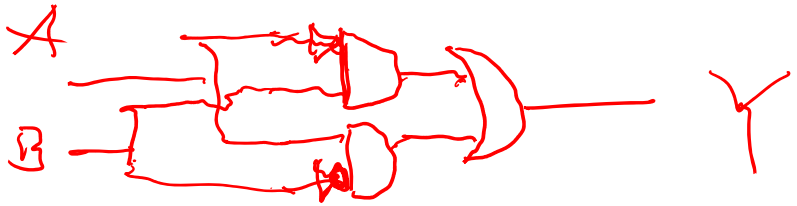
Build A Simple Computer

1. Any function defined on binary input and output variables can be implemented as Boolean expression. True or False?

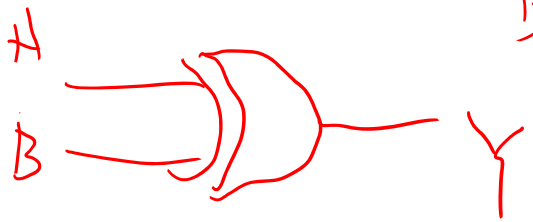
A. True
B. False

2. Which of the following is the canonical expression for XOR?

A. Y = ~ (A | B)
B. Y = ~A | ~B
C. Y= ~A & B
D. Y = (~A & B) | (A & ~B)



A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0



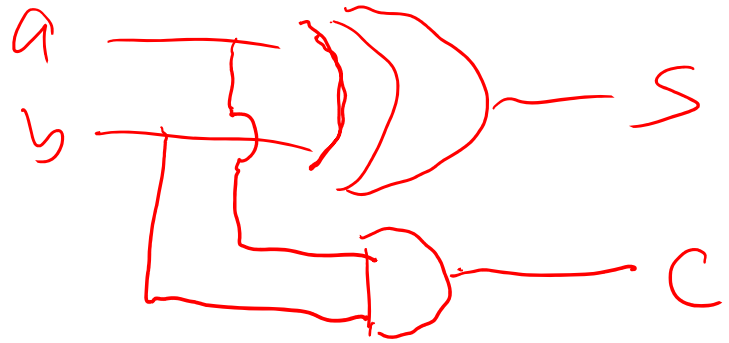
1 Half Adder

C = Carry

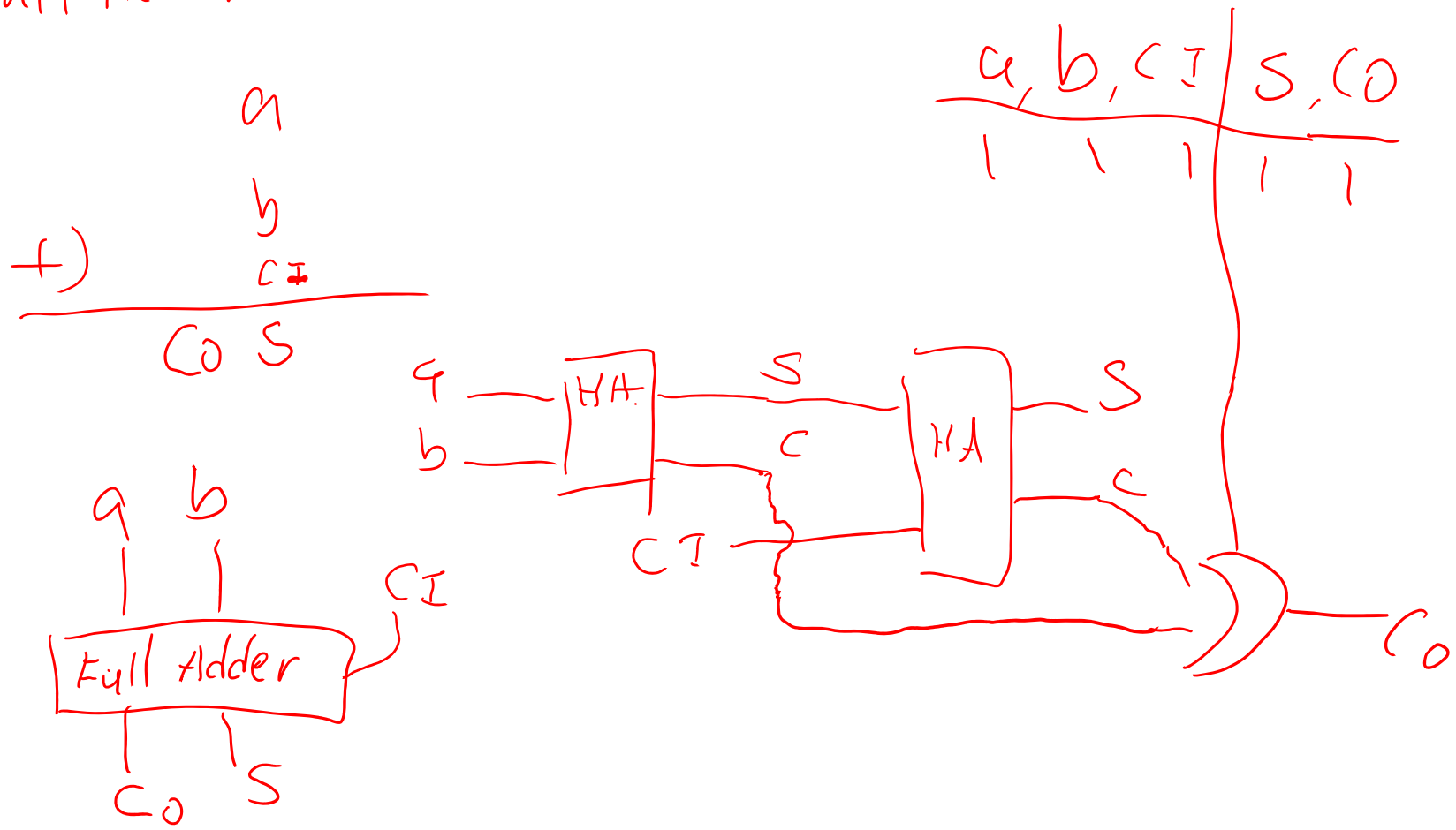
S = sum

a	b	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$S = f(a, b) = XOR(a, b)$



