

# 유전자 알고리즘 하드웨어 구현을 위한 전용 원칩 컴퓨터의 설계

박세현, 이연학, 박상필  
안동대학교 전자정보산업학부

## One-Chip Computer Design for Hard-Ware Implementation of Genetic Algorithm

Se-Hyun Park, Un-Hak Lee, Sang-Pil Park  
Electronic & Information Industrial Engineering, Andong National University

### 요 약

유전자 알고리즘을 구현하기 위해서 전용 원칩 컴퓨터를 설계하였다. 유전자 알고리즘의 전용 원칩 컴퓨터는 16Bit CPU CORE와 유전자 알고리즘의 하드웨어로 구성되어 있다. 구현된 전용 원칩 컴퓨터는 기존의 하드웨어 GAP와 달리 메인 컴퓨터에 독립적으로 동작되며 멀티미디어 통신에 사용되는 비트 동기용 하드웨어를 생성시켜본 결과 효과적임을 알 수 있었다.

### 1. 서론

유전자 알고리즘(GA:Genetic Algorithm)은 생물의 진화 원리로부터 착상된 확률적 탐색을 위한 기법이며 1975년에 소개된 홀랜드(Holland)의 저서 『Adaption in Natural and Artificial System』에 처음으로 소개되었다<sup>[1]</sup>. 1985년부터 개시된 유전자 알고리즘에 관한 국제회의(ICGA)가 개최된 이후 1990년부터 유전자 알고리즘에 대한 관심이 높아지기 시작하여 최근에는 패턴인식, 신경망 학습, 음성인식, 로봇 제어 등의 분야에서 사용되고 있다. 유전자 알고리즘은 연산의 반복 수행으로 인한 속도의 저하가 심각한 문제가 되어왔으며 그에 대한 해결방안으로 병렬 분산 처리, 하드웨어 구현 등 여러 방법이 시도되었다.

유전자 알고리즘의 하드웨어의 구현은 Scott 등에 의해 제안된 간단한 유전자 알고리즘(Standard simple genetic algorithm)에 기초한 Hardware-based Genetic Algorithm(이하 HGA)<sup>[2]</sup>가 그 시초이며 Yoshida 등은 이를 발전시켜 GAP(GA Processor)라는 하드웨어 지향의 GA를 제안하였다.<sup>[3]</sup> 또한 Wakabayashi 등은 범용 하드웨어로서 GAA(GA Accelerator) Chip이라 불리는 GA의 VLSI를 구현하였으나<sup>[4]</sup> 최근에는 Reprogramming이 가능한 FPGA를 사용한 구현이 국내·외 학계에서 연구되고

있다.<sup>[5]</sup>

기존의 HGA는 주로 유전자 알고리즘의 수행 속도 개선을 위해 유전자 알고리즘을 FPGA등의 하드웨어로 구현 하든가 상용 프로세서를 이용하여 펌웨어로 구현하였다. 이러한 HGA는 GA를 위한 각종 파라미터 등을 설정하거나 염색체의 적합도를 구하기 위해서 메인 컴퓨터에 의존하게 되어있다. HGA가 메인 컴퓨터를 의존하게 되면 특정 분야에 대한 GA의 활용도가 떨어지게 된다. 특히 Evolvable Hardware(이하 EHW: Evolvable Hardware) 분야에 에 GA를 적용하기 위해서는 HGA가 메인 컴퓨터에 독립적이어야 할 필요가 있다.

또한 기존의 HGA에 문제점은 염색체 길이가 32비트 혹은 64 비트이어서 고정 길이의 염색체에 대한 GA처리를 하였다. 그러나 이것을 실제의 GA 응용분야에 적용하는 데는 매우 적은 크기이며 효용가치가 없다.

따라서 본 연구에서는 GA의 하드웨어 구현에 있어서 HGA의 하드웨어와 HGA에서 GA를 위한 전용 원칩 컴퓨터와 인터페이스 한 시스템을 제안한다. 제안된 전용 원칩 컴퓨터는 기존의 하드웨어 HGA와 달리 대규모 길이의 염색체를 16비트 단위로 전과정을 Pipeline처리를 할 뿐 아니라 메인 컴퓨터에 독립적으

로 동작하게 한다. 그리고 제안된 시스템을 EHW에 적용하고 그 효용성을 알아본다.

