ENGG2020 DIGITAL LOGIC AND SYSTEMS

CHAPTER 4A: COMBINATIONAL LOGIC

By Dr. Anthony Sum
Department of Computer Science and Engineering
The Chinese University of Hong Kong

1

CONTENTS

- Combinational logic circuit
- Adder and Subtractor
- Comparator
- Decoder and Encoder
- Multiplexer and Demultiplexer
- Tri-state Buffer

2

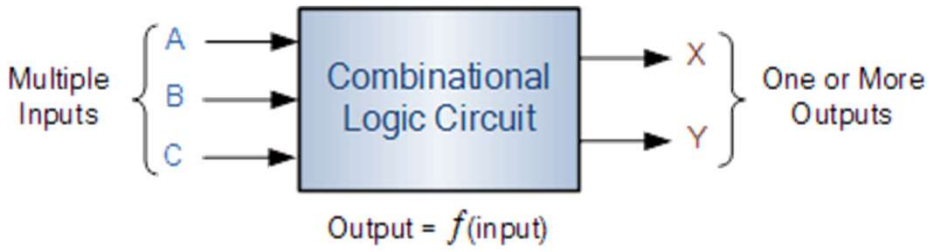
DIGITAL CIRCUIT CATEGORIES

- Combinational logic circuits
  - The outputs at any instant of time depend upon the inputs present at that time instant
  - No memory in these circuits
- Sequential logic circuits
  - The outputs at any instant of time depend upon the present inputs as well as the past inputs/outputs
  - There are memory elements used to store past information
- In this chapter, let's focus on Combinational Logic Circuits

3

COMBINATIONAL CIRCUIT

- A combinational circuit consists of input variables, logic gates, and output variables
- Combinational logic gates react to the values of the signals at their inputs and produce the value of the output signal



4

EXAMPLE (CODE CONVERSION)

- Design a combinational logic circuit to convert Binary Code Decimal (BCD) to Excess-3 Code

excess-3 code start with 0011=3

Table 4.2
Truth Table for Code Conversion Example

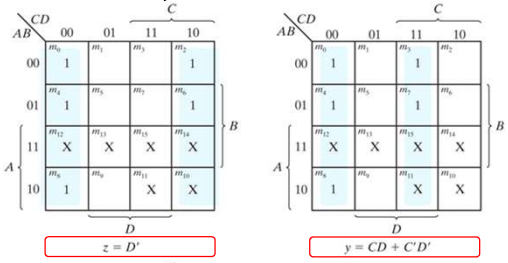
Input BCD				Output Excess-3 Code			
A	*B*	*C*	*D*	*w*	*x*	*y*	*z*
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0

5

EXAMPLE (CODE CONVERSION)

- Since we have 4 inputs and 4 outputs, 4x 4-variable K-maps are constructed for w, x, y, and z

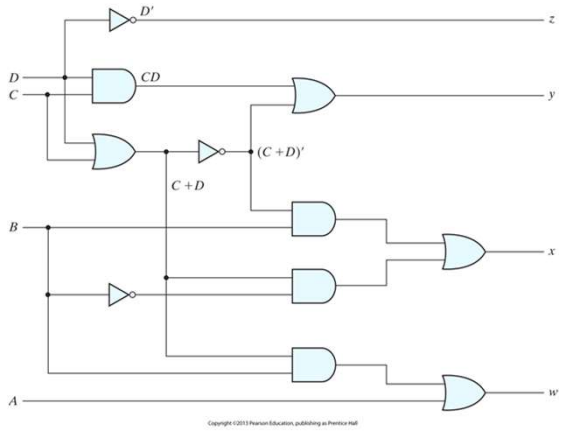
- Some of cells are X (i.e. "Don't Care") because there are only 10 digits (i.e. 0 ~ 9) in BCD



6

EXAMPLE (CODE CONVERSION)

- Use the simplified logic functions, the combinational logic circuit can be built as below:



7

8

BINARY ADDER AND SUBTRACTOR

HALF ADDER (HA)

- A half adder is a 1-bit adder which do not take carry-in into consideration
- The sum of two 1-bit binary numbers A and B, can be represented as S = A + B:
- 0 + 0 = 0
- 0 + 1 = 1
- 1 + 0 = 1
- 1 + 1 = 10, where 1 is known as carry-out bit, C_O

9

HALF ADDER (HA)

