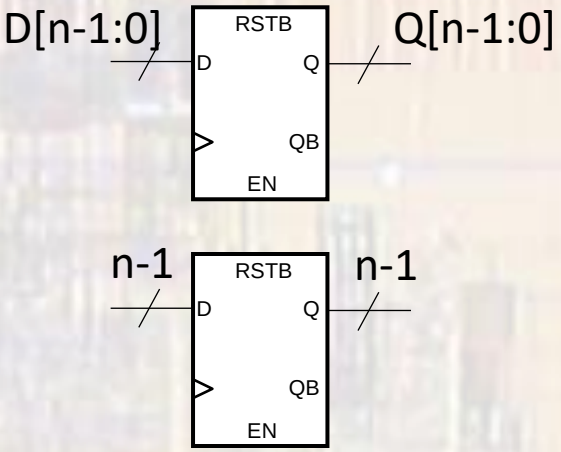
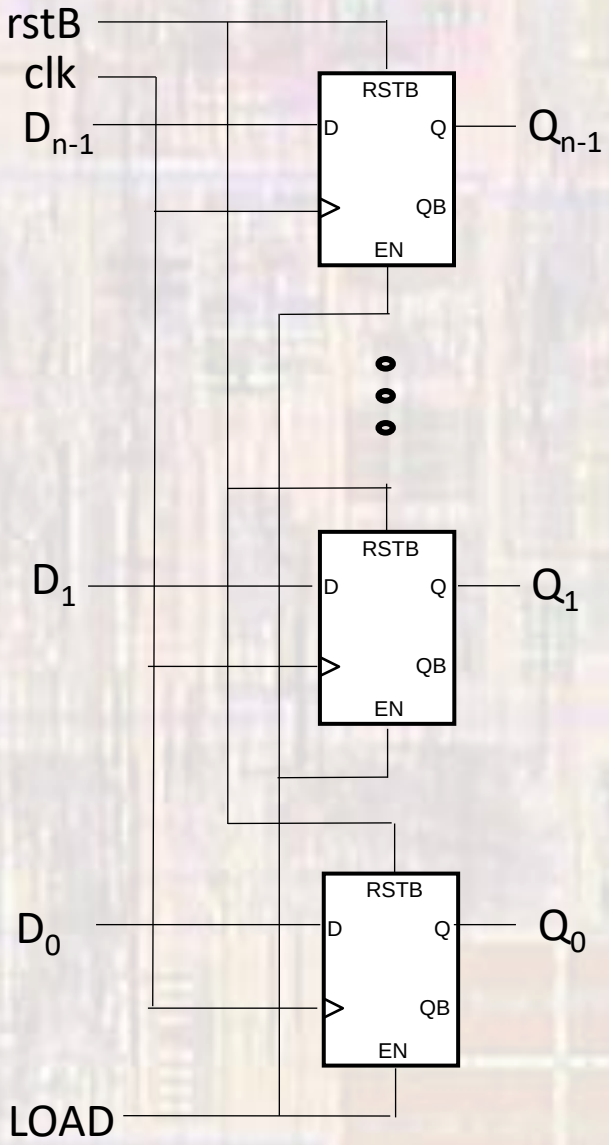
Flip-Flop Circuits

Last updated 1/14/21

Flip Flop Circuits

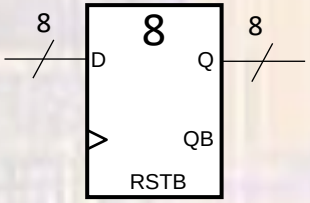
- Register
  - Collection of n Flip-Flops
  - Configured in parallel
  - Common clock, set/reset, enable
  - Stores n state variables

