Journal of Multidisciplinary Engineering Science Studies (JMESS)
ISSN: 2458-925X
Vol. 11 Issue 11, November - 2025

Design and simulation of inverter control SHEPWM tecnique based

Ing.Luis Antonio Gaytán García
Departamento de Estudios Multidiciplinarios
Universidad de Guanajuato
Yuriria, Guanajuato, México
La.gaytangarcia@ugto.mx

Dr.Misael López Ramírez
Departamento de Estudios Multidiciplinarios
Universidad de Guanajuato
Yuriria, Guanajuato, México
Lopez.misael@ugto.mx

Dr.Homero Miranda Vidales
Universidad Autónoma de San Luis Potosí
San Luis Potosí, México
hmirandav@uaslp.mx

***Abstract*—This article provides an overview of the development of a digital architecture for generating the control signal for a current inverter. It revisits the SHEPWM technique, which was studied a few decades ago and implemented in microcontrollers. Now, with design and simulation tools, its behavior can be observed in hardware designed specifically for this purpose.**

Keywords—Inverter; SHEPWM; VHDL; FPGA;

I. INTRODUCTION

The use of inverters is a crucial issue today, as there has been an increase in electricity consumption for the development and use of technology in our society. Thus, due to the high demand for electricity, efforts have been made to improve the procurement of renewable energy sources, such as solar energy, which requires an inverter system for its use [1]. Likewise, in the industrial sector, which relies on the operation of electric motors, three-phase inverters play a fundamental role in the use of appropriate modulation techniques to obtain low harmonic distortions [2].

The different modulation techniques are applied in the logic control stage of the inverter, which provides the switching patterns for the power semiconductors. Today, field-programmable gate arrays (FPGAs) are more accessible, and their use for this function has been increasing due to their parallel behavior, which makes them powerful tools for this task [3]. Therefore, in recent years there have been studies on the use of these devices, such as a random PWM technique to reduce the amplitude of emission peaks while maintaining a constant switching frequency [4], the implementation of a high-precision digital pulse width generator (DPWM) with excellent linearity to provide real-time control [5], and even the development of an FPGA-based scheme for real-time inference of neural networks [6]. With design languages such as VHDL that allow us to describe the hardware and simulation tools to see how our design behaves, we can develop projects like this one where we will obtain the expected signals and save time and costs in their implementation.

This article discusses the technique of pulse width modulation by selective harmonic elimination (SHEPWM), also known as programmed PWM, which is based on the control of rising/falling edges using pulses by comparing angles that are calculated in a predetermined manner, so that the output waveform is fixed [7]. A digital architecture is designed, which is capable of generating the waveform for a SHEPWM with third harmonic elimination. To do this, numerical methods are used to calculate the angles on a computer in order to subsequently obtain the simulation and verify that it provides the desired waveform.

II. OBTAINING THE COEFFICIENTS

The waveform of the 0° to 180° angle of SHEPWM with third harmonic elimination is shown in Figure 1. There are 6 switching angles $\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6$.

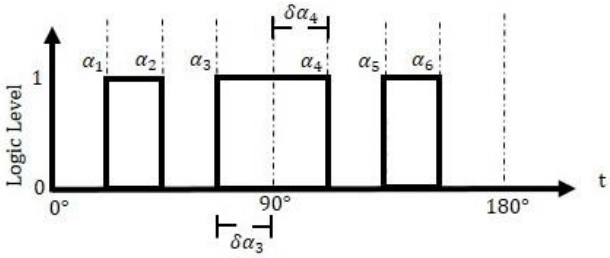

Figure 1. SHEPWM waveform with third harmonic elimination.

