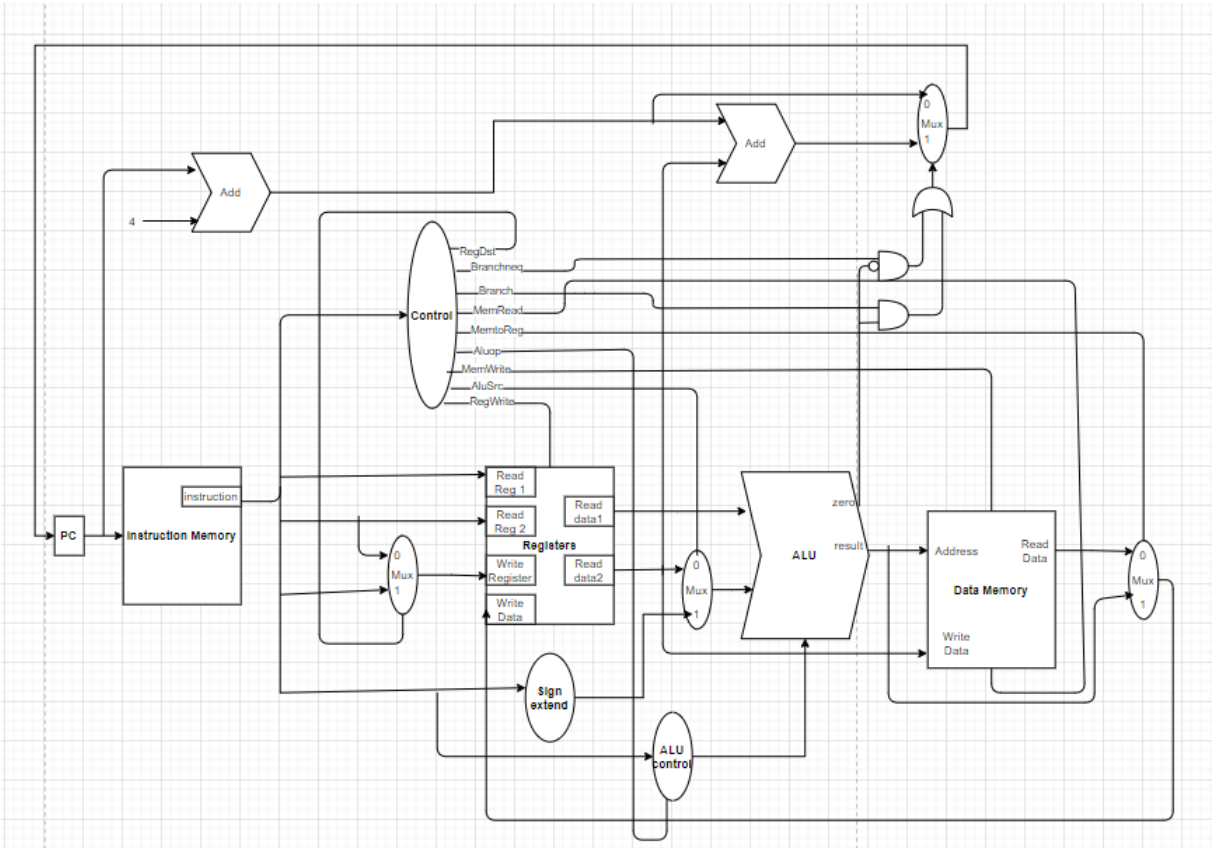
Assignment 4 Report – MIPS Single Cycle Processor

Aziz Can Akkaya / 1801042250

Name: Aziz Can Akkaya

Number: 1801042250

DATAPATH:

