

적응생존형 네트워크 프로세서의 생존성 향상을 위한 유전 알고리즘의 이용

원주호, 윤홍일

연세대학교 전기전자공학부

Genetic Algorithm for Improving the survivability of Self-Adaptive Network Processor

Joo-Ho Won and Hongil Yoon

Dept. of Electrical and Electronic Engineering, Yonsei University

Abstract

공정기술의 발달과 컴퓨터 구조적인 발전에 의해서, 시스템의 동작속도가 기하급수적으로 증가하고 있다. 동작속도의 증가는 CMOS로 구현된 chip의 RC 특성에 의해서 timing variation 문제가 발생할 가능성이 높아지면서 테스트 비용이 전체 설계비용에서 차지하게 되는 비중이 급격하게 증가하고 있다. 따라서 온라인 테스트와 진화하드웨어 등이 테스트 비용감소를 위해서 연구되고 있다. 본 논문에서는 네트워크 프로세서의 생존성을 위해서, 패킷엔진의 pipeline의 각 stage사이의 clock slack borrowing을 이용해서 timing variation 문제를 자체적으로 해결할 수 있다는 것을 mixed-mode simulation을 통해서 통합 검증하였다. 또한 기존의 off-chip 진화하드웨어에 비해서 on-chip 구현을 통해서 진화하드웨어의 성능향상과 메모리에 의해서 발생하는 overhead를 감소시키는 것이 가능함을 확인했다.

Key Words : 유전알고리즘, 생존하드웨어, 네트워크프로세서, 테스트

1. 서론

반도체 공정기술의 발달과 구조적인 발달에 의해서, 1GHz를 초월하는 시스템의 구현이 가능하게 되었다. 하지만, 이러한 동작속도의 증가는 과거에는 고려하지 않았던, timing variation에 의해서 발생하는 문제를 심각하게 만드는 부작용을 초래하게 되었다. 일반적으로 jitter나 skew와 같은 timing variation에 대한 문제에 대한 해결책은 chip의 제작이전의 설계단계에서 모든 variation을 없애는 방법이 존재했다. 하지만, 이러한 방법은 디자인과정의 특별한 환경조건에서만 해결책으로 동작환경에서 발생하는 문제에 대해서는 해결

책이 될 수가 없다. 따라서, 다양한 환경 및 동작 변화에 의해서 발생하는 timing variation에 대해서 실시간으로 해결할 수 있는 해결책이 필요하다.

이러한 해결책으로 FPGA와 같은 프로그램 가능한 소자와 유전알고리즘을 이용해서 환경이나 동작조건 변화에 자체적으로 적응해서 하드웨어의 기능이나 성능을 변화시키는 방식 [1] 및 off-chip으로 구현된 진화하드웨어를 이용해서, timing variation문제를 해결하고자 하는 방식 [2] 이 개발되었다.

본 논문에서는 on-chip 진화하드웨어를 네트워크 프로세서의 핵심유닛에 해당하는 패킷엔진에 적용해서, 실시간으로 clock의 slack borrowing을 통

한 생존성 향상을 확인하고, 기존의 off-chip진화하드웨어와 비교해서 on-chip구현을 통한 문제해결 시간의 단축을 mixed-mode simulation을 통한 검증을 실시하였다.

2. 이론

2.1. 적응생존형 네트워크프로세서

네트워크프로세서는 패킷의 포워딩, 스위칭을 위해서 만들어진 네트워크 전용 프로세서로, 라우터, 랜카드 등에 적용이 가능하지만, 전체 네트워크의 생존성을 결정하는 중요한 유닛에 해당된다 [3]. 네트워크프로세서는 General Purpose Processor (GPP)의 유연성과 Application Specific Integrated Circuit (ASIC)이 가지는 속도의 장점을 동시에 갖게 되는 시스템으로 본 연구에서 사용한 적응생존형 네트워크프로세서는 패킷에 대한 업무를 담당하는 패킷엔진과 패킷의 중간정보를 저장하는 메모리, 네트워크 트래픽 조건에 적합하게 패킷엔진의 동작을 변화시키기 위한 스티그머지엔진과 전체 block의 동작을 control하게 되는 controller로 구성이 된다.

