Combinational Logic Circuits

Object

To investigate the properties of combinational logic circuits, as illustrated by the half adder, full adder, and data selector/multiplexer.

Parts

(1) 7400 Quad 2 input NAND gate

(1) 7408 Quad 2 input AND gate

(2) 7486 Quad 2 input XOR gate

(1) 74151 Eight-line to one-line data selector

(1) 7404 Hex inverter

Study section

Computer Systems, Fourth Edition, Jones and Bartlett Publishers: Section 10.4, Combinational Devices.

General information

Combinational circuits are interconnections of logic elements in which the output at any time reflects the condition of the inputs at that time. Combinational circuits do not retain information as do sequential circuits, and their outputs change with any change in input. One example of the combinational circuit is the binary adder.

Procedure

1. The half adder

The half adder is the simpler of the two types of adders, as it does not use a carry input from a previous stage. The input consists only of the two binary bits to be added. The outputs are the sum of the two bits and the carry, if any, resulting from the addition. As there is no carry input, the half adder may be used only as the least significant adder of an array of adders.

The sum and carry outputs are base on the addition table of Figure 1(a). If the addition table is converted to a truth table, the truth table would appear as in Figure 1(b). Co is the carry out, and S is the sum.

Figure 1

$0 + 0 = 0$
$0 + 1 = 1$
$1 + 0 = 0$
$1 + 1 = 10$

(a) Addition table

Inputs		Co	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

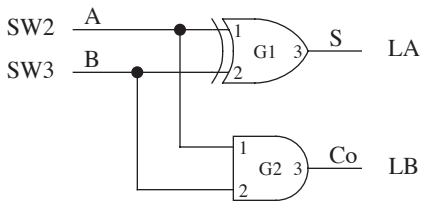
(b) Truth table

Note that the sum outputs of the truth table correspond to those of the XOR gate, and that the carry outputs correspond to those of the AND gate. Thus the half adder may be implemented with only two gates, as shown in Figure 2. In the figure, G1 stands for a gate in IC number 1 (a 7486), and G2 stands for a gate in IC number 2 (a 7408).

Construct the circuit as shown in Figure 2. Set SW2 and SW3 to the positions indicated in the truth table and record the light indications. DO NOT tear down the circuit after you have finished. You will need it for the circuit of Part 3.

Figure 2

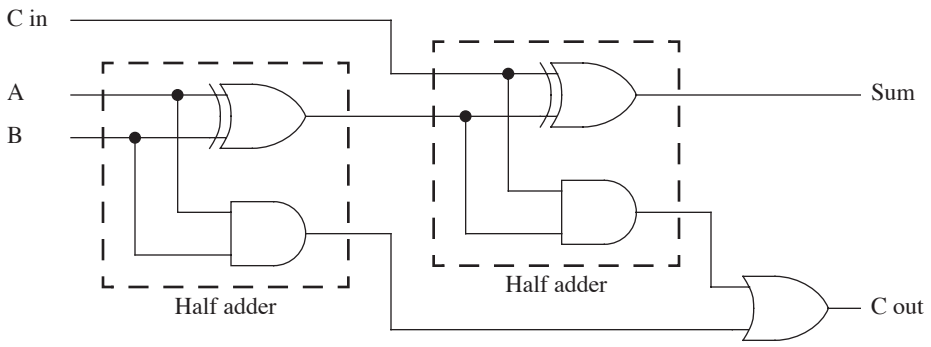
A	B	Co	S
0	0		
0	1		
1	0		
1	1		



2. The full adder

The full adder of Figure 3 differs from the half adder in that it has an additional input for the carry-in term from a previous stage. As shown in Figure 3 the implementation of the full adder requires two half adders and an OR gate. The first half adder performs the addition of the input bits, while the second half adder sums the result of the addition and the carry input to produce the sum output. The OR gate produces the carry output from the carries of the two half adders.

