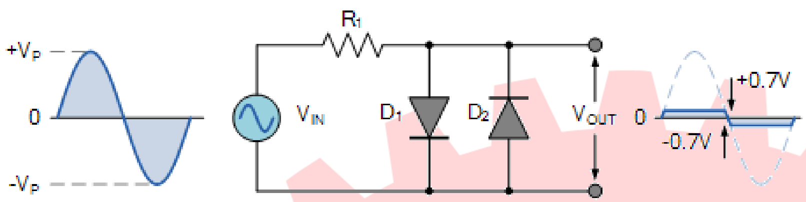
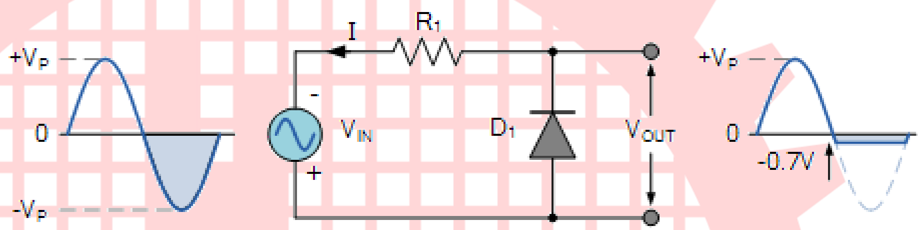
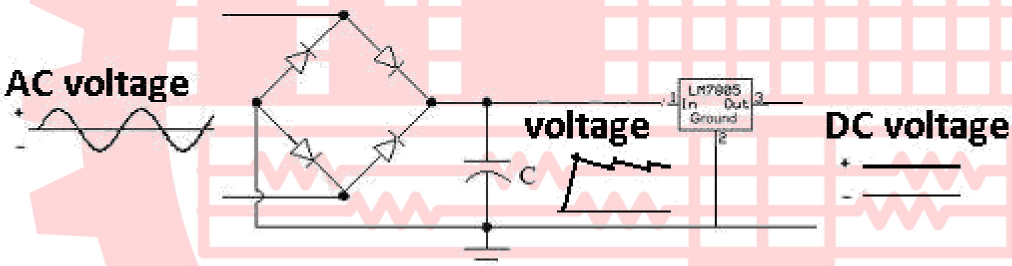
AC Signal Modifier: A Project on Clipping, Clamping & DC Regulation of Ac Input Signal.



Clipper



Clamper



DC Regulator

Submitted By:

Name: Sudipto Pramanik

Student ID: 1806172

Department: EEE

Level: 2

Term: 1

Course No: EEE212

Date of Submission: 30.07.2021

OBJECTIVE

Alternating Current Signals - AC Signals are the signals that periodically reverses direction and changes its magnitude continuously with time. An AC signal can be of different shapes such as sinusoidal, square, triangular or sawtooth. Various systems have the ability to modify an AC signal when it is sent to the input terminal of the system. These are called AC signal modifier. Modification of AC signal includes clipping, clamping, AC to DC regulation etc. Clipping is the operation where a portion of the input signal is cut-off or eliminated; without distorting the remaining part of the waveform. Clamping indicates the shifting of input waveform to a different DC level without changing the appearance of the applied signal. DC regulation ensures a constant DC voltage as the output in stead of receiving an AC voltage as input.

Diode is as a two-terminal electronic component that primarily conducts electric current only in one direction. This feature of diode has enabled it to play its role in many important applications. Besides its act as switch or rectifier, diode has significant role in designing clipper, clamper or DC regulator circuit. Resistor, capacitor and DC voltage source are the other components of these circuits.

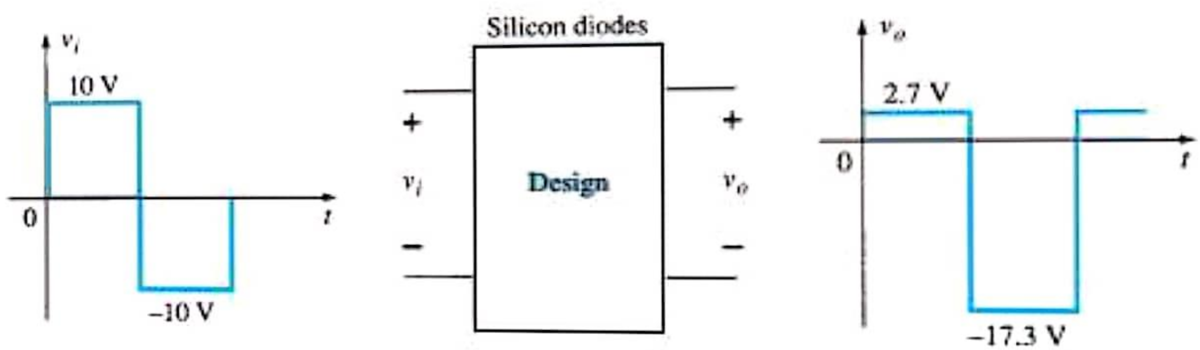
The objective of this project is, therefore, the implementation of AC signal modifier namely clipper, clamper and DC regulator circuits. Design of these circuits leads a student to clearly understand the applications of diode in practical life. This project also teaches him that how a engineering problem should have to trace, analyze and to solute. Moreover, from this project one can learn how the generalizations of possible inputs should be handled. To conclude, this project is a wonderful idea that enriches one with the practical knowledges along with gaining some problem solving skills.

TASK

Design an electronic circuit that will modify (clipping, clamping, regulated DC) a given AC voltage input.

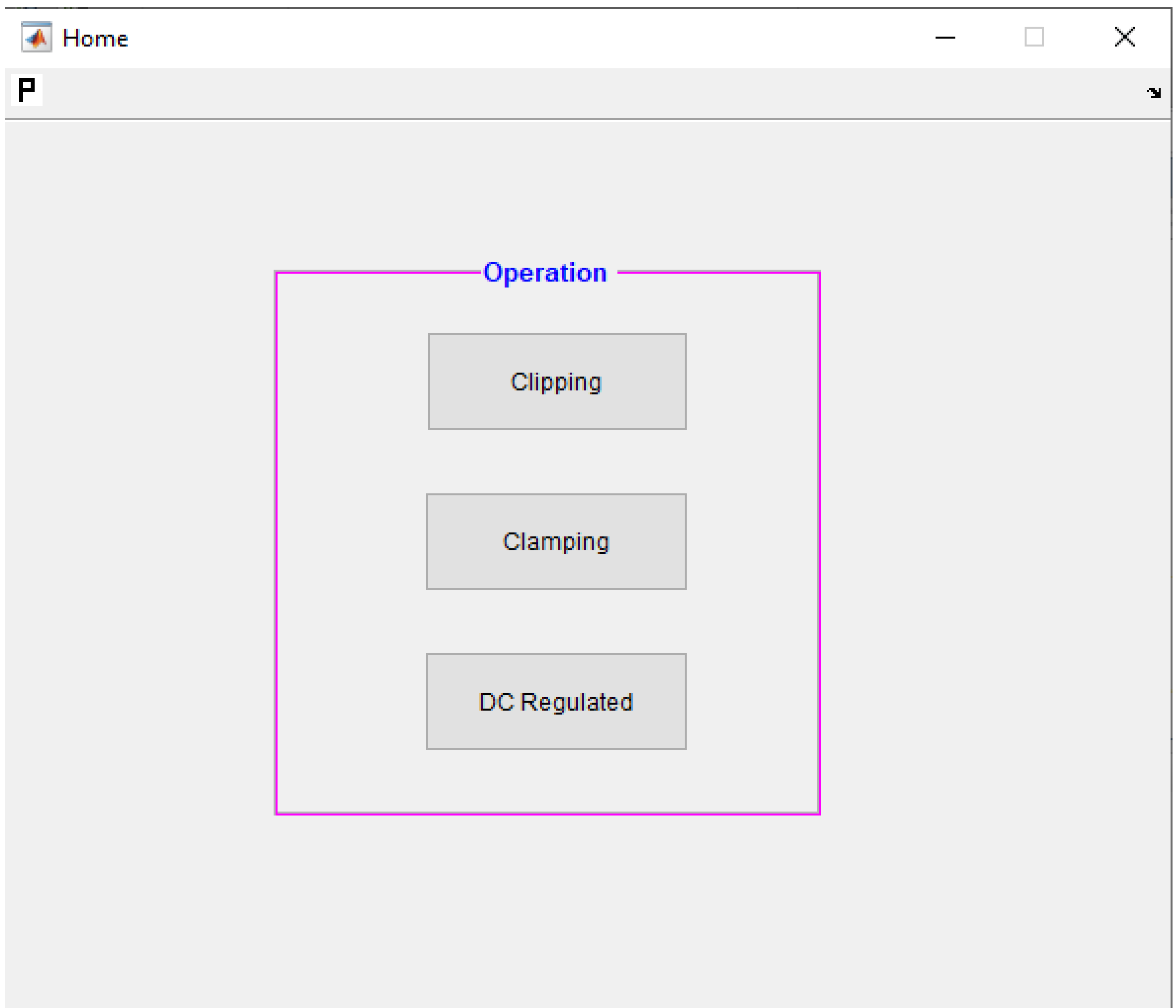
- ✓ Input AC voltage can be of any three of 1) Sinusoidal, 2) Square, 3) Triangular / Sawtooth. Input voltage parameter (peak value, frequency) can be chosen by user.
- ✓ Make a suitable GUI for your project. In home tab there will be 3 option to choose- 1) clipping 2) clamping 3) DC regulated
- ✓ After making choice it will direct user to next tab corresponding to previous choice. There user can choose
    - Upper and lower voltage level for clipper
    - Upper and lower voltage level for clamper
    - Regulated voltage level for DC regulated output
- ✓ You need to show the schematic of the designed circuit containing the value of each element.
- ✓ Also show the input and output voltage curve with time
- ✓ For any invalid choice (i.e. clipped voltage level greater than input voltage peak, regulated dc greater than input voltage peak etc) your design must show "Invalid choice".

For example, you may have to design the following circuit:



OPERATION AT A GLANCE

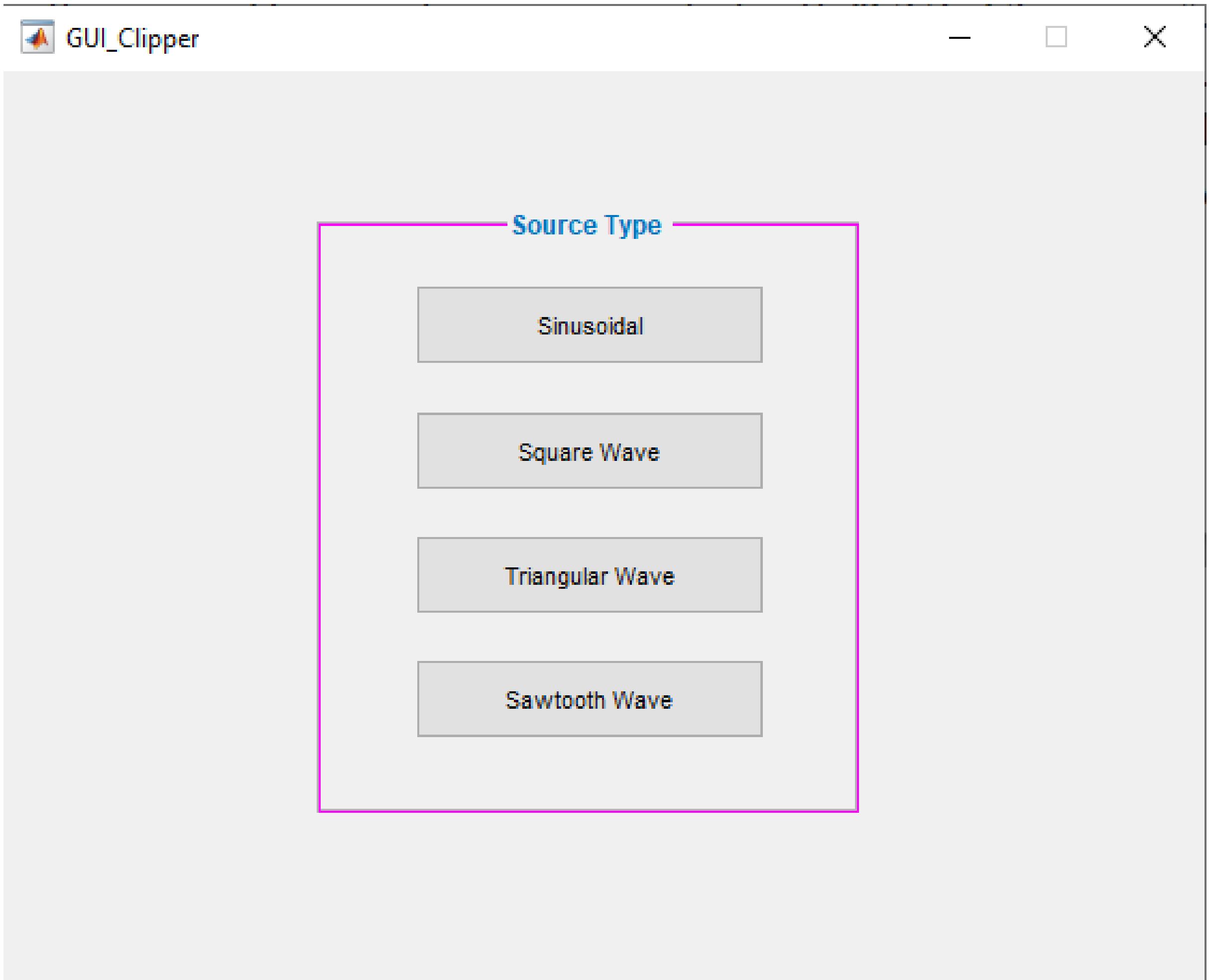
- "Home.m" corresponds to the home tab of the whole program. By running this .m file a GUI window with three options will be available as shown below.



Each option is associated with corresponding operation. For example, if the user wants to perform "Clipping" operation he will choose it, and same goes for "Clamping" and "DC Regulated".

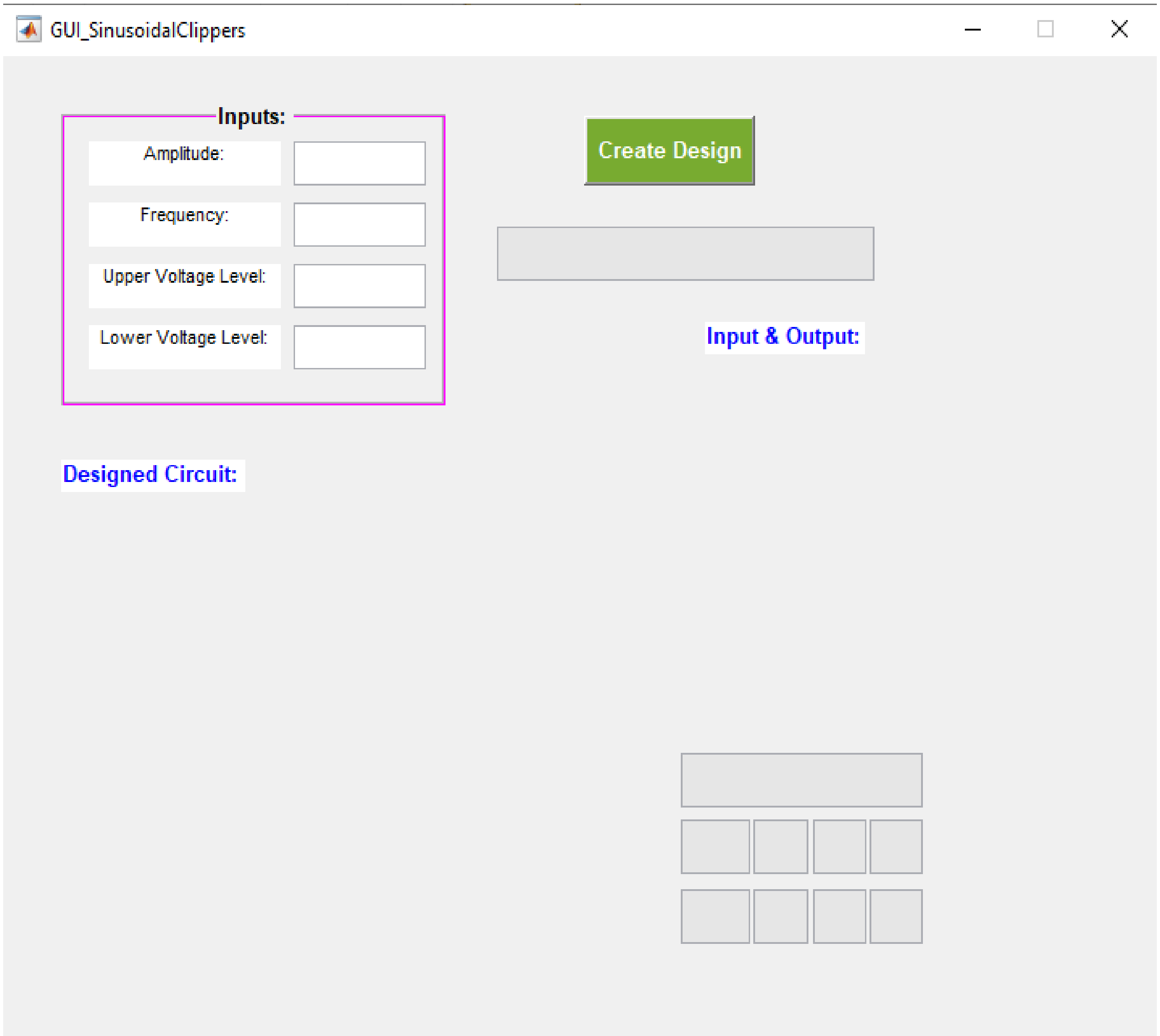
- Choosing any of three operations will let the user to enter another window. For example, if user chooses "Clipping" a window like below will appear. This window is associated with "GUI_Clipper.m". It asks the user for choosing the source type with which he wants to deal. Similarly, when "Clamping" is chosen, "GUI_Clamper.m" and when "DC

Regulated” is chosen, “GUI_DCRegulator.m” with their corresponding windows does the same thing.

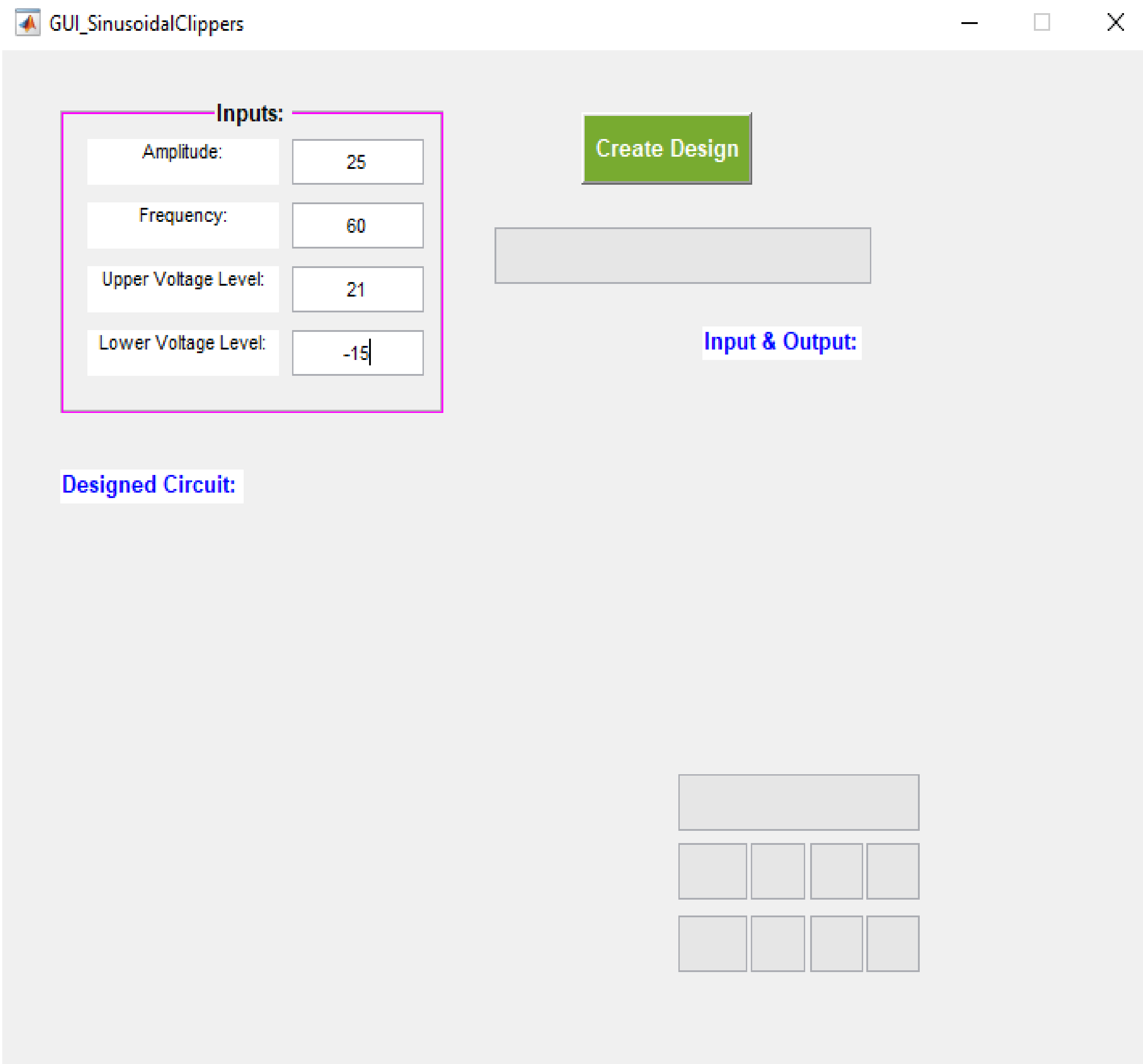


- Suppose “Sinusoidal” has been chosen as source type. Now a third window will open. “GUI_SinusoidalClippers.m” works behind this window. The user should give the necessary inputs such as “Amplitude”, “Frequency”, “Upper Voltage Level” and “Lower Voltage Level”. Next, the user should click on “Create Design” button. Both the schematic of the designed circuit and plots of input and output will appear. In this phase, “Sinusoidal_Clipper_Design.m” is called by the GUI file where designs are made and data are sent back to the GUI file, and the GUI file shows necessary outputs in the screen. The process is all same for other three types of sources.

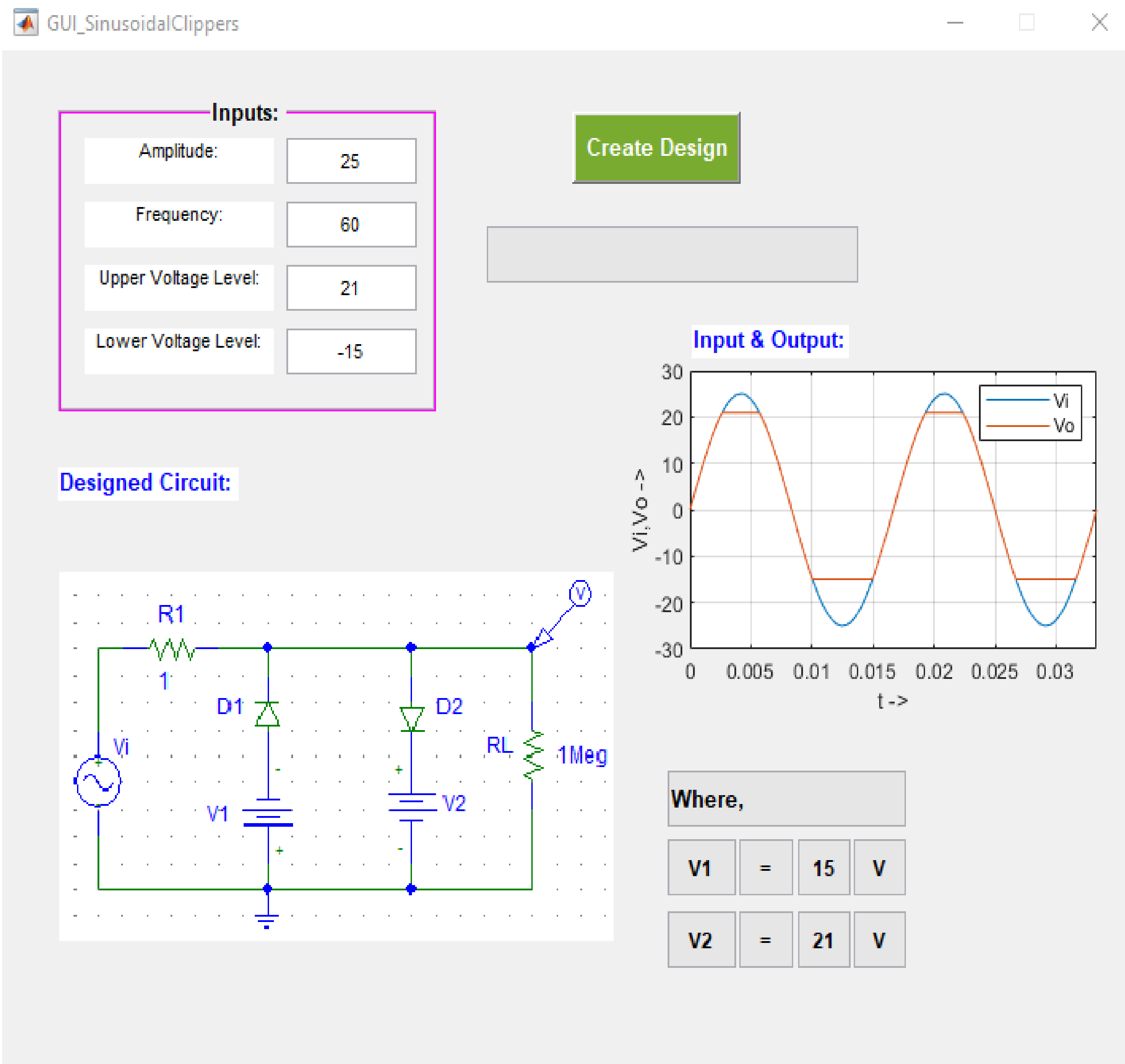
Phase-1: After Clicking "Sinusoidal"



Phase-2: After Giving Inputs.



Phase-3: After Clicking "Create Design".



- Corresponding operations of "Clamping" and "DC Regulated" also take place in the same way as it is in the case of "Clipping". Associated files handle these operations.

DETAILED OPERATION

- As told before, “Home.m” corresponds to the home tab of the whole program.

Code:

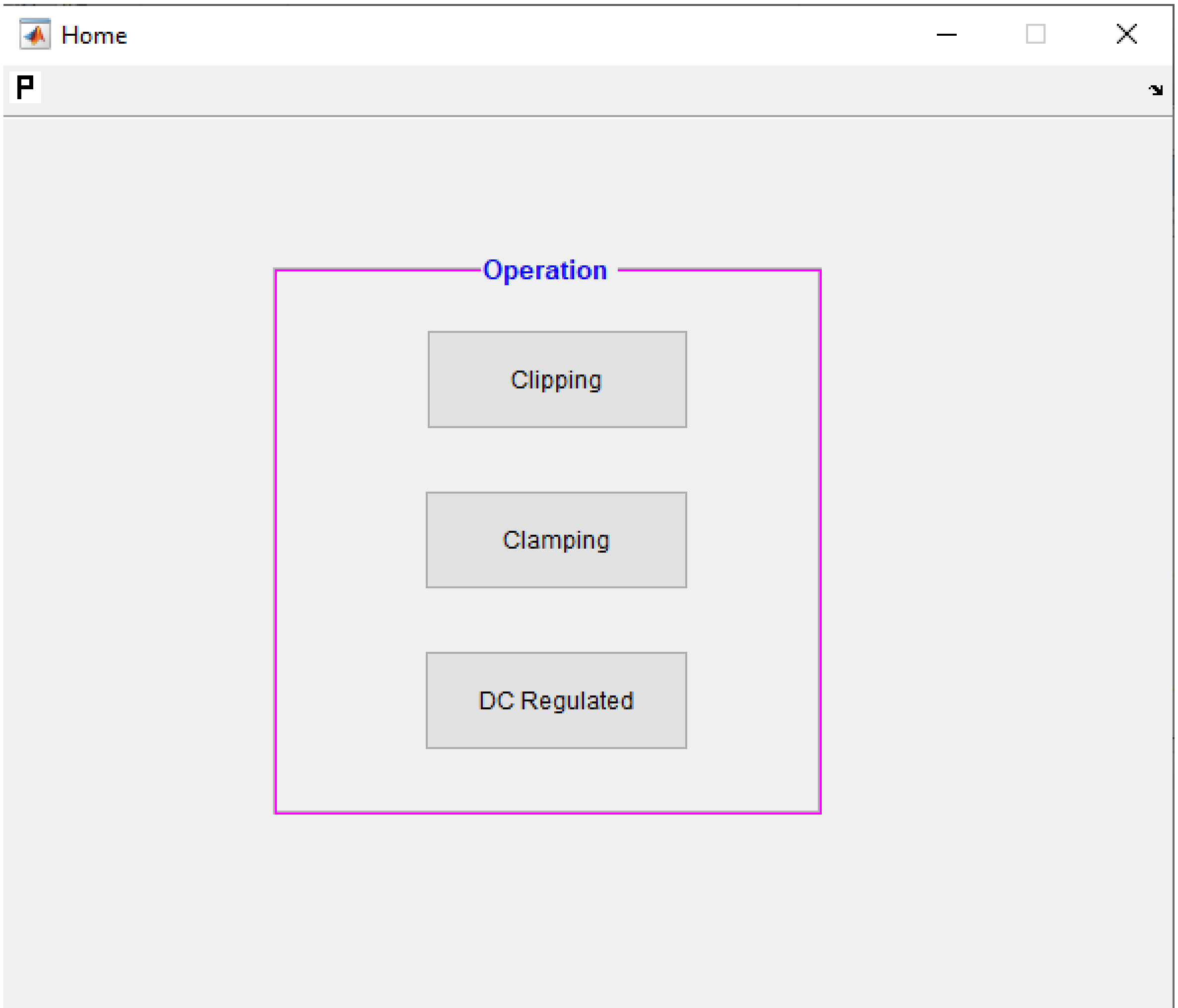
“Home.m” -

```
function varargout = Home(varargin)
% HOME MATLAB code for Home.fig
% HOME, by itself, creates a new HOME or raises the existing
% singleton*.
%
% H = HOME returns the handle to a new HOME or the handle to
% the existing singleton*.
%
% HOME('CALLBACK',hObject,eventData,handles,...) calls the local
% function named CALLBACK in HOME.M with the given input arguments.
%
% HOME('Property','Value',...) creates a new HOME or raises the
% existing singleton*.  Starting from the left, property value pairs are
% applied to the GUI before Home_OpeningFcn gets called.  An
% unrecognized property name or invalid value makes property application
% stop.  All inputs are passed to Home_OpeningFcn via varargin.
%
% *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help Home

% Last Modified by GUIDE v2.5 09-Jul-2021 00:45:39

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @Home_OpeningFcn, ...
```

```
               'gui_OutputFcn',  @Home_OutputFcn, ...
               'gui_LayoutFcn',  [] , ...
               'gui_Callback',   []);
if nargin && ischar(varargin{1})
   gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
   [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
   gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before Home is made visible.
function Home_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to Home (see VARARGIN)

% Choose default command line output for Home
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Home wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Home_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
```

```
varargout{1} = handles.output;


% --- Executes on button press in clipper.
function clipper_Callback(hObject, eventdata, handles)
% hObject    handle to clipper (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
GUI_Clipper


% --- If Enable == 'on', executes on mouse press in 5 pixel border.
% --- Otherwise, executes on mouse press in 5 pixel border or over clipper.
function clipper_ButtonDownFcn(hObject, eventdata, handles)
% hObject    handle to clipper (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% --- Executes during object creation, after setting all properties.
function clipper_CreateFcn(hObject, eventdata, handles)
% hObject    handle to clipper (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called


% --- Executes on button press in clamper.
function clamper_Callback(hObject, eventdata, handles)
% hObject    handle to clamper (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
GUI_Clamper


% --- Executes on button press in voltage_regulator.
function voltage_regulator_Callback(hObject, eventdata, handles)
% hObject    handle to voltage_regulator (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
GUI_DCRegulator
```

Corresponding Window:



1. Clipping:

- Suppose,“Clipping” has been chosen from the home tab. It will direct us to “GUI_Clipper.m”.

Code:

“GUI_Clipper.m” -

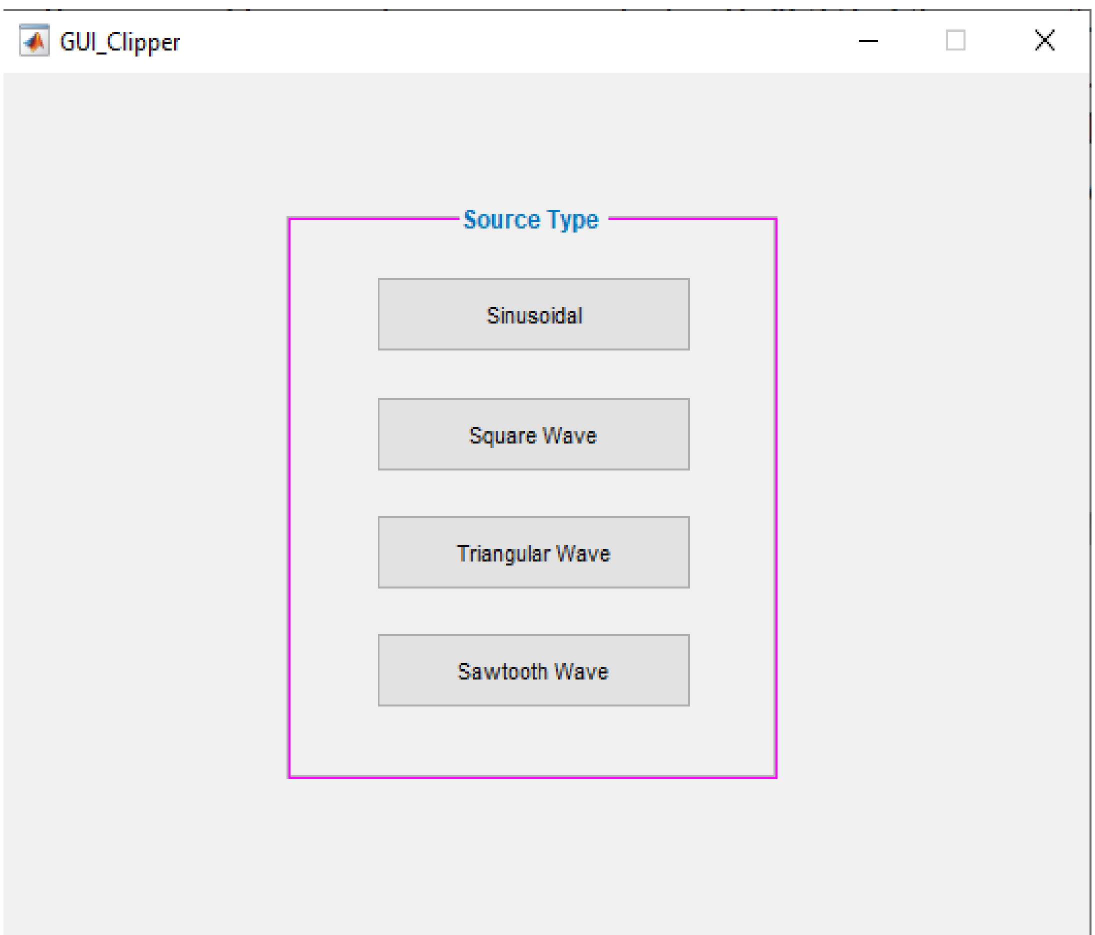
```
function varargout = GUI_Clipper(varargin)
% GUI_CLIPPER MATLAB code for GUI_Clipper.fig
% GUI_CLIPPER, by itself, creates a new GUI_CLIPPER or raises the existing
% singleton*.
%
```

```
%  H = GUI_CLIPPER returns the handle to a new GUI_CLIPPER or the handle to
%  the existing singleton*.
%
%  GUI_CLIPPER('CALLBACK',hObject,eventData,handles,...) calls the local
%  function named CALLBACK in GUI_CLIPPER.M with the given input arguments.
%
%  GUI_CLIPPER('Property','Value',...) creates a new GUI_CLIPPER or raises the
%  existing singleton*.  Starting from the left, property value pairs are
%  applied to the GUI before GUI_Clipper_OpeningFcn gets called.  An
%  unrecognized property name or invalid value makes property application
%  stop.  All inputs are passed to GUI_Clipper_OpeningFcn via varargin.
%
% *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%  instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help GUI_Clipper

% Last Modified by GUIDE v2.5 21-Jul-2021 02:31:33

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @GUI_Clipper_OpeningFcn, ...
                   'gui_OutputFcn',  @GUI_Clipper_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
```

```
% --- Executes just before GUI_Clipper is made visible.
function GUI_Clipper_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to GUI_Clipper (see VARARGIN)

% Choose default command line output for GUI_Clipper
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes GUI_Clipper wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = GUI_Clipper_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


% --- Executes on button press in sinusoidal.
function sinusoidal_Callback(hObject, eventdata, handles)
% hObject    handle to sinusoidal (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
GUI_SinusoidalClippers


% --- Executes on button press in square.
function square_Callback(hObject, eventdata, handles)
% hObject    handle to square (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
GUI_SquareClippers
```

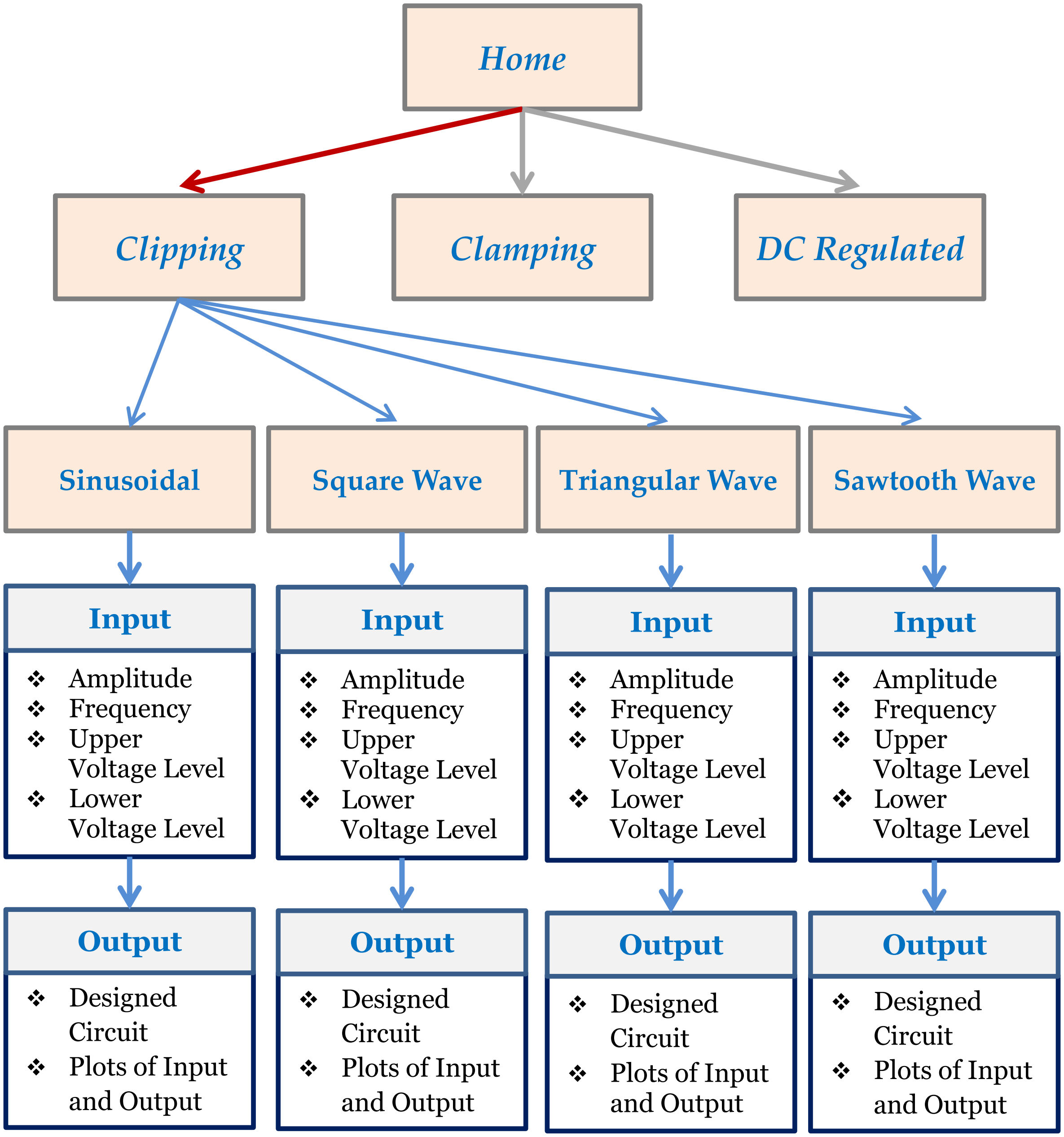
```
% --- Executes on button press in triangular.
function triangular_Callback(hObject, eventdata, handles)
% hObject    handle to triangular (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
GUI_TriangularClippers


% --- Executes on button press in SawtoothWave.
function SawtoothWave_Callback(hObject, eventdata, handles)
% hObject    handle to SawtoothWave (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
GUI_SawtoothClippers
```

Corresponding Window:



- We will work with all four types of sources one by one.

Flow Chart to Prform Clipping :



(i) **Sinusoidal Clippers:**

- For the time being, let us choose “Sinusoidal” source first.
- “GUI_SinusoidalClippers.m” interacts with the user and calls “Sinusoidal_Clipper_Design.m” that contains all necessary functions and codes to create the design and sends data back.

Codes:

“GUI_SinusoidalClippers.m” –

```
function varargout = GUI_SinusoidalClippers(varargin)
% GUI_SINUSOIDALCLIPPERS MATLAB code for GUI_SinusoidalClippers.fig
% GUI_SINUSOIDALCLIPPERS, by itself, creates a new GUI_SINUSOIDALCLIPPERS
or raises the existing
% singleton*.
%
% H = GUI_SINUSOIDALCLIPPERS returns the handle to a new
GUI_SINUSOIDALCLIPPERS or the handle to
% the existing singleton*.
%
% GUI_SINUSOIDALCLIPPERS('CALLBACK',hObject,eventData,handles,...) calls the
local
% function named CALLBACK in GUI_SINUSOIDALCLIPPERS.M with the given
input arguments.
%
% GUI_SINUSOIDALCLIPPERS('Property','Value',...) creates a new
GUI_SINUSOIDALCLIPPERS or raises the
%  existing singleton*.  Starting from the left, property value pairs are
%  applied to the GUI before GUI_SinusoidalClippers_OpeningFcn gets called.  An
%  unrecognized property name or invalid value makes property application
%  stop.  All inputs are passed to GUI_SinusoidalClippers_OpeningFcn via varargin.
%
%  *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%  instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help GUI_SinusoidalClippers
% Last Modified by GUIDE v2.5 15-Jul-2021 02:01:17
```

```
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @GUI_SinusoidalClippers_OpeningFcn, ...
                   'gui_OutputFcn',  @GUI_SinusoidalClippers_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before GUI_SinusoidalClippers is made visible.
function GUI_SinusoidalClippers_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to GUI_SinusoidalClippers (see VARARGIN)

% Choose default command line output for GUI_SinusoidalClippers
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes GUI_SinusoidalClippers wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = GUI_SinusoidalClippers_OutputFcn(hObject, eventdata, handles)
```

```
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

function amplitude_Callback(hObject, eventdata, handles)
% hObject    handle to amplitude (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of amplitude as text
% str2double(get(hObject,'String')) returns contents of amplitude as a double


% --- Executes during object creation, after setting all properties.
function amplitude_CreateFcn(hObject, eventdata, handles)
% hObject    handle to amplitude (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function frequency_Callback(hObject, eventdata, handles)
% hObject    handle to frequency (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of frequency as text
% str2double(get(hObject,'String')) returns contents of frequency as a double
% --- Executes during object creation, after setting all properties.
function frequency_CreateFcn(hObject, eventdata, handles)
% hObject    handle to frequency (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function upper_Callback(hObject, eventdata, handles)
% hObject    handle to upper (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of upper as text
% str2double(get(hObject,'String')) returns contents of upper as a double


% --- Executes during object creation, after setting all properties.
function upper_CreateFcn(hObject, eventdata, handles)
% hObject    handle to upper (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function lower_Callback(hObject, eventdata, handles)
% hObject    handle to lower (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of lower as text
%  str2double(get(hObject,'String')) returns contents of lower as a double

% --- Executes during object creation, after setting all properties.
function lower_CreateFcn(hObject, eventdata, handles)
% hObject    handle to lower (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in design.
function design_Callback(hObject, eventdata, handles)
% hObject    handle to design (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.invalidcheck,'String','');
set(handles.parameters,'String','');
set(handles.v11,'String','');
set(handles.v12,'String','');
set(handles.v13,'String','');
set(handles.v14,'String','');
set(handles.v21,'String','');
set(handles.v22,'String','');
set(handles.v23,'String','');
set(handles.v24,'String','');
axes(handles.axes1);
cla;
axes(handles.axes2);
cla;

amp=str2double(get(handles.amplitude,'String'));
f=str2double(get(handles.frequency,'String'));
h=str2double(get(handles.upper,'String'));
l=str2double(get(handles.lower,'String'));

if h>amp || -l>amp || f<=0 || l>h
   set(handles.invalidcheck,'String','Invalid Choice! Try Again.');
else
   [t,vin,vout,img,v1,v2]=Sinusoidal_Clipper_Design(amp,f,h,l);
   axes(handles.axes1);
```

```
    cla;
    plot(t,vin);
    hold on;
    plot(t,vout);
    axis([0,t(end),(-amp-amp/5),(amp+amp/5)]);
    grid on;
    legend('Vi','Vo');
    xlabel('t ->');
    ylabel('Vi,Vo ->');
    axes(handles.axes2);
    cla;
    imshow(img);
    if v1~=0 && v2==0
        set(handles.parameters,'String','Where,');
        set(handles.v11,'String','V1');
        set(handles.v12,'String','=');
        set(handles.v13,'String',num2str(v1));
        set(handles.v14,'String','V');
    elseif v1~=0 && v2~=0
        set(handles.parameters,'String','Where,');
        set(handles.v11,'String','V1');
        set(handles.v12,'String','=');
        set(handles.v13,'String',num2str(v1));
        set(handles.v14,'String','V');
        set(handles.v21,'String','V2');
        set(handles.v22,'String','=');
        set(handles.v23,'String',num2str(v2));
        set(handles.v24,'String','V');
    end
end

function parameters_Callback(hObject, eventdata, handles)
% hObject    handle to parameters (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of parameters as text
% str2double(get(hObject,'String')) returns contents of parameters as a double
% --- Executes during object creation, after setting all properties.
function parameters_CreateFcn(hObject, eventdata, handles)
```

```
% hObject    handle to parameters (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function v11_Callback(hObject, eventdata, handles)
% hObject    handle to v11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v11 as text
% str2double(get(hObject,'String')) returns contents of v11 as a double

% --- Executes during object creation, after setting all properties.
function v11_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function v12_Callback(hObject, eventdata, handles)
% hObject    handle to v12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v12 as text
%        str2double(get(hObject,'String')) returns contents of v12 as a double

% --- Executes during object creation, after setting all properties.
```

```
function v12_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function v13_Callback(hObject, eventdata, handles)
% hObject    handle to v13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v13 as text
% str2double(get(hObject,'String')) returns contents of v13 as a double


% --- Executes during object creation, after setting all properties.
function v13_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function v21_Callback(hObject, eventdata, handles)
% hObject    handle to v21 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of v21 as text
% str2double(get(hObject,'String')) returns contents of v21 as a double
% --- Executes during object creation, after setting all properties.
```

```
function v21_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v21 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function v22_Callback(hObject, eventdata, handles)
% hObject    handle to v22 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v22 as text
% str2double(get(hObject,'String')) returns contents of v22 as a double

% --- Executes during object creation, after setting all properties.
function v22_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v22 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function v23_Callback(hObject, eventdata, handles)
% hObject    handle to v23 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v23 as text
% str2double(get(hObject,'String')) returns contents of v23 as a double
```

```
% --- Executes during object creation, after setting all properties.

function v23_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v23 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function invalidcheck_Callback(hObject, eventdata, handles)
% hObject    handle to invalidcheck (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of invalidcheck as text
% str2double(get(hObject,'String')) returns contents of invalidcheck as a double


% --- Executes during object creation, after setting all properties.
function invalidcheck_CreateFcn(hObject, eventdata, handles)
% hObject    handle to invalidcheck (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function v14_Callback(hObject, eventdata, handles)
% hObject    handle to v14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of v14 as text
% str2double(get(hObject,'String')) returns contents of v14 as a double

% --- Executes during object creation, after setting all properties.
function v14_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end


function v24_Callback(hObject, eventdata, handles)
% hObject    handle to v24 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v24 as text
% str2double(get(hObject,'String')) returns contents of v24 as a double


% --- Executes during object creation, after setting all properties.
function v24_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v24 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end
```

“Sinusoidal_Clipper_Design.m” –

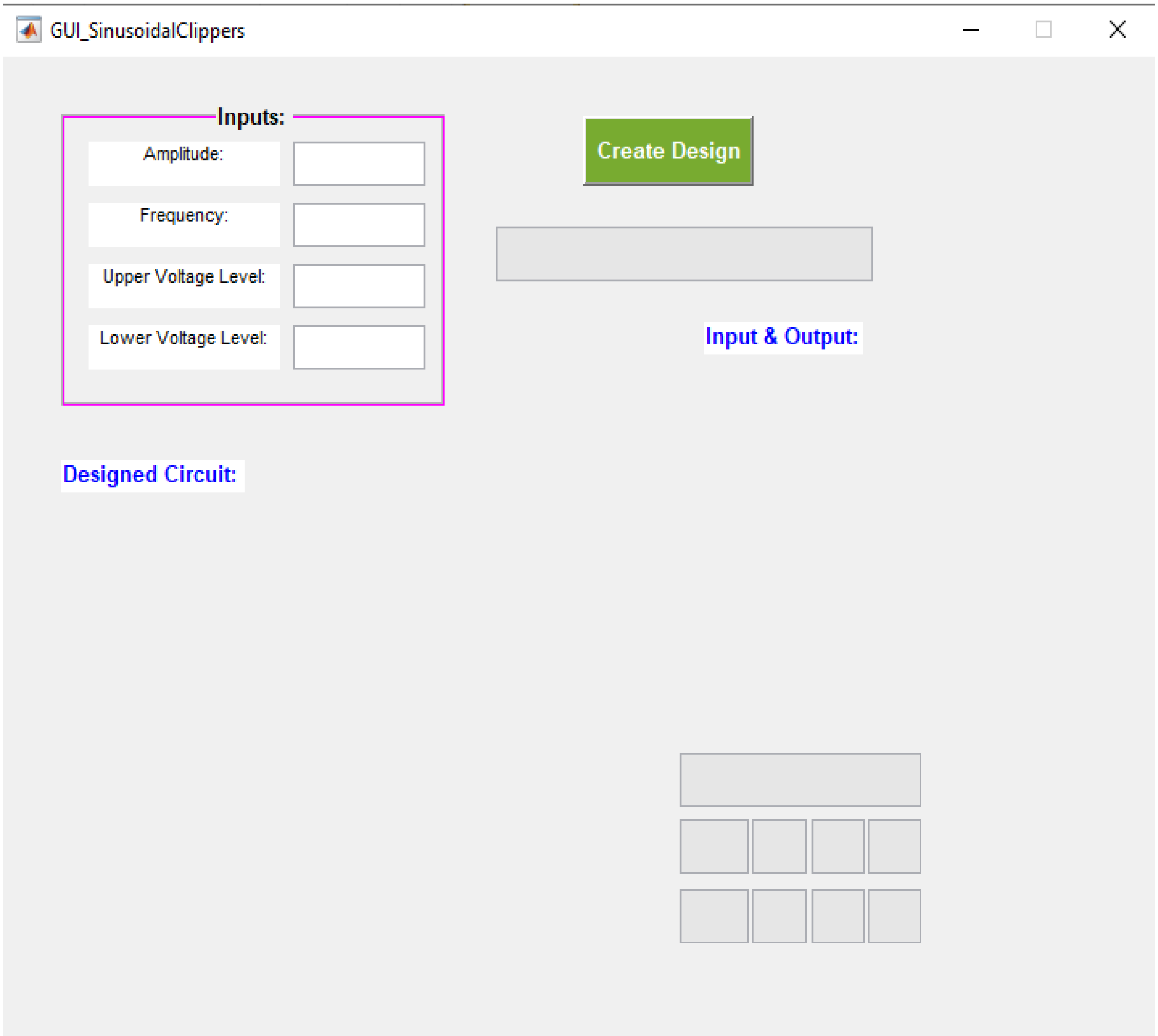
```
function [t,vin,vout,img,v1,v2]=Sinusoidal_Clipper_Design(vamp,f,vup,vlow)
   if vlow==0 && vup==vamp
      [t,vin,vout,img,v1,v2]=negative_clipper(vamp,f,vup,vlow);
   elseif vup==0 && vlow==-(vamp)
      [t,vin,vout,img,v1,v2]=positive_clipper(vamp,f,vup,vlow);
   elseif vlow==0 && vup==0
      [t,vin,vout,img,v1,v2]=both_negative_positive_clipper(vamp,f,vup,vlow);
   elseif 0<vup<vamp && vlow==-(vamp)
      [t,vin,vout,img,v1,v2]=partial_positive_clipper(vamp,f,vup,vlow);
   elseif 0<-(vlow)<vamp && vup==vamp
      [t,vin,vout,img,v1,v2]=partial_negative_clipper(vamp,f,vup,vlow);
   elseif 0<vup<vamp && vlow==0
      [t,vin,vout,img,v1,v2]=partial_positive_full_negative_clipper(vamp,f,vup,vlow);
   elseif vup==0 && 0<-(vlow)<vamp
      [t,vin,vout,img,v1,v2]=partial_negative_full_positive_clipper(vamp,f,vup,vlow);
   elseif 0<vup<vamp && 0<-(vlow)<vamp
      [t,vin,vout,img,v1,v2]=partial_negative_positive_clipper(vamp,f,vup,vlow);
   end
end
function [t,vin,vout,img,v1,v2]=negative_clipper(vamp,f,vup,vlow)
   v1=0;
   v2=0;
   T=1/f;
   t=linspace(0,2*T,1000);
   vin=vamp.*sin((2*pi*f)*t);
   vout=[];
   for i=1:1000
      if vin(i)>=0
         vout(i)=vin(i);
      else
         vout(i)=0;
      end
   end
   img=imread('SinusoidalClipper1.PNG');
end

function [t,vin,vout,img,v1,v2]=positive_clipper(vamp,f,vup,vlow)
   v1=0;
```

```
    v2=0;
    T=1/f;
    t=linspace(0,2*T,1000);
    vin=vamp.*sin((2*pi*f)*t);
    vout=[];
    for i=1:1000
        if vin(i)<=0
            vout(i)=vin(i);
        else
            vout(i)=0;
        end
    end
    img=imread('SinusoidalClipper2.PNG');
end

function [t,vin,vout,img,v1,v2]=partial_positive_clipper(vamp,f,vup,vlow)
    v1=vup;
    v2=0;
    T=1/f;
    t=linspace(0,2*T,1000);
    vin=vamp.*sin((2*pi*f)*t);
    vout=[];
    for i=1:1000
        if vin(i)>=vup
            vout(i)=v1;
        else
            vout(i)=vin(i);
        end
    end
    img=imread('SinusoidalClipper3.PNG');
end

function [t,vin,vout,img,v1,v2]=partial_negative_clipper(vamp,f,vup,vlow)
    v1=-vlow;
    v2=0;
    T=1/f;
    t=linspace(0,2*T,1000);
    vin=vamp.*sin((2*pi*f)*t);
    vout=[];
    for i=1:1000
```

```
    if vin(i)<=vlow
      vout(i)=-v1;
    else
      vout(i)=vin(i);
    end
  end
  img=imread('SinusoidalClipper4.PNG');
end

function [t,vin,vout,img,v1,v2]=partial_positive_full_negative_clipper(vamp,f,vup,vlow)
  v1=vup;
  v2=0;
  T=1/f;
  t=linspace(0,2*T,1000);
  vin=vamp.*sin((2*pi*f)*t);
  vout=[];
  for i=1:1000
    if vin(i)>=vup
      vout(i)=v1;
    elseif vin(i)<0
      vout(i)=0;
    else
      vout(i)=vin(i);
    end
  end
  img=imread('SinusoidalClipper5.PNG');
end

function [t,vin,vout,img,v1,v2]=partial_negative_full_positive_clipper(vamp,f,vup,vlow)
  v1=-(vlow);
  v2=0;
  T=1/f;
  t=linspace(0,2*T,1000);
  vin=vamp.*sin((2*pi*f)*t);
  vout=[];
  for i=1:1000
    if vin(i)>=0
      vout(i)=0;
    elseif vin(i)<=vlow
      vout(i)=-v1;
```

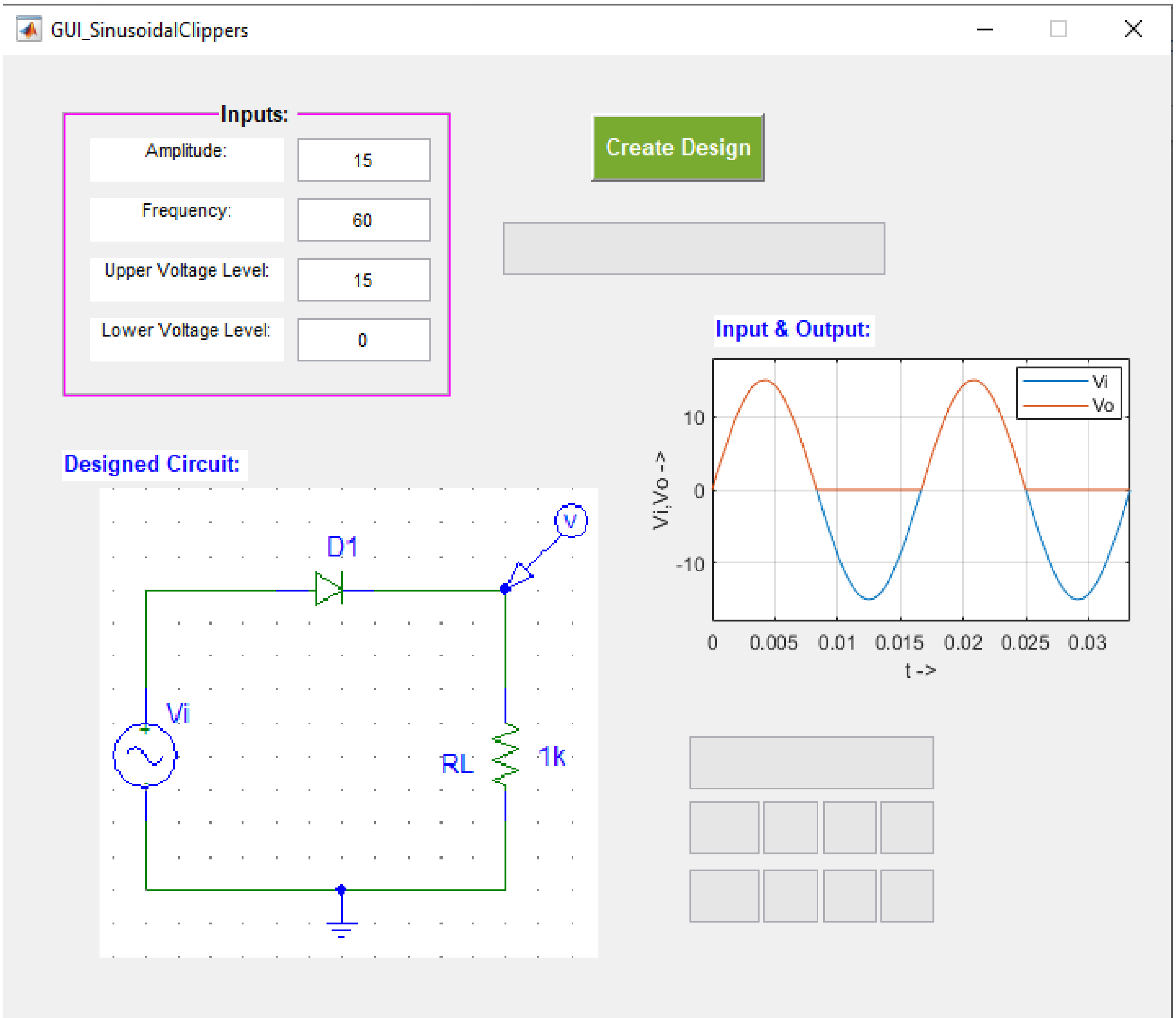
```
        else
            vout(i)=vin(i);
        end
    end
    img=imread('SinusoidalClipper6.PNG');
end

function [t,vin,vout,img,v1,v2]=partial_negative_positive_clipper(vamp,f,vup,vlow)
    v1=-(vlow);
    v2=vup;
    T=1/f;
    t=linspace(0,2*T,1000);
    vin=vamp.*sin((2*pi*f)*t);
    vout=[];
    for i=1:1000
        if vin(i)>=vup
            vout(i)=v2;
        elseif vin(i)<=vlow
            vout(i)=-v1;
        else
            vout(i)=vin(i);
        end
    end
    img=imread('SinusoidalClipper7.PNG');
end

function [t,vin,vout,img,v1,v2]=both_negative_positive_clipper(vamp,f,vup,vlow)
    v1=0;
    v2=0;
    T=1/f;
    t=linspace(0,2*T,1000);
    vin=vamp.*sin((2*pi*f)*t);
    vout=[];
    for i=1:1000
        vout(i)=0;
    end
    img=imread('SinusoidalClipper8.PNG');
end
```

Corresponding Window:

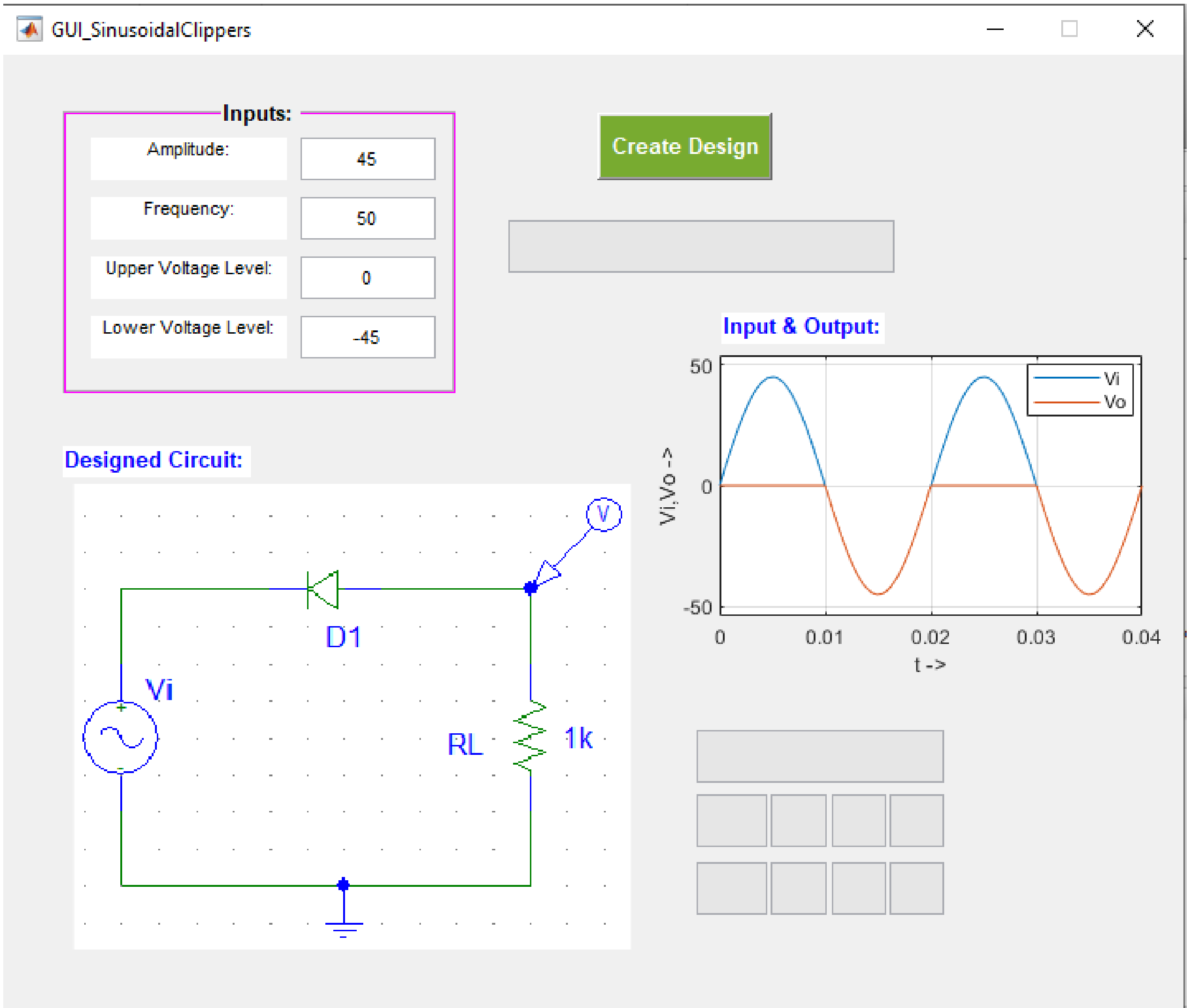


Sample Outputs:

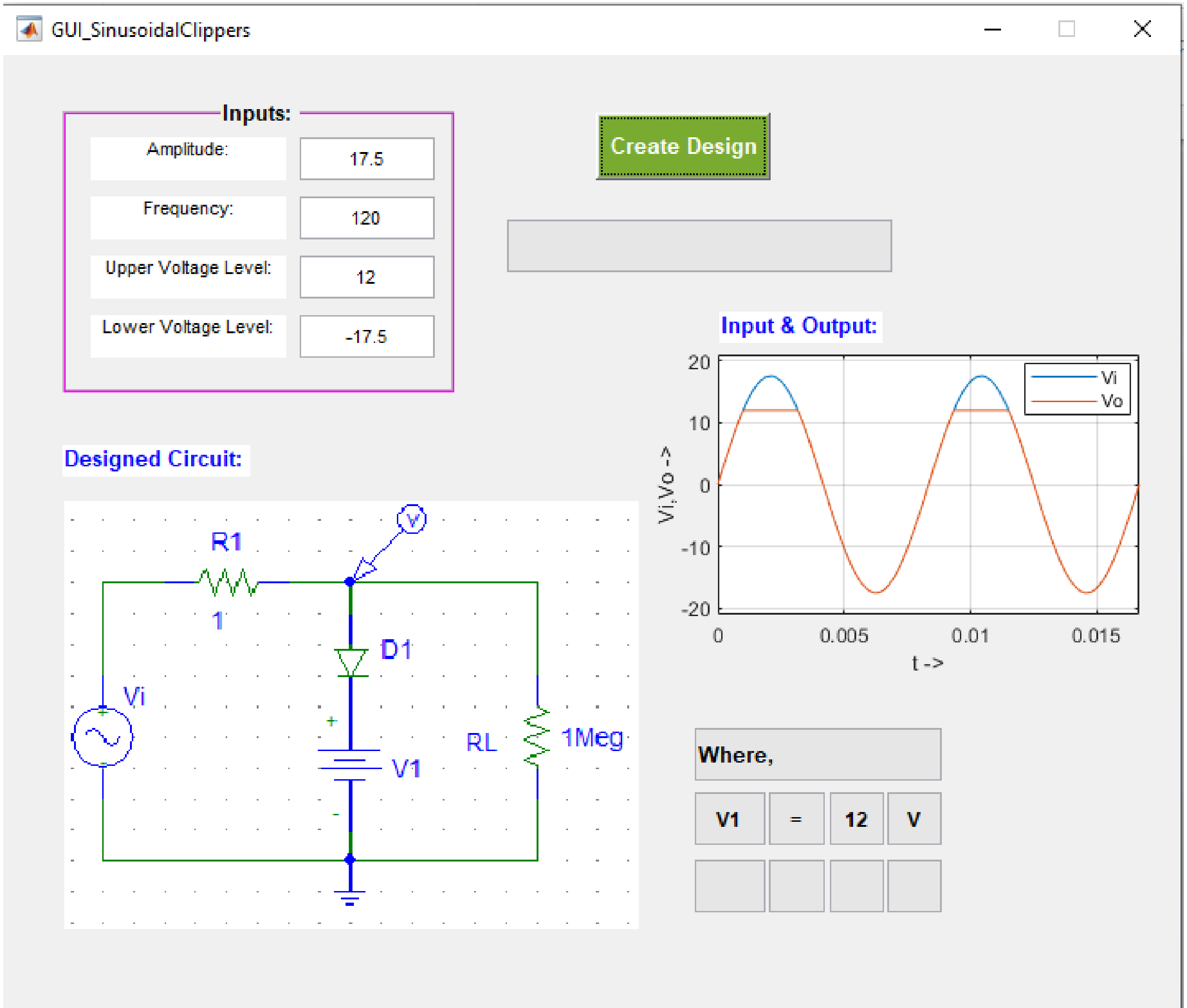
1. Negative Clipper :



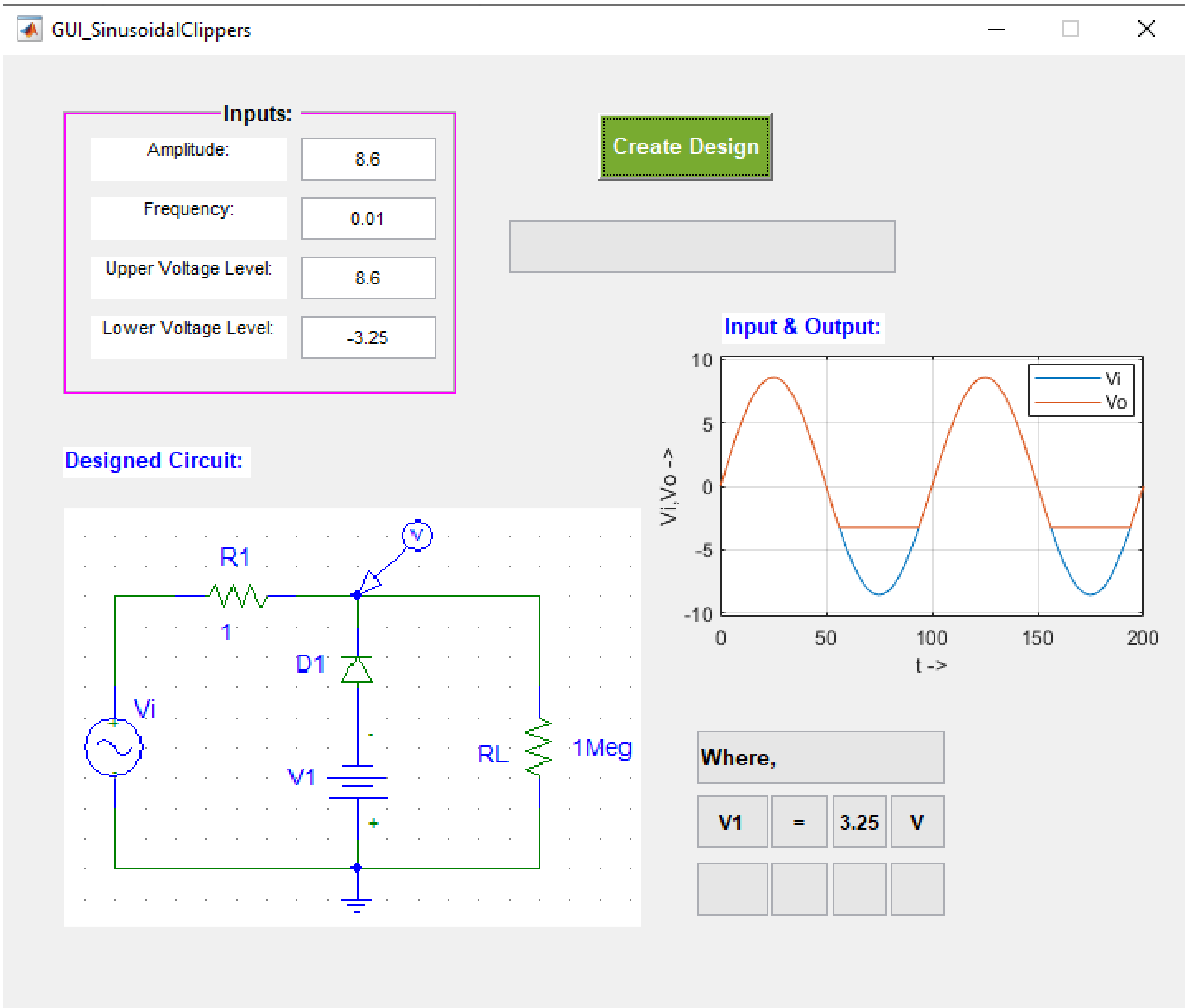
2. Positive Clipper :



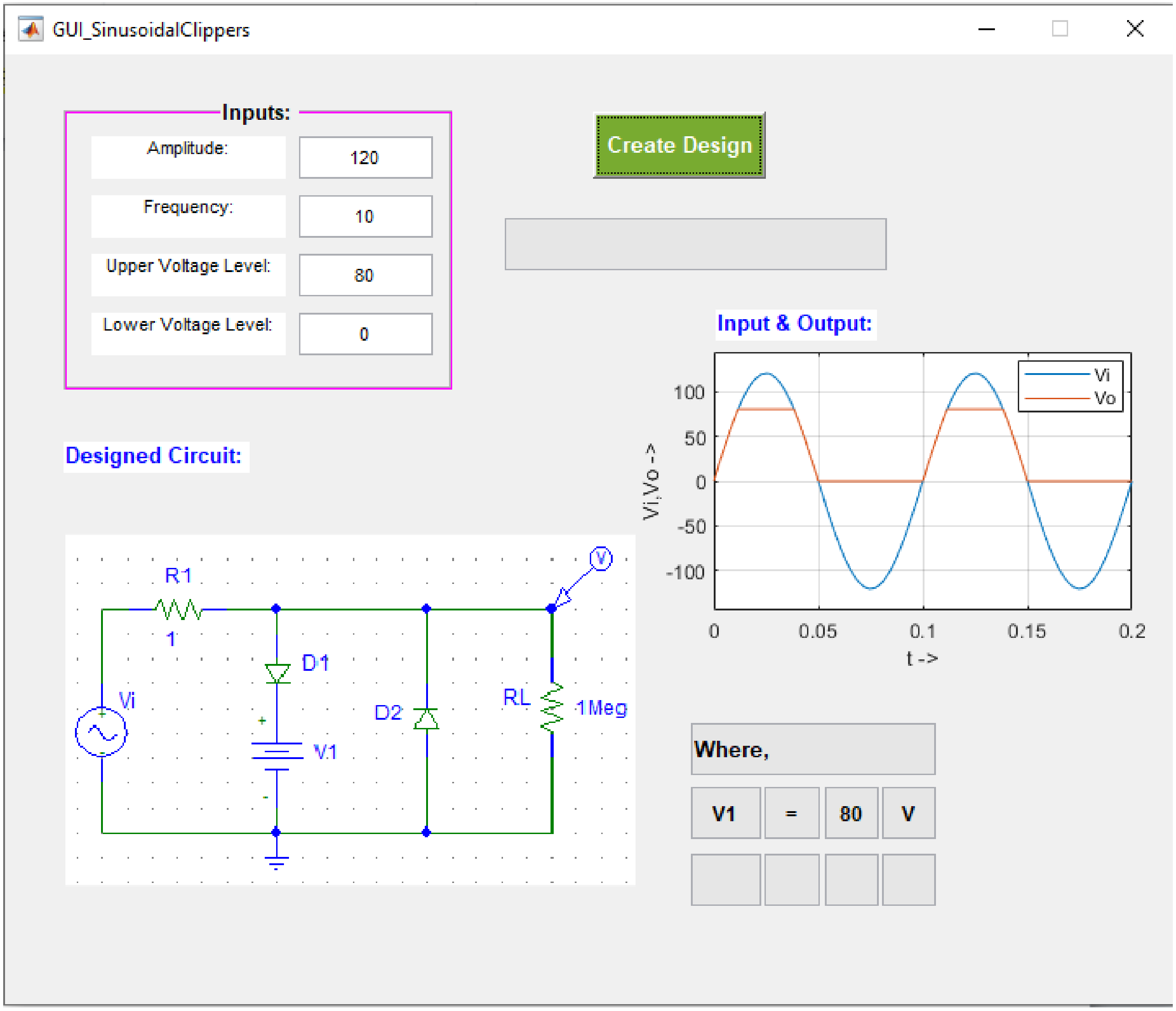
3. Clipping Positive Side Partially:



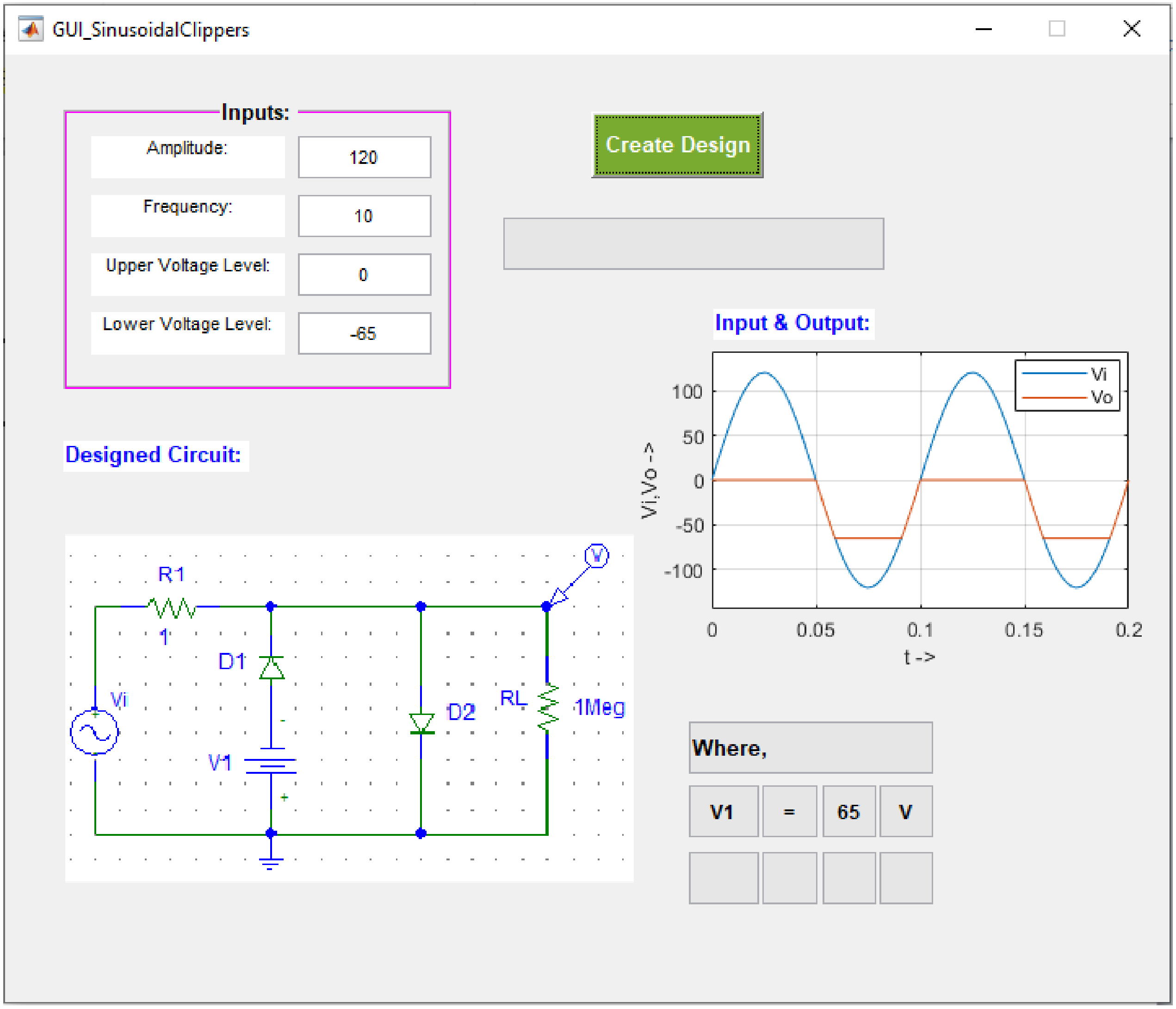
4. Clipping Negative Side Partially:



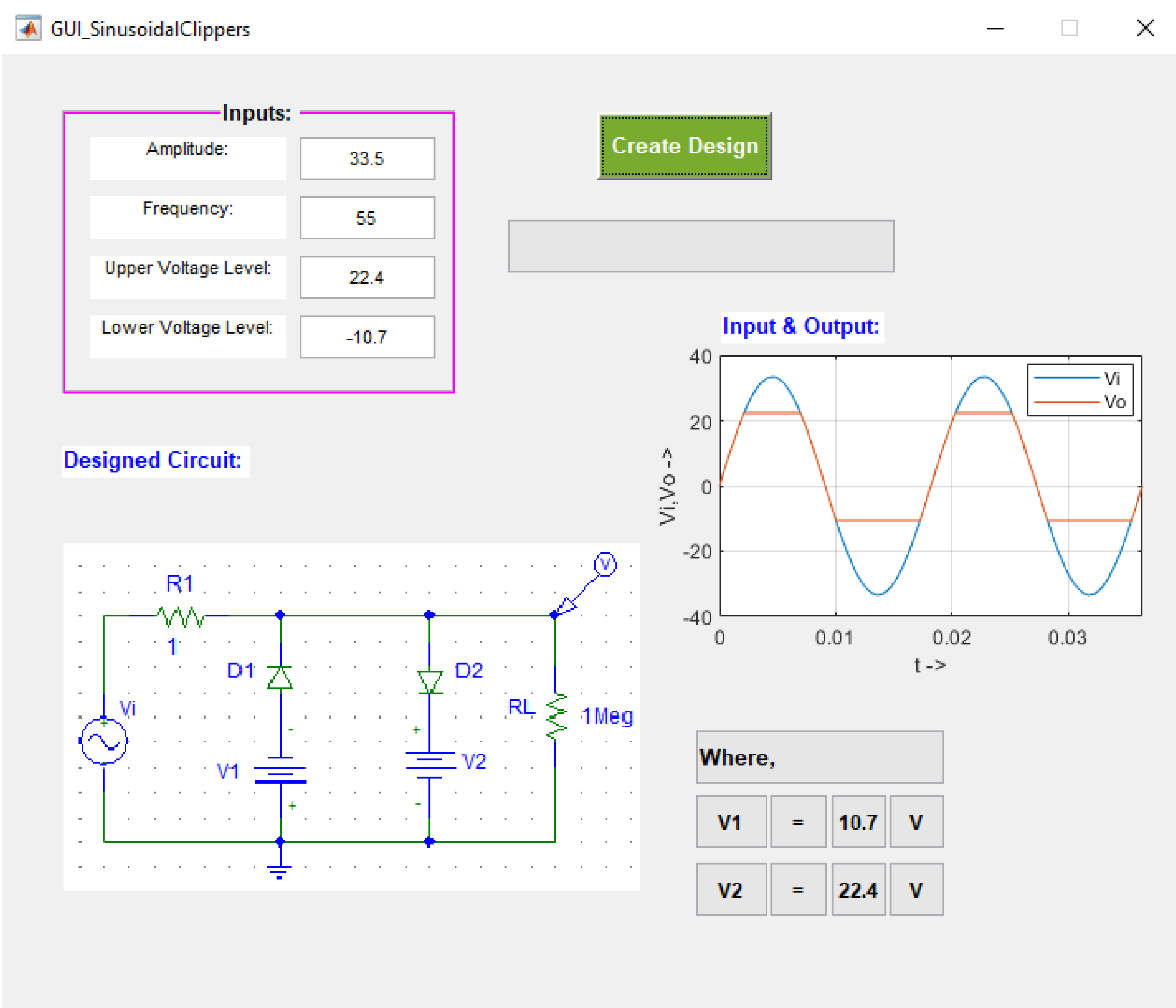
5. **Positive Side is Partially Clipped and Negative Side is Fully Clipped :**



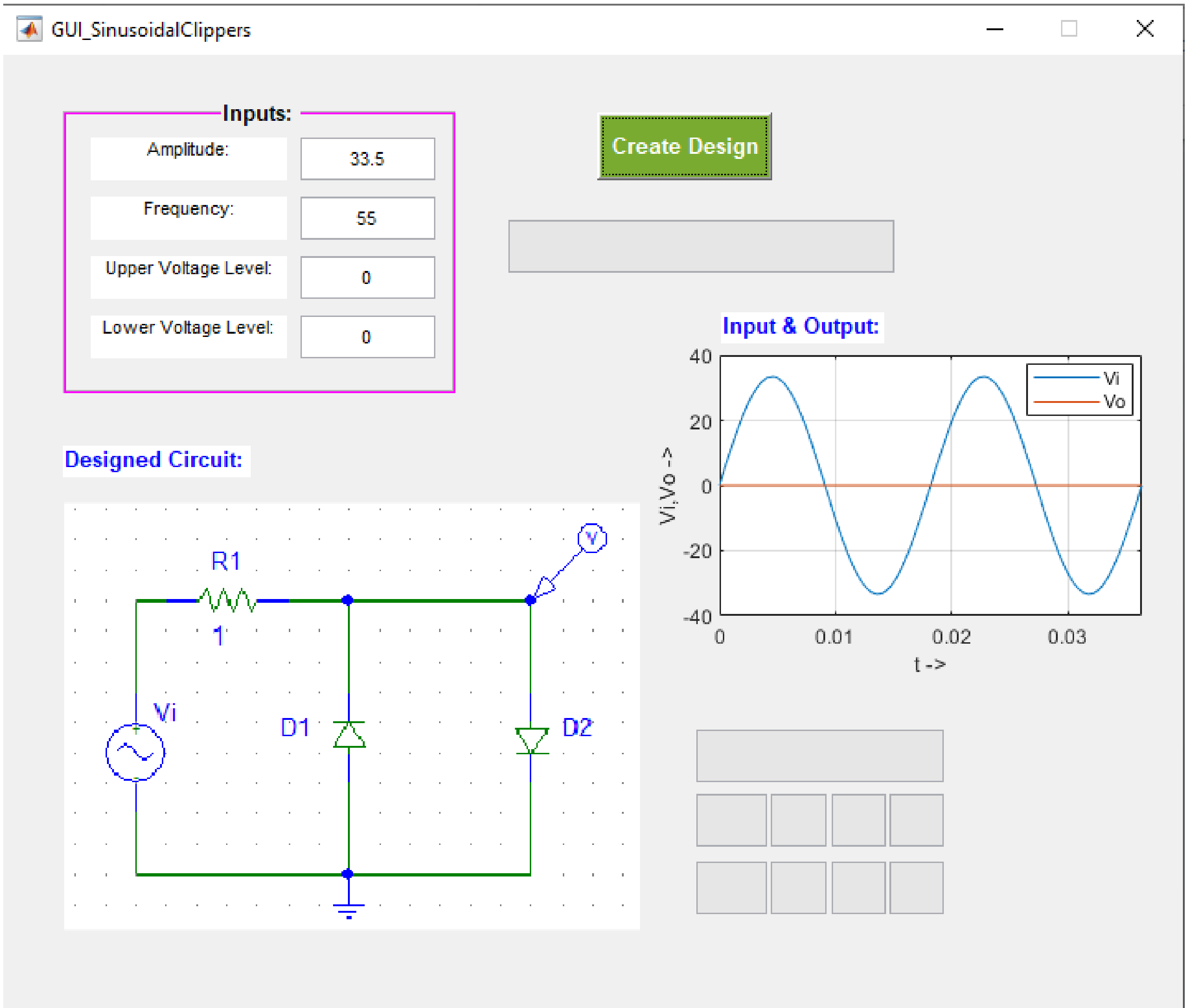
6. Negative Side is Partially Clipped and Positive Side is Fully Clipped :



7. **Both Positive and Negative Side are Partially Clipped:**

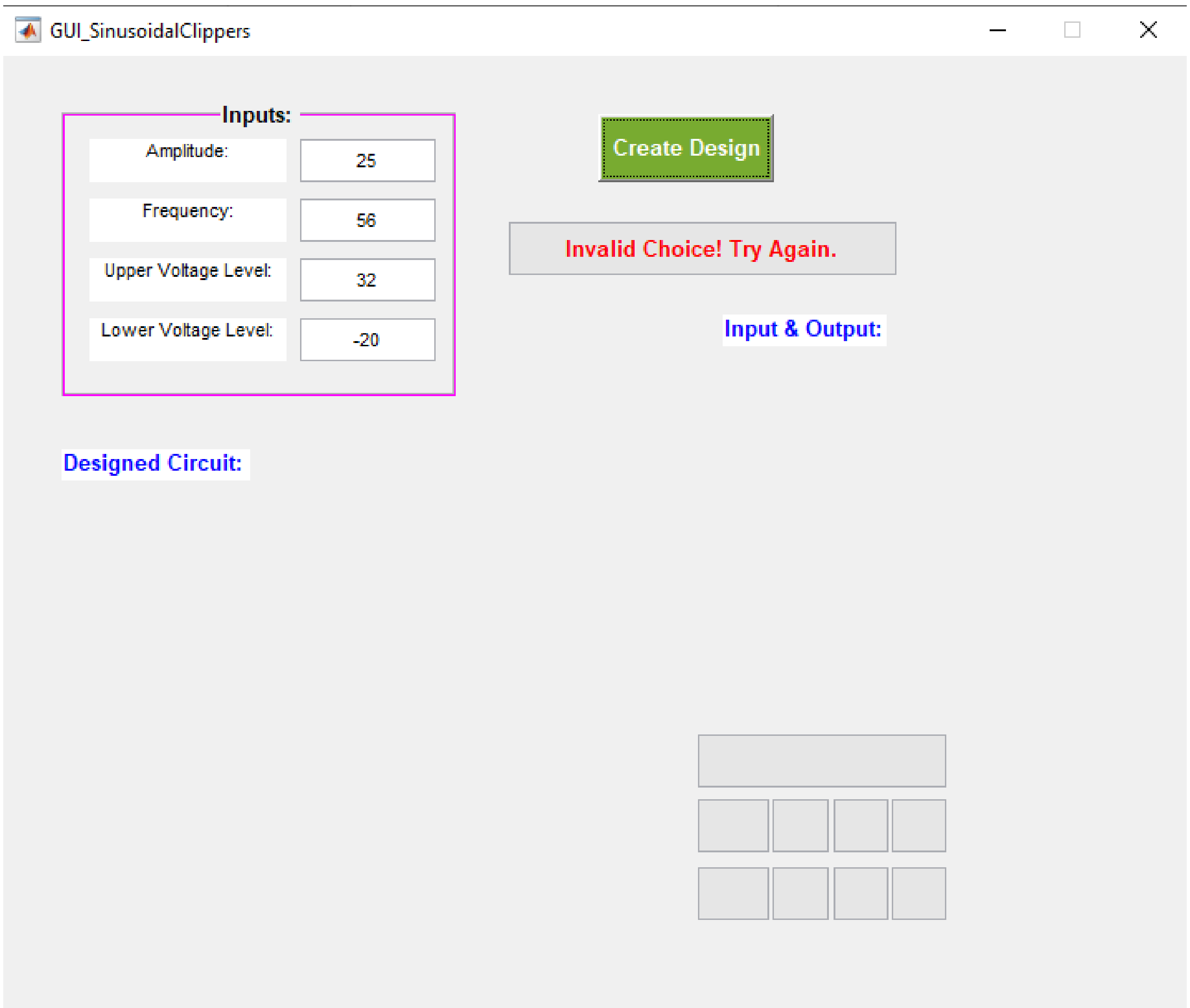


8. Both Positive and Negative Side are Fully Clipped:

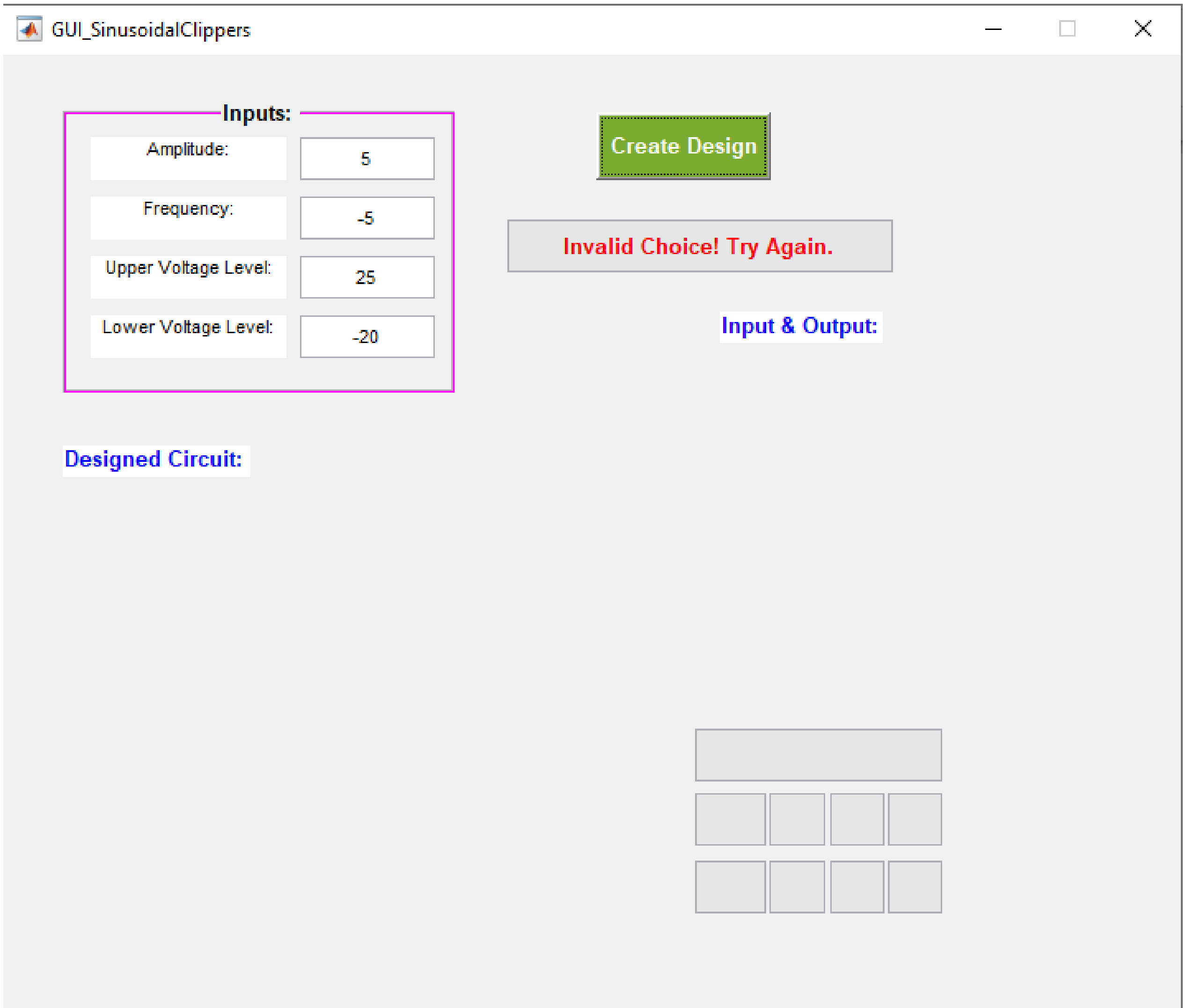


9. Invalid Cases:

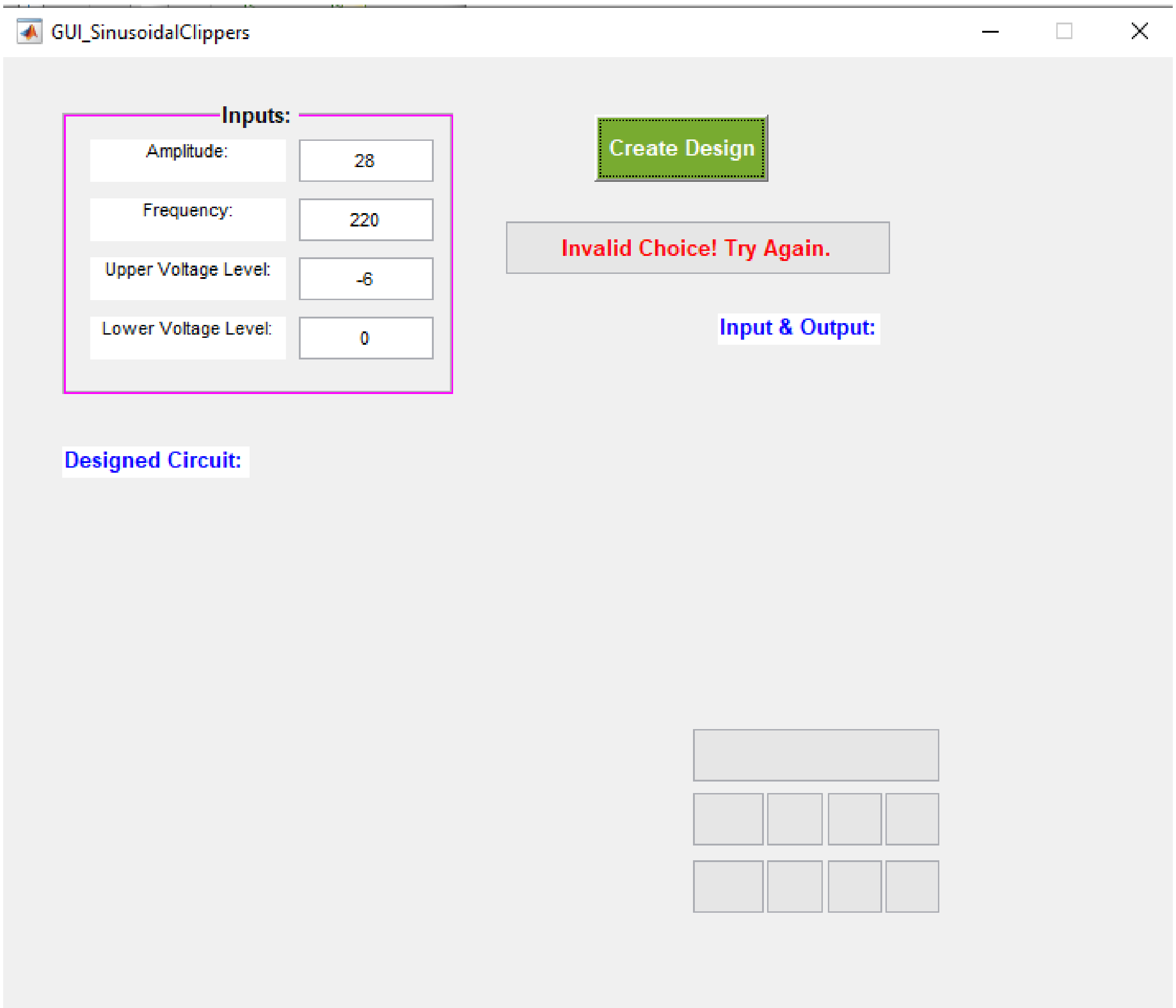
i. Clipped Voltage Level is Greater than Input Voltage Peak :



ii. Zero/ Negative Frequency:



iii. Lower Voltage Level is given greater than Upper Voltage Level:



N. B: Our program is able to design the circuit for any other kind of inputs as well. User is appreciated to check for more as he desires.

(ii) **Square Clippers:**

- Now, let us choose "Square Wave" source from "GUI_Clipper" window.
- "GUI_SquareClippers.m" interacts with the user and calls "SquareWave_Clipper_Design.m" that contains all necessary functions and codes to create the design and sends data back.

Codes:

"GUI_SquareClippers.m" –

```
function varargout = GUI_SquareClippers(varargin)
% GUI_SQUARECLIPPERS MATLAB code for GUI_SquareClippers.fig
% GUI_SQUARECLIPPERS, by itself, creates a new GUI_SQUARECLIPPERS or raises
the existing
% singleton*.
%
% H = GUI_SQUARECLIPPERS returns the handle to a new GUI_SQUARECLIPPERS
or the handle to
% the existing singleton*.
%
%  GUI_SQUARECLIPPERS('CALLBACK',hObject,eventData,handles,...) calls the local
%  function named CALLBACK in GUI_SQUARECLIPPERS.M with the given input
arguments.
%
%  GUI_SQUARECLIPPERS('Property','Value',...) creates a new
GUI_SQUARECLIPPERS or raises the
%  existing singleton*.  Starting from the left, property value pairs are
%  applied to the GUI before GUI_SquareClippers_OpeningFcn gets called.  An
%  unrecognized property name or invalid value makes property application
%  stop.  All inputs are passed to GUI_SquareClippers_OpeningFcn via varargin.
%
% *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%  instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help GUI_SquareClippers

% Last Modified by GUIDE v2.5 14-Jul-2021 17:51:54
```

```
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @GUI_SquareClippers_OpeningFcn, ...
                   'gui_OutputFcn',  @GUI_SquareClippers_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before GUI_SquareClippers is made visible.
function GUI_SquareClippers_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to GUI_SquareClippers (see VARARGIN)

% Choose default command line output for GUI_SquareClippers
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes GUI_SquareClippers wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
```

```
function varargout = GUI_SquareClippers_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;



% --- Executes on button press in design.
function design_Callback(hObject, eventdata, handles)
% hObject    handle to design (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.invalid_check,'String','');
set(handles.properties,'String','');
set(handles.v11,'String','');
set(handles.v12,'String','');
set(handles.v13,'String','');
set(handles.v14,'String','');
set(handles.v21,'String','');
set(handles.v22,'String','');
set(handles.v23,'String','');
set(handles.v24,'String','');
axes(handles.diagram);
cla;
axes(handles.plots);
cla;

amp=str2double(get(handles.amplitude,'String'));
f=str2double(get(handles.frequency,'String'));
h=str2double(get(handles.upper,'String'));
l=str2double(get(handles.lower,'String'));

if h>amp || -l>amp || f<=0 || l>h
   set(handles.invalid_check,'String','Invalid Choice! Try Again.');
else
   [t,vin,vout,img,v1,v2]=SquareWave_Clipper_Design(amp,f,h,l);
   axes(handles.plots);
```

```
    cla;
    plot(t,vin);
    hold on;
    plot(t,vout);
    axis([0,t(end),(-amp-amp/5),(amp+amp/5)]);
    grid on;
    legend('Vi','Vo');
    xlabel('t ->');
    ylabel('Vi,Vo ->');
    axes(handles.diagram);
    cla;
    imshow(img);
    if v1~=0 && v2==0
        set(handles.properties,'String','Where,');
        set(handles.v11,'String','V1');
        set(handles.v12,'String','=');
        set(handles.v13,'String',num2str(v1));
        set(handles.v14,'String','V');
    elseif v1~=0 && v2~=0
        set(handles.properties,'String','Where,');
        set(handles.v11,'String','V1');
        set(handles.v12,'String','=');
        set(handles.v13,'String',num2str(v1));
        set(handles.v14,'String','V');
        set(handles.v21,'String','V2');
        set(handles.v22,'String','=');
        set(handles.v23,'String',num2str(v2));
        set(handles.v24,'String','V');
    end
end


function properties_Callback(hObject, eventdata, handles)
% hObject    handle to properties (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of properties as text
% str2double(get(hObject,'String')) returns contents of properties as a double
```

```
% --- Executes during object creation, after setting all properties.
function properties_CreateFcn(hObject, eventdata, handles)
% hObject    handle to properties (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function v11_Callback(hObject, eventdata, handles)
% hObject    handle to v11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v11 as text
%        str2double(get(hObject,'String')) returns contents of v11 as a double


% --- Executes during object creation, after setting all properties.
function v11_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function v14_Callback(hObject, eventdata, handles)
% hObject    handle to v14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v14 as text
% str2double(get(hObject,'String')) returns contents of v14 as a double

% --- Executes during object creation, after setting all properties.
function v14_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function v13_Callback(hObject, eventdata, handles)
% hObject    handle to v13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v13 as text
% str2double(get(hObject,'String')) returns contents of v13 as a double


% --- Executes during object creation, after setting all properties.
function v13_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function v21_Callback(hObject, eventdata, handles)
% hObject    handle to v21 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v21 as text
% str2double(get(hObject,'String')) returns contents of v21 as a double


% --- Executes during object creation, after setting all properties.
function v21_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v21 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function v22_Callback(hObject, eventdata, handles)
% hObject    handle to v22 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v22 as text
% str2double(get(hObject,'String')) returns contents of v22 as a double


% --- Executes during object creation, after setting all properties.
function v22_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v22 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function v23_Callback(hObject, eventdata, handles)
% hObject    handle to v23 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v23 as text
% str2double(get(hObject,'String')) returns contents of v23 as a double

% --- Executes during object creation, after setting all properties.
function v23_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v23 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function invalid_check_Callback(hObject, eventdata, handles)
% hObject    handle to invalid_check (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of invalid_check as text
% str2double(get(hObject,'String')) returns contents of invalid_check as a double
% --- Executes during object creation, after setting all properties.
function invalid_check_CreateFcn(hObject, eventdata, handles)
% hObject    handle to invalid_check (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
```

```
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end


function amplitude_Callback(hObject, eventdata, handles)
% hObject    handle to amplitude (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of amplitude as text
% str2double(get(hObject,'String')) returns contents of amplitude as a double


% --- Executes during object creation, after setting all properties.
function amplitude_CreateFcn(hObject, eventdata, handles)
% hObject    handle to amplitude (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function frequency_Callback(hObject, eventdata, handles)
% hObject    handle to frequency (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of frequency as text
% str2double(get(hObject,'String')) returns contents of frequency as a double

% --- Executes during object creation, after setting all properties.
function frequency_CreateFcn(hObject, eventdata, handles)
```

```
% hObject    handle to frequency (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function upper_Callback(hObject, eventdata, handles)
% hObject    handle to upper (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of upper as text
% str2double(get(hObject,'String')) returns contents of upper as a double


% --- Executes during object creation, after setting all properties.
function upper_CreateFcn(hObject, eventdata, handles)
% hObject    handle to upper (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end
function lower_Callback(hObject, eventdata, handles)
% hObject    handle to lower (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of lower as text
% str2double(get(hObject,'String')) returns contents of lower as a double
```

```
% --- Executes during object creation, after setting all properties.
function lower_CreateFcn(hObject, eventdata, handles)
% hObject    handle to lower (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end


function v12_Callback(hObject, eventdata, handles)
% hObject    handle to v12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v12 as text
% str2double(get(hObject,'String')) returns contents of v12 as a double

% --- Executes during object creation, after setting all properties.
function v12_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function v24_Callback(hObject, eventdata, handles)
% hObject    handle to v24 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of v24 as text
%        str2double(get(hObject,'String')) returns contents of v24 as a double

% --- Executes during object creation, after setting all properties.
function v24_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v24 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end
```

"SquareWave_Clipper_Design.m" –

```
function [t,vin,vout,img,v1,v2]=SquareWave_Clipper_Design(vamp,f,vup,vlow)
   if vlow==0 && vup==vamp
      [t,vin,vout,img,v1,v2]=negative_clipper(vamp,f,vup,vlow);
   elseif vup==0 && vlow==-(vamp)
      [t,vin,vout,img,v1,v2]=positive_clipper(vamp,f,vup,vlow);
   elseif vlow==0 && vup==0
      [t,vin,vout,img,v1,v2]=both_negative_positive_clipper(vamp,f,vup,vlow);
   elseif 0<vup<vamp && vlow==-(vamp)
      [t,vin,vout,img,v1,v2]=partial_positive_clipper(vamp,f,vup,vlow);
   elseif 0<-(vlow)<vamp && vup==vamp
      [t,vin,vout,img,v1,v2]=partial_negative_clipper(vamp,f,vup,vlow);
   elseif 0<vup<vamp && vlow==0
      [t,vin,vout,img,v1,v2]=partial_positive_full_negative_clipper(vamp,f,vup,vlow);
   elseif vup==0 && 0<-(vlow)<vamp
      [t,vin,vout,img,v1,v2]=partial_negative_full_positive_clipper(vamp,f,vup,vlow);
   elseif 0<vup<vamp && 0<-(vlow)<vamp
      [t,vin,vout,img,v1,v2]=partial_negative_positive_clipper(vamp,f,vup,vlow);
   end
end
```

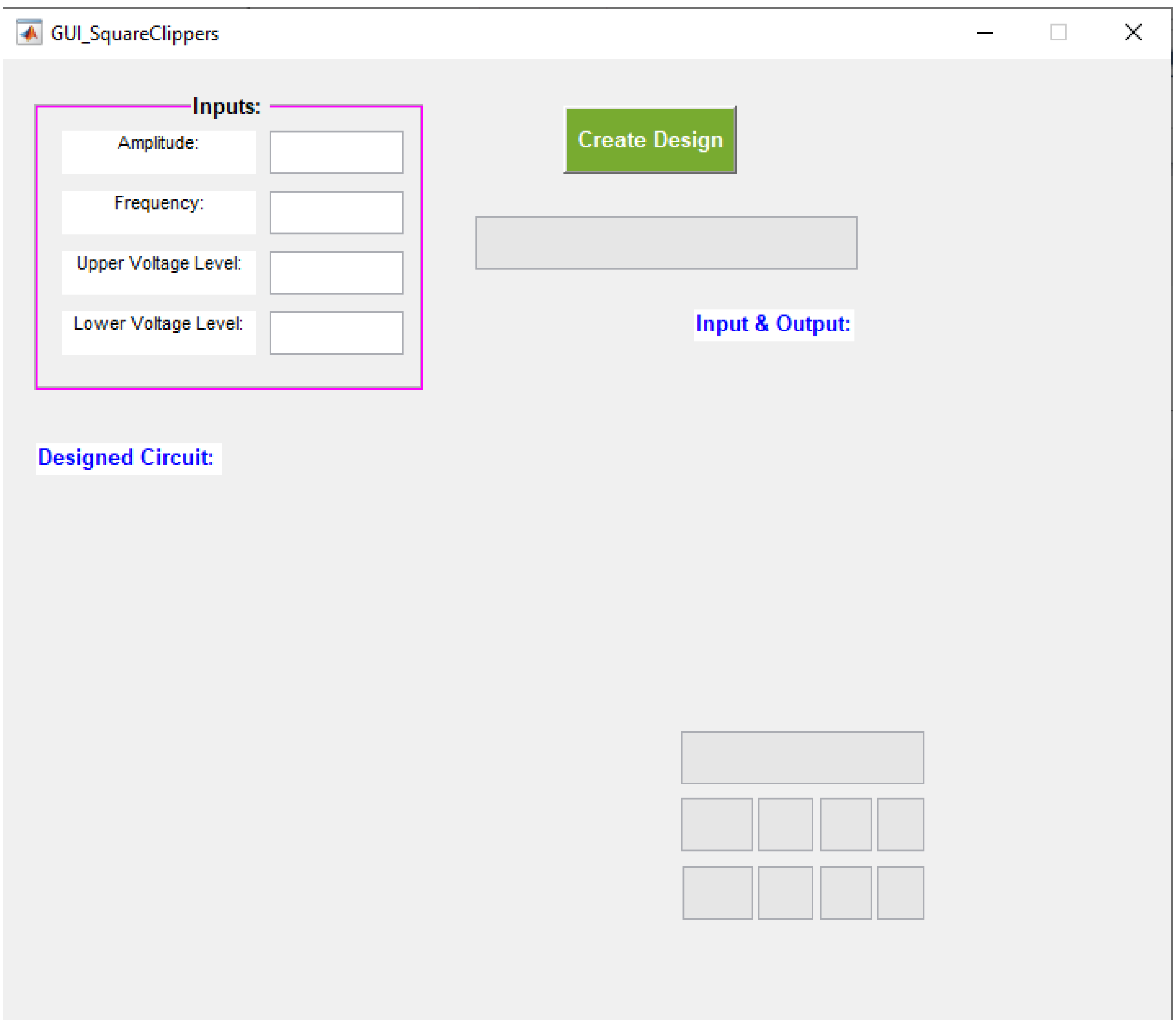
```
function [t,vin,vout,img,v1,v2]=positive_clipper(vamp,f,vup,vlow)
    v1=0;
    v2=0;
    T=1/f;
    t=linspace(0,2*T,1000);
    vin=vamp.*square((2*pi*f)*t);
    vout=[];
    for i=1:1000
        if vin(i)<=0
            vout(i)=vin(i);
        else
            vout(i)=0;
        end
    end
    img=imread('SquareClipper1.PNG');
end

function [t,vin,vout,img,v1,v2]=negative_clipper(vamp,f,vup,vlow)
    v1=0;
    v2=0;
    T=1/f;
    t=linspace(0,2*T,1000);
    vin=vamp.*square((2*pi*f)*t);
    vout=[];
    for i=1:1000
        if vin(i)>=0
            vout(i)=vin(i);
        else
            vout(i)=0;
        end
    end
    img=imread('SquareClipper2.PNG');
end

function [t,vin,vout,img,v1,v2]=both_negative_positive_clipper(vamp,f,vup,vlow)
    v1=0;
    v2=0;
    T=1/f;
    t=linspace(0,2*T,1000);
    vin=vamp.*square((2*pi*f)*t);
```

```
    vout=[];
    for i=1:1000
        vout(i)=0;
    end
    img=imread('SquareClipper8.PNG');
end

function [t,vin,vout,img,v1,v2]=partial_positive_clipper(vamp,f,vup,vlow)
    v1=vup;
    v2=0;
    T=1/f;
    t=linspace(0,2*T,1000);
    vin=vamp.*square((2*pi*f)*t);
    vout=[];
    for i=1:1000
        if vin(i)>=0
            vout(i)=v1;
        else
            vout(i)=vin(i);
        end
    end
    img=imread('SquareClipper3.PNG');
end

function [t,vin,vout,img,v1,v2]=partial_negative_clipper(vamp,f,vup,vlow)
    v1=-vlow;
    v2=0;
    T=1/f;
    t=linspace(0,2*T,1000);
    vin=vamp.*square((2*pi*f)*t);
    vout=[];
    for i=1:1000
        if vin(i)<=0
            vout(i)=-v1;
        else
            vout(i)=vin(i);
        end
    end
    img=imread('SquareClipper4.PNG');
end
```

```
function [t,vin,vout,img,v1,v2]=partial_positive_full_negative_clipper(vamp,f,vup,vlow)
   v1=vup;
   v2=0;
   T=1/f;
   t=linspace(0,2*T,1000);
   vin=vamp.*square((2*pi*f)*t);
   vout=[];
   for i=1:1000
      if vin(i)>=0
         vout(i)=v1;
      else
         vout(i)=0;
      end
   end
   img=imread('SquareClipper5.PNG');
end

function [t,vin,vout,img,v1,v2]=partial_negative_full_positive_clipper(vamp,f,vup,vlow)
   v1=-(vlow);
   v2=0;
   T=1/f;
   t=linspace(0,2*T,1000);
   vin=vamp.*square((2*pi*f)*t);
   vout=[];
   for i=1:1000
      if vin(i)>=0
         vout(i)=0;
      else
         vout(i)=-v1;
      end
   end
   img=imread('SquareClipper6.PNG');
end

function [t,vin,vout,img,v1,v2]=partial_negative_positive_clipper(vamp,f,vup,vlow)
   v1=-(vlow);
   v2=vup;
   T=1/f;
   t=linspace(0,2*T,1000);
   vin=vamp.*square((2*pi*f)*t);
```

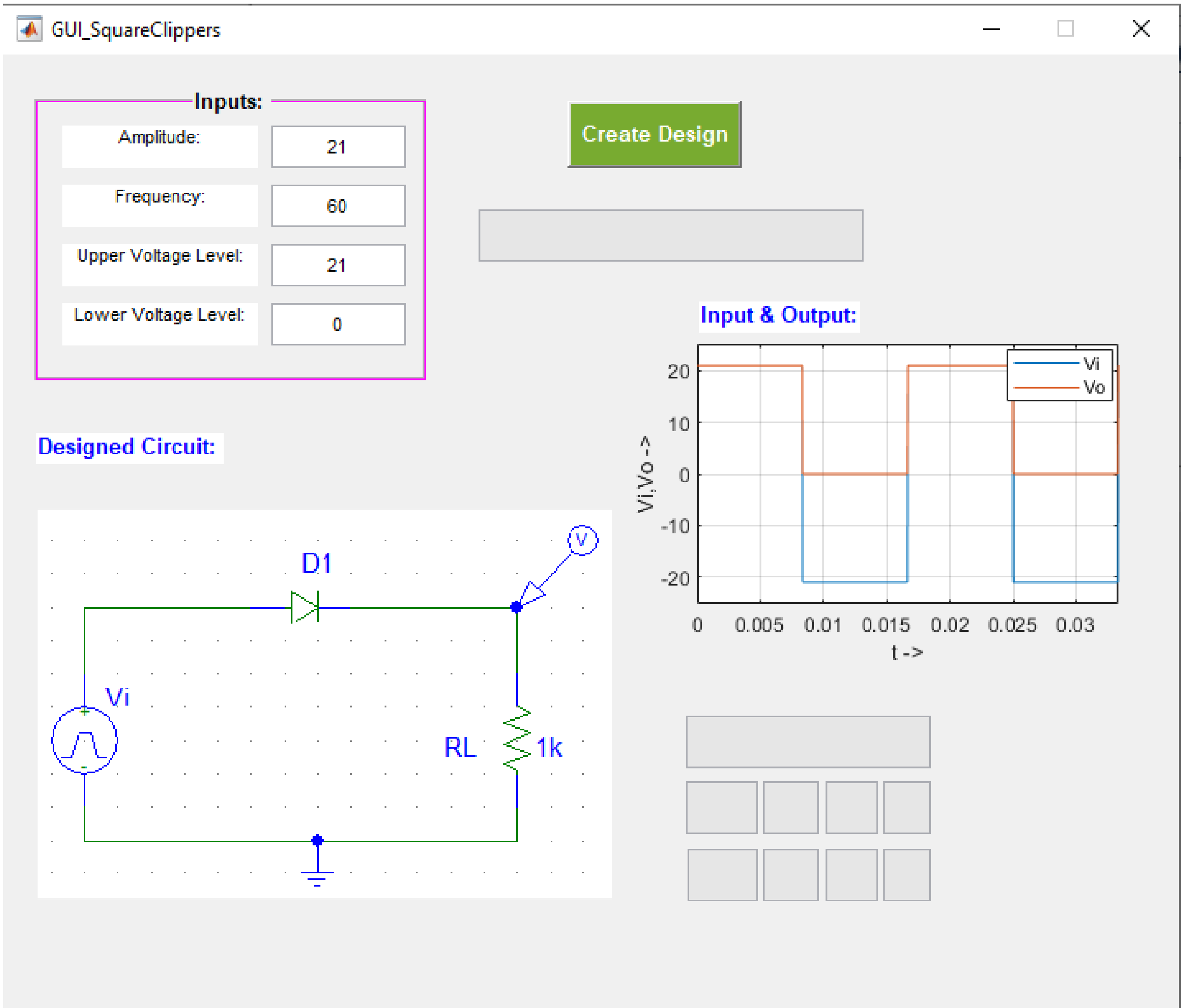
```
    vout=[];
    for i=1:1000
        if vin(i)>=0
            vout(i)=v2;
        else vin(i)<=0
            vout(i)=-v1;
        end
    end
    img=imread('SquareClipper7.PNG');
end
```

Corresponding Window:

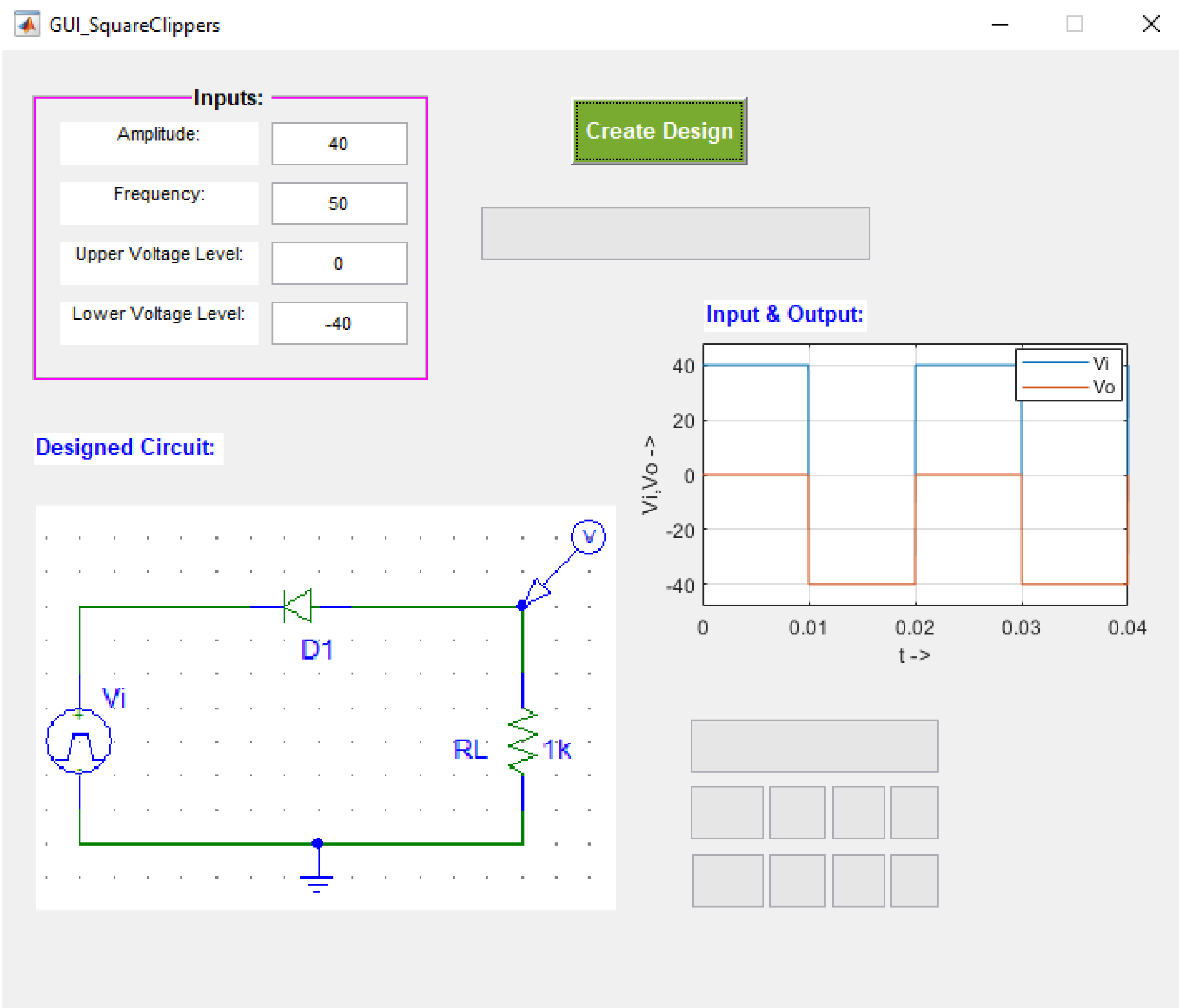


Sample Outputs:

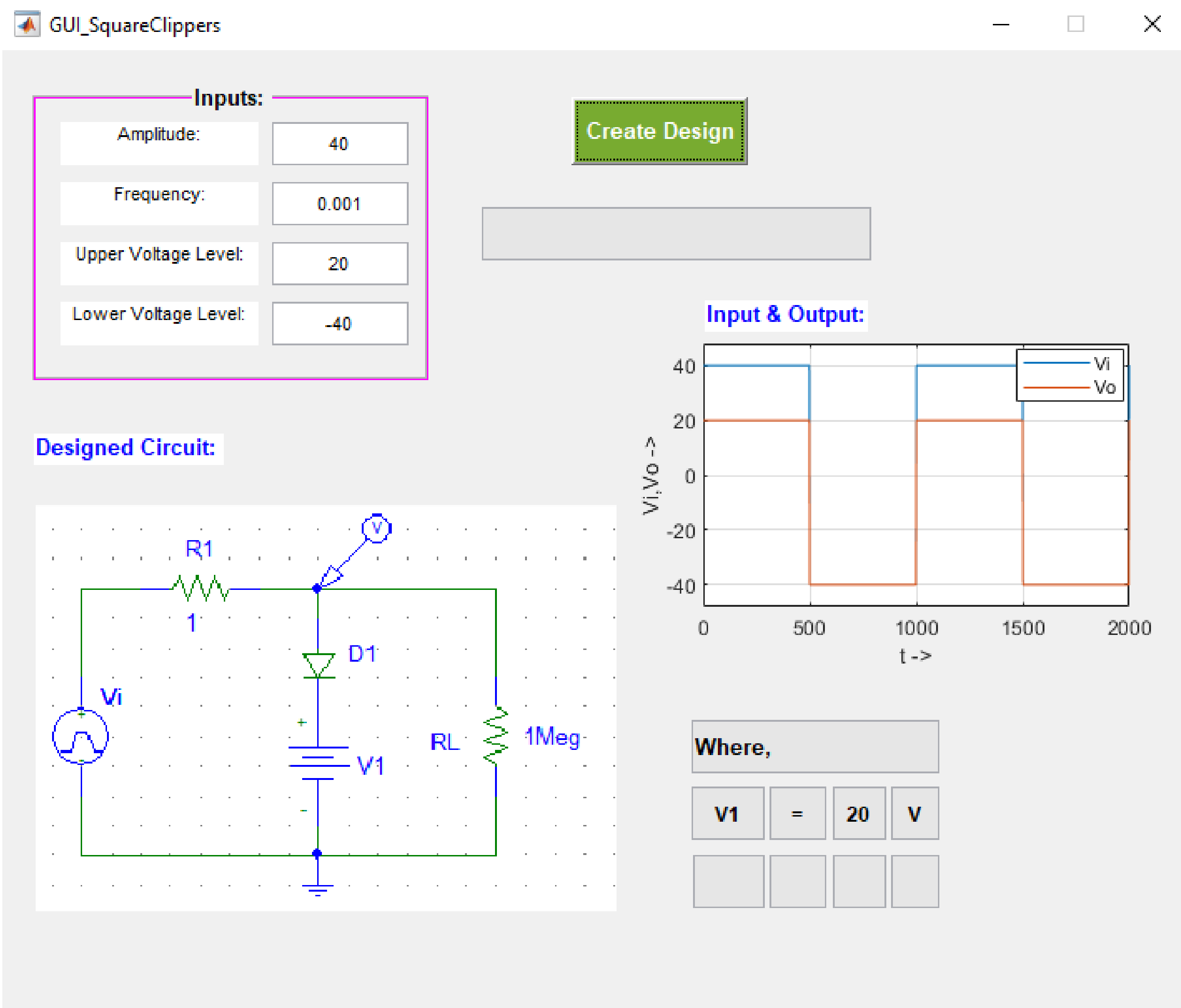
1. Negative Clipper :



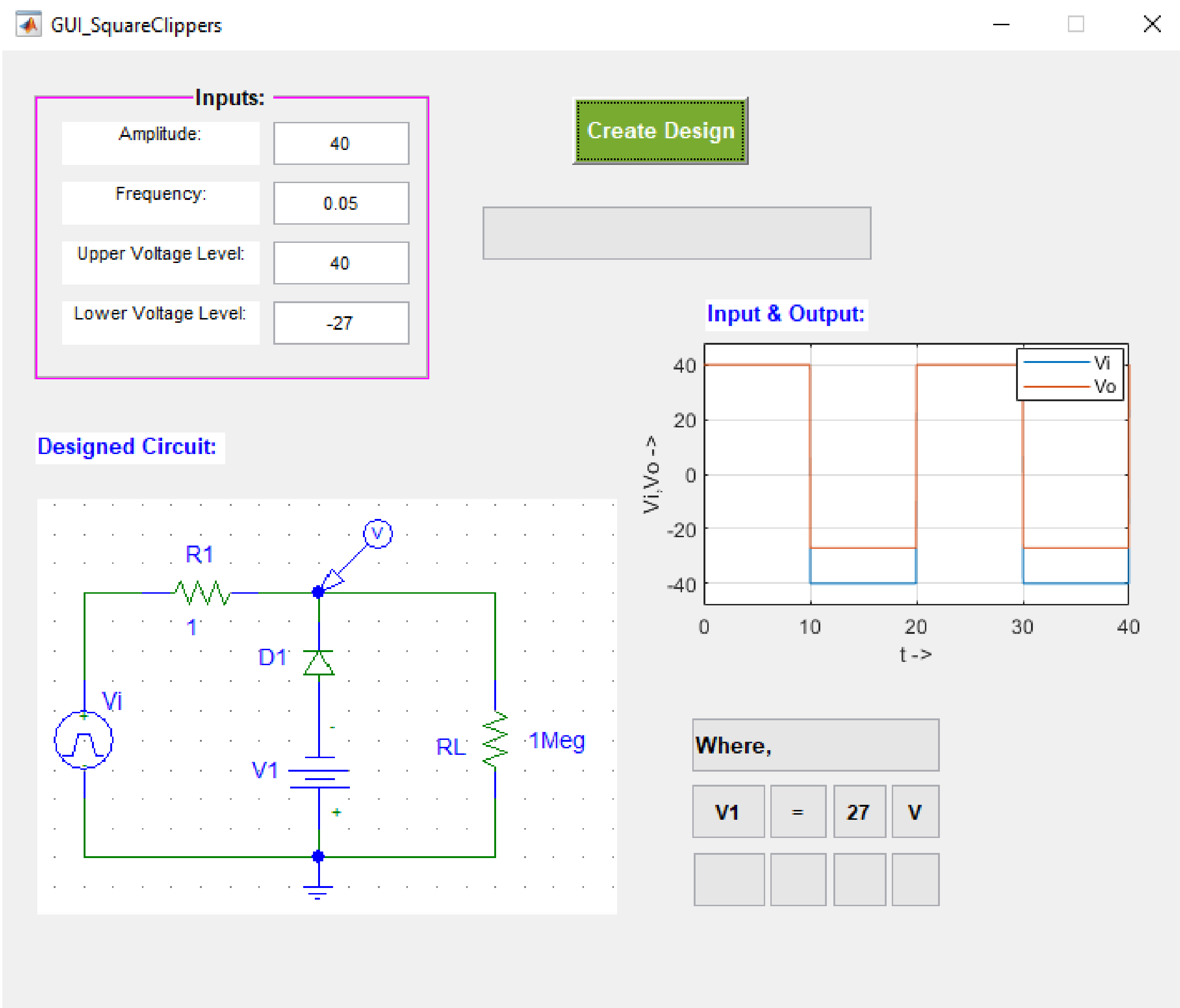
2. Positive Clipper :



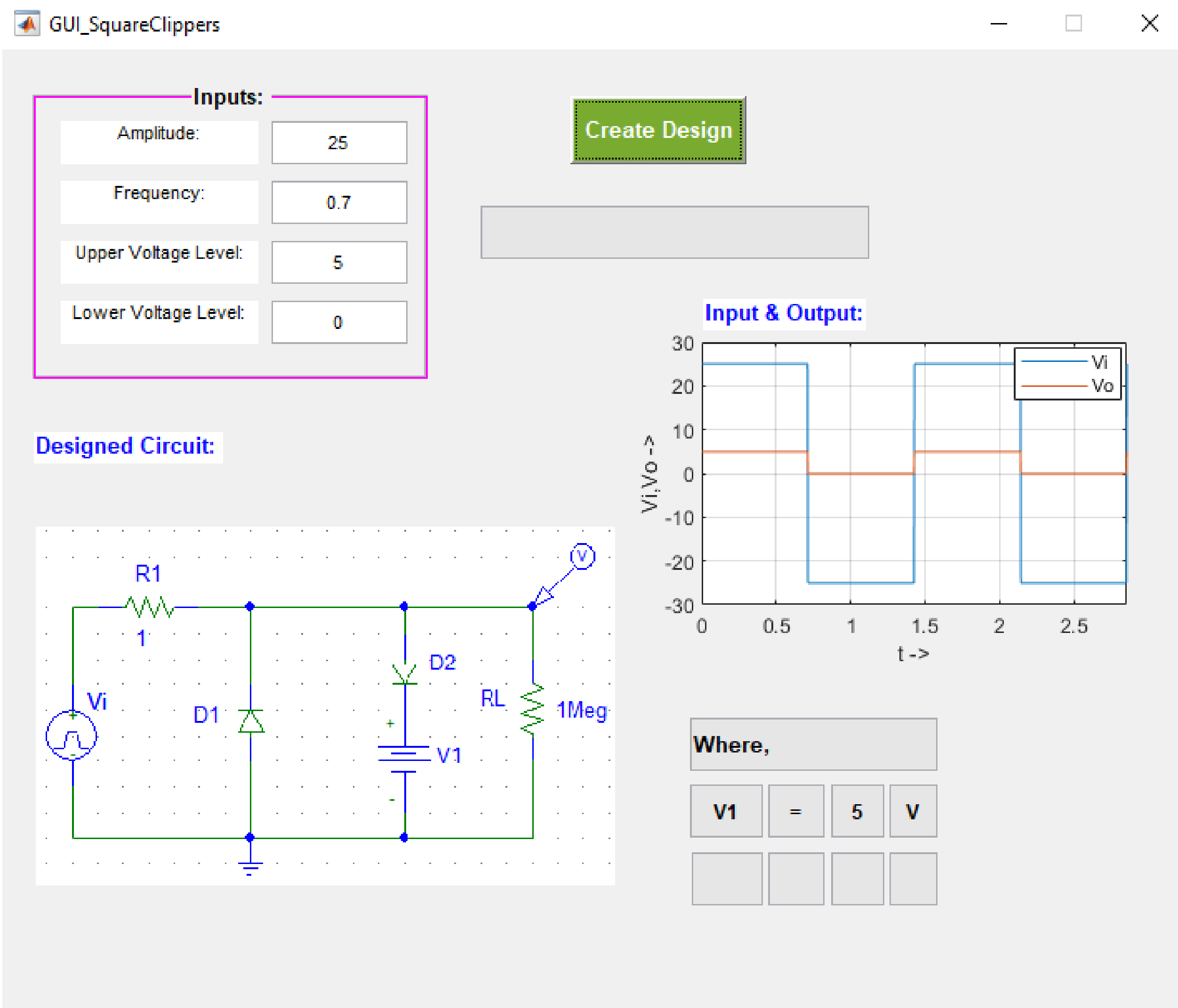
3. Clipping Positive Side to a Non-Zero Value :



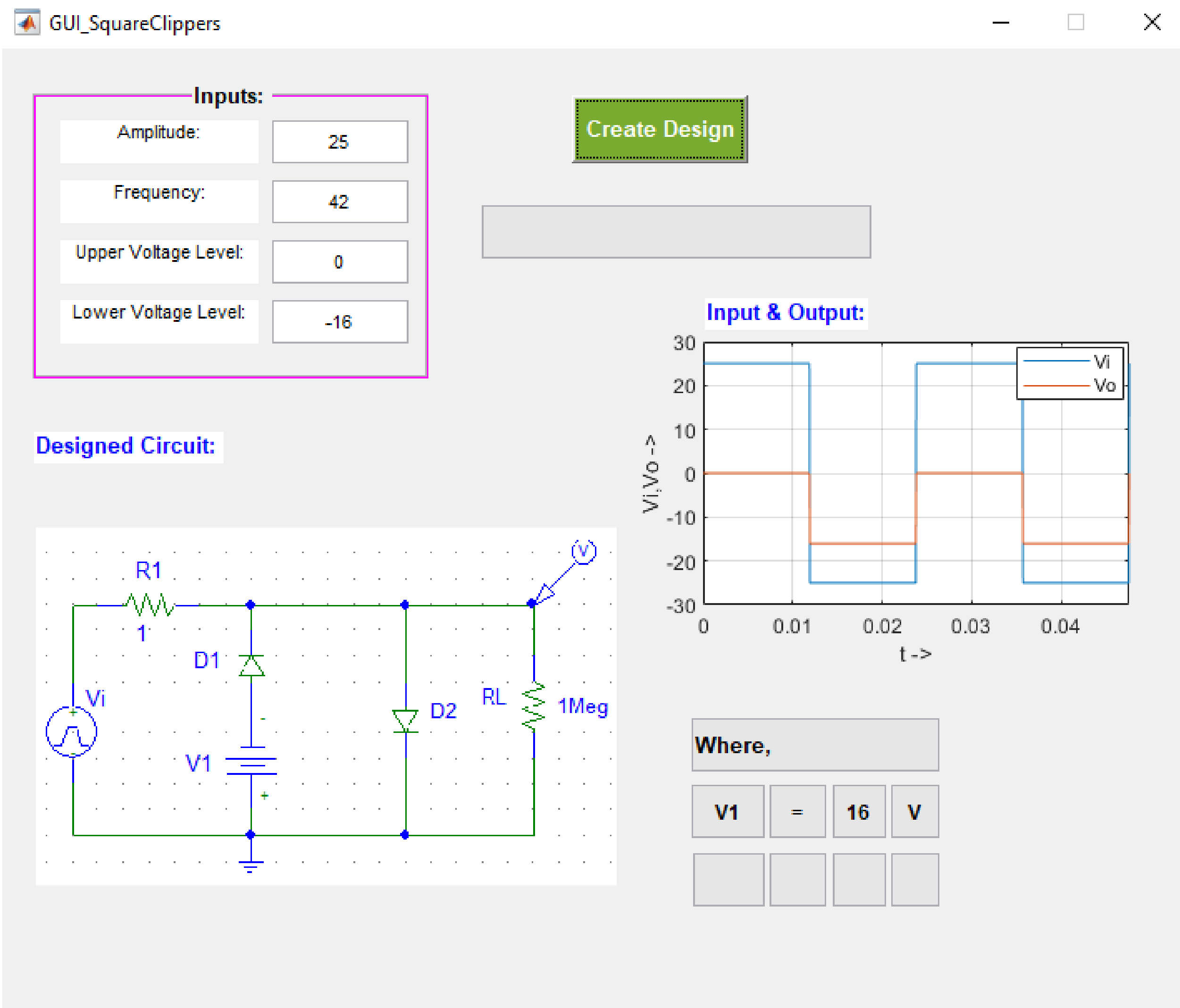
4. Clipping Negative Side to a Non-Zero Value :



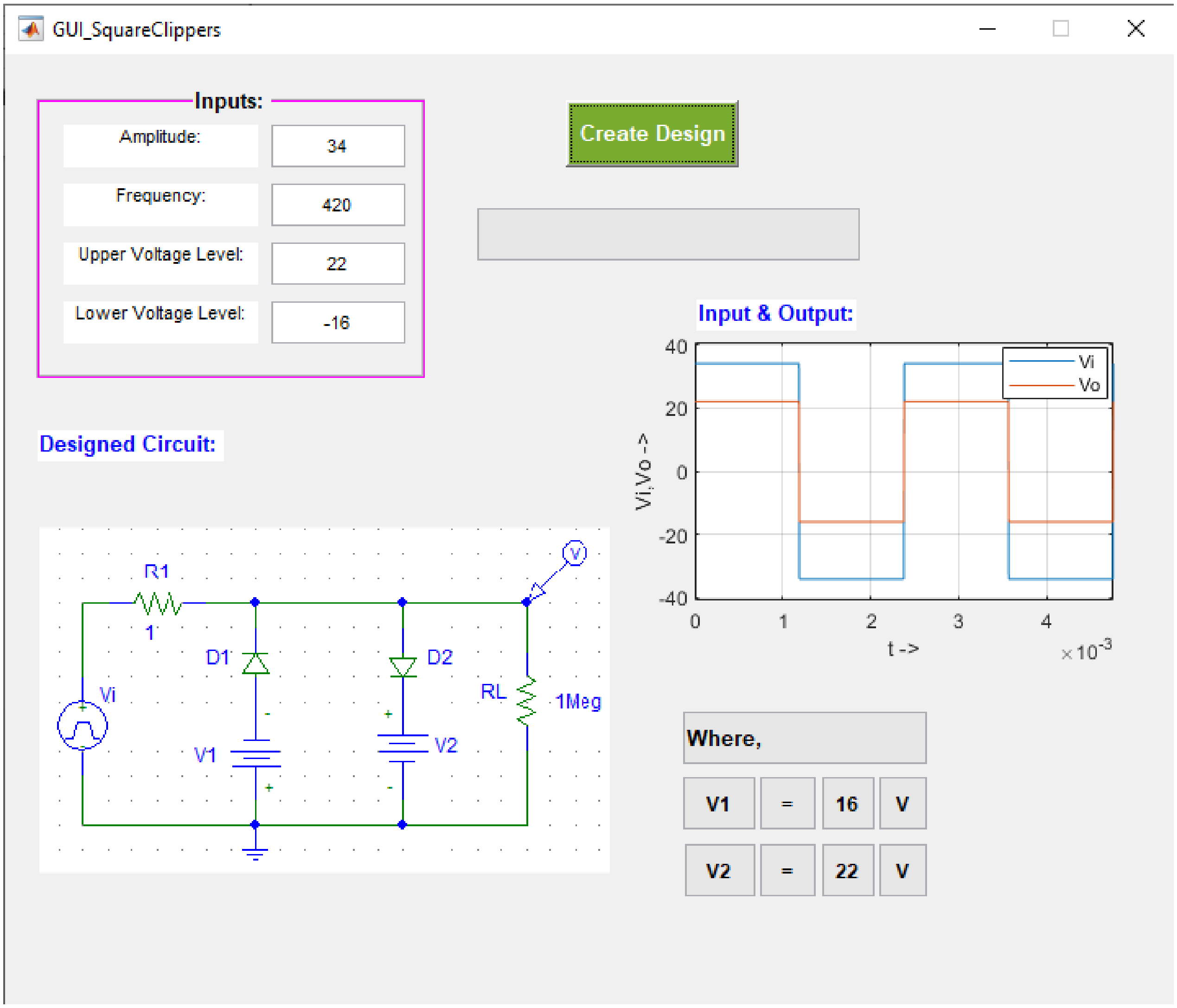
5. **Positive Side is Clipped to a Non-Zero Value and Negative Side is Clipped to Zero :**



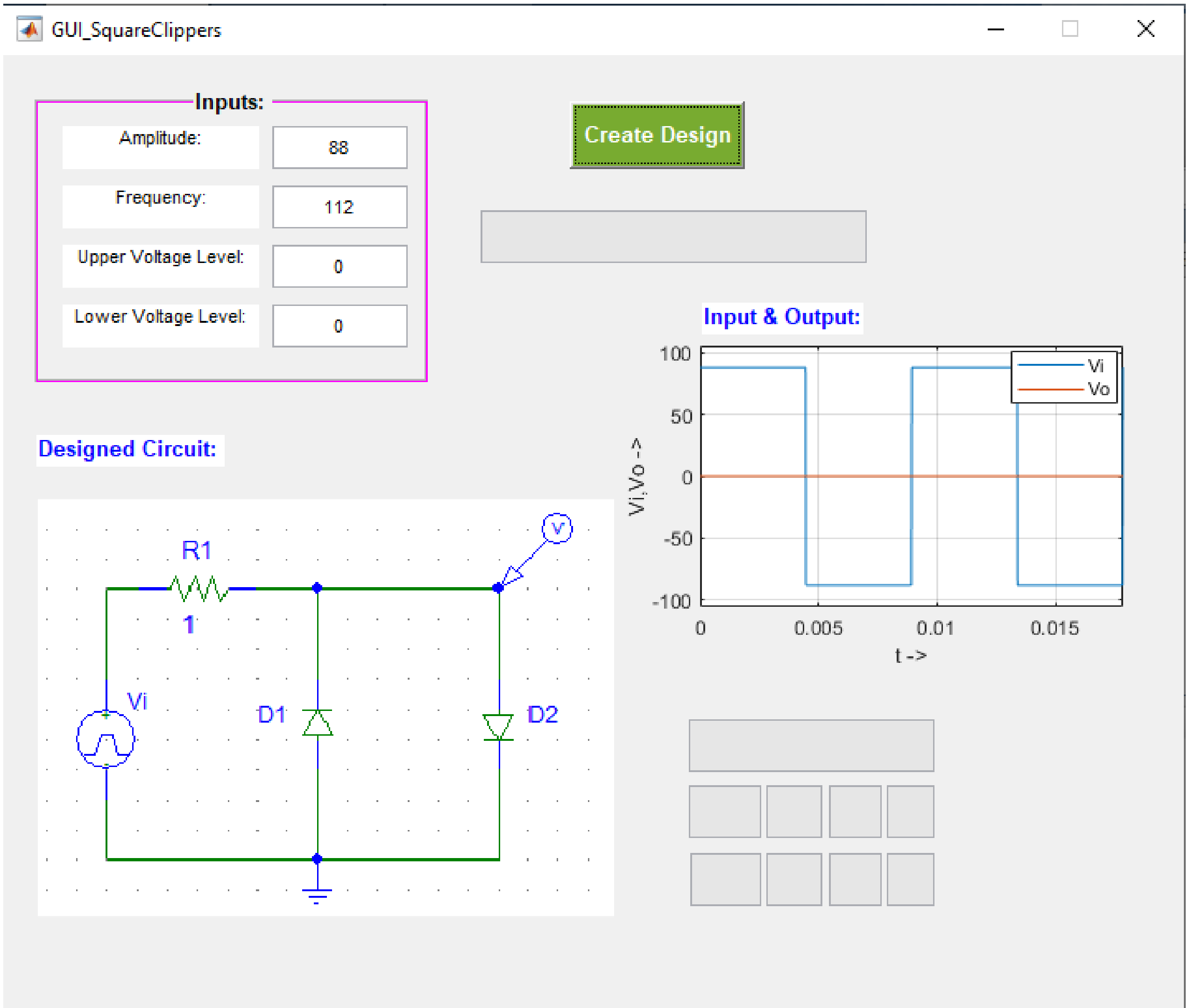
6. **Negative Side is Clipped to a Non-Zero Value and Positive Side is Clipped to Zero :**



7. **Both Positive and Negative Side are Clipped to Non-Zero Value:**

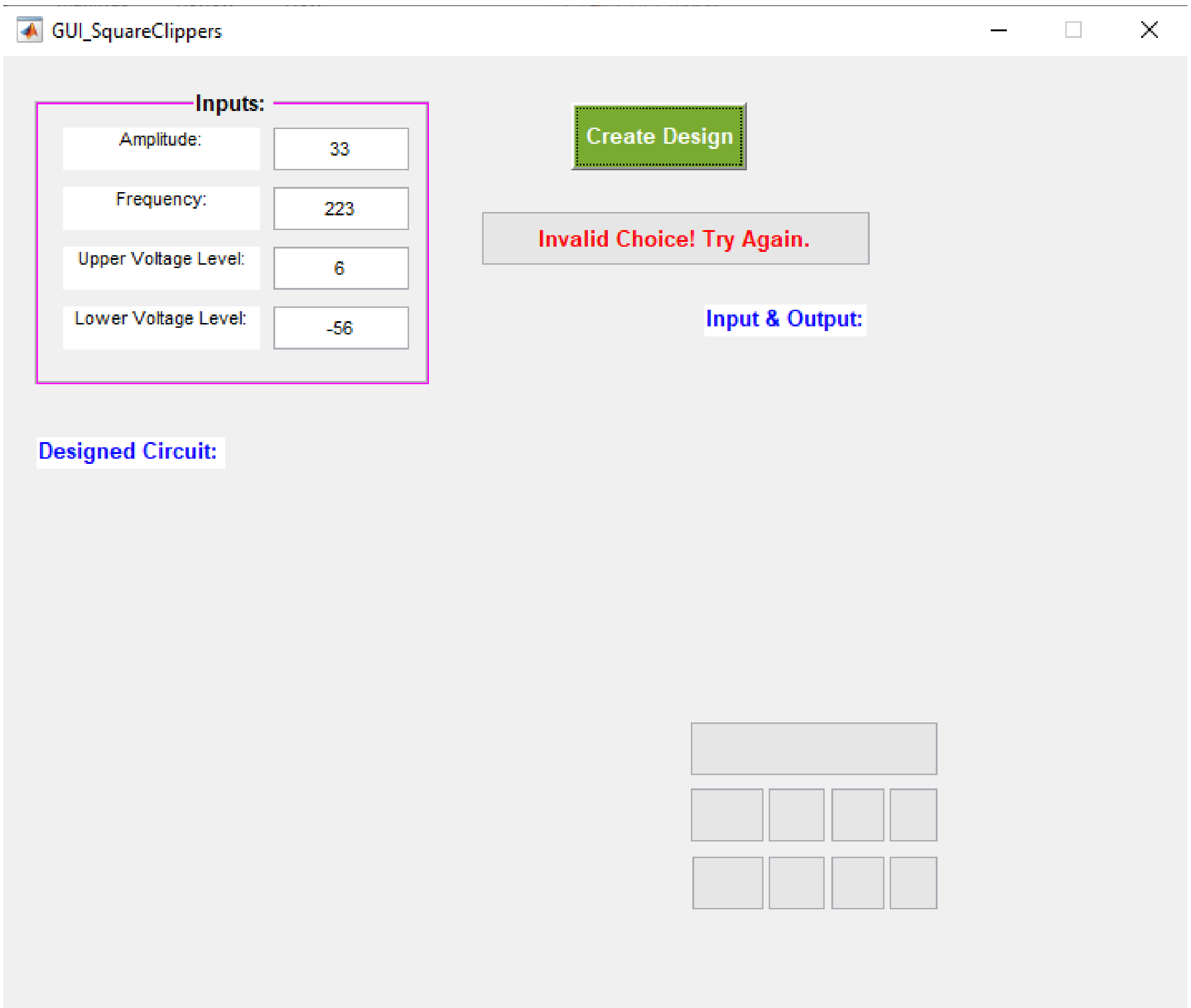


8. Both Positive and Negative Side are Clipped to Zero :

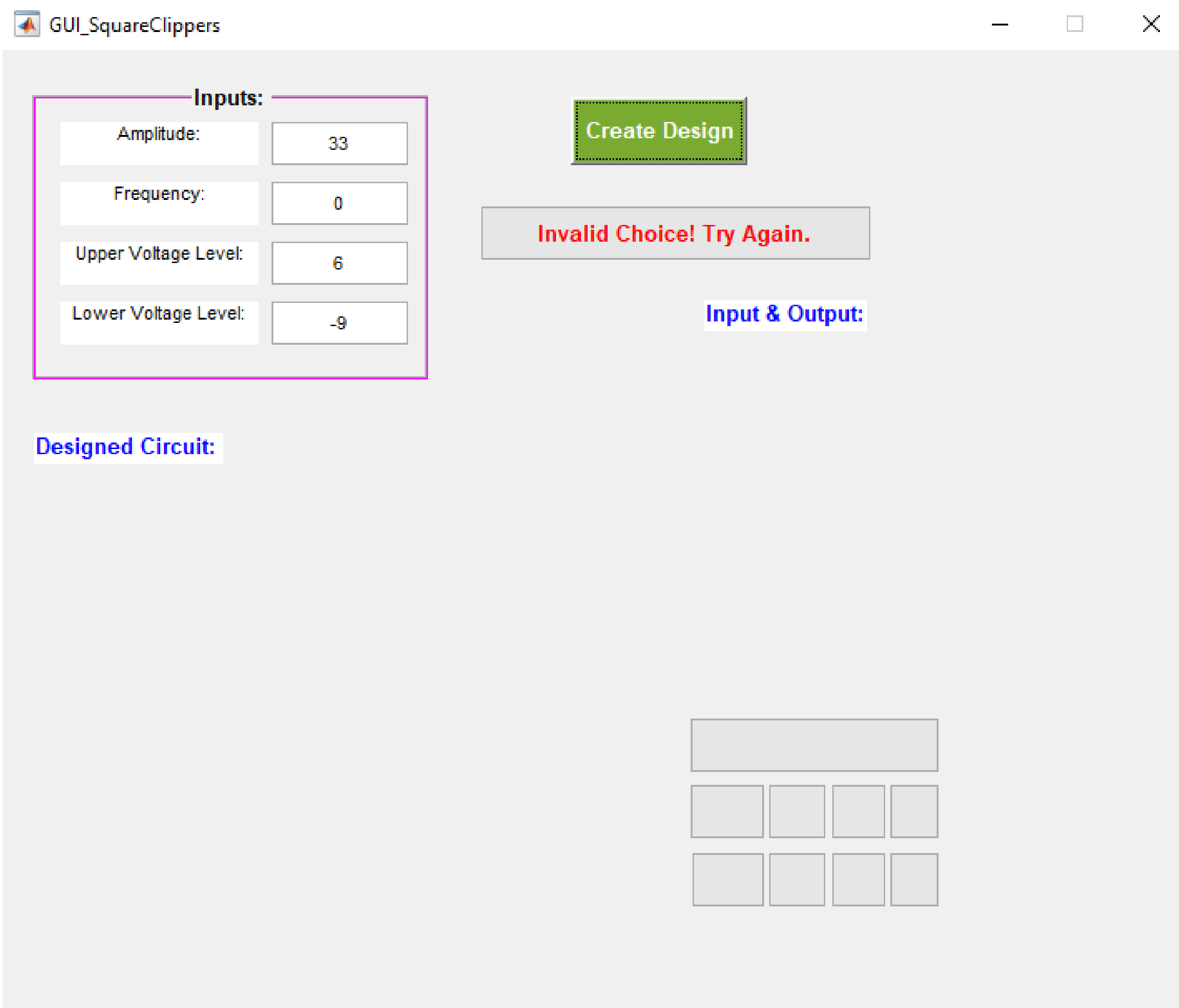


9. Invalid Cases:

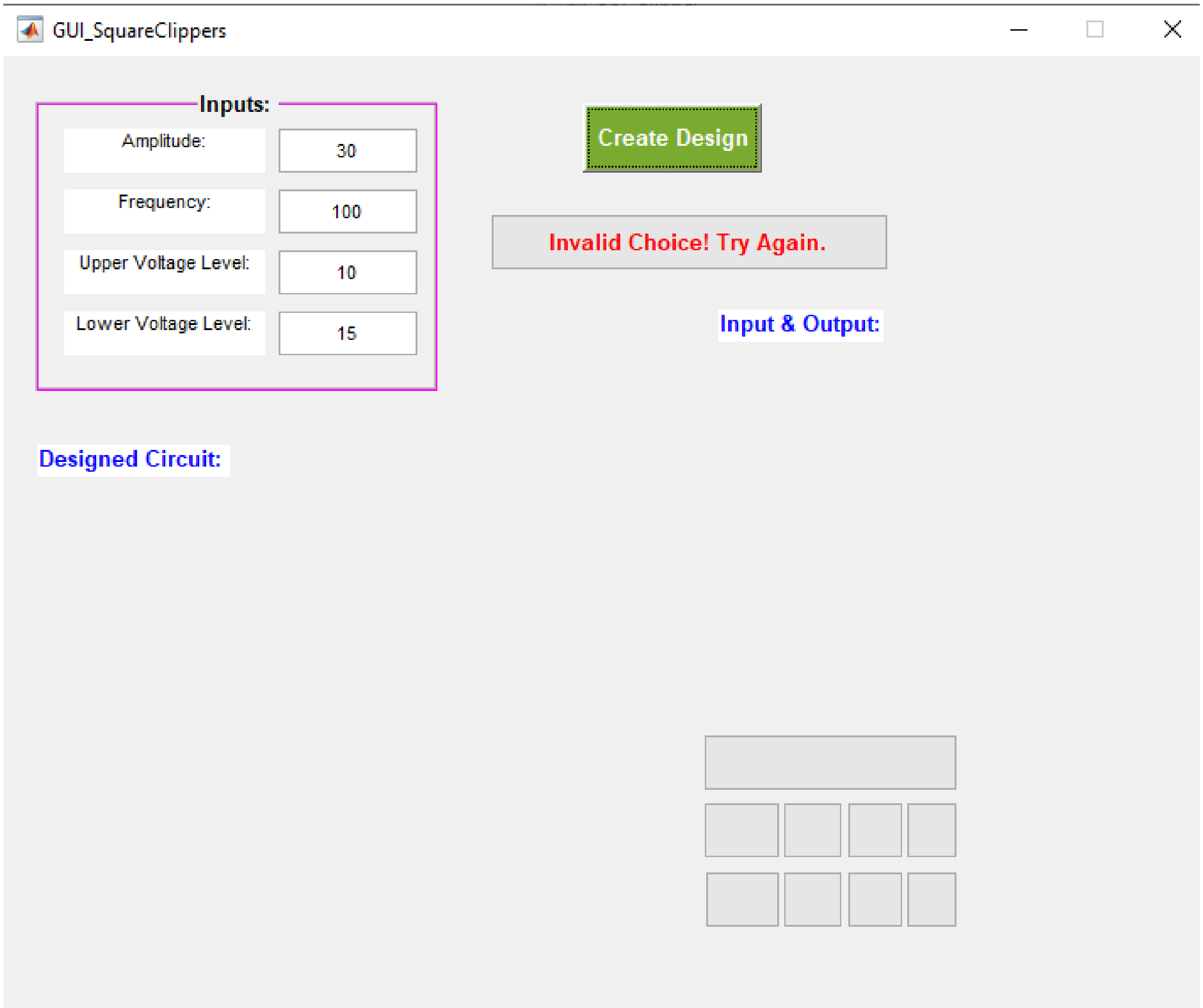
i. Clipped Voltage Level is Greater than Input Voltage Peak :



ii. Zero/ Negative Frequency:



iii. Lower Voltage Level is given greater than Upper Voltage Level:



N. B: Our program is able to design the circuit for any other kind of inputs as well. User is appreciated to check for more as he desires.

