

CDA 3201L - Thursday (3:30 - 4:45PM) Section 005

Lab #03 - Combinational Logic Circuits (3)

10/08/2016 - Steven Romeiro, Jennyfer Munoz

Purpose & Objectives:

To build an arithmetic logic system that can add or subtract two 4-bit numbers. Part A of the lab required us to show how the 74LS83 Adder worked by adding two numbers together. These numbers were assumed unsigned and two's complement. Part B asked us to build a arithmetic system that could perform 4 function calculations with 4-bit numbers. The designs, implementations and their results will be shown below.

Components Used:

Name	Type	Quantity
74LS86	XOR IC	2
74LS04	Inverter IC	1
74LS02	NOR IC	1
74LS83	Adder IC	1
470 Ω Resistor	Resistor	4
LED	Red LED	4
74LS247	BCD Decoder 7-Segment IC	1
LED	7-Segment	1
Power Supply	5v	1
Wire Kit	Assorted	1

Description:

Part A:

Demonstrate the addition operation of two 4-bit numbers:

- Unsigned:
 - $0001 + 0010 = 0011$
 - $0010 + 1001 = 1011$
 - $1101 + 1000 = 0101$ (carry 1)
 - $0110 + 0100 = 1010$
- Two's Complement:
 - $1111 + 1110 = 1101$
 - $1100 + 1110 = 1010$
 - $1001 + 1111 = 1000$
 - $1101 + 1110 = 1011$

Truth Tables:

- Assumes carry in Low

Input A ₁₋₄				Input B ₁₋₄				Sum Output				
A ₄	A ₃	A ₂	A ₁	B ₄	B ₃	B ₂	B ₁	Σ ₄	Σ ₃	Σ ₂	Σ ₁	C ₄
0	0	0	1	0	0	1	0	0	0	1	1	0
0	0	1	0	1	0	0	1	1	0	1	1	0
1	1	0	1	1	0	0	0	0	1	0	1	1
0	1	1	0	0	1	0	0	1	0	1	0	0

- Two's Complement - No Carry

Input A ₁₋₄				Input B ₁₋₄				Sum Output				
A ₄	A ₃	A ₂	A ₁	B ₄	B ₃	B ₂	B ₁	Σ ₄	Σ ₃	Σ ₂	Σ ₁	C ₄
1	1	1	1	1	1	1	0	1	1	0	1	N/A
1	1	0	0	1	1	1	0	1	0	1	0	N/A
1	0	0	1	1	1	1	1	1	0	0	0	N/A
1	1	0	1	1	1	1	0	1	0	1	1	N/A

