

Esercizio 1

Sia X la rappresentazione in complemento alla radice su n cifre in una base generica (pari) β del numero intero x . Sia Y la rappresentazione *in traslazione* dello stesso numero.

- 1) esprimere la relazione algebrica che consente di trovare Y in funzione di X ;
- 2) sintetizzare a costo minimo il circuito che produce Y avendo X in ingresso nel caso $\beta = 6$ (con codifica 421);
- 3) sintetizzare a costo minimo il circuito che produce Y avendo X in ingresso nel caso $\beta = 16$ (con codifica 8421).

Esercizio 2

Descrivere e sintetizzare il circuito XXX che (vedi Fig. 1) legge dalla EPROM e comanda il Convertitore A/D che emette una tensione unipolare $v = K \cdot x$, in accordo alle specifiche che seguono.

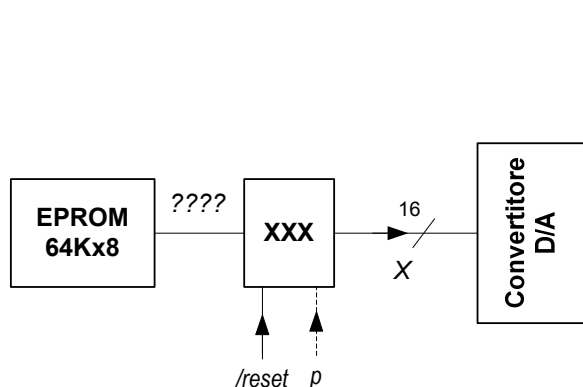


Fig. 1

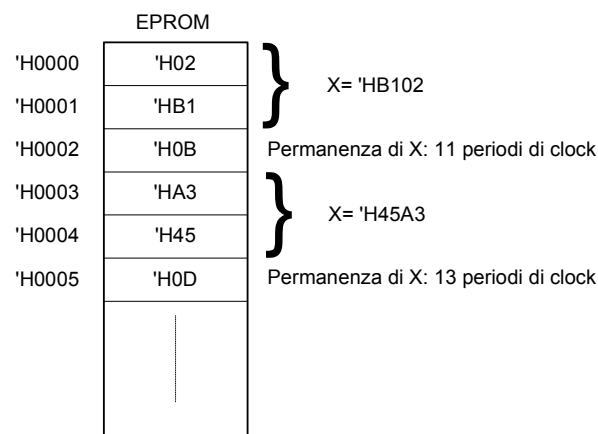


Fig. 2

Nel primo ciclo dopo il reset iniziale XXX emette un campione di valore 0 per (circa) 10 periodi di clock, poi emette quelli prelevati dalla EPROM. L'Unità XXX termina dopo aver emesso 20000 campioni (compreso quello emesso nel primo ciclo successivo al reset iniziale) lasciando in uscita l'ultimo campione emesso.

La struttura della EPROM è la seguente (Fig.2). Le prime due locazioni contengono un campione di x espresso come numero naturale X a 16 bit; la locazione seguente contiene una indicazione su quanto tempo tale campione deve essere mantenuto all'ingresso del convertitore. Stessa cosa per le locazioni successive.

