

## Overview

The IrDA creates an interoperable, low cost, low power, half-duplex serial data interconnection standard that supports a walk-up, point-to-point user model that is adaptable to a wide range of appliances and devices. IR technologies are better suited for short distance, low-to-medium data throughput, and wireless communication channels. This design example illustrates the implementation of an IrDA Fast Transmitter using SiliconBlue iCE65 FPGAs.

## Features Supported

- Hierarchical HDL Design for simple user modification
- Complete 4PPM Packet Support – Preamble, STA, Data, STO fields
- Configurable data block length
- Synchronous Parallel Processor Interface
- Read and Write Data FIFOs of 32x32-bit
- 8 configurable baud rates
- Interrupt on completion of transmission of data
- IEEE 802 CRC32 generation and detection
- VHDL RTL, testbench and Modelsim script for simulation
- IP-XACT version 1.2 compliant

## Features not supported

- Host UART Interface
- Not Hardware tested

## Resource Utilization

*Table 1: Resource Utilization*

LUTs	Registers	Memory	GBs	I/Os
739	269	2	0	0

Note: Resource Utilization is based on iCECube 2010.12.14671 release.

## System Block Diagram

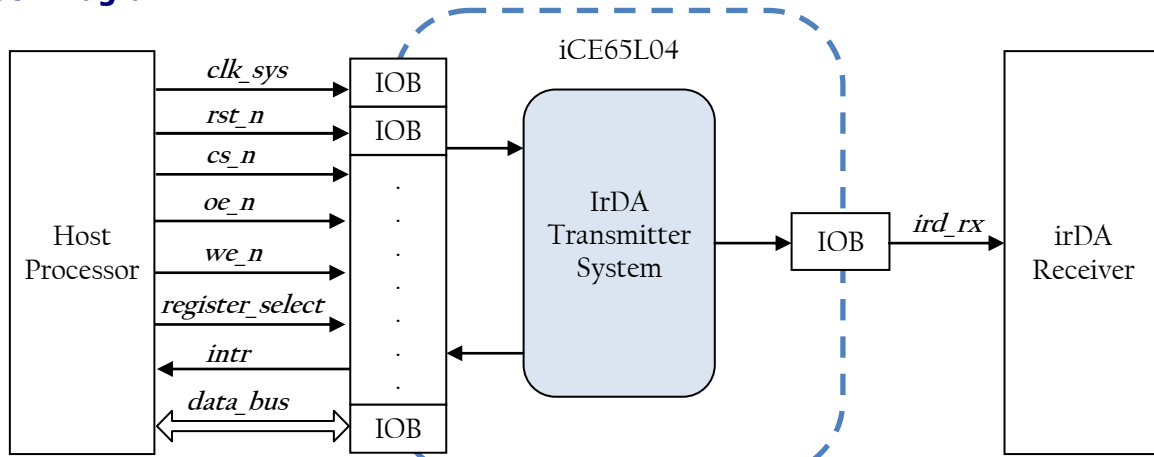


Figure 1: System Block Diagram

## Functional Block Diagram

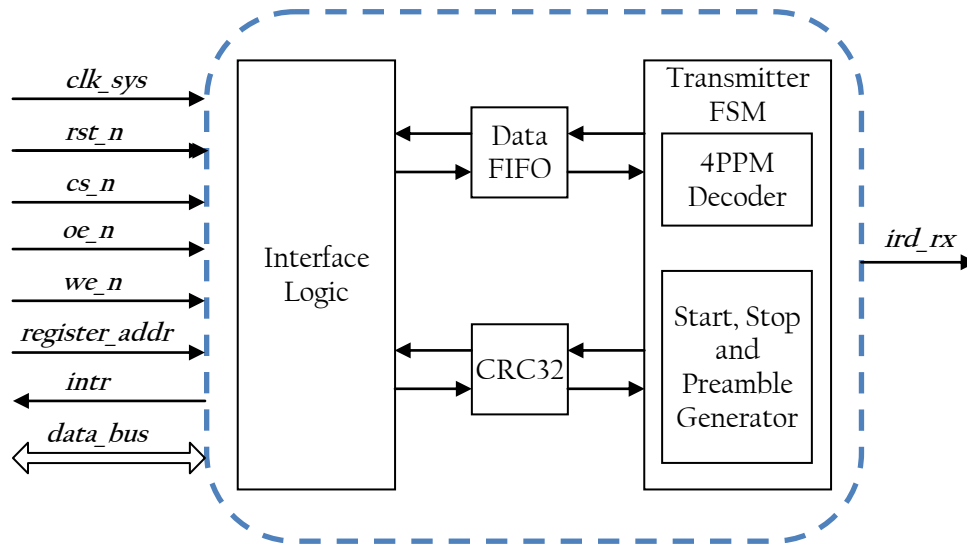


Figure 2: Functional Block Diagram

## Design Interface

*Table 2: Pin Description*

Signal Name	Pin Type	Signal Description
<b>cs_n</b>	Input	Active low chip select.
<b>oe_n</b>	Input	Active low output enable.
<b>we_n</b>	Input	Active low write enable.
<b>register_addr [1:0]</b>	Input	Register selection bus to select registers from – control register, status register
