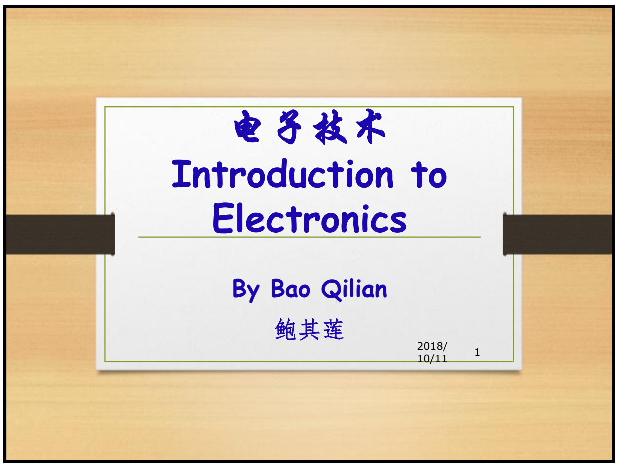
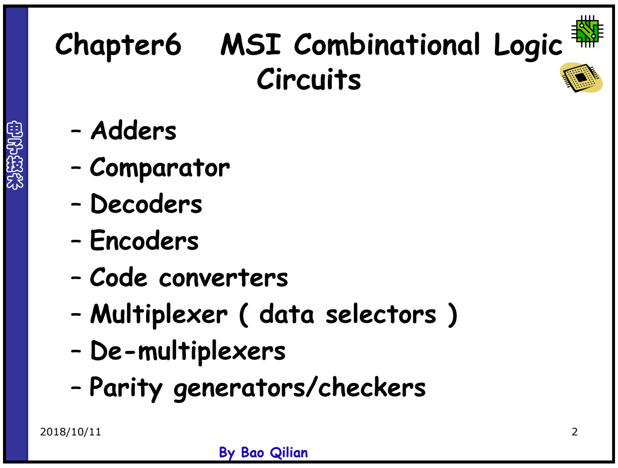


电子技术
Introduction to Electronics

By Bao Qilian

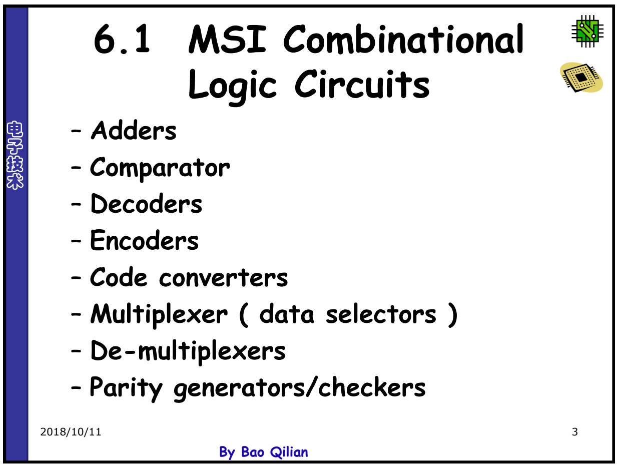
鲍其莲



电子技术

Chapter6 MSI Combinational Logic Circuits

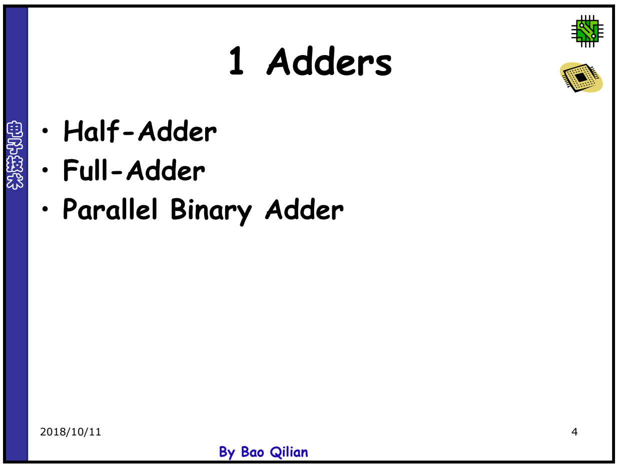
- Adders
- Comparator
- Decoders
- Encoders
- Code converters
- Multiplexer (data selectors)
- De-multiplexers
- Parity generators/checkers