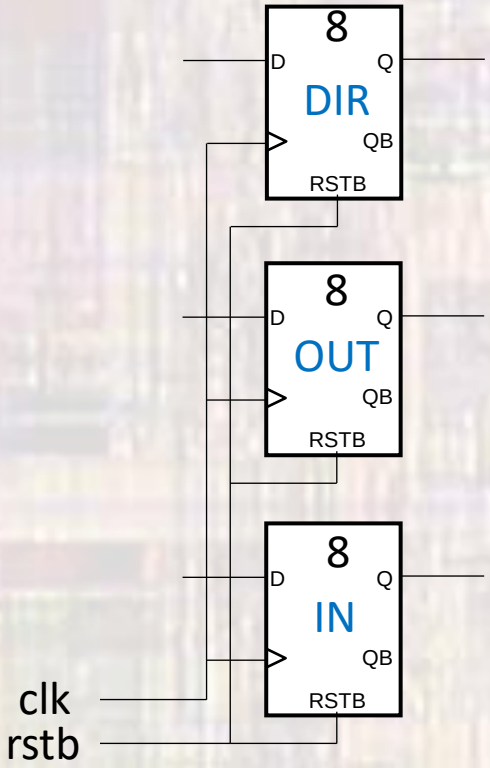




Flip Flop Circuits

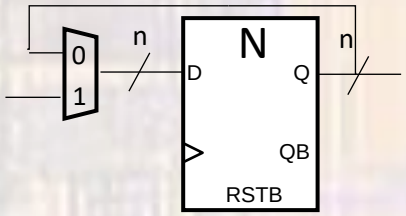
- Register
  - I/O Port on MSP432

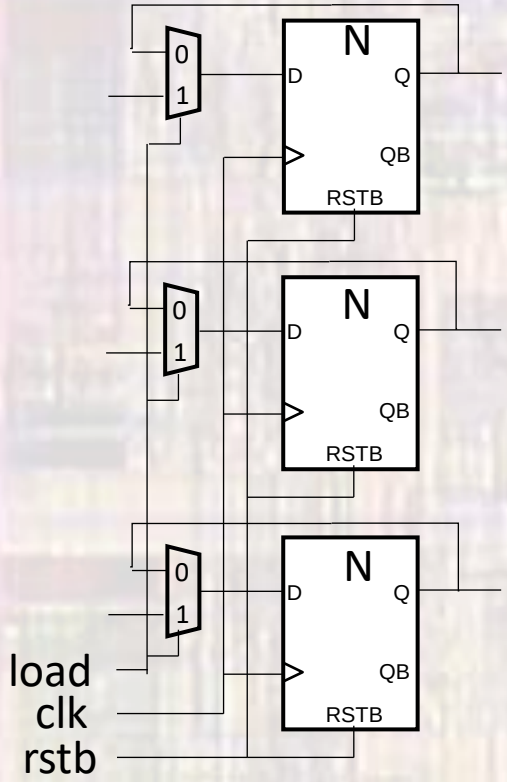




Flip Flop Circuits

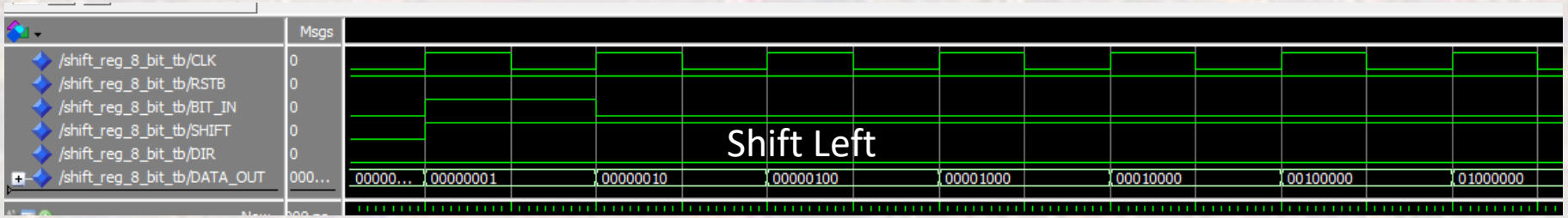
- Register
  - With Load (enable)





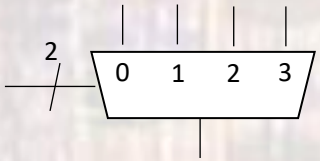
Flip Flop Circuits

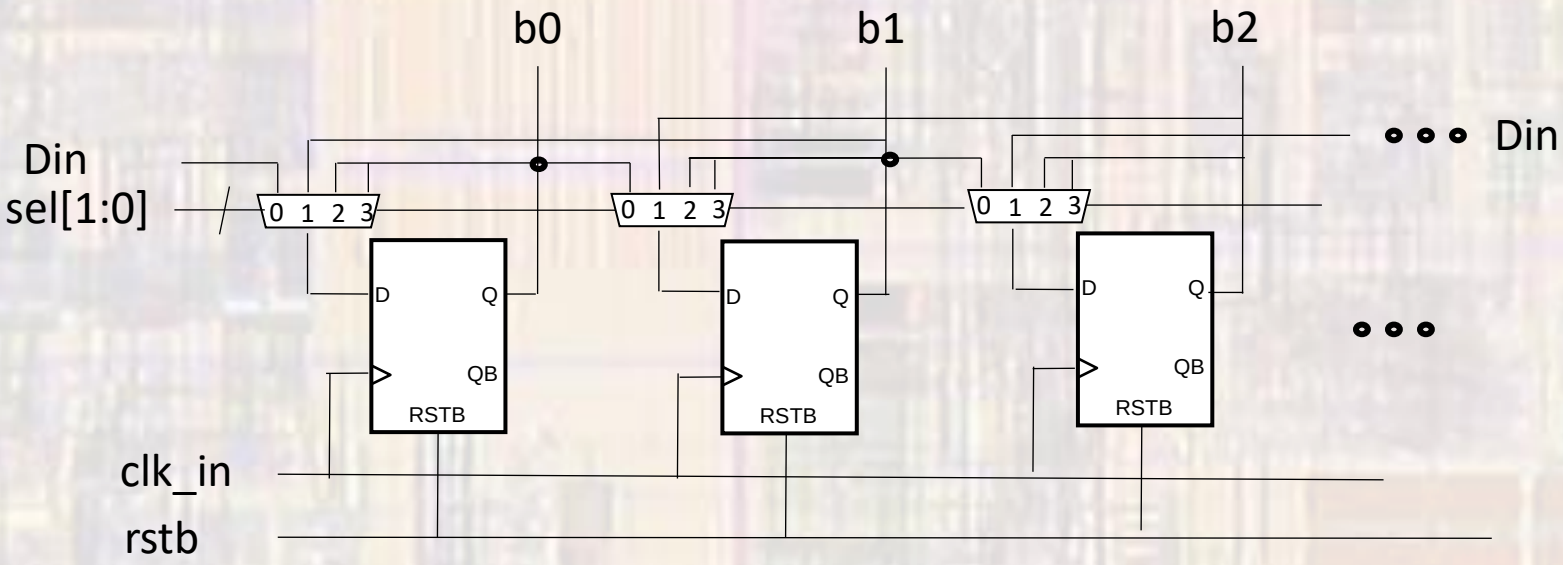
- Shift Register
  - Shift a series of bits within a register (1 element L/R)



