**International Journal of Electrical and Computer Engineering (IJECE)**
Vol. 11, No. 4, August 2021, pp. 3629~3638
ISSN: 2088-8708, DOI: 10.11591/ijece.v11i4.pp3629-3638 

# Modified digital space vector pulse width modulation realization on low-cost FPGA platform with optimization for 3-phase voltage source inverter

**Shalini Vashishtha[1], Rekha K. R.[2]**
[1]Department of ECE, SJB Institute of Technology Research Centre, Bangalore, India
[1]Visvesvaraya Technological University (VTU), Belagavi, India
[2]Department of ECE, SJB Institute of Technology, Bangalore, India

**Article Info**

***Article history:***

Received Sep 21, 2020
Revised Dec 26, 2020
Accepted Jan 13, 2021

***Keywords:***

Dead time
Digital space vector
FPGA
Phase-locked loop
Pulse width modulation
Sector calculation
VSI

**ABSTRACT**

The realization of power electronic applications on hardware is a challenging task. The digital control circuit strategies are used to overcome the analog control strategies by providing great flexibility with simple equipment and higher switching frequencies. In this manuscript, an area optimized, modified digital space vector (DSV) pulse width modulation is designed and realized on low-cost FPGA. The modified digital space vector pulse width modulation (DSVPWM) uses a phase-locked loop (PLL) to generate clocks using the digital clock manager (DCM). These DCM clocks are used in the DSVPWM module to synchronize the other sub-modules. The voltage generation unit generates the three-phase (3-Φ) voltages and is used in the alpha-beta generation and sector determination unit. The reference active vectors are made by the reference generation unit and used in switching time calculation. The PWM pulses are generated using switching time generation, and lastly, the dead time occurrence unit generates the final SVPWM gate pulses. The modified DSVPWM is synthesized and implemented on Spartan-3E FPGA. The modified DSVPWM utilizes 17% slices, works at 102.45 MHz, and consumes 0.070 W total power. The simulation results and the resource utilization of modified DSVPWM are represented in detail. The modified DSVPWM is compared with existing PWM approaches on different Spartan-series FPGAs with better chip area improvement.



***Corresponding Author:***

Shalini Vashishtha
Department of ECE,
SJB Institute of Technology Research Centre, Bangalore, India
Visvesvaraya Technological University (VTU)
Belagavi, India
Email: shalini.v@atria.edu

## 1. INTRODUCTION

The power converter plays a significant role in most power electronic devices to convert and deliver quality energy to the AC motor. The motor is controlled by the pulse width modulation (PWM) signals assigned to the MOSFET or power transistors' gates. The PWM based converter provides many advantages like easy to control and implement, consumes lower power, compatible with a modern microprocessor and other digital controllers. The symmetric PWM signals provide fewer harmonics in output current and voltages than asymmetric PWM signals [1]. Three popular PWM methods are available and used in most three-phase (3-Φ) voltage source inverters, namely, sinusoidal-PWM (SPWM), space vector-PWM

(SVPWM), and hysteresis-PWM (HPWM). The digital control schemes provide enormous advantages over the analog control schemes, including circuit modification, easy implementation, and vast functions to update the control circuits. The digital signal processors (DSP) provides low-cost, high-performance digital control PWM strategies. Still, bandwidth performance lagsneed additional hardware/software to improve the limited functions and control strategies [2, 3].

The hardware implementation of AC motor control with digital control methods is widespread in recent times, which reduces the software and system component costs. The very large scale integrated circuits (VLSI) based field programmable gate array (FPGA) provides attractive digital solutions like high density, programmable hard-wired features, reconfigurability features, and a lesser design cycle than any other digital devices [4]. In recent times, SVPWM techniques are used with a lot of multi-phase converters. [5, 6]. The SVPWM techniques are realized on FPGA, which provides enormous advantages over other hardware works, like Higher switching frequency capabilities, simple hardware with rapid prototyping, maintaining the synchronization between timing and logic modules, provides lesser harmonic content, and better DC utilization [7-9]. The SVPWM technique without using trigonometric functions improves the area resources to the multilevel converter for controlling the drive control peripherals [10].

The SVPWM technique with fixed-point realization provides greater flexibility to control the induction motors with low power and high performance on the FPGA platform [11]. The SVPWM using the bus-clamping technique [12] is designed to save the hardware resources on FPGA. The hardware resource-saving is achieved using a simple mechanism to judge the sectors and SVPWM generation without using trigonometric operation by the bus-clamping method. The 5-segment discontinues SVPWM method [13] is designed and realized on the FPGA platform. The discontinues SVPWM method achieves better switching frequency, lower current harmonic distortions, and low switching losses than the continuous SVPWM method. The five-phase sinusoidal PWM technique [14] is realized on the FPGA platform, which offers faster sampling frequency on on-chip and suitable to adopt in driver applications.

