COMP 303 TERM PROJECT REPORT

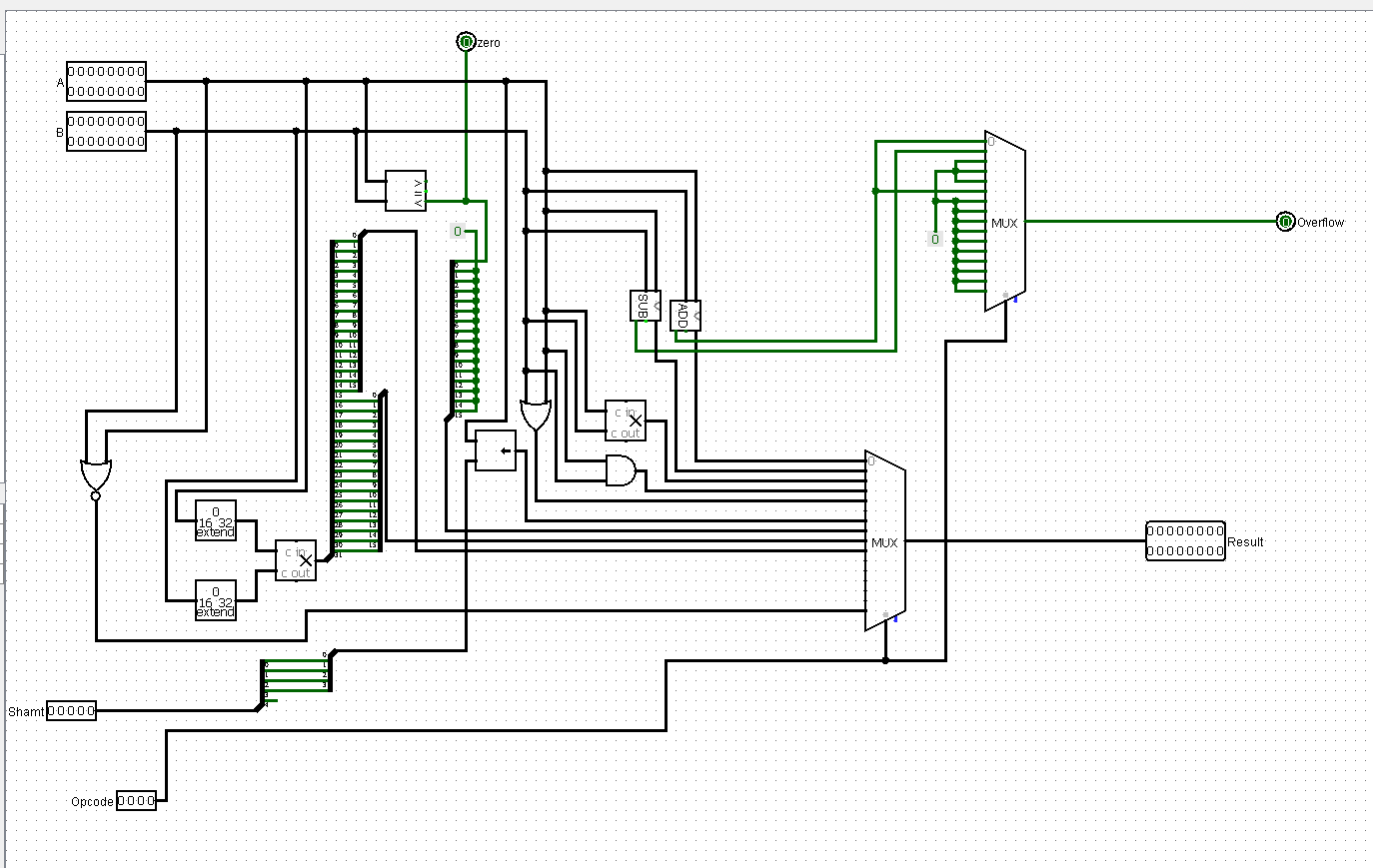
ALU

We designed an ALU to calculate the selected instruction based on opcode. It gets opcode from control unit ALUop signal. Control unit specifies the opcode according to the instruction types.

For example when opcode is LOAD, ALU must perform add operation, therefore ALUop n control unit change opcode of load with ADD’s opcode to enable ALU to execute add.

