccio ad assegnare un numero di pin ad un certo segnale?**

Nei corrispondenti ingressi ed uscite puoi inserire un numero di pin nella finestra di dialogo delle impostazioni del pin.

**Ho creato un file JEDEC. Come faccio a programmarlo in un GAL16v8 o GAL22v10?**

Per programmare un chip di quel tipo è necessario uno specifico programmatore hardware.

**Ho creato un circuito che voglio utilizzare in molti altri circuiti. Come posso farlo evitando di copiare il file nelle altre cartelle appropriate ripetutamente?**

Il circuito può essere salvato nella cartella "lib". Sarà poi disponibile in tutti gli altri circuiti.

## 9. Comandi da Tastiera

<b>Spazio</b>	Avvia o ferma la simulazione
<b>F6</b>	Apri il dialogo della tabella di misurazione.
<b>F7</b>	Esegui fino all'interruzione..
<b>F8</b>	Esegui i test case.
<b>C</b>	Esegui un singolo ciclo di clock (funziona solamente se c'è una simulazione in corso e c'è un componente di clock).
<b>V</b>	Esegue un singolo passo di porta.
<b>B</b>	Esegue tutti i singoli passi di porta fino a quando il circuito si è stabilizzato o alternativamente, se c'è un componente di interruzione, fino a quando la si raggiunge.
<b>F9</b>	Analisi del circuito.
<b>CTRL-A</b>	Seleziona tutto.
<b>CTRL-X</b>	Taglia i componenti selezionati nella clipboard.
<b>CTRL-C</b>	Copia i componenti selezionati nella clipboard.
<b>CTRL-V</b>	Inserisce i componenti dalla clipboard.
<b>CTRL-D</b>	Duplica la selezione corrente senza modificare la clipboard.
<b>R</b>	Quando si sta inserendo, ruota il componente.
<b>L</b>	Inserisce l'ultimo componente inserito un'altra volta.
<b>T</b>	Inserisce un tunnel.
<b>CTRL-N</b>	Nuovo circuito.
<b>CTRL-O</b>	Apri un circuito.
<b>CTRL-S</b>	Salva il circuito.
<b>CTRL-Z</b>	Annulla l'ultima modifica.
<b>CTRL-Y</b>	Rifai l'ultima modifica annullata.
<b>P</b>	Programma un diodo o FGFET.
<b>D</b>	Mentre si disegna un cavo, cambia alla modalità diagonale.
<b>F</b>	Mentre si disegna una linea, inverte l'orientazione.
<b>S</b>	Divide un cavo in due cavi.
<b>Esc</b>	Interrompe l'azione corrente.
<b>Canc</b>	Rimuove il componente selezionato.
<b>Backspace</b>	Rimuove i componenti selezionati.
<b>+</b>	Aumenta il numero di ingressi nel componente su cui il cursore è sopra. Se utilizzato con delle costanti, il valore è incrementato.



<b>-</b>	Diminuisce il numero di ingressi nel componente su cui il cursore è sopra. Se utilizzato con delle costanti, il valore è decrementato.
<b>CTRL +</b>	Incrementa lo zoom.
<b>CTRL -</b>	Decrementa lo zoom.
<b>F1</b>	Adatta la visuale alla dimensione del circuito.
<b>F5</b>	Mostra o nascondi la vista dell'albero dei componenti.

## B Impostazioni

Di seguito la descrizione delle impostazioni del simulatore disponibili.

### Impostazioni

Le impostazioni globali del simulatore specificano, tra l'altro, la lingua, il formato dei simboli da utilizzare e i percorsi per strumenti esterni.

#### Attributi

Utilizza forme IEEE 91-1984

Utilizza forme IEEE 91-1984 anziché forme rettangolari

Lingua

La lingua dell'interfaccia. Cambierà soltanto dopo un riavvio.

Formato

Formato delle espressioni nello schermo.

Schema di colori

Schema di colori

Colori Definiti dall'Utente

Colori Definiti dall'Utente

La vista della gerarchia dei componenti è visibile all'avvio.

Se impostato, la vista della gerarchia dei componenti è visibile all'avvio.

Mostra la Griglia

Mostra una griglia nella finestra principale.

Mostra il numero di cavi in un bus.

ATTENZIONE: Il valore è aggiornato solamente quando la simulazione viene avviata.

Suggerimenti sui Cavi.

Se impostato, le linee sono evidenziate quando il cursore passa sopra di esse.

Usa il Tasto Uguale

Utilizza il tasto "Uguale" piuttosto che il tasto "Più". Questo è sempre utile se il segno più non è una chiave primaria, ma la seconda assegnazione del carattere uguale come per esempio nelle tastiere francesi ed americane.

Increase numbers in identifiers when copying.

If set, in components whose label ends with a number, this number is incremented when copying.

Mostra il dialogo per rinominare automaticamente i tunnel.

Se impostato, una finestra di dialogo viene mostrata per rinominare automaticamente i tunnel con lo stesso nome dopo aver rinominato un tunnel.

Libreria

La cartella che contiene la libreria con dei circuiti predefiniti. Contiene, per esempio, i componenti della serie 74xx. Puoi anche aggiungere i tuoi circuiti posizionandoli in questa cartella. Bisogna assicurarsi che tutti i nomi dei file e sottocartelle siano univoci.

Libreria Java

Un file JAR contenente componenti aggiuntivi implementati in Java.

ATF15xx Fitter

Percorso al fitter per il ATF15xx. Inserisci la cartella che contiene il file fit15xx.exe fornito da Microchip (precedentemente ATMEL).

**ATMISP**

Percorso dell'eseguibile atmisp.exe. Se impostato, il software ATMISP può essere avviato automaticamente!

**GHDL**

Percorso all'eseguibile GHDL. Solamente necessario se si vuole usare GHDL per simulare i componenti definiti con VHDL.

**IVerilog**

Percorso all'eseguibile IVerilog. Solamente necessario se si vuole usare IVerilog per simulare i componenti definiti con VHDL.

**Configurazione della Toolchain**

Utilizzato per configurare l'integrazione di una toolchain. Consente l'utilizzo di strumenti esterno, come per esempio la programmazione di una FPGA.

**Dimensione del Font dei Menù [%]**

Dimensione dei font utilizzati nel menù espressa in percentuale rispetto a quella predefinita.

**Utilizza il click del mouse di macOS.**

Utilizza Ctrl+Click anziché il click destro.

**Consenti la connessione remota**

Se impostato, una porta TCP/IP viene aperta, attraverso la quale è possibile il controllo del simulatore.

**Numero della porta**

La porta sulla quale il server remoto viene aperto.

**Impostazioni specifiche del circuito**

Le impostazioni specifiche del circuito hanno effetto sul comportamento del circuito aperto. Per esempio, la figura che rappresenta il circuito quando è incorporato in altri circuiti. Queste impostazioni sono memorizzate insieme al circuito.

**Attributi****Etichetta**

Il nome di questo elemento.

**Larghezza**

Larghezza del simbolo se questo circuito è utilizzato come un componente in un altro circuito.

**Colore di Sfondo**

Il colore di sfondo del circuito quando viene incorporato dentro un altro circuito. Non è utilizzato per package DIL.

**Descrizione**

Una breve descrizione di questo elemento e del suo utilizzo.

**Modifica bloccata**

Il circuito è bloccato. Si può configurare diodi e FGF-FET.

**Figura**

La figura da utilizzare per la rappresentazione del circuito in un circuito che lo incorpora. Nella modalità "semplice", gli ingressi sono rappresentati sulla sinistra e le uscite sulla destra di un semplice rettangolo. Con la modalità "layout", la posizione degli ingressi e delle uscite e la loro orientazione nel circuito determina il posizionamento dei pin. Qui è possibile avere pin in alto o in basso. Quando si seleziona "chip DIL", un alloggiamento DIL è utilizzato per mostrare il circuito. I numeri dei pin degli ingressi e delle uscite determina la posizione dei pin nel case.

### Figura Personalizzata

Importa un file SVG

### Altezza

Altezza del simbolo se questo circuito è utilizzato come un componente in un altro circuito.

### Numero di pin DIL

Numero dei pin. Uno zero significa che il numero dei pin è determinato automaticamente.

### Contenuto delle ROM

Contenuto di tutte le ROM utilizzate

### Mostra i valori di misurazione all'avvio della simulazione.

Quando la simulazione si avvia, una tabella con i valori misurati è mostrata.

### Mostra il grafico delle misurazioni durante l'avvio della simulazione.

Quando la simulazione si avvia, un grafico con i valori misurati viene mostrato.

### Mostra il grafico della misurazione in modalità a passo singolo all'avvio della simulazione

Quando la simulazione si avvia, il grafico con i valori misurati nella modalità a passo singolo viene mostrato. Tutti i cambiamenti delle porte sono inclusi nel grafico.

### Massimo numero dei passaggi da mostrare

Il massimo numero dei valori memorizzati. Quando il massimo viene raggiunto il valore più vecchio è scartato.

### Precarica la memoria contenente il programma all'avvio.

Quando si simula un processore che utilizza una RAM come la memoria per il programma, è difficile avviare il processore perché il contenuto della RAM è sempre inizializzato con zeri all'avvio della simulazione. Questa impostazione consente il caricamento dei dati durante l'avvio. La memoria va indicata come tale.

### File del Programma

Il file che deve essere caricato nella memoria del programma all'avvio della simulazione.

### Usa il big endian durante l'importazione.

Utilizza l'ordinamento dei byte "big endian" durante l'importazione.

### Salta l'esportazione Verilog/VHDL

Salta l'esportazione della generazione dei componenti interni al circuito in Verilog/VHDL. I riferimenti ai circuiti sono mantenuti, rendendo possibile la sovrascrittura della loro implementazione.

### Il circuito è generico

Consente di creare un circuito generico

### Identificazione dell'oscillazione

Numero di propagazioni oltre il quale un'oscillazione viene identificata se il circuito non si è ancora stabilizzato.

### In case of oscillations, continue with random values.

If this option is set, the behavior of the simulation becomes unpredictable and extremely slow if there is an oscillation!

## C Interfaccia a Linea di Comando

`java -cp Digital.jar CLI`

`test -circ [String] [-tests [String]] [-allowMissingInputs] [-verbose]:`

Il primo nome di file specifica il circuito da testare. Se un secondo nome di file è specificato, i test case sono eseguiti da questo file. Se nessun secondo nome è specificato, i test case sono eseguiti dal primo file.

Opzioni:

`-circ [String(def: )]`

Nome del file da testare.

`[-tests [String(def: )]]`

Nome di un file con i test case.

`[-allowMissingInputs(def: false)]`

Consente la mancanza di ingressi nel circuito definiti nel test case. Questo può essere utile quando ci sono più possibili soluzioni che possono dipendere da ingressi differenti.

`[-verbose(def: false)]`

Se impostato, la tabella dei valori viene visualizzata nel caso di un errore.

`svg -dig [String] [-svg [String]] [-ieee] [-LaTeX] [-pinsInMathMode] [-hideTest] [-noShapeFilling] [-smallIO] [-noPinMarker] [-thinnerLines] [-highContrast] [-monochrome]:`

Può essere utilizzato per creare un file SVG da un circuito.

Opzioni:

`-dig [String(def: )]`

Il nome del file che rappresenta il circuito.

`[-svg [String(def: )]]`

Il nome del file SVG che verrà scritto.

`[-ieee(def: false)]`

Usa i simboli IEEE.

`[-LaTeX(def: false)]`

Il testo è inserito in notazione LaTeX. Inkscape è richiesto per ulteriore processamento.

`[-pinsInMathMode(def: false)]`

Per le etichette dei pin, utilizza la modalità matematica anche se nessun indice è contenuto.

`[-hideTest(def: false)]`

Nascondi i test case

`[-noShapeFilling(def: false)]`

Poligoni vuoti.

`[-smallIO(def: false)]`

Gli ingressi e le uscite sono rappresentate da piccoli cerchi.

`[-noPinMarker(def: false)]`

I marcatori rossi e blu del pin nei simboli sono omessi.

`[-thinnerLines(def: false)]`

Se impostato, le linee sono disegnate leggermente più sottili.

`[-highContrast(def: false)]`

I cavi e il testo dei pin sono visualizzati in nero-

`[-monochrome(def: false)]`

Soltanto sfumature di grigio sono utilizzate.

`stats -dig [String] [-csv [String]]:`

Crea un file SVG che contiene le statistiche del circuito. Tutti i componenti sono elencati nel file CSV.

Opzioni:

`-dig [String(def: )]`

Nome del file del circuito.

`[-csv [String(def: )]]`

Nome del file CSV da creare. Se questa opzione manca, la tabella è scritta nello standard output.

`run -dig [String]:`

Runs a circuit headless.

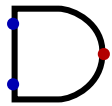
Opzioni:

`-dig [String(def: )]`

File name of the circuit.

## D Componenti

### 1. Logica



#### 1.1. And

Porta AND binaria. Restituisce alto se e solo se tutti gli ingressi sono impostati ad alto. Si possono anche usare i bus con più bit come ingresso ed uscita. In tal caso, una AND bit-a-bit è applicata ad ogni ingresso. Esportabile su VHDL/Verilog.

Ingressi

In\_1

Il 1. valore di ingresso per l'operazione logica.

In\_2

Il 2. valore di ingresso per l'operazione logica.

Uscite

out

Restituisce il risultato dell'operazione logica.

Attributi

Bit di Dati

Numero di bit dati utilizzati.

Numero di ingressi

Numero di ingressi utilizzati. Ogni ingresso deve essere connesso.

Ingressi invertiti

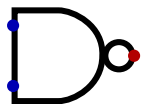
Puoi selezionare gli ingressi che devono essere invertiti.

Rotazione

L'orientazione dell'elemento nel circuito.

Figura Larga

Utilizza una figura più larga per visualizzare la porta.



#### 1.2. NAnd

Una combinazione di AND e NOT. Restituisce basso solo se tutti gli ingressi sono impostati ad alto. Se uno degli ingressi è impostato a basso, l'uscita è impostata alta. Si possono anche utilizzare i bus con più bit. In tal caso, una NAND bit-a-bit è applicata ad ogni ingresso. Esportabile su VHDL/Verilog.

### Ingressi

In\_1

Il 1. valore di ingresso per l'operazione logica.

In\_2

Il 2. valore di ingresso per l'operazione logica.

### Uscite

out

Restituisce il risultato dell'operazione logica.

### Attributi

Bit di Dati

Numero di bit dati utilizzati.

Numero di ingressi

Numero di ingressi utilizzati. Ogni ingresso deve essere connesso.

Ingressi invertiti

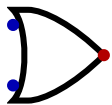
Puoi selezionare gli ingressi che devono essere invertiti.

Rotazione

L'orientazione dell'elemento nel circuito.

Figura Larga

Utilizza una figura più larga per visualizzare la porta.



## 1.3. Or

Porta OR. Restituisce alto se almeno uno degli ingressi è alto. Se tutti gli ingressi sono bassi, l'uscita sarà anch'essa bassa. Si possono anche utilizzare i bus con più bit. In tal caso, una OR bit-a-bit è applicata ad ogni ingresso. Esportabile su VHDL/Verilog.

