CSM 61C Spring 2019

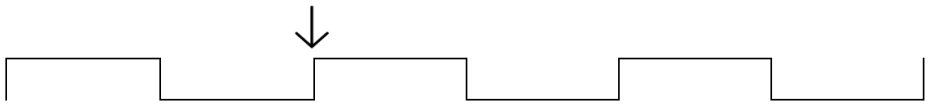
SDSs and FSMs

Mentoring 6: February 24, 2019

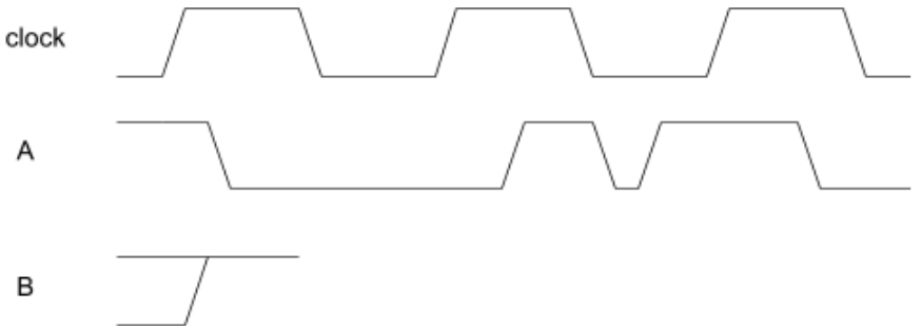
1 Synchronous Digital Systems

Synchronous Digital Systems (SDSs) are systems designed to process input as it comes in and output it as quickly as it comes in. The input is always a series of pins with either high or low voltage, and the output is another series of pins, each with either high or low voltage. A clock is used to tell the input and output when to switch to the next value; the switch always occurs on the rising edge (low to high voltage) of the clock. This is so that the output is always consistent, otherwise halfway through the switch of the input, the output might be based on the old input or the new, depending on the quality of the circuit and also some random chance based on electron movement. Note: the output always takes a small amount of time to stabilize after the rising edge.

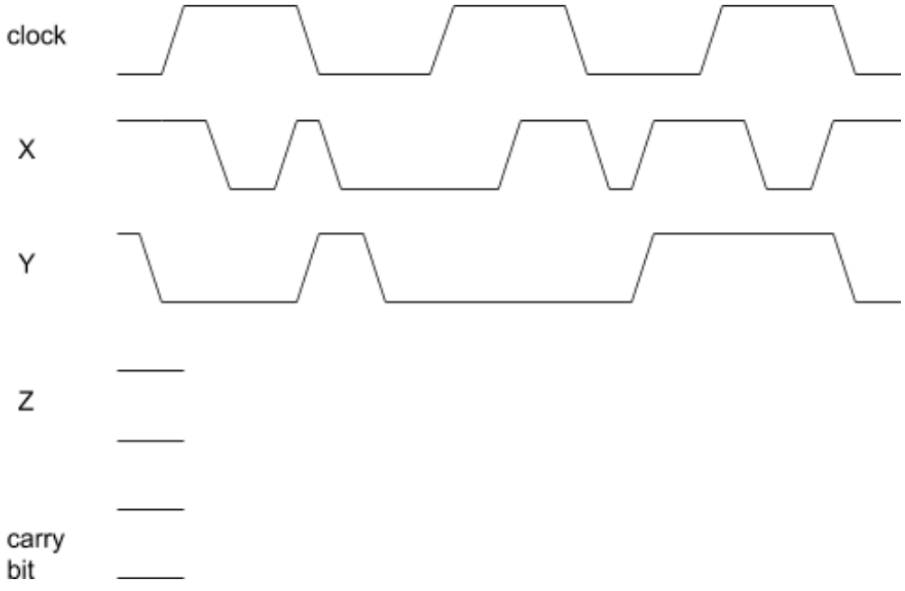
When slowed down, clock cycles will look something like below. This is three clock cycles in a row, and here is the rising edge:



1.1 On the rising edge of the clock, B is set to A and is held constant until the next rising edge. Fill in the diagram for Bs value.



