

POLITECNICO DI TORINO

Laurea Magistrale in Ingegneria Elettronica

Corso di Sistemi Digitali Integrati

Relazione UART



Professore

Massimo Rou Roch

Studenti

Cavagnetto Matteo:290149

Cicero Rosalia: 282502

Rizzello Stefano:288013

Sommario

Introduzione	4
Trasmettitore	6
Pallogramma Control Unit Trasmettitore	6
Timing Diagram Trasmettitore	7
Ricevitore	8
Pallogramma Control Unit Ricevitore	9
Timing Diagram Ricevitore	10
Bus Interface	11
Pallogramma Bus Interface	12
Timing Diagram Bus Interface	13
ATN.....	14
Pallogramma ATN.....	15
Timing Diagram ATN.....	16
Risultati Modelsim.....	17

Introduzione

In questo laboratorio è stato progettato un “Universal Asynchronous Receiver-Transmitter”, dispositivo impiegato nella conversione di flussi di bit di dati, da un formato parallelo ad un formato seriale e viceversa.

La modalità di trasmissione viene definita asincrona poiché il trasmettitore ed il ricevitore non condividono lo stesso segnale di clock. Per questa ragione è necessario definire un meccanismo di sincronizzazione, tale da permettere lo scambio di bit tra i due componenti. Per realizzare quanto detto si è stabilito che il frame debba iniziare con un bit di start forzato allo 0 logico, seguito da 8 bit di dato ed infine 1 bit di stop.

La UART in questione andrà a gestire la comunicazione seriale tramite il protocollo RS-232, il quale prevede la possibilità di una comunicazione “Full-Duplex”, in cui il trasmettitore e il ricevitore, indipendenti tra di loro, lavorano in contemporanea.

Lavorando con un protocollo asincrono, per avere delle informazioni sulla trasmissione dei simboli quello che si deve fare è fissare la velocità di trasmissione del ricevitore e del trasmettitore, ovvero il Baud rate, il quale definisce la durata di un simbolo. In questo caso specifico, è stato fissato ad una velocità di 115200 baud/s, per cui il tempo di simbolo sarà $8.68\mu s$. Il clock di sistema usato è a 16 MHz per cui un tempo di simbolo equivale a circa 139 colpi di clock.

L'intero blocco UART richiede che venga trasmesso e ricevuto un byte, quindi 8 bit, seguendo le seguenti specifiche di interfaccia:

- un bus dati in ingresso su 8 bit Din (7:0);
- un bus dati in uscita su 8 bit Dou (7:0);
- il clock;
- Rx_p: interfaccia seriale;
- Tx_p: interfaccia seriale;
- segnale di Chip Select attivo alto;
- un segnale $Read/\overline{Write}$;
- un bus di indirizzi su 3 bit;
- un segnale ATN;
- un segnale ATNACK.

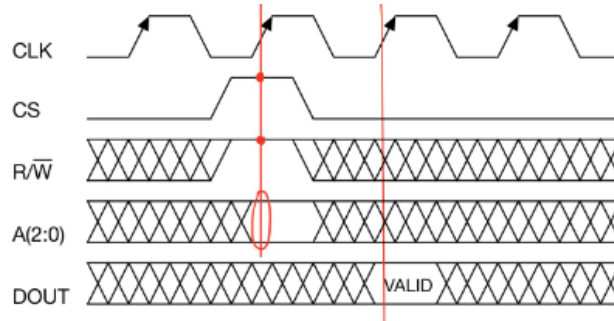
Per l'interfaccia a registri:

- RegTx: contenente il dato da trasmettere;
- RegRx: contenente il dato ricevuto;
- registro di stato: contenente le informazioni relative allo stato in cui si trova la UART;
- registro di controllo.

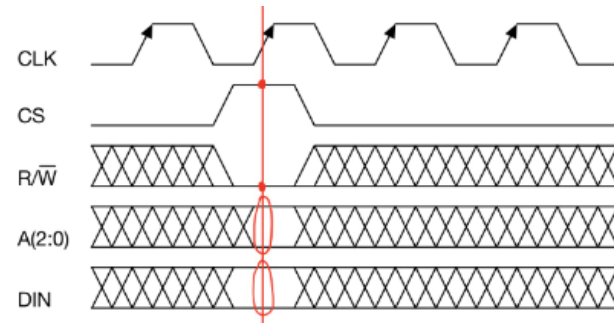
Per quanto riguarda il registro Rx, è stata utilizzata la tecnica a doppio buffer, in modo tale che il dato appena ricevuto venga salvato interamente in un ulteriore registro, al fine di concedere all'utente la possibilità di leggere il dato per un intero tempo di trasmissione, prima che questo venga sovrascritto dal nuovo dato.

I segnali devono rispettare i timing dati da specifica:

- In lettura



- In scrittura



Il progetto della UART è stato realizzato dividendo la struttura in quattro blocchi: il Trasmettitore, il Ricevitore, il Bus Interface ed infine il blocco ATN.

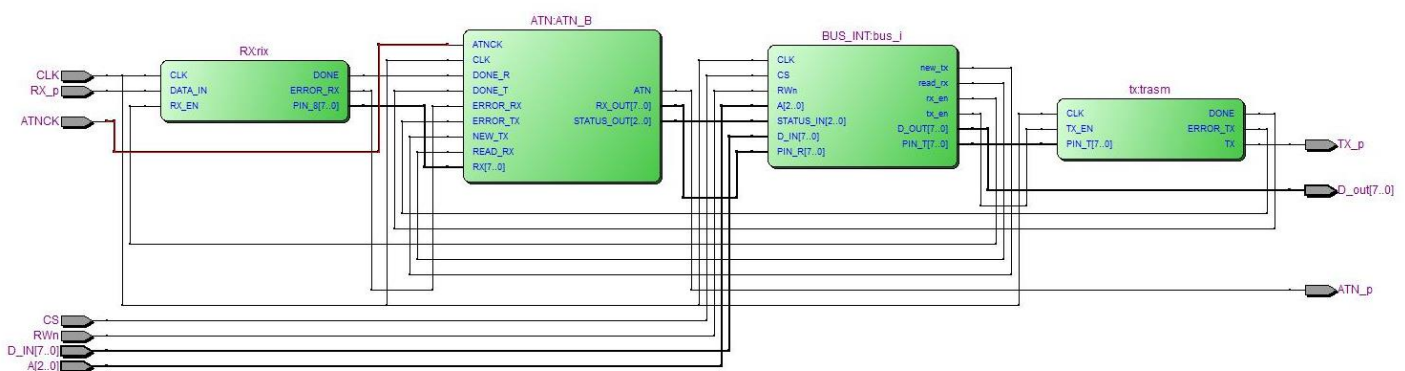


Figura 1: Data Path UART

Di seguito la descrizione di ogni singolo componente.

