PWM module implementation using Mealy FSM

Name - Arpit Paul
Dept. - Electronics and Communication Dept.
Year - 4th Year
College - National Institute of Technology, Sikkim

Introduction

Pulse Width Modulation (PWM) is a widely used technique in electronic systems for controlling the power supplied to electrical devices, such as motors, LEDs, and speakers. It works by rapidly switching a digital signal on and off to create an average voltage level, effectively simulating an analog output. PWM is popular because it offers precise control over the amount of power delivered to a load, making it ideal for applications where varying the output voltage or current is necessary.

Why is my PWM module different and unique from others in this project?

1. The use of **Mealy FSM** makes it different from others as for traditional use cases of block implementation.

2. Input controlled duty cycle means which duty cycles the users need is simply given to the input side to get a fruitful PWM pulse.

3. Space and power optimized design to get user defined PWM pulse, efficient space - power trade off.

4. Using **Gate Clocking** makes it more power efficient and proper resource utilizable.

Application of PWM Module

1. **Motor Speed Control:** PWM is widely used in motor speed control applications. By adjusting the duty cycle of the PWM signal, the average voltage applied to the motor can be controlled, thus regulating its speed. This is employed in devices like fans, electric vehicles, and robotics.

2. **LED Dimming:** PWM is commonly used in LED lighting systems for brightness control. By rapidly turning the LED on and off with varying duty cycles, the apparent brightness of the LED can be adjusted. This technique is found in LED bulbs, displays, and backlighting systems.

3. **Audio Amplification:** PWM is used in Class D audio amplifiers. These amplifiers use PWM signals to modulate the output and then filter the signal to reconstruct the amplified audio. They are more energy-efficient compared to traditional Class A, B, or AB amplifiers and are used in applications like portable speakers and high-efficiency home audio systems.

4. **Temperature Control:** PWM can be used in temperature control systems. For instance, in a heating system, the duty cycle of the PWM signal can control the power supplied to the heater, maintaining a desired temperature. Similarly, in cooling systems, it can control the speed of fans or compressors based on the cooling requirements.

5. **Servo Motors:** Servo motors in robotics and automation use PWM signals to determine the position of the motor shaft. By varying the pulse width, the servo motor can be accurately positioned, making it valuable in applications like robotic arms and remote-controlled devices.

6. **Solar Charge Controllers:** PWM-based solar charge controllers regulate the charging of batteries in solar power systems. They control the charging current and voltage to ensure efficient charging and prevent overcharging of batteries, thus prolonging battery life.

7. **Haptic Feedback:** In devices like smartphones and game controllers, PWM is used to create haptic feedback. By modulating the vibration motor with PWM signals, different vibration patterns and intensities can be produced, enhancing user experience through tactile feedback.

Verilog Code (Design code)

```
`timescale 1ns / 1ps

module pwm(
   dout, clk, rst, din
   );


 output reg dout;
 input clk, rst;

 input [3:0] din;

 parameter idle=0, s0=1, s1=2;


 reg [1:0] ps;                                    ///////Mealy FSM implementation

 integer i=0;

 always @ (posedge clk)
 begin

   if(rst)
     ps <= idle;

   else
     begin

       case(ps)
        idle: begin ps <= s1;
                dout<=1'b0;
             End                                   //////1 process methodology
         s1: begin
             if(i < din)
               begin
                ps <= s1;
```

```
               i <= i+1;
               dout<=1'b1;
               end
             else
               begin
               ps <= s0;
                i <= 0;


               end
         end

        s0: begin
             if(i < (10-din))
              begin
              ps <= s0;
              i <= i+1;
              dout<=1'b0;

              end
              else
               begin
               ps <= s1;
                i <= 0;

               end
         end
       endcase
    end
end
endmodule
```