- By summarizing the equations, the truth table of HA is shown below:

A	B	C_o	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

For S, it is a XOR operation of A and B

For C_O, it is an AND operation of A and B

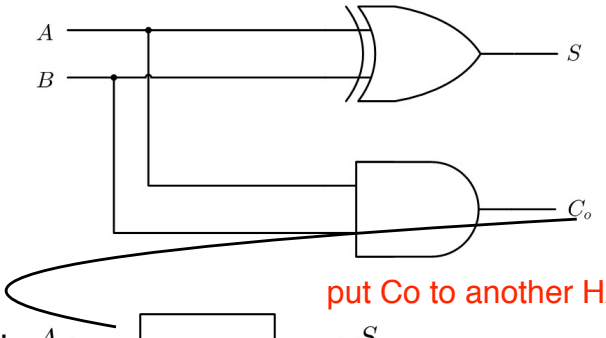
- The Boolean functions of HA: $S = \bar{A}B + A\bar{B} = A \oplus B,$

carry out bit $C_o = AB$

10

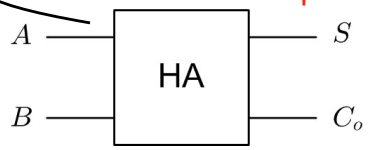
HALF ADDER (HA)

- The logic circuit of HA:



put Co to another HA to add it

- The block diagram of HA:



11

FULL ADDER (FA)

- When the operations require more than 1 bit, HA is not useful
  - Carry-out bits from the lower significant bits are not considered/used at the higher bit operations
- By extending the HA, one can implement a full adder (FA) which takes a carry-in bit as an extra input
  - The carry-out bit of the lower order bit can be connected to the carry-in bit of the immediate higher order bit
- FA becomes the basic unit for a multiple-bit adder

12

FULL ADDER (FA)

- Three inputs:
  - Two 1-bit numbers, A and B
  - One Carry-in, C_{in}
- Two outputs:
  - Sum, S
  - Carry-out, C_O
- The truth table of FA:
- Equations of FA:

$$S = \Sigma(1,2,4,7),$$
$$C_o = \Sigma(3,5,6,7)$$

A	B	C_{in}	C_o	S	
0	0	0	0	0	0
0	0	1	0	1	1
0	1	0	0	1	
0	1	1	1	0	
1	0	0	0	1	
1	0	1	1	0	2
1	1	0	1	0	
1	1	1	1	1	3

13

FULL ADDER (FA)

$$S = \Sigma(1,2,4,7),$$
$$C_o = \Sigma(3,5,6,7)$$

- Simplify the Boolean functions of FA:

	C_{in}' (0)	C_{in} (1)
A'B' (00)		1
A'B (01)	1	
AB (11)		1
AB' (10)	1	

$$S = A'B'C_{in} + A'BC'_{in} +,$$
$$ABC_{in} + AB'C'_{in},$$
$$= A \oplus B \oplus C_{in}$$

	C_{in}' (0)	C_{in} (1)
A'B' (00)		
A'B (01)		1
AB (11)	1	1
AB' (10)		1

$$C_o = AB + BC_{in} + AC_{in}$$

14

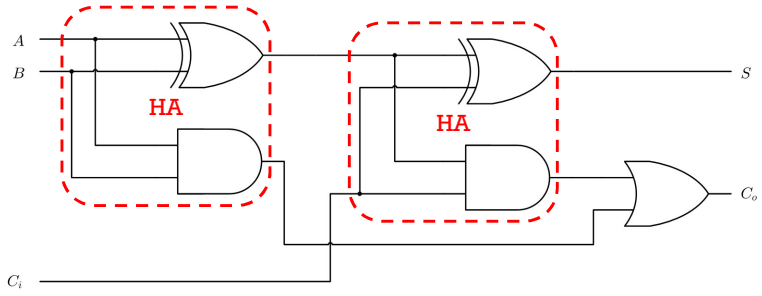
VERIFICATION OF S IN FA

S=A xor B xor C(finally need this form)

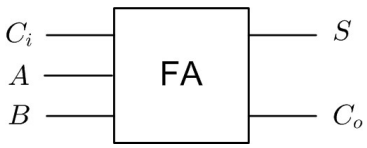
15

FULL ADDER (FA)

- The logic circuit of FA:



- The block diagram of FA:



16

SUMMARY OF HA & FA

- HA forms the basic of binary addition
- FA considers the carry-out bit from the lower significant HA as part of the inputs
- HA and FA can be used for multiple-bit addition
- Their block diagrams are frequently used rather than their logic circuit for high level design

17

HALF SUBTRACTOR (HS)

- A half subtractor is a 1-bit subtractor which do not take borrow bit into consideration
- The sum of two 1-bit binary numbers A and B, can be represented as Di = A - B:
- 0 - 0 = 0
- 0 - 1 = 1, where we need to indicate a borrowing, as the output borrow bit, B_O
- 1 - 0 = 1
- 1 - 1 = 0

18

HALF SUBTRACTOR (HS)

- By summarizing the equations, the truth table of HA is shown below:

A	B	Di	B_o
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

For Di, it is a XOR operation of A and B

For B_O, it is an AND operation of A' and B

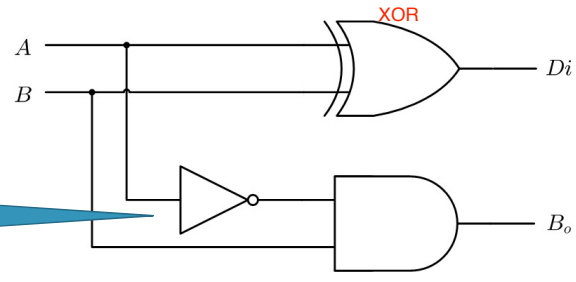
- The Boolean functions of HS:

$$Di = A \oplus B,$$
$$B_o = \bar{A}B$$

19

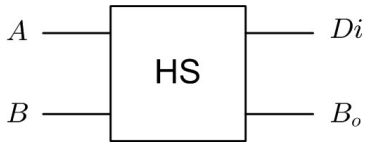
HALF SUBTRACTOR (HS)

- The logic circuit of HS:



HS is very similar to HA, with an additional inverter

- The block diagram of HS:



20

FULL SUBTRACTOR (FS)

- When the operations require more than 1 bit, HS is not useful
  - Borrow-out bits from the lower significant bits are not considered/used at the higher bit operations
- By extending the HS, one can implement a full subtractor (FS) which takes a borrow-in bit as an extra input
  - The borrow-out bit of the lower order bit can be connected to the borrow-in bit of the immediate higher order bit
- FS becomes the basic unit for a multiple-bit subtractor

21

FULL SUBTRACTOR (FS)

Di=A-B-Bin

borrow out bit

- The truth table of FS:

A	B	B_{in}	Di	B_o
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Example: 8-3=5

1000
- 0011
0101

B_{in}: 1110
B_O: 0111

- The Boolean functions of FS:

$$Di = \Sigma(1,2,4,7),$$
$$B_o = \Sigma(1,2,3,7)$$

22

FULL SUBTRACTOR (FS)

$$Di = \Sigma(1,2,4,7),$$
$$B_o = \Sigma(1,2,3,7)$$

- Simplify the Boolean functions of FA:

	B_{in}' (0)	B_{in} (1)
A'B' (00)		1
A'B (01)	1	
AB (11)		1
AB' (10)	1	

$$Di = A'B'B_{in} + A'BB'_{in} +, \\ ABB_{in} + AB'B'_{in}, \\ = A \oplus B \oplus B_{in}$$

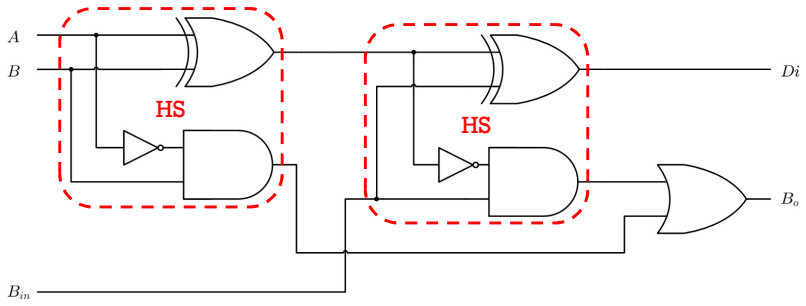
	B_{in}' (0)	B_{in} (1)
A'B' (00)		1
A'B (01)	1	1
AB (11)		1
AB' (10)		

$$B_o = A'B_{in} + BB_{in} + A'B$$

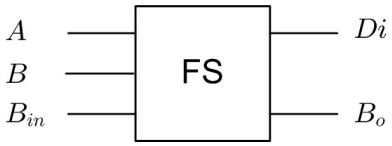
23

FULL SUBTRACTOR (FS)

