

EGC442

Class Notes

1/27/2023



Baback Izadi

Division of Engineering Programs

bai@engr.newpaltz.edu

1) Convert $(32.2)_4$ to base 8.

Handwritten conversion of $(32.2)_4$ to base 8:

$$\begin{array}{ccccccc} & 4^2 & & & & & \\ \text{00} & 11 & 10 & . & 100 & & \\ \hline (1 & 6 & . & 4) & & & \end{array}$$

Red annotations: A red line connects the '4' in the base to the '4' in the exponent. Red arrows point from the circled '4' in the base to the circled '2' in the exponent, and from the circled '8' in the base to the '2' and '3' in the exponent.

2) Convert $(1E2.C1)_{16}$ directly base 2.

8421

Handwritten binary representation of $(1E2.C1)_{16}$ using the 8421 method:

$$0001\ 1110\ 0100\ .\ 1101\ 0101$$

A red arrow points from the '1' in the base to the first '0' in the binary string.

3/

C.

$$\begin{array}{r} (35)_7 \\ \times (5.2)_7 \\ \hline 103 \\ + 244 \\ \hline (254.3)_7 \end{array}$$

$$10 \div 7 = 1$$

$$R = 3$$

$$7 \div 7 = 1$$

$$R = 0$$

$$25 \div 7 = 3$$

$$R = 4$$

$$18 \div 7 = 2$$

$$R = 4$$

5) Using a total of 8 bits, represent -123 in signed 2's complement format.

+ 123
- 123

S	64	32	16	8	4	2	1
X	X	X	X	X	X	X	X
0	1	1	1	1	0	1	1
1	0	0	0	0	1	0	1

6) The given numbers are represented in signed 2's complement. Carry out the indicated operation and specify the indicated flags.

Handwritten solution for the addition of two signed 2's complement numbers:

01101011
 - 11000010

64 32 16 8 4 2 1
 64 32 16 8 4 2 1
 64 32 16 8 4 2 1
 64 32 16 8 4 2 1

+ 01101011
 + 00111001

107
 62

 69

0011101

 0011110

CY = 0
 S = 1
 overflow = 1

64 32 16 8 4 2 1
 - 0101011
 (-87)

Detailed description: The image shows a handwritten solution for adding two 8-bit signed 2's complement numbers. The first number is 01101011 (decimal 11) and the second is -11000010 (decimal -10). The addition is performed bit-by-bit from right to left, with carry propagation. The result is 10762, which is written as 69. The carry flag (CY) is 0, the sign flag (S) is 1, and the overflow flag is 1. A separate calculation shows the 2's complement of the result, -0101011, which is -87. Bit weights (64, 32, 16, 8, 4, 2, 1) are indicated for each bit position.

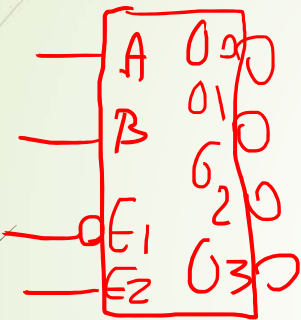
$2 \times 3 = 6 \rightarrow 4^2 0$

10) Show the **truth table only** of a combinational circuit that multiplies two numbers A (A_1A_0) and B (B_1B_0). The range of A is 0 to 2 and the range of B is 1 to 3.

A_1	A_0	B_1	B_0	M_2	M_1	M_0
0	0	0	0	X	X	X
0	0	0	1	0	0	X
0	0	1	0	0	0	0
0	0	1	1	0	0	0
0	1	0	0	0	0	0
0	1	0	1	0	0	0
0	1	1	0	0	0	0
0	1	1	1	0	0	0
1	0	0	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	0	0	0
1	0	1	1	0	0	0
1	1	0	0	0	0	0
1	1	0	1	0	0	0
1	1	1	0	0	0	0
1	1	1	1	0	0	0

$\downarrow \downarrow$
 4
 6

11) Design a 1-out-of-4 decoder with low active outputs and two enable lines, one active high and the other active low. Show the block diagram, the truth table and the internal circuitry.