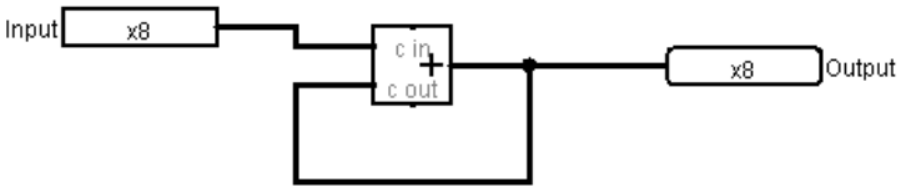
In a clock cycle, complex operations can occur. For example, most of the RISC-V instructions can be done in 1 clock cycle. Below, the diagram shows the equation Z = X + Y where Z is computed on the rising edge of the clock. If Z overflows, place the overflow in the carry bit.



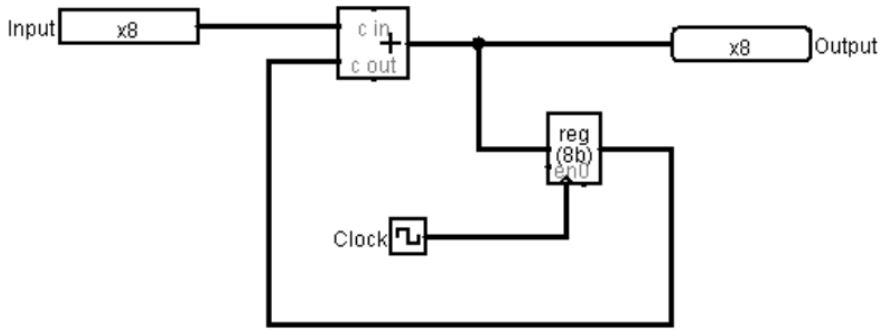
2 SDS With Registers

2.1 Registers, like the 32-bit ones in RSIC-V, are a valuable part of many circuits. They allow the retaining of a value and constant output no matter how the input changes unless they get a signal to accept a new input. This signal is usually the rising edge of the clock, upon which they will change their value and output to the input at the rising edge of the clock. This means that for any input, the input must be stable for a certain amount of time both before and after the rising edge of the clock because the rising edge takes time even though its a very small amount. The time before is called the setup time and the time after is called the hold time.

The circuit below is an accumulator that adds the input to the inputs previous values (i.e. after inputs 1 and 5, the accumulator should hold 6). Both the input and the output are 8-bit unsigned integers. The adder block (the square) adds the two inputs together and puts the result in the output wire immediately. What is the problem with this circuit once the input changes from 0x00?



2.2 b) The revised accumulator circuit has been modified below with a register. At every rising edge of the clock, registers will take in new input and then output it at the other end. Write the values of the output for time t = 0 through t = 3, given the register and input both held value 0 before time t = 0 and given the input stream 0x01, 0x5A, 0x00, and 0x13.



Time:	t = 0	t = 1	t = 2	t = 3
Input:	0x01	0x5A	0x00	0x13
Output:				

3 Finite State Machines

Suppose we want to design a FSM that takes a single bit (1/0) as input, and outputs a single bit (1/0). We want the FSM to output true (1) only if it has seen three consecutive 1s as its input. Lets design it!

3.1 Given the following input streams, fill in what the FSM should output at each time step:

(a)

In	0	1	0	1
Out				

(b)

In	1	1	0	0
Out				

(c)

In	0	1	1	1
Out				

3.2 Lets consider the design of the FSM with more formality.

(a) If a 0 is input into the FSM, what should the FSM output?

(b) If a 1 is input into the FSM, what does the FSM need to remember to make the correct decision?

(c) How many unique states does the FSM need?