SKEE1223: Digital Electronics

10 – Medium Scale Integrated (MSI) Circuits

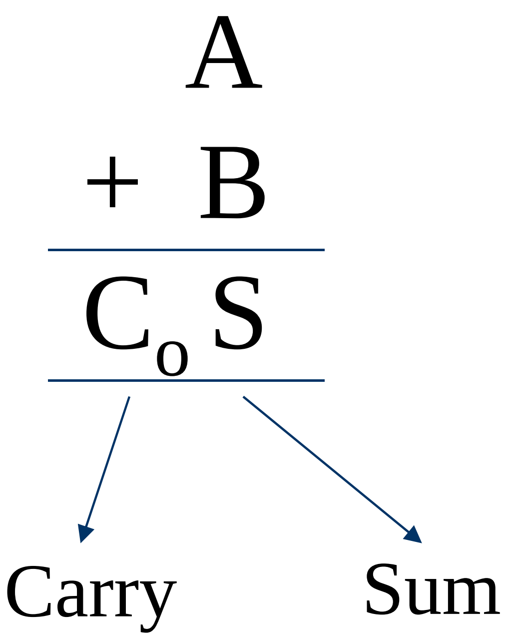
Dr Michael Tan Loong Peng
PhD (Cambridge)
Senior Lecturer
Faculty of Electrical Engineering
Universiti Teknologi Malaysia

MSI Circuits

- Multiplexers (Mux)
  - 2x1, 4x1, and 8x1 muxes
  - 74x151, 74x153, 74x157 devices
- Demultiplexers (Demux), Decoders, and Encoders
  - 74x138 and 74x139 decoders
  - Encoder, priority encoder and the 75x147 devices
  - BCD to 7-segment decoder and the 74x247 devices
  - Logic functions using muxes and decoders
- Adders and Comparators
  - Half, full and ripple carry adders
  - The 74x83 devices
  - Comparator and the 74x85 devices

Half-Adder

- Half-Adder is a circuit that add two binary numbers without input carry



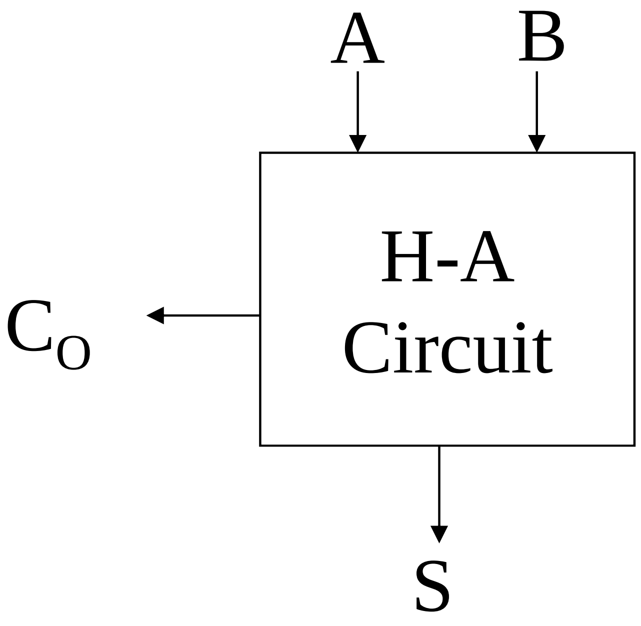
A

\+ B

C_o S

Carry

Sum



A

B

H-A Circuit

C_O

S

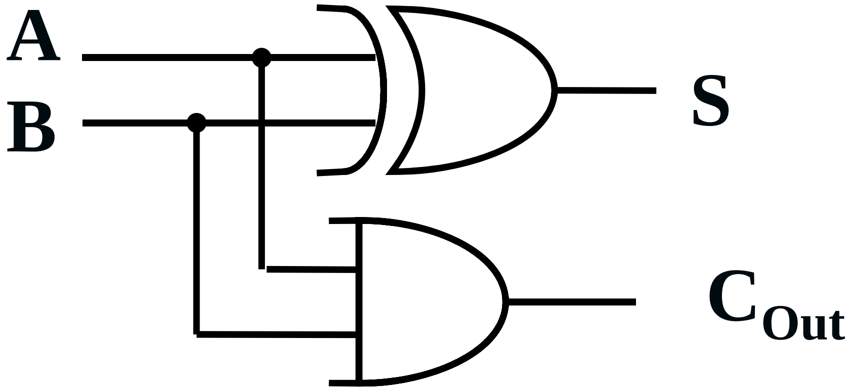
Half-Adder

- Truth table for 1-bit Half-Adder

A	B	C_{Out}	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

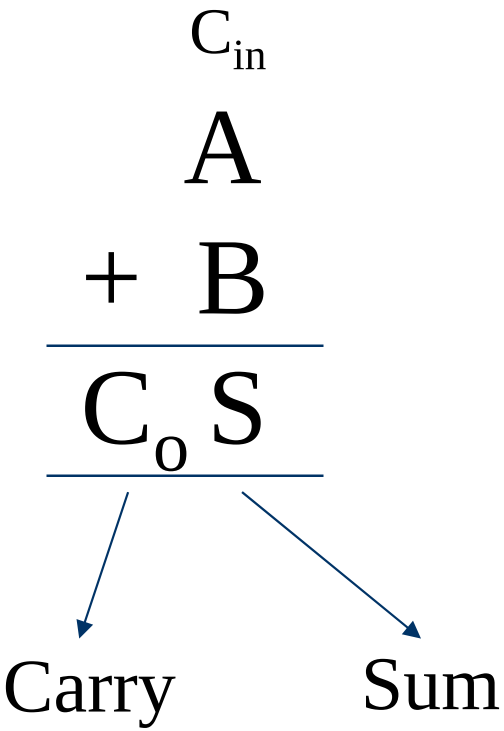
$$\mathbf{C_{out}} = A \cdot B$$

$$\mathbf{S} = A \oplus B$$



Full-Adder

- Full-Adder is a circuit that add two binary numbers with the existence of input carry



