Politecnico di Torino

Laurea Magistrale in Ingegneria Elettronica
Anno Accademico 2021-2022

Relazione di laboratorio di sistemi digitali integrati

Interfaccia UART

Docenti:

Prof. Massimo Ruo Roch
Prof. Maurizio Zamboni

Studenti:

Marco Massetti matr. 301163
Pietro Fagnani matr. 303490

Indice

1. Introduzione

L'obbiettivo di questo progetto è la creazione di un'interfaccia UART per poter scambiare dati tra FPGA e microcontrollore utilizzando lo standard RS232.

Le specifiche richieste sono:

- 9600 baud rate (± 3%)
- 8 bit per il dato
- 1 bit di start (livello logico 0)
- 1 bit di stop (livello logico 1)
- No parity bit

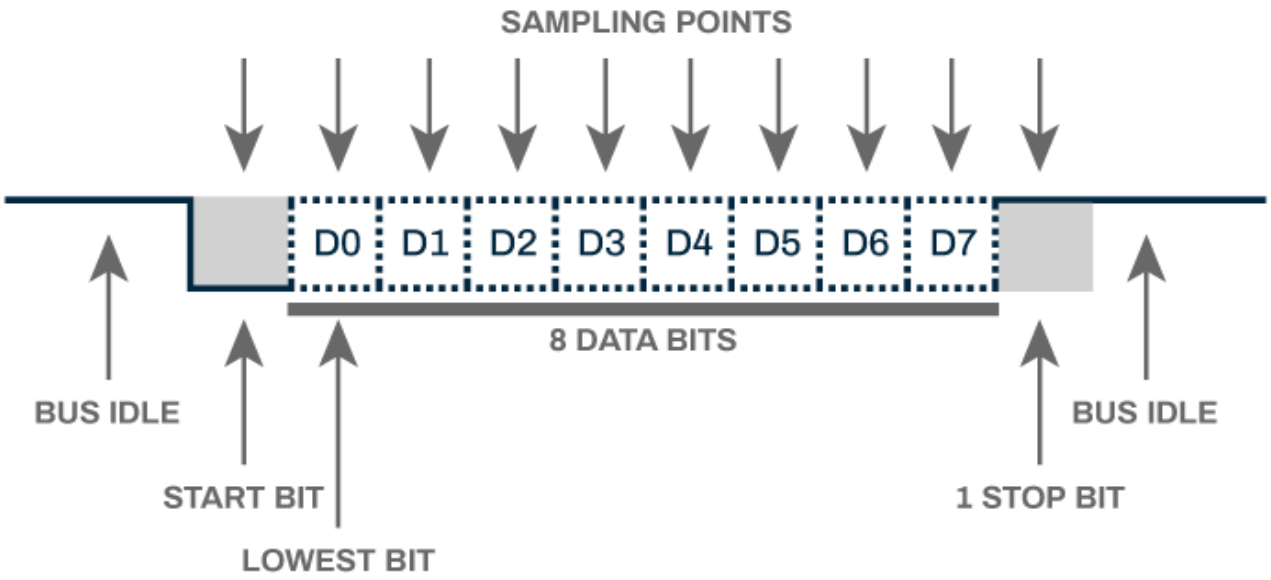

Figura 1 Frame comunicazione RS232

L'UART deve essere divisa in due parti, una per la trasmissione "*TX*" ed una per la ricezione "*RX*".

- TX: Ha una porta in ingresso "*Din*" ad 8 bit per ricevere il dato da trasmettere e un segnale "*Te*" per avviare la trasmissione. In uscita ha un segnale su un solo bit per trasmettere in modo seriale come nell'immagine ***(Figura 1)***.

- RX: In ingresso riceve un segnale come quello sopra indicato e in uscita ha una porta a 8 bit per il segnale "*Dout*" per fornire il dato ricevuto e un segnale "*Dr*" per indicare quando il dato ottenuto è valido.

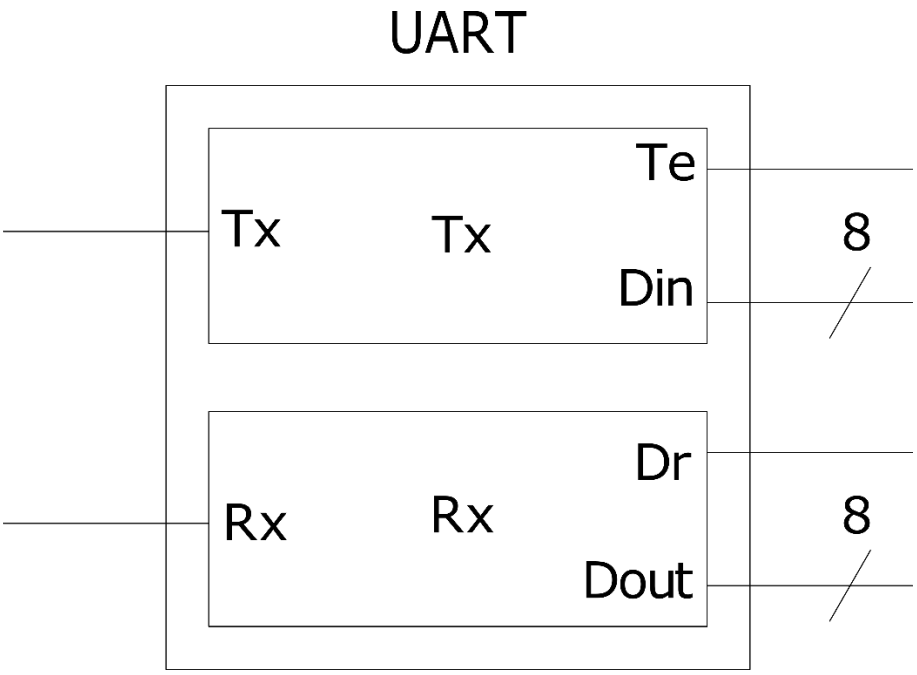

Figura 2 Schema a blocchi

2. Trasmettitore

2.1. Datapath

Quando viene ricevuto in ingresso il segnale "*Te*", il registro "*PIPE_REGISTER*" campiona il dato in ingresso e lo memorizza per tutta la durata utile.

Il dato viene poi mandato al registro "*SHIFT_REGISTER*", il quale ha in ingresso come MSB sempre '0' siccome è il valore dello START BIT. Esso funziona come un registro ad ingresso parallelo ed uscita sequenziale; quindi, in uscita manda un bit per volta partendo da quello più significativo fino a quello meno significativo.

L'uscita dello "*SHIFT_REGISTER*" è in ingresso in una porta OR insieme ad il segnale "*ONE*" della control unit, questo permette di avere il segnale "Tx" ad '1' durante lo stato di IDLE e per trasmettere lo STOP BIT.

Il "*COUNTER*" permette di lavorare ad una velocità di trasmissione di 9600 baud partendo dal clock a 10MHz della scheda; infatti, conta fino a 1040 e come si vede nel Timing Diagram in ***(Figura 4)*** con altri due stati della control unit si arriva ad ottenere che ogni bit tramesso ha una durata di 1042 colpi di clock.

$$\frac{10 \cdot 10^6}{1042} = 9597\ baud$$

La velocità di trasmissione ottenuta dividendo il clock è lievemente inferiore a quella teorica ma rientra ugualmente nelle specifiche grazie alla tolleranza del 3% di errore.

