

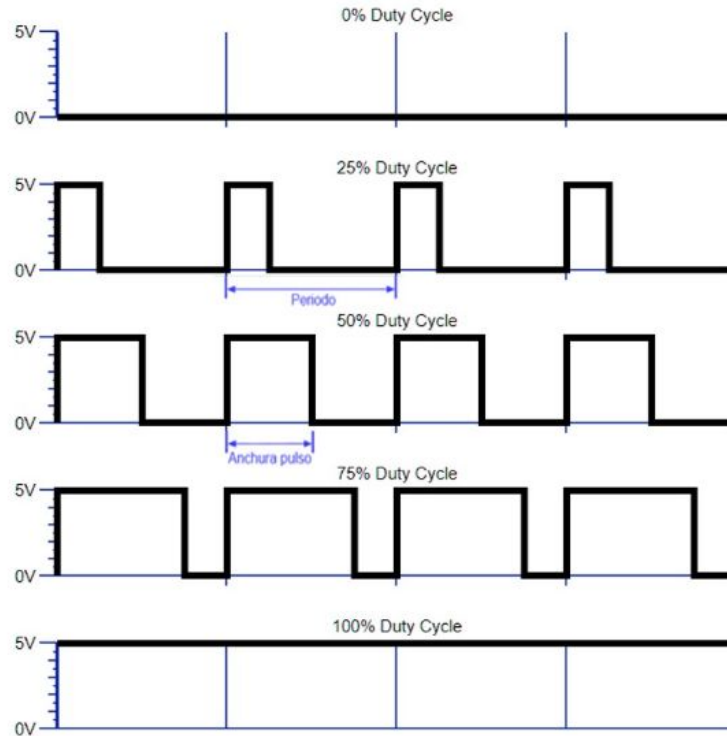
PWM Configurable

Trabajo integrador del curso
Circuitos Lógicos Programables

PWM: Introducción

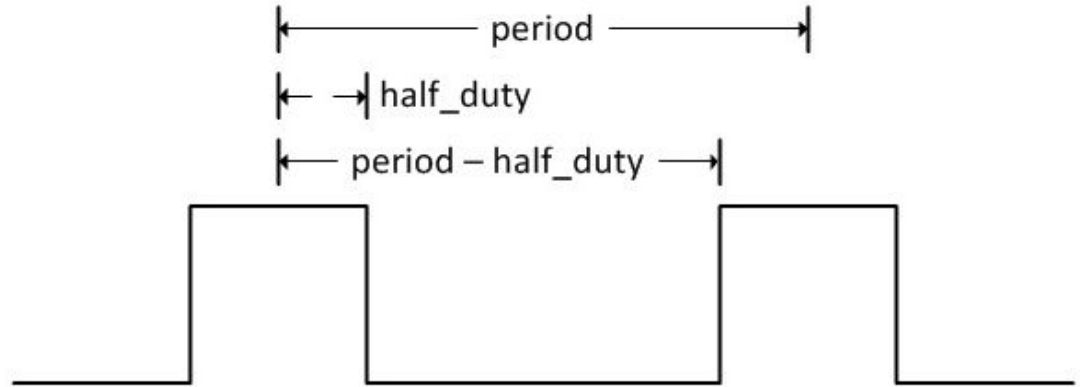
- Duty Cycle = $T_{on} / (T_{on} + T_{off})$
- $F(\text{sys}) (\sim\text{MHz}) \gg F(\text{pwm}) (\sim\text{KHz})$
- Número de pulsos del clock dentro de un período:
 - $T(\text{pwm}) = \text{Clk Cnt} = F(\text{sys}) / F(\text{pwm})$
- Dado una resolución N en bits, se calcula la cuenta de duty cycle:
 - $\text{Duty Cycle Cnt} = \text{Clk Cnt} * \text{Duty Cycle}(\text{input}) / (2^N)$

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_arch 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