2. Full Adder

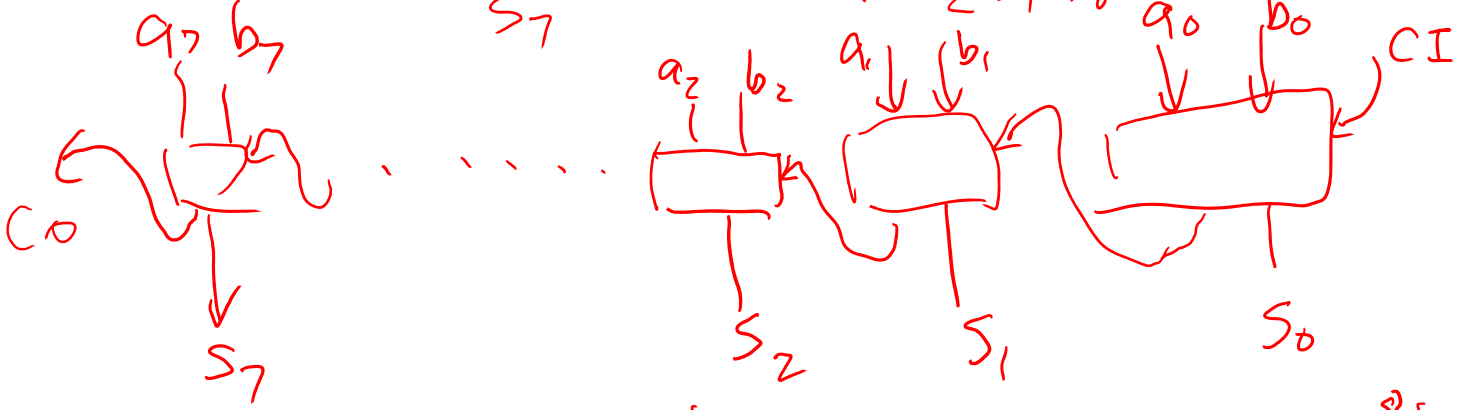


3. Adder

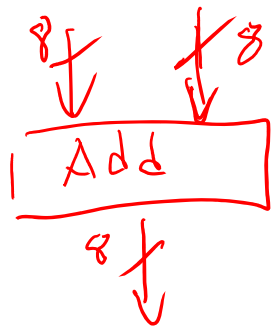
$a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0$

+) $b_7 b_6 \cdots \cdots b_0$

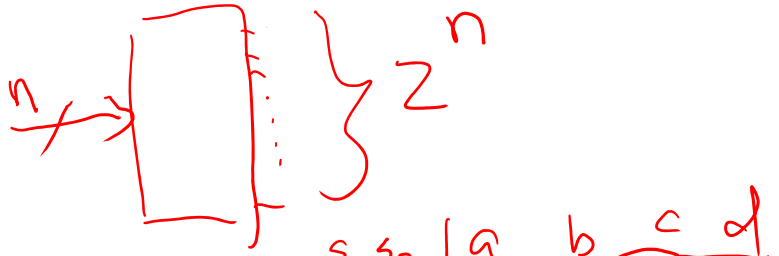
$S_7 \; - \; - \; \cdots \; S_2 \; S_1 \; S_0$



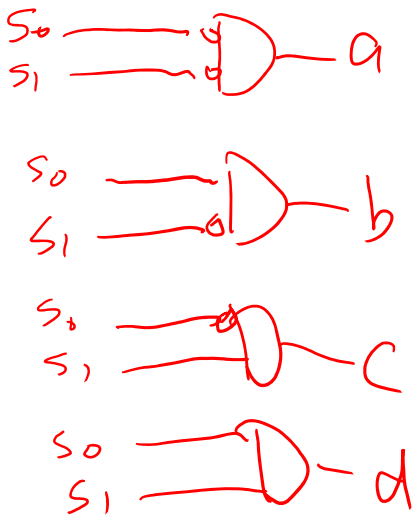
Ripple Adder



4. Decoder



S_1 S_0	a	b	c	d
0 0	1	0	0	0
