XAPP358 (v1.1) May 18, 2001

Wireless Transceiver for the CoolRunner CPLD

Summary

This document focuses on the design of a wireless transceiver using an XPLA3 CoolRunner CPLD. The wireless transceiver is implemented using the CoolRunner™ XPLA3™ demo board from Insight Electronics. The wireless transceiver is the perfect application of the low power capabilities of a CoolRunner CPLD. To obtain the VHDL code described below go to the section titled "VHDL Disclaimer and Download Instructions" on page 11.

Introduction

A wireless transceiver consists of two modules; receive and transmit. One CoolRunner demo board comprises the receive portion while the second demo board comprises the transmit portion. The design transmits the text string "CooLrunnEr," which is displayed on both the transmit and receive demo boards. The wireless communication is controlled by an RF module designed by RF Monothilics, Inc. (RFM®).

The protocol designed for the wireless transceiver obeys a custom wireless communication protocol. A designer could change the protocol has needed to meet the needs of a specific application.

The addition of keyboard control is also covered in this document. The VHDL code is not provided for this portion of the design. With keyboard control, a user can enter a text string into the transmitter and the string would be display on the receive side of the transceiver. The keyboard described is manufactured by Fujitsu Takamisawa America, Inc. (FBK7603) (http://www. fujitsu.takmisawa.com/pdf/EvalKits.pdf).

Figure 1: **CoolRunner Wireless Transceiver**

CoolRunner CPLD Transceiver Operation

This section describes the operation of the transceiver. The communication protocol is a custom transmit and receive scheme, using Manchester encoding and Bit-Oriented Protocol (BOP) theory.

Communication Protocol

The communication protocol is show in Figure 2. The preamble and postamble are used to contain the data to be transmitted. The total transmission is 36 bits. For error checking, the data is transmitted four times and compared to insure the proper data was received.

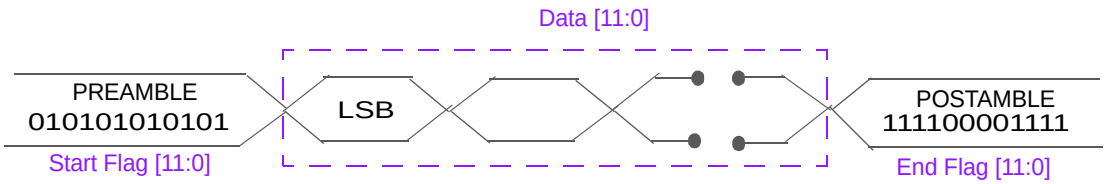

Figure 2: **Communication Protocol**

Transmit

A Manchester encoding scheme is used between the transmit and receive modules. Manchester coding ensures that each bit of the data is D.C. balanced. Also, this coding scheme provides an edge within each bit period that can be used to align the receiver's clock if needed. However, Manchester coding requires twice the bandwidth as compared to NRZ (Non-Return-to-Zero) codes. To decrease bandwidth, a symbol table is used. It consist of sixteen different symbols that can be generated using six bits which guarantees that no more than four consecutive bits are the same. This scheme requires only 1.5 times the bandwidth when compared with NRZ coding. For more information on Manchester and NRZ coding schemes, refer to the application note XAPP339 "Manchester Encoder-Decoder for Xilinx CPLDs" (http://www.xilinx.com/xapp/xapp339.pdf).

Receive

BOP is utilized on the receive side of the transceiver. BOP takes advantage of opening and closing flag insertion and deletion and zero bit insertion and deletion. Once an edge is detected, the incoming data is sampled and stored in a shift register. Once the most significant bits are equal to the postamble, the 12 bit data is stored in a register. This process occurs four times. This insures the data has time to be displayed on the LCD of the CPLD demo board and allows for more accurate error checking.

CoolRunner CPLD Transceiver Block Diagram

Transmit

The transmit block diagram is shown in Figure 3. Transmission comprises of three VHDL entities; DISPLAY_COUNT, SHIFT_ENABLE, and SHIFT_OUT. These three logic modules are controlled by the top level module, TX_MODULE. DISPLAY_COUNT controls the LCD common line, LCDCOM, which minimizes charging in the LCD. DISPLAY_COUNT also controls the time between display states. Each state determines which two digits are displayed on the LCD. It pulses the SWITCH_EN_H signal when it is time to change to the next state. This control line tells the SHIFT_ENABLE module to output the next state number, CUR_STATE, to the CHANGE_STATE look up table. When this is completed, it pulses the LOAD_DATA_H signal to tell the SHIFT_OUT module to load the current state data, CUR_STATE_DATA, output by the CHANGE_STATE look up table. This module also keeps track of how many transmissions have been sent. It pulses the LOAD_DATA_H signal four times for each state, controlling the time between transmissions. The data is sent four times to provide error checking on the receive side (See Receive). When SHIFT_OUT observes that LOAD_DATA_H has been pulsed, it loads the current state data, and begins to send the data, with a preamble and postamble, one bit at a time, to the RF module.

