

# LABORATORIO DI ARCHITETTURA DEI CALCOLATORI

## lezione n° 7. Esercitazioni

Prof. Rosario Cerbone

rosario.cerbone@libero.it

<http://digilander.libero.it/rosario.cerbone>

a.a. 2005-2006

## **Esercizio 1**

- Scrivere la tabella delle verità e rappresentare nel formato blif il circuito digitale corrispondente alla seguente espressione booleana:

$$f(x,y,z,v)=!xyz+!x!y+y!zv+xv!y+yz.$$

- Eseguire l'ottimizzazione con SIS usando il comando `full_simplify`.
- Fornire il grado di ottimizzazione confrontando l'area (numero porte logiche) e il ritardo (lunghezza cammino critico) del circuito prima e dopo l'ottimizzazione.
- Visualizzare l'espressione booleana corrispondente al circuito ottimizzato con il comando `write_eqn` e confrontarla con quella che si ottiene manualmente usando il metodo di Quine-McCluskey.

## ***Esercizio 1. Soluzione***

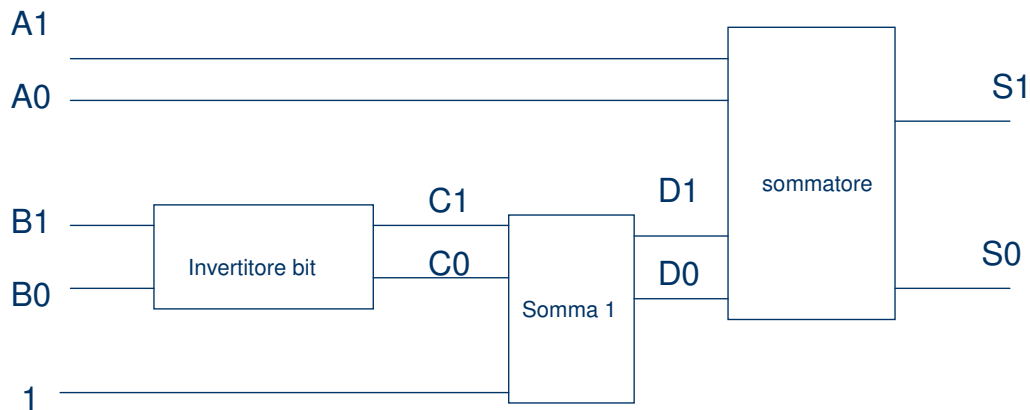
- ```
.model esercizio1
.inputs x y z v
.outputs f
.names x y z v f
0000 1
0001 1
0010 1
0011 1
0101 1
0110 1
0111 1
1001 1
1011 1
1101 1
1110 1
1111 1
.end
```

## ***Esercizio 2***

- Descrivere in formato blif il circuito digitale che esegue la sottrazione di 2 numeri binari a 2 bit rappresentati in complemento a 2. Eseguire l'ottimizzazione con SIS. Visualizzare l'espressione booleana corrispondente al circuito prima e dopo l'ottimizzazione e fornire il grado di ottimizzazione.

## Esercizio 2

- La schema del circuito è il seguente:



## Esercizio 2

```
• .model sottrazione2
  .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
```

```
• # 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
```

## ***Esercizio 3***

- Descrivere nel formato blif un circuito digitale che riceve in input una sequenza di 5 bit che rappresenta la codifica binaria di una lettera dell'alfabeto italiano considerando di associare in ordine crescente i numeri dallo 0 al 21 alle lettere dalla A alla Z (A=0, B=1, ..., Z=21).
- Il circuito ha un solo bit in uscita che vale 1 se e solo se la sequenza di input corrisponde ad una vocale.
- La tabella delle verità risulterà parzialmente specificata dal momento che con 5 cifre binarie è possibile rappresentare i numeri dallo 0 al 31.

