ICS 233

Project

Due Tuesday, May 10, 2022

Names: Esam Mohammed Jaafar

ID’s: 201751810

Section Number: 51,52

Instructor: Ibrahim Alnemer

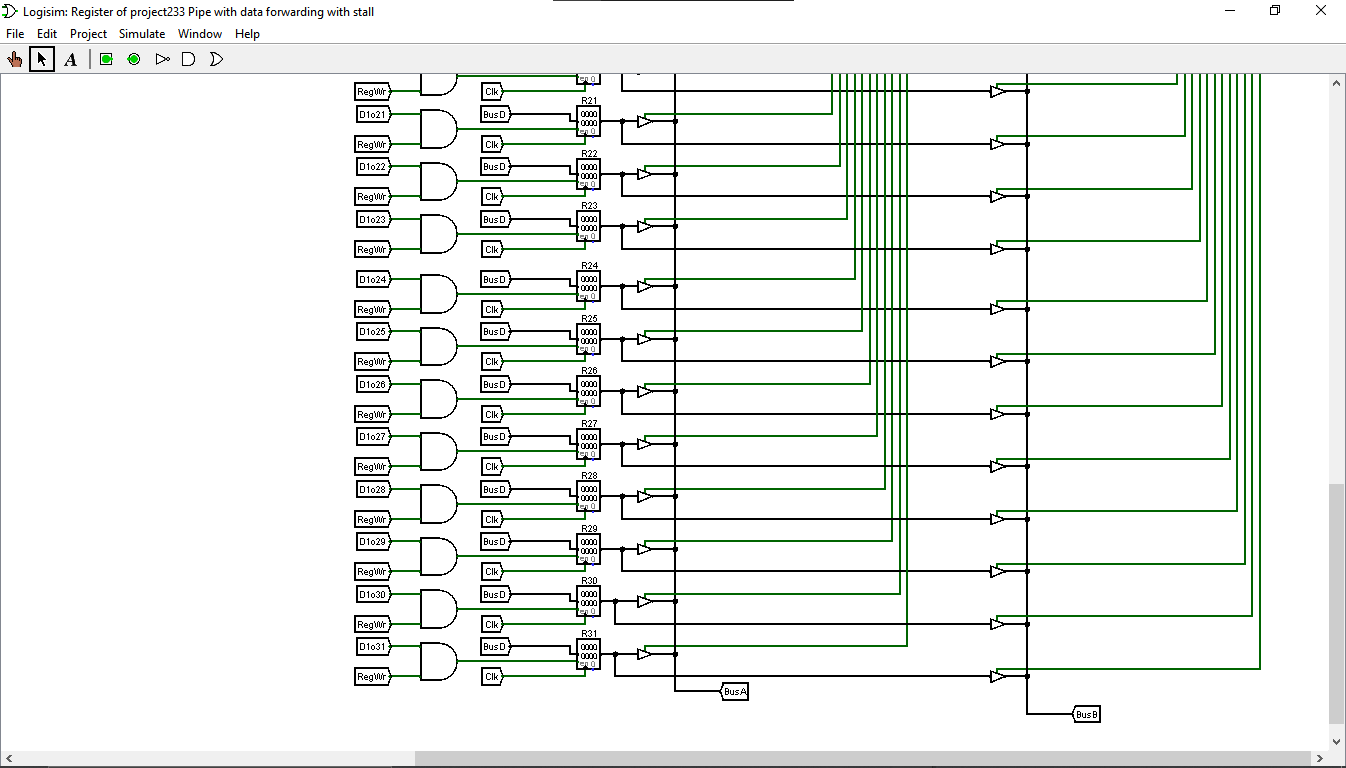
# Design and Implementation

1. In the design of our processor, we made sure to Implement most of the traditional design given while exploring some new ideas to enhance it and sometime get solutions in unique way for our design. Therefore, we tried to increase the number of components of circuits to make it more understandable by reducing wires and size in each one. We tried to change the way the branch is taken by using the op code bit and the zero flag to cover each branch given to us reducing signal input and circuits.
2. For the design drawings we will give most recent drawings as for below:Diagram, schematic

   Description automatically generatedDiagram, schematic

   Description automatically generatedA picture containing bar chart

   Description automatically generatedDiagram, schematic

   Description automatically generated

Diagram, schematic

Description automatically generatedDiagram, schematic

Description automatically generatedDiagram

Description automatically generatedDiagram, schematic

Description automatically generated

1. As for the control logic, signals, and forwarding for each instruction can been seen below for its description and values

