Send Orders for Reprints to reprints@benthamscience.ae

The Open Automation and Control Systems Journal, 2015, *7*, 1369-1376

Analog Circuit Simulation Approach with Multi-Fault Injector

Yuehai Wang and Hongling Zhao*

College of Electronic Information Engineering, North China University of Technology, Beijing 100144, P.R. China

Abstract: The process of simple fault injection into PSpice simulation environment has the disadvantage of low efficiency and low degree of automation. To address this problem, a multi-fault injector is designed and implemented. It can not only be capable of the single fault injection, but also have the function of the batch injection for the same fault type. Firstly, this paper explains the fault injector algorithm and the pre-processing of netlist file to avoid the problem that the single fault cannot be injected into multiple instantiated sub-circuits. Secondly, through analyzing the way to describe the integration component model in PSpice, the method adopts the sub-circuit way to create the fault model parameter file by using the substitution method and the equivalent circuit method without designing the fault model symbol library. Moreover, the fault list is closely associated with fault model according to fault injection algorithm. Finally, an experiment is designed to validate the practicability and validity of the proposed approach.

Keywords: Fault simulation, Multi-fault injector, Netlist flattening, Fault model, Fault list.

1. INTRODUCTION

In recent times, there has been a surge of usages of computer-aided test technology, especially in the circuit simulation, where the entry of fault analysis has brought about new efficiency challenges. By establishing fault model and injecting specific fault into the target circuit, the behavior of the circuit system can be observed under the faulty condition to perform fault detection and even to shorten the test cycle.

PSpice is a powerful general-purpose analog and mixed-mode circuit simulator which is used to verify and to predict the circuit behavior [1]. Recently, there are many circuit simulation application researches based on PSpice [2-4]. The circuit fault simulation method in PSpice environment combines circuit simulation with the fault injection by analyzing the fault simulation after the injection, to assess the testability verification, and to provide the basis for improving the system design. Manual-implemented fault injection of analog faults is the prevalent analog fault simulation methodology used in PSpice. As a consequence, comprehensive fault simulation of large mixed-signal circuits is almost impossible with today's tools. Moreover, manual fault injection into PSpice may damage the circuit diagram file and it is difficult to control the multi-injection accurately.

To avoid a series of complicated and tedious manual-implemented fault injection, the paper penetratingly researches the structure and the circuit description language of PSpice simulator and designs the multi-fault injector using object-oriented technique. The multi-fault injector can automatically extract the target information which the fault injection needs from the netlist file of simulation circuit.

*Address correspondence to this author at the College of Information Engineering, North China University of Technology, Beijing 100144, P.R. China; Tel: +86 10 88801652; Fax: +86 10 88801652;
Email: zhaohling@sina.cn

And it also can automatically flatten the netlist to overcome the problem that single fault cannot be injected into many instantiated sub-circuits. Via the traversal of the network table after flattening, it converts the fault-free network table into the fault netlist, and thus realizes the automation of fault injection process. In summary, the method achieves the automatic progress for injecting the specified faults to objective circuit.

2. DESIGN AND IMPLEMENTATION OF ANALOG CIRCUIT FAULT SIMULATION APPROACH

The fault simulation approach is designed to build a more general, a higher degree of automation, and a better testability verification system. Fig. (**1**) shows the framework of analog circuit fault simulation, which is composed of multi-fault injector, PSpice simulation engine, simulation controller, analyzer, component lib and fault library. Among them, the multi-fault injector is the core module of the approach, which injects automatically the fault model from the fault library into the target circuit in accordance with the user fault list.

It is mainly to parse the netlist file generated by PSpice and to analyze the source code of the corresponding data using c++ programming. More specifically, the multi-fault injector parses the circuit topology structure and extracts the component parameter by analyzing the netlist file. Then, the corresponding failure equivalent circuit models are incorporated into the circuit to substitute these fault-free device models. Combined with the other circuit models, these models are integrated into a new fault circuit network topology described by the fault factors. Finally, the injector loops through the entire netlist, until all the fault models are injected entirely. The detailed injected procedure is shown in section 2.1. All of the fault injection operation mentioned above is under supervisory of the simulation controller program.

