Generation of Pulse Width Modulated wave using FPGA

Mr. Kishore Ramesh Badgujar[1] Mr. Pankaj A. Salunkhe[2] Mr. Tushar M. Patil[3]
[1,2,3]Department of Electronics and Telecommunication Engineering
[1,2]Yadavrao Tasgaonkar College of Engineering & Management, University of Mumbai, Bhivpuri Road, Karjat, India
[3]G.H.Raisoni Institute of Engg. & Management North Maharashtra university Jalgaon, India

Abstract— Based on field programmable gate array Pulse width modulation (PWM) has been widely used in power electronics for controlling the various types of quantities such as speed, voltage, lightning, etc. Most high power level converters operate at switching frequencies up to 500 kHz; the disadvantage of such a low frequency application implies vibrations and nonlinear functionality of the devices. While operating frequencies in excess of 1 MHz at high power levels can be achieved and gives much more stable operation of the same operating devices.

Key words: Field Programmable Gate Array, Pulse Width Modulator

I. INTRODUCTION

DIGITAL control of motors has obtained great research attention due to their now well-known advantages [1]–[5], such as programmability, advanced control algorithms, reduced component count, low sensitivity to external factors or aging, ease of design and prototyping, etc.

The two most important disadvantages are linearity in output and vibrations in the application like motor.[6]. Regarding the second factor, resolution is limited mainly by the analog-to digital converter (ADC) and the pulse-width-modulation (PWM). However, the ADC resolution is becoming a less important problem, thanks to the windowed ADC technique [3] and because the PWM resolution needs to be higher than the AD C resolution for avoiding limit cycling [6], [7]. Traditional digital PWMs (DPWMs) are based on counters (see Section II-B). The advantage of these DPWMs is that they are very simple and obtain high linearity. However, their resolution cannot be very high, as the minimum time step is equal to the clock period of the counter. As the clock frequency increases the output of the PWM waveforms can be increased in accordance with the load. Furthermore, their power consumption is proportional to the clock frequency; so, trying to obtain a relatively high frequency results in high power consumption. In order to increase DPWMs' the sampling of the input data is to be made very fast so as to generate the fast switching PWM wave. In the last years we have the same techniques available using the analog devices but as they are not able to attain the frequency required for the minimizing of the vibrations in the DC motor we are trying for the same to be achieve using FPGA. However, they have lower linearity and non-monotonic behavior in some cases This paper proposes a DPWM generation. The synchronous block is counter based. However, the asynchronous block is not based on building a new delay-line structure, but using field-programmable gate array (FPGA) available delay locked loops (DLLs). This is the main difference from previous solutions, which develop their own delay line or DLL [8]–[15]. Almost every FPGA includes DLL blocks with phase-shifting features. Taking advantage of these resources, the proposed architecture obtains an excellent trade-off between linearity and time resolution. The proposed DPWM is, in principle, intended for FPGA implementation. An IC implementation would need the development of a DLL with phase-shifting capability, which is the most complex part of the architecture. However, as this block is already available in FPGAs, an FPGA implementation is very simple. Furthermore, the lowest cost FPGAs are suitable for the proposed DPWM, as shown in the experimental results.

The paper is organized as follows. The next section describes the generation of D PWM, while Section III shows the experimental results. Finally, Section IV gives the conclusions.

II. GENERATION OF PWM

Digital methods are the most suited form for designing PWM Generators. They are very flexible and less sensitive to environmental noise [2]. Also they are simple to construct and can be implemented very fastly. Most of the digital techniques employ counter and comparator based circuits.

Field Programmable Gate Array (FPGA) offers the most preferred way of designing PWM Generator for Power Converter Applications. They are basically interconnection between different logic blocks. When design is implemented on FPGA they are designed in such a way that they can be easily modified if any need arise in future. We have to just change the interconnection between these logic blocks. This feature of Re-programming capability of FPGA makes it suitable to make your design using FPGA [2]. Also using FPGA we can implement design within a short time. Thus FPGA is the best way of designing digital PWM Generators. Also implementation of FPGA-based digital control schemes proves less costly and hence they are economically suitable for small designs [4].

