CSSE2010/CSSE7201
Lecture 4

Combinational Logic

→ bm
→ gates
→ ADD
→

School of Information Technology and Electrical Engineering
The University of Queensland

Today…

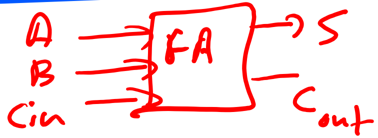
- Admin
- ✓ Binary Subtractors
- More Combinational Logic Circuits
  - ✓ Multiplexers
  - ✓ Decoders
  - Timing diagram representations

Admin

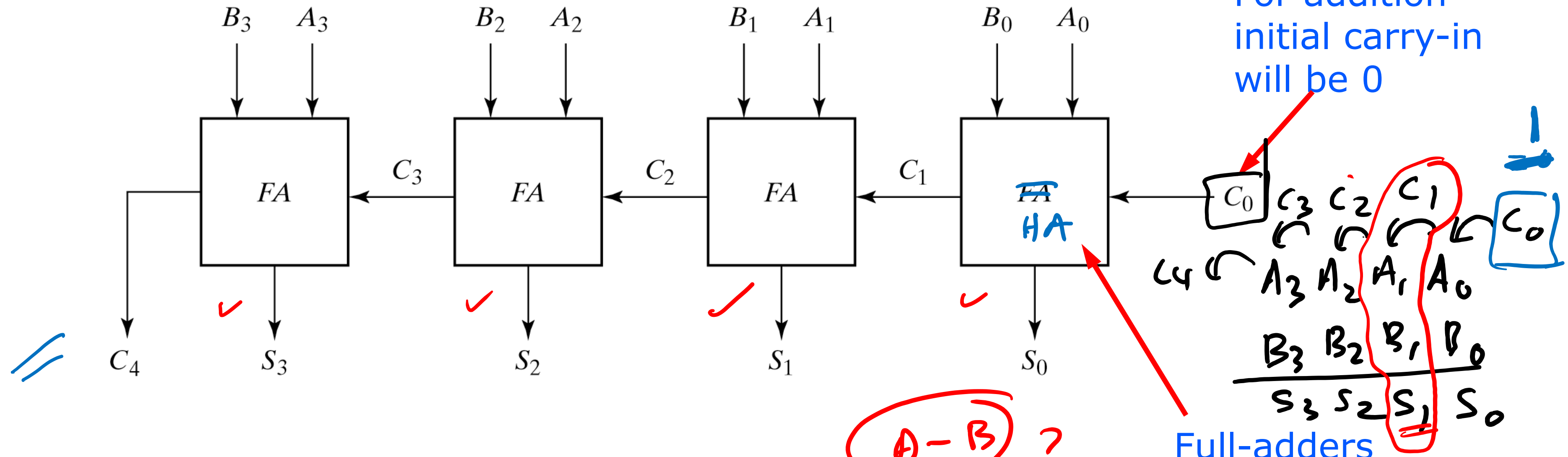
- Quiz 1 due on Friday 4pm – no extensions for quizzes
- Lab 4 preparation task – next week (week 3)
  - Will be Available on Blackboard: Learning Resources
  - Draw some circuit schematic diagrams
  - Bring to Lab 4 (Mon-Tue next week)
  - IN students – use Logic ICs on breadboards or Logisim
  - EX students – use Logisim software to simulate
- Supporting material for labs are now on Blackboard

Recall – Binary Adder

- Can cascade full adders to make binary adder
  - Example: for 4 bits…



For addition initial carry-in will be 0



Full-adders

A − B ?

- This is a **ripple-carry adder** ✓

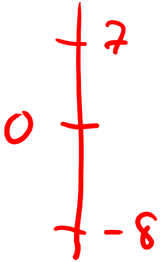
[Figure from Mano, "Digital Design", 3rd edition]

Binary subtraction

- A-B - usually implemented as A+(-B) 2's comp of B
  - A and B are multi-bit quantities
  - "+" in this case means addition (not OR)
  - -B means negative B – the two's complement of B
- Two's complement of B can be calculated by flipping bits and adding 1

~~3+3~~

4 bits



5-3 = 2

conventional sub

```
  0101
  0011
  ----
  0010
```

=

5-3 = 2   2's comp

```
        1  1
        0101
2's comp
of 3 →  1101   +
        ----
     1  0010
```

