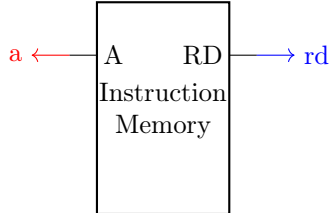


RISC-V Processor Circui*Tik*Z Library

March 12, 2025

1 Components

1.1 Instruction Memory

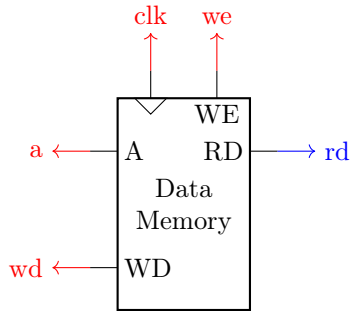


```

1 \begin{circuitikz}
2 \node[instrmem, align=center] (comp) {Instruction
3 \Memory};
4 \draw[->, red] (comp.a) -- ++(-.5, 0) node[left] {
5 a};
6 \draw[->, blue] (comp.rd) -- ++(.5, 0) node[right]
7 {rd};
8 \end{circuitikz}

```

1.2 Data Memory

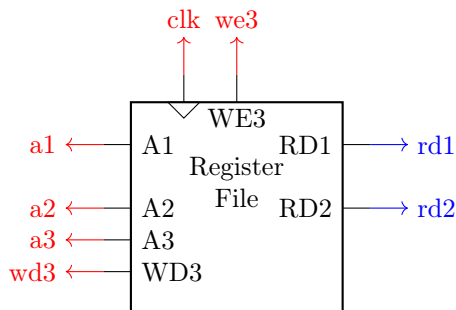


```

1 \begin{circuitikz}[]
2 \node[datamem, align=center] (comp) {Data\Memory
3 };
4 \draw[->, red] (comp.a) -- ++(-.5, 0) node[left]
5 {a};
6 \draw[->, red] (comp.wd) -- ++(-.5, 0) node[left]
7 {wd};
8 \draw[->, red] (comp.clk) -- ++(0, .5) node[above]
9 {clk};
10 \draw[->, red] (comp.we) -- ++(0, .5) node[above]
11 {we};
12 \draw[->, blue] (comp.rd) -- ++(.5, 0) node[right]
13 {rd};
14 \end{circuitikz}

```

1.3 Register File

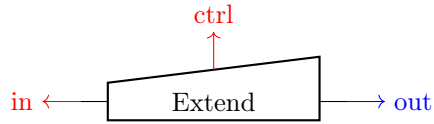


```

1 \begin{circuitikz}[]
2 \node[regfile, align=center] (comp) {
3 Register\
4 File};
5 \draw[->, red] (comp.a1) -- ++(-.5, 0)
6 node[left] {a1};
7 \draw[->, red] (comp.a2) -- ++(-.5, 0)
8 node[left] {a2};
9 \draw[->, red] (comp.a3) -- ++(-.5, 0)
10 node[left] {a3};
11 \draw[->, red] (comp.wd3) -- ++(-.5, 0)
12 node[left] {wd3};
13 \draw[->, red] (comp.clk) -- ++(0, .5)
14 node[above] {clk};
15 \draw[->, red] (comp.we3) -- ++(0, .5)
16 node[above] {we3};
17 \draw[->, blue] (comp.rd1) -- ++(.5, 0)
18 node[right] {rd1};
19 \draw[->, blue] (comp.rd2) -- ++(.5, 0)
20 node[right] {rd2};
21 \end{circuitikz}

```

1.4 Extend Unit

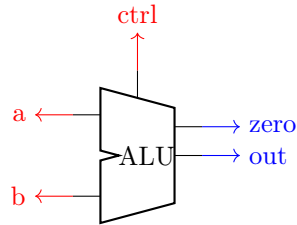


```

1 \begin{circuitikz}[]
2 \node[extend, align=center] (comp) {Extend
3 \draw[->, red] (comp.in) -- ++(-.5, 0)
  node[left] {in};
4 \draw[->, red] (comp.ctrl) -- ++(0, .5)
  node[above] {ctrl};
5 \draw[->, blue] (comp.out) -- ++(.5, 0)
  node[right] {out};
6 \end{circuitikz}

