Development of an FPGA-Based Gate Signal Generator for a Multilevel Inverter

F. Salim[1], N. Ahmad Azli[2]
[1] Electrical Engineering Department, Program Pengajian Diploma, Universiti Teknologi Malaysia, Malaysia.
[2]Energy Conversion Department, Faculty of Electrical Engineering, Universiti Teknologi Malaysia, Malaysia.

Abstract—**The application of Field Programmable Gate Array (FPGA) in the development of power electronics circuits control scheme has drawn much attention lately due to its shorter design cycle, lower cost and higher density. This paper presents an FPGA-based gate signal generator for a multilevel inverter employing an online optimal PWM switching strategy to control its output voltage. FPGA is chosen for the hardware implementation of the switching strategy mainly due to its high computation speed that can ensure the accuracy of the instants that gating signals are generated. The gate signal generator has been realized by an FPGA (FLEX10K20) from Altera. The design and development of the FPGA based gate signal generator is described in detail. Results from the timing simulation using MAX+PLUSII software are given and verified by the results obtained from the FLEX10K20 output.**

Keywords—**FPGA, optimal PWM, multilevel inverter.**

I. INTRODUCTION

The first concept of multilevel inverter studies began at the end of the seventies, but no practical use of it was made then [1]. In 1981, the Neutral Point Clamped (NPC) inverter was introduced [2]. In the early nineties, the interest in the multilevel structure was renewed with the emergence of various circuit topologies and modulation strategies. There are three main types of multilevel inverters, which can be classified as Diode Clamped Multilevel Inverter (DCMI), Imbricated Cells Multilevel Inverter (ICMI) and Modular Structured Multilevel Inverter (MSMI).

A work reported earlier [3][4][6] on the development of an online optimal PWM switching strategy for a 5-level MSMI employed a dSPACE DS1102 controller board to fulfill the tasks of calculating in real time the optimal PWM switching angles as well as generate the gating signals for each of the MSMI power devices. The DS1102 controller board is based on the Texas Instruments TMS320C31 floating-point DSP, which builds the main processing unit, providing fast instruction cycle time for numeric intensive algorithm. However, using the DS1102 controller board, the tasks mention above can only be accomplished within the sampling time of 100μsec that corresponds to a 1.8° resolution for the switching angles.

A large sampling interval holds the effect of deviating the timings of the gating signals generated by the DSP from the actual optimal PWM switching angles calculated by it using derived functions that will be explained later. This paper proposes the incorporation of an FPGA-based gate signal generator with the DSP to relief the later from time consuming computation task of the gating signals generation. Thus, the overall sampling time to generate the gating signals for the MSMI power devices can be reduced further for better timing precision. Description on the MSMI circuit topology is given in the following section. This is followed by the details on the MSMI control operation system and the development of the FPGA-based gate signal generator. The results obtained from the timing simulation and the actual FPGA output is presented for comparison purposes. Finally, a conclusion is made based on the work that has been done.

II. THE MSMI

In general, an MSMI consists of *(nl-1)/2* or *h* number of single-phase H-bridge inverters referred to as MSMI modules, that are connected in series to generate an *nl* level output phase voltage. A single-phase structure of a single-phase MSMI is shown in Fig. 1. The MSMI output phase voltage is equal to the summation of the output voltages of the respective modules that is,

$$V_o = V_{m1} + V_{m2} + \ldots\ldots V_{mh} \quad (1)$$

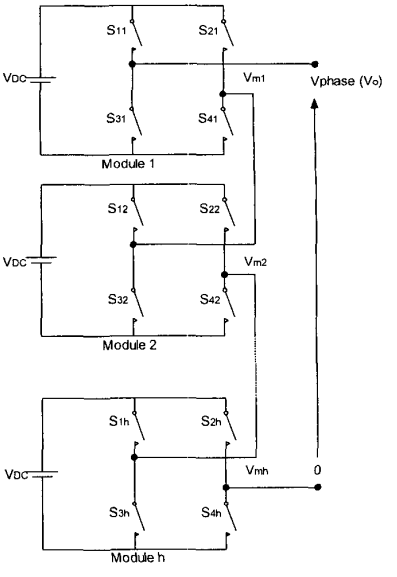

Fig. 1. Structure of a single-phase *nl*-level MSMI

Each MSMI module has its own DC source (V_{DC}) and consists of four power devices designated as S_{1r}, S_{2r}, S_{3r} and S_{4r} where r = 1,2, ... *h*. Each MSMI module can generate a three-level output namely + V_{DC}, 0 and – V_{DC}. This is made possible by connecting the DC source sequentially to the AC side via the four power devices. As an example, for an output phase voltage consisting of 5 levels, which are $+2V_{DC}$, $+V_{DC}$, 0, $-V_{DC}$ and $-2V_{DC}$, the number of modules required in the MSMI is two.

