

공정 변이 조건 하의 수율 인식 캐시 설계를 위한 캐시 메모리 내로우 밸류 검색

장형범¹, 정성우¹, 윤성로²

¹고려대학교 컴퓨터통신공학부, ²고려대학교 전기전자전파공학부
sryoon@korea.ac.kr

Searching Narrow Values of Cache Memory for Yield-Aware Cache Design under Process Variation

Hyung Beom Jang¹, Sung Woo Chung¹, and Sungroh Yoon²

¹Dept. of Computer and Communication Engineering, Korea University, Seoul, Korea

²School of Electrical Engineering, Korea University, Seoul, Korea (corresponding author)

요 약

공정 기술의 발전에 따라 공정 변이 (process variation)에 따른 수율 (yield) 감소 문제가 대두하고 있으며, 공정 변이 대응 설계 기법 (process variation tolerant design technique)은 하드웨어 제작 시 반드시 고려되어야 할 중요한 요소가 되었다. 캐시 메모리 (cache memory)의 경우에도 공정 변이로 인한 수율 감소 문제에 대처할 수 있는 설계 방법의 개발이 절실하다. 본 논문에서는 캐시에 저장되는 데이터의 특성 분석을 통해 공정 변이에 대응할 수 있는 새로운 캐시 구조 설계에 대한 연구를 소개한다.

1. 서론

반도체 기술의 지속적인 발전에 따라 단일 반도체 칩 (chip)의 크기뿐만 아니라 칩 상에 집적되는 트랜지스터 (transistor) 등 집적 소자의 크기 또한 작아지게 되었다. 하지만 이로 인해 채널 (channel) 길이, 산화물 (oxide) 두께, 임계 전압 (threshold voltage), 채널 상 임의 도핑 문제 (random dopant placement) 등 각종 소자 특성들이 공정에 따라 큰 변이를 보이게 되었다. 이러한 공정 변이 (process variation) 문제는 결국 접근 시간 (access latencies), 동적 전력 소모 (dynamic power consumption), 누설 전력 소모 (leakage power consumption), 그리고 동작 온도 등 하드웨어 (hardware) 전반에 걸쳐 매우 다양한 영향을 미치게 된다 [1][2].

캐시 메모리 (cache memory)의 경우, 이러한 공정 변이 (process variation) 문제로 인해 캐시 메모리 셀 (cache memory cell)에 사용되는 트랜지스터들이 각기

조금씩 다른 특성을 가지게 된다. 이 경우 각각의 셀이 서로 조금씩 다른 특성을 가지게 되어 결과적으로 메모리 셀 각각에 대한 접근 시간 (access time)이 균일화되지 못하거나, 공정 변이가 발생한 일부 캐시 라인 (cache line)에 오류 (fault)가 생기게 된다. 이로 인해 해당 캐시 라인에 읽기/쓰기 동작 (read/write operation)을 수행하지 못하는 경우가 발생하게 된다. 만일 오류가 발생한 캐시 라인 전체를 사용하지 못하게 되면 direct-mapped 캐시의 경우에는 시스템 오동작 (system failure)을 야기할 수 있으며, set-associative 캐시의 경우에는 캐시 일부 (way)를 쓸 수 없게 되어 성능에 있어 상당한 손해를 보게 된다.

이 뿐만 아니라 공정 변이는 수율 (yield) 감소라는 또 다른 문제를 야기하게 된다. 최근 공정 기술의 비약적인 발전에 따른 소자 크기의 급격한 축소에 따라 수율은 50% 이상 감소하였고, 이러한 문제는 공정 기술의 지속적인 발전에 따라 더욱 가중될 것이다. 따라서 계속해서 감소하는 소자 크기를 고려하면서도

일정 수준 이상의 수율을 유지하기 위해서는 오류가 발생한 캐시 라인 전체를 사용하지 않는 것이 아니라 오류 캐시 라인 일부를 계속 사용할 수 있게 해주는 것이 바람직하며, 이를 위한 공정 변이 대응 설계 기술 (process variation tolerant design technique)의 개발 필요성이 절실하다.

본 논문에서는 현재 저자들이 연구하고 있는 공정 변이 대응 설계 기술에 대해 소개한다. 우선 캐시에 저장되는 데이터의 특성 분석을 통해 많은 경우 이들 데이터가 내로우 밸류 (narrow value)라는 사실을 확인하였고 이를 공정 변이 대응 설계 기술 개발에 응용하기 위한 연구를 진행하고 있다.

