Simple CPU

CS6133 - Computer Architecture I

Vikram Padman

Polytechnic Institute of New York University

vikram@poly.edu

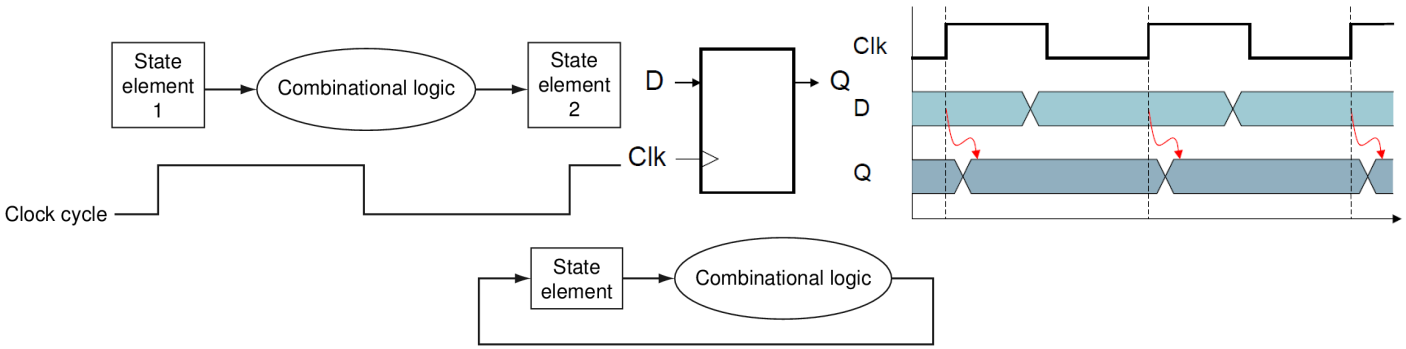
Agenda

1. **Clocking Methodology**
2. **Functional Units of a CPU**
   1. Program Counter (PC)
   2. Instruction Memory (IM)
   3. Register File (RF)
   4. ALU
   5. Data Memory (DM)
   6. Control Units (CU)
3. Activity

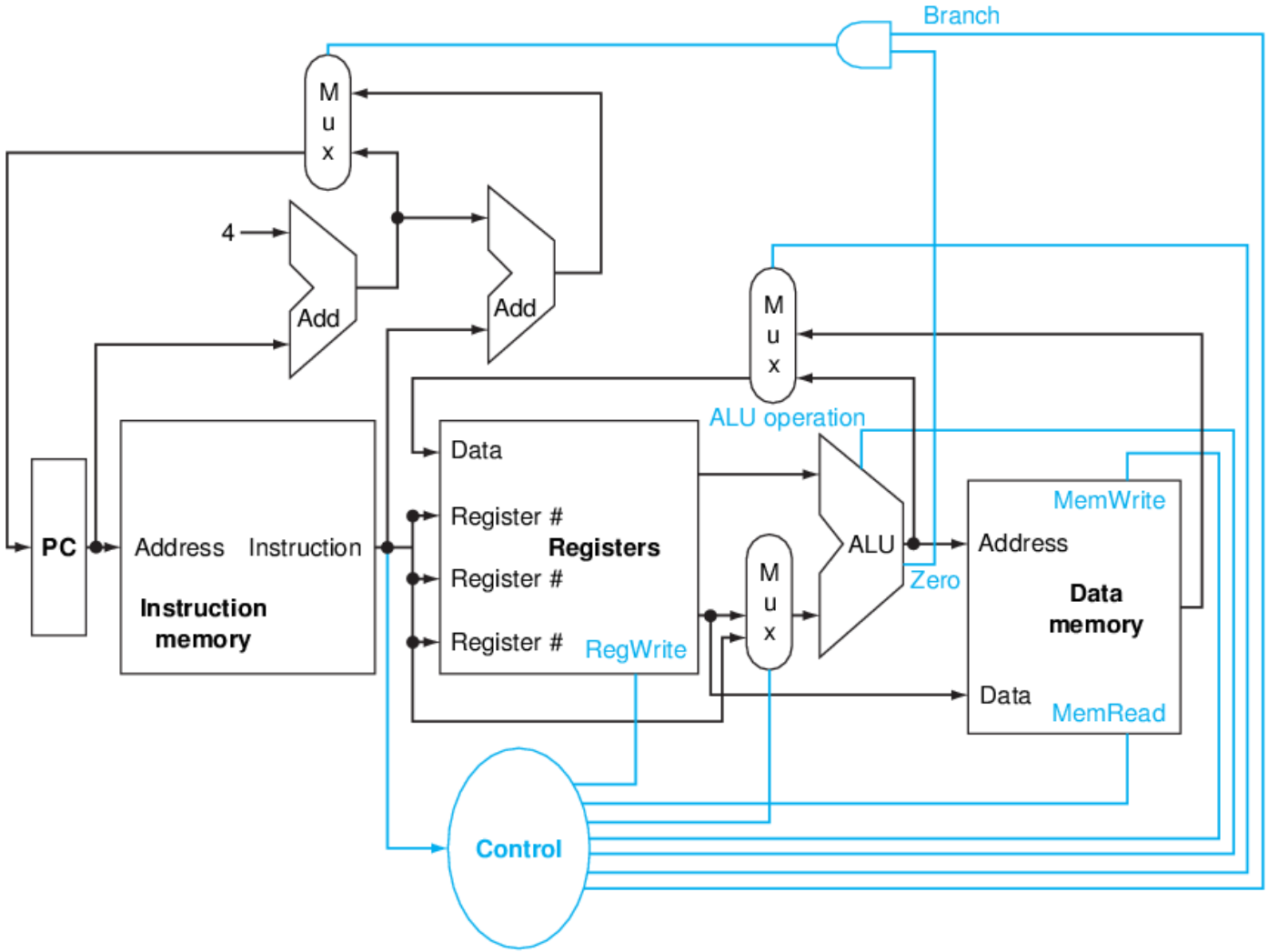
Lecture material from "Digital Design and Computer Architecture", Chapter 7 in first or Second edition, and "Computer Organization and Design" chapter 4 in fourth edition or chapter 5 in third edition

