A Josephson Integrated Circuit Simulator (JSIM) for Superconductive Electronics Application

Emerson S. Fang and Theodore Van Duzer

Department of Electrical Engineering and Computer Science
University of California at Berkeley
Berkeley, California 94720

Abstract

The Josephson SIMulator (JSIM) discussed in this paper uses a modified fixed-point method for solving nonlinear equations rather than the Newton-Raphson method commonly employed. Test runs show JSIM gives a several fold reduction in computation cost compared with JSPICE. Furthermore, preliminary analysis indicates the improvement factor should be larger for large circuits. In addition, the algorithm in JSIM can be easily adapted to run on parallel-processing computers, thus further enhancing simulation capability for complex Josephson circuits.

Introduction

Computer simulation is an integral part of any circuit design, including that of superconductive Josephson circuits. The Berkeley SPICE2 and IBM ASTAP programs have included Josephson models that have given them the capability to simulate Josephson circuits, but the intended purpose of these simulators is for semiconductor integrated circuits. They are not very well suited for Josephson circuits because semiconductor devices are very different from Josephson junctions. The most noticeable difference is the ac Josephson effect. A simulator is forced to take very small time steps to track the Josephson oscillation. Simulators like SPICE are not adapted for this particular kind of intensive numerical computation. As a result, only small Josephson circuits can be simulated in a reasonable amount of time. The impetus behind the development of JSIM (Josephson SIMulator) is to address this problem. In this paper, comparison will be made between JSIM and SPICE2G5 [1], which contains the Josephson model (we will refer to it as JSPICE).

The Basic Problem

The basic problem to be solved in most circuit simulators is finding the solution to a system of simultaneous ordinary differential equations:

$$dx/dt = F(x) \tag{1}$$

Since a digital computer cannot represent continuous time, the time interval has to be divided into a finite grid and all time derivatives have to be replaced by finite differences. In the above equation, solutions are found only for points on the grid. For the derivatives, the simplest method is to replace dx/dt by $(x_n - x_{n-1})/h_n$ where x_n is the value of x at time t_n and h_n is the nth time step. This is called the backward Euler method. It is stable, but has only first-order convergence rate. The most commonly used method is the trapezoidal integration rule, which has a second-order convergence rate. In this scheme, dx/dt is replaced by $\frac{2}{h_n}(x_n - x_{n-1}) - \left[\frac{dx}{dt}\right]_{n-1}$. This reduces the problem of solving a system of simultaneous ordinary differential equations to solving a system of simultaneous nonlinear equations given by

$$\frac{2}{h_n}(x_n - x_{n-1}) - \left[\frac{dx}{dt}\right]_{n-1} = F(x_n) \tag{2}$$

Still we cannot solve this directly, because no algorithm exists for direct solution of general nonlinear equations; an iterative method has to be used instead. The Newton-Raphson method is a well-known algorithm for solving f(x) = 0. The method requires an initial guess of the solution, and subsequent iteration is given by the Taylor series expansion of f(x) about the previous solution, truncated to the first two terms, $f(x^{k-1}) + f'(x_{k-1})(x^k - x^{k-1}) = 0$. In the case of system of equations, the derivative is replaced by a matrix of derivatives called the Jacobian. This iteration continues until $|x^k - x^{k-1}|$ is less than some predefined number which is the convergence tolerance. Now the problem has been reduced to solving systems of simultaneous linear equations of the form

$$\frac{2}{h_n}x^k{}_n - J(x^{k-1}{}_n)x^k{}_n = \frac{2}{h_n}x_{n-1} + \left[\frac{dx}{dt}\right]_{n-1} + F(x^{k-1}{}_n) \tag{3}$$

where $J(x^{k-1}{}_n)$ is the Jacobian of $F(x^{k-1}{}_n)$ evaluated at $x^{k-1}{}_n$ and $x^0{}_n$ is an initial guess of the solution x at time t_n. The Newton-Raphson method is second order and will converge to the solution if the initial guess $x^0{}_n$ is close enough to x_n.