The SHEPWM technique began to be studied a few decades ago [8], and it is known that the coefficients for switching at the logic level that forms a quarter of the waveform are given by equation (1).

$$a_n = \frac{4}{n\pi}\left[1 + 2\sum_{k=1}^{M}(-1)^k \cos n\alpha_k\right] \quad (1)$$

Using the functions and numerical methods provided by MATLAB software, we obtain α_1, α_2, and α_3, which correspond to angles up to 90° in a signal

with a fundamental frequency of 60 MHz. Also, using the functions provided by this computer tool, we can perform calculations based on differential equations to obtain different voltage amplitudes at the output in relation to the input [9]. Since it is a symmetrical waveform, to calculate the angles α_4, α_5, and α_6, which are up to 180°, as shown in Figure 1, $\delta\alpha_3$ is the distance from α_3 to 90°, and α_4 is at the same distance, therefore $\delta\alpha_3 = \delta\alpha_4$. If we have to

$$\begin{aligned}\delta\alpha_3 &= 90° - \alpha_3 = \delta\alpha_4 = \alpha_4 - 90°\\ \delta\alpha_2 &= 90° - \alpha_2 = \delta\alpha_5 = \alpha_5 - 90°\\ \delta\alpha_1 &= 90° - \alpha_1 = \delta\alpha_6 = \alpha_6 - 90°\end{aligned} \quad (2)$$

Since we know the values of α_1, α_2, α_3, we can calculate the values of $\delta\alpha_1, \delta\alpha_2, \delta\alpha_3$ and from equations (2) we obtain that

$$\begin{aligned}\alpha_4 &= 90° + \delta\alpha_3\\ \alpha_5 &= 90° + \delta\alpha_2\\ \alpha_6 &= 90° + \delta\alpha_1\end{aligned} \quad (3)$$

With all angles obtained within 180°, we can calculate the entire waveform up to 360°, which also represents the reverse conduction cycle, as shown in Figure 2, where T_o is the period of the fundamental frequency in our load.

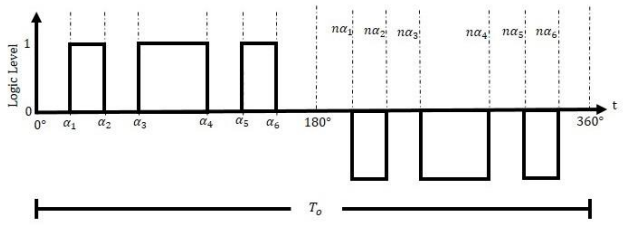

Figure 2.Waveform 360° SHEPWM

To calculate the values of $n\alpha_1$, $n\alpha_2$, $n\alpha_3$, $n\alpha_4$, $n\alpha_5$, $n\alpha_6$, simply shift all the values obtained previously by 180°. This method is just one of several that exist [10], and other software can also be used to perform the corresponding numerical methods and algorithms. The values obtained may vary in the decimal part depending on the tool used, but the important thing is to obtain the angle values. Tables 1 and 2 are attached with the values obtained using the method described in this article, which were calculated for different amplitudes.

Table 1. Angles from 0° to 180°

	α_1	α_2	α_3	α_4	α_5	α_6
50%	4.4°	66.4°	84.5°	95.4°	113.5°	175.5°
60%	5.2°	67.8°	83.4°	96.5°	112.1°	174.7°
70%	6.1°	69.2°	82.4°	97.5°	110.7°	173.8°
80%	7.0°	70.7°	81.4°	98.5°	109.2°	172.9°
90%	7.8°	72.3°	80.6°	99.3°	107.6°	172.1°
100%	8.6°	74.3°	80.2°	99.7°	105.6°	171.3°

Table 2. Angles from 180° to 360°

	$n\alpha_1$	$n\alpha_2$	$n\alpha_3$	$n\alpha_4$	$n\alpha_5$	$n\alpha_6$
50%	184.4°	246.4°	264.5°	275.4°	293.5°	355.5°
60%	185.2°	247.8°	263.4°	276.5°	292.1°	354.7°
70%	186.1°	249.2°	262.4°	277.5°	290.7°	353.8°
80%	187.0°	250.7°	261.4°	278.5°	289.2°	352.9°
90%	187.8°	252.3°	260.6°	279.3°	287.6	352.1°
100%	188.6°	254.3°	260.2°	279.7°	285.6°	351.3°

III. Development of a Digital Architecture for SHEPWM Based on Comparators

The single-phase half-bridge inverter can be represented as shown in Figure 3, where there are two semiconductor devices connected in series with a load connected to this junction. Note that whenever one transistor is activated, the other must remain deactivated. If ***Qa*** is activated and ***Qa'*** is deactivated, a voltage ***Va*** appears across the load. Conversely, if ***Qa'*** is activated and ***Qa*** is deactivated, the load will have a voltage ***–Va***. It is always necessary to prevent both semiconductors from being activated at the same time, as this would cause a short circuit.

