Engineering 43

Inside SPICE

Laurence Nagel
Omega Enterprises Consulting

SPICE HISTORY

- First Released in 1971 and announced in 1973 at the Sixteenth Midwest Symposium on Circuit Theory
  – Rapidly adopted by universities and industry in the early 1970's
- SPICE 2G6 became the de facto industry standard in the late 1970's
  – Why did this happen?

Origins of SPICE

- SPICE began as an innovative class project under the direction of Ron Rohrer in the academic year 1969-1970
- The class topic was circuit synthesis but became a class on circuit simulation
  - Students learned by doing --- They wrote a simulator!
- The final judge of success was Don Pederson
  - Larry Nagel was Liaison to Prof. Pederson

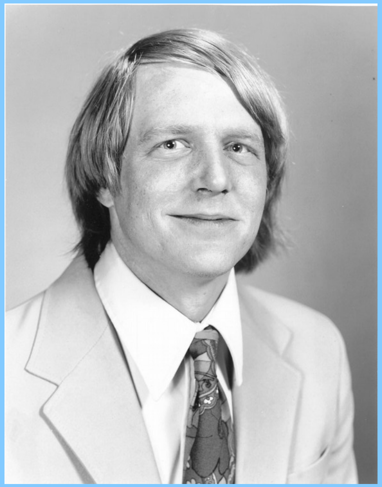
UC Berkeley SPICE Creators

- Don Pederson, Prof.
- Ron Rohrer, Prof.
- Bob Berry
- Shi-Ping Fan
- Frank Jenkins
- Larry Nagel
- Jesse Pipkin
- Steve Ratner
- Lynn Weber

Prof. Donald O. Pederson (September 30, 1925 - December 25, 2004) of UC Berkeley was a pioneering visionary in solid-state circuit design. He was the originator and, with his students, was the driving force behind the SPICE circuit simulation program, which has become the ubiquitous simulation program for detailed digital, memory, and analog circuit design at the transistor level throughout the industry, a position it has continued to maintain for over two decades to the present time. SPICE was released into the public domain, and quite literally spawned the EDA industry. His impact on the electronics industry was recognized most notably by the IEEE in 1998, when he was named the recipient of the IEEE Medal of Honor. He became a member of the National Academy of Engineering in 1974 and the National Academy of Sciences in 1982, as well as receiving many other awards.

More on SPICE from Inventor

- For More Info from someone who really Knows (Dr. Laurence Nagel) see:
  - http://www.omega-enterprises.net/The%20Origins%20of%20SPICE.html



Laurence W. Nagel

Ronald A. Rohrer

Donald O. Pederson

Why SPICE Was Successful

- FREE and in the Public Domain
- DC, AC, Transient, Noise, and Sensitivity Analyses in the same program
- Built-in models for diodes, bipolar transistors, MOSFETs, and JFETs
- Heavy use of SPICE by students led to many improvements in robustness
  - At the time, could handle fairly large circuits
- Written in fairly portable FORTRAN

SPICE2

- First released into the public domain in 1975
- Contained all features of SPICE
- Data structures totally revamped to incorporate dynamic memory allocation
- Thorough upgrade of DC convergence and transient numerical integration algorithms
- About 8,000 lines of FORTRAN
- SPICE2 was used as a platform for research that spawned hundreds of research projects

Industrial Use of SPICE2

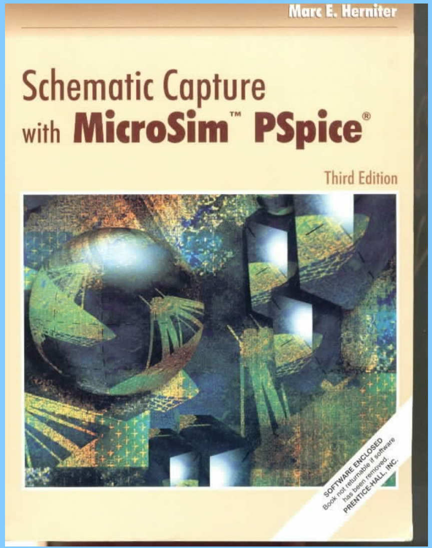
- Many industrial research centers adopted SPICE2 and developed proprietary versions of the program, including
  - Bell Labs (ADVICE), Texas Instruments (TISPICE), Motorola (MCSPICE)
- Shawn and Kim Hailey formed Meta Software (Campbell, CA) and modified a copy of SPICE 2E into the most successful version of a commercial SPICE known as HSPICE

SPICE3

- In 1989, SPICE3 was released into the public Domain
- This later version of SPICE3 was coded in the C language and utilized the more sophisticated data structures of C
- SPICE3 contains about 135,000 lines of C code
- The latest version, 3F5, was released in 1993

Commercial Use of SPICE3

- The Company **MicroSim** adapted a version of SPICE3 for the **most popular** of all SPICE programs - PSPICE
- Many other companies utilized SPICE3 as a platform for additional “alphabet SPICE”



Available SPICE Versions

- PSPICE – now owned by Cadence
- Ngspice – Open Source Version
- LTspice – by Linear Technology
- MultiSim – by National Instruments
- SIMetrix – SIMetrix Technologies
- Xyce – Parallel Electronic Simulator by Sandia National Labs

Inside SPICE

Break

Circuit Simulation Algorithms – Then and Now

- Circuit formulation techniques

  Construct a set of integro-differential equations describing the circuit

- Numerical integration techniques

  Solve the time-domain problem by a transient analysis of sequential timesteps

- Nonlinear equation techniques

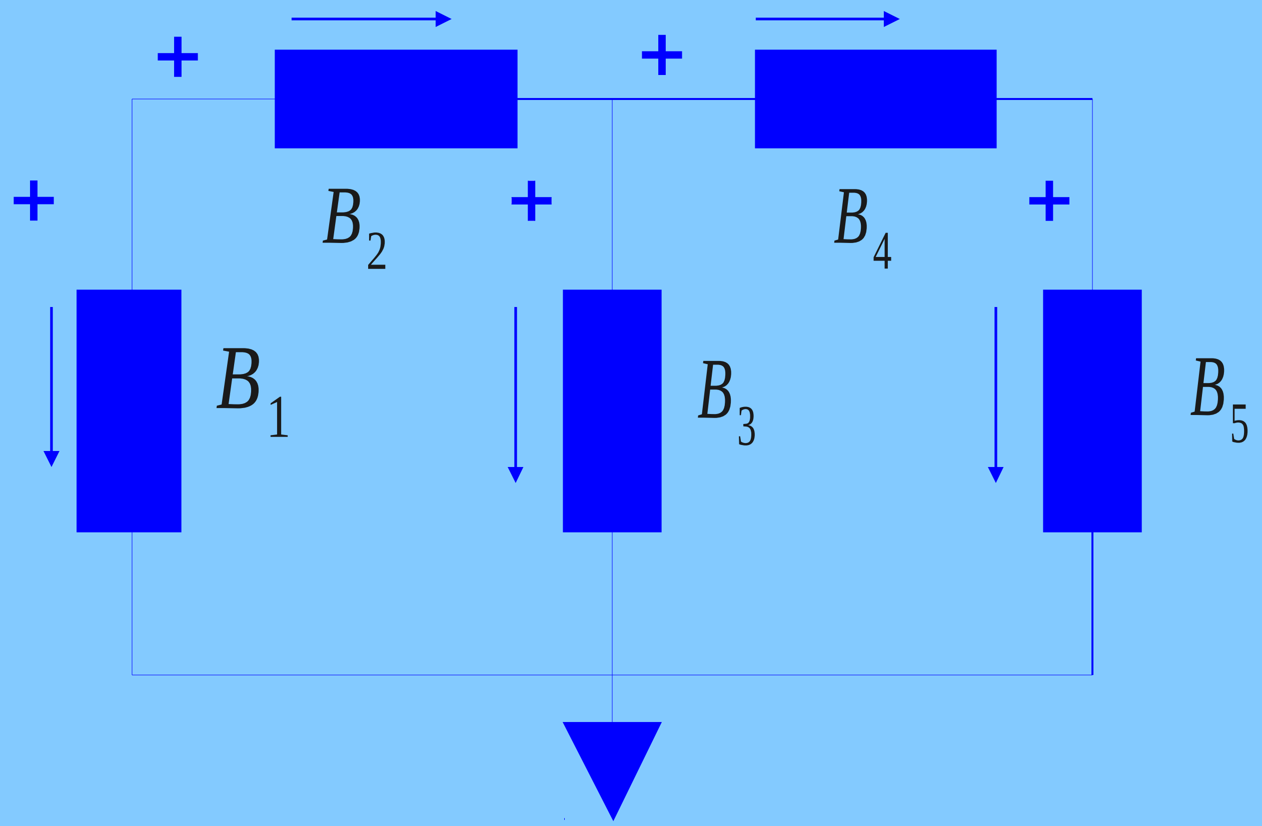
  Solve the set of nonlinear equations at each timestep by an iterative sequence of linear equations

- Linear equation techniques

  Solve the linear equations at each nonlinear iteration

Circuit Formulation

Represent circuit as an interconnection of branches



Circuit Formulation Constraints

- Branch constituent relations (BCR)

  Determines the branch current or the branch voltage as a function of circuit variables and environmental variables (time, temperature, etc.)

