Report for the implementation of a 7-segment display

Truth table for the 7-segment display

The following truth table was created using the appendix in the task description. The table consists of the input number and the non-inverted and inverted output as binary and hex representation.

As can be seen in the figure 1, every value in the table 1 is corresponding to the non-inverted and inverted output of the testbench. Therefore, the specification is met.

bin_i, hex	hex_o, binary (Segments: GFEDCBA)	hex_o, hex	hexn_o, binary (Segments: GFEDCBA)	hexn_o, hex
0	0111111	3F	1000000	40
1	0000110	06	1111001	79
2	1011011	5B	0100100	24
3	1001111	4F	0110000	30
4	1100110	66	0011001	19
5	1101101	6D	0010010	12
6	1111101	7D	0000010	02
7	0000111	07	1111000	78
8	1111111	7F	0000000	00
9	1101111	6F	0010000	10
A	1110111	77	0001000	08
B	1111100	7C	0000011	03
C	0111001	39	1000110	46
D	1011110	5E	0100001	21
E	1111001	79	0000110	06
F	1110001	71	0001110	0E

Table 1: Truth table

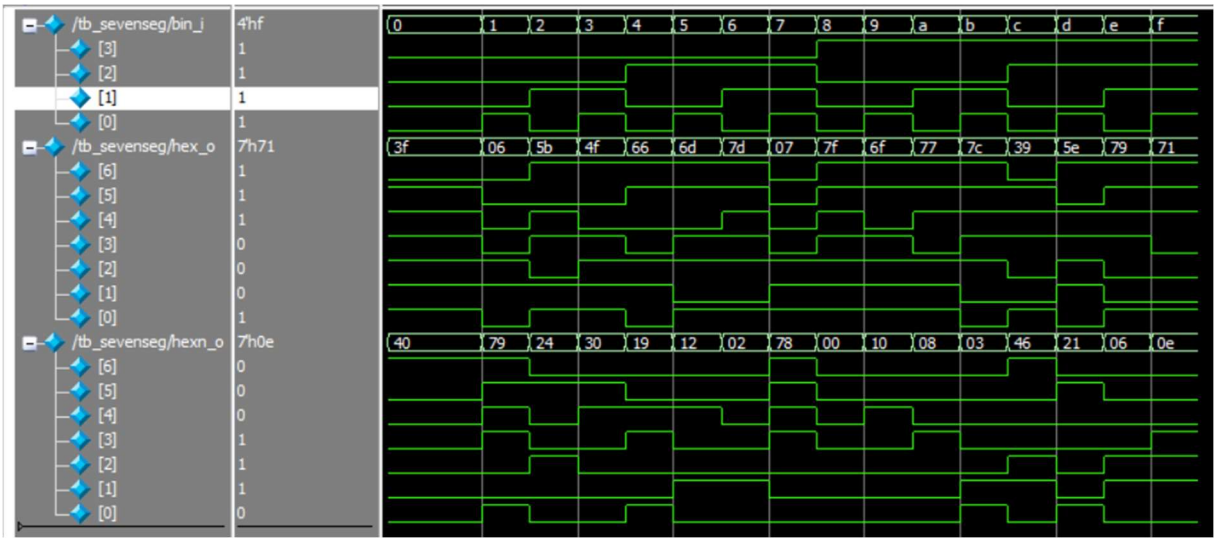

Figure 1: Outputs in respect of the inputs

Debug messages

For debugging purposes, a function was created to display the messages. All relevant data is copied to the function and then displayed. The output for every input is as follows:

```
# Displayed Value: 2
# Binary input: 0010
```

```
# LED Markings:                GFEDCBA
# State Output (non-inverted): 1011011
# State Output (inverted):     0100100
# ----------------------------------------------
```

Implementation method

For the implementation a case statement following the truth table was used to control every output simultaneously. The instruction is easy to read, and every bit represents one LED on the screen. So, every LED can be basically read without further decoding.

For exercising purposes, a second way for the implementation was also tested. In this second program, every line assigns the state to one LED segment, not the whole output. Here is an example for this method:

Segment A is active except the input has a value of 1, 4, B, D.
Segment B is active except the input has a value of 5, 6, B, C, E, F.

The negated logic was used because there are more zeroes than ones in every row of the truth table. This second implementation is not as self-explaining as the chosen implementation. This logic is not as easy to read and interpret. Therefore, the chosen implementation method was kept.