The MSMIs are known to eliminate the excessively large number of clamping diodes required by the DCMIs and capacitors required by the ICMIs. Its circuit configuration is simple and modular where by each of its module is identical and incorporate both input and output circuitry [5]. In addition it also requires the least number of components compared to the other types of multilevel inverter. These features provide the flexibility in extending the MSMI to higher number of levels without undue increase in circuit complexity simplifies fault finding and repair as well as facilitates packaging.

III. THE MSMI CONTROL OPERATION

Fig. 2 shows the block diagram of the MSMI control operation system. The amplitude of the fundamental of the MSMI output voltage in per unit (*ap1*) values are represented by a DC power supply output voltage that is varied between 0 and 10 V. This output voltage corresponds to the *ap1* values between 0 and 1. Depending on the *ap1* value, the DSP on the DS1102 controller board calculates the switching angles based on the online optimal PWM switching strategy. The FPGA then accepts the relevant information from the DSP that is later translated to counter values for it to generate the gating signals accordingly.

In this work, the output of the FPGA will not be tested on the MSMI circuit. Instead, concentration will be made on the development of the gate signal generator only. This covers the aspect of identifying the parameters that need to be fed to the FPGA, modifying the C program developed for DSP code generation so that it functions only as a optimal PWM switching angles calculator and designing the FPGA so that it will generate the gate signals based on the switching angles values effectively and efficiently.

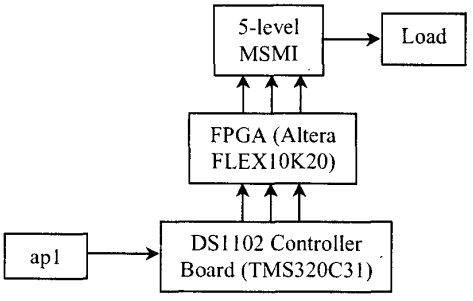

Fig. 2. Block Diagram of the MSMI control operation system

A. The DS1102 Controller Board

From the voltage signal sent to represent a particular ap1 value, the switching angles are calculated by the DS1102 controller board using the functions that represent each of the switching angles solution trajectories of the optimal PWM switching strategy. The functions are in the form of [6],

$$F_k = a_0 + a_1[(ap1\text{-}m)] + a_2[(ap1\text{-}m)^2] + \ldots\ldots a_y[(ap1\text{-}m)^y] \quad (2)$$

where, k = 1 and 2, ap1-m is the amplitude of the fundamental of each MSMI module's output in per unit value. Examples of the coefficient values used are as given by Table 1. The functions are accessed based on the multilevel control design [4][6] of the 5-level MSMI, which depends on the ap1 values. Fig. 3 generally illustrates how the multilevel control design is implemented on the DSP and the way the functions are being accessed to calculate the optimal PWM switching angles.

TABLE 1
EXAMPLES OF THE COEFFICIENT VALUES RELATED TO F_{K1} AND F_{K2}

	F_1		F_2	
	F_{11}	F_{12}	F_{21}	F_{22}
a_0	2392.4	-28.917	2392	-171.17
a_1	-3589.1	77.635	-3588.8	584.96
a_2	2034.1	-78.595	2033.1	-797.85
a_3	-539.5	30.76	-539.4	538.28
a_4	62.1	14.316	69	-177.49
a_5	17.1	-	17.1	44.045

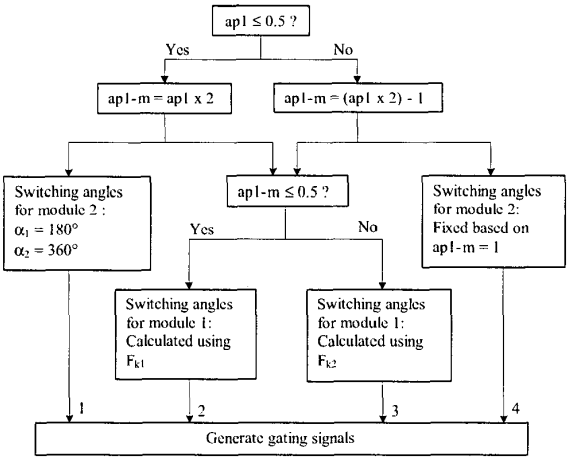

Fig. 3. Implementation of the multilevel control design on the DSP

B. The FPGA

The limitations in the number of input-output pins on the DS1102 controller board makes it impossible to feed the switching angles values directly to the FPGA. Thus, the approach taken in this case is to group the switching angles calculated online by the DSP depending on the *ap1* value into categories known as *case number*. Each *case number* relates to certain timing values of the gate signal, which are pre-calculated and stored in the FPGA. In other words, the FPGA generates the gate signals based on the *case number* received from the controller board. The *case number* can be determined by referring to all the count values for each *ap1* value. In this work, the value of count is calculated for *ap1* in steps of 0.02 and 25 cases have been determined.