- The logic circuit of FS:



- The block diagram of FS:



24

EXAMPLE (FULL SUBTRACTOR)

- Example: $A - B = 1110101_2 - 0011100_2 = 117_{10} - 28_{10} = 89_{10} = 1011001_2$

	A	1	1	1	0	1	0	1
-	B	0	0	1	1	1	0	0
	Di	1	0	1	1	0	0	1
	B_{in}	0	1	1	0	0	0	
	B_o	0	0	1	1	0	0	0

25

EXAMPLE (FULL SUBTRACTOR)

- Example: $A - B = 10011000_2 - 01011001_2 = 152_{10} - 89_{10} = 63_{10} = 00111111_2$

	A	1	0	0	1	1	0	0	0
-	B	0	1	0	1	1	0	0	1
	Di	0	0	1	1	1	1	1	1
	B_{in}	B_{in}	B_{in}	B_{in}	B_{in}	B_{in}	B_{in}	B_{in}	
	B_o	B_o	B_o	B_o	B_o	B_o	B_o	B_o	B_o

26

27

PARALLEL ADDER AND SUBTRACTOR

PRALLEL ADDERS AND SUBTRACTORS

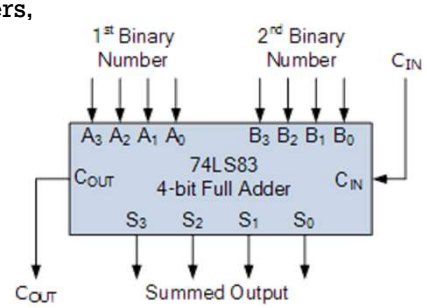
- Binary additions or subtractions could be implemented in either serial or parallel fashion
- Serial addition takes longer time to process as at any given time only a pair of bits are processed
- Parallel addition is much faster as all bits are processed nearly at the same time
- We can use FA and FS as the basic building blocks

28

~commercial package

4-BIT FULL ADDER

- Consider a 4-bit FA to process two 4-bit binary numbers,
  - $A = A_3A_2A_1A_0$ and $B = B_3B_2B_1B_0$
- Inputs: 4 bits for A, 4 bits for B, 1 bit for Carry-in (C_{IN})
- Outputs: 4 bits for Sum (S), 1 bit for Carry-out (C_{OUT})

