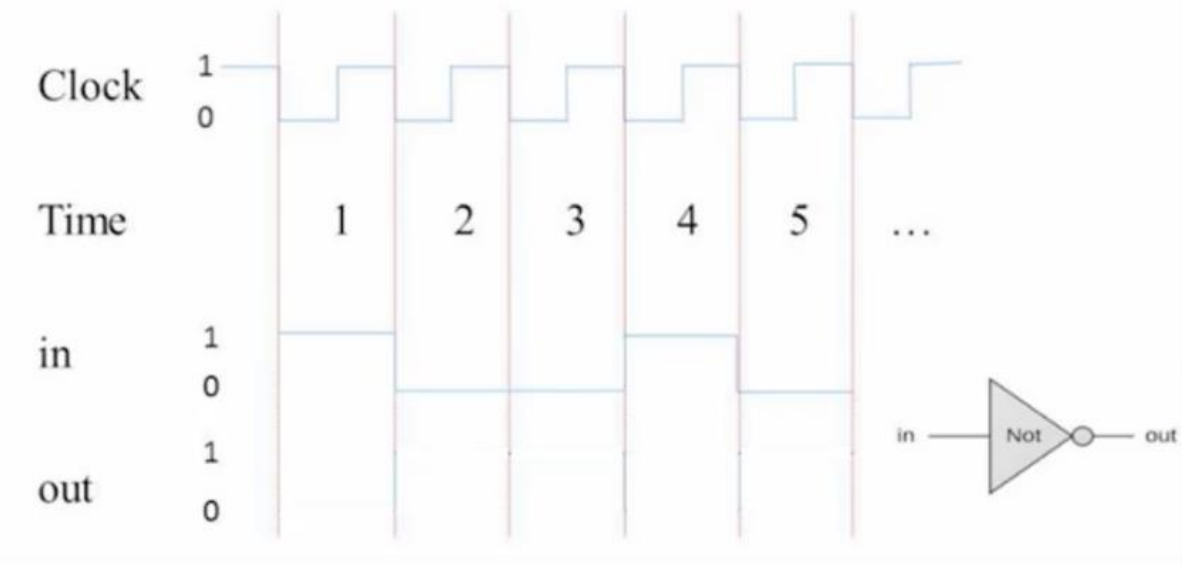
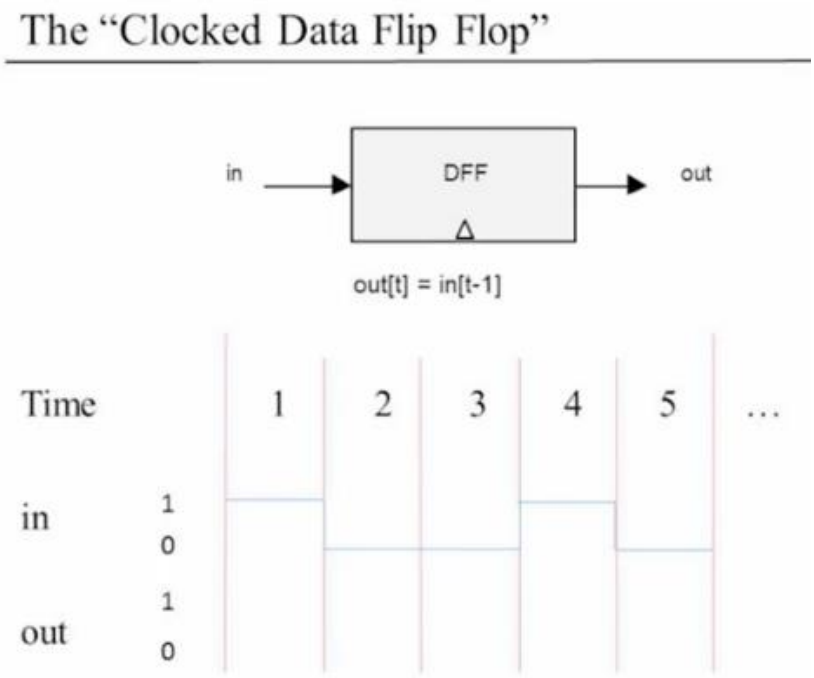
Lab #4 (Sequential Logic)

