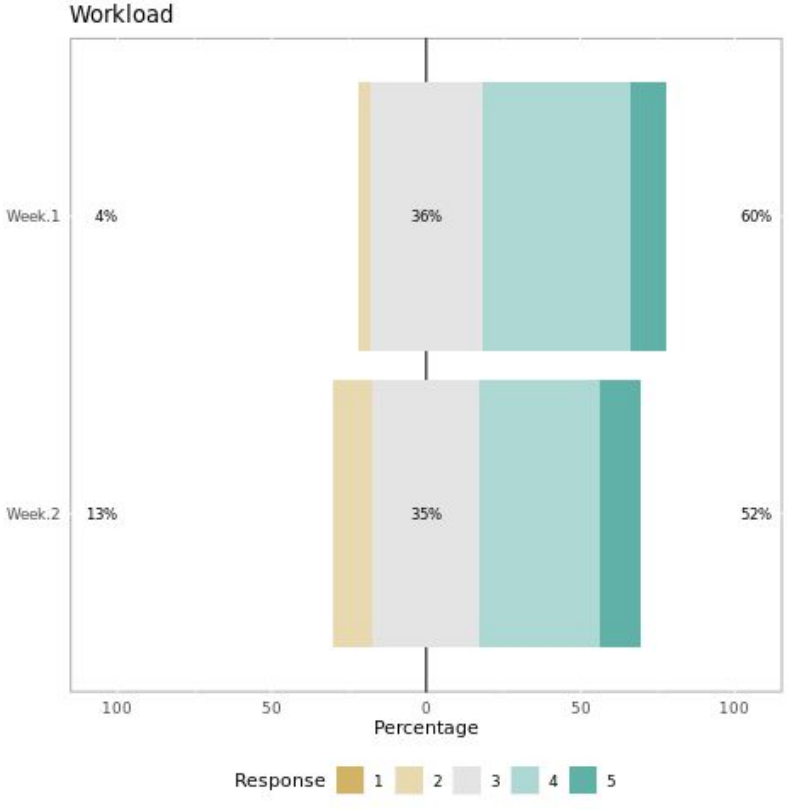
0x04 - Flops and Metrics

ENGR 3410: Computer Architecture

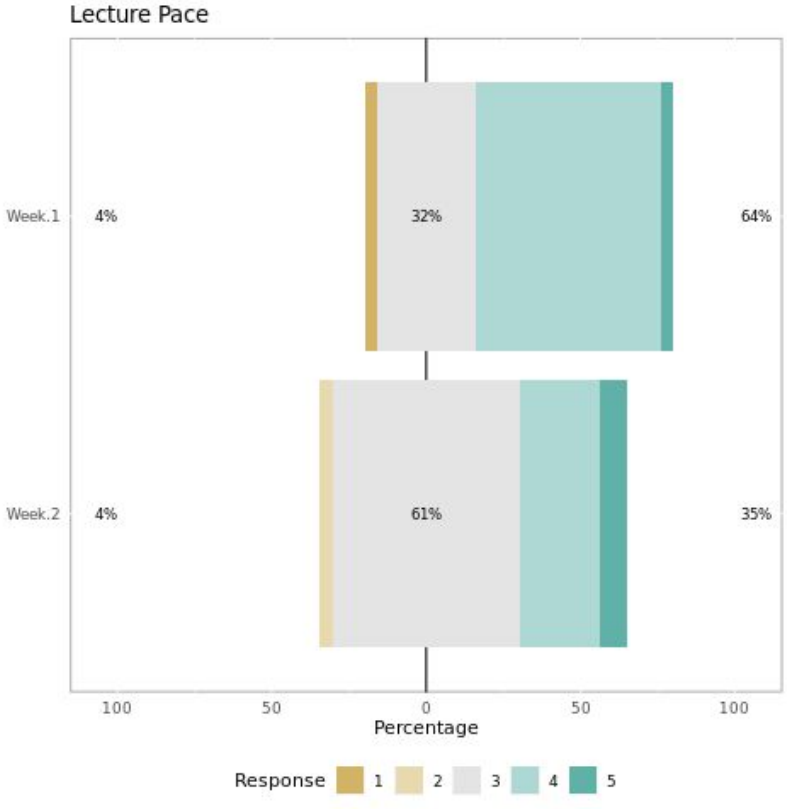
Jon Tse

Fall 2020

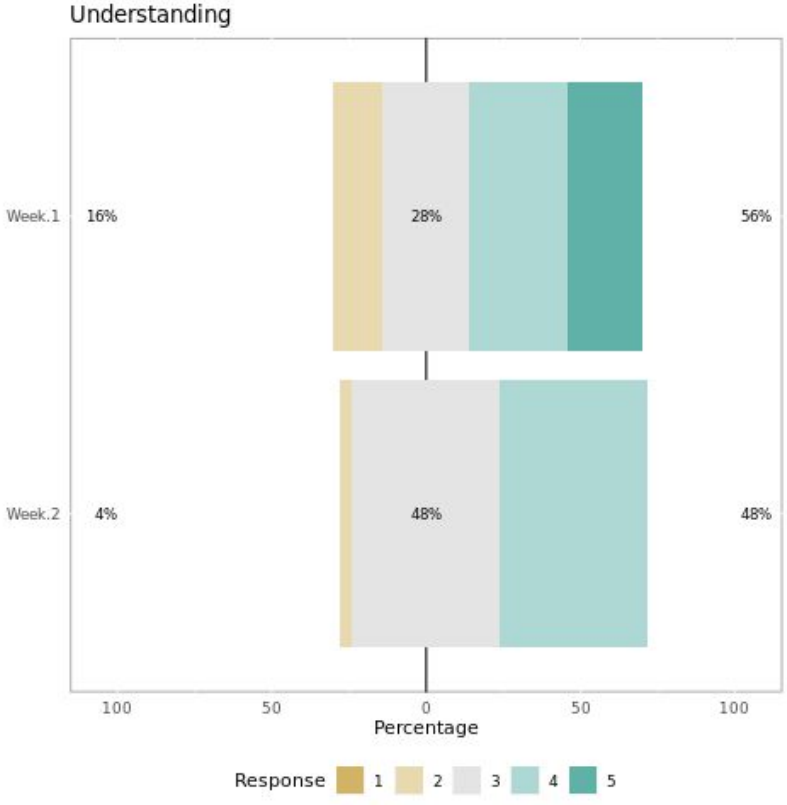
Feedback - Workload



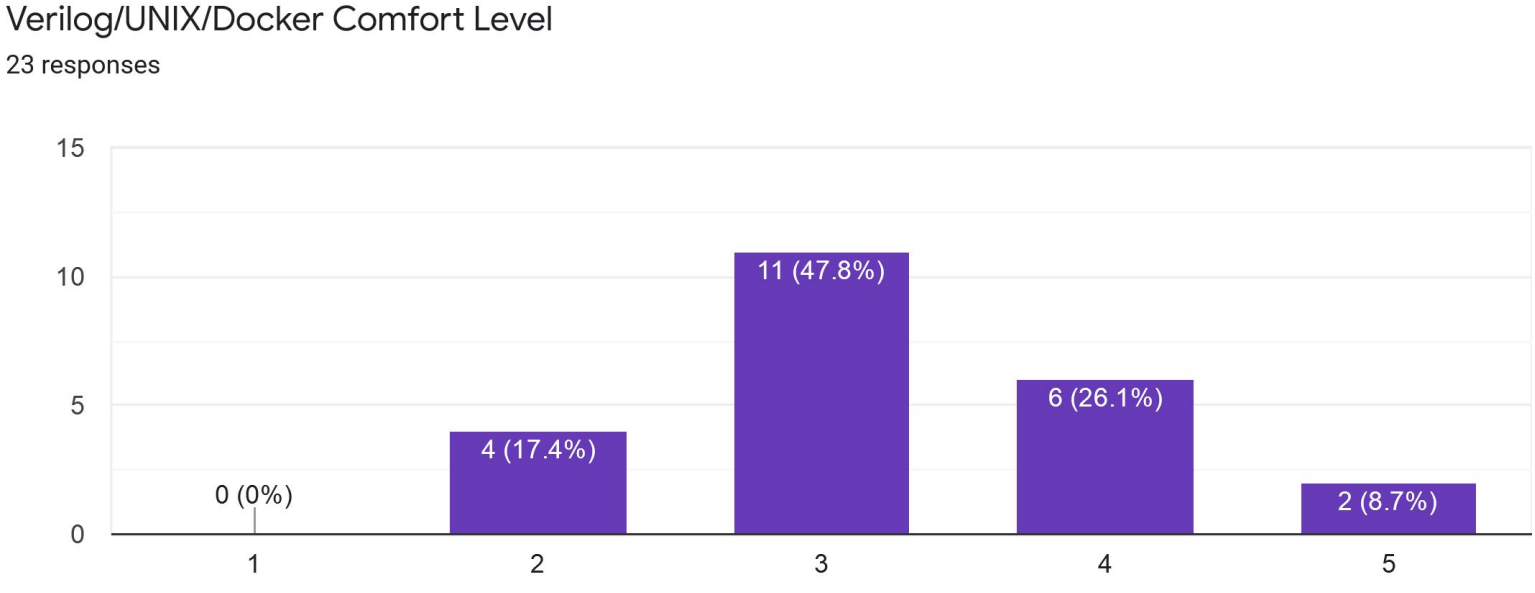
Feedback - Pace



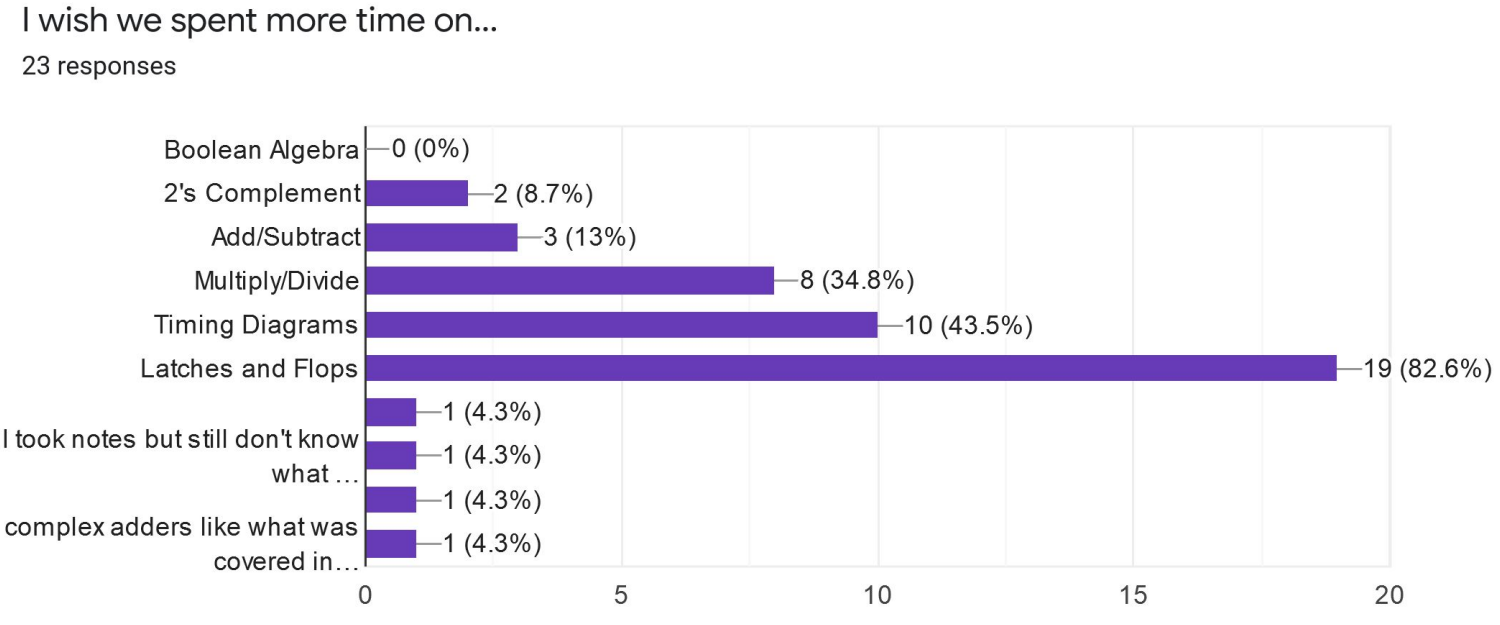
Feedback - Understanding



Feedback - Unix



Feedback - More Time On?



Feedback - Communication

Justified Expressions of Frustration

- I’m having issues with the lab.
- I have been debugging and can’t figure it out.
- I’m stuck!

Actionable Requests for Help

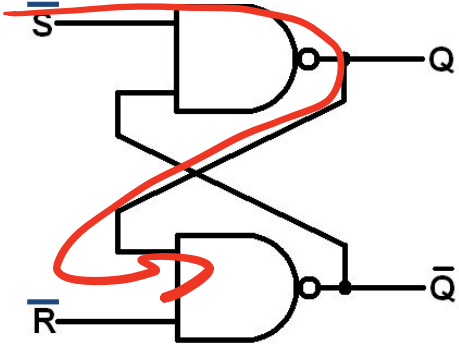
- My full and half adder are working, but my ripple carry adder testbench reports failures. I don’t understand why!
- iverilog gives me <X> error message!?