Flip Flop Circuits

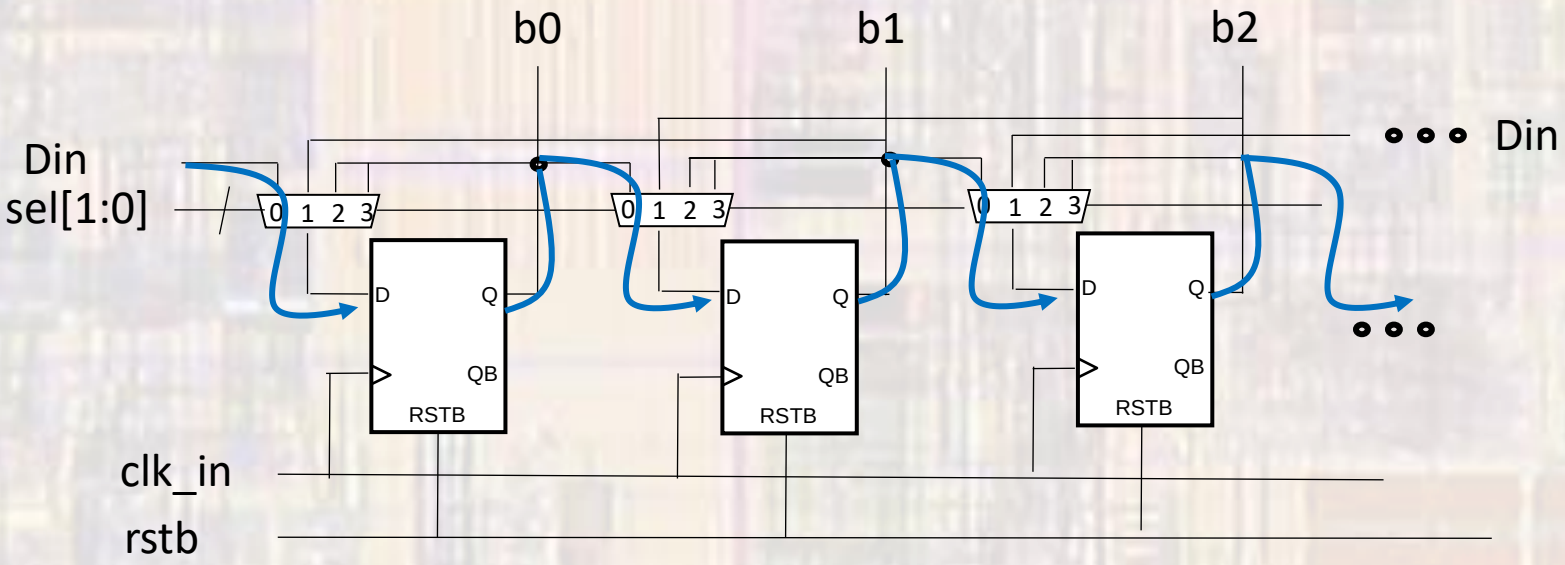
- Shift Register
  - Shift a series of bits within a register (1 element L/R)
  - sel[0]: (0) shift right, (1) shift left
  - sel[1]: (0) shift, (1) no-shift





