Implementation of Chaotic State Machine using Deterministic Chaos Function

Kwang-Hyeon Park, Jong-Sun Hwang, and Chong-Eun Chung

Abstract

For practical application of the concept of chaos, we propose a chaotic state machine as a sequential system. Chaotic state machine which is suggested and implemented in this paper has chaotic motions relying on the dynamics only through the deterministic chaos function. Also, we present and verify that the properties of chaotic state machine is equal to the characteristics of chaos.

I. Introduction

In recent years nonlinear systems often arise in engineering applications. Some nonlinear dynamic systems generate seemingly random, but actually deterministic, processes. Such a process is called deterministic chaos. As is well known, the dynamics of chaos has made very considerable progress in the understanding of nonlinear phenomena. The concept of chaotic behavior now pervades virtually all the sciences and technologies.

In order to apply the concept of chaotic motion to the electronics, an application of chaotic dynamics in the digital systems[1] - specially, on a sequential system - is presented in this paper.

The paper is organized as follows:

In section II, we first introduce Mealy and Moore machines as representative sequential systems, and then we also present chaotic state machine which has chaotic characteristics for comparing two previous machines and the chaotic state machine. To explain the essential difference between the traditional machines and chaotic state machine, we illustrate each of the state descriptions and canonical implementations for the above three state machines.

In section III, chaotic properties of chaotic sequential state machine are specified and verified. The last section concludes the paper with some remarks.

Manuscript received October 7, 1996; accepted January 16, 1998.
Park Kwang-Hyeon is with the Chung Ju National University of Technology.
Hwang Jong-Sun is with the Dam Yang College.
Chung Chong-Eun is with the Korea Atomic Energy Research Institute.

II. Chaotic state machine

1. Mealy and Moore machines[2][3][4]

In this subsection, we introduce the state descriptions and canonical implementations of Mealy and Moore machines for comparing these and our chaotic state machine.

- Mealy machine

A sequential system is specified by means of a state description.
The state description of a sequential system consists of the output function and state-transition function.

<State description>

State -transition function : $S(t+1) = G(S(t),X(t))$
Output function : $Z(t) = H(S(t),X(t))$

In a Mealy machine the output at time t depends on the state at time t and on the input at time t.

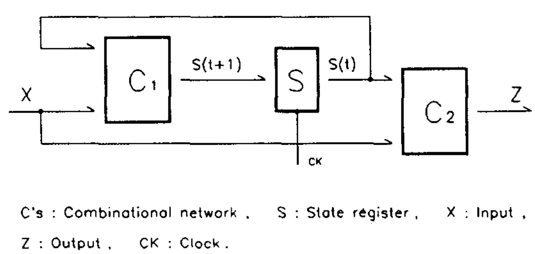


Fig. 1. Canonical implementation of Mealy machine

<Canonical implementation>

The canonical implementation (also called Huffman-Moore implementation) of a sequential system is based directly on its state description. Fig. 1 shows the canonical implementation of Mealy machine.

The separation of the combinational network into two independent subnetworks is made only to illustrate the difference; in a practical implementation this separation would not be made since both networks can share some modules.

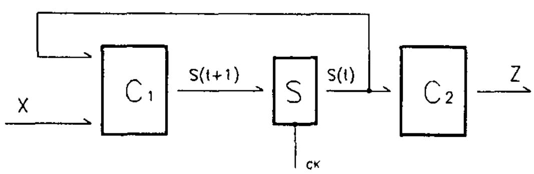
- Moore machine

<State description>

State-transition function : S(t+1) = G(S(t),X(t))
Output function : Z(t) = H(S(t))

In a Moore machine the output at time t depends only on the state at time t.

<Canonical implementation>



C's : Combinational network , S : State register , X : Input , Z : Output , CK : Clock .

Fig. 2. shows the canonical implementation of Moore machine.

2. *Chaotic State Machine*

In this subsection, we suggest a new state machine for generating chaotic but predictable states, inputs and outputs by using chaotic dynamics for the functions of one variable.

[Definition]

Chaotic state machine is a state machine which has the characteristics of chaotic motions.

Let us begin with the consideration of digital systems. Digital systems are classified into two classes: Sequential systems and combinational systems.

