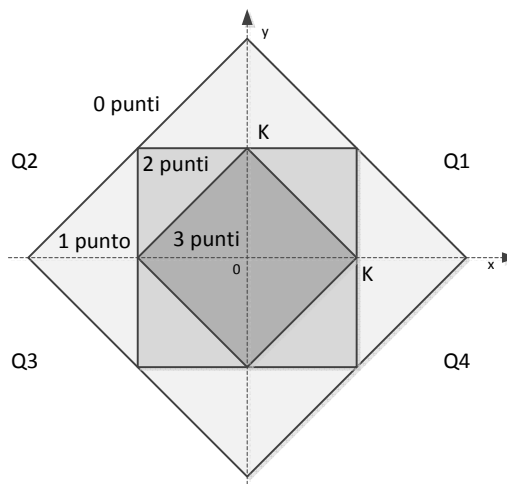


### Esercizio 1

Sia dato un piano cartesiano a coordinate intere, rappresentate in complemento alla radice su  $n$  bit. Sintetizzare una rete combinatoria (ovviamente composta da più sottoreti combinatorie) che prende in ingresso le coordinate di un punto, e produce in uscita il punteggio relativo alla zona del bersaglio colpita, come da figura. Si considerino le coordinate sui bordi come appartenenti ad una qualunque delle zone. Descrivere esplicitamente qualunque rete non descritta a lezione.

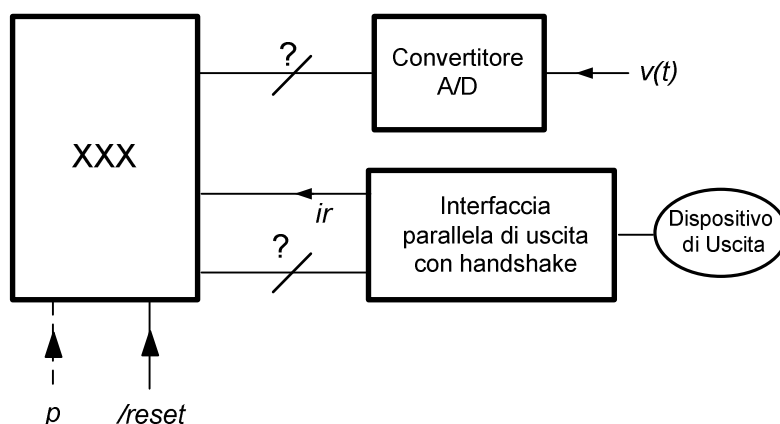
Per i quadrati inclinati, può essere utile ragionare separatamente per i quattro quadranti Q1-Q4, e successivamente cercare un'espressione unica che vada bene per tutti e quattro.



### Esercizio 2

L'Unità XXX della figura sottostante gestisce sia una *Interfaccia di uscita con handshake* sia un *Convertitore A/D*. L'Interfaccia, quando ha emesso un byte verso il dispositivo di uscita, invia una richiesta di interruzione mettendo a 1 il valore della variabile *ir*. L'Unità XXX si evolve come segue:

- 1) Preleva un byte dal Convertitore A/D e quando arriva una richiesta di interruzione dall'Interfaccia, lo emette tramite essa;
- 2) Torna al punto 1.



- 1) **Semplificare preventivamente** l'Interfaccia di uscita lasciando le sole circuiterie che servono affinché essa sia gestibile esclusivamente ad interruzione di programma, sia sempre selezionata e sia sempre abilitata ad inviare richieste di interruzione.
- 2) **Completare i collegamenti** fra Unità XXX, Interfaccia e Convertitore (**Si noti Convertitore e non Interfaccia per la Conversione**).
- 3) **Descrivere** l'Unità XXX (si supponga che al reset iniziale l'Interfaccia invii immediatamente la prima richiesta di interruzione), **non dimenticando di gestire TUTTI gli handshake**.

Ricordare che il linguaggio Verilog è case sensitive e che le parole chiave vanno scritte con lettere minuscole. Si consiglia inoltre, per leggibilità, di scrivere con lettere minuscole i nomi delle variabili e con lettere maiuscole i nomi dei registri.