Clocking Methodology

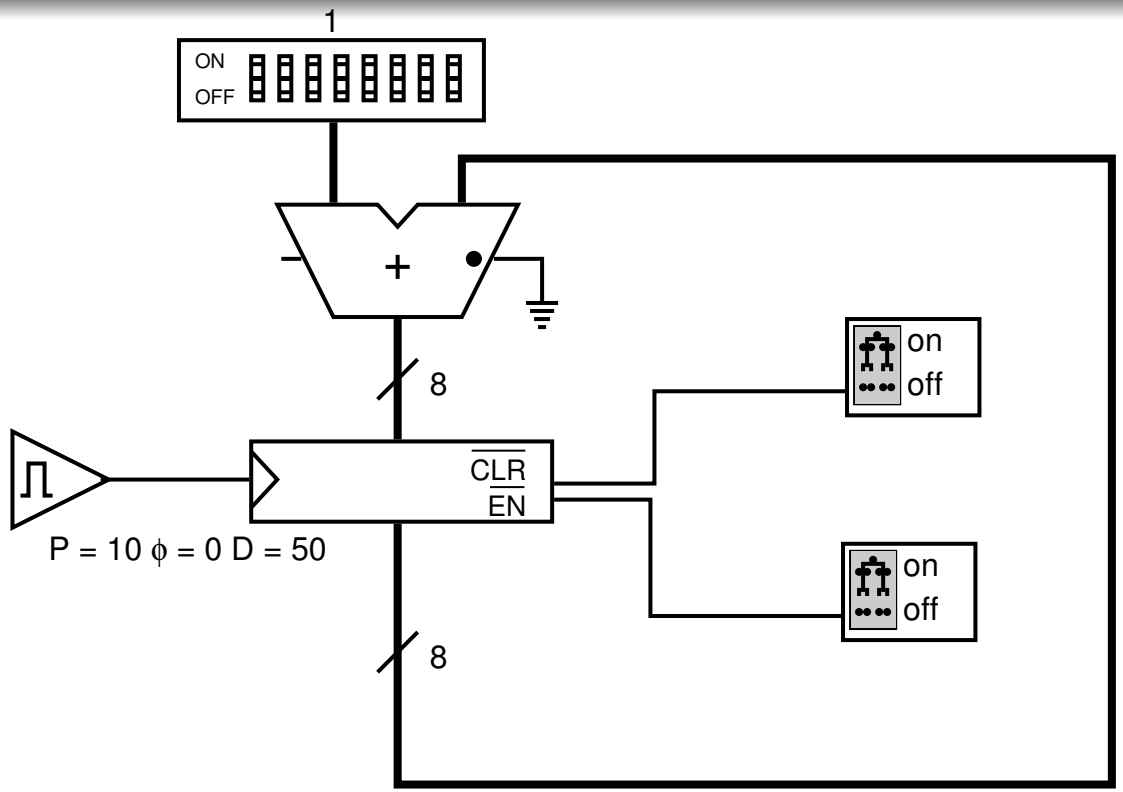
- Clock is the fundamental signal that provided timing to all logic elements
- Each component uses the clock signal to determine when it could read or write data
- A clocking methodology provided predictability.
- Commonly used clocking methodology:
  - Data is stored and updated by a state element (registers) only on a clock edge (Edge-Trigged)
  - Combinatorial logic element compute during clock cycle.

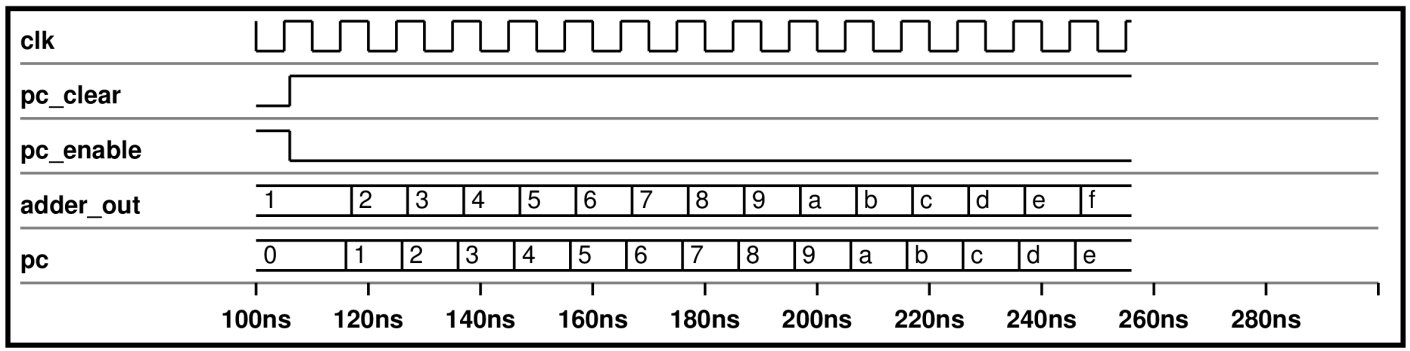


Functional Units of a CPU

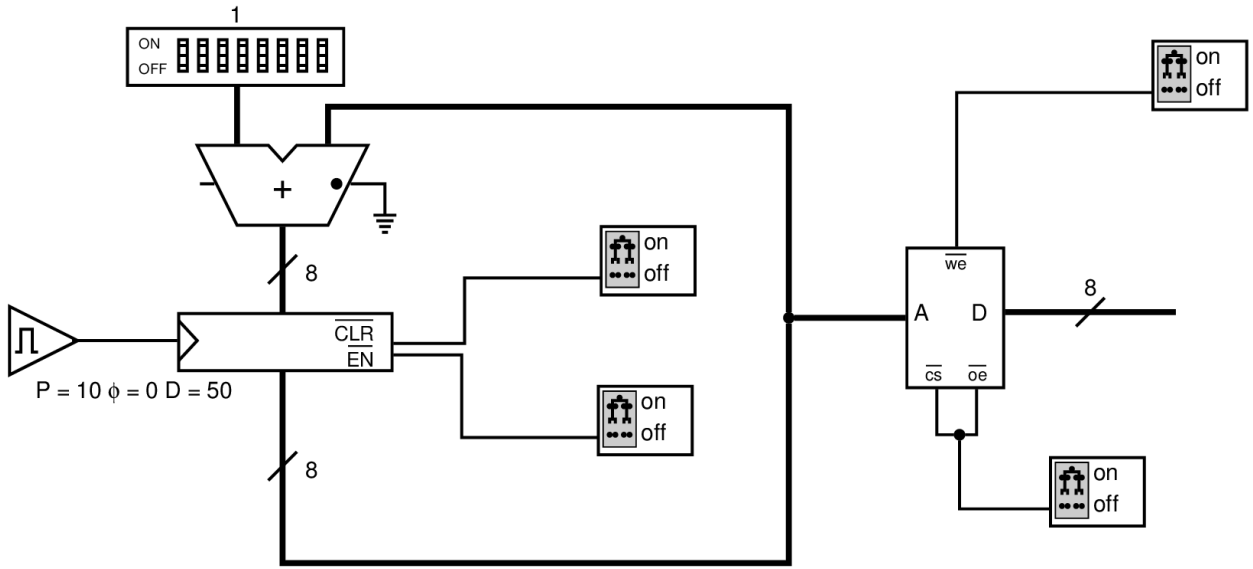


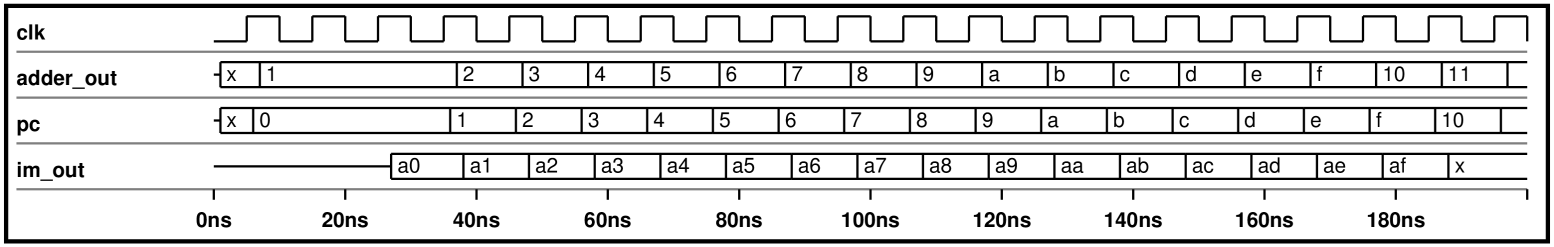
Program Counter (8-bit)



