IEEE Int. Symposium on Circuits and Systems (ISCAS'92); San Diego, CA, 10-13 May 1992, pp.81-84.
Copyright © 1992 IEEE (DOI 10.1109/ISCAS.1992.230009).

FLEXIBLE CIRCUIT SIMULATION WITH MIXED-DOMAIN AND MIXED-MODE APPLICATIONS

W.M. Zuberek

Department of Computer Science
Memorial University of Newfoundland
St. John's, Canada A1C–5S7

Abstract

An implementation of flexible circuit simulation through user-defined enhancements of "traditional" circuit analysis algorithms is presented. In particular, enhancements of the DC Transfer Curve analysis are used for implementation of mixed-domain simulation as required in analysis of self–heating effects of GaAs devices, while enhancements of time-domain analysis are used for simulation of mixed, analog-digital circuits at different levels of abstraction.

1. INTRODUCTION

Circuit simulation, or computer-aided circuit analysis, is one of well established and widely accepted CAD tools in the design of electronic circuits. Using circuit simulators, designers can easily determine the functionality and performance of their designs before the expensive and time-consuming fabrication takes place.

Traditional simulation programs are "closed" with respect to their functionality which means that they usually have one, fixed set of analyses, indicated by appropriate parameters in the circuit description. The execution of these analyses as well as specification of requested analyses, data and results must conform to established formats which are "built-into" the simulation programs. SPICE [Coh] is probably the most popular example in this class of circuit simulators. However, for more demanding applications, in which circuit simulators are integrated with other CAD tools, "closed" functionality becomes too inflexible, and an "open" simulation tool is needed which can easily be combined with other "open" tools. SPICE–PAC [Zub] is an "open" circuit simulation tool, which is upward compatible SPICE–like simulators, i.e., it accepts the same input language and performs the same circuit analyses as SPICE–like programs, but it also provides many new capabilities, that do not exist in traditional circuit simulation programs. Table–driven circuit elements and analyses, hierarchical naming scheme, dynamic (re)definition of parameters, or an accessing mechanism to internal representations of circuit elements are just a few examples of these new capabilities. SPICE-PAC also supports "enhanced circuit simulation" in which users can extend, modify or combine some "standard" simulation capabilities in order to increase efficiency, accuracy or applicability of circuit simulators. Hierarchical simulation, mixed–domain and mixed–mode simulation as well as new analysis modes and new properties of circuit elements can be implemented rather easily by such enhancements.

Two specific examples of enhanced circuit simulation are presented in greater detail in this paper. The first combines DC Transfer Curve analysis with temperature analysis to take into account self–heating effects of GaAs devices. The second enhances the traditional time–domain analysis of analog circuits with an event-driven simulation of digital devices in order to perform simulation of mixed analog–digital circuits.

2. ENHANCED SIMULATION

Flexibility of "standard" circuit analysis algorithms can be increased by introducing additional parameters of the analysis; selected values of these parameters control the execution of the "standard" algorithm introducing new features when they are needed. Since all these additional capabilities must be built–into the algorithm, the functionality of this approach is "closed" and cannot be changed without (further) modifications of the implementation.

Another approach increases flexibility through an "open" functionality of enhancements; the new, additional features are defined and supplied by users, usually as procedures which must conform to an established mechanism of communication. The only modification of the "standard" algorithm in this case is an implementation of interfaces to "external" enhancements, invoked when certain (predefined) conditions are satisfied.

Enhanced circuit analyses can conveniently be used for implementation of many improved or new simulation capabilities. For example, the DC Transfer Curve analysis [Coh] can be performed with variable step, controlled by an "external" process which increases the step for the flat parts of the transfer curve, and reduces it when more curvature is detected. The accuracy of analysis can easily be increased in such a way without increasing the computational effort. Circuit elements with new time–dependent characteristics can be implemented through enhanced definitions. New analyses can be implemented, and so on.

The concept of enhanced circuit simulation is sketched in Fig.1 in which a "standard" circuit simulator is enhanced by extensions of two analyses, ANALYSIS–1 and ANALYSIS–2.