The CONTROL signal is controlled by the TX MODULE which enables the RF MODULE to be in transmit mode. SYS_CLK_H and SYS_RST_L are external signals that are used as the system clock and the global system reset.

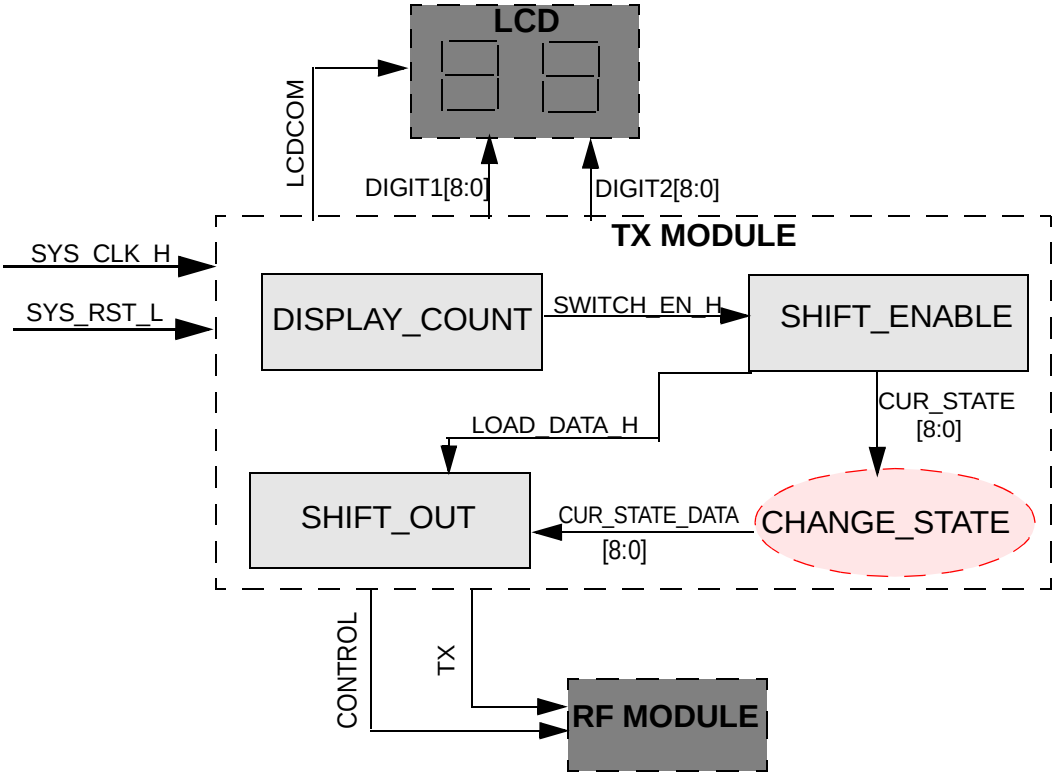

Figure 3: **Transmit Module Block Diagram**

Receive

The receive block diagram is shown in Figure 4. The data is read on the RX pin and shifted into a 3 bit shift register, RXIN, on every clock cycle. When an edge is detected (a logic 1) in the least significant bits of RXIN, a counter is enabled. This counter counts to approximately 3/4 of the bit period (due to non-ideal conditions, see Figure 5), samples the data, and shifts the bit into a 36-bit data register, SHIFT_DATA (see Figure 10). If there are consecutive bits in the stream, the counter continues to count 3/4 into the next bit period and samples the data again. If there is another edge detected, it restarts the counter, to keep the possibility of error due to drift to a minimum. Once the postamble is seen in the most significant 12 bits of the 36-bit shift register, the 12 bits of data are stored into a temporary register, REG1 through REG4, and the module gets ready for the next transmission. After the fourth transmission, if any two of the temporary registers are equal, the data is symbolized using the RX_SYMBOLIZE function, and the data is sent to the LCD.