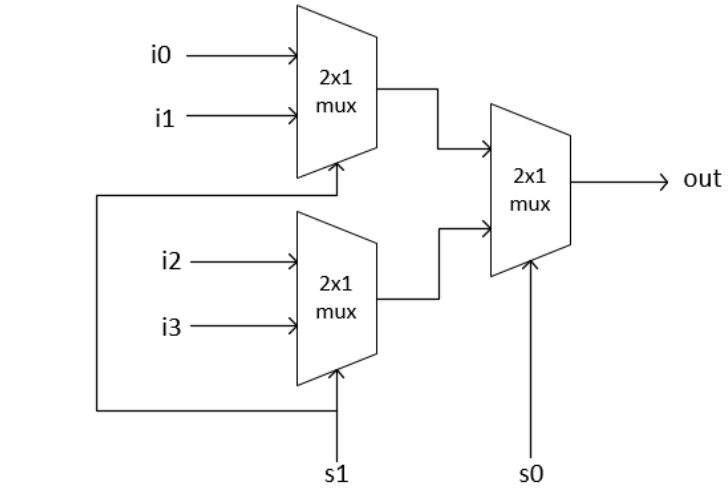


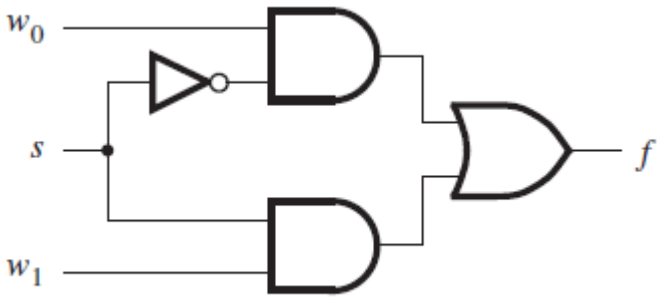
Due to jal and j have the same op code i was not able to convert them into a control signal. But for them to work we just simply need to a 3 input mux at PC's input and 4 input mux to the input of the registers input. The input will be determined by 3 signals.

MODULES:

- **MUX:**

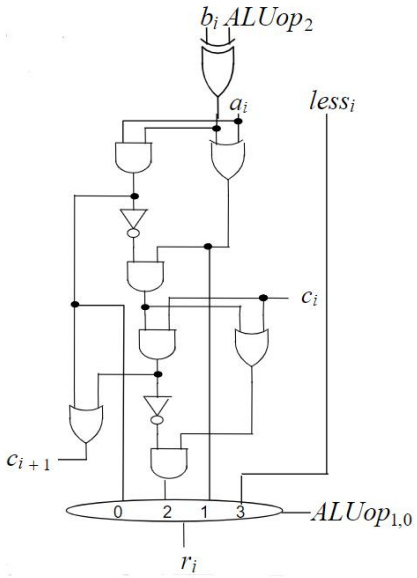
For the essential we used a simple single bite input multiplexer. The circuit design shown below:



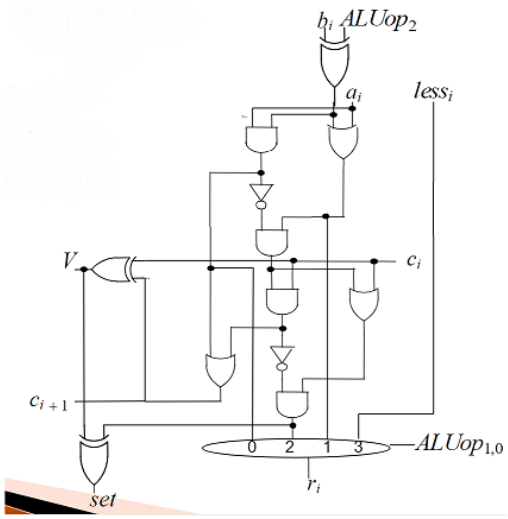


As for 4 input plexer we used our previous modules to come up with a design.

- **1 BITE ALU:**



- **MSB BITE ALU:**



ADD ONS:

A new signal called bracnneq added. This signal is for to detect bne instruction “and” it is and logiced with the “not” logiced ALU’s zero output. This will decide not only for two different branch instruction, it will set the configurations to the PC for branch.

New R-type instructions are handled inside register module with behavioral verilog.

