

Multiplexer & Demultiplexer

Lecture 7

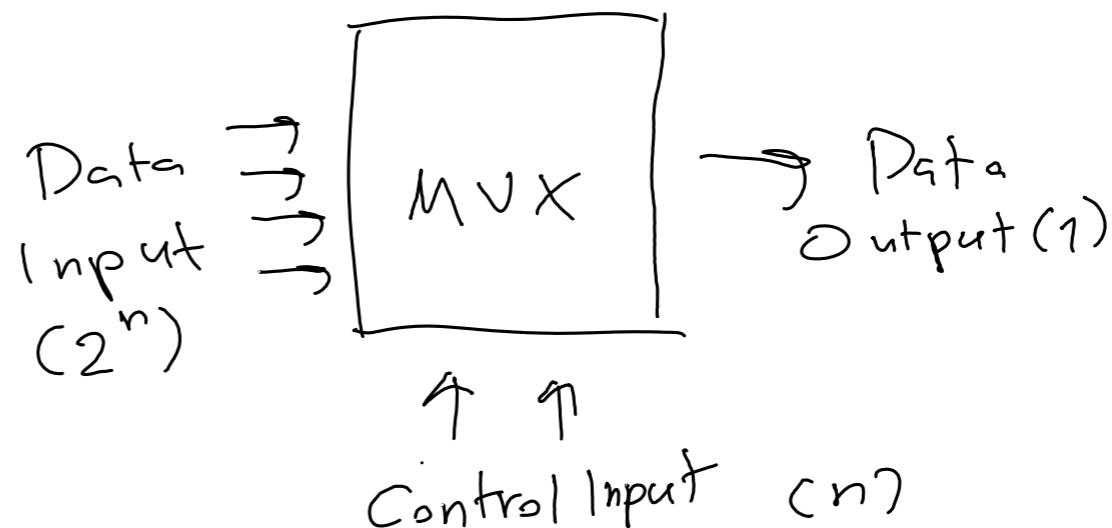
Outline

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

Multiplexer (MUX)

