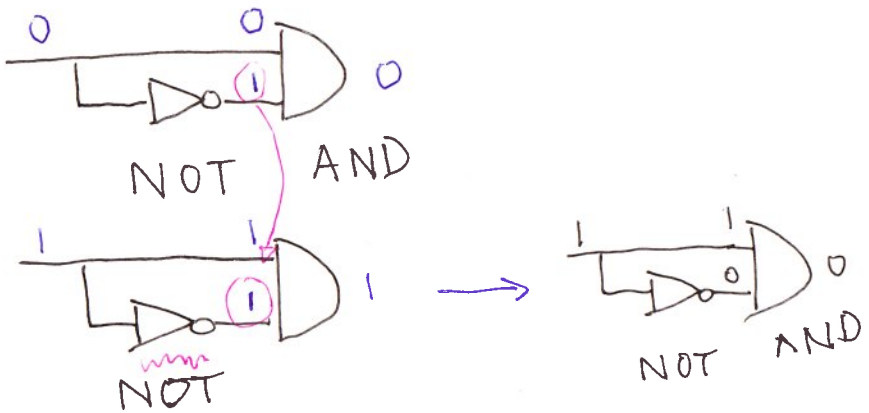
Set	Topic
A	edge trigger
B	bit shifter
C	1-bit ALU (Thursday?)
D	mux word problem
E	½ and full adders

(RISING) EDGE TRIGGER



During delay of 1 gate of time, it still outputs a 1

↳ occurs when going from 0 → 1

For the amount of time it takes to go through the NOT gate is the amount of time it "buffer lags" at 1

