UNIVERSITY OF TEHRAN

School of Electrical and Computer Engineering

Digital Logic Laboratory, ECE 045, Fall 1392

Experiment 2 (Sessions 3, 4)

Function Generator

Purpose

In this lab, you will learn about:

- Register Transfer Level design and implementation
- Hardware realization of mathematical equations
- The operation and design of a Pulse Width Modulator
- The use of Passive Low Pass Filter for producing analog outputs

Background Information

In this experiment you will design a Function Generator. Function Generator is an electronic equipment used to generate different types of electrical waveforms over a wide range of frequencies. Some of the common waveforms produced by the function generator are the sine, square, saw-tooth and arbitrary shapes. Schematic of a function generator is shown in Figure 1.

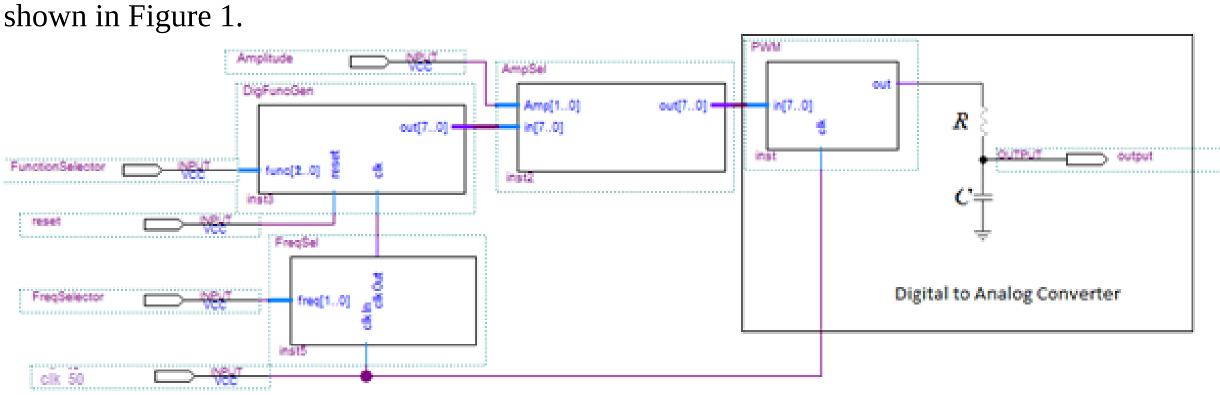

Figure 1 Schematic of the function generator

Pulse Width Modulation Unit (PWM)

PWM is a commonly used technique for controlling power to electrical devices. The average value of voltage fed to the load is controlled by turning the switch between supply and load on and off at a fast pace. **Period** of PWM has to be small enough so the effect of on/off switching be unnoticeable on load. **Pulse Width** (PW) determines amount of power that is fed to load. In this experiment period of PWM is fixed to 256 clocks and its pulse width is the value on 8-bit input of module. Figure 2 shows sample PWM waves. In each cycle first ‘PW’ clocks output value is ‘1’ and it is ‘0’ for rest of cycle.

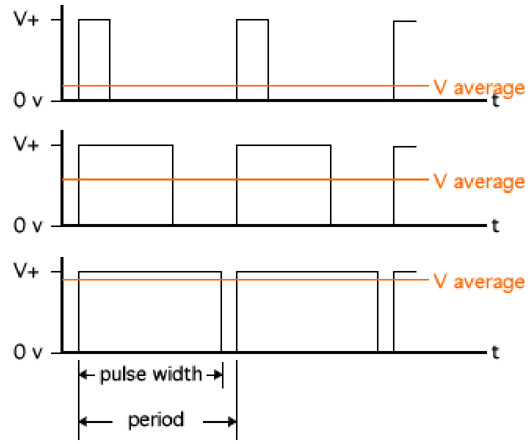

Figure 2 PWM wave

Digital to Analog Converter (DAC)

DAC is a device that converts a digital (usually binary) code to an analog signal (in this case voltage). Signals are easily stored and transmitted in digital form, but a DAC is needed for the signal to be recognized by human senses or other non-digital systems. DACs can be implemented with different methods. The easiest one is the PWM method. A stable voltage is switched into a low-pass analog filter with a duration determined by the digital input code. A passive low pass filter is shown in Figure 3.