## ***Esercizio 3***

- Ottimizzare il circuito con SIS nel caso in cui alle righe della tabella delle verità non specificate venga associato:
  - - il valore 0,
  - - il valore 1,
  - - il valore don't care.
- Quale dei tre circuiti è maggiormente ottimizzato?

## Esercizio 3. Soluzione

- #il don't care set è stato considerato come appartenente all'off-set

```
.model vocali0
.inputs i4 i3 i2 i1 i0
.outputs o

.names i4 i3 i2 i1 i0 o
00000 1
00100 1
01000 1
01100 1
10010 1

.end
```

## Esercizio 3. Soluzione

- #il don't care set è stato considerato come appartenente all'on-set

```
.model vocali1
.inputs i4 i3 i2 i1 i0
.outputs o

.names i4 i3 i2 i1 i0 o
00000 1
00100 1
01000 1
01100 1
10010 1
10101 1
10110 1
10111 1
11000 1
11001 1
11010 1
11011 1
11100 1
11101 1
11110 1
11111 1

.end
```

## Esercizio 3. Soluzione

- #il don't care set è stato considerato come appartenente al don't care-set

- .model vocali1  
.inputs i4 i3 i2 i1 i0 o  
.outputs o

```
.names i4 i3 i2 i1 i0 o  
00000 1  
00100 1  
01000 1  
01100 1  
10010 1
```

```
.exdc  
.names i4 i3 i2 i1 i0 o  
10101 1  
10110 1  
10111 1  
11000 1  
11001 1  
11010 1  
11011 1  
11100 1  
11101 1  
11110 1  
11111 1  
end
```

## Esercizio 4

- Descrivere nel formato blif un circuito che calcola la parte intera della radice quadrata delle 10 cifre decimali rappresentate da numeri binari a 4 bit.
- L'output sarà una sequenza composta da 2 bit.
- La tabella delle verità del circuito risulterà parzialmente specificata dal momento che con 4 bit è possibile rappresentare i numeri dallo 0 al 15.

## Esercizio 4

- Ottimizzare il circuito con SIS nel caso in cui alle righe della tabella delle verità non specificate venga associato:
  - - il valore 0,
  - - il valore 1,
  - - il valore don't care.
- Quale dei tre circuiti è maggiormente ottimizzato?

## Esercizio 4. Soluzione

- #il don't care set è stato considerato come appartenente all'off-set

```
.model radice_intera_0
.inputs i3 i2 i1 i0
.outputs o1 o0

.names i3 i2 i1 i0 o0
0001 1
0010 1
0011 1
1001 1

.names i3 i2 i1 i0 o1
0100 1
0101 1
0110 1
0111 1
1000 1
1001 1

.end
```

## Esercizio 4. Soluzione

- #il don't care set è stato considerato come appartenente all'on-set

- ```
.model radice_intera_1
.inputs i3 i2 i1 i0
.outputs o1 o0
```

```
.names i3 i2 i1 i0 o0
0001 1
0010 1
0011 1
1001 1
1010 1
1011 1
1100 1
1101 1
1110 1
1111 1
```

```
.names i3 i2 i1 i0 o1
0100 1
0101 1
0110 1
0111 1
1000 1
1001 1
1010 1
1011 1
1100 1
1101 1
1110 1
1111 1

.end
```

## Esercizio 4. Soluzione

- ```
.model radice_intera_0
.inputs i3 i2 i1 i0
.outputs o1 o0
```

```
.names i3 i2 i1 i0 o0
0001 1
0010 1
0011 1
1001 1
```

```
.names i3 i2 i1 i0 o1
0100 1
0101 1
0110 1
0111 1
1000 1
1001 1
```

```
.exdc
.names i3 i2 i1 i0 o0
1010 1
1011 1
1100 1
1101 1
1110 1
1111 1

.names i3 i2 i1 i0 o1
1010 1
1011 1
1100 1
1101 1
1110 1
1111 1

.end
```