电子技术

6.1 MSI Combinational Logic Circuits

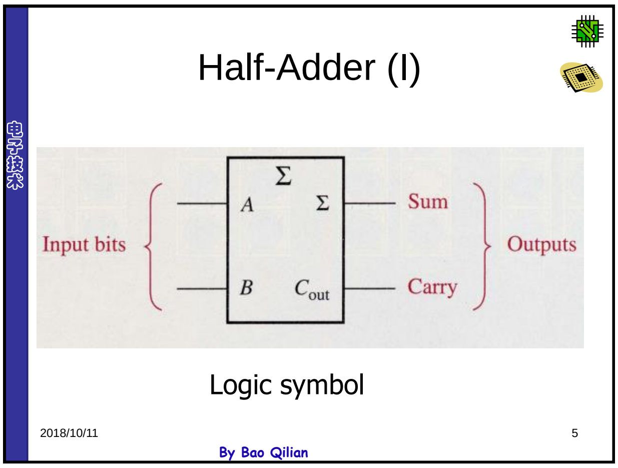
- Adders
- Comparator
- Decoders
- Encoders
- Code converters
- Multiplexer (data selectors)
- De-multiplexers
- Parity generators/checkers



电子技术

1 Adders

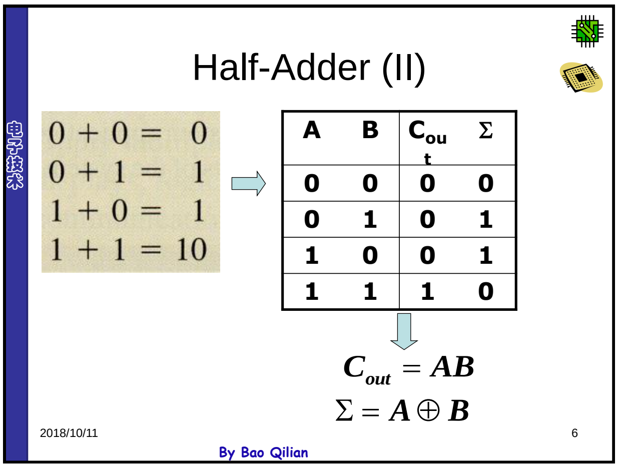
- Half-Adder
- Full-Adder
- Parallel Binary Adder



电子技术

Half-Adder (I)

Logic symbol



电子技术

Half-Adder (II)

$0 + 0 = 0$
$0 + 1 = 1$
$1 + 0 = 1$
$1 + 1 = 10$

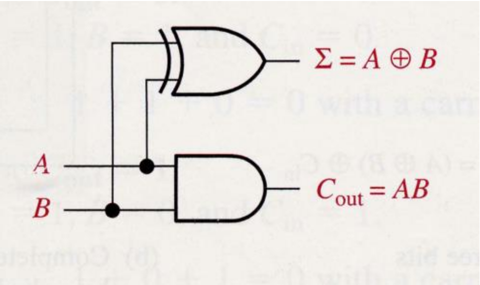
A	B	C_{out}	Σ
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

$$C_{out} = AB$$

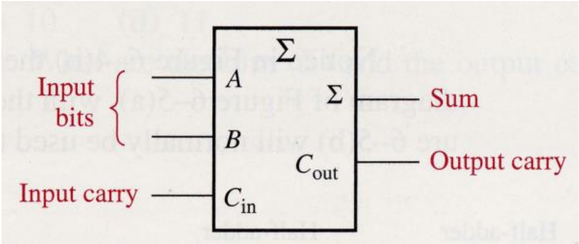
$$\Sigma = A \oplus B$$

Half-Adder (III)

Logic diagram



Full-Adder (I)



Full-Adder(II)