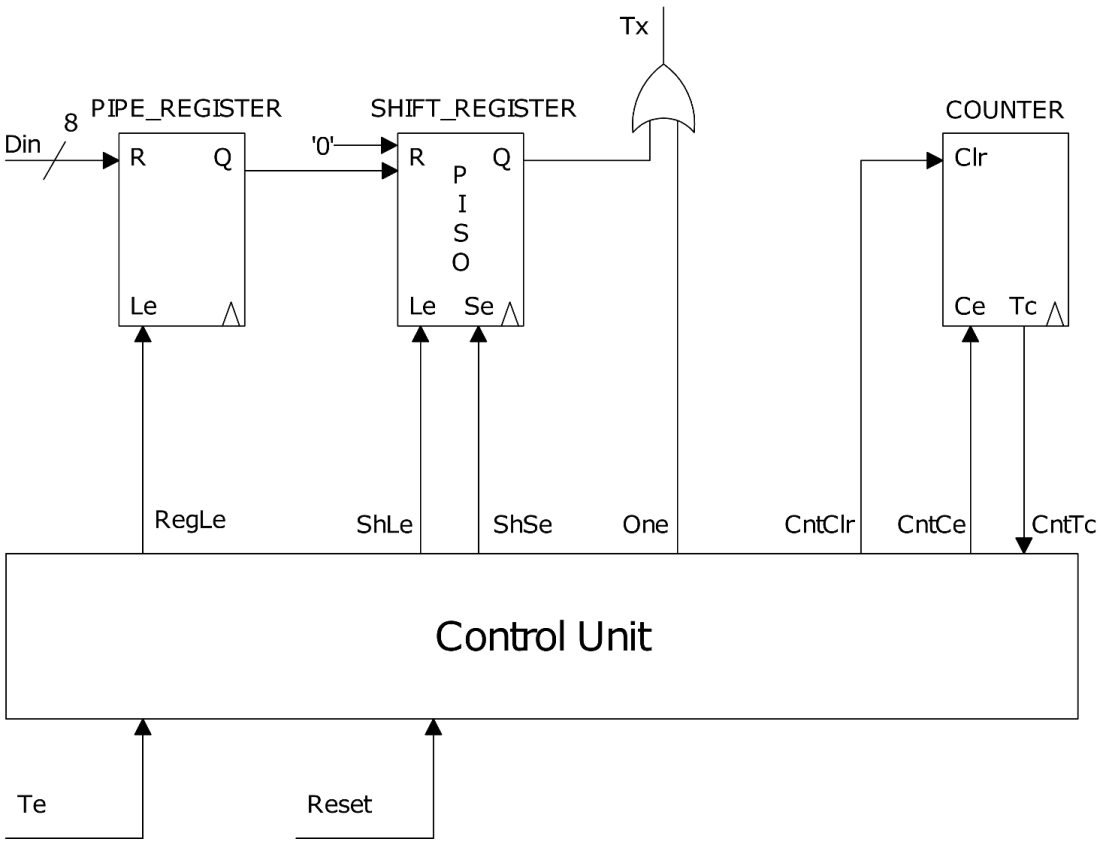

Figura 3 Datapath trasmettitore

2.2. Timing Diagram

Con l'ausilio del Timing Diagram sono state stabilite le tempistiche dei segnali e quindi il comportamento dei componenti.

Come si può notare il segnale che avvia la trasmissione è il "*Te*" che indica la presenza un dato in ingresso valido. Questo significa che il campionamento dei dati in ingresso può essere interrotto azzerando il segnale "*RegLe*" e tenendo così in memoria il dato valido.

Il colpo di clock successivo vengono caricati parallelamente i dati nella piso (asserzione del segnale "*ShLe*") e viene resettato il contatore.

Viene poi abilitato il contatore. Invece, il segnale "*One*", che è rimasto asserito per tutta la durata dell'idle, viene posto a 0.

Quando il counter arriva a 1040 porta ad '1' il terminal counter "*CntTc*".

Nel colpo di clock successivo viene effettuato lo shift della piso (segnale "*ShSe*") che di conseguenza cambia il bit in uscita e quindi il valore di Tx, viene inoltre abilitato il clear asincrono del counter "*CntClr*" che permette di avere il counter a '0' per due periodi di clock.

Dopo i segnali "*ShSe*" e "*CntClr*" vengono immediatamente disabilitati.

Questa sequenza viene ripetuta per tutti i bit con l'unica differenza che per lo STOP BIT il valore in uscita viene impostato ad '1' dal segnale "*One*".

Finita la trasmissione dello STOP BIT il segnale di enable del "*COUNTER"* torna a '0' e il "*PIPE_REGISTER*" ricomincia a campionare i dati in ingresso a seguito dell'asserzione del segnale "*RegLe*".

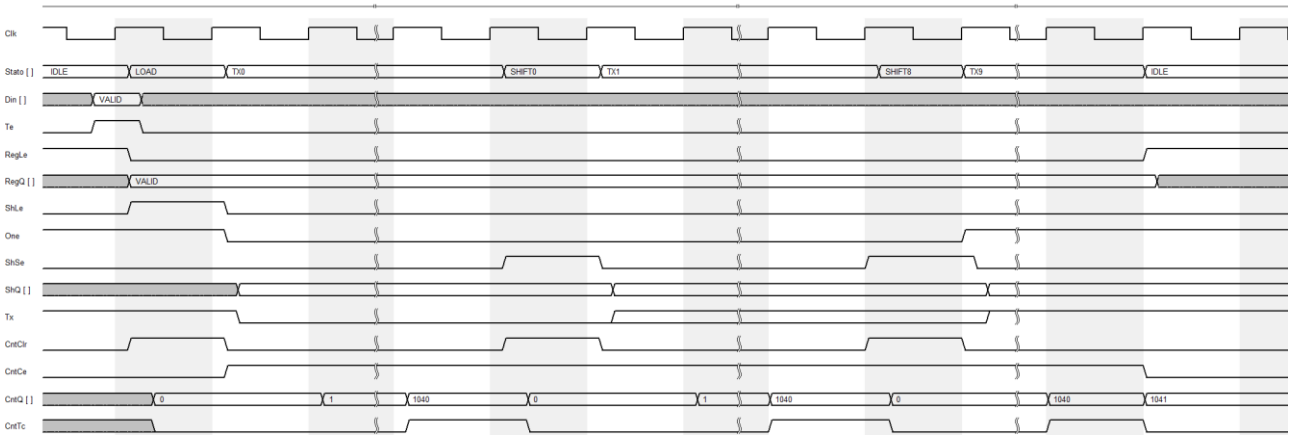

Figura 4 Timing diagram trasmettitore

2.3. Diagramma degli stati

Dal diagramma degli stati si può capire il funzionamento della Control Unit.

È presente uno stato per gestire i reset. Uno stato di "*IDLE*" per quando il circuito non deve trasmettere.

Appena viene ricevuto il segnale di "*Te*" la macchina entra nello stato di "*LOAD*" che permette di caricare i dati in ingresso e con "*TX0*" viene trasmesso lo START BIT.

Quando "Tc" = 1 viene effettuato lo shift sulla piso per iniziare a tramettere il secondo bit. La stessa operazione viene ripetuta per tutti i bit per poi tornare nello stato di "*IDLE*" e quindi essere pronti a trasmettere un nuovo dato.