5-3

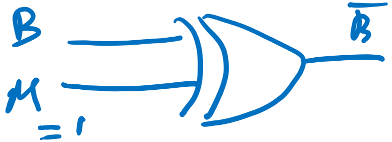
5+(-3)

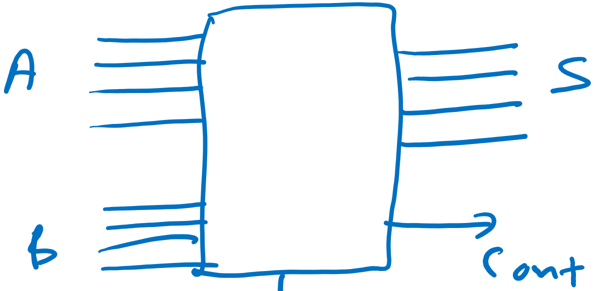
```
  0011
  1100
     1
  ----
  1101
```

Binary subtraction (cont.)

- How we can use a gate to flip a bit – but only sometimes? i.e.
  - Z=not(B) when M is 1
    Z=B when M is 0

$A+(-B)$?





XOR

B → x, M → y

x	y	XOR
0	0	0
0	1	1 ✓
1	0	1
1	1	0

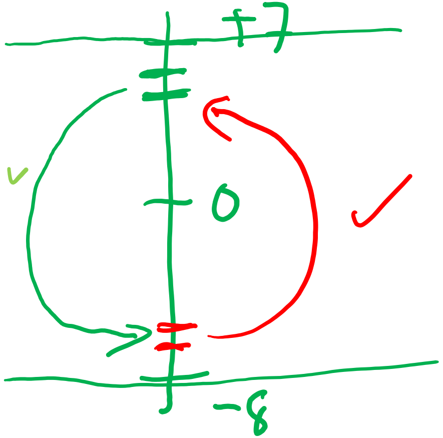
$$\boxed{\begin{aligned} x \oplus 0 &= x \\ x \oplus 1 &= \bar{x} \end{aligned}}$$

✓	$A+B$	$M=0$	ADD
✓	$A-B$	$M=1$	SUB

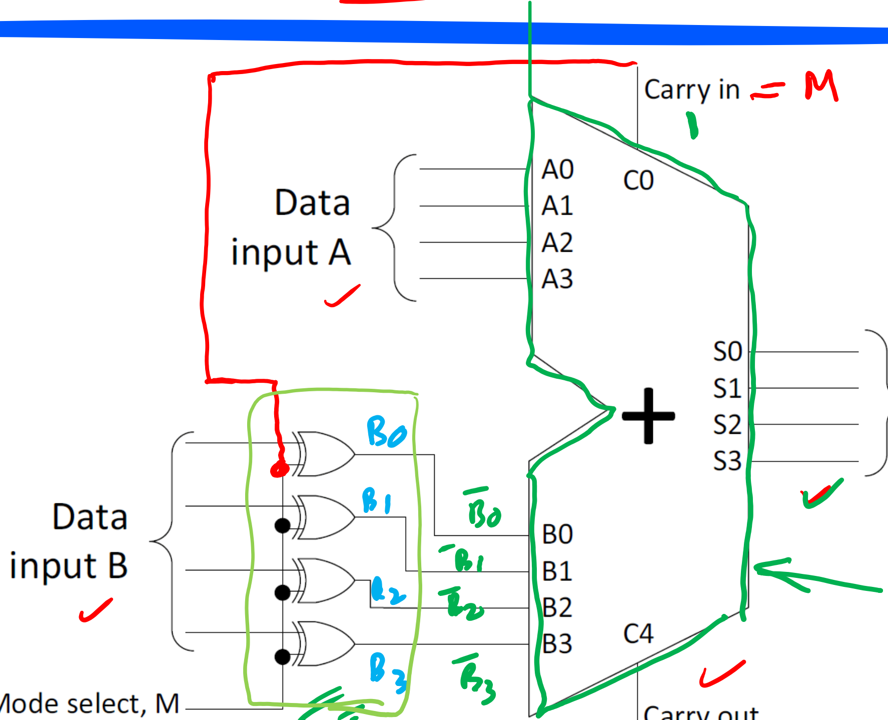
$$\begin{array}{r} \bar{B}_3\ \bar{B}_2\ \bar{B}_1\ \bar{B}_0 \\ +\ \ 0\ \ \ 0\ \ \ 0\ \ \ 1 \\ \hline \end{array}$$

Carry in to the LSB

Adder-subtractor



$C_{in} \oplus C_{out}$



Mode select, M

M=0 means add $A + B$

M=1 means subtract $A - B = A + (-B)$

- What should carry-in be?

