The University of Texas at Arlington

College of Engineering

Wave Generator

System on Chip Term Project

Submitted By

Edgar Hernandez, Angel Aguirre Dominguez

Submitted toward the partial completion of the requirements for

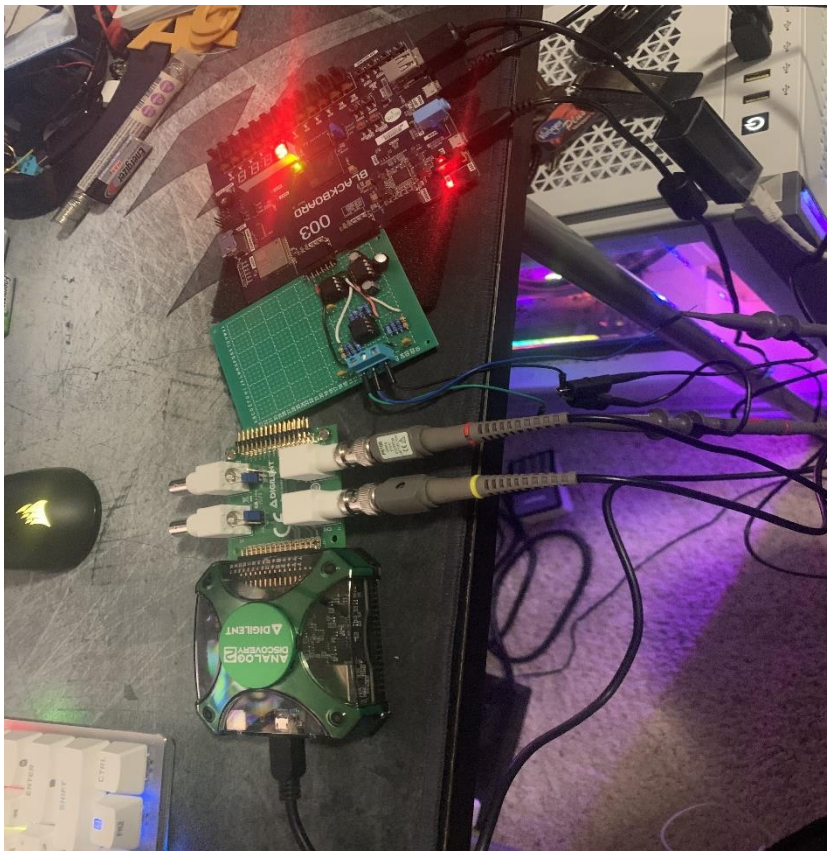
CSE 4356-001

12/04/2023

Table of Contents

Introduction

The wave generator is an FPGA peripheral that drives two channel signals from –2.5V to 2.5V corresponding to characteristics set by the user. The goal is to output one of five waveforms (sine, square, triangle, sawtooth, and DC) with specified frequencies, amplitudes, offsets, duty cycles, and cycles. These characteristics are set by writing from a processor to the FPGA register space via an AXI4 bus. Once set, the block generates the appropriate SPI signals to write a value from 0 – 4095 to the device hardware, which consists of a DAC, a charge pump, and an OP Amp. In theory, this would have versatile use cases such as pulse width modulation with a square wave and varying duty cycles or playing different tones through a speaker with sine waves of varying frequencies.



Parts List

Part	Quantity
CL 7660 Charge Pump	1
MCP 4822 Dual 12-bit SPI DAC	1
TLV2372 Dual Rail-to-rail I/O Op Amp	1
49.9Ω, 1% resistor (output series resistor)	2
9.31kΩ, 1% resistor (offset voltage divider)	2
10kΩ, 1% resistor (input resistor from filter)	2
25.5kΩ, 1% resistor (feedback)	2
31.6kΩ, 1% resistor (offset voltage divider)	2
47 ohm, 5% resistor (reconstruction filter)	2
0.1uF capacitor (reconstruction filter, DAC bypass)	3
10uF electrolytic capacitor (charge pump)	3
8-pin DIP socket	3
2x6 0.1” pitch right angle pin header (pmod)	1
PC board (7x10cm)	1
3 position, 0.2” pitch terminal blocks	1

Theory of Operation

To use the wave generator, the register fields in the FPGA space must be configured from the Linux side. The register configurations can be seen below. To write from the Linux side, both a mem map and a kernel module were used. The mem map involved various functions to write waves in a single line. The kernel module allows you to write to each register independently by listing them each as attributes for two groups – channel a and b. For DC wave, only the offset needs to be specified. For sine, triangle, and sawtooth, the frequency, amplitude, and offset had to be set. For square, the frequency, amplitude, offset, and duty cycle had to be set. For all waves, the cycles field specifies how many periods the wave would complete before outputting 0. To rerun these cycles, the user must stop and restart the waves.

Register	Reg Offset	Bits	Field	Function
MODE (R/W)	0	2:0	MODE_A	0=DC, 1=sine, 2=sawtooth, 3=triangle, 4=square, 5=arb (optional)
		5:3	MODE_B	As above
		31:6	*reserved*	
RUN	4	0	RUN_A	1=A output running, 0=A output stopped (set and clear runs bits in the same write to align the waveforms)
		1	RUN_B	1=B output running, 0=B output stopped
		31:2	*reserved*	
FREQ_A (R/W): reset 0	8	31:0	-	Frequency of output A in units of your choosing
FREQ_B	12	31:0	-	Frequency of output B in units of your choosing
OFFSET (R/W): reset 0	16	15:0	OFFSET_A	Offset voltage of output A. Value is signed in units of your choosing.
		31:16	OFFSET_B	Offset voltage of output B. Value is signed in units of your choosing.
AMPLITUDE (R/W): reset 0	20	15:0	AMPL_A	Amplitude of output A. Value is unsigned in units of your choosing.
		31:16	AMPL_B	Amplitude of output B. Value is unsigned in units of your choosing.
DTYCYC (R/W)	24	15:0	DC_A	Duty cycle of output A units of your choosing
		31:16	DC_B	Duty cycle of output V units of your choosing
CYCLES (R/W)	28	15:0	CYCLES_A	Cycles on output A (0=continuous)
		31:16	CYCLES_B	Cycles on output B (0=continuous)

(W = write, R = read)

Results

The wave generator peripheral was fully functional, as it produced two independent wave forms. However, due to a lack of time it was missing optional features such as phase offset and an arbitrary waveform. This functionality will have to be added later when we come back to improve the project in the future.