

A RISC-V Processor Components CircuiTikZ Library

Niklas Ladurner

October 13, 2025

Contents

| | | |
|----------|---|-----------|
| 1 | Introduction | 3 |
| 1.1 | Motivation | 3 |
| 1.2 | Usage | 3 |
| 2 | Component List | 4 |
| 2.1 | Memory Components | 4 |
| 2.2 | Arithmetic Components | 4 |
| 2.3 | Multiplexers | 5 |
| 2.4 | Control Units | 5 |
| 2.5 | Miscellaneous Components | 6 |
| 3 | Keys | 8 |
| 3.1 | CircuiTikZ keys | 8 |
| 3.2 | Special node keys | 8 |
| 4 | Examples | 10 |
| 4.1 | Single-Cycle RISC-V Processor | 10 |
| 4.2 | Single-Cycle RISC-V Processor (with Branch Logic) | 10 |
| 4.3 | Multi-Cycle RISC-V Processor | 11 |
| 4.4 | Pipelined RISC-V Processor | 12 |
| 4.5 | Pipelined RISC-V Processor with Hazard Unit | 12 |

1 Introduction

1.1 Motivation

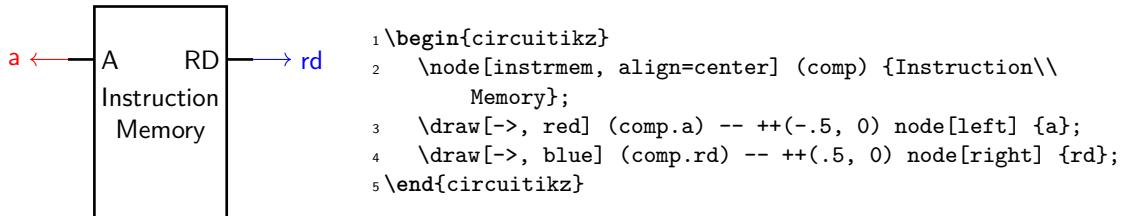
This CircuiTikZ library offers some components to efficiently draw RISC-V processors in L^AT_EX. The library was designed with the goal of resembling the RISC-V processor schematics as presented in ‘Digital Design and Computer Architecture: RISC-V Edition’ by Sarah L. Harris and David Harris.

1.2 Usage

To use the predefined components, you must include the library `riscvproc`. Your preamble should look like this:

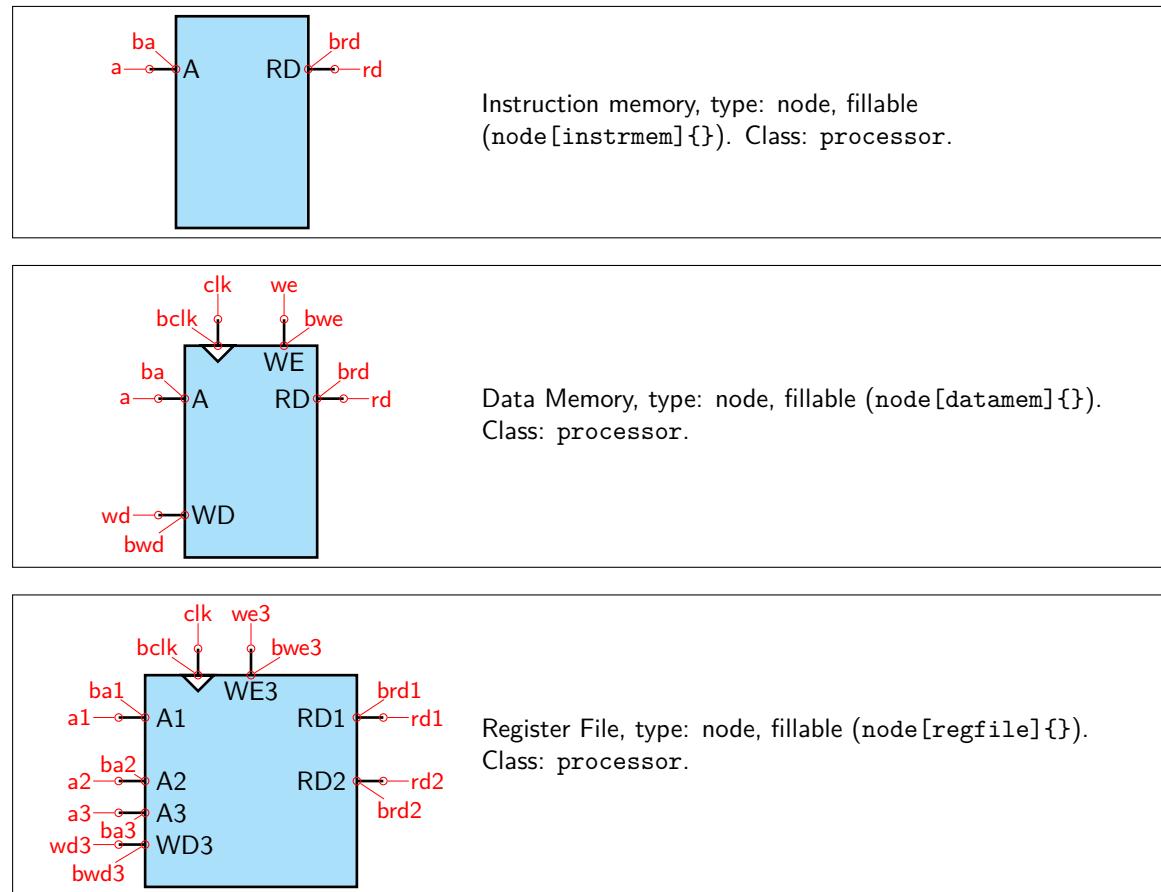
```
...
\usepackage{tikz}
\usepackage{circuitikz}
\usetikzlibrary{riscvproc}
...
```