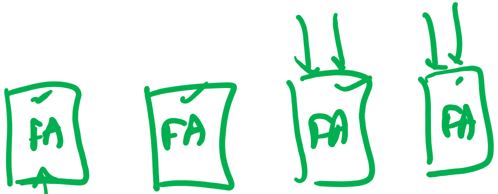
* $M = 0$

$A + B$

* $M = 1$

$A + (-B)$

2's comp of B



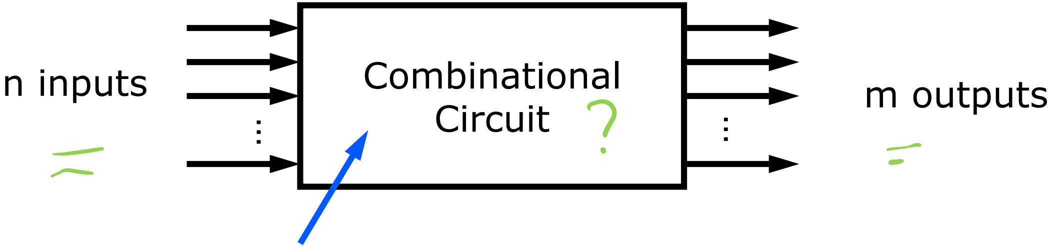
$S = A \oplus B \oplus C_{in}$

$C_{out} = AB + C_{in}(A \oplus B)$

Combinational Circuits

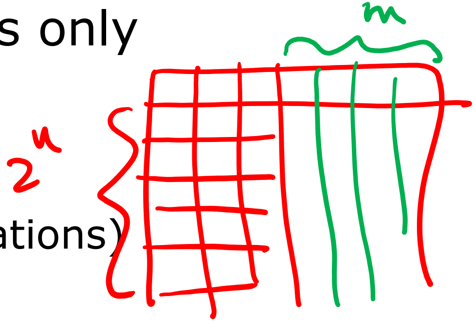
- Generally,

→ Combinational
→ Sequential



Combination of Logic gates

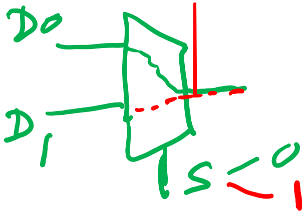
- Each output can be expressed as function of n input variables
- Output depends on current inputs only
- Can write truth table also:
  - n input columns
  - m output columns
  - 2^n rows (i.e. possible input combinations)

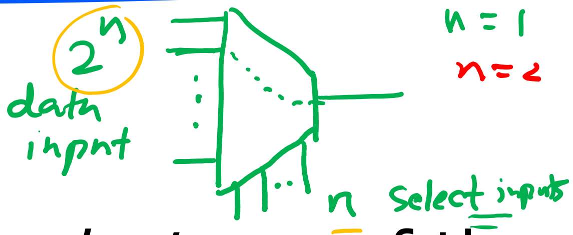


✓ADD
✓ADD/SUB
✓MUX
DeMux
✓Decoder
Encoder

Multiplexer (or Mux)

- 2^n data inputs
- 1 output
- n control (or ***select***) inputs – that *select* one of the inputs to be "sent" or "steered" to the output
- Example: 4-to-1 multiplexer