- Kirchhoff's Voltage Law (KVL)

  The sum of voltages around each loop must equal zero

- Kirchhoff's Current Law (KCL)

  The sum of currents at each node must equal zero

Branch Constituent Relations

Current-defined branches

$$I_b = f(V, I, t, T) \qquad I_b = \frac{dQ}{dt}(V, I, t, T)$$

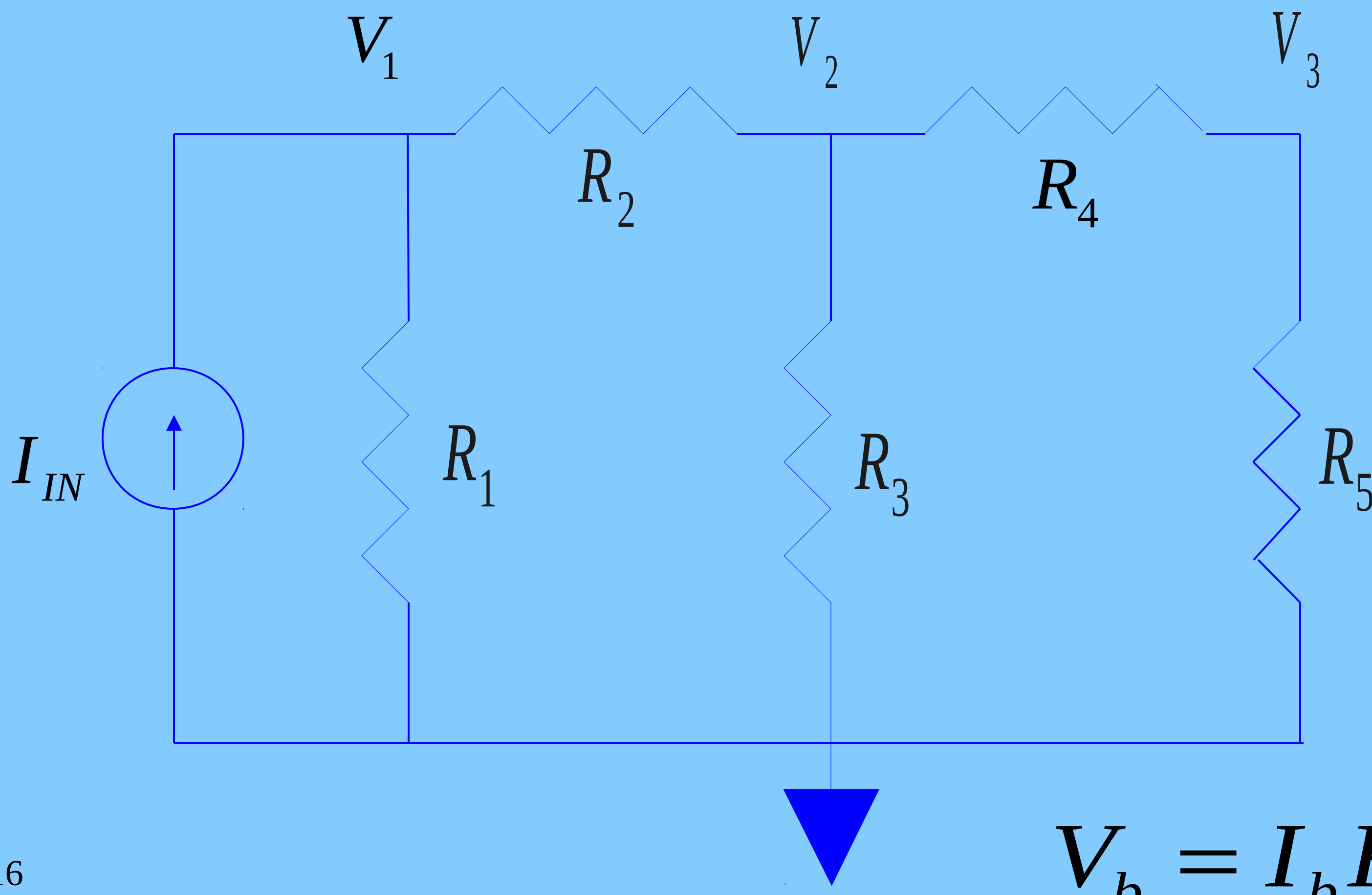
Voltage-defined branches

$$V_b = f(V, I, t, T) \qquad V_b = \frac{d\phi}{dt}(V, I, t, T)$$

Nodal Analysis

- Works only if circuit contains only current-defined branches
- Choose *n-1* node voltages as the unknown vector, automatically satisfying KVL
- Write *n-1* KCL equations for the *n-1* node voltages

DC Nodal Analysis Example



$$V_b = I_b R$$

Kirchhoff’s Current Laws

$$\frac{V_1}{R_1} + \frac{(V_1 - V_2)}{R_2} = I_{IN}$$

$$\frac{(V_2 - V_1)}{R_2} + \frac{V_2}{R_3} + \frac{(V_2 - V_1)}{R_4} = 0$$

$$\frac{(V_3 - V_2)}{R_4} + \frac{V_3}{R_5} = 0$$

The DC Nodal Equations

$$\begin{bmatrix} \frac{1}{R_1}+\frac{1}{R_2} & -\frac{1}{R_2} & 0 \\ -\frac{1}{R_2} & \frac{1}{R_2}+\frac{1}{R_3}+\frac{1}{R_4} & -\frac{1}{R_4} \\ 0 & -\frac{1}{R_4} & \frac{1}{R_4}+\frac{1}{R_5} \end{bmatrix} \begin{pmatrix} V_1 \\ V_2 \\ V_3 \end{pmatrix} = \begin{pmatrix} I_{IN} \\ 0 \\ 0 \end{pmatrix}$$

Frequency Domain Analysis

$$I = \frac{dQ}{dt}(V) \quad => \quad i(j\omega) = j\omega \frac{dQ}{dV} v(j\omega)$$

$$V = \frac{d\phi}{dt}(I) \quad => \quad v(j\omega) = j\omega \frac{d\phi}{dI} i(j\omega)$$

Frequency Domain Analysis

- In DC Analysis, there are *n-1* real linear equations to solve
- In AC (Frequency Domain) Analysis, there are *n-1* complex linear equations to solve, or *2(n-1)* real linear equations to solve

The Nodal Equations

$$Y_n V_n = I_s$$

$\boldsymbol{Y}_n$: Nodal Admittance Matrix

$\boldsymbol{V}_n$: Node Voltage Vector

$\boldsymbol{I}_s$: Current Excitation Vector

Modified Nodal Analysis

- Works with circuits with both current-defined branches and voltage-defined branches
- Include in the unknown vector *n-1* node voltages and the current in each voltage-defined branch
- Write *n-1* equations for KCL and a BCR for each voltage-defined branch

Modified Nodal Analysis

$$\begin{bmatrix} Y_n & 0 \\ 1 & Z_{bv} \end{bmatrix} \begin{pmatrix} V_n \\ I_{bv} \end{pmatrix} = \begin{pmatrix} I_s \\ V_s \end{pmatrix}$$

- Z_{bv} : V-def branch impedance matrix
- I_{bv} : V-def branch current vector
- V_s : Voltage Excitation Vector

Linear Solution Techniques

- Solve the Circuit Equations (usually Nodal Equations)
  - Gaussian Elimination
  - LU Factorization
- Circuit Equations are very sparse if chosen properly, so Sparse-Matrix techniques are utilized
- Nodal Equations are well conditioned, other formulations may need pivoting or heuristics

Nonlinear Solution Techniques

$$F(x+\delta x) \approx F(x) + \frac{dF}{dx}(x)\,\delta x$$

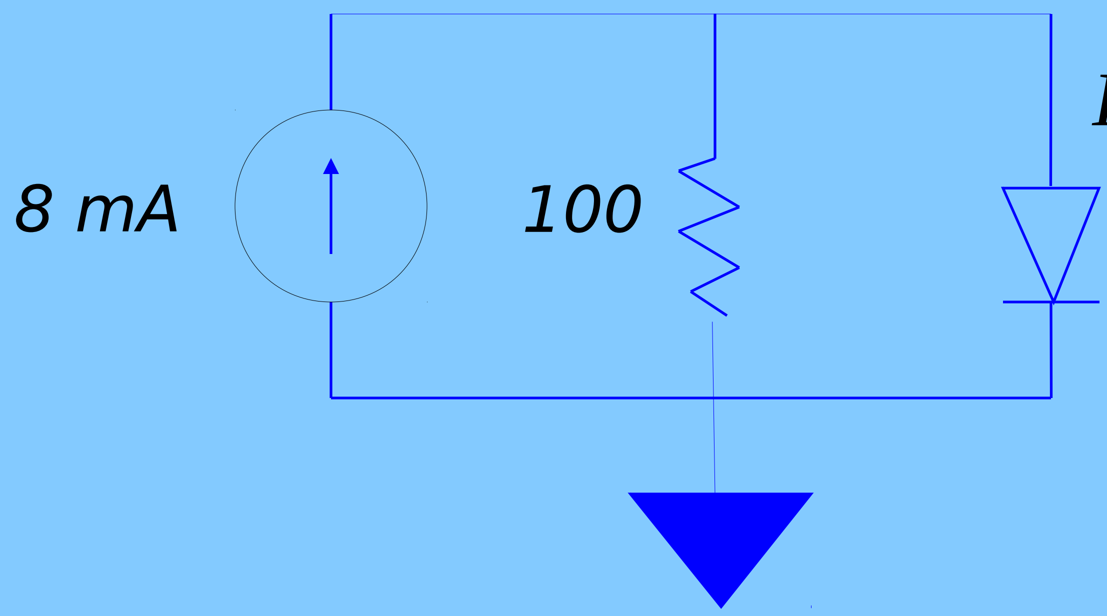
- Each nonlinear BCR is approximated by a linearized equivalent
- Newton-Raphson algorithm linearizes using Taylor Series expansion
- Process continues until $\delta x = 0$

