Descrizione del problema

Il nostro progetto ha l’obiettivo di studiare l’interfaccia seriale asincrona UART, in particolare gli aspetti legati alla sua composizione, alla costruzione, al suo utilizzo e funzionamento.

Attraverso la realizzazione di un unico componente, che può svolgere sia il ruolo di trasmettitore che di ricevitore, sperimenteremo entrambe le sue proprietà effettuando dei test su due dispositivi identici.

Ogni dispositivo ha una porta di ingresso e una porta di uscita dei dati lungo una certa direzione.

Istanziato, quindi, lo stesso componente due volte testeremo la sua capacità di trasmissione e ricezione prendendo un dato dal bus d’ ingresso dell’interfaccia di uno e trasmettendolo all’altro, che lo riceverà e lo porterà sul proprio bus d’uscita.

Se identifichiamo un device con la lettera A e l’altro con la lettera B, può avvenire contemporaneamente la trasmissione/ricezione di un dato in direzione A🡪B e la trasmissione/ricezione di un altro dato in direzione B🡪A.

Descrizione dell’interfaccia

L’interfaccia UART (Universal Asinchronous Receiver Transmitter) è un sistema che permette la comunicazione tra dispositivi, enormemente utilizzato ancora oggi negli apparati industriali.

Per comunicare sono necessari due dispositivi UART, uno utilizzato come trasmettitore dei dati e l’altro come ricevitore. Questo è possibile grazie al fatto che l’interfaccia è Full-Duplex: esiste una porta Tx da cui parte una linea per la trasmissione e una porta Rx collegata a una linea per la ricezione dei dati (ovvero una linea di trasmissione di un’altra interfaccia UART associata). Lo scambio dei dati, quindi, è bidirezionale e può avvenire contemporaneamente.

Il meccanismo UART, nello specifico, funziona in questo modo:

* Per il ruolo di trasmettitore, l’interfaccia presenta una porta a cui è collegato un Data-Bus di 8 bit. I bit arriveranno in parallelo a un serializzatore che avrà il compito di permettere la trasmissione di un bit alla volta attraverso la linea, uscente dalla porta Tx. I bit sono inviati in ordine partendo dal MSB (Most Significant Bit) fino al LSB (Least Significant Bit), oppure viceversa. In questo senso si intende che la comunicazione è seriale.
* Dal punto di vista della ricezione, la nostra interfaccia rileva i dati trasmessi da un altro dispositivo sulla porta Rx e attraverso un deserializzatore, li porterà in parallelo su un Data-Bus collegato a una porta di uscita.

Abbiamo definito 4 porte fondamentali sull’interfaccia. A queste se ne aggiungono altre. Sono necessari:

* Un segnale START (1 bit) in ingresso per permettere la trasmissione.
* Un segnale READY (1 bit) in uscita per segnalare che è stato ricevuto correttamente un byte.
* I segnali RTS (Request-To-Send) in uscita e CTS (Clear-To-Send) in entrata. RTS è collegato al CTS di un altro device e, dualmemte, CTS a RTS. Entrambi sono di 1 bit e implementano l’Hardware Flow Control, un meccanismo che permette al trasmettitore e al ricevitore di notificare reciprocamente il proprio stato, in modo da garantire uno scambio affidabile dei dati.
* Un segnale di Clock (1 bit) in ingresso.
* Un segnale di Reset (1bit) in ingresso.
* Un segnale per indicare la frequenza di funzionamento del dispositivo (baudrate o bit/s). Tipicamente le frequenze sono otto (9600, 19200, 38400, 57600, 115200, 230400, 460800 e 921600) per cui questo il segnale in entrata sarà composto da 3 bit.

Inizialmente la linea di trasmissione ha valore logico 1, i bit di START e READY sono a 0 e così anche i segnali RTS e CTS. Quando si verifica l’evento START = 1, il trasmettitore solleva RTS per chiedere se sia possibile inviare dati serializzati. Se il destinatario è in grado di ricevere, alzerà la linea CTS (del lato trasmettitore). Adesso è possibile trasferire i dati. Il sistema deve poter interrompere o eseguire la trasmissione o la ricezione a seconda del valore di questi segnali.

A questo punto il trasmettitore abbassa la linea di trasmissione a 0 e aspetta un tempo di bit (il tempo di bit è l’inverso della frequenza di funzionamento fissata). Questa operazione equivale a inoltrare uno start bit, ovvero informare il ricevitore che sta per iniziare la trasmissione dei dati vera e propria.

Vengono poi trasmessi 8 bit, seguiti da un bit di stop a valore logico 1.

La trasmissione avviene sempre per blocchi di 8 bit che, a seconda delle impostazioni possono essere 7 bit di dati più un bit di parità oppure 8 bit di dati, senza parità. In totale, con bit di start e stop, sono trasmessi 10 bit.

Sullo start bit il ricevitore si mette in ascolto, ma in che modo?

Poichè la trasmissione è asincrona, il trasmettitore può inviare dati in qualsiasi momento. I fronti di transizione dei bit non sono allineati ad un clock condiviso tra trasmettitore e ricevitore ma avvengono ad una frequenza specifica tra le otto selezionabili. I due dispositivi conoscono il baudrate con cui possono comunicare, il problema è che il segnale di clock di uno e il segnale di clock dell’altro sono sfasati.

Si possono generare degli errori di trasmissione legati a questo sfasamento.

In certe applicazioni le linee elettriche su cui il segnale viaggia sono spesso molto lunghe e gli ambienti possono essere disturbati. Una conseguenza di ciò è il fatto che le transizioni dei bit non sono perfettamente verticali ma delle onde, non si verificano dei fronti verticali netti e puliti.

L’ideale per il ricevitore sarebbe campionare il segnale d’ingresso a metà del tempo di bit.

Il metodo migliore per soddisfare tale richiesta è quello del sovracampionamento: il clock del ricevitore viene impostato a una velocità che è multipla di quella del clock di trasmissione (solitamente 8 o 16 volte più grande). Grazie a ciò è possibile identificare in maniera abbastanza approssimata il momento del fronte di discesa relativo allo start bit, quando il trasmettitore abbassa la linea da 0 a 1.

Da qui, dopo vengo contati 4 cicli, il clock del ricevitore si troverà sincronizzato al centro della trasmissione dello start bit. Questo è il punto di partenza. Ogni 8 clock il ricevitore sarà in grado, così, di campionare i bit al centro del loro tempo di trasmissione, con una buona approssimazione. Dopo 8 bit di dati campionati il ricevitore si aspetterà uno stop bit.

Poco prima che il dispositivo ricevente diventi indisponibile (quando il suo buffer di ricezione è quasi pieno), questo deasserisce il segnale RTS, che viene rilevato sul CTS del trasmittore, il quale interrompe immediatamente la trasmissione.

Il byte è stato quindi trasferito e verrà lanciato l’evento READY = 1 per segnalare che i dati sono stati correttamente inseriti nel bus di uscita.

A questo punto i dispositivi tornano alla propria configurazione iniziale.