6 Project Experiment 1: The Bus, Arithmetic Unit and Program Counter

6.1 Introduction

This lab focuses around building the bus of the computer and implementing several crucial components of the computer, namely the arithmetic unit and program counter. Three 74LS173 4-bit registers and a 74LS283 4-bit adder are used for this experiment.

6.2 A Brief Note on Timing Diagrams and Schematics

At this point in time, knowledge of how to read schematics is necessary. If you are unsure how to work with schematics, it is suggested that you read Appendix A of this lab manual. In the following sections, the use of labels is present. The most common one is *FROM TSG*. This is a label but not an actual connection. The location that the label points to is generic and it just solely implies that there is some connection from the timing signal generator and each label is not necessarily the same connection either. It is a very common mistake for students to simply write this label on their schematics for reports. Another common issue found in lab reports involves timing diagrams. It is highly suggested that students do read Appendix B to gain a good understanding of how to produce correct timing diagrams.

6.3 The Data Bus

A bus is a set of wires that numerous devices are connected to. Instead of patching numerous cables on a breadboard or traces on a circuit board, a set of wires is used for multiple devices to pass data through. To implement a bus, each row can be lengthened beyond the gaps on a breadboard by running one wire and connecting it to the next row. By doing this, long rows can be linked with the same logic value and can be used to connect different devices. This can be visualized in Figure 36. For this experiment, the bus lanes are B_3 to B_0, going from left to right. Students may decide to change the order but are responsible to keep track of their implementation.

An advantage of using a bus is that devices have a centralized point in which they can communicate without patching numerous cables in many directions. The two components being built in this lab use the bus to communicate. In this lab experiment, a four bit bus similar to what is shown in Figure 36 is constructed. The program counter and arithmetic unit are connected to this bus and communicate through it. These devices are controlled by the timing signal generator built previously. The first four pulses of the instruction cycle are used: T_0, T_1, T_2 and T_3.

When dealing with numerous devices on a bus, some method is needed so that the devices on the bus that output data do not interfere with each other. The data registers used in this lab experiment have tri-state outputs for this

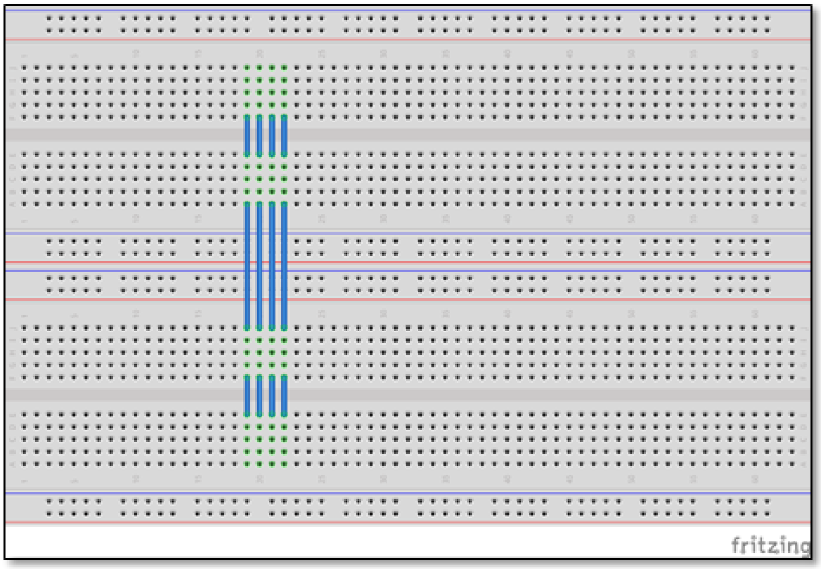

Figure 36: 4 Bit Bus Example

purpose. This means that they can output a logic 0, logic 1 or a *high-impedance value* (also known as a High-Z). When these devices' outputs are in a high impedance state, they effectively have no output on the bus. This is used so that devices can share the bus without having multiple devices try and output their data at the same time. When multiple devices output onto a bus at the same time, a *bus conflict* occurs. Bus conflicts are undesirable for several reasons including the possibility of damaging the devices connected to the bus.

