

System Generator를 이용한 카오스 신호 발생기의 FPGA 구현

허용원* · 하정우* · 장은영* · 변건식*

*동아대학교

FPGA Implementation of Chaotic Signal Generator Using System generator

Yong-won Hur* · Jeong-woo Ha* · Eun-young Jang* · Kun-sik Byon*

*Donga University

E-mail : ksbyon@dau.ac.kr

요 약

카오스 신호는 공학, 의학 그리고 생물학과 같은 모든 분야에서 많이 이용되고 있으며 최근 카오스 신호를 이용한 디지털 통신시스템에 대한 연구도 활발히 이루어지고 있다. 본 논문은 카오스 신호를 디지털 통신시스템에 적용할 목적으로 System Generator를 이용하여 비선형 방정식으로 구성된 6개의 카오스 신호 발생기를 설계하고 FPGA로 하드웨어를 구현하였다. 이 결과로부터 Hardware co-simulation으로 본 설계를 검증하기 위해 비트 스트림을 FPGA 보드로 로드하였다. 또한 6개의 카오스 발생기의 성능을 평가하기 위해 타이밍 해석을 통한 최대 동작 주파수와 사용한 로직의 resource량을 조사하여 비교하였다.

ABSTRACT

A chaos signal is used in all fields like engineering, a medical science and a biology very much, and study regarding the digital communication system that used a recent chaos signal is consisting actively. Applied a chaos signal in a digital communication system, and this paper designed six chaos signal generator to have been composed of by nonlinear equations as used System Generator, and implemented hardware to FPGA. Loaded bit stream to a FPGA board in order to verify this design to Hardware co-simulation from these results. Also, compared as investigated the maximum action frequency through timing analysis and resource of logic in order to evaluate performance of six chaos generator.

키워드

카오스, System Generator, FPGA

I. 서 론

카오스 신호는 비선형적이면서 랜덤하게 생성되는 특성을 가진다. 랜덤 한 신호는 어떤 수단으로 예측 불가능한 신호이며 현재 생성되는 값은 과거의 값과 무관하게 생성된다. 반면에 카오스 신호는 원리적으로 예측이 가능하며 미래의 결과 값은 과거의 값에 의해 확실한 방법으로 구해질 수가 있다. 하지만 초기 조건에 민감하게 종속하기 때문에 초기 조건을 정확히 모른다면 미래의 값을 예측할 수 없다.

본 논문에서는 이러한 특성을 가지는 6개의 카

오스 신호를 Xilinx사의 System Generator를 이용하여 FPGA를 구현하였으며 Target device는 Xilinx spartan-3 xc3s1000 fg676 -4를 사용하였다.

II. 카오스 신호 발생기 회로

1. Logistic map

Logistic map[1]은 Robert may가 제안한 방정식이다. 이 방정식은 개체군밀도의 성장을 R을 모델링하는 방법으로 방정식은 식 (1)과 같다.

$$x_{n+1} = Rx_n(1 - x_n) \quad (1)$$

Logistic 방정식은 R의 값이 3.57이하의 값을 가지면 현재상태의 출력 값 x_{n+1} 이 선형적인 특성을 가지게 되며 3.57이상의 값을 가지면 2차항 x_n 에 의해 현재상태의 출력 값이 비선형적인 특성을 가지게 된다. 그림 1은 Logistic map을 System Generator로 구현한 것이며 그림 2는 이의 Attractor를 보여준다.

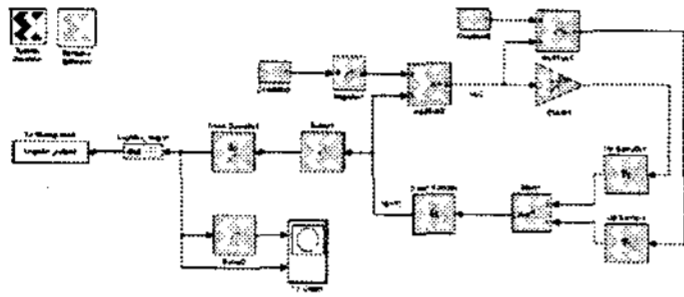


그림 1. Logistic map - System Generator

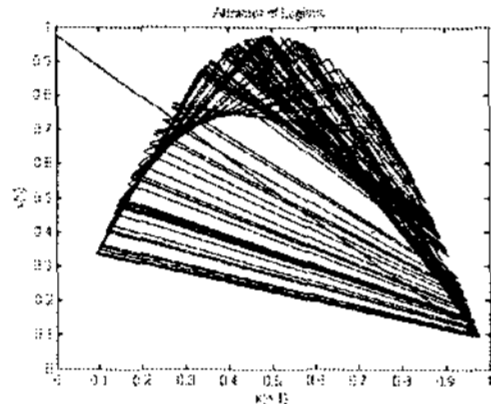


그림 2. Logistic map Attractor (R=3.9)

