Lab Manual
Department of Electrical and Computer Engineering
School of Engineering and Physical Sciences
North South University, Bashundhara, Dhaka-1229, Bangladesh

Experiment No: 07
Experiment Name: Build a single cycle datapath

Introduction:

The datapath is an important part of a processor, since it implements the fetch-decode-execute cycle. The general approach for datapath design is to

(1) determine the instruction classes and formats in the ISA,

(2) design datapath components and interconnect them for each instruction class or format, and

(3) compose the datapath segments designed in Step 2) for multiple instructions to yield a composite datapath.

A very simple datapath components include *memory* that stores the current instruction, *program counter or PC* that stores the address of current instruction, and *ALU* that executes current instruction. The interconnection of these simple components forms a basic datapath

Objective:

We will have following objectives to fulfill:

1) Design an Instruction Fetch Unit of the datapath

2) Design an R-format and Load/Store Datapath

4) compose the datapath segments designed above to yield a complete single cycle datapath

Experiment Details:

Assume, a 16 bit ISA with following fields. The formats of the instruction are as follows:

R-type

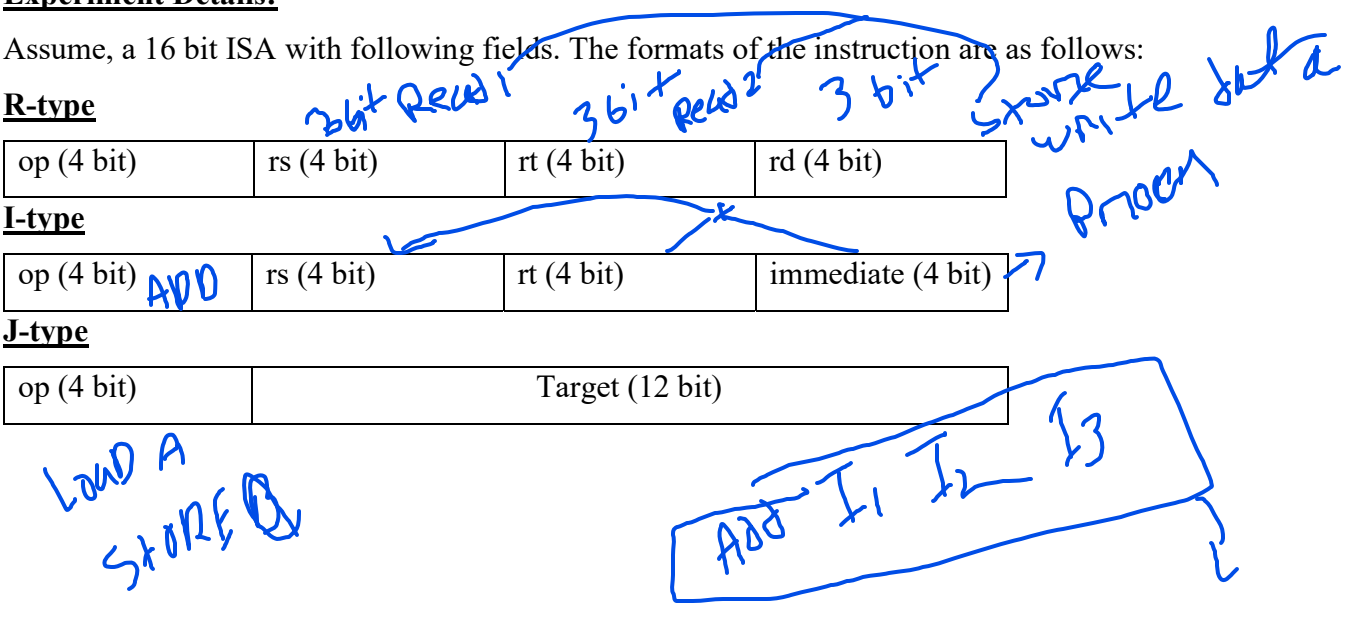
op (4 bit)	rs (4 bit)	rt (4 bit)	rd (4 bit)

I-type

op (4 bit)	rs (4 bit)	rt (4 bit)	immediate (4 bit)

J-type

op (4 bit)	Target (12 bit)