Ideal Diode Example

$$I_d = I_S\left[\exp\left(\frac{V_d}{V_T}\right) - 1\right]$$

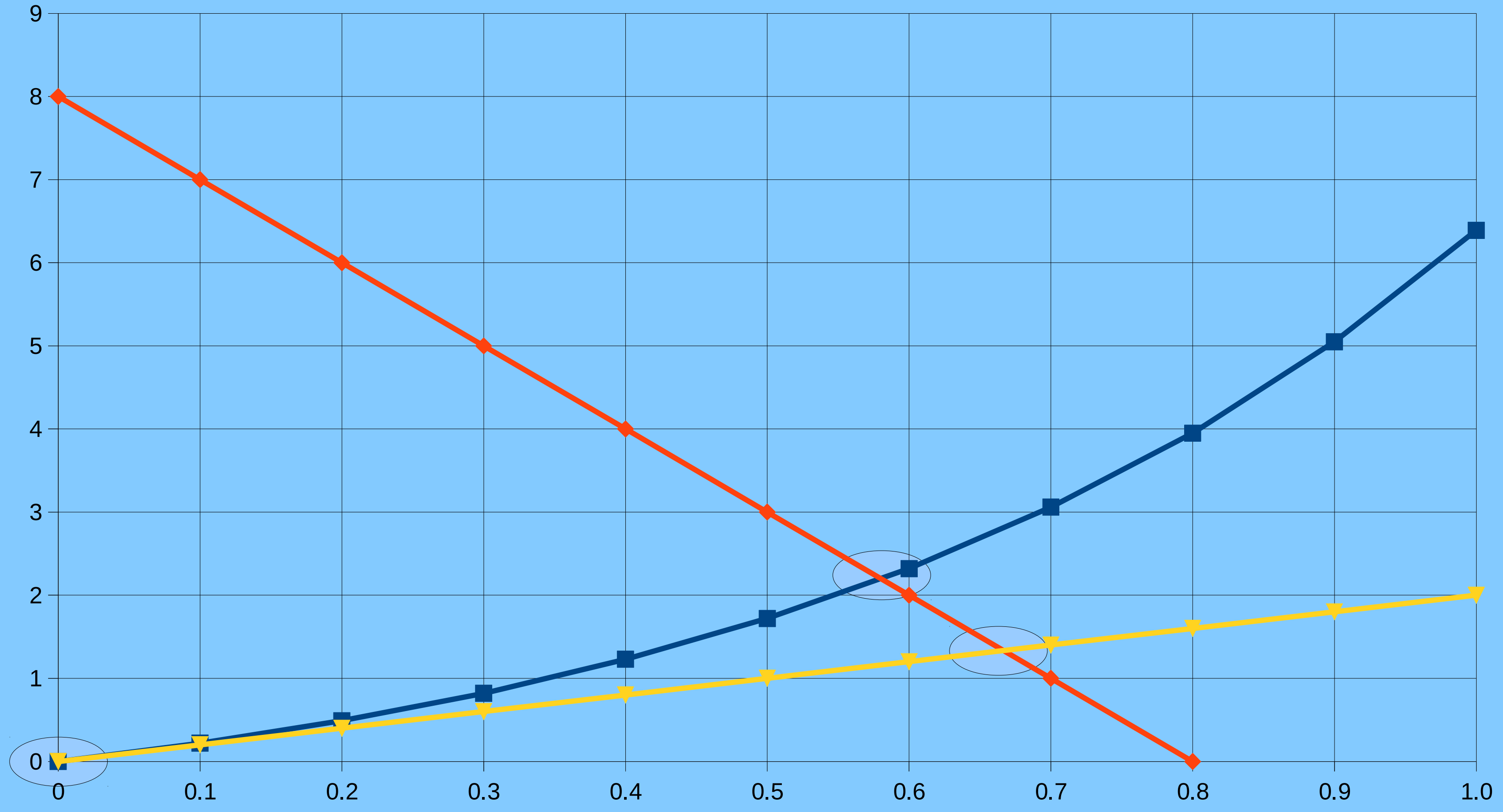
$$I_d + \delta I_d \approx I_S\left[\exp\left(\frac{V_d}{V_T}\right) - 1\right] + \frac{I_S}{V_T}\exp\left(\frac{V_d}{V_T}\right)\delta V_d$$

Ideal Diode Example



$$I_D = \exp(2V_D) - 1$$

Diode Example



Nonlinear Analysis Issues

- Quadratic Convergence

  SPICE will always converge, as long as the initial conditions are close enough to the correct solution

- Device models must have continuous derivatives (at least to second order)

- Device models must give reasonable results for unreasonable conditions

- Convergence aids

  Source stepping, GMIN stepping, pseudo-transient analysis, homotopy

Numerical Integration

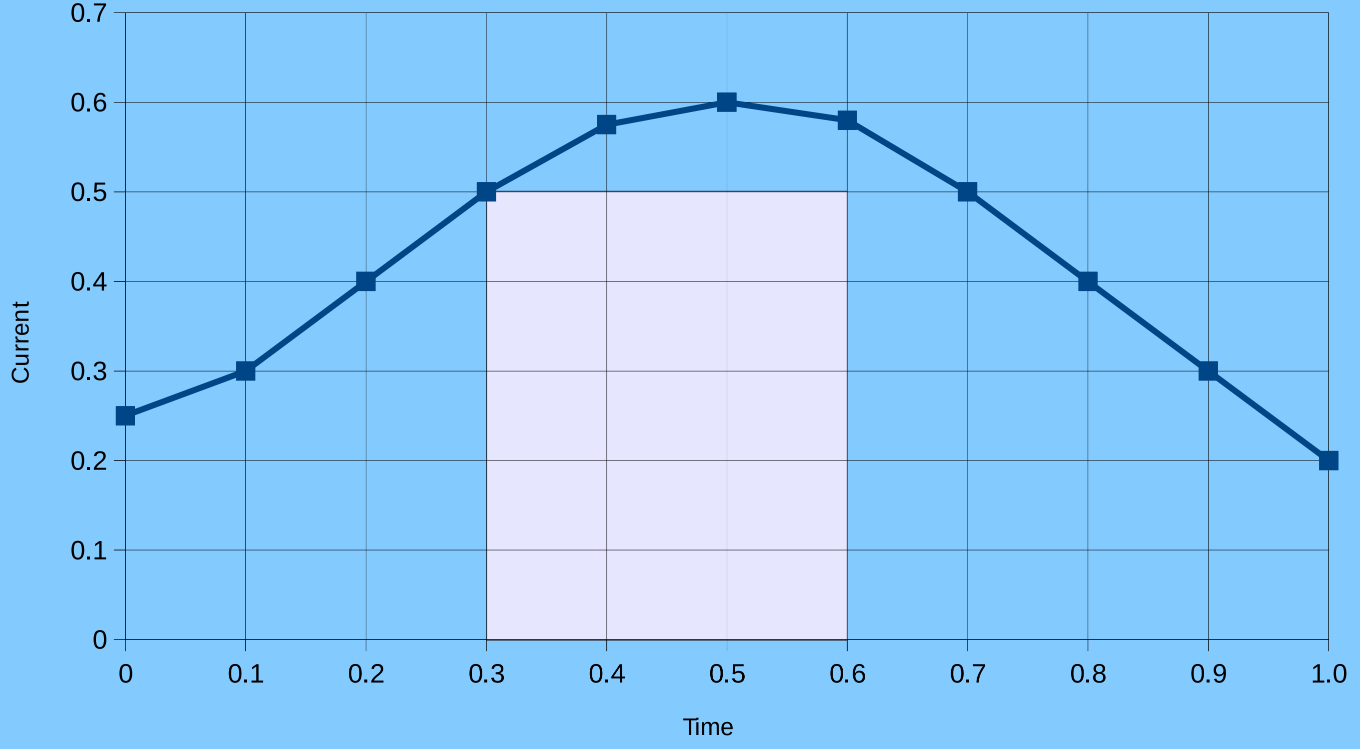
- Approximate the integral function with a polynomial
- Accuracy of the approximation, known as truncation error, depends upon the timestep and the order of the polynomial

Explicit Forward Euler Integration

$$\int_{t}^{t+h} i(t)\,dt \approx h\, i(t)$$

- Explicit: Integral approximation includes only past function value(s)

