

# Multiplexer & Demultiplexer

Lecture 7

# Outline

- How multiplexers work
- Multiplexer as a logic building block
- How demultiplexers work

# Multiplexer (MUX)

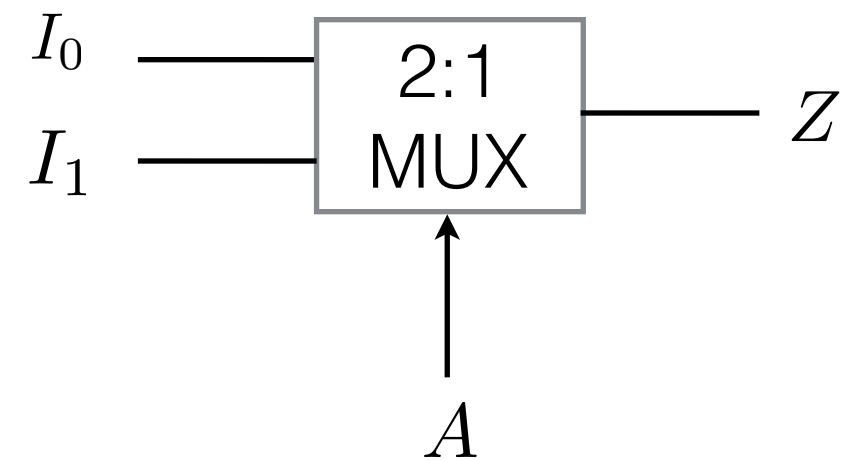
- เป็นวงจร Combinational Logic ที่ประกอบไปด้วย
  - $2^n$  Data Inputs
  - $n$  Control Inputs
  - 1 Data Output
- ค่าของ Control Input จะเป็นตัวเลือกที่จะผ่านค่าของ Data Input เพียง 1 ตัวไปยัง Output
- เราอาจเรียก MUX ว่า Selector ก็ได้