2. Lorenz

E. Lorenz에 의해 제안된 Lorenz[2] 카오스 신호 방정식은 초기 조건에 민감한 날씨 정보에서 기원하였다. Lorenz 방정식은 x, y, z 변수 값을 가지는 3개의 미분방정식으로 구성되어 있으며 다음의 식으로 표현할 수 있다.

$$\frac{dx}{dt} = 10(y - x), \quad \frac{dy}{dt} = 28x - y - xz, \quad \frac{dz}{dt} = xy - \frac{8}{3}x \quad (2)$$

그림 3은 Lorenz 방정식을 System Generator 블록을 이용하여 구현한 회로이다. 미분방정식의 상수 값은 CMult 블록을 이용하여 각각 설정하였으며 적분기는 Register 블록과 AddSub 블록을 이용하여 Subsystem으로 구성하였다.

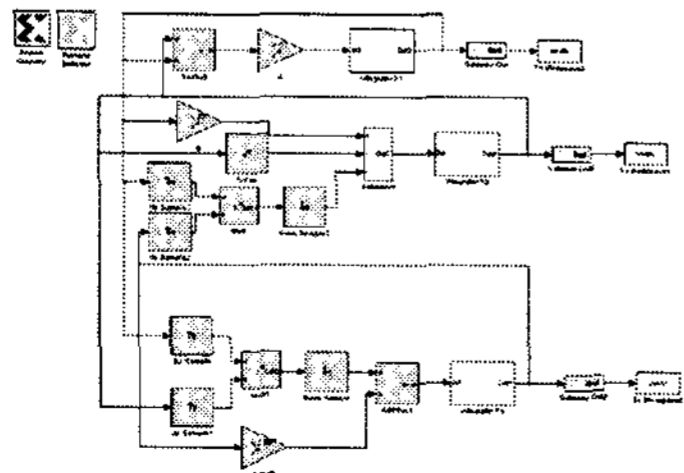


그림 3. Lorenz - System Generator

그림 4는 Lorenz 방정식의 Attractor를 나타내

며 3개의 출력 값을 3차원 형태로 표현한 결과이다. Lorenz 신호의 3차원 그래프는 나선형이며 같은 궤도를 반복하지 않는 비주기적인 신호임을 알 수 있다.

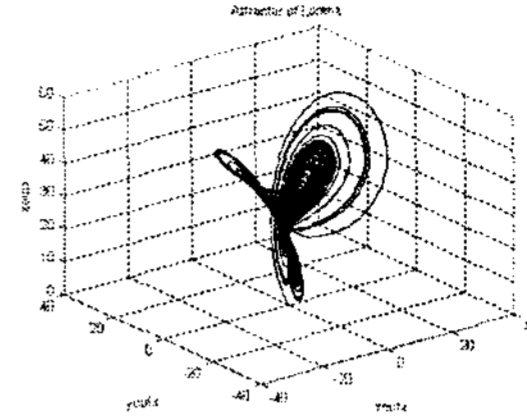


그림 4. Lorenz Attractor

3. Rössler

O. E. Rössler[3]에 의해 제안된 카오스 신호 방정식은 3개의 방정식에 대해 단지 하나의 비선형만을 가지고 있기 때문에 Lorenz 신호 발생기보다 더 간단하다. Rössler 방정식은 식 (3)과 같이 나타낼 수 있다.

$$\frac{dx}{dt} = -(y + x), \quad \frac{dy}{dt} = x + Ay, \quad \frac{dz}{dt} = B + z(x - C) \quad (3)$$

그림 5는 Rössler 방정식을 System Generator로 구현한 회로이며 그림 6은 Rössler 방정식의 Attractor를 나타낸다.

