International Journal of Research in Engineering, Science and Management
Volume-2, Issue-1, January-2019
www.ijresm.com | ISSN (Online): 2581-5792

FPGA Implementation of a PWM for Multi-Level Active Clamped Inverter

M. Muthukumaran[1], R. Sreenivasan[2]
[1,2]*Assistant Professor, Department of EEE, CK College of Engineering & Technology, Cuddalore, India*

***Abstract*: with the aim to implement a suitable controller for a multilevel active-clamped inverter to enable its use in practice, and as a first step toward a full closed-loop converter Control implementation into a single field-programmable gate array (FPGA) device, this project presents the structure and features of an FPGA implementation of an appropriate pulse width modulation (PWM) strategy. The selected PWM strategy guarantees dc-link capacitor voltage balance in every switching cycle, and covers both the under modulation and over modulation regions. A flexible implementation is conceived, allowing the variation of important operating parameters, such as the modulation index and switching frequency, through a simple user interface. The key aspects to achieve an efficient and robust FPGA implementation are discussed. Experimental results in a four-level converter prototype controlled with an Altera Cyclone III device under different operating conditions match fairly well with the expected results obtained through simulation, thus verifying the accurate performance of the FPGA based modulator.**

***Keywords*: Field-programmable gate array (FPGA), multilevel active-clamped (MAC) converter, pulse width modulation (PWM).**

1. Introduction

Power electronics converter technology evolves toward multilevel topologies with higher number of switching devices, higher switching frequencies to obtain higher power density and better dynamic performance, and increased control complexity. One of such topologies is the recently proposed Multi level active-clamped (MAC) topology shown in Fig. 1 (a) for a four-level case, which is built upon a single controlled switching semiconductor device with an anti parallel diode. The functional model of the converter leg is the same as for a diode clamped converter: a single-pole four-throw switch that connects the output leg terminal to one of the four input leg terminals through the application of the corresponding switching state. Fig. 2 presents these switching states for a four-level converter leg. The uncircled switches are off-state devices. The circled switches are on-state devices. The solid-line circled switches connect the output terminal to the desired input terminal and conduct the output terminal current (i0). The dotted-line-circled switches do not conduct any significant current and simply clamp the blocking voltage of the off-state devices to the voltage across adjacent input terminals.

Compared to a multilevel diode clamped converter, which presents a lower controlled switch count, the active-clamped topology's advantages are lower conduction loss, better distribution of switching losses, device blocking voltage always equal to one dc-link capacitor voltage, and increased fault-tolerance capacity. Motor drives, in particular the traction inverter of electric vehicles, is one of the applications where the MAC topology could bring benefits. For this purpose, three legs can be connected to a common dc-link to obtain a four-level three-phase active-clamped converter shown in Fig. 1(b). However, in order to take full advantage, in practice, of the topology benefits, a robust and efficient controller has to be developed at a reasonable cost and with reduced complexity. This is the final goal of the present study. Power converter control has been progressively migrating from the analog to the digital domain. With the improvement of digital control devices and control techniques, in many applications, the inherent drawbacks of digital control (i.e., finite resolution leading to a reduced accuracy and the need of sampling and processing time leading to delays) have been compensated by the following advantages: 1) increased control complexity 2) flexibility 3) repeatability 4) reliability 5) versatility and 6) expandability.

Typical power converter digital controllers are based on microprocessors digital signal processors (DSPs), field programmable gate arrays (FPGAs), or a combination of them. µp and DSPs contain a central processing unit (CPU, fixed hardware) in charge of sequentially processing a set of instructions. They are, therefore, known to be software-programmable. Microcontrollers and digital signal controllers embed the µp or DSP core, memory, and input/output peripherals (including PWM units) in the same chip. The programming can be performed with high-level algorithmic languages familiar to most designers. FPGAs basically contain an array of logic elements or cells (each cell containing look-up tables and registers), whose interconnections are decided by the contents of a random access memory (RAM). The FPGA design, typically using hardware description languages, decides the final circuit structure to achieve a desired functionality, and it is, therefore, said to be hardware-programmable. Besides a large number of configurable input/output blocks, FPGAs also typically contain additional optimized RAM blocks and multipliers to increase the performance of the device. Logic, adders, subtractions, comparators, multipliers, and special functions are easy to implement in these typical FPGAs.