Trasmittitore

Il blocco trasmettitore è stato progettato come mostrato in Figura 2:

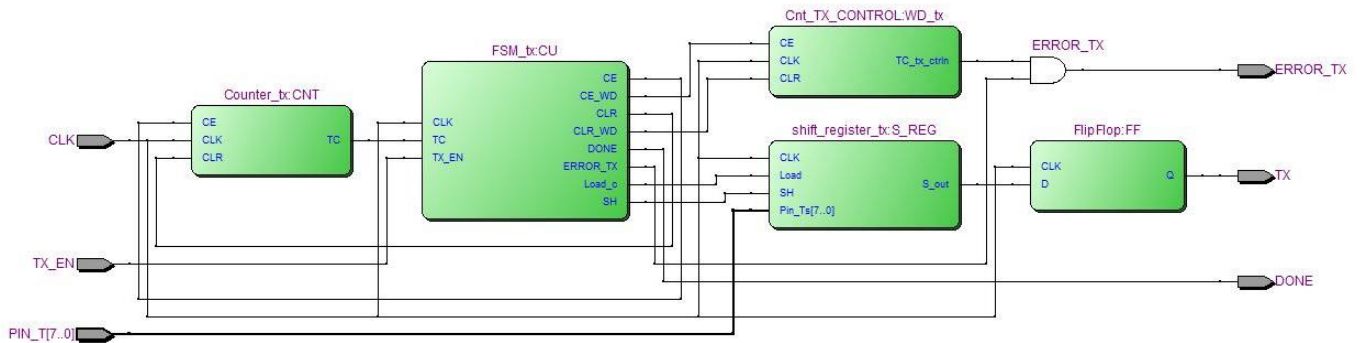


Figura 2: Data Path Trasmittitore

Come è possibile osservare i blocchi costituenti sono:

- Control Unit il cui pallogramma è in Figura 3
- Shift Register: in cui all'interno vengono caricati in parallelo gli 8 campioni da trasmettere;
- questo componente ha però un parallelismo di 10 bit in cui vengono aggiunti i bit di start e stop alla sequenza degli 8 bit di dato;
- Contatore CNT: atto a temporizzare la comunicazione seriale;
- Flip-Flop: posto sull'uscita seriale.
- Contatore di controllo: un hardware aggiuntivo posto a controllare che il segnale di completamento (done_tx) arrivi dopo che sia trascorso un tempo ragionevole all'esecuzione di una trasmissione intera. (questo componente presente anche al ricevitore è completamente superfluo perché si suppone non ci siano errori di progettazione)

Pallogramma Control Unit Trasmittitore

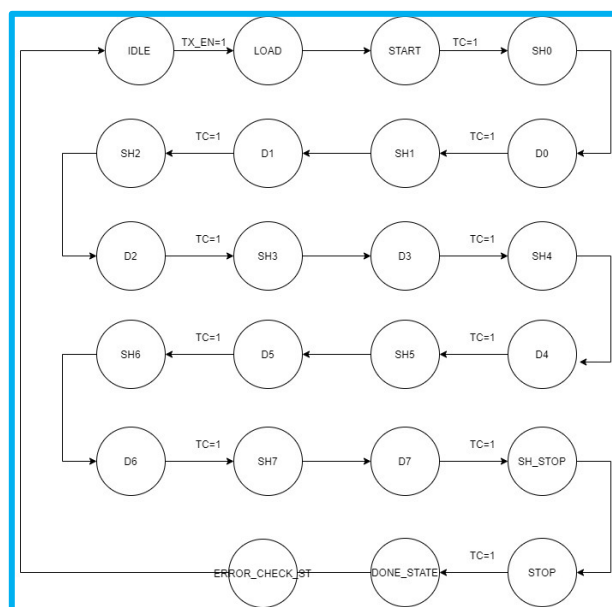


Figura 3: Pallogramma CU trasmettitore

Per il blocco trasmettitore, è stato individuato un pallogramma composto da 23 stati. La macchina si trova in uno stato di Idle in cui rimane fin quando non arriva il segnale TX_EN che abilita la trasmissione. Si passa così allo stato di Load in cui il dato presente sul Tx_Data viene caricato sullo shift register interno al trasmettitore. Passando allo stato di Start, il trasmettitore manda sulla porta seriale la codifica di inizio trasmissione, nonché il bit di start a 0. Da questo stato in poi verrà utilizzato il segnale TC del contatore, opportunamente resettato negli stati di shift, per temporizzare lo scalamento del registro di shift; così facendo i dati vengono trasmessi serialmente con frequenza voluta. Trasmesso l'ultimo dato, la macchina giungerà al completamento del protocollo, passando quindi in uno stato di Stop, in cui verrà inviato appunto l'omonimo bit che determinerà la fine della trasmissione. È anche presente uno stato Error_check, in cui ci si assicura che il segnale di Done arrivi dopo un tempo opportuno. La presenza di questo ultimo stato è dettata dall'impiego del contatore di controllo descritto precedentemente, che viene abilitato a contare ad inizio trasmissione. La macchina tornerà poi nuovamente nello stato di Idle.

Timing Diagram Trasmettitore

In Figura 4 si riporta il timing diagram del trasmettitore.