ULTERIORI SPECIFICHE

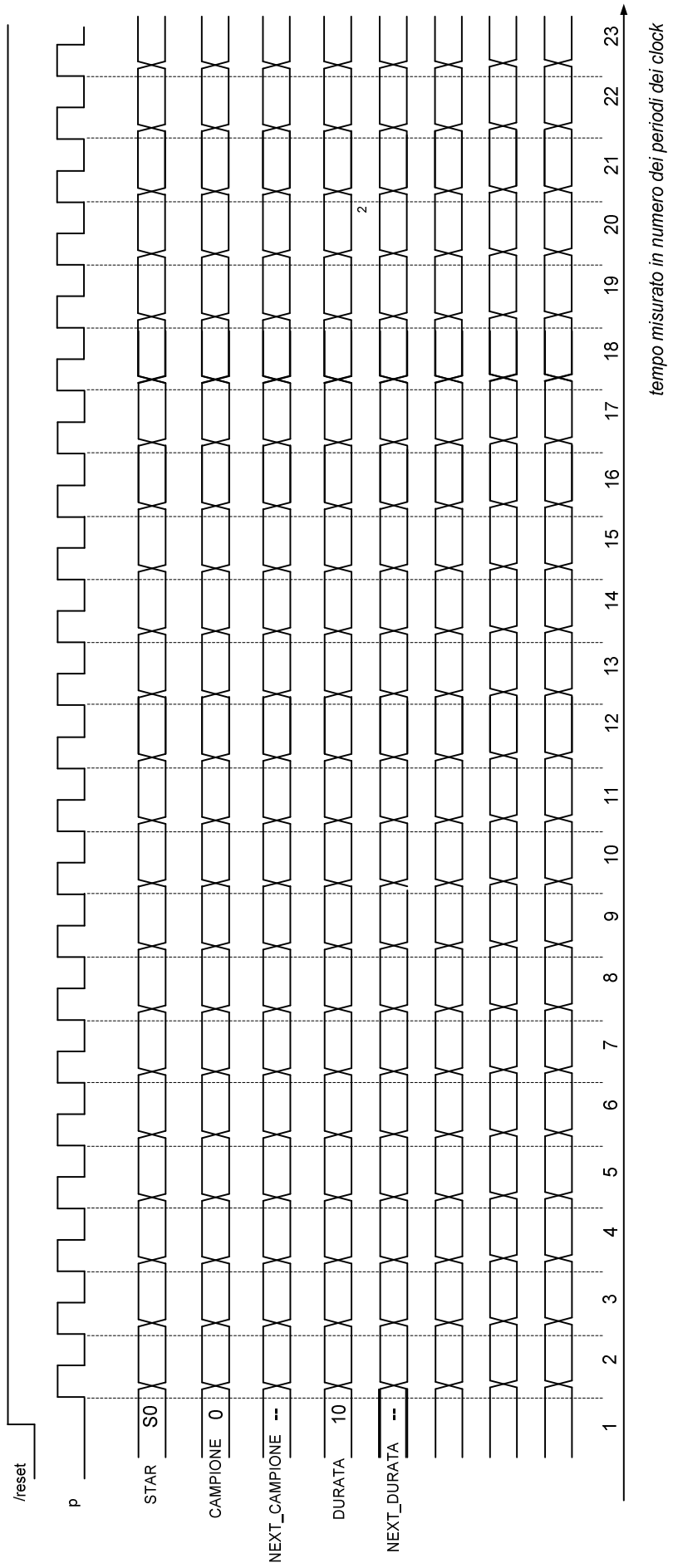
- 1: Gli 8 bit meno significativi di X si trovano nella locazione di indirizzo più basso.
- 2: L'indicazione sul tempo sta tra 10 e 250 e se tale indicazione vale N , allora X deve essere mantenuto per N periodi di clock.
- 3: La EPROM è sufficientemente veloce da non necessitare di stati di wait.

NOTA: Il caso minimo di 10 periodi di clock non può creare problemi in quanto XXX è descrivibile con un numero di stati interni ampiamente minore di 10.

Usare la seguente impostazione

```
module XXX(...,p,reset_);
  input      p,reset_;
  ...
  output [15:0] X;
  reg [15:0]    CAMPIONE,NEXT_CAMPIONE; assign X=CAMPIONE;
  reg [7:0]     DURATA,NEXT_DURATA;
  ...
  always @(posedge p or negedge reset_)
    if (reset_==0) begin DURATA=10; CAMPIONE=0; ...; STAR=S0; end
    else #3
      casex(STAR)
        S0: ...
        ...
      endcase
endmodule
```

Si tracci un diagramma di temporizzazione facendo riferimento alla EPROM di Fig 2 e terminando dopo aver emesso 2 campioni, compreso il campione di valore 0 emesso dopo il reset.



Es 1 - Una possibile soluzione

1) Se x è rappresentabile su n cifre in complemento alla radice, lo è anche in traslazione e viceversa. Pertanto l'operazione di calcolare Y dato X è sempre possibile.

La relazione tra il numero intero x e la sua rappresentazione in complemento alla radice è:

$$x = \begin{cases} X & \text{se } X_{n-1} < \beta/2 \\ X - \beta^n & \text{se } X_{n-1} \geq \beta/2 \end{cases}$$

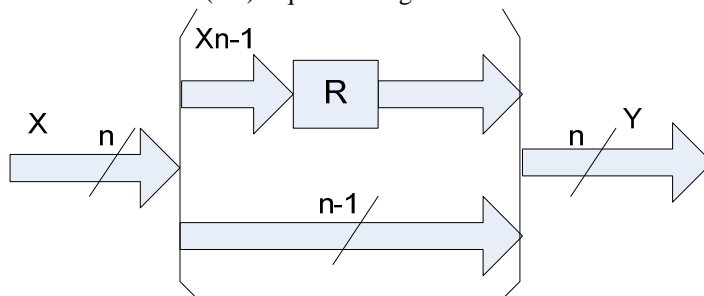
mentre quella tra Y e x è $Y = x + \beta^n/2$. Mettendo insieme le due si ottiene:

$$Y = \begin{cases} X + \beta^n/2 & \text{se } X_{n-1} < \beta/2 \\ X - \beta^n/2 & \text{se } X_{n-1} \geq \beta/2 \end{cases}$$

e considerando che $\beta^n/2 = \beta^{n-1} \cdot \beta/2$, si ottiene rapidamente

$$Y = \begin{cases} (X_{n-1} + \beta/2, X_{n-2}, \dots, X_0) & \text{se } X_{n-1} < \beta/2 \\ (X_{n-1} - \beta/2, X_{n-2}, \dots, X_0) & \text{se } X_{n-1} \geq \beta/2 \end{cases} \quad (1.1)$$

In una base generica, il circuito che realizza la (1.1) è quindi il seguente:



Dove la rete R modifica la cifra di peso più significativo, in accordo all'espressione scritta sopra.