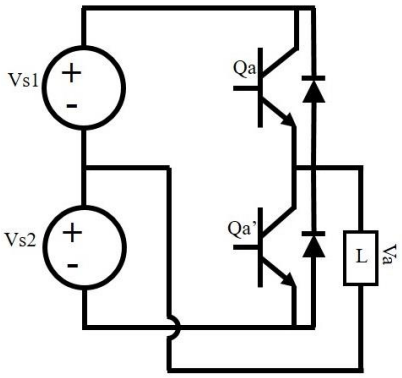

Figure 3. Half-bridge inverter

If the waveform shown in Figure 2 is as expected for the load in Figure 3, then we would have a voltage as shown in Figure 4. Note how from 0° to 180° the signal is already known for our SHEPWM, and from this point onwards, it is the complement of the same.

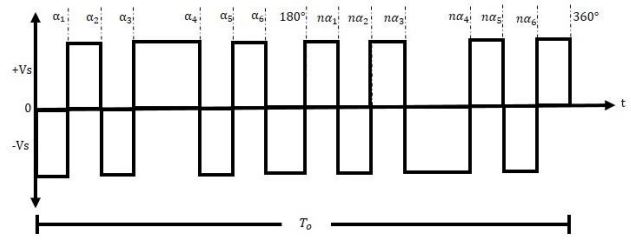

Figure 4. Voltage in the half-bridge inverter with the SHEPWM waveform

To begin the hardware design, the first step is to generate a fundamental frequency signal of 60 Hz, where its period T_o is given by

$$T_o = \frac{1}{F_o} \tag{4}$$

Therefore

$$T_o = \frac{1}{60} = 16.6ms \tag{5}$$

We have $T_o = 16.6\,ms$, remembering that the SHEPWM coefficients are given in degrees, and that the complete period forms 360°. With these two pieces of information, we can calculate the duration of each degree as follows

$$\begin{aligned} T_o &= 360° \\ T_o &= 16.6ms \\ 360° &= 16.6\,ms \\ 1° &= \frac{16.6\,ms}{360} = 46.1\mu s \end{aligned} \tag{6}$$

There is an up counter that starts at 0 and overflows at 359, and each count is activated by a timer signal that counts 46.1 µs, so we would have a signal with the fundamental period T_o. In addition, the count signal would be used as a reference for the SHEPWM angles. As shown in Figure 5, the synchronous system is connected to a 50 MHz clock signal, which is the working standard on FGPA boards, with an asynchronous reset signal.

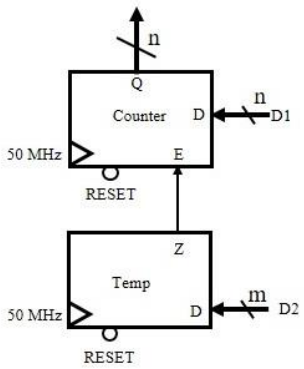

Figure 5. Timer-activated counter with overflow value.

The counter entity has an input **E** that enables counting and a **D** input that is a vector of size ***n***, where it connects to a fixed value that will indicate overflow. The output **Q**, which is the same size as the input, is the count that will start at 0. The timer entity has an input signal of size ***m***, which will be the overflow value that will activate the output signal **Z**, setting it to high, so that, as it is connected to the counter enable, the counter will increase its value.

Tables 1 and 2 show that the angles have a decimal part, so to achieve this precision we can set the counter to start at 0 and overflow at 3599, which will be the resolution, and for this we will also take the decimal part of each angle as an integer. This will give us a precision of 0.1°.

For the part of the cycle from 0° to 180°, Figure 6 shows us an array of comparators, each connected to a coefficient and all connected to the counter. As can be seen in Figure 4, the signal will remain high in the intervals from α_1 to α_2, α_3 to α_4, and α_5 to α_6. So, by applying an ***AND*** operation between the corresponding comparisons, we obtain the desired intervals, to which an ***OR*** operation is applied to obtain a single output.