1874-4443/15

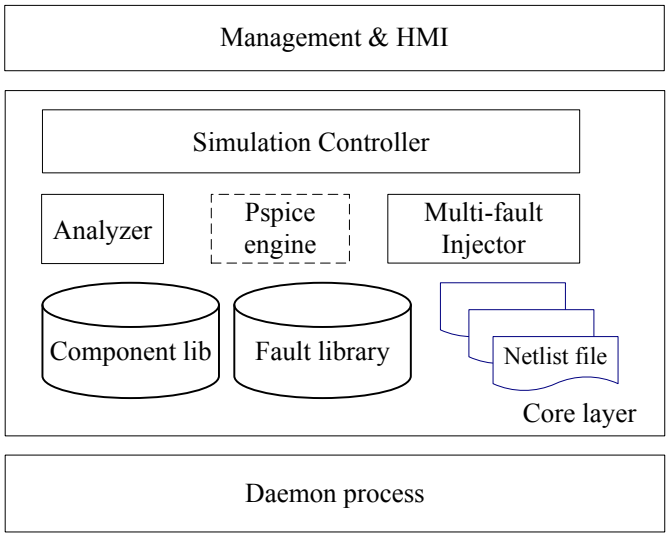


Fig. (1). Analog circuit fault simulation system framework.

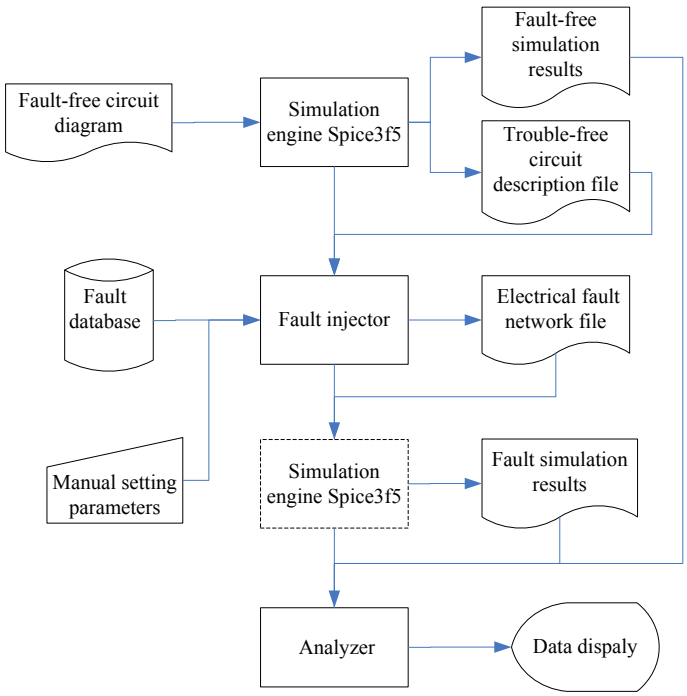


Fig. (2). Analog circuit fault injection process.

It provides interface program between users and fault simulation system to display information, to set fault parameters and to produce platform control commands. In addition, when the component fault model in the fault library can't satisfy the need of the user's simulation, the user can develop an unprecedented component failure type through the interface program provided by simulation controller, and even can add it to the fault library according to the specified path set by the user. The analyzer will compare the data before and after the fault injection to get the fault report on the basis of the fault model, verifying the effectiveness and correctness of this approach.

The fault simulation technique can implement the single fault injection of the fault model, batch injection of fault mode based on the similar devices, as well as parameter drift injection method based on the fault mode. As is shown in Fig. (**2**), the fault simulation process can be divided into four steps of data preparation, fault injection, fault simulation and result analysis. To begin with, the method starts from the bottom of the system structure to set their failure mode abiding strictly by each specific component failure. Meanwhile, PSpice engine can accept fault-free circuit netlist file to perform the process of circuit simulation, in which the engine also permits the input form of circuit diagrams files and

converts them into netlist files. Secondly, fault injection module, according to the circuit hierarchical decomposition tree, taking the user instruction for reference, combining with the component failure list and the fault mode library, performs the specified fault injection or the same fault mode batch injection based on the device type of failure factor into the specific components. The module modifies component model parameters or the corresponding circuit netlist file in order to obtain the specific fault circuit description file via human machine interface (HMI). Thirdly, the simulation controller automatically calls the PSpice simulation engine to simulate the netlist file output of fault injector to get fault simulation respond results. Finally by comparing the data before and after the fault injection, the analyzer will estimate weather or not this fault simulation succeeds. In general, preliminary results of the circuit fault simulation often are not consisted in the ideal goal. Generally, it needs perfecting mathematical model through the circuit analysis, and comparing the fault simulation results to improve the fault simulation model accuracy further.

