



Progettazione di un'interfaccia UART per comunicazione RS-232

Corso di Laurea Magistrale in Ingegneria Elettronica

Studenti

Amato Giovanni Luca Matr.267511

Cerbai Matilde Matr.274908

Chisciotti Laura Matr.274728

Goti Gianluca Matr.269825

Indice

1	Il protocollo RS-232	2
1.1	Caratteristiche generali	2
1.2	Descrizione del segnale	2
2	Struttura generale della UART	3
3	Trasmettitore	3
3.1	Struttura e caratteristiche principali	3
3.1.1	Data Path	4
3.2	Control Unit	5
3.3	Diagramma di stato	6
3.4	Timing Diagram	7
3.5	Simulazione su Modelsim	8
3.6	Test su board DE1	9
4	Ricevitore	9
4.1	Struttura e caratteristiche principali	10
4.2	Datapath	11
4.3	Timing Diagram	12
4.4	Diagramma di stato	14
4.5	Simulazione Modelsim	15
5	Uart	16
5.1	Simulazione su Modelsim	16
5.2	Test su Board DE1	16

1 Il protocollo RS-232

1.1 Caratteristiche generali

L'RS-232 è uno standard che permette lo scambio di informazioni a bassa velocità tra dispositivi digitali. Tale protocollo presenta tre caratteristiche principali:

- Seriale;
- Full-duplex;
- Asincrono;

Questo significa che è possibile trasmettere e ricevere contemporaneamente solo un bit alla volta e non viene scambiato alcun tipo di clock.

Nella sua forma più semplice, tale protocollo ha un supporto fisico costituito da due fili, "Tx" e "Rx", anche se nella realtà è più complesso, infatti presenta molte più connessioni dato che è stato progettato per comunicazioni a lunga distanza, dell'ordine di una decina di metri.

Per quanto concerne i livelli logici, non vengono usati dei valori di tensione comuni, motivo per cui tale protocollo è molto robusto, infatti si può avere fino a 9V di margine di rumore.

In uscita è presente:

- $+12\text{ V} = 0$ Logico " V_{OL} "
- $-12\text{ V} = 1$ Logico " V_{OH} "

In ingresso è possibile trovare fino a:

- $+3\text{ V} = V_{IL}$
- $-3\text{ V} = V_{IH}$

Naturalmente queste non sono tensioni logiche normali, esistono dei circuiti (MAX232) che trasformano tali livelli logici in TTL o CMOS compatibili e viceversa.

1.2 Descrizione del segnale

Lo standard trasmette un bit alla volta, ogni word da 8 bit è incapsulata (framing) con un bit di "START" (0 logico) e un bit di "STOP" (1 logico), il livello "neutro" è l'"IDLE" (1 logico). In generale le combinazioni possono essere molteplici, deve essere presente un accordo esplicito sulla lunghezza della word trasmessa, il bit di stop può durare per uno o più tempi di bit, inoltre ci può essere parità o meno.

Nel caso sotto esame il protocollo prevedeva:

- 8 bit (word)
- No parity
- 1 bit di STOP

Quando il sistema è in quiete si trova nello stato di "IDLE", che equivale all' "1" logico. Quando l'utente vuole iniziare una trasmissione ci deve essere sempre una transizione, infatti da "IDLE" si passa a "START" ottenendo così una transizione dall' "1" logico allo "0" logico. Il bit di "STOP" è importante in quanto se l'utente vuole effettuare una trasmissione continua, la transizione "1" logico e "0" logico è sempre garantita, poichè lo "STOP" riporta a livello alto la linea garantendo così la corretta transizione.

Il timing diagram di una generica trasmissione RS-232 si presenta come in Figura 1

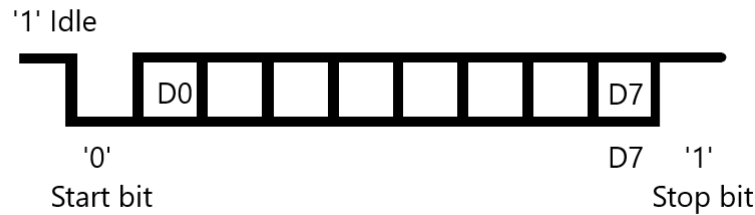


Figura 1: Timing diagram di una generica trasmissione RS-232

Come è possibile notare nella figura e come già detto in precedenza, il sistema passa da uno stato di quiete, ovvero "IDLE", al bit di "START", eseguendo così la transizione che farà capire al ricevitore che è stata iniziata una nuova trasmissione. Infine il sistema passa in "STOP" e dopodichè (se non vi è una nuova trasmissione) ritorna in "IDLE".

2 Struttura generale della UART

In Figura 2 è stata riportata la struttura della UART progettata.

L'obiettivo è stato quello di realizzare un'unità di ricezione e di trasmissione seriale di dati secondo il protocollo RS-232 precedentemente introdotto.

Il trasmettitore e il ricevitore sono stati realizzati indipendentemente, ma in modo tale da riuscire ad interfacciarsi nel rispetto delle specifiche a disposizione.

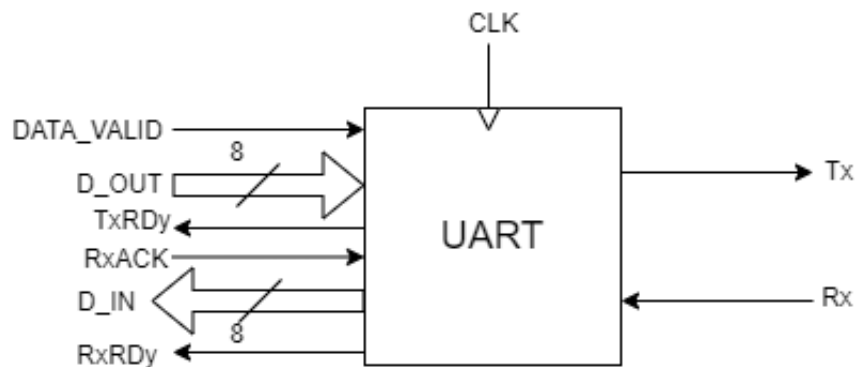


Figura 2: UART

3 Trasmettitore

3.1 Struttura e caratteristiche principali

In riferimento alla Figura 2, sono osservabili i principali segnali che arrivano e che partono dalla UART e che interessano il trasmettitore.

- **D_OUT**: bus da 8 bit, che carica in modo parallelo il dato che deve essere trasmesso;
- **Data_Valid**: segnale in ingresso, che dice al trasmettitore quando campionare il dato da trasmettere;
- **TX_RDY**: segnale di stato del trasmettitore, comunica all'esterno se il sistema è in trasmissione (busy) o meno;
- **TX**: uscita seriale del trasmettitore.

Quando il trasmettitore non lavora, il TX_RDY è tenuto alto, mentre quando è in trasmissione tale segnale è basso (busy).

Il funzionamento di tale trasmettitore è abbastanza semplice:

- L'utilizzatore esterno, per esempio un μC , manda la word che vuole trasmettere al bus D_OUT;
- Dopo aver mandato il dato da trasmettere (e mantenendolo per il tempo necessario) asserisce il Data_Valid, in questo modo il trasmettitore campiona i dati che dovrà trasmettere;
- Il trasmettitore acquisisce i dati e li trasmette seguendo la logica del protocollo e pone al livello logico basso la linea TX_RDY;
- Dopo che la trasmissione è finita, il TX_RDY è posto a livello logico alto e il sistema è pronto per una nuova trasmissione.

Il sistema lavora con un clock da 25 MHz, la trasmissione seriale è a 9600 baud e ciò significa che il tempo di bit è circa $104 \mu s$. Perciò, una trasmissione completa, che comprende quindi sia il bit di "START" e di "STOP", ha una durata di circa $1040 \mu s$.

3.1.1 Data Path

Il datapath del trasmettitore è mostrato in Figura 3.

