**APPLICATION OF A DIGITAL LATCH**

**Aayahna M. Herbert**

**September 19, 2017**

**ECE 3720 Section 007**

**Microcontroller Interfacing Laboratory**

**Abstract:** A circuit was built in order to send two inputs to the microcontroller from the DIO. The microcontroller read the inputs and sent them to the LEDs and the latch. When using the digital writer, the first two lights should light up and when the push button was used, it should have sent a signal to the MC. The two LEDs mirrors the input values, but the other two LEDs did not light up when the push button was pressed.

**INTRODUCTION**

The purpose of this lab was to design a circuit that sends two inputs to the microcontroller from Digital Input/Output, which then sends them to the D-Latch and two of the four LEDs that would consequently light up when controlled by the Digital Writer program. The other two LEDs would light up when the push button was pressed and its corresponding “button” on the Digital Writer was switched on.

**EXPERIMENTAL PROCEDURES**

The equipment used include one PIC32 MC, NI-ELVIS II board, a push button, a 74LS373 D-Latch, and a resistor. The first thing to be done, to avoid forgetting it later, is to connect the MC, D-Latch, and push button to ground and to the +5V power source. For the microcontroller, pins 40 and 3 were used to connect to ground and +5V power source, respectively. Ports 2 and 4 were used on the push button to ground it and then power it using the +5V power supply in that order; to properly power the push button, the resistor was placed between the button and wire going to the boards voltage supply. Gates 10 and 20 were used to ground and power the D-Latch gate, correspondingly. Once the grounding and powering are done, the microcontroller’s inputs and outputs are next to be connected. Pin numbers 4, 5, and 6 will be used for the inputs while pin numbers 35, 36, and 37 are for the outputs. The inputs for the MC are the push button and DIO ports 0 and 1; have one wire connect from the push button’s port 4 (connected directly from the device, not by the resistor) to MC’s pin 4. The next wire should connect DIO 0 located on the upper right hand side of the board to the pin number 5; the last input wire will connect DIO 1 to the microcontroller through pin number 6. For the outputs, the microcontrollers will be sending signals to two LEDs located on the board, the clock located in the D-Latch, and two of the latches input gates. Connect one wire from pin number 37 of the MC to gate number 11, which represents the latch’s clock. The last two wires will both be going from the microcontroller to two different input gates belonging to the D-Latch; the first wire will be from pin number 36 to 1D (gate number 3) while the second goes from pin 35 to 2D (gate number 4). From gate number 3, another wire will be connected there and go to LED 0 in the board; another wire will be used to the same for gate number 4, connecting it from there to LED 1.Now that the inputs and outputs of the MC are completed, the final wirings will be dealing with the outputs of the D-Latch. For the latch, we will need to connect the last two LEDs to two of the latch’s outputs that correspond to the appropriate input gates. So the LED 2 port will connect to 2Q (gate number 5) and LED 3 port to 1Q (gate number 2); this is done so LED 2 lights up with LED 0 and LED 3 lights up with LED 1. The remaining part making sure the code represents what the circuit shows. Using TRISx, set the pin numbers 4, 5, and 6 equal to one to declare them as inputs and pin numbers 35, 36, and 37 equal to zero so they are declared as outputs of the microcontroller. The last part of the code will use LATx and PORTx to write to where a specific pin is read; the code should be set up to where the clock of the D-Latch writes where the inverse (~) of the push button is read, latch output 1D writes where DIO 0 reads, and latch output 2D writes where DIO 1 is read.

**RESULTS**

After the circuit was hooked up and ready to be tested, the program detected no compiling errors when ran so the board could then be tested. While running tests on the Digital Writer to check whether or not all of the LEDs lit up when they were supposed to, the first two LEDs lit up when the 1 and 2 switches on the digital writer were turned on. However, when the push button was pressed, the second two LEDs did not respond.

**DISCUSSION**

Everything was going smoothly and the results coming back from the circuit matched the lab expectations up until the push button LEDs. The problem arose when the third and fourth LEDs didn’t light up with the digital writer nor with the pressing of the push button. The LEDs themselves were tested individually to make sure none of the ones in use were broken. Some possible sources of errors could have been from incorrect coding when trying to properly deal with the push button being low active, faulty wiring, or just a bad or dead push button. These possible sources of errors came to mind during the lab; the code was second-checked by the TA and the wires all came from a new pack of wires recently purchased to avoid the possibility of dead wires being used and affecting the lab results. The only possible error source left would be the use of a dead or incorrectly used push button since it was never checked to see if it worked independently; the push button itself was rather hard to deal with when placing and removing it on and from the board. In the future to improve future labs and avoid errors, all devices used in each lab experiment will be tested separately to make sure they are successfully functioning in general.