Components are then available in `circuitikz` environments:

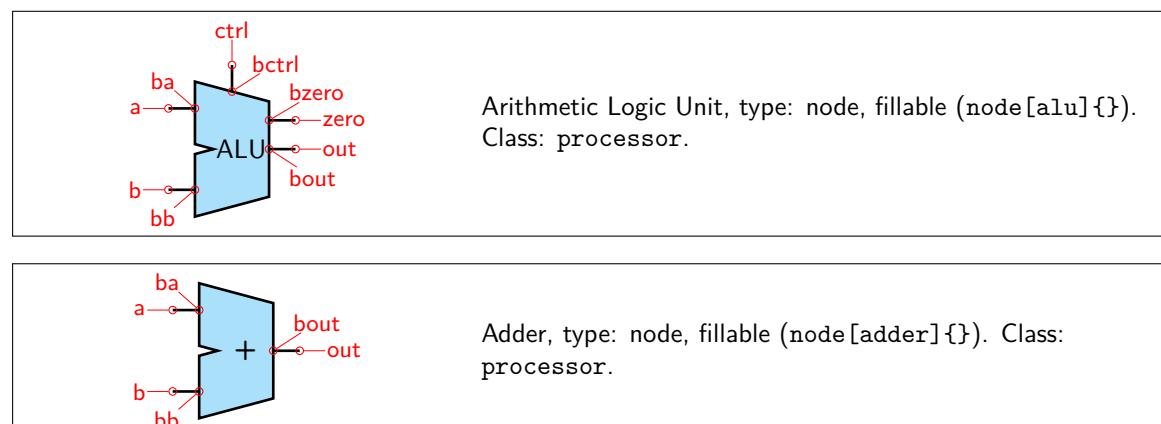


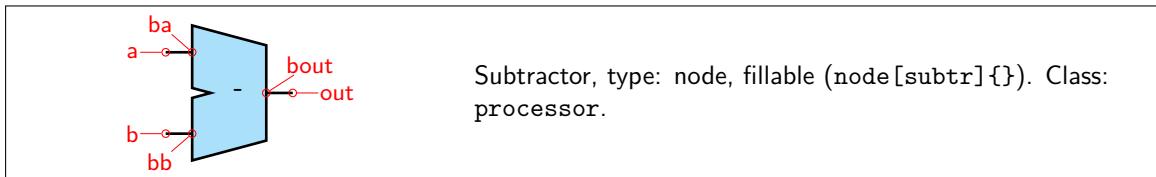
2 Component List

2.1 Memory Components

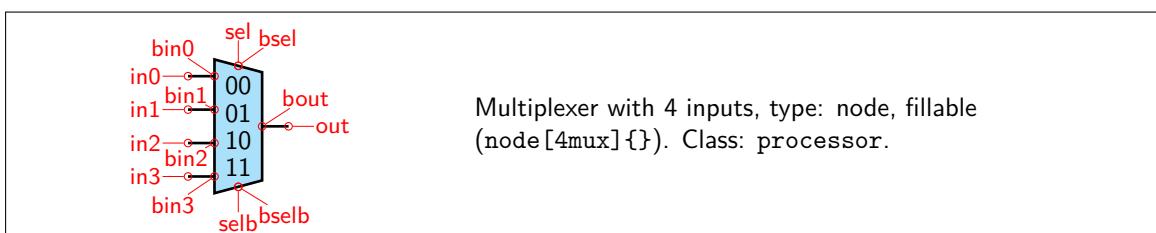
