퍼지 및 지능시스템학회 논문지 2001, Vol. 11, No. 5, pp. 426-432

FPGA를 이용한 진화형 하드웨어 셀계 및 구현에 관한 연구

A Study on Design of Evolving Hardware using Field Programmable Gate Array

반창봉 · 곽상영 · 이동욱 · 심귀보

Chang-Bong Ban, Sang-Young Kwak, Dong-Wook Lee, and Kwee-Bo Sim

중앙대학교 전자전기공학부

요 약

본 논문은 진화형 하드웨어를 이용하여 생물의 정보처리 시스템인 셀룰라 오토마타 신경망의 구현에 관한 연구이다. 셀룰라 오토마타 신경망은 진화 및 발생을 기반으로 한 신경망 모델이다. 진화는 다양성을 주요 근원을 제공하는 돌연변이 및 재 조합 비율에 의하여 비결정론이며, 발생은 결정론 적이며 지역적인 물리현상을 따른다. 셀룰라 오토마타 신경망은 셀룰라 오토마타에 의해 신경망 내부의 각 셀의 상태를 발생시키고, 초기 셀을 유전자 알고리즘의 개체로 간주하여 초기 셀이 진화 알고리즘을 통해 진화함으로서 신경망이 진화하는 시스템이다. 본 논문은 이 시스템을 진화형 하드웨어 이용하여 하드웨어로 구현하였다. 진화형 하드웨어는 진화 알고리즘과 재구성하드웨어의 결합체이다. 즉, 재구성 하드웨어의 구성에 필요한 bit를 유전자 알고리즘의 개체로 간주한 것이다. 진화 알고리즘을 수행하기 위해 유전자 알고리즘 프로세서를 설계하였으며, 셀룰라 오토마타 신경망이 유전자 알고리즘의 개체와 셀룰라 오토마타 룰에 의해 자동적으로 신경망을 생성하기 위해 신경망을 이루는 셀들을 설계하였다. 제안된 시스템의 효율성을 검증하기 위해 Exclusive-OR 문제에 적용하였다.

Abstract

This paper is implementation of cellular automata neural network system using evolving hardware concept. This system is a living creatures' brain based on artificial life techniques. Cellular automata neural network system is based on the development and the evolution, in other words, it is modeled on the ontogeny and phylogeny of natural living things. The phylogenetic mechanism are fundamentally non-deterministic, with the mutation and recombination rate providing a major source of diversity. Ontogeny is deterministic and local physics. Cellular automata is developed from initial cells, and evaluated in given environment. And genetic algorithms take a part in adaptation process. In this paper we implement this system using evolving hardware concept. Evolving hardware is reconfigurable hardware whose configuration si under the control of an evolutionary algorithm. We design genetic algorithm process for evolutionary algorithm and cells in cellular automata neural network for the construction of reconfigurable system. The effectiveness of the proposed system if verified by applying it to Exclusive-OR.

Key Words : 진화형 하드웨어, 유전자 알고리즘 프로세서, FPGA, 셀룰라 오토마타, 신경망

1. 서 론

불확실하고 복잡한 동적 환경에 대처하는 계산 모델로 인공생명의 개념을 이용한 지능정보처리 메커니즘이 있다. 인공생명 연구는 생명체의 특징을 이해하고, 이것을 인공적인 매체에 생명체와 유사한 기능을 갖도록 하는 인공 시스템을 구축하여 생명체가 가지는 우수한 특징을 실현하는 것을 목적으로 한다. 셀룰라 오토마타(CA), 린덴마이어 시스템(L-system), 인공신경망(ANNs), 퍼지 시스템(FS), 진화 알고리즘(EAs) 등은 대표적인 인공생명 모델이다[1-5]. 이 중에 특히 ANNs와 EAs는 인공생명 연구의 대표적인 모델이다. ANNs 의 성능 향상을 위해 ANNs의 구조와 파라미터의 최적화를 위하여 진화적인 접근방법을 도입하고 있다. 그러나 주어진 문제가 복잡해지면 엄청난 진화시간을 필요로 하게 되고, 경우에 따라서는 진화알고리즘의 파라미터 결정에서도 부가적인 문제를 수반하는 등 문제점은 존재한다. 이러한 문제점을 근본적으로 해결하기 위해 신경망의 합성 규칙을 코드화하는 방법이 있다. 이것은 생물체의 발생 과정에서 힌트를 얻은 것으로 L-시스템 기반의 모듈형 신경망[6]과 셀룰라 오토마타 기반의 CAM-Brain[7] 등이 있다. 이러한 방법은 신경망의 크기가 커지더라도 복잡성이 증가하지 않는다는 장점이 있다.

진화형 하드웨어는 유전자 알고리즘과 재구성 가능한 하드웨어의 결합체이다. 즉, 재구성 가능한 하드웨어의 구성에 필요한 bit를 진화 알고리즘의 개체로 간주하여 진화 알고리즘을 수행하여 환경에 적합한 구조로 재구성되는 시스템이다

접수일자 : 2001년 6월 19일
완료일자 : 2001년 9월 30일
감사의 글 : 본 연구는 2001년도 서울시 · 중소기업청 산학연 공동기술개발 컨소시엄사업의 연구비와 과학기술부 뇌과학 프로젝트(Braintech21)에서 일부 연구비를 지원 받아 수행하였습니다. 연구비 지원에 감사 드립니다.