Feedback - Final Project

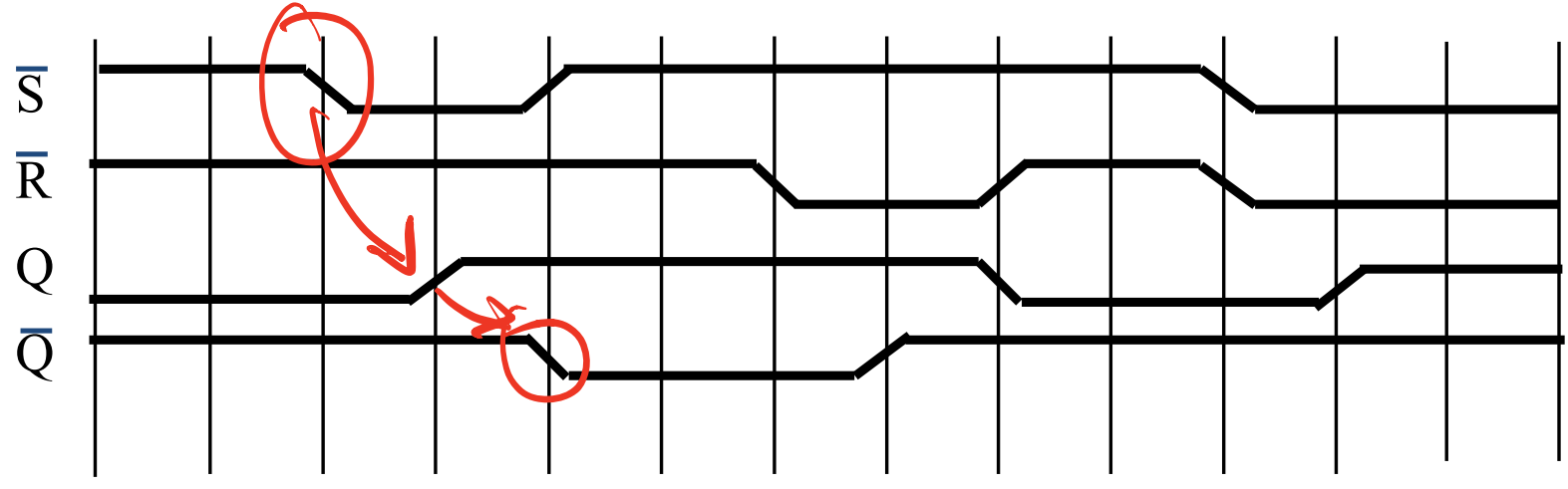
Scheduling Question

Lab Discussion

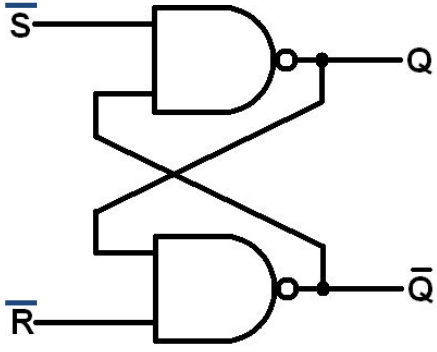
Review



$\overline{S}$	$\overline{R}$	Q
0	0	:(
0	1	1
1	0	0
1	1	Q

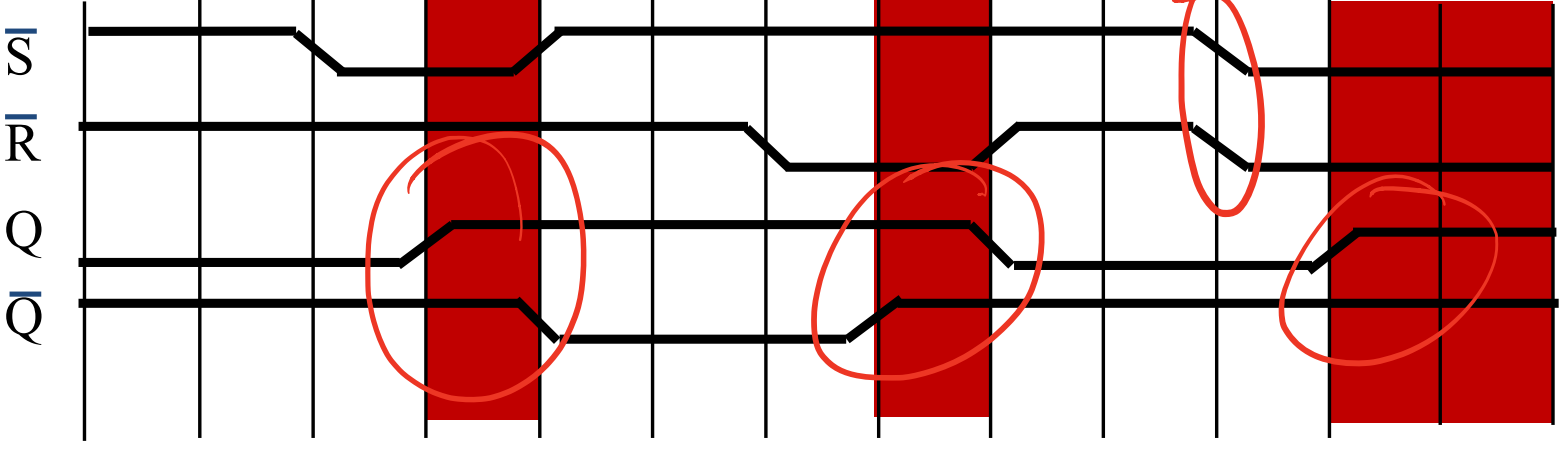


Review

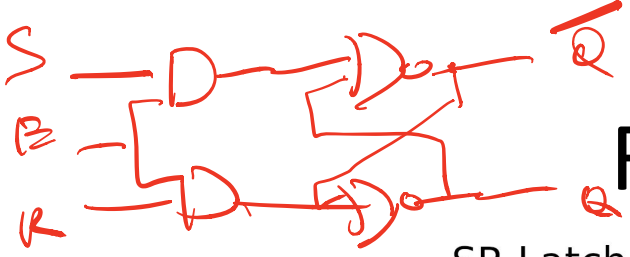


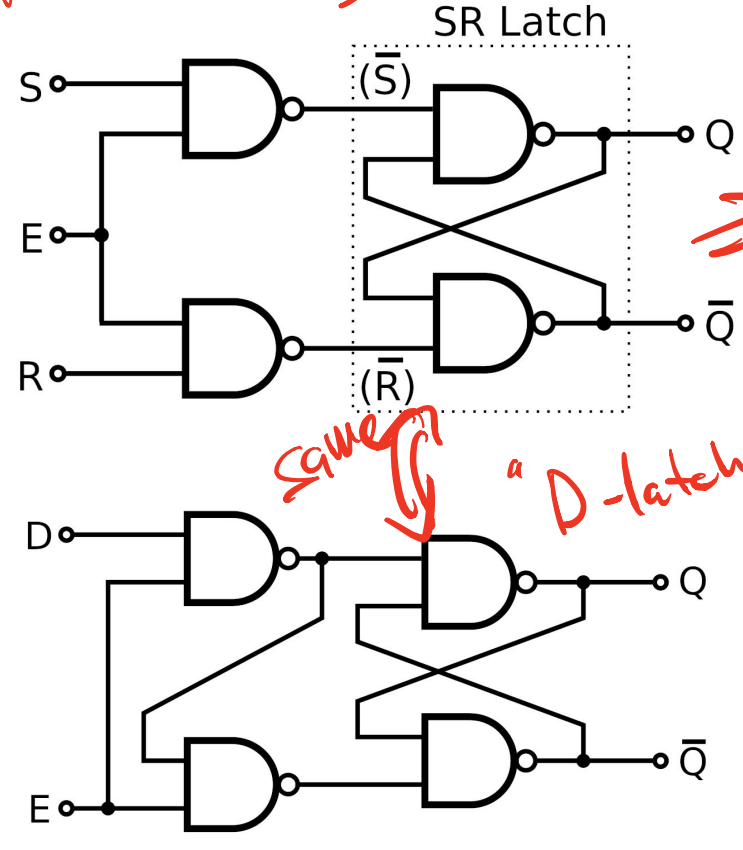
$\overline{S}$	$\overline{R}$	Q
0	0	:(
0	1	1
1	0	0
1	1	Q

illegal

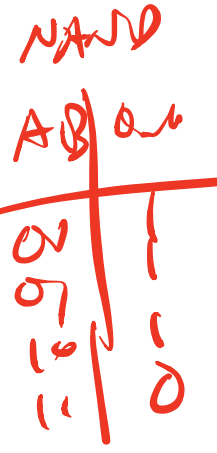


Review





S	R	E	Q_{next}
0	0	1	Q_{curr}
0	1	1	0
1	0	1	1
1	1	1	:(
X	X	0	Q_{curr}



D	E	Q_{next}
0	0	Q_{curr}
1	0	Q_{curr}
0	1	0
1	1	1

Today

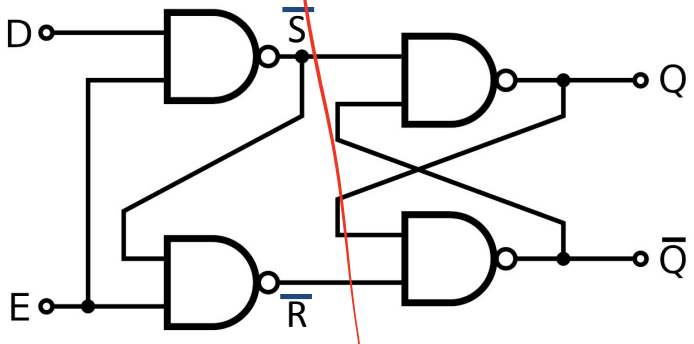
Finish up State Holding Elements

Look at Examples of Design Metrics

How to evaluate said metrics

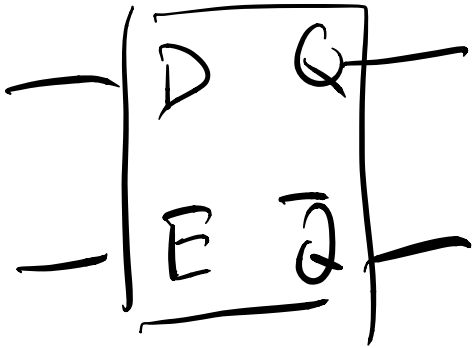
Digestible examples, not memorization fodder

