International Journal of Power Electronics and Drive System (IJPEDS)
Vol. 7, No. 1, March 2016, pp. 94~106
ISSN: 2088-8694

VHDL Implementation of Capacitor Voltage Balancing Control with Level-Shifted PWM for Modular Multilevel Converter

Chuen Ling Toh, Lars Einar Norum
Department of Electrical Power Engineering, Norwegian University of Science and Technology (NTNU), Norway

Article Info

Article history:

Received Sep 29, 2015
Revised Jan 6, 2016
Accepted Jan 22, 2016

Keyword:

Capacitor voltage balancing control
Level-shifted PWM
Modular multilevel converter
Vhdl
Xilinx 7-series FPGA

ABSTRACT

Power electronics converters are a key component in high voltage direct current (HVDC) power transmission. The modular multilevel converter (MMC) is one of the latest topologies to be proposed for this application. An MMC generates multilevel output voltage waveforms which reduces the harmonics contents significantly. This paper presents a VHDL implementation of the capacitor voltage balancing control and level-shifted pulse width modulation (LSPWM) for MMC. The objective is to minimize the processing time with minimum gate counts. The design details are fully described and validated experimentally. An experiment is conducted on a small scale MMC prototype with two units of power cells on each arm. The test results are enclosed and discussed.

Corresponding Author:

Chuen Ling Toh,
Department of Electrical Power Engineering,
Norwegian University of Science and Technology (NTNU),
Trondheim, Norway
Email: chuenling@uniten.edu.my, tohchuenling@yahoo.com

1. INTRODUCTION

High voltage direct current (HVDC) has been recognized as a cost effective solution for long distance power transmission [1], [2].

Figure 1 (a) shows a single line circuit diagram of an HVDC system [3]. The terminals of an HVDC transmission system are formed by power electronics converters. The ac power is first converted into dc power at the sending end using an ac-dc power converter. The dc power is then converted back into ac power at the receiving end with the control of a dc-ac power converter. The most recent power converter topologies used in today's HVDC systems are modular multilevel converters (MMCs) [4], [5], [6].

MMCs are constructed using a series of identical power electronics building blocks (PEBBs). Its basic structure and the functional block diagram of a PEBB are shown in

Figure 1 (b) and (c). PEBB is a fully integrated device formed by different groups of hardware components, including power cell, gate driver, heat sink and sensors [7], [8]. Each PEBB acts as a controllable voltage source. Commutation of these PEBBs enables the summation of the capacitor voltages to reach high voltage levels at the output terminal. MMCs produce low harmonic content output voltages by increasing the number of PEBBs [9], [10]. This minimizes the ac harmonics filter size, which consequently reduces the footprint of HVDC stations. An MMC with more than two hundreds units of PEBB per arm has been commercialized [11].

In order to control a large number of PEBBs, a high speed digital controller is required to perform the complex control and modulation algorithms necessary, examples of which include a Digital Signal Processor (DSP), FPGA, or System on Chip (SoC) [12]-[15]. DSPs are typically used in power electronics

converter control; however they have a limited number of PWM output pins which is insufficient to control an MMC with more than hundreds units of PEBBs per phase. Therefore, an FPGA has been coupled with DSP boards, mainly to distribute these control signals [16]. In [17] – [19], an FPGA is also used to shorten the processing time by executing simple logic functions such as performing carrier-based PWM.

This paper presents VHDL implementation of MMC control. Capacitor voltage balancing control and level-shifted pulse width modulation (LSPWM) are chosen for implementation, mainly due to the simplicity of the calculations involved with these schemes. The objective is to minimize the processing time when the MMC employs a large number of PEBBs. The proposed design is capable to sort out a hundred units of PEBB capacitor voltage measurements in less than ten microsecond's time. In addition, the design is optimized to save some gate counts, for example only a single carrier wave is implemented for LSPWM. By reducing the use of resources and processing time, some other control algorithms can be configured together in the same FPGA. The proposed design is modular. It can be duplicated to configure a three phase system by introducing a balance three-phase modulating waves. An overview of the MMC will be given in Section 2. Theory of the control and modulation technique to be implemented in this paper will be briefly explained. Section 3 will discuss the optimum design details. The controller is then validated through a small scale MMC in Section 4. Xilinx 7-series FPGA will be used for implementing the proposed controller. All experimental results will be presented with full discussion.