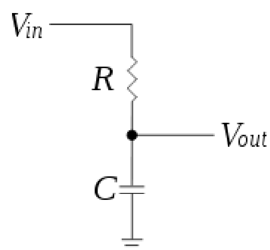

Figure 3 Passive single order low pass RC filter

In this implementation resolution of DAC is equal to resolution of the PWM and its maximum sampling rate is switching frequency (1/period) of PWM. Output value of this DAC can be approximated by Duty Cycle of PWM signal multiplied by Vcc value. Figure 4 shows output of the DAC according to state of PWM signal.

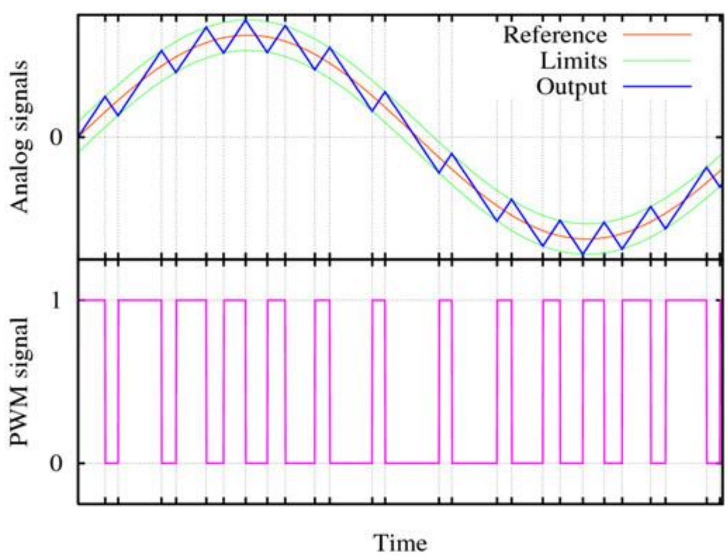

Figure 4 Output of DAC vs. PWM signal

Digital Function Generator

This module is the heart of project. It produces desired functions over time. Functions generated by this module have the fixed period of 256 clocks. Output of this module is an 8-bit digital representing amplitude of signal. It should be noted that due to characteristics of implemented DAC valid range on output is between Vcc and Gnd. Supported functions are sine, square, triangle and saw-tooth. Waveforms for mentioned functions are shown in Figure 5.

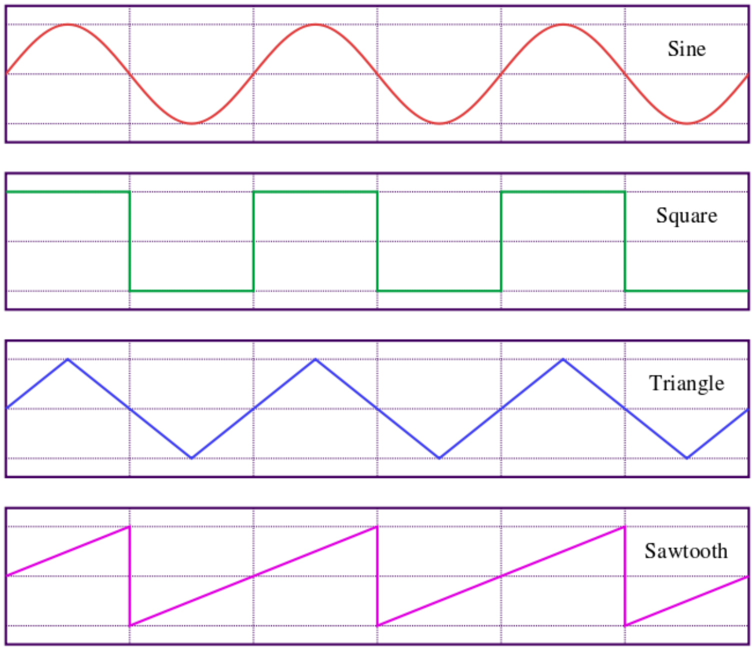

Figure 5 Sine, square, triangle, and sawtooth waveforms

Generation of Sine function can be difficult. The following second order differential equation can be used to generate Sine function:

