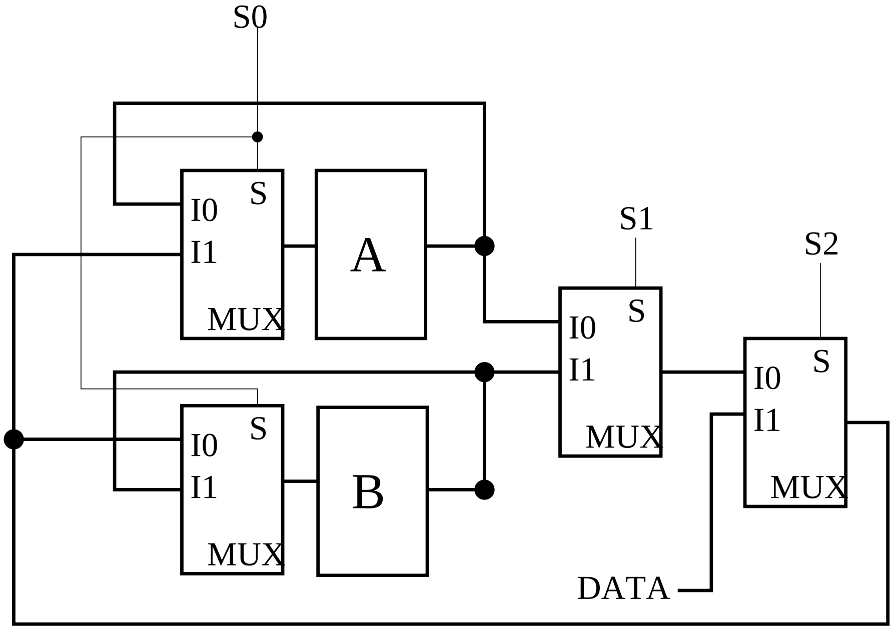
HW #6

CSEE W3827 - Fundamentals of Computer Systems Spring 2022

Prof. Rubenstein
04/4/2022

Topics: Memory, Coincident Selection, Register File
Note that this homework has 3 HARDER problems and is 5 pages long.

Warmup Problems



1. Consider the circuit pictured above, where

   - A and B are k-bit registers
   - DATA is a k-bit data input, set to some arbitrary value
   - MUX is a 2-to-1 multiplexer

   With the 3 selector inputs, there are 8 possible input combinations. For each combination of $S_2 S_1 S_0$, describe what this circuit does.

2. This problem covers old material but is needed for question 2b of the harder problems.

   A 1MASK circuit takes a k-bit input, $\mathcal{I}$ and produces a 2^k-bit output whose ith bit is set to 1 when and only when $\mathcal{I}$, interpreted as an unsigned binary value, is **larger** than (and not equal to) i.

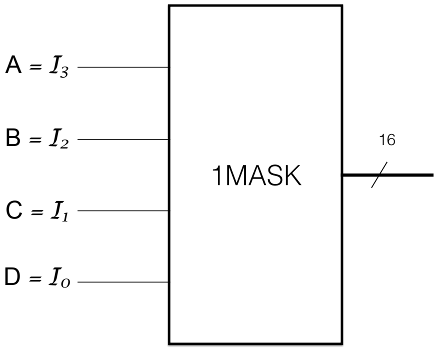
   For the remainder of the question, assume $k = 4$.