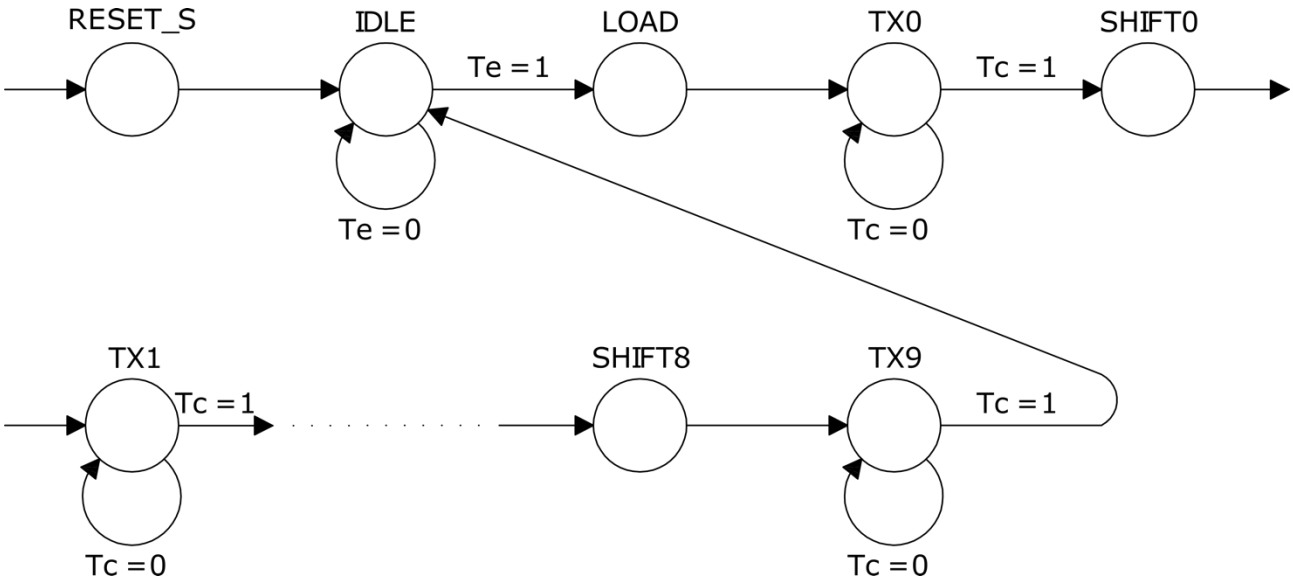

Figura 5 Diagamma degli stati trasmettitore

3. Ricevitore

3.1. Datapath

La ricezione dei dati deve essere avviata ogni volta che viene rilevato il bit di start, quindi, per evitare errori, è necessario identificare con precisione il momento di inizio della comunicazione. È inoltre importante eseguire un campionamento dei dati in ingresso nell'istante centrale del tempo di bit in modo da minimizzare l'interferenza intersimbolica. Per questi motivi sulla linea dati è eseguito un "oversampling 8X", ovvero un campionamento della linea a otto volte più veloce del baud rate, questo viene realizzato tramite un contatore "*COUNTER_8*".

Il terminal count del contatore è stato ricavato in modo tale da minimizzare l'errore rispetto alla velocità di trasmissione rischiesta. Per ottenere un campionamento otto volte superiore al boud rate è necessario dividere di 130 volte il clock.

$$\frac{10 \cdot 10^6}{130 \cdot 8} = 9615\ baud$$

Considerando che, come è possibile notare dal timing diagram in ***(Figura 8)***, ad ogni ciclo del contatore vengano aggiunti due colpi di clock, per ogni campione il terminal count impostato è pari a 128. La ricezione risulta lievemente più rapida del valore ottimo ma rientra nei limiti imposti.

I dati vengono campionati tramite in un registro a caricamento seriale e lettura parallela "*INPUT_REGISTER*" in modo da permettere, tramite un comparatore, l'identificazione del bit di start.

L'inizio della comunicazione è identificato grazie al comparatore che riconosce il fronte di discesa del bit di start confrontando gli ultimi otto campioni contenuti in "*INPUT_REGISTER*" con l'apposita codifica.
In caso di un positivo accertamento di inizio della comunicazione la macchina ha ricevuto solo mezzo bit di start, quindi, per avere all'interno del registro di ingresso solo i campioni relativi al bit ricevuto è necessario campionare per un ulteriore mezzo tempo di bit, e per questo è utilizzato il contatore "*COUNTER_SYNC*".
Esso ha un valore di terminal count impostabile dall'esterno che, dato l'oversampling otto volte maggiore del baud rate, in questo caso è impostato a quattro a causa dello sfasamento iniziale, e verrà successivamente aumentato ad otto tramite un multiplexer per prelevare i campioni di ogni bit.

Il primo bit ricevuto, ovvero il bit di start, viene momentaneamente salvato nel registro di uscita ma sarà poi sovrascritto siccome non contiene informazioni.

Una volta acquisiti gli otto campioni appartenenti al primo bit contenente informazioni ne vengono prelevati i tre centrali, per ridurre il rischio di letture errate causate dall'interferenza intersimbolica. Su di essi viene inoltre eseguita un'operazione di voting per ridurre la probabilità di avere errori a causa del rumore. Il risultato di questa operazione è infine inserito all'interno di un registro di uscita *"OUTPUT_REGISTER*"

Le operazioni di lettura sono ripetute per ogni bit da ricevere contando il numero di ripetizioni tramite il contatore "*COUNTER_W*".

Una volta letti tutti i bit di dato viene eseguito un ultimo campionamento per caricare il bit di stop nel registro di input per verificare la corretta ricezione. Se la ricezione è avvenuta con successo viene asserito il segnale di "*DataReady*" per un singolo colpo di clock e, successivamente, la macchina torna in stato di idle in attesa di una nuova comunicazione.

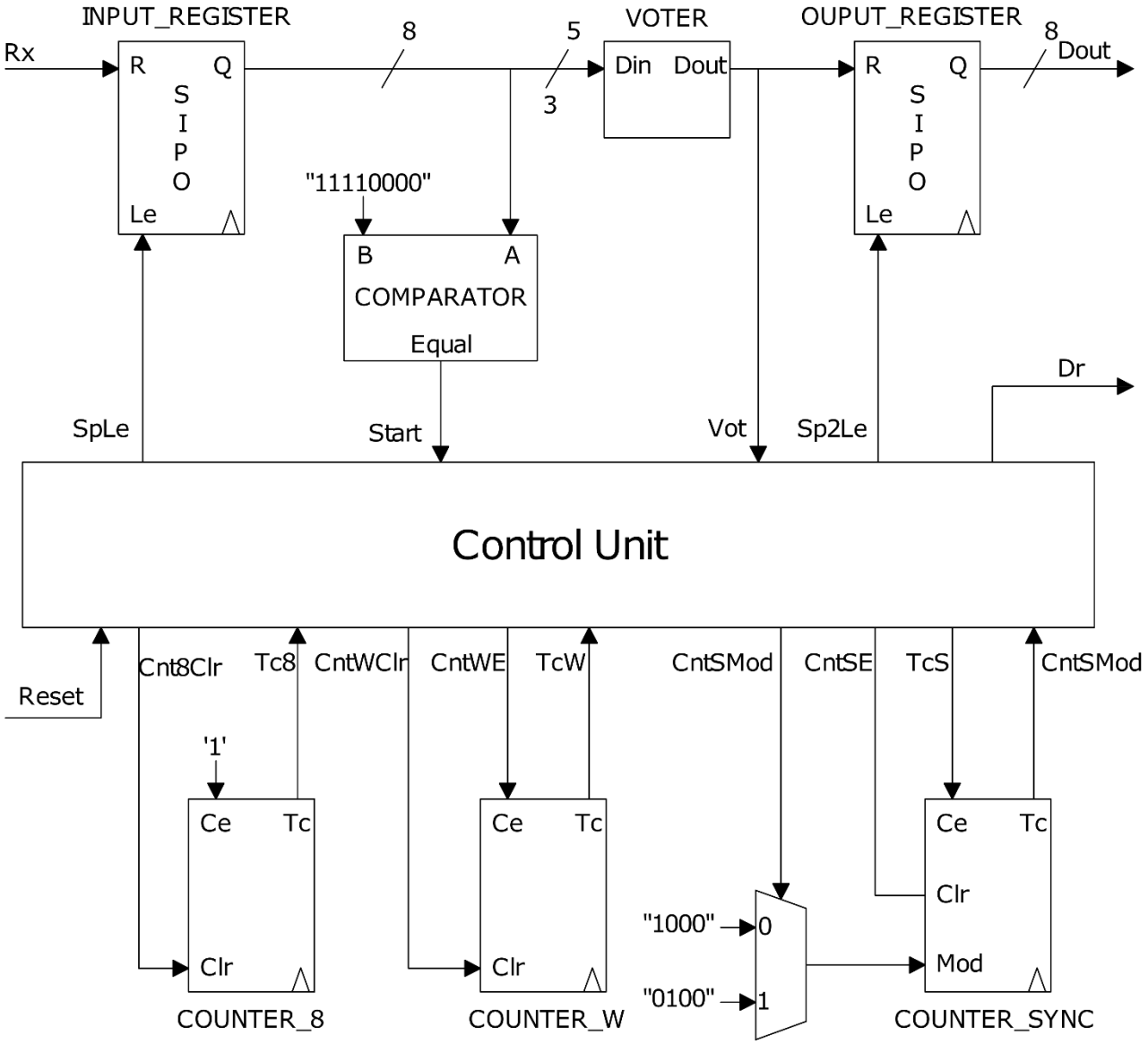

Figura 6 Datapath ricevitore

3.2. Timing Diagram

Come nel caso del trasmettitore è stato creato il timing diagram al fine di derivare gli stati della control unit.

