以開放原始碼工具來學習 SPICE 之新式學習方法

*李博明
南臺科技大學電子工程系
*pmlee@mail.stust.edu.tw

摘要

現今大多數主修電子工程的學生都必須學習 EDA (Electronic Design Automation – 電子設計自動化)工具。其中 SPICE (Simulation Program with Integrated Circuit Emphasis) 是電子系學生所必須熟悉眾多工具中最重要的且最基礎的一個。然而，商用 SPICE 工具非常昂貴以致於大多數學生無法負擔。即便使用學校版本也有許多限制，例如功能閹割或是僅有有限的套數可以執行，這些限制或多或少的影響了學生的學習成效。在這篇論文中，我們將使用開放原始碼工具，也就是 NGSPICE 來發展 SPICE 學習流程，所提出之學習流程將會解除商用 SPICE 工具的限制。本論文中所提及之學習流程已經使用在實際課程上超過三年，在我們的實際教學測試中，學生及教授都能因此而獲益。

關鍵詞：電子設計自動化、開放原始碼

A Novel SPICE Learning Flow by Using Open Source Tools

***Po-Ming Lee**
Department of Electronic Engineering, Southern Taiwan University of Science and Technology

Abstract

Most students, whose major in EE (Electrical Engineering), are required to learn EDA (Electronic Design Automation) tools nowadays. SPICE (Simulation Program with Integrated Circuit Emphasis) is one of the most basic but essential tools that EE students must be familiar with. However, it is well known that commercial SPICE tools are very expensive that most students can not afford. Even through an academic version is available, there are some limitations such as reduced function version or limited software licenses. These limitations somehow restricted the study results more or less. In this paper, we will use an open source, namely, NGSPICE to develop a SPICE learning flow. The proposed learning flow will solve the restrictions in the commercial SPICE tools. This learning flow has been used in class for more than 3 years. In our field test, students and advisors were both benefited from the free alternative.

Keywords: Electronic Design Automation, Open Source

Received: Nov. 14, 2013; accepted: May, 2014.

*Corresponding author: P-M. Lee

I. Introduction

Full custom design flow is the fundamental SoC (System-on-Chip) implementation method in VLSI (Very Large Scale Integrated circuits) design. Fig. 1 represents a full custom design flow and its corresponding commercial EDA tools. Several tools are used in the flow, and the functions are described as the follows.

- Cadence [1] composer or Springsoft [3] ADP (Analog Design Platform) is used to create schematic entry.
- Synopsys [2] HSPICE or Cadence Spectre is used to perform circuit simulation.
- Cadence Analog Environment or Synopsys Cosmos Scope is used to observe simulation waveforms.
- Cadence Virtuoso or Springsoft laker is used to create layout.
- Mentor Graphics [4] Calibre is used to do DRC/LVS verifications.

