

## Reti logiche - Prova scritta del 7 Giugno 2016

Cognome e nome: \_\_\_\_\_ Matricola: \_\_\_\_\_

### Esercizio 1

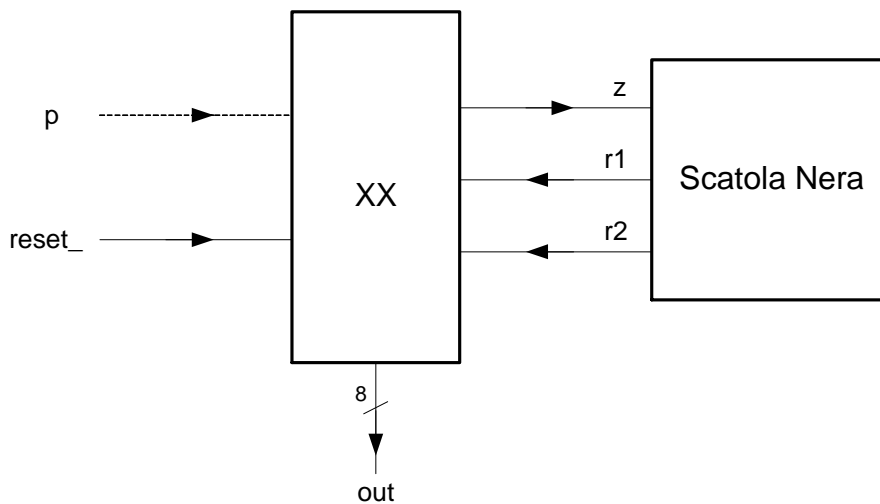
Sintetizzare in forma PS la rete sequenziale asincrona che sostituisce i Latch SR e i formatori di impulsi in un'interfaccia parallela di uscita con handshake (cap. 6, fig. 17 del libro), riportata nella figura a lato. Dimensionare il tempo minimo di permanenza di uno stato di ingresso a tale rete.

SIP \ e <sub>B</sub> rfd						
	00	01	11	10	/dav	fo
S <sub>0</sub>	—	(S <sub>0</sub> )	S <sub>1</sub>	—	1	1
S <sub>1</sub>	—	S <sub>2</sub>	(S <sub>1</sub> )	—	1	0
S <sub>2</sub>	S <sub>3</sub>	(S <sub>2</sub> )	—	—	0	0
S <sub>3</sub>	(S <sub>3</sub> )	S <sub>0</sub>	—	—	1	0

SIS

### Esercizio 2

Si consideri il sistema riportato in figura.



La Scatola Nera, risponde ad ogni transizione della variabile  $z$  con una identica transizione sia della variabile  $r1$  che della variabile  $r2$ . I tempi di risposta non sono noti e nessuna ipotesi è fatta su di essi tranne le seguenti: a) la risposta su  $r2$  arriva dopo quella su  $r1$  e b) entrambe le risposte arrivano entro 200 cicli del clock  $p$ .

Partendo da una condizione iniziale in cui  $z$ ,  $r1$  ed  $r2$  sono a 0, l'unità  $XX$  compie all'infinito le seguenti azioni:

- 1) Porta  $z$  ad 1 e mentre attende che sia  $r1$  che  $r2$  vadano ad 1 compie una semplice elaborazione (vedi specifica successiva)
- 2) Porta  $z$  a 0 e attende che sia  $r1$  che  $r2$  vadano a 0
- 3) Attende che passi del tempo (vedi specifica successiva) ed emette tramite l'uscita  $out$  il risultato dell'elaborazione di cui al punto 1) e torna al punto 1).

### Specifiche:

- L'elaborazione di cui al punto 1) è la seguente: Calcolare un unsigned pari al numero dei periodi di clock che intercorrono tra l'arrivo del fronte in salita su  $r1$  e l'arrivo del fronte in salita su  $r2$ .
- L'uscita  $out$  deve essere cambiata esattamente ogni 500 cicli di clock.