LCDCOM minimizes charging in the LCD. The CONTROL signal is controlled by the receive MODULE which enables the RF MODULE to be in receive mode. SYS_CLK_H and SYS_RST_L are external signals that are used as the system clock and the global system reset.

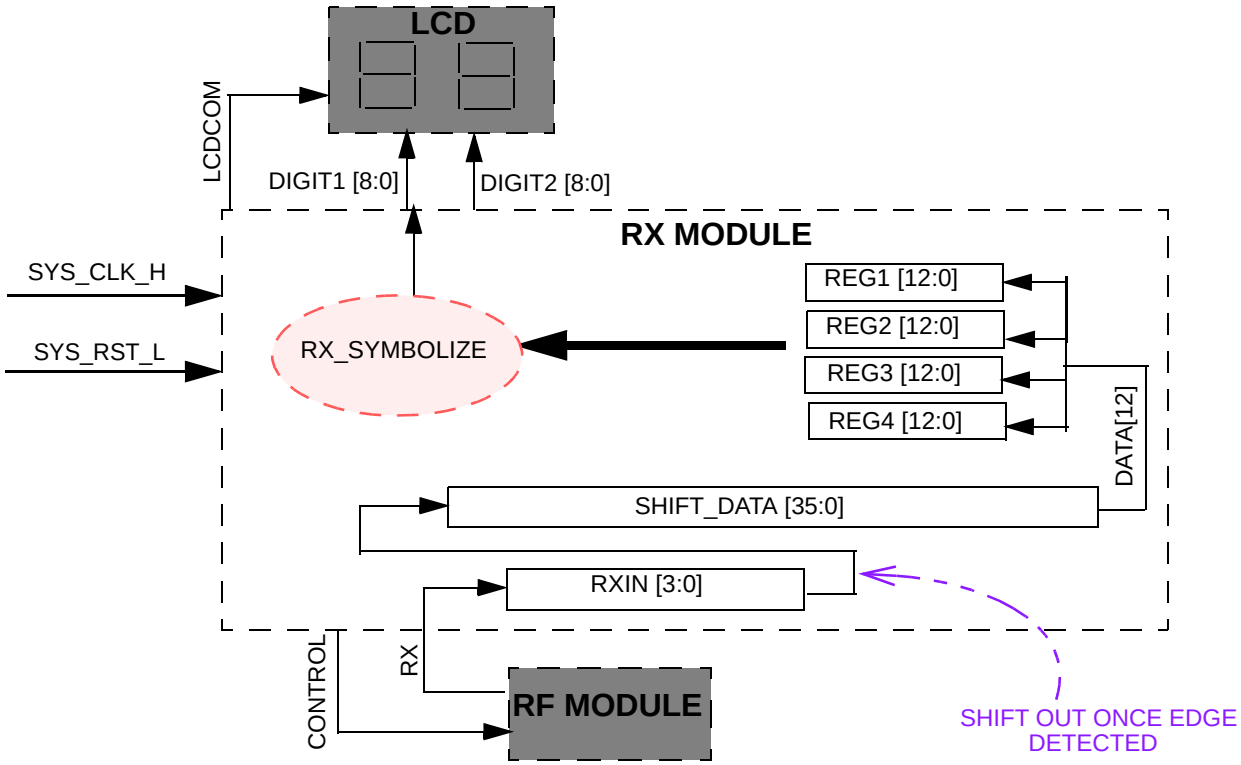

Figure 4: **Receive Module Block Diagram**

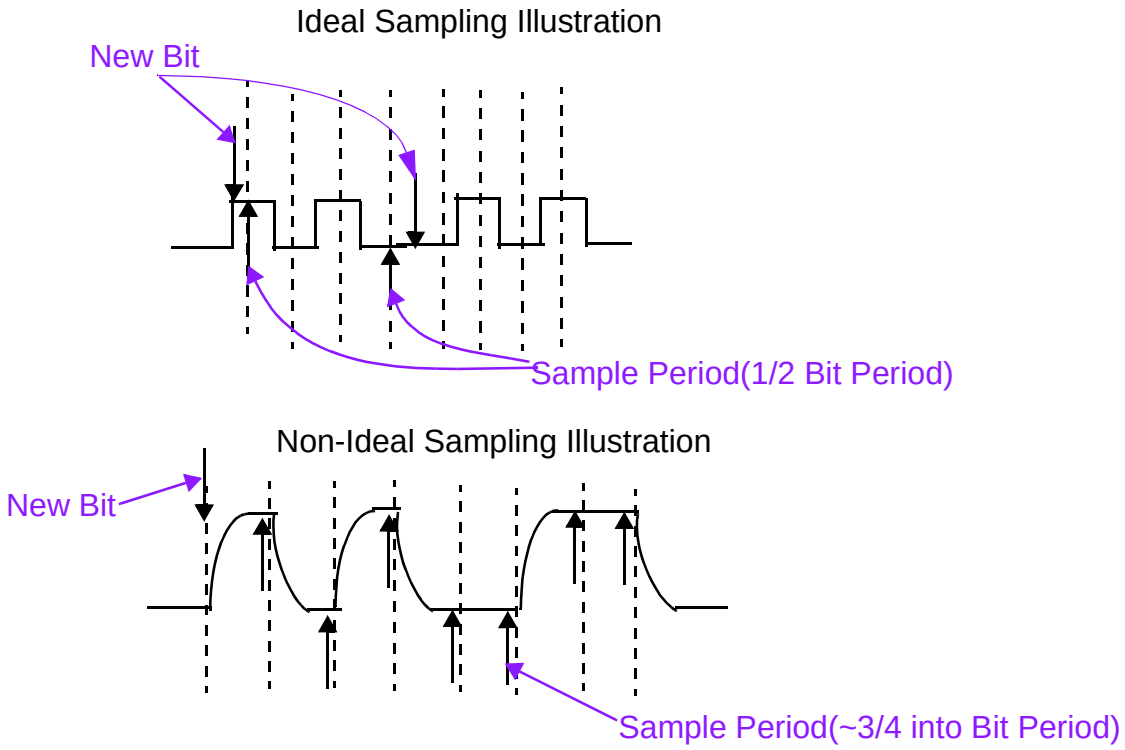

Figure 5: **Receive Timing Illustrations**

CPLD Transmit Design

Transmit module contains the look up tables: CHANGE_STATE, RX_SYMBOLIZE, BIN7SEG. The latter two are used to display the letters being transmitted. CHANGE_STATE changes the current state of TX_MODULE (the data to be transmitted), which is sent from the SHIFT_ENABLE