<b>intr</b>	Output	IrDA Transmit interrupt.
<b>data_bus [31:0]</b>	Inout	Processor data bus.
<b>ird_rx</b>	Output	IR device data output.
<b>rst_n</b>	Input	Asynchronous active low reset. This signal is used to initialize the internal state machine to a known state.
<b>clk_sys</b>	Input	System clock.

## Configurable Parameters

None

## Register Map

*Table 3: Register Address Map*

Registers	Address
<b>Configuration Register</b>	0x1
<b>Read (Tx) data FIFO Register</b>	0x2
<b>Status Register</b>	0x3

## Design Details

IrDA defines a set of specifications, or protocol stack, that provides for the establishment and maintenance of a link so that error free communication is possible.

The IrDA Standards include three mandatory specifications: the Physical Layer, Link Access Protocol (IrLAP), and Link Management Protocol (IrLMP). Beginning with Version 1.1 of the IrDA physical layer specification, a 4 Mbps data rate is supported. The FIR(4 Mbps) scheme uses the four pulse position modulation (4PPM) scheme, where one complete symbol is represented by 4 equal time slices called “chips”. In FIR every chip has a 125ns duration, and every symbol represents 2 bits of data. Because there are four unique chip positions within each symbol in 4PPM, four independent symbols exist in which only one chip is logically a “one” while all other chips are logically a “zero.” We define these four unique symbols to be the only legal data symbols (DD) allowed in 4PPM. Each DD represents two bits of payload data, or a single “data bit pair (DBP)”, so that a byte of payload data can be represented by four DDs in sequence.

Figure 3 illustrates FIR(4PPM) modulation/demodulation schemes.

