Department of Electrical and Computer Engineering
The University of Texas at Austin

ECE 460N/382N.1 Fall 2024
Instructor: Yale N. Patt
TAs: Anna Guo, Nadia Houston, Logan Liberty, Luke Mason, Abhay Mittal, Asher Nederveld, Edgar Turcotte
Final Exam
December 13, 2024

Name: ____________________________________________

**PART A**

Problem 1 (10 points): _______

Problem 2 (10 points): _______

Problem 3 (10 points): _______

Problem 4 (15 points): _______

**Total Part A (45 points): ________**

**PART B**

Problem 5 (30 points): _______

Problem 6 (30 points): _______

Problem 7 (30 points): _______

**Total Part B (90 points): ________**

Note: Please be sure that your answers to all questions (and all supporting work that is required) are contained in the space provided.

Note: Please be sure your name is recorded on each sheet of the exam.

Please read the following sentence, and if you agree, sign where requested:
I have not given nor received any unauthorized help on this exam.

Signature: ____________________________________________

**GOOD LUCK!**

Name: ________________________________

**Problem 1 (10 points): Tomasulo**

An out-of-order processor executes the following program segment using Tomasulo's original algorithm (no ROB).

ADD R0, R0, R0

ADD R1, R2, R3

MUL R4, R3, R5

MUL R6, R0, R1

ADD R7, R4, R6

There are four stages: Fetch, Decode, Execute, and Writeback.

- Fetch, Decode, and Writeback take one clock cycle each.
- Fetch, Decode, and Writeback can only operate on one instruction at a time.
- Instructions with no dependencies can start executing immediately after Decode.
- There are two functional units:
  - A pipelined adder that takes 3 cycles.
  - A pipelined multiplier that takes 5 cycles.
- Entries are put into Reservation Stations at the end of Decode and removed at the end of Writeback. There are 8 unified Reservation Stations.
- The results of an instruction are broadcast at the end of Writeback and a dependent instruction can begin execution in the next cycle.
- The Writeback bus supports only one result being stored at a time. Earlier instructions take priority for Writeback.

**Your job:** Fill in the timing diagram below for the code above. Use **F** for Fetch, **D** for Decode, **A** for Add, **M** for Multiply, and **WB** for Writeback. Use - to indicate waiting in a Reservation Station, and * to indicate a stall that clock cycle.

| | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **I1** | F | | | | | | | | | | | | | | | | | | |
| **I2** | | F | | | | | | | | | | | | | | | | | |
| **I3** | | | F | | | | | | | | | | | | | | | | |
| **I4** | | | | F | | | | | | | | | | | | | | | |
| **I5** | | | | | F | | | | | | | | | | | | | | |

Name: ________________________________

**Problem 2 (10 points): Floating Point**

**Part a (3 points):** Consider an 8-bit IEEE-like floating point representation. In this format, 00100101 represents the value 21/32 exactly. How many bits are used for the fraction and exponent, and what is the bias?

| # of Fraction bits: | # of Exponent bits: | Bias: |
| --- | --- | --- |
| | | |