Solving Ax = b

To obtain the solution of our basic problem, we need to be able to solve efficiently a large system of linear equations. The size of the matrix A is directly proportional to the complexity of the circuit. A circuit with N nodes will have at least N equations and the matrix A will be at least of dimension N by N. The Gauss-Jordan elimination method for solving a system of equations is a well-known and very efficient method. However, in its common form, it is not very well suited for digital computation; a slight modification is needed. This leads to the LU decomposition method [2]. L stands for lower triangular matrix and U stands for upper triangular matrix. Any nonsingular matrix can be decomposed to a product of two matrices where one is lower and the other upper triangular. The steps

This work was sponsered by AF Contract 19628-86-K-0033 with RADC funded by SDIO IST.

leading to this decomposition are identical to the steps in Gauss-Jordan method, except the matrix L is used to record the steps. Once A is decomposed to LU, the solution is easily obtained in two steps. First we solve Ly = b, and next we get the solution by solving Ux = y. Both steps are simple. The former involves only forward substitutions and the later only backward substitutions.

Josephson Junction in SPICE (JSPICE)

The Josephson junction model was added to SPICE, and it is currently the simulation tool at Berkeley and many other organizations for superconductive integrated circuit design. SPICE uses the modified nodal analysis method (MNA) to represent circuit equations [3]. The phase of a junction is treated as a separate voltage node to which no circuit element may be connected. In SPICE, nonlinear equations are solved by Newton-Raphson method, and time integration is typically done by trapezoidal rule. Given these conditions, the entries to the matrix (called MNA stamp) for the junction can be shown to be

$$\begin{bmatrix} N_+ & N_- & N_\phi \\ \frac{2C}{h_n}+\frac{1}{R} & -\frac{2C}{h_n}-\frac{1}{R} & I_c\cos\phi_n^0 \\ -\frac{2C}{h_n}-\frac{1}{R} & \frac{2C}{h_n}+\frac{1}{R} & -I_c\cos\phi_n^0 \\ -\frac{h_n}{2}\frac{2e}{\hbar} & \frac{h_n}{2}\frac{2e}{\hbar} & 1 \end{bmatrix} \begin{bmatrix} \text{RHS} \\ I_s \\ -I_s \\ \phi_{n-1}+\frac{h_n}{2}\frac{2e}{\hbar}v_{n-1} \end{bmatrix}$$

where $I_s = \frac{2C}{h_n}v_{n-1} + C\dot{v}_{n-1} - I_c\sin\phi_n^0 + I_c\phi_n^0\cos\phi_n^0$, and C, R, and I_c are junction capacitance, voltage-dependent resistance and critical current, respectively. Note h_n is the time step, not to be confused with $\hbar$ which is Plank's constant, and ϕ_n^0 is the guessed solution for the Newton-Raphson iterations.

Problems with JSPICE

Typical clock frequencies for Josephson circuits range from 100 MHz to multi-GHz. Even at the highest clock frequency, it takes many internal integration time steps to simulate a single clock period. This behavior can be explained using the Josephson equations. The value $\frac{2e}{h}$ is approximately 484×10^{12}, and a typical gap voltage is around 2.5 mV, which gives a Josephson oscillation frequency close to the tera-hertz range. Even circuits whose junctions do not switch to the voltage state (i.e., a one-junction SQUID), have transient behavior determined by the junctions' plasma frequency, $\frac{1}{2\pi\sqrt{L_JC}}$, where $L_J = \frac{\hbar}{2eI_c}$. A typical plasma frequency is about 200 GHz. This implies an integration time scale less than a picosecond, and is the reason that JSPICE recommends that the user set the maximum internal integration time step to about 0.2 ps. The time-step limitation is quite fundamental. Unless we are willing to suffer aliasing problems, we cannot take time steps larger than the Nyquist period.

