

## Compito di Reti Logiche 15/02/2018

Cognome e Nome: \_\_\_\_\_ Matricola \_\_\_\_\_

Prima della consegna barrare una delle due caselle sottostanti. L'opzione scelta non può essere modificata dopo la consegna.

Chiedo che la mia prova scritta sia corretta e valutata subito, perché intendo sostenere la prova orale in questo appello. Prendo atto che, a seguito della mia decisione, la mia prova scritta cesserà di essere valida al termine di questo appello e non potrà essere usata per l'appello straordinario di Aprile. ☐

Chiedo che la mia prova scritta sia corretta e valutata dopo la fine dell'appello in corso, perché ho diritto a ed intenzione di rimandare la prova orale all'appello straordinario di Aprile. Prendo atto che il mio diritto a rimandare la prova orale sarà oggetto di verifica, e che dovrò ripetere l'intero esame da capo se la verifica darà esiti negativi (per qualunque motivo). ☐

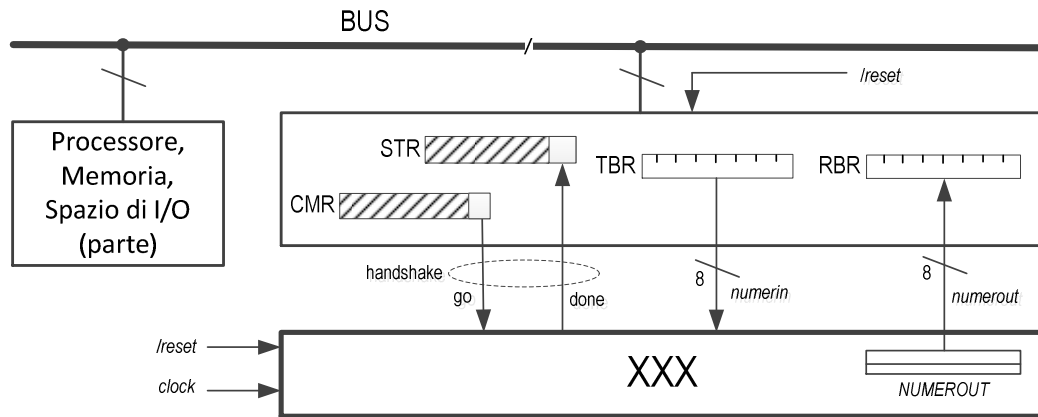
-----

### Esercizio 1

*Descrivere* una rete sequenziale *asincrona* con due variabili di ingresso  $x_1$  e  $x_0$  ed una variabile di uscita  $z$ , che si evolve come segue: partendo da una condizione di stabilità in cui  $x_1 = x_0 = z = 0$ , la variabile  $z$  va ad 1 se e solo se le variabili d'ingresso sono entrambe ad 1, e  $x_0$  è andata ad 1 per ultima; la variabile  $z$  torna a 0 se e solo se le variabili d'ingresso sono entrambe a 0, e  $x_0$  è ritornata a 0 per ultima.

Sintetizzare la rete secondo il modello ad elementi neutri di ritardo, sintetizzando le reti combinatorie in forma PS. Dimensionare i ritardi di marcatura e scrivere quale deve essere il tempo minimo di permanenza di uno stato di ingresso.

(l'esercizio 2 è sul retro del foglio)

**Esercizio 2**

L'Unità XXX dovrà sostenere, con l'interfaccia indicata in figura, un handshake congruente con quanto descritto nel seguente spezzone di programma, prelevando quindi, ad ogni ciclo di handshake, un numero da elaborare e emettendo un numero come risultato (l'elaborazione è dettagliata in seguito). L'interfaccia è stata progettata in modo che al reset iniziale i registri *STR* e *CMR* vengono azzerati.

**Spezzone di programma che gestisce l'interfaccia**

```
byte gestione_interfaccia(byte numero_da_elaborare) {
#define TBR_offset ...
#define RBR_offset ...
#define CMR_offset ...
#define STR_offset ...

byte tmp;

// Invio del numero che XXX deve elaborare
outport(TBR_offset, numero_da_elaborare);

// Handshake: Richiesta a XXX di una nuova elaborazione
// e attesa che il risultato sia pronto
outport(CMR_offset, 0x01);
do {tmp=inport(STR_offset)&0x01;} while (tmp==0x00);

// Handshake: Fase di chiusura
outport(CMR_offset, 0x00);
do {tmp=inport(STR_offset)&0x01;} while (tmp!=0x00);

//Ritorno del risultato della elaborazione
return inport(RBR_offset);
}
```