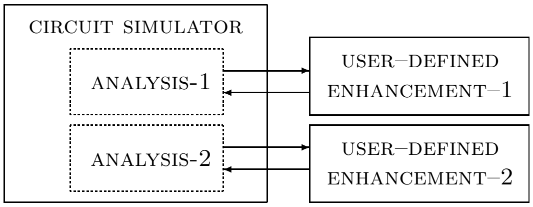

Fig.1. Outline of enhanced circuit simulation.

Interfaces to these enhancements are implemented in such a way that when a corresponding enhancement is not provided (by the user), the simulator performs the "standard" execution of the analysis; on the other hand, if the enhancement is supplied, it is invoked (with corresponding parameters) whenever the corresponding interface (within the analysis) is activated.

The interfaces for external enhancements are slightly different for different analyses because the sets of parameters which are made available for different enhancements are not the same; in all cases, however, they include the indicated components of the most recent solution of circuit equations as well as the information which determines the next step of analysis, and which can be modified by the enhancement.

Two specific applications of enhanced simulation are presented in the two following sections. Section 3 describes an implementation of mixed–domain analysis which combines DC Transfer Curve analysis with thermal analysis in order to take into account self–heating effects due to power dissipated in the transistor. Section 4 outlines an approach to analysis of mixed, analog–digital circuits in which the time–domain analysis of the analog part of the circuit is enhanced by an event–driven analysis of the digital part, and the interfacing mechanism between these two parts performs coordination of the two timing mechanisms, the variable–step time–stepping method with an event–driven time scheduler.

3. MIXED–DOMAIN SIMULATION

In SPICE–like programs, the (nonlinear) DC analysis is performed by the Newton–Raphson method which iterates the solution by solving consecutive linear approximations to the original system of nonlinear equations. DC Transfer Curve analysis [Coh] is basically a sequence of DC solutions in which a selected (independent) voltage or current source changes its value within some bounds.

The following outline of a modified implementation the DC Transfer Curve analysis uses parameters `DCstrt`, `DCstop`, `DCincr` and `SOURCE` to denote the initial value of the voltage (or current) sweep, the final value of the sweep, the increment and the independent source used for the sweep, respectively; the logical condition `enhanced` is TRUE only if the external enhancement is used:

```
initialize;
if enhanced then call enhancement(initial parameters);
value:=DCstrt;
while (DCincr>0 and value<DCstop) or
      (DCincr<0 and value>DCstop) do
   update(SOURCE,value);
repeat:
   solve_the_system_of_circuit_equations;
   if nonconvergent then
   {  if run_time control then
         call enhancement(nonconvergent);
      stop_analysis };
   store_results;
   if enhanced then
   {  call enhancement(parameters);
      if solution_unsatisfactory then
      {  delete_last_results;
         go to repeat } };
   value:=value+DCincr
endwhile;
```

In enhanced DC Transfer Curve analysis, an additional "external" analysis is performed after the DC solution for each sweep point is found; this external analysis can change some circuit parameters and then repeat the DC solution if the present solution does not satisfy some externally imposed conditions. The enhancement can be used for implementation of mixed–domain analysis in which self–heating effects of semiconductor devices are taken into account through (external) temperature iteration for each sweep point of the DC Transfer Curve. An outline of such an enhancement is as follows:

```
if initial_invocation then
   get_the_reference_temperature_from_the_circuit(Tref);
   Told:=Tref;
   get_reference_values_of_temperature_dependent_parame-
      ters_from_the_circuit };
find_temperature_increase_due_to_released_power(Tdelta);
Tact:=Tref+Tdelta;
if abs(Told-Tact) > accuracy then
{  update_the_ambient_temperature_of_the_circuit(Tact);
   find_values_of_temperature_dependent_parameters(Tref,
      Told,Tact);
   Told:=Tact;
   set_return_flag(solution_unsatisfactory) }
else
{  Told:=Tref;
   set_return_flag(continue) };
```

Fig.2 and Fig.3 show the DC I–V characteristics of the same heterojunction bipolar transistor without self–heating effects and with these effects, respectively.