OUTPUT:

- MUX 32 BIT

```
# time =  0, i0 = 00000000000000000000000000001000, i1 = 00000000000000000000000000000100, sel = 0, res = 00000000000000000000000000001000
# time = 10, i0 = 00000000000000000000000000001000, i1 = 00000000000000000000000000000100, sel = 1, res = 00000000000000000000000000000100
# time = 20, i0 = 00000000000000000000000000001000, i1 = 00000000001000110001101010001000, sel = 0, res = 00000000000000000000000000001000
# time = 30, i0 = 00000000000000000000000000001000, i1 = 00000000001000110001101010001000, sel = 1, res = 00000000001000110001101010001000
# time = 40, i0 = 00000000000000001110000000001000, i1 = 00000000001000110001101010001000, sel = 0, res = 00000000000000001110000000001000
# time = 50, i0 = 00000000000000001110000000001000, i1 = 00000000001000110001101010001000, sel = 1, res = 00000000001000110001101010001000
```

- MUX 4X1

```
# time =  0, a = 0, b = 0, c = 0, d = 0, sel = 00 out = 0
# time =  5, a = 0, b = 1, c = 0, d = 1, sel = 10 out = 0
# time = 10, a = 0, b = 1, c = 0, d = 1, sel = 01 out = 1
# time = 15, a = 0, b = 1, c = 0, d = 1, sel = 11 out = 1
```

- MUX 2X1

```
# time =  0, a = 0, b = 0, sel = 0, out = 0
# time =  5, a = 1, b = 0, sel = 0, out = 1
# time = 10, a = 0, b = 1, sel = 0, out = 0
# time = 15, a = 1, b = 1, sel = 0, out = 1
# time = 20, a = 0, b = 0, sel = 1, out = 0
# time = 25, a = 1, b = 0, sel = 1, out = 0
# time = 30, a = 0, b = 1, sel = 1, out = 1
# time = 35, a = 1, b = 1, sel = 1, out = 1
```

- MSB ALU

```
# time =  0, a = 0, b = 0, ci = 0, less = 0,op = 000, r = z, co = z, v = z
# time = 20, a = 1, b = 0, ci = 0, less = 0,op = 000, r = z, co = z, v = z
# time = 40, a = 0, b = 1, ci = 0, less = 0,op = 000, r = z, co = z, v = z
# time = 60, a = 1, b = 1, ci = 0, less = 0,op = 000, r = z, co = z, v = z
# time = 80, a = 0, b = 0, ci = 0, less = 0,op = 001, r = z, co = z, v = z
# time = 100, a = 1, b = 0, ci = 0, less = 0,op = 001, r = z, co = z, v = z
# time = 120, a = 0, b = 1, ci = 0, less = 0,op = 001, r = z, co = z, v = z
# time = 140, a = 1, b = 1, ci = 0, less = 0,op = 001, r = z, co = z, v = z
# time = 160, a = 0, b = 0, ci = 0, less = 0,op = 010, r = z, co = z, v = z
# time = 180, a = 0, b = 0, ci = 1, less = 0,op = 010, r = z, co = z, v = z
# time = 200, a = 1, b = 0, ci = 0, less = 0,op = 010, r = z, co = z, v = z

# time = 220, a = 1, b = 0, ci = 1, less = 0,op = 010, r = z, co = z, v = z
# time = 240, a = 0, b = 1, ci = 0, less = 0,op = 010, r = z, co = z, v = z
# time = 260, a = 0, b = 1, ci = 1, less = 0,op = 010, r = z, co = z, v = z
# time = 280, a = 1, b = 1, ci = 0, less = 0,op = 010, r = z, co = z, v = z
# time = 300, a = 1, b = 1, ci = 1, less = 0,op = 010, r = z, co = z, v = z
# time = 320, a = 0, b = 0, ci = 0, less = 0,op = 110, r = z, co = z, v = z
# time = 340, a = 0, b = 0, ci = 1, less = 0,op = 110, r = z, co = z, v = z
# time = 360, a = 1, b = 0, ci = 0, less = 0,op = 110, r = z, co = z, v = z
# time = 380, a = 1, b = 0, ci = 1, less = 0,op = 110, r = z, co = z, v = z
# time = 400, a = 0, b = 1, ci = 0, less = 0,op = 110, r = z, co = z, v = z

# time = 420, a = 0, b = 1, ci = 1, less = 0,op = 110, r = z, co = z, v = z
# time = 440, a = 1, b = 1, ci = 0, less = 0,op = 110, r = z, co = z, v = z
# time = 460, a = 1, b = 1, ci = 1, less = 0,op = 110, r = z, co = z, v = z
# time = 480, a = 1, b = 1, ci = 1, less = 1,op = 111, r = z, co = z, v = z
# time = 500, a = 1, b = 1, ci = 1, less = 0,op = 111, r = z, co = z, v = z
```

