Modellazione dei ritardi e mapping tecnologico in SIS

A cura di:
Ing. **Alessandra De Benedictis**, alessandra.debenedictis@unina.it

CDL Magistrale Ing. Informatica - Prof. Antonino Mazzeo
Architettura dei Sistemi di Elaborazione

Mapping tecnologico in SIS

- Le tecniche per la minimizzazione di funzioni combinatorie a due o più livelli viste sono completamente indipendenti dalla libreria tecnologica di porte logiche con la quale sarà realizzato il circuito reale e necessitano quindi di una fase finale di associazione a una libreria di celle.

- SIS esegue questa operazione con il comando **map** che deve essere eseguito dopo aver caricato in memoria una libreria di porte. Il comando **read_library** permette di effettuare questa operazione leggendo le caratteristiche delle porte da un file in formato ***.genlib***.
Questo tipo di file riporta le caratteristiche di ogni porta in termini di equazioni, area e ritardi. Una volta caricato un file di libreria è possibile esaminare le caratteristiche di ogni porta eseguendo il comando **print_library**.

Mapping tecnologico: libreria mcnc.genlib

		Area	Function	Propagation Delay
1	GATE inv1	1	O=!a;	PIN * INV 1 999 0.9 0.3 0.9 0.3
2	GATE inv2	2	O=!a;	PIN * INV 2 999 1.0 0.1 1.0 0.1
3	GATE inv3	3	O=!a;	PIN * INV 3 999 1.1 0.09 1.1 0.09
4	GATE inv4	4	O=!a;	PIN * INV 4 999 1.2 0.07 1.2 0.07
5	GATE nand2	2	O=!(a*b);	PIN * INV 1 999 1.0 0.2 1.0 0.2
6	GATE nand3	3	O=!(a*b*c);	PIN * INV 1 999 1.1 0.3 1.1 0.3
7	GATE nand4	4	O=!(a*b*c*d);	PIN * INV 1 999 1.4 0.4 1.4 0.4
8	GATE nor2	2	O=!(a+b);	PIN * INV 1 999 1.4 0.5 1.4 0.5
9	GATE nor3	3	O=!(a+b+c);	PIN * INV 1 999 2.4 0.7 2.4 0.7
10	GATE nor4	4	O=!(a+b+c+d);	PIN * INV 1 999 3.8 1.0 3.8 1.0
11	GATE and2	3	O=a*b;	PIN * NONINV 1 999 1.9 0.3 1.9 0.3
12	GATE or2	3	O=a+b;	PIN * NONINV 1 999 2.4 0.3 2.4 0.3
13	GATE xor	5	O=a*!b+!a*b;	PIN * UNKNOWN 2 999 1.9 0.5 1.9 0.5
14	GATE xor	5	O=!(a*b+!a*!b);	PIN * UNKNOWN 2 999 1.9 0.5 1.9 0.5
15	GATE xnor	5	O=a*b+!a*!b;	PIN * UNKNOWN 2 999 2.1 0.5 2.1 0.5
16	GATE xnor	5	O=!(!a*b+a*!b);	PIN * UNKNOWN 2 999 2.1 0.5 2.1 0.5
17	GATE aoi21	3	O=!(a*b+c);	PIN * INV 1 999 1.6 0.4 1.6 0.4
18	GATE aoi22	4	O=!(a*b+c*d);	PIN * INV 1 999 2.0 0.4 2.0 0.4
19	GATE oai21	3	O=!((a+b)*c);	PIN * INV 1 999 1.6 0.4 1.6 0.4
20	GATE oai22	4	O=!((a+b)*(c+d));	PIN * INV 1 999 2.0 0.4 2.0 0.4
21	GATE zero	0	O=CONST0;	
22	GATE one	0	O=CONST1;	

Il ritardo di propagazione è formato da 4 valori:

- il massimo ritardo
- il massimo ritardo addizionale per fanout
- il minimo ritardo
- il minimo ritardo addizionale per fanout

La libreria **mcnc** contiene alcuni componenti elementari come invertitori, nand, nor, and, or ecc., per i quali è specificata ad esempio l'occupazione di area

Mapping tecnologico: caricamento di una libreria

```
UC Berkeley, SIS 1.2 (compiled May 21 2002 09:23:42)
sis> read_library sis-1.2\sis\sis_lib\mcnc.genlib
sis> print_library
class: inv1
    gate: inv1        area:  1.00  dual_class: inv1
          {O} = a'
    gate: inv2        area:  2.00  dual_class: inv1
          {O} = a'
    gate: inv3        area:  3.00  dual_class: inv1
          {O} = a'
    gate: inv4        area:  4.00  dual_class: inv1
          {O} = a'

class: nand2
    gate: nand2       area:  2.00  dual_class: nor2
          {O} = (a b)'

class: nand3
    gate: nand3       area:  3.00  dual_class: nor3
          {O} = (a b c)'

class: nand4
    gate: nand4       area:  4.00  dual_class: nor4
          {O} = (a b c d)'
```

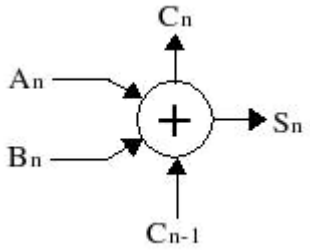
- La libreria viene caricata con il comando:
  read_library *nome_libreria.genlib*

