Democratic and Popular Republic of Algeria

UNIVERSITYOF KASDIMERBAH-OUARGLA

Faculty of new Technologies of Information and Communication Department of Electronics and Telecommunication

Submitted in Partial Fulfillment of the Requirement

For an Academic Master Degree

DISSERTATION ACADEMIC MASTER

Domain: Electronics

Specialty: Electronics Embedded systems

Submitted by:

CHERRAD AL-Hachani CHEBBOUB Ayoub

TITLE:

design and implementation of Single-Phase inverter based on FPGA

Public ally defended

On: 06/06/2022

Before the jury:

Mr. MELHEGUEG Nacer	*MAA*	*Examiner*	*UKM Ouargla*
Mr. AOUF Anouaressadate	*MAA*	*President*	*UKM Ouargla*
Mr. BOUZIDI Mansour	*MCA*	*Supervisor*	*UKM Ouargla*

Academic Year: 2021/2022

Dedication

First of all, I thank Almighty God for gave me health and courage and patience and especially the will during all these years of study.

I extend my deepest thanks to me the honorable family ,and supervisor Mr. BOUZIDI Mansour as well as to all teachers Department of Electronics and Telecommunication

Acknowledgement

We want absolutely to express our thanks
and deep gratitude
to our supervisor Dr. BOUZIDI Mansour,
for his consistent support, orientation, advices
and encouragement during the
accomplishment of this project.
we would like to thank all the people who
contributed in any way to the work
described in this thesis.

Contents

List of Abbreviations

AC	Alternating Current
DC	Direct Current
RESR	renewable energy sources
UPSU	uninterruptible power supply
VSIV	voltage source inverters
CSIC	current source inverters
H-bridge	half-bridge
VHDL	Very-High-speed integrated Hardware Description Language
CLK	Clock
FPGA	Field Programmable Gate Array
GNDG	Ground
SPWM	Sinusoidal-Pulse Width Modulation
THD	Total Harmonic Distortion
PLD	programmable logic devices
PROM	programmable read-only memories
RAM	Random Access Memory
SRAM	Static Random Access Memory
AP	All Programmable
PS	Processing System
PL	Programmable Logic
SPWM	Space Vector Pulse Width Modulation
IGBT	Insulated Gate Bipolar Transistor
THD	total harmonic distortion

List of Symbols

Vo	output voltage
P_0	output power
v_o	output voltage
i_o	output current
S_1 ,S_2 ,S_3 ,S_4 , D_1 ,D_2 ,D_3 ,D_4	Switch
V_m	amplitude of the reference or control signal
T	period of the reference voltage.
V_p	amplitude of the carrier signal
$\boldsymbol{v}^*$	Amplitude
$\boldsymbol{v_p}$	period of the carrier signal.
T_p	period of the carrier signal.
$\boldsymbol{f_p}$	carrier frequency
t_{pHL}	Switching speed

List of Figures

List of tables

General introduction

The applications of electronics were for a long time limited to high frequency techniques.
The possibilities of application were limited by the unreliability of the electronic elements available.
This reliability was insufficient in front of the requirements required by the new applications in the industrial field.
It was only as a result of the development of special electronic components of higher reliability and tolerance.
more restricted, that new techniques can be envisaged, thus a new branch of electronics called power electronics [1] is born.
Currently, this discipline of electrical engineering affects vast and very diverse fields of application for powers covering a wide range (from a few watts to several hundred megawatts).
Voltage inverters can be driven in several control types.

Full-wave control is the basic technique used for single-phase inverters
In this case, the control signal and the desired voltage at the output have the same frequency.
However, the latter has power limitations and a large number of annoying harmonics.
The drawbacks of these basic commands have prompted research to explore new strategies to improve inverter performance. The most advanced technique is pulse width modulation.
This strategy consists in controlling the inverter at high frequency, it thus makes it possible to vary both the frequency and the amplitude while keeping the DC source constant.
However, the implementation of these strategies requires very costly and cumbersome installations and circuits.
To remedy this, the digital solution remains the most appropriate.
Indeed, the development of digital processors has boosted research work and shortened the time it takes to complete these orders.
The objective of this work is twofold. On the one hand, we want to make a comparative study of the different control strategies of single-phase inverters, and on the other hand, to experiment with the technological aspect of a digital control on the FPGA board.
To achieve this goal, we proceeded as follows:
In the first chapter, we briefly recall the operating principle of a single-phase and three-phase voltage inverter, as well as the different types of inverters and their

application areas.
The second chapter highlights the different inverter control strategies with a detailed study of basic techniques such as full-wave control, offset control, and advanced PWM techniques such as natural PWM,

the simulation of the different control techniques presented and the presentation of the simulat*ion results .*
The last chapter is a study of the different elements used for the transition from analog to digital control and the implementation of the latter on the FPGA board.

Chapter I

Overview of DC to AC converters

I.1. Introduction

The dc-ac converter, also known as the inverter, converts dc power to ac power at desired output voltage and frequency. The dc power input to the inverter is obtained from an existing power supply network or from a rotating alternator through a rectifier or a battery, fuel cell, photovoltaic array or magneto hydrodynamic generator. The filter capacitor across the input terminals of the inverter provides a constant dc link voltage. The inverter therefore is an adjustable-frequency voltage source. The configuration of ac to dc converter and dc to ac inverter is called a dc-link converter.

Inverters are used in the following applications:

- Ac motor drive systems;
- Electric power transmission systems;
- Renewable energy sources (RES);
- Active filters used for harmonic suppression and electric power quality;

improvement

- Active power factor correction systems;
- Uninterruptible power supply (UPS) systems;

Inverters depending on their type of input source are categorized as[2]:

a) Voltage source inverties (VSI)

b) Current source inverters (CSI)

Furthermore, they can be subdivided as:

i) Single-phase half-bridge inverters

ii) Single-phase full-bridge or H-bridge inverters

iii) Three-phase full-bridge inverters.

Fig. (1.1) shows the various topologies of two-level inverters, which are used today in the applications mentioned above. They are called two-level because their output phase voltages, as will be shown later on, are synthesized using only two different voltage

values $V_{in}/2$ and $V_{in}/2$. In this chapter, various topologies of inverters will be presented and analyzed under ideal operating conditions[3].

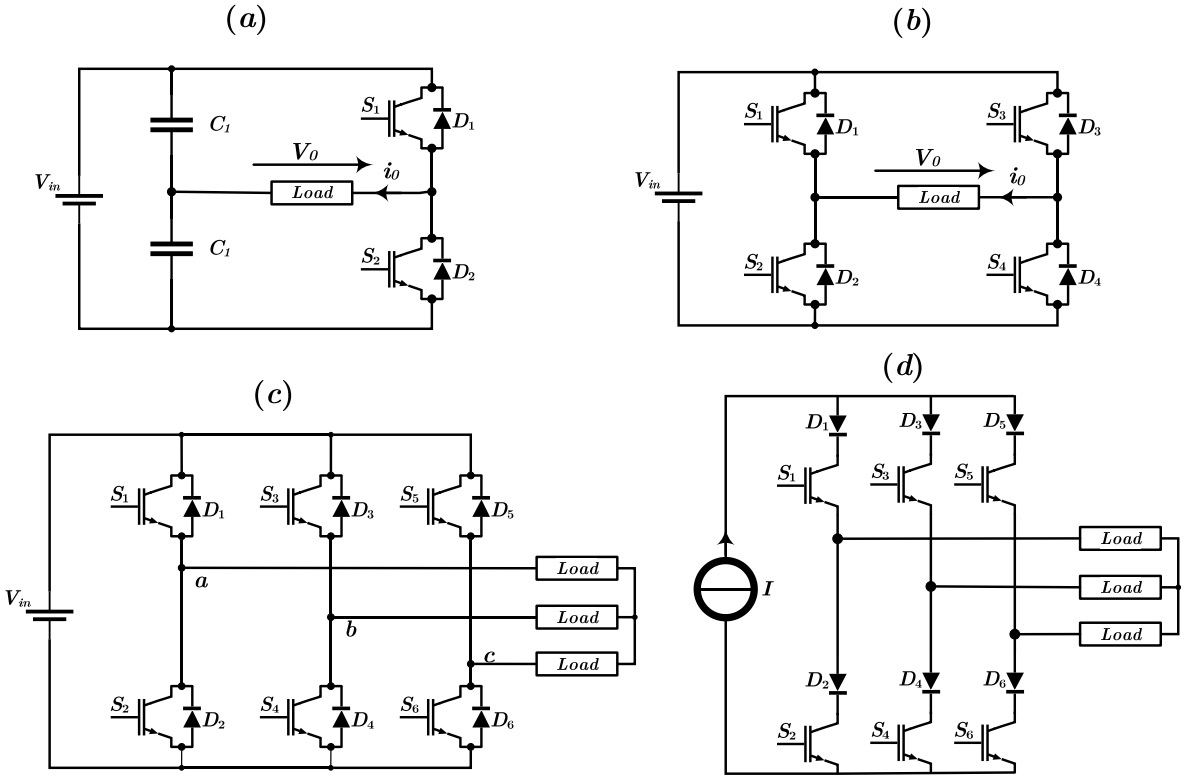


Figure (I.1) : Various topologies of two-level inverters. (a) Single-phase half-bridge voltage source inverter (VSI); (b) single-phase full-bridge VSI; (c) three-phase VSI; (d) three-phase current source inverter.

I.2. Single-Phase Half-Bridge Inverter

Fig. (1.2) shows the power circuit and the generated output voltage of a single-phase half-bridge inverter. The choice of the semiconductor switches for implementing this inverter depends on its power and switching requirements. The freewheeling diodes across the switches are included in the same power module as the switches. The output voltage v_o is a two-level square-wave pulse ($V_{in}/2$ and -$V_{in}/2$)[4].[5].

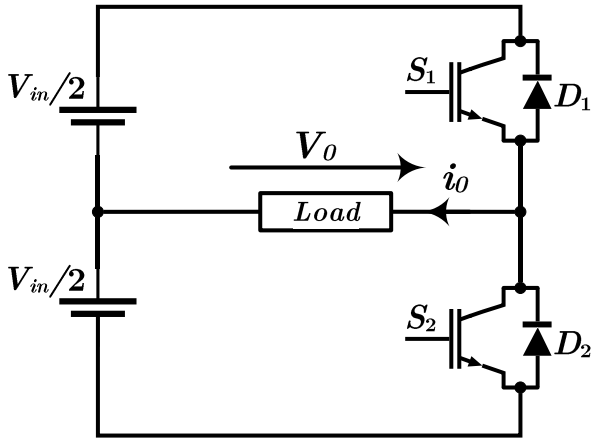


Figure (I.2) : Single-phase half-bridge inverter.

When S_1 or D_1 is conducting, the output voltage is $v_o = V_{in}/2$. When S_2 or D_2 is conducting, the output voltage is $v_o = -V_{in}/2$. The output voltage control can be achieved by varying the pulse widths of the output voltage through the gating signals of the semiconductor switches. When an increase is required in the output voltage, then the conduction time intervals of S_1 and S_2 are increased. When the output voltage is required to be decreased, then the conduction time intervals of S_1 and S_2 are decreased. Therefore, if the output voltage v_o is required to be kept constant, due to the dc input voltage variations or voltage drops within the power circuit, this is achieved through the pulse width variations of the gating signals and, consequently, the output voltage pulse widths.

The freewheeling diodes of the inverter are needed to provide current bidirectionality to the semiconductor switches. When the current is positive, it flows through the semiconductor switch and when it is negative, it flows through the respective antiparallel diode or the so-called freewheeling diode. The freewheeling diodes provide flexibility to the inverter not only to deliver power from the input dc source to the load but also to deliver power from the load to the input dc source. Moreover, the freewheeling diodes are needed to provide a circuit path for the discharging of the load inductances otherwise voltage spikes will appear across the semiconductor switches that will result in their failure.

In case of resistive load, when both switches are not conducting, then during these

intervals the output voltage is zero.

In case of inductive load, when both switches are not conducting, the load current continues to flow through the freewheeling diode D_1 or D_2.

The half-bridge single-phase inverter when operating with inductive resistive load has the following five states of operation:

i)*State 1*($v_o = V_{in}/2$, $i_o > 0$): During this state, only the power semiconductors witch S_1 is conducting, creating an output voltage equal to $V_{in}/2$. During this state the output currents positive. The conduction path of this state is shown in Fig. (I.3)(b). The active power is delivered from the source to the load ($P>0$).

ii)*State 2*($v_o = -V_{in}/2$, $i_o < 0$):During this state, only the power semiconductors witch S2 is conducting, creating an output voltage equal to $-V_{in}/2$. During this state the output current is negative. The conduction path of this state is shown in Fig. (I.3)(c). The active power is delivered from the source to the load ($P>0$).

iii) *State 3* ($v_o = V_{in}/2$, $i_o < 0$): During this state only the freewheeling diode D_1 is conducting, creating an output voltage equal to $+V_{in}/2$. During this state the output current is negative. The conduction path of this state is shown in Fig. (1.3)(d). The active power is delivered from the load to the source ($P<0$). This switching state is also called regenerative state.

iv)*State 4* ($v_o = -V_{in}/2$, $i_o > 0$): During this state only the freewheeling diode D_2 is conducting, creating an output voltage equal to $-V_{in}/2$. During this state the output current is positive. The conduction path of this state is shown in Fig. (I.3)(e). The active power is delivered from the load to source ($P<0$). This switching state is also called regenerative state.

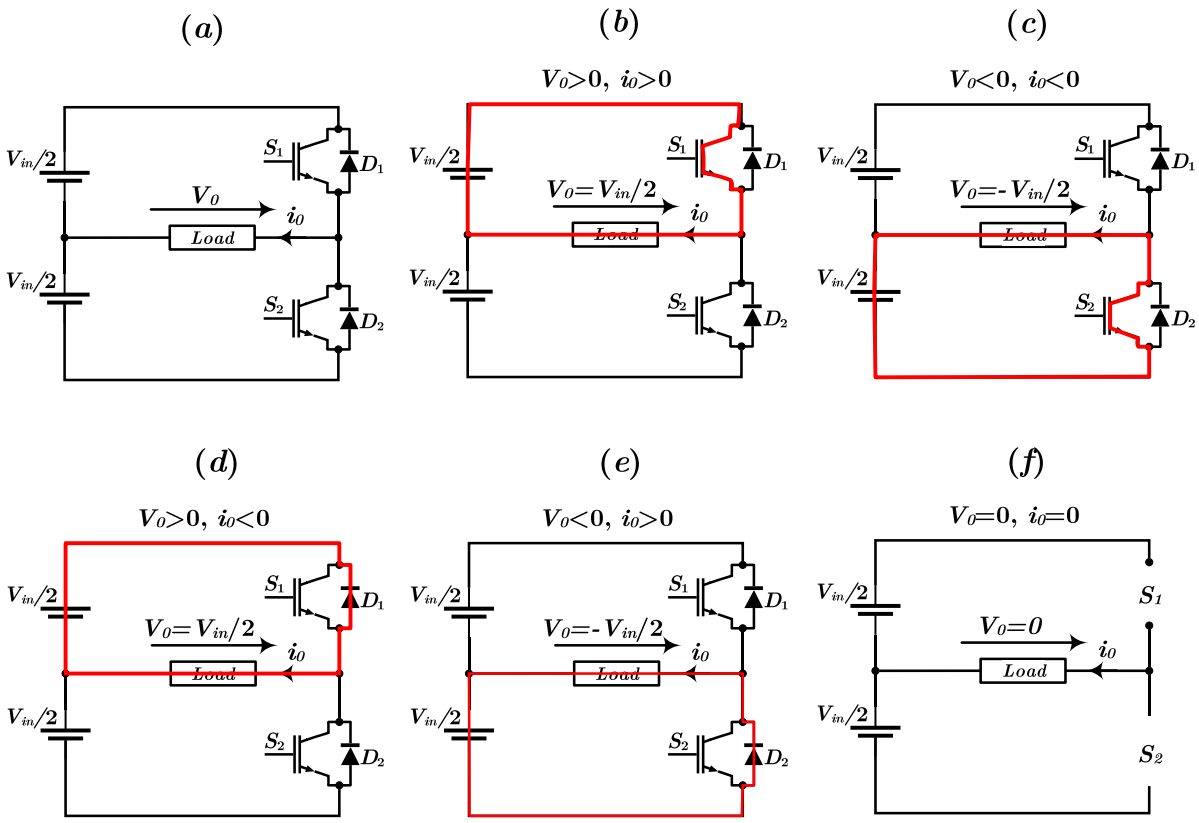


Figure (I.3) : Possible states of operation and their respective conduction paths of single-phase half-bridge inverter. (a) Equivalent circuit of half-bridge inverter when the input voltage across the capacitors is pure dc; (b) State 1; (c) State 2; (d) State 3; (e) State 4; (f) State 5.

v)***State 5*** (v_o **=0,** i_o**= 0**): During this state none of the semiconductor devices is conducting, creating zero load current and under certain loads zero load voltage. The equivalent circuit of the inverter during this state of operation is shown in Fig. (1.3)(f). The switching states of operation of the single-phase half-bridge inverter are summarized in Table (I.1)[5].