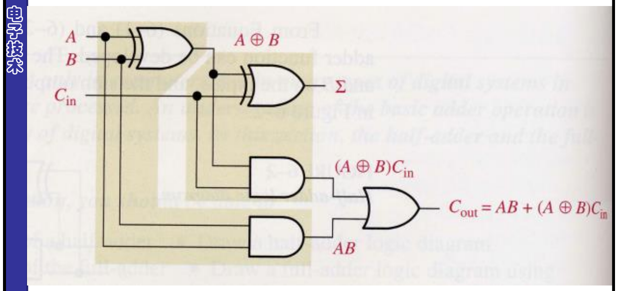
$$C_{out} = AB + (A \oplus B)C_{in}$$

$$\Sigma = (A \oplus B) \oplus C_{in}$$

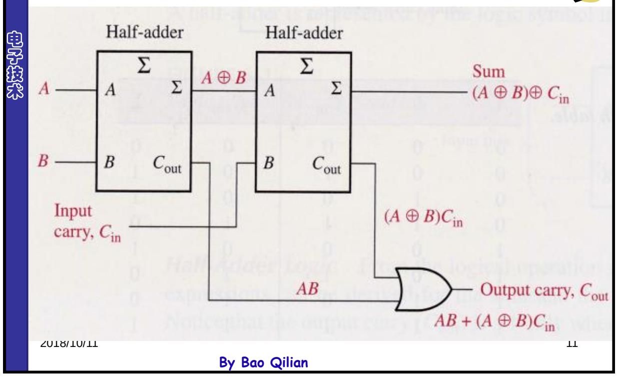
$$\begin{aligned}
\Sigma &= \overline{A}\overline{B}C_{in} + \overline{A}B\overline{C}_{in} + A\overline{B}\overline{C}_{in} + ABC_{in} \\
&= (\overline{A}B + A\overline{B})\overline{C}_{in} + (\overline{A}\overline{B} + AB)C_{in} \\
&= (A \oplus B)\overline{C}_{in} + \overline{A \oplus B}C_{in} \\
&= A \oplus B \oplus C_{in} \\
C_{out} &= \overline{A}BC_{in} + A\overline{B}C_{in} + AB\overline{C}_{in} + ABC_{in} \\
&= (\overline{A}B + A\overline{B})C_{in} + AB \\
&= (A \oplus B)C_{in} + AB
\end{aligned}$$

A	B	C_{in}	C_{out}	Σ
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

Logic Diagram



Full-adder implemented with two half-adders.



Parallel Binary Adders

- Two or more full-adders are connected to form parallel binary adders.
- A single full-adders is capable of adding two 1-bit binary numbers and an input carry. To add binary numbers with more than one bit, additional full-adders must be used.

