Serial Port Project

CS 330

By: Nathan Lord

12/5/23

Purpose:

The purpose of the project is to be able to send data from the hardwired project, receive and interpret the data at the Arduino, and then the Arduino will then send the data back to the project for it to be displayed. In essence it is a two-way serial communication project between your hardware and the Arduino.

The project loops through from A-Z as well as gives the ASCII values for the corresponding letter. When data is sent from the project the Arduino will check and make sure the value falls between the ASCII values for A-Z inclusive. If it does not it will be ignored. If it does the loop will start from that point and will continue from there going through A-Z inclusive. The data is both sent to the backpack display to be displayed and the project's two seven segment displays through the TR1863/1865.

Hardware:

1 Western Digital TR1863/TR1865 UART (Universal Asynchronous Receiver/Transmitter)

2 7447's (Convert 4 bits into 7 segments for the display)

2 330 Ω Resistor Packs (limit the current and prevent burning out individual Segments)

2 7 seven segment (common anode) digital display chip (Display 4 bit numbers)

Clock:

To set up the clock look at the wiring diagram. The variable resistor is used to modify the clock speed when connected to the XSC1. When the clock is at the desired speed (19,200 Hz) connect it to the project. The baud rate is 1,200 bps.

1 555 timer

2 100nF Capacitors

1 1kΩ variable resistor

1 267Ω resistor

1 10.0kΩ resistor

TR1863/TR1865 Western Digital – Universal Asynchronous Receiver/Transmitter (UART) - Pinout

1. 5 V
2. No connection
3. 0 V Ground
4. 0 V Ground
5. D
6. C
7. B
8. A
9. D
10. C
11. B
12. A
13. LED D
14. LED C
15. LED B
16. 0 V Ground
17. Connect to #40 Pin
18. SW Ȳ
19. LED A
20. Connect RX20 to TX1 when connecting Arduino (Otherwise RX20 to TX25)
21. 0 V Ground
22. LED E
23. SW x̄
24. LED F
25. Connect TX25 to RX1 when connecting Arduino (Otherwise RX20 to TX25)
26. SW 8
27. SW 7
28. SW 6
29. SW 5
30. SW 4
31. SW 3
32. SW 2
33. SW 1
34. 5 V
35. 5 V
36. 0 V Ground (for one stop bit)
37. 5 V (37 and 38 being high represents for 8 data bits)
38. 5 V
39. No connection
40. Clock Input

5-8 are the high 4 bits

9-12 are the low 4 bits

Function:

Data input – Comes from SW1-SW8.

SW1-SW4 represents the high four bits.

SW5-SW8 represents the low four bits.

S**W** $\bar{\mathbf{x}}$ - Used to send data on the negative edge of the switch (when flipping back down).

SW $\bar{\mathbf{y}}$ - Used to reset the data received flag (LED A).

THRE – LED E – Transmitter Holding Register Empty. Represents when high that data has been transferred and can receive new data.

DR – LED A – Data Received. Represents when high that the character has been received and transferred to the Receiver Holding Register.

OE – LED B – Overrun Error. Represents when high that **DR** was not reset before the new data was transferred to the Receiver Holding Register.

Data output – the bottom display is the (SW1-SW4 high four bits in hex).

the top display is the (SW5-SW8 low four bits in hex).

Timing Diagram:

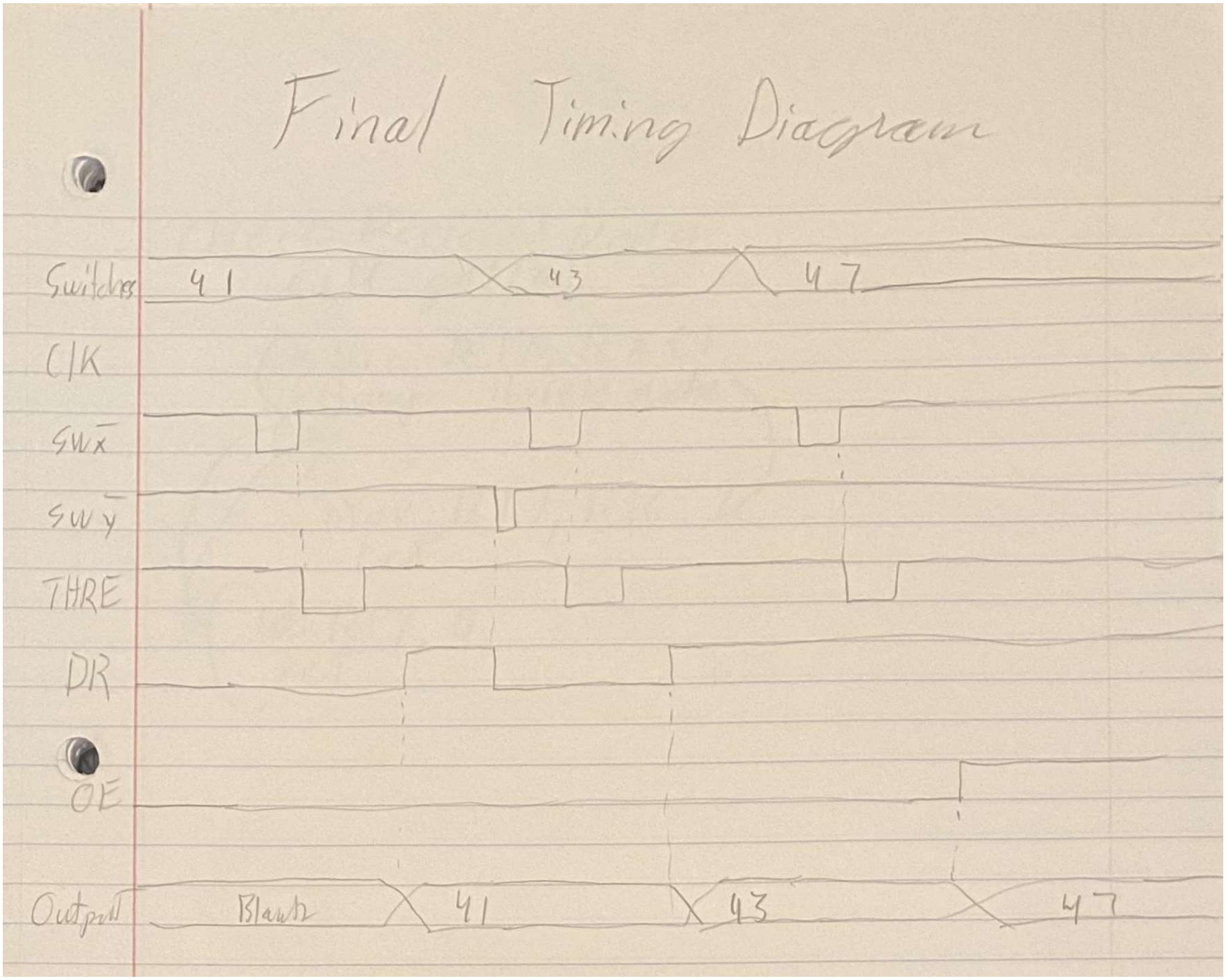
Walk through of the timing diagram.

The switches are set to 41 (in hex, 0b01000100) and currently there is no output. When S**W** $\bar{\mathbf{x}}$ is flipped up and back down, on the down flip (negative edge), the data is sent. The **THRE** goes off/low indicating that the data is currently being transferred and cannot receive new data. It then flips back to high, and **DR** goes high the moment the output displays 41 indicating that data has been received. We can put **DR** back low by flipping SW $\bar{\mathrm{y}}$. On the upstroke of the flip the **DR** will go low.

We can then set the new input/switches to 43. We go through the same sequences of flipping the S**W** $\bar{\mathbf{x}}$ and the LEDs light up the same as well as 43 now being the output. However, here we do not flip SW $\bar{\mathrm{y}}$. Therefore, the **DR** does not go low and instead stays high.

Now we load in 47 to the switches. S**W** $\bar{\mathbf{x}}$ is flipped and **THRE** acts the same as always, but **DR** is still high from before. When the data is then output **OE** goes high indicating that we have an overrun error because we did not reset **DR** before sending the 47 data transmission.

Timing Diagram



Function Cont:

These are the other LEDS but do not affect the function of our project. For example, **PE** deals with the parity error but on the TR1863 we do not have anything wired to pin 39 for parity.

FE – LED C – Framing Error. Represents when high that the received character has no valid stop bit.

PE – LED D – Parity Error. Represents when high that the received parity differ from that which is programmed by the Even Parity Enable (pin 39).

TRE – LED F – Transmitter Register Empty Represents when high has completed transmission including stop bits.

Other important functions.

Pins 33-26 represent the hex value from the switch inputs. 33-30 is the high hex value and 29-26 is the low hex value. These are then converted to pins 12-5 which are split from the low 12-9 and the high 8-5. Pins 8-5 split to the high display track (bottom of wiring diagram) which convert the bits to a seven-segment display – 7447 to 330 Ω resistor to 7 segment display. Pins 12-9 split to the top track which is the same makeup of the bottom track.

Wiring Diagram