.UBAfiuba
FACULTAD DE INGENIERÍA

PWM Configurable

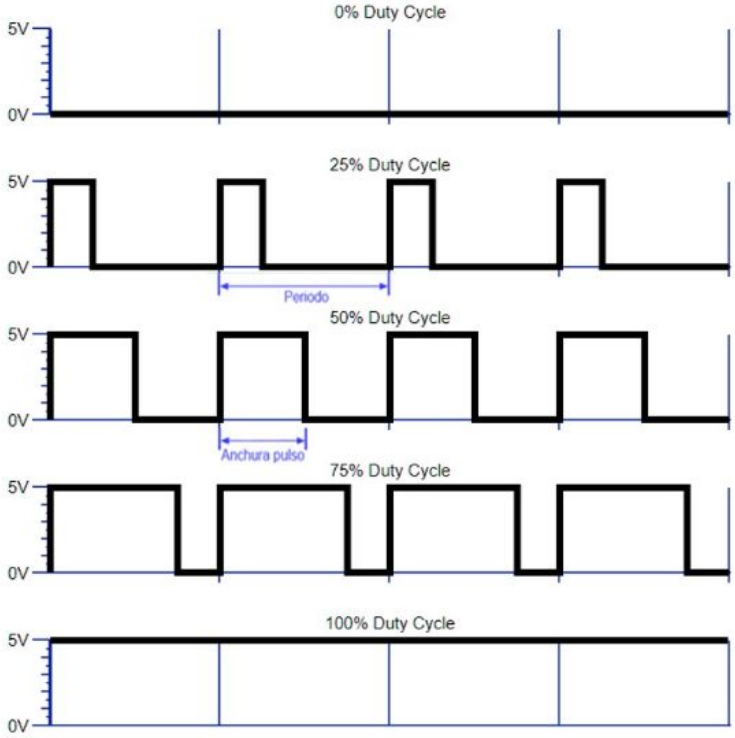
Trabajo integrador del curso
Circuitos Lógicos Programables

Autor:
Ing. Lautaro Juan Bautista Vera

PWM: Introducción

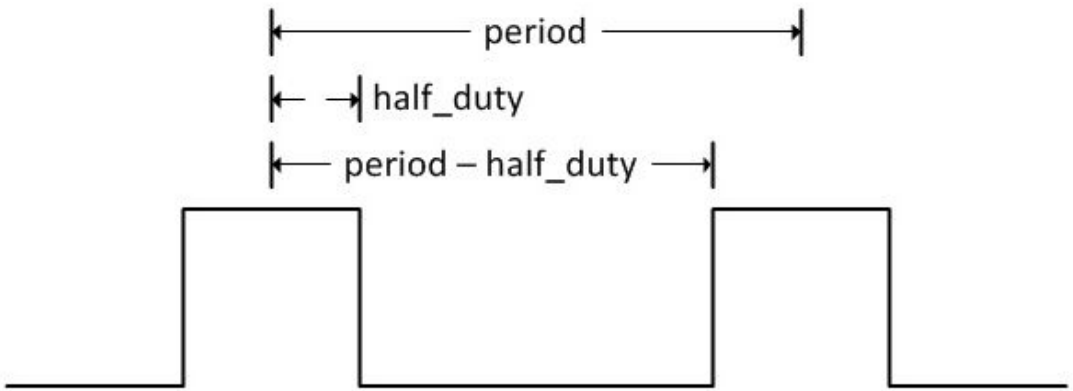
- Duty Cycle = Ton / (Ton + Toff)
- F(sys) (~MHz) >> F(pwm) (~KHz)
- Número de pulsos del clock dentro de un período:
  - T(pwm) = Clk Cnt = F(sys) / F(pwm)
- Dado una resolución N en bits, se calcula la cuenta de duty cycle:
  - Duty Cycle Cnt = Clk Cnt * Duty Cycle(input) / (2 ^ N)

PWM: Duty Cycle



VHDL: Diseño

- Se toma como período PWM la mitad del pulso para evitar los flancos de la señal.