[8]. 본 논문에서는 진화형 하드웨어를 이용하여 셀룰라 오토마타 신경망을 FPGA 칩에 구현하였고 그 유효성을 검증하였다. 셀룰라 오토마타 신경망(ECANS)은 두 가지 인공생명 모델인 발생모델의 셀룰라 오토마타와 진화 모델의 진화 알고리즘으로 구성되어 있다[9]. 셀룰라 오토마타 신경망은 진화를 통하여 환경에 대한 적응성을 획득한다. 이것은 동물의 본능과 같은 것으로서 유전적 정보에 따라 결정된다. 따라서 이 시스템의 주요한 요인은 뉴런의 종류와 배열이다. 이는 여러 가지 뉴런의 배열에 따라서 신경망의 기능이 결정되는 것에 기인한다. 신경망은 초기 셀의 배열에 따라 다양한 형태를 갖게 된다. 이 시스템에서 배열의 형태는 셀룰라 오토마타에 의해 결정된다. 셀룰라 오토마타 신경망이 기존의 신경망과의 다른 점은 연결강도 보다는 뉴런과의 연결방식에 따라 전체 시스템의 기능이 달라진다는 점이다. 따라서 각 뉴런은 기존의 뉴런에 비해 복잡한 특성을 갖는 Nagumo-Sato가 제안한 뉴런[10-11]을 사용하였고, 이를 FPGA 칩 내에 구현하기 용이하도록 수정하여 적용하였다. 이 시스템을 FPGA에 적용하기 위해 유전자 알고리즘 프로세서를 이용하였다. 유전자 알고리즘 프로세서 내에서는 유전자 알고리즘의 연산자인 교차와 돌연변이를 위한 연산 모듈(Operating Module)와 개체군의 재생산을 위한 재생산 모듈(Re-Production Module) 그리고, 주 제어기로 이루어져 있다. 그리고, 개체군과 적합도 값을 저장하기 위 한 외부 메모리도 존재한다. 그리고, 신경망 내부의 하나의 뉴런은 진화 알고리즘 프로세서에 의해 새로 생성된 비트열들에 의해 내부구조를 스스로 바꿀 수 있도록 설계되었다. 유전자 알고리즘 프로세서를 이용하여 신경망이 자동적으로 진화함으로써, 진화형 하드웨어를 이루게 된다. 제안한 시스템의 유효성을 검증하기 위하여 Exclusive-OR 문제에 적용하였다.

2. 진화하는 셀룰라 오토마타 신경망의 개요

발생, 진화, 학습은 자연계 생물체에서 일어나는 3대 자기조직화 현상이다. Sipper등은 이를 기반으로 POE(Phylogeny, Ontogeny, Epigenesis) model을 제안하였다. 본 논문에서는 이중 진화와 발생을 기반으로 한 신경망 모델을 개발하였다. 진화는 다양성을 주요 근원을 제공하는 돌연변이 및 재 조합 비율에 의하여 기본적으로는 비결정론이다. 이러한 다양성은 변화하는 환경에서 계속적인 적응을 통한 종의 생존을 위하여 필수적이다. 한편, 발생은 다세포 생물의 발생과정으로, 본질적으로 결정론적이며 지역적인 물리현상을 따른다. 그림 1은 제안된 구조의 개념도이다. 셀룰라 오토마타 신경망은 발생과 진화의 두 단계를 거쳐 생성된다. 즉, 네트워크는 초기 셀의 발생을 통해 생성되며, 주어진 환경에 적응할 수 있는 구조로 발전하도록 진화한다.

3. 셀룰라 오토마타 신경망

셀룰라 오토마타[12~13]는 시간과 공간의 상태가 이산적인 동적 시스템이다. 셀룰라 오토마타에서 이산적인 양으로 분할된 공간을 셀이라 부른다. 이 셀은 한 시간에 유한개의 상태 중의 한가지를 가질 수 있다. 격자구조 속의 셀의 상태는 국소적인 규칙에 의하여 수정된다. 국소적인 규칙에 의해 주어진 시간의 셀의 상태는 한 단계 전의 자기 자신의 상태와 근처 주변 셀의 상태에만 의존한다. 또한 격자상의 모든

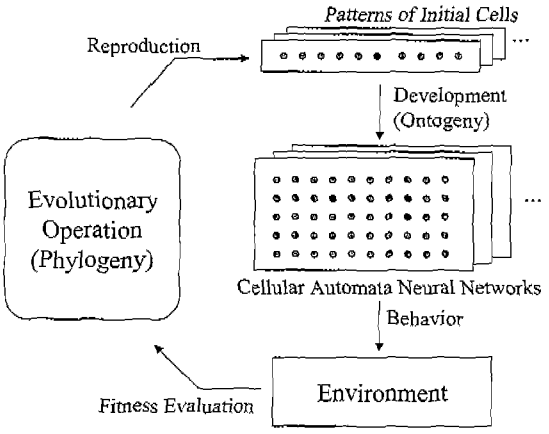


그림 1. 진화하는 셀룰라 오토마타 신경망의 개념도
Fig. 1. A conceptual diagram of ECANS

셀의 상태는 일괄적으로 수정된다.

