Half-Adder and Full-Adder Circuit

OBJECTIVE

Understanding the characteristics of half-adder and full-adder in the arithmetic unit.

Summary

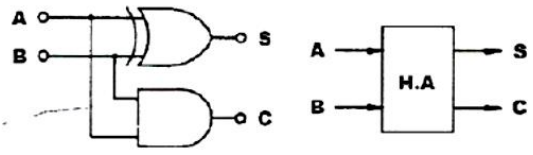
Adders can be divided into "Half-Adder" (HA) and "Full-Adder" (FA). Half-adders follow the rules of binary addition and consider only the addition of 1 bit. The result of addition is a "carry" and a "sum". In binary additions, a "carry" is generated when the sum of two numbers are greater than 1.Refer to the half-adder addition below :

```
   1                      1   ← Previous Carry
 + 1                    1 0   ← Augent
  10                  + 1 0   ← Addend
                        1 0 0
Carry ↑ ↑ Sum       Carry ↑   ↑ Sum
```

When "1" and "1" are added the sum is 0 and the carry is 1. The half-adder is limited to the addition of 1-bit numbers.

The full-adder can perform additions of numbers greater than 2-bits in length. Refer to the full-adder operation shown below. It can be constructed using two half-adder. Fig. 2-37 (a) and (b) shows half-adder and full-adder circuits and symbols respectively.

(a) Half-adder



(b) Full-adder

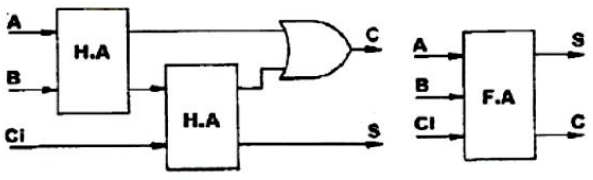


Fig. 2-37 Half-adder/Full-adder

To perform additions of numbers greater than 2-bits in length, the connection shown in Fig. 2-38, or "Parallel Input" should be used to generate sums simultaneously.

However, the sum of the next adder will be stable only after the previous adder's carry has stablized. For example, in Fig. 2-38, the sum of FA2 will not be stable unless the carry of FA1 is stable.

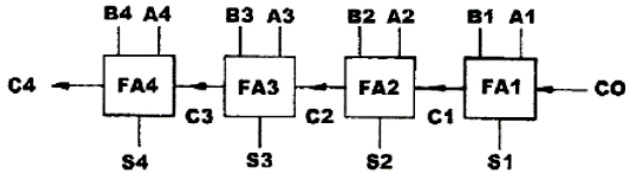


Fig. 2-38

When FA1 adds A1 and B1, a sum S1 and a carry C1 is generated. C1 will be added to A2 and B2 by FA2, generating another sum S2 and another carry C2. In the case of Fig. 2-38,sum of the four adders do not stablize at the same time, dalaying the adding process. This delay can be eliminated by using the "Look-Ahead" adder.
Look-ahead adders do not have to wait for the previous adder to stablize before performing the next addition, saving valuable time. In Boolean expression we assume :

$$Pi = Ai \oplus Bi$$
$$Gi = Ai \times Bi$$

The output and carry can be expressed as :

$$Si = Pi \oplus Ci$$
$$Ci+1 = Gi + PiCi$$

Gi is called "Carry Generate". When Ai and Bi are both "1", Gi is "1" and unrelated to the carry input.

Pi is called "Carry Transmit", related to the carry transmit between Ci and Ci+1.

If we substitute the carry function of each stage by the previous carry we get :

$$C2 = G1 + P1\,C1$$
$$C3 = G2 + P2\,C2 = G2 + P2\,G1 + P2\,P1\,C1$$
$$C4 = G3 + P3\,C3 = G3 + P3\,P2\,G1 + P3\,P2\,P1\,C1$$

EQUIPMENTS REQUIRED

KL-31001 Digital Logic Lab, Module KL-33003/KL-33004

PROCEDURES

(a) Constructing HA with Basic Logic Gates

1. Insert connection clips according to Fig. 2-41, using U2a and U3a to assemble the half-adder circuit of Fig. 2-42. Connect Vcc to +5V.

