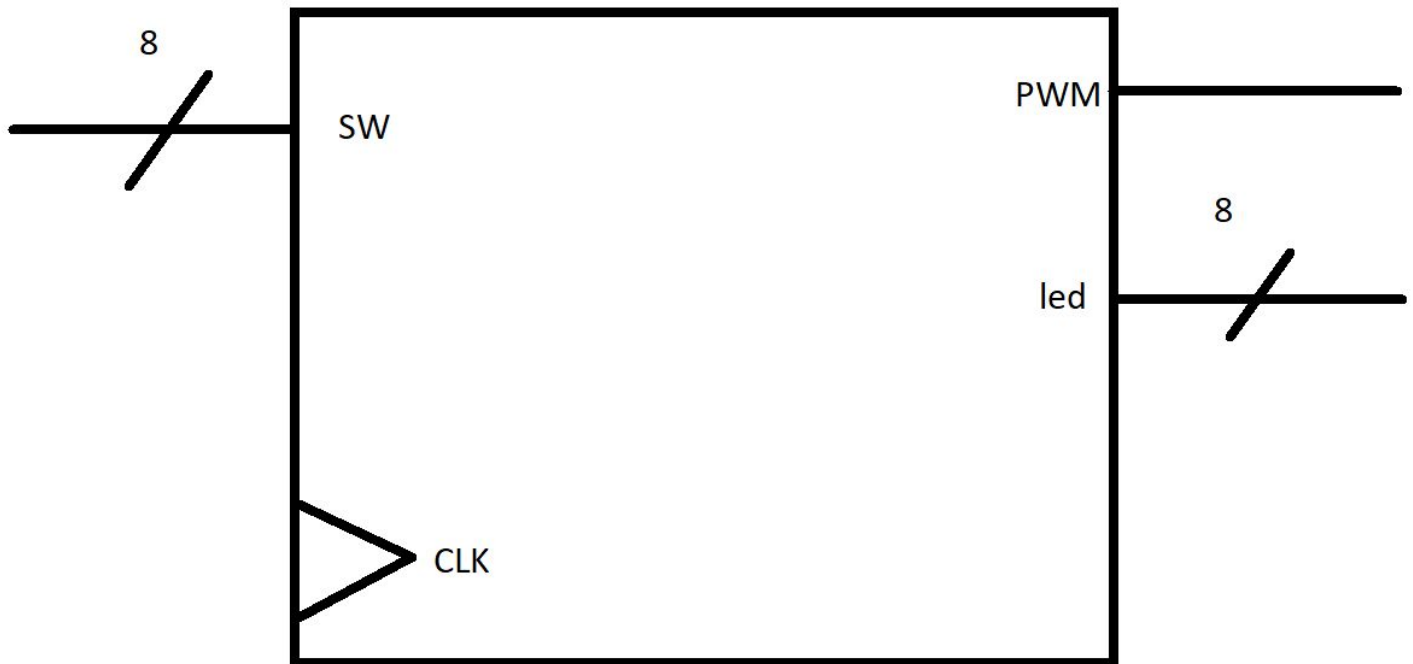CPE 233: Peripheral assignment 2, DAC

Prof. Bridget Benson

Luis Gomez, Matt Bailey

BBD

Behavior

The DAC or Digital to Analog Converter is comprised of an external circuit connected to a PWM generator (loaded on the Basys 3 FPGA). Most of this report describes the design of the PWM generator, which can be described as having the following: 8-bit switch input (SW), 8-bit LED output (led), and a 1-bit PWM output signal. The PWM generator operates at ~ 392.157 kHz, and can produce duty cycles ranging from 0-100%, with 255 uniquely possibly duty cycles.

