EGR 270
Fundamentals of Computer Engineering
File: EGR270SampleLab

Lab Report Format

Each lab report submitted should contain the following four sections:
1) Title Page
2) Preliminary Work
3) Laboratory Results
4) Discussion/Conclusion

Title Page

The Title Page should contain the following items as a minimum: Lab Number, Lab Title, your name, the course number and name, and the semester.

Preliminary Work

Even though the Preliminary Work is checked at the beginning of each class, it must also be included in the lab report where it will be graded in detail. Be sure to make the Preliminary Work as readable as possible by including all given information or instructions and by explaining your steps.

Preliminary Work will generally require **documented circuit layouts**. This layout must include the following information:

A. **Pinouts** - Include pinouts for all IC's to be used in the experiment

B. **Detailed Circuit Diagram** - The circuit diagram should include
  - IC numbers (U1, U2, …)
  - pin numbers
  - labeling for all input signals, output signals, and intermediate signals
  - all input and output devices, such as switches, clocks, LEDs, 7-segment displays, etc.

C. **IC Diagram** - Show each IC to be used, arranged as they will appear on the circuit board. Label each IC (U1, U2, ….). Show the connections for Vcc and ground for each IC.

Laboratory Work

Clearly present the results of all testing performed in lab. List any changes to the design. Also see the lab guide for additional requirements for this section.

Discussion/Conclusion

Clearly and specifically state whether the objectives of the lab were accomplished. Did the circuits tested produce the expected results and what design principles were verified by these results? What did you prove or verify in the lab? Discuss any significant design changes. Also see the lab guide for additional requirements for this section.

Refer to the **Sample Lab** and the **Sample Lab Report** on the following pages. Include all sections indicated in each report. Reports should be presented neatly and professionally. Use proper grammar, employ good writing style, and make the reports as readable as possible. Readability of reports is greatly enhanced by simple techniques such as:

- stating what you are about to show or adding comments at various points when you display your results
- adding titles or headings before equations, graphs, tables, etc.

Sample Lab Guide

EGR 270
Fundamentals of Computer Engineering
File: SAMPLE_LAB.DOC

Lab # ?? (Sample Lab)

2-bit Adder/Subtractor with Carry-in/Borrow-in

A. Objectives

The objective of this laboratory is the design and realization of adding and subtracting circuits using combinational switching circuits.

B. Introduction

It is expected that the student reviews those topics relating to modeling switching functions using Karnaugh Maps and Boolean algebra. The student should also review the function definitions of NAND and XOR (exclusive-OR) gates.

C. Preliminary Work

1. Construct the truth table for a 2-bit (logic variables B and A) binary adder/subtractor with a carry-in/borrow-in input. An input logic signal called X is to control the circuit to perform either addition or subtraction as follows:
   if X = 0, perform the addition B + A
   if X = 1, perform the subtraction B - A
   Inputs to the circuit include B, A, and X as described above, as well as Cin/Bin, the carry-in/borrow-in bit. The circuit should compute the result, referred to as SUM/DIF and the carry-out/borrow-out, referred to as Cout/Bout.
2. Use K-maps and Boolean algebra to simplify the equations for each output signal.
3. Design a reasonably simplified or minimal multilevel realization using only 2-input NAND gates (7400), XOR gates (7486), and inverters (7404). Draw a logic diagram.
4. Explain any minimization techniques used.
5. Present a ***documented circuit layout*** for your design including:
   - pinouts for all IC's used
   - detailed circuit diagram with pin numbers labeled
   - IC diagram showing the ordering of the IC's on the breadboard and the numbering (U1, U2, …) of each IC

D. Laboratory Work

1. Construct your circuit from the wiring list. Note any changes.
2. Test your circuit, filling in a truth table prepared for recording the circuit response for all possible input combinations.

Sample
Lab
Report

Lab # ?? (Sample Lab)

2-bit Adder/Subtractor with Carry-in/Borrow-in

By: John Doe
Partner: Joe Smith

EGR 270
Fundamentals of Computer Engineering
Spring 20??

