# Instruction Set Architecture Design and Implementation

## Learning Outcomes

At the end of this lab, you should be able to:

1. design and implement a simple ISA;
2. write a simple assembler

## Contents

## 1 Resources

- Video: https://youtu.be/CzyXb_T-xgU.
- Source Codes: https://git.io/JU3al
- Online VHDL Tool: https://www.edaplayground.com/home

## 2 Discussion

The **processor(CPU)** is composed of the **datapath** and **control**. In the previous labs, you learned that combinational and sequential circuit elements are used as building blocks to create the functional components of the datapath and control. Examples of these functional elements include the **ALU, Register File, Program Counter, and Memory**. You also learned that a **clock** drives the execution and control is implemented using a **finite state machine** for **fetch-decode-execute cycle**. One question that we can answer next is: *How do we program the CPU?*

### 2.1 Instruction Set Architecture

Instruction Set Architecture (ISA) is an **abstraction** between the hardware and the lowest-level software. It includes anything programmers need to know to make a binary machine language program work correctly.

allows computer designers to talk about functions indepedently from the hardware that performs them.

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
```

```
ENTITY clocker_tb IS
END clocker_tb;
ARCHITECTURE behavior OF clocker_tb IS
   --100Mhz
   CONSTANT frequency: integer := 100e6;
   CONSTANT period : time := 1000 ms / frequency;
   SIGNAL clk : std_logic := '0';
BEGIN
   clk <= not clk after period / 2;
   -- do some stuff here using clk as input
END ARCHITECTURE;
```

## 3 Summary

In this lab, you learned some of the sequential elements that are useful in the design of a processor as well as the importance of clocks. We also showed the design and implementation of a simple traffic light system using finite state machines since a simple truth table is not enough to characterize a sequential system.

You should now be able to tell whether a functional component of a datapath and control is composed of a combinational or sequential element.

## 4 Learning Activities

Download the source codes for this lab then try experimenting by adding more test cases in the testbenches. Submit a PDF document that shows screenshots of your modifications and runs.

## 5 Self-Assessment Questions

1. What is the main purpose of clocks in sequential circuits?
2. What is the difference between a clocked latch and a flip-flop?
3. Why can't a multiplexer be used in RAM?
4. Why is SRAM more expensive than DRAM?
5. If my CPU is clocked at 800 MHz, what is the period?

## 6 Deliverable

Your final deliverable for this lab is implement the RAM in Figure **??**. Submit the VHDL code including a testbench as well as images of the waveforms. NOTE: Enable lines should be connected to the output of the decoder and the rightmost Din in the figure should be Din[0].

## 7 Further Reading

- https://www.doulos.com/knowhow/vhdl/simple-ram-model/

## References

[1] David A. Patterson and John L. Hennessy. *Computer Organization and Design: The Hardware/Software Interface, ARM Edition.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, arm edition, 2017.