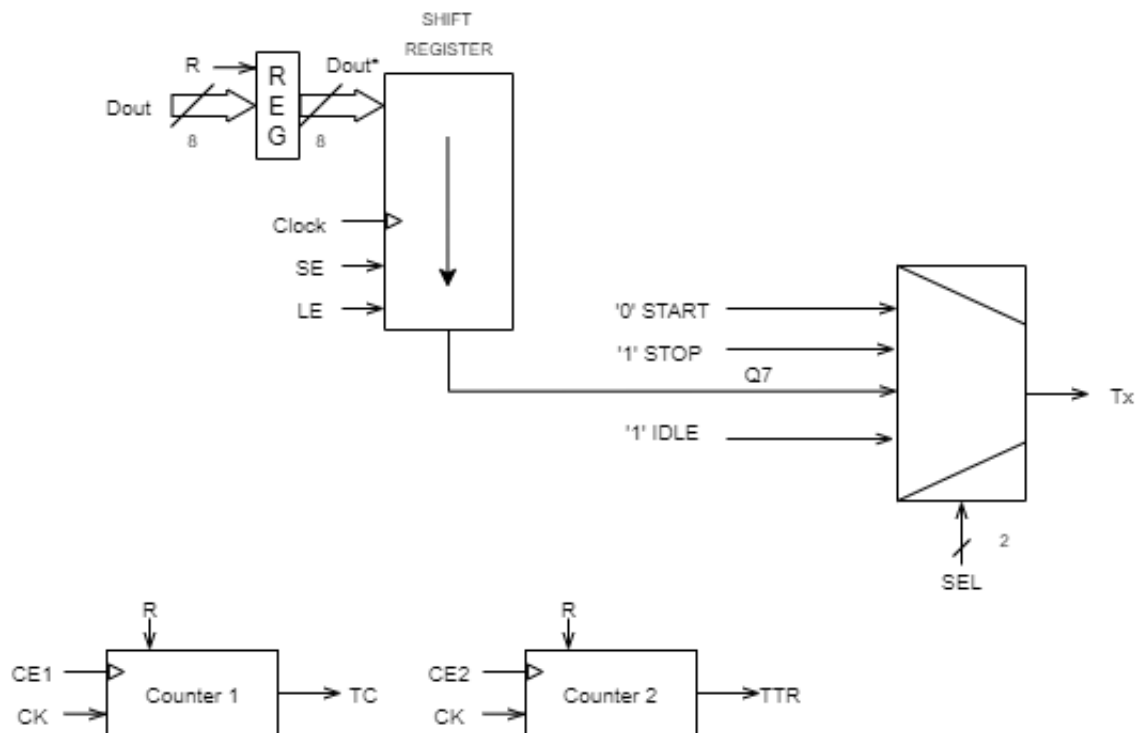


Figura 3: Datapath del trasmettitore

Lo shift register è il cuore di questo sistema, infatti è tale componente che esegue la "serializzazione" del dato, in particolare questo ha un caricamento di tipo parallelo, una porta che abilita lo shift "SE" (shift enable) e una porta che abilita il load "LE" (load enable).

Idealmente appena il sistema avverte il Data_Valid, lo shift register dovrebbe caricare i dati all'interno, questo però comporterebbe la creazione di una macchina di Mealy, in quanto un segnale esterno provoca un'immediata reazione nella macchina.

Ovviamente si vuole evitare in ogni modo tale situazione, infatti la soluzione adottata è la macchina di Moore, per questo il sistema risponde al Data Valid con un comando di "load" dello Shift register al colpo di clock successivo. Risulta quindi immediatamente evidente la necessità di un registro che permetta di ritardare del tempo necessario, in questo caso un colpo di clock, per dare la possibilità

al sistema di rispondere correttamente e quindi di campionare il dato, caricandolo così all'interno dello shift register. La temporizzazione della trasmissione è accertata da due contatori, il "Counter 1", che assicura la corretta temporizzazione tra la trasmissione di un bit e l'altro, ed il "Counter 2", che conta il numero di bit trasmessi durante la comunicazione, decretandone in seguito la fine. Entrambi i contatori hanno un terminal count che viene mandato alla Control Unit (CU), la quale risponde al colpo di clock successivo col comando corretto (può per esempio abilitare lo shift dello shift register oppure girare il mux).

3.2 Control Unit

La Control Unit (CU) utilizzata per questo progetto è costituita da una parte sequenziale che fa progredire gli stati, occupandosi quindi anche di eventuali salti e una parte puramente combinatoria che manda i comandi al data path. In Figura 4 è presente uno schema di principio della CU utilizzata.

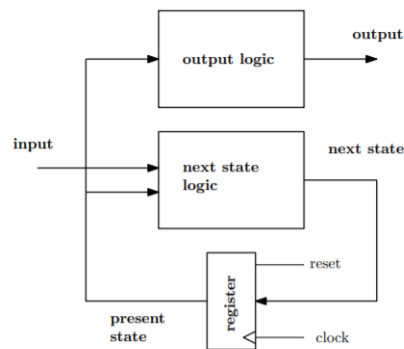


Figura 4: FSM a due processi

Come si può notare dalla figura, gli input entrano nella "next state logic", ma sono filtrati da un registro, che permette di svincolarsi dall'esterno, evitando di creare una macchina di Mealy. Nel progetto, ogni volta che un segnale arriva dal data path ed entra nella CU, provoca una risposta di questa con un adeguato comando solamente al colpo di clock successivo.

3.3 Diagramma di stato

In Figura 5 è riportato il diagramma di stato del trasmettitore, dove per ragioni di chiarezza e pulizia dello schema non sono state incluse tutte le connessioni che portano allo stato di "RESET", attivate con la condizione "RST=1".

