Rensselaer Polytechnic Institute
Computer Hardware Design – ECSE 4770

Lab Assignment 5 - UART Laboratory

Revision D

Introduction

This laboratory is an introduction to I/O interfacing. The student will design an interface that allows effective communication between a peripheral and a RAM circuit that stores 16 characters. The interface will be built, tested and debugged in the lab.

Required Reading and Other References

UART Macros

Two UART macros are available for embedding in your Altera-based design, in lieu of an external UART. These macros allow you to reduce the amount of external circuitry needed to implement your Altera-based design. **Note: You will need one of these sets of files to complete the design for this lab.**

- UART VHDL Code : http://www.ecse.rpi.edu/courses/F13/ECSE-4770/LAB_5/UART_6402.vhd

- Pennybacker UART: http://www.ecse.rpi.edu/courses/F13/ECSE-4770/LAB_5/pennybacker.hd6402.vhd

Optoisolators

- 4n30 : http://www.datasheetcatalog.com/info_redirect/datasheet/motorola/4N30.pdf.shtml
- 4N37 : http://www.datasheetcatalog.com/info_redirect/datasheet/motorola/4N35.pdf.shtml

Other Information

- Open Directory, “Computer Hardware Busses”, http://www.dmoz.org/Computers/Hardware/Buses/RS-232/ : Here is a directory of sites with information about serial communications
- Beyondlogic.com, “Interfacing The Serial/RS-232 Port”, http://www.beyondlogic.org/serial/serial1.htm#part4 : Part 4 in particular has some relevant discussion especially a beginner’s level description of how a serial “frame” is constructed.
- Altera DE2 Users Manual : http://www.altera.com/education/univ/materials/boards/de2/unv-de2-board.html

Universal Asynchronous Receiver/Transmitter: Protocol Interface

Asynchronous data transmission is data transmission between devices operating at incompatible speeds. For example, when one wishes to transmit data from a tele-type (TTY) to a computer, one is using an input device that is several orders of magnitude slower than the computer. Also, when transferring data, it might be required to reformat the data so that the receiver obtains the data in the format in which it operates. Serial to parallel and parallel to serial conversion as well as other reformatting measures are common.

The Universal Asynchronous Receiver/Transmitter (UART) is a device that can conveniently carry out these functions. The UART receives and transmits data over a serial line, presenting a parallel interface to the computing hardware.

Each serial character consists of a start bit, five to eight data bits, an optional parity bit, and one or two stop bits. For example, the terminal output character consists of one start bit (a 0, ensuring a transition from the normally high level of a line that is idling), eight data bits, no parity bit, and one stop bit. A parallel character, on the other hand, consists of eight bits being output simultaneously and requires 8 lines.

Optoisolator and Current Loop: Electrical Interface

The optoisolator is one of the best ways of transferring signals between electrically unconnected circuits. The principle of operation is extremely simple. An optoisolator package consists of an LED (Light Emitting Diode) and a light sensitive transistor. When current flows through the LED, it emits light. When this light falls on the photosensitive transistor, the greater the potential difference and, if the output circuit is completed, the greater the current that flows through it.

The major advantage of designing with optoisolators is the degree of isolation provided. Note that no signal originating in the transistor can be fed back through the diode to the input circuit. This isolation is required where delicate equipment may be sourcing the LED.

A “current loop” employs current-based signaling. A logical “1” or “mark” is represented by 20mA flowing on the serial line; an open circuit with no current represents a logical “0” or “space”. While limited in speed as compared to RS-232, current loop offers better noise immunity and transmission distances in excess of 2000 feet. It is also possible to daisy chain multiple current loop devices as long as only one is active (supplying current to the loop) and the rest passive (switching or interupting the current on the loop). Since a serial line should idle at a logical “1”, this means that full current should be flowing when no characters are being transmitted. A current loop circuit using optoisolators is shown in Figure 1.

The PC in the lab emulates a terminal by using HyperTerminal, set use the com1 port (RS232). The RS232 cable has a current loop converter module installed. The terminal is a passive device in this loop. The external circuit will be the active device and will have to provide the 20 mA on each common line. The +12V sources shown in Figure 1 must be supplied in the lab. All the terminal does is switch the current on and off. The terminal you will use in the lab employs the ASCII code and transmits and receives at 9600 baud.

Prelab Assignment

1. Design the input/output circuitry for a terminal/UART interface as part of a 20 mA current loop. This design requires a suitable interface between the terminal (TTY) and the TTL UART device. When idling, both transmitter and receiver loops should have 20 mA

flowing. The 4N37 optoisolator has a current transfer ratio of 100%. Use the optoisolator data at the location given in the reference section to find the input diode forward voltage and output transistor Vce at saturation; assume that the drop across the TTY receiver (or transmitter) equivalent diode is 1.5 V when current is flowing in the loop. Show the calculations for all resistances in Figure 1. Select resistors R2 and R3 so that transistors are in saturation when idling; R1 so that the input current results in an appropriate output current; R4 so that TTL compatible voltage levels are achieved when the current loop is cloded or open. Discuss the design technique used for this circuit. The pinout for the current loop cable is shown in Figure 2. NOTE: The terminal is passive in this circuit. Therefore you have to supply the +12V power supply to the receiver and transmitter circuits.

