



Politecnico di Milano

Dipartimento di Elettronica e Informazione

prof.ssa Anna Antola

prof.

Fabrizio Ferrandi

Reti Logiche A – Esame del 14 febbraio 2006

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 (5 punti) _____

Esercizio 3 (6 punti) _____

Esercizio 4 (6 punti) _____

Esercizio 5 (7 punti) _____

Esercizio 6 (3 punti) _____

Con Soluzioni

Esercizio n. 1

Eseguire la generazione degli implicanti primi con il metodo di Quine McCluskey per la seguente funzione multiuscita $F(F1;F2;F3)$.

$F1 =$ on-set(m0, m7, m8, m10, m15)
dc-set(m2)

$F2 =$ on-set(m5, m7, m8, m10, m15)
dc-set(m6, m13)

$F3 =$ on-set(m0, m2, m6, m10, m13, m15)
dc-set(m4, m5, m7)

2) Per la funzione **F3** sopra riportata:

- I) Sulla mappa di Karnaugh individuare gli implicanti primi **riportandone la forma algebrica** e separando gli implicanti *primi* da quelli *primi ed essenziali*.
- II) Ricavare tutte le forme minime scegliendo una opportuna copertura della funzione sulla mappa, che minimizzi il numero di implicanti utilizzati ed il numero di letterali. Se è il caso, giustificare formalmente il fatto che la copertura sia unica.
- III) Ricavare il costo della copertura ottenuta, utilizzando come costo il numero di letterali.
- IV) Motivare brevemente il fatto che gli implicanti primi individuati per la funzione $F3$ tramite il metodo di Karnaugh siano di meno di quelli individuati, sempre per la funzione $F3$ tramite il metodo di Quine McCluskey.

Soluzione:

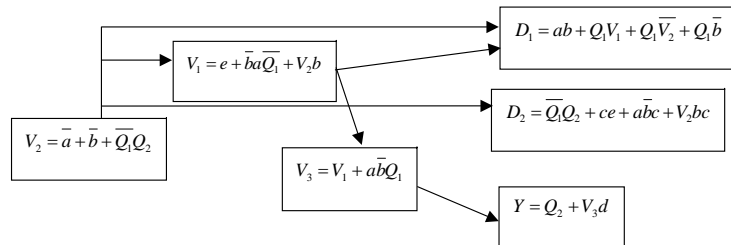
1)
Rilasso il problema trasformando il DC set in ON-Set

```
m0 0000 101 V
--
m2 0010 101 V
m4 0100 001 V
m8 1000 110 V
--
m5 0101 011 V
m6 0110 011 V
m10 1010 111 (A)
--
m7 0111 111 V
m13 1101 011 V
--
m15 1111 111 V
```

```
m0m2 00-0 101 (B)
m0m4 0-00 001 V
m0m8 -000 100 V
--
m2m6 0-10 001 V
m2m10 -010 101 (C)
m4m5 010- 001 V
m4m6 01-0 001 V
m8m10 10-0 110 (D)
--
m5m7 01-1 011 V
m5m13 -101 011 V
m6m7 011- 011 (E)
--
```


Esercizio n. 3

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:



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

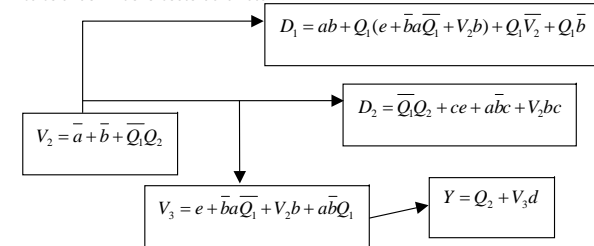
- I) Applicare in sequenza alla rete multilivello le trasformazioni sotto indicate e rispondere alle domande dove richiesto. Disegnare anche il modello della rete finale.
Nota Bene: per ogni trasformazione è **obbligatorio** riportare il **risultato della trasformazione** e **mostrare chiaramente tutti i passaggi** effettuati per ottenere il risultato stesso.
- 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.
 - ELIMINATE(V1,5):** eliminazione vincolata del nodo V1. Il parametro 5 indica la soglia di incremento di area: la trasformazione è accettata se l'incremento di area, in letterali, è minore della soglia indicata.
 - SIMPLIFY(D1):** Minimizzazione a due livelli di D1. Tramite mappe di Karnaugh o manipolazione algebrica ottima, il risultato della minimizzazione è $D_1 = ab + Q_1$
 - SIMPLIFY(V3):** Minimizzazione a due livelli di V3. $V_3 = e + ab + V_2b$
 - FACTOR(D2):** Fattorizzazione del nodo D2. $D_2 = Q_1Q_2 + c(e + ab + V_2b)$
 - COST():** Calcolo del numero di letterali.
 - SUBSTITUTE(V3):** inserisce V3 in tutti i nodi della rete, dove è possibile. La trasformazione ha effetto se produce un guadagno in termini di letterali.
 - COST():** Calcolo del numero di letterali.
- II) Realizzare la rete combinatoria ottenuta al punto h. tramite PAL (si suppone di avere a disposizione tutti i termini prodotto necessari) con piano OR costituito da OR a **due** ingressi. 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:

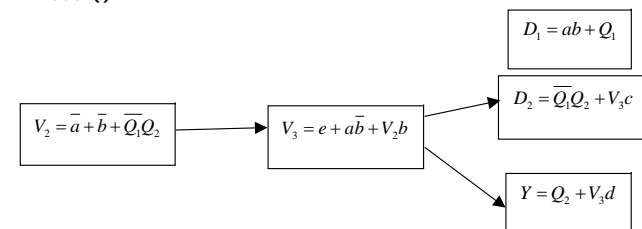
I) La macchina è una macchina di Mealy poiché l'uscita dipende anche dagli ingressi, oltre che dallo stato presente.

- COST():** 35.

- ELIMINATE(V1,5):** Eliminazione vincolata del nodo V1. Il parametro 5 indica la soglia di incremento di area: la trasformazione è accettata se l'incremento di area, in letterali, è minore della soglia indicata. Applicando l'espressione per il calcolo di incremento di area $n^*I - n - I$ (con $I=6$, numero di letterali di V1 e $n=2$, due nodi V_2 e D_1 assorbiti da V1), l'incremento risulta = 4. E' quindi inferiore al valore 5 della soglia di accettazione. La trasformazione viene accettata. Lo stesso risultato si poteva ottenere eliminando il nodo e calcolando il nuovo costo della rete.



- SIMPLIFY(D1):** Minimizzazione a due livelli di D1. Tramite mappe di Karnaugh o manipolazione algebrica ottima, il risultato della minimizzazione è $D_1 = ab + Q_1$
- SIMPLIFY(V3):** Minimizzazione a due livelli di V3. $V_3 = e + ab + V_2b$
- FACTOR(D2):** Fattorizzazione del nodo D2. $D_2 = Q_1Q_2 + c(e + ab + V_2b)$
- COST():** 23
- SUBSTITUTE(V3):** inserisce V3 in tutti i nodi della rete, dove è possibile. La trasformazione ha effetto se produce un guadagno in termini di letterali. L'unico nodo che presenta già V3 come fattore (non è quindi necessario eseguire la divisione algebrica) è D2. L'espressione di D2 diventa:
 $D_2 = Q_1Q_2 + V_3c$
e il costo del nodo (l'unico modificato nella rete) diminuisce di 4 letterali. Quindi la trasformazione ha effetto.
- COST():** 19



II) 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)

$$P_1 = \overline{a}$$
$$P_2 = \overline{b}$$
$$P_3 = \overline{Q_1 Q_2}$$
$$P_4 = V_{21}$$
$$P_5 = e$$
$$P_6 = a\overline{b}$$
$$P_7 = V_2 b$$
$$P_8 = V_{31}$$
$$P_9 = ab$$
$$P_{10} = Q_1$$
$$P_{11} = \overline{Q_1 Q_2}$$
$$P_{12} = V_3 c$$
$$P_{13} = Q_2$$
$$P_{14} = V_3 d$$

$$(*)V_{21} = P_1 + P_2$$
$$(*)V_2 = P_3 + P_4$$
$$(*)V_{31} = P_5 + P_6$$
$$(*)V_3 = P_7 + P_8$$

$$D_1 = P_9 + P_{10}$$
$$D_2 = P_{11} + P_{12}$$
$$Y = P_{13} + P_{14}$$

Esercizio n. 4

Data la seguente tabella degli stati:

- 1. Si esegua l'analisi di compatibilità
- 2. Si determinino le classi di massima compatibilità (si faccia uso dell'algoritmo visto a lezione).
- 3. Si tracci la nuova tabella degli stati della macchina ridotta (mostrare i passaggi per giungere alla soluzione; si faccia uso degli algoritmi visti a lezione).

