

Improve performance of the digital sinusoidal generator in FPGA by memory usage optimization

Aiman Zakwan Jidin, Irna Nadira Mahzan, A. Shamsul Rahimi A Subki, Wan Haszerila Wan Hassan
Centre for Telecommunication Research & Innovation, Fakulti Teknologi Kejuruteraan Elektrik & Elektronik (FTKKE), Universiti Teknikal Malaysia Melaka, Malaysia

Article Info

Article history:

Received Aug 12, 2018

Revised Dec 20, 2018

Accepted Jan 2, 2019

Keywords:

FPGA

Function generator

Memory optimization

Sinusoidal

ABSTRACT

This paper presented the improvement in the performance of the digital sinusoidal signal generator, which was implemented in FPGA, by optimizing the usage of the available memory onboard. The sine wave was generated by using a Lookup Table method, where its pre-calculated values were stored in the onboard memory, and its frequency can be adjustable by changing the incremental step value of the memory address. In this proposed research, the memory stores only 25000 samples of the first quarter from a period of a sine wave and thus, the output signal accuracy was increased and the output frequency range was expanded, compared to the previous research. The proposed design was successfully developed and implemented in ALTERA Cyclone III DE0 FPGA Development Board, and its functionality was validated via functional simulation in Modelsim and also hardware experimental results observation in SignalTap II.

Copyright © 2019 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

Aiman Zakwan Jidin,
Centre for Telecommunication Research & Innovation,
Fakulti Teknologi Kejuruteraan Elektrik & Elektronik (FTKKE),
Universiti Teknikal Malaysia Melaka,
Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia.
Email: aimanzakwan@utem.edu.my

1. INTRODUCTION

Signal generator or sometimes called as a function generator, is an equipment which is very widely used in order to generate the input signals for tests and experiments in many industrial applications, like in electronics measurement, telecommunication systems and machine controls, as well as its very popular usage inside the laboratories for teaching and learning purposes [1]. A signal generator is typically a device which is capable of constructing and delivering repetitive signals, which type can be selected for different options available such as the sine wave, the square wave, and the triangular wave. Besides, it is also capable to produce signals with a specific frequency and amplitude values accurately [2]. This will allow users to replicate the input signal targeted for the circuits under tests.

A function generator can be implemented on programmable devices such as a microcontroller [3]. For example, by using a high-level programming language like C language, designers can deploy the predefined `sin ()` function in order to generate the sine wave. Despite its design simplicity, the execution time of microcontrollers are generally quite slow and may degrade the performance. As the matter of fact, all the instruction sets of the microcontrollers are executed sequentially and in most of the cases, only one instruction can be executed at a time. Hence, this weakness can be overcome by using a Field Programmable Gate Array (FPGA). It is an adequate solution for high-performance computations and it is widely used many high-speed applications, owing to its low cost, its ability to implement pipelined and parallel computations, and its capability to operate at high-frequency clocks [4], [5].

In fact, there are different ways of developing and implementing the signal generator in FPGA. One of the popular methods is to generate signals by using the Direct Digital Synthesizer (DDS), a technique which is able to produce output signals with high-frequency range and accurate frequency adjustment. In this technique, the analog output signal is produced by generating the time-varying signal in a digital form, then converted into the analog signal via digital-to-analog conversion. Its principle is to vary the frequency of the clock, which is used to read the pre-calculated waveform amplitude data digitally stored in a memory. Then, the data which have been read is converted to become the analog signal [6]-[8].

Meanwhile, research in [9] proposed a method which is almost similar to the DDS technique. However, the frequency of the operating clock is fixed and the generated signal frequency is tunable simply by adjusting the incremental step value of the address counter in the proposed signal generator.

Besides, research in [10] had implemented the waveform generator in Xilinx Virtex II FPGA, by using the embedded microprocessor. In this research, a soft processor called MicroBlaze, which control the system operation, is interfaced to peripherals such as memories and DAC. However, to achieve high-bandwidth signal generator, it is required to use high-end FPGA such as Virtex FPGA which cost very expensive.