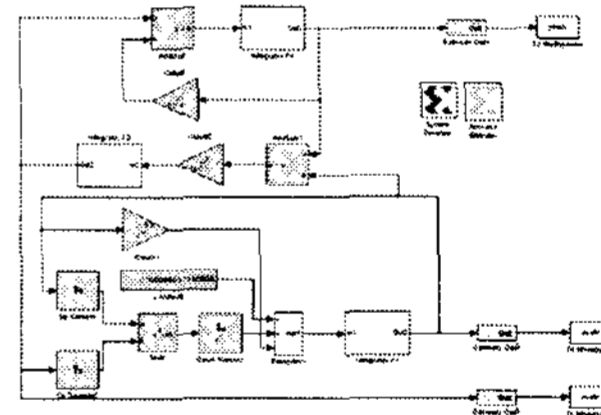


그림 5. Rössler - System Generator

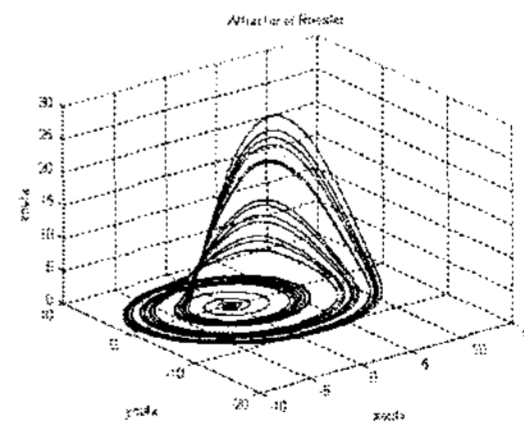


그림 6. Rössler Attractor

4. Linz-Sprott

Linz-Sprott[4] 방정식은 절대치 형태인 $|x|$ 에 대해 비선형을 가진 방정식이다. 매개변수 A는 Attractor의 행적에 영향을 주며 매개변수 B는 Attractor의 크기에만 영향을 준다. Linz-Sprott 방정식은 식 (4)와 같이 3개의 식으로 표현된다.

$$\frac{dx}{dt} = y, \quad \frac{dy}{dt} = z,$$

$$\frac{dz}{dt} = -Az - By \pm (|x| - 1) \quad (4)$$

그림 7은 Subsystem으로 구현한 적분기 3개를 나란히 배열하여 Linz-Sprott 방정식을 구현한 회로를 나타내었다. 매개변수 A의 값은 CMult 블록을 이용하여 0.6으로 설정하였으며 매개변수 B의 값은 Constant 블록을 이용하여 1로 설정하였다.

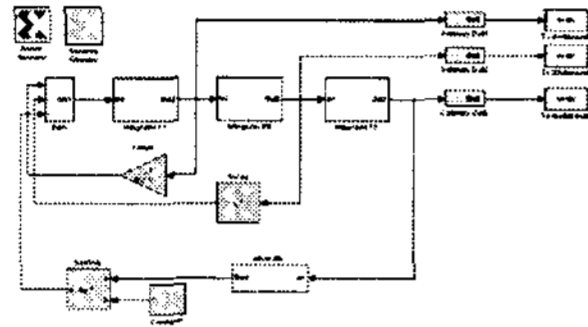


그림 7. Linz-Sprott - System Generator

그림 8은 그림 7의 System Generator 모델을 시뮬레이션 한 결과 그래프이며 Linz-Sprott 방정식의 Attractor를 나타낸다.

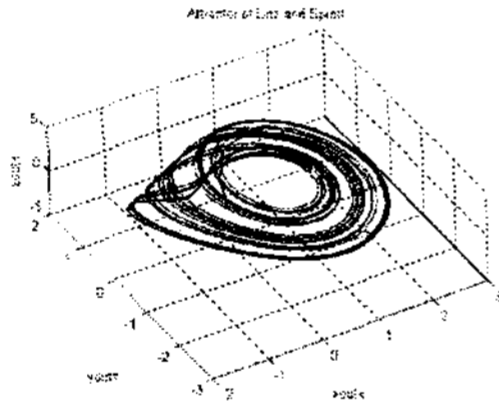


그림 8. Linz-Sprott Attractor

5. Chua

Matsumoto에 의해 제안된 Chua[5] 방정식은 3차 시스템으로 가역성을 가지며 1개의 변수 함수로 특성화되는 비선형 소자와 4개의 선형 소자인 (R, L, C1, C2)로 구성되는 회로이다. Chua 방정식은 다음과 같이 표현할 수 있다.

$$\begin{aligned} C_1 \frac{dv_{c_1}}{dt} &= G(v_{c_2} - v_{c_1}) - g(v_{c_1}) \\ C_2 \frac{dv_{c_2}}{dt} &= G(v_{c_1} - v_{c_2}) - i_L \\ L \frac{di_L}{dt} &= -v_{c_2} - R_L i_L \\ g(V_R) &= m_0 V_R + \frac{1}{2}(m_1 - m_0)(|V_R + B_R| - |V_R - B_R|) \end{aligned} \quad (5)$$

그림 9는 Chua 방정식을 System Generator로 구현한 회로이며 그림 10은 Chua 방정식의 Attractor를 나타낸다. Chua 신호 발생기는 회로의 복잡성이 높은 반면에 많은 매개변수를 가지므로 다양한 Attractor를 생성할 수 있는 장점이 있다.

