PASS- Computer Systems

Week 3

1. Binary Representation
   a. Signed Vs Unsigned Binary

	Negative? Positive?	Smallest number can be handled in k-bit binary number	Largest number can be handled in k-bit binary number
Signed Binary (2's complement)	Yes	-2^{k-1}	$2^{k-1} - 1$
Unsigned Binary	only pos	0	$2^{k} - 1$

   b. answer questions in decimal form

The most negative 8-bit binary number with 2's complement is	$-2^{7} = -128$
The most positive 16-bit binary number with 2's complement is	$2^{15} - 1 = 32767$
The largest negative 32-bit binary number with 2's complement is (ie. 1111 1111 1111 1111 1111 1111 1111 1111)	-2^{31}

2. Two's complement conversion
   a. Convert binary to decimal

2's complement	Decimal
0101 1+0+4+0	5
1001 0111 1+2+4	-7
1101 0011 1+2	-3
1010 0101 0110 2+4	-6
1111 0001 2	-1
1111 1111	-1

   b. Convert decimal to binary (8 bits)

2's complement (8-bit)	Decimal
100010000	-112
00000101	5
10000011	-125
11111011	-5

112	
56	0
28	0
14	0
7	0
3	1
1	1
0	1

01110000
10001111
10010000

5	
2	1
1	0
0	1

00000101

125	
62	1
31	0
15	1
7	1
3	1
1	1
0	1

01111101
10000010
10000011

5	
2	1
1	0
0	1

00000101
11111010
11111011

c. How to negate a number (e.g., converting 2 to -2)?

flip bits, add 1

3. Binary Addition

a. Calculate 3+4 as a signed 4-bit number with 2’s complement. Enter the result as a decimal. The answer must start with a + or – sign.

$2^4 = 16$

+7

b. Calculate 30000 (0111010100110000) + 2767 (0000101011001111) as a signed 16-bit number with 2’s complement. Enter the result as a decimal. The answer must start with a + or – sign.

+32767

$2^{16-1} - 1 = 32767$

c. What if the above question is 30000 (0111010100110000) + 2768 (0000101011010000)?

−32768

$> 2^{16-1} - 1$

d. Calculate (-4) + (-6) as a signed 4-bit number with 2’s complement. Enter the result as a decimal. The answer must start with a + or – sign

−10

-2^3

$= -8$

$2^3 - 1$

7

= 6

e. Calculate 5+10 as a signed 4-bit number with 2’s complement. Enter the result as a decimal. The answer must start with a + or – sign

$2^3 - 1$ -2^3

7 $= -8$

$\boxed{= -1}$

4. Sequential logic vs Combinational logic

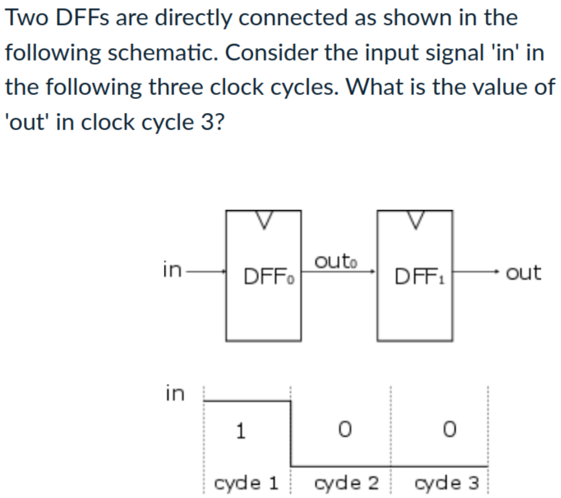
	Does it operate on data?	Does it operate on clock signal?	Does the output depend on past input or present input?	Does it have state (memory)?
Sequential logic				
Combinational logic				

?

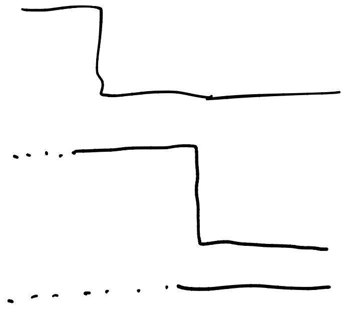
5. DFF
   - What is DFF?

Data Flip Flop, takes an input and outputs at the next cycle

- What is the limit of DFF? (Why can't we use DFF as a register?)