This paper presents the improvement of the performance of the digital sinusoidal generator which was developed and implemented in FPGA. The improvement was able to be made by optimization the usage of the available memory resources on-board. In this proposed research, the sine wave is generated by using the lookup table method, where the pre-calculated signal data are stored in a memory and the signal frequency is adjustable by modifying the value of the address counter incremental steps. The sine wave can be configurable within a range of 1 kHz to 10 MHz, with the frequency resolution of 1 kHz. In this paper, no digital-to-analog conversion is involved and thus, the proposed research produced the digital sine wave with accurate frequency.

2. SINE WAVE GENERATION METHOD

2.1. Lookup table

The lookup table contains the data which represent the samples of the sine waveform, which was pre-calculated offline by using a data processing tool like the Microsoft Excel. In the previous research [9], the lookup table was utilized to stored 20000 16-bit data, which represents the 1 kHz sine waveform data sampled at every 50 ns. Thus, all the 20000 sampling will make a complete cycle with a period of 1 Ms. However, in this proposed research, the lookup table contains 25000 data which was sampled from only the first quarter of a complete cycle of a sine wave. As the matter of fact, this is doable owing to the characteristic of a sine wave which is symmetric.

By looking at the wave in Figure 1, the region 2 of the wave is symmetric to the region 1, while the region 3 and region 4 are symmetric to the region 1 and region 2 with respect to the x-axis, respectively. Therefore, the region 2 of the waveform can be obtained simply by counting down, instead of counting up, the memory address counter, from the maximum to the minimum address of the lookup table. While the waveform in region 3 and region 4 are simply the negation of the values obtained from the region 1 and region 2, respectively.

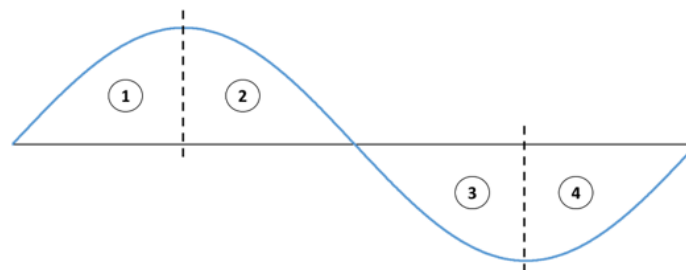


Figure 1. The four regions of a sine wave

Owing to this, the memory resources utilization can be optimized and in this case, since a quarter of the sine waveform is represented by 25000 data, there are in fact a total of 100000 data samples for the complete cycle of the sine wave. In addition, every pre-calculated data stored in this lookup table are written only in 13 bits. Therefore, in the case where its amplitude is adjustable by +/-10, a 16-bit output signal shall be enough to encode the maximum or the minimum value of the generated signal.

2.2. Memory initialization in FPGA

In order to use the onboard memory in the FPGA board, a memory initialization file is needed. This file contains the necessary information such as the memory depth, the size of each data, and also the data to be stored in the memory which can be written either in binary or hexadecimal format. For this research purpose, the type of memory used is the Read-Only Memory (ROM) because all the data stored in the memory are pre-calculated and their values shall remain unchanged when the system is operated.

2.3. Memory address counter

All the data stored in the lookup table can be accessed by using their addresses as shown in Table 1. In this proposed system, a memory address counter is used to increase the address value at every clock cycle. The counter behavior depends on the region of the sine waveform, as shown in Figure 1. It will count up until it reaches or almost reaches the maximum address in region 1 and region 3, while it will count down until it reaches or almost reaches the minimum address in region 3 and region 4.

Table 1. The Lookup Table for Sine Wave Generation

Address	Data
0	0
1	2
2	4
...	...
12499	3275
12500	3276
12501	3275
...	...
24997	6
24998	4
24999	2

Here, the choice of the clock frequency is substantial in order to produce accurate signal frequency at the output. As previously mentioned, the base signal frequency for the proposed system is $F_{base}=1$ kHz. Hence, the base period $T_{base}=1$ ms. Therefore, in order to determine the sampling clock period:

$$T_{sampling} = \frac{T_{base}}{n}$$

where n is the number of samples. Hence, when n is equal to 100000 samples, the sampling clock period shall be equal to 10 ns. Subsequently, a 100 MHz sampling clock must be used. In this case, at every 10 ns, the address counter value will be changed by an incremental step value, which is equal to the frequency of the sine wave to be generated in kHz. For example, in order to produce a 500 kHz sine wave, the address counter must be increased or decreased by 500 at every sampling clock cycle.

