

논문 2011-48CI-3-8

캐시 메모리 동기화 전력 감소를 위한 이벤트 메커니즘에 대한 연구

(Research on Event Mechanism for Reducing Power Overheads in Cache Memory Synchronization)

박 영 진*, 정 하 영*, 이 용 석**

(Youngjin Pak, Hayoung Jeong, and Yongsurk Lee)

요 약

본 논문에서는 다중 프로세서 구조에서 캐쉬 메모리 동기화 시 생기는 전력 오버헤드를 줄이기 위한 애니캐스트 방식의 이벤트 드리븐(Event Driven) 동기화 방법을 제안한다. 제안하는 동기화 방법은 기본적인 동기화 프로토콜에서 SHI(Snoop Hit Invalidate) 또는 SHR(Snoop Hit Read) 발생 시, 락 권한 획득에 실패 하였을 때 발생하는 불필요한 폴링 동작을 줄여 줌으로서 버스 대역폭이 낭비되는 것을 막아주고, 통신에 의한 전력 오버헤드를 감소시켜준다. 더 나아가 브로드캐스트 방식의 이벤트 드리븐 동기화 방법에 비해 불필요한 절전 상태 변화로 인하여 생기는 트랜지션 전력을 줄여 전력 소모를 더욱 감소시킬 수 있다는 장점이 있다. 시뮬레이션 결과 기존 스핀-락 방식에 비해 15.3% 정도의 에너지 절감효과를 얻을 수 있었고 브로드캐스트 방식에 비해서 4.7%의 에너지 절감 효과를 얻을 수 있었다. 따라서 제안하는 동기화 방법은 저전력 다중 프로세서 시스템에 적합한 방식이라고 할 수 있다.

Abstract

In this paper, we propose an anycast event driven synchronization mechanism to reduce power overheads. Our proposed mechanism can reduce unnecessary polling operations on SHI(Snoop Hit Invalidate) or SHR(Snoop Hit Read) states. It prevents wasting bandwidth and reduces power overheads on polling operation. Also it decreases transition power of state change compared to broadcast model. Simulation results indicated that the proposed architecture had about 15.3% of power decrease compared to spin-lock model and about 4.7% of power decrease compared to broadcast model. Overall results indicated that proposed synchronization mechanism could increase power efficiency of multi-core system by reducing power overheads.

Keywords : 멀티스레드 프로세서, 공유 메모리 구조, 이벤트 드리븐 메커니즘, 프로세스 동기화

I. 서 론

다중 프로세서는 시스템 설계 분야에서 많은 관심을 받고 있다. 더 높은 성능을 만족시키면서도 일정 수준 이상의 소비 전력을 요구하지 않기 위해, 기존의 주파

수를 높이던 방법에서 이제는 다중 프로세서를 지향하는 방법으로 확실한 전환이 이루어지고 있다.^[1]

그러나 다중 프로세서도 그 성능에 한계는 있다. 성능이 코어 숫자에 비례하지 않고, 코어의 수가 늘어날 수록 데이터 공유를 위해 코어간의 통신이 늘어나고 그에 따른 전력 소모가 늘어나는 등 더 많은 통신 및 전력 오버헤드가 생겨나게 된다. 이처럼 동기화에 따른 추가적인 오버헤드는 다중 프로세서 구조에서 가장 큰 문제로 대두되고 있다.

따라서 공유 메모리 다중 프로세서 구조에서 효율적인 데이터 동기화 방법의 기술 발전은 동기화 오버헤드

* 학생회원, ** 정회원, 연세대학교 전기전자공학과
(School of Electrical & Electronic Engineering, Yonsei University)

※ 본 논문은 지식경제부 출연금으로 ETRI 시스템반도체진흥센터에서 수행한 시스템반도체 융복합형 설계인재양성사업의 연구결과입니다.

접수일자: 2010년12월23일, 수정완료일: 2011년5월12일

를 줄여 나가는 방향으로 나아가고 있고, 기존 방식의 통신 오버헤드 및 전력 오버헤드를 줄이기 위한 이벤트 드리븐 메커니즘^[2~3] (Event Driven Mechanism)과 같은 새로운 기술들이 연구되고 있다.

기존의 데이터 동기화 방식인 스핀-락 동기화는 여러 프로세스 중 하나의 프로세스만 해당 데이터에 대한 권한을 획득하고 나머지는 실패를 한다. 권한 획득에 실패한 프로세스들은 폴링(Polling)을 반복하게 되고 이는 통신 오버헤드를 높이는 결과를 초래한다.^[4]

이러한 문제를 해결하기 위하여 여러 방식의 기술들이 제안이 되었고, 이벤트 드리븐 방식은 기존의 폴링을 사용하는 동기화 구조에 비해 성능 및 저전력 측면에서 더 많은 효율성을 가지고 있다. 이벤트 드리븐 메커니즘의 동기화에서는 이벤트 신호에 따라 동작하므로 폴링을 줄일 수 있어 전체적인 통신 오버헤드가 감소한다. 더욱 나아가 애니 캐스트 방식의 이벤트 드리븐 메커니즘을 사용하면 필요 없는 상태 전환을 없애 트랜지션 파워도 줄일 수 있다.