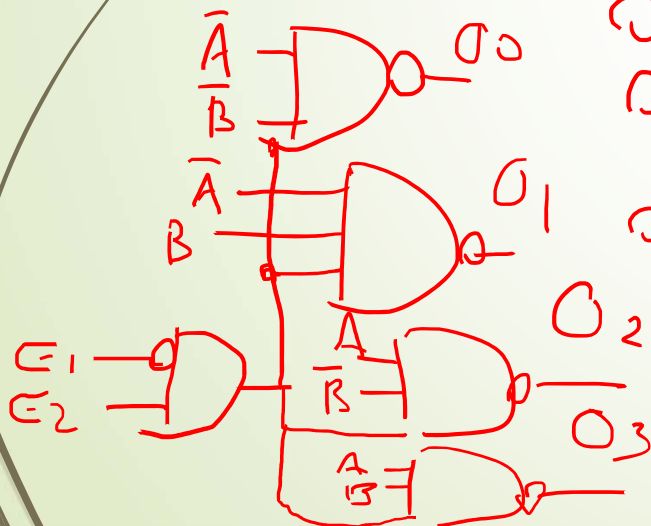
E_1	E_2	A	B	O_3	O_2	O_1	O_0
1	X	X	X	1	1	1	1
X	0	X	X	1	1	1	1
0	1	0	0	1	1	1	0
0	1	0	1	1	1	0	1
0	1	1	1	0	1	1	1

