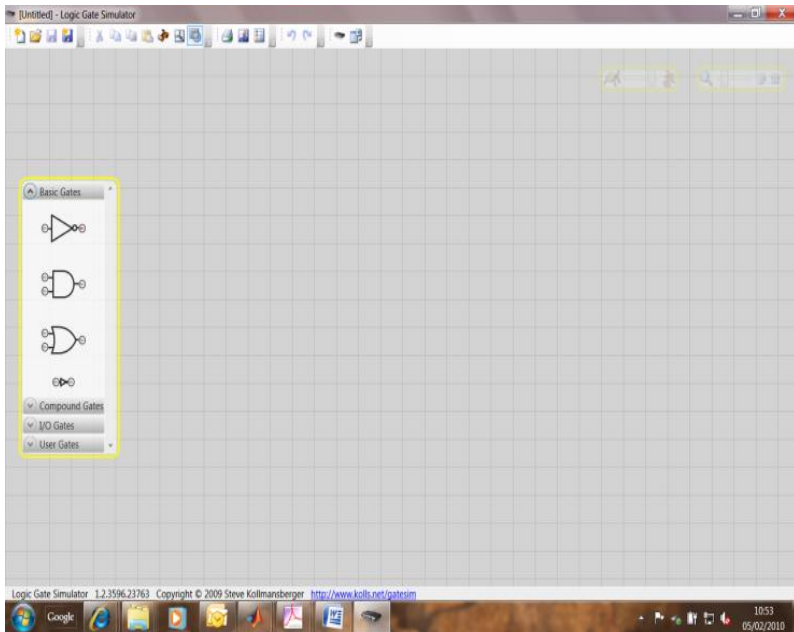


Introduzione a Gatesim e circuiti combinatori

Gatesim

- Gatesim (Logic Gate Simulator)
- Download: <http://www.kolls.net/gatesim/>
- Video introduttivo (porte di input/output, porte loghce, creazione di circuiti personalizzati, velocità di propagazione all'interno del circuito):
<http://www.kolls.net/gatesim/gatesim%20demo.swf>

Gatesim: schermata iniziale



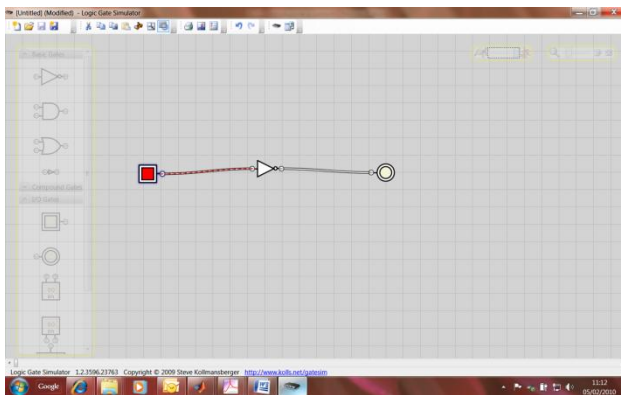
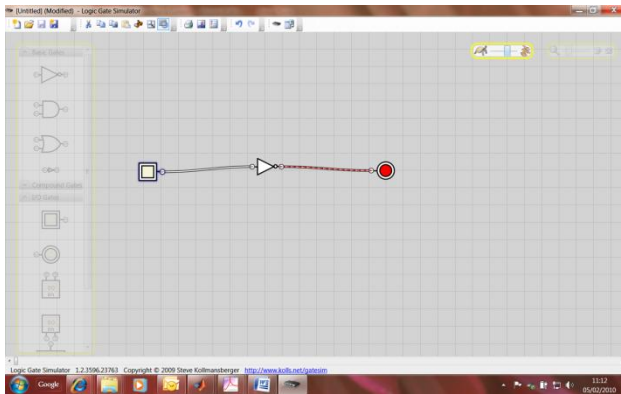
- Nel menu a sx, è possibile accedere alle porte di base (Basic Gate: NOT, AND, OR & Buffer).
- Le porte devono essere trascinate sullo schermo per l'inserimento all'interno del circuito.
- Per fare funzionare una porta logica è necessario fornire degli ingressi logici. Per tale motivo nel menu di sx sono presenti le porte di input/output (I/O Gates).
- La porta di input permette di inserire un segnale true/false.
- La porta di output permette di leggere un segnale true/false.
- Le porte di input / output possono essere rinominate utilizzando il tasto destro del mouse.
- Le porte sono connesse tra di loro semplicemente trascinando con il mouse (tasto sinistro premuto) i connettori di input/output. Da un connettore di output possono uscire più fili!

