Simulation of Counter Based DPWM for implementation on FPGA

Vindhyali Sood[1]*, M.P.R.Prasad[2]

[1] SVE Dept., National Institute of Technology, Kurukshetra. Haryana, India

[2] Member, IEEE, EE Dept., National Institute of Technology, Kurukshetra, Haryana, India

*Corresponding author

doi: https://doi.org/10.21467/proceedings.114.60

Abstract

A Digital Pulse Width Modulation technique based on the counter has been considered for analysis and simulation. It utilizes one of the many functions due to the advanced characteristics already present on the field-programmable gate array (FPGA) which is a huge advantage. Delay Locked Loop (DLL) is one of those features on the FPGA that is used. This architecture merged a counter-comparator-based synchronous block with an asynchronous block that uses the Delay Locked Loop (DLL). This action helps to achieve a better or higher resolution. The architecture proposed is to be executed on an inexpensive but lower-speed FPGA. This FPGA is given a 32 MHz clock externally that helps us to get a time resolution under 2ns. To use the DLL on FPGA, the Digital Clock Manager (DCM) block is used.

Keywords: Pulse Width Modulation (PWM), Delay Locked Loop (DLL), Digital Clock Manager (DCM), FPGA

1 Introduction

The enormous research attention received by control based on digital techniques is due to its advantages that are very well-known such as ease of programming or programmability (flexibility), control algorithms of advanced nature (expandability), reliability, etc. to name a few. However, there are certain drawbacks of digital control also. One of them is reduced resolution that ultimately results in reduced accuracy. Along with this, delays are also added because of sampling and processing times [12].

Pulse Width Modulation or PWM technique is a way by which the power delivered by an electrical signal is reduced. It does so, by effectively converting the electrical signal into discrete parts [3], [4]. In PWM the width or duration of the signal in the form of a pulse is decided by the sample values of the analog waveform. [5] Traditional DPWMs that have their structure based on Counters have the benefit of having a simple architecture along with elevated linearity. A high resolution, however, is tedious to achieve since the minimum time step is enough to the counter's clock period. To get a comparatively better resolution, clock frequency needs to be increased, but power utilized directly depends on the clock frequency, therefore in a quest to get better resolution the power utilization may increase. One solution to increase the Digital Pulse Width Modulator resolution is to use delay lines [5], [6], [7].

In recent years, many researchers have proposed various possible architectures for Digital Pulse Width Modulators. Many classifications of them have also been made. There are many differences in these architectures, but all of them mostly utilize delay lines. Less power utilization and a better resolution are the main advantages of delay line DPWM architectures [4], [5]. Despite the advantages drawbacks such as lower linearity along with non-monotonic behavior are also present in some cases. To overcome these issues and obtain optimum performance, hybrid architectures are suggested that utilize the pros of both counter-based DPWM and delay line DPWM.

Proceedings DOI: 10.21467/proceedings.114; Series: AIJR Proceedings; ISSN: 2582-3922; ISBN: 978-81-947843-8-8

In this paper simulation results of a hybrid architecture for Digital, Pulse Width Modulation are presented. The primary motive here is to use utilize the inbuilt Digital Clock Manager (DCM) present on the FPGA. The DCM [11] block serves the dual purpose of Delay Locked Loop (DLL) used to achieve phase-shifted versions of input clock frequency, along with multiplication of input clock frequency. For simulation purposes, the already available IP Core of the DCM block of Spartan 3 FPGA is utilized. This FPGA is inexpensive and provides less speed, but the same can be executed on higher-end FPGAs also. An externally generated clock frequency of 32 MHz is fed to the FPGA [1], [2], [3]. The mentioned architecture can also be designed by IC implementation, but that can be cumbersome. As a result, to avoid the tedious process of IC implementation this FPGA-based approach is proposed [8], [9].

The organization of the paper is as follows: Section I gives an introduction followed by Section II that explains the various blocks and their functioning. Section III presents the simulation results. Finally, in Section IV conclusions are presented followed by references.