The simple SVPWM is modeled for 2-level voltage source inverter (VSI) using the Simulink tool [15] and later adopted on the hardware platform. The vector control module is incorporated in the SVPWM model to reduce the time consumption while generating the PWM gate pulses. The induction motor and VSI circuits use the triggering pulses to reduce the switching loss and harmonic distortions. The SVPWM is designed for a dual 3-level T-type converter [16] on an FPGA platform. The T-type converter outputs are applied to the Induction machine to realize the harmonic content and performance metrics. The random SVPWM [17] is designed on FPGA for three-phase VSI, which offers flexibility and extensibility than the DSP based designs. The reconfigurable PWM generator [18] is designed using a dedicated control mechanism and configurable registers for power electronic converter applications on the FPGA platform. Simplified SVPWM control mechanism [19] is designed for a 2-level three-phase VSI for educational platform using MATLAB GUI and FPGA. The user has to provide the DC voltage input and angle values, which provide SVPWM pulses. These pulses are sent to the FPGA device via serial communication to realize the output pulses in real-time. The real-time SVPWM technique [20] is designed for delta–inverter using a Xilinx system generator. The inverter output feed to the induction machine, which offers a high-quality load current. The induction motor (DTIM) drive with Dual-three phase [21] is designed using different SVPWM technique to realize harmonic content and better DC utilization.

In this manuscript, The modified DSVPWM is designed and implemented on low-cost Spartan-3E FPGA and used for Three-Phase (3Φ)- VSI applications. The optimization of the chip area for SVPWM is a challenging task because of its complex architecture. Most of the current work fails to improve the chip area on low-cost FPGA. The proposed work overcomes these issues with better area improvement. The fundamental principles of SVPWM techniques with mathematical equations are explained in section 1. The modified DSVPWM with detailed hardware architecture is explained in section 2. Section 3 provides simulation and synthesized results for DSVPWM and comparative analysis of proposed DSVPWM with existing methods with chip area improvements. The conclusion is highlighted in Section 4.

The typical three-phase VSI contains mainly six power switches ($S_1$-$S_6$) connected with the DC voltage controlled by switching variables and is represented in Figure 1. The upper switches ($S_1$, $S_3$, and $S_5$) and lower switches ($S_4$, $S_6$, and $S_2$), either on or off states are based on switching variables. The switches ($S_1$, $S_3$, and $S_5$) combinations generate the output voltage. The $S_1$ is connected with $S_4$; similarly, $S_3$is connected with $S_6$, and $S_5$ is connected with $S_2$. The three common points ($V_a$, $V_b$, and $V_c$) generate the output voltage. These voltage points are connected to any AC motor or induction motor appliances.

The SVPWM provides minimal harmonic distortion in the output voltage of the three-phase VSI by using proper switching sequences (Upper –$S_1$, $S_3$, and $S_5$), which can be used in any AC motor or induction motor to use an appropriate supply voltage. The SVPWM is implemented with the dq-reference frame's help, which contains a horizontal (α) and vertical (β) axis. The 3-Φ voltage vector ($V_a$, $V_b$, $V_c$) as a reference frame is transformed into the αβ -reference frame using the (1):

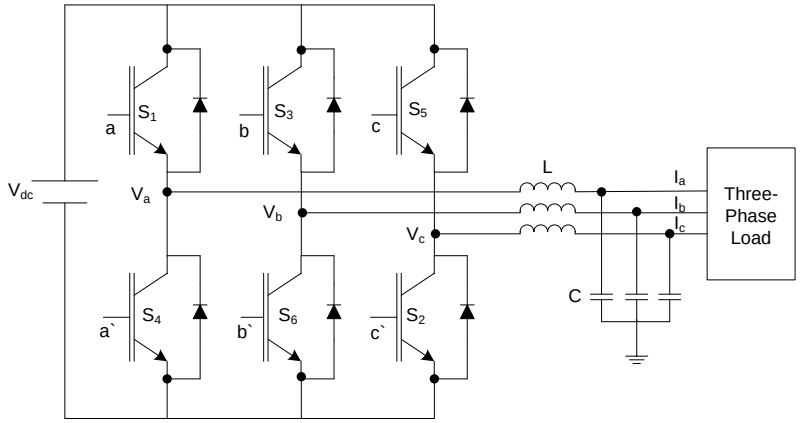


Figure 1 Typical voltage source inverter (VSI) diagram [11]

$$\begin{bmatrix} V_\alpha \\ V_\beta \end{bmatrix} = \frac{2}{3}\begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & \sqrt{3}/2 & -\sqrt{3}/2 \end{bmatrix}\begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} \quad (1)$$

The fundamental SVPWM- sectors and switching vectors are represented in Figure 2. The αβ-reference results in six active vector ($V_1$-$V_6$) and two null vectors ($V_0$ and $V_7$). A total of eight space vectors like $V_0$-$V_7$ are applied to output voltages to calculate the reference voltage vector $V_{ref}$ in αβ-plane (hexagonal plane).