logic module. The logic function RX_SYMBOLIZE is a look up table to convert 6-bits of each digit of data into a 4-bit number. BIN7SEG is a lookup table that takes the 4-bit symbolized number from the RX_SYMBOLIZE function and converts it into an 8-bit number sent to the LCD digits. The block diagram for TX_MODULE is shown in Figure 6.

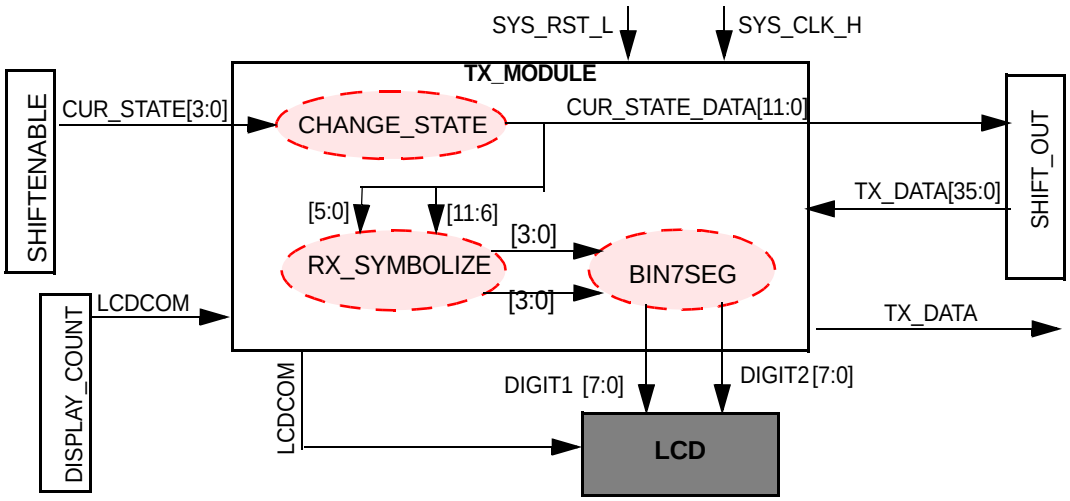

Figure 6: **TX_MODULE Block Diagram**

Display Count

The DISPLAY_COUNT block diagram is shown in Figure 7. This logic module controls the time between each state and the LCDCOM signal. STATE_COUNT is incremented and then enables SWITCH_EN_H. SWITCH_EN_H then enables the logic module SHIFT_ENABLE to change state (transmit new data).

