

Reti logiche - Prova scritta del 11 Settembre 2018

Cognome e Nome: _____ Matricola _____

Prima della consegna barrare una delle due caselle sottostanti. L'opzione scelta non può essere modificata dopo la consegna.

Chiedo che la mia prova scritta sia corretta e valutata subito, perché intendo sostenere la prova orale in questo appello. Prendo atto che, a seguito della mia decisione, la mia prova scritta cesserà di essere valida al termine di questo appello e non potrà essere usata per l'appello straordinario di Novembre.

☐

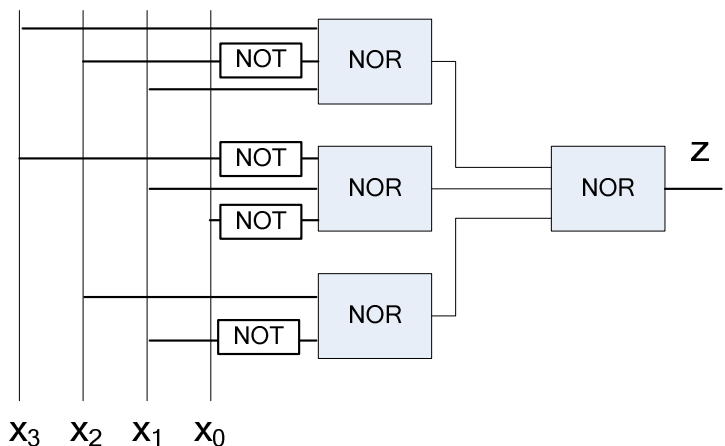
Chiedo che la mia prova scritta sia corretta e valutata dopo la fine dell'appello in corso, perché ho diritto a ed intenzione di rimandare la prova orale all'appello straordinario di Novembre. Prendo atto che il mio diritto a rimandare la prova orale sarà oggetto di verifica, e che dovrò ripetere l'intero esame da capo se la verifica darà esiti negativi (per qualunque motivo).

☐

Esercizio 1

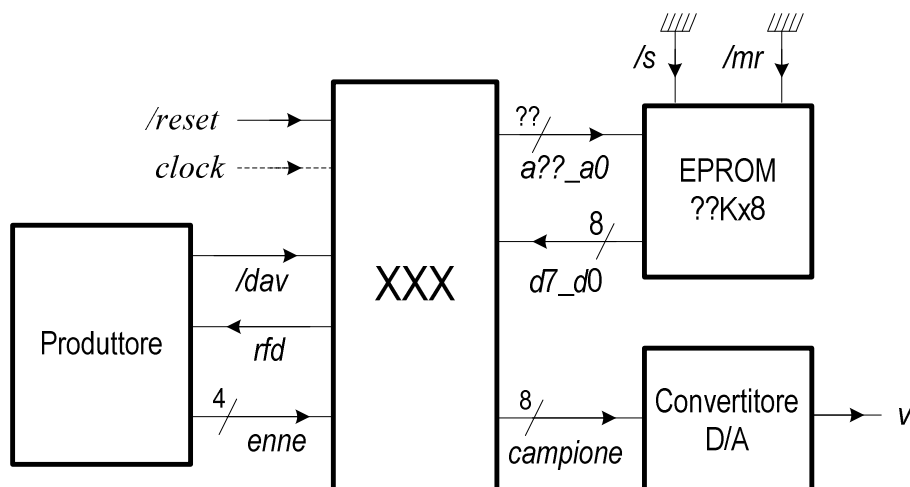
Data la rete combinatoria di figura:

- 1) disegnare la mappa di Karnaugh per la legge z , sapendo che non è possibile che si presentino stati di ingresso in cui tutte le variabili hanno lo stesso valore.
- 2) Individuare e classificare gli implicant principali, e trovare *tutte* le liste di copertura irridondanti. Sintetizzare la rete in forma SP, scegliendo la realizzazione di costo minimo secondo il criterio a porte.
- 3) Individuare, classificare ed eliminare eventuali alee sulla realizzazione di cui al punto 2.



Esercizio 2

La EPROM in figura contiene 16 segnali digitali, ognuno avente 1024 campioni da 8 bit. I campioni del segnale identificato dal numero n ($n = 0, 1, \dots, 15$) sono memorizzati nella EPROM a partire dall'indirizzo $n * 1024$. L'unità XXX riceve dal Produttore il numero identificativo di un segnale ed invia i 1024 campioni del segnale a un convertitore D/A, mantenendo ciascun campione all'ingresso del convertitore per due cicli di clock. Nei tempi morti (cioè tra un segnale ed un altro) invia verso il convertitore il byte 'H00.



Descrivere e sintetizzare l'unità XXX, evidenziando lo schema circuitale relativo al registro che sostiene gli indirizzi verso la EPROM. Nel descrivere l'unità si supponga che la EPROM sia veloce a rispondere, cosicché *non siano necessari stati di wait*.

Soluzione Es 1

1) Dalla mappa si ricava immediatamente: $z = \overline{(x_3 + x_2 + x_1)} + \overline{(x_3 + x_1 + x_0)} + \overline{(x_2 + x_1)}$, da cui:

$\bar{z} = (\bar{x}_3 \cdot \bar{x}_2 \cdot \bar{x}_1) + (\bar{x}_3 \cdot \bar{x}_1 \cdot \bar{x}_0) + (\bar{x}_2 \cdot \bar{x}_1)$. Da quest'ultima, si ricava la mappa di Karnaugh per \bar{z} , e, tenendo conto di quanto scritto al punto 1), anche quella di z , riportate di seguito.