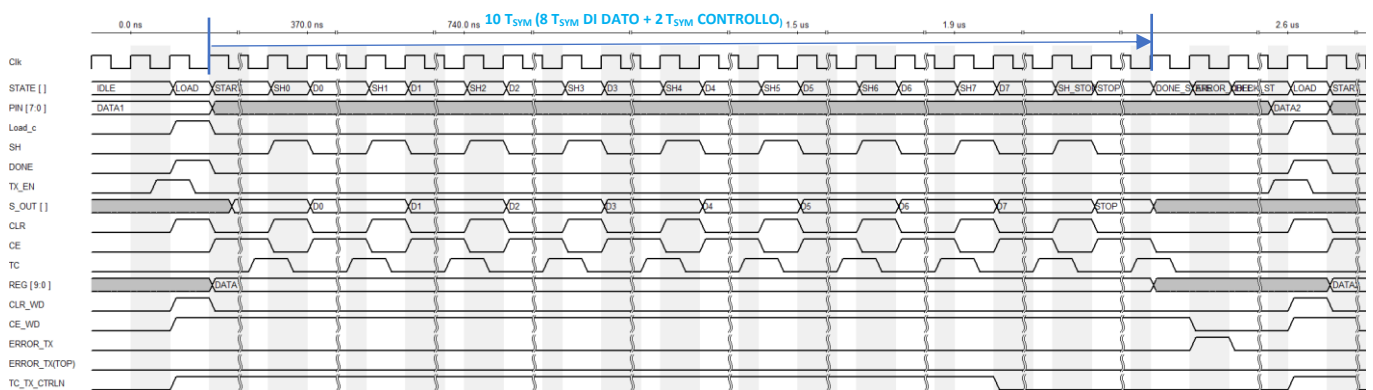


Figura 4: Timing diagram del blocco Trasmettitore

Ricevitore

Dopo avere progettato il trasmettitore, si è proseguito con il blocco ricevitore, realizzato come in Figura 5:

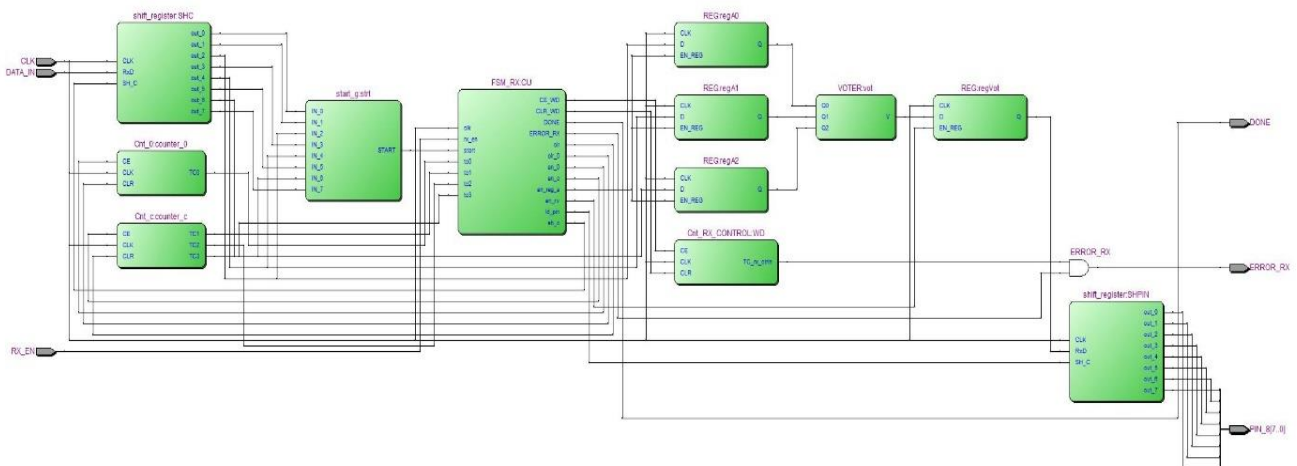


Figura 5: Data Path Ricevitore

I blocchi costituenti sono:

- Control Unit, il quale pallogramma è mostrato in Figura 8;
- Shift Register: in cui in modo seriale e continuativo vengono caricati e shiftati i singoli campioni ricevuti, fino ad un massimo di 8 campioni;
- Contatore 0: impiegato nella conta dei bit ricevuti e controllare lo shift register del dato in uscita al ricevitore;
- Contatore c: volto a temporizzare le fasi di campionamento;
- Logica di start: componente che ricevendo i campioni discrimina l'invio del segnale di start;
- 3 registri su cui memorizzare i 3 campioni da inviare al Voter;
- Voter: logica combinatoria che stima il risultato facendo una media su 3 campioni ricevuti, e decidendo a maggioranza. Questo blocco viene impiegato per andare a ridurre la probabilità che ci siano degli errori in fase di campionamento.
- Registro del Voter: in cui vengono salvati in sequenza i bit in uscita del Voter.
- Shift Register: è un blocco con ingresso seriale ed uscita parallela, in cui vengono salvati gli 8 bit che escono dal Voter.
- Contatore di controllo: ovvero un hardware aggiuntivo posto a controllare che il segnale di completamento (Done_rx) arrivi dopo che sia trascorso un tempo ragionevole all'esecuzione di una trasmissione intera.

