Lab 7 – LC-3 Datapath

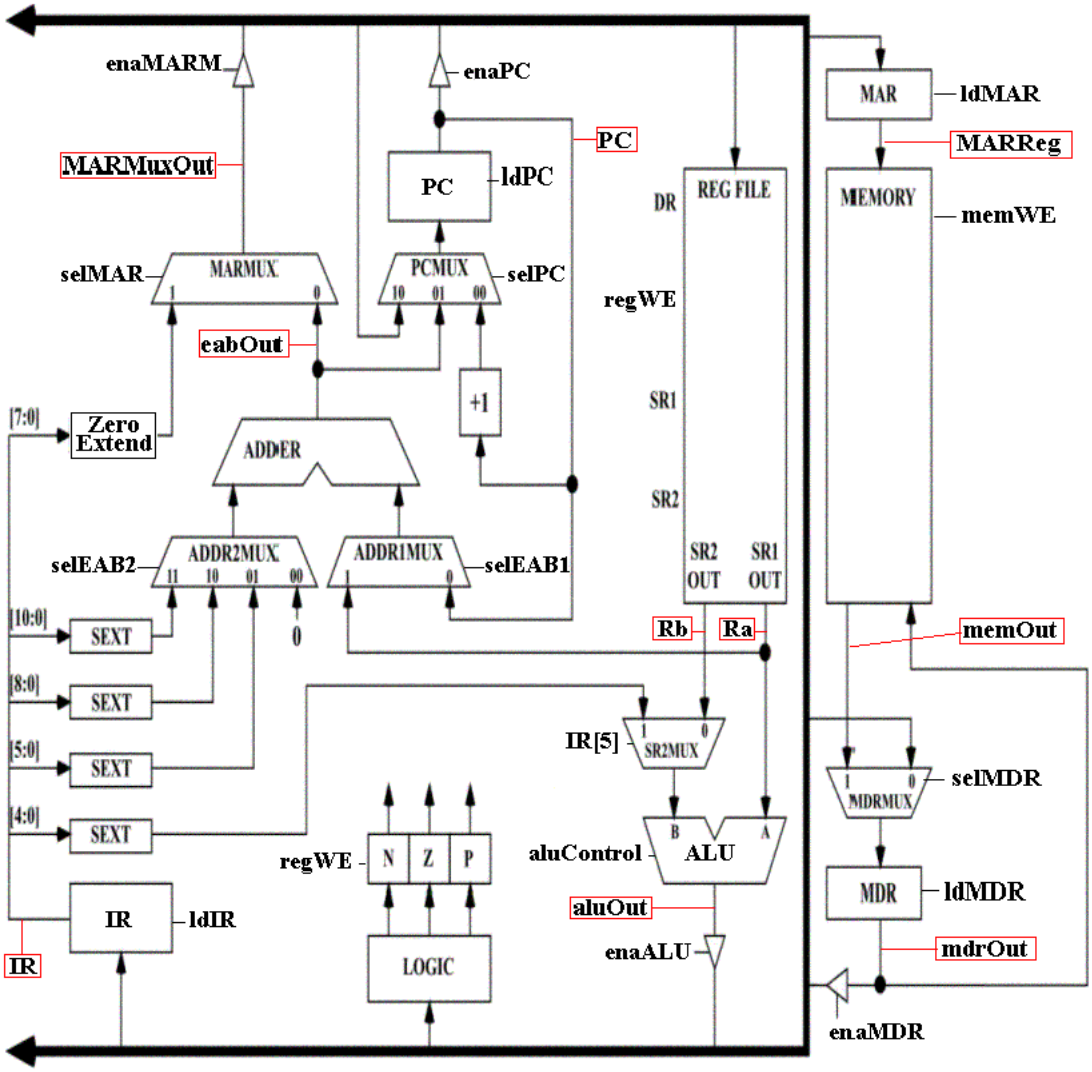
1 Objective

To understand and to design the major elements of the LC3 Datapath.

