

LABORATORY MANUAL

Power Electronics



EE-308

Department of Electrical Engineering

INDIAN INSTITUTE OF TECHNOLOGY ROPAR

List of Experiment

S.No.	Name of Experiment	Page No.
1.	To Study about Inductor design	1-6
2.	To study about switch selection and Gate-driver circuit	7-9
3.	To study about Buck converter and validate it in Simulink and PCB	10-13
4.	To study about Boost converter and validate results using Simulink and PCB	14-17
5.	To study about Buck-Boost converter and validate results using Simulink and PCB	18-21
6.	Simulation in Vivado Environment	22-33
7.	To study, simulate, and analyse the operation and performance of 1-Ø full-Bridge Inverter	34-36
8.	Design and simulate 3-Ø Inverter	37-44
9.	To study about full-wave Bridge Rectifier and validate results using Simulink	45-50

EXPERIMENT-1

Aim

To Study about Inductor design.

Components required

Magnetic Core, Bobbin, Copper wire, LCR meter

Theory

For designing any converter magnetic components like inductor and transformers plays a critical role in defining the overall efficiency of the system. For efficient inductor design a proper core with good material and size is to be selected. This selection is based on various parameters such as:

Magnetic Parameters

Saturation flux Density (B_{sat}):

Determines the maximum flux the core can handle before saturation.

Higher B_{sat} allows higher current handling for a given core size.

Permeability(μ) :

Affects inductance for a given number of turns.

High μ reduces the number of turns needed but can lower frequency range.

Core Losses (Hysteresis + Eddy Current Losses)

Should be minimized at the operating frequency to improve efficiency.

Ferrites have low core loss at high frequency. Powdered cores are used for mid frequency operations, nano-crystalline cores are preferred for higher frequency operations

B-H Loop Shape

A narrow hysteresis loop means lower hysteresis losses.

Electrical Parameters

Operating Frequency

High frequency → ferrites (low eddy current loss).

Low frequency → laminated silicon steel or powdered iron (higher B_{sat}).

Required Inductance

Material permeability affects number of turns and winding size.

A material with high μ gives higher inductance per turn.

Current Ripple Handling

Large ripple may push the core into saturation; choose material with higher B_{sat} and proper gap.

Thermal Performance

Core Loss vs. Temperature

Core loss increases with temperature for most materials.

Need materials with stable μ and B_{sat} over an operating temperature range.

Curie Temperature (T_c)

Above T_c , the material loses ferromagnetic properties.

Ferrites: 200–300 °C, powdered iron: 500 °C, steel: higher.

Heat Dissipation Capability

Thermal conductivity of the material affects how quickly heat is removed.

DESIGN CALCULATIONS (for core size and wire selection) :

For designing the core size the area product approach is generally preferred, mathematically it is given as:

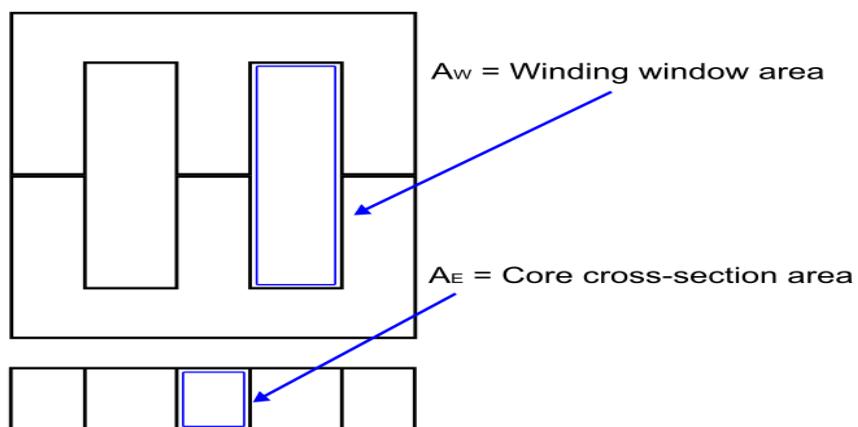


Figure 1 Window and cross-section area of a inductor core

For the shown core in Fig.1 the area product is given as:

$$Ap = \underline{A_w} \cdot \underline{A_E} \dots\dots\dots (1)$$

Where : A_w is the available window area

A_E is the cross-sectional area of core

For any material the linear relation between flux(Φ), inductance(L) and max current(I_{max}) is given as:

$$\Phi = L I_{\max} \dots \dots \dots \quad (2)$$

The above equation of flux can also be written in terms of cross-sectional area of conductor wire (A_c), peak flux (B_{pk}) passing through it as:

$$\Phi = N\Phi = N A_c B_{pk} \dots \dots \quad (3)$$

The cross-sectional area (A_c) can be written as:

$$Ac = \frac{I_{max}}{J_m} \dots \dots \dots \quad (4)$$

Where, I_{max} is the maximum current which the conductor can bear,

J_m denotes the maximum current density of the used conductor.

For determining the number of turns (N) which can be accommodated in the window having an area of A_w , window utilization factor K_u , and conductor with cross-sectional area as A_c , the following equation is used:

$$N = \frac{K_u A_w}{A_c} \dots \dots \dots \quad (5)$$

The maximum energy which can be stored in the inductor can be given as :

From equation (2) and (3) flux(Φ) can be equated and given as:

$$LI_{max} = NA_c B_{pk} = \frac{K_u A_w}{A_c} A_E B_{pk} \dots (7)$$

From the above equation the area product(A_p) is obtained as:

$$A_p = A_w A_E = \frac{LI_{max}^2}{K_u I_m B_{nk}}. \dots \dots \dots (8)$$

The above equation shows the required area product for designing an inductor, after obtaining a specific value of area product a core with matching area product is selected from the product datasheet.

CONCLUSION:

In this experiment we have learned about how to design a inductor for specific application also we have learned about core material selection for various applications.

APPENDIX:

CORE SIZE and WIRE SIZE SELECTION CHART

Properties of Few Ferrite Cores

Cores without air gap	Mean length per turn (mm)	Mean magnetic length l_m (mm)	Core cross-section area A_c (mm ²)	Window area, A_w (mm ²)	Area product A_p (mm ⁴)
Pot Cores					
P18/11	35.6	26	43	27	1161
P26/16	52	37.5	94	53	4982
P30/19	60	45.2	136	75	10200
P36/22	73	53.2	201	101	20301
P42/29	86	68.6	264	181	47784
P66/56	130	123	715	518	370370
EE Cores					
E20/10/5	38	42.8	31	47.8	1481
E25/9/6	51.2	48.8	40	78	3120
E25/13/7	52	57.5	55	87	4785
E30/15/7	56	66.9	59.7	119	7104.3
-E36/18/11	70.6	78	131	141	18471
E42/21/9	77.6	108.5	107	256	27392
E42/21/15	93	97.2	182	256	46592
E42/21/20	99	98	235	256	60160
E65/32/13	150	146.3	266	537	142842
UU Cores					
UU 15	44	48	32	59	1888
UU 21	55	68	55	101	5555
UU 23	64	74	61	136	8296
UU 60	183	184	196	1165	228340
UU 100	29.3	308	645	2914	1879530
Toroids					
T 10	12.8	23.55	6.2	19.6	121.52
T 12	19.2	30.4	12	44.2	530.4
T 16	24.2	38.7	20	78.5	1570
T 20	25.2	47.3	22	95	2090
T 27	34.1	65.94	42	165.1	6934.2
T 32	39.6	73	61	165.1	10071.1
T 45	54.7	114.5	93	615.7	57260.1

Wire Size Table

SWG	Diameter with enamel (mm)	Area of bare conductor (mm ²)	R/km @20°C Ω	Weight (kg/km)
45	0.086	0.003973	4,340	0.0369
44	0.097	0.005189	3,323	0.0481
43	0.109	0.006567	2,626	0.061
42	0.119	0.008107	2,127	0.075
41	0.132	0.009810	1,758	0.0908
40	0.142	0.011675	1,477	0.1079
39	0.152	0.013700	1,258	0.1262
38	0.175	0.018240	945.2	0.1679
37	0.198	0.023430	735.9	0.2202
36	0.218	0.029270	589.1	0.2686
35	0.241	0.035750	482.2	0.3281
34	0.264	0.042890	402	0.3932
33	0.287	0.050670	340.3	0.465
32	0.307	0.059100	291.7	0.5408
31	0.33	0.06818	252.9	0.6245
30	0.351	0.07791	221.3	0.7121
29	0.384	0.09372	184	0.8559
28	0.417	0.11100	155.3	1.014
27	0.462	0.13630	126.5	1.245
26	0.505	0.16420	105	1.499
25	0.561	0.20270	85.1	1.851
24	0.612	0.24520	70.3	2.233
23	0.665	0.29190	59.1	2.655
22	0.77	0.39730	43.4	3.607
21	0.874	0.51890	33.2	4.702
20	0.978	0.65670	26.3	5.939
19	1.082	0.81070	21.3	7.324

SWG	Diameter with enamel (mm)	Area of bare conductor (mm²)	R/km @20°C Ω	Weight (kg/km)
18	1.293	1.16700	14.8	10.537
17	1.501	1.589	10.8	14.313
16	1.709	2.075	8.3	18.678
15	1.92	2.627	6.6	23.64
14	2.129	3.243	5.3	29.15
13	2.441	4.289	4	38.56
12	2.756	5.48	3.1	49.22
11	3.068	6.818	2.5	61
10	3.383	8.302	2.1	74
9	3.8	10.51	1.6	94
8	4.219	12.97	1.3	116

EXPERIMENT – 2

AIM

To study about switch selection (MOSFET) and gate driver circuit, and using it to drive a MOSFET.

COMPONENTS REQUIRED

MOSFET (IRFP450), Driver IC- TLP250, function generator, DC source, Resistor(4K7), Connecting wires.

THEORY:

Switch Selection:

While selecting a power electronics switch for a specific application following parameters must be taken care of:

- (1) Type of switch :** Selection of channel and type (N channel, P channel, enhancement or depletion type) is must as per application,
- (2) Drain to source voltage (V_{ds}):** it will decide the maximum voltage a switch can block.
- (3) Current (I_d) :** It will decide the maximum current a switch can bear for any typical application, exceeding this limit can destroy the switch.
- (4) ON-State Resistance(R_{dsON}):** R_{dsON} should be chosen low to avoid on state drop to avoid efficiency drop.
- (5) Gate to source Voltage(V_{gs}):** A suitable V_{gs} will ensure that where a switch is completely ON or not.
- (6) Operation temperature**
- (7) Maximum operating frequency**

Gate Driver:

A gate drive is a circuit required to drive a power electronics switch efficiently. The various operations performed by gate drive circuit are:

- 1) Voltage level shifting:**
Many MOSFETS/IGBTS/BJTS requires gate-to-source voltages of 10-15 V to turn ON fully while at the same time the logic signals requires only 3-5V this is taken care by driver circuit.
- 2) High Current Drive Capability:**
As the gate terminals of switches are mostly capacitive in nature so a fast switching requires large currents for very small time (ns). This is supplied by driver circuit.

3) Isolation

Isolation ensures safe control without directly connecting control side circuit by high voltage circuits.

4) Protection features

Generally gate driver circuits often have undervoltage lockout, short circuit and D_{sat} protection to protect the power devices.

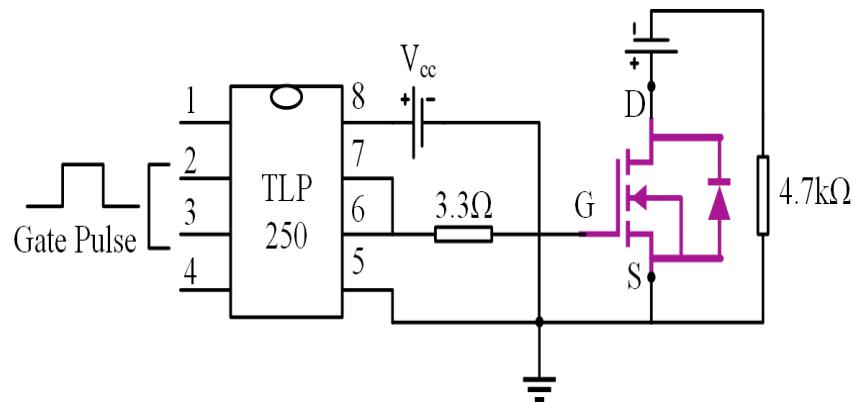


Figure 1 Gate driver circuit with a MOSFET

PROCEDURE:

- 1) Make the connections as shown in the figure 1.
- 2) From a function generator generate a pulse of 15kHz with an amplitude of 5v.
- 3) Connect the output of the function generator to the anode and cathode (PIN-2 & 3) of the TLP-250 IC.
- 4) Connect the V_{cc} (PIN-8) and ground (PIN-5) of the TLP-250 IC to the positive and ground terminals of the dc source respectively.
- 5) Connect the MOSFET (IRFP450) gate terminal to the output of the driver ic (PIN-6 or 7).
- 6) Connect the Drain terminal of the MOSFET to the DC source with a load resistor (4K7) and the source terminal to the ground.
- 7) After turning ON the circuit check for : V_{gs}, V_{ds}, and voltage across the resistor and current flowing through the resistor.

PRECAUTIONS:

- (1) Make sure that the driver IC and the MOSFETS are operated within the rated limits.
- (2) As this Driver IC is optically isolated, it must be checked that signal side ground and power side ground must be isolated.

CONCLUSION:

From this experiment we have studied about selection of switch based on load application and various switches parameters, also we have studied about a gate driver circuit it's necessity and how it is used in a circuit to drive a power electronics switch.

Experiment - 3

Aim: To study about Buck converter and validate results using Simulink and PCB.

Components Required:

DC voltage source, MOSFET (IRFP450), Driver IC (TLP250), Diode, Inductor, Capacitor, Resistor

Theory:

A Buck Converter is a step-down DC-DC converter that reduces input voltage to a lower output voltage while maintaining high efficiency. It operates by switching a transistor (MOSFET) at high frequency and using an inductor-capacitor (LC) filter to smooth the output.

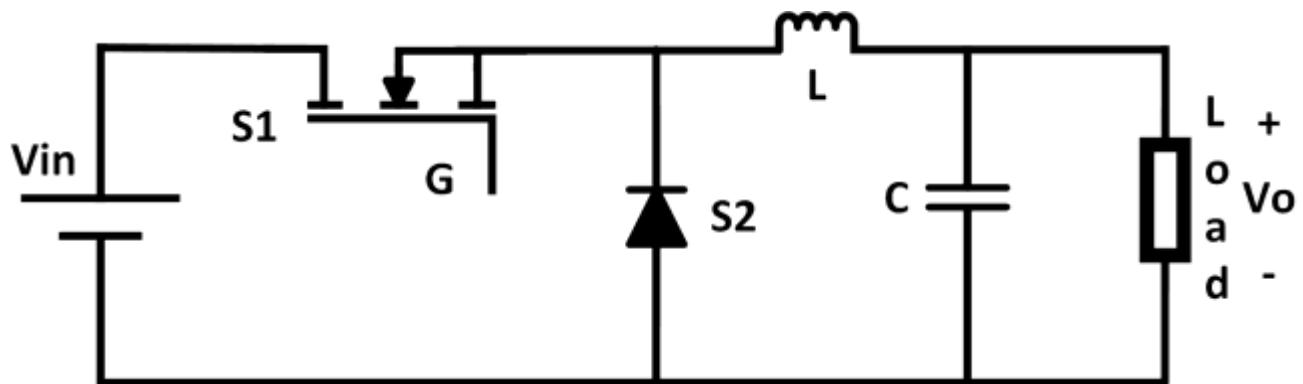


Fig 3.1 : Buck Converter

When the switch (MOSFET) is ON, current flows through the inductor and load. The inductor stores energy in its magnetic field, and the capacitor helps maintain constant output voltage. When the switch is OFF, the inductor resists the sudden drop in current and continues to supply current to the load by releasing stored energy through the diode. By rapidly switching ON and OFF, an average voltage lower than the input voltage is obtained at the output.

Average output voltage is given as

$$V_{out} = D \times V_{in}$$

Where $0 < D < 1$. The duty ratio is the ratio between switch ON time to the total switching period.

Mode of Operation:

- **Continuous Conduction Mode (CCM):** Inductor current never goes to zero. In CCM, the inductor current stays positive throughout the switching period. To stay in CCM, the average output current must be high enough.

CCM Boundary Condition: The converter will operate in CCM if the load current I_{load} is greater than the critical (boundary) current I_{crit}

$$I_{\text{load}} > I_{\text{crit}} = \frac{V_o \cdot (1 - D)}{2 \cdot L \cdot f_s}$$

- **Discontinuous Conduction Mode (DCM):** Inductor current drops to zero during part of the cycle. This happens when the load current is too low or the inductance is too small than critical value.

The converter enters DCM if:

$$I_{\text{load}} < \frac{V_o \cdot (1 - D)}{2 \cdot L \cdot f_s}$$

Value of inductor is calculated based on amount of current ripple Δi_L that is allow by designer for give application.

The value of inductor is given as

$$L = \frac{V_{in} \cdot D}{\Delta I_L \cdot f_s} \quad (\text{during DTs period})$$

Value of capacitor is calculate based on amount of voltage ripple ΔV_o that is allow by designer for give application.

The value of capacitor is given as

$$C = \frac{I_o \cdot D}{\Delta V_o \cdot f_s}$$

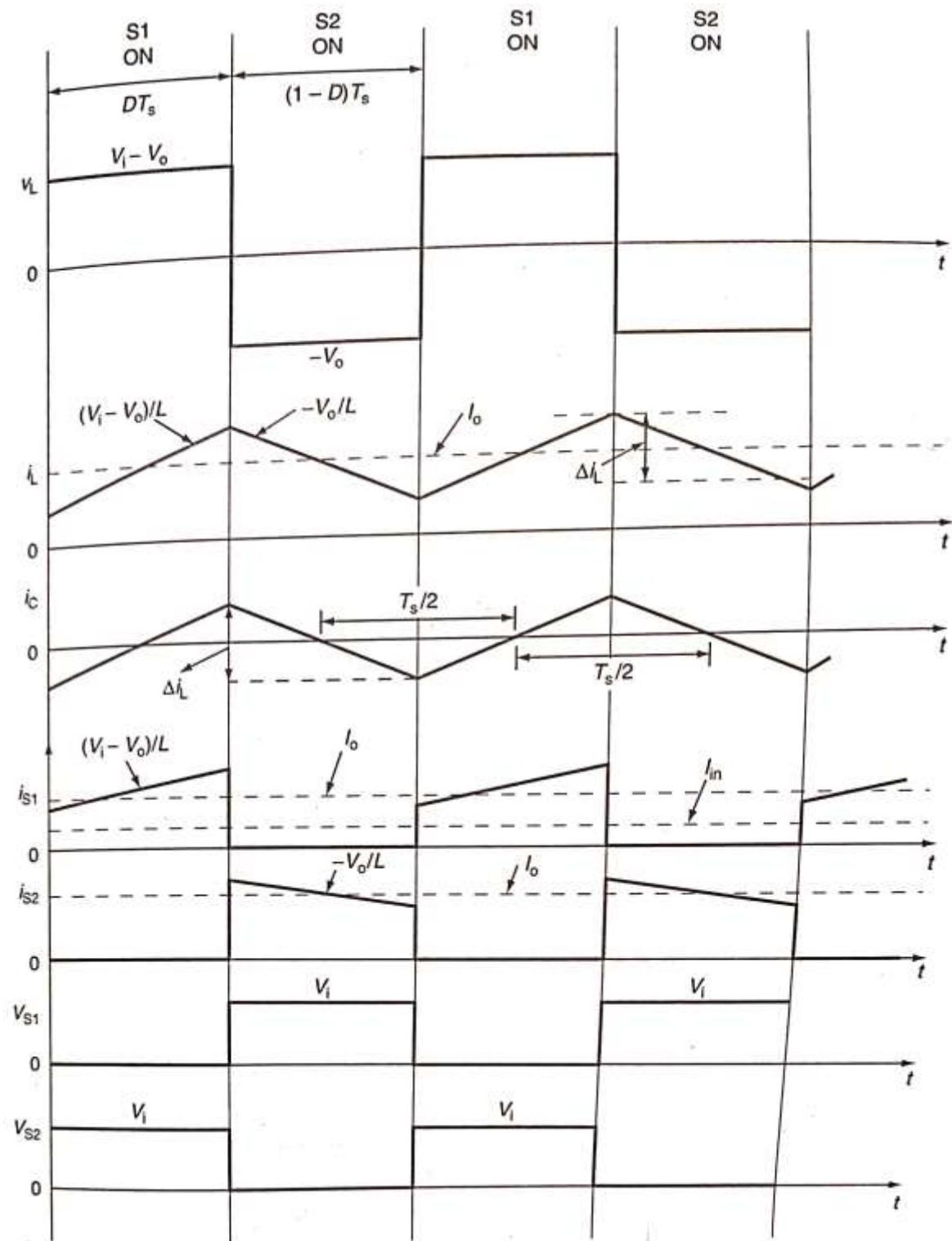


Fig 3.2 : Waveforms of buck converter

Procedure

For Matlab Simulink

1. Create a new Simulink model
2. Place DC Source, inductor, MOSFET, diode, capacitor, load resistor according to circuit diagram and connect them.
3. Add a PWM or pulse generator with frequency 20 kHz and duty $D \approx 0.33$ to drive the MOSFET gate.
4. Insert voltage and current measurement blocks to measure and observe input/output voltage, current waveforms.
5. Run the simulation and record measurements. also repeat for different test cases (vary duty ratio, load change).