## 2. 본론

### 2.1 기본이론

유전자 알고리즘은 기본적으로 Generate-and-Test 형의 알고리즘으로서, Darwin의 적자 생존 이론을 기초로 한다. 즉, 문제에 대한 가능한 해들을 정해진 형태의 자료구조로 표현한 선택(selection), 교차(crossover), 돌연변이(mutation)의 세 가지 종류의 유전자 조작으로 이루어진 재생산(Reproductio) 과정을 반복 수행하여 개체내의 정보를 교환함으로써 최적의 해를 생산하는 알고리즘이다.

일반적으로 개체집단  $P(t) = \{x^t_1, \dots, x^t_n\}$ 에서 유전자 알고리즘의 처리순서는 아래와 같다.

```

genetic algorithm
begin
  t ← 0
  initialize P(t)
  evaluate P(t)
  while (not termination-condition) do
    begin
      t ← t + 1
      select P(t) from P(t-1)
      crossover P(t)
      mutation P(t)
    end
  end
end
    
```

### 2.2 유전자 알고리즘을 위한 전용 원칩 컴퓨터의 설계와 유전자 하드웨어

본 연구에서는 유전자 알고리즘을 하드웨어로 구성하고 이 유전자 알고리즘의 하드웨어를 인터페이스할 전용 원칩 컴퓨터를 설계한다. 설계된 전용 원칩 컴퓨터는 외부 메모리로부터 염색체의 LOAD와 STORE를 위해 DMA를 사용한다. 설계된 원칩 컴퓨터는 내부에 교차부, 돌연변이부, 선택부로 구성되어 있으며 염색체 보관은 외부 메모리에 저장한다. 전용 원칩 컴퓨터의 구조를 교차부, 돌연변이부 및 선택부 순으로 설명한다.

#### (1) 전용 원칩 컴퓨터

구현된 CPU CORE는 그림 1과 같은 기본 구조를

가지고 있으며 Altera사의 FLEX10K100ARC를 사용하였다. 원칩 컴퓨터는 내부 16 비트의 버스를 가지고 있고 내부에 프로그램 메모리와 데이터 메모리를 가진다. 염색체의 저장을 위한 메모리는 외부에 두고 있으며 DMA에 의해 구동시킨다. 실제의 GA 응용분야는 염색체의 길이가 길다. 특히 EHW 분야에 GA를 적용하려면 염색체의 표현에 있어서 대단히 많은 비트가 필요하다. 따라서 대규모 크기의 염색체를 병렬 처리 할 수 있는 교차 부와 돌연변이 부의 하드웨어를 설계한다는 것은 엄청난 하드웨어 자원이 요구되므로 비현실적이다. 기존의 HGA에 있어서 염색체의 처리는 32비트 혹은 64 비트의 처리를 하였다. 그러나 이것을 실제의 GA 응용분야에 적용하는 데는 매우 적은 크기이므로 효용가치가 없다. 따라서 본 연구에서는 염색체를 외부 대용량 메모리에 두고 16비트 단위로 DMA로 CPU에 이동시켜 Pipeline 처리를 한다. 제안된 전용 CPU CORE의 목적은 GA의 처리 속도의 향상은 물론 HGA를 실제 GA 응용분야 특히 EHW 분야에 적용할 수 있도록 메인 컴퓨터에 독립적으로 동작할 수 있게 하는 것이다.

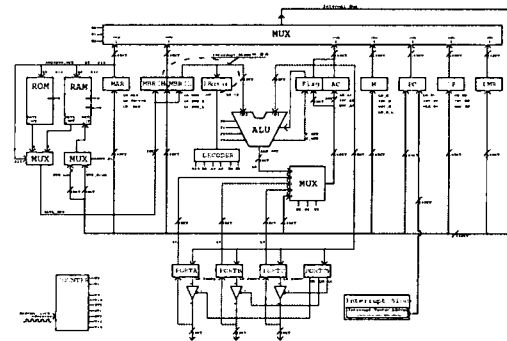


그림 1. 16Bit CPU Core 블록도

설계된 CPU는 소프트웨어(내부) 인터럽트와 하드웨어(외부) 인터럽트의 구조를 가지고 있으며 각각 4개의 소스를 받아들일 수 있다. 어드레스 벡터는 각 염색체의 16비트 단위의 DMA 이동에 있어 인터럽트 처리를 이용할 수 있다. 그림 2는 인터럽트의 하드웨어 구조를 보여주고 있다.