In sequential systems, the output at time t depends not only on the input at time t but also on previous inputs. The system has memory with more than one state.

The state description of sequential system is

State-transition function : S(t+1) = G(X(t),S(t)),
Output function : Z(t) = H(X(t),S(t)).

In the case of Moore, output function which was already shown in subsection A is

$$Z(t) = H(S(t)).$$

In combinational systems, on the other hand, the output at time t depends only on the input at time t. We can say that the system has no memory, since the output does not depend on previous inputs. Because of this, the concept of state has no significance for combinational systems, and therefore state-transition function does not exist.

The output function of combinational system is

$$Z(t) = H(X(t)).$$

With the consideration of the descriptions for traditional sequential systems, we specify a chaotic state machine which has the characteristics of chaotic dynamics for functions of one variable.

[Proposition]

A state description and the canonical implementation of chaotic state machine are based on the descriptions of traditional sequential systems.

(Proof) One advantage with the above proposition is that we can obtain the concept of chaotic state machine easily.

<State description>

The state description of a chaotic state machine is written as follows:

State-transition function is

$$S(t+1) = D(S(t), X(t)).$$

Output function is

$$Z(t) = H(X(t)).$$

[Lemma 1] Function D is a deterministic chaotic function.

(proof) Chaotic output is produced either when function D is a deterministic chaotic function[10] or when a strongly chaos function is used.[7]

Recall that the form of the state description is similar to that of the Moore system.

[Theorem]

Chaotic state machine using the characteristics of chaotic dynamics for functions of one variable has the properties of chaotic motions.

(Proof) The proof of the properties for a chaotic state machine is given in section III.

<Canonical implementation>

The canonical implementation of a chaotic state machine is also obtained directly from its state description. We show the canonical implementation in Fig. 3.

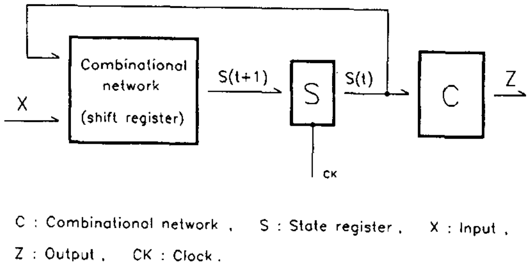


Fig. 3. Canonical implementation of chaotic state machine

<Time behavior>

Time behavior of a chaotic state machine corresponds to the description of a function D such that

$$Z = D(X, S_{INIT})$$

where X and Z are time functions and not the values at a particular t (these we call X(t) and Z(t), respectively), and S_{INIT} is the state before the input time function is applied. Specially, in the case of chaotic state machine, S_{INIT} is directly connected to input so that its output at time t+1 is equal to the input at time t. Time behavior is illustrated in Fig. 4.

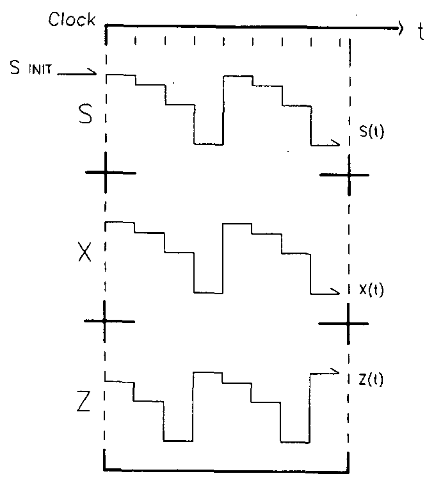


Fig. 4. Time behavior of chaotic state machine

III. Chaotic properties of chaotic state machine

In order to demonstrate the properties of chaotic state machine, we introduce the three common characteristics of chaos as follows [8][9][10][12]:

1. Sensitive dependence on initial conditions
2. Dense periodic points, and
3. Mixing (also called transitivity [7])

Then, to show the above characteristics for chaotic state machine, we are now going to discuss in turns how these appear in chaotic state machine.