3.1 셀의 종류

제안된 시스템에서 셀의 상태는 셀룰라 오토마타의 셀과 같이 이웃 셀과의 관계에 따라 정의된다. 기본적인 연결방식은 단지 인접한 몇 개의 셀들과만 연결을 하는 것이다. 연결은 흥분성 연결, 억제성 연결, 연결 없음 의 3가지 형태 중 하나를 가진다. 각 셀은 이웃의 셀 및 다음 상태의 자신의 셀과 연결할 수 있다. 따라서 총 연결할 수 있는 가짓수는 이 된다. 셀룰라 오토마타에서 시스템의 복잡도 및 규칙(rule)의 크기는 하나의 셀이 가질수 있는 상태에 기인한다. 본 논문에서는 연결의 대칭성을 고려해 그림 2와 같은 8가지의 셀을 사용하였다.

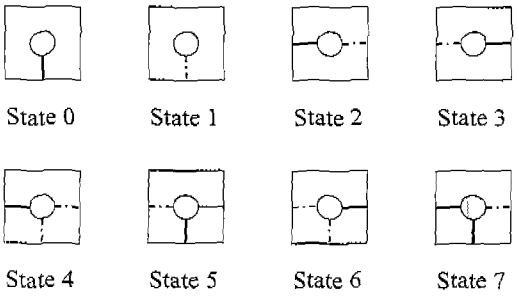


그림 2. 셀의 연결방식에 따른 셀의 8가지 상태
(실선: 흥분성 연결, 점선: 억제성 연결)
Fig. 2. States of cells (cell's connection type)
(dotted line : inhibitory, solid line : exciatory)

3.2 카오스 뉴런 모델

셀룰라 오토마타 신경망의 뉴런은 실제의 신경세포막에 충실하게 만들어진 Hodgkin-Huxley 방정식을 토대로 재구성한 Nagumo-Sato의 신경모델을 변형하여 사용하였다. 이 모델은 비교적 단순한 구조를 가지고 있음에도 불구하고 매우 복잡한 특성을 가진다. 셀룰라 오토마타 신경망의 셀은 동적인 특성을 갖는 카오스 뉴런으로 모델링하고 CA의 규칙에 따라서 발생의 단계를 거쳐 구조화하여 다양한 문제에 적용할 수 있다. Nagumo-Sato의 카오스 뉴런의 동작 방정식

은 다음과 같다.

$$y(t+1) = u(x(t+1)) \tag{1}$$

$$x(t+1) = I(t) - \alpha \sum_{d=0}^{t} k^d y(t-d) - \theta \tag{2}$$

단, $y(t)$는 시간 t에서 출력, $x(t)$는 내부 상태 값, $I(t)$는 입력, $u(\cdot)$는 단위 계단 함수, k^d는 0에서 1사이의 값을 가지는 감쇠계수, 상수 α는 양의 파라미터이며 θ는 뉴런의 문턱 값이다.

카오스 뉴런 모델에서 뉴런의 동작함수는 단위계단 함수의 형태로 나타나기 때문에 입출력 신호는 펄스 형태이다. 따라서 신호의 세기는 펄스의 밀도 변화에 의해서 측정된다. 셀룰라 오토마타 신경망에서 뉴런간의 연결강도는 흥분성 결합, 억제성 결합, 무 결합의 세 가지 상태만 다지므로 네트워크는 매우 단순하게 구성된다. 그러나 뉴런들의 복잡한 특성 때문에 네트워크 전체는 매우 복잡한 행동을 보인다.

3.3 셀룰라 오토마타 신경망

신경망을 구성하기 위해 초기 셀은 셀의 발생과정을 거쳐야 하지만, 얼마나 많은 단계를 발생시켜야 하는지, 얼마나 많은 초기의 셀이 필요한지에 대한 뚜렷한 방법은 없다. 여러 가지의 실험 결과 문제의 복잡도에 따라 약간씩은 다르지만 기본적으로 초기 셀은 입력 뉴런의 2~3배, 발생단계는 5~10단계가 적당하였다. 하지만, 이것은 경험상의 수치일 뿐 보다 타당한 계산법은 더 연구해야할 과제이다.

네트워크는 이미 설계된 뉴런을 배열하여 구현하였다. 그림 4에서는 초기 셀들로부터 셀룰라 오토마타 모듈에 의해 단계적으로 발생하여 네트워크를 구성하는 방법을 보여주고 있다. 네트워크의 각 뉴런과의 연결은 흥분성, 억제성, 무 연결만을 가지므로 부울리안 네트워크와 유사하다.

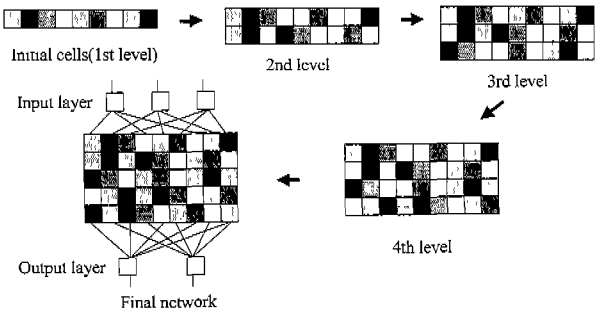


그림 4. ECANS의 발생단계
Fig. 4. Developmental process of ECANS

주어진 문제에 적용하기 위해 진화적 적응방식을 이용한다. 진화를 위해 조작하게 될 염색체는 네트워크의 초기 셀이다. 생물학적인 견지에서는 초기 셀과 함께 세포가 생성되는 규칙이 진화되어야 하지만 본 논문에서는 규칙을 고정시킨 상태에서 초기 셀의 배열을 진화하도록 하였다.