2:1

4:1

8:1

16:1

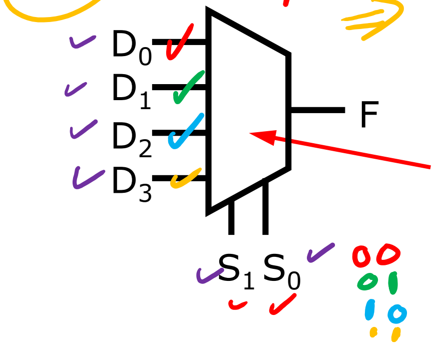
*

4:1

⇒

2^6

64



Logic symbol for multiplexer

*

Function table

S_1	S_0	F
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

0 1

0 1

"Select" inputs

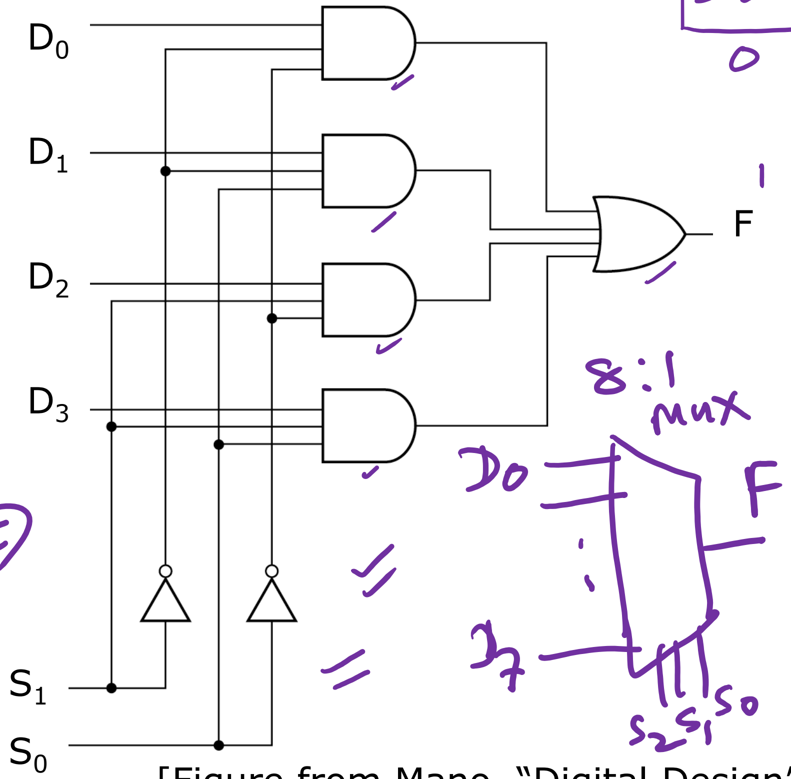
4-to-1 Multiplexer
Logic Circuit Implementation

Function table

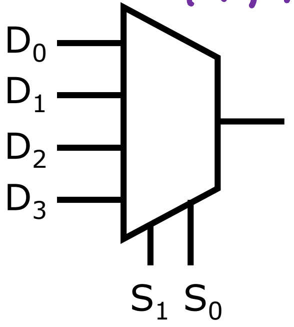
$S_1 S_0$	F
✓ 0 0	D_0 ✓
✓ 0 1	D_1 ✓
✓ 1 0	D_2 ✓
✓ 1 1	D_3 ✓

$$F = \bar{S}_0 \bar{S}_1 D_0 + \bar{S}_1 S_0 D_1 + S_1 \bar{S}_0 D_2 + S_0 S_1 D_3$$

D_0	D_1	D_2	D_3	S_0	S_1	F
0	0	0	0	0	0	0
		⋮				⋮
1	1	1	[illegible]	1	1	0



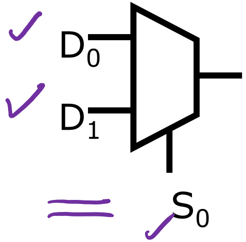
4:1 mux



[Figure from Mano, "Digital Design", 3rd edition]

Clicker Question: 2-to-1 Multiplexer

- Consider a 2-to-1 multiplexer
  - Data Inputs: D_0 and D_1
  - Control Input: S_0
  - Output: F
- What is the truth table that goes with this circuit?



Function Table

S_0	F
0	D_0
1	D_1

1.

S_0	D_0	D_1	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

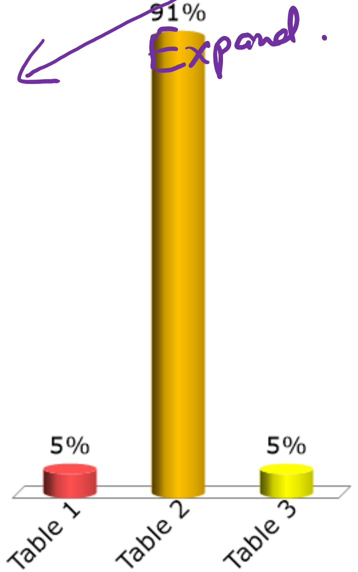
2.