2.2. On-Chip 진화하드웨어

기존의 off-chip 진화하드웨어는 FPGA와 같은 programmable device와 유전알고리즘을 구현하는 유닛이 DSP나 PC에 프로그램의 형태로 구현이 되어서, 통신채널의 한계에 따른 속도 제한을 받을 수밖에 없었다[2]. 따라서 on-chip 진화하드웨어의 구현을 위해서, chip내부에 PLL과 같은 시스템에서 critical path의 model에서 사용하게 되는 Fanout-4 (FO4) inverter chain과 이진위상검출기를 이용해서, 진화하드웨어의 evaluation block으로 구현하고, 그림 1에서와 같이 구성된다 [4].

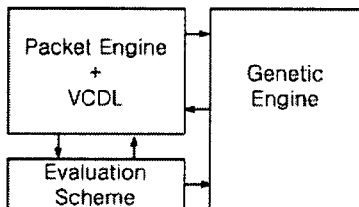


그림 1. 패킷엔진에 적용된 On-chip 진화하드웨어의 구조

2.3. 유전알고리즘

유전알고리즘은 최적화 문제의 집합을 유한 길이의 이진수열로 변환하여 처리함으로써 기존의 greedy algorithm과 같은 알고리즘이 경사 하강법 (gradient decent method)을 사용하는 것과는 달리 지역극소에 빠지지 않는다는 것과 효율적으로 최적화해를 찾기 위해서 교배연산과 돌연변이 연산을 이용한다는 것이 큰 장점이다 [5]. 일반적인 유전알고리즘에서는 다음과 같은 적합도함수를 사용하게 된다.

$$fitness = \sum_{i=1}^{\#samples} (V_{target} - V_{out})^2 \quad (1)$$

적합도 함수를 이용하는 진화하드웨어 [6]를 구현하기 위해서는 모든 테스트 패턴을 저장하고, 해당 테스트패턴의 정상동작을 저장하는 공간이 필요하기 때문에, on-chip 구현의 제한으로 작용하게 된다. 따라서 본 논문에서는 적합도함수를 각 위치의 정상, 비정상을 표현하는 1-bit 이진수만으로 축소해서 사용하였다. 이러한 축소된 적합도함수를 이용하기 위해서 유전알고리즘의 교배연산과 돌연변이 연산도 축소된 개념으로 변형되었다.

전체 유전자는 25bit의 이진수로 구성하였고, 25bit는 5bit로 구성되는 5개의 block으로 나뉘지고, 교배연산은 2개의 부모유전자사이에서 1개의 block에 대해서만 이루어지고, 돌연변이는 각각의 block에 한정되어서 up-down counter를 이용하였다. On-chip 진화하드웨어에 사용된 진화연산을 그림2에 나타내었다.

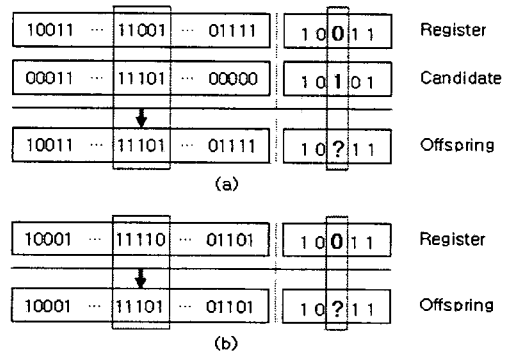


그림 2. On-chip 진화하드웨어에 사용된 (a)교배연산과 (b)돌연변이연산

3. 실험 및 구현

3.1. 실험계획

패킷엔진의 pipeline은 fetch, decode, execute, register write, write back 의 5-stage로서, system clock에 동기 되어서 동작한다. 따라서 133 MHz의 시스템은 각 pipeline stage가 대략 7.5ns동안 주어진 기능을 완료해야 한다. 이러한 상황에서 환경에 의한 고장을 모델링하기 위해서 clock의 속도를 153MHz로 증가시킬 경우, pipeline의 모든 stage에서 고장이 발생하게 된다.

