UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

Proiect

Generation of a siren sound

Realizat de :Cristea Luisa Monica

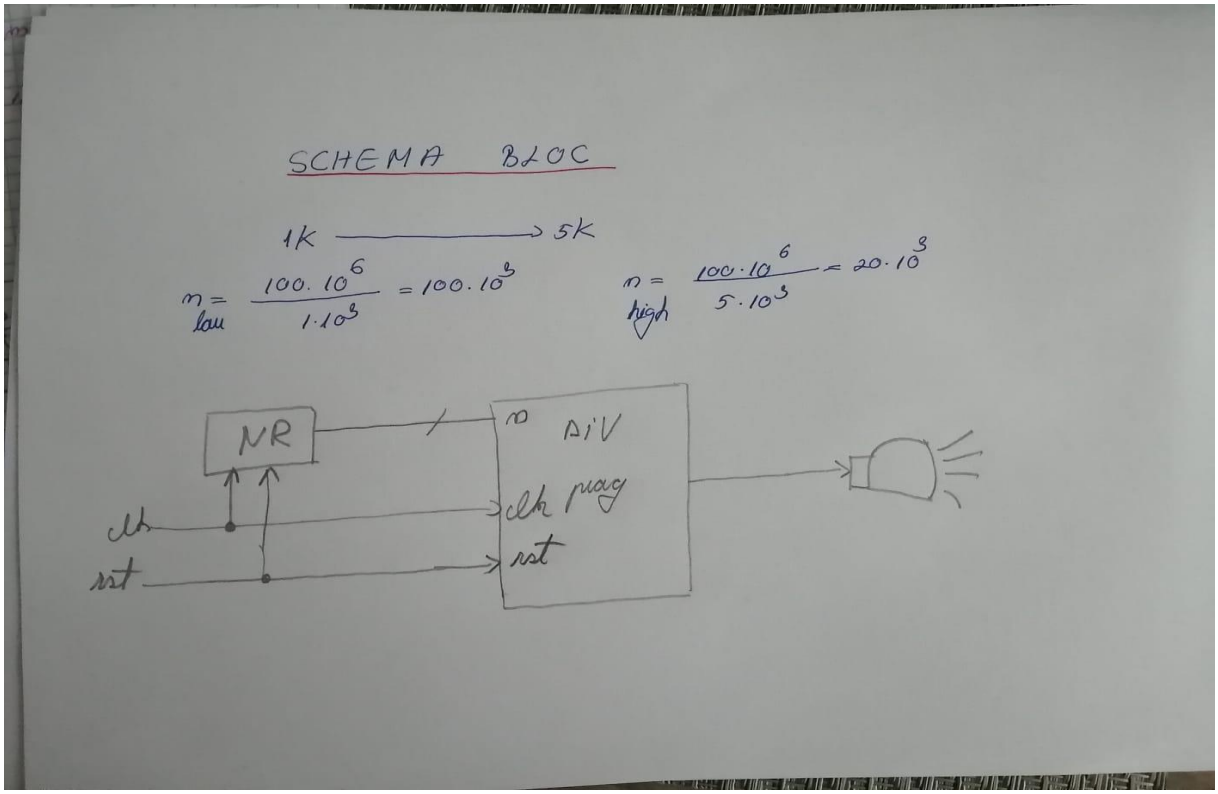
Îndrumător: profesor Paul Farago

Cerință proiect:

La activarea unui comutator , un sunet de sirena este generat pe un modul difuzor extern, care este o combinație între semnale de frecvență crescătoare și descrescătoare. Sirena va avea sunet: de exemplu un sunet ar consta în creșterea frecventei de la 1 kHz la 5 kHz, timp de 1.5 secunde.

Notă: Sunetul sirenei poate fi corelat cu un far (adică cu LED-uri)

Schema bloc :



Codul sursa:

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 05/05/2022 02:45:28 PM
-- Design Name:
-- Module Name: divizor_sunet - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
```

```
-- Revision 0.01 - File Created
-- Additional Comments:
--
----------------------------------------------------------------------------------


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;