$x_3x_2 \backslash x_1x_0$	00	01	\bar{z} 11	10
00	-	0	1	1
01	1	1	0	0
11	0	1	-	0
10	0	1	1	1

$x_3x_2 \backslash x_1x_0$	00	01	z 11	10
00	-	1	0	0
01	0	0	1	1
11	1	0	-	1
10	1	0	0	0

2) La sintesi in forma SP può essere fatta come segue:

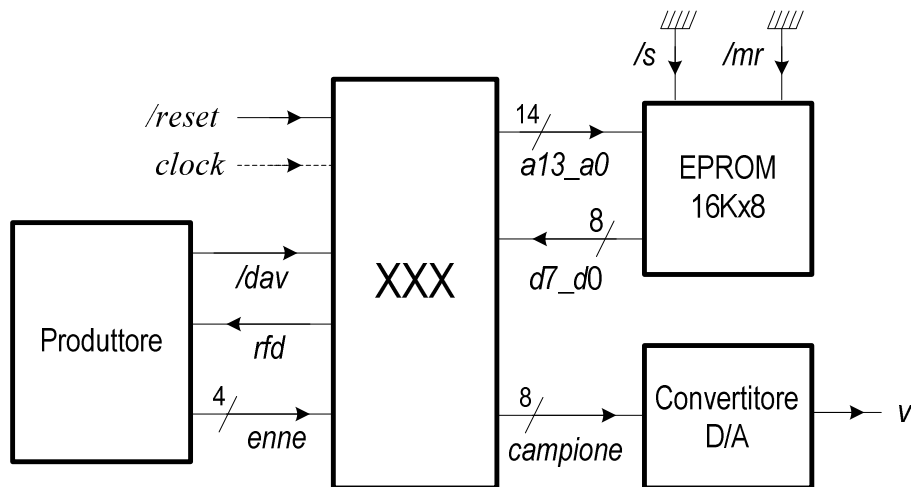
$x_3x_2 \backslash x_1x_0$	00	01	z 11	10
00	-	1	0	0
01	0	0	1	1
11	1	0	-	1
10	1	0	0	0

- Implicanti essenziali: $A = x_2 \cdot x_1$, $B = \bar{x}_3 \cdot \bar{x}_2 \cdot \bar{x}_1$
- Implicanti assolutamente eliminabili: *nessuno*
- Implicanti semplicemente eliminabili: $C = x_3 \cdot \bar{x}_1 \cdot \bar{x}_0$, $D = \bar{x}_2 \cdot \bar{x}_1 \cdot \bar{x}_0$, $E = x_3 \cdot x_2 \cdot \bar{x}_0$
- Liste di copertura irridondanti: $\{A, B, C\}$, $\{A, B, D, E\}$
- Lista di copertura di costo minimo: $\{A, B, C\}$

Quindi, la sintesi di costo minimo in forma SP è: $z = (x_2 \cdot x_1) + (\bar{x}_3 \cdot \bar{x}_2 \cdot \bar{x}_1) + (x_3 \cdot \bar{x}_1 \cdot \bar{x}_0)$

3) La sintesi in accordo alla lista di copertura di costo minimo presenta delle alee statiche del 1° ordine sul livello 1, in corrispondenza delle transizioni 1110-1100 e 1000-0000. Per eliminare le suddette alee, è necessario introdurre gli implicanti che mancano (D e E).

Esercizio 2 - Una possibile soluzione



```

module XXX(enne,dav_,rfd, a13_a0, d7_d0, campione, clock ,reset_);
input      clock, reset_;
input[3:0]  enne;
input      dav_;
output     rfd;
output[13:0] a13_a0;
input[7:0]  d7_d0;
output[7:0] campione;

reg        RFD;          assign rfd=RFD;
reg[13:0]  A13_A0;       assign a13_a0=A13_A0;
reg[7:0]   CAMPIONE;     assign campione=CAMPIONE;
reg[10:0]  COUNT;

reg[1:0]   STAR;         parameter [1:0] S0=0,S1=1,S2=2,S3=3;
parameter num_campioni=1024;

always @(reset_==0) begin RFD<=1; COUNT<=num_campioni; STAR<=S0; end
always @(posedge clock) if (reset_==1) #3

casex(STAR)
// Attesa che venga inviato il numero d'ordine di un segnale
S0: begin RFD<=1; CAMPIONE<='H00; A13_A0<={enne*num_campioni};
        STAR<=(dav_==1)?S0:S1; end
        //la moltiplicazione va fatta concatenando degli 0

// Chiusura dell'handshake
S1: begin RFD<=0; STAR<=(dav_==0)?S1:S2; end

// Emissione del segnale digitale
S2: begin CAMPIONE<=d7_d0; STAR<=S3; end
S3: begin COUNT <=(COUNT==1)?num_campioni:(COUNT-1); A13_A0<=A13_A0+1;
        STAR<=(COUNT==1)?S0:S2; end
endcase
endmodule

```