For Hardware

1. Make connection for Buck converter on generalize DC-DC converter PCB.
2. Connect the input DC supply to PCB.
3. Provide the PWM drive: use a function generator/ DSP/FPGA based microcontroller to produce PWM parameters as simulation (20 kHz, $D \approx 0.33$). use a gate driver for MOSFET gate-level shifting and protection.
4. Place oscilloscope probes: ground clip to common star ground point; measure output voltage across load, inductor current (current probe), and all other waveforms across different components.
5. Record steady-state measurements and repeat for different test cases (vary duty ratio, load change)

Discussion

In this section you analyzed the buck converter, its design aspects with simulation as well as with hardware implementation and discuss issues you facing.

Conclusions

Here you summarized the essential aspects and findings of our work and analysis.

Experiment – 4

Aim : To study about Boost converter and validate results using Simulink and PCB.

Components Required :

DC voltage source, MOSFET (IRFP450), Driver IC (TLP250), Diode, Inductor, Capacitor, Resistor

Theory:

A boost converter is a type of DC–DC converter that step up (boosts) an input voltage to a higher output voltage. It operates by temporarily storing energy in an inductor during the ON period of the switch (MOSFET) and releasing it to the load through the diode during the OFF period. The duty ratio D controls the output voltage, where $0 < D < 1$.

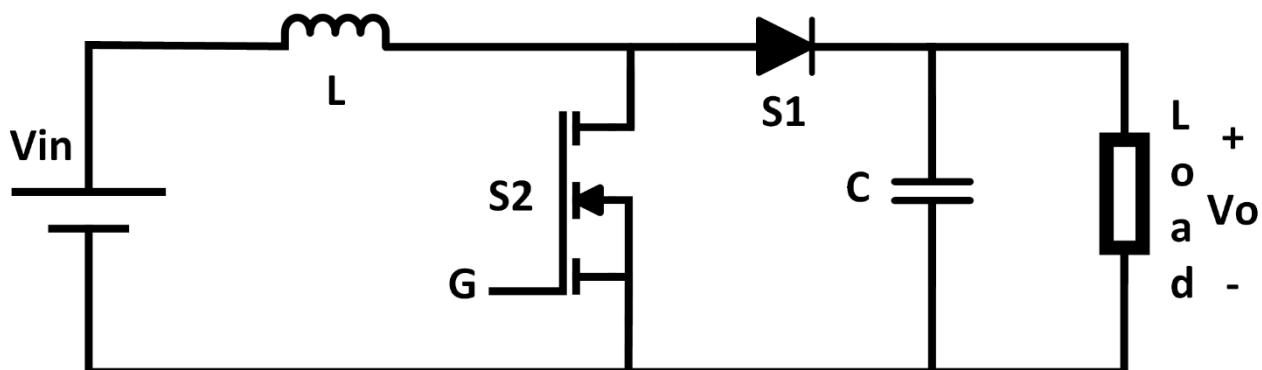


Fig 4.1 : Boost Converter

When MOSFET is ON and Diode is OFF, Inductor stores energy from the source, The diode is reverse-biased so the load is powered by the capacitor. When MOSFET is OFF and Diode is ON, inductor is connected to the load via the diode. The Source and inductor supply power to the load and capacitor, increasing the output voltage above the input voltage.

Average output voltage is given as

$$V_o = \frac{V_{in}}{(1-D)}$$

Mode of Operation:

- **Continuous Conduction Mode (CCM):** Inductor current never goes to zero. In CCM, the inductor current stays positive throughout the switching period. To stay in CCM, the average output current must be high enough.

CCM Boundary Condition: The converter will operate in CCM if the load current I_{load} is greater than the critical (boundary) current I_{crit}

$$I_{load} > I_{cri} = \frac{V_o \cdot D \cdot (1 - D)^2}{2 \cdot L \cdot f_s}$$

- **Discontinuous Conduction Mode (DCM):** Inductor current drops to zero during part of the cycle. This happens when the load current is too low or the inductance is too small than critical value.

The converter enters DCM if:

$$I_{load} < \frac{V_o \cdot D \cdot (1 - D)^2}{2 \cdot L \cdot f_s}$$

Value of inductor is calculated based on amount of current ripple Δi_L that is allow by designer for give application.

The value of inductor is given as $L = \frac{V_s \cdot D}{L \cdot f_s}$ (during DTs period)

Value of capacitor is calculate based on amount of voltage ripple ΔV_o that is allow by designer for given application.

The value of capacitor is given as $C = \frac{I_o \cdot D}{\Delta V_o \cdot f_s}$.

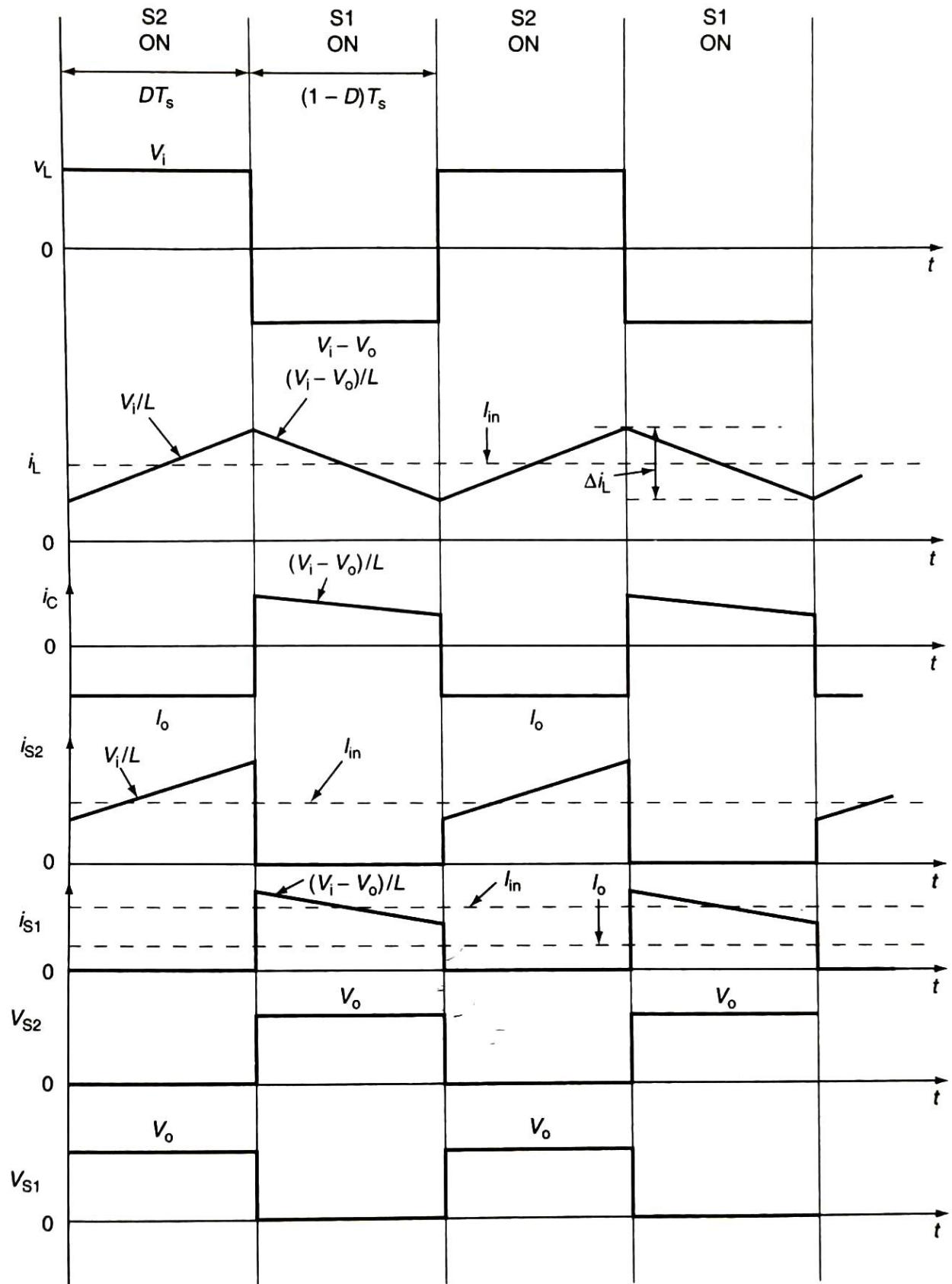


Fig 4.2 : Waveforms of Boost Converter

Procedure:

For Matlab Simulink

1. Create a new Simulink model
2. Place a DC Source, Inductor, MOSFET, Diode, Capacitor, Load-Resistor according to circuit diagram and connect them.
3. Add a PWM or pulse generator with frequency 20kHz and duty D = 0.33 to drive the MOSFET gate.
4. Insert voltage and current measurement blocks to measure and observe input/output voltage, current waveforms.
5. Run the simulation and record measurements, also repeat for different test cases (vary duty ratio, vary load).

For Hardware

1. Make connection for Boost converter on generalize DC-DC converter PCB.
2. Connect the input DC supply to PCB.
3. Provide the PWM drive: use a function generator/ DSP/FPGA based microcontroller to produce PWM parameters as simulation (20 kHz, D = 0.33). use a gate driver for MOSFET gate-level shifting and protection.
4. Place oscilloscope probes: ground clip to common star ground point; measure output voltage across load, inductor current (current probe), and all other waveforms across different components.
5. Record steady-state measurements and repeat for different test cases (vary duty ratio, load change)

Discussion:

In this experiment you analysed boost converter, its design aspects with simulation as well as with hardware implementation and discuss issues you faced.

Conclusions:

Here you summarized the essential aspects and findings of your work and analysis.

Experiment – 5

Aim : To study about Buck-Boost converter and validate results using Simulink and PCB.

Components Required :

DC voltage source, MOSFET (IRFP450), Driver IC (TLP250), Diode, Inductor, Capacitor, Resistor

Theory

A buck-boost converter is a type of DC-DC converter that can step up (boost) or step down (buck) an input voltage to a desired output voltage depending on operating duty ratio D. It combines the principles of both buck and boost converters and is especially useful when the input voltage can vary above or below the desired output voltage. The duty ratio D can take on values between 0 And 1. If D is between 0 and 0.5 then it results in buck operation. If D is between 0.5 and 1 then it results in boost operation.

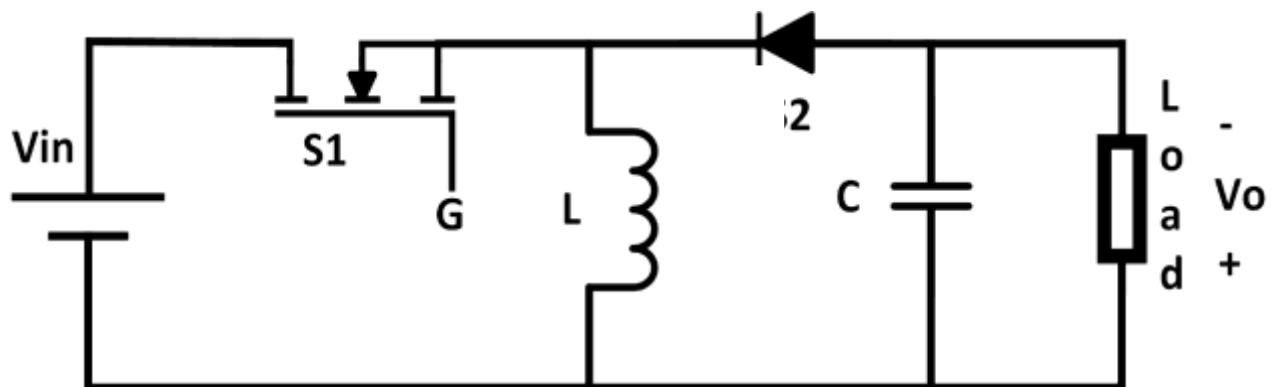


Fig 5.1 : Buck Boost Converter

When MOSFET is ON and Diode is OFF, Inductor stores energy from the source, The diode is reverse-biased, so the load is powered by the capacitor. When MOSFET is OFF and Diode is ON, inductor is connected to the load via the diode. The inductor releases stored energy to the load and capacitor. Current flows from the inductor to the output. The output voltage polarity is negative with respect to circuit ground.

Average output voltage is given as

$$V_o = \frac{-D}{(1-D)} V_{in}$$

Mode of Operation:

- **Continuous Conduction Mode (CCM):** Inductor current never goes to zero. In CCM, the inductor current stays positive throughout the switching period. To stay in CCM, the average output current must be high enough.

CCM Boundary Condition: The converter will operate in CCM if the load current I_{load} is greater than the critical (boundary) current I_{crit}

$$I_{\text{load}} > I_{\text{crit}} = \frac{V_o \cdot (1 - D)^2}{2 \cdot L \cdot f_s}$$

- **Discontinuous Conduction Mode (DCM):** Inductor current drops to zero during part of the cycle. This happens when the load current is too low or the inductance is too small than critical value.

The converter enters DCM if:

$$I_{\text{load}} < \frac{V_o (1 - D)^2}{2 L f_s}$$

Value of inductor is calculated based on amount of current ripple Δi_L that is allow by designer for give application.

The value of inductor is given as $L = \frac{V_{in} \cdot D}{\Delta i_L \cdot f_s}$ (during DTs period)

Value of capacitor is calculate based on amount of voltage ripple ΔV_o that is allow by designer for given application.

The value of capacitor is given as $C = \frac{I_o \cdot D}{\Delta V_o \cdot f_s}$

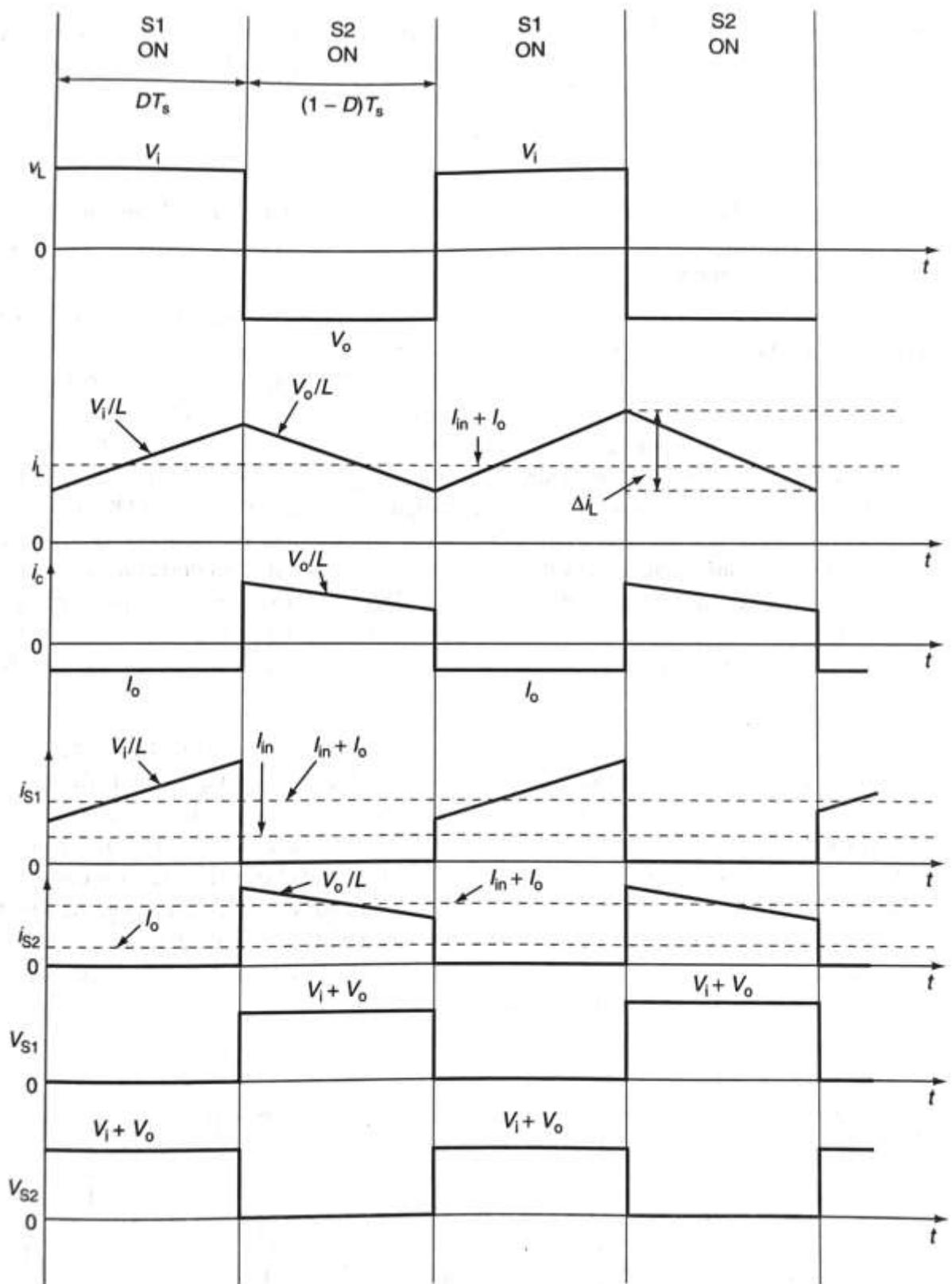


Fig 5.2 : Waveforms of Buck Boost Converter

Procedure

For Matlab Simulink

1. Create a new Simulink model
2. Place DC Source, inductor, MOSFET, diode, capacitor, load resistor according to circuit diagram and connect them.
3. Add a PWM or pulse generator with frequency 20 kHz and duty $D \approx 0.33$ to drive the MOSFET gate.
4. Insert voltage and current measurement blocks to measure and observe input/output voltage, current waveforms.
5. Run the simulation and record measurements. also repeat for different test cases (vary duty ratio, load change).

For Hardware

1. Make connection for Buck Boost converter on generalize DC-DC converter PCB.
2. Connect the input DC supply to PCB.
3. Provide the PWM drive: use a function generator/ DSP/FPGA based microcontroller to produce PWM parameters as simulation (20 kHz, $D \approx 0.33$). use a gate driver for MOSFET gate-level shifting and protection.
4. Place oscilloscope probes: ground clip to common star ground point; measure output voltage across load, inductor current (current probe), and all other waveforms across different components.
5. Record steady-state measurements and repeat for different test cases (vary duty ratio, load change)

Discussion

In this section you analysed the buck boost converter, its design aspects with simulation as well as with hardware implementation and discuss issues you facing.

Conclusions

Here you summarized the essential aspects and findings of our work and analysis.

Experiment No. 06

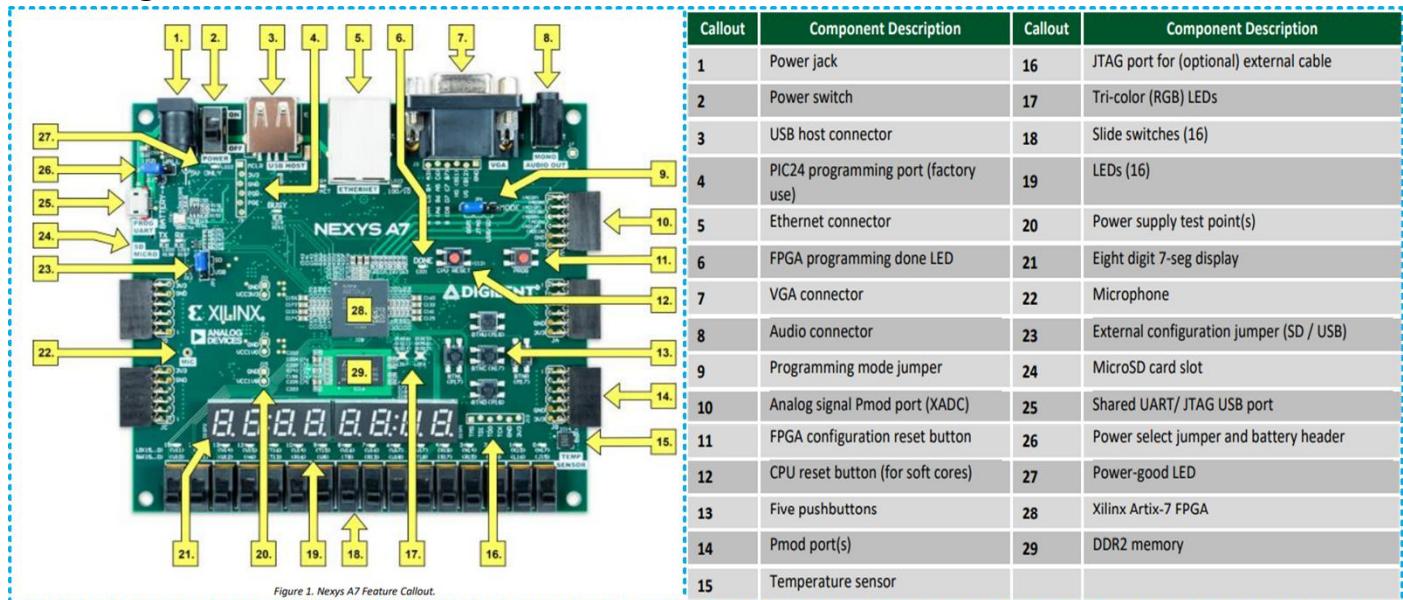
Title of The Experiment: Simulation in Vivado Environment

Aim of The Experiment:

- To familiarize with the Nexys-A7 100t FPGA development board.
- To create and configure a project file for FPGA implementation.
- Code Burning into FPGA board

Familiarization with the Nexys-A7 100t FPGA Development Board:

PIN configuration



Create and configure a project file for FPGA implementation:

To create a project file please follow the step-by-step instructions.