C_{in}

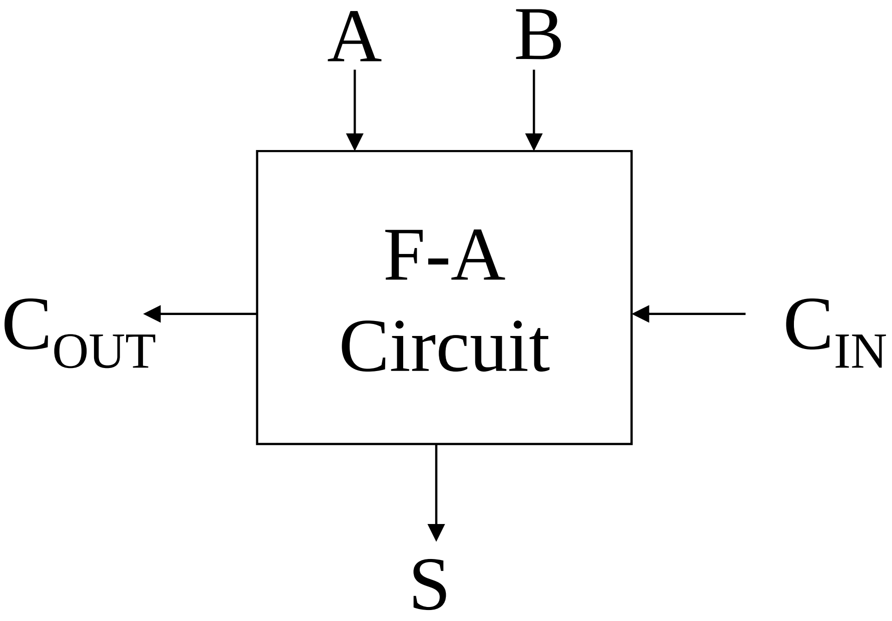
A

+ B

C_o S

Carry

Sum



A

B

F-A Circuit

C_{OUT}

C_{IN}

S

Full-Adder

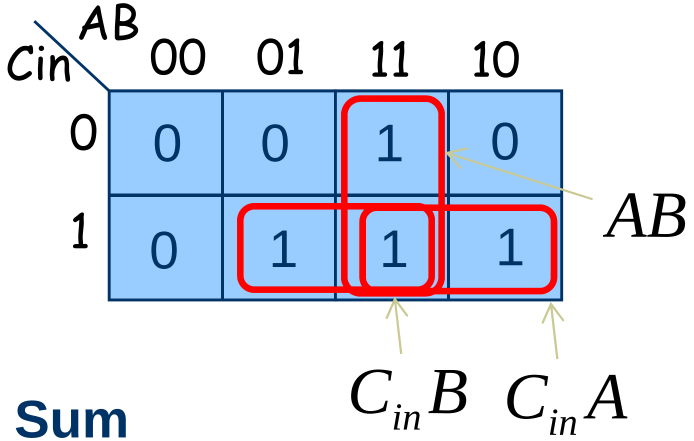
- Truth Table for 1-bit Full-Adder

A	B	C_{IN}	C_{OUT}	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Full Adder (cont.)

- Design of Full Adder

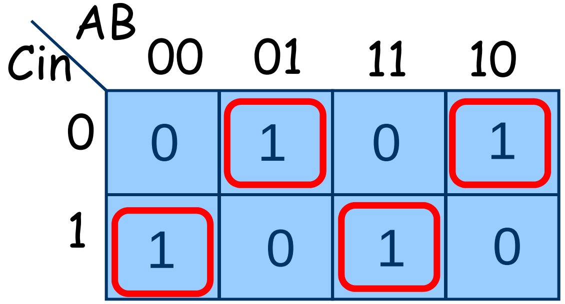
C_{out}



Cin \ AB	00	01	11	10
0	0	0	1	0
1	0	1	1	1

$$C_{out} = C_{in}B + C_{in}A + AB$$

Sum



Cin \ AB	00	01	11	10
0	0	1	0	1
1	1	0	1	0

$$S = C_{in}\bar{A}\bar{B} + C_{in}AB + \overline{C_{in}}\bar{A}B + \overline{C_{in}}A\bar{B}$$

$$= C_{in}(\bar{A}\bar{B} + AB) + \overline{C_{in}}(\bar{A}B + A\bar{B})$$

$$= C_{in}\overline{(\bar{A}B + A\bar{B})} + \overline{C_{in}}(\bar{A}B + A\bar{B})$$