Table (I.1) : Possible states of operation of the single-phase half-bridge square-wave inverter, their respective output voltage level, and instant output active power

Inverter state	Switching states				Output voltage	Output current	Output power
	S1	S2	D1	D2			
1	1	0	0	0	$V_{in}/2$	$i_0>0$	$P>0$, Power is delivered from the dc source to the load
2	0	1	0	0	$-V_{in}/2$	$i_0<0$	$P>0$, Power is delivered from the dc source to the load
3	0	0	1	0	$V_{in}/2$	$i_0>0$	$P<0$, Power delivered from the load to the dc source
4	0	0	0	1	$-V_{in}/2$	$i_0<0$	$P<0$, Power delivered from the load to the dc source
5	0	0	0	0	0	0	$P=0$

I.3. Single-Phase Full-Bridge Inverter

The half-bridge inverter of Fig. (I.1) (a) exhibits the following three major disadvantages [1].[2].[3].[6]:

i) Two electrolytic capacitors connected in series are needed at the dc input side.

ii) It is unable to generate zero output voltage intervals for non-resistive loads.

iii) The amplitude of the output voltage pulses is half of the dc input voltage.

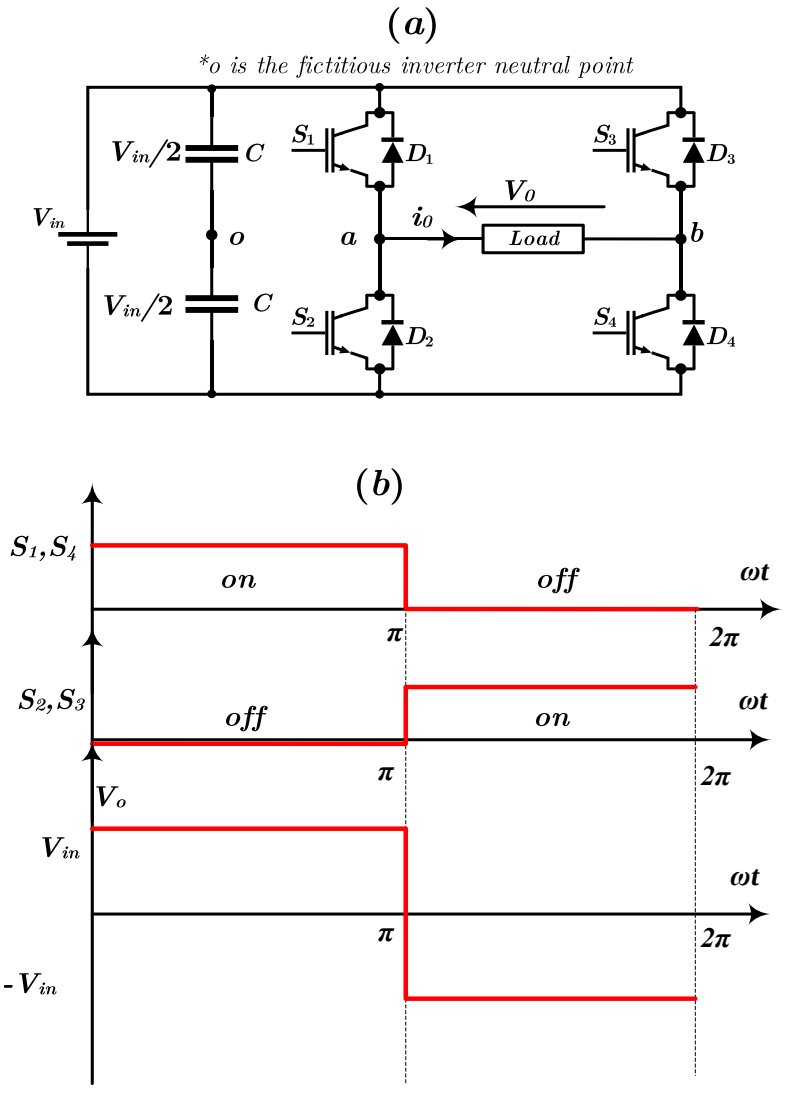


Figure (I.4): Single-phase full-bridge inverter with insulated gate bipolar transistors. (a)Power circuit; (b) inverter waveforms for square-wave output voltage generation

Fig. (1.4) (a) shows the power circuit of the single-phase full-bridge inverter (composed of two half-bridge inverters), which does not present any of the drawbacks mentioned above. Fig. (1.4) (b) shows the gating pulses of the power semiconductors witches and the generated square-wave output voltage. Furthermore, Fig. (I.5) shows the gating signals and the generated quasi square-wave output voltage (pulse width =120°). Moreover, Fig. (I.5) shows the conducting power semiconductor devices for different types of inductive loads.

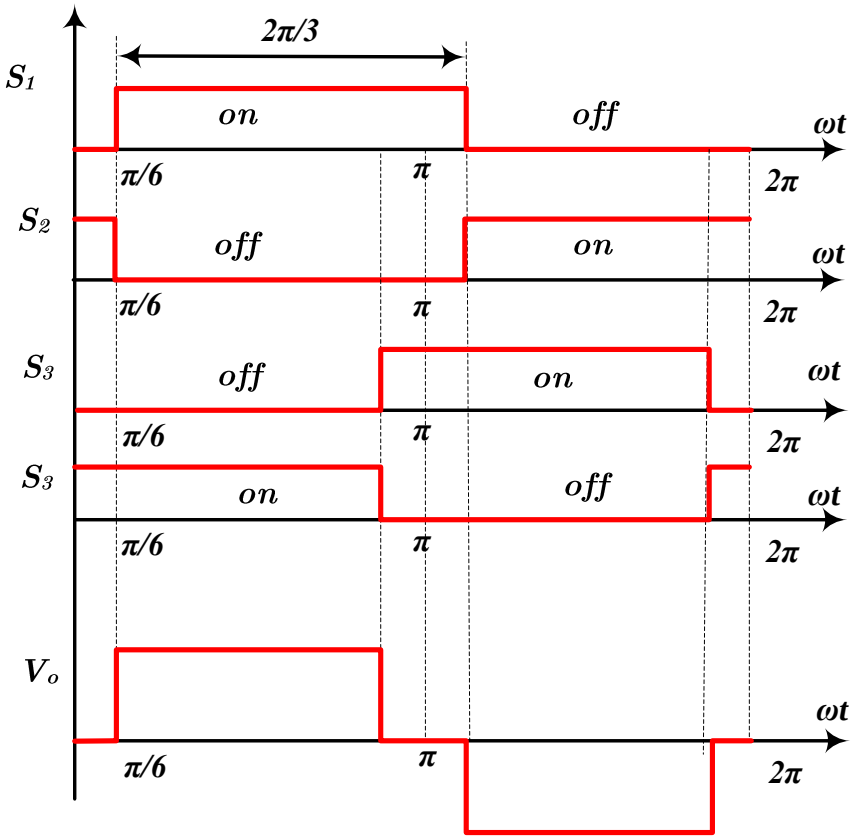


Figure (I.5): Single-phase full-bridge quasi square-wave output voltage generation

Examining the power circuit of Fig. (I.4) (a) and the waveforms of Fig. (I.5), it can be seen that the operation of the single phase full-bridge inverter can be divided into the following six states:

i) State 1 ($v_o = V_{in}$, $i_o > 0$): During this state the semiconductor switches S_1 and $S_4$are conducting, creating across the load an output voltage equal to V_{in} and, at the same time, positive current flows through the load. The conduction path of this mode is shown in Fig. (I.6) (a). Since P> 0, active power is delivered from the source to the load.

ii) State 2 ($v_o = -V_{in}$, $i_o < 0$): During this state the semiconductor switches S_2 and S_3 are conducting, creating across the load an output voltage equal to $-V_{in}$ and at the same time negative current flows through the load. The conduction path of this state is shown in Fig. (I.6) (b). Since P> 0, active power is delivered from the source to the load.

iii) State 3 ($v_o = 0$, $i_o > 0$): This state is created by two different operations. The first operations achieved through the conduction of D_2, S_4 and the second through the conduction of S_1, D_3. During this state, which is called freewheeling state, the output voltage is zero and the load current is positive. The two possible conduction paths of this state are shown in Fig. (I.6) (c) and (d). Since P=0, there is no power delivered from the source to the load or vice versa.

iv) State 4 ($v_o = 0$, $i_o < 0$): This state is created in two different ways. The first way is achieved through the conduction of the semiconductor devices D_1, S_3 and the second through the conduction of S_2, D_4. During this mode, which is called freewheeling mode, the output voltage is zero and the load current is negative. The two possible conduction paths of this state are shown in Fig. (I.6) (e) and (f). Since P=0, there is no power delivered from the source to the load or vice versa.

v) State 5 ($v_o = V_{in}$, $i_o < 0$): During this state the freewheeling diodes D_1 and D_4 are conducting, creating across the load an output voltage equal to V_{in} and at the same time negative current flows through the load. The conduction path of this state is shown in Fig. (1.6) (g). Since P< 0, active power is delivered from the load to the source. This state is a regenerative state.

iv) State 6 ($v_o = -V_{in}$, $i_o > 0$): During this state the freewheeling diodes D_2 and $D_3$are conducting, creating across the load an output voltage equal to $-V_{in}$ and at the same time positive current flows through the load. The conduction path of this state is shown in Fig. (1.6)(h).Since P< 0, active power is delivered from the load to the source. This state is a regenerative state.

The operating states of a power converter can be represented by the possible switching states of the semiconductor switches. The possible switching states of operation of the single-phase full-bridge two-level inverter are summarized in Table (1.2).

Table (I.2) : Possible states of operation of the single-phase full-bridge two-level inverter, their respective output voltage, and output instant active power

Inverter state	Switching states								Output voltage v_o Output current i_o	Output power P_o
	S_1	S_2	S_3	S_4	D_1	D_2	D_3	D_4		
1	1	0	0	1	0	0	0	0	$v_o = V_{in}$, $i_o > 0$	$P_o > 0$
2	0	1	1	0	0	0	0	0	$v_o = -V_{in}$, $i_o < 0$	$P_o > 0$
3	0	0	0	1	0	1	0	0	$v_o = 0$, $i_o > 0$	$P_o = 0$
	1	0	0	0	0	0	1	0		
4	0	0	1	0	1	0	0	0	$v_o = 0$, $i_o < 0$	$P_o = 0$
	0	1	0	0	0	0	0	1		
5	0	0	0	0	1	0	0	1	$v_o = V_{in}$, $i_o < 0$	$P_o < 0$
6	0	0	0	0	0	1	1	0	$v_o = -V_{in}$, $i_o > 0$	$P_o < 0$

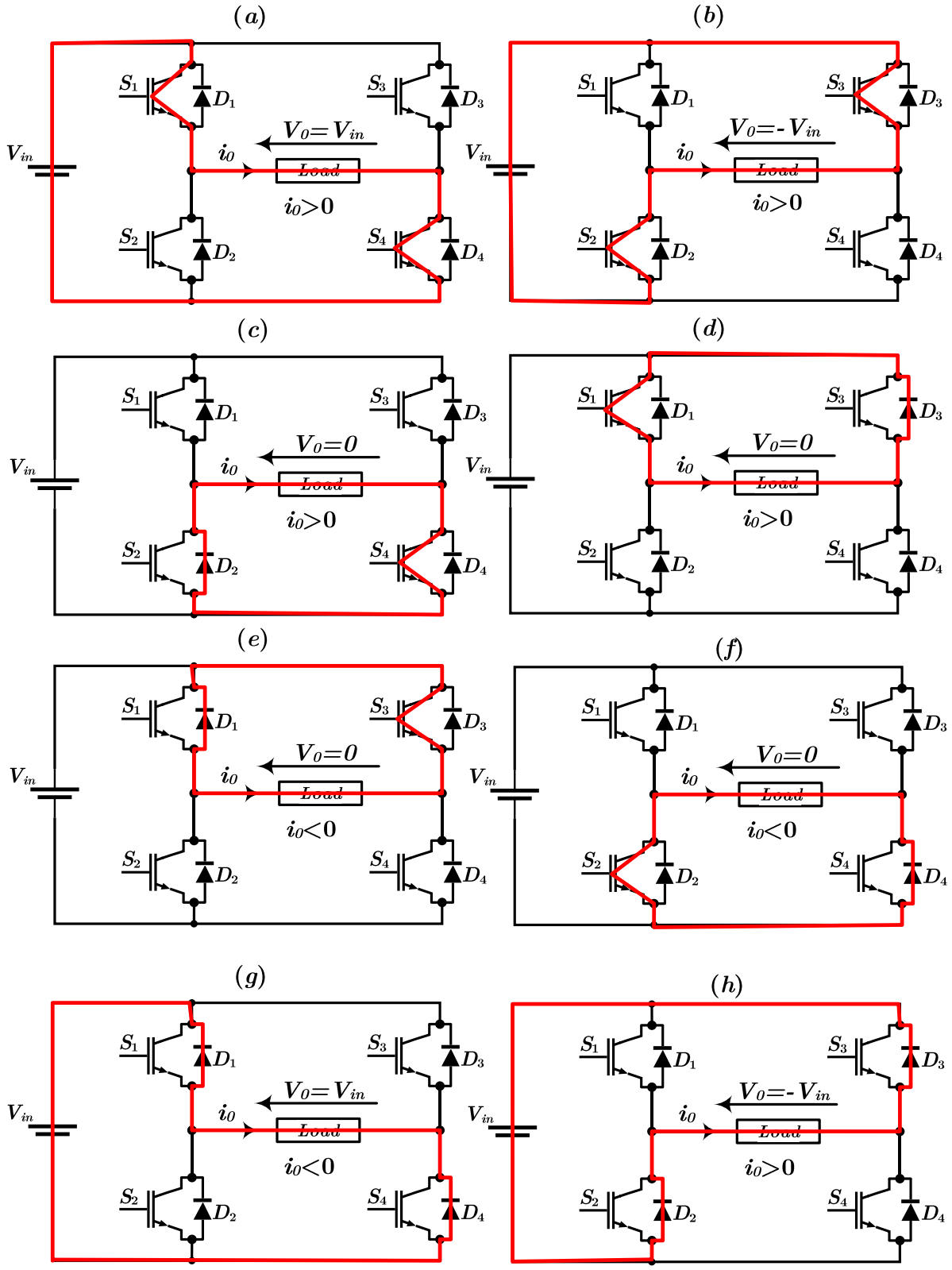


Figure (I.6) : Possible states of operation and their respective conduction paths of a single-phase full-bridge inverter depending on the required output voltage and current.(a) State 1; (b) State 2; (c) State 3 possible path 1; (d) State 3 possible path 2; (e) State 4possible path 1; (f) State 4 possible path 2; (g) State 5; (h) State 6.

I.4. Conclusion

In this chapter, some topologies of two-level AC/DC converter have been presented, their power circuits, their types as well as their operation principal have been shown. In this chapter also, the operation principal and all possible switching states of both single-phase half/full bridge topologies have been presented in detail.

Next chapter will be devoted on modulation strategies for full-bridge single phase inverter.

Chapter II

Pulse width modulation techniques for single-phase inverters

II.1. Introduction

The fundamental magnitude of the output voltage from an inverter can be controlled to be constant by exercising control within the inverter itself that is no external control circuitry is required. The most efficient method of doing this is by Pulse Width Modulation (PWM) control used within the inverter. In this scheme the inverter is fed by a fixed input voltage and a controlled ac voltage is obtained by ad justing the on and the off periods of the inverter components. The advantages of the PWM control scheme are [7]:

- The output voltage control can be obtained without addition of any external components.
- PWM minimizes the lower order harmonics, while the higher order harmonics can be eliminated using a filter.

The disadvantage possessed by this scheme is that the switching devices used in the inverter are expensive as they must possess low turn on and turn off times, nevertheless PWM operated are very popular in all industrial equipment. PWM techniques are characterized by constant amplitude pulses with different duty cycles for each period. The width of these pulses is modulated to obtain inverter output voltage control and to reduce its harmonic content. There are different PWM techniques which essentially differ in the harmonic content of their respective output voltages, thus the choice of a particular PWM technique depends on the permissible harmonic content in the inverter output voltage.

II.2. Sinusoidal-Pulse Width Modulation (SPWM)

The sinusoidal PWM (SPWM) method is very popular in industrial applications and is extensively reviewed in the literature [1-2]. two PWM switching schemes are discussed in this chapter, which improve the characteristics of the inverter. In Figure (II .1), the top

devices are assigned to be S_{11} and S_{21} while the bottom devices as S_{12} and S_{22}.[7].[8].

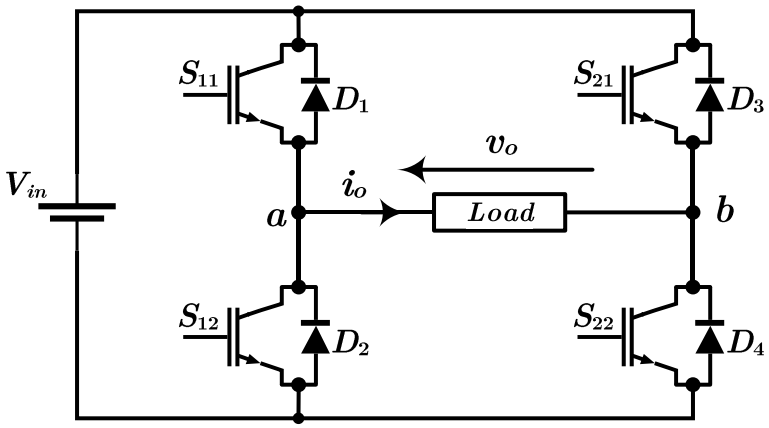


Figure (II.1): Schematic of a Single Phase Full-Bridge Inverter.

