

# TINYCPU 本体

## 入出力ポート

|    | ポート名  | ビット数 |                             |
|----|-------|------|-----------------------------|
| 入力 | clk   | 1    | グローバルクロック                   |
| 入力 | reset | 1    | グローバルリセット                   |
| 入力 | run   | 1    | 動作開始                        |
| 入力 | in    | 16   | 機械語命令 IN により<br>データを読み込むポート |
| 出力 | cs    | 3    | 現在の状態 *                     |
| 出力 | pcout | 12   | プログラムカウンタの<br>値 *           |
| 出力 | irout | 16   | 命令レジスタの値 *                  |
| 出力 | qtop  | 16   | スタックトップの値 *                 |
| 出力 | abus  | 12   | アドレスバスの値 *                  |
| 出力 | dbus  | 16   | データバスの値 *                   |
| 出力 | out   | 16   | 出力バッファの値                    |

上の表の \* は外部で CPU の動作を観測するためのもので、なくてもよい。

## 命令セット

### TINYCPU の命令セット

## ソースコード

```
`include "defs.v"

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

    input clk, reset, run;
    input [15:0] in;
    output [2:0] 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, cont, pcinc, push, pop, abus2pc, dbus2ir, dbus2qtop, dbus2ram,
        dbus2obuf, pc2abus, ir2abus, ir2dbus, qtop2dbus, alu2dbus, ram2dbus,
        in2dbus;

    counter #(12) pc0(.clk(clk), .reset(reset), .load(abus2pc), .inc(pcinc),
        .d(abus), .q(pcout));
    counter #(16) ir0(.clk(clk), .reset(reset), .load(dbus2ir), .inc(0),
        .d(dbus), .q(irout));
    state state0(.clk(clk), .reset(reset), .run(run), .cont(cont),
        .halt(halt), .cs(cs));
    stack stack0(.clk(clk), .reset(reset), .load(dbus2qtop), .push(push),
        .pop(pop), .d(dbus), .qtop(qtop), .qnext(qnext));
    alu alu0(.a(qtop), .b(qnext), .f(irout[4:0]), .s(aluout));
    ram #(16,12,4096) ram0(.clk(clk), .load(dbus2ram), .addr(abus[11:0]),
        .d(dbus), .q(ramout));
    counter #(16) obuf0(.clk(clk), .reset(reset), .load(dbus2obuf), .inc(0),
```

```

        .d(dbus), .q(out));

always @(pc2abus or ir2abus or pcout or irout)
    if(pc2abus) abus <= pcout;
    else if(ir2abus) abus <= irout[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; push = 0; pop = 0; cont = 0; abus2pc = 0;
    dbus2ir = 0; dbus2qtop = 0; dbus2ram = 0; dbus2obuf = 0; pc2abus = 0;
    ir2abus = 0; ir2dbus = 0; qtop2dbus = 0; alu2dbus = 0; ram2dbus = 0;
    in2dbus = 0;
    if(cs == `FETCHA)
        begin
            pcinc = 1; pc2abus = 1;
        end
    else if(cs == `FETCHB)
        begin
            ram2dbus = 1; dbus2ir = 1;
        end
    else if(cs == `EXECA)
        case(irout[15:12])
            `PUSH:
                begin
                    ir2dbus = 1; dbus2qtop = 1; push = 1;
                end
            `PUSH:
                begin
                    ir2abus = 1; cont = 1;
                end
            `POP:
                begin
                    ir2abus = 1; qtop2dbus = 1; dbus2ram = 1; pop = 1;
                end
            `JMP:
                begin
                    ir2abus = 1; abus2pc = 1;
                end
            `JZ:
                begin
                    if(qtop == 0)
                        begin
                            ir2abus = 1; abus2pc = 1;
                        end
                    pop = 1;
                end
            `JNZ:
                begin
                    if(qtop != 0)
                        begin
                            ir2abus = 1; abus2pc = 1;
                        end
                    pop = 1;
                end
            `IN:
                begin
                    in2dbus = 1; dbus2qtop = 1; push = 1;
                end
            `OUT:
                begin
                    qtop2dbus = 1; dbus2obuf = 1; pop = 1;
                end
            `OP:
                begin
                    alu2dbus = 1; dbus2qtop = 1;
                    if(irout[4] == 0) pop = 1;
                end
            default:
                halt = 1;
        endcase
    else if(cs == `EXECB)
        if(irout[15:12] == `PUSH)
            begin
                ram2dbus = 1; dbus2qtop = 1; push = 1;
            end
        end
end

```

```
end  
endmodule
```

#### 注意

ISE Foundation( および ISE WebPack) では、18kbit( 1つのブロック RAM の容量) より大きいメモリの論理合成に膨大な時間を必要とする。これを避けるために、ram のインスタンス化時のパラメータを #(16,10,1024) としてメモリのサイズを 16kbit とすれば、大幅に論理合成時間を短縮することができる。