ex. Remote TV

- 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

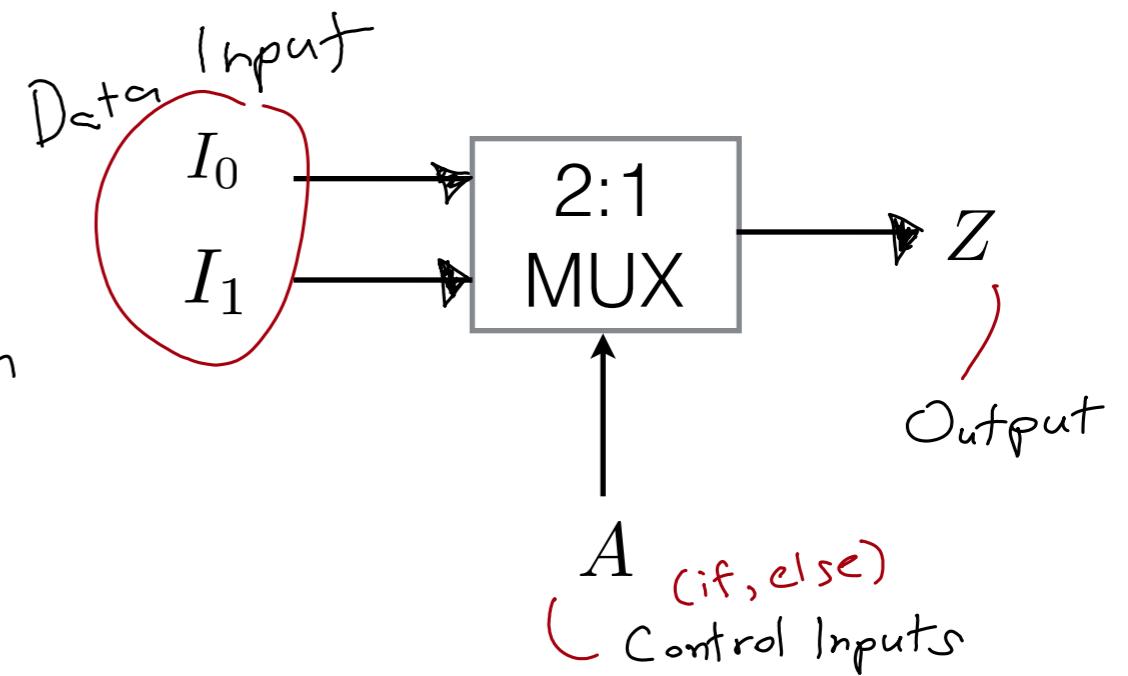


2:1 MUX

called 2 to 1 MUX

- MUX อย่างย่อยที่สุดคือแบบ 2:1 MUX

A	Z		
0	I ₀		
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



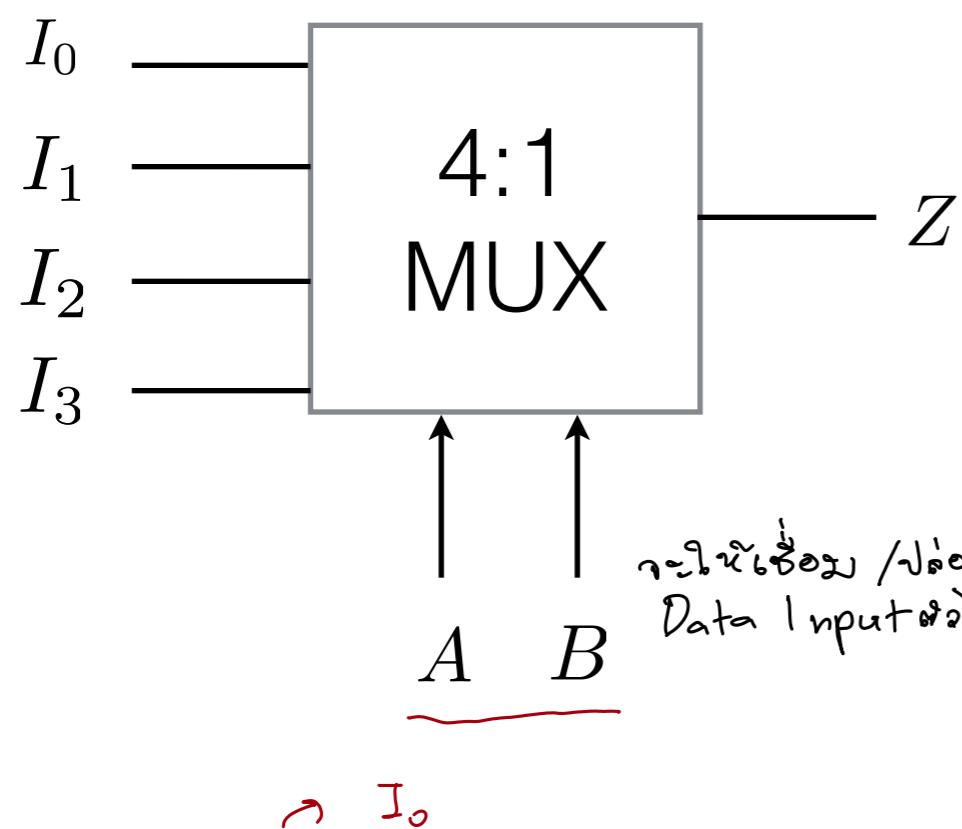
A 0 1 แล้ว I₁ ขา

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

↑ ถ้า A = 0, Output ขึ้นกับ I₀
ถ้า A = 1, Output ขึ้นกับ I₁

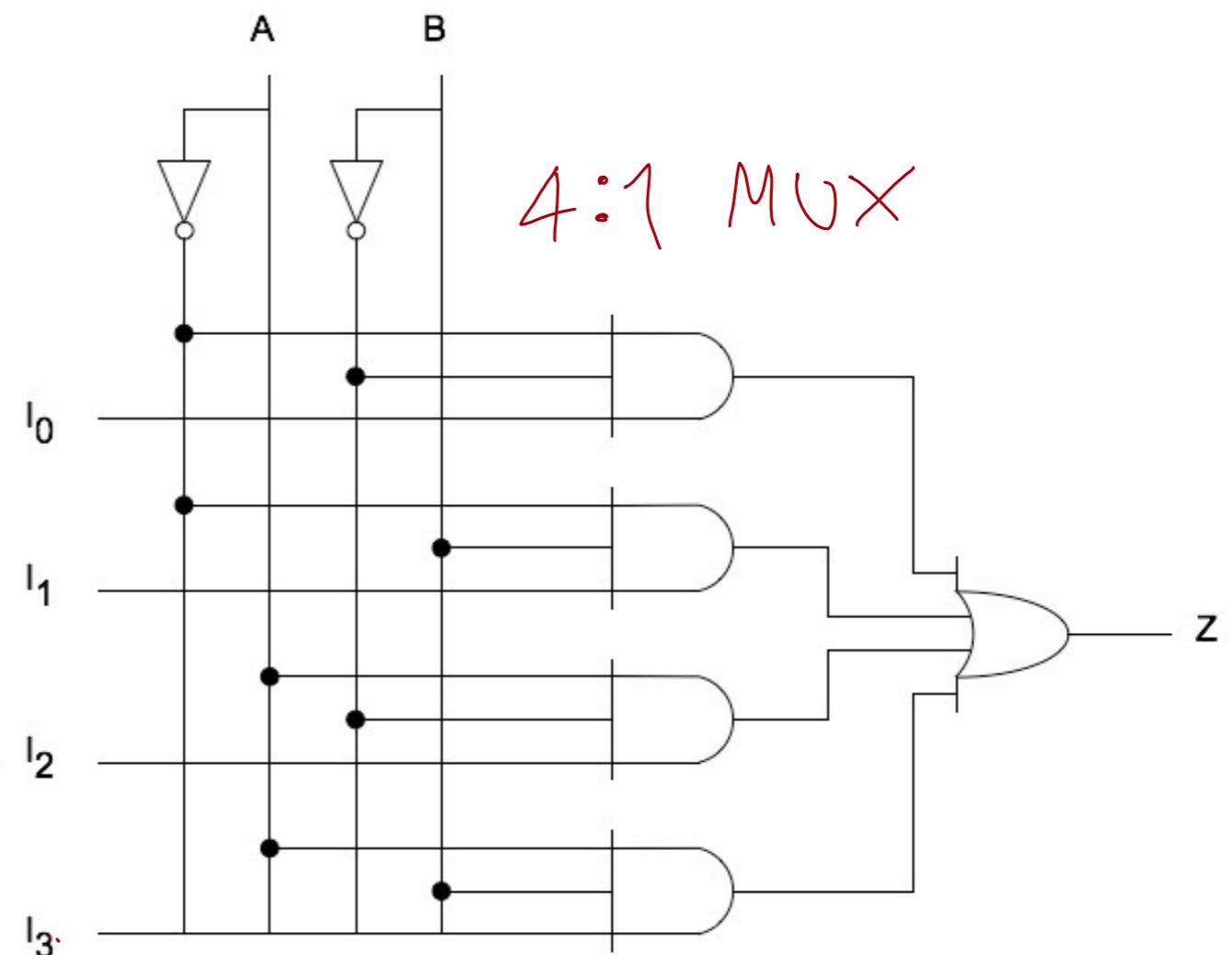
เขียน schematic
diagram

4:1 MUX



$\rightarrow I_0$
 $0\ 0$ (Binary)