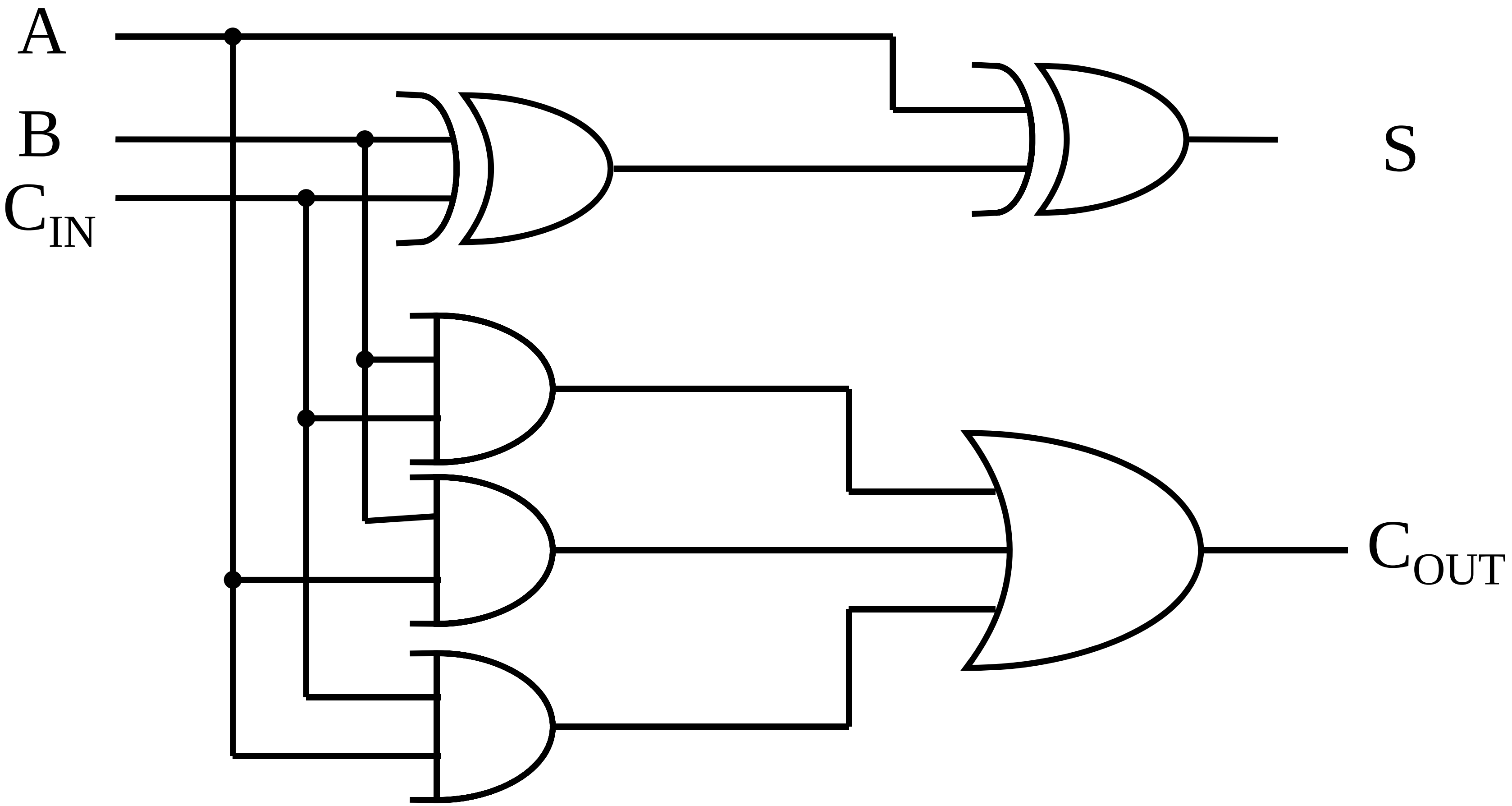
$$= C_{in} \oplus A \oplus B$$

Full-Adder

- Boolean expression for C_{OUT} and Sum, S.