Es. 1

- Si disegni il circuito che realizza l'operazione di negazione di un segnale chiamato A in Gatesim.

Soluzione e osservazioni

Cliccando sulla porta di input, il segnale viene messo a true/false. La velocità di propagazione del segnale può essere controllata con l'apposita slider.



Es. 2

Si disegni il circuito che realizza:

$$X = (A \text{ and (not B)}) \text{ or } C.$$

Si derivi la tabella della verità del circuito e si controlli la correttezza dei risultati utilizzando Gatesim.

Soluzione

- Per derivare la tabella della verità, dobbiamo considerare tutte le possibili combinazioni degli ingressi. Dobbiamo quindi calcolare l'output della funzione. E' utile calcolare i risultati intermedi e mettere anch'essi nella tabella.

Soluzione

$X = (A \text{ and } (\text{not } B)) \text{ or } C$

A	B	C	Z1 = not B	Z2 = A and Z1	X = Z2 or C
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

Soluzione

$X = (A \text{ and } (\text{not } B)) \text{ or } C$

A	B	C	Z1 = not B	Z2 = A and Z1	X = Z2 or C
0	0	0	1		
0	0	1	1		
0	1	0	0		
0	1	1	0		
1	0	0	1		
1	0	1	1		
1	1	0	0		
1	1	1	0		