Il diagramma è stato qua suddiviso in sezioni per maggiore chiarezza.

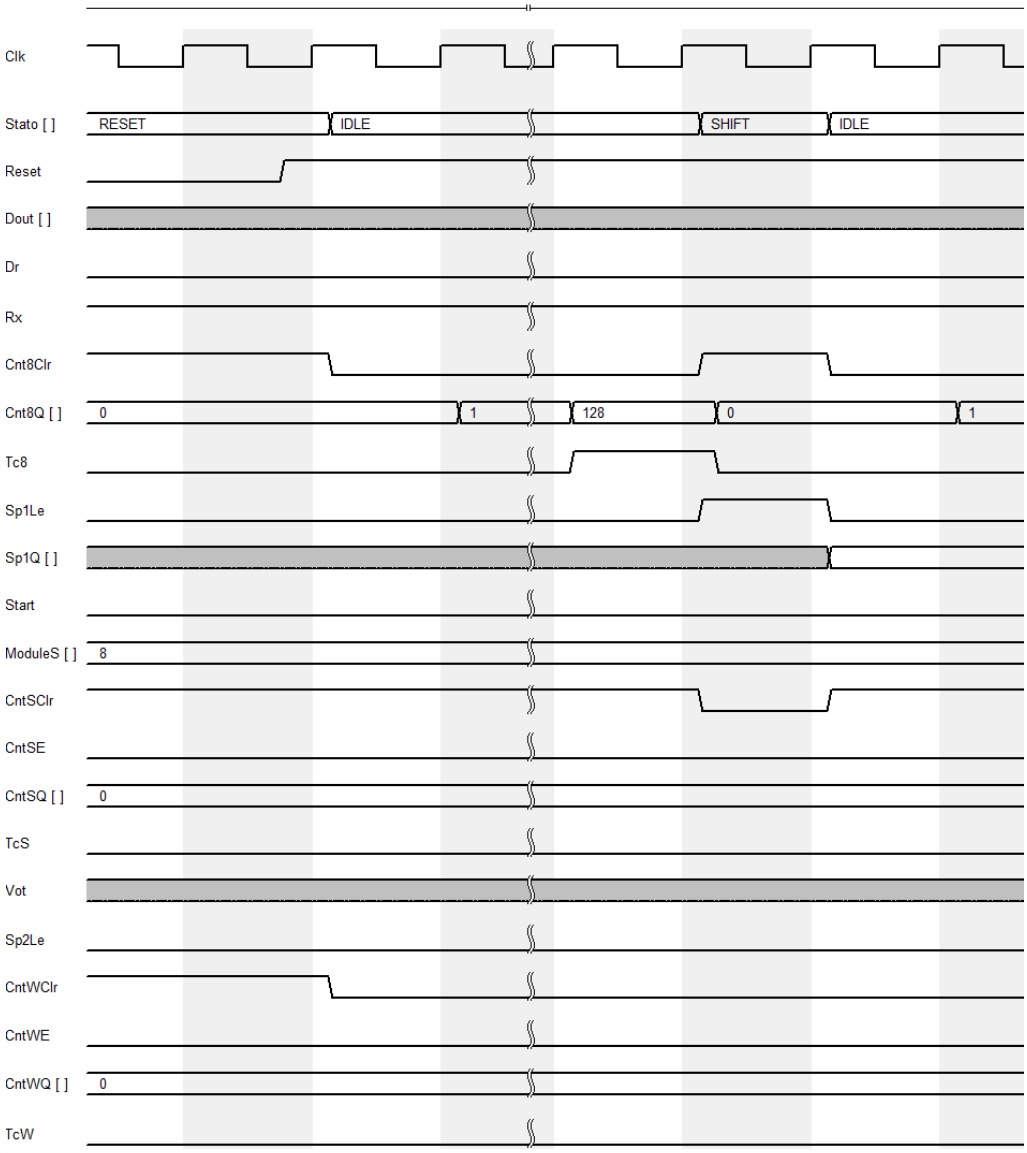

Figura 7 Idle e campionamento ad 8x

Finché non viene ricevuto il bit di start la linea dati viene campionata dal registro di ingresso ad una velocità otto volte superiore al boud rate.

L'intervallo di campionamento è scandito dal segnale di terminal count "*Tc8*" del contatore "*COUNTER_8*".