Verilog Code (Testbench Code)

```
`timescale 1ns / 1ps

module tb;
  wire dout;
  reg clk, rst;
  reg [6:0] din;

  pwm dut(dout, clk, rst, din);

    initial begin

    rst =0;
    #10;
    rst =1;
    #15;
    rst=0;


    end

    initial begin
     din  = 7'd2;           /////// duty cycle : 20 %
   //  din  = 7'd5;          /////// duty cycle : 50 %
   //  din  = 7'd8;          /////// duty cycle : 80 %

    end

   initial begin
    clk = 0;

    forever #10 clk = ~clk;

    end

Endmodule
```

RTL design and block diagram

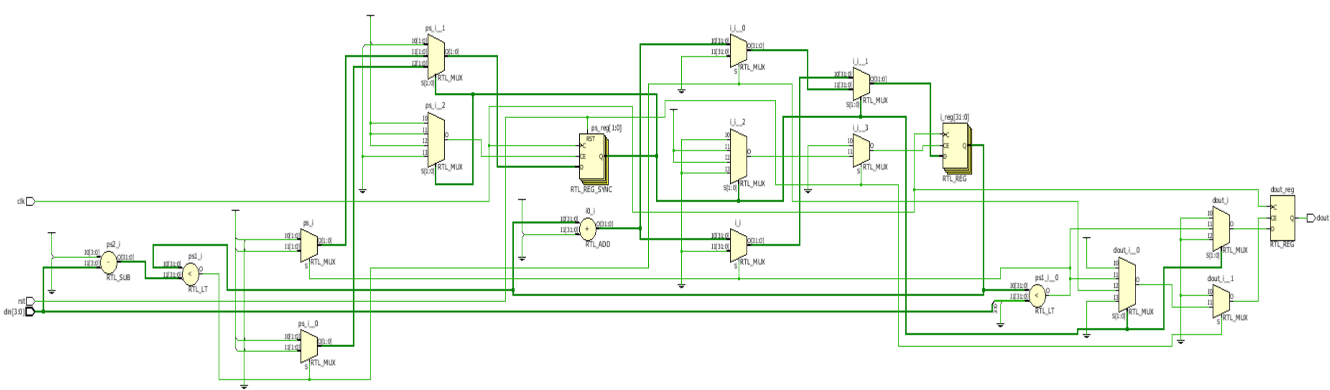