Josephson Integrated Circuit Simulator (JSIM)

In light of the limitation discussed above, the effort in development of JSIM concentrated on saving computation effort at each time step, instead of trying to save on the number of time steps. The only nonlinear element in a Josephson circuit is the Josephson junction, and the junction's behavior is determined by the phase and by its derivative, which is proportional to the voltage drop. Therefore, if we can make a reasonable guess of the phase at the next time step and solve the remaining linear network, we would have a fairly good knowledge of the solution at the next time step. Given such a situation, we can use a fixed-point method to solve the nonlinear equations instead of the Newton-Raphson method, since the former is much simpler but requires a good initial guess.

The Newton-Raphson method is expensive because at each time step and each Newton iteration, an LU decomposition must be done in order to solve Ax=b. This is evident from equation (3). The Jacobian on the left-hand side needs to be updated during each iteration; therefore a new LU decomposition is required. To solve f(x) = 0, the fixed-point method rearranges the equation to x = x - f(x), and like the Newton-Raphson method, starts iteration with an initial guess x^0 and subsequent iterative solutions are given by $x^k = x^{k-1} - f(x^{k-1})$. In our modified fixed-point method, we put all nonlinear elements on the right hand side and the linear elements on the left. Equation (3) then becomes

$$\frac{2}{h_n}x^k{}_n - LNx^k{}_n = \frac{2}{h_n}x_{n-1} + \left[\frac{dx}{dt}\right]_{n-1} + NLN(x^{k-1}{}_n) \qquad (4)$$

where $LNx^k{}_n$ is the linear part of F and $NLN(x^{k-1}{}_n)$ is the nonlinear part. The matrix A does not change during fixed-point iteration. Therefore, a maximum of one LU decomposition is needed for each time step. Furthermore, if the current time step is the same as the previous step, the matrix A needs no updating; hence, no new LU decomposition is required.

Implementation of Josephson Junction in JSIM

As previously mentioned, JSIM uses the modified fixed-point method for solving the nonlinear equations involving Josephson junctions. Using the trapezoidal integration rule, we have the MNA stamp for a junction

$$\begin{bmatrix} N_+ & N_- & N_\phi \\ \frac{2C}{h_n}+\frac{1}{R} & -\frac{2C}{h_n}-\frac{1}{R} & 0 \\ -\frac{2C}{h_n}-\frac{1}{R} & \frac{2C}{h_n}+\frac{1}{R} & 0 \\ -\frac{h_n}{2}\frac{2e}{\hbar} & \frac{h_n}{2}\frac{2e}{\hbar} & 1 \end{bmatrix} \begin{bmatrix} \text{RHS} \\ I_s \\ -I_s \\ \phi_{n-1}+\frac{h_n}{2}\frac{2e}{\hbar}v_{n-1} \end{bmatrix}$$

where $I_s = -I_c\sin\phi_n^0 + \frac{2C}{h_n}v_{n-1} + C\dot{v}_{n-1}$.

The convergence of a fixed-point method depends even more strongly on the initial guess than does the Newton-Raphson method. It was decided that something better than a zeroth-order guess (i.e., the previous value) should be used for the initial guess of phase of each junction. Still, we want to spend as little computation effort as possible in making the guess. Since the trapezoidal rule gives both the voltage and the derivative of the voltage across the junction, a first-order extrapolation of the voltage across the junction at the next time step can be obtained with little effort, and the phase is easily computed by integrating the voltage to provide us a second order guess for the phase.