Code 1: Glow an LED using Logic Gate

Step 1

Quick Start
Create Project > **Click on Create Project**
Open Project >
Open Example Project >

Step 2

Create a New Vivado Project
This wizard will guide you through the creation of a new project.
To create a Vivado project you will need to provide a name and a location for your project files. Next, you will specify the type of flow you'll be working with. Finally, you will specify your project sources and choose a default part.

Step 3

New Project
Project Name
Enter a name for your project and specify a directory where the project data files will be stored.

Project name: Logic_Gates **Enter Project Name**
Project location: E:/PCAD LAB **Choose the location**
 Create project subdirectory
Project will be created at: E:/PCAD LAB/Logic_Gates

Step 4

New Project
Project Type
Specify the type of project to create.

Tick these
 RTL Project
 Do not specify sources at this time
 Project is an extensible Vitis platform

Click on Next

Step 5

New Project
Default Part
Choose a default Xilinx part or board for your project.

Click on Boards

Parts | **Boards**
To fetch the latest available boards from git repository, click on 'Refresh' button. Dismiss
Reset All Filters
Vendor: All Name: All Board Rev: Latest

Display Name	Preview	Status	Vendor	File Version	Part
Nexys4		Installed	digilentinc.com	1.1	
Nexys Video		Installed	digilentinc.com	1.1	
Nexys A7-100T Select This		Installed	digilentinc.com	1.3	xc7a100tcs324-1
Nexys A7-50T		Installed	digilentinc.com	1.3	xc7a50tcs324-1L

Refresh Catalog was last updated on 04/08/2025 6:57:48 AM

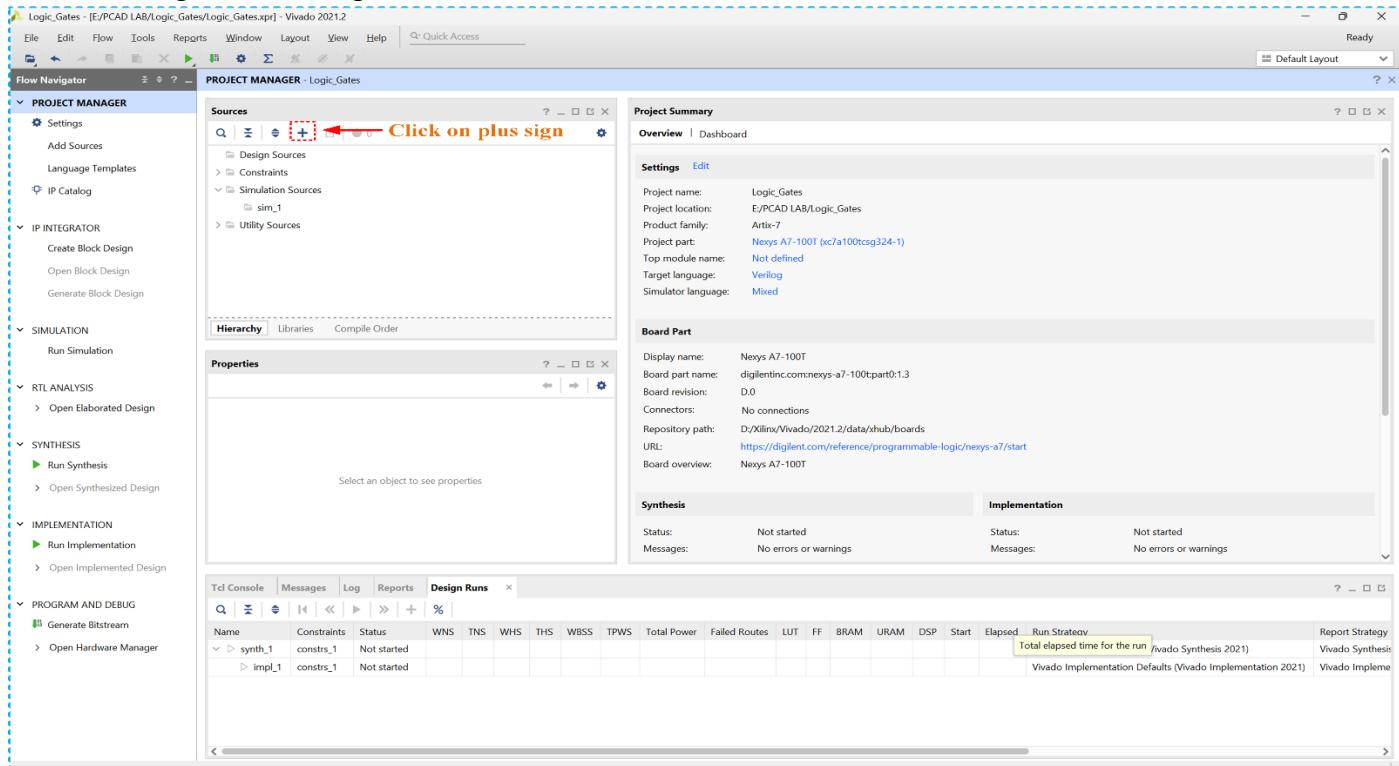
Click on Next

Step 6

New Project Summary
A new RTL project named 'Logic_Gates' will be created.
The default part and product family for the new project:
Default Board: Nexys A7-100T
Default Part: xc7a100tcs324-1
Family: Artix-7
Package: csg324
Speed Grade: -1

Click on Finish

The following window is opened.



Add Sources

Add Sources

Add or Create Design Sources

Specify HDL, netlist, Block Design, and IP files, or directories containing those file types to add to your project. Create a new source file on disk and add it to your project.

Create Source File

Create a new source file and add it to your project.

File type: Verilog

File name: logic

File location: <Local to Project>

Click on this

Click on this

Give file name

Click on this

The screenshot shows the E/PCAD software interface. On the left, a 'Define Module' dialog is open, prompting the user to define a module and specify I/O Ports. It includes fields for 'Module name' (set to 'logic') and an 'I/O Port Definitions' table. The table has columns for 'Port Name', 'Direction', 'Bus', 'MSB', and 'LSB'. A row is present with 'input' as the direction and '0' for both MSB and LSB. A red arrow points to the 'OK' button at the bottom of the dialog. On the right, another 'Define Module' dialog is shown, asking if the user is sure they want to use the current values. It has 'Yes' and 'No' buttons, with 'Yes' highlighted by a red box and a red arrow pointing to it. Below these dialogs is a 'Project Summary' window titled 'logic.v' containing Verilog code for a logic module.

```

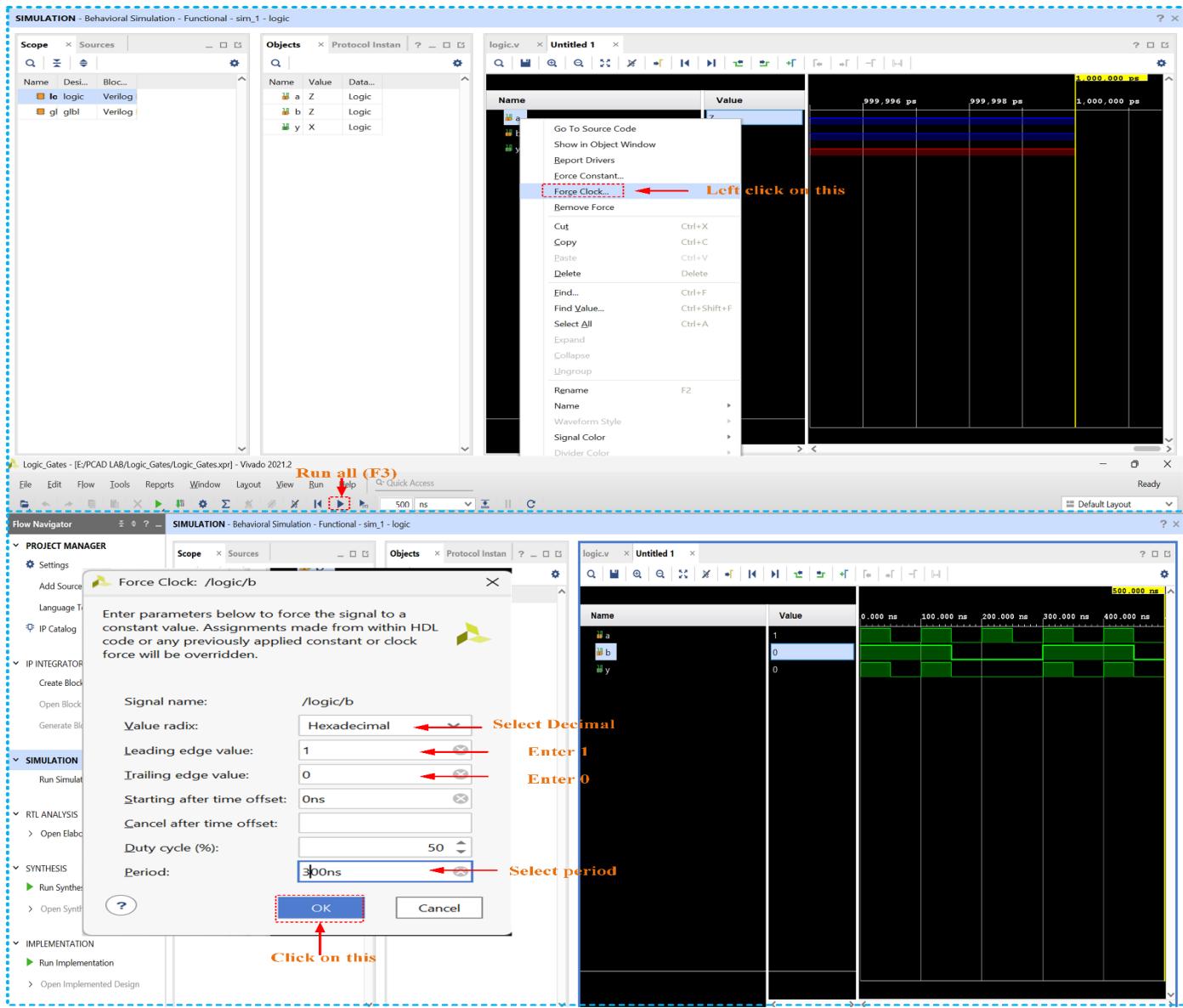
1 `timescale 1ns / 1ps
2 //////////////////////////////////////////////////////////////////
3 // Company:
4 // Engineer:
5 //
6 // Create Date: 11.08.2025 13:29:09
7 // Design Name:
8 // Module Name: logic
9 // Project Name:
10 // Target Devices:
11 // Tool Versions:
12 // Description:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 //////////////////////////////////////////////////////////////////
21
22
23 module logic(
24   input wire a,
25   input wire b,
26   output wire y
27 );
28   assign y = a & b;
29 endmodule

```

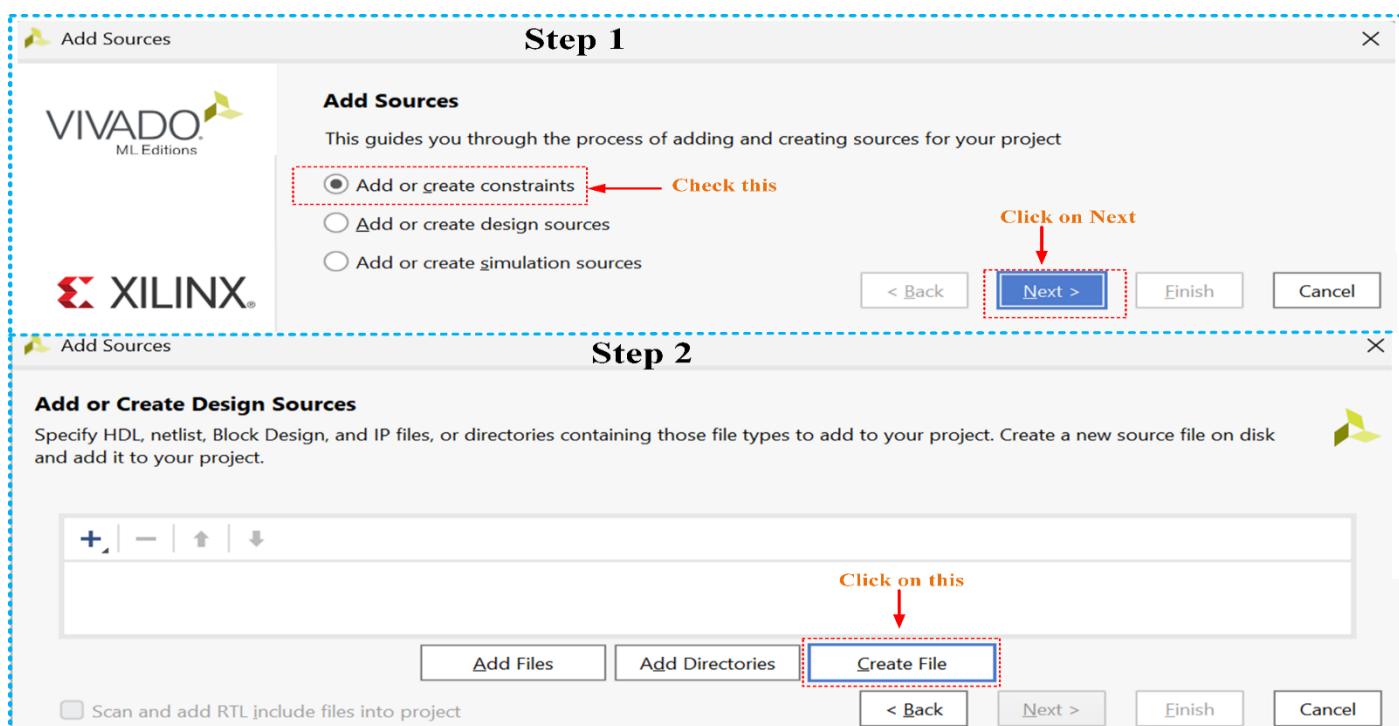
This window will open and write code here

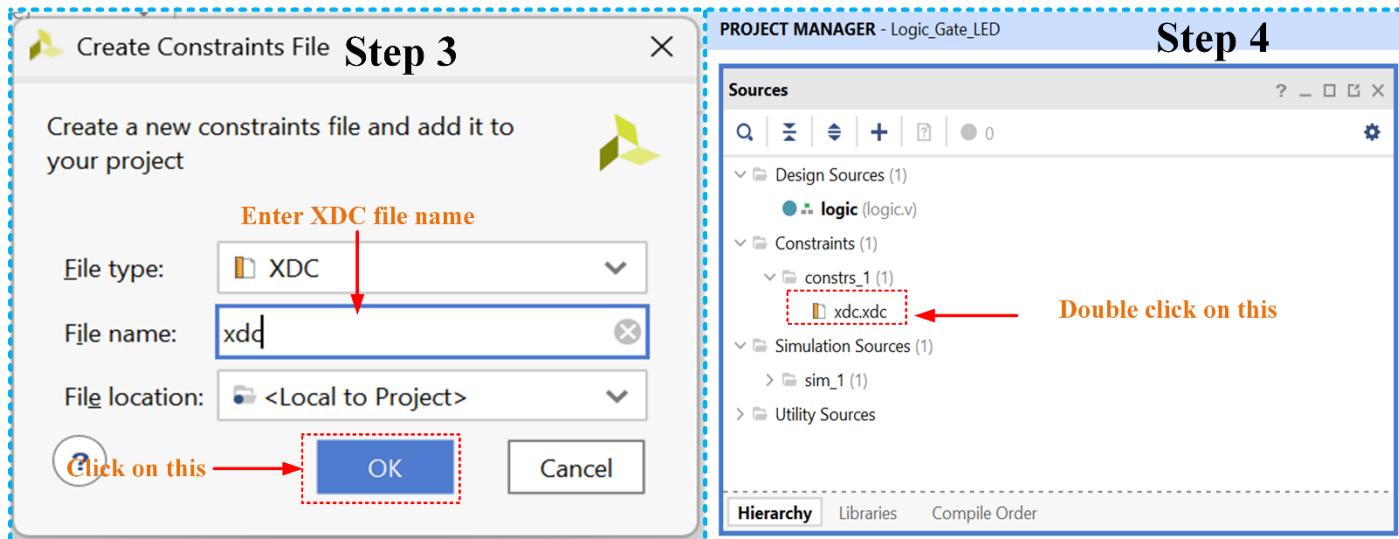
Step for behavioral Simulation:

Then Go to Simulation and Click on behavioral Simulation. Then below window will open. Where left click on the clk and click on forced clock and click on ok.



Steps for XDC file creation:





Step 5

Project Summary logic.v

E:/PCAD LAB/Logic_Gate_LED/Logic_Gate_LED.srsc/sources_1/new/logic.v

```

1 `timescale 1ns / 1ps
2 ///////////////////////////////////////////////////////////////////
3 // Company:
4 // Engineer:
5 //
6 // Create Date: 11.08.2025 13:29:09
7 // Design Name:
8 // Module Name: logic
9 // Project Name:
10 // Target Devices:
11 // Tool Versions:
12 // Description:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 ///////////////////////////////////////////////////////////////////
21
22 module logic(
23   input wire a,
24   input wire b,
25   output wire y
26 );
27   assign y = a & b;
28 endmodule
30 
```

Project Summary logic.v xdc.xdc

E:/PCAD LAB/Logic_Gate_LED/Logic_Gate_LED.srsc/constrs_1/new/xdcx.xdc

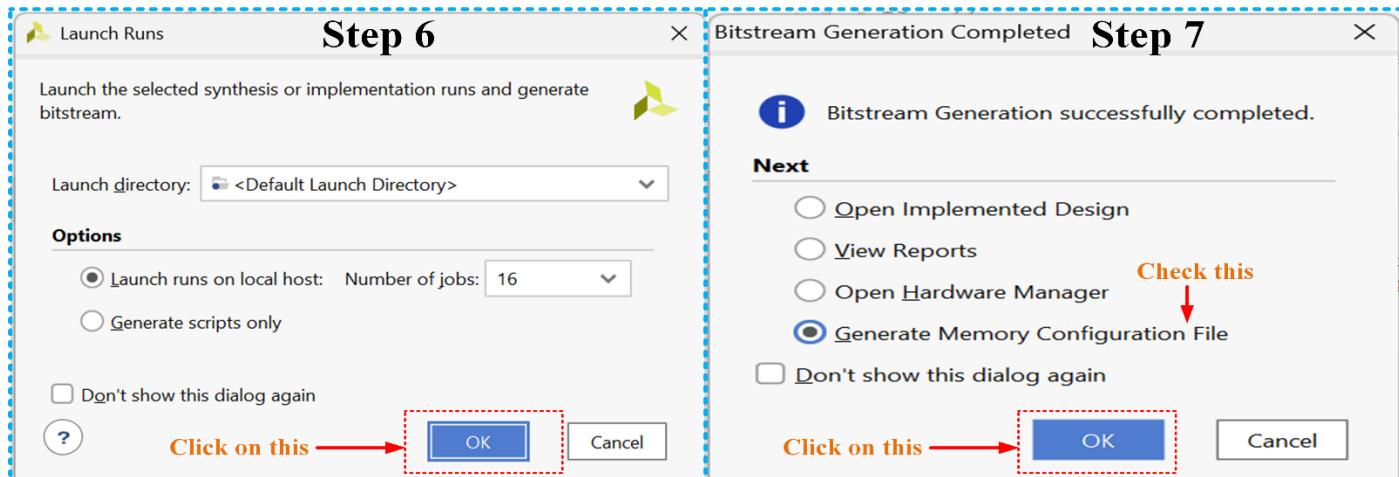
```

1 ## This file is a general .xdc for the Nexys A7-100T
2 ## - uncomment the lines corresponding to used pins
3 ## - rename the used ports (in each line, after get_ports) according to the top level signal names in the project
4 ## Note: As the Nexys 4 DDR was rebranded to the Nexys A7 with no substantial changes, this XDC file will also work for
5 ## the A7
6 ## Clock signal
7 #set_property -dict { PACKAGE_PIN E3 IOSTANDARD LVCMS33 } {get_ports { CLK100MHZ }; #IO_L12P_T1_MRCC_35 Sch=clk100}
8 #create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} {get_ports {CLK100MHZ}}
9
10 ##Switches
11 #set_property -dict { PACKAGE_PIN J15 IOSTANDARD LVCMS33 } {get_ports { a }}; #IO_L24N_T3_RS0_15 Sch=sw[0]
12 #set_property -dict { PACKAGE_PIN K16 IOSTANDARD LVCMS33 } {get_ports { b }}; #IO_J3N_T0_DQS_EMCLK_14 Sch=sw[1]
13 #set_property -dict { PACKAGE_PIN M13 IOSTANDARD LVCMS33 } {get_ports { SW[2] }}; #IO_L6N_T0_D08_VREF_14 Sch=sw[2]
14 #set_property -dict { PACKAGE_PIN R15 IOSTANDARD LVCMS33 } {get_ports { SW[3] }}; #IO_L13N_T2_MRCC_14 Sch=sw[3]
15 #set_property -dict { PACKAGE_PIN R17 IOSTANDARD LVCMS33 } {get_ports { SW[4] }}; #IO_L12N_T1_MRCC_14 Sch=sw[4]
16 #set_property -dict { PACKAGE_PIN T18 IOSTANDARD LVCMS33 } {get_ports { SW[5] }}; #IO_L7N_T1_D10_14 Sch=sw[5]
17 #set_property -dict { PACKAGE_PIN U18 IOSTANDARD LVCMS33 } {get_ports { SW[6] }}; #IO_L17N_T2_A13_D29_14 Sch=sw[6]
18 #set_property -dict { PACKAGE_PIN R13 IOSTANDARD LVCMS33 } {get_ports { SW[7] }}; #IO_L5N_T0_D07_14 Sch=sw[7]
19 #set_property -dict { PACKAGE_PIN T8 IOSTANDARD LVCMS18 } {get_ports { SW[8] }}; #IO_L24N_T3_34 Sch=sw[8]
20 #set_property -dict { PACKAGE_PIN U8 IOSTANDARD LVCMS18 } {get_ports { SW[9] }}; #IO_25_34 Sch=sw[9]
21 #set_property -dict { PACKAGE_PIN R16 IOSTANDARD LVCMS33 } {get_ports { SW[10] }}; #IO_L15P_T2_DQS_RDWR_B_14 Sch=sw[10]
22 #set_property -dict { PACKAGE_PIN T13 IOSTANDARD LVCMS33 } {get_ports { SW[11] }}; #IO_L23P_T3_A03_D19_14 Sch=sw[11]
23 #set_property -dict { PACKAGE_PIN H6 IOSTANDARD LVCMS33 } {get_ports { SW[12] }}; #IO_L24P_T3_35 Sch=sw[12]
24 #set_property -dict { PACKAGE_PIN U12 IOSTANDARD LVCMS33 } {get_ports { SW[13] }}; #IO_L20P_T3_A08_D24_14 Sch=sw[13]
25 #set_property -dict { PACKAGE_PIN U11 IOSTANDARD LVCMS33 } {get_ports { SW[14] }}; #IO_L19N_T3_A09_D25_VREF_14 Sch=sw[14]
26 #set_property -dict { PACKAGE_PIN V10 IOSTANDARD LVCMS33 } {get_ports { SW[15] }}; #IO_L21P_T3_DQS_14 Sch=sw[15]
27
28 ## LEDs
29 #set_property -dict { PACKAGE_PIN H17 IOSTANDARD LVCMS33 } {get_ports { y }}; #IO_L18P_T2_A24_15 Sch=led[0]
30 #set_property -dict { PACKAGE_PIN K15 IOSTANDARD LVCMS33 } {get_ports { LED[1] }}; #IO_L24P_T3_RS1_15 Sch=led[1]

```

Steps for Generating Bitstream:

Then click on ‘Generate Bitstream’ and follow the step 4, 5.