Preliminary Work

1. The truth table shown below is for a 2-bit binary adder/subtractor with a carry-in/borrow-in input. The inputs and outputs are described as follows:

Inputs:

B, A	binary inputs
Cin/Bin	carry input or borrow input
X	input to indicate whether the circuit is to perform addition or subtraction

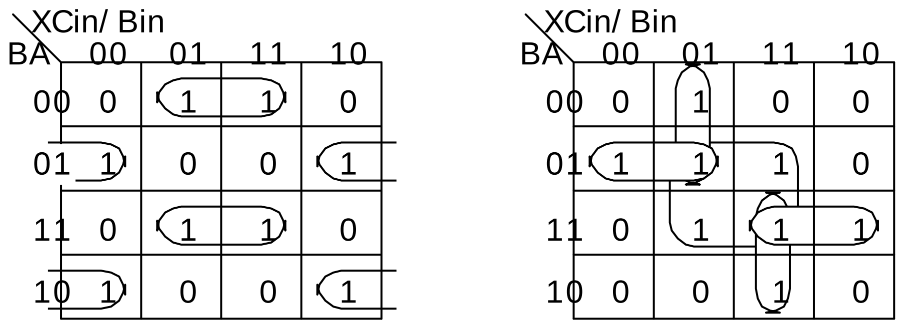
Outputs:

SUM/DIFF= B + A if X = 0
= B - A if X = 1
Cout/Bout carry output or borrow output

Helpful Hint: Clearly present the solution so that the reader is aware of all details. Either write out the given information or show it in some other form.

Inputs				Outputs	
X	Cin/Bin	B	A	SUM/DIFF	Cout/Bout
0	0	0	0	0	0
0	0	0	1	1	1
0	0	1	0	1	0
0	0	1	1	0	0
0	1	0	0	1	1
0	1	0	1	0	1
0	1	1	0	0	0
0	1	1	1	1	1
1	0	0	0	0	0
1	0	0	1	1	0
1	0	1	0	1	0
1	0	1	1	0	1
1	1	0	0	1	0
1	1	0	1	0	1
1	1	1	0	0	1
1	1	1	1	1	1

2. Shown below are the K-maps and logic equations for the two output signals: SUM/DIFF and Cout/Bout.



K-map for K-map for

Logic equations are generated for each output in Sum-Of-Products (SOP) form using the groupings indicated on the K-Maps above.

$$SUM/DIFF = Cin/Bin \bullet \overline{B} \bullet \overline{A} + \overline{Cin/Bin} \bullet \overline{B} \bullet A + Cin/Bin \bullet B \bullet A + \overline{Cin/Bin} \bullet B \bullet \overline{A}$$

$$Cout/Bout = Cin/Bin \bullet A + \overline{X} \bullet Cin/Bin \bullet \overline{B} + X \bullet B \bullet A + \overline{X} \bullet \overline{B} \bullet \overline{A} + X \bullet Cin/Bin \bullet B$$

3,4. The logic equations generated above are in minimal SOP form (with ANDs, ORs, and inverters only), but they can be further minimized using XOR gates as follows.

$$SUM/DIFF = Cin/Bin \bullet \left(\overline{B} \bullet \overline{A} + B \bullet A\right) + \overline{Cin/Bin} \bullet \left(\overline{B} \bullet A + B \bullet \overline{A}\right)$$

$$SUM/DIFF = Cin/Bin \bullet \left(\overline{B \oplus A}\right) + \overline{Cin/Bin} \bullet (B \oplus A)$$

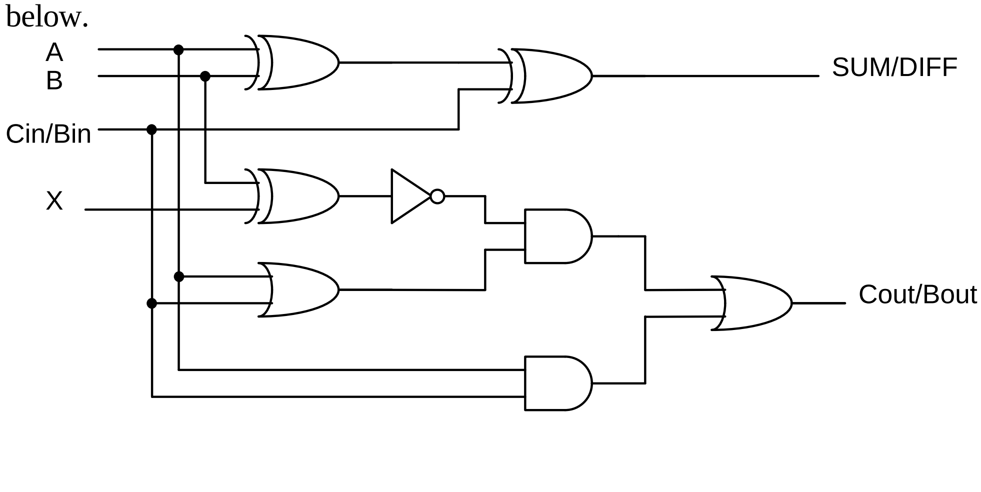
$$\boxed{SUM/DIFF = Cin/Bin \oplus (B \oplus A)}$$ (final minimized result)

$$Cout/Bout = A \bullet \left(\overline{X} \bullet \overline{B} + X \bullet B\right) + Cin/Bin\left(\overline{X} \bullet \overline{B} + X \bullet B\right) + Cin/Bin \bullet A$$