Fig. 1 - RTL Design for PWM Module

Synthesized Design

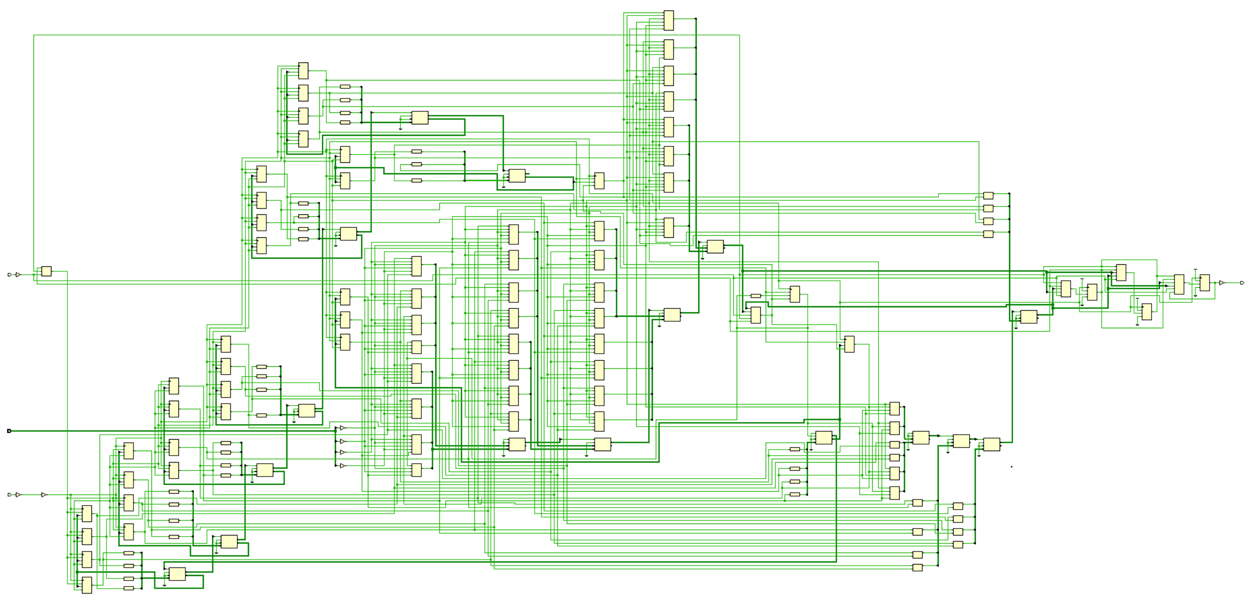

Fig. 2 - Synthesized Design for PWM Module

Timing Diagram

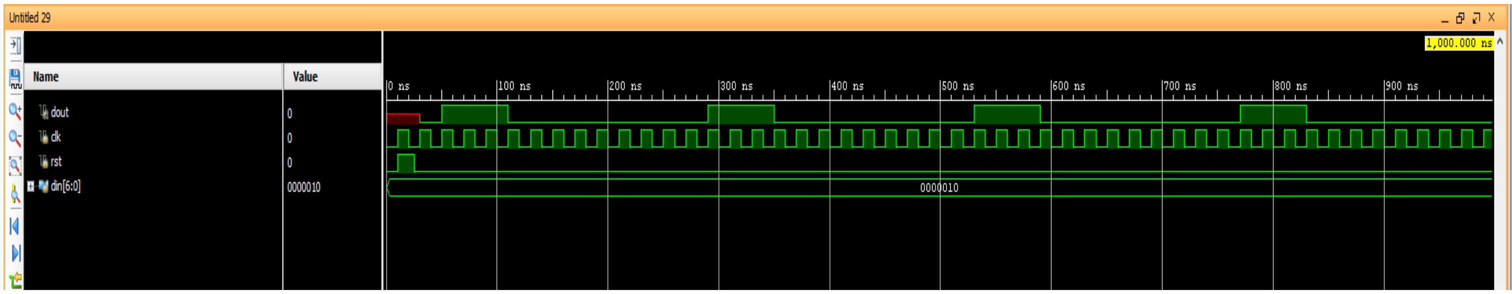

Fig. 3 - 20% duty cycle

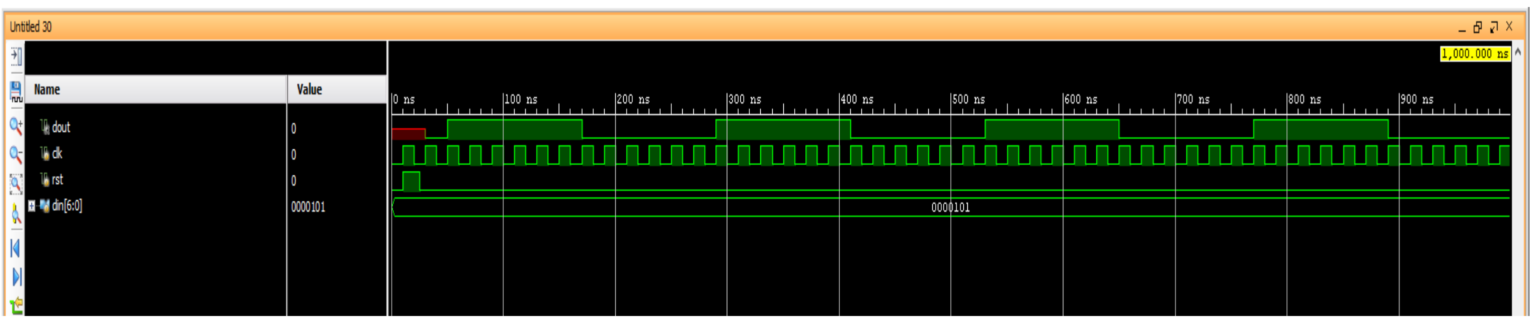

Fig. 4 - 50% duty cycle

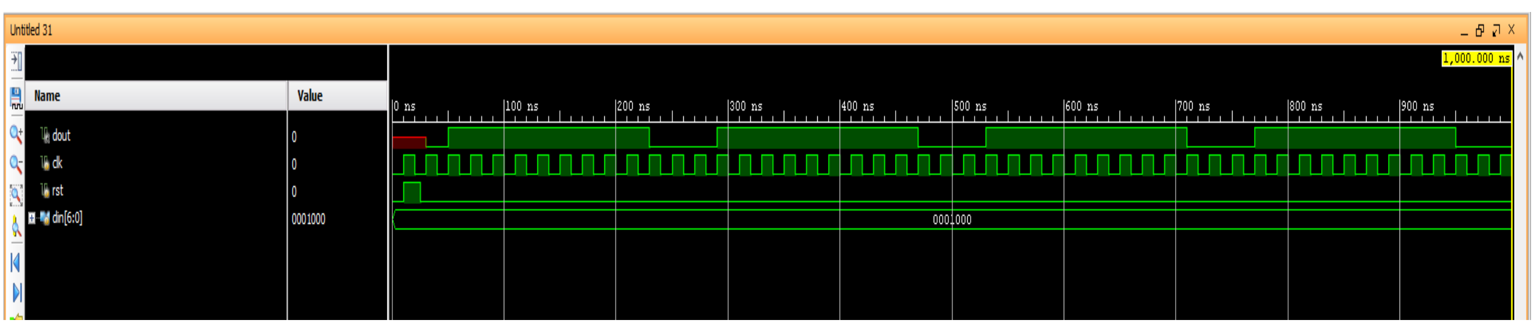