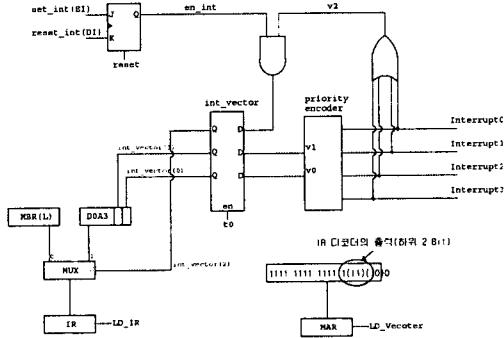


그림 2. 인터럽트 하드웨어 구조

(2) 교차부(crossover part)

교차는 두 부모의 염색체를 조합하여 바꾼 후 자손의 염색체를 만드는 조작이다. 즉, 일정한 규칙으로 염색체를 교환함으로써 부모의 형질을 적절히 조합하여 새로운 개체를 생성하는 것을 말한다. 이 교차는 하나의 교차위치를 가지는 일점교차, 두 개이상의 교차 위치를 가지는 두점교차, 균일교차 등의 방법이 있다.

그림 2는 구현된 GA의 교차부의 블록도이다.

Scott의 HGA, Yoshida의 GAP 그리고 Wakabayashi의 GAA Chip에서는 일점교차만을 사용하고 있으나 본 연구에서는 그림 3에 보는 바와 같이 Decoder 값에 따라 일점교차, 두점교차, 균일교차 등 필요에 따라 선택할 수 있도록 설계하였다. 그리고 교차를 또한 조절이 가능하도록 설계하였다. 또한 염색체 16비트 단위로 1 Clock에서 교차가 이루어지는 구조로 설계되었다.

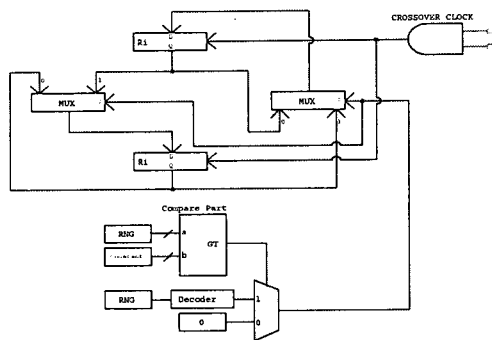


그림 3. 교차부(crossover part)

(3) 돌연변이부(mutation part)

돌연변이는 유전자를 일정한 확률로 변화시키는

조작이다. 돌연변이를 너무 큰 확률로 설정하면 부모의 유전자 형질이 변경될 수 있지만 국소 탐색에 빠지지 않도록 하는 조작이다.

그림 4는 본 연구에서 설계한 돌연변이부에 대한 블록도이다. 돌연변이부도 교차부와 같은 기능의 비교부(compare part)를 사용하여 돌연변이율의 조절을 가능하게 하였고 Decoder의 조절로 돌연변이될 Bit의 선택도 자유롭게 하였다. 돌연변이 속도 역시 교차부와 마찬가지로 염색체의 16비트 단위로 1 Clock에서 돌연변이가 이루어지는 구조로 설계되었다.

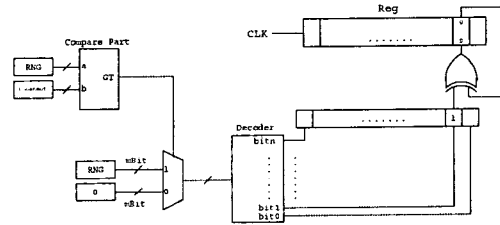


그림 4. 돌연변이부(selection part)

(3) 선택부(selection part)와 적합도 평가부(fitness evaluation part)

선택은 개체군에서 개체를 주어진 규칙에 따라 다음 세대(자식 세대)에 생존할 개체를 확률적으로 결정하는 것을 말한다.

본 연구에서는 하드웨어 구현에 적합한 정상상태 GA(steady-state GA)를 토너먼트 선택(tournament selection)과 함께 사용한다.

정상상태 GA(Steady-State GA)는 열성의 부모를 우성의 자손으로 대체하는 것이다. 이것은 세대모텔(Generational GA)에 비해 메모리 사용량을 줄일 수 있다. 토너먼트 선택은 GA에서 일반적인 방법으로 사용되는 룰렛 선택(Roulette wheel selection)에 비해 하드웨어 자원을 줄인다는 면과 처리 속도에서 우수하다.

그림 5는 제안한 선택부(selection part)와 적합도 평가부(fitness evaluation part)이다. DMA로 부터 전송한 16비트의 4개의 염색체를 토너먼트로 선택하여 2개의 우성 염색체를 교차와 돌연변이 하게된다. 16 비트 비트를 단위 처리로 하여 전 염색체를 처리하게 되는데 교차와 돌연변이 그리고 적합도 평가부에 이르기까지 Pipeline 처리를 하게 되어 있다. 16비트의 단위처리는 32비트 이상의 처리 보다 속도 면에서 크게 뒤지지 않는다. 왜냐하면 HGA에서 속도저하

를 가져오는 요인은 교차와 돌연변이 로직에 있지 않고 적합도를 평가하는 평가부에서 있기 때문이다. 본 연구의 적합도 평가부는 2개의 검색체를 동시에 처리할 수 있게 이중 구조를 지니고 Pipeline 처리하여 DMA에 의해 외부 메모리에 저장시킨다.