S_0	D_0	D_1	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

3.

S_0	D_0	D_1	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Expand.

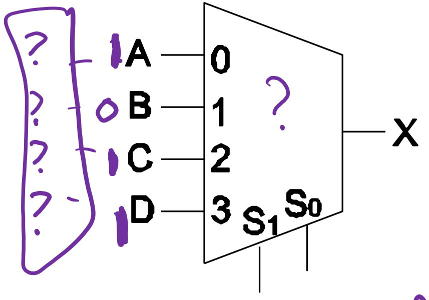


0

Clicker Question (2)

Consider the multiplexer shown. What must the inputs A,B,C,D be so that the multiplexer output is

$$\boxed{X = \overline{S_0} + S_1} \; ?$$



x

✓ 0 0 → A → 1+0=1

✓ 0 1 → B → 0+0=0

✓ 1 0 → C → 1+1=1

✓ 1 1 → D → 0+1=1

10% 1. A=0, B=1, C=0, D=0

8% 2. A=0, B=1, C=1, D=1

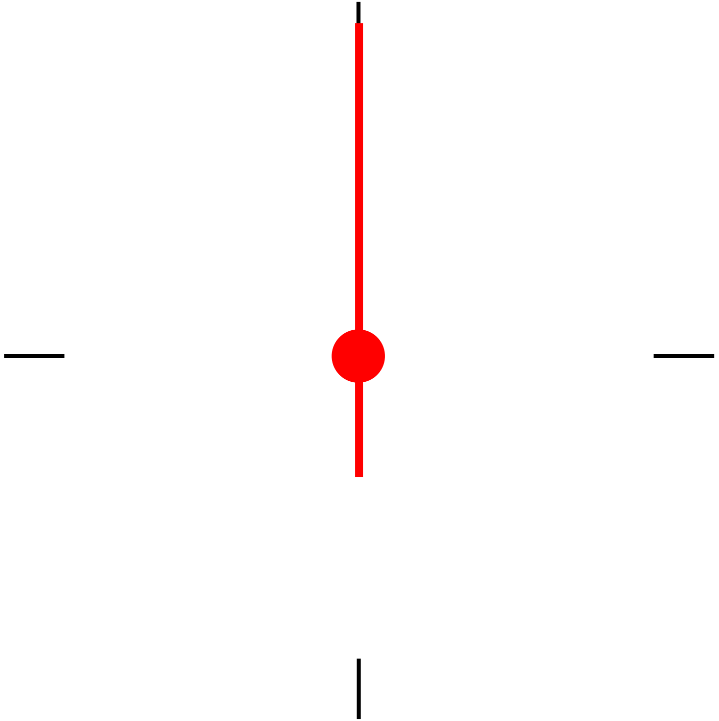
✓ 55% 3. A=1, B=0, C=1, D=1 ✓✓

27% 4. A=1, B=1, C=0, D=1

0

Short Break

- Stand up and stretch



A B

00 → $\bar{A}\bar{B}$

01 → $\bar{A}B$

10 → $A\bar{B}$

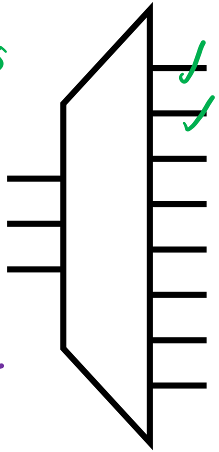
11 → AB

minterms

Decoder

- Converts n-bit input to a logic-1 on exactly one of 2^n outputs
- Example:
  3-to-8 decoder

address



0 0
0 0
1 0

n

2^n

O/P

MSB LSB

$\bar{A}\bar{B}\bar{C}$

$\bar{A}\bar{B}C$

2^n

A	B	C	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0	0	0	**1**	0	0	0	0	0	0	0
0	0	1	0	**1**	0	0	0	0	0	0
0	1	0	0	0	**1**	0	0	0	0	0
0	1	1	0	0	0	**1**	0	0	0	0
…	…	…	…	…	…	…	…	…	…	…

