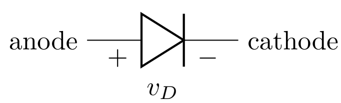
SPICE 3: Diodes

ECE 3410, Utah State University

Diodes in SPICE

A diode in SPICE is instantiated by starting a line with the letter `D`:




```
D<name> <anode> <cathode> <modelname>
```

Diode Models

Since diodes have complicated mathematical models, the model parameters must be provided by using the `.model` keyword. As an example, we can model a diode with these parameters:

Parameter	Symbol	Value
Scale current	IS	5nA
Slope factor	N	2
Reverse breakdown voltage	BV	100V

Using the `.model` keyword, we can group these parameters into a model named `mydiode` like this:

```
.model mydiode D
+ Is=5e-9
+ N=2.0
+ BV=100
```

After the model is defined, a diode with those parameters can be instantiated like so:

```
D1 n1 n2 mydiode
```

Model of the 1N914 Diode

A widely used general-purpose diode is the 1N914. The model parameters for this diode, obtained from ON Semiconductor, are provided in the file `models/D1N914.md` which has these contents:

```
.MODEL D1N914 D
+ IS=5.0e-09
+ RS=0.8622
+ N=2.0
+ ISR=9.808e-14
+ NR=2.0
+ CJO=8.68e-13
+ M=0.02504
+ VJ=0.90906
+ FC=0.5
+ TT=6.012e-9
+ BV=100
+ IBV=1e-07
+ EG=0.92
```

Exercise 1: Simulate the I/V Characteristic

To better understand the diode's function, run a simulation of the circuit shown below, using the D1N914 diode model. The netlist and simulation commands are explained in the following slides.

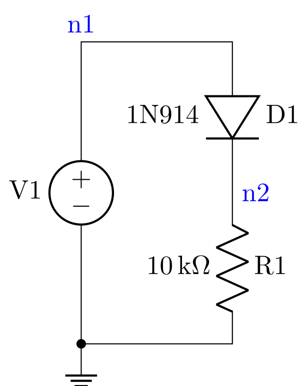


Figure 1: Schematic for I/V test.

`netlists/exercise1.sp`

```
* Diode I/V Test
.include models/D1N914.md
```

```
V1 n1 0   DC 1V

D1 n1 n2  D1N914
R1 n2 gnd 10k

.control
  dc V1 5 20 0.25

  let vd=v(n1)-v(n2)            $ Voltage across diode
  let id=-i(v1)                 $ Current through diode
  let ip='5e-9*exp(vd/.052)'  $ Predicted current

  plot id ip VS vd

  hardcopy plots/iv_test.svg id ip VS vd

  meas dc vf FIND vd WHEN id=1e-3
.endc
.end
```

Include the Diode Model

Whenever a netlist uses semiconductor devices like diodes, the model must be defined prior to any component instances. Therefore in the netlist header we have a `.include` line to load the model:

```
.include models/D1N914.md
```

Define the Circuit Netlist

Next we place the components.

Notice that this circuit uses a resistor in series with the diode. If there were no series resistor, the diode could draw a very large current when forward biased, possibly resulting in damage to the diode. In the worst case, a short-circuited diode or battery could overheat leading to explosive or incindiary hazards.

```
V1 n1 0   DC 1V

D1 n1 n2  D1N914
R1 n2 gnd 10k
```

DC Sweep Simulation

Next we perform a DC sweep simulation of the diode's forward bias characteristic. Since the diode is in series with a 10kΩ resistor, we will sweep over a wide voltage range from 5V to 20V. The diode itself will only see 0V up to 0.7V, with the rest of the voltage appearing across the resistor.

In this range, the minimum current will be less than 0.5mA and the maximum will be approximately $(20\text{V} - 0.7\text{V})/10\text{k}\Omega = 1.93\text{mA}$.

```
.control
  dc V1 5 20 0.25
```

Measure the I/V Values

Using the `let` command we compute expressions for the voltage drop across D1, the current observed through the voltage source V1, and the predicted current using the exponential model. Recall the exponential forward bias model equation:

$$i_P = I_S e^{v_D/nU_T}$$

where $U_T \approx 26\text{mV}$.