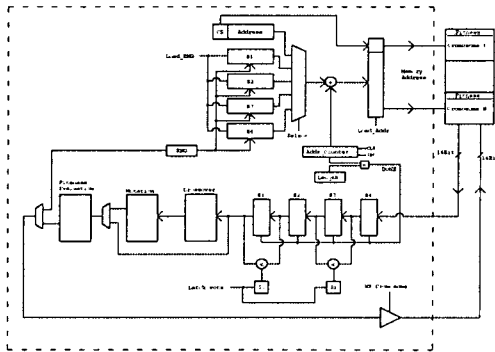


그림 5. DMA를 사용한 선택부와 적합도 평가부

### 3. 실험 및 고찰

유전자 알고리즘의 전용 원칩 컴퓨터는 EPF10K100A와 SRAM으로 설계하였다. 설계된 전용 원칩 컴퓨터를 사용하여 하드웨어 진화에 적용하여 보았다. EHW의 대상으로 비트 동기 추적 하드웨어를 진화시켜 보았다.

그림 6은 유전자 알고리즘을 적용할 수 있도록 범용 state machine을 ROM으로 구성하였다. 그러나 평가시에는 ROM은 GAL, PLA와 달리 전용 장치의 도움이 없더라도 검색체의 읽기와 쓰기가 쉬운 RAM으로 설계하였다.

그림 6의 범용 state machine에서 ROM의 내용은 그림 7과 같은 유전자 알고리즘을 적용시킨 검색체가 저장된다. ROM의 주소입력은 입력 X와 함께 ROM의 출력 Q0 - Q7이 피드백되어 플립플롭을 거쳐 입력된다. 그리고 ROM의 주소 입력에 의해 출력 Q0 - Q8이 출력된다.

입력 X는 NRZ 랜덤 비트 신호를 입력시키는 단자이며 출력 Y는 NRZ 랜덤 신호에 동기되어 입력 비트 구간의 중간을 샘플링하게 하며 제안된 시스템으로 학습시킨다.

집단수는 50개, 검색체 비트수는 188개이다. 그리고 돌연변이는 2% 교차는 80% 이다.

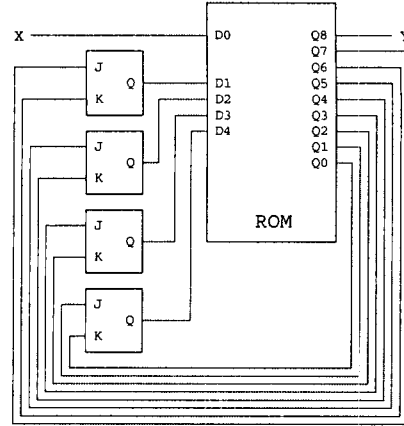


그림 6. 적합도 평가 회로

그림 7은 비트 동기 추적 시스템을 구현하기 위해 유전자 알고리즘을 적용하여 출력한 적합도 99.88%의 검색체이다. 출력된 검색체는 비트 동기 추적 시스템을 위해 사용되며 ROM에 들어갈 데이터가 된다. 이 데이터는 ROM의 주소와 데이터 및 출력신호를 포함하고 있다.

```

0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1,
0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1,
0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1,
1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0,
1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1,
0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0,
1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0,
0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0,
1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1,
0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0,
1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 0,
1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0,
1, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0,
1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1,
1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1,
1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0,
0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1,
    
```

그림 7. 유전자 알고리즘에 의해 출력된 검색체

그림 8은 제안된 시스템을 사용하여 비트 동기 추적 하드웨어의 세대수와 적합도의 관계를 그래프로 표현한 것이다. 유전자 알고리즘의 처리순서는 초기집단이 생성된 후 종료 조건이 만족될 때까지 적합도 평가, 선택, 교차, 돌연변이를 반복 수행한다. 최대 적합도 2500을 기준으로 세대수 25000에 따른 적합도와 평균치의 변화를 그림 8에서 알 수 있다. 유전자 알고리즘을 시작시켜 첫 세대에서 나온 적합도는 1883(75.32%)이며 세대수 1000에서 적합도가 급속하

계 올라가는 것을 볼 수 있으며 그때의 적합도는 2227(89.08%)이다. 그리고 세대수 5000에서 적합도는 2229(89.16%)이고 세대수 10000에서 적합도는 2436(97.44%), 세대수 15000에서는 2494(99.76%), 세대수 19000부터 25000까지는 적합도가 2497(99.88%)이 계속해서 나온다. 제안된 시스템에서 비트 동기 추적 하드웨어를 진화 시켜본 결과 최대 적합도가 99.88%임을 알 수 있었다.