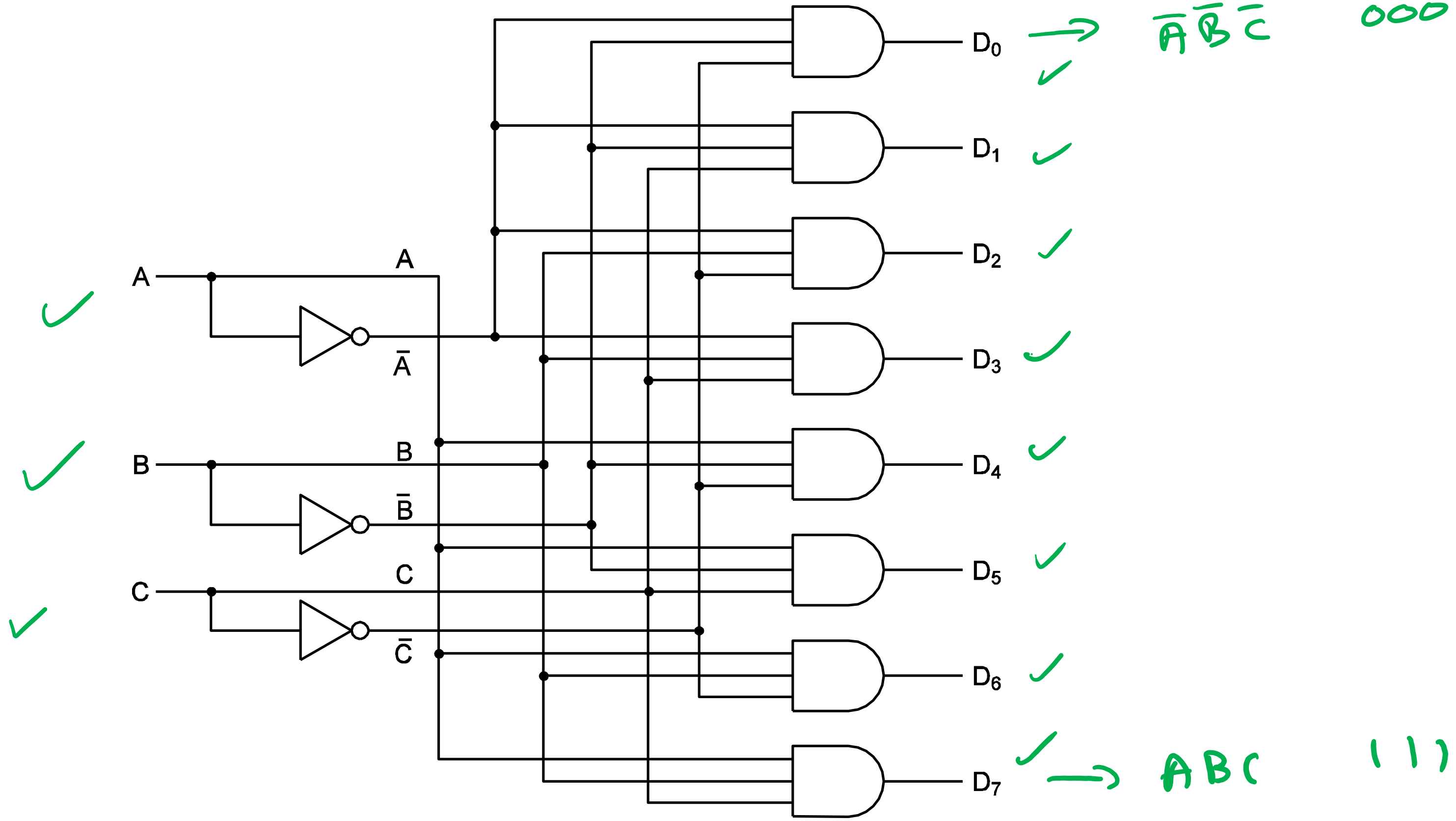
$n : 2^n$ decoder

2 : 4

3 : 8

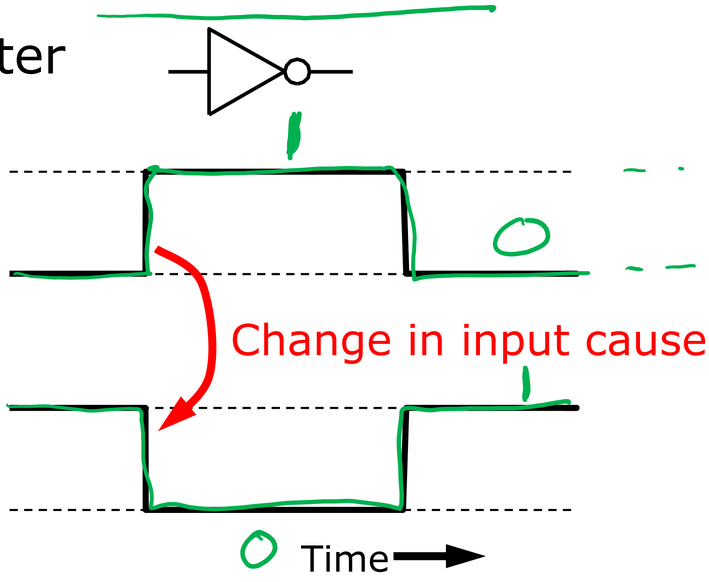
Minterm

3-to-8 Decoder Logic Circuit Implementation



Another Logic Representation: Timing Diagram

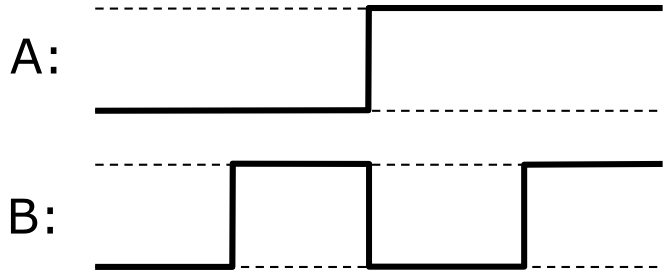
- Consider an inverter



- Like a truth table, but graphical format
  - Input waveforms show all possible combinations