**Disegnare**, come primo passo, l'handshake completo in termini delle variabili *numerin*, *numerout*, *go*, *done*.

**Descrivere** l'Unità XXX in modo che: a) sostenga l'handshake; b) interpreti come numeri naturali i byte ricevuti tramite *numerin* ed emessi tramite *numerout*; c) emetta, in risposta ad un nuovo numero naturale ricevuto, il numero naturale massimo fra tutti quelli ricevuti, purché sia dispari, altrimenti emetta l'ultimo numero precedentemente emesso. Al reset iniziale si ponga *NUMEROUT* a 'H01

**Sintetizzare** XXX (tutta), evidenziando la parte circuitale relativa al registro *NUMEROUT* e alla logica che *implementa* la condizione indicata nel punto c).

### Esercizio 1 - Soluzione

La rete sequenziale deve riconoscere una sequenza di ingressi del tipo 00, {*qualsunque stato d'ingresso diverso da 10*}, 10, 11 partendo dallo stato stabile  $x_1 = x_0 = z = 0$ , mentre deve riconoscere una sequenza di ingressi del tipo 11, {*qualsunque stato d'ingresso diverso da 01*}, 01, 00 partendo dallo stato stabile  $x_1 = x_0 = z = 1$ . La tabella di flusso corrispondente è la seguente:

$x_1x_0$						$z$
		00	01	11	10	
$S_0$	$S_0$	$S_0$	$S_0$	$S_0$	$S_1$	0
$S_1$	$S_0$	—		$S_2$	$S_1$	0
$S_2$	$S_2$	$S_3$	$S_2$	$S_2$	$S_2$	1
$S_3$	$S_0$	$S_3$	$S_2$	—		1

Adottando le codifiche  $S_0 = 00$ ,  $S_1 = 01$ ,  $S_2 = 11$  e  $S_3 = 10$ , la rete è esente da corse critiche. Con riferimento al modello strutturale con elementi neutri di ritardo (necessari perché la rete è affetta da alee essenziali), le mappe di Karnaugh relative alle uscite della rete CN1 sono:

$y_1y_0$		$x_1x_0$			
		00	01	11	10
00	00	00	00	00	01
01	00	--		11	01
11	11	10	10	11	11
10	00	10	11		--

$a_1a_0$

Le forme PS corrispondenti (esenti da alee statiche) sono:

$$a_1 = (x_0 + y_0) \cdot (x_0 + y_1) \cdot (y_1 + y_0),$$

$$a_0 = (x_1 + y_0) \cdot (x_1 + y_1) \cdot (\overline{x_0} + y_1 + y_0) \cdot (x_1 + \overline{x_0}).$$

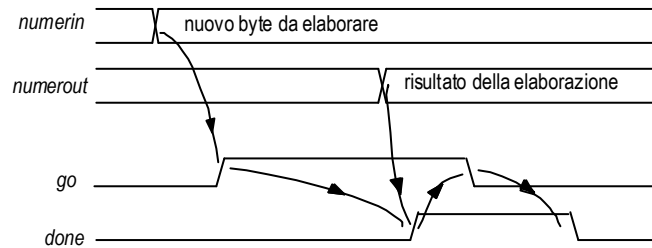
Per la rete CN2, è immediato:

$$z = y_1.$$

Per quanto riguarda il pilotaggio, abbiamo  $T_{mark} \geq T_A$ , e la rete è normale, quindi  $T_{min} = 3 \cdot T_A$ .

## Esercizio 2 - soluzione

### Handshake



### Descrizione di XXX

```
module XXX (numerin,numerout,go,done,clock,reset_);
  input  clock, reset_;
  input  [7:0] numerin;
  output [7:0] numerout;

  input      go;
  output     done;

  reg [7:0] NUMEROUT; assign numerout=NUMEROUT;
  reg      DONE;      assign done=DONE;
  reg [1:0] STAR;      parameter S0=0,S1=1,S2=2;

  always @(reset_==0) begin NUMEROUT<='H01; DONE<=0; STAR<=S0; end
  always @(posedge clock) if (reset_==1)#3
    casex(STAR)
      S0: begin DONE<=0; STAR<=(go==0)?S0:S1; end
      S1: begin NUMEROUT<=((numerin>NUMEROUT)&(numerin[0]==1))?numerin:NUMEROUT;
              STAR<=S2; end
      S2: begin DONE<=1; STAR<=(go==1)?S2:S0; end
    endcase
endmodule
```