## Soluzione Esercizio 1

Dette A, B, C le zone riquadrate in ordine di punteggio decrescente e P un punto sul piano, si possono definire tre variabili  $in3$ ,  $in2$ ,  $in1$  che indichino, in funzione delle coordinate del punto, in quale zona si collochi il punto:

```
wire in3=((ABS(x)+ABS(y))<K)?1:0      ;    //in3 a 1 indica punto nella zona A
wire in2=((ABS(x)<K)&(ABS(y)<K))?1:0 ;    //in2 a 1 indica punto nella zona B
wire in1=((ABS(x)+ABS(y))<2*K)?1:0    ;    //in1 a 1 indica punto nella zona C
```

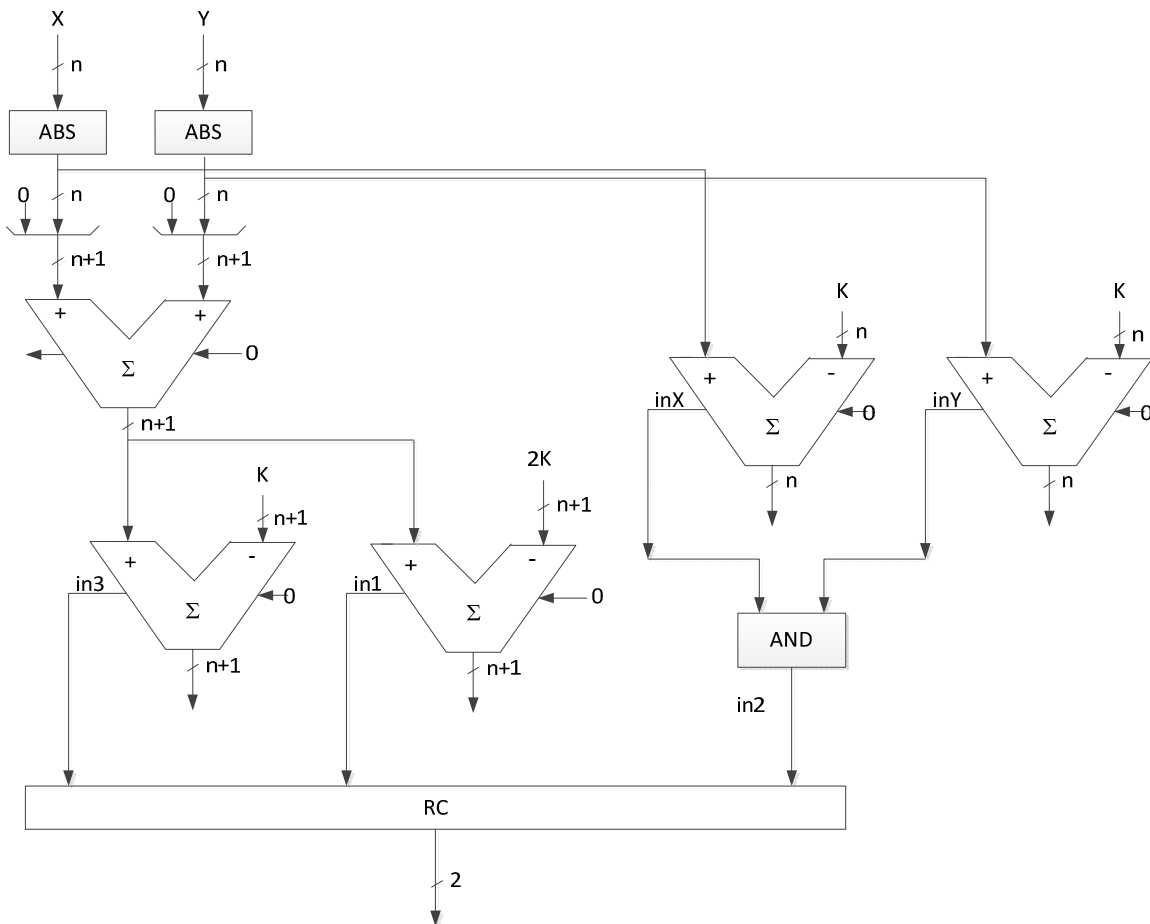
L'espressione per  $in2$  è banale. L'espressione per  $in3$  (quella per  $in1$  è del tutto simile come struttura) può essere ricavata in base alle seguenti considerazioni che, per semplicità, vengono applicate ragionando sul quadrante **Q2**

L'equazione della retta che delimita la zona C è  $y = x + K$ . Nel quadrante **Q2** la coordinata  $y$  è positiva mentre la coordinata  $x$  è negativa.