Special functions are implemented using look-up tables in RAM blocks. However, RAM size increases exponentially with

the resolution of the input variables. Multipliers and dividers by powers of two are immediate. Nevertheless, general dividers are more complex. The hardware programmability of FPGAs allows both the concurrent or parallel processing of data and pipelining, which may dramatically reduce the required processing time and increase the number of computations per switching cycle, compared to µp or DSP Additionally, in cases where the implementation of the desired control benefits from the use of a CPU, this can be embedded within the FPGA through the programming of the corresponding circuit. FPGAs have been employed in the control of a number of different topologies and applications but specially for the control of multilevel converters and the implementation of PWMs predictive controls and fault-tolerant controls due to wide device availability and familiarity with software Programming µp or DSP were the preferred designer's choice in the first digital control implementations.

As control requirements became more stringent from the viewpoint of processing time and number of PWM outputs, many designers opted for a combination of a µp or DSP with an FPGA. The sequential processor is in charge of the high-level control tasks, while the FPGA typically implements the PWM strategy and subsequent switch control signal generation. A critical reason to combine two devices is that current microcontrollers do not contain enough PWM units to control multilevel converters. This is not the case of FPGAs, which contain a large number of possible output pins that can be controlled precisely to generate the switch control signals. Due to the FPGA's inherent advantages and expected future improvements, the authors foresee a progressive trend toward a full closed-loop converter control implementation within a single FPGA. In addition to performance improvements, this also saves costs and avoids the cumbersome synchronization and communication between the µp or DSP and the FPGA. Therefore, a full FPGA-based implementation of a three-phase dc–ac MAC converter controller is concluded to be the best choice. The authors' final goal is to implement a closed-loop controller with relevant voltages and currents sampled as close as possible to the beginning of the switching cycle where these values are applied. As a first step, in this paper, the controller is implemented in open loop with the same philosophy: updating the PWM variables as close as possible to the beginning of the switching cycle. The full description of the remaining closed-loop control implementation will be presented in a complementary future publication. The project is organized as follows. Section II reviews the PWM strategy and presents a suitable algorithm from which the FPGA implementation can be discussed. Section III explains the implementation structure and relevant design details to obtain an efficient and robust controller. In Section IV, simulation and experimental results are compared to verify the good performance of the designed modulator under different operating conditions. Section V presents a preliminary discussion of the full closed-loop control implementation. Finally, Section VI presents the conclusions.

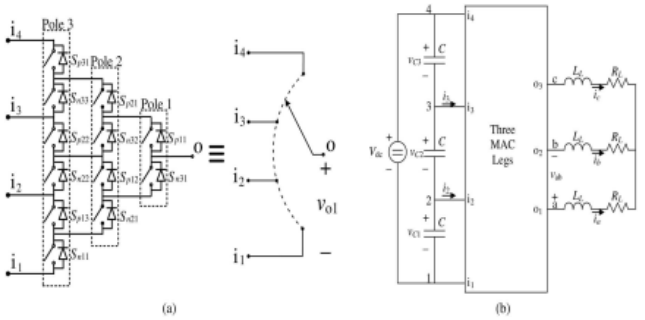

Fig. 1. MAC converter (a) Four-level leg topology. (b) Four-level three-phase dc–ac converter.

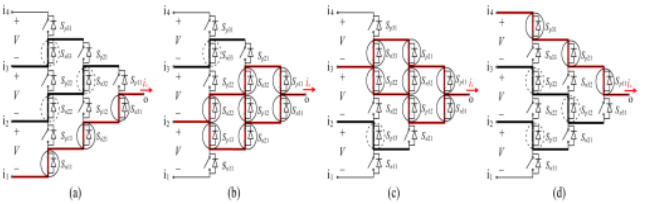

Fig. 2. Four-level MAC-leg switching states. (a) Connection to node i_1, (b) Connection to node i_2, (c) Connection to node i_3, (d) Connection to node i_4.

