MIPS single cycle implementation:

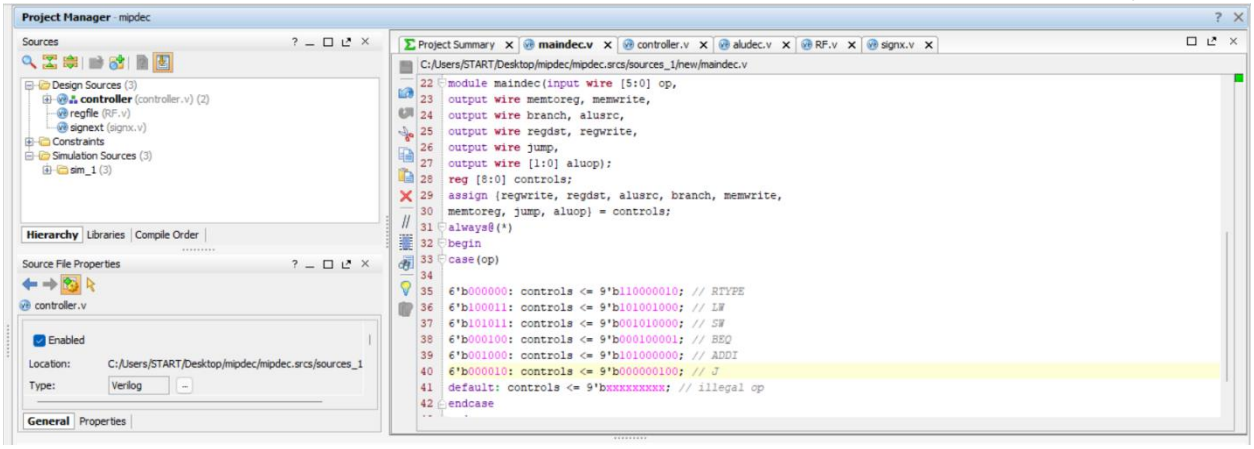
Control Unit:

Control unit inputs:

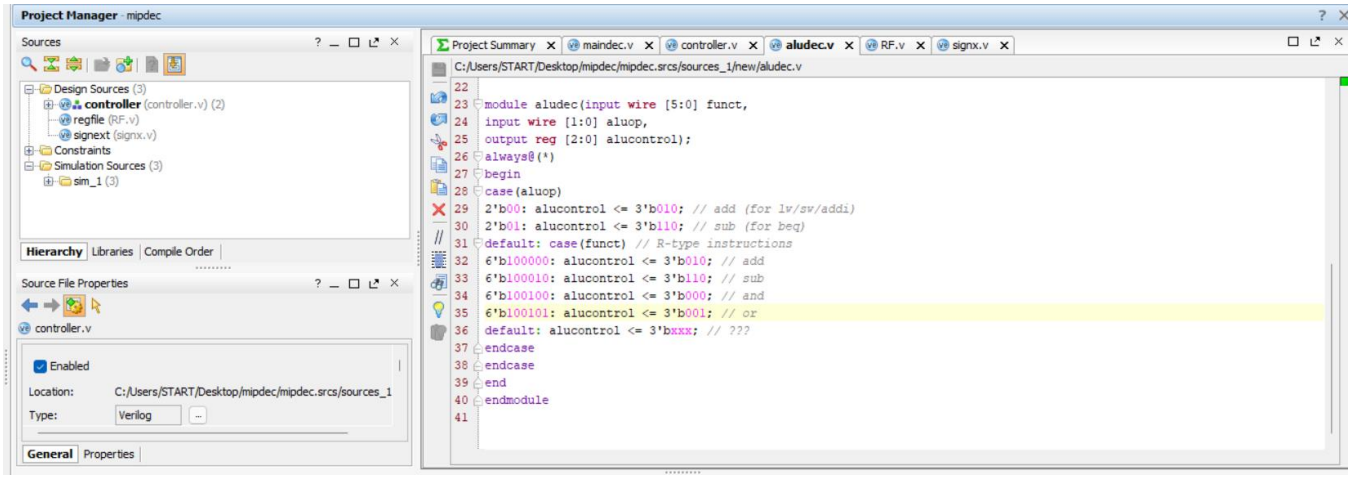
- 6 bits opcode ([31:26] bits of the output instruction)
  - R-type opcodes are: 6'b000000
  - LW opcode: 6'b100011
  - SW opcode: 6'b101011
  - Addi opcode: 6'b001000
  - J opcode: 6'b000010

- 6 bits func ([5:0] bits of the instruction)
  - Only used to decode R-Type instructions
  - Add Funct: 6'b100000
  - Sub Funct: 6'b100010
  - And Funct: 6'b100100
  - Or Funct: 6'b100101

First, we can create a module to handle the operation code to determine the type of instr and according to its type we can determine the control signals of our components the control signals are regwrite, regdst, alusrc, branch, memwrite, memtoreg, jump, aluop (aluop is further decoded to finally reach the ALU SRC)