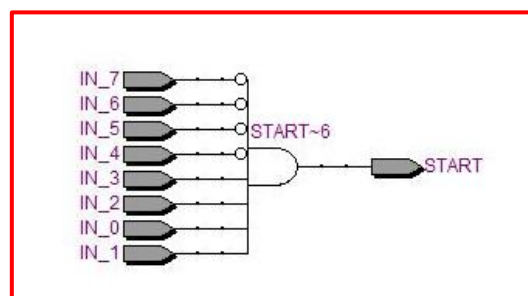


Figura 6: Logica del blocco Start

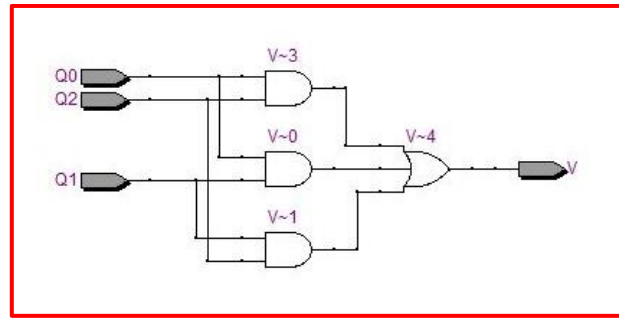


Figura 7: Logica del blocco Voter

Pallogramma Control Unit Ricevitore

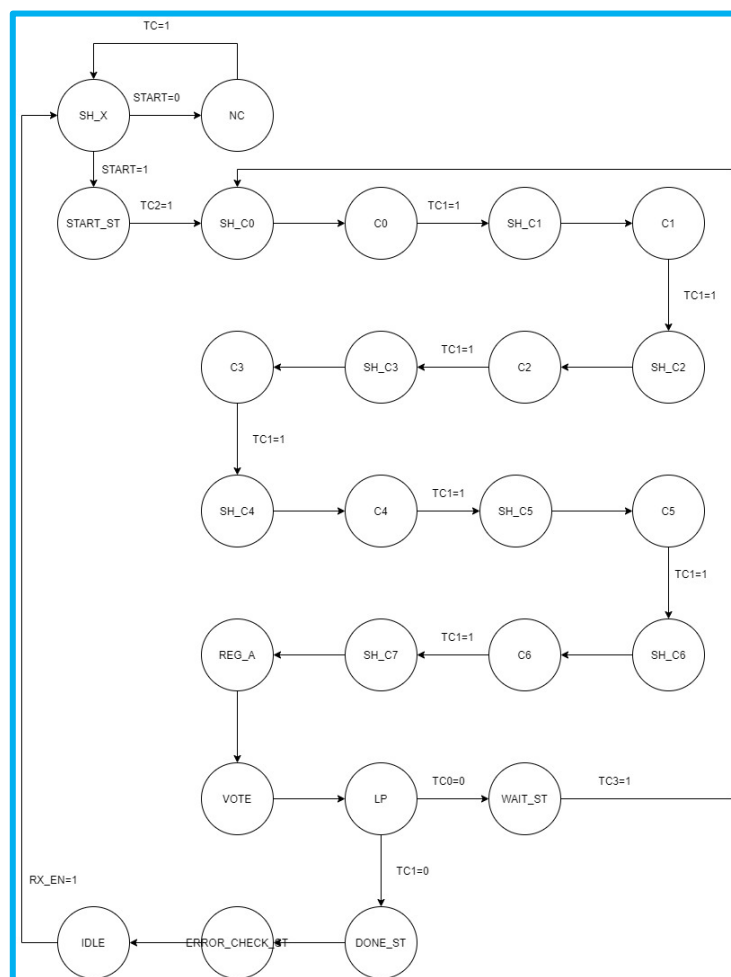


Figura 8: Pallogramma CU ricevitore

Il pallogramma illustrato in Figura 8 descrive il funzionamento del blocco ricevitore.

Partendo da uno stato di Idle se viene ricevuto il segnale Rx_EN, viene abilitata la ricezione e si passa ad uno stato in cui la macchina inizia a campionare i dati in ingresso al fine di individuare il segnale di start. Se questo viene individuato, si passa all'omonimo stato START_st in cui la macchina tramite

il segnale TC2 attende che sia trascorso il simbolo di inizio trasmissione che precede i dati da campionare. Fatto ciò, inizia il campionamento dei dati, prendendo 8 campioni per ogni simbolo; questo è reso possibile tramite segnale TC1 del contatore opportunamente resettato. Giunti all'ultimo campionamento, si andrà nello stato REGA in cui i campioni 2,4 e 6 vengono caricati sui tre registri regA0, regA1, regA2 i quali invieranno tali campioni al Voter. Passando quindi nello stato LP in cui la macchina avendo ottenuto il primo simbolo, lo caricherà nello shift register d'uscita. Tramite il contatore 0 è possibile ripetere tale procedura per tutti ed 8 i simboli; nel caso in cui venga ricevuto l'ottavo campione la macchina va nello stato di Done_st per concludere l'operazione. Viceversa, andrà in uno stato di attesa WAIT regolato terminal count 3 del counter C per continuare il campionamento.

Timing Diagram Ricevitore

In Figura 9 si riporta il timing diagram del trasmettitore.

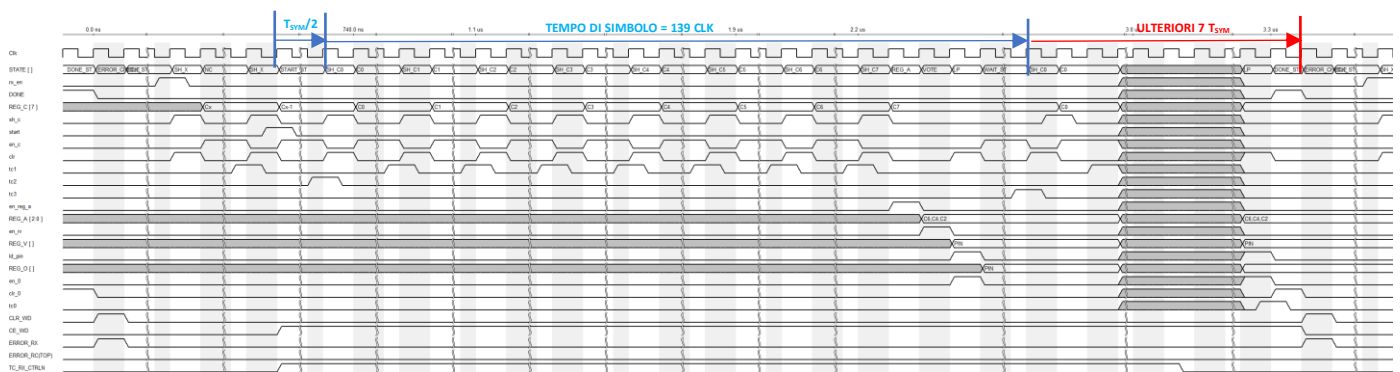


Figura 9: Timing Diagram blocco Ricevitore

Come è possibile osservare in Figura 9, un tempo di simbolo sarà dato da:

$$T_{SYM} = 139clk$$

8 clk per gli 8 shift e 3 clk per gli stati REGA , VOTE, LP

$$139clk = 8clk + 3clk + 7C + Wait$$

$$Wait + 3 = C$$

$$139clk = 8clk + 3 + 8 Wait + 21$$

Il tempo di Wait sarà dunque $13clk$, mentre il tempo di attesa tra un campione e l'altro indicato con C sarà $16clk$.