Compared with the past fault simulation methods, the object of the injector operation is fault-free circuit netlist file rather than the fault-free circuit diagram file. Therefore the free-fault simulation results are still available after executing the fault-free simulation again.

2.1. Fault Injection

Fault injection technology plays a key role in analog circuit fault simulation. By injecting specific fault into the system, the behavior of the analog circuit fault can be observed [5]. Based on the PSpice system platform, there are three generic fault injection methods: modifying the circuit principle diagram, modifying the network topology file, and changing model definition. After comparing and analyzing all of the generic fault-injection methods, the conclusion can be reached that the essential of fault-injection methods is that the designed fault models replace fault-free device models to form a circuit network topology involving the described fault factors. In order to improve the automation degree of fault injection, almost all the fault injection this paper discusses employ the method to change model definition to conduct fault injection. In essence, this method is the process in which the injector searches the netlist, identifies and extracts component parameter information from circuit netlist file in order to modify and rebuild new table containing the fault model.

2.1.1. Preprocessing Netlist File

With extremely flexible writing of netlist file, PSpice circuit description language consists of attribute, parameter, model and annotation of each component in circuit, circuit simulation end mark, as well as command and operation, etc. And it intends to describe the title of circuit simulation, circuit topology hierarchical structure, the composition of circuit components, power supply and so on. Nevertheless, provided that the formal methods have not been applied to extract those information, it is extremely difficult to adapt to flexible network grammar phenomenon in the table.

In the process of simulation, PSpice can take advantage to capture software to finish mapping on a drawing page and carry on the simulation in terms of simple circuit. For large-scale complicated electronic equipment, in contrast, the whole circuit is impossible to accomplish on a piece of graph paper. Therefore, it adopts the modular and hierarchical design to decompose the whole circuit into several pieces to solve this problem. The hierarchical and modular design can keep the circuit connection and logical relationship between them.

However, if some sub-circuits contain the device model triggering fault, fault models have to be inserted into the sub-circuit when the fault injects. As a result, all instances of this sub-circuit cannot inject fault model, and thus it often can not reach the specified purpose of fault injection. So the intermediate data format into which the netlist formation is converted also needs flattening to eliminate the hierarchy.

In this paper, the multi-fault injector has designed the layer flattening module to decompose the netlist file to achieve the normal one exclusive of hierarchical and modular structure. After obtaining hierarchy, the injector can inject lower-level circuit description module into the upper level net table until all the networks don't contain a sub-circuit model. The modular and hierarchical flattening process is shown in Fig. (**3**).

2.1.2. Fault Injection Algorithm

Fault injection algorithm intends to connect with fault model library and fault list, so as to complete netlist file modification, and even to execute fault injection. Through the HMI, the user instructions about fault input model are expressed as a 3-tuple.

$$I = \{C, M, P\} \tag{1}$$

C is the normal model of faulty components, M is failure mode, and P is the parameters related to failure model, which can be used as the default values such as resistance in short-circuit. It also can be manually set through the human-computer interaction, which can be passed to fault library via the interface. For example, in the capacitance parameter drift , P is the value of capacitance parameter drift.

Fault model in the fault library can be expressed as a 4-tuple that is shown as follows:

$$B = \{M, C, C', P\} \tag{2}$$

C' is fault model of circuit component. C' and C can be the same or different. Such as, the fault resistance which can be found in Table 1 can be simulated with inductance model in short circuit. But the integrated chip applies sub-circuit to express its fault mode in the restructuring method. Therefore C and C' can be the same with the beginning of the 'X' letter.