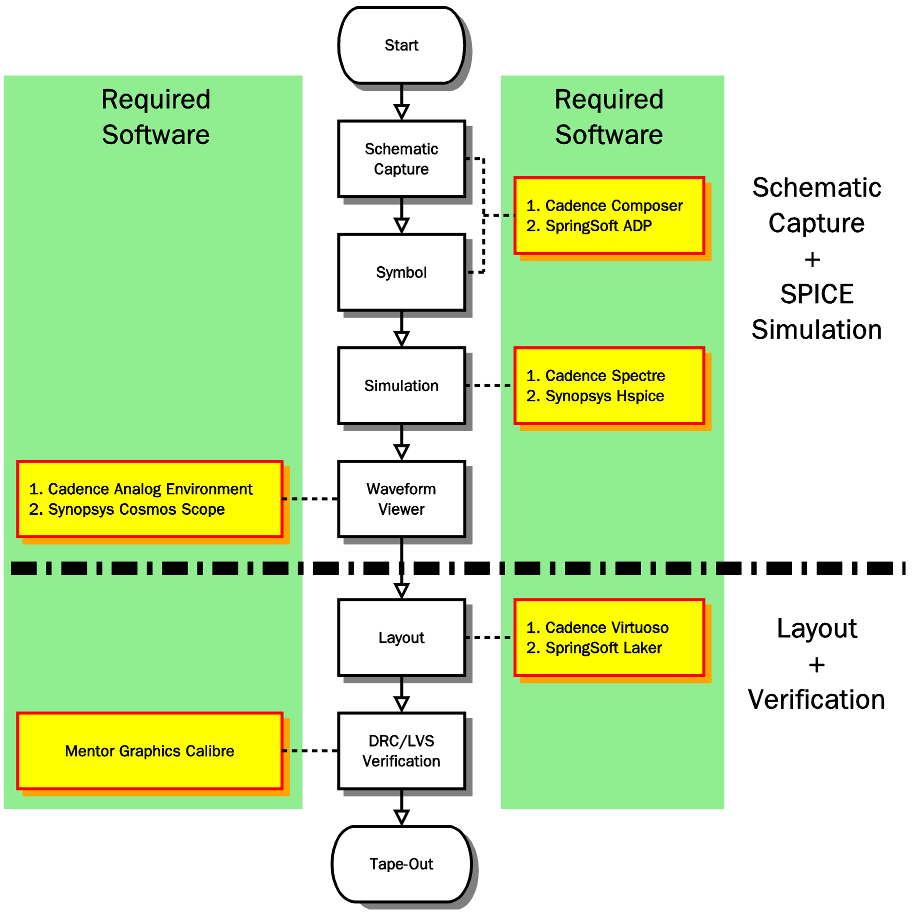


Fig 1 Full custom design flow with its corresponding EDA tools

It can be seen from Fig. 1 that the ‘Schematic Capture + SPICE Simulation’ part plays a very important role in the flow. Hence, it is crucial for students to master these tools. In order to train students to familiar with these EDA tools, series of classes are arranged for entry level students as well as senior level students. However, it is impractical for students who use commercial EDA tools as beginners. The reasons are as follows:

- Commercial EDA tools are usually deployed with limited licenses. It may not have sufficient licenses for all students.
- Commercial EDA tools must be deployed in the computer laboratory due to license check issue. However, the computer laboratory cannot open 24 hours a day and may not have enough capacity for all students.

- Senior level students are usually asked to do their graduation projects. On the other hand, entry level students are just asked to simulate and analysis circuits as their homeworks. In this case, the senior level students must have higher priority to use the mentioned commercial EDA tools and computer laboratory over the entry level students.

In short, commercial EDA tools are powerful but with variety limitations. It is impractical to let all students use commercial EDA tools to do 'Schematic Capture + SPICE Simulation' in Fig. 1. Hence, an open source alternative is proposed in this paper. The proposed alternative not only reduces the need of the commercial EDA tools, but also remains the study effect. Both students and advisors can be benefited from the proposed alternative.

II. The scope of this paper

In this paper, students are divided into two groups: entry level students and senior level students. As shown in Fig 2, the entry level students are those who begin to study electronic design while the senior level students are those who should complete their graduate projects. To be more specific, the entry level students means freshman and sophomore students and the senior level students means junior and senior students. Besides, the right side of Fig 2 are the courses for the corresponding students. It should be noted that the class 'Unix/Linux Operation System' is arranged for freshman as a precursor class since Unix/Linux is the major platform for almost all EDA tools. Students should learn how to use Unix/Linux Operation System, e.g., vi, CLI (Command Line Interface) usage, before they study EDA tools. Otherwise, the learning results can be very poor.

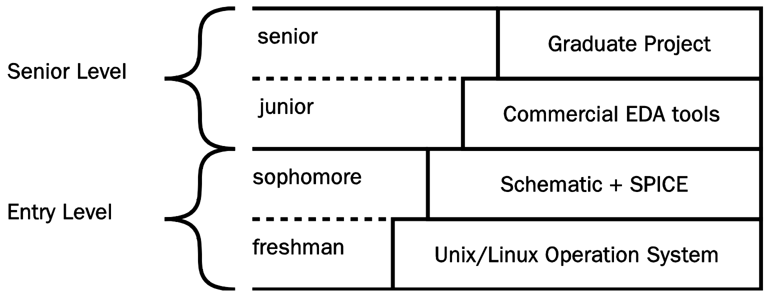


Fig 2 Definition of students and their study scopes

Commercial SPICE tools like HSPICE or Spectre are powerful tools used in the full custom design flow. However, students have to complete computers/licenses with other students in order to use SPICE due to the mentioned difficulties. As a result, the arrangement of the computer laboratory is always a headache problem for all advisors. Thus, the scope of this paper is to propose an open source alternative for the entry level students to learn SPICE.

Here, we would like to discuss the essence of SPICE learning. Our primary goal is to ensure that students have learned how to write SPICE language in order to perform circuit analysis and simulation, not to use commercial SPICE tools. SPICE simulator is just the tool we use, not the purpose we study. Hence, is it really necessary to use commercial SPICE tools with so many restrictions for students to learn SPICE? Or, we can discuss a completely different scenario: Will the driving schools provide BMW or Mercedes for their rookies to learn how to drive? Well, the answer is quite obvious. Using commercial SPICE tools for the entry level students to learn SPICE is just like using BMW/Mercedes for the rookies to learn how to drive.