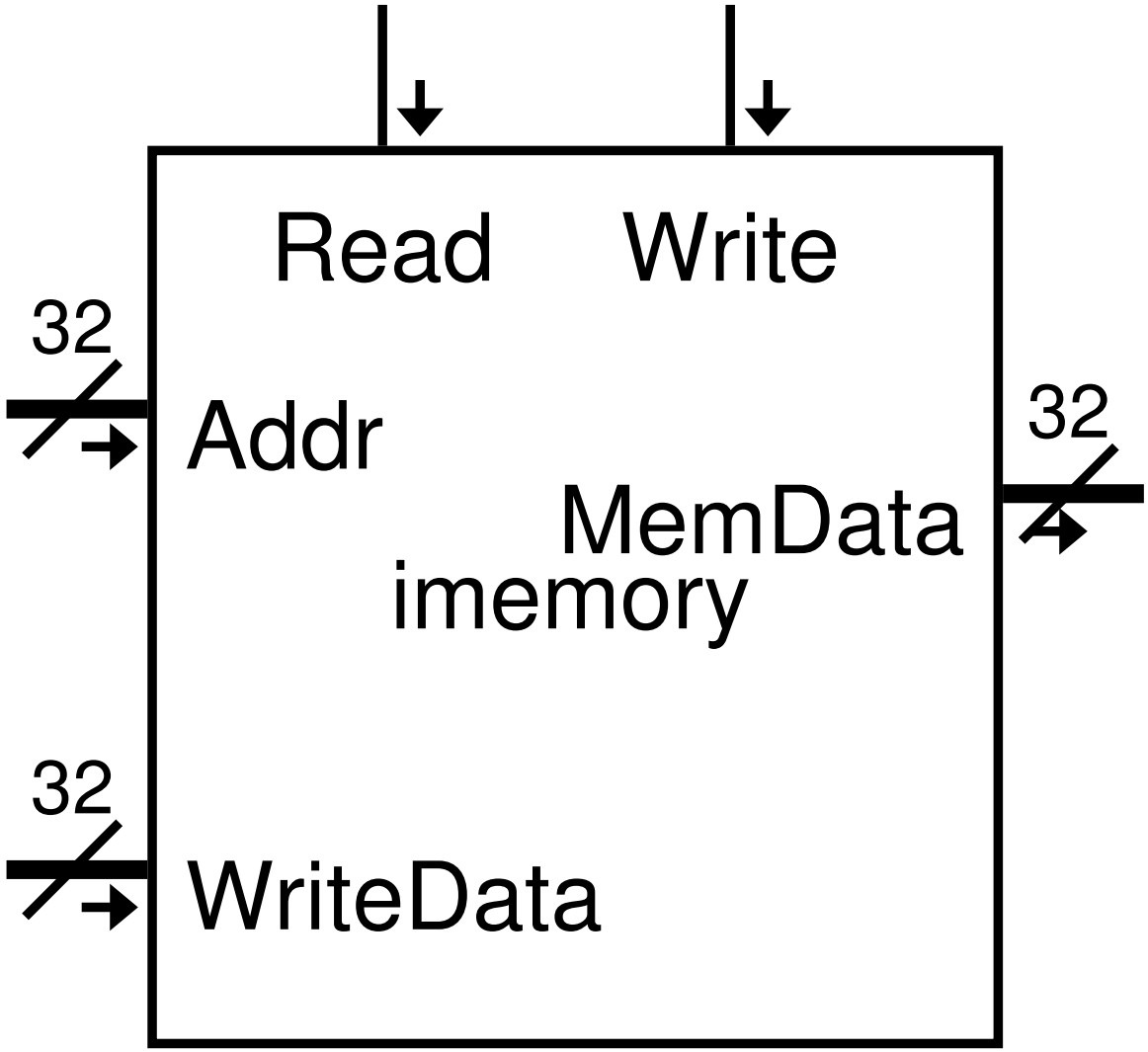


Instruction Memory (8-bit)