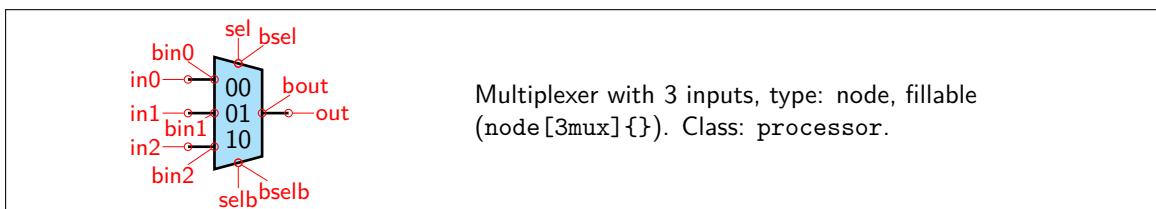
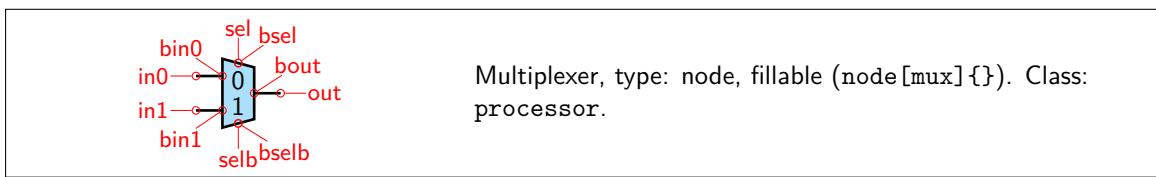


2.2 Arithmetic Components

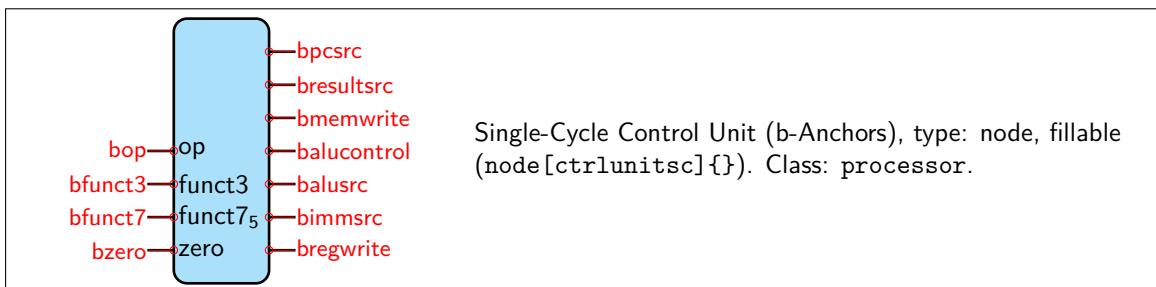
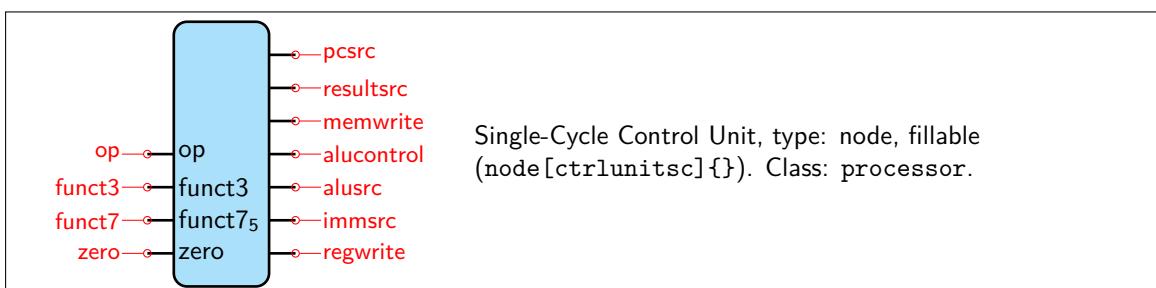


