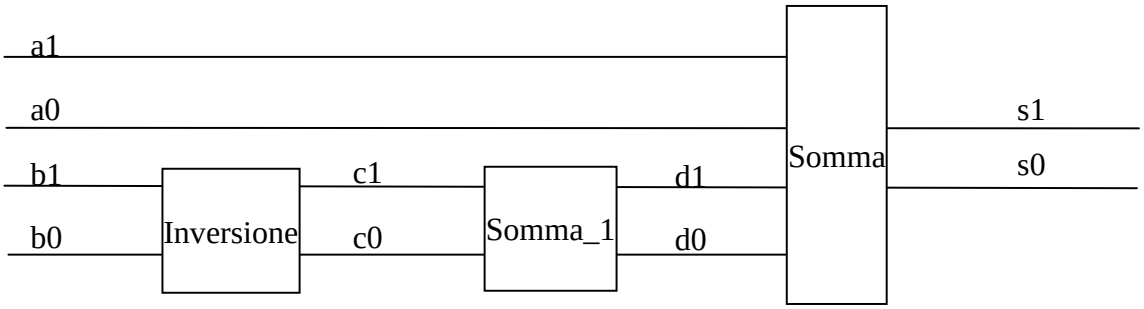
Laboratorio di Architettura degli Elaboratori

Nicola Bombieri
Dipartimento di Informatica
Università di Verona
A.A. 2014/2015

Esercizio svolto

Descrivere in formato blif il circuito digitale che esegue la sottrazione di 2 numeri binari su 2 bit rappresentati in complemento a 2 con risultato ancora su 2 bit in complemento a 2. Lo schema del circuito è il seguente:



Soluzione

```
.model sottrattore2
.inputs a1 a0 b1 b0
.outputs s1 s0

# INVERSIONE
.names b1 b0 c1
00 1
01 1

.names b1 b0 c0
00 1
10 1

# SOMMA 1

.names c1 c0 d1
01 1
10 1

.names c1 c0 d0
00 1
10 1
```

c1, c0, d1 e d0 sono dei segnali interni al circuito; non devono quindi essere dichiarati né insieme ai segnali di input né insieme ai segnali di output

```
# SOMMA

.names a1 a0 d1 d0 s1
0010 1
0011 1
0101 1
0110 1
1000 1
1001 1
1100 1
1111 1

.names a1 a0 d1 d0 s0
0001 1
0011 1
0100 1
0110 1
1001 1
1011 1
1100 1
1110 1

.end
```

Simulazione

La correttezza del circuito si può provare via simulazione, applicando in ingresso alcuni segnali (corrispondenti a valori interi in complemento a due, a due bit) e controllando il valore in uscita:

$ **sis**
UC Berkeley, SIS 1.3
(compiled 25-Jan-02 at 10:59 AM)
sis> **read_blif** "sottrattore2.blif
sis> **simulate 1 1 1 1**
Network simulation:
Outputs: **0 0**
Next state:
sis> **simulate 1 1 0 0**
Network simulation:
Outputs: **1 1**
Next state:
sis> **...**

Minimizzazione di circuiti combinatori multilivello

In questa lezione vengono riassunti i concetti fondamentali della minimizzazione approssimata multi. In particolare viene mostrato come utilizzare SIS per effettuare tali operazioni.

Minimizzazione approssimata multi-livello

La minimizzazione multi-livello consente al progettista di bilanciare area e ritardo di un circuito con un maggior grado di libertà rispetto alla minimizzazione a 2 livelli. Tuttavia, non esistono tecniche esatte efficienti che portino alla realizzazione di configurazioni ottime usando la minimizzazione multi-livello; pertanto si ricorre a tecniche euristiche che garantiscono buone soluzioni in tempi di calcolo ragionevoli.