Forward Euler Integration

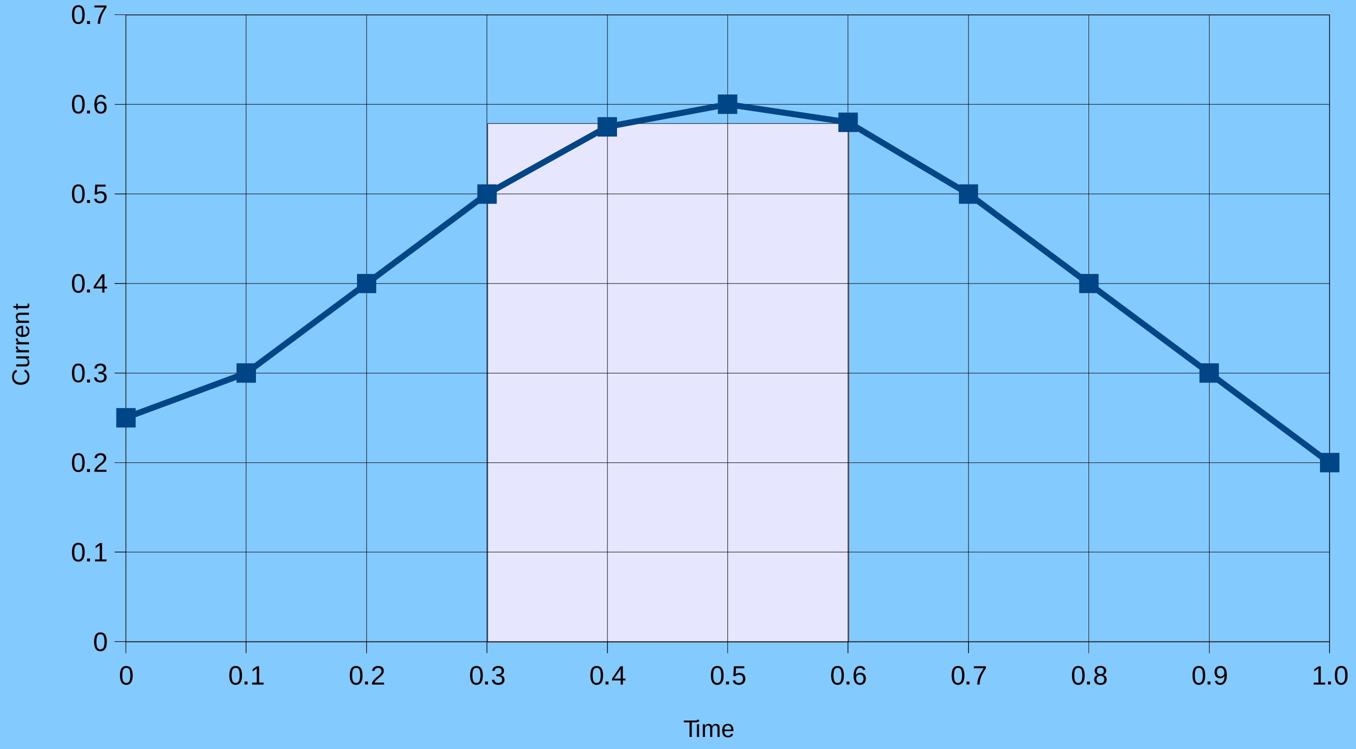


Implicit Backward Euler Integration

$$\int_{t}^{t+h} i(t)\,dt \approx h\,i(t+h)$$

- Implicit: Integral approximation includes both past and future function value(s)

Backward Euler Integration



Backward Euler Integration for Ideal Capacitor

$$i(t) = C\frac{dv}{dt}(t)$$

$$\int_{t}^{t+h} i(t)\,dt = C\left[v(t+h) - v(t)\right] \approx h\,i(t+h)$$

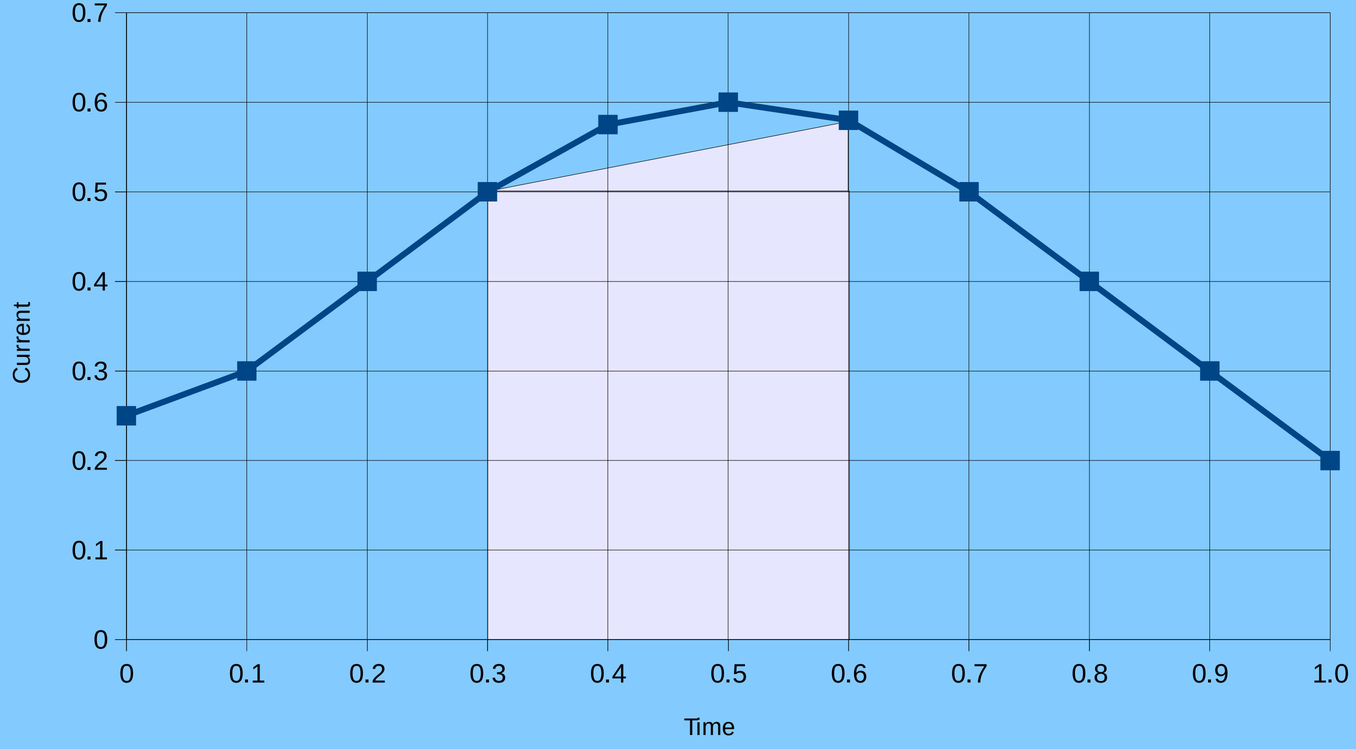
$$i(t+h) \approx \frac{C}{h}\left[v(t+h) - v(t)\right]$$

Trapezoidal Integration

$$\int_{t}^{t+h} i(t)dt \approx \frac{h}{2}\left[i(t)+i(t+h)\right]$$

- Implicit: Integral approximation includes both past and future function value(s)

Trapezoidal Integration



Trapezoidal Integration for Ideal Capacitor

$$i(t) = C\frac{dv}{dt}(t)$$

$$\int_{t}^{t+h} i(t)dt = C\left[v(t+h) - v(t)\right] \approx \frac{h}{2}\left[i(t) + i(t+h)\right]$$

$$i\left(t+h\right) \approx -i\left(t\right) + \frac{2C}{h}\left[v\left(t+h\right) - v\left(t\right)\right]$$

Numerical Integration Issues

- Accuracy
  - Higher order yields better accuracy
  - Smaller timestep yields better accuracy
- Stability – Stable circuits should yield stable solutions, and *vice versa*
  - Higher order methods are less stable
- Timestep control
  - Need algorithms to control timestep to maintain accuracy

SPICE Transient Algorithm (Recap)

- Outer loop for each time point (100 -1000)
  - Evaluate sources
  - Calculate integration coefficients
  - Timestep Newton Iteration (2-5 per timepoint)
    - Evaluate equations for each device (most of effort)
    - Construct Nodal Admittance matrix
    - Solve Nodal Admittance matrix
  - Evaluate time point truncation error and select next timestep

Inside SPICE

Break

Download PSPICE 16.6

- Visit the OrCAD Download Site

http://www.orcad.com/resources/orcad-downloads

- Go to the section titled

OrCAD 16.6 Lite Demo Software (Capture & PSpice Only)

- Click on the link

Download FREE - OrCAD 16.6 demo software (646MB)

Download PSPICE 16.6

OrCAD 16.6 Lite Demo Software (Capture & PSpice Only)

Designers around the world rely on the powerful yet intuitive Cadence® OrCAD® personal productivity tools. OrCAD has a long history of providing individuals and teams with a complete set of technologies that offer unprecedented productivity, seamless tool integration, and exceptional value—the OrCAD 16.6 release continues with that tradition.