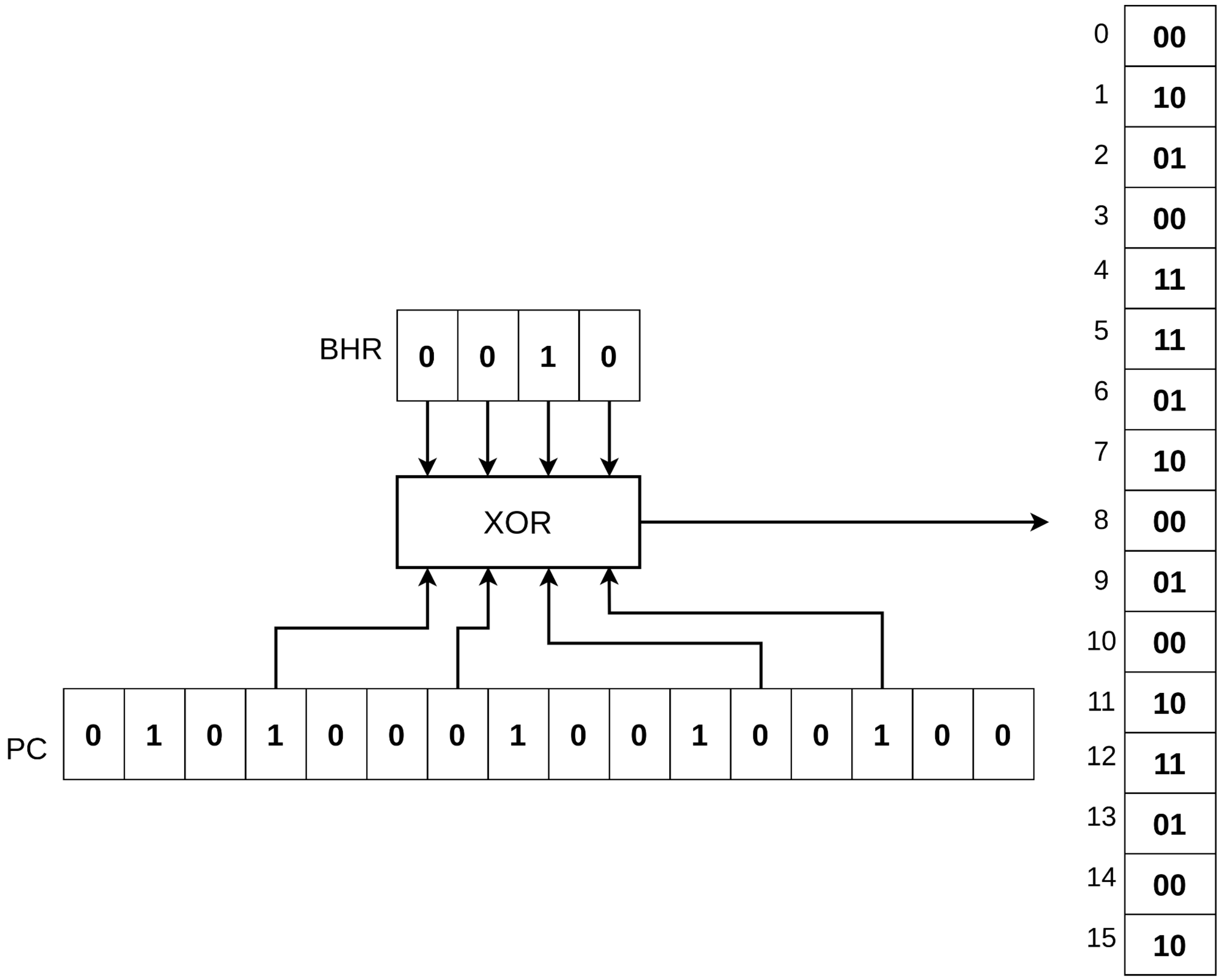
**Part b (4 points):** Now add that number, 00100101, to the floating point number represented by 00011101, using an always round-up scheme. What is the result? (Leave the answer as a fraction)

**Part c (3 points):** In class, we discussed another rounding mode, "unbiased nearest rounding". What makes "unbiased nearest rounding" unbiased?

Name: ________________________________

**Problem 3 (10 points): Branch Prediction**

Below you are given a 2-level branch predictor with the g-share modification.




**Part a (5 points):** Given the values in the schematic above, would the predictor predict taken or not taken? Briefly explain.

**Part b (5 points):** What is the prediction without the g-share modification? (using a GAg predictor) Briefly explain.

Name: ________________________________

**Problem 4 (15 Points): Virtual Memory**

Consider the LC-3b as implemented in Lab 5. Recall the following details:

- The LC-3b uses a **1-level** page table for each access.
- VA contains 7 bits for VPN and 9 bits for offset. There is no region bit.
- PTE contains 5 bits for PFN as well as P, V, M, and R bits. The rest of the bits are 0s.

**Part a (5 points):** You want to design a TLB (Translation Lookaside Buffer) for the LC-3b. The TLB has 8 entries and a random replacement policy. The TLB is fully associative. How many total storage bits do you need to implement the TLB if you don't include any unused zero bits in the PTE? Show your work.

| |
|---|
| |

**Part b (2 points):** In fewer than 15 words each, give one benefit of 1-level virtual memory and one benefit of 2-level virtual memory.

| |
|---|
| One level: |
| Two level: |

**PROBLEM CONTINUES ON THE NEXT PAGE**

Name: ________________________________

**Part c (8 points):** The following program runs on the LC-3b with a correctly implemented lab5. You are also given the data at a few memory locations before execution starts. The PBR contains the value x1000. Remember, the PTE format is as follows:

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | PFN | | | | | 0 | 0 | 0 | 0 | 0 | P | V | M | R |

| **Memory locations before execution start** | |
|---|---|
| **Addr** | **Data** |
| x102E | x220C |
| x103E | x3004 |
| x1040 | x3006 |
| x1042 | x3002 |
| x1044 | x3001 |
| x1080 | x3806 |
| x1082 | x3C04 |
| x1084 | x3804 |

**Program**

```
            .ORIG x4000
            LEA R1, ADDRESS
            LDW R2, R1, #0
            LDW R1, R1, #1
LOOP        STB R2, R2, #0
            ADD R2, R2, #1
            ADD R1, R1, #-1
            BRNP LOOP
            HALT
ADDRESS     .FILL x8000
ITER        .FILL x0300
            .END
```

Fill in all VPNs accessed and their corresponding PFNs in the order they are first accessed. The first entry has been filled in for you. You need not use all the boxes.

| **VPN** | **PFN** |
|---|---|
| 0x20 | 0x18 |
| | |
| | |
| | |
| | |

Name: ________________________________

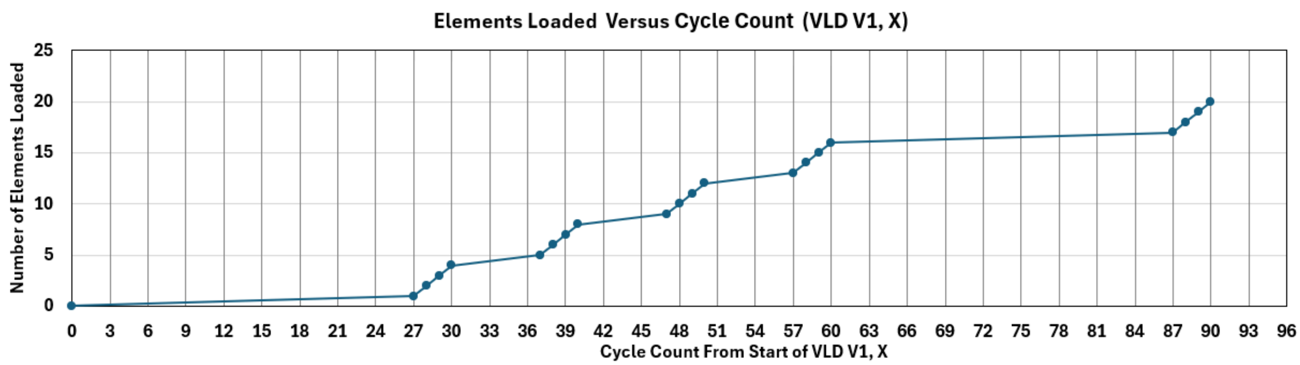
**Problem 5 (30 points)**: **Caches and Vectors**

A computer contains a vector unit, and write-back L1 and L2 data caches. An engineer executes the following program, where X, Y, and Z are the starting addresses of three one-dimensional arrays in memory. Each array starts at the **beginning of a cache line (in both caches)**.

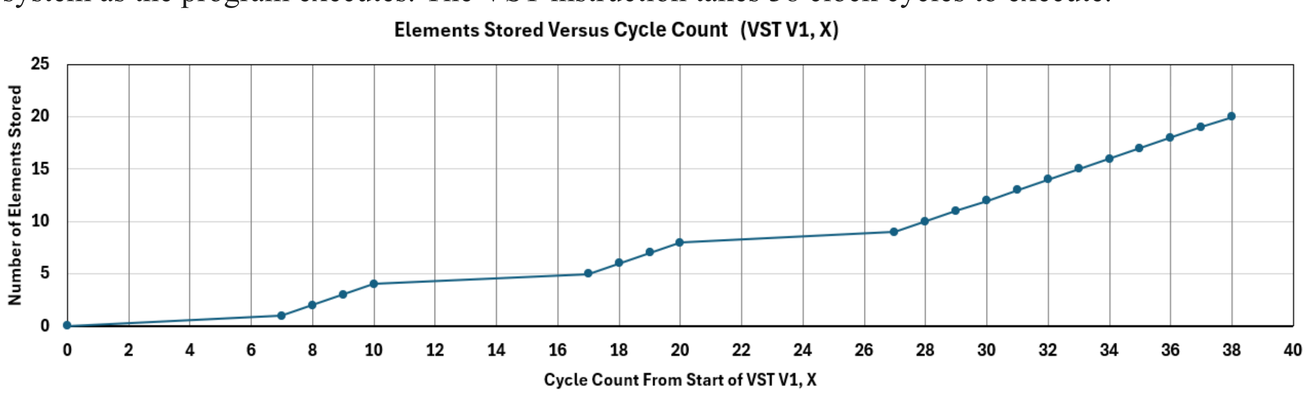
```
LVL 20              ; set vector length
LVS 1               ; set vector stride (in elements)
VLD V1, X           ; vector load
VLD V2, Y
VLD V3, Z
VMUL V2, V3, V2     ; vector multiply
VADD V1, V1, V2     ; vector add
VST V1, X           ; vector store
```