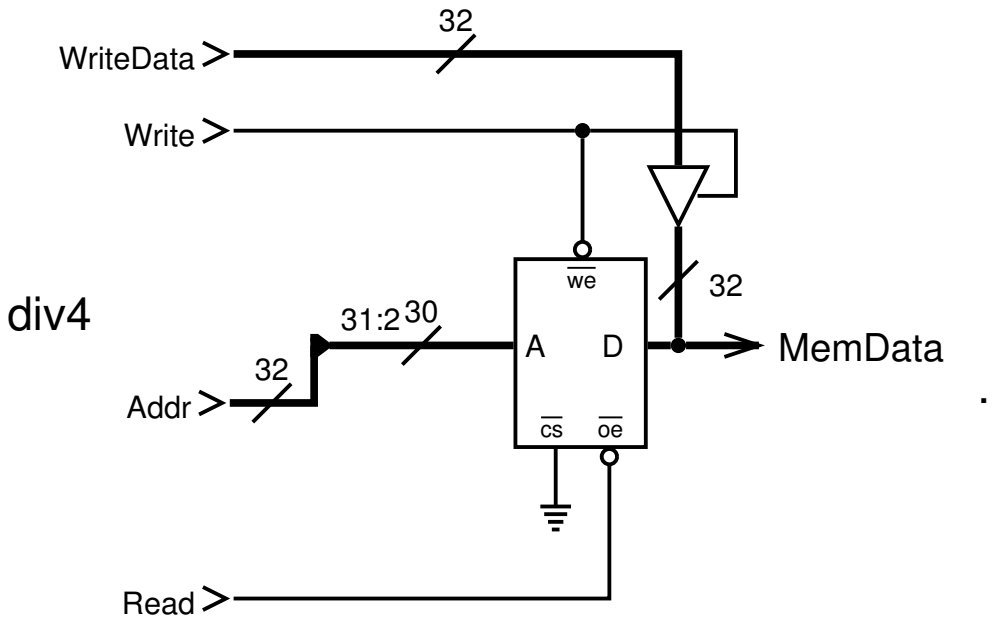




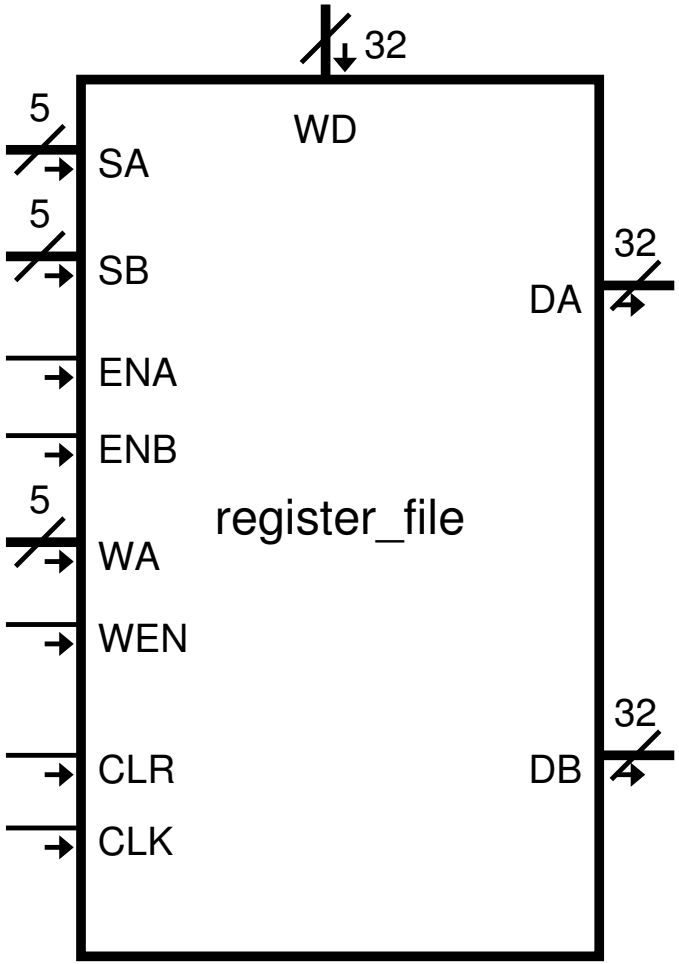
Instruction Memory Interface



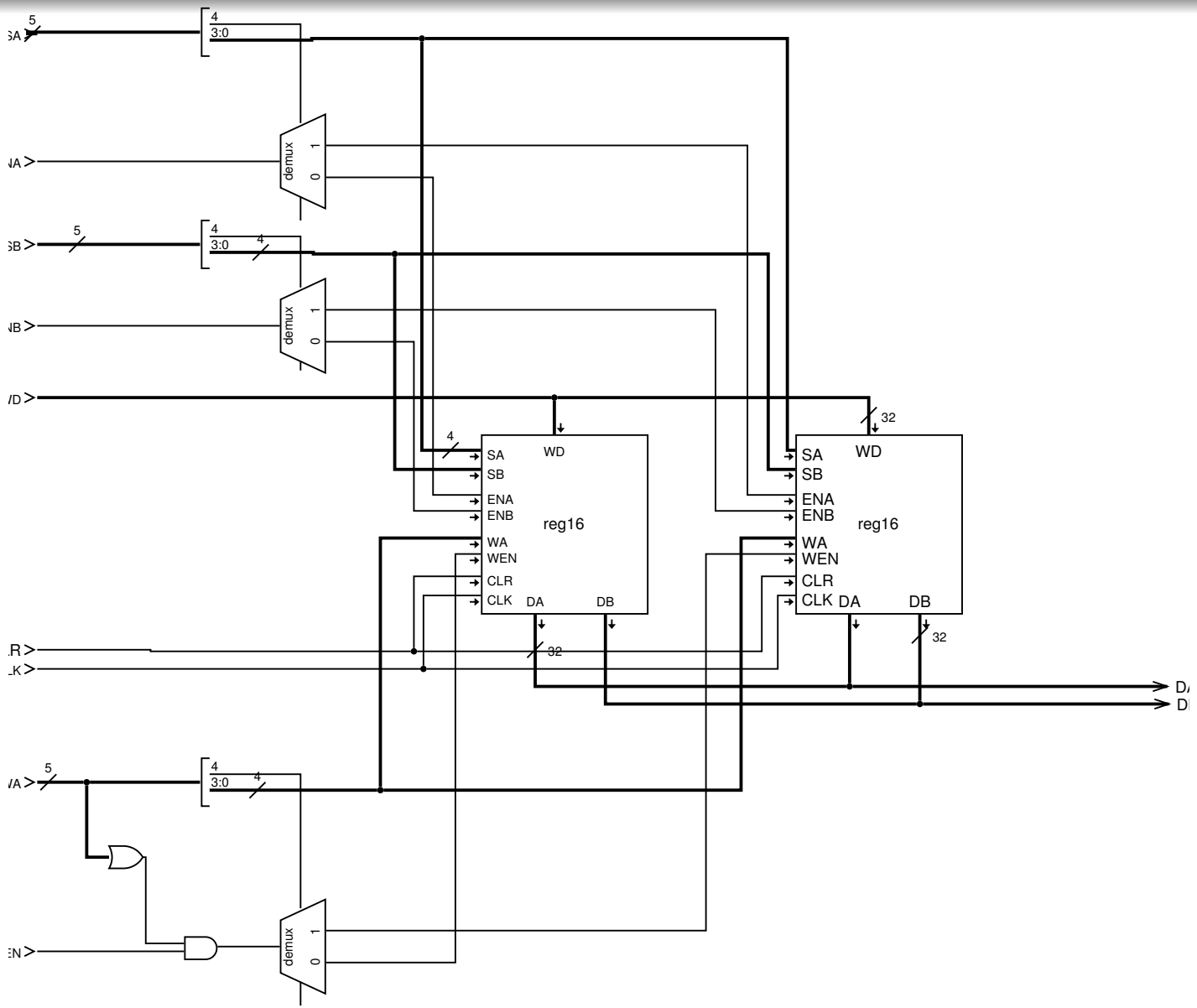
Instruction Memory Level 1



