# ELC 363 - LABORATORY #5

# ARM IMPLEMENTATION

In this laboratory, the single cycle ARM architecture will be implemented and verified in Verilog® using an as high as possible level of abstraction design methodology. This implementation will only have to support the following instructions (simplified ISA):

- memory-reference instructions:
    - LDUR, STUR
- arithmetic-logical instructions:
    - ADD, SUB, AND, ORR
- control flow instructions:
    - CBZ, B

Since the processor has been architected already (this is an implementation job), the design approach should be bottoms-up using the ALU that you designed in LABORATORY #3.

The next step is to build the register file using Fig. 1 and Fig. 2 as a guide. In Fig. 2, the AND gates should be 3-input AND gates with the third input connected to the CLOCK.

- Built using D flip flops

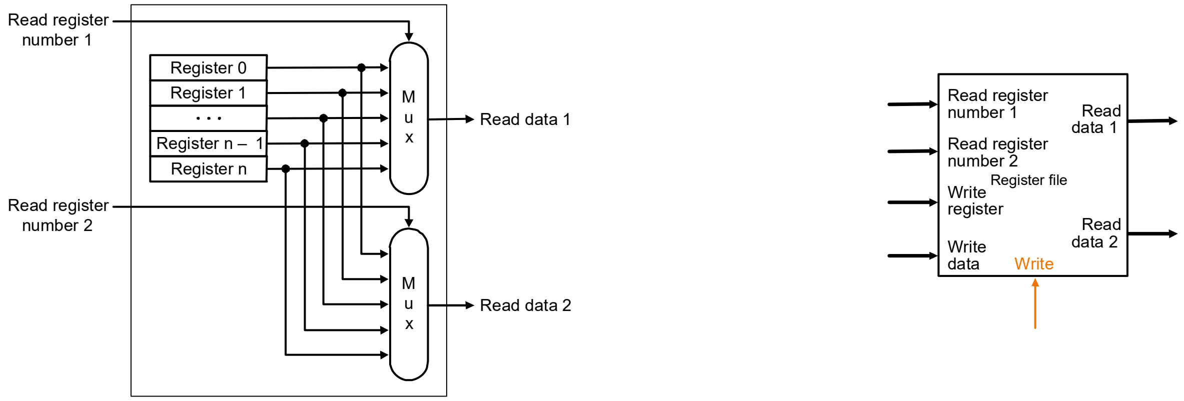


Fig. 1. Register File Architecture (a)

- Note: we still use the real clock to determine when to write

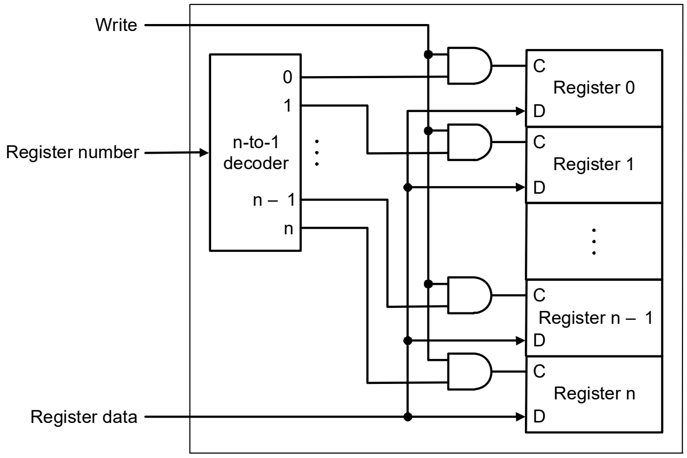


Fig. 2. Register File Architecture (b)

At this point, the overall architecture can be implemented per Fig. 3. Note that the memories do not belong inside the implementation of the processor and need to be developed as separate modules, and connected to the processor in the test bench. The memories, as well as the register file, are of the asynchronous read, synchronous write type. The Verilog® Reference Sheet handed of in class contains a memory declaration that can be built on. The control for the one cycle implementation can be inferred from Fig. 4 and Fig. 5.