2. PWM strategy

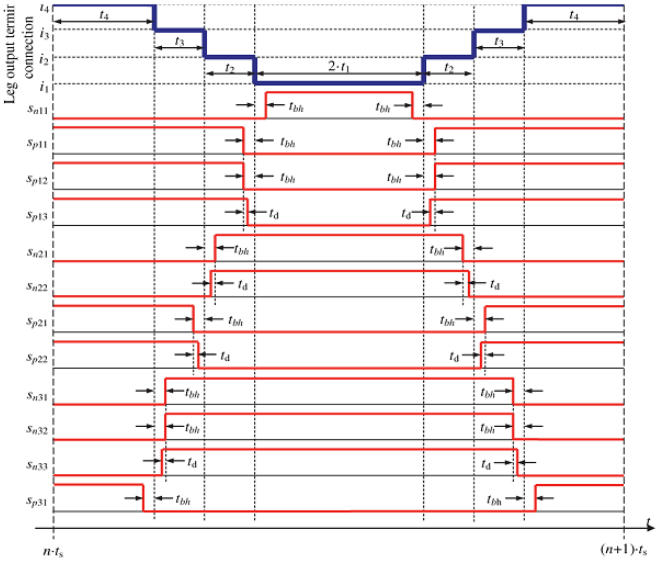

Fig. 3. Distribution of the leg output terminal connections over a switching cycle and corresponding switch control signals (high level: switch ON; low level: switch OFF).

The PWM presented in and extended to the over modulation region in is selected to operate the converter in Fig. 1 (b). This PWM, originally defined applying the virtual-vector concept, guarantees the dc-link capacitor voltage balance in every switching cycle and all operating conditions, as long as the phase currents are approximately constant over the switching cycle and their addition is equal to zero. This allows minimizing the size of the dc-link capacitors, leading to a high converter power density. The PWM assumes that the switching frequency ($f_s=1/t_s$) where t_s is the switching or sampling period) is much larger than the fundamental frequency. This PWM allows

modulation index values m € [0,1.1027], where $m=V_{ab,1,pk}/V_{dc}$ and $V_{ab,1,pk}$ is the peak value of the fundamental component of the line-to-line converter voltage. Therefore, it covers both the under modulation (UM) and over modulation (OM) ranges up to six-step operation. The over modulation region is further divided into two sub regions (OMI and OMII) with different types of applied reference vector trajectories. This PWM can be defined with the following algorithm. The desired instantaneous values of the three converter ac-side voltages define a desired rotating reference vector in the α-β plane of the converter space vector diagram (SVD). The desired reference vector-normalized length and angle (with reference to axis) are designated as m* and θ* respectively. In the UM region, this desired reference vector can be directly synthesized with the available converter space vectors. However, in the OM region, this vector cannot be always synthesized because it sometimes falls outside the SVD outer hexagon. Therefore, an alternative viable reference vector trajectory inside the SVD outer hexagon is defined so as to produce the same fundamental three-phase output voltages, despite the introduction of low order harmonics. Let us designate the length and angle of the applied reference vector as m and θ Thus, the first step of the algorithm consists on determining the value of m and θ from m* and θ* Due to the six-fold symmetry of the SVD, this analysis can be performed in the first sextant of the SVD considering the first sextant- equivalent reference vectors with angles θ_1* and θ_1 The values of m and θ_1 are determined with (1), (2), and (3). These equations have been extracted and rewritten here to facilitate the PWM implementation. The number assigned to each sextant of the SVD is different for each leg ($sext_a$, $sext_b$ and $sext_c$) to explicitly state the three-phase symmetry of the SVD, which will help in simplifying the PWM implementation. Variable mode contains two bits to identify the operating mode mode=01, mode=00: OMI, mode=10:OMII Variable hbc≤1 is a special parameter that regulates the capacitor voltage balance control margin in the over modulation region.