Nella zona C del quadrante **Q2** si ha pertanto che le coordinate dei punti sulla retta soddisfano la relazione  $ABS(y) = -ABS(x) + K$  cioè  $ABS(x) + ABS(y) = K$ , mentre le coordinate dei punti interni soddisfano la relazione  $ABS(x) + ABS(y) < K$ . Data la simmetria, tale relazione tra coordinate vale per tutti i punti dell'area C.

Nelle espressioni Verilog di sopra, i punti sul bordo sono pertanto considerati parte delle zone più esterne.

La rete composta che assegna il punteggio è descritta nella figura sottostante.

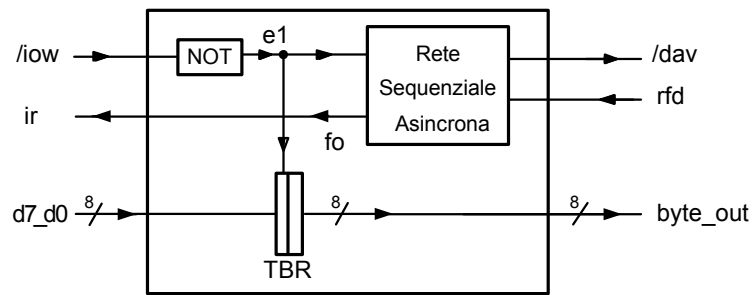


La tabella di verità di RC finale è la seguente:

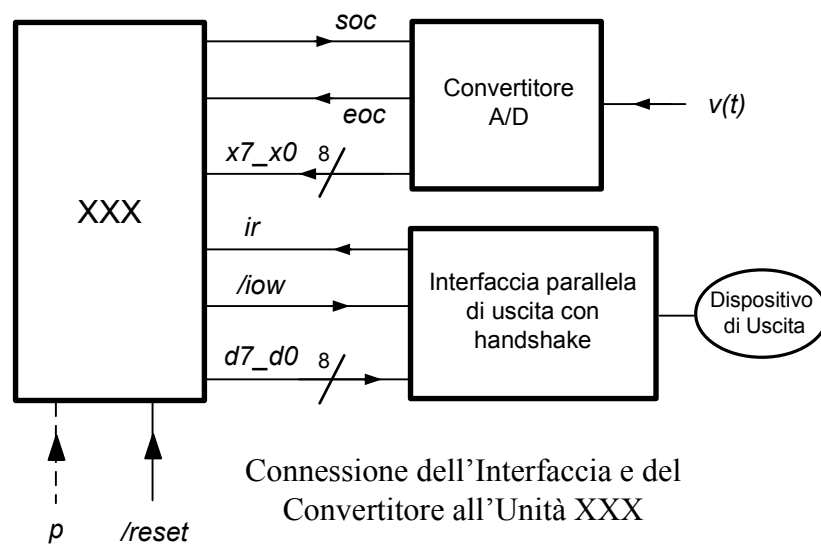
$in3$	$in2$	$in1$	$z1$	$z0$
1	1	1	1	1
0	1	1	1	0
0	0	1	0	1
0	0	0	0	0
others			-	-

da cui  $z_1 = in_2$ ,  $z_0 = in_3 + in_1 \cdot \overline{in_2}$

## Soluzione esercizio 2



Interfaccia di uscita semplificata



Connessione dell'Interfaccia e del Convertitore all'Unità XXX

```

module XXX(soc,eoc, x7_x0, ir,iow_,d7_d0, p,reset_);
  input      p, reset_;
  output     soc;
  input      eoc;
  input [7:0] x7_x0;
  input      ir;
  output     iow_;
  output [7:0] d7_d0;

  reg        SOC,IOW_;    assign soc=SOC, iow_=IOW_;
  reg [7:0]   MBR;        assign d7_d0=MBR;
  reg [2:0]   STAR;       parameter S0=0,S1=1,S2=2,S3=3,S4=4;

  always @(posedge p or negedge reset_)
    if (reset_==0) begin SOC<=0; IOW_<=1; STAR=S0; end else #3
      casex(STAR)
        // Prelievo di un byte dal Convertitore A/D
        S0: begin SOC<=1; STAR<=(eoc==1)?S0:S1; end
        S1: begin SOC<=0; MBR<=x7_x0; STAR<=(eoc==0)?S1:S2; end
        // Attesa della richiesta di interruzione ed emissione del byte
        S2: begin STAR<=(ir==0)?S2:S3; end
        S3: begin IOW_<=0; STAR<=S4; end
        S4: begin IOW_<=1; STAR<=(ir==1)?S4:S0; end
      endcase
endmodule

```