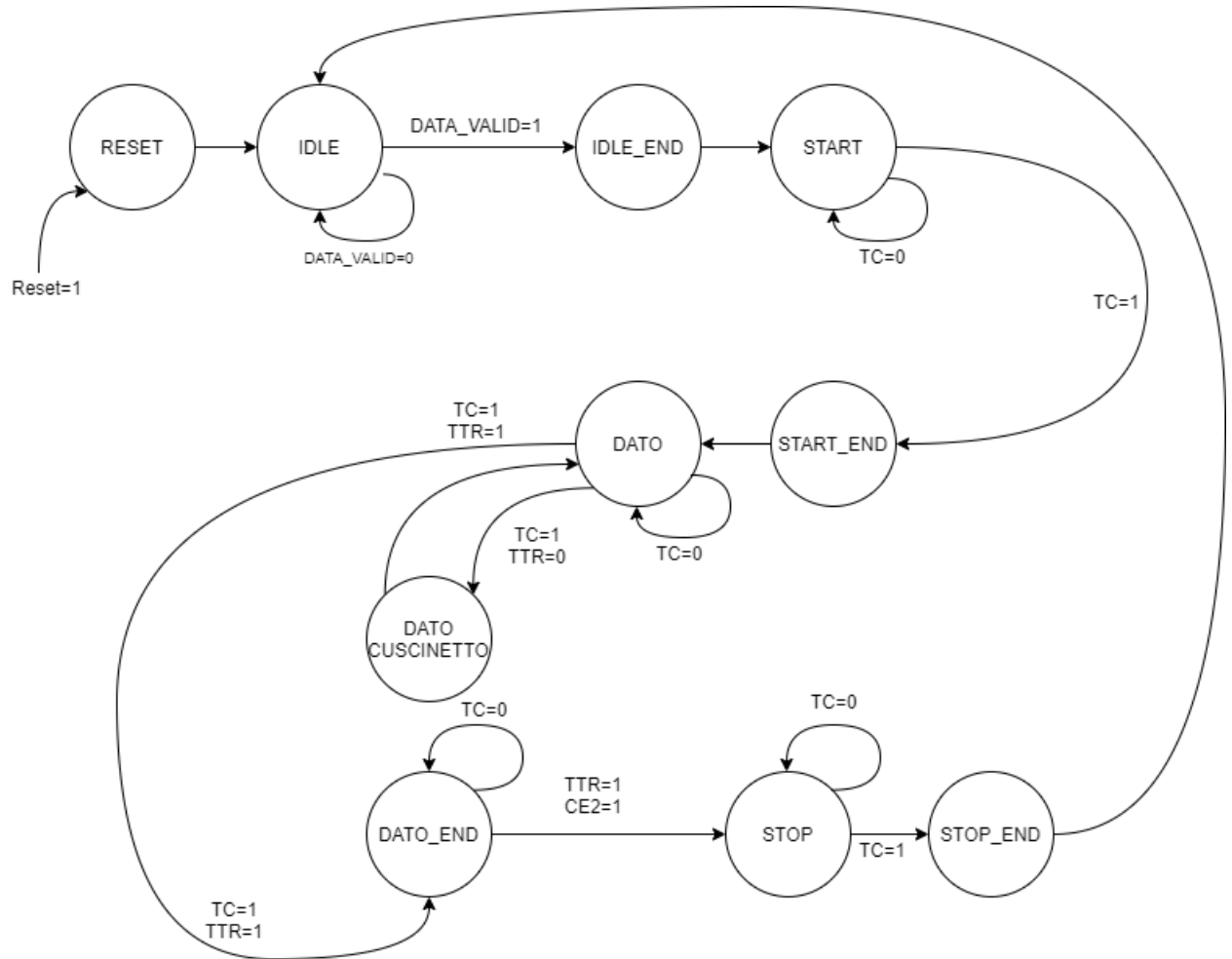


Figura 5: Diagramma di stato del trasmettitore

Come è possibile notare dal diagramma, il sistema rimane in stato di "IDLE" finché non riceve il segnale "DATA_VALID=1", in seguito, dopo essere transitati dallo stato "IDLE_END" il sistema approda in "START" dove, con la progressione allo stato successivo, verranno dati i comandi per inviare il bit di start. In tale stato vi può essere una progressione solamente se il segnale "TC=1", cioè il terminal count del contatore che garantisce il tempo di bit. Se tale condizione non è verificata il sistema rimane nello stato attuale. Nello stato "DATO" il sistema resta in attesa della condizione "TC=1", che permette di saltare in "DATO_CUSCINETTO" e di mandare quindi in uscita i comandi necessari allo shift register per far uscire i dati in sequenza (in questo caso è stato creato un loop per inviare tutti gli 8 bit della trasmissione). Dopodiché il sistema torna nello stato "DATO", dove vengono abbassati tutti i comandi relativi all'invio dei dati. Inoltre il dispositivo contiene un contatore, che conta i bit di dato trasmessi ed il cui terminal count ("TTR") viene alzato quando conta 7. Tale segnale è stato sfruttato per generare una condizione di salto insieme alla condizione "TC=1", quindi, quando questa è verificata, si salta nello stato che manda in uscita l'ultimo bit e che permette di preparare il sistema ad inviare il bit di stop, una volta giunti nello stato "STOP_END", la CU gira il mux per inviare il bit di "IDLE" e in seguito il sistema torna in quiete nello stato di "IDLE", in attesa di una nuova trasmissione e quindi di un nuovo "DATA_VALID".

3.4 Timing Diagram

In Figura 17 è riportato il timing del trasmettitore, di cui sono state analizzate alcune sue parti in riferimento ai segnali interessati:

- **Clk**: clock a 25 MHz;
- **Dout**: word da trasmettere;
- **Data valid**: segnale che identifica quando il dato può essere campionato;
- **SE**: shift enable dello shift register;
- **TxRDy**: segnale che identifica quando la UART può trasmettere;
- **LE**: load enable dello shift register;
- **Dout***: Dout ritardato di un colpo di clock;
- **SEL**: select del mux;
- **CE1**: counter enable del COUNTER 1, che fornisce il tempo di bit;
- **TC**: terminal count del contatore 1;
- **Tx**: uscita seriale del trasmettitore;
- **CE2**: counter enable del COUNTER 2;
- **TTR**: terminal transmission del contatore 2;
- **Reset**: reset del sistema;

Il flow del timing diagram è il seguente:

1. Dout è il dato da trasmettere, l'utilizzatore, per esempio un PC o un μC , pone la word sul bus;
2. l'utilizzatore invia il Data_Valid quando il dato da trasmettere è stabile. Tale segnale comunica alla UART che può campionare il dato, caricandolo così istantaneamente i dati nello shift register. Questa situazione, però, ci riporterebbe ad una macchina di Mealy, caso da evitare;
3. a tal proposito, il segnale da trasmettere è ritardato di un colpo di clock ed è indidato con Dout*. Questo permette anche di svincolarsi dal mondo esterno;
4. il load enable (LE), per creare una macchina di Moore, è attivato al colpo di clock successivo al Data_Valid, permettendo così di caricare i dati all'interno dello shift register;
5. nel momento in cui il dato viene campionato, lo shift enable dello shift register (SR) è disabilitato, evitando così di shiftare il dato appena caricato;
6. TxRDy passa quindi da livello alto a basso, comunicando che è in corso una trasmissione (stato "Busy");
7. nello stesso istante, il contatore 1 è abilitato per contare il tempo necessario per trasmettere un bit. Il segnale terminal count (TC) viene attivato tutte le volte che sarà passato tale lasso di tempo (1 bit trasmesso);
8. l'uscita dello shift register porta il dato shiftato in ingresso al multiplex;
9. il comando SEL permette di girare il mux, secondo codifica, per trasmettere IDLE (00);
10. Tx, che rappresenta l'uscita del mux, passa quindi dallo stato IDLE allo START al colpo di clock successivo alla ricezione del TC da parte della control unit;

11. Dopo la trasmissione del bit di START il sistema è in quiete finché il TC non diventa alto. Al colpo di clock successivo la control unit invia i comandi necessari per lo shift del dato in uscita (SE) e l'abilitazione del contatore 2 (CE2), che conta il numero di bit della word che sono stati trasmessi;
12. il segnale Terminal transmission (TTR) indica la fine della trasmissione della word e sarà attivo dal momento in cui viene contato il passaggio dell'ottavo bit a quando il mux non viene girato per trasmettere STOP;
13. Per completezza è stata anche inserita la routine di reset, la quale disabilita i counter enable, il load enable e lo shift enable dello shift register, pone il TX_RDY in modalità "Busy" e pone il mux in posizione "00" così da trasmettere lo stato di "IDLE" in uscita.

3.5 Simulazione su Modelsim

Per valutare in prima fase il corretto funzionamento del trasmettitore, si è sfruttato il software Modelsim per effettuare svariate simulazioni con word di 8 bit scelte arbitrariamente, si riporta di seguito, in Figura 6 , a titolo di esempio una simulazione effettuata.

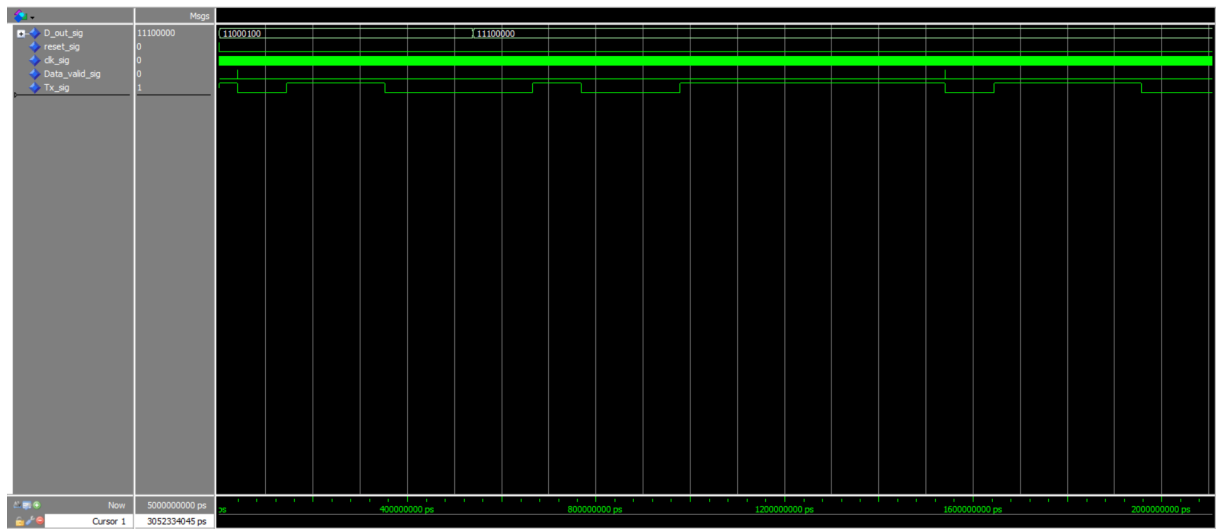


Figura 6: Simulazione di una trasmissione

Dalla simulazione è possibile osservare che il sistema inizialmente si trova in "IDLE" (linea Tx alta), dopo aver applicato il "DATA __VALID" il sistema invia il bit di start (linea bassa) e poi inizia la trasmissione partendo dall'LSB della word. Alla fine della trasmissione il sistema manda il bit di stop (linea alta) e torna in "IDLE", in attesa di una nuova word da trasmettere, ciò è da poter notare dalla simulazione, in quanto la linea TX rimane alta in attesa di un nuovo "DATA __VALID".