2 Different -Blocks In Proposed Architecture

2.1 DLL Block

FPGAs are digital devices with various blocks present in them that offer a plethora of functions. One of these functions is managing the clock signals. To achieve this functionality the Delay Locked Loop (DLL) or Phase Locked Loop (PLL) on-board the FPGA are used. The multiplied or divided clock frequencies can be phase-shifted. For this architecture, four clocks that are phase-shifted i.e. shifted by 0°, 90°, 180°, and 270° can also be generated by many of these DLLs directly [10]. In this architecture, the DLL's feature of multiplying the clock frequency is utilized. For counter-based DPWM a frequency in a higher range is used for the clock internally. On the other hand, the rest of the controller which is based on digital technique works on an external frequency in the lower range. This is a huge advantage as it helps to decrease power utilization.

For this suggested architecture we try to achieve a 128 MHz clock internally. This is done by multiplying the on-chip 32 MHz clock by 4. Internal clock multiplication significantly improves the resolution. The rest of the controller module can work at a frequency in the lower range as this serves the dual purpose i.e. less power utilization along with a design that is easy to make. Block Diagram for clock multiplication and phase shifting of the clock is shown in fig 1.

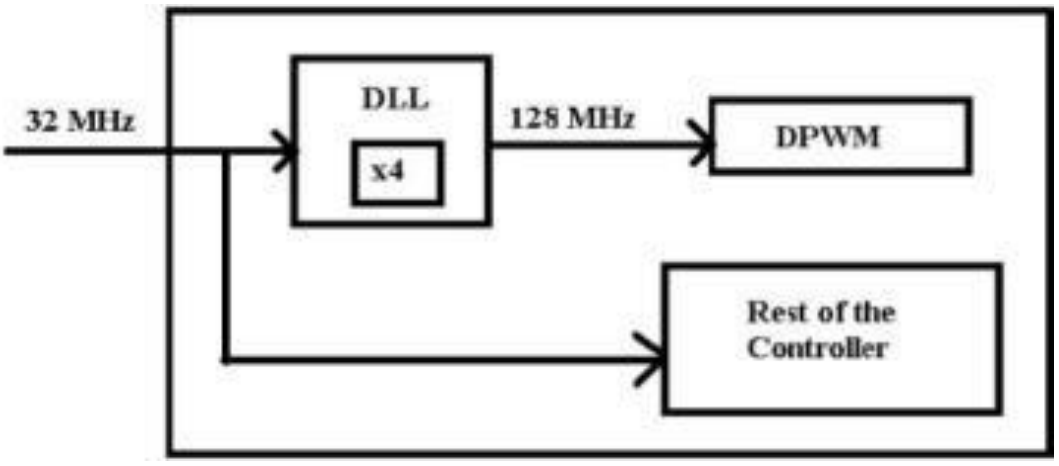

Fig. 1: Clock Distribution for DPWM and the rest of the controller [1].

This approach is very beneficial because a Digital Pulse Width Modulator with architecture based on the counter is easy to design. At the same time, it works well at frequencies in the higher range. On the contrary, other blocks in this design are comparatively complex in terms of design and can only work at frequencies in the lower range. Hence, in this architecture separate clock frequencies are employed i.e. high clock frequency for Digital Pulse Width Modulator, while lower one for remaining blocks of the controller.
Another DLL feature is the main contributor to the proposed DPWM. The on-board DLL block [11] on the FPGA generates clock outputs with different phase shifts as specified by the user. In these, four clocks shifted 0◦, 90◦, 180◦, and 270◦ are available. Due to this feature, the time resolution can be increased by multiplying with 4 because of the two additional LSB bits.

2.2 Counter Block

The counter block is synchronous since it gives the output signals as per the clock signal that it receives. For our purpose here, to increase the resolution a higher frequency is fed to this block. Hence, the use of clock frequency multiplication.
This block makes use of the MSBs of the input signal i.e. duty cycle d [n-1, 2], where 'n' is the total number of bits.
A counter and comparison approach is used that takes after the analog Pulse width Modulation where saw tooth wave is used as the input signal.
This approach is illustrated with the help of fig 2.

