

iCE65 FPGA as an IrDA Fast Receiver



04-DEC-2010 (1.1)

Design Example #007

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 Receiver using SiliconBlue iCE65 FPGAs.

Features Supported

- Hierarchical HDL Design for simple user modification
- 4 Mbps Baud rate
- Complete 4PPM Packet Support – Preamble, STA, Data, STO fields
- Configurable data block length
- Synchronous Parallel Processor Interface
- Read and Write Data FIFOs of 5x32-bit
- Interrupt on completion of reception 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

- Configurable baud rate
- Host UART Interface
- Not Hardware tested

Resource Utilization

Table 1: Resource Utilization

LUTs	Registers	Memory	GBs	I/Os
612	242	2	0	0

Note: Resource Utilization is based on iCEcube 2010.12.14671 release.

System Block Diagram

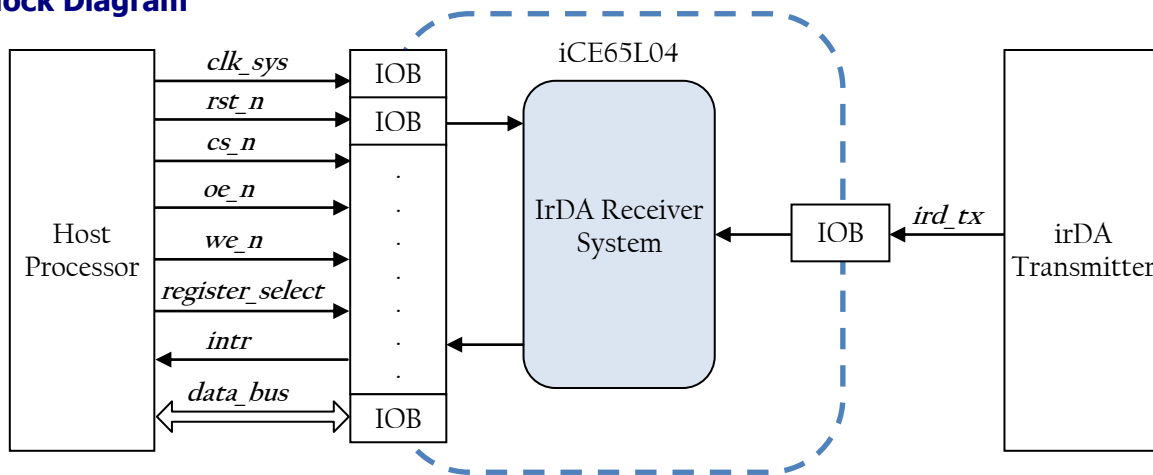


Figure 1: System Block Diagram

Functional Block Diagram

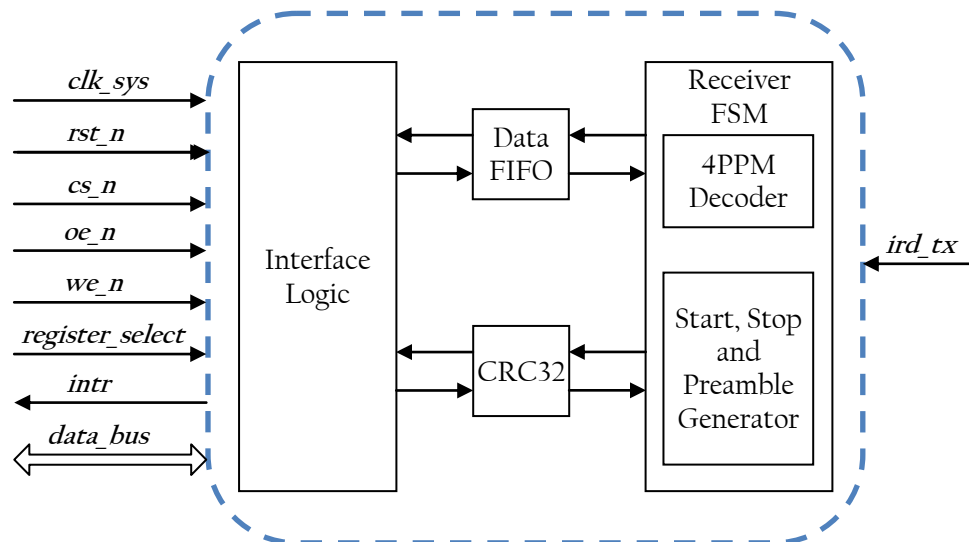


Figure 2: Functional Block Diagram

Design Interface

Table 2: Pin Description

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

Configurable Parameters

None

Register Map

Table 3: Register Address Map

Registers	Address
Configuration Register	0x1
Read (Rx) data FIFO Register	0x2
Status Register	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.