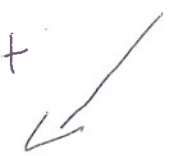
Merina Leong
Cody Clattenburg

Bit Shifters

Found in microprocessors used to perform accurate multiplication and division

Example. Let's look at 6_{10} in BINARY

0 1 1 0



Left Shift

Right Shift

1 1 0 0

IN DECIMAL 12_{10}

As we shift LEFT we

Multiply by stored base

IN this case $6 \times 2 = 12$

Binary = base 2

0 0 1 1

IN DECIMAL 3_{10}

As we shift RIGHT we

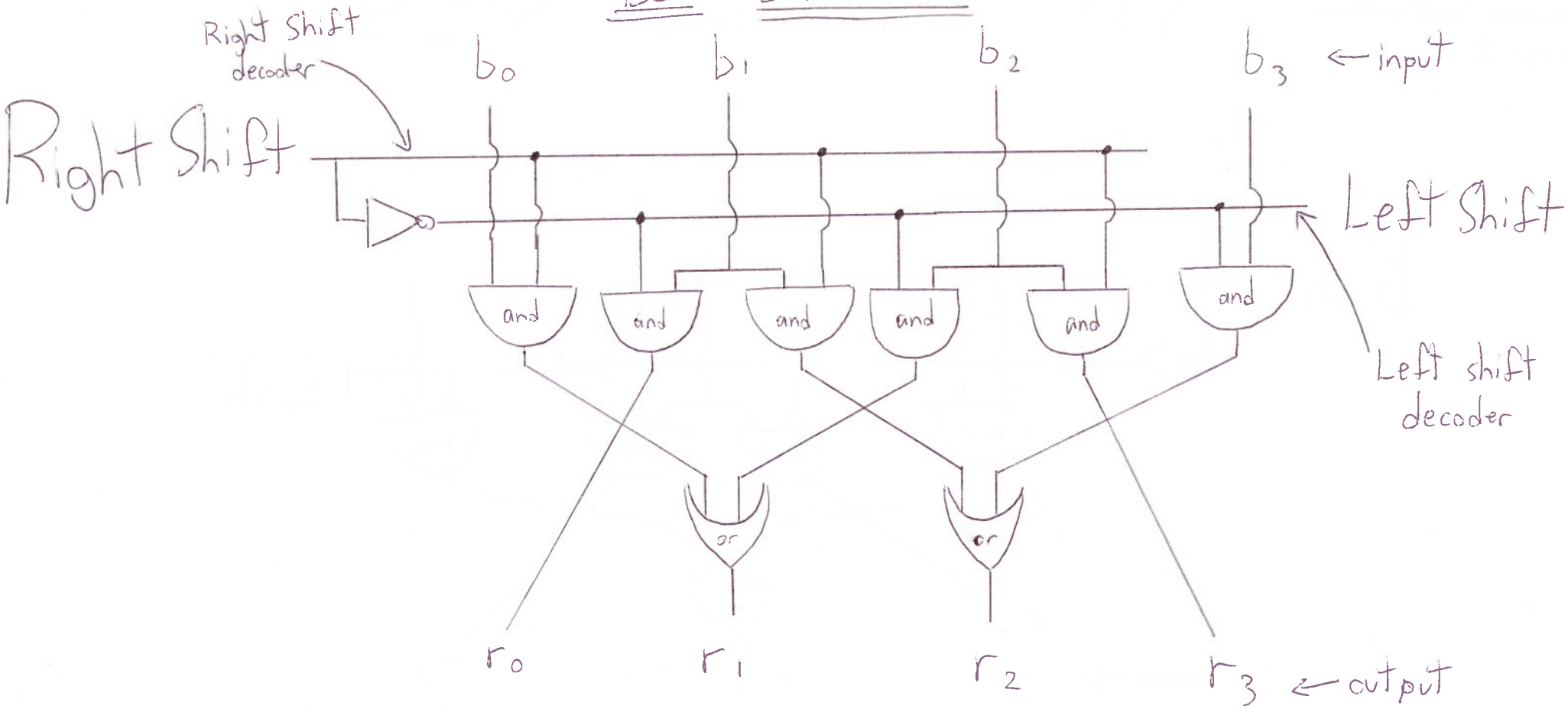
Divide by stored base

IN this case $6 / 2 = 3$

Binary = base 2

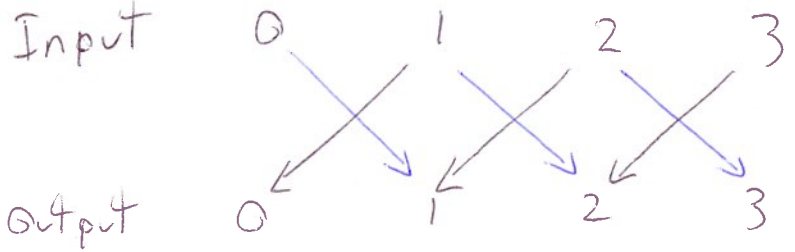
(Chill) Cameron Warje
YuQing Qiu
Li-Yun Tong

BIT SHIFTERS



Want to left shift? Make left shift decoder 1, and the right shift decoder 0. Follow the lines with the correct decoder value and your input bits, and you'll get the shift you want.

Dumbed down...