- La libreria caricata viene stampata a video con il comando: **print_library**
- Si può ricercare una specifica porta all'interno di una determinata libreria con il comando **print_library** *funzione_implementata:* verranno restituiti i gate che implementano la funzione specificata

Esempio: Full-adder a 1 bit - Descrizione .blif rete a due livelli

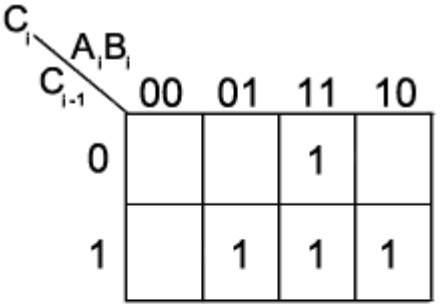
File *sommatore_2liv.blif*

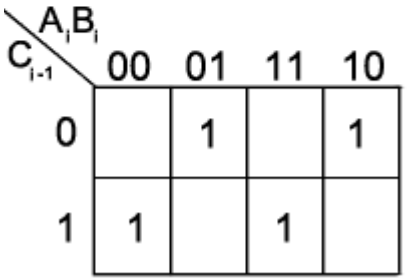
```
.model SOMMATORE
.inputs x y cin
.outputs s cout
.names x y cin s
001 1
010 1
100 1
111 1
.names x y cin cout
011 1
101 1
110 1
111 1
.end
```



An	Bn	Cn-1	Sn	Cn
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Tabella della verità del full adder



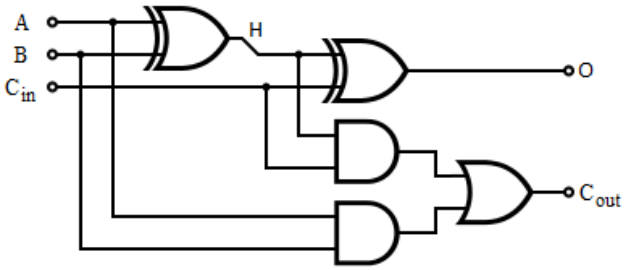


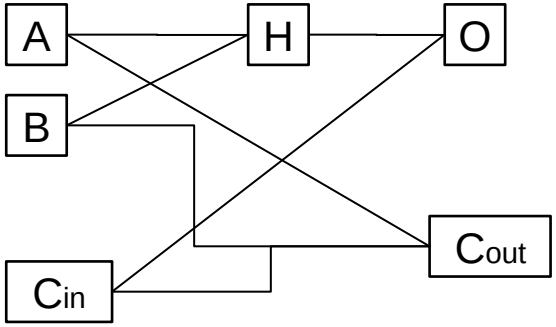
Sommatore a 1 bit: descrizione .blif rete multilivello

File *sommatore.blif*

```
model SOMMATORE
.inputs A B CIN
.outputs O COUT
.names A B H
10 1
01 1
.names H CIN O
10 1
01 1
.names A B CIN COUT
11- 1
1-1 1
-11 1
.end
```

Equazioni del full adder:
H = A XOR B
O = A XOR B XOR Cin = H XOR Cin
COUT = (A AND B) OR (A AND Cin) OR (B AND Cin)





Sommatore a 1 bit: mapping tecnologico

```
sis> read_blif sommatore.blif
sis> read_library mcnc.genlib
sis> map -W -m 0 -s
>>> before removing serial inverters <<<
# of outputs:           2
total gate area:        17.00
maximum arrival time: (7.70,7.70)
maximum po slack:     (-4.70,-4.70)
minimum po slack:     (-7.70,-7.70)
total neg slack:      (-12.40,-12.40)
# of failing outputs:  2
>>> before removing parallel inverters <<<
# of outputs:           2
total gate area:        17.00
maximum arrival time: (7.70,7.70)
maximum po slack:     (-4.70,-4.70)
minimum po slack:     (-7.70,-7.70)
total neg slack:      (-12.40,-12.40)
# of failing outputs:  2
# of outputs:           2
total gate area:        17.00
maximum arrival time: (7.70,7.70)
maximum po slack:     (-4.70,-4.70)
minimum po slack:     (-7.70,-7.70)
total neg slack:      (-12.40,-12.40)
# of failing outputs:  2
```

Col comando
map –W –m 0 –s
si effettua un mapping con l'obiettivo di minimizzare *l'**area*** del circuito risultante.

- Il *maximum arrival time* indica il massimo ritardo, relativo al percorso critico (nota: di default il tempo richiesto di arrivo è settato a zero)
- Il *maximum po slack* e il *minimum po slack* indicano rispettivamente lo slack maggiore e minore
- Il *total neg slack* fornisce lo slack negativo totale, dato dalla somma degli slack negativi

Scelta dei criteri di costo nel mapping

- L'opzione **–m** del comando **map** consente di specificare la funzione di costo rispetto alla quale ottimizzare il mapping della rete: specificando il valore 0 (default) la funzione di costo scelta è l'area del circuito risultante, mentre indicando 1 si scelglie di minimizzare il ritardo.
  - *E' possibile specificare un valore intermedio (ad es 0.5) in modo da ottenere una combinazione lineare delle due funzioni.*

- L'opzione **–m** utilizza un algoritmo di copertura semplificato e veloce; solo per la minimizzazione del ritardo, ovvero con valore 1, è possibile specificare l'opzione **–n** invece di –m, per indicare a SIS di utilizzare un algoritmo di copertura più complesso che tiene conto anche dei carichi sulle uscite, sensibilmente più lento.

