



Politecnico di Milano

Dipartimento di Elettronica e Informazione

prof.ssa Anna Antola
prof. Fabrizio Ferrandi

prof.ssa Cristiana Bolchini

Reti Logiche A – Esame del 19 febbraio 2007

Matricola _____

Cognome _____ Nome _____

Istruzioni

- Scrivere solo sui fogli distribuiti. Non separare questi fogli.
- È vietato portare all'esame libri, eserciziari, appunti e calcolatrici. Chiunque venga trovato in possesso di documentazione relativa al corso – anche se non strettamente attinente alle domande proposte – vedrà annullata la propria prova.
- Non è possibile lasciare l'aula conservando il tema della prova in corso.
- Tempo a disposizione: 2h:30m.

Esercizio 1 (5 punti) _____

Esercizio 2 (6 punti) _____

Esercizio 3 (7 punti) _____

Esercizio 4 (7 punti) _____

Esercizio 5 (4 punti) _____

Esercizio 6 (3 punti) _____

Con Soluzioni

Esercizio n. 1

Data la seguente tabella di copertura:

	F1								F2							
	m1	m2	m3	m4	m5	m9	m10	m11	m12	m13	m14	m15	m17	m18		
A	x							x							4	
B	x				x		x	x				x		x	4	
C					x				x	x	x				3	
D		x		x			x	x	x	x	x		x		3	
E			x			x		x	x						2	
F											x		x		2	
G	x		x		x	x						x			2	
H				x							x		x			

Si trovi una copertura minima utilizzando il metodo di Quine McCluskey. Descrivere con chiarezza ogni singolo passo svolto per arrivare alla soluzione.

Esercizio n. 2

Sia data una macchina sequenziale sincrona con ingressi (a, b, c, d, e) e uscita (Y), la cui rete combinatoria che realizza le funzioni λ (uscita) e δ (stato prossimo) è rappresentata dalla seguente rete multilivello:

$$V_1 = ((\overline{a+b})c + \overline{ab})d + \overline{d}(a+b) + \overline{a+b+bc}$$

$$V_2 = \overline{Q}_1 + \overline{d} + V_1$$

$$V_3 = \overline{b}Q_2 + \overline{d}Q_2 + V_2$$

$$Y = V_1 + \overline{acd}$$

$$D_1 = \overline{abcd} + \overline{acd} + \overline{ab} + \overline{b}Q_2 + \overline{a}Q_2 + \overline{abc}Q_2$$

$$D_2 = \overline{acd} + \overline{c}Q_1 + \overline{b}Q_2$$

Dove

- Q1 e Q2 rappresentano le variabili stato presente
 - L'espressione associata a Y costituisce la funzione λ .
 - Le espressioni associate a D1 e D2 costituiscono la funzione δ
 - V1, V2 e V3 sono nodi intermedi
1. Applicare in sequenza alla rete multilivello le trasformazioni sotto indicate e rispondere alle domande dove richiesto. Disegnare anche il modello della rete finale.
Nota Bene: per ogni trasformazione è **obbligatorio** riportare il **risultato della trasformazione** e **mostrare chiaramente tutti i passaggi** effettuati per ottenere il risultato stesso.
- a) **COST()**: Calcolo del numero di letterali. La funzione COST() calcola il costo in letterali indipendentemente dalla forma (SOP o Multilivello) delle espressioni algebriche dei nodi.
- b) **SIMPLIFY(V₁)**: Minimizzazione a due livelli di V₁.
- c) **FACTOR(V₃)**: Fattorizzazione del nodo V₃.
- d) **COST()**: Calcolo del numero di letterali.
- e) **SUBSTITUTE(V₁)**: Inserisce V₁ in tutti i nodi della rete dove è possibile. La trasformazione ha effetto solo se produce un guadagno in termini di letterali.
- f) **[V₄] = EXTRACT(D₁, D₂)**: Estrazione di un fattore comune a D₁ e D₂. Il nodo V₄ derivato dall'estrazione può essere un nuovo nodo o un nodo già presente nella rete.
- g) **COST()**: Calcolo del numero di letterali.
2. Realizzare la rete combinatoria tramite PLA. Si indichino esplicitamente i termini prodotto del piano AND e le espressioni relative al piano OR, si disegni anche lo schema logico delle interconnessioni da programmare.

Soluzione:

- a) **COST()**: **xx**
- b) **SIMPLIFY(V₁)**: Minimizzazione a due livelli di V₁. Tramite mappe di Karnaugh o manipolazione algebrica ottima, il risultato della minimizzazione è
- $$V_1 = ((\overline{a+b})c + \overline{ab})d + \overline{d}(a+b) + (\overline{a+b+bc})$$
- $$V_1 = \overline{b} + \overline{d}$$
- c) **FACTOR(V₃)**: Fattorizzazione del nodo V₃.
- $$V_3 = \overline{b}Q_2 + \overline{d}Q_2 + V_2$$

Il risultato della fattorizzazione è:

$$V_3 = (\overline{b} + \overline{d})Q_2 + V_2$$

- d) **COST()**: **xx**
- e) **SUBSTITUTE(V₁)**: Inserisce V₁ in tutti i nodi della rete dove è possibile. La trasformazione ha effetto solo se produce un guadagno in termini di letterali.