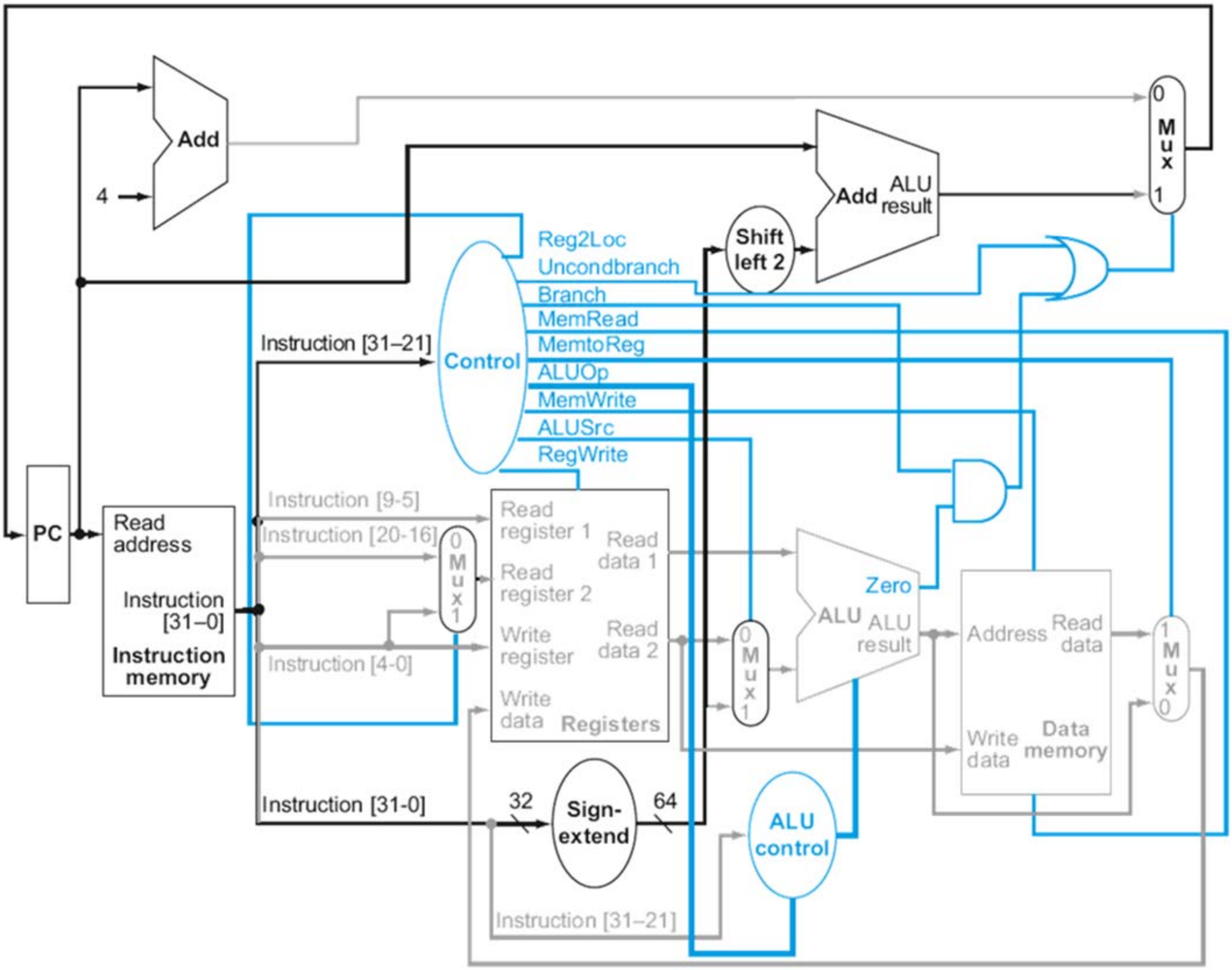
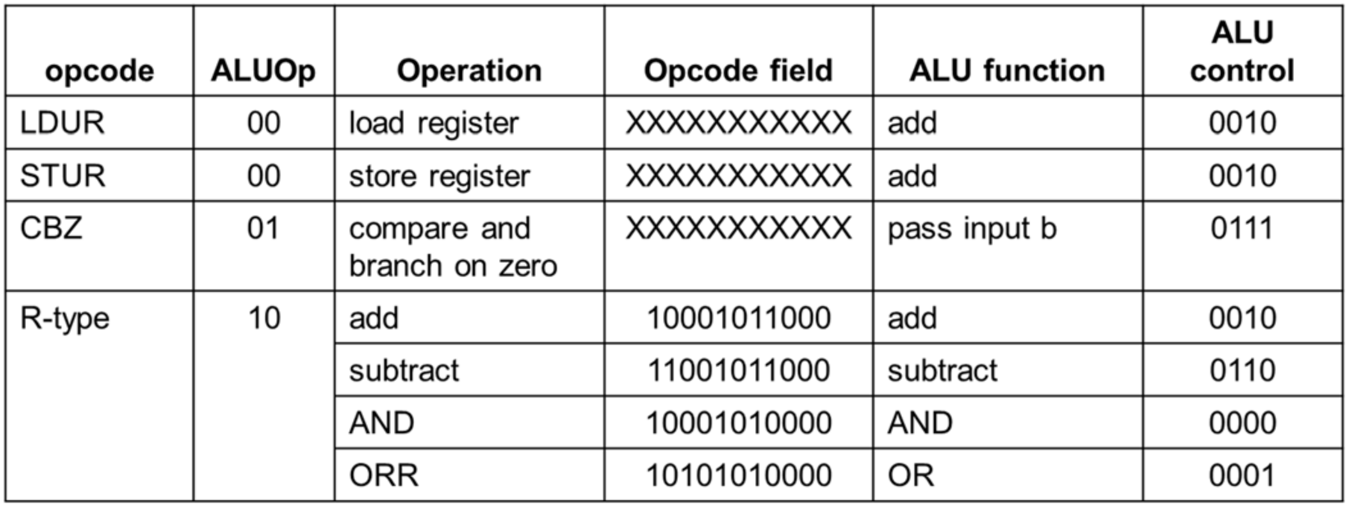
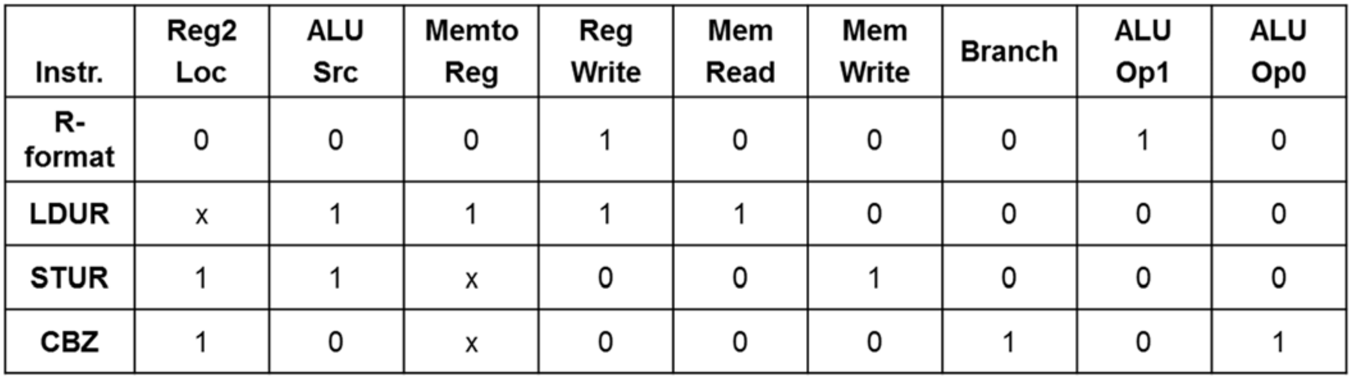