# 2:1 MUX

- MUX อย่างย่อที่สุดคือแบบ 2:1 MUX

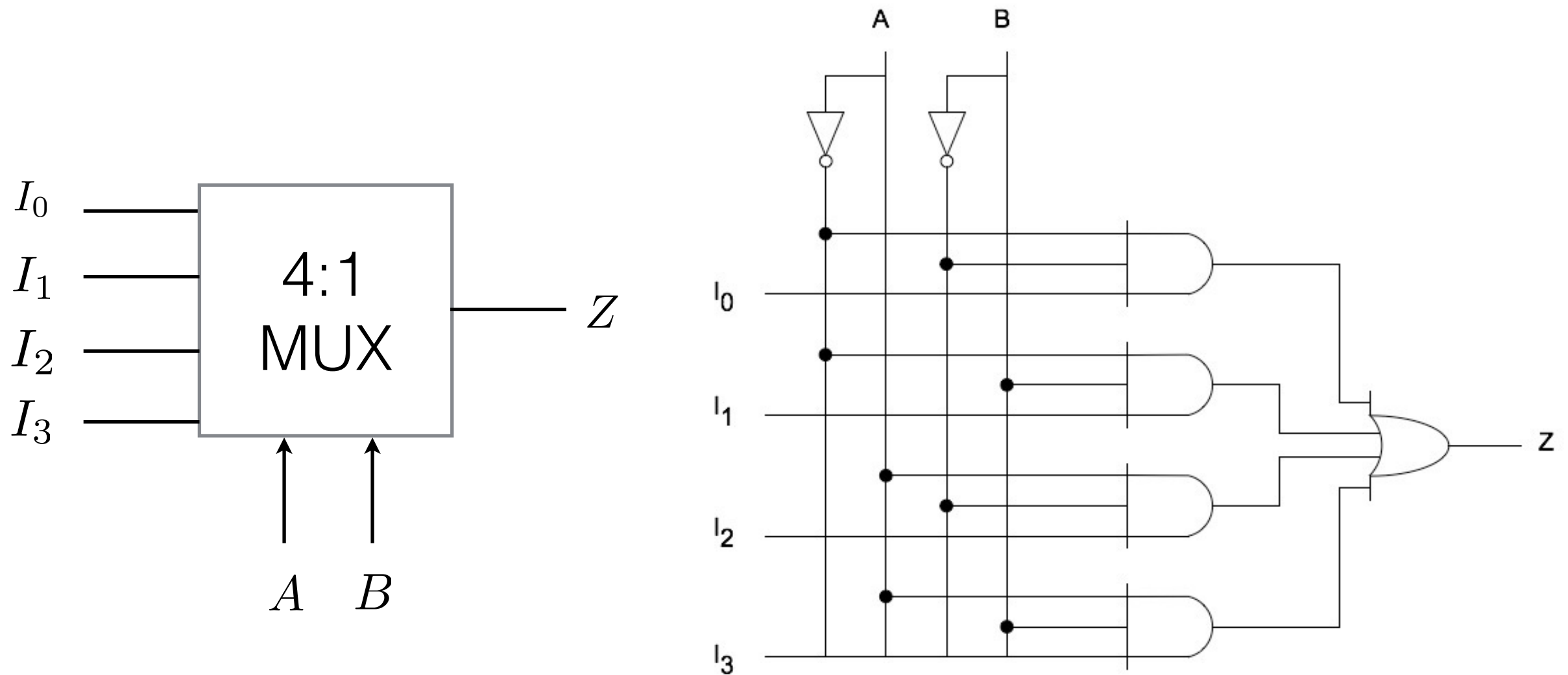
A	Z
0	$I_0$
1	$I_1$

$I_1$	$I_0$	A	Z
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



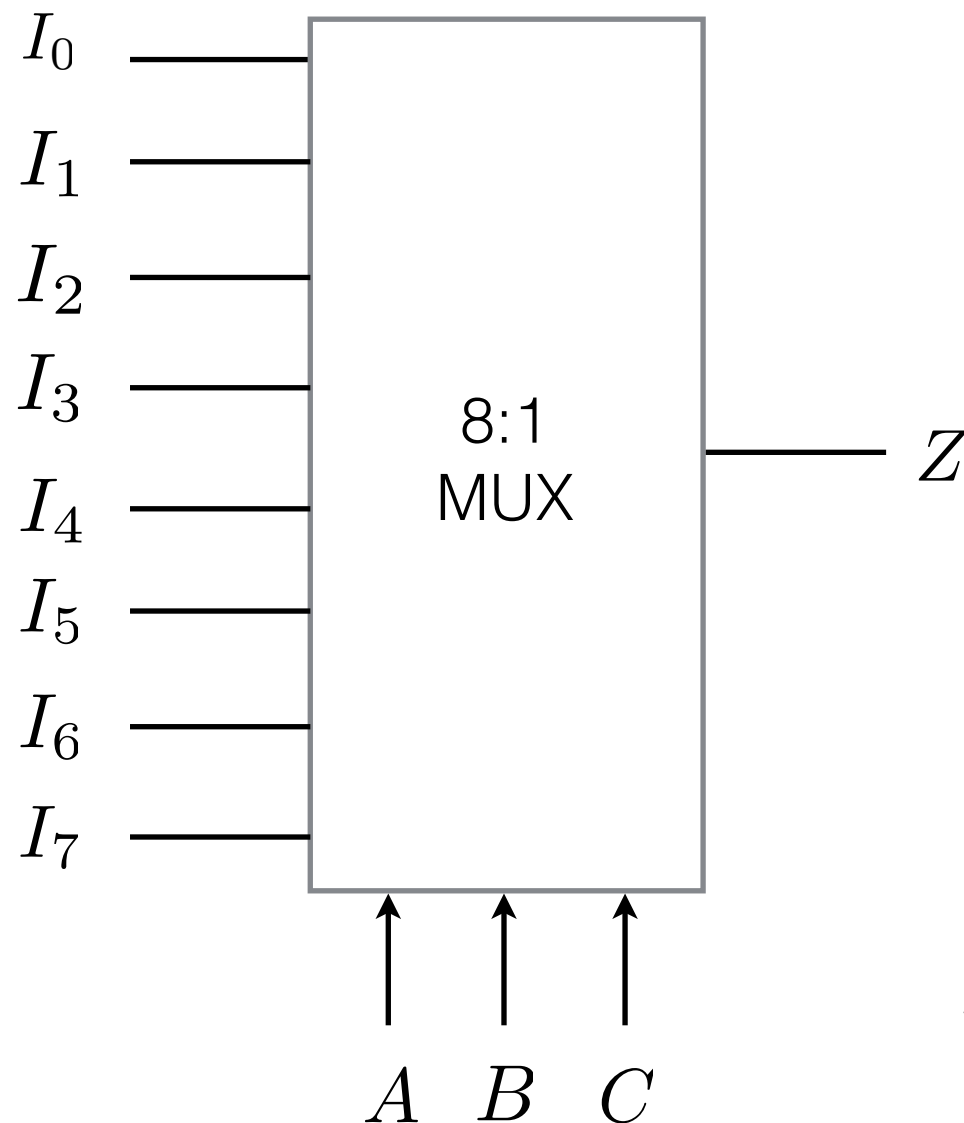
$$Z = \overline{A}I_0 + AI_1$$

# 4:1 MUX



$$Z = \overline{A}\overline{B}I_0 + \overline{A}BI_1 + A\overline{B}I_2 + ABI_3$$

# 8:1 MUX



สมการบูลีนสำหรับ 8:1 MUX

$$Z = \overline{A}\overline{B}\overline{C}I_0 + \overline{A}\overline{B}CI_1 + \overline{A}B\overline{C}I_2 + \cdots + ABCI_7$$