Bus Interface

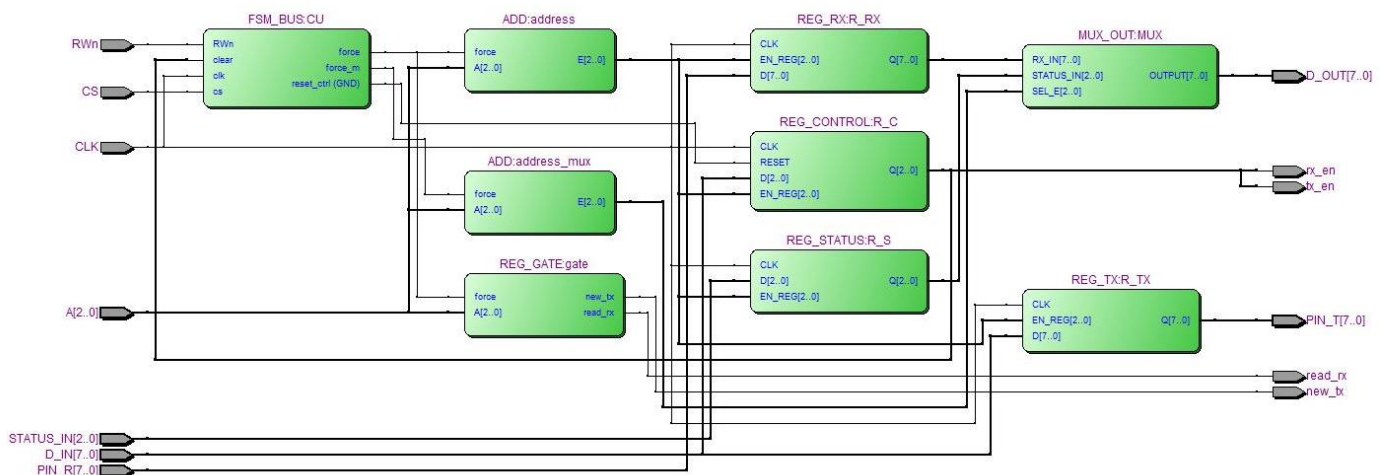


Figura 10: Data Path Bus Interface

Questo blocco svolge la funzione di relazionarsi con l'interfaccia esterna della UART e comprende i seguenti componenti:

- Control Unit, il quale pallogramma è in Figura 13;
- 4 registri di interfaccia: data_TX, data_RX, control, status.
- Multiplexer per selezionare se mettere nella porta di uscita il registro data_RX o il registro di controllo.
- Logiche con porte AND: address, address_mux in cui il segnale contenente l'indirizzo viene fatto passare bit a bit in delle porte and con un segnale di controllo di force. In questo modo vengono abilitati gli indirizzi solamente quando necessario accedere ai registri puntati dall'utente esterno.
- Logica combinatoria gate, che permette tramite un segnale di force, unito al segnale di indirizzo esterno, di generare in uscita i segnali nex_tx e read_rx atti ad indicare che si sta accedendo rispettivamente al registro data_tx per inserire un nuovo stato da trasmettere o la registro data_rx per leggere il dato ricevuto.

Per il registro di controllo si è stabilita la seguente convenzione: bit 0 per il trasmettitore e bit 1 per il ricevitore, dove il livello logico '1' indica l'abilitazione e lo '0' la disabilitazione. Non è stata assegnata alcuna azione al bit 2.

Ciò che si ottiene è il seguente schema:

- Registro di controllo "000" -> RX e TX disabilitati
- Registro di controllo "001" -> solo TX abilitato
- Registro di controllo "010" -> solo RX abilitato
- Registro di controllo "011" -> RX e TX abilitati

Nel caso in cui arrivi il segnale di disabilitazione con una procedura già avviata, essa viene portata a termine prima della disattivazione del componente.

L'abilitazione e indirizzamento dei registri presenta la seguente configurazione:

- "000" -> DISABLE;
- "001" -> ENABLE REG_STATUS;
- "010" -> ENABLE REG_CONTROL;
- "011" -> ENABLE RX_DATA;
- "100" -> ENABLE TX_DATA;