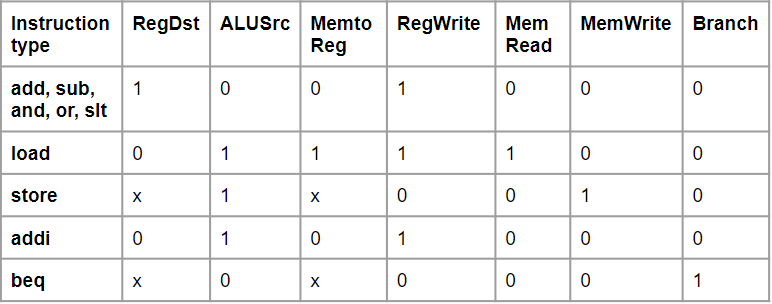
|  |  |
| --- | --- |
| OPCODE | INSTRUCTION |
| 000000 | ADD |
| 000001 | SUB |
| 000010 | MULT |
| 000011 | AND |
| 000100 | OR |
| 000101 | ADDI |
| 000110 | SLL |
| 000111 | SLT |
| 001000 | MFHI |
| 001001 | MFLO |
| 001010 | LW |
| 001011 | SW |
| 001100 | BEQ |
| 001101 | BLEZ |
| 001110 | JUMP |
| 001111 | NOR (extra instruction) |

LOGISIM Design ALU

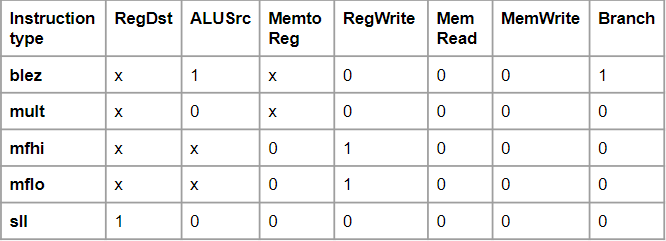


CONTROL UNIT

Opcode, first two bits decide on instruction formats (R-type, I-type) and our own operation. After then, rest of 4 bits complete the function of chosen. The signal generation through selected function showing in the control signals table.

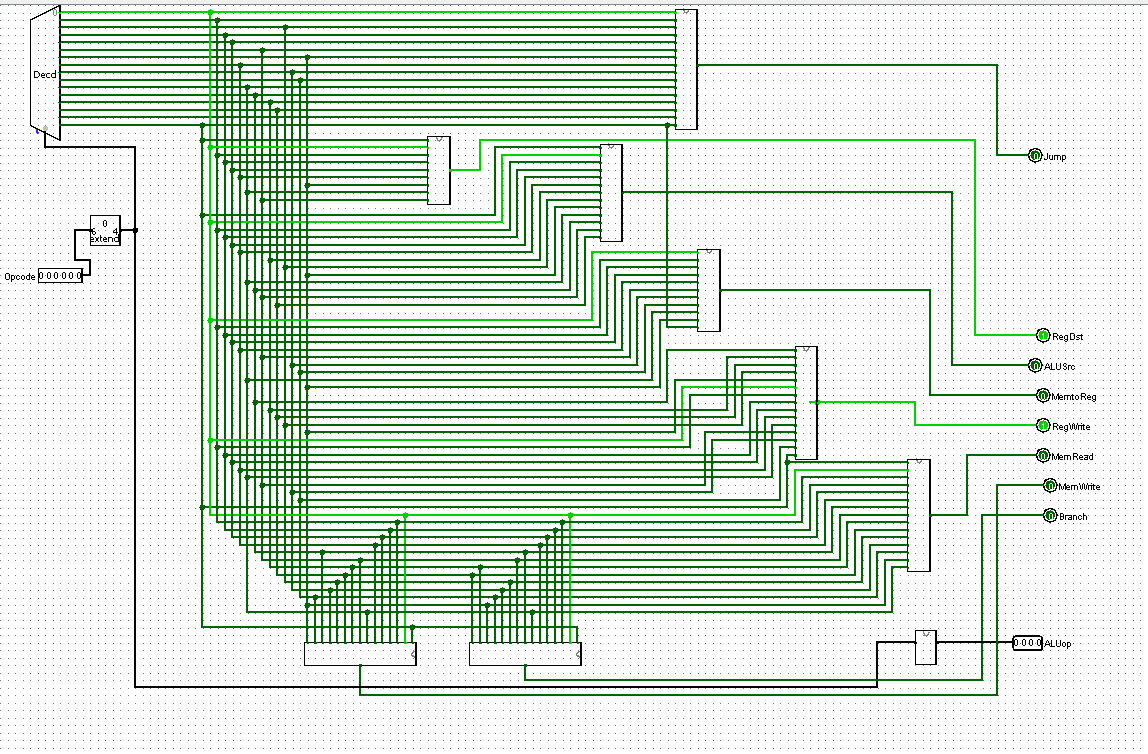


As an example of this, in terms of using ‘sub’ function, branch value is getting low.