3. PROPOSED SYSTEM ARCHITECTURE

Figure 2 illustrates the block diagram of the proposed system for generating sine wave digitally in FPGA. In fact, it consists three main blocks: phase-lock loop (PLL), address counter and sine wave table. In this project, the PLL block serves as the clock frequency multiplier. As the matter of fact, the frequency of the clock oscillator which is available on the FPGA board used is 50MHz. Therefore, in order to produce a 100 MHz clock signal as the sampling clock (clk_100MHz), the frequency of the 50 MHz clock (sys_clk) needs to be multiplied by 2 by using the PLL block.

The sampling clock from the PLL is connected to both the address counter and the sine wave table. The former will increase or decrease the address by the step value it receives from the external. Next, the new value of the address is passed to the latter in order to access the sine wave data which had been stored inside.

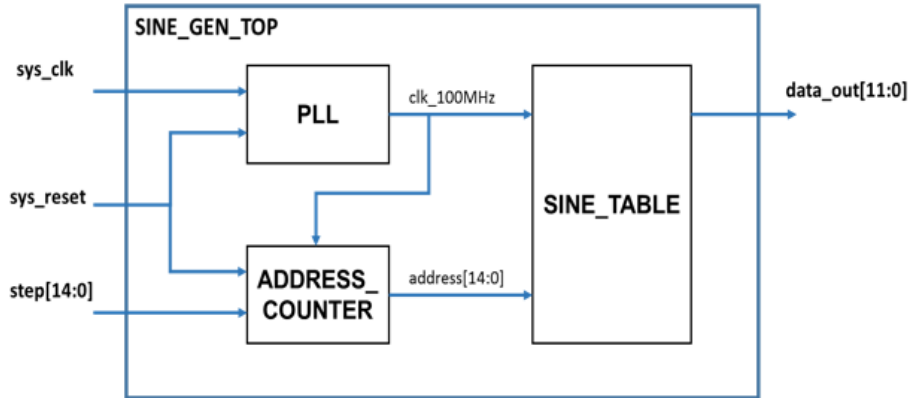


Figure 2. The block diagram which represents the proposed system architecture

The address counter operation is controlled by two finite state machines. The first state machine is essential to generate the correct memory address and also the output signal value sign, depending on the region of the waveform. Figure 3 depicts the state diagram of the state machine, which is consisted of four states, which represent the four regions: ONE, TWO, THREE and FOUR.

In state ONE, the address starts from 0 and it will be increased by step value at each clock cycle. Once the sum of its current address and the step value is larger than or equal to the maximum memory address, which is 24999, the state machine is transiting to state TWO. At this state, the counter will decrease the address by the step value, the subtraction of the current address and the step value is lesser or equal to 0, where the state machine is transiting to state THREE. The same producers take place for the transition from state THREE to FOUR and from state FOUR to ONE. In addition, during states THREE and FOUR, the data read from the memory will be multiplied by -1 in order to have the negative value of the generated sine wave.

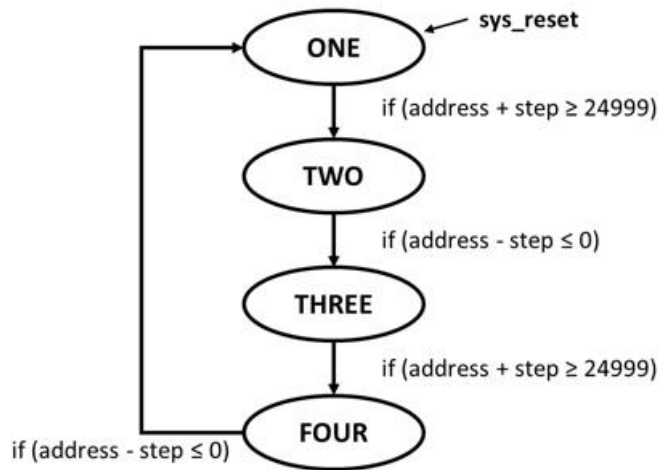


Figure 3. The finite state machine to determine the counter behavior based on the region

While the second state machine, as shown in Figure 4, is used to ensure that the system will generate correct and accurate signal output whenever the step value is updated. This state machine has three different states: COUNT, UPDATE and RESET. In COUNT state, the address counter is in its normal operation, where it increases or decreased the address by the given step value. Then, when the step value is changing, the state machine will update the new count step value in UPDATE state, and it will reset back the address value to 0 in the RESET state. It will stay in this state for 50 ns before transiting to the COUNT state, where the address counter resumes its normal operation with the newly updated count step value.