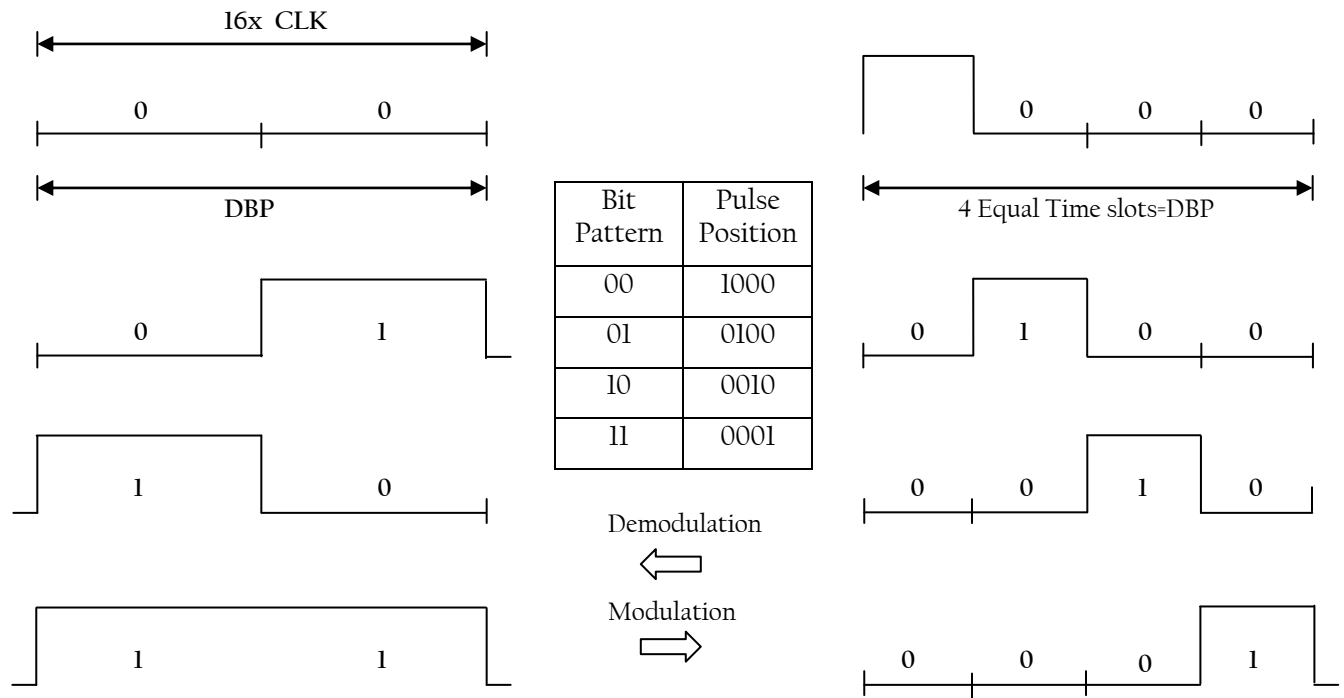


Figure 3: FIR(4PPM) modulation/demodulation schemes

In IrDA FIR mode, a special packet format is used for data transfer. The packet format is divided into 4 blocks as shown in Table 4.

1. PA: Preamble-consists of exactly 16 repeated transmission of following symbols (“1000 0000 1010 1000”).
2. STA (Start Sequence): It occurs only once and has the the following symbols(“0000 0011 0000 0011 0110 0000 0110 0000”). After the preamble, the receiver looks for STA sequence for synchronization.
3. DD (Actual Data encoded in 4ppm scheme along with CRC32 bit): After the start sequence, the transmitter transmits the data iin encoded form along with CRC32 to facilitate error detection at receiver end.  

$$CRC(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$
4. STO (Stop Sequence): It also occurs once and has the sequence(“0000 0011 0000 0011 0000 0110 0000 0110”), which indicates the end of a data frame.

Transmitter uses 2-bit address select signals for programming control registers, reading/writing data and reading status register. Table 3 lists the address mapping of Transmitter. These register address along with active low output enable(oe\_n) and write enable(we\_n) configures the read/write mode for the register addressed.

Table 4: IrDA Packet Format

Preamble	STA	DD along with CRC32	STO
----------	-----	---------------------	-----

Transmitter starts data transaction as soon as TXSTART bit asserted. Transmit data are sliced in to 32-bit length. Configuration register bits {0 to 3} specify length of data transaction in multiples of 32-bit data block. Example: “1000” specifies data length of 8 such 32-bit words. Tx Write FIFO is of 32x32-bit each. This allows data transaction of maximum of 1024 bits of data per packet.

### Transmitter Configuration Register:

- TXSTART : Active High, Start data transaction
- INTRCLR : Clear Interrupt
- INTREN : Enable Interrupt
- FIFORST – Reset FIFO

- BLKLEN{4:0}: Block Length
- CFG BAUD [15 : 13] – Baud Rate

### Transmitter Status Register:

- TX BUSY : Transmitter Busy
- WR EMPTY/HFULL/FUL denote Read and Write FIFO empty, half full and full respectively.

Figure 4 illustrates the timing diagram of Write/Read Data/Configuration/Status register of IrDA Transmitter.