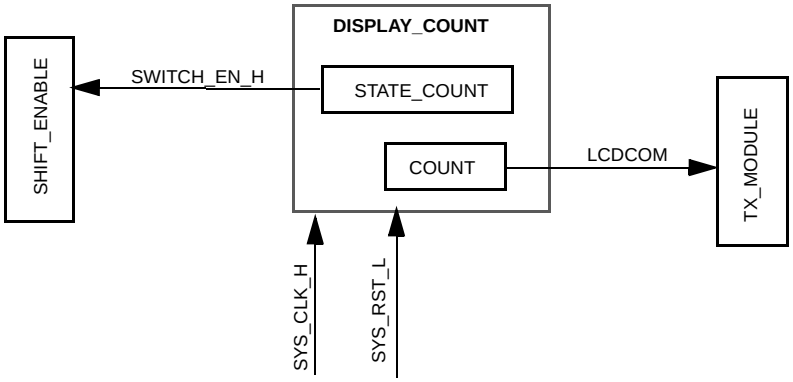

Figure 7: **Display Count Block Diagram**

Shift Enable

The SHIFT_ENABLE logic module increments the state variable to change states, and sends an edge to an enable signal (LOAD_DATA_H) to update a register in the SHIFT_OUT module with the new state value. The block diagram is shown in Figure 8.

TRANS_BETWEEN_COUNT determines the time between each state. TRANS_COUNT controls the number of transmissions between states.

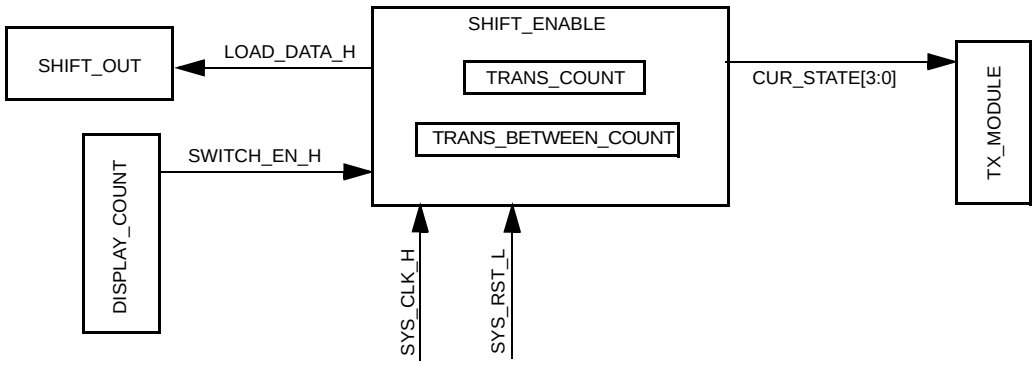

Figure 8: **SHIFT_ENABLE Block Diagram**

Shift Out

The SHIFT_OUT logic module sends the TX_DATA to TX_MODULE for transmission. LOAD_DATA_H enables the SHIFT_OUT module to load the current data. The block diagram is shown in Figure 9.