## Esercizi

### lezione 6

- **Esercizio 4:** Descrivere nel formato blif il circuito rappresentato dalla funzione
- booleana  $(x,v,w,z) = f(a,b,c,d,e)$  descritto dai seguenti nodi:
- $f = !abe + cd + a!cd$
- $g = ed + acb + a!cd$
- $h = ae + cd + ab$
- $i = g + !h$
- $l = ag + bclg + ae$
- $m = f + i + cb$
- $n = l + !ia$
- $o = m + a + !be$
- $x = f$
- $v = o$
- $z = l$
- $w = n$
- Eseguire la minimizzazione multi-livello usando lo script script.rugged.
- Qual è il cammino con maggior ritardo prima e dopo aver eseguito il comando reduce\_depth per diminuire il ritardo dei cammini critici?

## Esercizi

### lezione 6. Soluzione

```
.model esercizio4
.inputs a b c d e
.outputs x v z w

.names a b c d e f
01--1 1
--11- 1
1-01- 1

.names a b c d e g
---11 1
111-- 1
1-01- 1

.names a b c d e h
1---1 1
--11- 1
11--- 1

.names g h i
1- 1
-0 1

.names a b c e g l
1---1 1
-11-0 1
1--1- 1

.names b c i f m
---1 1
--1- 1
11-- 1
.names a i l n
--1 1
10- 1
.names a b e m o
---1 1
1--- 1
-01- 1
.names f x
1 1
.names o v
1 1
.names l z
1 1
.names n w
1 1

.end
```

## ***Esercizi*** lezione 7.

- Descrivere nel formato blif il circuito rappresentato dalla funzione booleana  $(x,y,w,z) = f(a,b,c,d,e)$  descritto dai seguenti nodi:
- $v = !ad + bd + !cd + a !e$
- $p = ce + de$
- $r = p + !a$
- $s = r + !b$
- $q = a + b$
- $u = !qc + q!c + qc$
- $t = ac + ad + bc + bd + e$
- $w = v$
- $x = s$
- $z = u$
- $y = t$
  
- Eseguire la minimizzazione multi-livello usando lo script `script.rugged`

## ***Esercizi*** lezione 7. Indicazioni

- Caricare il file blif
- Verificarne la correttezza con `wl`
- Applicare lo script con il comando `source -x` dopo `set autoexec ps`
- Verificare le successive semplificazioni

## **Esercizi**

### lezione 7. Soluzione

```
• .model rete5in4out
.inputs a b c d e
.outputs w x z y

.names c e d p
11- 1
-11 1

.names a b q
1- 1
-1 1

.names p a r
1- 1
-0 1

.names r b s
1- 1
-0 1

.names a c d b e t
11--- 1
1-1-- 1
-1-1- 1
--11- 1
----1 1

.names q c u
01 1
10 1
11 1

.names a d b c e v
01--- 1
-11-- 1
-1-0- 1
1---1 1

.names v w
1 1

.names s x
1 1

.names t y
1 1

.names u z
1 1

.end
```

## **Esercizio 7.1**

- Dato un numero intero  $n$ , ( $0 \leq n < 32$ ), in forma binaria, progettare un circuito che preveda in uscita un bit che segnali la presenza in ingresso di un quadrato perfetto ed un numero opportuno di linee che restituiscano la radice quadrata del numero in ingresso qualora il bit di segnalazione sia attivo, altrimenti 0.

## ***Esercizio 7.2***

- Realizzare un circuito a quattro ingressi ABCD ed un codice in uscita a 2 bit con le seguenti assegnazioni:
- 00 se la parola in ingresso è divisibile per 2;
- 01 se la parola in ingresso è divisibile per 3;
- 10 se la parola di ingresso è divisibile per 5;
- 11 altro.

## ***Esercizio 7.3***

- Si realizzi una rete con 5 ingressi ed una uscita che valga 1 se il numero degli "1" presenti in ingresso supera il numero degli "0".