II.2.*1. SPWM With Bipolar Switching*

This control technique is based on comparing a reference signal (voltage v^*) with a carrier signal v_p. The reference voltage and the carrier signal are characterized by the following parameters [9].[10].[11]:

$$v^*(t) = V_m \sin(2\pi ft) \tag{2.1}$$

V_m: amplitude of the reference or control signal,

T=1/f: period of the reference voltage

V_p: amplitude of the carrier signal

T_p=1/f_p: period of the carrier signal

$m_a = V_m / V_p$: amplitude modulation factor

m_f=f_p/f : frequency modulation factor

In this scheme the diagonally opposite transistors S_{11}, S_{22} and S_{21} and S_{12} are turned on or turned off at the same time. The output of leg A is equal and opposite to the output of leg B. The output voltage is determined by comparing the reference voltage $\boldsymbol{v}^*$, and the triangular signal $\boldsymbol{v_p}$ as shown in Figure (II .2) (a) to get the switching pulses for the

devices, and the switching pattern is as follows:

$$v^* > v_p \;\; S_{11} \text{ and } S_{22} \text{ ON}, \; v_o = V_{in} \tag{2.2}$$

$$v^* \leq v_p \;\; S_{12} \text{ and } S_{21} \text{ ON}, \; v_o = -V_{in} \tag{2.3}$$

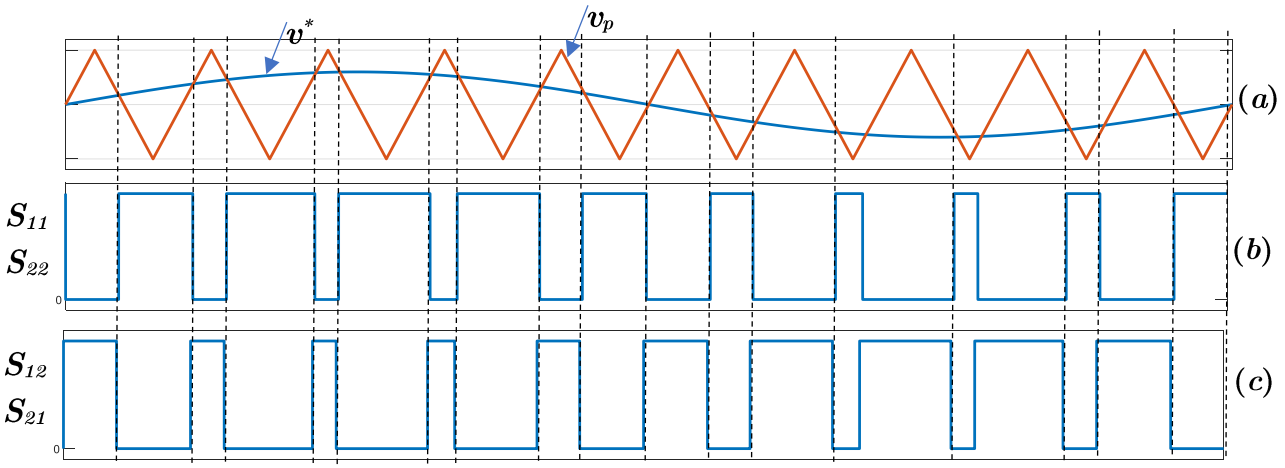


Figure (II .2):Bipolar PWM (a) Sine-triangle comparison (b) Switching pulses for S_{11}/S_{22} (c) Switching pulses for S_{12}/S_{21}

II.*2.2. SPWM With Unipolar Switching*

In this scheme, the devices in one leg are turned on or off based on the comparison of the modulation signal $\boldsymbol{v}^*$ with a high frequency triangular wave. The devices in the other leg are turned on or off by the comparison of the modulation signal $\boldsymbol{-v}^*$with the same high frequency triangular wave. Figure (II .3) shows the unipolar scheme for a single-phase full bridge inverter, with the modulation signals for both legs and the associated comparison to yield switching pulses for both the legs [7]. [9].[12].[13] .

The logic behind the switching of the devices in the leg connected to 'a' is given as:

$$\begin{aligned} v^* > v_p &: S_{11} \text{ ON} \\ v^* \leq v_p &: S_{11} \text{ OFF} \end{aligned} \tag{2.4}$$

and that in the leg connected to 'b' is given as:

$$\begin{aligned} -v^{*} > v_{p}&: S_{21} \text{ ON} \\ -v^{*} \leq v_{p}&: S_{21} \text{ OFF} \end{aligned} \tag{2.5}$$

Figure (II .3) shows the sine triangle comparison, the switching pulses for S_{11} and S_{21} are shown. The switching for the other two devices is obtained as:

$$\begin{aligned} S_{12} &= \overline{S}_{11} \\ S_{22} &= \overline{S}_{21} \end{aligned} \tag{2.6}$$

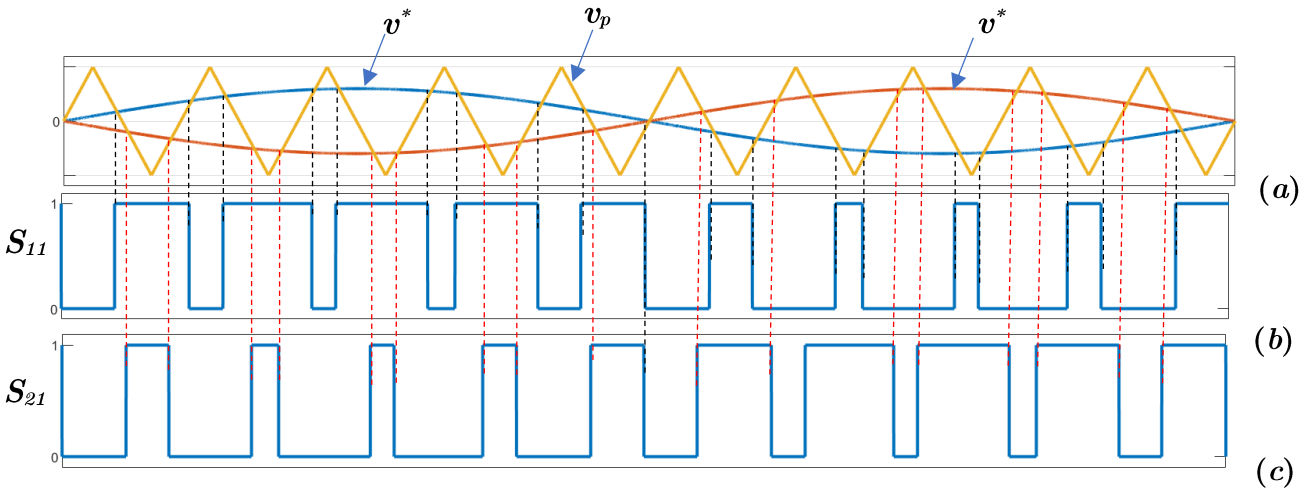


Figure (II .3): Unipolar PWM voltage switching scheme (a) Sine triangle comparison (b) switching pulses for S_{11} (c) switching pulses for S_{21}.

II.3. Simulation results

In this section, simulation results based on MATLAB Simulink are carried-out to validate the presented PWM techniques for full-bridge single phase inverter. The DC voltage at the input is fixed at V_{in}=20V, the frequency of the reference voltage at the output is f=50Hz, the amplitude modulation factor is m_a=0.8, which means that the amplitude of the reference voltage is $V_{in} * m_a = 16$V. The carrier frequency is f_p=5 kHz, which

means that the frequency modulation factor is $m_f = f_p / f = 100$. The single-phase inverter feeds an inductive load L=9mH, R=50Ω.

Figure (II .4): shows the output voltage and load current obtained using the bipolar PWM scheme. It can be seen that the output voltage takes two values of +20 and -20, the load current is sinusoidal with a considerable ripple. As shown in Figure II.5, The THD of load current is 9.21%.

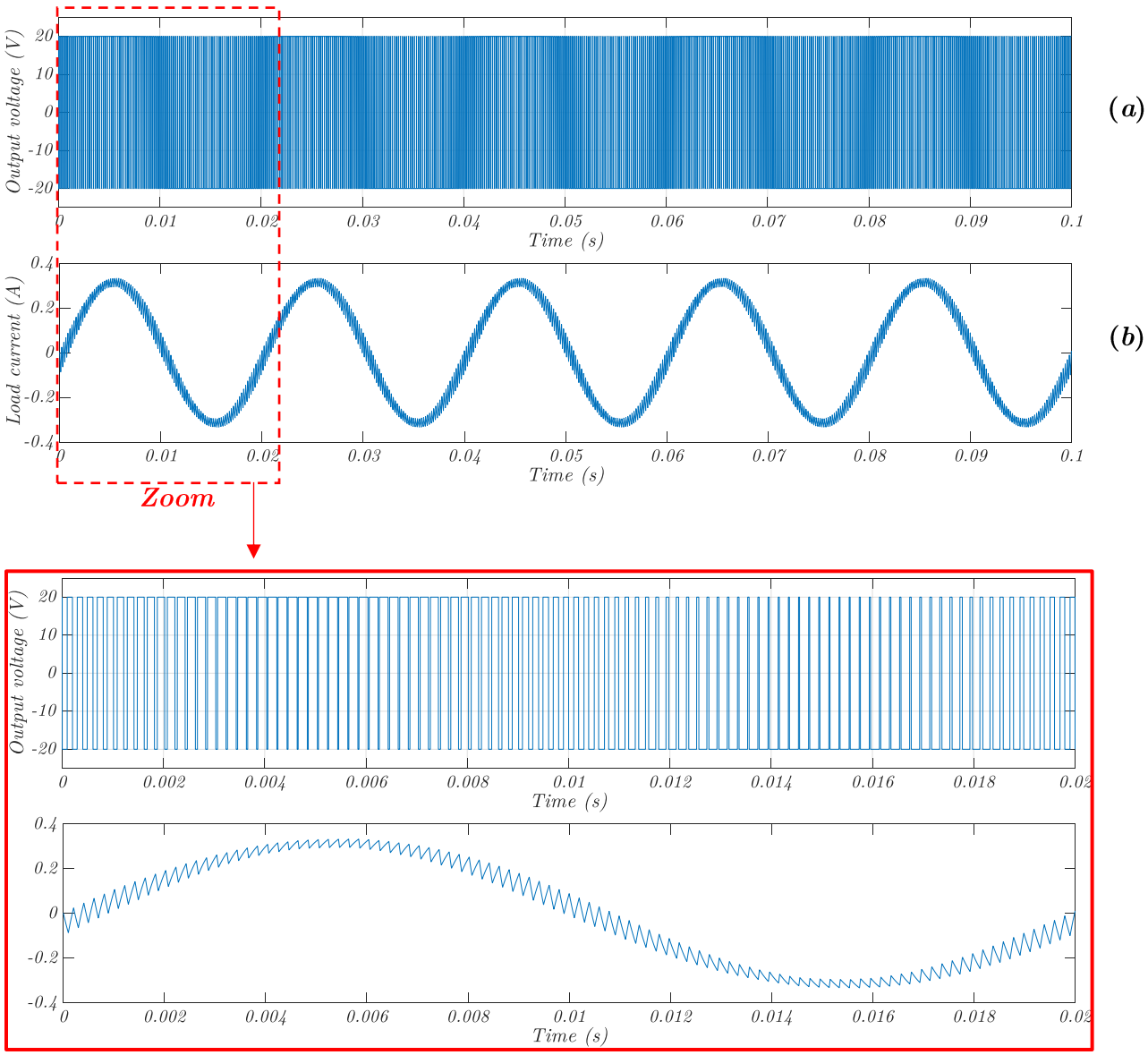


Figure (II .4): Bipolar PWM scheme (a) Output voltage v_o (b) Load current i_o .

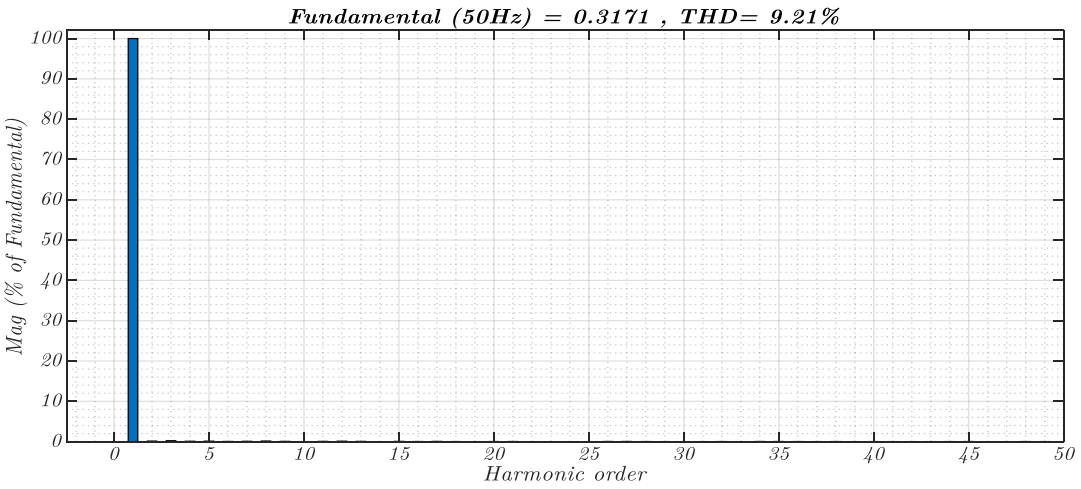
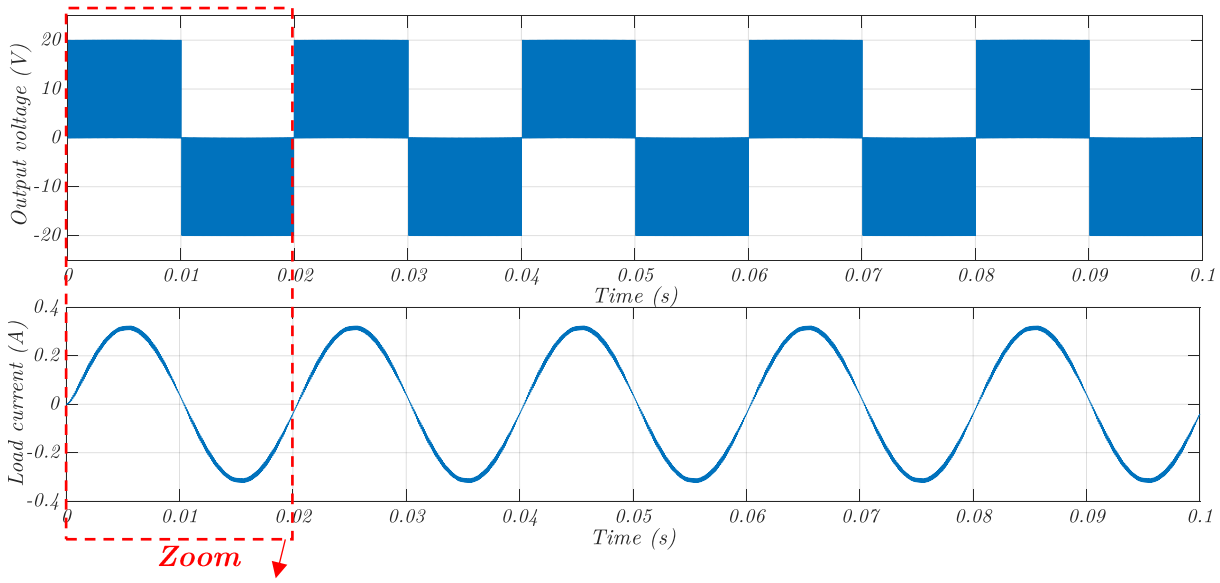


Figure (II .5):Harmonic Spectrum of the load current using the bipolar PWM technique



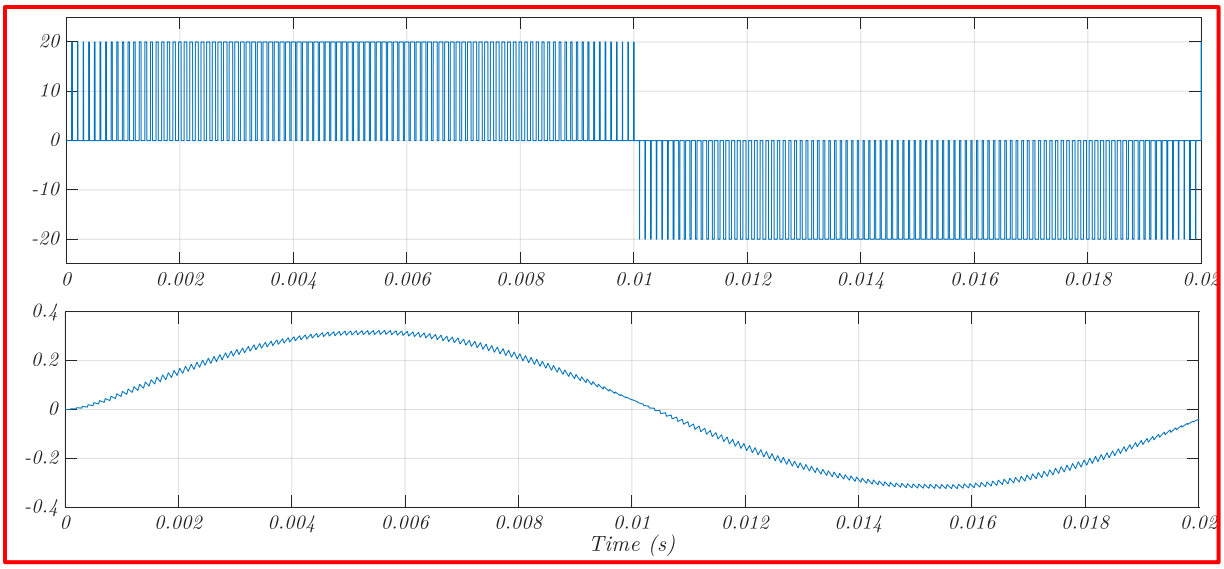


Figure (II .6)**:** Unipolar PWM voltage switching scheme (a) output voltage v_o (b) load current i_o