2. Process Variation Tolerant Technique Using Narrow Values

내로우 밸류 (narrow value)란 워드 (word) 상위 비트 (bit)의 반 이상이 연속적인 영 (0)으로 채워진 값을 말한다. 즉 32-bit 데이터를 기준으로 하였을 때, 상위 16-bit 가 모두 0 인 데이터 워드를 말한다. 따라서 상위 비트에 있는 값들은 아무 의미 없는 값이다. 우리는 마이크로프로세서 (microprocessor) 상에서 사용되는 값들의 상당수가 내로우 밸류라는 점에 착안하여, 내로우 밸류의 특성을 공정 변이 대응 설계 기술에 응용하는 방법에 대해 연구하고 있다.

지금까지 내로우 밸류 (narrow value)와 관련된 연구들은 주로 슈퍼스칼라 프로세서 (superscalar microprocessor)의 성능을 향상시키거나 [3], 전력 소모 혹은 누설 전력 소모 (leakage power consumption)를 줄이기 위한 방법 [4]에 초점이 맞추어져 있었다.

본 연구에서는 내로우 밸류 들이 특정한 공정 변이 대응 기술에 어떻게 이용될 수 있는지 살펴보고자 했다. 이를 위해 우선 캐시에 접근되는 모든 데이터 중에서 내로우 밸류가 차지하는 비율이 어느 정도인지에 대해 살펴보았다. 이러한 내로우 밸류 비율에 대한 분석을 통해 공정 변이로 인한 각종 문제를 완화시킬 수 있는 방법과 오류 발생 시 이에 적절히 대처할 수 있는 캐시 구조를 개발 위한 연구를 수행하고 있다.

기존에는 일부 캐시 라인에 공정 변이로 인하여 오류 (fault)가 발생하게 된다면, 해당 캐시 라인, 더 나아가서는 캐시 메모리 전체를 사용할 수 없었다. 하지만 이러한 오류에 적절히 대처하면서 동작하도록 하는 캐시 설계 기술을 이용한다면, 오류가 발생한 정도 또는 캐시에 들어오는 데이터의 특성에 따라, 문제 있는 캐시 라인까지도 사용할 수 있게 될 것이다.

본 연구를 통해 제안하는 오류 대응 캐시는 다음과 같은 방식으로 동작한다. 오류가 발생한 캐시 라인에 내로우 밸류가 저장되어야 한다면, 무의미한 값을 저장하고 있는 상위 비트 위치에 오류가 발생한 비트의 값들을 저장한다. 이와 반대로 오류가 발생한 캐시 라인에 내로우 밸류가 아닌 값 (와이드 밸류, 혹은 wide value라 불림)이 저장된다면 WVB(Wide Value Buffer)라 불리는 임시 버퍼를 사용하게 된다. 현재 이와 관련하여 진행중인 연구는 4장에 설명되어 있다.

3. Evaluation and Result Analysis

캐시 라인에 접근되는 데이터 중 내로우 밸류의 비율을 조사함으로써, 오류가 발생한 일부 캐시 라인을 위한 버퍼의 크기를 효과적으로 예측할 수 있게 된다.

캐시 메모리에 저장되는 모든 데이터에 대해 해당 데이터가 내로우 밸류인지 아닌지의 여부를 조사하기 위해, 사이클 (cycle) 당 시뮬레이션 기능을 지원하는 simplescalar 3.0 [5]을 수정하여 사용하였다. 그리고 SPEC2000 벤치마크 (benchmark suite) [6]에 포함된 26 개의 애플리케이션 (application)을 모두 실행시켜 각 애플리케이션에 대한 내로우 밸류 비율을 조사해보았다. 또한 읽기/쓰기 동작 (read/write operation) 각각에 대해 내로우 밸류의 비율이 얼마나 되는지를 측정해 보았다. 그림 1 과 그림 2 는 이에 대한 결과를 보여주고 있다.