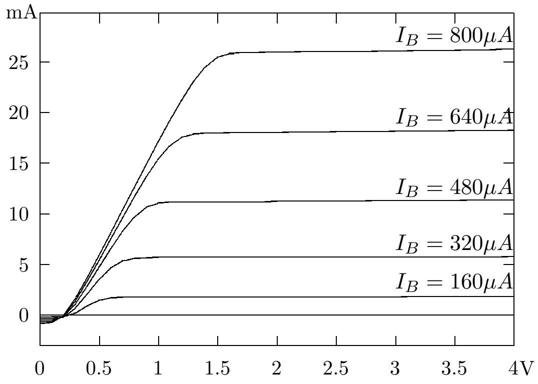

Fig.2. DC I–V characteristics without self–heating effects.

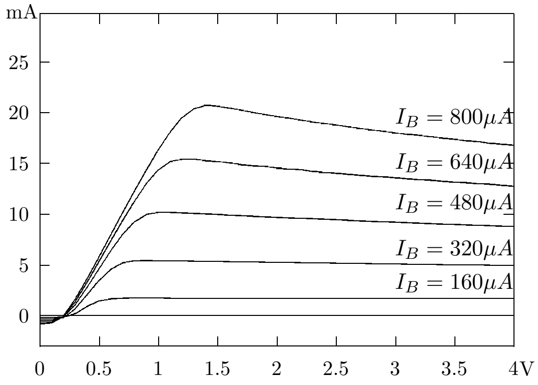

Fig.3. DC I–V characteristics with self–heating effects.

It can be observed that the characteristics are rather different, and that Fig.3 clearly exhibits the negative slopes of the I–V characteristics [GrOk].

The temperature increase due to the self-heating effects of a heterojunction bipolar transistor is shown in Fig.4.

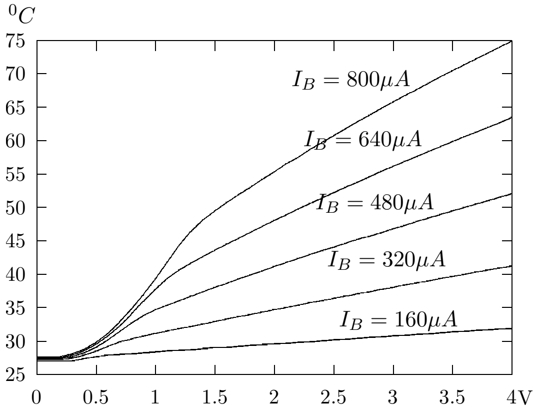

Fig.4. Temperature characteristics due to self–heating effects.

Implementation of mixed–domain analysis through enhanced simulation is reasonably efficient. The total numbers of iteration steps per DC Transfer Curve analysis without and with self–heating effects (and with temperature accuracy of 0.1 degrees Celsius), are as follows:

base current I_B	without self–heating	with self–heating
160 μA	62	113
320 μA	63	112
480 μA	62	154
640 μA	62	176
800 μA	70	215

The self–heating effects increase the total number of Newton–Raphson iteration steps on average from 2 to 3 times (for this accuracy), and this average value grows as the self–heating effects become more significant.

4. MIXED–MODE SIMULATION

Three fundamental approaches have been taken toward true mixed-mode simulation: the "coupled" (or "glued") approach uses separate analog and digital simulators and establishes some communication mechanism between them [CoWi]; the "augmented" approach extends either a digital simulator to handle behavioral analog models, or an analog simulator to handle digital models; finally, the "integrated" approach presumes that a tightly coupled pair of algorithms handles both analog and digital simulation [TMHS].

Mixed–mode simulation implemented through an enhancement of the time–domain analysis belongs to the integrated approach. General organization of such an enhancement is shown in Fig.5, where the digital simulation is performed either by user-supplied code (at any level or any combination of levels of abstraction), or by a typical digital simulator with a "flexible" time control mechanism.