If we can break the study myth of using commercial SPICE tools, a new, free SPICE simulator can than be introduced. This is where NGSPICE [7] meets our goal.

III. NGSPICE

1. Introduction

NGSPICE is the descendant of the legendary Berkeley SPICE [8] with a lot of improvements. It is also part of the famous gEDA project [9]. The major advantages of NGSPICE can be categorized as follows,

- NGSPICE is free software, students are freely to use it without any charge.
- No license check is required which means unlimited number of students can use NGSPICE simultaneously.
- NGSPICE uses the same SPICE language as well as other commercial SPICE simulator. Those who study NGSPICE can migrate to commercial SPICE simulator flawlessly.
- NGSPICE supports both windows and linux platforms, students can run NGSPICE on their own computers. This will reduce the demand of computer laboratory and hence relief the headache of the advisors.

In short, NGSPICE can be used for the entry level students who begin to study SPICE. In the following sections, we will provide further discussions to see why NGSPICE can be used as a better open source alternative.

2. Simulation of R L C Circuits

Using NGSPICE to simulate RLC (Resistor, Inductor, and Capacitor) circuits is quite strait forward. Thus, we do not address this issue in depth in this paper.

3. Simulation of CMOS Circuits

Using SPICE tools to simulate CMOS (Complementary Metal Oxide Semiconductor) circuits is essential nowadays. Major Fabs like TSMC [10] or IBM [11] all provide CMOS process which realized most of the chips today. Thus, it is necessary for students to know how to simulate CMOS circuits using SPICE tools. In order to simulate CMOS circuits, SPICE simulators like HSPICE or NGSPICE are not just enough. The SPICE models we used in the simulations are the key to determine whether the simulation is correct or not. After all, it is pointless to simulate CMOS circuit without appropriate SPICE models. Neither students nor advisors will know if their simulations are accurate or not. However, commercial SPICE models are proprietary which can not be distributed freely. In this case, we need to have not only reliable but also freely available CMOS SPICE models for the students to use. The question is: Are there any freely available SPICE models for the students to run CMOS circuit simulations? Luckily, MOSIS [12] provides a great help regarding this issue.

4. Obtain Suitable CMOS Model

MOSIS is well known as an organization that provides chip implementation service for the academy. However, few people knew that MOSIS also provides measured SPICE parameters from the tapped-out wafer they received. The measured SPICE parameters are in different CMOS processes from 0.18 μm through 0.50 μm and are freely available on the MOSIS website. These SPICE models can be used in NGSPICE to perform CMOS circuit simulations. Fig. 3 is the ‘Test Data’ link provided by MOSIS on their website [13]. We can access different SPICE models in the ‘Test Data’ page, and the SPICE model we need can be then be abstracted.

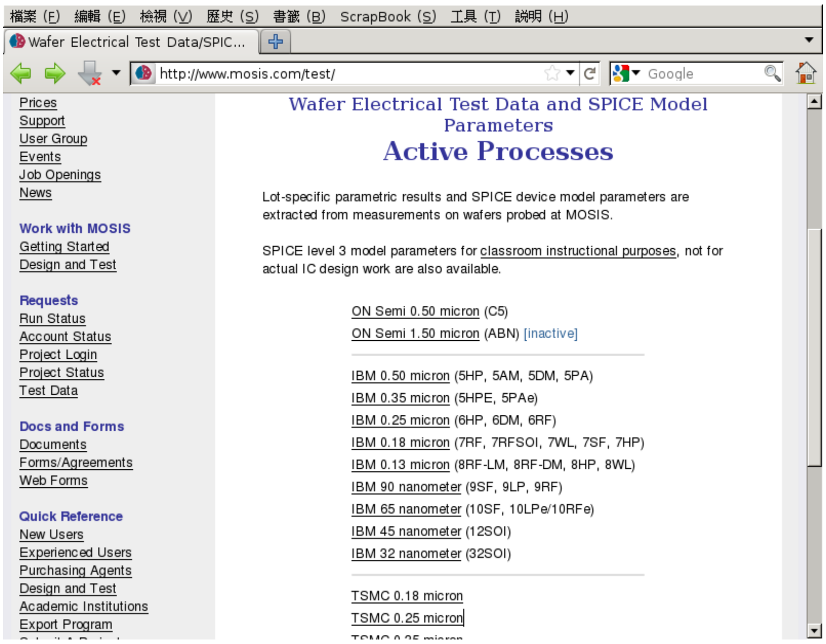


Fig 3 MOSIS Test Data

Listing 1 is an example of the extracted SPICE model from the MOSIS website. It is based on TSMC 0.35 μm CMOS process. Please be noted that the entire SPICE model is not shown due to page limit. This SPICE model is saved as 'tsmc035.model' for later use.