- L'opzione –s mostra le statistiche relative ai risultati del mapping, mentre –W indica di non visualizzare i warning

Per ulteriori opzioni si rimanda all'help (digitare *help map*).

Sommatore a 1 bit: mapping tecnologico con ottimizzazione del ritardo

```
sis> map -W -m 1 -s
>>> before removing serial inverters <<<
# of outputs:              2
total gate area:           33.00
maximum arrival time: (9.00,9.00)
maximum po slack:     (-6.20,-6.20)
minimum po slack:     (-9.00,-9.00)
total neg slack:      (-15.20,-15.20)
# of failing outputs:  2
>>> before removing parallel inverters <<<
# of outputs:              2
total gate area:           24.00
maximum arrival time: (7.80,7.80)
maximum po slack:     (-3.90,-3.90)
minimum po slack:     (-7.80,-7.80)
total neg slack:      (-11.70,-11.70)
# of failing outputs:  2
# of outputs:              2
total gate area:           22.00
maximum arrival time: (7.80,7.80)
maximum po slack:     (-3.90,-3.90)
minimum po slack:     (-7.80,-7.80)
total neg slack:      (-11.70,-11.70)
# of failing outputs:  2
```

Per minimizzare il *ritardo* del circuito risultante dal mapping invece si usa il comando:
map –W –m 1 –s

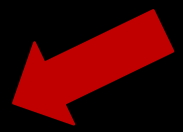
Sommatore a 1 bit: statistiche sul circuito mappato

```
sis> read_blif sommatore.blif
sis> print_stats
SOMMATORE       pi= 3   po= 2   nodes=  3       latches= 0
lits(sop)=  14
sis> map -W -m 0 -s
>>> before removing serial inverters <<<
# of outputs:           2
total gate area:        17.00
maximum arrival time: (7.70,7.70)
maximum po slack:     (-4.70,-4.70)
minimum po slack:     (-7.70,-7.70)
total neg slack:      (-12.40,-12.40)
# of failing outputs:  2
>>> before removing parallel inverters <<<
# of outputs:           2
total gate area:        17.00
maximum arrival time: (7.70,7.70)
maximum po slack:     (-4.70,-4.70)
minimum po slack:     (-7.70,-7.70)
total neg slack:      (-12.40,-12.40)
# of failing outputs:  2
# of outputs:           2
total gate area:        17.00
maximum arrival time: (7.70,7.70)
maximum po slack:     (-4.70,-4.70)
minimum po slack:     (-7.70,-7.70)
total neg slack:      (-12.40,-12.40)
# of failing outputs:  2
sis> print
     [1497] = B'
     [1496] = A'
     H = A' B' + A' [1496]' + B' [1497]' + [1496]' [1497]'
     <O> = CIN H + CIN' H'
     [1521] = A' B' + CIN'
     <COUT> = [1496]' [1497]' + [1521]'
sis> print_gate
[1497]      inv1            1.00
[1496]      inv1            1.00
H           aoi22           4.00
<O>         xnor            5.00
[1521]      oai21           3.00
<COUT>      oai21           3.00
sis> print_stats
SOMMATORE       pi= 3   po= 2   nodes=  6       latches= 0
lits(sop)=  20
sis> print_map_stats
Total Area              =  17.00
Gate Count              = 6
Buffer Count            = 0
Inverter Count          = 2
Most Negative Slack     =  -7.70
Sum of Negative Slacks  = -12.40
Number of Critical PO   = 2
```

- **print**: dopo che il circuito è stato mappato, il comando **print** mostra tutti i nodi della node_list in forma SOP
- **print_gate**: mostra informazioni sulle porte della libreria usate nel mapping
- **print_map_stats**: stampa a video alcune caratteristiche del circuito mappato:
  - l'occupazione di area totale
  - il numero di porte utilizzate
  - il numero di buffer utilizzati
  - il numero di invertitori presenti nel circuito
  - la slack più negativa associata ad un percorso del circuito e la somma delle slack negative
  - il numero di punti critici di uscita del circuito

Sommatore a 1 bit: rappresentazione del circuito con porte di libreria

```
sis> write_blif -n
.model SOMMATORE
.inputs A B CIN
.outputs O COUT
.default_input_arrival 0.00 0.00
.default_output_required 0.00 0.00
.default_input_drive 0.10 0.10
.default_output_load 2.00
.default_max_input_load 999.00
.gate inv1 a=B O=[1497]
.gate inv1 a=A O=[1496]
.gate aoi22 a=[1497] b=A c=[1496] d=B O=H
.gate xnor a=CIN b=H O=O
.gate oai21 a=A b=B c=CIN O=[1521]
.gate oai21 a=[1496] b=[1497] c=[1521] O=COUT
.end
```



- *Default input arrival:* tempo di set degli ingressi primari
- *Default output required:* tempo richiesto per avere a disposizione gli output
- *Default input drive:* tempo necessario affinchè gli ingressi siano effettivi ai singoli nodi

Possono essere visualizzati anche col comando
> **constraints [node list]**

Col comando
>**write_blif –n**
È possibile visualizzare una rappresentazione del circuito associato alle porte di libreria in formato *net-list*