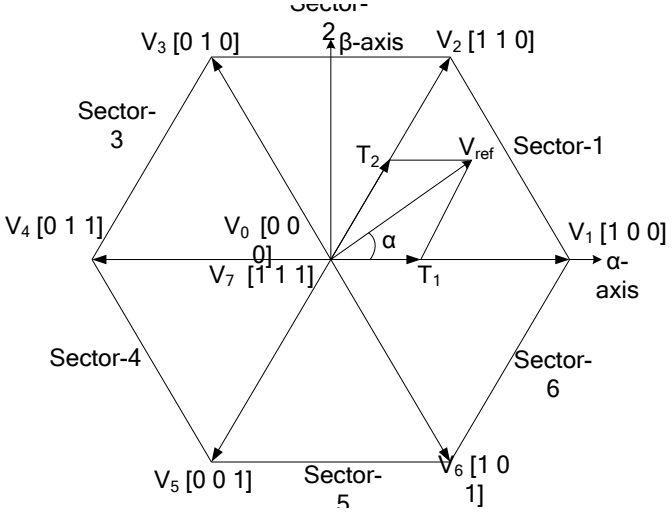


Figure 2. SVPWM sectors and switching vectors representation

The reference voltage vector ($V_{ref}$) and corresponding angle 'θ' are calculated using the (2) and (3):

$$\left|V_{ref}\right| = \sqrt{V_\alpha{}^2 + V_\beta{}^2} \quad (2)$$

$$\theta = \tan^{-1}\left(\frac{V_\beta}{V_\alpha}\right) = \omega_s t = 2\pi f_s t \quad (3)$$

where *fs* is fundamental frequency and *t* is period.

The SVPWM is designed and implemented using three significant steps: i) Calculate the $V_\alpha$, $V_\beta$, $V_{ref}$, and θ, ii) generate time durations $T_0$, $T_1$, and $T_2$, iii) create the switching time for each switch ($S_1$-$S_6$).

## 2. MODIFIED DSVPWM ARCHITECTURE

The proposed modified DSVPWM hardware architecture is represented in Figure 3 and explained in this section. The DSVPWM mainly contains phase-locked loop (PLL) module, 10 MHz clock unit, clk_svpwm generation unit, 3-Φ voltage generation (VG) unit, Alpha-Beta generation (ABG) unit, Sector generation (SG) unit, reference vector generation (RVG) unit, Switching time generation (STG) unit, SVPWM gate pulse generation (GPG) unit, and dead time calculation (DTC) unit. The individual DSVPWM architecture submodules are explained in detailed in this section and used for 3-Φ voltage source Inverter (VSI) applications.

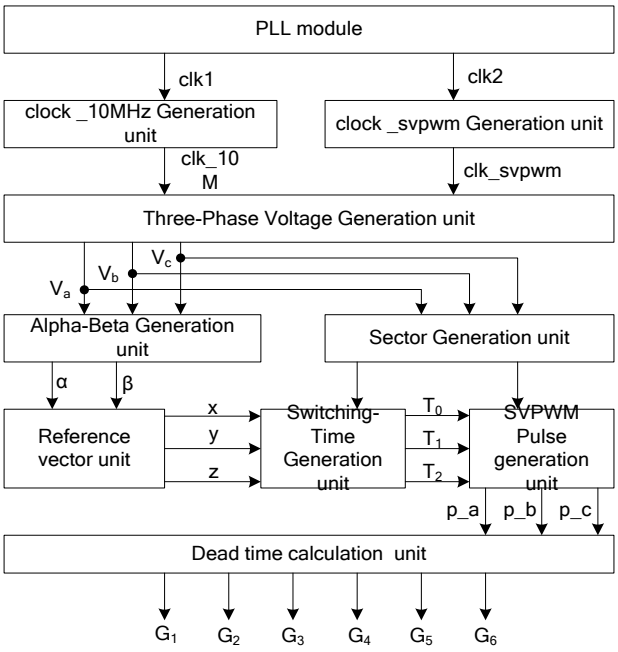


Figure 3. Detailed hardware architecture of DSVPWM

The phase-locked loop (PLL) is designed using digital clock manager (DCM) or Xilinx clocking wizard at an input clock frequency of 37.5 MHz. The PLL generates the two outputs, clock1 (clk1) and clock 2 (clk2). The clk1 provides a dedicated frequency synthesizer and fully-digital output to the DCM and provides the feedback clock output, working in low-frequency mode. The clock2 (clk2) is a simple buffered clock signal, which operates the same as the input clock. The clk1 works at 20 MHz and clk2 work at 37.5 MHz. The clk1 is input to the clock_10 MHz generation unit and generates the 10 MHz clock output by toggling the input clock (clk1). The clk2 is input to the clk_svpwm generation unit and produces the 150 KHz clock output. The clock frequency of clk_svpwm is always less than the clock_10 MHz generation unit.

The 3-Φ voltage generation (VG) unit receives the clk_10 MHz and clk_svpwm as clock inputs and performs the 3-Φ voltage generation. The clk_svpwm is used for the address generation, and the clk_10 MHz is used for the sinewave generation using block random-access memory (BRAM) with 3072 memory locations. Each phase voltage is set with specific counter values. Each phase is updated in BRAM and generates the 3-Φ phase outputs ($V_a$, $V_b$, and $V_c$).

Alpha-Beta generation (ABG) unit receives the 3-Φ voltages $V_a$, $V_b$, and $V_c$. It performs the following equations using right shift operators for $V_\alpha$ (Alpha) and $V_\beta$ (Beta) in signed digit format. The (4) and (5) represents the $V_\alpha$ (Alpha) and $V_\beta$ (Beta) calculation, as shown [22].

$$V_\alpha = \frac{2}{3} * (V_a - 0.5V_b - 0.5V_c) \tag{4}$$

$$V_\beta = \frac{2}{3} * (\frac{\sqrt{3}}{2} V_b - \frac{\sqrt{3}}{2} V_c) \tag{5}$$

The sector generation (SG) unit also receives the 3-Φ voltages like $V_a$, $V_b$, and $V_c$ along with DC voltage and generates the active reference vectors (A [2], A [1], and A [0]). The active vector generation and sector determination are tabulated in Tables 1 and 2 respectively.