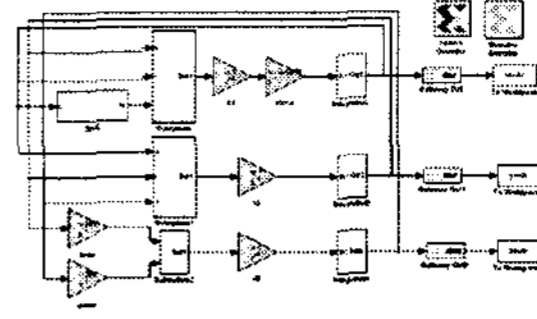


그림 9. Chua - System Generator

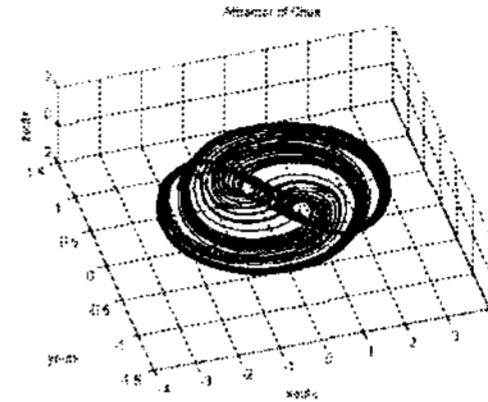


그림 10. Chua Attractor

6. Tent map

Tent map 방정식은 현재 입력으로 이전 출력 값을 사용하며 매개변수 α 의 값이 증가할수록 출력 값의 Attractor의 궤적은 더 큰 폭으로 변화하게 된다. Tent map의 방정식은 식 (6)과 같다.

$$x_{n+1} = \alpha |0.5 - |0.5 - x_n || \quad (6)$$

그림 11은 Tent map 방정식을 System Generator로 구현한 회로이다. 매개변수 α 의 값은 1.9로 설정하였으며 초기치는 Constant 블록을 이용해 0.1로 설정하였다. 그림 12는 Tent map의 Attractor를 나타내며 삼각형 모양으로 궤적을 그리는 것을 확인 할 수 있다.

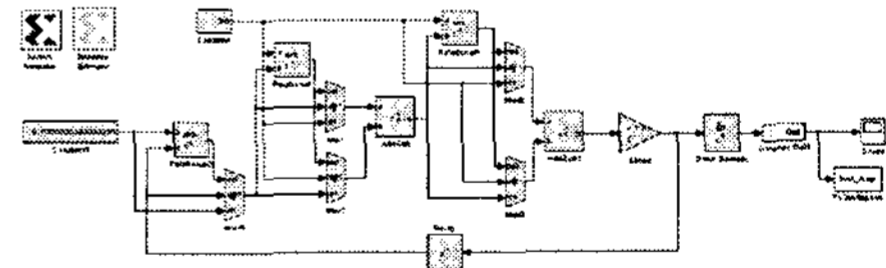


그림 11. Tent map - System Generator

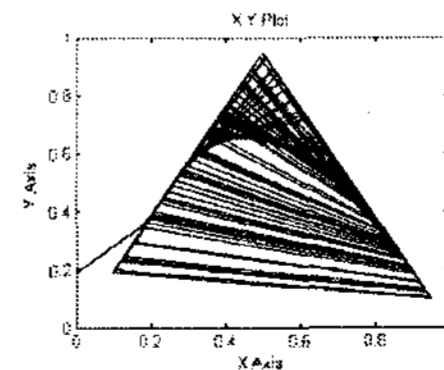


그림 12. Tent map Attractor ($\alpha=1.9$)

III. Hardware Co-simulation

설계한 6개 카오스 신호 발생기 각각의 System Generator 블록에서 Hardware Co-simulation 항

목을 선택 후 Generate를 시키면 JTAG Co-sim 블록이 생성된다. 이 JTAG Co-sim 블록은 합성된 비트스트림 파일을 Target device에 전송하여 시뮬레이션을 시키는 역할을 한다. 그림 13은 생성된 JTAG Co-sim 블록으로 회로를 재구성한 회로이다. Spartan3 board에 JTAG 케이블을 연결한 후 시뮬레이션 하여 회로를 검증하였다.

