

## Additional Digital Design Project

### Brief Explanation of the Project

The project is done using BASYS3 FPGA and VHDL. The objective is to design a linear radar system with leds. There are 8 leds present in the project which are responsible for indicating the distance of the detected object. For every additional 5cm, a led turns on if the object is getting closer and the frequency of the buzzer increases for a better stimulation. System captures objects within 45cm range. The project primarily focuses on clock divisions and counters.

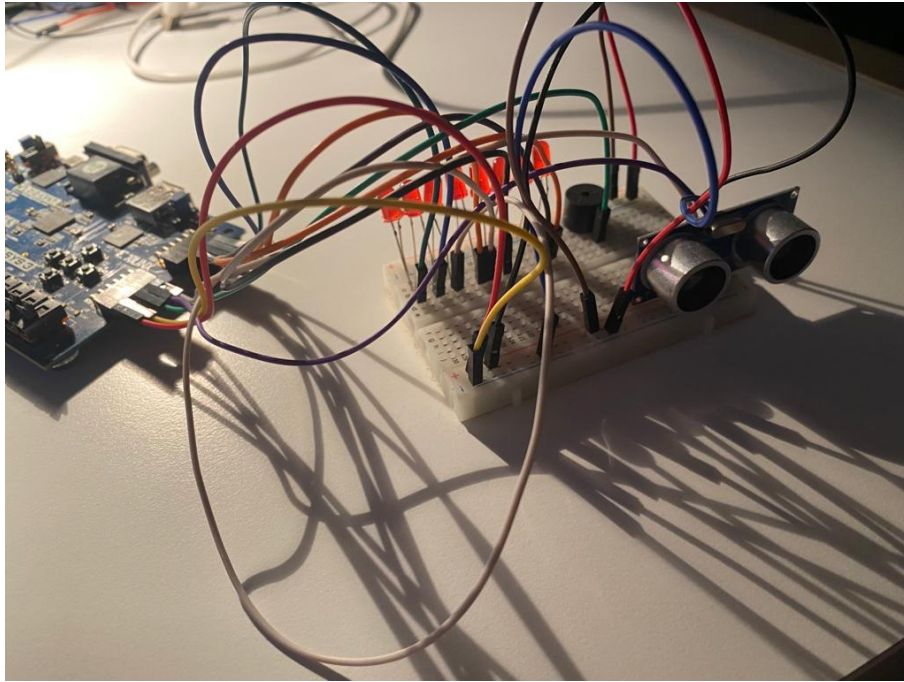


Fig1.1 hardware implementation

The visual demonstration of the Project is available in the following link:

<https://www.youtube.com/watch?v=YZ4MMU4cYd0>

### VHDL code:

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity test1 is
    port(
        CLOCK: in std_logic;
        LED: out std_logic_vector(7 downto 0);
        TRIG: out std_logic;
```

```

    ECHO: in std_logic;
    buzzer: out std_logic

```

```

    );
end test1;

```

architecture behaviour of test1 is

```

signal microseconds: std_logic;
signal counter: std_logic_vector(17 downto 0);
signal leds: std_logic_vector(7 downto 0);
signal trigger: std_logic;
signal clk_counter : natural range 0 to 100000000 := 0;
signal blinker: std_logic := '0';
signal change: std_logic_vector(7 downto 0);
begin

    process(CLOCK)
        variable count0: integer range 0 to 101;
        begin
            if rising_edge(CLOCK) then
                clk_counter <= clk_counter + 1;
                if clk_counter >= 90000000 and change = "10000000" then
                    blinker <= not blinker;
                    clk_counter <= 0;
                end if;
                if clk_counter >= 80000000 and change = "11000000" then
                    blinker <= not blinker;
                    clk_counter <= 0;
                end if;
                if clk_counter >= 70000000 and change = "11100000" then

```