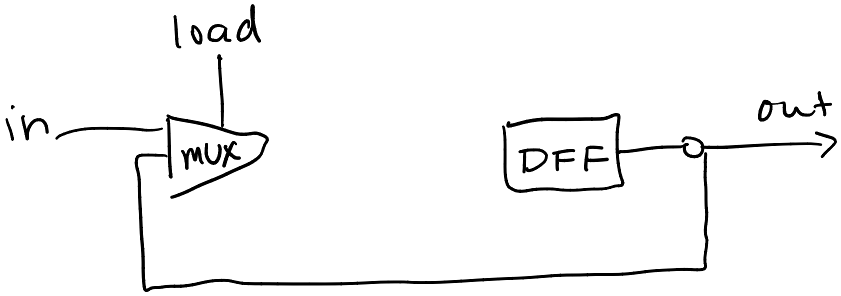


Two DFFs are directly connected as shown in the following schematic. Consider the input signal 'in' in the following three clock cycles. What is the value of 'out' in clock cycle 3?

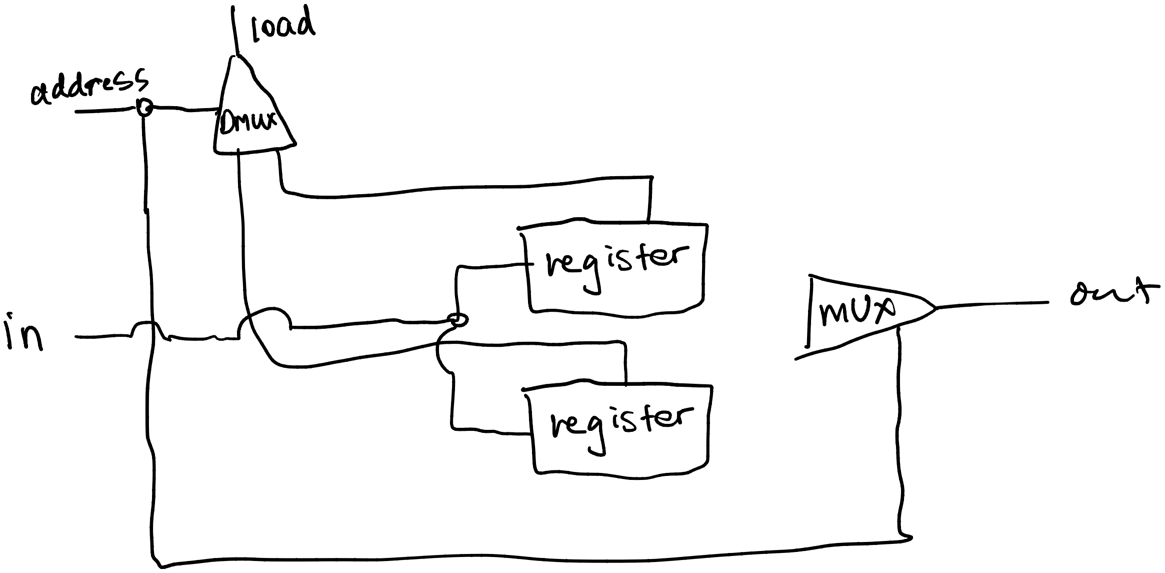


Answer	can't be used as memory

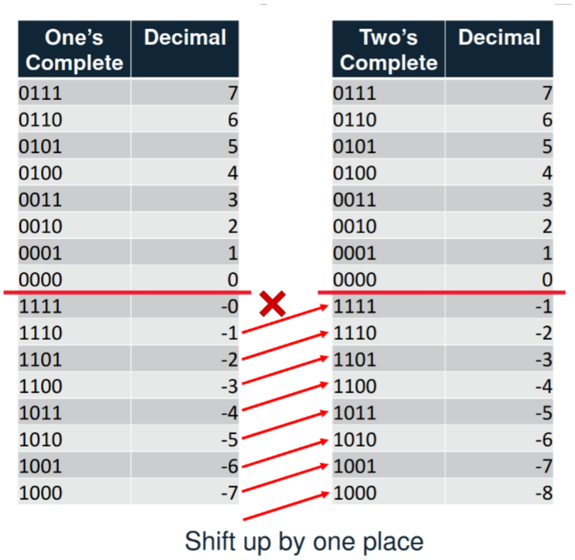
6. Can you draw the implementation of 1-bit register?