3.6 Test su board DE1

In seguito sono stati effettuati dei test sulla Board Altera DE1 presente in laboratorio, si è reso quindi necessario la progettazione di un testbench "hardware" per verificare il funzionamento del trasmettitore attraverso l'oscilloscopio. Lo schema del testbench progettato è mostrato in Figura 7.

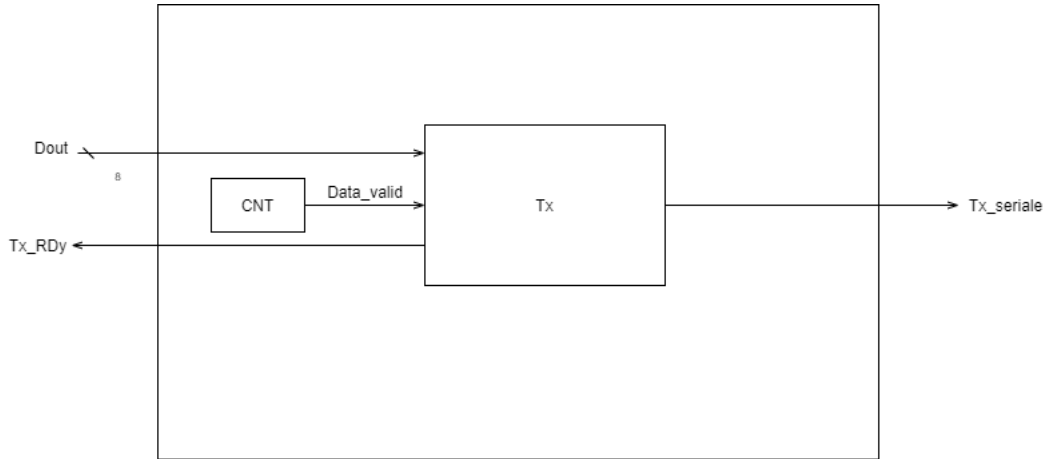


Figura 7: Schema test HW per il trasmettitore

Per valutare il corretto funzionamento del trasmettitore, è stata sviluppata una top-level entity contenente il trasmettitore e un contatore il cui terminal count è stato collegato al Data Valid. Il valore del contatore è stato settato in modo da inviare un DATA_VALID ogni 1,25 s così da permettere la visualizzazione della trasmissione sull'oscilloscopio. Il bus Dout è stato collegato agli switch della board DE1, così da variare il dato trasmesso ed il risultato è stato verificato attraverso l'oscilloscopio. Di seguito in Figura 8 è riportato, a titolo di esempio, un test effettuato in laboratorio dove è stata trasmessa la word "01100100", in cui il primo bit a sinistra rappresenta l'LSB.

Come impostazioni sono stati posti $100\mu\text{s}/\text{DIV}$, in modo tale da poter vedere circa un bit per divisione, dato che ciascun bit ha la durata di $104\mu\text{s}$. Inoltre sull'oscilloscopio è osservabile a sinistra, prima dell'LSB, il bit di IDLE (1) ed il bit di START(0), ed a destra, successivo all'MSB, il bit di STOP (1). Successivo al bit di STOP, è possibile osservare che ci sono ancora qualche altro μs a livello logico alto, questi rappresentano una frazione dell'IDLE, facente parte della seconda word trasmessa.

4 Ricevitore

In tale sezione è stato analizzato il blocco di ricezione della UART realizzata secondo le specifiche di progetto, il quale, per attuare al meglio le metodologie apprese, è stato idealmente considerato indipendente dal trasmettitore, assumendo quindi un segnale di clock differente.

Come è stato precedentemente detto, il Tx lavora a 9600 baud, ma non disponendo di un riferimento temporale è stato considerato che il bit di START potesse arrivare in qualsiasi momento, com'è ragionevole aspettarsi in una comunicazione reale. Per gestire tale situazione è stato deciso di sovracampionare ciascun bit di un fattore 8, andando ad aumentare il baud rate (9600×8), in modo tale da riuscire a capire quando effettivamente il bit ricevuto fosse quello di START e dove avvenisse la transizione da IDLE, cioè livello logico alto, a START, corrispondente a livello logico basso.

Come verifica della reale avvenuta transizione, è stato posto di rilevare questa solo nel momento in cui sono presenti nel primo shift register quattro 1, riferiti all'IDLE, e quattro 0, relativi allo START. Tale condizione è stata verificata tramite l'utilizzo di una semplice porta AND, che pone l'uscita a 1 se e solo se è verificata la condizione precedentemente esposta.

Finché tale condizione è vera l'AND mantiene alto il segnale di "START TRANSITION", che indica la transizione da livello logico alto a basso. Questo si abbasserà successivamente con l'arrivo del quinto bit di START.

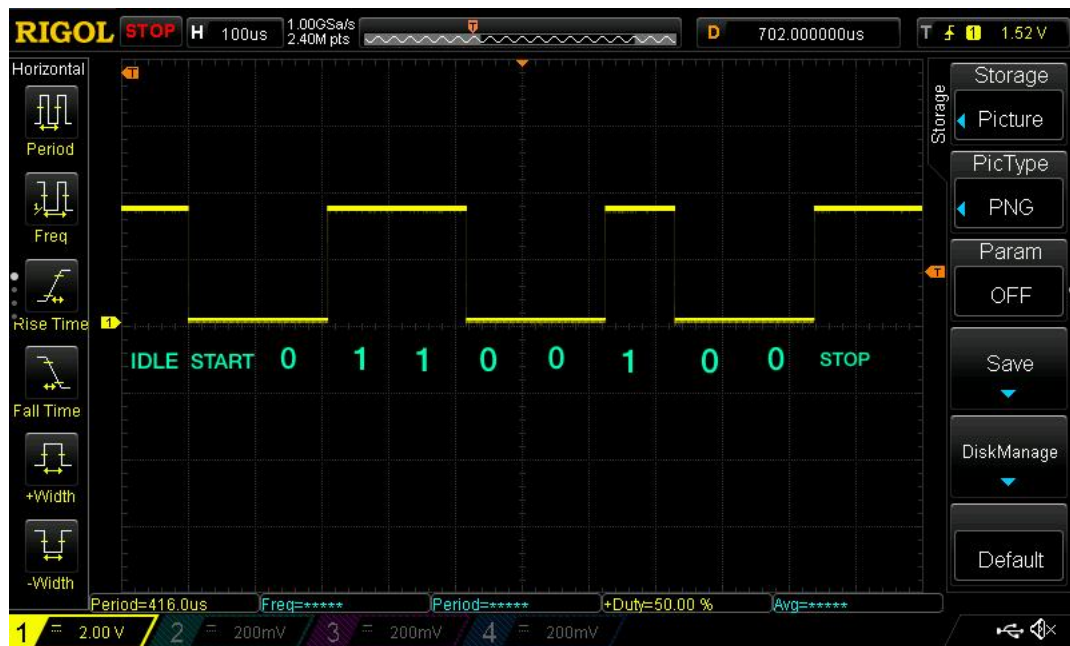


Figura 8: Test su oscilloscopio

Per esplicitare meglio il concetto è stato riportato il disegno in Figura 9.

