

# Implementing a Tone Generator on an IGLOO<sup>®</sup> FPGA

## 1.0 Introduction

The main objective of this design is to generate tones of desired frequency, duty cycle, and duration. The control inputs are available through hardware signals or through a serial interface.

## 2.0 General Implementation Overview

This design uses a 16-bit PWM logic block which has a programmable tone period (frequency), tone duty cycle (volume), 6-bit tone duration counter (with a 2-bit prescaler), and tone damping features (6-bit damping counter with a 2-bit prescaler). The 2-bit prescaler allows for a wide range of duration of the tone output.

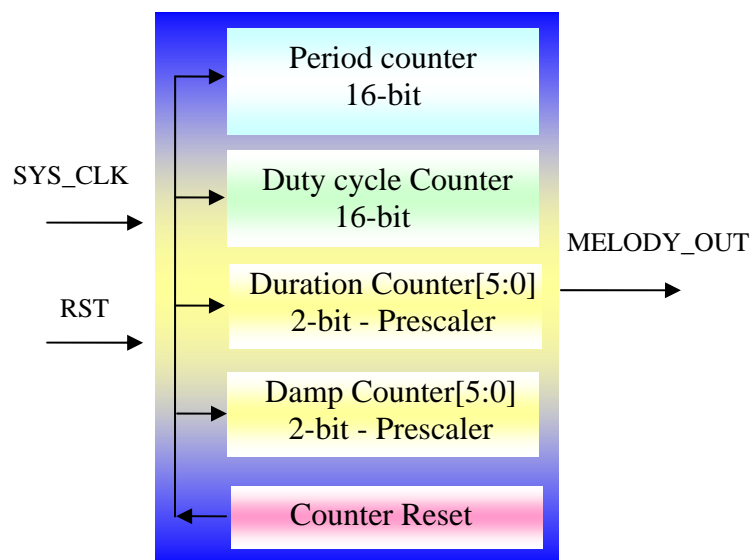
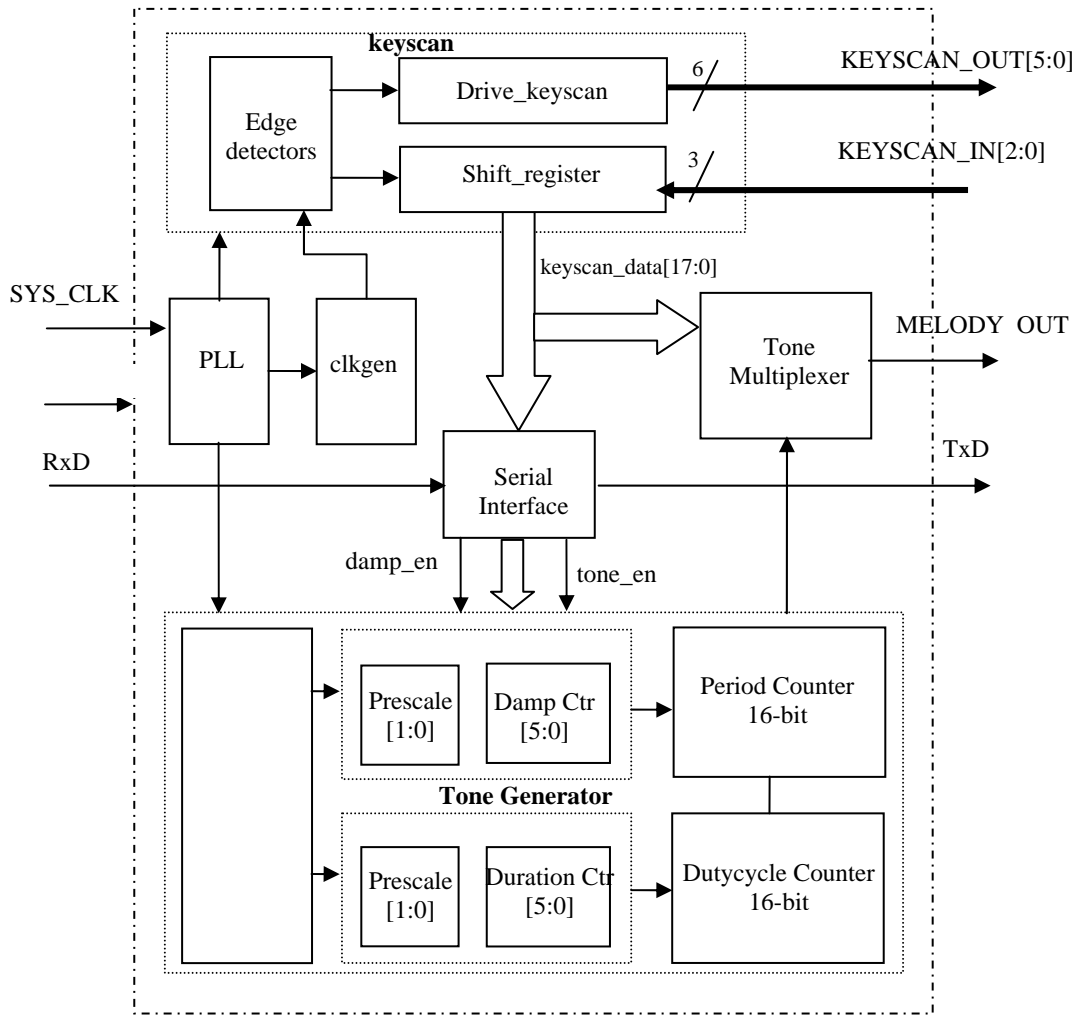