This equation is implemented using an expression in the final `let` command, to compute the predicted current within SPICE, so that we can easily compare it to the simulated current.

```
  let vd=v(n1)-v(n2)           $ Voltage across diode
  let id=-i(v1)                $ Current through diode
  let ip='5e-9*exp(vd/.052)'   $ Predicted current
```

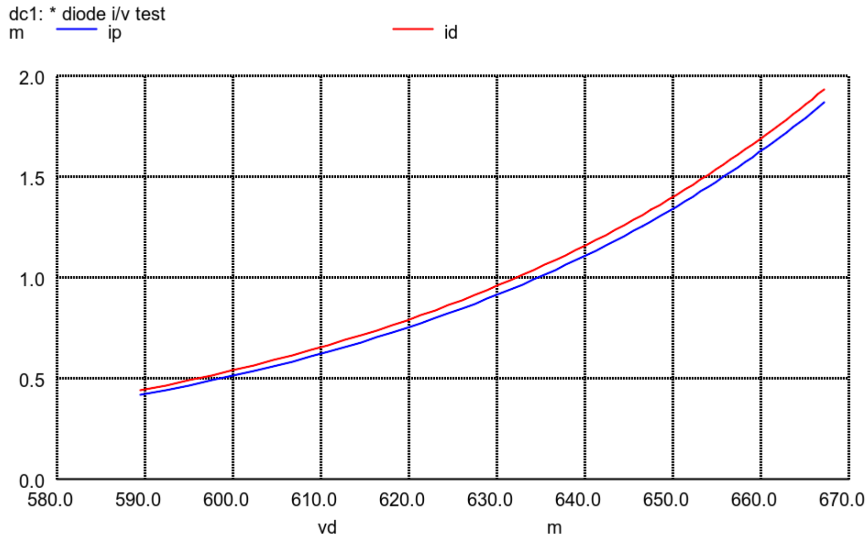
Plot Results

Next we plot together the measured current and the predicted current. In the plot command, we use the `VS` keyword to use the diode's voltage `vd` as the X axis. Here I've used capital letters for the `VS` keyword so it is easily recognized:

```
  plot id ip VS vd

  hardcopy plots/iv_test.svg id ip VS vd
```

The simulation should show that the simulation and prediction are close but not an exact match, as seen in the figure below. Our exponential equation does not account for all the physical details that affect a real diode. SPICE handles a more complicated set of parameters and equations.



Measure the Forward Voltage

The “typical” textbook diode has a forward voltage drop of 0.7V when its current is 1mA. The specific 1N914 diode is a little different. We use the `meas` command to obtain the precise measurement for this particular diode:

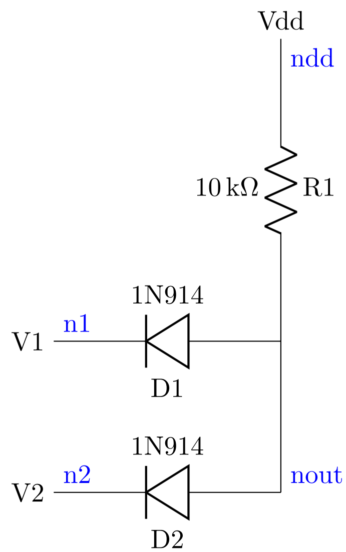
```
  meas DC vf FIND vd WHEN id=1e-3
.endc
.end
```

This command will measure the value of `vd` at the point where `id` is 1mA, and save the result in a variable named `vf`. After simulating you should see the console report a forward voltage of 0.632V.

Exercise 2: Diode AND Gate

Now create a file named `netlists/diode_AND.sp` and implement the circuit shown below. Make sure that your netlist has these features:

- The top line is a title comment to identify the circuit.
- The second line is a `.include` statement to load the model parameters.



More Advanced Exercises

In the remaining exercises, you will model and simulate all of the diode circuit experiments that you will perform in the laboratory. For each circuit you will make a netlist and a testbench. Some of the testbenches require performing several transient simulations at different frequencies, amplitudes and/or offset voltages. As a reference to help you get started with the more complex testbench designs, an example is provided in these files:

- `netlists/superdiode.sp`
- `tests/superdiode_test.sp`

Please study those files, run the simulation, and use them as a guide for setting up your own simulation files.