   For example, the following values of i produce the following 16-bit outputs, ordered from high bit $(i = 15)$ to low bit $(i = 0)$:

   - $\mathcal{I} = 0 = 0000 \rightarrow$ 0000 0000 0000 0000
   - $\mathcal{I} = 1 = 0001 \rightarrow$ 0000 0000 0000 0001
   - $\mathcal{I} = 7 = 0111 \rightarrow$ 0000 0000 0111 1111
   - $\mathcal{I} = 15 = 1111 \rightarrow$ 0111 1111 1111 1111

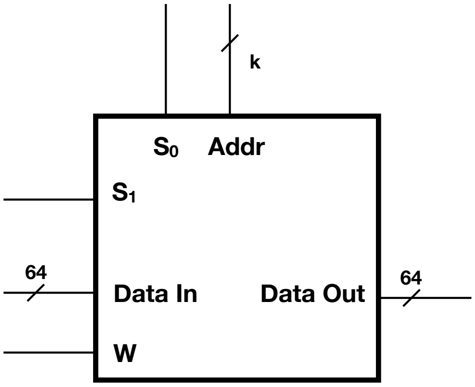
   (a) Give simplified (sum-of-product) expressions that describe the circuits that take as input A,B,C,D (high-to-low ordering of input bits) and produce the:

   i. 0-th output bit
   ii. 6-th output bit
   iii. 8-th output bit .
   iv. 14-th output bit



   (b) Using 4-to-16 1MASK circuits as pictured above, construct a 5-to-32 1-MASK circuit using multiple 1MASK circuits and external (AND, OR, NOT) gates. (Hint: think about the effect that the highest-order input bit has on the 16 low-order output bits and the effect it has on the 16 high-order output bits)

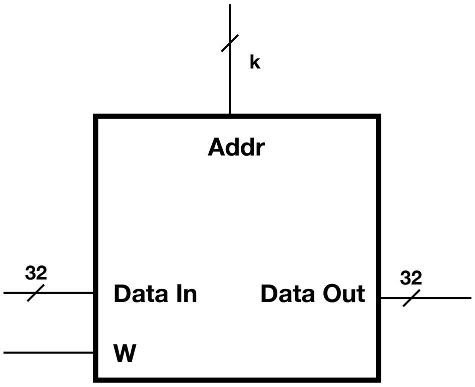
Harder Problems



1. The memory chip pictured above can be used to store 2^k 64-bit words, or alternatively can be used to store 2^{k+1} 32-bit words. It behaves as follows:

    - It takes a 64-bit DATA input, I, a k-bit address input $A = A_{k-1}A_{k-2}\cdots A_1A_0$, a 1-bit WRITE input W, and 2-bit selector S_1S_0, and produces a 64-bit DATA output O,
    - When $W = 1$, the memory is written to.
    - When $S_1 = 1$, the chip reads or writes 64 bits of data from/to address A.
    - When $S_1 = 0$, the chip reads or writes 32 bits of data with the following behavior:
        - the address is specified by $S_0A_{k-1}A_{k-2}\cdots A_1A_0$ (i.e., a $k+1$-bit address).
        - When writing to memory, data is written from the 32 least significant bits of the input, i.e., the most significant 32 bits are unused for write.
        - When reading from memory, the data is written to the 32 least significant bits of the output, and the 32 most significant bits are all set to 0.

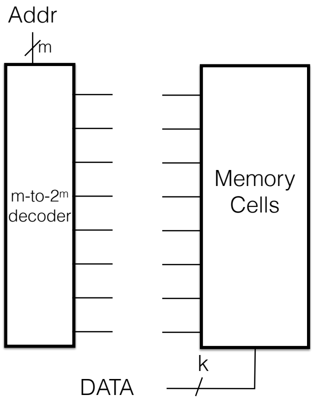
    In this problem, you will build this chip from two 32-bit word 2^k address chips (one is pictured below) and basic circuitry (e.g., MUXes, Decoders, AND, OR, NOT gates).



    (a) Let's enumerate the two chips you will use as Chip 0 and Chip 1. Draw circuitry showing what feeds into the DataIn and Addr inputs of these two chips (i.e., in terms of S_0, S_1, A, and I).

    (b) Give algebraic expressions for W_i, which will feed into Chip i's W input.

    (c) Draw (in a separate diagram from part a) the circuitry to produce the 64-bit output of the chip being built.