The fixed value of count for each MSMI module is also stored in the FPGA. The count value for the power devices in module 1 of the MSMI is called *ncounter* while for power devices in module 2 the count value is known as *counter*. For example, the count value for module 1 of the MSMI can be calculated using (2) where α is the switching angle.

$$counter = \frac{0.01\alpha}{180(samplingtime)} \quad (2)$$

After all values of the *ncounter* and *counter* for each value of *ap1* have been calculated, the *case number* is determined. This number represents different sets of count values. Based on the relevant values of *ncounter* and *counter* identified by the FPGA for each *case number*, the gate signals are generated. The design based on this approach is described in schematic and Very High Speed Integrated Circuit Hardware Description Language (VHDL). It is then verified via both functional and timing simulations using the MAX+PLUSII software.

Fig. 4 shows the top-level module for the gate signal generator. Module *counter58us* and *ncounter58us* represent the timing values while module *gensig58us* serves to generate the gate signals. The design is then downloaded to the FPGA, in this case the FLEX10K20, which continues to operate as long as it is power up.

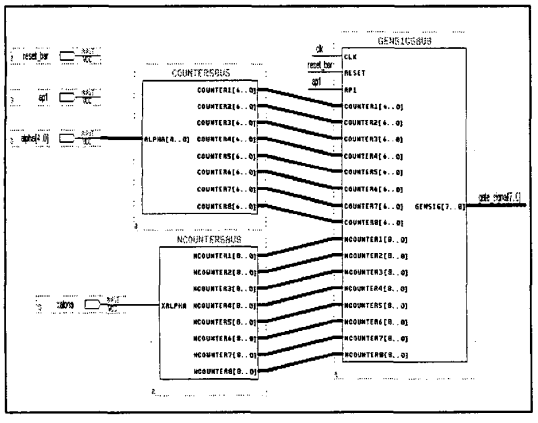

Fig. 4. Schematic diagram of the gate signal generator

IV. RESULTS AND ANALYSIS

With the introduction of the FPGA based gate signal generator, the DSP function has been limited to only calculating the switching angles. The sampling time to generate the gate signals has been reduced to 58 μsec and the switching angles resolution is 1.044°. This value represents the sampling time achievable by the DSP in handling the tasks of calculating the optimal PWM switching angles and determining the *case number*. The FPGA is actually constrained to this sampling time in order to generate the gate signals. Fig. 5 and Fig. 6 show the simulation results obtained from the gate signal generator for *ap1* = 0.4 and *ap1* = 0.7. Fig. 7and Fig. 8 show the results obtained from the FLEX10K20 output for *ap1* = 0.4 and *ap1* = 0.7 that represents the actual gate signals for the MSMI power devices. Only four signals are shown, as the other four signals are merely the complement of these four signals as can be depicted from the simulation results.