$$v_n^0 = v_{n-1} + \dot{v}_{n-1}h_n$$

$$\phi_n^0 = \phi_{n-1} + \frac{h_n}{2}\frac{2e}{\hbar}(v_{n-1} + v_n^0)$$

We found that the convergence tolerance requirement affects the computation of junctions' phases the most. As a consequence, the local truncation errors for node voltages and branch currents other than the phase node are always much smaller than the required tolerances, so an option is provided for the user to bypass the calculation of local truncation error. Because of oscillatory nature of Josephson junction dynamics,

the time-step control is accomplished by measuring the change in the phase. If $|\phi_n - \phi_{n-1}|$ is larger than the value of the global variable "max_phi_step", the step is rejected, and a smaller step is used instead. If on the other hand, the change is less than "max_phi_step", the step is accepted. Furthermore, if the change is less than 1/4 of "max_phi_step", the step size is doubled for the next time step. The number 1/4 is chosen to reduce the possibility of frequent changes of step size. This happens with any variable time-step integration method. After obtaining the solution with a small time step h, we find that a bigger time step, typically 2h, can be used for the next time point. However, at the conclusion of the new time point, we find the error is too close to the error tolerance and would be unwise to use 2h again, and revert back to the smaller time step h. This kind of time step variation will not result in large saving of time steps, and is avoided as much as possible in JSIM. This is important because the saving is not enough to offset the increase in number of LU decompositions in comparison to using a constant time step h for that particular time interval. Another situation to avoid is that of taking a big step, and either having the iteration to solve the nonlinear equations fail to converge or having the solution not satisfy error-control requirement, and then being required to start over with a smaller step. This is different from the previous situation, but has the same effect of an unnecessary increase of matrix computation. The ideal solution is not take a big step in the first place. We use a second-order guess for the initial phase across each junction for our fixed-point iteration in JSIM, so that if $|\phi_n^0 - \phi_{n-1}|$ is larger than "max_phi_step", the step is reduced, and a new guess is made before initiating nonlinear iteration. This reduces the possibility of doing a lot of matrix computation that is to be abandoned later. The success of such *a priori* step size control depends on the accuracy of the guess ϕ_n^0. This method is not possible if a zeroth-order guess for the phase is used.

Comparison with JSPICE

Several Josephson circuits are selected. The abbreviated descriptions of the circuits are listed in Table 1. The circuit schematic of simulation #10 is also shown in Fig. 1. We have done transient simulations using JSIM and JSPICE with the same relative error tolerance and user-set maximum time step (a simulator will not take a step larger than the user-specified maximum internal time step). The relative tolerance is 10^{-3} and the user set maximum time step is 0.2 ps, which is recommended by JSPICE. For JSIM, the measure for computation time is given by the fixed-point iteration count and the LU decomposition count. In the case of JSPICE, the relevant number is the Newton iteration count, which is also the LU decomposition count. The summary of results is given in the Tables 2 and 3. The output from JSPICE and JSIM for simulation #10 are shown in Figs. 2 and 3, respectively. The two outputs are identical, thus indicating the correctness of JSIM.

Table 1. Summary of the circuits selected.

Simulation Number	Circuit Description	Number of Junctions	Number of Linear Elements
1,2	sampler [4]	2	4
3,4	two-junction SQUID	2	10
5	single junction	1	2
6	one-junction SQUID	1	3
7,8	CLAM comparator [5]	3	11
9	VPP comparator [6]	5	20
10	SGA comparator [7]	10	48

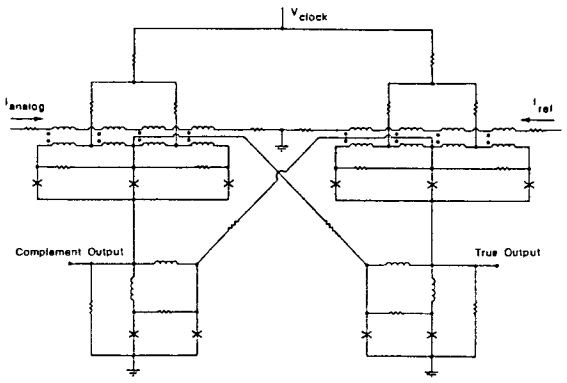

Fig. 1. The SGA (Self-Gating AND) comparator.

