WAVEFORM GENERATION WITH FPGA USING DIGITAL SIGNAL PROCESSING TECHNIQUES

Vincent Kwok[1], Lim Zu Liang[2]
[1]NUS High School, 20 Clementi Ave 1, Singapore 129957
[2]DSO National Laboratories, 12 Science Park Drive, Singapore 118225

Abstract

This research aims to explore the possibilities of signal generation and digital signal processing (DSP) implemented in a Field-Programmable Logic Array (FPGA) chip. Signal processing algorithms benefit from parallel computational capabilities, especially where low latency and high throughput are desired. Thus, performance-critical signal processing pipelines are commonly implemented as customised hardware blocks in either application-specific integrated circuit (ASIC) or FPGA chips, due to their phenomenal parallel processing capability and performance determinism. As FPGAs possess the benefits of customised logic blocks while allowing on-the-fly reconfiguration (i.e. modifying configuration and connections of said logic blocks), they are a preferred choice for applications where hardware revisions are inconvenient (e.g. systems deployed in the field) or impossible (e.g. remote systems).

In this study, we will generate fundamental signals and apply various signal processing techniques utilising an FPGA chip, underscoring the significance and versatility of utilising FPGA chips as a method for generating and processing signals.

1 Introduction

1.1 Background

A signal is a physical quantity whose variations convey information. Most signals found in nature are analog signals, which are continuous signals. In order for these signals to be manipulated by digital signal processing (DSP) systems, they have to first be converted to digital signals, which are discrete in time (i.e. they represent samples taken at specific instants in time) and amplitude (i.e. they have a finite set of levels from which the signal level can be chosen from), through a process known as quantisation. DSP systems are usually constructed with signal processing pipelines, in which various processing blocks can be configured to act on the signal as desired.

DSP pipelines may operate on various frequency ranges, from lower frequency audible (20Hz–20kHz) to higher frequency radio signals (>100MHz), each with several important applications. DSP plays a crucial role in audio processing pipelines, such as powerful audio transformations including room resonance correction to be conducted in real-time. DSP is also commonly employed by systems such as software-defined radios for versatile signal generation and in radars, extensively used in defence applications. Although low-cost general purpose audio signal processing DSP accelerators are widely available in the market, they are unable to meet the requirements in processing higher frequencies (such as in radios). Thus the Field Programmable Logic Array (FPGA) used to provide a more specialised and performant solution is necessary.

1.1.1 FPGA

The FPGA is a semiconductor device consisting of hardware logic blocks which can be configured after manufacture, allowing the creation of arbitrary and reconfigurable logic in hardware. Configuration is typically developed through a hardware definition language (HDL), of which Verilog and VHDL are most used. DSP pipelines are commonly deployed and developed on FPGA chips due to their favourable signal, deterministic and parallel processing characteristics — desirable for low-latency and complex signal processing. As the logic blocks in FPGA devices are easily reconfigured, they can be used in environments where rapid firmware update or debugging are required.

1.1.2 Fast Fourier Transform (FFT)

Fast Fourier transform (FFT), a faster implementation of the Discrete Fourier Transform (DFT), is a basic mathematical transformation technique underpinning a wide array of DSP pipelines. It is the process of transforming signals from the time to frequency domain, allowing frequency-based signal processing and manipulation.

Some examples of DSP pipelines which incorporate FFTs are audio equalisers, and radio frequency (RF) processing such as frequency-based filters and frequency-keyed deframing (used in radio communications).

1.1.3 Signals

As part of this research, 3 types of waveforms were generated – triangle, sinusoidal, square, each of which will be further elaborated upon as follows:

1. **Sawtooth** (Eq. 1): Non-sinusoidal signal; Either a constant upward *or* downward ramp segment followed by a transient back to the starting level of the ramp.

$$x(t) = 2\left(\frac{t}{p} - \left\lfloor \frac{1}{2} + \frac{t}{p} \right\rfloor\right) \quad (1)$$

2. **Triangle** (Eq. 2): Non-sinusoidal signal; Constant upward and downward ramp, named after the triangle shape of its waveform. Can be mathematically represented as a shifted absolute sawtooth wave:

$$x(t) = 2\left(\frac{1}{p}\left(t - \left\lfloor t + \frac{1}{2} \right\rfloor\right)\right) \quad (2)$$

