Reti logiche - Prova scritta dell'8 Gennaio 2019

Cognome e Nome: _____ Matricola _____

Esercizio 1

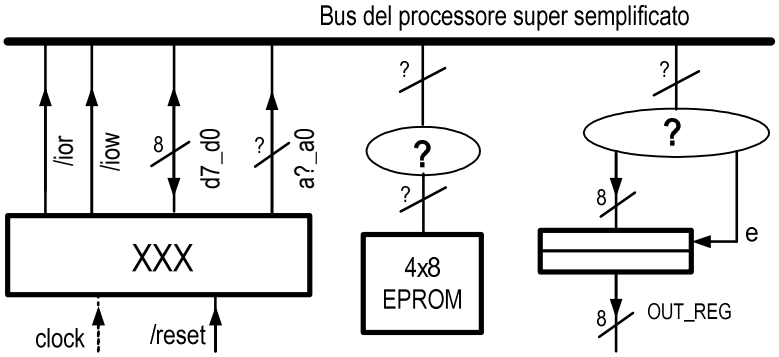
Sintetizzare, fino al livello delle porte logiche elementari:

1) una rete combinatoria che *raddoppia numeri naturali in base 6 su n cifre*. La rete ha in ingresso un numero in base 6 ad *n* cifre, e come uscite un numero in base 6 ad *n* cifre ed un *carry*, che vale 1 se il risultato dell'operazione non è rappresentabile su *n* cifre.
2) Una rete combinatoria che prende in ingresso un numero naturale in base 6 su *n* cifre e lo moltiplica per 72, producendo un'uscita su *n*+2 cifre ed un carry che vale 1 se il risultato dell'operazione non è rappresentabile su *n*+2 cifre.

Attenzione: verrà considerata nulla una sintesi che non si spinga fino al livello delle porte logiche elementari.

Esercizio 2

Descrivere l'unità *XXX* in modo che, ciclicamente, emetta il contenuto delle locazioni della EPROM tramite il registro OUT_REG, mantenendovi il contenuto di ogni locazione per 9 cicli di clock. Si faccia l'assunzione che il tempo di risposta della EPROM sia tale da non richiedere stati di wait. Chiamare COUNT il registro utilizzato per effettuare il conteggio e **evidenziare**, anche **circuitalmente,** la porzione della parte operativa relativa a tale registro.

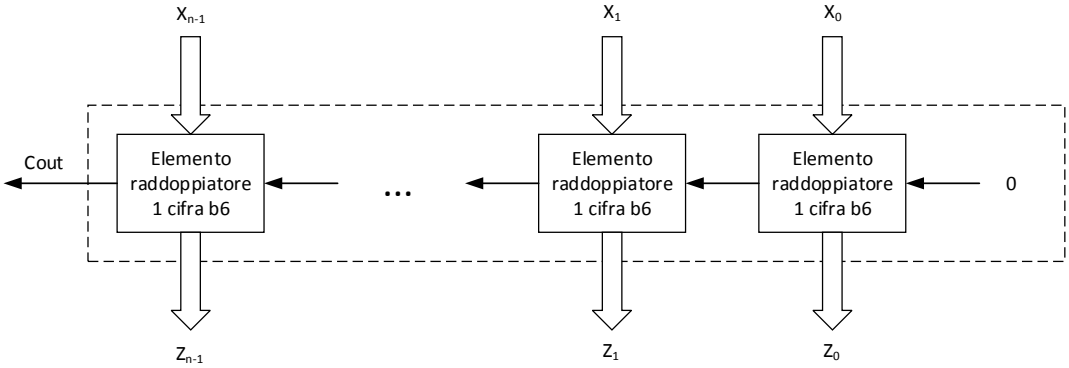


Reti logiche - Prova scritta dell'8 Gennaio 2019

Cognome e Nome: _____ Matricola _____

Soluzione Esercizio 1

1) La rete ad *n* cifre può essere scomposta in *n* elementi raddoppiatori in montaggio ripple carry, come in figura.