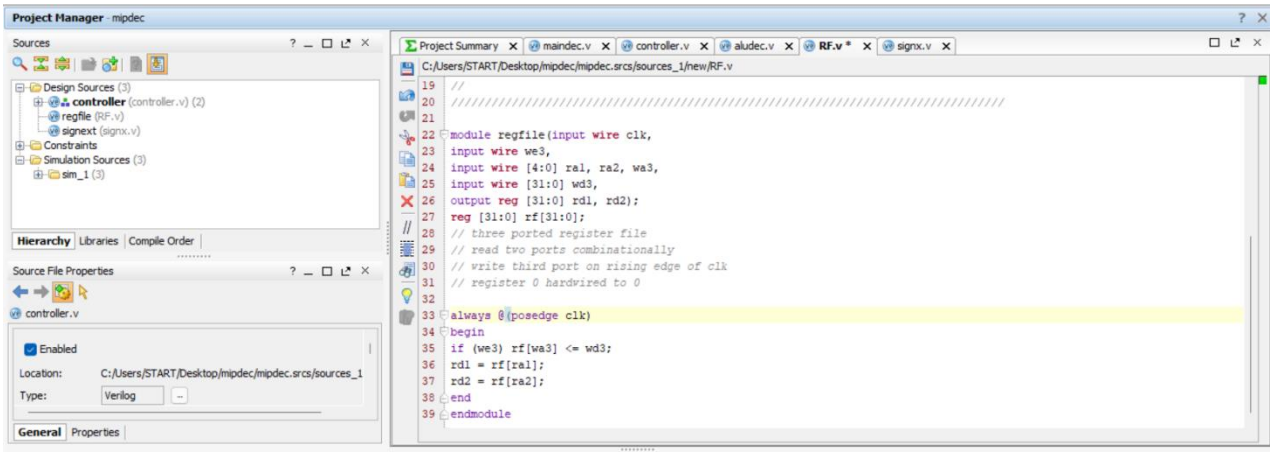
We can create a module specifically to handle the Aluop resulted from the main decoder which differentiates between



This module first checks the aluop it receives from the main decoder module. this signal is 2'b10 if the instr is R-type and is 2'b00 if the instr is a LW or a SW instruction which both require the ALU to perform addition.

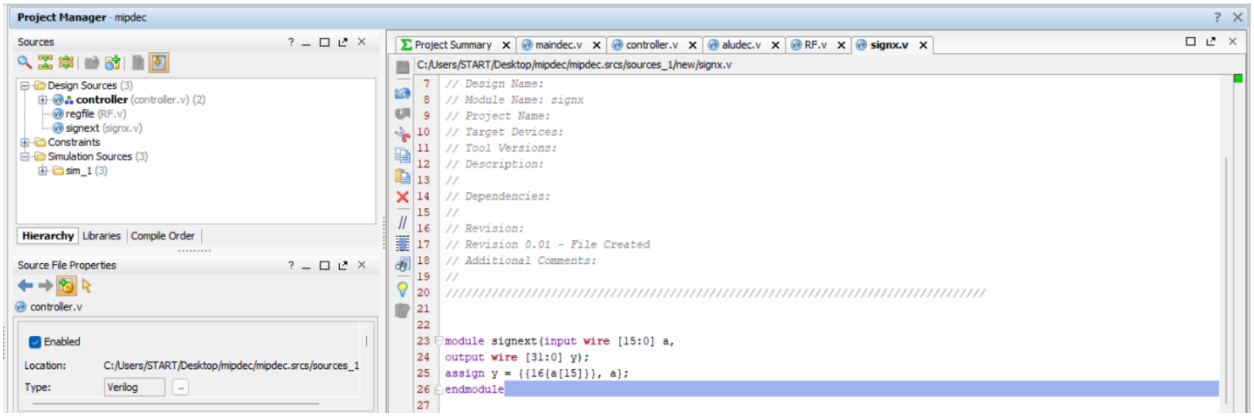
Register File:

- ra1
  - 5-bit input to determine our first source register
- ra2
  - 5-bit input to determine our second source register
- wa3
  - 5-bit input to determine our destination register
- wd3
  - 32-bit data required to be stored
- we3
  - control signal to determine whether we will write the wd input in the register determined by the wa3 input
- clk
- rd1
  - 32 -bit output of the value stored in the register determined by the ra1 input signal
- rd2
  - 32-bit output of the value stored in the register determined by the ra1 input signal



Sign Extend:

- a: 16-bit input of an immediate value
- b: 32-bit output of the same value of a



Data memory:

- clk
- we: we enable signal (1 when using store instruction, 0 otherwise)
- wd: 32-bit input data which is required to be stored
- a: 32-bit signal to determine the address in the memory to either read its data or write the wd in it
- rd: 32-bit output of the data in location (a)

modify the Register File module code