Sommatore a 1 bit: valutazione dei ritardi

```
sis> map -W -m 0 -s
>>> before removing serial inverters <<<
# of outputs:           2
total gate area:        17.00
maximum arrival time: (7.50,7.50)
maximum po slack:     (-4.70,-4.70)
minimum po slack:     (-7.50,-7.50)
total neg slack:      (-12.20,-12.20)
# of failing outputs:  2
>>> before removing parallel inverters <<<
# of outputs:           2
total gate area:        17.00
maximum arrival time: (7.50,7.50)
maximum po slack:     (-4.70,-4.70)
minimum po slack:     (-7.50,-7.50)
total neg slack:      (-12.20,-12.20)
# of failing outputs:  2
# of outputs:           2
total gate area:        17.00
maximum arrival time: (7.50,7.50)
maximum po slack:     (-4.70,-4.70)
minimum po slack:     (-7.50,-7.50)
total neg slack:      (-12.20,-12.20)
# of failing outputs:  2
sis> print_delay
 ... using library delay model
A          : arrival=( 0.30  0.30) required=(-7.20 -7.20) slack=(-7.50 -7.50)
B          : arrival=( 0.30  0.30) required=(-7.20 -7.20) slack=(-7.50 -7.50)
CIN        : arrival=( 0.30  0.30) required=(-4.40 -4.40) slack=(-4.70 -4.70)
[1591]     : arrival=( 1.80  1.80) required=(-5.70 -5.70) slack=(-7.50 -7.50)
[1592]     : arrival=( 1.80  1.80) required=(-5.70 -5.70) slack=(-7.50 -7.50)
H          : arrival=( 4.60  4.60) required=(-2.90 -2.90) slack=(-7.50 -7.50)
<O>        : arrival=( 7.50  7.50) required=( 0.00  0.00) slack=(-7.50 -7.50)
[1614]     : arrival=( 2.30  2.30) required=(-2.40 -2.40) slack=(-4.70 -4.70)
<COUT>     : arrival=( 4.70  4.70) required=( 0.00  0.00) slack=(-4.70 -4.70)
sis> print_delay O
 ... using library delay model
<O>        : arrival=( 7.50  7.50) required=( 0.00  0.00) slack=(-7.50 -7.50)
```

Il comando:

>**print_delay *node-list***

Effettua un'analisi statica sul ritardo considerando uno specifico modello (indicato nell'opzione –m). Se non si specifica alcun modello si assume quello di default *library,* che presuppone che la rete sia stata mappata su una data libreria tecnologica.

Il comando visualizza i ritardi richiesti ed effettivi per i nodi indicati e il relativo slack, dato da:

Slack = output required arrival time – maximum arrival time

Settaggio dei ritardi

- Il *maximum arrival time, maximum po slack* e *minimum po slack* vengono calcolati in base ai valori impostati per i parametri *Default input arrival, Default output required, Default input drive.*
- Tali parametri possono essere settati mediante il comando **set_delay**, utilizzando le opportune opzioni:
  - **A**[f]: imposta al valore f il default arrival time per tutti gli input primari
  - **R**[f]: imposta al valore f il default required time per tutti gli output
  - **D**[f]: imposta al valore f il default drive time per tutti gli input

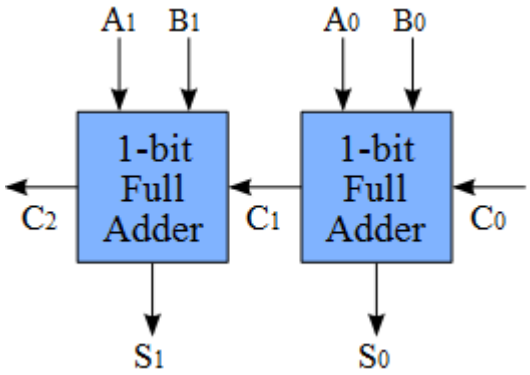
```
sis> set_delay -A0 -R8 -D0
sis> map -W -m 0 /s
sis> map -W -m 0 -s
>>> before removing serial inverters <<<
# of outputs:           2
total gate area:        17.00
maximum arrival time: (7.20,7.20)
maximum po slack:     (3.60,3.60)
minimum po slack:     (0.80,0.80)
total neg slack:      (0.00,0.00)
# of failing outputs:  0
>>> before removing parallel inverters <<<
# of outputs:           2
total gate area:        17.00
maximum arrival time: (7.20,7.20)
maximum po slack:     (3.60,3.60)
minimum po slack:     (0.80,0.80)
total neg slack:      (0.00,0.00)
# of failing outputs:  0
# of outputs:           2
total gate area:        17.00
maximum arrival time: (7.20,7.20)
maximum po slack:     (3.60,3.60)
minimum po slack:     (0.80,0.80)
total neg slack:      (0.00,0.00)
# of failing outputs:  0
```

Sommatore a 2 bit

Un sommatore a 2 bit può essere rappresentato in SIS come la composizione di due sommatori a 1 bit opportunamente collegati:

File *sommatore2.blif*

```
.model SOMMATORE2
.inputs A1 A0 B1 B0 CIN
.outputs O1 O0 COUT
.subckt SOMMATORE x=A0 y=B0 cin=CIN s=O0
cout=C0
.subckt SOMMATORE x=A1 y=B1 cin=C0 s=O1
cout=COUT
.end
.search c:\\sommatore.blif
```



Grazie alla direttiva **.subckt** è possibile includere un componente all'interno di un modello blif secondo la seguente sintassi:

.**subckt** nomecomponente parametroformale=parametroattuale
…
.**search** nomefilecomponente.blif