## Register Descriptions

### Transmitter configuration Register (Read/Write)

*Table 5: Receiver Configuration Register*

Offset: 0x1

	Bits 31-16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bits 8-5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Read:		CFG BAUD2	CFG BAUD1	CFG BAUD0	TX START	INTR CLR	INTR EN	FIFO RST		BLK LEN4	BLK LEN3	BLK LEN2	BLK LEN1	BLK LEN0
Write:														
Reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0

=Unimplemented

TXSTART - '1' indicates Start of data transaction

INTRCLR - '1' Clears the interrupts

INTREN - '1' Enables the interrupts

FIFORST - '1' Clears Read/Write FIFO

BLKLEN - Block length of data transaction

CFG BAUD – Baud Rate Selection (Please refer Table 6)

*Table 6: Configurable Baud Rates*

Baud Rate (bps)	Configuration Reg [15 : 13]
<b>115.2K</b>	000
<b>57.6K</b>	001
<b>38.4K</b>	010
<b>19.2K</b>	011
<b>9600</b>	100
<b>4800</b>	101
<b>2400</b>	110
<b>1200</b>	111

## Transmitter Status Register (Read Only)

Table 7: Receiver Status Register

Offset:	0x3				
	Bits 31-4	Bit 3	Bit 2	Bit 1	Bit 0
Read:		WRFULL	WRHFULL	WREMPY	TXBUSY
Write:					
Reset:	0	0	0	1	0
	=Unimplemented				

WRFULL - '1' indicates Write FIFO Full

WRHFULL - '1' indicates Write FIFO Half Full

WREMPY - '1' indicates Write FIFO Empty

TXBUSY - '1' indicates Transmitter is busy

## Initialization Conditions

A reset signal assertion is needed to initialize the transmitter to the proper operating state. Reset is Asynchronous active low reset. `ird_rx` (Output of the Transmitter) is set to 0 at reset.

## Timing Diagrams

Signals definition:

`clk_sys` - system clock

`cs_n` - Chip Select - Active low

`register_addr` - Register selection - to select registers from - control register, status register

`oe_n` - Output Enable - Active low

`we_n` - Write Enable - Active low

`data_bus` - Processor Data Bus

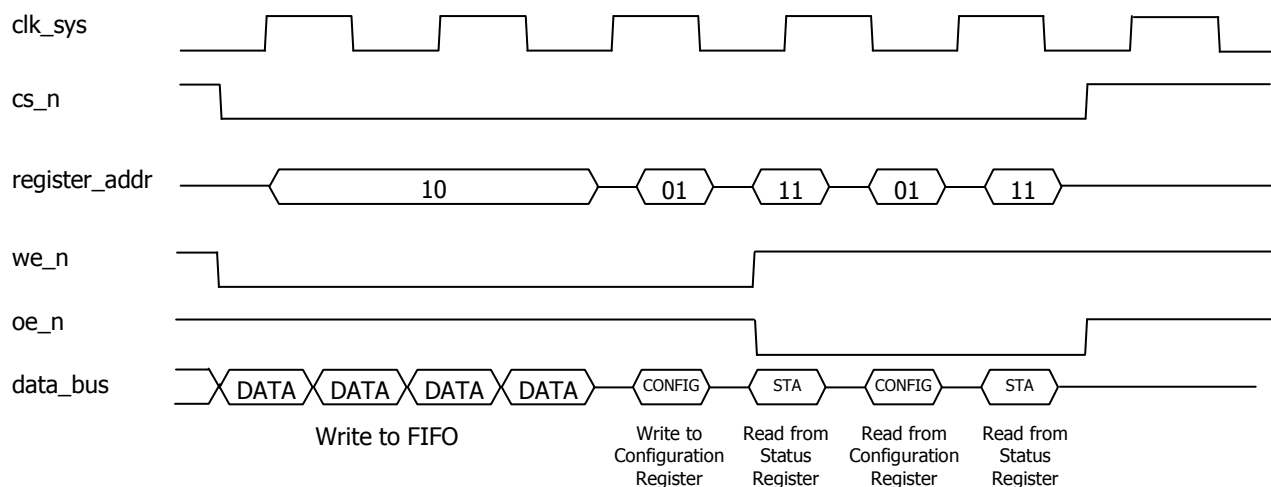
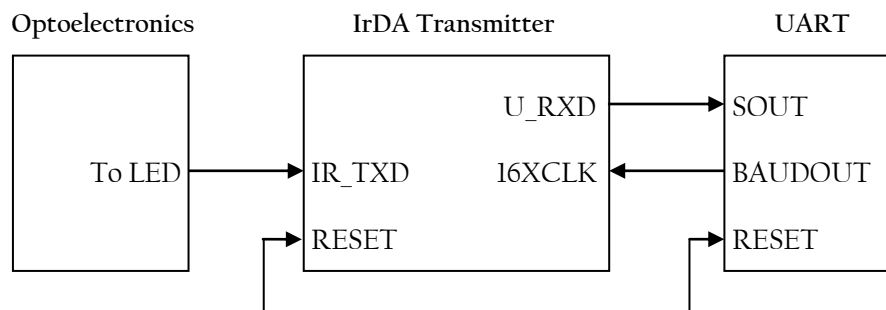