The netlist file can be expressed by a 4-tuple that is shown as follows:

$$N = \{C, F, T, P'\} \tag{3}$$

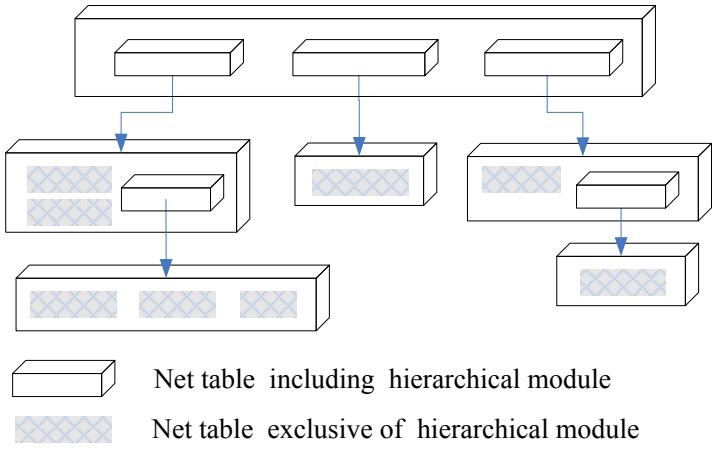


Fig. (3). Modular and hierarchical flattening process.

Table 1. The fault list of resistance.

Fault Model (M)	Model Name (C)	Fault Model Name (C')	Fault Parameter (P')
DC Short-circuit	R*	LR*	1
DC Open Circuit	R*	CR*	1
AC Short	R*	R*_ASHORT	X_AOPEN K
AC Open	R*	R*_AOPEN	X_AOPEN K
Parameter Drift	R*	R*_DRIFT	X_DRIFT K

F is the beginning net label of fault device pin in the netlist file, and T is the end. P' is normal device parameter, which is to be replaced.

Fault injection algorithm is shown as follows:

The function of *Fault-Proc* is to implement the injection of a certain fault type. The parameter N_{in}, is a vector of input normal netlist, i is user fault input, B is a vector of fault library, and N_{out} is a vector of output fault netlist. If faults injection involves many types, invoke it for many times, per type per conduct.

To manifest the above algorithm, it lists a small part of the user fault list in Table **1**.

2.1.3. Implementation of Multi-Fault Injector

The full injector system is made up of several models, including lexical analysis, parsing, semantic analysis, intermediate code generation, hierarchy flattening and netlist generation modules. In the course of the process, the lexical analysis, the parsing, the semantic analysis and the intermediate code generation are the core parts of the fault injector. They finish the analysis of the network table information, and are responsible for identification and treatment of the complex grammatical phenomenon based on PSpice engine.

In the process all the various commands, the devices in the netlist file are represented with c++ objects. Then they are organized in the abstract syntax tree to form intermediate code. Eventually intermediate code is unfolded to become flat table exclusive of a hierarchical structure, into which the response fault modes are injected according to the requirement. The implementation of multi-fault injection is shown in Fig. (**4**). Specific steps are described as follows.

2.1.3.1. Netlist Parsing

The fault injector reads the selective netlist file and conducts lexical analysis. During this process, the source code is decomposed into a series of symbols and is delivered into the symbol table. Then in parsing analysis phase those symbols are organized into the abstract syntax tree forming in order to check the grammatical mistakes and to extract useful information including netlist hierarchical structure from a network table. Once obtaining the syntax tree, some preparation can be made for the subsequent fault injection through traversing them. The semantic analysis is simple, just does some ancillary works for the intermediate code generation. Intermediate code generation module converts the data from the netlist file into c++ data structure for the purpose of the convenience of fault injection operation. Obviously the subsequent operation takes intermediate code as the research objects.