### Ingressi

In\_1

Il 1. valore di ingresso per l'operazione logica.

In\_2

Il 2. valore di ingresso per l'operazione logica.

### Uscite

out

Restituisce il risultato dell'operazione logica.

### Attributi

Bit di Dati

Numero di bit dati utilizzati.

Numero di ingressi

Numero di ingressi utilizzati. Ogni ingresso deve essere connesso.

Ingressi invertiti

Puoi selezionare gli ingressi che devono essere invertiti.

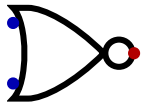


Rotazione

L'orientazione dell'elemento nel circuito.

Figura Larga

Utilizza una figura più larga per visualizzare la porta.



#### 1.4. NOR

Una combinazione di OR e NOT. Restituisce basso se uno degli ingressi è alto. Se tutti gli ingressi sono bassi, allora anche l'uscita sarà bassa. Si possono anche utilizzare i bus con più bit. In tal caso, una NOR bit-a-bit è applicata ad ogni ingresso. Esportabile su VHDL/Verilog.

Ingressi

In\_1

Il 1. valore di ingresso per l'operazione logica.

In\_2

Il 2. valore di ingresso per l'operazione logica.

Uscite

out

Restituisce il risultato dell'operazione logica.

Attributi

Bit di Dati

Numero di bit dati utilizzati.

Numero di ingressi

Numero di ingressi utilizzati. Ogni ingresso deve essere connesso.

Ingressi invertiti

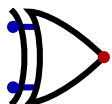
Puoi selezionare gli ingressi che devono essere invertiti.

Rotazione

L'orientazione dell'elemento nel circuito.

Figura Larga

Utilizza una figura più larga per visualizzare la porta.



#### 1.5. XOR

Se si utilizzano solo due ingressi, l'uscita è bassa se i bit di ingresso sono uguali. Altrimenti l'uscita è alta. Se si utilizzano più, allora si comporta come una cascata di porte XOR ( $A \text{ XOR } B \text{ XOR } C = (A \text{ XOR } B) \text{ XOR } C$ ). Si possono anche utilizzare i bus con più bit. In tal caso, una XOR bit-a-bit è applicata ad ogni ingresso. Esportabile su VHDL/Verilog.

### Ingressi

In\_1

Il 1. valore di ingresso per l'operazione logica.

In\_2

Il 2. valore di ingresso per l'operazione logica.

### Uscite

out

Restituisce il risultato dell'operazione logica.

### Attributi

Bit di Dati

Numero di bit dati utilizzati.

Numero di ingressi

Numero di ingressi utilizzati. Ogni ingresso deve essere connesso.

Ingressi invertiti

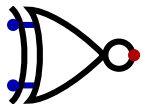
Puoi selezionare gli ingressi che devono essere invertiti.

Rotazione

L'orientazione dell'elemento nel circuito.

Figura Larga

Utilizza una figura più larga per visualizzare la porta.



## 1.6. XNOr

Una combinazione di XOR e NOT. Gli ingressi sono combinati con l'operazione XOR. Il risultato è poi negato. Si possono anche usare i bus con più bit. In tal caso, una XNOR bit-a-bit è applicata ad ogni ingresso. Esportabile su VHDL/Verilog.

### Ingressi

In\_1

Il 1. valore di ingresso per l'operazione logica.

In\_2

Il 2. valore di ingresso per l'operazione logica.

### Uscite

out

Restituisce il risultato dell'operazione logica.

### Attributi

Bit di Dati

Numero di bit dati utilizzati.

Numero di ingressi

Numero di ingressi utilizzati. Ogni ingresso deve essere connesso.

Ingressi invertiti

Puoi selezionare gli ingressi che devono essere invertiti.

Rotazione

L'orientazione dell'elemento nel circuito.

Figura Larga

Utilizza una figura più larga per visualizzare la porta.



## 1.7. Not

Inverte il valore dell'ingresso. Un valore alto diventa basso e viceversa. Si possono anche usare i bus con più bit. In tal caso, una NOT bit-a-bit è applicata ad ogni ingresso. Esportabile su VHDL/Verilog.

Ingressi

in

L'ingresso della porta NOT.

Uscite

out

L'ingresso invertito.

Attributi

Bit di Dati

Numero di bit dati utilizzati.

Rotazione

L'orientazione dell'elemento nel circuito.

Figura Larga

Utilizza una figura più larga per visualizzare la porta.

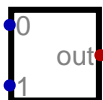


Tabella di ricerca

## 1.8. Tabella di Lookup

Ottiene il valore dell'uscita da una tabella memorizzata. Così, questa porta può emulare ogni porta combinatoria. Esportabile su VHDL/Verilog.

Ingressi

0

Ingresso 0. Questo ingresso in combinazione con tutti gli altri definisce l'indirizzo del valore memorizzato che deve essere restituito.

1

Ingresso 1. Questo ingresso in combinazione con tutti gli altri definisce l'indirizzo del valore memorizzato che deve essere restituito.

Uscite

out

Restituisce il valore memorizzato all'indirizzo impostato tramite gli ingressi.

## Attributi

## Bit di Dati

Numero di bit dati utilizzati.

## Numero di ingressi

Numero di ingressi utilizzati. Ogni ingresso deve essere connesso.

## Etichetta

Il nome di questo elemento.

## Dati

I valori memorizzati in questo elemento.

## Rotazione

L'orientazione dell'elemento nel circuito.

## 2. IO



### 2.1. Uscita

Può essere utilizzato per visualizzare il segnale d'uscita di un circuito. Questo elemento è anche utilizzato per connettere un circuito al suo padre. In questo caso la connessione è bidirezionale. Tramite questo si può anche assegnare un numero del pin, se il codice per CPLD o FPGA è generato. Esportabile su VHDL/Verilog.

## Ingressi

in

Questo valore è usato per la connessione dell'uscita.

## Attributi

## Bit di Dati

Numero di bit dati utilizzati.

## Etichetta

Il nome di questo elemento.

## Descrizione

Una breve descrizione di questo elemento e del suo utilizzo.

## Rotazione

L'orientazione dell'elemento nel circuito.

## Formato Numerico

Il formato utilizzato per mostrare i numeri.

## Cifre decimali

Il numero delle cifre decimali binarie

## Numero del pin

Numero di questo pin. Usato per la rappresentazione di un circuito in un package DIL e per l'assegnazione del pin durante la programmazione di un CPLD. Se ci sono più bit, tutti i numeri di pin possono essere specificati in una lista separata da virgola.

## Mostra nel Grafico Misurazioni

Mostra il valore nel grafico delle misurazioni.

## Piccola Figura

Se selezionato, una figura più piccola sarà utilizzata.



## 2.2. LED

Un LED può essere utilizzato per visualizzare il valore di uscita. Accetta un bit singolo. Si illumina se l'ingresso è alto.

Ingressi

in

Ingresso del LED. Si illumina se l'ingresso è alto.

Attributi

Etichetta

Il nome di questo elemento.

Colore

Il colore dell'elemento.

Rotazione

L'orientazione dell'elemento nel circuito.

Dimensione

La dimensione della figura del circuito.

Mostra nel Grafico Misurazioni

Mostra il valore nel grafico delle misurazioni.



## 2.3. Ingresso

Può essere utilizzato per manipolare in forma interattiva un segnale di ingresso in un circuito con il mouse. Questo elemento è anche utilizzato per connettere un circuito al suo padre. In questo caso, la connessione è bidirezionale. Tramite questo si può anche assegnare un numero del pin, se il codice per CPLD o FPGA è generato. Esportabile su VHDL/Verilog.

Uscite

out

Restituisce il valore che è connesso a questo ingresso.

Attributi

Bit di Dati

Numero di bit dati utilizzati.

Etichetta

Il nome di questo elemento.

Descrizione

Una breve descrizione di questo elemento e del suo utilizzo.

Rotazione

L'orientazione dell'elemento nel circuito.

Predefinito

Questo valore è impostato se il circuito è avviato. Una "Z" significa uno stato di alta impedenza.

**Ingresso a tre stati**

Se impostato, all'ingresso viene consentito di essere in alta impedenza. Questo viene consentito anche all'ingresso del componente se l'alta impedenza ("Z") è impostata come valore predefinito.

**Nessuna uscita bassa.**

Evita un'uscita bassa. Questo è specialmente utile quando si impostano circuiti di relay. Può essere attivato soltanto se una uscita ad alta impedenza è consentita.

**Formato Numerico**

Il formato utilizzato per mostrare i numeri.

**Cifre decimali**

Il numero delle cifre decimali binarie

**Numero del pin**

Numero di questo pin. Usato per la rappresentazione di un circuito in un package DIL e per l'assegnazione del pin durante la programmazione di un CPLD. Se ci sono più bit, tutti i numeri di pin possono essere specificati in una lista separata da virgola.

**Mostra nel Grafico Misurazioni**

Mostra il valore nel grafico delle misurazioni.

**Piccola Figura**

Se selezionato, una figura più piccola sarà utilizzata.



## 2.4. Ingresso Clock

Un segnale di clock. Lo si può controllare da un clock a tempo reale. A seconda della complessità del circuito, la frequenza del clock raggiunta potrebbe essere minore di quella selezionata. Se la frequenza è più alta di 50Hz, la rappresentazione del circuito non sarà più aggiornata per ogni ciclo di clock in modo tale da non aggiornare più i colori dei cavi. Se il clock a tempo reale non è attivato, il clock stesso può essere controllato da click del mouse. Tramite questo si può anche assegnare un numero del pin, se il codice per CPLD o FPGA è generato. Esportabile su VHDL/Verilog.

**Uscite****C**

Alterna da basso ad alto all'interno della frequenza di clock selezionata.

**Attributi****Etichetta**

Il nome di questo elemento.

**Avvia il clock a tempo reale**

Se abilitato, il clock a tempo reale è avviato quando il circuito è avviato

**Frequenza/Hz**

La frequenza in tempo reale utilizzata per questo clock.

**Rotazione**

L'orientazione dell'elemento nel circuito.

**Numero del pin**

Numero di questo pin. Usato per la rappresentazione di un circuito in un package DIL e per l'assegnazione del pin durante la programmazione di un CPLD. Se ci sono più bit, tutti i numeri di pin possono essere specificati in una lista separata da virgola.

Piccola Figura

Se selezionato, una figura più piccola sarà utilizzata.



## 2.5. Bottone

Un semplice bottone a pressione che ritorna al suo stato originale una volta rilasciato.

Uscite

out

Il segnale di uscita del bottone.

Attributi

Etichetta

Il nome di questo elemento.

Attivo Basso

Se selezionato l'uscita è bassa se il componente è attivo.

Mappa a tastiera

Un bottone è mappato alla tastiera. Per usare i tasti del cursore, usa UP (Sù), DOWN (Giù), RIGHT (Destra) e LEFT (Sinistra) come etichette.

Rotazione

L'orientazione dell'elemento nel circuito.

Mostra nel Grafico Misurazioni

Mostra il valore nel grafico delle misurazioni.



## 2.6. Interruttore DIP

Semplice interruttore DIP il cui output può essere solo alto o basso.

Uscite

out

Il valore d'uscita dell'interruttore.

Attributi

Etichetta

Il nome di questo elemento.

Descrizione

Una breve descrizione di questo elemento e del suo utilizzo.

Rotazione

L'orientazione dell'elemento nel circuito.

Uscita Alta

Il valore d'uscita predefinito di un interruttore DIP quando la simulazione parte.



## 2.7. Sonda

Un valore di misurazione che può essere mostrato nel grafico dei dati o nella tabella delle misurazioni. Questo componente può essere usato per osservare facilmente i valori nei circuiti incorporati.

Ingressi

in

Il valore della misurazione.

Attributi

Etichetta

Il nome di questo elemento.

Modalità di Visualizzazione

Definisce se il valore o un contatore viene mostrato.

Rotazione

L'orientazione dell'elemento nel circuito.

Formato Numerico

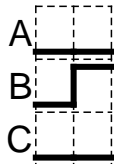
Il formato utilizzato per mostrare i numeri.

Cifre decimali

Il numero delle cifre decimali binarie

Mostra nel Grafico Misurazioni

Mostra il valore nel grafico delle misurazioni.



## 2.8. Grafico Dati

Mostra un grafico all'interno del pannello del circuito. Puoi graficare cicli di clock completi o semplici cambiamenti delle porte. Non ha alcun effetto sulla simulazione.

Attributi

Mostra i passaggi singoli.

Mostra tutti i passaggi singoli nel grafico.

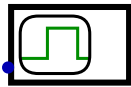
Massimo numero dei passaggi da mostrare

Il massimo numero dei valori memorizzati. Quando il massimo viene raggiunto il valore più vecchio è scartato.

Allinea alla Griglia

Se impostato, il componente è allineato alla griglia.





## 2.9. Grafico Dati ad Attivazione

Mostra un grafico di valori misurati, dove ogni valore misurato è memorizzato soltanto se il segnale di ingresso cambia. La memorizzazione avviene quando il circuito si è stabilizzato. L'attivazione non dà origine alla misurazione come in un vero oscilloscopio, ma ogni evento di attivazione fa memorizzare un singolo valore di misurazione per ogni segnale mostrato. Come ingresso diretto c'è soltanto l'attivazione. Gli ingressi ed uscite del circuito, flip-flop e registri e i componenti sonni possono essere utilizzati come segnali. Questo può essere abilitato nei rispettivi componenti.

Ingressi

T

Un cambiamento in questo ingresso causa la memorizzazione di un valore di misurazione.

Attributi

Etichetta

Il nome di questo elemento.

Attivatore

Condizione di attivazione per la registrazione dei dati.

Massimo numero dei passaggi da mostrare

Il massimo numero dei valori memorizzati. Quando il massimo viene raggiunto il valore più vecchio è scartato.

## 3. IO - Schermi



### 3.1. LED RGB

Un LED RGB il cui colore può essere controllato tramite tre ingressi. In ognuno dei tre ingressi, un canale del colore è connesso.

Ingressi

R

Il canale rosso del colore.

G

Il canale verde del colore.

B

Il canale blu del colore.

Attributi

Bit di Dati

Numero di bit dati utilizzati.

Etichetta

Il nome di questo elemento.

Rotazione

L'orientazione dell'elemento nel circuito.

Dimensione

La dimensione della figura del circuito.



### 3.2. LED con due connessioni.

Un LED con connessioni per anodo e catodo. Il LED si illumina se l'anodo è connesso ad un segnale alto ed il catodo ad un segnale basso. Questo LED non può essere utilizzato come un resistore pull-down. Agisce solamente come un elemento visivo. Il resistore è mostrato solo per simboleggiare il resistore in serie per limitare la corrente.

Ingressi

A

La connessione all'anodo del LED

C

La connessione al catodo del LED

Attributi

Etichetta

Il nome di questo elemento.

Colore

Il colore dell'elemento.

Rotazione

L'orientazione dell'elemento nel circuito.