Figure 3



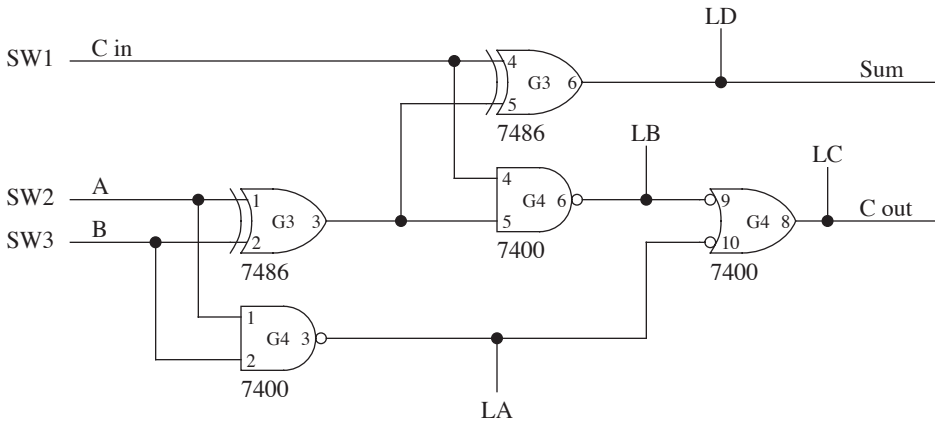
Leaving the circuit you constructed for the half adder on the board for future use, construct the full adder of Figure 4 on the next page. Note that the two XOR gates (G3) should be constructed from 1C number 3, which will be a different chip from G1 that you used in Part 1.

Also note that you must use a single NAND chip (G4) for the AND/OR part of the circuit. The design of the circuit has been altered slightly from that of Figure 3. If you were to build the circuit as shown in Figure 3, you would need three IC's; an XOR, an AND, and an OR. By using the negated-input OR concept which was demonstrated in the previous lab you can reduce the IC count to two—an XOR and a NAND. The entire full adder must be wired from only two chips, G3 and G4.

Set SW1, SW2 and SW3 to the positions indicated in the truth table of Figure 4 and record the light values.

Figure 4

C in	A	B			C out	Sum
SW1	SW2	SW3	LA	LB	LC	LD
0	0	0				
0	0	1				
0	1	0				
0	1	1				
1	0	0				
1	0	1				
1	1	0				
1	1	1				



3. A two-bit adder

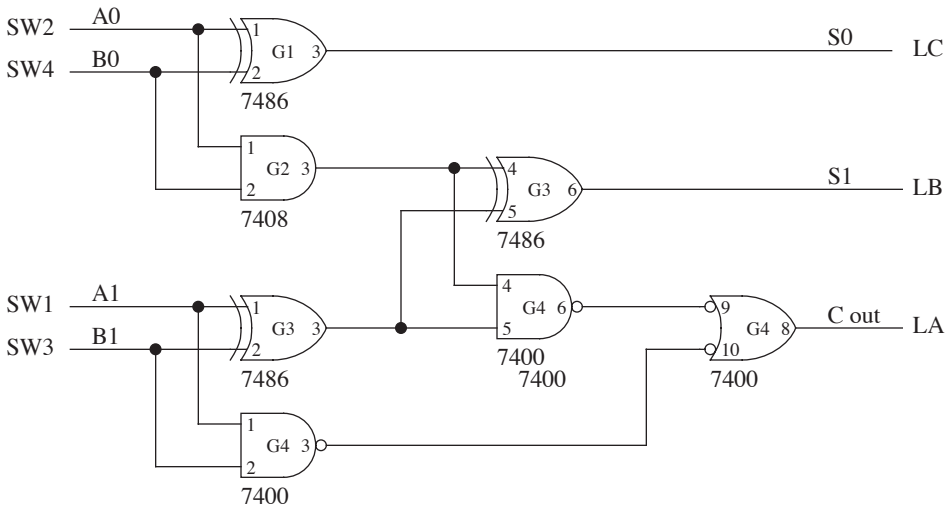
Connect the carry function from the half adder of Part 1 to the full adder of Part 2 to form a two-bit adder array as shown in Figure 5. A0 is the least significant bit (LSB) and A1 is the most significant bit (MSB) of a two bit number. Similarly for the other input B1, B0, and the sum S1, S0. Measure the output and complete the truth table in Figure 5 for this array. Have the lab instructor verify your circuit.

Fill out the table of Figure 5 with the decimal equivalents of the binary numbers as shown. Does the sum reflect the proper addition?