**CONCLUSIONS**

In general, the take-away of this lab was to get a firm understanding of how to apply a digital latch to a microcontroller and to get a mirroring results of the input values and push button from the circuit board through the LEDs. Since this was also the first lab done, it also laid how to start off each lab experiment and the basic coding structure of the MC chip.

**FIGURES AND TABLES**

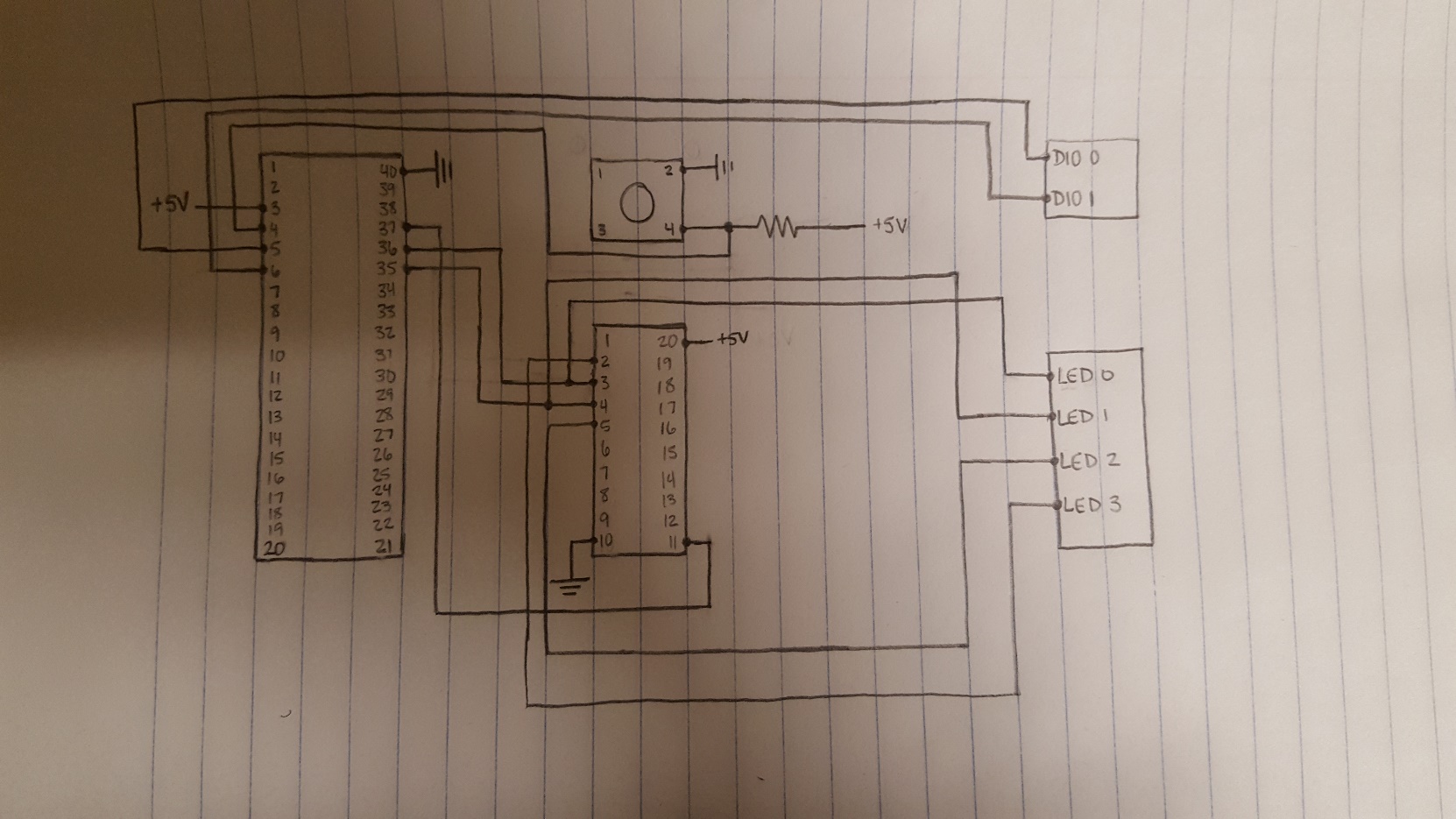


Figure 1: Circuit Schematic

**CODE**

#include<plib.h>

void main(void)

{

TRISBbits.TRSIB9 = 1;

TRISCbits.TRISC6 = 1;

TRISCbits.TRISC7 = 1;

TRISBbits.TRISB7 = 0;

TRISBbits.TRISB6 = 0;

TRISBbits.TRISB5 = 0;

while(1)

{

LATBbits.LATB7 = ~PORTBbits.RB9;

LATBbits.LATB6 = PORTCbits.RC6;

LATBbits.LATB5 = PORTCbits.RC7;

}

}