1 test

2 test

results:

Table 1: Theoretical vs. Measured Results for Full-Wave Rectifier

	Measured Voltage [V]	Theoretical Voltage [V]	Percentage Error
Peak	8.906	8.6	3.6%
Trough	8.593	8.6	0.1%

3 Discussion

3.1 Transient Response of Diodes

Both the pn-junction diode and Schottky diode yielded waveforms that agree well with theory. In analyzing the two types of diodes, the advantages and disadvantages of each are made clear. The Schottky diode is observed to have a nonexistent storage time whereas the pn-junction diode is observed to have a storage time of approximately $4\mu s$. The very apparent delay in rectification from the storage time of the pn-junction diode makes that type of diode not suited for high frequency applications. The Schottky diode, however, is very commonly used in high frequency applications like RF. What is not shown in this experiment is that Schottky diodes have significantly lower reverse breakdown voltages than pn-junction diodes (2), therefore Schottky diodes may not be suitable for high voltage environments. It should also be noted that the diode's 10% error or so in Table (1) is likely a result of nonidealities in the manufacturing process of both the diode and the resistor. The resistor used in the measurement likely deviates from the 300Ω value, and the diode has some internal resistance as well. Moreover, the diode's properties, such as V_{on} in equations (5) and (6) are also subject to variations in the manufacturing process, such as in doping concentrations. The important part of the experiment is the shape of the waveform in figure (4). Better characterization of the properties of the resistor and the diode are required for more accurate results.

3.2 Full-Wave Bridge Rectifier

The full-wave bridge rectifier results align quite well with theory. However, the peak measurement does have slightly more error than the trough measurement. These errors are a result of either an asymmetry in the source voltage or an asymmetry in the threshold voltages of the diodes. The former is not possible to prove due to the fact that the source voltage is not probed in the experiment. However, if the sinusoidal source voltage's peak is higher than the reported value on the power supply, this would explain why the measured voltage for the output voltage's peak is higher than expected. Furthermore, the diodes may differ in threshold voltage. Differences in threshold voltage are a result of either temperature, material, or doping variations:

$$V_{bi} = \frac{k_B T}{q} ln(\frac{N_A N_D}{(n_i(T))^2})$$

Temperature is likely not the cause since the temperature in the room in which the experiment is conducted is essentially uniform, which would not explain the asymmetry in the measurements. A better explanation is that the doping concentrations vary slightly from diode to diode due to slight nonidealities in the manufacturing process. So, the diodes along the enabled path when the source voltage is positive could have slightly lower threshold voltages than the diodes along the other path.

4 Appendix

The following script is used to generate the theoretical pn-junction diode current plot in figure (2):

```
#!/usr/bin/python

import numpy as np
import matplotlib.pyplot as plt

If = 3
Ir = 2
t_s = 4

region1 = np.linspace( -10 , 0 , 100 )
region2 = np.linspace( 0.01 , t_s , 100 )
region3 = np.linspace( 4.01 , 10 , 100 )

region1_i = np.array( [ If ] * len( region1 ) )
region2_i = np.array( [ -Ir ] * len( region2 ) )
region3_i = ( -Ir ) * np.exp( -( region3 - t_s ) )

t = np.concatenate( [ region1 , region2 , region3 ] )
i = np.concatenate( [ region1_i , region2_i , region3_i ]
    )

plt.plot( t , i )
plt.xlabel( "t" )
plt.ylabel( "i(t)" )
plt.savefig( "diode_current.PNG" )
```

The following script is used to generate the theoretical pn-junction diode voltage plot in figure (3):

```
#!/usr/bin/python

import numpy as np
import matplotlib.pyplot as plt

Vf = 2
ts = 4

region1 = np.linspace( -10 , 0 , 100 )
region2 = np.linspace( 0.01 , ts , 100 )

region1_v = np.array( [ Vf ] * len( region1 ) )
region2_v = Vf * np.sqrt( 1 - ( ( region2 / ts ) ** 2 ) )

t = np.concatenate( [ region1 , region2 ] )
v = np.concatenate( [ region1_v , region2_v ] )

plt.plot( t , v )
plt.ylim( 0 , 4 )
plt.xlabel( "t" )
plt.ylabel( "$V_{diode}$" )
plt.savefig( "diode_voltage.PNG" )
```