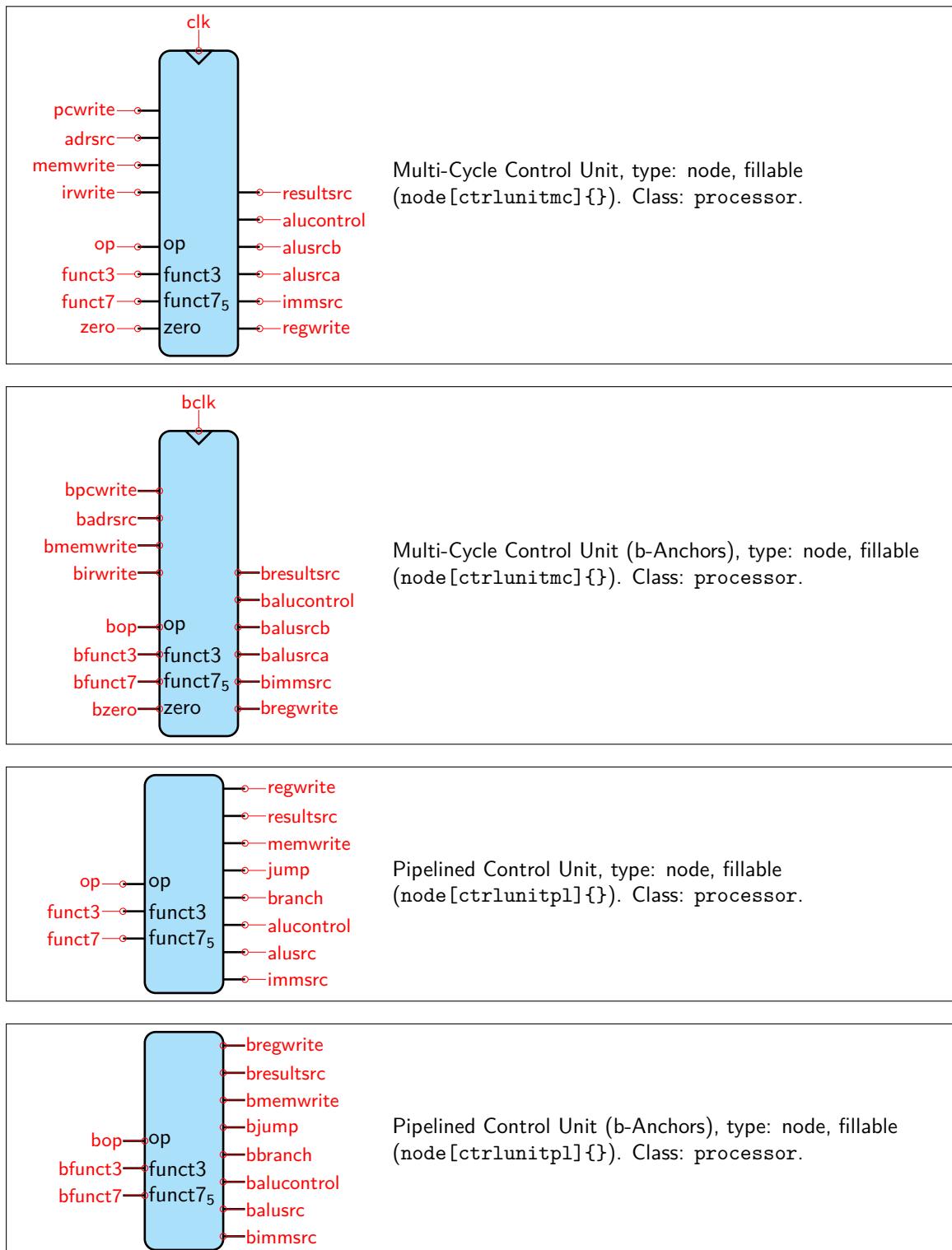


2.3 Multiplexers

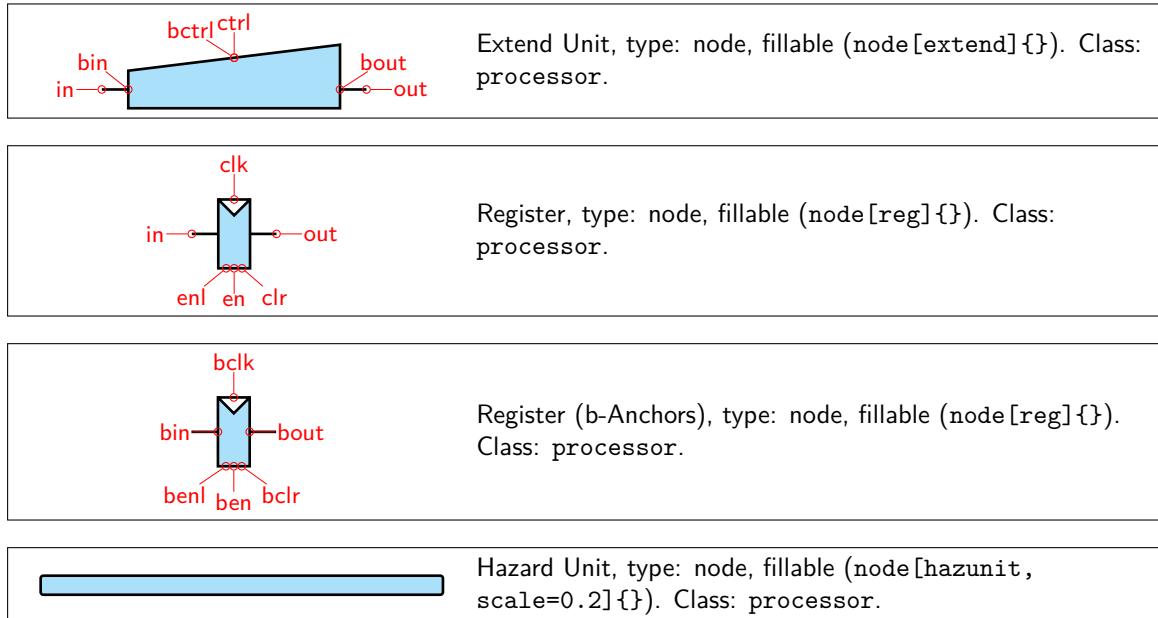


2.4 Control Units





2.5 Miscellaneous Components



3 Keys

3.1 CircuiTikZ keys

The desired CircuiTikZ key can be set via `\ctikzset{processor/<key>=value}`. E.g. if one wishes to set the line width of all components to 4, the line `\ctikzset{processor/thickness=4}` would have to be included in the specific circuitikz picture. A list of all CircuiTikZ keys can be found in Table 1. A list of component families can be found in Table 2.

| Key | Description | Default value |
|------------------|---|------------------------|
| scale | Sets scale for all processor components. | 1 |
| thickness | Sets line width for all processor components. | 1 |
| leadthickness | Sets line width for all leads. | 1 |
| font | Sets font family for all labels of processor components. | <code>\rmfamily</code> |
| memory/height | Sets height for all memory components. | 2 |
| memory/width | Sets width for all memory components except <code>regfile</code> . | 1.25 |
| control/heightsc | Sets height for <code>ctrlunitsc</code> . | 2.5 |
| control/heightmc | Sets height for <code>ctrlunitmc</code> . | 3.5 |
| control/width | Sets width for control components. | 0.9 |
| control/radius | Sets border radius for control components. | 5 |
| arith/height | Sets height for arithmetic components. | 0.9 |
