

```

1  --*****
2  --* Copyright (C) 2016 by Trevor Smouter
3  --*
4  --* All rights reserved.
5  --*
6  --* Redistribution and use in source and binary forms, with or without
7  --* modification, are permitted provided that the following conditions
8  --* are met:
9  --*
10 --* 1. Redistributions of source code must retain the above copyright
11 --* notice, this list of conditions and the following disclaimer.
12 --* 2. Redistributions in binary form must reproduce the above copyright
13 --* notice, this list of conditions and the following disclaimer in the
14 --* documentation and/or other materials provided with the distribution.
15 --* 3. Neither the name of the author nor the names of its contributors may
16 --* be used to endorse or promote products derived from this software
17 --* without specific prior written permission.
18 --*
19 --* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
20 --* "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
21 --* LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
22 --* FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL
23 --* THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
24 --* INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
25 --* BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS
26 --* OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED
27 --* AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
28 --* OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF
29 --* THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
30 --* SUCH DAMAGE.
31 --*
32 --*****
33 --* History:
34 --*
35 --* 04.02.2016 First Version
36 --*****
37
38
39 -- *****
40 -- * Library *
41 -- *****
42
43 library IEEE;
44 use IEEE.std_logic_1164.all;
45 use IEEE.numeric_std.all;
46
47
48 -- *****
49 -- * Entity *
50 -- *****
51
52 entity segment7_mux is
53     port (
54         clk          : in std_logic := '0';
55         DIN2         : in std_logic_vector(6 downto 0);
56         DIN1         : in std_logic_vector(6 downto 0);
57         DOUT         : out std_logic_vector(6 downto 0);
58         DIG2         : out std_logic;
59         DIG1         : out std_logic
60     );
61 end entity segment7_mux;
62
63 -- *****
64 -- * Architecture *
65 -- *****
66
67 architecture syn of segment7_mux is
68
69     signal toggle      : std_logic;
70     signal DOUT_TEMP   : std_logic_vector(6 downto 0);
71
72 begin
73
74     -----
75     -- Register File
76     -----

```

```

77
78
79   clk_proc:process(CLK)
80   variable COUNT      :unsigned(10 downto 0) := "0000000000";
81   begin
82       if (rising_edge(CLK)) then
83           COUNT := COUNT + 1;
84       else
85           COUNT := COUNT;
86       end if;
87       toggle <= COUNT(10);
88   end process clk_proc;
89   DIG1 <= NOT toggle;
90   DIG2 <= toggle;
91
92   DOUT_TEMP(0) <= (DIN2(0)) WHEN (toggle = '1') ELSE (DIN1(0));
93   DOUT_TEMP(1) <= (DIN2(1)) WHEN (toggle = '1') ELSE (DIN1(1));
94   DOUT_TEMP(2) <= (DIN2(2)) WHEN (toggle = '1') ELSE (DIN1(2));
95   DOUT_TEMP(3) <= (DIN2(3)) WHEN (toggle = '1') ELSE (DIN1(3));
96   DOUT_TEMP(4) <= (DIN2(4)) WHEN (toggle = '1') ELSE (DIN1(4));
97   DOUT_TEMP(5) <= (DIN2(5)) WHEN (toggle = '1') ELSE (DIN1(5));
98   DOUT_TEMP(6) <= (DIN2(6)) WHEN (toggle = '1') ELSE (DIN1(6));
99   -- DOUT_TEMP(7) <= (DIN2(7)) WHEN (toggle = '1') ELSE (DIN1(7));
100
101   DOUT(0) <= '0' WHEN (DOUT_TEMP(0) = '0') ELSE '1';
102   DOUT(1) <= '0' WHEN (DOUT_TEMP(1) = '0') ELSE 'Z'; --open drain
103   DOUT(2) <= '0' WHEN (DOUT_TEMP(2) = '0') ELSE '1';
104   DOUT(3) <= '0' WHEN (DOUT_TEMP(3) = '0') ELSE '1';
105   DOUT(4) <= '0' WHEN (DOUT_TEMP(4) = '0') ELSE '1';
106   DOUT(5) <= '0' WHEN (DOUT_TEMP(5) = '0') ELSE 'Z'; --open drain
107   DOUT(6) <= '0' WHEN (DOUT_TEMP(6) = '0') ELSE 'Z'; --open drain
108   -- DOUT(7) <= '0' WHEN (DOUT_TEMP(7) = '0') ELSE '1';
109
110
111
112
113
114   end architecture syn;
115

```