62070501056 กิฟ้ลณี อิ่มใจ
62070501064 อรวิภา คูเจริญไพศาล

Lab 5 - Binary Adder/Subtractor

Objectives

- To build subcircuits in Proteus
- To apply combinatorial knowledge to design and implement a 4-bit adder from full adders.

Procedure

Follow the instruction.

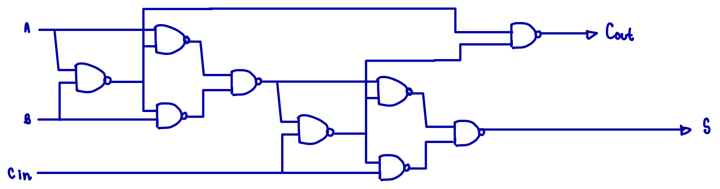
1. Design a 1-bit full adder(FA) using NAND gates only.

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

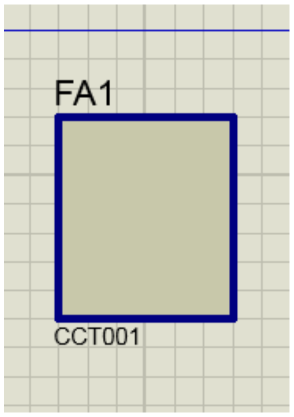
Cin\AB	00	01	11	10
0	0	0	1	0
1	0	1	1	1

Boolean Expressions: S = $z \oplus (x \oplus y)$ Cout = $z(xy' + x'y) + xy$

Sketch FA circuit using NAND gates only:



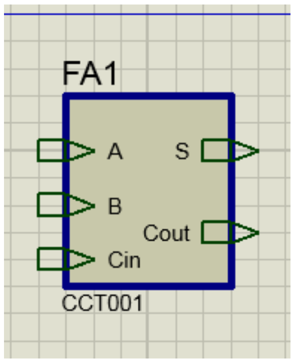
2. In Proteus, add the following Parts: LOGICPROBE(BIG), LOGICSTATE, NAND
3. Build a 1-bit full adder circuit as a subcircuit. Use the TERMINAL Mode( ) to add terminals. Use WRITE LABEL Mode( ) to change the subcircuit name SUB? to FA1.