3. **Sinusodial** (Eq. 3): A wave whose waveform is of the mathematical sine function, and is the most natural representation of how many things in nature change state. Signals can be reconstructed from the frequency to time domain via summation of its harmonics — sinusoidal waves of various frequencies, phase offsets and amplitudes.

$$x(t) = \sin t \quad (3)$$

2 Materials and Methods

2.1 Hardware and Software

A Nexys Video Artix-7 FPGA evaluation board was used for practical research of signal generation and processing. It contains a Xilinx Artix-7 FPGA chip, and an onboard audio codec containing a ADC for digitalising (quantisation) and a DAC for reconstruction of analog signals from their digital binary representations in the audio frequency range (20Hz-20kHz). For this application, the 24-bit resolution of the audio codec provides satisfactory signal-to-noise ratio and a sufficiently faithful reconstruction of an analog signal.

Output waveforms were generated through the FPGA and streamed to the audio codec, where they are reconstructed as an analog signal. An oscilloscope and spectrum analyser were then used to visualise and analyse the output waveforms. The FPGA configuration for signal processing and generation was developed in the Vivado development environment, with a combination of modules written in both Verilog and VHDL.

2.2 Signal Synthesis

Signal synthesis is first conducted to produce fundamental signals suitable for further processing. Each of the 4 generated signal types was written as a self-contained module in Verilog, then individually generated to be observed through an oscilloscope. The choice of waveform type can be configured through a VIO (Virtual Input/Output) IP block or GPIO switches on the evaluation board.

Generators for the triangle, sawtooth and square waveform were coded, and a signal generation IP core (which references an internal sine lookup table) was employed for generation of sinusoidal waveform.

2.3 Signal Processing/Manipulation

Following signal generation, we can run several signal processing algorithms to evaluate their performance.

2.3.1 Sinusoidal Wave Summation

Sinusoidal wave summation (a is an important technique in signal processing, used for reconstructing signals from the frequency to time domain and RF modulation, to name a few. Equation 5 below shows an example of the summation of two sine waves of frequency $f1$ and $f2$:

$$x(t) = sin(2\pi f_1 t) + sin(2\pi f_2 t) \quad (4)$$

We can explore this by summing the outputs of 2 (or more) configurable sinusoidal waveform generators.

Figure 1 illustrates the results (red) of a summation of 2 arbitrary sine waveforms (green and blue).

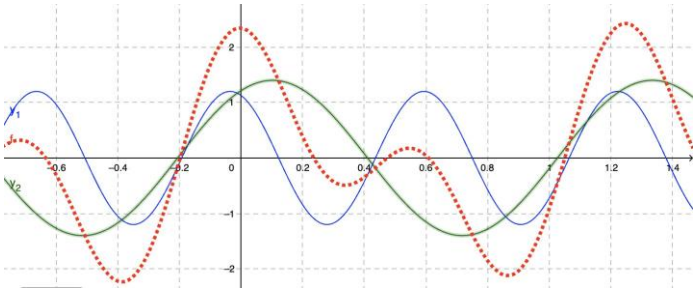


Figure 1: Summation of 2 sine waveforms

2.3.2 Pulse-Width Modulation (PWM)

Pulse-width modulation is a common signal modulation technique used to approximate an analog signal using a digital waveform. A PWM signal has a constant period (and hence frequency) while its duty cycle — that is, the proportion of time the signal is on in a period, is varied to approximate an analog signal.

PWM techniques are commonly used to control inductive loads such as motors, and approximate analog outputs thru a digital signal. Figure 2 visualises a simulation current control through an inductive load via PWM, where V is the PWM output voltage and B the current through the load.

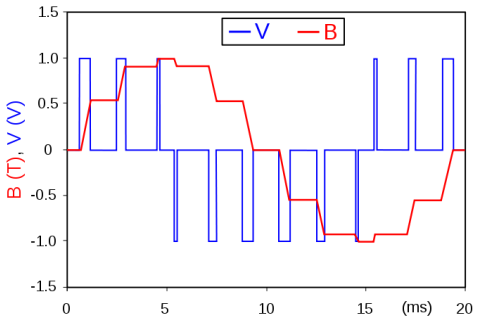


Figure 2: Current control via PWM

2.3.3 Frequency Modulation (FM)

Frequency of a signal is varied dependant on the modulation scheme and input signal.