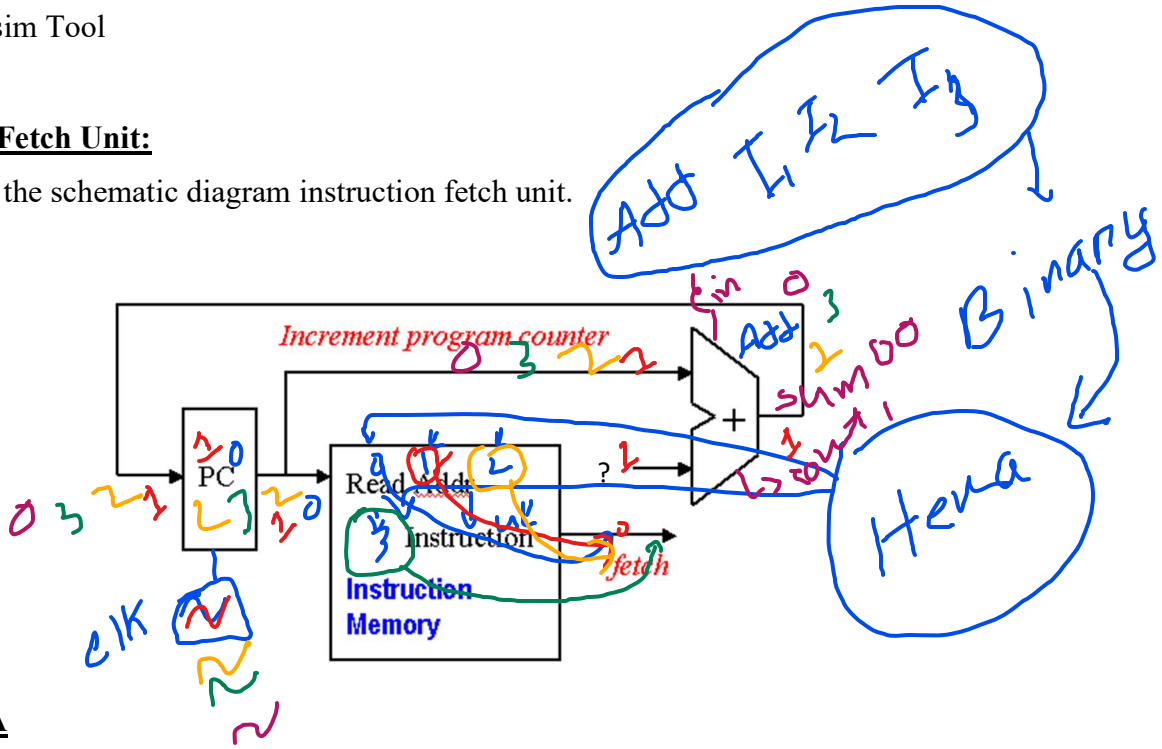


Equipment/Tool

Logisim Tool

Instrcution Fetch Unit:

Following is the schematic diagram instruction fetch unit.