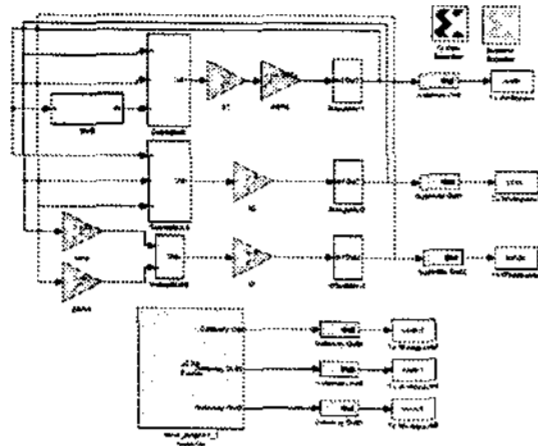


그림 13. Chua - Hardware Co-simulation

Chua 회로의 System Generator 출력과 JTAG Co-sim 블록의 출력을 각각 Workspace에 저장한 뒤 두 결과 값을 3차원 그래프로 출력하여 비교하면 두 출력의 Attractor가 정확히 일치함을 확인할 수 있으며 나머지 5개의 카오스 신호 발생기에 대해서도 System Generator의 출력과 Hardware Co-simulation한 결과 출력이 동일함을 확인하였다.

IV. 타이밍 해석 및 resource량 추정을 통한 성능 평가

6개의 카오스 신호 발생기를 Timing Analyzer를 이용하여 최대 동작 주파수를 측정하였다. 카오스 신호 발생기의 타이밍 해석의 결과는 표 1과 같다. 최대 동작 주파수 관점에서 성능을 평가할 때 Chua, Linz and Sprott, Tent map, Logistic 이 4가지의 결과는 비슷한 것을 확인할 수 있었으며 Lorenz 및 Rössler의 결과는 다소 떨어짐을 확인 하였다.

표 1. 타이밍 해석 결과

	최대 동작 주파수(MHz)
Logistic map	47.501 MHz
Lorenz	33.981 MHz
Linz and Sprott	50.058 MHz
Rössler	35.159 MHz
Chua	52.392 MHz
Tent map	50.005 MHz

FPGA에서 Mapping 과정을 거치게 되면 사용자가 작성한 모든 로직은 FPGA 내부에 있는 기본유닛과 일대일로 Mapping되므로 얼마나 많은 로직을 사용했는지 알아보기 위해 Synthesis, Implementation 과정을 거친 후 Map report를

확인하여 Resource량을 추정하였다. 표 2는 6개 카오스 신호 발생기의 Resource량을 추정한 결과이다.

표 2. Resource량 추정 결과

	Logistic	Lorenz	Linz-Sprott
Slices	505	7395	2124
FFs	452	8269	468
LUTs	698	13825	4094
Emb. Mults	4	0	0
	Rössler	Chua	Tent map
Slices	5372	1413	913
FFs	3407	1371	521
LUTs	10327	2708	1643
Emb. Mults	0	0	0

V. 결론

본 논문에서는 6개의 카오스 방정식을 System Generator를 이용하여 카오스 신호 발생기를 설계하였다. System Generator를 이용하여 HDL 코드를 생성한 후 Synthesis, Implementation 과정을 Xilinx의 ISE 소프트웨어를 이용하여 FPGA로 하드웨어를 구현하였다.

Timing Analyzer 와 Mapping report를 이용하여 성능을 평가하였다. 설계한 6개의 카오스 신호 발생기를 비교하면 Resource면에서는 Logistic 신호 발생기가 가장 적은 기본유닛을 가지는 것을 확인하였으며 최대 동작주파수 면에서는 Chua 신호 발생기가 가장 빠른 것을 확인하였다. 최종적으로 Hardware Co-simulation을 이용하여 설계를 검증하였다.

참고문헌

- [1] R. M. May, "Simple Mathematical Models with Very Complicated Dynamics," Nature 261, 1976.
- [2] N. E Lorenz, "Deterministic non periodic flows," Journal of Atmospheric Science, Vol. 20 n. 02, 1963.
- [3] O. E Rössler, "An equation for continuous chaos," Physics Letter 15A, 1976.
- [4] J. C. Sprott, S. J. Linz, "Algebraically Simple Chaotic Flows." International Journal of Chaos Theory and Applications, vol. 5, n. 2000.
- [5] L. O. Chua, "The genesis of Chua's circuit," Archiv fur Elektronik und Uebertragungstechnik, vol. 46, n. 04, 1992.
- [6] <http://www.xilinx.com>