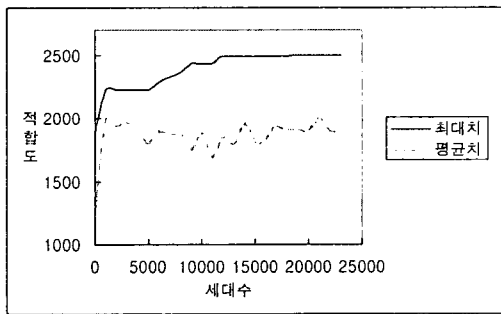


그림 8. 적합도와 세대수와의 관계

그림 9는 유전자 알고리즘을 적용한 비트 동기 추적 하드웨어에서 입력 X에 NRZ 랜덤 신호를 입력하고 출력 Y로 비트 동기 추적 신호를 샘플링 한 모습을 나타낸 것이다. 동기된 출력 신호는 NRZ의 입력 비트 구간의 중앙을 정확히 샘플링하는 것을 알 수 있다.

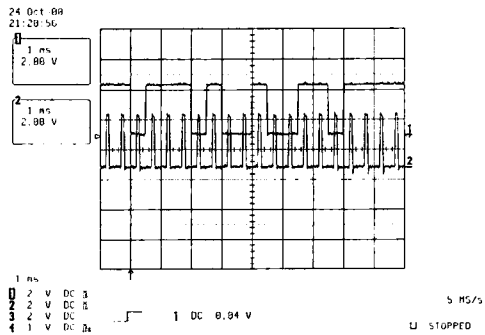


그림 9. NRZ 신호에 의한 비트 동기 과정

#### 4. 결론

유전자 알고리즘을 구현하기 위해서 전용 원칩 컴퓨터를 설계하였다. 구현된 유전자 알고리즘의 전용 원칩 컴퓨터는 16Bit CPU CORE로 되어 있고 16 비트의 내부 버스와 인터럽트, DMA 및 HGA 등의 기능이 있다.

제안된 시스템은 주어진 염색체를 16 비트 단위로 단위 처리하며 교차와 돌연변이 그리고 적합도 평가부에 이르기까지 전과정이 Pipeline 처리를 하게 되어 있다. HGA에서 속도저하를 가져오는 요인은 교차와 돌연변이 로직에 있지 않고 적합도 평가부에서 있기 때문에 16비트의 단위처리는 32비트 이상의 처리보다 속도 면에서 크게 뒤지지 않는다. 따라서 제안된 시스템은 염색체 길이에 무관하게 대규모의 크기의 염색체를 처리할 수 있으므로 GA의 실제 적용에 효과적이다.

설계된 전용 원칩 컴퓨터는 기존의 하드웨어 GAP와 달리 메인 컴퓨터에 독립적으로 동작된다.

설계된 시스템을 사용하여 하드웨어 진화( EHW: Evolvable Hardware)에 적용하여 비트 동기 추적 하드웨어를 진화시켜 보았다. 제안된 시스템에서 비트 동기 추적 하드웨어를 진화 시켜본 결과 최대 적합도가 99.88%임을 알 수 있었다.

#### [참고문헌]

- [1] J. Holland, "Adaptation in Natural and Artificial System", The University of Michigan, 1975. , and MIT Press, 1992.
- [2] S.D. Scott, A. Samal and S. Seth, "HGA:A hardware-based genetic algorithm", Proc. ACM/SIMDA 3rd International Symp. on FPGA, pp.53-59, 1995
- [3] N. Yosida, T. Moriki and T. Yasuoka, "GAP:Genetic VLSI processor for genetic algorithm", 1Second International ICSC Symp. on Soft Computing, pp.341-345, 1997
- [4] Shin'ichi Wakabayashi et al., "GAA:A VLSI genetic algorithm accelerator with on-the-fly adaptation of crossover operators", ISCAS 98, 1998
- [5] Jin Jung Kim, Duck Jin Chung, "Implementation of Genetic Algorithm based on Hardware Optimization", TENCON '99 1999;