읽기 동작의 경우, 정수 벤치마크 애플리케이션 (integer benchmark applications)들을 실행시켰을 때, 평균적으로 전체 데이터의 약 34.02%가 내로우 밸류였으며, 부동 소수점 벤치마크 애플리케이션 (floating point benchmark applications)을 실행시켰을

경우 평균적으로 30.84%가 내로우 밸류라는 것을 알 수 있었다. 쓰기 동작의 경우, 정수 및 부동 소수점 벤치마크 애플리케이션을 실행시켰을 때 각각 평균적으로 44.51%, 33.63%가 내로우 밸류라는 것을 알 수 있었다. 이러한 결과는 향후 공정 변이 대응 기술 적용을 위한 임시 버퍼 설계 시 구조 및 크기 결정을 위해 매우 유용하게 사용될 수 있을 것이다.

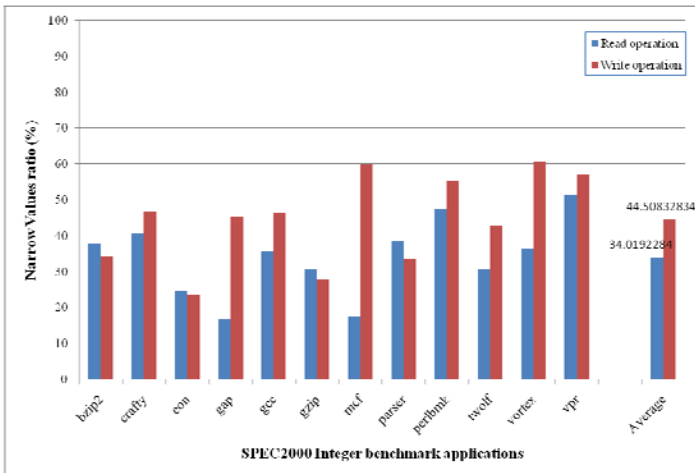


그림 1. 정수 벤치마크 애플리케이션 상의 내로우 밸류 비율

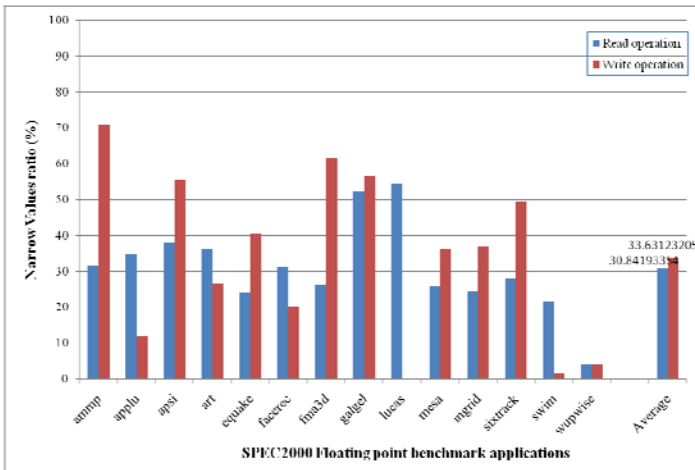


그림 2. 부동 소수점 벤치마크 애플리케이션 상의 내로우 밸류 비율

4. Future Work

공정 변이 (process variation)를 극복한 캐시 메모리 설계를 위해 다음 기술을 제안한다. 우선 새롭게

제안하고자 하는 캐시 구조는 그림 3 과 같다. 이 캐시 구조에서는 기존 구조와는 다르게 WVB(Wide Value Buffer) 라는 임시 버퍼가 추가되었다. 이 WVB 는 캐시 라인의 인덱스 (index)와 데이터를 저장하기 위한 2 개의 필드 (field)로 구성되어있다. 이 임시 버퍼는 오류 (fault)가 발생한 캐시 라인에 내로우 밸류가 아닌 값을 저장해야 할 경우에 사용 된다.

제안된 구조를 사용하는 오류 대응 기술의 첫 번째 방법은 다음과 같다. 공정 변이로 인하여 특정 캐시 라인에 오류가 발생하였을 경우, 우선 이 캐시 라인에 저장해야 할 값이 내로우 밸류 인지 아닌지를 먼저 판단한다. 만일 이 값이 내로우 밸류가 아니라면 WVB 버퍼에 저장을 하게 된다. 반면 해당 값이 내로우 밸류일 경우, 오류가 발생한 캐시 라인의 남은 비트들 (redundant bits)을 이용해서 내로우 밸류를 저장한다.