4. 진화형 하드웨어 설계

4.1 진화형 하드웨어

재구성 가능한 하드웨어[14]는 사용자가 소프트웨어적으로 구조를 변경할 수 있는 반도체 집적회로로서 반도체 제조 공정을 거치지 않고 다양한 구조의 반도체를 구현할 수 있기 때문에 많은 분야에서 응용되고 있다. 또한 이것은 환경의 변화에 적응하고 견함에도 견고한 하드웨어 시스템을 구축할 수 있는 길을 열어주고 있다. 재구성 가능한 하드웨어의 대표적 예는 FPGA(field programmable gate array) 이다. 이것은 하드웨어 내부의 구성을 결정하는 비트스트링을 다운로드 받음으로서 임의의 하드웨어 기능을 구현할 수 있다.

하드웨어의 재구성을 적응적으로 수행하기 위해서 현재 가장 많은 사람들의 주목을 받고 있는 기술 중의 한 가지는 진화 알고리즘(evolutionary algorithms)이다[15]. 진화 알고리즘에 의해 그 구조가 자동적으로 변하는 하드웨어를 진화 하드웨어(evolvable hardware)라고 한다. 하드웨어의 구조를 나타내는 비트 스트링을 진화 알고리즘의 염색체로 표현하여 적합도 기반으로 하드웨어의 구조를 진화 하고자 하는 것이다. 즉, FPGA에서는 하드웨어 구조를 결정하는 비트스트링이 있고, 이것을 바꾸어 씀으로 해서 여러 가지의 논리회로를 실현할 수 있는데 이것은 진화 알고리즘에서 해의 후보를 이진 비트스트링으로 나타내고, 이것을 탐색하여 최적의 해를 발견하는 방식을 적용할 수 있게 한다. 적응적 설계 방법은 FPGA의 비트스트링을 유전자 알고리즘에 있어서 염색체로 생각하고 환경에 가장 적합한 비트스트링을 찾는 방법이다.

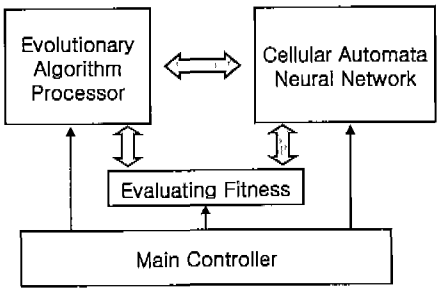


그림 5. 전체 시스템의 개요도
Fig. 5. Diagram of system

본 논문에서는 진화형 하드웨어를 이용하여 셀룰라 오토마타 신경망을 구현한다. FPGA내에 구성될 신경망의 내부 구조를 결정하게 될 비트스트링은 셀룰라 오토마타 신경망의 초기 셀로 한다. 초기 셀이 결정되면 신경망내부에서 스스로 셀 간의 연결 구조를 변경하게 된다. 그리고, 진화 알고리즘의 대표적인 알고리즘인 유전자 알고리즘을 사용하였다. 유전자 알고리즘을 수행하기 위해 유전자 알고리즘 프로세서를 셀룰라 오토마타 신경망과 연결하여, 대부분의 처리 과정을 하드웨어가 담당한다. 유전자 알고리즘의 각 개체들은 적합도 평가부(Evaluating Fitness)에서 따로 이루어진다. 그림 5는 제안된 시스템의 블록 다이어그램이다.

4.2 유전자 알고리즘 프로세서

유전자 알고리즘은 자연계의 생물의 진화과정을 모방한 것이다. 자연계의 생물의 진화과정은, 어떤 세대를 형성하는 개체들의 집합, 즉 개체군 중에서 환경에 대한 적합도가 높은 개체가 높은 확률로 살아 남아 재생되며, 이때 교차나 돌연변이에 의해서 다음 개체군을 형성하게된다[16]. 각 개체는 비트스트링으로 구성되는 염색체로 표현된다. 즉, 유전자 알고리즘은 선택과 교차, 돌연변이 등의 연산자를 이용하여 주어진 문제에 대한 후보 해를 표현하는 개체군을 새로운 개

체군으로 반복하여 생성하면서, 문제의 해를 찾아가게 된다. 개체들은 그들의 적합도에 따라 선택되어 교차나 돌연변이를 거쳐 자식을 복제하게 되는데, 이 때 적합도가 높은 개체가 적합도가 낮은 개체보다 평균적으로 더 많은 자식을 생산하게 된다. 교차는 선택된 개체사이에서 염색체의 일부의 교체에 의해 새로운 개체를 생성하는 연산자며, 돌연변이는 염색체상의 어떤 유전자 좌의 값을 다른 대립 유전자로 치환하는 연산자이다. 유전자 알고리즘은 구조가 간단하고 방법이 일반적이어서 응용범위가 매우 넓지만, 연산 시간이 길다는 문제점을 가지고 있다. 유전자 알고리즘의 심각한 문제인 연산 시간을 줄이기 위해서 유전자 알고리즘의 효율적인 하드웨어 구현 방법이 요구된다[17].