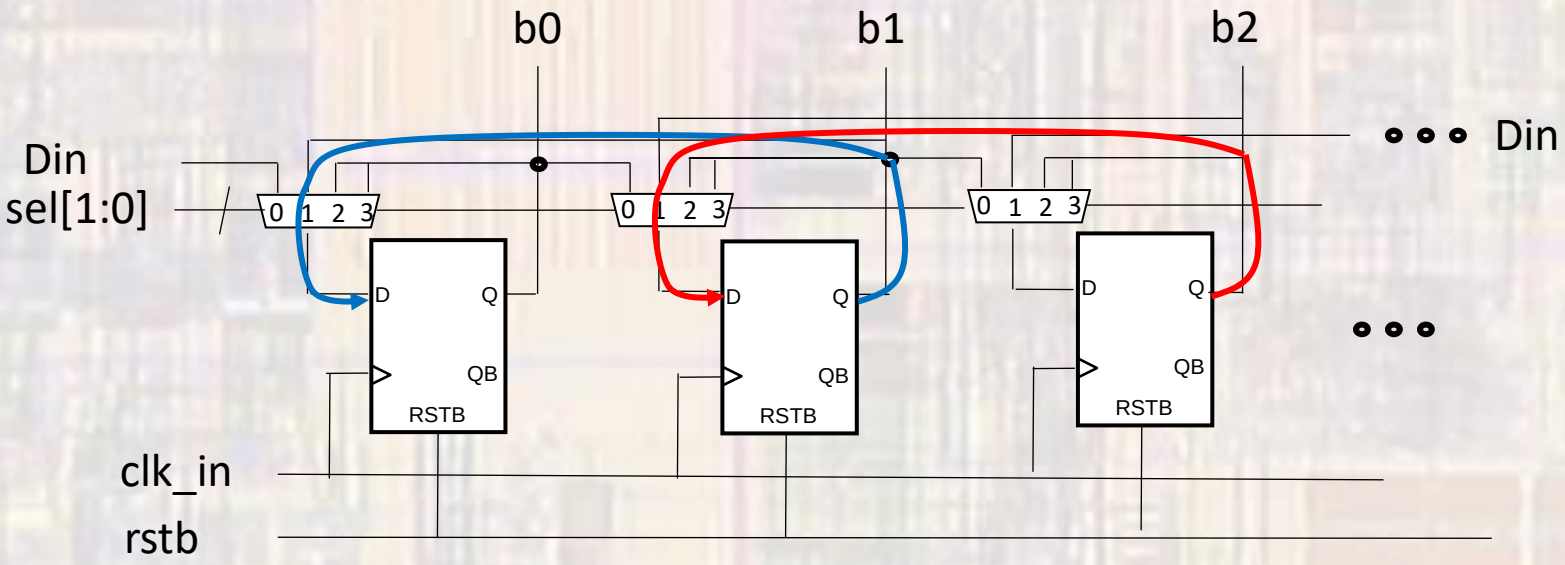
Flip Flop Circuits

- Shift Register
  - Shift a series of bits within a register (1 element L/R)
  - sel = 0,0 shift left



Flip Flop Circuits

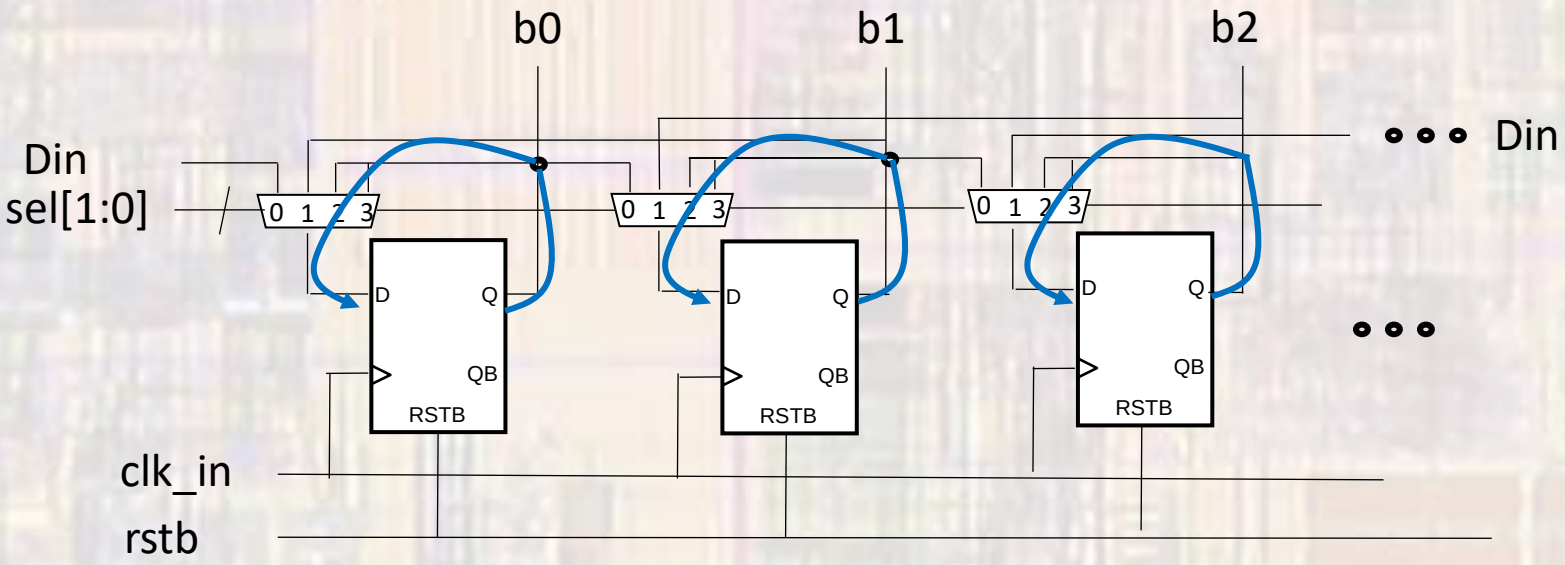
- Shift Register
  - Shift a series of bits within a register (1 element L/R)
  - sel = 0,1 shift right