Table 1. Active vector generation

| Equation | A[2] | A[1] | A[0] |
|---|---|---|---|
| $(V_a\text{-}V_b) \leq V_{dc}$ | 1 | 0 | 0 |
| $(V_b\text{-}V_c) \leq V_{dc}$ | 0 | 1 | 0 |
| $(V_c\text{-}V_a) \leq V_{dc}$ | 0 | 0 | 1 |

Table 2. Sector determination

| A[2]A[1]A[0] | Sector |
|---|---|
| 010 | 1 |
| 100 | 2 |
| 110 | 3 |
| 001 | 4 |
| 011 | 5 |
| 101 | 6 |

The active vector A [2] is activated only when $(V_a\text{-}V_b) \leq V_{dc}$ else A [1] or A [0] is activated. Similarly, if $(V_b\text{-}V_c) \leq V_{dc}$, then an active vector A [1] is activated; otherwise, A [0] or A [2] is activated. If $(V_c\text{-}V_a) \leq V_{dc}$, then active vector A [0] is activated; otherwise, A [1] or A [2] is activated. Based on the reference active vectors, each sector is calculated as per Table 1.

The reference vector generation (RVG) unit is designed using DC voltage ($V_{dc}$) and sampling time ($T_s$) along with Alpha-Beta ($V_\alpha$ and $V\beta$) values. The reference vector values (X, Y, and Z) are calculated using (6) in signed digit format with right shift operations [13]. These reference vector values are used for the formation of switching time generation in each sector.

$$X = \sqrt{3} * V_\beta * \left(\frac{T_s}{V_{dc}}\right)$$
$$Y = \sqrt{3} * T_s * (\sqrt{3} * V_\alpha + V_\beta) * \left(\frac{V_{dc}}{2}\right)$$
$$Z = \sqrt{3} * T_s * (-\sqrt{3} * V_\alpha + V_\beta) * \left(\frac{V_{dc}}{2}\right) \quad (6)$$

The switching time generation (STG) unit generates the time duration $T_0$, $T_1$, and $T_2$ based on each Sector and is tabulated in Table 3. The $T_0$ is calculated using X, Y, and Z reference vector values and sample time $T_s$. Similarly, $T_1$ and $T_2$ are calculated by assigning the proper X, Y, and Z reference vector values based on the sector. The switching time $T_0$ is right-shifted by two times, which means the multiplication of ¼. The switching time $T_1$ and $T_2$ are right-shifted by one time, which means the multiplication of ½ with corresponding reference active vectors.

Table 3. Switching time generation for each sector

| Sector | $T_0$ | $T_1$ | $T_2$ |
|---|---|---|---|
| 1 | $(T_s – Z – X)>>2$ | $Z>>1$ | $Y>>1$ |
| 2 | $(T_s – Y + X)>>2$ | $Y>>1$ | $–X>>1$ |
| 3 | $(T_s + Z – X)>>2$ | $–Z>>1$ | $X>>1$ |
| 4 | $(T_s + X – Y)>>2$ | $–X>>1$ | $Z>>1$ |
| 5 | $(T_s – X + Y)>>2$ | $X>>1$ | $–Y>>1$ |
| 6 | $(T_s + Y + Z)>>2$ | $–Y>>1$ | $–Z>>1$ |

SVPWM gate pulse generation (GPG) unit generates the PWM gate pulses using a PWM counter and switching times. The PWM counter should be less than the sampling time $T_s$. The three PWM gate pulses (PWM_A, PWM_B, and PWM_C) for sector-1 are tabulated in Table 4. The switching time combinations like $T_0$, $T_0+T_1$, $T_0+T_1+T_2$, $3T_0+T_1+T_2$, $3T_0+T_1+2T_2$, and $3T_0+2T_1+2T_2$ are equal to the PWM counter value

for the generation of three PWM gate pulses. If the switching combination $T_0+T_1+T_2$ is equal to the PWM counter value, then the PWM_A=1, PWM_B=1, and PWM_C =1, which is common in all the sector. Similarly, If the switching combination $3T_0+2T_1+2T_2$ is equal to the PWM counter value, then the PWM_A=0, PWM_B=0, PWM_C=0, which is common in all the sectors. The gate pulses (PWM_A, PWM_B, and PWM_C) values are different in switching combinations $T_0$, $T_0+T_1$, $3T_0+T_1+T_2$, and $3T_0+T_1+2T_2$ for each sector.