Listing 1 SPICE model 'tsmc035.model' extracted from MOSIS

```
* DATE: Sep 27/05
* LOT: T55Z WAF: 2002
* Temperature_parameters=Default
.MODEL CMOSN NMOS (                          LEVEL    =  49
+VERSION    =  3.1       TNOM    =  27       TOX  =  7.8E-9
+XJ  =  1E-7          NCH  =  2.2E17       VTH0=  0.5380719
+K1  =  0.5836113     K2   =  0.0141372    K3   =  80.6140862
...
The rest of the NMOS model is omitted.
)

.MODEL CMOSP PMOS (                          LEVEL    =  49
+VERSION    =  3.1       TNOM    =  27       TOX  =  7.8E-9
+XJ      =  1E-7      NCH  =  8.52E16  VTH0=  -0.6614536
+K1      =  0.4432406 K2   =  -0.0123618   K3   =  55.5705007
...
The rest of the PMOS model is omitted.
)
```

5. Using The Abstract SPICE Model

Once TSMC 035 model file is obtained, it is pretty easy to use this SPICE model. Simply add a line as the follows

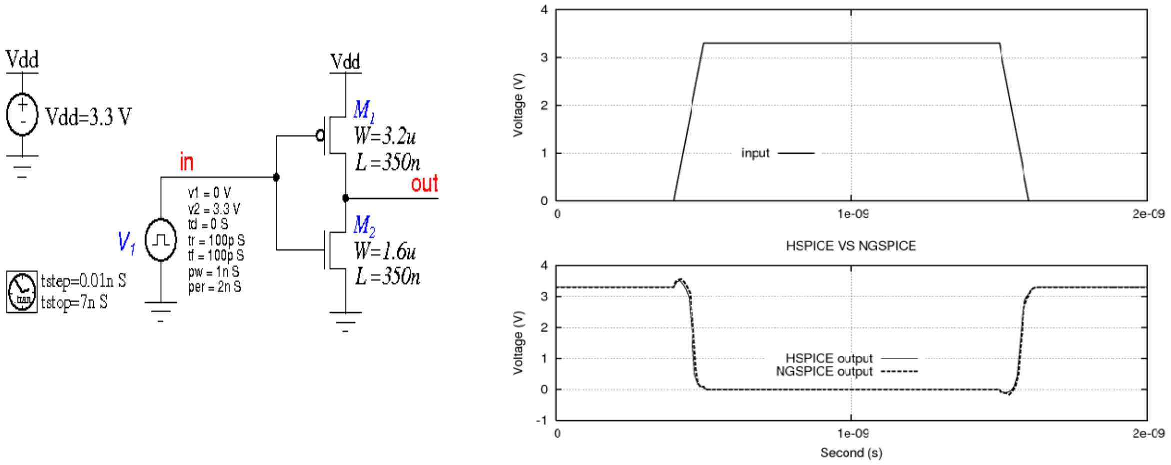
```
1     . include tsmc035 .model
```

in our netlist, and we are ready to use the SPICE model to run CMOS circuit simulations. In the following section, a more detailed example will be provided to show how this works.

6. Comparison of NGSPICE and HSPICE

Although NGSPICE is a free SPICE simulator, the reason we use it is not just price. Simulation accuracy is also a very important issue. In this section, we will compare NGSPICE with commercial HSPICE using different CMOS circuits. The SPICE model used in the comparisons was obtained from MOSIS website as mentioned in the prior section.