A	B	Carry
0	0	0
0	1	0
1	0	0
1	1	1

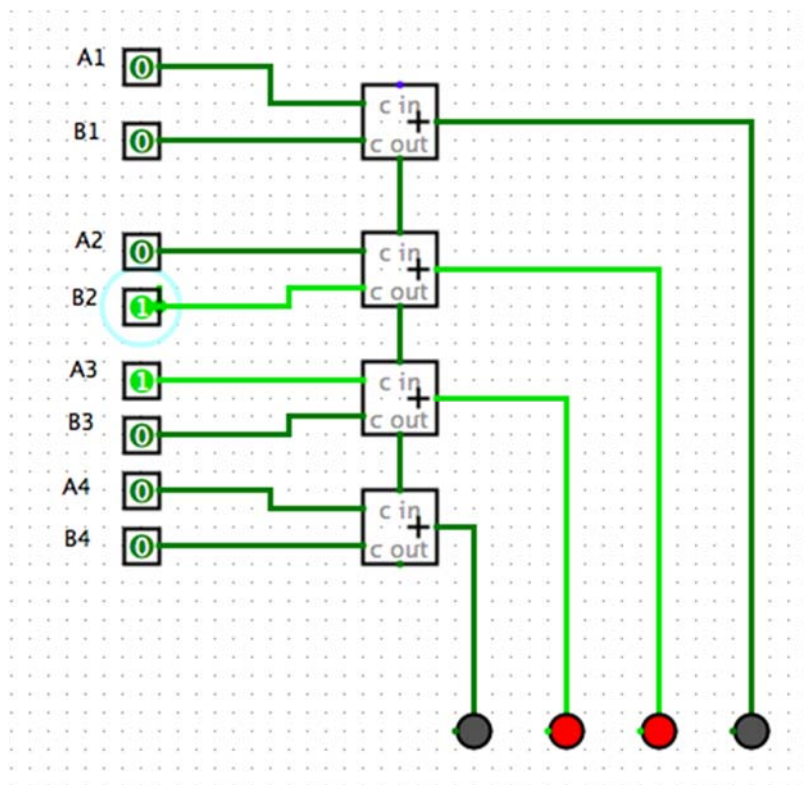
A	B	Σ
0	0	0
0	1	1
1	0	1
1	1	0

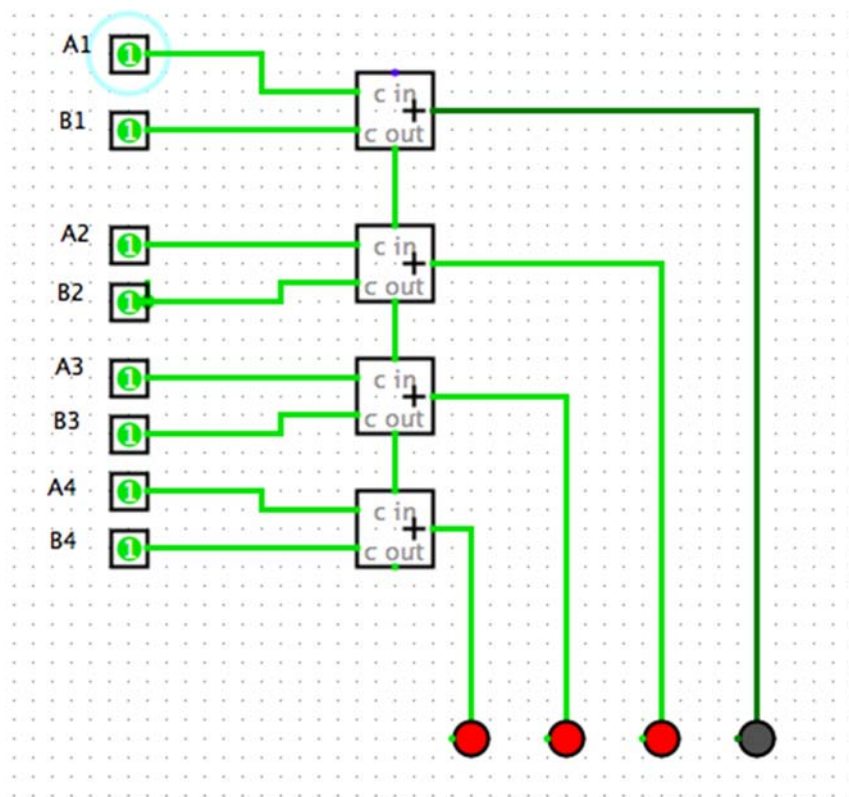
Boolean Expression:

$$\Sigma = A'B + AB' = A \oplus B$$

$$\text{Carry} = AB$$

Diagram:





Part B:

Circuit that implements the functions A+1, A-B, A+B, A-1 using the TTL 7483 4-bit adder.

Truth Table:

- **Out** = Modified **B₁₋₄** inputs on the Adder

B₁₋₄	S₀	S₁	Out
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Kmap:

S_0S_1				
B \	00	01	11	10
0	0	1	1	0
1	0	0	1	1

Boolean Expression: $(B'S_1) + (BS_0)$

- **Out** = **C_{in}** inputs on the adder.