- Memory is byte-addressable. Each vector register holds up to 64 elements of 32 bits each.
- Memory accesses work as follows: first, the L1 data cache is checked. If it misses, the processor then checks the L2 data cache. On an L2 data cache miss, the processor accesses DRAM. Data is filled into both the caches as part of the DRAM access.
- There is one port to the memory hierarchy. DRAM accesses have a fixed latency.
- Assume there are no page faults. All caches are initially empty.

The following diagram shows the cycle in which each component of V1 is loaded. For example, the 3rd component of V1 is loaded in clock cycle 29. Note that 90 cycles are required to load V1.




The diagram below shows the clock cycle in which each element of V1 is stored to the memory system as the program executes. The VST instruction takes 38 clock cycles to execute.




**PROBLEM CONTINUES ON THE NEXT PAGE**

Name: ______________________________

**Part a (4 points):** What is the size of a cache line for the L1 and L2 caches in bytes?

**L1:** [ ] **L2:** [ ]

**Part b (6 points):** How many clock cycles does it take to access the L1 cache, the L2 cache, and DRAM?

| L1 Cache | L2 Cache | DRAM |
|---|---|---|
| | | |

**Part c (8 points).** The L2 cache is fully associative with LRU replacement. What is the lower bound for the size of the L2 cache? Explain

**Part d (6 points):** How many clock cycles would this program take to execute without vector chaining? For this problem, LVL and LVS each take 1 cycle to complete. VADD takes 6 cycles to complete per element, and VMUL takes 8 cycles to complete per element. Both the Adder and Multiplier are pipelined. The cycle counts for VLD and VST depend on whether they hit in the L1 cache, L2 cache, or DRAM.

**Part e (6 points):** If vector chaining is implemented, how many clock cycles would this program take to execute?

Name: ________________________________

**Problem 6 (30 points): Datapath**

The top graduating student at Texas A&M created an addition to the LC-3b ISA. Unfortunately, he did not include documentation for his instruction, and you don't know its name or function. Luckily, you have his instruction encoding and most of his design, which you will complete in this problem.

Below is the encoding of the mystery instruction:

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | SR | 1 | 0 | Length |

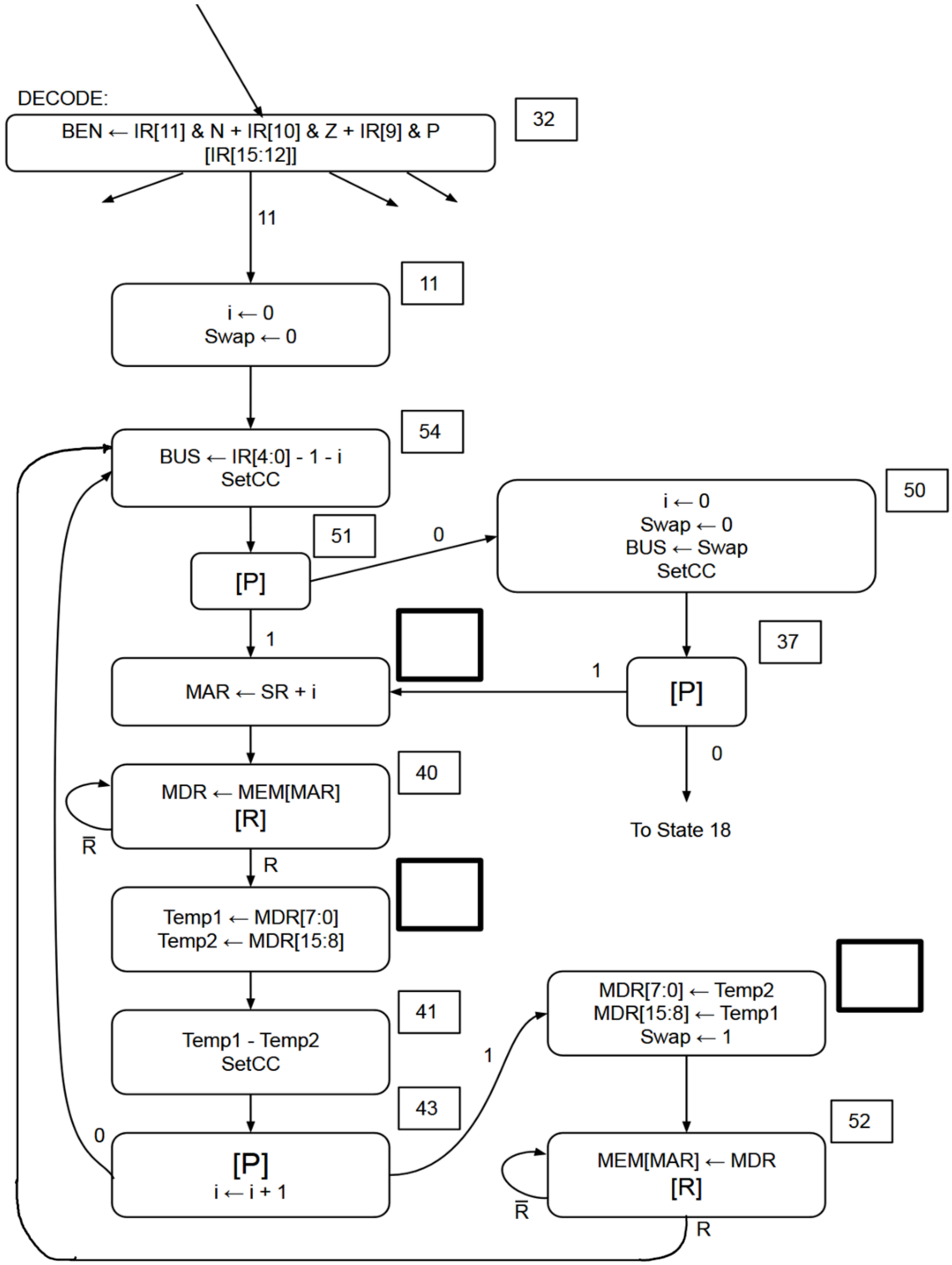
SR contains the starting address of an **array of bytes**. Length refers to the number of elements in the array.

**Part a (9 points):** Examine the state machine on the next page. What function does this new instruction perform? Explain in 15 words or fewer. Hint: try using an example array.