The OrCAD 16.6 demo software will let you experience all the features and functionality of the actual software*. So go ahead, discover how easy it is to use these state-of-the-art OrCAD technologies. * Limitations are in the size and complexity of the design.

The OrCAD 16.6 PCB Designer Lite includes demo versions of the following tools: OrCAD Capture, OrCAD Capture CIS, PSpice A/D, PSpice Advanced Analysis.

Download FREE - OrCAD 16.6 demo software (646MB) Includes OrCAD flow tutorial with example design files
(You must use the path/folder option in your zip tool when extracting this archive.)

Having issues with your download? Request a hard copy of the OrCAD 16.6 Lite DVD

Download PSPICE 16.6

- You then will have to fill our a questionaire saying you are a student at Chabot College
- If you succeed, you will receive an e-mail with a link telling you where to get the download

Download PSPICE 16.6

From: donotreply@cadence.com <donotreply@cadence.com>

Sent: Monday, April 13, 2015 5:05 PM

To: Laurence Nagel

Subject: Cadence.com Downloads

Please click on the following link to validate your email address and continue to download:

http://www.cadence.com/products/orcad/Pages/orcaddownloads.aspx?regid=f155556359674210b2b37e4ac49609bd&ver=2

Thank you.

Cadence.com Support Team

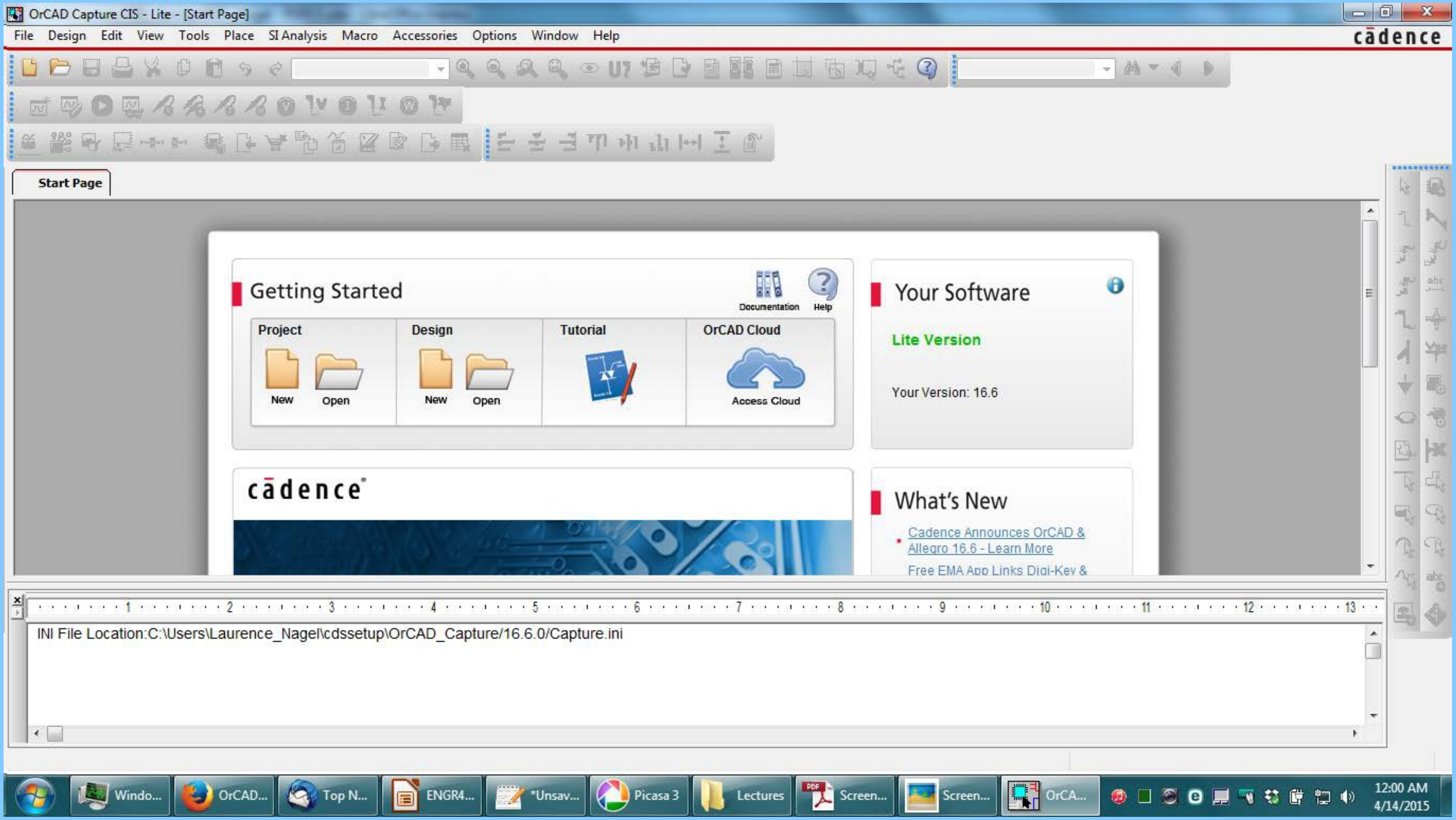
Download PSPICE 16.6

- The download is a ZIP file

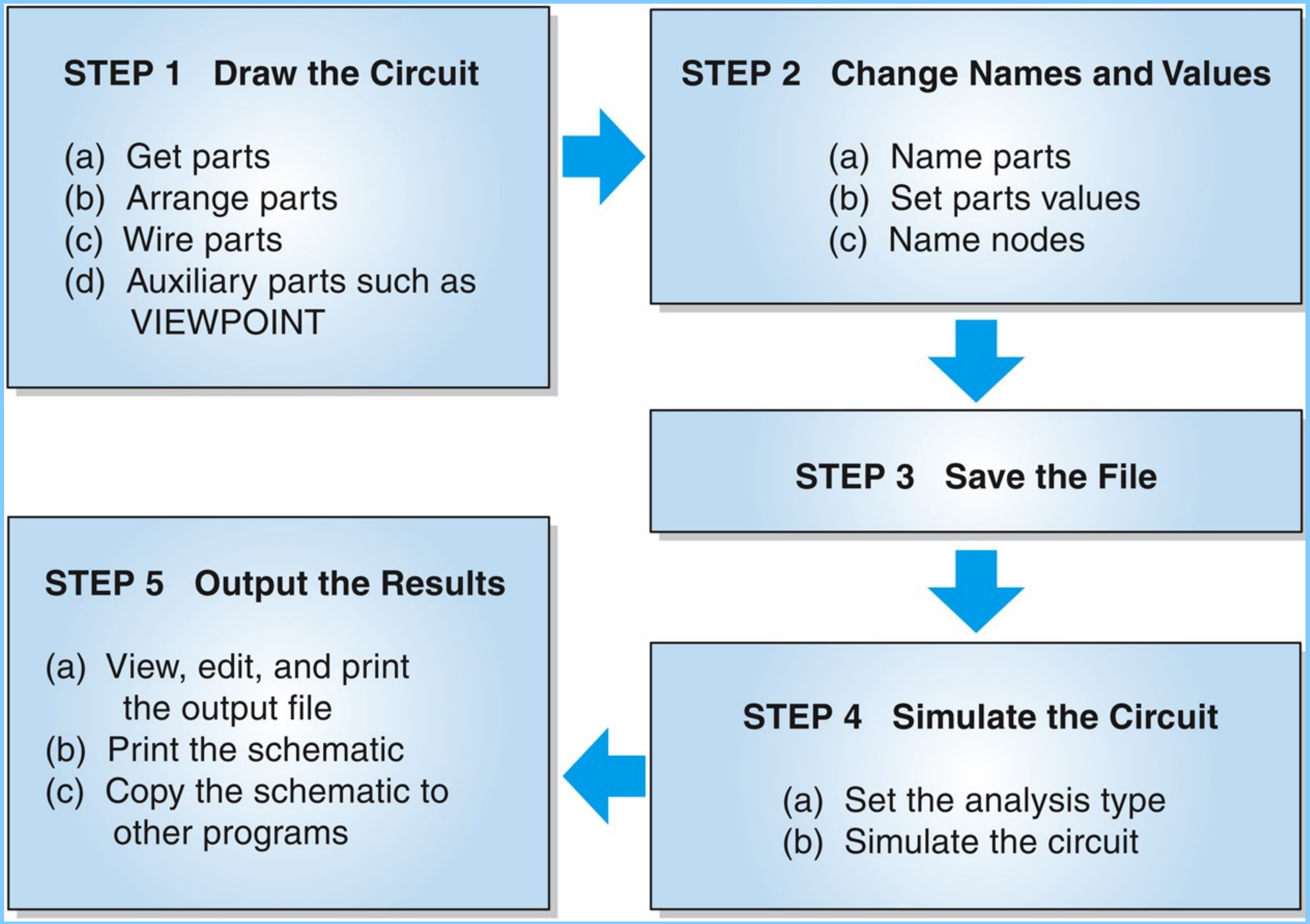
16.6_OrCAD_Lite_Capture&PSpice_Products_2014.zip