0 1	0	1	0	0
1 0	0	0	1	0
1 1	0	0	0	1



- ADD
- SUB
- AND
- NOT
- OR
- SLL
- SRL
- ⋮

Output is "one-hot" decoding of the Input

3. Which of the following is the canonical expression for the function defined by the following truth table?

multiplexer

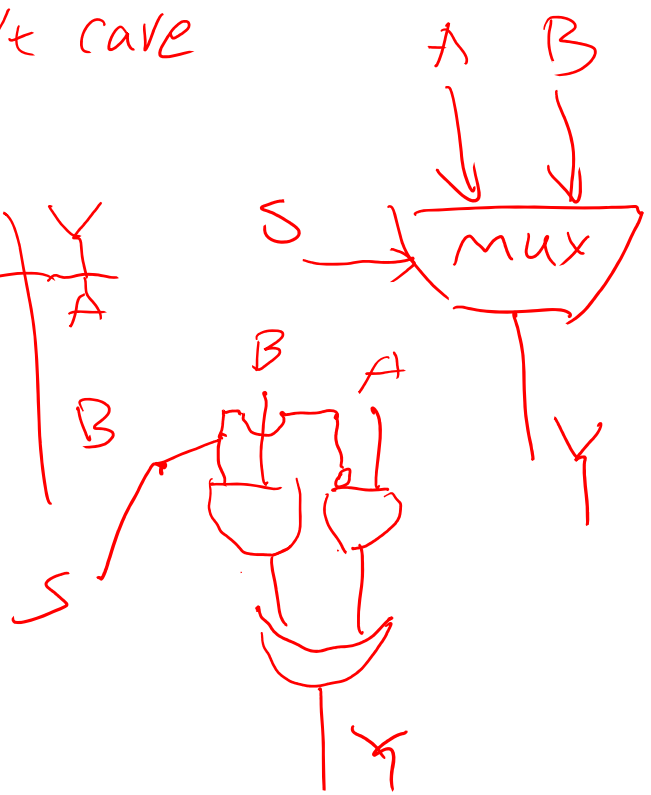
A. Y = A.~B.~S
B. Y = A.B.~S
C. Y= ~A.B.S
D. Y = A.B.S
E. Y = A.~B.~S + A.B.~S + ~A.B.S + A.B.S