	IN=0	IN=1
A	C / -	A / 0
B	E / 1	A / 0
C	E / 0	A / 1
D	C / 0	D / -
E	D / 0	A / 1
F	E / 1	B / 0
G	A / 1	- / 1

Soluzione:

Classi di massima compatibilità: ABF – CED – AD – G
Le stesse classi risultano comporre la macchina ridotta, rinominate rispettivamente S0 - S1 - S2 - S3.

	IN=0	IN=1
S0	S1 / 1	S0 / 0
S1	S1 / 0	S2 / 1
S2	S1 / 0	S2 / 0
S3	S0 / 1	- / 1

Esercizio n. 5

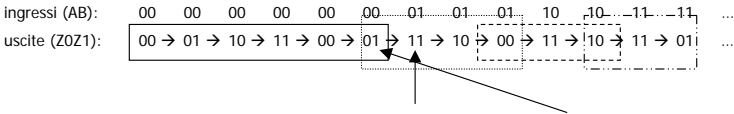
Effettuare la sintesi della macchina sequenziale minima di Moore che realizza la funzionalità di seguito descritta, utilizzando bistabili di tipo SR, fornendone anche la rappresentazione circuitale.

Il dispositivo ha due ingressi A e B e due uscite Z1 e Z2. Per ciascuna delle possibili combinazioni dei valori d'ingresso il dispositivo propone sulle uscite una sequenza di valori. Le **sequenze d'uscita** sono le seguenti:

- ☐ A = 0, B = 0: sequenza 00 → 01 → 10 → 11 → da capo 00 ...
- ☐ A = 0, B = 1: sequenza 00 → 01 → 11 → 10 → da capo 00 ...
- ☐ A = 1, B = 0: sequenza 00 → 11 → 10 → 01 → da capo 00 ...
- ☐ A = 1, B = 1: sequenza 00 → 10 → 11 → 01 → da capo 00 ...

Qualora il valore degli ingressi cambi mentre si è in mezzo ad una sequenza, sulle uscite si procede a mostrare i valori relativi alla *nuova* sequenza, mostrando la configurazione successiva (della nuova sequenza) rispetto a quella mostrata all'istante precedente.

Ad esempio:



Soluzione:

Tabella degli stati minima

Stato pres.	inAB= 00	inAB= 01	inAB= 11	inAB= 10	Z1Z2
A	B	B	D	C	00
B	D	C	A	A	01
C	A	D	B	D	11
D	C	A	C	B	10

Con l'assegnamento A=00, B=01, C=11, D=10 la funzione LAMBDA della macchina di Moore si riduce ad una identità e cioè le uscite Z1Z2 coincidono con lo stato presente dei bistabili.

E' quindi necessario sintetizzare con bistabili SR solo la funzione stato prossimo.

Esercizio n. 6

Data la seguente descrizione VHDL disegnare il circuito logico corrispondente.

```
library IEEE;
use IEEE.std_logic_1164.all;
entity Q is port (
    din, op1, op2, clock : in std_logic;
    dout : out std_logic );
end Q;
architecture arch of Q is
    type T_STATE is ( A, B, C );
    signal state : T_STATE;
    signal int : std_logic;
begin
    P1 : process( clock )
    begin
        if ( clock'event and clock='1' ) then
            case state is
                when A => if ( din='0' ) then
                    state <= B; int <= '1';
                else
                    state <= C; int <= '1';
                end if;
                when B => if ( din='0' ) then
                    state <= A; int <= '0';
                else
                    state <= B; int <= '1';
                end if;
                when C => if ( din='0' ) then
                    state <= B; int <= '1';
                else
                    state <= A; int <= '1';
                end if;
            end case;
        end if;
    end process;
    P2 : process( int, state, op1, op2 )
    begin
        if ( int='0' ) then
            dout <= op1 xor op2;
        else
            if ( state=A or state=B ) then
                dout <= op1 and op2;
            else
                dout <= op1 or op2;
            end if;
        end if;
    end process;
end arch;
```

Rispondere quindi alle seguenti domande:

- 1. Il codice descrive un circuito combinatorio o sequenziale? In quest'ultimo caso, la macchina è di Mealy o di Moore?
- 2. Descrivere sommariamente i due processi P1 e P2 e darne una rappresentazione grafica (ad es., schema circuitale per circuiti combinatori o diagramma degli stati per macchine a stati finiti)