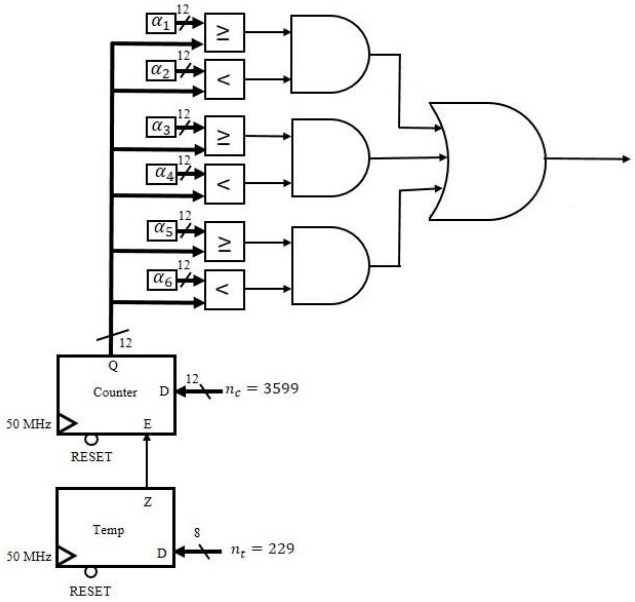

Figure 6. Design to generate waveforms from 0° to 180°.

Since the decimal part is taken as an integer for each angle, 12 bits are needed to represent it, as for the counter overflow value n_c, which will be 3599, thus providing the aforementioned precision. To calculate the timer activation, equations (6) are taken into account, now adjusting an activation period T_a equivalent to 0.1°, which will be given by the period T_o divided by the resolution, which is 3600, as shown in equations (7). Thus, it is understood that the count n_c will be equal to the **resolution – 1**.

$$\begin{aligned} T_a &= \frac{T_o}{Resolution} \\ T_a &= \frac{16.6ms}{3600} = 4.61\mu s \end{aligned} \tag{7}$$

With the system operating frequency of 50 MHz, the operating period T_{clock} is obtained, and the timer count n_t is calculated so that it activates in a time equivalent to T_a, as shown below.

$$n_t = \frac{T_a}{T_{clock}} = \frac{4.61\ \mu s}{1/50\ Mhz} = 230 \tag{8}$$

Now, for the part of the cycle from 180° to 360°, note that it is the complement of the first part of the signal, so it will now start at 1 from this point and will remain at 0 for each coefficient value. Therefore, the comparisons go from 180° to $n\alpha_1$, $n\alpha_2$ to $n\alpha_3$, $n\alpha_4$ to $n\alpha_5$, $n\alpha_6$ to 360°. See Figure 7, where the operating principle is the same as that mentioned for the first part of the waveform.

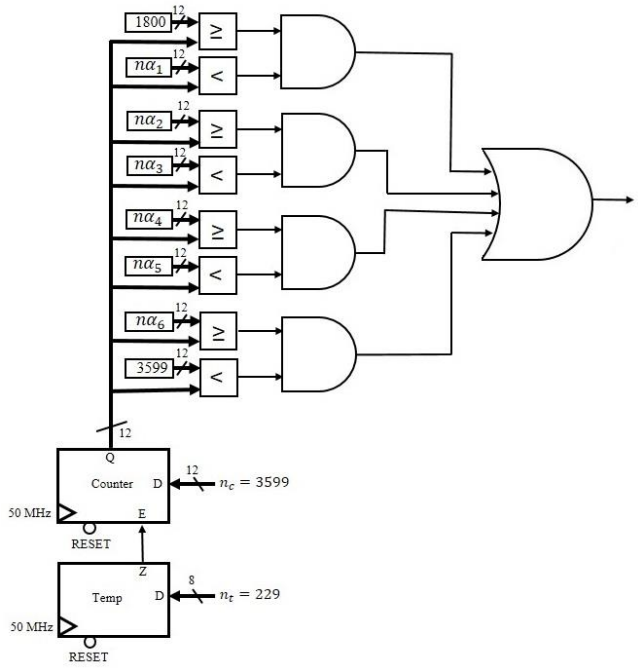

Figure 7. Design to generate waveforms from 180° to 360°.