(iii) **<u>Triangular Clippers:</u>**

- Now, let us choose "Triangular Wave" from "GUI_Clipper" window.
- "GUI_TriangularClippers.m" interacts with the user and calls "Triangular_Clipper_Design.m" that contains all necessary functions and codes to create the design and sends data back.

<u>Codes:</u>

"GUI_TriangularClippers.m" –

```
function varargout = GUI_TriangularClippers(varargin)
% GUI_TRIANGULARCLIPPERS MATLAB code for GUI_TriangularClippers.fig
% GUI_TRIANGULARCLIPPERS, by itself, creates a new
GUI_TRIANGULARCLIPPERS or raises the existing
% singleton*.
%
% H = GUI_TRIANGULARCLIPPERS returns the handle to a new
GUI_TRIANGULARCLIPPERS or the handle to
% the existing singleton*.
%
% GUI_TRIANGULARCLIPPERS('CALLBACK',hObject,eventData,handles,...) calls the
local
% function named CALLBACK in GUI_TRIANGULARCLIPPERS.M with the given
input arguments.
%
%  GUI_TRIANGULARCLIPPERS('Property','Value',...) creates a new
GUI_TRIANGULARCLIPPERS or raises the
%  existing singleton*.  Starting from the left, property value pairs are
%  applied to the GUI before GUI_TriangularClippers_OpeningFcn gets called.  An
%  unrecognized property name or invalid value makes property application
%  stop.  All inputs are passed to GUI_TriangularClippers_OpeningFcn via varargin.
%
%  *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help GUI_TriangularClippers
% Last Modified by GUIDE v2.5 15-Jul-2021 15:31:15
```

```
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @GUI_TriangularClippers_OpeningFcn, ...
                   'gui_OutputFcn',  @GUI_TriangularClippers_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before GUI_TriangularClippers is made visible.
function GUI_TriangularClippers_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to GUI_TriangularClippers (see VARARGIN)

% Choose default command line output for GUI_TriangularClippers
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes GUI_TriangularClippers wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = GUI_TriangularClippers_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


% --- Executes on button press in design.
function design_Callback(hObject, eventdata, handles)
% hObject    handle to design (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.invalidcheck,'String','');
set(handles.properties,'String','');
set(handles.v11,'String','');
set(handles.v12,'String','');
set(handles.v13,'String','');
set(handles.v14,'String','');
set(handles.v21,'String','');
set(handles.v22,'String','');
set(handles.v23,'String','');
set(handles.v24,'String','');
axes(handles.diagram);
cla;
axes(handles.plots);
cla;

amp=str2double(get(handles.amplitude,'String'));
f=str2double(get(handles.frequency,'String'));
h=str2double(get(handles.upper,'String'));
l=str2double(get(handles.lower,'String'));

if h>amp || -l>amp || f<=0 || l>h
    set(handles.invalidcheck,'String','Invalid Choice! Try Again.');
else
    [t,vin,vout,img,v1,v2]=Triangular_Clipper_Design(amp,f,h,l);
    axes(handles.plots);
    cla;
    plot(t,vin);
    hold on;
    plot(t,vout);
```

```
    axis([0,t(end),(-amp-amp/5),(amp+amp/5)]);
    grid on;
    legend('Vi','Vo');
    xlabel('t ->');
    ylabel('Vi,Vo ->');
    axes(handles.diagram);
    cla;
    imshow(img);
    if v1~=0 && v2==0
        set(handles.properties,'String','Where,');
        set(handles.v11,'String','V1');
        set(handles.v12,'String','=');
        set(handles.v13,'String',num2str(v1));
        set(handles.v14,'String','V');
    elseif v1~=0 && v2~=0
        set(handles.properties,'String','Where,');
        set(handles.v11,'String','V1');
        set(handles.v12,'String','=');
        set(handles.v13,'String',num2str(v1));
        set(handles.v14,'String','V');
        set(handles.v21,'String','V2');
        set(handles.v22,'String','=');
        set(handles.v23,'String',num2str(v2));
        set(handles.v24,'String','V');
    end
end

function properties_Callback(hObject, eventdata, handles)
% hObject    handle to properties (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of properties as text
%        str2double(get(hObject,'String')) returns contents of properties as a double

% --- Executes during object creation, after setting all properties.
function properties_CreateFcn(hObject, eventdata, handles)
% hObject    handle to properties (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function v11_Callback(hObject, eventdata, handles)
% hObject    handle to v11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v11 as text
%        str2double(get(hObject,'String')) returns contents of v11 as a double


% --- Executes during object creation, after setting all properties.
function v11_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function v12_Callback(hObject, eventdata, handles)
% hObject    handle to v12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v12 as text
%        str2double(get(hObject,'String')) returns contents of v12 as a double


% --- Executes during object creation, after setting all properties.
function v12_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v12 (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function v13_Callback(hObject, eventdata, handles)
% hObject    handle to v13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v13 as text
%        str2double(get(hObject,'String')) returns contents of v13 as a double

% --- Executes during object creation, after setting all properties.

function v13_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function v21_Callback(hObject, eventdata, handles)
% hObject    handle to v21 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v21 as text
%        str2double(get(hObject,'String')) returns contents of v21 as a double
```

```
% --- Executes during object creation, after setting all properties.
function v21_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v21 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function v22_Callback(hObject, eventdata, handles)
% hObject    handle to v22 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v22 as text
%        str2double(get(hObject,'String')) returns contents of v22 as a double


% --- Executes during object creation, after setting all properties.
function v22_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v22 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function v23_Callback(hObject, eventdata, handles)
% hObject    handle to v23 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of v23 as text
%        str2double(get(hObject,'String')) returns contents of v23 as a double

% --- Executes during object creation, after setting all properties.
function v23_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v23 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function invalidcheck_Callback(hObject, eventdata, handles)
% hObject    handle to invalidcheck (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of invalidcheck as text
%        str2double(get(hObject,'String')) returns contents of invalidcheck as a double


% --- Executes during object creation, after setting all properties.
function invalidcheck_CreateFcn(hObject, eventdata, handles)
% hObject    handle to invalidcheck (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function v14_Callback(hObject, eventdata, handles)
% hObject    handle to v14 (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v14 as text
%        str2double(get(hObject,'String')) returns contents of v14 as a double


% --- Executes during object creation, after setting all properties.
function v14_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function v24_Callback(hObject, eventdata, handles)
% hObject    handle to v24 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v24 as text
%        str2double(get(hObject,'String')) returns contents of v24 as a double

% --- Executes during object creation, after setting all properties.
function v24_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v24 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function amplitude_Callback(hObject, eventdata, handles)
% hObject    handle to amplitude (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of amplitude as text
% str2double(get(hObject,'String')) returns contents of amplitude as a double


% --- Executes during object creation, after setting all properties.
function amplitude_CreateFcn(hObject, eventdata, handles)
% hObject    handle to amplitude (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function frequency_Callback(hObject, eventdata, handles)
% hObject    handle to frequency (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of frequency as text
% str2double(get(hObject,'String')) returns contents of frequency as a double


% --- Executes during object creation, after setting all properties.
function frequency_CreateFcn(hObject, eventdata, handles)
% hObject    handle to frequency (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
```

```
    set(hObject,'BackgroundColor','white');
end


function lower_Callback(hObject, eventdata, handles)
% hObject    handle to lower (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of lower as text
% str2double(get(hObject,'String')) returns contents of lower as a double


% --- Executes during object creation, after setting all properties.
function lower_CreateFcn(hObject, eventdata, handles)
% hObject    handle to lower (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function upper_Callback(hObject, eventdata, handles)
% hObject    handle to lower (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of lower as text
% str2double(get(hObject,'String')) returns contents of lower as a double


% --- Executes during object creation, after setting all properties.
function upper_CreateFcn(hObject, eventdata, handles)
% hObject    handle to lower (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

“Triangular_Clipper_Design.m” –

```
function [t,vin,vout,img,v1,v2]=Triangular_Clipper_Design(vamp,f,vup,vlow)
    if vlow==0 && vup==vamp
        [t,vin,vout,img,v1,v2]=negative_clipper(vamp,f,vup,vlow);
    elseif vup==0 && vlow==-(vamp)
        [t,vin,vout,img,v1,v2]=positive_clipper(vamp,f,vup,vlow);
    elseif vlow==0 && vup==0
        [t,vin,vout,img,v1,v2]=both_negative_positive_clipper(vamp,f,vup,vlow);
    elseif 0<vup<vamp && vlow==-(vamp)
        [t,vin,vout,img,v1,v2]=partial_positive_clipper(vamp,f,vup,vlow);
    elseif 0<-(vlow)<vamp && vup==vamp
        [t,vin,vout,img,v1,v2]=partial_negative_clipper(vamp,f,vup,vlow);
    elseif 0<vup<vamp && vlow==0
        [t,vin,vout,img,v1,v2]=partial_positive_full_negative_clipper(vamp,f,vup,vlow);
    elseif vup==0 && 0<-(vlow)<vamp
        [t,vin,vout,img,v1,v2]=partial_negative_full_positive_clipper(vamp,f,vup,vlow);
    elseif 0<vup<vamp && 0<-(vlow)<vamp
        [t,vin,vout,img,v1,v2]=partial_negative_positive_clipper(vamp,f,vup,vlow);
    end
end

function [t,vin,vout,img,v1,v2]=negative_clipper(vamp,f,vup,vlow)
    v1=0;
    v2=0;
    T=1/f;
    t=linspace(0,2*T,1000);
    vin=vamp.*sawtooth((2*pi*f)*t,0.5);
    vout=[];
    for i=1:1000
        if vin(i)>=0
            vout(i)=vin(i);
```

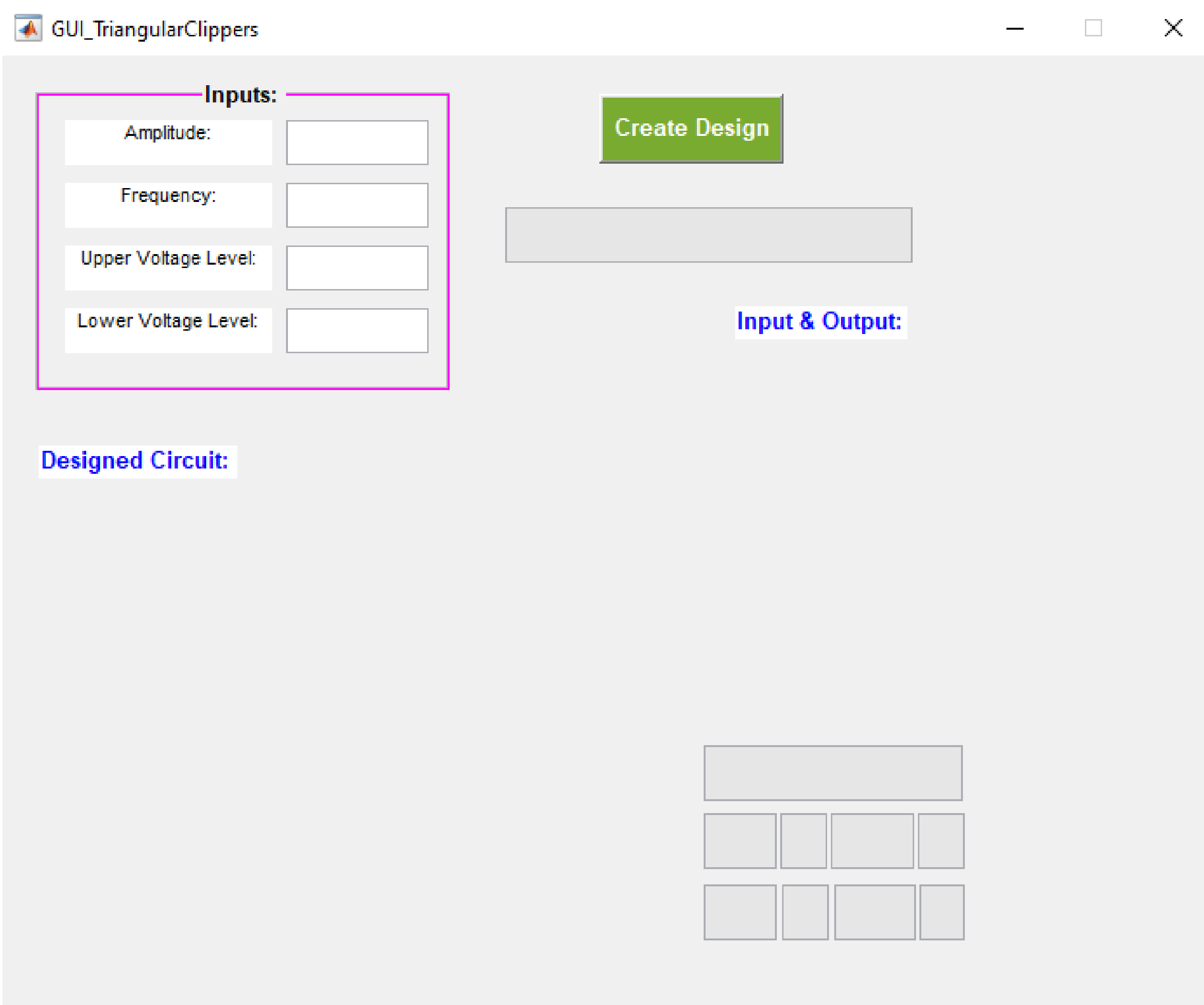
```
        else
            vout(i)=0;
        end
    end
    img=imread('TriangularClipper2.PNG');
end

function [t,vin,vout,img,v1,v2]=positive_clipper(vamp,f,vup,vlow)
    v1=0;
    v2=0;
    T=1/f;
    t=linspace(0,2*T,1000);
    vin=vamp.*sawtooth((2*pi*f)*t,0.5);
    vout=[];
    for i=1:1000
        if vin(i)<=0
            vout(i)=vin(i);
        else
            vout(i)=0;
        end
    end
    img=imread('TriangularClipper1.PNG');
end

function [t,vin,vout,img,v1,v2]=partial_positive_clipper(vamp,f,vup,vlow)
    v1=vup;
    v2=0;
    T=1/f;
    t=linspace(0,2*T,1000);
    vin=vamp.*sawtooth((2*pi*f)*t,0.5);
    vout=[];
    for i=1:1000
        if vin(i)>=vup
            vout(i)=v1;
        else
            vout(i)=vin(i);
        end
    end
    img=imread('TriangularClipper3.PNG');
```

```
end

function [t,vin,vout,img,v1,v2]=partial_negative_clipper(vamp,f,vup,vlow)
   v1=-vlow;
   v2=0;
   T=1/f;
   t=linspace(0,2*T,1000);
   vin=vamp.*sawtooth((2*pi*f)*t,0.5);
   vout=[];
   for i=1:1000
      if vin(i)<=vlow
         vout(i)=-v1;
      else
         vout(i)=vin(i);
      end
   end
   img=imread('TriangularClipper4.PNG');
end

function [t,vin,vout,img,v1,v2]=partial_positive_full_negative_clipper(vamp,f,vup,vlow)
   v1=vup;
   v2=0;
   T=1/f;
   t=linspace(0,2*T,1000);
   vin=vamp.*sawtooth((2*pi*f)*t,0.5);
   vout=[];
   for i=1:1000
      if vin(i)>=vup
         vout(i)=v1;
      elseif vin(i)<0
         vout(i)=0;
      else
         vout(i)=vin(i);
      end
   end
   img=imread('TriangularClipper5.PNG');
end

function [t,vin,vout,img,v1,v2]=partial_negative_full_positive_clipper(vamp,f,vup,vlow)
   v1=-(vlow);
```

```
    v2=0;
    T=1/f;
    t=linspace(0,2*T,1000);
    vin=vamp.*sawtooth((2*pi*f)*t,0.5);
    vout=[];
    for i=1:1000
        if vin(i)>=0
            vout(i)=0;
        elseif vin(i)<=vlow
            vout(i)=-v1;
        else
            vout(i)=vin(i);
        end
    end
    img=imread('TriangularClipper6.PNG');
end

function [t,vin,vout,img,v1,v2]=partial_negative_positive_clipper(vamp,f,vup,vlow)
    v1=-(vlow);
    v2=vup;
    T=1/f;
    t=linspace(0,2*T,1000);
    vin=vamp.*sawtooth((2*pi*f)*t,0.5);
    vout=[];
    for i=1:1000
        if vin(i)>=vup
            vout(i)=v2;
        elseif vin(i)<=vlow
            vout(i)=-v1;
        else
            vout(i)=vin(i);
        end
    end
    img=imread('TriangularClipper7.PNG');
end

function [t,vin,vout,img,v1,v2]=both_negative_positive_clipper(vamp,f,vup,vlow)
    v1=0;
    v2=0;
```

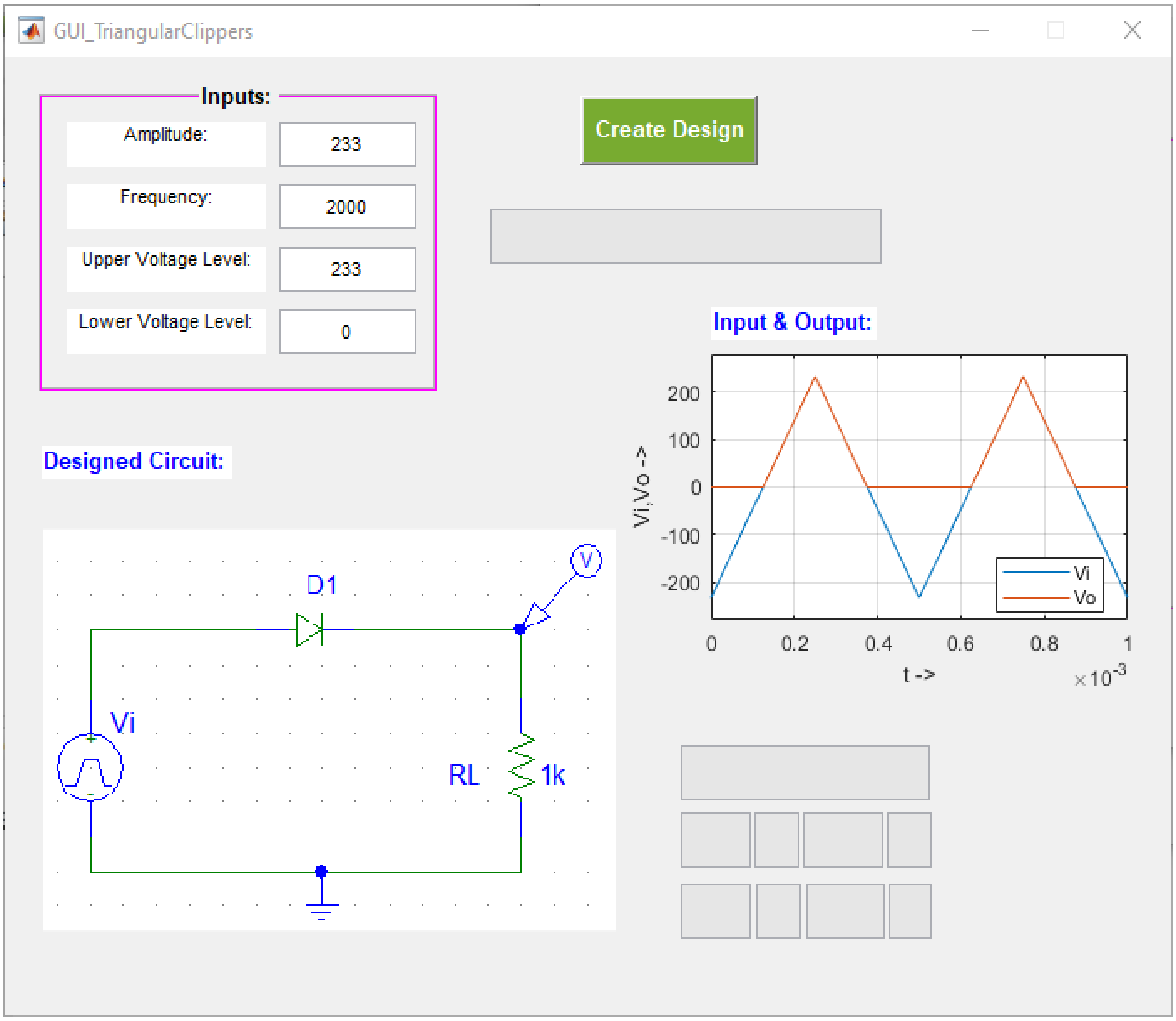
```
    T=1/f;
    t=linspace(0,2*T,1000);
    vin=vamp.*sawtooth((2*pi*f)*t,0.5);
    vout=[];
    for i=1:1000
        vout(i)=0;
    end
    img=imread('TriangularClipper8.PNG');
end
```

Corresponding Window:

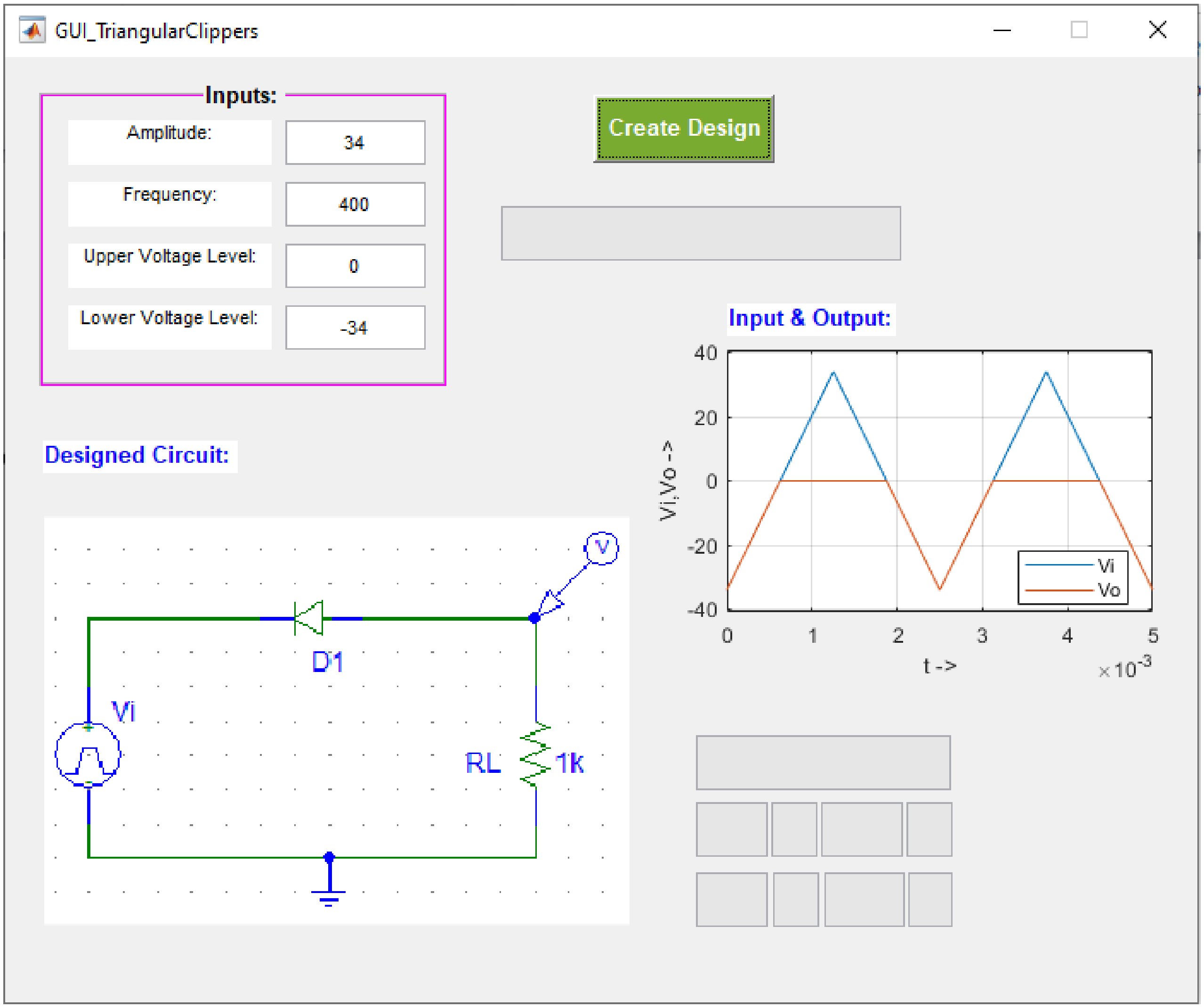


Sample Outputs:

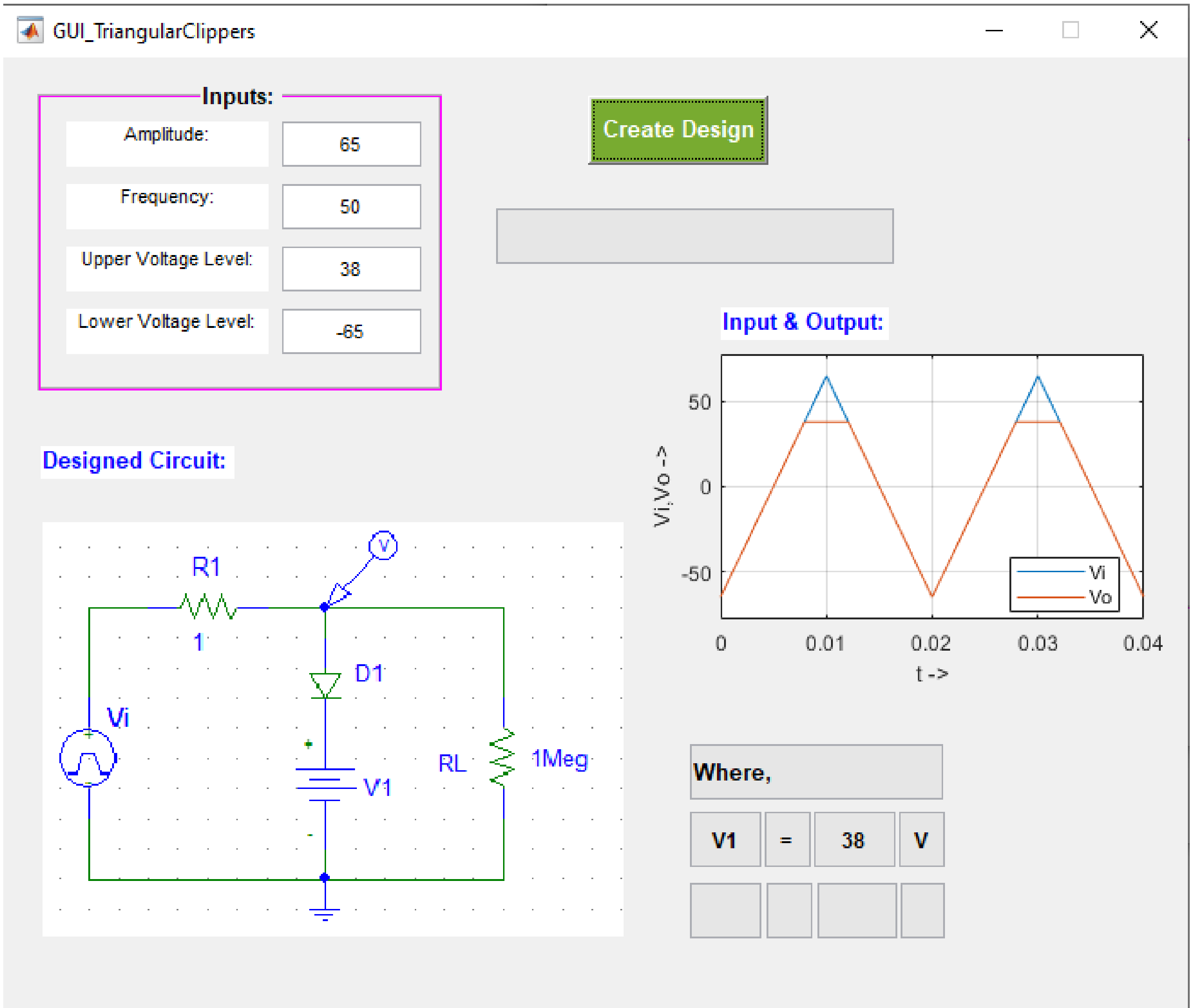
1. Negative Clipper :



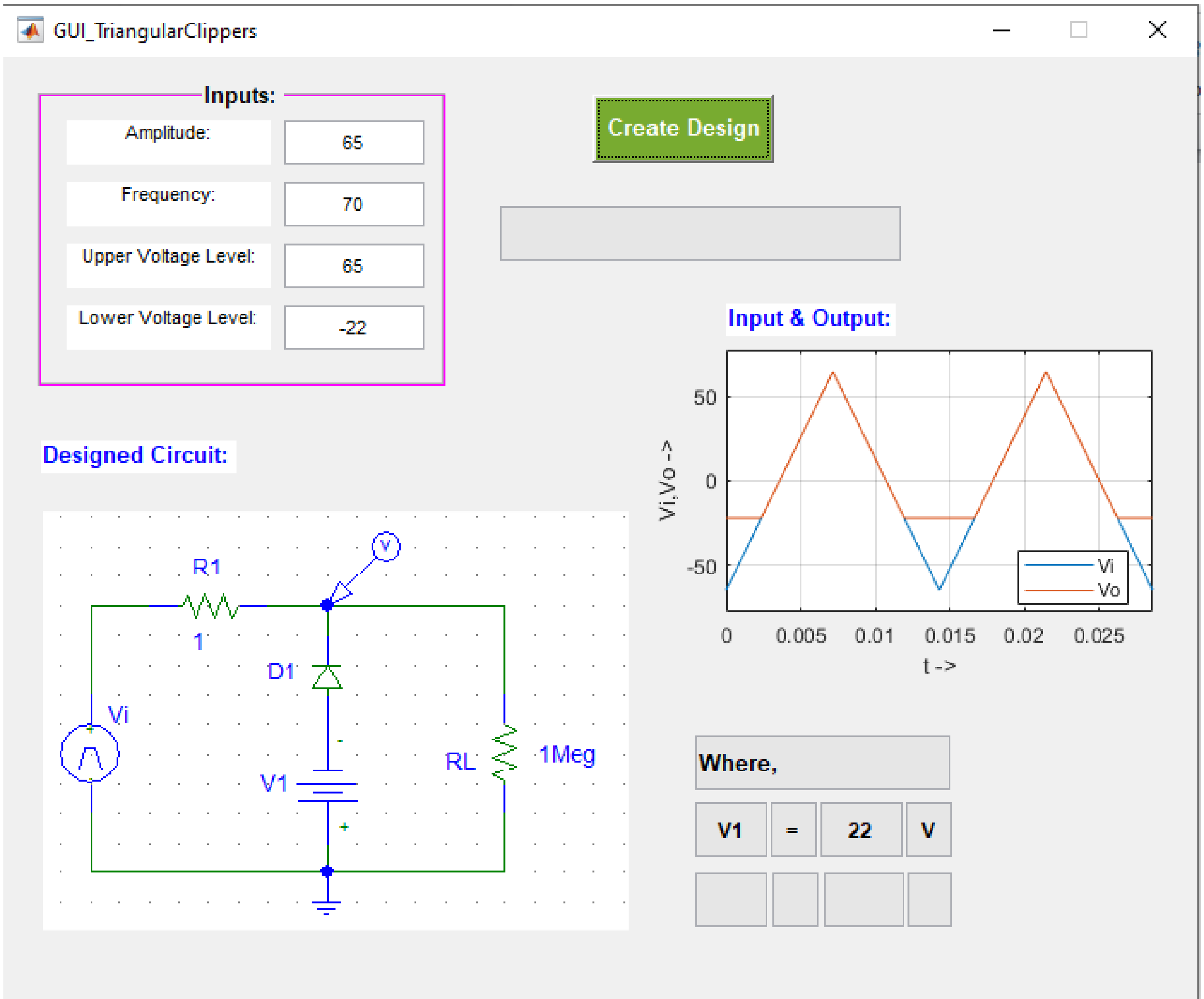
2. Positive Clipper :



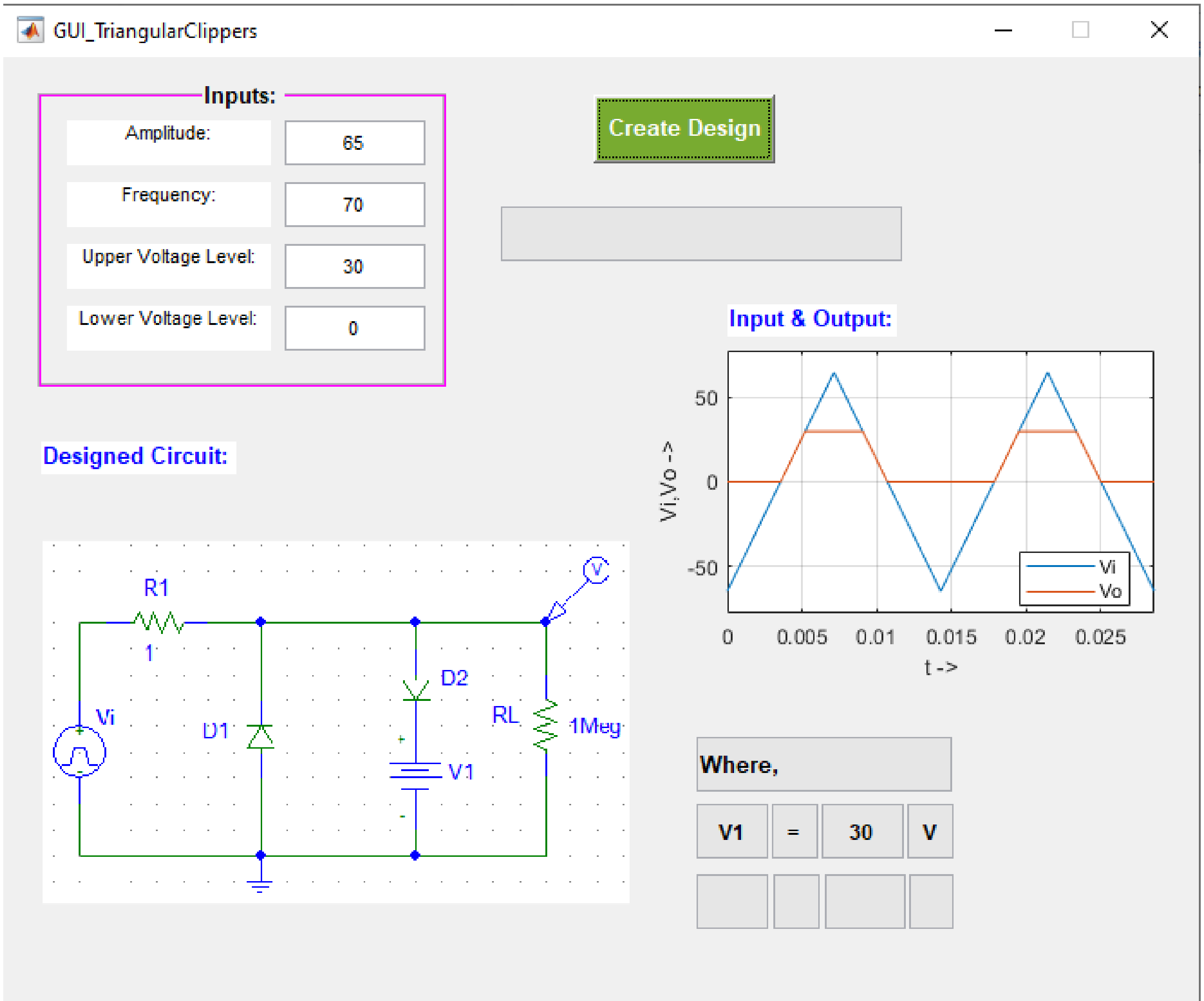
3. Clipping Positive Side Partially:



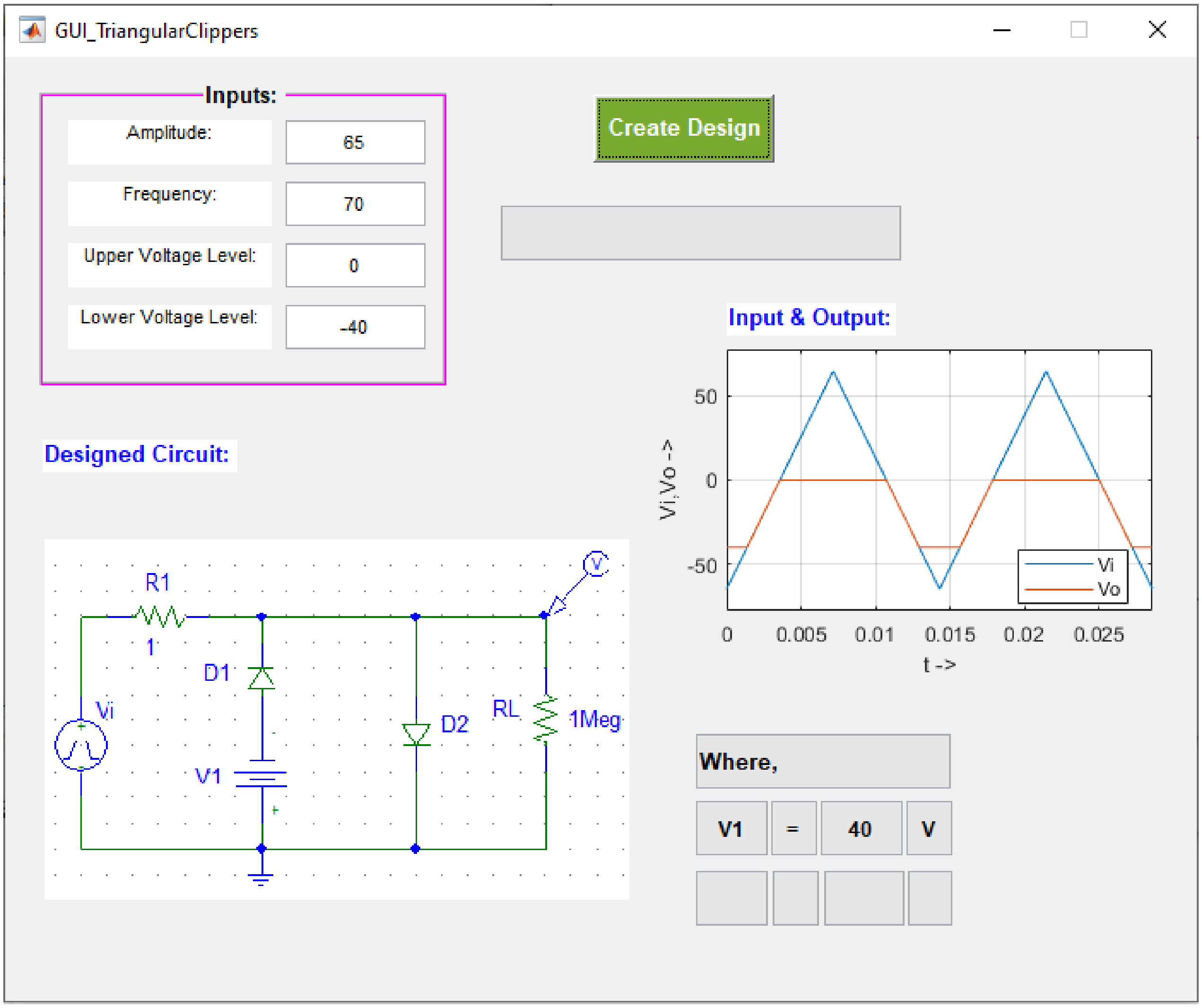
4. Clipping Negative Side Partially:



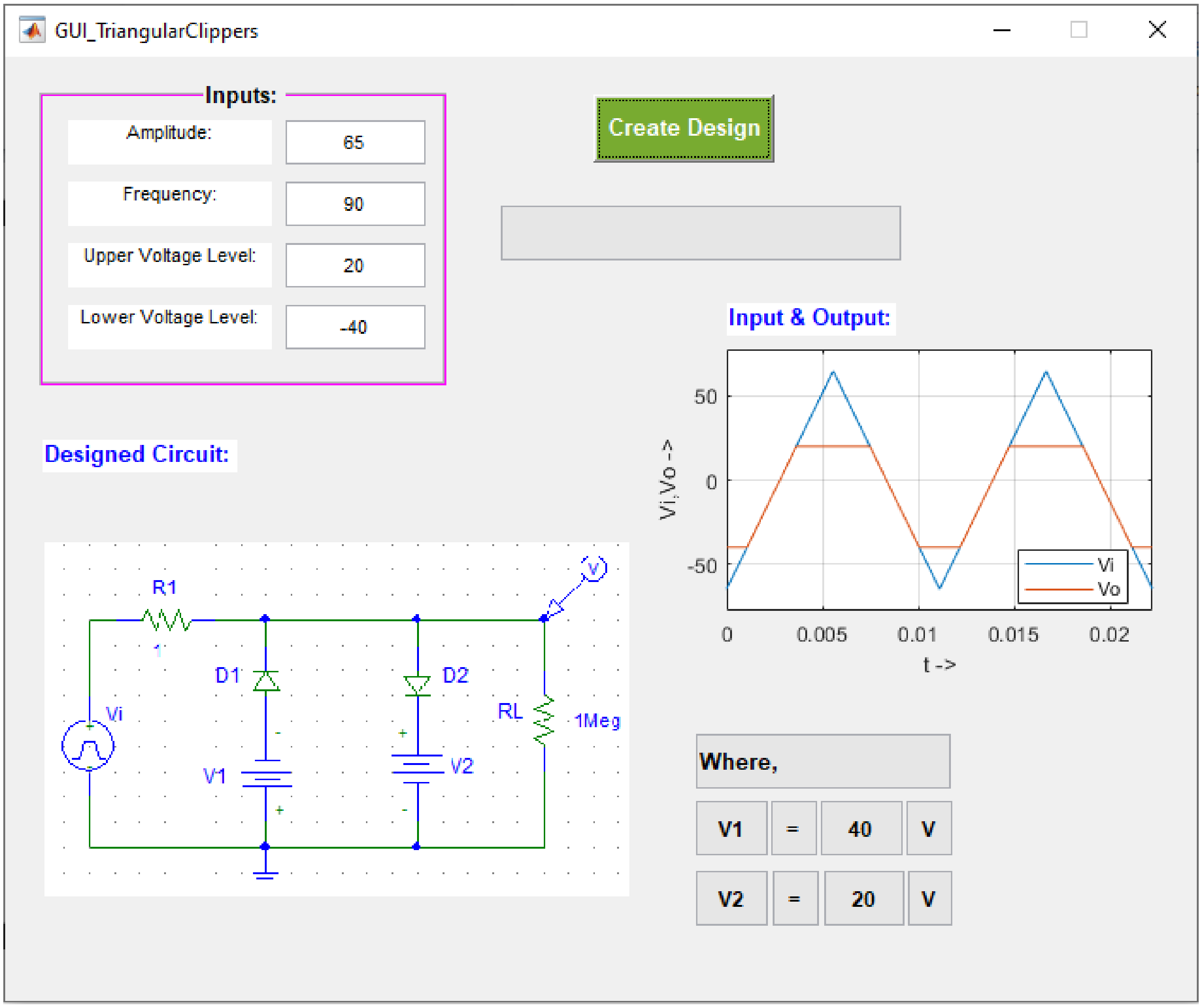
5. Positive Side is Partially Clipped and Negative Side is Fully Clipped :



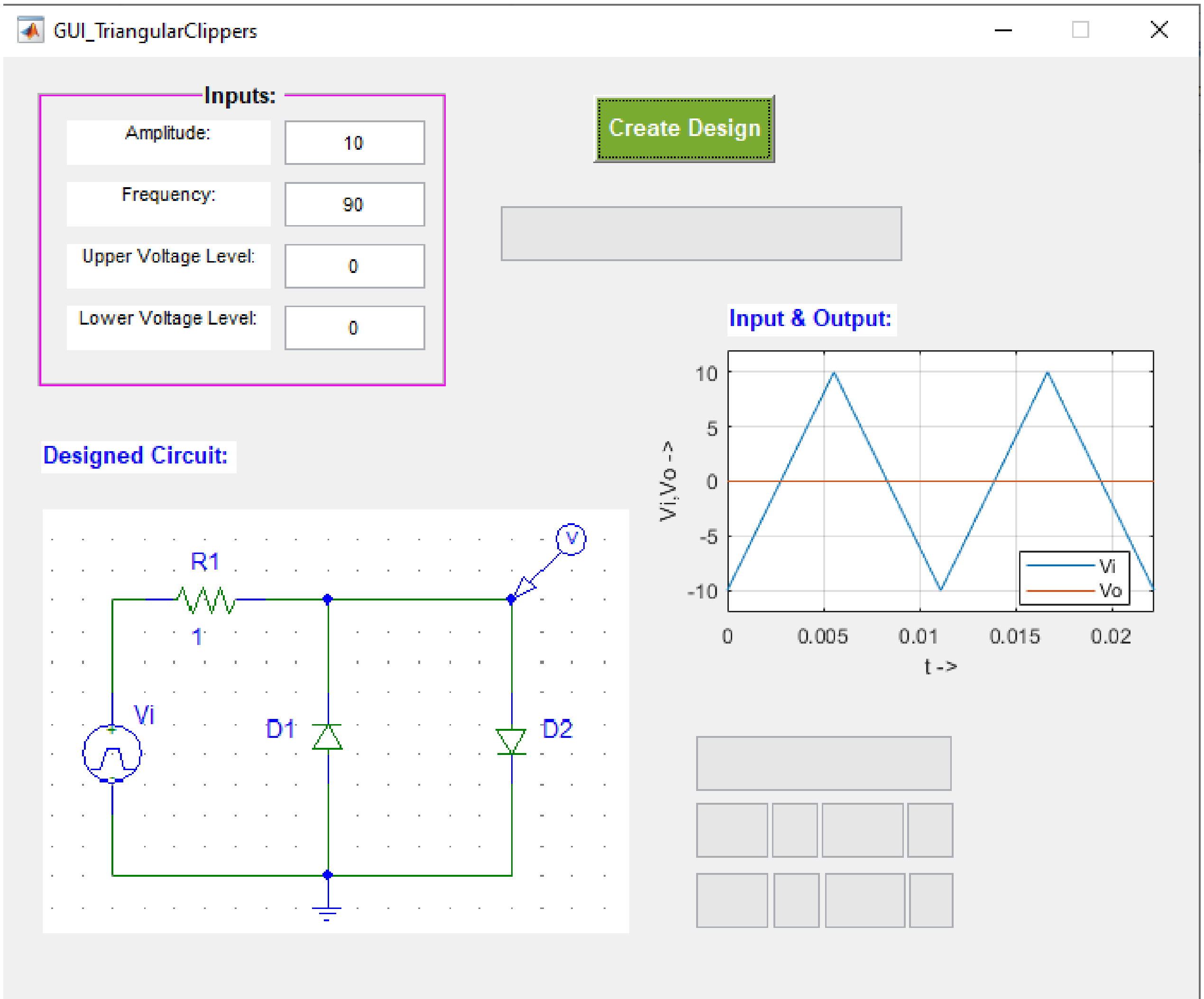
6. **Negative Side is Partially Clipped and Positive Side is Fully Clipped :**



7. Both Positive and Negative Side are Partially Clipped:

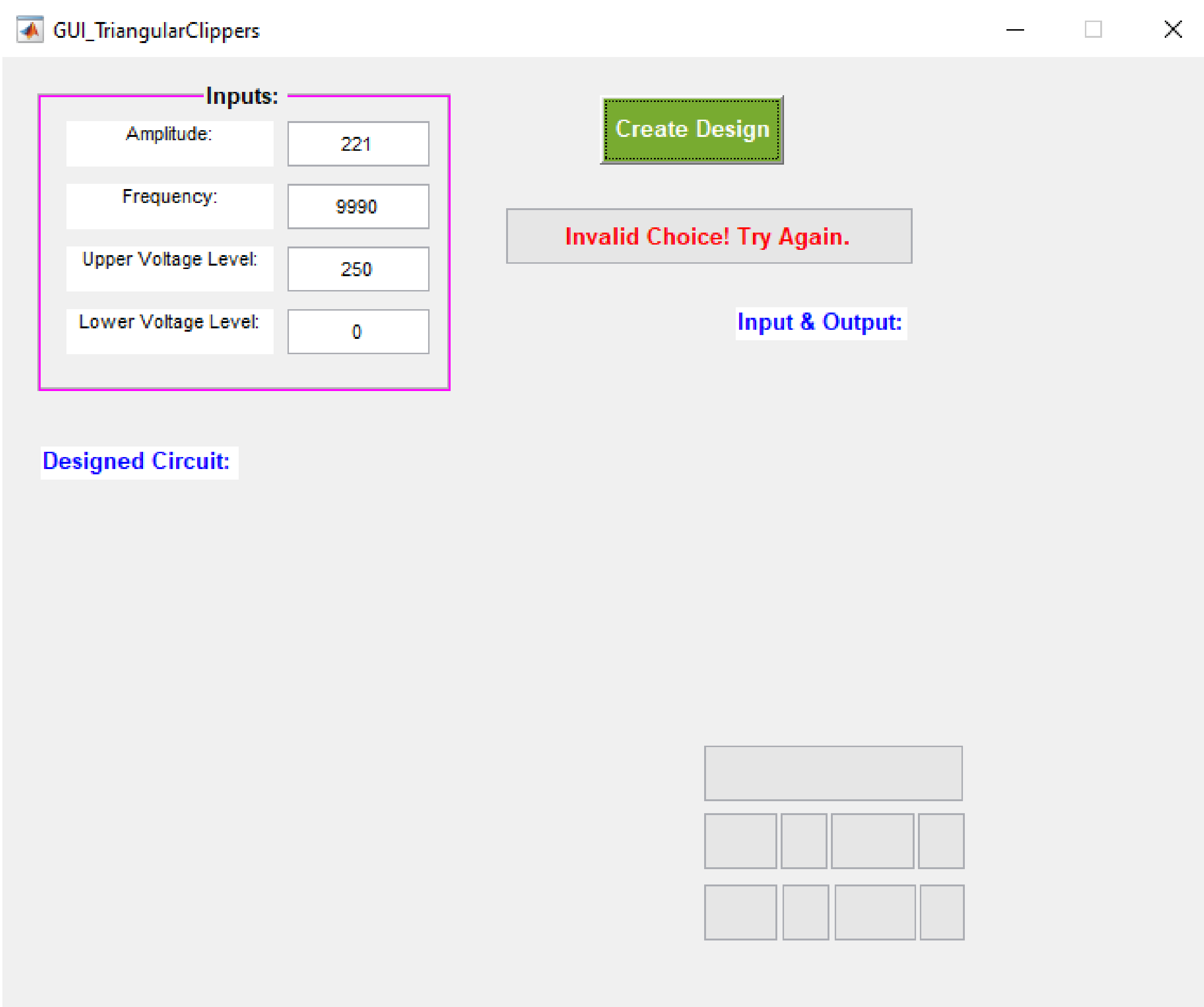


8. Both Positive and Negative Side are Fully Clipped:

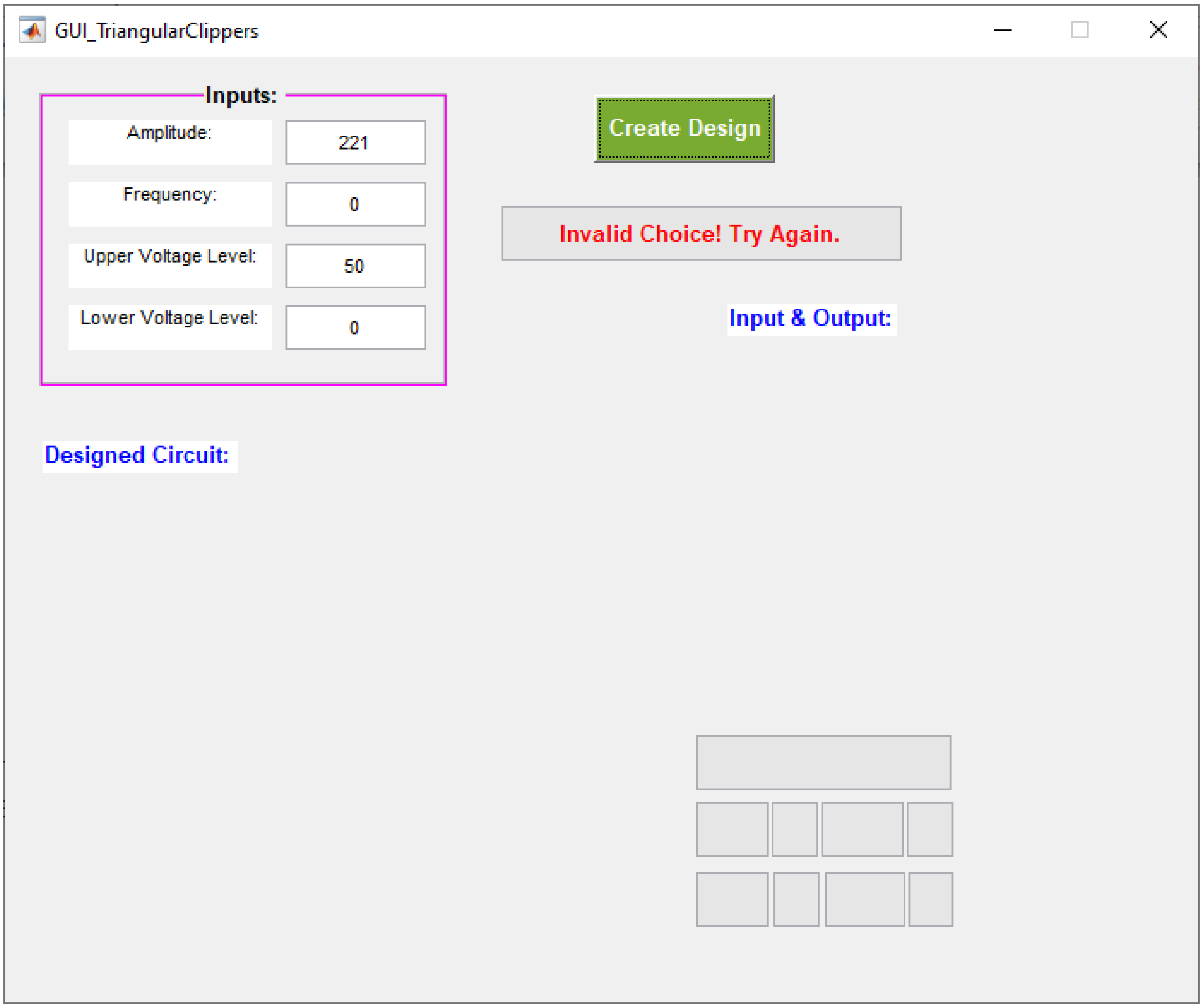


9. Invalid Cases:

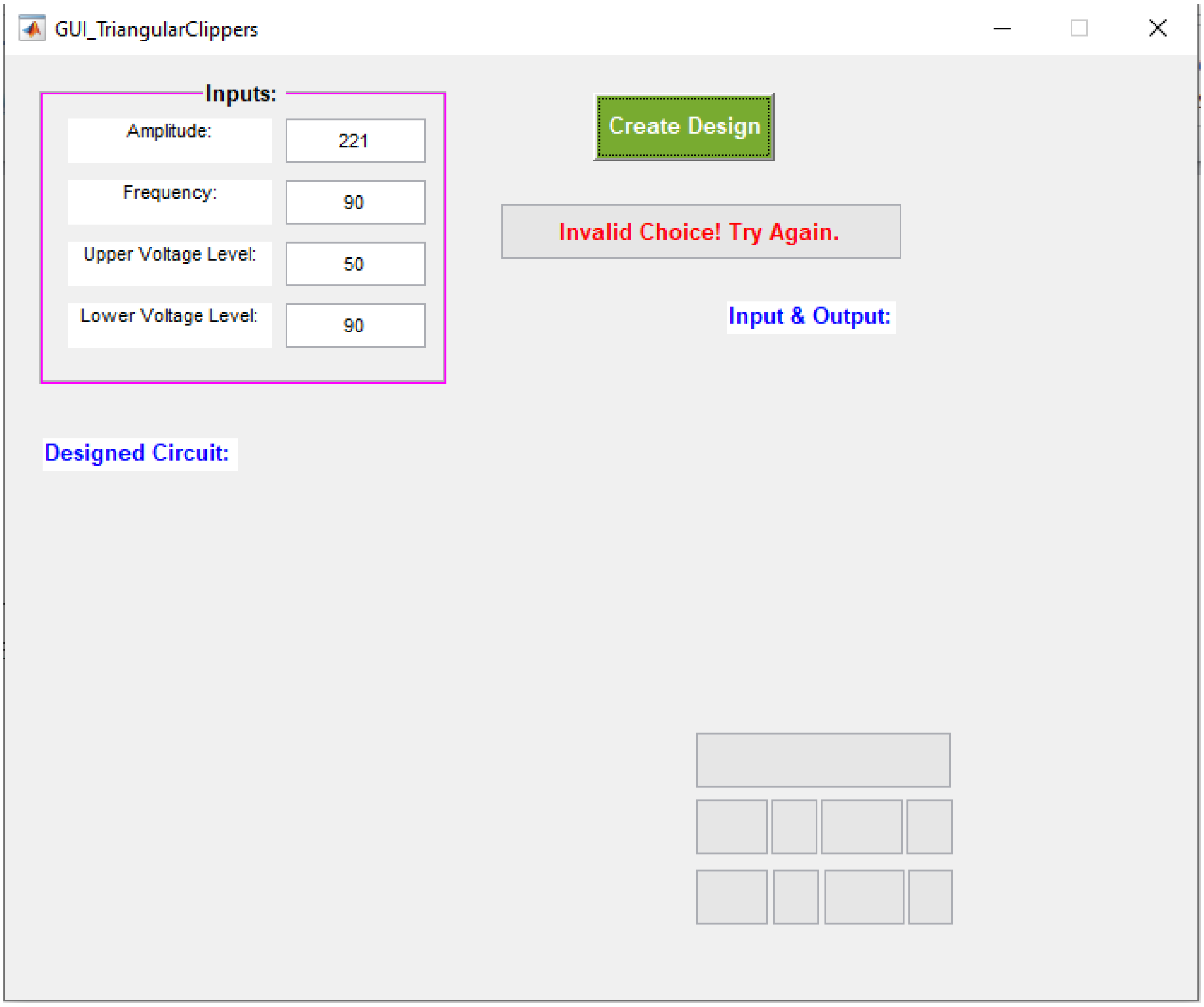
i. Clipped Voltage Level is Greater than Input Voltage Peak :



ii. Zero/ Negative Frequency:



iii. Lower Voltage Level is given greater than Upper Voltage Level:



<u>N. B:</u> Our program is able to design the circuit for any other kind of inputs as well. User is appreciated to check for more as he desires.

(iv) **<u>Sawtooth Clippers:</u>**

- Now, let us choose "Sawtooth Wave" from "GUI_Clipper" window.
- "GUI_SawtoothClippers.m" interacts with the user and calls "Sawtooth_Clipper_Design.m" that contains all necessary functions and codes to create the design and sends data back.

<u>Codes:</u>

"GUI_SawtoothClippers.m" –

```
function varargout = GUI_SawtoothClippers(varargin)
% GUI_SAWTOOTHCLIPPERS MATLAB code for GUI_SawtoothClippers.fig
% GUI_SAWTOOTHCLIPPERS, by itself, creates a new GUI_SAWTOOTHCLIPPERS or
raises the existing
% singleton*.
%
% H = GUI_SAWTOOTHCLIPPERS returns the handle to a new
GUI_SAWTOOTHCLIPPERS or the handle to
% the existing singleton*.
%
% GUI_SAWTOOTHCLIPPERS('CALLBACK',hObject,eventData,handles,...) calls the
local
% function named CALLBACK in GUI_SAWTOOTHCLIPPERS.M with the given input
arguments.
%
% GUI_SAWTOOTHCLIPPERS('Property','Value',...) creates a new
GUI_SAWTOOTHCLIPPERS or raises the
% existing singleton*.  Starting from the left, property value pairs are
% applied to the GUI before GUI_SawtoothClippers_OpeningFcn gets called.  An
% unrecognized property name or invalid value makes property application
% stop.  All inputs are passed to GUI_SawtoothClippers_OpeningFcn via varargin.
%
% *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
% instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help GUI_SawtoothClippers
```

```matlab
% Last Modified by GUIDE v2.5 21-Jul-2021 03:46:31

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @GUI_SawtoothClippers_OpeningFcn, ...
                   'gui_OutputFcn',  @GUI_SawtoothClippers_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before GUI_SawtoothClippers is made visible.
function GUI_SawtoothClippers_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to GUI_SawtoothClippers (see VARARGIN)

% Choose default command line output for GUI_SawtoothClippers
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes GUI_SawtoothClippers wait for user response (see UIRESUME)
% uiwait(handles.figure1);
```

```
% --- Outputs from this function are returned to the command line.
function varargout = GUI_SawtoothClippers_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in design.
function design_Callback(hObject, eventdata, handles)
% hObject    handle to design (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.invalidcheck,'String','');
set(handles.properties,'String','');
set(handles.v11,'String','');
set(handles.v12,'String','');
set(handles.v13,'String','');
set(handles.v14,'String','');
set(handles.v21,'String','');
set(handles.v22,'String','');
set(handles.v23,'String','');
set(handles.v24,'String','');
axes(handles.diagram);
cla;
axes(handles.plots);
cla;

amp=str2double(get(handles.amplitude,'String'));
f=str2double(get(handles.frequency,'String'));
h=str2double(get(handles.upper,'String'));
l=str2double(get(handles.lower,'String'));

if h>amp || -l>amp || f<=0 || l>h
    set(handles.invalidcheck,'String','Invalid Choice! Try Again.');
else
    [t,vin,vout,img,v1,v2]=Sawtooth_Clipper_Design(amp,f,h,l);
    axes(handles.plots);
```

```
    cla;
    plot(t,vin);
    hold on;
    plot(t,vout);
    axis([0,t(end),(-amp-amp/5),(amp+amp/5)]);
    grid on;
    legend('Vi','Vo');
    xlabel('t ->');
    ylabel('Vi,Vo ->');
    axes(handles.diagram);
    cla;
    imshow(img);
    if v1~=0 && v2==0
        set(handles.properties,'String','Where,');
        set(handles.v11,'String','V1');
        set(handles.v12,'String','=');
        set(handles.v13,'String',num2str(v1));
        set(handles.v14,'String','V');
    elseif v1~=0 && v2~=0
        set(handles.properties,'String','Where,');
        set(handles.v11,'String','V1');
        set(handles.v12,'String','=');
        set(handles.v13,'String',num2str(v1));
        set(handles.v14,'String','V');
        set(handles.v21,'String','V2');
        set(handles.v22,'String','=');
        set(handles.v23,'String',num2str(v2));
        set(handles.v24,'String','V');
    end
end


function properties_Callback(hObject, eventdata, handles)
% hObject    handle to properties (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of properties as text
% str2double(get(hObject,'String')) returns contents of properties as a double
```

```
% --- Executes during object creation, after setting all properties.
function properties_CreateFcn(hObject, eventdata, handles)
% hObject    handle to properties (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function v11_Callback(hObject, eventdata, handles)
% hObject    handle to v11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v11 as text
% str2double(get(hObject,'String')) returns contents of v11 as a double


% --- Executes during object creation, after setting all properties.
function v11_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function v12_Callback(hObject, eventdata, handles)
% hObject    handle to v12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v12 as text
% str2double(get(hObject,'String')) returns contents of v12 as a double

% --- Executes during object creation, after setting all properties.
function v12_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function v13_Callback(hObject, eventdata, handles)
% hObject    handle to v13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v13 as text
% str2double(get(hObject,'String')) returns contents of v13 as a double


% --- Executes during object creation, after setting all properties.
function v13_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function v21_Callback(hObject, eventdata, handles)
% hObject    handle to v21 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v21 as text
% str2double(get(hObject,'String')) returns contents of v21 as a double

% --- Executes during object creation, after setting all properties.
function v21_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v21 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function v22_Callback(hObject, eventdata, handles)
% hObject    handle to v22 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v22 as text
% str2double(get(hObject,'String')) returns contents of v22 as a double

% --- Executes during object creation, after setting all properties.
function v22_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v22 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
```

```
    set(hObject,'BackgroundColor','white');
end


function v23_Callback(hObject, eventdata, handles)
% hObject    handle to v23 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v23 as text
% str2double(get(hObject,'String')) returns contents of v23 as a double


% --- Executes during object creation, after setting all properties.
function v23_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v23 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end


function invalidcheck_Callback(hObject, eventdata, handles)
% hObject    handle to invalidcheck (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of invalidcheck as text
% str2double(get(hObject,'String')) returns contents of invalidcheck as a double


% --- Executes during object creation, after setting all properties.
function invalidcheck_CreateFcn(hObject, eventdata, handles)
% hObject    handle to invalidcheck (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function v14_Callback(hObject, eventdata, handles)
% hObject    handle to v14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v14 as text
% str2double(get(hObject,'String')) returns contents of v14 as a double


% --- Executes during object creation, after setting all properties.
function v14_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end


function v24_Callback(hObject, eventdata, handles)
% hObject    handle to v24 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v24 as text
% str2double(get(hObject,'String')) returns contents of v24 as a double
% --- Executes during object creation, after setting all properties.
```

```
function v24_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v24 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function amplitude_Callback(hObject, eventdata, handles)
% hObject    handle to amplitude (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of amplitude as text
% str2double(get(hObject,'String')) returns contents of amplitude as a double

% --- Executes during object creation, after setting all properties.
function amplitude_CreateFcn(hObject, eventdata, handles)
% hObject    handle to amplitude (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end


function frequency_Callback(hObject, eventdata, handles)
% hObject    handle to frequency (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of frequency as text
% str2double(get(hObject,'String')) returns contents of frequency as a double
```

```
% --- Executes during object creation, after setting all properties.
function frequency_CreateFcn(hObject, eventdata, handles)
% hObject    handle to frequency (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function upper_Callback(hObject, eventdata, handles)
% hObject    handle to upper (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of upper as text
% str2double(get(hObject,'String')) returns contents of upper as a double


% --- Executes during object creation, after setting all properties.
function upper_CreateFcn(hObject, eventdata, handles)
% hObject    handle to upper (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function lower_Callback(hObject, eventdata, handles)
% hObject    handle to lower (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of lower as text
```

```
%       str2double(get(hObject,'String')) returns contents of lower as a double
% --- Executes during object creation, after setting all properties.
function lower_CreateFcn(hObject, eventdata, handles)
% hObject    handle to lower (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

"Sawtooth_Clipper_Design.m" –

```
function [t,vin,vout,img,v1,v2]=Sawtooth_Clipper_Design(vamp,f,vup,vlow)
    if vlow==0 && vup==vamp
        [t,vin,vout,img,v1,v2]=negative_clipper(vamp,f,vup,vlow);
    elseif vup==0 && vlow==-(vamp)
        [t,vin,vout,img,v1,v2]=positive_clipper(vamp,f,vup,vlow);
    elseif vlow==0 && vup==0
        [t,vin,vout,img,v1,v2]=both_negative_positive_clipper(vamp,f,vup,vlow);
    elseif 0<vup<vamp && vlow==-(vamp)
        [t,vin,vout,img,v1,v2]=partial_positive_clipper(vamp,f,vup,vlow);
    elseif 0<-(vlow)<vamp && vup==vamp
        [t,vin,vout,img,v1,v2]=partial_negative_clipper(vamp,f,vup,vlow);
    elseif 0<vup<vamp && vlow==0
        [t,vin,vout,img,v1,v2]=partial_positive_full_negative_clipper(vamp,f,vup,vlow);
    elseif vup==0 && 0<-(vlow)<vamp
        [t,vin,vout,img,v1,v2]=partial_negative_full_positive_clipper(vamp,f,vup,vlow);
    elseif 0<vup<vamp && 0<-(vlow)<vamp
        [t,vin,vout,img,v1,v2]=partial_negative_positive_clipper(vamp,f,vup,vlow);
    end
end

function [t,vin,vout,img,v1,v2]=negative_clipper(vamp,f,vup,vlow)
    v1=0;
    v2=0;
    T=1/f;
```