Mostra nel Grafico Misurazioni

Mostra il valore nel grafico delle misurazioni.



### 3.3. Bottone con LED

Un semplice bottone a pressione che ha un LED che può essere attivato da un segnale di ingresso.

Ingressi

in

L'ingresso per controllare il LED del bottone.

Uscite

out

Il segnale di uscita del bottone.

## Attributi

## Etichetta

Il nome di questo elemento.

## Attivo Basso

Se selezionato l'uscita è bassa se il componente è attivo.

## Mappa a tastiera

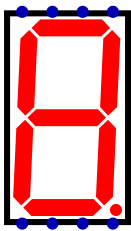
Un bottone è mappato alla tastiera. Per usare i tasti del cursore, usa UP (Sù), DOWN (Giù), RIGHT (Destra) e LEFT (Sinistra) come etichette.

## Colore

Il colore dell'elemento.

## Rotazione

L'orientazione dell'elemento nel circuito.



### 3.4. Display a Sette Segmenti

Display a sette segmenti, ogni segmento ha il suo ingresso di controllo.

## Ingressi

a

Questo ingresso controlla la linea orizzontale in alto.

b

Questo ingresso controlla la linea verticale in alto a destra.

c

Questo ingresso controlla la linea verticale in basso a destra.

d

Questo ingresso controlla la linea orizzontale in basso.

e

Questo ingresso controlla la linea verticale in basso a sinistra.

f

Questo ingresso controlla la linea verticale in alto a sinistra

g

Questo ingresso controlla la riga orizzontale in mezzo.

dp

Questo ingresso controlla il punto decimale.

## Attributi

## Colore

Il colore dell'elemento.

## Connessione Comune

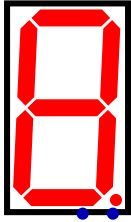
Se selezionato, anche un ingresso catodo o anodo comune è simulato.

## Comune

Tipologia di connessione comune.

Persistenza della visione

Specifica la durata dell'afterglow. Più grande il valore, più lunga la durata dell'effetto.



### 3.5. Display a Sette Segmenti Esadecimale

Display a sette segmenti con un ingresso a 4 bit esadecimali

Ingressi

d

Il valore in questo ingresso è visualizzata nel display.

dp

Questo ingresso controlla il punto decimale.

Attributi

Colore

Il colore dell'elemento.

Dimensione

La dimensione della figura del circuito.



### 3.6. Display a 16 Segmenti

Questo ingresso LED ha 16 bit che controllano i segmenti. Il secondo ingresso controlla il punto decimale.

Ingressi

led

Bus a 16 bit per controllare i LED.

dp

Questo ingresso controlla il punto decimale.

Attributi

Colore

Il colore dell'elemento.

Dimensione

La dimensione della figura del circuito.



### 3.7. Lampada

Una lampada con due connessioni. Se la corrente scorre, la lampada si illumina. La direzione della corrente non importa. La lampada si accende quando gli ingressi hanno valori differenti. La lampada si comporta in maniera simile ad una porta XOR.

Ingressi

- A  
Connessione
- B  
Connessione

Attributi

- Etichetta  
Il nome di questo elemento.
- Colore  
Il colore dell'elemento.
- Rotazione  
L'orientazione dell'elemento nel circuito.



Matrice di LED

### 3.8. Matrice di LED

Una matrice di LED. I LED sono mostrati in una finestra separata. I LED di una colonna del display sono controllati da una parola di bit. In un altro ingresso, la colonna corrente è selezionata. Quindi un display multiplexato è realizzato. I LED potrebbero rimanere accesi indefinitamente nella simulazione per prevenire sfarfallio del display.

Ingressi

- r-data  
Lo stato della riga dei LED di una colonna. Ciascun bit in questa parola di bit rappresenta lo stato di una riga della colonna corrente.
- c-addr  
Il numero della colonna corrente il cui stato è attualmente visibile nell'altro ingresso.

Attributi

- Etichetta  
Il nome di questo elemento.
- Righe  
Specifica il numero di righe tramite la specifica del numero di bit nella parola della riga.
- Bit di indirizzo delle colonne  
Indirizza le colonne individualmente. Tre bit significano otto colonne.

**Colore**

Il colore dell'elemento.

**Evita lo Sfarfallio**

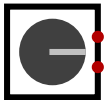
Non è possibile incrementare la frequenza così tanto da far sparire lo sfarfallio.

Per sopprimerlo, un "afterglow" può essere impostato come attivo per i LED con questa opzione. Se abilitato, i LED rimangono accesi anche se uno dei pin cambia valore verso alta impedenza. Questo simula una frequenza al di sopra della soglia di frequenza di fusione dello sfarfallio.

**Rotazione**

L'orientazione dell'elemento nel circuito.

## 4. IO - Meccanici



### 4.1. Trasduttore di Posizione a Rotella

Rotella con codificatore. Utilizzato per rilevare rotazioni.

**Uscite**

A

segnale A del codificatore

B

segnale B del codificatore

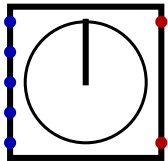
**Attributi**

Etichetta

Il nome di questo elemento.

Rotazione

L'orientazione dell'elemento nel circuito.



### 4.2. Motore Passo-Passo Unipolare

Motore passo-passo unipolare con due finecorsa. Sono supportati azionamento a passo completo, mezzo passo e passo d'onda.

### Ingressi

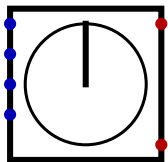
- P0  
Fase 0
- P1  
Fase 1
- P2  
Fase 2
- P3  
Fase 3
- com  
Collegamento comune della bobina centrale

### Uscite

- S0  
Finecorsa 0, diventa alto quando l'angolo del motore è di 0°.
- S1  
Finecorsa posizione 1, diventa alto quando l'angolo del motore è di 180°.

### Attributi

- Etichetta  
Il nome di questo elemento.
- Uscita invertita  
Se selezionato l'uscita è invertita.
- Rotazione  
L'orientazione dell'elemento nel circuito.



## 4.3. Motore Passo-Passo Bipolare

Motore passo-passo bipolare con due finecorsa. Sono supportati azionamento a passo completo, mezzo passo e passo d'onda.

### Ingressi

- A+  
Bobina A, positivo
- A-  
Bobina A, negativo
- B+  
Bobina B, positivo
- B-  
Bobina B, negativo

## Uscite

S0

Finecorsa 0, diventa alto quando l'angolo del motore è di 0°.

S1

Finecorsa posizione 1, diventa alto quando l'angolo del motore è di 180°.

## Attributi

Etichetta

Il nome di questo elemento.

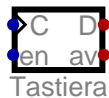
Uscita invertita

Se selezionato l'uscita è invertita.

Rotazione

L'orientazione dell'elemento nel circuito.

## 5. IO - Periferiche



### 5.1. Tastiera

Una tastiera può essere utilizzata per inserire del testo. Questo componente bufferizza l'ingresso, che può poi essere letto. Una finestra separata sarà aperta per inserire il testo.

## Ingressi

C

Clock. Un fronte di salita rimuove il carattere più vecchio dal buffer.

en

Se impostato alto, l'uscita D diventa attiva ed un carattere si presenta nell'uscita.  
Abilita anche l'ingresso del clock.

## Uscite

D

L'ultimo carattere scritto, oppure zero se nessun carattere è disponibile. L'uscita è il valore a 16 bit assegnato da Java.

av

Questa uscita indica che dei caratteri sono disponibili. Può essere utilizzata per attivare un interrupt.

## Attributi

Etichetta

Il nome di questo elemento.

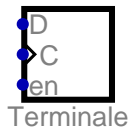
Ingressi invertiti

Puoi selezionare gli ingressi che devono essere invertiti.

Rotazione

L'orientazione dell'elemento nel circuito.





## 5.2. Terminale

Puoi scrivere caratteri ASCII nel terminale. Il terminale apre la propria finestra per visualizzare l'output.

Ingressi

D

Il dato da scrivere nel terminale

C

Clock. Un fronte di salita scrive il valore dell'ingresso nella finestra del terminale.

en

Un valore alto in questo ingresso abilita l'input del clock.

Attributi

Caratteri per Linea

Il numero di caratteri mostrato per una singola linea.

Linee

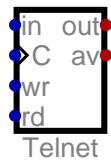
Il numero di linee da mostrare.

Etichetta

Il nome di questo elemento.

Rotazione

L'orientazione dell'elemento nel circuito.



## 5.3. Telnet

Abilita una connessione Telnet al circuito. Si può ricevere ed inviare caratteri tramite Telnet.

Ingressi

in

I dati da inviare.

C

Input del clock

wr

Se impostato, il byte nell'ingresso dati viene inviato.

rd

Se impostato, il byte ricevuto viene visualizzato.

## Uscite

out

Uscita Dati

av

Inserisce nell'uscita un dato, se presente.

## Attributi

Etichetta

Il nome di questo elemento.

Modalità Telnet

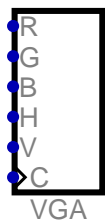
Se impostato, i comandi di controllo Telnet sono valutati. In aggiunta, il server invia i comandi SGA e ECHO. Se questa opzione è disabilitata, il server è un semplice server TCP.

Porta

La porta che viene aperta dal server.

Rotazione

L'orientazione dell'elemento nel circuito.



## 5.4. Monitor VGA

Analizza i segnali video in ingresso e visualizza la grafica corrispondente. Visto che la simulazione non può essere eseguita in tempo reale, il clock dei pixel è richiesto in aggiunta ai segnali video.

### Ingressi

R

Il componente rosso del colore

G

Il componente verde del colore

B

Il componente blu del colore

H

Il segnale di sincronizzazione orizzontale

V

Il segnale di sincronizzazione verticale

C

Il clock dei pixel

### Attributi

Etichetta

Il nome di questo elemento.

**Rotazione**

L'orientazione dell'elemento nel circuito.

**5.5. MIDI**

Utilizza il sistema MIDI per riprodurre delle note.

**Ingressi**

N

Nota

V

Volume

OnOff

Se impostato, questo si traduce nella pressione di un tasto nella tastiera (evento tasto premuto), se non impostato allora questo si traduce nel rilascio (evento tasto rilasciato).

en

Abilita il componente

C

Clock

**Attributi**

Etichetta

Il nome di questo elemento.

Canale MIDI

Seleziona il canale MIDI da usare.

Strumento MIDI

Seleziona lo strumento MIDI da usare.

Consenti la modifica del programma

Aggiunge un nuovo ingresso PC. Se questo ingresso è alto, il valore all'ingresso N è utilizzato per cambiare il programma (strumento).

Rotazione

L'orientazione dell'elemento nel circuito.

**6. Cavi****6.1. Massa**

Una connessione a massa. L'uscita è sempre zero. Esportabile su VHDL/Verilog.

#### Uscite

out

L'uscita restituisce sempre bassa.

#### Attributi

Bit di Dati

Numero di bit dati utilizzati.

Etichetta

Il nome di questo elemento.

Rotazione

L'orientazione dell'elemento nel circuito.



### 6.2. Tensione di alimentazione

Una connessione alla tensione di alimentazione. L'uscita è sempre alta. Esportabile su VHDL/Verilog.

#### Uscite

out

Questa uscita è sempre alta.

#### Attributi

Bit di Dati

Numero di bit dati utilizzati.

Etichetta

Il nome di questo elemento.

Rotazione

L'orientazione dell'elemento nel circuito.

1•

### 6.3. Valore costante

Un componente che restituisce un valore come una semplice costante. Il valore può essere impostato nel dialogo degli attributi. Esportabile su VHDL/Verilog.

#### Uscite

out

Restituisce il valore sotto forma di costante.

#### Attributi

Bit di Dati

Numero di bit dati utilizzati.

Valore

Il valore della costante.

Rotazione

L'orientazione dell'elemento nel circuito.

**Formato Numerico**

Il formato utilizzato per mostrare i numeri.

**Cifre decimali**

Il numero delle cifre decimali binarie

**6.4. Tunnel**

Connette componenti senza un cavo. Tutti i componenti tunnel, che hanno lo stesso nome della rete, sono connessi insieme. Funziona soltanto localmente, quindi non è possibile connettere circuiti differenti. I tunnel senza nome sono silenziosamente ignorati. Esportabile su VHDL/Verilog.

**Ingressi**

in

La connessione al tunnel.

**Attributi****Nome di rete**

Tutte le reti hanno un nome identico se connesse insieme.

**Rotazione**

L'orientazione dell'elemento nel circuito.

**6.5. Scissione/Divisione**

Divide o crea un insieme di cavi o un bus dati con più di un bit. Con un bus è per esempio possibile generare connessioni a 16 bit senza aver bisogno di piazzare 16 cavi individuali. Tutte le 16 connessioni possono essere unite in un unico cavo. Lo splitter ha una direzione, ovvero può trasmettere segnali soltanto verso una direzione. Esportabile su VHDL/Verilog.

**Ingressi**

0-3

I bit di ingresso 0-3.

4-7

I bit di ingresso 4-7.

**Uscite**

0-7

I bit di uscita 0-7.

**Attributi**

**Splitting dell'ingresso**

Se per esempio quattro bit, due bit e due altri bit sono usati per l'ingresso, questo può essere configurato come "4,2,2". Il numero indica il numero di bit. Per convenienza, l'asterisco può essere usato con "[Bit]\*[Numero]" come "1\*16". Si può anche specificare i bit da utilizzare direttamente in qualsiasi ordine. Per esempio "4-7,0-3" configura i bit 4-7 e 0-3. Questa notazione consente qualsiasi configurazione di bit. I bit di ingresso devono essere specificati completamente e senza ambiguità.

**Divisione dell'uscita**

Se per esempio quattro bit, due bit e due altri bit sono usati per l'ingresso, questo può essere configurato come "4,2,2". Il numero indica il numero di bit. Per convenienza, l'asterisco può essere usato con "[Bit]\*[Numero]" come "1\*16". Si può anche specificare i bit da utilizzare direttamente in qualsiasi ordine. Per esempio "4-7,0-3" configura i bit 4-7 e 0-3. Questa notazione consente qualsiasi configurazione di bit. I bit di ingresso devono essere specificati completamente e senza ambiguità.

**Rotazione**

L'orientazione dell'elemento nel circuito.

**Specchio**

Specchia il componente nel circuito.

**Spargimento**

Configura lo spreading degli ingressi ed uscite nel circuito.

**6.6. Driver**

Un driver può essere usato per connettere un valore di un segnale ad un altro cavo. Se l'ingresso di selezione è basso, l'uscita è in alta impedenza. Se l'ingresso di selezione è alto, l'uscita è impostata al valore in ingresso. Esportabile su VHDL/Verilog.

**Ingressi**

in

Il valore in ingresso del driver.

sel

Pin per controllare il driver. Se il valore è alto, l'ingresso è impostato all'uscita. Se il valore è basso, l'uscita è in alta impedenza.

**Uscite**

out

Se l'ingresso di selezione è alto, l'ingresso è assegnato a questa uscita. Se l'ingresso di selezione è basso, questa uscita è in alta impedenza.