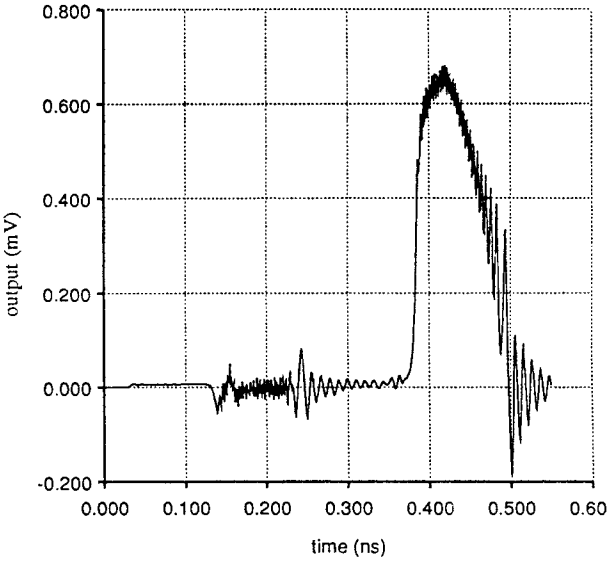

Fig. 2. Simulation result for the True Output of the SGA comparato using JSPICE.

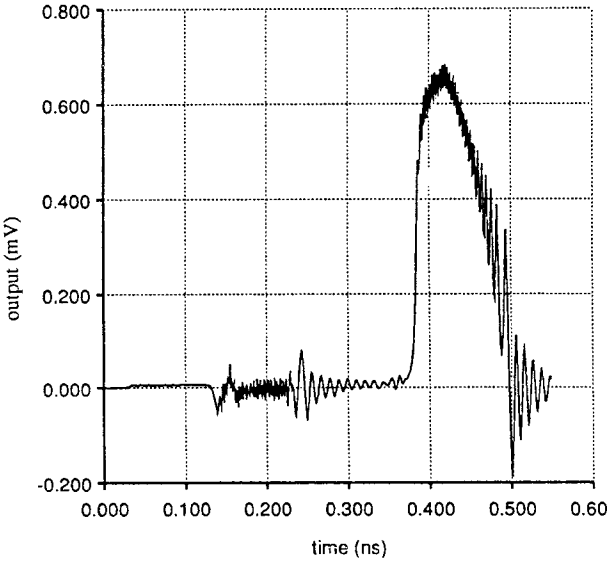

Fig. 3. Simulation result for the True Output of the SGA comparato using JSIM.

Table 2. Comparison of matrix computation count.

Simulation Number	JSIM LU count	Iteration Count		Time Steps	
		JSIM	JSPICE	JSIM	JSPICE
1	7	2292	4599	1232	1210
2	7	712	2379	528	526
3	3	2280	3363	1258	1144
4	1	535	1477	535	516
5	6	2184	3697	923	1254
6	27	1560	2290	853	637
7	3	2180	3205	1207	925
8	3	2275	4229	1256	1227
9	5	3073	5937	1584	1620
10	5	6305	12295	3998	3673

Table 3. Comparison of CPU time.

Simulation Number	JSIM CPU Time	JSPICE CPU Time	JSPICE Transient CPU	Speed Improvement Factor
1	2.2	10.8	10.1	5.0
2	0.9	5.5	4.8	6.1
3	3.1	12.7	11.9	4.1
4	1.0	5.3	4.6	5.3
5	1.1	5.8	5.3	5.2
6	1.1	4.1	3.4	3.7
7	3.6	12.2	11.4	3.4
8	4.0	17.8	17.0	4.4
9	7.8	36.7	36.2	4.7
10	40.0	236.8	235.4	5.9