Any integrated mixed-mode simulation must solve two basic questions: (i) conversion of continuous analog signals into discrete digital ones, and vice versa, and (ii) synchronization of time–control mechanisms used for analog and digital simulations. In enhanced time–domain simulation supported by SPICE–PAC, the interfaces between analog and digital parts of the circuit are specified by two new parameterized directives, PUTLIST (analog–to–digital) and GETLIST (digital –to–analog):

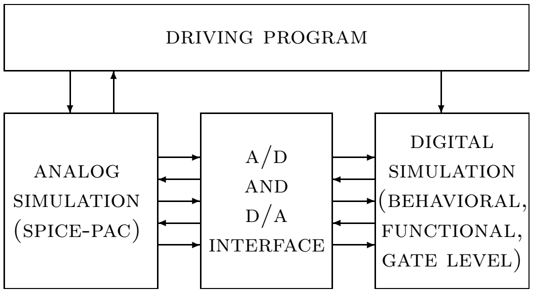
DRIVING PROGRAM
ANALOG SIMULATION (SPICE-PAC)
A/D AND D/A INTERFACE
DIGITAL SIMULATION (BEHAVIORAL, FUNCTIONAL, GATE LEVEL)

Fig.5. Organization of integrated mixed-mode simulation.

```
.PUTLIST:Tname1 Voutput1,Voutput2,...
.GETLIST:Tname1:Tname2 Vsource1,Vsource2,...
```

which use multi–threshold mappings of subranges of analog voltages into digital signals, and piecewise linear functions for continuous representation of discrete digital signals, with parameterized slopes and delay values.

The synchronization of (variable–step) analog and (event–driven) digital simulators is performed by invoking the analog–to–digital interfacing routines for each successfully solved timepoint [Coh]. The interfacing routines perform the analog–to–digital conversion of all PUTLIST signals, and then check:

- if any digital value created during this conversion differs from the corresponding (digital) value created in the previous invocation; and if there is at least one "new" digital value after the conversion, the timepoint is (iteratively) adjusted to a value corresponding to the closest conversion threshold, and then the digital simulation is is invoked to perform the simulation at the gate, functional or behavioral level for the actual timepoint;
- if all digital values remain the same, another test is made that checks if there is any (digital) event waiting for execution at this timepoint (such waiting events are created by internal delays in the digital part of the circuit); if there is such an event, it is executed by invocation of the digital simulation;
- finally, the next (analog) timestep is compared with the next (digital) event (if any), and if the event is scheduled to happen before the next timepoint, the timestep is adjusted to the timepoint corresponding to the waiting event; for internal digital delays, the analog simulation is thus "synchronized" with the digital simulation.

After completion of digital simulation, the digital-to-analog conversions are performed for those (digital) signals which are indicated in the GETLIST specifications and which changed their (digital) values during the simulation, and then the analog simulation resumes.

The coordination of analog and digital simulation can be described by the following scheme in which `Time` denotes the actual timepoint (or the simulated time) and `Timestep` is the next timestep:

```
while Time < simulation_interval do
   perform_one_analog_step_and_update(Timestep);
   check_the_signals_on_a/d_interfaces(change);
   if any_subrange_changed then
   {  adjust_the_time_to_the_nearest_threshold(Time);
      perform_one_step_of_digital_simulation;
      update_the_signals_on_d/a_interfaces }
   else
      if waiting_events then
      if next_event_time = Time then
      {  perform_one_step_of_digital_simulation;
         update_the_signals_on_d/a_interfaces }
      else
         if next_event_time < Time + Timestep then
            Timestep := next_event_time - Time;
   Time := Time + Timestep
endwhile;
```

Fig.6 and Fig.7 show the results of a very simple mixed analog–digital circuit in which characteristic "glitches" and "spikes" are generated by internal delays of digital gates:

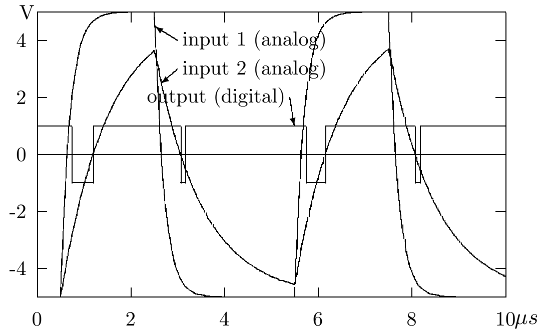

Fig.6. Analog input and digital output waveforms.

