

```
1  -----
2  -- Company:
3  -- Engineer:
4  --
5  -- Create Date:    14:52:33 05/14/2021
6  -- Design Name:
7  -- Module Name:    TimerMealyMachine - 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 use IEEE.NUMERIC_STD.ALL;
23
24 -- Uncomment the following library declaration if using
25 -- arithmetic functions with Signed or Unsigned values
26
27
28 -- Uncomment the following library declaration if instantiating
29 -- any Xilinx primitives in this code.
30 --library UNISIM;
31 --use UNISIM.VComponents.all;
32
33 entity Timer is
34     Port ( reset : in  STD_LOGIC;
35           clock  : in  STD_LOGIC;
36           timer_reset : in  STD_LOGIC; -- To Reset Timer --
37           FourHzPulse : out STD_LOGIC; -- Four Hz Pulse --
38           MotorPulseOutput : out STD_LOGIC; -- Pulse For Stepper Motor --
39           CounterOutput : out STD_LOGIC_VECTOR (4 downto 0) -- Timer Output --
40         );
41 end Timer;
42
43 architecture Behavioral of Timer is
44
45     -- 1KHz = 1000 counts per second --
46     -- 1 second = 1000 counts --
47     -- 0.25 second = 250 counts --
48     -- 1 rotation in 1 second = 12 pulses = 83 counts --
49
50     signal counter : unsigned (9 downto 0); -- Unsigned Signal To Store Counter for
51     Generating Pulses --
52     signal timekeeper : unsigned (4 downto 0); -- Unsigned Signal To Store The Amount Of
53     Seconds --
54
55     begin
56
57     CounterOutput <= STD_LOGIC_VECTOR (timekeeper); -- Set CounterOutput Equal To
```

```
timekeeper --
56 FourHzPulse <= counter (8); -- Generate 4Hz Pulse According To The 8th Digit Of
Counter --
57
58
59 process (reset, timer_reset, clock) -- Whenever A Button Is Pressed, It Will Reset
The Timer; Button Is Mapped To Reset --
60
61 variable MotorCount: unsigned (9 downto 0); -- Unsigned Variable To Keep Track Of
Motor Pulse --
62
63 begin
64     if (reset = '1') then -- Master Reset --
65         counter <= (others => '0'); -- Reset counter To 0 --
66         timekeeper <= (others => '0'); -- Reset timekeeper To 0 --
67         MotorPulseOutput <= '0'; -- Reset MotorPulse To 0 --
68         MotorCount := "0001010011"; -- First Motor Pulse --
69
70     elsif (timer_reset = '1') then -- If Timer Is To Be Reset --
71         counter <= (others => '0'); -- Reset counter To 0 --
72         timekeeper <= (others => '0'); -- Reset timekeeper To 0 --
73         MotorPulseOutput <= '0'; -- Reset MotorPulse To 0 --
74         MotorCount := "0001010011"; -- First Motor Pulse --
75
76     elsif (rising_edge(clock)) then
77         counter <= counter + 1; -- Increment counter By 1 --
78
79         if (counter = "1111101000") then -- If counter Reached 1 second --
80             timekeeper <= timekeeper + 1; -- Increment timekeeper By 1 --
81
82         end if;
83
84         if (counter = MotorCount) then -- If counter Is The Same As MotorCount --
85             MotorPulseOutput <= '1'; -- MotorPulse Will Output 1 --
86             MotorCount := MotorCount + "0001010011"; -- Increment MotorPulse By 83
counts --
87
88         else
89             MotorPulseOutput <= '0'; -- If counter Is Not The Same As MotorCount,
MotorPulse Will Output 0 --
90
91         end if;
92
93     end if;
94
95 end process;
96 end Behavioral;
```