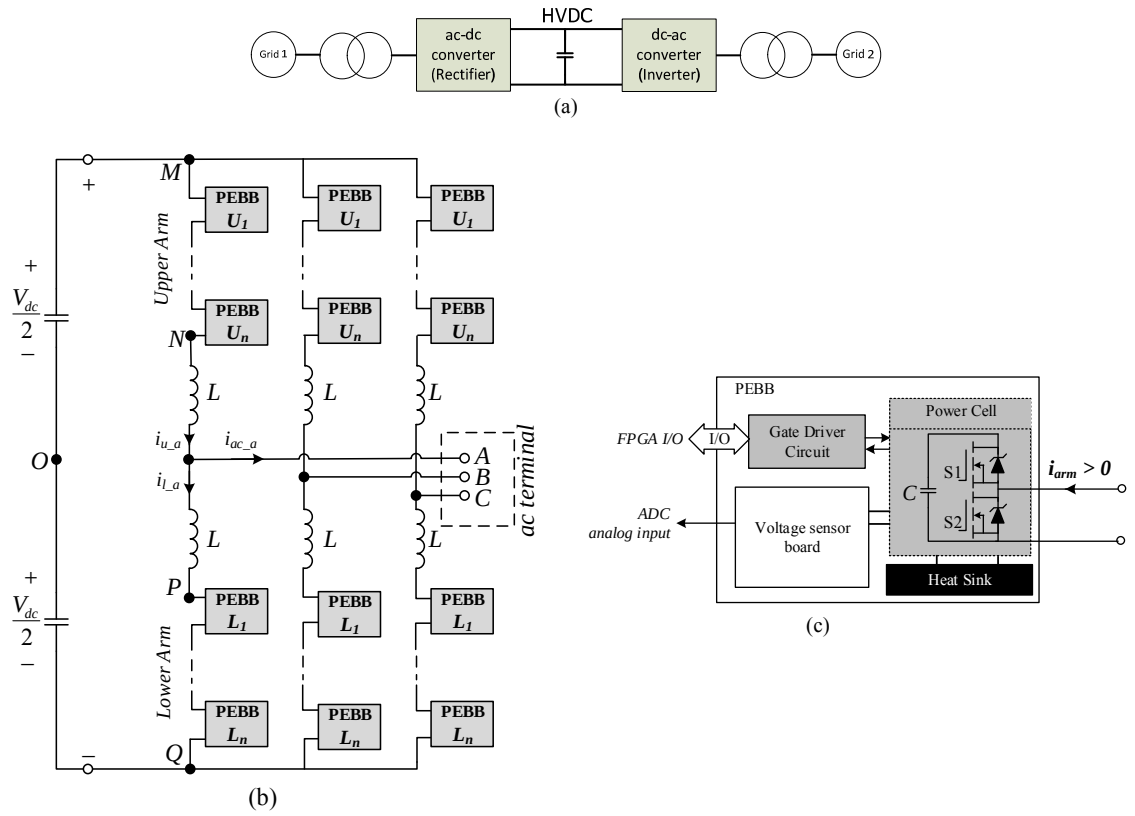


Figure 1. (a) Single line circuit diagram of high voltage direct current (HVDC) System (b) MMC with *n* units of power electronics building blocks (PEBBs) per arm; (c) PEBB functional block diagram

2. MODULAR MULTILEVEL CONVERTER (MMC)

The basic structure and principal of MMC is fully described in [10], [11], and [20]. The general control and modulation schemes used in MMC are summarized in [21]. This section will briefly present the concept of MMCs with the selected control and modulation schemes.

2.1. Principle

An MMC always contains an equivalent number of PEBBs on upper and lower arms for each phase. Each PEBB consists of an energy storage element, i.e. the capacitor which is connected in parallel to the

power semiconductors (*S1* and *S2*). In general, an MMC contains *n* units of PEBB per arm. Each capacitor voltage must be maintained as follows:

$$V_C = \frac{V_{dc}}{n} \tag{1}$$

When *S1* is triggered on, the PEBB will contribute its full capacitor voltage, V_c, to the MMC. Inversely, when *S2* is switched on, this PEBB is bypassed, contributing approximately 0V to the power circuit. Therefore, the energy transfer between dc and ac terminals is easily controlled by inserting or bypassing the PEBBs.

Equation (2) formulates the phase-a output voltage by neglecting the voltage drop across the arm inductors referring to MMC circuit in

Figure 1 (b).

$$v_{AO} = v_{NM}(t) + \frac{V_{dc}}{2} = v_{PQ}(t) - \frac{V_{dc}}{2} \tag{2}$$

Upper arm current i_{u_a} is defined as positive when it flows from the positive dc terminal towards the ac terminal. A positive lower arm current i_{l_a} will flow from the ac terminal to the negative dc terminal; hence, the ac terminal phase current, i_{ac_a} is derived as

$$i_{ac\_a} = i_{u\_a}(t) - i_{l\_a}(t) \tag{3}$$

Equation (2) and (3) are also applicable to other phases. The mean value of i_{u_a} and i_{l_a} form the circulating current, which is limited by the arm inductors. A full derivation of the circulating current and its harmonics expression are given in [22].

2.2. Capacitor Voltage Balancing Control

The main control objective in this paper is to regulate balance capacitor voltages inside each arm [23]. The capacitor of each PEBB will be ordered for charging or discharging based on its capacitor voltage level and arm current direction. A simple voltage-balancing algorithm, proposed in [24], will be implemented in this paper. This method measures the arm current and PEBB's capacitor voltages periodically. The PEBBs on each arm will be sorted based on its V_c value in ascending order. When the sorting result turns valid, the arm current direction will be used to distribute the duty cycle determined from the modulation scheme. For instance, if the sampled arm current is found to be positive, the PEBB with lowest V_c level must be assigned the longest duty cycle to charge up that particular capacitor. Inversely, PEBBs which rank at the bottom of the sorting list should execute the shortest duty cycle. This control method is simple but the processing time for sorting algorithm increases drastically as hundreds of PEBBs are employed in each arm. This paper proposes a logic design which will only occupy *n* clock cycles to sort out *n* number of PEBBs in each arm. By driving the sorting algorithm logics using a high frequency clock (100MHz), the required time to sort out a hundred units of PEBB is less than 10 μs.

2.3. Level-Shifted Pulse Width Modulation (LSPWM)

LSPWM is easily implemented using FPGA. Each phase of the MMC requires a pair of 180° out of phase modulating waves. These signals are named as the upper modulating wave, v_{mod_up}, and lower modulating wave, v_{mod_low}. Each of these modulating waves will be compared with *n* units of triangular carrier waves to determine *n* number of duty cycles for PEBBs on each arm. In this paper, the carrier waves are set in phase disposition. They are identical and synchronous in phase [25]. The amplitude ratio of each carrier wave is set as follows:

$$v_{tri} = \frac{1}{n} \tag{4}$$

Figure 2 shows the above mentioned control and modulation example to an MMC using four units of PEBBs in one phase. Assume that the dc link voltage is set at 100V. The data sampling at time instant *A* are depicted in Figure 2 (a), which include arm currents direction and capacitor voltage measurements. Upper arm's PEBBs are then sorted in sequence as U1 followed by U2, whereas, lower arm's PEBBs are listed as L2 followed L1. Figure 2 (b) presents the concept of LSPWM. Both upper and lower modulating signals must be kept within the two carrier waves to ensure the MMC works in linear modulation mode. The gating

signal for switch S1 of a PEBB is obtained by comparing the modulating signals with a triangular wave. Switch S2 always trigger complementally to switch S1. For instance between time instant t_1 and t_2, a unit of upper arm's PEBB must be fully turned on with another PEBB carries out PWM modulation. Conversely, a PEBB at the lower arm will also carry out PWM modulation with another PEBB is permanently switched off. In summary, there are three switching mode which will be applied to a PEBB, i.e. fully turn-on, PWM or permanently turn-off.

As positive arm current is sampled on the upper arm at time instant *A*, PEBB U1 must fully turned on for changing up its capacitor. PWM switching mode will be carried out by PEBB U2. On the other hand, PEBB L2 which listed at the top of lower arm sorting list will be permanently turned off to maintain its capacitor voltage level when a negative arm current is sampled at time instant *A*. PWM switching will be assigned to PEBB L1.