X: don't care

A	B	S	Y
0	0	0	0
0	1	0	0
1	0	0	1
1	1	0	1
0	0	1	0
0	1	1	1
1	0	1	0
1	1	1	1

⇒

A	B	S	Y
A	X	0	A
X	B	1	B



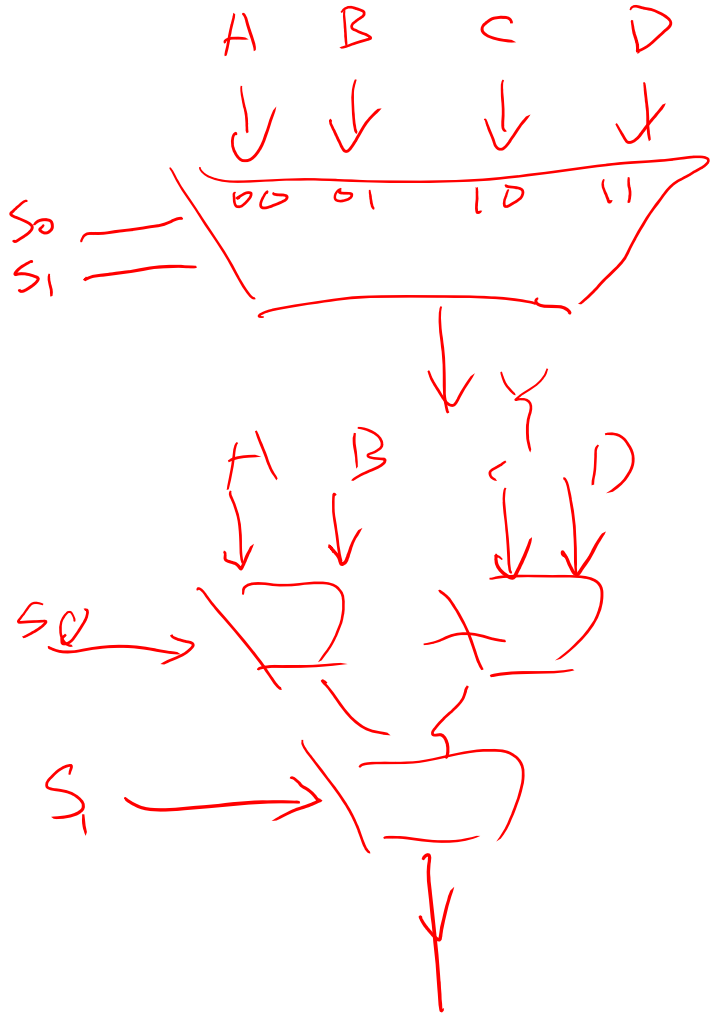
4. Are the following two Boolean expressions equivalent to each other for the function Y = f(A,B,S) defined by the following truth table?