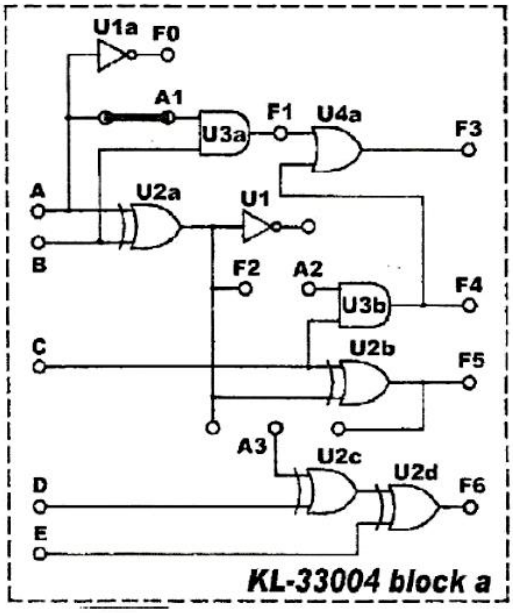


Fig. 2-41

2. Connect inputs A and B to Data Switches SW0 and SW1.Connect outputs F1 and F2 to Logic Indicator L1 and L2. Follow the input sequences for A and B in Table 2-16 and record the output states. Determine which output is the sum and which is the carry

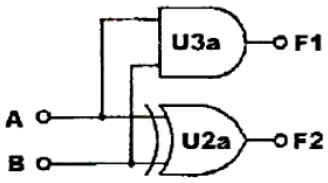


Fig. 2-42

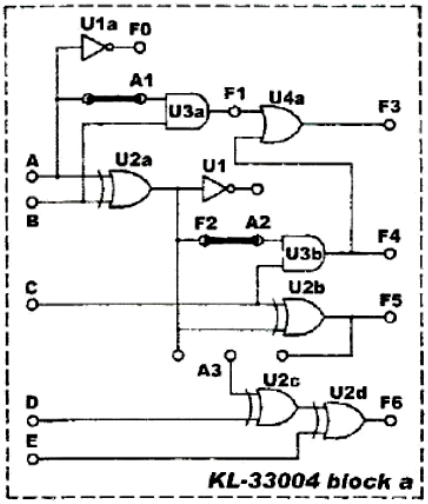
INPUT		OUT	
SW1(B)	SW0(A)	F1	F2
0	0		
0	1		
1	0		
1	1		

Table 2-16

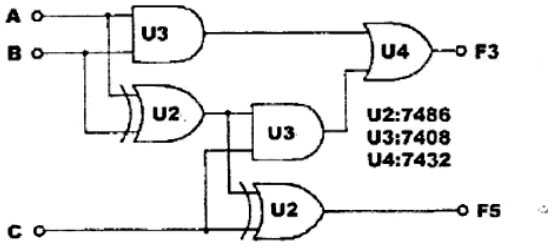
3. Reassemble the circuit according to Fig. 2-42 (a) to construct the full-adder circuit shown in Fig. 2-42 (b).
Connect A, B, C to SW1, SW2 and SW3. A and B are augends while C is the previous carry. Connect F3 to L1, F5 to L2. Follow the input sequences in Table 2-17 and record output states. Determine which output is the sum and which is the carry.

OUTPUT			OUT	
SW3(C)	SW2(B)	SW1(A)	F3	F5
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Table 2-17



(a)



(b)

Fig. 2-43 Full-adder circuit

(b) Full-Adder Circuit with IC

1. U5 on block b of module KL-33004 is used as a 4-bit adder. Connect input Y5 to "0", so the XOR gates U6a~U6d, which are connected to Y0~Y3, will act as buffers.
   Connect inputs X0~X3 (addends), Y0~Y3 (augends) to DIP Switches DIP2.0~2.3 and DIP1.0~1.3 respectively. Connect F1, Σ0, Σ1, Σ2, Σ3 to L1~L5. Follow input sequences in Table 2-18, record F1 and Σ in hexadecimal numbers.
   (X and Y can also be connected to the Thumbwheel Switches)