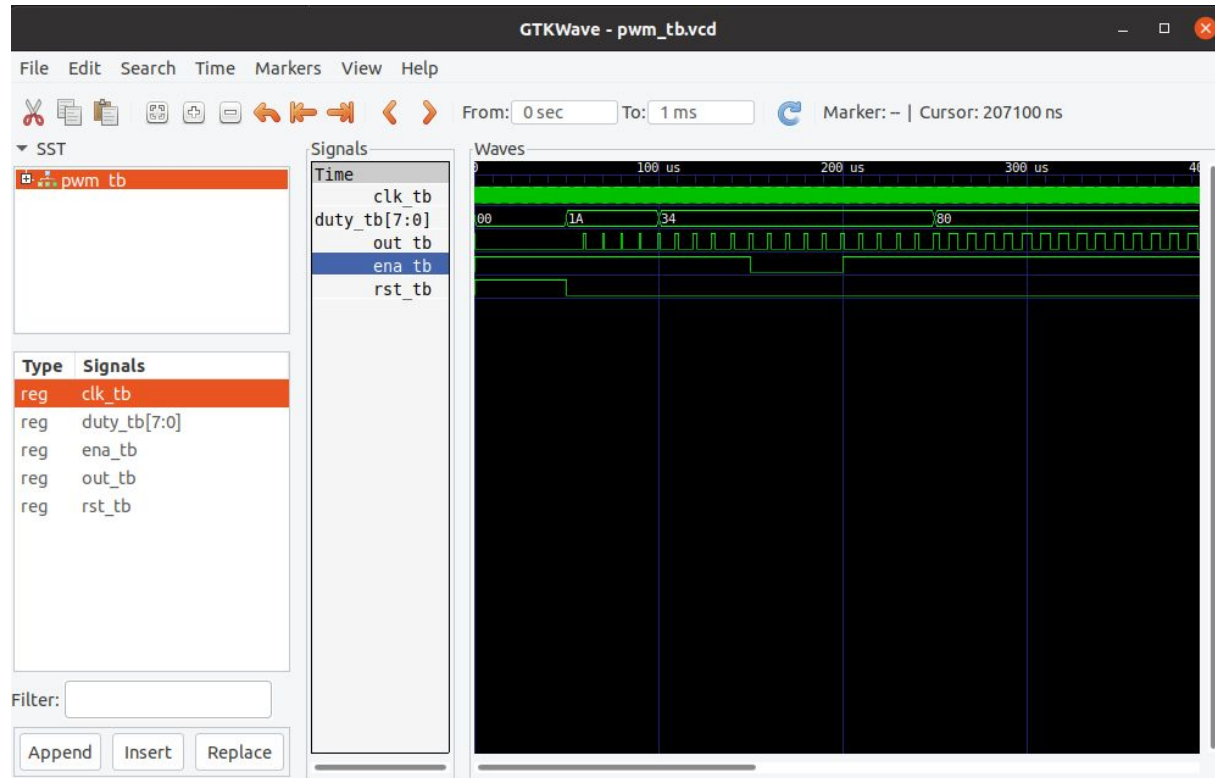
```
begin

    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;

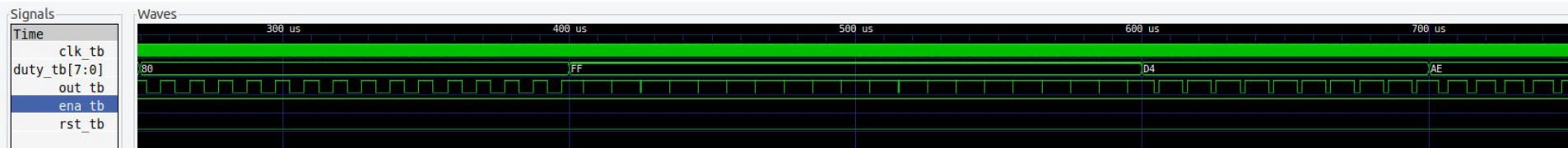
    DUT: pwm
    generic map(
        sys_freq => sys_freq_tb,
        pwm_freq => pwm_freq_tb,
        pwm_bits => pwm_bits_tb
    )
    port map(
        clk_i => clk_tb,
        rst_i => rst_tb,
        ena_i => ena_tb,
        duty_i => duty_tb,
        out_o => out_tb
    );

end;
```