Fig. 3. Overall Architecture



| opcode | ALUOp | Operation | Opcode field | ALU function | ALU control |
|---|---|---|---|---|---|
| LDUR | 00 | load register | XXXXXXXXXXX | add | 0010 |
| STUR | 00 | store register | XXXXXXXXXXX | add | 0010 |
| CBZ | 01 | compare and branch on zero | XXXXXXXXXXX | pass input b | 0111 |
| R-type | 10 | add | 10001011000 | add | 0010 |
| | | subtract | 11001011000 | subtract | 0110 |
| | | AND | 10001010000 | AND | 0000 |
| | | ORR | 10101010000 | OR | 0001 |

Fig. 4. ALU Control Truth Table



| Instr. | Reg2 Loc | ALU Src | Memto Reg | Reg Write | Mem Read | Mem Write | Branch | ALU Op1 | ALU Op0 |
|---|---|---|---|---|---|---|---|---|---|
| R-format | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| LDUR | x | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| STUR | 1 | 1 | x | 0 | 0 | 1 | 0 | 0 | 0 |
| CBZ | 1 | 0 | x | 0 | 0 | 0 | 1 | 0 | 1 |

Fig. 5. Main Control Truth Table

The next step is to write a test bench to verify the design. The test bench should connect the CPU and the memories, initialize the program memory, and provide a clock and a reset. The memory should be initialized with a program that should have been hand assembled to exercise all the instructions in the simplified ISA at least once. The Verilog® Reference Sheet handed of in class and the Verilog® presentation discussed in class contains examples of clock declarations, initial memory loading, and others that can be used as starting points in this laboratory. Use non-blocking assignments and a small propagation delay for synchronous assignments. The test bench will have to run long enough (for enough clock cycles) for the program to execute. The minimum deliverables for this laboratory are the following:

a) All Verilog® code files (design code files and test-bench code file).

b) Test program in ARM assembly language.

c) Initial program memory load file (this is the assembled test program in machine language).

d) Waveforms that show the state of the CPU (PC and pertinent registers), and the pertinent memory contents after each instruction has been executed.

The work should be done independently by each team. A report with, at a minimum, all the items requested to be turned in is to be submitted by student by the due date discussed in class. All reports should be written in a word processor and similar productivity computer tools; no hand written reports will be accepted.

GRADING RUBRIC: The total grade for this assignment will be 28 points normalized to 100% for your report. Part (a) above will be worth 12 points, parts (b) and (c) will be worth 1 point each, and part (d) will be worth 8 points. The rest of your report will be worth 6 points, for a total of 28 points.

REPORT FORMAT: Free form, but it must be:

a. One report per team.

b. Have a cover sheet with identification: Title, Class, Your Name, etc.

c. COMPLETELY word-processed

d. Double spaced

e. 12 pt Times New Roman font

f. Fully justified (optional)

g. Outline of the body of the report: Introduction, Problem Description, Results, Discussion, and Conclusions.