Our circuit does not indicate what to shift into the last bit location

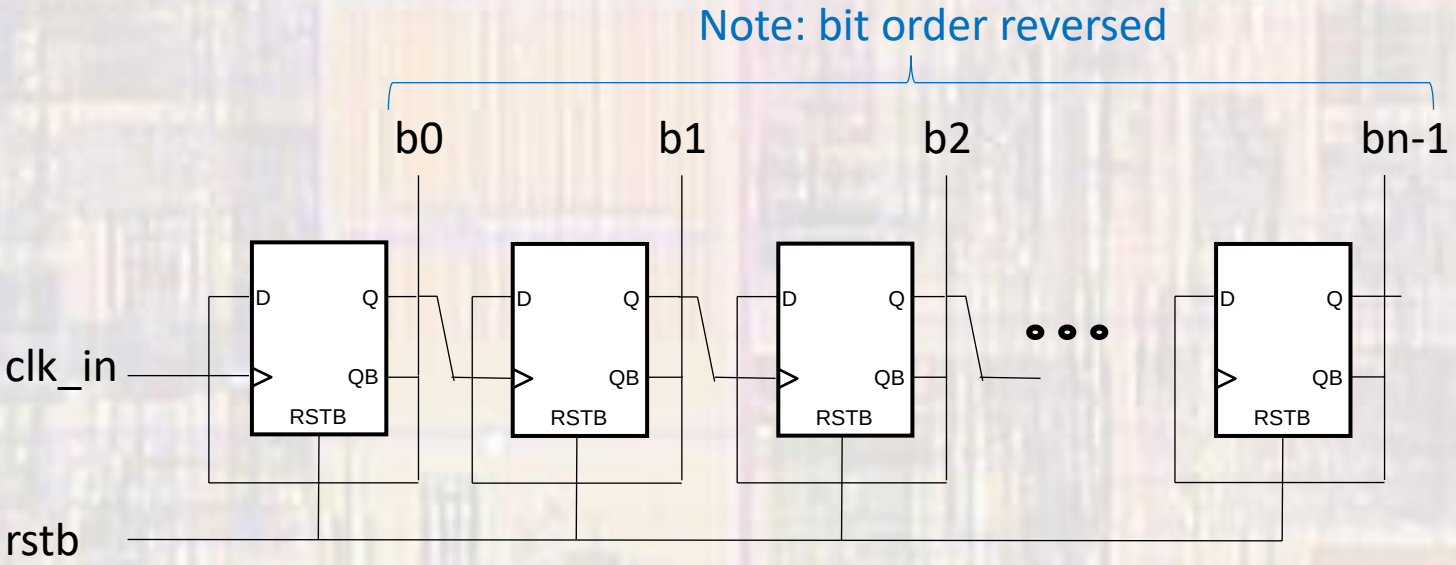
Flip Flop Circuits

- Shift Register
  - Shift a series of bits within a register (1 element L/R)
  - sel = 1,0 or 1,1 no shift



Flip Flop Circuits

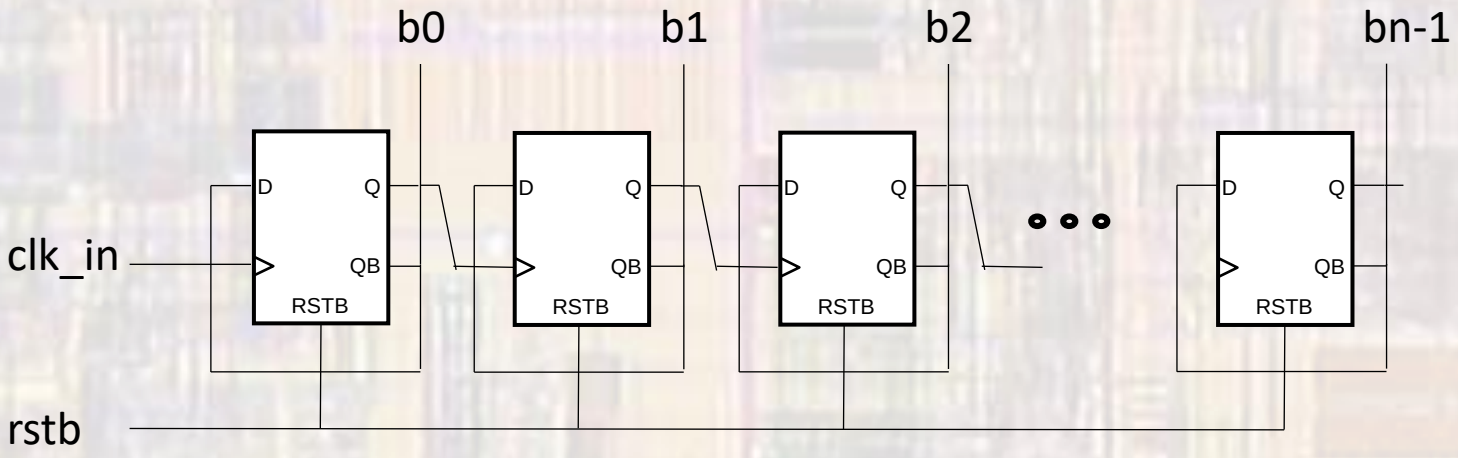
- Counter
  - Count in binary
  - 0000 → 0001 → 0010 → 0011 → 0100 … 1111 → 0000 …



