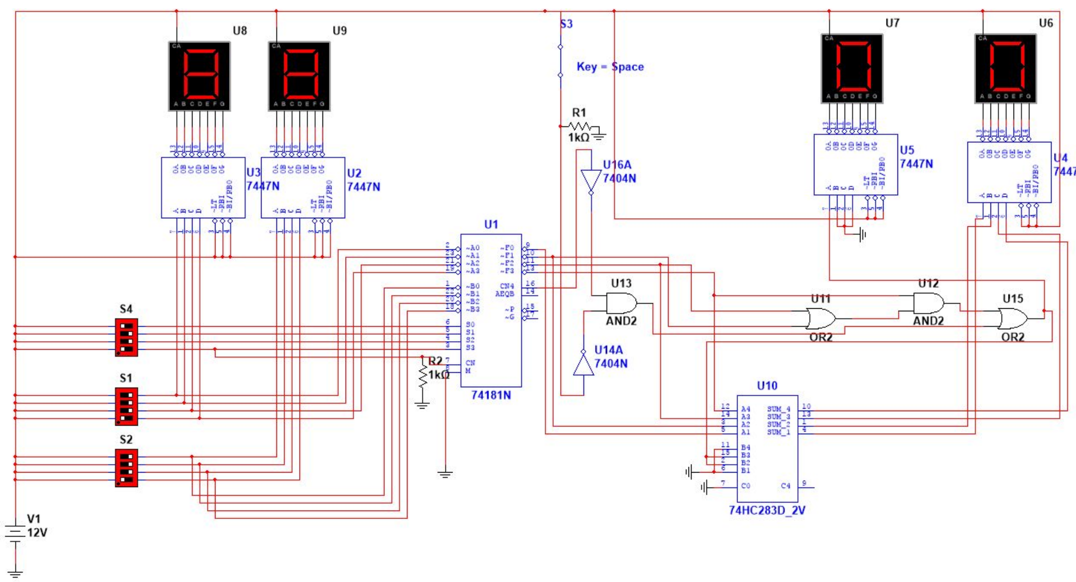
Adder/Subtractor With ALU



Kevin Downing

Raymond Makynen

Andrew O’Toole

Ryan Shaw

12/14/20-12/18/20

Objectives:

During this lab, you will be able to:

- Build an adder/subtractor using an ALU chip

Procedure:

During this lab, you will use the following materials:

- Multisim
- 74181N ALU chip
- OR Gates
- AND Gates
- 7404N IC Chip (Inverters)
- 7447N IC Chip (Decoders) (4)
- 74HC283D IC Chip
- 7 Segment Display (4)
- DSWPK Switch (3)
- 1kΩ resistors (14)

To begin this lab, start by creating an adder subtractor using a 74181N ALU chip, whose pinout is shown in figure 1-1.

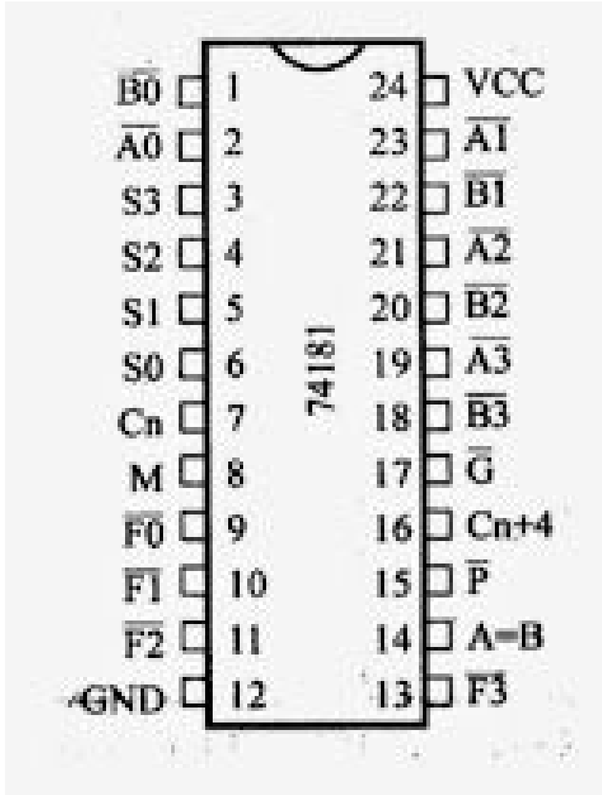


Figure 1-1 (74181N)

With this IC, attach the A and B inputs to Vcc with switches for each input, which will be inverted by the ALU. Then, attach pulldown resistors going to ground to each of these inputs as well. These will make sure that if the switches are set to open, there won't be anything floating. Then, connect CN to S3 because the only two functions will be addition and subtraction, and with addition, both of these will be high, for a high on C_N will cause an additional 1 to be added to the circuit. With subtraction, this additional 1 is not wanted, meaning that it needs to be set to a 0. Pin M will go to ground, for this will only be using arithmetic and not logic.

Next, set up the correction circuit and the two AND gates and two OR gates connecting to one another. The first AND gate should have both inputs inverted, with one connected to power and ground through a switch, while the other is connected to C_{N+4} on the ALU. Next there is an OR gate which has the inputs set to the middle two output sums of the ALU. The output of the OR gate connects to the input of another AND gate. The other input is connected to the output of the ALU which is representing eight. The output of this AND gate then connects to an OR gate. The other input is set to the output of the first AND gate used.