**Attributi**

Bit di Dati

Numero di bit dati utilizzati.

Uscita invertita

Se selezionato l'uscita è invertita.

Inverti la posizione del selettore

Questa opzione ti consente di spostare il pin di selezione dal lato opposto del plexer.

**Rotazione**

L'orientazione dell'elemento nel circuito.

**6.7. Driver, selezione invertita**

Un driver può essere utilizzato per connettere un valore di un segnale ad un altro cavo. Se l'ingresso di selezione è alto, l'uscita è in alta impedenza. Se l'ingresso di selezione è basso, l'uscita è impostata al valore di ingresso. Esportabile su VHDL/Verilog.

**Ingressi**

in

Il valore in ingresso del driver.

sel

Pin per controllare il driver. Se il valore è basso, l'ingresso è impostato all'uscita. Se il valore è alto, l'uscita è in alta impedenza.

**Uscite**

out

Se l'ingresso di selezione è alto, l'ingresso è assegnato a questa uscita. Se l'ingresso di selezione è basso, questa uscita è in alta impedenza.

**Attributi**

Bit di Dati

Numero di bit dati utilizzati.

Uscita invertita

Se selezionato l'uscita è invertita.

Inverti la posizione del selettore

Questa opzione ti consente di spostare il pin di selezione dal lato opposto del plexer.

Rotazione

L'orientazione dell'elemento nel circuito.

**6.8. Ritardo.**

Ritarda il segnale di un tempo di ritardo di propagazione. Ritarda un segnale per un numero di ritardo della porta configurabile. Tutti gli altri componenti in Digital hanno un ritardo di porta equivalente ad un ritardo di propagazione. Questo componente può essere utilizzato per realizzare in qualsiasi tempo di propagazione necessario.

**Ingressi**

in

Ingresso del segnale su cui applicare un ritardo.

**Uscite**

out

L'ingresso del segnale ritardato di un certo ritardo porta.

## Attributi

## Bit di Dati

Numero di bit dati utilizzati.

## Durata

Tempo di ritardo in unità del ritardo di propagazione di porta comune.

## Rotazione

L'orientazione dell'elemento nel circuito.



### 6.9. Restitore di Pull-Up

Se una rete è in uno stato di alta impedenza, questo resistore porta la rete ad un valore alto. In qualsiasi altro caso, questo componente non ha effetto.

## Uscite

## out

Un segnale alto "debole".

## Attributi

## Bit di Dati

Numero di bit dati utilizzati.

## Rotazione

L'orientazione dell'elemento nel circuito.



### 6.10. Resistore di Pull-Down

Se la rete è in uno stato di alta impedenza, questo resistore porta la rete ad un valore basso. In qualsiasi altro caso, questo componente non ha effetto.

## Uscite

## out

Un segnale basso "debole".

## Attributi

## Bit di Dati

Numero di bit dati utilizzati.

## Rotazione

L'orientazione dell'elemento nel circuito.





### 6.11. Non Connesso

Questo componente può essere usato per impostare un cavo ad alta impedenza. Se un ingresso di una porta logica è impostato ad alta impedenza, il valore in lettura è indefinito. Nota che in realtà in molti casi un eccessivo consumo di corrente e anche danneggiamento può verificarsi se un ingresso digitale non è impostato ad alto o basso ma rimane scollegato.

Uscite

out

Questa uscita è sempre ad alta impedenza.

Attributi

Bit di Dati

Numero di bit dati utilizzati.

## 7. Plexer



### 7.1. Multiplexer

Un componente che usa il valore di un pin di selezione per decidere quale valore di ingresso impostare nell'uscita. Esportabile su VHDL/Verilog.

Ingressi

sel

Questo ingresso è utilizzato per selezionare l'ingresso da riflettere nell'uscita.

in\_0

Il 0. ingresso dati del multiplexer.

in\_1

Il 1. ingresso dati del multiplexer.

Uscite

out

Il valore dell'ingresso selezionato.

Attributi

Bit di Dati

Numero di bit dati utilizzati.

Numero dei bit di Selezione

Numero dei bit utilizzati per l'input di selezione.

Inverti la posizione del selettore

Questa opzione ti consente di spostare il pin di selezione dal lato opposto del plexer.

Rotazione

L'orientazione dell'elemento nel circuito.



## 7.2. Demultiplexer

Un componente che può mettere il valore in ingresso ad una delle sue uscite. Tutte le altre sono impostate al valore predefinito. Esportabile su VHDL/Verilog.

### Ingressi

sel

Questo pin seleziona l'uscita da usare.

in

Il valore di questo ingresso è dato all'uscita dati selezionata.

### Uscite

out\_0

Uscita dati 0.

out\_1

Uscita dati 1.

### Attributi

Bit di Dati

Numero di bit dati utilizzati.

Numero dei bit di Selezione

Numero dei bit utilizzati per l'input di selezione.

Inverti la posizione del selettore

Questa opzione ti consente di spostare il pin di selezione dal lato opposto del plexer.

Rotazione

L'orientazione dell'elemento nel circuito.

Predefinito

Questo valore è impostato quando il circuito viene avviato. Nel demultiplexer, questo valore è impostato per uscite non selezionate.



## 7.3. Decodificatore

Un pin dell'output selezionabile è impostato ad alto, tutte le altre uscite a basso. Esportabile su VHDL/Verilog.

### Ingressi

sel

Questo ingresso seleziona l'uscita abilitata. L'uscita selezionata è impostata ad alto, tutte le altre a basso.

## Uscite

out\_0

Uscita 0. Questa è alta se è selezionata dall'ingresso selezione.

out\_1

Uscita 1. Questa è alta se è selezionata dall'ingresso selezione.

## Attributi

Numero dei bit di Selezione

Numero dei bit utilizzati per l'input di selezione.

Inverti la posizione del selettore

Questa opzione ti consente di spostare il pin di selezione dal lato opposto del plexer.

Rotazione

L'orientazione dell'elemento nel circuito.

**7.4. Selettore del Bit**

Seleziona un singolo bit da un bus dati. Esportabile su VHDL/Verilog.

## Ingressi

in

Il bus in ingresso

sel

Questo ingresso seleziona il bit

## Uscite

out

Il bit selezionato.

## Attributi

Numero dei bit di Selezione

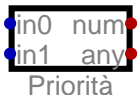
Numero dei bit utilizzati per l'input di selezione.

Inverti la posizione del selettore

Questa opzione ti consente di spostare il pin di selezione dal lato opposto del plexer.

Rotazione

L'orientazione dell'elemento nel circuito.

**7.5. Codificatore a Priorità**

Se uno degli ingressi è impostato, il suo numero è l'uscita. Se più di un ingresso è impostato simultaneamente, il numero più alto diventa l'uscita. Esportabile su VHDL/Verilog.

## Ingressi

in0

Il 0. ingresso del codificatore a priorità.

in1

Il 1. ingresso del codificatore a priorità.

## Uscite

num

Numero dell'ingresso impostato.

any

Se questa uscita è impostata, almeno uno degli ingressi è impostato.

## Attributi

Etichetta

Il nome di questo elemento.

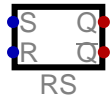
Numero dei bit di Selezione

Numero dei bit utilizzati per l'input di selezione.

Rotazione

L'orientazione dell'elemento nel circuito.

## 8. Flip-Flop



### 8.1. Flip-flop RS

Un componente per memorizzare un singolo bit. Fornisce funzioni di "set" (memorizzazione di un valore alto) e "reset" (impostazione di un valore basso). Se entrambe gli ingressi sono alti, anche tutte e due le uscite saranno alte. Se entrambe gli ingressi sono reimpostati a basso nello stesso momento, lo stato finale è imprevedibile.

## Ingressi

S

L'ingresso di "set".

R

L'ingresso di "reset".

## Uscite

Q

Restituisce il valore memorizzato.

 $\neg Q$ 

Restituisce il valore memorizzato, negato.

## Attributi

Etichetta

Il nome di questo elemento.

**Ingressi invertiti**

Puoi selezionare gli ingressi che devono essere invertiti.

**Rotazione**

L'orientazione dell'elemento nel circuito.

**Specchio**

Specchia il componente nel circuito.

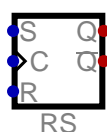
**Predefinito**

Questo valore è impostato quando il circuito viene avviato. Nel demultiplexer, questo valore è impostato per uscite non selezionate.

**Usa come valore di misurazione**

Se impostato, il valore è una misurazione e appare nel grafico e nella tabella dati.

In aggiunta, una etichetta deve essere specificata che serve per l'identificazione del valore.



## 8.2. RS-Flip-flop con Clock

Un componente per memorizzare un singolo bit. Fornisce funzioni di "set" (memorizzazione di un valore alto) e "reset" (impostazione di un valore basso). Se entrambe gli ingressi (S, R) sono impostati nel fronte di salita del clock, lo stato finale è imprevedibile.

**Ingressi**

S

L'ingresso di "set".

C

L'ingresso del clock. Un fronte di salita indica una transizione di stato.

R

L'ingresso di "reset".

**Uscite**

Q

Restituisce il valore memorizzato.

$\neg Q$

Restituisce il valore memorizzato, negato.

**Attributi**

Etichetta

Il nome di questo elemento.

Ingressi invertiti

Puoi selezionare gli ingressi che devono essere invertiti.

Rotazione

L'orientazione dell'elemento nel circuito.

Specchio

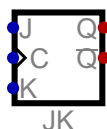
Specchia il componente nel circuito.

**Predefinito**

Questo valore è impostato quando il circuito viene avviato. Nel demultiplexer, questo valore è impostato per uscite non selezionate.

**Usa come valore di misurazione**

Se impostato, il valore è una misurazione e appare nel grafico e nella tabella dati. In aggiunta, una etichetta deve essere specificata che serve per l'identificazione del valore.

**8.3. Flip-flop JK**

Ha la possibilità di memorizzare ( $J=K=0$ ), impostare ( $J=1, K=0$ ), resettare ( $J=0, K=1$ ) o invertire ( $J=K=1$ ) il valore memorizzato. Un cambiamento di stato ha luogo soltanto nel fronte di salita del clock all'ingresso C. Esportabile su VHDL/Verilog.

**Ingressi**

J

L'ingresso di "set" del flip-flop.

C

L'ingresso del clock. Un fronte di salita indica una transizione di stato.

K

L'ingresso di "reset" del flip-flop.

**Uscite**

Q

Restituisce il valore memorizzato.

 $\neg Q$ 

Restituisce il valore memorizzato, negato.

**Attributi****Etichetta**

Il nome di questo elemento.

**Ingressi invertiti**

Puoi selezionare gli ingressi che devono essere invertiti.

**Rotazione**

L'orientazione dell'elemento nel circuito.

**Specchio**

Specchia il componente nel circuito.

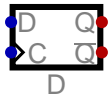
**Predefinito**

Questo valore è impostato quando il circuito viene avviato. Nel demultiplexer, questo valore è impostato per uscite non selezionate.

**Usa come valore di misurazione**

Se impostato, il valore è una misurazione e appare nel grafico e nella tabella dati.

In aggiunta, una etichetta deve essere specificata che serve per l'identificazione del valore.



## 8.4. Flip-flop D

Un componente usato per memorizzare un valore. Il valore nel pin D è memorizzato nel fronte di salita del clock C. Il numero di bit può essere selezionato, il che consente di memorizzare più di un bit. Esportabile su VHDL/Verilog.

### Ingressi

D

L'ingresso del bit da memorizzare.

C

L'ingresso del clock per memorizzare un valore. Il valore nell'ingresso D è memorizzato durante un fronte di salita su questo pin.

### Uscite

Q

Restituisce il valore memorizzato.

$\neg Q$

Restituisce il valore memorizzato, negato.

### Attributi

Bit di Dati

Numero di bit dati utilizzati.

Etichetta

Il nome di questo elemento.

Ingressi invertiti

Puoi selezionare gli ingressi che devono essere invertiti.

Rotazione

L'orientazione dell'elemento nel circuito.

Specchio

Specchia il componente nel circuito.

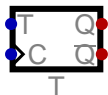
Predefinito

Questo valore è impostato quando il circuito viene avviato. Nel demultiplexer, questo valore è impostato per uscite non selezionate.

Usa come valore di misurazione

Se impostato, il valore è una misurazione e appare nel grafico e nella tabella dati.

In aggiunta, una etichetta deve essere specificata che serve per l'identificazione del valore.



## 8.5. Flip-flop D

Memorizza un singolo bit. Inverte lo stato con un fronte di salita sul clock C.

## Ingressi

T

Abilita la funzione di inversione.

C

Ingresso del clock. Un fronte di salita inverte l'uscita, se l'ingresso T è alto.

## Uscite

Q

Restituisce il valore memorizzato.

 $\neg Q$ 

Restituisce il valore memorizzato, negato.

## Attributi

Etichetta

Il nome di questo elemento.

Ingresso Enable

Se abilitato un ingresso enable (T) è disponibile.

Ingressi invertiti

Puoi selezionare gli ingressi che devono essere invertiti.

Rotazione

L'orientazione dell'elemento nel circuito.

Specchio

Specchia il componente nel circuito.

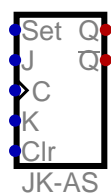
Predefinito

Questo valore è impostato quando il circuito viene avviato. Nel demultiplexer, questo valore è impostato per uscite non selezionate.

Usa come valore di misurazione

Se impostato, il valore è una misurazione e appare nel grafico e nella tabella dati.

In aggiunta, una etichetta deve essere specificata che serve per l'identificazione del valore.



## 8.6. JK-Flip-flop, asincrono

Ha la possibilità di memorizzare ( $J=K=0$ ), impostare ( $J=1, K=0$ ), resettare ( $J=0, K=1$ ) o invertire ( $J=K=1$ ) il valore memorizzato. Un cambiamento di stato ha luogo soltanto durante il fronte di salita nell'ingresso del clock C. Ci sono due ingressi aggiuntivi che impostati o resettano il valore memorizzato immediatamente senza un segnale di clock. Esportabile su VHDL/Verilog.



## Ingressi

## Set

"set" asincrono. Un valore alto effettua un "set" immediato nel flip-flop.

## J

L'ingresso "set" del flip-flop.

## C

L'ingresso del clock. Un fronte di salita inizia un cambiamento di stato.

## K

L'ingresso "reset" del flip-flop.

## Clr

"clear" asincrono. Un valore alto effettua un "clear" immediato nel flip-flop.

## Uscite

## Q

Restituisce il valore memorizzato.

 $\neg Q$ 

Restituisce il valore memorizzato, negato.

## Attributi

## Etichetta

Il nome di questo elemento.

## Ingressi invertiti

Puoi selezionare gli ingressi che devono essere invertiti.

## Rotazione

L'orientazione dell'elemento nel circuito.

## Specchio

Specchia il componente nel circuito.

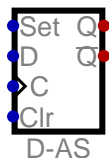
## Predefinito

Questo valore è impostato quando il circuito viene avviato. Nel demultiplexer, questo valore è impostato per uscite non selezionate.