Data “D” Latch

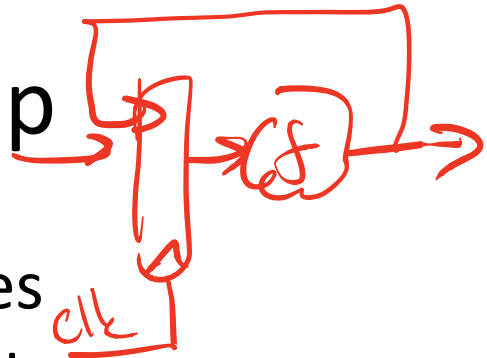


D	E	Q_{next}
0	0	Q_{curr}
1	0	Q_{curr}
0	1	0
1	1	1

D	E	$\bar{S}$	$\bar{R}$
X	0	1	1
0	1	1	0
1	1	0	1

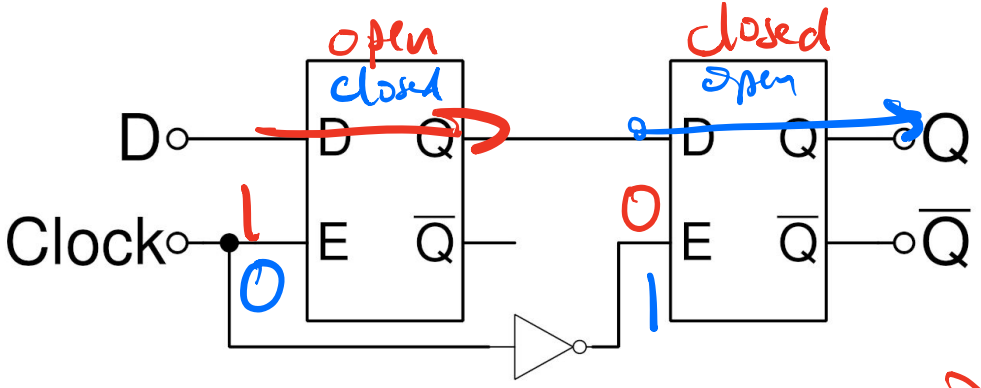


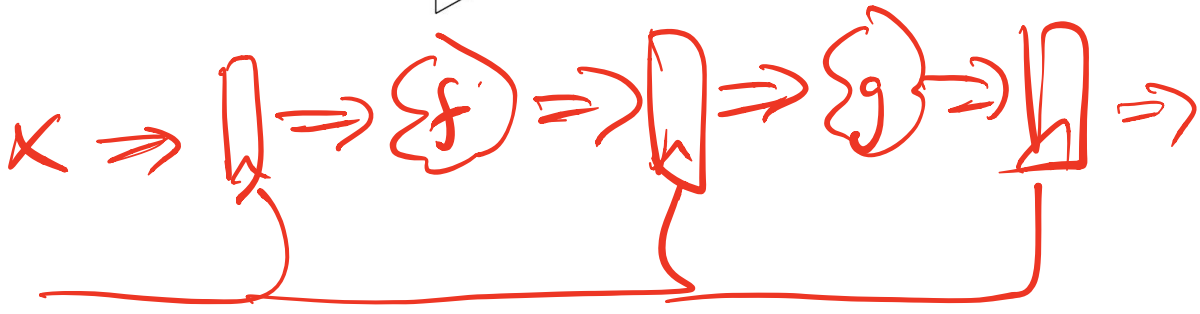
Data "D" Flip Flop



g(f(x))

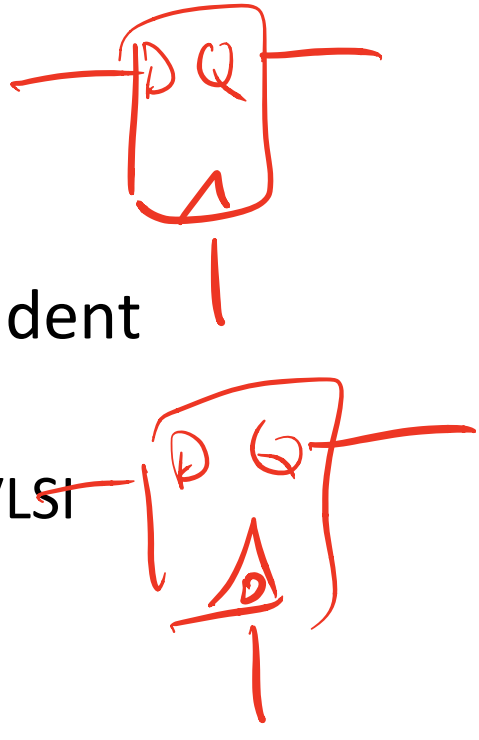
Two D-Latches in Series
Opposite Enable Polarity





Edge Triggered D-Flip Flop

- Functionally similar to prior dual latch flip flop
- Edge Triggered, not Pulse
- Implementation process dependent
  - Can be roughly three SR Latches
  - Can be dynamic logic – learn in VLSI

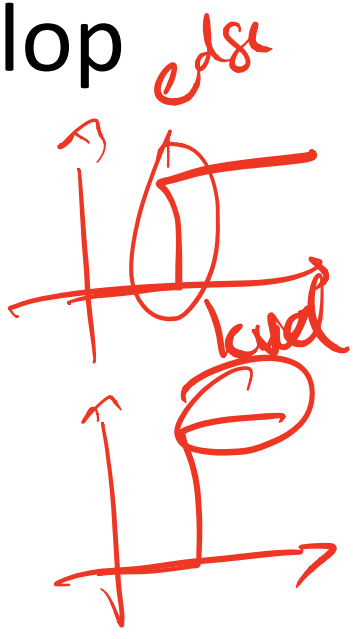


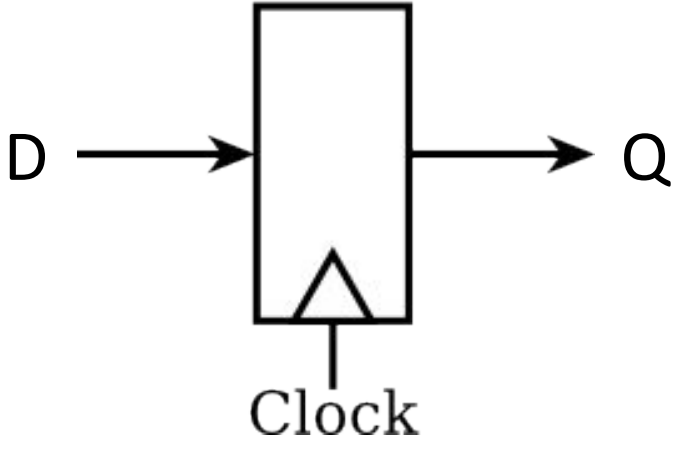
Edge Triggered D-Flip Flop

- Always has:
  - Clk, D, Q

Clk	D	Q
↑	0	0
↑	1	1
Other	X	Q

Rising edge

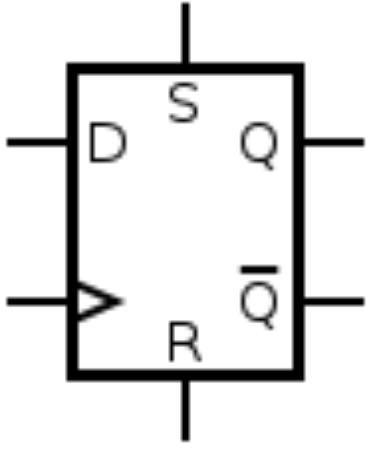




Edge Triggered D-Flip Flop

- Always has:
  - Clk, D, Q

Clk	D	Q
↑	0	0
↑	1	1
Other	X	Q



- May also have:
  - S, R, ~Q
  - **S**et and **R**eset are asynchronous (ignore clock)
    - Can be used to define initial conditions (e.g. at boot)

