```
1    -- Testbench for Laboratory 11 Spring 2024
2
3
4
5    library ieee;
6    use ieee.std_logic_1164.all;
7    use ieee.numeric_std.all;
8    use work.all;
9
10
11   entity spi_mux_digit_driver_tb is
12   end spi_mux_digit_driver_tb;
13
14   architecture TB_ARCHITECTURE of spi_mux_digit_driver_tb is
15
16       -- Stimulus signals - signals mapped to the input and inout ports of
     tested entity
17       signal rst_bar : std_logic;
18       signal clk : std_logic;
19       signal mosi : std_logic;
20       signal sck : std_logic;
21       signal ss_bar : std_logic;
22       signal sel : std_logic_vector(1 downto 0) := "00";
23       -- Observed signals - signals mapped to the output ports of tested
     entity
24       -- Signal data_out : std_logic_vector(7 downto 0);
25       signal seg_drive : std_logic_vector(7 downto 0);
26
27       -- system clock period is being specified relative to shift
28       -- clock period so that effect of changing the system clock
29       -- on system's operation can be observed
30       constant sck_period : time := 4.0 us;
31       constant period : time := sck_period/4.0;
32       signal end_sim : boolean := false;
33
34
35   begin
36
37       -- Unit Under Test port map
38       UUT : entity spi_mux_digit_driver
39       port map (
40           rst_bar => rst_bar,
41           clk => clk,
42           mosi => mosi,
43           sck => sck,
44           ss_bar => ss_bar,
45           sel => sel,
46           seg_drive => seg_drive
47           );
48
49
50       -- generate system reset
51       rst_bar <= '0', '1' after period;
52
53
54       -- system clock runs until end_sim = false
55       clock_gen : process
```

```
56      begin
57          clk <= '0';
58          loop
59              wait for period/2;
60              clk <= not clk;
61              exit when end_sim = true;
62          end loop;
63          wait;
64      end process;
65
66
67      -- Generate SPI Shift Clock and MOSI data
68      send_spi_byte: process
69          variable data_in : std_logic_vector(7 downto 0);
70          variable addr : unsigned(7 downto 0) := "00000000";
71
72      begin
73          for k in 0 to 15 loop
74              data_in := "10000000" or std_logic_vector(addr) or
    std_logic_vector(to_unsigned(k, 8));
75              ss_bar <= '1';  -- select slave
76              sck <= '0';     -- starting shift clock value CPOL = 0
77              mosi <= '0';
78
79              wait for 2 * sck_period;
80              ss_bar <= '0';
81              wait for sck_period;
82              for i in 7 downto 0 loop    -- generate 8 data bits

83                  mosi <= data_in(i);     -- and shift clock pulses
84                  wait for sck_period/2;
85                  sck <= not sck;
86                  wait for sck_period/2;
87                  sck <= not sck;
88              end loop;       -- i index
89              wait for sck_period;
90              ss_bar <= '1';  -- deselect slave
91
92              for n in 0 to 3 loop
93                  sel <= std_logic_vector(to_unsigned(n, 2));
94                  wait for 10 * sck_period;
95              end loop;   -- n indexed loop
96
97              addr := (addr + "00010000") and "00110000";
98          end loop;   -- k indexed loop
99          end_sim <= true;     -- stop system clock
100         std.env.finish;     -- stop simulation
101     end process;
102
103 end tb_architecture;
104
105
106
107
```