Figure 5

A		B		Sum			Decimal		
A1	A0	B1	B0	C out	S1	S0	A	B	Sum
0	0	0	0						
0	0	0	1						
0	0	1	0						
0	0	1	1						
0	1	0	0						
0	1	0	1						
0	1	1	0						
0	1	1	1						
1	0	0	0						
1	0	0	1						
1	0	1	0						
1	0	1	1						
1	1	0	0						
1	1	0	1						
1	1	1	0						
1	1	1	1						

Instructor verification: _____



4. The data multiplexer as a logic function generator

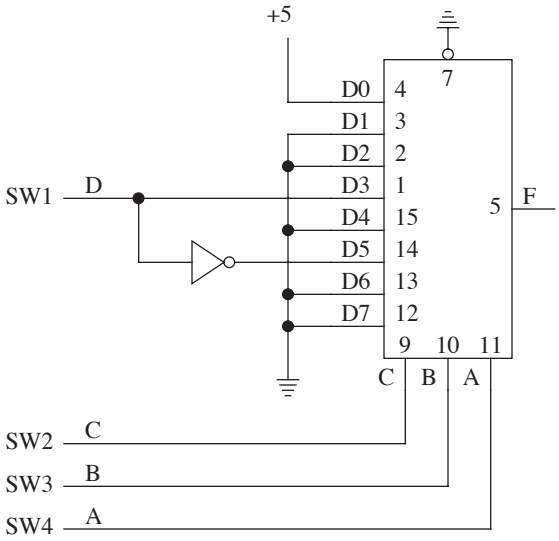
One method of generating various functions of a number of variables uses an *n*-line to 1 line data selector/multiplexer circuit. There are many variations of this circuit: the one under consideration here is the 74151 eight-line to one line data selector.

The specification sheet in your parts list shows the internal gate configuration of the circuit, and gives the truth table. In this application, only the non-inverting (Y) output will be used. The chip enable input (strobe) will be grounded to place the IC in the enabled condition.

An inspection of the truth table reveals that for each possible combination of the ones and zeros at the data select inputs (C, B, A), one and only one of the data input lines will be gated through the output. The particular data line (DO through D7) will correspond to the three-bit binary number applied to the data select lines. A is the low order bit of the select lines and C is the high order bit of the select line. For example, CBA = 001 selects D1 and CBA = 100 selects D4.

If the three data select lines and the data input line associated with a particular combination are considered as four variables, then various four-input logic functions may be created. For example, in Figure 6 the function F = C′B′A′ + DC′BA + D′CB′A′ is implemented. To determine why this function is implemented by the circuit, examine each product term.

Figure 6



$$F = \overline{C}\,\overline{B}\,\overline{A} + D\overline{C}BA + \overline{D}C\overline{B}A$$

If CBA = 000 then C′B′A′ will be 1 and F should be 1 regardless of the value of D. In the circuit CBA = 000 selects input DO. Because DO is tied high, F will be 1 regardless of the value of D.

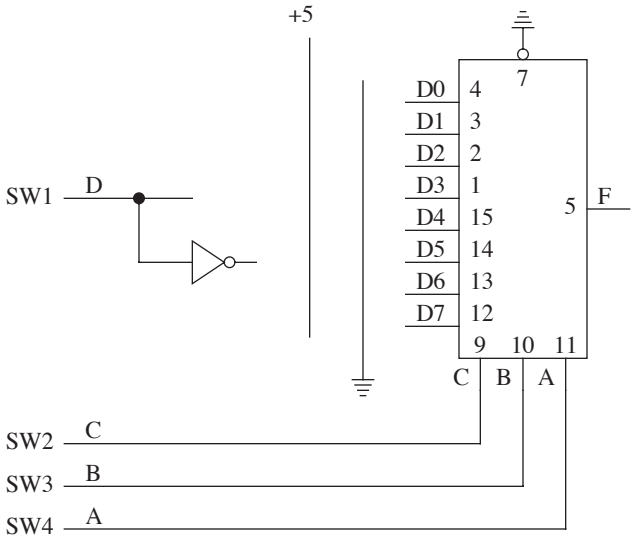
Examining the second term similarly, if CBA = 011 then C′BA will be 1, and F should be 1 if D is 1, and should be 0 if D is 0. That is, if CBA = 011 then F should be D. In the circuit, CBA = 011 selects D3, and D3 is tied to the input D. Therefore the circuit produces 1 if D is 1, and 0 if D is 0, as required.

The product term D′CB′A is analyzed similarly, under the condition that CBA = 101.