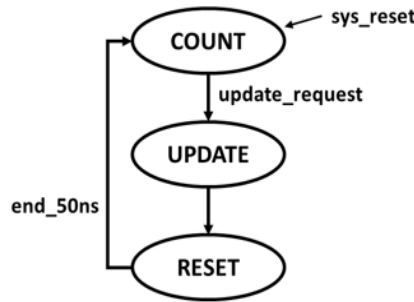


Figure 4. The finite state machine which controls the address counter operation when the step value is updated

4. RESULTS AND DISCUSSION

The proposed design was successfully developed by using Verilog HDL code. In order to validate the correctness of the proposed system functionalities, both functional simulation and hardware experimental tests are performed. In both cases, validation was executed on four different sets of desired frequency values: 125 kHz, 667 kHz, 2000 kHz, and 7500 kHz.

4.1. Functional simulation

In this research, the simulation was executed by using Mentor Graphic ModelSim Altera Edition software. Before running the simulation, the testbench for the tests was written in Verilog. Figure 5 shows the results of the functional simulation with five different frequency values. From these waveforms, the output signal frequencies can be obtained by measuring the period of each signal using two different cursors. The measured frequencies are then compared to the desired frequencies, as listed in Table 2. Based on this observation, the frequency of the generated sine wave was very accurate for the first three signals. However, it produced a small error of 1kHz (0.2%) when generating the 7500 kHz sine wave.

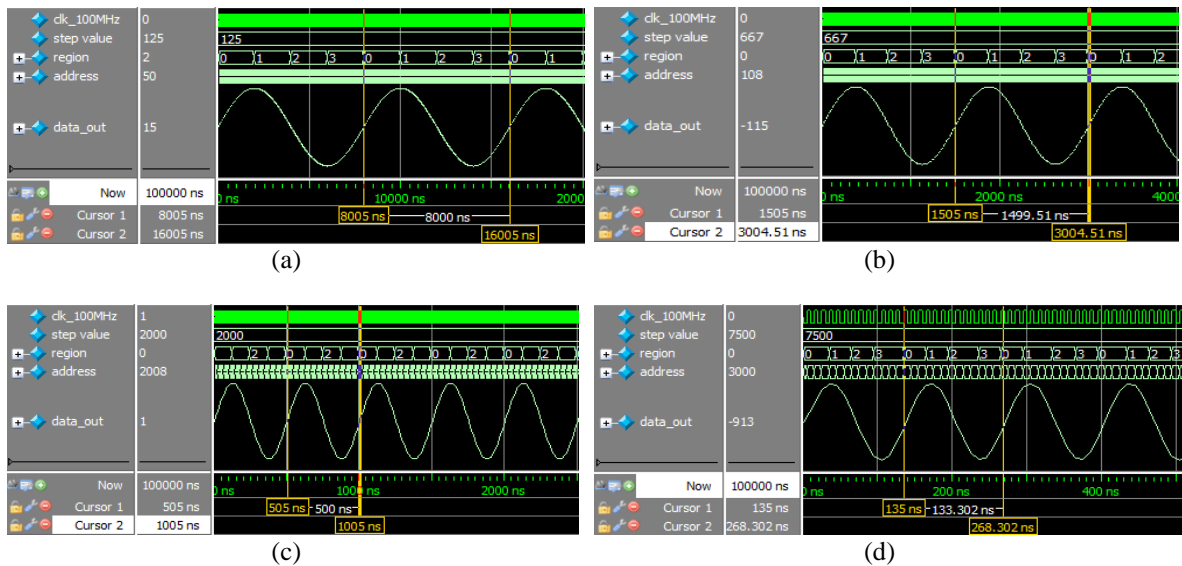


Figure 5. The waveform results for functional simulation by using different frequency values: (a) 125 kHz, (b) 667 kHz, (c) 2 MHz, (d) 7.5 MHz