| arith/width | Sets height for arithmetic components. | 0.7 |
| arith/slope | Sets slope for arithmetic components in degrees. | 15 |
| extend/height | Sets height for big side of extend components. | 0.6 |
| extend/width | Sets height for extend components. | 2 |
| extend/slope | Sets slope for extend components in degrees. | 7 |
| mux/slope | Sets slope for multiplexers in degrees. | 15 |
| misc/smallheight | Sets height for small components. | 0.65 |
| misc/smallwidth | Sets width for small components. Also affects the CLK input triangle. | 0.3 |
| misc/leadlen | Sets length for input and output leads. | 0.25 |
| hazard/height | Sets height for <code>hazunit</code> . | 0.9 |
| hazard/width | Sets width for <code>hazunit</code> . | 18 |
| hazard/radius | Sets border radius for <code>hazunit</code> . | 5 |

Table 1: List of CircuiTikZ keys

3.2 Special node keys

Some keys are also defined as Tikz keys and can therefore be directly passed to nodes like shown in Figure 1. A list of all these keys can be found in Table 3.

More keys might be added in future.

| Component family | Component list |
|-----------------------|----------------------------|
| memory components | instrmem, datamem, regfile |
| control components | ctrlunitsc, ctrlunitmc |
| arithmetic components | alu, add, subtr |
| extend components | extend |
| small components | mux, reg |

