3.3.2. Flow Chart

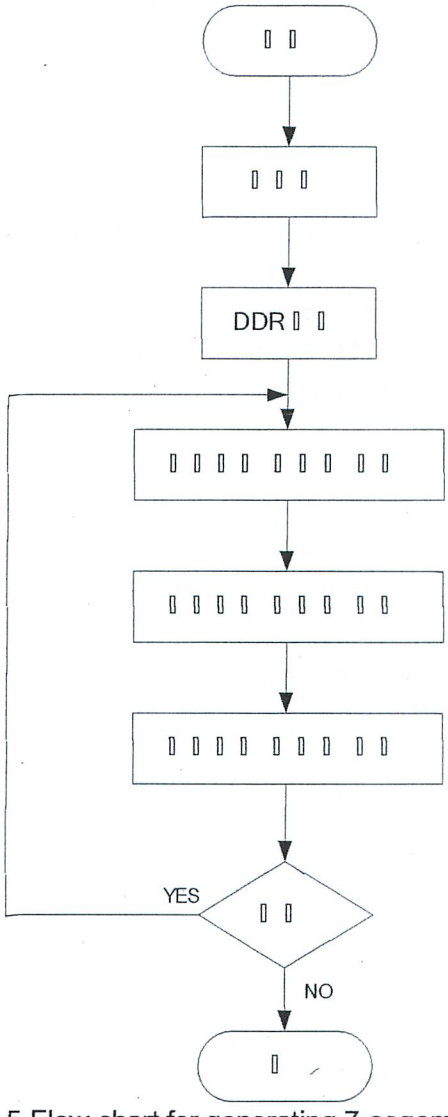


Figure 3.3.5 Flow chart for generating 7-segements data

Let's examine the method of using 7-segment in MA-VIN through the following simple program.

3.3.3. Example Program 1

```
1 :  #include "newbeetle.h"
2 :  void main(void)
3 :  {
4 :      Initialize();
5 :      u08 i,data;
6 :      char   segment[16]={0x48,0xeb,0x8c,0x89,0x2b,0x19,0x18,
7 :                          0x4b,0x08,0x09,0x0a,0x38,0x5c,0xa8,0x1c,0x1e};
8 :  //-----fnd port IO setup for 7segment operation--------
9 :      sbi(DDRB , 0); sbi(DDRB , 1);      sbi(DDRB , 2);
10 : //----------래치를 0으로 설정----------
11 :     cbi(PORTB,2);
12 :     while(1)
13 :     {
14 : //----------데이타 전송----------
15 :         data=segment[7];
16 :         for(i=0;i<8;i++){
17 :             if((data<<i) & 0x80){
18 :             sbi(PORTB,0);
19 :             }else{
20 :             cbi(PORTB,0);
21 :             }
22 : //----------클럭 생성----------
23 :         sbi(PORTB,1);
24 :         cbi(PORTB,1);
25 :         }
26 : //---------래치를 1로 설정----------
27 :         sbi(PORTB,2);
28 :     }
29 : }
```

The above example displays the number '7' on 7-segment.

Line 6: it is the array of data to represent 0 to F, which is the output of 7-segment. It is a value to display on the segment after selecting the desired output data. Since this value can be changed according to the circuit design, keep in mind that you won't be able to obtain the desired output in MA-VIN if the values are changed voluntarily.

Line 7: assumes that 7-segment module is connected to slot #1 in the program. Thus, 'Data Direction Register' becomes 'B'. Since 'Data Direction Register Bit' must be HIGH to turn 7-segment on, the output is set up by 'sbi' command.

Line 10: to understand the statement, "Latch is set to 0", understanding the operation of 74HC595 is necessary. This can be verified in the function table of 74HC595 (Table 3.3.1). Figure 3.3.6 shows the logic diagram of 74HC595, which will help you the understanding of function table.

