SGS-THOMSON MICROELECTRONICS

7 SEGMENT DISPLAY DRIVE USING THE ST6-REALIZER

by Olivier Rouy

INTRODUCTION

Seven segment devices are often preferred in display applications where high luminescence is required, both for indoor and outdoor applications.

Controlling the display is normally but one of many tasks handled by a typical application. The techniques described in this note are of a general nature and may be applied to a variety of applications.

Hardware considerations are reviewed and generation of control software using the ST6-REALIZER is described.

1 PRINCIPLE OF OPERATION

A 7-Segment Display consists of 7 LED's arranged in a figure-eight pattern, such that by selectively powering various combinations of segments alphanumeric characters may be displayed; a further LED is present which, when powered, causes a dot or decimal point to be displayed (Figure 1). Each LED is turned on by forward biasing its respective anode, all segment cathodes being commoned (Common Anode devices also exist).

The Digital Point LED is generally used to display numbers containing a non-integer part. Table 1 illustrates the required segment patterns for numeric representation, including the optional Decimal Point character.

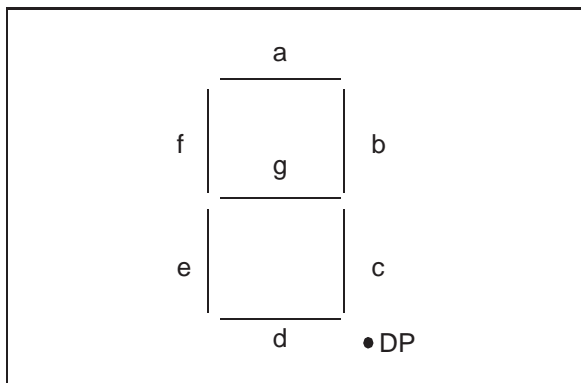**Figure 1. Structure of a 7 segment display**

Table 1. LED's turned on for a given digit

	a	b	c	d	e	f	g	DP
0	1	1	1	1	1	1	0	X
1	0	1	1	0	0	0	0	X
2	1	1	0	1	1	0	1	X
3	1	1	1	1	0	0	1	X
4	0	1	1	0	0	1	1	X
5	1	0	1	1	0	1	1	X
6	1	0	1	1	1	1	1	X
7	1	1	0	0	0	0	0	X
8	1	1	1	1	1	1	1	X
9	1	1	1	1	0	1	1	X
DP	X	X	X	X	X	X	X	1

X = Don't care

Biasing can be either continuous or multiplexed as long as the refresh frequency is high enough to ensure image persistence for the human eye (at least 20 cycles per second to avoid flicker).

SGS-THOMSON
MICROELECTRONICS

2 PRACTICAL IMPLEMENTATION

2.1 Hardware layout

Biasing is achieved using the ST62 MCU's I/O lines. One line is used to bias the common cathode, while 7 lines are assigned to LED segments a - g and a further line, if required, to the DP. Each segment of the display can thus be turned on or off, depending on the value written to the corresponding I/O line.

2.2 Software generation

Control software is automatically generated by the ST6-REALIZER, on the basis of a functional description.

The digit to be displayed is available as a variable, "Digit value". The display control software must then write the relevant value to the I/O lines in accordance with Table 1.