Table 2: List of component families

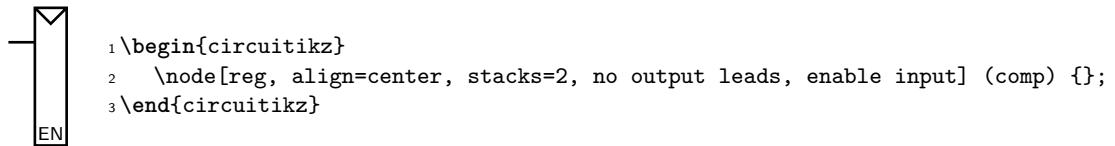


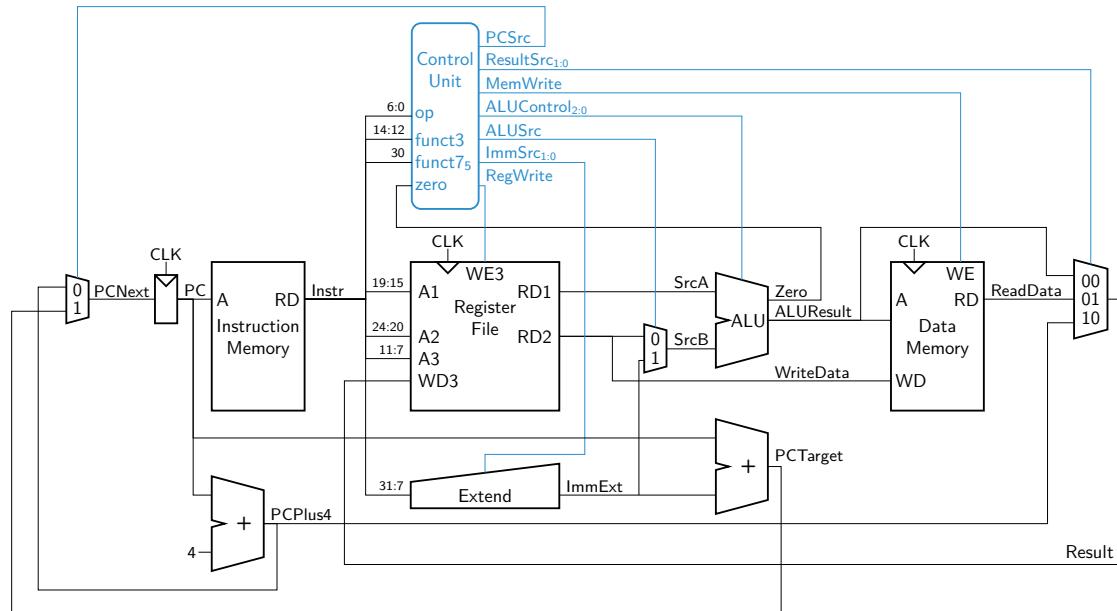
Figure 1: Passing options to a node

| Key | Description | applicable to |
|--------------|--|----------------------|
| input leads | Specifies whether to draw input leads. | all components |
| output leads | Specifies whether to draw output leads. | all components |
| leads | Specifies whether to draw leads at all. | all components |
| stacks | Sets height of a register in multiples of the default height, allows for stretched registers. | reg |
| enable input | Specifies whether to draw an enable input or not. This also gives two new anchors, en and ben. | reg |
| clear input | Specifies whether to draw a clear input or not. This also gives two new anchors, clr and bclr. For Usage of enable and clear inputs, use the enl and benl anchors. | reg |
| clock | Specifies whether to draw a clk input on a component that supports it. | all timed components |