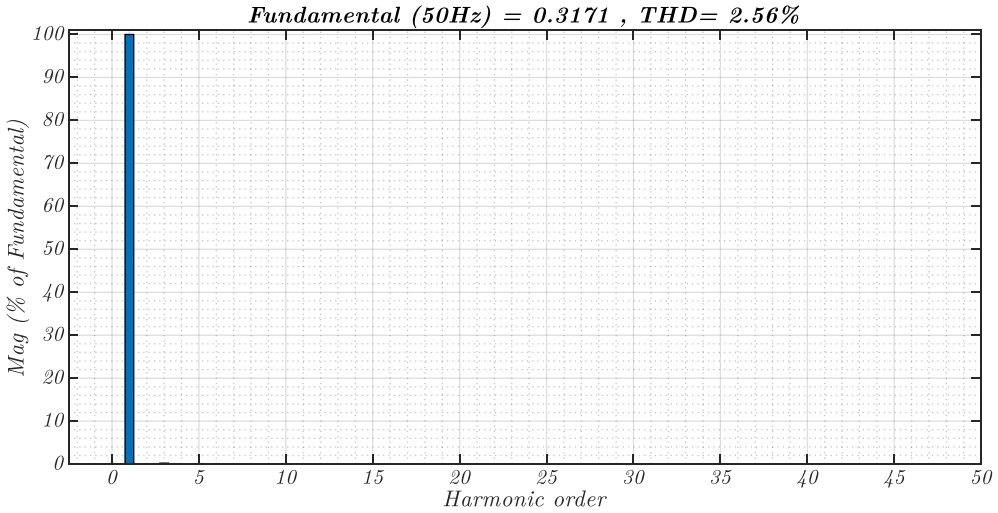


Figure (II .7): Harmonic Spectrum of the load current using the unipolar PWM technique

Figure (II.6) shows the output voltage and load current obtained using the unipolar PWM scheme. It can be observed that the output voltage takes three values of +20, 0 and -20, the load current is sinusoidal with a small ripple, the THD of load current is 2.56% as shown in Figure (II.7), which is better than that obtained using bipolar PWM technique.

Table (II.1). shows a comparative investigation between bipolar and unipolar SPWM techniques for single-phase full-bridge inverter. The unipolar SPWM requires two reference signals while the bipolar SVPWM needs only one reference signal, however, both techniques require one carrier signal.

In the other side, the quality of the output voltage and load current is better when the unipolar SPWM technique is used, the THD of the load current and output voltage is 2.56% and 77.14% respectively which are less than those obtained using bipolar SPWM technique (9.21% and 146.07% for the current and voltage respectively).

Table (II.1). Comparative study between bipolar and unipolar SPWM techniques for single-phase full-bridge inverter.

	SPWM unipolar	***SPWM bipolar***
Total reference signal number	2	1
Peak voltage amplitude	16V	16V
Total carrier signal number	1	1
Load current THD	2.56%	9.21%
Output voltage THD	77.14%	146.07%

II.4. Conclusion

In this chapter, two SPWM techniques have been presented and analyzed for single-phase-full H-bridge inverter (bipolar and unipolar SVPWM schemes). Their algorithms and operation principals are shown in detail. Simulation results shows that the unipolar SVPWM give better results compared to the bipolar SPWM technique in term of the quality of the output voltage and load current.

The next chapter will be dedicated to the real implementation of these control techniques using FPGA.

Chapter III

Real time implementation of SPWM techniques for single-phase inverter

III.1. Introduction

This chapter is reserved for the experimental validation of the SPWM techniques for the single-phase full H-bridge inverter.

Firstly, the hardware power circuit of the inverter will be presented in detail, after that, we will present how to synthesize the SPWM algorithms using the Xilinx VIVADO 2018.3 software. Finally, the experimental results of both unipolar and bipolar SPWM techniques for the single-phase inverter will be provided.

III.2. Introduction FPGA

The field-programmable gate array (FPGA) is an integrated circuit that consists of internal hardware blocks with user-programmable interconnects to customize operation for a specific application. The interconnects can readily be reprogrammed, allowing an FPGA to accommodate changes to a design or even support a new application during the lifetime of the part [14].

The FPGA has its roots in earlier devices such as programmable read-only memories (PROMs) and programmable logic devices (PLDs). These devices could be programmed either at the factory or in the field, but they used fuse technology (hence, the expression "burning a PROM") and could not be changed once programmed. In contrast, FPGA stores its configuration information in a re-programmable medium such as static RAM (SRAM) or flash memory. FPGA manufacturers include Intel, Xilinx, Lattice Semiconductor, Microchip Technology and Microsemi [15].

III.2.1. The map ZedBoard Zynq-7000:

The ZedBoard is an evaluation and development board based on the Xilinx ZynqTM-7000 All Programmable SoC (AP SoC). Combining a dual Corex-A9 Processing System (PS) with 85,000 Series-7 Programmable Logic (PL) cells, the Zynq-7000 AP SoC can be targeted for broad use in many applications. The ZedBoard's robust mix of on-board

peripherals and expansion capabilities make it an ideal platform for both novice and experienced designers [16].[17].

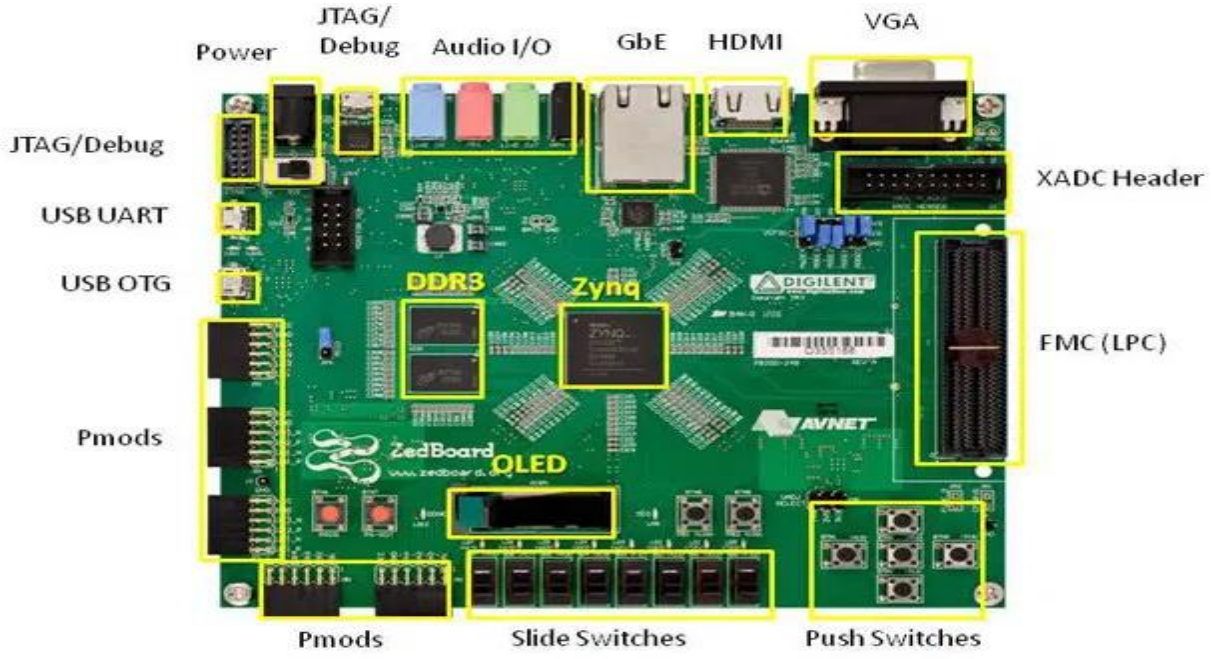


Figure (III.1) : Images de the ZedBoard Zynq-7000 Board

III.2.2. Featuresof ZedBoard Zynq-7000

- ***Xilinx® XC7Z020-1CSG484CES EPP***
  - Primary configuration = QSPI Flash
  - Auxiliary configuration options
  - Cascaded JTAG
  - 4 GB SD Card
- ***Memory***
  - 512 MB DDR3 (128M x 32)
  - 256 Mb QSPI Flash
- ***Interfaces***
  - USB-JTAG Programming using Digilent SMT1-equivalent circuit

- Accesses PL JTAG
- PS JTAG pins connected through PS Pmod
- 10/100/1G Ethernet
- USB OTG 2.0
- SD Card
- USB 2.0 FS USB-UART bridge
- Five Digilent Pmod ™ compatible headers (2x6) (1 PS, 4 PL)
- One LPC FMC
- One AMS Header
- Two Reset Buttons (1 PS, 1 PL)
- Seven Push Buttons (2 PS, 5 PL)
- Eight dip/slide switches (PL)
- Nine User LEDs (1 PS, 8 PL)
- DONE LED (PL)

- ***On-board Oscillators***
  - 33.333 MHz (PS)
  - 100 MHz (PL)
- Display/Audio
  - HDMI Output
  - VGA (12-bit Color)
  - 128x32 OLED Display
  - Audio Line-in, Line-out, headphone, microphone
- ***Power***
  - On/Off Switch
  - 12V @ 5A AC/DC regulator

- ***Software***
  - ISE® Web PACK Design Software
  - License voucher for Chip Scope™ Pro locked to XC7Z020

III.2.3. Applications

- Video processing
- Motor control
- Software acceleration
- Linux/Android/RTOS development
- Embedded ARM® processing
- General Zynq™-7000 AP SoC prototyping

III.3. XILINX VIVADO2018.3 Software:

In this work, we use the XILINX VIVADO 2018.3 software to implement the presented SPWM techniques, the main window of this software is shown in Figure (III.2).[18].

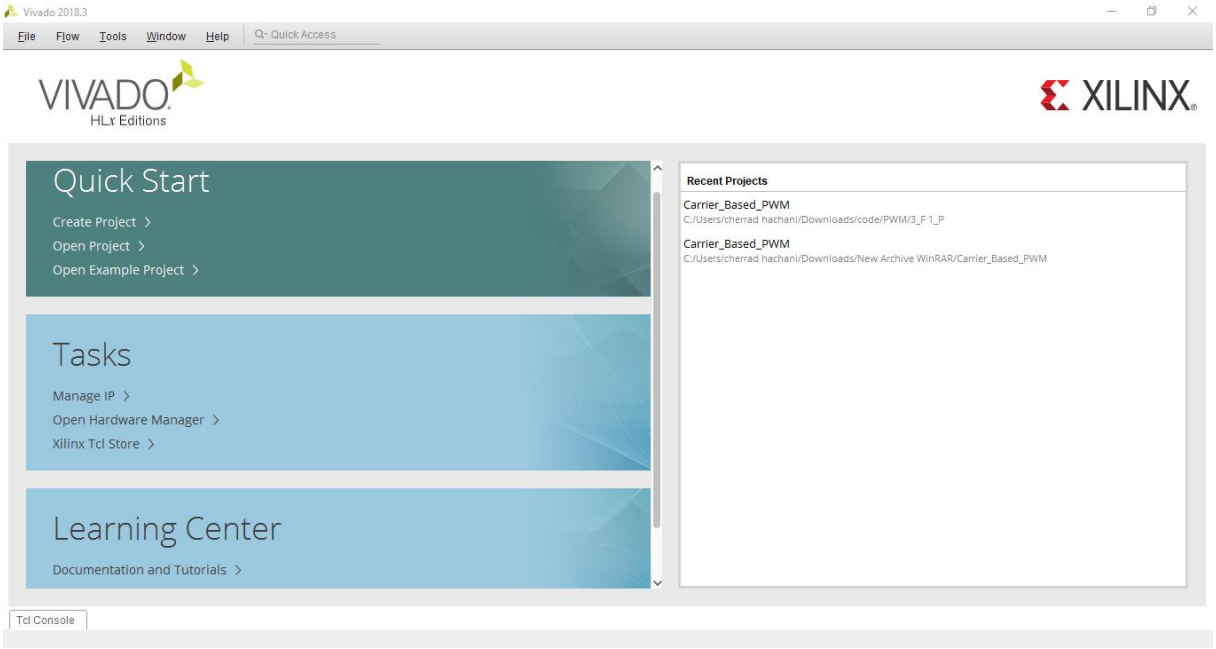
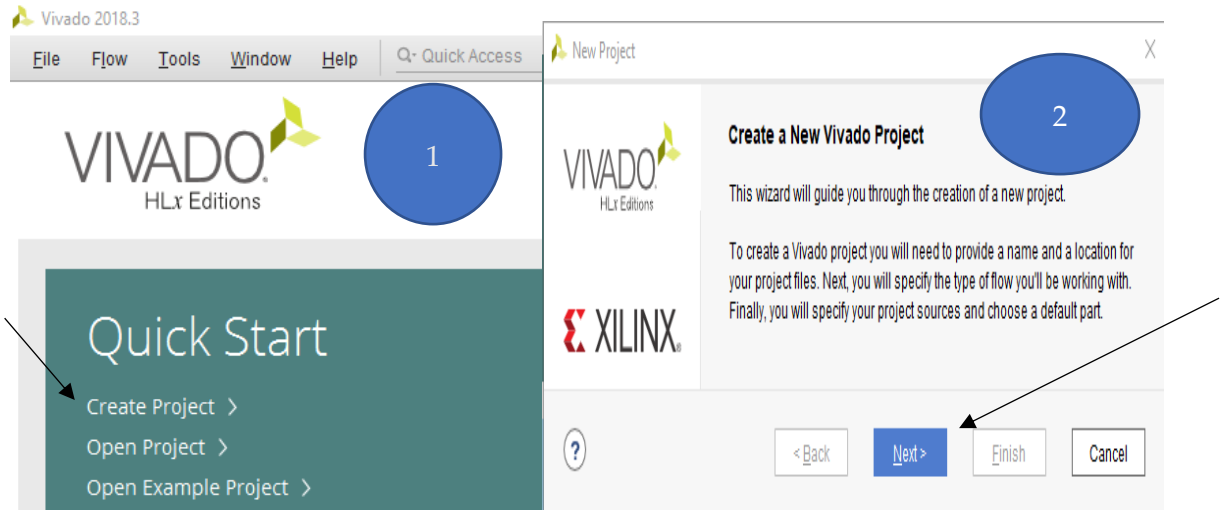
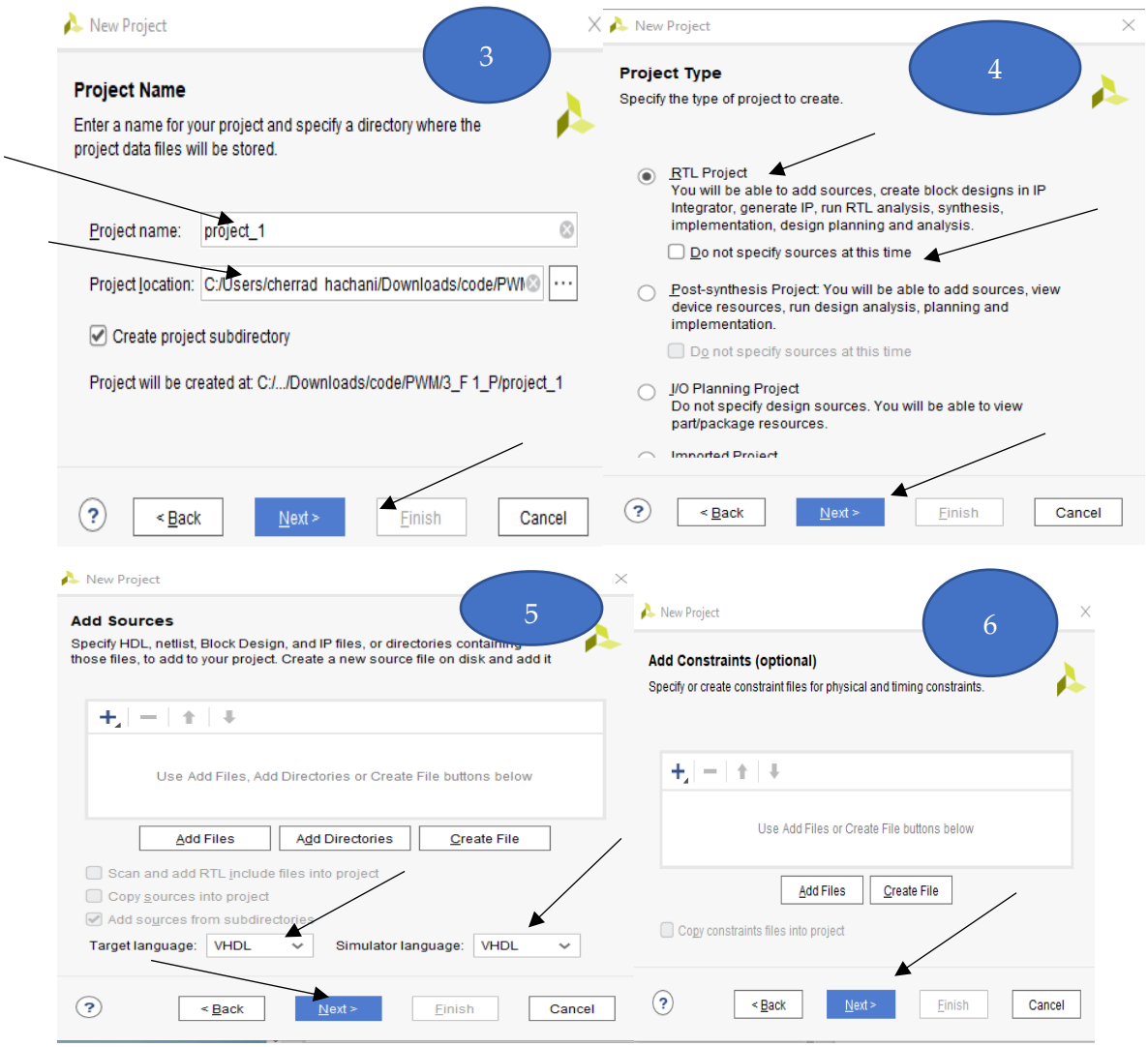