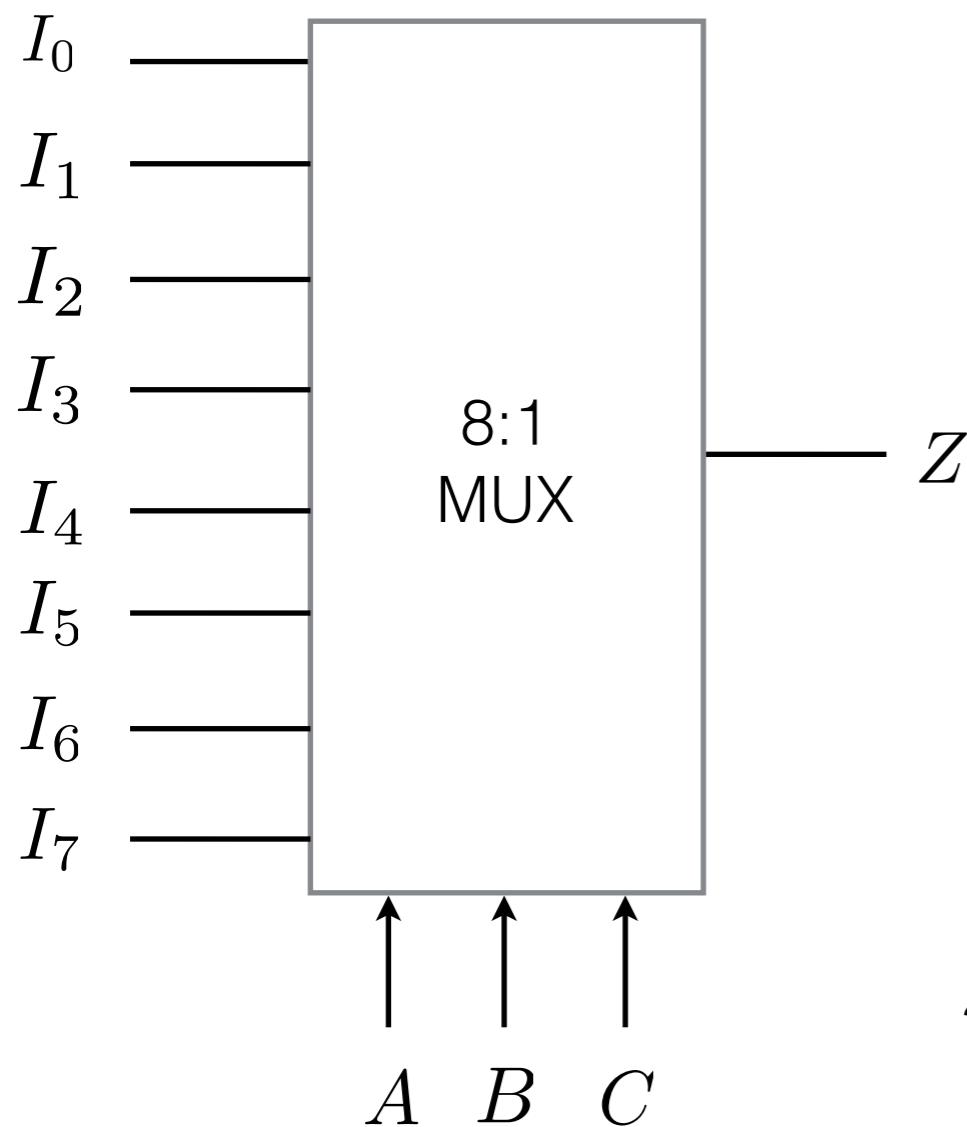
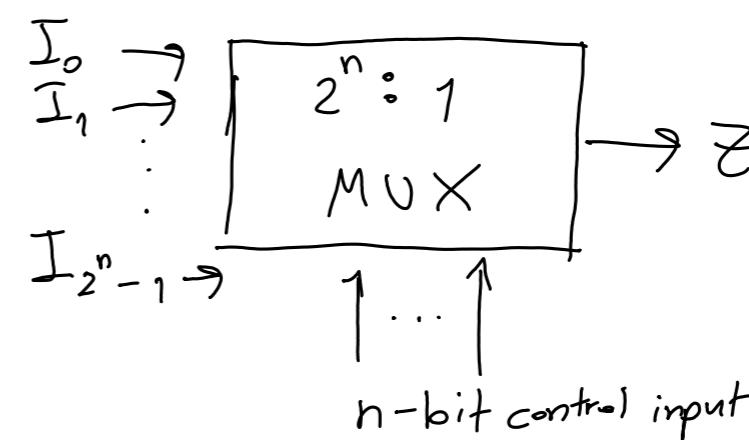
$$Z = \overline{A} \overline{B} I_0 + \overline{A} B I_1 + A \overline{B} I_2 + A B I_3$$



$1\ 1$ (3 in decimal) $\rightarrow I_3$

8:1 MUX

$2^n : 1$
MUX



สมการบูรณาการสำหรับ 8:1 MUX

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

000 001 010

สมการบูรณาการสำหรับ $2^n:1$ MUX

m_{sub-k}
 $(minterm ต่อ ก ของ n)$

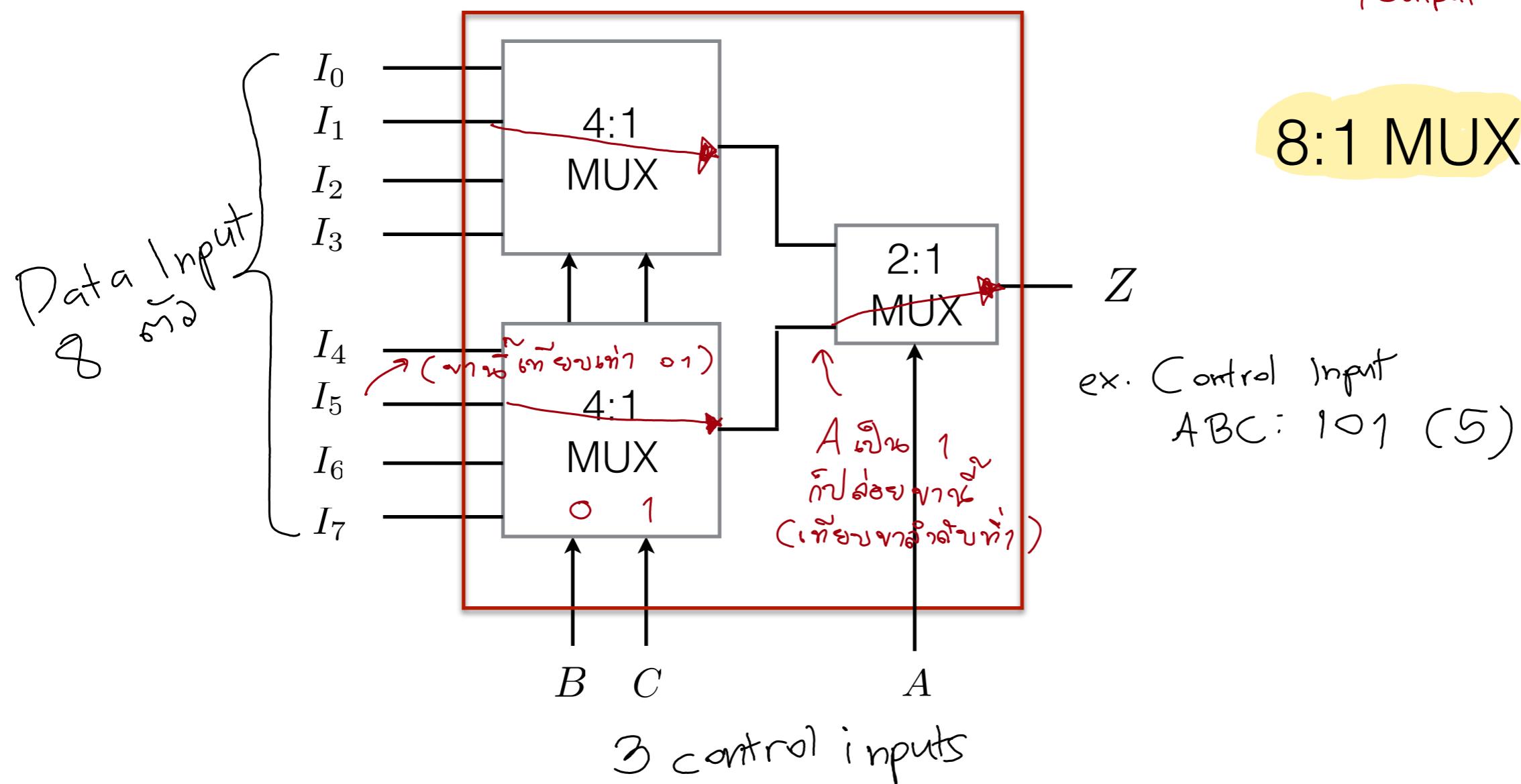
$$Z = \sum_{k=0}^{2^n-1} m_k \cdot I_k$$