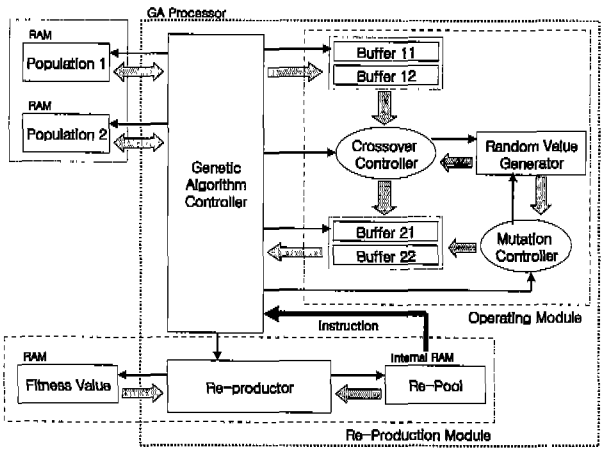


그림 6. 유전자 알고리즘 프로세서의 블록 다이어그램
Fig. 6. Block diagram of genetic algorithm process

그림 6은 제안된 유전자 알고리즘 프로세서의 블록 다이어그램을 나타낸다. 제안된 프로세서 내에는 유전자 알고리즘의 기본 연산자인 교차와 돌연변이 기능을 하는 연산 모듈(Operating Module)와 개체군의 재생산을 위한 재생산 모듈(Re-Production Module), 개체군을 저장하는 2개의 외부메모리, 적합도 값을 저장하는 1개의 외부메모리, 그리고 전체 기능을 제어하는 제어기(Controller)로 이루어져 있다.

기존의 유전자 알고리즘 프로세서는 하드웨어 지향 알고리즘으로, 실제 세대라는 개념을 가지지 않는 정상상태 모델을 기반으로 하였다. 그러나, 제안된 프로세서는 2개의 외부 메모리를 이용하여 세대의 개념을 갖게된다. 즉, 유전자 알고리즘의 각 세대마다 메모리의 위치를 바꾸어 가며 개체군을 저장하게 된다. 그리고 각 개체에 해당하는 적합도 값을 제 3의 외부 메모리를 이용하여 저장하게 된다. 이는 새로운 세대를 구성하기 위해 재생산을 하게 될 때 적합도 값만을 참조하기 위함이다.

Op-Code	Parent 1 ID	Parent 2 ID	Direction1	Direction2

그림 7. 명령어의 구성
Fig. 7. Instruction format

재생산 모듈(Re-Production Module)에서는 적합도 값을 저장하고 있는 메모리에서 적합도 값을 참조하여, 룰렛 선택을 통해 재생될 개체들을 재생 공간(Re-Pool)에 저장한다. 이 때 그림 7과 같이 교차 연산을 하게 될 두 부모 개체의 ID(메모리 내에 저장된 위치를 가리키는 값)에 연산 코드(Op-Code)와 교차 후와 돌연변이 후 자손들이 저장될 위치를 지정해주는 방향 코드(Direction-Code)를 붙여서 하나의 명령어(Instruction)을 만들어 저장한다. 연산 코드는 연산수행의 유무, 데이터 전송의 유무를 가리키는 2개의 비트로 구성되어 있고, ID와 방향코드는 개체가 메모리 내에 저장되어 있거나 저장하게될 상대적 위치 값을 표시한다. 이런 방식으로 다음 세대에 재생산될 개체들을 명령어 형태로 만들어 저장해 놓으면, 제어기(Controller)에서 이 명령어를 읽어들여 실행하게 된다. 두 개의 개체 ID를 참조하여 메모리에서 두 개체를 읽어 들이고, 연산 코드를 참조하여 두 개체의 교차와 돌연변이 연산을 수행한다. 그리고, 연산 후 생성된 자손은 방향 코드(Direction1, 2)를 참조하여 다른 메모리에 차례로 저장하게 된다.

연산 모듈(Operating Module)에서는 유전자 알고리즘의 기본 연산인 교차와 돌연변이를 수행하게 된다. 그림 8은 연산 모듈의 내부 구조를 나타낸다. 그리고, 선택된 두 부모 개체가 교차 직후 돌연변이를 수행하여 빠른 처리 속도를 갖을 수 있다.