Structural

The structural design of the PWM generator for the DAC is found below.
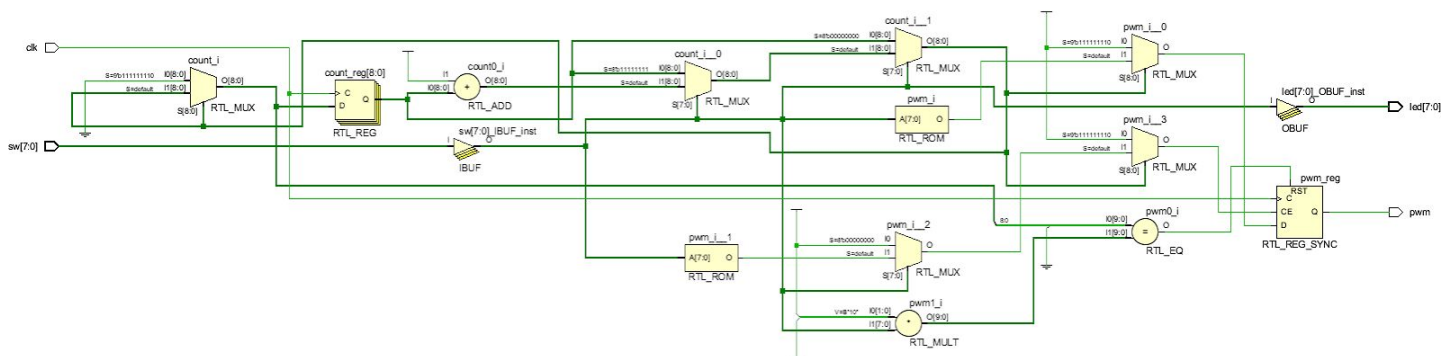
Specifications

Operating Speed: 100Mhz	**8 convenient LEDS** to indicate enabled switches
Input Clock: 100Mhz	**Output Voltage:** [0 - 3.3] V
Generates Output Voltages and Duty Cycles with **less than 5% error**	

The **Digital to Analog Converter (DAC)** is a System Verilog module which utilizes an input clock signal and an 8-bit switch input which creates a variable 8-input. This 8-bit value can be used to specify the analog voltage to output. Using 8-bits allows us to have 256 different analog voltages including 0V, to be output with a resolution of:

$$\frac{3.3V}{255} = 0.013V$$

The **Pulse Width Modulation (PWM) generator** is a System Verilog module which utilizes an input clock signal and our 8-bit switch input to vary the width of the high or low of a square wave without changing the frequency of the pulses. This changes the signals duty cycle, or the percent of the time it is low v. high. For example if we have a 0% duty cycle it will always be low, where a 100% duty cycle will be always high. Once again with our 8-bit input we are allowed 256 different cycles, with a 100 MHz internal clock for the Basys 3 Board we can calculate out frequency.

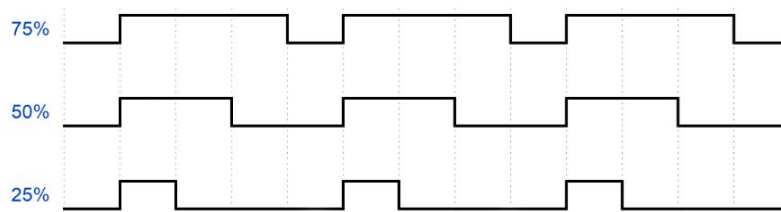$$\frac{100Mhz}{255} = 392.157\ kHz$$

Figure 1 : Examples of Pulse Width Modulation

Our **Low Pass Filter** take the square wave output from our PWM generator will cause the capacitor to charge and discharge on the high and low cycles. We built the low pass filter with a 1.6kΩ and a capacitor of 100nF assembled in ther configuration show in Figure 2. The output of the low pass filter will be a steady voltage rating based upon the duty cycle of the square wave being input.
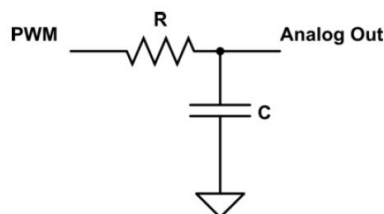