```

    blinker <= not blinker;

    clk_counter <= 0;

end if;

if clk_counter >= 60000000 and change = "11110000" then

    blinker <= not blinker;

    clk_counter <= 0;

end if;

if clk_counter >= 50000000 and change = "11111000" then

    blinker <= not blinker;

    clk_counter <= 0;

end if;

if clk_counter >= 40000000 and change = "11111100" then

    blinker <= not blinker;

    clk_counter <= 0;

end if;

if clk_counter >= 20000000 and change = "11111110" then

    blinker <= not blinker;

    clk_counter <= 0;

end if;

if clk_counter >= 5000000 and change = "11111111" then

    blinker <= not blinker;

    clk_counter <= 0;

end if;

if change = "00000000" then

    blinker <= '0';

    clk_counter <= 0;

end if;

if count0 = 100 then

    count0 := 0;

else

```

```

        count0 := count0 + 1;
    end if;
    if count0 = 0 then
        microseconds <= not microseconds;
    end if;
end if;

end process ;

process(microseconds)
    variable count1: integer range 0 to 262143;
    begin
        if rising_edge(microseconds) then
            if count1 = 0 then
                counter <= "00000000000000000000";
                trigger <= '1';
            elsif count1 = 20 then
                trigger <= '0';
            end if;
            if ECHO = '1' then
                counter <= counter + 1;
            end if;
            if count1 = 249999 then
                count1 := 0;
            else
                count1 := count1 + 1;
            end if;
        end if;
    end if;
end process;

process(ECHO)

```

```
begin
  if falling_edge(ECHO) then
    if counter < 291 then
      leds <= "11111111";
      change <= "11111111";

    elsif counter < 581 then
      leds <= "11111110";
      change <= "11111110";

    elsif counter < 871 then
      leds <= "11111100";
      change <= "11111100";

    elsif counter < 1161 then
      leds <= "11111000";
      change <= "11111000";

    elsif counter < 1451 then
      leds <= "11110000";
      change <= "11110000";

    elsif counter < 1741 then
      leds <= "11100000";
      change <= "11100000";
```

```

elsif counter < 2031 then
    leds <= "11000000";
    change <= "11000000";

```

```

elsif counter < 2321 then
    leds <= "10000000";
    change <= "10000000";

```

```

else
    leds <= "00000000";
    change <= "00000000";

```

```

end if;

```

```

end if;

```

```

end process;

```

```

buzzer <= blinker;

```

```

LED <= leds;

```

```

TRIG <= trigger;

```

```

end behaviour;

```

#### **Code for Constraints:**

```

set_property PACKAGE_PIN W5 [get_ports CLOCK]

```

```

set_property IOSTANDARD LVCMOS33 [get_ports CLOCK]

```

```

set_property PACKAGE_PIN A14 [get_ports {LED[0]}]

```

```

set_property IOSTANDARD LVCMOS33 [get_ports {LED[0]}]

```

```

set_property PACKAGE_PIN A16 [get_ports {LED[1]}]

```

```

set_property IOSTANDARD LVCMOS33 [get_ports {LED[1]}]

```

```

set_property PACKAGE_PIN B15 [get_ports {LED[2]}]

```

```

set_property IOSTANDARD LVCMOS33 [get_ports {LED[2]}]

```

```
set_property PACKAGE_PIN B16 [get_ports {LED[3]]}
set_property IOSTANDARD LVCMOS33 [get_ports {LED[3]]}
set_property PACKAGE_PIN A15 [get_ports {LED[4]]}
set_property IOSTANDARD LVCMOS33 [get_ports {LED[4]]}
set_property PACKAGE_PIN A17 [get_ports {LED[5]]}
set_property IOSTANDARD LVCMOS33 [get_ports {LED[5]]}
set_property PACKAGE_PIN C15 [get_ports {LED[6]]}
set_property IOSTANDARD LVCMOS33 [get_ports {LED[6]]}
set_property PACKAGE_PIN C16 [get_ports {LED[7]]}
set_property IOSTANDARD LVCMOS33 [get_ports {LED[7]]}
set_property PACKAGE_PIN K17 [get_ports {TRIG}]
set_property IOSTANDARD LVCMOS33 [get_ports {TRIG}]
set_property PACKAGE_PIN M18 [get_ports {ECHO}]
set_property IOSTANDARD LVCMOS33 [get_ports {ECHO}]
set_property PACKAGE_PIN N17 [get_ports {buzzer}]
set_property IOSTANDARD LVCMOS33 [get_ports {buzzer}]
```