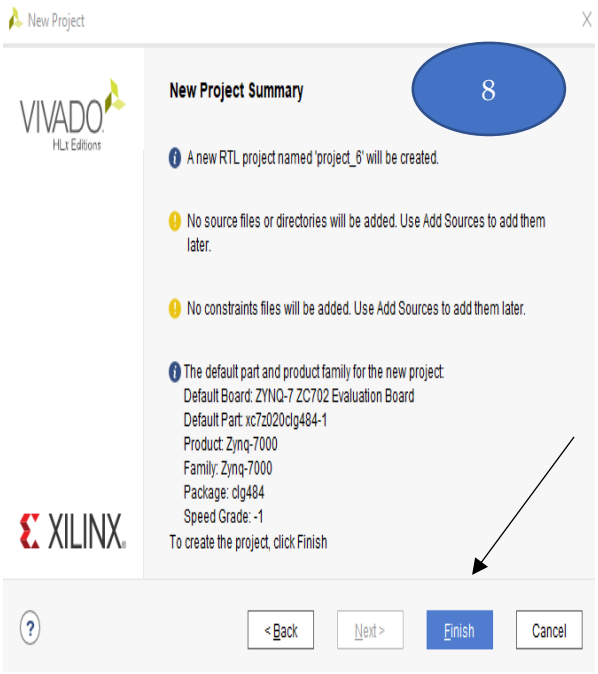
Figure (III.2) : Main window of XILINX VIVADO 2018.3 software

III.3.1. Steps of creating the project

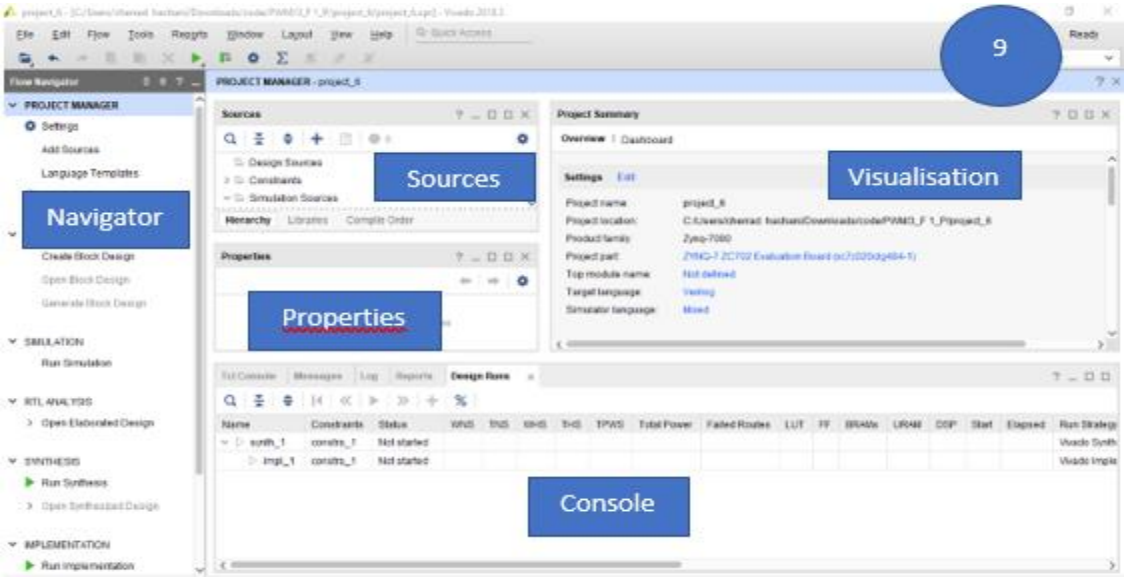
1) Click on “Create Project”.
2) In the next window, click on the “Next” button.
3) In the window that opens, type the name of the tp1 project, then click Next.
4) In the next window, check that “RTL Project” is selected, then click on the “Next” button.
5) In the next window, check that “VHDL” is the selected language, then click on the “Next” button.
6) In the next window, click on the “Next” button.
7)
8) You should see the following window before clicking “Finish” .



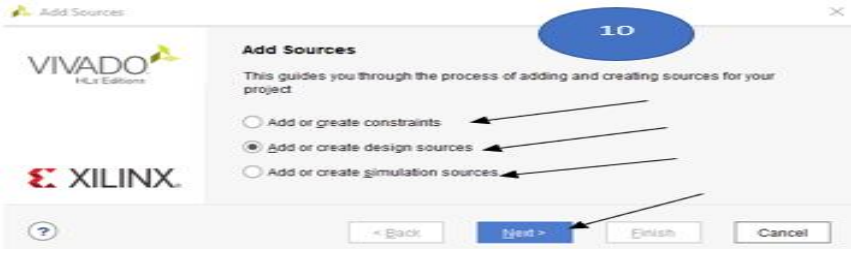

New Project
3
Project Name
Enter a name for your project and specify a directory where the project data files will be stored.
Project name: project_1
Project location: C:/Users/cherrad hachani/Downloads/code/PWM
Create project subdirectory
Project will be created at: C:/.../Downloads/code/PWM/3_F 1_P/project_1
< Back
Next >
Finish
Cancel
New Project
4
Project Type
Specify the type of project to create.
RTL Project
You will be able to add sources, create block designs in IP Integrator, generate IP, run RTL analysis, synthesis, implementation, design planning and analysis.
Do not specify sources at this time
Post-synthesis Project: You will be able to add sources, view device resources, run design analysis, planning and implementation.
Do not specify sources at this time
I/O Planning Project
Do not specify design sources. You will be able to view part/package resources.
< Back
Next >
Finish
Cancel
New Project
5
Add Sources
Specify HDL, netlist, Block Design, and IP files, or directories containing those files, to add to your project. Create a new source file on disk and add it
Use Add Files, Add Directories or Create File buttons below
Add Files
Add Directories
Create File
Scan and add RTL include files into project
Copy sources into project
Add sources from subdirectories
Target language: VHDL
Simulator language: VHDL
< Back
Next >
Finish
Cancel
New Project
6
Add Constraints (optional)
Specify or create constraint files for physical and timing constraints.
Use Add Files or Create File buttons below
Add Files
Create File
Copy constraints files into project
< Back
Next >
Finish
Cancel



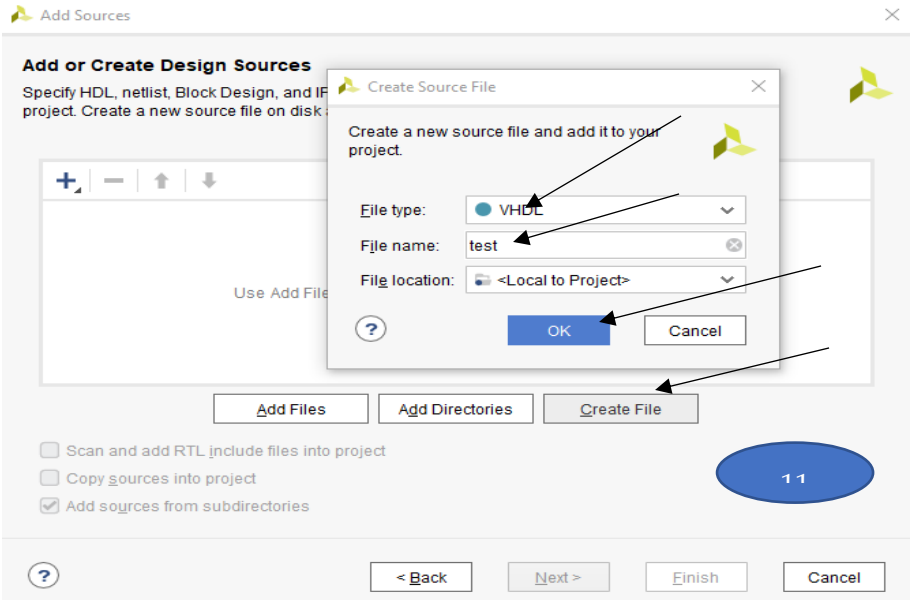
9) Vivado opens on the created project. Here are the different areas of the interface.



10) In the window that opens, select "Add or Create design sources" then click on "Next".

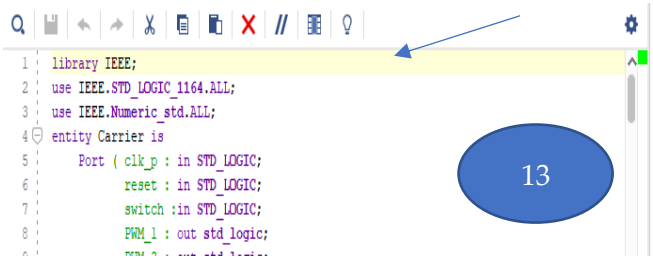


11) In the VHDL file creation window, type test in the "File Name" field then clicks



12) The VHDL file should now appear in the Project Manager (after Updating).

13) Syntax is checked on the fly as you type. Without syntax error, a small green square is displayed at the top right of the editing window.



III.4. VHDL-codes of the SPWM algorithms

VHDL code is the core part of the design, we will explain how to write the three codes used in the implementation.

III.4.1. Symmetrical control of single-phase H-bridge inverter

a- Entity

In the first two lines we define the library used in this case is

STD_LOGIC_1164.ALL;
Numeric_std.ALL;

Then define the input and output of the module, in this control method, one input which is the clock and on output which is the pulse.

clk_p (input)

PWM_1 (output).

b- Architecture:

In the architecture part we find counter identifier with signal integer=0, This means that a counter is a variable that initially has a value of 0.

c- Process:

Choose the rising pulse to determine the role.

Basically, this program contains a counter that increases in each cycle, maximum value is 2,000,000 Then it returns to zero, the counter fulfills two basic conditions:

1- **the first (counter < 1000000) As long as the condition is satisfied, the output PWM_1 = 0.**
2- **As for the second condition (counter < 2000000) as long as it is satisfied PWM_1 = 1.**

The full VHDL code of the symmetrical control technique is presented in the appendix.

III.4.2. Bipolar SPWM technique

The whole code of this control technique is presented in appendix. And here some explanations of how the code is work.

We use the same library.

a- Entity: We have three inputs and four outputs

Inputs:

- **clk_p**: clock is a signal inside any digital circuit the determines how fast a flip flop runs.
- **Reset:** button to reset to the initial state.
- **Switch:** A button to determine the value of switching frequency (carrier frequency), when the switch = 0, f_p=5khz, and when the switch = 1, f_p=10khz.

Outputs:

The outputs are the gate signals for the four IGBT, which are named as follows:

PWM_1, PWM_2, PWM_3, PWM_4.

b- Architecture:

- The periodic function defined by a ray, this ray determines the number of places and the largest value.
- 8 variables, each serving a specific role. We will address them when needed.

c- Process:

To achieve the switch part, 3 variables must be used. The first (**MH**) carries the value of the half- period, the second carries the value of the period (**2MH**), and the third (**MH_increase**) carries the value of the increment in each cycle. Thus, value is determined by the condition (switch='0').

The reference voltage named as **sine_function** which is a sine wave, we define it as an array contains 600 samples. The time between each of these values is the variable

sine_counter.

We express the carrier signal in the code by the variable **function_A**. which is determined by adding the number of increments in each cycle **MH_increase** as long as (**counter**) is less than (**MH**). But if (**counter**) is greater (**2MH**), (**MH_increase**) is subtracted from (**function_A**). The operation is returned when the counter reaches a value greater than (**2MH**).

III.4.3. Unipolar SPWM technique

The whole code of this control technique is presented in appendix. And here some explanations of how the code is work.

This is very similar to the previous one, except that it contains two reference voltages. We will only explain different part.

a- Architecture:

sine_counter_B, **sine_function_B**, **sine_table_counter_B**, these variables have the same role with its similarities in the previous code.

b- Process:

The first difference is that the variable **sine_table_counter_B** is not equal to zero at the beginning, which makes the reference functions does not take the same values, but rather they are reversed.

The **PWM_1** is the complement of **PWM_2** and **PWM_3** is the complement of **PWM_4**.

The value of **PWM_1** is determined by comparing **function_A** and **sine_function_A**.

The value of **PWM_3** is determined by comparing **function_A** and **sine_function_B**.

III.5. Hardware prototype

Figure (III.3) shows a picture of our hardware setup, which is composed the following parts:

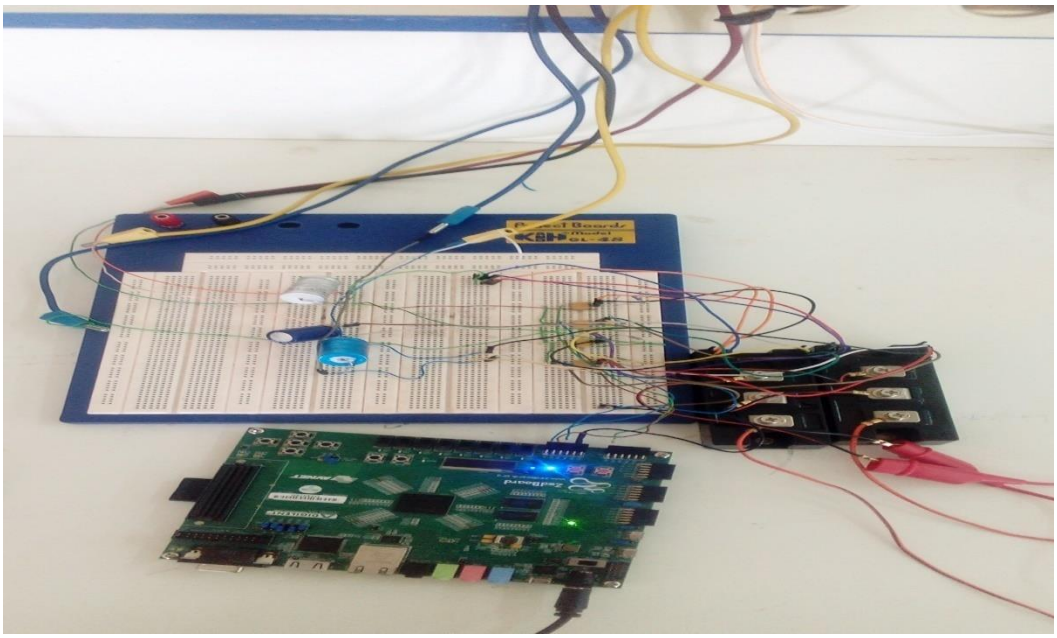


Figure (III.3) : Photograph of hardware setup

1) **Computer :** The computer allows us to program and provides the FPGA with the necessary software to work. It also gives us the ability to get an idea of the output through simulation.
2) **ZedBoard :** Control and processing unit for the program used in the experiment , allows to generate the PWM signal to the IGBT.
3) **Pulse generation circuit:** The part that allows us to generate the gate signals to turn on and off the IGBTs by a level voltage of 15V, it is based on photocouplers TLP2531.
4) **DC Voltage supply:** Generates a variable DC voltage.
5) **The oscilloscope:** used to display the output signals.

III.5.1. Photocoupler

The TOSHIBA TLP2531 dual photocouplers consist of a pair of GaAℓAs light emitting diode and integrated photodetector. Separate connection for the photodiode bias and output transistor collectors improves the speed up to a hundred times that of a conventional phototransistor coupler by reducing the base−collector capacitance.

- TTL compatible
- Switching speed: t_{pHL}=0.3µs, t_{pLH}=0.3µs.
- Guaranteed performance over temp: 0~70°C.
- Isolation voltage: 2500 V_{rms}.

We use this photocouplers as an intermediary interface between FPGA and IGBT to protect our kit.

1: Anode.1
2: Cathode.1
3: Cathode.2
4: Anode.2
5: GND
6: VO2 (output 2)
7: VO1 (output 1)
8: VCC

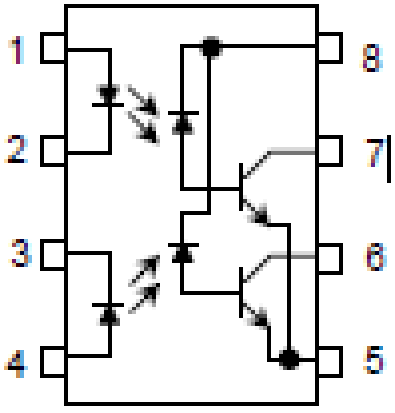


Figure (III.4): Pin configuration of the Photocoupler TLP2531

We connect the first input to one of the FPGA outputs of the control signal and the second to the GND. 3,4 and 6, we will not use it, leaving with input 8 connected to output pin 2 of IGBT and input 7

connected to IGBT control output (pin 3 of IGBT), 5 is a GND connected to the IGBT GND [19].

