

Esercizio 1

Si consideri un piano cartesiano su cui insistono tre punti A, B, C, a coordinate intere rappresentate su n bit in complemento alla radice. Sintetizzare la rete combinatoria che ha:

- In ingresso, le rappresentazioni delle coordinate dei tre punti
- In uscita, due variabili eq e iso , che sono a 1 se il triangolo ABC è, rispettivamente, equilatero o isoscele, e zero altrimenti.

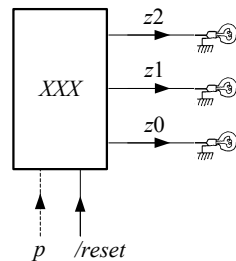
Descrivere esplicitamente qualunque rete non decritta a lezione.

Si consiglia di scomporre la rete in sottoreti più semplici per semplificare il disegno.

Esercizio 2

Con riferimento alla Fig. 1, descrivere, disegnare il diagramma di temporizzazione e sintetizzare l'Unità XXX che al reset iniziale accende le lampade, le tiene accese fino allo scadere del primo ciclo di clock e poi compie ciclicamente i seguenti passi:

- 1) spegne le lampade e le tiene spente per **21 (4 nella simulazione)** cicli di clock
- 2) accende *la lampada* gestita da $z2$ e poi le altre, intervallando le accensioni di **7 (2 nella simulazione)** cicli di clock
- 3) tiene tutte le lampade accese per **12 (3 nella simulazione)** cicli di clock e torna al passo 1).

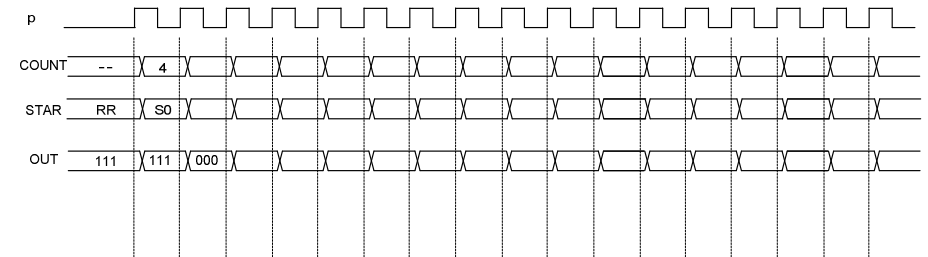
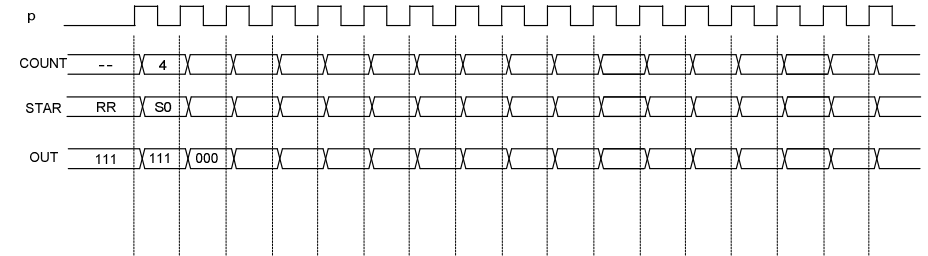
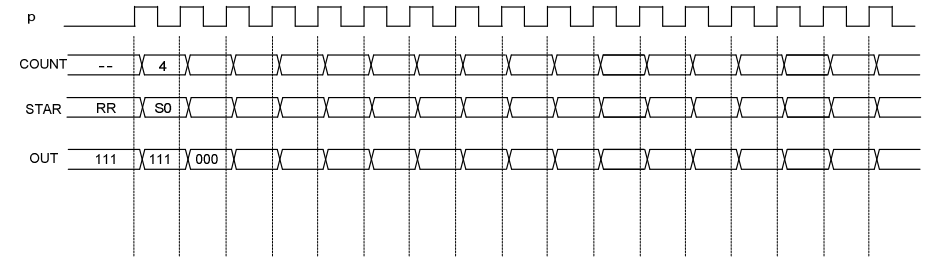
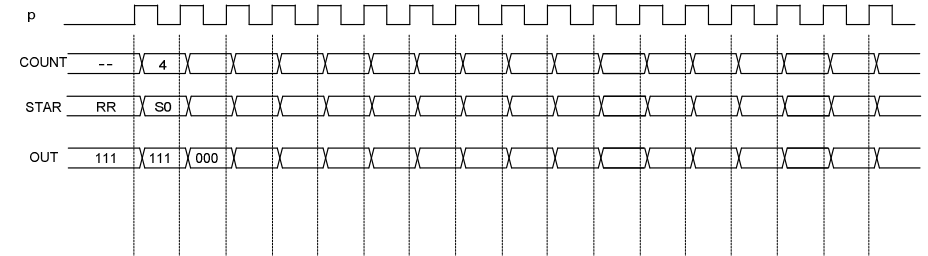