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity divizor_sunet is
    Port ( clk_in : in STD_LOGIC;
           reset : in STD_LOGIC;
           clk_out : out STD_LOGIC);
end divizor_sunet;

architecture Behavioral of divizor_sunet is

signal temp: std_logic := '0';
signal counter: integer;

signal clk_div : std_logic := '0';

begin
```

```
div_frecv: process (reset, clk_in)
variable cnt : integer := 0;
begin

if(reset='1') then
cnt:=0;
elsif rising_edge(clk_in) then
if(cnt= 5000000) then  -- 5000000
clk_div <= NOT(clk_div);
cnt := 0;
else
cnt:=cnt+1;
clk_div<=clk_div;
end if;
end if;

end process;


factor_de_div: process(reset,clk_div)

begin
if(reset='1') then
--temp<='0';
counter<=20*10*4; --*10*4;
elsif rising_edge(clk_div) then
if(counter=5*10*4) then  --*10*4
--temp<= NOT(temp);
counter<=20*10*4; --*10*4;
else
counter<=counter - 4*10*4; -- * 10 * 4;
```

```
--temp<=temp;
end if;
end if;
end process;

sirena: process (reset, clk_in)
variable q : integer := 0;
begin

if(reset='1') then
q:=0;
elsif rising_edge(clk_in) then
if(q= counter/2) then
temp <= NOT(temp);
q := 0;
else
q:=q+1;
temp<=temp;
end if;
end if;

end process;

clk_out<= temp;

end Behavioral;
```

Cod simulare:

```
--------------------------------------------------------------------------------
-- Company:
-- Engineer:
```

```
--
-- Create Date: 05/05/2022 02:57:20 PM
-- Design Name:
-- Module Name: sim_div - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
----------------------------------------------------------------------------------


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;

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity sim_div is
--  Port ( );
```

```
end sim_div;

architecture Behavioral of sim_div is
COMPONENT divizor_sunet
	PORT(
		clk_in : IN  std_logic;
		reset  : IN  std_logic;
		clk_out: OUT std_logic
	);
	END COMPONENT;

	COMPONENT divizor_desc
      PORT(
	   clk_in : IN  std_logic;
	   reset  : IN  std_logic;
	   clk_out: OUT std_logic
      );
      END COMPONENT;


	signal clk_in  : std_logic := '0';
	signal reset   : std_logic := '0';
	-- Outputs
	signal clk_out : std_logic;
	constant clk_in_t : time := 20 ns;
BEGIN

	uut: divizor_sunet PORT MAP (
		clk_in  => clk_in,
		reset   => reset,
		clk_out => clk_out
	);
```

```
--        uut1: divizor_desc PORT MAP (
--           clk_in  => clk_in,
--           reset   => reset,
--           clk_out => clk_out
--        );


          ent_process :process
                    begin
                    clk_in <= '0';
                    wait for clk_in_t / 2;
                    clk_in <= '1';
                    wait for clk_in_t / 2;
          end process;


--        sim: process
--        begin
                    reset <= '1' after 0 ns, '0' after 100 ns;
--        end process;
end Behavioral;
```

Cod xdc:

```
## This file is a general .xdc for the Basys3 rev B board
## To use it in a project:
## - uncomment the lines corresponding to used pins
## - rename the used ports (in each line, after get_ports) according to the top level signal names in the project

## Clock signal
set_property -dict { PACKAGE_PIN W5   IOSTANDARD LVCMOS33 } [get_ports clk_in]
create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports clk_in]
```

```
## Switches
#set_property -dict { PACKAGE_PIN V17   IOSTANDARD LVCMOS33 } [get_ports {}]
#set_property -dict { PACKAGE_PIN V16   IOSTANDARD LVCMOS33 } [get_ports {}]
#set_property -dict { PACKAGE_PIN W16   IOSTANDARD LVCMOS33 } [get_ports {}]
#set_property -dict { PACKAGE_PIN W17   IOSTANDARD LVCMOS33 } [get_ports {}]
#set_property -dict { PACKAGE_PIN W15   IOSTANDARD LVCMOS33 } [get_ports {sw[4]}]
#set_property -dict { PACKAGE_PIN V15   IOSTANDARD LVCMOS33 } [get_ports {sw[5]}]
#set_property -dict { PACKAGE_PIN W14   IOSTANDARD LVCMOS33 } [get_ports {sw[6]}]
#set_property -dict { PACKAGE_PIN W13   IOSTANDARD LVCMOS33 } [get_ports {sw[7]}]
#set_property -dict { PACKAGE_PIN V2    IOSTANDARD LVCMOS33 } [get_ports {sw[8]}]
#set_property -dict { PACKAGE_PIN T3    IOSTANDARD LVCMOS33 } [get_ports {sw[9]}]
#set_property -dict { PACKAGE_PIN T2    IOSTANDARD LVCMOS33 } [get_ports {sw[10]}]
#set_property -dict { PACKAGE_PIN R3    IOSTANDARD LVCMOS33 } [get_ports {sw[11]}]
#set_property -dict { PACKAGE_PIN W2    IOSTANDARD LVCMOS33 } [get_ports {sw[12]}]
#set_property -dict { PACKAGE_PIN U1    IOSTANDARD LVCMOS33 } [get_ports {sw[13]}]
#set_property -dict { PACKAGE_PIN T1    IOSTANDARD LVCMOS33 } [get_ports {sw[14]}]
#set_property -dict { PACKAGE_PIN R2    IOSTANDARD LVCMOS33 } [get_ports {sw[15]}]