두 번째 방법은 다음과 같다. 만일 캐시에 저장해야 될 값이 와이드 밸류라면, 상기 첫 번째 방법과 마찬가지로 WVB 버퍼에 해당 값을 저장을 한다. 반면 저장되는 값이 내로우 밸류일 경우에는 오류가 발생하지 않은 캐시 라인에 저장하는 것과 동일하게 저장하되, 오류가 발생한 비트에 위치한 데이터들만 남은 비트들 (redundant bits)에 저장을 하는 방법이다.

캐시에 접근되는 데이터들의 내로우 밸류 비율 분석을 통해 상기 제안된 두 가지의 공정 변이 대응 기술에 사용되는 적절한 WVB 버퍼의 크기를 정할 수 있을 것이며, 현재 이에 관련한 연구를 수행하고 있다.

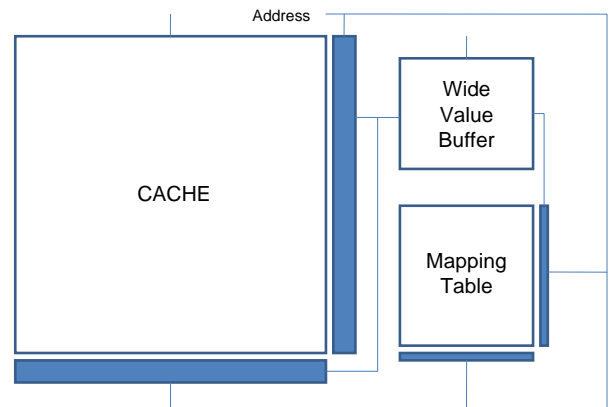


그림 3. 제안된 캐시 구조

5. 결론

우리는 마이크로 프로세서에서 사용되는 값들이 많은 경우 내로우 밸류에 해당한다는점에 착안하여 연구를 시작하였다. 본 논문에서는 공정 변이에 효과적으로 대응하며 작동하는 캐시 메모리의 설계를 위한 선행 작업으로 캐시에 접근하는 데이터들의 내로우 밸류 여부를 조사하였다. 이 결과, 정수 벤치마크 (integer benchmark)의 경우 평균적으로 전체 데이터 중 약 34.02%가 내로우 밸류로서 읽기 동작 시 캐시에서 추출되며, 쓰기 동작 시 전체 데이터 중 약 44.51%가 내로우 밸류로서 캐시에 저장된다는 것을 알 수 있었다. 또한 부동 소수점 벤치마크 (floating point benchmark)의 경우 읽기 및 쓰기 동작 시 각각 30.84%, 33.63%의 데이터가 내로우 밸류로서 캐시에서 읽히거나 쓰여진다는 사실을 알 수 있었다. 이러한 결과는 공정 변이로 인해 특정 캐시 라인에 오류가 발생하였을 때, 이에 대처하며 동작하는 캐시를 설계하기 위한 방법으로 내로우 밸류를 이용하는 것이 가능하다는 점을 나타내준다.

6. 참고문헌

- [1] M. Madhu, N. Vijaykrishnan, "Working with Process Variation Aware Caches", in *Proceedings of Design, Automation and Test in Europe (DATE)*, pp.1152-1157, April 2007.
- [2] O. Unsal, J. W. Tschanz, K. Bowman, V. De, X. Vera, A. Gonzalez and O. Ergin, "Impact of Parameter Variations on Circuits and Microarchitecture", *IEEE Micro*, vol. 26, no. 6, pp.30-39, Nov/Dec 2006.
- [3] D. Brooks and M. Martonosi, "Dynamically Exploiting Narrow Width Operands to Improve Processor Power and Performance," in *Proceedings of The Fifth International Symposium on High Performance Computer Architecture (HPCA)*, pp.13-22, Jan. 1999.
- [4] R. Gonzalez, A. Cristal, D. Ortega, A. Veidenbaum and M. Valero, "A Content Aware Integer Register File Organization," in *Proceedings of The 31st*

International Symposium on Computer Architecture (ISCA), pp.314-324, Jun. 2004.

- [5] T. Austin, E. Larson and D. Ernst, "SimpleScalar: An Infrastructure for Computer System Modeling", *IEEE Computer Magazine*, vol. 35, no. 2, pp. 59-67, Feb. 2002.
- [6] SPEC 2000 Benchmark. <http://www.spec.org>.