(nella simulazione)



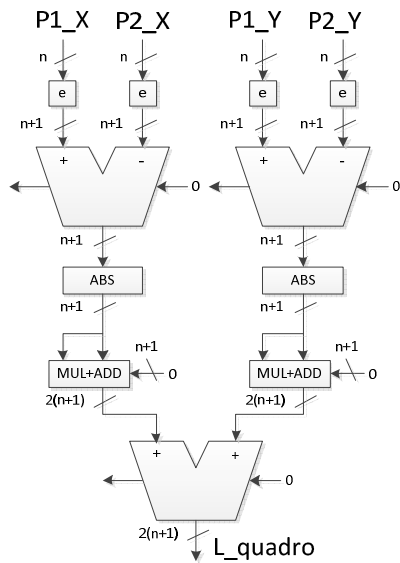
Impostare la descrizione come segue:

```
module XXX(z2_z0,p,reset_);
input  p,reset_;
output [2:0] z2_z0;
reg    [2:0] OUT; assign z2_z0=OUT;
reg    [4:0] COUNT;
reg    [...:0] STAR; parameter S0=0, S1=1, ..., RR=...;
parameter T_spente=..., I_accensione=..., T_accese=...;
always @(posedge p or negedge reset_)
    if (reset_==0) begin OUT='B111; STAR=RR; end else #3
        casex(STAR)
            RR: begin COUNT=T_spente; STAR<= S0; end
            S0: begin OUT<='B000; ...
            ...
        endcase
endmodule
```

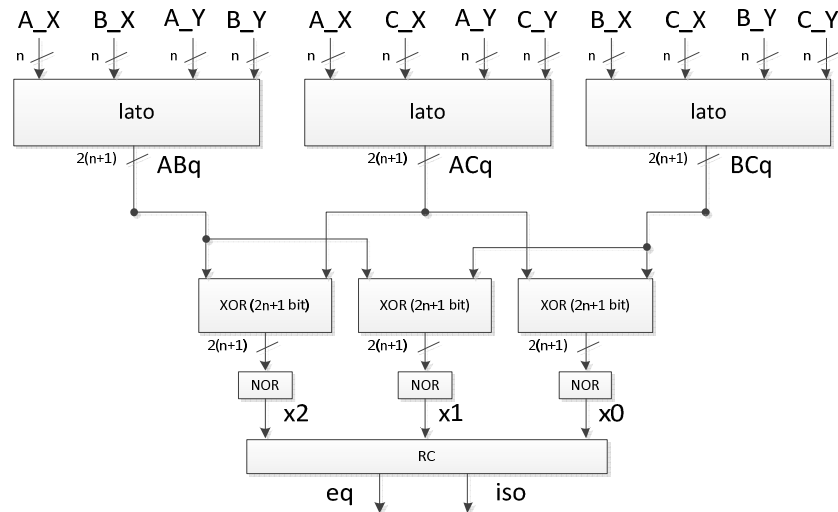


Esercizio 1 - Soluzione

La rete “lato” che calcola il quadrato di un lato dati i due punti sul piano cartesiano è la seguente:



Con questa rete, la soluzione è la seguente:



In cui la tabella di verità di RC è la seguente:

x2	x1	x0	eq	iso
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	-	-
1	0	0	0	1
1	0	1	-	-
1	1	0	-	-
1	1	1	1	0

Esercizio 2 - Una Possibile Soluzione (non troppo ottimizzata)

```

module XXX(z2_z0,p,reset_);
input  p,reset_;
output [2:0] z2_z0;

reg    [2:0] OUT;  assign z2_z0=OUT;
reg    [4:0] COUNT;

reg    [2:0] STAR; parameter S0=0, S1=1, S2=2, S3=3, RR=4;

parameter T_spente=4, I_accensione=2, T_accese=3;

always @(posedge p or negedge reset_)
    if (reset_==0) begin OUT='B111; STAR=RR; end else #3
        casex(STAR)

            RR: begin COUNT=T_spente; STAR<= S0; end

            S0: begin OUT<='B000; COUNT<=(COUNT==1)?I_accensione:(COUNT-1);
                    STAR<=(COUNT==1)?S1:S0; end

            S1: begin OUT<='B100; COUNT<=(COUNT==1)? I_accensione:(COUNT-1);
                    STAR<=(COUNT==1)?S2:S1; end

            S2: begin OUT<='B110; COUNT<=(COUNT==1)? T_accese:(COUNT-1);
                    STAR<=(COUNT==1)?S3:S2; end

            S3: begin OUT<='B111; COUNT<=(COUNT==1)? T_spente:(COUNT-1);
                    STAR<=(COUNT==1)?S0:S3; end

        endcase
endmodule

```