Code burning into FPGA board:

Step 8

Create a configuration file to program the device

Format: MCS

Memory Part: Custom

Filename:

Options Filter

Interface: Manufacturer: All Type: All Density (Mb): All Width: All

Select Configuration Memory Part

Search:

Name	Part	Manufact...	Alias	Family	Type	Density (...	Width
s25fl032p-spi-x1_x2_x4	s25fl032p	Spansion		s25flxxp	spi	32	x1, ^
s25fl064i-spi-x1_x2_x4	s25fl064i	Spansion		s25flxxd	spi	64	x1,
s25fl064p-spi-x1_x2_x4	s25fl064p	Spansion		s25flxxp	spi	64	x1,
s25fl116k-spi-x1_x2_x4	s25fl116k	Spansion		s25fl11	spi	16	x1,
s25fl128l-spi-x1_x2_x4	s25fl128l	Spansion		s25flxxd	spi	128	x1,
s25fl128sxxxxx0-spi-x1_x2_x4	s25fl128sxxxxx0	Spansion	s25fl127s-spi-x1_x2_x4	s25flxxs	spi	128	x1, v

Commands:

OK Cancel

Step 9

Create a configuration file to program the device

Format: MCS

Memory Part: s25fl128sxxxxx0-spi-x1_x2_x4

Custom Memory Size (MB): 16

Filename: E:/PCAD LAB/Logic_Gates/logic.mcs

Options

Interface: SPIx4

Load bitstream files Daisy chain configuration file

Start address: 00000000 Direction: up Bitfile:

Load data files

Start address: 00000000 Direction: up Datafile:

Write checksum

Disable bit swapping

Overwrite

Command: write_cfm -format mcs -size 16 -interface SPIx4 -file "E:/PCAD LAB/Logic_Gates/logic.mcs"

OK Cancel

Step 10

Create a configuration file to program the device

Format: MCS

Memory Part: s25fl128sxxxxx0-spi-x1_x2_x4

Custom Memory Size (MB): 16

Filename: E:/shem.mcs

Options

Interface: SPIx4

Load bitstream files Daisy chain configuration file

Start address: 00000000 Direction: up Bitfile:

Specify Datafile Filename

Look in:

File name: logic.bit

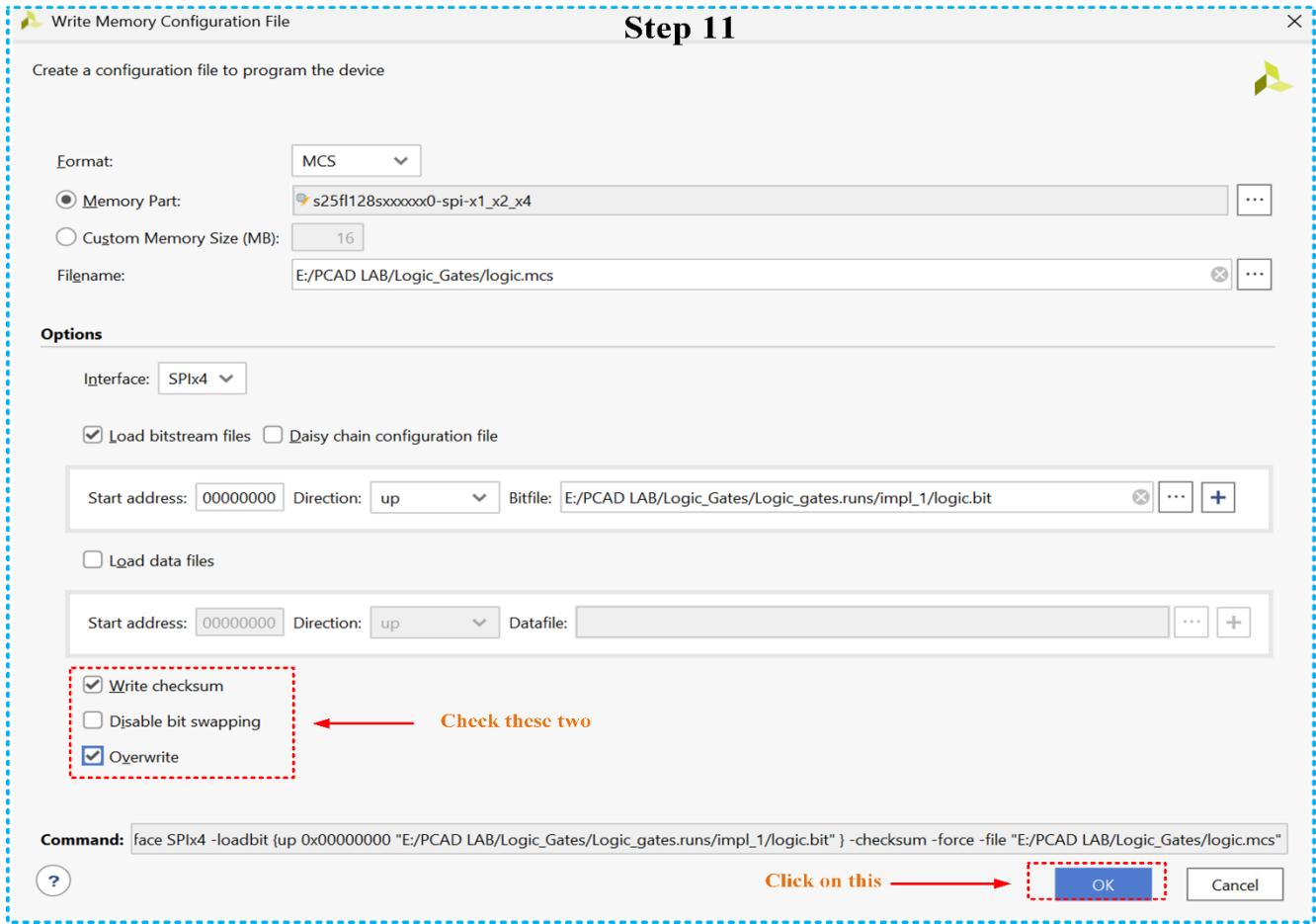
Recent Directories: E:/PCAD LAB/Logic_Gates/Logic_gates...

File Preview

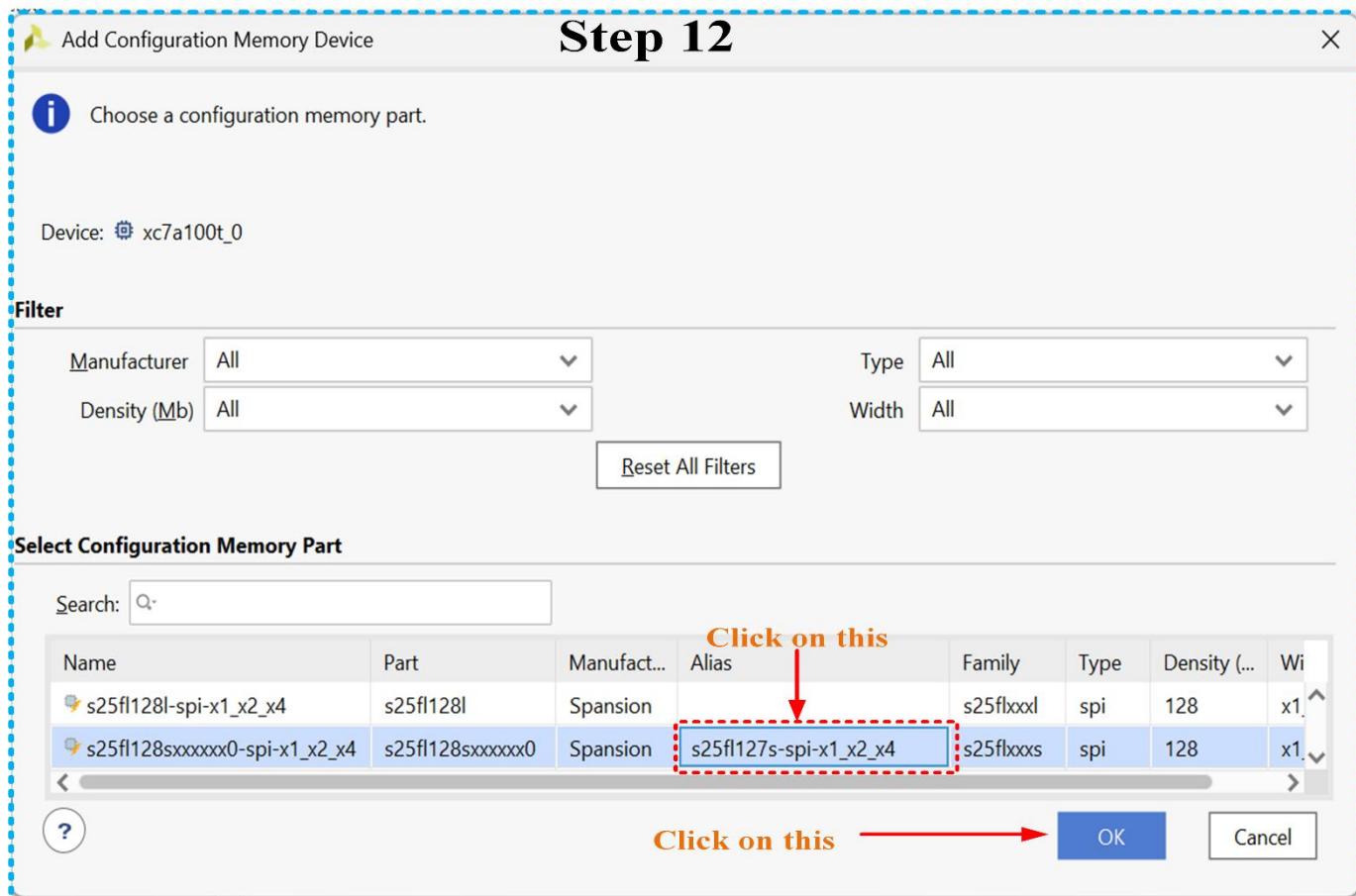
File name: logic.bit

Files of type: Bitstream Files (.bit, .bin, .rbt)

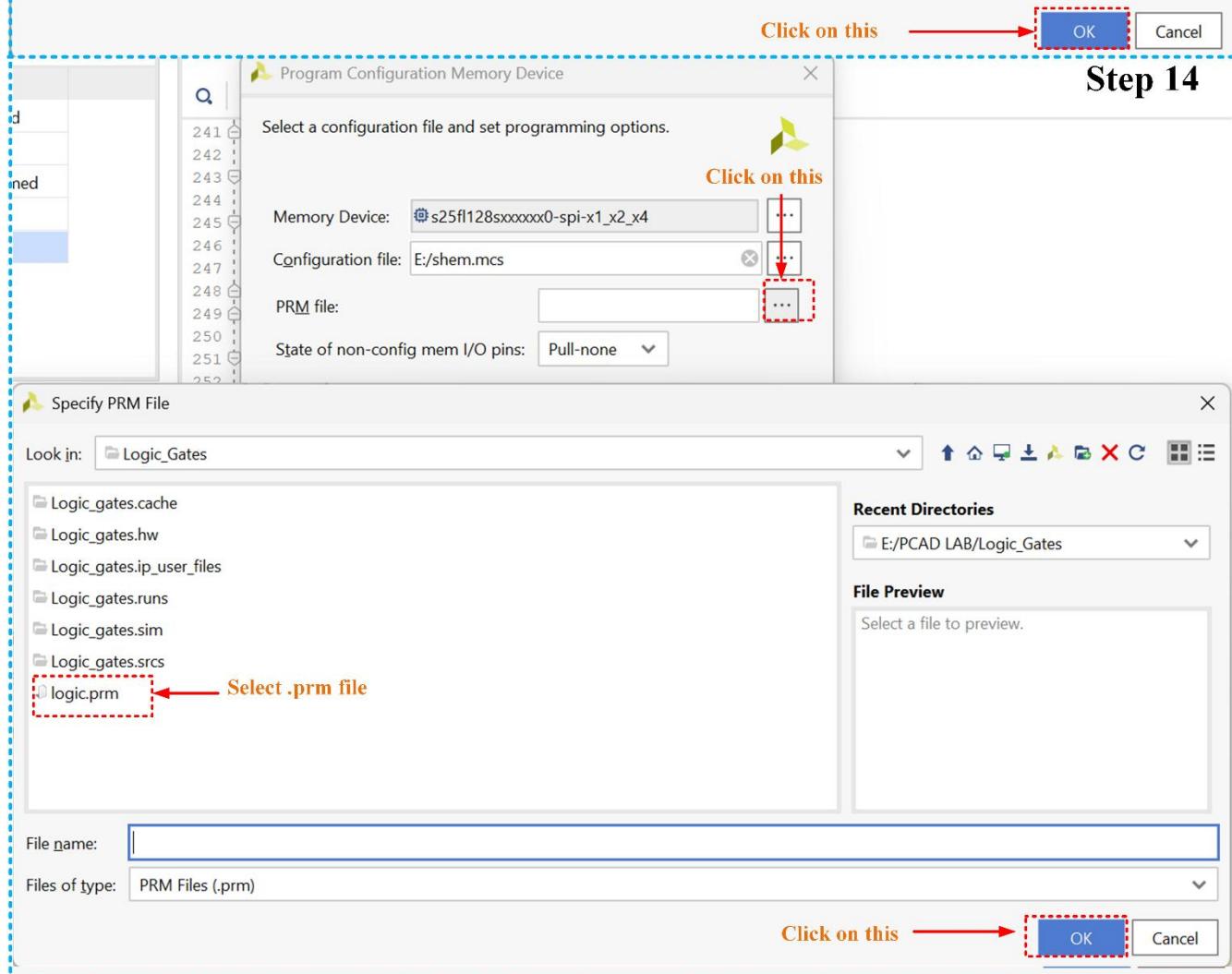
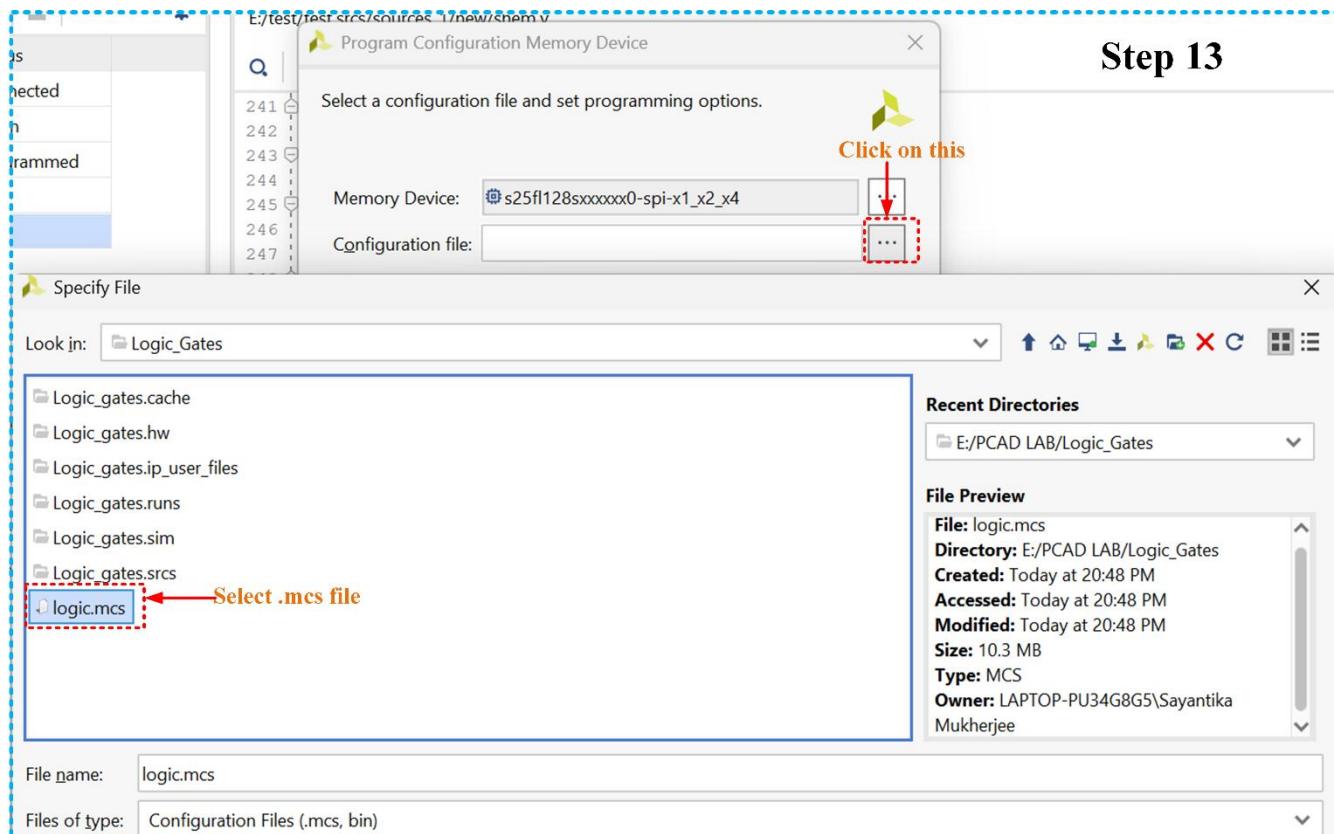
OK Cancel

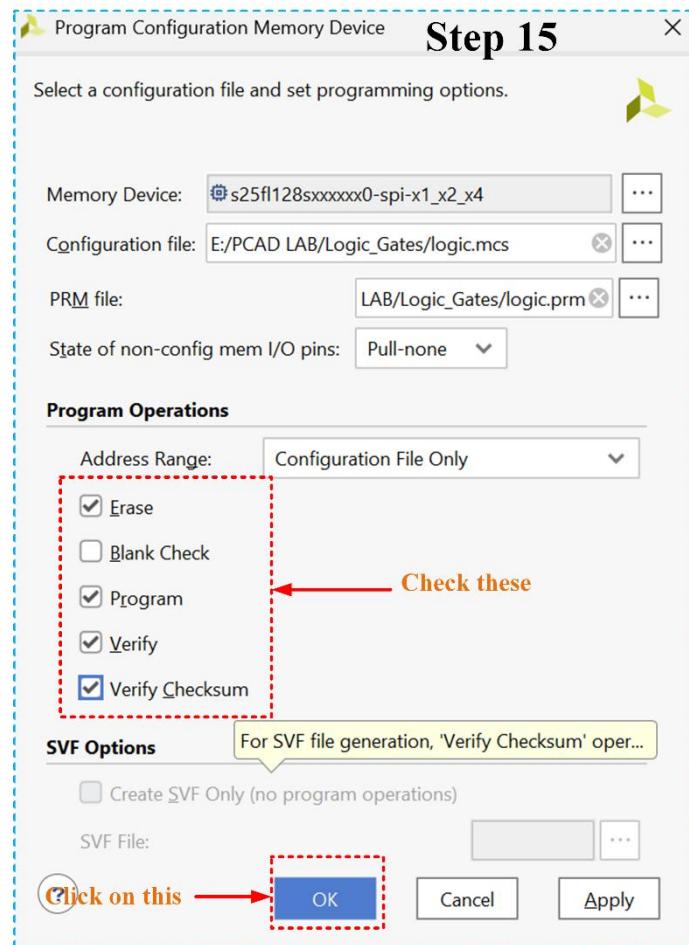


After this now open ‘Hardware Manager’ and click on ‘Open Target’. Then click on ‘Auto Connect’. Now click on ‘Programme Device’ and select ‘xc7a100t_0’ and again select the same .bit file and click on programme. Now click ‘Add Configuration Memory Device’ and select ‘xc7a100t_0’.



