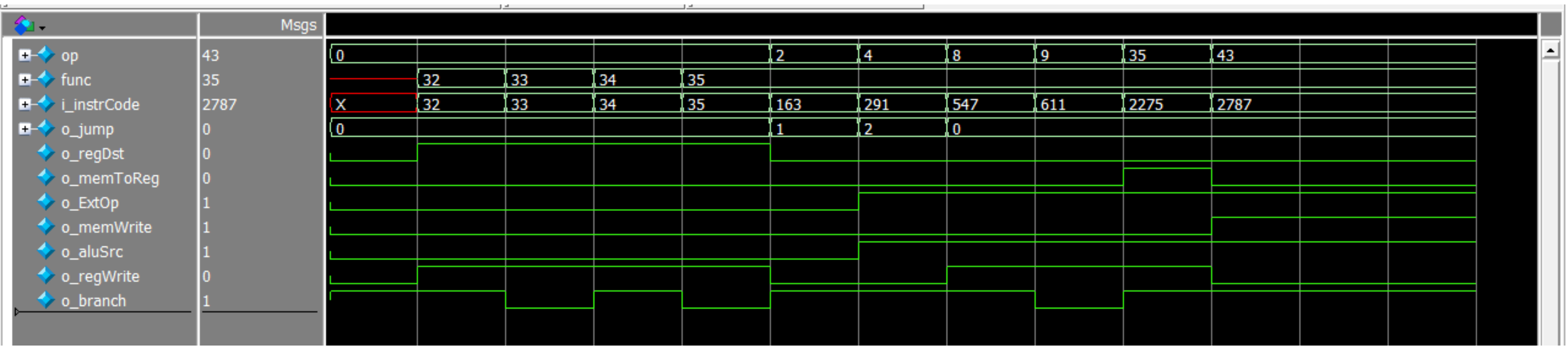
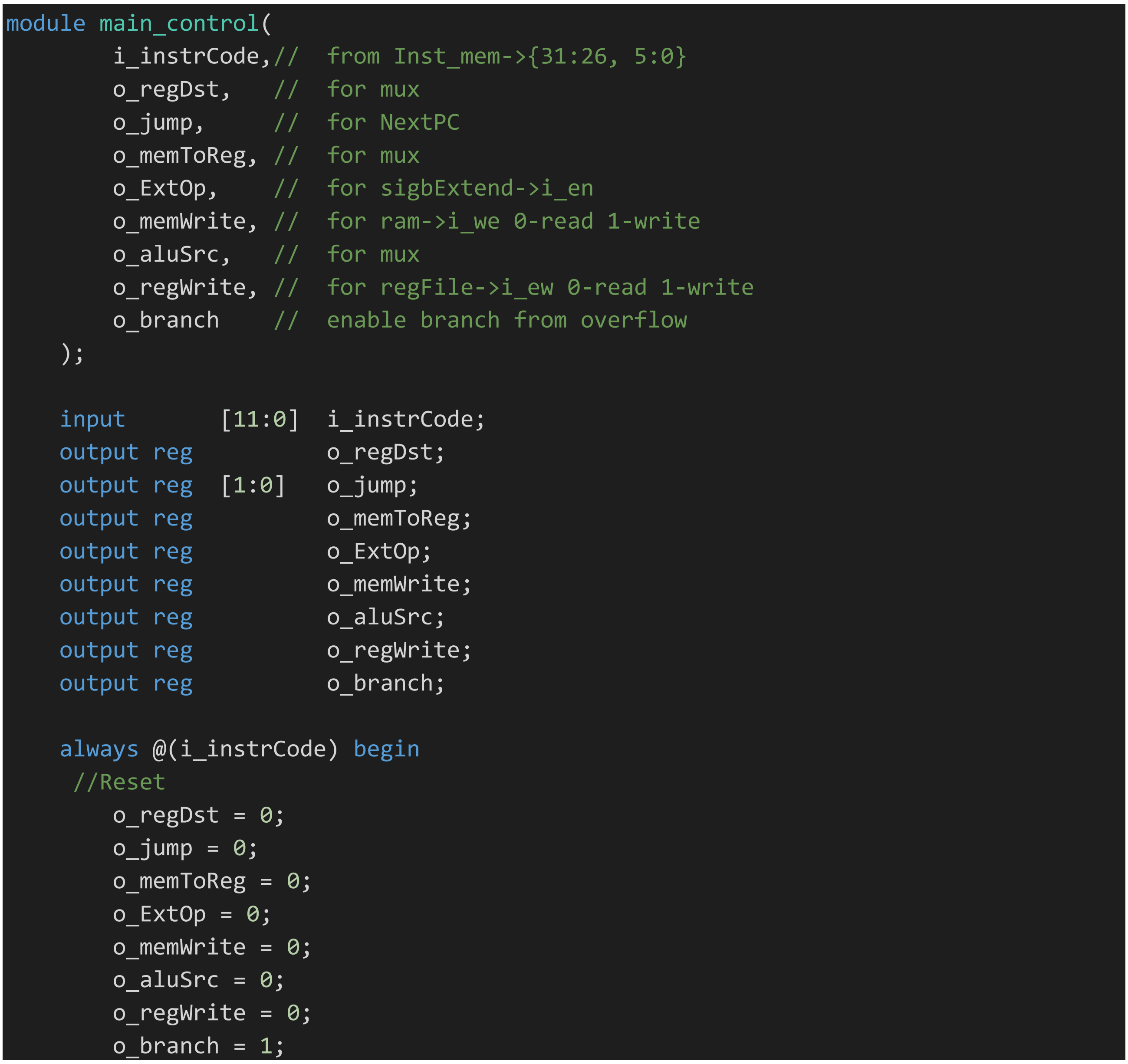
Створив модуль main_control. Протестував його роботу як окремого модуля. В коді вказав куди які провідники підключаються. Залишилось не зрозумілим куди поступає сигнал o_branch

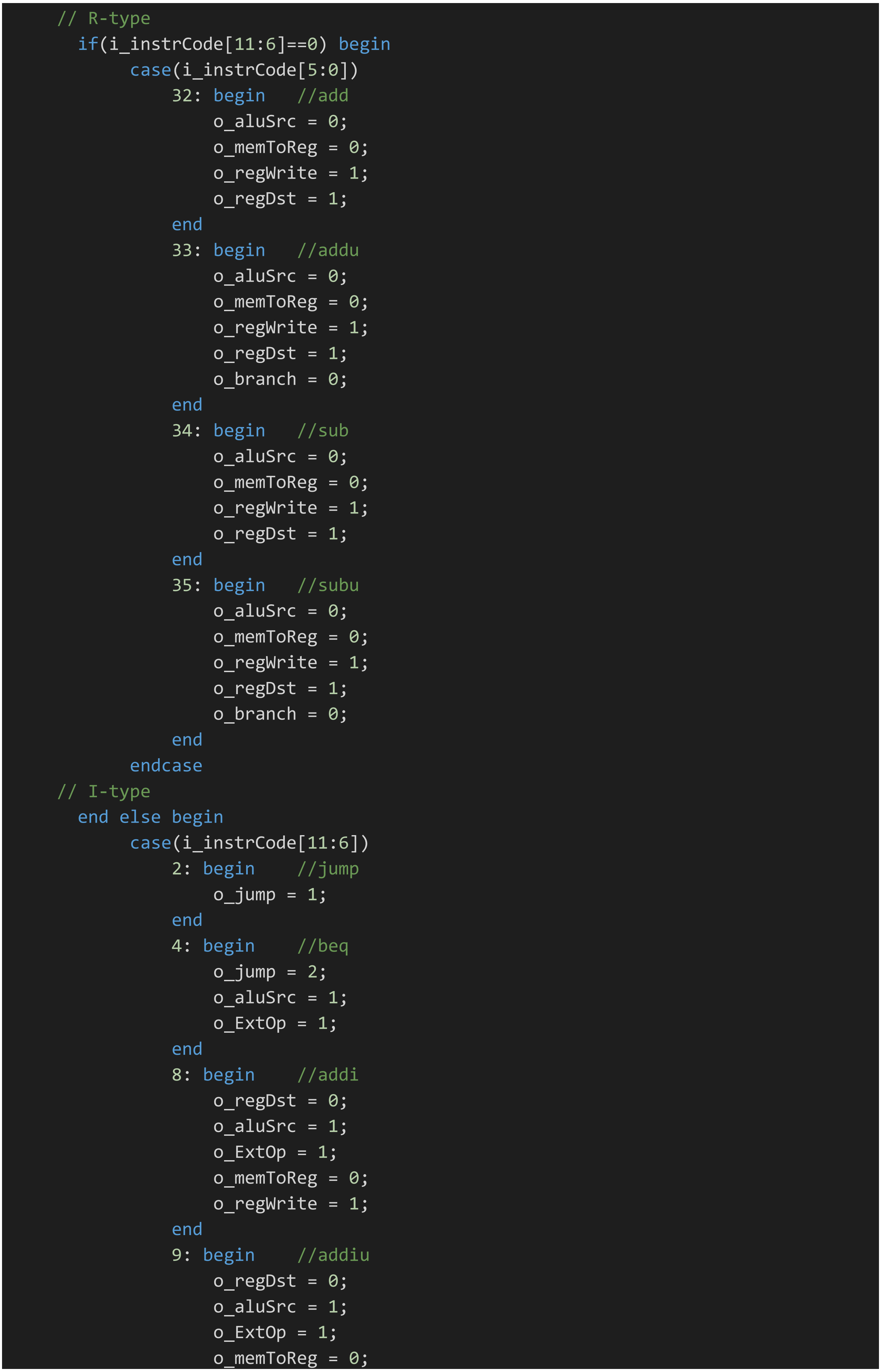




```
module main_control(
        i_instrCode,//  from Inst_mem->{31:26, 5:0}
        o_regDst,   //  for mux
        o_jump,     //  for NextPC
        o_memToReg, //  for mux
        o_ExtOp,    //  for sigbExtend->i_en
        o_memWrite, //  for ram->i_we 0-read 1-write
        o_aluSrc,   //  for mux
        o_regWrite, //  for regFile->i_ew 0-read 1-write
        o_branch    //  enable branch from overflow
    );

    input       [11:0]  i_instrCode;
    output reg          o_regDst;
    output reg  [1:0]   o_jump;
    output reg          o_memToReg;
    output reg          o_ExtOp;
    output reg          o_memWrite;
    output reg          o_aluSrc;
    output reg          o_regWrite;
    output reg          o_branch;

    always @(i_instrCode) begin
     //Reset
        o_regDst = 0;