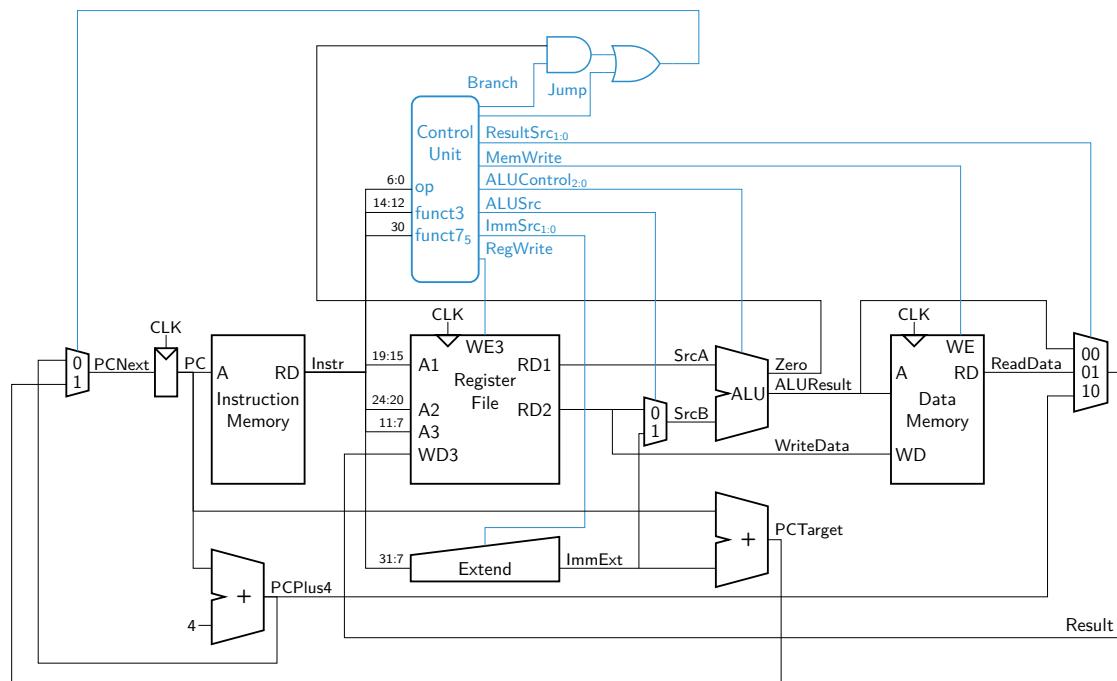
Table 3: List special node keys

4 Examples

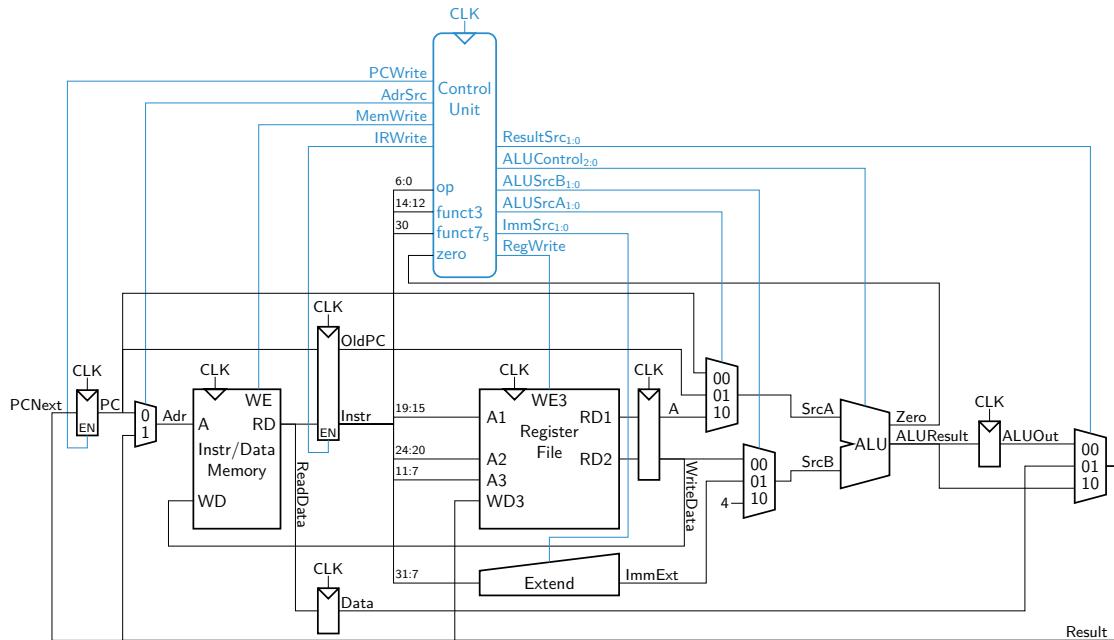
4.1 Single-Cycle RISC-V Processor



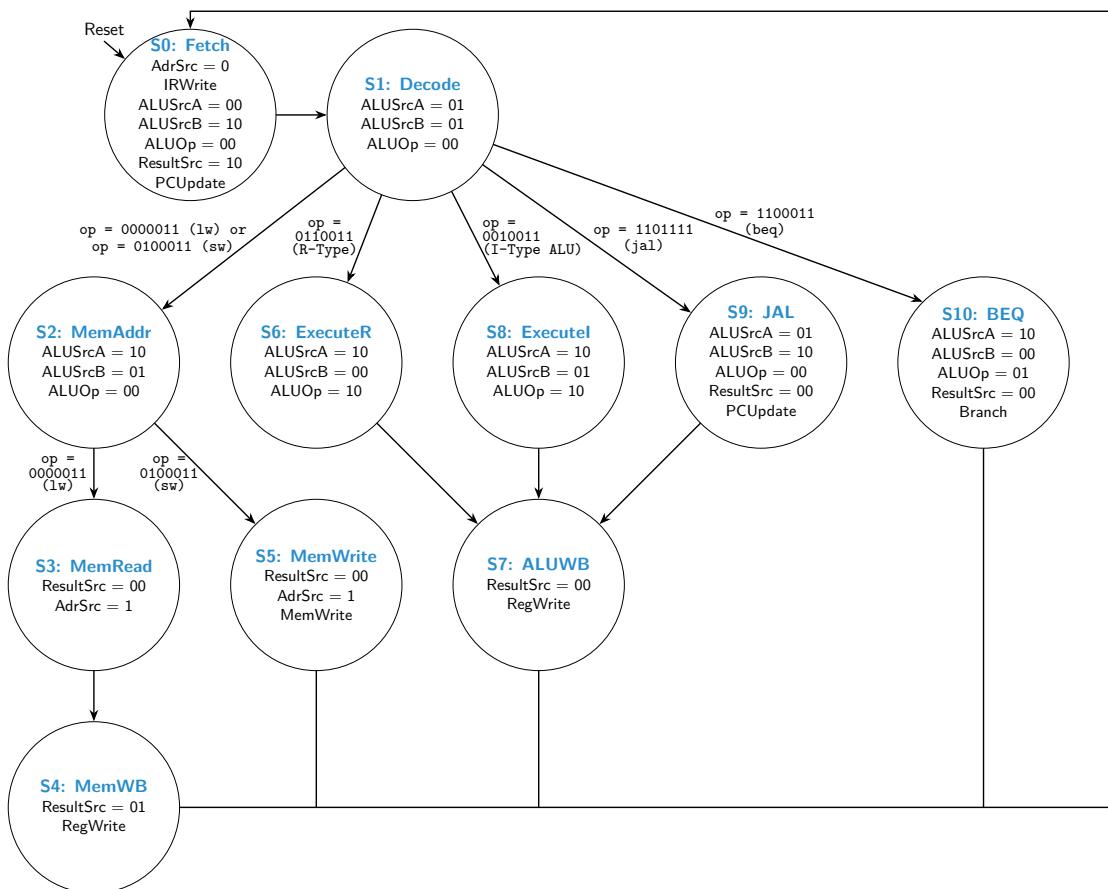