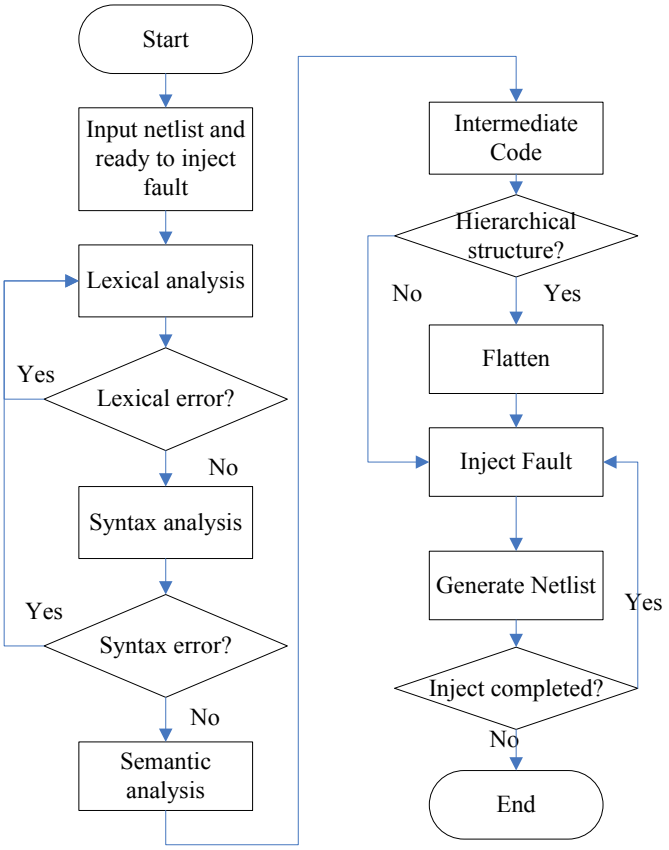


Fig. (4). Implementation of multi-fault injector.

2.1.3.2. Flattening and Injection

The intermediate code is unfolded into the flat table exclusive of hierarchical and modular sub-circuit. After them, the fault injection module adds the specified fault model to the objective intermediate code or modifies the original circuit, in conformity with the corresponding device fault requirement, so as to generate new intermediate code representing fault circuit netlist. Then the method employs the netlist generation module to transform the intermediate fault code into the fault netlist file in accord with PSpice circuit description language.

2.1.3.3. Judgment

Finally, it judges whether the netlist is completed to inject or not. If not, the above-mentioned operation will circulate until all of the user-defined fault device models are replaced. Otherwise the operation is ended.

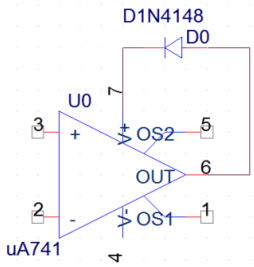
2.2. Fault Model

Because PSpice software does not have own proprietary fault model library, the fault simulation approach needs to conduct fault modeling consistent with PSpice standard by taking for reference multitudinous failure behaviors of diverse components during the simulation. The process of designing fault mode is closely related to circuit fault injection. In general, the design of fault model follows the two standards that the designed model should conform to the requirements of the simulation accuracy and should be able to be simulated in PSpice simulation environment.

2.2.1. Fault Model Representation

PSpice component model library consists of two parts. One is used to map circuit diagram, called the component symbol library, which stores mainly graphical information related to component model and the pin connection sequence, etc. The other is mainly used for the simulation equation establishment which stores the parameter of the component model. Since the method adopts fault injection by modifying model definition in the netlist file and does not change original circuit diagram, the fault model only needs to create the fault model parameter file, with does not need the fault model symbol library.

PSpice engine divides the whole components into 16 kinds of basic types, and sets rules for each of these components including their parameters, their formulas and equations available. Moreover, their model equations are embedded in the engine. Since the model is highly integrated with the simulator, it is very difficult to establish a new model type. Instead, those conditions make it possible to design fault modeling only through setting the new parameters of




```
*UA741_PS IS THE FAULT MODEL
* OF UA741, WHOSE OUTPUT
* MAINTAINS THE NEGATIVE
* POWER SUPPLY VOLTAGE
.SUBCKT   UA741_PS   1 2 3 4 5
X_UA741   1 2 3 4 5   UA741
D1    3 5   D1N4148
```

Fig. (5). The fault model of uA741.