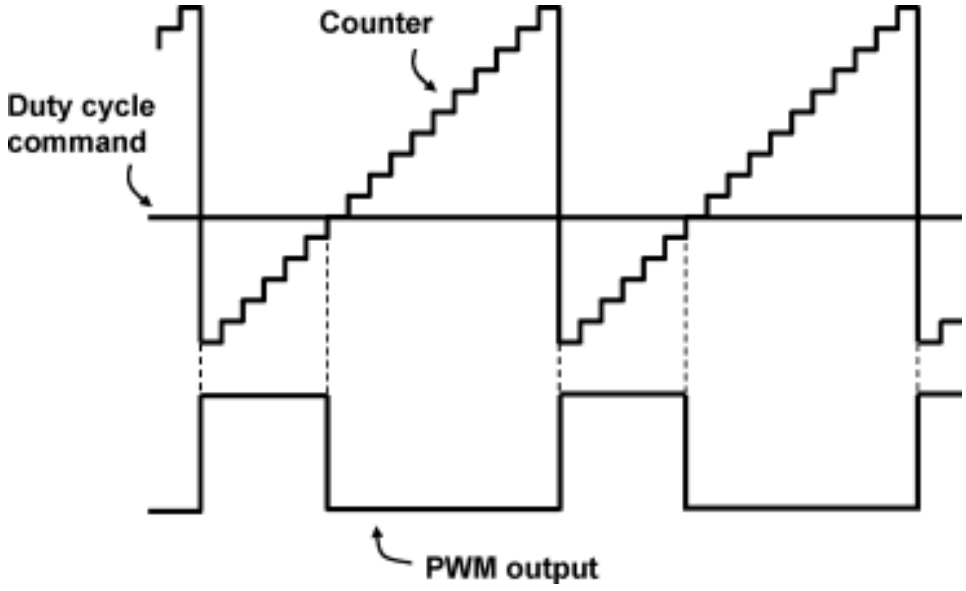

Fig 2: Counter and comparison based synchronous block's approach [1]

The functionality of this block is explained as follows: the counter output here which is identical to an analog sawtooth waveform is compared to the duty cycle command i.e. the input signal. Based on this comparison the output goes high for a counter value less than input and goes low for a counter value greater than the input.
This resolution can be obtained as:

$$\text{RESOLUTION} = \frac{f_{clk}}{f_{sw}}$$

where is the clock frequency is denoted by *fclk* and the switching frequency is denoted by *fsw*.

3 Time Distribution

The function of time distribution is carried out by the asynchronous block that uses the last two bits i.e. the two LSBs, d [1, 0]. The Delay Locked Loop (DLL) generates four phase-shifted clocks. To select between them the last two LSBs are used. The next step is to generate a QUARTER CYCLE signal. As the name indicates this signal is high for a quarter cycle instead of half a cycle. To get this signal the four phase-shifted clocks are merged by using AND gates. The figure below shows how the QUARTER CYCLE signal changes according to LSBs of the duty cycle command.

