International Symposium on Nonlinear Theory and its Applications
Xi'an, PRC, October 7–11, 2002

Analog Computation Programming in Switched-Current Circuits

Yasukazu Seiki, Kei Eguchi †, and Takahiro Inoue ††

†Kumamoto National College of Technology
2659-2 Suya, Nishigoshi, Kikuchi, Kumamoto, 861-1102 Japan
Phone:+81-96-242-6101, Fax:+81-96-242-6106
Email: seiki@cs.knct.ac.jp

††Kumamoto University
2-39-1 Kurokami, Kumamoto, Kumamoto, 860-8555 Japan

1. Introduction

Computations in analog circuits are classified into two types, i.e., continuous time mode and discrete time mode. For instance, we can solve differential equations by analog computers in continuous time mode. On the other hand, CMOS switched-current (SI) circuits, which are characterized by current memories and voltage controlled switching by clock pulses, are designed for discrete time mode.

In this paper, we will discuss some fundamental issues on analog computation circuits in current mode, especially the SI circuits concerning SI-memory and clocking schemes. By using SI-memories, Delay elements, and clocking schemes, some algorithms are implemented as SI circuits. How to translate conventional computer programs into circuit implementations is our concern.

Field Programmable Analog Arrays (FPAAs) are considered as another type of approach to circuit implementation for analog computations.

The circuit implementations in this paper are verified by SPICE simulations. In order to reduce the complexity of circuit model, various SPICE macro-models are introduced by using current dependent current sources and voltage dependent current sources, etc.

2. Switched Current Memory

In this paper, we restrict the topic of programming to the mathematical problems, especially the algorithms to obtain the solutions of equations with unknown variables. In general, the digital computer programming is executed step by step according to the order of the description. However, the circuit implementaion of the algorithm sometimes enables us to obtain the solutions without executing any procedure in the programming. A good example is an analog computer, by which the solutions are instantly obtained when the circuit implementation of equations is given.

On the contrary to analog computers, the SI circuit follows the time sequence generated by clock schemes controlling a group of switches in this circuit. Therefore, the speed of computation on SI circuits is dependent on the frequency of clock pulses. In spite of this weakness, SI circuits are considered as more program-oriented than analog computers.

In practical system control problems, some mathematical equations or relations concerning known variables and unknown variables are required for a representation of the system and a model for controlling and evaluation. But, the methods for obtaining solutions, especially the analytical solutions, are not always easy or possible. The distinction between the representation of the theoretical solution and its practial method leads us to the various approaches for obtaining solutions, e.g., the approximated numerical calculations, equivalent circuits simulations, etc.

Now, let's compare the style of the digital programming and that of the analog computation. Apparently, it is not possible to treat the analog values in digital computer programming, so these values are converted into digital data and processed in digital computers. The results are again converted into analog data and applied as the control parameters of the systems. However, it is evident that the data conversion processes cause the loss of information and delay of execution time.

Of course, the direct calculations of analog data, such as addition, subtraction, division, mutliplication, and other mathematical operations, are possible in analog circuits. This report proposes the programming style for the analog computation in order to process analog signals or data directly and in real time. Analog circuits are classified into two types, i.e., continuous time systems and discrete time systems. The SI circuit is a

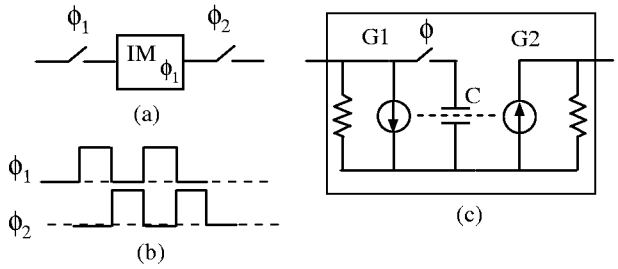

Figure 1: SI Memory Cell.

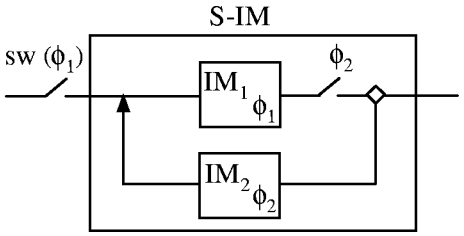

Figure 2: Stationary SI Memory Cell.

representative example of discrete time systems, and the behavior is characterized by a set of switches which are controlled by the scheme of clock cycles.

