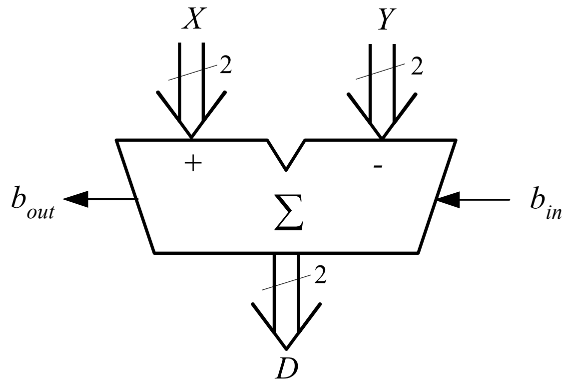
Compito di Reti Logiche del 26/6/18

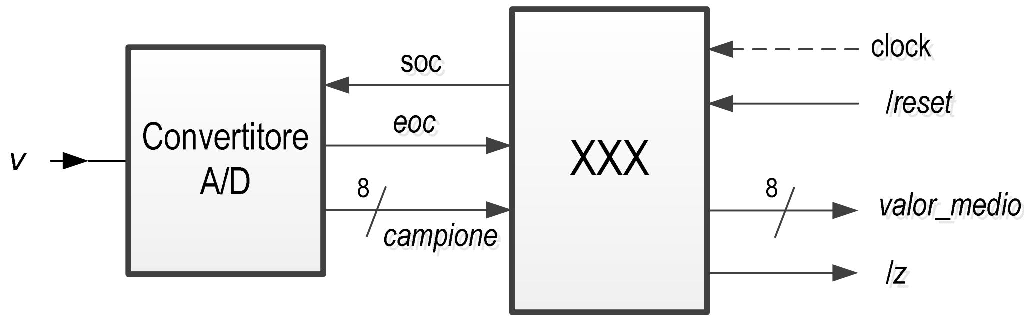
Esercizio 1

In figura è rappresentato un **sottrattore** per naturali a 2 cifre in base 8.

1) Rappresentare la relazione tra gli ingressi e le uscite in forma di equazione.
2) Assumendo sia $Y = (17)_8$, indicare – giustificando la risposta – la configurazione degli ingressi X e b_{in} che produce in uscita:
   a) $b_{out} = 0$, $D = (60)_8$
   b) $b_{out} = 1$, $D = (60)_8$.



Esercizio 2



Il Convertitore A/D opera in **binario bipolare** su 8 bit e l'unità XXX effettua all'infinito le seguenti operazioni:

1. Preleva 32 campioni dal *convertitore A/D*;
2. Calcola la media aritmetica dei valori dei campioni prelevati e la presenta in **binario bipolare** sull'uscita *valor_medio*. L'uscita */z*, inizialmente a 1, viene portata a 0 da XXX dopo che ha presentato sull'uscita *valor_medio* un nuovo risultato e riportata a 1 dopo esattamente un periodo di clock.

Descrivere e **sintetizzare** l'unità XXX, evidenziando circuitalmente come viene calcolato il valor medio.

Soluzione Esercizio 1

La relazione fra ingressi ed uscite di un sottrattore per numeri naturali rappresentati in base β su n cifre è la seguente:

$$-b_{out}\beta^n + D = X - Y - b_{in}.$$

Da essa, si ottiene la seguente equazione per il caso in esame:

$$X - b_{in} = -b_{out}\beta^n + D + Y = -b_{out}(100)_8 + D + (17)_8.$$

Nel primo caso, $b_{out} = 0$ e $D = (60)_8$, sostituendo i valori nell'equazione si ottiene:

$$X - b_{in} = (60)_8 + (17)_8 = (77)_8.$$

Una possibile soluzione è allora $b_{in} = 0$ e $X = (77)_8$. Nel secondo caso, $b_{out} = 1$ e $D = (60)_8$, sostituendo i valori si ottiene invece:

$$X - b_{in} = -(100)_8 + (60)_8 + (17)_8 = -(1)_8.$$

Poiché necessariamente è $X \geq 0$, l'unica soluzione possibile in questo caso è $b_{in} = 1$ e $X = (0)_8$.

Soluzione Esercizio 2

Per leggibilità nella soluzione viene usato il **costrutto duplicatore** $\{N\{\alpha\}\}$ che equivale a $\alpha, \alpha, \ldots, \alpha$, con α espressione qualsiasi ripetuta *N* volte. In alternativa, è possibile usare una rete combinatoria (di complessità nulla) che estende su *n* volte il bit passato come ingresso.

```
module XXX(campione,eoc,soc, valor_medio,z, clock,reset_);
 input        clock,reset_;
 output       z;
 output [7:0] valor_medio;
 output      soc;
 input       eoc;
 input [7:0] campione;
 reg [7:0] VALOR_MEDIO;   assign valor_medio=VALOR_MEDIO;
 reg       Z, SOC;        assign z=Z;  assign soc=SOC;

 reg [12:0] SUM;          // su 13 bit per evitare overflow
 reg [4:0]  COUNT;

 reg  [2:0] STAR;  parameter S0=0,S1=1,S2=2,S3=3,S4=4,S5=5;
 always @(reset_==0) begin Z<=1; SOC<=0; STAR=S0; end
 always @(posedge clock) if (reset_==1) #3
  casex(STAR)
   S0:  begin Z<=1; SUM<=0; COUNT<=0; STAR<=S1; end
   S1:  begin SOC<=1; STAR<=(eoc==1)?S1:S2; end
   S2:  begin SOC<=0; STAR<=(eoc==0)?S2:S3; end
   //Prelievo e somma, previa conversione in complemento a 2
   S3:  begin SUM<=SUM+{{6{!campione[7]}},campione[6:0]};
              COUNT=COUNT+1; STAR<=(COUNT==31)?S4:S1; end
  //Divisione per 32 e riconversione in binario bipolare
   S4:  begin VALOR_MEDIO<={!SUM[12],SUM[11:5]}; STAR<=S5; end
   S5:  begin Z<=0; STAR<=S0; end
  endcase
endmodule
```