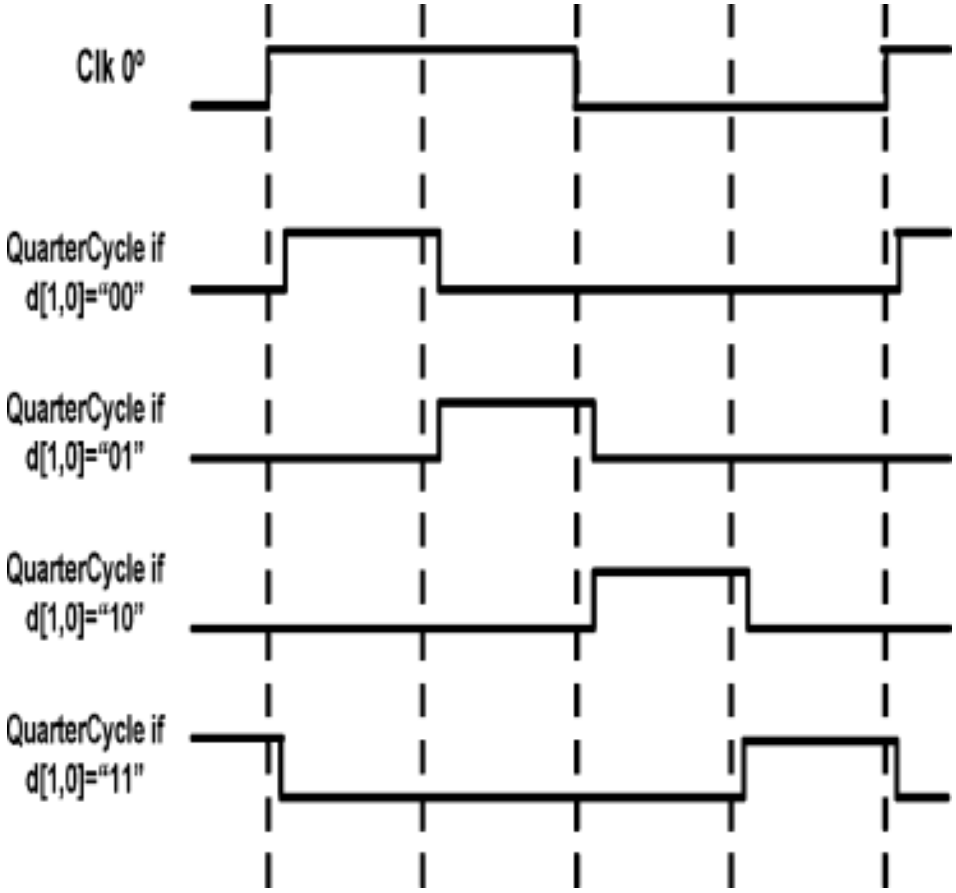

Fig 3: Variation of Quarter Cycle signal according to LSB of duty cycle command [1]

The prime motive behind using clock signals with phase shifts is to obtain four switching instants. Therefore, multiplication of resolution is done by 4.

'm' asynchronous bits i.e. 2^m phase-shifted clocks gives the total delay as:

$$\text{RESOLUTION} = 2^m \frac{f_{clk}}{f_{sw}}$$

Where the number of asynchronous bits is given by **'m'**, f_{clk} the clock frequency is **'f_{clk}'**, and the switching frequency is **'f_{sw}'**.

4 Functionality Of Counter Based Dpwm

It is shown in Fig 4 that the DPWM architecture suggested has two main blocks, namely a synchronous block and an asynchronous block. These blocks constitute the backbone of this architecture.