VHDL: Código fuente

- Jerarquía plana => pwm.vhd
- Entidad:

```
entity pwm is
    generic(
        sys_freq    : integer := 50e6;
        pwm_freq    : integer := 100e3;
        pwm_bits    : integer := 4
    );
    port(
        clk_i   : in std_logic;
        rst_i   : in std_logic;
        ena_i   : in std_logic;
        duty_i  : in std_logic_vector(pwm_bits-1 downto 0);
        out_o   : out std_logic
    );
end;
```

VHDL: Código fuente

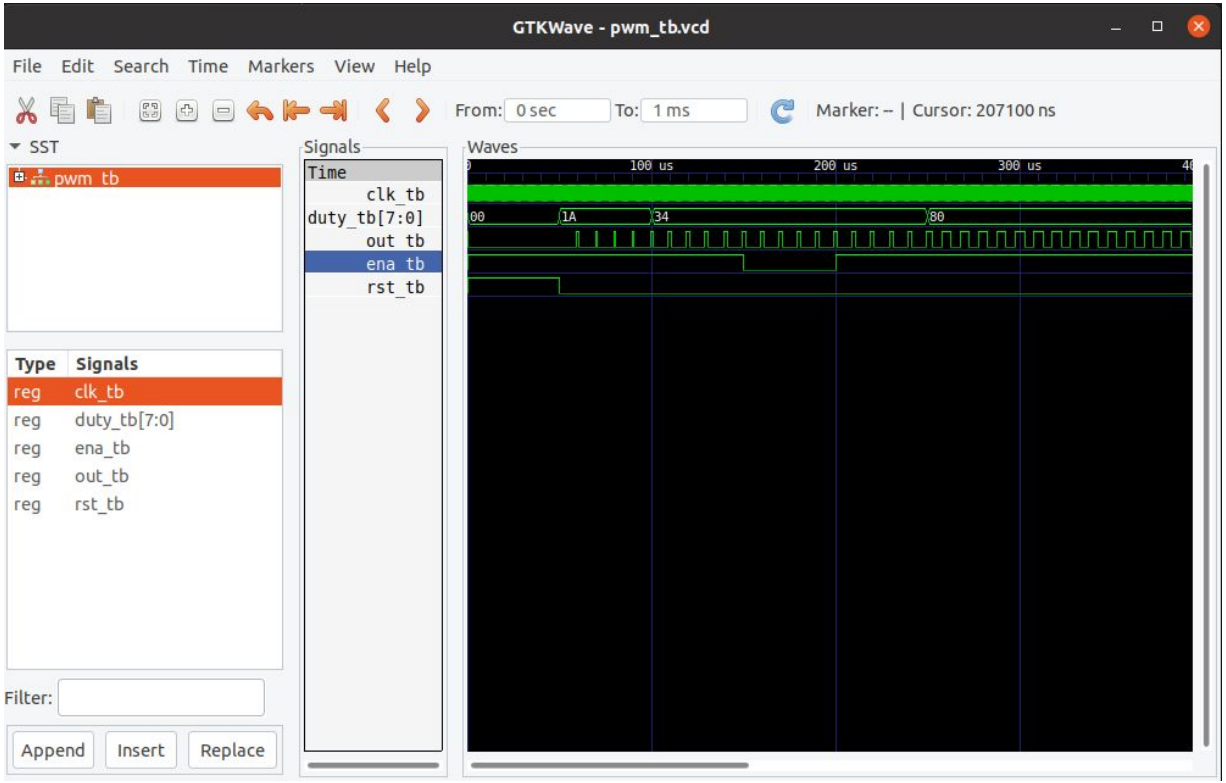
- Arquitectura:

```
19
20  architecture pwm_arq of pwm is
21      -- Parte declarativa
22
23      constant pwm_period : integer := sys_freq/pwm_freq;
24
25      signal count      : integer range 0 to pwm_period - 1 := 0;
26      signal half_duty  : integer range 0 to pwm_period/2 := 0;
27
28  begin
29      -- Parte descriptiva
30
31      process(clk_i, rst_i)
32      begin
33          if (rst_i = '1') then
34              count <= 0;
35              out_o <= '0';
36          elsif rising_edge(clk_i) then
37              if (ena_i = '1') then
38                  half_duty  <= to_integer(unsigned(duty_i))*pwm_period/(2**pwm_bits)/2;
39              end if;
40              if (count = pwm_period - 1) then
41                  count <= 0;
42              else
43                  count <= count + 1;
44              end if;
45              if (count = half_duty) then
46                  out_o <= '0';
47              elsif (count = pwm_period - half_duty) then
48                  out_o <= '1';
49              end if;
50          end if;
51      end process;
52
53  end;
54
```