Sommatore a 2 bit

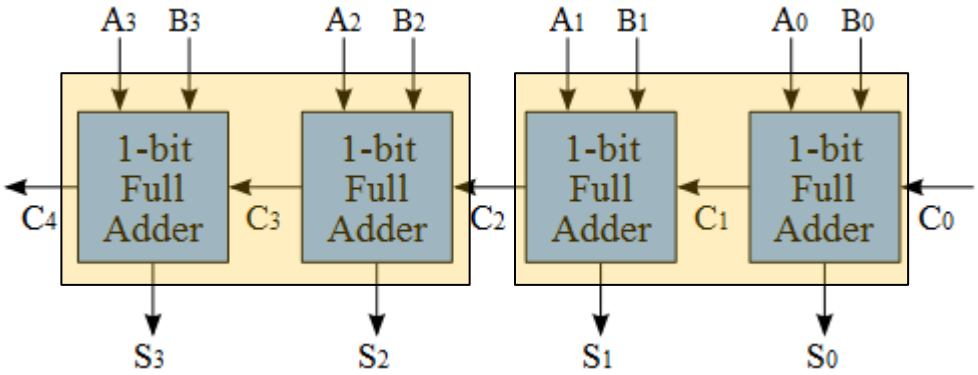
```
sis> read_library mcnc.genlib
sis> read_blif c:\sommatore2.blif
sis> map -W -m 0 -s
>>> before removing serial inverters <<<
# of outputs:           3
total gate area:        35.00
maximum arrival time: (9.50,9.50)
maximum po slack:     (-7.70,-7.70)
minimum po slack:     (-9.50,-9.50)
total neg slack:      (-26.30,-26.30)
# of failing outputs:  3
>>> before removing parallel inverters <<<
# of outputs:           3
total gate area:        35.00
maximum arrival time: (9.50,9.50)
maximum po slack:     (-7.70,-7.70)
minimum po slack:     (-9.50,-9.50)
total neg slack:      (-26.30,-26.30)
# of failing outputs:  3
# of outputs:           3
total gate area:        35.00
maximum arrival time: (9.50,9.50)
maximum po slack:     (-7.70,-7.70)
minimum po slack:     (-9.50,-9.50)
total neg slack:      (-26.30,-26.30)
# of failing outputs:  3
sis> print_map_stats
Total Area              =  35.00
Gate Count              = 14
Buffer Count            = 0
Inverter Count          = 6
Most Negative Slack     =  -9.50
Sum of Negative Slacks  = -26.30
Number of Critical PO   = 3
sis> print_delay O0 O1 COUT
  ... using library delay model
{O0}       : arrival=( 7.70  7.70) required=( 0.00  0.00) slack=(-7.70 -7.70)
{O1}       : arrival=( 9.10  9.10) required=( 0.00  0.00) slack=(-9.10 -9.10)
{COUT}     : arrival=( 9.50  9.50) required=( 0.00  0.00) slack=(-9.50 -9.50)
sis>
```

Come si può notare l'area del circuito è raddoppiata rispetto a quella del circuito base.
Il cammino critico è legato alla generazione di COUT che produce uno slack negativo di
-9.50

Sommatore a 4 bit

Un sommatore a 4 bit può essere rappresentato in SIS come la composizione di due sommatori a 2 bit opportunamente collegati:

File *sommatore4.blif*




```
.model SOMMATORE4
.inputs A3 A2 A1 A0 B3 B2 B1 B0 CIN
.outputs O3 O2 O1 O0 COUT
.subckt SOMMATORE2 A1=A1 A0=A0 B1=B1 B0=B0 CIN=CIN \
O1=O1 O0=O0 COUT=C0
.subckt SOMMATORE2 A1=A3 A0=A2 B1=B3 B0=B2 CIN=C0 \
O1=O3 O0=O2 COUT=COUT
.end
.search c:\\sommatore2.blif
```

Sommatore a 4 bit

```
sis> read_library mcnc.genlib
sis> read_blif c:\sommatore4.blif
sis> map -W -m 0 -s
>>> before removing serial inverters <<<
# of outputs:          5
total gate area:       71.00
maximum arrival time: (19.10,19.10)
maximum po slack:     (-7.70,-7.70)
minimum po slack:     (-19.10,-19.10)
total neg slack:      (-68.50,-68.50)
# of failing outputs:  5
>>> before removing parallel inverters <<<
# of outputs:          5
total gate area:       71.00
maximum arrival time: (19.10,19.10)
maximum po slack:     (-7.70,-7.70)
minimum po slack:     (-19.10,-19.10)
total neg slack:      (-68.50,-68.50)
# of failing outputs:  5
# of outputs:          5
total gate area:       71.00
maximum arrival time: (19.10,19.10)
maximum po slack:     (-7.70,-7.70)
minimum po slack:     (-19.10,-19.10)
total neg slack:      (-68.50,-68.50)
# of failing outputs:  5
sis> print_map_stats
Total Area               =  71.00
Gate Count               = 30
Buffer Count             = 0
Inverter Count           = 14
Most Negative Slack      = -19.10
Sum of Negative Slacks   = -68.50
Number of Critical PO    = 5
sis> print_delay O3 O2 O1 O0 COUT
  ... using library delay model
(O3)       : arrival=(18.70 18.70) required=( 0.00  0.00) slack=(-18.70 -18.70)
(O2)       : arrival=(13.90 13.90) required=( 0.00  0.00) slack=(-13.90 -13.90)
(O1)       : arrival=( 9.10  9.10) required=( 0.00  0.00) slack=(-9.10 -9.10)
(O0)       : arrival=( 7.70  7.70) required=( 0.00  0.00) slack=(-7.70 -7.70)
(COUT)     : arrival=(19.10 19.10) required=( 0.00  0.00) slack=(-19.10 -19.10)
sis>
```

