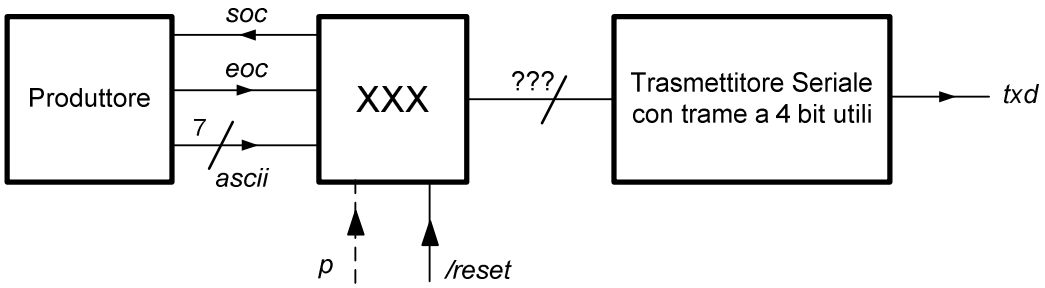
Compito 14/1/2014 - Matricola_____ Nome e Cognome _____

Esercizio 1

1) Descrivere e sintetizzare (come rete SP a costo minimo) un moltiplicatore per interi ad una cifra in base 3. Individuare, classificare e rimuovere eventuali alee sulle singole uscite.
2) Descrivere e sintetizzare (come rete SP a costo minimo) la rete che prende in ingresso l'uscita della rete precedente e produce, su ? bit, il corrispondente numero in base due.

Esercizio 2



Descrivere il l'unità **XXX** che si evolve come segue:

1) Preleva un una Codifica ASCII a 7 bit dal Produttore, sostenendo un **handshake *soc*, *eoc***,
2) Se la Codifica è quella di una cifra decimale, invia al Trasmettitore la rappresentazione binaria della cifra e torna al punto 1, altrimenti torna immediatamente al punto 1.

Per verificare se la codifica è o non è quella buona, si usi una struttura combinatoria del tipo:

```
test =  testM(ascii[6:4]) &
        testL(ascii[3:0])
```

che genera 1 se il test ha successo, 0 altrimenti.