$m^*_{maxi}=3.In(3)/\pi$

$m^*_{maxi}=(2/\pi).sqrt(3)$ (1)

$sext_a=1+floor(\theta^*/(\pi/3)$

$sext_b=mod(sext_a+4,6)$

$sext_c=mod(sext_a+2,6)$

$\theta^*1=mod(\theta^*, \pi/3)$ (2)

if($m^*.hbc.m^*_{maxII}$){

$m^*=hbc.m^*_{maxi}$

}end

Mode=01

$m=m^*$

$\theta_1=\theta^*$

if($hbc<m^*\leq hbc.m^*_{maxI}$){

mode=00;

$\theta_{lim}=(\pi/6).(1-(m^*-hbc)/(hbc.m^*_{maxi}-hbc))$

$k_{m1}=hbc/sin(\theta_{lim}+\pi/3)$

$k_{m2}=1/sin(\theta^*_1+\pi/3)$

if($\theta_{lim<}\ \theta^*_1\leq \pi/3-\theta_{lim}$){

$m=hbc.k_{m2}$

}else{

$M=k_{m1}$

}end

}elseif($hbc.m^*_{maxi}<m^*\leq hbc.m^*_{maxi}$){

Mode=10

$\theta_{lim=}\ \pi/6.(m^*-hbc.m^*_{maxi})/(hbc.m^*_{maxi}-hbc.m^*_{maxi})$

km1=hbc.2/sqrt(3)

$km2=1/sin(\theta^*_1+\pi/3)$

if($\theta^*_{1<}\ \theta_{lim}$){

m=km1

$\theta_1=0$

}elseif($\theta^*_{1\geq}\ \pi/3-\theta_{lim}$){

M=km1

$\theta_{1=}\ \pi/3$

}else{

M=hbc.km2

}end

}end (3)

Next, the value of auxiliary variables t_α, t_β, t_γ and tδ are computed as

$t_\alpha=m.kt\alpha.t_s/2$, $kt\alpha=cos(\theta_{1+}\ \pi/6)$

$t_\beta=m.k_{t\beta}.t_s/2$, $k_{t\beta=}cos(\theta_{1-}\ \pi/6)$

$t_{\gamma=}\ t_{\beta-}\ t_\alpha$

$t\delta=(1-t_{\beta)/2}$ (4)

These auxiliary variables will be used to define the final dwell times of connection to the dc-link terminals t_α, t_β, t_γ and t_δ with reference to Fig. 3. But previously, they need to be modified, so that final dwell times fall into a specified value range, as will be commented later

$$[\ t_{\alpha m}, t_{\beta m}, t_{\gamma m}, t_{\delta m1,}\ t_{\delta m2]} = f\ ([t_\alpha, t_\beta, t_\gamma, t_{\delta])} \quad (5)$$

Fig. 4 illustrates the resulting leg duty-ratio pattern ($d_i=2.t_i/t_s$) over a line cycle for four different values of the other two legs have the same duty-ratio pattern but phase Shifted ±120º The distribution of the leg output terminal connections over a switching cycle and the generation of 12 switch control signals is performed according to Fig. 3, where $t_{bh}=t_h/2$ is half of the blanking time required between the turn-OFF of a set of devices and the turn-ON of others to produce a change of the leg switching state, and t_d is a delay time used whenever several switch control signals change concurrently, so that switching losses of a transition between switching states concentrate on one specific switch.