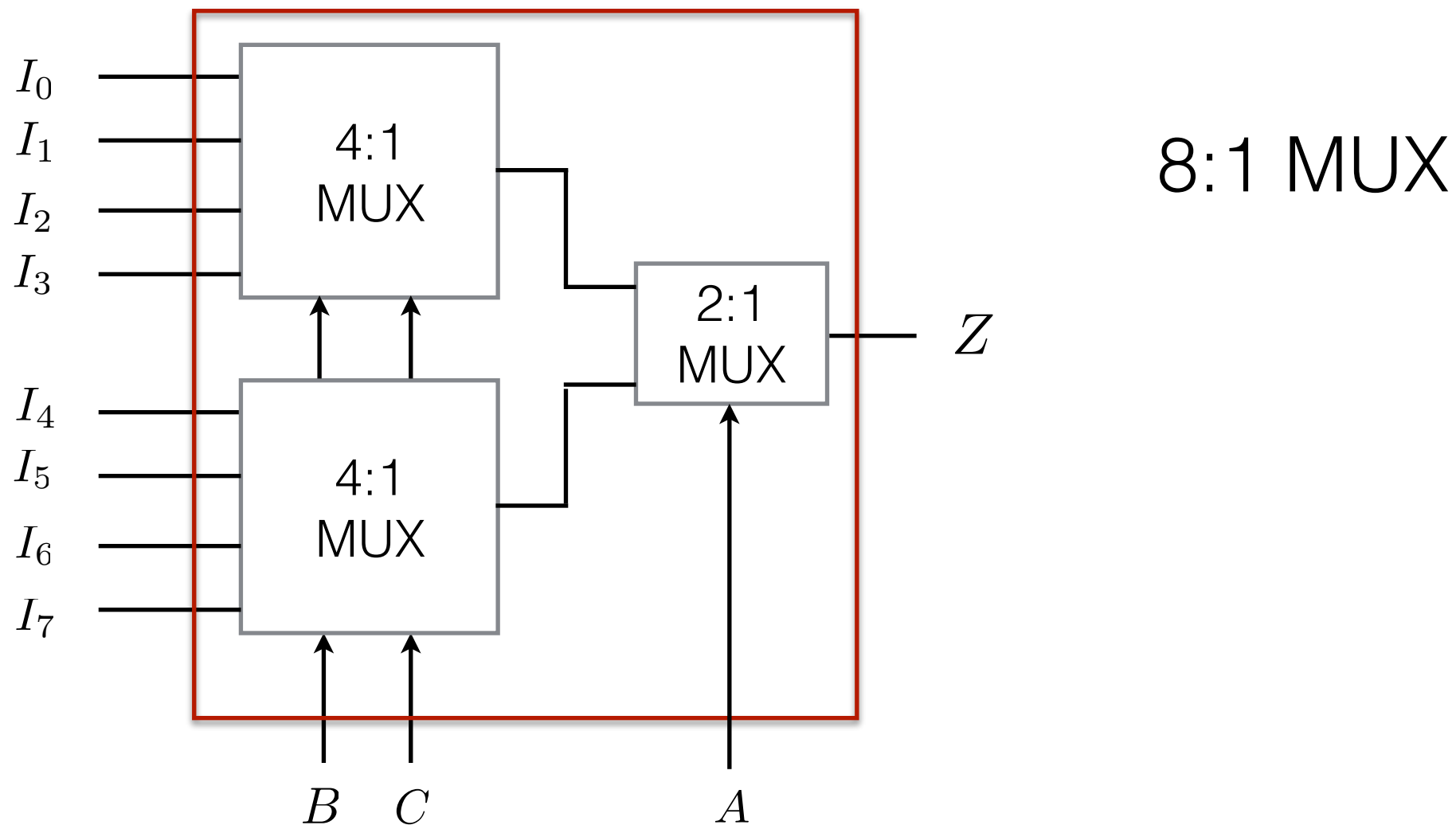
สมการบูลีนสำหรับ  $2^n:1$  MUX

$$Z = \sum_{k=0}^{2^n-1} m_k \cdot I_k$$