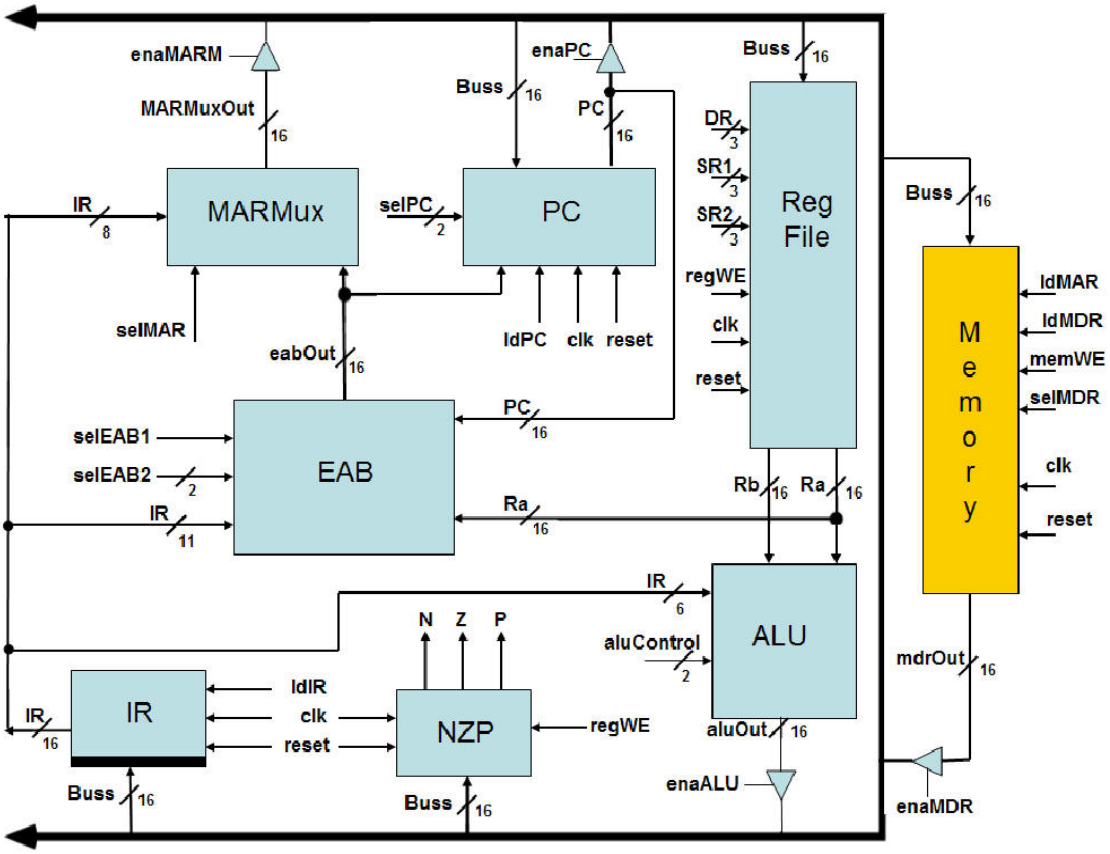
2 Introduction

In this assignment you will design most of the major functional blocks in LC3 Datapath. The specific parts which you will design include the following : Effective Address Block, MARMux, PC Block, Instruction Register, NZP flags, ALU, and Register File.

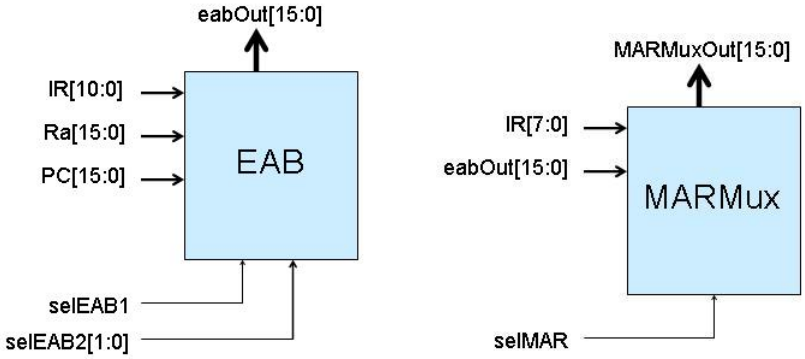
The LC-3 Datapath is as follows:

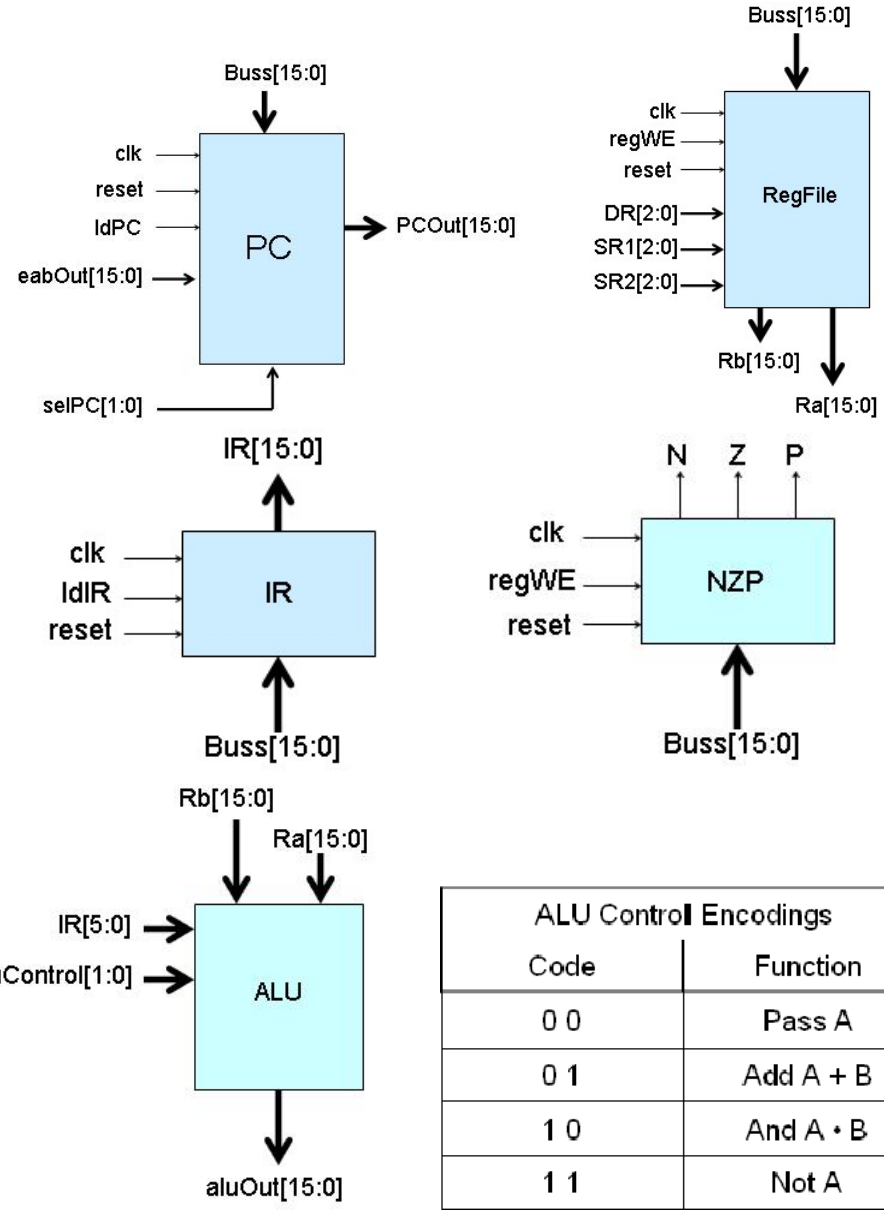


In the datapath we will combine several blocks to form this:



The individual blocks are described as follows:





ALU Control Encodings	
Code	Function
0 0	Pass A
0 1	Add A + B
1 0	And A • B
1 1	Not A

2.1 Procedure

1. Design each of the above functional blocks as individual VHDL components. For the purposes of consistency and testability, name all of the modules **exactly** as shown above. Also name the I/O signals to/from the blocks **exactly** as shown above.

2. Test each block to your own satisfaction. Prove to yourself that it works. This step is very **critical**. These blocks will be used in the following two labs. If they do not operate correctly, your upcoming labs will not work. On the web page for this labe, you will find some testbench files to use as a starting point for testing the modules. Please Note: these testbenches do **NOT** fully test the modules.

   In order to completely verify the modules, you will have to add more tests to these .do

files. (For the purposes of the lab submission, however, you only need to turn in the simulations using the files that have been given to you). Document your VHDL code, testbench files and output waveforms for each functional block.

2.2 Hints

1. The ADDR2MUX in the EAB and the PCMux are each made from a 4 to 1 Mux which takes four 16-bit values as inputs and produces a 16-bit value as an output.

   The general form for a 4 to 1 Mux using VHDL is::

```
result <= a when sel = "00" else
          b when sel = "01" else
          c when sel = "10" else
          d when sel = "11" else
          (others => '0');
```

   It is important that you include all possible combinations of the select line and set a **default** case.

2. All registers should use the register you designed as part of the Register file lab.

3. The ALU Block

   The ALU is very similar to the one you built earlier in the semester. You may utilize the work you have already done, however, the ALU block is much easier to implement using design techniques we have discussed. Don't forget to include an ALUMux in your design.

4. The Register File

   Use the register file designed in the last lab. Ensure that all of the input/output ports have exactly the names specified above.

5. Below you will find the code for a Tri-state Bus Driver.

```
buss <= my_16bit_wire when ctrl = '1' else
    (others => 'Z');
```

Document your VHDL code, testbench files and output waveforms for each functional block. Don't forget your TOC, assumptions and anomalies page.

To pass of this lab, show a TA the waveforms generated by simulating the testbench files you were given.