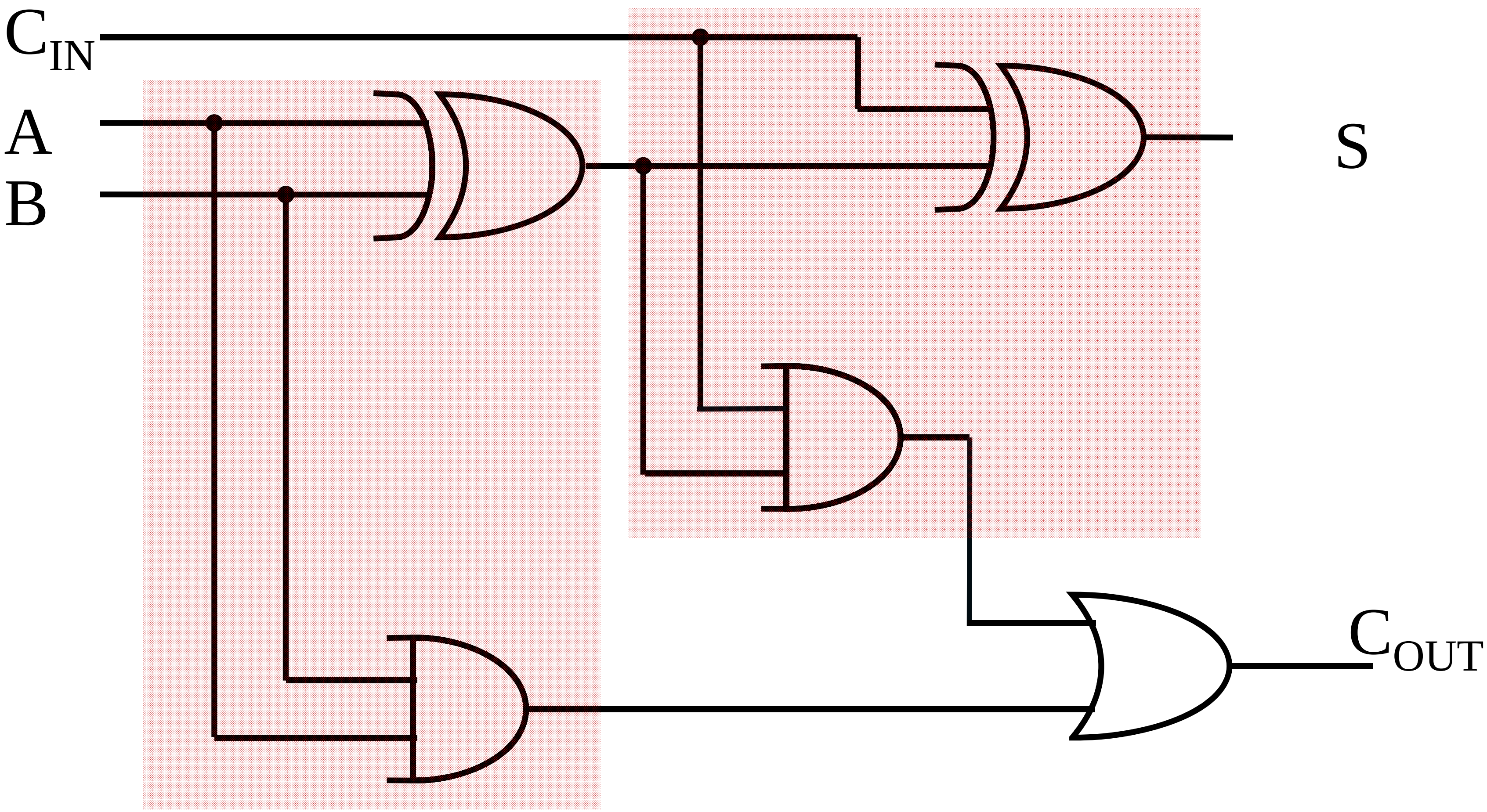
$$C_{OUT} = AC_{IN} + BC_{IN} + AB$$

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



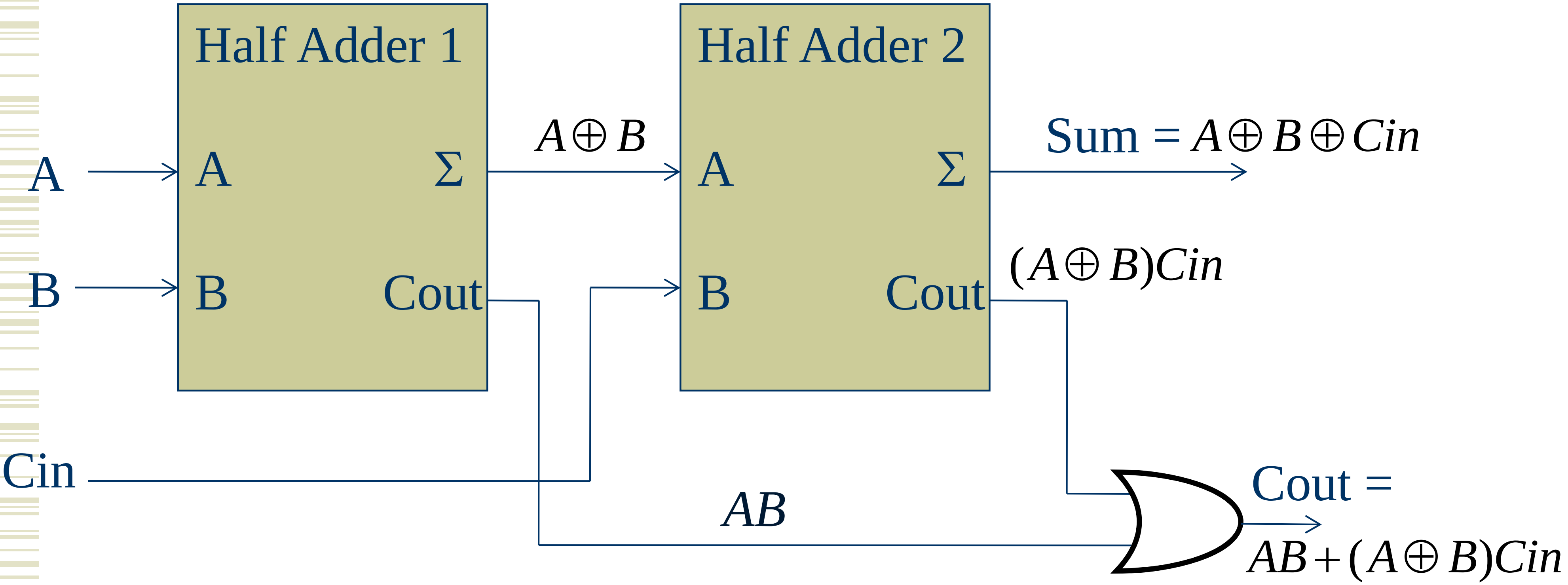
Full-Adder from Half Adders

- Logic circuit of Full Adder from two half adders:



Full-Adder

- Block diagram



Ripple Carry Adder/Parallel Adder

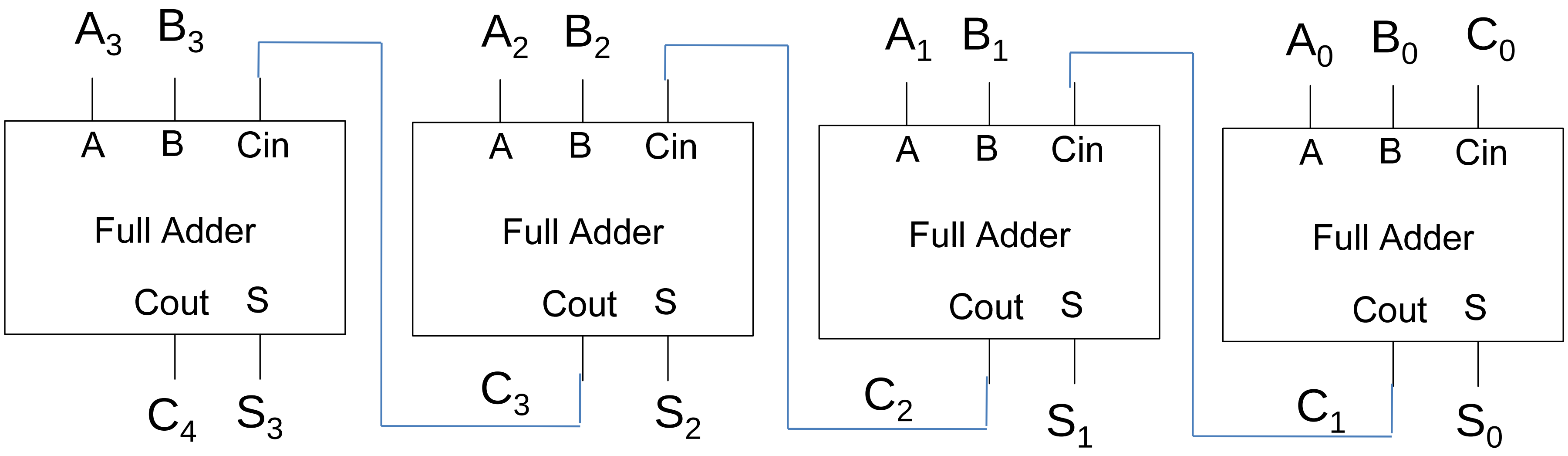
- Ripple carry adder is used to add multiple bit binary numbers, for example:

$$\begin{array}{rrrrrrl} & 0 & 0 & 0 & 1 & 0 & \leftarrow C_{4,}C_{3,}C_{2,}C_{1,}C_0 \\ & & 1 & 0 & 0 & 1 & \leftarrow A_3,A_2,A_1,A_0 \\ + & & 0 & 1 & 0 & 1 & \leftarrow B_3,B_2,B_1,B_0 \\ \hline & & 1 & 1 & 1 & 0 & \leftarrow S_3,S_2,S_1,S_0 \end{array}$$

To implement the above, we use the 4-bit ripple carry adder, which adds two four bit numbers $A_3A_2A_1A_0$ and $B_3B_2B_1B_0$ and generates the sum $S_3S_2S_1S_0$ with carries $C_4C_3C_2C_1C_0$

Ripple Carry Adder (cont.)