Soluzione

$X = (A \text{ and } (\text{not } B)) \text{ or } C$

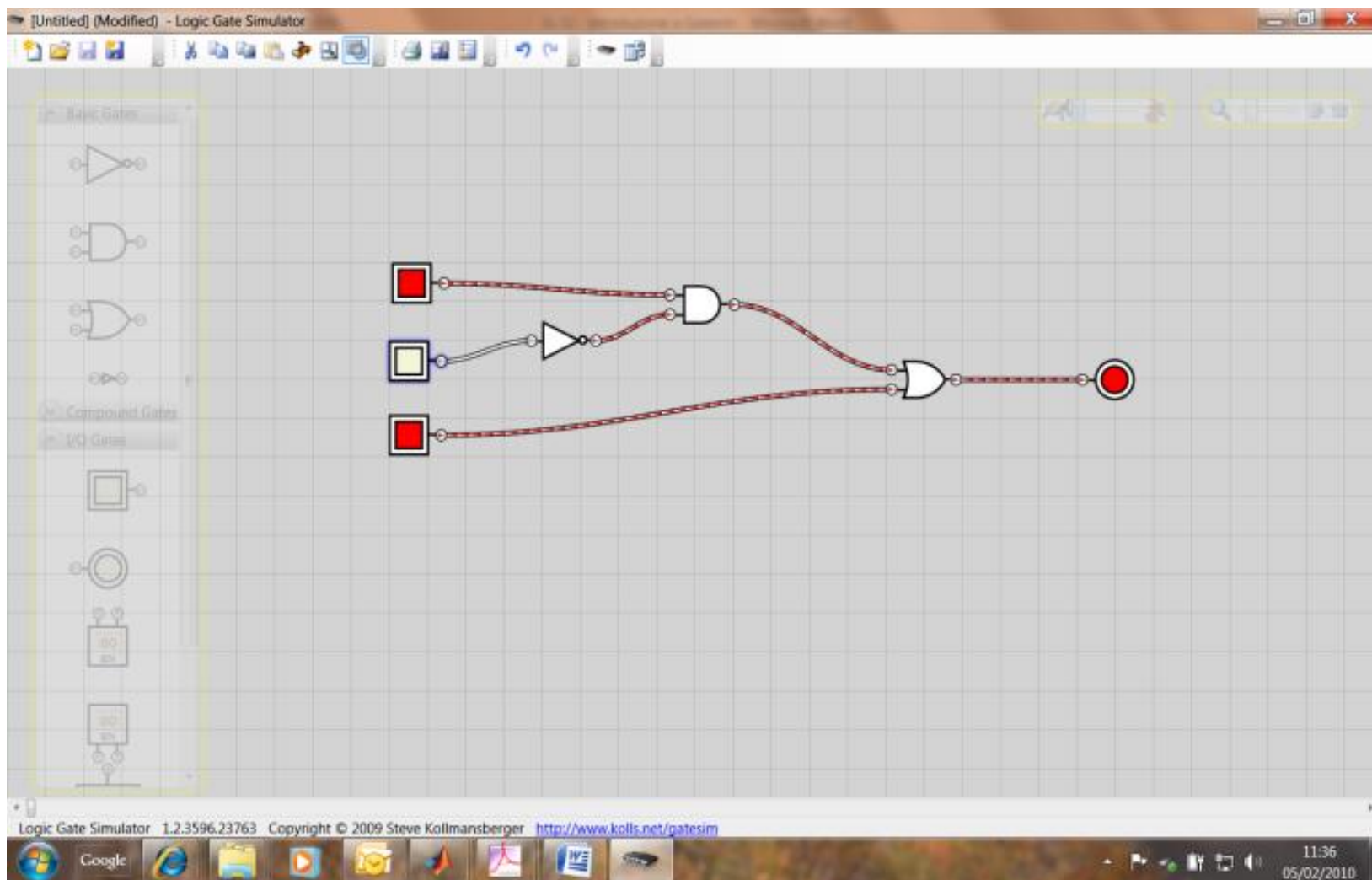
A	B	C	Z1 = not B	Z2 = A and Z1	X = Z2 or C
0	0	0	1	0	
0	0	1	1	0	
0	1	0	0	0	
0	1	1	0	0	
1	0	0	1	1	
1	0	1	1	1	
1	1	0	0	0	
1	1	1	0	0	

Soluzione

$$X = (A \text{ and } (\text{not } B)) \text{ or } C$$

A	B	C	Z1 = not B	Z2 = A and Z1	X = Z2 or C
0	0	0	1	0	0
0	0	1	1	0	1
0	1	0	0	0	0
0	1	1	0	0	1
1	0	0	1	1	1
1	0	1	1	1	1
1	1	0	0	0	0
1	1	1	0	0	1

Soluzione



Porte “compound”