It has seen widespread use in the field of radio communications due to its effectiveness in achieving higher signal-to-noise ratios (and in turn higher data rates) than other modulation types. A common application in the context of radio communications is frequency-shift keying (FSK), where the frequency of the carrier signal is shifted between 2 distinct frequencies to communicate binary information.

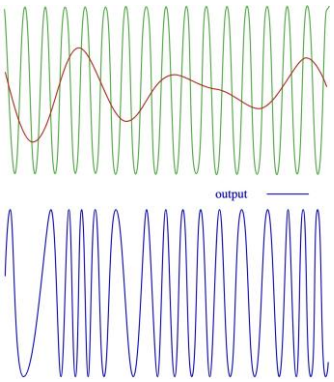


Figure 3: FM visualisation

3 Results

3.1 Fundamental Signals

The following signals are the waveforms generated by the FPGA via an audio codec on the Nexys Digilent board, captured through an oscilloscope.

3.1.1 Triangle

The following is a snippet of a Verilog module which generates a triangle wave.

```
always @(posedge clk)
begin
  if (dir) out <= out + 24'd1;
  else out <= out - 24'd1;
  if (out == -24'h7ffff0) dir <= 1'b1;
  else if (out == 24'h7ffff0) dir <= 1'b0;
end
```

The Verilog code generates the triangular waveform by ramping a signal up and down in a cyclic manner, where the direction of the ramp is dictated by the variable ”dir”.

”dir” will invert itself when the output voltage has reached its upper or lower limit.

Figure 4 is a snapshot of a generated triangle wave as visualised on an oscilloscope.

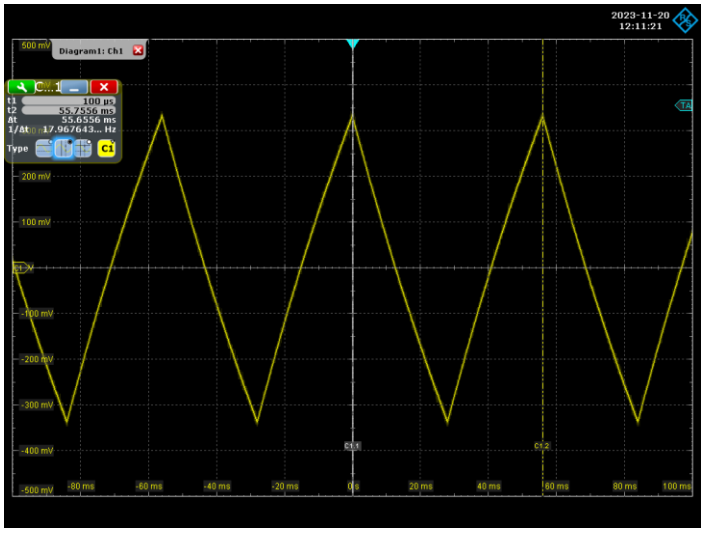


Figure 4: Oscilloscope visualisation of generated triangle wave

The generated wave had a peak-to-peak frequency of approximately 18Hz. Note that the triangle wave reproduced by the audio codec had slight distortions due to nature of its DAC.

3.1.2 Sawtooth

Refer to Figure 5 for a snapshot of a generated sawtooth waveform.

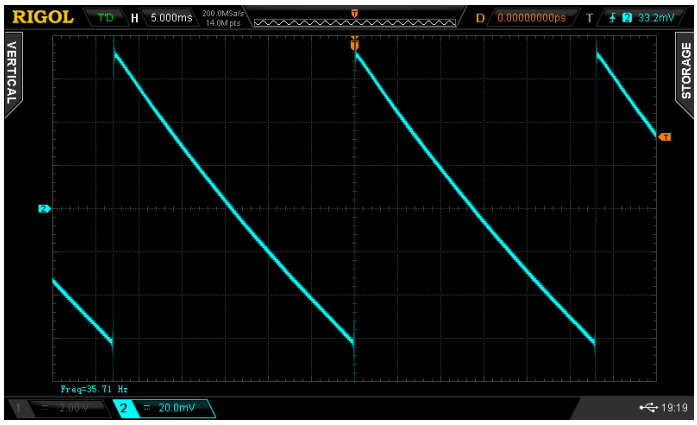


Figure 5: Oscilloscope visualisation of generated sawtooth wave

Due to the non-linearity of the DAC, it can be observed that the downwards slope of the sawtooth waveform is not ideal.