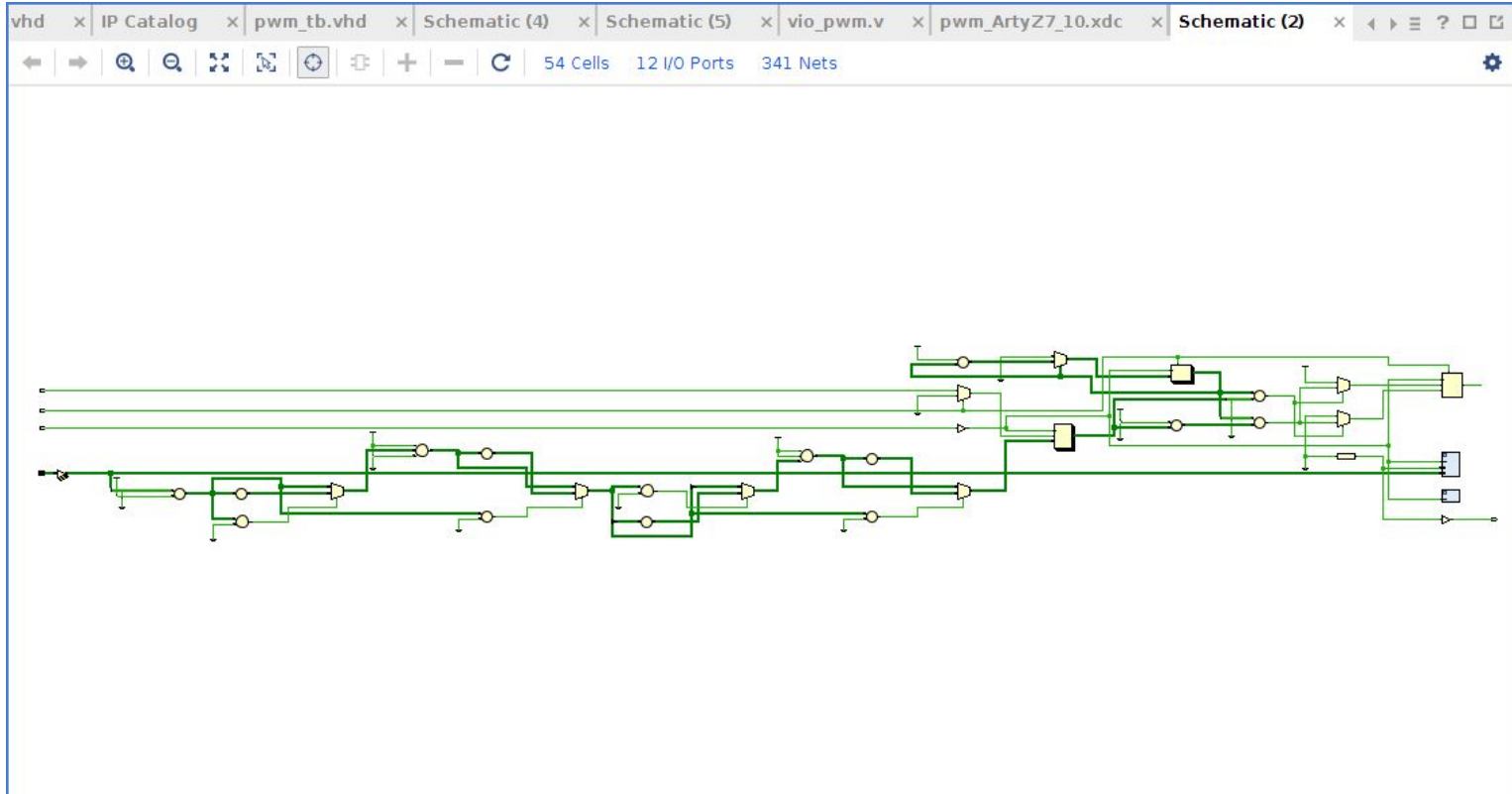
Simulación



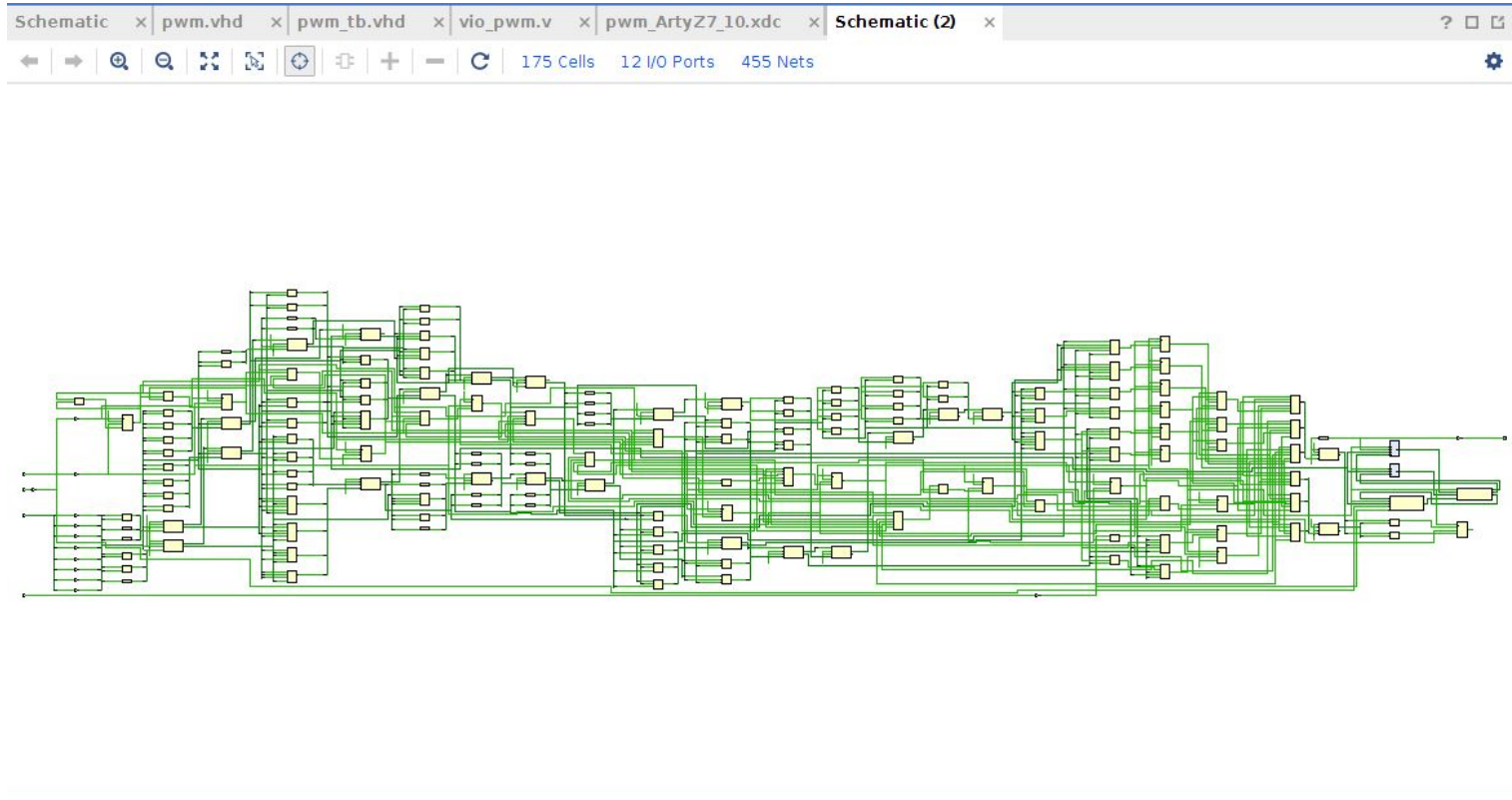
Simulación



Vivado: Diseño Elaborado



Vivado: Síntesis



Vivado: Constraints

```
1 #####
2 # Arty Z7 Pin Assignments
3 #####
4
5 # Duty Cycle
6 set_property -dict {PACKAGE_PIN T14 IOSTANDARD LVCMOS33} [get_ports {duty_i[0]}]
7 set_property -dict {PACKAGE_PIN U12 IOSTANDARD LVCMOS33} [get_ports {duty_i[1]}]
8 set_property -dict {PACKAGE_PIN U13 IOSTANDARD LVCMOS33} [get_ports {duty_i[2]}]
9 set_property -dict {PACKAGE_PIN V13 IOSTANDARD LVCMOS33} [get_ports {duty_i[3]}]
10 set_property -dict {PACKAGE_PIN V15 IOSTANDARD LVCMOS33} [get_ports {duty_i[4]}]
11 set_property -dict {PACKAGE_PIN T15 IOSTANDARD LVCMOS33} [get_ports {duty_i[5]}]
12 set_property -dict {PACKAGE_PIN R16 IOSTANDARD LVCMOS33} [get_ports {duty_i[6]}]
13 set_property -dict {PACKAGE_PIN U17 IOSTANDARD LVCMOS33} [get_ports {duty_i[7]}]
14
15 # CLK source 50 MHz
16 set_property -dict {PACKAGE_PIN H16 IOSTANDARD LVCMOS33} [get_ports clk_i]
17
18 # Rst Btn
19 set_property -dict {PACKAGE_PIN L19 IOSTANDARD LVCMOS33} [get_ports rst_i]
20
21 # Ena Btn
22 set_property -dict {PACKAGE_PIN D19 IOSTANDARD LVCMOS33} [get_ports ena_i]
23
24 # PWM
25 set_property -dict {PACKAGE_PIN Y18 IOSTANDARD LVCMOS33} [get_ports out_o]
26
27 #####