The complete design is the arrangement of all comparisons in a single **OR** operation where the output branches, one line being the **Q** output and the other applied to a **NOT** operation to obtain **Q'**, each entering a dead time routine.

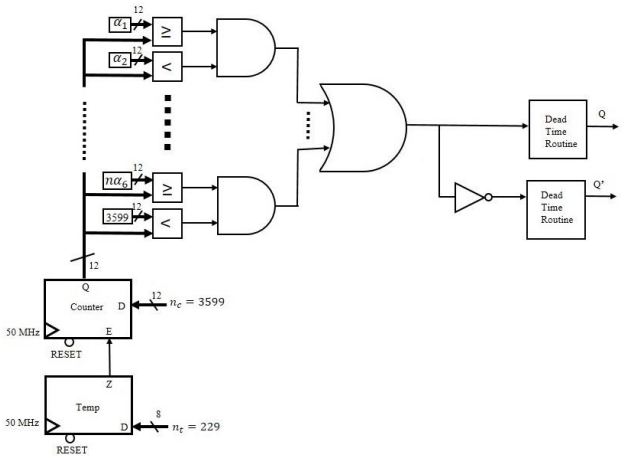

Figure 8. Complete design of SHEPWM signal generator.

IV. Results and Discussion

There are several software programs available for simulating hardware descriptions, including those offered by FPGA distributors, but for this article we will use Active-HDL software because of its clear interface for displaying signals. First, we will check that the fundamental period T_o is correct. We will check with voltage amplitudes of 50%, 70%, and 90%, which can be seen in Figures 9, 10, and 11. We can see that the period is 16.5 ms, which is quite acceptable considering the precision of the FPGA clock.

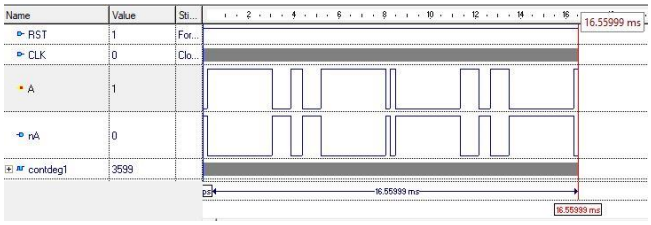

Figure 9. SHEPWM waveform with 50% amplitude.

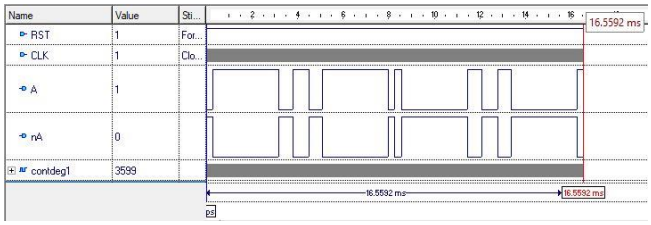

Figure 10. SHEPWM waveform with 70% amplitude.

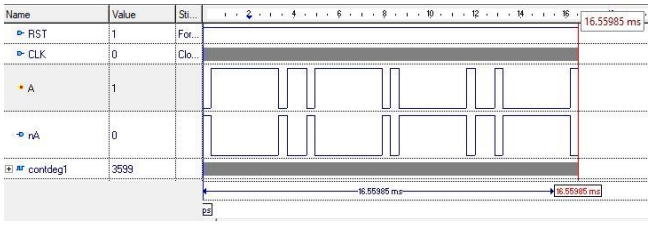

Figure 11. SHEPWM waveform with 90% amplitude.

Observing the signals in the simulation, **A** equals **Q**, **nA** equals **Q'**, **contdeg1** is the counter signal, which also helps us to know at what angle the logic level changes occur. To check the angles, we will take two samples with a signal at 90%, the first will be with angle α_3 and the second with $n\alpha_4$, so one will be within the range less than 180° and the other within the greater range. As shown in Figure 12, the **contdeg1** signal indicates 807. Remember that we are using the decimal part as an integer, so it would be equivalent to 80.7°, and in Figure 13, **contged1** indicates 2794, which would be equivalent to 279.4°. If we compare these results with those in Tables 1 and 2, we can see that there is a margin of error of 0.1°, but taking into account that the cursor shows the value of the next state change at the edges, we would have exact precision.