$$O_0 = \bar{E}_1 \bar{E}_2 \bar{A} \bar{B}$$

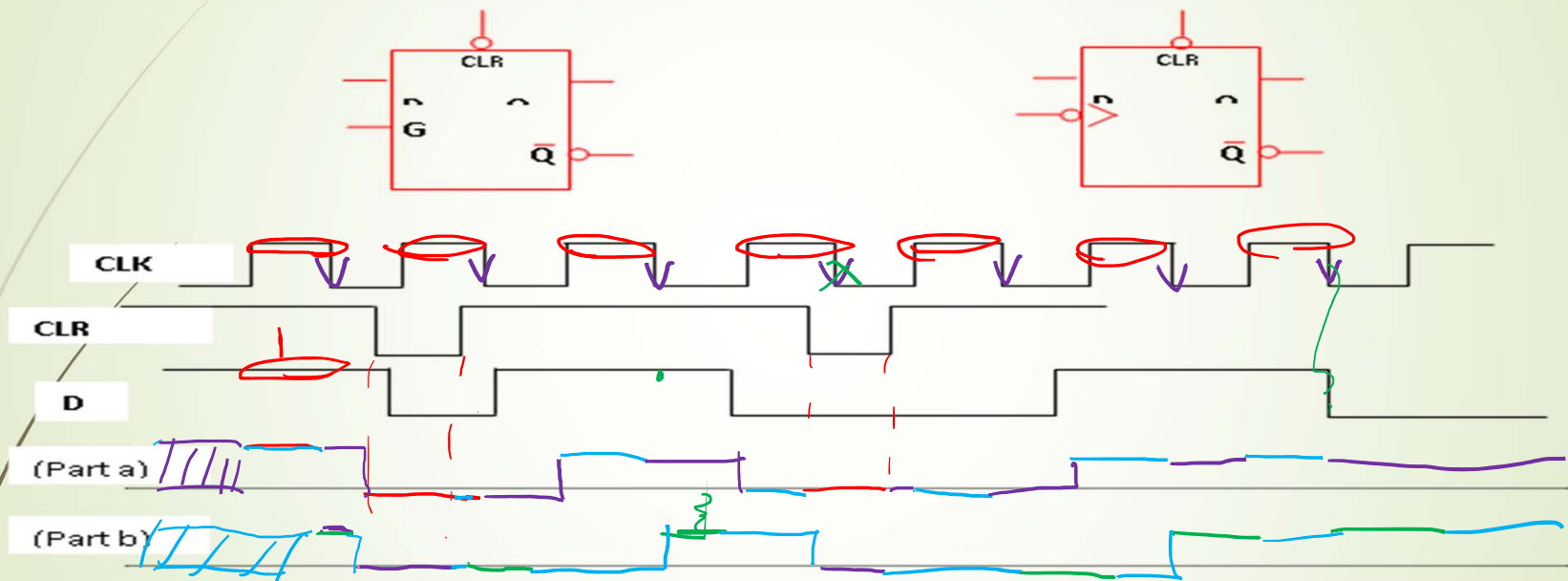
$$O_1 = \bar{E}_1 \bar{E}_2 \bar{A} B$$

$$O_2 = \bar{E}_1 \bar{E}_2 A \bar{B}$$

$$O_3 = \bar{E}_1 \bar{E}_2 A B$$

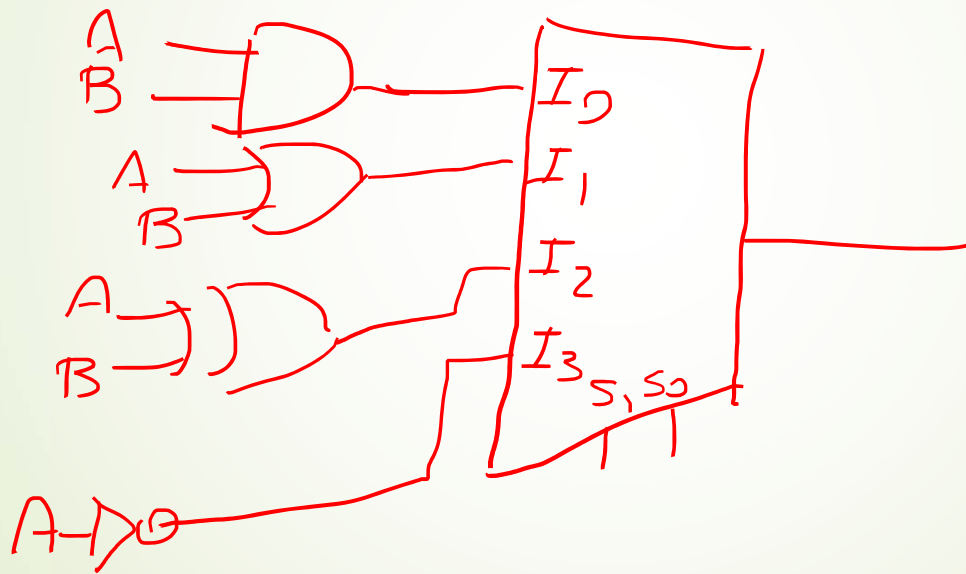


16) Complete the following timing diagram for the given device. Assume Q to be initially 0.



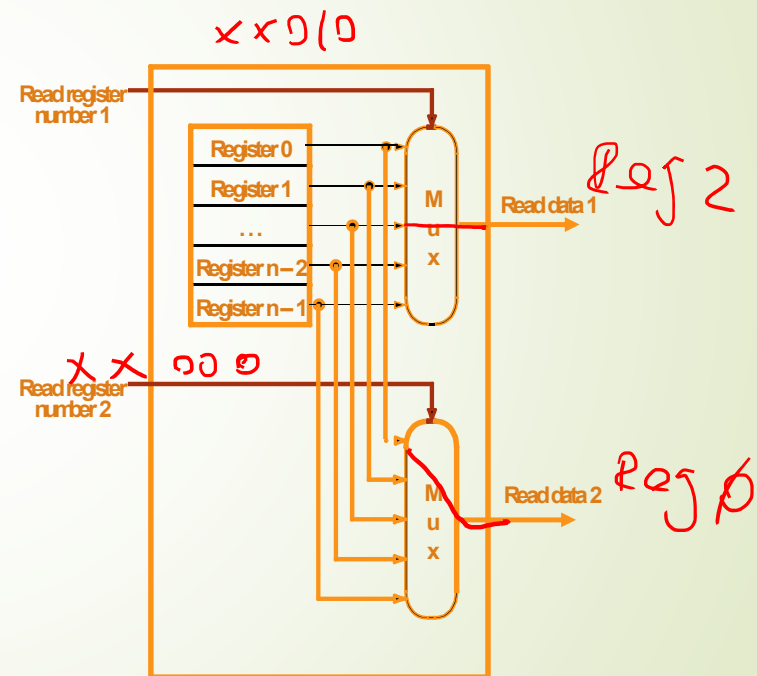
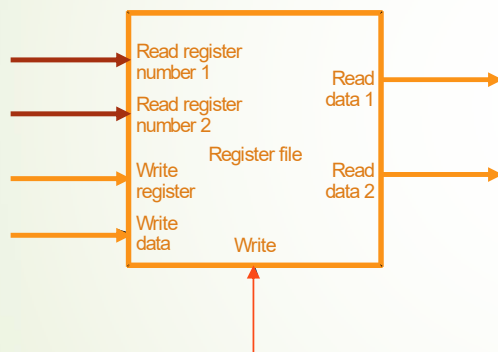
15) Using gates and multiplexer(s), design a one-bit ALU that performs the following logical op

