

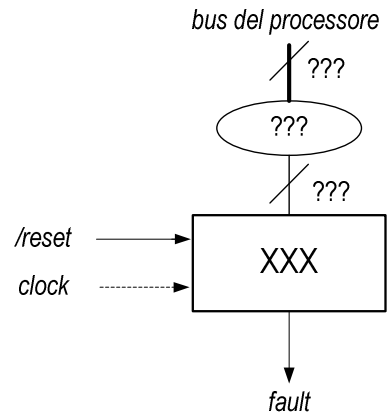
### Esercizio 1

Descrivere e sintetizzare una rete di Mealy che riconosce la sequenza di stati di ingresso 00, 00, 01,xy, dove xy sono due bit di valore diverso. Si presti attenzione a non perdere nessuna sequenza utile. Sintetizzare le reti combinatorie in forma PS.

### Esercizio 2

L'Unità XXX si presenta al processore visto a lezione come una periferica di uscita senza handshake e va collegata al bus in modo che il processore vi acceda eseguendo le istruzioni:

```
MOV $un_byte,AL
OUT AL,0x0323
```



L'Unità XXX dispone sia del classico registro TBR (usato tuttavia per tutt'altri scopi rispetto a un'interfaccia di uscita standard) sia di un registro ad 1 bit FAULT che sostiene la vera variabile di uscita *fault*. Lo scopo dei XXX consiste nel controllare che il processore esegua cicli di accesso in scrittura *validi* e nel porre *fault* a 1 quando ritiene che ciò non accada. Più in dettaglio.

Al reset, XXX inizializza *fault* a 0 e TBR a 1, poi si evolve in accordo alle seguenti specifiche:

1. Si mette in attesa che il processore acceda in scrittura a TBR.
2. Considera *valido* l'*i*-esimo accesso in scrittura ( $i=1, 2, \dots$ ), se il dato che riceve dal processore vale  $|i|_{256}$ .
3. Qualora consideri l'accesso non valido, pone *fault* =1 per **un solo** clock, reinizializza TBR a 1 e quindi riparte dal punto 1
4. Qualora consideri l'accesso valido, predispone il contenuto di TBR per il prossimo ciclo di accesso e quindi riparte dal punto 1.

Disegnare i collegamenti di XXX, montandolo correttamente nello spazio di I/O. Quindi descrivere e sintetizzare XXX disegnando la circuiteria che riguarda il registro TBR.

NOTE:

- a) Si faccia l'ipotesi che il clock che riceve XXX sia **molto più veloce di quello del processore**.
- b) Si noti come TBR sia anche il contatore del numero di accessi

### Esercizio 1 – una possibile soluzione

La tabella di flusso che descrive la rete è la seguente:

$X_1X_0$				
	00	01	11	10
$S_0$	$S_1/0$	$S_0/0$	$S_0/0$	$S_0/0$
$S_1$	$S_2/0$	$S_0/0$	$S_0/0$	$S_0/0$
$S_2$	$S_2/0$	$S_3/0$	$S_0/0$	$S_0/0$
$S_3$	$S_1/0$	$S_0/1$	$S_0/0$	$S_0/1$

Adottando le codifiche  $S_0 = 00$ ,  $S_1 = 01$ ,  $S_2 = 11$ , e  $S_3 = 10$ , con riferimento al modello strutturale con flip-flop D-positive-edge-triggered come elementi di registro, si ottiene quanto segue:

$x_1x_0$	$a_1a_0$			
	00	01	11	10
$y_1y_0$	00	01	11	10
00	01	00	00	00
01	11	00	00	00
11	11	10	00	00
10	01	00	00	00

$x_1x_0$	$z$			
	00	01	11	10
$y_1y_0$	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	0	0	0
10	0	1	0	1

$$\begin{aligned}\bar{z} &= x_1 \cdot x_0 + \bar{x}_1 \cdot \bar{x}_0 + y_0 + \bar{y}_1 \\ z &= \overline{(x_1 \cdot x_0) + (\bar{x}_1 \cdot \bar{x}_0) + y_0 + \bar{y}_1} \\ &= \overline{(x_1 \cdot x_0)} \cdot \overline{(\bar{x}_1 \cdot \bar{x}_0)} \cdot \bar{y}_0 \cdot y_1 \\ &= (\bar{x}_1 + \bar{x}_0) \cdot (x_1 + x_0) \cdot \bar{y}_0 \cdot y_1\end{aligned}$$

$$\begin{aligned}\bar{a}_1 &= \bar{y}_0 + x_1 + \bar{y}_1 \cdot x_0 \\ a_1 &= y_0 \cdot \bar{x}_1 \cdot (y_1 + \bar{x}_0)\end{aligned}$$

$$\begin{aligned}\bar{a}_0 &= x_0 + x_1 \\ a_0 &= \bar{x}_0 \cdot \bar{x}_1\end{aligned}$$

## Esercizio 2 – una possibile soluzione

```
module XXX(s_,iow_,d7_d0, fault, clock,reset_);
  input clock, reset_;
  input s_,iow_;
  input [7:0]d7_d0;
  output fault;

  reg [7:0] TBR;
  reg FAULT;      assign fault=FAULT;
  reg[1:0] STAR; parameter S0=0, S1=1, S2=2;

  always @ (reset_==0) begin FAULT<=0; TBR<=1;
                           STAR<=S0; end
  always @ (posedge clock) if (reset_==1) #3
  casex(STAR)
    S0: begin STAR<=({s_,iow_}=='B00)?S1:S0; end
    S1: begin TBR<=(TBR==d7_d0)?(d7_d0+1):1; FAULT<=(TBR==d7_d0)?0:1;
            STAR<=S2; end
    S2: begin FAULT<=0; STAR<=(iow_==1)?S0:S2; end
  endcase
endmodule
```

**oppure**

```
S0: begin STAR<=({s_,iow_}=='B00)?S1:S0; end
S1: begin TBR<=d7_d0; FAULT<=(TBR==d7_d0)?0:1; STAR<=S2; end
S2: begin FAULT<=0; TBR<=(FAULT==1)?1:(TBR+1); STAR<=S3; end
S3: begin STAR<=(iow_==1)?S0:S3; end
```