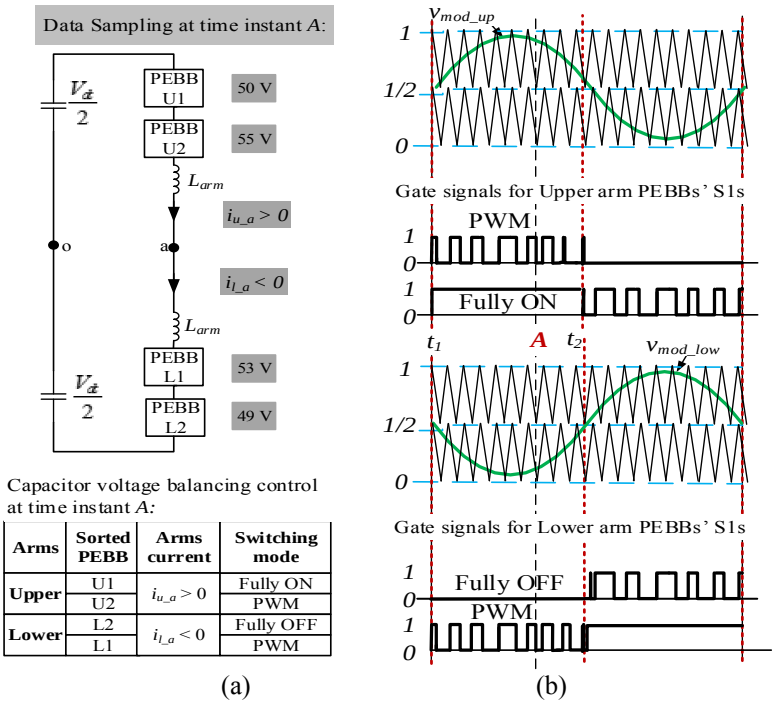


Figure 2. An example of capacitor voltage balancing control and LSPWM for one phase of MMC which employs 4 PEBBs

3. MMC CONTROLLER DESIGN

The MMC controller is fully implemented in programmable logic using very high-speed integrated circuits, hardware description language (VHDL). This section will discuss the controller design for one phase of the MMC. This design (top-level wrapper) can then be easily duplicated for three-phase MMC implementation. The MMC switching frequency is defined as 2 kHz with the input data sampling rate set as 4 kHz.

A functional block diagram of the MMC controller is presented in Figure 3. This controller is divided into five modules. The Clocks module is essential to produce appropriate clock frequencies and generate control signals for different functional blocks. The sine wave generator module is used to generate a 50Hz sine wave for the Modulation Module. The Modulation Module will perform LSPWM, whilst the capacitor voltage balancing control is carried out by the Control Module. The results produced from these two blocks will be sent to the Switching Commands Distribution Module. A series of multiplexers are used to distribute the specified duty cycles to the corresponding PEBBs. Each logic block will be further described in the following sub-sections.

3.1. Clocks Module

The MMC controller will be tested using Xilinx ZC702 evaluation board in this paper. The system clock source on the evaluation board is provided by a 200MHz oscillator [26]. In order to provide different clock frequencies to other logic modules, Xilinx LogicCORE IP clocking wizard is implemented as a frequency synthesizer [27], [28]. Two output clock frequencies are generated, i.e. 150 MHz and 2.5 MHz. Since they are synthesized from a same root clock using a unit of clocking wizard, they are recognized as synchronous clocks. The clocking wizard ensures the clock phase relationship is maintained throughout the chip. These clocks are phase synchronous and skew matched, i.e. after 60 rising edge of fast clock there will always be overlap of rising edge for both fast and slow clocks. The 150 MHz clock is used to drive most of the logic blocks in the controller, such as to produce a 2 kHz triangular carrier wave, and to drive the capacitor voltage sorting algorithm. Alternatively, the 2.5 MHz clock is used as the base clock to produce a 50 Hz modulating wave.

Some control pulses will be generated periodically to enable synchronous input data sampling and initializing the sorting algorithm in this module. All of these signals are generated in 150 MHz clock domain.

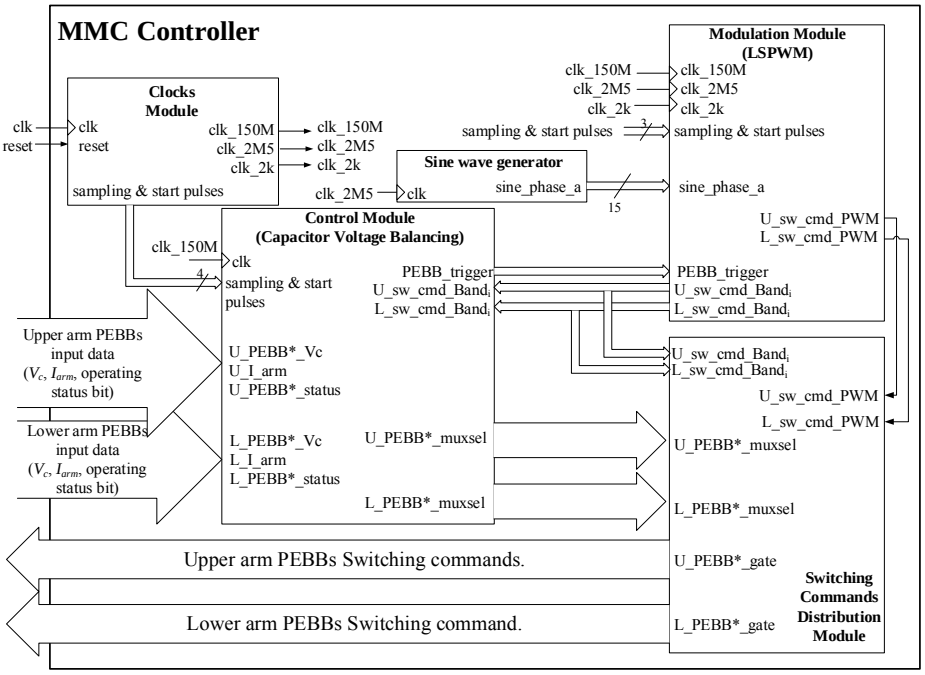


Figure 3. Functional block diagram of the MMC control (top level)

3.2. Sine Wave Generator

A sine wave signal can be produced using different methods, such as direct look-up table, linear interpolation, and recursive methods [29]. Direct look-up table is the most commonly used technique. It is simple compared to other methods which involve complex mathematical function computation. The accuracy of the direct look-up table method is proportional to the table length. Increasing the table length will produce high resolution sine wave which at the same time reduces quantization error. It has been estimated in [30] that, a 512-entries look-up-table (the minimum length) would avoid sub-harmonic distortion in the generated sine wave. Using a good quality sine wave as a reference signal to control a power electronics converter may reduce total harmonics distortion in output voltage and current waveforms.