본 논문에서는 다중 프로세서 환경에서의 효율적인 데이터 동기화를 위하여 애니 캐스트 방식의 이벤트 드리븐 구조를 제안한다. 제안하는 구조는 다중 프로세스 간에 데이터를 공유하는 과정에서 동기화 신호를 브로드캐스팅 하지 않고 애니캐스팅 하는 구조로써 다중 프로세서 간의 데이터 동기화의 효율을 더욱 높여 주도록 설계가 되었다. 이 구조를 사용함으로써 동기화에 필요한 통신이 줄어드는 것은 물론, 전력소모를 줄여 줌으로써 통신 및 전력 오버헤드를 줄여 줄 수 있다는 특징이 있다.

본 논문의 구성은 다음과 같다. 먼저 본론 1절에서는 기본적인 동적 전력관리 기법에 대하여 설명하고, 본론 2절에서는 기존의 동기화 과정에서 생기게 되는 문제점을 설명한다. 본론 3절에서는 제안하는 애니캐스트 방식의 이벤트 드리븐 방법에 대하여 설명한다. 실험에서는 도입 효과를 시뮬레이션 하고, 마지막으로 본 논문의 결론을 맺는다.

II. 본 론

1. 동적 전력 관리

다중 프로세서에 있어서 가장 큰 문제 중의 하나는 프로세싱 과정에 있어 소모 전력을 최소화 하는데 있다. 따라서 동적 전력 관리 (Dynamic Power

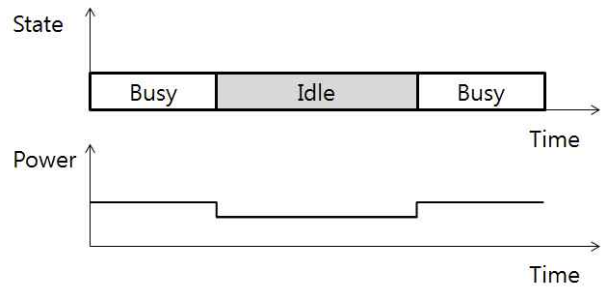


그림 1. 기존 모델의 전력 소모량
Fig. 1. Conventional Power Model.

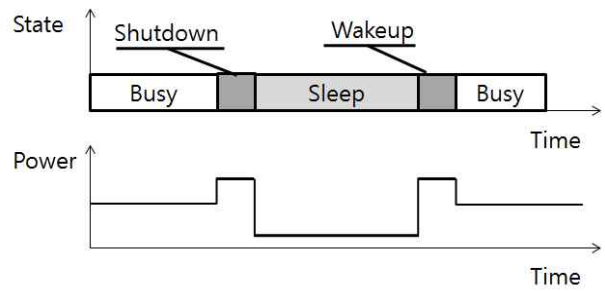


그림 2. 동적 전력 관리 기법의 전력 소모량
Fig. 2. Dynamic Power Management Model.

Management) 및 동적 전압 조절 (Dynamic Voltage Scaling) 등 많은 저전력 기술들이 연구되어지고 있다. 기존의 구조에서는 그림 1과 같이 작업이 없을 때에도 일정량의 전력 소모가 유지되었던 반면, 동적 전력 관리 시스템에서는 동작하는 과정에서 처리할 작업이 없는 경우 그림 2에서와 같이 전원 소비가 낮은 실행모드로 전환하여 에너지를 절약하도록 하는 역할을 수행한다. 그러나 동적 전력 관리 기법에서는 절전 상태에 들어가거나 나오기 위해서 순간적으로 전력 소모량이 증가하게 된다. 따라서 효율적인 동적 전력 관리를 위해서는 Shutdown, Sleeping 시간 및 Wakeup 시간 동안의 에너지 소비량이 기존 모델의 Idle 시간 동안의 에너지 소비량 보다 작아야 한다. Sleeping 상태에서의 시간이 짧을 경우 오히려 에너지 소비량이 늘어나는 문제점이 생기게 된다.^[5]

2. 스핀-락

스핀-락 방식의 동기화 과정은 소프트웨어에서의 동기화 방식으로 락(Lock)을 기초로 한 동기화 방식 중에서는 가장 간단한 메커니즘이다. 스핀-락 방식은 병렬적으로 수행이 되는 스레드들 사이에서 데이터를 공유하는 과정은 해당 데이터에 대한 권한을 획득 할 수 있을 때까지 반복적인 권한 요청을 하는 것이다. 이러한

방식은 “busy-wait” 방식의 메커니즘으로써 획득하려는 권한을 다른 스레드들과 경쟁하는 방식을 사용한다. 권한을 요청한 스레드가 권한을 획득하였을 시에는 공유 데이터에 대한 연산을 모두 수행한 다음에 그 권한을 내려놓게 됨으로써 다른 스레드들에 대한 수행이 가능하게 한다.