## Usa come valore di misurazione

Se impostato, il valore è una misurazione e appare nel grafico e nella tabella dati.

In aggiunta, una etichetta deve essere specificata che serve per l'identificazione del valore.



## 8.7. D-Flip-flop, asincrono

Un componente utilizzato per memorizzare un valore. Il valore nel pin D è memorizzato durante un fronte di salita del pin del clock C. Ci sono due ingressi aggiuntivi che impostano o resettano lo stato immediatamente senza un segnale di clock. Il numero di bit può essere selezionato, il che consente di memorizzare più bit. Esportabile su VHDL/Verilog.

## Ingressi

## Set

"set" asincrono. Un valore alto imposta tutti i bit memorizzati ad un valore alto.

## D

Ingresso del bit da memorizzare.

## C

Pin di controllo per memorizzare un bit. Il bit nell'ingresso D è memorizzato durante un fronte di salita di questo pin.

## Clr

"clear" asincrono. Un valore alto imposta tutti i bit memorizzati ad un valore basso.

## Uscite

## Q

Restituisce il valore memorizzato.

 $\neg Q$ 

Restituisce il valore memorizzato, negato.

## Attributi

## Bit di Dati

Numero di bit dati utilizzati.

## Etichetta

Il nome di questo elemento.

## Ingressi invertiti

Puoi selezionare gli ingressi che devono essere invertiti.

## Rotazione

L'orientazione dell'elemento nel circuito.

## Specchio

Specchia il componente nel circuito.

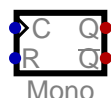
## Predefinito

Questo valore è impostato quando il circuito viene avviato. Nel demultiplexer, questo valore è impostato per uscite non selezionate.

## Usa come valore di misurazione

Se impostato, il valore è una misurazione e appare nel grafico e nella tabella dati.

In aggiunta, una etichetta deve essere specificata che serve per l'identificazione del valore.



## 8.8. Monoflop

Il monoflop è impostato ad un fronte di salita del clock. Dopo un tempo di ritardo configurabile, il monoflop sarà resettato automaticamente. Il monoflop può essere riattivato. Può essere usato soltanto se c'è esattamente un solo componente di clock presente nel circuito. Questo componente di clock è utilizzato come la base temporale con cui vengono misurati ritardi di tempo.

## Ingressi

C

Ingresso del clock. Un fronte di salita imposta il monoflop.

R

Ingresso del reset. Un valore alto resetta il monoflop.

## Uscite

Q

uscita

 $\neg Q$ 

uscita invertita

## Attributi

Etichetta

Il nome di questo elemento.

Larghezza dell'Impulso

La larghezza dell'impulso è misurata in cicli di clock.

Ingressi invertiti

Puoi selezionare gli ingressi che devono essere invertiti.

Rotazione

L'orientazione dell'elemento nel circuito.

Specchio

Specchia il componente nel circuito.

Predefinito

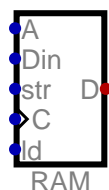
Questo valore è impostato quando il circuito viene avviato. Nel demultiplexer, questo valore è impostato per uscite non selezionate.

Usa come valore di misurazione

Se impostato, il valore è una misurazione e appare nel grafico e nella tabella dati.

In aggiunta, una etichetta deve essere specificata che serve per l'identificazione del valore.

## 9. Memoria - RAM



### 9.1. RAM, porte separate

Un modulo RAM con ingressi separati per memorizzare e uscite per leggere i dati memorizzati. Esportabile su VHDL/Verilog.

## Ingressi

A

L'indirizzo da cui leggere o scrivere.

Din

I dati da memorizzare nella RAM.

str

Se questo ingresso è alto quando c'è un fronte di salita nel clock, i dati sono memorizzati.

C

Ingresso del clock

Id

Se questo ingresso è alto allora l'uscita è attivata e i dati sono visibili.

## Uscite

D

Il pin dell'uscita dati

## Attributi

Bit di Dati

Numero di bit dati utilizzati.

Bit di Indirizzo

Numero dei bit di indirizzo utilizzati.

Etichetta

Il nome di questo elemento.

Rotazione

L'orientazione dell'elemento nel circuito.

Formato Numerico

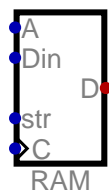
Il formato utilizzato per mostrare i numeri.

Cifre decimali

Il numero delle cifre decimali binarie

Memorial del Programma

Rende questa ROM una memoria del programma. In questo modo può essere programmata da un IDE esterno.

**9.2. Block-RAM, porte separate**

Un modulo RAM con ingressi di memorizzazione e uscite di lettura dati separati. La RAM aggiorna le sue uscite durante un fronte di salita del clock. Questo consente l'utilizzo di una Block-RAM su una FPGA. Esportabile su VHDL/Verilog.

## Ingressi

A

L'indirizzo da cui leggere o scrivere.

Din

I dati da memorizzare nella RAM.

str

Se questo ingresso è alto e c'è un fronte di salita del clock, i dati sono memorizzati.

C

L'ingresso del clock.

## Uscite

D

Il pin dell'uscita dati.

## Attributi

Bit di Dati

Numero di bit dati utilizzati.

Bit di Indirizzo

Numero dei bit di indirizzo utilizzati.

Etichetta

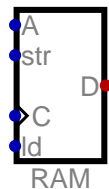
Il nome di questo elemento.

Rotazione

L'orientazione dell'elemento nel circuito.

Memorial del Programma

Rende questa ROM una memoria del programma. In questo modo può essere programmata da un IDE esterno.

**9.3. RAM, porte bidirezionali**

Un modulo RAM con pin bidirezionali per lettura e scrittura dei dati.

## Ingressi

A

L'indirizzo di scrittura o lettura dati.

str

Se l'ingresso è alto e c'è un fronte di salita del clock, i dati sono memorizzati.

C

Clock

Id

Se l'ingresso è alto l'uscita è attivata e i dati diventano visibili.

## Uscite

D

Connessione bidirezionale per i dati.

## Attributi

Bit di Dati

Numero di bit dati utilizzati.

Bit di Indirizzo

Numero dei bit di indirizzo utilizzati.

Etichetta

Il nome di questo elemento.

Rotazione

L'orientazione dell'elemento nel circuito.

Formato Numerico

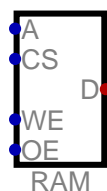
Il formato utilizzato per mostrare i numeri.

Cifre decimali

Il numero delle cifre decimali binarie

Memorial del Programma

Rende questa ROM una memoria del programma. In questo modo può essere programmata da un IDE esterno.



## 9.4. RAM, Chip Select

Un modulo RAM con una connessione bidirezionale per la lettura e scrittura dati. Se l'ingresso chip-select è basso, il componente è disabilitato. Questo consente la costruzione di una RAM più grande a partire da tanti altri moduli RAM più piccoli tramite l'utilizzo di un decodificatore di indirizzi. Il ciclo di scrittura funziona in questo modo: mettendo chip-select ad alto, il componente diventa selezionato; un fronte di salita nel write-enable scrive l'indirizzo e il fronte di discesa successivo memorizza i dati.

## Ingressi

A

L'indirizzo di lettura e scrittura dati.

CS

Se impostato ad alto, questo modulo di RAM è abilitato. Altrimenti l'uscita sarà sempre in alta impedenza.

WE

Se impostato ad alto, i dati sono scritti nella RAM.

OE

Se impostato ad alto, il valore memorizzato è presente nell'uscita.

## Uscite

D

La connessione dati bidirezionale.

## Attributi

Bit di Dati

Numero di bit dati utilizzati.

Bit di Indirizzo

Numero dei bit di indirizzo utilizzati.

Etichetta

Il nome di questo elemento.

Ingressi invertiti

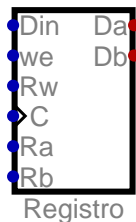
Puoi selezionare gli ingressi che devono essere invertiti.

Rotazione

L'orientazione dell'elemento nel circuito.

Memorial del Programma

Rende questa ROM una memoria del programma. In questo modo può essere programmata da un IDE esterno.



## 9.5. File Registro

Memoria con una operta che consente di scrivere e due porte che consentono di leggere dalla memoria simultaneamente. Può essere usata per implementare i registri di un processore. Due registri possono essere letti simultaneamente ed un terzo può essere scritto. Esportabile su VHDL/Verilog.

## Ingressi

Din

I dati da memorizzare nel registro Rw.

we

Se impostato ad alto, durante un fronte di salita del clock, i dati sono memorizzati.

Rw

Il registro in cui i dati sono scritti.

C

Clock

Ra

Il registro visibile nella porta A.

Rb

Il registro visibile nella porta B.

## Uscite

Da

-Porta d'uscita A

Db

Porta d'uscita B

## Attributi

Bit di Dati

Numero di bit dati utilizzati.

Bit di Indirizzo

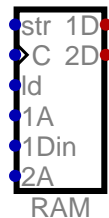
Numero dei bit di indirizzo utilizzati.

Etichetta

Il nome di questo elemento.

Rotazione

L'orientazione dell'elemento nel circuito.



## 9.6. RAM, Doppia Porta

Una RAM con una porta che consente di scrivere e leggere ed una seconda porta di sola lettura. Questa seconda porta può essere sata per fornire un accesso ai contenuti della memoria per la logica grafica. In questo modo, un processore può scrivere nella memoria video e la logica grafica visualizzerà il contenuto letto dalla RAM simultaneamente. Esportabile su VHDL/Verilog.

## Ingressi

str

Se impostato ad alto quando c'è un fronte di salita nel clock, i dati sono memorizzati.

C

Clock

ld

Se impostato ad alto, l'uscita è attivata e i dati sono resi visibili nell'uscita 1D.

1A

L'indirizzo in cui la porta 1 è letta o scritta.

1Din

I dati da memorizzare nella RAM.

2A

L'indirizzo utilizzato per leggere dalla porta 2.



## Uscite

1D

Porta di uscita 1

2D

Porta di uscita 2

## Attributi

Bit di Dati

Numero di bit dati utilizzati.

Bit di Indirizzo

Numero dei bit di indirizzo utilizzati.

Etichetta

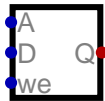
Il nome di questo elemento.

Rotazione

L'orientazione dell'elemento nel circuito.

Memorial del Programma

Rende questa ROM una memoria del programma. In questo modo può essere programmata da un IDE esterno.



RAM, asincrona

**9.7. RAM, asincrona**

Fino a quando il write-enable è impostato, l'ingresso è memorizzato. Corrisponde ad una semplice RAM, dove le linee di dati ed indirizzi sono direttamente connesse ai decodificatori delle celle di memoria. Esportabile su VHDL/Verilog.

## Ingressi

A

L'indirizzo in cui la lettura o scrittura ha luogo.

D

I dati da memorizzare.

we

Write enable. Fino a quando l'ingresso è alto, il valore applicato a D è memorizzato nell'indirizzo applicato su A in qualsiasi caso in cui A o D cambia.

## Uscite

Q

Uscita corrispondente ai dati memorizzati.

## Attributi

Bit di Dati

Numero di bit dati utilizzati.

Bit di Indirizzo

Numero dei bit di indirizzo utilizzati.

Ingressi invertiti

Puoi selezionare gli ingressi che devono essere invertiti.

**Etichetta**

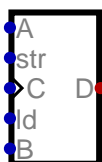
Il nome di questo elemento.

**Rotazione**

L'orientazione dell'elemento nel circuito.

**Memorial del Programma**

Rende questa ROM una memoria del programma. In questo modo può essere programmata da un IDE esterno.



Memoria RAM grafica

## 9.8. RAM Grafica

Utilizzato per mostrare una immagine grafica. Questo elemento si comporta come una RAM. In aggiunta, mostra il suo contenuto in uno schermo grafico. Ogni pixel è rappresentato da un indirizzo di memoria. Il valore memorizzato definisce il colore del pixel, usando una paletta di colori prefissata. Ci sono due buffer dello schermo implementati per supportare il page flipping. L'ingresso B seleziona quale buffer sarà mostrato. Quindi, la memoria totale sarà  $dx * dy * 2$  parole. La paletta di colori utilizzata è strutturata come segue: gli indici 0-9 corrispondono ai colori bianco, nero, rosso, verde, blu, giallo, azzurro, viola, arancio e rosa. Gli indici 32-63 mappano valori di grigio e gli indici 64-127 mappano 64 valori di colore ciascuno con due bit per canale. Questo risulta in una semplice paletta che può essere indirizzata con solo 7 bit. Se l'architettura supporta un indice a 16 bit, dall'indice 0x8000, una modalità di "high-color" con 5 bit per canale può essere usata, per un totale di 32768 colori.

**Ingressi**

A

L'indirizzo di lettura o scrittura dati.

str

Se impostato ad alto quando c'è un fronte di salita del clock, i dati sono memorizzati.

C

Clock

Id

Se impostato ad alto, l'uscita è attivata e i dati sono resi visibili nell'uscita.

B

Seleziona il buffer da visualizzare nello schermo.

**Uscite**

D

La connessione dati bidirezionale.

**Attributi**

Bit di Dati

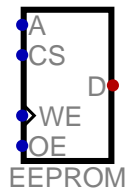
Numero di bit dati utilizzati.

Etichetta

Il nome di questo elemento.

- Larghezza in pixel
  - La larghezza dello schermo in pixel.
- Altezza in pixel
  - L'altezza dello schermo in pixel.
- Rotazione
  - L'orientazione dell'elemento nel circuito.

## 10. Memoria - EEPROM



### 10.1. EEPROM

Un modulo RAM con una connessione bidirezionale per la lettura e scrittura dati. Se l'ingresso chip-select è basso, il componente è disabilitato. I dati sono memorizzati come in una ROM. Vengono quindi preservati quando la simulazione termina e/o è riavviata. Il ciclo di scrittura funziona in questo modo: mettendo chip-select ad alto, il componente diventa selezionato; un fronte di salita nel write-enable scrive l'indirizzo e il fronte di discesa successivo memorizza i dati.

#### Ingressi

- A
  - L'indirizzo di lettura o scrittura dati.
- CS
  - Se impostato ad alto, la EEPROM è abilitata. Altrimenti l'uscita è sempre in alta impedenza.
- WE
  - Se impostato ad alto, i dati sono scritti nella EEPROM.
- OE
  - Se impostato ad alto, il valore memorizzato è presente nell'uscita.

#### Uscite

- D
  - La connessione dati bidirezionale.

#### Attributi

- Bit di Dati
  - Numero di bit dati utilizzati.
- Bit di Indirizzo
  - Numero dei bit di indirizzo utilizzati.
- Etichetta
  - Il nome di questo elemento.
- Ingressi invertiti
  - Puoi selezionare gli ingressi che devono essere invertiti.

**Dati**

I valori memorizzati in questo elemento.

**Rotazione**

L'orientazione dell'elemento nel circuito.

**Formato Numerico**

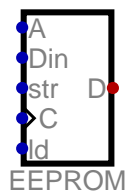
Il formato utilizzato per mostrare i numeri.