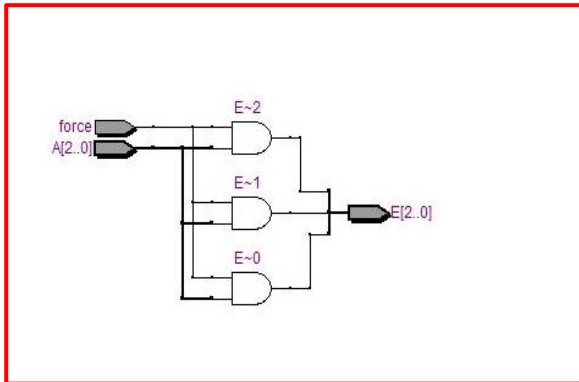


Figura 11: Logica combinatoria con porte AND

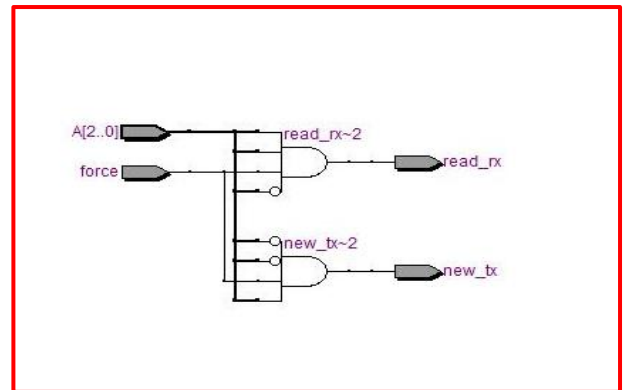


Figura 12: Logica combinatoria del Gate

Pallogramma Bus Interface

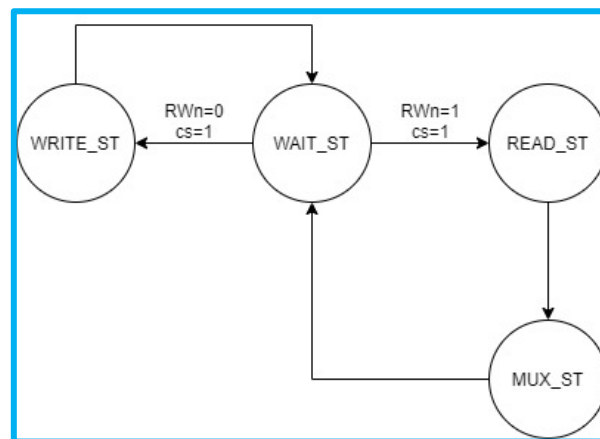


Figura 13: Pallogramma CU Bus_int

Timing Diagram Bus Interface

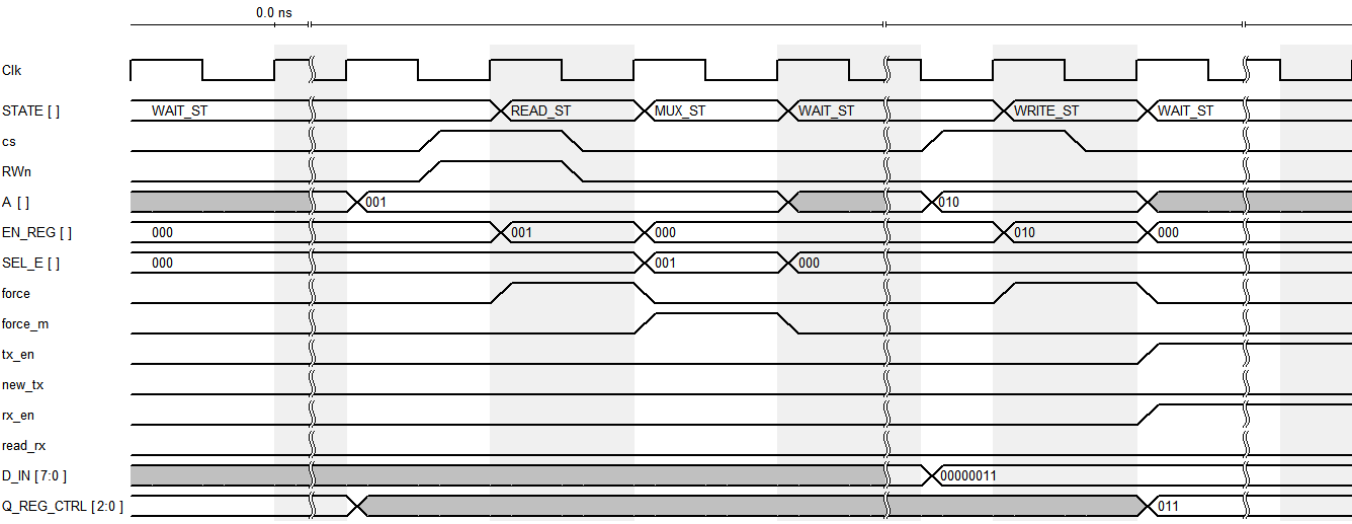


Figura 14: Timing Diagram blocco BUS INTERFACE

ATN

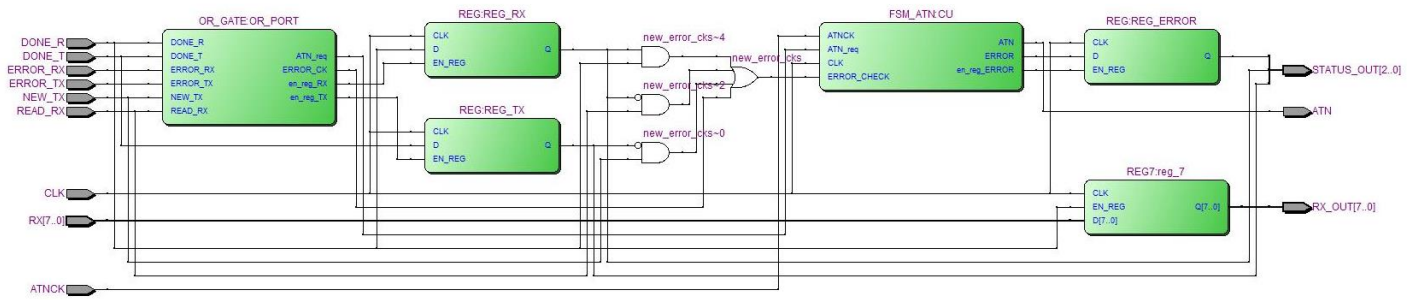


Figura 15: Data Path ATN

L'ultimo blocco realizzato è l'ATN, il quale è dedicato alla gestione delle richieste di interrupt.

I blocchi che lo compongono sono i seguenti:

- Control Unit, mostrata in figura 17;
- Reg7: registro aggiuntivo in cui viene salvato il dato Rx finale, per realizzare la struttura a doppio buffering ;
- 3 Flip-Flop: TX, RX, ERROR in cui vengono salvati le condizioni di stato di trasmettitore, ricevitore ed errore, in cui il bit memorizzato al valore logico 1 corrisponde rispettivamente a possibilità di inviare un nuovo dato da trasmettere, dato pronto in lettura oppure il verificarsi di un errore. L'abilitazione e il dato d'ingresso D del flipflop ERROR è gestita direttamente dai segnali di controllo della CU mentre per i flipflop TX e RX il segnale di ingresso è collegato rispettivamente ai segnali di DONE. Le uscite Q invece sono collegate agli ingressi del registro di Stato della bus Interface.
- Logica combinatoria per generare i segnali di enable dei registri precedentemente descritti e il segnale di richiesta di errore da inviare alla control unit. In particolare, i registri RX e TX vengono abilitati da una porta or tra i rispettivi segnali di DONE e i segnali read_rx e new_tx; ciò rende possibile l'abilitazione dei flip flop sia quando il dato è disponibile (oppure tx libero) sia quando l'utente esterno ha effettuato la lettura (oppure ha inviato un nuovo dato al tx).