S ₁	S ₀	Function
0	0	AND
0	1	OR
1	0	XOR
1	1	NOT



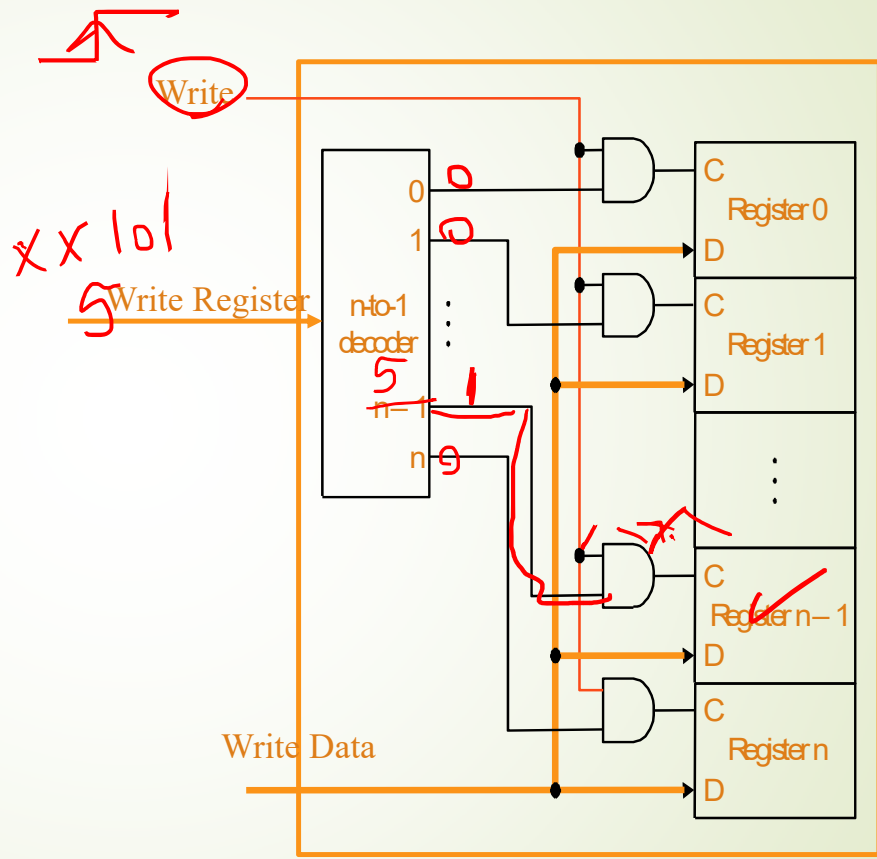
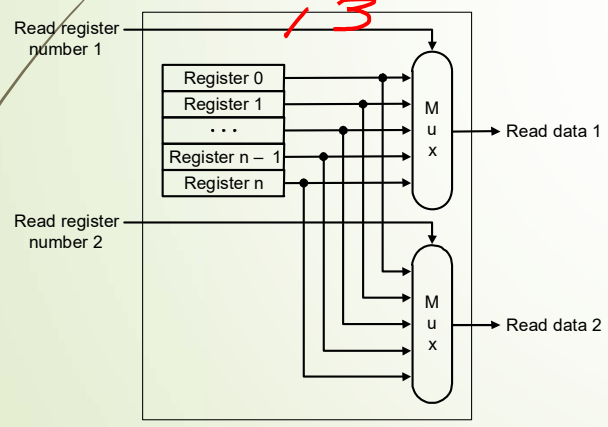
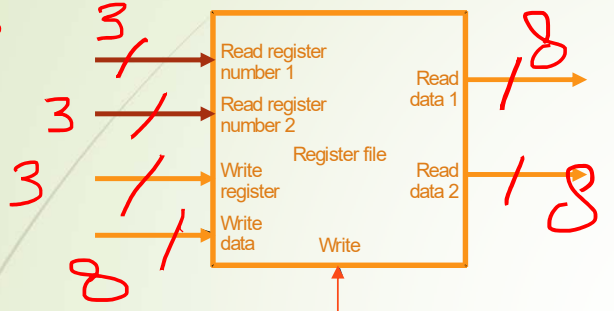
Design of Processor's Register File