Flip Flop Circuits

- Counter

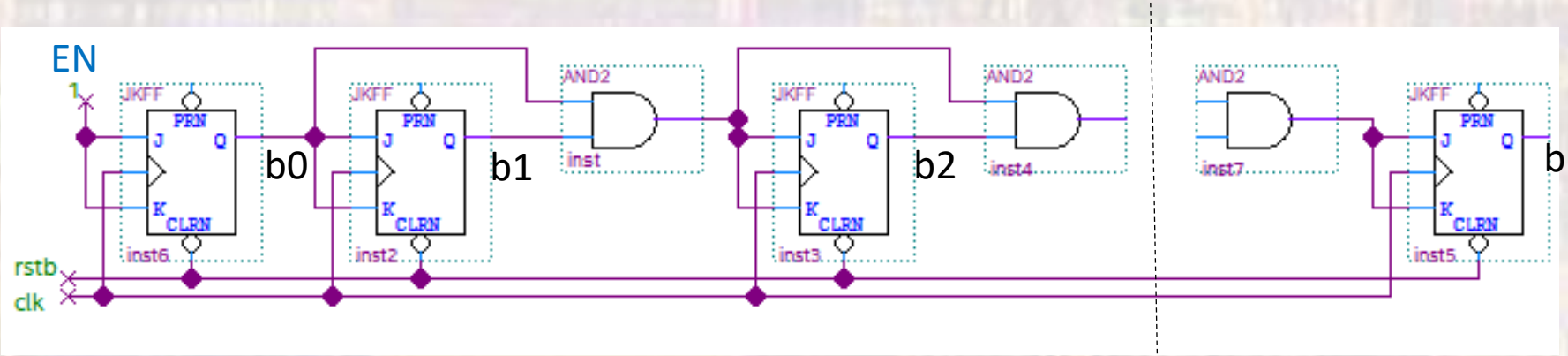
	0	0	0	0
clk ↑	1	0	0	0
clk ↑ Q1 ↑	0	1	0	0
clk ↑	1	1	0	0
clk ↑ Q1 ↑ Q2 ↑	0	0	1	0



What's wrong with this solution

Flip Flop Circuits

- Counter
  - Count in binary
  - 0000 → 0001 → 0010 → 0011 → 0100 … 1111 → 0000 …



en	b0	b1	b2	b3	bn-1
1	0	0	0	0	0
1	1	0	0	0	0
1	0	1	0	0	0
1	1	1	0	0	0
1	0	0	1	0	0
1	1	0	1	0	0
1	0	1	1	0	0
1	1	1	1	0	0
1	0	0	0	1	0

Flip Flop Circuits

- Modulo 10 Counter
  - Mod 10
  - Counts from 0 to 9, then back to 0, …

  - If output is 9 (1001), want next value to be 0 (0000)

    1001 → 1010 but we want 1001 → 0000

    no change in LSB

    don't allow bit 1 to change (force JK to 0 – *and* with B3 *not*)