Table 2. Comparison between Desired Frequency vs Measured Frequency

Desired Frequency	Measured Period	Measured Frequency
125 kHz	8000.0 ns	125.0 kHz
667 kHz	1499.5 ns	666.9 kHz
2000 kHz	500.0 ns	2000.0 kHz
7500 kHz	133.3 ns	7501 kHz

4.2. Hardware experimental test

The developed Verilog code for the proposed design was successfully compiled by Altera Quartus II software and then implemented in Altera Cyclone III DE0 FPGA development board. In addition, since there is no digital-to-analog converter involved in this research, the generated output signal was observed and analyzed by using the SignalTap II Logic Analyzer. The hardware experimental setup as shown in Figure 6.

The output results of the hardware experimental tests in FPGA are shown in Figure 7. By using the same method used in the simulation, the period of the generated output sine waves are measured and thus, the generated signal frequencies can be obtained and then compared with their expected values. It can be observed that the results obtained from the hardware experimental test are the same as the functional simulation, as previously shown in Table 2.

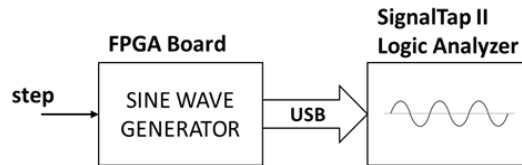


Figure 6. The hardware experimental setup

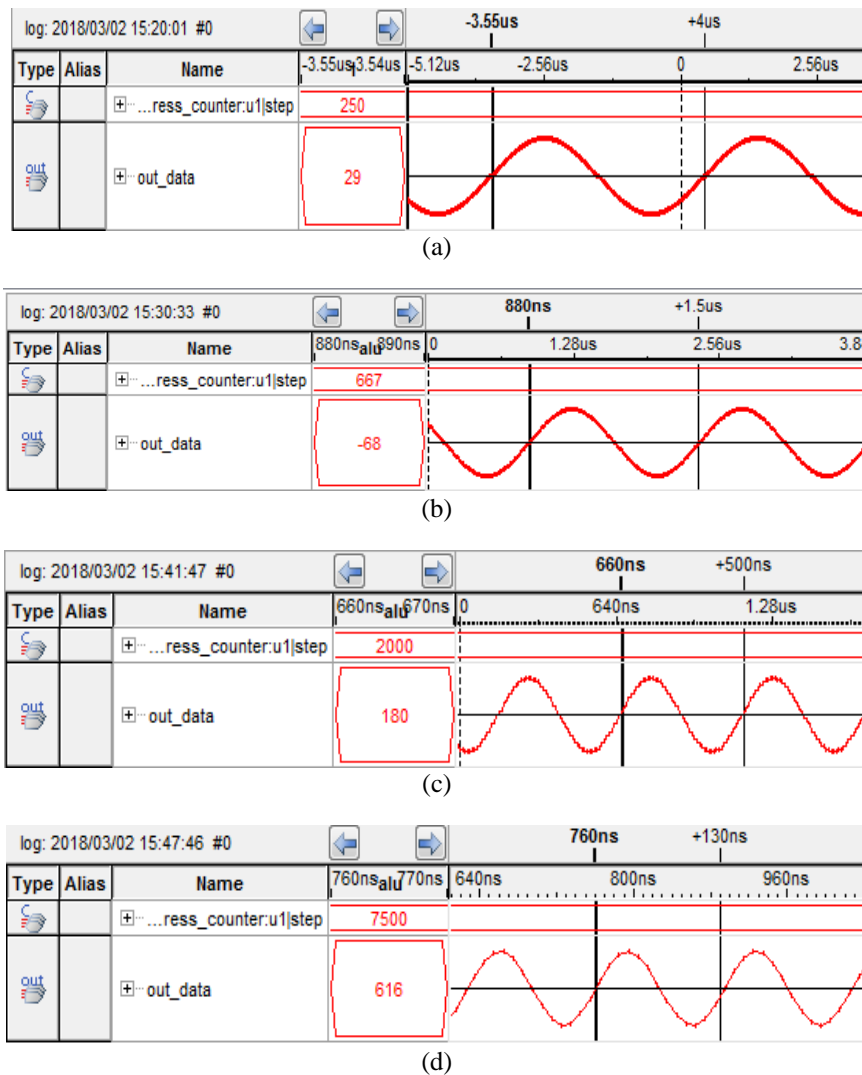


Figure 7. The waveform results from the FPGA hardware test observed in SignalTap Logic Analyzer by using different frequency values: (a) 125 kHz, (b) 667 kHz, (c) 2 MHz, (d) 7.5 MHz

As the matter of fact, the small precision errors for high-frequency generated signal observed in the hardware experimental tests are due to the limitation of the utilized tools: since the generated signal is a discrete signal which had been sampled every 10 ns, the time unit resolution is limited only to 10 ns. Hence, for the 7.5 MHz sine wave, the time interval for a period of the signal is shown as 130 ns, instead of 133 ns.

Table 3 shows the number of the hardware resources used in the proposed research, which was obtained from the generated compilation report in Altera Quartus II software.

Table 3. FPGA Hardware Resources Utilization for Proposed System Implementation

Design	Hardware Resource Usage			
	Logic Elements	Register	Onboard Memory	PLL Block
Proposed design without SignalTap	190 (1.2%)	23 (0.1 %)	325000 (63.0%)	1 (25.0%)
Proposed design with SignalTap	1008 (7.0%)	739 (4.8.0%)	435592 (84.4%)	1 (25.0%)

While Table 4 compares the overall performance of the proposed system with the previous work [9]. As can be seen from this table, the former consume lesser logic elements than the latter, but its memory utilization is a little bit higher (by 1% more) than the latter. However, the frequency range supported by the former, which is between 1 kHz to 10 MHz, is much better than the latter, which can only generate a signal ranging from 1 kHz to 1 MHz.