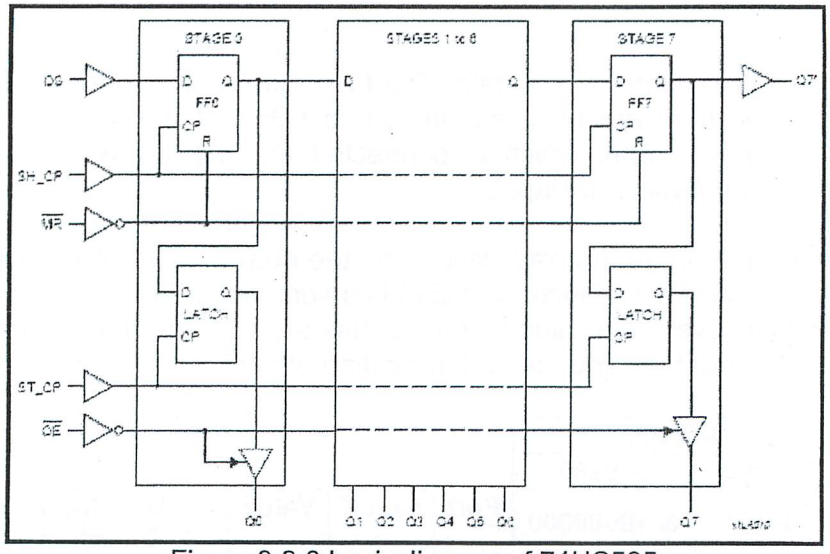
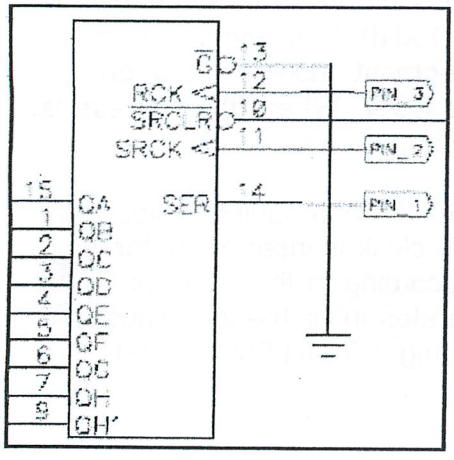


Figure 3.3.6 Logic diagram of 74HC595

As shown in Figure 3.3.6, 74HC595 uses two clocks: 'SH_CP' and 'ST_CP'. As described in the pin description of 74HC595 (Table 3.3.2), 'SH_CP' and 'ST_CP' are used as the register clock of 74HC595.

11	SH_CP	shift register clock input
12	ST_CP	storage register clock input

Since these two clocks operate when they are high-edge as shown in the function table, they act as a clock when the initial value is set to '0' and changes to '1' afterward. In that case, let's examine the circuit in MA-VIN now.



In MA-VIN, PIN1 to PIN3 are connected to 'SER', 'SRCK', and 'RCK'. You have confirmed that 'RCK' and 'SRCK' in the circuit (Figure 3.3.4) act as a clock. If you verify the PIN number, PIN2 and PIN3 are used as 'SH_CP' and 'ST_CP', respectively.

Since this corresponds to the port 1 and port 2 (assuming it is connected to slot #1), the port value, which will be used as a clock, can be adjusted by changing the setting as below.

74HC595 uses two clocks. First of all, the reason for setting the clock of latch can be understood by looking at the logic diagram. When the DS input enters into the flip-flop, the output value of flip-flop goes to the input of latch according to 'SH_CP'. Then the final output value comes out according to the operation of 'ST_CP'. Since the outcome is obtained according to 'ST_CP', the value of 'ST_CP' is made to 0.

Line 14: next is the transmission of data. The total number of inputs of 74HC595 is three: two of them are used as clock, and substantial input that is used to represent data is one, which is connected to 'PIN 1'. Keep in mind that 74HC595 is an 8-bit serial input.

Data uses the value in array, which has the output value of segments. Let's examine 'for' loop statement. Since 74HC595 is a serial input, the value of data must be stored in consecutive order. Since the input has an 8-bit, the data is entered over 8 times. Table 3.3.3 will help you the understanding of data input.

	Number					
	7(0x4b)	& 0x80				
i	01001011	& 10000000	Port	On/off	Value	Data Input Direction
0	01001011	00000000	sbi	Off	0	↓
1	10010110	10000000	cbi	On	1	
2	00101100	00000000	sbi	Off	0	
3	01011000	00000000	sbi	Off	0	
4	10110000	10000000	cbi	On	1	
5	01100000	00000000	sbi	Off	0	
6	11000000	10000000	cbi	On	1	
7	10000000	10000000	cbi	On	1	

Table 3.3.3 Example of 74HC595 data input

Table 3.3.3 is an example of entering the number '7'. Remember that '7' is treated as a character instead of a constant. The value related to '7' is represented as '0x4b', which can be verified in the array. '0x4b' has the value of '01001011' in binary representation. In Table 3.3.3, the value of '01001011' is entered in order. Since the input voltage, Vcc, is supplied to the segment module, LED on the segment module turns on when the value of port is LOW. When this process is repeated, entering a serial 8-bit data becomes practicable.

Line 22: we have examined that two clocks operate in 74HC595 earlier. A clock that is setting now is the clock of flip-flop. When this clock is inserted in 'for' loop statement, the data can be entered in order according to the value of 'i'. As mentioned in the function table, 74HC595 operates in high-edge. Thus, the clock starts to operate when the value of port changes from LOW to HIGH.

sbi(PORTB,2);

"Setting a latch to '1'" has the identical reasoning of "setting a latch to '0'" before. Since the final output appears when the output of flip-flop enters to the latch,

the clock starts to operate through 'for' loop statement and the input value comes to output in parallel to operate 7-segment.