ECE 301 Digital Electronics

BCD Adders

Revised Fall 2013

K. Hintz & C. Lorie

OBJECTIVE

Understand the design and operation of BCD adder circuits.

PREPARATION

Design and construct the circuit.

PROCEDURE

1. Use a 74LS47 BCD to 7-segment Driver/Decoder to drive a seven segment display.

Note that the data sheet indicates the *BI/RBO* = 1, *LT* = 1 and *RBI* = 1 for proper operation. Verify operation by exercising the inputs for all input combinations.

2. Using two 74LS283 build the circuit in Figure 4-14 below. See also the 74LS283 data sheet.

Hint: Use the binary switches as inputs, L_0 to L_3 as Sum indicators, and the Logic Probe as the carry-out indicator.

3. Connect the output of part 2 to the input of part 1 and verify that the circuit adds correctly.

BCD Adder

Consider the arithmetic addition of two decimal digits in BCD, together with an input carry from a previous stage. Since each input digit does not exceed *9*, the output sum cannot be greater than $9 + 9 + 1 = 19$, the one in the sum being an input carry. Suppose we apply two BCD digits to a four-bit binary adder. The adder will form the sum in binary and produce a result that ranges from *0* through *19*. These binary numbers are listed in Table 4.5 and are labeled by symbols K, Z_8, Z_4, Z_2, and Z_1. K is the carry and the subscripts under the letter Z represent the weights *8, 4, 2,* and *1* that can be assigned to the four bits in the BCD code. The columns under the binary sum list the binary value that appears in the outputs of the four-bit binary adder. The output sum of two decimal digits must be represented in BCD and should appear in the form listed in the columns under *BCD Sum*. The problem is to find a rule by which the binary sum is converted to the correct BCD digit representation of the number in the BCD sum.

In examining the contents of the table, it can be seen that when the binary sum is equal to or less than *1001*, the corresponding BCD number is identical, and therefore no conversion is needed. When the binary sum is greater than *1001*, we obtain an invalid BCD representation.

Table 4.5
Derivation of BCD Adder

Binary Sum					BCD Sum					Decimal
K	Z_8	Z_4	Z_2	Z_1	C	S_8	S_4	S_2	S_1	
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	2
0	0	0	1	1	0	0	0	1	1	3
0	0	1	0	0	0	0	1	0	0	4
0	0	1	0	1	0	0	1	0	1	5
0	0	1	1	0	0	0	1	1	0	6
0	0	1	1	1	0	0	1	1	1	7
0	1	0	0	0	0	1	0	0	0	8
0	1	0	0	1	0	1	0	0	1	9
0	1	0	1	0	1	0	0	0	0	10
0	1	0	1	1	1	0	0	0	1	11
0	1	1	0	0	1	0	0	1	0	12
0	1	1	0	1	1	0	0	1	1	13
0	1	1	1	0	1	0	1	0	0	14
0	1	1	1	1	1	0	1	0	1	15
1	0	0	0	0	1	0	1	1	0	16
1	0	0	0	1	1	0	1	1	1	17
1	0	0	1	0	1	1	0	0	0	18
1	0	0	1	1	1	1	0	0	1	19

The addition of binary $6_{10} = (0110)_2$ to the binary sum converts it to the correct BCD representation and also produces an output carry as required.

The logic circuit that detects the necessary correction can be derived from the entries in the table. It is obvious that a correction is needed when the binary sum has an output carry $K = 1$. The other six combinations from *1010* through *1111* that need a correction have a 1 in position Z_8. To distinguish them from binary *1000* and *1001*, which also have a 1 in position Z_8, we specify further that either Z_4 or Z_2 must have a *1*. The condition for a correction and an output carry can be expressed by the Boolean function

$$C = K + Z_8 Z_4 + Z_8 Z_2$$

When $C = 1$, it is necessary to add *0110* to the binary sum and provide an output carry for the next stage.

A BCD adder that adds two BCD digits and produces a sum digit in BCD is shown in Fig. 4.14. The two decimal digits, together with the input carry, are first added in the top four-bit adder to produce the binary sum. When the output carry is equal to *0*, nothing is added to the binary sum. When it is equal to *1*, binary *0110* is added to the binary sum through the bottom four-bit adder. The output carry generated from the bottom adder can be ignored, since it supplies information