3. PWM implementation

The previous PWM strategy is implemented into an Altera Cyclone III EP3C16F484C6 FPGA device with a 50-MHz oscillator, mounted on a DE0 development and education board from Terasic. The device architecture structure and functionality have been fully described in very-high-speed integrated-circuit hardware description language (VHDL). The

simulation has been performed with Model Sim software from Mentor Graphics. The synthesis has been performed with Quartus II software from Altera. Simulation results have been compared with the results obtained from MATLAB-Simulink simulations. The main PWM parameters are set to be user configurable through 10 slide switches (SdS) to introduce the parameter code and value, and one pushbutton switch (PBS) to validate the data, while the converter is in the off state and their value range and resolution, from which a 158 Hz–625 kHz range of possible switching frequencies can be easily derived. This allows thoroughly testing the implemented modulator performance under a wide range of operating conditions and power hardware prototypes. Two additional PBS are used to turn $_{on/off}$ the converter and to reset the system.

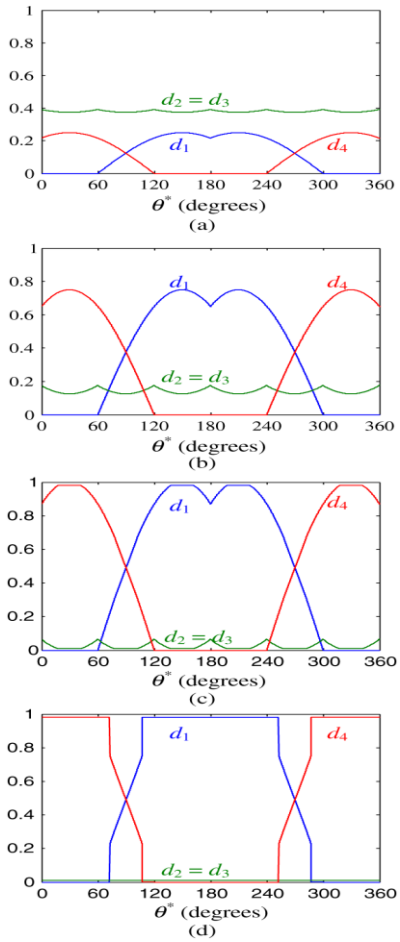

Fig. 4. Leg "a" duty-ratio pattern over a line cycle (hbc=0.98)
(a) m*=0.25 (b) m*=0.75 (c) m*1.0 (d) m*=1.05

A. Basic design aspects

Variable values are represented in fixed-point format to reduce hardware needs. The advantages of a floating point representation, such as wider variable value ranges, are not of special interest in the considered application, and it would lead to a more complex hardware implementation. The number of bits of every variable is selected to guarantee the desired accuracy and to optimize the use of FPGA resources. Identification of symmetries, value offset, and value scaling are applied whenever possible to minimize the number of bits required to obtain a specified resolution. All operations in the algorithm of Section II are easy to implement, except for divisions and transcendental functions. Expression (1) contains simple constants that can be pre calculated. Expression (2) highlights the 60-degree symmetry of the algorithm. The divisions and trigonometric functions in (3) require a detailed analysis to obtain the simplest implementation. Fig. 5 deploys the pattern for k_{m2}, $k_{t\alpha}$, and $k_{t\beta}$ from which 30-degree symmetries, proper offset values, and proper scaling values can be identified.

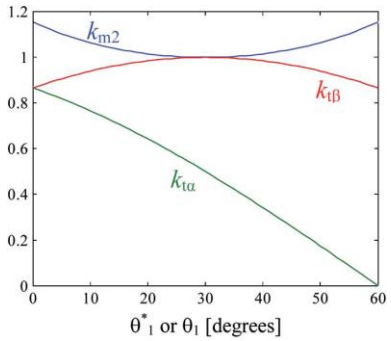

Fig. 5. Parameter k_{m2}, $k_{t\alpha}$, and $k_{t\beta}$ as a function of θ_1* or θ_1