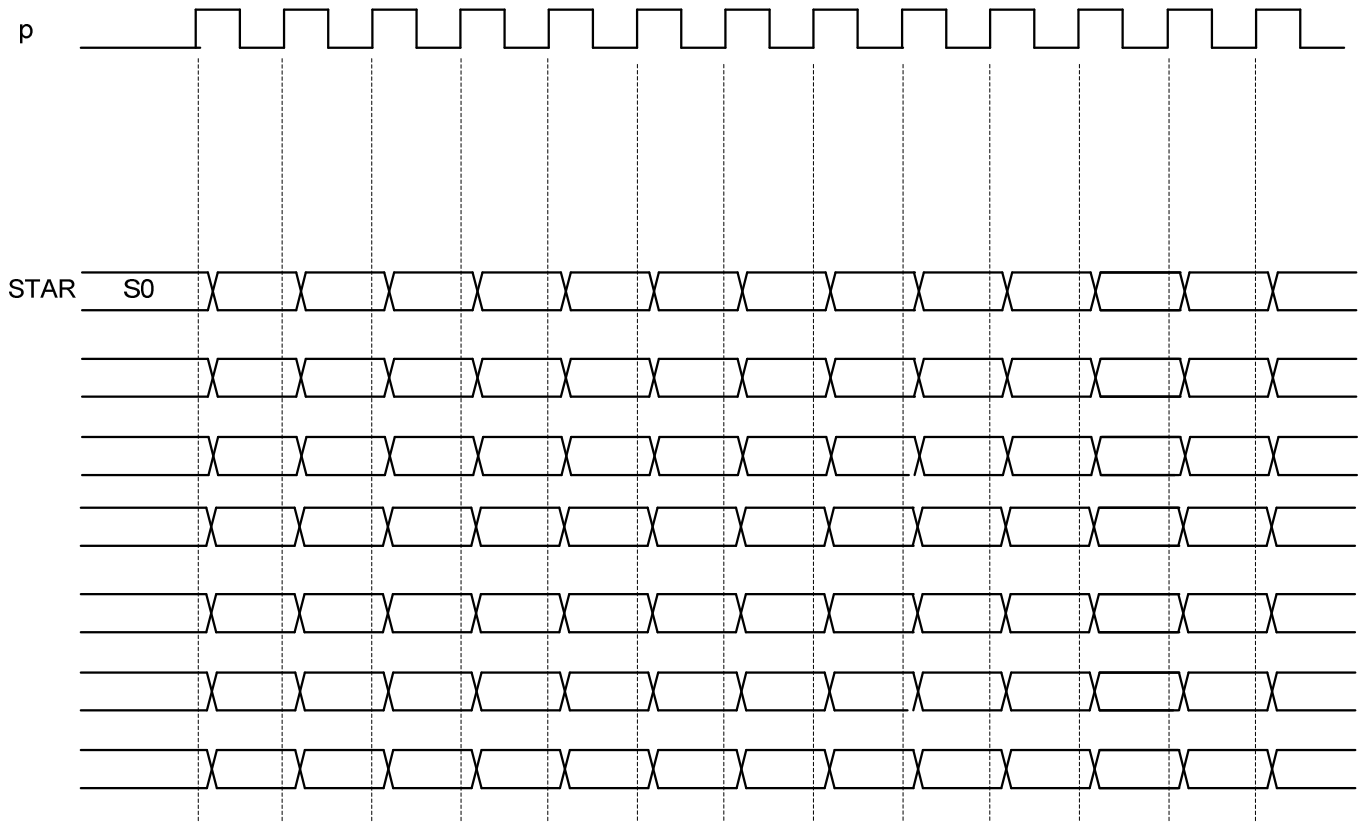
Trovare l'espressione algebrica minima per la sottorete `testL`