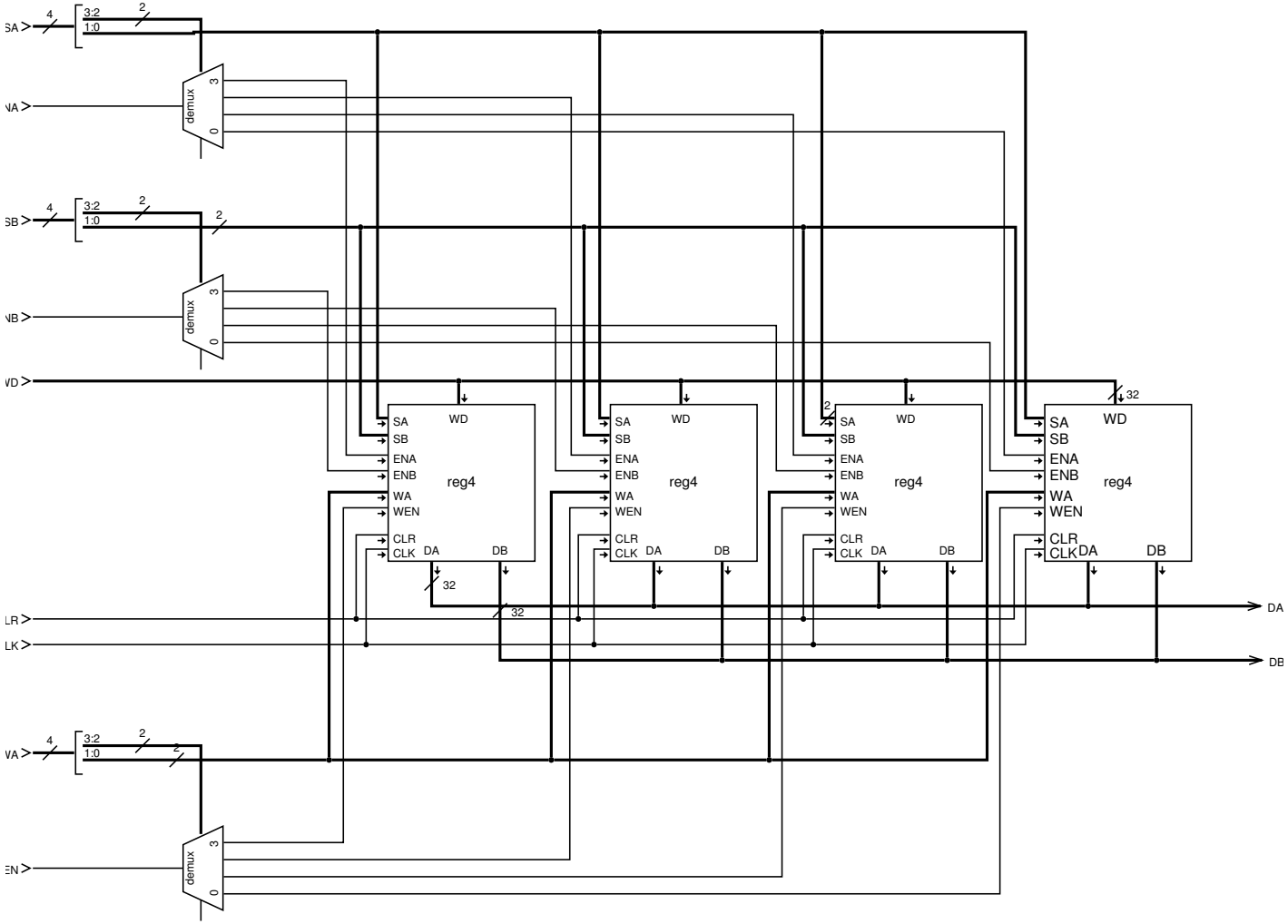
Register File Interface



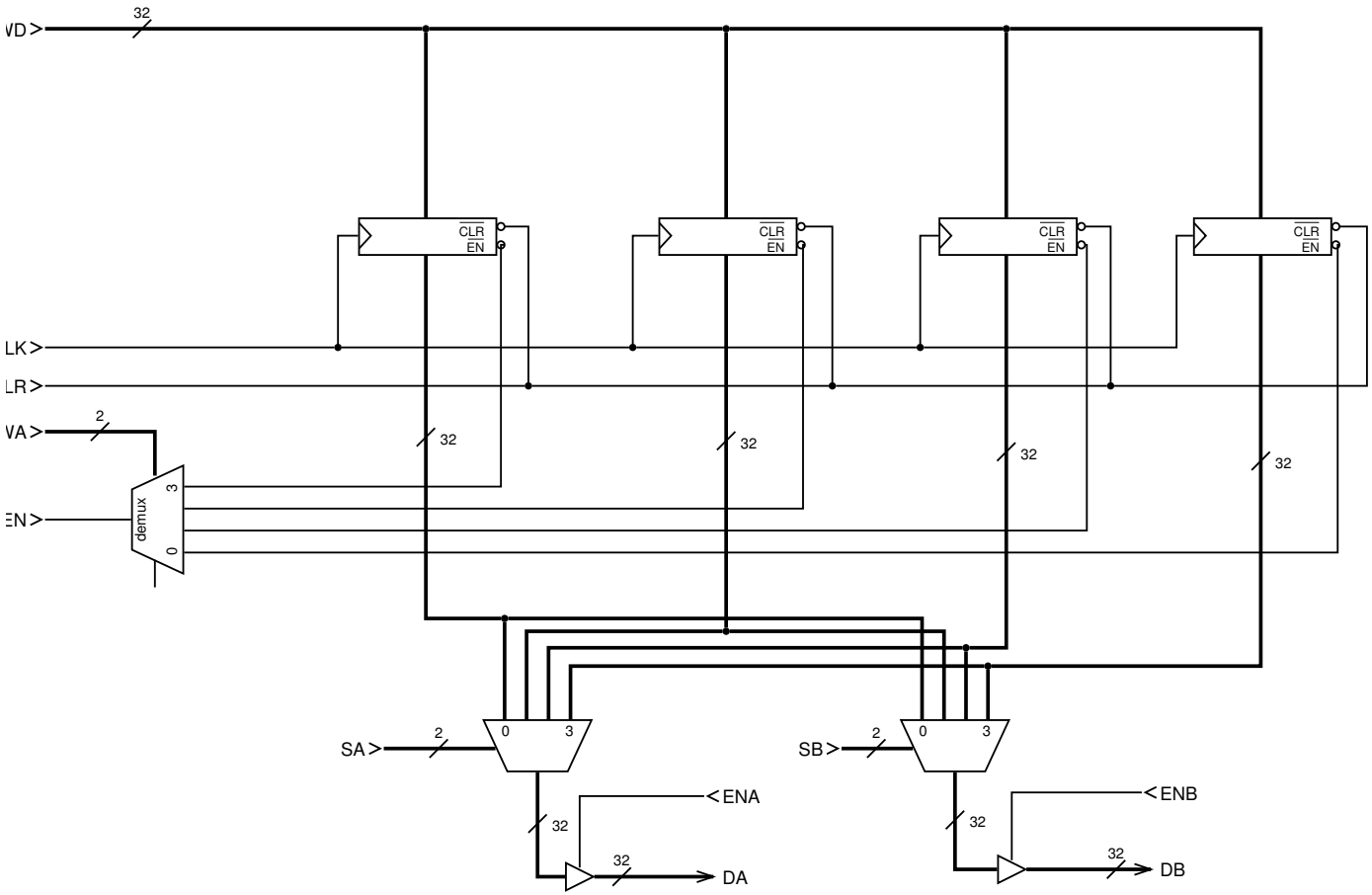
Register File Level 1



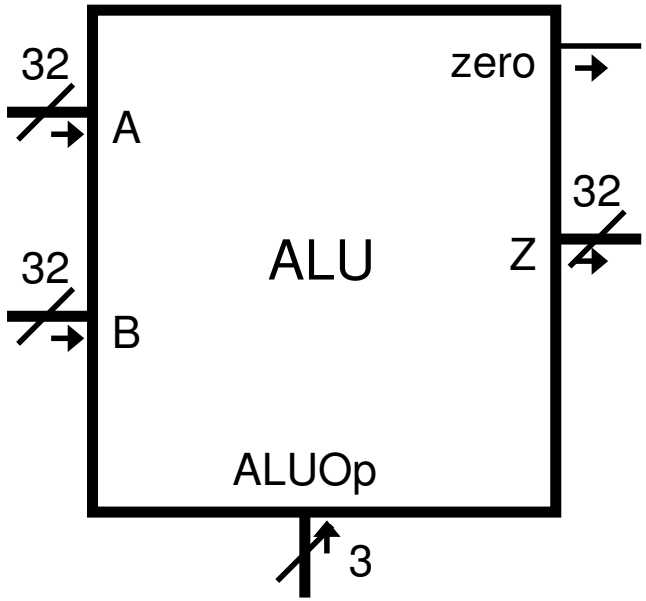
Register File Level 2 (reg16)



