

## Reti logiche - Prova scritta del 22 Settembre 2017

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 Novembre.

☐

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 Novembre. 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 combinatoria che:

- Ha come ingresso quattro bit  $x_1, x_0, y_1, y_0$  ed un bit  $b$
- interpreta  $x_1, x_0, y_1, y_0$  come le cifre delle rappresentazioni  $X$  e  $Y$  di due numeri interi  $x$  e  $y$  a 2 bit, che considera il *minuendo* ( $x$ ) ed il *sottraendo* ( $y$ ) di una sottrazione avente riporto entrante  $b$ .
- calcola e restituisce in uscita un bit  $s$  che identifica *segno* del risultato della sottrazione (0 se positivo, 1 se negativo). Nota: il segno del risultato della sottrazione, *non* della sua rappresentazione su 2 bit (che non necessariamente esiste).

Sintetizzare tale rete sotto l'ipotesi che l'ingresso  $b$  sia collegato a VCC. Si segua una sintesi a porte NAND a costo minimo. Individuare, classificare e rimuovere eventuali alee.

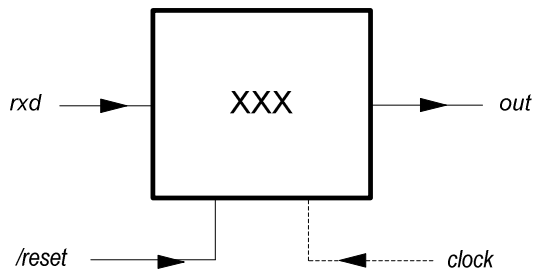
Si usino le seguenti mappe di Karnaugh per la descrizione.

		$x_1x_0$					
		00	01	11	10		
$y_1y_0$	00					$b=0$	
	01						
	11						
	10						

		$x_1x_0$					
		00	01	11	10		
$y_1y_0$	00					$b=1$	
	01						
	11						
	10						

## Esercizio 2

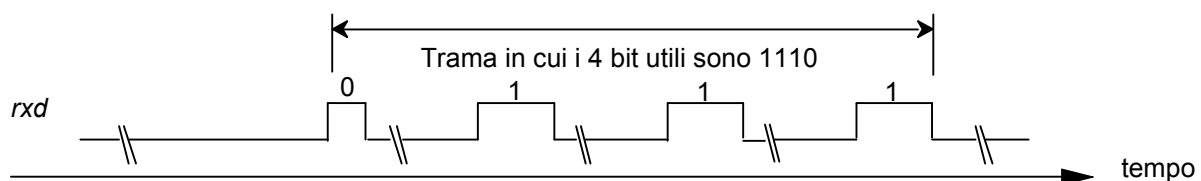
L'Unità XXX è, rispetto alla variabile di ingresso *rx**d*, un ricevitore seriale di trame con 4 bit utili. Ogni volta che ha ricevuto una trama, XXX ne confronta i due bit più significativi con i due bit meno significativi: se il confronto dà esito positivo (coincidenza), XXX mette ad 1 la variabile *out* per un ciclo di clock, altrimenti lascia tale variabile a 0. Torna quindi ad aspettare una nuova trama, e così via all'infinito.



Il formato delle trame è illustrato sotto ed è *estremamente diverso* da quello delle trame viste a lezione e non ci sono ne' il bit di START ne' il bit di STOP. Il bit ricevuto per primo è comunque, come nelle trame viste a lezione, il bit meno significativo dei 4 bit utili.

In dettaglio:

- Tra un bit utile e un altro e tra una trama e un'altra, *rx**d* sta a 0 per un tempo imprecisato, ma sufficientemente lungo da non creare alcun problema di alcun tipo;
- L'arrivo di un bit utile è notificato dalla circostanza che *rx**d* va a 1;
- La durata della permanenza di *rx**d* a 1 indica se un bit utile vale 1 oppure 0, in accordo alle seguenti specifiche:
  - a) *rx**d* permane a 1 esclusivamente per 5 o per 10 cicli di clock
  - b) Se *rx**d* permane a 1 per 5 (cioè per **0**101) cicli di clock, allora il bit utile vale **0**
  - c) Se *rx**d* permane a 1 per 10 (cioè per **1**010) cicli di clock, allora il bit utile vale **1**



### NOTA

- Usare un registro a 4 bit di nome DURATA per memorizzarvi la permanenza di *rx**d* ad 1. Un registro a 4 bit di nome BUFFER per memorizzarvi i quattro bit utili della trama via via che arrivano. Un registro di nome COUNT per contare e verificare che i quattro bit della trama siano arrivati.

**Descrivere XXX e sintetizzare e disegnare lo schema** della parte operativa relativa al registro BUFFER

## Soluzione Es 1

		b=0			
		$x_1x_0$ 00	01	11	10
$y_1y_0$	00	0	0	1	1
	01	1	0	1	1
	11	0	0	0	1
	10	0	0	0	0

		b=1			
		$x_1x_0$ 00	01	11	10
$y_1y_0$	00	1	0	1	1
	01	1	1	1	1
	11	0	0	1	1
	10	0	0	0	1

Tutti gli implicant principali A, B, C, D, E sono essenziali. La sintesi di costo minimo è quindi  $s = x_1 \cdot \overline{y_1} + \overline{x_0} \cdot \overline{y_1} + x_1 \cdot y_0 + x_1 \cdot \overline{x_0} + \overline{y_1} \cdot y_0$ . La sintesi non può presentare alee di nessun tipo, essendo ottenuta usando tutti gli implicant principali.

		b=1			
		$x_1x_0$ 00	01	11	10
$y_1y_0$	00	1	0	1	1
	01	1	1	1	1
	11	0	0	1	1
	10	0	0	0	1

Diagramma Karnaugh con implicant principali evidenziati:

- A**: Implicant  $x_1 \cdot \overline{y_1}$  (cella 00, 11)
- B**: Implicant  $\overline{x_0} \cdot \overline{y_1}$  (celle 00, 01)
- C**: Implicant  $x_1 \cdot y_0$  (celle 11, 10)
- D**: Implicant  $\overline{x_0} \cdot y_0$  (celle 01, 10)
- E**: Implicant  $\overline{y_1} \cdot y_0$  (celle 00, 01, 11, 10)

## Esercizio 2 - Una possibile soluzione

```

module XXX(out, rxd, clock, reset_);
    input      clock, reset_;
    input      rxd;
    output     out;
    reg [3:0] DURATA;
    reg [3:0] BUFFER;
    reg [2:0] COUNT;
    reg      OUT; assign out=OUT;
    reg [1:0] STAR; parameter S0=0, S1=1, S2=2, S3=3;
    // Come evidenziato nel testo, il bit DURATA[3] coincide con il bit utile
    wire bit_utile; assign bit_utile=DURATA[3];
    always @(reset_==0) begin OUT=0; DURATA<=0; COUNT=4; STAR=S0; end
    always @(posedge clock) if (reset_==1) #3
        casex(STAR)
            S0: begin OUT<=0; DURATA<=DURATA+rxd; STAR<=(rxd==0)?S0:S1; end
            S1: begin DURATA<=DURATA+rxd; STAR<=(rxd==1)?S1:S2; end
            S2: begin BUFFER<={bit_utile, BUFFER[3:1]};
                  COUNT<=COUNT-1; DURATA<=0; STAR<=(COUNT==1)?S3:S0; end
            S3: begin OUT<=(BUFFER[3:2]== BUFFER[1:0])?1:0; COUNT<=4;
                  STAR<=S0; end
        endcase
endmodule

```