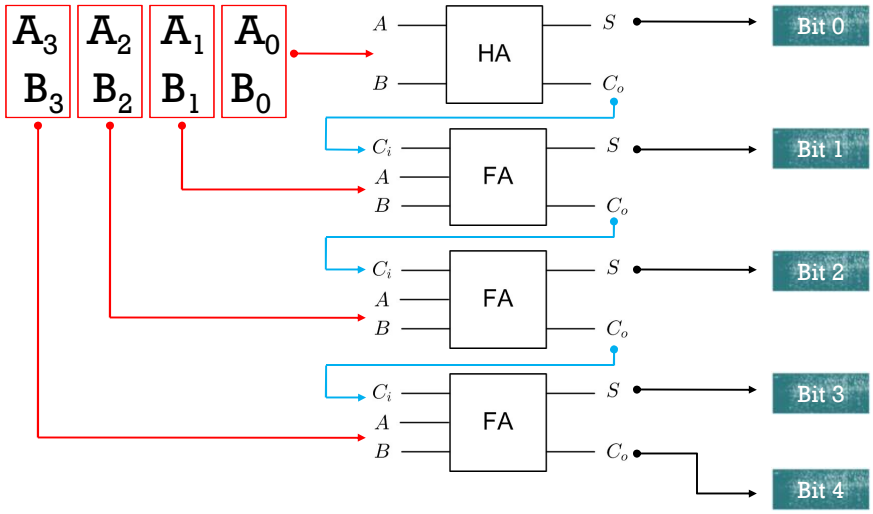


- To build a 4-bit FA, 1x HA and 3x FAs, or 4x FAs are required

29

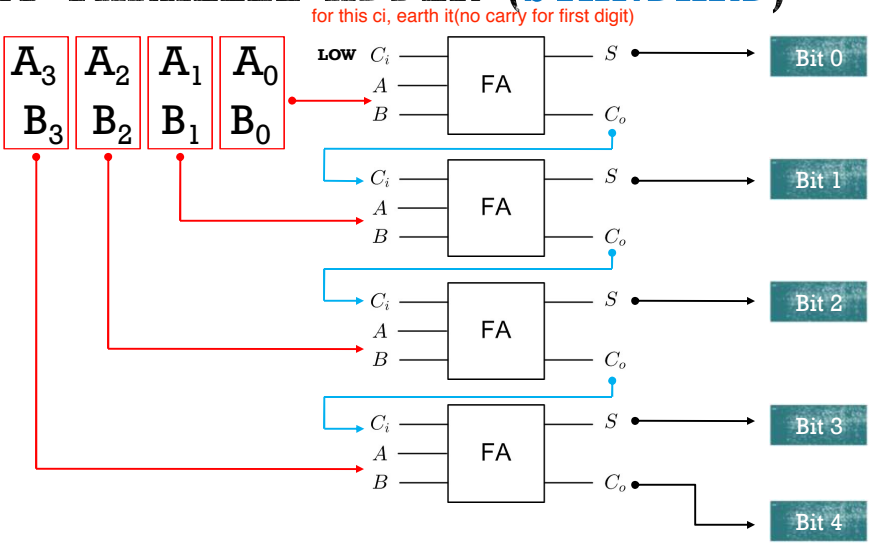
4-BIT PARALLEL ADDER (HYBRID)

because ha and fa also use



30

4-BIT PARALLEL ADDER (STANDARD)



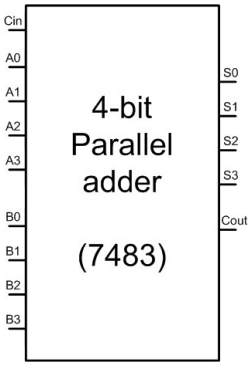
31

COMMERCIAL PACKAGE OF 4-BIT ADDER

- A commercial IC 7483 is a 4-bit parallel adder that implements a FA

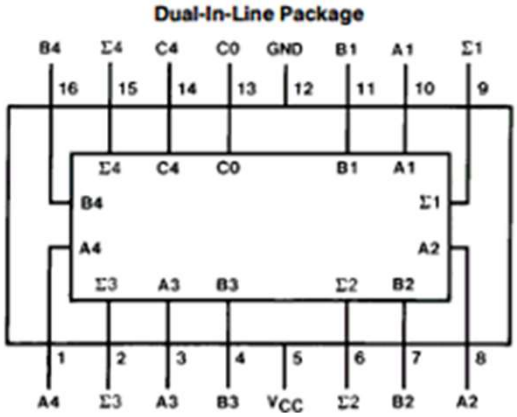
- A total of 9 input pins
  - 4 bits per binary number
  - 1 bit for carry-in

