Drive Control High Resolution PWM Creation Using FPGA And Observance On Digital CRO

Authors

Prof. C. S. Patil¹, S. M. Mansute², R. S. Shimpi³

¹HOD, Department of E&TC, Shri. Gulabrao Deokar College of Engineering, Jalgaon, India.
²PG Student, Department of E&TC, Shri. Gulabrao Deokar College of Engineering, Jalgaon, India
³Lecturer, Department of Electrical, Sandip Institute of Polytechnic, Nasik, India

Email- Patilcs777@gmail.com, swapnilmansute@gmail.com rameshwarshimpi@gmail.com

ABSTRACT

In the earlier days, requirement of power is limited to some extent so a to achieve that requirement a rheostat is connected in series with the motor so that current proportion should be adjusted through the motor., but since of this wasted power as heat in the resistor element. It was considered as an incompetent method, but reasonable because the overall power required for the motor was low. According to general AC power regulation there are also some methods like variable autotransformers, including the trademarked 'Autrastat' for theatrical lighting; and the Variac. All methods are costly but they are enough efficient. In this paper we proposes an architecture of a new digital pulse width modulation (DPWM) which has advanced characteristics as advantage of the field programmable gate arrays (FPGA), basically the delay-locked loop (DLLs) present in the almost each FPGA. In the DPWM architecture which combines together a synchronous (counter based) block with an asynchronous block for increased resolution without unreasonably increasing the clock frequency.

Keywords- Pulse width modulation (PWM), Digital control field-programmable gate arrays (FPGAs), least significant bits (LSBs), printed circuit board (PCB), delay-locked loop (DLLs)

I. INTRODUCTION

In various cases it is essential to vary the speeds of the operations like as motors while various operations at the same time operating in the industries. To have such type of PWM control we can vary the speed of the motor very linearly in stage of 80% to 100% of its maximum speed of the motor for the speed of motor using the high resolution. In AC drive for digital control has obtained great study attention due to their
various advantages\[1\] [4][5], such as low sensitivity to external factors or aging, ease for design and prototyping programmability, advanced controlling algorithms, reduced component count. As there are also some disadvantages as in case of digital control as two most important as the limited resolution and processing or sampling delay [5]. About the first factor, resolution is limited generally by the conversion from analog-to- digital converter (ADC) and the pulse-width modulation (PWM). However, the conversion of analog to digital i.e. ADC resolution is becoming a fewer important difficulty, thankfulness to the windowed ADC technique [4] and because PWM resolution needs to be high than ADC resolution for avoiding limit cycling[7].Traditional digital PWMs (DPWMs) are base on the counters. Due to the use of DPWMs is that they are very simple and obtain high linearity. But as they are highly linear their resolution cannot be very high, as the minimum time step is equal to the clock period of the counter. But to obtain high resolution results in high power consumption because their power consumption is proportional to the clock frequency. The solution on this to increase the DPWMs’ resolution, delay lines can be used. Given the awareness of obtaining high-resolution DPWMs, a lot of dissimilar architectures have been anticipated in the most recent years [14]. Although all these architectures dissimilar from each other, nearly all of them make use of delay lines.

### II. BLOCK DIAGRAM

![Block Diagram](image)

**Figure 1.** Block diagram for motor control of High resolution PWM

High resolution and low power consumption are obtain by DPWMs but they have lesser linearity and non-monotonic behavior in most of the cases. For achieving exchange between the advantages of counter-based and delay line DPWMs, hybrid architectures had as well been proposed. As there are higher the number of bits in the synchronous (counter based) part, the linearity achieved are higher. And concerning in other way,
the higher the number of bits in the asynchronous (delay line) part, the lower the power consumption. Concerning resolution, the same resolution of delay line DPWMs can generally be achieved by the least significant bits (LSBs) in the delay-line part of the circuit.