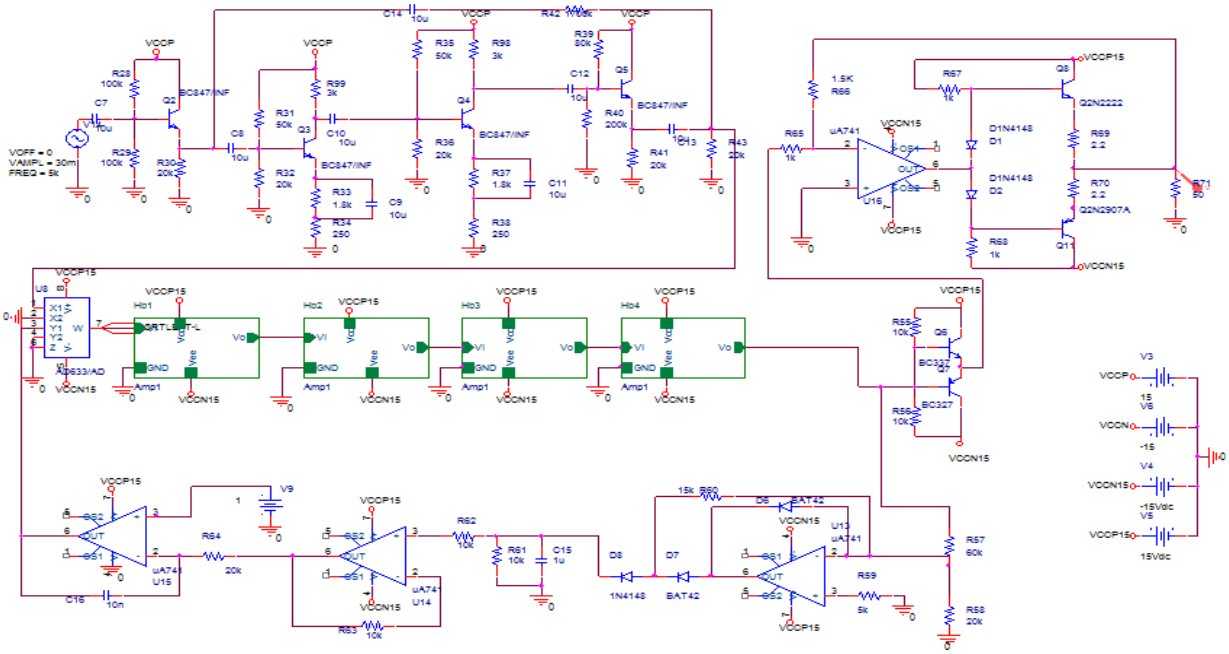


Fig. (6). Simulation circuit of the wave generator.

the existing model type or recombining several models using its rich component model library. Besides, complex components, such as amplifiers and oscillators, use the sub-circuit method to describe themselves in the model file, and the sub-circuit is composed of several of these basic components through the interconnection. Owing to applying the expression of macro model, the existing modularized component model can be mapped to fault model which PSpice engine can identify to build the module failure mode and to implement fault simulation further [6].

2.2.2. Fault Modeling Method

This paper adopts the substitution method and the equivalent circuit method to analyze the typical failure mode of the components and the way to trigger fault. Two-terminal elements normally contain resistors, capacitors, inductors and diodes. It adopts the alternative method to modify the relevant parameters directly in component simulation model. Impedance device realizes their fault modeling by changing the characteristic impedance values of the components. Similarly, the other two-terminal components can set the open-circuit impedance and short-circuit impedance respectively to complete open-circuit and short-circuit fault modeling. By modifying the characteristic parameters, the parameter drift fault modeling can be achieved.

In terms of the integrated circuit, since its internal structure is extremely complex and physical faults differ in thousands ways, the equivalent circuit method can be used. This method does not consider the inside structure of the components only except components failure pin. Hence, the method possesses a strong universality.

The integrated component can be connected in parallel, series, or any combination thereof with additional model to establish fault simulation model. Then, they are encapsulated into an integral whole. It stands to reason that the constituent model can be described by sub-circuit.

Take UA741 for example. It could be carried on the comprehensive test applying the above fault injection model. Especially, the output voltage varies near negative power supply voltage. Drawing on the volt-ampere characteristics of the diode phase combined with the characteristics of the integrated circuit itself, using the equivalent circuit method, the failure model can guarantee the stability of output to maintain the negative power supply voltage in Fig. (**5**).

