Assembly & Verilog

Richard Robinson, B.Eng. Cand.

September 21, 2018

Chapter 1

Using RISC V

1.1 Start of Program

A typical RISC V or Assembly program starts with the following lines:

```
ORG     96                  # declares init address
DD      42, 100, 19, 2000   # stores values in mem
addi    x1, x0, 96          # initializes init address
```

The first line contains the `ORG` command, which declares the initial address. The `DD` command stores its arguments in memory to be later used by the program. Lastly, the `addi` command with the `x1, x0` arguments initializes the registers to the initial address in `ORG`.

Values need not be explicitly stored in memory as will be discussed in the following section.

1.2 Using Registers

There are two primary methods of loading values into registers;

```
addi    x4, x0, 42          # method 1
ld      x4, 8(x1)           # method 2
```

In the first method, the value 42 is directly loaded into `x4`. In the second method, the value must first be stored in memory via the `DD` command. To load the nth number of `DD`, `8n(x1)` is used as $8n + 96$ is the address of the value.

Consequently, values are loaded from registers into memory via

```
sd      x19, 88(x1)         # stores reg in mem
```

which stores the value contained in `x19` into memory address `88 + x1`, which in this example would be $88 + 96$.