7. Can you draw the implementation of a RAM with 2 registers?



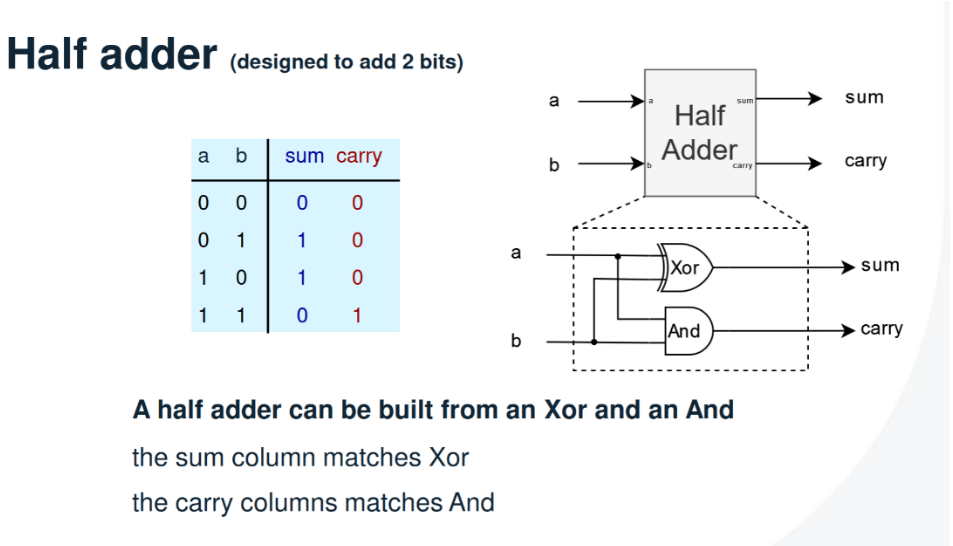
Important Slides for reference:



One's Complete	Decimal
0111	7
0110	6
0101	5
0100	4
0011	3
0010	2
0001	1
0000	0
1111	-0
1110	-1
1101	-2
1100	-3
1011	-4
1010	-5
1001	-6
1000	-7

Two's Complete	Decimal
0111	7
0110	6
0101	5
0100	4
0011	3
0010	2
0001	1
0000	0
1111	-1
1110	-2
1101	-3
1100	-4
1011	-5
1010	-6
1001	-7
1000	-8

Shift up by one place



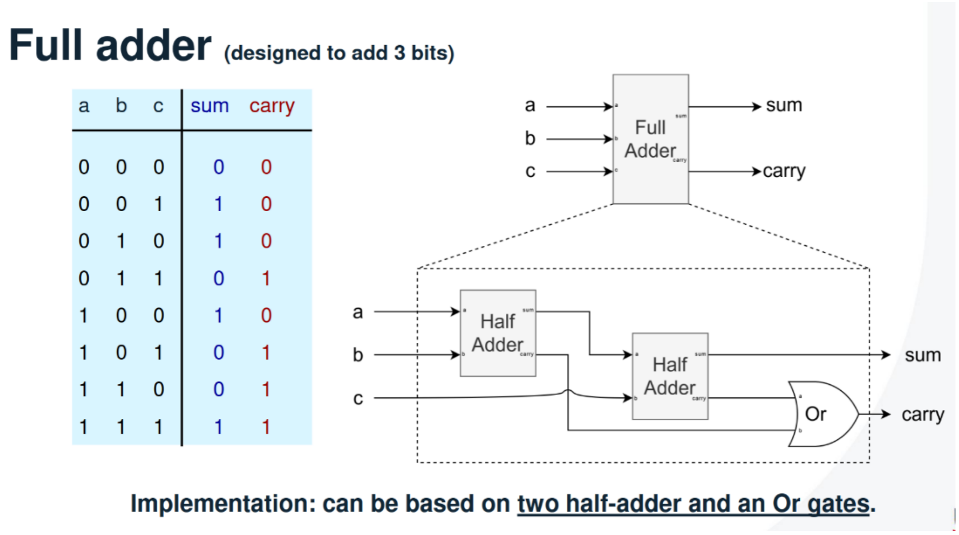
Half adder (designed to add 2 bits)

a	b	sum	carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

A half adder can be built from an Xor and an And

the sum column matches Xor

the carry columns matches And



Full adder (designed to add 3 bits)

a	b	c	sum	carry
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

Implementation: can be based on two half-adder and an Or gates.