Addition of Two-bit Binary Number

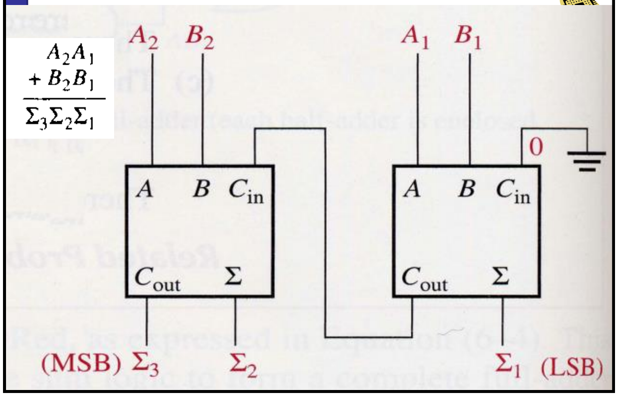
Carry bit from right column

10
11
+01

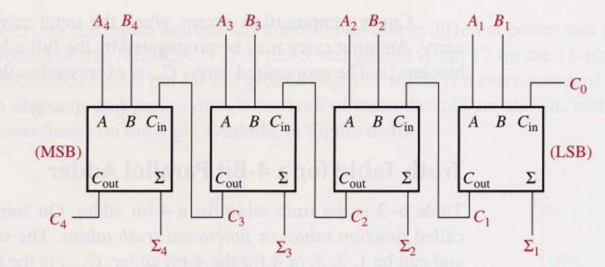
100

Carry bit to the next column

2-bit parallel adder using two full-adders

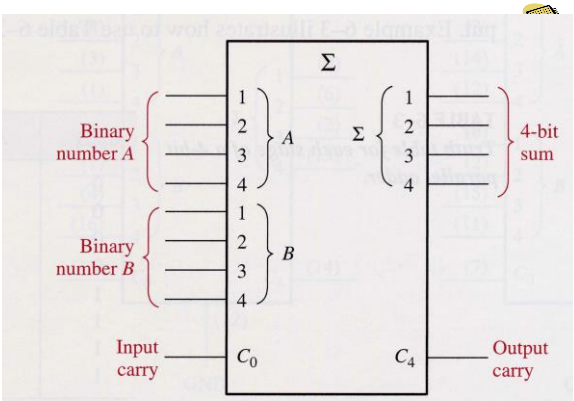


4-bit ripple adder using four full-adders



Block diagram

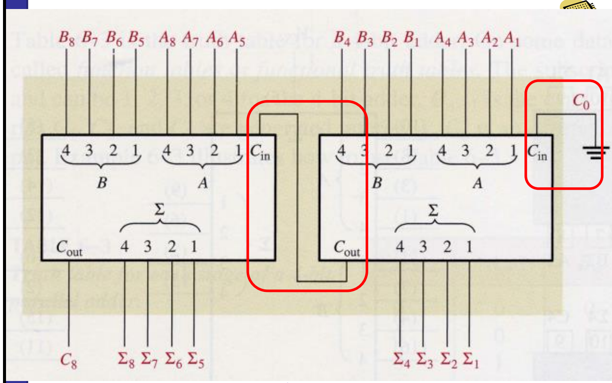
Logic symbol



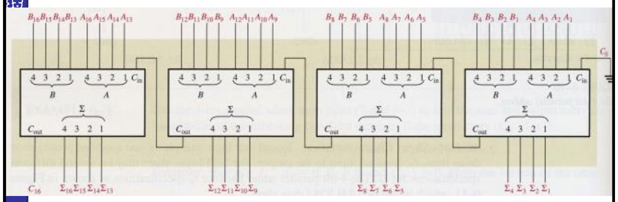
Adder Expansion

- 8-bit adder
- 16-bit adder

8-bit adder


B_8 B_7 B_6 B_5 A_8 A_7 A_6 A_5
B_4 B_3 B_2 B_1 A_4 A_3 A_2 A_1
C_0
4 3 2 1 4 3 2 1 C_{in}
B A
Σ
C_{out} 4 3 2 1
C_8
Σ_8 Σ_7 Σ_6 Σ_5
Σ_4 Σ_3 Σ_2 Σ_1

16-bit adder


$B_{16} B_{15} B_{14} B_{13}$ $A_{16} A_{15} A_{14} A_{13}$
$B_{12} B_{11} B_{10} B_9$ $A_{12} A_{11} A_{10} A_9$
B_8 B_7 B_6 B_5 A_8 A_7 A_6 A_5
B_4 B_3 B_2 B_1 A_4 A_3 A_2 A_1
C_0
4 3 2 1 4 3 2 1 C_{in}
B
A
Σ
C_{out} 4 3 2 1
C_{16}
$\Sigma_{16} \Sigma_{15} \Sigma_{14} \Sigma_{13}$
$\Sigma_{12} \Sigma_{11} \Sigma_{10} \Sigma_9$
Σ_8 Σ_7 Σ_6 Σ_5
Σ_4 Σ_3 Σ_2 Σ_1

Carry look-ahead addition

- The *carry look-ahead adder* anticipates the output carry based on the input bits, and produces the output carry by either *carry generation* or *carry propagation.*

Two cases:
1.AB=1 , Cin=0,1
2.A⊕B=1 , Cin=1

C_{out}=1

$$(\boldsymbol{CO})_i = \boldsymbol{A_i B_i} + (\boldsymbol{A_i} \oplus \boldsymbol{B_i})(\boldsymbol{CI})_i$$

$$\textbf{\textit{Let}} \quad \boldsymbol{A_i B_i} = \boldsymbol{G_i} \quad (\textbf{\textit{carry generation}})$$

$$\boldsymbol{A_i} \oplus \boldsymbol{B_i} = \boldsymbol{P_i} \quad (\textbf{\textit{carry propagation}})$$

$$(\boldsymbol{CO})_i = \boldsymbol{G_i} + \boldsymbol{P_i}(\boldsymbol{CI})_i$$

$$= \boldsymbol{G_i} + \boldsymbol{P_i}(\boldsymbol{G_{i-1}} + \boldsymbol{P_{i-1}}(\boldsymbol{CI})_{i-1})$$

$$\boldsymbol{S_i} = \boldsymbol{A_i \overline{B_i}}(\overline{\boldsymbol{CI}})_i + \boldsymbol{\overline{A_i} B_i}(\overline{\boldsymbol{CI}})_i + \boldsymbol{\overline{A_i}\,\overline{B_i}}(\boldsymbol{CI})_i + \boldsymbol{A_i B_i}(\boldsymbol{CI})_i$$

$$= \boldsymbol{A_i} \oplus \boldsymbol{B_i} \oplus (\boldsymbol{CI})_i$$

For nth bit:

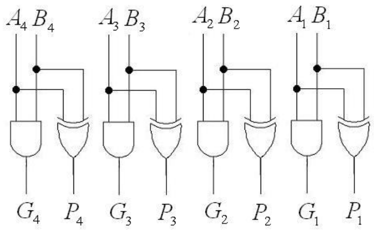
$$\Sigma_n = f(A_n, B_n, C_{n-1}) = A_n \oplus B_n \oplus C_{n-1} = P_n \oplus C_{n-1}$$
$$C_n = g(A_n, B_n, C_{n-1}) = A_n B_n + (A_n \oplus B_n) C_{n-1} = G_n + P_n C_{n-1}$$

Four-Bit Carry Look-ahead Adders

$$\begin{cases} C_1 = G_1 + P_1 C_0 = G_1 \\ C_2 = G_2 + P_2 C_1 = G_2 + P_2 G_1 \\ C_3 = G_3 + P_3 C_2 = G_3 + P_3 G_2 + P_3 P_2 G_1 \\ C_4 = G_4 + P_4 C_3 = G_4 + P_4 G_3 + P_4 P_3 G_2 + P_4 P_3 P_2 G_1 \end{cases}$$

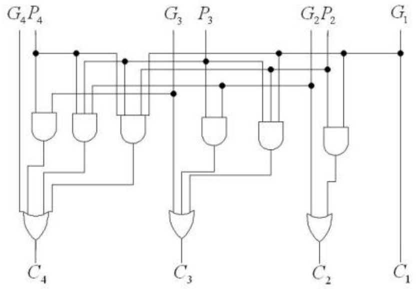
$$\begin{cases} \Sigma_1 = P_1 \oplus C_0 \\ \Sigma_2 = P_2 \oplus C_1 \\ \Sigma_3 = P_3 \oplus C_2 \\ \Sigma_4 = P_4 \oplus C_3 \end{cases}$$

Adder module producing G and P



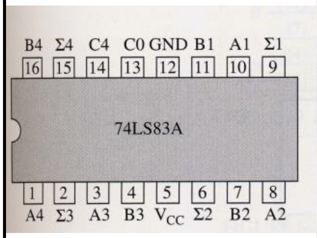
电子技术

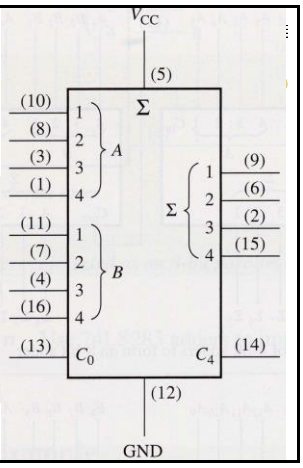
Carry Look-ahead (CLA) Circuit



MSI Adders(I)

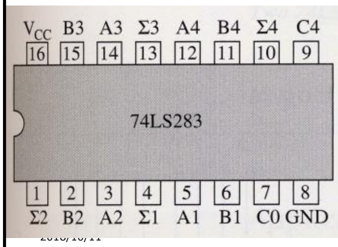
74LS83A

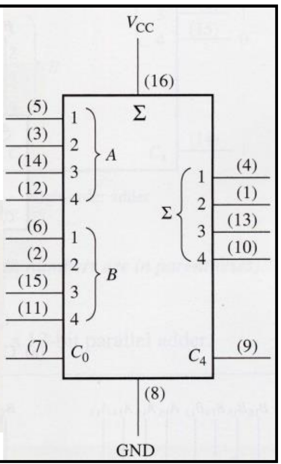




MSI Adders(II)