The substitution is one of the common operations in various types of programming. However, for analog circuits, especially in continuous time mode, the technique of memorization has not well established because of the difficulty of obtaining memory device for analog data. On the other hand, switched current circuits, which behave in discrete time mode, consist of analog current memories and voltage-controlled switches. In Figure 1, (a) the basic structure of switched-current memory cell, (b) its clock scheme with two alternative pulses $\phi 1$ and $\phi 2$, and (c) the SPICE macro model are shown. The SI memory has two phases, memorization phase and restitution phase [1]. The SI memory stores the current value during the half clock of unit cycle and restitutes it during the remaining half clock.

SI memories can hold data for unit clock cycle. Therefore, SI memory is used mainly as an instant storage for intermediate computations.

It is often required to store the analog data until a task terminates, and to fetch these data for the next task. The stationary memory (S-IM) is designed for this purpose, and it is easily implemented by two SI memory cells as shown in Figure 2.

A function with multiple variables can be caluculated only when all the variables are determined. If the calculation of the function for each value of the variables is executed following the time schedule, it is necessary to hold the data for each variables and wait until all the results are obtained. In this case, the stationary mermory will play an important role to synchronize the

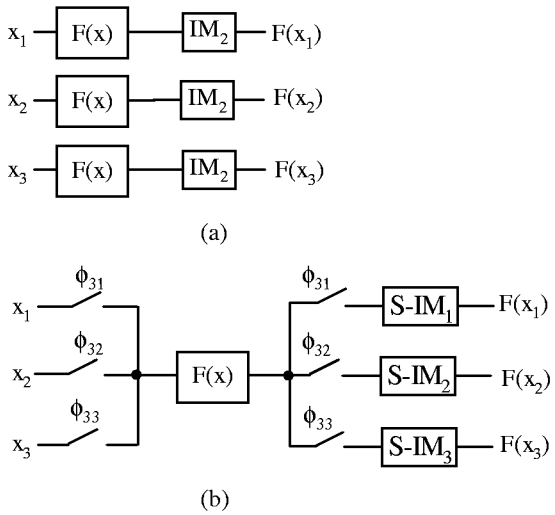

Figure 3: Sequentially executed Function Circuits.

processes with various time delays. An application of S-IM is shown in the following section.

3. Parallel Circuits and Sequential Execution

For sequential programming, the simultaneous execution of distinct operations is impossible even if these operations are independently executable. However, it is easy to implement simultaneous computation on analog circuits. Besides, sequential execution of parallel algorithm is also possible in SI circuits. For instance, a group of procedures using the identical function are represented sequentially in programming. This part of algorithm is rearranged in SI circuits as an array of the circuits associated with the function.

Apparently, it seems efficient for caluculation to prepare the parallel connection using the silimar function circuits [2]. However, if the requirement for speed of computation is not so strict, then it is desirable to use the same function circuit repeatedly for the sake of reducing the cost and complexity of the circuit. Furthermore, an iterative use of the identical circuit facilitates the uniform setting of parameters for computing, and its circuit realization is easily given by SI circuits, as shown in Figure 3. In this figure, the result for each parameter is stored in SI-memory. If the stored memory is applied to other calculations, it is necessary to prepare the stationary memories for adjustment of time delays.

In translating algorithms into SI circuits, it is necessary to determine the clock scheme according to the number of steps for calculations. Figure 4 shows an example of clock scheme to assign the value of function $F(x)$ for each variable $x_i (i = 1, 2, 3)$. In this case, three cycles are used to complete the task while the control clock ϕ_4 is on.

