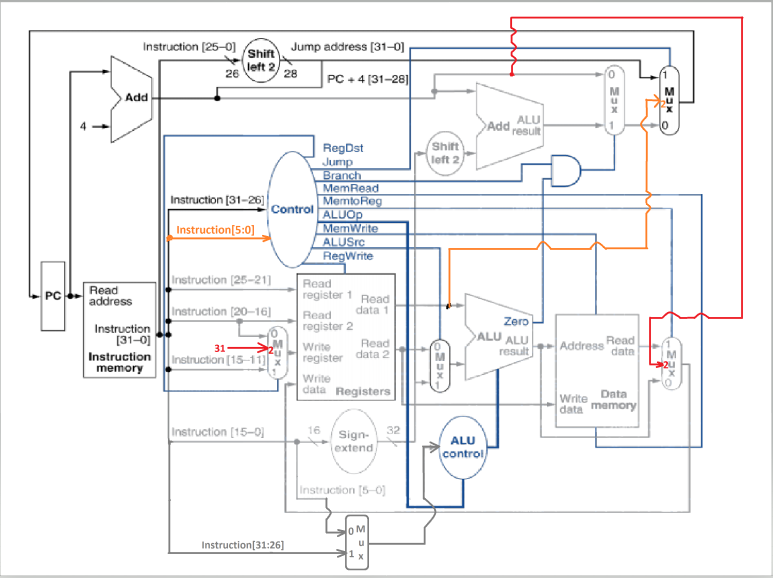
REPORT



**RED EXTENSION:** JAL Implementation

**ORANGE EXTENSION:** JR Implementation

**GRAY EXTENSION:** Choice for Opcode or Funct is going to be passed to ALU Control

When all the instructions are written to the instructions txt, it is not working properly. Examined it but did not find to do it. Apologize, but it is running only when the instructions are written lonely.

Some test captures added to the file to show you the results.

**Control Unit:**  The signal values for the desired operations are assigned to the parameters. Decision made accordingly to the opcodes of the operations. Additionally jump signal and funct parameter added for jump operations to use it in multiplexer.

**ALU Control:** The isBgez parameter added to separate BGEZ and BLTZ operations. Size of gout extended to 4 to create more option. The existing before parameters are stores funct field for R type, and stores opcode field for I type operations.

**Processor:** A new multiplexer added to the processor to decide whether the operation’s opcode or funct field is going to be passed to the alucont . Some of the existing multiplexers are overwritten to give 3 options to assign for the selector. Also, given datapath in lecture presentation Processors for the Jump operation implemented. Jump address field created.

**ADD**

Instruction: 000000 00011 00100 00101 00000 100000

Binary: 0000 0000 0110 0100 0010 1000 0010 0000

Hexadecimal: 00 64 28 20

**SUB**

Instruction: 000000 00011 00100 00101 00000 100010

Binary: 0000 0000 0110 0100 0010 1000 0010 0010

Hexadecimal: 00 64 28 22

**ADDI**

Instruction: 001000 00011 00100 00101 00000 100010

Binary: 0010 0000 0110 0100 0000 0000 0000 0001

Hexadecimal: 20 64 00 01

**ANDI**

Instruction: 001100 00011 00100 00000 00001 010001

Binary: 0011 0000 0110 0100 0000 0000 0101 0001

Hexadecimal: 30 64 00 51

**ORI**

Instruction: 001101 00011 00100 00000 00001 010001

Binary: 0011 0100 0110 0100 0000 0000 0101 0001

Hexadecimal: 34 64 00 51

**NOR**

Instruction: 000000 00011 00100 00101 00000 000001

Binary: 0000 0000 0110 0100 0010 1000 0000 0001

Hexadecimal: 00 64 28 01

**BNE**

Instruction: 000101 00011 00100 00000 00000 000001

Binary: 0001 0100 0110 0100 0000 0000 0000 0001

Hexadecimal: 14 64 00 01

**BGEZ**

Instruction: 000001 00011 00101 00000 00000 000001

Binary: 0000 0100 0110 0101 0000 0000 0000 0001

Hexadecimal: 04 65 00 01

**BGTZ**

Instruction: 000111 00011 00101 00000 00000 000001

Binary: 0001 1100 0110 0100 0000 0000 0000 0001

Hexadecimal: 1C 64 00 01

**BLEZ**

Instruction: 000101 00011 00101 00000 00000 000001

Binary: 0001 0100 0110 0100 0000 0000 0000 0001

Hexadecimal: 18 64 00 01

**BLTZ**

Instruction: 000001 00011 00101 00000 00000 000001

Binary: 0000 0100 0110 0100 0000 0000 0000 0001

Hexadecimal: 04 64 00 01

**J**

Instruction: 000001 10000 00000 00000 00000 000001

Binary: 0000 1100 0000 0000 0000 0000 0000 0001

Hexadecimal: 0C 00 00 01

**JR**

Instruction: 000001 10000 00000 00000 00000 000001

Binary: 0000 0000 0110 0100 0000 0000 0000 1000

Hexadecimal: 00 64 00 08

**JAL**

Instruction: 000001 10000 00000 00000 00000 000001

Binary: 0000 1100 0000 0000 0000 0000 0000 0010

Hexadecimal: 0C 00 00 02

Yusuf Alptuğ Aydın

260201065