**Cifre decimali**

Il numero delle cifre decimali binarie

**Memorial del Programma**

Rende questa ROM una memoria del programma. In questo modo può essere programmata da un IDE esterno.

**10.2. EEPROM, porte separate**

Un modulo EEPROM con ingressi per memorizzazione e uscite per lettura dati separati.

**Ingressi****A**

L'indirizzo da cui leggere o scrivere.

**Din**

I dati da memorizzare nella EEPROM.

**str**

Se questo ingresso è alto e c'è un fronte di salita del clock, i dati sono memorizzati.

**C**

L'ingresso del clock

**Id**

Se questo ingresso è alto l'uscita è attivata e i dati diventano visibili.

**Uscite****D**

Il pin dell'uscita dati

**Attributi****Bit di Dati**

Numero di bit dati utilizzati.

**Bit di Indirizzo**

Numero dei bit di indirizzo utilizzati.

**Etichetta**

Il nome di questo elemento.

**Dati**

I valori memorizzati in questo elemento.

**Rotazione**

L'orientazione dell'elemento nel circuito.

**Formato Numerico**

Il formato utilizzato per mostrare i numeri.

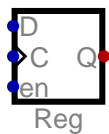
**Cifre decimali**

Il numero delle cifre decimali binarie

**Memorial del Programma**

Rende questa ROM una memoria del programma. In questo modo può essere programmata da un IDE esterno.

## 11. Memoria



### 11.1. Registro

Un componente utilizzato per memorizzare valori. Il numero di bit della parola dati può essere selezionato. Al contrario di un D-Flip-flop, il registro fornisce un ingresso che abilita il clock. Esportabile su VHDL/Verilog.

**Ingressi**

**D**

Pin di ingresso della parola dati da memorizzare.

**C**

Ingresso del clock. Un fronte di salita memorizza il valore presente nel pin D.

**en**

Pin di enable. La memorizzazione di un valore funziona soltanto se questo pin è impostato ad alto.

**Uscite**

**Q**

Restituisce il valore memorizzato.

**Attributi****Bit di Dati**

Numero di bit dati utilizzati.

**Etichetta**

Il nome di questo elemento.

**Ingressi invertiti**

Puoi selezionare gli ingressi che devono essere invertiti.

**Rotazione**

L'orientazione dell'elemento nel circuito.

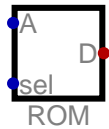
**Program Counter**

Rende questo registro un "program counter". Il valore di questo registro è restituito all'assemblatore o IDE esterno per marcare la linea di codice corrente durante il debug.

Usa come valore di misurazione

Se impostato, il valore è una misurazione e appare nel grafico e nella tabella dati.

In aggiunta, una etichetta deve essere specificata che serve per l'identificazione del valore.



## 11.2. ROM

Un componente di memoria non-volatile. I dati memorizzati possono essere modificati dal dialogo degli attributi. Esportabile su VHDL/Verilog.

Ingressi

A

Questo pin definisce un indirizzo della parola dati da mettere nell'uscita.

sel

Se l'ingresso di selezione è alto, l'uscita è abilitata. Se è basso, l'uscita dati è in alta impedenza.

Uscite

D

La parola dati selezionata se l'ingresso di selezione è alto.

Attributi

Bit di Dati

Numero di bit dati utilizzati.

Bit di Indirizzo

Numero dei bit di indirizzo utilizzati.

Etichetta

Il nome di questo elemento.

Dati

I valori memorizzati in questo elemento.

Rotazione

L'orientazione dell'elemento nel circuito.

Formato Numerico

Il formato utilizzato per mostrare i numeri.

Cifre decimali

Il numero delle cifre decimali binarie

Memorial del Programma

Rende questa ROM una memoria del programma. In questo modo può essere programmata da un IDE esterno.

Ricarica all'avvio del modello

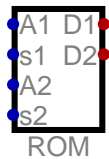
Ricarica il file esadecimale ogni volta che il modello viene avviato.

File

Il file da caricare nella ROM.

Usa il big endian durante l'importazione.

Utilizza l'ordinamento dei byte "big endian" durante l'importazione.



### 11.3. ROM a doppia porta

Un componente di memoria non-volatile. I dati memorizzati possono essere modificati dal dialogo degli attributi.

#### Ingressi

A1

Questo pin definisce l'indirizzo della parola dati da mettere nell'uscita su D1.

s1

Se l'ingresso è alto, l'uscita D1 è attivata. Se è basso, l'uscita dati è in alta impedenza.

A2

Questo pin definisce l'indirizzo della parola dati da mettere nell'uscita D2.

s2

Se l'ingresso è alto, l'uscita D2 è attivata. Se è basso, l'uscita dati è in alta impedenza.

#### Uscite

D1

La parola dati selezionata se l'ingresso S1 è alto.

D2

La parola dati selezionata se l'ingresso di S2 è alto.

#### Attributi

Bit di Dati

Numero di bit dati utilizzati.

Bit di Indirizzo

Numero dei bit di indirizzo utilizzati.

Etichetta

Il nome di questo elemento.

Dati

I valori memorizzati in questo elemento.

Rotazione

L'orientazione dell'elemento nel circuito.

Formato Numerico

Il formato utilizzato per mostrare i numeri.

Cifre decimali

Il numero delle cifre decimali binarie

Memorial del Programma

Rende questa ROM una memoria del programma. In questo modo può essere programmata da un IDE esterno.

Ricarica all'avvio del modello

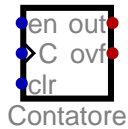
Ricarica il file esadecimale ogni volta che il modello viene avviato.

**File**

Il file da caricare nella ROM.

Usa il big endian durante l'importazione.

Utilizza l'ordinamento dei byte "big endian" durante l'importazione.

**11.4. Contatore**

Un semplice contatore. L'ingresso del clock incrementa il contatore. Può essere resettato a zero con l'ingresso CLR. Il numero di bit può essere impostato nel dialogo degli attributi. Esportabile su VHDL/Verilog.

**Ingressi**

en

Se impostato ad alto, il contatore è abilitato.

C

L'ingresso del clock. Un fronte di salita incrementa il contatore.

clr

Reset sincrono del contatore se impostato ad alto.

**Uscite**

out

Restituisce il valore del contatore.

ovf

Uscita dell'overflow. Questo pin è impostato ad un valore alto se il contatore è al suo valore massimo e anche l'ingresso è impostato ad alto.

**Attributi**

Bit di Dati

Numero di bit dati utilizzati.

Ingressi invertiti

Puoi selezionare gli ingressi che devono essere invertiti.

Etichetta

Il nome di questo elemento.

Rotazione

L'orientazione dell'elemento nel circuito.

Usa come valore di misurazione

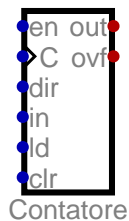
Se impostato, il valore è una misurazione e appare nel grafico e nella tabella dati.

In aggiunta, una etichetta deve essere specificata che serve per l'identificazione del valore.

Program Counter

Rende questo registro un "program counter". Il valore di questo registro è restituito all'assemblatore o IDE esterno per marcare la linea di codice corrente durante il debug.





### 11.5. Contatore con preset

Un contatore il cui valore può essere impostato. In aggiunta, un valore massimo e una direzione di conteggio possono essere specificate. Esportabile su VHDL/Verilog.

#### Ingressi

- en  
Se impostato ad alto il contatore è abilitato.
- C  
L'ingresso del clock. Un fronte di salita incrementa o decrementa il contatore.
- dir  
Specifica la direzione del conteggio. Un valore basso significa un conteggio in avanti.
- in  
Questa parola dati è memorizzata nel contatore quando "ld" è impostato.
- ld  
Se impostato, il valore all'ingresso "in" è memorizzato nel contatore al prossimo segnale di clock.
- clr  
Reset sincrono del contatore se impostato ad alto.

#### Uscite

- out  
Restituisce il valore del contatore.
- ovf  
Uscita dell'overflow. Questo viene impostato ad alto se l'ingresso "en" è alto e se il contatore raggiunge il suo massimo valore quando conta verso l'alto oppure se ha raggiunto lo zero contando verso il basso.

#### Attributi

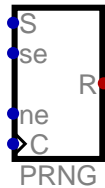
- Bit di Dati  
Numero di bit dati utilizzati.
- Valore Massimo  
Se uno zero è inserito, il valore massimo è usato (tutti i bit ad uno).
- Ingressi invertiti  
Puoi selezionare gli ingressi che devono essere invertiti.
- Etichetta  
Il nome di questo elemento.
- Rotazione  
L'orientazione dell'elemento nel circuito.

Usa come valore di misurazione

Se impostato, il valore è una misurazione e appare nel grafico e nella tabella dati. In aggiunta, una etichetta deve essere specificata che serve per l'identificazione del valore.

Program Counter

Rende questo registro un "program counter". Il valore di questo registro è restituito all'assemblatore o IDE esterno per marcare la linea di codice corrente durante il debug.



## 11.6. Generatore di Numeri Casuali

Può essere utilizzato per generare numeri casuali. Quando la simulazione viene avviata, il generatore è reinizializzato così che una nuova sequenza di numeri pseudo-casuali verrà generata. Il generatore può essere inizializzato durante una simulazione già avviata con un seme per generare una sequenza di numeri predefinita e deterministica.

Ingressi

- S  
Nuovo valore del seme per il generatore.
- se  
Se impostato, il generatore è reinizializzato con il nuovo seme al prossimo fronte di salita del clock.
- ne  
Se impostato, un nuovo numero casuale è impostato nell'uscita al prossimo fronte di salita del clock.
- C  
L'ingresso del clock.

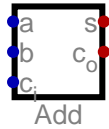
Uscite

- R  
Uscita del numero pseudo-casuale.

Attributi

- Bit di Dati  
Numero di bit dati utilizzati.
- Etichetta  
Il nome di questo elemento.
- Rotazione  
L'orientazione dell'elemento nel circuito.

## 12. Aritmetica



### 12.1. Sommatore

Un componente per semplici addizioni. Effettua l'addizione tra i due ingressi numerici interi A e B ( $A+B$ ). Il risultato sarà incrementato di uno se l'ingresso del riporto è impostato. Esportabile su VHDL/Verilog.

Ingressi

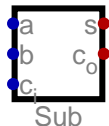
- a Primo ingresso da sommare.
- b Secondo ingresso da sommare.
- c\_i Ingresso del riporto. Se impostato questo fa incrementare il risultato di uno.

Uscite

- s Il risultato dell'addizione.
- c\_o Uscita del riporto. Impostato se si è verificato un overflow.

Attributi

- Etichetta Il nome di questo elemento.
- Bit di Dati Numero di bit dati utilizzati.
- Rotazione L'orientazione dell'elemento nel circuito.



### 12.2. Sottrattore

Un componente per semplici sottrazioni. Sottrae numeri binari negli ingressi A e B ( $A-B$ ). Se l'ingresso di riporto è impostato ad alto, allora il risultato è decrementato di uno. Esportabile su VHDL/Verilog.

### Ingressi

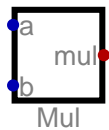
- a  
Ingresso A per la sottrazione.
- b  
Ingresso B per la sottrazione.
- c\_i  
Ingresso del riporto. Se impostato questo fa decrementare il risultato di uno.

### Uscite

- s  
Il risultato della sottrazione.
- c\_o  
Impostato ad alto se si è verificato un underflow.

### Attributi

- Etichetta  
Il nome di questo elemento.
- Bit di Dati  
Numero di bit dati utilizzati.
- Rotazione  
L'orientazione dell'elemento nel circuito.



## 12.3. Moltiplicatore

Un componente per la moltiplicazione. Moltiplica i numeri interi negli ingressi A e B. Esportabile su VHDL/Verilog.

### Ingressi

- a  
Ingresso A per la moltiplicazione.
- b  
Ingresso B per la moltiplicazione.

### Uscite

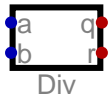
- mul  
Uscita per il risultato della moltiplicazione.

### Attributi

- Etichetta  
Il nome di questo elemento.
- Operazione con Segno  
Se selezionato l'operazione è effettuata con il segno (complemento a due).
- Bit di Dati  
Numero di bit dati utilizzati.

**Rotazione**

L'orientazione dell'elemento nel circuito.

**12.4. Divisore**

Un componente per la divisione. Divide il numer intero applicato all'ingresso A per l'altro numero intero applicato all'ingresso B. Se il divisore è zero, è diviso per uno. Nella divisione con segno, il resto è sempre positivo.

**Ingressi**

- a  
dividendo
- b  
divisore

**Uscite**

- q  
quoziente
- r  
resto

**Attributi****Etichetta**

Il nome di questo elemento.

**Bit di Dati**

Numero di bit dati utilizzati.

**Operazione con Segno**

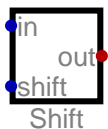
Se selezionato l'operazione è effettuata con il segno (complemento a due).

**Resto sempre positivo**

Se impostato, il resto di una operazione con segno sarà sempre positivo.

**Rotazione**

L'orientazione dell'elemento nel circuito.

**12.5. Barrel shifter**

Un componente per lo scorrimento dei bit. Scorre i bit del valore di ingresso per un numero specificato dall'ingresso dello scorrimento bit.

**Ingressi**

in

L'ingresso dati contenente i bit da scorrere.

shift

L'ingresso che indica di quanti bit scorrere i dati.

**Uscite**

out

L'uscita con il valore sul quale è stato applicato lo scorrimento dei bit.

**Attributi**

Etichetta

Il nome di questo elemento.

Bit di Dati

Numero di bit dati utilizzati.

L'ingresso del registro a scorrimento ha il segno

L'ingresso dati del registro a scorrimento ha il formato in complemento a due

Direzione

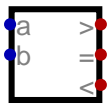
Imposta la direzione.

Modalità

Modalità del registro a scorrimento

Rotazione

L'orientazione dell'elemento nel circuito.

**12.6. Comparatore**

Un componente per comparare valori binari. Compara i numeri binari nei due ingressi A e B ed imposta le rispettive uscite. Esportabile su VHDL/Verilog.

**Ingressi**

a

Ingresso A da comparare.

b

Ingresso B da comparare.

**Uscite**

&gt;

L'uscita è alta se l'ingresso A è più grande dell'ingresso B.

=

L'uscita è alta se i due ingressi sono uguali.

&lt;

L'uscita è alta se l'ingresso A è più piccolo dell'ingresso B.

**Attributi**

Etichetta

Il nome di questo elemento.

Bit di Dati

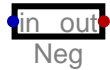
Numero di bit dati utilizzati.

Operazione con Segno

Se selezionato l'operazione è effettuata con il segno (complemento a due).

Rotazione

L'orientazione dell'elemento nel circuito.



## 12.7. Negazione

Negazione nel complemento a due. Esportabile su VHDL/Verilog.

Ingressi

in

Ingresso della parola dati da negare con il complemento a due.

Uscite

out

Restituisce il valore della negazione con complemento a due.

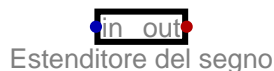
Attributi

Bit di Dati

Numero di bit dati utilizzati.