III.5.2. Module IGBT PM75DSA120:

Mitsubishi Intelligent Power Modules (IPMs) are advanced hybrid power devices that combine high speed, low loss IGBTs with optimized gate drive and protection circuitry. Highly effective over-current and short-circuit protection is realized through the use of advanced current sense IGBT chips that allow continuous monitoring of power device current. System reliability is further enhanced by the IPM's integrated over temperature and under voltage lock out protection. The schematic of the module M75DSA120 is shown in Figure (III.4) [20].

(a)

(b)

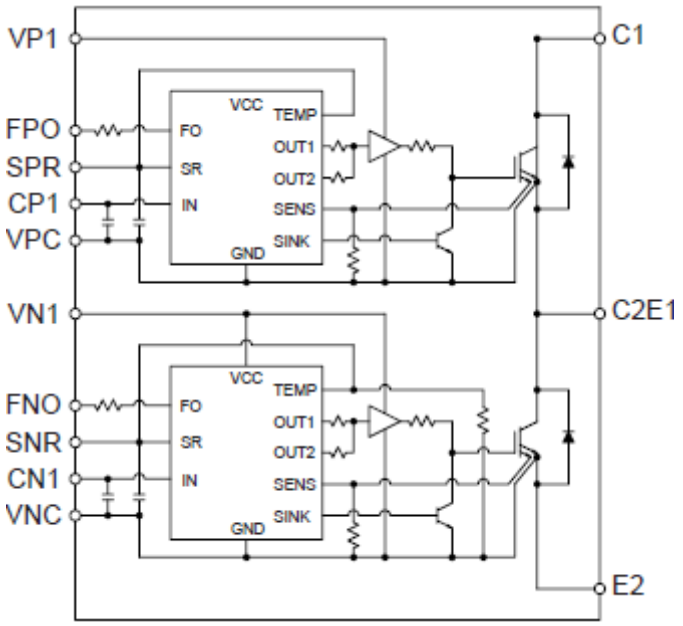


Figure (III.5) Module IGBT PM75DSA120 MITSUBISHI, (a) Its real picture, (b) Its schematic

The recommended conditions for use of the IGBT module PM75DSA120 are shown in Table (III.1)

Table (III.1) The recommended conditions for use for the IGBT module PM75DSA120

Pins	functional	Requirements
VP1 - VN1	Supply Voltage input	15V
FPO - FNO	Fault Output pin	5V
SPR - SNR	output for pouring photocouplers	5V
CP1 - CN1	Input for control signal	
VPC - VNC	GND	0V

III.5.3. Dead Time

In the design of inverters, it is necessary to include a "*dead time* "or "*Blanking time*" in switching signals in order to avoid simultaneous conduction of the devices on the same electrical trajectory or leg.

The turn-off time of power devices is usually longer than its turn-on time, and, therefore an appropriate dead time must be inserted between the upper and lower gating signals.

The incorporation of the dead time provides a safety zone, but it causes distortion of the output voltage and reduces its magnitude. The dead time for each leg was set to $4\ \mu s$ as usually used in these applications. The block diagram of dead time module is shown in Figure (III.6).[21].

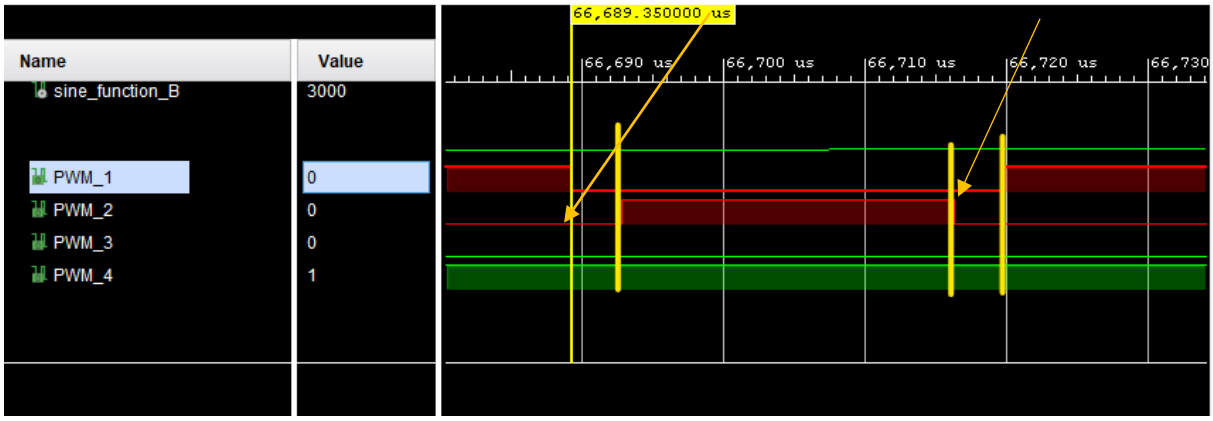
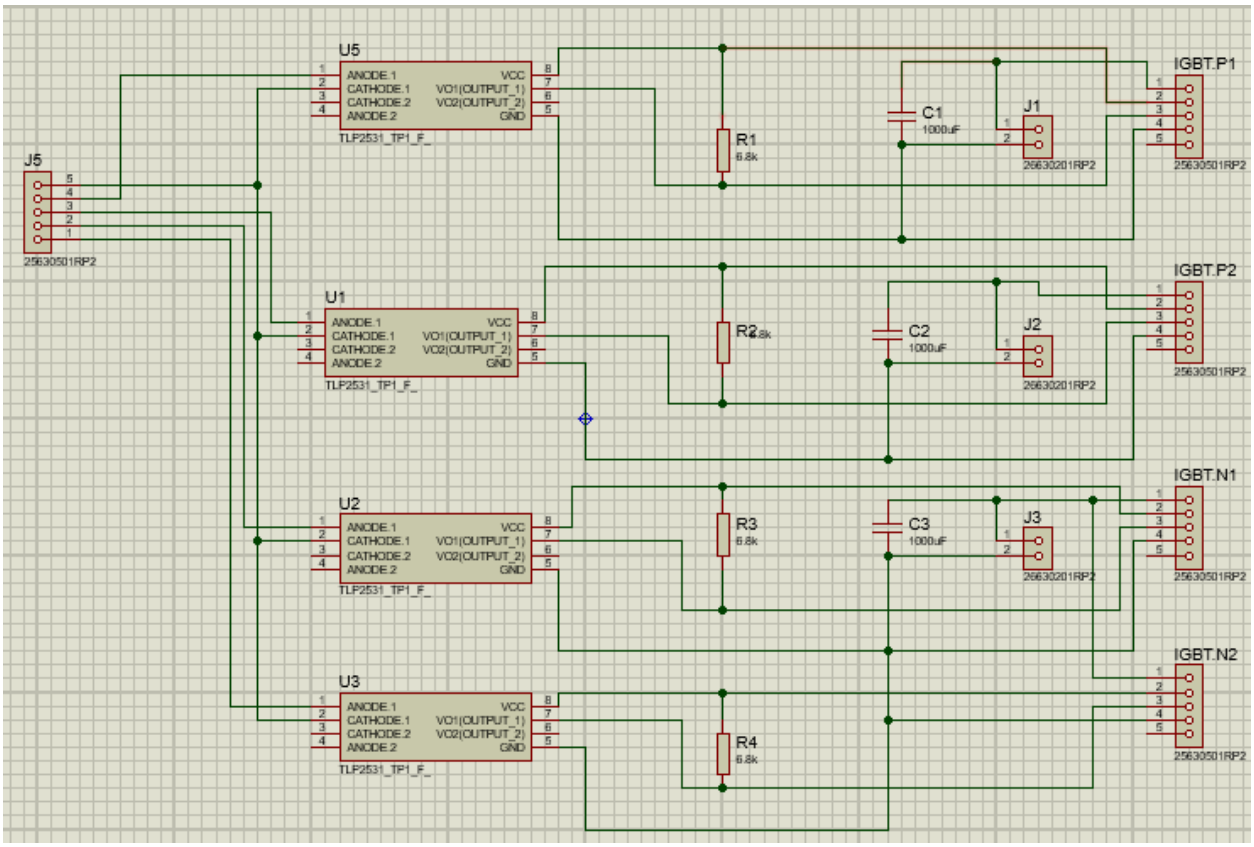


Figure (III.6): Dead Time creation between two complimentary switches.

III.6. Experimental results

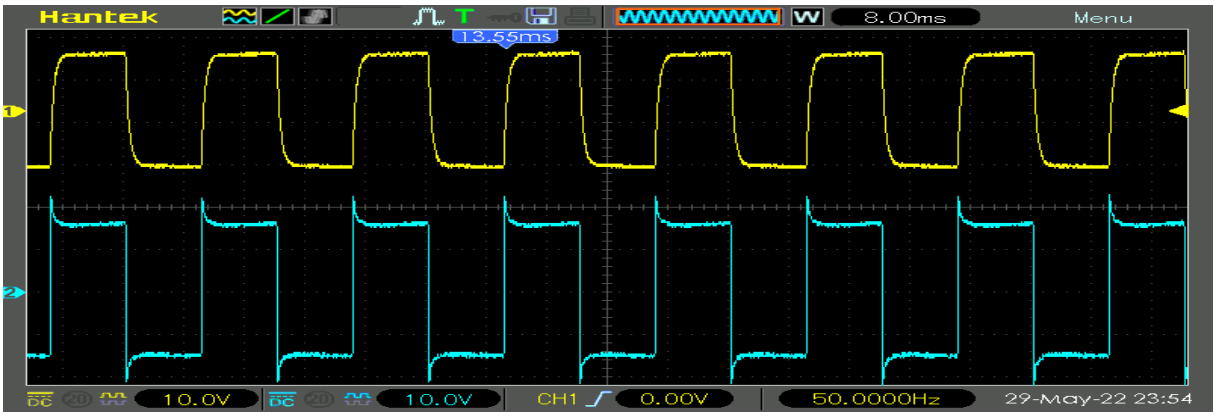
In this section, experimental results have been provided to validate the presented VHDL codes and SPWM techniques. The parameters used here are the same those used for simulation.

III.6.1. circuit diagram

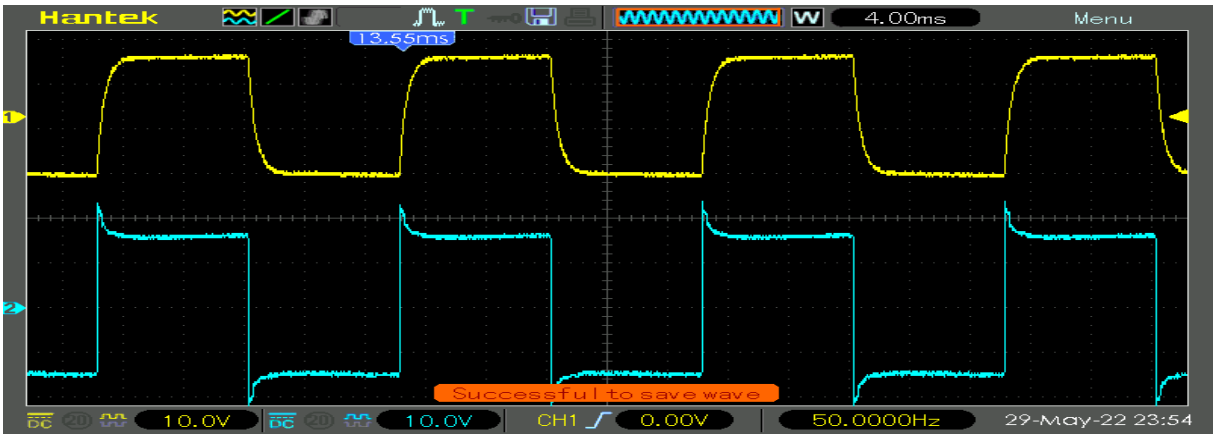


III.6.2. Symmetrical control (180°)

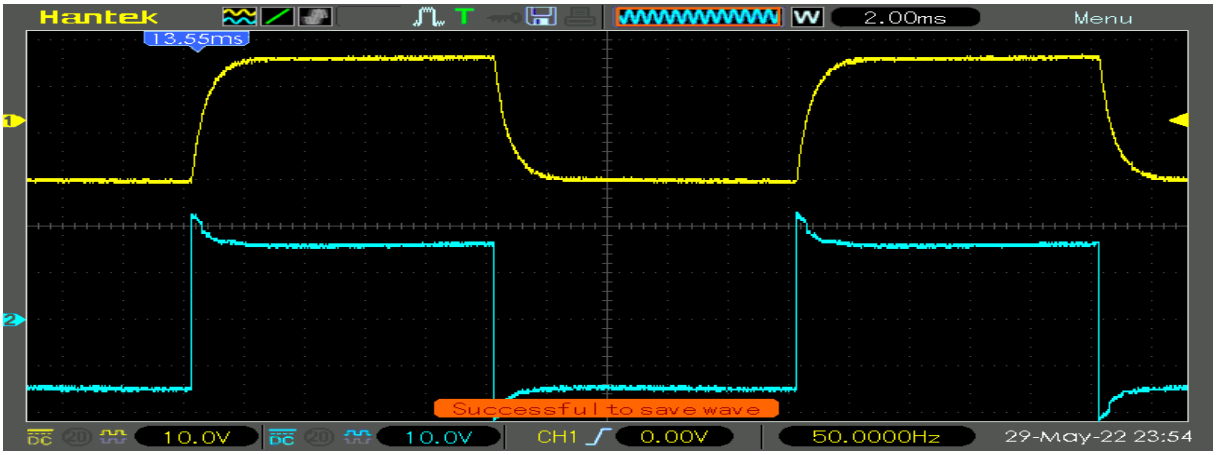
Figure (III.7) shows the output voltage and load current of the single-phase H-bridge inverter controlled using the symmetrical control method.



(a)



(b)

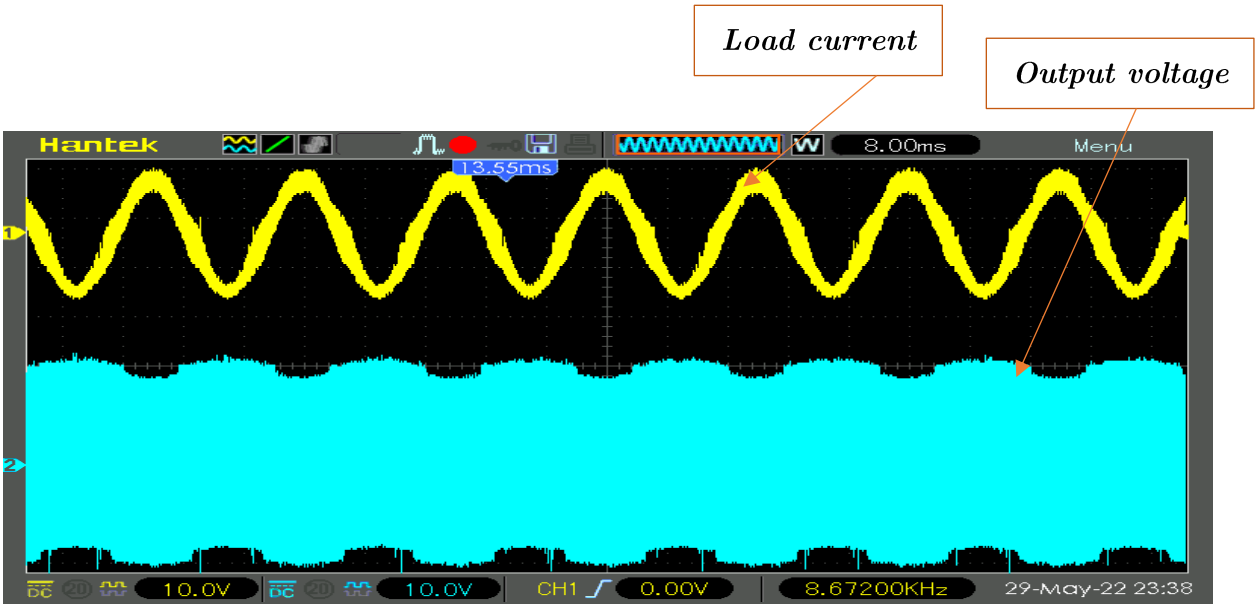


(C)

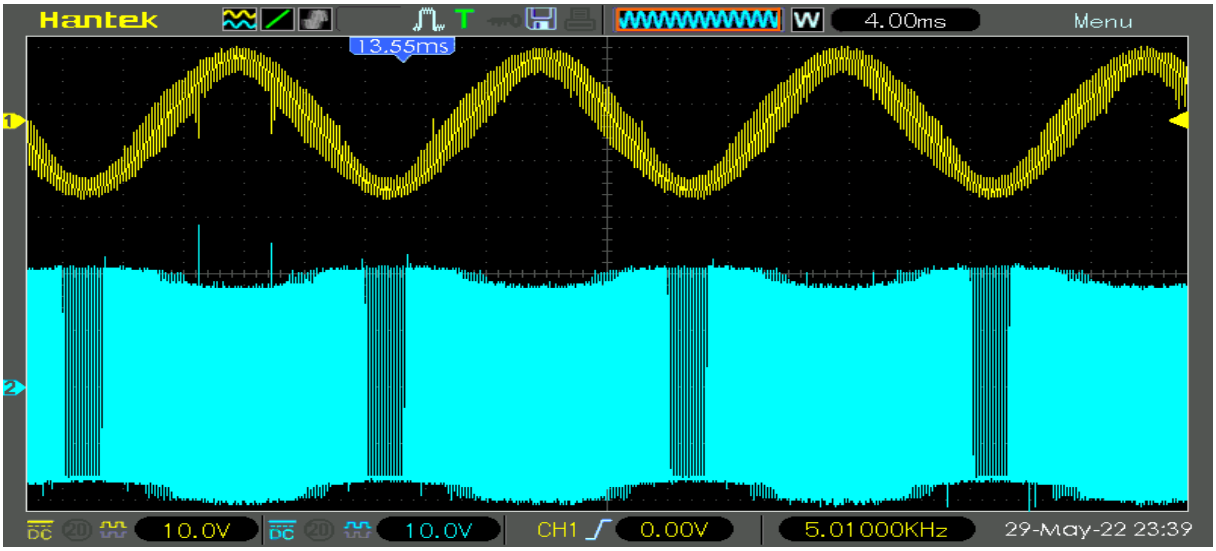
Figure (III.7): Output voltage and load current of the single-phase H-bridge inverter controlled using by the symmetrical control method, (a) 8ms, (b) 4ms, (c) 2ms.