   X = X3 X2 X1 X0
   Y = Y3 Y2 Y1 Y0
   Σ = Σ3 Σ2 Σ1 Σ0

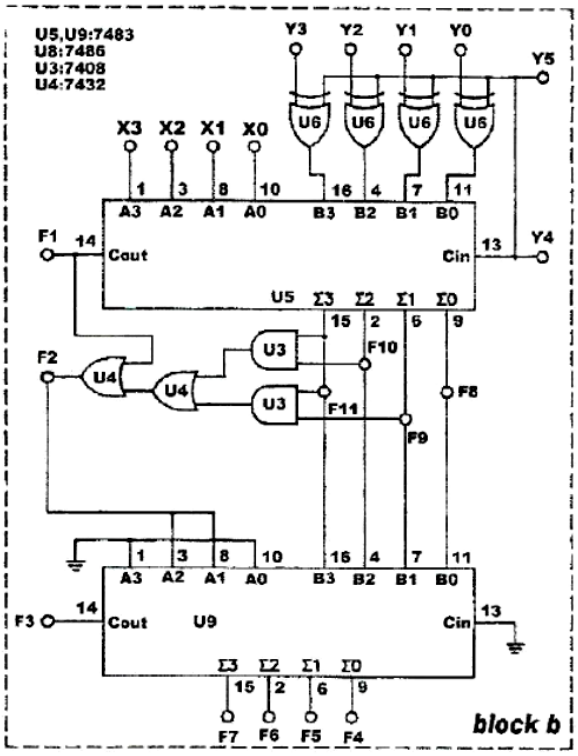


Fig. 2-44

INPUT		OUTPUT	
Y	X	Σ	F1(CARRY)
0	0		
0	1		
0	6		
0	9		
0	F		
1	3		
1	6		
1	8		
3	6		
4	8		
4	F		
8	7		
9	9		
A	B		
C	E		
F	F		