B. Controller structure and module description

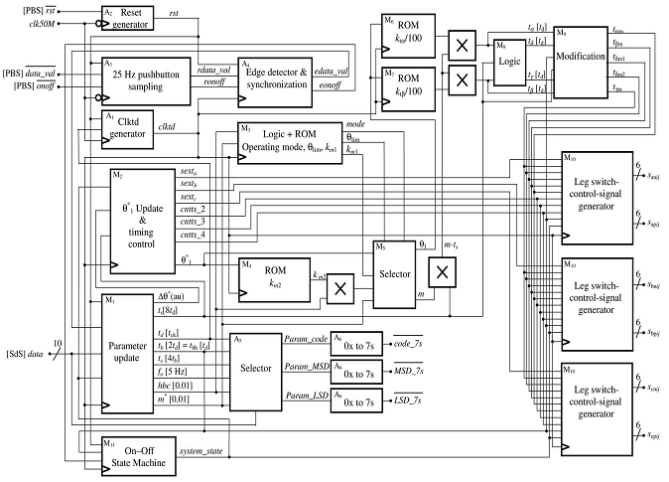

Fig. 6. FPGA-controller global structure

The designed controller structure, applying concurrent processing as much as possible, is depicted in Fig. 6. It is composed of a set of blocks or modules, classified as auxiliary A_1-A_6 and main M_1-M_{11} modules. The applied variable value offset and scaling to optimize the implementation is not explicitly shown in the diagram. From the 50-MHz system clock signal clk 50M, having a period t_{ck}=20 ns module A_1 generates the clock signal clktd, with period, used in most of the relevant processes. Module A_2 applies simple logic to generate a synchronous system reset signal rst from the state of the corresponding PBS. Module A_3 samples at low frequency the value of the signals generated by the other two PBS, and A_4 generates a pulse of length t_d in the output signal whenever a rising edge occurs in the corresponding input signal. Finally,

modules A_5 and A_6 are in charge of generating the required signals to show in three seven-segment displays the code and stored value [more significant hexadecimal digit (MSD) and less significant hexadecimal digit (LSD)] of a given system parameter, determined by the position of the SdS. Module M_1 updates the registered value of a given system parameter, defined by the 10 SdS, whenever the data validation PBS is pressed. The values are expressed and stored in the units indicated within brackets in the block output signals. Parameter limits and coherence are checked.

C. Consumed FPGA resources

A special effort has been made in the design to save FPGA resources. Table III indicates the resources used to synthesize the previous controller, as reported by Quartus II software. These resources represent roughly around 20% of the total FPGA resources, leaving enough space to implement a full closed-loop converter controller, including the closed-loop control of the dc link capacitor voltages and three-phase currents. A reduced number of resources will be necessary if a fixed value of Table II parameters is assumed, which is typically the case in practical applications.

4. Simulation and experimental results

The previous FPGA-based control has been tested with a three-phase dc–ac converter configuration feeding a three-phase series R–L load, as shown in Fig. 1(b). The converter prototype with the DE0 board on top, containing the Altera Cyclone III EP3C16F484C6 FPGA, is presented in Fig. 7.

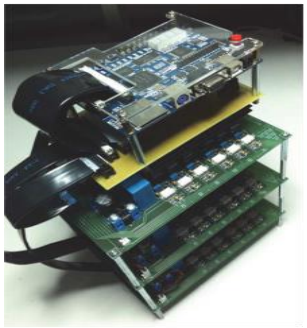

Fig. 7. Experimental prototype combining DE0 development and educational

Board with a four-level three-phase dc–ac converter prototype using 100 V. Breakdown voltage metal-oxide-semiconductor field-effect transistors.

Fig. 8. presents the experimental switch control signals and output voltage of one leg, which follows the pattern described in Fig. 3. Note that the analog V01 voltage waveform is delayed with reference to the digital switch control signals due to gate driver and measurement delays. Since the converter is no ideal and it is operating in open-loop, there is some unbalance among the Capacitor voltages that could be easily corrected with the introduction of a closed-loop control. The operating principle of such closed-loop control can be summarized as follows. From the sensed dc-link capacitor voltages V_{c1}, V_{c2}, and V_{c3} the value of the capacitor voltage unbalance is determined. This unbalance is then corrected through a slight modification of the open-loop leg duty-ratio pattern as shown in Fig. 4, which produces a switching- cycle average current through the inner dc-link points [i_2 and i_3 in Fig. 1(b)] different from zero, as needed to correct the unbalance. Figs.9 and 10 present simulation and experimental results at different modulation indexes, covering the UM and OM ranges. MATLAB-Simulink is used to perform the simulations. The obtained experimental results fairly match those from simulation, verifying the accurate performance of the programmed controller.