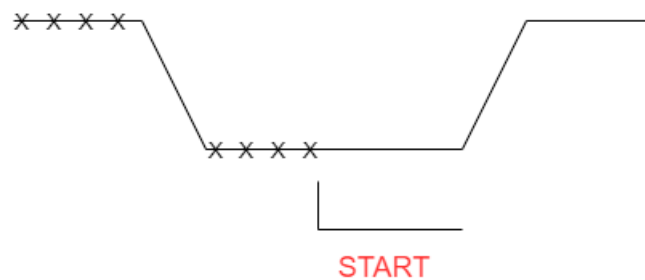


Figura 9: START Transition

4.1 Struttura e caratteristiche principali

In riferimento alla Figura 2, sono stati considerati i principali segnali che interessano il ricevitore:

- **D_IN**: bus da 8 bit, che presenta in parallelo il dato ricevuto in modo seriale;
- **RX**: segnale di ingresso seriale del ricevitore;
- **RX_RDY**: tale segnale è messo alto o basso dal ricevitore stesso, se è alto vuol dire che il sistema è in ricezione e non può ricevere una nuova word;
- **RX_ACK**: se asserito indica che la word ricevuta è stata acquisita dal sistema utilizzatore;

4.2 Datapath

Nella figura sottostante è stato riportato il datapath del ricevitore:

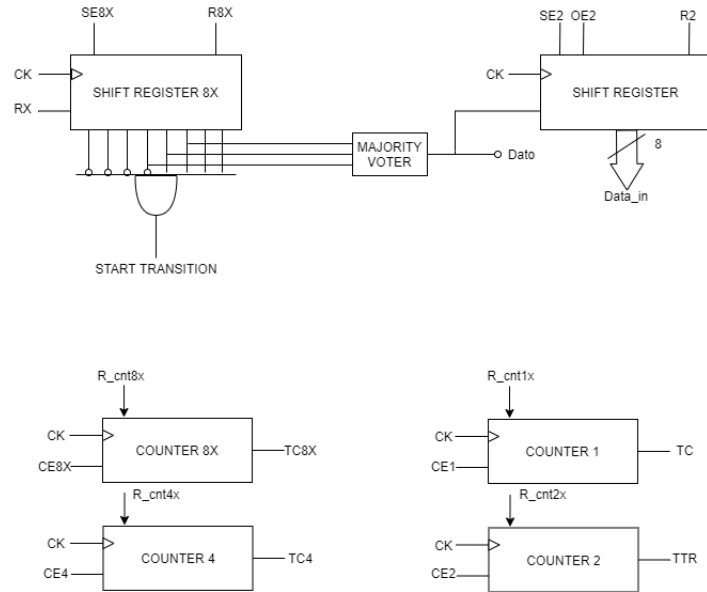


Figura 10: Datapath ricevitore

Data la word di 8 bit di cui è composto ogni dato, è stato considerato uno shift register 8x, che risulta essere sempre attivo in modo tale da shiftare sempre i bit e effettuare il sovracampionamento. Il majority voter (MV): permette di prendere il valore al centro del bit. Tale blocco considera più campioni adiacenti, nel caso specifico 3 bit per eliminare il minore e prendere il valore che compare più volte nei 3 bit scelti, cioè 0 o 1. E' stato deciso di prendere i bit 2,3 e 4 dello shift register in quanto, dopo altri quattro shift a partire dall'individuazione dell'inizio di una nuova trasmissione, lo shift register risulta essere "riempito" da tutti gli 8 campioni di un generico bit della word e quelle specifiche posizioni risultano essere in posizione centrale (in termini temporali) al bit.

Come già spiegato, l'individuazione dello start transition non serve solo a capire quando avviene una nuova trasmissione, ma serve anche a posizionarsi al centro del bit. Infatti dopo altri quattro shift dello shift register, che opera il sovracampionamento, questo sarà riempito da tutti gli 8 campioni del bit di start, questo rende possibile l'individuazione del centro del bit, posizionando in modo opportuno il majority voter. Trascorso un tempo di bit, cioè 8 campionamenti a una velocità 8x, all'interno del majority voter "entra" il nuovo centro del bit della word.

L'uscita del MV, entrerà nel secondo shift register, che viene attivato tramite il comando OE2 solo ogni 8 shift del primo Shift Register, una volta che viene riempito di tutti gli 8 bit della word, il comando di output enable viene abilitato e la word è mandata in uscita sul bus DATA_in.

Si hanno quattro contatori che garantiscono l'adeguata temporizzazione di segnali che interessano la parte di ricezione:

- counter8x: conta 324 colpi di clock ed assicura il sovracampionamento 8X,
- counter 1: conta 2603 colpi di clock, assicurando il tempo di bit (9600 baud),
- counter 2: conta i bit della word ricevuti,
- counter 4: conta 4 shift, permettendo così l'individuazione del centro del bit.

4.3 Timing Diagram

Il timing relativo all'operazione di ricezione è stato riportato in Figura 18, la quale è situata alla fine di tale relazione.

I segnali interessati nell'operazione di ricezione sono i seguenti:

- **Clk**: clock a 25 MHz;
- **D_IN**: dato trasmesso. E' l'uscita del secondo shift register;
- **Reset**: quando RST=1 il reset riporta il sistema di elaborazione allo stato iniziale;
- **Rx_RDY**: segnale con cui la UART comunica al μ processore che il secondo shift register è pronto per mandare in uscita D_IN;
- **Ack**: indica 'acknowledge' e si attiva quando il μ processore ha ricevuto tutto il dato. E' il μ processore stesso che manda alla UART tale segnale;
- **Rx**: bit trasmessi;
- **TC8X**: rappresenta il terminal count (uscita) del contatore, che conta 325 colpi di clock;
- **SE8x**: è lo shift enable del contatore COUNTER 8x
- **STR**: indica la start transmission e si ha quando è stata identificata una transizione di START;
- **CE8x**: è il counter enable del contatore 8x;
- **TC4x**: è il terminal count del contatore 4x, che conta 4 shift;
- **CE4**: counter enable del contatore 4x
- **CE1**: counter enable del COUNTER 1, che conta 2603 colpi di clock;
- **TC**: terminal counter del COUNTER 1, viene inviato quando termina di contare;
- **SE2**: shift enable del secondo shift register;
- **CE2**: counter enable del COUNTER 2, che conta i bit ricevuti;
- **TTR**: terminal transmission, è l'uscita del COUNTER 2, che viene inviata quando termina di contare;
- **OE2**: output enable del secondo shift register;

Di seguito sono riportati alcuni chiarimenti per comprendere più facilmente il timing:

1. dato che la trasmissione è continua, Rx parte come attivo alto ("IDLE");
2. l'asserimento di TC8X permette l'attivazione di SE8X. Questi due sono asseriti per un solo colpo di clock. Ciò consente di sovracampionare il dato in ricezione ad una velocità 8x. Tale processo è continuo ed è necessario in quanto la ricezione di una nuova word non può essere predetta, quindi non è possibile sapere temporalmente quando lo START, mandato dal trasmettitore, sarà ricevuto;
3. quando nel primo shift register sono presenti quattro 0 (dello start) e quattro 1 (dell'idle), c'è la sicurezza di ricevere una transizione di inizio trasmissione ("START"), quindi a livello di timing avviene l'asserimento del segnale STR. Ciò viene verificato in concomitanza con l'abbassarsi dello shift enable (SE8X) del quarto zero dello start.
Il segnale STR, dato che è prodotto da una porta AND, rimane alto finché lo shift register non fa scorrere un nuovo dato al suo interno;
4. CE4X viene asserito un colpo di clock dopo che STR è stato alzato.
CE4X abilita il COUNTER 4, che deve contare 4 shift (del sovracampionatore) in modo tale da far entrare tutti gli 8 bit di start e quindi di centrare questo rispetto al VOTER;