- Extract the contents of the ZIP file and double click on the file setup.exe
- You must have administrator privileges
- The install procedure will change the HOME environment variable. You may want to change it back

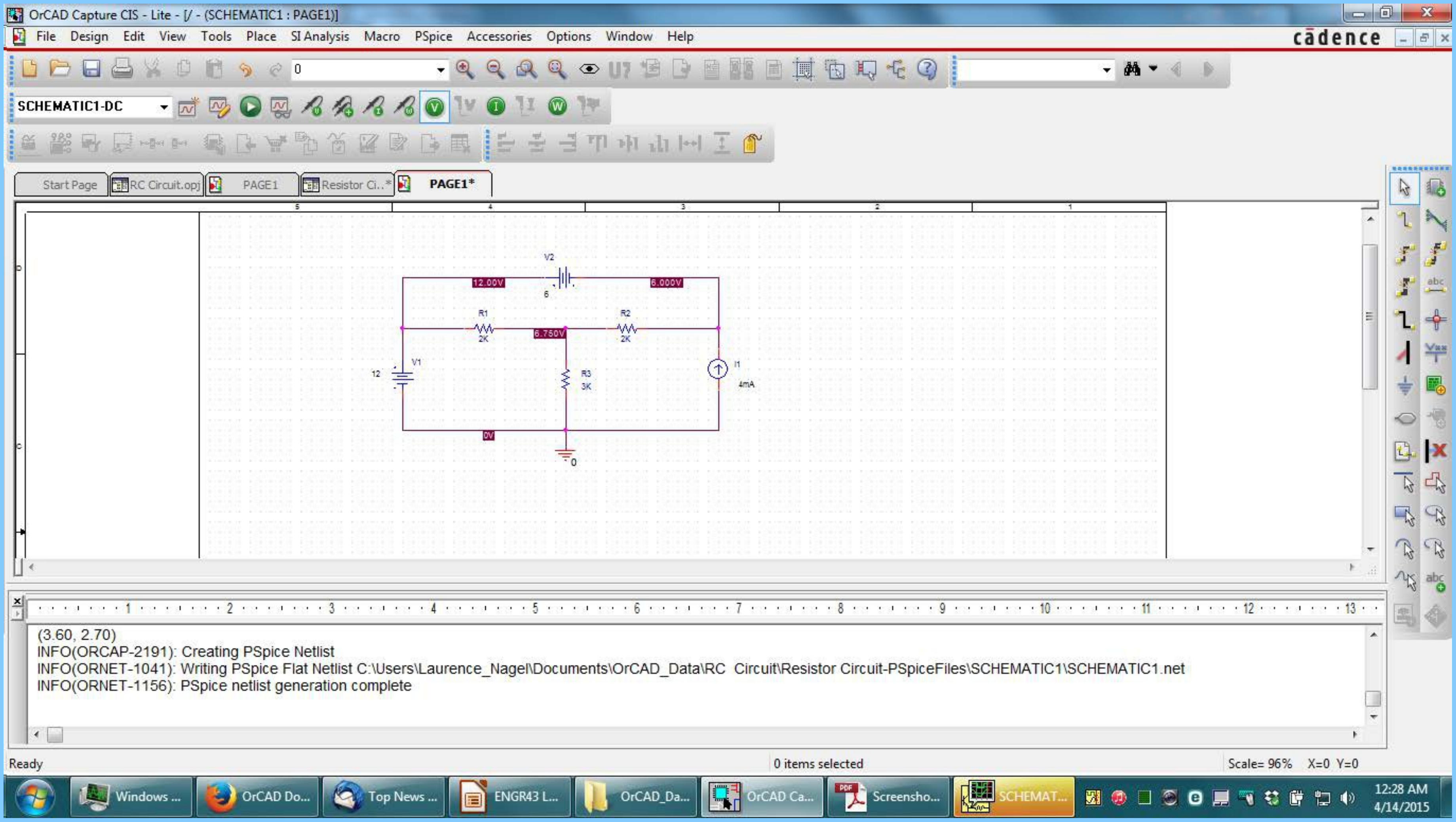
Successful Launch of PSPICE



SPICE Simulation Game Plan

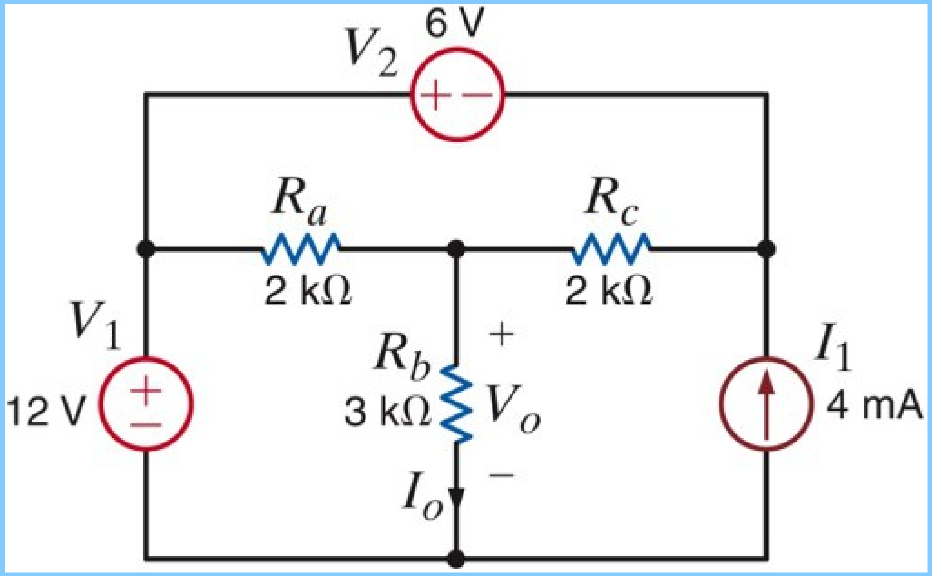


PSPICE Simulation of Resistive Circuit

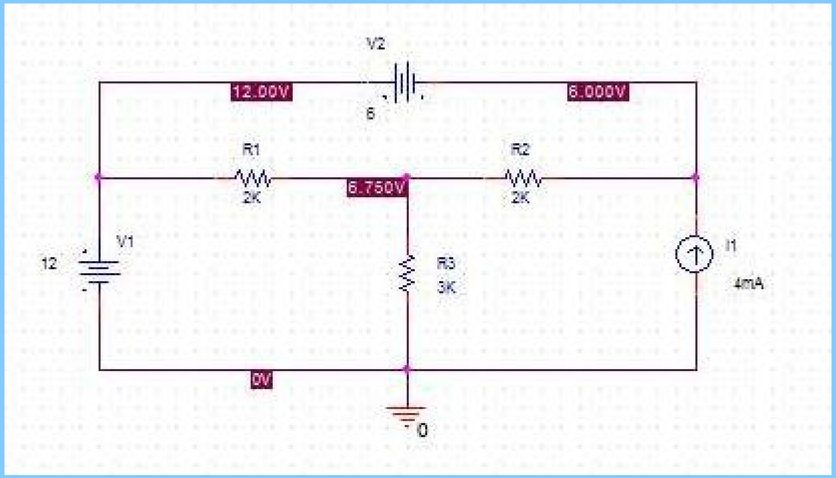


Complete PSPICE Schematic

Design Schematic

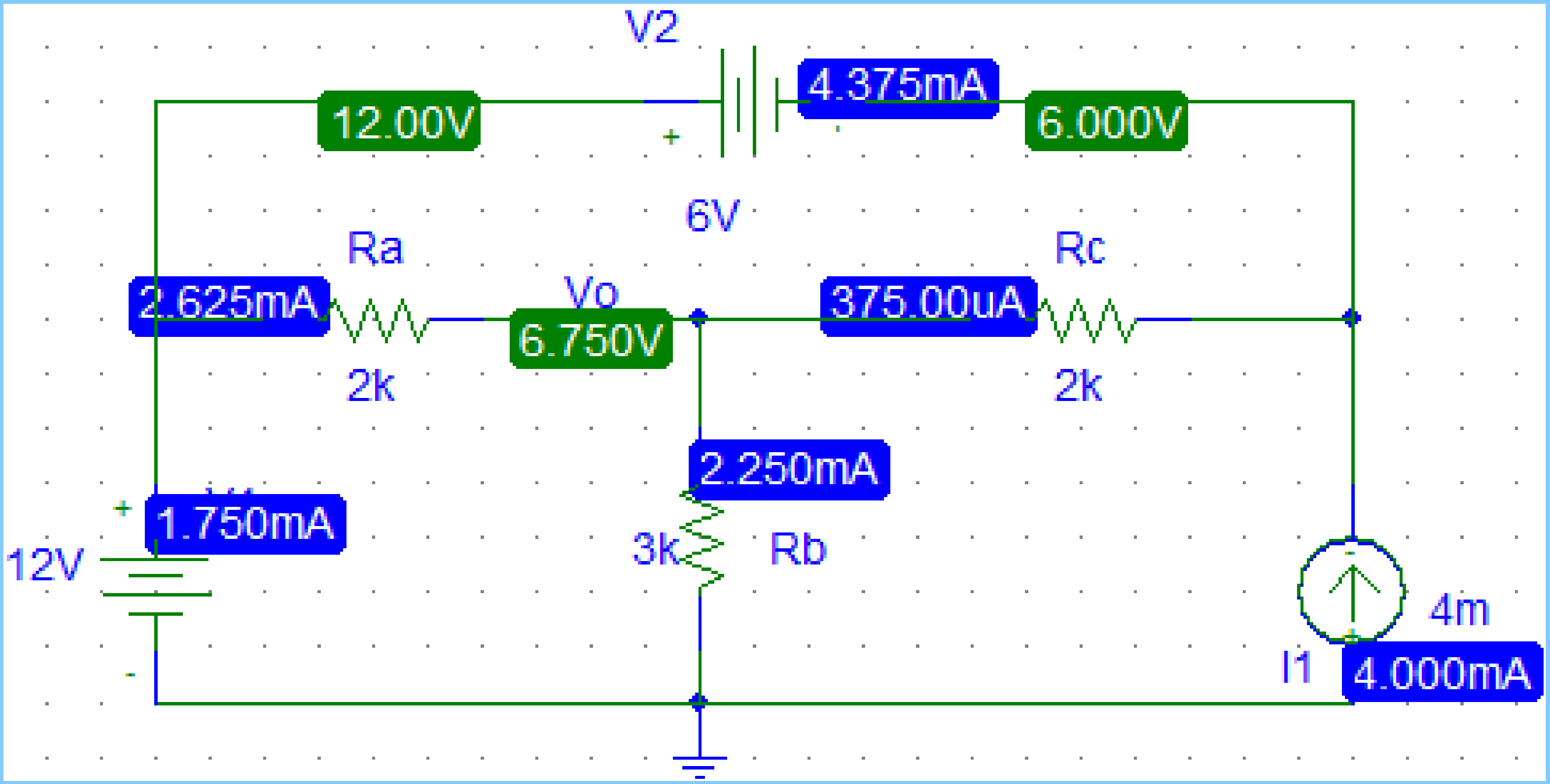


PSPICE Schematic

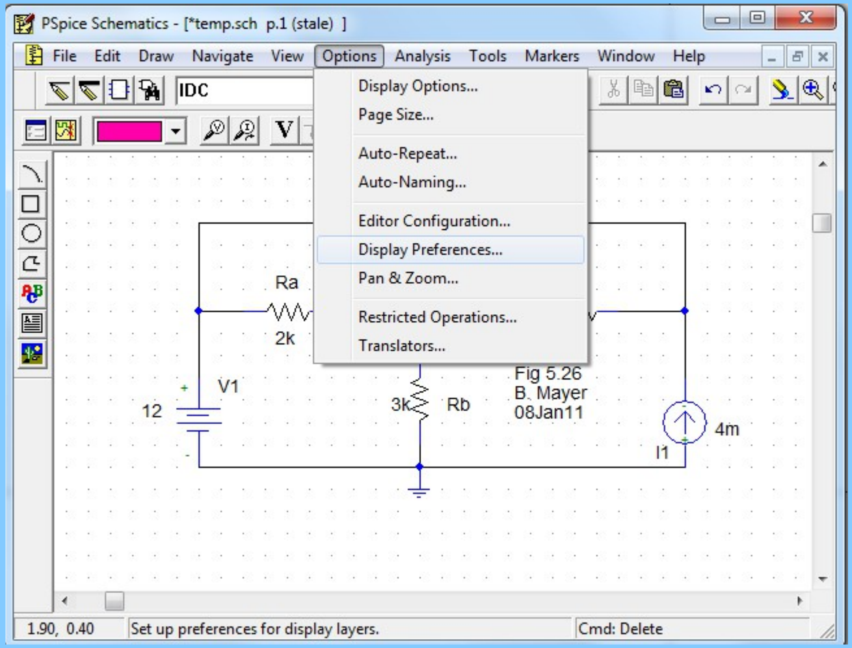


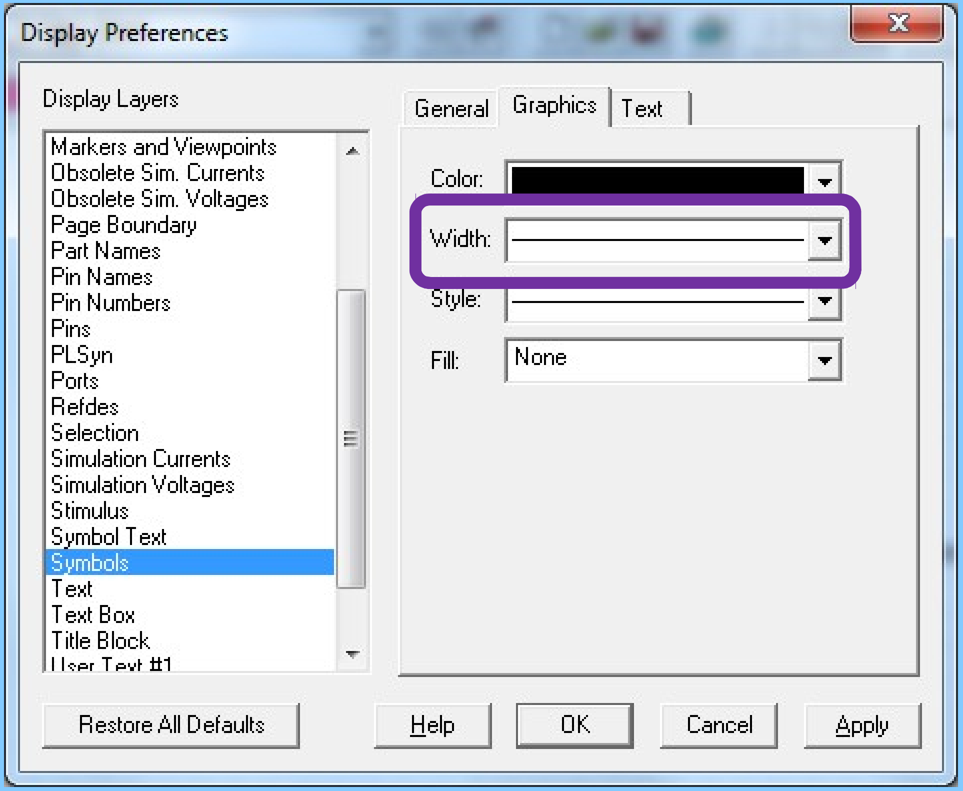
Simulation RESULT

- Showing ALL I's & V's



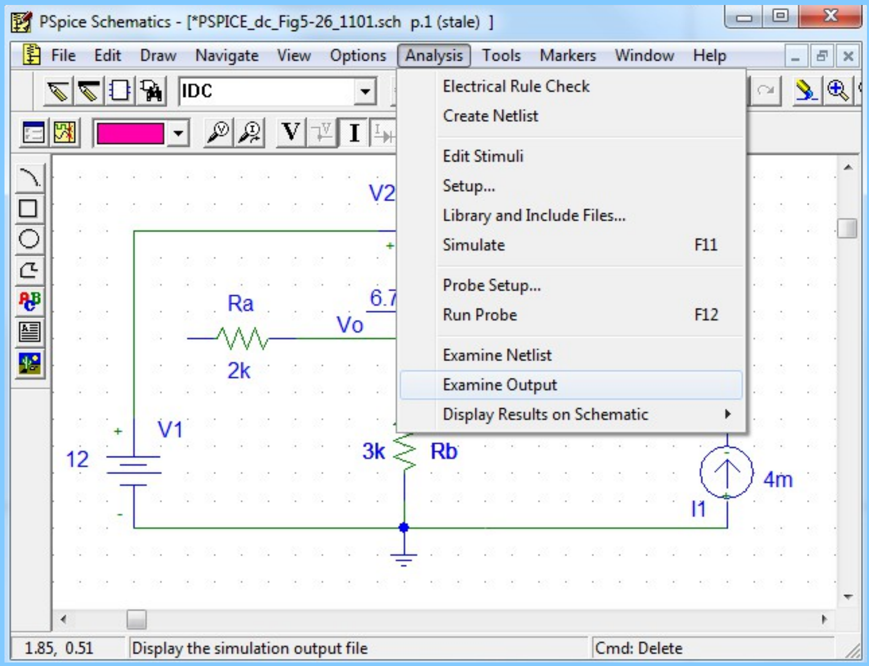
Changing Colors & LineWidths

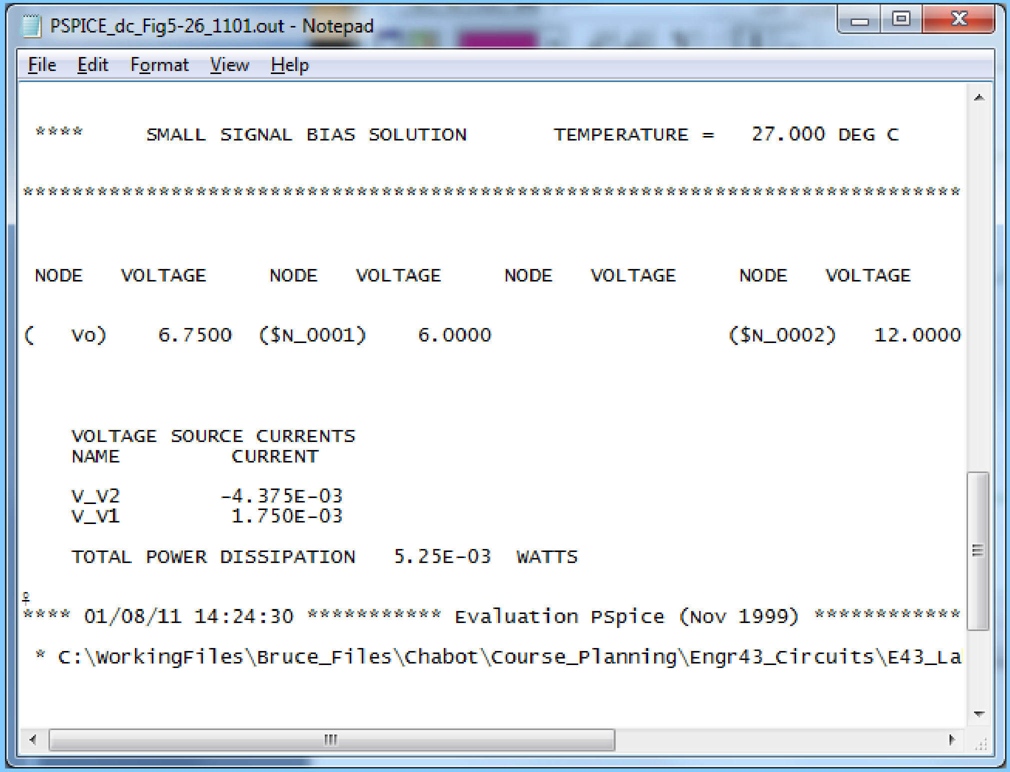




- Suggest making **Width** greater for all Circuit elements, including wires

Output File (MS NotePad)



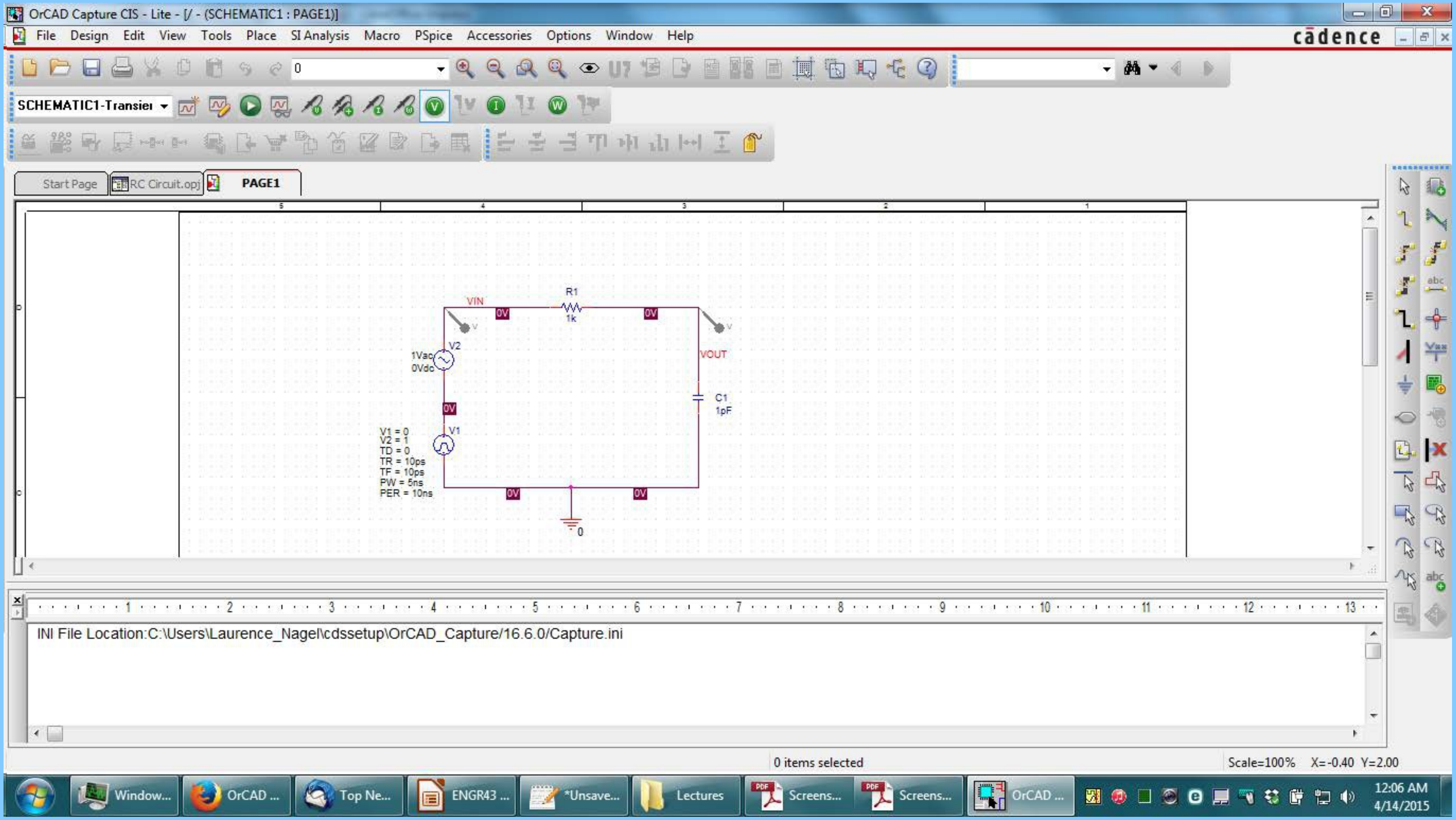


Full Output File

```
**** 01/08/11 14:24:30 *********** Evaluation PSpice
(Nov 1999) **************
C:\WorkingFiles\Bruce_Files\Chabot\Course_Planning\Engr4