http://en.wikipedia.org/wiki/File:D-Type_Flip-flop.svg

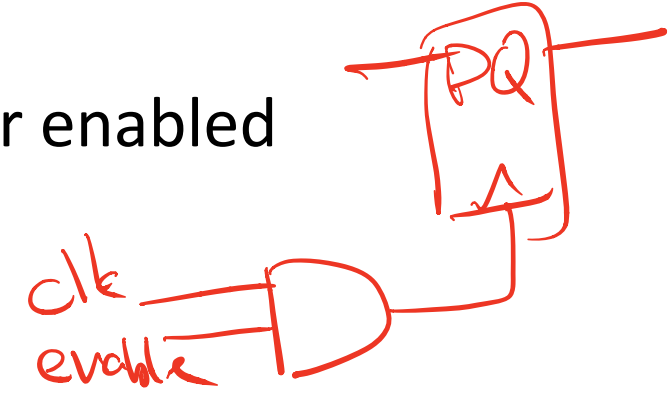
Reminder

Lots of different Flop/Latch types

Each type has little idiosyncrasies

Big Picture - Hold State, Time Matters

Can be clocked or enabled



Common Uses

- In between processing stages
- “Debounce” inputs
  - Hide external noise / uncertainty from the inputs
- Synchronization

Design Evaluation

Area

Delay/Performance

Energy/Power - Next Time

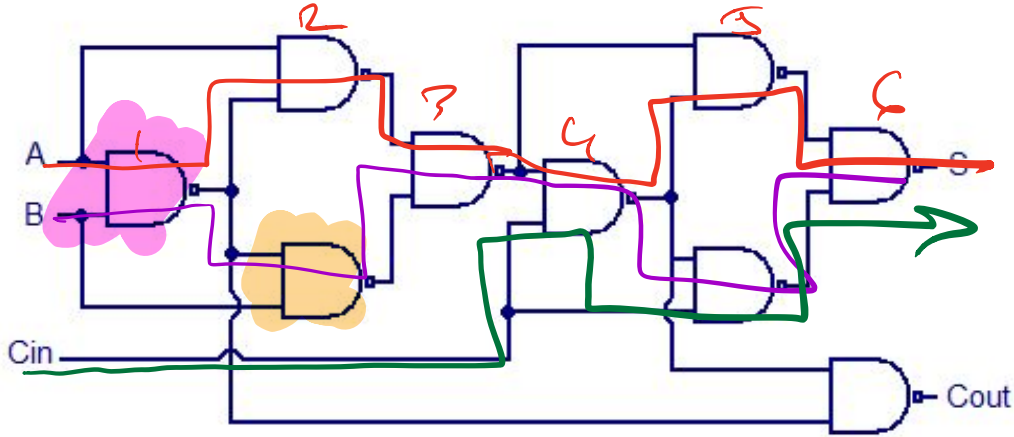
Full Adder

Calculate Propagation Delays

Multiple Paths? Choose Worst

	A	B	Cin
Sum	6	6	3
Cout	5	5	2

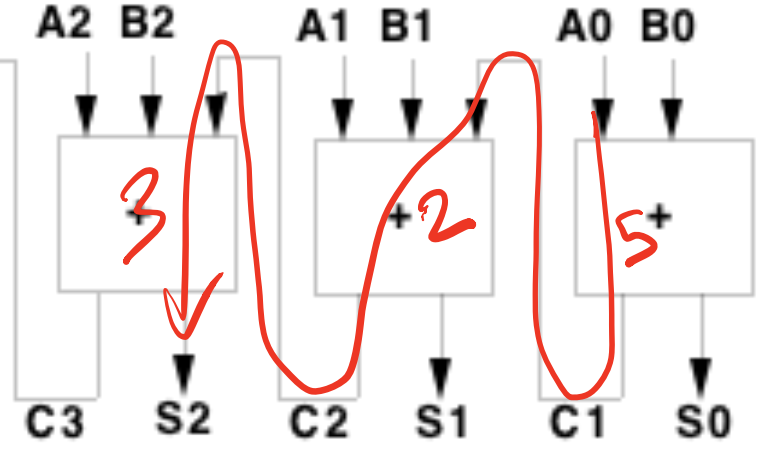
A	B	CI	CO	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



Full adder using NAND logic

www.circuitstoday.com

Multi-Bit Addition

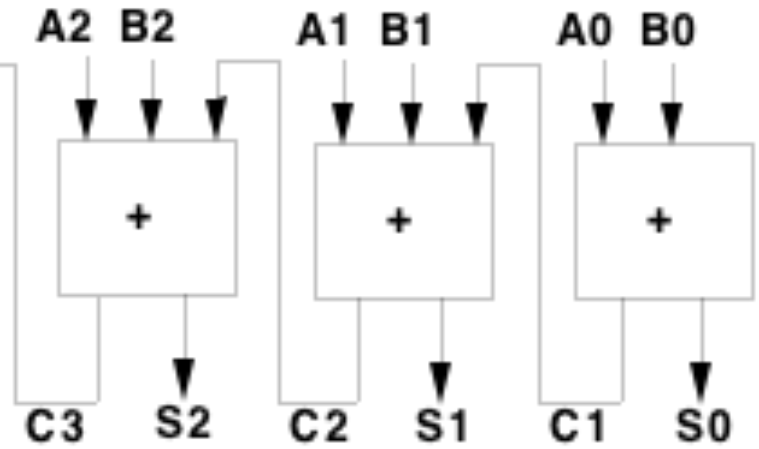


	A	B	Cin
Sum	6	6	3
Cout	5	5	2

Using previous slides' design, find worst prop delay

	A/B0	A/B1	A/B2
S0	6	–	–
C1	5	–	–
S1	5+3	6	–
C2	5+2	5	–
S2	5+2+3	5+3	6
C3	5+2+2	5+2	5

Multi-Bit Addition



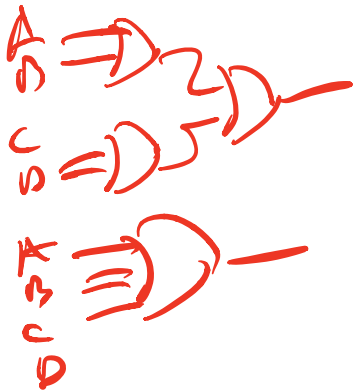
	A	B	Cin
Sum	6	6	3
Cout	5	5	2

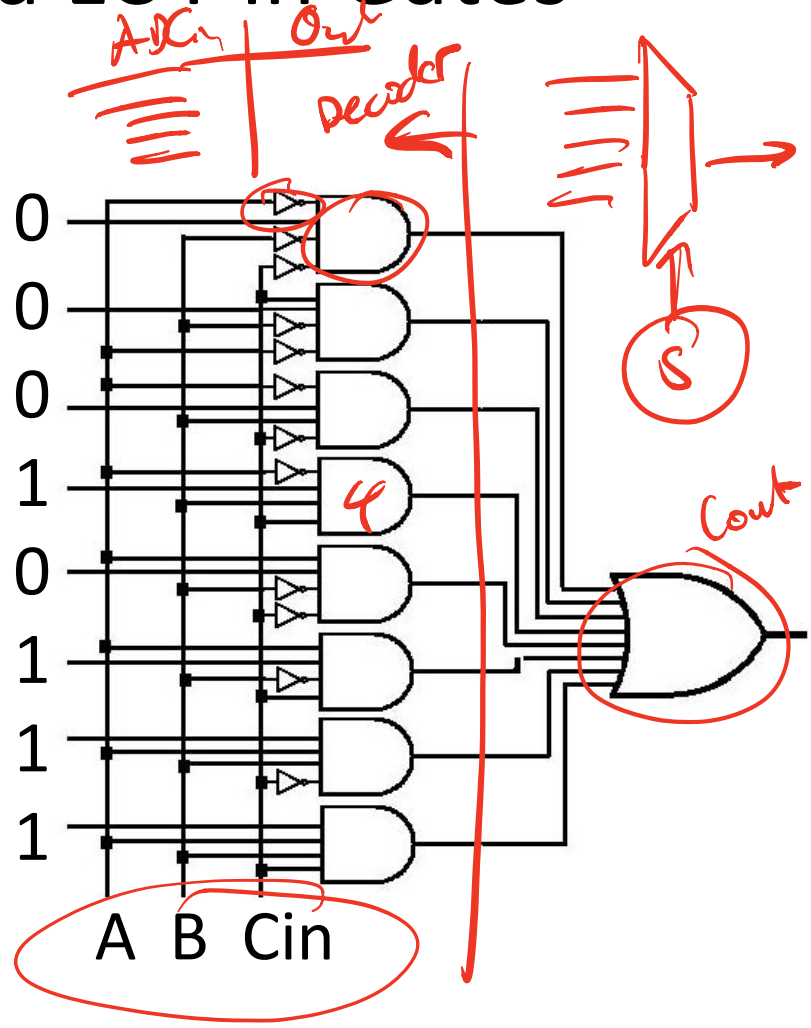
Using previous slides' design, find worst prop delay