Y = A & ~S | B & S
Y = A &~B &~S | A & B & ~S | ~A & B & S | A & B & S

A	B	S	Y
0	0	0	0
0	1	0	0
1	0	0	1
1	1	0	1
0	0	1	0
0	1	1	1
1	0	1	0
1	1	1	1

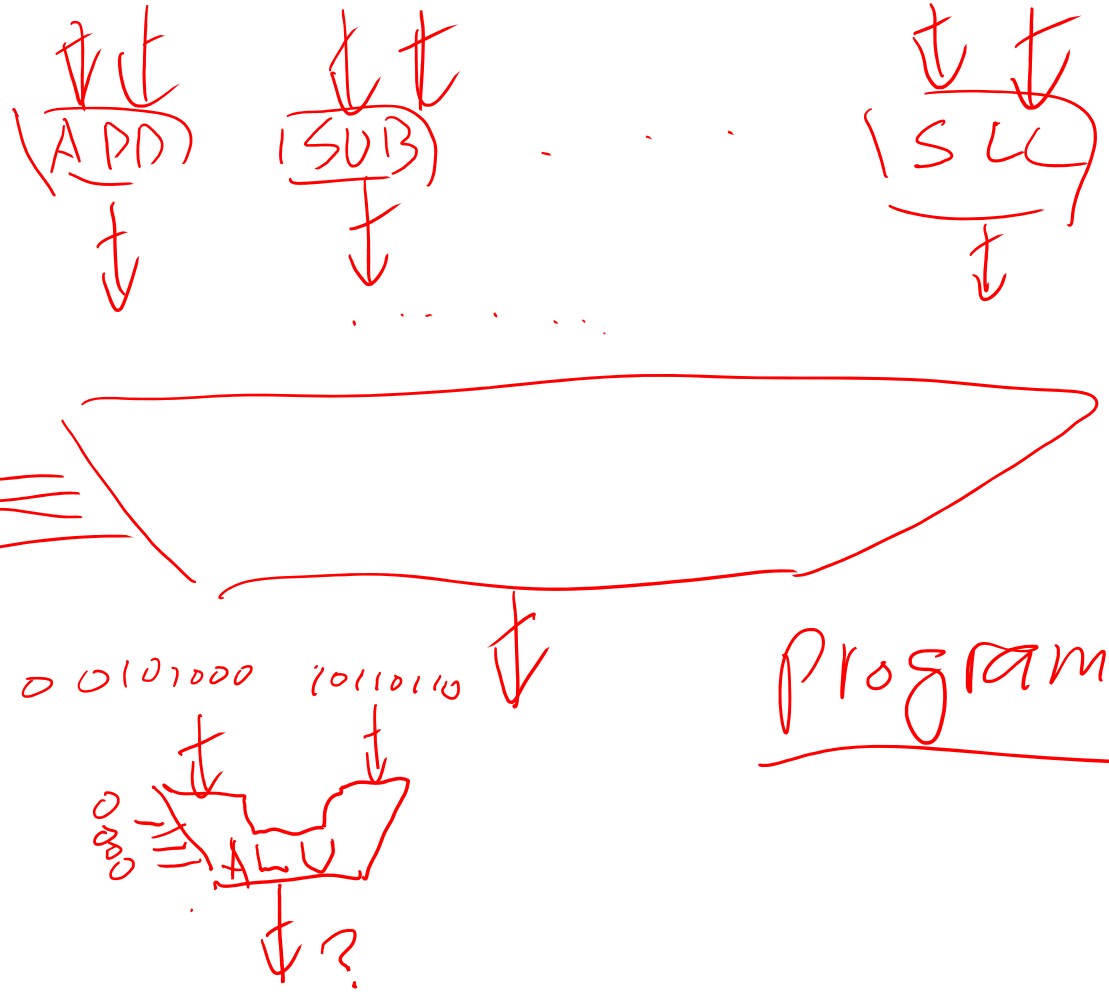
A. Yes
B. No

5 MuX



A	B	C	D	S_0	S_1	Y
A	x	x	x	0	0	A
x	B	x	x	0	1	B
x	x	C	x	1	0	C
x	x	x	D	1	1	D

6. ALU



Programmable!!