3.1.3 Sinusoidal

An IP module was used to generate sinusodial waves as follows:

```
dds_compiler dds(
  .aclk(clk),
  .m_axis_data_tdata(out)
);
```

The Direct Digital Synthesizer (DDS) is used to generate a a sinusoidal signal whose phase is auto-incremented on every clock cycle.

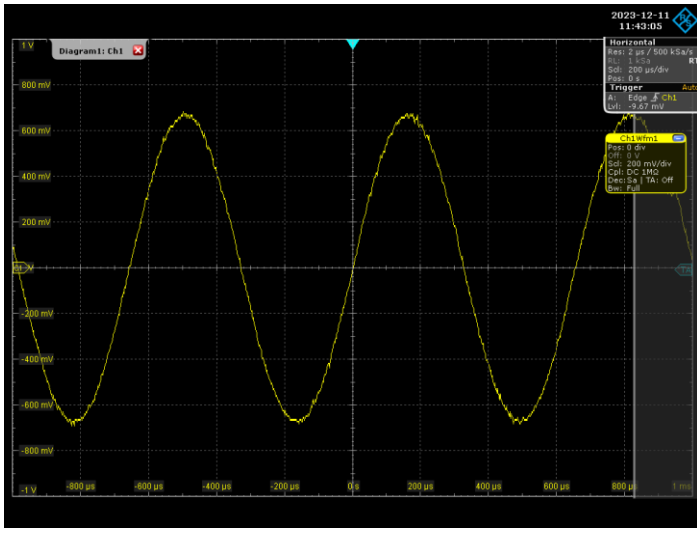


Figure 6: Oscilloscope visualisation of generated sinusoidal wave

3.2 Signal Processing

3 processing operations (as mentioned earlier) were applied onto the fundamental waveforms. Note that the waveforms included below are simulated for clearer visualisations.

3.2.1 Sinusodial Wave Summation

$$\frac{\sin(2x+\pi)}{3} + \frac{3\sin x}{4} \tag{5}$$

The figure below shows the summation of two sine waves of different frequencies and amplitudes (Eq. 5).

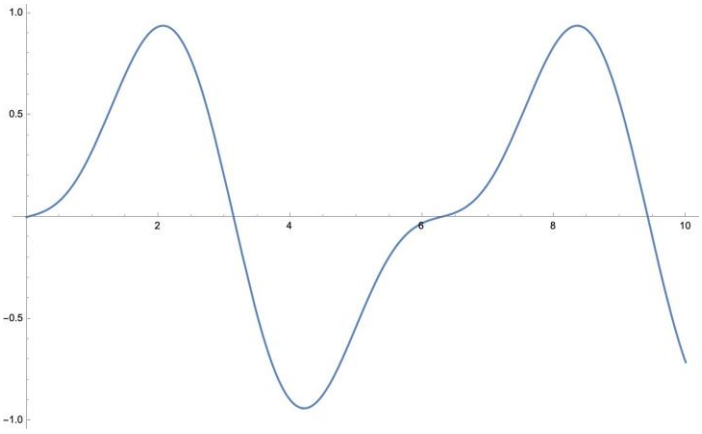


Figure 7: Visualisation of sine summation

3.2.2 Pulse-width Modulation

The Pulse Width Modulated (PWM) signal is generated by a comparison function between a sine (red) and a sawtooth (gray) wave. The PWM signal is asserted 'high' when the sine wave has a higher amplitude than the sawtooth wave, generating an asymmetrical PWM signal — i.e. pulses are not centered within periods.