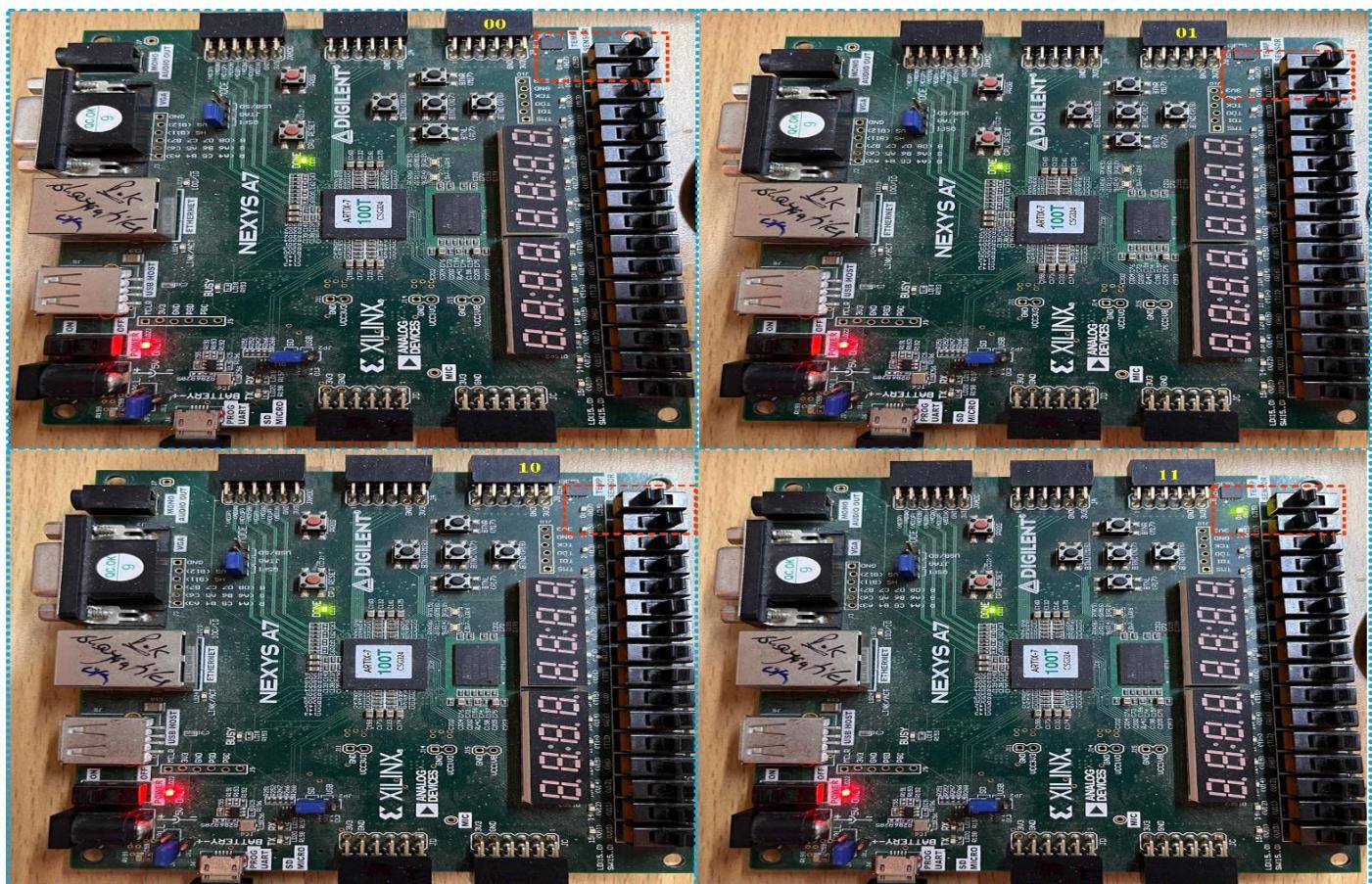
Now programme the configuration memory device by click on 'Ok'





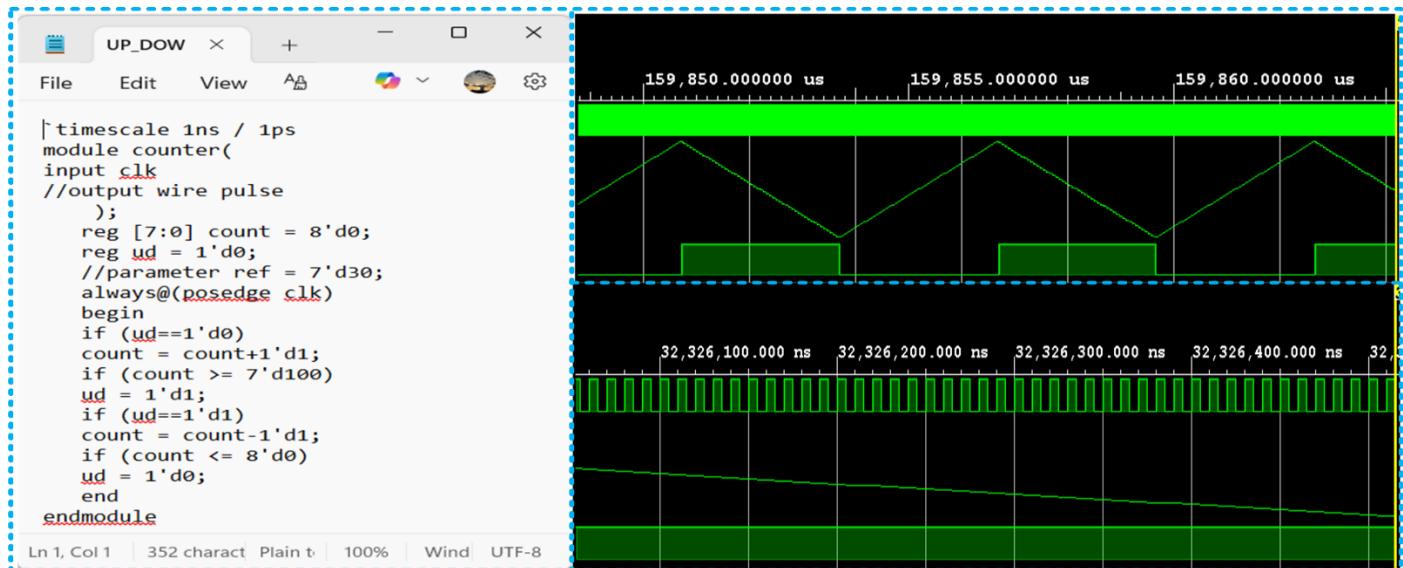
FPGA is successfully programmed.

Result:

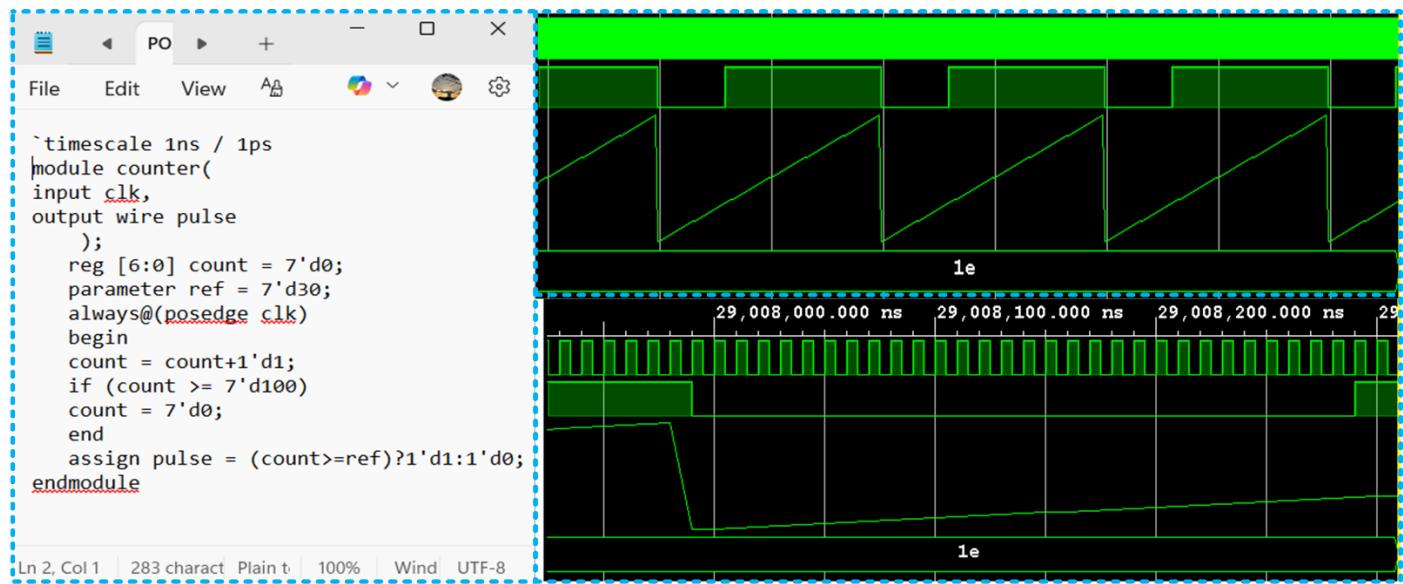


Other sample codes:

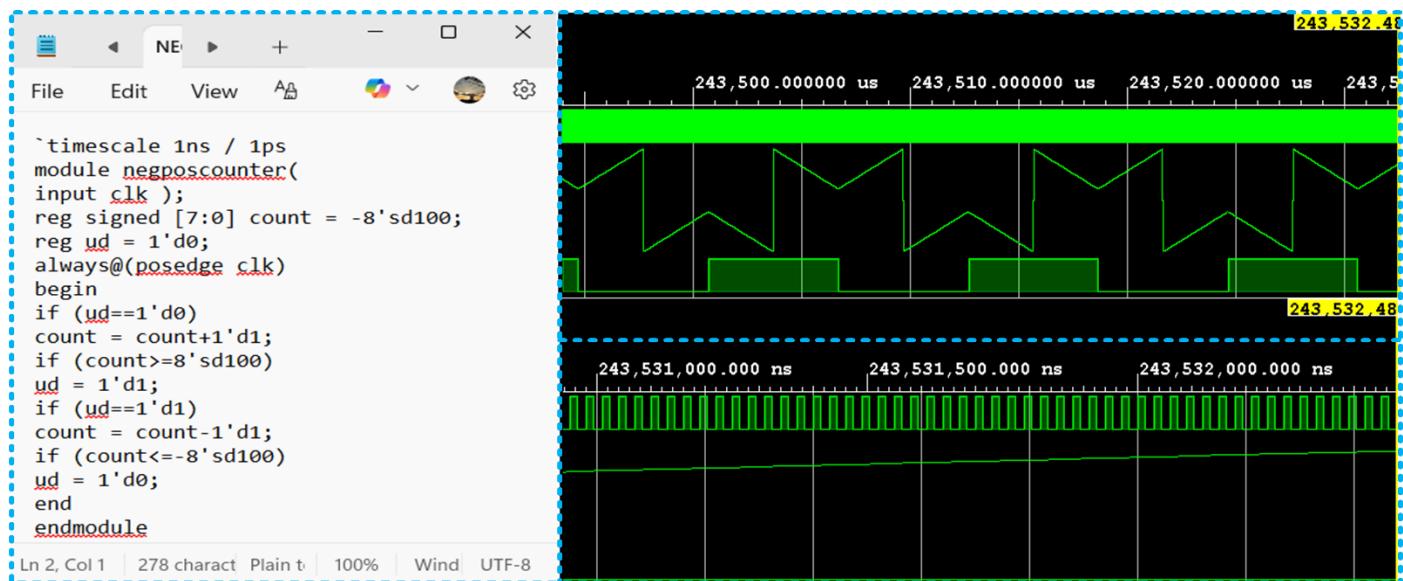
Code 2: Up-Down Counter



Code 3: Generating Positive Gate Pulse



Code 4: Generating -ve to +ve Counter



Scan the QR code to download the XDC file for Nexys A7-100t FPGA development board.



EXPERIMENT NO: 7

SINGLE PHASE INVERTER

OBJECTIVE:

To study, simulate, and analyse the operation and performance of a single-phase Full-Bridge inverter

COMPONENT REQUIRED:

DC Supply, Mosfet (IRFP450), Gate Driver (TLP250), Capacitor, PWM Controller, Load (R, RL), Connecting Wires & Safety Equipment

THEORY:

A single-phase inverter is a converter topology converts DC power into AC power using switching devices such as MOSFETs or IGBTs arranged in full-bridge configuration.

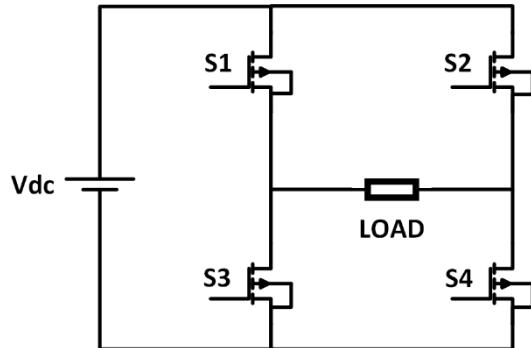


Fig 7.1 : Single phase inverter

The output waveform can be controlled using switching strategy like Square wave switching, sinusoidal pulse width modulation (SPWM). SPWM can be implemented in bipolar or unipolar modes, each with distinct harmonic characteristics.

$$Ma = \frac{V_{ref}}{V_{carrier}} \quad V_o = \frac{ma V_d}{2}$$

Where Ma = modulation index ($0 < Ma < 1$)

$$THD = \sqrt{\sum_{n=2}^{\infty} V_n^2} \times 100\%$$

Types of Switching Techniques:

Square Wave Switching:

The switches are turned on and off to produce a square-shaped output voltage waveform, switching is done at the fundamental frequency.

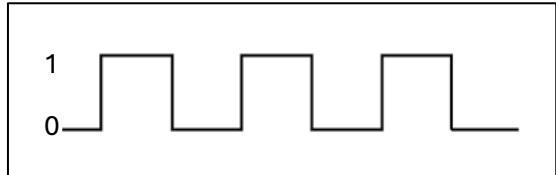


Fig 7.2 : Square wave switching

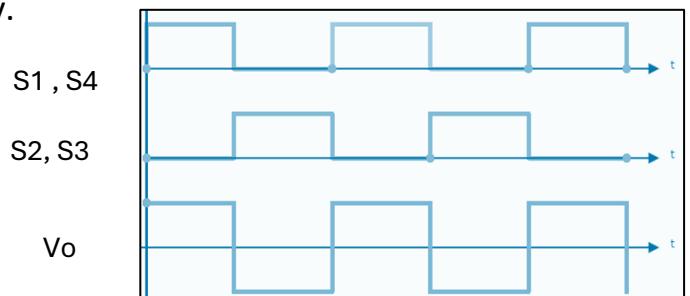


Fig 7.3: output voltage using square wave switching

Sinusoidal pulse width Modulation:

A sinusoidal reference V_{ref} signal is compared with a high-frequency triangular carrier signal $V_{carrier}$ to generate gate pulses.

Bipolar SPWM:

The output voltage of the inverter switches directly between $+V_{dc}$ and $-V_{dc}$

If $V_{ref} > V_{carrier} \rightarrow$ switch S1, S4 is ON \rightarrow output = $+V_{dc}$

If $V_{ref} < V_{carrier} \rightarrow$ switch S3, S2 is ON \rightarrow output = $-V_{dc}$

Unipolar SPWM:

In the unipolar SPWM method, two sinusoidal reference signals of the same frequency but 180° out of phase are compared with a common high-frequency triangular carrier signal

This creates switching between $+V_{dc}, 0$, and $-V_{dc}$ instead of only $+V_{dc}$ and $-V_{dc}$

If $V_{ref} > V_{carrier}$ and $-V_{ref} < V_{carrier} \rightarrow$ switch S1, S4 is ON \rightarrow output = $+V_{dc}$

If $V_{ref} < V_{carrier}$ and $-V_{ref} > V_{carrier} \rightarrow$ switch S3, S2 is ON \rightarrow output = $-V_{dc}$

If $V_{ref} > V_{carrier}$ and $-V_{ref} > V_{carrier} \rightarrow$ switch S1, S2 is ON \rightarrow output = 0

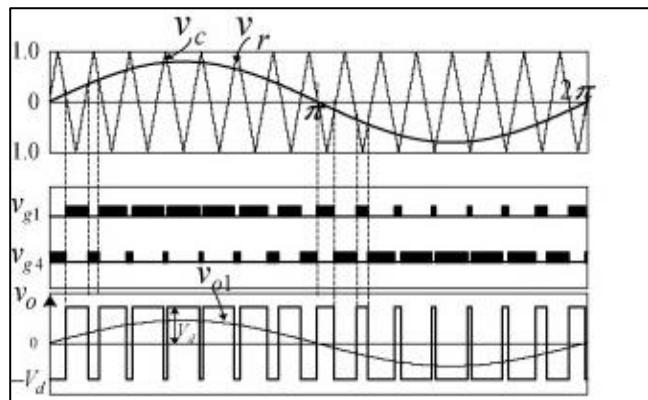


Fig 7.4: output voltage using Bipolar SPWM

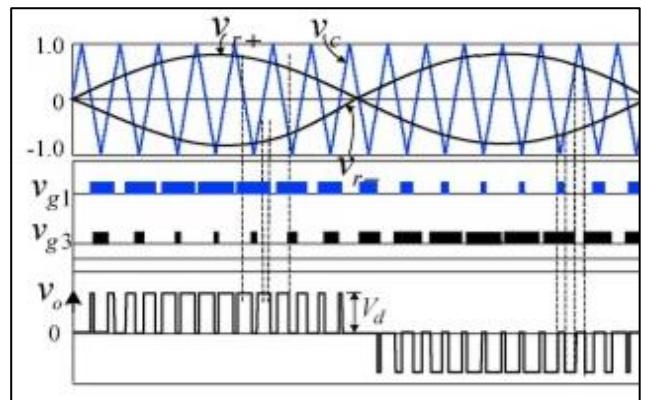


Fig 7.5: output voltage using unipolar SPWM

PROCEDURE:

MATLAB/Simulink

- Create a new model, set solver to discrete, and define stop time
- Add DC source, H-bridge/MOSFETs, and load block
- Generate the pulse for MOSFET gate terminal from any switching technique
- Add measurement blocks, scopes, and optional LC filter
- Run simulation and analyse voltage, current waveforms
- Check FFT analysis of the converters using Matlab Simulink tool

Hardware:

- Connect the DC supply to the DC link capacitor to provide a stable input voltage for the inverter.
- Assemble the H-bridge using four IRFP450 MOSFETs with proper heat sinks and short interconnections.
- Interface each MOSFET gate with a TLP250 gate driver for isolation and level shifting.
- Drive the TLP250 inputs using PWM signals from the PWM controller with appropriate dead-time.
- Connect the inverter output terminals to the load (R or RL) for testing.
- Power the gate drivers and PWM controller, then gradually apply DC supply voltage and measure output voltage/current on the load.

DISCUSSION:

In this section you analyse and discuss your results of simulation and measured values, Waveforms (shape and magnitudes) also check how much deviation in values you are getting from simulation and hardware.

CONCLUSION:

Here you summarize the essential aspects and findings of your work and analysis.

Experiment No. 08

Title of The Experiment: Design and Simulate 3-Φ Inverter with 180° Conduction Mode in MATLAB.

Design and Simulate 3-Φ, 2-level Voltage Source Inverter Using Sine PWM Technique.

Aim of the Experiment:

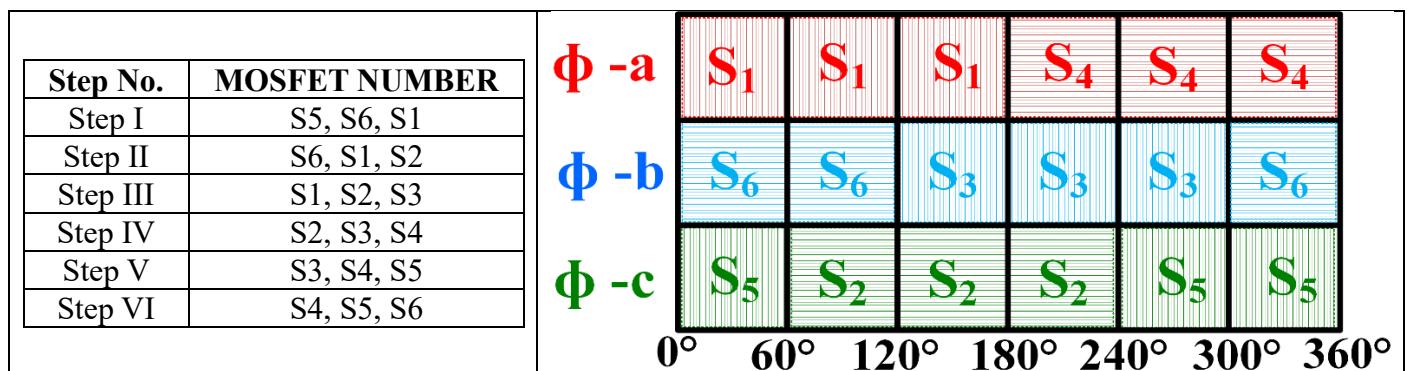
- Generating gate pulses for switching the MOSFET.
- Design a 3-Φ Inverter with a fixed DC voltage.
- Generating sine wave
- Generating triangular signal.