In this paper, we aim to generate high resolution sine wave with minimum design effort. Thus, Xilinx LogiCORE IP DDS (direct digital synthesizer) [31] is implemented. The IP Core will generate a sine wave based on the input phase angle. A 16-bit counter is used to generate the phase angle. The required phase increment, $\Delta\theta$, can be calculated as follows:

$$\Delta\theta = \frac{f_{out} \times 2^{16}}{f_{clk}} \quad (5)$$

By setting the input clock frequency as 2.5 MHz (f_{clk}), the $\Delta\theta$ value should be incremented with a value of 1.31 every clock cycle to produce a 50 Hz (f_{out}) sine wave.

An analysis has been carried out in order to determine an integer of $\Delta\theta$ which is much easier to implement in programmable logic. Figure 4 (a) and (b) illustrates the analysis results for the first few clock cycles. Figure 4 (a) shows the theoretical result of the generated phase angle using 1.31 as the value for $\Delta\theta$. Figure 4 (b) illustrates that, by incrementing $\Delta\theta$ between an integer value of 1 and 2, the phase angle increases proportionally with the theoretical value. In order to fine tune the controller, the incrementing of $\Delta\theta$ must be reset to '1' every 28th clock. A finite state machine (FSM) is designed to satisfy the above analysis. Figure 4 (c) illustrates the state transition diagram of an FSM. During initialization, the phase angle is reset to zero so that the FSM will enter state S0 and set the FSM counter (*FSMc*) to zero. The FSM counter is used to keep watching for the 28th clock event. When this event occurs, the FSM will transition from state S2 to state S0, reset the *FSMc*, and proceed to state S1; else the FSM will rotate in the sequence of S1, S2, and S3 repeatedly. State S1 and S2 will output a $\Delta\theta$ value of 1 while state S3 gives a $\Delta\theta$ value of 2. In order to accurately produce a 50Hz sine wave, the phase angle must be reset to zero when it reaches $\theta = \text{FFEC}_H$ the value of which is obtained through a simulation. The generated sine wave is presented in bipolar 15-bit data.

3.3. Modulation Module (LSPWM)

This module mainly executes a few tasks in parallel, i.e. produces a pair of modulating waves, generates a triangular carrier wave and determines the switching commands for PEBBs.

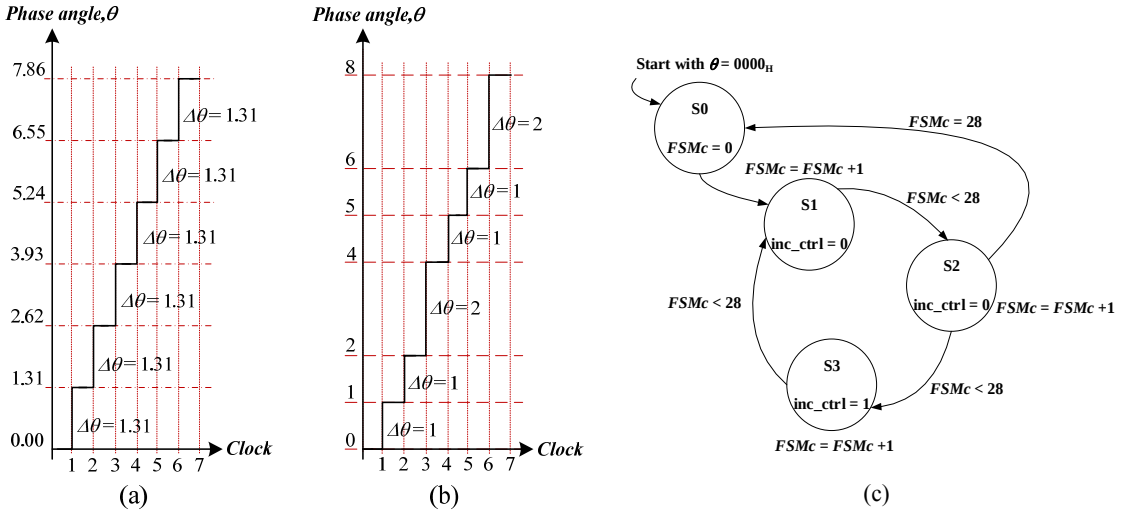


Figure 4. (a) Phse angle increment with $\Delta\theta$ set as 1.31 (theoretical value); (b) proposed phase increment with integer value. (c) Phase increment control Finite State Machine.

The receiving sine wave, v_{sin}, is converted into an upper modulating wave, v_{mod_up}, and a lower modulating wave, v_{mod_low}, as follows:

$$v_{\text{mod\_}up} = v_{\sin} + 4000_H \tag{6}$$

$$v_{\text{mod\_}low} = \textbf{NOT}\,(v_{\sin}) + 4000_H \tag{7}$$

The converted signals are in unipolar mode. They are then shifted according to the total number of PEBBs employed in each arm. Thus, the level-shifting magnitude, A_{LS}, can be determined by dividing the amplitude of the modulating wave, A_m, with n number of PEBBs:

$$A_{LS} = \frac{A_m}{n} \tag{8}$$

Then a set of level shifting bands can be calculated

$$Band_i = A_{LS} \times i \tag{9}$$

where i = 1, 2, … , (n–1). By comparing the modulating wave with the level shifting bands, (n–1) number of switching commands will first be determined as follows:

$$\begin{aligned} v_{\text{mod}} > Band_i, \quad sw\_cmd\_Band_i = 1 \\ v_{\text{mod}} < Band_i, \quad sw\_cmd\_Band_i = 0 \end{aligned} \tag{10}$$

where i = 1, 2, … , (n–1).

Figure 5 (a) shows the upper and lower modulating waves (upper and lower arms) for an MMC which consists of two PEBBs per arm (n = 2). The unipolar sine wave is generated in 15-bit full scale (A_m = $7FFF_H$). Hence, level-shifting magnitude, A_{LS}, is equivalent to $3FFF_H$. Consequently, $Band_1$ is set as $3FFF_H$. Figure 5 (b) illustrates the shifted modulating waves which will be used for PWM. Figure 5 (c) shows the gating signal obtained from (10). To increase the resolution, the shifted modulating waves will be amplified back to full scale (A_m = $7FFF_H$) before being compared with the carrier wave as shown in Figure 6 (a).

Figure 6 (b) shows the multi clocks crossing region in this module. Since the sine wave is generated in 2.5 MHz clock domain, it will first be sampled using 150 MHz clock before the data is being compared. As these two clocks are synchronized, there would be no data loss when crossing to 150 MHz clock for processing because the sine wave is generated once in every 60 cycles of the fast clock [32].