- Ability to read from two registers and write to one register
- Read operation using D flip-flops and MUX's



8 reg. each reg 8 bit Register File – Read and Write

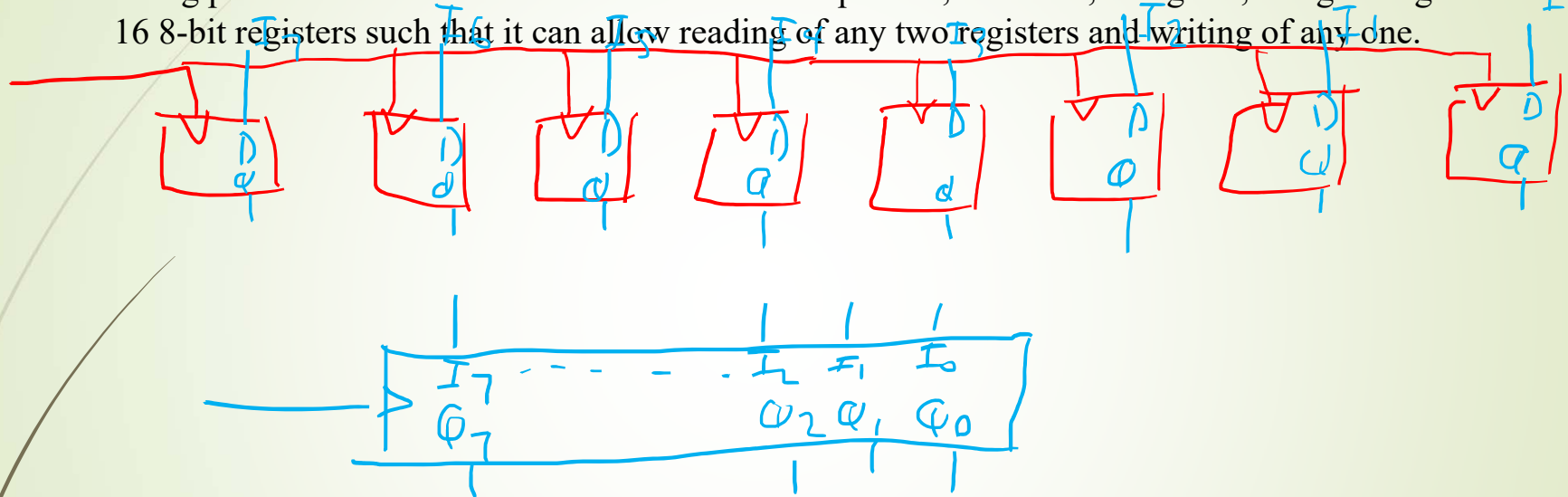
$2^3 = 8$

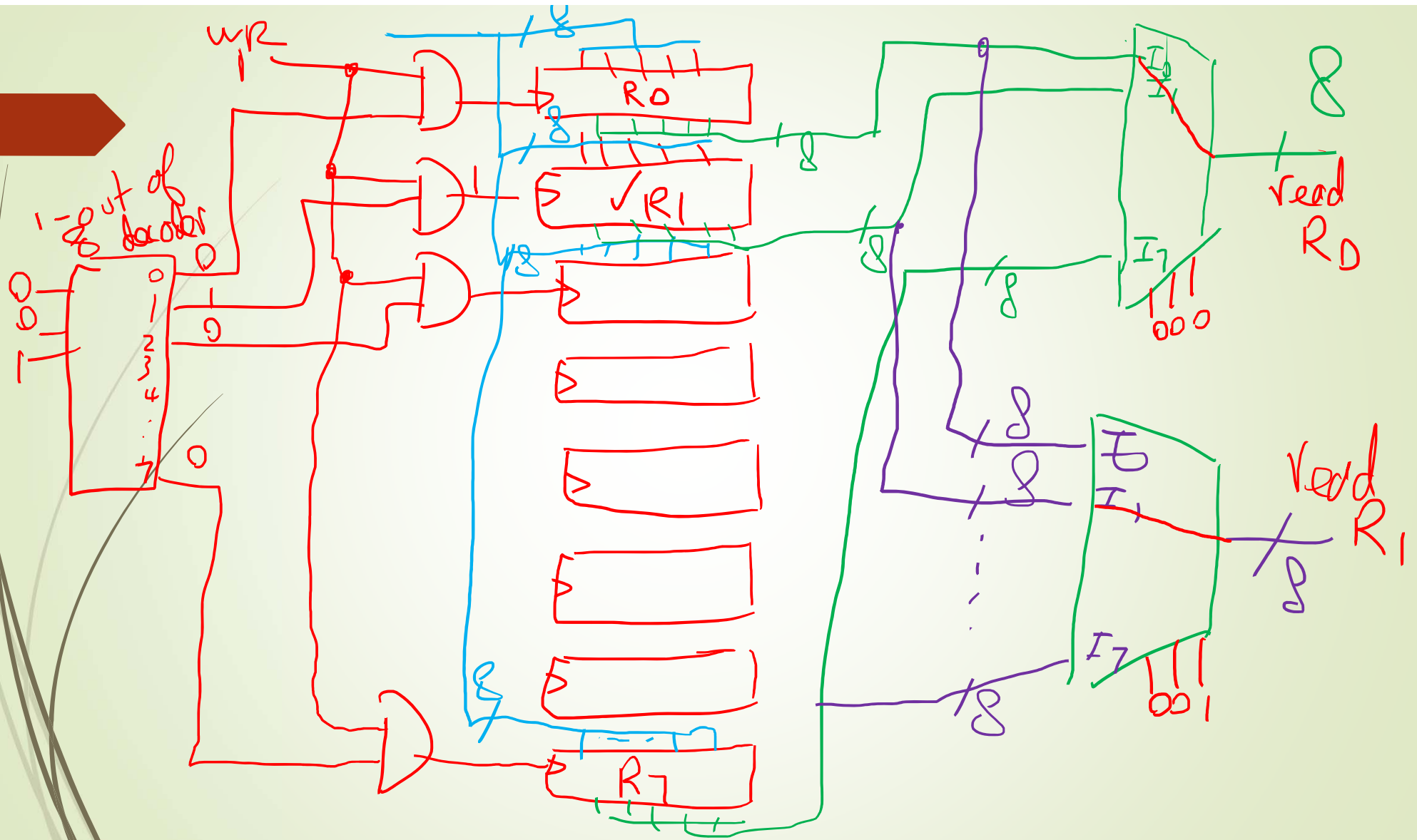


17.

a. Using D flip-flop, design an 8-bit register.

