Introduction to Computing: Task 4

Fall 2018

Dr. Ramesh Yerraballi

Designed by Matthew Barondeau

The University of Texas at Austin

Due: Friday 11/30, 11:59pm

Purpose

In the previous task, you built an ALU and connected it to a bus. In this task, you will expand the previous task to execute custom assembly instructions and see how they execute on the datapath. You will also add operations with immediate values. Then you will write a report tying together the previous two tasks.

Task Requirements

1. Create an 8-bit Instruction Register. This Register will hold the current instruction that will be executed.
2. Add a mux that will allow the ALU to choose between an immediate value of 8 bits or SR2. The control signal to the mux will be determined by the instruction being executed.
3. Remove all control signals buttons. The only button in your system should be the clock and LD.REG.
4. Wire the control signals to the Instruction register. Different bits in the IR will need to control different parts of the data path.
5. The instruction register bits will be used to drive the control signals. Look below on the next page for the encoding of the instructions. Connect the datapath back up and ensure that all 8 variants run correctly.
6. Sign extend the 3-bit immediate value to 8 bits when feeding it to the ALU. See the block diagram below
7. Draw a diagram explaining your connections. Specifically, which bits of the instruction registers did you use for the signals? Did you have to add any extra logic?
8. Answer the following questions:
    a. What is faster to execute: Add or XOR? Why?
    b. Using this architecture, there are two ways to multiply a register by 4. Explain them both
    c. In what case might it be beneficial to include a pass through for the ALU?
    d. What is the model of computing with only 1 bus?
    e. How would you implement a divide instruction using the instructions given? Suppose the divisor is not a power of 2.
    f. What is missing in this architecture? Hint: where do I get the next instruction?
    g. What was the biggest hurdle in completing these tasks?

Instructions

ADD DR, SR

7 6	5 4	3	2	1 0
00	DR	0	0	SR

ADD DR, Immediate

7 6	5 4	3	2 0
00	DR	1	Immediate

AND DR, SR

7 6	5 4	3	2	1 0
01	DR	0	0	SR

AND DR, Immediate

7 6	5 4	3	2 0
01	DR	1	Immediate

XOR DR, SR

7 6	5 4	3	2	1 0
10	DR	0	0	SR

XOR DR, Immediate

7 6	5 4	3	2 0
10	DR	1	Immediate

LSHF DR, Immediate

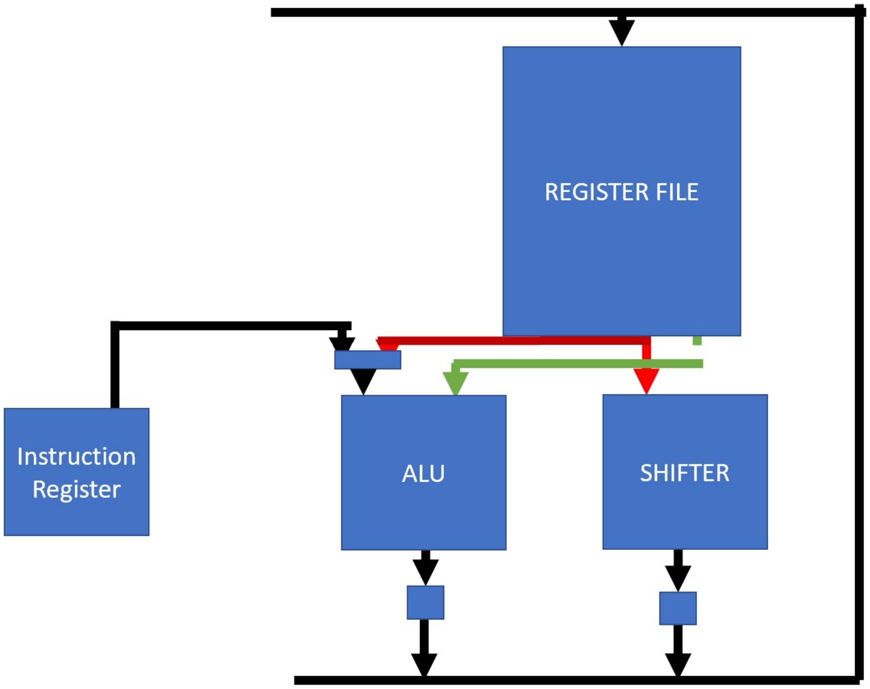
7 6	5 4	3	2 0
11	DR	1	Immediate

RSHF DR, Immediate

7 6	5 4	3	2 0
11	DR	0	Immediate

Block Diagram

Shown below is a basic block diagram for this system. I have not included the other control signals that will come from the IR. It is your job to figure those out. You are free to talk to your classmates, TAs and Professor about these signals.



Submission

Submit a Task4.circ containing the working architecture to canvas. You will also submit one pdf titled “Task4_Readme.pdf” that contains the answer to your questions and a list of what you connected all the signals to.

Useful Blocks

The following blocks were used by my implementation and you may find them useful for forwarding control signals:

Decoder, Demultiplexer, Splitter, Bit Extender, Controlled Butter, and Multiplexer

Grading

Item	**Points**
Instruction Register Implemented	10
Immediate Mux	15
Sign Extend 3 bit to 8 bit	5
ADD Register	5
ADD Immediate	5
AND Register	5
AND Immediate	5
XOR Register	5
XOR Immediate	5
LSHF	5
RSHF	5
Signal Mapping	15
Questions	15