III.6.3. Bipolar SPWM

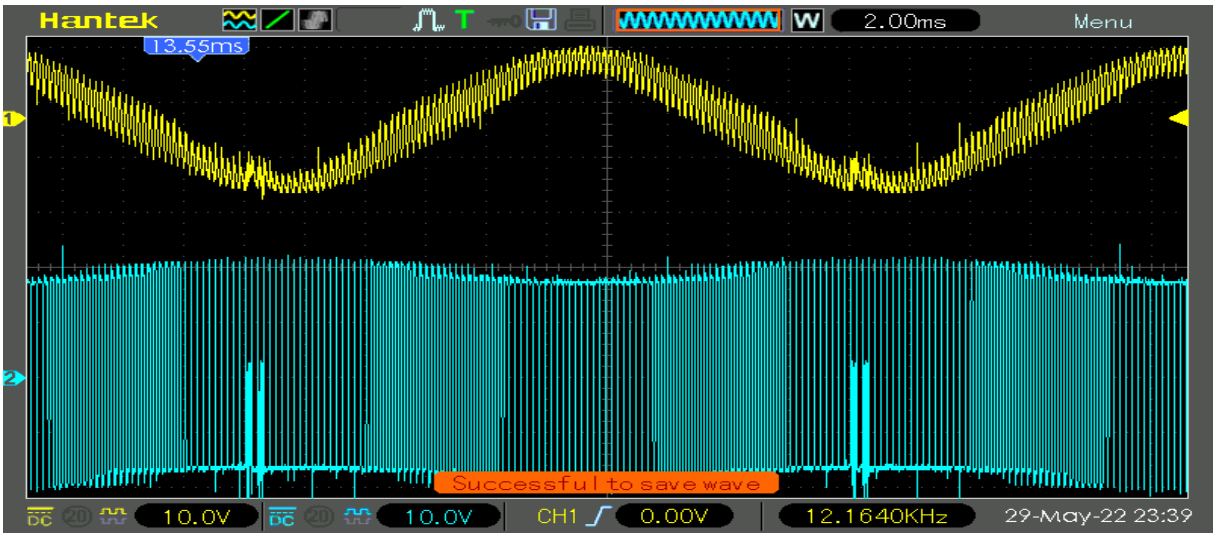
Figure (III.8): shows the output voltage and load current of the single-phase H-bridge inverter for the bipolar using the symmetrical control method.



(a)



(b)

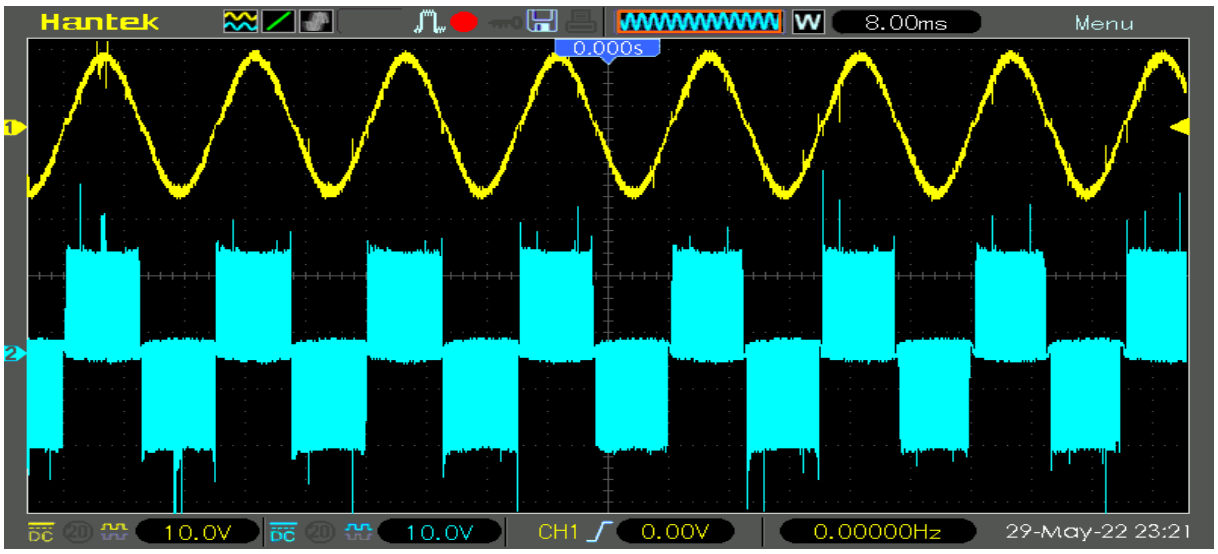


(c)

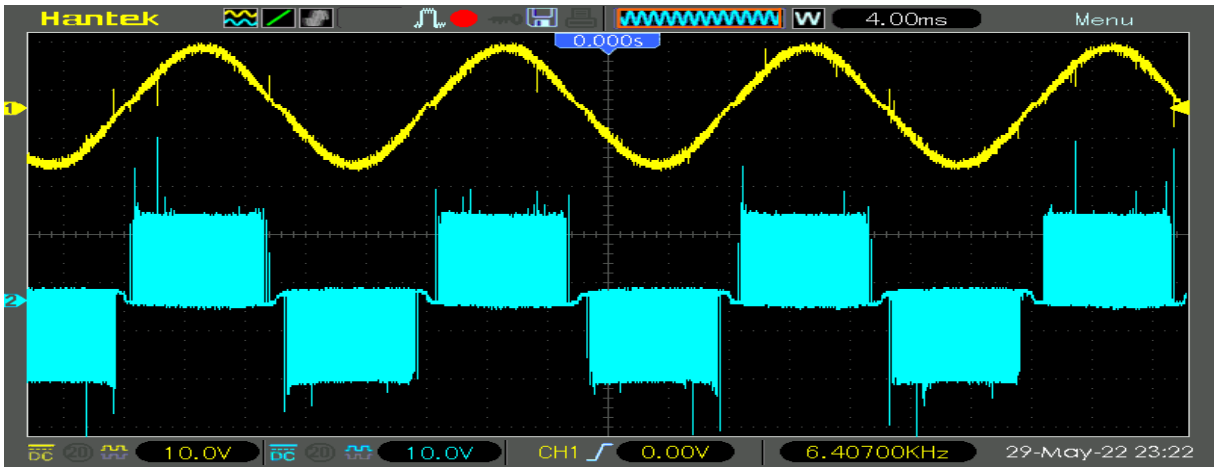
Figure (III.8): Output voltage and load current of the single-phase H-bridge inverter controlled using by the bipolar SPWM control method, (a) 8ms, (b) 4ms, (c) 2ms.

III.6.4. Unipolar SPWM

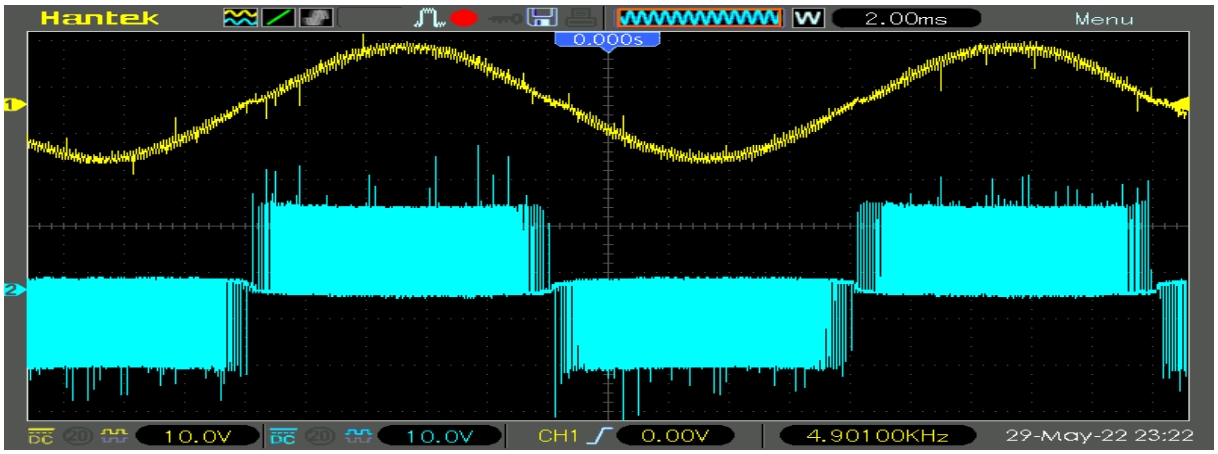
In the following (Figure III.9), the output voltage and load current are shown for unipolar SPWM.



(a)



(b)



(c)

Figure (III.9) : Output voltage and load current of the single-phase H-bridge inverter controlled using by the unipolar SPWM control method ,(a) 8ms ,(b) 4ms ,(c) 2ms .

III.7. Conclusion

This last chapter is reserved for the practical validation of the PWM pulse width modulation techniques for a single-phase H-bridge inverter.

We started this chapter with the different steps of the realization of the inverter and the implementation of the PWM algorithm in detail, then a general description for the development board is well illustrated. Finally, we provided the experimental results with their interpretations.

The experimental and simulation results during this end-of-study project are identical, which shows our success for the realization of the inverter and for the implementation of the PWM techn.

General conclusion

This work aims to present a theoretical study, simulation and practical realization of the single-phase voltage inverters controlled by PWM.
The stated objectives were firstly: the representation of the different types of inverters and their applications, as well as their operating modelling. all in showing their principle for different controls (control by square signals and control by PWM).
And in second point the work consisted in choosing among the available strategies of adjust both amplitude and frequency keeping the DC source constant (bridge to diode).
They are based on the adjustment band and the harmonic rate of the output voltages. Finally, we presented a comparative study between these different strategies.
The simulation results obtained are very satisfactory and adapt with the operation of the single-phase voltage inverter, depending on the type of control applied.
This work highlighted the importance of the practical realization, it is true that the simulation systems helped us to know where is going in the experiment, but the realization taught us many things and techniques that we would never have had with the simulation.

References

[1] S.MECHERI. “REALISATION D'UN ONDULEUR DE TENSION MONOPHASE CONTROLE PAR UNE CARTE ARDUINO“ Memory of Master in Electrical Engineering, UNIVERSITY OF CONSTANTINE 1,2014, pp. 1.

[2] Data sheet of Inverter: Types and Applications "Inverter: Types and Applications - Electrical Article ".

[3] Data sheet of Types of Inverters and their Applications "https://www.electricaltechnology.org/2020/06/types-of-inverters.html".

[4] Data sheet of Half Bridge Inverter : Circuit Diagram & Its Working " https://www.elprocus.com/what-is-half-bridge-inverter-circuit-diagram-its-working/ ".

[5] Data sheet of Single-Phase Bridge Inverter" https://www.sciencedirect.com/topics/engineering/single-phase-bridge-inverter".

[6] K. CHETTIH , "Réalisation d’un onduleur monophasé avec un bras redondant" , Memory of Master in Electrical Engineering, University of Constantine Mentouri Brothers,pp.12, ,Juin 2015.

[7] Ahmed M. T. Ibraheem, " FPGA Based Three-Phase Sinusoidal PWM Control for Voltage Source Inverter Fed IM " ,Memory of the Technical Master Degree in Electrical Power Technology Engineering, Technical College / Mosul ,Republic of Iraq, pp.9,2010.

[8] Data sheet of Single Phase Pulse Width Modulated Inverters "CHAPTER 2 (tntech.edu)".

[9] Data sheet of Pulse Width Modulated Inverters " Microsoft Word - Lecture 9 Pulse Width Modulated Inverters.doc (dit.ie)" .

[10] Data sheet of Single PWM Inverters " Single PWM Inverters | DC-TO-AC INVERTER | Electronics Tutorial (electronics-tutorial.net)".

[11] Data sheet of Design and Simulation of two Stages Single Phase PV Inverter operating in Standalone Mode without Batteries "Unipolar SPWM switching scheme (a) Bipolar SPWM generation circuit, (b)... | Download Scientific Diagram (researchgate.net) ".

[12] E,Aboadla; S,Khan; M,Habaebi; T,Gunawan; A,Belal ,"Effect of modulation index of pulse width modulation inverter on Total Harmonic Distortion for Sinusoidal" 2016 International Conference on Intelligent Systems Engineering (ICISE),pp.12, May 2016.

[13] I,JIRST , "Unipolar and Bipolar PWM Inverter", International Journal for Innovative Research in Science & Technology , Volume 1 , Issue 7 , December 2014.

[14] A.B. Afarulrazi, M. Zarafi, W. M. Utomo and A. 2010 International Conference on Computer Applications and Industrial Electronics (ICCAIE 2010), December 5-7, 2010, Kuala Lumpur, Malaysia , Department of Electrical Power Engineering Faculty of Electrical and Electronic Engineering, UTHM Johore, Malaysia

[15] Data sheet of Single-Phase Bridge Inverter"Single-Phase Bridge Inverter - an overview | ScienceDirect Topics".

[16] Avnet, "ZynqTM Evaluation and Development Hardware User's Guide," 2018.[Online].Available:https://reference.digilentinc.com/reference/programmableloc/zedboard/reference-manual. [Accessed: 01-Aug-2012].

[17] Data sheet of ZedBoard " https://digilent.com/reference/programmable-logic/zedboard/start " .

[18] Data sheet of Introduction à la conception numérique en VHDL "https://easytp.cnam.fr/alexandre/index_fichiers/support/zynq_cours_tp_vivado_basys3.pdf ".

[19] Data sheet of " Photocoupler GaAℓAs Ired & Photo IC TLP2530, TLP2531",2002. https://pdf1.alldatasheet.com/datasheet.pdf/view/101146/TOSHIBA/TLP2531.html?fbclid=IwAR3luuA90eTmQshBL9XzJzmEYRssvCNyWk-zqCUXbu8b6gAx7bzhE6mFHps.

[20] Data sheet of ,"PM75DSA120"

https://www.digchip.com/datasheets/parts/datasheet/375/PM75DSA120pdf.php?fbclid=IwAR15bQGUaJBxt0TtBGUA8xY6iyMeFgU6p9YAPN4MJVLzeZMNyFkm.

[21] V.M. Cardenas; S. Horta; R. Echavarria " Elimination of dead time effects in three phase inverters" V IEEE International Power Electronics Congress Technical Proceedings, CIEP 96 , Cuernavaca, Mexico , pp.7 ,August 2002.