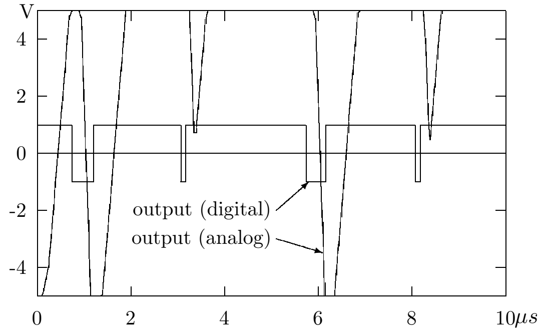

Fig.7. Analog and digital output waveforms.

The analog part mainly generates the "inputs" for the digital part, which is a pair of gates with nonzero rise and fall times [HoZu]:

```
INPUTS a,b;
OUTPUT y;
DEFINE nor = nor2 [rise=1E-7,fall=1E-7];
CIRCUIT y = or2(nor(a,b),b) END
```

where `INPUTS` match the `PUTLIST` analog signals and `OUTPUT` corresponds to the `GETLIST` analog signal.

In integrated mixed–mode simulation, the description of the digital part can be given at any level of abstraction or at any combination of such levels, and the digital simulation can be performed either by the user–supplied enhancements or by one of "traditional" digital simulators [AlZu].

5. CONCLUDING REMARKS

Two applications of flexible circuit simulation have been presented in this paper. One uses an enhancement of DC Transfer Curve analysis to perform mixed–domain, electrical–thermal analysis of self–heating effects in heterojunction bipolar transistors. The second application performs mixed–mode analysis of analog–digital circuits through an enhanced time–domain analysis. In the second case, the enhancement performs event–driven simulation of digital circuits at any abstraction level, or even any combination of abstraction levels.

It should be noted that many other applications can be implemented without any further modifications of the circuit simulation tool; as long as the required capabilities can be defined as enhancements, the existing interfaces to traditional analyses are sufficient for practical implementations.

Flexible circuit simulation is also a convenient mechanism for integration of software tools; for example, SPICE–PAC is one of the basic components of a new simulation–based parameter extraction program called FIT–3.

Acknowledgements

Collaboration with Dr. P.E. Allen of Georgia Institute of Technology, Atlanta, Georgia, and with Dr. A. Konczykowska, Dr. C. Algani and Dr. H. Wang, all of Centre National d'Etudes des Tèlècommunications, Laboratoire de Bagneux, Bagneux, France, is gratefully acknowledged.

The Natural Sciences and Engineering Research Council of Canada partially supported this research through Operating Grant A8222.

References

[AlZu] P.E. Allen, W.M. Zuberek, "Mixed–mode, analog–digital simulation using SPICE–like circuit analysis programs"; IEE Journal on Semicustom ICs, vol.8, no.3, pp.33–45, 1990.

[Coh] E. Cohen, "Program reference for SPICE 2"; Memorandum UCB/ERL M592, University of California, Berkeley, CA 94720, 1976.

[CoWi] T. Corman, M.U. Wimbrow, "Coupling a digital logic simulator and an analog circuit simulator"; VLSI Systems Design, vol.9, no.2, pp.38-47, 1988.

[GrOk] P.C. Grossman, A. Oki, "A large signal DC model for $GaAs/Ga_{1-x}Al_xAs$ heterojunction bipolar transistors"; Rec. IEEE Bipolar Circuits Technology Mtg., Minneapolis, MN, pp.258-261, 1989.

[HoZu] S.M. Hogan, W.M. Zuberek, "YADIS-1 : an introduction"; Technical Report #8905, Department of Computer Science, Memorial University of Newfoundland, St. John's, Canada A1C-5S7, 1989.

[TMHS] H. El Tahawy, G. Mazare, B. Hennion, P. Senn, "A new implementation technique for the simulation of mixed (digital-analog) VLSI circuits"; Proc. IEEE Int. Conf. on Computer-Aided Design (ICCAD'87), pp.396-399, 1987.

[Zub] W.M. Zuberek, "SPICE-PAC 2G6c - an overview"; Technical Report #8903, Department of Computer Science, Memorial University of Newfoundland, St. John's, Canada A1C-5S7, 1989.