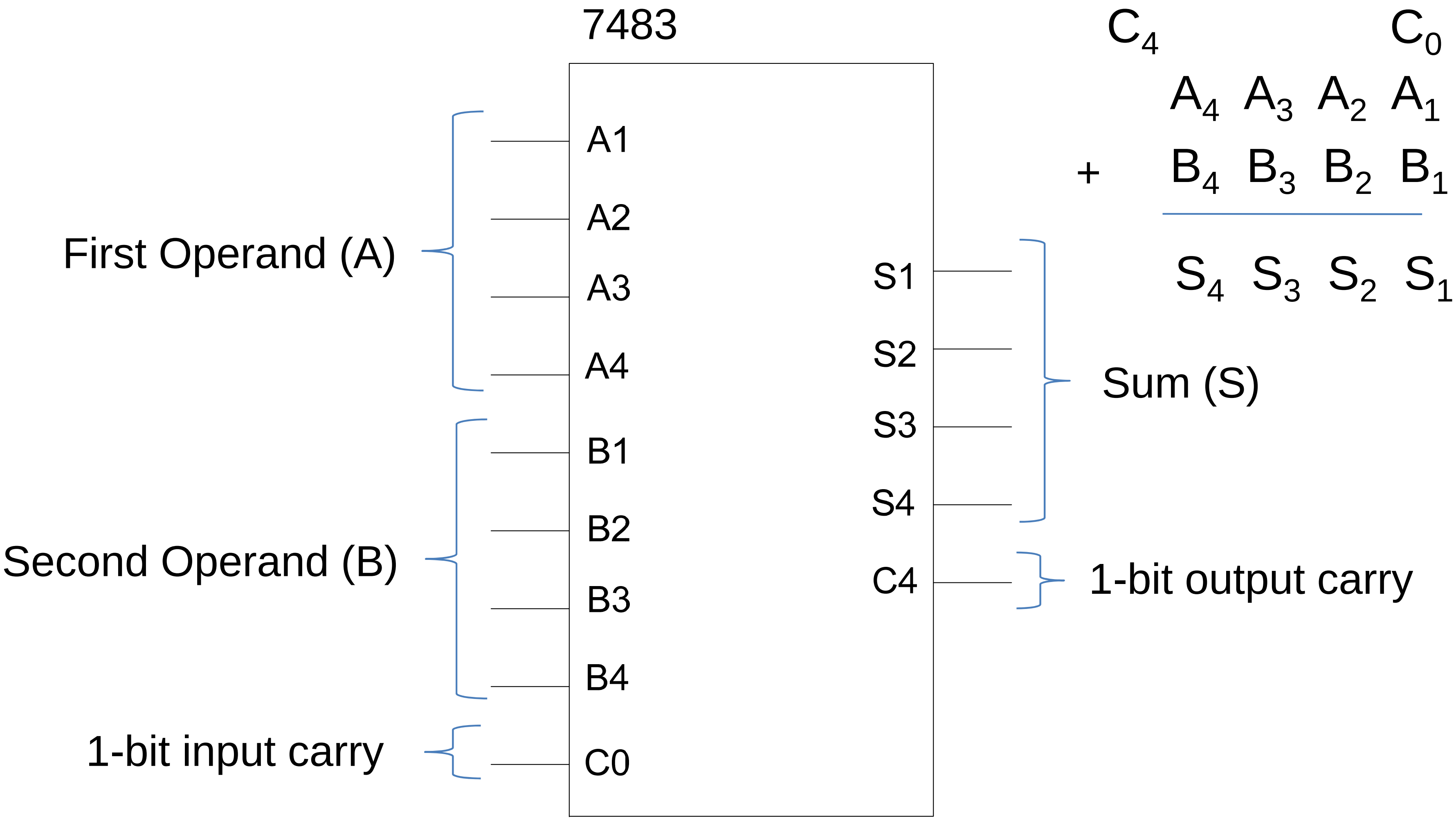
- How to design 4-bit ripple carry adder using four full adders?
  - Inputs $A_3A_2A_1A_0$, $B_3B_2B_1B_0$, and C_0 (first carry)
  - Outputs $S_3S_2S_1S_0$ and C_4 (final carry)



Dr Michael Tan, michael@fke.utm.my

Ripple Carry Adder (cont.)

- 7483 Single 4-bit Fast Adder

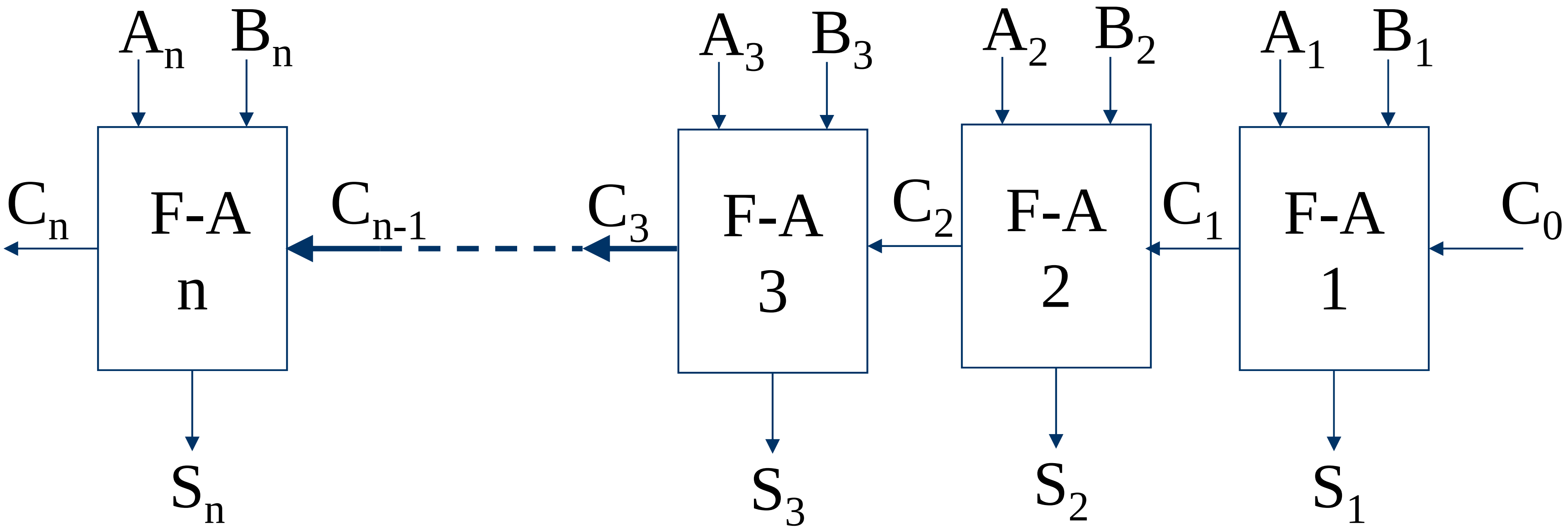


$$\begin{array}{rcccc} C_4 & & & & C_0 \\ & A_4 & A_3 & A_2 & A_1 \\ + & B_4 & B_3 & B_2 & B_1 \\ \hline & S_4 & S_3 & S_2 & S_1 \end{array}$$