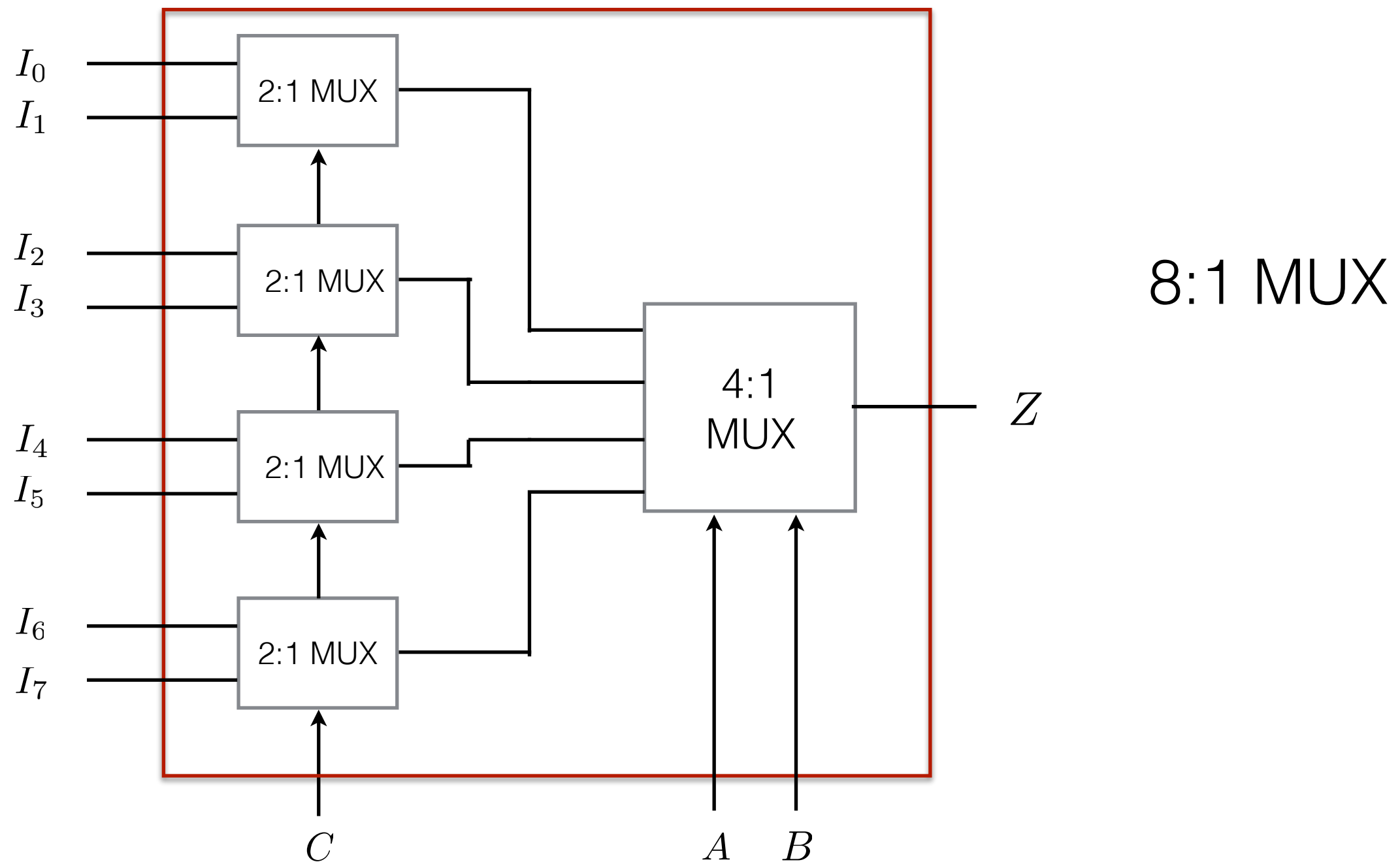
โดยที่  $m_k$  คือ k-th minterm ของ control input