1. Sensitive dependence on initial conditions

There are mamy chaos functions that can produce chaotic behaviors. However, in our discussion and in this subsection, the focus of our presentation is how the baker's function (also called saw-tooth function) appears as the basic chaos generator in chaotic state machine. Though this approach may seem rather artificial, it has turned out that the qualitative phenomena of the saw-tooth operation are in fact the one of paradigm of chaos in dynamical systems. And by the saw-tooth transformation the properties of chaotic state machine can be observed and completely analyzed mathematically. The characteristic of sensitivity is central to chaos. To illustrate sensitive dependence on initial conditions of chaotic state machine, we first turn to the saw-tooth transformation. We begin with a new notation for the saw-tooth function which is different from the original definition as follows:

Original equation is

$$S(X) = \begin{cases} 2X & \text{if } X < \dfrac{\text{Unit interval}}{2} \\ 2X-1 & \text{if } X \geq \dfrac{\text{Unit interval}}{2} \end{cases}$$

where unit interval = [0,1), in which ") " means that "1" is not included.

New notation is

$$\text{Frac}(X) = X-k \quad \text{if } k \leq X < k+1, \ k \text{ integer.}$$

With new notation, the saw-tooth transformation can be written as

$$S(X) = \text{Frac}(2X) \quad \text{for } 0 \leq X \leq 1 ,$$

and we reveal a new interpretation by passing to binary representation of the real number X in unit interval or for any decimal number.[6]

For example, $\frac{1}{2}$ = 0.1, $\frac{3}{4}$ = 0.11, $\frac{1}{3}$ = 0.010101 · · · = 0.$\overline{01}$ (over lining means periodic repeatition), 7 = 111, 12 = 1100, etc.

Now we can consider the saw-tooth transformation to be shift operation. Multiplication by 2 in the equation S(X) = Frac(2X) means passing from $0.a_1a_2a_3 \cdots a_k$ to $a_1a_2a_3 \cdots a_k$, where the a_k is binary digit, i.e., each a_k is either 0 or 1.[5]

Moreover, if we remove the binary point in the binary representation of a real number x in the unit interval, there is no difference between the binary repesentation of a real number x in the unit interval and that of any decimal number. A binary representation without the binary point is only desired as an input of the shifter or state register in the electronic circuits.

Let us argue that the property of sensitivity holds for the shift operations of the shifter. One useful observation related to binary expansions is the following.

Let x and y be decimal numbers having 4-bit binary representations

$$x = 1110 \text{ and } y = 1101$$

and we implement a chaotic state machine with the block diagrams of shifter and state register in Fig. 5.

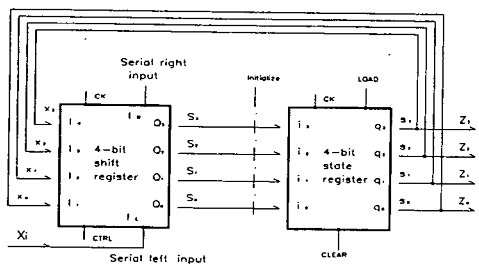


Fig. 5. Implementation of chaotic state machine

Tables 1 and 2 illustrate the operation of a 4-bit bidirectional shift register in Fig. 5.

Comparing table 1 and table 2, we can verify that there exists number x arbitrarily close to number y such that the outcome of the shift operation started at numbers x and y will eventually differ by a certain threshold. This threshold must be the same for all numbers x of the binary representation and is called the sensitivity constant.

However, the phenomenon of sensitivity always magnifies even by one digit deviation. By the previous descriptions and illustrations, the proof of sensitivity in chaotic state machine is completed. Next we will exhibit the property of dense periodic points.

2. Dense periodic points

What happens if we specify the initial state of chaotic state machine as follows:

$$x = a_1a_2a_3\cdots a_8.$$

Undoubtedly we have an 8-cycle state repeats after 8th left shift operation.

Also, what happens if we specify the initial state as follows:

$$y = a_1a_2a_3\cdots a_k.$$

By left shift operation, clearly we can find periodic points in all 2^k subintervals. But more importantly, for any given number x, we can find a number y arbitrarily close to x, which is periodic by shifter.