28 # Arty Z7 Timing Assignments
29 #####
30
31 # define clock and period
32 create_clock -period 20.000 -name clk_pin -waveform {0.000 10.000} [get_ports clk_i]
33
34
```

Vivado: Implementación y Depuración

The screenshot displays the Vivado IDE interface for a project named `project_pwm`. The top menu bar includes File, Edit, Flow, Tools, Reports, Window, Layout, View, and Help. The top status bar indicates the project path and the Vivado version (2018.1).

The left sidebar contains the following sections:

- SIMULATION**
 - Run Simulation
- RTL ANALYSIS**
 - Open Elaborated Design
 - Report Methodology
 - Report DRC
 - Schematic
- SYNTHESIS**
 - Run Synthesis
 - Open Synthesized Design
 - Constraints Wizard
 - Edit Timing Constraints
 - Set Up Debug
 - Report Timing Summary
 - Report Clock Networks
 - Report Clock Interaction
 - Report Methodology
 - Report DRC
 - Report Utilization
 - Report Power
 - Schematic
- IMPLEMENTATION**
 - Run Implementation
 - Open Implemented Design
- PROGRAM AND DEBUG**
 - Generate Bitstream
 - Open Hardware Manager
 - Open Target
 - Program Device
 - Add Configuration Memory Dev

The main workspace is divided into several panels:

- Hardware Manager**: Shows the hardware components and their status. The `hw_ila_2` component is selected, showing its properties (Name: `hw_ila_2`, Cell: `u_ila_0`, Device: `xc7z010_1`, HW core: `core_6`, Capture sample count: 0 of 1024, Core status: Idle).
- Debug Pro**: Shows the debug probes and their status. The `hw_ila_2` component is selected, showing its properties (Name: `hw_ila_2`, Cell: `u_ila_0`, Device: `xc7z010_1`, HW core: `core_6`, Capture sample count: 0 of 1024, Core status: Idle).
- Waveform - hw_ila_2**: Shows the waveform for the `hw_ila_2` component. The waveform is currently idle, with a red vertical line indicating the current time (0).
- Tcd Console**: Shows the console output. The output indicates that the synthesis was successful and that the hardware was programmed. The console output includes the following information:

```
INFO: [Common 17-83] Releasing license: Synthesis
18 Infos, 11 Warnings, 2 Critical Warnings and 0 Errors encountered.
synth_design completed successfully
synth_design: Time (s): cpu = 00:00:35 ; elapsed = 00:00:48 ; Memory (MB): peak = 1701.535 ; gain = 478.172 ; free physical = 1725 ; free virtual = 4370
INFO: [Common 17-1381] The checkpoint '/home/lauro/CESE/4_Bimestre/CLP/Trabajo Final/project_pwm/project_pwm_runs/synth_1/pwm.dcp' has been generated.
INFO: [runctl-4] Executing: report_utilization -file pwm_utilization_synth.rpt -pb pwm_utilization_synth.pb
```