# Building Larger MUX

- เราสามารถสร้าง MUX ที่ใหญ่ขึ้นจาก MUX ขนาดย่อยได้



# Building Larger MUX (2)





# Example

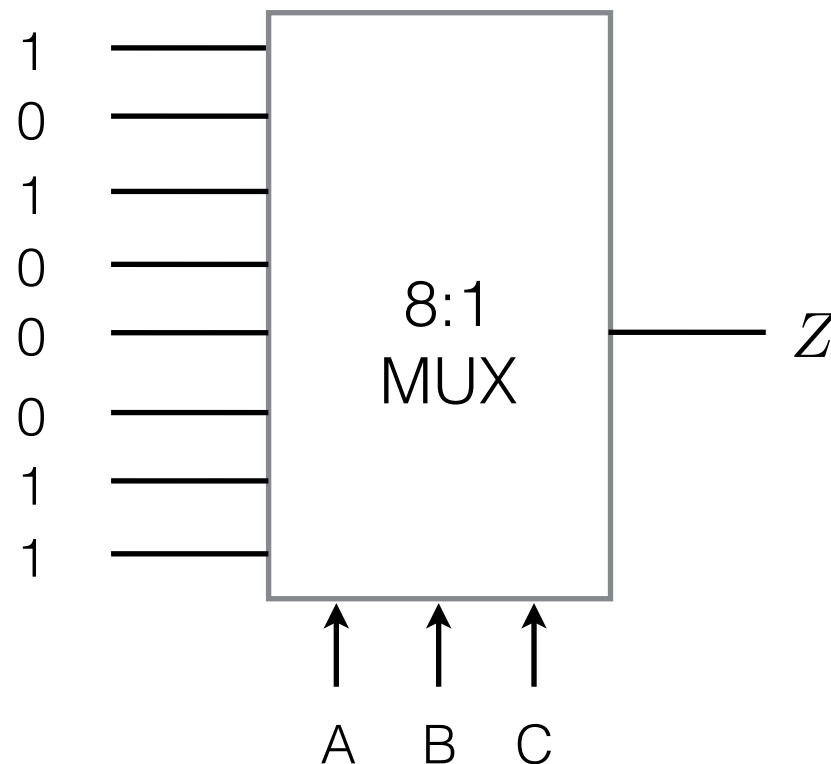
- จงสร้าง 4:1 MUX จาก 2:1 MUX

# MUX as a Logic Building Block

- เราสามารถใช้ MUX สร้างฟังก์ชันตรรกต่างๆ ที่อยู่ในรูป SOP ได้
- โดยเชื่อม  $I_i$  ไปที่ลอจิก 1 ถ้าฟังก์ชันนั้นมี minterm  $m_i$  และเชื่อมอินพุตอื่นไปที่ ลอจิก 0