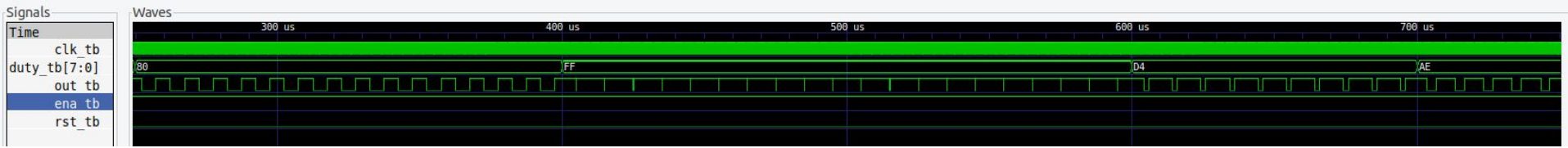
VHDL: Código de banco de pruebas

```vhdl
begin

    clk_tb  <= not clk_tb after 10 ns;
    rst_tb  <= '0' after 50 us;
    ena_tb  <= '0' after 150 us, '1' after 200 us, '0' after 900 us;
    duty_tb <= std_logic_vector(to_unsigned(26, pwm_bits_tb)) after 50 us,
               std_logic_vector(to_unsigned(52, pwm_bits_tb)) after 100 us,
               std_logic_vector(to_unsigned(128, pwm_bits_tb)) after 250 us,
               std_logic_vector(to_unsigned(255, pwm_bits_tb)) after 400 us,
               std_logic_vector(to_unsigned(212, pwm_bits_tb)) after 600 us,
               std_logic_vector(to_unsigned(174, pwm_bits_tb)) after 700 us,
               std_logic_vector(to_unsigned(100, pwm_bits_tb)) after 800 us;

    DUT: pwm
        generic map(
            sys_freq => sys_freq_tb,
            pwm_freq => pwm_freq_tb,
            pwm_bits => pwm_bits_tb
        )
        port map(
            clk_i   => clk_tb,
            rst_i   => rst_tb,
            ena_i   => ena_tb,
            duty_i  => duty_tb,
            out_o   => out_tb
        );

end;
```