The following script is used to generate the theoretical full-wave recitifer plot in figure (15):

```
#!/usr/bin/python

import numpy as np
import matplotlib.pyplot as plt

V0 = 10
Vth = 0.7

t = np.linspace( 0 , 100 , 10000 )
Vin = V0 * np.sin( 0.2 * t )
Vout = np.array([])
for point in Vin:
        out_val = 0
        if abs( point ) > 2 * Vth:
                out_val = abs( point ) - ( 2 * Vth )
        Vout = np.append( Vout , [ [ out_val ] ] )
vin_plt , = plt.plot( t , Vin , label = "$V_{in}$" )
vout_plt , = plt.plot( t , Vout , label = "$V_{out}$" )
plt.tick_params( labelbottom = 'off' )
plt.xlabel( "t" )
plt.ylabel( "$V_{A}$" )
```

```
plt.legend( handles = [ vin_plt , vout_plt ] )
plt.savefig( "../images/full_wave_rect.PNG" )
```

The following script is used to generate tables (3) and (4):

```
#!/usr/bin/python

# Roman Parise
# Full-Wave Rectifier Data

# Input src: 20 Vpp

import common

TABLES_DIR = "../tables/"
FWR_DATA_FNAME = "fwr_data.csv"
FWR_ERR_FNAME = "fwr_err.csv"

# Data for peaks and troughs table

rectified_amplitude_peak = 8.906 #V
rectified_amplitude_trough = 8.593 #V

source_peak = 10 #V
source_trough = 10 #V

diff_peak = abs( source_peak - rectified_amplitude_peak )
diff_trough = abs( source_trough -
    rectified_amplitude_trough )

data_headings = [ "" , "Measured Voltage [ V ]" , "Source
     Voltage [ V ]" , "Difference [ V ]" ]
peak_data = [ rectified_amplitude_peak , source_peak ,
    diff_peak ]
trough_data = [ rectified_amplitude_trough ,
    source_trough , diff_trough ]
data_matrix = [ data_headings , [ "Peak" ] + peak_data ,
    [ "Trough" ] + trough_data ]

common.write_csv_from_matrix( TABLES_DIR + FWR_DATA_FNAME
     , data_matrix )

# Errors from theory table

threshold_voltage = 0.7 #V

th_rect_amp = lambda V : abs( V ) - ( 2 *
```

```
    threshold_voltage )
th_rect_amp_peak = th_rect_amp( source_peak )
th_rect_amp_trough = th_rect_amp( source_trough )

data_headings_err = [ "" , "Measured Voltage [ V ]" , "
    Theoretical Voltage [ V ]" , "Percentage Error" ]
peak_data_err = [ rectified_amplitude_peak ,
    th_rect_amp_peak , common.fmt_perc_err(
    rectified_amplitude_peak , th_rect_amp_peak ) ]
trough_data_err = [ rectified_amplitude_trough ,
    th_rect_amp_trough , common.fmt_perc_err(
    rectified_amplitude_trough , th_rect_amp_trough ) ]
data_matrix_err = [ data_headings_err , [ "Peak" ] +
    peak_data_err , [ "Trough" ] + trough_data_err ]

common.write_csv_from_matrix( TABLES_DIR + FWR_ERR_FNAME
    , data_matrix_err )
```

Lastly, this script is used to generate figure (10):

```
#!/usr/bin/python

# Roman Parise

import matplotlib.pyplot as plt
import numpy as np

I0 = 1e-10
Vthermal = 26e-3
infty = 1e25

Va_shock = np.linspace( 0 , 1.0 , 1000 )
shockley_diode = I0 * ( np.exp( Va_shock / Vthermal ) - 1
     )

Va_ideal = np.linspace( 0 , 0.7 , 1000 )
ideal_diode = [ 0 ] * ( len( Va_ideal ) - 1 ) + [ infty ]

plt.xlabel( "$V_{applied}$ [ V ]" )
plt.ylabel( "$I_{diode}$ [ A ]" )

plt.ylim( -1 , 100 )

shock_plt , = plt.plot( Va_shock , shockley_diode , label
    = "Shockley's Diode Equation" )
ideal_plt , = plt.plot( Va_ideal , ideal_diode , label = "
    Perfectly Ideal Diode" )
```

```
plt.legend( handles = [ shock_plt , ideal_plt ] )
plt.savefig( "../images/ideal_diode.PNG" )
```

5 References

1. https://tex.stackexchange.com/questions/236359/making-a-list-of-websites
2. http://www.radio-electronics.com/info/data/semicond/schottky_diode/schottky_barrier_diode.php