- เช่น

$$F(A, B, C) = m_0 + m_2 + m_6 + m_7$$

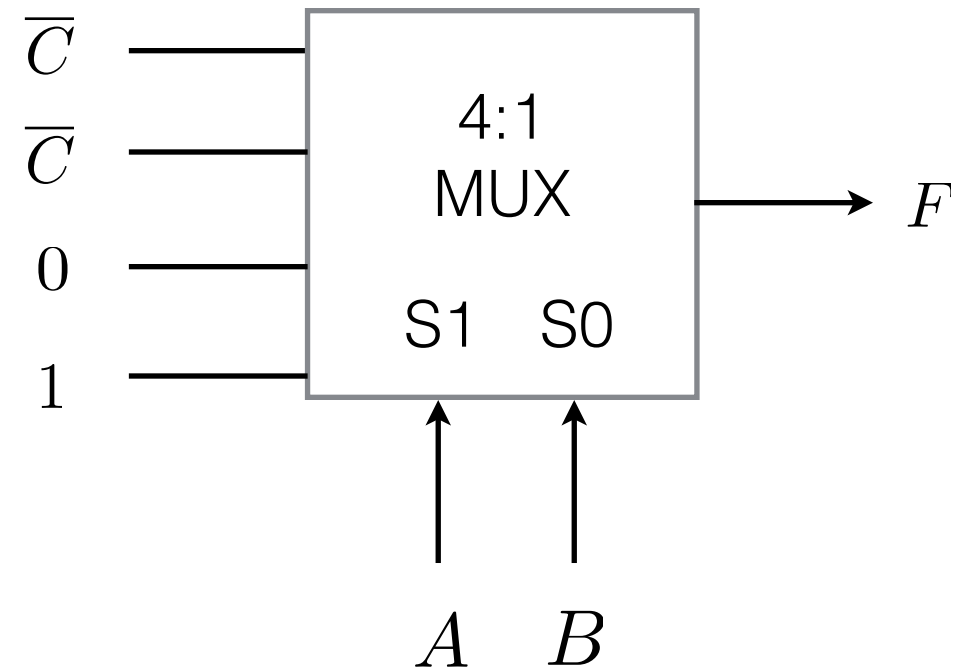


# Smaller Design

- เราสามารถสร้างวงจรที่ซับซ้อนน้อยลงได้โดยอาศัยการจัดกลุ่มที่ดี

$$F = \overline{A}\overline{B}\overline{C} + \overline{A}B\overline{C} + A\overline{B}(0) + AB(1)$$

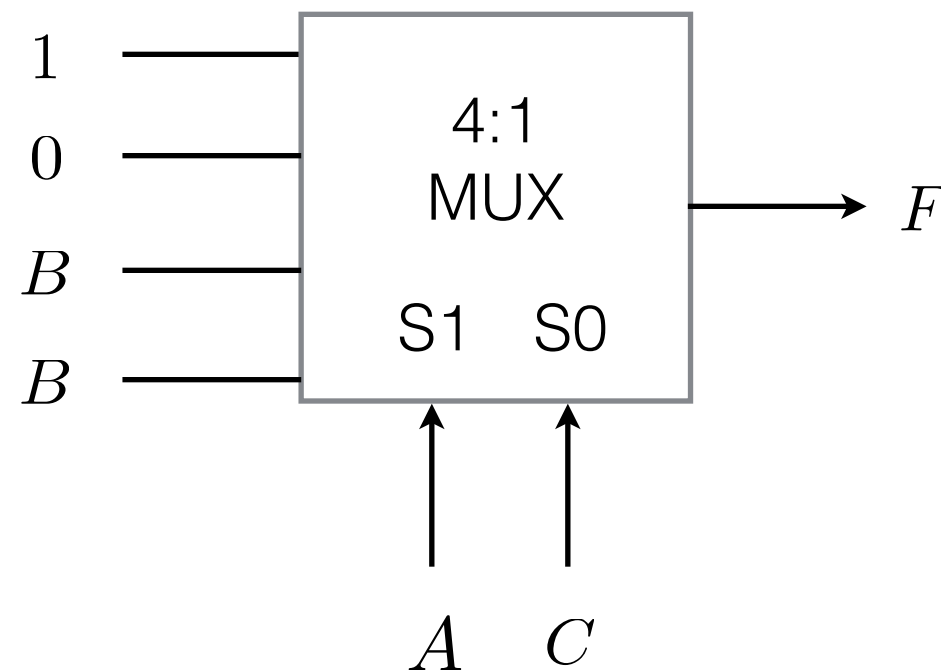
A	B	C	F	
0	0	0	1	$\overline{C}$
0	0	1	0	
0	1	0	1	$\overline{C}$
0	1	1	0	
1	0	0	0	0
1	0	1	0	
1	1	0	1	1
1	1	1	1	