Fig. 5 - 80% duty cycle

Power Utilization of PWM

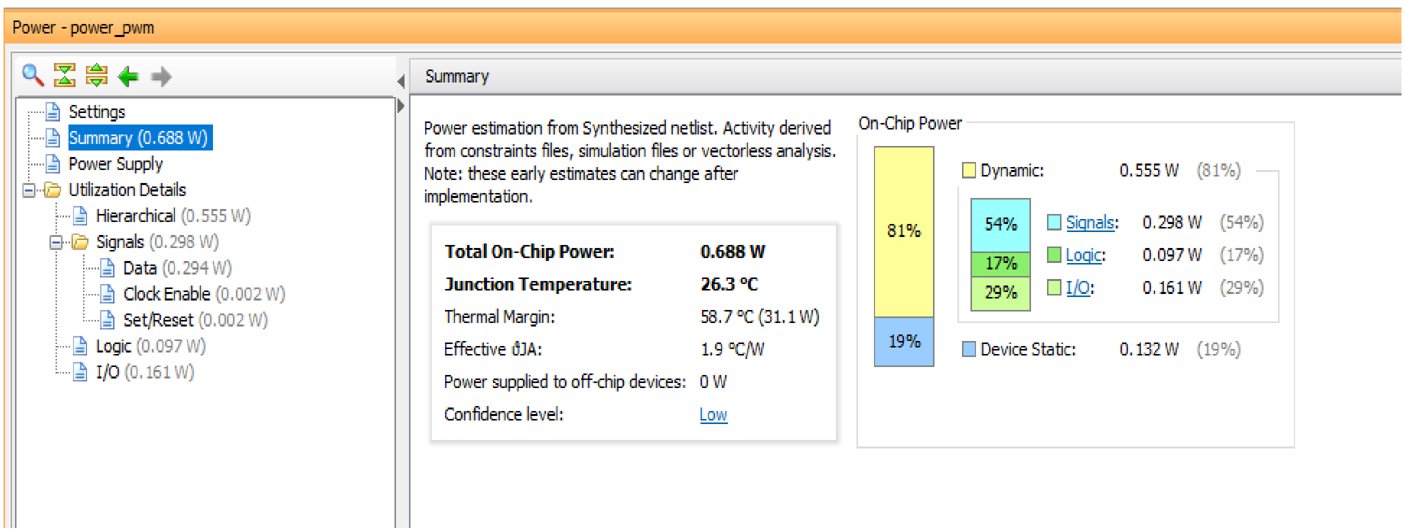

Fig. 6 - Power utilization chart

Resource utilization of PWM

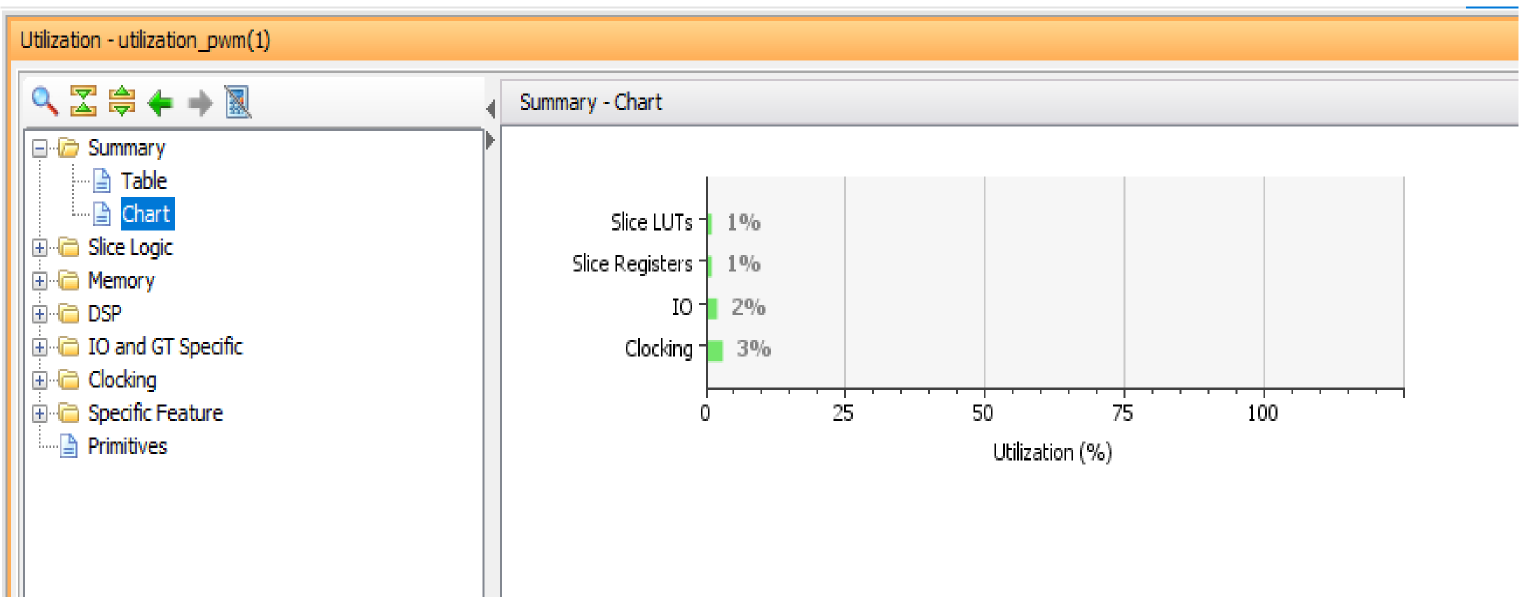

Fig. 7 - Resource utilization chart

************End of the Project************