## LEDs
#set_property -dict { PACKAGE_PIN U16   IOSTANDARD LVCMOS33 } [get_ports {led[0]}]
#set_property -dict { PACKAGE_PIN E19   IOSTANDARD LVCMOS33 } [get_ports {led[1]}]
#set_property -dict { PACKAGE_PIN U19   IOSTANDARD LVCMOS33 } [get_ports {led[2]}]
#set_property -dict { PACKAGE_PIN V19   IOSTANDARD LVCMOS33 } [get_ports {led[3]}]
#set_property -dict { PACKAGE_PIN W18   IOSTANDARD LVCMOS33 } [get_ports {led[4]}]
#set_property -dict { PACKAGE_PIN U15   IOSTANDARD LVCMOS33 } [get_ports {led[5]}]
#set_property -dict { PACKAGE_PIN U14   IOSTANDARD LVCMOS33 } [get_ports {led[6]}]
#set_property -dict { PACKAGE_PIN V14   IOSTANDARD LVCMOS33 } [get_ports {led[7]}]
#set_property -dict { PACKAGE_PIN V13   IOSTANDARD LVCMOS33 } [get_ports {led[8]}]
#set_property -dict { PACKAGE_PIN V3    IOSTANDARD LVCMOS33 } [get_ports {led[9]}]
```

```
#set_property -dict { PACKAGE_PIN W3    IOSTANDARD LVCMOS33 } [get_ports {led[10]}]
#set_property -dict { PACKAGE_PIN U3    IOSTANDARD LVCMOS33 } [get_ports {led[11]}]
#set_property -dict { PACKAGE_PIN P3    IOSTANDARD LVCMOS33 } [get_ports {led[12]}]
#set_property -dict { PACKAGE_PIN N3    IOSTANDARD LVCMOS33 } [get_ports {led[13]}]
#set_property -dict { PACKAGE_PIN P1    IOSTANDARD LVCMOS33 } [get_ports {led[14]}]
#set_property -dict { PACKAGE_PIN L1    IOSTANDARD LVCMOS33 } [get_ports {led[15]}]


##7 Segment Display
#set_property -dict { PACKAGE_PIN W7   IOSTANDARD LVCMOS33 } [get_ports {seg[0]}]
#set_property -dict { PACKAGE_PIN W6   IOSTANDARD LVCMOS33 } [get_ports {seg[1]}]
#set_property -dict { PACKAGE_PIN U8   IOSTANDARD LVCMOS33 } [get_ports {seg[2]}]
#set_property -dict { PACKAGE_PIN V8   IOSTANDARD LVCMOS33 } [get_ports {seg[3]}]
#set_property -dict { PACKAGE_PIN U5   IOSTANDARD LVCMOS33 } [get_ports {seg[4]}]
#set_property -dict { PACKAGE_PIN V5   IOSTANDARD LVCMOS33 } [get_ports {seg[5]}]
#set_property -dict { PACKAGE_PIN U7   IOSTANDARD LVCMOS33 } [get_ports {seg[6]}]

#set_property -dict { PACKAGE_PIN V7   IOSTANDARD LVCMOS33 } [get_ports dp]

#set_property -dict { PACKAGE_PIN U2   IOSTANDARD LVCMOS33 } [get_ports {an[0]}]
#set_property -dict { PACKAGE_PIN U4   IOSTANDARD LVCMOS33 } [get_ports {an[1]}]
#set_property -dict { PACKAGE_PIN V4   IOSTANDARD LVCMOS33 } [get_ports {an[2]}]
#set_property -dict { PACKAGE_PIN W4   IOSTANDARD LVCMOS33 } [get_ports {an[3]}]