7. Storage

- read
- write
- Addressable

Sequential logic
V.S
combinational logic

⇒

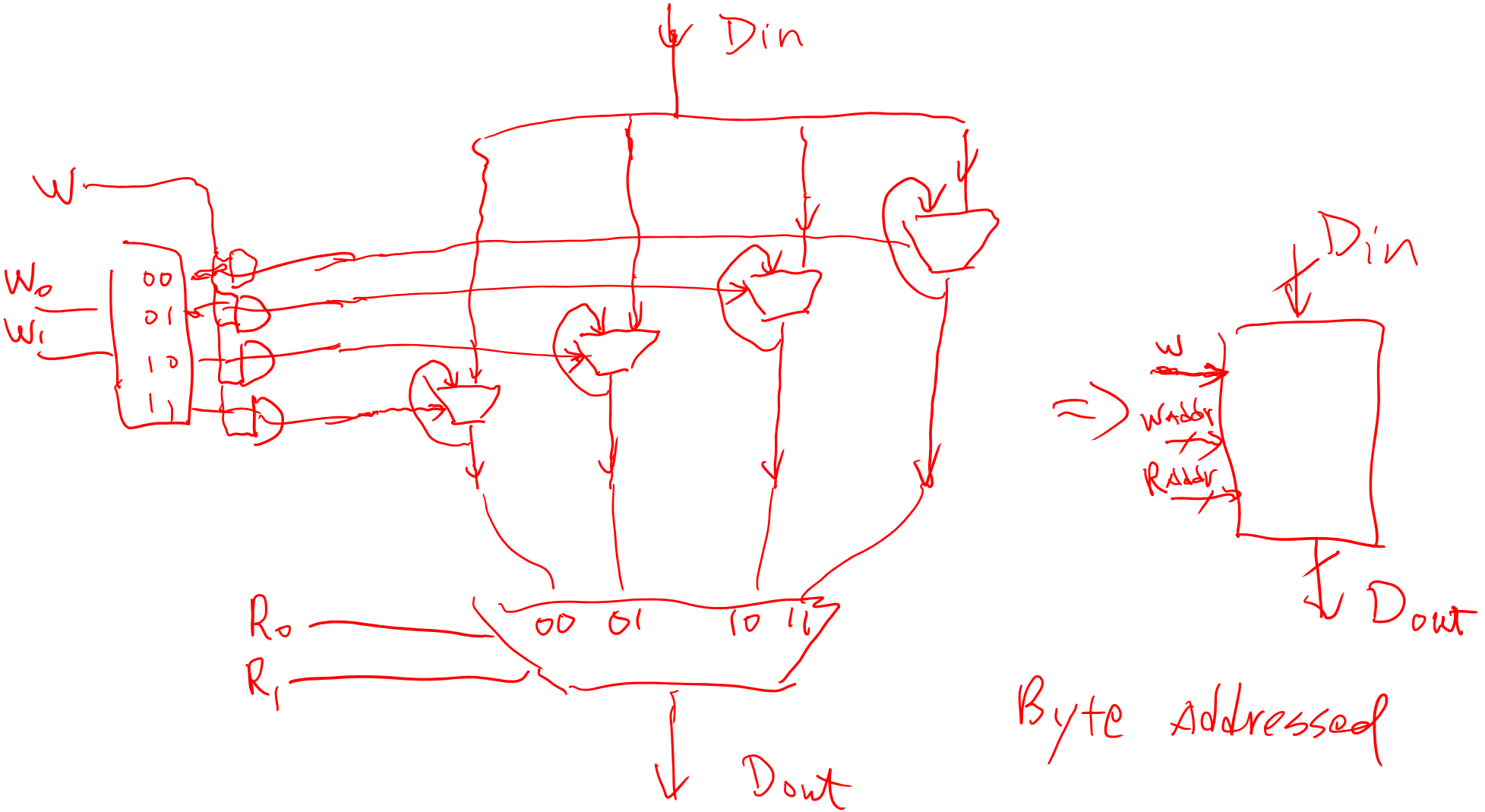
B	S_0	Y
0	0	?
1	0	?
1	1	1
1	0	1
0	0	1
0	1	0
0	0	0