3_Circuits\E43_Labs\E43_Labs_1101\PSPICE_dc_Fig5-
26_1101.sch
 ****     CIRCUIT DESCRIPTION
*********************************************************
* Schematics Version 9.1 - Web Update 1
•  Sat Jan 08 14:24:30 2011
** Analysis setup **
.OP
.OP
* From [PSPICE NETLIST] section of pspiceev.ini:
.lib "nom.lib“
.INC "PSPICE_dc_Fig5-26_1101.net“
**** INCLUDING PSPICE_dc_Fig5-26_1101.net ****
•  Schematics Netlist *
I_I1         0 $N_0001 DC 4m
R_Rc         Vo $N_0001  2k
V_V2         $N_0002 $N_0001 6
R_Ra         $N_0002 Vo  2k
V_V1         $N_0002 0 12
R_Rb         0 Vo  3k
**** RESUMING PSPICE_dc_Fig5-26_1101.cir ****
.INC "PSPICE_dc_Fig5-26_1101.als“
**** INCLUDING PSPICE_dc_Fig5-26_1101.als ****
* Schematics Aliases *
.ALIASES
I_I1            I1(+=0 -=$N_0001 )
R_Rc            Rc(1=Vo 2=$N_0001 )
V_V2            V2(+=$N_0002 -=$N_0001 )
R_Ra            Ra(1=$N_0002 2=Vo )
V_V1            V1(+=$N_0002 -=0 )
R_Rb            Rb(1=0 2=Vo )
_    _(Vo=Vo)
.ENDALIASES
```

```
**** RESUMING PSPICE_dc_Fig5-26_1101.cir ****
.probe
.END
