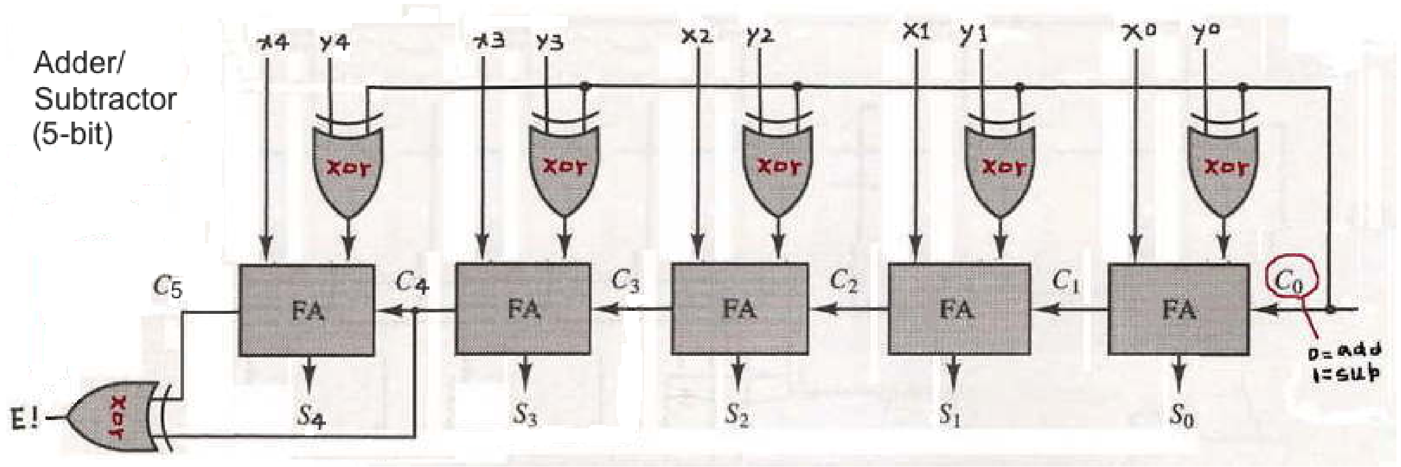
CSc 137 Verilog Programming #3: Adder/Subtractor

1. In this lab we expand your Verilog program from the previous assignment to make it also do subtraction. Submit only 1 program file (call it ***AddSub.v***) to your assigned "3rdPrgAssig" folder. (If you've already submitted with a different name, it's OK.)

2. This time, you code the test module yourself. Expand from the previous test module. Again, make and use a "V2" (revision 2) folder under that folder if you need to re-submit. (And, use "V3" if still a newer revision, etc.) Misplacing files may now get penalized.



3. The first carry-in bit **C0**, as shown in the diagram, can be given 0 or 1. It does subtraction when it is 1.

4. The full-adder module is not changed from the previous assignment. The "xor" logic gates are part of the adder-subtractor, not in the full adder. The xor gates "flip" the *y* bits when *C0* is given 1 (to perform subtraction).

5. The *xor* of the last two carries (C4 and C5) makes the **E**, an indicater of Exception (overflow). If so, C5 should be part of the result, placed in the front. Otherwise, the default result is fine.

6. The definition of the full-adder module does not include an *xor* gate which should be outside of it as the diagram shows. They are really part of the whole adder-subtractor module. The xor gates flip *y bits* when *C0* is 1.

7. The *xor* of the last two carries (C4 and C5) generates the **E** indicater. If E is 0, the sum bits are OK as a 2's complement result. If E is 1, then C5 is needed to be added to the front of the sum. Otherwise, the sum will not be correct as the signed 2's complement bitmap. E is the true overflow indicator. (Even when C5 is 1 but E 0, there is no overflow.)
8. The keyboard input asks for two numbers X and Y, ranging from 0 to 15 (by entering '00'to '15') each. Then, enter either a plus or minus sign ('+' or '-'). Copy the demo and test-run it:

```
atoz[32]% cp  -p  ~changw/html/137/prg/3/demo-a.out  .
atoz[33]% demo-a.out

Enter X (range 00 ~ 15):
10
Enter Y (range 00 ~ 15):
03
Enter either '+' or '-':
+
X=01010 (10) Y=00011 ( 3) C0=0
Result=01101 (as unsigned 13)
C4=0 C5=0 E=0
Since E is 0, C5 is not needed.

atoz[34]% demo-a.out
Enter X (range 00 ~ 15):
12
Enter Y (range 00 ~ 15):
13
Enter either '+' or '-':
-
X=01100 (12) Y=01101 (13) C0=1
Result=11111 (as unsigned 31)
C4=0 C5=0 E=0
Since E is 0, C5 is not needed.
```

```
atoz[35]% demo-a.out
Enter X (range 00 ~ 15):
01
Enter Y (range 00 ~ 15):
15
Enter either '+' or '-':
-
X=00001 ( 1) Y=01111 (15) C0=1
Result=10010 (as unsigned 18)
C4=0 C5=0 E=0
Since E is 0, C5 is not needed.

atoz[36]% demo-a.out
Enter X (range 00 ~ 15):
11
Enter either Y (range 00 ~ 15):
15
Enter either '+' or '-':
+
X=01011 (11) Y=01111 (15) C0=0
Result=11010 (as unsigned 26)
C4=1 C5=0 E=1
Since E is 1, correct with C5 in front: 011010
```

- A List of Useful Linux and vi Commands
- Access Dropbox from a Linux Shell