Simulate the AND gate

Make a testbench named `tests/diode_AND_test.sp` with these features:

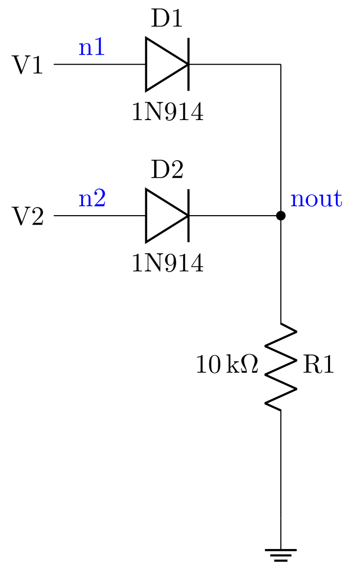
- Top line is a title comment.
- Second line is a `.include` to load the `diode_AND.sp` netlist.
- A `.control` section to simulate the experiments described in the diode lab procedures:
  - Set V1 to 3V and sweep V2 from 2 to 4 volts in steps of 0.5V.
  - Print the results to the console and log file so they can be later compared with your experiment.

- Copy the printed results and save them in a text file named `data/and_gate.txt`
- Plot the results and save a hardcopy to `plots/diode_AND_test.svg`

Exercise 3: Diode OR Gate

Now create a file named `netlists/diode_OR.sp` and implement the circuit shown below. Make sure that your netlist has these features:

- The top line is a title comment to identify the circuit.
- The second line is a `.include` statement to load the model parameters.



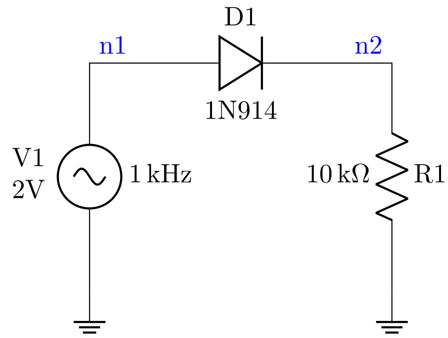
Simulate the OR gate

Make a testbench named `tests/diode_OR_test.sp` with these features:

- Top line is a title comment.
- Second line is a `.include` to load the `diode_OR.sp` netlist.
- A `.control` section to simulate the experiments described in the diode lab procedures:
  - Set V1 to 3V and sweep V2 from 2 to 4 volts in steps of 0.5V.
  - Print the results to the log file so they can be later compared with your experiment.
  - Copy the printed results and save them in a text file named `data/or_gate.txt`
  - Plot the results and save a hardcopy to `plots/diode_OR_test.svg`

Exercise 4: Simulate the Half-Wave Rectifier

Create a netlist named `netlists/half_wave_rectifier.sp` and implement the schematic shown below. Remember to import the model parameters.



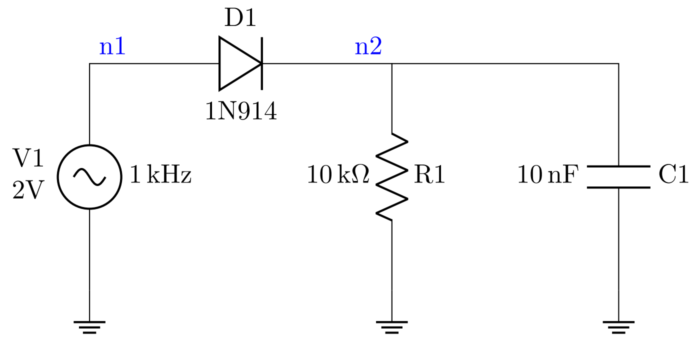
Testbench for the Half-Wave Rectifier

Create a testbench named `tests/half_wave_rectifier_test.sp` and perform these simulations:

- V1 is a **sinusoidal voltage source** with zero-to-peak amplitude of 2V.
- Perform a **transient** simulation for a duration of four signal periods.
- Measure the **maximum** and **minimum** voltages at `n2`, and record them in a file named `data/half_wave_rectifier.txt`
- Save a plot of the results in a hardcopy named `plots/half_wave_rectifier.svg`.

Exercise 5: Simulate the Peak Rectifier

Copy the half-wave rectifier netlist to a new file named `netlists/peak_rectifier.sp` and make one change, adding a capacitor in parallel with R1, so that it implements the schematic below.