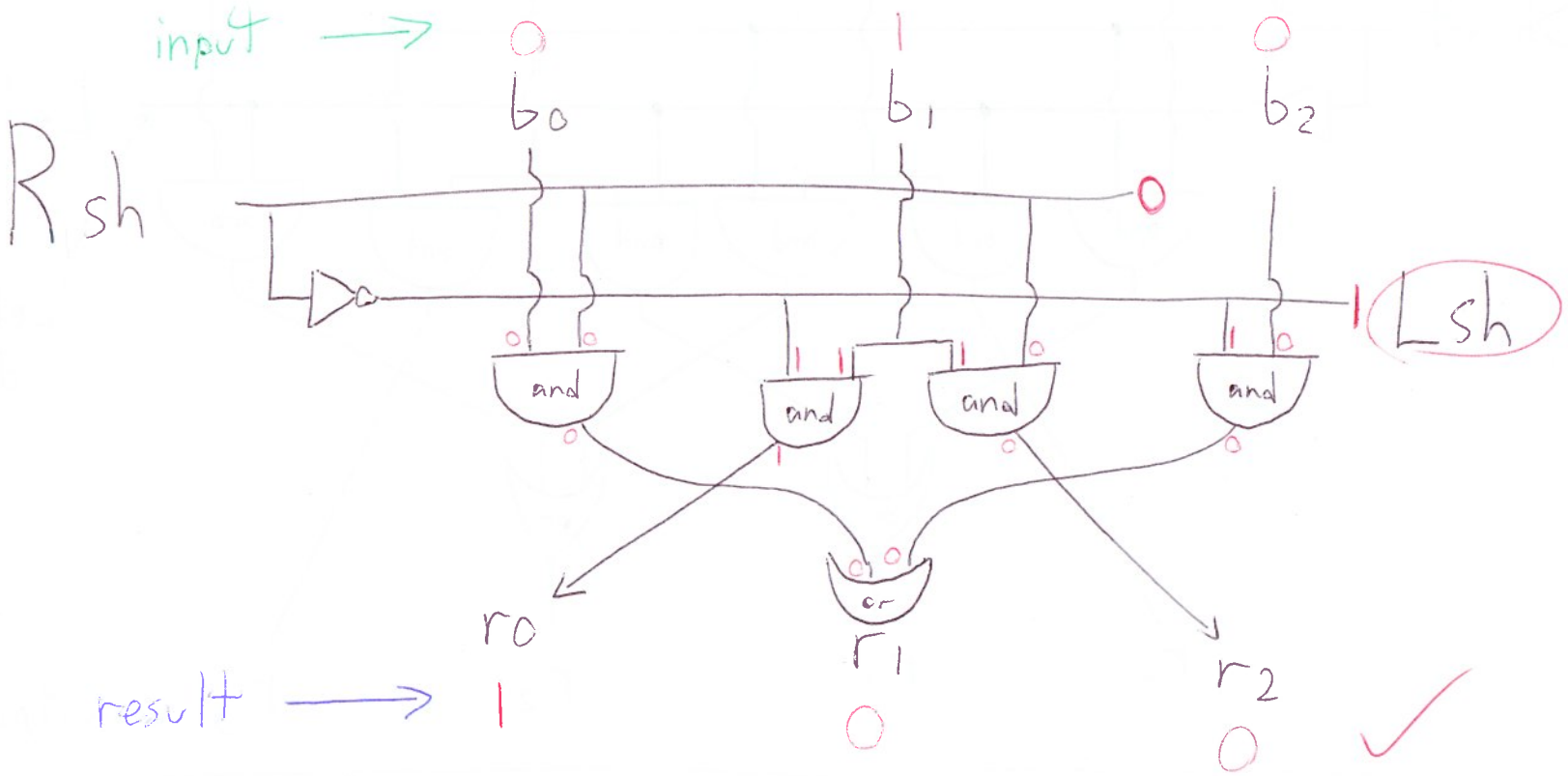


Li, Yuqing, Cameron

BIT SHIFTERS

page 2

3 bit left shift example. $010_2 \rightarrow 100_2$ ($2_{10} \rightarrow 4_{10}$)