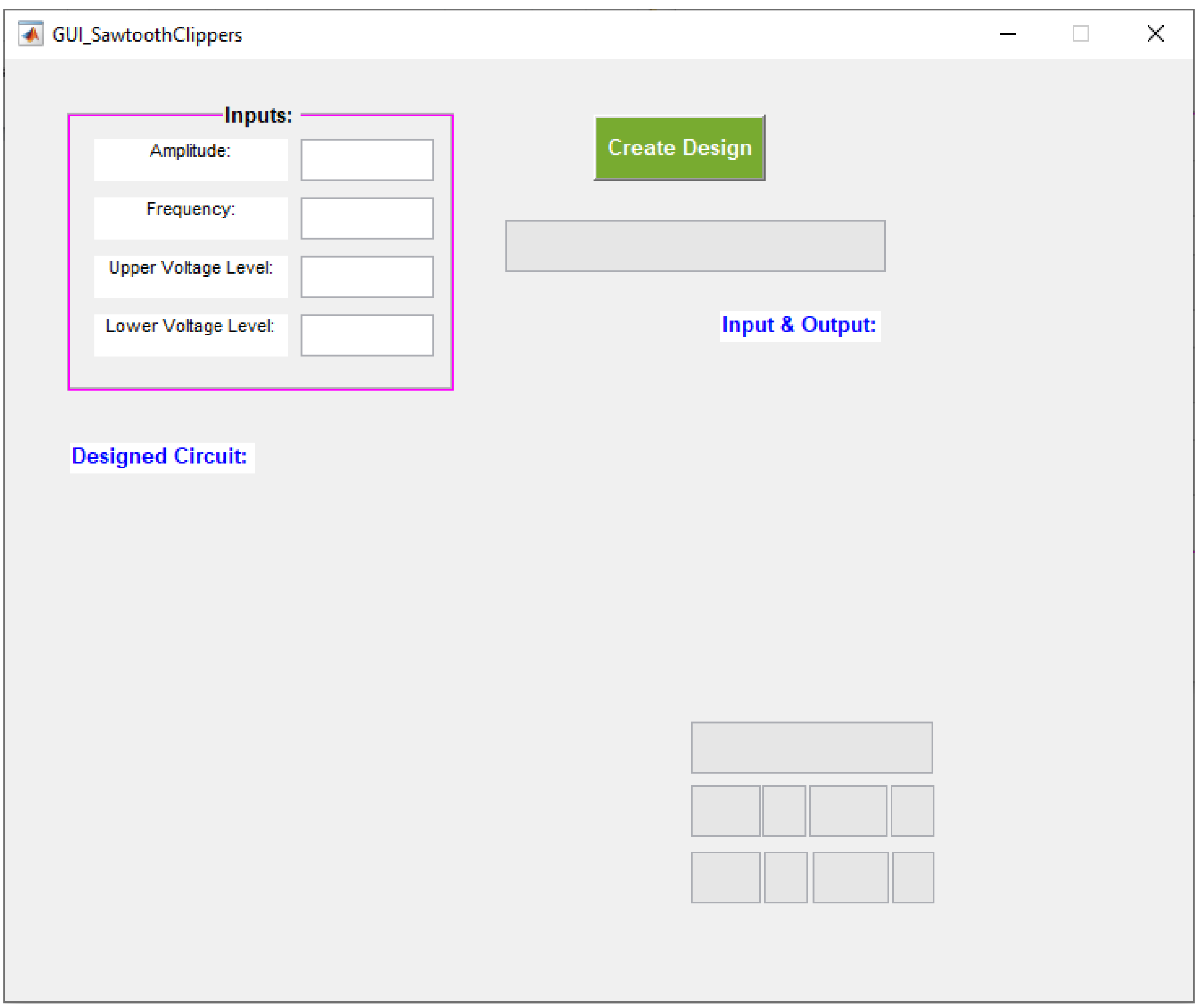
```
    t=linspace(0,2*T,1000);
    vin=vamp.*sawtooth((2*pi*f)*t);
    vout=[];
    for i=1:1000
        if vin(i)>=0
            vout(i)=vin(i);
        else
            vout(i)=0;
        end
    end
    img=imread('SawtoothClipper2.PNG');
end

function [t,vin,vout,img,v1,v2]=positive_clipper(vamp,f,vup,vlow)
    v1=0;
    v2=0;
    T=1/f;
    t=linspace(0,2*T,1000);
    vin=vamp.*sawtooth((2*pi*f)*t);
    vout=[];
    for i=1:1000
        if vin(i)<=0
            vout(i)=vin(i);
        else
            vout(i)=0;
        end
    end
    img=imread('SawtoothClipper1.PNG');
end

function [t,vin,vout,img,v1,v2]=partial_positive_clipper(vamp,f,vup,vlow)
    v1=vup;
    v2=0;
    T=1/f;
    t=linspace(0,2*T,1000);
    vin=vamp.*sawtooth((2*pi*f)*t);
    vout=[];
    for i=1:1000
        if vin(i)>=v1
            vout(i)=v1;
```

```
        else
            vout(i)=vin(i);
        end
    end
    img=imread('SawtoothClipper3.PNG');
end

function [t,vin,vout,img,v1,v2]=partial_negative_clipper(vamp,f,vup,vlow)
    v1=-vlow;
    v2=0;
    T=1/f;
    t=linspace(0,2*T,1000);
    vin=vamp.*sawtooth((2*pi*f)*t);
    vout=[];
    for i=1:1000
        if vin(i)<=-v1
            vout(i)=-v1;
        else
            vout(i)=vin(i);
        end
    end
    img=imread('SawtoothClipper4.PNG');
end

function [t,vin,vout,img,v1,v2]=partial_positive_full_negative_clipper(vamp,f,vup,vlow)
    v1=vup;
    v2=0;
    T=1/f;
    t=linspace(0,2*T,1000);
    vin=vamp.*sawtooth((2*pi*f)*t);
    vout=[];
    for i=1:1000
        if vin(i)>=vup
            vout(i)=v1;
        elseif vin(i)<0
            vout(i)=0;
        else
            vout(i)=vin(i);
        end
    end
```

```
    img=imread('SawtoothClipper5.PNG');
end

function [t,vin,vout,img,v1,v2]=partial_negative_full_positive_clipper(vamp,f,vup,vlow)
    v1=-(vlow);
    v2=0;
    T=1/f;
    t=linspace(0,2*T,1000);
    vin=vamp.*sawtooth((2*pi*f)*t);
    vout=[];
    for i=1:1000
        if vin(i)>=0
            vout(i)=0;
        elseif vin(i)<=vlow
            vout(i)=-v1;
        else
            vout(i)=vin(i);
        end
    end
    img=imread('SawtoothClipper6.PNG');
end

function [t,vin,vout,img,v1,v2]=partial_negative_positive_clipper(vamp,f,vup,vlow)
    v1=-(vlow);
    v2=vup;
    T=1/f;
    t=linspace(0,2*T,1000);
    vin=vamp.*sawtooth((2*pi*f)*t);
    vout=[];
    for i=1:1000
        if vin(i)>=vup
            vout(i)=v2;
        elseif vin(i)<=vlow
            vout(i)=-v1;
        else
            vout(i)=vin(i);
        end
    end
    img=imread('SawtoothClipper7.PNG');
end
```

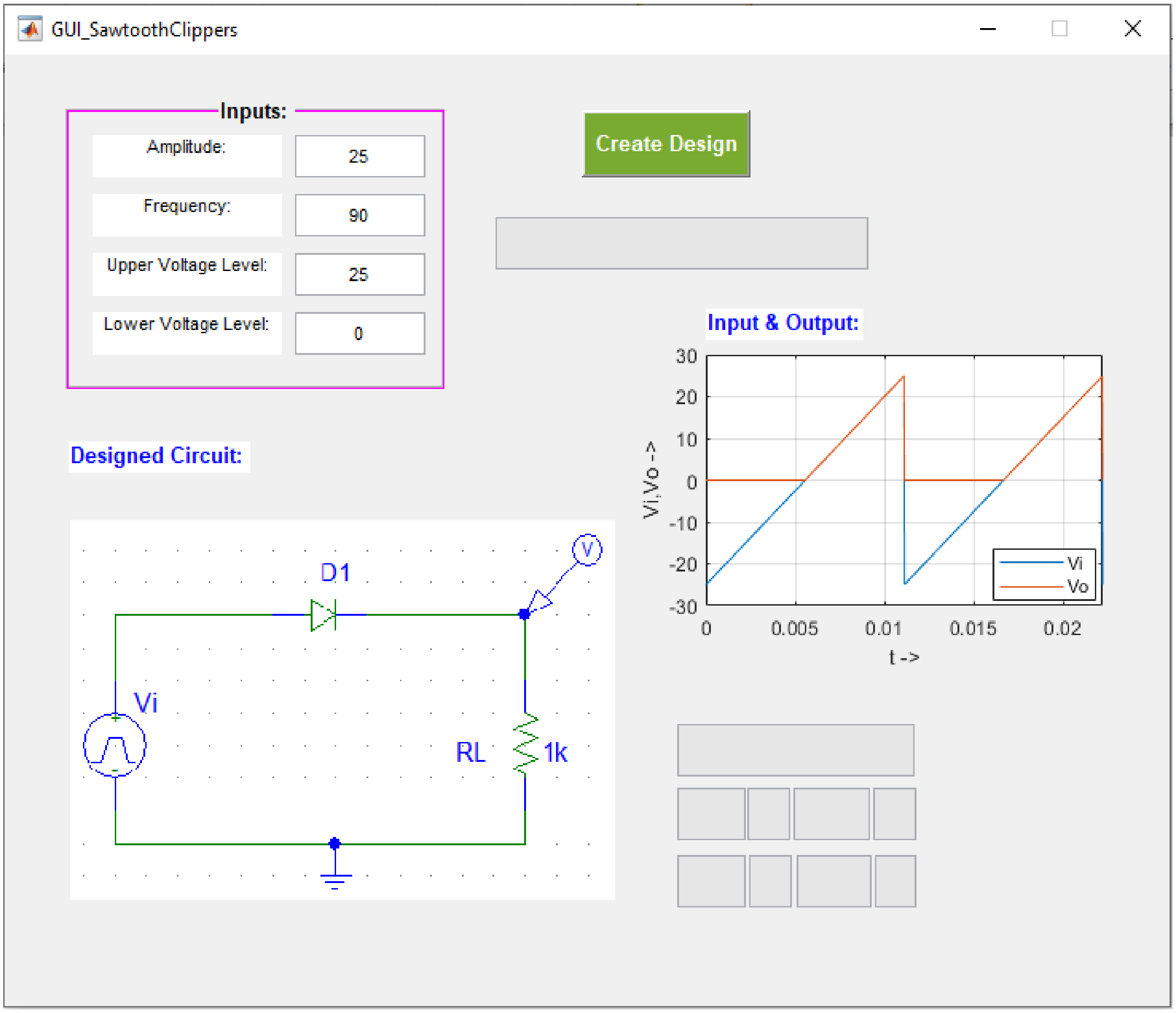
```
function [t,vin,vout,img,v1,v2]=both_negative_positive_clipper(vamp,f,vup,vlow)
   v1=0;
   v2=0;
   T=1/f;
   t=linspace(0,2*T,1000);
   vin=vamp.*sawtooth((2*pi*f)*t);
   vout=[];
   for i=1:1000
      vout(i)=0;
   end
   img=imread('SawtoothClipper8.PNG');
end
```

Corresponding Window:

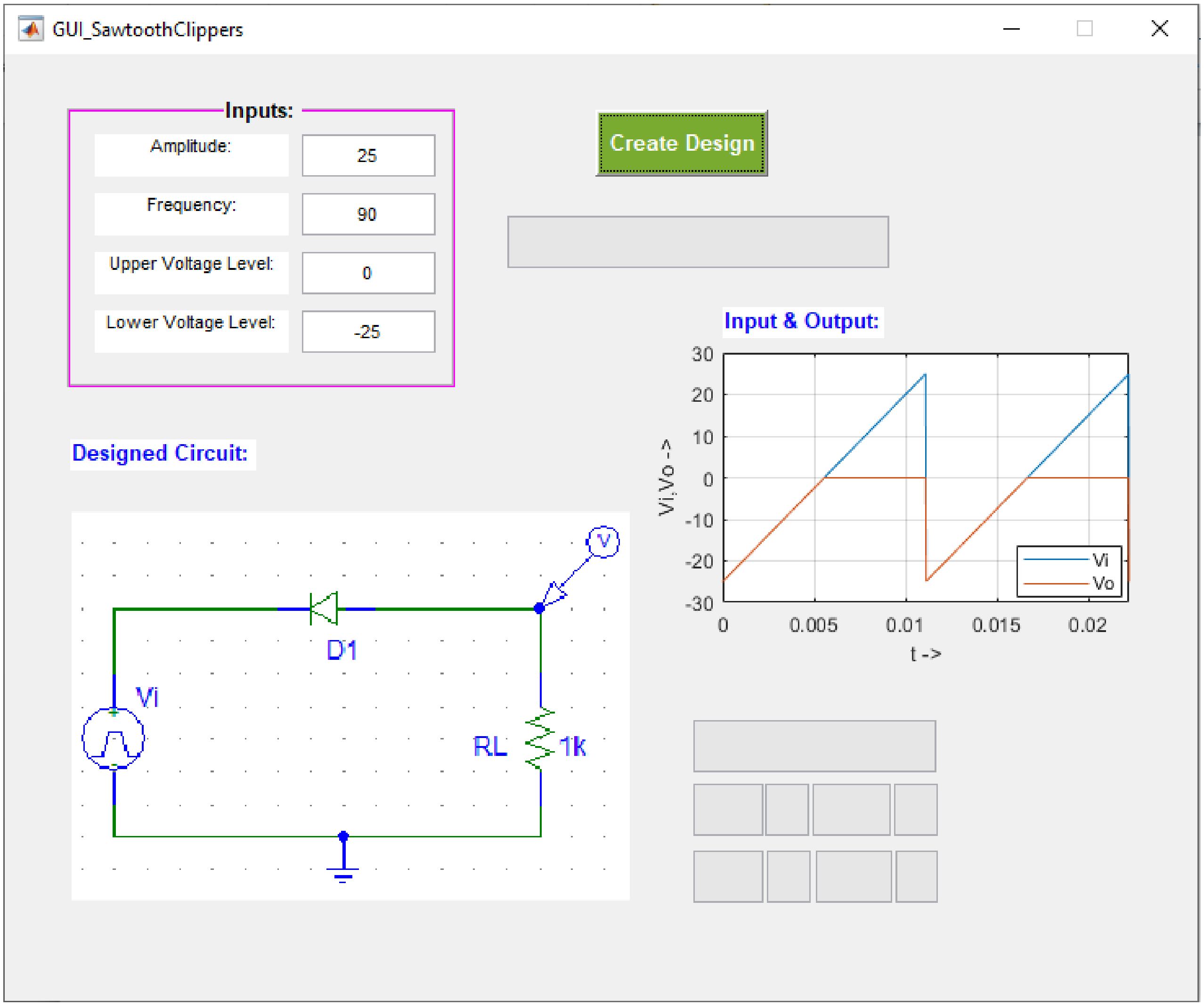


Sample Outputs:

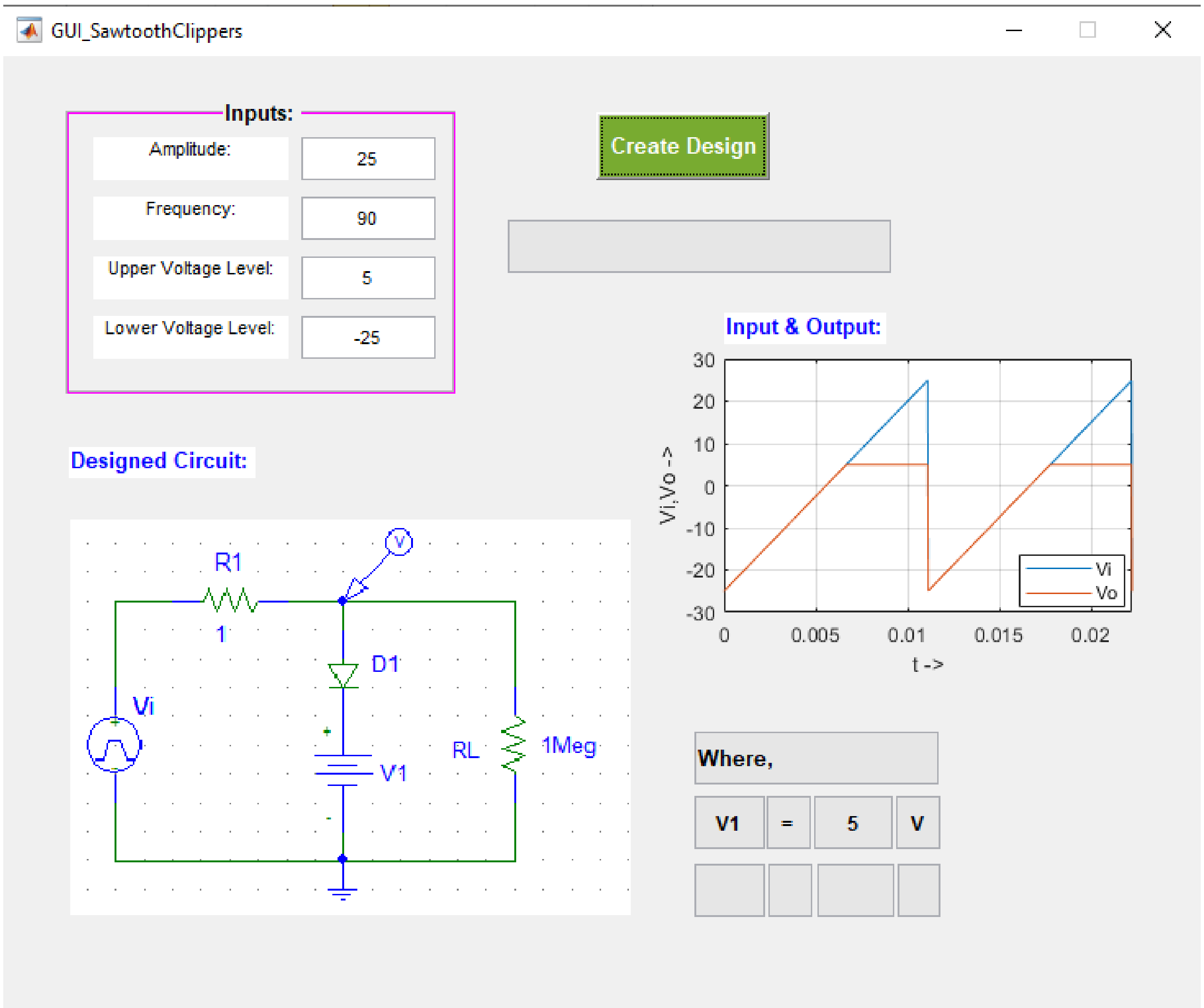
1. Negative Clipper :



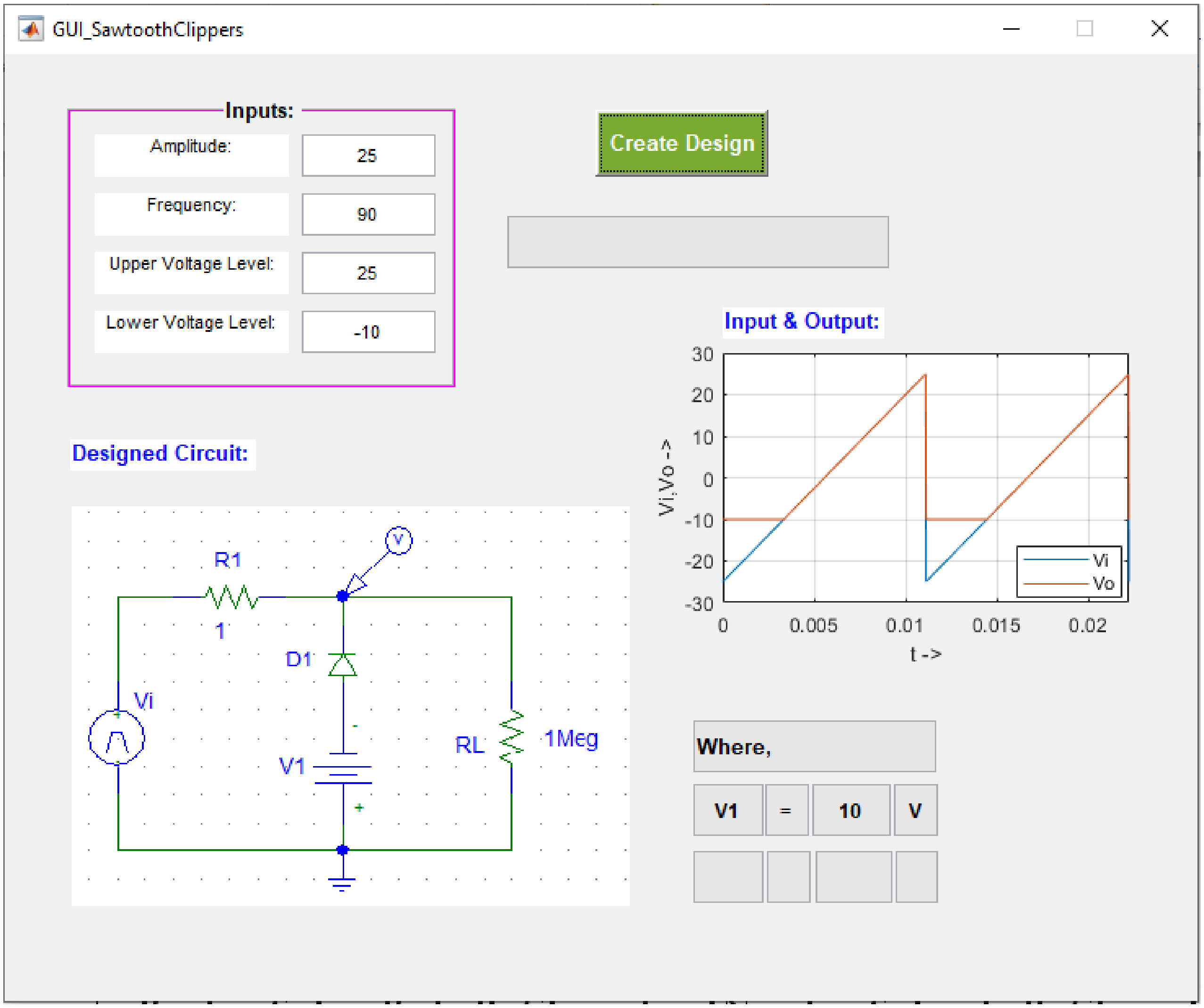
2. Positive Clipper :



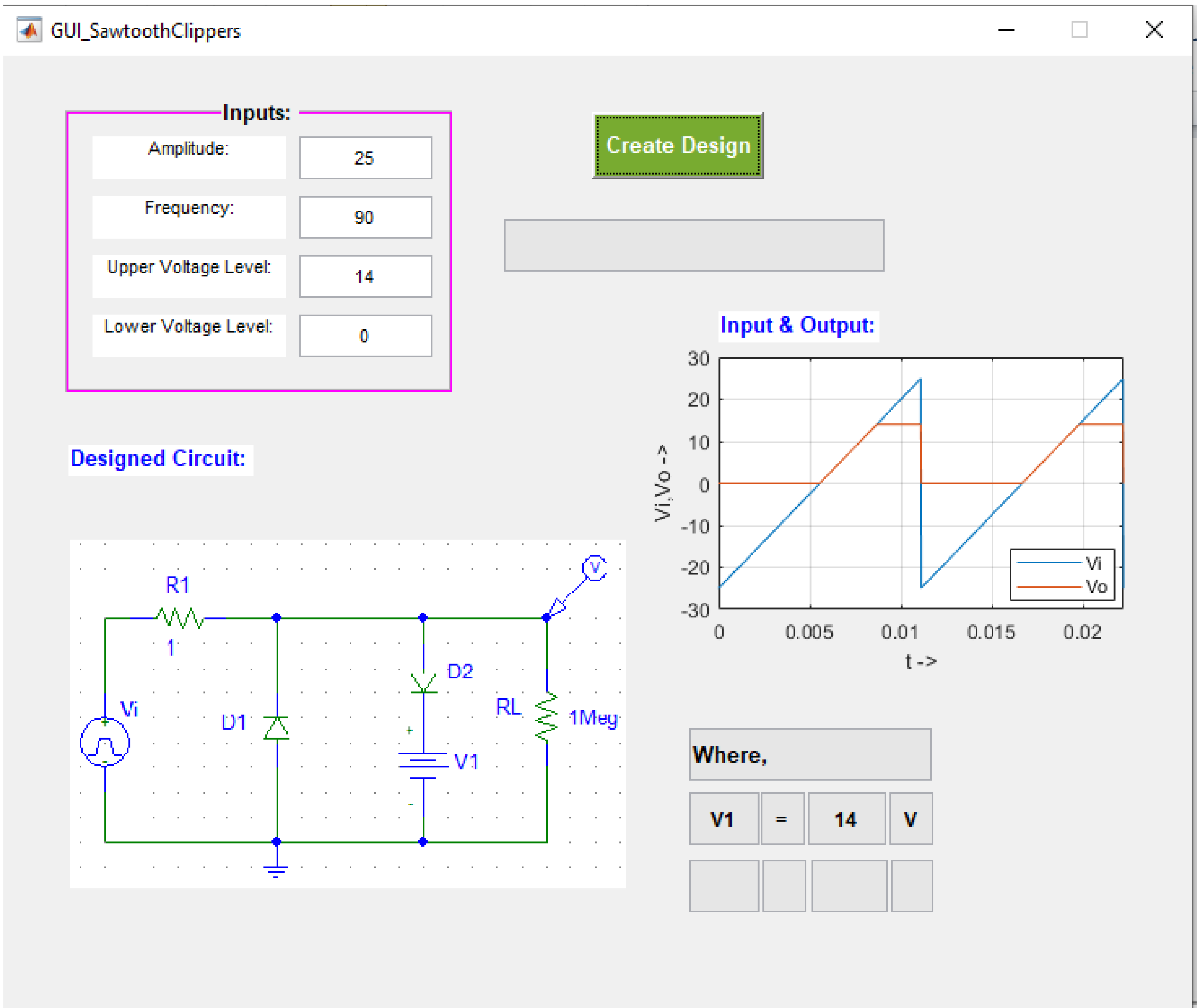
3. Clipping Positive Side Partially:



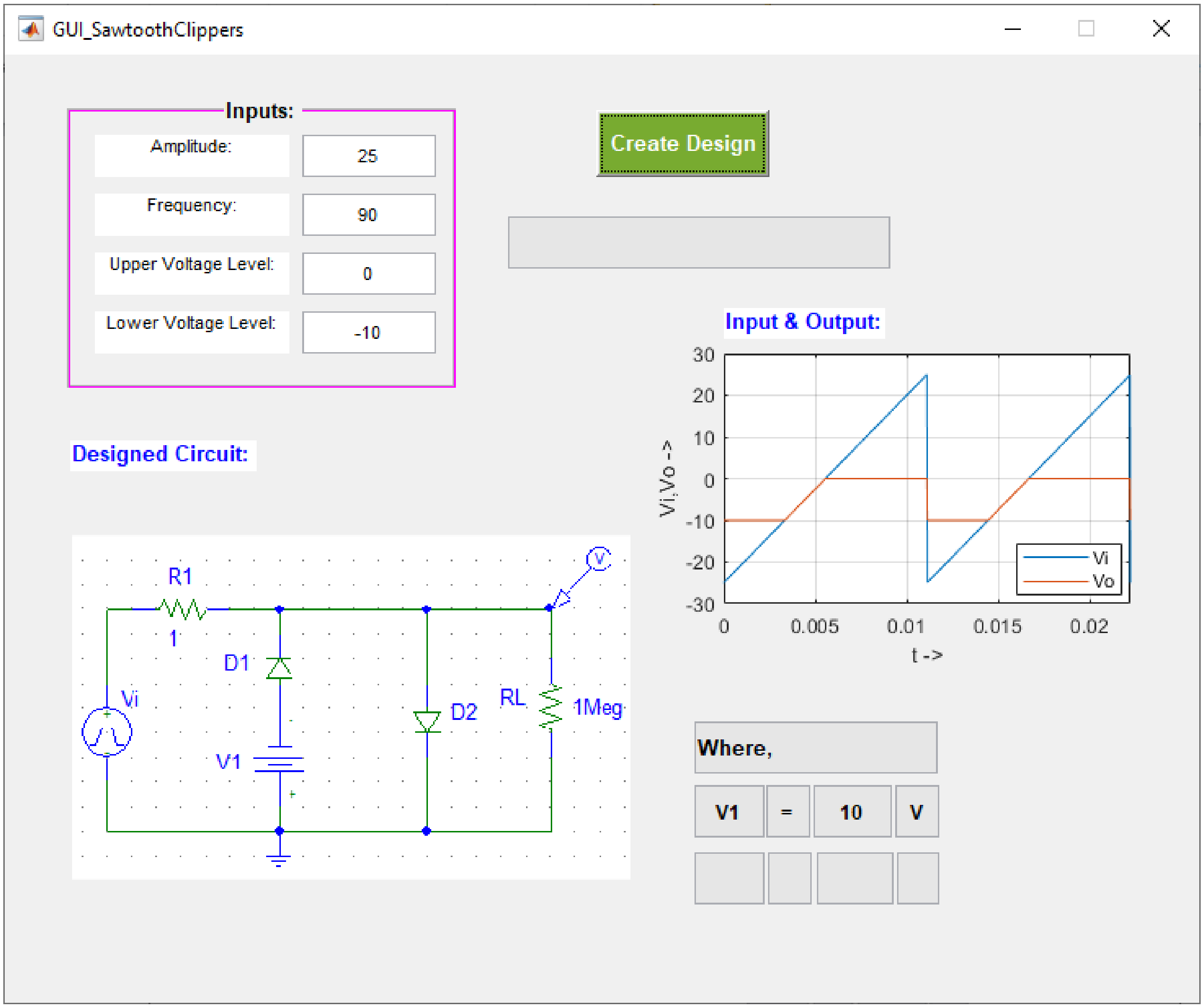
4. Clipping Negative Side Partially:



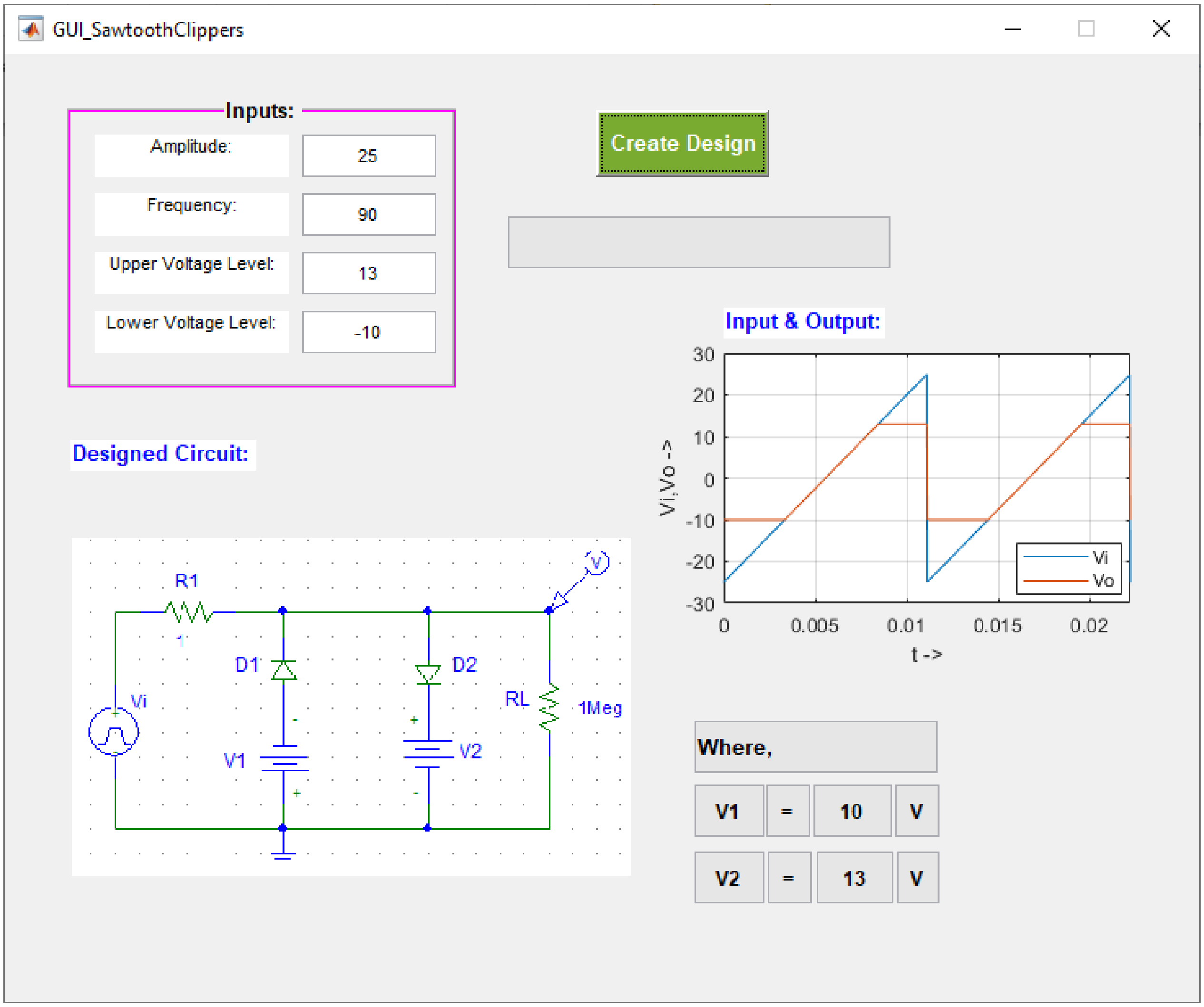
5. **Positive Side is Partially Clipped and Negative Side is Fully Clipped :**



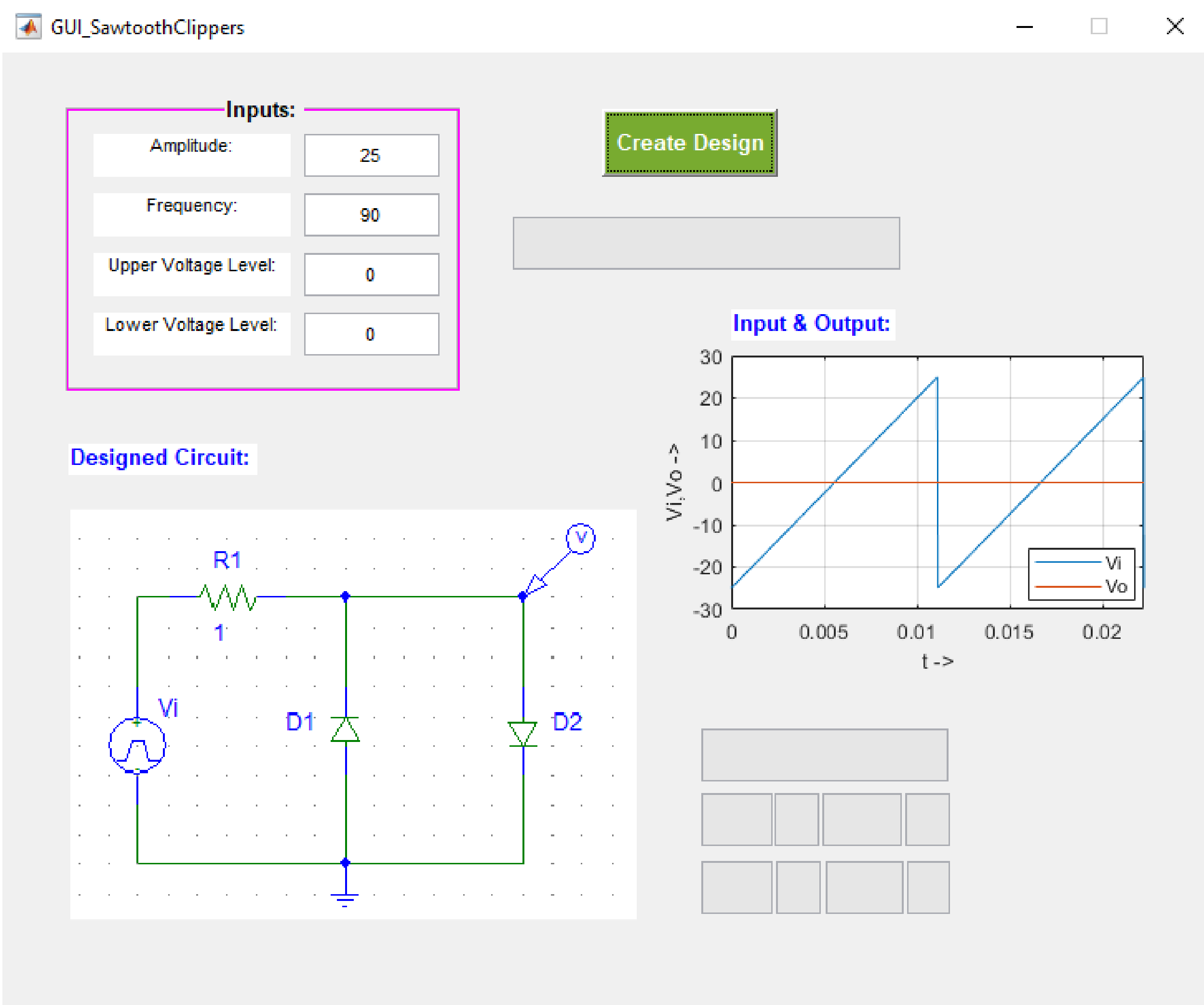
6. Negative Side is Partially Clipped and Positive Side is Fully Clipped :



7. Both Positive and Negative Side are Partially Clipped:

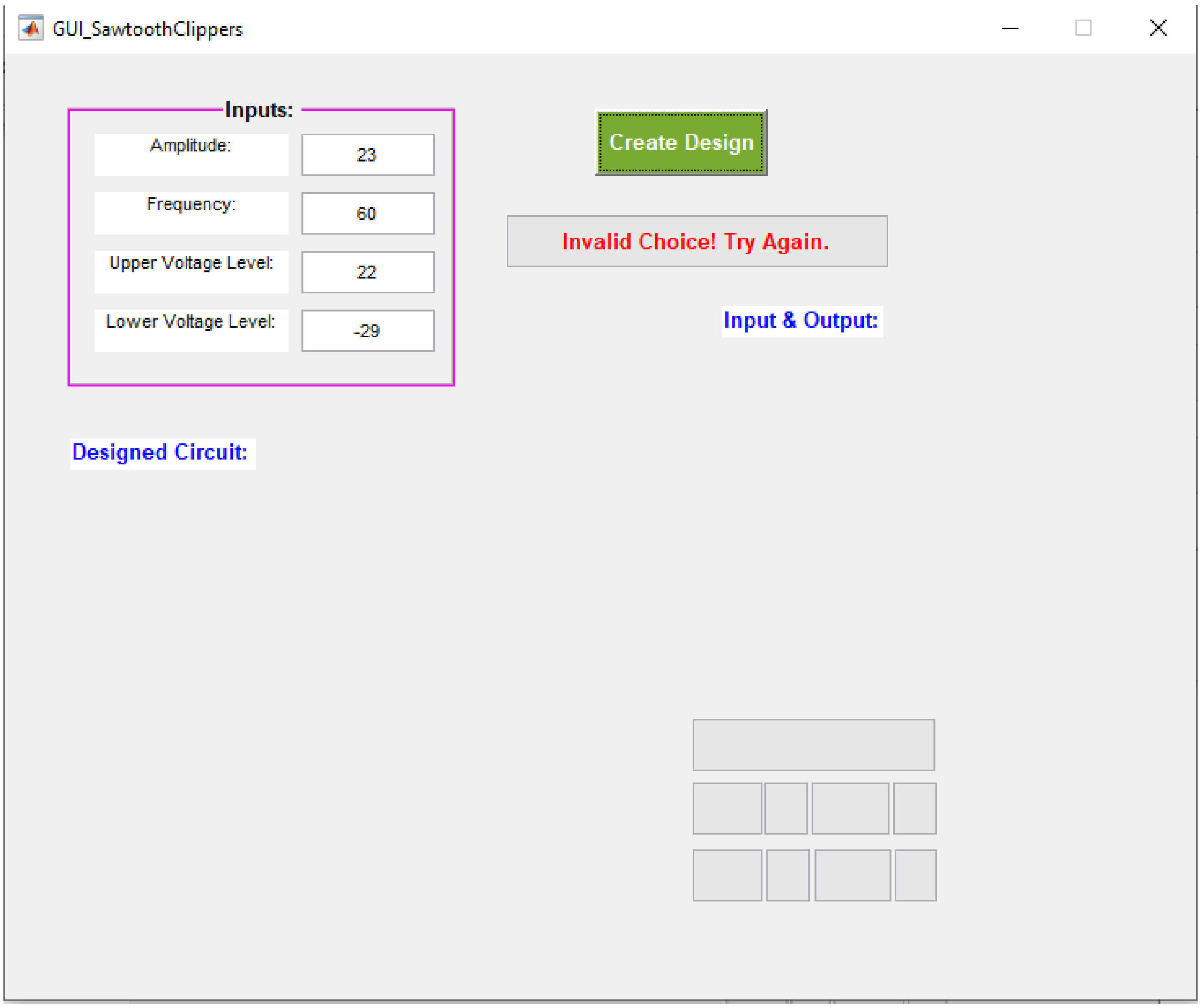


8. Both Positive and Negative Side are Fully Clipped:

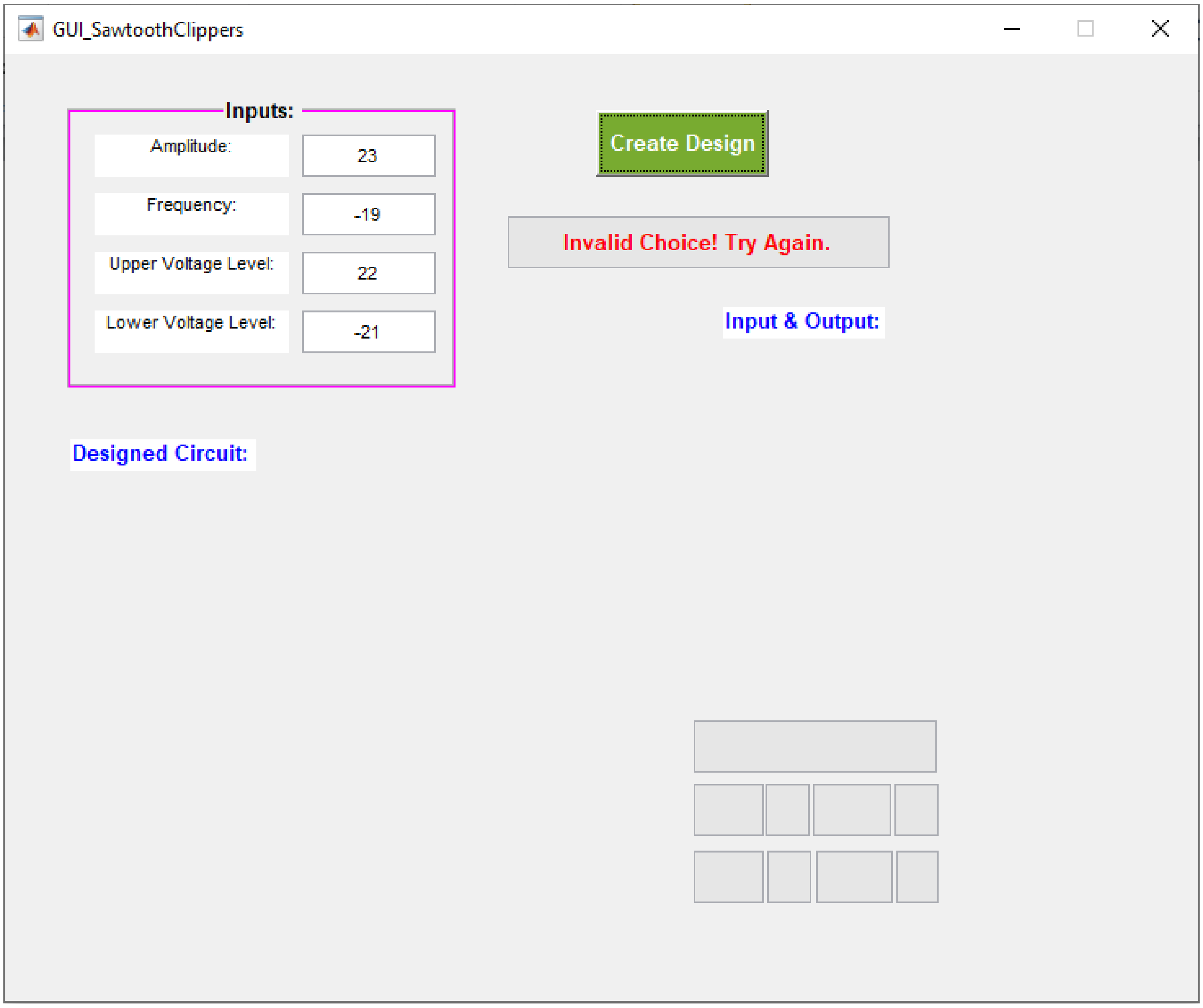


9. Invalid Cases:

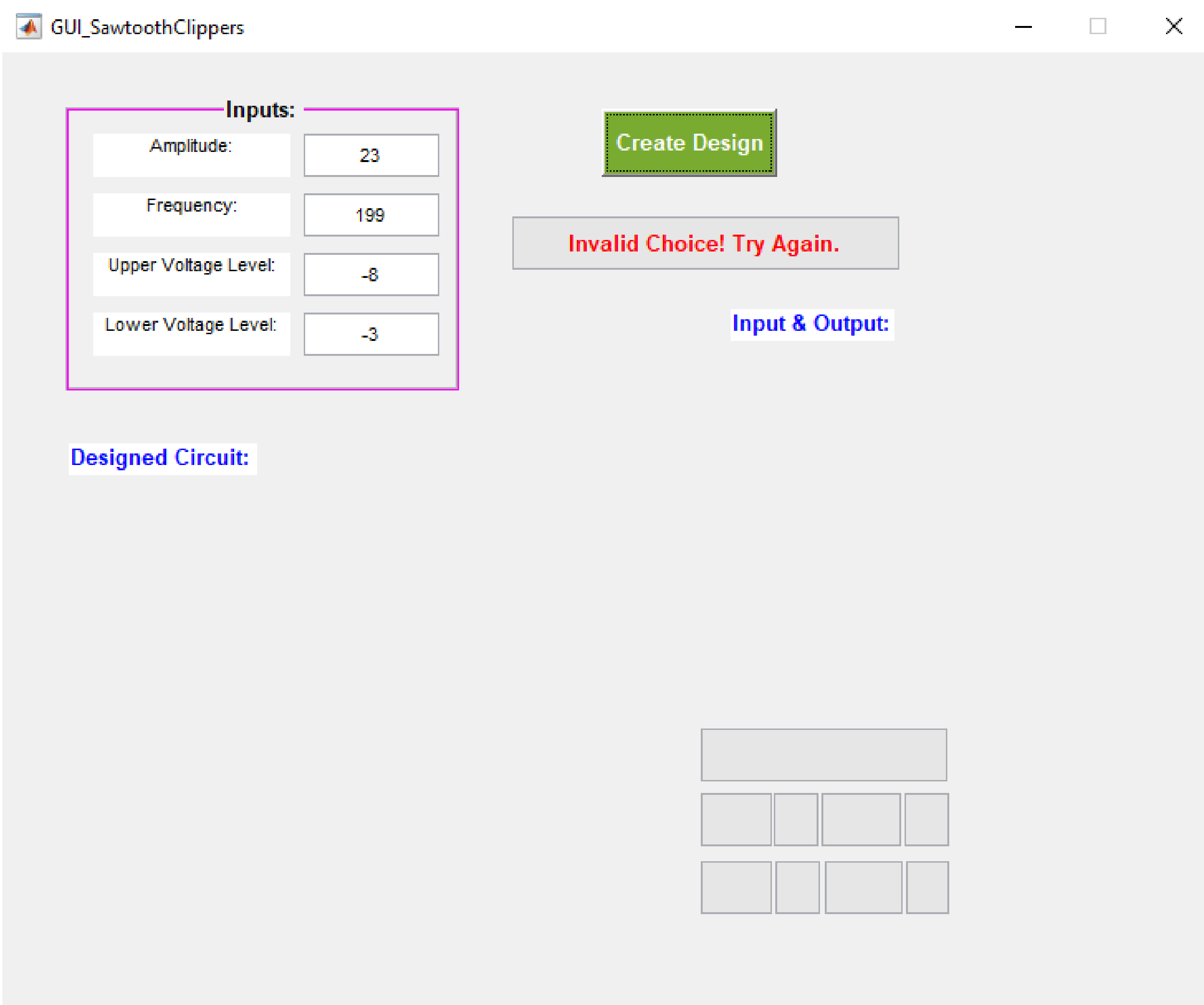
i. Clipped Voltage Level is Greater than Input Voltage Peak :



ii. Zero/ Negative Frequency:



iii. Lower Voltage Level is given greater than Upper Voltage Level:



<u>N. B:</u> Our program is able to design the circuit for any other kind of inputs as well. User is appreciated to check for more as he desires.

2. Clamping:

- Now suppose, “Clamping” has been chosen from the home tab. It will direct us to “GUI_Clamper.m”.

Code:

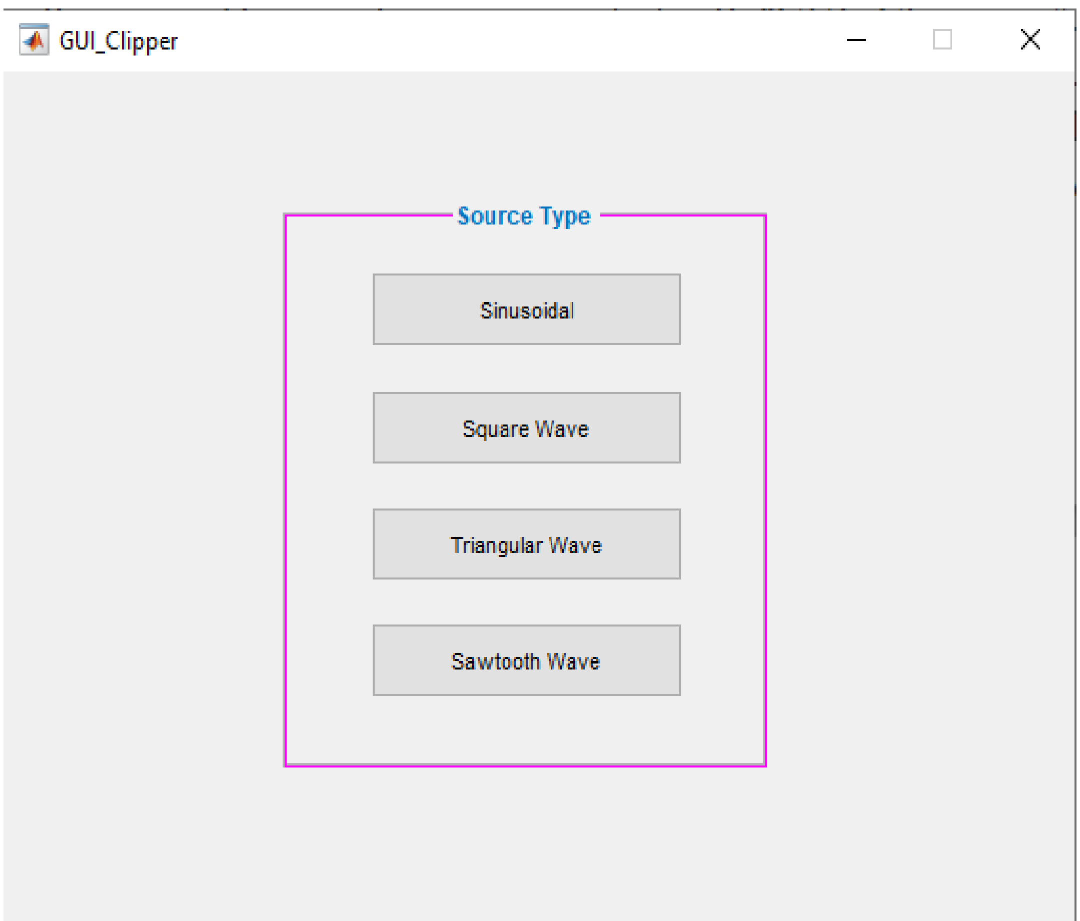
“GUI_Clamper.m” -

```
function varargout = GUI_Clamper(varargin)
% GUI_CLAMPER MATLAB code for GUI_Clamper.fig
% GUI_CLAMPER, by itself, creates a new GUI_CLAMPER or raises the existing
% singleton*.
% H = GUI_CLAMPER returns the handle to a new GUI_CLAMPER or the handle to
% the existing singleton*.
% GUI_CLAMPER('CALLBACK',hObject,eventData,handles,...) calls the local
% function named CALLBACK in GUI_CLAMPER.M with the given input arguments.
% GUI_CLAMPER('Property','Value',...) creates a new GUI_CLAMPER or raises the
% existing singleton*.  Starting from the left, property value pairs are
% applied to the GUI before GUI_Clamper_OpeningFcn gets called.  An
% unrecognized property name or invalid value makes property application
% stop.  All inputs are passed to GUI_Clamper_OpeningFcn via varargin.
% *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
% instance to run (singleton)".
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help GUI_Clamper
% Last Modified by GUIDE v2.5 20-Jul-2021 05:16:57
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @GUI_Clamper_OpeningFcn, ...
                   'gui_OutputFcn',  @GUI_Clamper_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
```

```
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before GUI_Clamper is made visible.
function GUI_Clamper_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to GUI_Clamper (see VARARGIN)

% Choose default command line output for GUI_Clamper
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes GUI_Clamper wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = GUI_Clamper_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in Sinusoidal.
function Sinusoidal_Callback(hObject, eventdata, handles)
% hObject    handle to Sinusoidal (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
GUI_SinusoidalClampers

% --- Executes on button press in SquareWave.
function SquareWave_Callback(hObject, eventdata, handles)
% hObject    handle to SquareWave (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

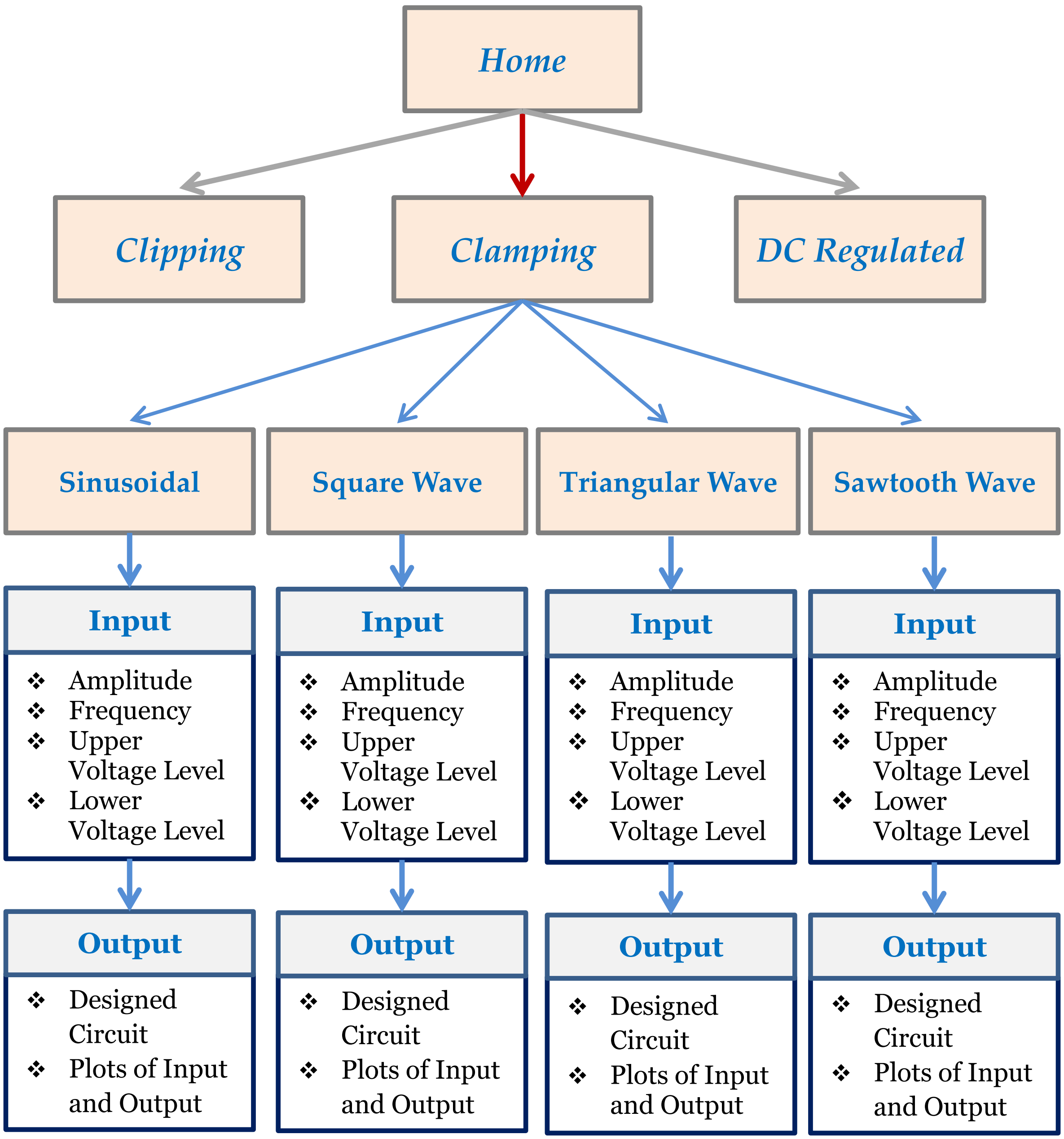
```
GUI_SquareClampers

% --- Executes on button press in TriangularWave.
function TriangularWave_Callback(hObject, eventdata, handles)
% hObject    handle to TriangularWave (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
GUI_TriangularClampers

% --- Executes on button press in SawtoothWave.
function SawtoothWave_Callback(hObject, eventdata, handles)
% hObject    handle to SawtoothWave (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
GUI_SawtoothClampers
```

Corresponding Window:



- We will work with all four types of sources one by one.

Flow Chart to Prform Clamping :



(i) **Sinusoidal Clampers:**

- For the time being, let us choose "Sinusoidal" source first.
- "GUI_SinusoidalClampers.m" interacts with the user and calls "Sinusoidal_Clamper_Design.m" that contains all necessary functions and codes to create the design and sends data back.

Codes:

"GUI_SinusoidalClampers.m" –

```
function varargout = GUI_SinusoidalClampers(varargin)
% GUI_SINUSOIDALCLAMPERS MATLAB code for GUI_SinusoidalClampers.fig
% GUI_SINUSOIDALCLAMPERS, by itself, creates a new
GUI_SINUSOIDALCLAMPERS or raises the existing
% singleton*.
%
% H = GUI_SINUSOIDALCLAMPERS returns the handle to a new
GUI_SINUSOIDALCLAMPERS or the handle to
% the existing singleton*.
%
% GUI_SINUSOIDALCLAMPERS('CALLBACK',hObject,eventData,handles,...) calls the
local
% function named CALLBACK in GUI_SINUSOIDALCLAMPERS.M with the given
input arguments.
%
% GUI_SINUSOIDALCLAMPERS('Property','Value',...) creates a new
GUI_SINUSOIDALCLAMPERS or raises the
% existing singleton*.  Starting from the left, property value pairs are
% applied to the GUI before GUI_SinusoidalClampers_OpeningFcn gets called.  An
% unrecognized property name or invalid value makes property application
% stop.  All inputs are passed to GUI_SinusoidalClampers_OpeningFcn via varargin.
%
% *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
% instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help GUI_SinusoidalClampers
```

```
% Last Modified by GUIDE v2.5 18-Jul-2021 02:34:10

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @GUI_SinusoidalClampers_OpeningFcn, ...
                   'gui_OutputFcn',  @GUI_SinusoidalClampers_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before GUI_SinusoidalClampers is made visible.
function GUI_SinusoidalClampers_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to GUI_SinusoidalClampers (see VARARGIN)

% Choose default command line output for GUI_SinusoidalClampers
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes GUI_SinusoidalClampers wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = GUI_SinusoidalClampers_OutputFcn(hObject, eventdata, handles)
```

```
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in design.
function design_Callback(hObject, eventdata, handles)
% hObject    handle to design (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.invalidcheck,'String','');
set(handles.properties,'String','');
set(handles.c11,'String','');
set(handles.c12,'String','');
set(handles.c13,'String','');
set(handles.c14,'String','');
set(handles.v11,'String','');
set(handles.v12,'String','');
set(handles.v13,'String','');
set(handles.v14,'String','');
axes(handles.diagram);
cla;
axes(handles.plots);
cla;

amp=str2double(get(handles.amplitude,'String'));
f=str2double(get(handles.frequency,'String'));
h=str2double(get(handles.upper,'String'));
l=str2double(get(handles.lower,'String'));

if  f<=0 || (h-l)~=(2*amp)
    set(handles.invalidcheck,'String','Invalid Choice! Try Again.');
else
    [t,vin,vout,img,c1,v1]=Sinusoidal_Clamper_Design(amp,f,h,l);
    axes(handles.plots);
    cla;
    plot(t,vin);
```

```
    hold on;
    plot(t,vout);
    if max(vout)>=max(vin)
        axis([0,t(end),(min(vin)-amp/2),(max(vout)+amp/2)]);
    else
        axis([0,t(end),(min(vout)-amp/2),(max(vin)+amp/2)]);
    end
    grid on;
    legend('Vi','Vo');
    xlabel('t ->');
    ylabel('Vi,Vo ->');
    axes(handles.diagram);
    cla;
    imshow(img);
    if  v1==0
        set(handles.properties,'String','Where,');
        set(handles.c11,'String','C1');
        set(handles.c12,'String','=');
        set(handles.c13,'String',num2str(c1));
        set(handles.c14,'String','F');
    else
        set(handles.properties,'String','Where,');
        set(handles.c11,'String','C1');
        set(handles.c12,'String','=');
        set(handles.c13,'String',num2str(c1));
        set(handles.c14,'String','F');
        set(handles.v11,'String','V1');
        set(handles.v12,'String','=');
        set(handles.v13,'String',num2str(v1));
        set(handles.v14,'String','V');
    end
end


function properties_Callback(hObject, eventdata, handles)
% hObject    handle to properties (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of properties as text
```

```
%       str2double(get(hObject,'String')) returns contents of properties as a double

% --- Executes during object creation, after setting all properties.
function properties_CreateFcn(hObject, eventdata, handles)
% hObject    handle to properties (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function c11_Callback(hObject, eventdata, handles)
% hObject    handle to c11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of c11 as text
%        str2double(get(hObject,'String')) returns contents of c11 as a double

% --- Executes during object creation, after setting all properties.
function c11_CreateFcn(hObject, eventdata, handles)
% hObject    handle to c11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function c12_Callback(hObject, eventdata, handles)
% hObject    handle to c12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of c12 as text
% str2double(get(hObject,'String')) returns contents of c12 as a double


% --- Executes during object creation, after setting all properties.
function c12_CreateFcn(hObject, eventdata, handles)
% hObject    handle to c12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end


function c13_Callback(hObject, eventdata, handles)
% hObject    handle to c13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of c13 as text
%        str2double(get(hObject,'String')) returns contents of c13 as a double


% --- Executes during object creation, after setting all properties.
function c13_CreateFcn(hObject, eventdata, handles)
% hObject    handle to c13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function v11_Callback(hObject, eventdata, handles)
% hObject    handle to v11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v11 as text
%        str2double(get(hObject,'String')) returns contents of v11 as a double

% --- Executes during object creation, after setting all properties.
function v11_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function v12_Callback(hObject, eventdata, handles)
% hObject    handle to v12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v12 as text
%        str2double(get(hObject,'String')) returns contents of v12 as a double


% --- Executes during object creation, after setting all properties.
function v12_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
```

```
    set(hObject,'BackgroundColor','white');
end


function v13_Callback(hObject, eventdata, handles)
% hObject    handle to v13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v13 as text
% str2double(get(hObject,'String')) returns contents of v13 as a double


% --- Executes during object creation, after setting all properties.
function v13_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end


function invalidcheck_Callback(hObject, eventdata, handles)
% hObject    handle to invalidcheck (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of invalidcheck as text
%        str2double(get(hObject,'String')) returns contents of invalidcheck as a double


% --- Executes during object creation, after setting all properties.
function invalidcheck_CreateFcn(hObject, eventdata, handles)
% hObject    handle to invalidcheck (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function c14_Callback(hObject, eventdata, handles)
% hObject    handle to c14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of c14 as text
%        str2double(get(hObject,'String')) returns contents of c14 as a double

% --- Executes during object creation, after setting all properties.
function c14_CreateFcn(hObject, eventdata, handles)
% hObject    handle to c14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function v14_Callback(hObject, eventdata, handles)
% hObject    handle to v14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v14 as text
%        str2double(get(hObject,'String')) returns contents of v14 as a double
```

```
% --- Executes during object creation, after setting all properties.
function v14_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function amplitude_Callback(hObject, eventdata, handles)
% hObject    handle to amplitude (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of amplitude as text
%        str2double(get(hObject,'String')) returns contents of amplitude as a double


% --- Executes during object creation, after setting all properties.
function amplitude_CreateFcn(hObject, eventdata, handles)
% hObject    handle to amplitude (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function frequency_Callback(hObject, eventdata, handles)
% hObject    handle to frequency (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of frequency as text
%        str2double(get(hObject,'String')) returns contents of frequency as a double


% --- Executes during object creation, after setting all properties.
function frequency_CreateFcn(hObject, eventdata, handles)
% hObject    handle to frequency (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function upper_Callback(hObject, eventdata, handles)
% hObject    handle to upper (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of upper as text
%        str2double(get(hObject,'String')) returns contents of upper as a double

% --- Executes during object creation, after setting all properties.
function upper_CreateFcn(hObject, eventdata, handles)
% hObject    handle to upper (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function lower_Callback(hObject, eventdata, handles)
% hObject    handle to lower (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of lower as text
%        str2double(get(hObject,'String')) returns contents of lower as a double



% --- Executes during object creation, after setting all properties.
function lower_CreateFcn(hObject, eventdata, handles)
% hObject    handle to lower (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

“Sinusoidal_Clamper_Design.m” –

```
function [t,vin,vout,img,c1,v1]=Sinusoidal_Clamper_Design(vamp,f,vup,vlow)
    c1=10^-6;
    R=10^6;
    T=1/f;
    while 5*R*c1<=T/2
        c1=c1*10;
    end

    if vlow==0 && vup==2*vamp
        [t,vin,vout,img,v1]=positive_clamper(vamp,f,vup,vlow);
    elseif vlow==-2*vamp && vup==0
        [t,vin,vout,img,v1]=negative_clamper(vamp,f,vup,vlow);
    elseif vlow>0
        [t,vin,vout,img,v1]=positive_bias_positive_clamper(vamp,f,vup,vlow);
```

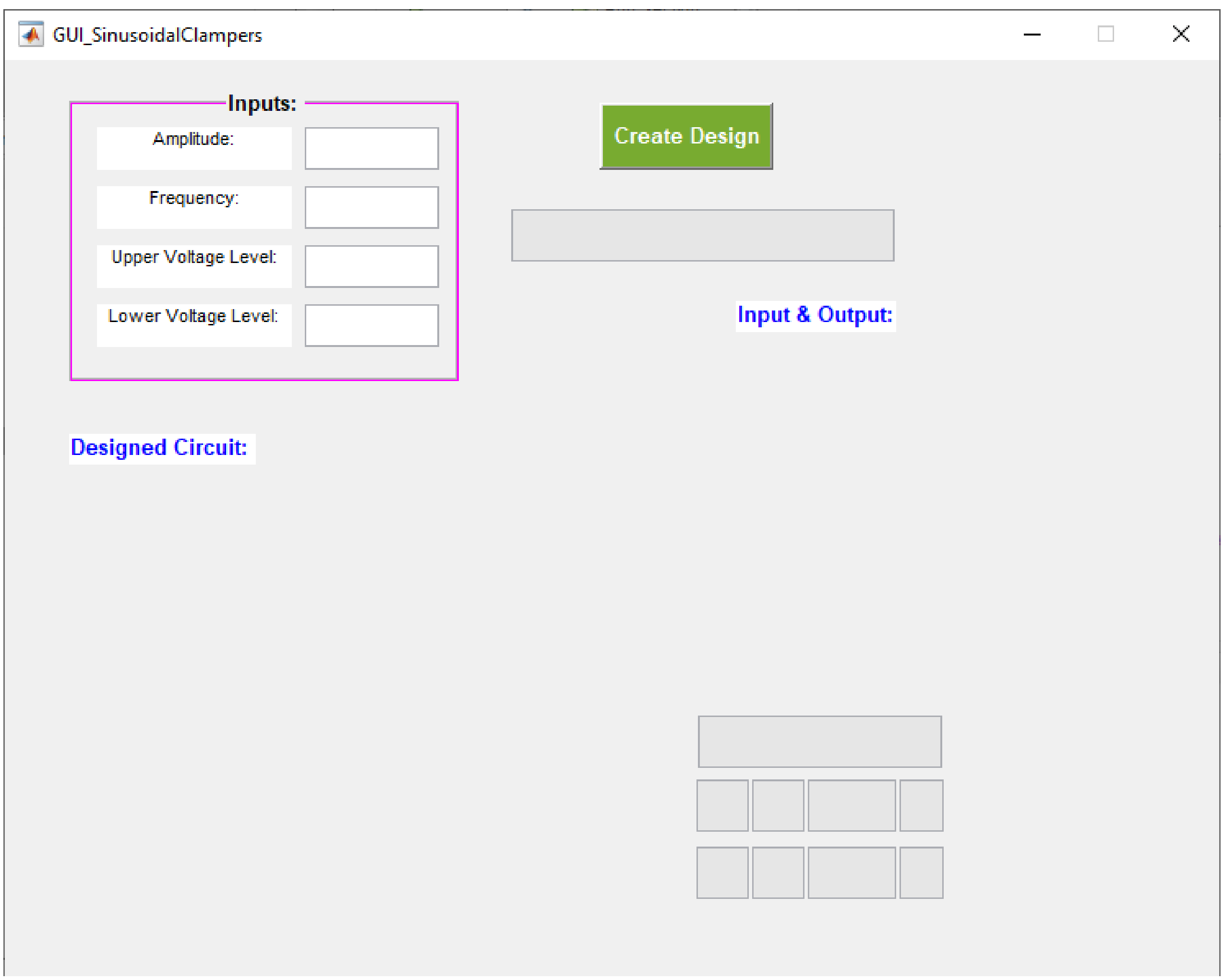
```
    elseif vlow<0 && vlow>(-vamp)
        [t,vin,vout,img,v1]=negative_bias_positive_clamper(vamp,f,vup,vlow);
    elseif vup>0 && vup<=vamp
        [t,vin,vout,img,v1]=positive_bias_negative_clamper(vamp,f,vup,vlow);
    elseif vup<0
        [t,vin,vout,img,v1]=negative_bias_negative_clamper(vamp,f,vup,vlow);
    end
end

function [t,vin,vout,img,v1]=positive_clamper(vamp,f,vup,vlow)
    v1=0;
    T=1/f;
    t=linspace(0,3*T,1200);
    vin=vamp.*sin((2*pi*f)*t);
    vout(1:1200)=0;
    i=1;
    while i<=1200
        if i<=200
            vout(i)=vin(i);
        elseif (i>200) && (i<=300)
            vout(i)=0;
        else
            vout(i)=vamp+vin(i);
        end
        i=i+1;
    end
    img=imread('SinusoidalClamper1.PNG');
end

function [t,vin,vout,img,v1]=negative_clamper(vamp,f,vup,vlow)
    v1=0;
    T=1/f;
    t=linspace(0,3*T,1200);
    vin=vamp.*sin((2*pi*f)*t);
    vout(1:1200)=0;
    i=1;
    while i<=1200
        if i<=100
            vout(i)=0;
        else
```

```
      vout(i)=-vamp+vin(i);
    end
    i=i+1;
  end
  img=imread('SinusoidalClamper2.PNG');
end

function [t,vin,vout,img,v1]=positive_bias_positive_clamper(vamp,f,vup,vlow)
  v1=vlow;
  T=1/f;
  t=linspace(0,3*T,1200);
  vin=vamp.*sin((2*pi*f)*t);
  vout(1:1200)=0;
  i=1;
  while i<=1200
    if i<=200
      vout(i)=vin(i)+v1;
    elseif (i>200) && (i<=300)
      vout(i)=v1;
    else
      vout(i)=vamp+v1+vin(i);
    end
    i=i+1;
  end
  img=imread('SinusoidalClamper3.PNG');
end

function [t,vin,vout,img,v1]=negative_bias_positive_clamper(vamp,f,vup,vlow)
  v1=-vlow;
  T=1/f;
  t=linspace(0,3*T,1200);
  vin=vamp.*sin((2*pi*f)*t);
  vout(1:1200)=0;
  i=1;
  while i<=1200
    if i<=200
      vout(i)=vin(i);
    elseif (i>200) && (i<=300)
      if vin(i)>=-v1
        vout(i)=vin(i);
```

```
        else
            vout(i)=-v1;
        end
    else
        vout(i)=vin(i)+vamp-v1;
    end
    i=i+1;
  end
  img=imread('SinusoidalClamper4.PNG');
end

function [t,vin,vout,img,v1]=positive_bias_negative_clamper(vamp,f,vup,vlow)
  v1=vup;
  T=1/f;
  t=linspace(0,3*T,1200);
  vin=vamp.*sin((2*pi*f)*t);
  vout(1:1200)=0;
  i=1;
  while i<=1200
    if i<=100
        if vin(i)<=v1
            vout(i)=vin(i);
        else
            vout(i)=v1;
        end
    else
        vout(i)=vin(i)-vamp+v1;
    end
    i=i+1;
  end
  img=imread('SinusoidalClamper5.PNG');
end

function [t,vin,vout,img,v1]=negative_bias_negative_clamper(vamp,f,vup,vlow)
  v1=-vup;
  T=1/f;
  t=linspace(0,3*T,1200);
  vin=vamp.*sin((2*pi*f)*t);
  vout(1:1200)=0;
  i=1;
```