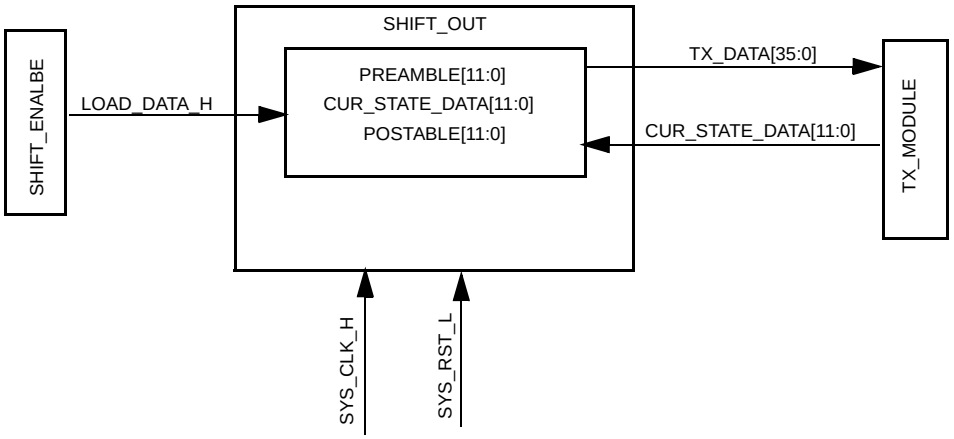

Figure 9: **SHIFT_OUT Block Diagram**

Receive Module Edge Detection

Receive

The receiver operation is included in one receive VHDL entity shown in Figure 4. Figure 10 shows the edge detection and sampling scheme of the ideal sampling model. Once an edge is detected, a counter insures the correct sampling and thus the storing of transmitted data. If non-ideal conditions exist, the location of sampling may need to be changed (see Figure 5).

The counter size and value used to sample the incoming bits is determined by the system clock and the baud rate. The RF module allows for a baud rate between 2.4 kbps to 19.2 kbps. With a 32.7kHz clock, a 2.4 kbps can be accurately modeled with a 5-bit counter. If the user wishes to change the baud rate, the value of the sampling counter must also be changed.

Further, the counter is re-initialized when a edge is detected. As previously discussed, this allows drift to be reduced to a minimum. Therefore, it is recommended that an encoding scheme which does not allow for long lengths of consecutive bits in the stream be used.

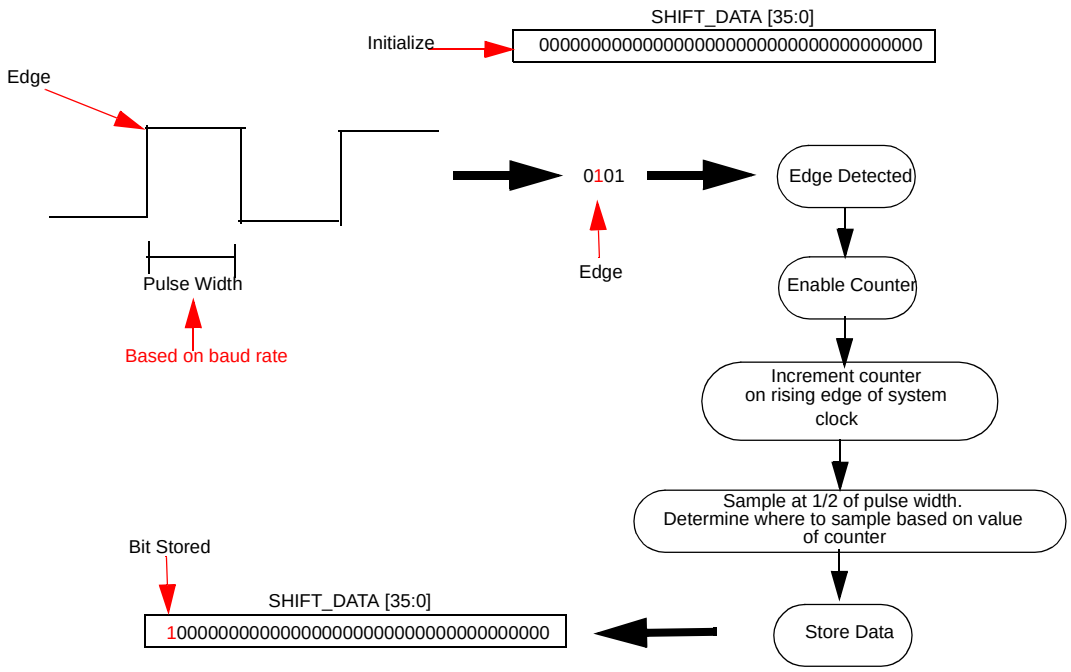

Figure 10: **Receive Edge Detection**

Hardware Description

The following describes the hardware used to develop the CoolRunner CPLD wireless transceiver.

RF Hardware

The RF transmission was preformed by the DR3000 module, manufactured RFM. The DR3000 is designed for short-range and low power applications with a carrier frequency of 916.5 MHz. Both On-Off Keyed (OOK) and Amplitude-Shift Keyed (ASK) modulation schemes are supported by the DR3000 module. The transceiver utilizes an Amplifier-Sequenced Hybrid (ASH) architecture and supports 2.4 to 19.2 kbps baud rates. The baud rates can be controlled with additional hardware changes to the RF module. The CoolRunner transceiver utilizes the 2.4 kbps transmission. The 2.4 baud rate was chosen due to the clock frequency available on the CPLD demo board.