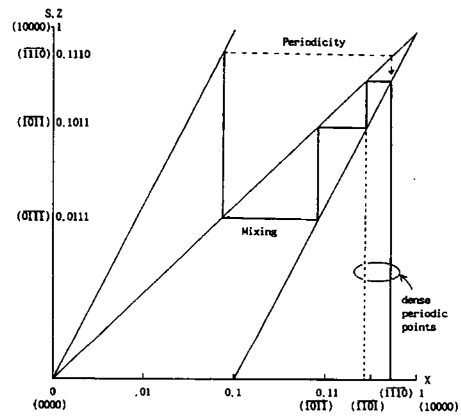
Let us see how they work like dense periodic points. If $x = a_1a_2a_3\cdots a_{n-1}a_n$ for some n then choose $y = .a_1a_2a_3\cdots a_{k-1}^{*}a_k^{*}$ for some k

where n=k and a^* means the dual binary digit, that is,

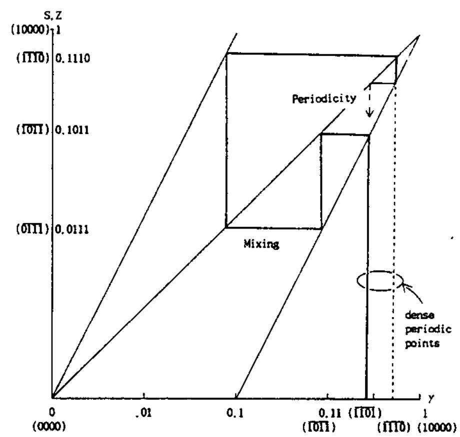
$$a^* = \begin{cases} 1 & \text{if } a=0 \\ 0 & \text{if } a=1. \end{cases}$$

then x and y differ only by (at most) 2^1, and y is periodic. This means that periodic points are dense for the shift operation.

An illustrative example for dense periodic points is given in Fig. 6 by using Tables 1 and 2.



(a) Cycle of periodic 4 for initial state x=1110



(b) Cycle of periodic 4 for initial state y=1101

Fig. 6. Illustration of dense periodic points and mixing by using two period 4 cycles

Table 1. Left shift operations for x = 1110

Number of Shift Operations	Output	Q	Decimal Numbers	Period
	INITIAL State	1110	14	
	LOAD I=1110	1110	14	
1	LSH	1101	13	1st period with periodic 4-cycle
2	LSH	1011	11	
3	LSH	0111	7	
4	LSH	1110	14	
5	LSH	1101	13	2nd period with periodic 4-cycle
6	LSH	1011	11	
7	LSH	0111	7	
8	LSH	1110	14	
9	LSH	1101	13	3rd period
.	.		.	.
.	.		.	.
.	.		.	.

Table 2. Left shift operations for y = 1101

Number of Shift Operations	Output	Q	Decimal Numbers	Period
	INITIAL State	1101	13	
	LOAD I=1101	1101	13	
1	LSH	1011	11	1st period with periodic 4-cycle
2	LSH	0111	7	
3	LSH	1110	14	
4	LSH	1101	13	
5	LSH	1011	11	2nd period with periodic 4-cycle
6	LSH	0111	7	
7	LSH	1110	14	
8	LSH	1101	13	
9	LSH	1011	11	3rd period
.	.		.	.
.	.		.	.
.	.		.	.

3. Mixing

The mixing property is if we can get everywhere from anywhere. It is straightforward to check this property for chaotic state machine.

Let us choose any two initial number x and y as follows:

x = 1110, and y = 0111.

Next we initialize the state register with number x which, after exactly 3 iterations of the left shift operation, will be equal to number y.

In the case of the saw-tooth transformation (in other words, by shift operation), we can hit any target number and state with chaotic state machine according to time t.

An illustrative example was already shown in Fig. 6.

With the previous two subsections and this subsection, the proof of three characteristics for chaotic state machine is completed.

IV. Conclusion

In this paper our goal has been to provide chaotic motion to a sequential systems or digital systems, and we have shown that chaotic motion of the output and state of chaotic state machine also rely on the dynamics only through the deterministic chaotic functions.

We have demonstrated the motion of chaotic state machine using only the saw-tooth function because of the simplicity and substitution property into other chaotic functions, like the tent function and quadratic functions.[8][10][11]

If chaotic state machines are designed and implemented by using other chaotic functions of more than one variable, there must be more various chaotic motions.