- A total of 5 output pins
  - 4 bits for the sum
  - 1 bit for carry-out



32

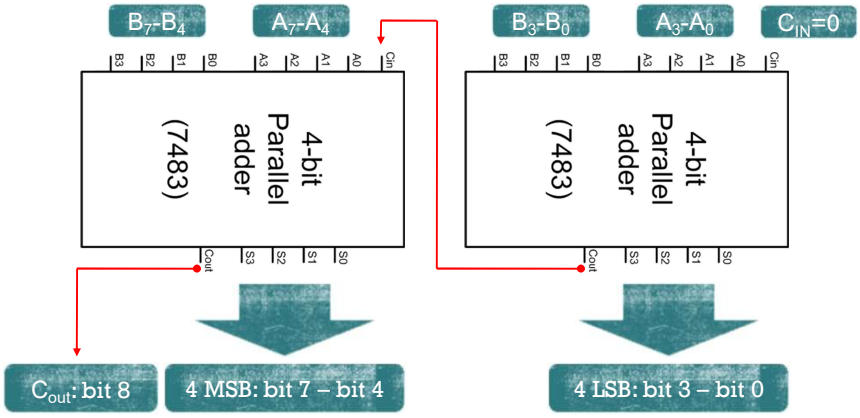
PIN ASSIGNMENT OF 7483



33

8-BIT PARALLEL ADDER USING 7483

- Let's connect the following 2x 7483 to form a 8-bit parallel adder



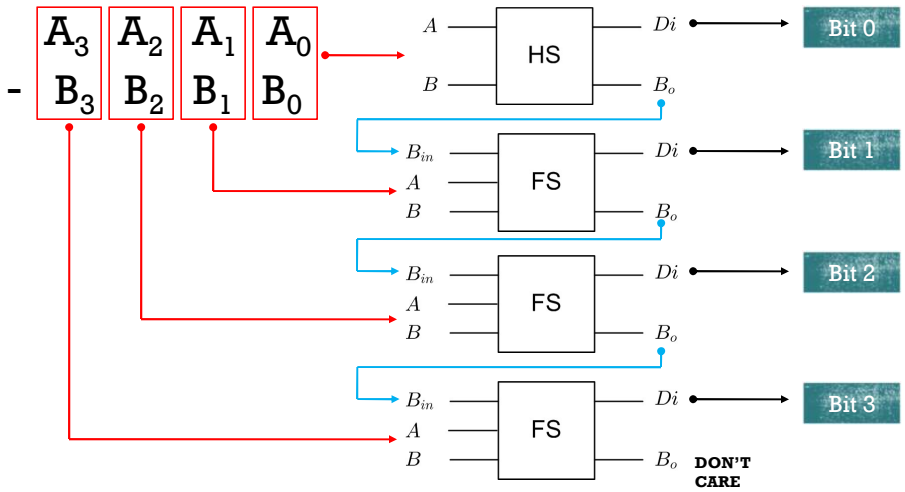
34

4-BIT FULL SUBTRACTOR

- Consider a 4-bit FS to process two 4-bit binary numbers,
  - A = $A_3A_2A_1A_0$ and B = $B_3B_2B_1B_0$
- Inputs: 4 bits for A, 4 bits for B, 1 bit for Borrow-in (B_{in})
- Outputs: 4 bits for Difference (Di), 1 bit for Borrow-out (B_O)
- To build a 4-bit FA, 1x HS and 3x FSs, or 4x FSs are required

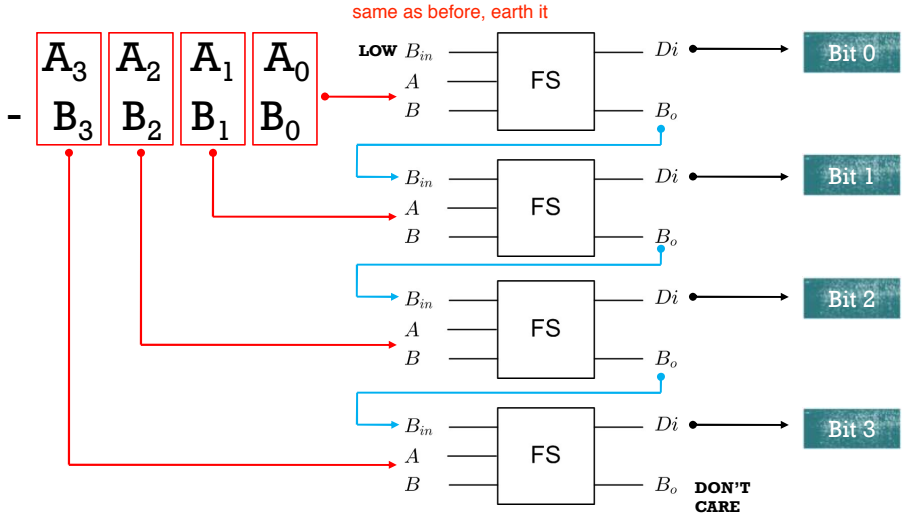
35

4-BIT PARALLEL SUBTRACTOR (HYBRID)



36

4-BIT PARALLEL SUBTRACTOR (STANDARD)



37

for dont care, there is no more next digit, so just dont care

PARALLEL SUBTRACTOR USING 2'S COMPLEMENT

- Binary additions can be used for subtractions
- 2's complement is generated by
  - Flipping the bit values from 1 to 0 or 0 to 1
  - Followed by an addition of 1
- In other words, all high bit subtractor can be implemented using the commercial package parallel adder, such as 7483

