EGCP 1010
Digital Logic Design (DLD)
Laboratory #2

Introduction to Combinational

Logic Design

Prepared By:
Alex Laird

on

September 19, 2007

Lab Partner:

Objective:

The goal of this laboratory is to expand the students knowledge of laboratory equipment, perform procedures, and implement the logic design of a Full Adder. The student should have a better knowledge of the laboratory and should successfully build a Full Adder by the end of this lab.

Procedure and Results:

For this lab, the class was introduced to a Full Adder. A Full Adder will take two binary bits, A and B, and a potential binary carry bit, C, and add the three together. The result will be a two bit binary answer of 0_{10} (00_2), 1_{10} (01_2), 2_{10} (10_2), or 3_{10} (11_2). The potential outputs from the Full Adder are shown in Table 1.

Table 1: Full Adder Truth Table

C_{IN}	A	B	C_{OUT}	Sum
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

From this truth table, we can deduce the K-Maps shown in Table 2 and Table 3.

Table 2: C_{OUT} K-Map

0	0
0	1
1	1
0	1

Table 3: Sum K-Map

0	1
1	0
0	1
1	0

From these K-Maps, you can deduce the following Sum-Of-Products (SOP) solutions.

C_{OUT} = AB + AC + BC
Sum = A'B'C + A'BC' + ABC + AB'C'

All of the sets are EPITs. The K-Map in Table 3 is a checkerboard pattern, starting with a 0 (false) in the 00_2 position. This pattern is that of an XOR solution, so the Sum SOP solution can be rewritten.

Sum = A ⊕ B ⊕ C

Given that a three (3) input XOR gate is not available in the lab, nor is a three (3) input OR gate available in the lab, the equations must be factored to the following to, final solutions.

C_{OUT} = (AB + BC) + BC
Sum = (A ⊕ B) + C

These solutions are put into a full, gate-level schematic to represent the logic flow. The full, gate-level schematic is shown in Figure 1.

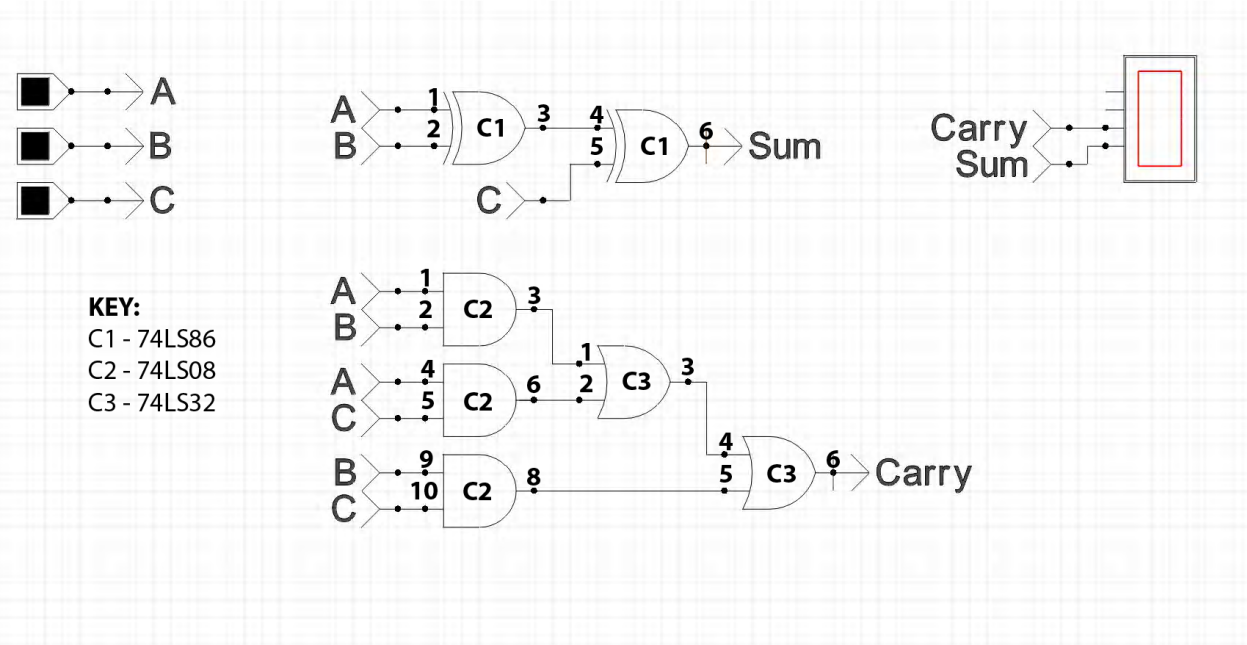

Figure 1: Gate-Level Schematic of Full Adder

Finally, this full, gate-level schematic was implemented into circuitry.

A 74LS86 (two (2) XOR), 74LS08 (two (2) AND), and 74LS32 (two (2) OR) chip were all connected to the +5V power supply (via pin fourteen (14)) and ground (via pin seven (7)).

For the 74LS86 chip, inputs for gate one (1) were connected to binary switches S1 and S2. The output from gate one (1) was passed on as an input to gate two (2). The second input for gate two (2) was connected to binary switch S3. The output from gate two (2), which is the Sum bit, was connected to the Logic Monitor LED, #8.