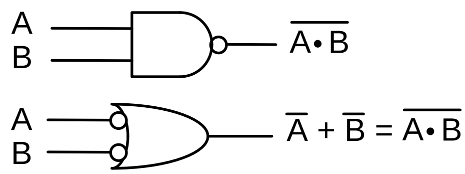
$$Cout/Bout = (A + Cin/Bin) \bullet \left(\overline{X \oplus B}\right) + Cin/Bin \bullet A$$

$$\boxed{Cout/Bout = (A + Cin/Bin) \bullet \left(\overline{X \oplus B}\right) + Cin/Bin \bullet A}$$ (final minimized result)

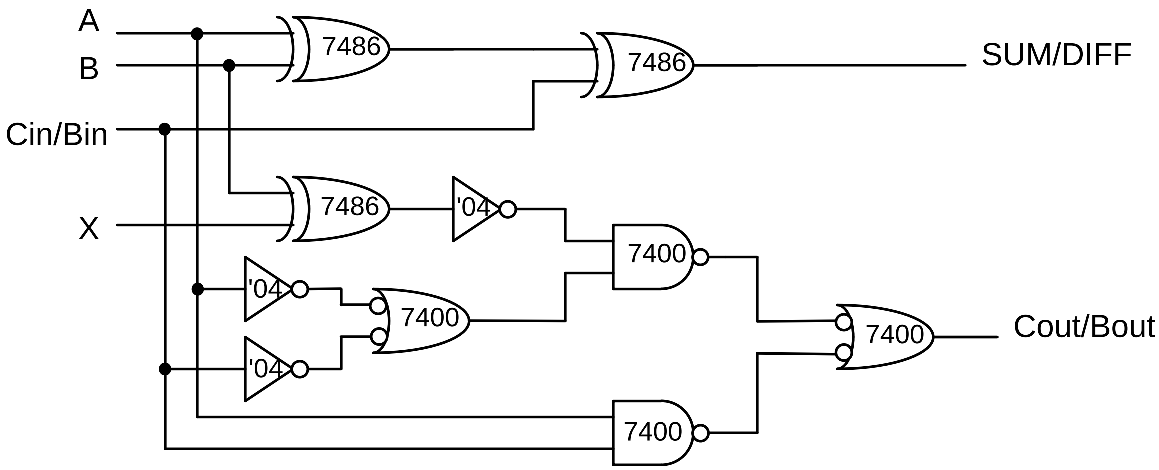
The two output equations generated above could be implemented using 8 gates (4 IC's) as shown below.



The AND gates and OR gates can easily be replaced by NAND gates in the previous circuit. Shown below are the two ways of drawing a NAND gate:



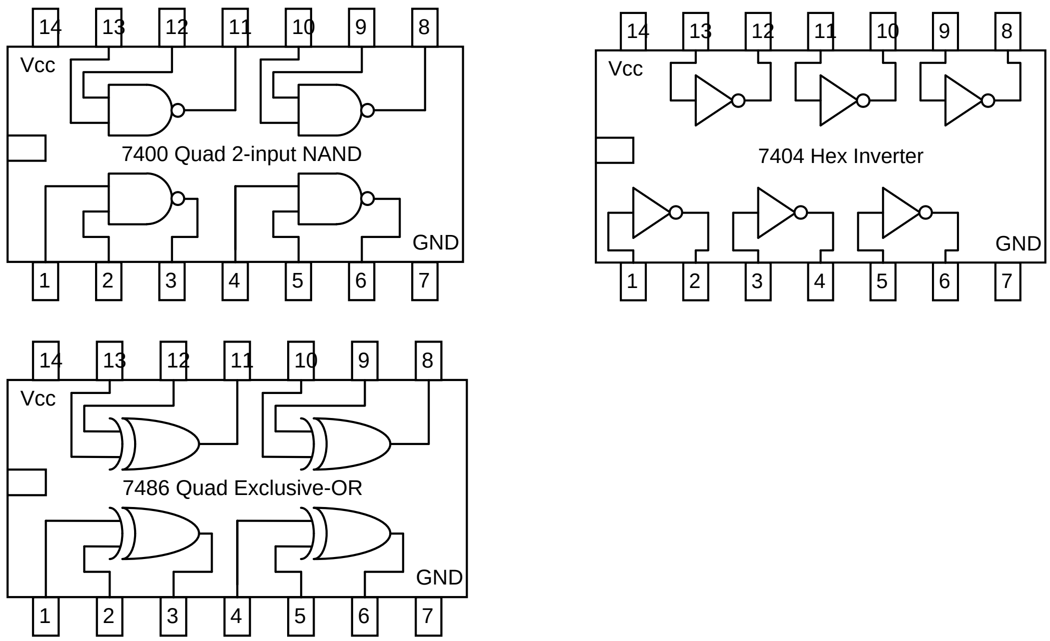
If “bubbles” are added to the circuit on the previous page to change the AND gates and OR gates into NAND gates and then the circuit is “balanced” using inverters, the following circuit is generated.



