Reti Logiche A - A.A. 2010-2011
Ingegneria Informatica
Facoltà di Ingegneria dell'Informazione
Prof.ssa Cristiana Bolchini
Appello del **05/09/2011**

CONSEGNARE COMPILATO

Cognome	**Nome**	**Matricola**	**Voto:** . . . /**30**

Quesito:	1	2	3	4	5	Tot.
Max:	5	5	7	8	5	30
Punti:						

Quesito 1 (5 punti) Punteggio ottenuto: . . . /5

Dati $A = +33.25_{10MS}$ e $B = +17.5_{10MS}$ codificati in base dieci notazione modulo e segno, si convertano i valori in base due, notazione IEEE 754 singola precisione, quindi si effettui l'operazione $A + B$ e si riporti il risultato.
Nello svolgimento si mostrino tutti i passaggi fatti.

Traccia soluzione:

Conversione di $A = +33.25_{10MS}$

Conversione della parte intera $33_{10} = 100001_2$.

Conversione della parte decimale, mediante moltiplicazioni ripetute per la base: $0.25_{10} = 01000\ldots0_2$

Rappresentazione in virgola fissa: $100001.01000\ldots0$.

Per portarsi nella forma $(1 + M) \times 2^e$ è necessario "spostare" la virgola di 5 posizioni, ossia:

$100001.01000\ldots0 = 1.0000101000\ldots0 = 1 + 0000101000\ldots0 \times 2^5$

La notazione è in eccesso e dunque nel campo esponente va rappresentato il valore $5 + 127 = 132$. $132_{10} = 10000100_2$.

Quindi, la sequenza di 32 bit che rappresenta la codifica in base 2, notazione IEEE 754, singola precisione è la seguente.

$A = 0\ 10000100\ 0000101000\ldots0$

Conversione di $B = +17.5_{10MS}$

Conversione della parte intera $17_{10} = 10001_2$.

Conversione della parte decimale, mediante moltiplicazioni ripetute per la base: $0.5_{10} = 1000\ldots0_2$

Rappresentazione in virgola fissa: $10001.1000\ldots0$.

Per portarsi nella forma $(1 + M) \times 2^e$ è necessario "spostare" la virgola di 4 posizioni verso destra, ossia:

$10001.1000\ldots0 = 1.00011000\ldots0 = 1 + 00011000\ldots0 \times 2^4$

La notazione è in eccesso e dunque nel campo esponente va rappresentato il valore $4 + 127 = 131$. $131_{10} = 10000011_2$.

Quindi, la sequenza di 32 bit che rappresenta la codifica in base 2, notazione IEEE 754, singola precisione è la seguente.

$B = 0\ 10000011\ 00011000\ldots0$

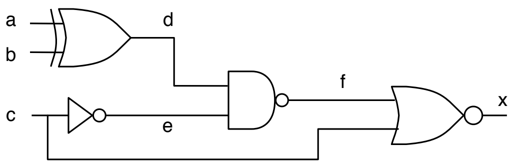
Differenza tra gli esponenti: $10000100 - 10000011 = 1$. È quindi necessario effettuare uno scorrimento a destra del valore di B di 1 posizione, per ottenere operandi con ugual esponente, per poi procedere con l'operazione di somma. Gli operandi sono concordi, quindi si effettuerà poi la somma tra le mantisse.

	A + B	
$\|A\|$	1.00001010000000000000000	+
$\|B\| \times 2^{-1}$	0.10001100000000000000000	
	1.10010110000000000000000	

$A + B = 0\ 10000100\ 10010110000000000000000_2$

Quesito 2 (5 punti) Punteggio ottenuto: . . . /5

Si consideri il circuito di seguito riportato che realizza la funzione $x(a, b, c)$.



Si chiede di

- ottimizzare la realizzazione derivando la tabella delle verità ed effettuando poi la sintesi ottima a due livelli;
- a partire dalla soluzione individuata al punto precedente, manipolare ulteriormente l'espressione algebrica ottenuta applicando le regole dell'algebra booleana e facendo uso anche di operatori come xor e xnor, al fine di ridurre il numero totale degli operatori nella forma finale utilizzando operatori a due ingressi.