Name: _____
Section/Time: _____
Date: _____

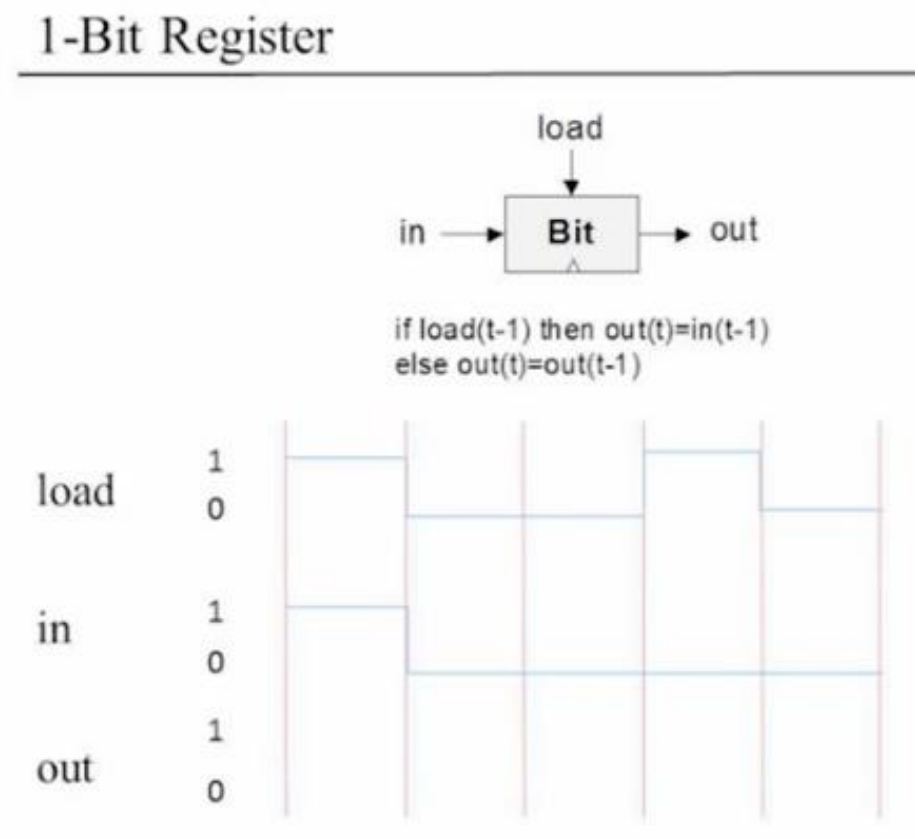
1. For the following combinational logic (not gate):
    a. Label all the ticks and tocks on the clock line.
    b. Complete (draw) the output line, given the input during that time.
    c. Circle all the latch points.



2. For the following sequential logic (clocked data flip-flop (DFF)):
    a. Complete (draw) the output line, given the input during that time.
    b. Circle all the latch points.



3. For the following sequential logic (1-Bit Register):
    a. Complete (draw) the output line, given the input during that time.
    b. Circle all the latch points.

