

iCE65 as IrDA Tranceiver

Overview

IrDA devices provide a walk-up, point-to-point method of data transfer that is adaptable to a broad range of computing and communicating devices. IR technologies are better suited for short distance, low-to-medium data throughput, wireless communication channels. This design example illustrates the implementation of an IrDA Tranceiver using SiliconBlue iCE65 FPGAs.

Description

IrDA data communications operate in half-duplex mode because while transmitting, a device's receiver is blinded by the light of its own transmitter, and thus, full-duplex communication is not feasible. The data in the transmitter design is first encoded before being transmitted as IR pulses using NRZ (non return to zero) encoding. NRZ encoded outputs do not transition during the bit period, and may remain high or low for consecutive bit periods. This is not an efficient method for IR data transmission with LEDs. To limit the power consumption of the LED, IrDA requires pulsing the LED in a RZI (return to zero inverted) modulation scheme so that the peak power to average power ratio can be increased. IrDA requires the minimum pulse width to be 4/16th of the bit period. A 16x clock is required, and counting three clock cycles can easily be done to encode the transmitted data.

The IrDA link cannot send and receive data at the same time. The IrDA link is a half-duplex interface and a time delay must be allowed from when a link stops transmitting until it can receive data again. A time period with duration of 10 ms must be allowed between transmitting and receiving data.

Implementation

The figure 1 illustrates the high level block diagram of IrDA communication module. It essentially consists of two parts – Transmit Encoder and Receive Decoder. This design example uses 4/16 NRZ format for encoder/decoder. Figure 2 and Figure 3 shows sample 4/16 NRZ data encoding and decoding of this design example.

Transmit Encoder

The input for the transmit encoder is in NRZ form format. NRZ encoded output do not change its transition during bit period and output can be high or low for consecutive bit periods. This is not an efficient method as power consumption on LED is more. To limit it, IrDA needs to convert the pulses in RZ (return to zero) modulation so that average power to peak power will be increased. IrDA requires minimum pulse width of 4/16 clock cycles. Hence, a 16x clock cycle is needed and counting four cycles can be done to encode the transmitted data.

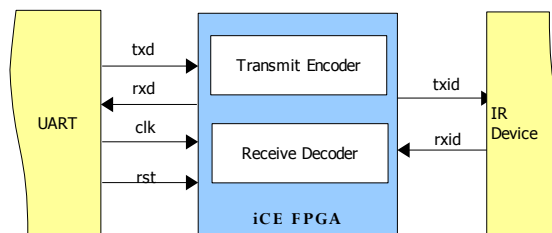


Fig 1: Block diagram of IrDA system

Receive Decoder

The input for the receive decoder is in RZ (Return to zero) form. IrDA demodulates the 4/16 IR_RXD signal to its 16 clock cycle NRZ bit form and sends it to microcontroller or microprocessor through UART or RS-232. Both encoded and decoded signals are available after 1 clock cycle delay.

Table 1 lists the pin description of this design example. Table 2 summarizes the post P&R resource utilization summary of this Design Example when implemented using iCE65 FPGA.

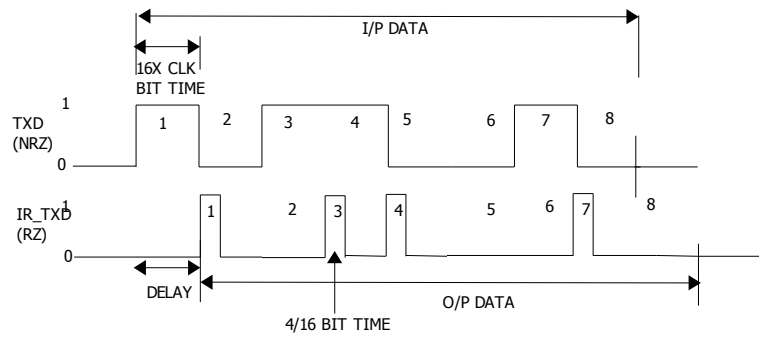


Fig 2: IrDA 4/16 Data Encoding

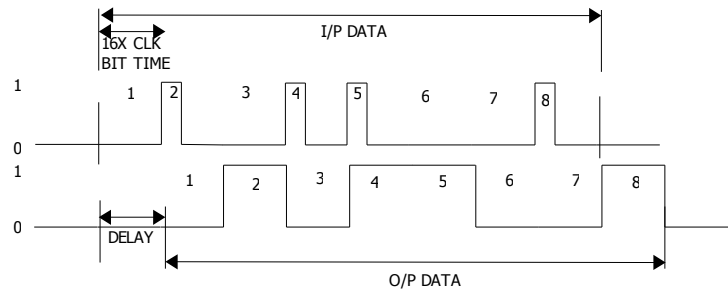


Fig 3: IrDA 4/16 Data Decoding

Pin	Direction	Description
clk	Input	System clock
rst	Input	Active low system reset
txd	Input	Input for transmit encoder
rxid	Input	Input for receive decoder
txid	Output	IrDA input after encoding
rxid	Output	IrDA output - to be decoded

Table 1: Pin Description

Device	Logic Cells	IO Cells
iCE65L04-UCB284	81	6

Table 2: Resource Utilization

Conclusion

This design example demonstrates the implementation of a IrDA Tranceiver using iCE FPGAs. iCE FPGA's very low power capabilities makes iCE FPGAs an obvious choice for implementing a IrDA encoder/decoder for battery operated compact/handheld devices like PDAs, cellular phones etc..

Disclaimer

SiliconBlue is providing this document, design example, or information "as is." SiliconBlue does not ensure that this implementation or information is void of any claims of infringement. As a reader / implementer, you are responsible for obtaining any rights you may require for your implementation. SiliconBlue also does not warranty the fitness of this design example to be readily implemented in any specific context. It is your sole responsibility to validate this design example and its correctness while implementing it for your requirement.