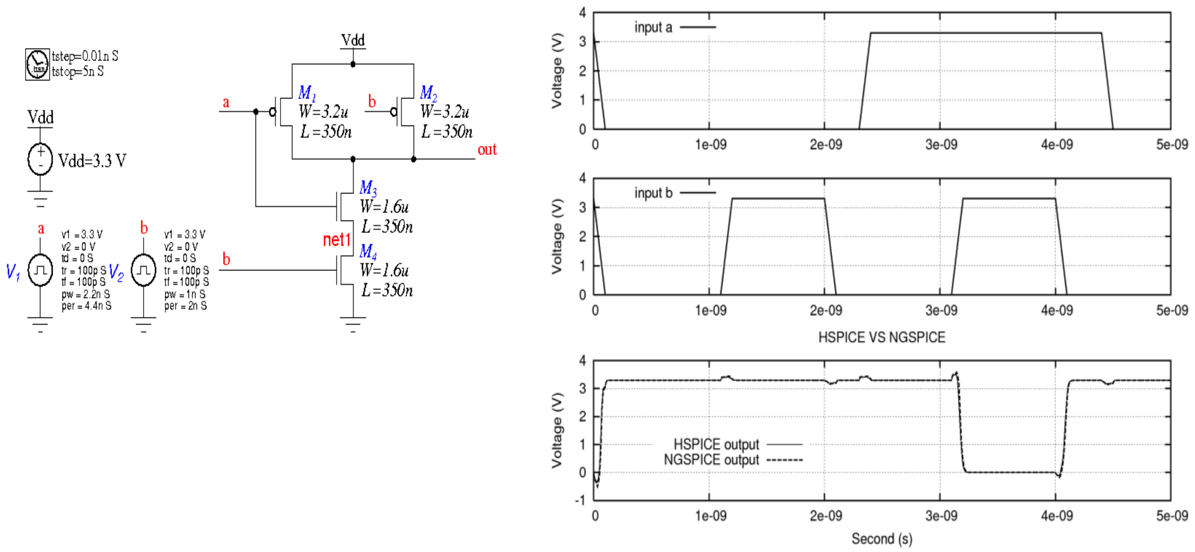
1) Inverter：Fig. 4(a) is the schematic of an inverter while Fig. 4(b) is the simulation comparison between NGSPICE and HSPICE. The solid line is the simulation result by HSPICE while the dotted line is the simulation result by NGSPICE. This figure shows that the simulation result is almost identical which means that NGSPICE can generate output very similar to HSPICE. However, one simple circuit simulation is not sufficient to persuade us to use NGSPICE. Hence, more simulation comparisons are performed to see if NGSPICE is capable to be an alternative to HSPICE for students to use.



(a) Inverter schematic (b) NGSPICE compare with HSPICE

Fig 4 Inverter simulation comparison between NGSPICE and HSPICE

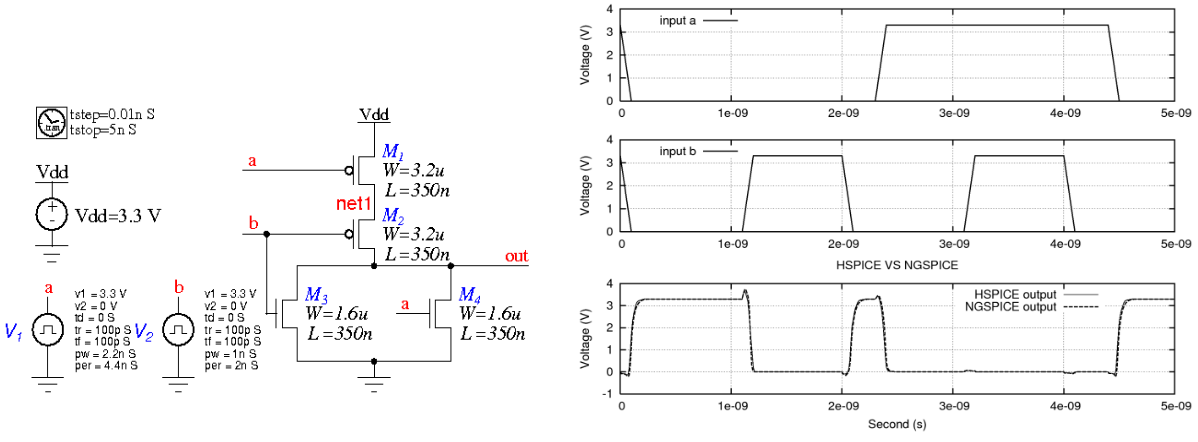
2) Two Input NAND Gate: The second circuit we compared is a two input NAND gate as shown in Fig. 5(a) while the simulation result is shown in Fig. 5(b). The simulation result between NGSPICE and HSPICE is almost identical.