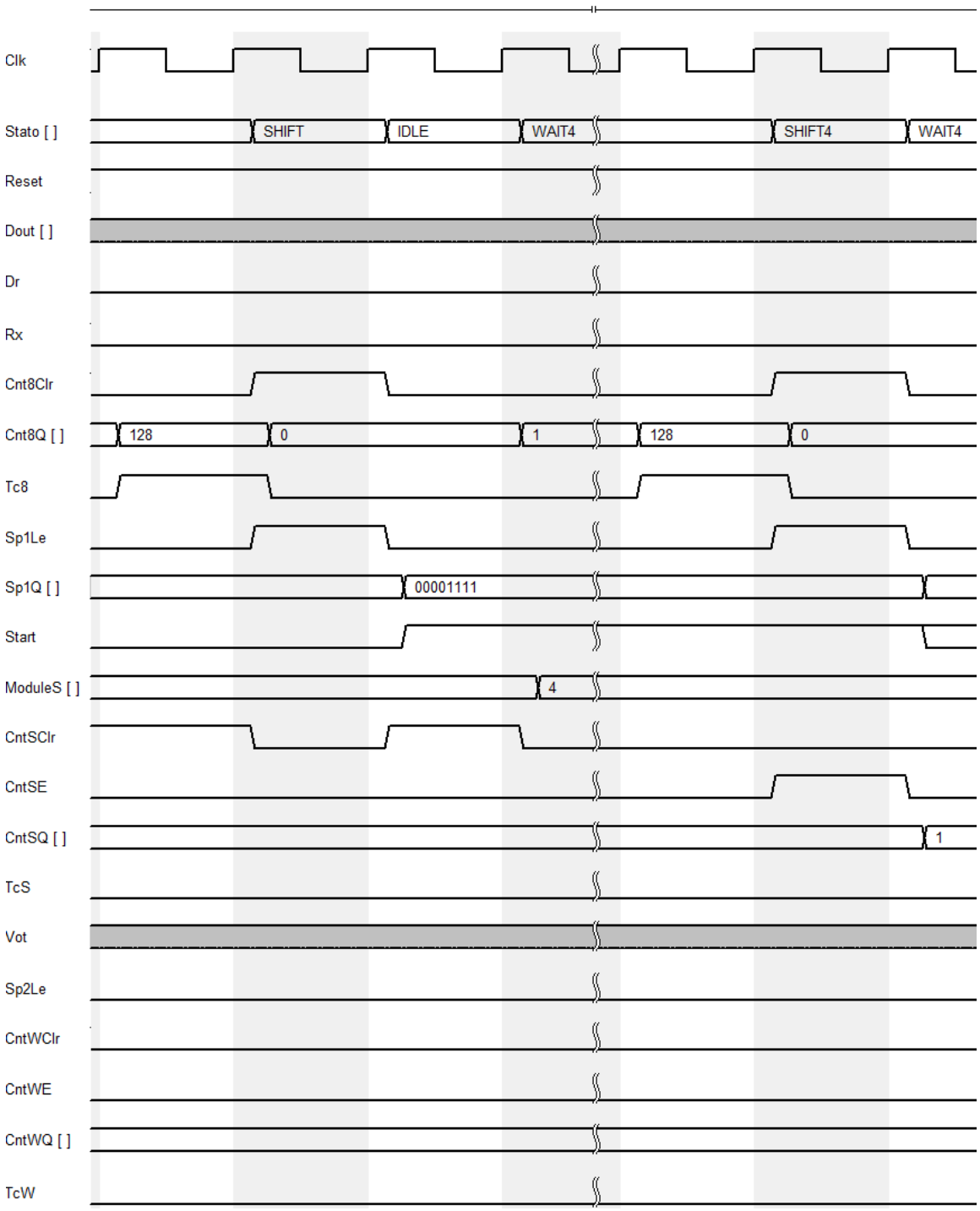

Figura 8 Rilevamento "Start" e attesa 4 campioni

Se, tramite il comparatore, viene identificato il segnale di start vengono prelevati gli ulteriori quattro campioni per portare tutti gli otto campioni del bit all'interno del registro di input: le operazioni di "*SHIFT4*" e "*WAIT4*" vengono quindi eseguite quattro volte. Sono necessarie per il conteggio di quattro cicli del "*COUNTER_8*" in modo tale da poter avere gli 8 campioni del registro di ingresso associati ad un singolo bit ricevuto.

Il conteggio del numero di campionamenti è fatto tramite il contatore "*COUNTER_SYNC*".

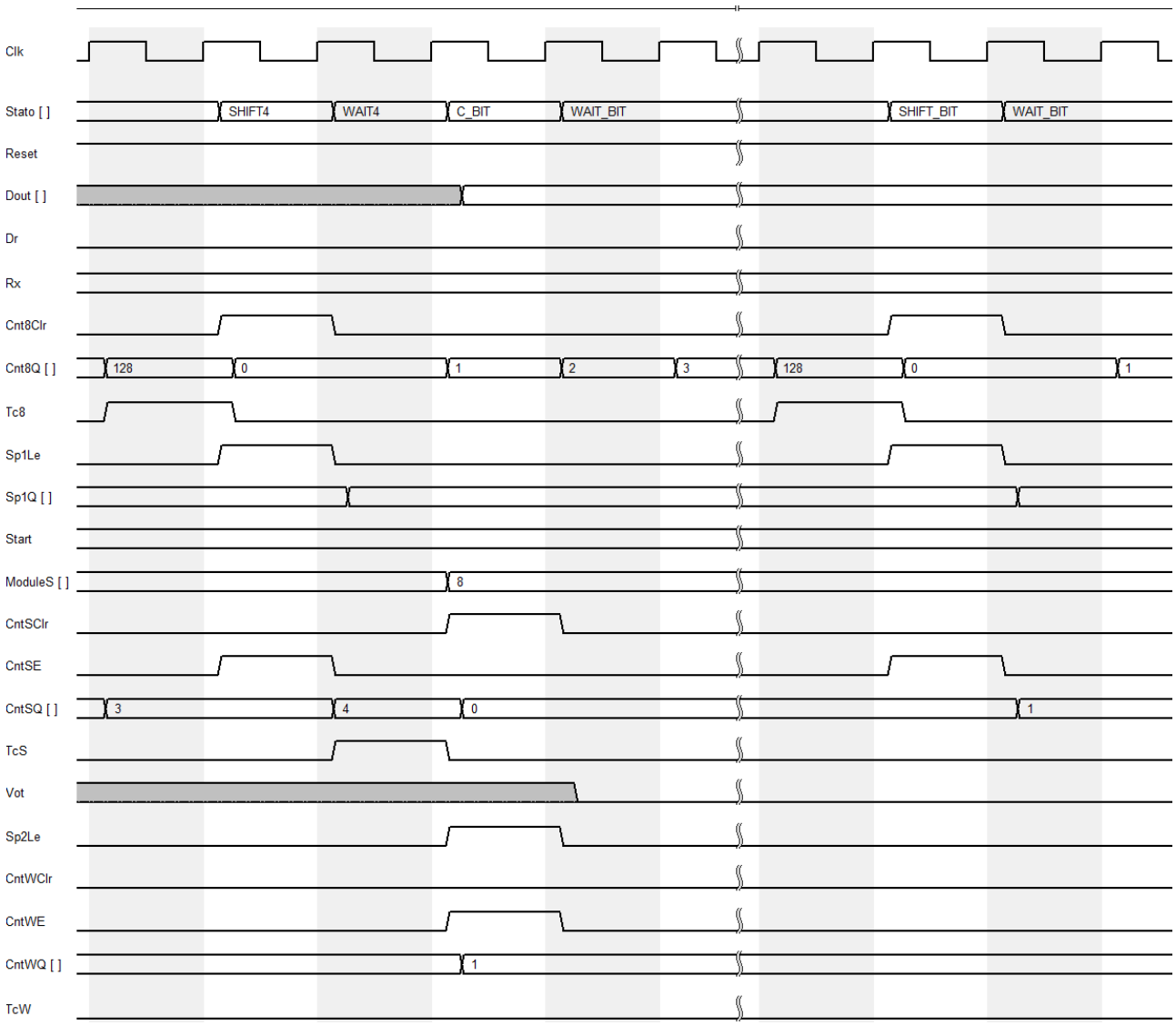

Figura 9 Campionamento del bit

Quando il bit è interamente contenuto nel registro di input viene eseguita l'operazione di voting e, data la natura combinatoria del componente "*VOTER*", è possibile campionare l'output nello stesso colpo di clock.

Successivamente vengono prelevati i nuovi campioni del bit seguente. L'operazione di "*C_BIT*" viene eseguita nove volte, ovvero una volta per ogni bit di dato più la prima esecuzione dovuta al bit iniziale. Il bit di start viene caricato quindi nel registro di uscita ma verrà poi sovrascritto.
Le operazioni di "*SHIFT_BIT*" e "*WAIT_BIT*" vengono eseguite per ogni campione di ogni bit e permettono di aspettare con l'utilizzo dei counter che i campioni del registro in ingresso siano di un singolo bit.

Il valore di terminal count del contatore "*COUNTER_SYNC*" viene incrementato ad otto per permettere di conoscere il numero di campioni prelevati di ogni bit.

Il numero di bit campionati è verificato tramite il contatore "*COUNTER_W*".