CPLD Hardware

The CoolRunner XPLA3 demo board from Insight Electronics is used for the CoolRunner wireless transceiver. The demo board contains a two-digit LCD, 32.768 KHz clock, prototyping area and the Xilinx CoolRunner XPLA3 XCR3256XL TQ144 CPLD.

Hardware Setup

If using the AC adapter provided with the CoolRunner demo board, ensure that the resister, R7 (refer to the DR300 data sheet), is removed from the DR3000. If R7 is not removed, the DR3000 will heat up and no longer function properly. Also, ensure the RF module is attached to a proper power/ground plane to minimize ground loops.

The DR3000 requires a level shifter to correctly drive the CPLD I/O pin (see Figure 11). The RF module can not drive loads stronger than 500k ohms.

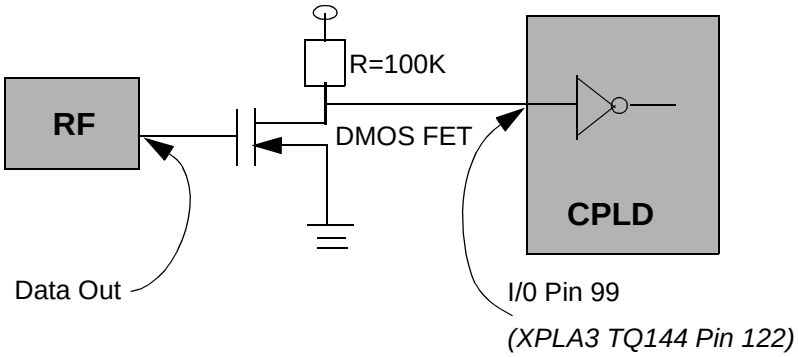

Figure 11: **Additional MOSFET Circuitry**

Keyboard Entry Option

The following is a design implementation option for using keyboard entry with the CoolRunner wireless transceiver. CPLD design implementation is left to the user to develop.

PS/2® Protocol

The keyboard interfaces with the CPLD using the PS/2 protocol. The PS/2 protocol works on serial communication between a host and a peripheral device. The bus can be in three states: idle, inhibit, and request to send. The device can transmit a byte to the host only when the bus is idle. In order for the bus to be idle, both the CLK and DATA pins must be high (logic 1). Table 1 is the pin layout for the PS/2 cable.

Table 1: **PS/2 Cable Pin Configuration**

Pin 1	PS/2 DATA
Pin 2	N/C
Pin 3	GROUND (0V)
Pin 4	POWER (+5V)
Pin 5	PS/2 CLK
Pin 6	N/C

The byte transmission includes a start bit (logic 0), eight data bits (LSB first), a parity bit (odd parity), and a stop bit (logic 1). The transmission occurs by having the device transmit a byte of data by pulsing the CLK low and high 11 times, sampling the DATA line. Figure 12 depicts the waveform for one PS/2 transmission.

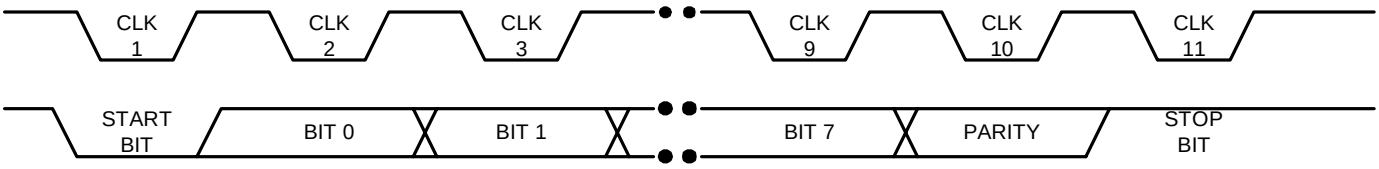

Figure 12: **PS/2 Transmission Waveform**

Hardware Description

In order to use a keyboard, a keyboard encoder must be used to manipulate data. The keyboard encoder used for this implementation is the Semtech Greencoder ™ (UR5HCFJL) Zero Power™ Keyboard Encoder for Portable Systems. This keyboard encoder is the device used between the keyboard and the peripheral device. It works on a matrix (8 X 16) format with the capability to support a 128 key keyboard. The keyboard encoder has three states that it operates in: sleep, stand by, and active. These states are used to efficiently manage power consumption, making this device a good fit for use with CoolRunner. The keyboard encoder used for this design implementation can function using 3, 3.3, or 5 volts and uses the PS/2 protocol to receive data from the keyboard.