##Buttons
set_property -dict { PACKAGE_PIN U18   IOSTANDARD LVCMOS33 } [get_ports reset]
#set_property -dict { PACKAGE_PIN T18   IOSTANDARD LVCMOS33 } [get_ports btnU]
#set_property -dict { PACKAGE_PIN W19   IOSTANDARD LVCMOS33 } [get_ports btnL]
#set_property -dict { PACKAGE_PIN T17   IOSTANDARD LVCMOS33 } [get_ports btnR]
#set_property -dict { PACKAGE_PIN U17   IOSTANDARD LVCMOS33 } [get_ports btnD]
```

##Pmod Header JA

#set_property -dict { PACKAGE_PIN J1   IOSTANDARD LVCMOS33 } [get_ports {clk_out}];#Sch name = JA1

#set_property -dict { PACKAGE_PIN L2   IOSTANDARD LVCMOS33 } [get_ports {JA[1]}];#Sch name = JA2

#set_property -dict { PACKAGE_PIN J2   IOSTANDARD LVCMOS33 } [get_ports {JA[2]}];#Sch name = JA3

#set_property -dict { PACKAGE_PIN G2   IOSTANDARD LVCMOS33 } [get_ports {JA[3]}];#Sch name = JA4

#set_property -dict { PACKAGE_PIN H1   IOSTANDARD LVCMOS33 } [get_ports {JA[4]}];#Sch name = JA7

#set_property -dict { PACKAGE_PIN K2   IOSTANDARD LVCMOS33 } [get_ports {JA[5]}];#Sch name = JA8

#set_property -dict { PACKAGE_PIN H2   IOSTANDARD LVCMOS33 } [get_ports {JA[6]}];#Sch name = JA9

#set_property -dict { PACKAGE_PIN G3   IOSTANDARD LVCMOS33 } [get_ports {JA[7]}];#Sch name = JA10

##Pmod Header JB

set_property -dict { PACKAGE_PIN A14   IOSTANDARD LVCMOS33 } [get_ports {clk_out}];#Sch name = JB1

#set_property -dict { PACKAGE_PIN A16   IOSTANDARD LVCMOS33 } [get_ports {JB[1]}];#Sch name = JB2

#set_property -dict { PACKAGE_PIN B15   IOSTANDARD LVCMOS33 } [get_ports {JB[2]}];#Sch name = JB3

#set_property -dict { PACKAGE_PIN B16   IOSTANDARD LVCMOS33 } [get_ports {JB[3]}];#Sch name = JB4

#set_property -dict { PACKAGE_PIN A15   IOSTANDARD LVCMOS33 } [get_ports {JB[4]}];#Sch name = JB7

#set_property -dict { PACKAGE_PIN A17   IOSTANDARD LVCMOS33 } [get_ports {JB[5]}];#Sch name = JB8

#set_property -dict { PACKAGE_PIN C15   IOSTANDARD LVCMOS33 } [get_ports {JB[6]}];#Sch name = JB9

#set_property -dict { PACKAGE_PIN C16   IOSTANDARD LVCMOS33 } [get_ports {JB[7]}];#Sch name = JB10

##Pmod Header JC

#set_property -dict { PACKAGE_PIN K17 IOSTANDARD LVCMOS33 } [get_ports {JC[0]}];#Sch name = JC1

#set_property -dict { PACKAGE_PIN M18 IOSTANDARD LVCMOS33 } [get_ports {JC[1]}];#Sch name = JC2

#set_property -dict { PACKAGE_PIN N17 IOSTANDARD LVCMOS33 } [get_ports {JC[2]}];#Sch name = JC3

#set_property -dict { PACKAGE_PIN P18 IOSTANDARD LVCMOS33 } [get_ports {JC[3]}];#Sch name = JC4

#set_property -dict { PACKAGE_PIN L17 IOSTANDARD LVCMOS33 } [get_ports {JC[4]}];#Sch name = JC7

#set_property -dict { PACKAGE_PIN M19 IOSTANDARD LVCMOS33 } [get_ports {JC[5]}];#Sch name = JC8

#set_property -dict { PACKAGE_PIN P17 IOSTANDARD LVCMOS33 } [get_ports {JC[6]}];#Sch name = JC9

#set_property -dict { PACKAGE_PIN R18 IOSTANDARD LVCMOS33 } [get_ports {JC[7]}];#Sch name = JC10

##Pmod Header JXADC

#set_property -dict { PACKAGE_PIN J3 IOSTANDARD LVCMOS33 } [get_ports {JXADC[0]}];#Sch name = XA1_P