# Smaller Design (2)

- ไม่ต้องใช้ inverter

$$F = \overline{A}\overline{C}(1) + \overline{A}C(0) + A\overline{C}B + ACB$$



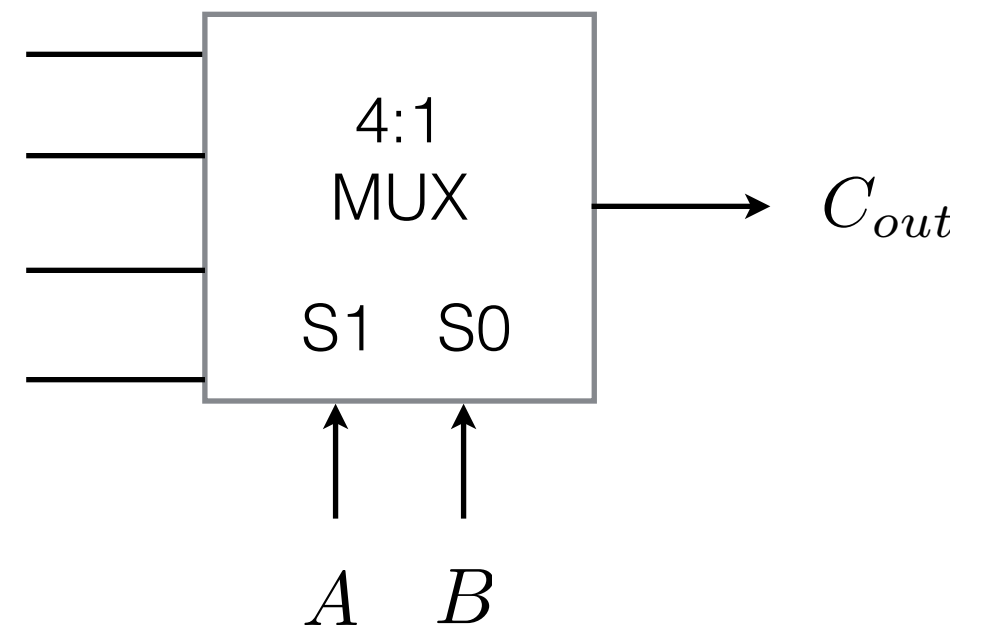
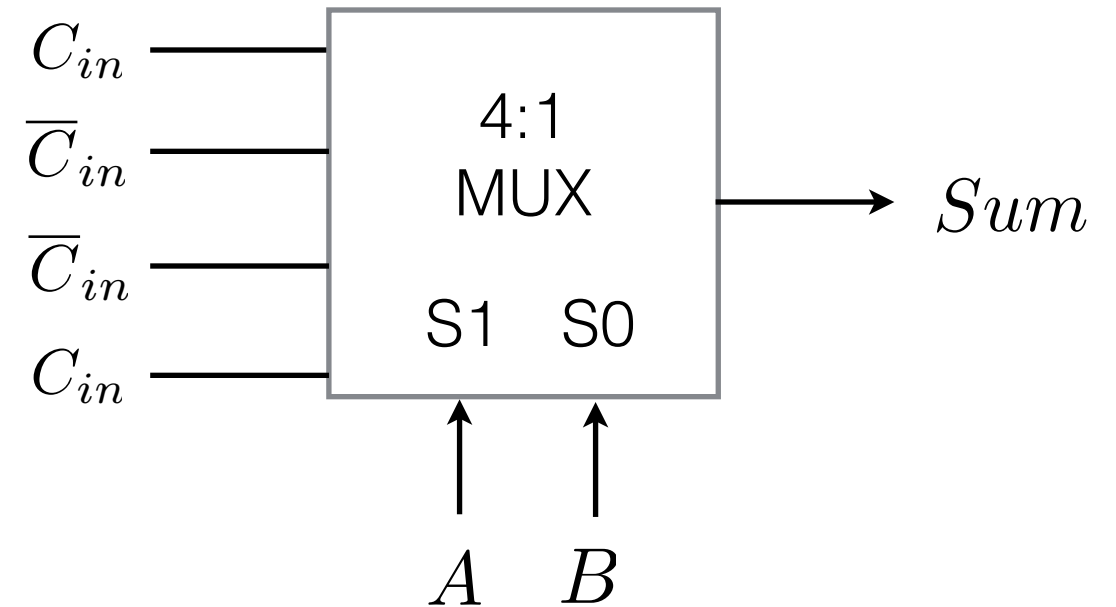
# General Principles in Circuit Design with MUX

- สำหรับฟังก์ชันที่มีอินพุตจำนวน  $n$  ตัว เลือกอินพุตจำนวน  $n-1$  ตัว มาใช้เป็น control inputs ของ MUX
- อินพุตอีก 1 ตัวที่เหลือของฟังก์ชันจะถูกใช้เป็น data input ของ MUX ร่วมกับ 0 และ 1