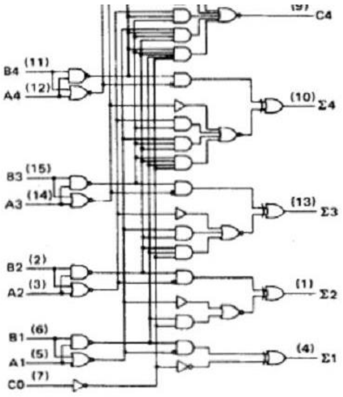
74LS283



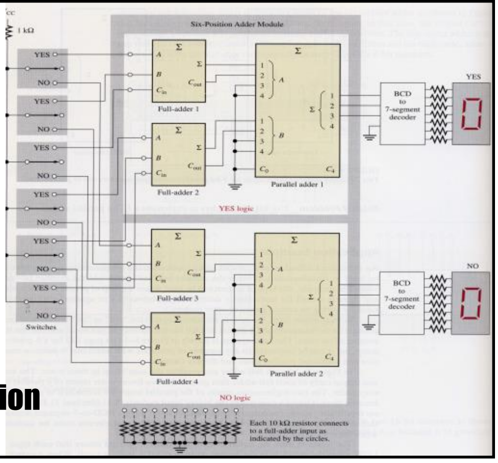


电子技术

logic diagram of 74LS283



A voting system

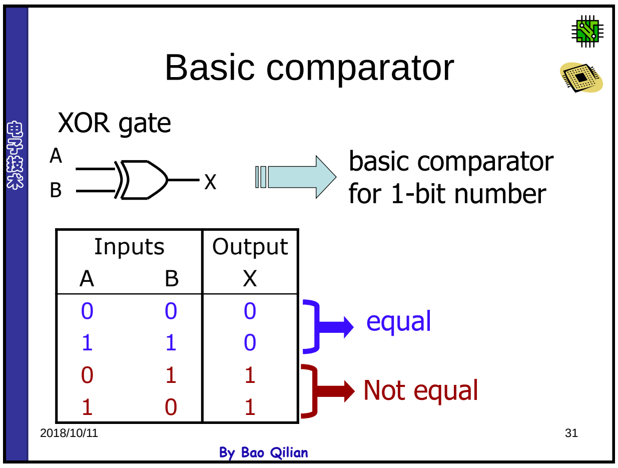


Application Example

电子技术

2 Comparator

- Basic comparator—Exclusive-OR
- Magnitude comparator
- Expansion of comparators