|  |  |  |  |
| --- | --- | --- | --- |
| **Instruction** | | Meaning | |
| SLL | Rd=Shift Left Logical (Ra, Rb [4:0]) | | |
| SRL | Rd=Shift Right Logical (Ra, Rb [4:0]) | | |
| SRA | Rd=Shift Right Arithmetic (Ra, Rb [4:0]) | | |
| ROR | Rd=Rotate Right (Ra, Rb [4:0]) | | |
| ADD | Rd=Ra + Rb | | |
| SUB | Rd=Ra – Rb | | |
| SLT | Rd= (Ra < Rb) signed | | |
| SLTU | Rd= (Ra < Rb) unsigned | | |
| XOR | Rd= Ra ^ Rb | | |
| OR | Rd= Ra | Rb | | |
| AND | Rd= Ra & Rb | | |
| NOR | Rd = ~ (Ra | Rb) | | |
| MUL | Rd = (Ra \* Rb) [31:00] | | |
| SLLI | Rd=Shift Left Logical (Ra, sa) | | |
| SRLI | Rd=ShiftRightLogical(Ra, sa) | | |
| SRAI | Rd=ShiftRightArith(Ra, sa) | | |
| RORI | Rd=RotateRight(Ra, sa) | | |
| ADDI | | | Rd=Ra + sign\_extend(imm16) |
| SLTI | | | Rd=(Ra < sign\_extend(imm16)) signed |
| SLTIU | | | Rd=(Ra < sign\_extend(imm16)) unsigned |
| XORI | | | Rd=Ra ^ zero\_extend(imm16) |
| ORI | | | Rd=Ra | zero\_extend(imm16) |
| ANDI | | | Rd=Ra & zero\_extend(imm16) |
| NORI | | | Rd = ~(Ra | zero\_extend(imm16)) |
| LUI | | | Rd=imm16<<16 |
| JALR | | | PC=Ra+sign\_extend(imm16), Rd=PC+1 |
| LW | | | Rd=Mem[Ra+sign\_extend(imm16)] |
| SW | Mem[Ra+sign\_extend({imm11U, imm5L})]=Rb | | |
| BEQ | if (Ra == Rb)  PC=PC+sign\_extend({imm11U, imm5L}) | | |
| BNE | if (Ra != Rb)  PC=PC+sign\_extend({imm11U, imm5L}) | | |
| BLT | if (Ra < Rb)  PC=PC+sign\_extend({imm11U, imm5L}) | | |
| BGE | if (Ra >= Rb)  PC=PC+sign\_extend({imm11U, imm5L}) | | |
| BLTU | if (Ra < Rb) unsigned  PC=PC+sign\_extend({imm11U, imm5L}) | | |
| BGEU | if (Ra >= Rb) unsigned PC=PC+sign\_extend({imm11U, imm5L}) | | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| N | Instructions | selBusB | MemRead | MemWrite | RegWrite | BusWrite | Opcode | aluOp | function |
| 1 | SLL | 0 | 0 | 0 | 1 | 0 | 0 | 16 | 0 |
| 2 | SRL | 0 | 0 | 0 | 1 | 0 | 0 | 17 | 1 |
| 3 | SRA | 0 | 0 | 0 | 1 | 0 | 0 | 18 | 2 |
| 4 | ROR | 0 | 0 | 0 | 1 | 0 | 0 | 19 | 3 |
| 5 | ADD | 0 | 0 | 0 | 1 | 0 | 0 | 20 | 4 |
| 6 | SUB | 0 | 0 | 0 | 1 | 0 | 0 | 21 | 5 |
| 7 | SLT | 0 | 0 | 0 | 1 | 0 | 0 | 22 | 6 |
| 8 | SLTU | 0 | 0 | 0 | 1 | 0 | 0 | 23 | 7 |
| 9 | XOR | 0 | 0 | 0 | 1 | 0 | 0 | 24 | 8 |
| 10 | OR | 0 | 0 | 0 | 1 | 0 | 0 | 25 | 9 |
| 11 | AND | 0 | 0 | 0 | 1 | 0 | 0 | 26 | 10 |
| 12 | NOR | 0 | 0 | 0 | 1 | 0 | 0 | 27 | 11 |
| 13 | MUL | 0 | 0 | 0 | 1 | 0 | 0 | 28 | 12 |
| 14 | SLLI | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 15 | SRLI | 1 | 0 | 0 | 1 | 0 | 2 | 1 | 0 |
| 16 | SRAI | 1 | 0 | 0 | 1 | 0 | 3 | 2 | 0 |
| 17 | RORI | 1 | 0 | 0 | 1 | 0 | 4 | 3 | 0 |
| 18 | ADDI | 1 | 0 | 0 | 1 | 0 | 5 | 4 | 0 |
| 19 | SLTI | 1 | 0 | 0 | 1 | 0 | 6 | 6 | 0 |
| 20 | SLTIU | 1 | 0 | 0 | 1 | 0 | 7 | 7 | 0 |
| 21 | XORI | 1 | 0 | 0 | 1 | 0 | 8 | 8 | 0 |
| 22 | ORI | 1 | 0 | 0 | 1 | 0 | 9 | 9 | 0 |
| 23 | ANDI | 1 | 0 | 0 | 1 | 0 | 10 | 10 | 0 |
| 24 | NORI | 1 | 0 | 0 | 1 | 0 | 11 | 11 | 0 |
| 25 | LUI | 1 | 0 | 0 | 1 | 0 | 12 | 12 | 0 |
| 26 | JALR | X | 0 | 0 | X | 2 | 13 | 13 | 0 |
| 27 | LW | 0 | 1 | 0 | 1 | 0 | 14 | 14 | 0 |
| 28 | SW | 0 | 0 | 1 | 1 | 0 | 15 | 15 | 0 |
| 29 | BEQ | 0 | 0 | 0 | 0 | 0 | 16 | 4 | 0 |
| 30 | BNE | 0 | 0 | 0 | 0 | 0 | 17 | 4 | 0 |
| 31 | BLT | 0 | 0 | 0 | 0 | 0 | 18 | 5 | 0 |
| 32 | BGT | 0 | 0 | 0 | 0 | 0 | 19 | 5 | 0 |
| 33 | BLTU | 0 | 0 | 0 | 0 | 0 | 20 | 6 | 0 |
| 34 | BGTU | 0 | 0 | 0 | 0 | 0 | 21 | 6 | 0 |

# Simulation and Testing

# Teamwork

**Essam Jaffar:**

* **Enhancing design readability, writing the report, and managing project meeting and schedules**

**Sultan Alkunian:**

* **Writing test cases, and problem reporting.**

**Ali Salih:**

* **Designing the circuit, writing follow up questions**