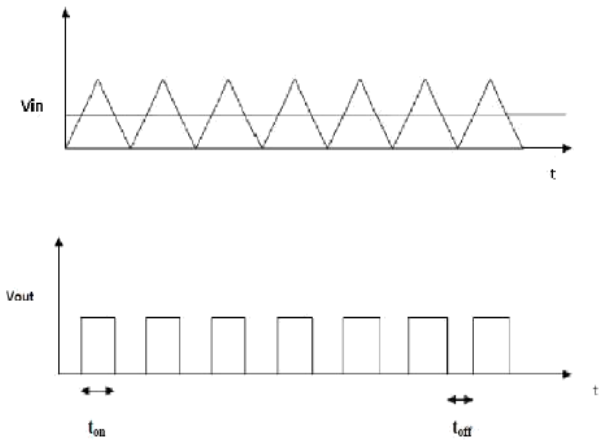

Fig. 1: High frequency Counter Based PWM Generator

This architecture was proposed by E.Koutroulis , A.Dollas and K.Kalaitzakis in [9]. According to this

architecture there is a high speed N-bit free running counter whose output is compared with register output; which stores desired input duty cycle(N-bit value); with the help of comparator. The comparator output is set equal to 1 when both these values are equal. This comparator output is used to set RS latch .The overflow signal from counter is used to reset RS latch. The output of RS latch gives the desired PWM output. This overflow signal is also used to load new N-bit duty cycle in Register.

The advantage of these method is that it is used to generate High-frequency PWM output which is not possible in normal counter based approach. **Fig.2** shows the corresponding block diagram of this architecture.

III. PROPOSED DPWM ARCHITECTURE

A. DLL Block:

The key of the proposed DPWM architecture is that it takes advantage of the advanced DLL features that are available in almost every FPGA nowadays. Digital devices such as FPGAs have specific blocks that can manage clock signals: DLL or phase-locked loop (PLL). Using these DLLs or PLLs, it is possible to multiply or divide the clock frequency. Many of these DLLs can also generate four phase-shifted clocks (shifted 0◦,90◦, 180◦, and 270◦) directly (Spartan and Virtex families of Xilinx) or allow to generate phase-shifted versions of the clock(Cyclone and Stratix families of Altera, ProAsic3, Fusion, or Accelerator families of Actel).

The first feature of these DLLs that is used in the proposed DPWM is multiplying the clock frequency. The advantage of doing so is that a high clock frequency can be internally used in the DPWM, while an external lower frequency is generated and also used in the rest of the digital controller. This is done for two main purposes. It is difficult to drive the package pin and the printed circuit board (PCB) line at very high frequencies due to their sizes, which are orders of magnitude above the size of internal chip connections (millimeter or centimeter instead of micrometer). Size is also responsible for the second purpose, which is decreasing power consumption. The parasitic capacitance of each element is proportional to its size, and the power consumption is also proportional to the parasitic capacitance. Internal clock multiplication is a well-known and widely spread digital technique (i.e., no gigahertz clock signal can be found in the PCB of a laptop or desktop computer because the external clock is multiplied inside the microprocessor). Clock multiplication is proposed for the DPWM. In the implementation shown in the experimental results, a 32 MHz external clock is multiplied by 4 in order to obtain a 128 MHz internal clock used in the DPWM. Thanks to the clock multiplication, time resolution increases from31.25 to 7.81 ns.

However, the rest the low clock frequency for decreased power consumption and easier design (longer critical paths in these blocks are valid), as shown in Fig. 1. This is very useful, given that a counter-based DPWM is very simple and can work at high frequencies, but other blocks in the controller are usually not so simple and need lower frequencies. Therefore, different clock frequencies are proposed in the architecture: the high-frequency clock is used for the DPWM and the low-frequency clock for the rest of the controller .However, the main contribution of the proposed DPWM comes from another DLL feature. Most DLLs in FPGAs also generate phase-shifted versions of the output clock. In many of these DLLs, four clocks shifted 0◦, 90◦, 180◦, and 270◦ are available. This allows us to multiply time resolution by 4 (two additional bits) beyond the maximum resolution achievable with a counter-based technique. If more phase-shifted clocks were available, the proposed architecture could obtain a higher resolution, multiplying the resolution of a counter-based solution by the number of available clocks. Anyhow, using phase-shifted clocks implies using both synchronous and asynchronous techniques that are explained with more detail in the next sections.