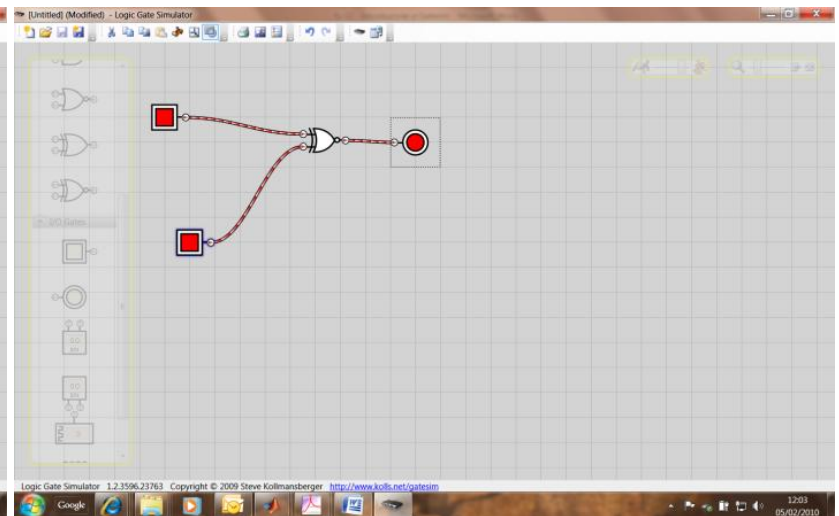
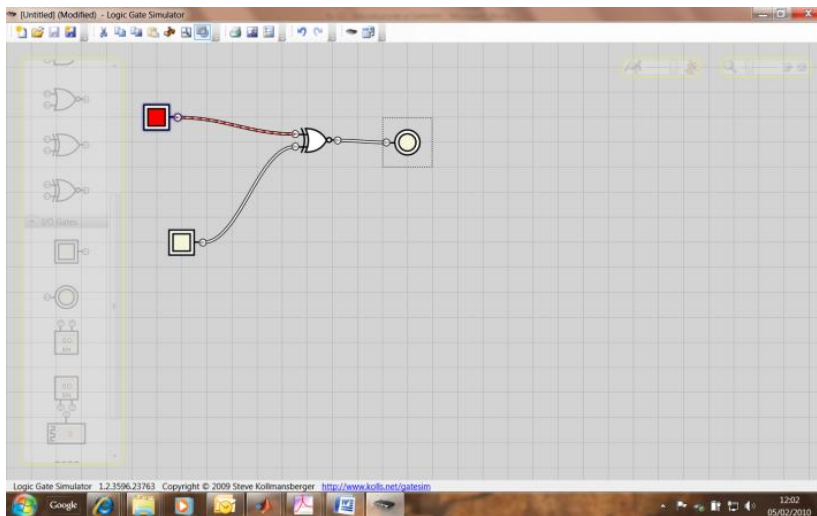
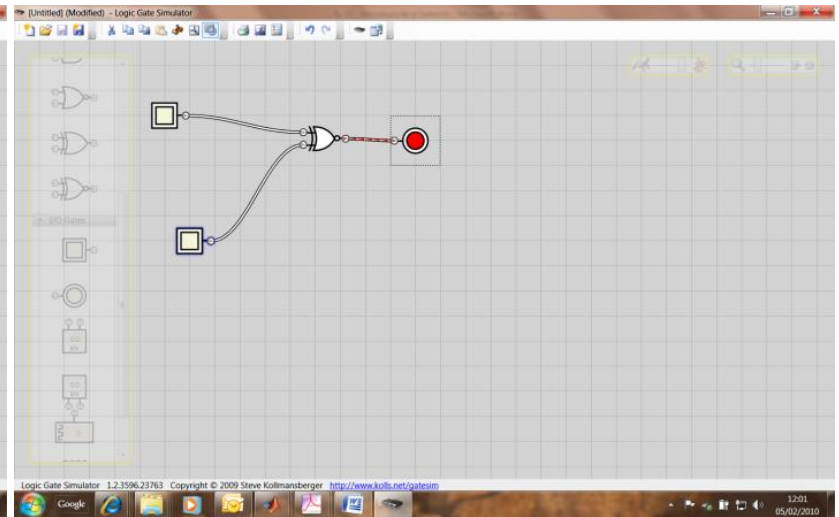
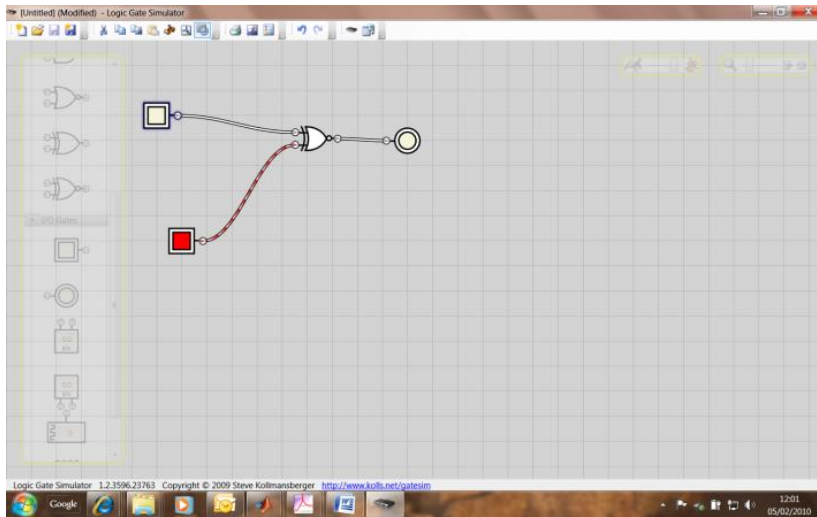
- Oltre alle porte di base ed alle porte di I/O, sono presenti delle porte “compound” che rappresentano le funzioni NAND, NOR, XOR e XNOR.
- Queste sono utili, ad esempio, quali porte universali per la realizzazione di un circuito.

