Esercizio 1

1) Descrivere una rete sequenziale sincronizzata di Moore ad <u>1</u> ingresso che riconosce la sequenza **0,0,1,0,1,1,0**. Si presti particolare attenzione a non perdere nessuna sequenza.
2) Sintetizzare la rete descritta al punto precedente. La sintesi delle reti CN1 e CN2 deve essere a costo minimo in forma SP.
3) Rispondere alla seguente domanda: è possibile sintetizzare una rete sequenziale *asincrona* che riconosce la sequenza sopra specificata? Se sì, come? Se no, perché?

Soluzione

1) La rete può essere descritta come in figura (gli stati in neretto nella tabella corrispondono all'evoluzione degli stati conseguente al riconoscimento di una sequenza):

x	0	1	z
S0	**S1**	S0	0
S1	**S2**	S0	0
S2	S2	**S3**	0
S3	**S4**	S0	0
S4	S2	**S5**	0
S5	S1	**S6**	0
S6	**S7**	S0	0
S7	S1	S0	1

2) Per quanto riguarda la sintesi, si può osservare quanto segue

- adottando la codifica degli stati $S_i = (i)_{b2}$, la rete CN2 è $z = y_2 \cdot y_1 \cdot y_0$, a costo minimo.
- la rete CN1 ha 4 ingressi (3 variabili di stato, 1 variabile di ingresso). La mappa di Karnaugh di CN1 è quella riportata sotto:

a₂a₁a₀ → $a_2a_1a_0$

y_1y_0 \ y_2x	00	01	11	10
00	001	000	101	010
01	010	000	110	001
11	100	000	000	001
10	010	011	000	111

Per la sintesi di tutte e tre le variabili di uscita tutti gli implicanti sono essenziali. Si ottiene quanto segue:

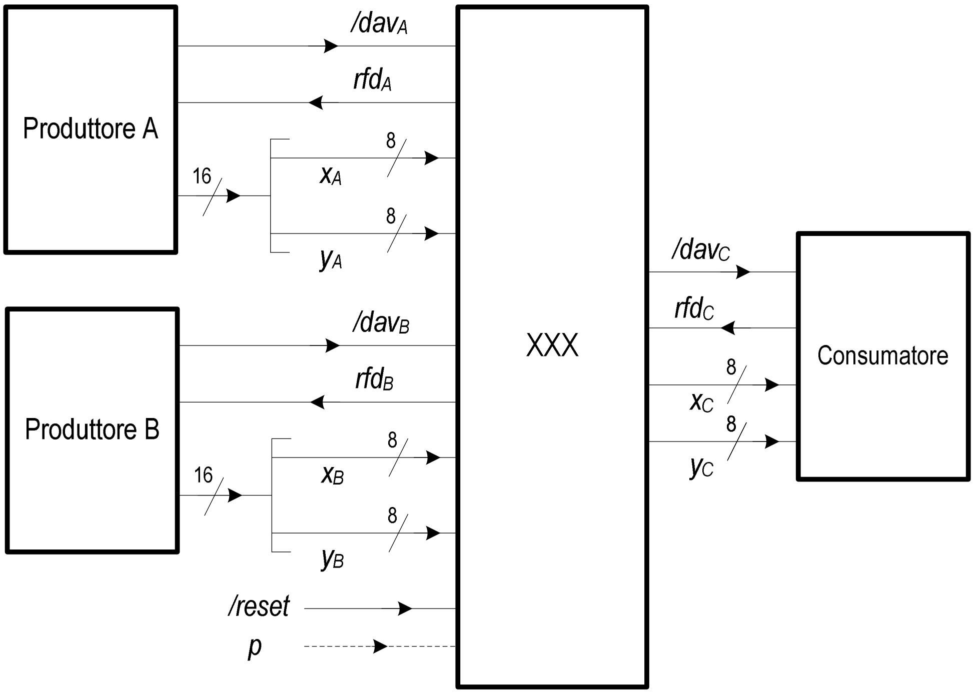
$$a_2 = y_2 \cdot \overline{y_1} \cdot x + \overline{y_2} \cdot y_1 \cdot y_0 \cdot \overline{x} + y_2 \cdot y_1 \cdot \overline{y_0} \cdot \overline{x}$$

$$a_1 = \overline{y_2} \cdot y_1 \cdot \overline{y_0} + y_2 \cdot \overline{y_0} \cdot \overline{x} + y_2 \cdot \overline{y_1} \cdot y_0 \cdot x + \overline{y_2} \cdot \overline{y_1} \cdot y_0 \cdot \overline{x}$$

$$a_0 = \overline{y_2} \cdot \overline{y_1} \cdot \overline{y_0} \cdot \overline{x} + y_2 \cdot \overline{y_1} \cdot \overline{y_0} \cdot x + \overline{y_2} \cdot y_1 \cdot \overline{y_0} \cdot x + y_2 \cdot y_1 \cdot \overline{x} + y_2 \cdot y_0 \cdot \overline{x}$$

3) Non è possibile. Una rete asincrona evolve al cambiare degli ingressi. Non è possibile pertanto riconoscere il permanere di un medesimo stato di ingresso.

Descrivere e **sintetizzare** l'Unità XXX in modo che compia ciclicamente le seguenti azioni: i) prelevare dai Produttori A e B quattro numeri naturali a 8 bit che rappresentano le coordinate (x_A, y_A) di un punto A e le coordinate (x_B, y_B) di un punto B nel quadrante positivo di uno spazio a due dimensioni; ii) inviare al Consumatore le coordinate del punto (A o B) più vicino all'origine.



NOTA: Non si faccia alcuna ipostesi sui tempi di risposta dei Produttori e del Consumatore

Una soluzione

```
module XXX(rfdA,davA_,xA,yA, rfdB,davB_,xB,yB, davC_,rfdC,xC,yC, p,reset_);
 input       p,reset_;
 input       davA_,davB_,rfdC;
 output      rfdA, rfdB,davC_;
 input [7:0] xA,yA,xB,yB;
 output[7:0] xC,yC;
 reg       RFDAeB,DAVC_;   assign rfdA= RFDAeB, rfdB= RFDAeB, davC_=DAVC_;
 reg [7:0] XC,YC;          assign xC=XC, yC=YC;
 reg F;   //memorizza il risultato del confronto
 reg [2:0] STAR;   parameter S0=0,S1=1,S2=2,S3=3,S4=4;
 function [16:0] distanza_al_quadrato;
  input [7:0]  x,y;
  distanza_al_quadrato=(x*x)+(y*y);
 endfunction
 always @(posedge p or negedge reset_)
  if (reset_==0) begin RFDAeB=1; DAVC_=0; STAR=S0; end else #3
  casex(STAR)
   S0: begin F<=(distanza_al_quadrato(xA,yA)<distanza_al_quadrato(xB,yB))?1:0;
       STAR<=((davA_ | davB_)==1)?S0:S1; end
   S1: begin RFDAeB<=0; XC<=(F==1)?xA:xB; YC<=(F==1)?yA:yB; STAR<=S2; end
   S2: begin STAR<=((davA_ & davB_)==1)?S3:S2; end
   S3: begin RFDAeB<=1; DAVC_<=0; STAR<=(rfdC==1)?S3:S4; end
   S4: begin DAVC_<=1; STAR<=(rfdC==0)?S4:S0; end
  endcase
endmodule
```