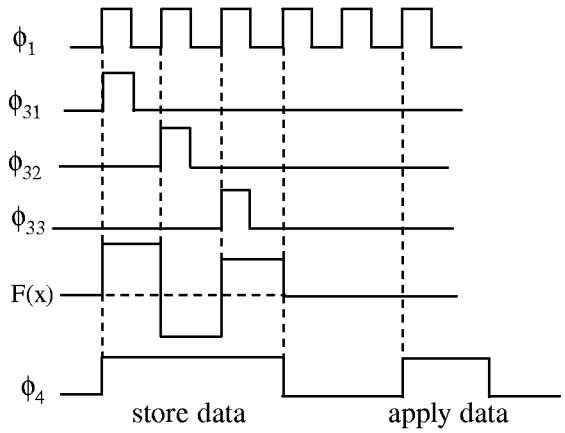

Figure 4: Clock Cycles for Execution Algorithms.

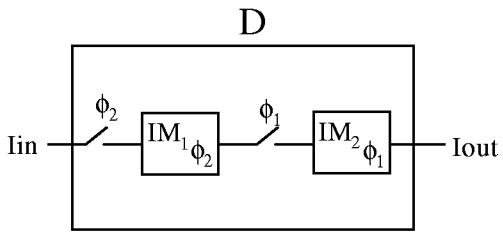

Figure 5: Full-clock delay

4. Circuit Implementation of Algorithm

Many algorithms include iterations of procedures, updating data, and logical judgements. In the case of iteration or updating data, the form $x_{n+1} := update(x_n)$ is applied as an implicit rule for the description $x := update(x)$. In order to implement such an algorithm, delay elements are used to distinguish the present data and the following data explicitely. The delay element can be easily implemented by two SI memories as shown in Figure 5.

An example of circuit implementation for the following algorithm is shown in Figur 6. This example is well known as the bisectional method for obtaining the approximated solutions of mathematical equations $F(x) = 0$. The algorithm is simple and can be easily programmed.

$a := -1;\ b := 2;\ c := (a+b)/2;\ \{initial\ values\}$
while condition is satisfied
if $F(a) * F(c) < 0$ *then*
begin $a := a;\ b := c$ *end*
else
begin $a := c;\ b := b$ *end;*
$condition_{new} :=$ *altered condition*
end $\{of\ while\}$

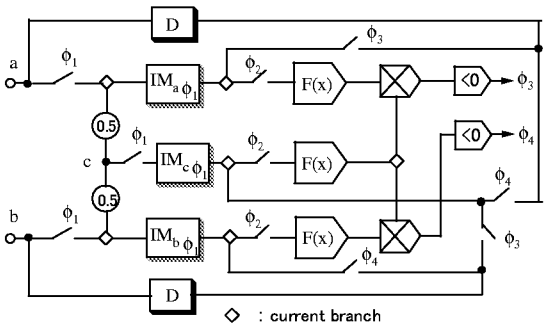

Figure 6: Circuit Implementation of Bisectional Method

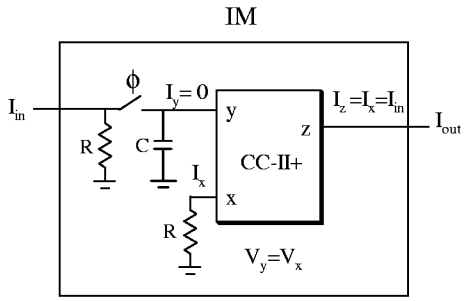

Figure 7: SImemory by CC-II+.

In this algorithm, $a := a;\ b := c$ means $a_{next} := a;\ b_{next} := c$. Of course, a description like $a := a$ is useless in any programming language, and this procedure is executed implicitly in iterations. However, in circuit implementation, the procedure is indispensable for updating the values explicitly in repeated procedures, as mentioned above. In Figure 6, repeated updating of a and b is realized by two delay elements. The selection of updated data is controlled by ϕ_3, ϕ_4 and the device with symbol (< 0) which outputs a positive voltage when the input current is negative.

In this example, for a simlicity of circuit representation, three circuits are prepared for the function $F(x)$. If these circuits are rather complex, the sequential execution method (as discussed in Section 3), should be applied during at least three clock cycles. Similarly to the digital programming, it is necessary to determine the condition when to stop the calculation according to the required accuracy of data.