The triangular carrier wave is generated using a 16-bit counter in 150 MHz clock domain. This counter will count up and down repeatedly after system initialization. Figure 6 (a) illustrates the triangular carrier design in this paper. A 2 kHz clock signal, clk_{sw} with 50% duty ratio will be used to control the counter operation. This signal requires double flop synchronizer for clock crossing into 150 MHz clock domain as shown in Figure 6 (b). The 16-bit counter will start counting up at the falling edge of clk_{sw}. Inversely, the rising edge of clk_{sw} will enable the countdown process. The instantaneous value of the counter forms the desired triangular carrier. In this paper, the amplitude of the carrier wave, $A_{tri\_carrier}$, is estimated as $927C_H$.

$$A_{tri\_carrier} = \frac{f_{counter}}{f_{sw} \times 2} \tag{11}$$

In order to change the switching frequency of MMC, a new clock signal which represents the carrier frequency (clk_{sw}) must first be determined. Then the new $f_{counter}$ can be selected by tuning the desired $A_{tri\_carrier}$. The $A_{tri\_carrier}$ must be set in the following range to ensure the MMC operates in linear modulation mode.

$$A_m < A_{tri\_carrier} < FFFF_H \tag{12}$$

where A_m is defined as the modulating wave amplitude.

The 15-bit full scale level shifted modulating wave will be adjusted so it is positioned in the middle of the triangular wave. The offset can be calculated as follows:

$$offset_m = \frac{A_{tri\_carrier} - A_m}{2} \tag{13}$$

Thus, the n^{th} switching command will be obtained by comparing the shifted modulating wave with the triangular carrier using the following equations:

$$\begin{aligned} v_{\text{mod}} > v_{tri}, \quad sw\_cmd\_PWM = 1 \\ v_{\text{mod}} < v_{tri}, \quad sw\_cmd\_PWM = 0 \end{aligned} \tag{14}$$

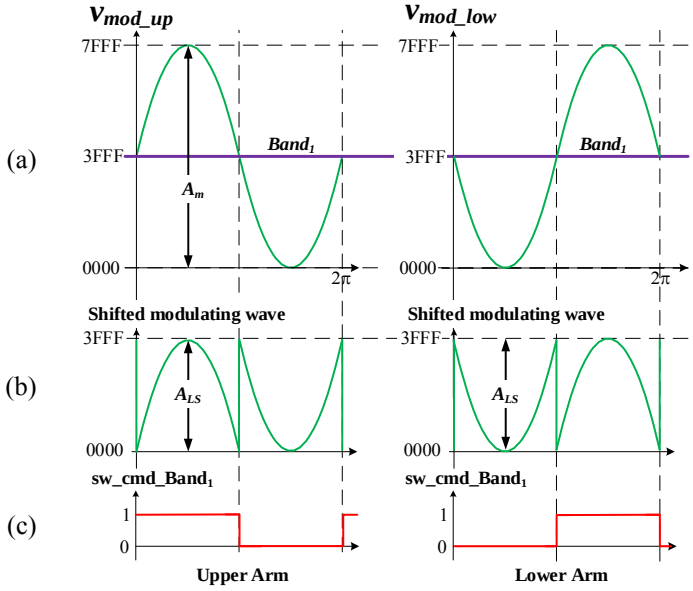


Figure 5. (a) Upper arm and lower arm modulating waveforms, (b) Shifted modulating waves for PWM, (c) gate signals obtained by comparing modulating wave with the level shifting bands.

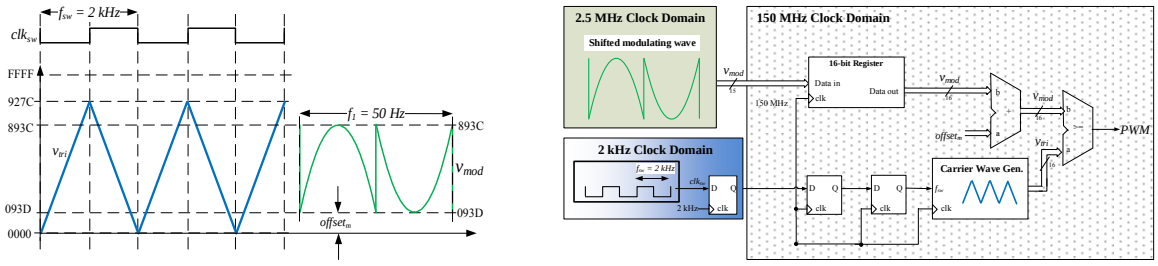


Figure 6. (a) The design of triangular wae generation which always keeps the level shifted modulating wave inside it, (b) Multi clock domains crossing in Modulation Module

3.4. Control Module (Capacitor Voltage Balancing Control)

The logic design discussed in this sub-section is customized for individual arm control. The functional logic can be duplicated to realize the entire MMC control by feeding in the appropriate arm's measurements, i.e. each PEBB's capacitor voltages level and arm current direction. The sampling event occurs every 250 µs. It is finely tuned to take place when the carrier wave reaches the top and bottom of the wave. The capacitor voltage, V_c, from each PEBB will be sorted in ascending order using parallel bubble sort (PBS) [33]. Before entering the PBS process, each V_c will be tagged with a unique ID. Figure 7 shows the data structure. In general, PBS always requires an even number of input data for the sorting process. Therefore, MMC with an odd number of PEBBs per arm may include an additional V_c (dummy value) and set the status bit, PEBB_NA_hardware to '1'. The sorting algorithm will sort this dummy (V_c) as the largest value and eliminate it at the end of the sorting process.

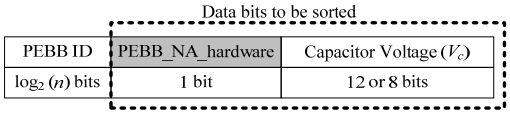


Figure 7. Input data structure for sorting algorithm, PEBB ID will bot be sorted

Figure 8 shows an example of PBS with four data inputs. These data will first be stored in array a_0, a_1, a_2, and a_3. Then they will be grouped in pair in sequence (a_x, a_y) and sorted in parallel. A sorting sequence is started with even pair sorting. An even pair is defined by the first element a_x having an even subscription number, such as (a_0, a_1) and (a_2, a_3). The sorted results will be used by odd pair sorting. An odd pair is defined by a_x having an odd subscription number element, i.e. (a_1, a_2). These steps (even-odd sorting) will be repeated. The desired number of required even-odd sorting sequences, n_{eos}, is always equivalent to half of the total number of input data. Therefore, for MMC with n PEBBs per arm, n_{eos} is defined as:

$$n_{eos} = \frac{n}{2} \tag{15}$$

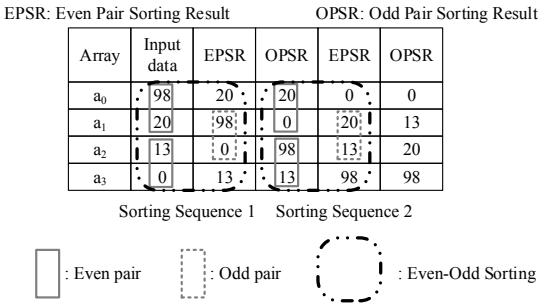


Figure 8. Parallel Bubble Sorting example with four input data

Figure 9 (a) shows the logic diagram to sort out two data inputs. This inputted data will first be compared. The comparison result triggers a pair of multiplexers to pass the appropriate input data to the corresponding registers. Register *Dout_min* keeps the smaller input data and register *Dout_max* stores the other data. By neglecting the signal propagation delay in a concurrent circuit, the sorting result will be available after one clock cycle, T. This logic block is named as a 2-data sorting module. It will be used as the modular block to accomplish the PBS. For example, three units of modular blocks will be connected as shown in Figure 9 (b) to accomplish the first sequence of even-odd sorting events presented in Figure 8. Thus, this circuit is called an even-odd sorting module. This process will complete in 2 clock cycles ($2T$). In order to optimize the logic design, the output result from the even-odd sorting module can be fed back to the input to repeat the even-odd sorting process. As a result, the design of PBS is fully optimized. For MMC with n units of PEBB, only (n–1) units of modular block will be required to develop a basic even-odd sorting module. The total processing time to accomplish PBS for n data, t_{PBS} can be derived as follows:

$$t_{PBS} = n_{eos} \times 2T \tag{16}$$

By substituting (15) into (16), this controller manages to sort out n PEBBs per arm in approximately n clock cycles. When PEBBs are arranged in ascending order, a simple look-up-table (LUT) can be used to distribute the switching commands.

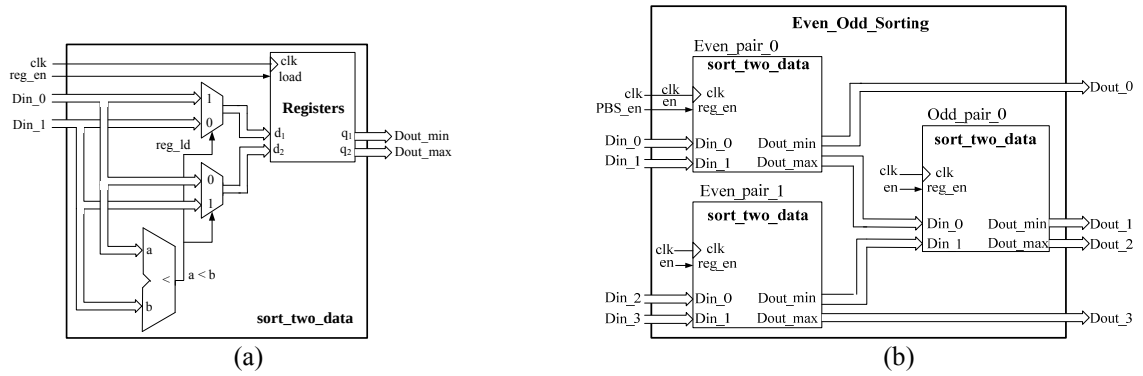


Figure 9. (a) 2-data sorting module (modular block), (b) Even-odd sorting module example for four input data

Figure 10 shows an LUT example for MMC. Assume two PEBBs are available on each arm. Input data for this LUT includes arm current direction, the position of the modulating wave, v_{mod} trajectory and the sorted PEBBs' ID. Since only 2 PEBBs are used, we divide the sorted PEBBs' ID input into "Lower V_c" and "Higher V_c". Arm current direction is clearly defined in

Figure 1 (c). The capacitor on the PEBB will be charged up by positive arm current and discharged by negative arm current. The modulating wave and level shifting bands for this example are given by Figure 5 (a). If the modulating wave trajectory is detected to swing between 3FFF and 7FFF, the PEBB sorted in the "Lower V_c" column must be allocated with $sw\_cmd\_Band_1$. This will enable the capacitor to be charged for the entire switching period (100% duty cycle). Thus, the capacitor voltage level will rise due to this charging activity. Another PEBB which is sorted in the "Higher V_c" column will carry out the PWM switching command (*sw_cmd_PWM*). Inversely, if the negative arm current is sampled, the PEBB with a higher V_c value should be fully turned on for discharging, while another PEBB should carry out the PWM duty cycle.

Arm current	Position of Modulating wave, v_{mod} trajectory	Sorted PEBB in ascending order	
		Lower V_c	Higher V_c
$i_{arm} > 0$	3FFF < v_{mod} < 7FFF	0	1
	0000 < v_{mod} < 3FFF	1	0
$i_{arm} < 0$	3FFF < v_{mod} < 7FFF	1	0
	0000 < v_{mod} < 3FFF	0	1
		(DO_c0)	(DO_c1)

LUT Output legend:
0 : sw_cmd_Band1
1 : sw_cmd_PWM

Figure 10. Capacitor voltage balancing control LUT example referring to LSPWM in Figure 5.

The LUT output data are distributed via a set of de-multiplexers. The sorted PEBB's ID will be used as the demux select signal. Figure 11 shows the demux circuit for example in Figure 10. Lastly, some OR gates are used to wrap up each individual PEBB mux select signal:

$$PEBB_{i\_muxsel} = PEBB_{i\_DO\_c0} + PEBB_{i\_DO\_c1} + \ldots + PEBB_{i\_DO\_cn} \quad (17)$$

where $i = 1, 2, \ldots, n$.

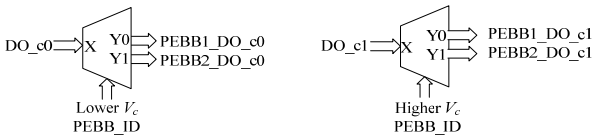


Figure 11. 2 units of de-multiplexer are required for example given in Figure 10. to distribute the LUT output data for each individual PEBB

3.5. Switching Commands Distribution Module

A total of 2*n* units of multiplexer will be employed in this module. They are used to distribute the duty cycles for each PEBB in one phase. For an MMC with 2 units of PEBB per arm, this module will employ four units of multiplexer. The multiplexer circuit for upper arms' PEBB U1 is drawn in Figure 12. It is clearly shown that switching commands ($sw\_cmd\_Band_1$ and *sw_cmd_PWM*) are connected as the input signals to the multiplexer. A valid switching command will be selected based on the results determined from the Control Module, i.e. via PEBB_U*_mux_sel and PEBB_L*_mux_sel signals.