5. Durante il processo di ricerca della transizione dello start, SE2 non è asserito, quindi il secondo shift register è disabilitato in quanto è stato scelto di estrapolare la word dal frame ricevuto;
6. TC4X indica il termine dello shift degli ultimi 4 zeri dello start e viene asserito dal contatore in concomitanza del TC8X dell'ultimo bit di start (questo perchè i conteggi stanno in un rapporto 1:4), per essere poi abbassato al colpo di clock successivo;
7. in contemporanea all'asserimento dello shift enable riferito all'ultimo bit dello start, viene disabilitato il CE4X precedentemente attivato e viene asserito il CE1, il quale fa partire il conteggio di 2603 colpi di clock, che assicura il tempo di bit;
8. Dopo lo shift dell'ultimo bit di start, iniziano ad essere ricevuti i bit del dato trasmesso. Per tali bit va fatto riferimento ai segnali TC e SE2, che indicano quando entrano i bit del dato trasmesso nel secondo shift register.
In questo momento ci sono contemporaneamente uno shiftaggio nel primo shift register ed uno nel secondo shift register, dato che il COUNTER 1 è abilitato a contare dall'asserimento del CE1 e questo porta ad introdurre all'interno del secondo shift register i singoli bit valutati dal Majority Voter dopo il sovraccampionamento;
9. Una volta che il secondo shift register è riempito dagli 8 bit della word trasmessa, il CE2, che era stato alzato in concomitanza dello shift enable (SE2) del primo dato trasmesso, va basso, poichè sono stati ricevuti tutti i bit necessari;
10. Contemporaneamente all'abbassarsi del CE2, il segnale TTR va alto, per poi abbassarsi al colpo di clock successivo. Questo indica che la trasmissione è terminata e che sono stati ricevuti tutti i bit della word (8 bit);
11. La word viene mandata in uscita sul bus e l'Output enable del secondo shift register è asserito, il sistema inoltre pone la linea RX_Rdy a livello logico alto, indicando così che un dato ricevuto è disponibile;
12. Una volta che il sistema utilizzatore ha campionato il dato, fornisce il segnale Ack per un unico colpo di clock, al colpo successivo l' RX_Rdy e OE2 sono disattivati ed il sistema rimane in attesa di una nuova ricezione;
13. Se viene mandato il Reset alto, il cui asserimento dura un unico colpo di clock, tutte le impostazioni vengono resettate, riportando il sistema alle condizioni iniziali. Tutti i segnali di enable dei contatori e degli shift register sono disabilitati, oltre al reset dei componenti stessi, Inoltre è disattivato anche il counter enable, che permette di sovraccampionare.

4.4 Diagramma di stato

Nella figura sotto è riportato il pallogramma del ricevitore.

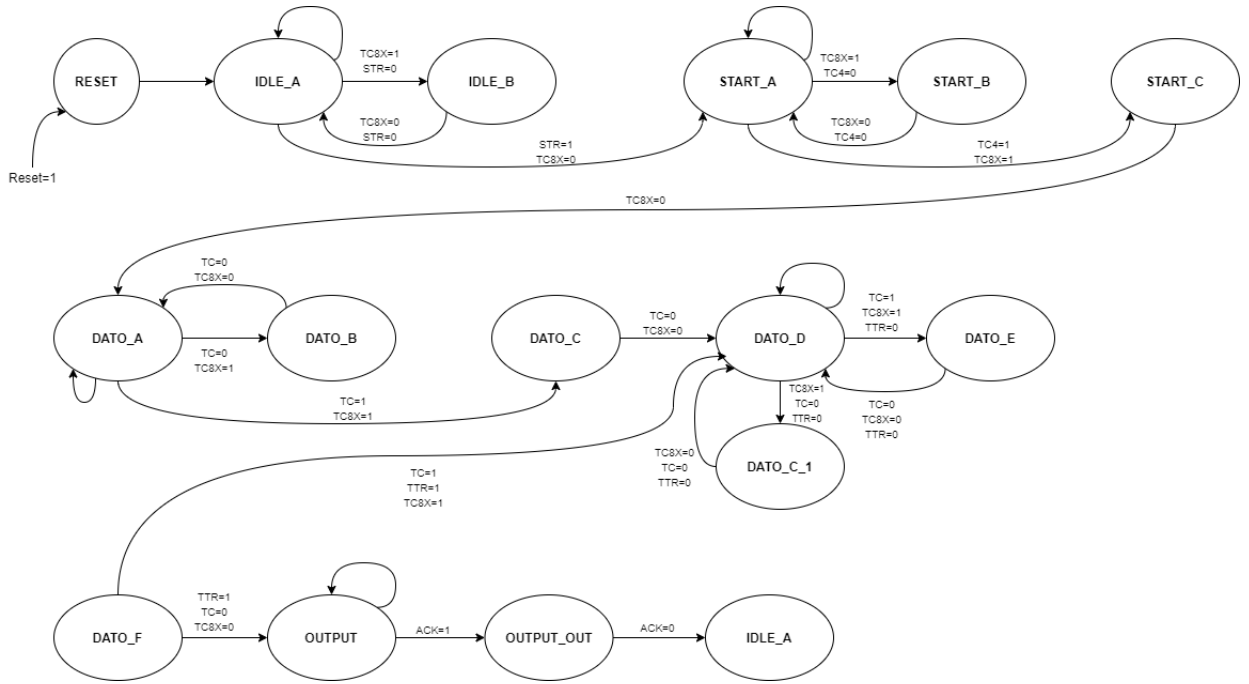


Figura 11: Palleogramma Rx

In tale diagramma di stato, com'è stato fatto anche nella sezione precedente per il trasmettitore, per aiutare la comprensione dello schema ed evitare di intaccare la chiarezza, non sono state incluse tutte le connessioni da ogni stato a quello di "RESET" con la condizione RST=1.

Quindi partendo dallo stato di "IDLE_A", il sistema rimane lì finché non riceve i segnali che lo conducono in due possibili stati:

- se $TC8X=1$ e $STR=0$, ovvero ancora non è avvenuta la transizione da IDLE a START, ma il contatore 8X ha terminato di contare, il sistema salta nello stato "IDLE_B";
- se $TC8X=0$ e $STR=1$, ovvero se il contatore 8X non ha ancora terminato di contare, ma è avvenuta la start transition, allora il sistema salta nello stato "START_A".

Successivamente in "IDLE_B", al colpo di clock dopo, è presente la condizione $TC8X=0$ e $STR=0$, che permette di tornare in "IDLE_A". Questo loop concede alla CU di inviare i comandi necessari al sovracampionamento.

Per quanto riguarda invece il caso in cui il sistema si trovi in "START_A", si presenta nuovamente il caso con più strade ovvero il sistema resta in quello stato finché non arrivano diverse condizioni che lo portano in due differenti stati:

- se $TC8X=1$ e $TC4=0$, ovvero il contatore 8X ha terminato di contare, mentre il contatore, che conta i 4 shift dopo il rilevamento della transizione, non ha ancora terminato di contare, allora il sistema salta nello stato "START_B" e fa tutto ciò che deve fare per eseguire lo shift nello shift register;
- se $TC8X=1$ e $TC4=1$, cioè entrambi i contatori hanno terminato di contare, il sistema salta nello stato "START_C".

In seguito a ciò, se il sistema è nello stato "START_B", allora al colpo di clock successivo soddisferà le condizioni $TC8X=0$ e $TC4=0$, che lo riporteranno in "START_A"; mentre se si trova nello stato "START_C", al colpo di clock dopo, per $TC8X=0$, salterà in "DATO_A", ovvero lo stato che indica l'inizio della ricezione del dato trasmesso.

Una volta che il sistema è in questo stato, ha diverse possibilità a seconda delle condizioni che vengono soddisfatte, quindi questo resterà nello stato corrente fino a quando non avverrà che:

- TC8X=1 e TC=0, ovvero il contatore 8X ha terminato di contare, mentre il contatore, che conta 2603 colpi di clock, non ha ancora terminato di contare, allora il sistema salta nello stato "DATO_B";
- TC8X=1 e TC=1, cioè entrambi i contatori hanno terminato di contare, il sistema salta nello stato "DATO_C".

Al colpo di clock successivo, per entrambi tali stati viene soddisfatta la condizione TC8X=0 e TC=0, che nel caso "DATO_B", riporterà il sistema nello stato "DATO_A", mentre nel caso "DATO_C" lo porta nello stato "DATO_D".