	A/B0	A/B1	A/B2
S0	6	-	-
C1	5	-	-
S1	5+3	6	-
C2	5+2	5	-
S2	5+2+3	5+3	6
C3	5+2+2	5+2	5

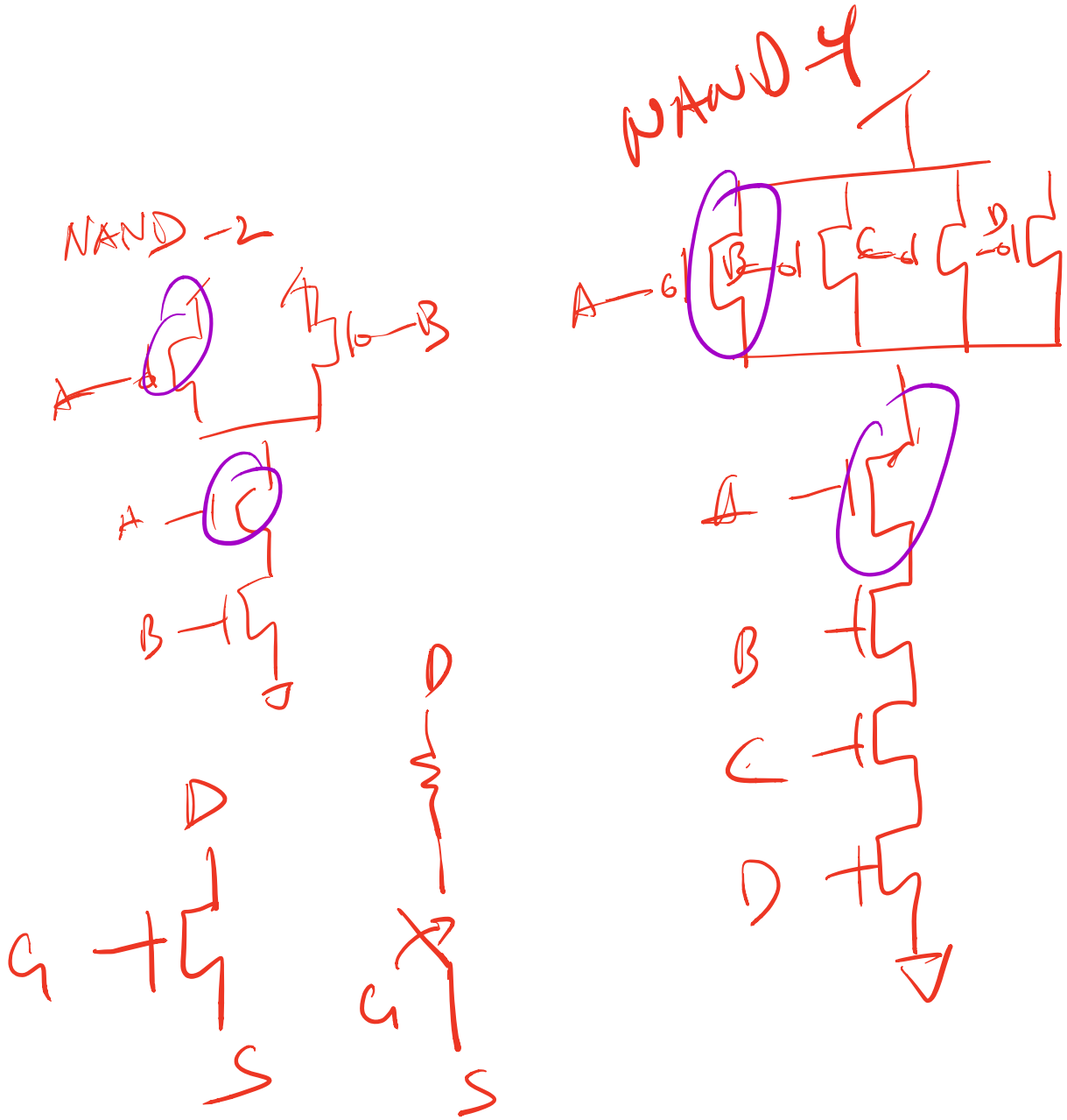
Full Adder as a LUT in Gates

- Only half, slide too small to show Sum
- Area Cost?
- Speed?

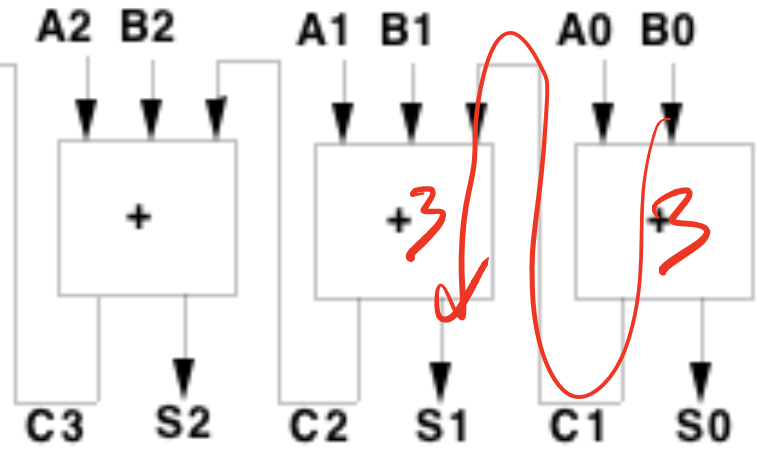




	A	B	Cin
Sum			
Cout			


NAND-2
A
B
A
B
D
G
S
D
G
S
NAND 4
A
B
C
D
A
B
C
D

Timing with Alternate Topology



	A	B	Cin
Sum	3	3	3
Cout	3	3	3

	A/B0	A/B1	A/B2
S0	3	-	-
C1	3	-	-
S1	3+3	3	-
C2	3+3	3	-
S2	3+3+3	3+3	3
C3	3+3+3	3+3	3

Using new prop delays, find total delay

Growth Rates

- What changes as a specific design ‘grows’
  - Propagation delay
  - Silicon area
- Propagation Delay for an N bit adder?
  - Design 1: NAND Gates
  - Design 2: Small LUT

Growth Rates

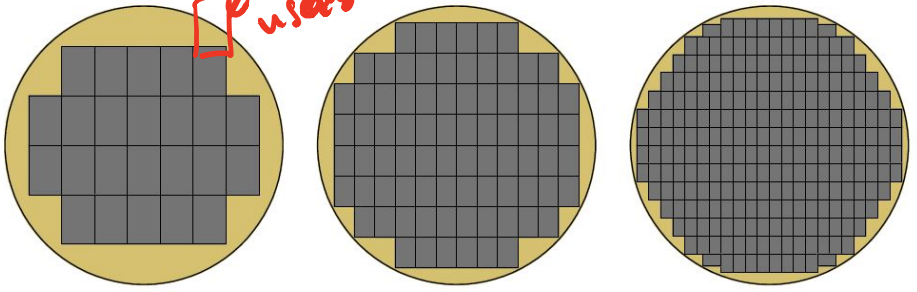
- What changes as a specific design ‘grows’
  - Propagation delay
  - Silicon area

- Propagation Delay for an N bit adder?
  - Design 1: 2N + 5
  - Design 2: 3N

- Which design is faster? Larger?