Table 4. Three PWM gate pulse generation for sector-1

| PWM counter = | PWM_A | PWM_B | PWM_C |
|---|---|---|---|
| $T_0$ | 0 | 1 | 0 |
| $T_0+T_1$ | 1 | 1 | 0 |
| $T_0+T_1+T_2$ | 1 | 1 | 1 |
| $3T_0+T_1+T_2$ | 1 | 1 | 0 |
| $3T_0+T_1+2T_2$ | 0 | 1 | 0 |
| $3T_0+2T_1+2T_2$ | 0 | 0 | 0 |

Deadtime calculation (DTC) unit introduces the dead time and generates the final six SVPWM gate pulses ($G_1$-$G_6$) using, Three PWM gate pulses (PWM_A, PWM_B, and PWM_C). The six individual counters are used for each Gate pulses $G_1$- $G_6$ generation; if PWM_A=1, and after 2µs, the gate pulse $G_1$is activated, otherwise PWM_A=0, and after 2 µs, the gate pulse $G_4$ is activated. Similarly, If PWM_B=1, and after 2 µs, the gate pulse $G_3$is activated; otherwise, PWM_B=0, and after 2 µs, the gate pulse $G_6$is activated. If PWM_C=1, and after 2µs, the gate pulse $G_5$is activated; otherwise, PWM_C=0, and after 2 µs, the gate pulse $G_2$ is activated. The time difference of 2 µs is set for each gate pulses using six individual counters. The $G_1$, $G_3$, and $G_5$ appear at upper switching pulse patterns, whereas $G_4$, $G_6$, and $G_2$ appear at low switching pulse patterns.

## 3. RESULTS AND DISCUSSION

The proposed modified DSVPWM module is designed and implemented on a low-cost Spartan-3E FPGA device. The Spartan-3E FPGA contains an XC3S250E device with a package of TQ144. The modified DSVPWM module is modeled using Verilog-HDL on Xilinx -14.7 ISE environment. The simulations are analyzed with valid test cases using the ISE simulator, and synthesized results are obtained after the place and routing operation in Xilinx ISE. The overall simulation results of modified DSVPWM gate pulses are represented in Figure 4. The DSVPWM process starts after the clock (clk) is activated with an active low reset (rst). The clk works at 37.5MHz. The six gate pulses ($G_1$, $G_3$, $G_5$, $G_4$, $G_6$, and $G_2$) and the corresponding sector are shown in the waveform. All the gate pulses are varied in each Sector and used for 3-phase VSI applications.

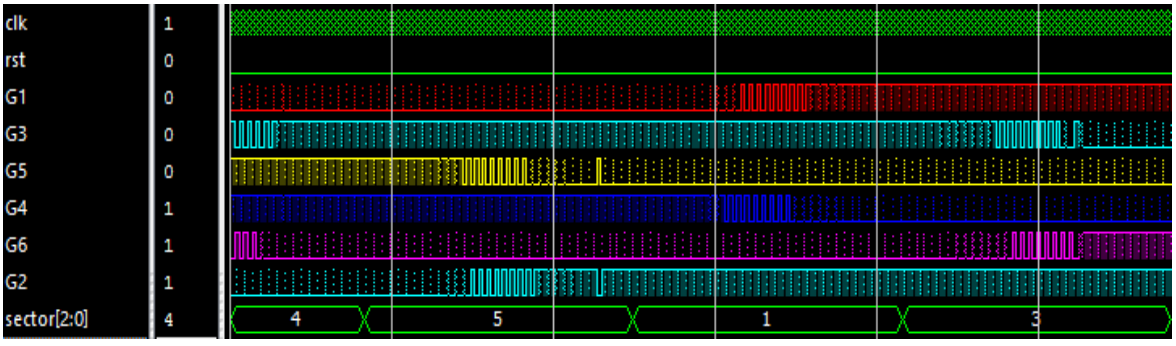


Figure 4. Simulation results of overall DSVPWM gate pulses

The DSVPWM gate pulses at sector-4 simulation results are represented in Figure 5. The gate pulses $G_1$, $G_3$, $G_5$, are inverted and represent in $G_4$, $G_6$, and $G_2$, respectively. The Gate pulses are generated based on the switching time generation. The dead time occurrence simulation is represented in Figure 6. The Three dead time occurrence is highlighted in the waveform.