Es. 3

- Si definiscano in Gatesim due segnali A e B. Si utilizzi la porta XNOR per calcolare $X = A \text{ XNOR } B$.
- Si derivi, analizzando l'output X, la tabella di verità di XNOR. A quale funzione logica corrisponde?
- Si implementi in gatesim un circuito equivalente a XNOR utilizzando le porte AND, OR e NOT necessario. Si verifichi la correttezza dell'implementazione confrontando l'uscita di XNOR con l'uscita del circuito implementato.
- hint: le due uscite devono essere uguali per qualsiasi configurazione di ingresso -> è possibile utilizzare la porta XNOR stessa per effettuare questo controllo!

Soluzione

Analizziamo l'uscita della porta XNOR con gatesim:



Soluzione

Deriviamo quindi la tabelle di verità per XNOR:

A	B	A XNOR B
0	0	1
0	1	0
1	0	0
1	1	1

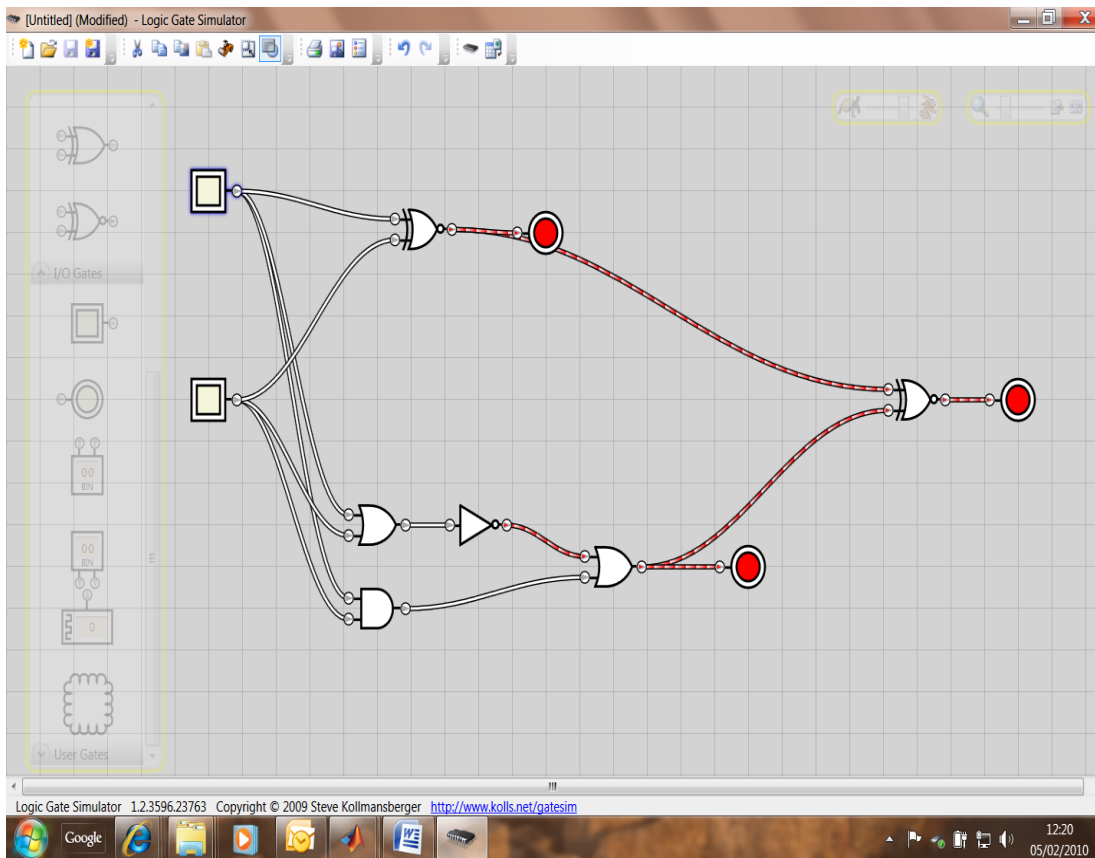
- L'uscita della porta XNOR è pari a 1 quando A e B sono uguali, è pari a 0 altrimenti. La porta determina se i due bit sono uguali o diversi!
- Da qui la possibilità di utilizzare la XNOR per confrontare l'uscita di due circuiti.