โดยที่ m_k คือ k -th minterm ของ control input

Building Larger MUX

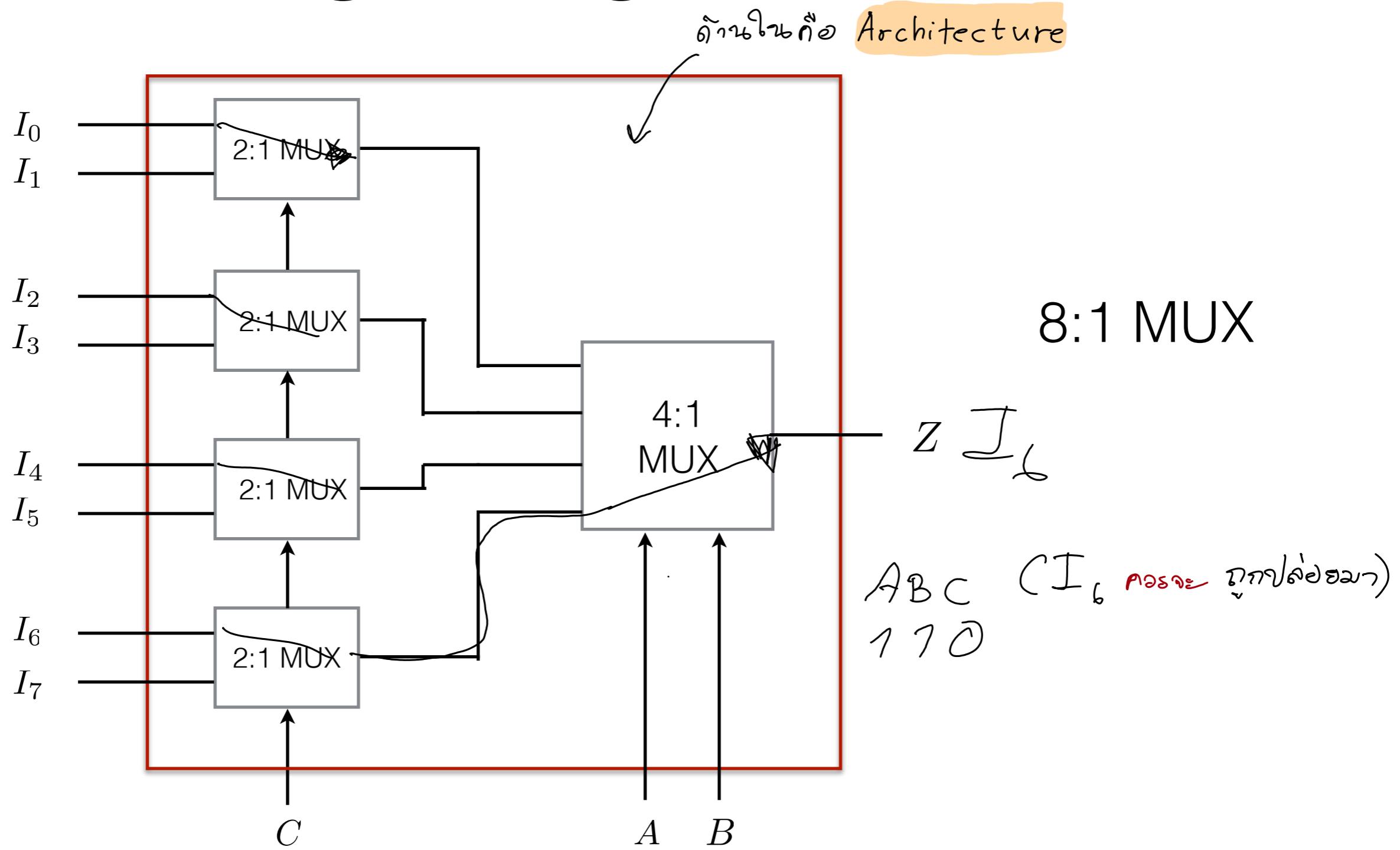
- เรารสามารถสร้าง MUX ที่ใหญ่ขึ้นจาก MUX ขนาดย่ออยได

ឧបាយមេក (ឧបករណីបច្ចេក) មុនវគ្គគំរូ 8:1 Mux ទិន្នន័យ 8 Datainput, 3 controlinput
1 Output