The ST6-REALIZER library provides a component, the *lookuptable,* for this kind of application*.* This table assigns to each variable input value a constant output value defined in the ROM memory of the ST62 MCU. If a lookup table is defined for each LED or segment, the output data is of BIT type as shown in the Figure 2 for segment g.
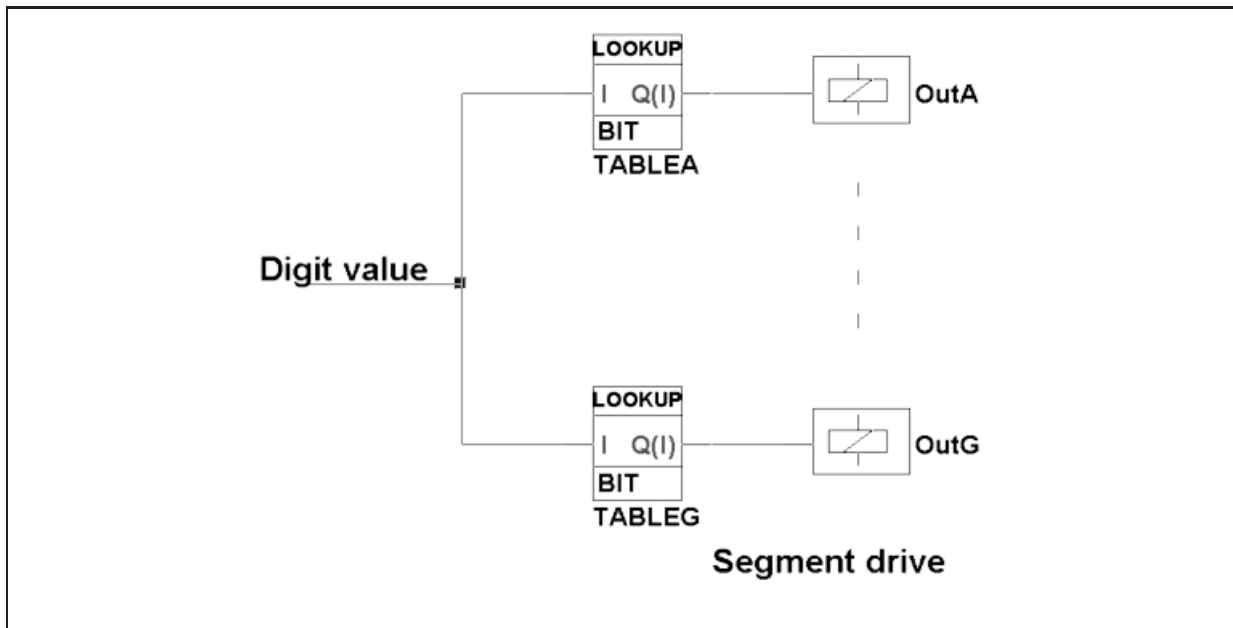
Figure 2. Lookup table for segment g
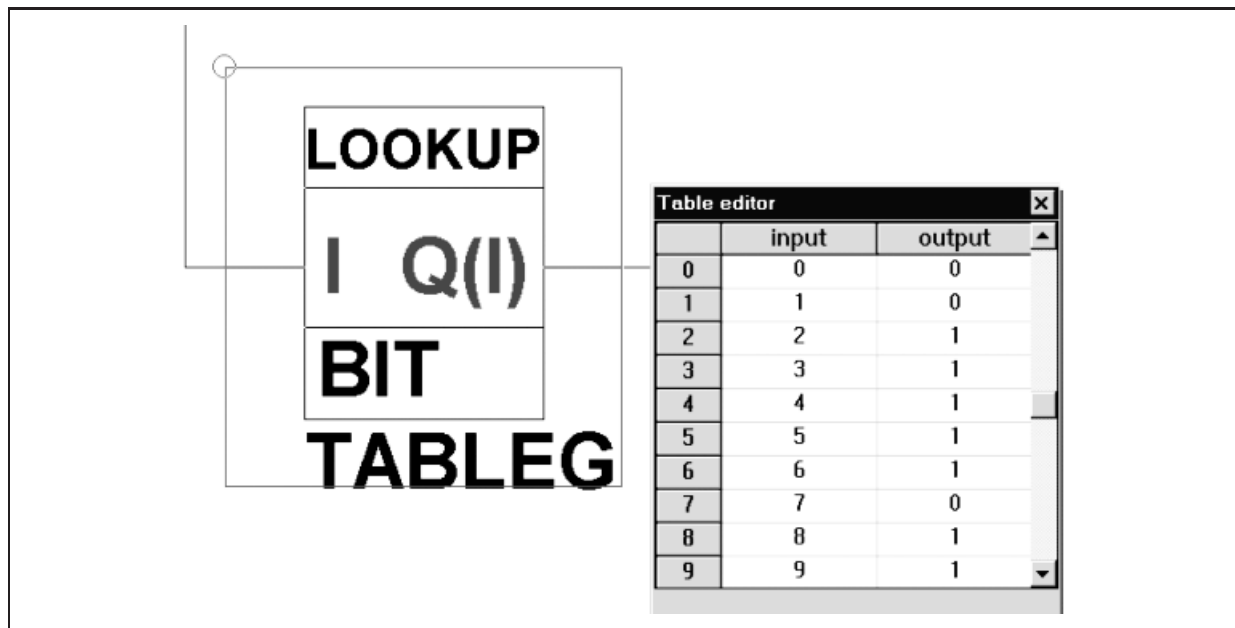
Finally, the resultant bits are transferred to the I/O port via the *digout* components (Figure 3).