Table 2 shows that the LU count in JSIM is much lower than in JSPICE and the iteration count is also significantly lower. We would expect the LU decomposition count in JSIM to be much less than that of JSPICE, but expect the iteration count to be higher, given the fact that the number of time steps taken by the two are about the same (has to be about the same, because they use the same integration method and the same error tolerance). This expectation comes from the knowledge that the fixed-point method has a first-order convergence rate while Newton-Raphson method is of second-order; hence, the former requires more iterations for convergence. From the simulation results, we see this is not true. This is due to the contributions of two factors. First, the initial guess for the phase of each junction is good (we used a second-order estimation), so that even though the fixed-point method has a lower order of convergence, the number of iterations required can be actually less than that of Newton-Raphson method. Second, the extrapolation done for the initial guess of junction phase gives us the ability to prevent making too large a time step that has to be abandoned after carring the iteration to convergence or exceeding the nonlinear iteration limit per time step.

The smaller iteration count cannot account for the speed increase; it is mainly the result of extremely low LU count. For a full matrix of dimension N, an LU decomposition would require a number of multiplications and divisions proportional to N^3, while for solving LUx = b, the computation count is only proportional to N^2. However, a node in a circuit has on average 3 elements attached and this results in matrix A being large and sparse; in such cases, the computation count for LU decomposition and solving LUx = b becomes proportional to N^α and N respectively, and $\alpha > 1$. For a typical sparse matrix, α can vary from 1.2 to 1.5. The saving in LU decomposition results a speed improvement of at least 2 for a small circuit, and the speed improvement factor grows as the size of the circuit increases. This is evident from simulations #7 to #10, where the speed increase grows from 3.4 to about 6. The circuit, for simulation #10 shown in Fig. 1, is still not a very large circuit, by most standards. We would expect better results for circuits that are more complex. More improvement can be expected on a parallel-processing computer. The LU decomposition is not very well adaptable to parallel processing, but the forward and backward substitution part is. Since the bulk of the computation cost in JSIM comes from that part, major improvement can be achieved using a parallel processor.

Conclusion

The objective of JSIM is to find a method that will improve the computation speed over JSPICE by a significant amount; the results from the simulation comparisons indicate that JSIM has met this goal.

JSIM is a stand-alone simulator. Currently only resistors, inductors, capacitors, mutual inductances, piecewise-linear voltage and current sources, dc dummy voltage sources (with values of 0 V) and Josephson junctions are supported. For future improvement, more devices such as transmission lines will be added, and the user interface will be refined to make it more user-friendly.

References

1. R. E. Jewett, "Josephson Junction in SPICE 2G5", Electronics Research Laboratory internal report, University of California, Berkeley, December 1982.
2. L. Chua and P. Lin, *Computer Aided Analysis of Electronic Circuits: Algorithm & Computational Techniques,* Prentice-Hall, Englewood Cliffs, New Jersey, 1975, pp.178-181.
3. Larry Nagel, "A Computer Program to Simulate Semiconductor Circuits", Memo No. ERL-M520, ERL, EECS Dept., University of California, Berkeley, May 1975.
4. D. A. Petersen, H. Ko and T. Van Duzer, "Dynamic Behavior of a Josephson Latching Comparator for Use in a High-Speed Analog-to-Digital Converter", *IEEE Trans. Mag.*, Vol. MAG-23, No. 2, March 1981, pp. 891-894.
5. C. A. Hamilton, "Josephson A/D Converter", *National Bureau of Standards,* Report Number SR-724-03-83, January 1983.
6. E. Fang, V. Nandakumar, D. A. Petersen and T. Van Duzer, "High Speed A/D Converters and Shift Registers", *Extended Abstracts of International Superconductivity Electronics Conference (ISEC),* Tokyo, Japan, August 1987, pp. 325-328.
7. D. A. Petersen, "Design and Testing of a Josephson Analog-to-Digital Converter", Masters Report, EECS Dept., University of California, Berkeley, August 1985.