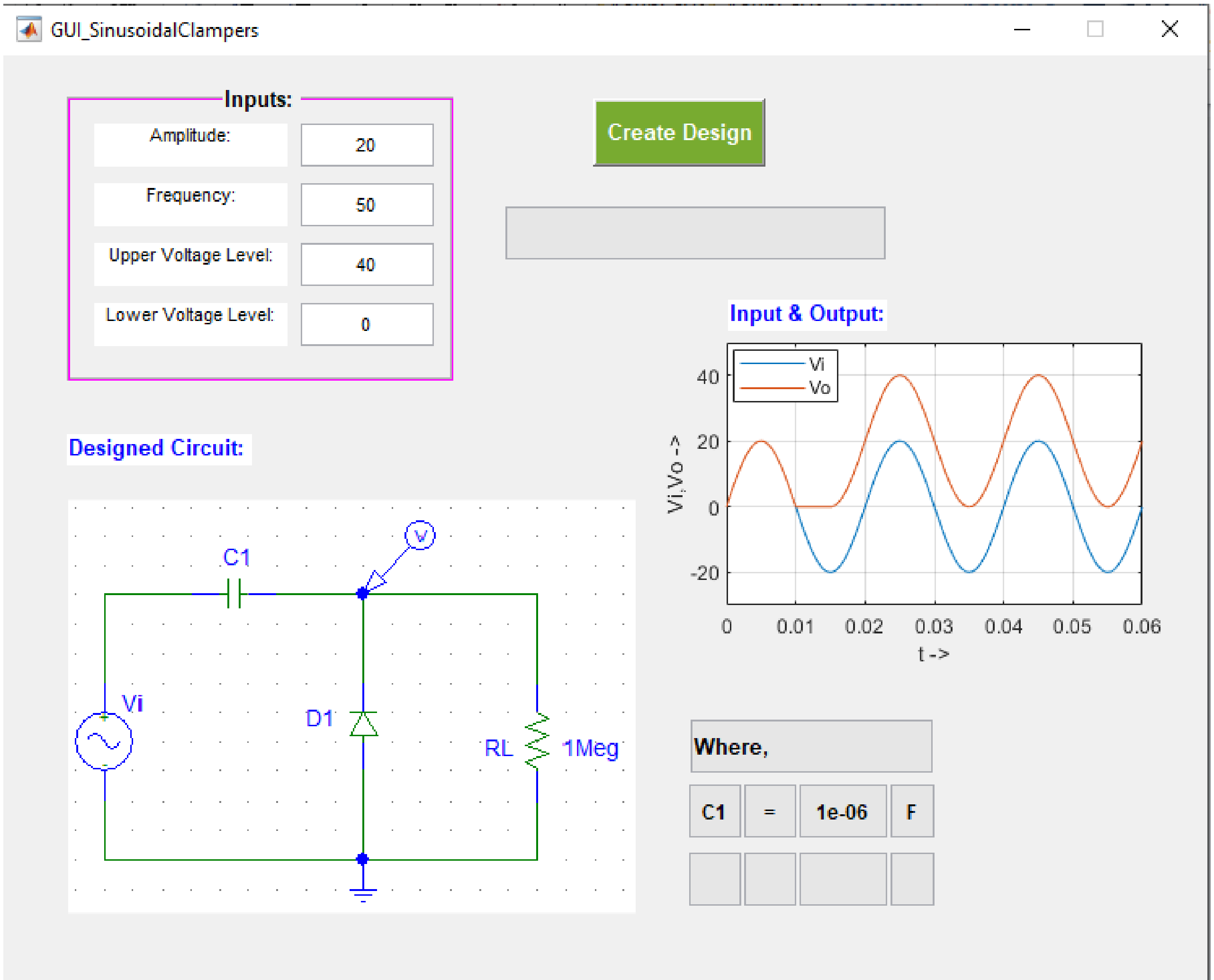
```
  while i<=1200
     if i<=100
        vout(i)=-v1;
     else
        vout(i)=vin(i)-vamp-v1;
     end
     i=i+1;
  end
  img=imread('SinusoidalClamper6.PNG');
end
```

Corresponding Window:

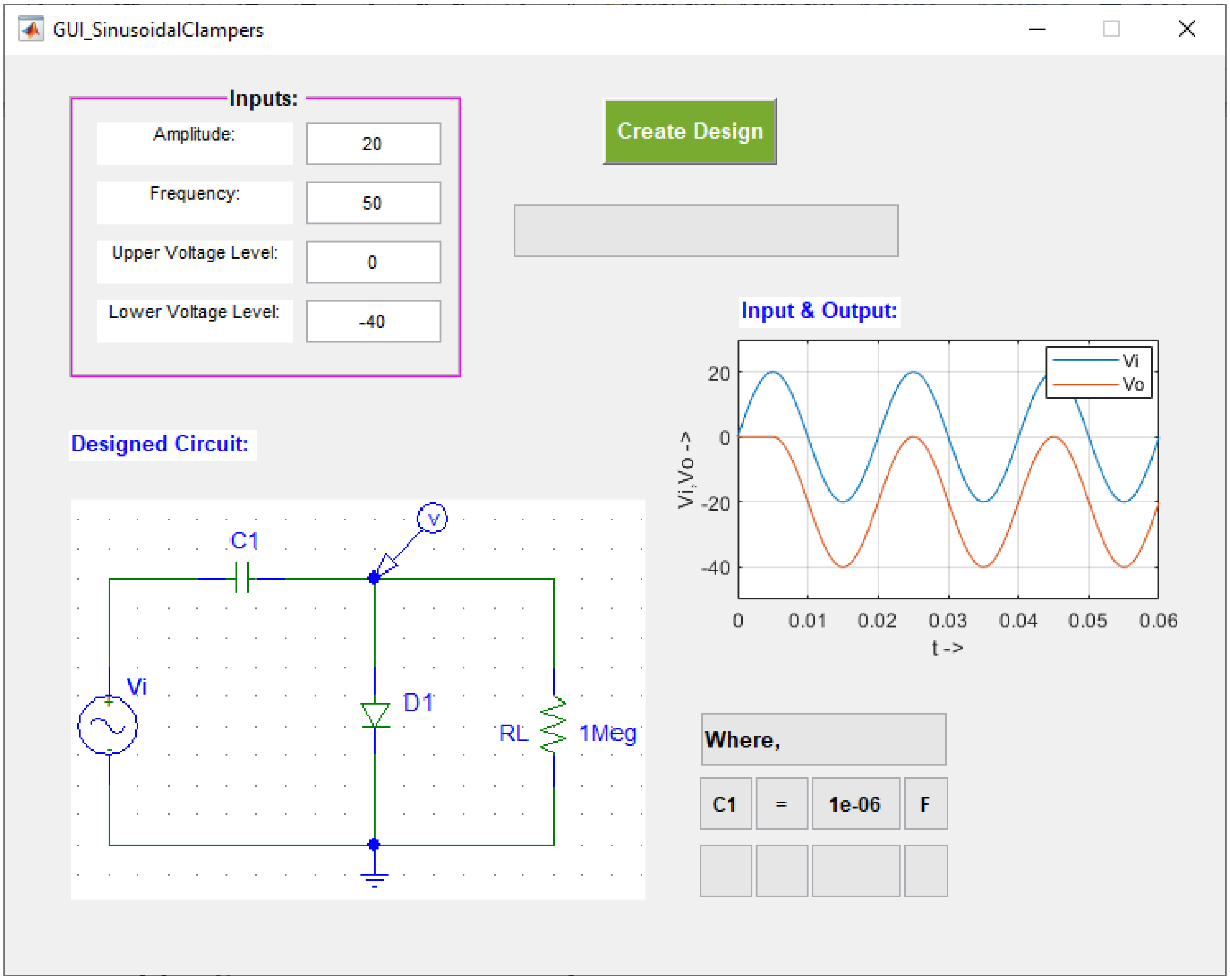


Sample Outputs:

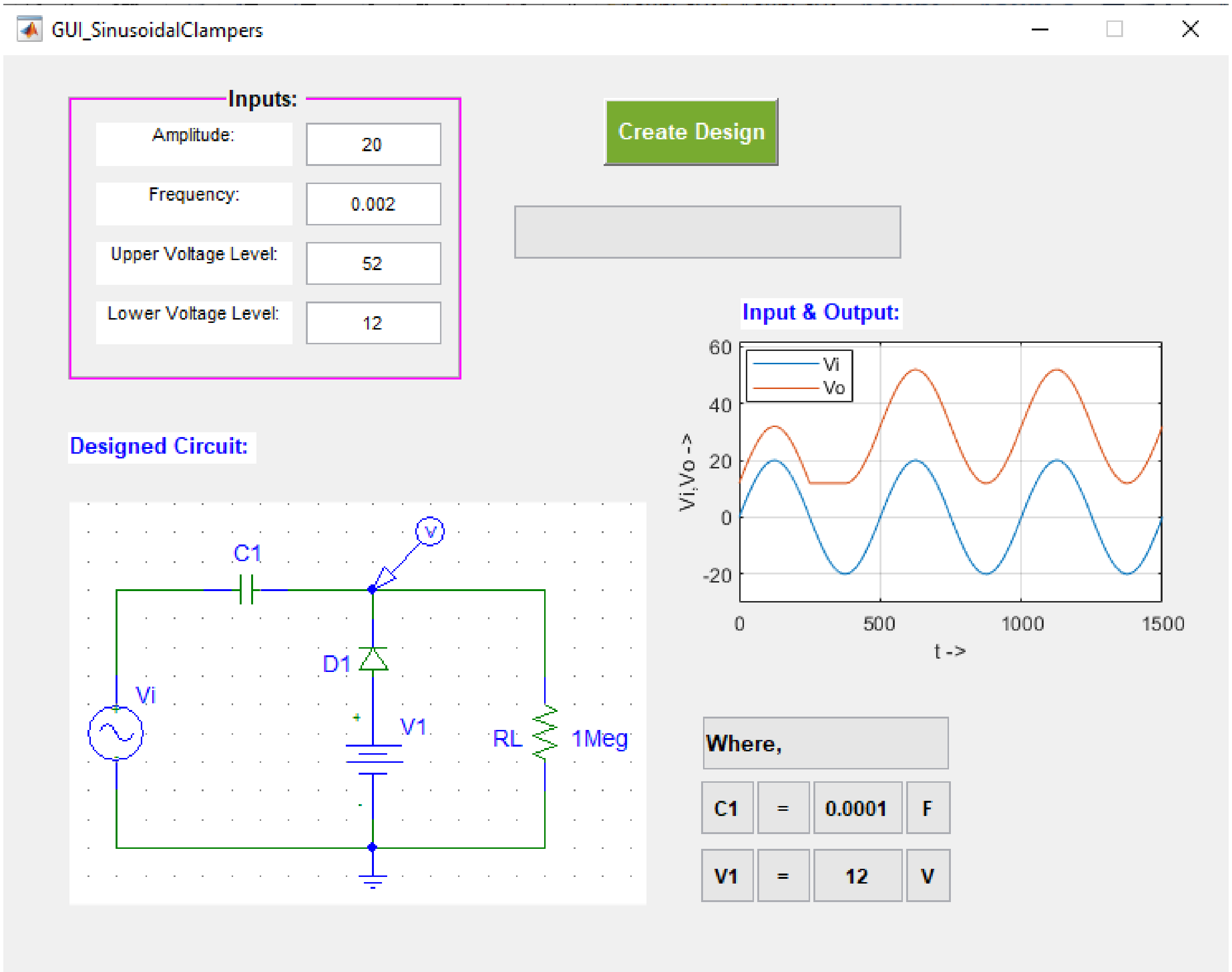
1. Positive Clamper :



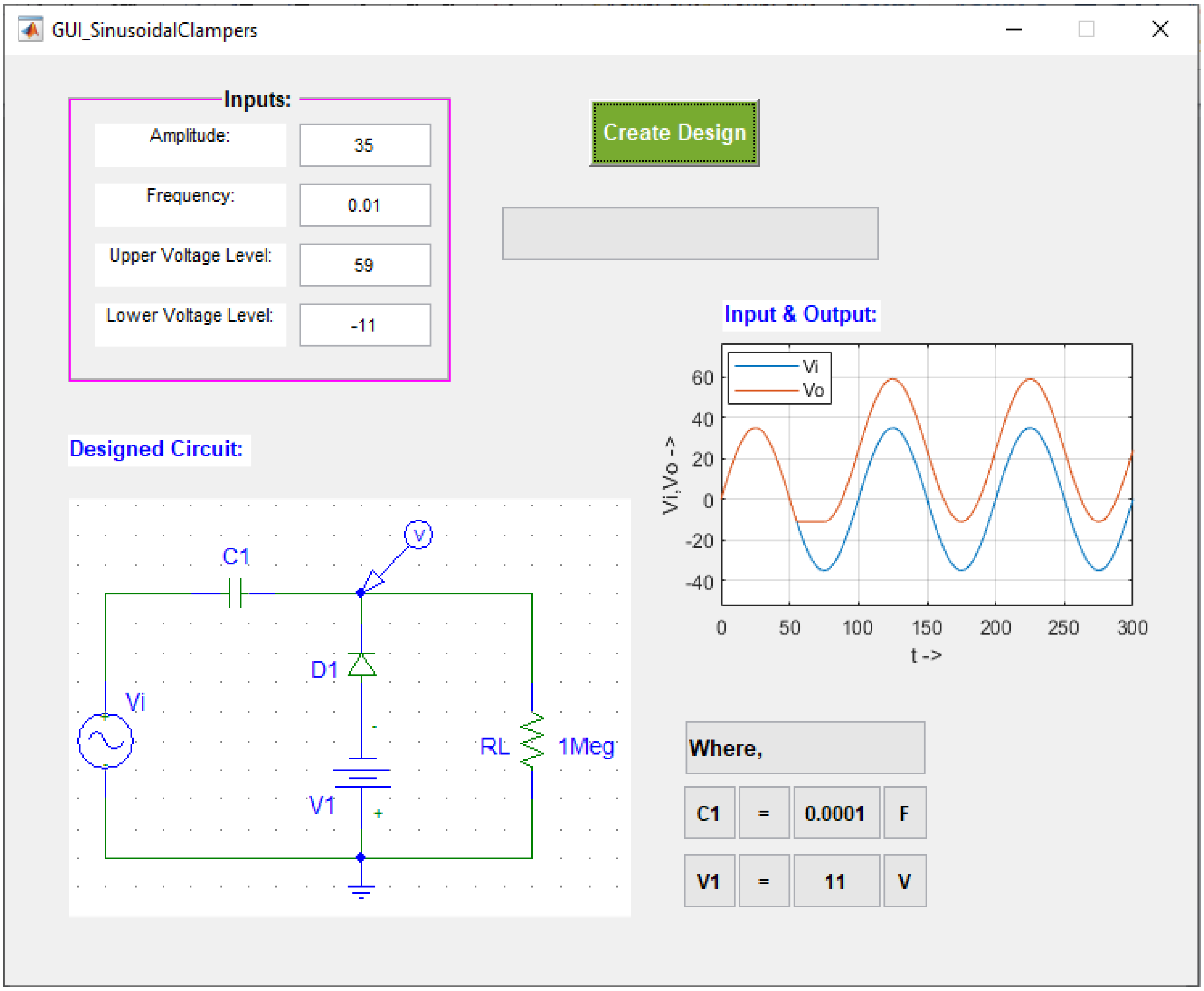
2. Negative Clamper :



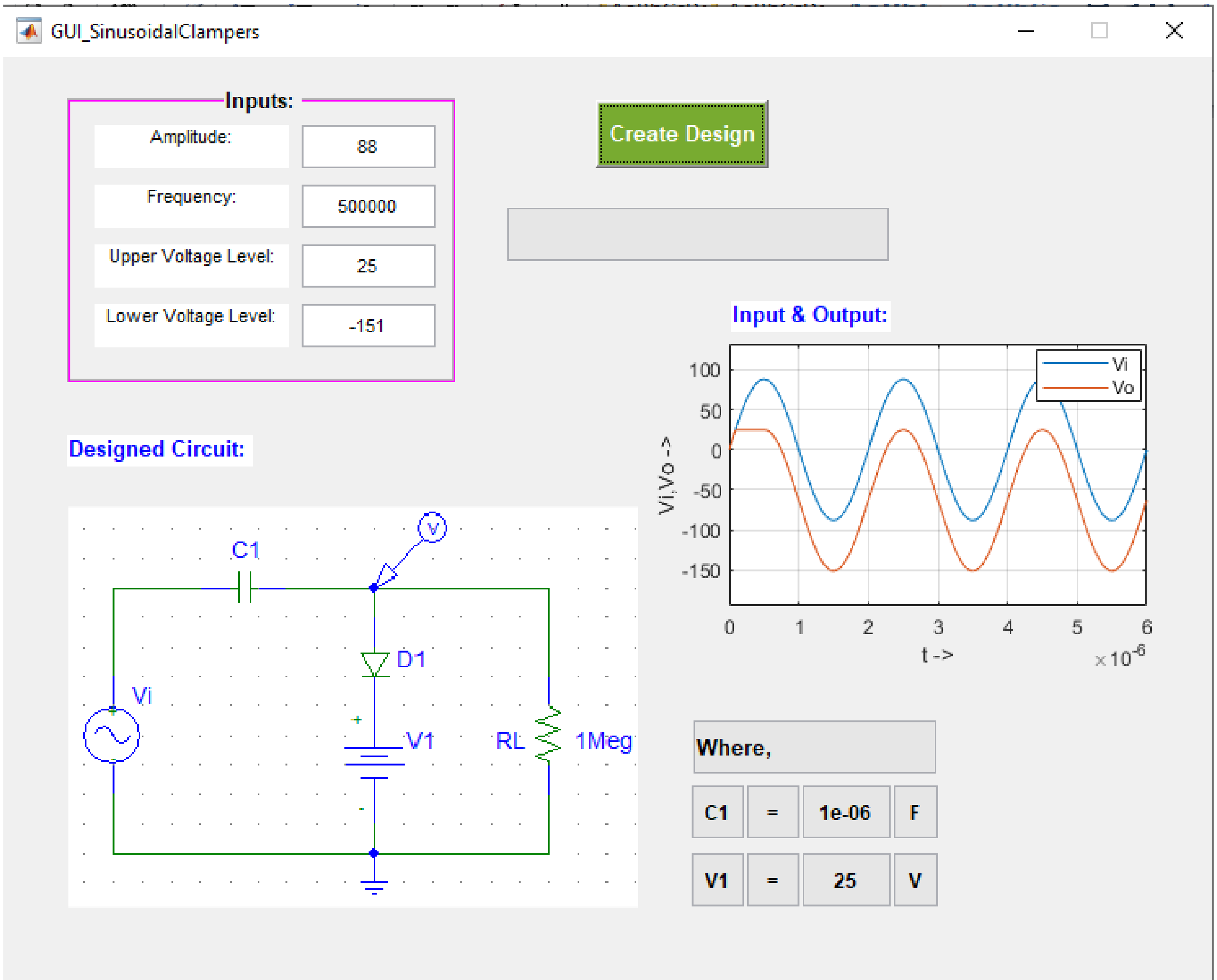
3. **Positive Bias - Positive Clamper :**



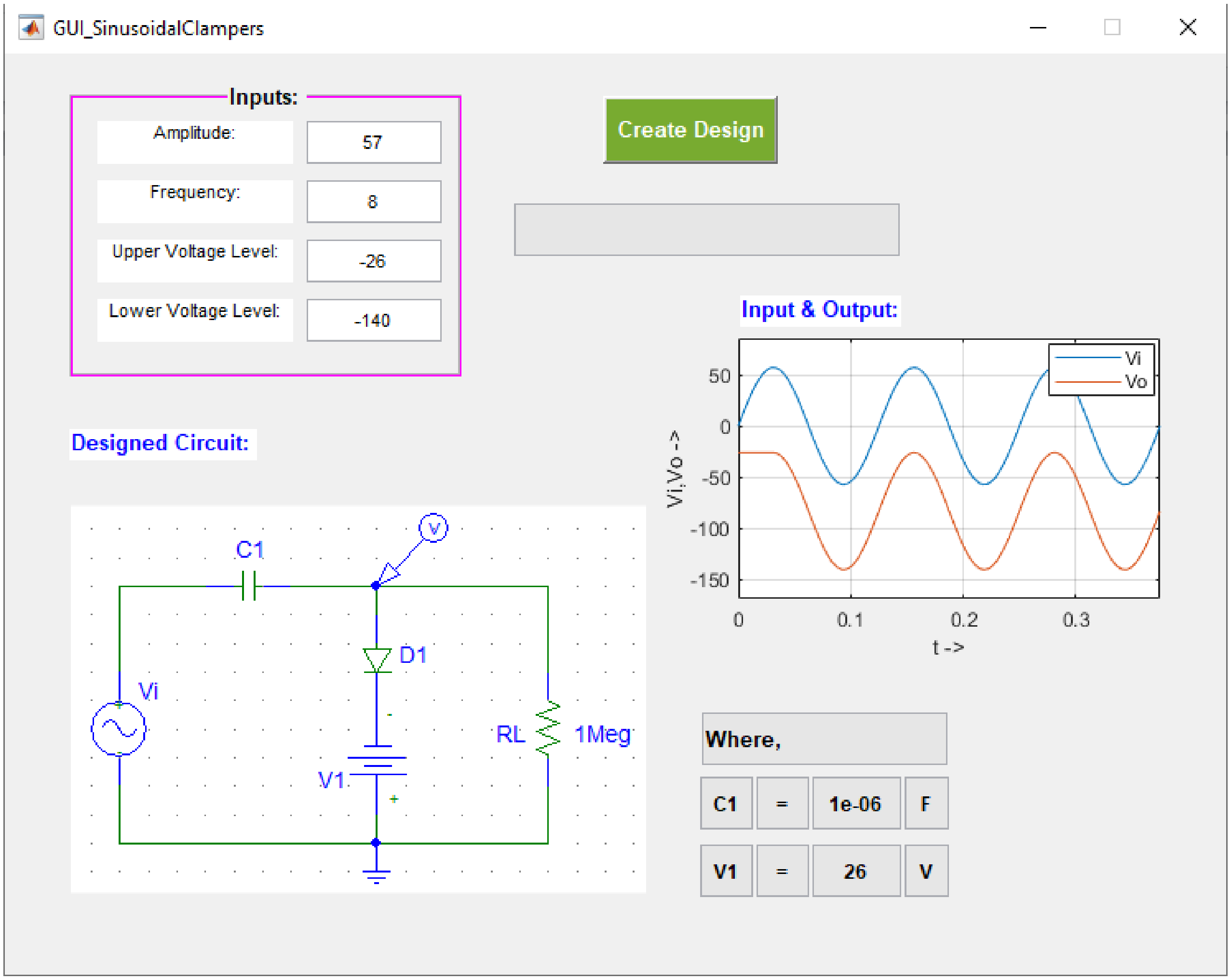
4. Negative Bias – Positive Clamper:



5. Positive Bias - Negative Clamper :

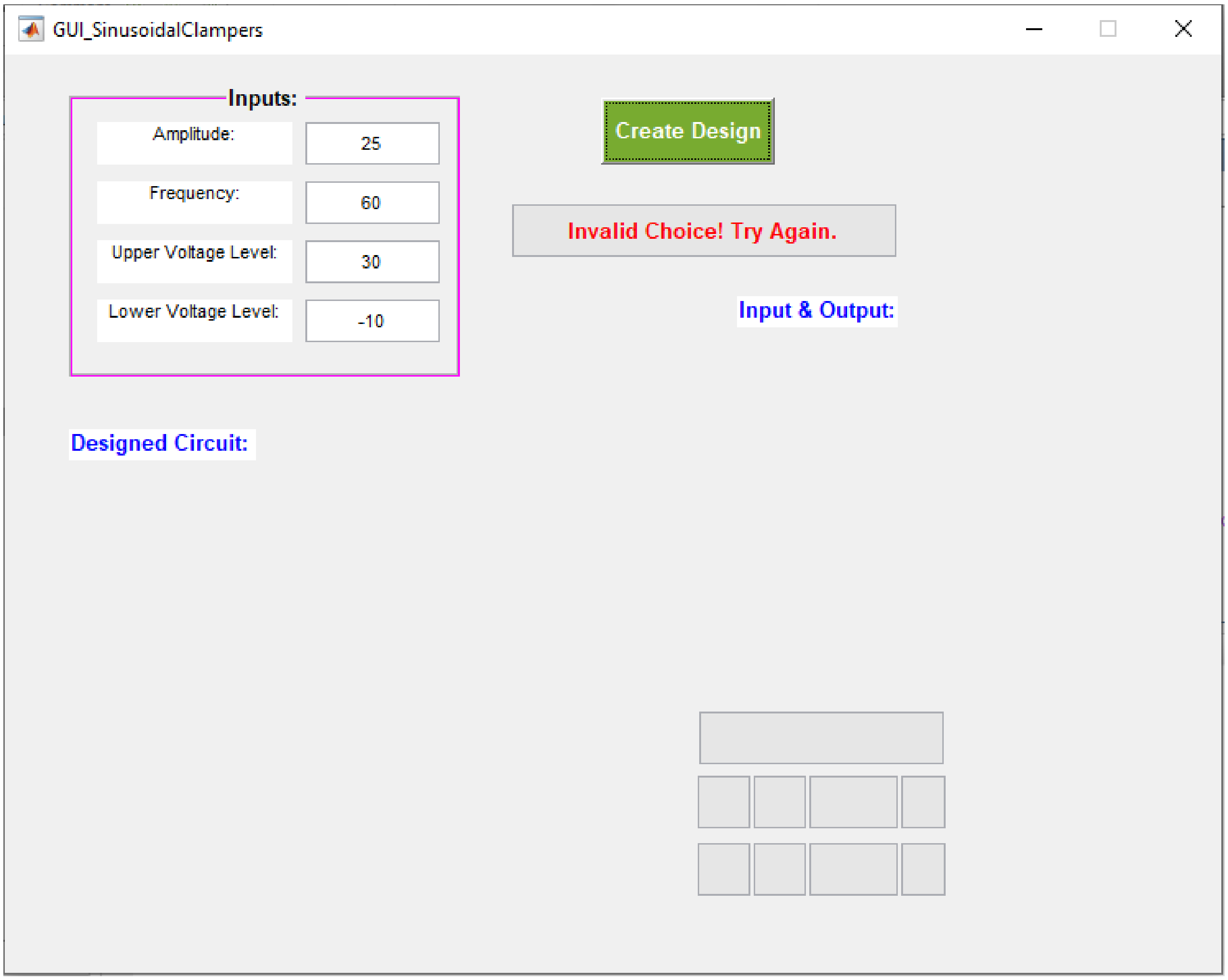


6. Negative Bias - Negative Clamper :

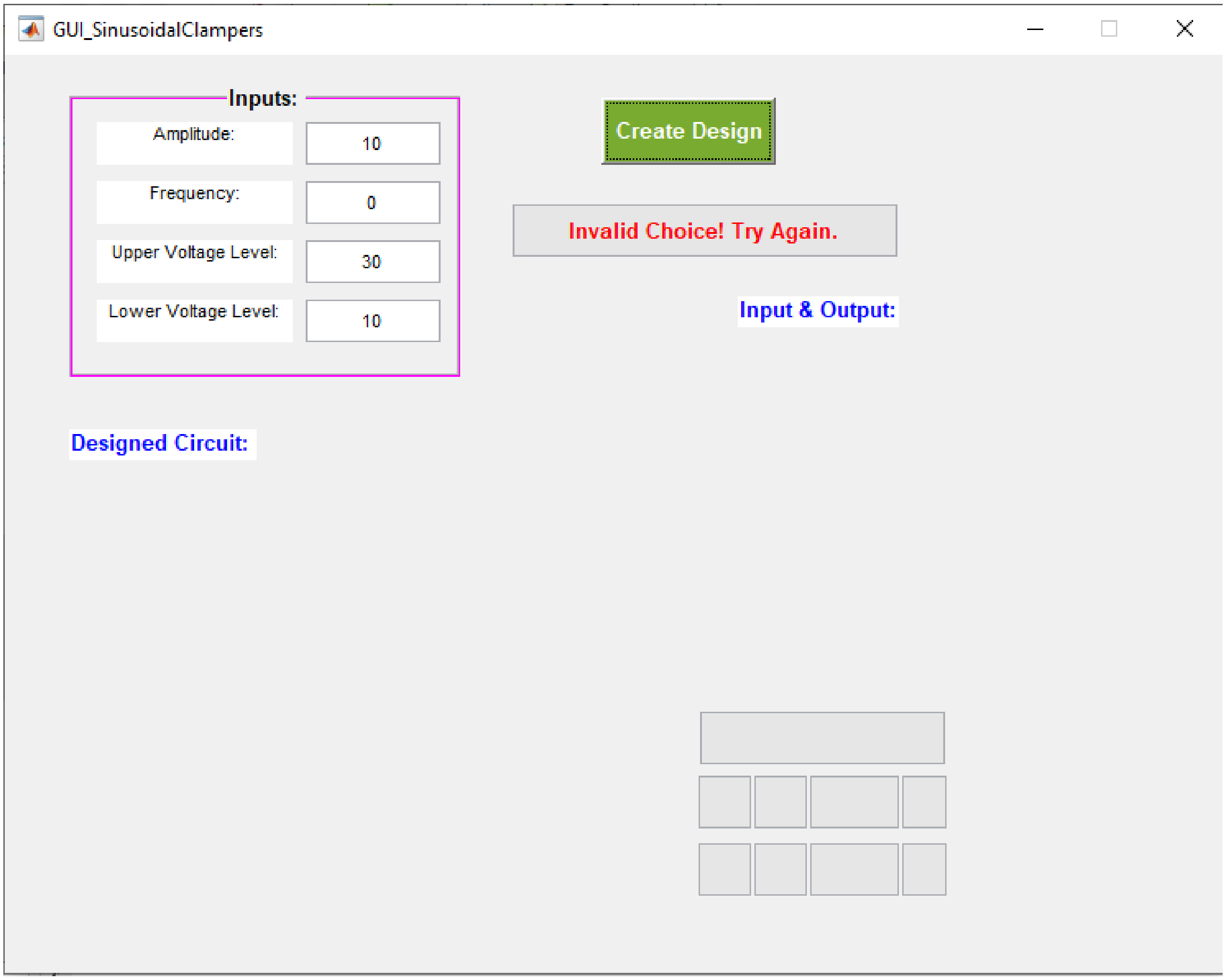


7. Invalid Cases:

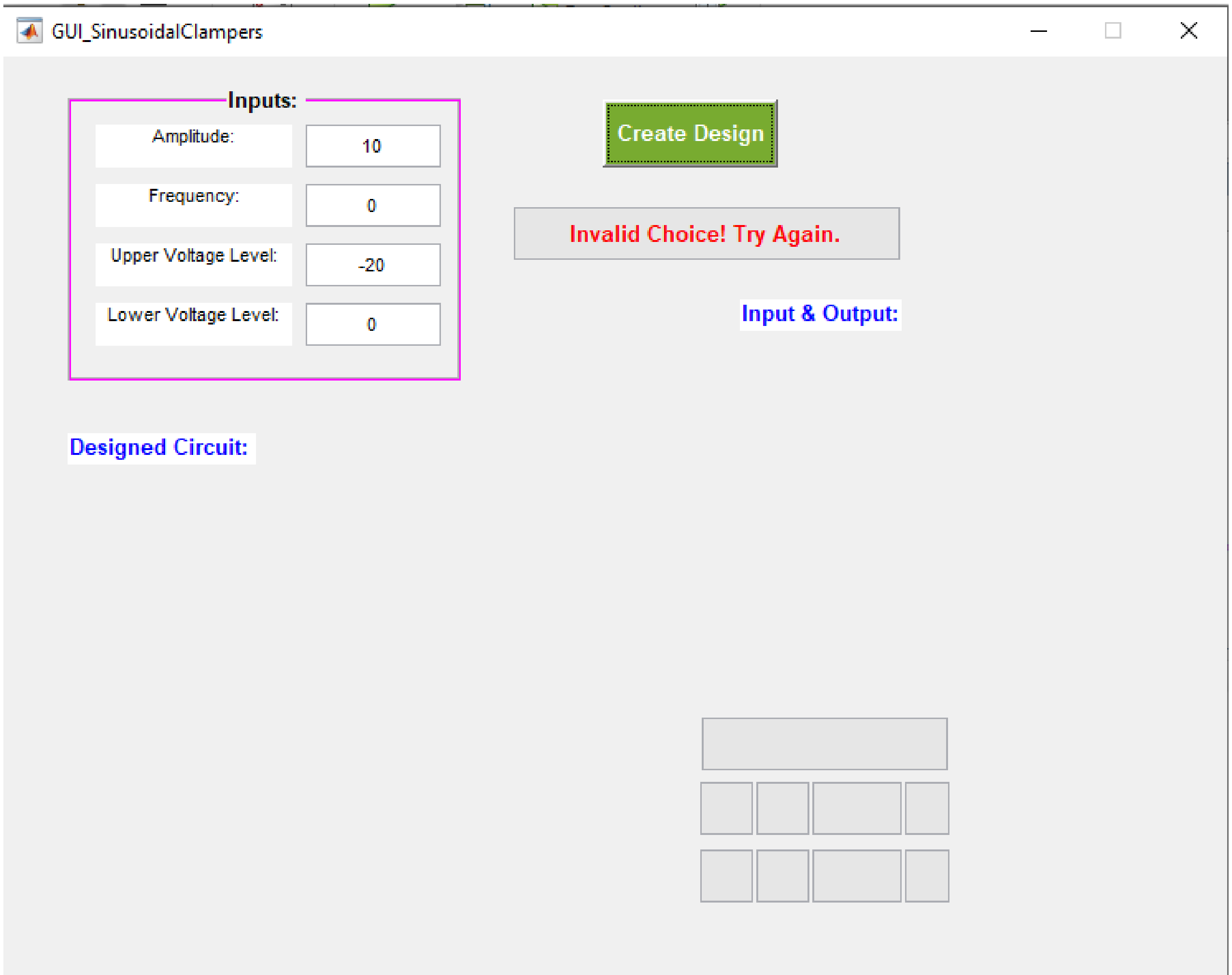
i. (Upper Voltage Level-Lower Voltage Level) $\neq$ (2 $\times$ Amplitude) :



ii. Zero/ Negative Frequency:



iii. Lower Voltage Level is given greater than Upper Voltage Level:



<u>N. B:</u> Our program is able to design the circuit for any other kind of inputs as well. User is appreciated to check for more as he desires.

(ii) **Square Clampers:**

- Now, let us choose "Square Wave" from "GUI_Clamper" window.
- "GUI_SquareClampers.m" interacts with the user and calls "SquareWave_Clamper_Design.m" that contains all necessary functions and codes to create the design and sends data back.

Codes:

"GUI_SquareClampers.m" –

```
function varargout = GUI_SquareClampers(varargin)
% GUI_SQUARECLAMPERS MATLAB code for GUI_SquareClampers.fig
% GUI_SQUARECLAMPERS, by itself, creates a new GUI_SQUARECLAMPERS or
raises the existing
% singleton*.
%
% H = GUI_SQUARECLAMPERS returns the handle to a new
GUI_SQUARECLAMPERS or the handle to
% the existing singleton*.
%
% GUI_SQUARECLAMPERS('CALLBACK',hObject,eventData,handles,...) calls the local
% function named CALLBACK in GUI_SQUARECLAMPERS.M with the given input
arguments.
%
% GUI_SQUARECLAMPERS('Property','Value',...) creates a new
GUI_SQUARECLAMPERS or raises the
% existing singleton*.  Starting from the left, property value pairs are
% applied to the GUI before GUI_SquareClampers_OpeningFcn gets called.  An
% unrecognized property name or invalid value makes property application
% stop.  All inputs are passed to GUI_SquareClampers_OpeningFcn via varargin.
%
% *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%  instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help GUI_SquareClampers

% Last Modified by GUIDE v2.5 19-Jul-2021 03:58:11
```

```
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @GUI_SquareClampers_OpeningFcn, ...
                   'gui_OutputFcn',  @GUI_SquareClampers_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before GUI_SquareClampers is made visible.
function GUI_SquareClampers_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to GUI_SquareClampers (see VARARGIN)

% Choose default command line output for GUI_SquareClampers
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes GUI_SquareClampers wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = GUI_SquareClampers_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in design.
function design_Callback(hObject, eventdata, handles)
% hObject    handle to design (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.invalidcheck,'String','');
set(handles.properties,'String','');
set(handles.c11,'String','');
set(handles.c12,'String','');
set(handles.c13,'String','');
set(handles.c14,'String','');
set(handles.v11,'String','');
set(handles.v12,'String','');
set(handles.v13,'String','');
set(handles.v14,'String','');
axes(handles.diagram);
cla;
axes(handles.plots);
cla;

amp=str2double(get(handles.amplitude,'String'));
f=str2double(get(handles.frequency,'String'));
h=str2double(get(handles.upper,'String'));
l=str2double(get(handles.lower,'String'));

if  f<=0 || (h-l)~=(2*amp)
    set(handles.invalidcheck,'String','Invalid Choice! Try Again.');
else
    [t,vin,vout,img,c1,v1]=SquareWave_Clamper_Design(amp,f,h,l);
    axes(handles.plots);
    cla;
    plot(t,vin);
    hold on;
    plot(t,vout);
```

```
    if max(vout)>=max(vin)
        axis([0,t(end),(min(vin)-amp/2),(max(vout)+amp/2)]);
    else
        axis([0,t(end),(min(vout)-amp/2),(max(vin)+amp/2)]);
    end
    grid on;
    legend('Vi','Vo');
    xlabel('t ->');
    ylabel('Vi,Vo ->');
    axes(handles.diagram);
    cla;
    imshow(img);
    if  v1==0
        set(handles.properties,'String','Where,');
        set(handles.c11,'String','C1');
        set(handles.c12,'String','=');
        set(handles.c13,'String',num2str(c1));
        set(handles.c14,'String','F');
    else
        set(handles.properties,'String','Where,');
        set(handles.c11,'String','C1');
        set(handles.c12,'String','=');
        set(handles.c13,'String',num2str(c1));
        set(handles.c14,'String','F');
        set(handles.v11,'String','V1');
        set(handles.v12,'String','=');
        set(handles.v13,'String',num2str(v1));
        set(handles.v14,'String','V');
    end
end

function properties_Callback(hObject, eventdata, handles)
% hObject    handle to properties (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of properties as text
%        str2double(get(hObject,'String')) returns contents of properties as a double

% --- Executes during object creation, after setting all properties.
```

```
function properties_CreateFcn(hObject, eventdata, handles)
% hObject    handle to properties (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function c11_Callback(hObject, eventdata, handles)
% hObject    handle to c11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of c11 as text
%        str2double(get(hObject,'String')) returns contents of c11 as a double


% --- Executes during object creation, after setting all properties.
function c11_CreateFcn(hObject, eventdata, handles)
% hObject    handle to c11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function c12_Callback(hObject, eventdata, handles)
% hObject    handle to c12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of c12 as text
%        str2double(get(hObject,'String')) returns contents of c12 as a double

% --- Executes during object creation, after setting all properties.
function c12_CreateFcn(hObject, eventdata, handles)
% hObject    handle to c12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function c13_Callback(hObject, eventdata, handles)
% hObject    handle to c13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of c13 as text
%        str2double(get(hObject,'String')) returns contents of c13 as a double


% --- Executes during object creation, after setting all properties.
function c13_CreateFcn(hObject, eventdata, handles)
% hObject    handle to c13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function v11_Callback(hObject, eventdata, handles)
% hObject    handle to v11 (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v11 as text
%        str2double(get(hObject,'String')) returns contents of v11 as a double


% --- Executes during object creation, after setting all properties.
function v11_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function v12_Callback(hObject, eventdata, handles)
% hObject    handle to v12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v12 as text
%        str2double(get(hObject,'String')) returns contents of v12 as a double


% --- Executes during object creation, after setting all properties.
function v12_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
```

```
    set(hObject,'BackgroundColor','white');
end


function v13_Callback(hObject, eventdata, handles)
% hObject    handle to v13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v13 as text
%        str2double(get(hObject,'String')) returns contents of v13 as a double


% --- Executes during object creation, after setting all properties.
function v13_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function invalidcheck_Callback(hObject, eventdata, handles)
% hObject    handle to invalidcheck (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of invalidcheck as text
%        str2double(get(hObject,'String')) returns contents of invalidcheck as a double


% --- Executes during object creation, after setting all properties.
function invalidcheck_CreateFcn(hObject, eventdata, handles)
% hObject    handle to invalidcheck (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function c14_Callback(hObject, eventdata, handles)
% hObject    handle to c14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of c14 as text
% str2double(get(hObject,'String')) returns contents of c14 as a double


% --- Executes during object creation, after setting all properties.
function c14_CreateFcn(hObject, eventdata, handles)
% hObject    handle to c14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function v14_Callback(hObject, eventdata, handles)
% hObject    handle to v14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v14 as text
%        str2double(get(hObject,'String')) returns contents of v14 as a double
% --- Executes during object creation, after setting all properties.
```

```
function v14_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function amplitude_Callback(hObject, eventdata, handles)
% hObject    handle to amplitude (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of amplitude as text
%        str2double(get(hObject,'String')) returns contents of amplitude as a double

% --- Executes during object creation, after setting all properties.
function amplitude_CreateFcn(hObject, eventdata, handles)
% hObject    handle to amplitude (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function frequency_Callback(hObject, eventdata, handles)
% hObject    handle to frequency (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of frequency as text
%        str2double(get(hObject,'String')) returns contents of frequency as a double
```

```
% --- Executes during object creation, after setting all properties.
function frequency_CreateFcn(hObject, eventdata, handles)
% hObject    handle to frequency (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function upper_Callback(hObject, eventdata, handles)
% hObject    handle to upper (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of upper as text
%        str2double(get(hObject,'String')) returns contents of upper as a double

% --- Executes during object creation, after setting all properties.
function upper_CreateFcn(hObject, eventdata, handles)
% hObject    handle to upper (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function lower_Callback(hObject, eventdata, handles)
% hObject    handle to lower (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of lower as text
```

```
%       str2double(get(hObject,'String')) returns contents of lower as a double

% --- Executes during object creation, after setting all properties.
function lower_CreateFcn(hObject, eventdata, handles)
% hObject    handle to lower (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

“SquareWave_Clamper_Design.m” –

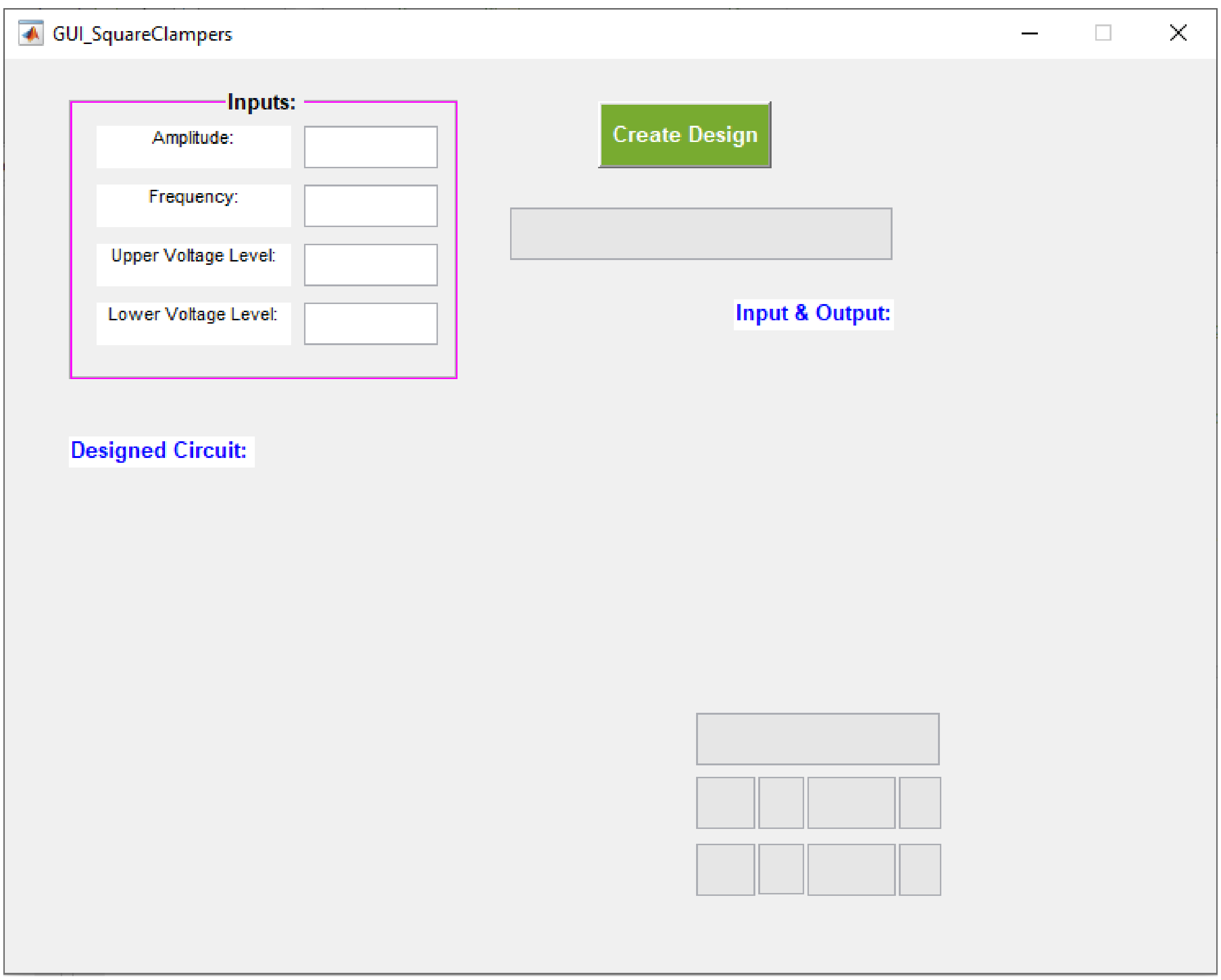
```
function [t,vin,vout,img,c1,v1]=SquareWave_Clamper_Design(vamp,f,vup,vlow)
    c1=10^-6;
    R=10^6;
    T=1/f;
    while 5*R*c1<=T/2
        c1=c1*10;
    end
    if vlow==0 && vup==2*vamp
        [t,vin,vout,img,v1]=positive_clamper(vamp,f,vup,vlow);
    elseif vlow==-2*vamp && vup==0
        [t,vin,vout,img,v1]=negative_clamper(vamp,f,vup,vlow);
    elseif vlow>0
        [t,vin,vout,img,v1]=positive_bias_positive_clamper(vamp,f,vup,vlow);
    elseif vlow<0 && vlow>(-vamp)
        [t,vin,vout,img,v1]=negative_bias_positive_clamper(vamp,f,vup,vlow);
    elseif vup>0 && vup<=vamp
        [t,vin,vout,img,v1]=positive_bias_negative_clamper(vamp,f,vup,vlow);
    elseif vup<0
        [t,vin,vout,img,v1]=negative_bias_negative_clamper(vamp,f,vup,vlow);
    end
end
```

```
function [t,vin,vout,img,v1]=positive_clamper(vamp,f,vup,vlow)
    v1=0;
    T=1/f;
    t=linspace(0,3*T,1200);
    vin=vamp.*square((2*pi*f)*t);
    vout(1:1200)=0;
    for i=1:1200
        if vin(i)==vamp
            vout(i)=2*vamp;
        elseif vin(i)==-vamp
            vout(i)=0;
        end
    end
    img=imread('SquareClamper1.PNG');
end

function [t,vin,vout,img,v1]=negative_clamper(vamp,f,vup,vlow)
    v1=0;
    T=1/f;
    t=linspace(0,3*T,1200);
    vin=vamp.*square((2*pi*f)*t);
    vout(1:1200)=0;
    for i=1:1200
        if vin(i)==-vamp
            vout(i)=-2*vamp;
        elseif vin(i)==vamp
            vout(i)=0;
        end
    end
    img=imread('SquareClamper2.PNG');
end

function [t,vin,vout,img,v1]=positive_bias_positive_clamper(vamp,f,vup,vlow)
    v1=vlow;
    T=1/f;
    t=linspace(0,3*T,1200);
    vin=vamp.*square((2*pi*f)*t);
    vout(1:1200)=0;
    for i=1:1200
        if vin(i)==vamp
```

```
        vout(i)=2*vamp+v1;
    elseif vin(i)==-vamp
        vout(i)=v1;
    end
  end
  img=imread('SquareClamper3.PNG');
end

function [t,vin,vout,img,v1]=negative_bias_positive_clamper(vamp,f,vup,vlow)
  v1=-vlow;
  T=1/f;
  t=linspace(0,3*T,1200);
  vin=vamp.*square((2*pi*f)*t);
  vout(1:1200)=0;
  for i=1:1200
    if vin(i)==vamp
        vout(i)=2*vamp-v1;
    elseif vin(i)==-vamp
        vout(i)=-v1;
    end
  end
  img=imread('SquareClamper4.PNG');
end

function [t,vin,vout,img,v1]=positive_bias_negative_clamper(vamp,f,vup,vlow)
  v1=vup;
  T=1/f;
  t=linspace(0,3*T,1200);
  vin=vamp.*square((2*pi*f)*t);
  vout(1:1200)=0;
  for i=1:1200
    if vin(i)==vamp
        vout(i)=v1;
    elseif vin(i)==-vamp
        vout(i)=-2*vamp+v1;
    end
  end
  img=imread('SquareClamper5.PNG');
end

function [t,vin,vout,img,v1]=negative_bias_negative_clamper(vamp,f,vup,vlow)
```

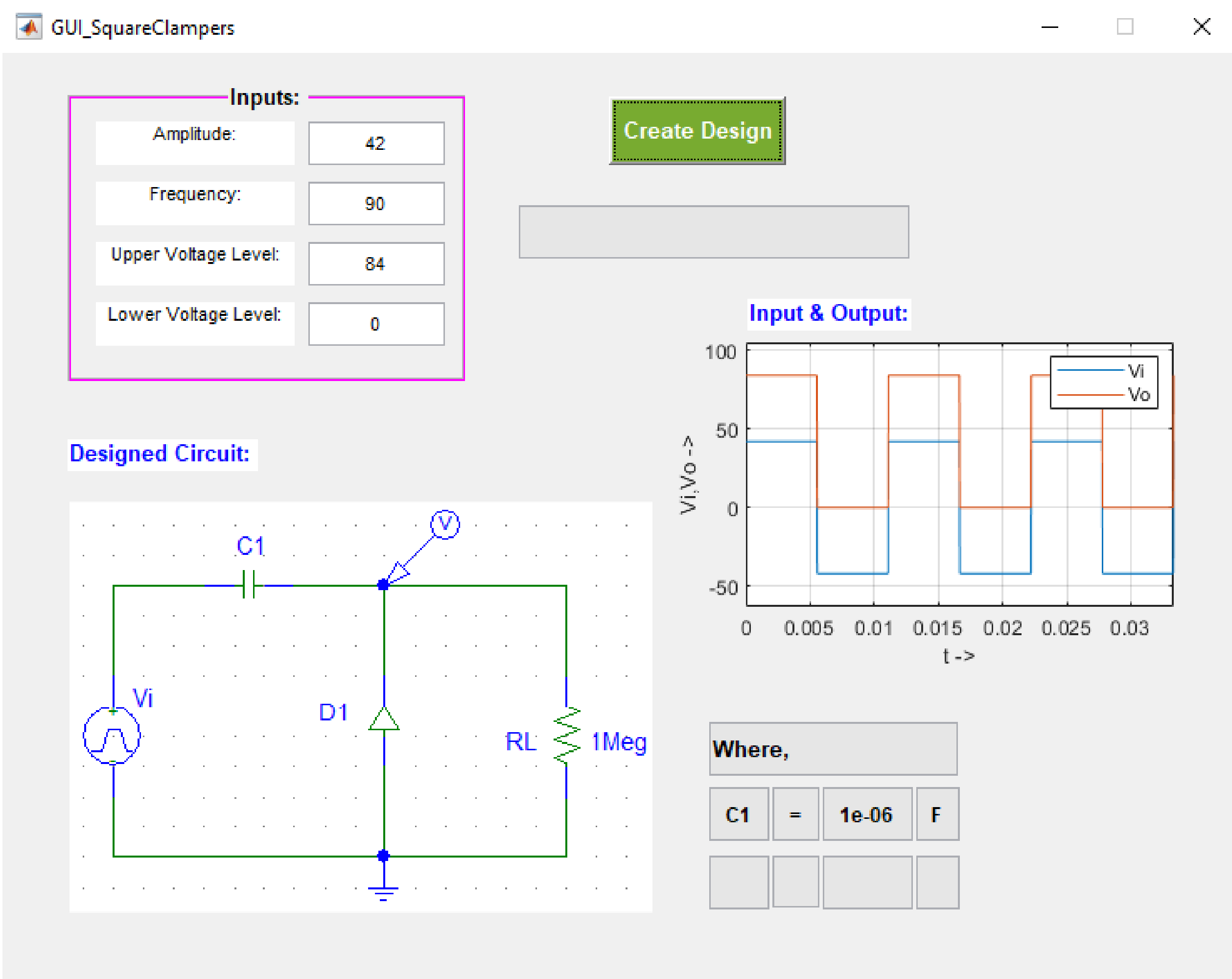
```
  v1=-vup;
  T=1/f;
  t=linspace(0,3*T,1200);
  vin=vamp.*square((2*pi*f)*t);
  vout(1:1200)=0;
  for i=1:1200
     if vin(i)==vamp
        vout(i)=-v1;
     elseif vin(i)==-vamp
        vout(i)=-2*vamp-v1;
     end
  end
  img=imread('SquareClamper6.PNG');
end
```

Corresponding Window:

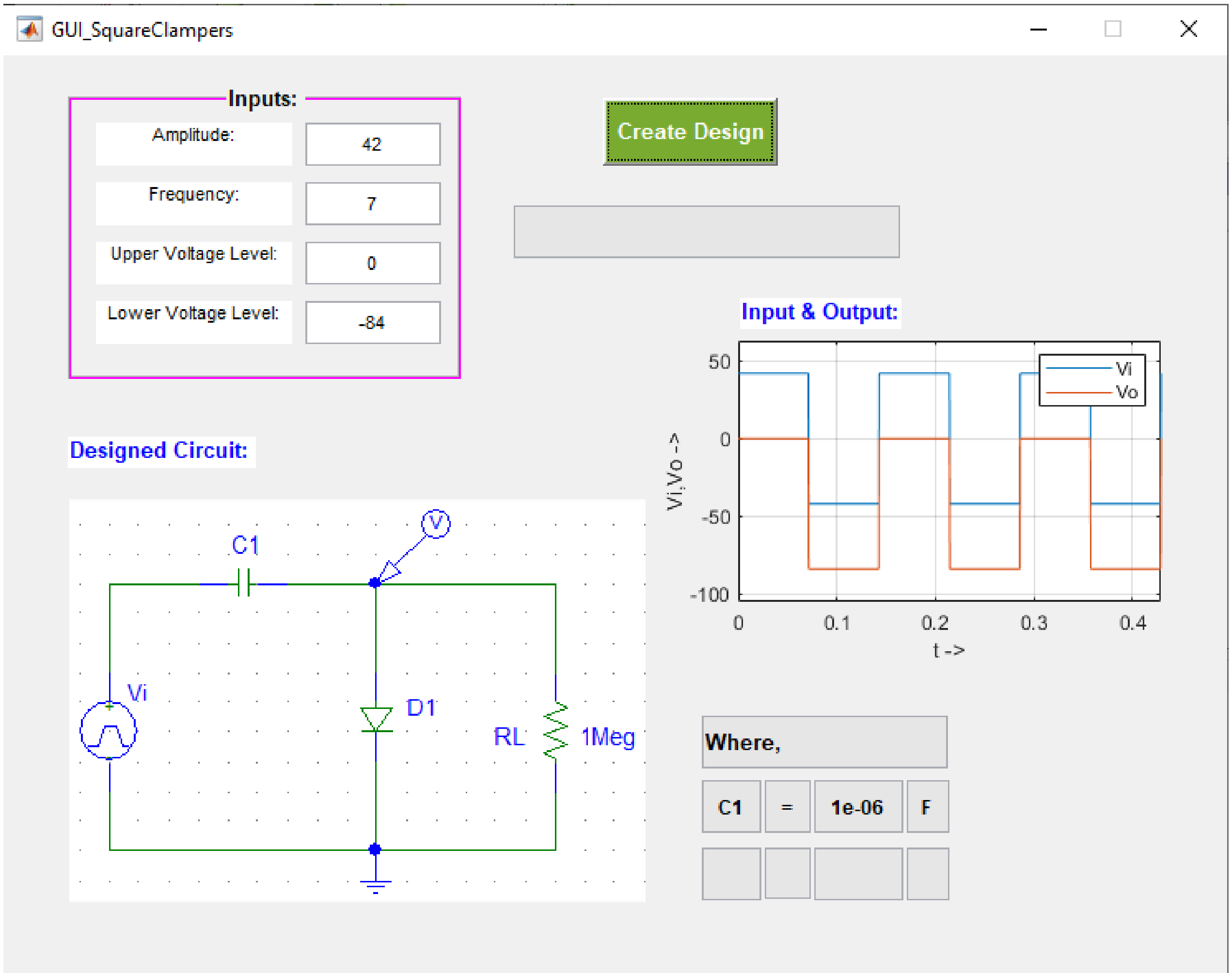


Sample Outputs:

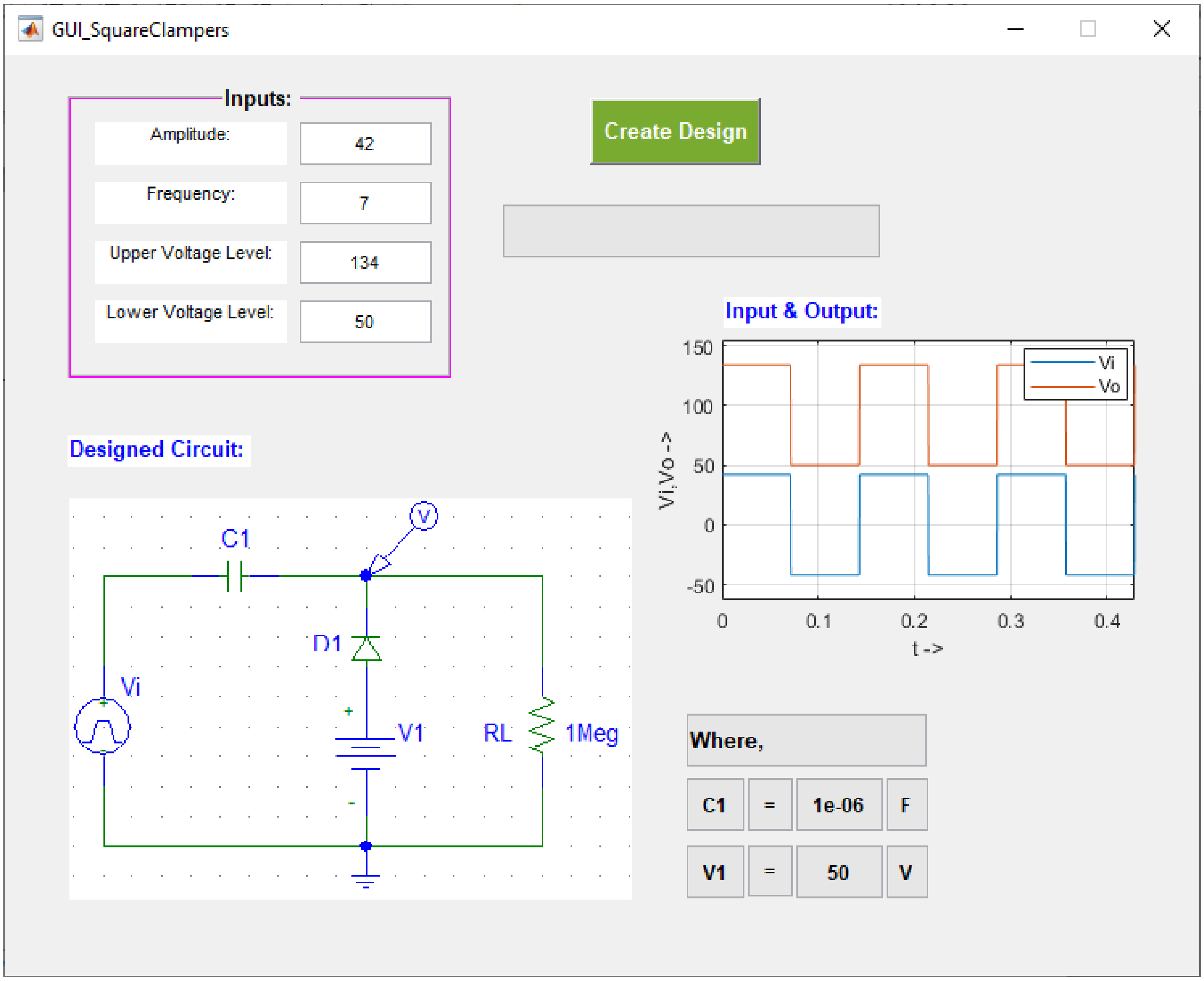
1. Positive Clamper :



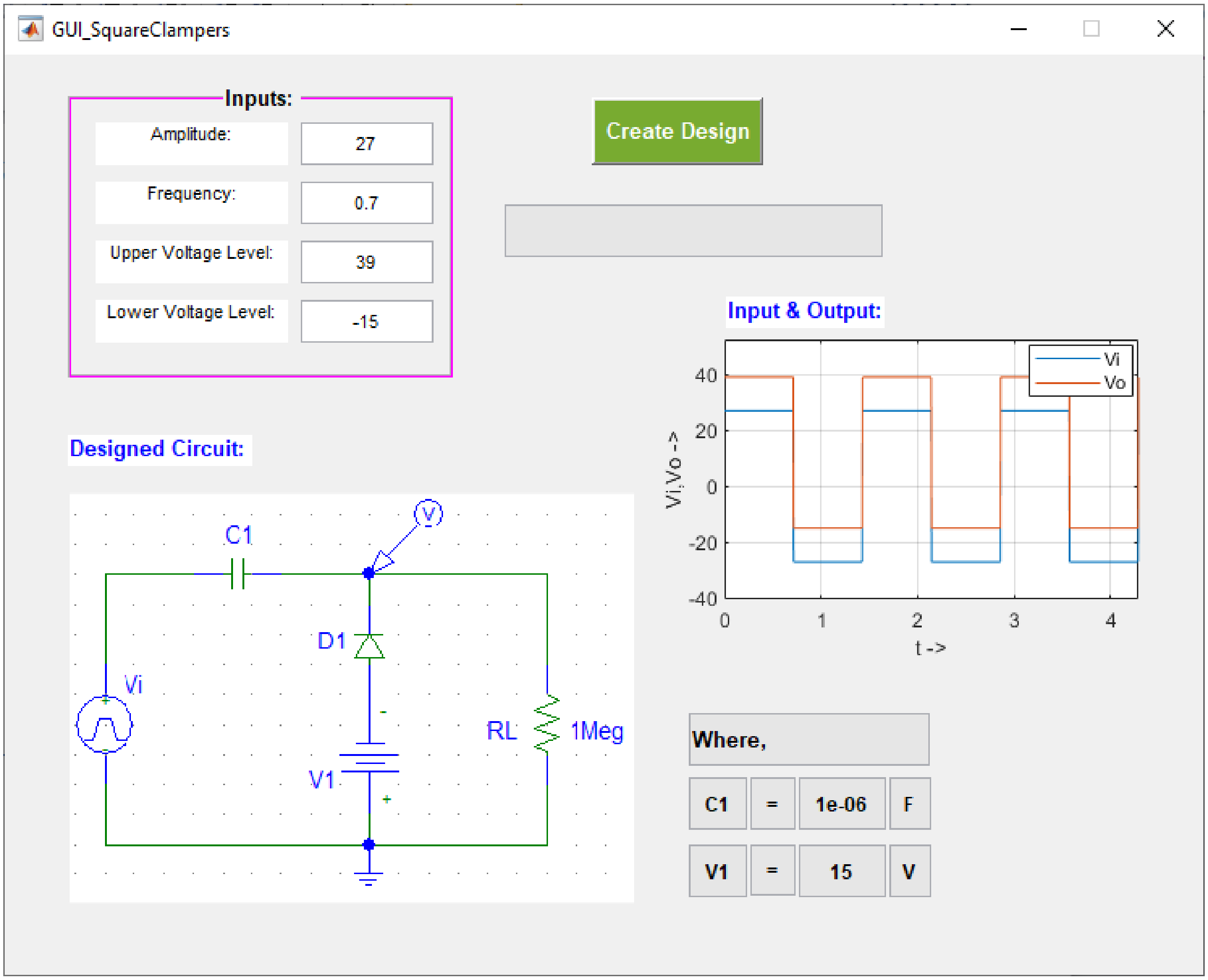
2. Negative Clamper :



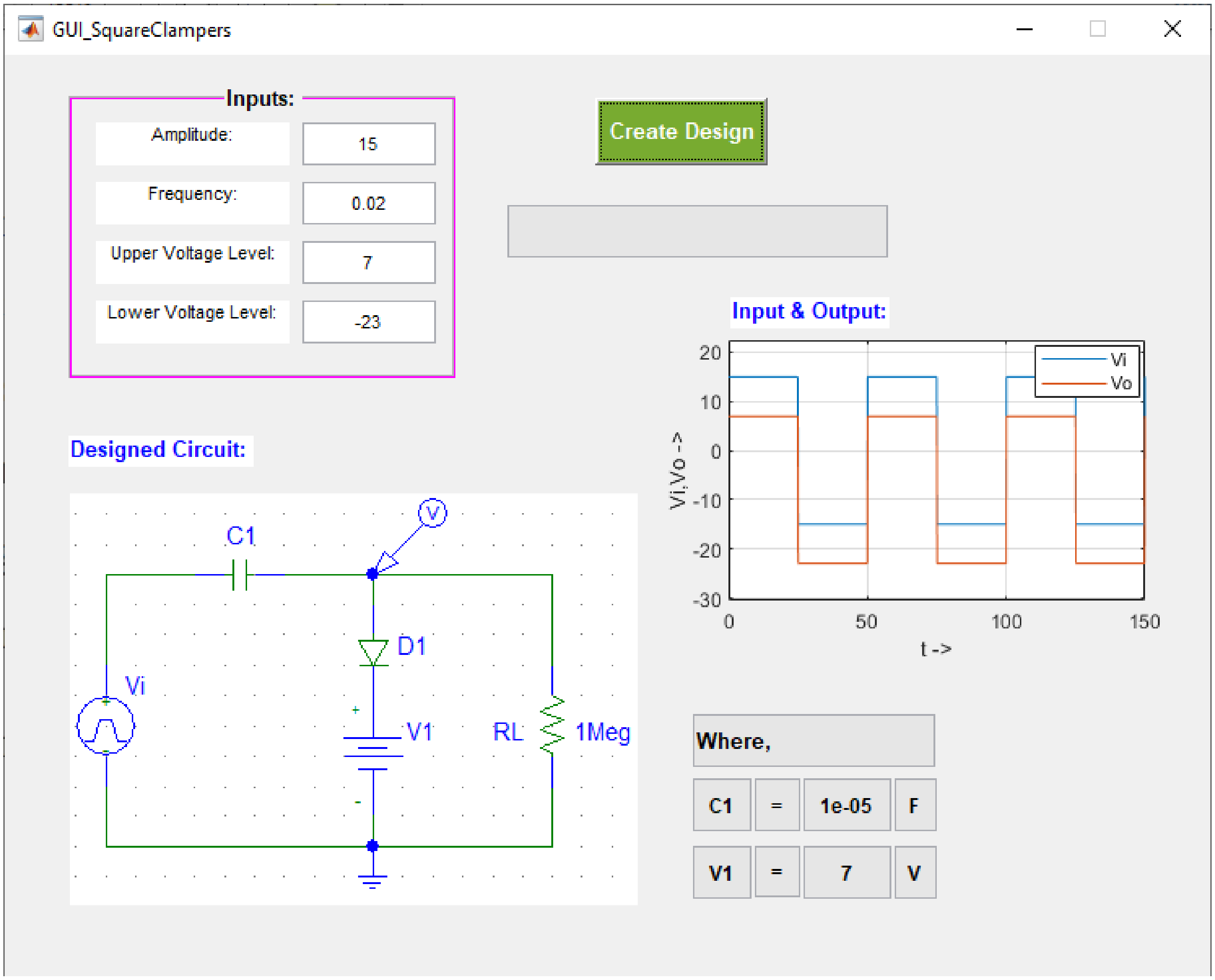
3. Positive Bias - Positive Clamper :



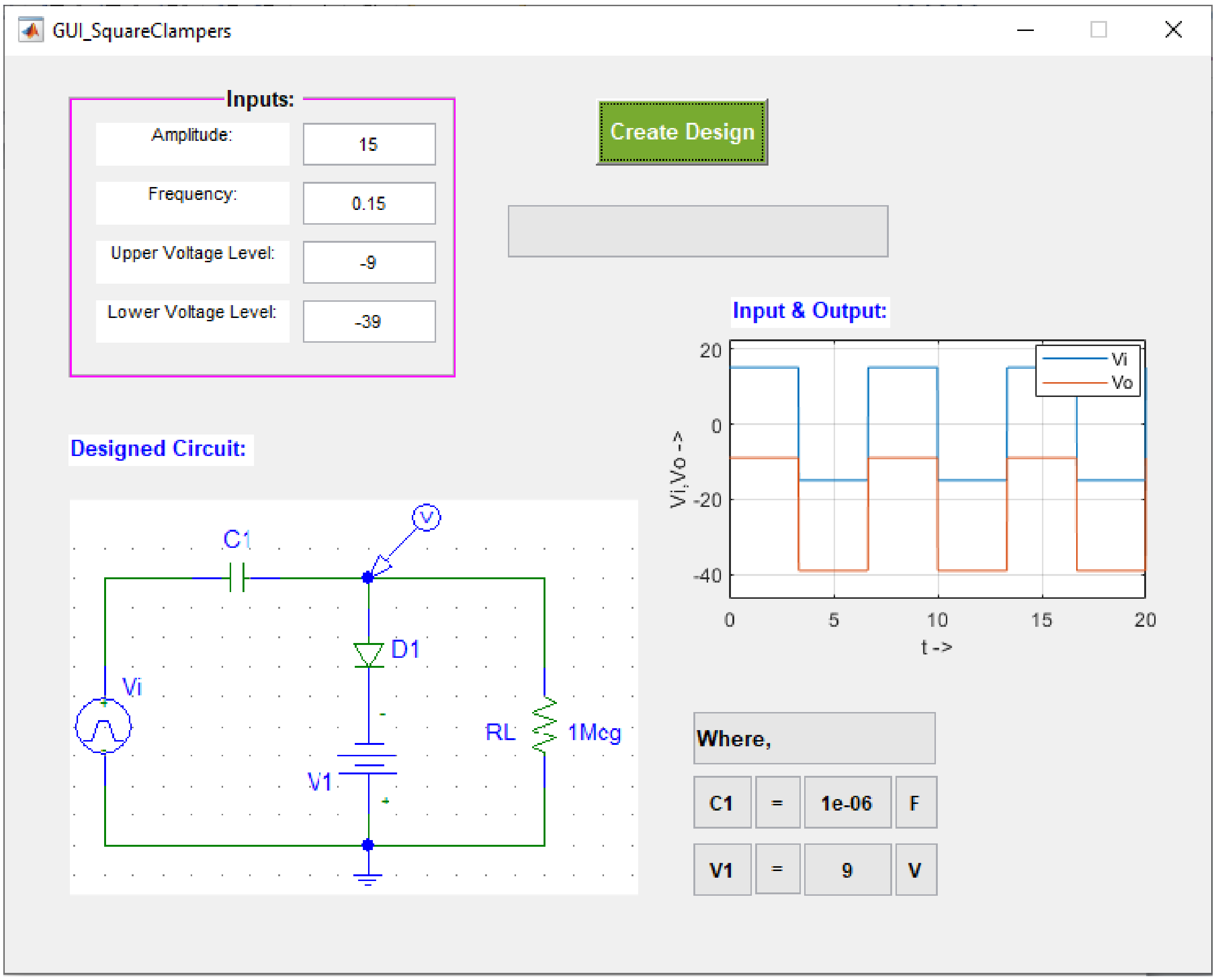
4. Negative Bias – Positive Clamper:



5. Positive Bias - Negative Clamper :

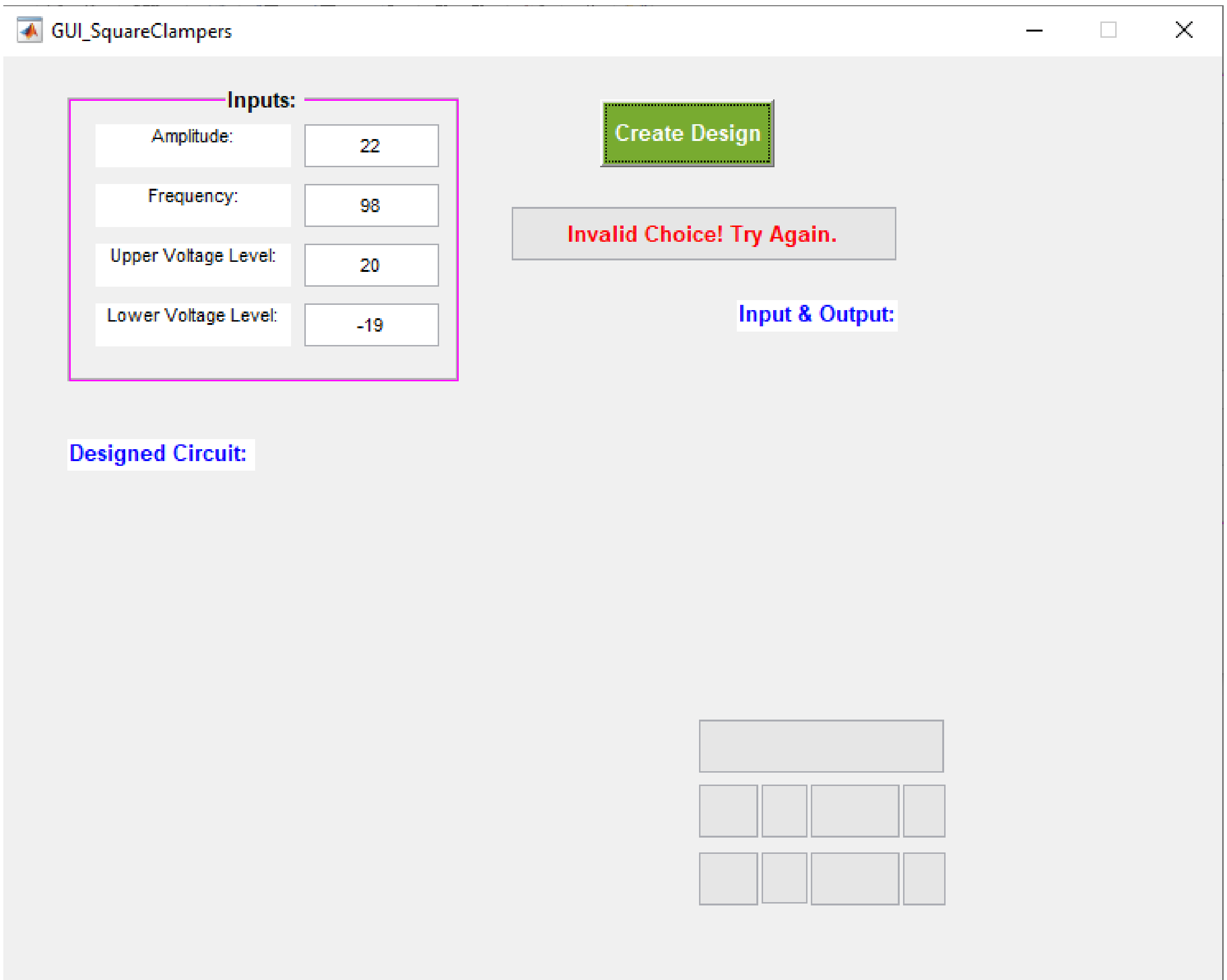


6. Negative Bias - Negative Clamper :

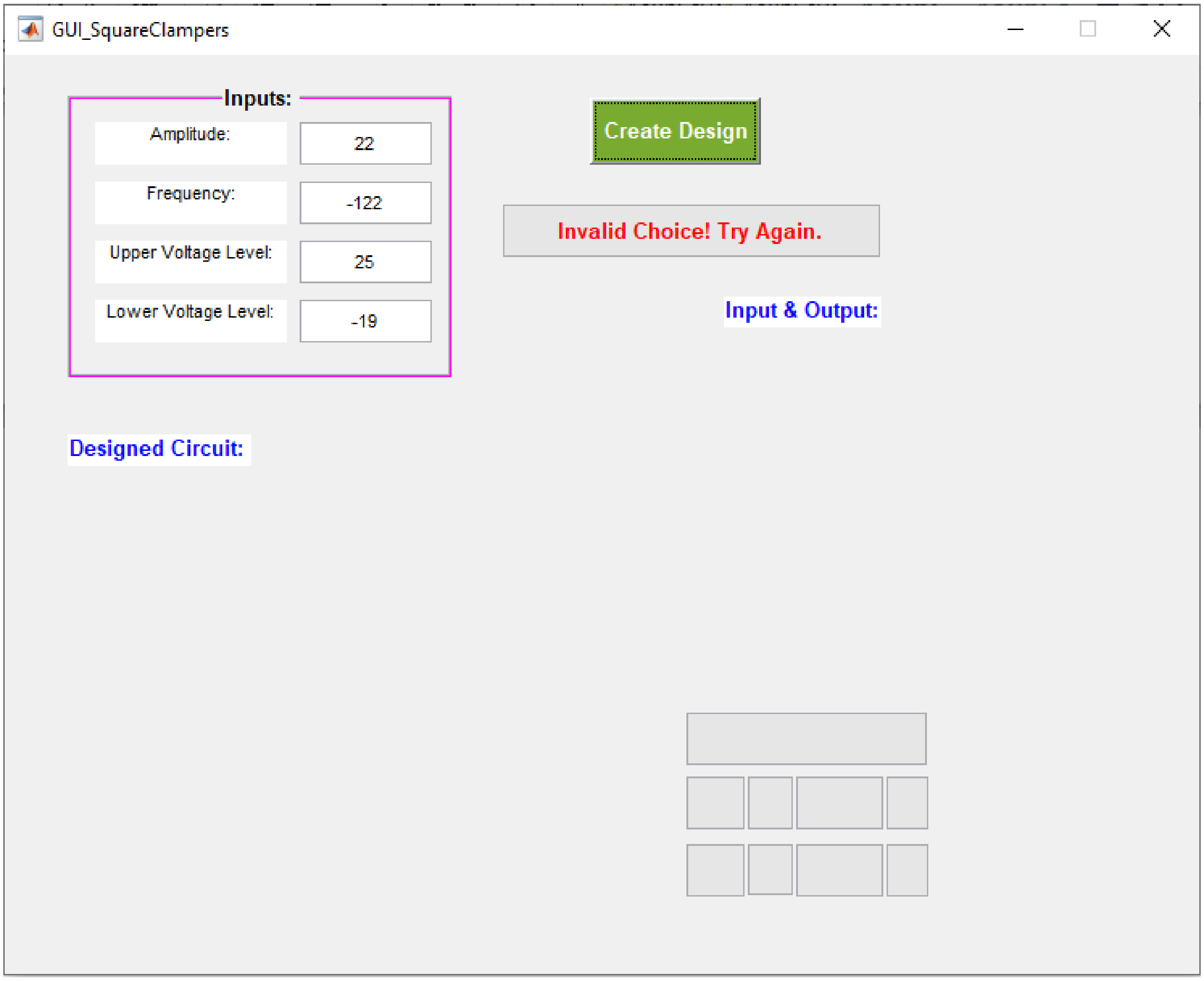


7. Invalid Cases:

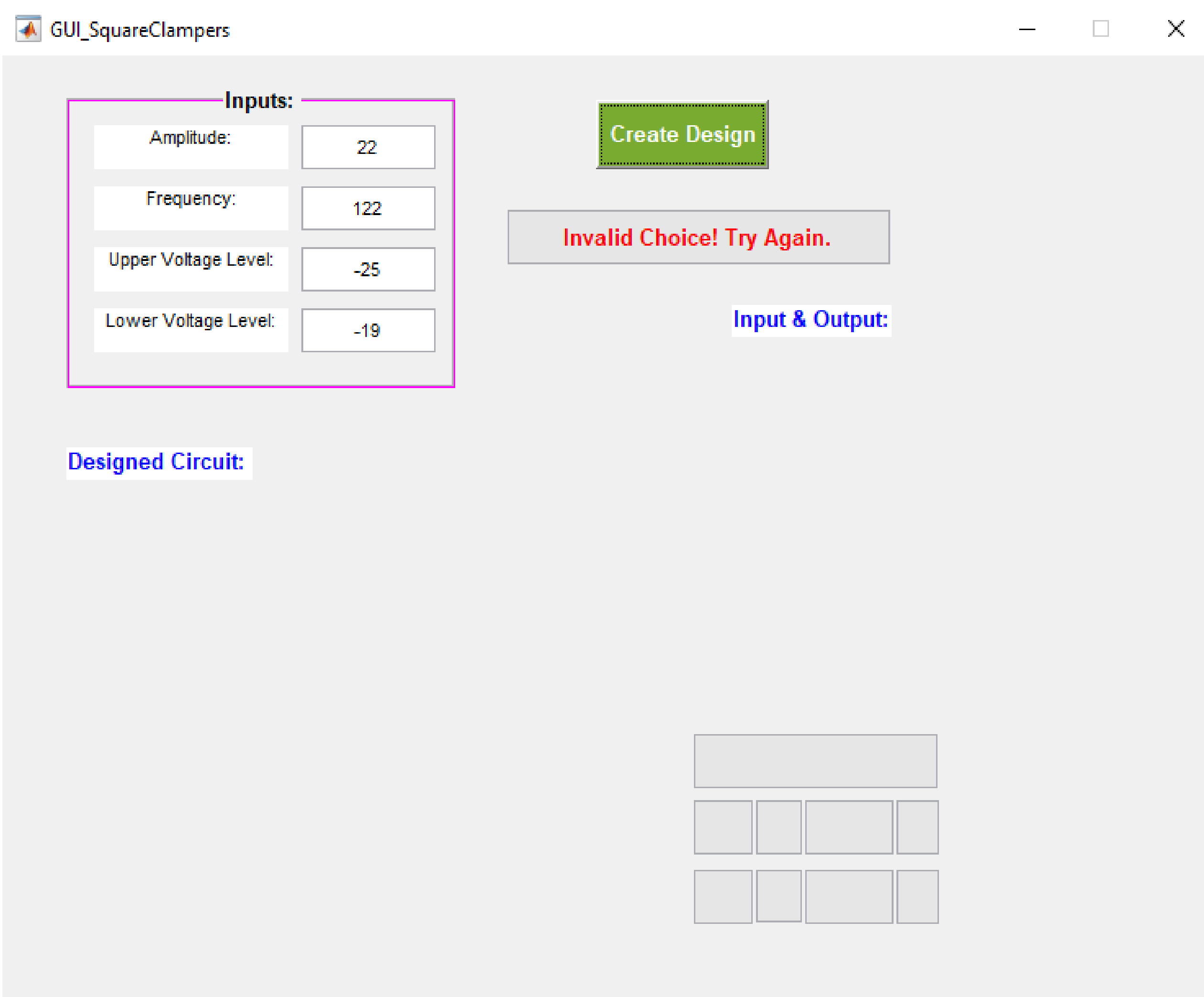
i. (Upper Voltage Level-Lower Voltage Level) ≠ (2 × Amplitude) :



ii. Zero/ Negative Frequency:



iii. Lower Voltage Level is given greater than Upper Voltage Level:



<u>N. B:</u> Our program is able to design the circuit for any other kind of inputs as well. User is appreciated to check for more as he desires.

(iii) **<u>Triangular Clampers:</u>**

- Now, let us choose "Triangular Wave" from "GUI_Clamper" window.
- "GUI_TriangularClampers.m" interacts with the user and calls "TriangularWave_Clamper_Design.m" that contains all necessary functions and codes to create the design and sends data back.

<u>Codes:</u>

"GUI_TriangularClampers.m" –