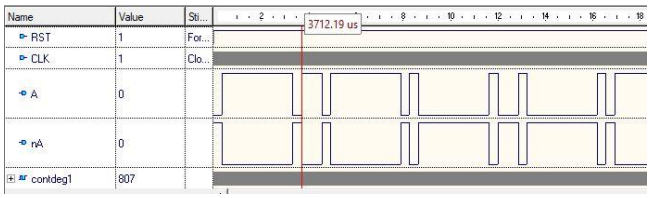

Figure 12. The cursor indicates a3 at 90% signal.

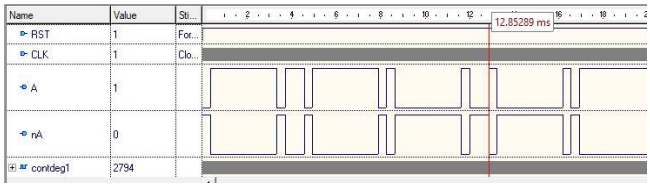

Figure 13. The cursor indicates na4 at 90% signal.

V. CONCLUSIONS

Using design and simulation tools, we can describe hardware while checking that it is working correctly. This allows us to revisit a technique used in inverters, for which specific hardware was designed with the advantage of parallel operation.

When the simulations are compared with the theoretical values, they verify that the results obtained coincide with those obtained in the calculations for the SHEPWM technique.

By using a timer-activated counter and an array of comparators, we ensure that we have a synthesizable design if we wish to implement it on an FPGA board in the future.

REFERENCES

[1] A. A. Gautam and V. Laxmi, "Fpga controlled 3phase voltage source inverter," in 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT). IEEE, 2020, pp. 522–528.

[2] 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). IEEE, 2020, pp. 1–5.

[3] Y. Luo, M. Awal, W. Yu, and I. Husain, "Fpga implementation for rapid prototyping of high performance voltage source inverters," CPSS Transactions on Power Electronics and Applications, vol. 6, no. 4, pp.320–331, 2021.

[4] T. Morales-Leal, A. Moreno-Munoz, M. A. Ortiz-Lopez, S. R. Geninatti, and F. J. Quiles-Latorre, "New random pwm method at constant switching frequency and maximum harmonic reduction created with a flexible fpga-based test bench," IEEE Access, vol. 11, pp. 19 385–19 394, 2023.

[5] B. Xu, Q. Xu, P. Guo, Y. Jia, Y. Chen, and A. Luo, "High-resolution digital pwm optimization method for critical path delay in general fpga," CPSS Transactions on Power Electronics and Applications, vol. 9, no. 2, pp. 190–206, 2024.

[6] T. Schindler and A. Dietz, "Real-time inference of neural networks on fpgas for motor control applications," in 2020 10th International Electric Drives Production Conference (EDPC). IEEE, 2020, pp. 1–6.

[7] M. Wu, S. Pan, K. Wang, G. Konstantinou, J. Pou, Y. W. Li, and K. Yang, "A discrete selective harmonic elimination formulation with common-mode voltage elimination ability," IEEE Transactions on Power Electronics, vol. 39, no. 4, pp. 3962–3967, 2023.

[8] H. S. Patel and R. G. Hoft, "Generalized techniques of harmonic elimination and voltage control in thyristor inverters: Part i–harmonic elimination," IEEE Transactions on Industry applications, no. 3, pp.310–317, 2007.

[9] ——, "Generalized techniques of harmonic elimination and voltage control in thyristor inverters: part ii—voltage control techniques," IEEE Transactions on Industry Applications, no. 5, pp. 666–673, 2007.

[10] P. N. Enjeti, P. D. Ziogas, and J. F. Lindsay, "Programmed pwm techniques to eliminate harmonics-a critical evaluation," in Conference Record of the 1988 IEEE Industry Applications Society Annual Meeting. IEEE, 1988, pp. 418–430.