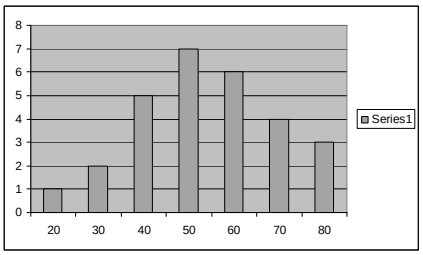
28 September 2004

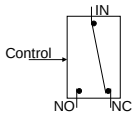
- Questions?
- Test Review
- Chapter 4 – Logic Gates
- Read in Chapter 4 pages 250-258, 265-274, section 4.8 through top of page 288, section 4.10

Grade Distribution



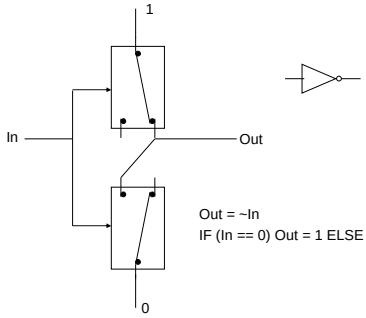
Mean = 60, Min = 21, Max = 84, Standard Deviation = 17

Ideal Switch



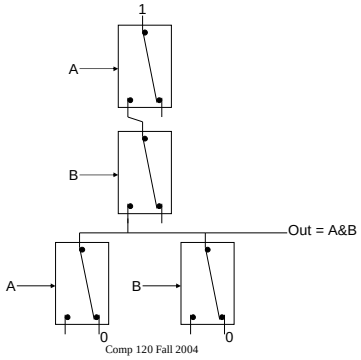
When Control == 0
NC == IN (NC is connected to IN)
When Control == 1
NO == IN (NO is connected to IN)

Inverter

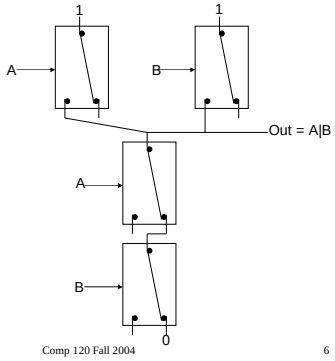


Out = ~In
IF (In == 0) Out = 1 ELSE Out = 0

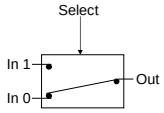
AND



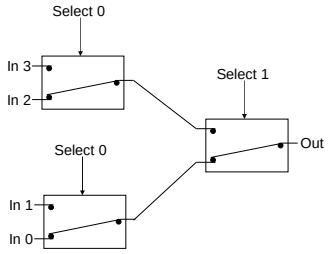
OR



2 Input MUX

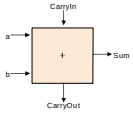


4-input MUX



1-bit ALU for Addition

- Let's look at implementing out 1-bit adder with gates



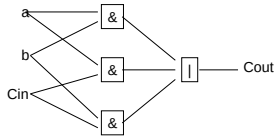
$c_{out} = a\&b \mid a\&c_{in} \mid b\&c_{in}$

$sum = a\&\sim b\&\sim c_{in} \mid \sim a\&b\&\sim c_{in} \mid \sim a\&\sim b\&c_{in} \mid a\&b\&c_{in}$

a	b	cin	sum	cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

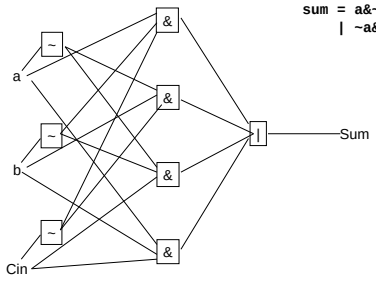
Cout in Gates

$c_{out} = a\&b \mid a\&c_{in} \mid b\&c_{in}$



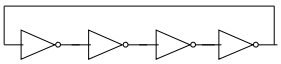
Sum in Gates

$sum = a\&\sim b\&\sim c_{in} \mid \sim a\&b\&\sim c_{in} \mid \sim a\&\sim b\&c_{in} \mid a\&b\&c_{in}$

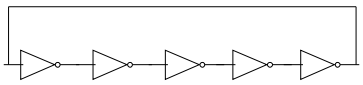


How Fast?

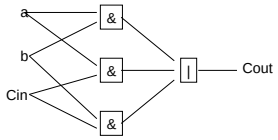
What does this do?



How about this?

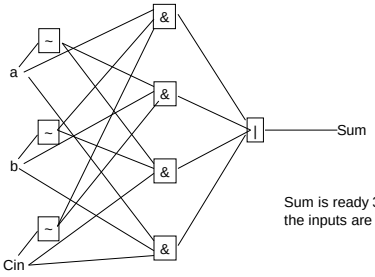


How many gate delays for Cout?



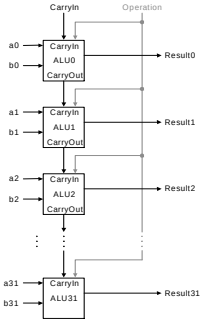
Cout is ready 2 gate delays after the inputs are ready

How many gate delays for Sum?



Sum is ready 3 gate delays after the inputs are ready

How slow can you go?



Cin for ALU31 is ready 62 gate delays after the initial input was ready

Result31 will be ready 3 gate delays later for a total of 65 gate delays

If a gate delay is 1ns what is the fastest clock rate for addition to happen in 1 cycle?

What about AND?