```
function varargout = GUI_TriangularClampers(varargin)
% GUI_TRIANGULARCLAMPERS MATLAB code for GUI_TriangularClampers.fig
%      GUI_TRIANGULARCLAMPERS, by itself, creates a new
GUI_TRIANGULARCLAMPERS or raises the existing
%      singleton*.
%
%      H = GUI_TRIANGULARCLAMPERS returns the handle to a new
GUI_TRIANGULARCLAMPERS or the handle to
%      the existing singleton*.
%
%      GUI_TRIANGULARCLAMPERS('CALLBACK',hObject,eventData,handles,...) calls
the local
%      function named CALLBACK in GUI_TRIANGULARCLAMPERS.M with the given
input arguments.
%
%      GUI_TRIANGULARCLAMPERS('Property','Value',...) creates a new
GUI_TRIANGULARCLAMPERS or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before GUI_TriangularClampers_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to GUI_TriangularClampers_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help GUI_TriangularClampers
```

```
% Last Modified by GUIDE v2.5 19-Jul-2021 03:53:14

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @GUI_TriangularClampers_OpeningFcn, ...
                   'gui_OutputFcn',  @GUI_TriangularClampers_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before GUI_TriangularClampers is made visible.
function GUI_TriangularClampers_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to GUI_TriangularClampers (see VARARGIN)

% Choose default command line output for GUI_TriangularClampers
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes GUI_TriangularClampers wait for user response (see UIRESUME)
% uiwait(handles.figure1);
```

```
% --- Outputs from this function are returned to the command line.
function varargout = GUI_TriangularClampers_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;



% --- Executes on button press in design.
function design_Callback(hObject, eventdata, handles)
% hObject    handle to design (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.invalidcheck,'String','');
set(handles.properties,'String','');
set(handles.c11,'String','');
set(handles.c12,'String','');
set(handles.c13,'String','');
set(handles.c14,'String','');
set(handles.v11,'String','');
set(handles.v12,'String','');
set(handles.v13,'String','');
set(handles.v14,'String','');
axes(handles.diagram);
cla;
axes(handles.plots);
cla;

amp=str2double(get(handles.amplitude,'String'));
f=str2double(get(handles.frequency,'String'));
h=str2double(get(handles.upper,'String'));
l=str2double(get(handles.lower,'String'));

if  f<=0 || (h-l)~=(2*amp)
    set(handles.invalidcheck,'String','Invalid Choice! Try Again.');
else
    [t,vin,vout,img,c1,v1]=TriangularWave_Clamper_Design(amp,f,h,l);
```

```
    axes(handles.plots);
    cla;
    plot(t,vin);
    hold on;
    plot(t,vout);
    if max(vout)>=max(vin)
        axis([0,t(end),(min(vin)-amp/2),(max(vout)+amp/2)]);
    else
        axis([0,t(end),(min(vout)-amp/2),(max(vin)+amp/2)]);
    end
    grid on;
    legend('Vi','Vo');
    xlabel('t ->');
    ylabel('Vi,Vo ->');
    axes(handles.diagram);
    cla;
    imshow(img);
    if  v1==0
        set(handles.properties,'String','Where,');
        set(handles.c11,'String','C1');
        set(handles.c12,'String','=');
        set(handles.c13,'String',num2str(c1));
        set(handles.c14,'String','F');
    else
        set(handles.properties,'String','Where,');
        set(handles.c11,'String','C1');
        set(handles.c12,'String','=');
        set(handles.c13,'String',num2str(c1));
        set(handles.c14,'String','F');
        set(handles.v11,'String','V1');
        set(handles.v12,'String','=');
        set(handles.v13,'String',num2str(v1));
        set(handles.v14,'String','V');
    end
end


function properties_Callback(hObject, eventdata, handles)
% hObject    handle to properties (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of properties as text
%        str2double(get(hObject,'String')) returns contents of properties as a double


% --- Executes during object creation, after setting all properties.
function properties_CreateFcn(hObject, eventdata, handles)
% hObject    handle to properties (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function c11_Callback(hObject, eventdata, handles)
% hObject    handle to c11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of c11 as text
%        str2double(get(hObject,'String')) returns contents of c11 as a double


% --- Executes during object creation, after setting all properties.
function c11_CreateFcn(hObject, eventdata, handles)
% hObject    handle to c11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function c12_Callback(hObject, eventdata, handles)
% hObject    handle to c12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of c12 as text
%        str2double(get(hObject,'String')) returns contents of c12 as a double


% --- Executes during object creation, after setting all properties.
function c12_CreateFcn(hObject, eventdata, handles)
% hObject    handle to c12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function c13_Callback(hObject, eventdata, handles)
% hObject    handle to c13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of c13 as text
%        str2double(get(hObject,'String')) returns contents of c13 as a double


% --- Executes during object creation, after setting all properties.
function c13_CreateFcn(hObject, eventdata, handles)
% hObject    handle to c13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function v11_Callback(hObject, eventdata, handles)
% hObject    handle to v11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v11 as text
%        str2double(get(hObject,'String')) returns contents of v11 as a double


% --- Executes during object creation, after setting all properties.
function v11_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function v12_Callback(hObject, eventdata, handles)
% hObject    handle to v12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v12 as text
%        str2double(get(hObject,'String')) returns contents of v12 as a double


% --- Executes during object creation, after setting all properties.
function v12_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v12 (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function v13_Callback(hObject, eventdata, handles)
% hObject    handle to v13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v13 as text
%        str2double(get(hObject,'String')) returns contents of v13 as a double


% --- Executes during object creation, after setting all properties.
function v13_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function invalidcheck_Callback(hObject, eventdata, handles)
% hObject    handle to invalidcheck (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of invalidcheck as text
```

```
%       str2double(get(hObject,'String')) returns contents of invalidcheck as a double

% --- Executes during object creation, after setting all properties.
function invalidcheck_CreateFcn(hObject, eventdata, handles)
% hObject    handle to invalidcheck (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function c14_Callback(hObject, eventdata, handles)
% hObject    handle to c14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of c14 as text
%       str2double(get(hObject,'String')) returns contents of c14 as a double

% --- Executes during object creation, after setting all properties.
function c14_CreateFcn(hObject, eventdata, handles)
% hObject    handle to c14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function v14_Callback(hObject, eventdata, handles)
% hObject    handle to v14 (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v14 as text
%        str2double(get(hObject,'String')) returns contents of v14 as a double


% --- Executes during object creation, after setting all properties.
function v14_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function amplitude_Callback(hObject, eventdata, handles)
% hObject    handle to amplitude (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of amplitude as text
%        str2double(get(hObject,'String')) returns contents of amplitude as a double


% --- Executes during object creation, after setting all properties.
function amplitude_CreateFcn(hObject, eventdata, handles)
% hObject    handle to amplitude (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
```

```
    set(hObject,'BackgroundColor','white');
end


function frequency_Callback(hObject, eventdata, handles)
% hObject    handle to frequency (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of frequency as text
%        str2double(get(hObject,'String')) returns contents of frequency as a double


% --- Executes during object creation, after setting all properties.
function frequency_CreateFcn(hObject, eventdata, handles)
% hObject    handle to frequency (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function upper_Callback(hObject, eventdata, handles)
% hObject    handle to upper (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of upper as text
%        str2double(get(hObject,'String')) returns contents of upper as a double


% --- Executes during object creation, after setting all properties.
function upper_CreateFcn(hObject, eventdata, handles)
% hObject    handle to upper (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function lower_Callback(hObject, eventdata, handles)
% hObject    handle to lower (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of lower as text
%        str2double(get(hObject,'String')) returns contents of lower as a double


% --- Executes during object creation, after setting all properties.
function lower_CreateFcn(hObject, eventdata, handles)
% hObject    handle to lower (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

"TriangularWave_Clamper_Design.m" –

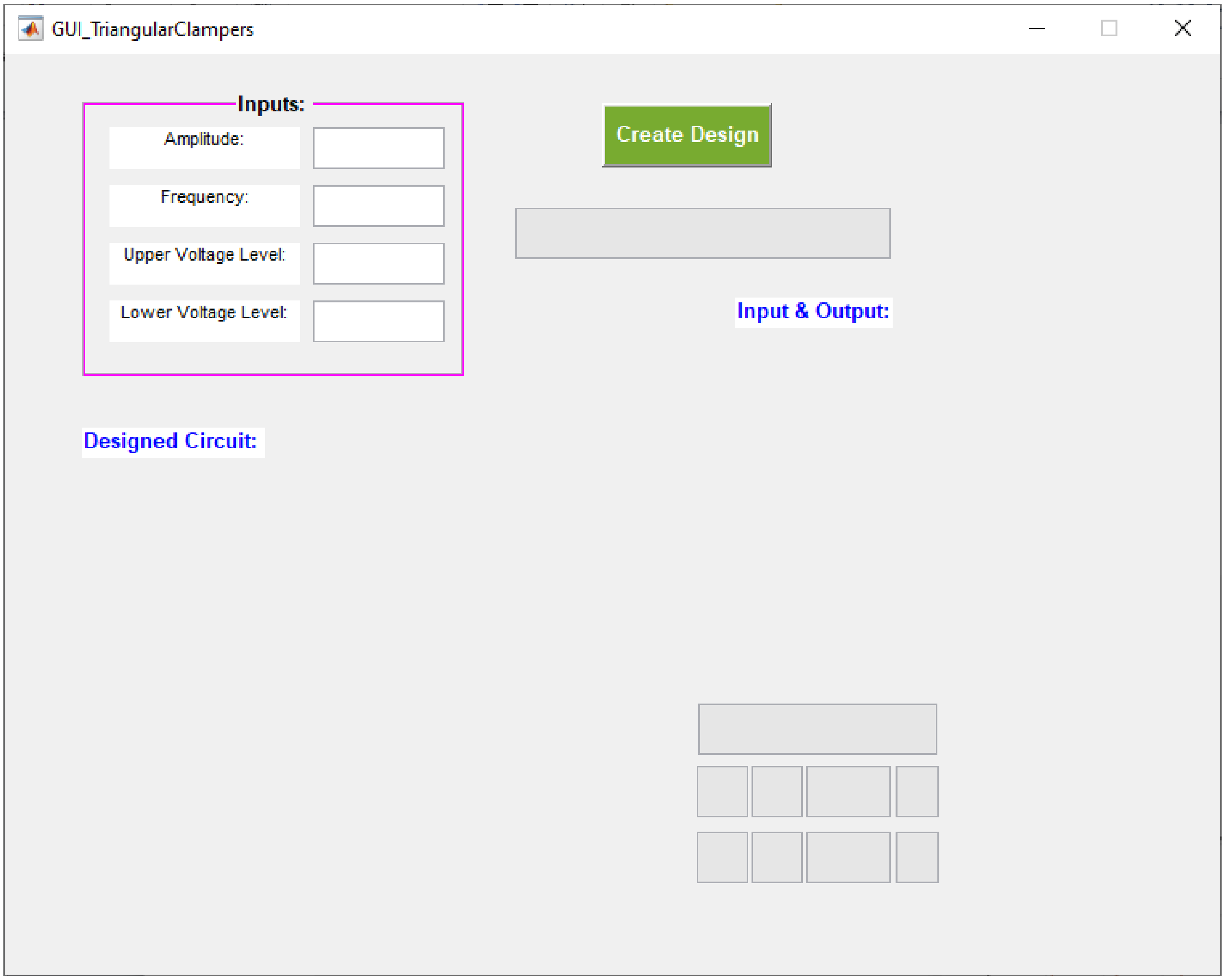
```
function [t,vin,vout,img,c1,v1]=TriangularWave_Clamper_Design(vamp,f,vup,vlow)
    c1=10^-6;
    R=10^6;
    T=1/f;
```

```
    while 5*R*c1<=T/2
        c1=c1*10;
    end

    if vlow==0 && vup==2*vamp
        [t,vin,vout,img,v1]=positive_clamper(vamp,f,vup,vlow);
    elseif vlow==-2*vamp && vup==0
        [t,vin,vout,img,v1]=negative_clamper(vamp,f,vup,vlow);
    elseif vlow>0
        [t,vin,vout,img,v1]=positive_bias_positive_clamper(vamp,f,vup,vlow);
    elseif vlow<0 && vlow>(-vamp)
        [t,vin,vout,img,v1]=negative_bias_positive_clamper(vamp,f,vup,vlow);
    elseif vup>0 && vup<=vamp
        [t,vin,vout,img,v1]=positive_bias_negative_clamper(vamp,f,vup,vlow);
    elseif vup<0
        [t,vin,vout,img,v1]=negative_bias_negative_clamper(vamp,f,vup,vlow);
    end
end

function [t,vin,vout,img,v1]=positive_clamper(vamp,f,vup,vlow)
    v1=0;
    T=1/f;
    t=linspace(0,3*T,1200);
    vin=vamp.*sawtooth((2*pi*f)*t,0.5);
    vout(1:1200)=0;
    i=1;
    for i=1:1200
        vout(i)=vin(i)+vamp;
    end
    img=imread('TriangularClamper1.PNG');
end

function [t,vin,vout,img,v1]=negative_clamper(vamp,f,vup,vlow)
    v1=0;
    T=1/f;
    t=linspace(0,3*T,1200);
    vin=vamp.*sawtooth((2*pi*f)*t,0.5);
    vout(1:1200)=0;
    i=1;
    while i<=1200
```

```
        if i<=200
            vout(i)=0;
        else
            vout(i)=-vamp+vin(i);
        end
        i=i+1;
    end
    img=imread('TriangularClamper2.PNG');
end

function [t,vin,vout,img,v1]=positive_bias_positive_clamper(vamp,f,vup,vlow)
    v1=vlow;
    T=1/f;
    t=linspace(0,3*T,1200);
    vin=vamp.*sawtooth((2*pi*f)*t,0.5);
    vout(1:1200)=0;
    i=1;
    while i<=1200
        vout(i)=vamp+v1+vin(i);
        i=i+1;
    end
    img=imread('TriangularClamper3.PNG');
end

function [t,vin,vout,img,v1]=negative_bias_positive_clamper(vamp,f,vup,vlow)
    v1=-vlow;
    T=1/f;
    t=linspace(0,3*T,1200);
    vin=vamp.*sawtooth((2*pi*f)*t,0.5);
    vout(1:1200)=0;
    i=1;
    while i<=1200
        vout(i)=vin(i)+vamp-v1;
        i=i+1;
    end
    img=imread('TriangularClamper4.PNG');
end

function [t,vin,vout,img,v1]=positive_bias_negative_clamper(vamp,f,vup,vlow)
    v1=vup;
```

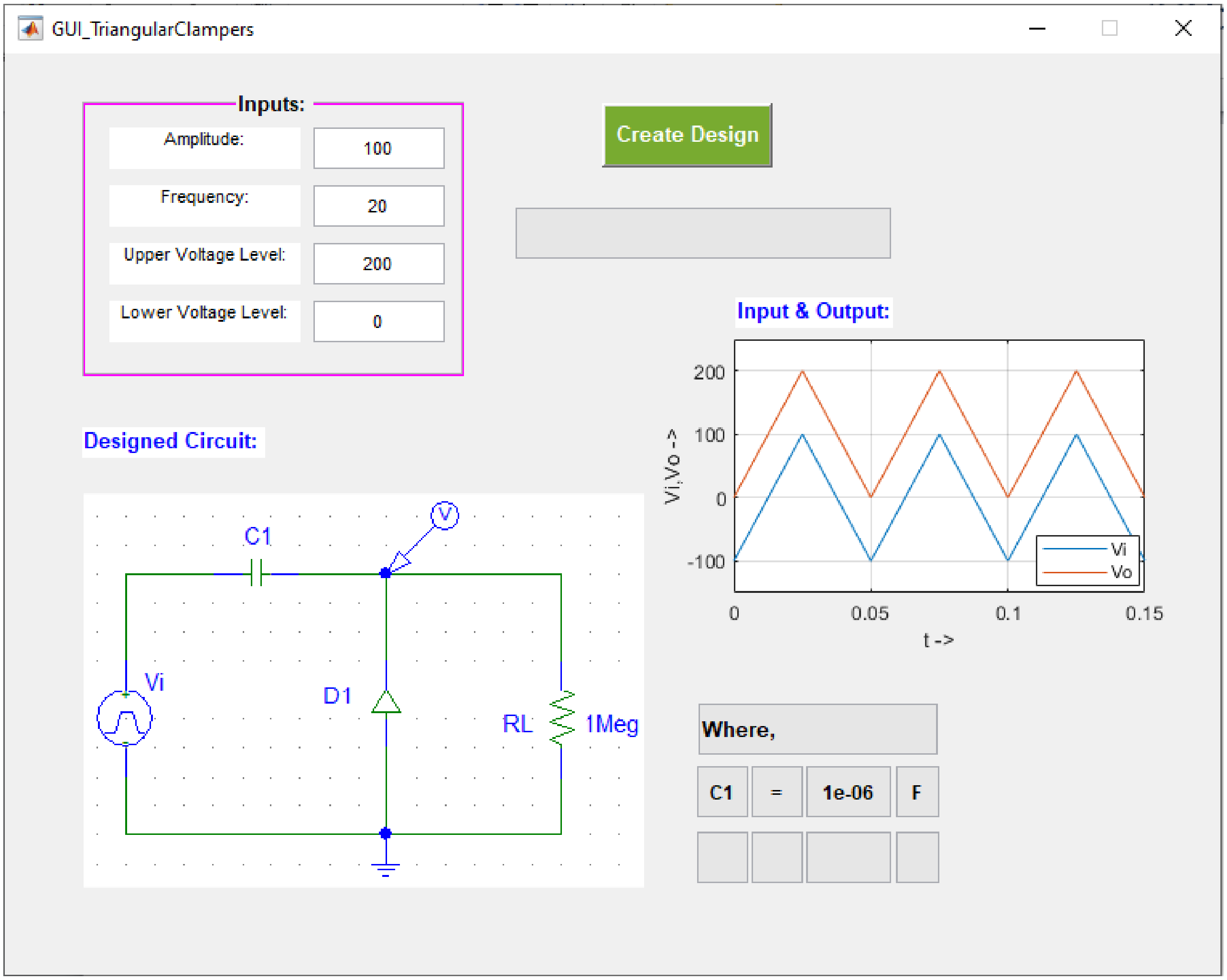
```
    T=1/f;
    t=linspace(0,3*T,1200);
    vin=vamp.*sawtooth((2*pi*f)*t,0.5);
    vout(1:1200)=0;
    i=1;
    while i<=1200
        if i<=100
            if vin(i)+vamp<=v1
                vout(i)=vin(i)+vamp;
            else
                vout(i)=v1;
            end
        elseif i>100 && i<=200
            vout(i)=v1;
        else
            vout(i)=vin(i)-vamp+v1;
        end
        i=i+1;
    end
    img=imread('TriangularClamper5.PNG');
end

function [t,vin,vout,img,v1]=negative_bias_negative_clamper(vamp,f,vup,vlow)
    v1=-vup;
    T=1/f;
    t=linspace(0,3*T,1200);
    vin=vamp.*sawtooth((2*pi*f)*t,0.5);
    vout(1:1200)=0;
    i=1;
    while i<=1200
        if i<=200
            vout(i)=-v1;
        else
            vout(i)=vin(i)-vamp-v1;
        end
        i=i+1;
    end
    img=imread('TriangularClamper6.PNG');
end
```

Corresponding Window:

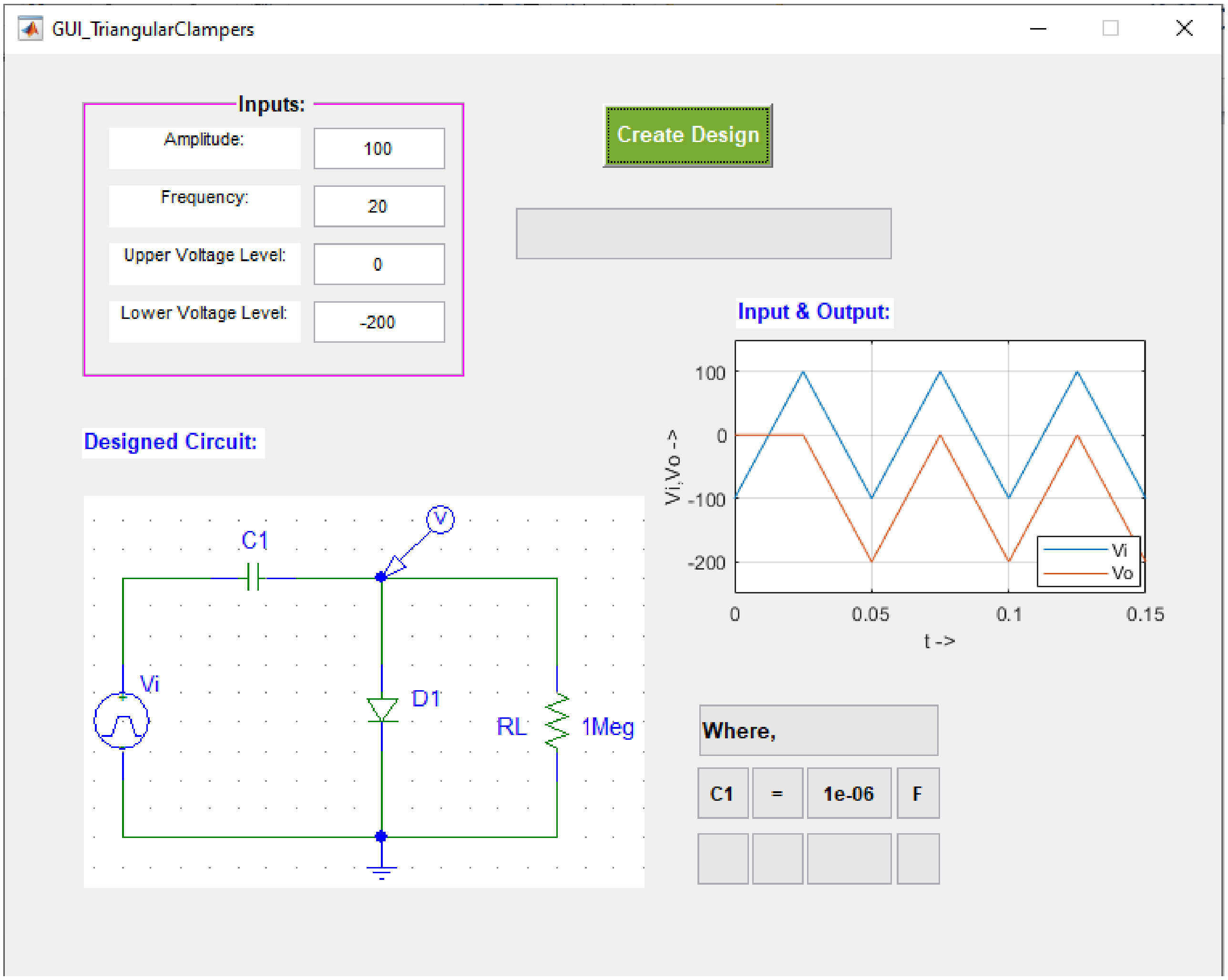


Sample Outputs:

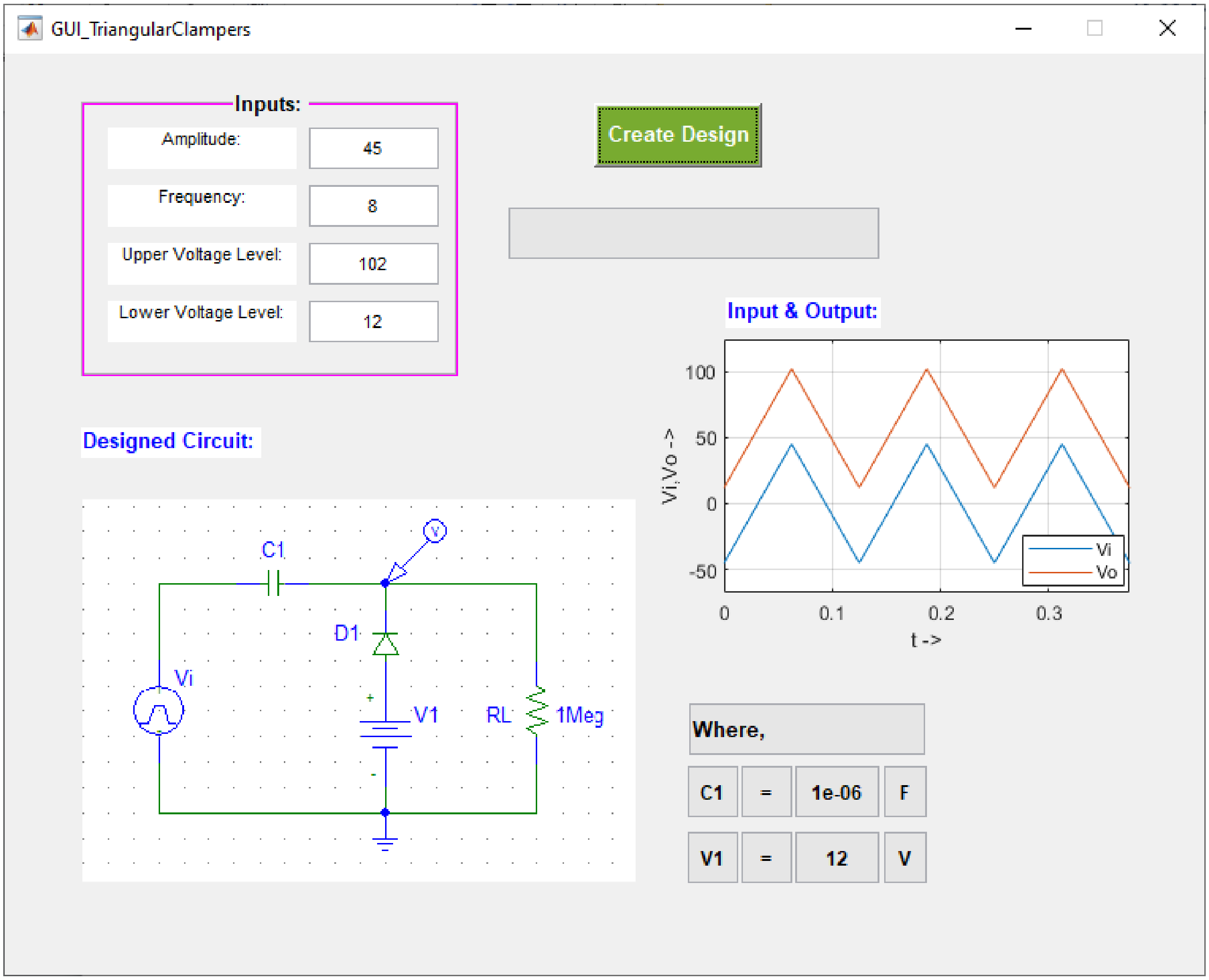
1. Positive Clamper :



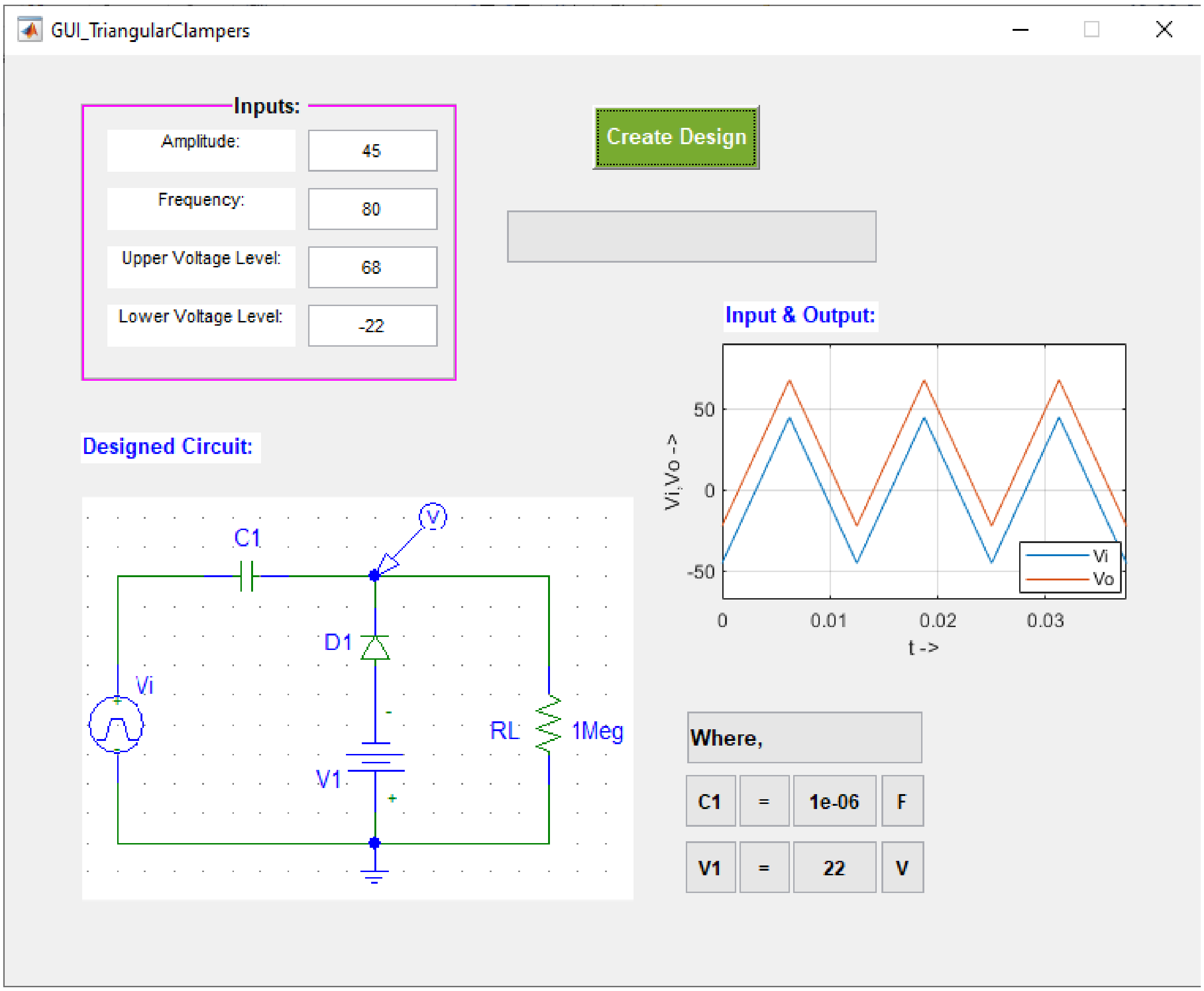
2. Negative Clamper :



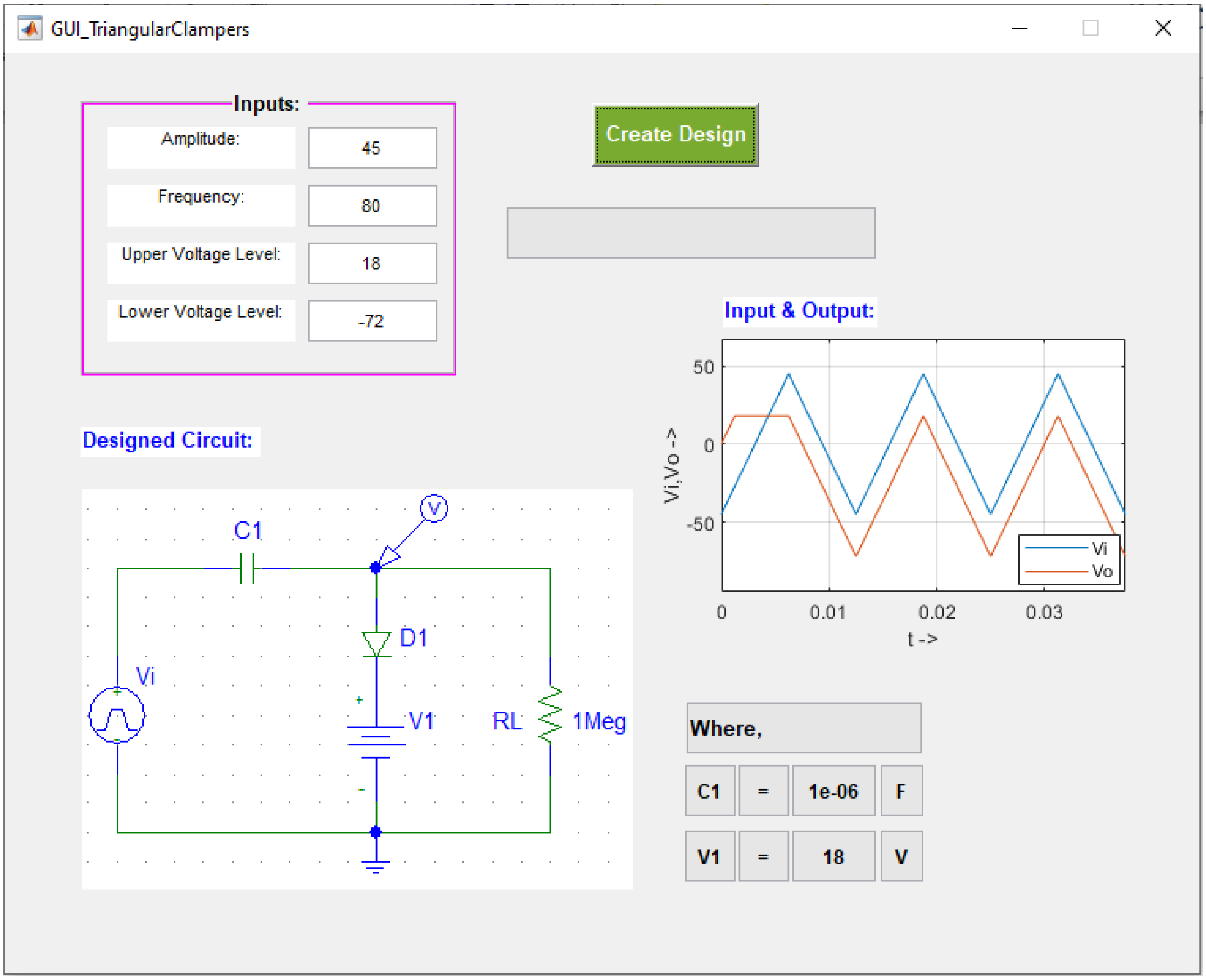
3. Positive Bias - Positive Clamper :



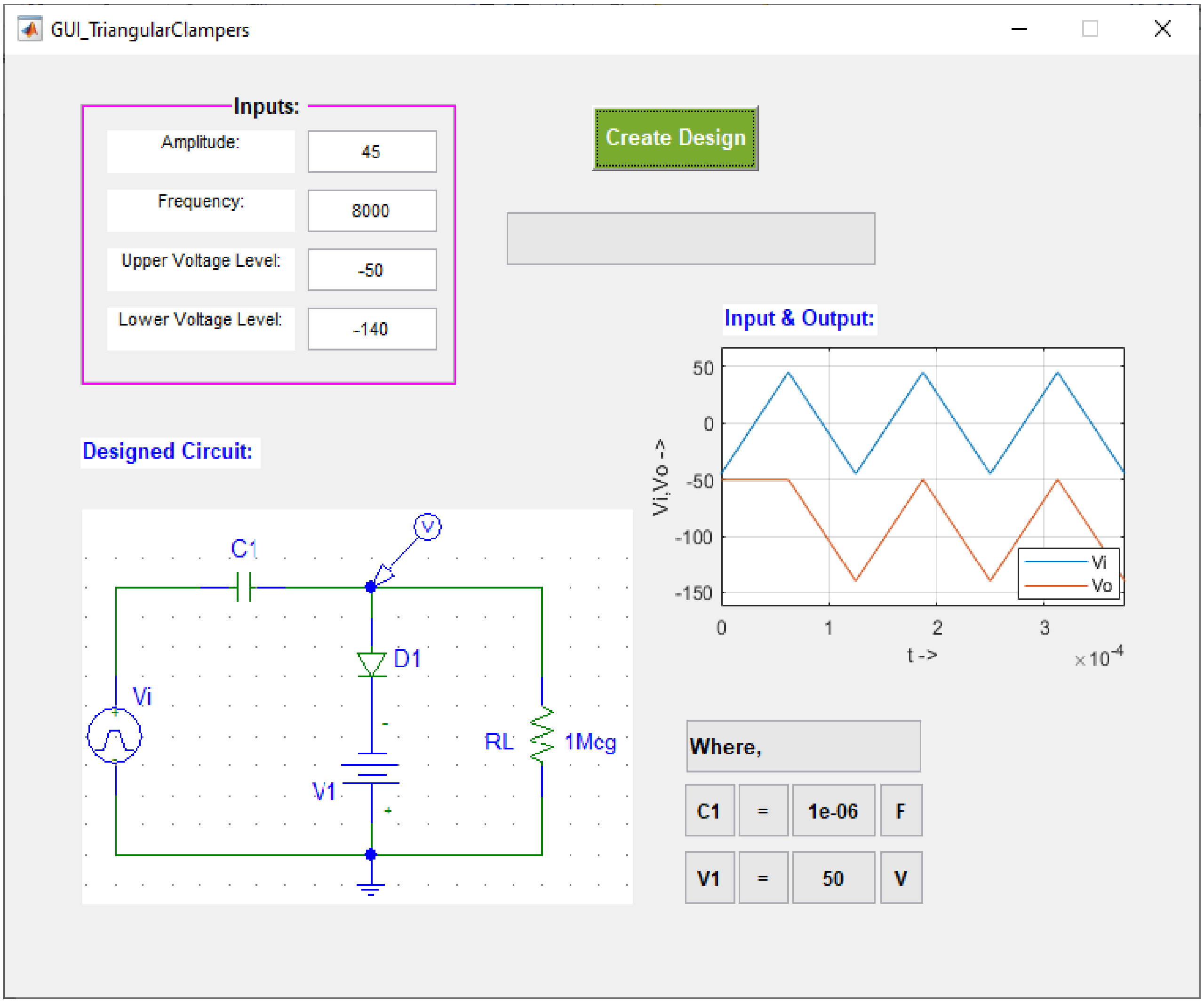
4. Negative Bias – Positive Clamper:



5. Positive Bias - Negative Clamper :

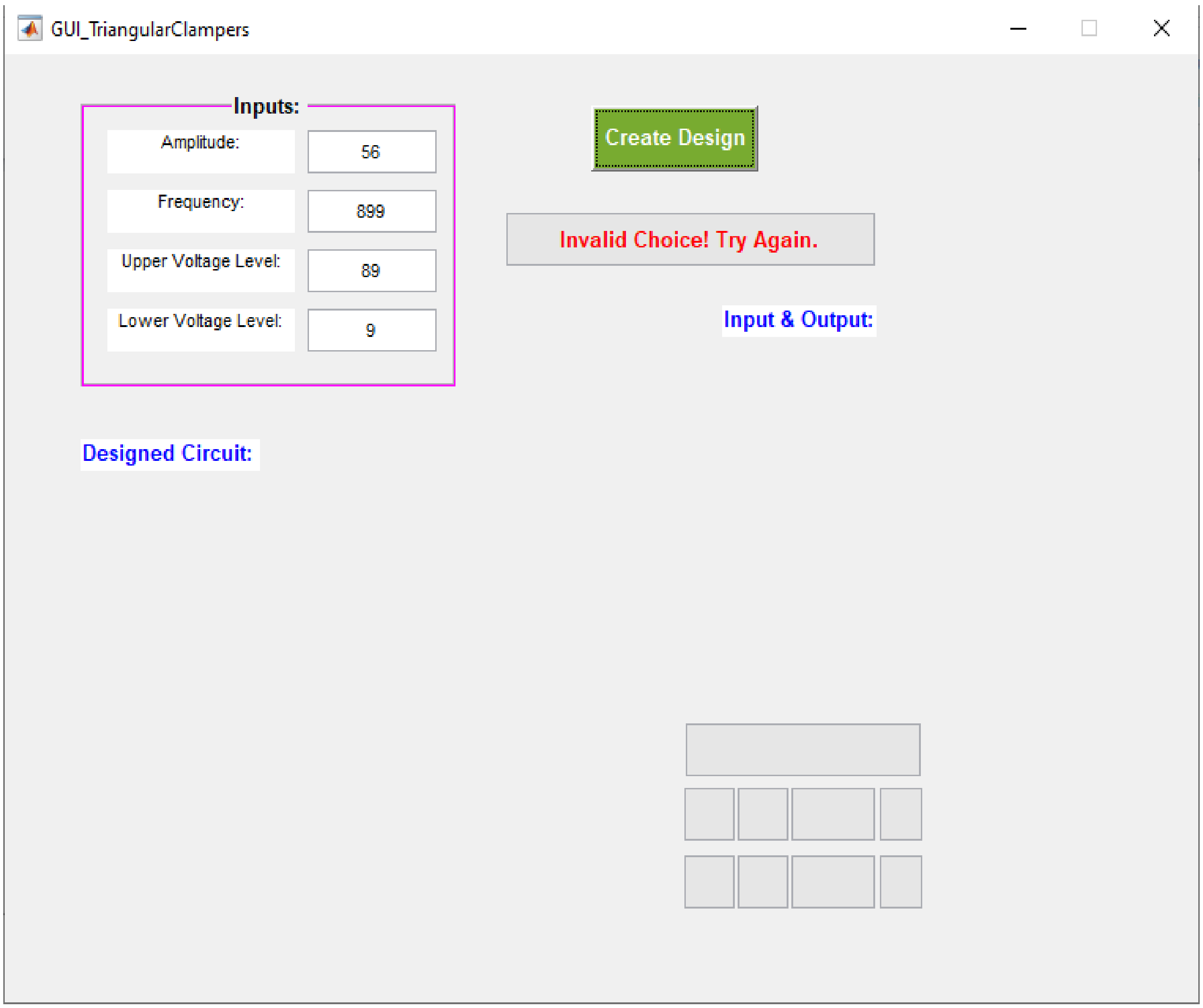


6. Negative Bias - Negative Clamper :

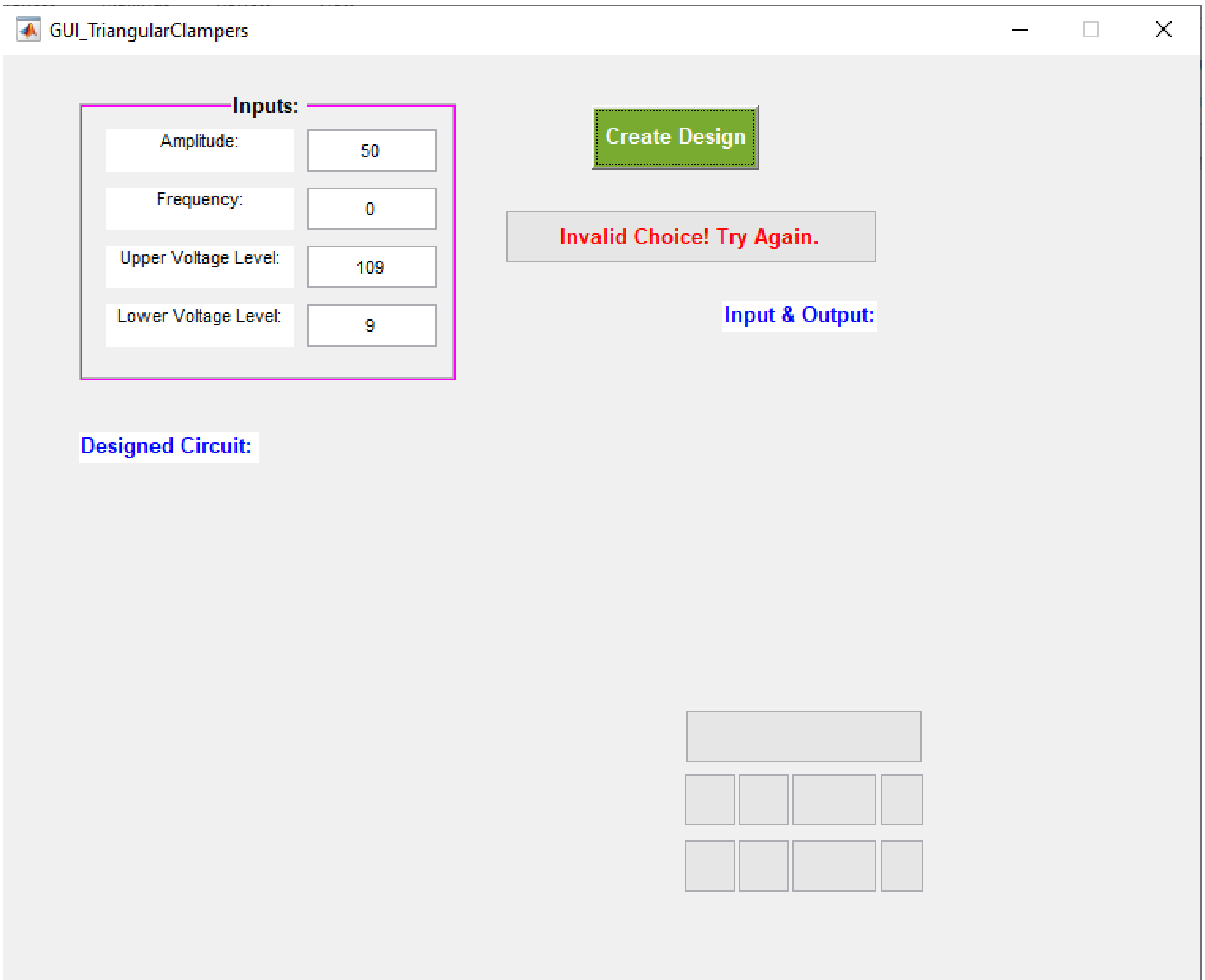


7. Invalid Cases:

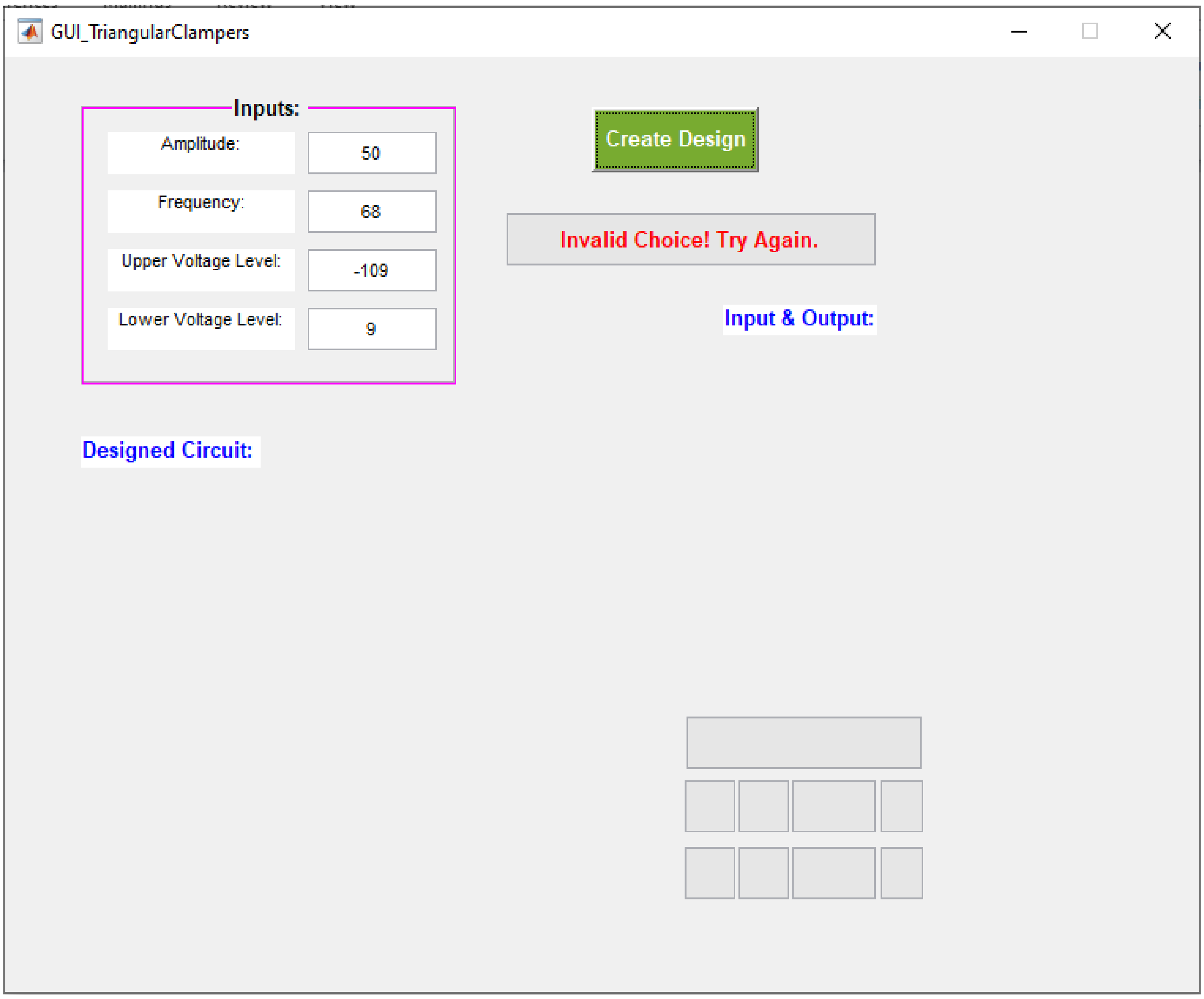
i. (Upper Voltage Level-Lower Voltage Level) $\neq$ (2 $\times$ Amplitude) :



ii. Zero/ Negative Frequency:



iii. Lower Voltage Level is given greater than Upper Voltage Level:



<u>N. B:</u> Our program is able to design the circuit for any other kind of inputs as well. User is appreciated to check for more as he desires.

(iv) **<u>Sawtooth Clampers:</u>**

- Now, let us choose "Sawtooth Wave" from "GUI_Clamper" window.
- "GUI_SawtoothClampers.m" interacts with the user and calls "SatoothWave_Clamper_Design.m" that contains all necessary functions and codes to create the design and sends data back.

Codes:

"GUI_SawtoothClampers.m" –

```
function varargout = GUI_SawtoothClampers(varargin)
% GUI_SAWTOOTHCLAMPERS MATLAB code for GUI_SawtoothClampers.fig
%      GUI_SAWTOOTHCLAMPERS, by itself, creates a new
GUI_SAWTOOTHCLAMPERS or raises the existing
%      singleton*.
%
%      H = GUI_SAWTOOTHCLAMPERS returns the handle to a new
GUI_SAWTOOTHCLAMPERS or the handle to
%      the existing singleton*.
%
%      GUI_SAWTOOTHCLAMPERS('CALLBACK',hObject,eventData,handles,...) calls
the local
%      function named CALLBACK in GUI_SAWTOOTHCLAMPERS.M with the given
input arguments.
%
%      GUI_SAWTOOTHCLAMPERS('Property','Value',...) creates a new
GUI_SAWTOOTHCLAMPERS or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before GUI_SawtoothClampers_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to GUI_SawtoothClampers_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help GUI_SawtoothClampers
```

```
% Last Modified by GUIDE v2.5 20-Jul-2021 05:28:11

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @GUI_SawtoothClampers_OpeningFcn, ...
                   'gui_OutputFcn',  @GUI_SawtoothClampers_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before GUI_SawtoothClampers is made visible.
function GUI_SawtoothClampers_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to GUI_SawtoothClampers (see VARARGIN)

% Choose default command line output for GUI_SawtoothClampers
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes GUI_SawtoothClampers wait for user response (see UIRESUME)
% uiwait(handles.figure1);
```

```
% --- Outputs from this function are returned to the command line.
function varargout = GUI_SawtoothClampers_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;



% --- Executes on button press in design.
function design_Callback(hObject, eventdata, handles)
% hObject    handle to design (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.invalidcheck,'String','');
set(handles.properties,'String','');
set(handles.c11,'String','');
set(handles.c12,'String','');
set(handles.c13,'String','');
set(handles.c14,'String','');
set(handles.v11,'String','');
set(handles.v12,'String','');
set(handles.v13,'String','');
set(handles.v14,'String','');
axes(handles.diagram);
cla;
axes(handles.plots);
cla;

amp=str2double(get(handles.amplitude,'String'));
f=str2double(get(handles.frequency,'String'));
h=str2double(get(handles.upper,'String'));
l=str2double(get(handles.lower,'String'));
if  f<=0 || (h-l)~=(2*amp)
    set(handles.invalidcheck,'String','Invalid Choice! Try Again.');
else
    [t,vin,vout,img,c1,v1]=SawtoothWave_Clamper_Design(amp,f,h,l);
    axes(handles.plots);
```

```
    cla;
    plot(t,vin);
    hold on;
    plot(t,vout);
    if max(vout)>=max(vin)
        axis([0,t(end),(min(vin)-amp/2),(max(vout)+amp/2)]);
    else
        axis([0,t(end),(min(vout)-amp/2),(max(vin)+amp/2)]);
    end
    grid on;
    legend('Vi','Vo');
    xlabel('t ->');
    ylabel('Vi,Vo ->');
    axes(handles.diagram);
    cla;
    imshow(img);
    if  v1==0
        set(handles.properties,'String','Where,');
        set(handles.c11,'String','C1');
        set(handles.c12,'String','=');
        set(handles.c13,'String',num2str(c1));
        set(handles.c14,'String','F');
    else
        set(handles.properties,'String','Where,');
        set(handles.c11,'String','C1');
        set(handles.c12,'String','=');
        set(handles.c13,'String',num2str(c1));
        set(handles.c14,'String','F');
        set(handles.v11,'String','V1');
        set(handles.v12,'String','=');
        set(handles.v13,'String',num2str(v1));
        set(handles.v14,'String','V');
    end
end


function properties_Callback(hObject, eventdata, handles)
% hObject    handle to properties (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of properties as text
%        str2double(get(hObject,'String')) returns contents of properties as a double


% --- Executes during object creation, after setting all properties.
function properties_CreateFcn(hObject, eventdata, handles)
% hObject    handle to properties (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function c11_Callback(hObject, eventdata, handles)
% hObject    handle to c11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of c11 as text
%        str2double(get(hObject,'String')) returns contents of c11 as a double


% --- Executes during object creation, after setting all properties.
function c11_CreateFcn(hObject, eventdata, handles)
% hObject    handle to c11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function c12_Callback(hObject, eventdata, handles)
% hObject    handle to c12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of c12 as text
%        str2double(get(hObject,'String')) returns contents of c12 as a double


% --- Executes during object creation, after setting all properties.
function c12_CreateFcn(hObject, eventdata, handles)
% hObject    handle to c12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function c13_Callback(hObject, eventdata, handles)
% hObject    handle to c13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of c13 as text
%        str2double(get(hObject,'String')) returns contents of c13 as a double


% --- Executes during object creation, after setting all properties.
function c13_CreateFcn(hObject, eventdata, handles)
% hObject    handle to c13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function v11_Callback(hObject, eventdata, handles)
% hObject    handle to v11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v11 as text
%        str2double(get(hObject,'String')) returns contents of v11 as a double


% --- Executes during object creation, after setting all properties.
function v11_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function v12_Callback(hObject, eventdata, handles)
% hObject    handle to v12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v12 as text
%        str2double(get(hObject,'String')) returns contents of v12 as a double


% --- Executes during object creation, after setting all properties.
function v12_CreateFcn(hObject, eventdata, handles)
```

```
% hObject    handle to v12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function v13_Callback(hObject, eventdata, handles)
% hObject    handle to v13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v13 as text
%        str2double(get(hObject,'String')) returns contents of v13 as a double


% --- Executes during object creation, after setting all properties.
function v13_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function invalidcheck_Callback(hObject, eventdata, handles)
% hObject    handle to invalidcheck (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of invalidcheck as text
%        str2double(get(hObject,'String')) returns contents of invalidcheck as a double


% --- Executes during object creation, after setting all properties.
function invalidcheck_CreateFcn(hObject, eventdata, handles)
% hObject    handle to invalidcheck (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function c14_Callback(hObject, eventdata, handles)
% hObject    handle to c14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of c14 as text
%        str2double(get(hObject,'String')) returns contents of c14 as a double


% --- Executes during object creation, after setting all properties.
function c14_CreateFcn(hObject, eventdata, handles)
% hObject    handle to c14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
```

```
    set(hObject,'BackgroundColor','white');
end


function v14_Callback(hObject, eventdata, handles)
% hObject    handle to v14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v14 as text
%        str2double(get(hObject,'String')) returns contents of v14 as a double


% --- Executes during object creation, after setting all properties.
function v14_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function amplitude_Callback(hObject, eventdata, handles)
% hObject    handle to amplitude (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of amplitude as text
%        str2double(get(hObject,'String')) returns contents of amplitude as a double


% --- Executes during object creation, after setting all properties.
function amplitude_CreateFcn(hObject, eventdata, handles)
% hObject    handle to amplitude (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called


% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function frequency_Callback(hObject, eventdata, handles)
% hObject    handle to frequency (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of frequency as text
%        str2double(get(hObject,'String')) returns contents of frequency as a double


% --- Executes during object creation, after setting all properties.
function frequency_CreateFcn(hObject, eventdata, handles)
% hObject    handle to frequency (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function upper_Callback(hObject, eventdata, handles)
% hObject    handle to upper (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of upper as text
```

```
%       str2double(get(hObject,'String')) returns contents of upper as a double


% --- Executes during object creation, after setting all properties.
function upper_CreateFcn(hObject, eventdata, handles)
% hObject    handle to upper (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function lower_Callback(hObject, eventdata, handles)
% hObject    handle to lower (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of lower as text
%       str2double(get(hObject,'String')) returns contents of lower as a double


% --- Executes during object creation, after setting all properties.
function lower_CreateFcn(hObject, eventdata, handles)
% hObject    handle to lower (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

"SawtoothWave_Clamper_Design.m" –

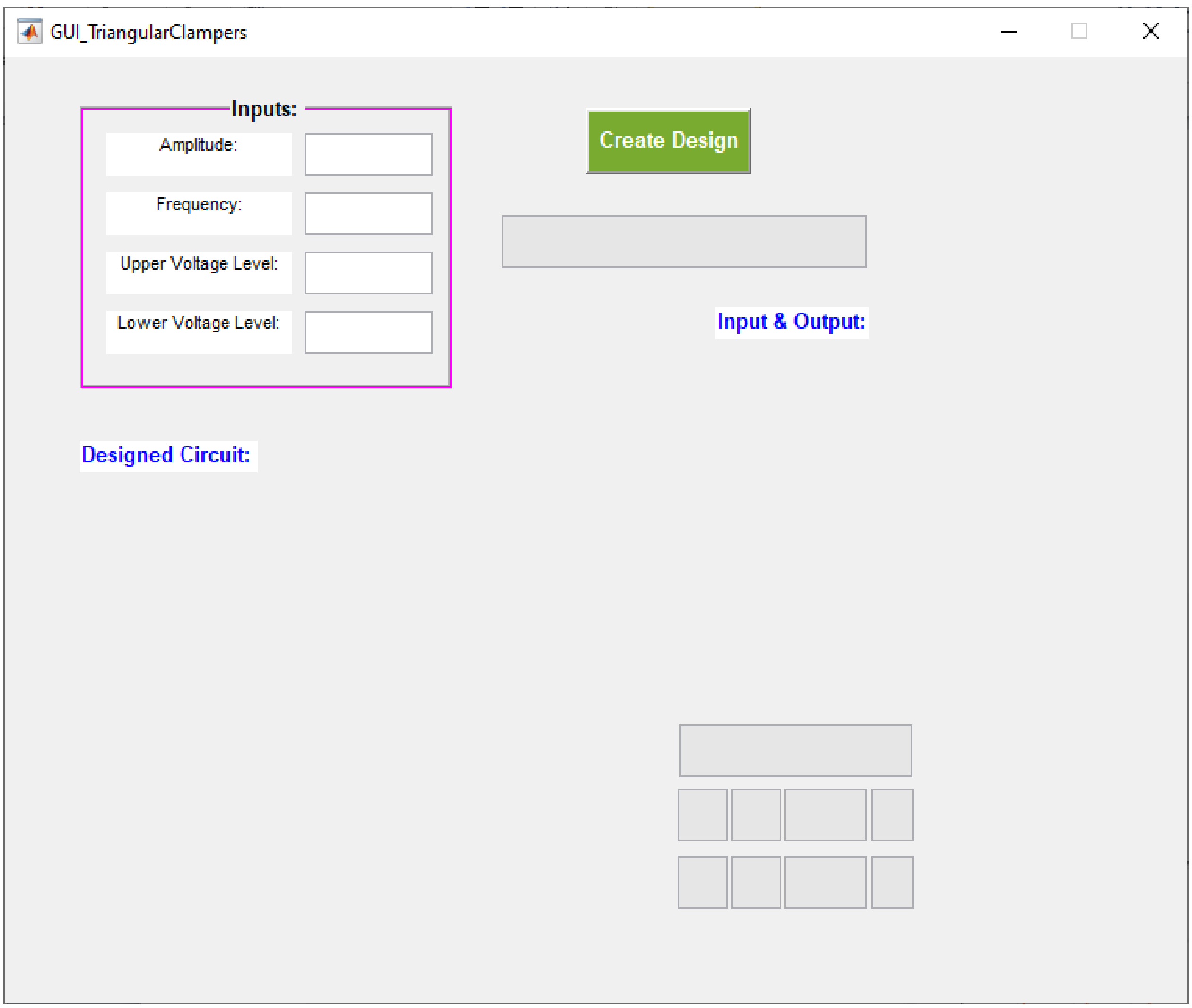
```
function [t,vin,vout,img,c1,v1]=SawtoothWave_Clamper_Design(vamp,f,vup,vlow)
   c1=10^-6;
   R=10^6;
   T=1/f;
   while 5*R*c1<=T/2
      c1=c1*10;
   end

   if vlow==0 && vup==2*vamp
      [t,vin,vout,img,v1]=positive_clamper(vamp,f,vup,vlow);
   elseif vlow==-2*vamp && vup==0
      [t,vin,vout,img,v1]=negative_clamper(vamp,f,vup,vlow);
   elseif vlow>0
      [t,vin,vout,img,v1]=positive_bias_positive_clamper(vamp,f,vup,vlow);
   elseif vlow<0 && vlow>(-vamp)
      [t,vin,vout,img,v1]=negative_bias_positive_clamper(vamp,f,vup,vlow);
   elseif vup>0 && vup<=vamp
      [t,vin,vout,img,v1]=positive_bias_negative_clamper(vamp,f,vup,vlow);
   elseif vup<0
      [t,vin,vout,img,v1]=negative_bias_negative_clamper(vamp,f,vup,vlow);
   end
end

function [t,vin,vout,img,v1]=positive_clamper(vamp,f,vup,vlow)
   v1=0;
   T=1/f;
   t=linspace(0,3*T,1200);
   vin=vamp.*sawtooth((2*pi*f)*t);
   vout(1:1200)=0;
   i=1;
   for i=1:1200
      vout(i)=vin(i)+vamp;
   end
   img=imread('SawtoothClamper1.PNG');
end

function [t,vin,vout,img,v1]=negative_clamper(vamp,f,vup,vlow)
   v1=0;
```

```
    T=1/f;
    t=linspace(0,3*T,1200);
    vin=vamp.*sawtooth((2*pi*f)*t);
    vout(1:1200)=0;
    i=1;
    while i<=1200
        if i<=400
            vout(i)=0;
        else
            vout(i)=-vamp+vin(i);
        end
        i=i+1;
    end
    img=imread('SawtoothClamper2.PNG');
end

function [t,vin,vout,img,v1]=positive_bias_positive_clamper(vamp,f,vup,vlow)
    v1=vlow;
    T=1/f;
    t=linspace(0,3*T,1200);
    vin=vamp.*sawtooth((2*pi*f)*t);
    vout(1:1200)=0;
    i=1;
    while i<=1200
        vout(i)=vamp+v1+vin(i);
        i=i+1;
    end
    img=imread('SawtoothClamper3.PNG');
end

function [t,vin,vout,img,v1]=negative_bias_positive_clamper(vamp,f,vup,vlow)
    v1=-vlow;
    T=1/f;
    t=linspace(0,3*T,1200);
    vin=vamp.*sawtooth((2*pi*f)*t);
    vout(1:1200)=0;
    i=1;
    while i<=1200
        vout(i)=vin(i)+vamp-v1;
        i=i+1;
```