L'area del circuito è quadruplicata rispetto a quella del circuito base.
Il cammino critico è legato alla generazione di COUT che produce uno slack negativo di
-19.10

Sommatore a 4 bit

- **Per comprimere questo ritardo si può rappresentare un sommatore ad anticipo di riporto in cui l'equazione di ogni bit (di riporto o di uscita) viene espressa in funzione dei soli bit di ingresso**
- **anticipando quindi il calcolo rispetto alla propagazione del riporto**
- **Per ottenere questa descrizione, bisogna trasformare il sommatore con riporto da una descrizione multilivello ad una a due livelli**
  - **in cui la dipendenza di ogni uscita dagli ingressi diventa esplicita.**

Sommatore 4 bit

- **Per far questo è sufficiente**
  - **collassare i nodi della rete nei soli cinque nodi corrispondenti alle uscite ed**
  - **eliminare le eventuali dipendenze tra questi nodi**

Sommatore a 4 bit : collapse

```
sis> read_blif c:\sommatore4.blif
sis> print_stats
SOMMATORE4       pi= 9   po= 5   nodes= 12       latches= 0
lits(sop)=  56
sis> collapse
sis> print_stats
SOMMATORE4       pi= 9   po= 5   nodes=  5       latches= 0
lits(sop)= 684
```

Il comando ***collapse [node]*** collassa l'espressione del nodo specificato in modo che sia funzione dei soli input primari; se non si specificano argomenti il comando collassa l'intera rete su due livelli.
L'operazione di collapse potrebbe generare una rete troppo grande; in alternativa è possibile utilizzare il comando eliminate, che constituisce una collapse selettiva.

Sommatore a 4 bit

- **Ora il numero di letterali è notevolmente aumentato**
- **poiché non si sfrutta più la fattorizzazione della rete che era prima presente nella rappresentazione basata sulle porte della libreria.**
- **Se si stampa la rete:**

Sommatore a 4 bit: rete collassata

```
sis> collapse
sis> write_eqn
INORDER = A3 A2 A1 A0 B3 B2 B1 B0 CIN;
OUTORDER = O3 O2 O1 O0 COUT;
O3 = A3*!B3*!B2*!B1*!B0*!CIN + !A3*B3*!B2*!B1*!B0*!CIN + A3*!A2*!B3*!B1*!B0*!
CIN + !A3*!A2*B3*!B1*!B0*!CIN + A3*!A1*!B3*!B2*!B0*!CIN + !A3*!A1*B3*!B2*!B0*!
CIN + A3*!A2*!A1*!B3*!B0*!CIN + !A3*!A2*!A1*B3*!B0*!CIN + A3*!A0*!B3*!B2*!B1*!
CIN + !A3*!A0*B3*!B2*!B1*!CIN + A3*!A2*!A0*!B3*!B1*!CIN + !A3*!A2*!A0*B3*!B1*!
CIN + A3*!A1*!A0*!B3*!B2*!CIN + !A3*!A1*!A0*B3*!B2*!CIN + A3*!A2*!A1*!A0*!B3*!
CIN + !A3*!A2*!A1*!A0*B3*!CIN + !A3*!B3*B2*B1*B0*CIN + A3*B3*B2*B1*B0*CIN + !
A3*A2*!B3*B1*B0*CIN + A3*A2*B3*B1*B0*CIN + !A3*A1*!B3*B2*B0*CIN + A3*A1*B3*B2*
B0*CIN + !A3*A2*A1*!B3*B0*CIN + A3*A2*A1*B3*B0*CIN + !A3*A0*!B3*B2*B1*CIN + A3*
A0*B3*B2*B1*CIN + !A3*A2*A0*!B3*B1*CIN + A3*A2*A0*B3*B1*CIN + !A3*A1*A0*!B3*B2*
CIN + A3*A1*A0*B3*B2*CIN + !A3*A2*A1*A0*!B3*CIN + A3*A2*A1*A0*B3*CIN + A3*!A0*!
B3*!B2*!B1*!B0 + !A3*!A0*B3*!B2*!B1*!B0 + A3*!A2*!A0*!B3*!B1*!B0 + !A3*!A2*!A0*
B3*!B1*!B0 + A3*!A1*!A0*!B3*!B2*!B0 + !A3*!A1*!A0*B3*!B2*!B0 + A3*!A2*!A1*!A0*!
B3*!B0 + !A3*!A2*!A1*!A0*B3*!B0 + !A3*A0*!B3*B2*B1*B0 + A3*A0*B3*B2*B1*B0 + !
A3*A2*A0*!B3*B1*B0 + A3*A2*A0*B3*B1*B0 + !A3*A1*A0*!B3*B2*B0 + A3*A1*A0*B3*B2*
B0 + !A3*A2*A1*A0*!B3*B0 + A3*A2*A1*A0*B3*B0 + A3*!A1*!B3*!B2*!B1 + !A3*!A1*B3*!

B2*!B1 + A3*!A2*!A1*!B3*!B1 + !A3*!A2*!A1*B3*!B1 + !A3*A1*!B3*B2*B1 + A3*A1*B3*
B2*B1 + !A3*A2*A1*!B3*B1 + A3*A2*A1*B3*B1 + A3*!A2*!B3*!B2 + !A3*!A2*B3*!B2 + !
A3*A2*!B3*B2 + A3*A2*B3*B2;
O2 = A2*!B2*!B1*!B0*!CIN + !A2*B2*!B1*!B0*!CIN + A2*!A1*!B2*!B0*!CIN + !A2*!A1*
B2*!B0*!CIN + A2*!A0*!B2*!B1*!CIN + !A2*!A0*B2*!B1*!CIN + A2*!A1*!A0*!B2*!CIN
 + !A2*!A1*!A0*B2*!CIN + !A2*!B2*B1*B0*CIN + A2*B2*B1*B0*CIN + !A2*A1*!B2*B0*
CIN + A2*A1*B2*B0*CIN + !A2*A0*!B2*B1*CIN + A2*A0*B2*B1*CIN + !A2*A1*A0*!B2*
CIN + A2*A1*A0*B2*CIN + A2*!A0*!B2*!B1*!B0 + !A2*!A0*B2*!B1*!B0 + A2*!A1*!A0*!
B2*!B0 + !A2*!A1*!A0*B2*!B0 + !A2*A0*!B2*B1*B0 + A2*A0*B2*B1*B0 + !A2*A1*A0*!
B2*B0 + A2*A1*A0*B2*B0 + A2*!A1*!B2*!B1 + !A2*!A1*B2*!B1 + !A2*A1*!B2*B1 + A2*
A1*B2*B1;
O1 = A1*!B1*!B0*!CIN + !A1*B1*!B0*!CIN + A1*!A0*!B1*!CIN + !A1*!A0*B1*!CIN + !
A1*!B1*B0*CIN + A1*B1*B0*CIN + !A1*A0*!B1*CIN + A1*A0*B1*CIN + A1*!A0*!B1*!B0
 + !A1*!A0*B1*!B0 + !A1*A0*!B1*B0 + A1*A0*B1*B0;
O0 = A0*!B0*!CIN + !A0*B0*!CIN + !A0*!B0*CIN + A0*B0*CIN;
COUT = B3*B2*B1*B0*CIN + A3*B2*B1*B0*CIN + A2*B3*B1*B0*CIN + A3*A2*B1*B0*CIN
 + A1*B3*B2*B0*CIN + A3*A1*B2*B0*CIN + A2*A1*B3*B0*CIN + A3*A2*A1*B0*CIN + A0*
B3*B2*B1*CIN + A3*A0*B2*B1*CIN + A2*A0*B3*B1*CIN + A3*A2*A0*B1*CIN + A1*A0*B3*
B2*CIN + A3*A1*A0*B2*CIN + A2*A1*A0*B3*CIN + A3*A2*A1*A0*CIN + A0*B3*B2*B1*B0
 + A3*A0*B2*B1*B0 + A2*A0*B3*B1*B0 + A3*A2*A0*B1*B0 + A1*A0*B3*B2*B0 + A3*A1*
A0*B2*B0 + A2*A1*A0*B3*B0 + A3*A2*A1*A0*B0 + A1*B3*B2*B1 + A3*A1*B2*B1 + A2*A1*
B3*B1 + A3*A2*A1*B1 + A2*B3*B2 + A3*A2*B2 + A3*B3;
sis>
```