3.2. 구현 및 검증

진화하드웨어의 on-chip구현을 위해서, 유전알고리즘을 담당하는 Genetic Engine (GE)을 C++를 이용해서 알고리즘의 동작을 확인하고 Verilog를 이용해서 behavioral level model (BLM)로 모델링해서 검증하였다. 또한 하드웨어수준의 구현 가능성을 검증하기 위해서 BLM을 0.35um 공정을 사용한 synthesis를 통해 gate-level model (GLM)로 동작을 검증하였다. 본 논문에서는 이렇게 timing variation이나 clock의 동작속도의 변화에 의해서 발생하는 문제에 대해서, on-chip 진화하드웨어를 이용해서, 문제를 해결해나가는 과정을 schematic으로 구현된 evaluation block과 BLM과 GLM으로 구현된 GE를 통합 검증하였다. Simulation을 위해서 Mentor Graphic의 Advanced MS를 사용하였다.

4. 결과 및 고찰

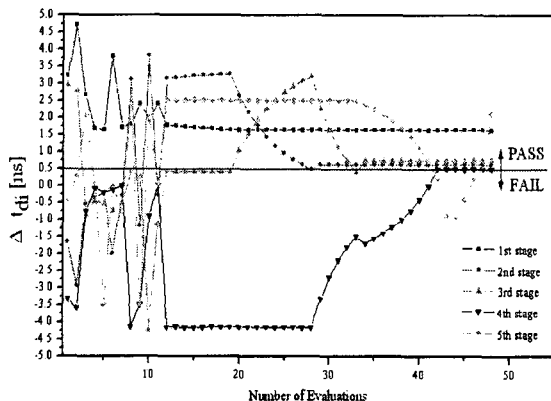


그림 3. 진화하드웨어의 고장처리과정 통합시물레이션결과

Pipeline의 각 stage가 정상적으로 동작하기 위해서는 inter-stage register의 setup time이전에 모든 동작을 완료해야한다.

그림 3에서는 진화하드웨어가 세대가 진행되면서, 각 세대에서의 각 stage의 setup time을 보여주고 있다. 유전알고리즘의 교배연산과 돌연변이연산을 이용해서, 모든 stage가 setup time 기준인 0.5ns를 만족하는 조건으로 진화함을 확인할 수가 있다.

유전알고리즘은 확률적 천이규칙을 사용하게 되는 알고리즘으로, 난수발생기의 난수특성에 따라서 전체적인 영향을 받게 된다. C++으로 구현된 GE는 우수한 난수특성을 갖게 되므로, 유전알고리즘의 본래적인 특성에 따라 다양한 결과를 얻게 되지만, HDL로 구현된 GE는 난수발생기로 사용한 Linear Feedback Shift Register (LFSR)가 주기적인 난수특성을 갖게 되므로 항상 46평가 만에 최적의 조건을 찾아내게 된다.

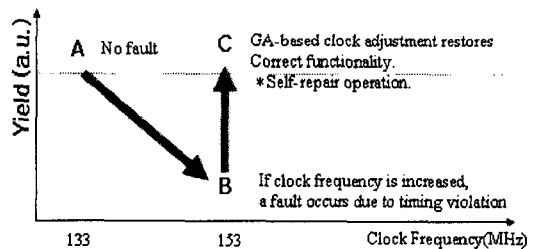


그림 4. 패킷엔진에 적용된 진화하드웨어의 고장모델과 고장처리과정

이 연구에서 적용된 진화하드웨어의 전체적인 고장처리과정을 그림 4에 나타내었다. 133MHz에서 정상 동작하는 (A) 시스템에 고장을 모델링하기 위해서 153MHz으로 동작하게 만들면, 모든 stage에서 고장이 발생하고 (B), 이러한 고장상태를 진화하드웨어를 통해서 정상상태 (C)로 자체적으로 치유할 수가 있는 것이다.