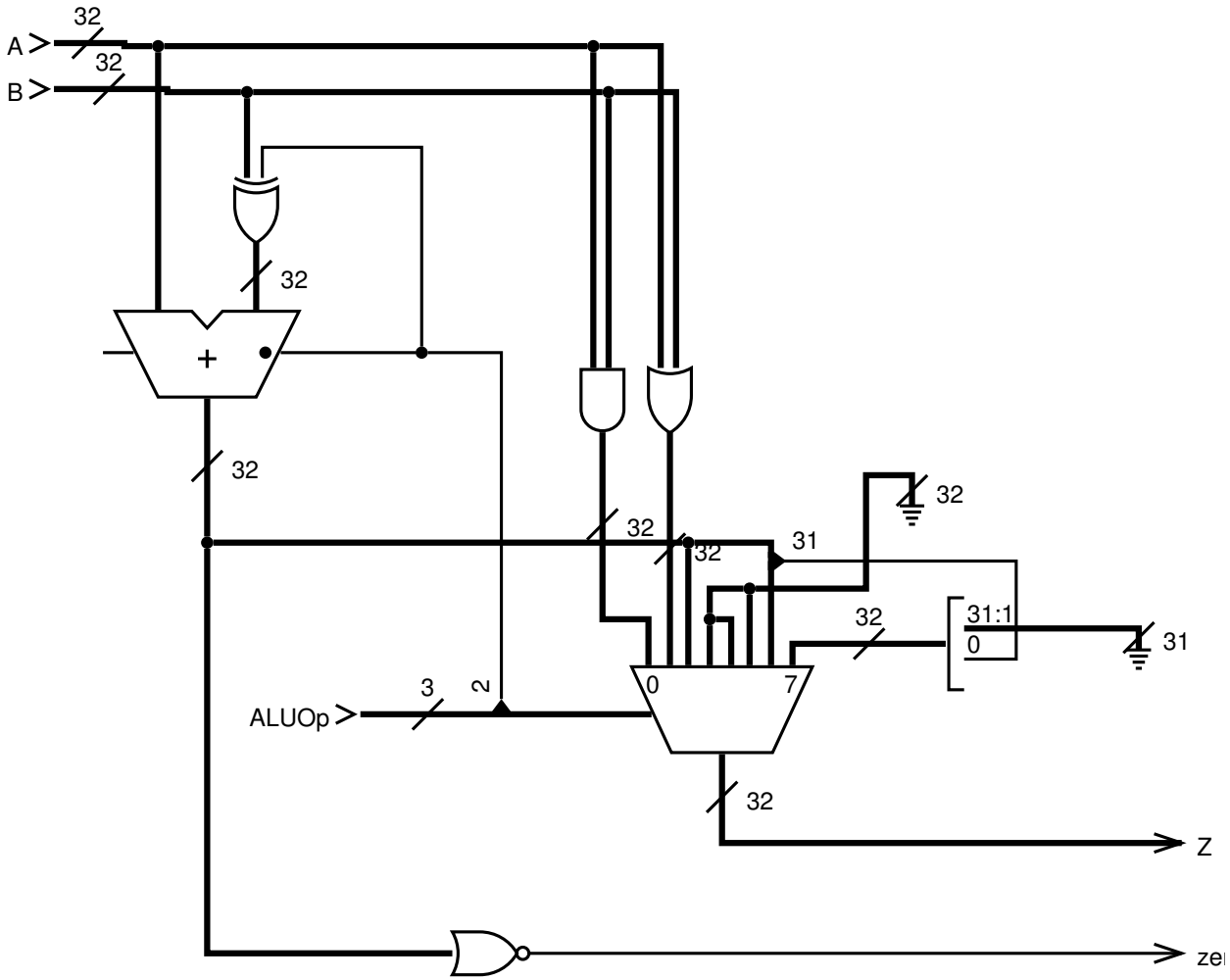
Register File Level 3 (reg4)