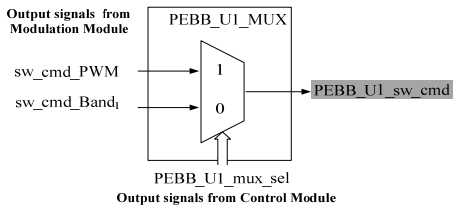


Figure 12. A multiplexer used to distribute the switching command for upper arm PEBB_U1

4. EXPERIMENTAL RESULTS

A small scale MMC with 2 units of PEBB per arm will be used to evaluate the design presented in Section 3. Table 1 summarizes the parameters of the prototype. This experiment will be conducted on one phase as shown in Figure 13. The MMC controller is configured into Xilinx XC7Z020 programmable logic. Capacitor voltage measurements will be sampled in continuous sequential mode using Xilinx built in ADC. Arm currents are sampled synchronously using a pair of ADCs. The duty cycles for each individual PEBB are directly sent out through Xilinx FPGA Mezzanine Card (FMC) XM105 Debug card I/O ports. Table 2 depicts the used resources of the entire design. The maximum operating frequency of the MMC controller is 25 MHz.

Figure 14 shows the voltages and currents results. Figure 14 (a) indicates two-level arm voltages are generated. These voltages produce a five-level load voltage, *Vao*. The upper arm voltage, *V_Uarm* and lower arm voltage, *V_Larm* are 180° phase shifted. They are constantly regulated at 5V and 10V levels. These results have proven that the proposed control and modulation schemes are implemented correctly. Arm current waveforms are depicted in Figure 14 (b). A sinusoidal load current, *Iao*, is obtained in this test.

An extension to a global MMC topology with variable number of PEBBs could be made. However using Xilinx Zynq ZC702 board, PEBBs on each phase of MMC are limited to less than 10 units. The number of available I/O pins and complexity of wiring system become the main constraint of implementation. Therefore, a ring control interface has been proposed [34].

Table 1. Parameters of the MMC Setup

Rated power	500 W
Number of PEBBs per arm	2 units
SM's capacitance	6800 µF, 100 V
Arm inductance, L_{arm}	1.3 mH
DC Link Voltage, V_{dc}	10 V
DC link Capacitor	3300 µF, 450 V
RL Load	R = 5 Ω, L = 8mH
Number of output voltage levels	5
Switching frequency	2 kHz

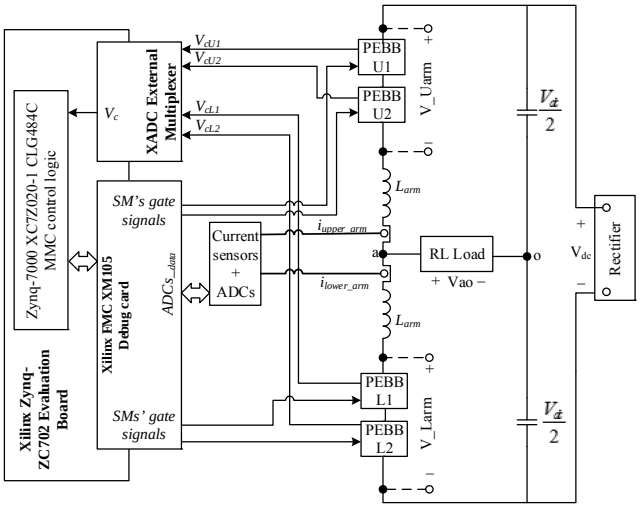
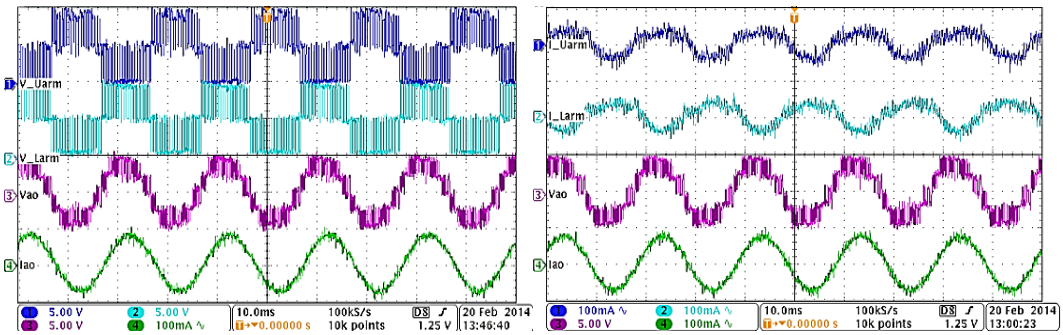


Figure 13. Experimental setup for MMC with two PEBB per arm

Table 2. Resources sed in MMC Controller

Xilinx XC7Z020	106400 Flip Flop	53200 LUTs	17400 Memory LUT	200 I/O	140 BRAM	32 BUFG	4 MMCM
MMC Controller	2044 1.92 %	1633 3.07 %	69 0.40 %	36 18%	7 5.00%	6 18.75%	1 25.00%



(a) (b)

Figure 14. Experiment results: (a) Upper arm voltage, V_Uarm; lower arm voltage, V_Larm; load voltage, Vao and load current, Iao; (b) Upper arm current, I_Uarm; lower arm current, I_Larm; load voltage, Vao and load current, Iao.

5. CONCLUSION

This paper has presented full programmable logic designed of capacitor voltage balancing control and level-shifted PWM (LSPWM) strategies in modular multilevel converter (MMC). A high frequency clock (150 MHz) is used in the proposed design. Hence, it is possible to sort out a hundred measurements of capacitor voltages in a processing time of less than 10 µs. Parallel Bubble Sort algorithm is implemented using a basic set of even-odd-sorting modules. The design is optimized by feeding back the sorting results using multiplexers to accomplish the subsequence sorting process. Differing from the conventional LSPWM, this design only implements a single triangular carrier wave. However, the modulating wave is shifted accordingly to the total number of PEBBs employed in each arm of the MMC. Thus, the gate counts are further minimized. The feasibility of the proposed controller design has been evaluated experimentally using one phase of MMC. This MMC prototype consists of two units of PEBBs per arm. As a result, the MMC produces a 5-level output voltage with a sinusoidal current wave.