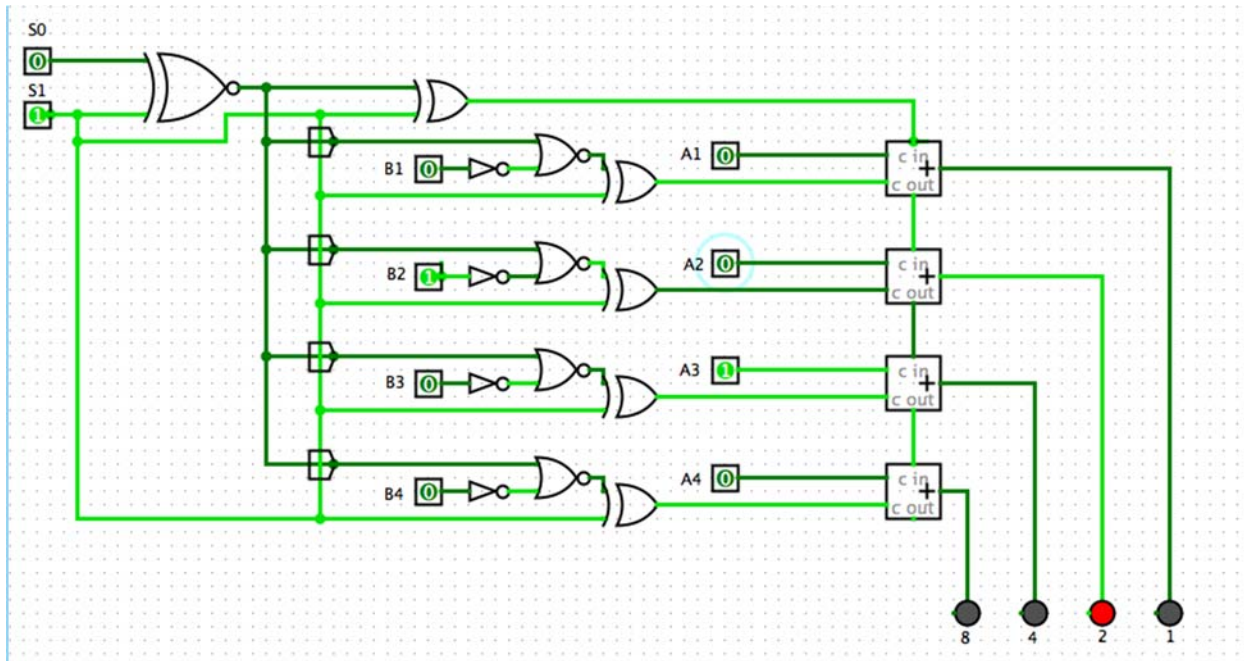
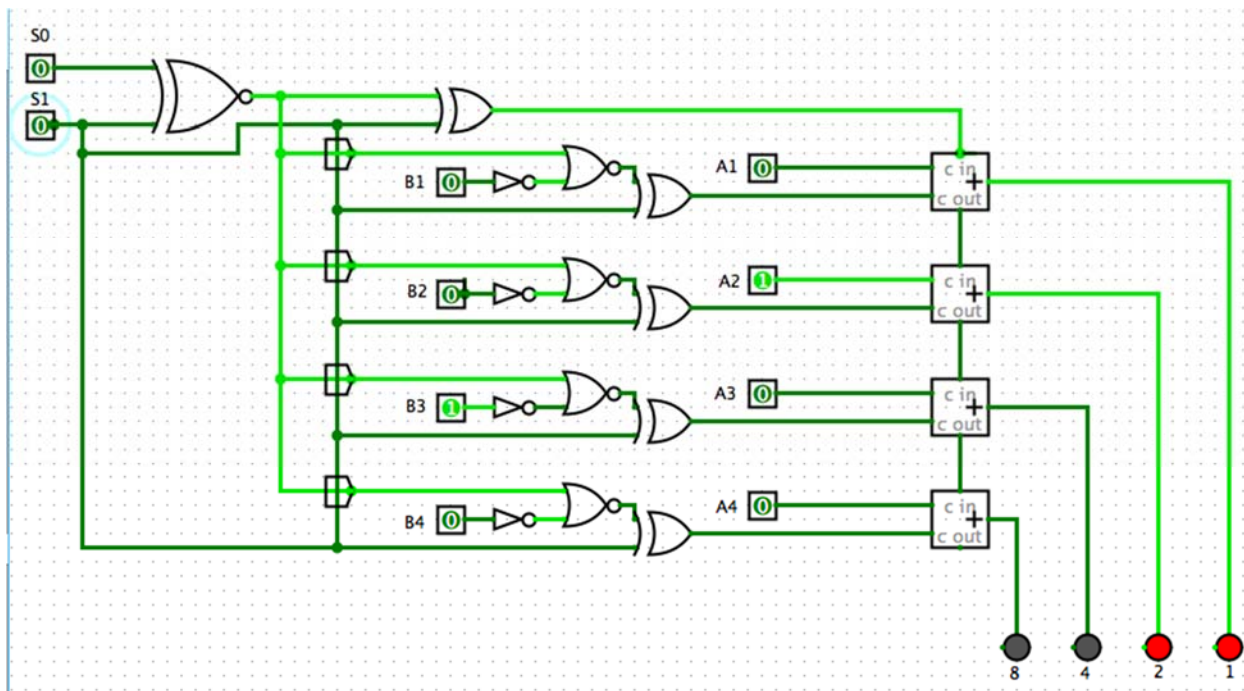
S₀	S₁	Out
0	0	1
0	1	1
1	0	0
1	1	0