# Example

- ใช้ MUX สร้าง Full Adder

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



# Demultiplexer (DEMUX)

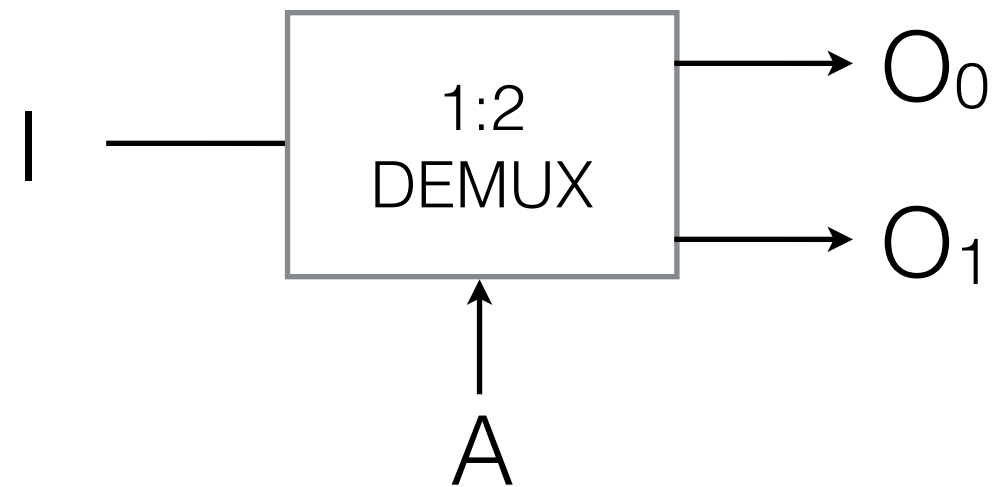
- ทำหน้าที่ตรงข้ามกับ MUX
- ประกอบไปด้วย
  - 1 data input
  - $n$  control signals
  - $2^n$  output lines

# 1:2 DEMUX

- เมื่อ  $I = 0$ , Output ทั้งคู่จะเป็น 0
- เมื่อ  $I = 1$ , Output จะขึ้นอยู่กับ  $A$

$$O_0 = \bar{A}I$$

$$O_1 = AI$$





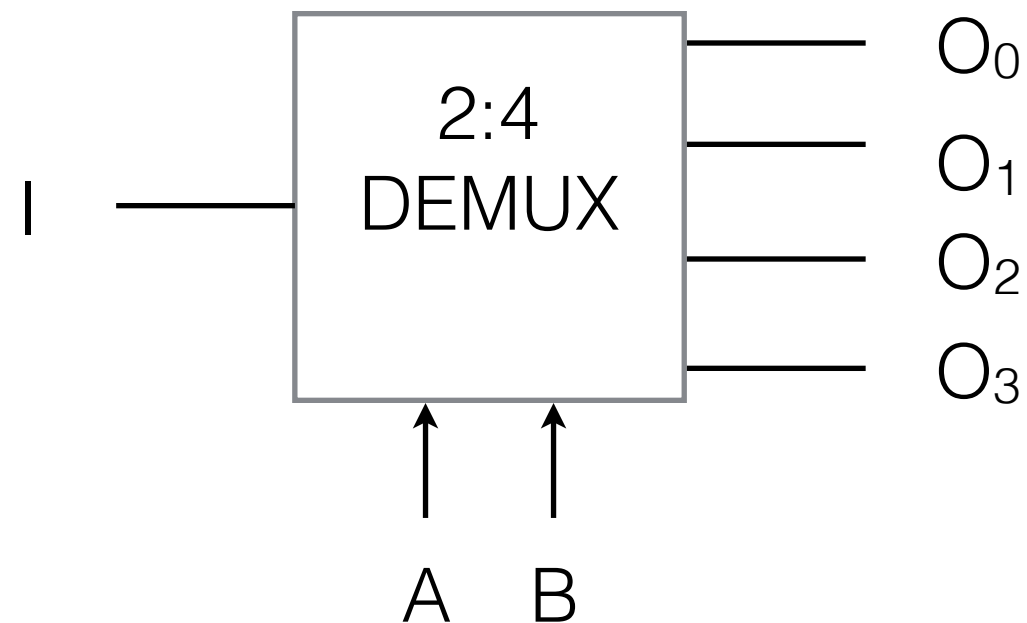
# 2:4 DEMUX

$$O_0 = \bar{A} \bar{B} I$$

$$O_1 = \bar{A} B I$$

$$O_2 = A \bar{B} I$$

$$O_3 = A B I$$



# 3:8 DEMUX