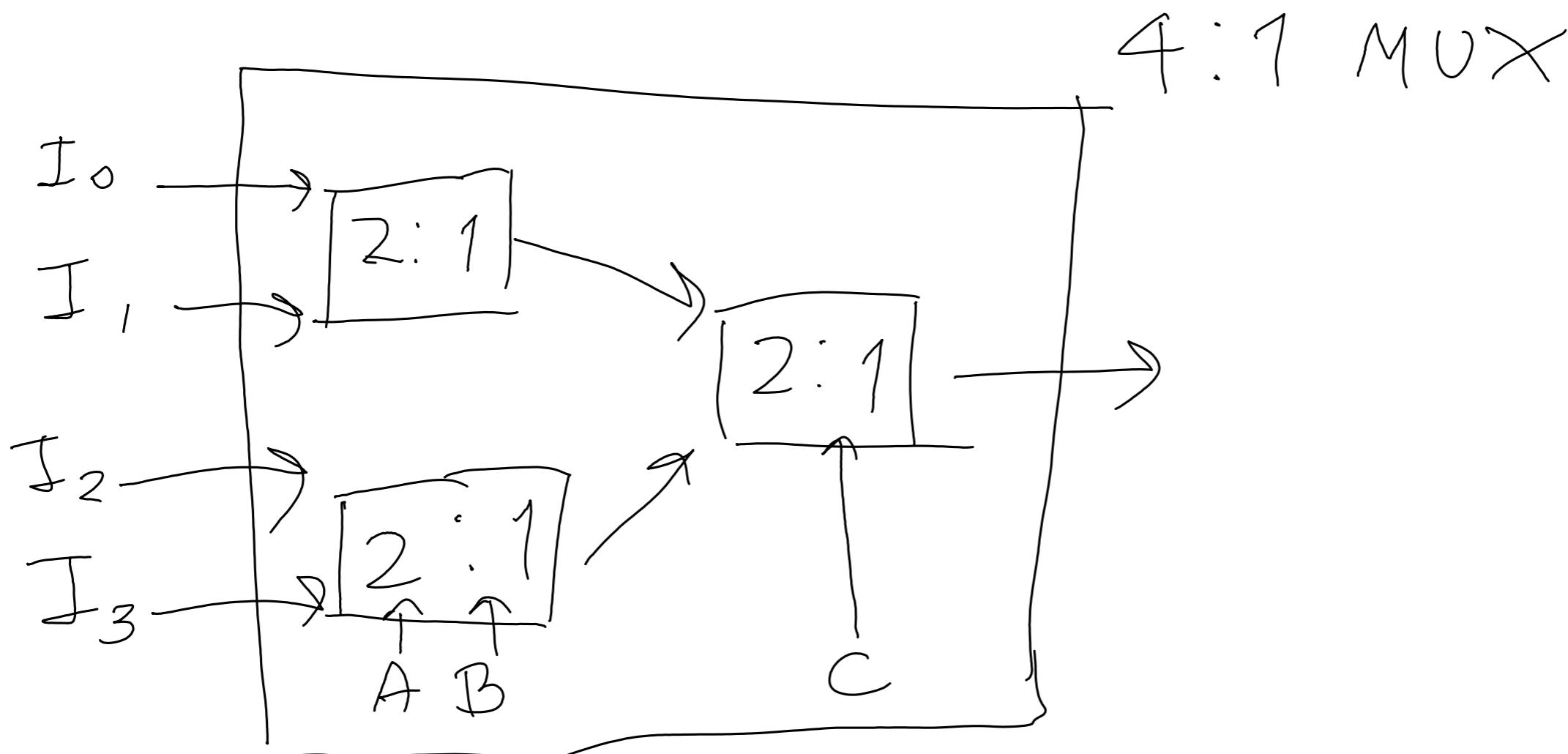
ភាពីមែន MUX ទៅលើល្អ, នៅនូវរយៈការបង្កើត MUX នឹងធ្វើការដោយប្រើប្រាស់ប្រព័ន្ធឌីជីថី

Building Larger MUX (2)