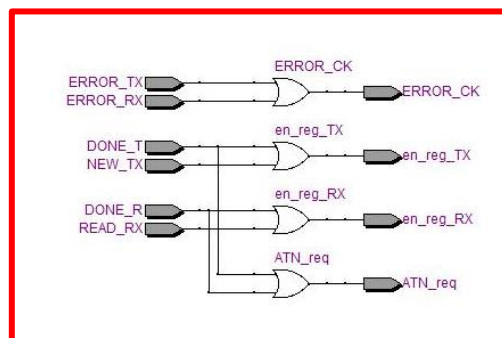


Figura 16: Logica del componente or gate

Pallogramma ATN

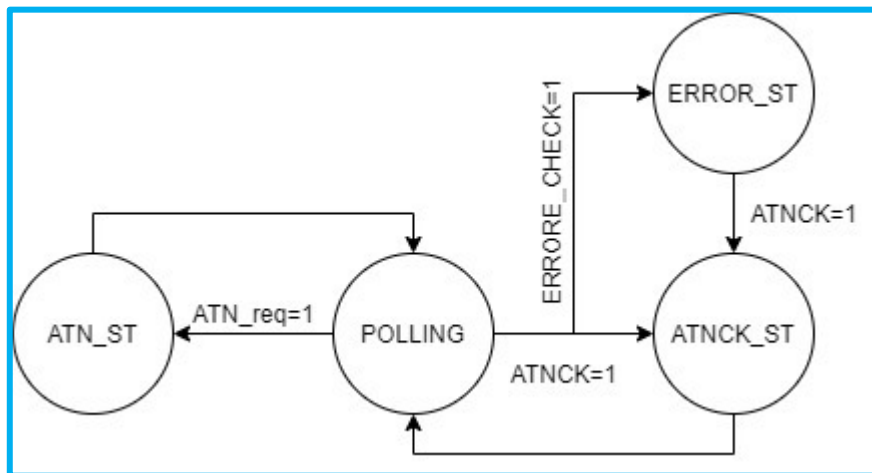


Figura 17: Pallogramma CU ATN

Per quanto riguarda la gestione errori il blocco ATN presenta uno stato `error_st` in cui viene imposto al valore logico '1' il terzo bit del registro di stato e indicato all'esterno dal valore ad 1 del segnale ATN; in questo caso la macchina rimane nello stato di errore finché l'utente esterno non prende in carico la gestione dell'anomalia rispondendo con il segnale `ATNCK = '1'`. Gli errori che possono portare a tale stato sono:

- Invio da parte dell'utente di un nuovo dato da trasmettere al registro `tx_data`, prima che quello mandato in precedenza venga caricato nel registro interno del trasmettitore. Ciò provocherebbe la sovrascrittura del dato precedente. (condizione data da `new_tx = 1` con `Q_ del FF TX = 0`)
- Mancata lettura da parte dell'utente dal dato in uscita dal ricevitore prima che questo venga sovrascritto da quello successivo. (condizione data da `DONE_RX = 1` con `Q_ del FF RX = 1`)
- Lettura di un dato del registro `dx_data` già letto in precedenza. (condizione data da `READ_RX = 1` con `Q_ del FF RX = 0`)
- Errori legati ai contatori aggiuntivi di trasmettitore e ricevitore che controllano che i rispettivi segnali di done arrivino dopo che sia trascorso un tempo ragionevole al completamento dell'operazione.

Nel caso in cui invece la richiesta di ATN (segnale `ATN_req`) arrivi dal completamento dell'operazione del trasmettitore o ricevitore, la CU tramite lo stato `ATN_st` impone al valore '1' il segnale esterno ATN, senza però rimanere in quello stato come invece accade per l'errore. Il segnale ATN verrà poi abbassato nello stato `ATNCK_st`, ovvero quando l'utente esterno tramite il segnale `ATNCK` comunica di aver preso in gestione la richiesta.

Timing Diagram ATN

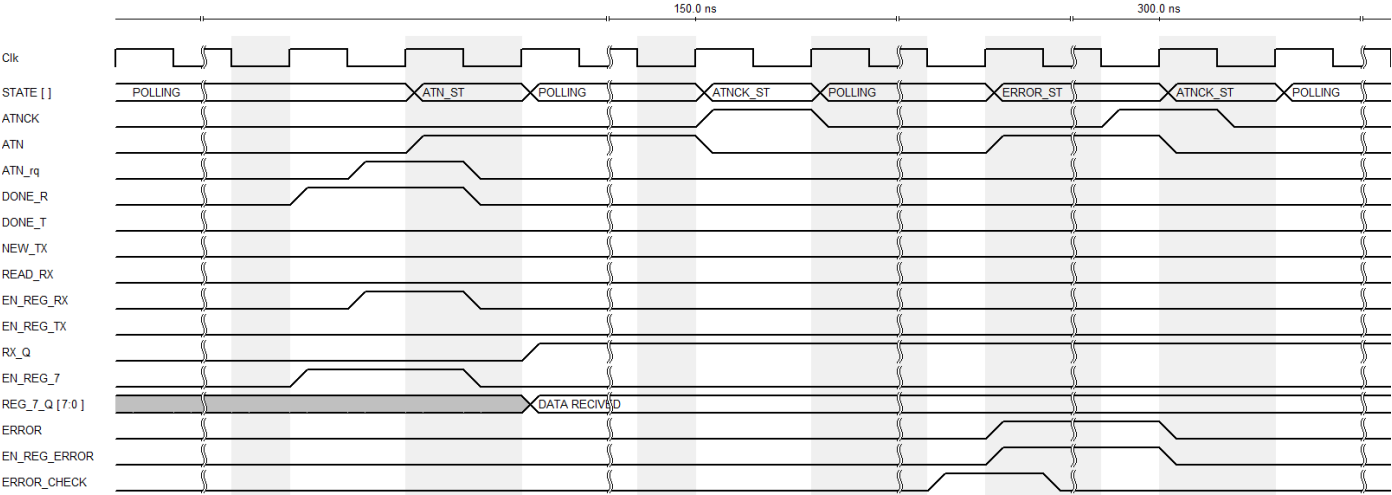


Figura 18: Timing diagram blocco ATN

Risultati Modelsim

Di seguito vengono riportati i risultati di due simulazioni riguardanti rispettivamente il funzionamento del trasmettitore e del ricevitore.