Traccia soluzione: La tabella delle verità del dispositivo è di seguito riportata.

ingresso			interni			uscita
a	b	c	d	e	f	x
0	0	0	0	1	1	0
0	0	1	0	0	1	0
0	1	0	1	1	0	1
0	1	1	1	0	1	0
1	0	0	1	1	0	1
1	0	1	1	0	1	0
1	1	0	0	1	1	0
1	1	1	0	0	1	0

La mappa di Karnaugh che se ne deriva per effettuare la sintesi ottima esatta è la seguente:

$a \backslash bc$	00	01	11	10
0	0	0	0	1
1	1	0	0	0

Da cui si deriva: $x = !ab!c + a!b!c$

Applicando le regole dell'algebra booleana per ridurre ulteriormente questa espressione utilizzando solo operatori a due ingressi si ottiene

$$\begin{aligned} x &= !ab!c + a!b!c \\ &= (!ab + a!b)!c \\ &= (a \; xor \; b)!c \end{aligned}$$

Quesito 3 (7 punti) Punteggio ottenuto: . . . /7

Data la tabella degli stati di una FSM sincrona, sotto riportata, si trovi la macchina minima riportandone la tabella degli stati. In relazione a tale macchina trovata, si selezioni uno stato di RESET che non provochi ulteriori riduzioni e se ne effettui una realizzazione ottima, utilizzando bistabili di tipo FFD. Si arrivi fino a disegnare il circuito finale completo. **Risolvere il problema adottando algoritmi e non procedimenti ad occhio, e mostrare con cura tutti i passaggi.**

	0	1
S0	S4/0	S1/0
S1	S4/0	S2/0
S2	S3/0	S2/0
S3	S5/0	S1/1
S4	S5/0	S1/0
S5	S5/0	S1/0
S6	S4/0	S7/1
S7	S3/0	S7/0

Traccia soluzione:

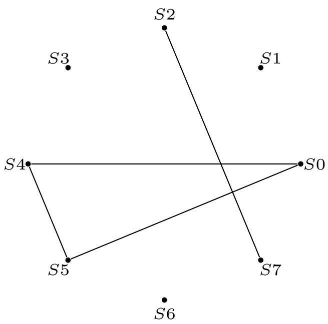
Ricerca di stati indistinguibili per una possibile riduzione del numero degli stati tramite la regola di Paull-Unger.

$S1$	$S1, S2$						
$S2$	$S3, S4$ $S1, S2$	$S3, S4$					
$S3$	$\times$	$\times$	$\times$				
$S4$	$S4, S5$	$S4, S5$ $S1, S2$	$S3, S5$ $S1, S2$	$\times$			
$S5$	$S4, S5$	$S4, S5$ $S1, S2$	$S3, S5$ $S1, S2$	$\times$	$\sim$		
$S6$	$\times$	$\times$	$\times$	$S4, S5$ $S1, S7$	$\times$	$\times$	
$S7$	$S3, S4$ $S1, S7$	$S3, S4$ $S2, S7$	$\sim$	$\times$	$S3, S5$ $S1, S7$	$S3, S5$ $S1, S7$	$\times$
	$S0$	$S1$	$S2$	$S3$	$S4$	$S5$	$S6$

$S1$	$\times$						
$S2$	$\times$	$\times$					
$S3$	$\times$	$\times$	$\times$				
$S4$	$S4, S5$	$S4, S5$ $S1, S2$	$\times$	$\times$			
$S5$	$S4, S5$	$S4, S5$ $S1, S2$	$\times$	$\times$	$\sim$		
$S6$	$\times$	$\times$	$\times$	$S4, S5$ $S1, S7$	$\times$	$\times$	
$S7$	$\times$	$\times$	$\sim$	$\times$	$\times$	$\times$	$\times$
	$S0$	$S1$	$S2$	$S3$	$S4$	$S5$	$S6$

Al primo passo si propagano le distinguibilità $S1, S2$, $S3, S4$ e $S3, S5$.