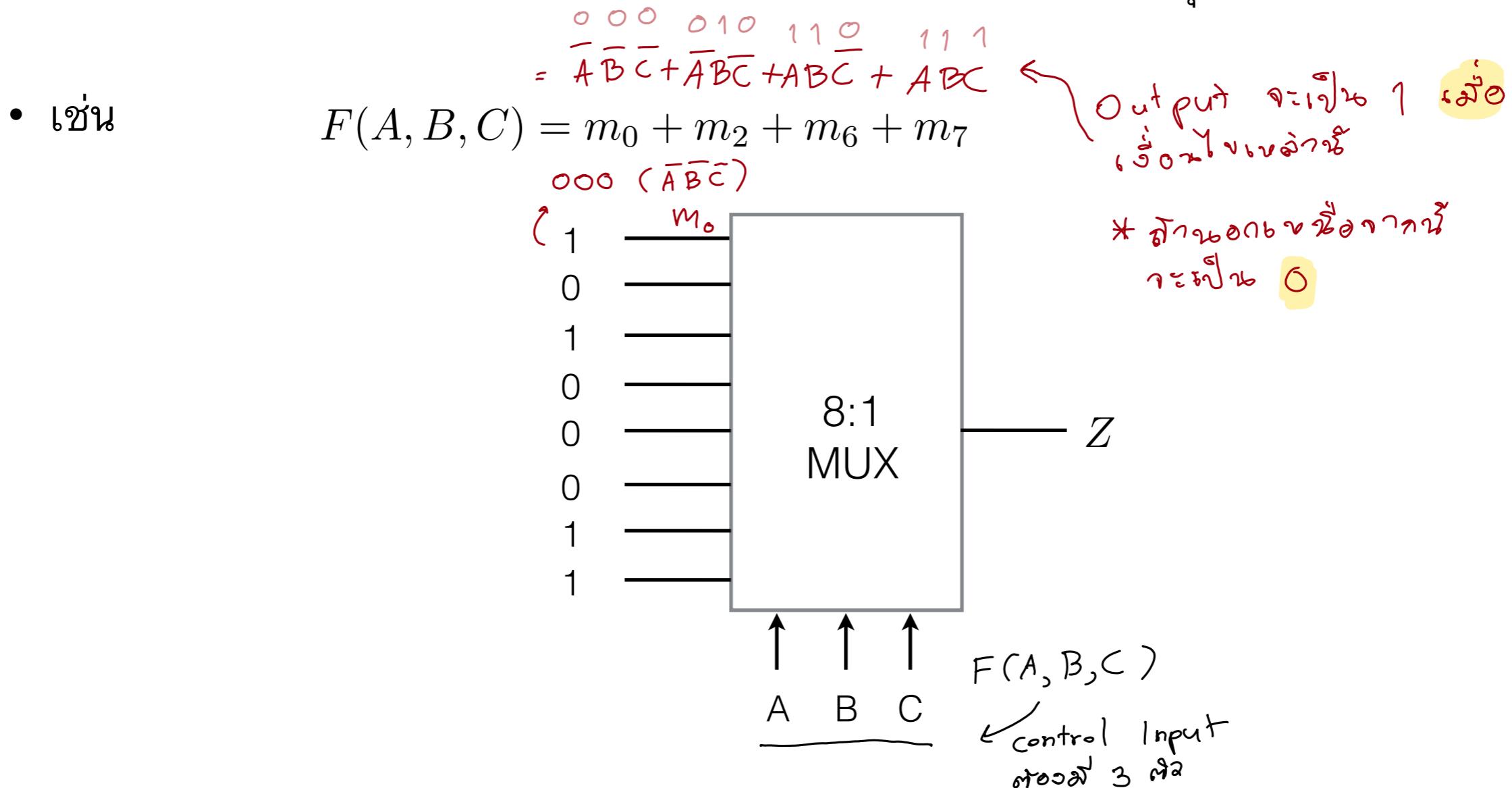
Example

- จงสร้าง 4:1 MUX จาก 2:1 MUX



MUX as a Logic Building Block

- อันนี้เป็นวิธีการอ่าน
- เรามาสามารถใช้ MUX สร้างฟังก์ชันตรรกต่างๆ ที่อยู่ในรูป SOP ได้
- โดยเชื่อม I_i ไปที่ลอจิก 1 ถ้าฟังก์ชันนั้นมี minterm m_i และเชื่อมอินพุทอื่นไปที่ ลอจิก 0



- For a function with n inputs, pick $n-1$ of them to be the control inputs of the MUX, and use the other as the data input

Smaller Design

ถ้ามี n input, เลือก $n-1$ ตัว (อัปเปอร์) มาเป็น control inputs และใช้ตัวที่เหลือเป็น data input

* 1 control input แล้ว 1 ตัวไป

- เราสามารถสร้างวงจรที่ซับซ้อนน้อยลงได้โดยอาศัยการจัดกลุ่มที่ดี

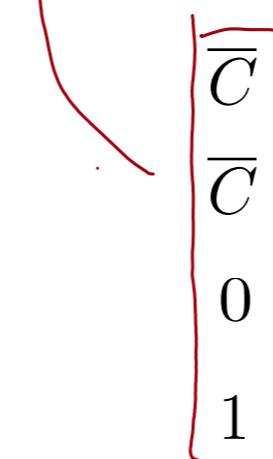
$$F = \overline{A} \overline{B} \overline{C} + \overline{A} B \overline{C} + A \overline{B} (0) + AB(1)$$

ถ้า $A, B=0$
ดูว่า F ตัวเดียว
กรณีซ้ำ \overline{C}

ก็จะหา I_1

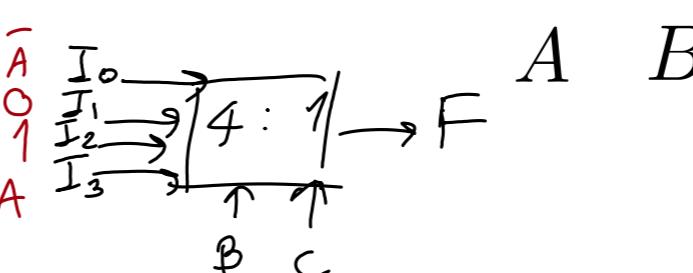
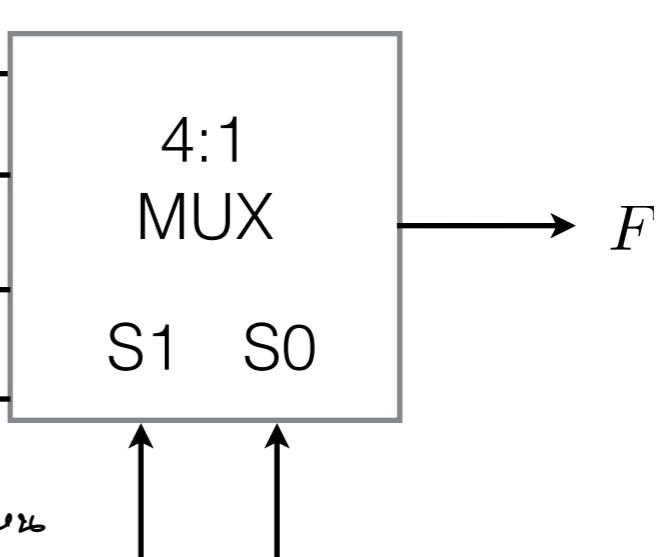
A	B	C	F	Input ตัวที่ 1 ของ F ที่มี
0	0	0	1	$m_0 \overline{C}$
0	0	1	0	
0	1	0	1	$m_2 \overline{C}$
0	1	1	0	
1	0	0	0	0
1	0	1	0	
1	1	0	1	$m_6 \overline{C}$
1	1	1	1	$m_7 \overline{C}$