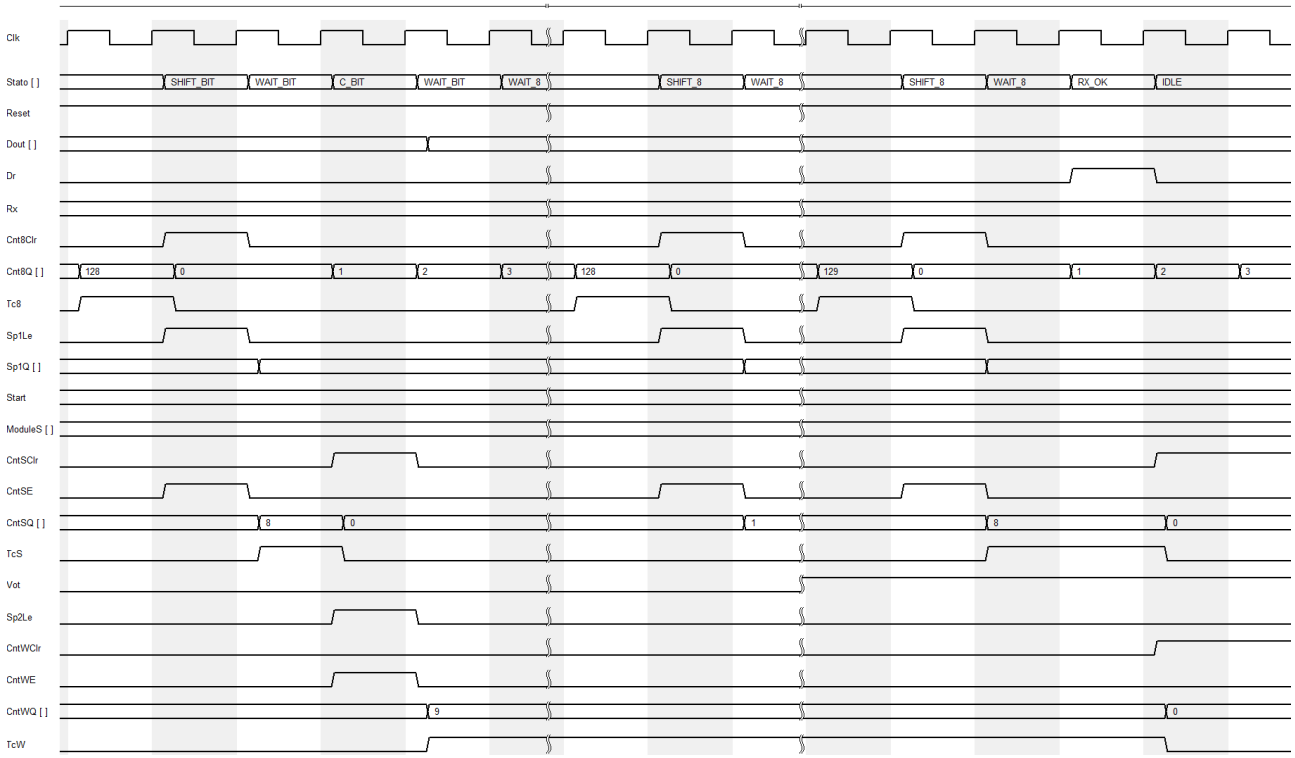

Figura 10 Campionamento bit di stop e ritorno in idle

Una volta ricevuti gli otto bit di dato viene eseguita un'ultima sequenza di otto campionamenti per ricevere il bit di stop. Quest'ultimo non viene inserito nel registro di uscita ma, come è stato fatto per tutti i bit, viene sottoposto ad un'operazione di voting per verificarne la correttezza.

Se il voting da esito positivo, ovvero il bit di stop ricevuto è asserito, viene comunicata la corretta ricezione tramite l'asserzione del segnale "*DataRecive*" e i dati letti sono disponibili all'uscita "*Dout*".

Una volta conclusa la ricezione la macchina può tornare in stato di idle in attesa di una nuova comunicazione.

3.3. Diagramma degli stati

Dall'analisi del timing diagram è stato ricavato il seguente diagramma degli stati.

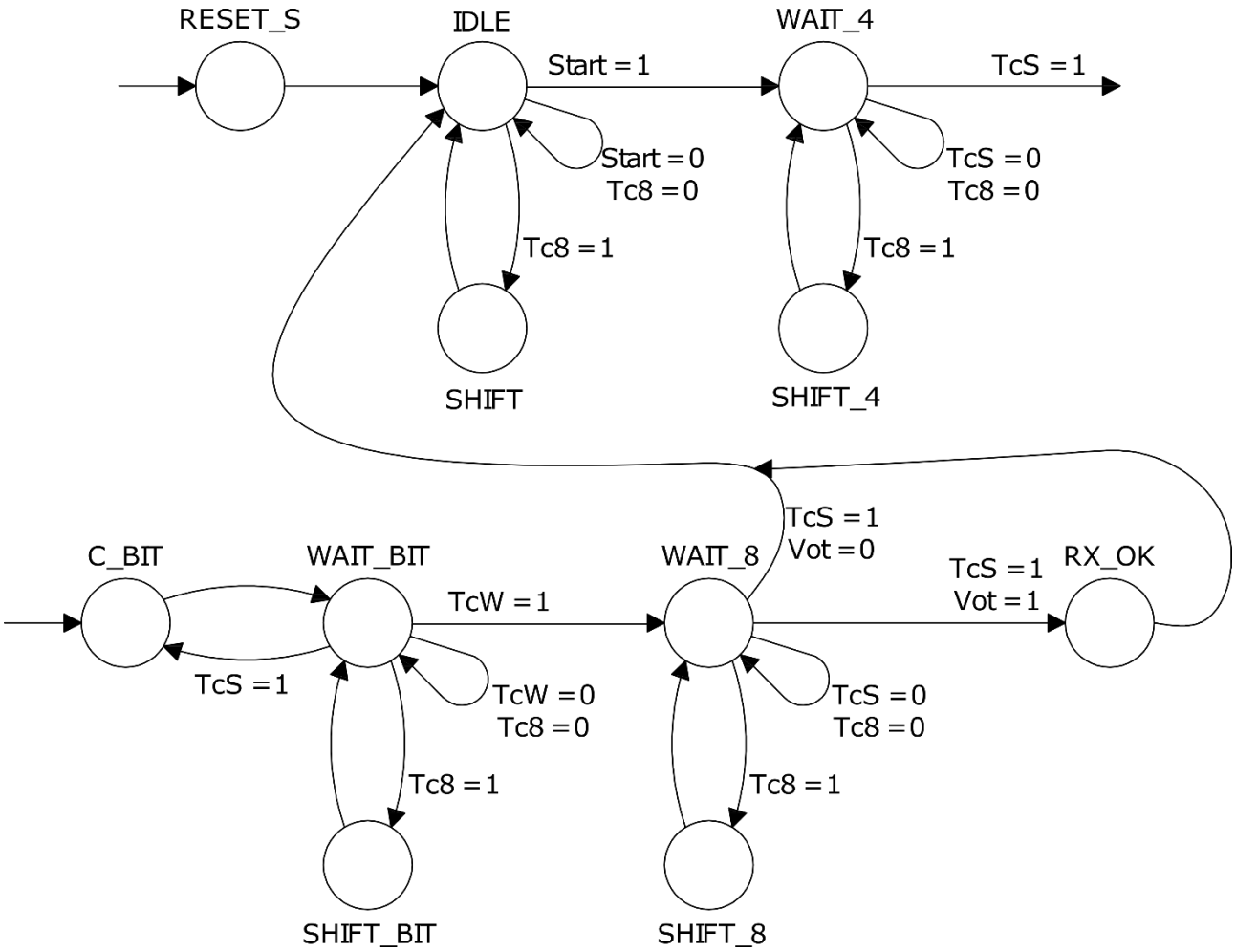

Figura 11 Diagramma degli stati ricevitore

Descrizione degli stati:

- RESET_S: stato raggiungibile in maniera asincrona da ogni altro stato a seguito della asserzione del segnale di reset. Viene eseguito l'azzeramento di tutti i contatori del datapath.

- IDLE: stato in cui la macchina ritorna al termine di ogni ricezione. Viene atteso il terminal count del "*COUNTER_8*" per eseguire il campionamento dell'ingresso. Se viene rilevato il bit di start viene avviata la ricezione.

- SHIFT: avviene il campionamento dell'ingresso prima di aver rilevato il bit di start.

- WAIT_4: viene atteso il terminal count del "*COUNTER_SYNC*" per eseguire il voting ed il salvataggio nel registro di uscita del bit ricevuto. Viene usato questo stato per portare tutti i campioni del bit all'interno del registro di input. Ad ogni terminal count del "*COUNTER_8*" viene prelevato un campione dalla linea di ricezione.

