

진화적응을 위한 유전알고리즘의 하드웨어 구현

*동성수

용인송담대학 정보미디어학부

e-mail : *ssdong@ysc.ac.kr*

A Hardware Implementation of Simple Genetic Algorithm for Evolvable System

*Sung-Soo Dong

School of Information and Media

Yonginsongdam College

Abstract

This paper presents the hardware-based genetic algorithm, written in VHDL. Due to parallel computation and no function call overhead, a hardware-based GA advantage a speedup over a software-based GA. The proposed architecture is constructed on a field-programmable gate arrays, which are easily reconfigured. Since a general-purpose GA requires that the fitness function be easily changed, the hardware implementation must exploit the reprogrammability.

I. 서론

유전알고리즘(genetic algorithm)은 우수한 인자가 살아남는 자연의 적응생태계 법칙을 기반으로 하는 최적화(optimization) 기법이다[1]. 기존의 최적화 방법과 비교해서 다음의 장점이 있다. 최적화 문제의 파라메타 집합을 코딩된 문자열을 사용하므로 파라메타 영역의 연속성이나 미분가능의 제약이 없고, 한 점이 아니라 점들의 집단을 탐색함으로써 잘못된 최고점을 찾을 확률이 준다. 또한 다른 보조적인 정보가 필요치 않고

문자열과 관련된 목적함수 값만 필요로 하며 효율적 탐색 방향을 위해 확률적 천이 규칙을 사용한다. 이러한 장점들로 인해 유전알고리즘은 강인하고 강력하다 [2]. 따라서 전력 시스템 제어, 고장 검출, 제어 시스템, 신호처리 등 많은 분야에서 적용된다[3]. 그러나 소프트웨어로 구현된 유전알고리즘은 많은 NP-complete 문제에서 증가하는 계산량 때문에 속도가 느려질 수밖에 없다. 그러므로 하드웨어로 구현하여 병렬특성을 살리면 속도 면에서 이득이 있고 연산엔진으로써 활용될 수 있다.

본 논문에서는 기본적인 구조의 유전알고리즘을 하드웨어로 구현하여 연산엔진으로 활용 할 수 있는 바탕을 마련한다. 그리고 응용문제마다 목적함수가 다르므로 쉽게 변경하기 위하여 FPGA로 구현한다.

II. 본론

2.1 유전알고리즘

자연환경에서 유전자는 생물의 특징을 나타내고, 다음 세대에 우성인자를 유전시킨다. 유전알고리즘에서 유전자 또는 개체(individual)는 염색체(chromosome)들의 집합이며, 여러 개의 개체들이 모인 집단을 군집(population)이라 한다. 염색체 표현 방법(encoding scheme)에는 여러 가지가 있지만 기본적인 유전알고

리즘(simple genetic algorithm)에서는 보통 이진문자열로 표현한다. 각 개체가 환경에 얼마나 잘 적응하는지를 나타내는 값이 적합도(fitness)이며, 문제를 정의하는 환경을 적합도 함수 또는 목적 함수(objective function)라 한다[1, 2].

2.2 동작 원리와 흐름

유전알고리즘의 수행과정은, 먼저 목적함수를 선정하고 다음 초기 유전자 집단을 랜덤하게 생성한 후, 각 개체에 대해 목적함수에 따른 적합도를 계산한다. 계산된 적합도를 근거로 다음 세대를 생산하는 재생산(reproduction)과정을 거친다. 이 때 선택(selection), 교배(crossover), 돌연변이(mutation) 연산자(operators)를 사용하여 보다 좋은 해를 가지는 개체의 발생을 기대한다. 새로운 개체에 대해 적합도를 계산하여 평가한 후 결과에 만족하면 종료하고, 아니면 재생산 단계로 다시 돌아가서 반복한다[4].

III. 구현

구조는 다음의 모듈들로 구성되었다. 모든 파라메타 및 결과를 공유하는 메모리와 상호연결 하는 모듈(mem_io), 연산자 모듈들에 사용할 무작위 수를 발생시키는 모듈(random_gen), 집단의 순서를 결정해주는 모듈(sequencer), 룰렛 휠 방법을 기초로 하는 선택 모듈(selection), 교배 및 돌연변이 연산 모듈(xover_mut), 새로운 집단의 개체를 생성하기 위한 적합도 모듈(fitness)로 구성되어 있다[5]. 각 모듈은 VHDL로 코딩을 하여 Xilinx xc3s200-4ft256 FPGA에 구현하였다. 개체는 4, 적합도는 5, 집단의 크기는 32 비트, 세대수는 1024로 하여 설계한 결과에 대해 그림 1에 FPGA 이용률(device utilization)을 나타내었다.

	# of slice flip-flop	# of 4 input LUTs	equivalent gate count	maximum frequency (MHz)	# of slice (routing)
sequencer	31	61	665	121.095	36
random_gen	47	55	757	307.125	34
selection	106	233	6459	68.376	127
xover_mut	38	92	868	128.916	51
fitness	147	395	3801	112.994	222
mem_io	57	138	1341	134.825	79

그림 1. FPGA 이용률(device utilization)

그림 2에는 합성된 개략도(synthesized schematic)를 나타내었다. 여기에서는 파라메타들과 결과를 저장하

는 공유메모리가 제외된 상태의 그림이다.

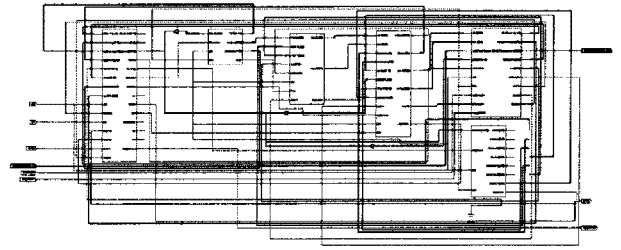


그림 2. 합성된 구성도

IV. 결론 및 향후 연구 방향

최적화 기법에 사용되는 유전알고리즘을 하드웨어로 구현하였다. 이는 소프트웨어로 작성된 알고리즘을 이용하여 문제를 해결하는 경우보다 연산의 수가 매우 많은 경우 속도에 장점이 있기 때문이다. 그리고 예측 가능한 상황에서 동작하는 기존의 하드웨어 시스템 보다, 예측하지 못한 상황의 변화에서도 대응하여 문제를 해결할 수 있는 시스템인 적응하드웨어에서 유전알고리즘은 유용하게 쓰이고 하드웨어 엔진으로서의 역할이 중요한 부분을 담당할 수 있다[4]. 앞으로 구현된 모듈을 여러 응용에 적용하고, 좀 더 발전적인 구조로 변경하여 효율적인 진화적응 하드웨어의 역할을 수행할 수 있도록 하는 부분에 목적을 둘 것이다.

참고문헌

- [1] David E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Publishing Company, 1989.
- [2] 오성권 저, *프로그래밍에 의한 컴퓨터지능*, 내하출판사, 2002.
- [3] A. M. S. Zalzala, Ed., *Proceedings of the First IEE/IEEE International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications*, IEE, 1995.
- [4] Greenwood, Tyrrell, *Introduction to Evolvable Hardware*, Wiley-Interscience, 2007.
- [5] Scott D, Seth S and Samal A, "A Hardware Engine for Genetic Algorithms", Technical Report UNL-CSE-97-001, 1997.