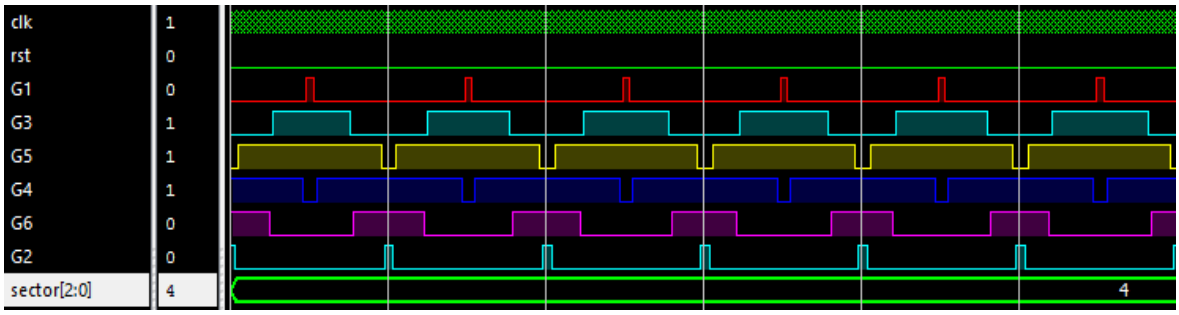


Figure 5. Simulation results of DSVPWM Gate pulses at sector-4

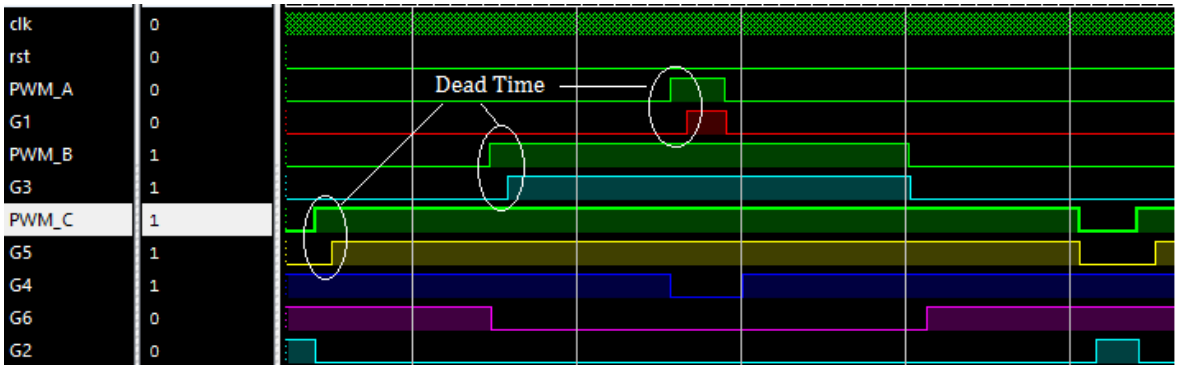


Figure 6. Simulation results of the dead time calculation unit

The dead time difference between PWM_A and output gate pulse $G_1$ is 2.11µs. Similarly, the PWM_B and output gate pulse $G_3$ dead time difference is 2.08 µs. The third one, the PWM_C and output gate pulse $G_5$ dead time difference, is 2.05 µs. The $G_4$, $G_6$, and $G_2$ are reflections of $G_1$, $G_3$, and $G_5$, respectively.

The resource utilization of modified DSVPWM architecture on Spartan-3E FPGA is tabulated in Table 5. The slices of 432 with 17% utilization, Slices-FFs of 188 with 3% utilization, 4-input LUTs of 810 with 16% utilization, block RAM (BRAM) of 3 with 25% utilization. The three BRAM used for sine wave generation in the three-phase voltage generation module. The five multipliers are used with 41% utilization. The four global clocks (GCLK's) are used, including the primary clock (clk), PLL generated clock, 10MHz clock, and SVPWM clock. The one digital clock manager (DCM) with 25% utilization is used to create PLL in DSVPWM architecture.

Table 5. Resource utilization of DSVPWM on Spartan-3E FPGA

| Logic Utilization | Used | Available | Utilization |
|---|---|---|---|
| Slices | 432 | 2448 | 17% |
| Slice Flip Flops | 188 | 4896 | 3% |
| 4 input LUTs | 810 | 4896 | 16% |
| Bonded IOBs | 8 | 108 | 7% |
| BRAMs | 3 | 12 | 25% |
| MULT18X18SIOs | 5 | 12 | 41% |
| GCLKs | 4 | 24 | 16% |
| DCMs | 1 | 4 | 25% |

The resource utilization of DSVPWM-Submodules on Spartan-3E FPGA are tabulated in Table 6. The resource utilization includes occupied slices, Slice-FFs, and 4-input LUT utilization for each DSVPWM Submodule. The 3-Phase voltage generation unit uses 87 slices, 92 slices FFs, and 135 LUTs. Similarly, the SVPWM pulse generation unit utilizes 156 slices, 19 slices FFs, and 295 LUTs. The 3-Phase voltage generation unit uses the BRAM module for sine value generation, and the SVPWM pulse generation unit matches the PWM counter value with switching values in each Sector. The other sub-modules utilize few chip area resources in DSVPWM architecture.

Table 6. Resource utilization of DSPWM-submodules on Spartan-3E FPGA

| DSVPWM submodules | Slices | Slice Flip Flops | Four input LUTs |
|---|---|---|---|
| Clk_10M gen.unit | 1 | 1 | 1 |
| Clk_svpwm gen. unit | 7 | 8 | 15 |
| 3-Phase voltage gen. Unit | 87 | 92 | 135 |
| Alpha-Beta gen. unit | 26 | NA | 46 |
| Sector gen.unit | 26 | NA | 51 |
| Reference gen. unit | 24 | NA | 48 |
| Switching Time gen. unit | 89 | NA | 171 |
| SVPWM Pulse gen. unit | 156 | 19 | 295 |
| Deadtime calculation unit | 33 | 36 | 54 |

The resource comparison of DSPWM architecture with existing PWM approaches [23-27] are tabulated in Table 7. For comparison, few parameters are considered the number of bits used in architectures, FPGA family with the device, Chip areas utilization like slice –FFs, 4-input LUTs and slices. The low-cost Spartan series FPGA based existing PWM methods are considered for comparison with DSVPWM architecture. The proposed DSVPWM architecture utilizes 188 Slice-FFs, 810 LUTs, 432 slices, and compared with existing PWM approaches.

The sinusoidal PWM [23] is implemented on Spartan-3 FPGA and uses 8-bits for SPWM modeling. The present DSVPWM resource utilization is improved by 44.8% in slice-FFs, 26.4 % in 4-input LUTs, and 41.3% in slices than the SPWM approach [23]. Similarly, The proposed DSVPWM resources like 4-input LUTs and slices utilized significantly less than the SVPWM [24]. The SVPWM [25] is designed in two versions: 8-bits and 12-bits for model creation on Spartan-3 FPGA.