Rappresentazione mediante libreria della rete collassata

```
sis> read_library mcnc.genlib
sis> map -W -m 0 -s
>>> before removing serial inverters <<<
# of outputs:              5
total gate area:           143.00
maximum arrival time: (14.20,14.20)
maximum po slack:     (-8.90,-8.90)
minimum po slack:     (-14.20,-14.20)
total neg slack:      (-58.30,-58.30)
# of failing outputs:  5
>>> before removing parallel inverters <<<
# of outputs:              5
total gate area:           143.00
maximum arrival time: (14.20,14.20)
maximum po slack:     (-8.90,-8.90)
minimum po slack:     (-14.20,-14.20)
total neg slack:      (-58.30,-58.30)
# of failing outputs:  5
# of outputs:              5
total gate area:           141.00
maximum arrival time: (14.20,14.20)
maximum po slack:     (-8.90,-8.90)
minimum po slack:     (-14.20,-14.20)
total neg slack:      (-58.30,-58.30)
# of failing outputs:  5
```

Ritardi della rete collassata

```
sis> print_delay O0 O1 O2 O3 COUT
  ... using library delay model
(O0)       : arrival=( 8.90  8.90) required=( 0.00  0.00) slack=(-8.90 -8.90)
(O1)       : arrival=(10.00 10.00) required=( 0.00  0.00) slack=(-10.00 -10.00)
(O2)       : arrival=(11.60 11.60) required=( 0.00  0.00) slack=(-11.60 -11.60)
(O3)       : arrival=(13.60 13.60) required=( 0.00  0.00) slack=(-13.60 -13.60)
(COUT)     : arrival=(14.20 14.20) required=( 0.00  0.00) slack=(-14.20 -14.20)
sis>
```

Per ridurre il ritardo della rete è possibile anche utilizzare il comando ***reduce_depth –d[k]*** che riduce il cammino critico portando la rete su k livelli
Se non si specifica alcun parametro la rete viene portata su due livelli