Basic comparator

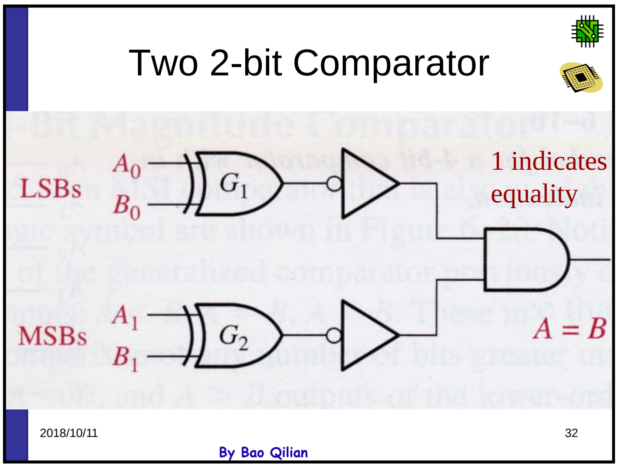
XOR gate

basic comparator for 1-bit number

Inputs		Output
A	B	X
0	0	0
1	1	0
0	1	1
1	0	1

equal

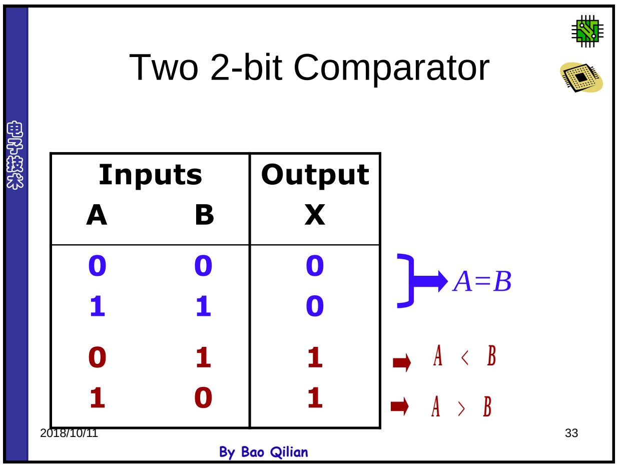
Not equal



Two 2-bit Comparator

1 indicates equality

$A = B$



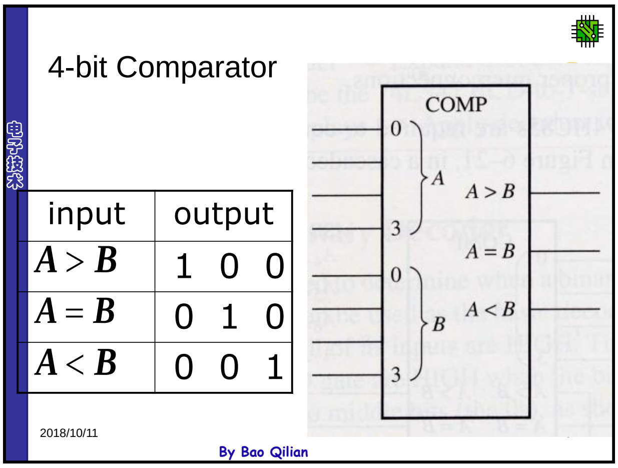
Two 2-bit Comparator

Inputs		Output
A	B	X
0	0	0
1	1	0
0	1	1
1	0	1

$A=B$

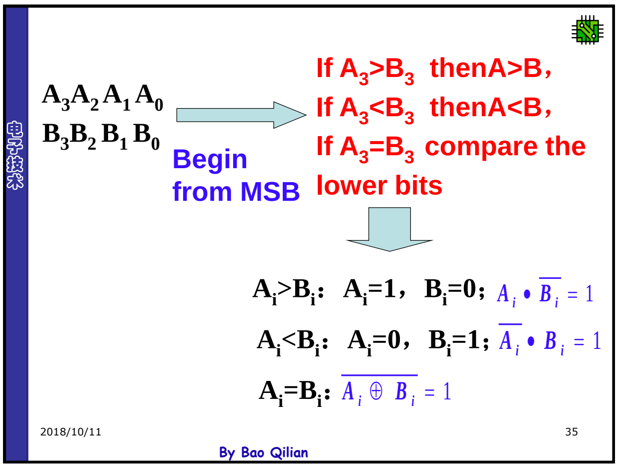
➡ $A < B$

➡ $A > B$



4-bit Comparator

input	output
$A > B$	1 0 0
$A = B$	0 1 0
$A < B$	0 0 1



$A_3A_2A_1A_0$

$B_3B_2B_1B_0$

Begin from MSB

If A_3>B_3 thenA>B,

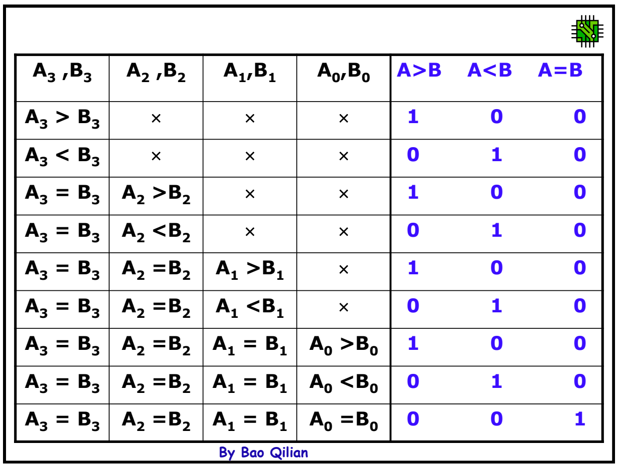
If A_3<B_3 thenA<B,

If A_3=B_3 compare the lower bits

A_i>B_i: A_i=1, B_i=0; $A_i \bullet \overline{B_i} = 1$

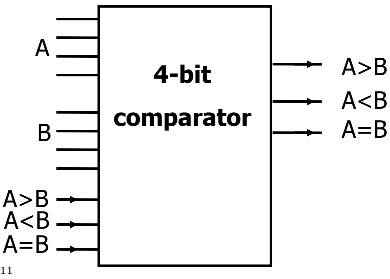
A_i<B_i: A_i=0, B_i=1; $\overline{A_i} \bullet B_i = 1$

A_i=B_i: $\overline{A_i \oplus B_i} = 1$

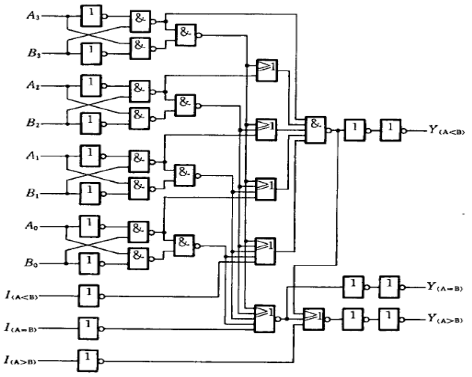


A_3 ,B_3	A_2 ,B_2	A_1,B_1	A_0,B_0	A>B	A<B	A=B
A_3 > B_3	×	×	×	1	0	0
A_3 < B_3	×	×	×	0	1	0
A_3 = B_3	A_2 >B_2	×	×	1	0	0
A_3 = B_3	A_2 <B_2	×	×	0	1	0
A_3 = B_3	A_2 =B_2	A_1 >B_1	×	1	0	0
A_3 = B_3	A_2 =B_2	A_1 <B_1	×	0	1	0
A_3 = B_3	A_2 =B_2	A_1 = B_1	A_0 >B_0	1	0	0
A_3 = B_3	A_2 =B_2	A_1 = B_1	A_0 <B_0	0	1	0
A_3 = B_3	A_2 =B_2	A_1 = B_1	A_0 =B_0	0	0	1