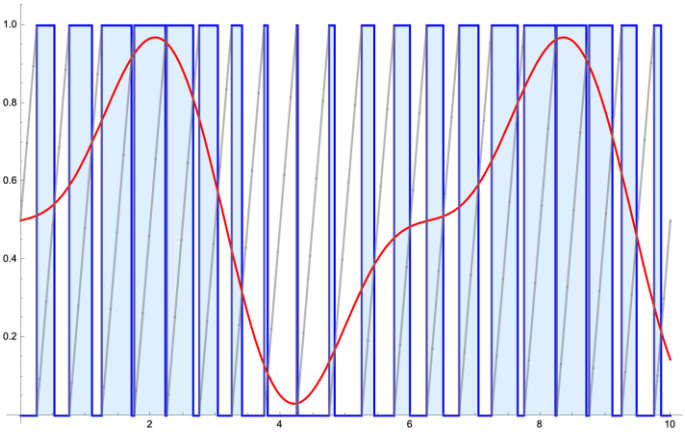


Figure 8: PWM generation from summed sine wave

Following this, we can evaluate the performance of the digital PWM signal in reproducing the original analog signal. Do note that the PWM frequency in this case, 2Hz, is relatively low for demonstration purposes especially in comparison to the signal frequency. Running the PWM signal through an appropriate low-pass filter (as would usually be implemented in hardware) yields the following analog reproduction as seen in Figure 9

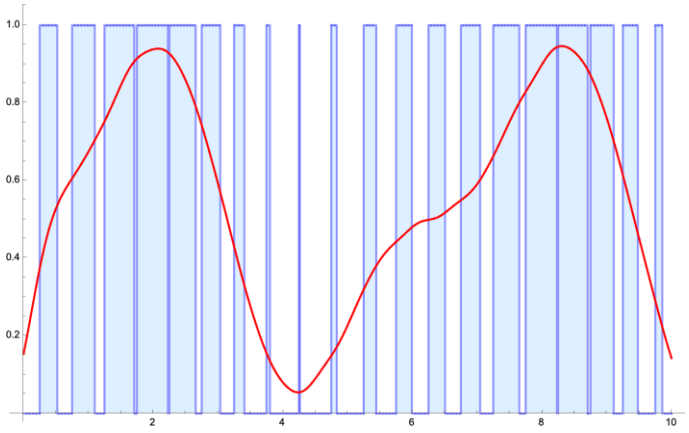
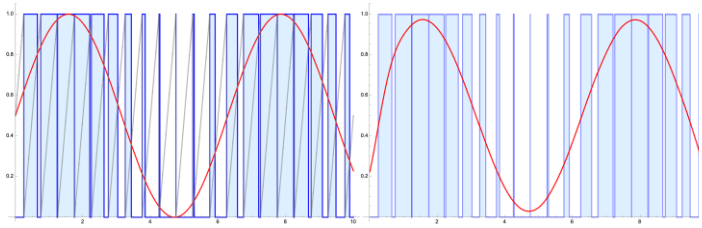


Figure 9: Reproduction of analog signal from PWM

We can see that the reconstructed signal largely resembles the original one in Figure 8, but has some undesirable distortions.

Let us run this pipeline again with a pure sine wave of lower frequency (1Hz) to demonstrate the effect signal frequency, relative to PWM frequency, has. The respective PWM conversion and subsequent reconstruction is shown in Figure 10.



(a) PWM generation (b) Reconstruction

Figure 10: PWM generation and analog reconstruction of lower frequency sine signal

Notice the significantly more faithful reproduction of the signal when it is composed of lower frequency component(s) as compared to the PWM frequency.

4 Discussion

Through this research, it can be seen that signal generation through FPGA logic has great potential in high-performance DSP settings, especially those operating on high-frequency signals (such as radio frequencies). Though this paper only touches the surface of DSP with FPGA chips, it demonstrates the utility of its flexibility in allowing detailed generation parameters and waveforms to be updated on-the-fly. FPGAs being an inherently parallel allows for massive scalability of the DSP algorithms, for example running multiple complex sinusodial wave summations for recreating signals from the frequency domain.

It can be noted that the waveforms generated from the Verilog code is only limited to the audible range (less than 20kHz) as the signal output port is an audio port on the evaluation board. However, the board is able to faithfully produce fundamental waveform like sine waves within the frequency range of the hardware.

In the reconstruction of the summed sine wave from the Pulse-Width modulated waveform, it should be noted that the reconstruction of the sine wave is less than ideal as the frequency of the PWM is insufficient to capture the higher frequency component of the summed sine wave. Increasing the PWM frequency would allow the summed sine wave to be (more) faithfully recreated. It is however possible to fairly faithfully reconstruct a signal composed of lower frequency components relative to the PWM frequency.

5 Conclusion

In conclusion, waveforms were generated by the FPGA on the Nexys evaluation board, through an audio port. The characteristics and limitations of the waveforms generated were discussed. Basic digital signal processing techniques were simulated and discussed.