L'unico nodo che presenta già V₁ come fattore (non è quindi necessario eseguire la divisione algebrica) è V₃. L'espressione di V₃ diventa:

$$V_3 = V_1Q_2 + V_2$$

e il costo del nodo (l'unico modificato nella rete) diminuisce di 2 letterali. Quindi la trasformazione ha effetto.

- f) **[V₄] = EXTRACT(D₁, D₂)**: Estrazione di un fattore comune a D₁ e D₂.

$$D_1 = \overline{abcd} + \overline{acd} + \overline{ab} + \overline{b}Q_2 + \overline{a}Q_2 + \overline{abc}Q_2$$

$$D_2 = \overline{acd} + \overline{c}Q_1 + \overline{b}Q_2$$

e quindi

$$V_4 = \overline{acd} + \overline{b}Q_2$$

$$D_1 = \overline{abcd} + \overline{ab} + V_4 + \overline{a}Q_2 + \overline{abc}Q_2$$

$$D_2 = \overline{c}Q_1 + V_4$$

- g) **COST()**: **xx**

2) Termini prodotto e sezione OR da realizzare (l'asterisco indica le funzioni OR che vanno retroazionate. Per D1 e D2 il segnale portato in retroazione è l'uscita Q del bistabile D corrispondente all'ingresso omonimo)

Data la tabella degli stati di seguito riportata, si calcoli la tabella degli stati della macchina ridotta, mostrando i passaggi fatti (analisi di compatibilità, classi di compatibilità massima e prima, copertura minimale). È necessario adottare algoritmi, non è consentito risolvere il problema "a occhio".

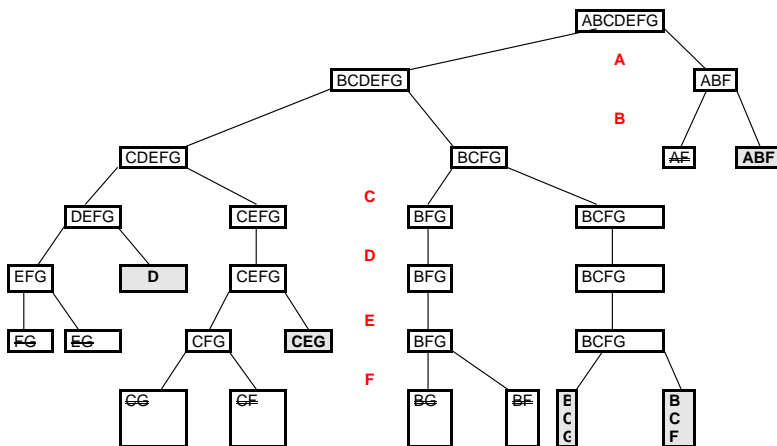
	IN=00	IN=01	IN=11	IN=10
A	D,0	E,1	-, -	A,0
B	-, -	C,1	G,0	-, -
C	-, -	E, -	-, -	A,1
D	-, -	B,1	A, -	F,0
E	G,0	-, -	D,0	-, -
F	-, -	E, -	G,0	A,0
G	G,0	E,1	-, -	A,1

Soluzione

1 - Analisi compatibilità

E	CE					
D	X	CE				
C	BE	BC	X			
B	AE	AG				
A	DG	DG	V	AD		
F	V	CE	V	BE	DG	
G	X	CE	V	AF		
	A	B	C	D	E	F

2 - Calcolo delle classi di MASSIMA COMPATIBILITA' tramite l'albero di compatibilità



Le classi di massima compatibilità sono: ABF, BCG, BCF, CEG, D.

3 - Calcolo delle classi di COMPATIBILITA' PRIMA

Le classi di compatibilità prima sono: ABF, BCG, BCF, CEG, D, AF, CF, B.

4 - Calcolo della copertura partendo dalle classi di compatibilità prima

classi compatibilità prima	di vincoli	passo 1 - COSTI	passo 2 - COSTI	passo 2 - COSTI	
ABF	CE	$3 + 0 - 1 = 2$	$3 + 0 - 0 = 3$		SCELTA PASSO 2
CEG	Ø	$3 + 0 - 1 = 3$			SCELTA PASSO 1
BCF	CE	$3 + 0 - 1 = 2$	$2 + 0 - 0 = 2$	$0 + 0 - 0 = 0$	
BCG	CE	$3 + 0 - 1 = 2$	$1 + 0 - 0 = 1$	$0 + 0 - 0 = 0$	
D	Ø	$1 + 0 - 0 = 1$	$1 + 0 - 0 = 1$	$1 + 0 - 0 = 1$	SCELTA PASSO 3
AF	Ø	$2 + 0 - 0 = 2$	$2 + 0 - 0 = 2$		
CF	Ø	$2 + 0 - 0 = 2$	$1 + 0 - 0 = 1$	$0 + 0 - 0 = 0$	
B	Ø	$1 + 0 - 0 = 1$	$1 + 0 - 0 = 1$		

Una copertura minimale (e anche minima) per la macchina è costituita dalle classi ABF, CEG, D.