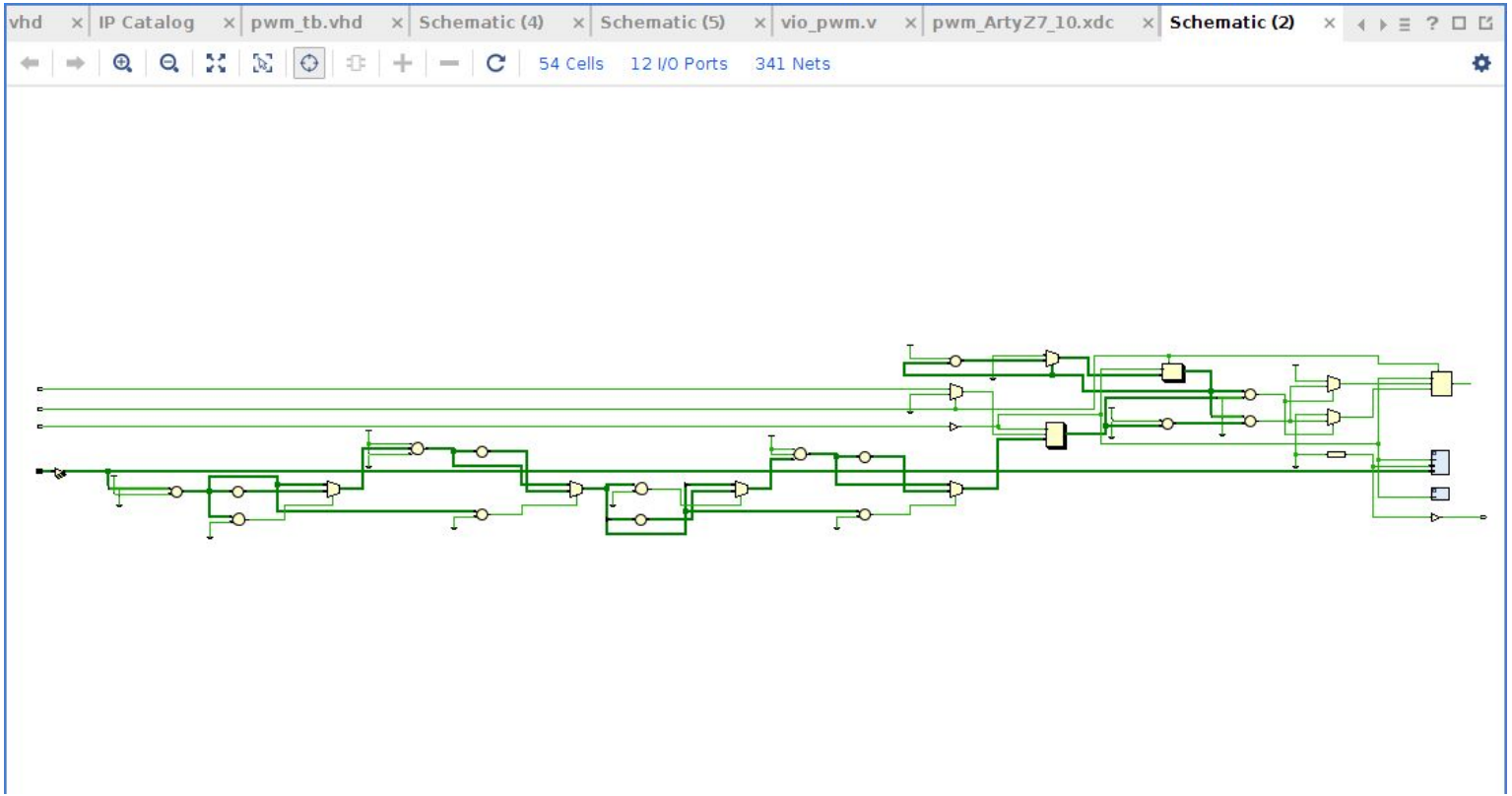
Simulación


GTKWave - pwm_tb.vcd
File Edit Search Time Markers View Help
From: 0 sec
To: 1 ms
Marker: -- | Cursor: 207100 ns
SST
pwm_tb
Type Signals
reg clk_tb
reg duty_tb[7:0]
reg ena_tb
reg out_tb
reg rst_tb
Filter:
Append Insert Replace
Signals
Time
clk_tb
duty_tb[7:0]
out_tb
ena_tb
rst_tb
Waves
100 us
200 us
300 us
00
1A
34
80