As in figure 1 diagram is designed for a creation of high resolution PWM wave which includes the basic blocks as High resolution generator, Comparators, Clock generator, Drivers. First of all we are going to generate the high resolution PWM in FPGA and applying that PWM to the Motor for controlling the speed of the motor. The obtained high resolution PWM can be apply for the Harmonic control for motor, SMPS, communication system, Inverter, UPS, Speed control of the motor, etc.

III. FUTURED DPWM ARCHITECTURE

A. DLL Block

![Diagram](image.png)

**Figure 2.** Dissimilar clock Frequencies for the DPWM and the rest of the controller.

The key of the future DPWM architecture is that it takes benefit of the advanced delay locked loop (DLL) features that are available in nearly every FPGA currently. Digital devices such as FPGAs have specific blocks that can handle clock signals of DLL or phase-locked loop (PLL). By the use of these DLLs or PLLs, it is likely to multiply or divide the clock frequency. Several of these DLLs can too generate four phase-shifted clocks (shifted 0°, 90°, 180°, and 270°) directly or allow generating phase-shifted versions of the clock.

The main role of the proposed DPWM comes from another DLL characteristic. Nearly all of the DLLs in FPGAs also produce phase-shifted versions of the output clock. In various of these DLLs, four clocks shifted 0°, 90°, 180°, and 270° are available. This will allows us to multiply time resolution by 4 (two additional bits) further than the maximum resolution reachable with a counter-based technique. If more phase-shifted clocks were obtainable, the future architecture could obtain a higher resolution, multiplying the resolution of a counter-based solution by the number of obtainable clocks.
The first characteristic of these DLLs that is used in the future DPWM is multiplying the clock frequency. The benefit of doing so is that a high clock frequency can be inside used in the DPWM, while an external lower frequency is generated and also used in the rest of the digital controller. This has been done for two main purposes as it is difficult to drive the package pin and the printed circuit board (PCB) line at very high frequencies due to their sizes, which is of the order of magnitude above the sizes of internal chip connections (millimeter or centimeter instead of micrometer). Another purpose is that size is also responsible for the which is going to decrease the power consumption. The parasitic capacitance of every element is proportional to its size and the parasitic capacitance is also proportional to the power consumption. By considering internal factors internal clock multiplication is a well-known and extensively spread digital technique. DPWM have been future for the clock multiplication. In the execution 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. A well awareness thanks to the clock multiplication, time resolution increases from 31.25 to 7.81 ns. However, the rest of the controller modules can work at the low clock frequency for decreased power consumption and easier design as shown in Figure 2. This is very helpful, given that a counter-based DPWM is very easy and can work at high frequencies, but other blocks in the controller are usually not so easy and require lower frequencies. Hence, dissimilar clock frequencies are future in the architecture the high-frequency clock is used for the DPWM and the low-frequency clock for the rest of the controller.

B. Synchronous Block
The synchronous block is a DPWM based on counter that makes use of the $n - 2$ most significant bits (MSBs) of the duty cycle, $d [n - 1, 2]$, $n$ being the total number of bits. As shown in Figure 3, the synchronous block is based on a counter and comparison structure, which resembles analog PWMs based on saw-tooth signals. The operability of this block is in the following: way as if duty cycle command is over the counter value, which is corresponding 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 so, it can work at high clock frequency.
frequencies. In the synchronous block, resolution is specified by both the clock and the switching frequencies and the resolution is obtained by:

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

Where $f_{\text{clk}}$ is the clock frequency and $f_{\text{sw}}$ is the switching frequency.

C. Time Distribution

The two LSBs, $d[1, 0]$, are used in the asynchronous block. The available two bits are used to choose between the four phase-shifted clocks generated by the FPGA’s DLL. In fact, these clocks are joint using AND gates for reference see figure 4 in order to obtain other four phase-shifted signals that are high only a quarter of a cycle instead of half a cycle for reference see figure 5. The basic idea of using these four phase-shifted signals is to obtain the four possible switching instants during every clock cycle. Therefore, resolution is multiplied by 4. In general, using $m$ asynchronous bits (and $2^m$ phase-shifted clocks), the total resolution is calculated as:

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

Where $m$ is the number of asynchronous bits, $f_{\text{clk}}$ is the clock frequency, and $f_{\text{sw}}$ is the switching frequency.