Tabella degli stati della macchina ridotta (dove $S_0=ABF$, $S_1=CEG$, $S_2=D$)

	IN=00	IN=01	IN=11	IN=10
S0	S2,0	S1,1	S1,0	S0,0
S1	S1,0	S1,1	S2,0	S0,1
S2	-,-	S0,1	S0,-	S0,0

5 - Calcolo della copertura partendo dalle classi di compatibilità MASSIMA

Le classi di massima compatibilità sono: ABF, BCG, BCF, CEG, D.

Passo 1

Scelgo ABF

Lista L=CE

Elimino da grafo e lista vincoli risolti dalla classe scelta: nessuno

Elimino da grafo nodi della classe scelta che non compaiono ne in vincoli del grafo ne in lista: posso eliminare A B e F

Il grafo risultante è costituito da due classi di massima compatibilità prive di vincoli: CEG e D

Passo 2

Scelgo CEG

Lista L=CE

Elimino da grafo e lista vincoli risolti dalla classe scelta: elimino CE da lista

Lista L = \emptyset

Elimino da grafo nodi della classe scelta che non compaiono ne in vincoli del grafo ne in lista: posso eliminare C E e G

Il grafo risultante è costituito dalla classe di massima compatibilità D

Passo 2

Scelgo D: a questo punto Grafo vuoto e lista vuota. Le classi scelte costituiscono un insieme chiuso.

Esercizio n. 4

Effettuare la sintesi del circuito minimo che realizza la funzionalità di seguito descritta, utilizzando bistabili di tipo SR, fornendone anche la rappresentazione circuitale.

Il circuito ha due ingressi (A e B) e una uscita Z, e ha il seguente comportamento:

- ☐ quando $A = 0$ è irrilevante il valore presente sull'ingresso B e l'uscita è posta a 0
- ☐ quando $A = 1$
 - o se nelle ultime 3 occasioni (anche inframezzate da $A = 0$) in cui $A = 1$ anche $B = 1$, allora l'uscita viene posta a 1
 - o altrimenti $Z = 0$

Esempio:

A	0	1	0	1	1	0	1	0	1	1	0	1	1	0	1	1
B	1	1	0	0	1	1	1	0	1	1	1	0	0	1	1	1
Z	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1
Ok		1		R	1		2		3	3		R	R	1		2 3

Soluzione

Esercizio n. 5

Dati due numeri decimali **A= -12,875** e **B= -7,375** fornire la codifica completa in virgola mobile a singola precisione di A e B.

Effettuare la somma A+B indicando tutti i passaggi relativi sia alla codifica che alla somma.

Soluzione

$-12,875_{10} = 1\ 10000010\ 1001110000000000000000$

$-7,375_{10} = 1\ 10000001\ 1101100000000000000000$

Denormalizzo B e sommo:

$$\begin{aligned} &1.100111000000000000000000 + \\ &0.111011000000000000000000 = \\ &10.100010000000000000000000 \end{aligned}$$

La codifica normalizzata e'
 $-20.250_{10} = 1\ 10000011\ 0100010000000000000000$

Esercizio n. 6

Data la seguente descrizione di circuito in VHDL

```
library ieee;
    use ieee.std_logic_1164.all;
    use ieee.std_logic_unsigned.all;

entity exam is port(
    in1,in2 : in std_logic_vector(3 downto 0);
    ctrl : in std_logic;
    clk: in std_logic;
    out1 : out std_logic_vector(3 downto 0));
end exam;

architecture mixed of exam is
    signal a, b : std_logic_vector(3 downto 0);
    signal c, d, e : std_logic;
begin

    PROC1 : process (reset, clk)
    begin
        if(e='1') then
            x <= "0001";
        elsif (clk = '0' and clk'event) then
            x <= x_next;
        end if;
    end process;

    PROC2: process( in1, in2, ctrl )
    begin
        if ( ctrl='1' ) then
            a <= in1;
            b <= in2;
        elsif (clk = '0' and clk'event) then
            a <= a_next;
            b <= b_next;
        end if;
    end process;

    PROC3: process(a, b )
    begin
        if ( a>b) then
            a_next <= a-b;
        else
            b_next <= b-a;
        end if;
    end process;
    x_next <= x + 1;
    c <= a = "0000";
    d <= b = "0000";
    out1 <= x;

end mixed;
```

Disegnare un circuito composto da componenti elementari (porte logiche, multiplexer, bistabili) che implementi il modello VHDL proposto.