Dr Michael Tan, michael@fke.utm.my

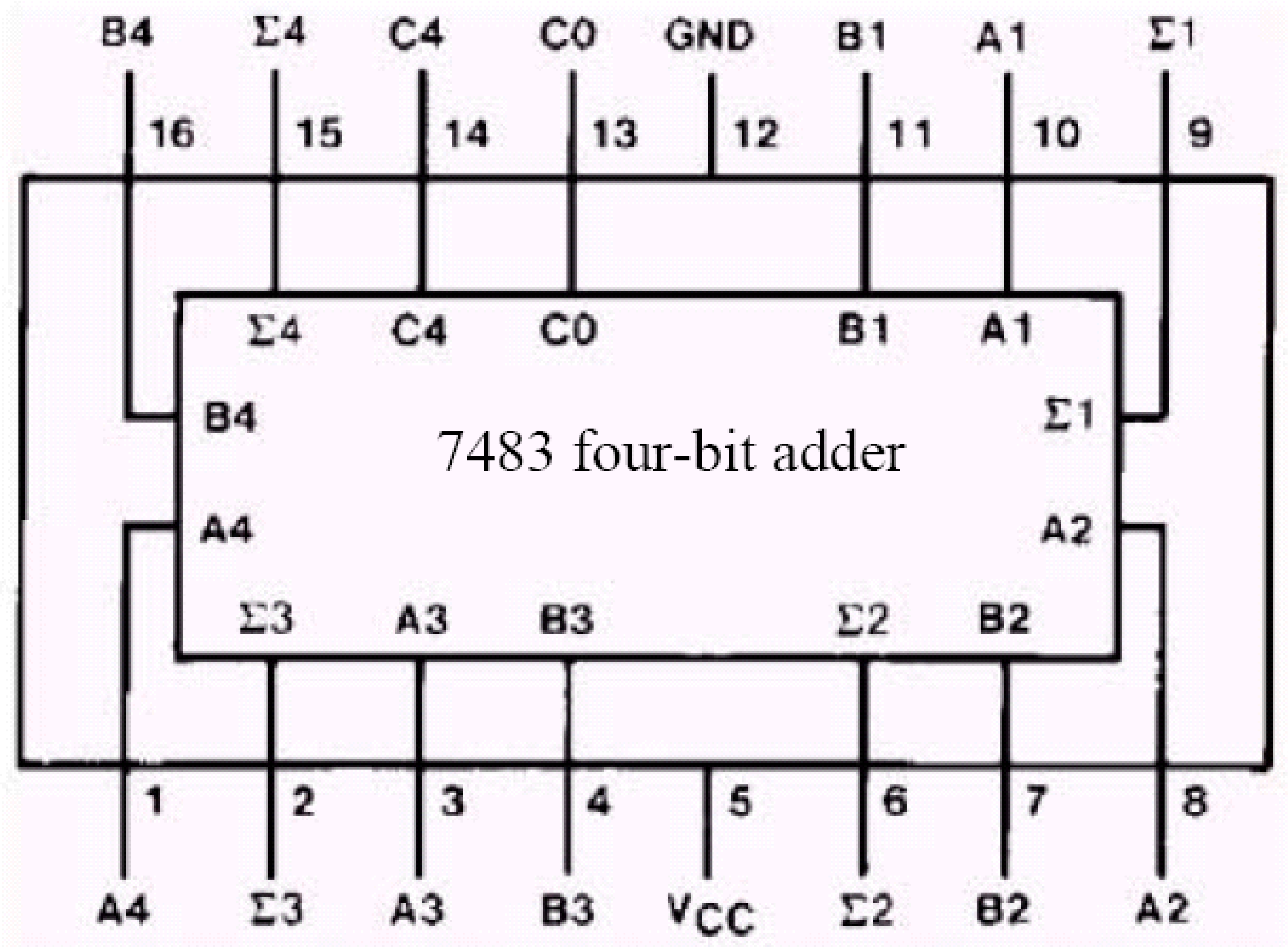
Ripple Full-Adder

- Adding two numbers for bit greater than 1 (N-bit number)