![Figure 4. Future DPWM architecture](image-url)

![Figure 5. Quarter Cycle signal depending on the value of $d [1, 0]$.](image-url)
D. Functionality of the DPWM Architectur

The future DPWM, as seen in Figure 4, is produced by two blocks a synchronous block and an asynchronous block. The output of the synchronous block (counter-based block using the external frequency multiplied by 4) sets the high side MOSFET (HSM) output of the DPWM depending on the MSBs of \( d \) (duty cycle command). The synchronous DPWM uses only the 0\(^{\circ}\) shifted clock, while the asynchronous block wants all of the four phase-shifted clocks. The future DPWM architecture is intentional for a synchronous multiphase buck converter, so it creates driving signals for together the HSMs and low side MOSFETs (LSMs) with programmable dead times, as shown in Figure 4. It can be easily adapted to any other SMPS topology. The thought is that the turn-on instant of the HSM is always coincident with a 0\(^{\circ}\) clock edge, while the turn-off instant can be at any of the four clock edges, depending on the LSBs. The opposite is done for the LSM: turned on with any of the four clock edges, while turned off with the 0\(^{\circ}\) shifted clock. We have shown in Figure 5, the asynchronous block generates the signal named [10]Quarter Cycle, which corresponds to a quarter of the clock cycle. This signal is turned on in the rising edge of one of the four phase-shifted clocks, depending on the value of the two LSBs of the duty cycle as we can see in figure 5. For the HSM, the output is reset when the synchronous block is already OFF and Quarter Cycle arrives. Hence, the HSM output is active an integer number of clock cycles plus 0–3 quarters of a cycle, depending on the two LSBs. The opposite is done for the LSM output. It is set in one of the four clock edges and reset always with the 0\(^{\circ}\) clock.

Though, using synchronous techniques involves delay problems. In this case, signal Quarter Cycle suffers from these problems its value corresponding to the 270\(^{\circ}\) clock also has a short ON time at the start of the 0\(^{\circ}\) clock we can see in Figure 5. This difficulty can produce a nonmonotonic behavior (duty cycle commands ending in “11” would be related to “00” commands). In order to avoid the nonmonotonic behavior, some changes in the DPWM architecture are futured.

![Figure 6. Proposed DPWM architecture for avoiding nonmonotonic behavior.](image)

E. Modifications for Avoiding the Nonmonotonic Behavior

As we have seen the problems of nonmonotonic behavior when the two LSBs of the duty cycle, \( d \), are “11”. These are the cases as, Quarter- Cycle turns ON after the rising edge of the 270\(^{\circ}\) clock, but is also on a short
period of the first quarter of the clock cycle due to asynchronous delays. With the purpose of guarantee monotonicity, some added resources are used for avoiding the problems caused by asynchronous delays. The proposed block diagram as seen in figure 6.

If the two LSBs of the duty cycle are in the range of “00” to “10,” monotonic behavior is not a difficulty. In that case, both the set signal and the reset signal before the AND gate come from a synchronous flip-flop driven by the 0° clock, as shown in Fig. 7 (---dash lines represent the 0° clock edges and .....dot lines the rest of the clock edges). This is true for the HSM output, while the LSM output uses the opposite for set and reset.