**PROBLEM CONTINUES ON THE NEXT PAGE**

Name: ________________________________

**Part b (9 points):** Fill out the three missing state numbers in the bold boxes.



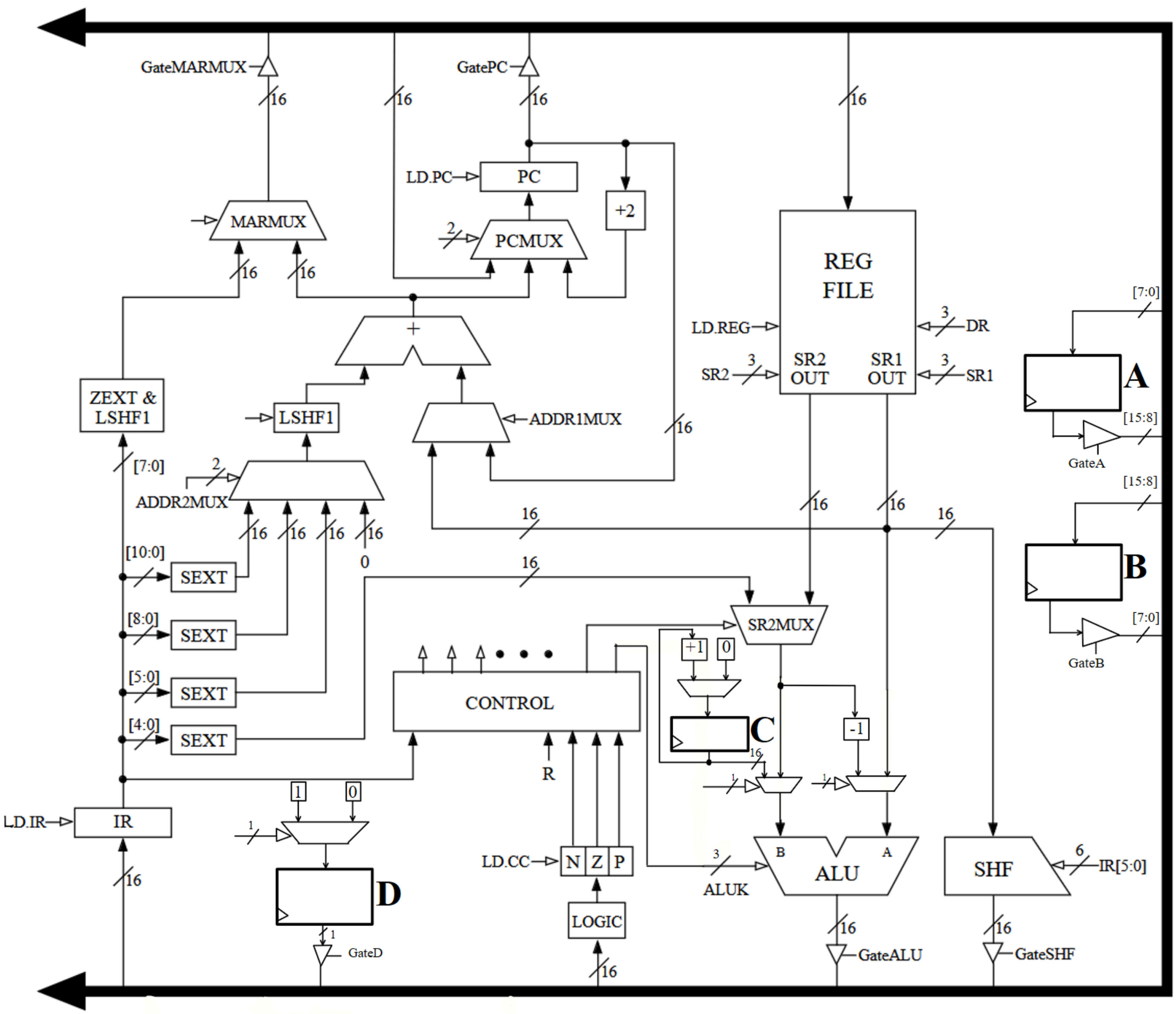

**PROBLEM CONTINUES ON THE NEXT PAGE**

Name: ________________________________

**Part c (6 points):** The datapath has been augmented with registers **TEMP1**, **TEMP2**, **SWAP**, and **i**. Fill in these names in the corresponding boxes labeled A, B, C, and D.

In addition to the other datapath changes you see:

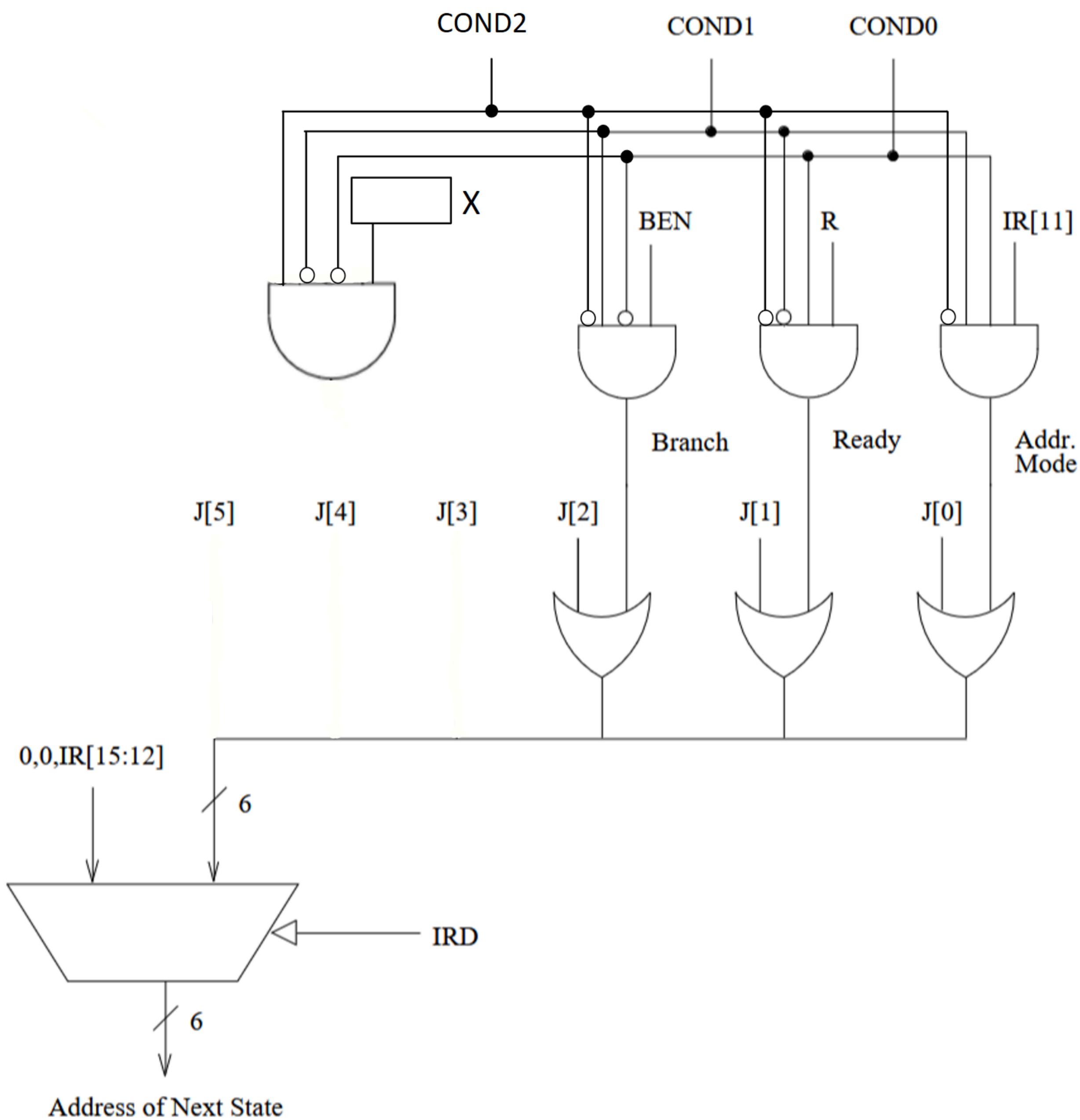
- The ALU has been modified to support the ability to subtract in the format of A - B.
- Memory has been modified to support unaligned accesses.




**PROBLEM CONTINUES ON THE NEXT PAGE**

Name: ___________________________________

**Part d (6 points):** This new instruction involved changes to the microsequencer. A new control signal, COND2, has been added to the microsequencer. Complete all missing wires and determine the signal X.



Name: ______________________________

**Problem 7 (30 points)**: **Physical Memory**

A student writes the following code to transpose an N by N matrix A and stores the transposed matrix into B. Each element is 64-bit.

```
for (int i = 0; i < N; i=i+1) {
  for (int j = 0; j < N; j=j+1) {
    int64_t temp = A[i*N+j];
    // note: the following store cannot begin executing until the
    // previous load returns (data won't be in temp otherwise)
    B[j*N+i] = temp;
  }
}
```

You may assume that local variables, such as i, j, and temp, are stored in registers and all matrix accesses go to memory. Matrix A starts at x4000, and matrix B starts at x8000. Suppose we have DRAM memory with the following address scheme:

| 15    13 | 12    4 | 3    2 | 1    0 |
|---|---|---|---|
| Row | Column | Bank | BoB |

- The processor is byte-addressable
- Due to the bus width, it takes 2 memory accesses to load an element from the matrix
- Only one request can be sent to DRAM memory per clock cycle
- Loads and stores are sent to the DRAM in order, but the data may return out of order
- All row buffers are initially empty
- It takes 60 clock cycles to open a row buffer
- It takes 30 clock cycles to get data from a row buffer hit

A bank conflict occurs **only** when the oldest request cannot be sent to DRAM because the bank to be accessed is busy dealing with another request.

**PROBLEM CONTINUES ON THE NEXT PAGE**

Name: ______________________________

**Part a (12 points):** How many bank conflicts occur when transposing a 4x4 matrix? Show your work

How does this change if we increase the number of banks to 8 by making **bit 4 a bank bit**? Explain

**Part b (18 points):** How many bank conflicts occur when transposing a 16x16 matrix? Show your work

How does this change if we increase the number of banks to 8 by making **bit 13 a bank bit**? Explain

Name: ________________________________

**THIS PAGE IS LEFT INTENTIONALLY BLANK FOR SCRATCH WORK**