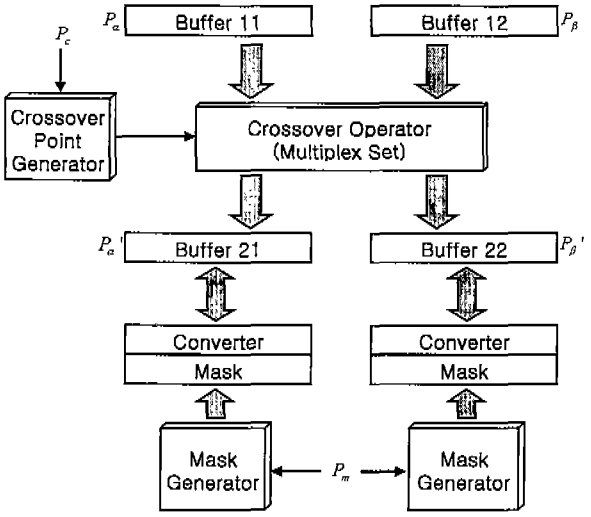


그림 8. 연산 모듈의 내부 구조
Fig. 8. Internal structure of operating module

교차 연산은 단순 교차 연산을 한다. 먼저 교차 연산을 수행하게 될 두 부모 개체를 메모리에서 읽어 들어 Buffer11과 Buffer2에 저장한다. 동시에, 교차점 생성기(Crossover Point Generator)에서는 0과 1사이의 난수를 발생시켜 그 수가 P_c 보다 크면 교차점을 발생하고, 그렇지 않으면 교차점을 0으로 설정한다. 교차점은 1과 염색체의 길이보다 1보다 작은 사이의 정수 값을 난수로 발생시킨다. 교차 연산기(Crossover Operator)에서는 두 부모 개체와 교차점을 참조하여 교차 연산을 수행하고, 연산 후 생산된 자손을 Buffer21과 Buffer22에 저장한다.

돌연변이 연산은 점 돌연변이 연산을 한다. 교차 연산 후 자손이 Buffer21, Buffer22에 저장될 과 동시에, 마스크 발생기(Mask Generator)에서 0과 1 사이의 난수를 발생 시켜 그 수가 P_m보다 크면 마스크 비트(Mask Bit)를 1로 설정하고, 그렇지 않으면 0으로 설정한다. 이 때 염색체의 길이 만큼 마스크 비트를 설정하여 마스크를 생성한다. 변환기(Converter)에서는 마스크를 참조하여 염색체의 해당위치의 마스

크 비트가 1이면 변환하고, 0이면 그대로 두어 돌연변이 연산을 수행한다. 돌연변이 연산 후 자손은 다시 Buffer21, Buffer22에 저장된다.

4.2 진화 신경망의 설계

본 논문에서는 FPGA를 이용하여 셀룰라 오토마타 신경망을 구현하였다. 유전자 알고리즘 프로세서에서는 셀룰라 오토마타 신경망의 초기 셀을 개체로 간주한다. 각 개체들은 신경망의 내부 구성의 열쇠가 된다. 개체는 신경망에 입력되어 초기 셀을 설정하고, 이 셀들로부터 정해진 셀룰라 오토마타 룰에 의해 나머지 셀들의 상태가 결정된다. 각 셀들의 상태는 셀들 간의 연결 방법에 따라 달라진다. 본 논문에서는 이 연결을 자동적으로 구성하도록 신경망 내의 셀들을 설계하여 진화 신경망을 구현하였다. 진화 신경망 내의 각각의 셀들은 셀룰라 오토마타 모듈(Cellular automata Module)과 연결 모듈(Interconnection Module), 그리고 뉴런 모듈(Neuron Module) 의 세 가지 모듈로 구성되어 있다. 그림 8은 제안된 진화 신경망 내부 셀의 블록 다이어그램이다.

셀룰라 오토마타 모듈은 룰 테이블과 이웃 셀들의 상태를 참조하여 자기 셀 상태를 결정한다. 상태가 결정되면, 상태 값은 연결 모듈의 제어 신호로 입력되어 상태에 따른 셀들 간의 연결이 결정된다. 뉴런 모듈에서는 식 (1), (2)와 같은 카오스 뉴런 모델이 구현된다. 식 (1), (2)를 다른 뉴런으로부터의 입력 항을 고려해 식 (3)~(5)와 같이 나누어 나타낼 수 있다.

$$a_i(t+1) = k_e a_i(t) + \sum_{j=0}^{M} v_{ij} I_j(t) \quad (3)$$

$$b_i(t+1) = k_f b_i(t) + \sum_{j=0}^{N} w_{ij} y_j(t) \quad (4)$$

$$c_i(t+1) = k_r c_i(t) - \alpha y_i(t) - \theta_i(1-k_r) \quad (5)$$

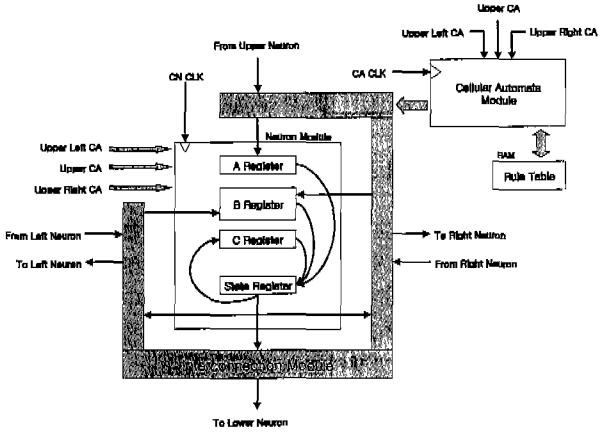


그림 9. 진화 신경망 내부 셀의 블록 다이어그램
Fig. 9. Block diagram of cell in evolving neural network

위 세 식은 그림 9의 A, B, C Register 에 의해 구현되면, 최종 결과는 State Register에 저장된다. 각 Register들은 동작 클럭이 들어 올 때마다 곱 연산을 위해 저장된 값들을 쉬프트 시키고, Register에 들어오는 값을 더하게 된다. 연결 모듈에서는 셀룰러 오토마타 룰에 의해 결정된 셀의 상태를 고려하여 뉴런의 출력과, 뉴런으로 들어오는 입력을 연결하거나 연결을 차단한다.