$S1$	$\times$						
$S2$	$\times$	$\times$					
$S3$	$\times$	$\times$	$\times$				
$S4$	$\sim$	$\times$	$\times$	$\times$			
$S5$	$\sim$	$\times$	$\times$	$\times$	$\sim$		
$S6$	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	
$S7$	$\times$	$\times$	$\sim$	$\times$	$\times$	$\times$	$\times$
	$S0$	$S1$	$S2$	$S3$	$S4$	$S5$	$S6$



Al secondo passo si propagano le distinguibilità $S1, S7$ e $S1, S2$.

Classi di equivalenza: $\alpha = \{S0, S4, S5\}$, $S1$, $\beta = \{S2, S7\}$, $S3$, $S6$.

La tabella della macchina ridotta equivalente è la seguente:

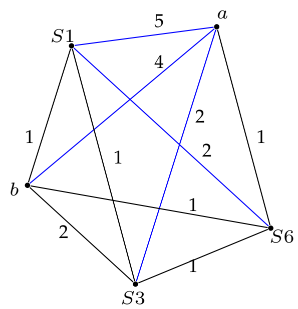
	0	1
α	$\alpha/0$	S1/0
S1	$\alpha/0$	$\beta/0$
β	S3/0	$\beta/0$
S3	$\alpha/0$	S1/1
S6	$\alpha/0$	$\beta/1$

Si seleziona come stato di RESET lo stato S6, altrimenti non raggiungibile (e quindi darebbe luogo ad una ulteriore riduzione della macchina).

Per effettuare la sintesi ottima, poiché si utilizzano bistabili FFD è necessario adottare politiche per l'assegnamento dello stato.

	priorità alta	priorità bassa	tot
α, S1	1	4	5
α, β	0	4	4
α, S3	2	0	2
α, S6	1	0	1
S1, β	1	0	1
S1, S3	1	0	1
S1, S6	2	0	2
β, S3	0	2	2
β, S6	1	0	1
S3, S6	1	0	1

Il grafo dei vincoli delle adiacenze, con gli archi pesati è il seguente:



Un assegnamento che consente di massimizzare le adiecenze è il seguente:
α = 000
S1 = 001
β = 100
S3 = 010
S6 = 101

L'assegnamento scelto dà luogo alla tabella degli stati assegnati e tabella delle transizioni (poichè si utilizzano bistabili di tipo D) di seguito riportata. A partire da tale tabella si derivano le 3 mappe di Karnaugh per la sintesi della funzione stato prossimo e della funzione d'uscita.

Tabella delle transizioni

$q_1q_2q_3$	0	1
000	000/0	001/0
001	000/0	100/0
010	000/0	001/1
100	010/0	100/0
101	000/0	100/1

D_1

q_1q_2 \ q_3i	00	01	11	10
00	0	0	1	0
01	0	0	–	–
11	–	–	–	–
10	0	1	1	0

D_2

q_1q_2 \ q_3i	00	01	11	10
00	0	0	0	0
01	0	0	–	–
11	–	–	–	–
10	1	0	0	0

D_3

q_1q_2 \ q_3i	00	01	11	10
00	0	1	0	0
01	0	1	–	–
11	–	–	–	–
10	0	0	0	0

Z

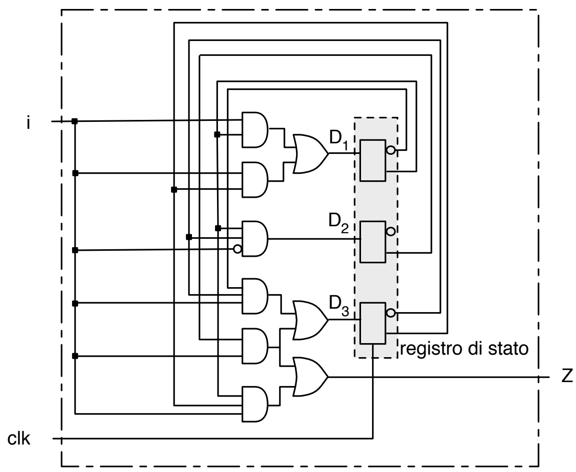
q_1q_2 \ q_3i	00	01	11	10
00	0	0	0	0
01	0	1	–	–
11	–	–	–	–
10	0	0	1	0