Table 4. Comparison Between Proposed System Vs Previous Work

Design	Memory Utilization (bits)	Logic Elements	Sampling Rate (MHz)	Frequency Range
Proposed System	325000 (63.0%)	190 (1.2%)	100	1 kHz – 10 MHz
Previous Work	320000 (6.0%)	240 (1.5%)	20	1 kHz – 1 MHz

For further improvements, in order to improve the test and validation of the generated signal, a digital-to-analog converter (DAC) could be added at the output and thus, the output signal can be visualized by using the oscilloscope, a tool that is normally capable of measuring the frequency of the signal more accurately. Besides, the total harmonic distortion of the generated signal could also be analyzed in order to verify its frequency purity.

Moreover, more features need to be added to this proposed design to make it more useful to users. For example, common signal types like triangle, sawtooth and pulse may be added as an option. Besides, functionalities such as amplitude and phase adjustment will be very useful in many applications. Furthermore, it is also possible to have a multi-channel function generator, where two or more signals can be generated simultaneously. But, this one may depend on the FPGA device capabilities in term of hardware resources.

5. CONCLUSION

This paper has discussed on the improvement of the implementation of the digital sine wave generator in FPGA, which had been achieved by optimizing the memory resources utilization. In this proposed research, the generated frequency accuracy and also its frequency range was improved by increasing the number of samples for one period of the signal and thus, increasing its sampling rate. This was achievable owing to the symmetric characteristic of the sine wave and thus, only the first quarter the signal need to be sampled and stored in the memory. The proposed research had been successfully implemented in Altera Cyclone III DE0 FPGA and the correctness of its functionality had been verified by using both the functional simulation and also the FPGA hardware experimental test, where the output produced by the latter was observed in SignalTap II Logic Analyzer.

ACKNOWLEDGEMENTS

The authors wish to acknowledge Universiti Teknikal Malaysia Melaka (UTeM) and the Malaysia Ministry of Higher Education for the financial funding under Grant No. FRGS/1/2015/TK04/FTK/03/F00285 and providing instrumentation devices support for this project.

