

Esercizio 1

1) Descrivere e sintetizzare un circuito che ha in ingresso un numero naturale a una cifra in base 6, X , ed un bit di riporto entrante, r_{in} , e produce in uscita un numero naturale a una cifra in base 6, Y , ed un bit r_{out} , dove:

$$X + r_{in} \cdot 6 = Y \cdot 2 + r_{out}$$

In altre parole, Y ed r_{out} sono il quoziente e resto della divisione per due del numero X (se $r_{in} = 0$) o del numero $X + 6$ (se $r_{in} = 1$).

2) Usando il circuito di cui al punto precedente, sintetizzare un circuito che prende in input numeri in base 6 a n cifre e calcola resto e quoziente della loro divisione per due.

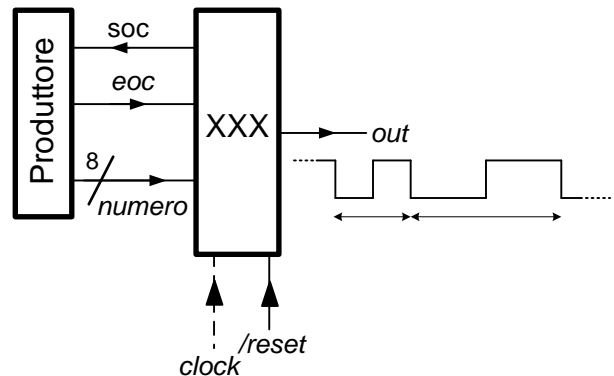
Esercizio 2

Descrivere l'Unità XXX in modo che la variabile di uscita *out* stia a 1 per un tempo pari a quello in cui è stata precedentemente a 0 e la durata in cui starà di nuovo a 0 dipenda dal *numero* che XXX preleva dal Produttore mentre *out* era a 1.

Più precisamente l'Unità XXX si evolve all'infinito nel seguente modo:

- 1) In una prima fase tiene *out* a 0 per un numero di clock pari all'ultimo *numero* che essa ha prelevato dal Produttore;
- 2) Entra quindi in una seconda fase in cui tiene *out* a 1 per uno stesso numero di clock. **Mentre è in corso** questa seconda fase, preleva anche dal Produttore un **nuovo numero**
- 3) Torna al punto 1;

Si tracci il diagramma temporale di una evoluzione completa di XXX. Si chiami NUMERO il registro che contiene il valore di *numero* prelevato da XXX mentre essa sta tenendo *out* a 1. Si chiami COUNT il registro utilizzato per il conteggio della durata di ciascuna delle due fasi e si descriva in Verilog la porzione della Parte Operativa che lo riguarda e poi la corrispondente struttura circuitale.



Si supponga che:

- Il Produttore emetta numeri abbastanza grandi e sia rapido cosicché l'handshake con XXX si concluda nel corso della seconda fase e termini sempre prima che essa termini (senza che si creino problemi a nessuno, di nessun tipo e per nessuna ragione).
- Al reset iniziale XXX ritenga di aver prelevato un *numero* uguale a 6, ovvero:

```
always @(reset == 0) begin
    OUT<=0; SOC<=0; NUMERO<=6;
    COUNT<=...; STAR<=S0; end
```

Esercizio 1 - soluzione

La tabella di verità è la seguente:

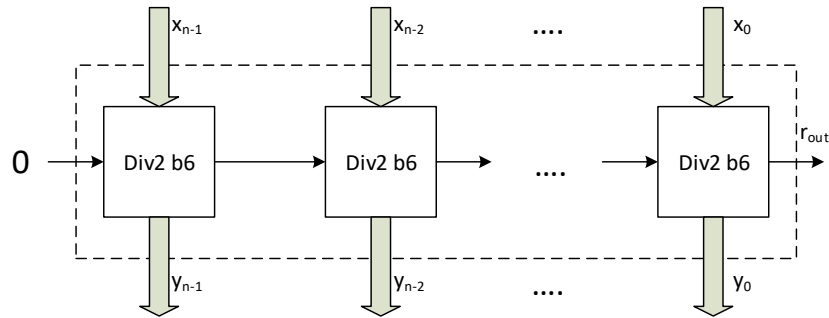
r_{in}	x_2	x_1	x_0	y_2	y_1	y_0	r_{out}
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	1
0	1	0	0	0	1	0	0
0	1	0	1	0	1	0	1
0	1	1	0	-	-	-	-
0	1	1	1	-	-	-	-
1	0	0	0	0	1	1	0
1	0	0	1	0	1	1	1
1	0	1	0	1	0	0	0
1	0	1	1	1	0	0	1
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	-	-	-	-
1	1	1	1	-	-	-	-

Dalla quale si ricava la mappa di Karnaugh:

$x_1 x_0$ \ $r_{in} x_2$					
		00	01	11	10
	00	0000	0100	1010	0110
	01	0001	0101	1011	0111
	11	0011	----	----	1001
	10	0010	----	----	1000
		y_2	y_1	y_0	r_{out}

Una sintesi SP è la seguente: $y_2 = r_{in} \cdot x_2 + \overline{r_{in}} \cdot x_1$, $y_1 = \overline{r_{in}} \cdot x_2 + r_{in} \cdot \overline{x_2} \cdot \overline{x_1}$, $y_0 = \overline{r_{in}} \cdot x_1 + r_{in} \cdot \overline{x_1}$,
 $r_{out} = x_0$

2) Il circuito richiesto non è che il concatenamento dei circuiti visti al punto precedente, secondo lo schema che segue:



Esercizio 2 – soluzione

```
module XXX(soc,eoc,numero,out,clk,reset_);
input      clk,reset_;
input      eoc;
output     soc,out;
input[7:0] numero;
reg        soc,out;
reg[7:0]   COUNT,NUMERO;
reg[1:0]   STAR;    parameter S0=0,S1=1,S2=2,S3=3;

always @(reset_==0) begin OUT<=0; SOC<=0; NUMERO<=6; COUNT<=6; STAR<=S0; end
always @(posedge clk) if (reset_==1) #3
case (STAR)
S0: begin COUNT<=(COUNT==1)?NUMERO:(COUNT-1); OUT<=(COUNT==1)?1:0;
      STAR<=(COUNT==1)?S1:S0; end
S1: begin COUNT<=COUNT-1; SOC<=1; STAR<=(eoc==1)?S1:S2; end
S2: begin COUNT<=COUNT-1; SOC<=0; NUMERO<=numero; STAR<=(eoc==0)?S2:S3; end
S3: begin COUNT<=(COUNT==1)?NUMERO:(COUNT-1); OUT<=(COUNT==1)?0:1;
      STAR<=(COUNT==1)?S0:S3; end
endcase
endmodule
```

