Comp E 475

MICROPROCESSORS

HW8: Decoder/Control

Student: Vano Mazashvili

Red ID #: 822059245

Contents

[Task Description 2](#_Toc59397558)

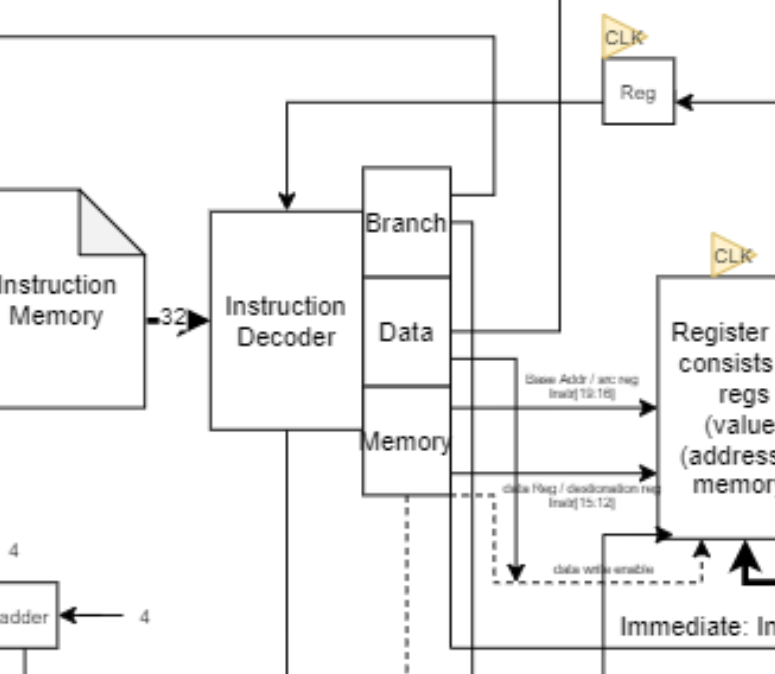
[Solution 3](#_Toc59397559)

[Simulation & Verification 5](#_Toc59397560)

[Conclusion 6](#_Toc59397561)

# Task Description

For this assignment, I wrote a Verilog module responsible for decoding branch, data processing and memory instructions (ARM architecture).

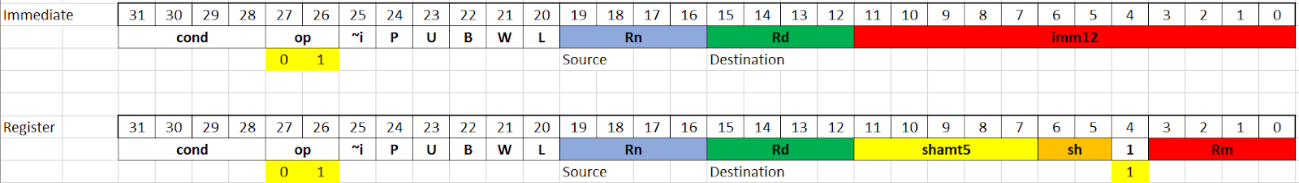


It gets 32 bit instruction from the instruction memory and flags from the ALU. Similarly to HW5, we had to implement a simple if-else, or case statements to perform the operations.