Ciascuno degli elementi ha in ingresso un carry ed una cifra in base 6, codificata su tre variabili binarie, ed in uscita una cifra in base 6 ed un carry. Un elemento è descritto dalla seguente tabella di verità:

Cin	x2	x1	x0	z2	z1	z0	Cout
0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0
0	0	1	0	1	0	0	0
0	0	1	1	0	0	0	1
0	1	0	0	0	1	0	1
0	1	0	1	1	0	0	1
0	1	1	0	-	-	-	-
0	1	1	1	-	-	-	-
1	0	0	0	0	0	1	0
1	0	0	1	0	1	1	0
1	0	1	0	1	0	1	0
1	0	1	1	0	0	1	1
1	1	0	0	0	1	1	1
1	1	0	1	1	0	1	1
1	1	1	0	-	-	-	-
1	1	1	1	-	-	-	-

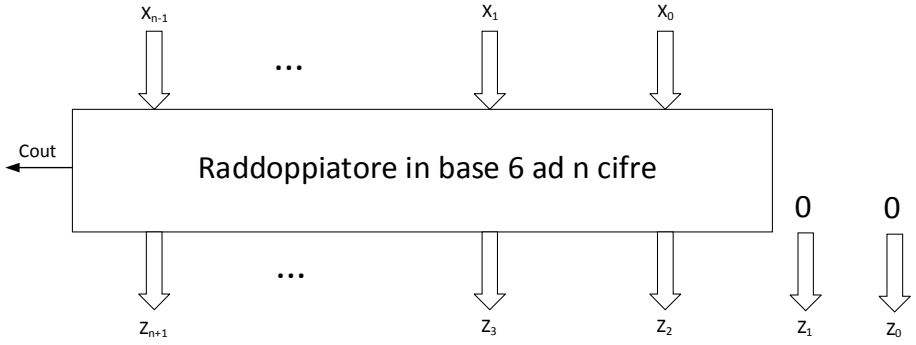
x_0C_{in} \ x_2x_1	00	01	11	10
00	0000	1000	-	0101
01	0010	1010	-	0111
11	0110	0011	-	1011
10	0100	0001	-	1001

($z_2z_1z_0C_{out}$)

Una sintesi in forma SP per l'elemento raddoppiatore è:

$z_2 = x_1 \cdot \overline{x_0} + x_2 \cdot x_0$, $z_1 = \overline{x_2} \cdot \overline{x_1} \cdot x_0 + x_2 \cdot \overline{x_0}$, $z_0 = C_{in}$, $C_{out} = x_2 + x_1 \cdot x_0$

2) Visto che 72=2*6^2, e la moltiplicazione per una potenza della base si fa aggiungendo zeri in coda, la rete richiesta è la seguente:



Reti logiche - Prova scritta dell'8 Gennaio 2019

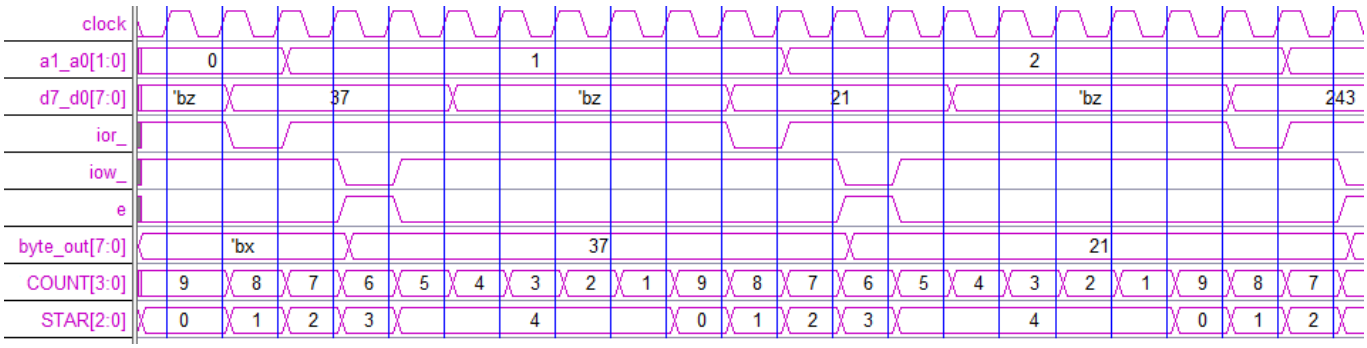
Cognome e Nome: _____ Matricola _____

Esercizio 2 – una soluzione

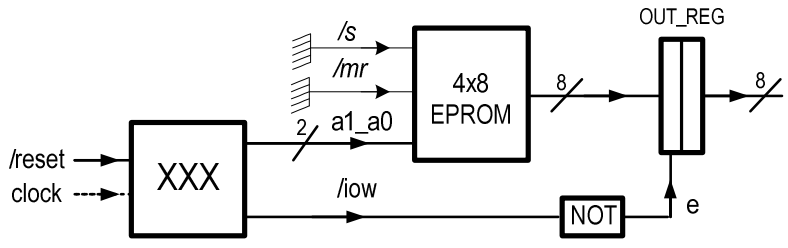
La EPROM necessita di 2 fili di indirizzo e può essere sempre selezionata e attivarsi quando */ior* va a 0. La variabile *e* del registro può coincidere con */iow* negato:

```
module XXX(d7_d0,a1_a0, ior_,iow_,clock,reset_);
 input         clock,reset_;
 output        ior_,iow_;
 output [1:0]  a1_a0;
 inout  [7:0]  d7_d0;
 reg       IOR_;  assign ior_=IOR_;
 reg       IOW_;  assign iow_=IOW_;
 reg [1:0] A1_A0; assign   a1_a0=A1_A0;
 reg       DIR;
 reg [7:0] D7_D0; assign   d7_d0=(DIR==1)?D7_D0:'HZZ;
 reg [3:0]     COUNT;
 reg [2:0]     STAR; parameter [2:0] S0=0,S1=1,S2=2,S3=3,S4=4;
 parameter Num_Cicli=9;
 always @(reset_==0)#1 begin DIR<=0; IOR_<=1; IOW_<=1; A1_A0<=0;
                             COUNT<=Num_Cicli; STAR<=S0; end
 always @(posedge clock) if (reset_==1) #3
  casex(STAR)
   S0:    begin COUNT<=COUNT-1; IOR_<=0; STAR<=S1; end
   S1:    begin COUNT<=COUNT-1; D7_D0<=d7_d0; IOR_<=1; A1_A0<=A1_A0+1;
                DIR<=1; STAR<=S2; end
   S2:    begin COUNT<=COUNT-1; IOW_<=0; STAR<=S3; end
   S3:    begin COUNT<=COUNT-1; IOW_<=1; STAR<=S4; end
   S4:    begin COUNT<=(COUNT==1)?Num_Cicli:(COUNT-1);
                DIR<=0; STAR<=(COUNT==1)?S0:S4; end
  endcase
endmodule
```

Simulazione nell'ipotesi che la EPROM contenga 37,21,243,255



Soluzione più semplice: *XXX* non ha le variabili */ior* e *d7_d0*; la EPROM è sempre selezionata e attiva in lettura e invia direttamente i dati al registro OUT_REG. L'unità *XXX* si limita a cambiare gli indirizzi alla EPROM, a inviare il commando di scrittura per OUT_REG e ad effettuare le opportune operazioni di conteggio.



Reti logiche - Prova scritta dell'8 Gennaio 2019

Cognome e Nome: _____ Matricola _____

```
module XXX(a1_a0, iow_,clock,reset_);
 input         clock,reset_;
 output        iow_;
 output [1:0]  a1_a0;
 reg       IOW_;  assign iow_=IOW_;
 reg [1:0] A1_A0; assign   a1_a0=A1_A0;
 reg [3:0]     COUNT;
 reg [2:0]     STAR; parameter [2:0] S0=0,S1=1,S2=2,S3=3,S4=4;
 parameter Num_Cicli=9;
 always @(reset_==0)#1 begin IOW_<=1; A1_A0<=0; COUNT<=Num_Cicli; STAR<=S0; end
 always @(posedge clock) if (reset_==1) #3
  casex(STAR)
   S0:    begin COUNT<=COUNT-1; STAR<=S1; end
   S1:    begin COUNT<=COUNT-1; IOW_<=0; STAR<=S2; end
   S2:    begin COUNT<=COUNT-1; IOW_<=1; STAR<=S3; end
   S3:    begin COUNT<=COUNT-1; A1_A0<=A1_A0+1; STAR<=S4; end
   S4:     begin  COUNT<=(COUNT==1)?Num_Cicli:(COUNT-1);  STAR<=(COUNT==1)?S0:S4;
end
  endcase
endmodule
```