Una volta in questo stato, il sistema si trova nuovamente a restare nello stato attuale oppure:

- se TC8X=1, TC=1 e TTR=0, ovvero il contatore 8X ed il contatore 1 hanno terminato di contare, mentre il contatore 2, che conta i bit della word ricevuti, non ha ancora terminato di contare, allora il sistema salta nello stato "DATO_E", questo loop permette di sovracampionare il dato in ricezione e di shiftare il bit ricevuto nel secondo shift register;
- se TC8X=1, TC=0 e TTR=0, cioè solo il contatore 8X ha terminato di contare, il sistema salta nello stato "DATO_C_1", questo loop è necessario in quanto solamente in istanti temporali particolari un bit della word è acquisito, però ogni volta che il contatore 8x finisce di contare si opera uno shift del sovracampionatore;
- se TC8X=1, TC=1 e TTR=1, indica che tutti e tre i contatori hanno terminato di contare e che la ricezione è entrata nella fase finale, il sistema si sposta in "DATO_F".

A questo punto se il sistema si trova in "DATO_E" oppure in "DATO_C_1", vengono soddisfatte le condizioni TC8X=0, TC=0 e TTR=0, che lo riportano in "DATO_D". Se invece si trova in "DATO_F", per TC8X=0, TC=0 e TTR=1 salta nello stato "OUTPUT", in cui nel secondo shift register entra l'ottavo e quindi ultimo dato trasmesso della word ed in cui si attiva OE2, ovvero l'abilitazione dell'uscita del dato interno al secondo shift register, dato che ora contiene l'intera word ricevuta dal trasmettitore.

Successivamente con la condizione ACK=1, che è inviata dal sistema utilizzatore ed indica che questo ha ricevuto la word, in sistema salta in "OUTPUT_OUT" e al colpo di clock successivo salta nuovamente nello stato iniziale "IDLE_A" in base alla condizione ACK=0.

4.5 Simulazione Modelsim

Al fine di testare il corretto funzionamento del ricevitore è stato deciso di far ciò sfruttando l'intera UART, il dato ricevuto è stato fornito dal trasmettitore della stessa. Tale test è descritto nella sezione relativa alla UART.

5 Uart

Il progetto finale consiste nel design di una UART completa, riportata precedentemente in Figura 2, e per tale motivo è stata progettata una top level entity, che racchiudesse sia il ricevitore che il trasmettitore.

5.1 Simulazione su Modelsim

Le verifiche preliminari del sistema sono state effettuate nell'ambiente Modelsim, simulando via software tutti i segnali necessari all'invio e alla ricezione di un dato, per fare ciò è stato collegato il trasmettitore insieme al ricevitore. I risultati della simulazione sono visibili in Figura 12.



Figura 12: Simulazione di una ricezione

Dalla simulazione è possibile notare nettamente che il dato trasmesso viene ricevuto dopo un certo lasso di tempo (dettato dal tempo di bit), insieme alla ricezione e alla presentazione del dato viene alzata anche la linea "RX_RDY". Il ricevitore mantiene il dato ricevuto finché non viene assegnato l' "ACK", in seguito il bus è posto a zero in attesa di una nuova ricezione. Come si può notare anche dalla simulazione Modelsim, il dato trasmesso è lo stesso ricevuto.

5.2 Test su Board DE1

La verifica finale del sistema è stata fatta progettando un ulteriore top-level per sfruttare gli switch, i led e i button della board, questo è stato fatto utilizzando il Pin Assignment di Quartus, i test sono stati condotti sulla FPGA ALTERA Cyclone V "5CSEMA5F31C6N" fornita dal laboratorio. E' stato deciso di mostrare il dato ricevuto sui led, il dato è stato inserito tramite gli switch così da cambiare il contenuto della word ad ogni trasmissione, mentre il reset è stato collegato all'interruttore a bottone. Dato che vi era la necessità di verificare il risultato tramite i led, è stato sfruttato un contatore con un delay adeguato per generare il "DATA_VALID". Tale contatore è fornito di un counter enable collegato ad uno switch. Quindi con tale sistema è stato possibile avere controllo delle trasmissioni con un timing adeguato a verificare il corretto funzionamento. Il segnale di Acknowledgement è stato fornito tramite uno switch.

Inoltre tale sistema lavora a 25 MHz, ma la scheda DE1 forniva solo un clock a 50 MHz, quindi è stato reso necessario l'utilizzo di un PLL, implementato utilizzando le librerie Intel disponibili su Quartus.

Le periferiche utilizzate sono:

- KEY(0): Reset
- KEY(1): ACK

- LEDR(9): TX_RDY
- LEDR(8): RX_RDY
- SW(7 downto 0): D_OUT
- SW(8): Enable (contatore esterno)
- LEDR(7 downto 0): D_IN

Lo schema a blocchi del test hardware è riportato in figura 13.

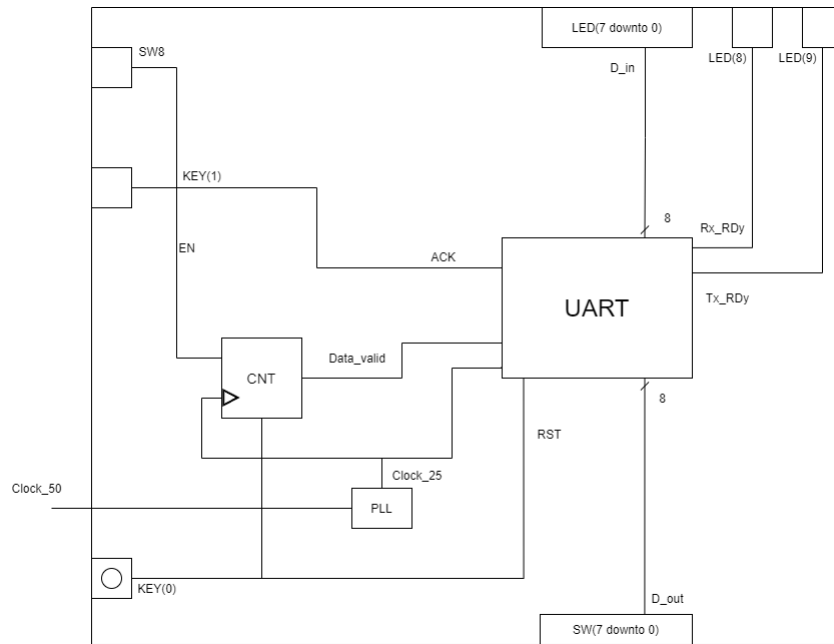


Figura 13: FPGA test

Di seguito nelle Figure 14, 15, 16 sono riportati i vari test effettuati in laboratorio con la board DE1.

