FILES

Data.mem instruction.mem registers.mem

NEW MODULES

SignExtendImm_16bit_to_32bit it does { 16{ immediate[15]}, immediate[15:0]}

ZeroExtendImm_16bit_to_32bit it does { 16b0, immediate[15:0] }

_8to32 it does { 24b0, immediate[7:0] }

mux32bit_4_1(result, S1, S0, I3, I2, I1, I0)

mux32bit_2_1(result, S0, I1, I0)

Control_Unit(signal_RegWrite, signal_ALUSrc, signal_MemtoReg1, signal_MemtoReg2, signal_MemRead, signal_MemWrite, signal_sb, signal_sh, signal_sw, OpCode)

Registers_Unit_32bit(Read_data_1, Read_data_2, Read_reg_1, Read_reg_2, Write_register, Write_data, signal_RegWrite, clk)

Data_Memory_Unit_32bit (Read_data, memory_adress, write_data, read_signal, write_signal, signal_sb, signal_sh, signal_sw, clk)

Instruction_Memory_Unit_32bit(Instruction, PC)

NOTES

All test benches are right and they test their own modules except mips32bit module,
it has some failures like sometimes right x to registers, it reads right but when it try to write it is corrupt ı dont know why .

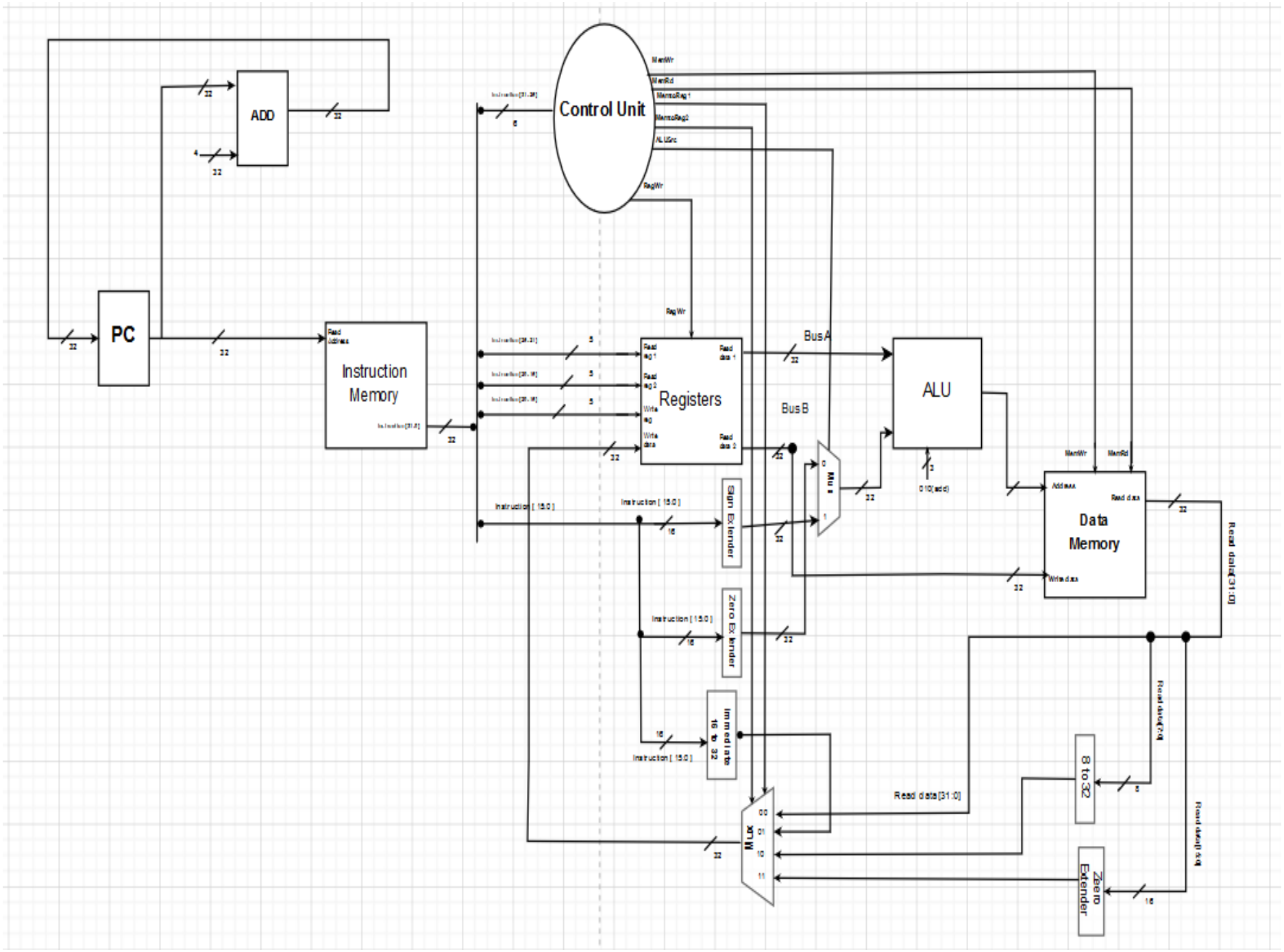
WARNING

You have to change the path of reading or writing file, if yo do not change it in register, instruction, data memory unit program will not executed.

Example:

$readmemb("C:/altera/13.1/workspace/org/hw3/registers.mem", Registers)

$writememb("C:/altera/13.1/workspace/org/hw3/registers.mem", Registers)


ADD
Control Unit
PC
Instruction Memory
Registers
ALU
Data Memory
Sign Extender
Zero Extender
Immediate 16 to 32
8 to 32
Zero Extender
MemWr
MemRd
MemtoReg1
MemtoReg2
ALUSrc
RegWr
Bus A
Bus B
Instruction [15:0]
Read data[31:0]