Proceedings DOI: 10.21467/proceedings.114
ISBN: 978-81-947843-8-8

Series: AIJR Proceedings
ISSN: 2582-3922

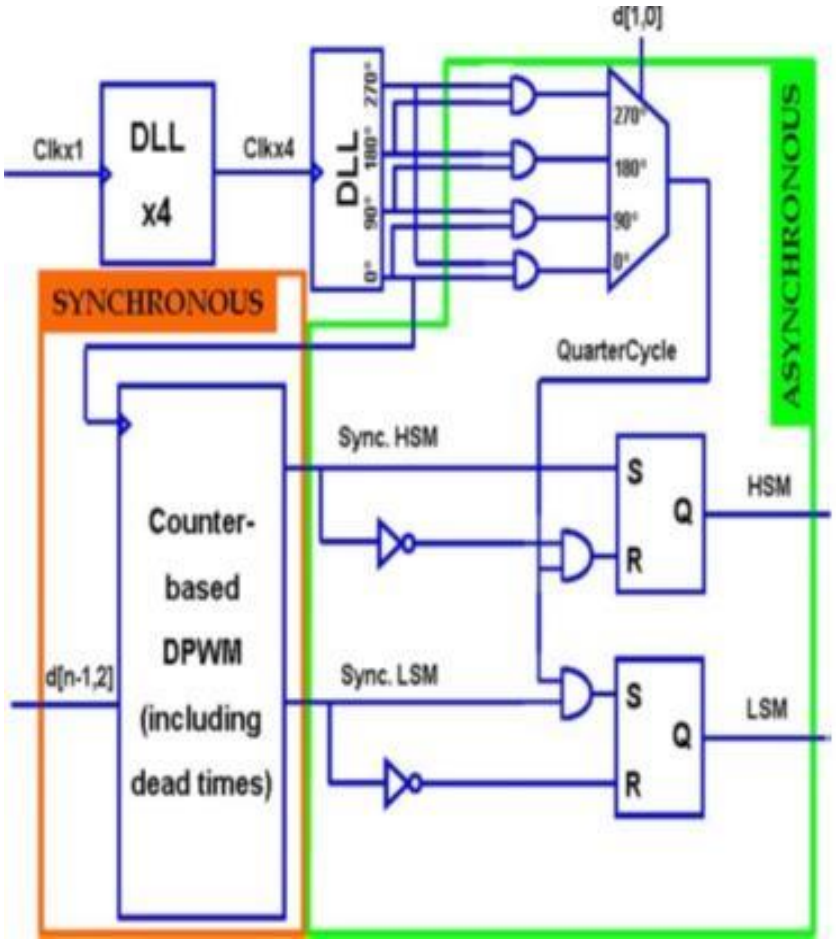

Fig 4: Block diagram of the proposed counter based Digital Pulse Width Modulator [1]

High side MOSFET (HSM) output is turned high by the synchronous block's output which is based on counter comparator approach. External frequency multiplied by 4 is given to this block. The output of synchronous block depends on Most Significant Bits of duty cycle command i.e. 'd'. The counter is fed the multiplied clock frequency and the asynchronous block utilizes 4 versions of the input clock frequency with different phase shifts.

In this suggested work driving signals for both HSM as well as low side MOSFET (LSM) are generated and they have dead times that are programmable. The output HSM has its turn ON instant coinciding with the 0◦ clock, while any of the four clock edges can serve as the turn OFF instant. Clearly, this action is dependent on the LSBs. On the contrary, LSM goes high according to any of the four clock edges and goes low only with 0◦ clock.

The signal Quarter Cycle shown in Fig 3 is generated by the asynchronous block and stays high for a quarter of the clock cycle. This signal goes high at the rising edge of any of the four phase-shifted clock cycles depending on the LSBs of the duty cycle. When the QUARTER CYCLE signal arrives and the synchronous block is off, HSM output goes low. Utilizing the (using n-2 most significant Bits by counter) integer number of clock cycles and the QUARTER CYCLE signal (that depends on the LSBs), HSM output goes high.

5 Simulation Results

The simulation of the architecture explained previously is done on Xilinx ISE Design Suite 14.7 that makes use of ISim Simulator. The language used is Verilog, although, VHDL can also be used for this purpose.

The exact device is XC3S200FT256-4. FT256 is the package used and -4 is the speed grade. Fig 5 and Fig 6 show the simulation results and the generated RTL schematic respectively.