Kmap:

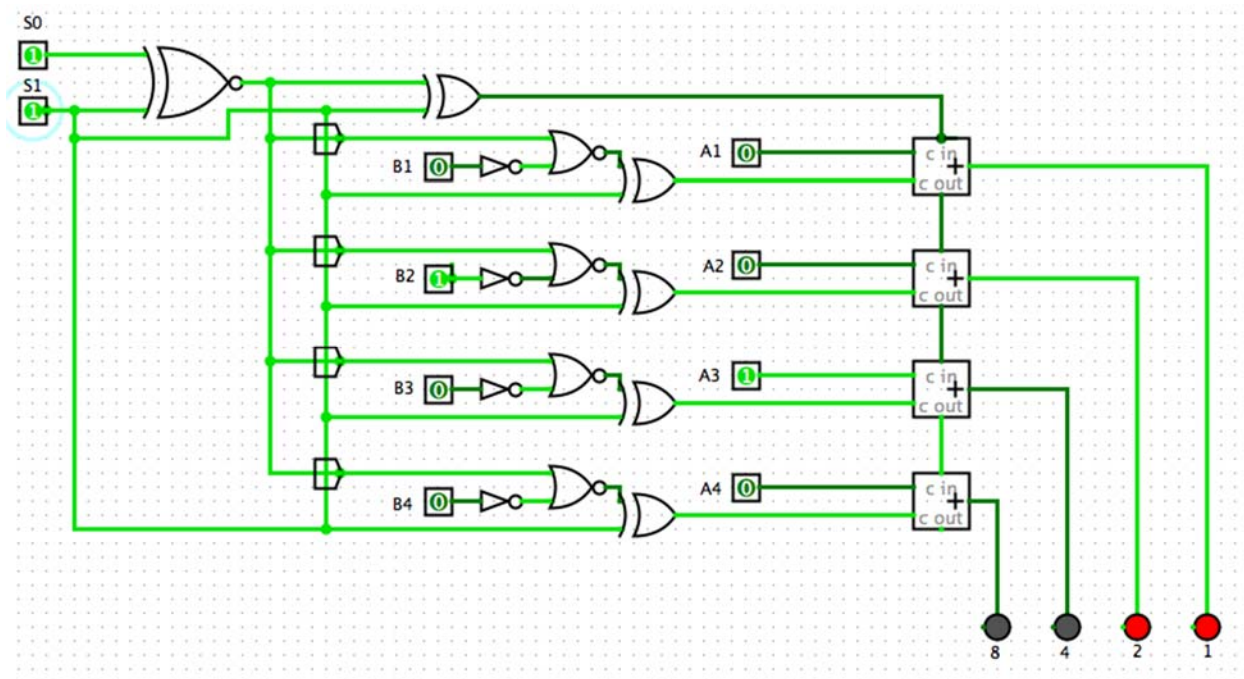
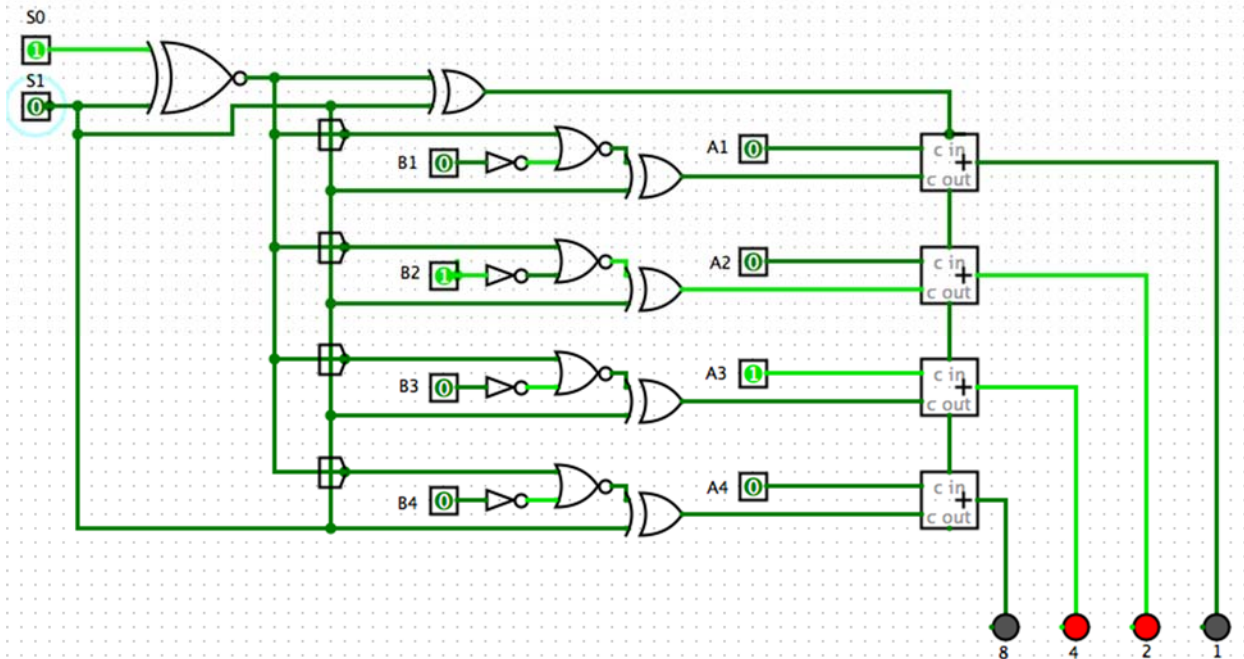
S_0		
S_1 \	0	1
0	1	0
1	1	0