4.2 Single-Cycle RISC-V Processor (with Branch Logic)



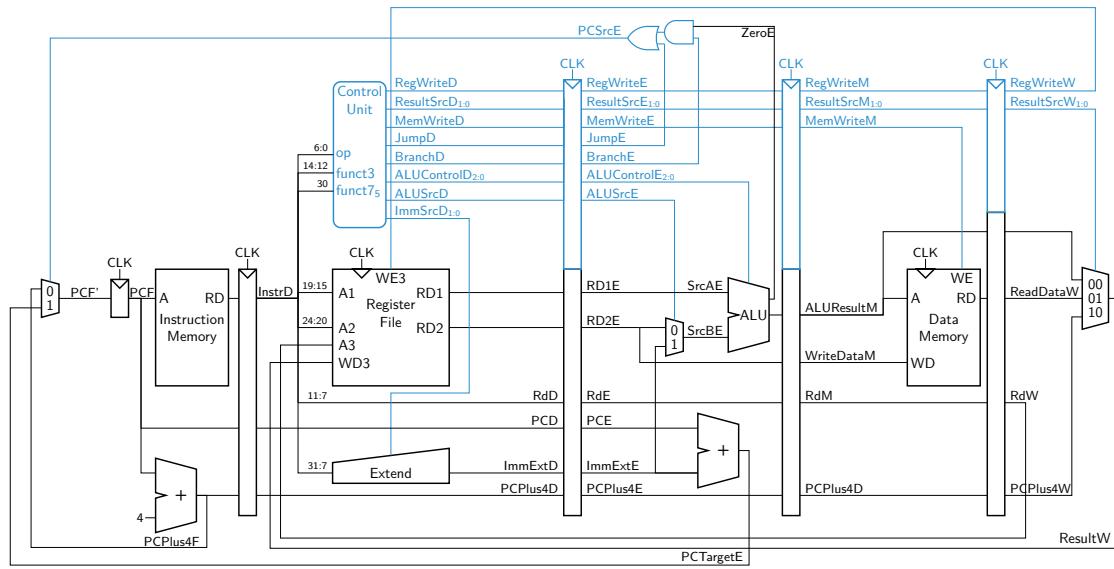
4.3 Multi-Cycle RISC-V Processor



Automaton:



4.4 Pipelined RISC-V Processor



4.5 Pipelined RISC-V Processor with Hazard Unit