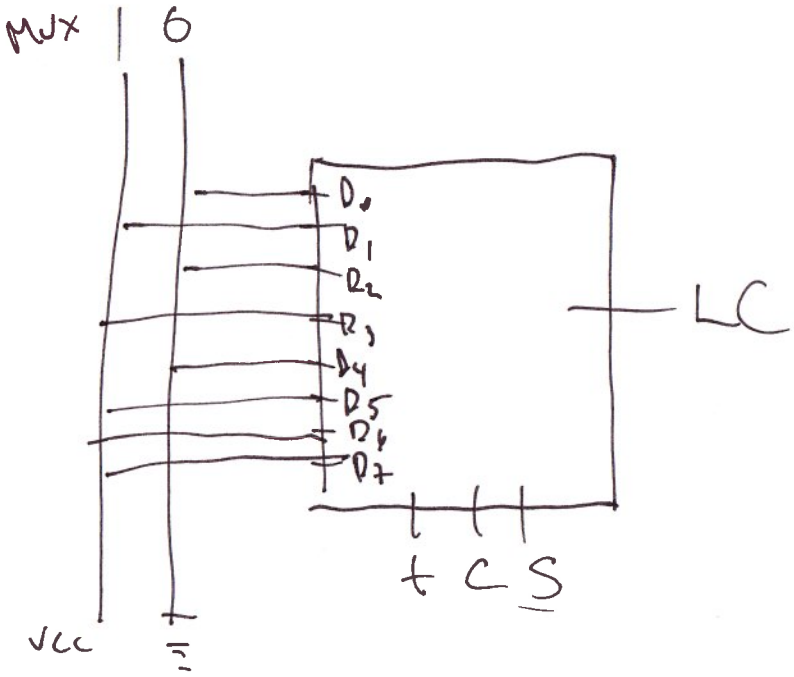


~~5a~~

Java Lab will be cancelled if

- tomorrow with Chris
- Snowstorm

	t	c	s	LC
D_0	0	0	0	0 ←
D_1	0	0	1	
D_2	0	1	0	0
D_3	0	1	1	1 ←
D_4	1	0	0	0
D_5	1	0	1	1 ←
D_6	1	1	0	1 ←
D_7	1	1	1	1 ←



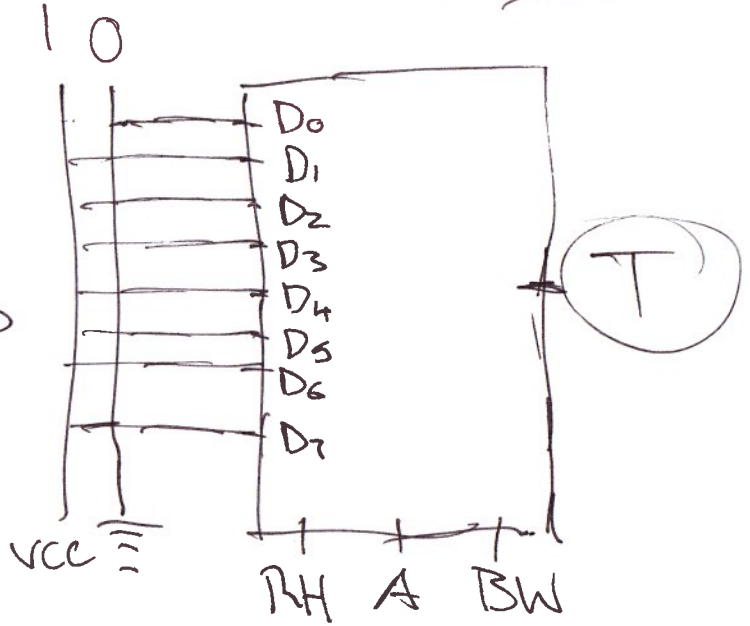
$SoP = t'c's + t'cs + tc's + tcs' + tcs$

black box mux

John + Sam

At Rush hour you'll get stuck in traff or you'll get stuck if theres been an accident or Bad weather