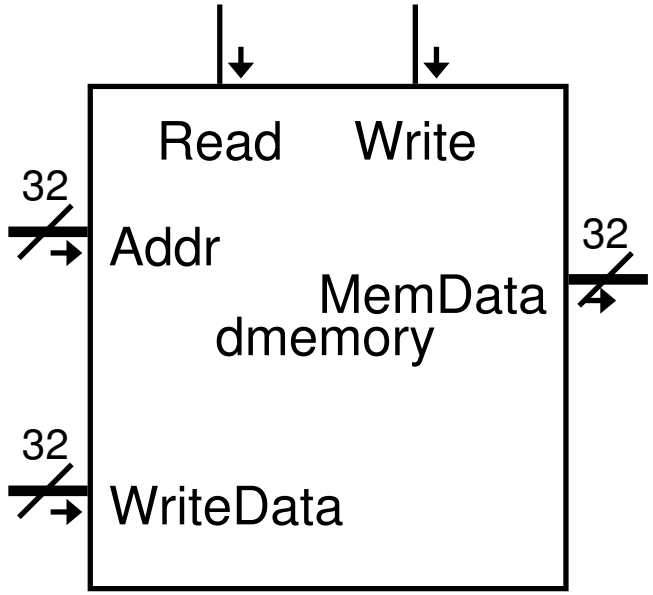
ALU Interface



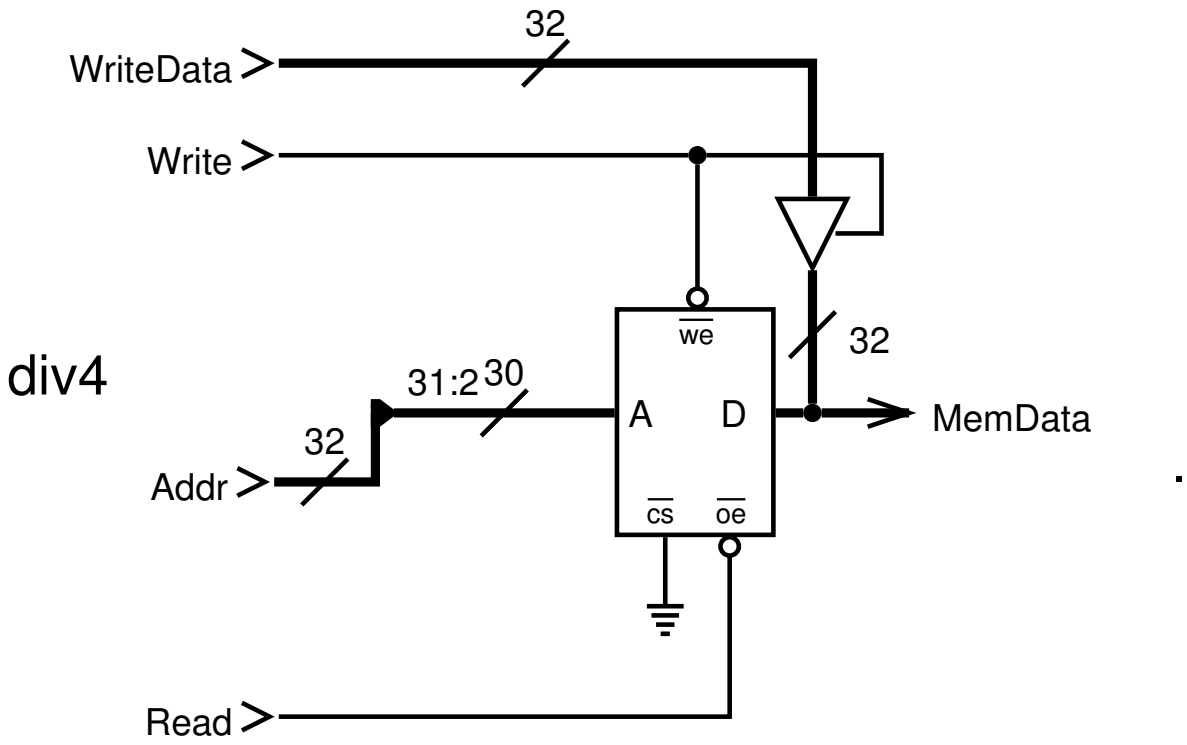
ALU Level 1



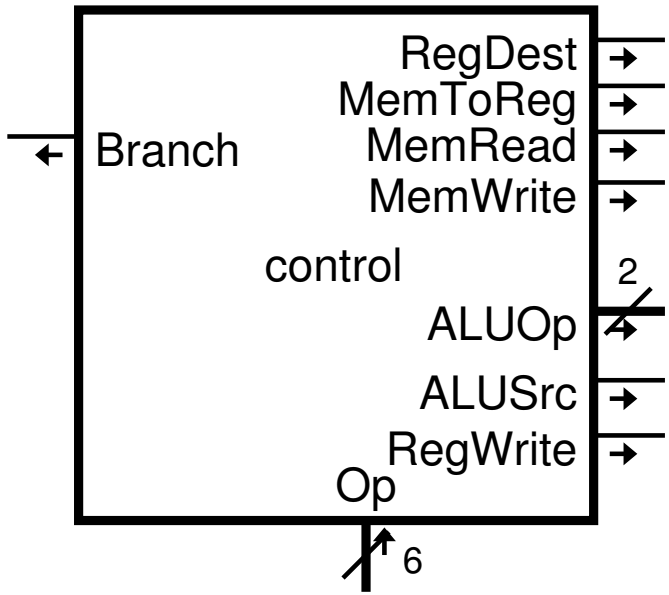
Data Memory Interface



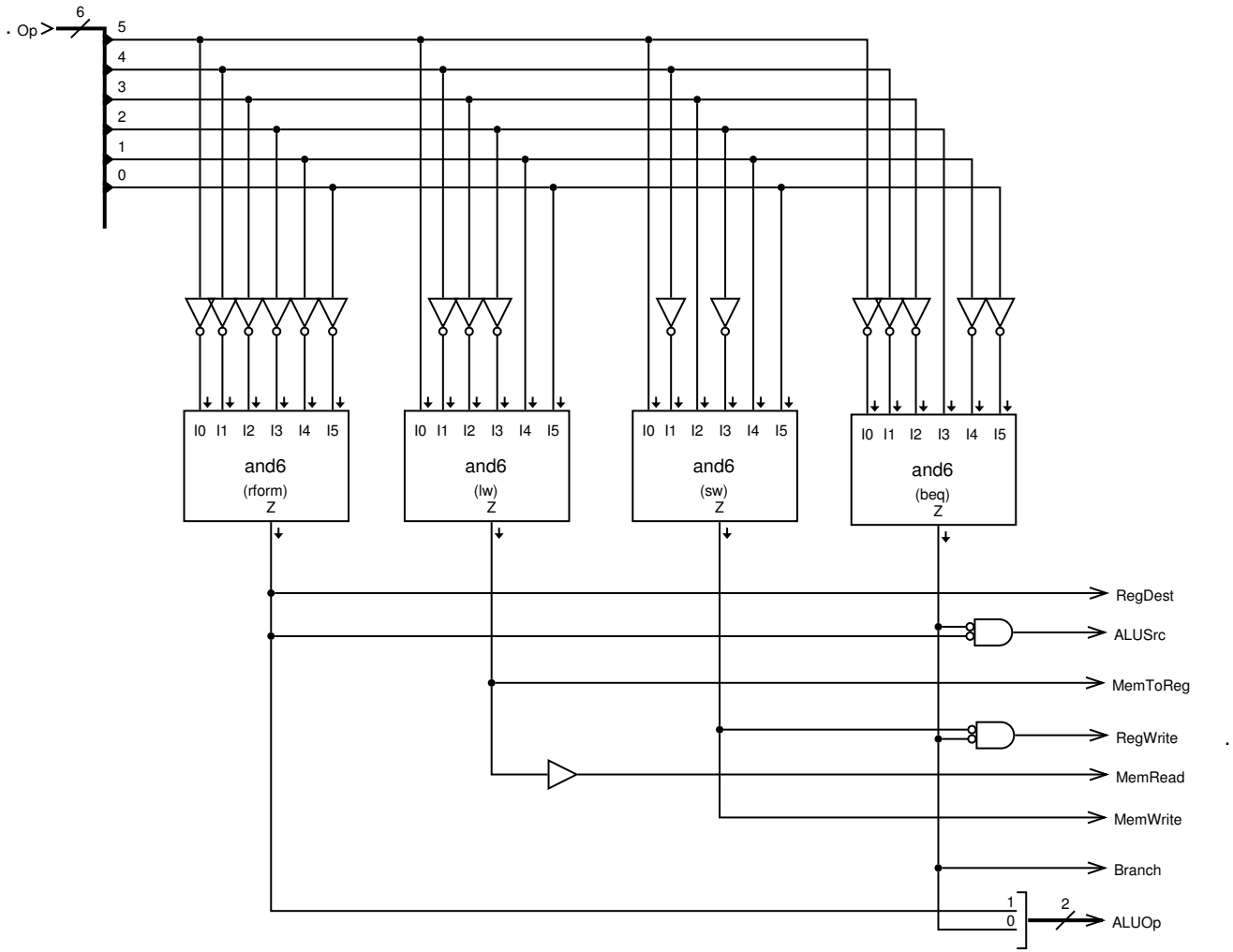
Data Memory Level 1



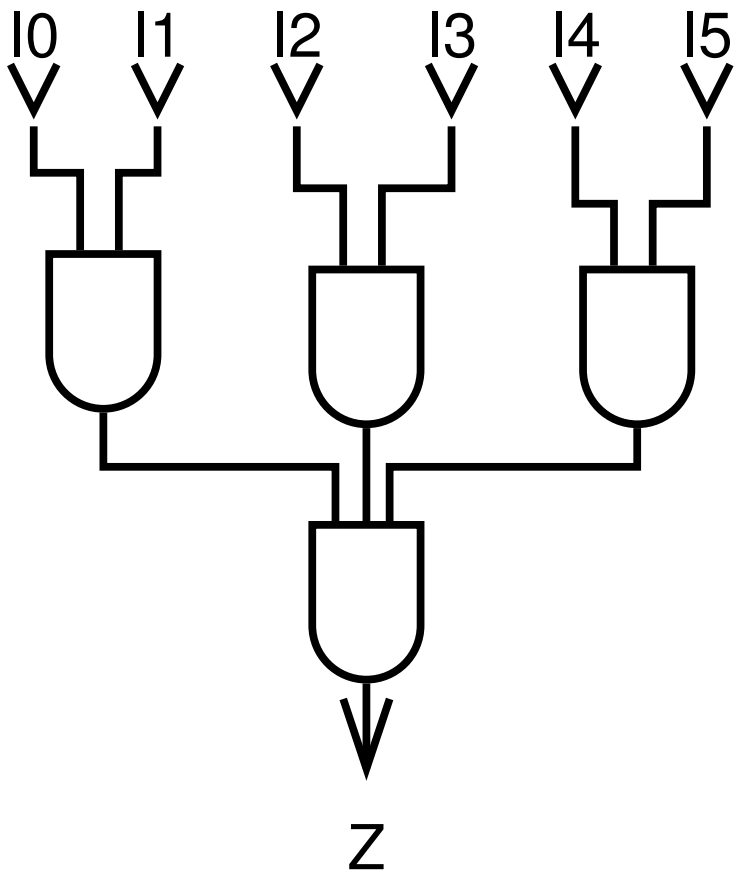
Control Unit Interface



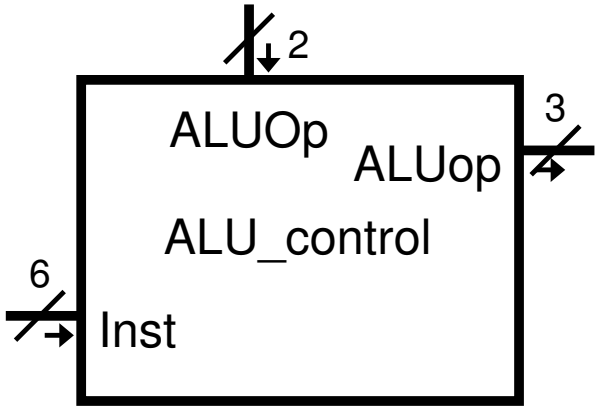
Control Unit Level 1



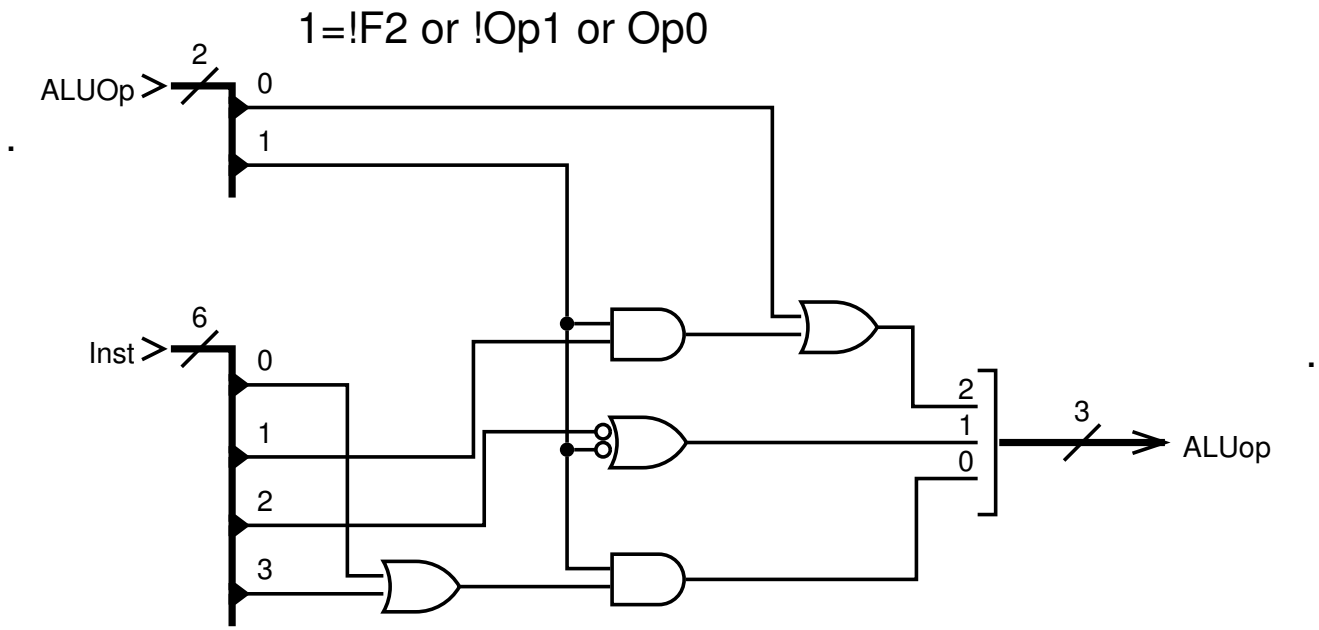
Control Unit Level 2



ALU Control Unit Interface



ALU Control Unit Level 1



Week 4 Activity 1

Consider the following logic delay:

Temp. \ L.E.	Gate	Register
	Propagation Delay	
85 °C	.9ns	.9ns
50 °C	.7ns	.9ns
10 °C	.6ns	.7ns

1. For each temperature profile, calculate the longest and smallest combinatorial path delay in the Simple CPU described in this lecture.
   - Hints: The longest combinatorial path has the maximum number logic gates from input to output. Smallest is the minimum number of gates from input to output. Total delay $\approx$ sum of all module delays

Week 4 Activity 2

SignalTap II and Memory Editor: The simple CPU will uses Quartus II SignalTap II and Memory Editor software features. SignalTap II is an in-logic analyser used to debug designs implemented in Altera FPGAs. In this activity, you will have an opportunity to get introduce to these features and get accustomed to their debugging capabilities. For this activity you would be using mips_basic.qar project archive.

You will using three Quartus tools in this activity:

1. Custom GUI tcl script (gui_front_2.tcl) and "quartus_stp"
2. Quartus II - In-System Memory Content Editor
3. Quartus II - Signal Tap II

Week 4 Activity 2

1. **Memory Initialization File MIF**:
   - What is a .mif file and how is it used?
   - Describe .mif file format that is used by Altera's Quartus II software.
   - Using "In-System Memory Content Editor" write the following into memories and submit a screen capture for each.