    this works since no B1 change between 1000 and 1001

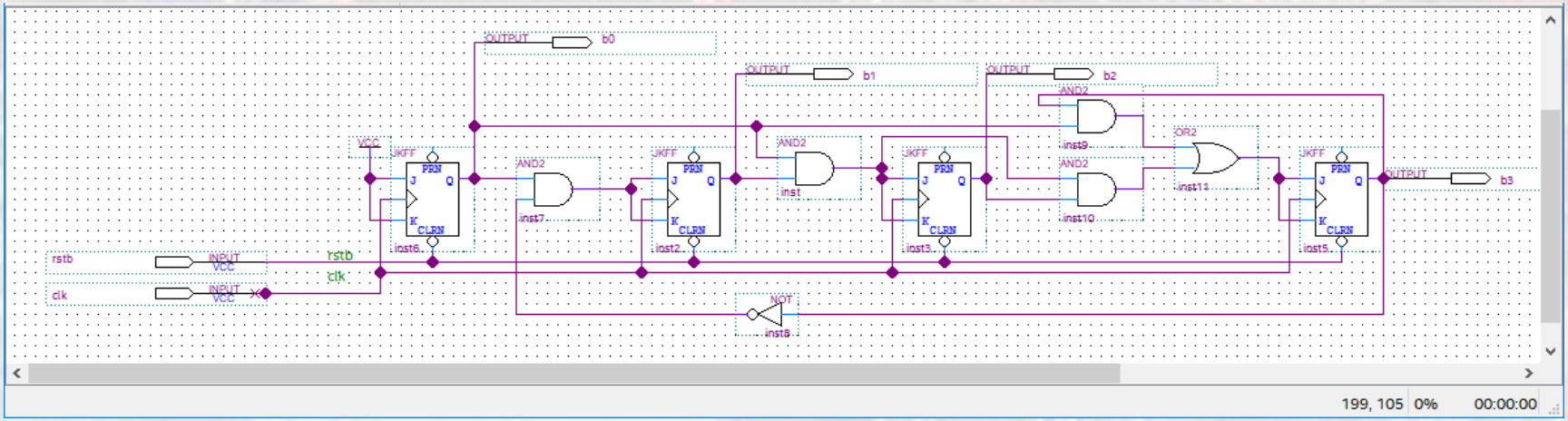
    no change in bit 2

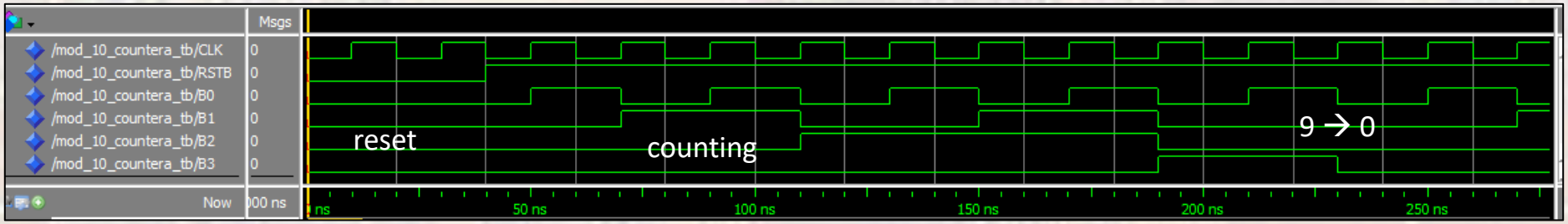
    force bit 3 to zero (force JK to 1 – add in a path with B0 *and* B1)

    once in 0 state – progresses normally

Flip Flop Circuits

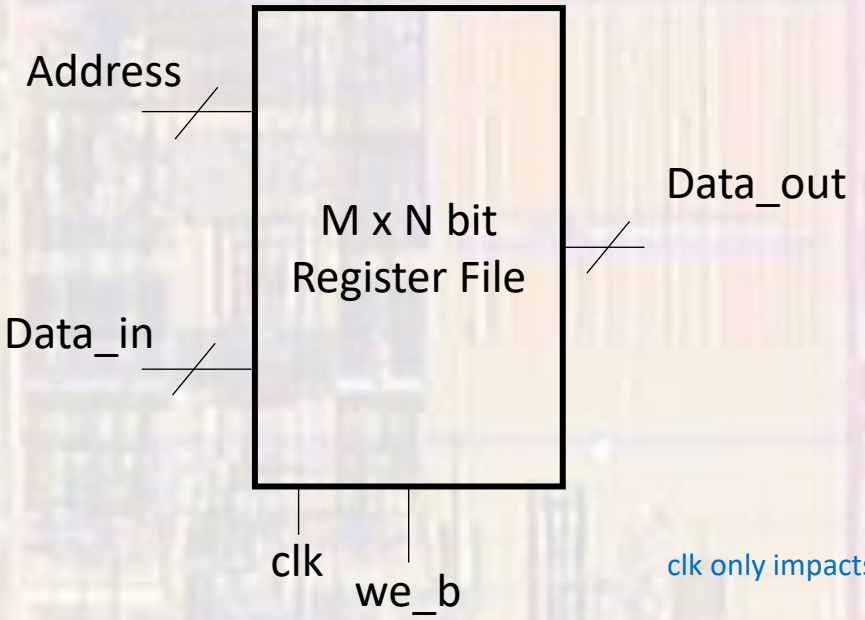
- Modulo 10 Counter
  - Mod 10
  - Counts from 0 to 9, then back to 0, …



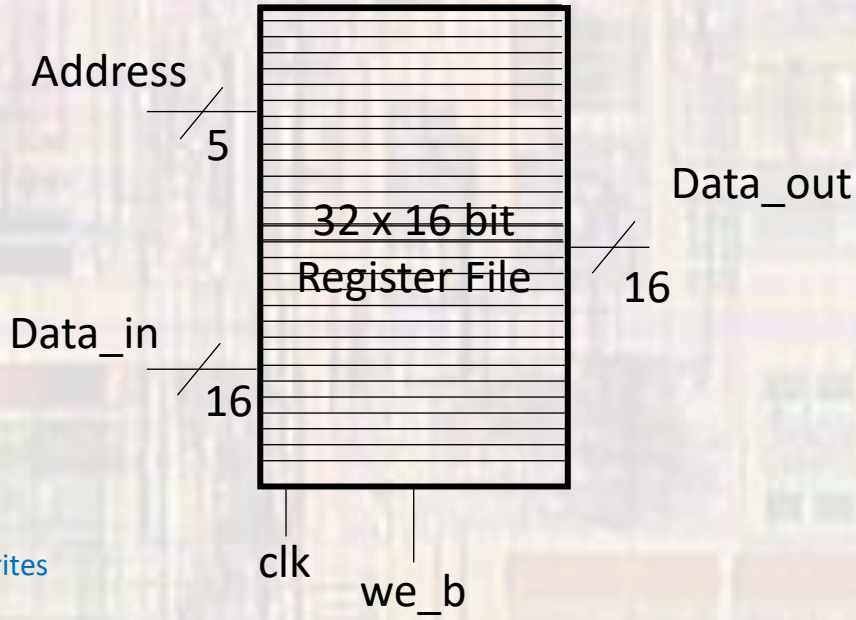


Flip Flop Circuits

- Register File – simple
  - Group of n-bit registers
  - Accessed via a multiplexor (address)