Riassumiamo di seguito i concetti principali relativi alle tecniche di sintesi applicate durante la minimizzazione multi-livello. Si consideri che il circuito viene rappresentato come un insieme di nodi interconnessi tra loro, e che **ad ogni *nodo* corrisponde una funzione booleana a una sola uscita** (si veda la sezione `nodes` dell'output di `print_stats`). Ogni nodo, pertanto, rappresenta una piccola porzione dell'intero sistema. Le tecniche descritte nei seguenti punti vengono solitamente ripetute secondo un determinato ordine fino al raggiungimento di una configurazione che soddisfi le aspettative del progettista.

- ***Sweep:*** eliminazione dei nodi con un'unica linea di ingresso e di nodi con valore costante.

- ***Eliminazione:*** eliminazione di un nodo interno alla rete. Si consideri che il nodo N rappresenti la funzione y = (a + b)*c. Gli ingressi di N sono pertanto a, b, c mentre l'uscita è y. Si consideri, inoltre, che la variabile di uscita y venga utilizzata come input in alcuni nodi successivi. L'eliminazione di N prevede la sostituzione della variabile y in tutti i nodi che la utilizzano con l'espressione booleana (a + b) * c.

- ***Scomposizione:*** sostituzione di un nodo interno con un'insieme di nodi la cui funzionalità sia equivalente a quella del nodo sostituito. L'operazione viene effettuata per diminuire la complessità di un nodo.

- ***Estrazione:*** estrazione di una sottoespressione comune a più nodi che viene rappresentata con un nuovo nodo.

- ***Semplificazione:*** riduzione della complessità di ogni singolo nodo con algoritmo di Quine-McCluskey.

Minimizzazione di circuiti combinatori multi-livello tramite SIS

Una volta descritto il circuito desiderato nel formato BLIF è possibile utilizzare i seguenti comandi per effettuare la minimizzazione. Per ulteriori approfondimenti sull'uso dei comandi di ottimizzazione e sulle opzioni disponibili, si veda il relativo help fornito da SIS tramite il comando `help comando`.

- sweep	Esegue l'operazione di sweep;
- eliminate *n*	Esegue l'operazione di eliminazione rimuovendo i nodi tali che la loro rimozione non aumenti il numero di letterali di una quantità superiore a "n" (numero intero). Per eliminare i nodi che sono utilizzati una sola volta utilizzare il valore -1;
- resub *lista*	Esegue l'operazione di scomposizione dei nodi indicati nella lista. Se la lista non viene specificata, la sostituzione viene eseguita per tutti i nodi della rete. I nodi nella lista devono essere specificati con il nome della loro uscita e vanno intervallati tra loro da uno spazio;
- fx	Esegue l'operazione di estrazione;
- full_simplify	Esegue l'operazione di semplificazione su ogni nodo della rete;
- source *script*	Carica lo script ed esegue tutti i comandi contenuti al suo interno. Lo script che fornisce generalmente i risultati migliori è `script.rugged;`
- set autoexec *comando*	Stampa automaticamente il risultato del comando specificato dopo l'esecuzione di un qualunque altro comando.

Esercizi

Esercizio 1: Eseguire la minimizzazione multi-livello usando lo script `script.rugged` su tutti i circuiti realizzati durante la seconda esercitazione ("Ottimizzazione esatta a 2 livelli"). Quale dei 4 dispositivi viene maggiormente ottimizzato?

Esercizio 2: Descrivere nel formato blif il circuito rappresentato dalla funzione booleana *(x, v, w, z) = f(a, b, c, d, e)* descritta dai seguenti nodi:

$f=\bar{a}\,be+cd+a\bar{c}\,d$
$g=de+abc+a\bar{c}\,d$
$h=ae+cd+ab$
$i=g+\bar{h}$
$l=ag+bc\,\bar{g}+ae$
$m=f+i+bc$
$n=l+a\,\bar{i}$
$o=m+a+\bar{b}\,e$
$x=f$
$v=o$
$w=n$
$z=l$

Eseguire la minimizzazione multi-livello in due modi:

1. usando lo script `script.rugged`.
2. usando una sequenza greedy user-defined in modo da migliorare ulteriormente il risultato ottenuto con lo script.rugged, distinguendo l'ottimizzazione per area o ritardo.