```
module CPU (clock);
parameter LW = 6'b100011, SW = 6'b101011, BEQ=6'b000100, J=6'd2;
input clock; //the clock is an external input
// The architecturally visible registers and scratch registers for implementation
reg [31:0] PC, Regs[0:31], Memory [0:1023], IR, ALUOut, MDR, A, B;
reg [2:0] state; // processor state
wire [5:0] opcode; //use to get opcode easily
wire [31:0] SignExtend,PCOffset; //used to get sign-extended offset field
assign opcode = IR[31:26]; //opcode is upper 6 bits
assign SignExtend = {{16{IR[15]}},IR[15:0]}; //sign extension of lower 16 bits of instruction
assign PCOffset = SignExtend << 2; //PC offset is shifted
// set the PC to 0 and start the control in state 0
initial begin PC = 0; state = 1; end


//The state machine--triggered on a rising clock
always @(posedge clock) begin
    Regs[0] = 0; //make R0 0 //shortcut way to make sure R0 is always 0
    case (state) //action depends on the state
      1: begin // first step: fetch the instruction, increment PC, go to next state
         IR <= Memory[PC>>2];
         PC <= PC + 4;
         state = 2; //next state
      end

      2: begin // second step: Instruction decode, register fetch, also compute branch address
         A <= Regs[IR[25:21]];
         B <= Regs[IR[20:16]];
         state = 3;
         ALUOut <= PC + PCOffset; // compute PC-relative branch target
      end


      3: begin // third step: Load-store execution, ALU execution, Branch completion
          state = 4; // default next state
          if ((opcode==LW) |(opcode==SW)) ALUOut <= A + SignExtend; //compute effective address
          else if (opcode==6'b0) case (IR[5:0]) //case for the various R-type instructions
            32: ALUOut = A + B; //add operation
            default: ALUOut = A; //other R-type operations: subtract, SLT, etc.
          endcase
```

FIGURE e4.14.5 A behavioral specification of the multicycle MIPS design. This has the same cycle behavior as the multicycle design, but is purely for simulation and specification. It cannot be used for synthesis.