(a) NAND gate schematic (b)NGSPICE compare with HSPICE

Fig 5 NAND gate simulation comparison between NGSPICE and HSPICE

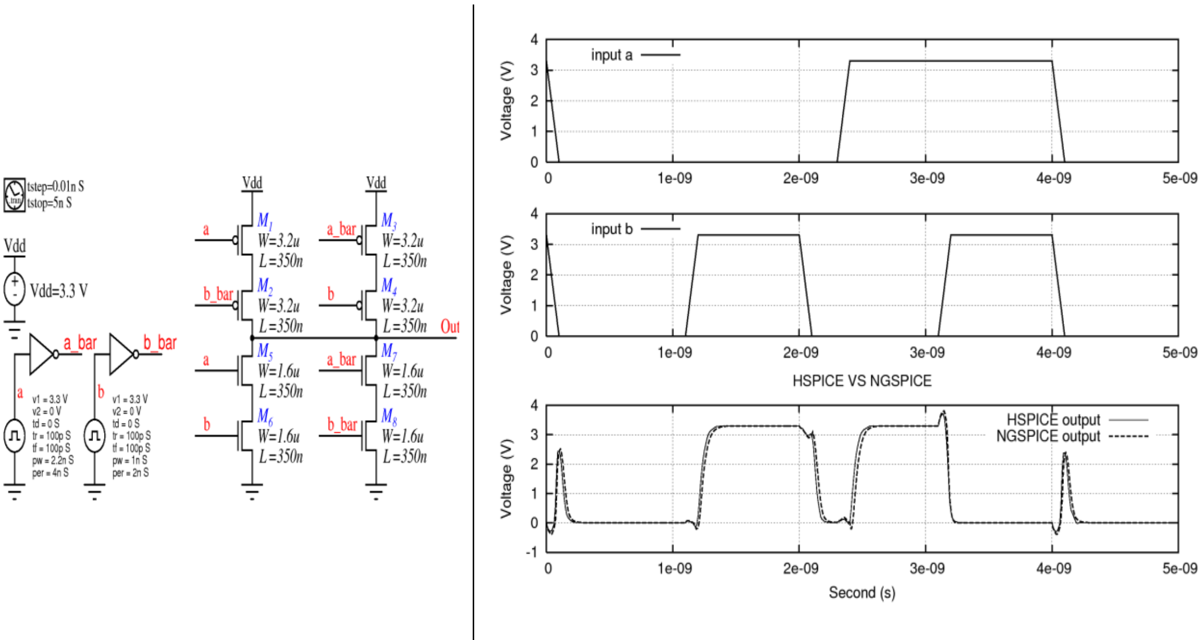
3) Two Input NOR Gate: The third circuit we compared is a 2 input NOR gate as shown in Fig. 6(a), it is quite obvious that the simulation comparison as shown in Fig. 6(b) is still very similiar.



(a) NOR gate schematic (b) NGSPICE compare with HSPICE

Fig 6 NOR gate simulation comparison between NGSPICE and HSPICE

4) XOR Gate: Last but not least, we present an XOR gate comparison in Fig. 7(a). As Fig. 7(b) demonstrates, both simulation results from HSPICE and NGSPICE are still very close.



(a) XOR gate schematic (b) NGSPICE compare with HSPICE

Fig 7 XOR gate simulation comparison between NGSPICE and HSPICE

It is not possible to run all circuit comparisons between NGSPICE and HSPICE since we do not have enough man power and time. However, we can tell from the above comparisons that NGSPICE does generate reliable simulation results.

7. Shortcoming of NGSPICE

According to our prior simulation comparisons, NGSPICE generates nearly identical outputs as HSPICE does. Therefore, a question may arise in our mind that is NGSPICE mature enough to replace HSPICE ? The answer to this question is NO. The prior comparisons were simulated using SPICE model from MOSIS. If a commercial CMOS process, say, TSMC 0.18 μm 1P6M CMOS process, SPICE model is used in NGSPICE. Many SPICE parameters will not be supported and hence will be omitted by NGSPICE. In this case, simulation results can be doubtful. Thus, although NGSPICE is a good, free SPICE simulator, it still can not replace HSPICE in the process of chip realization due to its limitations.

Another disadvantage of NGSPICE is it does not provide schematic capture function. However, this is not really a disadvantage after all since HSPICE does not provide schematic capture neither. The product position of NGSPICE or HSPICE is a SPICE simulator, schematic capture is a feature that may or may not be needed.

8. Summary

From the educational perspective of view, NGSPICE produces simulation results that are very near as HSPICE does. The price of NGSPICE is incredibly ZERO dollar. No license check is needed thus students can run NGSPICE on their own computers. Hence, it can be used as a free alternative to commercial SPICE tools for the entry level students to study SPICE. However, NGSPICE provides no schematic capture capabilities. To solve this problem, another tool called XCircuit [5] will be introduced in the following section.