Soluzione

- Proviamo a pensare come realizzare il circuito della XNOR (oggi utilizziamo l'intuito – vedremo più che esistono tecniche che rendono il compito di sintesi del circuito molto più semplice).
- Dobbiamo realizzare una porta che va ad 1 quando $A=0$ e $B=0$, oppure quando $A=1$ e $B=1$.
- La porta OR va a zero solo quando $A=0$ e $B=0$, quindi la negazione di OR (detta NOR) va ad 1 solo quando $A=0$ e $B=0$.
- La porta AND va ad 1 solo quando $A=1$ e $B=1$.
- Possiamo quindi generare due segnali ($A \text{ NOR } B$, $A \text{ AND } B$), quando uno di questi due va ad 1 anche la porta XNOR va ad 1.
- Possiamo quindi usare la porta OR per legare i due segnali ed ottenere l'output desiderato...
- Dovrebbe quindi valere: $(\text{NOT}(A \text{ OR } B)) \text{ OR } (A \text{ AND } B) = A \text{ XNOR } B$.

Soluzione

Verifichiamo il risultato con Gatesim:



La parte alta del cricuito è:
A XNOR B.

La parte bassa del c circuito è:
 $(\text{NOT}(A \text{ OR } B)) \text{ OR } (A \text{ AND } B)$.

Soluzione

E' possibile verificare che l'output dei due circuiti è lo stesso per qualsiasi configurazione di ingresso di A e B. Tale verifica può essere fatta anche ponendo in ingresso ad una porta XNOR l'output dei due circuiti e verificando che l'output della porta XNOR è 1 per qualsiasi configurazione di ingresso.

Quale ulteriore verifica (ed esercizio), ricaviamo la tabella della verità di $(\text{NOT}(A \text{ OR } B)) \text{ OR } (A \text{ AND } B)$ e verifichiamo che l'output sia lo stesso di $A \text{ XNOR } B$.