Boolean Expression: S_0

Diagrams:



Continuing Diagrams:



Discussion & Conclusion:

For Part A we displayed our understanding of the Adder IC by expressing the expected outputs as truth tables and boolean expressions. The expected outputs verify with actual results as seen in the simulation diagrams and as displayed in the lab environment. We can clearly see that the

Adder simply does the operations as requested regardless of what binary number representation we use. The evaluation is carried out and it is up to the user to determine the convention being used.

For Part B we developed the arithmetic logic system that carried out the 4 function operations. These functions were selected via control units S_0 and S_1 . We built this logic circuit by tackling each function at a time. Starting with the simpler functions that simply add one and add 'B', we expanded the system as needed to accommodate more functions. After these functions were accommodated and the logic arithmetic circuit could perform the requested operations, we started working on the boolean expressions and truth tables for this design. We followed this approach to see if the system could be developed from scratch through a layering approach rather than starting from expected results and working inwards. The experience of layering a working system was incredibly valuable, it simulated a production environment where the need of the users continually change and the system is modified to keep up with demand. It took creativity and innovation on our part to develop a system through this layered approach. It was satisfying to see how our thinking process and system evolution matched up with the final, expected results.

Questions:

1. Is the output valid for the following input combinations:

a. $S_0 = 0, S_1 = 0, A = 7, B = 3?$

$$(A + 1) : 0111 + 0001 = 1000$$

- 8 in unsigned - valid
- -8 in two's complement - valid

b. $S_0 = 0, S_1 = 1, A = 7, B = 3?$

$$(A - B) : 0111 - 0011 = 0100$$

- 4 in unsigned - valid
- 4 in two's complement - valid

c. $S_0 = 1, S_1 = 0, A = -4, B = -5?$

$$(A + B) : 1100 + 1011 =$$

- Unsigned has no negative number representations.
- Two's Complement this operation is invalid since it's out of range.

d. $S_0 = 1, S_1 = 1, A = -8, B = 6?$

$$(A - 1) : 1000 - 0001 =$$

- Unsigned has no negative number representations.
- In Two's Complement this operation is invalid since it goes out of range

2. What is the range of inputs (for both A and B) that will produce the valid output for all the functions?

If it is an unsigned representation, then the range is from 0 to 15.

If it is in Two's Complement then the range is from -8 to +7.