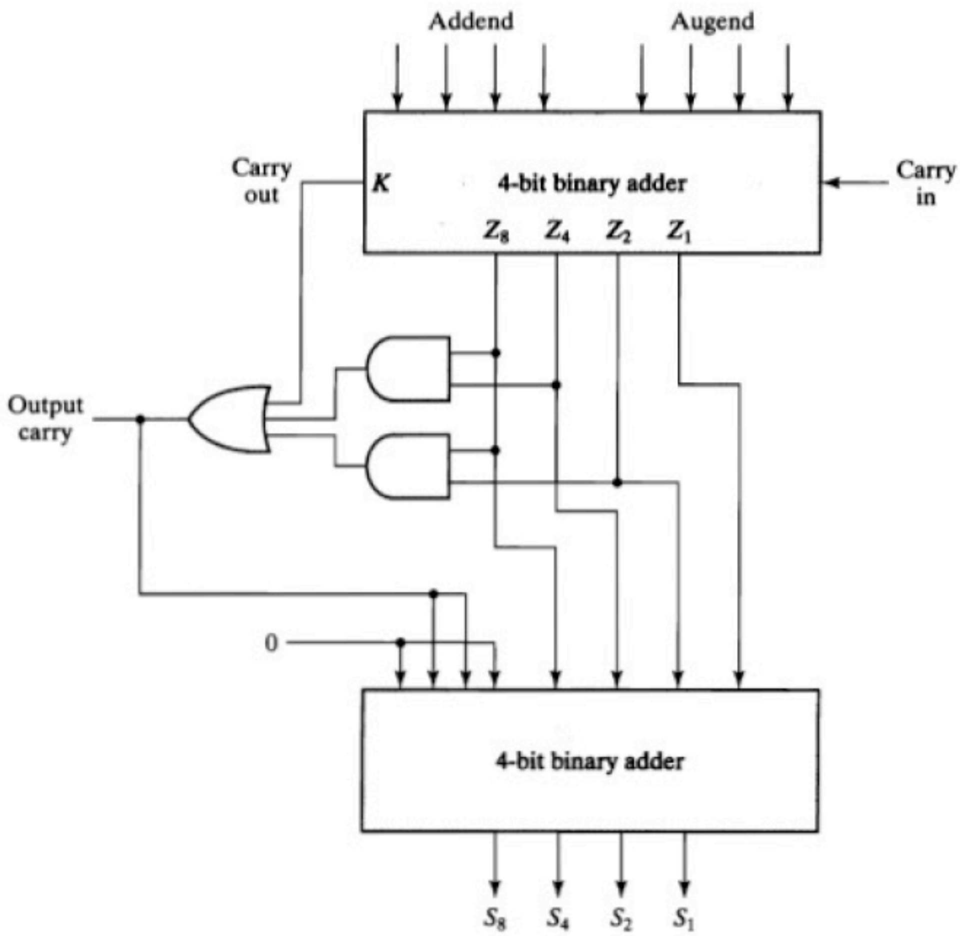

FIGURE 4.14
Block diagram of a BCD adder

already available at the output carry terminal. A decimal parallel adder that adds *n* decimal digits needs *n* BCD adder stages. The output carry from one stage must be connected to the input carry of the next higher order stage.

Seven-Segment Display

A seven-segment indicator is used to display any one of the decimal digits *0* through *9*. Usually, the decimal digit is available in BCD. A BCD-to-seven-segment decoder accepts a decimal digit in BCD and generates the corresponding seven-segment code.

Figure 11.8 shows the connections necessary between the decoder and the display. The 7447 IC is a BCD-to-seven-segment decoder/driver that has four inputs for the BCD digit. Input *D* is the most significant bit and input *A* the least significant bit. The four-bit BCD code is converted to a seven-segment code with outputs *a* through *g*. The outputs of the 7447 are applied to the inputs of the 7730 seven-segment display. This display contains the seven light-emitting diode (LED) segments on top of the package. The input at pin 14 is the common anode (*CA*) for all the LEDs. A 47 Ohm resistor to V_{cc} is needed in order to supply the proper current to the selected LED segments. Other equivalent seven-segment display ICs may have additional anode terminals and may require different resistor values.

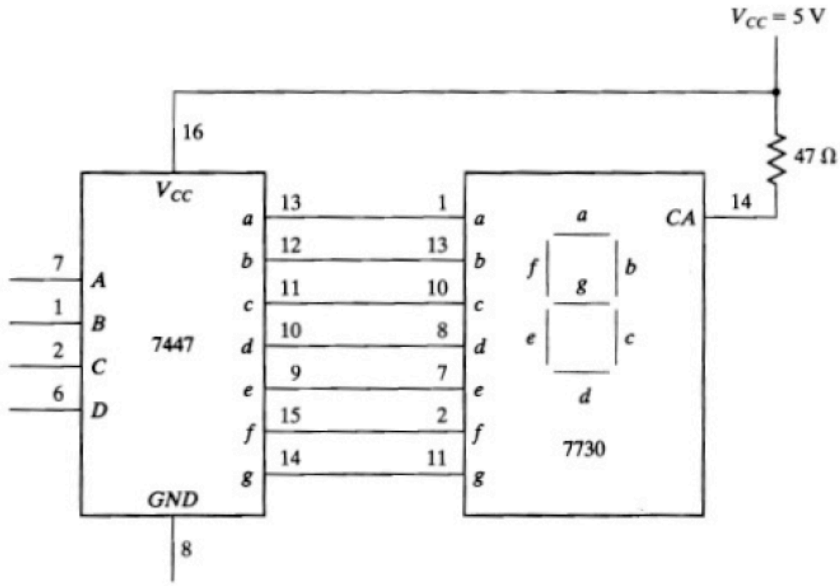

FIGURE 11.8
BCD-to-seven-segment decoder (7447) and seven-segment display (7730)

Construct the circuit shown in Fig.11.8. Apply the four-bit BCD digits through four switches, and observe the decimal display from *0* to *9*. Inputs *1010* through *1111* have no meaning in BCD.

PARALLEL ADDER

IC type 74283 (see data sheet) is a four-bit binary parallel adder. The two, four-bit input binary numbers are A_1 through A_4 and B_1 through B_4. The four-bit sum is obtained from S_1 through S_4. C_0 is the input carry and C_4 the output carry.

Test the four-bit binary adder 74283 by connecting the power supply and ground terminals. Then connect the four *A* inputs to a fixed binary number such as *1001*, and the *B* inputs and the input carry to five toggle switches. The five outputs are applied to indicator lamps. Perform the addition of a few binary numbers and check that the output sum and output carry give the proper values. Show that when the input carry is equal to *1*, it adds *1* to the output sum.