```
    end
    img=imread('SawtoothClamper4.PNG');
end

function [t,vin,vout,img,v1]=positive_bias_negative_clamper(vamp,f,vup,vlow)
    v1=vup;
    T=1/f;
    t=linspace(0,3*T,1200);
    vin=vamp.*sawtooth((2*pi*f)*t);
    vout(1:1200)=0;
    i=1;
    while i<=1200
        if i<=200
            if vin(i)+vamp<=v1
                vout(i)=vin(i)+vamp;
            else
                vout(i)=v1;
            end
        elseif i>200 && i<=400
            vout(i)=v1;
        else
            vout(i)=vin(i)-vamp+v1;
        end
        i=i+1;
    end
    img=imread('SawtoothClamper5.PNG');
end

function [t,vin,vout,img,v1]=negative_bias_negative_clamper(vamp,f,vup,vlow)
    v1=-vup;
    T=1/f;
    t=linspace(0,3*T,1200);
    vin=vamp.*sawtooth((2*pi*f)*t);
    vout(1:1200)=0;
    i=1;
    while i<=1200
        if i<=400
            vout(i)=-v1;
        else
            vout(i)=vin(i)-vamp-v1;
```

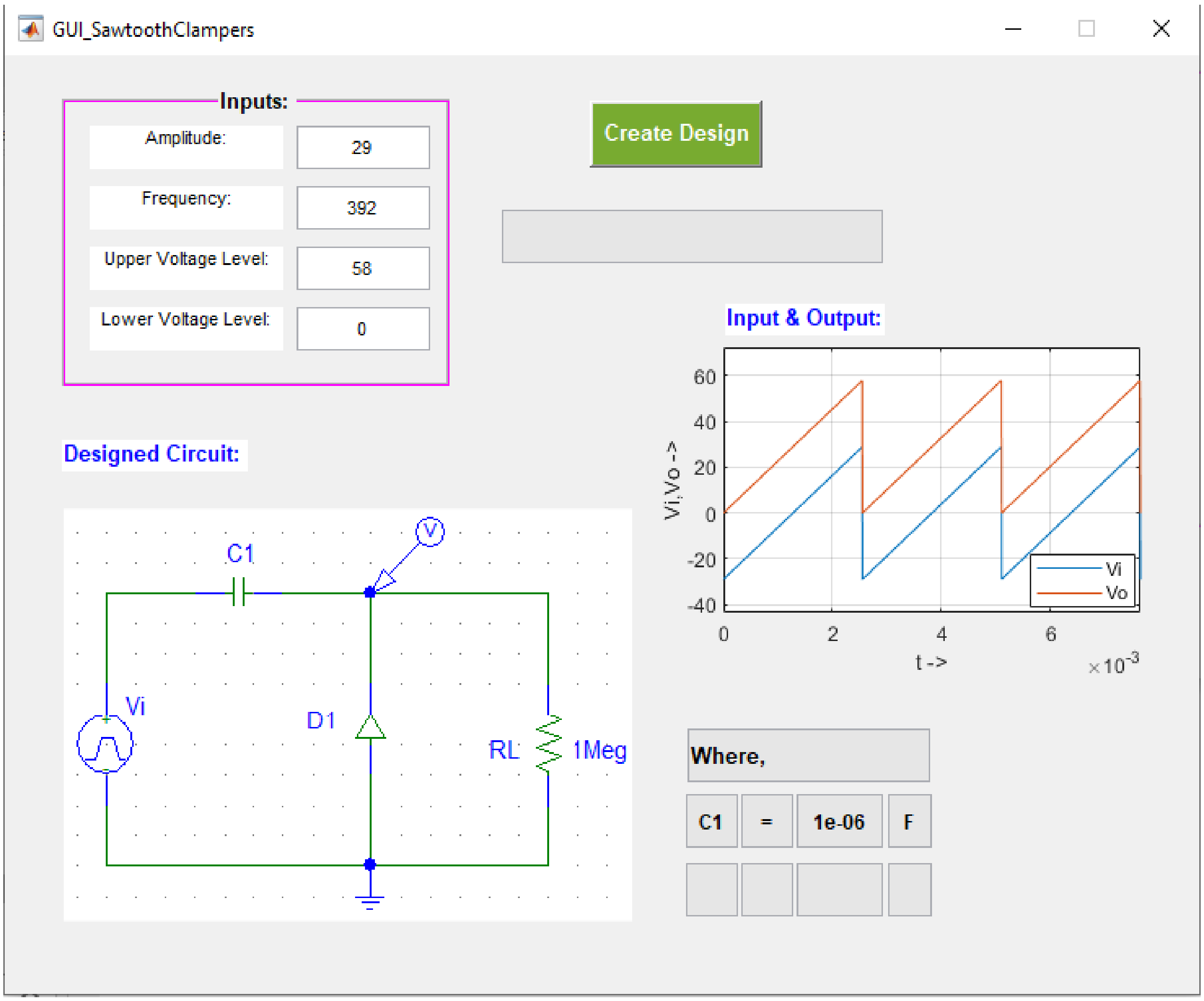
```
    end
    i=i+1;
  end
  img=imread('SawtoothClamper6.PNG');
end
```

Corresponding Window:

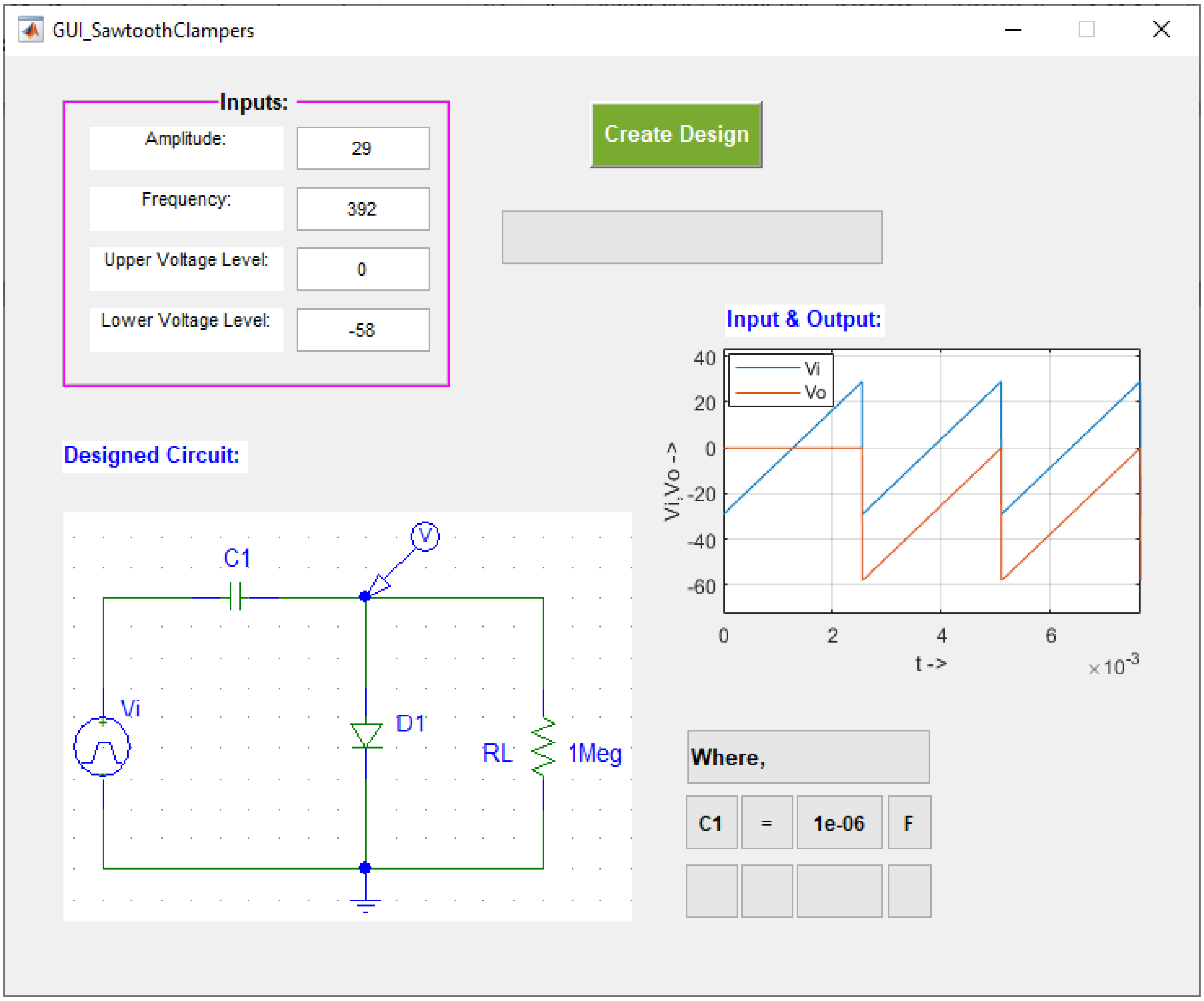


Sample Outputs:

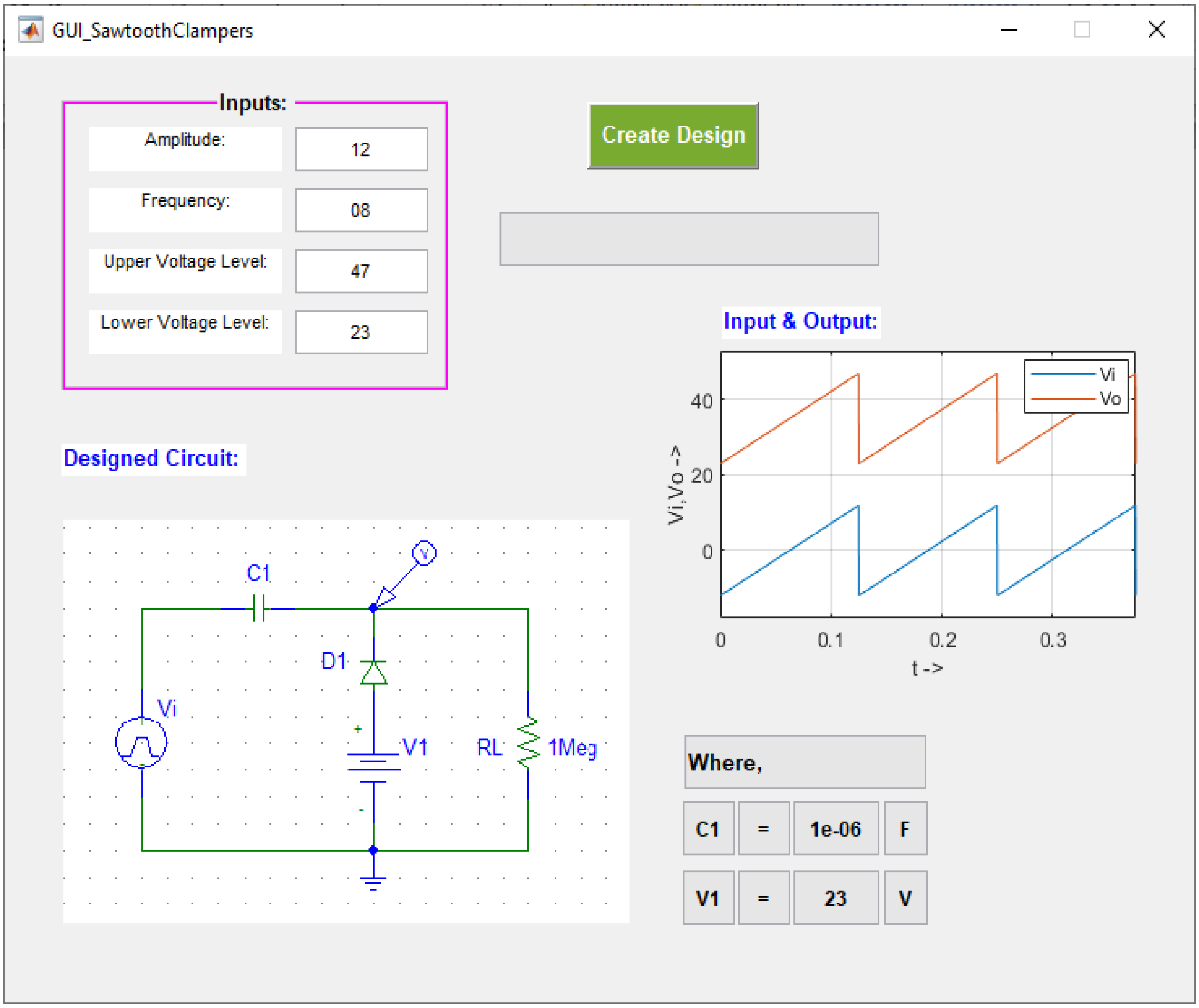
1. Positive Clamper :



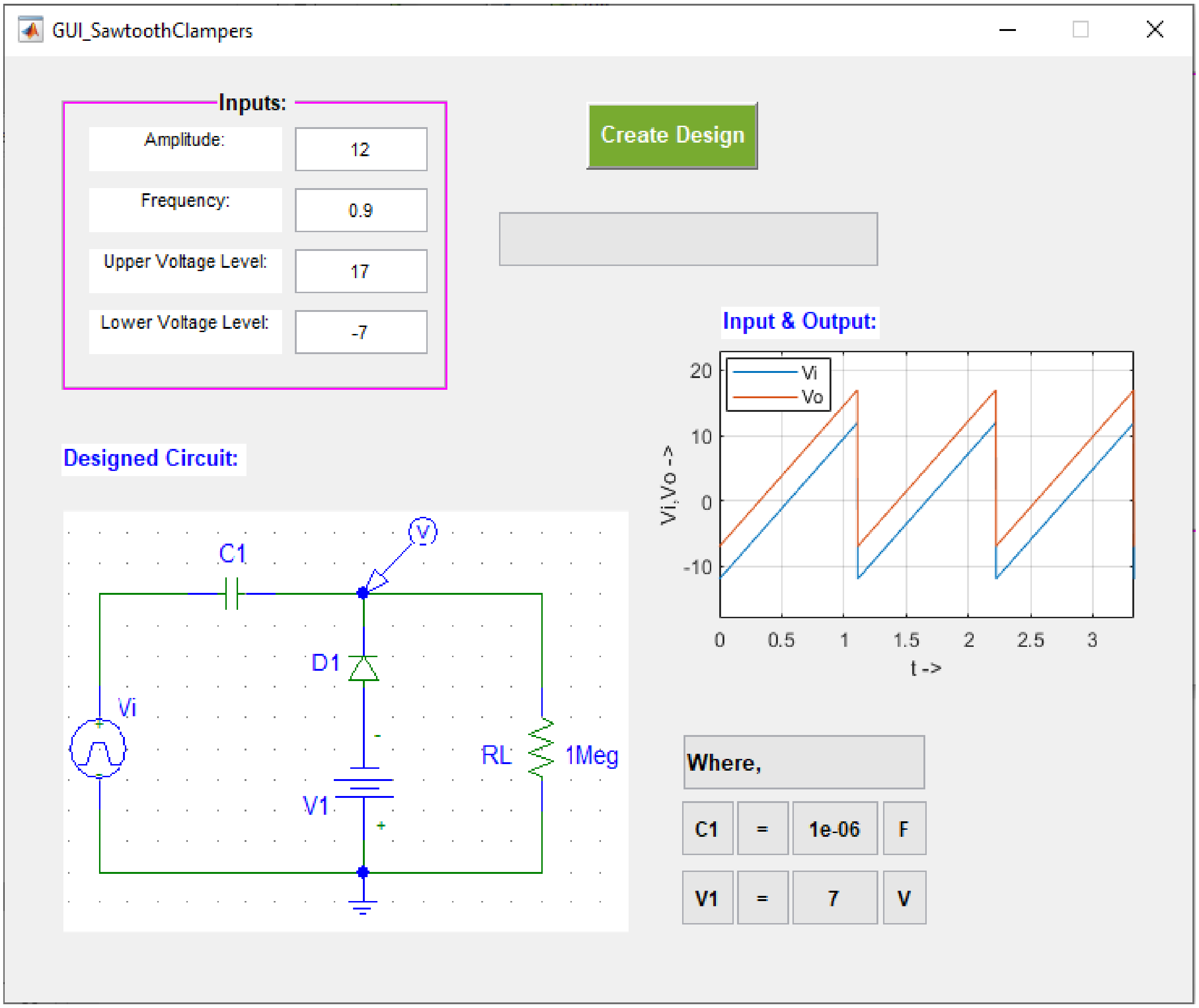
2. **Negative Clamper :**



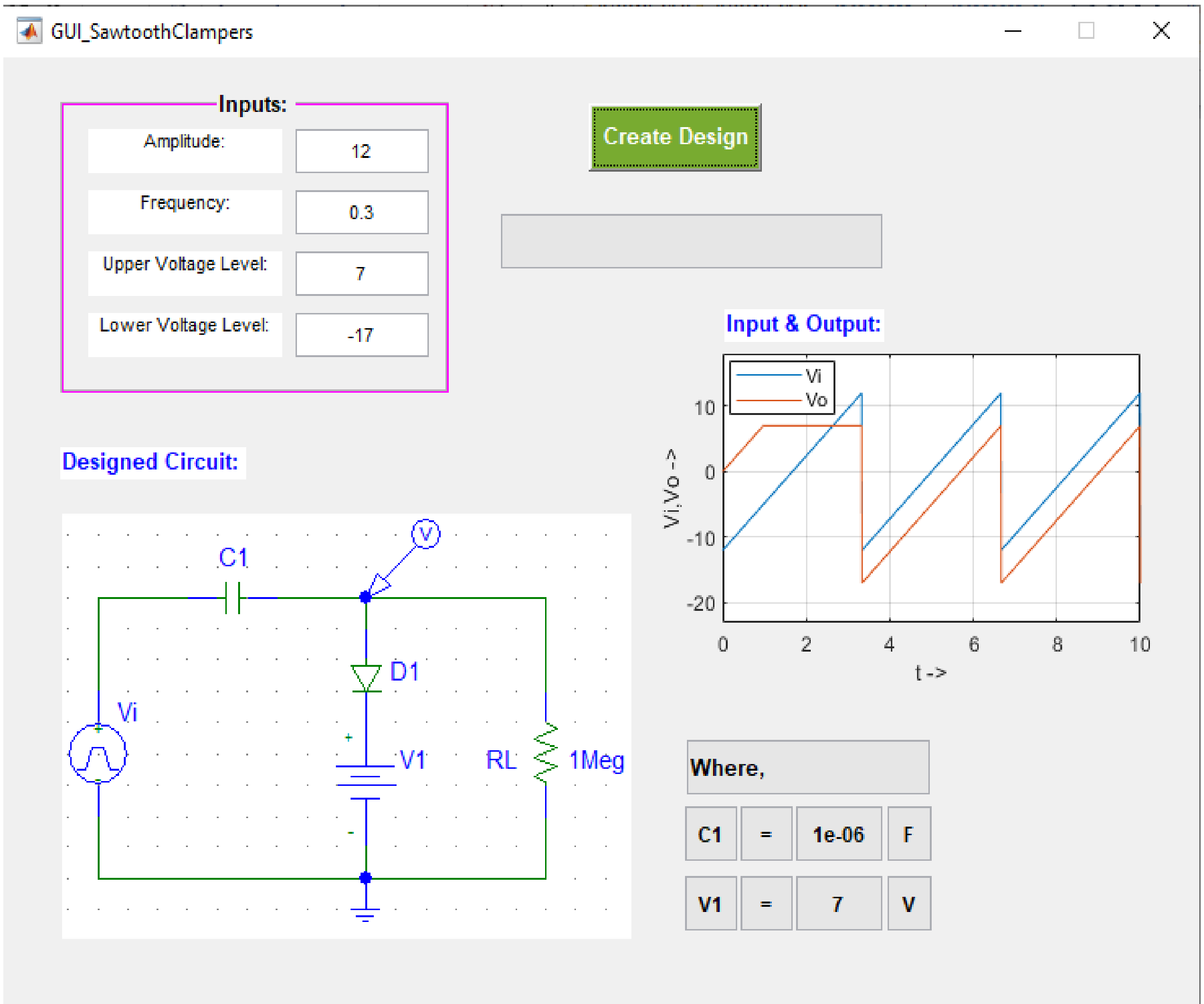
3. Positive Bias - Positive Clamper :



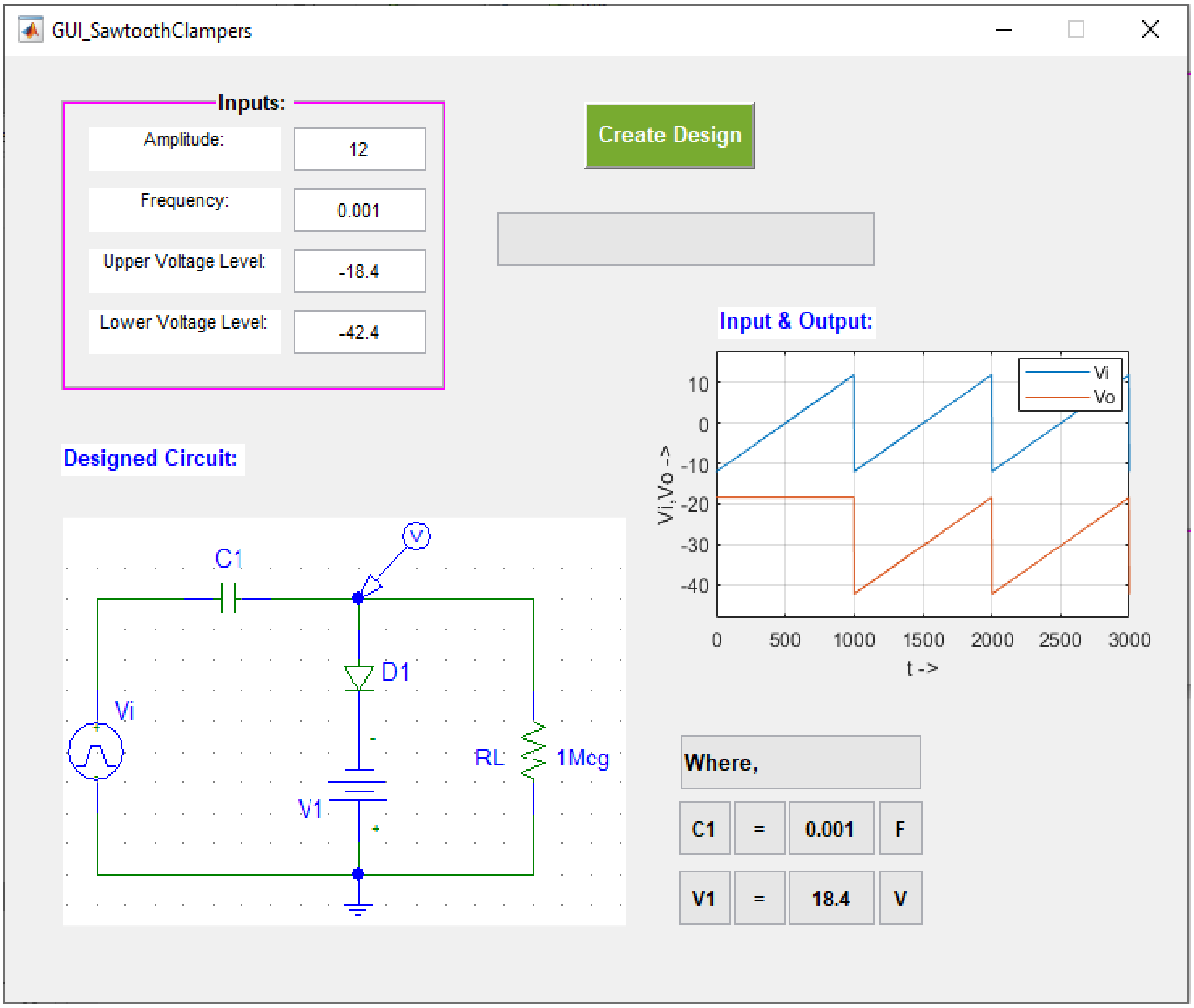
4. Negative Bias – Positive Clamper:



5. Positive Bias - Negative Clamper :

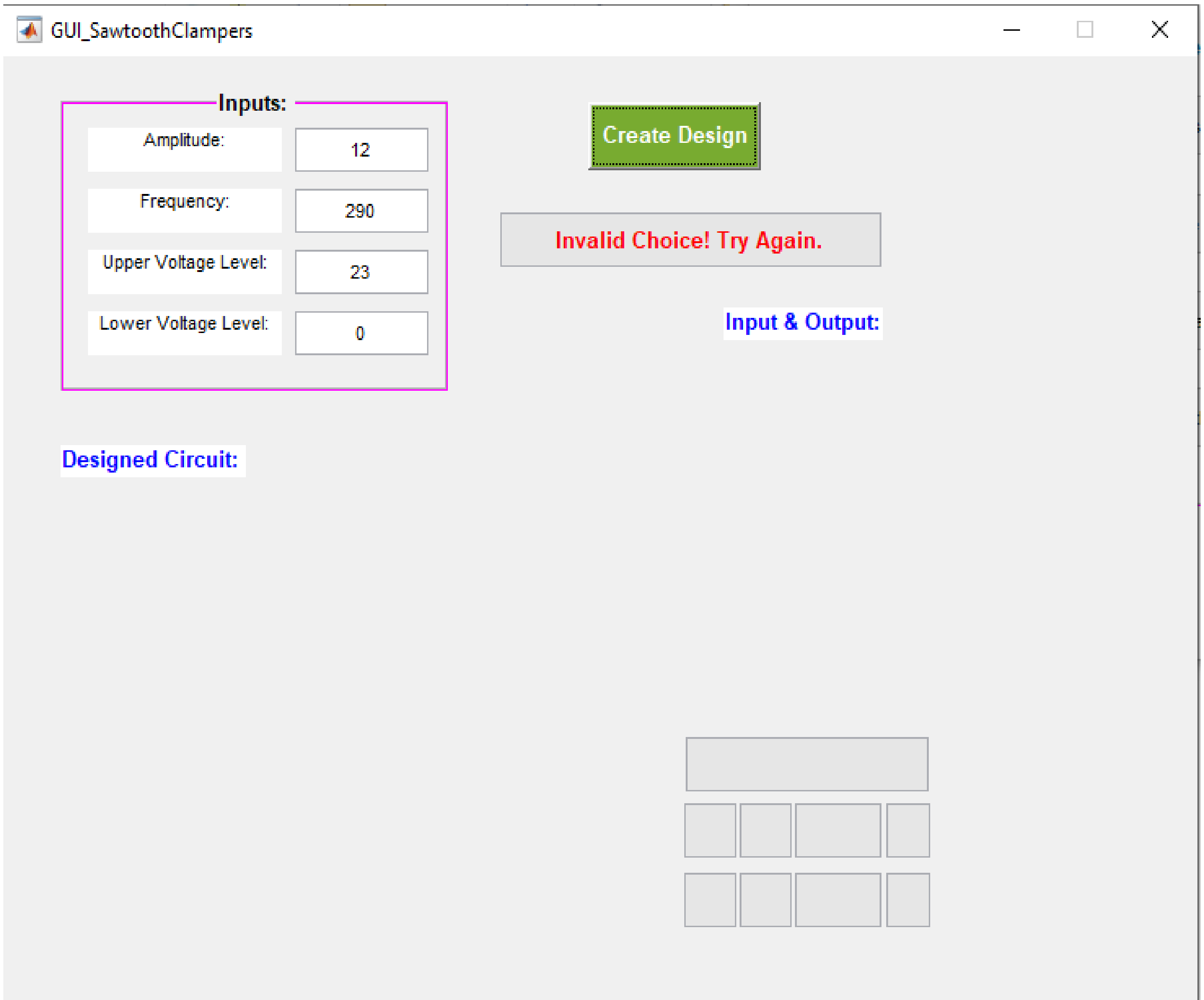


6. Negative Bias - Negative Clamper :

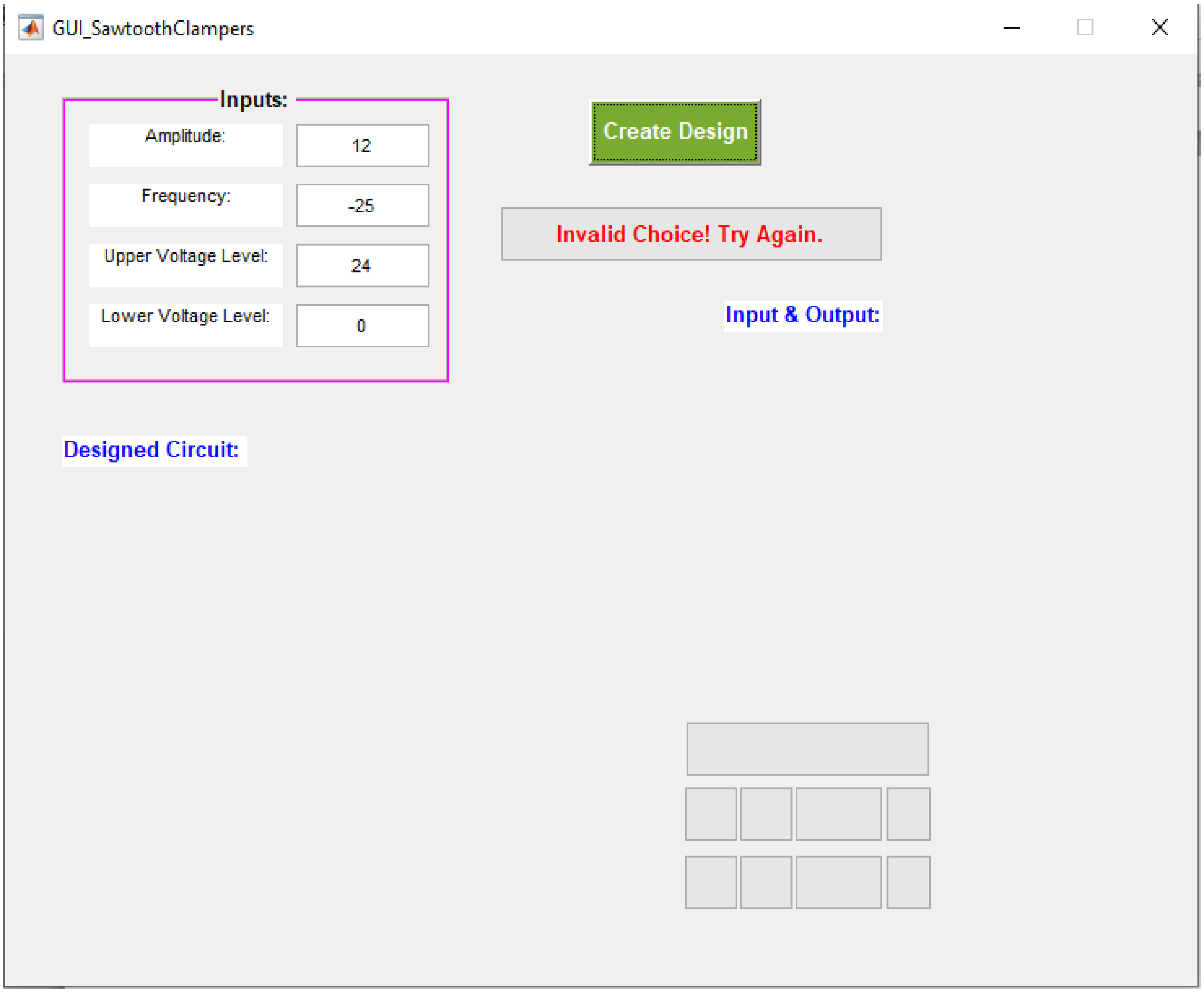


7. Invalid Cases:

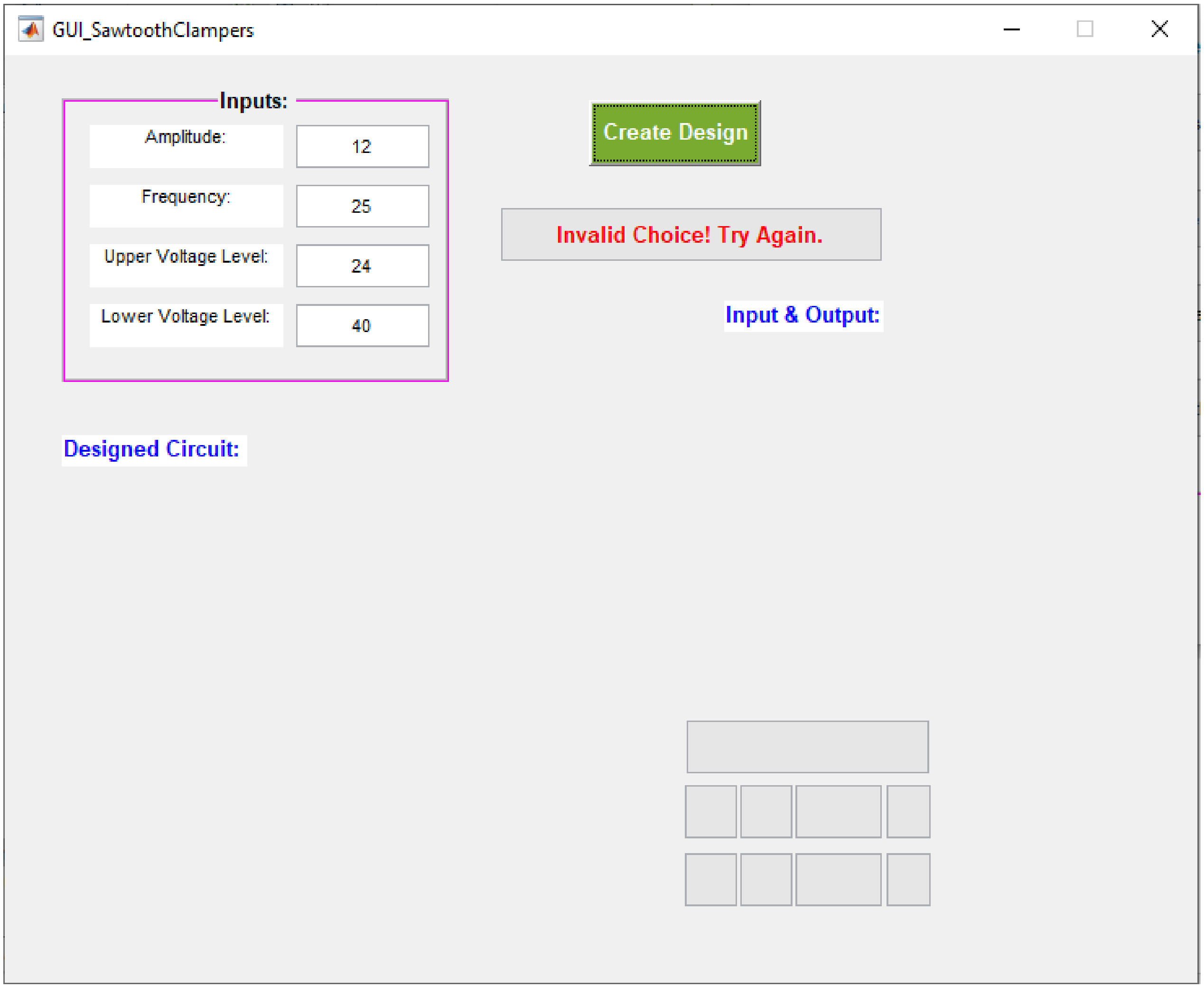
i. (Upper Voltage Level-Lower Voltage Level) $\neq$ (2 $\times$ Amplitude) :



ii. Zero/ Negative Frequency:



iii. Lower Voltage Level is given greater than Upper Voltage Level:



N. B: Our program is able to design the circuit for any other kind of inputs as well. User is appreciated to check for more as he desires.

3. DC Regulating:

- Now suppose, “DC Regulated” has been chosen from the home tab. It will direct us to “GUI_DCRegulator.m”.

Code:

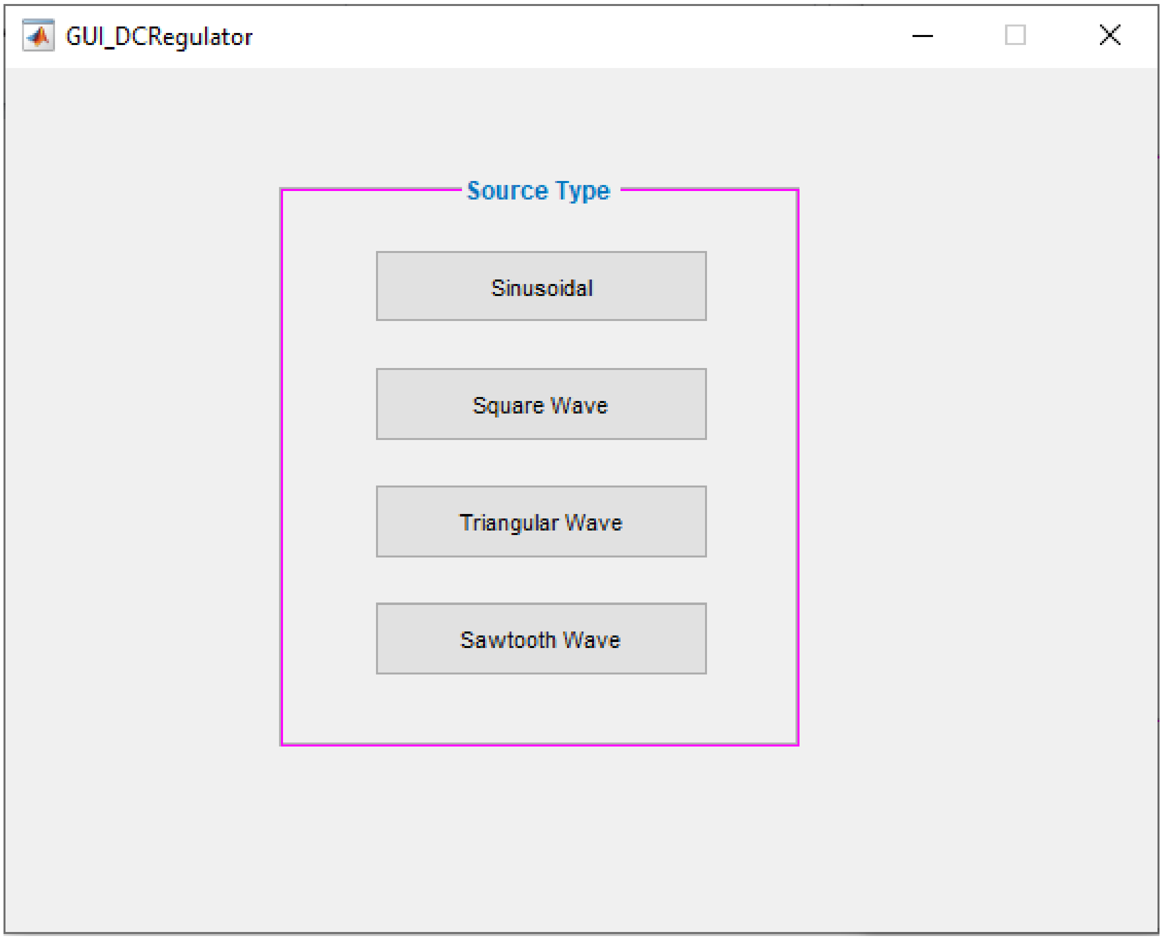
“GUI_DCRegulator.m” -

```
function varargout = GUI_DCRegulator(varargin)
% GUI_DCREGULATOR MATLAB code for GUI_DCRegulator.fig
%      GUI_DCREGULATOR, by itself, creates a new GUI_DCREGULATOR or raises the
existing
%      singleton*.
%
%      H = GUI_DCREGULATOR returns the handle to a new GUI_DCREGULATOR or
the handle to
%      the existing singleton*.
%
%      GUI_DCREGULATOR('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in GUI_DCREGULATOR.M with the given input
arguments.
%
%      GUI_DCREGULATOR('Property','Value',...) creates a new GUI_DCREGULATOR
or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before GUI_DCRegulator_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to GUI_DCRegulator_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help GUI_DCRegulator

% Last Modified by GUIDE v2.5 21-Jul-2021 04:58:57

% Begin initialization code - DO NOT EDIT
```

```
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @GUI_DCRegulator_OpeningFcn, ...
                   'gui_OutputFcn',  @GUI_DCRegulator_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);

if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before GUI_DCRegulator is made visible.
function GUI_DCRegulator_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to GUI_DCRegulator (see VARARGIN)

% Choose default command line output for GUI_DCRegulator
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes GUI_DCRegulator wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = GUI_DCRegulator_OutputFcn(hObject, eventdata, handles)
```

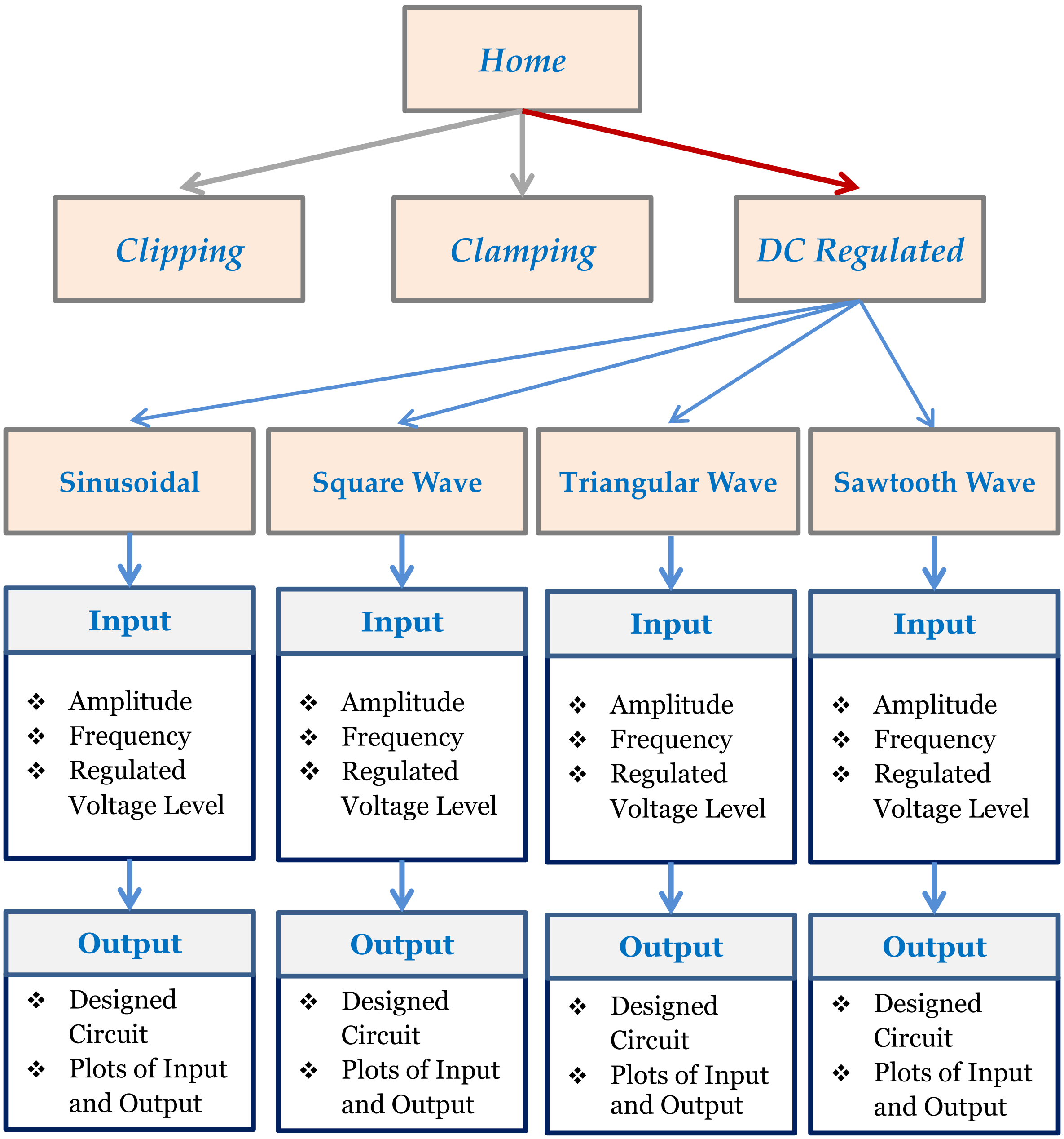
```
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


% --- Executes on button press in Sinusoidal.
function Sinusoidal_Callback(hObject, eventdata, handles)
% hObject    handle to Sinusoidal (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
GUI_SinusoidalRegulator


% --- Executes on button press in SquareWave.
function SquareWave_Callback(hObject, eventdata, handles)
% hObject    handle to SquareWave (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
GUI_SquareRegulator


% --- Executes on button press in TriangularWave.
function TriangularWave_Callback(hObject, eventdata, handles)
% hObject    handle to TriangularWave (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
GUI_TriangularRegulator


% --- Executes on button press in SawtoothWave.
function SawtoothWave_Callback(hObject, eventdata, handles)
% hObject    handle to SawtoothWave (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
GUI_SawtoothRegulator
```

Corresponding Window:



- We will work with all four types of sources one by one.

Flow Chart to Prform DC Regulation :



(i) **Sinusoidal DC Regulator:**

- For the time being, let us choose "Sinusoidal" source first.
- "GUI_SinusoidalRegulator.m" interacts with the user and calls "Sinusoidal_Regulator_Design.m" that contains all necessary functions and codes to create the design and sends data back.

Codes:

"GUI_SinusoidalRegulator.m" –

```
function varargout = GUI_SinusoidalRegulator(varargin)
% GUI_SINUSOIDALREGULATOR MATLAB code for GUI_SinusoidalRegulator.fig
%      GUI_SINUSOIDALREGULATOR, by itself, creates a new
GUI_SINUSOIDALREGULATOR or raises the existing
%      singleton*.
%
%      H = GUI_SINUSOIDALREGULATOR returns the handle to a new
GUI_SINUSOIDALREGULATOR or the handle to
%      the existing singleton*.
%
%      GUI_SINUSOIDALREGULATOR('CALLBACK',hObject,eventData,handles,...)
calls the local
%      function named CALLBACK in GUI_SINUSOIDALREGULATOR.M with the
given input arguments.
%
%      GUI_SINUSOIDALREGULATOR('Property','Value',...) creates a new
GUI_SINUSOIDALREGULATOR or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before GUI_SinusoidalRegulator_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to GUI_SinusoidalRegulator_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help GUI_SinusoidalRegulator
```

```
% Last Modified by GUIDE v2.5 23-Jul-2021 19:18:12

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @GUI_SinusoidalRegulator_OpeningFcn, ...
                   'gui_OutputFcn',  @GUI_SinusoidalRegulator_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before GUI_SinusoidalRegulator is made visible.
function GUI_SinusoidalRegulator_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to GUI_SinusoidalRegulator (see VARARGIN)

% Choose default command line output for GUI_SinusoidalRegulator
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes GUI_SinusoidalRegulator wait for user response (see UIRESUME)
% uiwait(handles.figure1);
```

```
% --- Outputs from this function are returned to the command line.
function varargout = GUI_SinusoidalRegulator_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;



% --- Executes on button press in design.
function design_Callback(hObject, eventdata, handles)
% hObject    handle to design (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.invalidcheck,'String','');
set(handles.properties,'String','');
set(handles.v11,'String','');
set(handles.v21,'String','');
set(handles.v22,'String','');
set(handles.v23,'String','');
set(handles.v24,'String','');
axes(handles.diagram);
cla;
axes(handles.plots);
cla;

amp=str2double(get(handles.amplitude,'String'));
f=str2double(get(handles.frequency,'String'));
dc=str2double(get(handles.DClevel,'String'));

R=10^3;
C=470*10^-6;
ripple=amp/(2*f*R*C);

if  f<=0 || dc<0 || dc>amp
    set(handles.invalidcheck,'String','Invalid Choice! Try Again.');
elseif dc>amp-ripple && dc<=amp
```

```
    set(handles.invalidcheck,'String','Invalid Choice! In between Ripple.');
else
    [t,vin,vout,img,Vz]=Sinusoidal_Regulator_Design(amp,f,dc);
    axes(handles.plots);
    cla;
    plot(t,vin);
    hold on;
    plot(t,vout);
    axis([0,t(end),(-amp-amp/5),(amp+amp/5)]);
    legend('Vi','Vo');
    xlabel('t ->');
    ylabel('Vi,Vo ->');
    grid on;
    axes(handles.diagram);
    cla;
    imshow(img);

    set(handles.properties,'String','Where,');
    set(handles.v11,'String','Zener Voltage of Dz :');
    set(handles.v21,'String','Vz');
    set(handles.v22,'String','=');
    set(handles.v23,'String',num2str(Vz));
    set(handles.v24,'String','V');
end


function properties_Callback(hObject, eventdata, handles)
% hObject    handle to properties (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of properties as text
%        str2double(get(hObject,'String')) returns contents of properties as a double


% --- Executes during object creation, after setting all properties.
function properties_CreateFcn(hObject, eventdata, handles)
% hObject    handle to properties (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function v11_Callback(hObject, eventdata, handles)
% hObject    handle to v11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v11 as text
%        str2double(get(hObject,'String')) returns contents of v11 as a double


% --- Executes during object creation, after setting all properties.
function v11_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function v12_Callback(hObject, eventdata, handles)
% hObject    handle to v12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v12 as text
%        str2double(get(hObject,'String')) returns contents of v12 as a double
```

```
% --- Executes during object creation, after setting all properties.
function v12_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function v13_Callback(hObject, eventdata, handles)
% hObject    handle to v13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v13 as text
%        str2double(get(hObject,'String')) returns contents of v13 as a double


% --- Executes during object creation, after setting all properties.
function v13_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function v21_Callback(hObject, eventdata, handles)
% hObject    handle to v21 (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v21 as text
%        str2double(get(hObject,'String')) returns contents of v21 as a double


% --- Executes during object creation, after setting all properties.
function v21_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v21 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function v22_Callback(hObject, eventdata, handles)
% hObject    handle to v22 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v22 as text
%        str2double(get(hObject,'String')) returns contents of v22 as a double


% --- Executes during object creation, after setting all properties.
function v22_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v22 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
```

```
    set(hObject,'BackgroundColor','white');
end


function v23_Callback(hObject, eventdata, handles)
% hObject    handle to v23 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v23 as text
%        str2double(get(hObject,'String')) returns contents of v23 as a double


% --- Executes during object creation, after setting all properties.
function v23_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v23 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function invalidcheck_Callback(hObject, eventdata, handles)
% hObject    handle to invalidcheck (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of invalidcheck as text
%        str2double(get(hObject,'String')) returns contents of invalidcheck as a double


% --- Executes during object creation, after setting all properties.
function invalidcheck_CreateFcn(hObject, eventdata, handles)
% hObject    handle to invalidcheck (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function v14_Callback(hObject, eventdata, handles)
% hObject    handle to v14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v14 as text
%        str2double(get(hObject,'String')) returns contents of v14 as a double



% --- Executes during object creation, after setting all properties.
function v14_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function v24_Callback(hObject, eventdata, handles)
% hObject    handle to v24 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v24 as text
%        str2double(get(hObject,'String')) returns contents of v24 as a double
```

```
% --- Executes during object creation, after setting all properties.
function v24_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v24 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function amplitude_Callback(hObject, eventdata, handles)
% hObject    handle to amplitude (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of amplitude as text
%        str2double(get(hObject,'String')) returns contents of amplitude as a double


% --- Executes during object creation, after setting all properties.
function amplitude_CreateFcn(hObject, eventdata, handles)
% hObject    handle to amplitude (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function frequency_Callback(hObject, eventdata, handles)
% hObject    handle to frequency (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

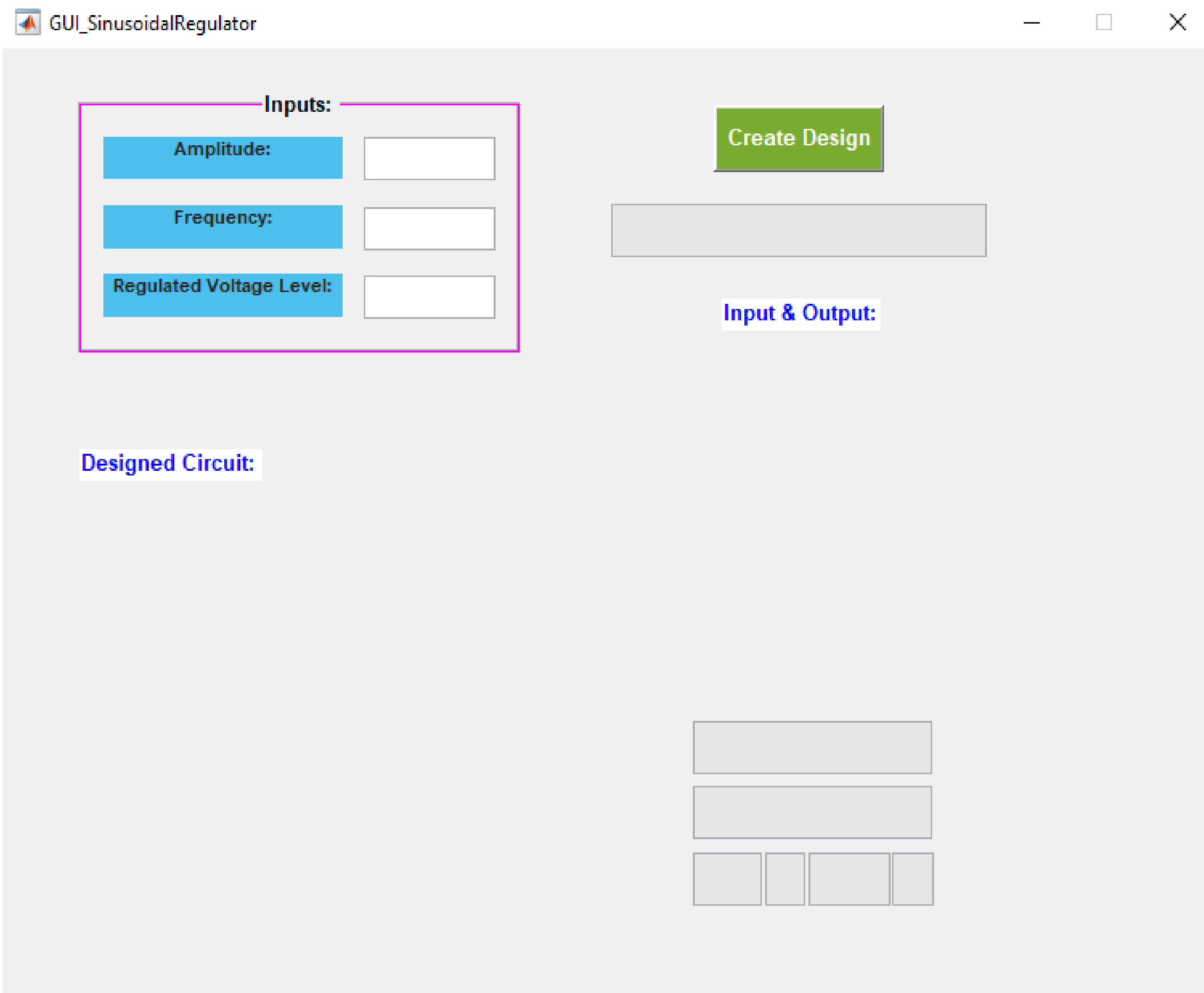
```
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of frequency as text
%        str2double(get(hObject,'String')) returns contents of frequency as a double


% --- Executes during object creation, after setting all properties.
function frequency_CreateFcn(hObject, eventdata, handles)
% hObject    handle to frequency (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function DClevel_Callback(hObject, eventdata, handles)
% hObject    handle to DClevel (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of DClevel as text
%        str2double(get(hObject,'String')) returns contents of DClevel as a double


% --- Executes during object creation, after setting all properties.
function DClevel_CreateFcn(hObject, eventdata, handles)
% hObject    handle to DClevel (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function edit4_Callback(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit4 as text
%        str2double(get(hObject,'String')) returns contents of edit4 as a double


% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), 
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

“Sinusoidal_Regulator_Design.m” –

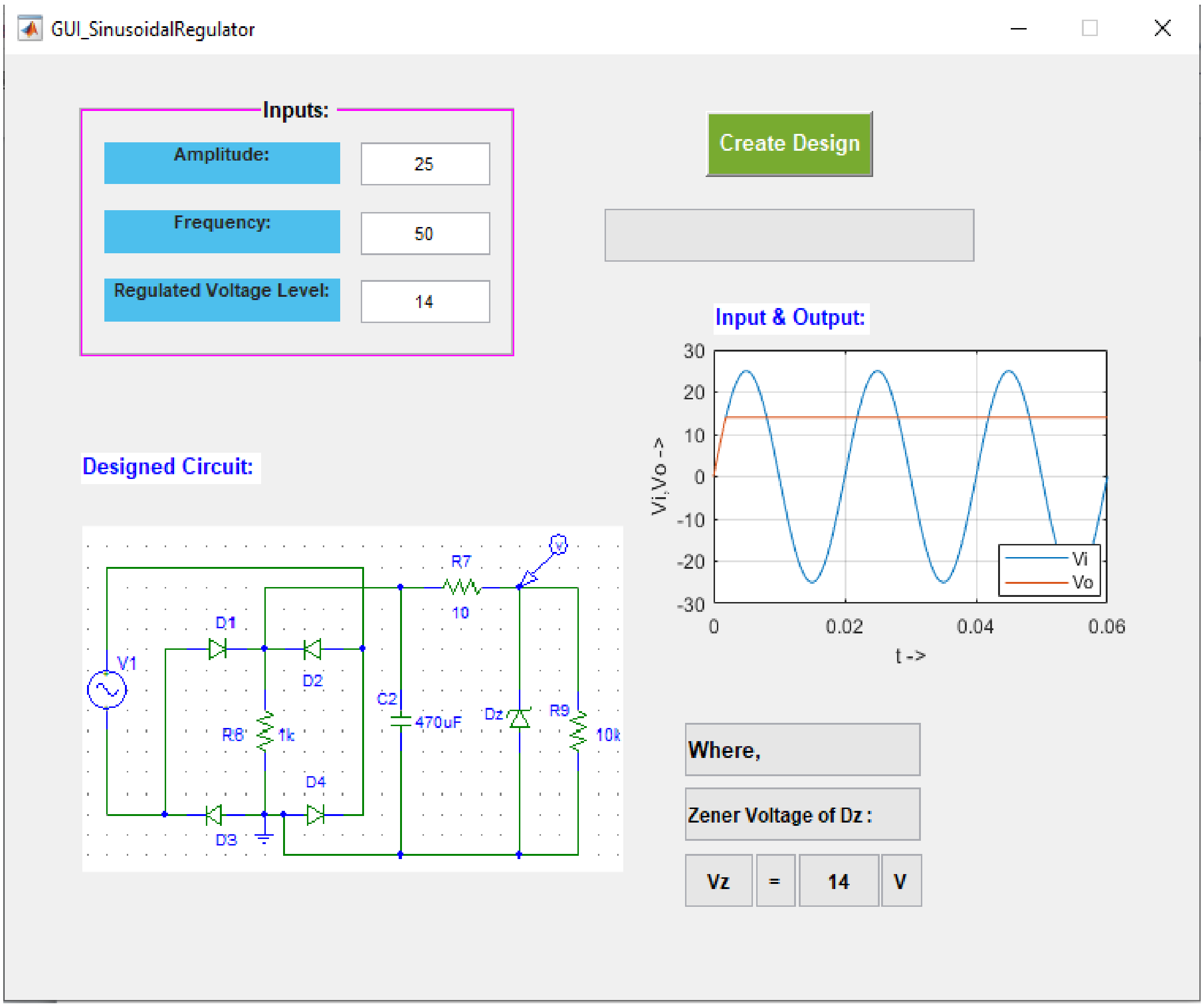
```
function [t,vin,vout,img,Vz]=Sinusoidal_Regulator_Design(amp,f,dc)
    Vz=dc;
    T=1/f;
    t=linspace(0,3*T,1200);
    vin=amp.*sin(2*pi*f*t);
    vout(1:1200)=0;
    for i=1:1200
        if i<=100 && vin(i)<=Vz
            vout(i)=vin(i);
        else
            vout(i)=Vz;
        end
    end
    img=imread('SinusoidalRegulator.PNG');
end
```

Corresponding Window:



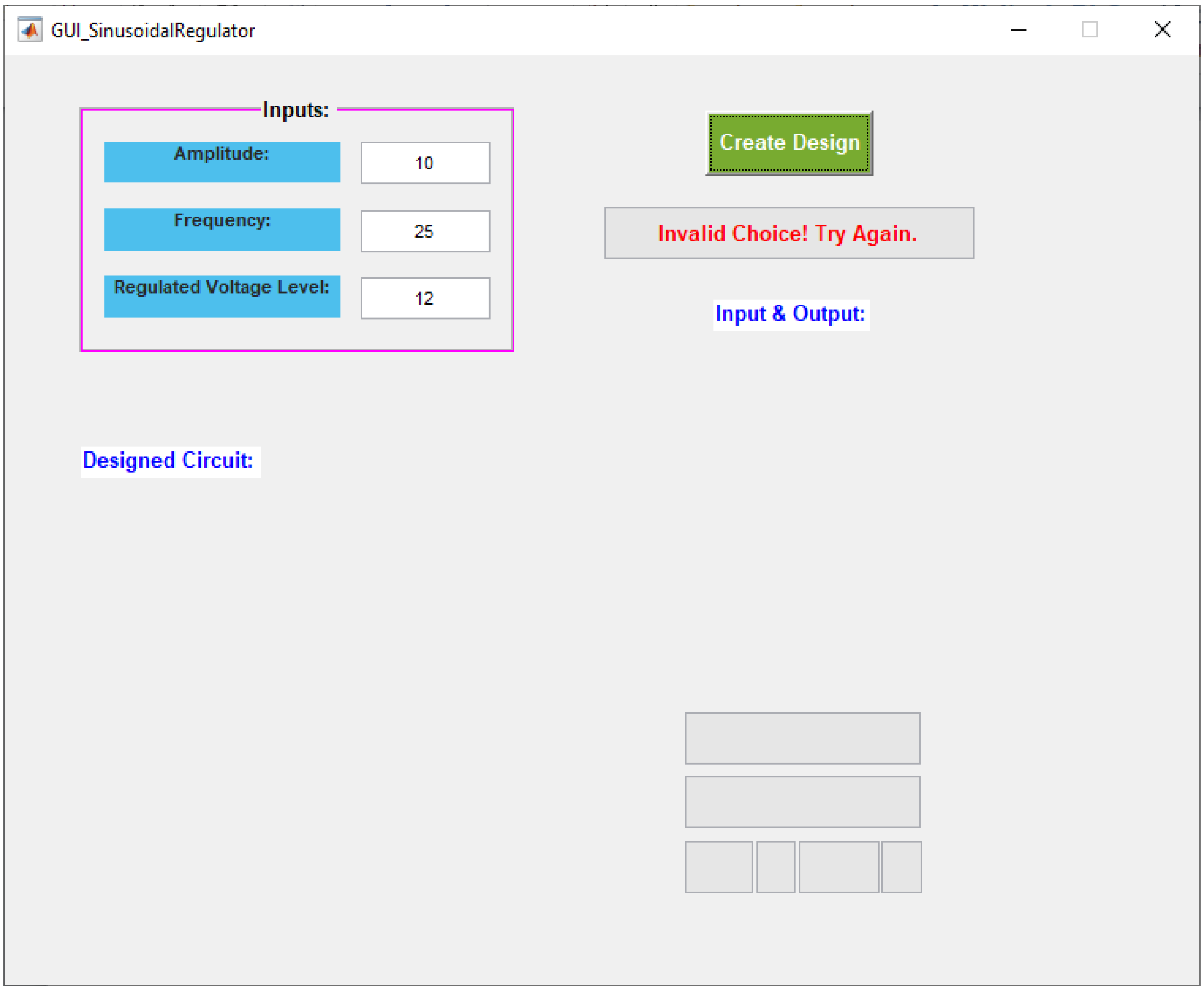
Sample Outputs:

1. Valid Inputs :

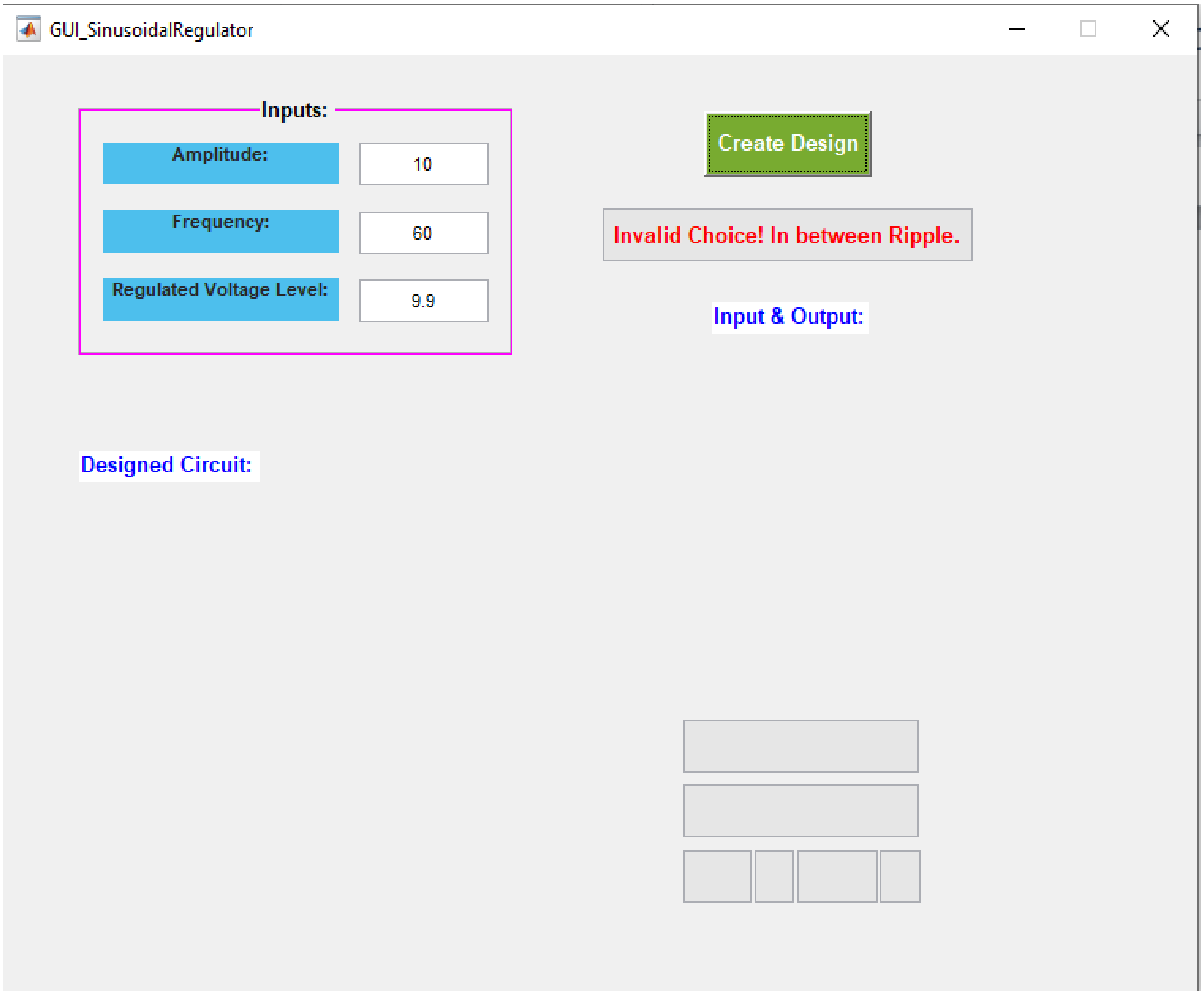


2. Invalid Cases:

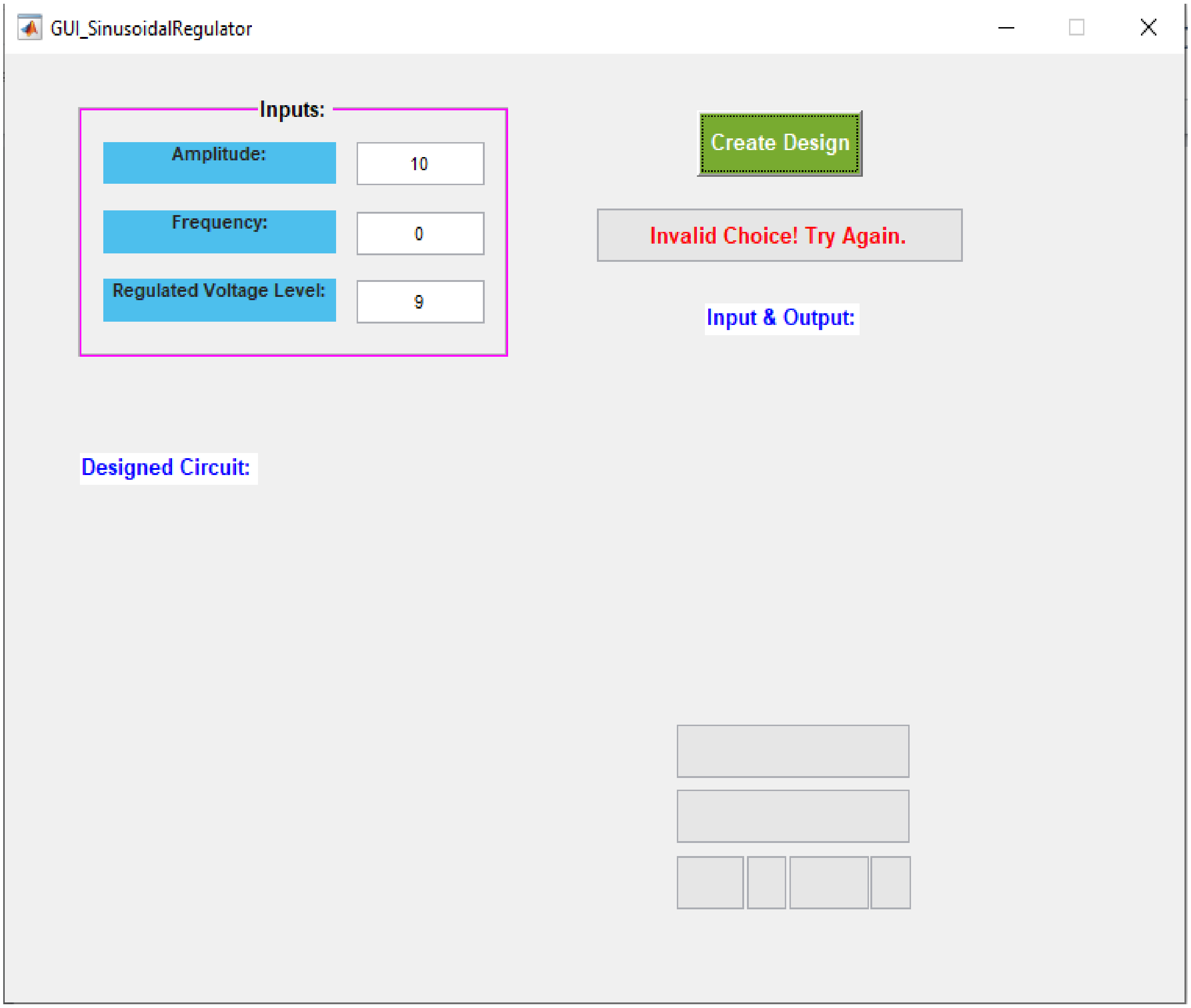
i. Demanded Regulated Voltage Level > Input Voltage Peak :



ii. Demanded Regulated Voltage Level is in between the Ripple :



iii. Zero/ Negative Frequency:



N. B: Our program is able to design the circuit for any other kind of inputs as well. User is appreciated to check for more as he desires.

(ii) **Square Wave DC Regulator:**

- Now, let us choose "Square Wave" from "GUI_DCRegulator" window.
- "GUI_SquareRegulator.m" interacts with the user and calls "SquareWave_Regulator_Design.m" that contains all necessary functions and codes to create the design and sends data back.

Codes:

"GUI_SquareRegulator.m" –

```
function varargout = GUI_SquareRegulator(varargin)
% GUI_SQUAREREGULATOR MATLAB code for GUI_SquareRegulator.fig
%      GUI_SQUAREREGULATOR, by itself, creates a new GUI_SQUAREREGULATOR
or raises the existing
%      singleton*.
%
%      H = GUI_SQUAREREGULATOR returns the handle to a new
GUI_SQUAREREGULATOR or the handle to
%      the existing singleton*.
%
%      GUI_SQUAREREGULATOR('CALLBACK',hObject,eventData,handles,...) calls the
local
%      function named CALLBACK in GUI_SQUAREREGULATOR.M with the given
input arguments.
%
%      GUI_SQUAREREGULATOR('Property','Value',...) creates a new
GUI_SQUAREREGULATOR or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before GUI_SquareRegulator_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to GUI_SquareRegulator_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help GUI_SquareRegulator
```

```
% Last Modified by GUIDE v2.5 26-Jul-2021 21:18:30

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @GUI_SquareRegulator_OpeningFcn, ...
                   'gui_OutputFcn',  @GUI_SquareRegulator_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before GUI_SquareRegulator is made visible.
function GUI_SquareRegulator_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to GUI_SquareRegulator (see VARARGIN)

% Choose default command line output for GUI_SquareRegulator
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes GUI_SquareRegulator wait for user response (see UIRESUME)
% uiwait(handles.figure1);
```

```
% --- Outputs from this function are returned to the command line.
function varargout = GUI_SquareRegulator_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


% --- Executes on button press in design.
function design_Callback(hObject, eventdata, handles)
% hObject    handle to design (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.invalidcheck,'String','');
set(handles.properties,'String','');
set(handles.v11,'String','');
set(handles.v21,'String','');
set(handles.v22,'String','');
set(handles.v23,'String','');
set(handles.v24,'String','');
axes(handles.diagram);
cla;
axes(handles.plots);
cla;

amp=str2double(get(handles.amplitude,'String'));
f=str2double(get(handles.frequency,'String'));
dc=str2double(get(handles.DClevel,'String'));

R=10^3;
C=470*10^-6;
ripple=amp/(2*f*R*C);

if  f<=0 || dc<0 || dc>amp
    set(handles.invalidcheck,'String','Invalid Choice! Try Again.');
elseif dc>amp-ripple && dc<=amp
    set(handles.invalidcheck,'String','Invalid Choice! In between Ripple.');
```

```
else
    [t,vin,vout,img,Vz]=SquareWave_Regulator_Design(amp,f,dc);
    axes(handles.plots);
    cla;
    plot(t,vin);
    hold on;
    plot(t,vout);
    axis([0,t(end),(-amp-amp/5),(amp+amp/5)]);
    legend('Vi','Vo');
    xlabel('t ->');
    ylabel('Vi,Vo ->');
    grid on;
    axes(handles.diagram);
    cla;
    imshow(img);

    set(handles.properties,'String','Where,');
    set(handles.v11,'String','Zener Voltage of Dz :');
    set(handles.v21,'String','Vz');
    set(handles.v22,'String','=');
    set(handles.v23,'String',num2str(Vz));
    set(handles.v24,'String','V');

end


function properties_Callback(hObject, eventdata, handles)
% hObject    handle to properties (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of properties as text
%        str2double(get(hObject,'String')) returns contents of properties as a double


% --- Executes during object creation, after setting all properties.
function properties_CreateFcn(hObject, eventdata, handles)
% hObject    handle to properties (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function v11_Callback(hObject, eventdata, handles)
% hObject    handle to v11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v11 as text
%        str2double(get(hObject,'String')) returns contents of v11 as a double


% --- Executes during object creation, after setting all properties.
function v11_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function v12_Callback(hObject, eventdata, handles)
% hObject    handle to v12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v12 as text
%        str2double(get(hObject,'String')) returns contents of v12 as a double
```

```
% --- Executes during object creation, after setting all properties.
function v12_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function v13_Callback(hObject, eventdata, handles)
% hObject    handle to v13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v13 as text
%        str2double(get(hObject,'String')) returns contents of v13 as a double


% --- Executes during object creation, after setting all properties.
function v13_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function v21_Callback(hObject, eventdata, handles)
% hObject    handle to v21 (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v21 as text
%        str2double(get(hObject,'String')) returns contents of v21 as a double


% --- Executes during object creation, after setting all properties.
function v21_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v21 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function v22_Callback(hObject, eventdata, handles)
% hObject    handle to v22 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v22 as text
%        str2double(get(hObject,'String')) returns contents of v22 as a double


% --- Executes during object creation, after setting all properties.
function v22_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v22 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
```

```
    set(hObject,'BackgroundColor','white');
end


function v23_Callback(hObject, eventdata, handles)
% hObject    handle to v23 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v23 as text
%        str2double(get(hObject,'String')) returns contents of v23 as a double


% --- Executes during object creation, after setting all properties.
function v23_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v23 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function invalidcheck_Callback(hObject, eventdata, handles)
% hObject    handle to invalidcheck (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of invalidcheck as text
%        str2double(get(hObject,'String')) returns contents of invalidcheck as a double


% --- Executes during object creation, after setting all properties.
function invalidcheck_CreateFcn(hObject, eventdata, handles)
% hObject    handle to invalidcheck (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function v14_Callback(hObject, eventdata, handles)
% hObject    handle to v14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v14 as text
%        str2double(get(hObject,'String')) returns contents of v14 as a double


% --- Executes during object creation, after setting all properties.
function v14_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function v24_Callback(hObject, eventdata, handles)
% hObject    handle to v24 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v24 as text
%        str2double(get(hObject,'String')) returns contents of v24 as a double
```

```
% --- Executes during object creation, after setting all properties.
function v24_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v24 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function amplitude_Callback(hObject, eventdata, handles)
% hObject    handle to amplitude (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of amplitude as text
%        str2double(get(hObject,'String')) returns contents of amplitude as a double