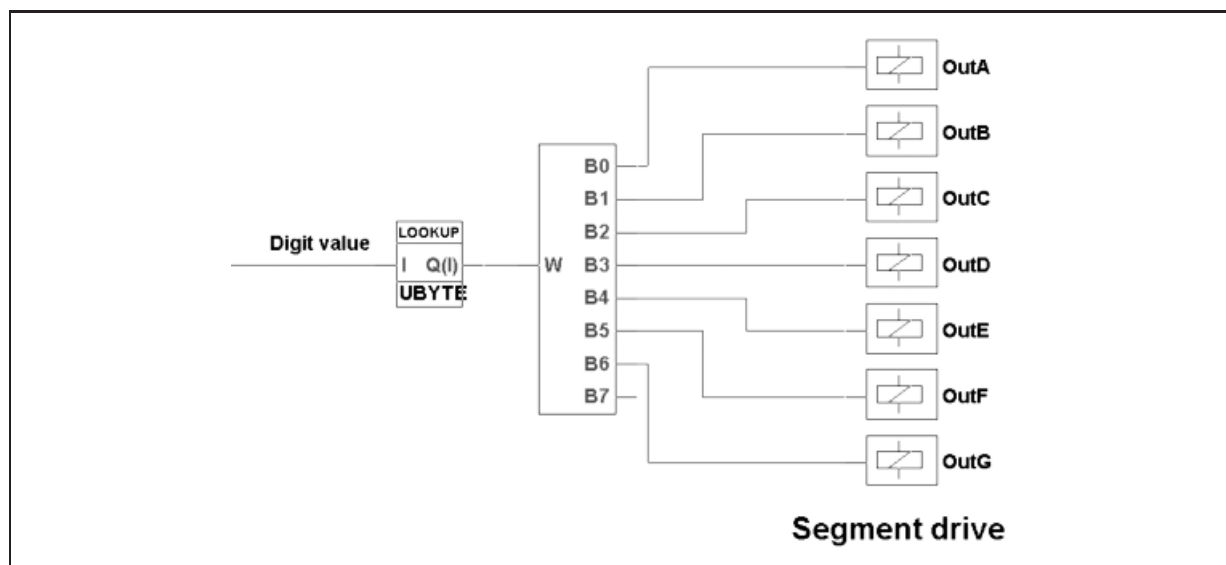**Figure 3. 7-segment drive with bitwise lookup table**

However this approach implies the use of 7 lookup tables (1 for each segment). This means the MCU has to scan a table 7 times, with consequent impact on execution time. A better alternative is to concatenate the 7 segments into a bytewise variable, and therefore use only one lookup table with output data of UBYTE type. In this case, the content of the lookup table is a byte defined as described in Table 2:

Table 2. 7-segment coding in one byte

D7	D6	D5	D4	D3	D2	D1	D0
Don't care	g	f	e	d	c	b	a

The resulting byte is then transferred to 7 *digout* components by using an unpacker component, *bunpack*, and gives the complete description as per Figure 4.