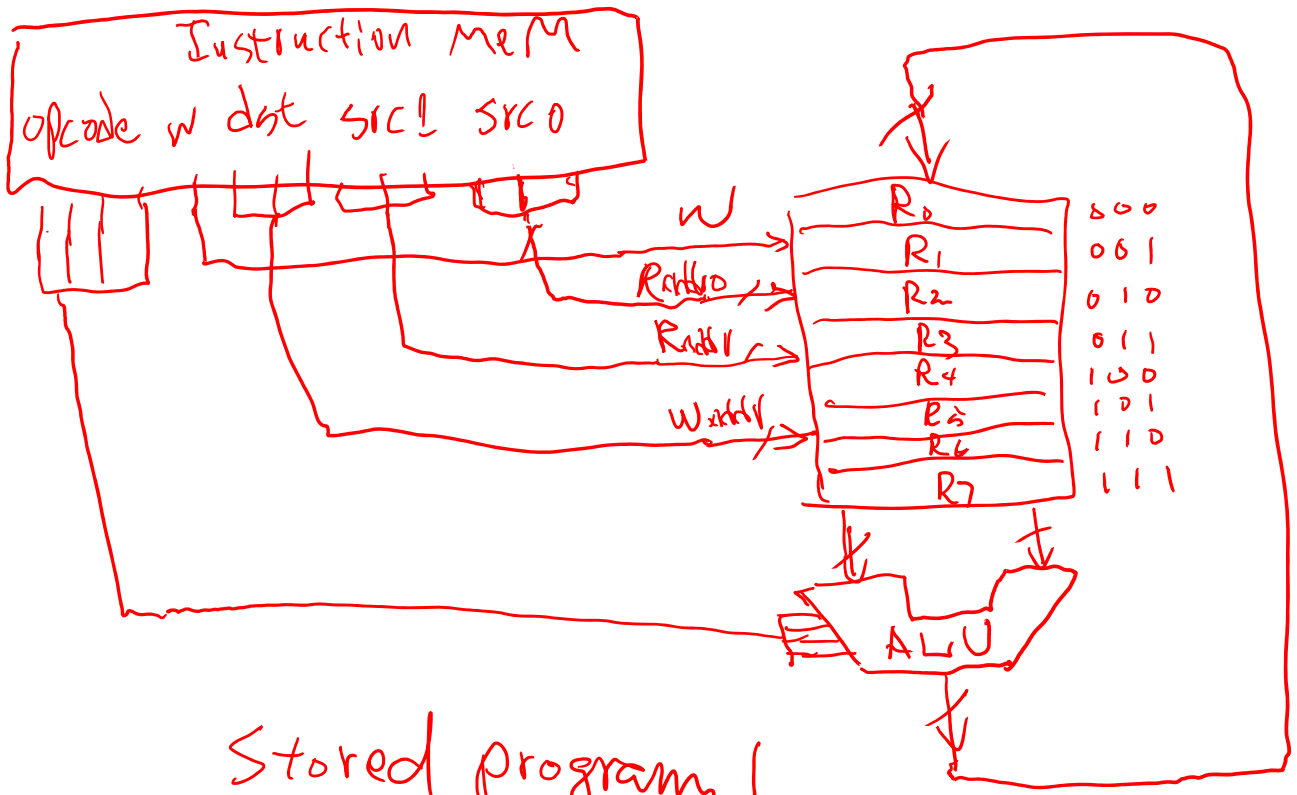
A	B	S_0	Y
A	X	0	A
X	B	1	B

Insanity: doing the same thing over and over again and expecting different results? -Albert Einstein

Do you agree or not?