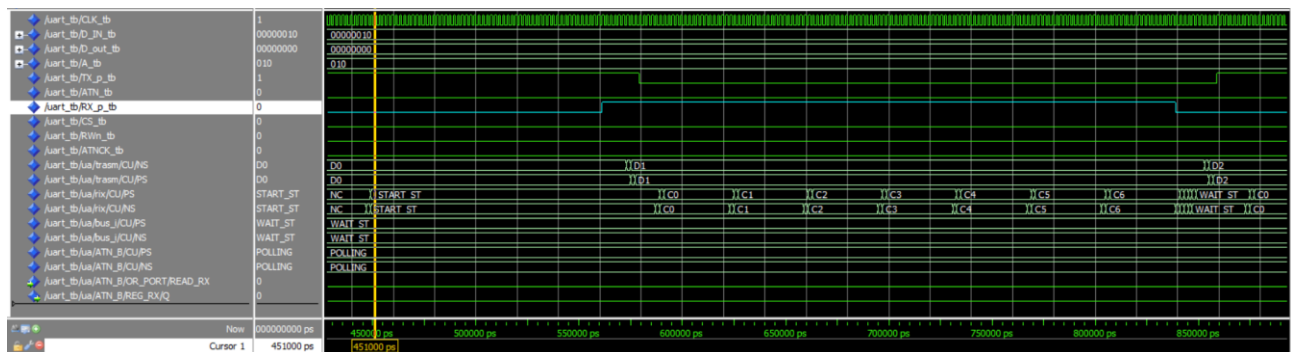


Figura 19: Campionamento di un simbolo

Per l'operazione di campionamento di un simbolo, mostrata in Figura 19, si è scelto di mandare al voter i campioni 2 4 6 in modo da compensare eventuali campioni errati come il primo e l'ultimo che rischiano di campionare il dato sbagliato.

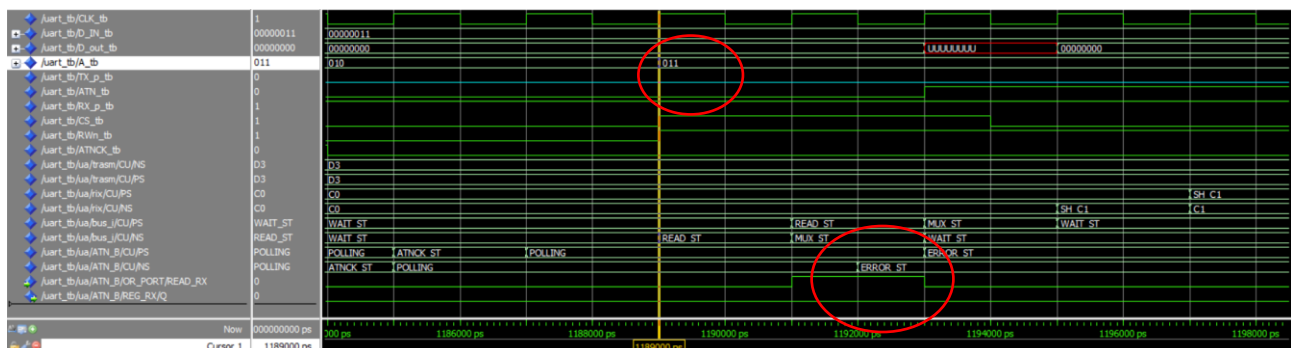


Figura 20: Errore di lettura

La figura 20 mostra un esempio di possibile errore ovvero il caso in cui l'utente chieda di leggere il dato del registro Rx_data (con A = 011) ma il valore contenuto nel registro non è valido REG_RX/Q è a '0' e non ad '1'.

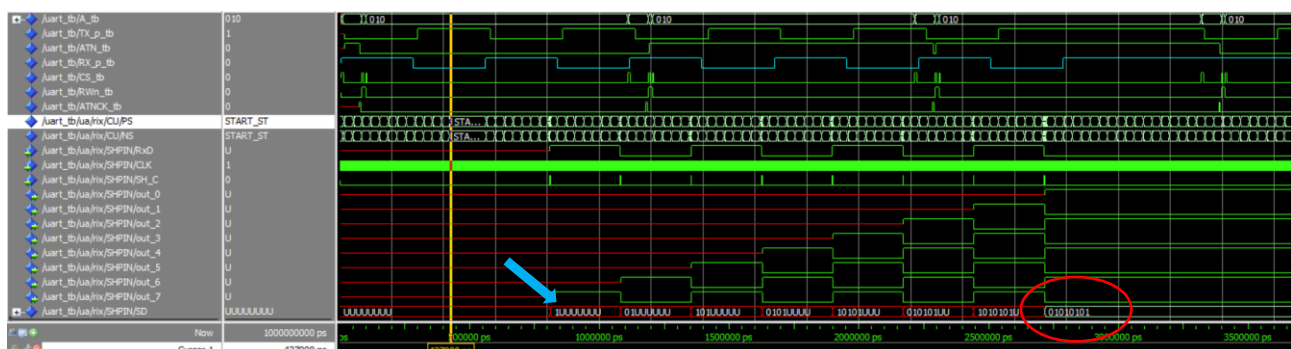


Figura 21: Ricevitore

In figura 21 è possibile vedere come lo shift register in uscita inserisca passo passo i valori dei simboli ricevuti, fino al completamento di tutti ed otto.

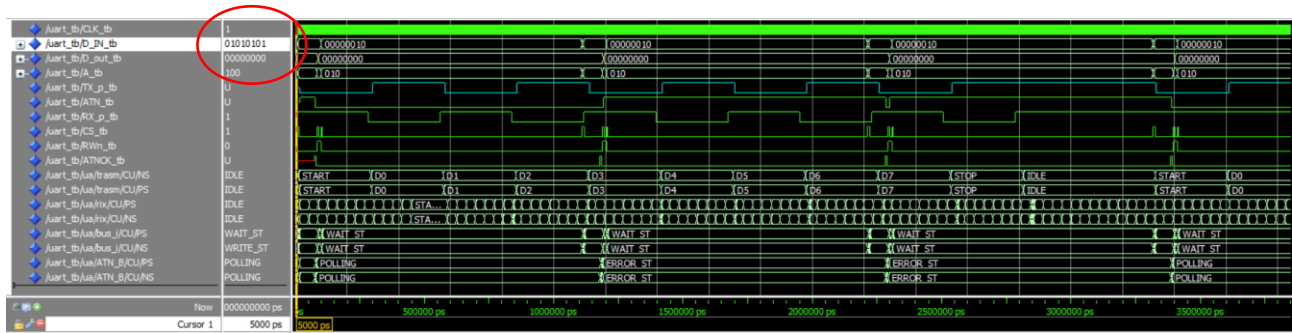


Figura 22: Trasmettitore

In figura 22 in blu è mostrata la porta seriale in uscita al trasmettitore che corrisponde al dato da trasmettere inviato dall'utente.