Figure 2: RC Low Pass Filter

Verification

	Expected		Measured		% error	
Input (sw)	ΔV out	Duty Cycle %	ΔV out	Duty Cycle %	ΔV out	Duty Cycle %
1	0.013	0.3921568627	0.0134	0.41	3.076923077	4.55
13	0.169	5.098039216	0.1689	5.11	0.05917159763	0.2346153846
31	0.403	12.15686275	0.401	12.17	0.4962779156	0.1080645161
127	1.651	49.80392157	1.645	49.81	0.3634161114	0.01220472441
128	1.664	50.19607843	1.658	50.21	0.3605769231	0.027734375
200	2.6	78.43137255	2.593	78.44	0.2692307692	0.011
254	3.302	99.60784314	3.294	99.60	0.2422774076	0.007874015748

Example verification, Input 13
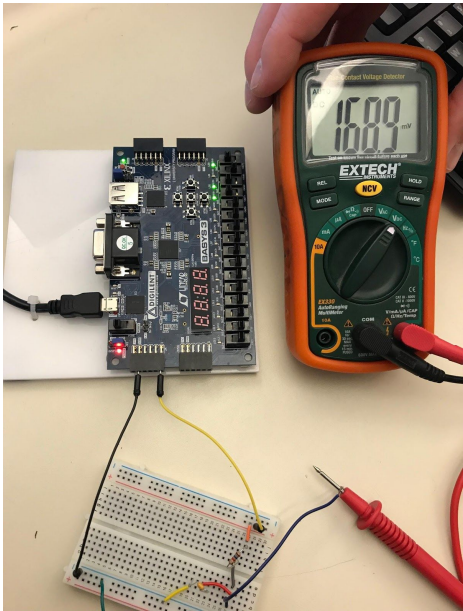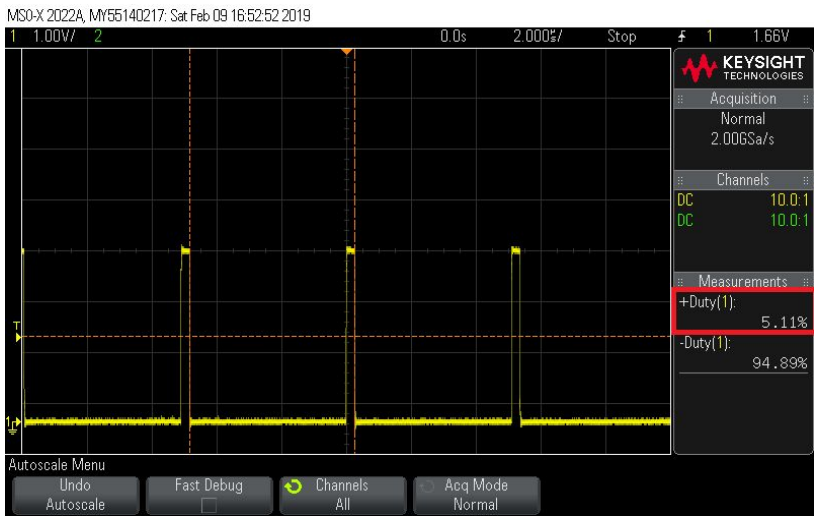
System Verilog Source Code

```systemverilog
`timescale 1ns / 1ps

module PWM(
    input clk,   // internal clock
    input [7:0] sw, // input switches
    output [7:0] led,
    output logic pwm = 0// output duty cycle signal
    );

    assign led = sw; // enables leds for switches
    logic [8:0] count = 0;
    logic [7:0] maxcount = 255; // numerical rep of half wave length

    always_ff @ (posedge clk)
    begin
        if (sw == 0)            // 0% duty cycle
            pwm = 0;
        else if (sw == 255) // % 100 duty cycle
            pwm = 1;
        else
            count = count + 1;
            if (count == 2*maxcount)
            begin
                count = 0;
                pwm = 1;
            end
            if (count == 2*sw)   // generates duty cycle
            begin
                pwm = 0;
            end
    end
endmodule
```

RAT Assembly Example Use Code

```
.EQU PWM_OUT = 0x03 ; pwm port on fpga
.CSEG
.ORG 0x01

; delay Immed -> PORT
; delay Reg -> PORT
; Immed or Reg values are 8-bit inputs to the PWM generator
; Registers Used:
; R1 -> stores value passed to delay function

    IN    R1, 0x00
    CALL delay 0xFF
    CALL delay R1
```