SGS-THOMSON
MICROELECTRONICS

Figure 4. 7-segment drive with bytewise lookup table

By using the simulation features of the ST6-REALIZER, the reduction in execution time has been quantified as follows:

7 bitwise lookup tables: 5170µs

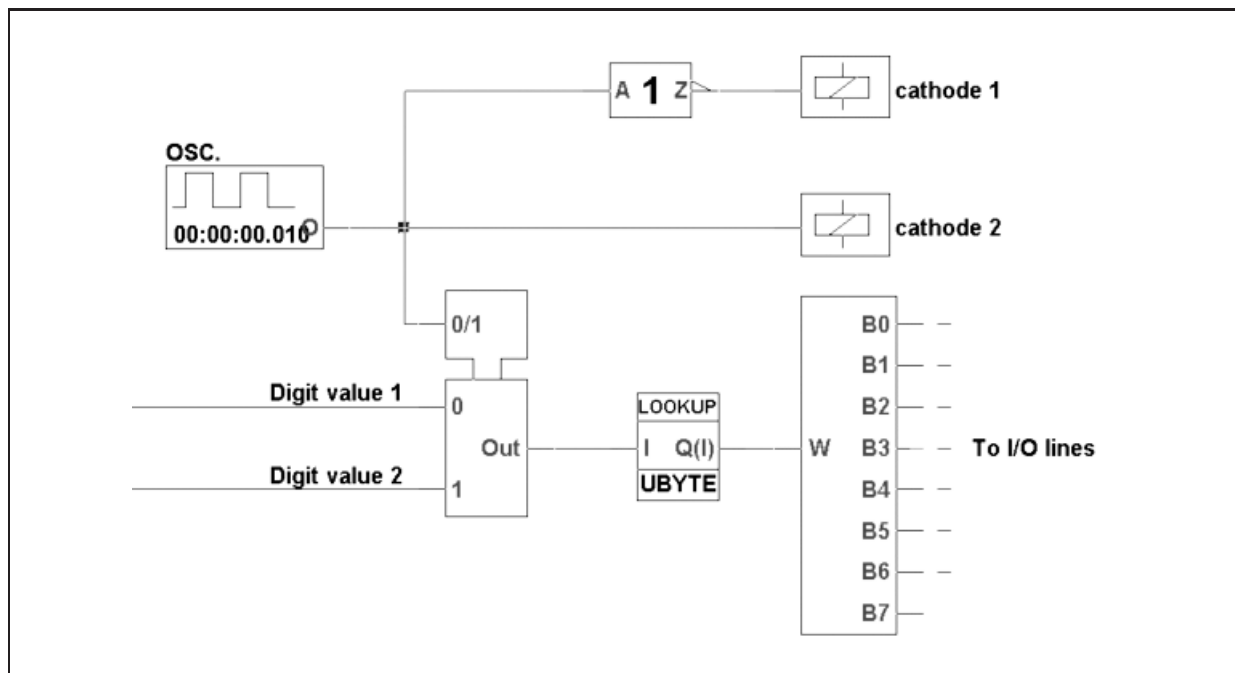1 bytewise lookup table: 1125µs

2.3 Multiple digit display

These concepts can be extended to multiple-digit display systems. The simplest approach is to drive each digit separately, which requires at least 7 I/O lines per digit (excluding DP and assuming the common cathodes are tied to an external supply). It is possible to greatly reduce the number of I/O lines by adopting a multiplexing technique. In this scheme, each of the segments are common for all display digits, while the cathodes are driven sequentially to select the active digit. The 7 I/O lines which drive the segments are then driven in accordance with the currently driven cathode. The lookup table is common to all digits but its input will vary depending on the digit to be displayed.

The input thus depends on the cathode selection signal, which must switch with a period shorter than the persistence time of the human eye in order to avoid flicker. In the present case, 20ms has been chosen. Figure 5 illustrates such an application, where two digits are used to display values ranging from 0 to 99.

Figure 5. Multiple digit display application

SGS-THOMSON
MICROELECTRONICS