ผลรวมของ m_0, m_2, m_6, m_7



Truth Table

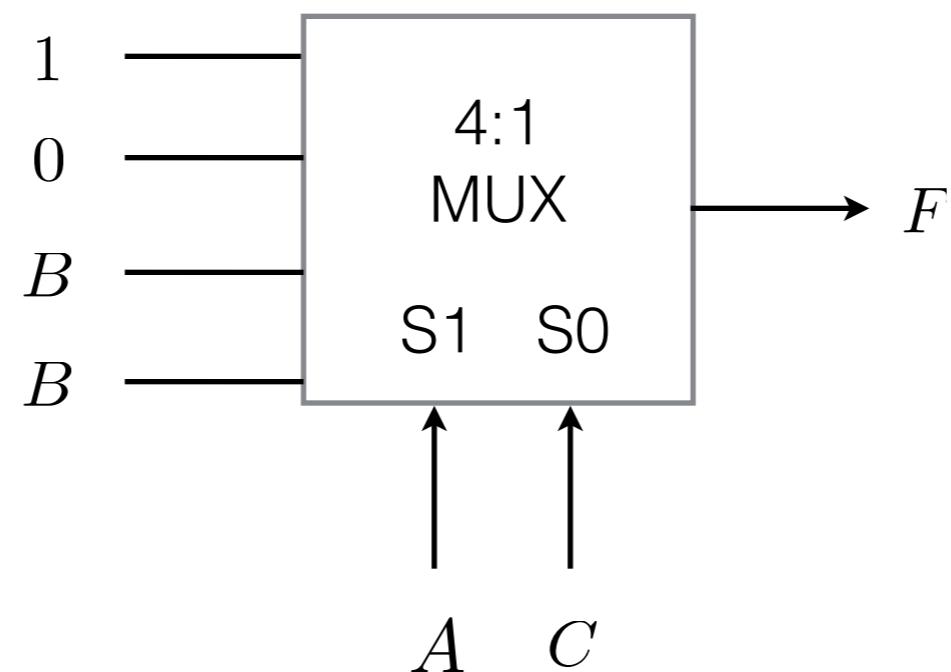
$$F(A, B, C) = m_0 + m_2 + m_6 + m_7$$



Smaller Design (2)

- ไม่ต้องใช้ inverter

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



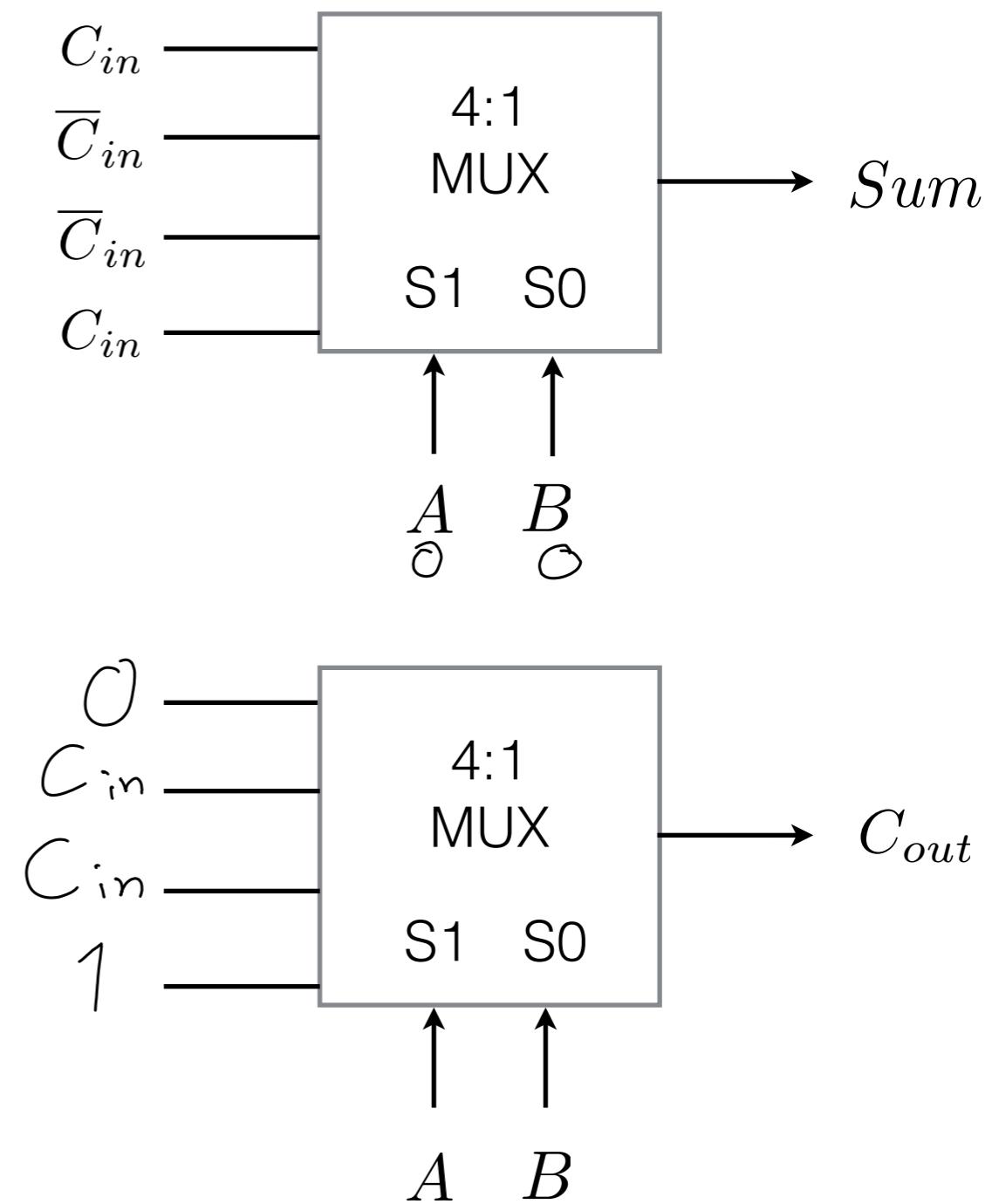
General Principles in Circuit Design with MUX

- สำหรับฟังก์ชันที่มีอินพุทจำนวน n ตัว เลือกอินพุทจำนวน $n-1$ ตัว มาใช้เป็น control inputs ของ MUX
- อินพุทอีก 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