If properly designed, chaotic system may capture the essential features of the complex electronic circuits with chaotic motions.[12] But these remain as future research areas.

References

[1] ZVI KOHAVI, Switching and finite automata theory, Newyork: McGraw-Hill, Inc., pp.275~315, 1978.

[2] Milos D. Ercegovac and Tomas Lang, Digital systems and hardware/firmware algorithms, Newyork: John Wiley & Sons, Inc., pp.269~381, pp.393~407, 1985.

[3] Goger C. Alfred, Programmable Logic Designer's Guide, Howard W. Sams & Company, pp.34~45, 1989.

[4] Anthony Palson and Edwin D. Reilly, Encyclopedia of computer science, Newyork: Van Nostrand Reinhold, pp.1191~1193, 1993.

[5] Saburo MUROGA, VLSI system design, Newyork: John Wiley & Sons Inc., pp.308~310, 1982.

[6] GORDON B. DAVIS, Computer Data Processing, McGraw-Hill Book Company, pp.164~186, 1981.

[7] DENNY GULICK, Encounters with Chaos, Newyork: McGraw-Hill, Inc., pp. 96~103, 1992.

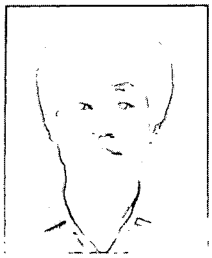
[8] Heinz-Otto Peitgen, Hartmut Jürgens, and Dietmar Saupe, Chaos and Fractals, Springer-Verlag Newyork, Inc., pp.509~583, 1992.

[9] R. L. Devaney, An Introduction to Chaotic Dynamical Systems, Addison-Wesley, Redwood City, 1989.

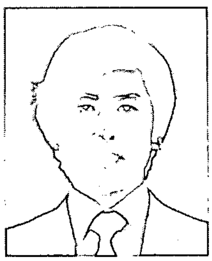
[10] Heinz Georg Schuster, Deterministic Chaos, Weinheim: VCH Verlags gesell-schaft mbH, pp.1, pp.21~36, 1989.

[11] Heinz-Otto Peitgen et al., FRACTALS, Springer-Verlag Newyork, Inc., pp.71~99, 1991.

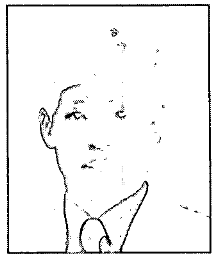
[12] Joseph L. McCauley, Chaos, dynamics, and fractals, Cambridge University Press, pp. 278~308, 1993.



Kwang Hyeon Park was born in 1954. He received the B.S. degree in radio communication enginnering from HanYang University, Korea, in 1977, and the M.S. and Ph.D. degrees in electronic engineering from ChongJu University, Korea, in 1989 and 1996, respectively. From 1980 to 1996 he wroked as a senior researcher at Korea Atomic Energy Research Institute. At present he serves as a lecturer at ChungJu National University of Technology. His current research interests include chaos communication and nonlinear dynamics.



Chong Eun Chung was born in 1954. He received the B.S. degree in electronic engineering from HanYang University, Korea, in 1977, and the M.S degree in electronic engineering from Yon Sei University, Korea, in 1983 and Ph.D. degree in electronic engineering from ChongJu University, Korea, in 1997. Since 1979 he has been working at the division of instrumentation technology at Korea Atomic Energy Research Institute. His current research interests include chaos communication and nonlinear dynamics.



Jong Sun Hwang was born in Korea on December 13, 1953. He received the B.S degree in electrical engineering from Hanyang University, Seoul, Korea, in 1977. The M.S. degree in industry engineering from Hanyang University, Seoul, Korea, in 1979. The Ph.D. degree in electrical engineering from Chonbuk National University, Chonju, Korea, in 1995. The Ph.D. degree in electrical engineering from Chonbuk National University, Chonju, Korea, in 1995. From 1987 to 1998 he worked as a principle researcher at Korea Atomic Energy Research Institute. At present he serves as a lecturer at Provincial college of Damyang. His current research interests include superconducting material, communication and circuit of applications.