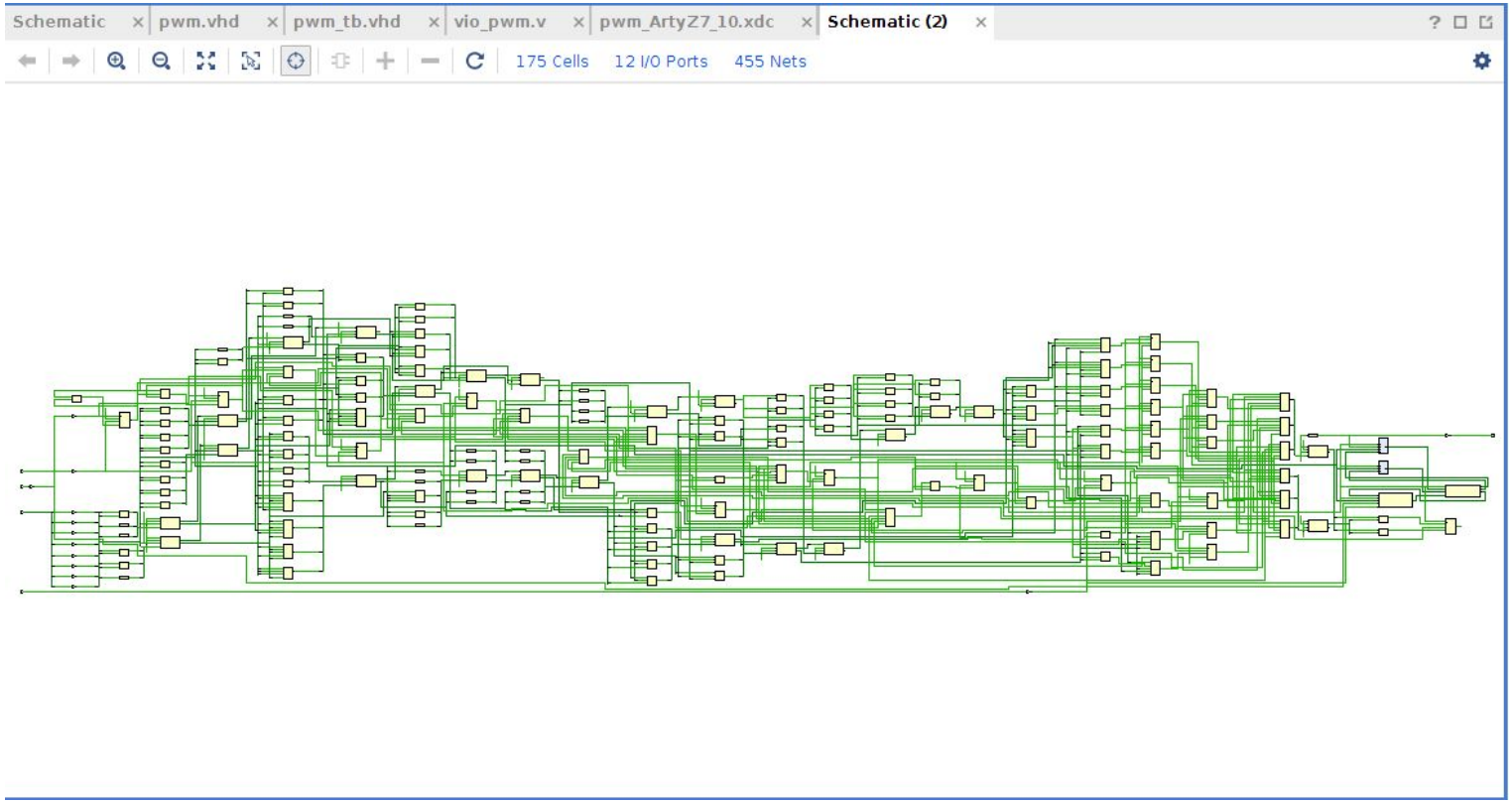
Simulación


Signals
Waves
Time
clk_tb
duty_tb[7:0]
out_tb
ena_tb
rst_tb
300 us
400 us
500 us
600 us
700 us
80
FF
D4
AE

Vivado: Diseño Elaborado



Vivado: Síntesis



Vivado: Constraints

```
############################
# Arty Z7 Pin Assignments
############################

# Duty Cycle
set_property -dict {PACKAGE_PIN T14 IOSTANDARD LVCMOS33} [get_ports {duty_i[0]}]
set_property -dict {PACKAGE_PIN U12 IOSTANDARD LVCMOS33} [get_ports {duty_i[1]}]
set_property -dict {PACKAGE_PIN U13 IOSTANDARD LVCMOS33} [get_ports {duty_i[2]}]
set_property -dict {PACKAGE_PIN V13 IOSTANDARD LVCMOS33} [get_ports {duty_i[3]}]
set_property -dict {PACKAGE_PIN V15 IOSTANDARD LVCMOS33} [get_ports {duty_i[4]}]
set_property -dict {PACKAGE_PIN T15 IOSTANDARD LVCMOS33} [get_ports {duty_i[5]}]
set_property -dict {PACKAGE_PIN R16 IOSTANDARD LVCMOS33} [get_ports {duty_i[6]}]
set_property -dict {PACKAGE_PIN U17 IOSTANDARD LVCMOS33} [get_ports {duty_i[7]}]

# CLK source 50 MHz
set_property -dict {PACKAGE_PIN H16 IOSTANDARD LVCMOS33} [get_ports clk_i]

# Rst Btn
set_property -dict {PACKAGE_PIN L19 IOSTANDARD LVCMOS33} [get_ports rst_i]

# Ena Btn
set_property -dict {PACKAGE_PIN D19 IOSTANDARD LVCMOS33} [get_ports ena_i]

# PWM
set_property -dict {PACKAGE_PIN Y18 IOSTANDARD LVCMOS33} [get_ports out_o]

############################
# Arty Z7 Timing Assignments
############################

# define clock and period
create_clock -period 20.000 -name clk_pin -waveform {0.000 10.000} [get_ports clk_i]


```

Vivado: Implementación y Depuración