Practice problems

For Exam 1: HW

Bit addition practice problem

ex

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

```
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?

Bit addition practice problem: solution

ex

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

$$\begin{array}{r} {\scriptstyle 1\ 1\quad\quad 1\ 1\ 1\ 1} \\ 110100101 \\ +\ 011001111 \\ \hline 001110100 \end{array}$$

Does not overflow. Justification:

- Inputs have different sign bits (overflow when sign bits are the same and output sign bit differs)
- OR: Carry in and carry out of the most significant bit are the same

Unsigned: same sum (001110100).

- Overflows because carry out of the most significant bit is dropped.

Building block choice practice problem

ex

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 .

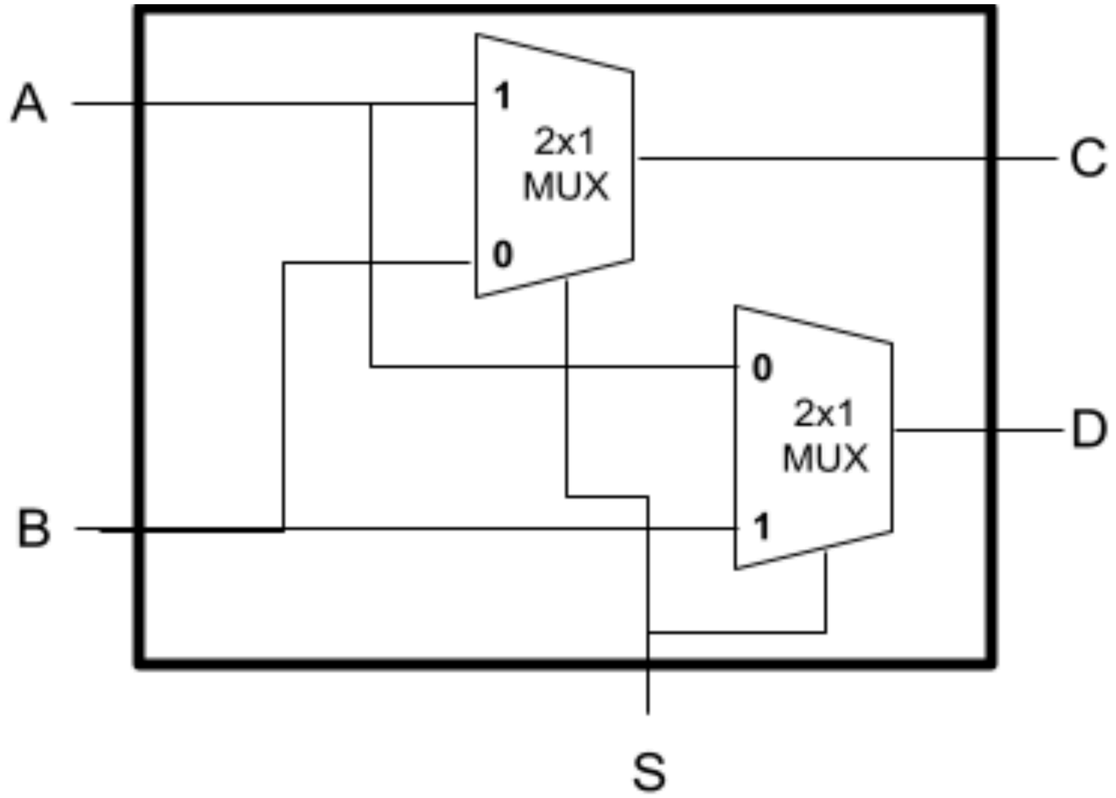
Use the most reasonable combinational building blocks or gates.

Building block choice practice problem: solution

ex

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.



Bit manipulation practice problem

ex

```
/*
 * 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 manipulation practice problem: solution 1

ex

```
/*
 * absVal - Return the absolute value of x
 *   Examples: absVal(-1) = 1
 *             absVal(240) = 240
 *   You may assume -TMax <= x <= TMax
 *   Legal ops: ! ~ & ^ | + << >>
 *   Max ops: 8
 *   Rating: 4
 */
int absVal(int x) {
    int mask = x >> 31; // All 0's if positive, all 1's if negative
    return (x & ~mask) | ((~x + 1) & mask);
}
```