Rotazione

L'orientazione dell'elemento nel circuito.



## 12.8. Estenditore del segno

Incrementa il numero di bit di un numero intero con segno mantenendo il segno. Se l'ingresso è un singolo bit, quel bit sarà impostato per ogni bit nell'uscita. Esportabile su VHDL/Verilog.

Ingressi

in

Valore di ingresso. Il numero di bit nell'ingresso deve essere più piccolo del numero di bit nell'uscita!

Uscite

out

Valore di ingresso esteso. Il numero di bit nell'ingresso deve essere più piccolo del numero di bit nell'uscita!

Attributi

Etichetta

Il nome di questo elemento.

Numero dei Bit in Ingresso

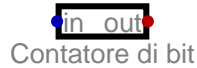
Il numero dei bit in uscita deve essere più grande di quello in ingresso.

Numero di Bit in Uscita

Il numero dei bit in uscita deve essere più grande di quello in ingresso.

Rotazione

L'orientazione dell'elemento nel circuito.



## 12.9. Contatore di bit

Restituisce il numero di bit ad 1 nel valore di ingresso.

Ingressi

in

L'ingresso da cui i bit a 1 sono conteggiati.

Uscite

out

L'uscita con il numero di bit ad 1.

Attributi

Bit di Dati

Numero di bit dati utilizzati.

Rotazione

L'orientazione dell'elemento nel circuito.

## 13. Interruttori



### 13.1. Interruttore

Un semplice interruttore. Non c'è nessun ritardo di porta. Un cambiamento di segnale è propagato immediatamente.

Uscite

A1

Una delle connessioni all'interruttore.

B1

Una delle connessioni all'interruttore.

Attributi

Bit di Dati

Numero di bit dati utilizzati.

Etichetta

Il nome di questo elemento.



Numero di poli

Numero dei poli disponibili.

Chiuso

Imposta lo stato iniziale dell'interruttore.

Rotazione

L'orientazione dell'elemento nel circuito.

Specchio

Specchia il componente nel circuito.

L'interruttore si comporta come un ingresso

Se il modello è analizzato, l'interruttore si comporta come un ingresso dove "aperto" corrisponde a zero e "chiuso" corrisponde ad uno.



### 13.2. Interruttore a Doppio Contatto

Un semplice interruttore. Non c'è nessun ritardo di porta. Un cambiamento di segnale è propagato immediatamente.

Uscite

A1

Una delle connessioni all'interruttore.

B1

Una delle connessioni all'interruttore.

C1

Una delle connessioni all'interruttore.

Attributi

Bit di Dati

Numero di bit dati utilizzati.

Etichetta

Il nome di questo elemento.

Numero di poli

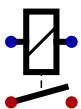
Numero dei poli disponibili.

Rotazione

L'orientazione dell'elemento nel circuito.

Specchio

Specchia il componente nel circuito.



### 13.3. Relay

Un relay è un interruttore che può essere controllato da una bobina. Se una corrente scorre nella bobina, l'interruttore è chiuso o aperto. Non c'è nessun diodo di protezione quindi la direzione della corrente non importa. L'interruttore è attuato se gli ingressi hanno valori differenti. Il relay si comporta in maniera simile ad una porta XOR.

### Ingressi

in1

Uno degli ingressi per controllare il relay.

in2

Uno degli ingressi per controllare il relay.

### Uscite

A1

Una delle connessioni all'interruttore.

B1

Una delle connessioni all'interruttore.

### Attributi

Bit di Dati

Numero di bit dati utilizzati.

Etichetta

Il nome di questo elemento.

Numero di poli

Numero dei poli disponibili.

Il relay è normalmente chiuso.

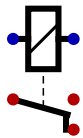
Se impostato il relay è chiuso se l'ingresso è basso.

Rotazione

L'orientazione dell'elemento nel circuito.

Specchio

Specchia il componente nel circuito.



## 13.4. Relay a Doppio Contatto

Un relay è un interruttore che può essere controllato da una bobina. Se una corrente scorre nella bobina, l'interruttore è chiuso o aperto. Non c'è nessun diodo di protezione quindi la direzione della corrente non importa. L'interruttore è attuato se gli ingressi hanno valori differenti. Il relay si comporta in maniera simile ad una porta XOR.

### Ingressi

in1

Uno degli ingressi per controllare il relay.

in2

Uno degli ingressi per controllare il relay.

**Uscite**

- A1  
Una delle connessioni all'interruttore.
- B1  
Una delle connessioni all'interruttore.
- C1  
Una delle connessioni all'interruttore.

**Attributi**

- Bit di Dati  
Numero di bit dati utilizzati.
- Etichetta  
Il nome di questo elemento.
- Numero di poli  
Numero dei poli disponibili.
- Rotazione  
L'orientazione dell'elemento nel circuito.
- Specchio  
Specchia il componente nel circuito.

**13.5. FET a Canale P**

Transistore ad effetto di campo a canale P. Il bulk è connesso alla corsia di tensione positiva ed il transistore è simulato senza un diodo di body.

**Ingressi**

- G  
Gate

**Uscite**

- S  
Source
- D  
Drain

**Attributi**

- Bit di Dati  
Numero di bit dati utilizzati.
- Unidirezionale  
I transistori unidirezionali propagano un segnale soltanto dal source al drain. Sono molto più veloci da simulare piuttosto che quelli bidirezionali. Visto che non c'è nessun feedback da drain a source, in questa modalità il transistore non fa corto circuito con i cavi connessi quando conduce. Quindi, questa modalità è necessaria per simulare certi circuiti CMOS.
- Etichetta  
Il nome di questo elemento.

Rotazione

L'orientazione dell'elemento nel circuito.

Specchio

Specchia il componente nel circuito.



### 13.6. FET a Canale N

Transistore ad effetto di campo a canale N. Il bulk è connesso alla massa ed il transistore è simulato senza un diodo di body.

Ingressi

G

Gate

Uscite

D

Drain

S

Source

Attributi

Bit di Dati

Numero di bit dati utilizzati.

Unidirezionale

I transistori unidirezionali propagano un segnale soltanto dal source al drain. Sono molto più veloci da simulare piuttosto che quelli bidirezionali. Visto che non c'è nessun feedback da drain a source, in questa modalità il transistore non fa corto circuito con i cavi connessi quando conduce. Quindi, questa modalità è necessaria per simulare certi circuiti CMOS.

Etichetta

Il nome di questo elemento.

Rotazione

L'orientazione dell'elemento nel circuito.

Specchio

Specchia il componente nel circuito.



### 13.7. Fusibile

Un fusibile è usato per costruire una memoria programmabile una sola volta.

Uscite

out1

Una delle connessioni all'interruttore.

out2

Una delle connessioni all'interruttore.

## Attributi

## Programmato

Se impostato, il diodo è impostato come "esploso" o "programmato". In un FET a floating gate, la floating gate è caricata. Puoi cambiare questa impostazione con il tasto [P].

## Rotazione

L'orientazione dell'elemento nel circuito.

**13.8. Diodo a VDD**

Un semplice diodo unidirezionale, usato per portare un cavo a VDD. Viene utilizzato per implementare una OR cablata. Quindi è necessario connettere un resistore pull-down all'uscita dei diodi. Nella simulazione i diodi si comportano come una porta attiva con una tabella di verità trivalente: Se l'ingresso è alto, l'uscita è anche alta. In tutti gli altri casi (ingresso basso o ad alta impedenza) l'uscita è in alta impedenza. Quindi due diodi anti-paralleli possono tenersi nello stato alto a vicenda, cosa che non è possibile con dei diodi reali. Questo è un diodo ideale: non c'è caduta di tensione in un diodo a polarizzazione diretta.

## Ingressi

in

Se l'ingresso è alto anche l'uscita lo sarà. In tutti gli altri casi l'uscita è in alta impedenza.

## Uscite

out

Se l'ingresso è alto anche l'uscita lo sarà. In tutti gli altri casi l'uscita è in alta impedenza.

## Attributi

## Programmato

Se impostato, il diodo è impostato come "esploso" o "programmato". In un FET a floating gate, la floating gate è caricata. Puoi cambiare questa impostazione con il tasto [P].

## Rotazione

L'orientazione dell'elemento nel circuito.

**13.9. Diodo a Massa**

Un semplice diodo unidirezionale, usato per portare un cavo a massa. Viene utilizzato per implementare una AND cablata. Quindi è necessario connettere un resistore pull-up all'uscita dei diodi. Nella simulazione i diodi si comportano come una porta attiva con una tabella di verità trivalente: Se l'ingresso è basso, l'uscita è anche bassa. In tutti gli altri casi (ingresso alto o ad alta impedenza) l'uscita è in alta impedenza. Quindi due diodi anti-

paralleli possono tenersi nello stato alto a vicenda, cosa che non è possibile con dei diodi reali. Questo è un diodo ideale: non c'è caduta di tensione in un diodo a polarizzazione diretta.

#### Ingressi

in

Se l'ingresso è basso l'uscita lo è anche. In tutti gli altri casi l'uscita è in alta impedenza.

#### Uscite

out

Se l'ingresso è basso l'uscita lo è anche. In tutti gli altri casi l'uscita è in alta impedenza.

#### Attributi

##### Programmato

Se impostato, il diodo è impostato come "esploso" o "programmato". In un FET a floating gate, la floating gate è caricata. Puoi cambiare questa impostazione con il tasto [P].

##### Rotazione

L'orientazione dell'elemento nel circuito.



### 13.10. FET a Canale P con Floating Gate

Transistore ad effetto di campo a canale P con floating gate. Il bulk è connesso alla massa ed il transistore è simulato senza un diodo di body.

#### Ingressi

G

Gate

#### Uscite

S

Source

D

Drain

#### Attributi

##### Bit di Dati

Numero di bit dati utilizzati.

##### Etichetta

Il nome di questo elemento.

##### Programmato

Se impostato, il diodo è impostato come "esploso" o "programmato". In un FET a floating gate, la floating gate è caricata. Puoi cambiare questa impostazione con il tasto [P].

Rotazione

L'orientazione dell'elemento nel circuito.

Specchio

Specchia il componente nel circuito.



### 13.11. FET a Canale N con Floating Gate

Transistore ad effetto di campo a canale P con floating gate. Il bulk è connesso alla massa ed il transistore è simulato senza un diodo di body. Se c'è una carica immagazzinata nella floating gate, il FET non conduce anche se il gate è impostato ad alto.

Ingressi

G

Gate

Uscite

D

Drain

S

Source

Attributi

Bit di Dati

Numero di bit dati utilizzati.

Etichetta

Il nome di questo elemento.

Programmato

Se impostato, il diodo è impostato come "esploso" o "programmato". In un FET a floating gate, la floating gate è caricata. Puoi cambiare questa impostazione con il tasto [P].

Rotazione

L'orientazione dell'elemento nel circuito.

Specchio

Specchia il componente nel circuito.



### 13.12. Transmission gate

Una vera transmission gate è costruita da due soli transistor. Quindi, è spesso usato per risparmiare transistor durante l'implementazione su silicio.

#### Ingressi

- S  
ingresso di controllo.
- $\neg S$   
ingresso di controllo invertito

#### Uscite

- A  
ingresso A
- B  
ingresso B

#### Attributi

- Bit di Dati  
Numero di bit dati utilizzati.
- Rotazione  
L'orientazione dell'elemento nel circuito.

## 14. Altro

### Test

#### 14.1. Test case

Descrive un test case. In un test case puoi descrivere come un circuito dovrebbe comportarsi. Si può quindi controllare automaticamente che il comportamento corrisponda alla descrizione fornita. Se c'è una discrepanza, un messaggio di errore viene mostrato. Esportabile su VHDL/Verilog.

#### Attributi

- Etichetta  
Il nome di questo elemento.
- Dati di Test  
La descrizione del test case. Dettagli della sintassi possono essere trovati nel dialogo di aiuto dell'editor dei dati di test.
- Abilitato  
Abilita o disabilita questo componente.

## 15. Altro - Decorazioni

### Testo

#### 15.1. Testo

Mostra un testo all'interno del circuito. Non ha alcun effetto sulla simulazione. Il testo può essere cambiato nel dialogo degli attributi.



**Attributi****Descrizione**

Una breve descrizione di questo elemento e del suo utilizzo.

**Dimensione del Font**

Imposta la dimensione del font per questo testo.

**Rotazione**

L'orientazione dell'elemento nel circuito.

**Orientamento**

Posizione delle coordinate relative al testo.

**Allinea alla Griglia**

Se impostato, il componente è allineato alla griglia.

**Testo****15.2. Rettangolo**

Mostra un rettangolo nel circuito.

**Attributi****Etichetta**

Il nome di questo elemento.

**Larghezza**

Larghezza in unità di griglia

**Altezza**

Altezza in unità di griglia

**Dimensione del Font**

Imposta la dimensione del font per questo testo.

**Testo Contenuto**

Testo contenuto all'interno del rettangolo.

**Testo in basso**

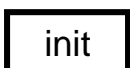
Testo in fondo al rettangolo.

**Testo a destra**

Testo a destra del rettangolo.

**Allinea alla Griglia**

Se impostato, il componente è allineato alla griglia.

**16. Altro - Generici****16.1. Inizializzazione Generica**

Codice che è eseguito per avviare un circuito generico direttamente. Se un circuito generico è avviato direttamente, questo componente deve essere presente. Esportabile su VHDL/Verilog.

**Attributi****Etichetta**

Il nome di questo elemento.

**Abilitato**

Abilita o disabilita questo componente.

**Parametrizzazione Generica**

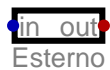
Dichiarazioni utilizzate per rendere generico il circuito.

**Codice****16.2. Codice**

Codice che può essere eseguito quando un circuito generico viene reso concreto. Può essere usato, per esempio, per aggiungere componenti o cavi ad un circuito. Esportabile su VHDL/Verilog.

**Attributi****Parametrizzazione Generica**

Dichiarazioni utilizzate per rendere generico il circuito.

**17. Altro - VHDL/Verilog****17.1. Esterno**

Un componente che esegue un processo esterno per il calcolo della funzione logica. Può essere usato per specificare il comportamento di un componente da VHDL o Verilog. L'attuale comportamento della simulazione deve essere fatto da un simulatore esterno. Al momento soltanto il simulatore VHDL "ghdl" ed il simulatore Verilog "Icarus Verilog" sono supportati. L'etichetta del componente deve essere corrispondente al nome dell'entità o modulo! Esportabile su VHDL/Verilog.

**Ingressi**

in

**Uscite**

out

**Attributi****Etichetta**

Il nome di questo elemento.

**Larghezza**

Larghezza del simbolo se questo circuito è utilizzato come un componente in un altro circuito.

**Ingressi**

Gli ingressi del processo esterno. Questa è una lista di nomi di segnali separati da virgola. Per ogni nome, un numero di bit separato da un due punti (:) può essere specificato. Gli ingressi di un sommatore ad 8 bit potrebbero essere quindi specificati come "a:8,b:8,c\_in".