A. Yes
B. No


Din
W
W0
W1
00
01
10
11
R0
R1
00 01 10 11
Dout
Din
W
Waddr
Raddr
Dout
Byte Addressed



Register file

operands

Assembly code

Stored program!

Von Neumann

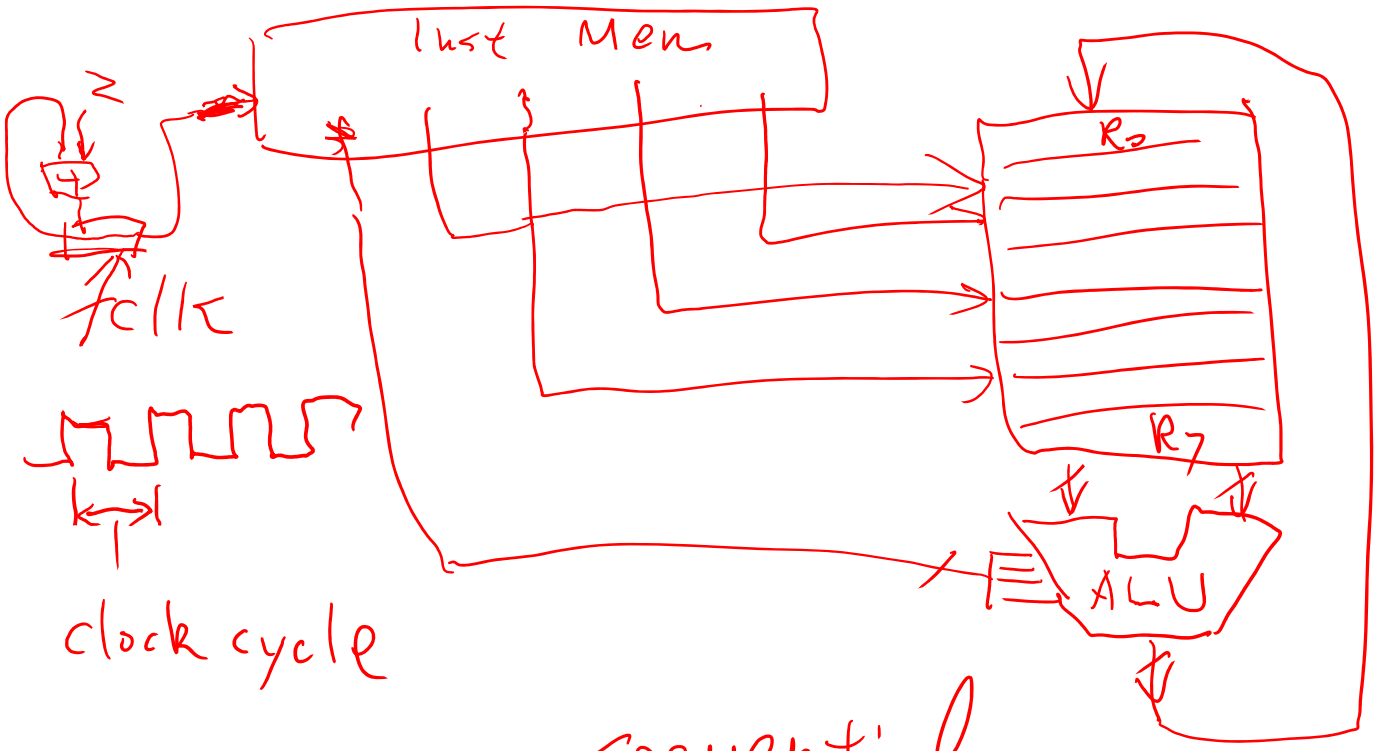
ADD R_7, R_0, R_1

$R_7 \leftarrow R_0 + R_1$

0000 | 111 001 000

opcode

Machine Instruction



sequential order execution of instructions