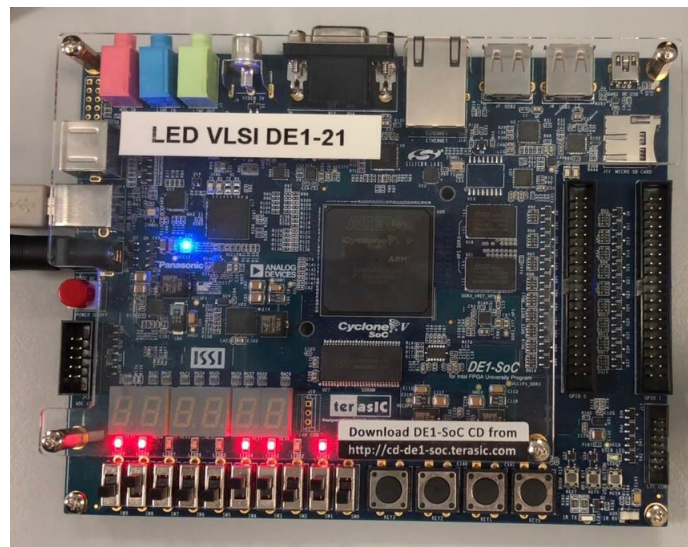


Figura 14: Test con dato trasmesso D_OUT=00011010

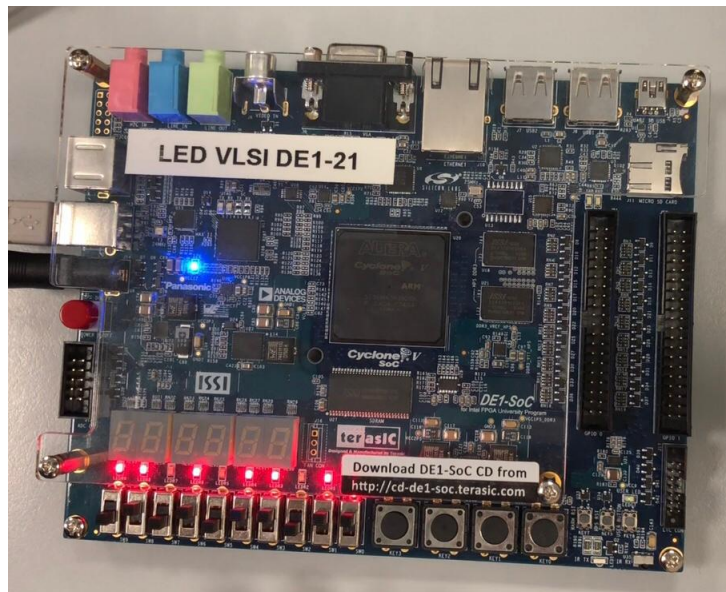


Figura 15: Test con dato trasmesso $D_OUT=01011011$

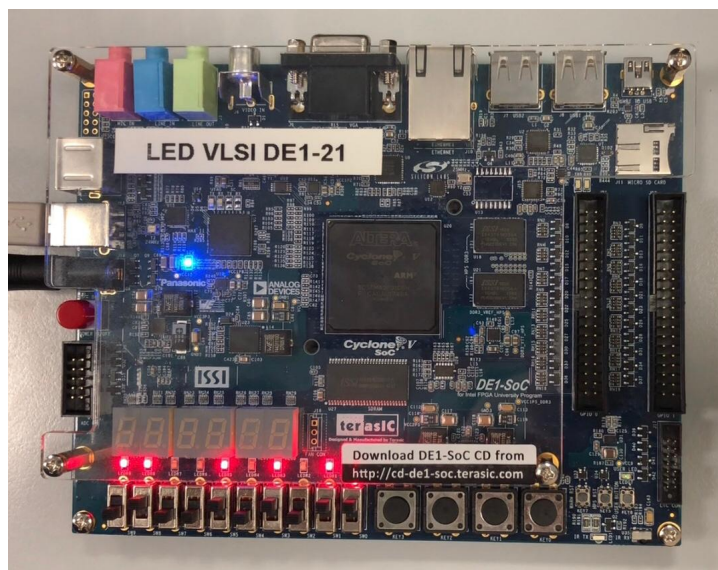


Figura 16: Test con dato trasmesso $D_OUT=00101011$

La validità dei test può essere verificata tramite i valori assunti dagli switch e i led accesi. Tutti i test eseguiti hanno dato esito positivo.

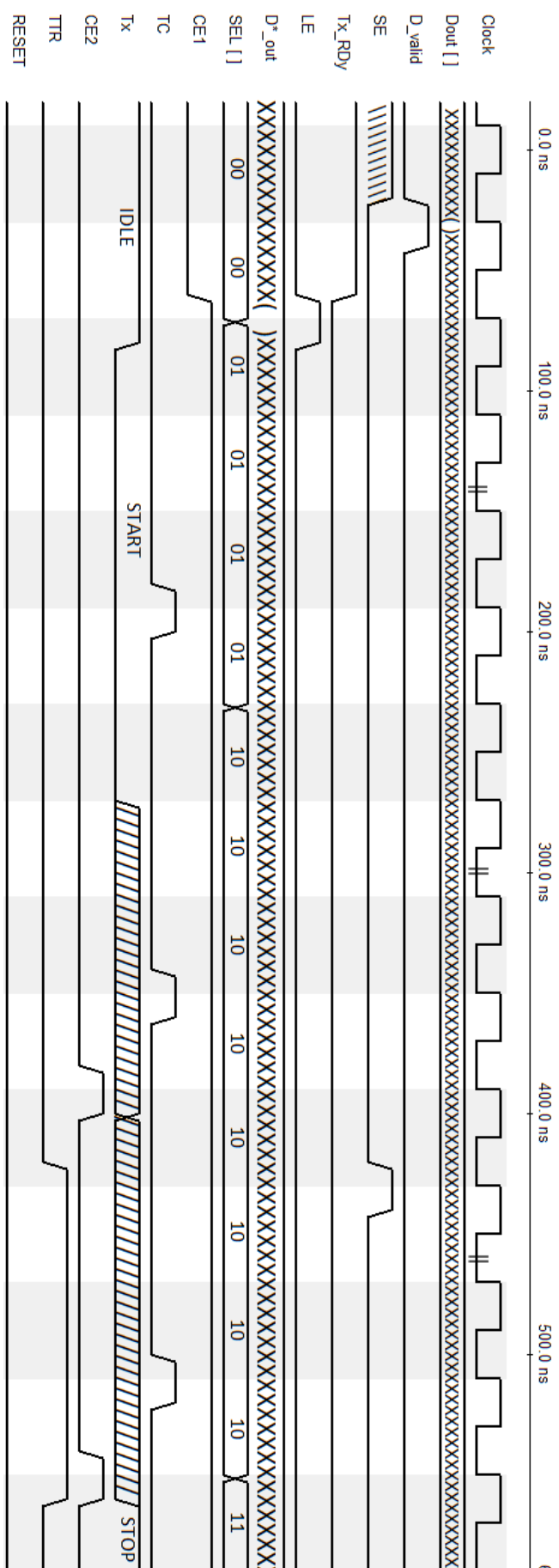


Figura 17: Timing trasmettitore

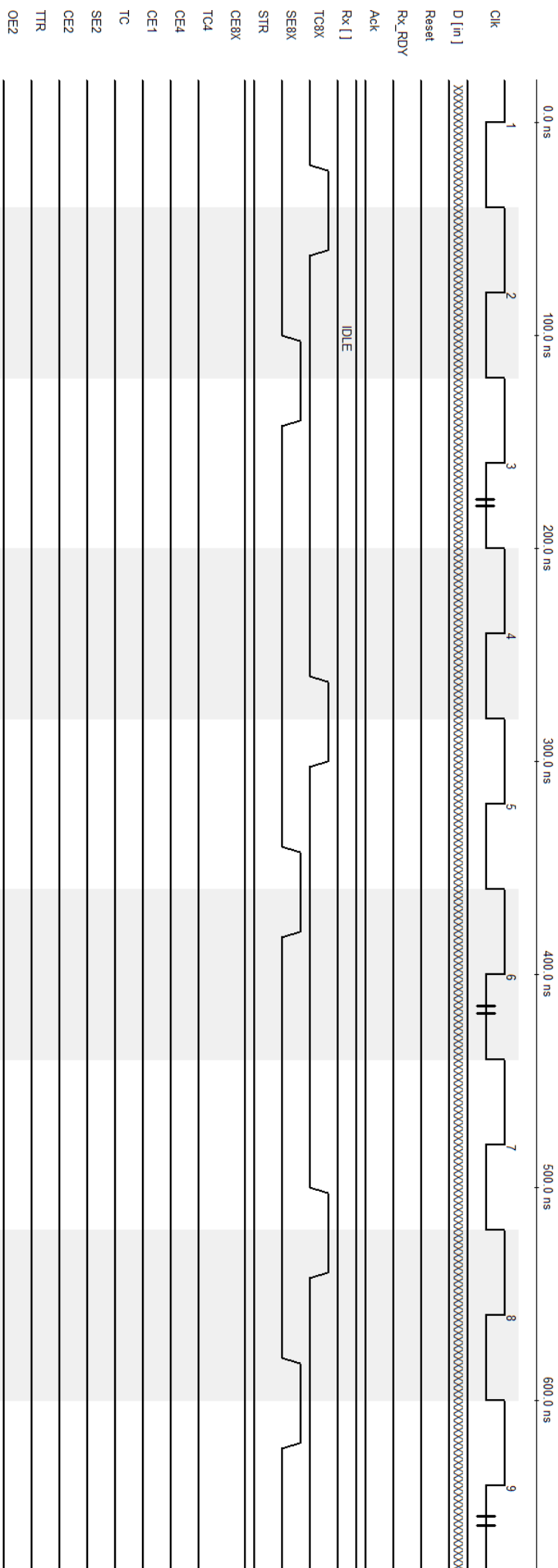


Figura 18: Timing Ricevitore