A	B	Z1 = A OR B	Z2 = NOT Z1	Z3 = A AND B	X = Z2 OR Z3
0	0	0	1	0	1
0	1	1	0	0	0
1	0	1	0	0	0
1	1	1	0	1	1

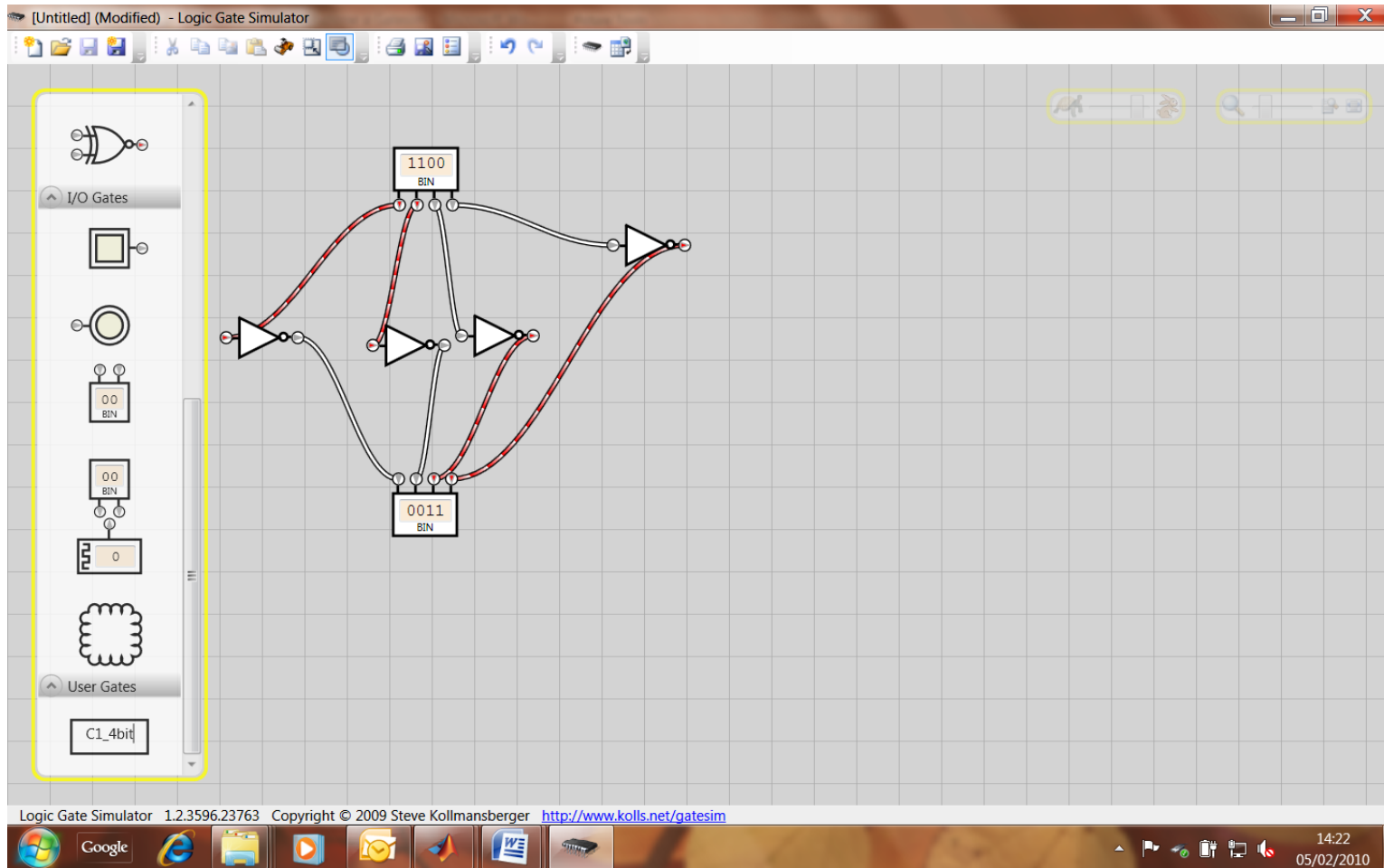
Porte a più bit / salvataggio

- In Gatesim è possibile definire delle porte di ingresso con più di un bit. Questa funzionalità è comoda per simulare, ad esempio, l'elaborazione di un byte.
- E' inoltre possibile salvare i circuiti progettati per poterli riutilizzare in futuro.