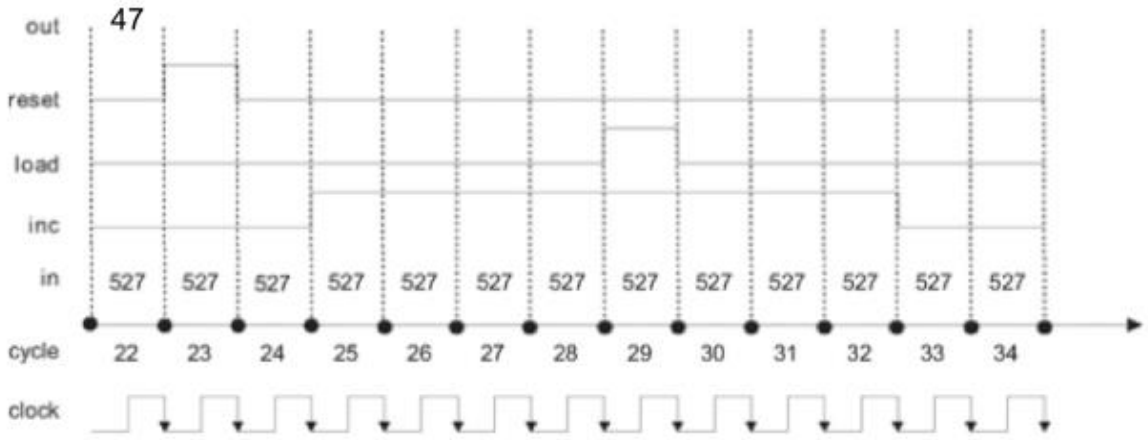


4. Why do we use discrete time steps instead of continuous time?

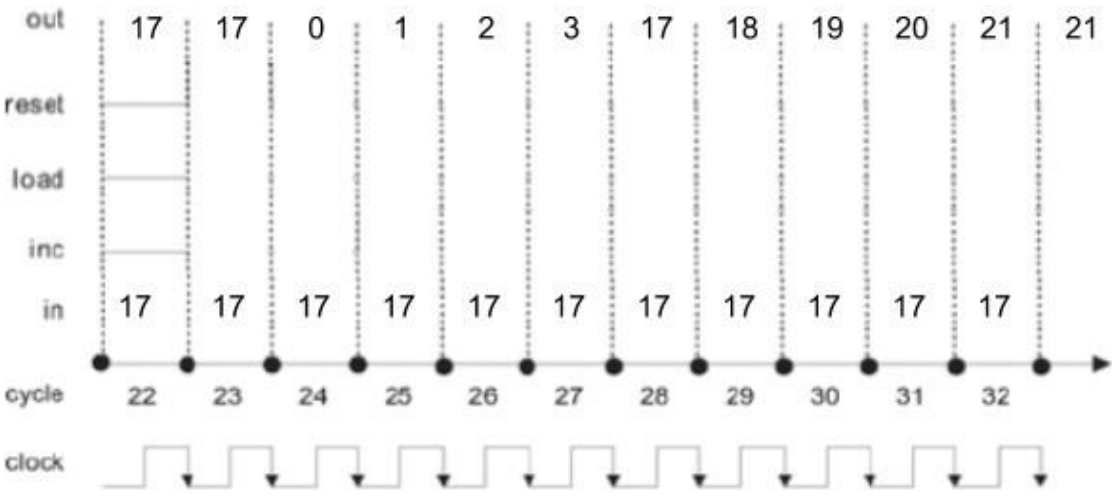
5. **What are the *out* values for each clock cycle below?** Note: input starts at 527, output starts at 47 (from previous clock cycle).

We assume that we start tracking the counter in time unit 22, and the counter's control bits (reset, load, inc) start at 0. All values below are arbitrary, to test your understanding of clock cycles, latching, etc.

HINT: Control bits dictate NEXT output because outputs latch on tock (end of cycle). Order matters! (top to bottom)



6. **What are the control bits (reset, load, inc) values given the input and output values below?** Note: All control bits start low (0). You can draw the lines and/or put the 0/1 values for each clock cycle and control bit



Lab #4 (Sequential Logic)