B. Synchronous Block:

The synchronous block is a counter-based DPWM that uses the $n - 2$ most significant bits (MSBs) of the duty cycle, $d[n - 1, 2]$, n being the total number of bits. As can be seen in Fig. 2, the synchronous block is based on a counter and comparison structure, which resembles analog PWMs based on saw-tooth signals. The functionality of this block is the following: if the duty cycle command is over the counter value, which is equivalent to a saw-tooth signal, the output is in the ON-state, and when the counter reaches duty cycle the output is turned OFF. This is a simple block, and therefore, it can work at high clock frequencies.

$$\text{Resolution} = fclk / fsw$$

Where *fclk* is the clock frequency and *fsw* is the switching frequency.

C. Time Distribution:

The two LSBs, $d[1,0]$, are used in the asynchronous block. These two bits are used to select between the four phase-shifted clocks generated by the FPGA's DLL. In fact, these clocks are combined using AND gates (see Fig. 3) in order to obtain other four phase-shifted signals that are high only a quarter of a cycle instead of half a cycle (see Fig. 4).The basic idea of using these four phase-shifted signals is obtaining four possible switching instants during each clock cycle. Therefore, resolution is multiplied by 4. In general, using m asynchronous bits (and $2m$ phase-shifted clocks), the total resolution is calculated as:

$$\text{Resolution} = 2^m \, fclk / fsw$$

where m is the number of asynchronous bits, *fclk* is the clock frequency, and *fsw* is the switching frequency.

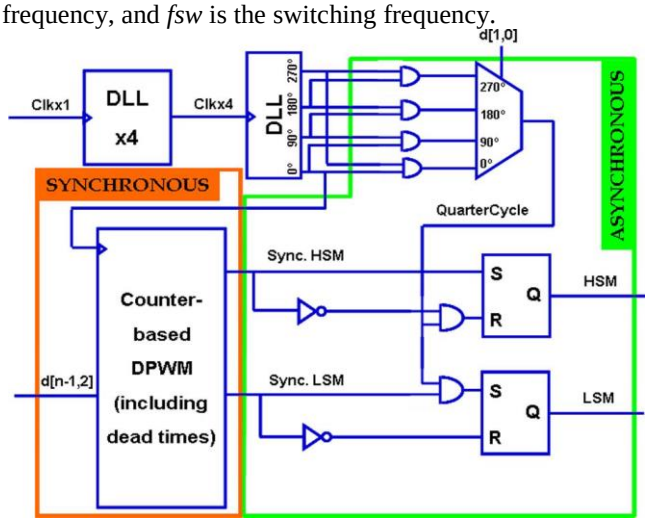

Fig. 2: Proposed DPWM architecture (basic outline).

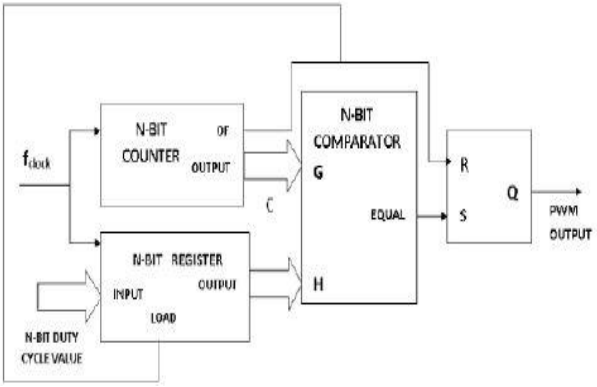

Fig. 3: Architecture of PWM Generator

IV. PWM CONTROL OF DC MOTOR

The application of PWM control in a Motor (DC/AC) is shown in Fig. 2. The PWM control signal, VPWM in Fig. 2, is generated from PWM generator. This VPWM is logically ANDED with rectangular pulse waveform coming from pulse generator and is fed to power switches S1 and S3.

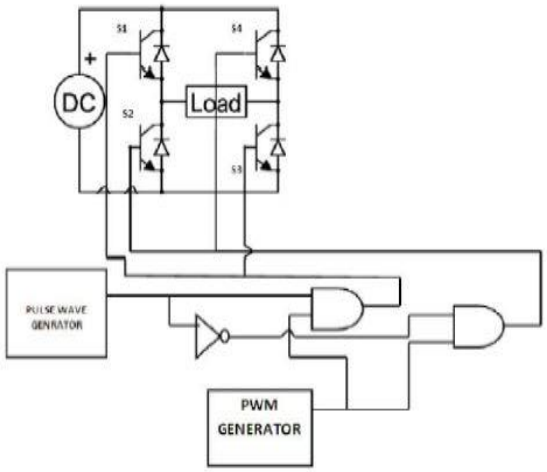

Fig. 4: The PWM control of Inverter

The inverted rectangular waveform is logically ANDED with PWM waveform and is fed to power switches S2 and S4. Thus ON and OFF time of power switches are controlled by this PWM control signal to The power switch is usually of MOSFET or IGBT. The size of Inverter depends on size of these power switches.