- SHIFT_4: avviene il campionamento dell'ingresso.

- C_BIT: il risultato dell'operazione di voting viene campionato dal registro di uscita

- WAIT_BIT: attesa della completa ricezione del bit. Ad ogni terminal count del "*COUNTER_8*" viene prelevato un campione dalla linea di ricezione. Ad ogni terminal count del "*COUNTER_SYNC*" viene salvato il bit ricevuto nel registro di uscita.

- SHIFT_BIT: avviene il campionamento dell'ingresso.

- WAIT_8: attesa della completa ricezione del bit di stop. Una volta ricevuto il terminal count del contatore "*COUNTER_SYNC*" viene verificato il valore del bit di stop e viene scelto se tornare in idle o passare allo stato "*RX_OK*"

- SHIFT_8: avviene il campionamento dell'ingresso.

- RX_OK: segnalazione della corretta ricezione e riporta la macchina in idle nel colpo di clock successivo.

4. Simulazioni

Prima della verifica su FPGA sono state eseguite delle simulazioni delle due parti che compongono l'interfaccia.

- Trasmettitore:

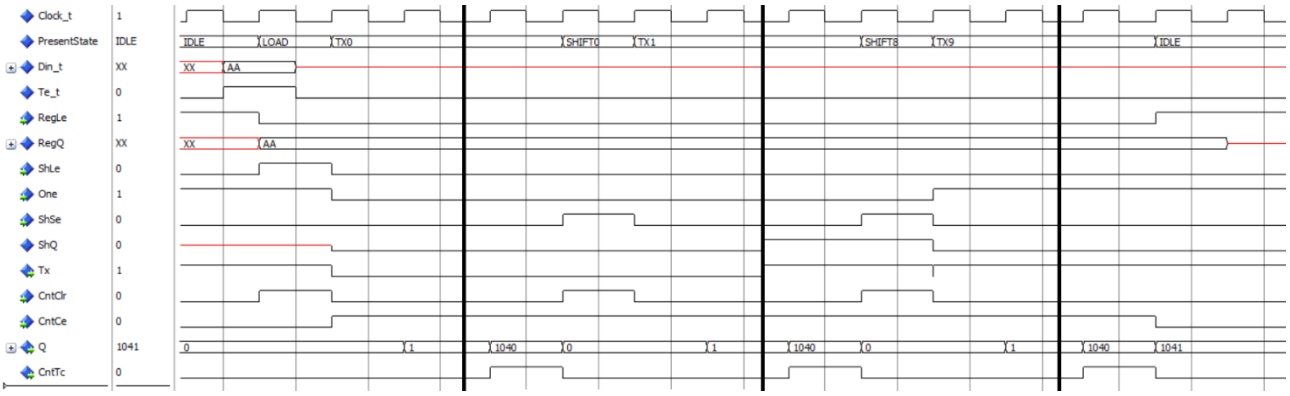

Figura 12 Simulazione trasmettitore

- Ricevitore:

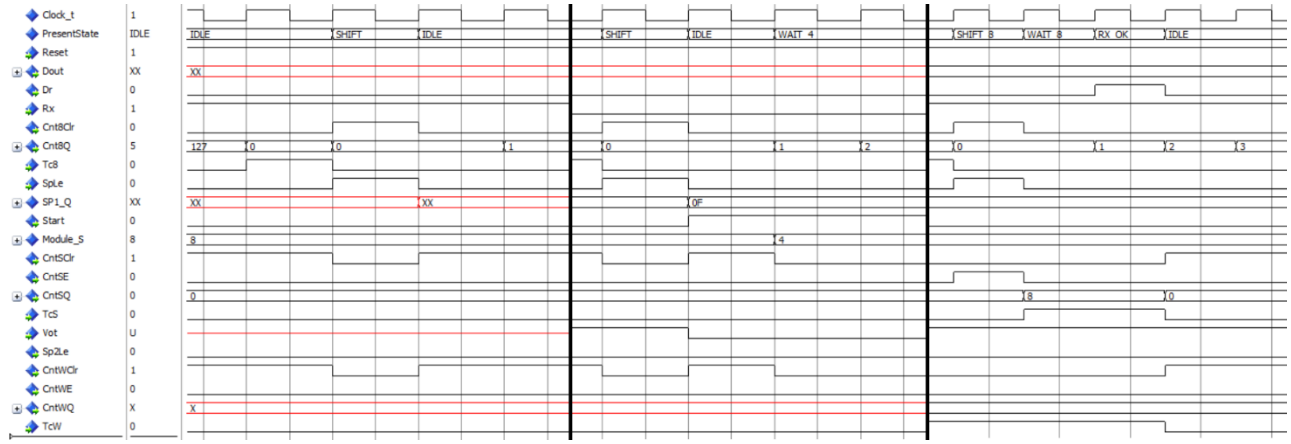

Figura 13 Simulazione ricevitore

Le due sezioni di simulazione mostrate corrispondono a quelle mostrate dei timing diagram creati nella fase di progetto. (Per il ricevitore sono state inserite solo le parti ritenute più significative al fine di semplificare la lettura)

È possibile vedere come ci sia una quasi perfetta corrispondenza tra i comportamenti attesi e quelli ottenuti.

Le uniche differenze visibili sono:

- Glitch dell'uscita del trasmettitore al termine di una trasmissione: È causato dal ritardo tra lo shift del registro di uscita e l'asserzione del segnale "One". Come si vedrà in ***(Figura 15)*** nel mondo reale non è visibile e, anche se fosse presente, avrebbe una durata ridotta.
- Overflow del contatore "Cnt8" nel ricevitore: Il contatore va in overflow dato che il valore da contare (128) va oltre la sua dinamica a 7 bit. Per come è stato descritto in VHDL il componente non è un problema dato che il terminal count viene asserito ugualmente.

5. Test e conclusioni

Per la verifica del circuito la porta di uscita del ricevitore a bordo della FPGA è stata collegata alla porta di ingresso del trasmettitore. In modo tale da ritrasmettere immediatamente i dati appena ricevuti. La porta di ricezione della UART sulla FPGA è stata a sua volta collegata al pin di trasmissione del microcontrollore.

In questo modo il microcontrollore trasmette valori casuali tramite seriale e riceve quello che la FPGA trasmette. I dati ricevuti ed inviati sono mostrati in esadecimale grazie ad un display, questo permette di verificare il funzionamento in quanto i due valori devono sempre coincidere.

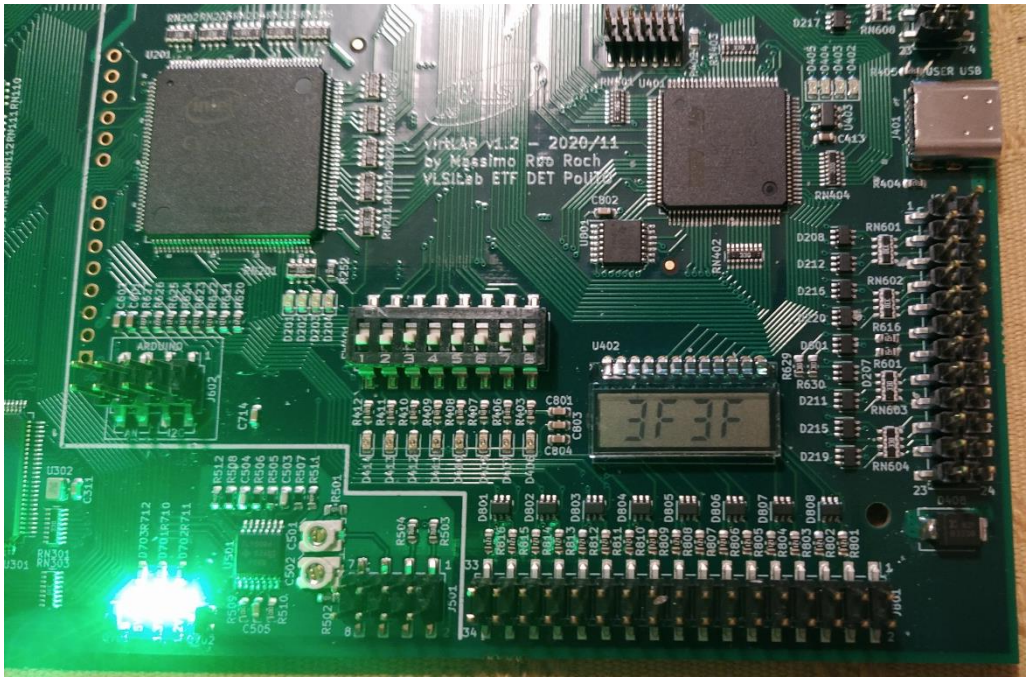

Figura 14 Corrispondenza dei byte trasmessi da microcontrollore e FPGA

Dopo aver osservato la trasmissione di diversi campioni è possibile dire che non ci sono errori di ricezione e trasmissione.