38

1-BIT SUBTRACTOR USING 2'S COMPLEMENT

- Consider two 1-bit binary numbers, A and B
- Their difference can be expressed as

$$Di = A - B = A + (-B),$$
$$= A + C_2(B) = A + (\bar{B} + 1) = S$$

  - where $C_2(B)$ denotes the 2's complement of B

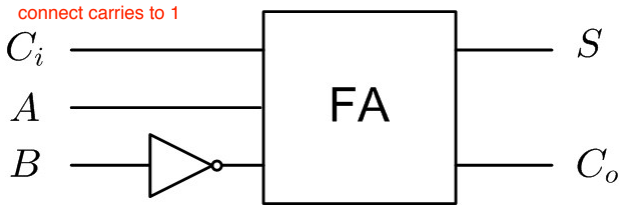
- From the equations, we noticed that
  - Two 1-bit adders are required
  - Three inputs are required, A, B', and 1
    - B' is the 1's complement of B
    - 1 is a HIGH

39

1-BIT SUBTRACTOR USING FA

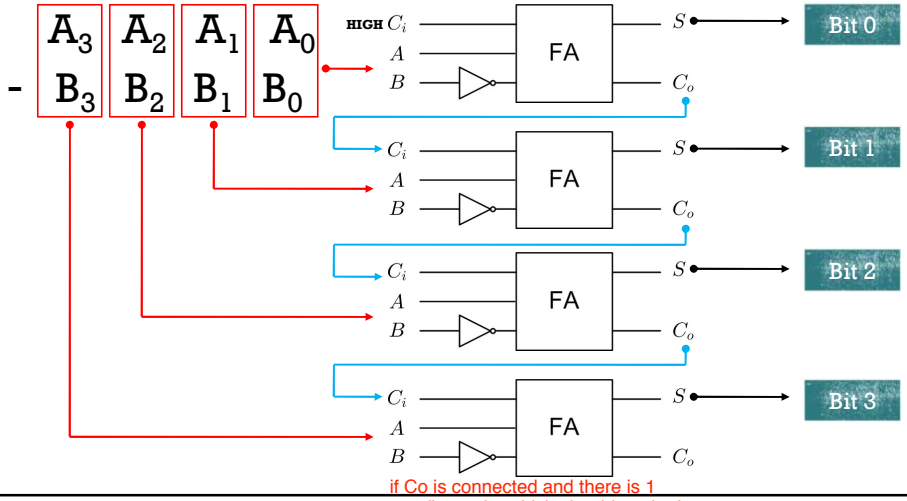
- The block diagram of 1-bit subtractor, with $C_i = 1$