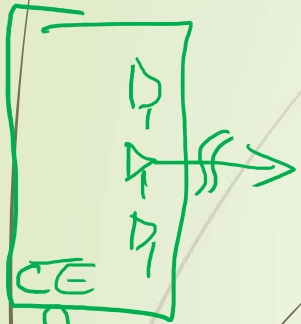
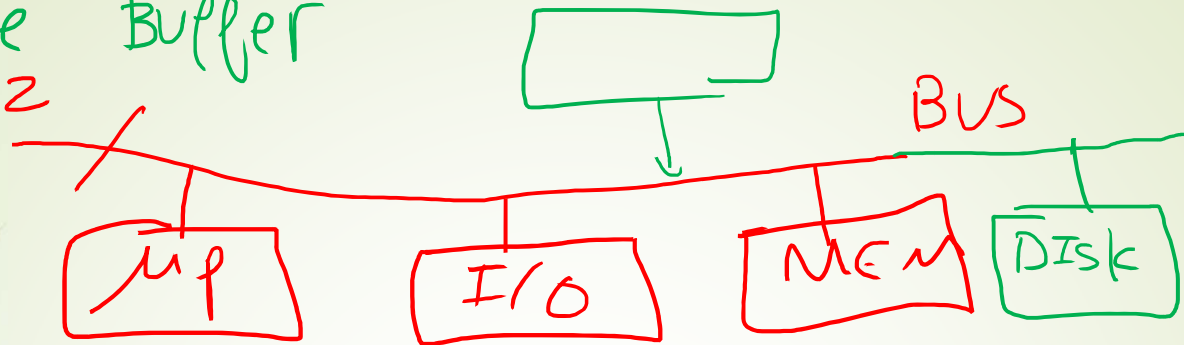
b. Using part a as well as other devices such as multiplexers, decoders, and gates, design a register file with 16 8-bit registers such that it can allow reading of any two registers and writing of any one.





TRI-state Buffer

32



0 → D_{out}

1 → HI-Z

Decode