È stato possibile verificare la corretta forma del segnale anche grazie all'utilizzo dell'oscilloscopio, vengono quindi analizzati i due segnali trasmessi dai dispositivi.

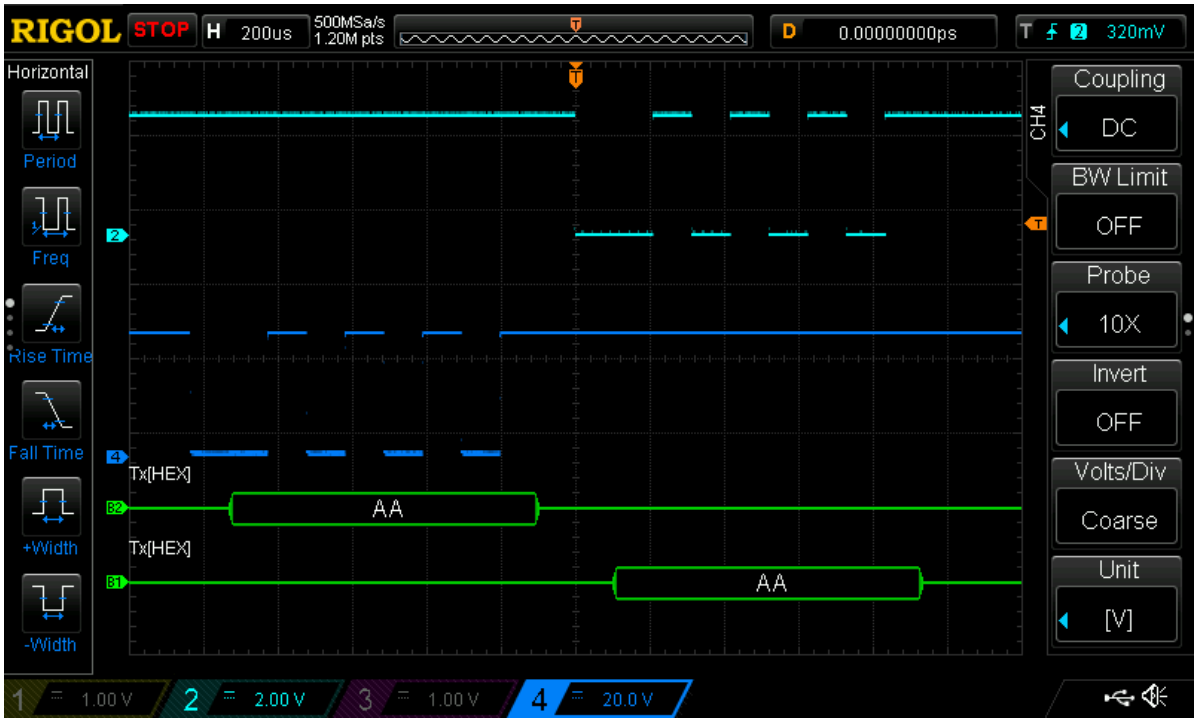

Figura 15 Trasmissione del microcontrollore ed eco del valore ricevuto dalla FPGA

In blu è rappresentato il segnale ricevuto dalla FPGA mentre in azzurro quello trasmesso. Come atteso i due segnali risultano essere identici.

Come ultima verifica è necessario verificare che il baud rate del trasmettitore coincida con quello calcolato teoricamente. Viene quindi visualizzata con l'oscilloscopio la durata di un bit trasmesso

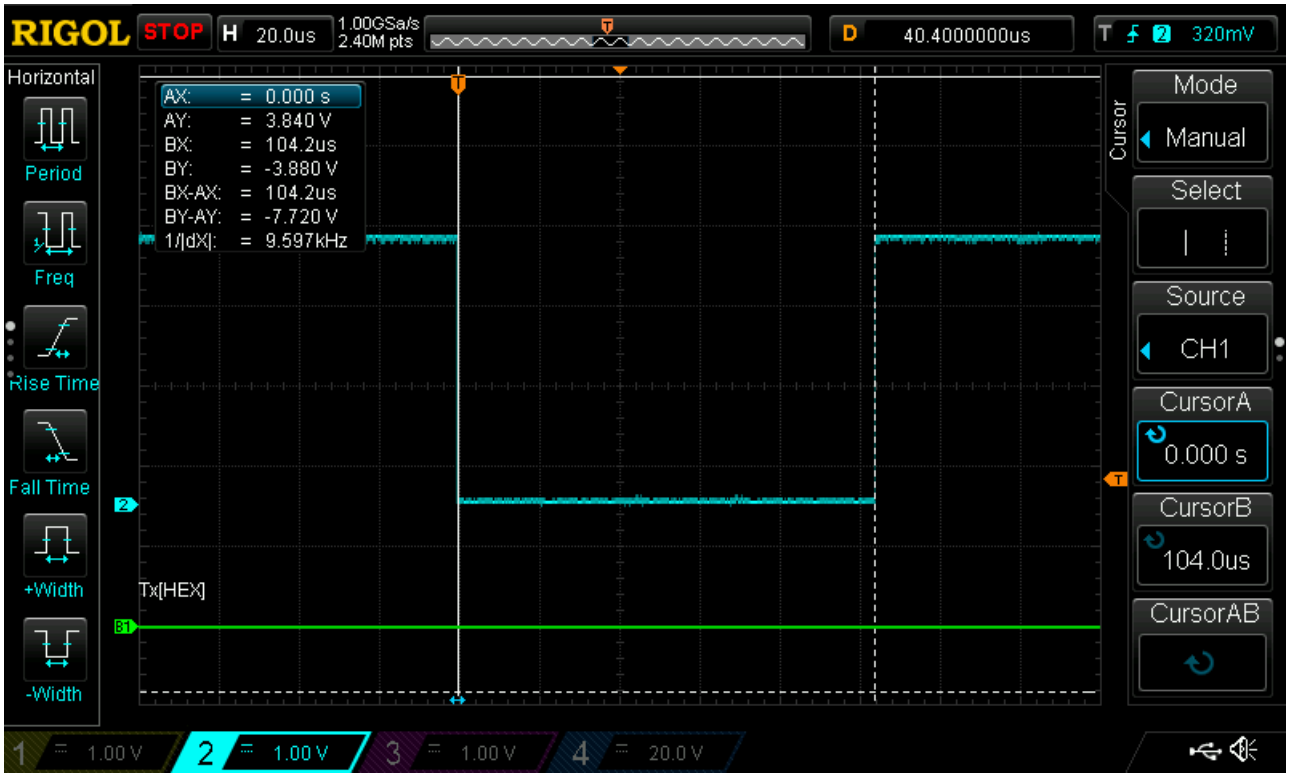

Figura 16 Misura del tempo di bit

Come ara stato precedentemente calcolato il baud rate ottenuto è circa 9597 baud.