IV. Xcircuit

1. The Demand of Schematic Capture

Schematic capture is a highly demanded feature in circuit design which is intuitive and increases the productivity of a circuit engineer. For example, if a voltage regulator is required, which way is preferred to design the circuit ? The easy way to create schematic capture like Fig. 8, or the hard way to write SPICE netlist like Listing 2 ? It goes without saying that the schematic capture way is much more preferred than the SPICE coding way. However, neither NGSPICE nor HSPICE provides schematic capture function. Although commercial EDA tools like Cadence composer and Springsoft ADP both provide schematic capture feature, free alternative is less mentioned. In this section, we will introduce a free alternative schematic capture tool: XCircuit.

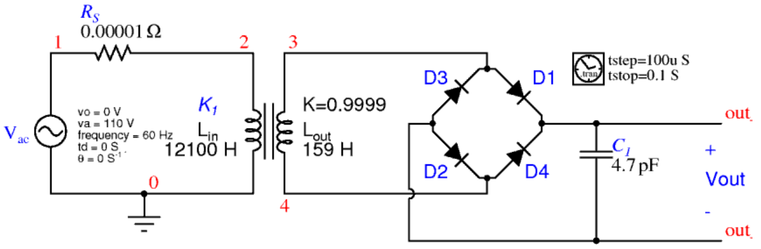


Fig 8 Schematic of a voltage regulator

Listing 2 Voltage regulator SPICE netlist

```
Vac   1 GND sin (0 110 60 0 0)
Lin   2 GND 12100 H
Lout 3 GND 159 H
K1 Lin Lout 0.9999
RS 1 2 0.00001

D1 3 out_p A_diode
D2 out_m GND A_diode
D3 out_m 3 A_diode
D4 GND out_p A_diode
C1 out_p out_m 4.7p

.model A_diode D (Is=0.01pA n=1.0675)

.tran 100us 0.1s
 .end
```

2. Introduction

XCircuit, in its literal meaning, is a circuit drawing software that runs on X-window platform [6]. However, XCircuit is capable to do more that just circuit drawing (or so called schematic capture). The advantages of XCircuit are listed as the follows.

- XCircuit uses postscript as its file saving format which results high quality vector-based images as shown in this paper.
- We can use XCircuit to create self-defined objects, e.g., power source, ground .. etc., as circuit building blocks. Fig. 9 demonstrates part of the self-defined objects we created.
- It is possible to ‘attach’ the corresponding SPICE codes to the self-defined objects. Hence, we can generate SPICE netlist directly from the schematic using XCircuit.
- XCircuit is free software, students are freely to use it without any restrictions.

As we can see from the prior section, some objects in Fig. 9 have already been used in Fig. 4 through Fig. 7. In addition, all schematics in this paper are created by XCircuit.

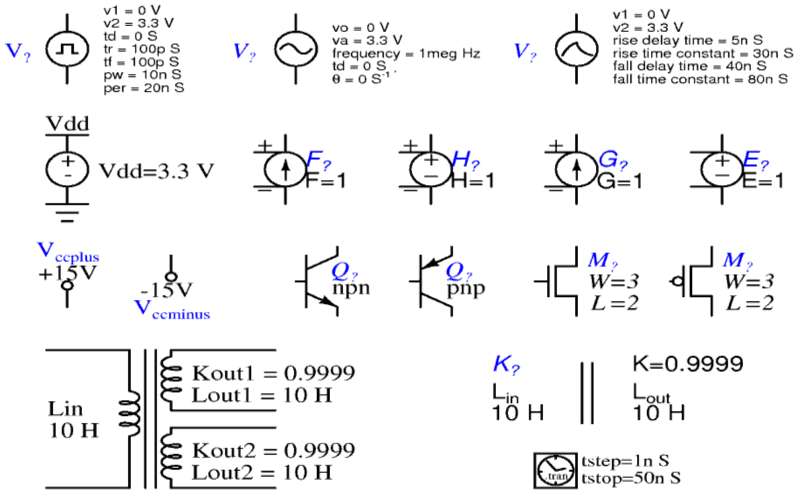


Fig 9 Self-defined objects in XCircuit

3. An Example Using XCircuit

Since we are able to create self-defined objects in XCircuit, it is possible to use the objects to create schematic. Besides, XCircuit can generate SPICE netlist directly from the schematic view. The SPICE netlist generated by XCircuit only requires little or no modification to simulate. Fig. 4(a) is an inverter created by XCircuit using self-defined objects. Listing 3 is the SPICE netlist generated by XCircuit. It should be noted that line 1 through line 7 are generated by XCircuit while line 8 is added afterward. This will notify NGSPICE to use the mentioned ‘tsmc035.model’ as the SPICE model.