3. THE FAULT SIMULATION INSTANCE

In order to validate the effectiveness of this fault simulation tool, an experiment circuit is designed in Fig. (**6**), which

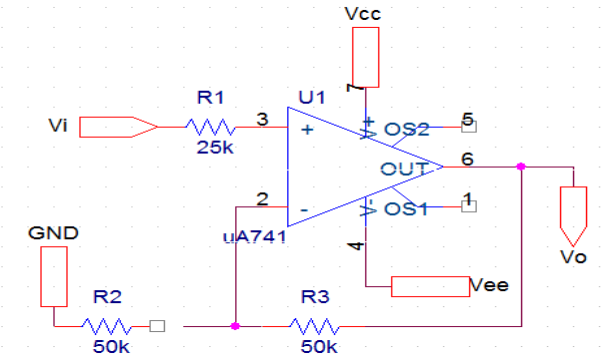


Fig. (7). Amp1 sub-circuit.

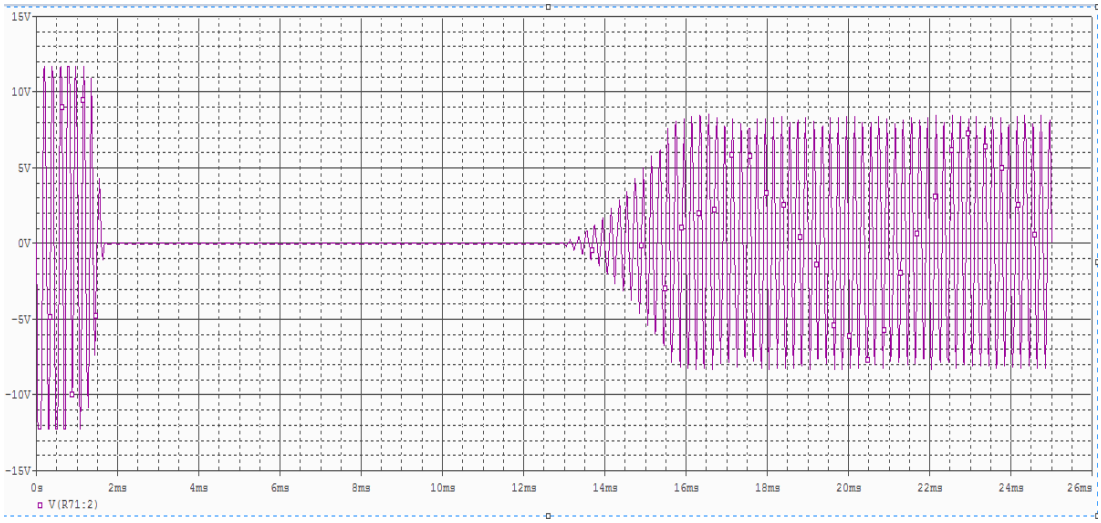


Fig. (8). Transient analysis results of normal circuit.

