```
1   ----------------------------------------------------------------------------------
2   -- Company:
3   -- Engineer:
4   --
5   -- Create Date:    13:22:06 10/25/2018
6   -- Design Name:
7   -- Module Name:    SevenSegmentA - Behavioral
8   -- Project Name:
9   -- Target Devices:
10  -- Tool versions:
11  -- Description:
12  --
13  -- Dependencies:
14  --
15  -- Revision:
16  -- Revision 0.01 - File Created
17  -- Additional Comments:
18  --
19  ----------------------------------------------------------------------------------
20  library IEEE;
21  use IEEE.STD_LOGIC_1164.ALL;
22
23  -- Uncomment the following library declaration if using
24  -- arithmetic functions with Signed or Unsigned values
25  --use IEEE.NUMERIC_STD.ALL;
26
27  -- Uncomment the following library declaration if instantiating
28  -- any Xilinx primitives in this code.
29  --library UNISIM;
30  --use UNISIM.VComponents.all;
31
32  entity SevenSegmentA is
33      Port ( SegAout : out  STD_LOGIC;
34             SegBout : out  STD_LOGIC;
35             SegCout : out  STD_LOGIC;
36             SegDout : out  STD_LOGIC;
37             SegEout : out  STD_LOGIC;
38             SegFout : out  STD_LOGIC;
39             SegGout : out  STD_LOGIC;
40             I0 : in  STD_LOGIC;
41             I1 : in  STD_LOGIC;
42             I2 : in  STD_LOGIC;
43             I3 : in  STD_LOGIC);
44  end SevenSegmentA;
45
46  architecture Behavioral of SevenSegmentA is
47
48  begin
49  SegAout <= (((not I3) and I2 and (not I1) and (not I0)) or ((not I3) and (not I2) and
    (not I1) and I0) or  (I3 and I2 and (not I1) and I0) or (I3 and (not I2) and I1 and I0
    ));
50  SegBout <= (((not I3) and I2 and (not I1) and I0) or (I2 and I1 and (not I0)) or (I3
    and I2 and (not I0)) or (I3 and I1 and I0));
51  SegCout <= (((not I3) and (not I2) and I1 and (not I0)) or (I3 and I2 and (not I0)) or
     (I3 and I2 and (not I0)) or (I3 and I2 and I1));
52  SegDout <= ((I2 and I1 and I0) or ((not I2) and (not I1) and I0) or ((not I3) and I2
    and (not I1) and (not I0)) or (I3 and (not I2) and I1 and (not I0)));
```

```
53    SegEout <= (((not I3) and I0) or ((not I2) and (not I1) and I0) or ((not I3) and I2
      and (not I1)));
54    SegFout <= ((I3 and I2 and (not I1) and I0) or ((not I3) and I1 and I0) or ((not I3)
      and (not I2) and I0) or ((not I3) and (not I2) and I1));
55    SegGout <= ((I3 and I2 and (not I1) and (not I0)) or ((not I3) and I2 and I1 and I0)
      or ((not I3) and (not I2) and (not I1)));
56
57    end Behavioral;
58
59
```