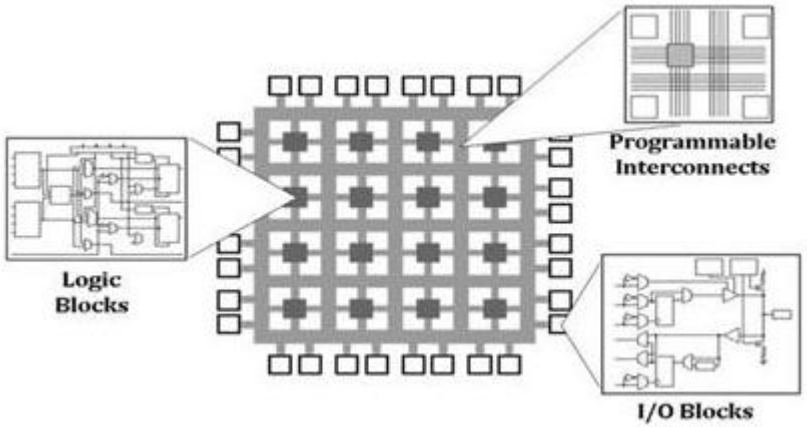
Reduce_depth

```
sis> read_blif c:\sommatore4.blif
sis> read_library mcnc.genlib
sis> reduce_depth
sis> map -W -m 0 -s
>>> before removing serial inverters <<<
# of outputs:           5
total gate area:        143.00
maximum arrival time: (14.20,14.20)
maximum po slack:     (-8.90,-8.90)
minimum po slack:     (-14.20,-14.20)
total neg slack:      (-58.30,-58.30)
# of failing outputs:  5
>>> before removing parallel inverters <<<
# of outputs:           5
total gate area:        143.00
maximum arrival time: (14.20,14.20)
maximum po slack:     (-8.90,-8.90)
minimum po slack:     (-14.20,-14.20)
total neg slack:      (-58.30,-58.30)
# of failing outputs:  5
# of outputs:           5
total gate area:        141.00
maximum arrival time: (14.20,14.20)
maximum po slack:     (-8.90,-8.90)
minimum po slack:     (-14.20,-14.20)
total neg slack:      (-58.30,-58.30)
# of failing outputs:  5
sis> print_delay O0 O1 O2 O3 COUT
  ... using library delay model
<O0>       : arrival=( 8.90  8.90) required=( 0.00  0.00) slack=(-8.90 -8.90)
<O1>       : arrival=(10.00 10.00) required=( 0.00  0.00) slack=(-10.00 -10.00)
<O2>       : arrival=(11.60 11.60) required=( 0.00  0.00) slack=(-11.60 -11.60)
<O3>       : arrival=(13.60 13.60) required=( 0.00  0.00) slack=(-13.60 -13.60)
<COUT>     : arrival=(14.20 14.20) required=( 0.00  0.00) slack=(-14.20 -14.20)
sis>
```

FPGA (Field-Programmable Gate Arrays)

Caratteristiche:

- un insieme di **celle** identiche organizzate secondo una struttura logica a matrice.
- ogni cella ha *n* ingressi dati e può essere configurata come una qualsiasi fra le possibili reti combinatorie a *n* ingressi



Associazione rete logica=>componenti di libreria usata precedentemente non è più valida.
Serve una associazione **sottorete=>cella logica**

Le celle logiche differiscono a seconda della famiglia di FPGA a cui appartengono

SIS e gli FPGA

- SIS ha dei comandi per associare un circuito alle celle di determinate famiglie di FPGA prodotte dalla Xilinx
- I comandi a disposizione verificano che ogni nodo della rete corrente sia realizzabile da una cella Xilinx (fanin dei nodi della rete <= # input di una cella), in caso positivo lo realizzano altrimenti bisogna fattorizzare la rete.
- Qualora sia possibile implementare la rete su fpga, è presente una fase di minimizzazione del numero di nodi della rete.
- Il comando per effettuare l'associazione tra cella e nodo è **xl_cover**
- Il comando per effettuare la fattorizzazione è **xl_imp**

SIS e le FPGA

```
sis> read_blif sommatore.blif
sis> write_eqn
INORDER = A B CIN;
OUTORDER = O COUT;
O = H*!CIN + !H*CIN;
COUT = B*CIN + A*CIN + A*B;
H = A*!B + !A*B;
sis> print_stats
SOMMATORE       pi= 3   po= 2   nodes=  3       latches= 0
lits(sop)=  14
sis> xl_cover -n 3
sis> write_eqn
INORDER = A B CIN;
OUTORDER = O COUT;
O = A*!B*!CIN + !A*B*!CIN + !A*!B*CIN + A*B*CIN;
COUT = B*CIN + A*CIN + A*B;
sis> print_stats
SOMMATORE       pi= 3   po= 2   nodes=  2       latches= 0
lits(sop)=  18
sis> xl_cover -n 2
Error: The network has a node with > 2 fanins
Run xl_imp (xl_ao) -n 2 to decompose these nodes
sis> xl_imp -n 2
sis> xl_cover -n 2
sis> write_eqn
INORDER = A B CIN;
OUTORDER = O COUT;
O = [1969] + [1967];
COUT = [1998] + [1996];
[1966] = B*!CIN + !B*CIN;
[1967] = !A*[1966];
[1968] = !B*!CIN + B*CIN;
[1969] = A*[1968];
[1996] = B*CIN;
[1997] = CIN + B;
[1998] = A*[1997];
sis> print_stats
SOMMATORE       pi= 3   po= 2   nodes=  9       latches= 0
lits(sop)=  22
```

ogni cella ha 3 ingressi

ogni cella ha 2 ingressi

Non è possibile implementare i nodi della rete con soltanto celle a due ingressi.

Tutti i nodi hanno al massimo 2 ingressi e quindi il loro numero è aumentato