6.4 74LS283 4-Bit Adder

The 74LS283 is a 4-bit adder in a 16 pin package. The pin-out diagram is shown in Figure 37. Pin 16 must be wired to V_{CC} and Pin 8 must be wired to *GND*. The pins labeled S_0, S_1, S_2 and S_3 are the sum outputs, X_0 to X_3 and Y_0 to Y_3 are the inputs. For this lab, the inputs Y_0 to Y_3 are grounded while pin 7, the carry in, is wired to V_{CC}. Inputs X_0 to X_3 are taken from the bus. By doing this, the sum will always be $PC = PC + 1$. *Note: The pin that corresponds to the carry out is intended not to be wired to anything.*

6.5 74LS173 4-Bit Register

The 74LS173 is a 4-bit data register with tri-state buffers. The pin-out diagram can be seen in Figure 38. The master reset pin (pin 15) is tied to *GND* to prevent the register from resetting since the pin is active-high. The clock input is located on pin 7 and is positive-edge triggered. Since the computer is negative-

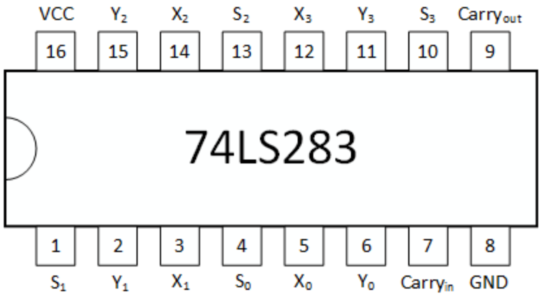

Figure 37: 74LS283 Pin-Out Diagram

edge triggered, the clock pulse must be inverted before being inputted to the register. This can be seen in Figure 39. *The entire lab is to be implemented with only one 7404 used as clock input inverters.*

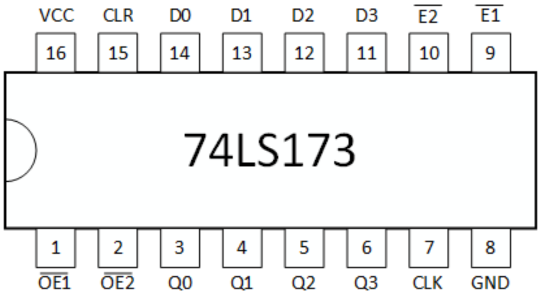

Figure 38: 74LS173 Pin-Out Diagram

Once again, pin 16 must be wired to V_{CC} and pin 8 to ground. Pins 9 and 10 are the write-enable inputs. As shown in Figure 39, they should always be tied to *GND*. Pins 1 and 2 are the output enable pins of the register. These two pins are to always be connected together so that they have the same inputs. The reason is that the register uses the following equation to determine if the tri-state buffers are in high impedance: $\overline{OutputEnable} = \overline{OE1} + \overline{OE2}$

Lastly, pins 3 to 6 are outputs and pins 11 to 14 are inputs. In Figure 38, the pin labeled D_0 is the input for bit 1. The output of bit 1 is pin Q_0. This can be summarized in Table 8

6.6 The Program Counter

The Program Counter of a computer is an essential register. It holds the location of the instruction to be executed. During the execution cycle, the program counter is incremented through the bus so that the next instruction can be read when needed. It can be envisioned as a bookmark used to keep the place in a

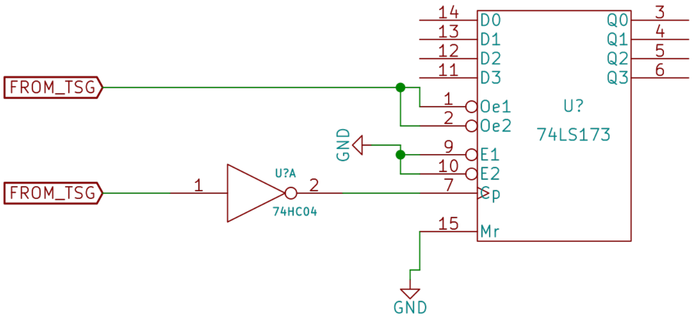

Figure 39: General 74LS173 Circuit

Bit Number	Pin Number	Function	Label
0	3	Output	Q0
1	4	Output	Q1
2	5	Output	Q2
3	6	Output	Q3
0	14	Input	D0
1	13	Input	D1
2	12	Input	D2
3	11	Input	D3

Table 8: Mapping of Inputs and Outputs for 74LS173

rather large book. It is implemented by using a 74LS173 data register. Since the 74LS173 is a four bit data register, the computer will have a program composed of a total of 16 instructions before it rolls over back to address 0000.

6.7 The Arithmetic Unit

The arithmetic unit provides the ability to increment the program counter. Without this, the computer would just execute a single instruction and would not be of much use. The arithmetic unit is created by using a 74LS173 data register and a 74LS283 4-bit adder. The inputs of the adder are taken from the bus, the output is fed to the inputs of the 74LS173 belonging to the arithmetic unit. The outputs of the 74LS173 are fed to the bus as well. Since the 74LS173 has the ability to control the state of its outputs, the Timing Signal Generator is used to ensure that only one register is outputting data at a time. Ultimately, the Timing Signal Generator allows the program counter to increment itself, allow the computer to run any program provided and to increment any variables used in a program.