**** 01/08/11 14:24:30 *********** Evaluation PSpice
(Nov 1999) **************
 *
C:\WorkingFiles\Bruce_Files\Chabot\Course_Planning\Engr4
3_Circuits\E43_Labs\E43_Labs_1101\PSPICE_dc_Fig5-
26_1101.sch
 ****     SMALL SIGNAL BIAS SOLUTION       TEMPERATURE =
  27.000 DEG C
*********************************************************
 NODE   VOLTAGE     NODE   VOLTAGE     NODE   VOLTAGE
 NODE   VOLTAGE
(   Vo)    6.7500  ($N_0001)    6.0000
($N_0002)   12.0000

    VOLTAGE SOURCE CURRENTS
    NAME         CURRENT

    V_V2        -4.375E-03
    V_V1         1.750E-03

    TOTAL POWER DISSIPATION   5.25E-03  WATTS
**** 01/08/11 14:24:30 *********** Evaluation PSpice
(Nov 1999) **************
C:\WorkingFiles\Bruce_Files\Chabot\Course_Planning\Engr4
3_Circuits\E43_Labs\E43_Labs_1101\PSPICE_dc_Fig5-
26_1101.sch
 ****     OPERATING POINT INFORMATION      TEMPERATURE =
  27.000 DEG C
*********************************************************
          JOB CONCLUDED

          TOTAL JOB TIME            0.00xxx
```

PSPICE Simulation of RC Circuit



PSPICE Simulation of RC Circuit

PSPICE Simulation of RC Circuit