REFERENCES

- [1] S. Ding, A. An, and X. Gou, "Digital Waveform Generator Based on FPGA," *Res. J. Appl. Sci. Eng. Technol.*, vol. 4(14), pp. 2160–2166, 2012.
- [2] A. H. Tirmare, M. S. R. Mohite, V. A. Suryavanshi, T. C. Department, B. Vidyapeeth, and E. Kolhapur, "FPGA Based Function Generator," *International Research Journal of Engineering and Technology (IRJET)*, vol. 2(9), pp. 2394–2399, 2015.
- [3] V. M. Ibrahim, O. Taiwo, and U. E. Uyoata, "Microcontroller Synthesized Function Generator," *International Journal of Engineering Research and Development*, vol. 2(7), pp. 19–25, 2012.
- [4] G. Brebner and W. Jiang, "High-Speed Packet Processing Using Reconfigurable Computing," *IEEE Micro*, vol. 34(1), pp. 8–18, 2014.
- [5] W. Zheng, R. Liu, M. Zhang, G. Zhuang, and T. Yuan, "Design of FPGA Based High-Speed Data Acquisition and Real-Time Data Processing System on J-TEXT Tokamak," *Fusion Eng. Des.*, vol. 89(5), pp. 698–701, 2014.
- [6] S. Yanbin, G. Jian, and C. Ning, "High Precision Digital Frequency Signal Source Based on FPGA," in *Physics Procedia*, vol. 25, pp. 1342–1347, 2012.
- [7] X. Ye, M. Gao, and J. Huang, "12 -Way High Accuracy Sine Signal Generator System Based on FPGA," *IEEE 16th International Conference on Communication Technology (ICCT)*, Hangzhou, pp. 833–836, 2015.
- [8] M. Herrero, J. J. Rodríguez-Andina, and J. Fariña, "FPGA-based Design, Implementation and Evaluation of Digital Sinusoidal Generators," in *IECON Proceedings (Industrial Electronics Conference)*, pp. 2459–2464, 2008.
- [9] A. Z. Jidin, I. N. Mahzan, N. Hassim, and A. F. Kadmin, "Low-Cost and Portable Interactive Sinusoidal Digital Signal Generator by Using FPGA," *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, vol. 10(1), pp. 19–24, 2018.
- [10] A. Goman, "Waveform Generator Implemented in FPGA with an Embedded Processor," *Master Thesis, Linköping University, Sweden*, no. LITH-ISY-EX-3412-2003, 2003.

BIOGRAPHIES OF AUTHORS



Aiman Zakwan Jidin obtained his MEng in Electronic and Microelectronic System Engineering from ESIEE Engineering Paris France in 2011. He has 2 years of working experience in designing digital IC and digital system in FPGA at Altera Corporation Malaysia, before joining Universiti Teknikal Malaysia Melaka (UTeM) as lecturer and researcher, in Electronics and Computer Engineering. His research interests include FPGA Design and Digital System Design



Irna Nadira Mahzan is currently in her final year of studies in Bachelor of Computer Engineering Technology in Universiti Teknikal Malaysia Melaka (UTeM). Recently, she had been working on the development of a low-cost portable signal generator by using FPGA as her final year project. Currently, she is undergoing her final year internship at UTAC Manufacturing Services Sdn Bhd, which is expected to be completed in July 2018.



A Shamsul Rahimi A Subki obtained his MSc in Microelectronics from Universiti Kebangsaan Malaysia in 2011. He has 2 years of working experience as Photolithography and Wet Etching Process Engineer at ON Semiconductor Malaysia and as Process Integration Engineer at SilTerra Malaysia focusing on maintaining process parameters for CMOS 0.16 nm technology, before joining Universiti Teknikal Malaysia Melaka (UTeM) as lecturer and researcher, in Electronics and Computer Engineering. His research interests include CMOS process flow optimization and characterization, and power electronics.



Wan Haszerila Wan Hassan holds a Master of Electrical, Electronic & Telecommunication Engineering from UTM and Bachelor of Electrical & Electronic Engineering from UTM. She has previous working experience in semiconductor industry at Infineon Technologies Malaysia. She join Universiti Teknikal Malaysia Melaka (UTeM) as lecturer on 2014. Her research area includes Photonics, Optical Communication and Wireless Communication.