Size in Silicon

- Physical Area drives manufacturing cost
  - # Chips per silicon wafer

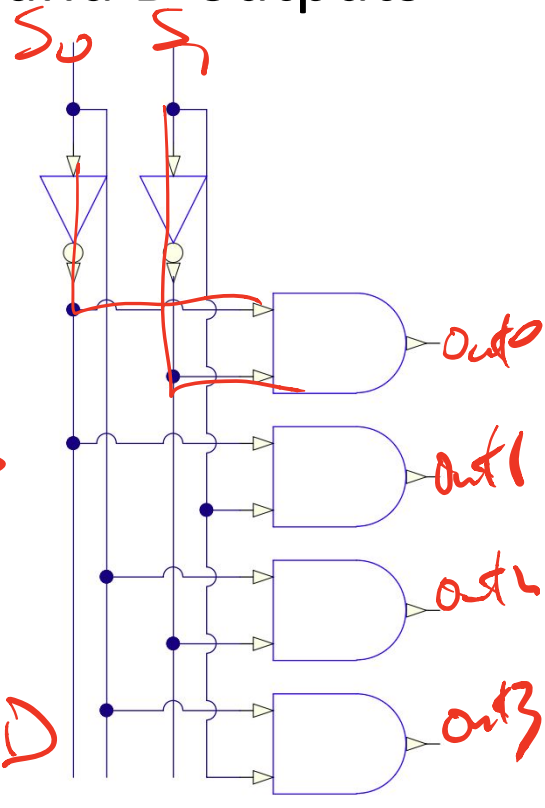


- Number of inputs drives gate size
  - Two transistors per input
  - Slightly worse than linear growth
- Total Gate Inputs □ Rough Cost Estimate

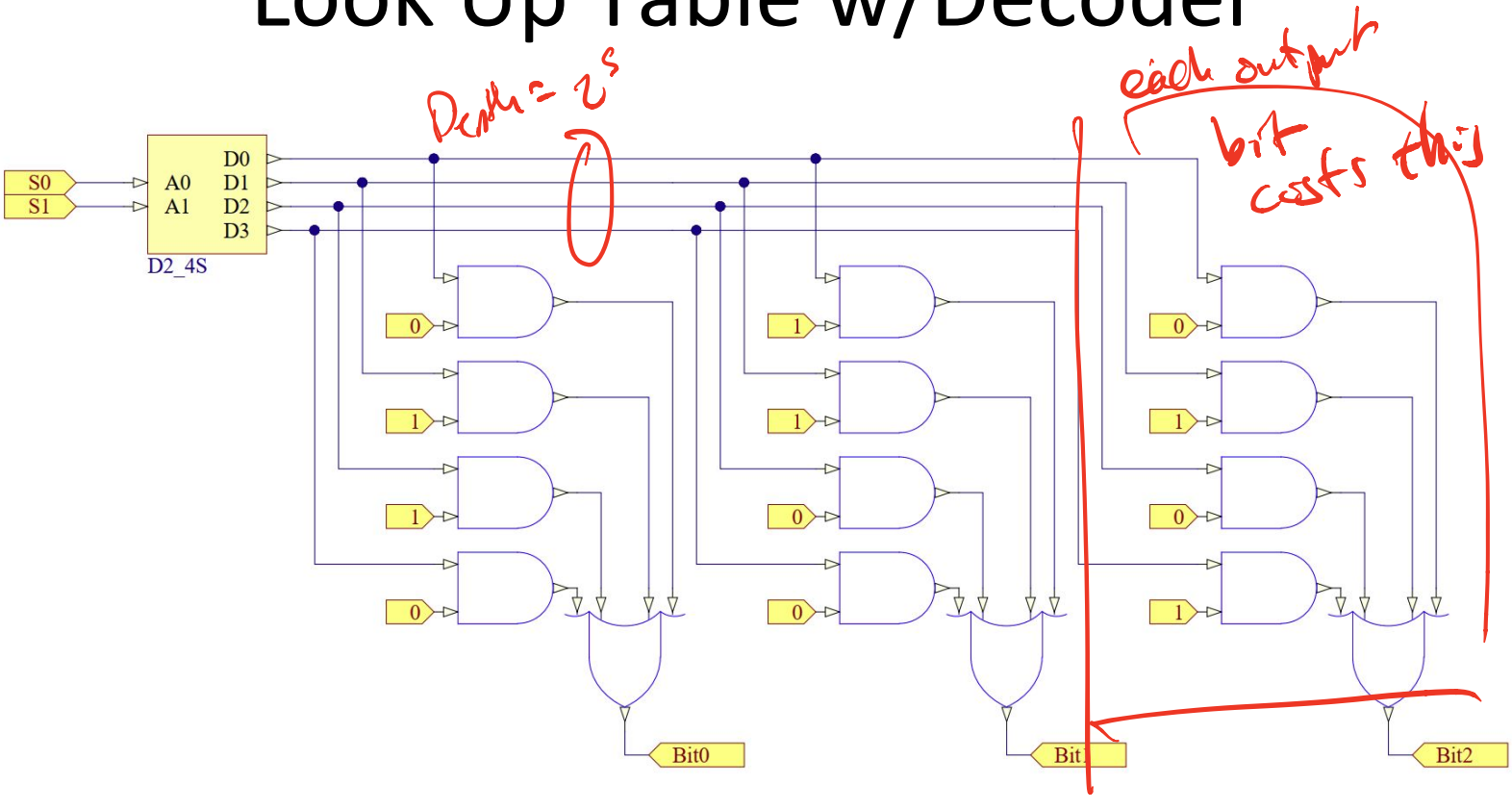
https://photo.stackexchange.com/questions/53826/why-does-increasing-sensor-size-necessarily-lead-to-lower-silicon-wafer-utilizat

Decoder Growth

- For a decoder with **S** select bits and **D** outputs
  - **D** = 2^S
- Need
  - **S** Inverters
  - **D** AND Gates with **S** inputs
- Total Size:

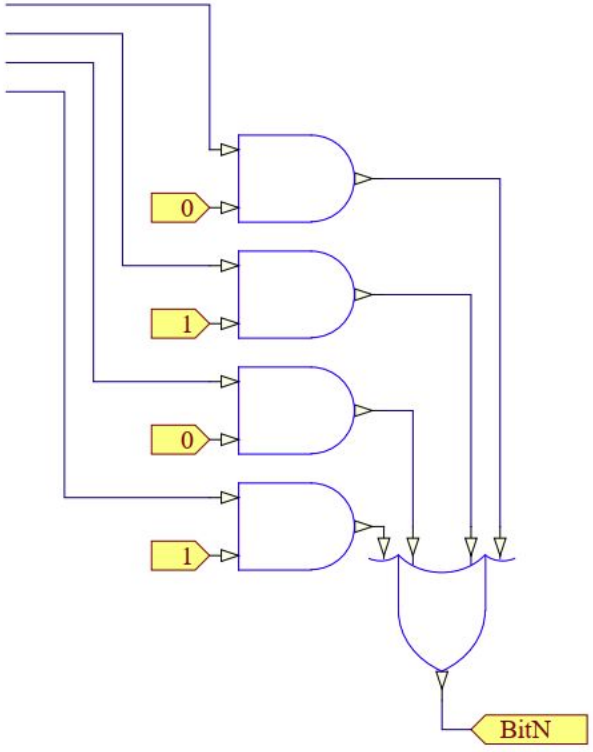


Look Up Table w/Decoder



Growth Characteristics

- This has 12 Gate Inputs
  - 4 AND2 (8)
  - 1 OR4 (4)
- Per decode line:
  - 1 AND2
  - 1 input of the OR
- (#Decode Lines)*3