When a function is selected, the control unit changes the ALU opcode for each function.

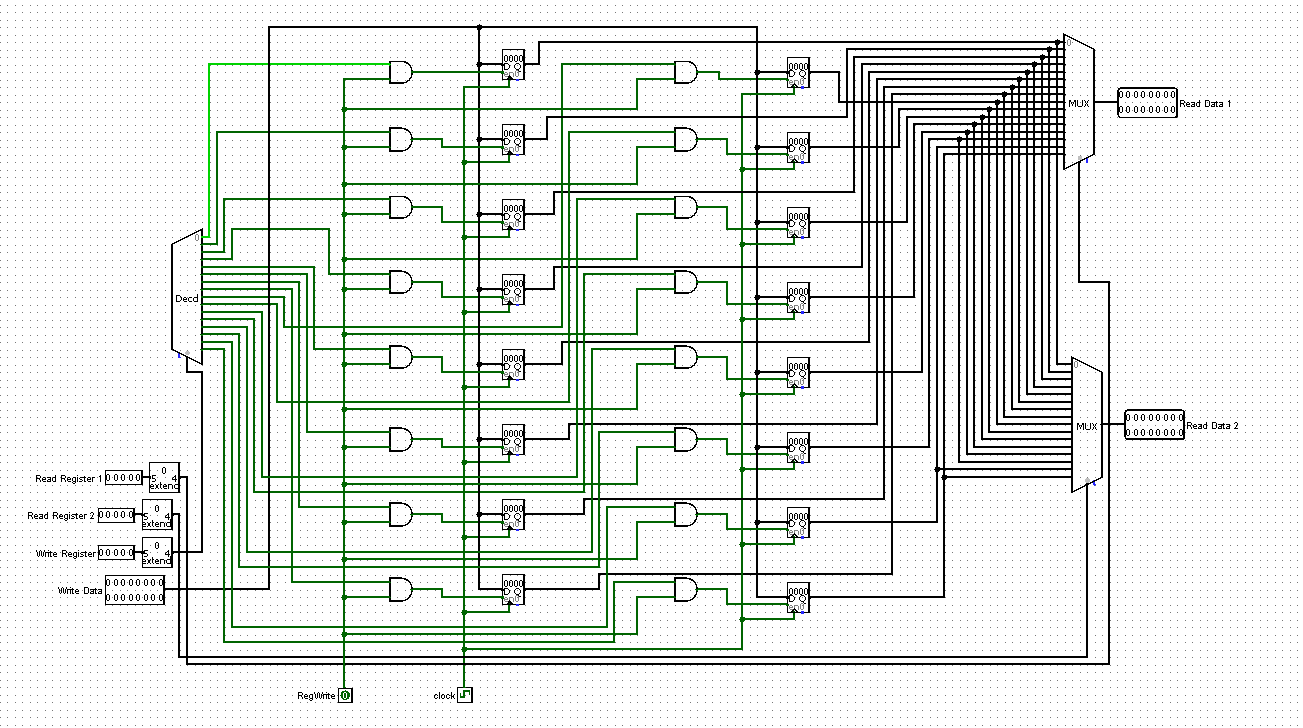
LOGISIM Control Unit Design



REGISTER FILE

We designed a register file with two read register address ports, two read register output ports, one write register address port, one write register data port, a register write control signal and a clock input. The purpose of the register file is to read simultaneously from two registers and write into one register.

LOGISIM Register File Design



INSTRUCTION MEMORY

Instruction memory reads instructions. Each clock cycle fetches the instruction from the address specified by the PC, and increments PC by 1 at the same time.We used RAM module to design instruction memory.

DATA MEMORY

Reads memory or writes to memory according to control unit MemRead and MemWrite signals.

We used RAM module to design data memory.

LOGISIM CPU Design