Name: _____
Section/Time: _____
Date: _____

Part 1: Definitions

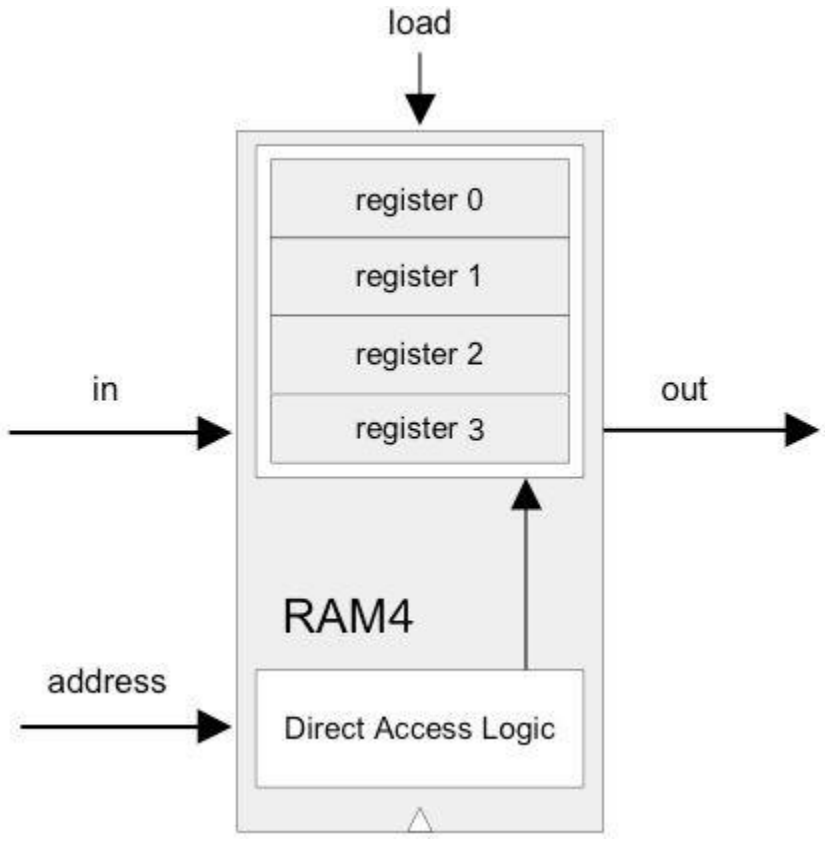
- **Define what a "word" is in computer architecture:**

- **What is the term for the size of a "word"?**

- **What is the size of the word in the HACK architecture?**

Part 2: Bus Sizes

- **Label the size of each of the inputs/outputs for the Ram4 chip below (in, address, load, out). You can assume the register sizes are the same as our HACK architecture.**



Part 3: Reasoning

- **Why did you choose the address size above?**

Part 4: Bus Sizes

- **Write out the address for each register:**

Register #	In Binary	
	[1]	[0]
0		
1		
2		
3		

- **Each register has it's own load and in values. Fill out the table below for each situation:**

	Register 0		Register 1		Register 2		Register 3		RAM4 OUTPUT (out, after end of cycle)
Situation:	in	load	in	load	in	load	in	load	
Store -15 in Register 0									
Store -15 in Register 1									
Store -15 in Register 2									
Store -15 in Register 3									
Store -15 in Register 1 and 2									

- **When does the load value matter? When doesn't it?**

- **What chip (that we've built in the course so far) would help for handling the load?**

- **When does the in value matter? When doesn't it?**

- **What chip (that we've built in the course so far) would help for handling the in?**

Write the following numbers in binary (NO CALCULATOR! Use back of paper):

- ***Tip: put a space between every 4 digits (grouping from right to left)***

Decimal Number:	Binary number:
7	
63	
511	
4095	
16383	

- **The above values can be thought of as the last register in a specific RAM chip**
  - **Ex: RAM8 = 8 registers = 0, 1, 2, 3, 4, 5, 6, 7 are possible registers**
- **What would be the address sizes for the following n Registers:**

RAMn (n = num. of registers)	Number of bits for address (think highest value…)
RAM8	
RAM64	
RAM512	
RAM4K (n=4096)	
RAM16K (n=16384)	

- **How does one calculate the size of the address (number of bits) from the word size for the registers?**