The proposed DSVPWM architecture utilized fewer resources in terms of 70.3% in 4-input LUTs and 57.3% in slices than the SVPWM approach [25]. The SVPWM [26] is implemented on Spartan-2E FPGA and uses 12-bits for SVPWM modeling. The DSVPWM resource utilization is improved by 35.1% in 4-input LUTs and 34.6% in slices than the SVPWM approach [26]. The Proposed DSVPWM architecture utilized fewer resources in terms of 26% in slice-FF's, 7% in 4-input LUTs, and 8.6 % in slices than the SVPWM approach [27].

Table 7. Resource comparison of DSPWM with existing approaches [23-27]

| Resources | Bits | FPGA Family | Device | Slice -FFs | 4-Input LUTs | Slices |
|---|---|---|---|---|---|---|
| SPWM [23] | 8-bits | Spartan-3 | XC3S400PQ208 | 341 | 1102 | 737 |
| SVPWM [24] | 12-bits | Spartan-3E | XC3S500EFG320 | 165 | 5988 | 3374 |
| SVPWM [25] | 8-bits | Spartan-3 | XC3S400PQ208 | NA | 1365 | 734 |
| SVPWM [25] | 12-bits | Spartan-3 | XC3S400PQ208 | NA | 2730 | 1013 |
| SVPWM [26] | 12-bits | Spartan-2E | XC2S200EPQ208 | 120 | 1248 | 661 |
| SVPWM [27] | 12-bits | Spartan-3 | XC3S400PQ208 | 251 | 871 | 473 |
| DSVPWM (Proposed) | 12-bits | Spartan-3E | XC3S250ETQ144 | 188 | 810 | 432 |

## 4. CONCLUSION

In this manuscript, Area optimized, modified DSVWPM hardware architecture is designed and implemented on a low-cost Spartan-3E FPGA device. The modified DSVPWM architecture contains PLL, clock generation units at 10 MHz for SVPWM, followed by 3-Φ VG unit, ABG unit, SG unit, RVG unit, and STG, SVPWM-GPG unit, and DTC unit. All the submodules are instantiated properly in DSVPWM for the generation of six gate pulses. The simulation results for DSVPWM are highlighted using the ISE simulator. The modified DSVWPM is synthesized and generates the resource utilization summary after the place and route operation. The modified DSVWPM utilizes 17% slices, 3% slice Flip-flops (FFs) and 16% four-input LUT's on Spartan-3E FPGA. The modified DSVWPM architecture works at a design frequency of 102.45 MHz and consumes a total power of 0.070 W. The modified SVPWM architecture is compared with similar PWM approaches with an average improvement of 30-40% in chip area utilization on different Spartan series FPGA's.

## REFERENCES

[1] Z. Yu, A. Mohammed, and I. Panahi, "A review of three PWM techniques," in *Proceedings of the 1997 American Control Conference*, 1997, vol. 1, pp. 257–261.

[2] Y. Y. Tzou and H. J. Hsu, "FPGA realization of space-vector PWM control IC for three-phase PWM inverters," *IEEE Transactions on power electronics*, vol. 12, no. 6, pp. 953–963, 1997.