$$\sin(n) = sin(n-1) + \ a\, cos(n-1)$$
$$\cos(n) = \cos(n-1) - \ a \sin(n)$$

In order to do the mathematical operations in reasonable accuracy; operations are done in 16-bit fixed point. Also considering the period of about 256 clocks equation turn to (assuming value between -32768 to 32767 for Sin and Cos):

$$\sin(n) = sin(n-2) + \frac{1}{32} cos(n-1)$$
$$\cos(n) = \cos(n-2) - \frac{1}{32} \sin(n-1)$$

Initialization of first values in differential equations is necessary (Use ‘0’ for Sin(0), ‘510’ for Sin(1), ‘30000’ for Cos(0) and ‘29700’ for Cos(1)). It should be noted that the result of Sinus and Cosine operations is Signed(-127 to +128), but since your PWM is not able to generate negative values, you should bias the output of these operations, and change its range to 0 to 255.

To Generate the arbitrary function, you will receive an input file that contains a voice data which sampled by 8Ksps (Kilo Sample per Second), you should initialize a 1-port ROM by these samples and then deign a circuit that read these samples and listen to it by forwarding them to a speaker. It should be noted that ROM is a **Read only Memory** (ROM) that is used as a class of storage medium. Data stored in ROM cannot be generally modified.

Type of function is selected by a 2-bit input according to Table 1.

Table 1 Function Selection

Func[2:0]	Function
3'b000	Sine
3'b001	Square
3'b010	Triangle
3'b011	Sawtooth
3'b100	Arbitrary

Amplitude Selector and Frequency Selector

One option in function generator is the amplitude of generated wave. This can be done by adding a module between digital function generator and DAC. Task of this module is to divide the input taken from digital function generator and divide it by a desired value and feed the result to DAC. Value of divisor is chosen by a 2-bit input.

Type of amplitude is selected by a 2-bit input according to Table 2.

Table 3 Amplitude Selection

Func[1:0]	Amplitude
2'b00	1
2'b01	2
2'b10	4
2'b11	8

Another option in function generator is the frequency of generated function. This can be done by dividing input clock frequency of digital function generator to adjustable values. Value of divisor is chosen by a 2-bit input. Since the period of Arbitrary Function is unknown use the "Frequency" parameter rate of samples that are read from ROM. (eg. 8KSPS for Func = 2'b00)

Type of frequency selector is chosen by a 2-bit input according to Table 3.

Table 3 Frequency Selection

Func[1:0]	Frequency (KHz)
2'b00	8
2'b01	16
2'b10	32
2'b11	64

Pre-Lab assignment

Before coming to the lab answer these questions. The pre-lab needs to be handed in at the start of the lab.

1. Describe why it is not a good idea to set divisor value of Frequency Module to a value less than 256.

2. Draw the corresponding Register Transfer Level schematic for PWM.
3. Draw the corresponding Register Transfer Level schematic for digital function generator module (Except the Arbitrary function generator part).

Lab Work

Read the instructions carefully while doing the lab. Your report must include the procedure you followed, as well as any observation and results.

1. Design the PWM module with Verilog and implement the DAC. Verify it's correctness on oscilloscope (Use LCD pins on board for output and Gnd).
2. Implement Frequency and Amplitude Selector modules with Verilog and verify their correctness with functional simulation.
3. Create the digital function generator with Verilog and connect it to other parts of system according to Figure 1. Verify its correctness on oscilloscope.