기존의 off-chip 진화하드웨어를 이용해서 timing variation 문제를 해결하는 방식은 최적의 조건을 찾아내는데 9분이 소요되었지만, 대부분의 시간이 programmable device와 유전알고리즘을 구현한 블록의 통신에 소요되었다 [2]. 하지만 본 논문에서 제안한 on-chip 진화하드웨어는 최적의 조건을 찾는 시간을 620us로 단축시켰다.

진화하드웨어의 성능을 더욱 발전시키기 위해서는 본 논문에서 제안한 evaluation scheme에서 적합함수의 부호뿐만이 아니라, 절대값을 이용함으로써, 진화하드웨어의 효율성을 증대시킬 수 있는 새로운 알고리즘을 연구하고, 제안된 scheme을 통해서 해결할 수 있는 timing variation의 폭을 확대하기 위해서는 Voltage Controlled Delay Line (VCDL)에서 발생시킬 수 있는 딜레이의 최소값과 최대값의 차이를 늘리고, VCDL에서 발생하는 필요없는 overhead를 감소시키기 위해서는 VCDL의 최소값을 낮추는 것이 필요하다.

5. 결론

네트워크프로세서의 패킷엔진에 적용된 진화하드웨어를 통해서, pipeline에서 각 stage사이의 clock slack borrowing을 이용해, 고속의 동작을 하는 시스템에서 발생가능성이 높은 timing variation문제를 자체적으로 해결할 수 있다는 것을 mixed-mode simulation을 통해서 검증하였다. 지금까지 evaluation block의 복잡성에 의해서 DSP또는 PC에 의해서 구현된 off-chip 진화하드웨어에 비해서, 성능평가를 위해 PLL등에서 사용되는 FO4 inverter chain을 이용해서 evaluation block을 on-chip화 시켜서, 최적의 유전자를 찾는 시간을 획기적으로 줄이는 것이 가능하고 성능을 평가에 사용하게 되는 test pattern을 저장하기위해 사용되는 memory의 overhead를 줄이는 것이 가능하다.

감사의 글

본 연구는 IDEC과 IT-SoC의 핵심설계인력양성 사업의 지원으로 수행되었음.

참고 문헌

- [1] A. Stoica. *et al*, "Reconfigurable VLSI Architectures for Evolvable Hardware: from Experimental Field Programmable Transistor Arrays to Evolution Oriented Chips", IEEE Trans. on VLSI Systems, Vol. 9, p. 227-232, Feb. 2001.
- [2] E. Takahashi. *et al*, "Post-Fabrication Clock Timing Adjustment Using Genetic Algorithm", IEEE Journal of Solid-State Circuits," Vol. 39, No. 4, p. 636-642, Apr. 2004.
- [3] J. Huang, "Network Processor Design," Proc. 5th International Conference on ASIC, p. 26-33, Oct. 2003.
- [4] G. Wei. *et al*, "A Fully Digital, Energy-Efficient Adaptive Power Supply Regulator," IEEE Journal of Solid-State Circuits, Vol. 34, No. 4, p. 520-529, Apr. 1999.
- [5] 고영돈, 강홍성, 정민창, 이상렬, 명재민, 윤일구, "신경망과 유전알고리즘을 이용한 광소자용 ZnO 박막 특성 공정 모델링 및 최적화," 2004 전기전자재료 하계학술대회 논문집
- [6] M. Trefzer. *et al*, "New Genetic Operators to Facilitate Understanding of Evolved Transistor Circuits", Proc. NASA/DoD Conference on Evolvable Hardware, p. 217-224, Seattle, USA, Jun. 2004.