5. Current Conveyors and SI-memories

The current conveyor (CC) is a current mode device and is utilized as a versatile device by which we can design an integrator, a current amplifier, and a current inverter, etc. One of the features of CCs is that the current I_y, which flows into the terminal y, is zero, and the voltage at the terminal y is equal to the volatage

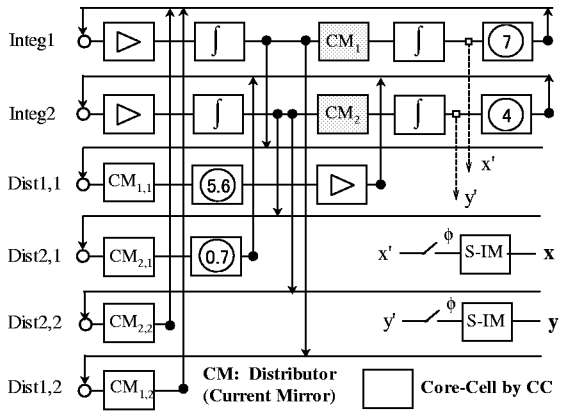

Figure 8: S-IM in FPAA.

at the terminal x. From these properties, it is easily seen that the capacitance inserted between terminal y and the ground holds the voltage across these terminals. As a result, we can obatain another SI memory using a CC as shown in Figure 7. Apparently, this circuit forms an integrator. Therefore, the circuit containing CCs can work as analog memory in switched-current mode.

Another approach for analog computations is to replace the CC integrators by SI integrators or SI differentiators. Differentiators in continuous time mode have no good performance. However, SI differentiators are known to have a good quality. From these facts, it is considered that the possibility of new circuit implementaion exists in combination of CCs and SI circuits.

6. SI circuits on FPAA

In the previous report [3], an analog computer circuit for differential equations (1) and (2) has been designed on FPAA with core circuits using CCs as shown in Figure 8.

$$\frac{d^2x}{dt^2} + \frac{dx}{dt} + 7x + \frac{dy}{dt} = 0 \qquad (1)$$

$$\frac{d^2y}{dt^2} + 0.7\frac{dy}{dt} + 4y - 5.6\frac{dx}{dt} = 0 \qquad (2)$$

In this figure, core cells are used as SI-memories to store the solutions of differential equations, although the execution speed is restricted by the clock pulse frequency. The programmability of FPAA must be considered from the following point of views: design level and working level. Many of conventional FPAAs are given their circuit functions by loading the design information stored in shift registers. This step is considered a design level, not a working level. Hence, it is difficult to alter the circuit structure while the system is working.

Here, we propose an approach to dynamic changeable circuits by introducing SI mode FPAAs. For this purpose, connections between core cells are to be replaced by switched-current circuits. Then, under the programmable clocking schems, dynamic processes of memorization and restitution in homogeneous array of circuits are considered to produce a new type of analog computations.

If the values of parameters are changeable by program, and these values are stored at any phase of clock, then a state of computation circuit can be changed to another state by clocking schemes. Assigning SI-memories on FPAAs and controlling state transitions of computations are new problems on the design and application of FPAAs in SI mode. The concept of the programmability concerning functions of cores and inter-connections within FPAAs should be reconstructed following analog computations on FPAAs together with SI memorization/restitution processes and path switching.

7. Conclusions

A Programming style for analog computation is proposed by SI technique. The roles of SI memories and the clock schemes are discussed on the circuit implementation of algorithms with some examples.

Analog computations on FPAAs with SI techniques are discussed on the possibilities for dynamic programmable circuits in SI mode.

More detail discussions on the above topics, especially on the following, are the future problems.

Design of analog computation circuits for Non-linear equations. Many practical system control problems require the real-time solutions for non-linear equations. In order to design the systems with rapid performance at a low cost, the circuit implementations without using AD/DA converters and micro-processors are desirable.

References

[1] M.Renovell,etc.,"Design-For-Testability for Switched-Current Circuits," Proc.VLSI Test Symposium, pp.370-375, 1998

[2] Hausner,A,"Analog and Analog/Hybrid Computer Programming," pp.594-595, Prentice-Hall, 1971

[3] Y.Seiki, K.Eguchi, S.Suzuki, T.Inoue, "An Analog Computation Circuit Using Current Conveyors and Its Implementation on Iterative Arrays," Proceedings of ITC-CSCC'2001 (Tokushima, Japan) Vol.I, pp.386-389, 2001