2. Referring to figure 1 and the optoisolator specifications, notice that the base lead of the output transistor is unconnected.
   (a) Under what situations, if any, could the base lead be used?
   (b) How could the optoisolator be used as an amplitude modulator?
3. Using a RAM macro, a counter, a UART, the current loop logic previously designed, and some additional glue logic, design a circuit (see Figure 3) that will accept and store any 16 characters typed from a terminal in current loop mode. When 16 characters have been accepted, the circuit should immediately retransmit the 16 characters to the terminal in the same order that they were originally typed (FIFO). You will be provided a clock module that produces a 153.6 KHz clock signal required to run a UART at 9600 baud, or you may choose to use one of the DE2 card oscillators.

Lab Assignment

Build and debug the prelab design of one of the lab group members. HyperTerminal on the lab pc is set for 9600 baud, 8 bits, no parity, and one stop bit for communication with your design.

First check the 20 mA current loop to TTL converter by tying the serial-out and serial-in lines together. **Note: tie these lines together after the inverting drivers to ensure that the receiving side of the optoisolation circuit is generating TTL-level signals.** The terminal will echo characters when in this configuration. Since it is essential that the conversion logic work before the rest of the I/O logic can be checked, it is important to check the conversion logic first. Once the current loop to TTL conversion circuit is debugged, connect the I/O circuit and begin testing and debugging.

In the final lab report include the final debugged design, debugging procedure, discussion of problems observed in the design (if any), revised schematics, and all other pertinent information.