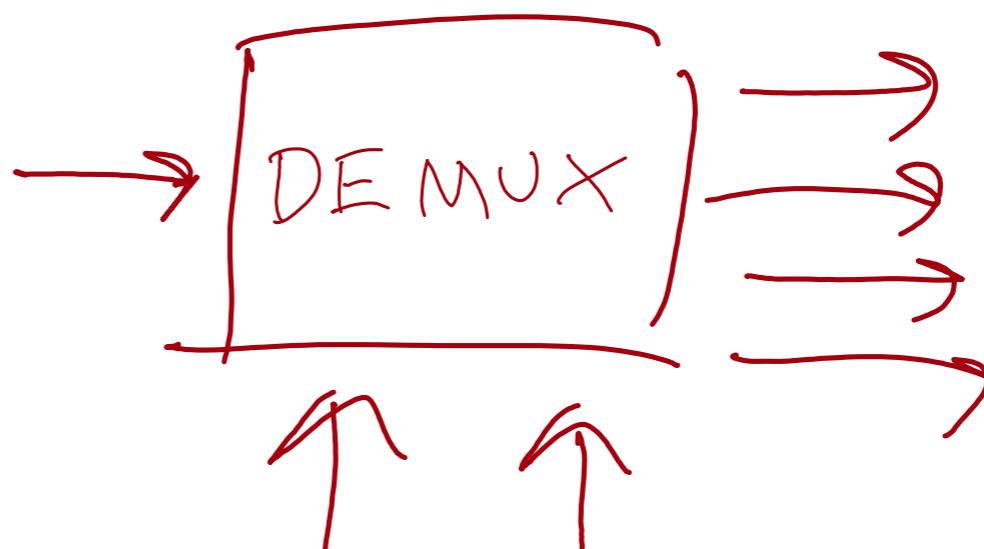


Demultiplexer (DEMUX)

- ทำหน้าที่ตรงข้ามกับ MUX

- ประกอบไปด้วย

- 1 data input
- n control signals
- 2^n output lines



1:2 DEMUX

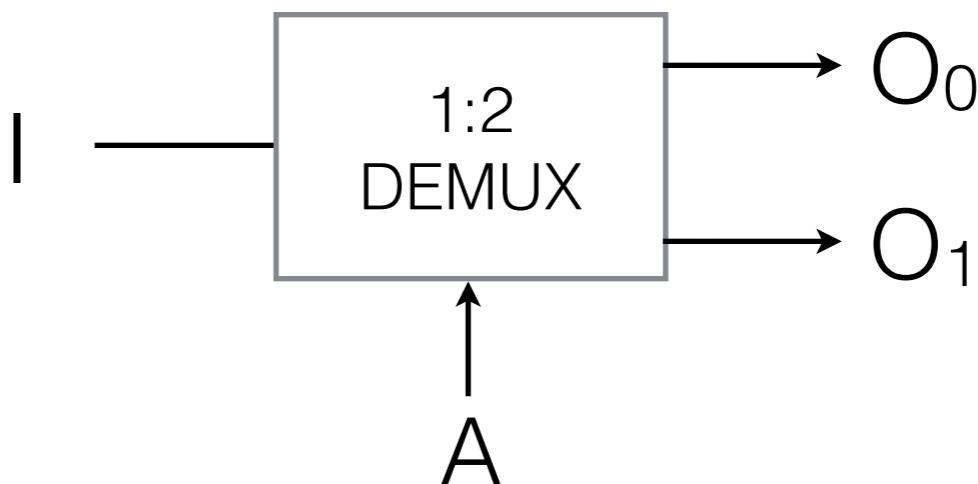
control input

Output

- เมื่อ $I = 0$, Output ทั้งคู่จะเป็น 0
 $\rightarrow 0 \text{ ឧបតករ }$
- เมื่อ $I = 1$, Output จะขึ้นอยู่กับ A

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

$$O_1 = AI$$



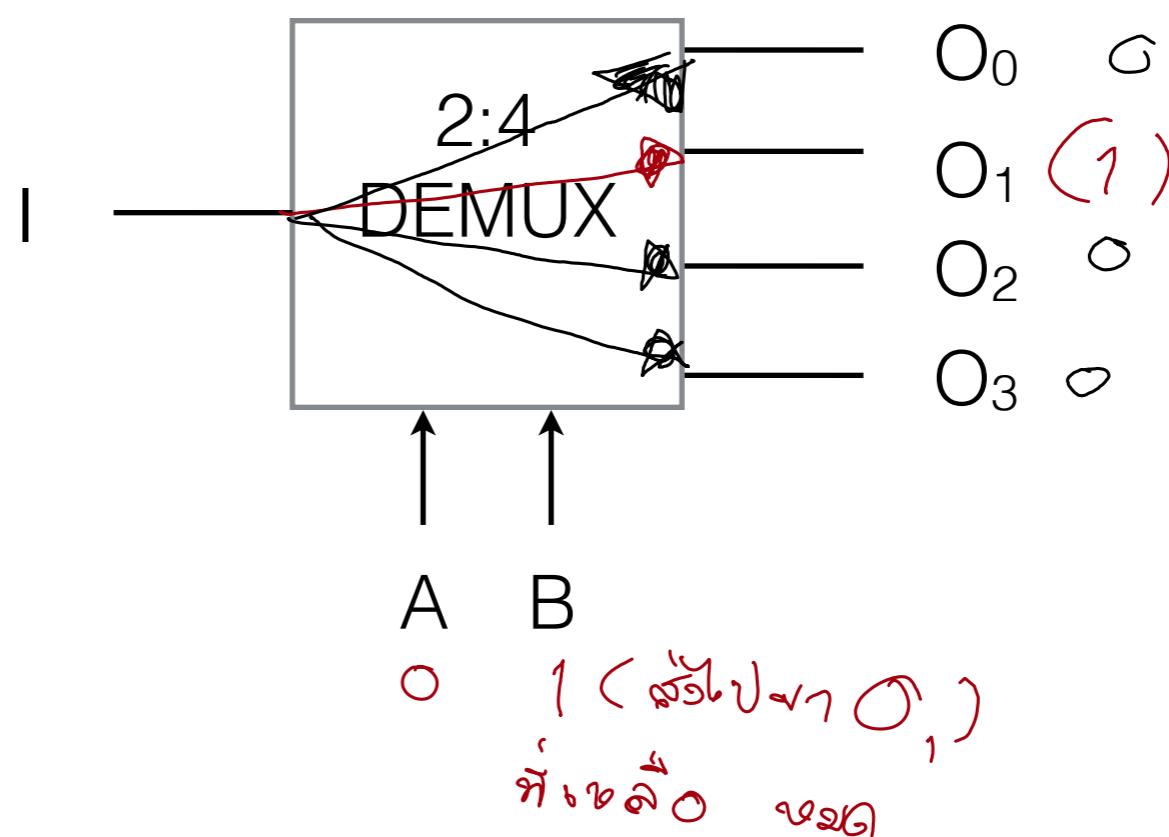
2:4 DEMUX

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

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

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

$$O_3 = A B I$$



3:8 DEMUX