4-bit Magnitude Comparator



logic diagram of 4-bit comparator CC14585



Logic functions of outputs

$$Y_{(A<B)} = \bar{A}_3 B_3 + (A_3 \odot B_3)\bar{A}_2 B_2 + (A_3 \odot B_3)(A_2 \odot B_2)\bar{A}_1 B_1$$

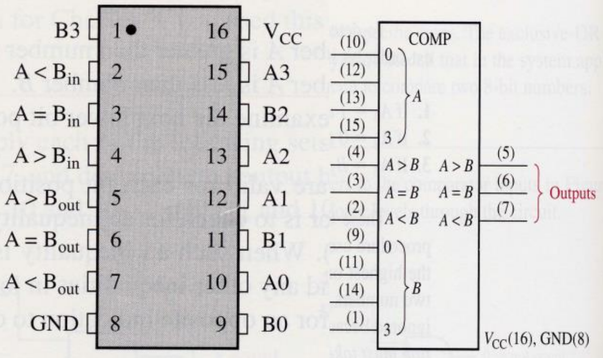
$$+ (A_3 \odot B_3)(A_2 \odot B_2)(A_1 \odot B_1)\bar{A}_0 B_0$$

$$+ (A_3 \odot B_3)(A_2 \odot B_2)(A_1 \odot B_1)(A_0 \odot B_0) I_{(A<B)}$$

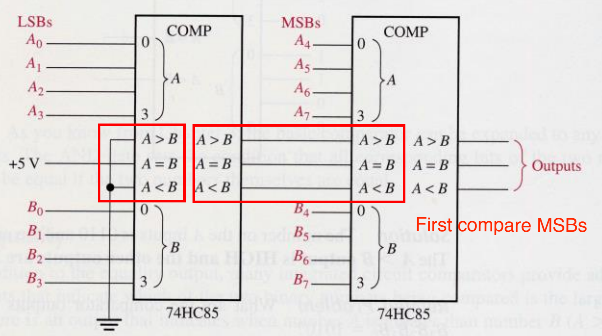
$$Y_{(A=B)} = (A_3 \odot B_3)(A_2 \odot B_2)(A_1 \odot B_1)(A_0 \odot B_0) I_{(A=B)}$$

$$Y_{(A>B)} = \overline{Y_{(A<B)} + Y_{(A=B)}}$$

MSI 4-bit Magnitude Comparator- 74HC85

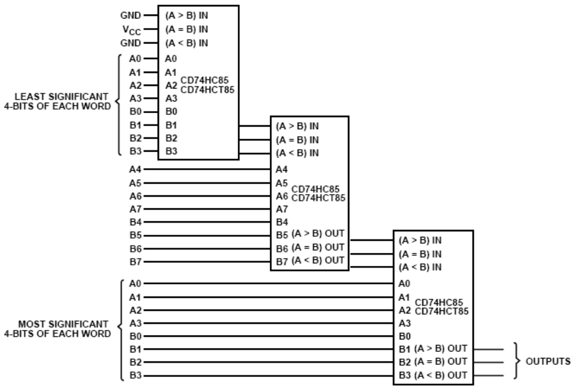


Expansion: 8-bit Comparator



First compare MSBs

Series Cascading – Comparing 12-Bit Words



Problems chapter 5:

2, 6(d), 12(c),

24(e), 26(g), 29.

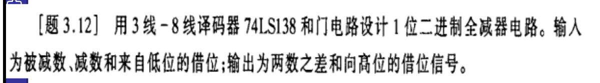
Chapter 6 Functions of Combinational Logic

Problems :

2，8，14，16(b), 21，

22，

补充题



［题 3.12］ 用 3 线－8 线译码器 74LS138 和门电路设计 1 位二进制全减器电路。输入为被减数、减数和来自低位的借位；输出为两数之差和向高位的借位信号。