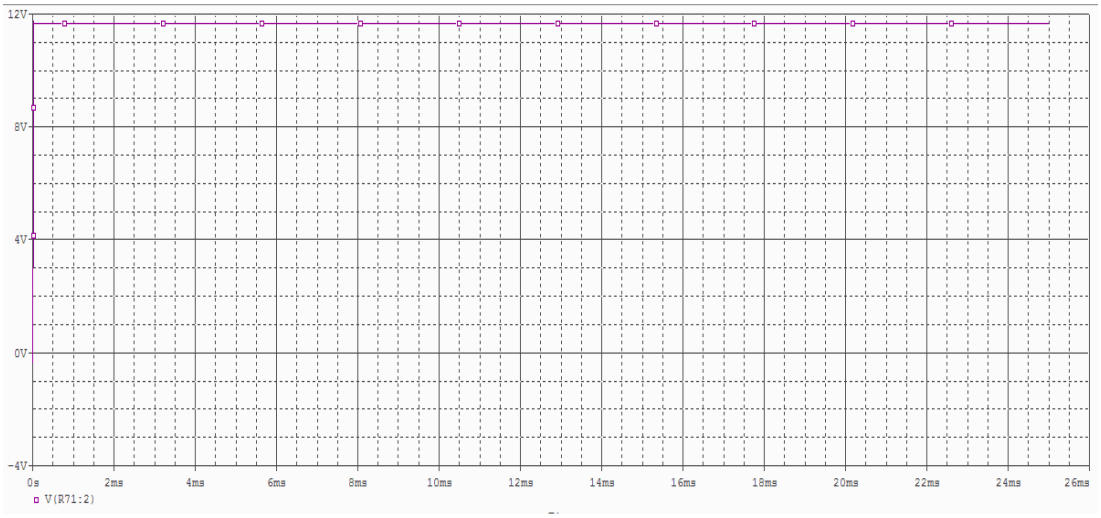


Fig. (9). Transient analysis results of fault circuit.

is like this: with applying multistage amplification to perform automatic gain, AD633 four-quadrant multiplier is used to adjust the magnitude of waveform. The simulation runs on transient analysis. The sub-circuit contained in the circuit is shown in Fig. (**7**). In the meanwhile, a set of parameters transient analysis are that start time is 0 ms, end time is 26ms, and step width is 10us.

The correct output of wave is as shown in Fig. (**8**). However, when all of the UA741 including sub-circuit chip trigger fault, the output is kept in the negative power supply voltage. The simulation waveform is demonstrated in Fig. (**9**). After a large number of experiments, the simulation waveform is in line with the actual data.

The above fault simulation test shows that the software can satisfy the requirement of most board-level circuit fault simulation, since the transient fault simulation needs a higher requirement for CPU and machine memory.

CONCLUSION

This paper proposes a simulation framework for analog circuit simulation by means of the multi-fault injection and calling spice3f5 simulation engine. The multi-fault injector is developed to perform the batch injection of the same fault. There are some advantages of the approach as follows:

1. To robotize the process of fault simulation acquisition.
2. To reduce the acquisition workload of human knowledge.
3. To push a more complete, comprehensive and practical analog circuit fault simulation.

In addition, our vision for the future of analog circuit fault simulation based on spice certainly includes more fault model libraries. And the multi-fault injector is hoped to have more comprehensive and powerful fault injection function.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflict of interest.

ACKNOWLEDGEMENTS

This work was supported by Cultivation Project for Young Talents of Beijing Municipal institutions (CIT&TCD 201504002). The authors are grateful to the editors and the anonymous reviewers of this paper for their insightful comments and practical suggestions on this work.

REFERENCES

[1] W. Zhao and P. Wei, "PSpice system simulation application in electronic circuit design," In: *32nd Chinese Control Conference (CCC)*, Xi'an, 2013, pp. 8634-8636.

[2] M.G. Seok, D.J. Park, G.R. Cho and T.G. Kim, "Framework for simulation of the Verilog/SPICE mixed model: Interoperation of Verilog and SPICE simulators using HLA/RTI for model reusability," In: *22nd IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC)*, Playa del Carmen, Mexico, 2014, pp.1-6.

[3] D. Cavaiuolo, M. Riccio, G. De Falco, and G. Romano, "An effective parameters calibration technique for PSpice IGBT models application," In: *International Symposium on Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM),* Ischia, Italy, 2014, pp. 133-138.

[4] C. Shetty, A. Kadle, and A.B. Raju, "A simplified approach to the first order approximations of a closed loop, non-isolated dc-dc converter with synchronous rectifier circuit behavior by using the ORCAD PSPICE," In: *5th International Conference on Advances in Recent Technologies in Communication and Computing (ARTCom),* Bangalore, 2013, pp. 309-318.

[5] R. Natella, D. Cotroneo, J. Duraes, and H. Madeira, "Representativeness analysis of injected software faults in complex software," In: *IEEE/IFIP International Conference on Dependable Systems and Networks (DSN),* Chicago, 2010, pp. 437-446.

[6] R. chacht, S. Rzepka, and B. Michel, "Parametric transient thermoelectrical PSPICE model for a power cable," In: *19th International Workshop on Thermal Investigations of ICs and Systems (THERMINIC),* Berlin, 2013, pp. 368-371.

Received: September 16, 2014 Revised: December 23, 2014 Accepted: December 31, 2014