     1. Write "CS6133-your name" into IRAM. See video for demonstration
     2. Write "CS6133-your last name" into DRAM. See video for demonstration

Week 4 Activity 2

2. **SignalTap II and gui_front_2.tcl**:
   - Launch SignalTap II by typing "quartus_stpw mips_basic.stp &" in the command line from the project directory. (See demonstration)
   - Launch gui_front_2.tcl by typing "quartus_stp -t gui_front_2.tcl" in the command line from the project directory
   - Read SignalTap II documentation [1] and submit procedures to perform the following tasks
     1. Testing ALU's functionalities using gui_front_2.tcl and SignalTap II
     2. Testing Program Counter's functionalities using gui_front_2.tcl and SignalTap II
     3. Reading the contents on instruction and data memory using SignalTap II.
     4. Reading the contents of register file using SignalTap II

[1] http://www.altera.com/literature/hb/qts/qts_qii53009.pdf

Week 4 Activity 2

3. Answer the following questions:
   1. Explain the purpose of strb_module in mips_basic.
   2. When and why does "led0" connected to ALU light up?
   3. The clock wire in the schematic seems to be disconnected, how is this working?
   4. What is purpose of "rst_out" signal?
   5. What is the clock frequency used in mips_basic?
   6. Is mips_basic ALU same as the ALU in this lecture? If not, why? support your answer with details.

Week 4 Activity 3

Consider the Simple CPU blocks, presented in this lecture, and Dr. Neumann's digital computer.

1. Is it possible to use a RAM/ROM for control unit? Justify your answer with details.
2. Compare the control units (both the main control unit and ALU control unit) with Dr. Neumann's CC and answer the following questions:
   1. List their similarities and differences
   2. Does Simple CPU's control units (Main and ALU control unit) confirm to CC's requirements? Justify your answer with details.