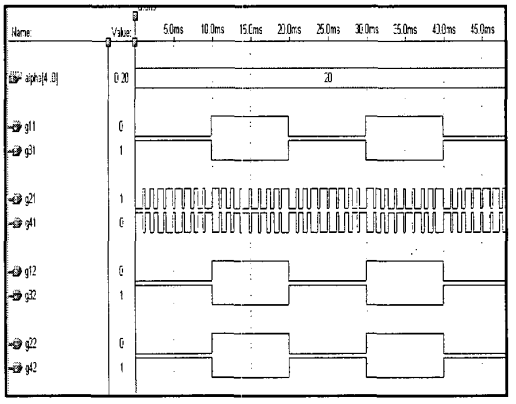

Fig. 5. Simulation results for *ap1* = 0.4

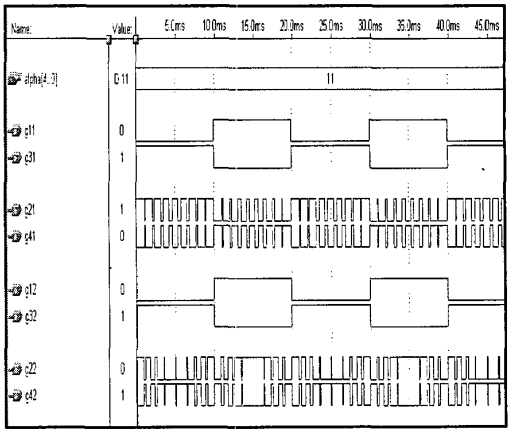

Fig. 6. Simulation results for *ap1* = 0.7

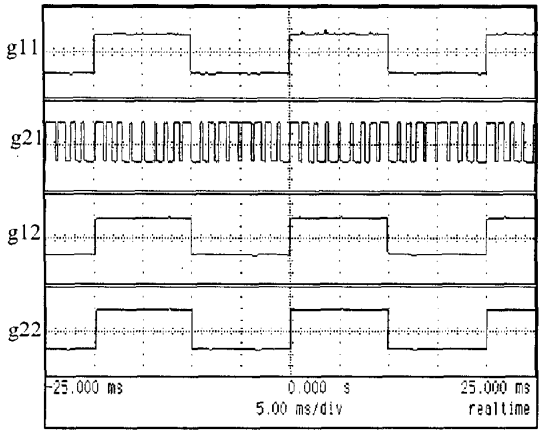

Fig. 7. Gate signals generated by the FLEX10K20 for $ap1 = 0.4$

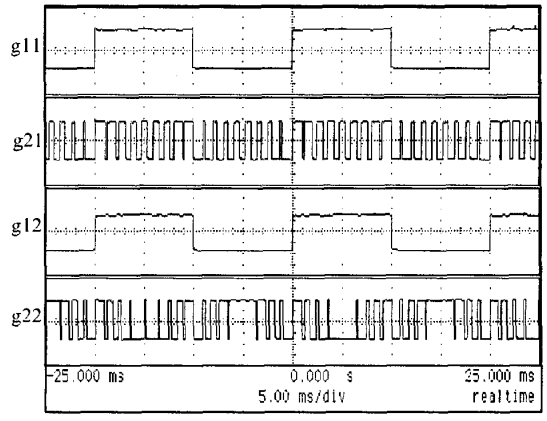

Fig. 8. Gate signals generated by the FLEX10K20 for $ap1 = 0.7$

The gate signals generated by the FLEX10K20 show good agreement to those obtained from the functional and timing simulations using the MAX+PLUSII software. Based on a hybrid PWM switching arrangement [6], any pair of the four power devices in each MSMI module is pulsewidth-modulated at a higher frequency while the other pair is commutated at a low output voltage fundamental frequency. In this work, S_{11} and S_{31} of module 1 and S_{12} and S_{32} of module 2 are assigned to switch at the MSMI output voltage fundamental frequency of 50 Hz. On the other hand, S_{21} and S_{41} of module 1 and S_{22} and S_{42} of module 2 are assigned to switch at the higher switching frequency. These features can be depicted from Fig. 5 to Fig. 8.

V. CONCLUSION

This paper presents the application of an FPGA in developing a gate signal generator for an MSMI based on the optimal PWM switching angles calculated by a DSP on a DS1102 controller board. The design and implementation of the gate signal generator using the MAX+PLUSII software and Altera's FLEX10K20 have been described. With the introduction of the FPGA based gate signal generator, the DSP's task has been reduced thus reducing the sampling time required in calculating the optimal PWM switching angles and determining the relevant *case number* as the input to the FPGA.

REFERENCES

[1] S. M. Tenconi, M. Carpita, C. Bacigalupo, and R. Cali, " Multilevel voltage source converters for medium voltage adjustable speed drives." in *Proc. of the IEEE International Symposium on Industrial Electronics ISIE'95*, pp. 91-98.

[2] A. Nabae, I, Takahashi and H. Akagy, "A neutral point clamped PWM inverter." *IEEE Transactions on Industry Applications*, vol. 17, no. 5, pp. 518-523, 1981.

[3] N. A. Azli and A. H. M. Yatim, "Modular structured multilevel inverter (MSMI) for high power AC power supply application", in *Proc. of the IEEE International Symposium on Industrial Electronics ISIE' 01*, Pusan, Korea, pp. 728–733..

[4] N. A. Azli, and A. H. M. Yatim, "Optimal pulsewidth modulation (PWM) online control of a modular structured multilevel inverter (MSMI)", presented at the 4th IEEE International Conference on Power Electronics and Drive Systems, Bali, Indonesia, 2001.

[5] C. Newton and M. Sumner, "Multi-level Convertors a Real Solution to Medium/high-voltage Drives?." *Power Engineering Journal*,. pp. 21-26, 1998.

[6] N. A. Azli,, "Online optimal pulsewidth modulation (PWM) for a modular structured multilevel inverter," Ph.D. dissertation, Elect. Eng. Program, Univ. Teknologi Malaysia, Johor, Malaysia, 2002.