CoolRunner XPLA3 CPLD Implementation

The CoolRunner transceiver is built using the CoolRunner XPLA3 Development Kit from Insight Electronics. Table 2 details the I/O pins on the demo board to the pins used on the XPLA3 256 macrocell part in the TQ144 package.

Table 2: **Prototyping Area I/O Cross Reference**

Transceiver Signal	Prototyping Area I/O	XPLA3 Pin Number
RX	I/O 99	119
TX	I/O 106	138
CONTROL	I/O 104	136

The wireless transceiver Receive module utilization in an XPLA3 256-macrocell device is shown in Table 3. The total utilization for the Receive Module allows room for additions and/or improvements to the design.

Table 3: **CoolRunner XPLA3 256-Macrocell Utilization for Receive**

Resource	Available	Used	Utilization (%(
Macrocells	256	168	65.63
P-terms	768	465	60.55
I/O Pins	116	20	17.25

The Transmit module utilization in an XPLA 256-macrocell device is shown in Table 4. Again, the total utilization for the transmit portion of the design allows room for addition and/or improvements to the design.

Table 4: **CoolRunner XPLA3-256 Macrocell Utilization for Transmit**

Resource	Available	Used	Utilization (%)
Macrocells	256	118	46.10
P-terms	768	202	26.31
I/O Pins	116	20	17.25

Design Verification

The design was verified in simulation and hardware implementation described previously in this document.

VHDL Disclaimer and Download Instructions

VHDL source code and test benches are available for this design. THE DESIGN IS PROVIDED TO YOU “AS IS”. XILINX MAKES AND YOU RECEIVE NO WARRANTIES OR CONDITIONS, EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE, AND XILINX SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGMENT, OR FITNESS FOR A PARTICULAR PURPOSE. XILINX DOES NOT WARRANT THE PERFORMANCE, FUNCTIONALITY, OR OPERATION OF THIS DESIGN WILL MEET YOUR REQUIREMENTS, OR THAT THE OPERATION OF THE DESIGN WILL BE UNINTERRUPTED OR ERROR FREE, OR THAT DEFECTS IN THE DESIGN WILL BE CORRECTED. FURTHERMORE, XILINX DOES NOT WARRANT OR MAKE ANY REPRESENTATIONS REGARDING USE OR THE RESULTS OF THE USE OF THE DESIGN IN TERMS OF CORRECTNESS, ACCURACY, RELIABILITY OR OTHERWISE.

XAPP353 - http://www.xilinx.com/products/xaw/coolvhdlq.htm

Conclusion

This document has detailed the design of the CoolRunner CPLD logic for a wireless transceiver. The design is targeted for a 3.3V, 256 macrocell CoolRunner CPLD (XCR3256XL TQ144). This device, as well as the RF module discussed in this paper, has extremely low static and dynamic power dissipation and therefore is ideally suited for this application. The design of the CoolRunner wireless transceiver is also provided as an example of using a CoolRunner CPLD in a portable application and can be extended to many other types of portable applications

References

1. Zetez Semiconductors Data Sheet - ZVNL110A N-Channel Enhancement Mode Vertical D(Double Diffused) MOS FET
2. USAR GreenCoder™ Evaluation Board Data Sheet - EVK5-FJL-7603-200
3. Anthes, John. "Unique Considerations for Data Radio UARTs." RF Monolithics, Inc.
4. RF Monolithics Data Sheet - DR3000 916.5 MHz Transceiver Module

Acknowled-gements

The CoolRunner wireless transceiver was development with the senior design team (May 01) of the University of New Mexico (UNM), Electrical and Computer Engineering Department. Design team included: Erin Isaacson (Xilinx), Lisa Burckel (UNM), Jeremy Dencklau (UNM), Kristina MIller (UNM), Parveen Sidu (UNM).

Additional thanks to Jim Beneke, Dennis Schlaht, and Lara Kieltyka of Insight Electronics and Bruce DeVisser of Fujitsu Takamisawa who donated time and equipment to the transceiver project.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
05/18/01	1.0	Initial Xilinx release.