RH	A	BW	Traffic
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1



John & Sam

¬a $\bar{a}$

Test Half Adder

a	b	Sum	Cout
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

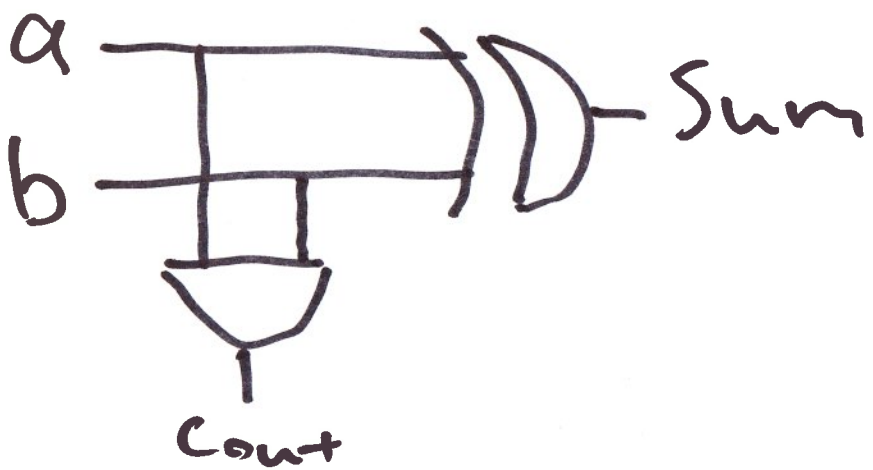
Sum
CSOP: a'b + ab'
↓
XOR

Cout
CSOP: ab
↓
AND

1 1
+1
2

1 1
+1
7

10

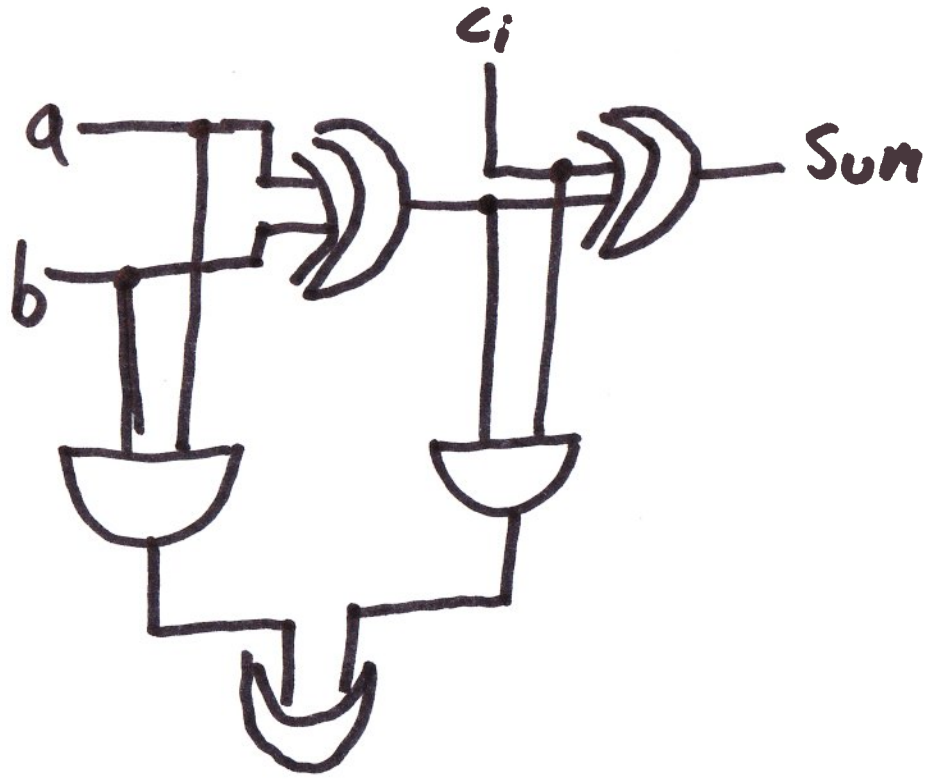


Set 2E
Wei Zhang
Kerry Regan

Full Adder

- Adds together binary numbers and carry-in values from previous stage. This produces two bits of output; The sum, and the carry-out

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



SOP

sum | $a'b'c + a'bc' + ab'c' + abc$

cout | $a'bc + ab'c + abc' + abc$