Design and Simulate 3-Φ Inverter with 180° Conduction Mode in MATLAB

Theory:

Each MOSFET is conduct for 180°. MOSFET pair in each arm S1, S4; S3, S6; S5, S2 are turned on with an interval of 180° i.e., for positive half-cycle if S1 conducts for 180° then the next negative cycle S4 conducts for another 180°. If S1 is fired at 0° then S3 is fired at 120° and S5 is fired at 120°. For 360°, a complete cycle there are 6 steps, and each step is 60°. At each step one MOSFET is turned on.

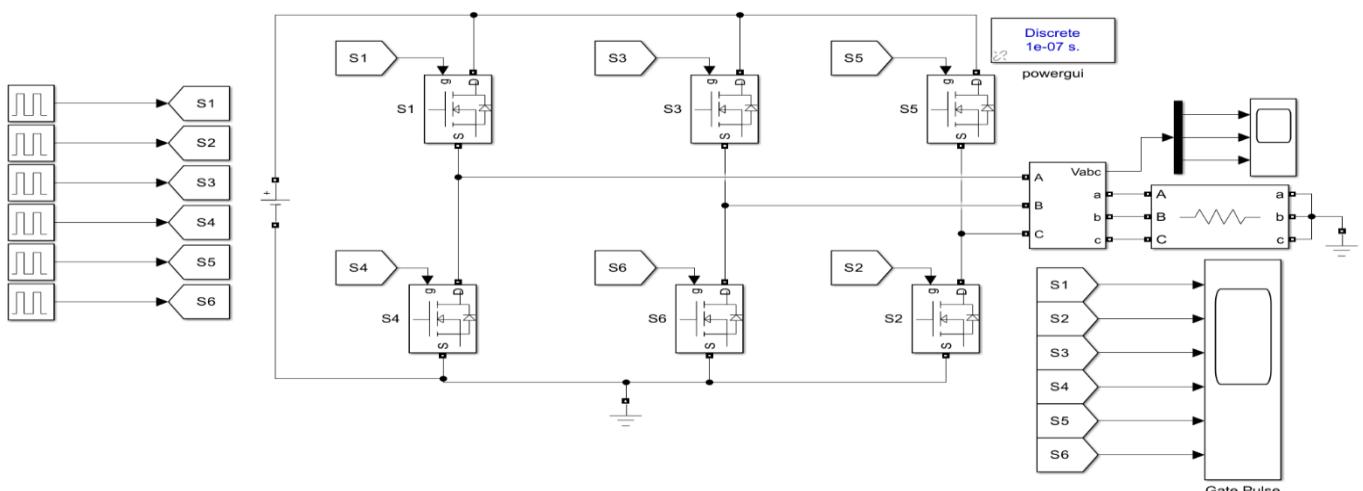
Firing Pattern:



Control Scheme:

Phase ‘a’ start from 0°. First S1 will turn on and it is turned on till 180° and after 180° S1 will turn off and S4 will turn on. Phase ‘b’ is 120° out of phase with phase ‘a’ Phase ‘b’ will start to conduct from 120°. First S3 is turned on and it will conduct for next 180° i.e., 300°. After 300° S3 will turn off and S6 will turn on and conduct for next 120°. Phase ‘c’ is 240° out of phase with phase ‘a’. S5 will turn on at 240° and conduct for next 180° after that S2 will be turned on.

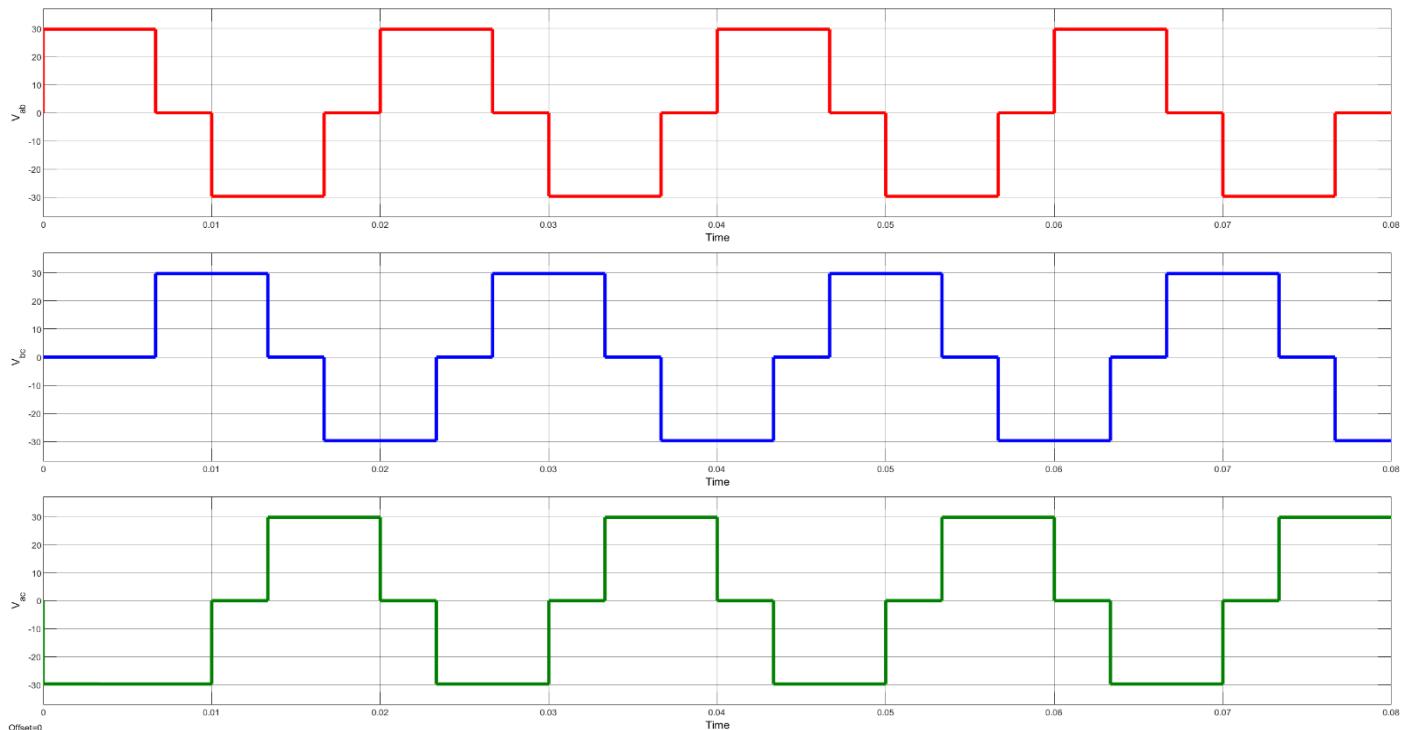
Simulation Circuit Diagram:



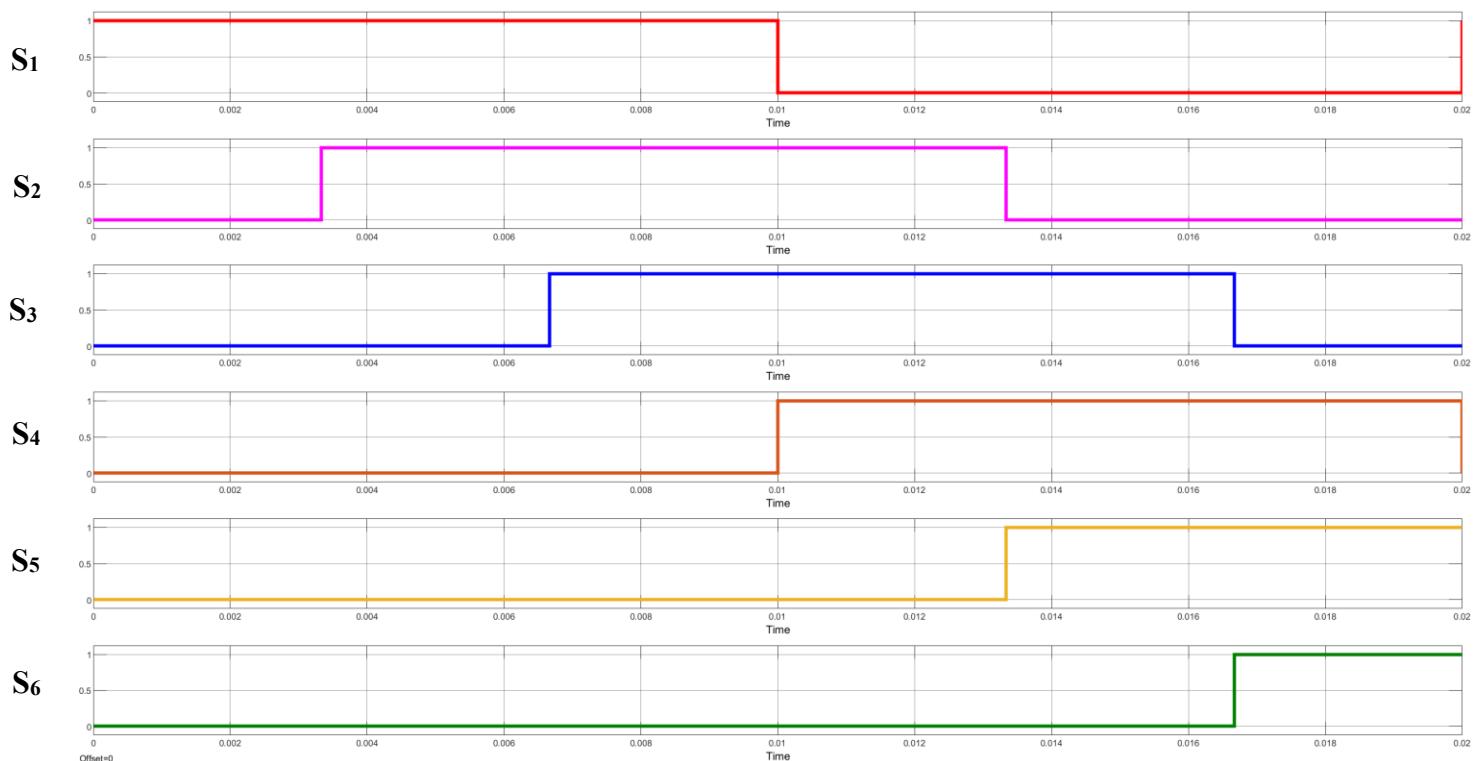
Phase Voltage Pattern:

Firing Angle	V_a	V_b	V_c	V_{ab}	V_{bc}	V_{ca}	MOSFET Sequence
$0^\circ - 60^\circ$	$V_{dc}/3$	$-2V_{dc}/3$	$V_{dc}/3$	V_{dc}	$-V_{dc}$	0	S1, S5, S6
$60^\circ - 120^\circ$	$2V_{dc}/3$	$-V_{dc}/3$	$-V_{dc}/3$	V_{dc}	0	$-V_{dc}$	S1, S6, S2
$120^\circ - 180^\circ$	$V_{dc}/3$	$V_{dc}/3$	$-2V_{dc}/3$	0	V_{dc}	$-V_{dc}$	S1, S3, S2
$180^\circ - 240^\circ$	$-V_{dc}/3$	$2V_{dc}/3$	$-V_{dc}/3$	$-V_{dc}$	V_{dc}	0	S4, S2, S3
$240^\circ - 300^\circ$	$-2V_{dc}/3$	$V_{dc}/3$	$V_{dc}/3$	$-V_{dc}$	0	V_{dc}	S4, S3, S5
$300^\circ - 360^\circ$	$-V_{dc}/3$	$-V_{dc}/3$	$2V_{dc}/3$	0	$-V_{dc}$	V_{dc}	S4, S6, S5

Line Voltage Waveform:

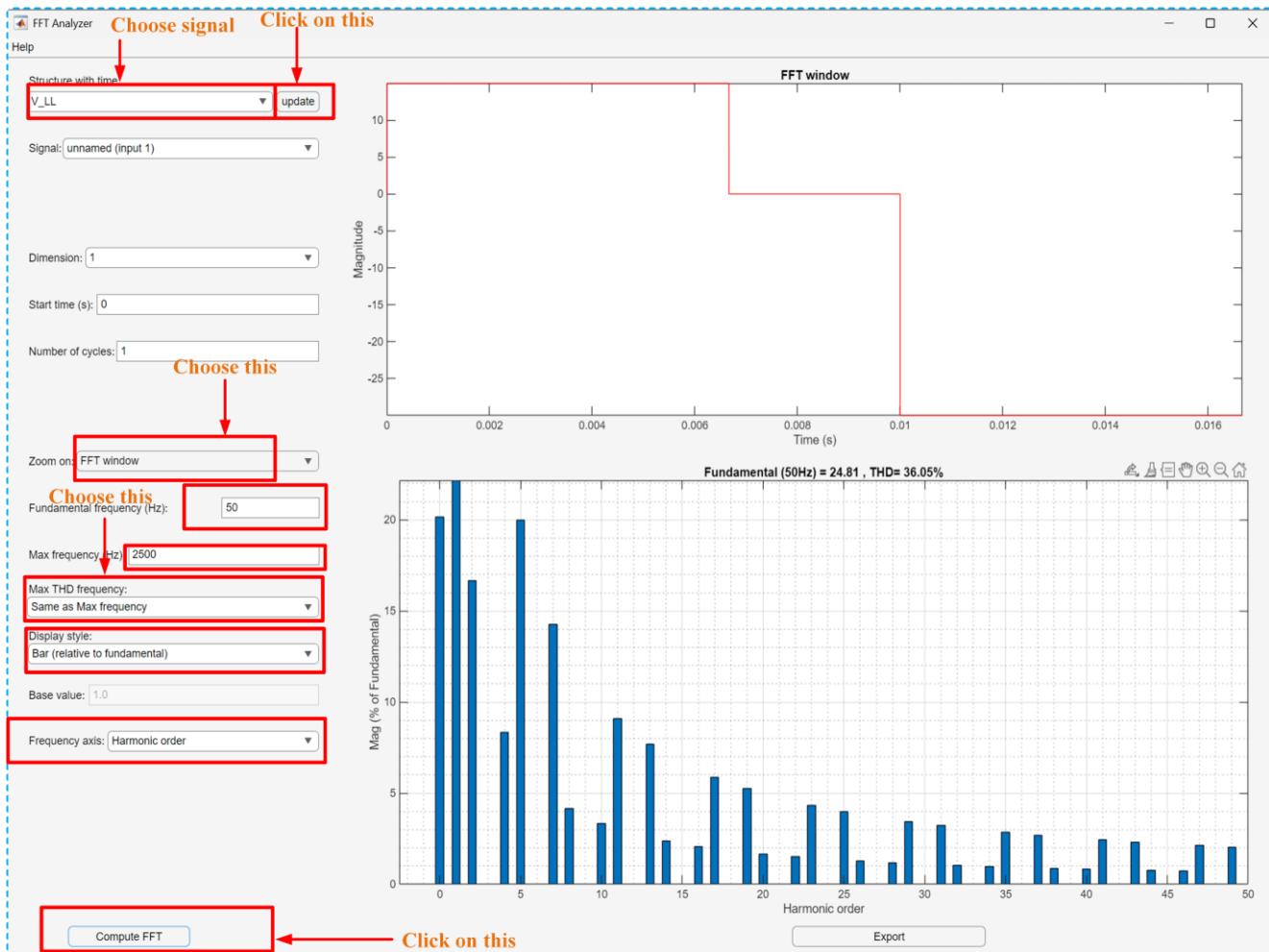
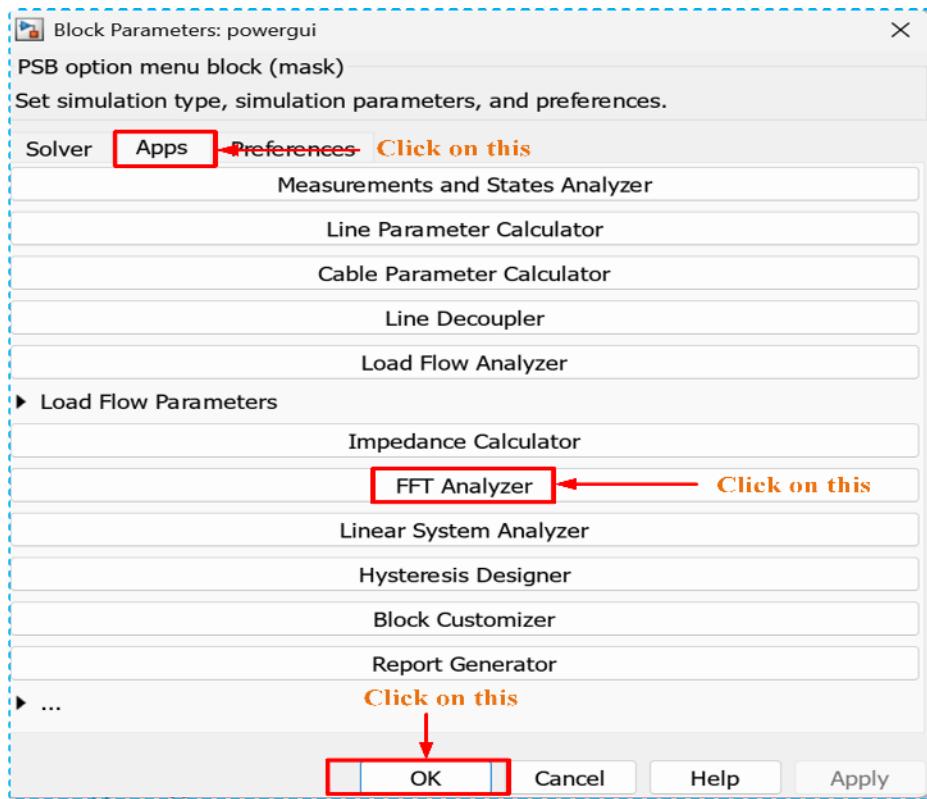


Gate Pulse:



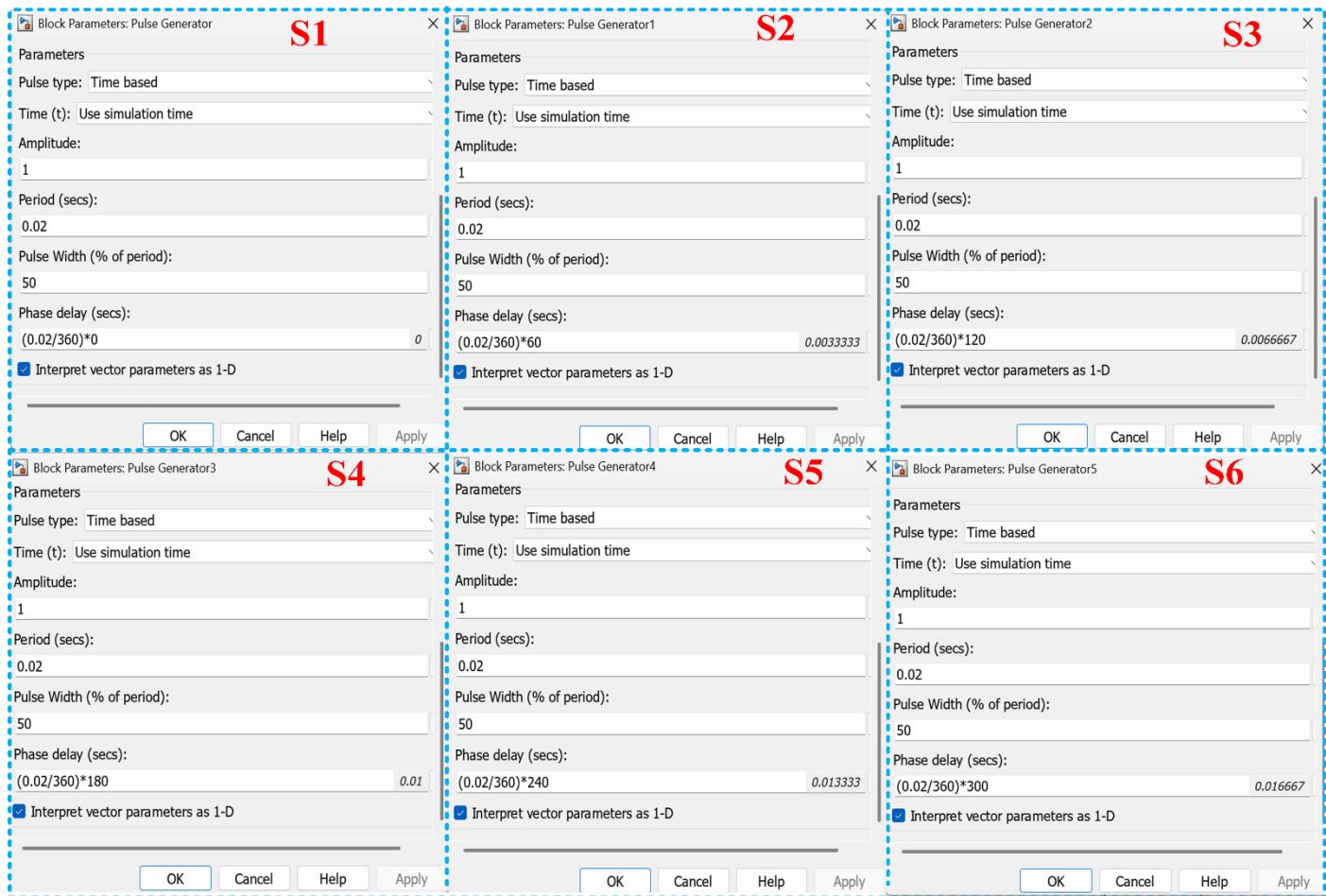
FFT Analysis:

Click on ‘powergui’ block.