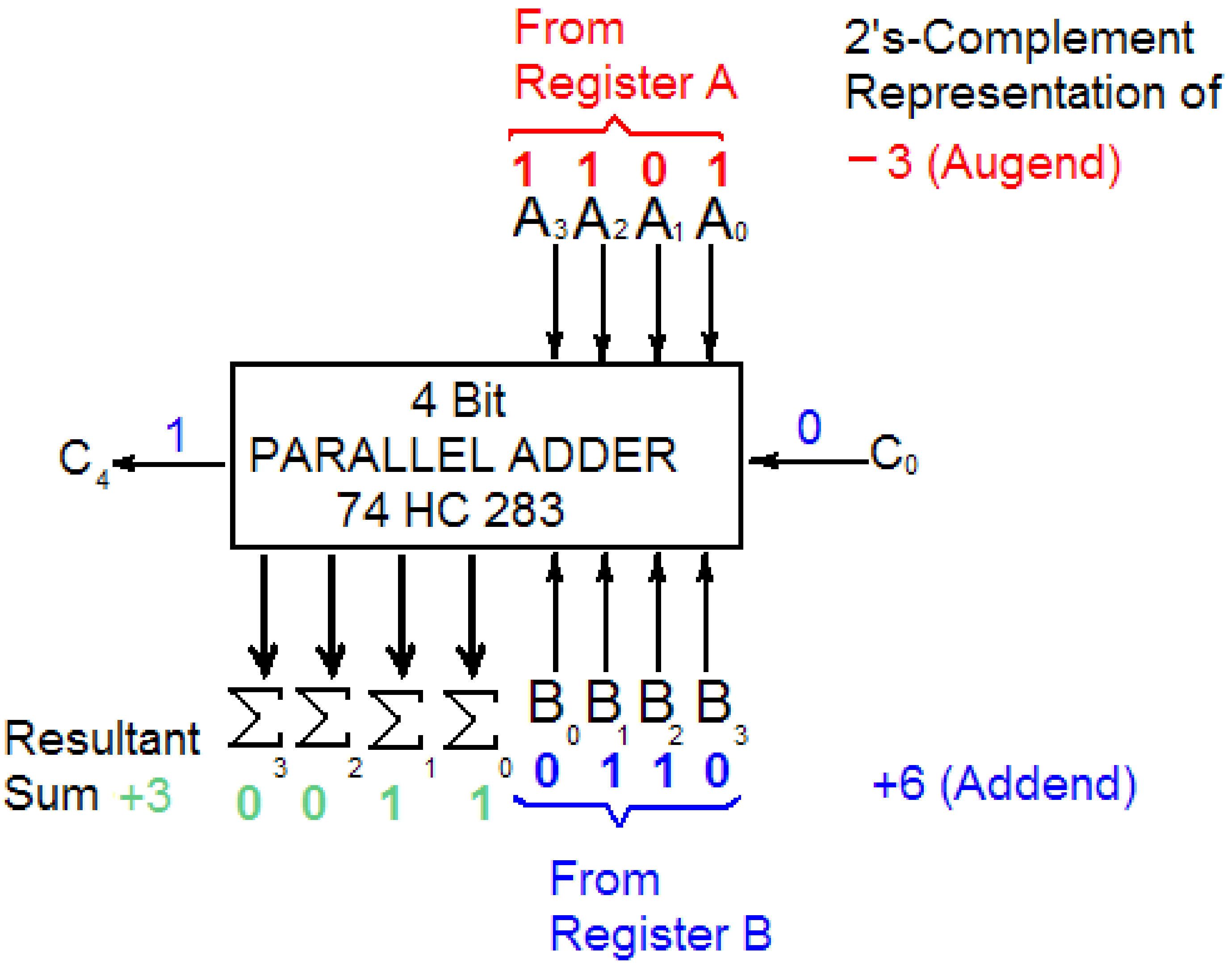


Full-Adder

- 7483 4-bit Full-Adder



Parallel Adders



Comparator

- Comparator is a device that compares two digital numbers or more. Comparison is made in terms of :
  - **Equal to ‘=‘**
  - **Less than ‘<‘**
  - **Greater than ‘>’**
  - **Less or equal than ‘≤’**
  - **Greater or equal than ‘≥’**

Comparator

- Basically, comparison is made between two 1-bit numbers! It can then be extended to any number with greater bit.
- Comparison between two 1-bit number are as follows:

(hint: obtain the expression using truth table and K-map)

A>B

The Boolean expression is $Y = A \bullet \overline{B}$

A<B

The Boolean expression is $Y = \overline{A} \bullet B$

Comparator

A=B

The Boolean expression is $Y = \overline{A \oplus B}$

A ≤ B

The Boolean expression is $Y = \overline{A \bullet \overline{B}}$

A ≥ B

The Boolean expression is $Y = \overline{\overline{A} \bullet B}$

Comparator

- Comparison between two 2-bit numbers, (**A_1A_0** and **B_1B_0**)
- For example, design a circuit that compares two 2-bit numbers such that the output **$X= A_1A_0>B_1B_0$**
  - Thus, the Boolean expression are:-

1) **$A_1>B_1$** → $A_1 \bullet \overline{B}_1$
2) **$A_1=B_1$** dan **$A_0>B_0$** → $\overline{(A_1 \oplus B_1)} \cdot (A_0\overline{B}_0)$

$$\mathbf{X} = A_1 \bullet \overline{B}_1 + \overline{(A_1 \oplus B_1)} \cdot (A_0\overline{B}_0)$$

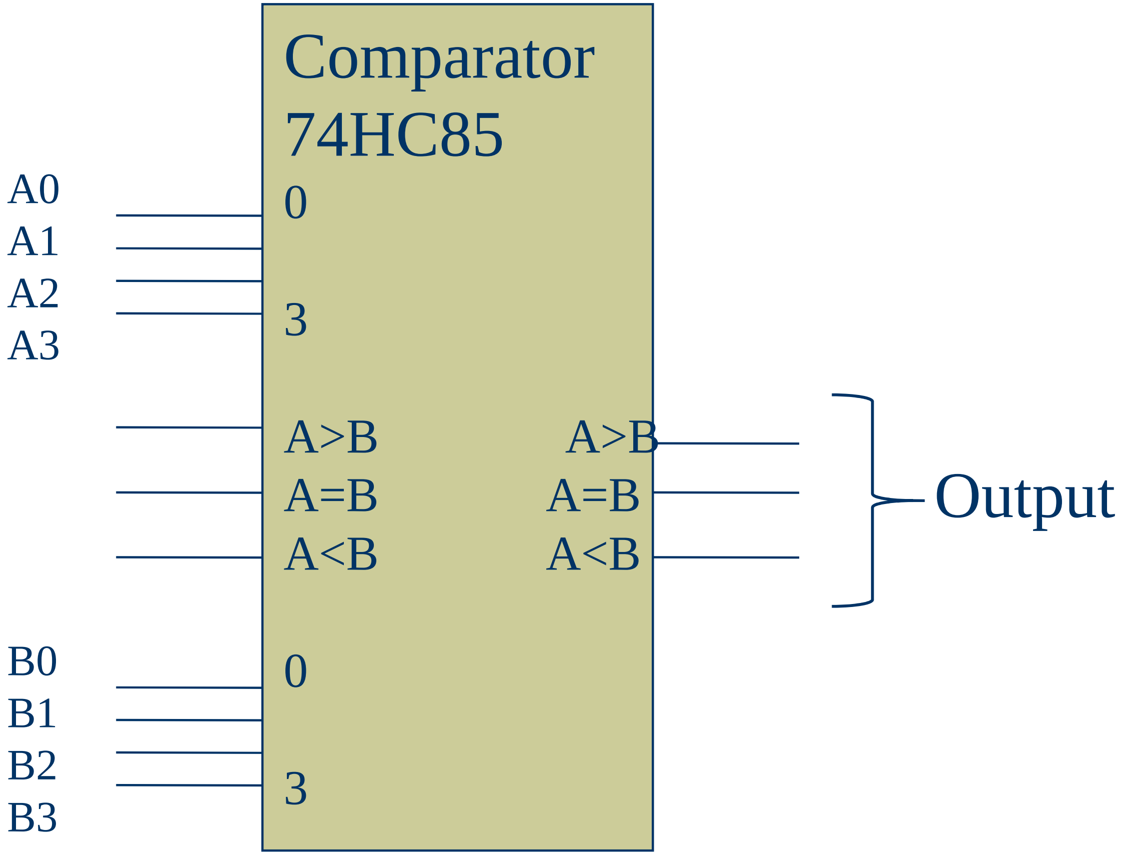
Comparator

- Design a circuit that compares two 3-bit numbers **A** and **B** (**$A=A_2A_1A_0$** and **$B=B_2B_1B_0$**) such that the output are as follows:-

  - **X = 1**, if **$A=B$ ($A_2A_1A_0 = B_2B_1B_0$)**
  - **Y = 1**, if **$A>B$ ($A_2A_1A_0 > B_2B_1B_0$)**
  - **Z = 1, if $A<B$ ($A_2A_1A_0 < B_2B_1B_0$)**

Solution Comparator

- Use 74HC85 Comparator chip



Comparators

- Function is to compare the magnitudes of two binary quantities
- 1-bit comparator:

F_1 = 1 when A = B
F_2 = 1 when A > B
F_3 = 1 when A < B

A = B → F_1
A > B → F_2
A < B → F_3

A B	F_1	F_2	F_3
0 0	1	0	0
0 1	0	0	1
1 0	0	1	0
1 1	1	0	0

What are the logic expressions?

$$F_1 = \overline{A \oplus B}$$

$$F_2 = A \cdot \overline{B}$$

$$F_3 = \overline{A} \cdot B$$

Dr Michael Tan, michael@fke.utm.my

Comparators (cont.)

