Practice problems

For Exam 1: HW

Exam 1: Computational Building Blocks (HW) + Memory

Lectures
- Digital Logic
- Data as Bits
- Integer Representation
- Combinational Logic
- Arithmetic Logic
- Sequential Logic
- A Simple Processor
- Programming with memory

Labs
- 1: Transistors to Gates
- 2: Data as Bits
- 3: Combinational Logic & Arithmetic
- 4: ALU & Sequential Logic
- 5: Processor Datapath
- 6: Pointers in C

Topics
- Transistors, digital logic gates
- Data representation with bits, bit-level computation
- Number representations, arithmetic
- Combinational and arithmetic logic
- Sequential (stateful) logic
- Computer processor architecture overview
- Basic C programming with pointers

Assignments
- Gates
- Zero
- Bits
- Arch
- Pointers

Mid-semester Exam 1: HW
March 9

Short answer practice problems:

ex

1. How does a D-latch differ in behavior from a D-flip-flop?

2. How are instructions stored in the HW ISA? How does the HW ISA processor know what instruction to execute next?

3. How many bits are needed to choose a register if the register file has 32 entries?

4. What does it mean for a gate to be universal?

Bit manipulation practice problem

```
/*
 * absVal - Return the absolute value of x
 *    Examples: absVal(-1) = 1
 *              absVal(240) = 240
 *    You may assume -TMax <= x <= TMax
 *    Legal ops: ! ~ & ^ | + << >>
 */
int absVal(int x) {



}
```

Bit addition practice problem

ex

What is the result of the following computation on 8-bit two's complement numbers?

$$\texttt{0b110100101 + 0b011001111}$$

Does it overflow? Justify your answer without converting to binary numbers.

Consider the same computation on unsigned numbers. What is the result? Does it overflow?
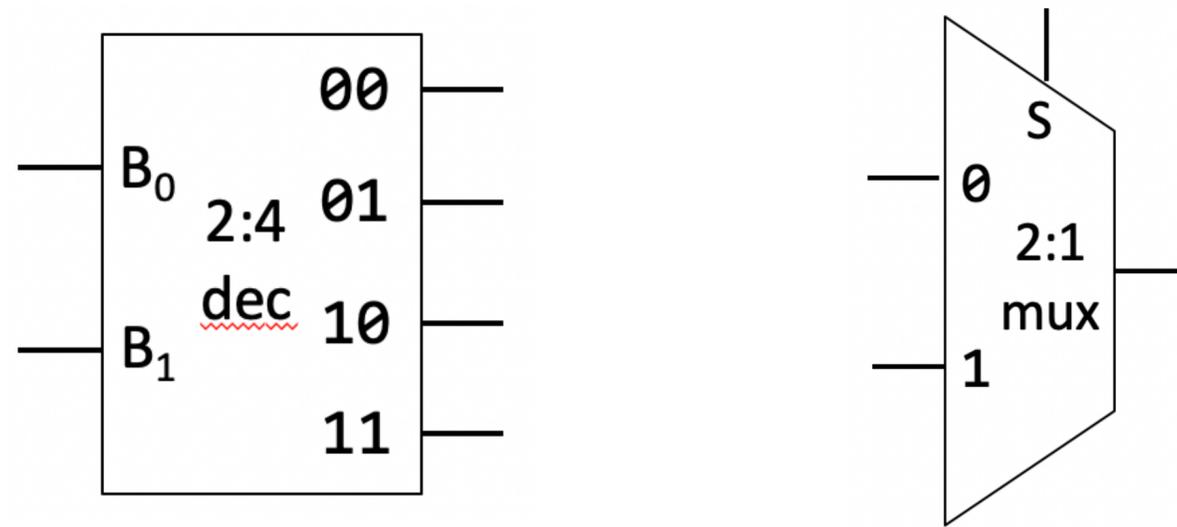
Building block choice practice problem

Draw a circuit to implement a switching network. If S=1, the network is in pass-through mode: C=A and D=B . If S=0 , the network is in crossing mode: C=B , and D=A .

Use the most reasonable combinational building blocks or gates.