Construct the circuit of Figure 6. Manipulate the switches SW1 through SW4 to verify that the function F = C′B′A′ + DC′BA + D′CB′A is being implemented.

Draw the connections and rewire the circuit to produce the functions shown in Figure 7 and Figure 8. For each D input, you must decide whether to connect it to power, ground, D, or D′ to implement the function. Have your lab instructor verify the operation of your circuits.

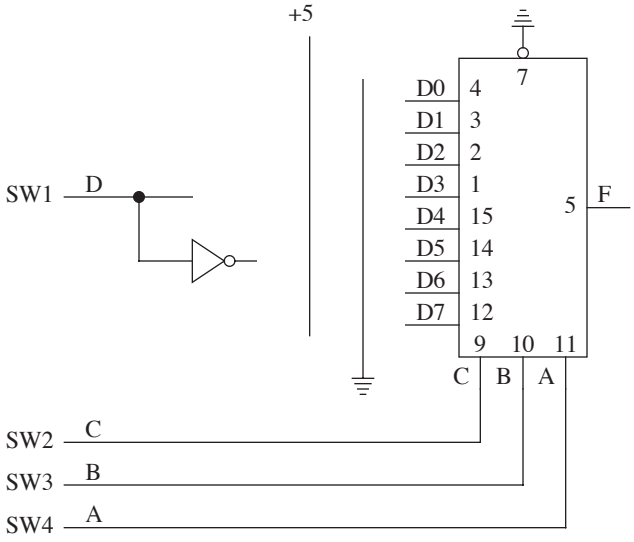
Figure 7



$F = D\overline{C}(B + A)$

Instructor verification: _____

Figure 8



$F = (DC \oplus B)\overline{A}$

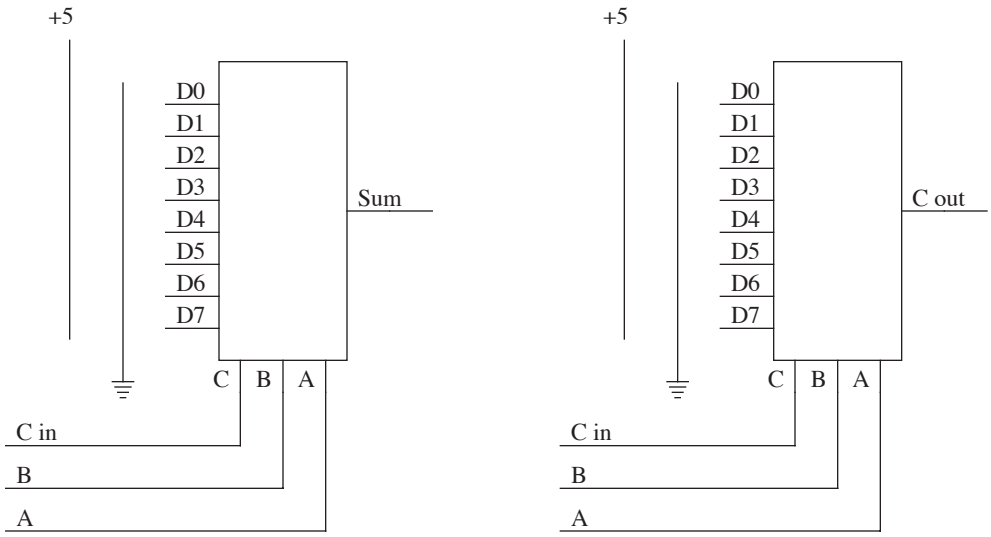
Instructor verification: _____

5. Questions

You may answer lab questions outside of lab, but your answers must be included when you turn in your lab report and will count toward your lab grade.

A full adder could be constructed from two 8-line to 1-line multiplexers. The carry-in and the two adder inputs can be used as inputs to the select lines of the multiplexers. One multiplexer output forms the sum output while the other forms the carry-out. In Figure 9 show what the inputs to the data lines must be to form the correct sum and carry-out functions.

Figure 9



A 4-line to 1-line multiplexer functions like the 8-line to 1-line multiplexer but only two select lines are needed to access data from four lines. In Figure 10 show how a full adder could be constructed from two 4-line to 1-line multiplexers, assuming A and B are the select inputs and C in, C in′, 0V, and +5V are available for the data line inputs.

Figure 10