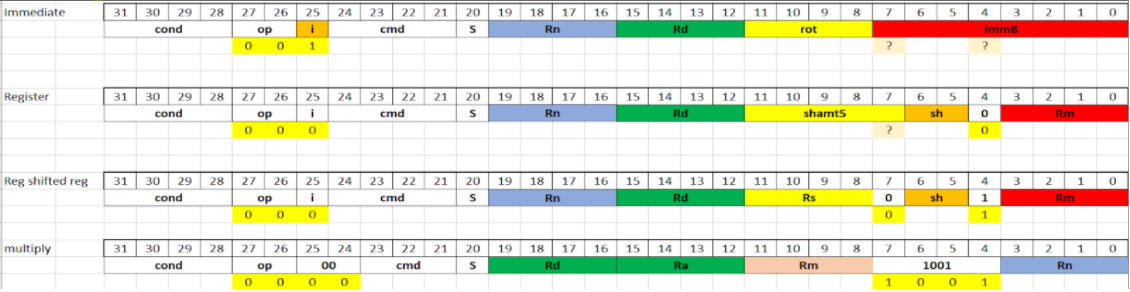
# Solution

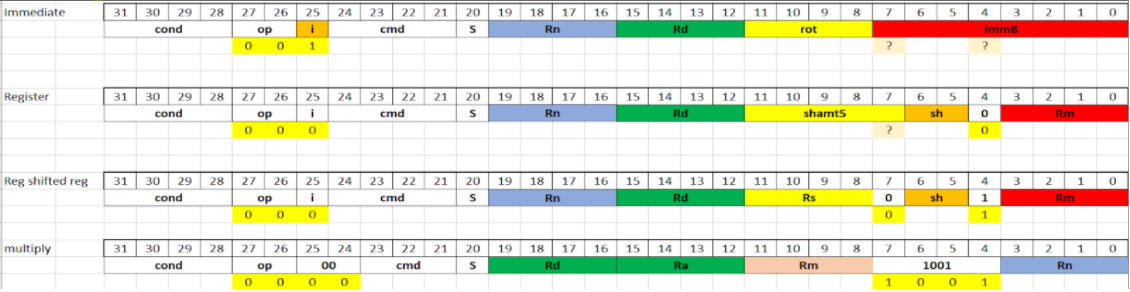
For this solution, I used the tables from the lecture slides.

This is for memory:

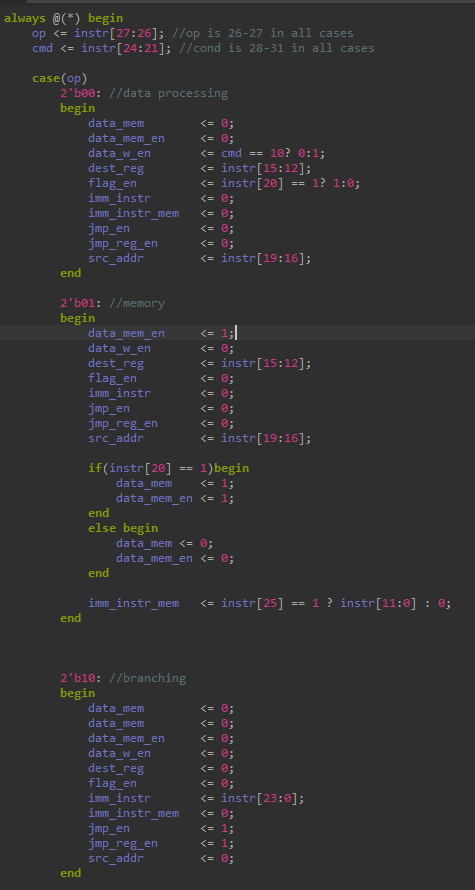


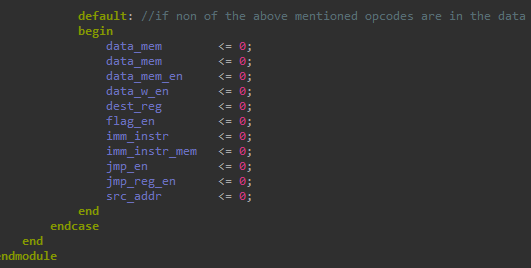
For the data processing:

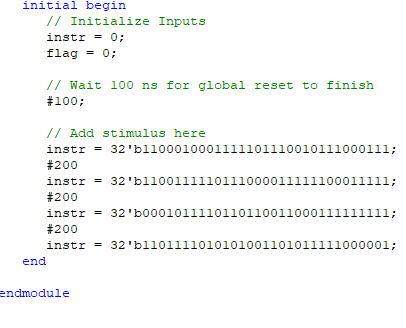


Branching: 

Below, I have presented the main part of the source code, which decodes data processing, memory, and branching instructions according to the above-mentioned tables.





This is a testbench, which tested the code for the 4 different opcode instructions 

# Simulation & Verification

The code synthesized without errors. One warning was about spartan3E 100x fpga’s resources, but after implementing the design on slightly faster FPGA, there were no issues.

# Conclusion

In conclusion, I have written the simple instruction decoder with case and if-else statements. The design behaves according to the provided ARM 32-bit instruction. The source code and synthesis report is on Github.com/vmazashvili in corresponding decoder-control repository.