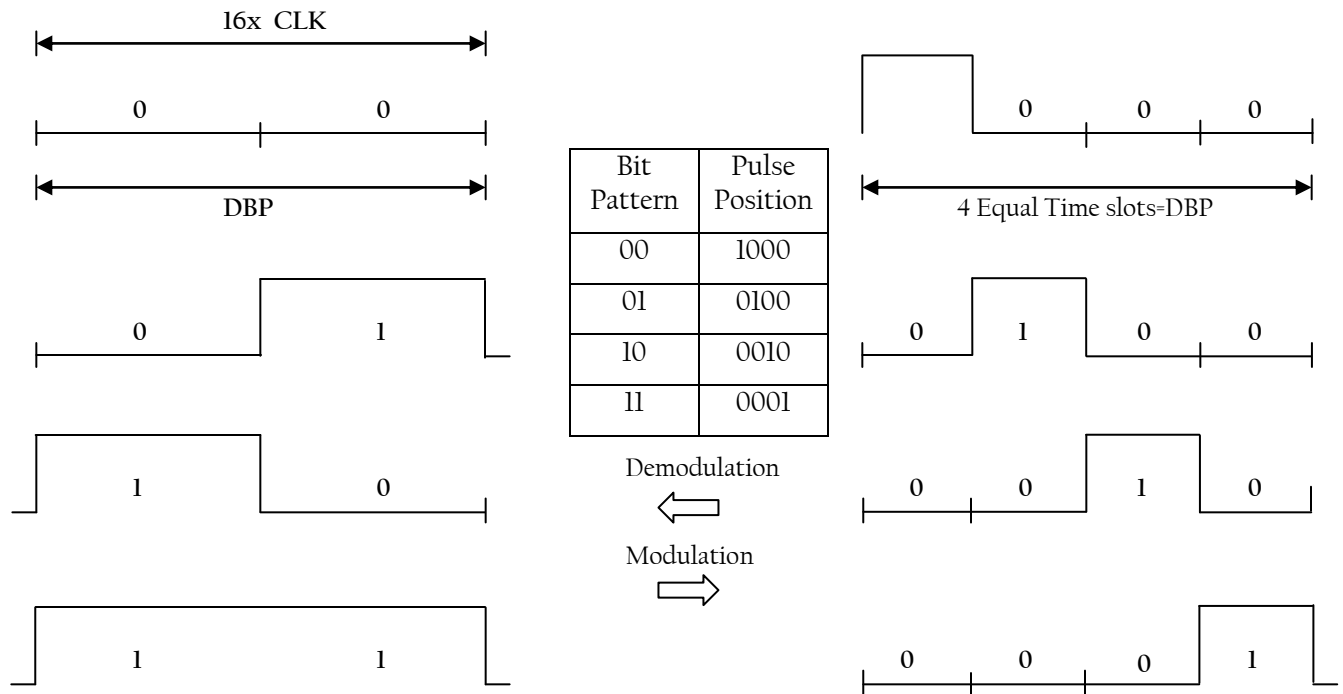


Figure 3: FIR(4PPM) modulation/demodulation schemes

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.

$$CRC(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

In receiver, the data gets decoded only after the STA flag is raised.

4. STO (Stop Sequence): It also occurs once and has the the sequence(“0000 0011 0000 0011 0000 0110 0000 0110”), which indicates the end of a data frame.

Receiver uses 2-bit address select signals for programming control registers, reading/writing data and reading status register. Table 3 lists the address mapping of Receiver. 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
----------	-----	---------------------	-----

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. Rx Read FIFO is of 32x32-bit. This allows data transaction of maximum of 1024 bits of data per packet.

Receiver Configuration Register:

- INTRCLR : Clear Interrupt
- INTREN : Enable Interrupt
- FIFORST – Reset FIFO

- BLKLEN{4:0}: Block Length

Receiver Status Register:

- RXBUSY : Receiver Busy
- RD EMPTY/HFULL/FULL denote Read FIFO empty, half full and full respectively.
- PAERR, STAERR, STOERR and CRCERR denote Preamble Error, Start Error, Stop Error and CRC error respectively.

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

Register Descriptions

Receiver configuration Register (Read/Write)

Table 5: Receiver Configuration Register

Offset: 0x1

	Bits 31-12	Bit 11	Bit 10	Bit 9	Bits 8-5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Read:		INTR CLR	INTR EN	FIFO RST		BLK LEN4	BLK LEN3	BLK LEN2	BLK LEN1	BLK LEN0
Write:		INTR CLR	INTR EN	FIFO RST		BLK LEN4	BLK LEN3	BLK LEN2	BLK LEN1	BLK LEN0
Reset:	0	0	0	0	0	0	0	0	0	0

=Unimplemented

INTRCLR - '1' Clears the interrupts

INTREN - '1' Enables the interrupts

FIFORST - '1' Clears Read/Write FIFO

BLKLEN - Block length of data transaction

Receiver Status Register (Read Only)

Table 6: Receiver Status Register

Offset: 0x2

	Bits 31-8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Read:		PA ERR	STA ERR	STO ERR	CRC ERR	RD FULL	RDHFULL	RDEMPY	RX BUSY
Write:									
Reset:	0	0	0	0		0	0	1	0

=Unimplemented

PAERR - Preamble Error

STAERR - Start Error

STOERR - Stop Error

CRCERR - CRC Error

RDFULL - '1' indicates Read FIFO Full

RDHFULL - '1' indicates Read FIFO Half Full

RDEMPY - '1' indicates Read FIFO Empty

RXBUSY - '1' indicates Receiver is busy

- Write the data to be transmitted to `irda_tx` line bit by bit. The overall data consists of 5 sections
 - Preamble bits – `x"1501"` is sent 16 times (LSB first)
 - Start bits – `x"06063030"` is sent once (LSB first)
 - Modulated Data bits – 32 32-bit words are sent
 - CRC bits – 64-bit CRC is sent
 - Stop bits – `x"60603030"` is sent (LSB first)
- Wait until the interrupt is high (signifying the complete reception of the data). Clear the interrupt by asserting the `INTRCLR` bit (Refer Table 5)
- Read the data from RX FIFO by selecting the address = "11", and check if it matches the reference data

System Designer Flow

IrDA Fast Receiver is compatible with System Designer/IP-XACT 1.2. The System Designer 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, click on the "Run" button to synthesize the System Designer generated files. Synplify Pro generates all the necessary files for P&R in iCECube2.

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

Revision History

Version	Date	Description
1.0	09-SEP-2010	Initial Draft Document
1.1	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