[3] S. R. Patil and V. N. Kalkhambkar, "Hybrid space vector pulse width modulation voltage source inverter-a review," in *2017 International Conference on Data Management, Analytics, and Innovation (ICDMAI)*, 2017, pp. 200–204.
[4] R. Rajendran and N. Devarajan, "Analysis and FPGA realization of a pulse width modulator based on voltage space vectors," *International Journal of Computer Applications*, vol. 2, no. 6, pp. 46-51, 2010.
[5] J. Alvarez, O. Lopez, F. D. Freijedo, and J. Doval-Gandoy, "Digital parametrizable VHDL module for multilevel multiphase space vector PWM," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 9, pp. 3946–3957, 2010.
[6] R. Foley, R. Kavanagh, W. Marnane, and M. Egan, "Multi-phase digital pulse width modulator," *IEEE Transactions on Power Electronics*, vol. 21, no. 3, pp. 842–846, 2006.
[7] Y. Wang and U. Schaefer, "Real-time simulation of an FPGA based space vector PWM controller," in *SPEEDAM 2010*, 2010, pp. 833–838.
[8] J. Somlal, M. V. G. Rao, and S. P. Karthikeyan, "Experimental investigation of an indirect current-controlled Fuzzy-SVPWM based Shunt Hybrid Active Power Filter," in *2016 IEEE Region 10 Conference (TENCON)*, 2016, pp. 801–806.
[9] A. O. Rait and P. Bhosale, "FPGA implementation of space vector PWM for speed control of 3-phase induction motor," in *2011 International Conference on Recent Advancements in Electrical, Electronics and Control Engineering*, 2011, pp. 221–225.
[10] D. Janik, T. Kosan, J. Sadsky, and Z. Peroutka, "Implementation of SVPWM algorithm without trigonometric functions," in *2014 International Conference on Applied Electronics*, 2014, pp. 131–134.
[11] M. D. Patil, D. Shah, and A. Kadam, "FPGA implementation of SVPWM control technique for three-phase induction motor drive using fixed point realization," in *2014 International Conference on Circuits, Systems, Communication, and Information Technology Applications (CSCITA)*, 2014, pp. 93–98.
[12] T. Sutikno, "Hardware-Resource Saving For Realization of Space Vector PWM Based on FPGA Using Bus-Clamping Technique," *TELKOMNIKA (Telecommunication, Computing, Electronics and Control)*, vol. 7, no. 3, pp. 161-168, 2009.
[13] T. Sutikno, A. Jidin, and N. R. N. Idris, "New approach FPGA-based implementation of discontinuous SVPWM," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 18, no. 4, pp. 499–514, 2010.
[14] T. Sutikno, N. R. N. Idris, I. M. Alsofyani, A. Jidin, and L. L. Raj, "FPGA based five-phase sinusoidal PWM generator," in *2012 IEEE International Conference on Power and Energy (PECon)*, 2012, pp. 314–318.
[15] E. Lotfi, M. Elharoussi, and E. Abdelmounim, "VHDL Design and FPGA Implementation of the PWM Space Vector of an AC Machine Powered by a Voltage Inverter," in *Proceedings of the Mediterranean Conference on Information & Communication Technologies 2015*, 2016, pp. 41–47.
[16] A. Salem, F. De Belie, and J. Melkebeek, "A novel space-vector PWM computations for a dual three-level T-type converter applied to an open end-winding induction machine," in *2016 Eighteenth International Middle East Power Systems Conference (MEPCON)*, 2016, pp. 633–638.
[17] T. Pu, F. Bu, W. Huang, and L. Zhu, "Implementation of random SVPWM strategy for three-phase voltage source inverter based on FPGA," in *2017 20th International Conference on Electrical Machines and Systems (ICEMS)*, 2017, pp. 1–4.
[18] A. M. Khiavi, J. Sobhi, Z. D. Koozehkanani, and M. F. Kangarlu, "FPGA-based reconfigurable PWM generator for power electronic converter applications," *Journal of Control, Automation, and Electrical Systems*, vol. 28, no. 4, pp. 516–531, 2017.
[19] R. C. García, I. E. S. Ono, V. dos S. Fahed, and J. O. P. Pinto, "Simplified educational platform for SVPWM control of a two-level three-phase inverter using Matlab GUI and FPGA," in *2017 Brazilian Power Electronics Conference (COBEP)*, 2017, pp. 1–6.
[20] A. Alouane, A. Ben Rhouma, M. Hamouda, and A. Khedher, "Efficient FPGA-based real-time implementation of an SVPWM algorithm for a delta inverter," *IET Power Electronics*, vol. 11, no. 9, pp. 1611–1619, 2018.
[21] K. A. Chinmaya and G. K. Singh, "Experimental analysis of various space vector pulse width modulation (SVPWM) techniques for dual three-phase induction motor drive," *International Transactions on Electrical Energy Systems*, vol. 29, no. 1, pp. 1-15, 2019.
[22] A. Donisi, L. Di Benedetto, G. D. Licciardo, A. Rubino, E. Piccirilli, and E. Lanzotti, "A Fully FPGA Implementation of SVPWM for Three-phase Inverters without External Reference Signals," in *2020 IEEE International Conference on Environment and Electrical Engineering and 2020 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe)*, 2020, pp. 1–5.
[23] R. K. Pongiannan, S. Paramasivam, and N. Yadaiah, "Dynamically reconfigurable PWM controller for three-phase voltage-source inverters," *IEEE Transactions on Power Electronics*, vol. 26, no. 6, pp. 1790–1799, 2010.
[24] M. J. Sumam and G. Shiny, "Rapid prototyping of high-performance FPGA controller for an induction motor drive," in *2018 8th International Conference on Power and Energy Systems (ICPES)*, 2018, pp. 76–80.
[25] R. K. Pongiannan, S. Paramasivam, and N. Yadaiah, "FPGA realization of an area-efficient SVPWM modulator for three-phase induction motor drive," *International Journal of Power Electronics*, vol. 2, no. 2, pp. 176–199, 2010.
[26] R. Subha, N. Kumar, N. Kaur, Pragathi M. S., and S. Agrawal, "FPGA implementation of an efficient space vector pulse width modulation Algorithm," *Elixir Elec. Engg.*, vol. 38, pp. 4506–4509, 2011.
[27] M. S. Kannan and S. Vasantharathna, "Dynamically Reconfigurable Control Structure for Asynchronous AC Drives," *International Journal of Engineering Trends in Electrical and Electronics (IJETEE)*, vol. 3, pp. 75–82, May 2013.

## BIOGRAPHIES OF AUTHORS

**Shalini Vashishtha,** She completed M.Tech from MS Ramaiah Institute of Technology, Bangalore (VTU) India. Currently, she is pursuing a Ph.D. at Visvesvaraya Technological University (VTU), Belagavi, India. Her areas of interest are VLSI design and Power Electronics. She is a member of IEEE & ISTE.

**Rekha K. R.** received a Ph.D. in Electronics and Communication and is working as Professor in the department of ECE at SJB Institute of Technology, Bengaluru, India. Her research area mainly consists of network design, high-speed optical transmission, and wireless sensors. She has published papers in National and International conferences and Journals. And has published a total of 98 articles in various Journals and Conferences.