$D_1 = q_1 i + q_3 i$

$D_2 = q_1 !q_3 !i$

$D_3 = !q_1 !q_3 i + q_2 i$

$Z = q_2 i + q_1 q_3 i$



Quesito 4 (8 punti) Punteggio ottenuto: . . . /8

Si desidera realizzare mediante una PLA una macchina sequenziale sincrona a 3 ingressi a, b e c e 1 uscita Y, tre variabili di stato Q_1, Q_2 e Q_3, realizzata mediante bistabili di tipo D (variabili di stato prossimo D_1, D_2 e D_3). La funzione di uscita λ è descritta tramite una rete multilivello (V_1, V_2 e V_3 sono nodi interni della rete), qui di seguito riportata:

$$
\begin{aligned}
V_1 &= ab + !Q_1Q_2 + !(!a + !b + !Q_2) + !abc + !Q_1!Q_2 \\
V_2 &= !V_1a!Q_2 + !Q_2!V_1 + Q_1V_1 \\
V_3 &= a + c \\
Y &= V_2 + c!Q_2 + Q_3V_3
\end{aligned}
$$

La funzione di stato prossimo δ è descritta tramite il ciclo di conteggio di seguito riportato:

Q_3	Q_2	Q_1
0	1	0
1	1	0
1	1	1
0	1	1
0	0	1

Per realizzare la macchina si svolgano i seguenti punti:

1. Ottimizzare, dove necessario, le espressione della rete multilivello della funzione λ **in modo da ottenere forme SOP ottime da realizzarsi mediante PLA**. Per effettuare l'ottimizzazione si devono utilizzare le proprietà e i teoremi dell'algebra booleana, indicandone nomi e passaggi. Riportare le equazioni della rete multilivello, ed il suo costo.
2. Sintetizzare in modo ottimo il contatore utilizzando bistabili FFD, riportando le equazioni della funzione stato prossimo δ (variabili di stato prossimo D_1, D_2 e D_3).
3. Realizzare le reti logiche combinatorie λ e δ tramite un'unica PLA, indicando esplicitamente i termini prodotto del piano AND e le espressioni relative al piano OR, utilizzando il corretto nome dei segnali generati (in relazione alla nomenclatura del dispositivo complessivo).
4. Disegnare lo schema logico **complessivo** (come un'unica rete e non con tutte le parti separate) del dispositivo programmato che realizza la macchina sequenziale.

Traccia soluzione:

Funzione λ: ottimizzazione multilivello. Applicazione proprietà dell'algebra:

$$
\begin{aligned}
V_1 &= ab + !Q_1Q_2 + !(!a + !b + !Q_2) + !abc + !Q_1!Q_2 \\
&= ab + !(!a + !b + !Q_2) + !abc + !Q_1(Q_2 + !Q_2) \\
&= ab + !Q_1 + abQ_2 + !abc \\
&= ab(1 + Q_2) + !Q_1 + !abc \\
&= ab + !Q_1 + !abc \\
&= !Q_1 + b(a + !ac) \\
&= !Q_1 + b(a + c) \\
&= !Q_1 + ab + bc \\
V_2 &= !V_1a!Q_2 + !Q_2!V_1 + Q_1V_1 \\
&= !V_1!Q_2(a + 1) + Q_1V_1 \\
&= !V_1!Q_2 + Q_1V_1 \\
V_3 &= a + c \\
Y &= V_2 + c!Q_2 + Q_3V_3
\end{aligned}
$$

Sintesi del contatore:

Q_3	Q_2	Q_1	D_3	D_2	D_1
0	1	0	1	1	0
1	1	0	1	1	1
1	1	1	0	1	1
0	1	1	0	0	1
0	0	1	0	1	0

Mappe di Karnaugh derivate:

D_3

q_3 \ q_2q_1	00	01	11	10
0	–	0	0	1
1	–	–	0	1

D_2

q_3 \ q_2q_1	00	01	11	10
0	–	1	0	1
1	–	–	1	1

D_1

q_3 \ q_2q_1	00	01	11	10
0	–	0	1	0
1	–	–	1	1

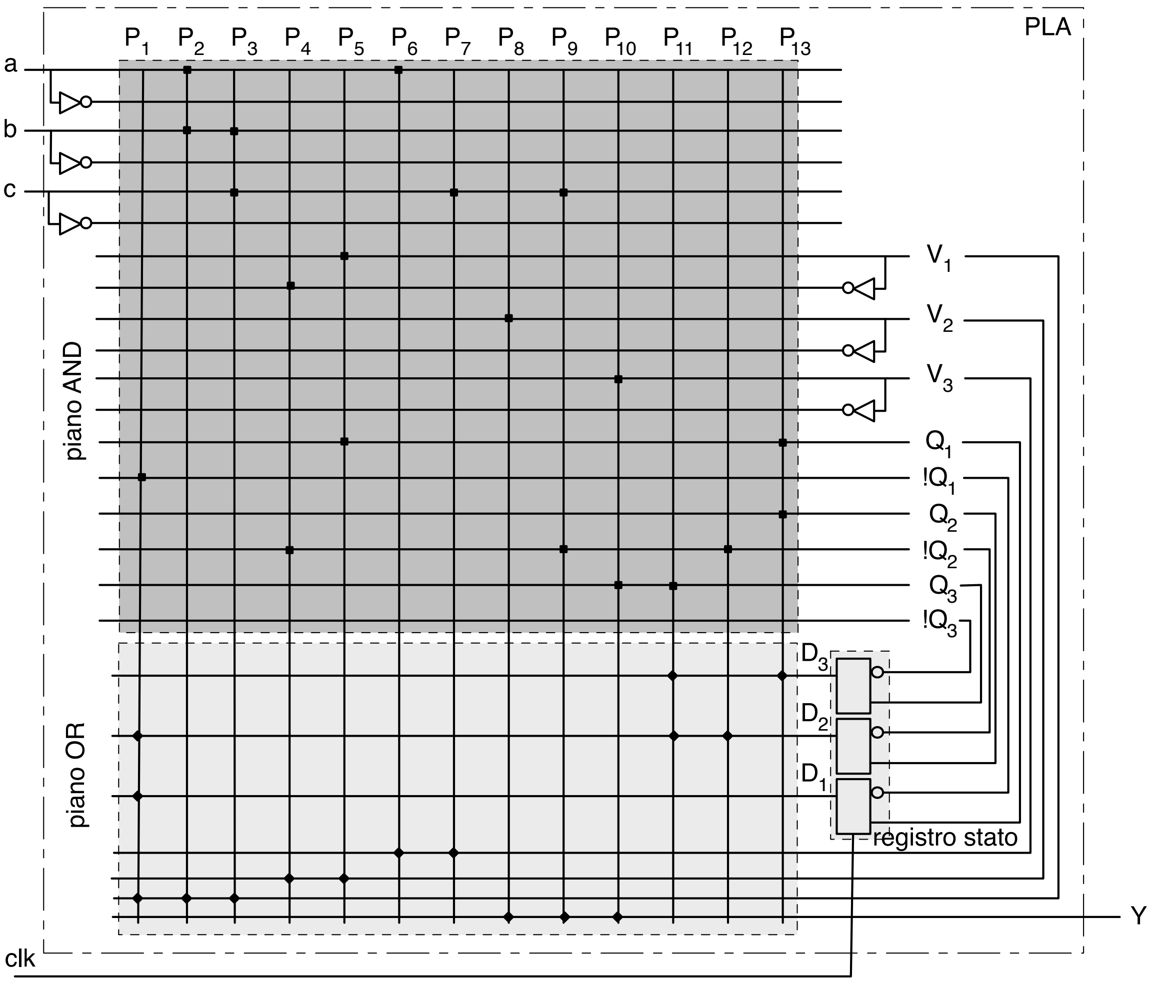
$D_3 = !Q_1$ $D_2 = Q_3 + !Q_2 + !Q_1$ $D_1 = Q_3 + Q_2Q_1$

Le equazioni algebriche da considerare per la realizzazione della rete logica combinatoria tramite PLA di seguito riportate (reti λ e δ) e da esse si derivano i termini prodotto per la realizzazione del piano AND.