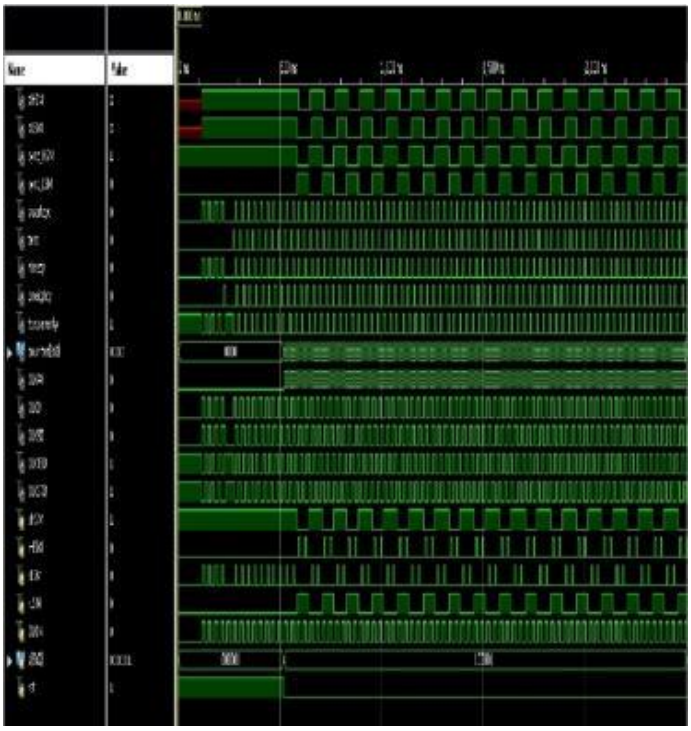

Fig 5: Simulation results of the Digital Pulse Width Modulator Architecture

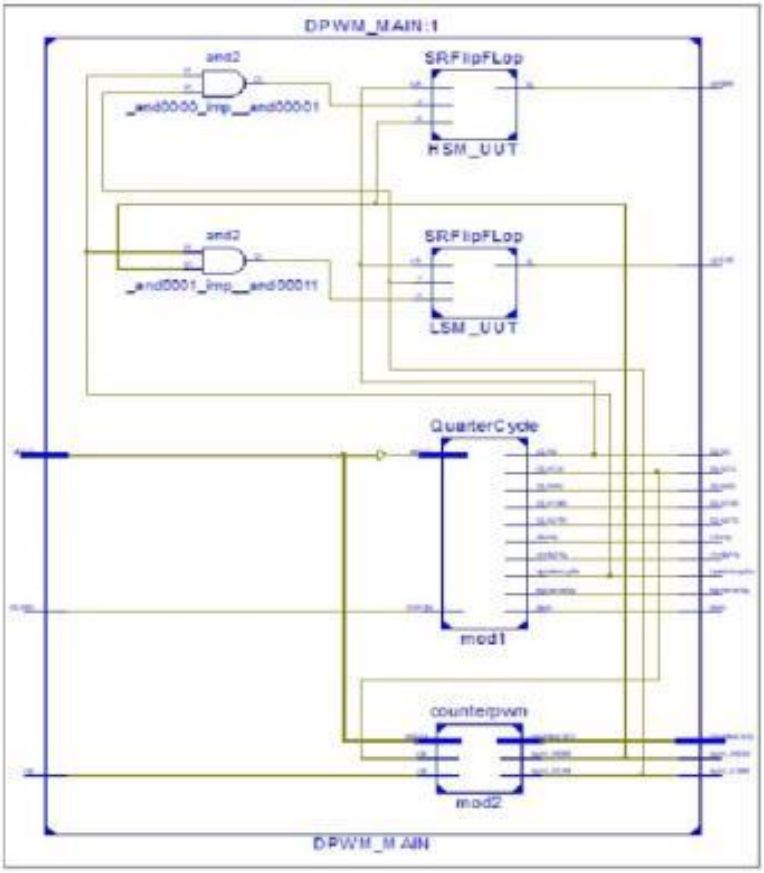

Fig 6: RTL Schematic of the architecture

The device utilization summary statistics are also shown:

Device Utilization Summary (estimated values)			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	8	1920	0%
Number of Slice Flip Flops	4	3840	0%
Number of 4 input LUTs	16	3840	0%
Number of bonded IOBs	26	173	15%
Number of GCLKs	1	8	12%
Number of DCMS	1	4	25%

Fig 7: Statistics of the device utilization

The simulation results are obtained using a clock with frequency of 32 MHz. The internal clock i.e. the clock fed to the counter is 128 MHz as a result of multiplying the input frequency of clock by 4. This function is carried out by the DCM block on the FPGA. The input clock is given to the DCM which generates 4 signals that are phase shifted using the DLL present inside it. To calculate the final resolution we use the following formula:

$$\Delta T = \frac{T_{clk4x}}{2^m}$$

where Tclk4x is 128 MHz clock and 'm' denotes the 2 asynchronous bits. The resolution finally obtained is therefore 1.95 ns.

6 Conclusion

An attempt has been made to find qualitative and quantitative analysis of a novel Digital Pulse Width Modulator Architecture and the results are obtained using Verilog instead of VHDL [1]. The motive here is to reduce the design complexity through implementation on an FPGA instead of designing an IC. Also, FPGA is cheaper compared to IC implementation.