2) Siano $d_2 \dots d_0$ i tre bit che rappresentano la cifra X_{n-1} , e $z_2 \dots z_0$ i tre bit che rappresentano Y_{n-1} . Il circuito che realizza la (1.1) in base 6 si trova come sintesi minima sulla seguente mappa di Karnaugh

$d_2 d_1$		d_0			
		00	01	11	10
d_0	0	011	101	---	001
	1	100	000	---	010
		$z_2 z_1 z_0$			

La cui sintesi SP a costo minimo è la seguente:

$$z_2 = d_1 \cdot \overline{d_0} + \overline{d_2} \cdot \overline{d_1} \cdot d_0$$

$$z_1 = d_2 \cdot d_0 + \overline{d_2} \cdot \overline{d_1} \cdot \overline{d_0}$$

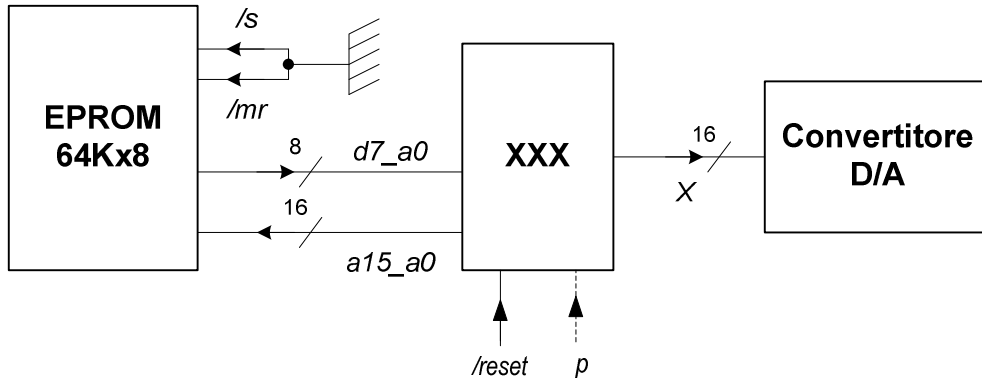
$$z_0 = \overline{d_0}$$

3) Siano $d_3 \dots d_0$ i quattro bit che rappresentano la cifra X_{n-1} , e $z_3 \dots z_0$ i quattro bit che rappresentano la cifra Y_{n-1} . La relazione richiesta è $z_3 = \overline{d_3}$, $z_i = d_i$ per $i = 0, 1, 2$, in quanto un numero in base 16 in codifica 8421 su n cifre ha la stessa rappresentazione di un numero in base 2 su $4n$ cifre, ed è nota dalla teoria dei convertitori la relazione tra la rappresentazione in complemento alla radice ed in traslazione per i numeri in base due. In ogni caso, il circuito che realizza la (1.1) in base 16 si trova come sintesi minima sulla seguente mappa di Karnaugh:

:

$d_3 d_2$		$d_1 d_0$			
		00	01	11	10
$d_1 d_0$	00	1000	1100	0100	0000
	01	1001	1101	0101	0001
	11	1011	1111	0111	0011
	10	1010	1110	0110	0010
		$z_3 z_2 z_1 z_0$			

Es 2 - Una possibile descrizione



```

module XXX(a15_a0, d7_d0,X,p,reset_);

input      p,reset_;
output [15:0] a15_a0;
input  [7:0] d7_d0;
output [15:0] X;

reg [15:0]    MAR;
reg [7:0]     MBR;
reg [15:0]    CAMPIONE,NEXT_CAMPIONE;
reg [7:0]     DURATA,NEXT_DURATA;
reg [2:0]     STAR;
parameter    S0=0, S1=1, S2=2, S3=3, S4=4;

assign X=CAMPIONE;
assign a15_a0=MAR;

always @(posedge p or negedge reset_)
if (reset_==0) begin DURATA=10; CAMPIONE=0; MAR=0; STAR=S0; end
else #0
casex(STAR)
    S0: begin DURATA<=DURATA-1; MBR<=d7_d0; MAR<=MAR+1; STAR<=S1; end
    S1: begin DURATA<=DURATA-1; NEXT_CAMPIONE<={d7_d0,MBR}; MAR<=MAR+1;
        STAR<=S2; end
    S2: begin DURATA<=DURATA-1; NEXT_DURATA<=d7_d0; MAR<=MAR+1;
        STAR<=S3; end

    //Si esce con COUNT==2 per evitare un microsalto a tre alternative in S4
    S3: begin DURATA<=DURATA-1; STAR<=(DURATA==2)?S4:S3; end

    //Quando sono stati emessi W campioni, l'indirizzo di MAR e' (W*3)
    S4: begin CAMPIONE<=(MAR==60000)?CAMPIONE:NEXT_CAMPIONE;
        DURATA<=NEXT_DURATA; STAR=(MAR==60000)?S4:S0; end

endcase
endmodule

```