**Descrivere e sintetizzare** la rete  $XX$  rispondente alle specifiche sopra riportate.

## Reti logiche - Prova scritta del 7 Giugno 2016

Cognome e nome: \_\_\_\_\_ Matricola: \_\_\_\_\_

### Soluzione Esercizio 1

Si può osservare che la tabella di flusso presenta molti stati successivi non specificati. Pertanto, l'applicazione del teorema di Unger per capire se la rete è soggetta o meno ad alee essenziali risulta malagevole.

La seguente codifica:  $S_0=00$ ,  $S_1=01$ ,  $S_2=11$ ,  $S_3=10$ , è priva di corse delle variabili di stato. Data questa, la sintesi della rete combinatoria per le uscite è la seguente:  $dav = \overline{y_1} \cdot \overline{y_0}$ ,  $fo = \overline{y_1} + \overline{y_0}$ .

La sintesi PS della RC1 – in accordo ad un modello con elementi neutri di ritardo - è la seguente:

eB rfd		a1 a0			
		00	01	11	10
y1 y0	00	--	00	01	--
	01	--	11	01	--
	11	10	11	--	--
	10	10	00	--	--

eB rfd		/a1 /a0			
		00	01	11	10
y1 y0	00	--	11	10	--
	01	--	00	10	--
	11	01	00	--	--
	10	01	11	--	--

Da cui:  $\overline{a_1} = e_B + \overline{y_0} \cdot rfd$ ,  $\overline{a_0} = rfd + \overline{y_0} \cdot \overline{e_B}$ , e quindi:  $a_1 = \overline{e_B + \overline{y_0} \cdot rfd} = \overline{e_B} \cdot (y_0 + rfd)$ ,  
 $a_0 = \overline{rfd + \overline{y_0} \cdot \overline{e_B}} = rfd \cdot (y_0 + e_B)$ . Le sintesi sono prive di alee del primo ordine. Si noti che, dopo aver scelto gli implicant per RC1, la tabella di flusso risulta tutta specificata come segue:

eB rfd		a1 a0			
		00	01	11	10
y1 y0	00	S3	(S0)	S1	(S0)
	01	S3	S2	(S1)	S0
	11	S3	(S2)	S1	S0
	10	(S3)	S0	S1	S0

La tabella sopra disegnata è normale e non soggetta ad alee essenziali. Pertanto si può concludere che – con questa implementazione di RC1 – il tempo minimo di permanenza di uno stato di ingresso è

$$T = T_{RC1} + 1 \cdot (T_{RC1} + 0) = 2 \cdot T_{RC1}.$$

## Reti logiche - Prova scritta del 7 Giugno 2016

Cognome e nome: \_\_\_\_\_ Matricola: \_\_\_\_\_

### Soluzione Esercizio 2

```
//-----  
module XXX(z,r1,r2,out,clock, reset_);  
input clock, reset_;  
input r1,r2;  
output z;  
output [7:0] out;  
  
reg Z;          assign z=Z;  
reg [7:0] RITARDO, OUT; assign out=OUT;  
reg [8:0] COUNT;  
reg [2:0] STAR; parameter S0=0, S1=1, S2=2, S3=3, S4=4 ;  
  
parameter Num_Periodi=500;  
  
always @(reset_==0) begin STAR=S0; COUNT<=Num_Periodi; end  
always @(posedge clock) if (reset_==1) #3  
casex(STAR)  
  S0: begin COUNT<=COUNT-1; Z<=1; RITARDO<=0; STAR<=S1; end  
  S1: begin COUNT<=COUNT-1; RITARDO<=RITARDO+(r1&(!r2));  
        STAR<=(r2==0)?S1:S2; end  
  S2: begin COUNT<=COUNT-1; Z<=0; STAR<=(r2==1)?S2:S3; end  
  S3 : begin COUNT<=(COUNT==1)?Num_Periodi:COUNT-1; OUT<=(COUNT==1)?RITARDO:OUT;  
        STAR<=(COUNT==1)?S0:S3; end  
endcase  
endmodule  
//-----
```