Figure 1. Tone Generator IP Block

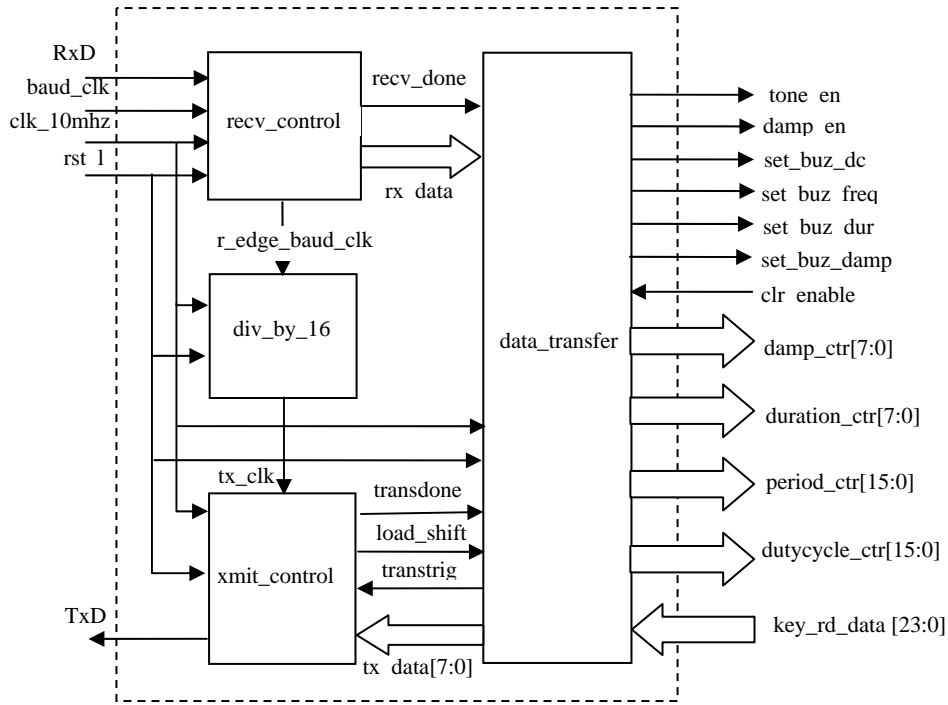
## 3.0 Reference Design Implementation

The pulse width modulation circuit works with two counters: the pulse period counter and the pulse width counter. The pulse period counter determines the period of the output pulse, which divides down the melody generator base clock accordingly. The pulse width counter determines the width of the pulse and it divides the same base clock according to the counter's value. To operate properly, the value stored in the pulse period counter must be larger than the value in the pulse width counter. During initial starts or when the pulse period counter overflows, the tone output will be set to a high level. The tone output level will reset to a low level whenever the pulse width counter overflows. Both are down counters and they will reload to preset values whenever the pulse period counter overflow occurs. The duration and damping counters are 6-bit counters, each with a 2-bit prescaler option to give enough flexibility in tone duration and damping time (if damping is enabled).