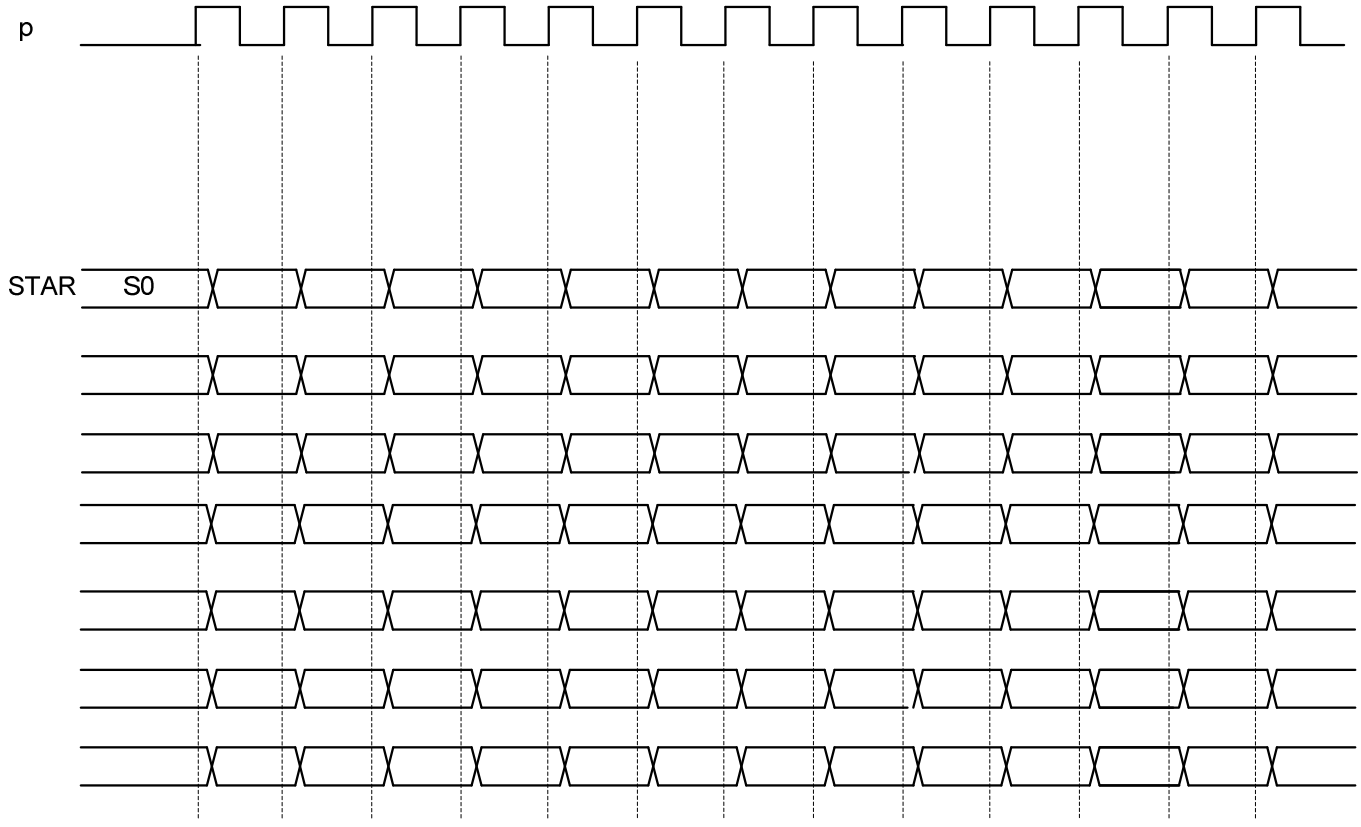
Fare un diagramma temporale che illustri l'evoluzione di **XXX**, supponendo che il Produttore presenti la codifica `'B011_0101`. Affinché il diagramma sia di dimensioni ragionevoli, smettere di tracciarlo quando **XXX** ha attivato il Trasmettitore e supporre anche che la risposta del Produttore durante l'handshake sia abbastanza veloce (tra i uno e due cicli del clock di **XXX**).

000	001	010	011	100	101	110	111	
NUL	DLE	SP	0	@	P	`	p	0000
SOH	XON	!	1	A	Q	a	q	0001
STX	DC2	"	2	B	R	b	r	0010
ETX	XOFF	#	3	C	S	c	s	0011
EQT	DC4	$	4	D	T	d	t	0100
ENQ	NAK	%	5	E	U	e	u	0101
ACK	SYN	&	6	F	V	f	v	0110
BEL	ETB	'	7	G	W	g	w	0111
BS	CAN	(	8	H	X	h	x	1000
HT	EM	)	9	I	Y	i	y	1001
LF	SUB	*	:	J	Z	j	z	1010
VF	ESC	+	;	K	[	k	{	1011
FF	FS	,	<	L	\	l	|	1100
CR	GS	-	=	M	]	m	}	1101
SO	RS	.	>	N	^	n	~	1110
SI	US	/	?	O	_	o	DEL	1111

Codifica (originale) ASCII dei caratteri

Compito 14/1/2014 - Matricola_____ Nome e Cognome _____





Compito 14/1/2014 - Matricola_____ Nome e Cognome _____

Esercizio 1 – Una soluzione

Il moltiplicatore per interi a 1x1 cifra ha un'uscita a due cifre in base 3. Detti *x* e *y* i numeri interi da moltiplicare, *z* il risultato, X, Y, Z le loro rappresentazioni in base 3 in complemento alla radice, e $x_1x_0, y_1y_0, z_3z_2 \mid z_1z_0$ le codifiche in base due delle rappresentazioni di X, Y e Z, si ottiene la seguente mappa di Karnaugh (sono riportati in rosso i numeri interi x e y, ed in blu Z per comodità di lettura):

x_1x_0 \ y_1y_0	0 (00)	+1 (01)	- (11)	-1 (10)
0 (00)	00 00 0 0	00 00 0 0	-- --	00 00 0 0
+1 (01)	00 00 0 0	00 01 0 1	-- --	10 10 2 2
-- (11)	-- --	-- --	-- --	-- --
-1 (10)	00 00 0 0	10 10 2 2	-- --	00 01 0 1