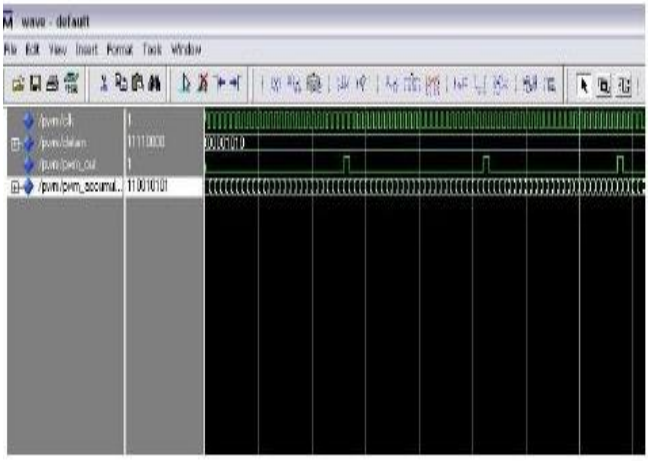

Fig. 5: Simulation Result of PWM

Since frequency of operation is inversely dependent upon Inverter size so we have to increase the switching frequency to reduce the Inverter size [1]. So we have to look into the frequency aspect of PWM Generator used so that we get optimized size of Inverter by proper selection of frequency of PWM wave.

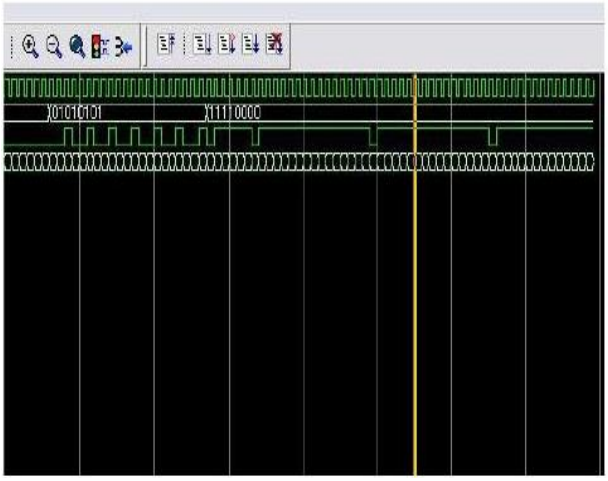

Fig. 6: Simulation Result of PWM

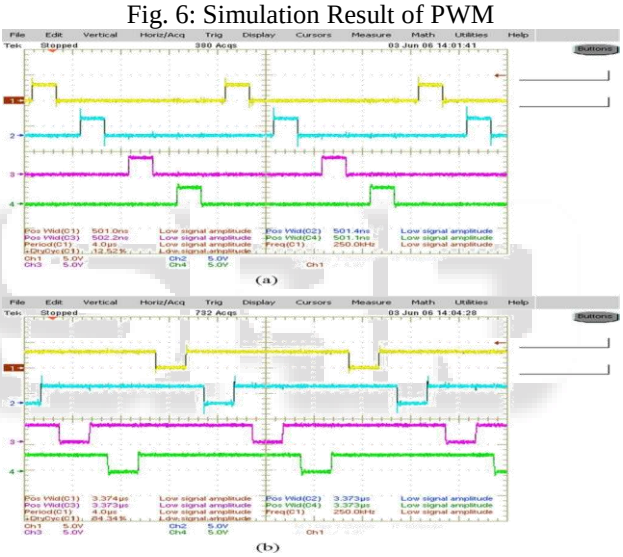

Fig.7: Expected output

V. CONCLUSION

A new hybrid counter-asynchronous DPWM architecture has been proposed. This DPWM, which is easy to design is mainly intended for FPGA implementation. The DLL raises the resolution of the DPWM. The external clock frequencies internally multiplied for a higher resolution**.**

The DLL raises the resolution of the DPWM in two ways. The external clock frequency is internally multiplied for a higher resolution of the counter-based block of the DPWM. Once the maximum possible resolution is achieved in the synchronous block, it is multiplied by 4 using four phase-shifted clock outputs of the DLL.