$$
\begin{aligned}
V_1 &= !Q_1 + ab + bc \\
V_2 &= !V_1!Q_2 + Q_1V_1 \\
V_3 &= a + c \\
Y &= V_2 + c!Q_2 + Q_3V_3 \\
D_1 &= !Q_1 \\
D_2 &= Q_3 + !Q_2 + !Q_1 \\
D_3 &= Q_3 + Q_2Q_1
\end{aligned}
$$

$$
\begin{aligned}
P_1 &= !Q_1 \\
P_2 &= ab \\
P_3 &= bc \\
P_4 &= !V_1!Q_2 \\
P_5 &= Q_1V_1 \\
P_6 &= a \\
P_7 &= c \\
P_8 &= V_2 \\
P_9 &= c!Q_2 \\
P_{10} &= Q_3V_3 \\
P_{11} &= Q_3 \\
P_{12} &= !Q_2 \\
P_{13} &= Q_2Q_1
\end{aligned}
$$

$$
\begin{aligned}
V_1 &= P_1 + P_2 + P_3 \\
V_2 &= P_4 + P_5 \\
V_3 &= P_6 + P_7 \\
Y &= P_8 + P_9 + P_{10} \\
D_1 &= P_1 \\
D_2 &= P_{11} + P_{12} + P_1 \\
D_3 &= P_{11} + P_{13}
\end{aligned}
$$



Quesito 5 (5 punti) Punteggio ottenuto: . . . /5

Data la descrizione di circuito in VHDL riportata di seguito, disegnare il circuito risultante come una unica rete, facendo uso di componenti elementari (multiplexer, bistabili, sommatori, · · ·) e logica sparsa (porte logiche). Indicare per ogni tipo di componente quante unità vengono allocate (ad esempio, 3 sommatori).

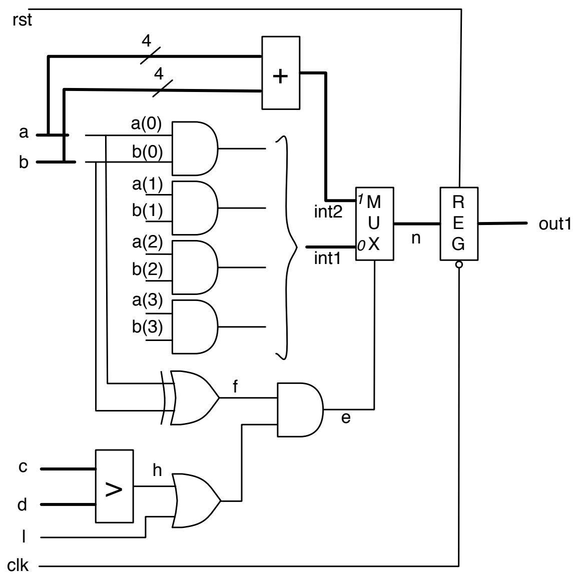
```
entity esercizio1 is port(
  a, b, c, d  : in  std_logic_vector(3 downto 0);
  l, clk, rst : in  std_logic;
  out1        : out std_logic_vector(3 downto 0));
end esercizio1;

architecture mixed of esercizio1 is
  signal n : std_logic_vector(3 downto 0);
  signal e, f, g, h: std_logic;
begin
  PROC1 : process(a, b, e)
  begin
    if e = '1' then
      n <= a + b;
    else
      n <= a and b;
    end if;
  end process;

  e <= f and g;
  h <= '0' when (c > d) else '1';
  g <= h or l;
  f <= a(0) xor b(0);

  PROC2 : process (rst, clk)
  begin
    if(rst = '1') then
      out1 <= "0000";
    elsif (clk = '0' and clk'event) then
      out1 <= n;
    end if;
  end process;
end mixed;
```

Traccia soluzione:



Vengono utilizzate:

- 5 porte AND
- 1 porta OR
- 1 porta XOR
- 1 MUX 2x1 con dati a 4 bit
- 1 SOMMATORE con dati a 4 bit
- 1 COMPARATORE con dati a 4 bit
- 1 REGISTRO di 4 bit (4 FFD) con clock attivo basso e reset asincrono attivo alto.