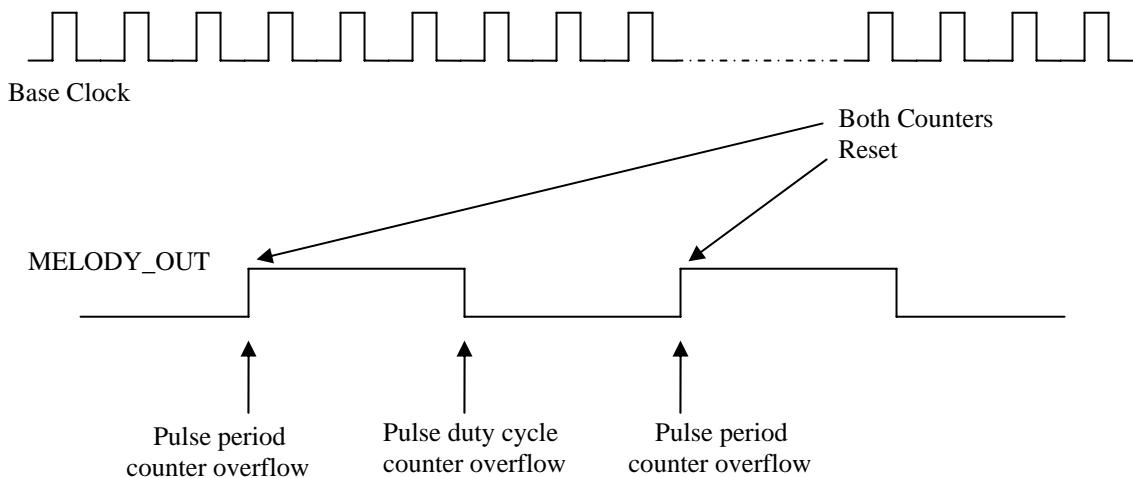
The keypad logic is provided in order to generate tones of fixed frequencies based on a particular key press event.

The serial interface block (Figure 3) for the tone\_ip is designed to control the various parameters of the tone\_generator, such as the duty cycle value, period value, duration, and damp values, along with the control features such as tone enable, damp enable, etc. The block diagram below shows the serial interface logic using the RxD and TxD signals. These signals can be connected to the USB-to-RS232 driver (CP210X) on the Icicle board.



**Figure 3. Serial Interface (tone generator)**

### 4.0 Waveforms



**Figure 4. Pulse Width Modulation Waveform**

The output tone generated is a function of the period counter and duty cycle counter. The rising edge of tone output occurs when the period counter overflows. The period counter determines the period of output pulse. The tone output is set to a low level when the duty cycle counter overflows, as shown in Figure 4.

## 5.0 I/Os

The following table describes the main I/Os in the design.

Signal	Input/Output	Description
MELODY_OUT	Output	Tone Output
KEYSCAN_OUT[5:0]	Output	6 Scan Lines Output
TxD	Output	Transmit – Serial Output
RxD	Input	Receive Serial Input
HW_SW	Input	Hardware/Software selection 1 = hardware 0 = software
KEYSCAN_IN[2:0]	Input	3 Return Lines Input
SYS_CLK	Input	System Clock – 20mhz
SYS_RESET_N	Input	Master Reset active low

## 6.0 Conclusion

There are several applications where an audible signal is necessary to draw attention. This flexible tone generator module can be used to generate a sequence of tones and also create a piece of music by controlling the tone frequency and duration. For enhanced experience, tone modulation techniques along with the damping features can also be implemented.

## Appendix A – Tone Generation Controller Design Example

### Design Files Summary

Files	Functionality
clk_by_2.v	Divides input clock by 2 - Toggle F/F
clk_gen.v	Clock Generator Block
data_transfer.v	Transfers the data according to commands given by PC
debounce.v	Debounce Logic
debounce_blk.v	Interconnects all debounce blocks.
div_by_16.v	Divide by 16 block for serial communication – baud clock
divideby5.v	Derived Clock for internal use
keyscan.v	This block scans keys
mbclk_gen.v	This block generates base clock for melody generator
recv_control.v	This block receives data serially on RxD.
serial.v	This block connects xmit_control, recv_control, baud_clk_gen and div_by_16
tone_generator.v	This block generates melody out (16 bit PWM Logic)
xmit_control.v	This block transmits data serially on TxD.
tone_ip.v	This block interconnects tone_generator, serial
top_tone_ip.v	This block interconnects PLL,tone_ip
top_tb.v	Testbench for tone_ip

## About Ishnatek

Ishnatek offers FPGA design and hardware prototyping services. Ishnatek offers Design and Verification Services for embedded solutions. We have IPs such as 8031, RTC, Timers, Enhanced PWM, UART/SIO/IrDA, I2C, LED Driver and Key scan, Parallel Port, ECP/EPP, etc., which can be building blocks for your embedded controller solutions.