62070501056 กิฟ้าคณี อิ่มใจ

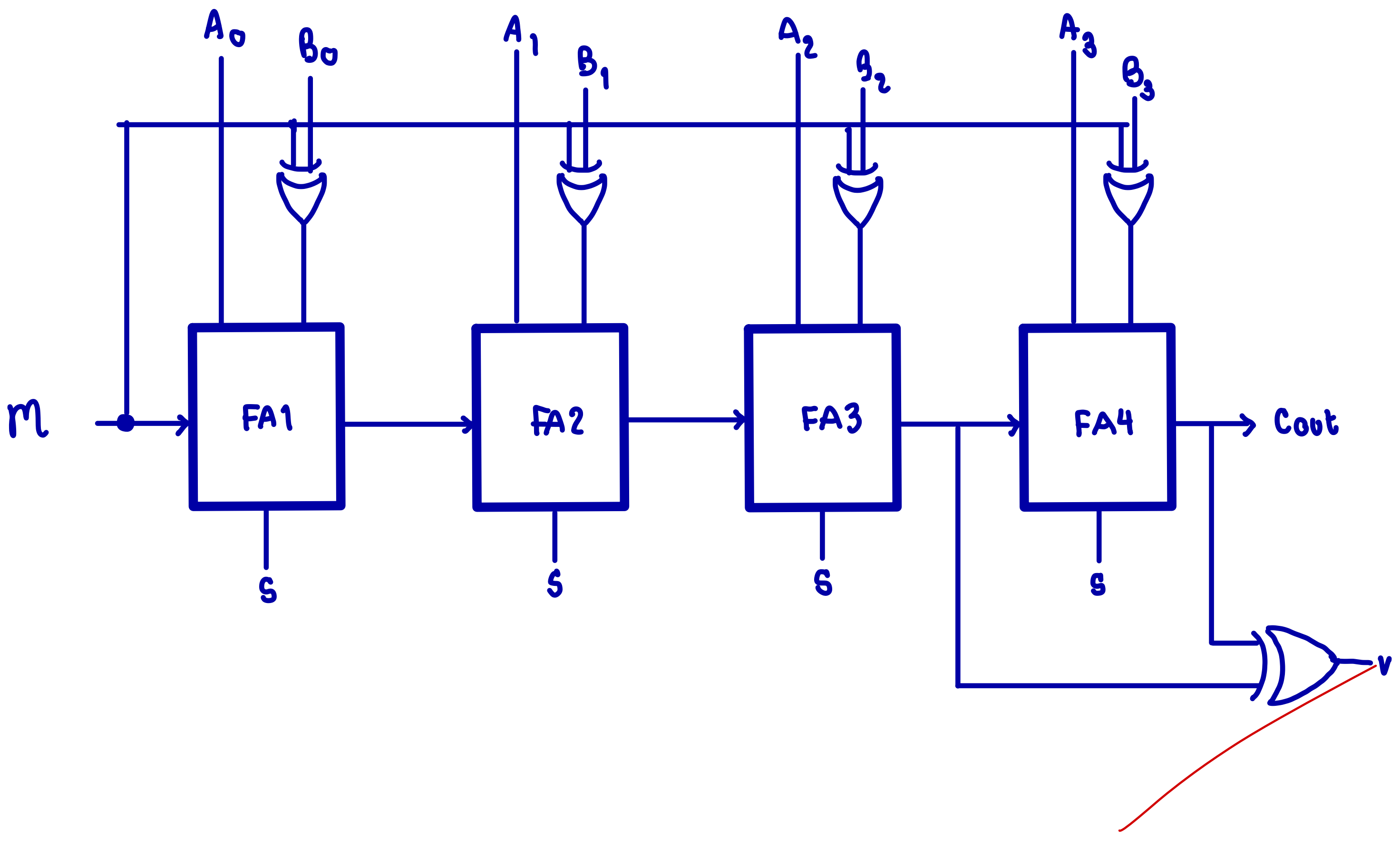
62070501064 อรวิภา คูเจริญไพศาล

4. Use SUBCIRCUIT Mode. Select INPUT to define input ports, select output to define output ports.



5. Make multiple copies of FA1. Rename them differently to make a 4-bit full adder. Test the adder circuit and fill out the worksheet.
6. Modify the circuit to achieve a 4-bit adder/subtractor.

Sketch 4-bit full adder diagram:



62070501056 ลิฟัลณี อิ่มใจ
62070501064 อรวิภา คูเจริญไพศาล

CPE 223 Lab 5 Worksheet

<u>This part must be handed in prior to the lab section.</u> This report template is useful to prepare for each exercise's checkoff. Fill in answers for the questions requested by exercise; you may use a different sheet of paper if more convenient, but be sure to follow the template.

The following table is a guide line to verify your circuit. After constructing and testing the circuit, demonstrate each step to a TA and fill in the tables.

Decimal	2's Compl. Binary	Decimal	2's Compl. Binary	Decimal	2's Compl. Binary	Decimal	2's Compl. Binary
0	0000	4	0100	-8	1000	-4	1100
1	0001	5	0101	-7	1001	-3	1101
2	0010	6	0110	-6	1010	-2	1110
3	0011	7	0111	-5	1011	-1	1111

4-bit Adder: S = A + B

A	a3	a2	a1	a0		B	b3	b2	b1	b0		C_{in}		S	s3	s2	s1	s0	
3	0	0	1	1		2	0	0	1	0		0		5	0	1	0	1	
4	0	1	0	0		3	0	0	1	1		0		7	0	1	1	1	
7	0	1	1	1		-1	1	1	1	1		0		6	0	1	1	0	
-2	1	1	1	0		-3	1	1	0	1		0		-5	1	0	1	1	
7	0	1	1	1		-8	1	0	0	0		0		-1	1	1	1	1	
7	0	1	1	1		1	0	0	0	1		0		-8	1	0	0	0	[C0 V1]
-4	1	1	0	0		-5	1	0	1	1		0		7	0	1	1	1	[C1 V1]

In a two's complement system, how does an overflow occur? Specify an overflow in the table above.

overflow ตอนเกิน bit ที่เราต้องการคือ 4-bit
และ การบวกจะ overflow ตอนเครื่องหมายเหมือนกัน

4-bit Subtractor: S = A - B

A	a3	a2	a1	a0		B	b3	b2	b1	b0		C_{in}		Cout	s3	s2	s1	s0	S	
3	0	0	1	1		-2	1	1	1	0		1		0	0	1	0	1	5	
4	0	1	0	0		-3	1	1	0	1		1		0	0	1	1	1	7	
7	0	1	1	1		1	0	0	0	1		1		1	0	1	1	0	6	
-2	1	1	1	0		3	0	0	1	1		1		1	1	0	1	1	-5	
7	0	1	1	1		-8	1	0	0	0		1		0	1	1	1	1	-1	[V1]
7	0	1	1	1		-1	1	1	1	1		1		0	1	0	0	0	-8	[V1]
-4	1	1	0	0		5	0	1	0	1		1		1	0	1	1	1	7	[V1]

Discuss the similarity/difference between two's complement addition and subtraction.

ต่าง : คอมพิวเตอร์คิดการลบไม่ได้หรือวุ่นวาย แต่สามารถคิดการบวกด้วยตัวเลขติดลบได้ง่ายกว่า A-B ✗ | A+(-B) ✓
เหมือน : คำตอบ

Check off. TA: Lab5_2A_G8
[illegible]
18/9/63