REFERENCES

[1] CK Kim, *et al.*, HVDC Transmission Power Conversion Applications in Power Systems, Singapore: John Wiley & Sons, 2009.
[2] B.R. Andersen, *et al.*, "Topologies for VSC transmission", *Power Eng. Journal,* vol. 16, no. 3, pp. 142-150, 2002.
[3] N. Flourentzou, *et al.*, "VSC-Based HVDC Power Transmission Systems: An Overview", *IEEE Transactions on Power Electronics,* vol. 24, no. 3, pp. 592-602, 2009.
[4] F. Wang, *et al.*, "An overview introduction of VSC-HVDC: State-of-art and potential applications in electric power systems", in *Cigrè International Symposium*, Bologna, 2011.
[5] X. Shi, *et al.*, "Characteristic Investigation and Control of a Modular Multilevel Converter-Based HVDC System Under Single-Line-to-Ground Fault Conditions", *IEEE Trans. Power Electronics,* vol. 30, no. 1, pp. 408-421, Jan 2015.
[6] S. Madichetty, D. Abhijit, S Jinka, "A survey and experimental verification of Modular Multilevel Converters", *International Journal of Power Electronics and Drive System (IJPEDS),* vol. 4, no. 3, pp. 363 - 375, 2014.
[7] T. Ericson, *et al.*, "PEBB - Power Electronics Building Blocks From Concept to Reality", in *The 3rd IET International Conference on Power Electronics, Machines and Drives*, 2006.
[8] A. Iyer, *et al.*, "Validation of the Plug-and-Play AC/AC Power Electronics Building Block for Medium Voltage Frid Control Applications", *IEEE Trans. on Industry Applications,* vol. 50, no. 5, pp. 3549 – 3557, Sept 2014.
[9] M. Glinka, "Prototype of multiphase modular-multilevel-converter with 2MW power rating and 17-level-output-voltage", in *IEEE 35th Power Electronics Specialists Conference*, Aachen, Germany, 2004.
[10] R. Marquardt, "Modular multilevel converter: an universal concept for HVDC-network and extended DC-bus-applications", in *International Power Electronics Conference*, 2010.
[11] B. Gemmell, *et al.*, "Prospects of multilevel VSC technologies for power transmission", in *Proc. IEEE/PES T&D Conf. Expo*, 2008.

[12] C. Buccella, *et al.*, "Digital control of power converters - a survey", *IEEE Trans. on Industrial Informatics,* vol. 8, no. 3, pp. 437-447, 2012.
[13] E. Monmasson, *et al.*, "FPGA based controllers", *IEEE Ind. Electron. Mag. ,* vol. 5, no. 1, pp. 14-26, Mar 2011.
[14] C. Sepulveda, *et al.*, "FPGA v/s DSP performance comparison for a VSC-based STATCOM control application", *IEEE Trans. Ind. Informatics,* vol. 9, no. 3, pp. 1351-1360, Aug. 2013.
[15] E. Monmasson, and M. N. Cirstea, "FPGA design methodology for industrial control systems-A review", *IEEE Trans. Ind. Electron.,* vol. 8, no. 3, pp. 1824-1842, 2007.
[16] H. Nademi, *et al.*, "Modular Multilevel Converter with adaptive observer of capacitor voltages", *IEEE Trans. on Power Electronics,* vol. 30, no. 1, pp.235 - 248, Jan 2015.
[17] S. Fan, *et al.*, "An improved control system for Modular Mulltilevel Converters with new modulation strategy and voltage balancing control", *IEEE Trans. on Power Electronics,* vol. 30, no. 1, pp. 358 – 371, Jan 2015.
[18] B. Li, *et al.*, "Analysis of the Phase-Shifted Carrier modulation for Modular Multilevel Converters", *IEEE Trans. on Power Electronics,* vol. 30, no. 1, pp. 297 – 310, Jan 2015.
[19] L. He, *et al.*, "A Repertitive control scheme for harmonics suppression of circulating current in Modular Multilevel Converters", *IEEE Trans. on Power Electronics,* vol. 30, no. 1, pp. 471 – 481, Jan 2015.
[20] A. Lesnicar, and R. Marquardt, "An innovative modular multilevel converter topology suitable for a wide power range", in *IEEE PowerTech Conference*, Bologna, 2003.
[21] M.A. Perez, *et al.*, "Circuit Topologies, Modelling, Control Schemes and Applications of Modular Multilevel Converters", *IEEE Trans. on Power Electronics,* vol. 30, no. 1, pp. 4 - 17, Jan 2015.
[22] K. Ilves, *et al.*, "A new modulation method for the Modular Multilevel Converter allowing fundamental switching frequency", in *IEEE 8th International Conference on Power Electronics-ECCE Asia*, Korea, 2011.
[23] L. Xu, Y. Han, "Study on control strategies for the multilevel cascaded converters", *TELKOMNIKA (Telecommunication Computing Electronics and Control),* vol. 11, no. 11, pp. 6329 - 6336, 2013.
[24] S. Rohner, *et al.*, "Modulation, Losses and Semiconductor Requirements of Modular Multilevel Converters", *IEEE Trans. Ind. Electron.,* vol. 57, no. 8, pp. 2633-2642, 2010.
[25] R.S. Alishah, S.H. Hosseini, "A new multilevel inverter structure for high-power applications using multi-carrier PWM switching strategy", *International Journal of Power Electronics and Drive System (IJPEDS)*, vol. 6, no. 2, pp. 318 -325, 2015.
[26] "ZC702 Evaluation board for the Zynq-7000 XC7Z020 All Programmable SoC User Guide", Xilinx, Jun 2014.
[27] "7 Series FPGAs Clocking Resources User Guide", Xilinx, Nov 2014.
[28] "LogiCORE IP Clocking Wizard v5.0", Xilinx, Oct 2014.
[29] V. Matic, *et al.*, "Comparison of digital signal processing methods for sine wave signal generation", in *IEEE International Telecommunications Symposium*, 1998.
[30] T. Abeyasekera, *et al.*, "Elimination of subharmonics in direct look-up table sine wave reference generators for low-cost microprocessor-controlled inverters", *IEEE Transactions on Power Electronics,* vol. 18, no. 6, pp. 1315 - 1321, 2003.
[31] "LogiCORE IP DDS Compiler v6.0", Xilinx, Oct 2014.
[32] S. Verma, and A.S. Dabbare, "Understanding clock domain crossing issues", *EE Times-India,* pp. 1-7, 2007.
[33] Rene Mueller, *et al.*, "Sorting Networks on FPGAs", *The VLDB Journal,* vol. 21, no. 1, pp. 1-23, 2012.
[34] S. Huang, *et al.*, "Analysis of communication based distributed control of MMC for HVDC", *15th European Conference on Power Electronics and Application*, Lille, 2013.