Listing 3 XCircuit generated Inverter SPICE code

```
*SPICE circuit <inv> from XCircuit v3.6 rev 135
VVDD VDD 0 3.3
VGND GND 0 0
M2 out in GND GND cmosn W=1.6u L=350n
M1 out in Vdd Vdd cmosp W=3.2u L=350n
V1 in GND pulse (0 3.3 0 100p 100p 1n 2n)
.tran 0.01n 7n
.include tsmc035 .model
.end
```

V. Evaluation and Discussion

Before NGSPICE and XCircuit were introduced in the SPICE classes, advisors need to prepare commercial EDA tool environments for the entry level students. That means a lot of trivial and dirty works like creating user account, disk space arrangement, computer laboratory reservation for students to exercise after class .. etc. However, students still failed to deliver their home works on time due to various of software/hardware/computer laboratory problems. According to our experience, there were about 20% - 30% of students who claimed that they have suffered various of difficulties mentioned before to do their homeworks. Thus, the homeworks assigned were not delivered on time. However, after we use NGSPICE and XCircuit as our learning platform, only 5% or less of the students failed to deliver their homeworks on time. In short, everyone is benefited from the open source alternative as the follows,

- Firstly, the entry level students can study basic SPICE usage without having to complete computer resources with the senior level students.
- Secondly, the senior level students can use commercial EDA tools to run their graduate projects without having to worry about insufficient computers or licenses. It should also be noted that the senior level students can be benefited from the open source alternative, too. The senior level students can not access commercial EDA tools all the time since the computer laboratory is usually available during normal working hours. They can use the mentioned alternative on their own computers to test and verify their circuits. When they are able to use computer laboratory again, the tested designs can than be migrated to commercial EDA environments.
- Even the advisors can be benefited from this scenario. Once the entry level students use free EDA tools to learn circuit simulation, they do not need critical resources like software licenses or school computers to do their homeworks. Most of the computers today are sufficient enough to run all the simulations. Thus, the demand of computer laboratory is reduced, and the advisors are saved from the hell of trivial dirty works mentioned before.

Using open source EDA tools as the study platform creates a win-win situation. The reaction of students to this approach is very positive. Most of the students are able to run SPICE simulations anywhere and anytime. There are no more restrictions to use these EDA tools which result better learning results. Moreover, students are able to migrate from the free EDA tools to the commercial EDA tools flawlessly. In short, both advisors and students are satisfied with this approach.

VI. Conclusion

In this paper, we propose an open source approach regarding SPICE learning. The tools used in our approach are freely available and without any restrictions. The simulation comparisons also indicate that our approach provide reliable results. In addition, feedback from the students using the mentioned tools in our classes is very positive. In cloclusion, our approach can be used a better alternative in SPICE learning.

References

[1] Cadence. (2014). Cadence Design Systems, Inc.. Retrieved from www.cadence.com.

[2] Synopsys. (2014). Synopsys Inc. Retrived from: www.synopsys.com

[3] Springsoft (2014). Springsoft Inc. Retrived from: www.springsoft.com

[4] Mentor Graphics (2014). Mentor Graphics. Retrived from: www.mentor.com

[5] XCircuit (2014). XCircuit. Retrived from: opencircuitdesign.com/xcircuit/

[6] X.org Foundation (2014). X.org Foundation. Retrived from: www.x.org

[7] Ngspice Project (2014). Ngspice Project. Retrived from: ngspice.sourceforge.net

[8] SPICE (2014). SPICE website at Berkeley University. Retrived from: bwrc.eecs.berkeley.edu/classes/icbook/spice/

[9] gEDA (2014). GPL'd suite of Electronic Design Automation tools, Retrived from:geda.seul.org

[10] TSMC (2014). Taiwan Semiconductor Manufacturing Company Limited. Retrived from:www.tsmc.com

[11] IBM (2014). International Business Machines Corp. Retrived from: www.ibm.com

[12] The MOSIS Service (2014). The MOSIS Service. Retrived from: www.mosis.com

[13] MOSIS Test Data (2014). MOSIS Test Data. Retrived from: www.mosis.com/test