ECE3071 Computer Systems

Laboratory Session 5

Programming the Serial Interface

Week 6 – Semester 1 2012

1. Objectives

The laboratory exercise introduces asynchronous serial communications and will give you experience of writing C programs that can handle multiple sources of interrupts. You will also investigate the interaction between interrupt code and normal code. The SOPC Builder is able to produce Verilog code for an RS232 asynchronous serial interface. Using the SOPC Builder you will design a Nios microprocessor system incorporating an RS232 interface. This system will then be programmed to send and receive serial data. In this exercise you will:

- Develop a Nios processor system incorporating an RS232 serial interface
- Write C code to send and receive serial data using interrupts
- Use an oscilloscope to examine serial data and confirm the structure of the data frames

Equipment

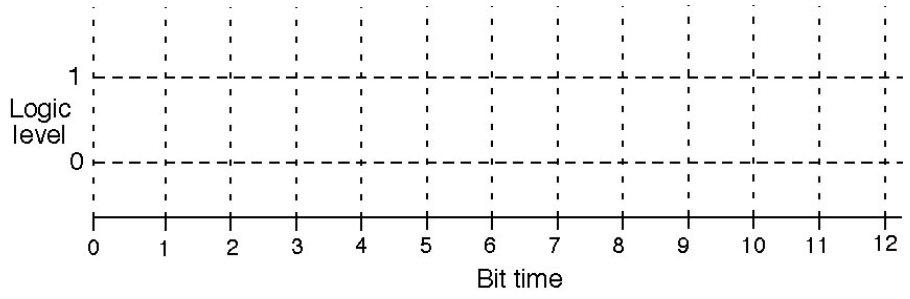
§ DE2 FPGA Development Board
§ A USB memory device provided by you to store your design files
§ A 2 or 4 channel 1GS/s digital oscilloscope
§ A 9-pin D-connector plug with pins 2 and 3 connected together.

2. Preliminary work

You must complete this preliminary work before attending the lab session.

Read through the whole of this document so that you understand all of the things you will be required to do.

If the ASCII character ‘a’ was being transmitted over an RS232 asynchronous serial interface with odd parity, 8 data bits and 1 stop bit, what changes in logic level would you expect to see on the transmitted data line? Fill in your answer on the following diagram:



Referring to the UART registers

a) which register do you read to access received serial data?

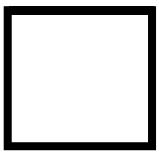
Register name ……..

Register offset ……

b) which register do you write to transmit serial data?

Register name ……..

Register offset ……

Demonstrator initial satisfactory preliminary work 

3. Design a Nios system for this lab exercise

Using the SOPC Builder create a Nios system with the following components:

- a Nios II/e processor,
- 16384 bytes of ROM
- 16384 bytes of RAM
- one 1-bit PIO input port (to read KEY1)
- one 1-bit PIO input port configured to produce an interrupt on an input rising edge ((MSB) of the 20-bit counter)
- an RS232 UART configured for odd parity, 8 data bits, 1 stop bit and a fixed Baud rate of 9600
- a JTAG UART

4. Produce a top level Verilog file

In your top-level Verilog file include:

- an instance of your Nios microcontroller system
- the Nios reset input connected to KEY[0]
- the Nios clock connected to the system 50-MHz clock
- a 20-bit counter clocked from the 50-MHz system clock with the MSB fed into the appropriate Nios 1-bit PIO input port
- the other 1-bit PIO input connected to KEY[1]
- directly connect the KEY[0] to LEDG[0] so you have visual confirmation of the reset signal
- arrange that the RS232 received data and transmitted data connections from the Nios microcontroller are connected to the correct DE2 board signals (UART_RXD,UART_TXD)

5. Write a C code program to do the following

a) Configure the Nios system to produce an interrupt on the rising edge of the MSB of the 20-bit counter.
b) In response to the 20-bit counter interrupt your service code should transmit one character via the RS232 serial interface. The characters will be “The quick brown fox jumps over the lazy dog “ and when this character string runs out the service code should send an ‘a’.
c) Configure the RS232 UART receiver to produce an interrupt when a character has been received.
d) In response to the RS232 receiver interrupt the received character should be read and sent to the JTAG serial interface. You can send a single character to the terminal window using the putchar() function.
e) In your normal code infinite loop you will monitor the state of KEY1 and when it is pressed (ie goes from being not pressed to being pressed) the interrupt code of part b) will start sending the quick brown fox character string again from the beginning.

6. Run your program and debug as necessary

Insert the loop-back connector into the DE2 board RS232 connector (the 9-pin D plug loop-back connector has pins 2 and 3 shorted together – connecting the transmitted data and received data lines).

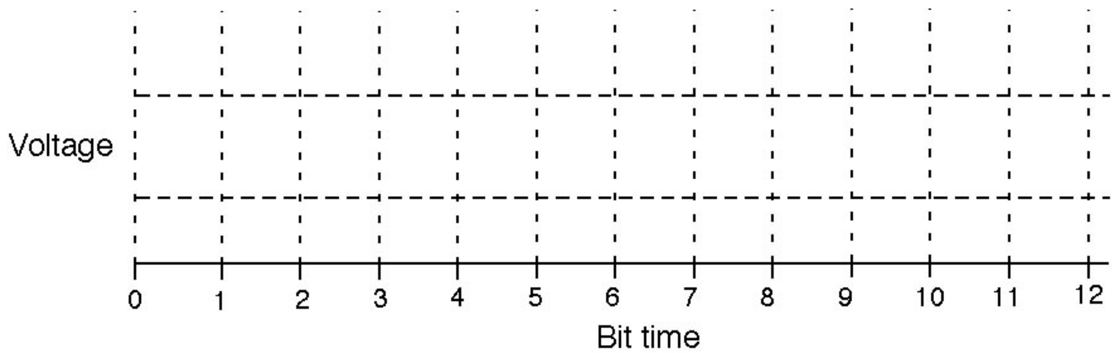
a) When you run your program the terminal window should display "The quick brown fox jumps over the lazy dog aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa".

Demonstrator initial satisfactory result from part a

b) When you press KEY1 the terminal window should stop displaying 'a's and display the quick brown fox string followed by more 'a's. "aaaaaaaaaaaaaaaaaaaThe quick brown fox jumps over the lazy dog aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa".

Demonstrator initial satisfactory result from part b

c) When the terminal window is receiving 'a's connect the first beam of the oscilloscope to the link between pins 2 and 3 of the serial connector. Trigger the oscilloscope to get a stable display. Then draw what you see:



How does this correspond to what you drew for the preliminary work?

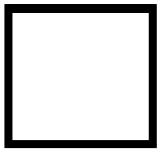
……..

Demonstrator initial satisfactory result from parts c

d) Using the oscilloscope measure the length of one bit time and calculate the associated Baud rate

Bit time ……………………………

Baud rate …………………………

Demonstrator initial satisfactory result from part d 

When you have complete the lab or when time is running out then start the assessment quiz for this laboratory exercise. The demonstrator will enter the password and record your mark (out of 5, one for each check box).

Ensure that your mark for this exercise is entered before you leave the lab.

Andy Russell 16/02/2012