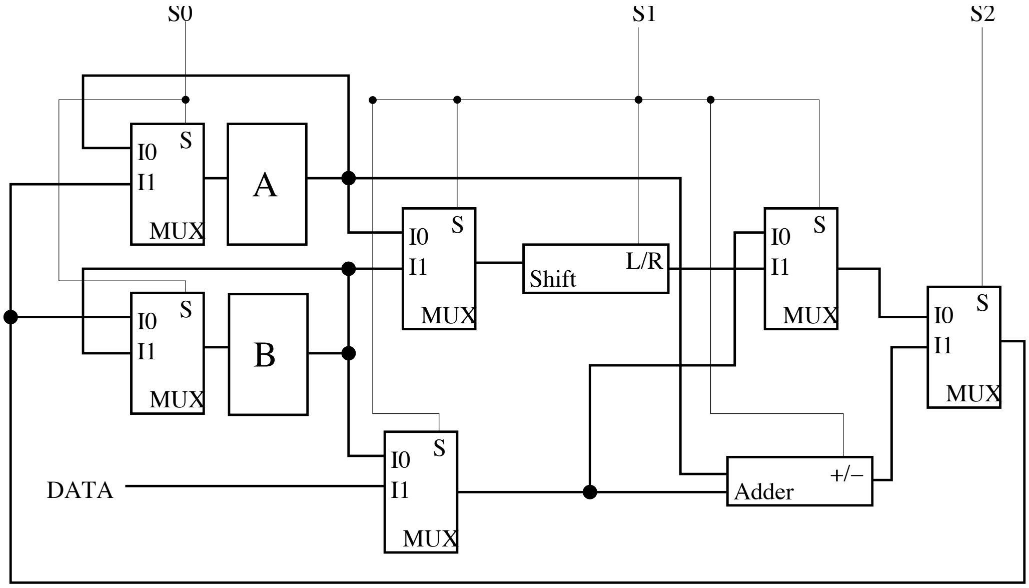


2. A memory-with-reset takes one additional input, **reset**. If **reset** = 1, then the entire memory is cleared (all cells set to 0).

   (a) In the figure above, representing a memory that can store 2^m k-bit words, assume that to implement the memory without reset, the outputs of the decoder can be hooked up directly to the row inputs of the memory cell, and DATA can be hooked up directly to the DATA input at the bottom. How would you modify those feeds so that a 1-bit **reset** clears the entire memory, but that the chip behaves normally when **reset** = 0? Draw the revised figure with the additional circuitry.

   (b) In part (a), the address input would be ignored during a reset. Suppose instead, when a **reset** is issued, only the addresses **less than** the specified address are cleared. How would you construct this circuit? (Hint: you may use the 1MASK circuit from the Warmup Problems in your solution).

   (c) Suppose coincident selection is used, such that the m_1 high order bits of an address are input into the row decoder, and $m_0 = m - m_1$ bits are input into the column decoder. Suppose, a contiguous region of memory must be cleared by **reset** (i.e., if addresses A and $B > A$ are both cleared, then all addresses between A and B must also be cleared). How does coincident selection limit the ranges of addresses that can be cleared?



3. Consider the circuit pictured above, where

- A and B are k-bit registers
- DATA is a k-bit data input, set to some arbitrary value.
- MUX is a 2-to-1 multiplexer
- Adder is an adder/subtracter that takes the bottom input and adds or subtracts it from the top input respectively depending on whether its selector input is set to 0 or 1
- Shift is a shifter that takes an input and shifts it left or right respectively depending on whether its selector input is set to 0 or 1.

(a) With the 3 selector inputs, There are 8 possible input combinations. For each combination of $S_2S_1S_0$, describe what this circuit does.

(b) Give all possible values of S_2, S_1, and S_0 and DATA that would perform a transfer of the value stored in A into B. (Hint: if you use DATA, you may choose how to set its value, e.g., to a constant of your choosing).