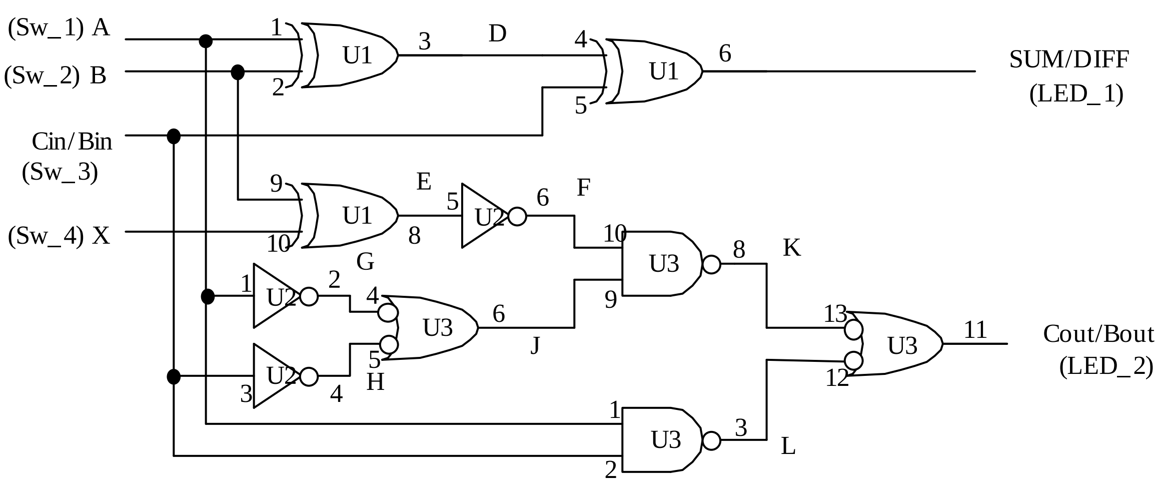
The circuit above meets the design requirement of using only 2-input NAND gates, XOR gates, and inverters. The total number of gates required is 10, but only 3 IC’s are needed.

5. Circuit documentation:

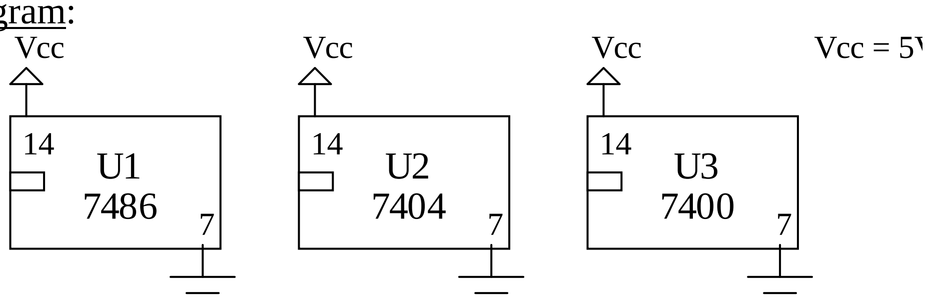
Pinouts for the required IC's:



Adder/Subtractor Circuit Diagram:



IC Diagram:



Laboratory Work

1. The adder/subtractor circuit was constructed according to the circuit diagram shown previously. No changes were needed.

2. The circuit was tested for all 16 input switch combinations. An output of 1 (HIGH) was indicated by the corresponding LED being lit and a 0 (LOW) was indicated by the LED not being lit.

Inputs				Outputs	
X	Cin/Bin	B	A	SUM/DIFF	Cout/Bout
0	0	0	0	0	0
0	0	0	1	1	1
0	0	1	0	1	0
0	0	1	1	0	0
0	1	0	0	1	1
0	1	0	1	0	1
0	1	1	0	0	0
0	1	1	1	1	1
1	0	0	0	0	0
1	0	0	1	1	0
1	0	1	0	1	0
1	0	1	1	0	1
1	1	0	0	1	0
1	1	0	1	0	1
1	1	1	0	0	1
1	1	1	1	1	1

Discussion/Conclusion

The truth table generated from testing the adder/subtractor circuit is identical to the truth table generated in the Preliminary Work, therefore the design presented has been validated. No problems occurred and no design modifications were required.