**Uscite**

Le uscite del processo esterno. Questa è una lista di nomi di segnali separati da virgola. Per ogni nome, un numero di bit separato da un due punti (:) può essere specificato. Le uscite di un sommatore ad 8 bit potrebbero essere quindi specificate come "s:8,c\_out".

**Codice del Programma**

Il codice del programma da eseguire dall'applicazione esterna.

**Applicazione**

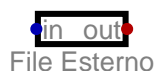
Definisce quale applicazione utilizzare.

**Opzioni GHDL**

Opzioni che sono utilizzate per tutti i passi di processo da GHDL.

**Opzioni IVerilog.**

Opzioni che sono utilizzate per tutti i passi di processo da IVerilog.

**17.2. File Esterno**

Un componente che esegue un processo esterno per il calcolo della funzione logica. Può essere usato per specificare il comportamento di un componente da VHDL o Verilog. L'attuale comportamento della simulazione deve essere fatto da un simulatore esterno. Al momento soltanto il simulatore VHDL "ghdl" ed il simulatore Verilog "Icarus Verilog" sono supportati. L'etichetta del componente deve essere corrispondente al nome dell'entità o modulo! Esportabile su VHDL/Verilog.

**Ingressi**

in

**Uscite**

out

**Attributi****Etichetta**

Il nome di questo elemento.

**Larghezza**

Larghezza del simbolo se questo circuito è utilizzato come un componente in un altro circuito.

**Ingressi**

Gli ingressi del processo esterno. Questa è una lista di nomi di segnali separati da virgola. Per ogni nome, un numero di bit separato da un due punti (:) può essere specificato. Gli ingressi di un sommatore ad 8 bit potrebbero essere quindi specificati come "a:8,b:8,c\_in".

**Uscite**

Le uscite del processo esterno. Questa è una lista di nomi di segnali separati da virgola. Per ogni nome, un numero di bit separato da un due punti (:) può essere specificato. Le uscite di un sommatore ad 8 bit potrebbero essere quindi specificate come "s:8,c\_out".

**Codice del Programma**

Il file contenente il codice del programma da eseguire dall'applicazione esterna.

**Applicazione**

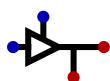
Definisce quale applicazione utilizzare.

**Opzioni GHDL**

Opzioni che sono utilizzate per tutti i passi di processo da GHDL.

**Opzioni IVerilog.**

Opzioni che sono utilizzate per tutti i passi di processo da IVerilog.

**17.3. Controllo del Pin**

Logica di controllo per un pin bidirezionale. Questo componente è necessario soltanto nel contesto di una generazione VHDL o Verilog, per creare una porta HDL bidirezionale. Se non vuoi usare una porta di I/O bidirezionale su una FPGA, non usare questo componente! Questo componente non può essere usato in un circuito incorporato, ma solo nel circuito principale (ovvero quello di più alto livello). Esportabile su VHDL/Verilog.

**Ingressi**

wr

Il dato da mettere nell'uscita.

oe

Attiva l'uscita.

**Uscite**

rd

Il dato da leggere.

pin

Il connettore per il pin. Solo una singola uscita deve essere collegata.

**Attributi**

Bit di Dati

Numero di bit dati utilizzati.

Rotazione

L'orientazione dell'elemento nel circuito.

Specchio

Specchia il componente nel circuito.

## 18. Altro



### 18.1. Alimentazione

Non ha alcuna funzionalità. Si assicura che alimentazione e massa siano connessi. Può essere usato nei circuiti 74xx per generare i pin per l'alimentazione, che sono testati per una corretta cablatura.

Ingressi

VDD

Deve essere connesso all'alimentazione!

GND

Deve essere connesso alla massa!

Attributi

Etichetta

Il nome di questo elemento.

Rotazione

L'orientazione dell'elemento nel circuito.



### 18.2. Splitter Bidirezionale

Può essere usato per i bus dati e semplifica in particolare la costruzione di un modulo di memoria in un package DIL, visto che l'implementazione del bus dati è semplificata.

Ingressi

OE

Quando impostato, il valore al terminale dati comune D è messo nell'uscita binaria D[i], altrimenti i bit D[i] sono inseriti nell'uscita comune D.

Uscite

D

La connessione dati comune.

D0

Il bit dati 0 dello splitter del bus.

Attributi

Bit di Dati

Numero di bit dati utilizzati.

Rotazione

L'orientazione dell'elemento nel circuito.

**Specchio**

Specchia il componente nel circuito.

**Spargimento**

Configura lo spreading degli ingressi ed uscite nel circuito.

**18.3. Reset**

L'uscita di questo componente è mantenuta alta durante l'inizializzazione del circuito. Dopo che il circuito si è stabilizzato, l'uscita diventa bassa. Se l'uscita viene invertita si comporta nel modo opposto. Esportabile su VHDL/Verilog.

**Uscite****Reset**

Uscita di Reset.

**Attributi****Etichetta**

Il nome di questo elemento.

**Uscita invertita**

Se selezionato l'uscita è invertita.

**Rotazione**

L'orientazione dell'elemento nel circuito.

**18.4. Interruzione**

Se questo componente è usato nel circuito, il bottone "Avvia fino all'interruzione" tra "Avvia" ed "Interrompi" è abilitato. Il bottone fa funzionare il circuito fino a quando un fronte di salita nel componente viene rilevato. Questo elemento può essere usato per effettuare il debugging del circuito in qualsiasi punto di interruzione (breakpoint). In questo modo una istruzione in assembly BRK può essere implementata. Questo consente l'esecuzione di un programma fino al raggiungimento di questo comando. Questa funzione può essere utilizzata soltanto se il clock a tempo reale è disattivato.

**Ingressi****brk**

Ferma la simulazione veloce se un fronte di salita viene individuato.

**Attributi****Etichetta**

Il nome di questo elemento.

**Abilitato**

Abilita o disabilita questo componente.

**Cicli di Timeout**

Se questo numero di cicli di timeout è raggiunto senza un segnale di interruzione, un errore viene generato.

Rotazione

L'orientazione dell'elemento nel circuito.



### 18.5. Fermata

Un fronte di salita in questo componente ferma la simulazione. Ha lo stesso effetto del bottone "Stop" nella barra degli strumenti.

Ingressi

stop

Un fronte di salita ferma la simulazione.

Attributi

Etichetta

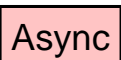
Il nome di questo elemento.

Ingressi invertiti

Puoi selezionare gli ingressi che devono essere invertiti.

Rotazione

L'orientazione dell'elemento nel circuito.



### 18.6. Temporizzazione Asincrona

Consente la configurazione della temporizzazione di un circuito sequenziale asincrono come per esempio una pipeline Muller. Il circuito deve essere avviato in modalità di porta a passo singolo e deve essere in grado di raggiungere uno stato stabile durante l'avvio. Il circuito sequenziale può quindi essere avviato interattivamente o con una porta di reset. Non viene consentito l'utilizzo di un clock in questa modalità.

Attributi

Avvia il clock a tempo reale

Se abilitato, il clock a tempo reale è avviato quando il circuito è avviato

Frequenza/Hz

La frequenza in tempo reale utilizzata per questo clock.

## E Libreria

- 27c801:** 8 Mbit (1Mb x 8) UV EPROM
- 28c010:** 1-Megabit (128K x 8) Paged Parallel EEPROM; DATA Polling for End of Write Detection not implemented!
- 28c16:** 16K (2K x 8) Parallel EEPROM; DATA Polling for End of Write Detection not implemented!
- 28c64:** 64K (8K x 8) Parallel EEPROM; DATA Polling for End of Write Detection not implemented!
- 28c256:** 256K (32K x 8) Paged Parallel EEPROM; DATA Polling for End of Write Detection not implemented!
- 28c512:** 512K-Bit (64K x 8) CMOS Parallel EEPROM; DATA Polling for End of Write Detection not implemented!
- 7400:** quad 2-input NAND gate
- 7401:** quad 2-input NAND gate with open-collector outputs
- 7402:** quad 2-input NOR gate
- 7403:** quad 2-input NAND gate with open-collector outputs, different pinout than 7401
- 7404:** hex inverter
- 7405:** hex inverter, open-collector output
- 7406:** hex inverter buffer, open-collector output
- 7407:** hex buffer, open-collector output
- 7408:** quad 2-input AND gate
- 7409:** quad 2-input AND gate with open-collector outputs
- 7410:** triple 3-input NAND gate
- 7411:** triple 3-input AND gate
- 7412:** triple 3-input NAND gate with open-collector outputs
- 7413:** dual 4-input NAND gate, Schmitt trigger
- 7414:** hex inverter, Schmitt trigger
- 7415:** triple 3-input AND gate with open-collector outputs
- 7416:** hex inverter buffer, open-collector output, same as 7406
- 7417:** hex buffer, open-collector output, same as 7407
- 7420:** dual 4-input NAND gate
- 7421:** dual 4-input AND gate
- 7425:** dual 4-input NOR gate with strobe
- 7427:** triple 3-input NOR gate
- 7428:** quad 2-input NOR buffer
- 7430:** 8-input NAND gate
- 7432:** quad 2-input OR gate
- 7434:** hex buffer
- 7440:** dual 4-input NAND buffer
- 7442:** 4-line BCD to 10-line decimal decoder
- 7447:** BCD to 7-segment decoder, active low
- 7448:** BCD to 7-segment decoder, active high
- 7451:** 2-input/3-input AND-NOR gate
- 7454:** 2-3-2-3-line AND NOR gate
- 7455:** 2 wide 4-input AND-NOR gate
- 7458:** dual AND OR gate
- 7474:** dual D-flip-flop
- 7476:** dual J-K flip-flops with preset and clear
- 7480:** Gated Full Adder with Complementary Inputs and Complementary Sum Outputs



**7482:** 2-bit binary full adder  
**7483:** 4-bit binary full adder  
**7483Real:** 4-bit binary full adder, real gates  
**7485:** 4-bit comparator  
**7486:** quad 2-input XOR gate  
**7489:** 64-bit RAM  
**7490:** asynchronous two - five - decimal addition counter  
**7493:** 4-bit Binary Counter. Connect QA to CKB and clock on CKA for full 4-bit counter.  
Connect QB to R1 and QD to R2 for a BCD counter.  
**74107:** dual J-K flip-flops with clear  
**74109:** Dual J-NOT-K flip-flop with set and reset; positive-edge-trigger  
**74112:** Dual J-K negative-edge-triggered flip-flop, clear and preset  
**74116:** dual 4-bit D-type latches  
**74125:** Quadruple bus buffer gates with 3-state outputs (active low output enable)  
**74126:** Quadruple bus buffer gates with 3-state outputs (active high output enable)  
**74133:** 13-input NAND gate  
**74138:** 3-line to 8-line decoder/demultiplexer, inverted out  
**74139:** dual 2-line to 4-line decoder/demultiplexer  
**74147:** 10-line to 4-line priority encoder  
**74148:** 8-line to 3-Line priority encoder  
**74150:** 4-line to 16-line data selectors/multiplexers  
**74151:** 3-line to 8-line data selectors/multiplexers  
**74153:** dual 4-line to 1-line data selectors/multiplexers  
**74154:** 4-line to 16-line decoders/demultiplexers  
**74157:** quad 2-line to 1-line data selectors/multiplexers  
**74160:** decimal synchronous counter, async clear  
**74161:** hex synchronous counter, async clear  
**74162:** decimal synchronous counter  
**74162Real:** decimal synchronous counter, real gates  
**74163:** hex synchronous counter  
**74164:** 8-bit parallel-out serial shift register, asynchronous clear  
**74165:** parallel-load 8-bit shift register  
**74166:** 8-Bit Parallel-In/Serial-Out Shift Register  
**74173:** quad 3-state D flip-flop with common clock and reset  
**74174:** hex D-flip-flop  
**74175:** quad D-flip-flop  
**74181:** 4-bit arithmetic logic unit  
**74182:** look-ahead carry generator  
**74189:** 64-Bit Random Access Memory with 3-STATE Outputs  
**74190:** Presettable synchronous 4-bit bcd up/down counter  
**74191:** Presettable synchronous 4-bit binary up/down counter  
**74193:** Synchronous 4-Bit Up/Down Binary Counter with Dual Clock  
**74194:** 4-Bit Bidirectional Universal Shift Register  
**74194real:** 4-Bit Bidirectional Universal Shift Register, Databook implementation.  
**74198:** 8-bit shift register  
**74238:** 3-line to 8-line decoder/demultiplexer  
**74244:** octal 3-state buffer/line driver/line receiver  
**74245:** octal bus transceivers with 3-state outputs  
**74247:** BCD to 7-segment decoder, active low, tails on 6 and 9  
**74248:** BCD to 7-segment decoder, active high, tails on 6 and 9

**74253:** dual tri state 4-line to 1-line data selectors/multiplexers  
**74257:** quad 2-line to 1-line data selectors/multiplexers (3-state output)  
**74260:** dual 5-input NOR gate  
**74266:** quad 2-input XNOR gate with open collector outputs  
**74273:** octal D-type flip-flop with clear  
**74280:** 9 bit Odd-Even Parity Generator-Checker  
**74283:** 4-bit binary full adder, alternative pinning  
**74299:** 8-Input Universal Shift/Storage Register with Common Parallel I/O Pins  
**74373:** octal transparent latches  
**74374:** octal positive-edge-triggered flip-flops  
**74377:** Octal D Flip-Flop with enable  
**74381:** 4-Bit Arithmetic Logic Unit with high-speed expansion  
**74382:** 4-Bit Arithmetic Logic Unit with ripple carry output  
**74540:** octal buffer/line driver, inverted  
**74541:** octal buffer/line driver  
**74573:** octal transparent latches, different pinout compared to 74373  
**74574:** octal positive-edge-triggered flip-flops, different pinout compared to 74374  
**74590:** 8-bit binary counter with tri-state output registers  
**74595:** 8-Bit Shift Registers with 3-State Output Registers  
**74670:** 3-state 4-by-4 Register File  
**74682:** 8-bit digital comparator  
**74688:** 8-bit identity comparator  
**74779:** 8-Bit Bidirectional Binary Counter with 3-STATE Outputs  
**74804:** hex 2-input NAND gate <https://www.ti.com/lit/ds/symlink/sn74as804b.pdf>  
**74805:** hex 2-input NOR gate <http://www.ti.com/lit/ds/symlink/sn54as805b.pdf>  
**74808:** hex 2-input AND gate <http://www.ti.com/lit/ds/symlink/sn54as808b.pdf>  
**74832:** hex 2-input OR gate <http://www.ti.com/lit/ds/symlink/sn54as832b.pdf>  
**744002:** dual 4-input NOR gate  
**744017:** Johnson decade counter with 10 decoded outputs  
**744075:** triple 3-input OR gate  
**747266:** quad 2-input XNOR gate  
**7440105:** 4-Bit x 16-Word FIFO Register  
**A623308A:** 8K X 8 BIT CMOS SRAM  
**RAM32Bit:** A 32-bit memory that allows byte access and can handle non-aligned memory addresses.