6.8 A Brief Overview on the Program Counter and Incrementer System

Each of the registers use some of the first four timing pulses generated by the 74LS164: T_0, T_1, T_2 and T_3. When T_0 becomes active, the program counter register outputs its contents (the address of the current instruction) onto the bus. The value flows into the 74LS283 4-bit adder and is incremented by 1. T_1 becomes active and the SUM register stores the data. Afterwards, when T_2 becomes active, the output of the SUM register becomes available on the bus. The last step in the cycle is when T_3 becomes active and causes the PC register to write the incremented value. The end result of this cycle can be expressed as $X = X + 1$. Once the register reaches 1111, it will roll over to 0000 and repeat the cycle. A third register is placed on the bus to be used as a tool to mirror any register. It must use the same timing signal of the register intended to be mirrored for it to work. The outputs are wired to LEDs and the inputs are taken from the bus.

6.9 Assembling the Circuit

This portion of the computer makes use of the Timing Signal Generator and its first four pulses, T_0 to T_3. The schematics are presented in Figures 40, 41, 42 and 43. *It is up to the student to determine what reference each register and part is.* The first signal, T_0, triggers the output of the program counter. It will be wired to the connection labelled PC_{out}. T_1 triggers the inputs of the SUM register so it will be connected to the pin labelled SUM_{in} respectfully. T_2 triggers the output of the SUM register, so it will be wired to the SUM_{out} pin. Lastly, T_3 triggers the input of the program register so it will be wired to the pin labelled PC_{in}. Additionally, the output enable on the mirror register is tied to Ground so that it will always output the contents to a set of LEDs, making it useful for a debugging tool. It is highly suggested to use T_1 as the clock input for the mirror register. By doing so, the mirror register will display the current instruction address.

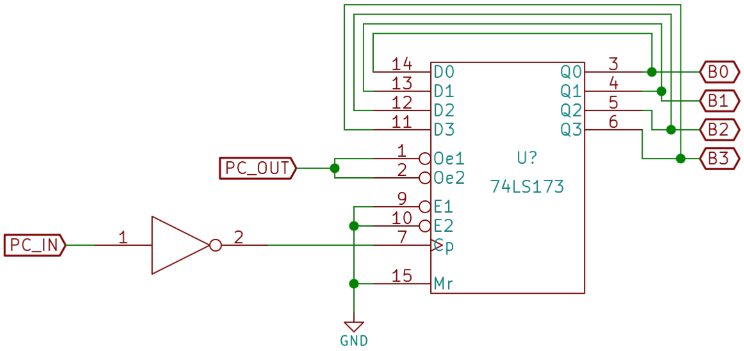

Figure 40: Program Counter Register Schematic

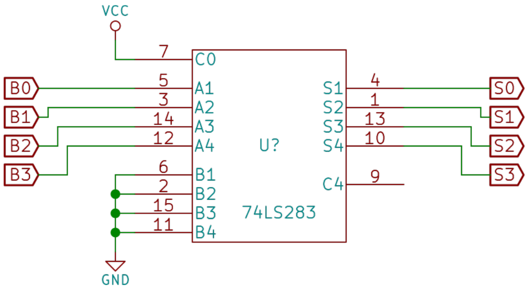

Figure 41: 4-Bit Adder Schematic

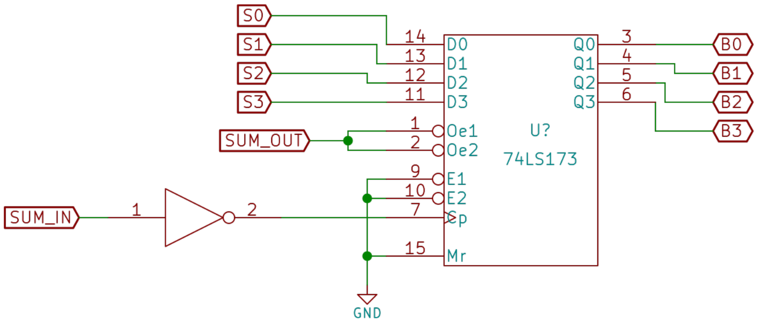

Figure 42: Sum Register Schematic

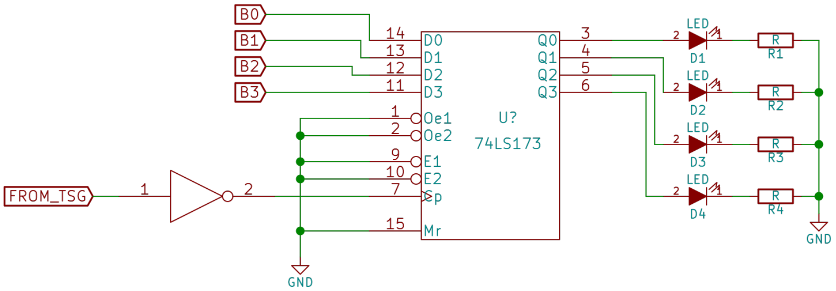

Figure 43: Mirror Register Schematic