- 2-bit comparator A (A1,A0) and B (B1, B0)

A_1 A_0 B_1 B_0	A = B (F_1)	A > B (F_2)	A < B (F_3)
0 0 0 0	1	0	0
0 0 0 1	0	0	1
0 0 1 0	0	0	1
0 0 1 1	0	0	1
0 1 0 0	0	1	0
0 1 0 1	1	0	0
0 1 1 0	0	0	1
0 1 1 1	0	0	1

A_1 A_0 B_1 B_0	A = B (F_1)	A > B (F_2)	A < B (F_3)
1 0 0 0	0	1	0
1 0 0 1	0	1	0
1 0 1 0	1	0	0
1 0 1 1	0	0	1
1 1 0 0	0	1	0
1 1 0 1	0	1	0
1 1 1 0	0	1	0
1 1 1 1	1	0	0

$F_1 = 1$ when $A = B$
$F_2 = 1$ when $A > B$
$F_3 = 1$ when $A < B$

Can you get the logic expression?

Dr Michael Tan, michael@fke.utm.my

Comparators (cont.)

- 2-bit comparator (cont.)

By doing K-Maps for F1, F2, and F3,

For A = B,

$$F_1 = \overline{A_1} \cdot \overline{A_0} \cdot \overline{B_1} \cdot \overline{B_0} + \overline{A_1} \cdot A_0 \cdot \overline{B_1} \cdot B_0 + A_1 \cdot \overline{A_0} \cdot B_1 \cdot \overline{B_0} + A_1 \cdot A_0 \cdot B_1 \cdot B_0$$

For A > B,

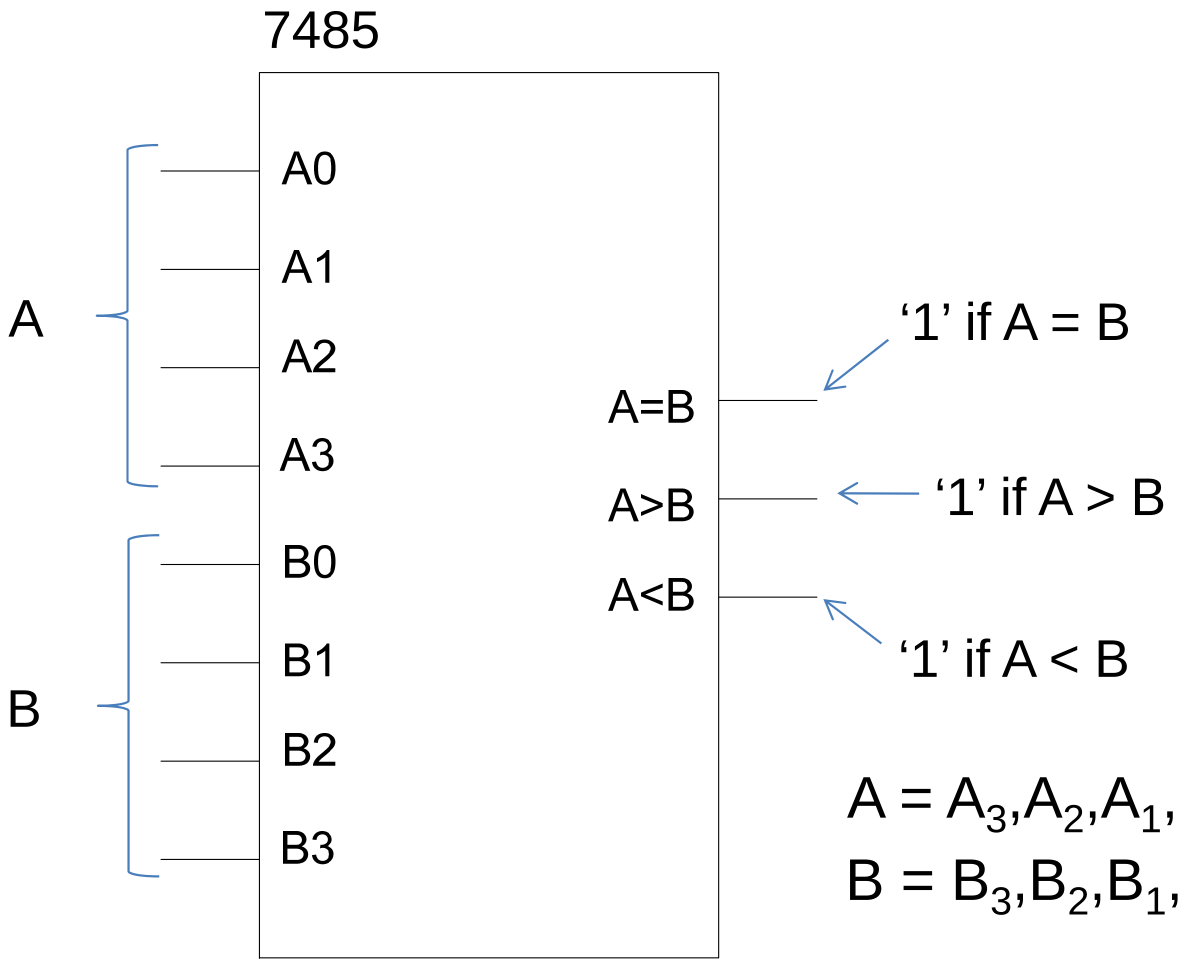
$$F_2 = A_1 \cdot \overline{B_1} + A_0 \cdot \overline{B_1} \cdot \overline{B_0} + A_1 \cdot A_0 \cdot \overline{B_0}$$

For A < B,

$$F_3 = \overline{A_1} \cdot B_1 + \overline{A_0} \cdot B_1 \cdot B_0 + \overline{A_1} \cdot \overline{A_0} \cdot B_0$$

Comparators (cont.)

- 7485 Single 4-bit comparator



How to design the circuit?
- The circuit is quite complicated due to the eight inputs, therefore will not be discussed in this class

$A = A_3,A_2,A_1,A_0$

$B = B_3,B_2,B_1,B_0$