#set_property -dict { PACKAGE_PIN L3 IOSTANDARD LVCMOS33 } [get_ports {JXADC[1]}];#Sch name = XA2_P

#set_property -dict { PACKAGE_PIN M2 IOSTANDARD LVCMOS33 } [get_ports {JXADC[2]}];#Sch name = XA3_P

#set_property -dict { PACKAGE_PIN N2 IOSTANDARD LVCMOS33 } [get_ports {JXADC[3]}];#Sch name = XA4_P

#set_property -dict { PACKAGE_PIN K3 IOSTANDARD LVCMOS33 } [get_ports {JXADC[4]}];#Sch name = XA1_N

#set_property -dict { PACKAGE_PIN M3 IOSTANDARD LVCMOS33 } [get_ports {JXADC[5]}];#Sch name = XA2_N

#set_property -dict { PACKAGE_PIN M1 IOSTANDARD LVCMOS33 } [get_ports {JXADC[6]}];#Sch name = XA3_N

#set_property -dict { PACKAGE_PIN N1 IOSTANDARD LVCMOS33 } [get_ports {JXADC[7]}];#Sch name = XA4_N

##VGA Connector

#set_property -dict { PACKAGE_PIN G19 IOSTANDARD LVCMOS33 } [get_ports {vgaRed[0]}]

#set_property -dict { PACKAGE_PIN H19   IOSTANDARD LVCMOS33 } [get_ports {vgaRed[1]}]

#set_property -dict { PACKAGE_PIN J19   IOSTANDARD LVCMOS33 } [get_ports {vgaRed[2]}]

#set_property -dict { PACKAGE_PIN N19   IOSTANDARD LVCMOS33 } [get_ports {vgaRed[3]}]

#set_property -dict { PACKAGE_PIN N18   IOSTANDARD LVCMOS33 } [get_ports {vgaBlue[0]}]

#set_property -dict { PACKAGE_PIN L18   IOSTANDARD LVCMOS33 } [get_ports {vgaBlue[1]}]

#set_property -dict { PACKAGE_PIN K18   IOSTANDARD LVCMOS33 } [get_ports {vgaBlue[2]}]

#set_property -dict { PACKAGE_PIN J18   IOSTANDARD LVCMOS33 } [get_ports {vgaBlue[3]}]

#set_property -dict { PACKAGE_PIN J17   IOSTANDARD LVCMOS33 } [get_ports {vgaGreen[0]}]

#set_property -dict { PACKAGE_PIN H17   IOSTANDARD LVCMOS33 } [get_ports {vgaGreen[1]}]

#set_property -dict { PACKAGE_PIN G17   IOSTANDARD LVCMOS33 } [get_ports {vgaGreen[2]}]

#set_property -dict { PACKAGE_PIN D17   IOSTANDARD LVCMOS33 } [get_ports {vgaGreen[3]}]

#set_property -dict { PACKAGE_PIN P19   IOSTANDARD LVCMOS33 } [get_ports Hsync]

#set_property -dict { PACKAGE_PIN R19   IOSTANDARD LVCMOS33 } [get_ports Vsync]

##USB-RS232 Interface

#set_property -dict { PACKAGE_PIN B18   IOSTANDARD LVCMOS33 } [get_ports RsRx]

#set_property -dict { PACKAGE_PIN A18   IOSTANDARD LVCMOS33 } [get_ports RsTx]

##USB HID (PS/2)

#set_property -dict { PACKAGE_PIN C17   IOSTANDARD LVCMOS33   PULLUP true } [get_ports PS2Clk]

#set_property -dict { PACKAGE_PIN B17   IOSTANDARD LVCMOS33   PULLUP true } [get_ports PS2Data]

##Quad SPI Flash

##Note that CCLK_0 cannot be placed in 7 series devices. You can access it using the

##STARTUPE2 primitive.

#set_property -dict { PACKAGE_PIN D18   IOSTANDARD LVCMOS33 } [get_ports {QspiDB[0]}]

#set_property -dict { PACKAGE_PIN D19   IOSTANDARD LVCMOS33 } [get_ports {QspiDB[1]}]

#set_property -dict { PACKAGE_PIN G18   IOSTANDARD LVCMOS33 } [get_ports {QspiDB[2]}]

#set_property -dict { PACKAGE_PIN F18   IOSTANDARD LVCMOS33 } [get_ports {QspiDB[3]}]

#set_property -dict { PACKAGE_PIN K19   IOSTANDARD LVCMOS33 } [get_ports QspiCSn]

Configuration options, can be used for all designs

set_property CONFIG_VOLTAGE 3.3 [current_design]

set_property CFGBVS VCCO [current_design]