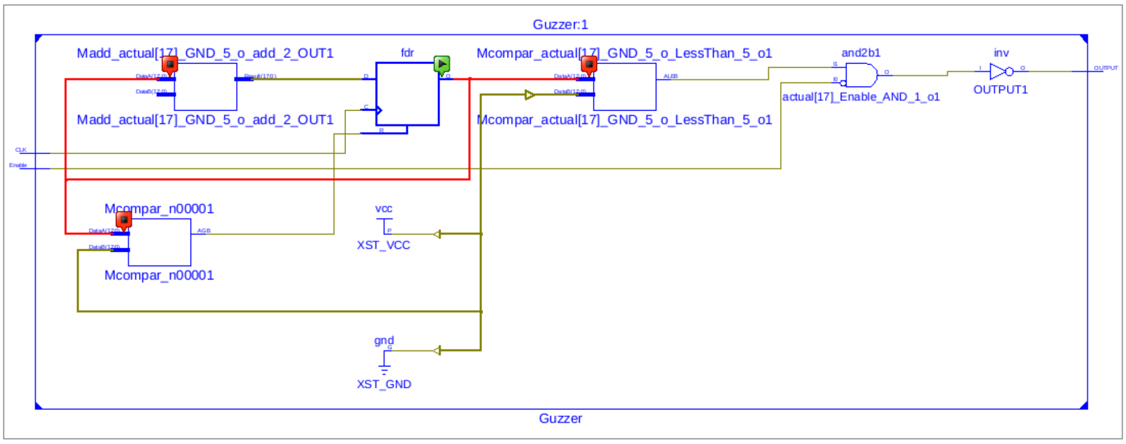
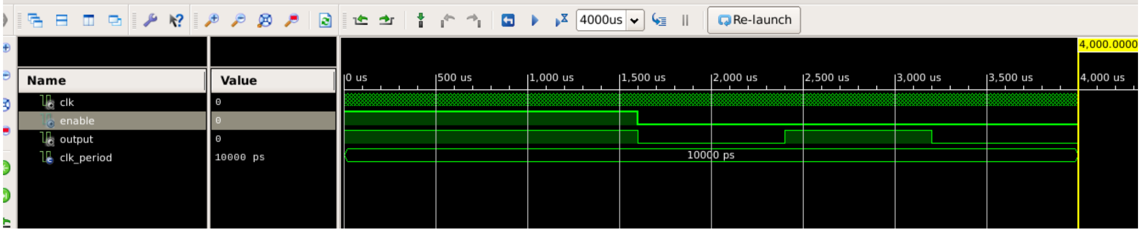
```
#------- IN ----

NET "Enable"           LOC = P124    | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = FAST |
PULLUP;

NET "CLK" LOC = P126;


#---- Out ----

NET "OUTPUT"               LOC = P35     | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = FAST
;   # IO_L1P_3               Sch = GPIO-P18
```

```
----------- Code -----------
------ Orlando Reyes ------
--------- Auf Das ---------
---------- Guzzer ----------
-------- 10/09/2024 --------
------- Main Library -------
library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
--------- Pin/out ---------
entity Guzzer is
    port
        (
        CLK, Enable : in std_logic;
        OUTPUT : out std_logic
        );
end Guzzer;

architecture juve3dstudio of Guzzer is
    signal BUZZER : STD_LOGIC;
    signal siguiente, actual : STD_LOGIC_VECTOR(17 downto 0) :=
"000000000000000000";
begin
    -- Memoria --
    actual <= siguiente when CLK'event and CLK = '1';
    -- Logica de estado Siguiente --
    siguiente <= actual + '1' when actual <= "100111000011111111" else
"000000000000000000";
    -- Logica de Salida --

    OUTPUT <= '0' when actual <= "010011100001111111"  and Enable='0'
else '1';
```

```
end juve3dstudio;
```

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--USE ieee.numeric_std.ALL;

ENTITY tb IS
END tb;

ARCHITECTURE behavior OF tb IS

    -- Component Declaration for the Unit Under Test (UUT)

    COMPONENT Guzzer
    PORT(
         CLK : IN  std_logic;
         Enable : IN  std_logic;
         OUTPUT : OUT  std_logic
        );
    END COMPONENT;


   --Inputs
   signal CLK : std_logic := '0';
   signal Enable : std_logic := '0';

   --Outputs
   signal OUTPUT : std_logic;

   -- Clock period definitions
   constant CLK_period : time := 10 ns;

BEGIN

   -- Instantiate the Unit Under Test (UUT)
   uut: Guzzer PORT MAP (
          CLK => CLK,
          Enable => Enable,
          OUTPUT => OUTPUT
        );

   -- Clock process definitions
   CLK_process :process
   begin
      CLK <= '0';
      wait for CLK_period/2;
      CLK <= '1';
      wait for CLK_period/2;
   end process;


   -- Stimulus process
   stim_proc: process
   begin
      -- hold reset state for 100 ns.
      Enable <= '0';
      wait for 1600 us;
      Enable <= '1';
      wait for 800 us;
      Enable <= '0';
      -- insert stimulus here

      wait;
   end process;

END;
```





Name	Value
clk	0
enable	0
output	0
clk_period	10000 ps