유전자 알고리즘 프로세서에서는 신경망의 초기 셀을 개체로 하여 유전자 알고리즘을 수행하고, 신경망 내부에서는 이 셀 값과 셀룰라 오토마타 룰을 참조하여 자동적으로 신경망의 구조를 변경하게된다. 즉, 유전자 알고리즘 프로세서와 신경망이 결합하여 진화하는 신경망을 설계하였다.

5. 실험 및 결과

5.1 Exclusive-OR(XOR) 문제

본 논문에서는 제안된 시스템의 성능을 시험해 보기 위하여 선형 분리가 불가능한 Exclusive-OR(XOR) 문제에 적용하였다. 이때 실험에 사용한 파라미터들은 표 1과 같다.

표 1. XOR 실험에 사용한 파라미터
Table 1. The parameters used in experiment

파라미터	XOR
입력의 수	3 : 2+1(bias)
출력의 수	1
초기 셀의 개수	9 : 3×3
셀의 발생 단계	5
상태의 수	8
규칙 : $\phi_i(\sigma_i) = (\sigma_{i-1} + \sigma_i + \sigma_{i+1})\%$(총 상태수)	
$k_e, k_f, k_r = 0.5,\ \theta = 0.5,\ \alpha = 1$	

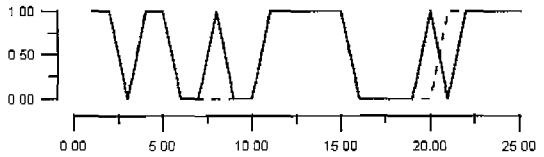


그림 10. XOR 응답 (실선: 응답, 점선: 이상 값)
Fig. 10. Response of XOR problem
(solid : response, dotted : ideal)

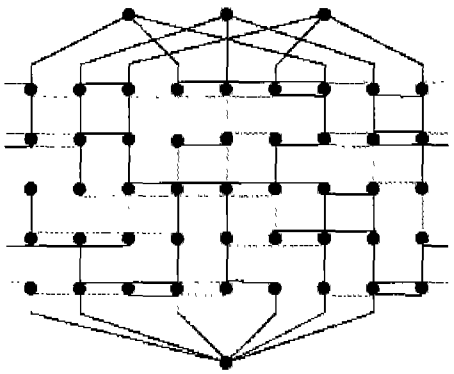


그림 11. 50세대 진화 후 얻어진 XOR 네트워크
Fig. 11. Obtained XOR network after 50 generation

신경망의 진화를 위하여 초기 셀의 패턴을 염색체로 사용하고 그 나머지 셀은 CA규칙에 의하여 발생시켰다. 셀의 종류는 총 8가지이므로 3비트로 코드화하였다. 그러므로 초기 셀의 개수가 9개이므로 염색체의 길이는 27비트이다. 진화

가 효율적으로 일어나기 위해서는 CA의 규칙이 주기와 카오스의 중간 상태인 복잡한 상태가 되도록 설정해야 한다. 이를 위해 CA의 규칙을 비교적 간단하면서도 모든 상태가 나올 확률이 같도록 위의 표와 같이 설계하였다. 그림 10은 50세대 진화 후 얻어진 XOR 신경망의 네트워크 출력 값이다. 25 스텝 동안 발생한 펄스 값을 가지고 값을 결정하는데 최초의 5스텝은 신경망의 입력에서 출력까지 신호가 전달되는 시간이므로 나머지 20스텝 동안의 발화 빈도가 출력 값이 된다. 그림 11은 이때 얻어진 XOR의 네트워크 구조를 나타낸다.

5. 결 론

본 논문에서는 진화 하드웨어를 이용하여 셀룰라 오토마타 신경망을 구현하였다. 제안된 시스템은 유전자 알고리즘 프로세서와 셀룰라 오토마타 신경망으로 구성된다. 유전자 알고리즘 프로세서는 신경망의 내부구성을 결정하는 구조적 bit를 개체로 간주하여, 유전자 알고리즘을 수행한다. 유전자 알고리즘 프로세서는 연산 모듈, 재생산 모듈, 제어기 그리고 개체군 및 개체군 및 적합도 값을 저장하는 메모리로 이루어져있다. 연산 모듈에서는 교차와 돌연변이를 수행하고, 재생산 모듈에서는 적합도 값에 따라 개체를 재생한 하고, 이를 이용하여 명령어를 만든다. 실행 시간을 줄이기 위해 재생산 개체들을 명령어화 하고, 교차와 돌연변이가 결합된 형태로 구성하였다. 셀계된 진화 하드웨어를 이용하여 셀룰라 오토마타 신경망을 구현하였다. 셀룰라 오토마타 신경망은 진화와 발생 모델을 기반으로 한 신경망 모델이다. 진화는 유전자 알고리즘 프로세서에 의해서 구현되며, 발생은 유전자 알고리즘 프로세서에 의해 진화 되는 개체들에 의해 자신의 내부 구조를 재구성하는 신경망에 의해 구현되었다.

본 논문에서는 진화형 하드웨어를 구현하여 생물 정보처리 시스템을 구현하였다. 즉, 자연계의 진화 및 발생의 원리를 이용한 시스템이 하드웨어로 구성될 수 있음 제시하고 그 유용성을 살펴보았다.

참 고 문 헌

[1] C.G. Langton ed, Artificial Life, Addison-Wesley, 1989.