Es. 4

- Si costruisca con Gatesim un circuito che calcoli il complemento a 1 di una sequenza di 4 bit (il complemento a 1 si ottiene semplicemente invertendo il valore dei singoli bit).
- Si salvi il circuito sviluppato con il nome di C1_4bit.

Soluzione

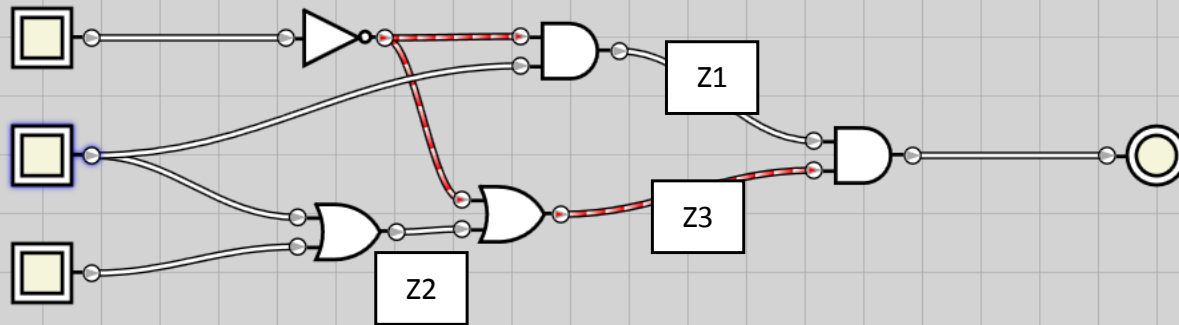


La soluzione proposta può essere salvata in Gatesim (Create IC), ma porta ad un circuito senza input ne output... Meglio utilizzare gli input/output singoli per salvare un circuito riutilizzabile.

Quando si inserisce un nuovo circuito -> tasto dx, in linea -> esplosione il circuito.

Es. 5

Si ricavi la tabella di verità del seguente circuito e se ne verifichi la correttezza.



Soluzione

- Abbiamo tre ingressi, che chiameremo A, B e C. Per calcolare la tabella che descrive l'uscita X del circuito, calcoliamo prima i “risultati” parziali delle operazioni...
- $Z1 = \text{not}(A) \text{ and } B$
- $Z2 = B \text{ or } C$
- $Z3 = \text{not}(A) \text{ or } Z2$
- $X = Z1 \text{ and } Z3 = (\text{not}(A) \text{ and } B) \text{ and } (\text{not}(A) \text{ or } Z2) = (\text{not}(A) \text{ and } B) \text{ and } (\text{not}(A) \text{ or } (B \text{ or } C))$

Soluzione

A	B	C	Z1	Z2	Z3	X
0	0	0				
0	0	1				
0	1	0				
0	1	1				
1	0	0				
1	0	1				
1	1	0				
1	1	1				

Soluzione

A	B	C	Z1	Z2	Z3	X
0	0	0	0			
0	0	1	0			
0	1	0	1			
0	1	1	1			
1	0	0	0			
1	0	1	0			
1	1	0	0			
1	1	1	0			

Soluzione

A	B	C	Z1	Z2	Z3	X
0	0	0	0	0		
0	0	1	0	1		
0	1	0	1	1		
0	1	1	1	1		
1	0	0	0	0		
1	0	1	0	1		
1	1	0	0	1		
1	1	1	0	1		

Soluzione

A	B	C	Z1	Z2	Z3	X
0	0	0	0	0	0	0
0	0	1	0	1	1	0
0	1	0	1	1	1	1
0	1	1	1	1	1	1
1	0	0	0	0	0	0
1	0	1	0	1	1	0
1	1	0	0	1	1	0
1	1	1	0	1	1	0

L'uscita va a 1 solo quando $A=0$ e $B=1$, indipendentemente dal valore di C .