% --- Executes during object creation, after setting all properties.
function amplitude_CreateFcn(hObject, eventdata, handles)
% hObject    handle to amplitude (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function frequency_Callback(hObject, eventdata, handles)
% hObject    handle to frequency (see GCBO)
```

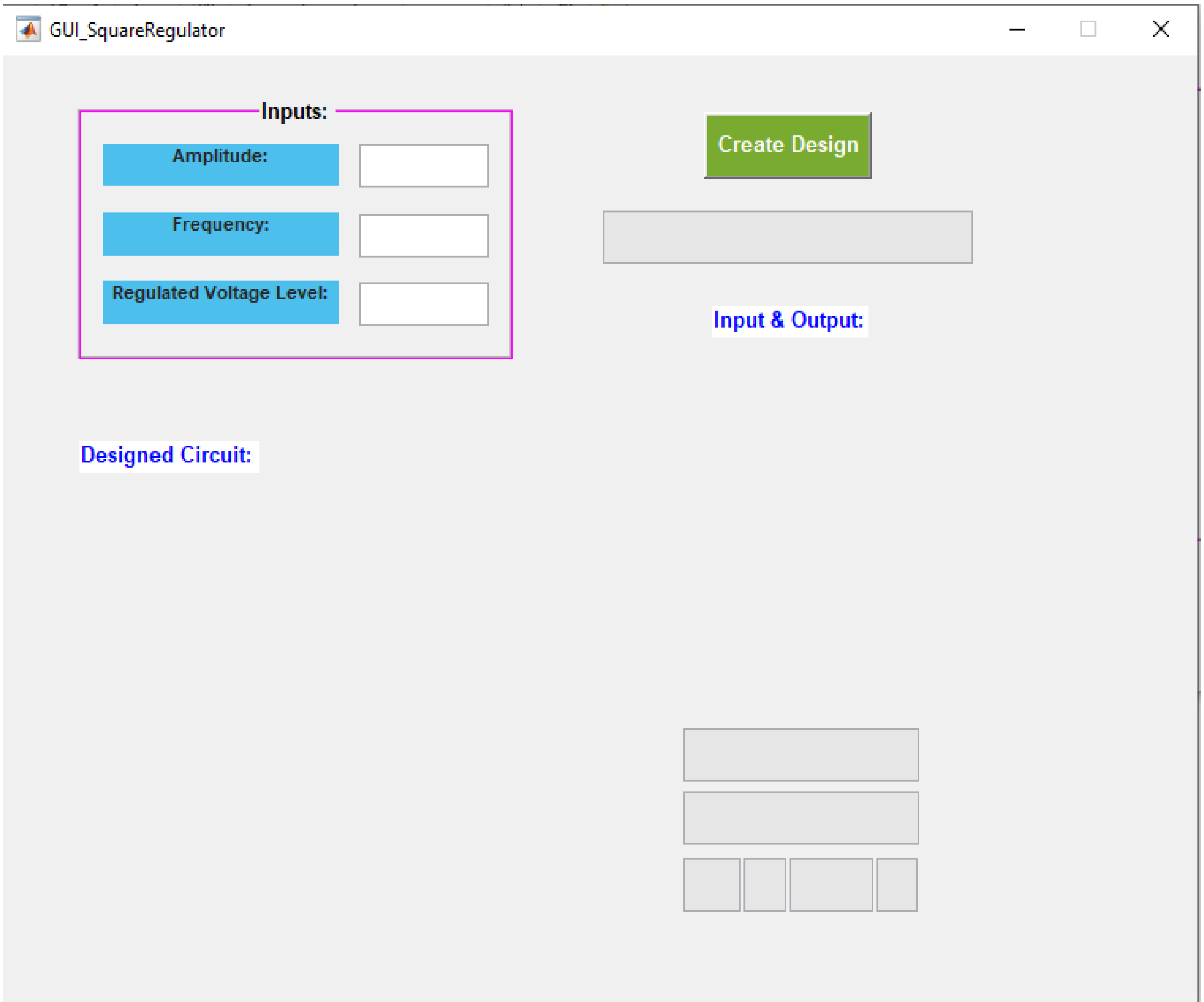
```
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of frequency as text
%        str2double(get(hObject,'String')) returns contents of frequency as a double


% --- Executes during object creation, after setting all properties.
function frequency_CreateFcn(hObject, eventdata, handles)
% hObject    handle to frequency (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function DClevel_Callback(hObject, eventdata, handles)
% hObject    handle to DClevel (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of DClevel as text
%        str2double(get(hObject,'String')) returns contents of DClevel as a double


% --- Executes during object creation, after setting all properties.
function DClevel_CreateFcn(hObject, eventdata, handles)
% hObject    handle to DClevel (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
```

```
    set(hObject,'BackgroundColor','white');
end



function edit4_Callback(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit4 as text
%        str2double(get(hObject,'String')) returns contents of edit4 as a double


% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

“SquareWave_Regulator_Design.m” –

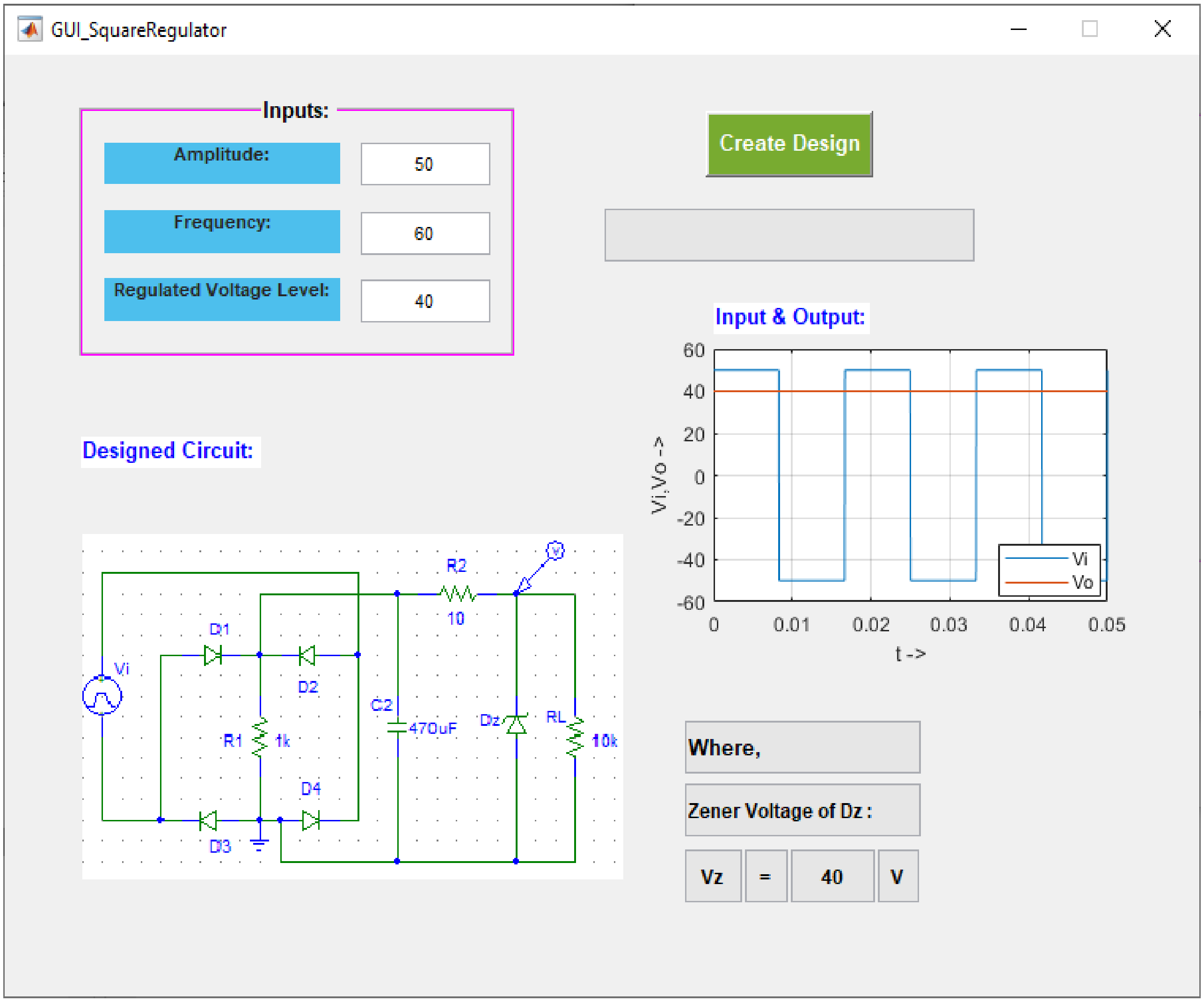
```
function [t,vin,vout,img,Vz]=SquareWave_Regulator_Design(amp,f,dc)
    Vz=dc;
    T=1/f;
    t=linspace(0,3*T,1200);
    vin=amp.*square(2*pi*f*t);
    vout(1:1200)=0;
    for i=1:1200
        vout(i)=Vz;
    end
    img=imread('SquareRegulator.PNG');
end
```

Corresponding Window:



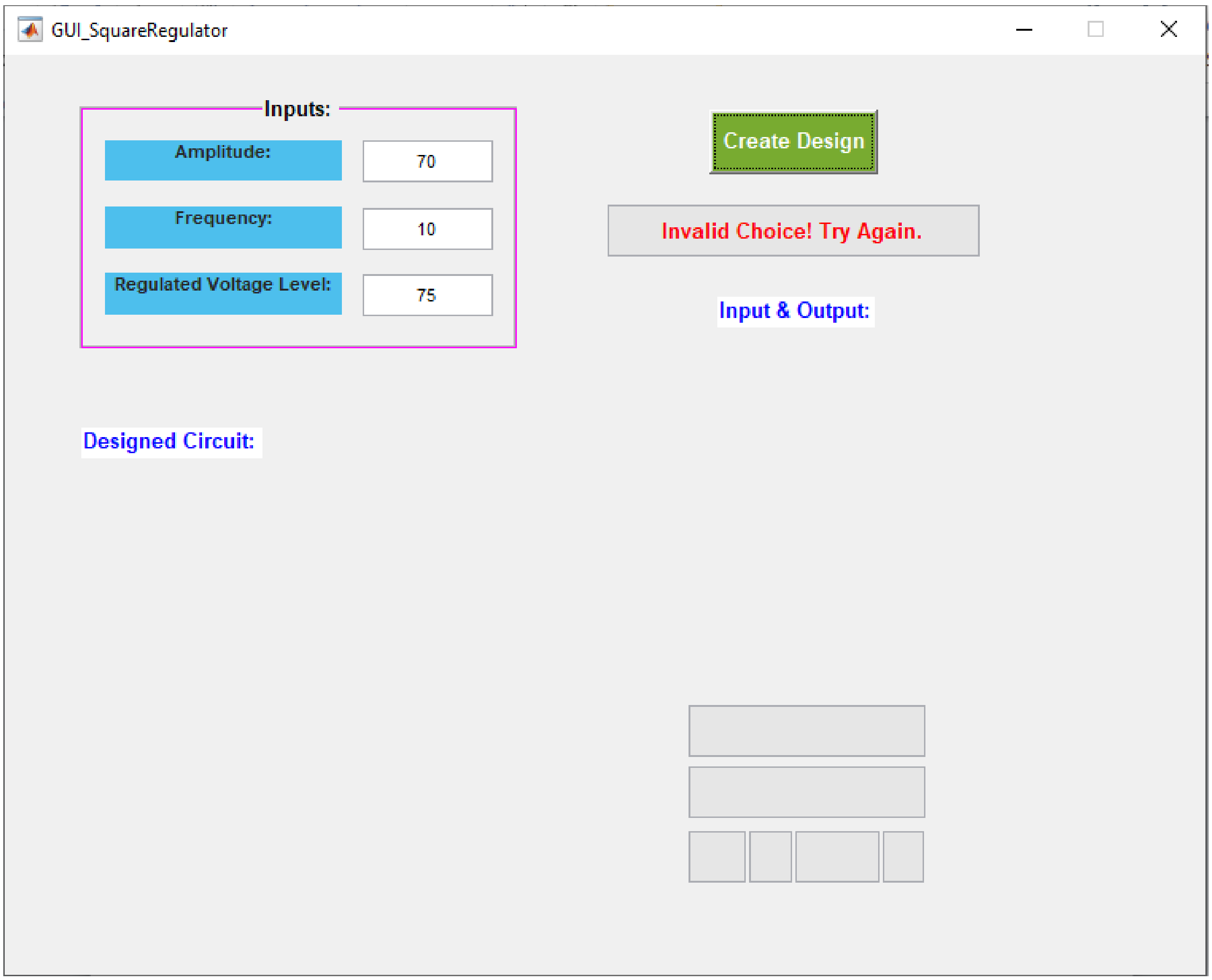
Sample Outputs:

1. Valid Inputs :

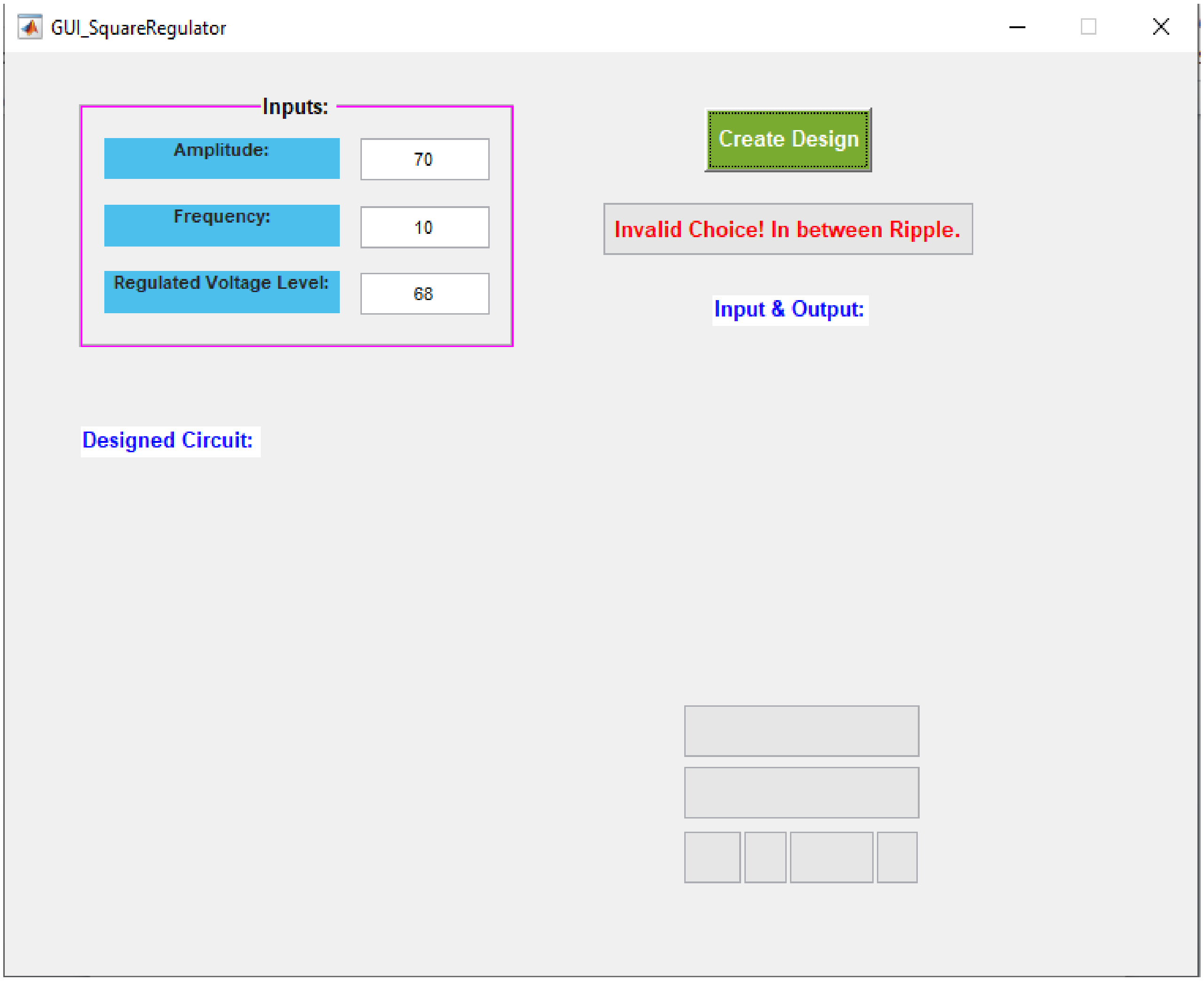


2. Invalid Cases:

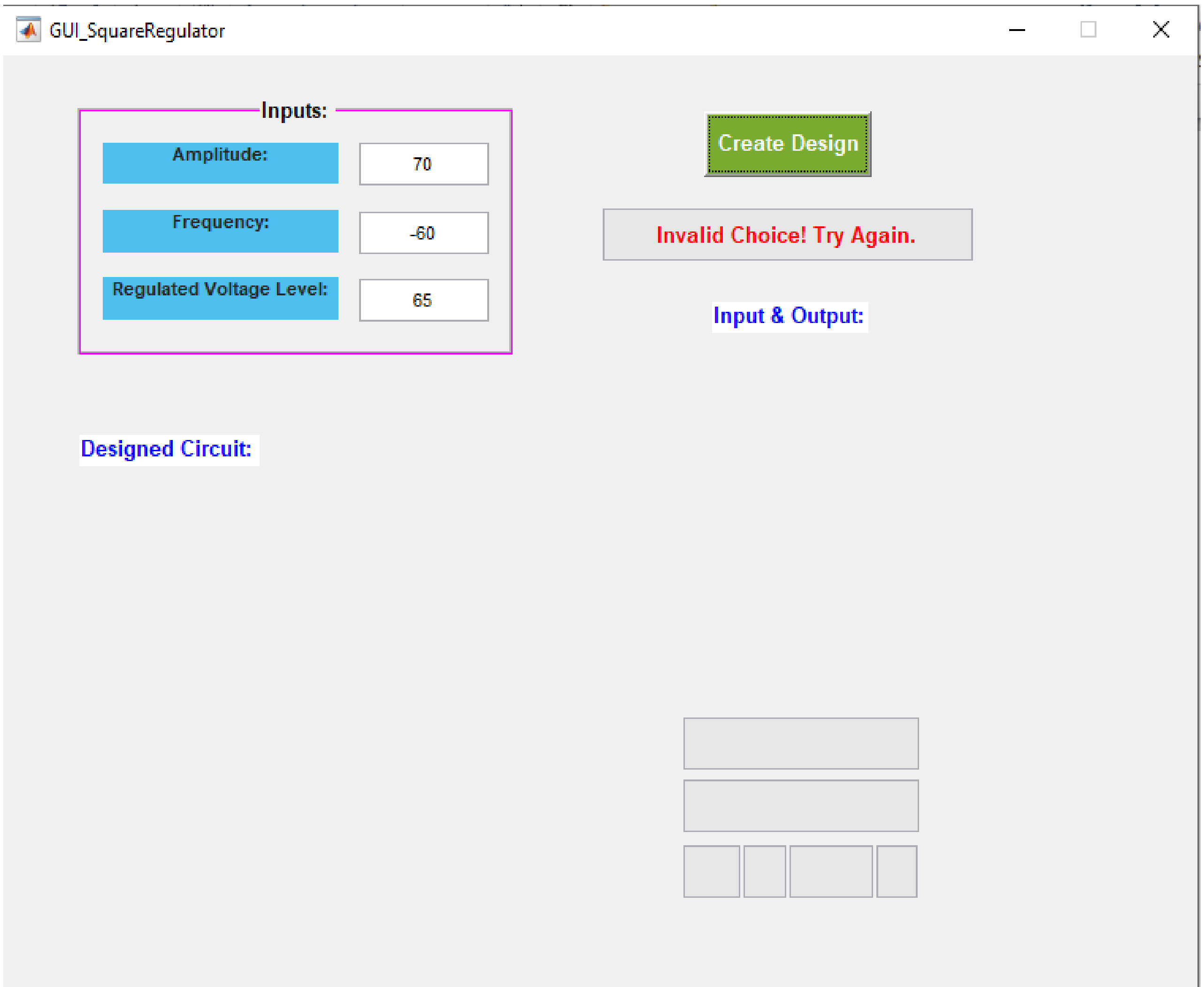
i. Demanded Regulated Voltage Level > Input Voltage Peak :



ii. Demanded Regulated Voltage Level is in between the Ripple :



iii. Zero/ Negative Frequency:



<u>**N. B:**</u> Our program is able to design the circuit for any other kind of inputs as well. User is appreciated to check for more as he desires.

(iii) **<u>Triangular Wave DC Regulator:</u>**

- Now, let us choose "Triangular Wave" from "GUI_DCRegulator" window.
- "GUI_TriangularRegulator.m" interacts with the user and calls "TriangularWave_Regulator_Design.m" that contains all necessary functions and codes to create the design and sends data back.

<u>Codes:</u>

"GUI_TriangularRegulator.m" –

```
function varargout = GUI_TriangularRegulator(varargin)
% GUI_TRIANGULARREGULATOR MATLAB code for GUI_TriangularRegulator.fig
%      GUI_TRIANGULARREGULATOR, by itself, creates a new
GUI_TRIANGULARREGULATOR or raises the existing
%      singleton*.
%
%      H = GUI_TRIANGULARREGULATOR returns the handle to a new
GUI_TRIANGULARREGULATOR or the handle to
%      the existing singleton*.
%
%      GUI_TRIANGULARREGULATOR('CALLBACK',hObject,eventData,handles,...)
calls the local
%      function named CALLBACK in GUI_TRIANGULARREGULATOR.M with the
given input arguments.
%
%      GUI_TRIANGULARREGULATOR('Property','Value',...) creates a new
GUI_TRIANGULARREGULATOR or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before GUI_TriangularRegulator_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to GUI_TriangularRegulator_OpeningFcn via
varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES
```

```
% Edit the above text to modify the response to help GUI_TriangularRegulator

% Last Modified by GUIDE v2.5 26-Jul-2021 21:40:23

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @GUI_TriangularRegulator_OpeningFcn, ...
                   'gui_OutputFcn',  @GUI_TriangularRegulator_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before GUI_TriangularRegulator is made visible.
function GUI_TriangularRegulator_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to GUI_TriangularRegulator (see VARARGIN)

% Choose default command line output for GUI_TriangularRegulator
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes GUI_TriangularRegulator wait for user response (see UIRESUME)
% uiwait(handles.figure1);
```

```
% --- Outputs from this function are returned to the command line.
function varargout = GUI_TriangularRegulator_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;



% --- Executes on button press in design.
function design_Callback(hObject, eventdata, handles)
% hObject    handle to design (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.invalidcheck,'String','');
set(handles.properties,'String','');
set(handles.v11,'String','');
set(handles.v21,'String','');
set(handles.v22,'String','');
set(handles.v23,'String','');
set(handles.v24,'String','');
axes(handles.diagram);
cla;
axes(handles.plots);
cla;

amp=str2double(get(handles.amplitude,'String'));
f=str2double(get(handles.frequency,'String'));
dc=str2double(get(handles.DClevel,'String'));
R=10^3;
C=470*10^-6;
ripple=amp/(2*f*R*C);

if  f<=0 || dc<0 || dc>amp
    set(handles.invalidcheck,'String','Invalid Choice! Try Again.');
elseif dc>amp-ripple && dc<=amp
    set(handles.invalidcheck,'String','Invalid Choice! In between Ripple.');
```

```
else
   [t,vin,vout,img,Vz]=TriangularWave_Regulator_Design(amp,f,dc);
   axes(handles.plots);
   cla;
   plot(t,vin);
   hold on;
   plot(t,vout);
   axis([0,t(end),(-amp-amp/5),(amp+amp/5)]);
   legend('Vi','Vo');
   xlabel('t ->');
   ylabel('Vi,Vo ->');
   grid on;
   axes(handles.diagram);
   cla;
   imshow(img);

   set(handles.properties,'String','Where,');
   set(handles.v11,'String','Zener Voltage of Dz :');
   set(handles.v21,'String','Vz');
   set(handles.v22,'String','=');
   set(handles.v23,'String',num2str(Vz));
   set(handles.v24,'String','V');

end


function properties_Callback(hObject, eventdata, handles)
% hObject    handle to properties (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of properties as text
%        str2double(get(hObject,'String')) returns contents of properties as a double


% --- Executes during object creation, after setting all properties.
function properties_CreateFcn(hObject, eventdata, handles)
% hObject    handle to properties (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function v11_Callback(hObject, eventdata, handles)
% hObject    handle to v11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v11 as text
%        str2double(get(hObject,'String')) returns contents of v11 as a double


% --- Executes during object creation, after setting all properties.
function v11_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function v12_Callback(hObject, eventdata, handles)
% hObject    handle to v12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of v12 as text
%        str2double(get(hObject,'String')) returns contents of v12 as a double


% --- Executes during object creation, after setting all properties.
function v12_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function v13_Callback(hObject, eventdata, handles)
% hObject    handle to v13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v13 as text
%        str2double(get(hObject,'String')) returns contents of v13 as a double


% --- Executes during object creation, after setting all properties.
function v13_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function v21_Callback(hObject, eventdata, handles)
% hObject    handle to v21 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v21 as text
%        str2double(get(hObject,'String')) returns contents of v21 as a double


% --- Executes during object creation, after setting all properties.
function v21_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v21 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function v22_Callback(hObject, eventdata, handles)
% hObject    handle to v22 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v22 as text
%        str2double(get(hObject,'String')) returns contents of v22 as a double


% --- Executes during object creation, after setting all properties.
function v22_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v22 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function v23_Callback(hObject, eventdata, handles)
% hObject    handle to v23 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v23 as text
%        str2double(get(hObject,'String')) returns contents of v23 as a double


% --- Executes during object creation, after setting all properties.
function v23_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v23 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function invalidcheck_Callback(hObject, eventdata, handles)
% hObject    handle to invalidcheck (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of invalidcheck as text
%        str2double(get(hObject,'String')) returns contents of invalidcheck as a double
```

```
% --- Executes during object creation, after setting all properties.
function invalidcheck_CreateFcn(hObject, eventdata, handles)
% hObject    handle to invalidcheck (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function v14_Callback(hObject, eventdata, handles)
% hObject    handle to v14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v14 as text
%        str2double(get(hObject,'String')) returns contents of v14 as a double


% --- Executes during object creation, after setting all properties.
function v14_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function v24_Callback(hObject, eventdata, handles)
% hObject    handle to v24 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v24 as text
%        str2double(get(hObject,'String')) returns contents of v24 as a double
% --- Executes during object creation, after setting all properties.
function v24_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v24 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function amplitude_Callback(hObject, eventdata, handles)
% hObject    handle to amplitude (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of amplitude as text
%        str2double(get(hObject,'String')) returns contents of amplitude as a double


% --- Executes during object creation, after setting all properties.
function amplitude_CreateFcn(hObject, eventdata, handles)
% hObject    handle to amplitude (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

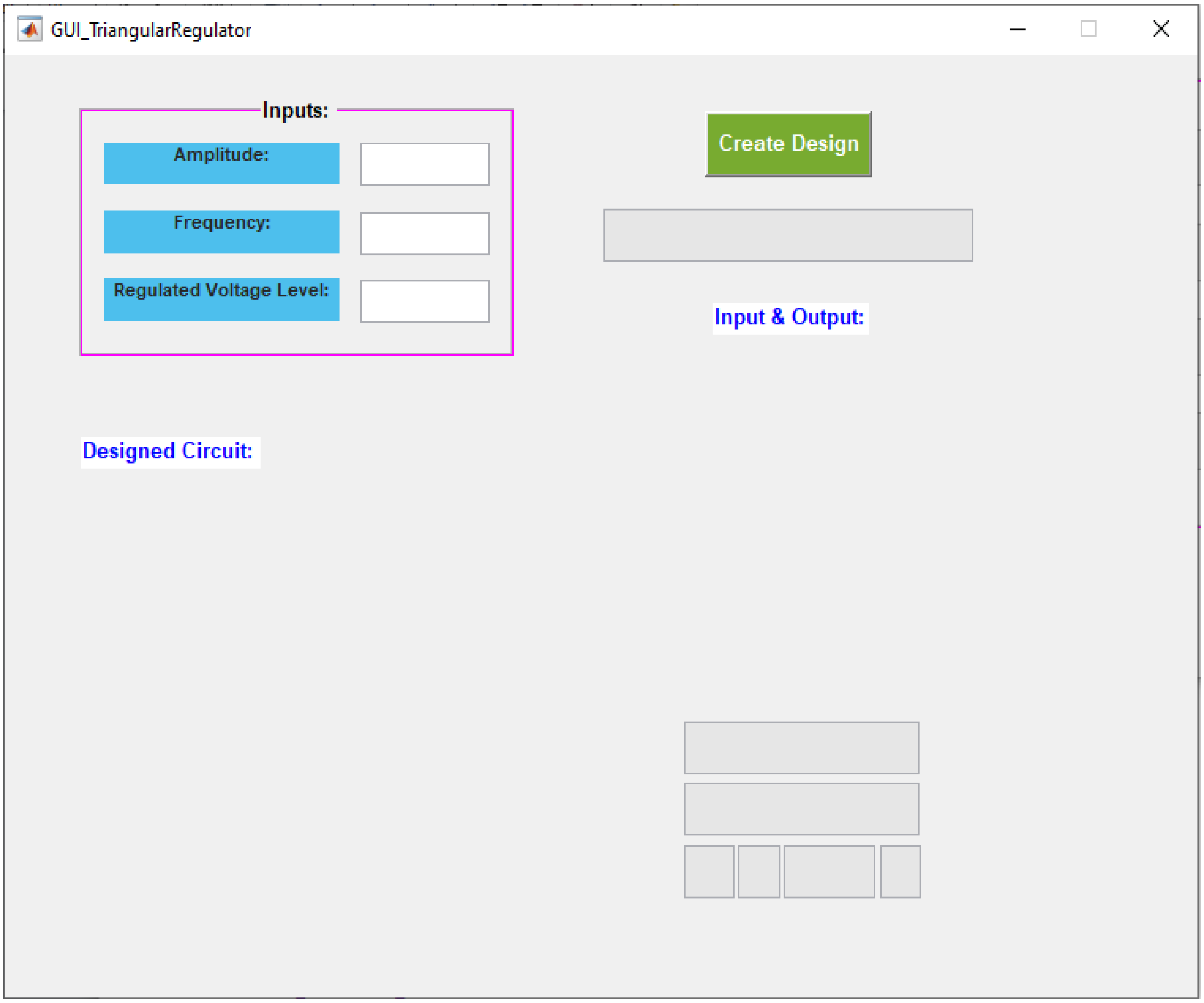
```
function frequency_Callback(hObject, eventdata, handles)
% hObject    handle to frequency (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of frequency as text
%        str2double(get(hObject,'String')) returns contents of frequency as a double


% --- Executes during object creation, after setting all properties.
function frequency_CreateFcn(hObject, eventdata, handles)
% hObject    handle to frequency (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function DClevel_Callback(hObject, eventdata, handles)
% hObject    handle to DClevel (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of DClevel as text
%        str2double(get(hObject,'String')) returns contents of DClevel as a double


% --- Executes during object creation, after setting all properties.
function DClevel_CreateFcn(hObject, eventdata, handles)
% hObject    handle to DClevel (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function edit4_Callback(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit4 as text
%        str2double(get(hObject,'String')) returns contents of edit4 as a double


% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

"TriangularWave_Regulator_Design.m" –

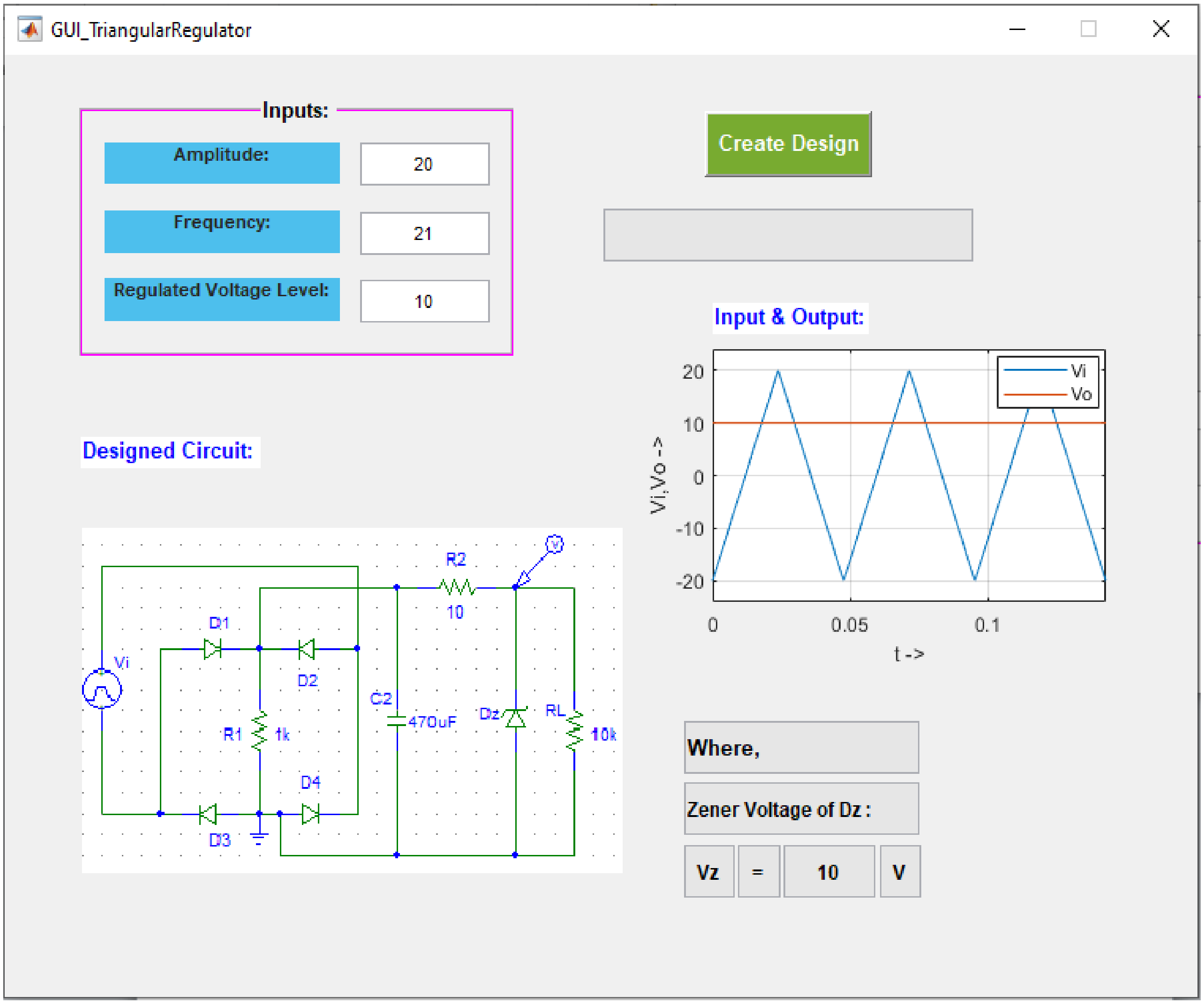
```
function [t,vin,vout,img,Vz]=TriangularWave_Regulator_Design(amp,f,dc)
    Vz=dc;
    T=1/f;
    t=linspace(0,3*T,1200);
    vin=amp.*sawtooth(2*pi*f*t,0.5);
    vout(1:1200)=0;
    for i=1:1200
        vout(i)=Vz;
    end
    img=imread('TriangularRegulator.PNG');
end
```

Corresponding Window:



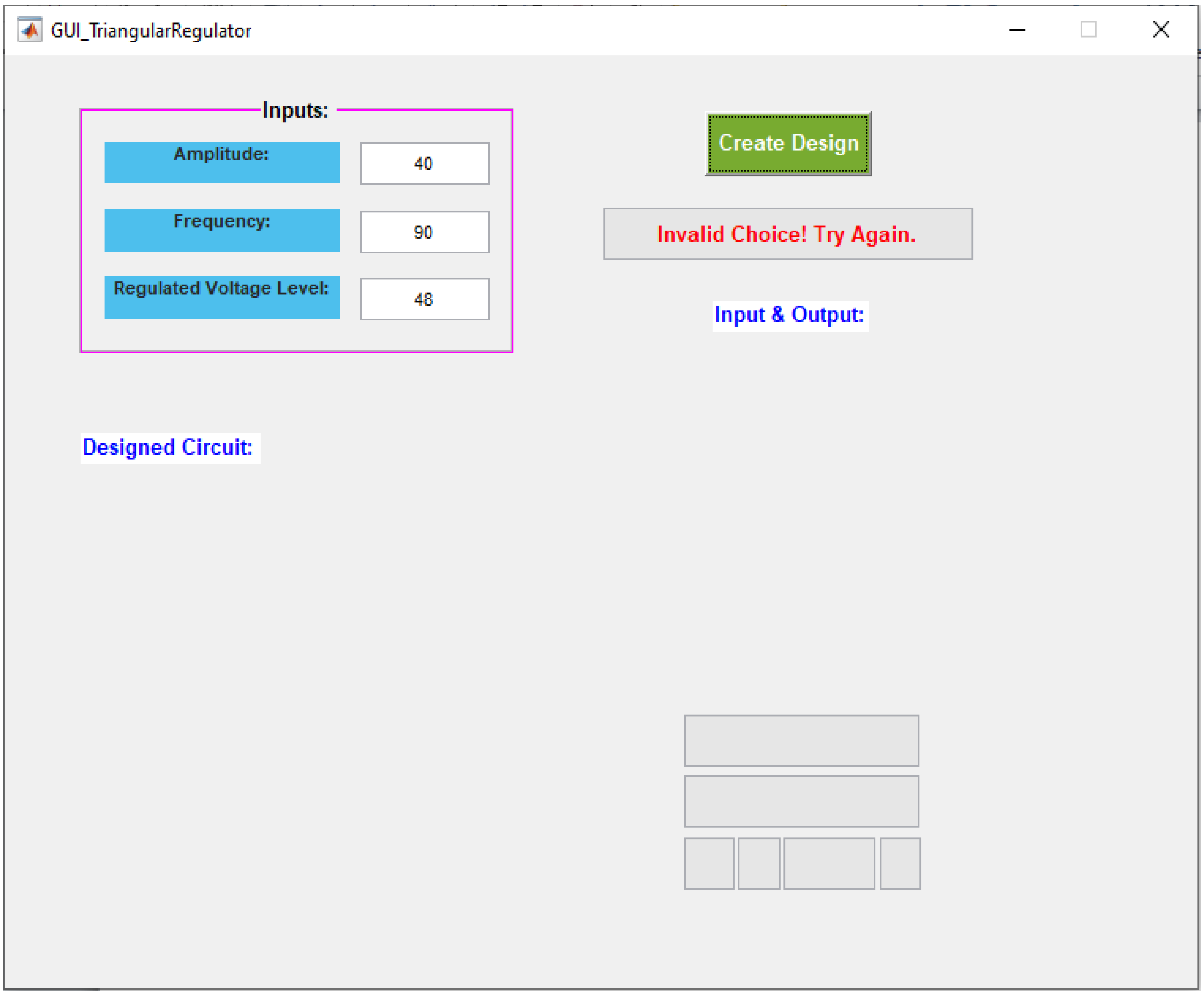
Sample Outputs:

1. Valid Inputs :

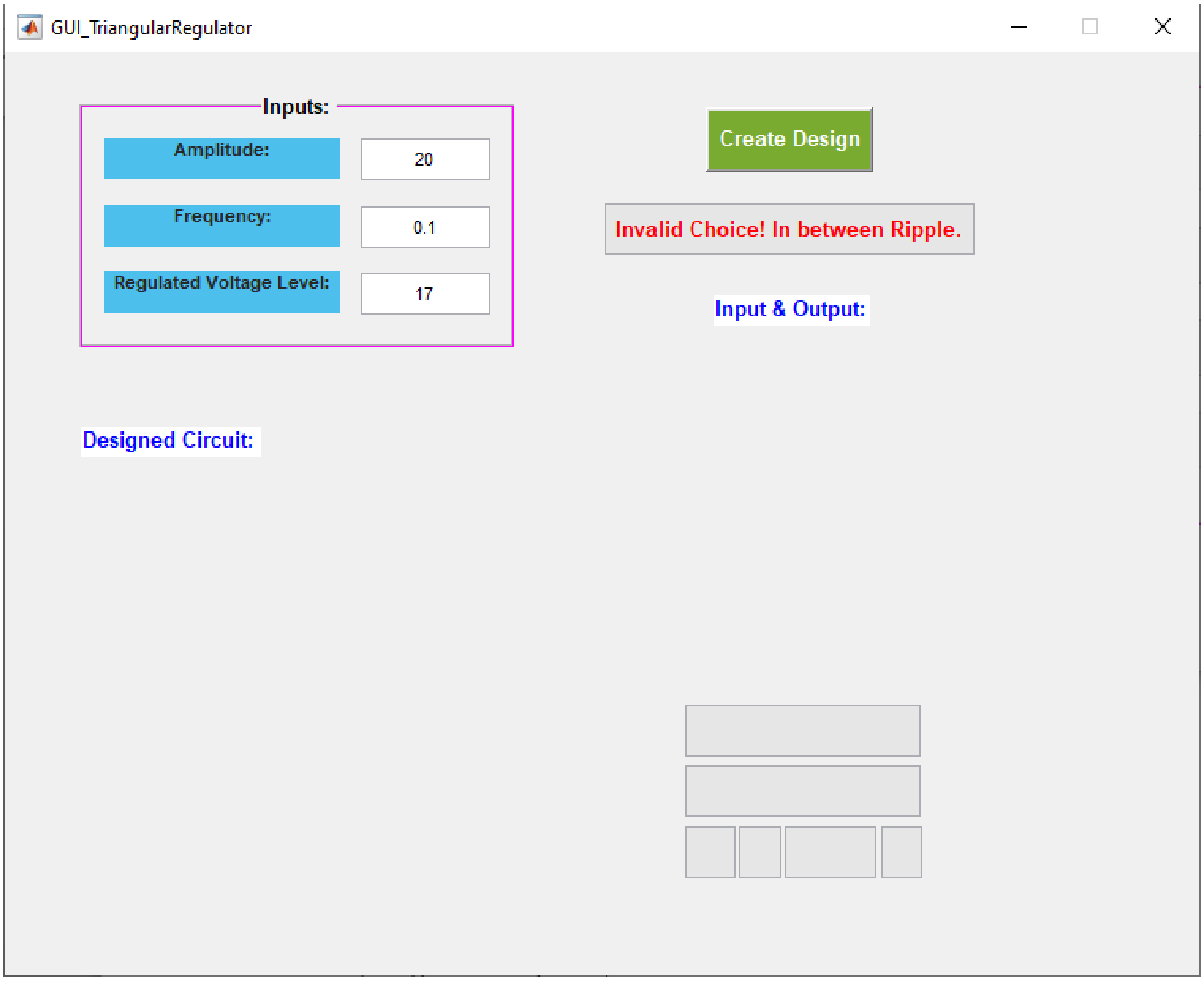


2. Invalid Cases:

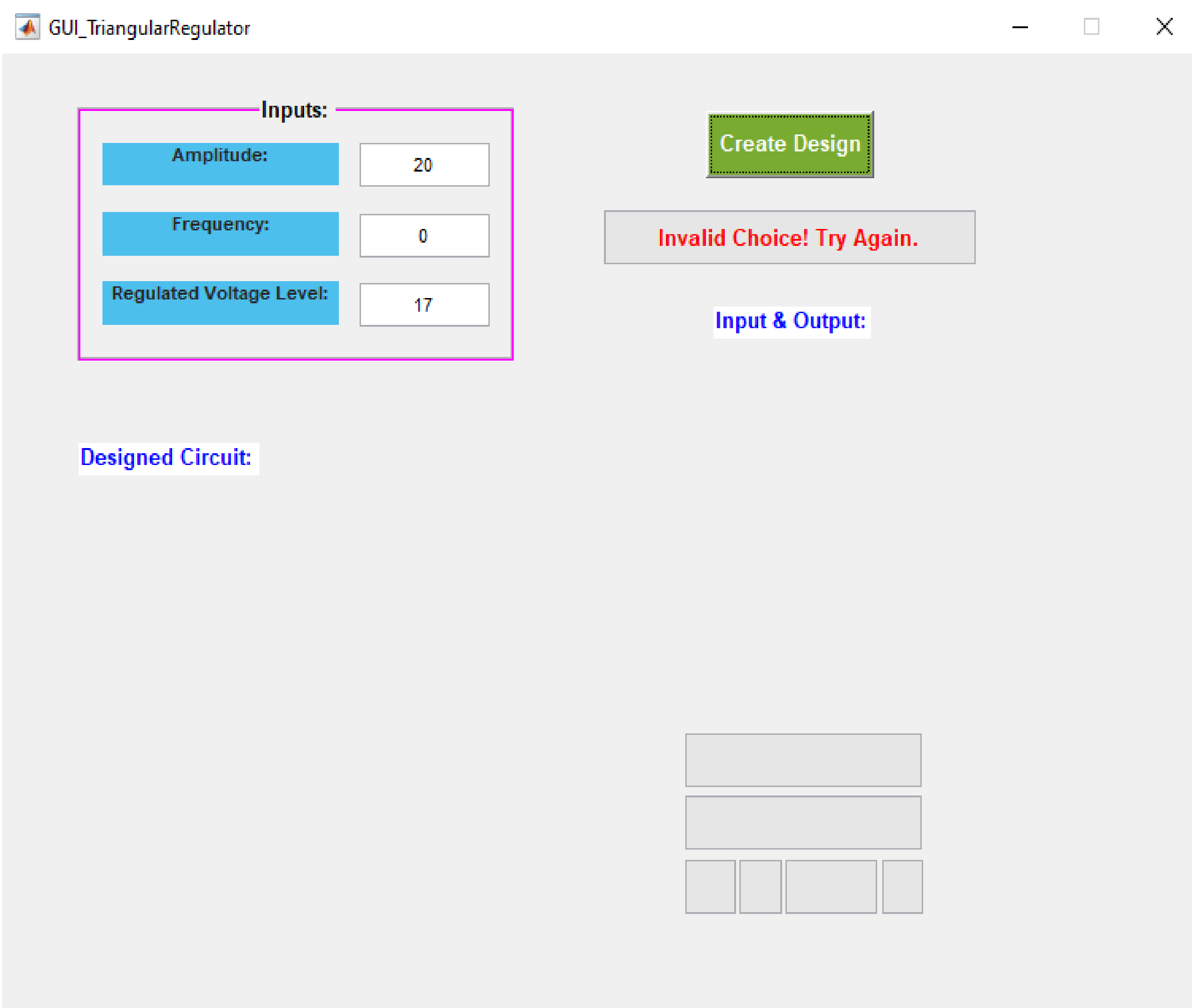
i. Demanded Regulated Voltage Level > Input Voltage Peak :



ii. Demanded Regulated Voltage Level is in between the Ripple :



iii. Zero/ Negative Frequency:



N. B: Our program is able to design the circuit for any other kind of inputs as well. User is appreciated to check for more as he desires.

(iv) **<u>Sawtooth Wave DC Regulator:</u>**

- Now, let us choose "Sawtooth Wave" from "GUI_DCRegulator" window.
- "GUI_SawtoothRegulator.m" interacts with the user and calls "SawtoothWave_Regulator_Design.m" that contains all necessary functions and codes to create the design and sends data back.

<u>Codes:</u>

"GUI_SawtoothRegulator.m" –

```
function varargout = GUI_SawtoothRegulator(varargin)
% GUI_SAWTOOTHREGULATOR MATLAB code for GUI_SawtoothRegulator.fig
%      GUI_SAWTOOTHREGULATOR, by itself, creates a new
GUI_SAWTOOTHREGULATOR or raises the existing
%      singleton*.
%
%      H = GUI_SAWTOOTHREGULATOR returns the handle to a new
GUI_SAWTOOTHREGULATOR or the handle to
%      the existing singleton*.
%
%      GUI_SAWTOOTHREGULATOR('CALLBACK',hObject,eventData,handles,...) calls
the local
%      function named CALLBACK in GUI_SAWTOOTHREGULATOR.M with the given
input arguments.
%
%      GUI_SAWTOOTHREGULATOR('Property','Value',...) creates a new
GUI_SAWTOOTHREGULATOR or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before GUI_SawtoothRegulator_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to GUI_SawtoothRegulator_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help GUI_SawtoothRegulator
```

```matlab
% Last Modified by GUIDE v2.5 26-Jul-2021 21:28:45

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @GUI_SawtoothRegulator_OpeningFcn, ...
                   'gui_OutputFcn',  @GUI_SawtoothRegulator_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before GUI_SawtoothRegulator is made visible.
function GUI_SawtoothRegulator_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to GUI_SawtoothRegulator (see VARARGIN)

% Choose default command line output for GUI_SawtoothRegulator
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes GUI_SawtoothRegulator wait for user response (see UIRESUME)
% uiwait(handles.figure1);
```

```
% --- Outputs from this function are returned to the command line.
function varargout = GUI_SawtoothRegulator_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;



% --- Executes on button press in design.
function design_Callback(hObject, eventdata, handles)
% hObject    handle to design (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.invalidcheck,'String','');
set(handles.properties,'String','');
set(handles.v11,'String','');
set(handles.v21,'String','');
set(handles.v22,'String','');
set(handles.v23,'String','');
set(handles.v24,'String','');
axes(handles.diagram);
cla;
axes(handles.plots);
cla;

amp=str2double(get(handles.amplitude,'String'));
f=str2double(get(handles.frequency,'String'));
dc=str2double(get(handles.DClevel,'String'));
R=10^3;
C=470*10^-6;
ripple=amp/(2*f*R*C);

if  f<=0 || dc<0 || dc>amp
   set(handles.invalidcheck,'String','Invalid Choice! Try Again.');
elseif dc>amp-ripple && dc<=amp
   set(handles.invalidcheck,'String','Invalid Choice! In between Ripple.');
else
```

```
    [t,vin,vout,img,Vz]=SawtoothWave_Regulator_Design(amp,f,dc);
    axes(handles.plots);
    cla;
    plot(t,vin);
    hold on;
    plot(t,vout);
    axis([0,t(end),(-amp-amp/5),(amp+amp/5)]);
    legend('Vi','Vo');
    xlabel('t ->');
    ylabel('Vi,Vo ->');
    grid on;
    axes(handles.diagram);
    cla;
    imshow(img);

    set(handles.properties,'String','Where,');
    set(handles.v11,'String','Zener Voltage of Dz :');
    set(handles.v21,'String','Vz');
    set(handles.v22,'String','=');
    set(handles.v23,'String',num2str(Vz));
    set(handles.v24,'String','V');

end


function properties_Callback(hObject, eventdata, handles)
% hObject    handle to properties (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of properties as text
%        str2double(get(hObject,'String')) returns contents of properties as a double



% --- Executes during object creation, after setting all properties.
function properties_CreateFcn(hObject, eventdata, handles)
% hObject    handle to properties (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
```

```
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function v11_Callback(hObject, eventdata, handles)
% hObject    handle to v11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v11 as text
%        str2double(get(hObject,'String')) returns contents of v11 as a double


% --- Executes during object creation, after setting all properties.
function v11_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function v12_Callback(hObject, eventdata, handles)
% hObject    handle to v12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v12 as text
%        str2double(get(hObject,'String')) returns contents of v12 as a double
% --- Executes during object creation, after setting all properties.
```

```
function v12_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function v13_Callback(hObject, eventdata, handles)
% hObject    handle to v13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v13 as text
%        str2double(get(hObject,'String')) returns contents of v13 as a double


% --- Executes during object creation, after setting all properties.
function v13_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function v21_Callback(hObject, eventdata, handles)
% hObject    handle to v21 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v21 as text
%        str2double(get(hObject,'String')) returns contents of v21 as a double


% --- Executes during object creation, after setting all properties.
function v21_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v21 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function v22_Callback(hObject, eventdata, handles)
% hObject    handle to v22 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v22 as text
%        str2double(get(hObject,'String')) returns contents of v22 as a double


% --- Executes during object creation, after setting all properties.
function v22_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v22 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function v23_Callback(hObject, eventdata, handles)
% hObject    handle to v23 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v23 as text
%        str2double(get(hObject,'String')) returns contents of v23 as a double


% --- Executes during object creation, after setting all properties.
function v23_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v23 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function invalidcheck_Callback(hObject, eventdata, handles)
% hObject    handle to invalidcheck (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of invalidcheck as text
%        str2double(get(hObject,'String')) returns contents of invalidcheck as a double


% --- Executes during object creation, after setting all properties.
function invalidcheck_CreateFcn(hObject, eventdata, handles)
% hObject    handle to invalidcheck (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function v14_Callback(hObject, eventdata, handles)
% hObject    handle to v14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v14 as text
%        str2double(get(hObject,'String')) returns contents of v14 as a double


% --- Executes during object creation, after setting all properties.
function v14_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function v24_Callback(hObject, eventdata, handles)
% hObject    handle to v24 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of v24 as text
%        str2double(get(hObject,'String')) returns contents of v24 as a double

% --- Executes during object creation, after setting all properties.
function v24_CreateFcn(hObject, eventdata, handles)
% hObject    handle to v24 (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function amplitude_Callback(hObject, eventdata, handles)
% hObject    handle to amplitude (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of amplitude as text
%        str2double(get(hObject,'String')) returns contents of amplitude as a double


% --- Executes during object creation, after setting all properties.
function amplitude_CreateFcn(hObject, eventdata, handles)
% hObject    handle to amplitude (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function frequency_Callback(hObject, eventdata, handles)
% hObject    handle to frequency (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of frequency as text
```

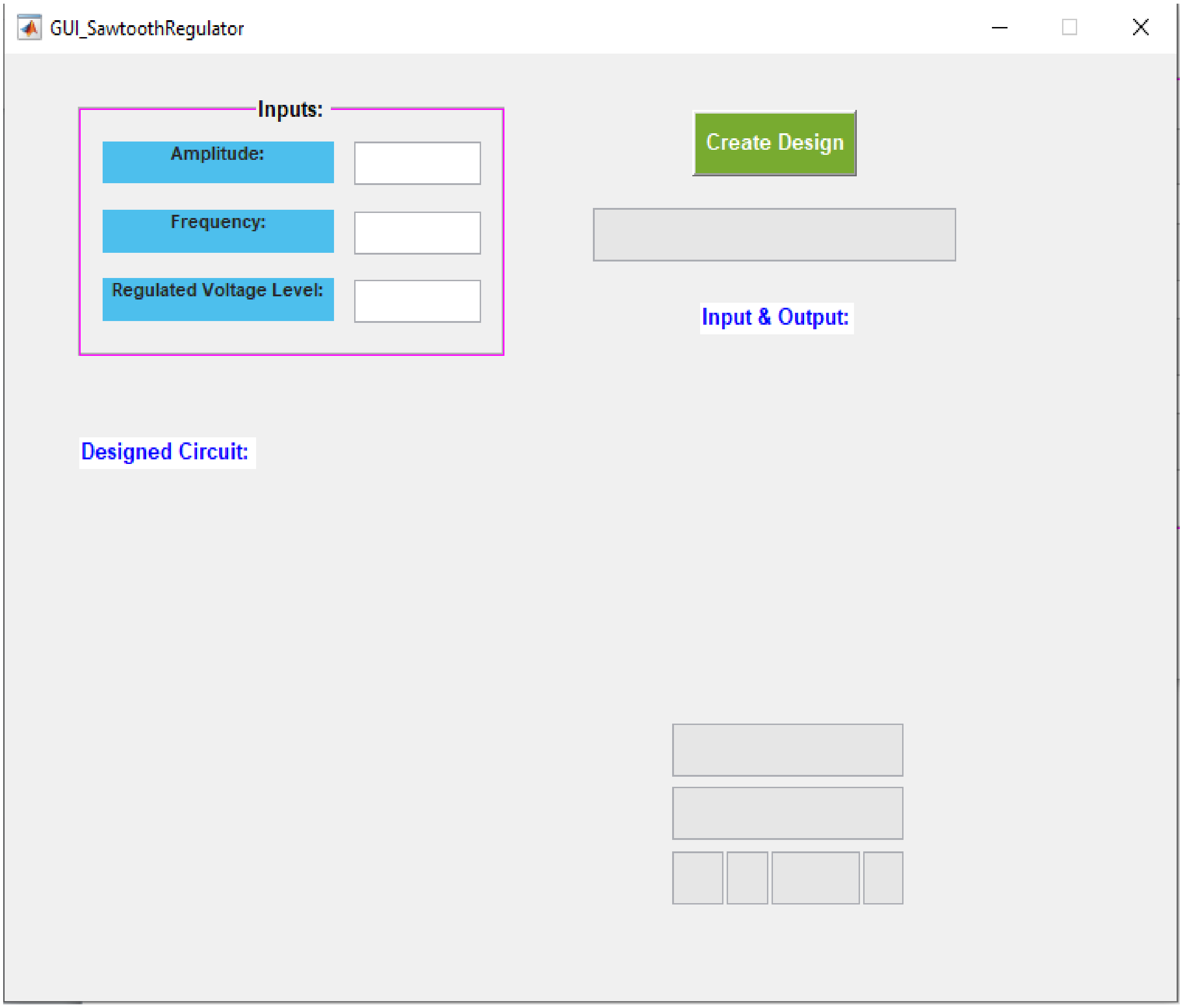
```
%       str2double(get(hObject,'String')) returns contents of frequency as a double


% --- Executes during object creation, after setting all properties.
function frequency_CreateFcn(hObject, eventdata, handles)
% hObject    handle to frequency (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function DClevel_Callback(hObject, eventdata, handles)
% hObject    handle to DClevel (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of DClevel as text
%       str2double(get(hObject,'String')) returns contents of DClevel as a double


% --- Executes during object creation, after setting all properties.
function DClevel_CreateFcn(hObject, eventdata, handles)
% hObject    handle to DClevel (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function edit4_Callback(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit4 as text
%        str2double(get(hObject,'String')) returns contents of edit4 as a double


% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

"SawtoothWave_Regulator_Design.m" –

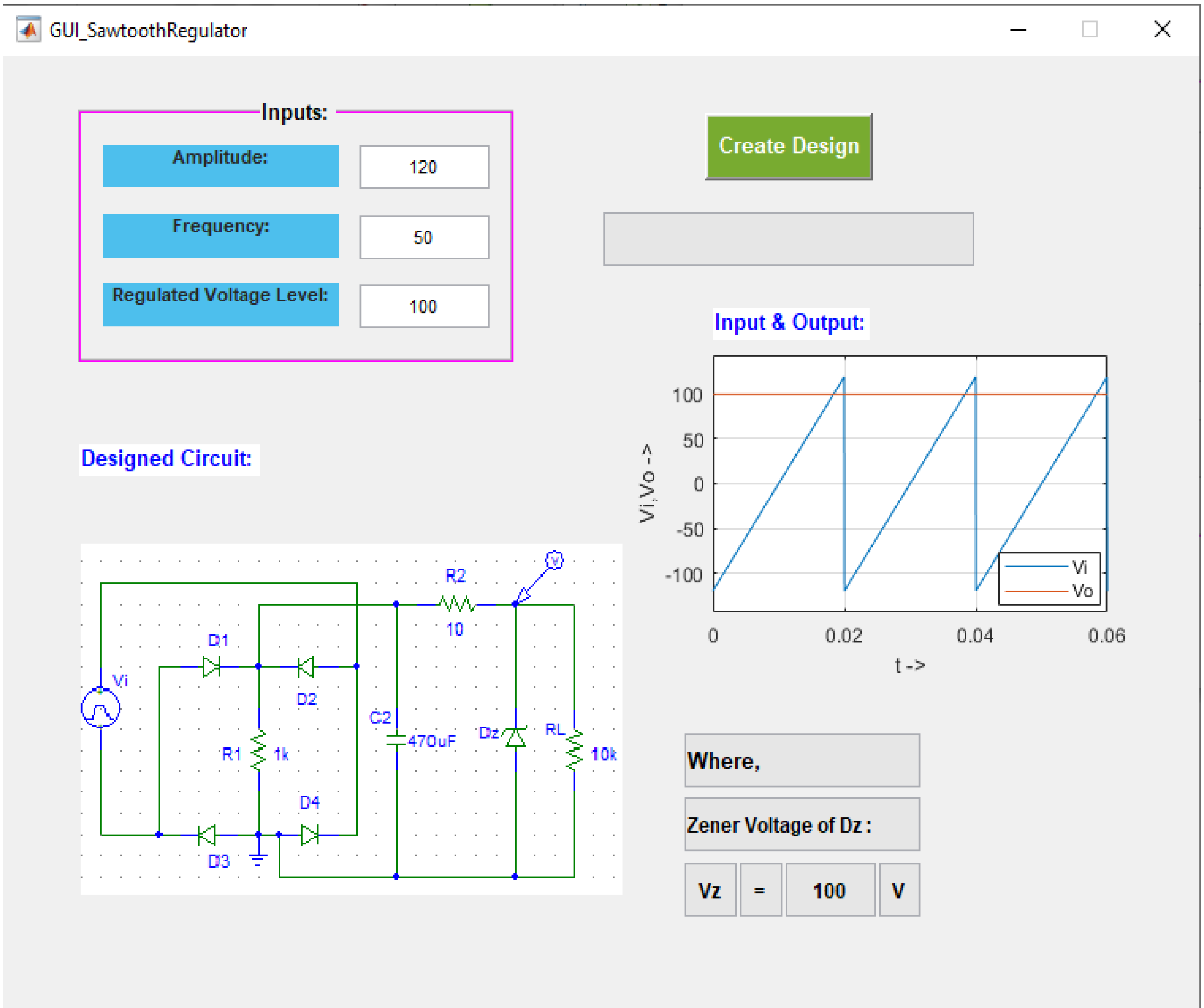
```
function [t,vin,vout,img,Vz]=SawtoothWave_Regulator_Design(amp,f,dc)
    Vz=dc;
    T=1/f;
    t=linspace(0,3*T,1200);
    vin=amp.*sawtooth(2*pi*f*t);
    vout(1:1200)=0;
    for i=1:1200
        vout(i)=Vz;
    end
    img=imread('SawtoothRegulator.PNG');
end
```

Corresponding Window:



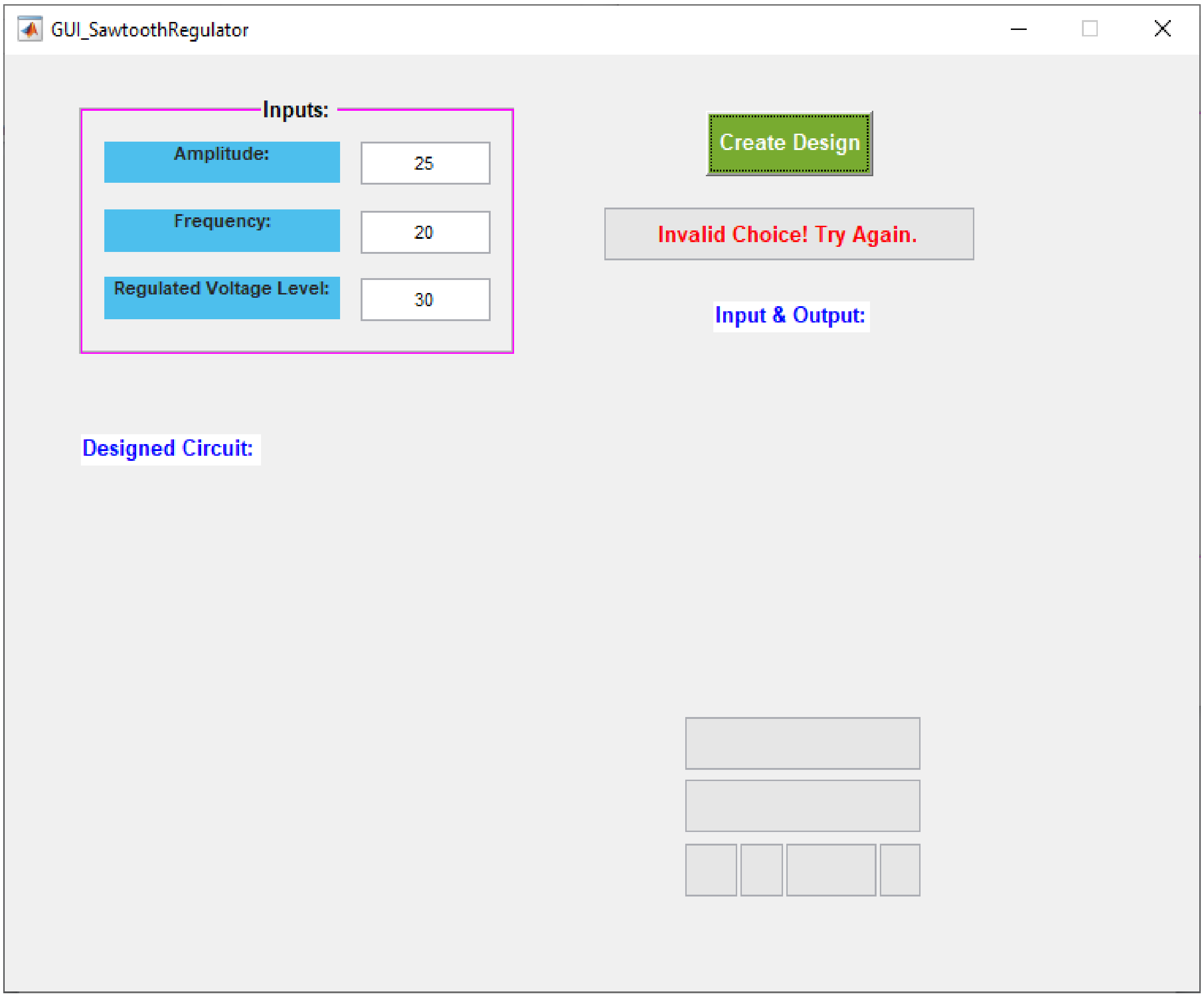
Sample Outputs:

1. Valid Inputs :

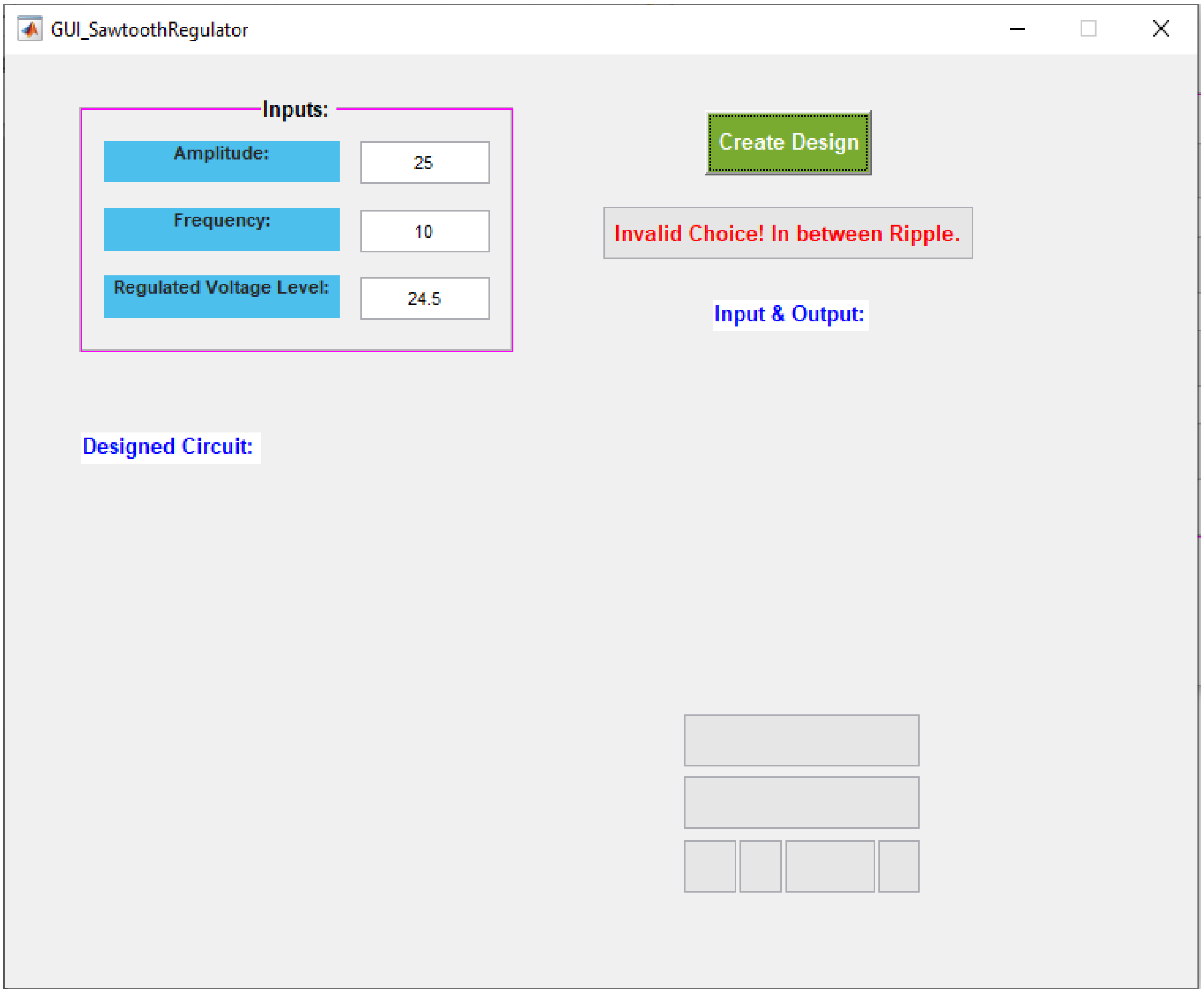


2. Invalid Cases:

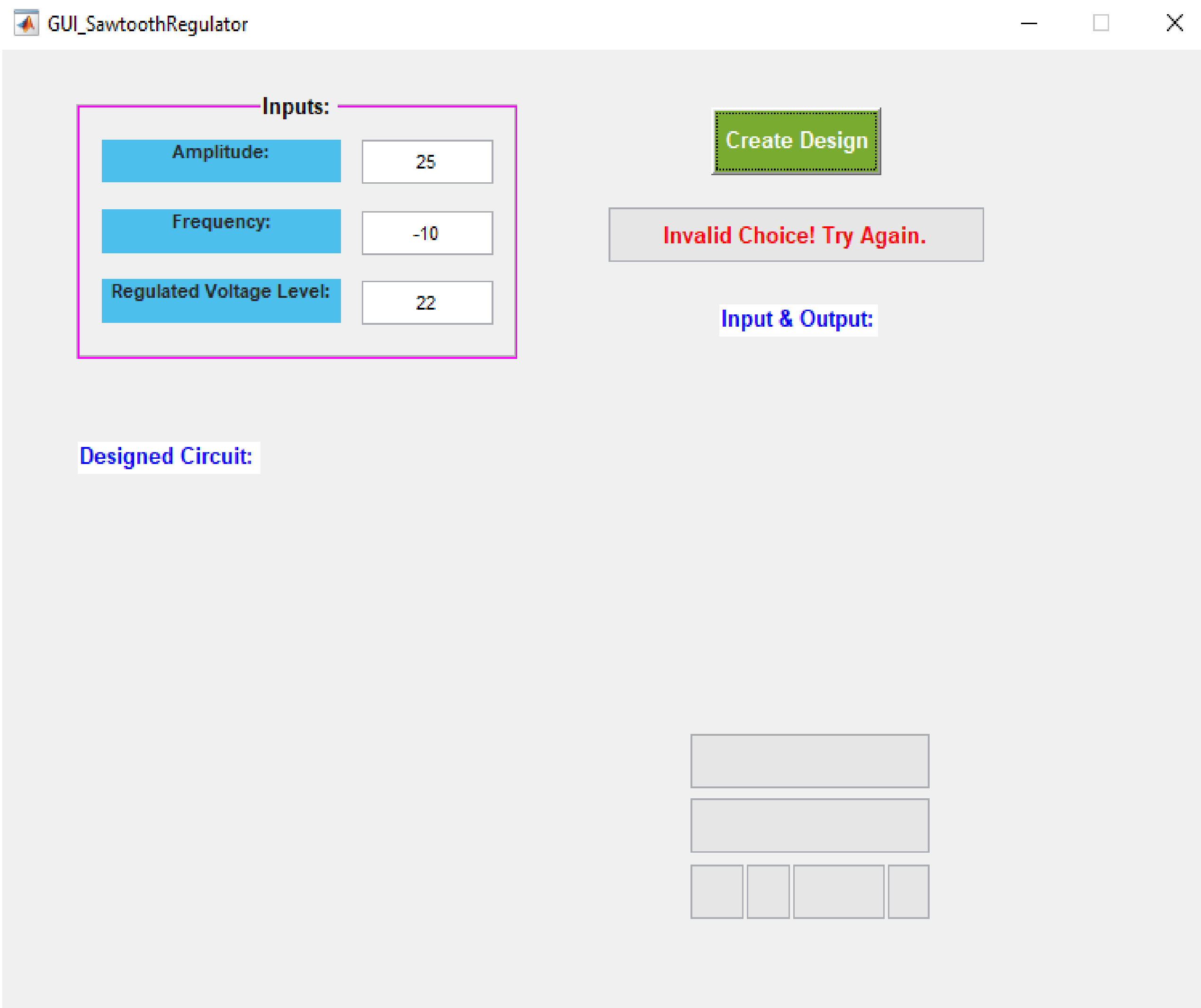
i. Demanded Regulated Voltage Level > Input Voltage Peak :



ii. Demanded Regulated Voltage Level is in between the Ripple :



iii. Zero/ Negative Frequency:



<u>N. B:</u> Our program is able to design the circuit for any other kind of inputs as well. User is appreciated to check for more as he desires.

DISCUSSION

Throughout the project, we have seen how diode along with resistor, capacitor and DC voltage source performs clipping, clamping and DC regulation. We have seen these operations in case of sources with sinusoidal, square, triangular and sawtooth wave shape. From one type to another, diodes get biased in little bit different fashion. Therefore, we have seen some early change in output shape from type to type. In case of clamper, capacitors are used. There, one should ensure that in the design $5\tau >> \frac{T}{2}$. This condition has also been implemented in our design so that the design becomes more general to the inputs. Next, in the regulator circuit the filter capacitor results in a smoothed wave with ripple. This ripple part can't be accessed as demanded DC voltage. In our program we have set this ripple part as invalid choice and the program shows a warning massage when a DC in this range is demanded. We have tested our program with various sets of input and this is worth to say that the program works for each and every combination. Hence, our program is well generalized. Next, our program is smart enough to handle any mistake from the user. For example, if the user accidentally gives a negative or zero frequency to operate, the program shows that the user's choice is invalid and he should try again with proper inputs. Similarly our program says, clipping level can't be greater than the amplitude and lower voltage level can't be greater than upper voltage level. Moreover, the user interface has been kept simple so that the user can easily understand the operation. The only thing he should ensure is choosing proper options step by step, giving valid inputs and hitting the "Create Design" button. The user never is to think what is going behind the window and he simply gets his desired output. Last but not the least, our program ensures the indefectible output within minimum amount of time.

FUTURE PROSPECT OF IMPROVEMENT :

Throughout the project we have considered ideal diodes. Under this assumption, we have assumed that the voltage drop across them is zero. That is, in forward bias they act as a short circuit. But, real life diodes permit a finite voltage drop across them. In case of silicon diode this voltage is 0.7 V and in case of germanium diode this voltage is 0.3 V. Thus, counting the voltage drop across the diode in our program would lead to more accurate result. Furthermore, the resistance of connecting wires has been neglected. One can bring this resistance in their model for more precise result. Finally, various complex circuits can produce complicated waveforms rather than the basic four we have covered. In that case, who will work next can attempt a common program for them. That is, if generalization of sources can made possible that will be, no doubt, a huge achievement.