$z_3z_2\,z_1z_0$

Pertanto si ottiene:

$$z_3 = z_1 = x_1 \cdot y_0 + y_1 \cdot x_0$$
$$z_2 = 0$$
$$z_0 = y_1 \cdot x_1 + y_0 \cdot x_0$$

Gli implicanti usati sono tutti essenziali, e non esistono alee di alcun tipo sulle singole uscite.

2) Il moltiplicatore in base 3 ad una cifra produce uno dei seguenti numeri: 0, +1, -1. In base due l'intervallo di numeri [-1; +1] può essere rappresentato su due bit w_1w_0. La mappa di Karnaugh della rete è quindi la seguente:

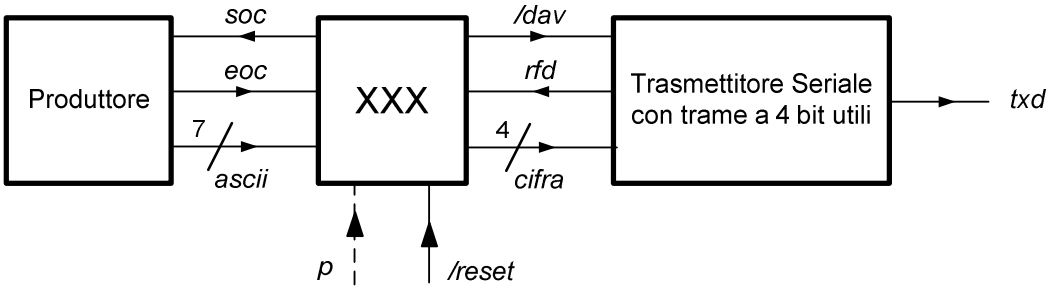
z_3z_2 \ z_1z_0	0 (00)	1 (01)	- (11)	2 (10)
0 (00)	00	01	--	--
1 (01)	--	--	--	--
-- (11)	--	--	--	--
2 (10)	--	--	--	11

w_1w_0

Pertanto, $w_1 = z_3$, $w_0 = z_0 + z_1$.

Compito 14/1/2014 - Matricola_____ Nome e Cognome _____

Esercizio 2 - UNA SOLUZIONE




```
module XXX (soc,eoc,ascii, dav_,rfd,cifra, p,reset_);
 input          p,reset_;
 output         soc;
 input          eoc;
 input   [6:0] ascii;
 output         dav_;
 input          rfd;
 output  [3:0] cifra;
 reg        SOC;      assign soc=SOC;
 reg        DAV_;     assign dav_=DAV_;
 reg [3:0] CIFRA;     assign cifra=CIFRA;
 reg [2:0] STAR;      parameter S0=0,S1=1,S2=2,S3=3,S4=4;
 wire testM ; assign testM=(ascii[6:4]=='B011)?1:0;
 wire testL ; assign testL=(ascii[3:0]<10)?1:0;
 wire test  ; assign test=testM&testL;
 always @(posedge p or negedge reset_)
  if (reset_==0) begin SOC<=0; DAV_<=1;  STAR=S0; end
  else #3
   casex(STAR)
    S0:    begin SOC<=1; STAR<=(eoc==1)?S0:S1; end
    S1:    begin SOC<=0; CIFRA<=ascii[3:0]; STAR<=(eoc==0)?S1:S2; end
    S2:    begin STAR<=(test==1)?S3:S0; end
    S3:    begin DAV_<=0; STAR<=(rfd==1)?S3:S4; end
    S4:    begin DAV_<=1; STAR<=(rfd==0)?S4:S0; end
   endcase
endmodule
```

Forma minima di `testL` (indicando con c_i la variabile `ascii[i]`): $\text{testL} = \overline{c}_3 + \overline{c}_2 \cdot \overline{c}_1$