

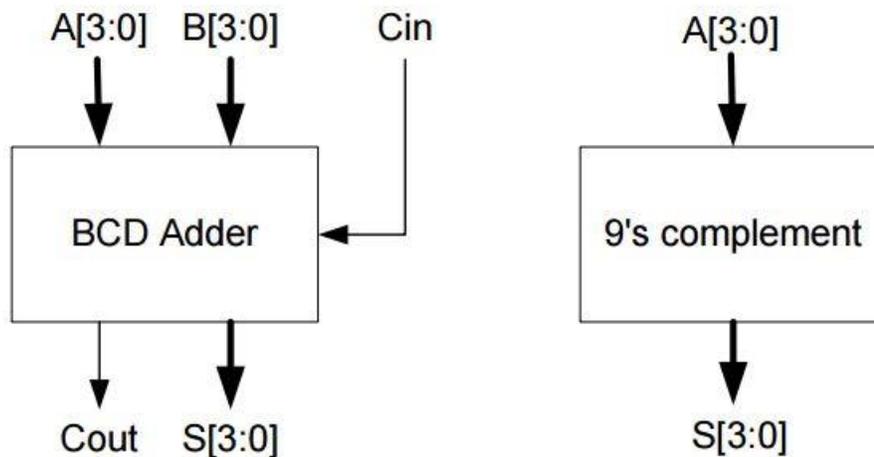


Digital Design

Sheet 6

- 1) Design a 4-bit Arithmetic Logic Unit (ALU) that adds, subtracts, decrements and increments.
- 2) Imagine that we have these two modules ready to use:
 - a. BCD adder
 - b. 9's complement circuit

The block diagrams for these modules are shown below: (A, B, S are all BCD digits)



For example if A="1001" (BCD of decimal 9) and B="0101" (BCD of decimal 5) then the outputs from BCD Adder will be Cout=1 and S = "0100" ("1 0100" is BCD of decimal 14).

And if A="0111" then the output from 9's complement will be S="0010"

Now we want to use the aforementioned modules to:

Design a 4-digit BCD adder-subtractor using four BCD adders (as shown above) and four 9's complement circuit (as shown above) and multiplexers. Your circuit has 3 set of inputs:

- 1) $X_3X_2X_1X_0$ (4-digit BCD input; for example 4896 or "0100 1000 1001 0110")
- 2) $Y_3Y_2Y_1Y_0$ (4-digit BCD input; for example 1267 or "0001 0010 0110 0111")
- 3) M (mode input; M=0 means add and M=1 means subtract)

And it has two sets of outputs:

- 1) $Z_3Z_2Z_1Z_0$ (4-digit BCD output showing the result of the add or subtract; for example the result of the add will be 6163 or "0110 0001 0110 0011" and the result of the subtract will be 3629 or "0011 0110 0010 1001" for the above examples)
- 2) Cout (Carry output)

Use block diagrams for each component, showing only inputs and outputs.

- 3) Suppose the adder-subtractor circuit shown in Fig. 1 has been designed for two's complement numbers. It computes the sum $Z = X + Y$ when the control line $SUB = 0$ and the difference when $SUB = 1$. An overflow flag v is to be added to the circuit, but it is not possible to access internal lines. In other words, only those data and control lines appearing in the figure can be used to compute v . Construct a suitable logic circuit to generate v .

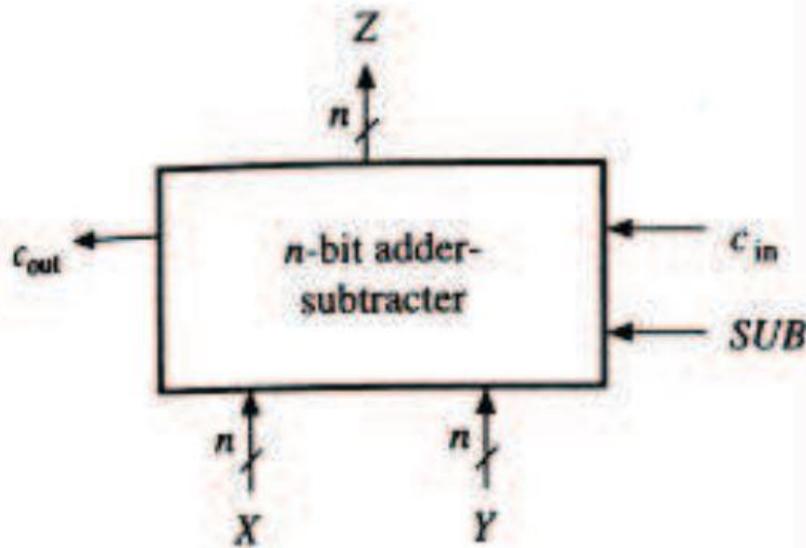


Fig. 1: An n -bit adder-subtractor circuit

- 4) Design an ALU to implement $A + B$, $A - B$, NAND, NOR, AND, OR, EQUAL and NOT EQUAL using three control bits. Give a table to illustrate how you would map each function to some control pattern. Draw your ALU design with two 1-bit inputs and one 1-bit output based on the table you just generated.
- 5) Fabricate your single-bit ALU design in problem (4) into a block with clearly defined interfaces (signal inputs and outputs). Assemble a two-bit ALU at block level based on the block you just fabricated.