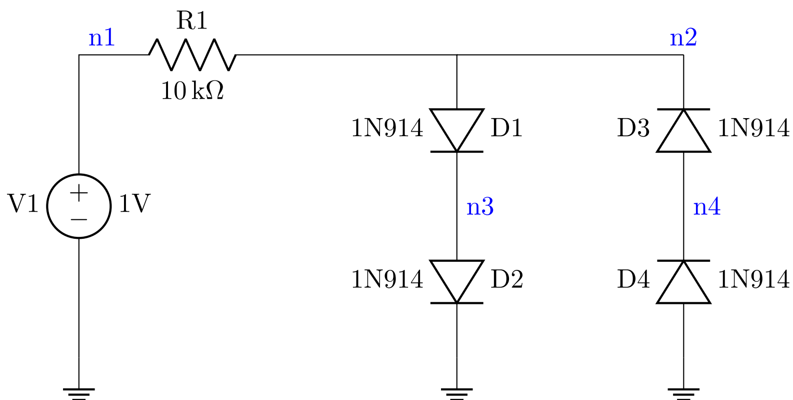
Testbench for the Peak Rectifier

Create a testbench named `tests/peak_rectifier_test.sp` and perform these simulations:

- At **each of three frequencies** 1kHz, 10kHz, and 100kHz, simulate:
  - V1 is a **sinusoidal voltage source** with zero-to-peak amplitude of 2V at the indicated frequency.
  - Perform a **transient** simulation for a duration of ten signal periods.
  - Measure the **peak-to-peak** voltage at `n2` **during the last period**, and record it in a file named `data/peak_rectifier.txt`
- Save plot of the transient results in hardcopy files named
  - `plots/peak_rectifier_1k.svg`
  - `plots/peak_rectifier_10k.svg`
  - `plots/peak_rectifier_100k.svg`

Exercise 6: Simulate the Limiter

Now create a netlist named `netlists/limiter.sp` implementing the schematic shown below.



Testbench for Limiter

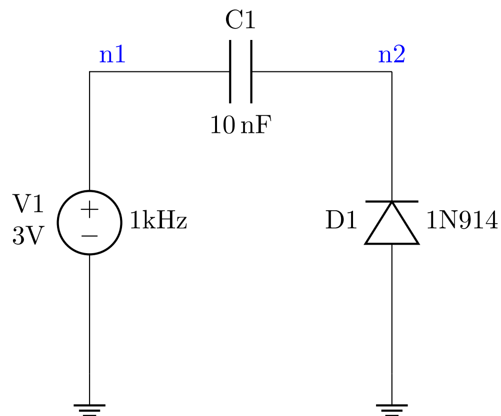
Create a testbench named `tests/limiter_test.sp` and perform these simulations:

- At **each of three amplitudes** 1V, 3V, and 4V, simulate:
  - V1 is a **sinusoidal voltage source** with the indicated zero-to-peak amplitude and a frequency of 1kHz.
  - Perform a **transient** simulation for a duration of four signal periods.
  - Measure the **peak-to-peak** voltage at `n2`, and record it in a file named `data/limiter.txt`
- Save plot of the transient results in hardcopy files named

- plots/limiter_1.svg
  - plots/limiter_3.svg
  - plots/limiter_4.svg

Exercise 7: Simulate the DC Restorer

Create a netlist named `netlists/dc_restorer.sp` and implement the schematic shown below.



Testbench for DC Restorer

Create a testbench named `tests/dc_restorer_test.sp` and perform these simulations:

- At **each of three offset voltages** 0V, 1V, and 2V:
  - V1 is a 1kHz sinusoid with zero-to-peak amplitude of 3V and the indicated offset.
- Perform a **transient simulation** for **ten signal** periods.
- Measure the **maximum** and **minimum** voltages at `n2`, and record them in a file named `data/dc_restorer.txt`
- Save plot of the transient results in hardcopy files named
  - `plots/limiter_1.svg`
  - `plots/limiter_3.svg`
  - `plots/limiter_4.svg`

Summary of Exercises

1. Diode I/V Simulation
2. AND gate
3. OR gate
4. Half-wave rectifier

5. Peak rectifier
6. Limiter
7. DC Restorer

Turning in Your Work

The preferred way to turn in your work is to use `git`. From the Linux terminal:

```
git add *
git commit -a -m "Submitting SPICE 3 assignment"
git push origin master
```

Alternatively you can upload a ZIP file to Canvas containing all your assignment files.