```
//clock signal
divider DIVIDER(.clkin(clkin), .clkout(clkout), .rst(reset), .clkcon(clkcon));
assign clk=(clkcon==3'b111)?hand:clkout;
clock CLOCK(.clk(clk), .clk_r(clk1), .reset(reset), .fetch(fetch));
//controller
cu_contrl  CUCON(.clk(clk1), .cu_ena(cu_ena), .fetch(fetch), .rst(reset));
//control unit
cu
CU(.clk(clk1), .cu_ena(cu_ena), .flag_in(flag_in), .op(ir_out[15:11]), .pc_inc(pc_inc),
.pc_ena(pc_ena), .ir_ena(ir_ena), .reg_read1(reg_read1), .reg_read2(reg_read2),
.reg_write1(reg_write1), .reg_write2(reg_write2), .alu_data_sel(alu_data_sel), .flag_se
t(flag_set),
.wr_m(wr_m), .rd_m(rd_m), .sp_pop(sp_pop), .sp_push(sp_push), .mar_sel(mar_sel), .mar_e
na(mar_ena),
.mdr_sel(mdr_sel), .mdr_ena(mdr_ena), .alu_ena(alu_ena), .hlt(hlt), .io(io), .state(sta
te), .run(run)
);
//instruction register
instreg
INSREG(.ir_out(ir_out), .data(mem_out), .ir_ena(ir_ena), .clk(clk1), .rst(reset));
//program counter
pc PC(.pc_value(pc_addr), .offset(ir_out[10:0]), .pc_inc(pc_inc), .clk(clk1), .sw(sw),
.pc_ena(pc_ena), .rst(reset));
//stack pointer
sp SP(.clk(clk1), .rst(reset), .sp_pop(sp_pop), .sp_push(sp_push), .sp_value(sp_addr));
//flags
flags
FLAGS(.clk(clk1), .rst(reset), .flag_set(flag_set), .flag_in(flag_out), .flag_value(fla
g_in));
//register array
regarray
REGARRAY(.clk(clk1), .rst(reset), .reg_read1(reg_read1), .reg_read2(reg_read2),
.addr1(ir_out[10:8]), .addr2(ir_out[7:5]), .reg_write1(reg_write1), .reg_write2(reg_wri
te2),
.data_in1(alu_out), .data_in2(hi), .data_in3(mem_out), .reg_out1(a), .reg_out2(b), .por
t(port)
);
//memory address register
mar MAR(.clk(clk1), .rst(reset), .mar_ena(mar_ena), .mar_sel(mar_sel), .ir_addr1(a),
.ir_addr2(b), .pc_addr(pc_addr), .sp_addr(sp_addr), .mar_addr(addr)
);
//memory data register
mdr
MDR(.clk(clk1), .rst(reset), .mdr_ena(mdr_ena), .mdr_sel(mdr_sel), .reg_in1(a), .reg_in
2(b),
.reg_out(outdata), .mem_in(indata), .mem_out(mem_out)
);
//arithmetic logic unit
alu ALU(.data_a(data_a), .data_b(data_b), .alu_ena(alu_ena), .alu_opr(ir_out[15:11]),
.clk(clk1), .flag_in(flag_in), .alu_out(alu_out), .flag_out(flag_out), .hi(hi));
//alu control unit
alu_in_contrl ALUCON(.clk(clk1), .in_a(a), .in_b(b), .data_a(data_a), .data_b(data_b),
.alu_sel(alu_data_sel), .imm(ir_out[7:0]));
```