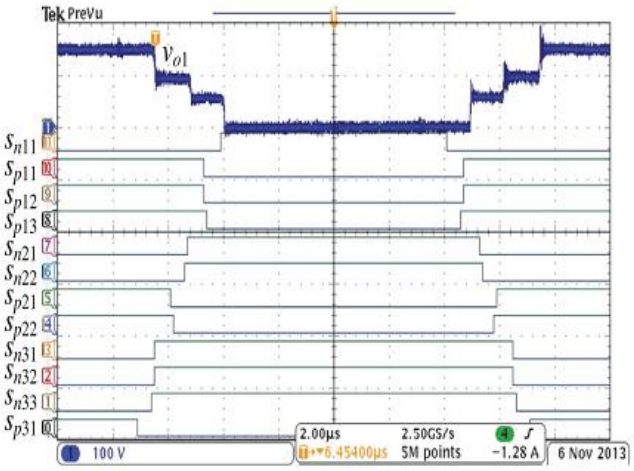

Fig. 8. Experimental leg output terminal voltage V01 and switch control signals

5. Conclusion

A full FPGA-based controller implementation is concluded to be the best choice to enable the practical use of MAC converters and take full advantage of their potential benefits at a reasonable cost. Conventional PWM strategies lead to the collapse of some dc-link capacitor voltages under a wide range of operating conditions for the case of converters with a number of levels higher than three. A singular and complete PWM strategy capable of guaranteeing the dc-link capacitor voltage balance in every switching cycle for all operating conditions has been selected to operate a four-level three-phase dc–ac active-clamped converter featuring 36 switching devices with independent gate control signals. This PWM strategy has been efficiently and robustly implemented into a medium performance FPGA, consuming only a limited amount of resources. The remaining FPGA resources can be employed to implement typical closed-loop controls for different applications with a high dynamic performance, and dynamically reconfiguring the control in runtime, which can be useful, for instance, in implementing fault-tolerant controls.

References

[1] E. F. F. Lima, N. P. Filho, and J. O. P. Pinto, “FPGA implementation of space vector PWM algorithm for multilevel inverters using non-orthogonal moving reference frame,” in Proc. *IEEE Int. Elect. Mach. Drives Conf.*, 2009, pp. 709–716.

[2] J. Han, T. Tang, B. Pu, Y. Xu, C. Liu, X. Wei et al., "FPGA based hybrid cascade multilevel converter," in Proc. *IEEE Int. Symp. Ind. Electron*., 2009, pp. 494–498.
[3] P. Karuppanan and K. K. Mahapatra, "FPGA based cascaded multilevel pulse width modulation for single phase inverter," in Proc. *Int. Conf. Environ. Elect. Eng*., 2010, pp. 273–276.
[4] W. A. Halim and N. A. Rahim, "FPGA-based pulse-width modulation control for single-phase multilevel inverter," in Proc. *IEEE Conf. Clean Energy Technol*., 2011, pp. 57–62.
[5] R. N. Alencar Leao e Silva, L. H. S. C. Barreto, A. U. Barbosa, D. R. Joca, and P. P. Praca, "Digital implementation of a modulation technique for a multilevel inverter on FPGA," in Proc. *IEEE Int. Conf. Ind. Appl*., 2012, pp. 1–6.
[6] M. P. Aguirre, L. Calvino, and M. I. Valla, "Multilevel current-source inverter with FPGA control," *IEEE Trans. Ind. Electron*., vol. 60, no. 1, pp. 3–10, Jan. 2013.
[7] A. de Castro and E. Todorovich, "High resolution FPGA DPWM based on variable clock phase shifting," *IEEE Trans. Power Electron*., vol. 25, no. 5, pp. 1115–1119, May 2010.