```

1.5 Arithmetic Logic Unit

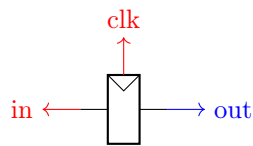


```

1 \begin{circuitikz}[]
2 \node[alu, align=center] (comp) {ALU};
3 \draw[->, red] (comp.a) -- ++(-.5, 0) node[left] {a};
4 \draw[->, red] (comp.b) -- ++(-.5, 0) node[left] {b};
5 \draw[->, red] (comp.ctrl) -- ++(0, .5) node[above] {
  ctrl};
6 \draw[->, blue] (comp.out) -- ++(.5, 0) node[right] {
  out};
7 \draw[->, blue] (comp.zero) -- ++(.5, 0) node[right] {
  zero};
8 \end{circuitikz}

```

1.6 Register

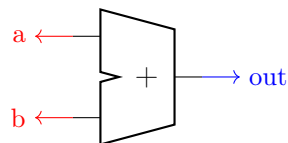


```

1 \begin{circuitikz}[]
2 \node[reg, align=center] (comp) {};
3 \draw[->, red] (comp.in) -- ++(-.5, 0) node[left] {in};
4 \draw[->, red] (comp.clk) -- ++(0, .5) node[above] {clk
5 \draw[->, red] (comp.en) -- ++(0, -.5) node[below] {en
  };
6 \draw[->, blue] (comp.out) -- ++(.5, 0) node[right] {out
  };
7 \end{circuitikz}

```

1.7 Adder

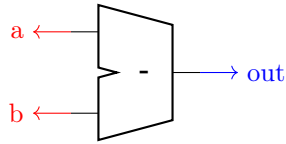


```

1 \begin{circuitikz}[]
2 \node[adder, align=center] (comp) {};
3 \draw[->, red] (comp.a) -- ++(-.5, 0) node[left] {a};
4 \draw[->, red] (comp.b) -- ++(-.5, 0) node[left] {b};
5 \draw[->, blue] (comp.out) -- ++(.5, 0) node[right] {
  out};
6 \end{circuitikz}

```

1.8 Subtractor

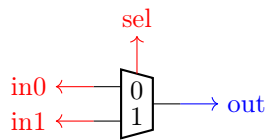


```

1 \begin{circuitikz}[]
2 \node[sub, align=center] (comp) {};
3 \draw[->, red] (comp.a) -- ++(-.5, 0) node[left] {a};
4 \draw[->, red] (comp.b) -- ++(-.5, 0) node[left] {b};
5 \draw[->, blue] (comp.out) -- ++(.5, 0) node[right] {
6 out};
7 \end{circuitikz}

```

1.9 Multiplexer

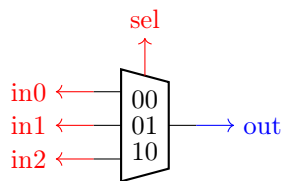


```

1 \begin{circuitikz}[]
2 \node[mux, align=center] (comp) {};
3 \draw[->, red] (comp.in0) -- ++(-.5, 0) node[left] {in
4 0};
5 \draw[->, red] (comp.in1) -- ++(-.5, 0) node[left] {in
6 1};
7 \draw[->, red] (comp.sel) -- ++(0, .5) node[above] {sel
8 };
9 \draw[->, blue] (comp.out) -- ++(.5, 0) node[right] {
10 out};
11 \end{circuitikz}

```

1.10 Multiplexer with 3 inputs

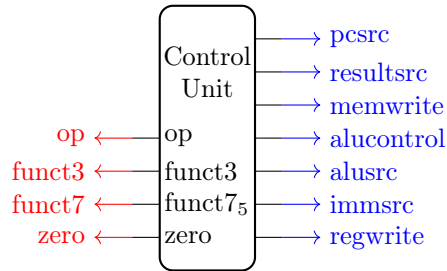


```

1 \begin{circuitikz}[]
2 \node[3mux, align=center] (comp) {};
3 \draw[->, red] (comp.in0) -- ++(-.5, 0) node[left] {in
4 0};
5 \draw[->, red] (comp.in1) -- ++(-.5, 0) node[left] {in
6 1};
7 \draw[->, red] (comp.in2) -- ++(-.5, 0) node[left] {in
8 2};
9 \draw[->, red] (comp.sel) -- ++(0, .5) node[above] {
10 sel};
11 \draw[->, blue] (comp.out) -- ++(.5, 0) node[right] {
12 out};
13 \end{circuitikz}