$$Di = S = A + \bar{B} + 1$$



40

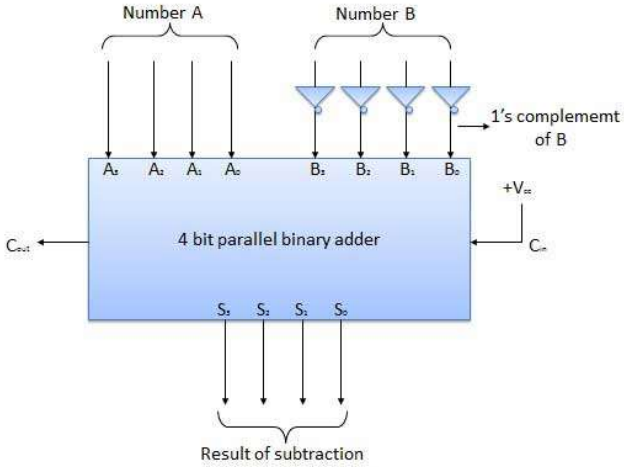
4-BIT SUBTRACTOR USING FA



if Co is connected and there is 1
overflow exist which should not be happen

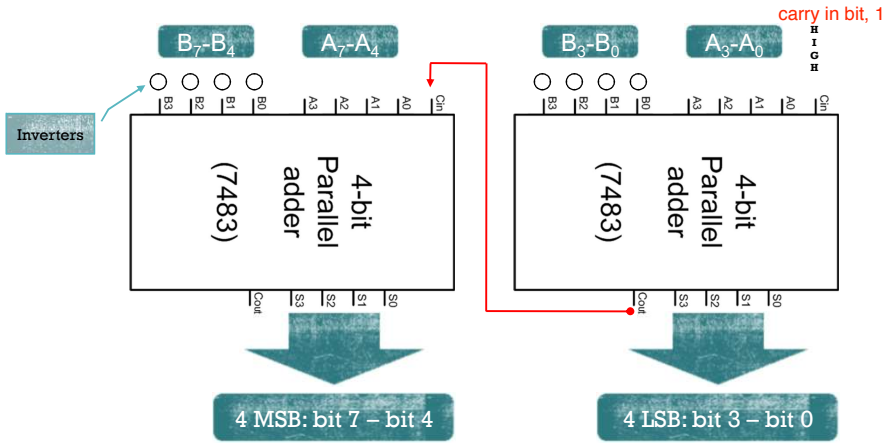
41

4-BIT SUBTRACTOR USING FA



42

8-BIT SUBTRACTOR USING 7483



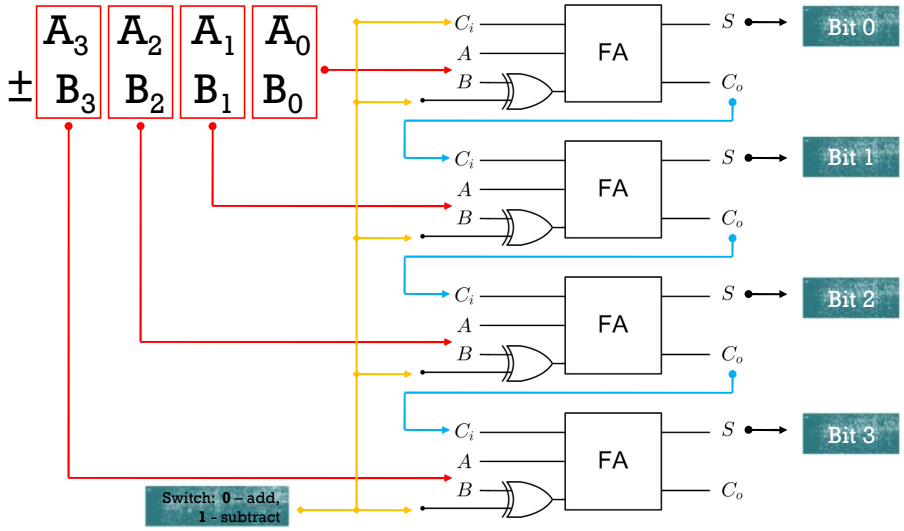
43

UNIVERSAL 4-BIT ADDER/SUBTRACTOR

- Both adder and subtractor can be implemented by FAs
- Is it possible to have one single circuit with a switch to implement both operations?
- Ans: Yes, provided that
- If C_{in} at the 1st FA is 1,
  - A subtraction is selected, and
  - The bits of number B will be 1's complemented

44

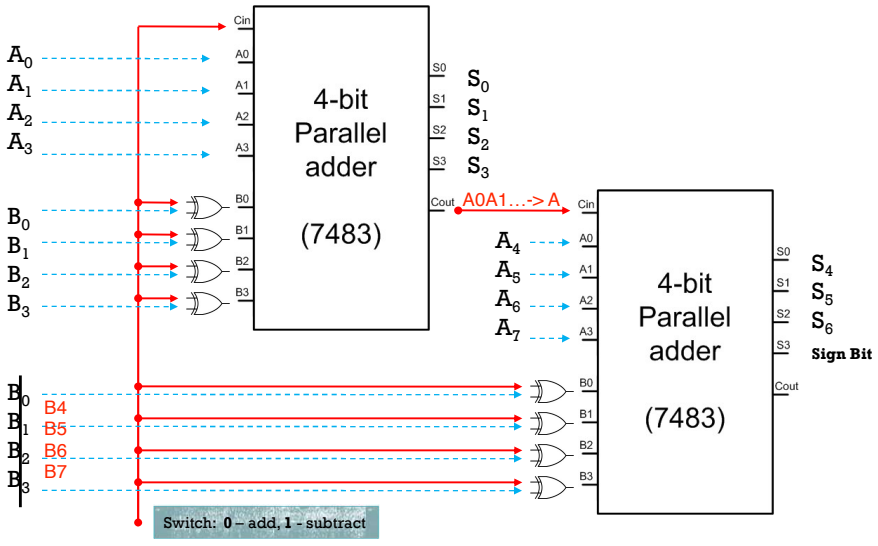
UNIVERSAL 8-BIT ADDER/SUBTRACTOR



45

control bit to the carry bit
and also help us to determine whehter we take carry bit or not

UNIVERSAL 4-BIT ADDER/SUBTRACTOR



46

ANY QUESTIONS ?

47