Table 2-18

(d) BCD Code Adder Circuit

1. The circuit shown in Fig. 2-46 will act as a BCD code adder.

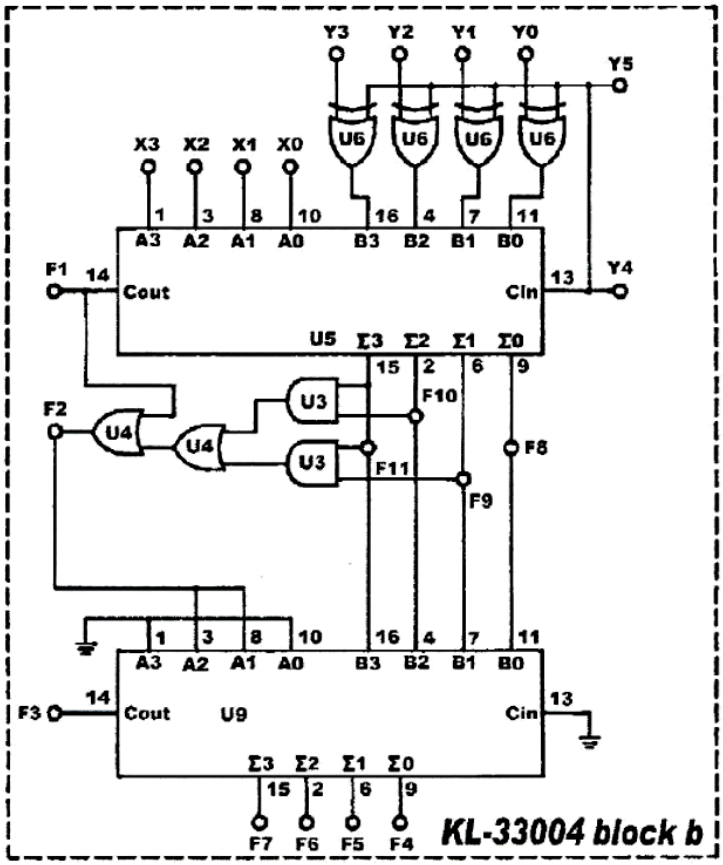


Fig. 2-46

2. Connect inputs X0~X3 to DIP1.0~1.3; Y0~Y3 to DIP2.0~2.3; Y5 to "0". Fig. 2-47 is the equivalent circuit.

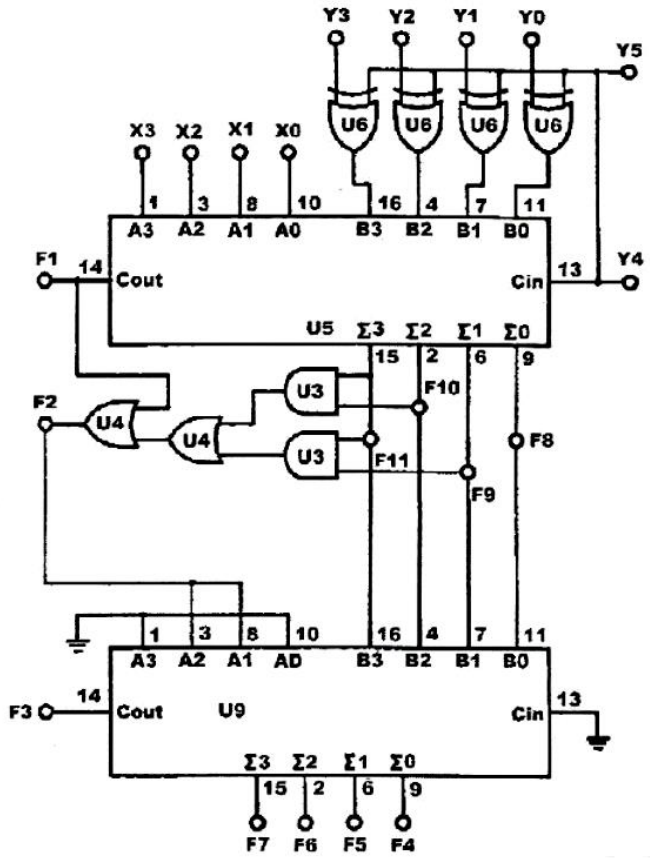


Fig. 2-47

U5 and U9 are 7483 look-ahead 4-bit BCD adders, connect outputs F8~F11 of U5 the inputs of one of the 7-Segment Digital Display. F8~F11 should also be connected to L1~L4. Connect F1, F2 to Logic Indicators L5, L6.

Connect outputs F4~F7 of U9 to another 7-segment display. Also connect F4~F7 to L7~L10 and F3 to L11.

3. F11~F8 are the sum of X0~X3 added to Y0~Y3 while F1 is the carry. Follow the input sequences for X0~X3 and Y0~Y3 in Table 2-20 and record the output tates.

INPUT								OUTPUT(U5)					LAST(U9)					
X3	X2	X1	X0	Y3	Y2	Y1	Y0	F1	F11	F10	F9	F8	F2	F3	F7	F6	F5	F4
0	0	0	0	0	0	0	0											
0	0	0	1	0	0	1	1											
0	0	1	1	0	1	0	0											
0	0	1	0	0	0	1	0											
0	0	1	0	1	0	0	0											
0	0	1	1	0	1	1	0											
0	1	0	0	0	0	1	0											
0	1	0	0	0	1	0	1											
0	1	0	0	0	1	1	0											
0	1	0	1	0	1	1	0											
0	1	1	0	0	1	1	1											
0	1	1	1	1	0	0	0											
0	1	1	1	1	0	0	1											
1	0	0	0	1	0	0	1											
1	0	0	1	1	0	0	1											
1	0	1	0	1	0	1	0											
1	0	1	0	1	0	1	1											
1	0	1	0	1	1	0	0											
1	0	1	1	1	1	1	0											
1	1	1	1	1	1	1	1											

Table 2-20

DISCUSSION:

1- Describe the difference between a half-adder and a full-adder?

2- Show how two 7483 a four-bit parallel adders can be connected to form an eight-bit parallel adder.
Show output form
$P_7P_6P_5P_4P_3P_2P_1P_0$=10111001 and $Q_7Q_6Q_5Q_4Q_3Q_2Q_1Q_0$=10011110

3- Show to connect two 7482 adders to form a four-bit adder?

4- Draw a block diagram for 8-bit parallel adder?

5- Determine an alternative method for implement the full-adder.
Hint: Write the expressions of the circuit and simplify using icarnaugh map.Then implement using AND-OR gates.

6- Design a logic cct using NAND gate and convert BCD code to Excess-3code.