```

1.11 Single-Cycle Control Unit

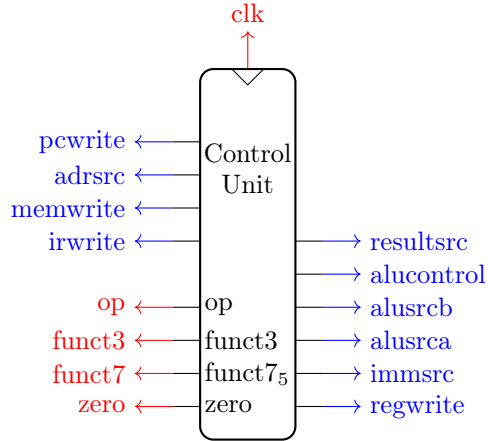


```

1 \begin{circuitikz}[]
2   \node[ctrlunitsc, align=center] (comp) {
3     Control\\Unit};
4   \draw[->, red] (comp.op) -- ++(-.5, 0)
5     node[left] {op};
6   \draw[->, red] (comp.funct3) -- ++(-.5,
7     0) node[left] {funct3};
8   \draw[->, red] (comp.funct7) -- ++(-.5,
9     0) node[left] {funct7};
10  \draw[->, red] (comp.zero) -- ++(-.5, 0)
11    node[left] {zero};
12
13  \draw[->, blue] (comp.pcsrc) -- ++(.5, 0)
14    node[right] {pcsrc};
15  \draw[->, blue] (comp.resultsrc) --
16    ++(.5, 0) node[right] {resultsrc};
17  \draw[->, blue] (comp.memwrite) -- ++(.5,
18    0) node[right] {memwrite};
19  \draw[->, blue] (comp.alucontrol) --
20    ++(.5, 0) node[right] {alucontrol};
21  \draw[->, blue] (comp.alusrc) -- ++(.5,
22    0) node[right] {alusrc};
23  \draw[->, blue] (comp.immsrc) -- ++(.5,
24    0) node[right] {immsrc};
25  \draw[->, blue] (comp.regwrite) -- ++(.5,
26    0) node[right] {regwrite};
27 \end{circuitikz}

```

1.12 Multi-Cycle Control Unit



```

1 \begin{circuitikz}[]
2   \node[ctrlunitmc, align=center] (comp)
3     {Control\\Unit};
4   \draw[->, red] (comp.op) -- ++(-.5, 0)
5     node[left] {op};
6   \draw[->, red] (comp.funct3) --
7     ++(-.5, 0) node[left] {funct3};
8   \draw[->, red] (comp.funct7) --
9     ++(-.5, 0) node[left] {funct7};
10  \draw[->, red] (comp.zero) -- ++(-.5,
11    0) node[left] {zero};
12  \draw[->, red] (comp.clk) -- ++(0,.5)
13    node[above] {clk};
14
15  \draw[->, blue] (comp.resultsrc) --
16    ++(.5, 0) node[right] {resultsrc};
17  \draw[->, blue] (comp.memwrite) --
18    ++(-.5, 0) node[left] {memwrite};
19  \draw[->, blue] (comp.alucontrol) --
20    ++(.5, 0) node[right] {alucontrol
21    };
22  \draw[->, blue] (comp.alusrcb) --
23    ++(.5, 0) node[right] {alusrcb};
24  \draw[->, blue] (comp.alusrca) --
25    ++(.5, 0) node[right] {alusrca};
26  \draw[->, blue] (comp.immsrc) --
27    ++(.5, 0) node[right] {immsrc};
28  \draw[->, blue] (comp.regwrite) --
29    ++(.5, 0) node[right] {regwrite};
30  \draw[->, blue] (comp.irwrite) --
31    ++(-.5, 0) node[left] {irwrite};
32  \draw[->, blue] (comp.adrsrc) --
33    ++(-.5, 0) node[left] {adrsrc};
34  \draw[->, blue] (comp.pcwrite) --
35    ++(-.5, 0) node[left] {pcwrite};
36 \end{circuitikz}

```

2 Single-Cycle RISC-V Processor

