// ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

// Module : MIPS Micro Processor

// Programmer : Ngan Ngoc Pham

// Description :

// This is an implementation of the RISC MIPS Micro Processor.

// It is implemented based on the model described in "Computer Organization

// and Design by Patterson & Hennessey, 2nd edition, Chapter 5

// ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

`include "controller.v"

`include "datapath.v"

`include "memory.v"

module MIPS ( Clock, // System Clock

Resetn ); // System Reset

// Port Declaration

input Clock, Resetn;

// Internal Nets and wires

wire PCWriteCond, PCWrite, IorD, MemRead, MemWrite, MemtoReg, IRWrite;

//!add

wire [31:0] MemData,Address,WriteData;

wire CauseWrite, IntCause, EPCWrite, ALUSrcA, RegWrite, RegDst, LoadByte; // CHANGE

wire Overflow;

wire [1:0] PCSource, ALUOp, ALUSrcB;

wire [5:0] Opcode;

// Instantiate a controller

controller CONTROLLER ( .Clock(Clock),

.Resetn(Resetn),

.Overflow(Overflow),

.Op(Opcode),

.PCWriteCond(PCWriteCond),

.PCWrite(PCWrite),

.IorD(IorD),

.MemRead(MemRead),

.MemWrite(MemWrite),

.MemtoReg(MemtoReg),

.IRWrite(IRWrite),

.CauseWrite(CauseWrite),

.IntCause(IntCause),

.EPCWrite(EPCWrite),

.PCSource(PCSource),

.ALUOp(ALUOp),

.ALUSrcB(ALUSrcB),

.ALUSrcA(ALUSrcA),

.RegWrite(RegWrite),

.RegDst(RegDst),

.LoadByte(LoadByte) // CHANGE

);

// Instatiate a datapath

datapath DATAPATH ( .Clock(Clock),

.Resetn(Resetn),

.PCWriteCond(PCWriteCond),

.PCWrite(PCWrite),

.IorD(IorD),

.MemRead(MemRead),

.MemWrite(MemWrite),

.MemtoReg(MemtoReg),

.MemData(MemData),

.Address(Address),

.WriteData(WriteData),

.IRWrite(IRWrite),

.CauseWrite(CauseWrite),

.IntCause(IntCause),

.EPCWrite(EPCWrite),

.PCSource(PCSource),

.ALUOp(ALUOp),

.ALUSrcB(ALUSrcB),

.ALUSrcA(ALUSrcA),

.RegWrite(RegWrite),

.RegDst(RegDst),

.Opcode(Opcode),

.overflowRegOut(Overflow),

.LoadByte(LoadByte) // CHANGE

);

Memory MEMORY (.MemData(MemData),

.Address(Address),

.WriteData(WriteData),

.MemRead(MemRead),

.MemWrite(MemWrite));

endmodule