        o_jump = 0;
        o_memToReg = 0;
        o_ExtOp = 0;
        o_memWrite = 0;
        o_aluSrc = 0;
        o_regWrite = 0;
        o_branch = 1;
```



```
    // R-type
      if(i_instrCode[11:6]==0) begin
           case(i_instrCode[5:0])
               32: begin   //add
                   o_aluSrc = 0;
                   o_memToReg = 0;
                   o_regWrite = 1;
                   o_regDst = 1;
               end
               33: begin   //addu
                   o_aluSrc = 0;
                   o_memToReg = 0;
                   o_regWrite = 1;
                   o_regDst = 1;
                   o_branch = 0;
               end
               34: begin   //sub
                   o_aluSrc = 0;
                   o_memToReg = 0;
                   o_regWrite = 1;
                   o_regDst = 1;
               end
               35: begin   //subu
                   o_aluSrc = 0;
                   o_memToReg = 0;
                   o_regWrite = 1;
                   o_regDst = 1;
                   o_branch = 0;
               end
           endcase
    // I-type
      end else begin
           case(i_instrCode[11:6])
               2: begin    //jump
                   o_jump = 1;
               end
               4: begin    //beq
                   o_jump = 2;
                   o_aluSrc = 1;
                   o_ExtOp = 1;
               end
               8: begin    //addi
                   o_regDst = 0;
                   o_aluSrc = 1;
                   o_ExtOp = 1;
                   o_memToReg = 0;
                   o_regWrite = 1;
               end
               9: begin    //addiu
                   o_regDst = 0;
                   o_aluSrc = 1;
                   o_ExtOp = 1;
                   o_memToReg = 0;
```

```
          o_regWrite = 1;
          o_branch = 0;
        end
        35: begin   //lw
          o_regDst = 0;
          o_aluSrc = 1;
          o_ExtOp = 1;
          o_memToReg = 1;
          o_regWrite = 1;
          o_memWrite = 0;
        end
        43: begin   //sw
          o_regDst = 0;
          o_aluSrc = 1;
          o_ExtOp = 1;
          o_regWrite = 0;
          o_memWrite = 1;
        end
      endcase
    end
  end
endmodule
```