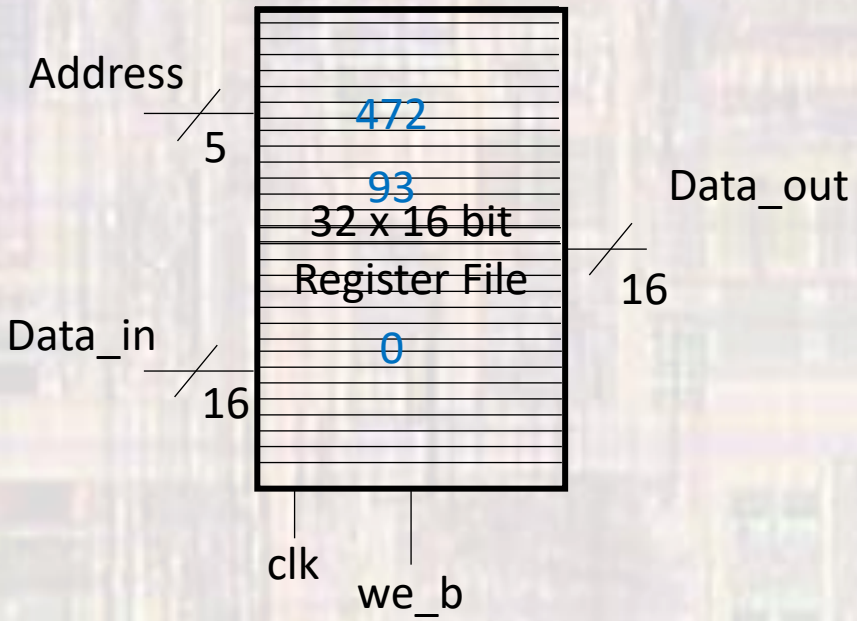
clk only impacts writes



Flip Flop Circuits

- Register File - simple

	we_b	addr	data_in	data_out
	1	7	472	?
clk ↑ →	0	7	472	?
	1	7	472	472
	1	7	93	472
	1	12	93	?
clk ↑ →	0	12	93	?
	1	12	93	93
	1	23	0	?
clk ↑ →	0	23	0	?
	1	23	0	0
	1	7	55	472
	1	12	55	93

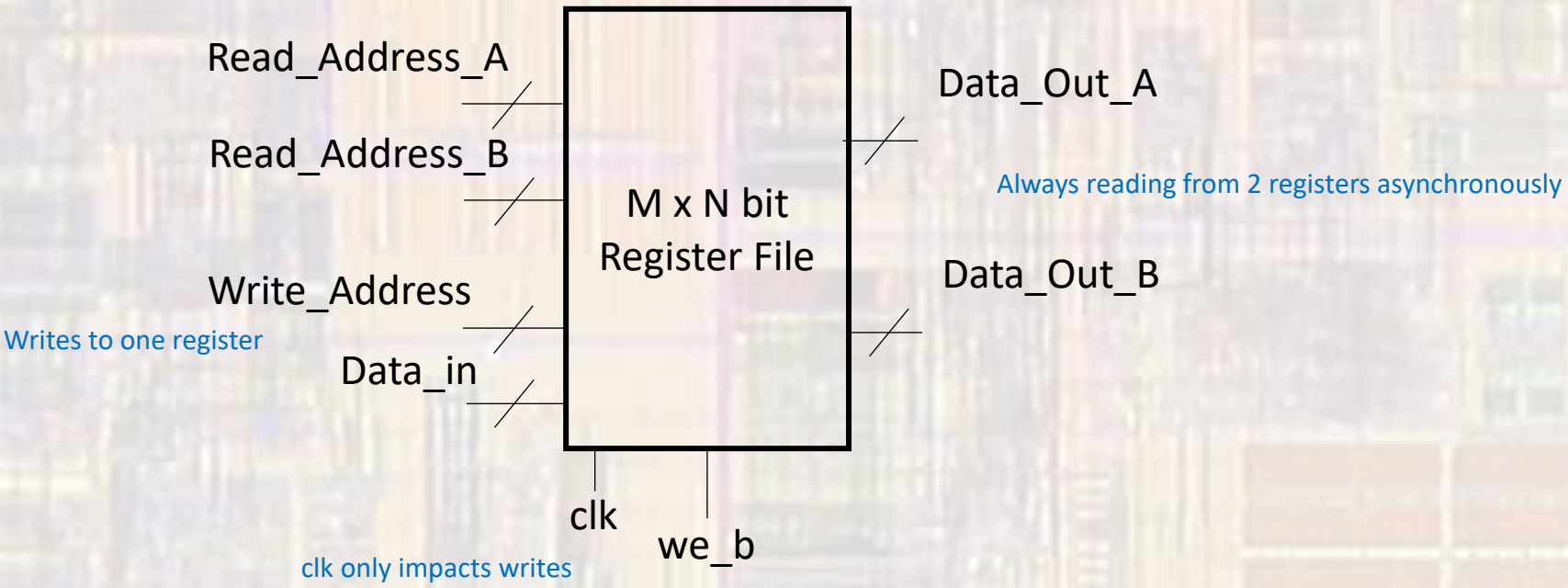


clk only required for writes, reads are asynchronous

In normal application, clock would always be running

Flip Flop Circuits

- Register File – multi-port
  - Group of n-bit registers
  - Accessed via a multiplexor
  - Can have multiple inputs and outputs



Flip Flop Circuits

- Clock Divider
  - Modulo counter
  - XOR gate to toggle the output clock

  - Divide by 24
    - Modulo 12 counter
    - XOR gate that toggles the output every time the counter hits 0