After this, insert a 74283 IC acting as a correction adder. The A inputs of these should connect to their respective F outputs on the ALU, which are inverted. Then, connect B1, B4, and Cin to ground. Also, connect B2 to B3 and to the output of the final OR Gate of the correction circuit.

Once this is done, insert four decoders, each going to their own 7-segment display. Two of these will be used for the inputs and two will be used for the outputs. The two decoders on the inputs will connect their A-D pins to the A and B inputs respectively. Then, the LT, RBI, and BI/RBO pins will connect to Vcc. Have the A-G outputs connecting to the inputs of the 7-segment displays.

The next two decoders will be used for the outputs. One will be representing the 10^0 column, and one will represent the 10^1 column. The decoder acting as the 10^0 column has its A-D inputs connecting to the outputs of the correction adder. The LT,RBI, and BI/RBO will again connect to Vcc and the A-G outputs will again connect to a 7-segment display. The 10^1 decoder will have its B-D inputs connecting to ground and the A input connecting to the output of the correction circuit. Again, LT,RBI, and BI/RBO will connect to Vcc and the outputs will connect to a 7-segment display.

These 7-segment displays are common anode, meaning that they all need to have their CA pin connected to Vcc. The final circuit is shown in figure 1-2.

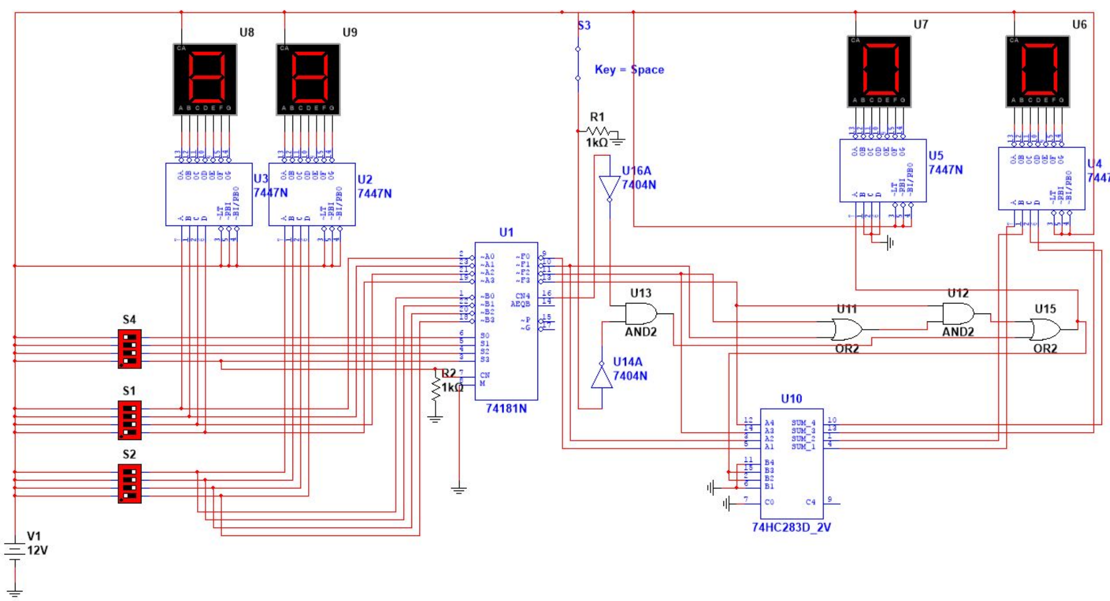


Figure 1-2

To test this circuit you would need to set the switches going to the A and B inputs of the ALU to whatever you wish to add or subtract. Then, their respective numbers will appear on the 7-segment displays. If adding, the switches connecting to the S0-S3 on the ALU will be High Low Low High, and the switch connecting to the inverter on the first AND gate of the correction circuit will be open. If subtracting, the switches will be Low High High Low and that switch will be closed.

<u>Discussion:</u>

This lab gave us some issues, which included a hiccup nearing the end of the creation of the circuit. The most prevalent of these issues, however, was when we had encountered an issue of not being able to add past 15. The change we needed was adding an inverter onto the C_{N+4} which allowed us to then add any two one-digit numbers. We also had a problem with our correction circuit not properly working preventing us from properly subtracting. We fixed this by adding a switch to the other input of the first AND gate in the correction circuit which allowed us then to open the switch when subtracting and closing it when adding. The reason for this is the number gets inverted so if the switch is open it will output a zero which gets changed to a one allowing the correction circuit to turn on during subtraction and vice versa for adding with the switch being closed.

Conclusion:

An ALU chip is an arithmetic logic unit that makes performing operations such as addition and subtraction much easier. The mode select input (M) determines the operation the ALU carries out, if low (0), arithmetic is carried out, and if high (1), logical operations are carried out. For this experiment, only arithmetic operations were needed, meaning that it just needed to be grounded. The inputs A and B are determined by their binary values. The function select inputs (S_0-S_3) determine what function is done to the A and B inputs. For example, if these are high, low, low, high, then A is added to B. Consequently, if they are low, high, high, low, then B is subtracted from A. The outcome of any operation is then output in F_0-F_3, which gets inverted by the ALU. When this occurs, the values travel through the correction circuit, which then corrects the binary values to BCD. These BCD values are then sent through decoders which convert the BCD values into readable numbers on the 7-segment displays connected to them.