But, if the two LSBs of the duty cycle are equal to “11,” the reset signal before the AND gate comes from a RS register instead of a synchronous flip-flop. This RS register is set only after the 90° clock arrival, eliminating the problem of Quarter Cycle being on at some point in the first quarter of the clock cycle, as shown in Figure 8. Moreover, this added RS register is reset a clock cycle before the set signal is active, avoiding the difficulty of having both the set and reset signals active at the same instant. The same technique is used for the LSM output, changing the set and reset signals. As a conclusion, the future DPWM attain high resolution, thankfulness to the use of four phase-shifted clocks while maintaining high linearity and monotonic behavior. Resolution is better four times as compared to only counter-based DPWMs. Higher resolution could be achieved if more than four phase-shifted clocks were obtainable in such case, the same architecture could be used with an exception. The equivalent to Quarter Cycle signal would be ON for 1/N clock cycles, N being the number of clocks. If N increased, the ON time of this signal Could be smaller than the minimum ON time of RS registers inputs. In that case, the equivalent to Quarter Cycle should be ON for
2/\(N\) or more fractions of a clock cycle, rising with the suitable phase-shifted clock. Possible nonmonotonic behavior would be addressed in a similar way to the one to give explanation design process time and complexity, while a well-known and tested design given by the FPGA producer enhance reliability.[9]

An important idea is to be able to use the automatic place and route tools for converting the VHDL code into physical implementations. There are no difficulty for this in the synchronous block, but dissimilar delays in every data path of the asynchronous block decrease linearity of the proposed DPWM. In order to retain linearity as high as possible, manual place and route can be used in the asynchronous block. But, as easy design is one of the objective of the proposed DPWM, this can be avoided using only manual placement of some key elements, such as the RS output registers and Quarter Cycle multiplexer. Placing the creation of Quarter Cycle at like distances from the dissimilar RS output registers helps the automatic place and route tool to do the rest of the placement and routing while maintaining high linearity.

![Figure 9 Output on digital CRO When binary data: 11111111](image)

**Table 1. Testing for speed control of motor**

<table>
<thead>
<tr>
<th>Sr. No.</th>
<th>Binary Inputs</th>
<th>Speed In RPM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0 0 0 0 0 0 0 0</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>0 0 0 0 0 0 0 1</td>
<td>34</td>
</tr>
<tr>
<td>3</td>
<td>0 0 0 0 0 0 1 1</td>
<td>113</td>
</tr>
<tr>
<td>4</td>
<td>0 0 0 0 0 1 1 1</td>
<td>214</td>
</tr>
<tr>
<td>5</td>
<td>0 0 0 0 1 1 1 1</td>
<td>485</td>
</tr>
<tr>
<td>6</td>
<td>0 0 0 1 1 1 1 1</td>
<td>1286</td>
</tr>
<tr>
<td>7</td>
<td>0 0 1 1 1 1 1 1</td>
<td>1364</td>
</tr>
<tr>
<td>8</td>
<td>0 1 1 1 1 1 1 1</td>
<td>1387</td>
</tr>
<tr>
<td>9</td>
<td>1 1 1 1 1 1 1 1</td>
<td>1410</td>
</tr>
</tbody>
</table>
The table shows the no of binary input to the corresponding speed of the motor. Also it is seen from the figure 9, the output of the implemented FPGA which is observed on the digital storage oscilloscope When binary data: “11111111”. The out-put pulse time of PWM is varies from 47ms to 1.2 ns. This is very high in the PWM resolution.

IV. CONCLUSION

In this the project proposes a new architecture of creating high resolution DPWM. The Digital Pulse width Modulation gives a good quality and smooth control for the motor such that Ac motor can be control easily like their counterparts DC motors. This control gives a good control and which will generates less vibration for motor, also consume less power than the previous techniques of motor control. And we have also shown the observance on Digital CRO for Binary data as “11111111”. The out-put pulse time of PWM is varies from 47ms to 1.2 ns. This is very high in the PWM resolution.

In a DPWM architecture a new hybrid counter-asynchronous has been futured. This DPWM, which is easy to design, is mainly planned for FPGA execution, as it takes advantage of the internal DLL available in almost each FPGA nowadays. The DLL increase the resolution of the DPWM in two ways as 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 attain in the synchronous block, it is multiplied by 4 using four phase-shifted clock outputs of the DLL.

V. REFERENCES

1. Santa Concepci´on Huerta, Angel de Castro, Oscar Garc´ia, Jos´e A. Cobos FPGA based Digital pulse width modulator with time resolution under 2 ns, IEEE Transactions on power electronics, VOL. 23, NO. 6, NOVEMBER 2008