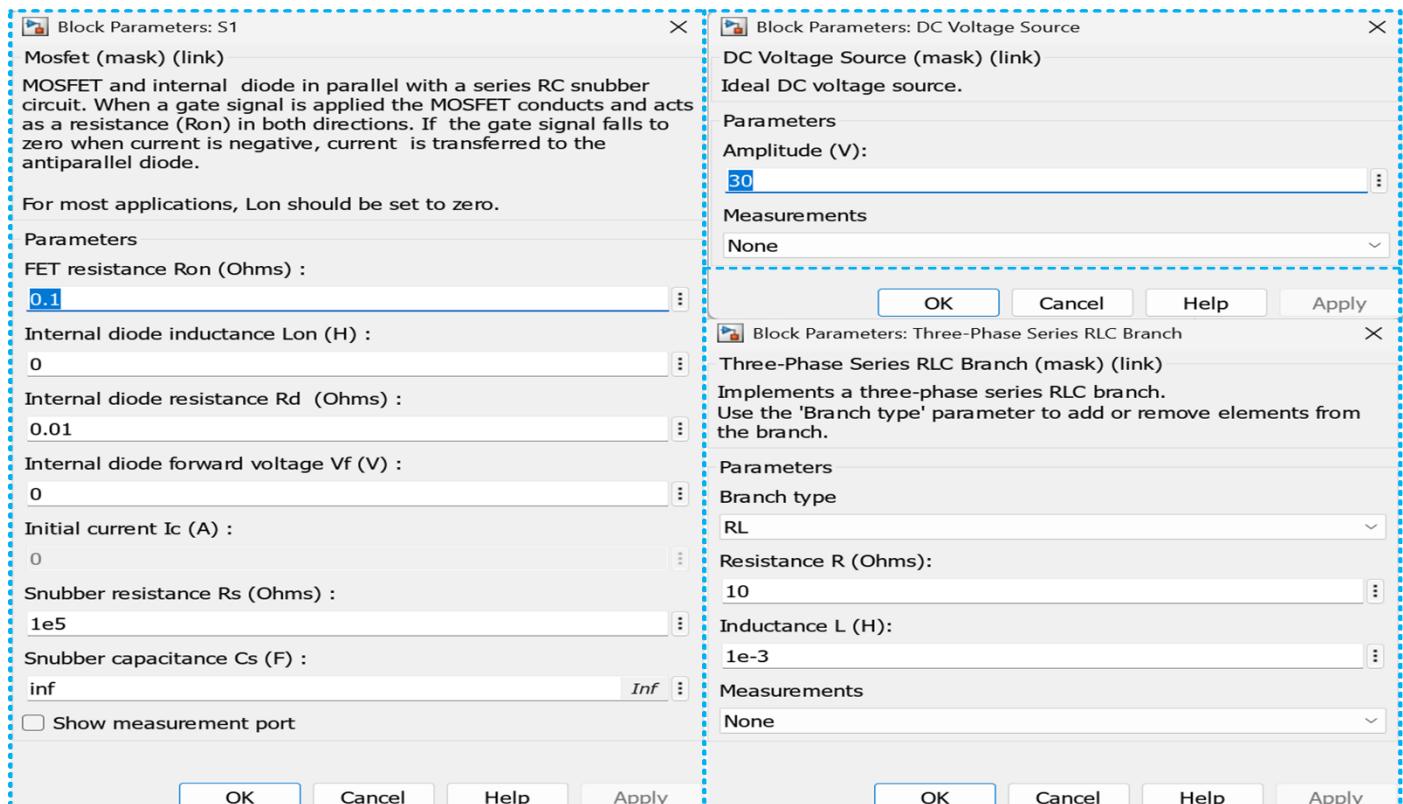


Specification Used for MATLAB:

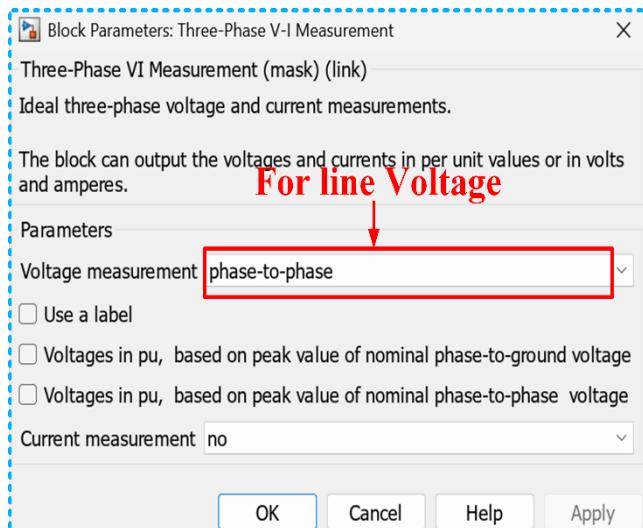
For all 6 Gate Pulses:



For Load and DC Supply:



For Voltage Measurement:



Design and Simulate 3-Φ 2 Level Voltage Source Inverter Using Sine PWM Technique in MATLAB

Theory: A three-phase Voltage Source Inverter (VSI) with SPWM (Sinusoidal Pulse Width Modulation) is a type of inverter that converts DC voltage into three-phase AC voltage with sinusoidal waveforms. It works by varying the pulse width of a high-frequency carrier signal according to the instantaneous amplitude of a reference sinusoidal waveform. In a 3-phase inverter, three separate SPWM signals are generated for each phase, by comparing a high-frequency triangular waveform with three sinusoidal reference waveforms (one for each phase) to determine the pulse widths of the inverter's switching devices. A three-phase VSI consists of six power semiconductor switches, typically insulated-gate bipolar transistors (IGBTs) or power MOSFETs, arranged in an H-bridge configuration. The switches are grouped into three pairs, each controlling one phase of the output voltage (V_a , V_b , and V_c). To generate the desired three-phase sinusoidal output, three reference sinusoidal waveforms (V_{ra} , V_{rb} , and V_{rc}) are generated. These reference waveforms have a fixed frequency (ω) and amplitude (V_m) and are phase-shifted by 120° relative to each other.

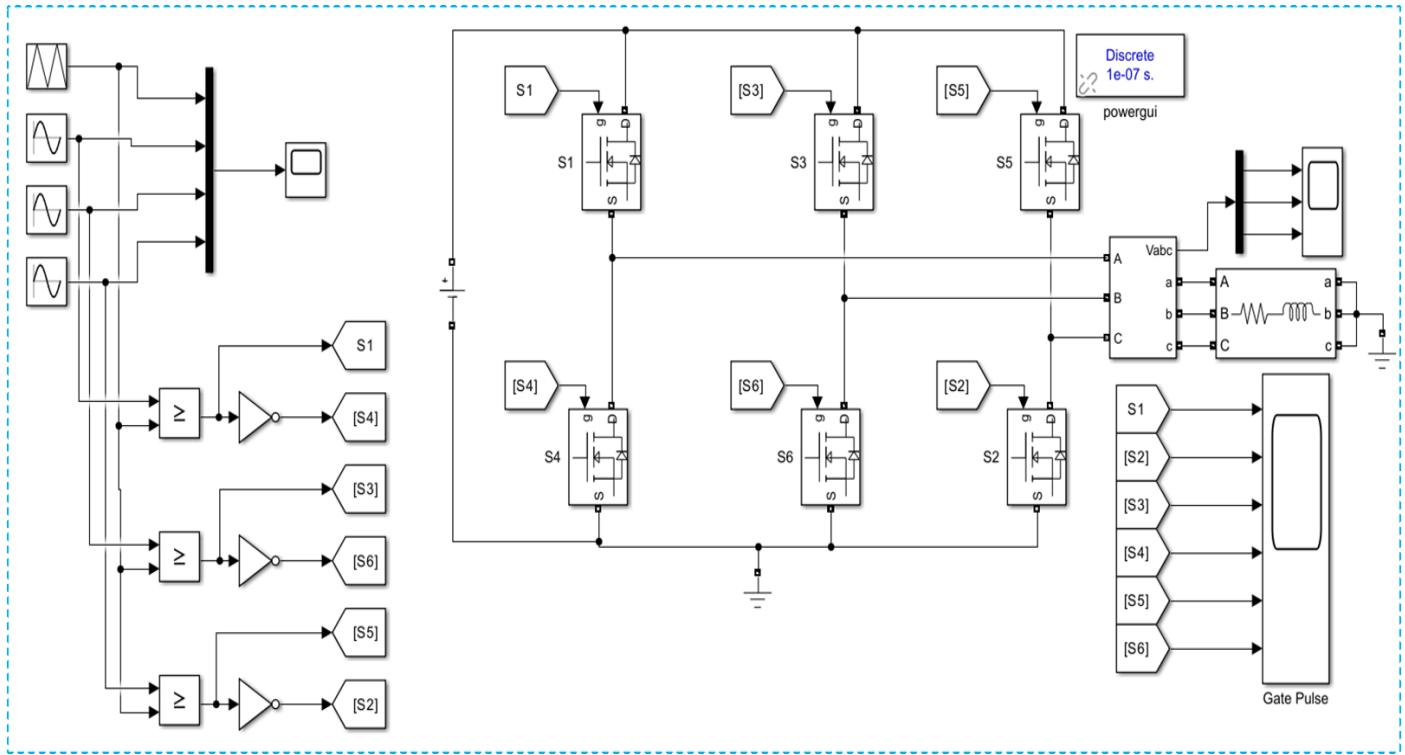
A high-frequency triangular carrier waveform (V_{carr}) is generated with a frequency (f_c) much higher than the desired output frequency of the inverter. The carrier waveform varies between $-V_c$ and $+V_c$, where V_c is the peak amplitude of the carrier waveform.

The instantaneous values of the reference sinusoidal waveforms (V_{ra} , V_{rb} , and V_{rc}) are continuously compared with the triangular carrier waveform (V_{carr}) using comparators. Based on the comparison, the inverter switches are turned ON or OFF for specific durations, determining the pulse width of the output voltage. When the reference waveform is greater than the carrier waveform, the corresponding switch is turned ON. When the reference waveform is smaller, the switch is turned OFF, during the positive half-cycle of the reference waveform $V_{ra}(t)$, the switch S_1 (top switch) and S_4 (bottom switch) for phase A are turned ON if $V_{ra}(t) > V_{carr}(t)$. The switches will remain ON until $V_{ra}(t) < V_{carr}(t)$ during the negative half-cycle.

Modulation index (MI): It determines the extent to which the reference waveforms are compared with the carrier waveform. It influences the amplitude of the output voltage. The modulation index is given by:

$$M_i = \frac{V_m}{V_c}, 0 \leq M_i \leq 1$$

Simulation Circuit Diagram:



Simulation Specification:

Triangle Generator (mask) (link)

Generate a symmetrical triangle wave with peak amplitude of +/- 1.

Parameters

- Frequency (Hz): $1e3$
- Phase (degrees): 0
- Sample time: 0

Sine Wave

Parameters

- Sine type: Time based
- Time (t): Use simulation time
- Amplitude: 1
- Bias: 0
- Frequency (rad/sec): $2\pi*50$
- Phase (rad): 0
- Sample time: 0

Three-Phase Series RLC Branch (mask) (link)

Implements a three-phase series RLC branch. Use the 'Branch type' parameter to add or remove elements from the branch.

Parameters

- Branch type: RL
- Resistance R (Ohms): 10
- Inductance L (H): $1e-3$
- Measurements: None

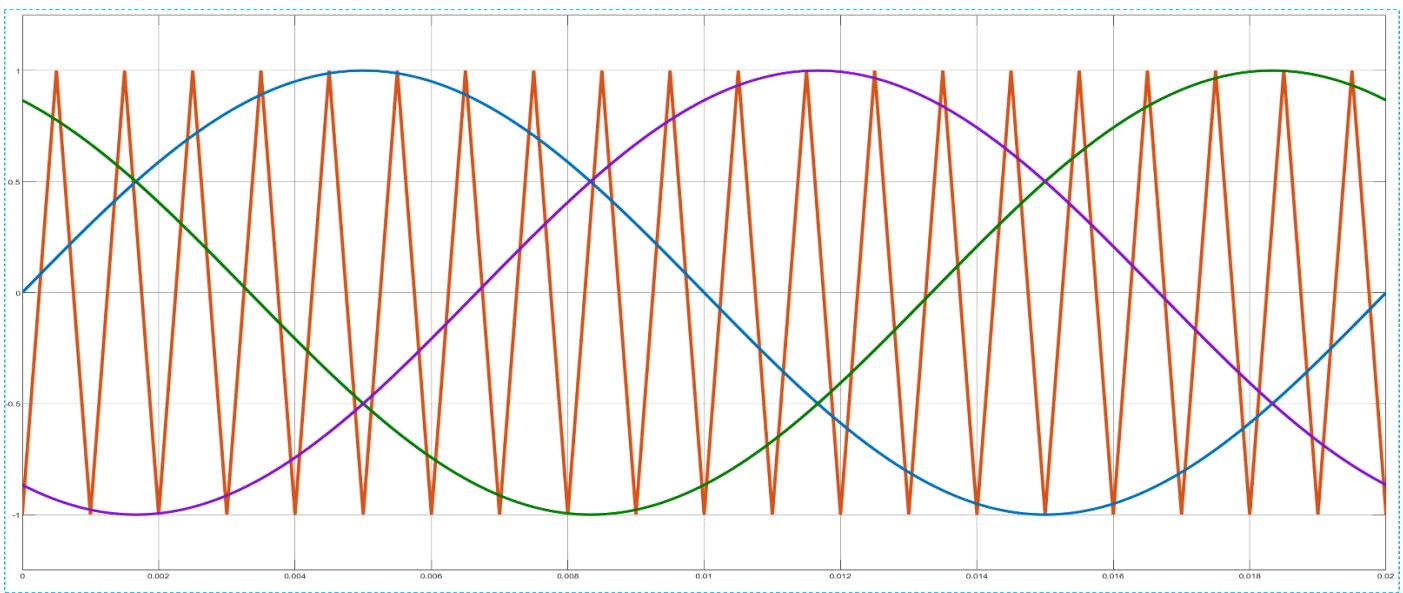
Three-Phase VI Measurement (mask) (link)

Ideal three-phase voltage and current measurements.

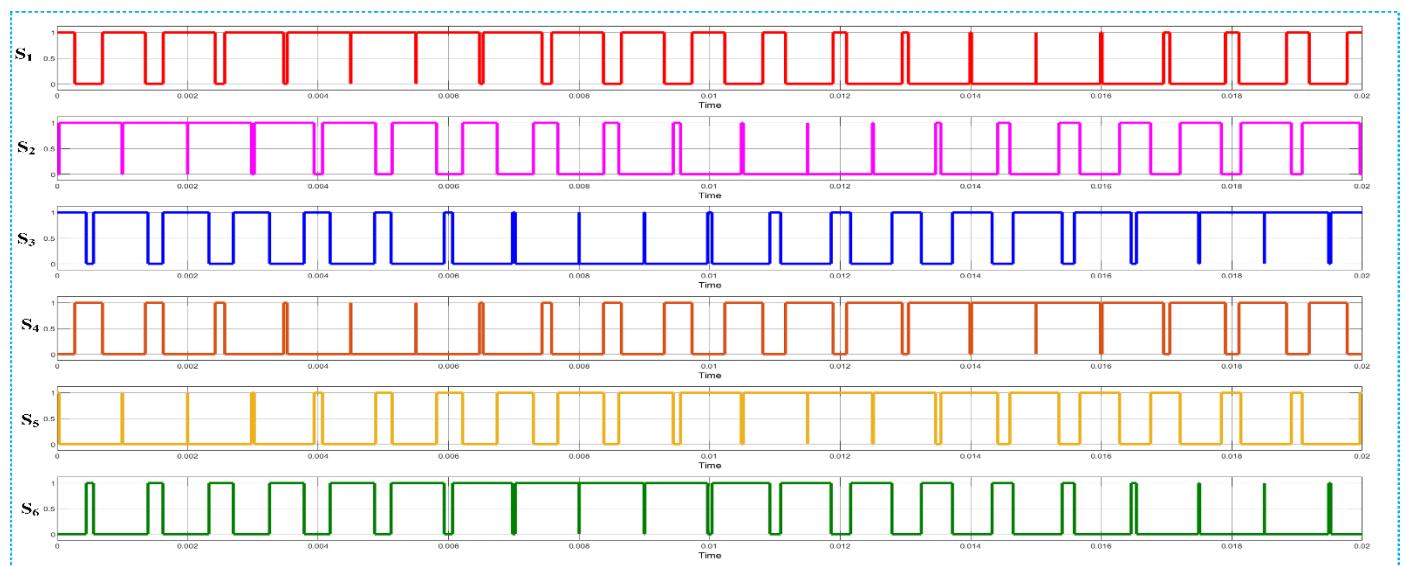
Parameters

- Voltage measurement phase-to-phase
- Use a label
- Volts in pu, based on peak value of nominal phase-to-ground voltage
- Volts in pu, based on peak value of nominal phase-to-phase voltage
- Current measurement no

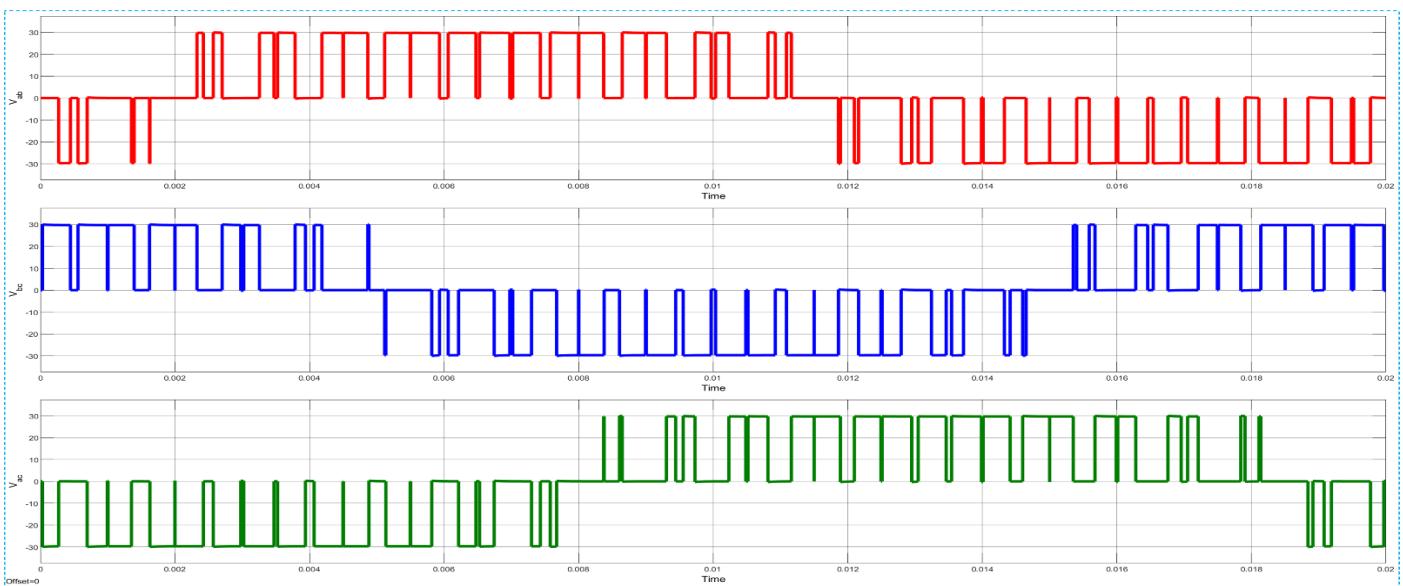
Triangle Sine Comparison:



Gate Pulse:

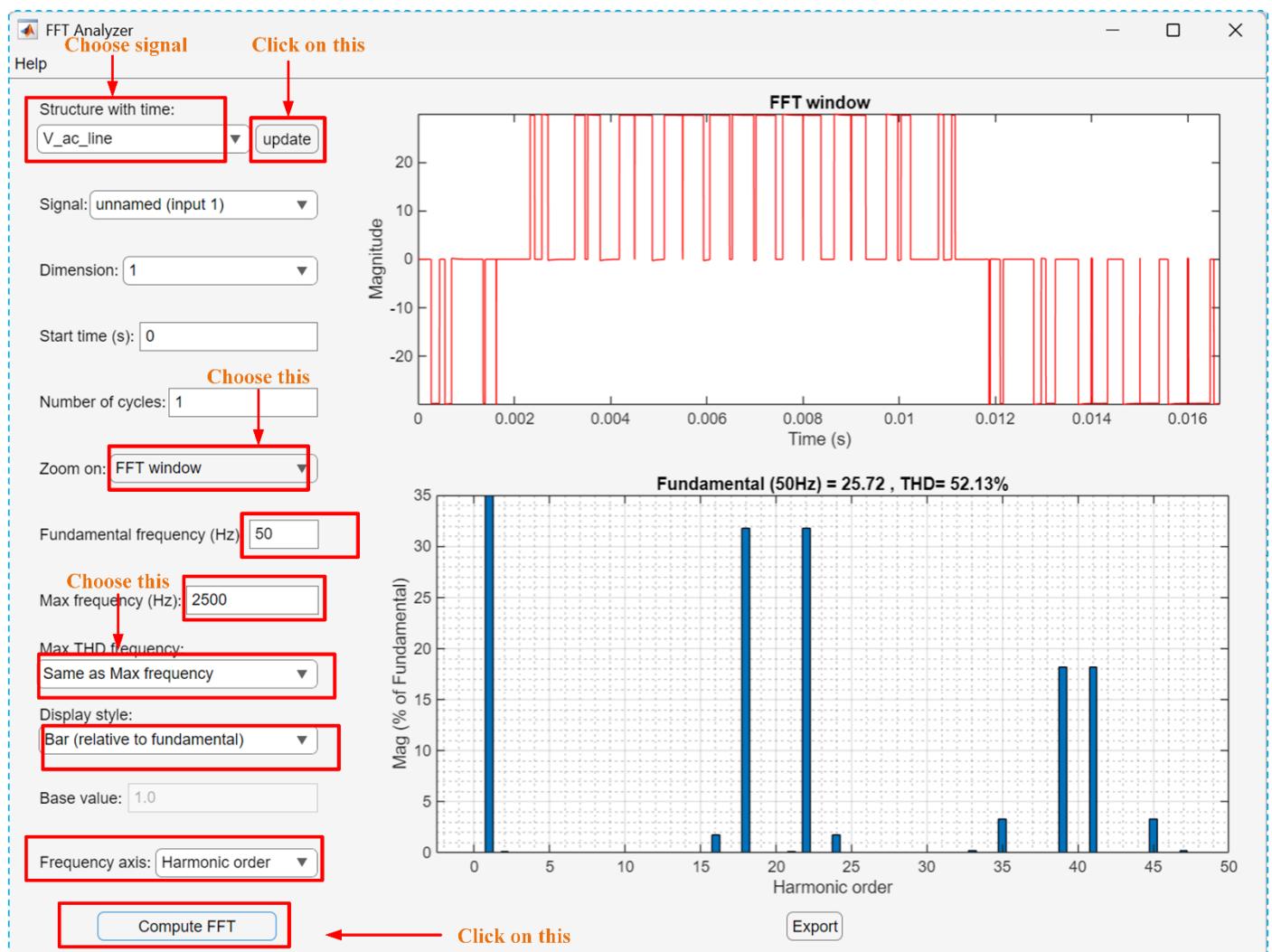
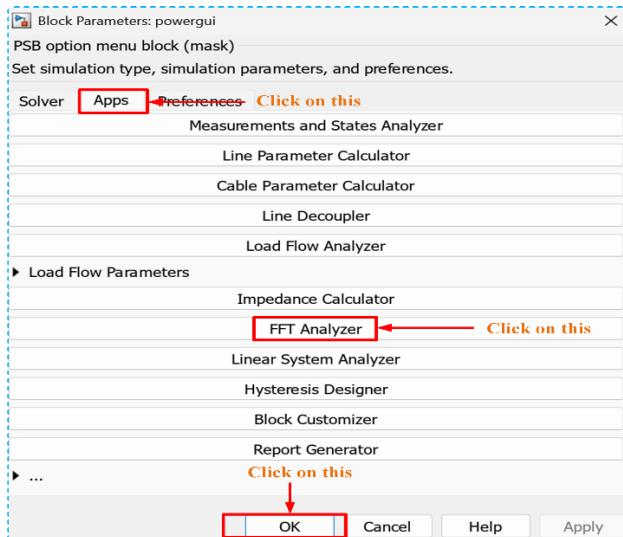


Output Voltage:



FFT Analysis:

Click on ‘powergui’ block.



EXPERIMENT - 9

OBJECTIVE:

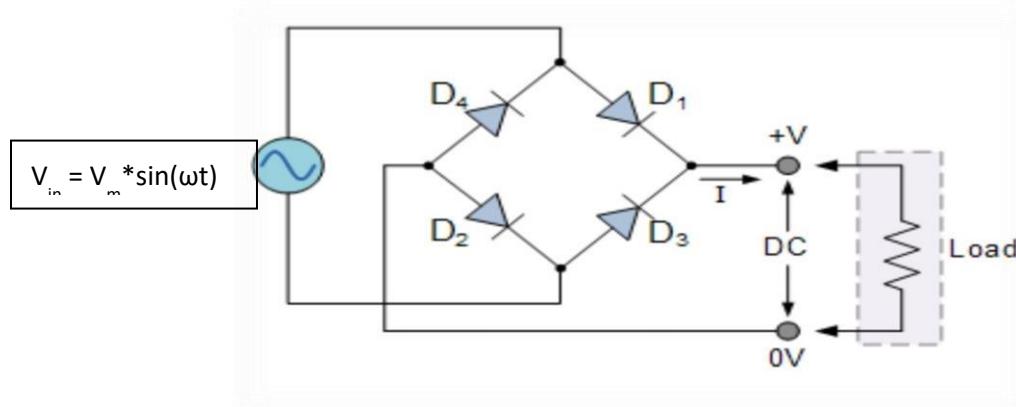
- 1 . Study input and output waveform of a full-wave diode bridge rectifier.
2. Finding ripple in output voltage with and without capacitor.

COMPONENTS:

NAME	SPECIFICATIONS
DIODE	IN4007
RESISTOR	

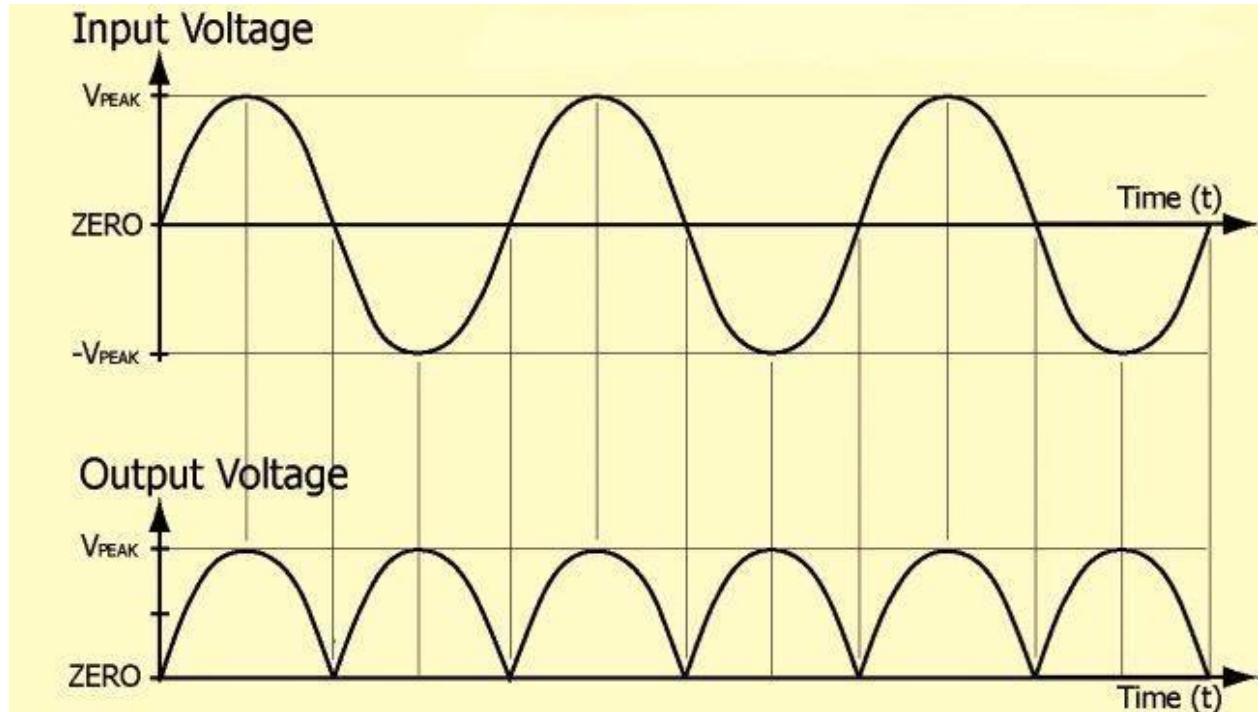
THEORY:

1. Circuit diagram of full bridge rectifier without capacitor:



The bridge rectifier is a circuit which converts AC voltage to DC voltage using both half cycle of the input AC voltage. The circuit has four diodes connected to form a bridge. The ac input voltage is applied to the diagonally opposite ends of the bridge. The load resistance is connected between

the other two ends of the bridge. For the positive half cycle of the input AC voltage, diodes D1 and D2 conduct, whereas diodes D3 and D4 remain in the OFF state. The conducting diodes will be in series with the load resistance RL and hence the load current flows through RL. For the negative half cycle of the input ac voltage, diodes D3 and D4 conducts whereas, D1 and D2 remain OFF. The conducting diodes D3 and D4 will be in series with the load resistance RL and hence the current flows through RL in the same direction as in the previous half cycle. Thus a bi-directional wave is converted into a unidirectional wave.



- The full bridge rectifier inverts the negative half of the input ac signal. □
- Output is a full wave rectified sine wave (pulsating DC).

Calculation of average output voltage:

$$\text{Since } V_{in}(t) = V_m * \sin(\omega t)$$

$$V_o = \frac{1}{\pi} \int_0^{\pi} V_m \sin(\omega t) dt$$

$$V_o = \frac{V_m}{\pi} * [-\cos(\omega t)]_0^{\pi}$$

$$V_o = \frac{V_m}{\pi} * [-\cos(\pi) + \cos(0)]$$

$$V_o = \frac{V_m}{\pi} * 2 = \frac{2V_m}{\pi}$$

Voltage ripple without capacitor:

Output waveform is pulsating so ripple is 100% (waveform oscillates between zero and peak value or maximum value).

2. Full bridge rectifier with capacitor filter:

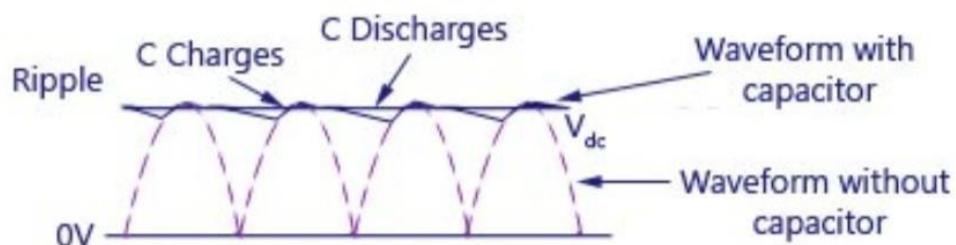
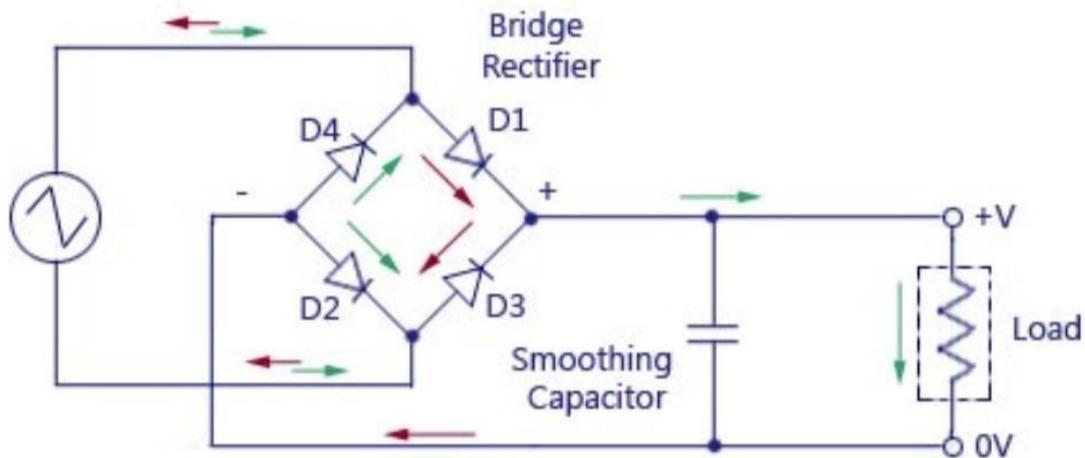
- The capacitor charges to near peak voltage V_m .
- Between peaks it discharges slowly through the load creating a smoother output.
- Output is a rippled DC with small fluctuations.
- AVERAGE OUTPUT VOLTAGE**

Assuming ideal diode and large capacitor. V_{out}

$$= V_{peak} = V_m$$

But if the diode is practical diode with internal resistance then due to drops in voltage across the diode resistance and capacitor discharge it is slightly less than V_m but $V_{out} \approx V_m$

$$V_{peak} \approx V_m . I_{out} = \frac{V_{out}}{R}$$



Resultant Output Waveform

RIPPLE VOLTAGE:

For full wave rectifier, ripple frequency $2*f$.

Ripple in capacitor voltage is equal to $I/2fC$.

PROCEDURE:

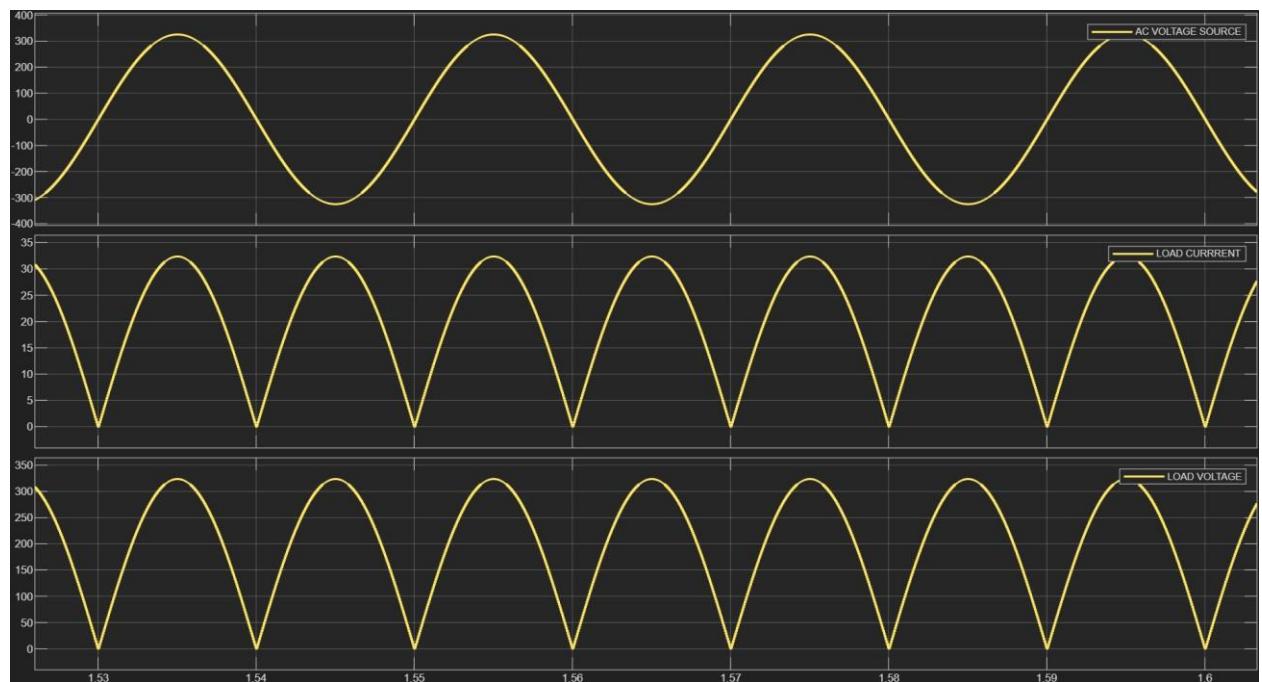
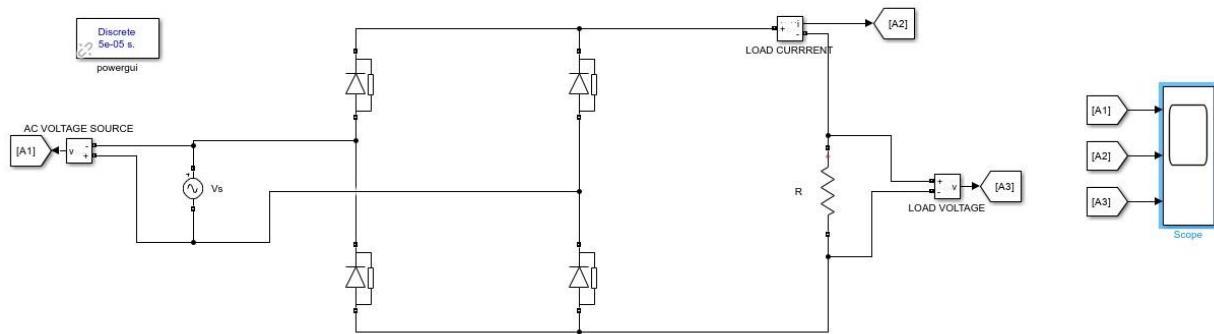
The following steps followed during the MATLAB simulation:

- Add AC voltage source to your Simulink model.
- Add four diodes and connect in a bridge configuration.
- Add resistive load across the bridge.
- Add voltage measurement block in the circuit.
- Add scope to visualize the voltage waveform.

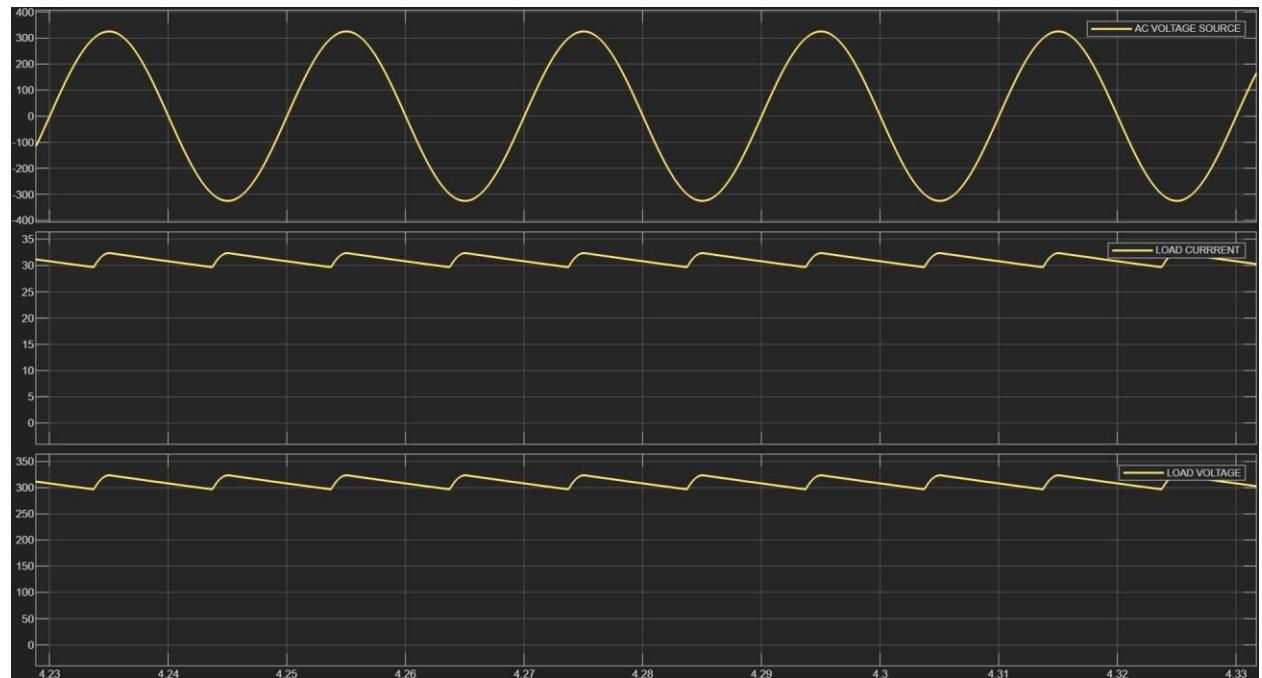
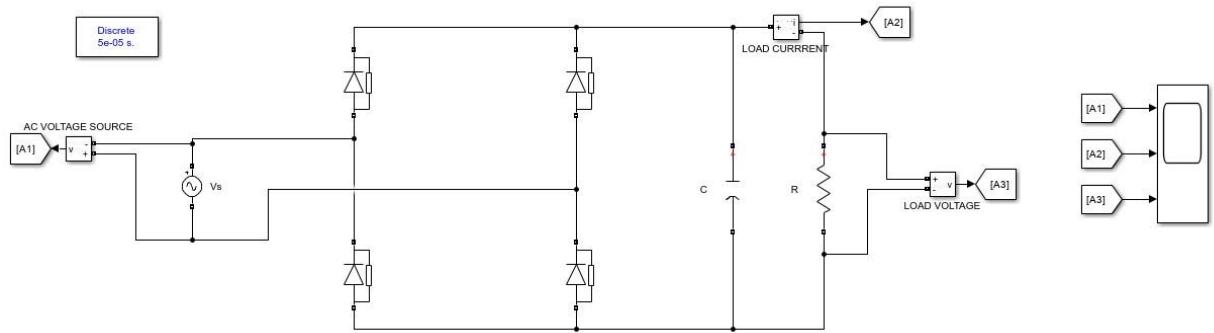
- Add powergui block.

RESULTS:

1. Simulation results without capacitor:



2. Simulation results with capacitor:



CONCLUSION:

We have performed the experiment of full wave diode bridge rectifier with and without capacitor and observed that output voltage ripple reduces with capacitor connected parallel to the load.