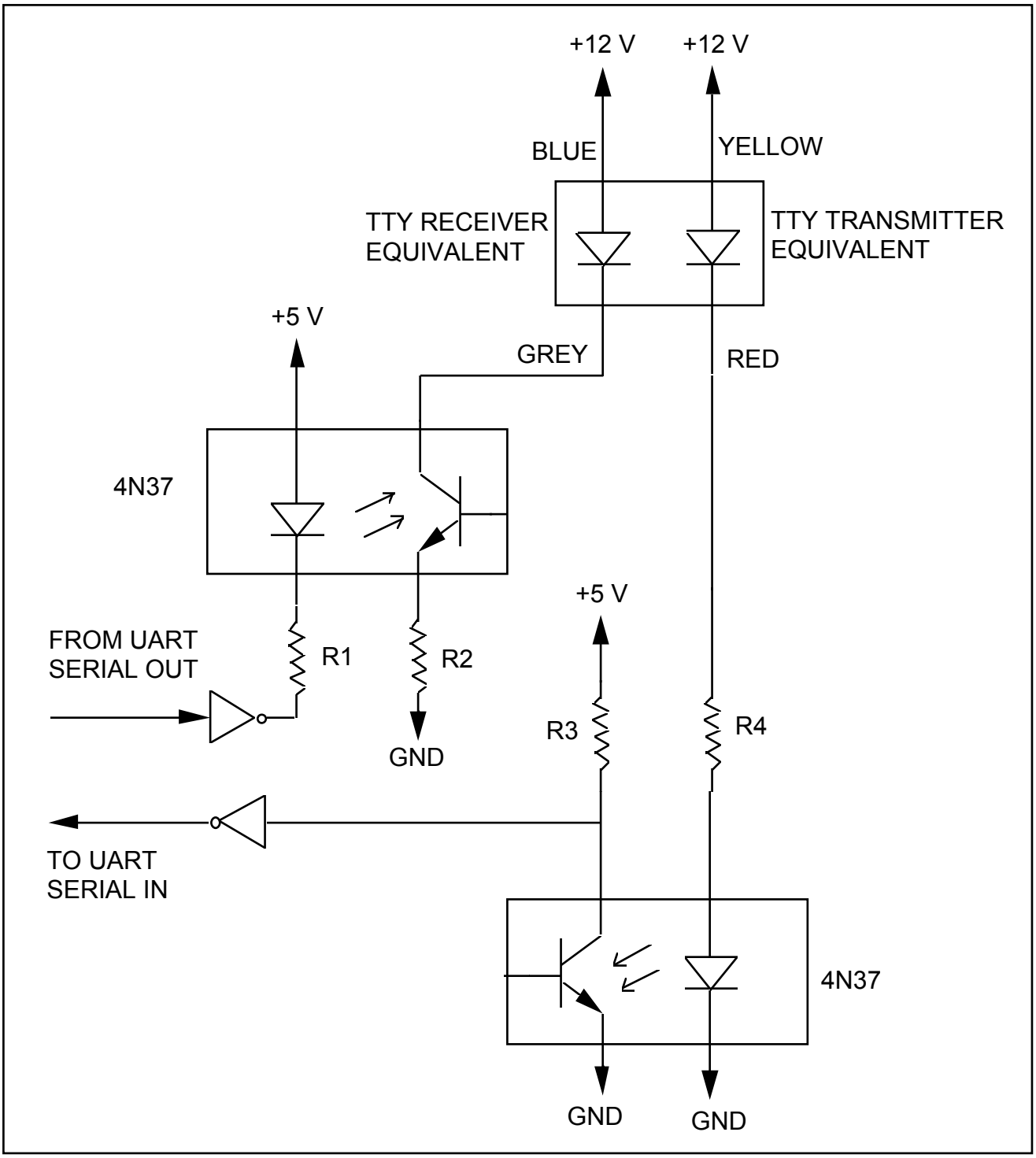

Figure 1: Optoisolator Circuit

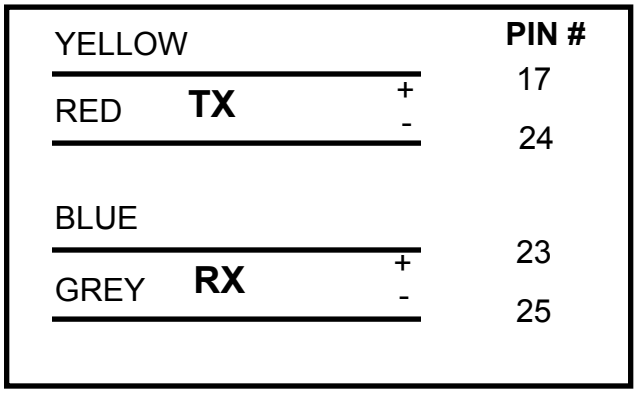

Figure 2: Current Loop Cable Pinout with Tail Colors

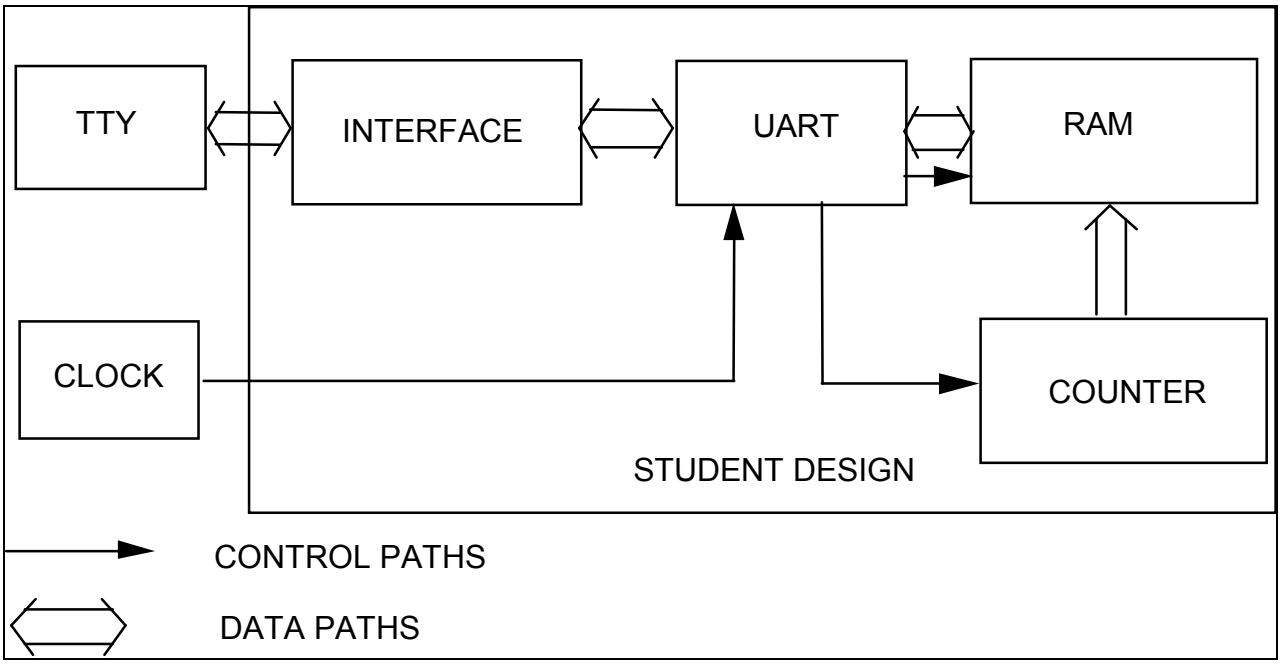

Figure 3: UART I/O interface system block diagram