MINICPU 本体

ソースコード

```
`include "defsm.v"

module minicpu(clk, reset, run, in, cs, pcout, irout, qtop, abus, dbus, out);

  input clk,reset,run;
  input [15:0] in;
  output [1:0] cs;
  output [15:0] irout, qtop, dbus, out;
  output [11:0] pcout, abus;
  wire [15:0] qnext, ramout, aluout;
  reg [11:0] abus;
  reg halt, jump, pcinc, push, pop, thru, qthru, dbus2qtop, dbus2ram, dbus2obuf, ir2dbus, qtop2dbus,
alu2dbus, ram2dbus, in2dbus;
  reg pop2, ir2abus, qtop2abus, qnext2abus;
  reg [11:0] pcout, pcnext;
  reg [15:0] out;

  statef statef0(.clk(clk),.reset(reset),.run(run),.halt(halt),.cs(cs));
      stackm     stackm0 (.clk(clk),.reset(reset),.load(dbus2 qtop),.push(push),.pop(pop),.pop2 (pop2
),.thru(qthru),.d(dbus),.dthru(ramout),.qtop(qtop),.qnext(qnext));
  alu alu0(.a(qtop),.b(qnext),.f(irout[4:0]),.s(aluout));
   dpram  #(16,10,1024)  dpram0(.clk(clk),.load1(dbus2ram),.addr1(abus),.addr2(pcnext),.d1(dbus),.q1
(ramout),.q2(irout));

  always @ (pcinc or jump or cs or pcout or irout)
    if(cs == `IDLEB) pcnext = pcout;
    else if(pcinc)  pcnext = pcout + 1;
    else if(jump)  pcnext = irout[11:0];
    else pcnext = 12'hxxx;

  always @ (posedge clk or negedge reset)
    if(!reset) pcout <= 0;
    else if(pcinc || jump) pcout <= pcnext;

  always @ (posedge clk or negedge reset)
    if(!reset) out <= 0;
    else if(dbus2obuf) out <= dbus;

  always @ (posedge clk or negedge reset)
    if(!reset) qthru <= 0;
    else qthru <= thru;

  always @ (ir2abus or qtop2abus or qnext2abus or irout or qtop or qnext)
  if(ir2abus) abus = irout[11:0];
  else if(qtop2abus) abus = qtop[11:0];
  else if(qnext2abus) abus = qnext[11:0];
  else abus = 12'hxxx;

  assign dbus = ir2dbus ? {{4{irout[11]}},irout[11:0]} : 16'hzzzz;
  assign dbus = qtop2dbus ? qtop : 16'hzzzz;
  assign dbus = alu2dbus ? aluout : 16'hzzzz;
  assign dbus = ram2dbus ? ramout : 16'hzzzz;
  assign dbus = in2dbus ? in : 16'hzzzz;

  always @(cs or irout or qtop)
    begin
      halt = 0; pcinc = 0; jump = 0; push = 0; pop = 0; pop2 = 0; thru = 0; dbus2qtop = 0; dbus2ram =
0; dbus2obuf = 0; ir2dbus = 0; qtop2dbus = 0; alu2dbus = 0; ram2dbus = 0; in2dbus = 0; ir2abus = 0;
qtop2abus = 0; qnext2abus = 0;
   if(cs == `EXEC)
     case(irout[15:12])
        `PUSHI:
          begin
            ir2dbus = 1; dbus2qtop = 1; push = 1; pcinc = 1;
          end
       `PUSH:
          begin
            ir2abus = 1; push = 1; thru = 1; pcinc = 1;
          end
       `POP:
          begin
            ir2abus = 1; qtop2dbus = 1; dbus2ram = 1; pop = 1; pcinc = 1;
          end
       `JMP: jump = 1;
```

```
      `JZ:
           begin
             if(qtop == 0) jump = 1;
             else pcinc = 1;
             pop = 1;
           end
      `JNZ:
          begin
            if(qtop != 0) jump = 1;
             else pcinc = 1;
             pop = 1;
          end
      `LD:
         begin
           qtop2abus = 1; thru = 1; pcinc = 1;
         end
      `ST:
         begin
           qnext2abus = 1; qtop2dbus = 1; dbus2ram = 1; pop2 = 1; pcinc = 1;
         end
     `IN:
         begin
          in2dbus = 1; dbus2qtop = 1; push = 1; pcinc = 1;
         end
      `OUT:
         begin
           qtop2dbus = 1; dbus2obuf = 1; pop = 1; pcinc = 1;
         end
      `OP:
           begin
             alu2dbus = 1; dbus2qtop = 1; pcinc = 1;
             if(irout[4] == 0) pop = 1;
           end
       default:
         begin
           pcinc = 1; halt = 1;
         end
      endcase
  end

endmodule
```