

## Esercizio 2: Verilog

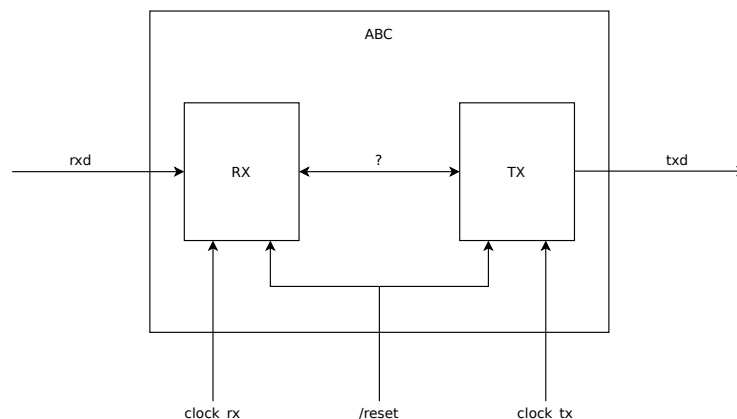


Figura 1: Schema del sistema

L'unità ABC in Figura 1 è un adattatore tra due linee seriali start/stop, che seguono entrambi lo standard visto a lezione ma con *tempi di bit* completamente diversi.

Di conseguenza, l'unità è composta da due reti sequenziali sincronizzate, RX e TX, che si occupano rispettivamente della ricezione (dalla prima linea) e trasmissione (sulla seconda linea) dei byte. Le due reti ricevono due segnali di clock completamente diversi, ciascuno dimensionato in base alla linea seriale a cui la rete è connessa:

- clock\_rx è dimensionato in modo che il suo periodo sia 1/16 del tempo di bit della linea rxd
- clock\_tx è dimensionato in modo che il suo periodo sia esattamente uguale al tempo di bit della linea txd

L'unità RX, ciclicamente, si comporta come segue:

- riceve un byte tramite la linea seriale rxd
- passa tale byte all'unità TX, secondo un protocollo la cui specifica è a scelta dello studente

L'unità TX, ciclicamente, si comporta come segue:

- ottiene un byte dall'unità RX, secondo un protocollo la cui specifica è a scelta dello studente
- trasmette tale byte tramite la linea seriale txd

Descrivere in Verilog l'unità ABC composta dalle unità RX e TX, come da schema in Codice 1.

Sintetizzare l'unità TX in accordo al modello con parte operativa e parte controllo. Si supponga che la parte controllo sia implementata secondo un modello basato su microindirizzi e si scriva la struttura della ROM come commento alla parte controllo medesima.

Note:

- Il protocollo di comunicazione tra RX e TX, che include sia l'interconnessione tra questi sia gli algoritmi seguiti nell'usare tale interconnessione, è a scelta dello studente. Sia la scelta che l'implementazione di tale protocollo sono oggetto di valutazione, secondo quanto parte del programma d'esame.
- I tempi di bit e l'intervallo tra un byte e l'altro sono tali per cui non si crea mai alcun problema di temporizzazione. Più in dettaglio
  - l'unità RX avrà sempre abbastanza tempo per passare un byte a TX prima di ricevere il successivo tramite rxd

- l'unità TX avrà sempre abbastanza tempo per trasmettere un byte tramite txd prima di ottenere il successivo da RX
- Non si faccia alcuna ipotesi circa le relazioni tra clock\_rx e clock\_tx, oltre a quanto consegue dalla nota precedente.

Codice 1: Schema in Verilog dell'unità ABC

```
module ABC(rxd, txd, clock_rx, clock_tx, reset_);
    input rxd, clock_rx, clock_tx, reset_;
    output txd;

    //dichiarazione interconnessione tra RX e TX
    ...

    RX rx(
        .rxd(rxd), .clock_rx(clock_rx), .reset_(reset_),
        ...
    );

    TX tx(
        .txd(txd), .clock_tx(clock_tx), .reset_(reset_),
        ...
    );
endmodule

module RX(rxd, clock_rx, reset_, ...);
    ...
endmodule

module TX(txd, clock_tx, reset_, ...);
    ...
endmodule
```