LUT Growth Review

- For a LUT with Depth **D** and Output Width **W**
- Need
  - 1 Decoder
  - **W** Output Structures
- Total Size:

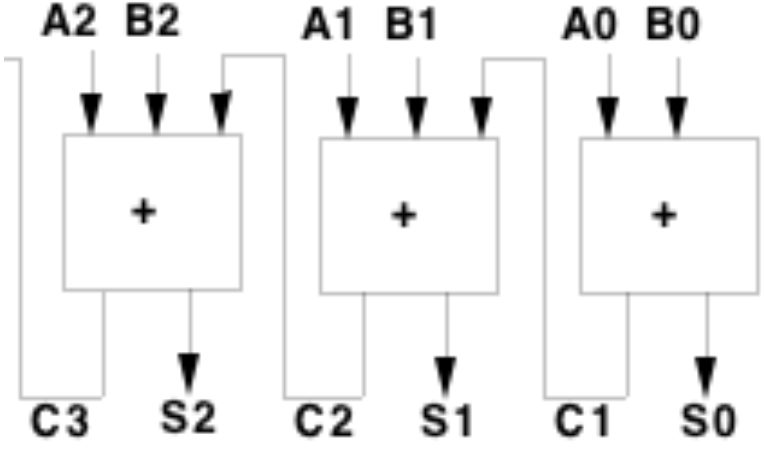
LUT Growth Review

- For a LUT with Depth **D** and Output Width **W**
- Need
  - 1 Decoder (**D**+1)log2(**D**)
  - **W** Output Structures **W**(3**D**)

- Total Size: (**D**+1)log2(**D**) + **W**(3**D**)

Single stage adder

- Returning to our adder example, how does the size of each implementation scale?



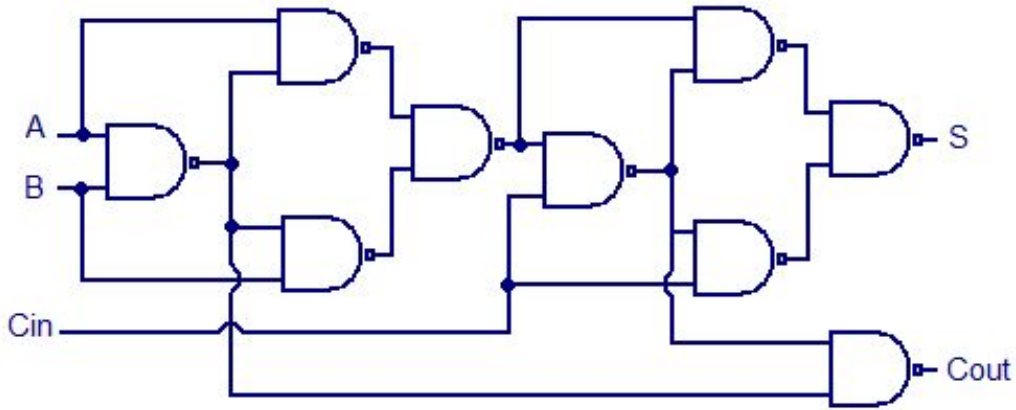
Full Adder

Calculate Propagation Delays

Multiple Paths? Choose Worst

	A	B	Cin
Sum	6	6	3
Cout	5	5	2

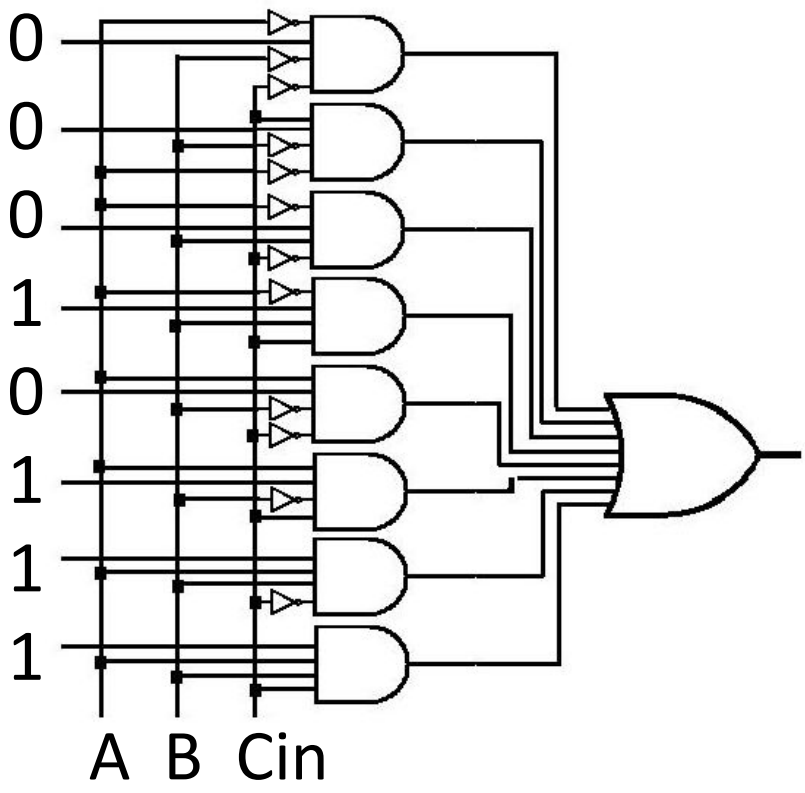
A	B	CI	CO	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



Full adder using NAND logic www.circuitstoday.com

Full Adder as a LUT in Gates

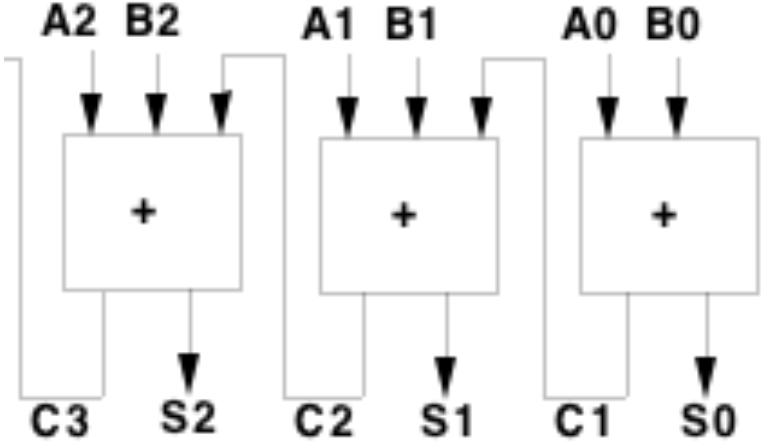
- Only shows Cout, need another LUT for Sum
- How to build a LUT



	A	B	Cin
Sum	3	3	3
Cout	3	3	**3**

Single stage adder

- Let's eliminate propagation delay by implementing an adder as large single LUT…



Adder Comparison

LUT for 32bit + 32bit = 32bit addition

- S = (32+32) = 64
- W = 32
- $D = 2^S = 18446744073709551616$

Total Size = $(D+1)\log_2(D) + W(3D)$

$295147905179352825862 4 \approx 3 \times 10^{21}$

Adder Comparison

- LUT for 32bit + 32bit = 32bit addition
  - Size $\approx 3 * 10^{21}$
- Chained Full Adder
  - 9 NAND2 Gates per bit
  - Size = 32 * 18 = 576
- Apollo Guidance Computer
  - 4100 NOR3

Better Gate Delay Estimation

- Model gate delay as proportional to the number of inputs in the gate
- Basic gates: NAND, NOR, NOT
- AND, OR, etc have “hidden” extra inverters
- e.g. for unit delay of 10:
  - 8-input NOR delay = 80
  - 8-input OR delay = 90 (extra inverter)