Bit manipulation practice problem: solution 2

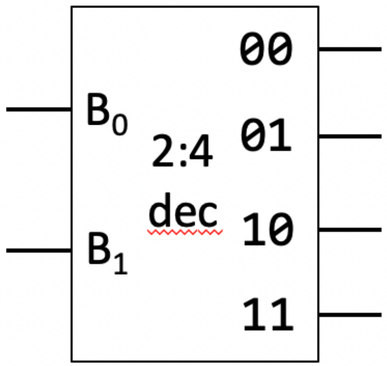
ex

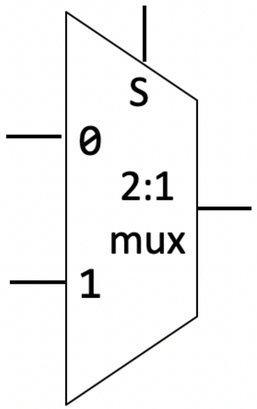
```
int absVal(int x) {
    // All 0's if positive, all 1's if negative
    int mask = x >> 31;

    // XOR with mask:
    //   no-op if mask is all 0's (0 XOR 0 = 0, 0 XOR 1 = 1)
    //   ~x if if mask is all 1's (0 XOR 1 = 1, 1 XOR 1 = 0)
    // Subtract mask:
    //   no-op if mask is all 0's
    //   +1 (subtract -1 = add 1) if mask is all 1's
    // Together:
    //   no-op if x is positive
    //   ~x + 1 = -x if x is negative
    return (x ^ mask) - mask;
}
```

Decoder + mux practice problem

ex



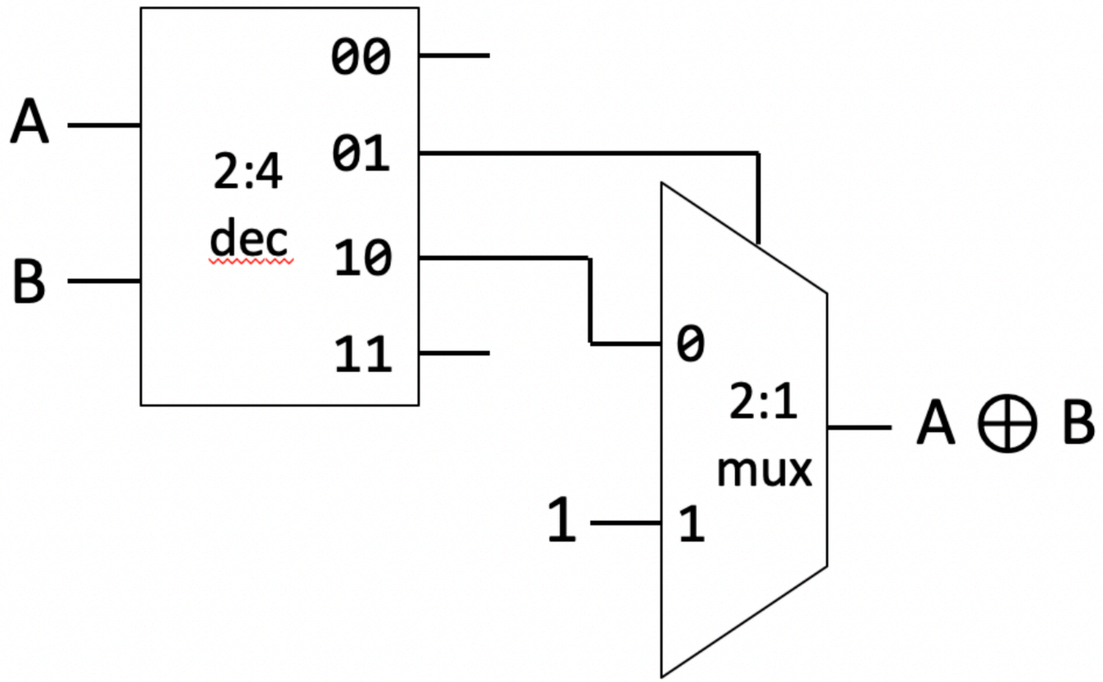


Use **one** 2:4 decoder and **one** 2:1 mux to implement A XOR B

Decoder + mux practice problem: solution

ex

Use **one** 2:4 decoder and **one** 2:1 mux to implement A XOR B



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 represent a register if the register file has 32 entries?

4. What is the purpose of the Mem control unit wire in the HW ISA processor microarchitecture? Why are 3 different multiplexers needed?

Short answer practice problems: solution

ex

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

The output of a D-latch changes based on input D the entire time input C = 1. The output of a D flip-flop changes based on input D only when the input C goes from 1 to 0 (falling edge).

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

Each instruction is encoded in 16 bits (as outlined in the instruction encoding table) and stored in a separate instruction memory. The PC registrar holds the address of the next instruction.

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

5. To choose between N options, we need $\log_2 N$ bits. $2^5 = 32$.

4. What is the purpose of the Mem control unit wire in the HW ISA processor microarchitecture? Why are 3 different multiplexers needed?

It determines whether to use only registers, or registers and memory in an instruction. 1 mux controls the ALU input, and 2 control the register write (one the address, one the data).