LABORATORIO DI ARCHITETTURA DEI CALCOLATORI
lezione n° 3

Prof. Rosario Cerbone

rosario.cerbone@libero.it

a.a. 2005-2006

Descrizioni di circuiti combinatori tramite SIS

- In questa lezione vengono introdotti i concetti fondamentali per utilizzare SIS, un programma che consente di analizzare, ottimizzare e sintetizzare reti logiche.
- SIS permette di memorizzare, analizzare, ottimizzare e sintetizzare reti logiche in vari formati.
- Il formato *blif* è utilizzato per rappresentare circuiti combinatori rappresentati da tabelle delle verità.

Descrizioni di circuiti combinatori tramite SIS

- Per descrivere una funzione booleana nel formato blif è necessario specificare le seguenti informazioni:
- - Nome della funzione tramite la keyword **.model**
- - Elenco degli input tramite la keyword **.inputs**
- - Elenco degli output tramite la keyword **.outputs**
- - On-set per ogni output tramite la keyword **.names**
- - Il file deve essere terminato dalla keyword **.end**

Descrizioni di circuiti combinatori tramite SIS

- Ad esempio il formato blif per rappresentare una porta AND a 3 ingressi è il seguente:
- .model and
- .inputs x y z
- .outputs a
- .names x y z a
- 111 1 *spazio obbligatorio prima dell'ultimo 1
- .end

Descrizioni di circuiti combinatori tramite SIS

- Il formato blif può essere letto tramite il comando **read_blif** dopo aver lanciato il programma tramite il comando **sis**.
- Il circuito può quindi essere simulato tramite il comando **simulate** seguito dai valori che devono essere assegnati agli ingressi.

Descrizioni di circuiti combinatori tramite SIS

- I passi da compiere sono i seguenti:

1. Scrivere la descrizione in formato blif con un editor di testo (blocco note) e salvare il file come nome.blif
2. Eseguire il programma sis
3. Leggere il file con l'istruzione read_blif nome.blif (abbreviato rl nome.blif)
4. Controllare se il file è stato letto correttamente con write_blif (abbreviato wl)
5. Il comando print_stats (psf) stampa le informazioni sul circuito
6. Provare il circuito con simulate (sim) seguito dai valori da assegnare alle variabili di ingresso separati da uno spazio. Es. simulate 1 0 1
7. Con write_eqn (we) il programma ricava le equazioni del circuito che possono essere salvate su file con write_eqn nome.blif
8. Per chiudere il programma: quit

Descrizioni di circuiti combinatori tramite SIS

- Altri comandi utili nell'utilizzo di sis sono i seguenti:
- - **help** fornisce l'elenco dei comandi disponibili
- - **help comando** fornisce aiuto per il comando indicato
- - **alias** fornisce l'elenco delle abbreviazioni attive
- - **print_stats** fornisce informazioni sul circuito, quali numero di input, output, elementi di memoria, letterali
- - **write_blif** stampa il file blif del circuito in memoria

Esercizi

- **Esercizio 1:** Descrivere in formato blif le seguenti funzioni booleane: *AND, OR, NOT, NAND, NOR, XOR, XNOR*.

- Simularne il loro comportamento con SIS.
- Ricavare l'espressione booleana con SIS.

Esercizi

- **Esercizio 2:** Descrivere in formato blif un sommatore binario ad un bit.
- Simularne il comportamento con SIS.
- Ricavare la funzione logica

Esercizi

- **Esercizio 3:** Scrivere la tabella delle verità di un circuito che conta quanti 1 sono presenti in una sequenza di 4 cifre binarie.
- Il circuito avrà quindi 4 ingressi e 3 uscite (al massimo è possibile avere quattro 1 in ingresso e in tal caso l'output del circuito deve essere il numero binario 100).
- Descrivere il circuito in formato blif e simularne il comportamento con SIS.

Soluzioni esercizio 1

Porta AND

- .model and
- .inputs x y
- .outputs z
- .names x y z
- 11 1
- .end

Soluzioni esercizio 1

Porta OR

- .model OR
- .inputs x y
- .outputs z
- .names x y z
- 01 1
- 10 1
- 11 1
- .end

Soluzioni esercizio 1

Porta NOT

- .model NOT
- .inputs x
- .outputs z
- .names x z
- 0 1
- .end

Soluzioni esercizio 1

Porta NAND

- .model NAND
- .inputs x y
- .outputs z
- .names x y z
- 00 1
- 01 1
- 10 1
- .end

Soluzioni esercizio 1

Porta NOR

- .model NOR
- .inputs x y
- .outputs z
- .names x y z
- 00 1
- .end

Soluzioni esercizio 1

Porta XOR

- .model XOR
- .inputs x y
- .outputs z
- .names x y z
- 01 1
- 10 1
- .end

Soluzioni esercizio 1

Porta XNOR

- .model XNOR
- .inputs x y
- .outputs z
- .names x y z
- 00 1
- 11 1
- .end

Soluzioni esercizio 2

Sommatore 1 bit

- .model full_adder
- .inputs x y cin
- .outputs s cout
- .names x y cin s
- 001 1
- 010 1
- 100 1
- 111 1
- .names x y cin cout
- 011 1
- 101 1
- 110 1
- 111 1

- .end

Soluzioni esercizio 3

Contatore di 1

X3	x2	x1	x0	z2	z1	z0
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	1	0
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	0	1	0
0	1	1	1	0	1	1
1	0	0	0	0	0	1
1	0	0	1	0	1	0
1	0	1	0	0	1	0
1	0	1	1	0	1	1
1	1	0	0	0	1	0
1	1	0	1	0	1	1
1	1	1	0	0	1	1
1	1	1	1	1	0	0

Soluzioni esercizio 3

Contatore di 1

- .model counter_one
- .inputs x0 x1 x2 x3
- .outputs z0 z1 z2
- .names x0 x1 x2 x3 z0
- 0001 1
- 0010 1
- 0100 1
- 0111 1
- 1000 1
- 1011 1
- 1101 1
- 1110 1
- .names x0 x1 x2 x3 z1
- 0011 1
- 0101 1
- 0110 1
- 0111 1
- 1001 1
- 1010 1
- 1011 1
- 1100 1
- 1101 1
- 1110 1
- .names x0 x1 x2 x3 z2
- 1111 1
- .end