[2] C.G. Langton, C Taylor, J.D. Farmer, S. Rasmussen ed, Artificial Life II, Addison-Wesley, 1992.

[3] C.G. Langton ed, Artificial Life III, Addison- Wesley, 1994.

[4] R.A. Brooks and P Maes ed, Artificial Life IV, The MIT Press, 1994.

[5] C.G. Langton and K. Shimohara ed, Artificial Life V, The MIT Press, 1997.

[6] E.J.W. Boers, H. Kuiper, B.L.M. Happel, and S. Kuyper, "Designing Modular Artificial Neural Networks," *Proceedings of Computer Science in the Netherlands*, pp. 87-96, 1993.

[7] Hugo de Garis, "CAM-BRAIN : The Genetic Programming of an Artificial Brain Which Grows/Evolves at Electronic Speeds in a Cellular Automata Machine," *Proceedings of The First International Conference on Evolutionary Computation*, vol. 1, pp. 337-339b, 1994.

[8] A. Stoica, D. Keymeuler, R. Tawel, C. Salazar-Lazaro, and W. Li, Evolutionary Experiments with a Fine-Grained Reconfigurable Architecture for Analog and Digital CMOS Circuits, Proceedings of the first NASA/DoD workshop on Evolvable Hardware, pp. 76-84, 1999.

[9] 이동욱, 심귀보, "셀룰라 오토마타 기법을 이용한 신경망의 자동설계에 관한 연구," *대한전자공학회 논문지*, 제 35권 S편, 제 11호, 1998.

[10] M. Ohta, A. Ogihara et. al., "A Study on The Mechanism of the Minimum Searching by the Chaotic Neural Network," *Proceedings of International Conference on Neural Networks*, pp. 1517-1520, 1995.

[11] S.H. Kim, G.W Jang et. al., "Trajectory Control of Robotic Manipulators using Chaotic Neural Networks," *Proceedings of International Conference on Neural Networks*, pp. 1685-1688, 1997.

[12] C.G. Langton, "Life at the Edge of Chaos," Artificial Life II, Addison-Wesley, pp. 41-91, 1992.

[13] M. Sipper, "Non-Uniform Cellular Automata : Evaluation in Rule Space and Formation of Complex Structures," *Artificial Life VI*, The MIT Press, pp. 394-399, 1994.

[14] A. Thompson, "An Evolved Circuit, Intrinsic in Silicon, Entwined with Physics," in Lecture Notes in Computer Sciencs, No. 1256, Evolvable Systems : From Biology to Hardware, Springer, 1997.

[15] X. Yao and T. Higuchi, "Promises and challenges of evolvable hardware," IEEE Transactions on Systems, Man & Cybernetics Part C: *Applications & Reviews 1997.*, vol. 29 pp. 87-97 no. 1, 1999.

[16] M. Mitchell and S. Forrest, "Genetic Algorithms and Artificial Life," *Artificial Life*, vol 1, no. 3, pp 267-289, 1994.

[17] S. D. Scott, A. Samal and S. Seth, "HGA : A Hardware based genetic algorithm," Proc. ACM/SIMDA 3rd International Symposium on FPGA, pp. 53-59, 1995.

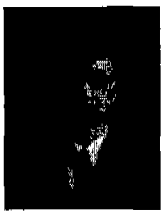
저 자 소 개

반창봉 (Chang-Bong Ban)

2000년 : 중앙대학교 전자전기공학부 공학사

2000년 ~ 현재 : 중앙대학교 대학원 전자전기공학부 석사과정

관심분야 : 진화하드웨어, 지능로봇 등



곽상영(Sang-Young Kwak)
1999년 : 단국대학교 전자공학과 공학사
2001년~현재 : 중앙대학교 대학원 전자전기공학부 석사과정

관심분야 : 진화하드웨어, 인공신경망, 지능로봇 등

이동욱(Dong-Wook Lee)
1996년 : 중앙대학교 제어계측공학과 공학사
1998년 : 중앙대학교 대학원 공학석사
2000년 : 중앙대학교 대학원 공학박사
2000년~현재 : 중앙대학교 대학원 박사 후 과정(연구원)

관심분야 : 인공생명, 인공두뇌, 인공면역계, 자율분산시스템, 진화하드웨어 등

심귀보(Kwee-Bo Sim)
1984년 : 중앙대학교 전자공학과 공학사
1986년 : 중앙대학교 대학원 전자공학과 공학석사
1989년 : The University of Tokyo 전자공학과 공학박사
1990년 : 동경대학 생산기술연구소 연구원
1991년~현재 : 중앙대학교 전자전기공학부 교수
1997년~현재 : 한국퍼지 및 지능시스템학회 편집이사 및 논문지 편집위원장
2000년~현재 : 제어·자동화·시스템공학회 논문지 편집위원 및 직선 평위원
2001년~현재 : 대한전기학회 제어 및 시스템 부문회 논문지 편집위원 및 학술이사

관심분야 : 인공생명, 진화연산, 지능로봇시스템, 뉴로-퍼지 및 소프트컴퓨팅, 자율분산시스템, 로봇비전, 진화하드웨어, 인공면역계 등

Tel : +82-2-820-5319
Fax : +82-2-817-0553
E-mail : kbsim@cau.ac.kr
URL : http://rics.cie.cau.ac.kr