The DLL does clock multiplication and shifts the input clock frequency to generate four different signals. The dual-purpose served by the on-board DLL block on FPGA greatly reduces the design complexity. This DPWM can serve as the driving block in various electronics and power electronics applications, like servo motor control.

References

[1] S.C. Huerta, A. de Castro, O. Garcia, J.A. Cobos, "FPGA based digital pulse width modulator with time resolution under 2 ns," *IEEE Trans. Power Electron.*, vol. 23, no. 6, pp. 3135–3141, 2008.

[2] A. de Castro and E. Todorovich, "DPWM based on FPGA clock phase shifting with time resolution under 100 ps," *IEEE Power Electronics Specialists Conference*, pp. 3054-3059, 2008.

[3] V. Yousefzadeh, T. Takayama, D. Maksimovic, "Hybrid DPWM with digital delay-locked loop," *Proc. IEEE Computers in Power Electronics Workshop (COMPEL)*, pp. 142-148, 2006.

[4] R. Foley, R. Kavanagh, W. Marnane, M. Egan, "Multiphase digital pulsewidth modulator," *IEEE Trans. Power Electronics*, vol. 21, n. 3, pp. 842-846, May 2006.

[5] A. Syed, E. Ahmed, D. Maksimovic, E. Alarcon, "Digital pulse width modulator architectures," *Proc. IEEE Power Electronics Specialists Conf. (PESC)*, vol. 6, pp. 4689-4695, June 2004.

[6] K. Wang, N. Rahman, Z. Lukic, A. Prodic, "All-digital DPWM/DPFM controller for low-power DC-DC converters," *Proc. IEEE Applied Power Electronics Conf. Expo (APEC)*, pp. 719-723, March 2006.

[7] R. F. Foley, R. C. Kavanagh, W. P. Marnane, and M. G. Egan, "An area efficient digital pulse width modulation architecture suitable for FPGA implementation," *in Proc. IEEE Appl. Power Electron. Conf. Expo (APEC)*, vol. 3, pp. 1412–1418, March 2005.

[8] M. Fernandez-Gomez, C. Fernandez, P. Zumel, A. Sanchez and A. d. Castro, "Design of DPWM with high resolution under 80 ps using low-cost Xilinx FPGA," *IEEE Applied Power Electronics Conference and Exposition (APEC)*, pp. 3043-3048, 2020.

[9] D. Costinett, M. Rodriguez, D. Maksimovic, "Simple Digital Pulse Width Modulator with 60 Picoseconds Resolution Using a Low-cost FPGA," *Int. Power Electron. Motion Control Conf.*, pp. 1–7, 2012.

[10] A. de Castro, E. Todorovich, "High Resolution FPGA DPWM Based on Variable Clock Phase Shifting," *IEEE Transactions on Power Electronics*, vol. 25, pp. 1115–1119, 2010.

[11] D. Navarro, Ó. Lucía, L. A. Barragán, J. I. Artigas, I. Urriza, and Ó. Jiménez, "Synchronous FPGA-based high-resolution implementations of digital pulse-width modulators," *IEEE Trans. Power Electron.*, vol. 27, no. 5, pp. 2515–2525, 2012.

[12] A. Peterchev and S. Sanders, "Quantization resolution and limit cycling in digitally controlled PWM converters," *IEEE Trans. Power Electron.*, vol. 18, no. 1, pp. 301 – 308, Jan 2003.

[13] L. S. Ge, Z. X. Chen, Z. J. Chen, and Y. F. Liu, "Design and implementation of a high resolution DPWM based on a low-cost FPGA," *in Proc. IEEE Energy Conv. Cong. Exp. (ECCE)*, pp. 2306 –2311, Sept. 2010.

[14] M. G. Batarseh, W. Al-Hoor, L. Huang, C. Iannello, and I. Batarseh, "Window-masked segmented digital clock manager–FPGA based digital pulse width modulator technique," *IEEE Trans. Power Electron.*, vol. 24, no. 11, pp. 2649–2660, Nov. 2009.

[15] Xilinx XAPP462 (v1.1), Using Digital Clock Managers in FPGAs, January 5, 2006