REFERENCES

[1] High-frequency pulse width modulation implementation using FPGA and CPLD ICs Journal of Systems Architecture 52 (2006) 332–344

[2] *IEEE Trans. Power Electron. (Spec. Issue Digit. Contr. Power Electron.)*, vol. 18, no. 1, Jan. 2003.

[3] B. J. Patella, A. Prodic, A. Zirger, and D. Maksimovic, "High-frequency digital controller IC for dc–dc converters," *IEEE Trans. Power Electron.*, vol. 18, no. 1, pp. 438–446, Jan. 2003.
[4] Peterchev, J. Xiao, and S. R. Sanders, "Architecture and IC implementation
[5] V. Yousefzadeh and D. Maksimovic,"Sensorless optimization of dead times in dc–dc converterswith synchronous rectifiers," *IEEE Trans.Power Electron.*, vol. 21, no. 4, pp. 994–1002, Jul. 2006.
[6] V. Yousefzadeh, N. Wang, Z. Popovic, and D. Maksimovic, "A digitally controlled dc/dc converter for an RF power amplifier," *IEEE Trans. Power Electron.*, vol. 21, no. 1, pp. 164–172, Jan. 2006.
[7] A. V. Peterchev and S. R. Sanders, "Quantization resolution and limit cycling in digitally controlled PWM converters," *IEEE Trans. Power Electron.*, vol. 18, no. 1, pp. 301–308, Jan. 2003.
[8] H. Peng, A. Prodic, E. Alarcon, and D. converters," *IEEE Trans. Power Electron.*, vol. 22, no. 1, pp. 208–215, Jan. 2007.
[9] A. P. Dancy and A. P. Chandrakasan, "Ultra low power control circuits for PWM converters," in *Proc. IEEE Power Electron. Spec. Conf. (PESC)*, St. Louis, Missouri, Jun. 1997, vol. 1, pp.21–27.
[10] E. O'Malley and K. Rinne, "A programmable digital pulse width modulator providing versatile pulse patterns and supporting switching frequencies beyond 15 MHz," in *Proc. IEEE Appl. Power Electron. Conf. Expo (APEC)*, Anaheim, CA, Feb. 2004, vol. 1, pp. 53–59.
[11] Z. Lukic, K. Wang, and A. Prodic,"High-frequency digital controller for dc– dc converters based on multi-bit sigma-delta pulse-width modulation," in *Proc. IEEE Appl. Electron. Conf. Expo (APEC)*, Austin, TX, Mar. 2005, vol. 1, pp. 35–40.
[12] R. F. Foley, R. C. Kavanagh, W. P.Marnane, and M. G. Egan, "An areaefficient digital pulsewidthmodulation architecture suitable for FPGA implementation," in *Proc. IEEE Appl. Power Electron. Conf. Expo (APEC)*, Austin, TX, Mar. 2005, vol. 3, pp. 1412–1418. pulsewidth modulator," *IEEE Trans. Power Electron.*, vol. 21, no. 3, pp. 842– 846, May 2006.
[13] K.Wang, N. Rahman, Z. Lukic, andA. Prodic, "All-digitalDPWM/DPFM controller for low-power dc–dc converters," in *Proc. IEEE Appl. Power Electron. Conf. Expo (APEC)*, Dallas, TX, Mar. 2006, pp. 719–723.
[14] V. Yousefzadeh, T. Takayama, and D. Maksimovic, "Hybrid DPWM with digital delay-locked loop," in *Proc. IEEE Comput. Power Electron. Workshop (COMPEL)*, Troy, NY, Jul. 16–19, 2006, pp. 142–148.
[15] A. Syed, E. Ahmed, D.Maksimovic, and E. Alarcon, "Digital pulse width modulator architectures," in *Proc. IEEE Power Electron. Spec. Conf. (PESC)*, Aachen, Germany, Jun. 2004, vol. 6, pp. 4689–4695.
[16] Koutroulis E., Dollas A. and Kalaitzakis K., "High-frequency pulse width modulation implementation using FPGA and CPLD ICs", Journal of Systems