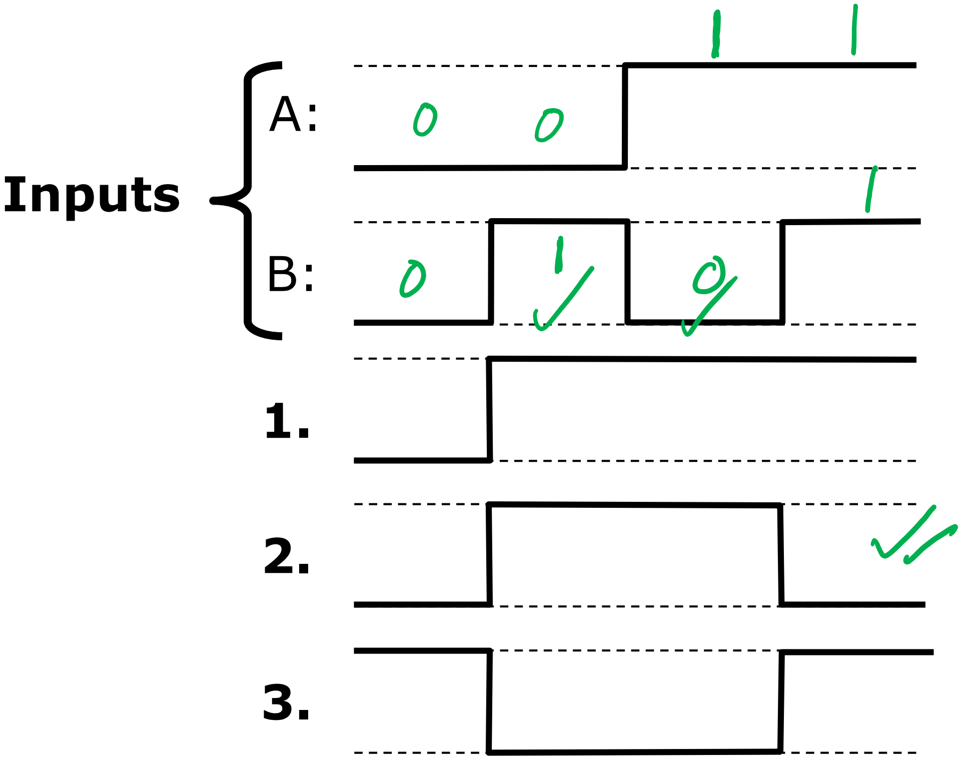
Consider a 2-input gate

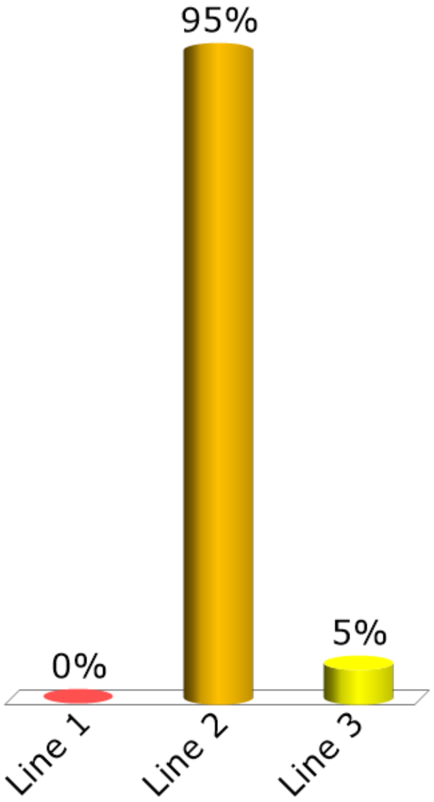
- Possible inputs



- All combinations of inputs are covered

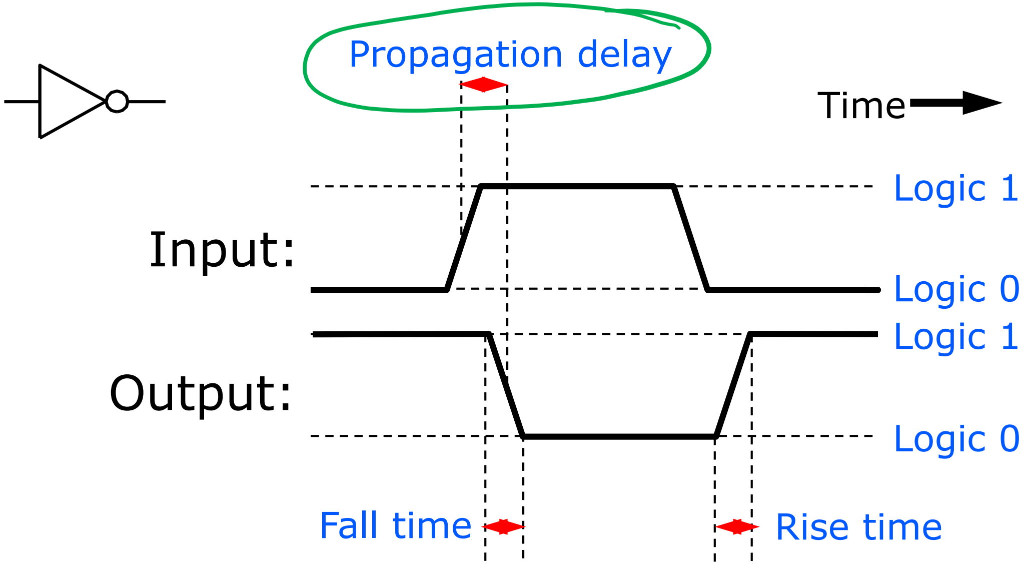
Which of the following is a timing diagram representation for an XOR gate?

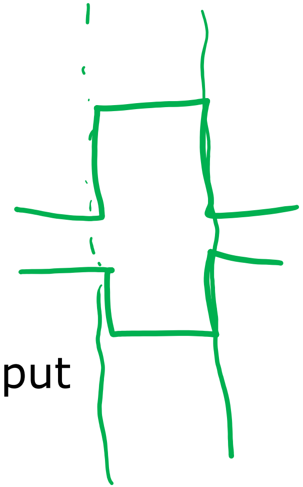




8

Gates Aren’t Perfect …
The Reality of Timing





- **Propagation delay** – time for change in input to affect output
- **Fall time** – time taken for output to fall from 1 to 0
- **Rise time** – time for output to rise from 0 to 1

Things To Do

- ✓ Complete quiz 1 by Friday 4pm this week
- Homework – Lab 4 preparation
  - By Lab 4 sessions on week 3
- ✓ Attempt the posted exercises and self-check questions
- Any doubts, use course consultation (need to book a time) or ask after the lecture.