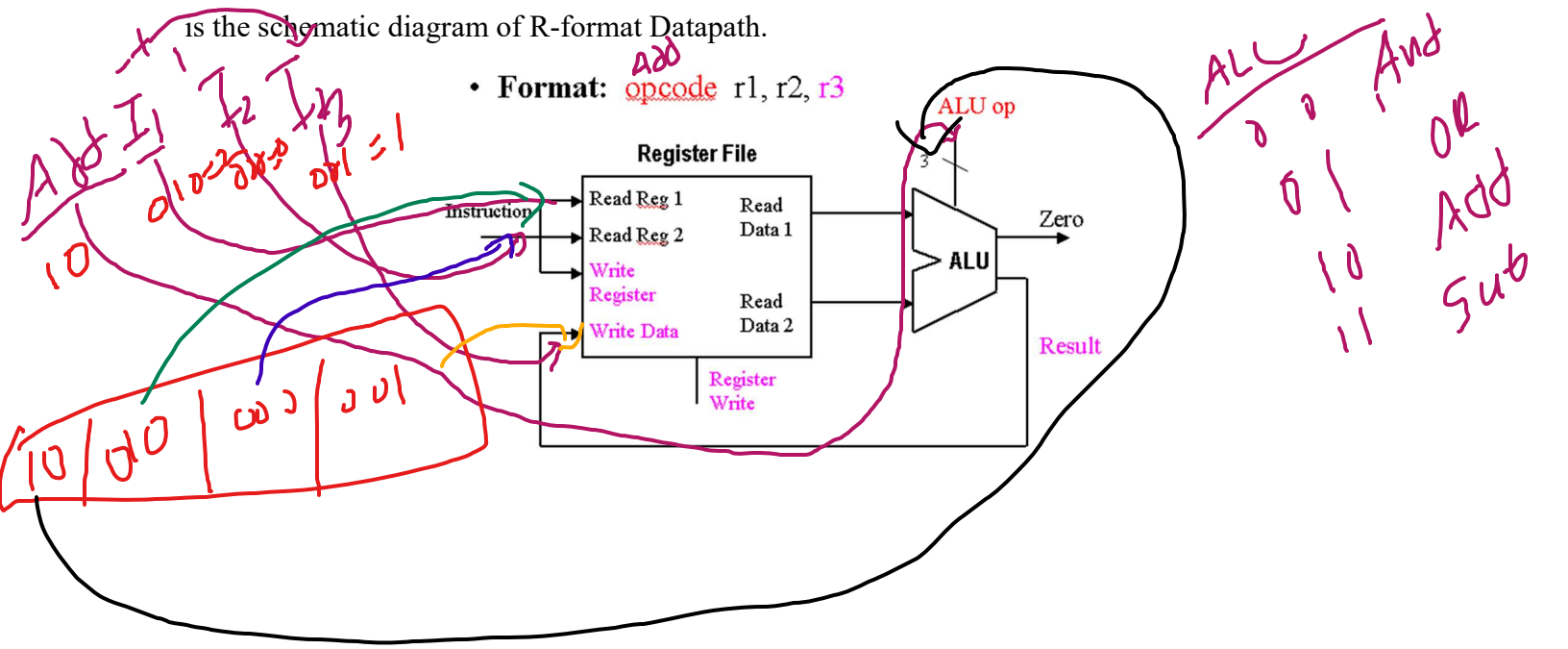


Task Lists A

1. Implement the Instruction Fetch unit based on the above diagram in logisim. Label each of the input/output/mux selections. Select the appropriate bus width.

R-format Datapath:

Implementation of R-format instructions datapath requires only *Register File* and the *ALU*. Therefore this is fairly straightforward. The ALU accepts its input from the ReadData1 and ReadData2 ports of the register file, and output of the ALU updates the contents of the register file. RegWrite signal must be enabled during the writing of output to the register file. Following is the schematic diagram of R-format Datapath.

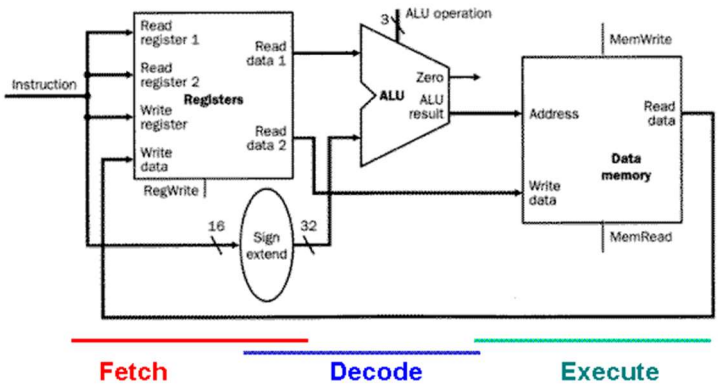


Task Lists B

1. Implement the R-format datapath based on the above diagram in logisim. Label each of the input/output/mux selections. Select the appropriate bus width.

I-format (Load/Store) Datapath:

The load/store datapath uses instructions such as *lw $t1, offset($t2)* and *lw $t1, offset($t2)*. Here offset denotes a memory address offset, that is applied to the base address in register $t2. The lw instruction reads from memory and writes that into register $t1 of the register file. On the other hand, the sw instruction reads from register $t1 of register file and writes that into memory. To compute the memory address, we have to sign-extend the 4-bit offset to a 16-bit signed value. This is done using the sign extender. Following is the schematic diagram of I-format(load/store) Datapath.



Task Lists B

1. Implement the I-format (load/store) datapath based on the above diagram in logisim. Label each of the input/output/mux selections. Select the appropriate bus width.
2. Combine Instruction Fetch unit of *Task List A* and R-format Datapath of *Task List B* with I-format (load/store) datapath to yield a single cycle datapath

Assignment:

1) Prepare the lab report.
2) Take a screenshot of your implementation and later include it in your lab report.