Figure 4: Timing diagram of Write/Read Data/Configuration/Status register of IrDA Transmitter

## Usage Example



- Transmitter is operational when reset is asserted.
- Write the data to be transmitted to Write FIFO by selecting the address = "10".
  - Sample data : X"0B0B1B0F", X"141BA41E" ....(In this case, 16 data words)
- Write to Transmitter Configuration Register by selecting address = "01", Make 10th bit = '1' to enable interrupt. Make 12th bit = '1' implies Start of Transmission
  - Give input "000000000000000000001010000010000" to data bus
- Read the status of transmitter by selecting the address "11" in between data transaction – irda\_busy signal will be High i.e. the LSB of the data bus should be '1'.
  - Content of data bus : X"00000005"
- Wait until Transmitter Interrupt to go HIGH indicating transmission of data is complete.
- Clear the Interrupt by writing x"00000800" i.e. (11th bit should be '1')
- Read the Status register of transmitter after data transmission by selecting the address = "11" - irda\_busy signal will go Low.
  - Content of data bus : X"00000000"

## System Designer Flow

IrDA Fast Transmitter is compatible with System Designer/IP-XACT 1.2. The System Design flow is as follows,

1. Launch the System Designer from Synplify Pro using menu 'Import -> Launch System Designer'.
2. Create a new project(open an existing old project, as necessary) and import the IP-XACT XML file
3. Drag and place the component from the 'Library' pane to the 'Design' pane
4. Click on the "Generate Files" button, which generates the necessary files required for synthesis and simulation.
5. Go to Synplify Pro and click on the "Run" button to synthesize the System Designer generated files. Synplify Pro generates all the necessary files for P&R in iCECube.

## References

The following references were used in the creation of this design:

- SiliconBlue Technologies, Inc. “[iCE65 Ultra Low-Power mobileFPGA Family](#)” datasheet (26-MAY-2010)
- [http://en.wikipedia.org/wiki/Infrared\\_Data\\_Association](http://en.wikipedia.org/wiki/Infrared_Data_Association)

## Revision History

Version	Date	Description
<b>1.0</b>	09-SEP-2010	Initial Draft Document
<b>1.1</b>	04-DEC-2010	IP-XACT format update

## Disclaimer

Copyright © 2007–2009 by SiliconBlue Technologies LTD. All rights reserved. SiliconBlue is a registered trademark of SiliconBlue Technologies LTD in the United States. Specific device designations, and all other words and logos that are identified as trademarks are, unless noted otherwise, the trademarks of SiliconBlue Technologies LTD. All other product or service names are the property of their respective holders. SiliconBlue products are protected under numerous United States and foreign patents and pending applications, maskwork rights, and copyrights. SiliconBlue warrants performance of its semiconductor products to current specifications in accordance with SiliconBlue's standard warranty, but reserves the right to make changes to any products and services at any time without notice. SiliconBlue assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by SiliconBlue Technologies LTD. SiliconBlue customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.



SiliconBlue Technologies Corporation

3255 Scott Blvd.,  
Building 7, Suite 101  
Santa Clara, CA 95054

Tel: 408-727-6101  
Fax: 408-727-6085

[www.SiliconBlueTech.com](http://www.SiliconBlueTech.com)