For the 74LS08 chip, the inputs for gate one (1) were connected to binary switches S1 and S2. The inputs for gate two (2) were connected to binary switches S1 and S3. The inputs for gate three (3) were connected to binary switches S2 and S3. The output for gate one (1) was sent to gate one (1) of the 74LS32 chip. The second input for gate two (2) of the 74LS32 chip comes from the output of gate two (2) on the 74LS08 chip. The output from gate one (1) on the 74LS32 chip was sent to gate two (2) of the 74LS32 chip. The second input of gate two (2) on the 74LS32 chip comes from the output of gate three (3) on the 74LS08 chip. The output from gate two (2) on the 74LS32 chip, which is the Carry bit, was connected to the Logic Monitor LED, #7.

After all of the circuits were connected, binary switches S1, S2, and S3 were set to on (1) and off (0) in every possible combination to see if the output would be identical to the initial truth table shown in Table 1. The results are shown in Table 4.

Table 4: Logic Monitor LED Outputs

C_{IN}	A	B	C_{OUT}	Sum
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	1	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

We observed that the circuit worked correctly everywhere except when S1, the Carry bit, was on, and S2 and S3 were off. For some reason, both bits were high when only the Sum should have been high. After a bit of debugging and consulting Dr. Kohl, we came to the conclusion that our 74LS08 chip was bad. We replaced the chip and the Full Adder worked properly. All outputs matched that of Table 1.

Conclusion and Suggestions:

To me, this lab was very fun, and illustrated the extreme usefulness of binary and circuitry operations. Even though this lab was very simple, it illustrated very plainly to me that all of the top level things that a complicated computer can do boil down, eventually, to the simplicity of binary circuitry.

Of course, my partner and I ran into a little bit of trouble with our faulty chip, but past that the lab experience was enjoyable and we had no trouble carrying it out.

Questions:

I. How many TTL chips (not gates) were needed in your design? How many unused gates were there in your design?

Three (3) TTL chips were used in my design: The 74LS86, 74LS08, and 74LS32 chips. There were two (2) unused gates on the 74LS86 chip, one (1) unused on the 74LS08 chip, and two (2) unused on the 74LS32 chip.

II. What is the minimum number of Full Adders required to perform the addition of $5_{10} + 3_{10}$?

Since 5_{10} in binary would be represented by 101_2, and 3_{10} in binary would be represented by 011_2, and since a Full Adder (according to my design) can add only one column at a time, <u>three</u> Full Adders would be needed to add $5_{10} + 3_{10}$. (Since the carry bit is simply set as the final bit, whether one (1) or zero (0), there is no need for a fourth Full Adder.)

$$\begin{aligned} \text{Carry} &\dashrightarrow 1110_2 \\ 5_{10} &\dashrightarrow 101_2 \\ 3_{10} &\dashrightarrow \underline{\ 011_2} \\ 5_{10} + 3_{10} = 8_{10} &\dashrightarrow 1000_2 \end{aligned}$$

III. Draw a block-level schematic showing all the Full Adders and connections necessary to perform $5_{10} + 3_{10}$ in binary.

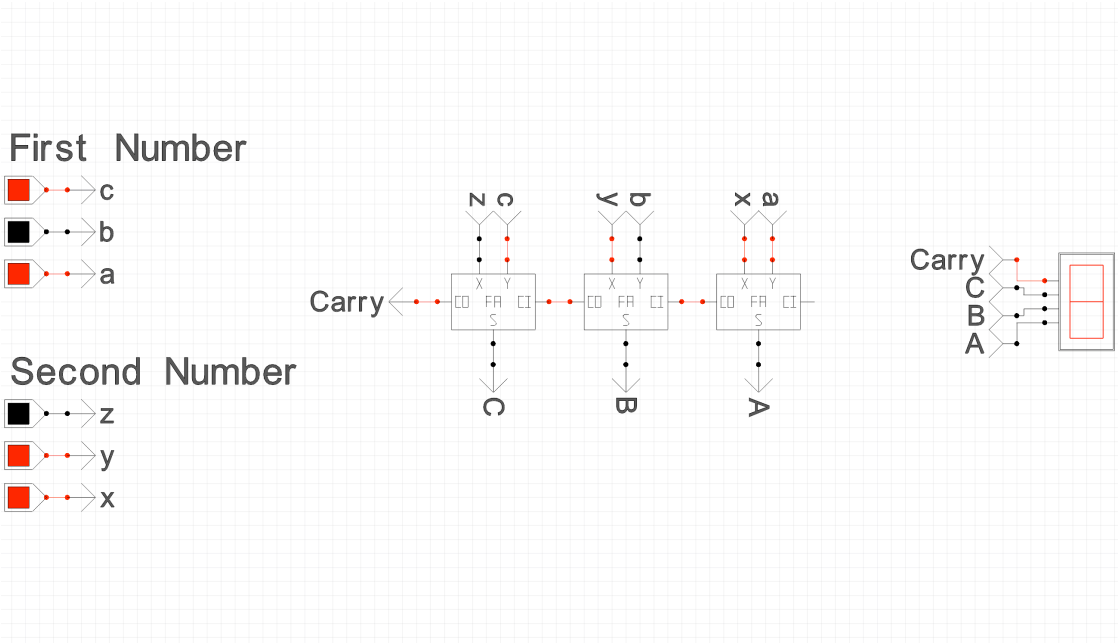

Figure 2: Block-Level Schematic for Full Adder of Three (3) Bit Number