Appendix

Code VHDL for symmetrical control

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.Numeric_std.ALL;
entity Carrier is
    Port ( clk_p : in STD_LOGIC;
reset : in STD_LOGIC;

          PWM_1 : out std_logic;
          PWM_2 : out std_logic;
          PWM_3 : out std_logic;
          PWM_4 : out std_logic);
end Carrier;

architecture Behavioral of Carrier is
signal counter:integer:=0;
signal counter1:integer:=0;
begin
process (reset, clk_p)
  begin
        if (reset = '1') then
                    else
                        if (rising_edge(clk_p)) then
                                    if counter < 999860 then
                PWM_1<='1';
                 PWM_2<='1';
                                                counter <= counter + 1 ;
                                                  else if counter < 2000140 then
                                                     PWM_1<='0';
                                                     PWM_2<='0';
                                                     counter <= counter + 1;
                                                     else if counter = 2000140 then
                                                     counter <= 0;
                                                     else counter <= counter + 1 ;
                                                     end if;endif;end if;

                                                if counter1 < 1000140 then
                PWM_3<='0';
                 PWM_4<='0';
                                                counter1 <= counter1 + 1 ;
                                                  else if counter < 1000280 then

                                                     counter1 <= counter + 1;
```

```
                                        else if counter1 < 1999860 then
                                       PWM_3<='1';
                                        PWM_4<='1';
                                        counter1 <= counter1 + 1 ;
                                        else
                                        counter1 <= 0;
                                        end if;endif;end if;


                        end if;end if;
end process;
end Behavioral;
```

Code VHDL for Bipolar SPWM

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.Numeric_std.ALL;
entity Carrier is
    Port ( clk_p : in STD_LOGIC;
reset : in STD_LOGIC;
switch :in STD_LOGIC;
           PWM_1 : out std_logic;
           PWM_2 : out std_logic;
           PWM_3 : out std_logic;
           PWM_4 : out std_logic);
end Carrier;
architecture Behavioral of Carrier is
signal counter:integer:=0;
signal function_A :integer:=0;
signal MH :integer:=0;
signal MH2 :integer:=0;
signal MH_increase :integer :=0;
signal sine_counter :integer :=0;
signal sine_function :integer :=0;
signal sine_table_counter :integer :=0;
type t_sine is array (0 to 599) of integer range 0 to 20000;
-------------------------------------100*[100+60*sin(x)]=y
signal r_sine : t_sine :=
(8000,8100,8200,8300,8300,8400,8500,8600,8700,8800,8800,8900,9000,9100,9200,9300,9300,9400
,9500,9600,9700,9700,9800,9900,10000,10100,10200,10200,10300,10400,10500,10600,10600,1070
0,10800,10900,10900,11000,11100,11200,11300,11300,11400,11500,11600,11600,11700,11800,119
00,11900,12000,12100,12100,12200,12300,12400,12400,12500,12600,12600,12700,12800,12800,12
900,13000,13000,13100,13200,13200,13300,13400,13400,13500,13500,13600,13700,13700,13800,1
```

```
3800,13900,13900,14000,14100,14100,14200,14200,14300,14300,14400,14400,14500,14500,14600,
14600,14700,14700,14800,14800,14800,14900,
14900,15000,15000,15100,15100,15100,15200,15200,15200,15300,15300,15300,15400,15400,1540
0,15500,15500,15500,15600,15600,15600,15600,15700,15700,15700,15700,15700,15800,15800,158
00,15800,15800,15900,15900,15900,15900,15900,15900,15900,15900,16000,16000,16000,16000,16
000,16000,16000,16000,16000,16000,16000,16000,16000,16000,16000,16000,16000,16000,16000,1
6000,16000,15900,15900,15900,15900,15900,15900,15900,15900,15800,15800,15800,15800,15800,
15700,15700,15700,15700,15700,15600,15600,15600,15600,15500,15500,15500,15400,15400,1540
0,15300,15300,15300,15200,15200,15200,15100,15100,15100,15000,15000,
14900,14900,14800,14800,14800,14700,14700,14600,14600,14500,14500,14400,14400,14300,1430
0,14200,14200,14100,14100,14000,13900,13900,13800,13800,13700,13700,13600,13500,13500,134
00,13400,13300,13200,13200,13100,13000,13000,12900,12800,12800,12700,12600,12600,12500,12
400,12400,12300,12200,12100,12100,12000,11900,11900,11800,11700,11600,11600,11500,11400,1
1300,11300,11200,11100,11000,10900,10900,10800,10700,10600,10600,10500,10400,10300,10200,
10200,10100,10000,9900,9800,9700,9700,9600,9500,9400,9300,9300,9200,9100,9000,8900,8800,88
00,8700,8600,8500,8400,8300,8300,8200,8100,
8000,7900,7800,7700,7700,7600,7500,7400,7300,7200,7200,7100,7000,6900,6800,6700,6700,6600,
6500,6400,6300,6300,6200,6100,6000,5900,5800,5800,5700,5600,5500,5400,5400,5300,5200,5100,
5100,5000,4900,4800,4700,4700,4600,4500,4400,4400,4300,4200,4100,4100,4000,3900,3900,3800,
3700,3600,3600,3500,3400,3400,3300,3200,3200,3100,3000,3000,2900,2800,2800,2700,2600,2600,
2500,2500,2400,2300,2300,2200,2200,2100,2100,2000,1900,1900,1800,1800,1700,1700,1600,1600,
1500,1500,1400,1400,1300,1300,1200,1200,1200,1100,
1100,1000,1000,900,900,900,800,800,800,700,700,700,600,600,600,500,500,500,400,400,400,400,3
00,300,300,300,300,200,200,200,200,200,100,100,100,100,100,100,100,100,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,100,100,100,100,100,100,100,100,200,200,200,200,200,300,300,300,300,300,400,400
,400,400,500,500,500,600,600,600,700,700,700,800,800,800,900,900,900,1000,1000,
1100,1100,1200,1200,1200,1300,1300,1400,1400,1500,1500,1600,1600,1700,1700,1800,1800,1900,
1900,2000,2100,2100,2200,2200,2300,2300,2400,2500,2500,2600,2600,2700,2800,2800,2900,3000,
3000,3100,3200,3200,3300,3400,3400,3500,3600,3600,3700,3800,3900,3900,4000,4100,4100,4200,
4300,4400,4400,4500,4600,4700,4700,4800,4900,5000,5100,5100,5200,5300,5400,5400,5500,5600,
5700,5800,5800,5900,6000,6100,6200,6300,6300,6400,6500,6600,6700,6700,6800,6900,7000,7100,
7200,7200,7300,7400,7500,7600,7700,7700,7800,7900);
begin
process (reset, clk_p)
  begin
      if (reset = '1') then
          counter <= 0;
function_A<=0;
sine_counter<=0;
sine_function<=0;
```

```
sine_table_counter<=0;
        MH <=0;
        MH2 <=0;
MH_increase<=0;
                    else
                        if (rising_edge(clk_p)) then
                        if switch='0'then
            MH<=10000;
MH_increase<=2;
            MH2<=MH+MH;
            else
            MH<=5000;
MH_increase<=4;
            MH2<=MH+MH;
            end if;
                                        if sine_counter< 3333 then
                                       sine_counter<= sine_counter+1;
                                        else
                                          if sine_table_counter< 599 then
                                       sine_counter<=0;
                                       sine_function<= r_sine(sine_table_counter)+2000;
                                       sine_table_counter<= sine_table_counter + 1;
                                            else
                                       sine_counter<=0;
                                       sine_function<= r_sine(sine_table_counter)+2000;
                                       sine_table_counter<=0;
                                            end if ;end if;

              if counter < MH then
function_A<= function_A + MH_increase;
                                                 counter <= counter + 1 ;
                                                   else if counter < MH2 then
                                                function_A<= function_A - MH_increase;
                                                     counter <= counter + 1;
                                                     else
                                                     counter <= 0;
                                                     end if;end if;

                                  if function_A< (sine_function) then
                                              PWM_1<='1';
```

```
                                        PWM_2<='1';
                                        else
                                        PWM_1<='0';
                                        PWM_2<='0';

                                        end if;
if  (function_A) < (sine_function) then
                                        PWM_3<='0';
                                        PWM_4<='0';
                                        else
                                        PWM_3<='1';
                                        PWM_4<='1';
                                        end if;
                        end if;end if;
  end process;
end Behavioral;
```

Code VHDL for unipolar SPWM

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.Numeric_std.ALL;
entity Carrier is
    Port ( clk_p : in STD_LOGIC;
reset : in STD_LOGIC;
switch :in STD_LOGIC;
           PWM_1 : out std_logic;
           PWM_2 : out std_logic;
           PWM_3 : out std_logic;
           PWM_4 : out std_logic);
end Carrier;
architecture Behavioral of Carrier is
signal counter:integer:=0;
signal function_A :integer:=0;
signal MH :integer:=0;
signal MH2 :integer:=0;
signal MH_increase :integer :=0;
signal sine_counter :integer :=0;
signal sine_counter_B :integer :=0;
signal sine_function :integer :=0;
```

```
signal sine_function_B :integer :=0;
signal sine_table_counter :integer :=0;
signal sine_table_counter_B :integer :=299;
type t_sine is array (0 to 599) of integer range 0 to 20000;
-----------------------------------100*[100+60*sin(x)]=y
signal r_sine : t_sine :=
(8000,8100,8200,8300,8300,8400,8500,8600,8700,8800,8800,8900,9000,9100,9200,9300,9300,9400
,9500,9600,9700,9700,9800,9900,10000,10100,10200,10200,10300,10400,10500,10600,10600,1070
0,10800,10900,10900,11000,11100,11200,11300,11300,11400,11500,11600,11600,11700,11800,119
00,11900,12000,12100,12100,12200,12300,12400,12400,12500,12600,12600,12700,12800,12800,12
900,13000,13000,13100,13200,13200,13300,13400,13400,13500,13500,13600,13700,13700,13800,1
3800,13900,13900,14000,14100,14100,14200,14200,14300,14300,14400,14400,14500,14500,14600,
14600,14700,14700,14800,14800,14800,14900,
14900,15000,15000,15100,15100,15100,15200,15200,15200,15300,15300,15300,15400,15400,1540
0,15500,15500,15500,15600,15600,15600,15600,15700,15700,15700,15700,15700,15800,15800,158
00,15800,15800,15900,15900,15900,15900,15900,15900,15900,15900,16000,16000,16000,16000,16
000,16000,16000,16000,16000,16000,16000,16000,16000,16000,16000,16000,16000,16000,16000,1
6000,16000,15900,15900,15900,15900,15900,15900,15900,15800,15800,15800,15800,15800,
15700,15700,15700,15700,15700,15600,15600,15600,15600,15500,15500,15500,15400,15400,1540
0,15300,15300,15300,15200,15200,15200,15100,15100,15100,15000,15000,
14900,14900,14800,14800,14800,14700,14700,14600,14600,14500,14500,14400,14400,14300,1430
0,14200,14200,14100,14100,14000,13900,13900,13800,13800,13700,13700,13600,13500,13500,134
00,13400,13300,13200,13200,13100,13000,13000,12900,12800,12800,12700,12600,12600,12500,12
400,12400,12300,12200,12100,12100,12000,11900,11900,11800,11700,11600,11600,11500,11400,1
1300,11300,11200,11100,11000,10900,10900,10800,10700,10600,10600,10500,10400,10300,10200,
10200,10100,10000,9900,9800,9700,9700,9600,9500,9400,9300,9300,9200,9100,9000,8900,8800,88
00,8700,8600,8500,8400,8300,8300,8200,8100,
8000,7900,7800,7700,7700,7600,7500,7400,7300,7200,7200,7100,7000,6900,6800,6700,6700,6600,
6500,6400,6300,6300,6200,6100,6000,5900,5800,5800,5700,5600,5500,5400,5400,5300,5200,5100,
5100,5000,4900,4800,4700,4700,4600,4500,4400,4400,4300,4200,4100,4100,4000,3900,3900,3800,
3700,3600,3600,3500,3400,3400,3300,3200,3200,3100,3000,3000,2900,2800,2800,2700,2600,2600,
2500,2500,2400,2300,2300,2200,2200,2100,2100,2000,1900,1900,1800,1800,1700,1700,1600,1600,
1500,1500,1400,1400,1300,1300,1200,1200,1200,1100,
1100,1000,1000,900,900,900,800,800,800,700,700,700,600,600,600,500,500,500,400,400,400,400,3
00,300,300,300,300,200,200,200,200,200,100,100,100,100,100,100,100,100,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,100,100,100,100,100,100,100,100,200,200,200,200,200,300,300,300,300,300,400,400
,400,400,500,500,500,600,600,600,700,700,700,800,800,800,900,900,900,1000,1000,
1100,1100,1200,1200,1200,1300,1300,1400,1400,1500,1500,1600,1600,1700,1700,1800,1800,1900,
1900,2000,2100,2100,2200,2200,2300,2300,2400,2500,2500,2600,2600,2700,2800,2800,2900,3000,
3000,3100,3200,3200,3300,3400,3400,3500,3600,3600,3700,3800,3900,3900,4000,4100,4100,4200,
```

```
4300,4400,4400,4500,4600,4700,4700,4800,4900,5000,5100,5100,5200,5300,5400,5400,5500,5600,
5700,5800,5800,5900,6000,6100,6200,6300,6300,6400,6500,6600,6700,6700,6800,6900,7000,7100,
7200,7200,7300,7400,7500,7600,7700,7700,7800,7900);
begin
process (reset, clk_p)
  begin
       if (reset = '1') then
          counter <= 0;
function_A<=0;
sine_counter<=0;
sine_function<=0;
sine_counter_B<=0;
sine_function_B<=0;
sine_table_counter<=0;
sine_table_counter_B<=299;
                      else
                          if (rising_edge(clk_p)) then
                          if switch='0'then
             MH<=10000;--2500*2=5000
MH_increase<=2;--200/2500=8
             MH2<=MH+MH;
             else
             MH<=5000;--10000/2=5000
MH_increase<=4;--200/5000=4
             MH2<=MH+MH;
             end if;
                                        if sine_counter< 3333 then  --62500*(31+1)=2000000
                                       sine_counter<= sine_counter+1;
                                        else
                                          if sine_table_counter< 599 then
                                       sine_counter<=0;
                                       sine_function<= r_sine(sine_table_counter)+2000;
                                       sine_table_counter<= sine_table_counter + 1;
                                             else
                                       sine_counter<=0;
                                       sine_function<= r_sine(sine_table_counter)+2000;
                                       sine_table_counter<=0;
                                             end if ;end if;

                                        if sine_counter< 3333 then  --62500*(31+1)=2000000
```

```
                                    else
                                    if sine_table_counter_B< 599 then
                                   sine_counter_B<=0;
                                   sine_function_B<= r_sine(sine_table_counter_B)+2000;
                                   sine_table_counter_B<= sine_table_counter_B + 1;
                                         else
                                   sine_counter_B<=0;
                                   sine_function_B<= r_sine(sine_table_counter_B)+2000;
                                   sine_table_counter_B<=0;
                                         end if ;end if ;
                  if counter < MH then
function_A<= function_A + MH_increase;
                                              counter <= counter + 1 ;
                                               else if counter < MH2 then
                                              function_A<= function_A - MH_increase;
                                                  counter <= counter + 1;
                                                  else
                                                  counter <= 0;
                                                  end if;end if;
                               if function_A< (sine_function) then
                                         PWM_1<='1';
                                         else
                                         PWM_1<='0';
                                         end if;
if  (function_A) < (sine_function+700) then
                                         PWM_2<='0';
                                         else
                                         PWM_2<='1';
                                         end if;
                                   if  (function_A) < (sine_function_B) then
                                         PWM_3<='1';
                                         else
                                         PWM_3<='0';
                                         end if;
                                         if function_A< (sine_function_B+700) then
                                         PWM_4<='0';
                                         else
                                         PWM_4<='1';
                                         end if;
```

```
                    end if;end if;
end process;
end Behavioral;
```

Abstract :

Within the framework of the preparation of the Diploma of Master in Electronics engineering,this work based on practical realization of the single-phase inverter the installation ofexperimental test benches, on the level of the Workshop of the Department of Electronics and Telecommunication.

This project is related to the study and realization of a single-phase voltage inverter. The first chapter is dedicated to the mathematical modeling of thissingle -phase inverter.

In the second chapter we presented the different control techniques of this inverter with their simulations in the environment of MATLAB software, in order to have the best control technique. And we practically validated our chosen control technique in the third chapter on a single-phase inverter that we realized at the level of the Workshop of the Department of Electronics and Telecommunication .

The experimental results and the simulation are identical, which shows our success for the realization of the inverter and for the implementation of the PWM technique.

Key words: single-phase inverter, PWM, static converter, FPGA card, experimental validation.

Résumé :

Dans le cadre de la préparation du Diplôme de Master en Génie Electronique, ce travail a basé sur la réalisation pratique de l'onduleur monophasé l'installation de bancs d'essais expérimentaux, au niveau de l'Atelier du Département d'Electronique et de Télécommunication.

Ce projet porte sur l'étude et la réalisation d'un onduleur de tension monophasé. Le premier chapitre est consacré à la modélisation mathématique de cet onduleur monophasé.

Dans le deuxième chapitre nous avons présenté les différentes techniques de contrôle de cet onduleur avec leurs simulations dans l'environnement du logiciel MATLAB, afin d'avoir la meilleure technique de contrôle. Et nous avons pratiquement validé notre technique de contrôle choisie dans le troisième chapitre sur un onduleur monophasé que nous avons réalisé au niveau de l'Atelier du Département d'Electronique et de Télécommunication.

Les résultats expérimentaux et la simulation sont identiques, ce qui montre notre succès pour la réalisation de l'onduleur et pour la mise en œuvre de la technique PWM.

Mots clés : onduleur monophasé, PWM, convertisseur statique, carte FPGA, validation expérimentale.

الملخص :

في إطار إعداد دبلوم الماستر في هندسة الإلكترونيات ، يعتمد هذا العمل على الإدراك العملي للعاكس أحادي الطور بتركيب مقاعد اختبار تجريبية ، على مستوى ورشة عمل قسم الإلكترونيات والاتصالات.

يرتبط هذا المشروع بدراسة وإنجاز عاكس جهد أحادي الطور. تم تخصيص الفصل الأول للنمذجة الرياضية لهذا العاكس احادي الطور.

في الفصل الثاني قدمنا تقنيات التحكم المختلفة لهذا العاكس مع عمليات المحاكاة الخاصة بهم في بيئة برنامج MATLAB ، من أجل الحصول على أفضل تقنية تحكم. ولقد تحققنا عمليًا من صحة أسلوب التحكم الذي اخترناه في الفصل الثالث على عاكس أحادي الطور أدركناه على مستوى ورشة عمل قسم الإلكترونيات والاتصالات.

النتائج التجريبية والمحاكاة متطابقة ، مما يدل على نجاحنا في تحقيق العاكس وتنفيذ تقنية PWM.

الكلمات الرئيسية

عاكس أحادي الطور PWM ، بطاقة FPGA ، محول ثابت ، التحقق التجريبي.