- DATA MEMORY

```
#  time =  0, addr = 00000000000000000000000010011110, data in = 10100101010101001100100110111010, memrd = 1, memwr = 0,
#  data read = 00000000000000000000000000011110
#  time = 10, addr = 00000000000000000000000001100100, data in = 00000101010100101000110110111010, memrd = 1, memwr = 0,
#  data read = 00000000000000000000000000000100
#  time = 20, addr = 00000000000000000000000001100100, data in = 00000101010100101000110110111010, memrd = 0, memwr = 1,
#  data read = 00000000000000000000000000000100
#  time = 30, addr = 00000000000000000000000001100100, data in = 00000101010100101000110110111010, memrd = 1, memwr = 0,
#  data read = 00000101010100101000110110111010
#  time = 40, addr = 00000000000000000000000000110111, data in = 00000101010100001011001001000101, memrd = 1, memwr = 0,
#  data read = 00000000000000000000000000010111
#  time = 50, addr = 00000000000000000000000000110111, data in = 00000101010100001011001001000101, memrd = 0, memwr = 1,
#  data read = 00000000000000000000000000010111
#  time = 60, addr = 00000000000000000000000000110111, data in = 00000101010100001011001001000101, memrd = 1, memwr = 0,
#  data read = 00000101010100001011001001000101
#  time = 70, addr = 00000000000000000000000000011110, data in = 00000101010100001011001001000101, memrd = 1, memwr = 0,
#  data read = 00000000000000000000000000011110
#  time = 80, addr = 00000000000000000000000000011110, data in = 00000000000000000000000000000001, memrd = 0, memwr = 1,
#  data read = 00000000000000000000000000011110
#  time = 90, addr = 00000000000000000000000000011110, data in = 00000101010100001011001001000101, memrd = 1, memwr = 0,
#  data read = 00000000000000000000000000000001
#  time = 100, addr = 00000000000000000000000000001010, data in = 00000000000000000000000000000000, memrd = 1, memwr = 0,
#  data read = 00000000000000000000000000001010
```

- CONTROL UNIT

```
# time =  0, inst = 100011, RegWrite = 1, ALUSrc = 1, MemtoReg = 1, MemRead = 1, MemWrite = 0, Branch = 0, Branchnot = 0, j/jal = 0, jr = 0 Aluop = 00
# time = 40, inst = 101011, RegWrite = 0, ALUSrc = 1, MemtoReg = 0, MemRead = 0, MemWrite = 1, Branch = 0, Branchnot = 0, j/jal = 0, jr = 0 Aluop = 00
# time = 80, inst = 000010, RegWrite = 0, ALUSrc = 0, MemtoReg = 0, MemRead = 0, MemWrite = 0, Branch = 0, Branchnot = 0, j/jal = 1, jr = 0 Aluop = 00
# time = 160, inst = 001000, RegWrite = 0, ALUSrc = 0, MemtoReg = 0, MemRead = 0, MemWrite = 0, Branch = 0, Branchnot = 0, j/jal = 0, jr = 1 Aluop = 00
# time = 200, inst = 000100, RegWrite = 0, ALUSrc = 0, MemtoReg = 0, MemRead = 0, MemWrite = 0, Branch = 1, Branchnot = 0, j/jal = 0, jr = 0 Aluop = 01
# time = 240, inst = 000101, RegWrite = 0, ALUSrc = 0, MemtoReg = 0, MemRead = 0, MemWrite = 0, Branch = 0, Branchnot = 1, j/jal = 0, jr = 0 Aluop = 00
# time = 280, inst = 000000, RegWrite = 1, ALUSrc = 0, MemtoReg = 0, MemRead = 0, MemWrite = 0, Branch = 0, Branchnot = 0, j/jal = 0, jr = 0 Aluop = 10
# time = 320, inst = 001101, RegWrite = 0, ALUSrc = 0, MemtoReg = 0, MemRead = 0, MemWrite = 0, Branch = 0, Branchnot = 0, j/jal = 0, jr = 0 Aluop = 00
```

- ALU CONTROL

```
# time =  0, func = 100000, aluop = 10, aluctr = 010
# time = 10, func = 100110, aluop = 10, aluctr = 100
# time = 20, func = 100100, aluop = 10, aluctr = 000
# time = 30, func = 100101, aluop = 10, aluctr = 001
```