A ——   —— C

B ——   —— D

S ——

Decoder + mux practice problem
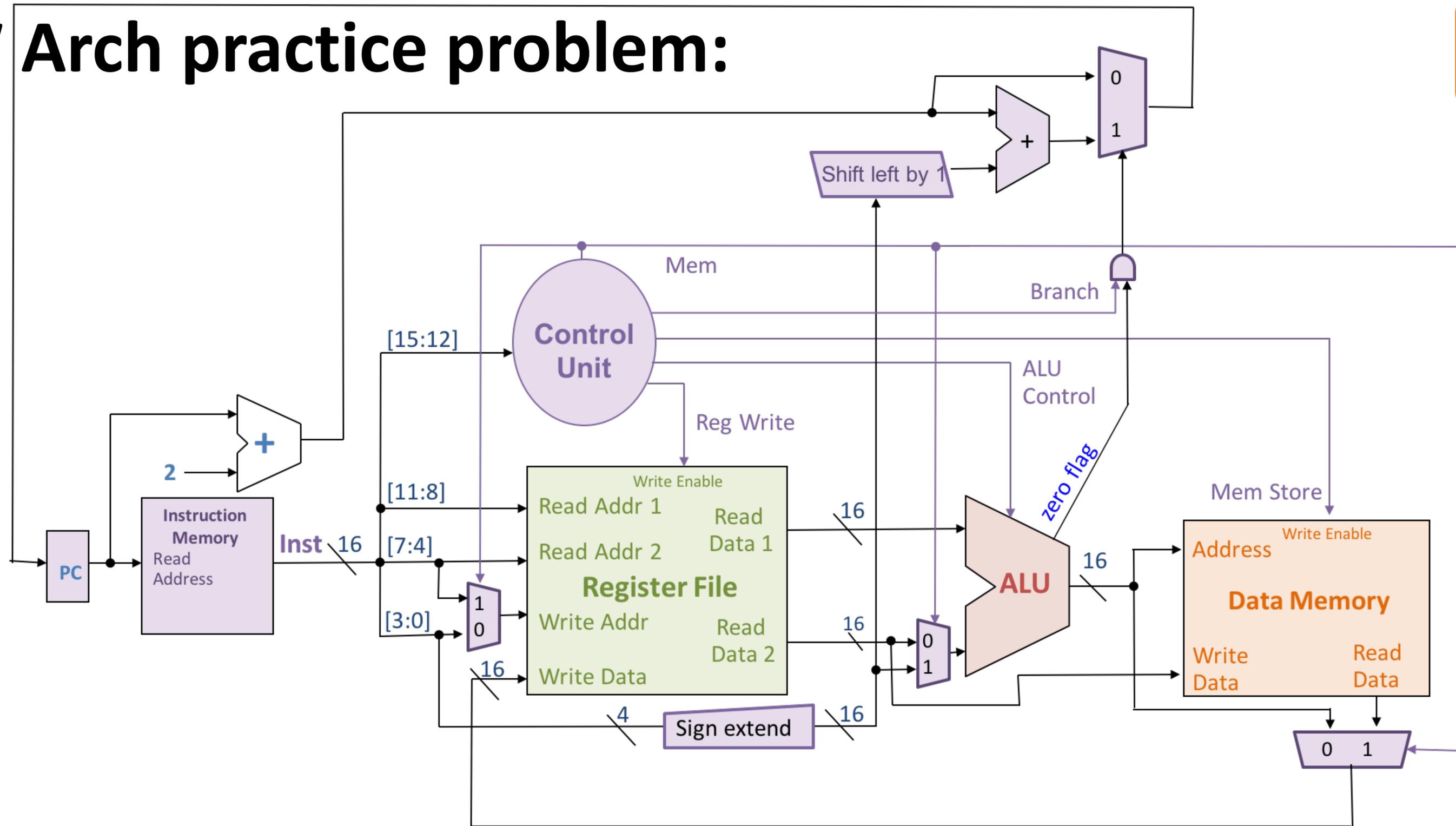
Use **one** 2:4 decoder and **one** 2:1 mux to implement A XOR B

HW Arch practice problem:

1. What is the purpose of each of the 4 mux components in the HW architecture above?

C programming with pointers

Write the following C function:

`double_string` should take in a null-terminated C string and return a newly-allocated null-terminated string that contains the same string, repeated twice in sequence. You do not need to check the result of `malloc`. You may use the `strlen` function.

For example, `double_string("abcd")` = "abcdabcd".