이러한 스핀-락 메커니즘은 간단한 메커니즘으로써 각 스레드들 사이에서의 임계구간이 작으면 작을수록 메커니즘의 효율성이 증가가 된다. 권한 획득을 못 할 시에는 기다리면서 계속적인 권한을 확인 하는 간단한 구조로써 운영체제 상에서 프로세스들에 대한 스케줄링을 다시 해주거나 “context switching”하는 구조를 사용하지 않음으로써 동기화 메커니즘의 오버헤드가 작은 구조이다. 하지만 각 스레드 임계 구간이 길면 길수록 스핀-락 메커니즘은 동기화로부터 오는 오버헤드가 다른 메커니즘들에 비해 더욱 비효율적으로 증가하는 단점이 있다. 권한의 상태를 계속적으로 확인해야 하는 타 스레드들은 돌아가면서 권한을 확인 한다는 의미에서 스핀-락이라고도 부른다.

3. 이벤트 드리븐 동기화

본 절에서는 이벤트 방식의 메커니즘을 사용하는 동기화 모델을 소개한다. 기존의 스핀-락 방식과는 달리, 이벤트 드리븐 방식은 공유 데이터를 필요로 하는 스레드가 실행되는 동안 락 쓰기 동작에 실패하였을 시 일시적으로 멈추어 두었다가 이벤트 신호가 발생할 때 이 사건에 대한 대응을 구성하는 방식이다. 이는 기존 방식보다 확장성 및 성능에서 더 효율적이다. 본 절에서는 ARM사의 Cortex-A9에서 사용되고 있는 브로드 캐스트(Broadcast) 방식의 이벤트 드리븐 메커니즘에 대하여 설명하고, 마지막으로 본 논문에서 제안하는 애니 캐스트(Anycast) 방식의 이벤트 드리븐 메커니즘에 대하여 설명하도록 한다.

가. 브로드캐스트 이벤트 드리븐 방법

본 소절에서 소개하는 브로드 캐스트 방식의 이벤트 드리븐 모델은 동기화 과정에서 생기는 이벤트 신호를 모든 프로세서에 브로드 캐스팅 하는 구조이다. 종래의 스누핑 기반 프로토콜에서는 SHI (Snoop Hit Invalidate) 나 SHR (Snoop Hit Read) 발생 시 락 쓰기를 시도하고, 락 권한 획득에 실패하면 락 쓰기에 성공할 때까지 반복하여 스누핑을 한다. 그리고 이 때 불필요한 폴링 동작이

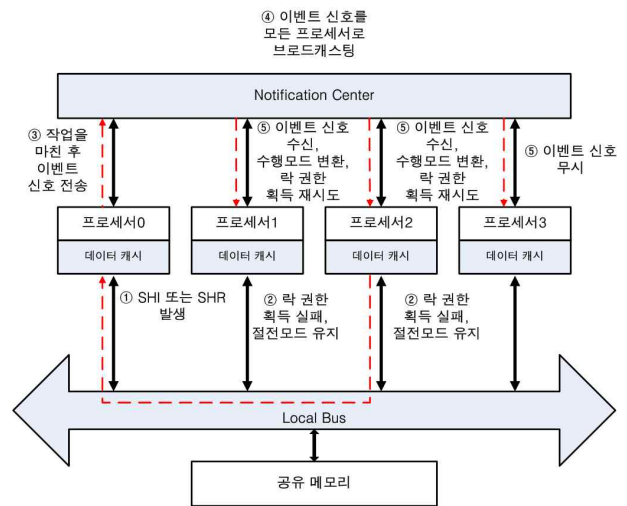


그림 3. 브로드캐스트 이벤트 드리븐 방식의 블록 다이어그램
Fig. 3. Block Diagram of Anycast Event Driven Mechanism.

생기게 된다. 이러한 불필요한 폴링 동작은 시스템 버스에서의 전송 대역폭을 낮추어 통신 오버헤드를 발생 시킬 뿐 아니라 반복되는 신호 트랜지션으로 인한 파워 오버헤드까지 야기시킨다.

따라서 브로드캐스팅 방식의 이벤트 드리븐 메커니즘에서는 반복되는 폴링 동작을 줄여주기 위하여 프로세서가 락 권한 획득에 실패하게 될 시 해당 프로세서는 모든 동작을 멈추고 WFE(Wait for Event) 또는 WFI(Wait for Interrupt) 상태가 되어 절전모드에 들어가게 된다. 그리고 현재 락 권한을 가진 프로세서는 작업을 마침과 동시에 noti피케이션 센터(Notification Center)로 신호를 보내고 noti피케이션 센터는 이를 브로드캐스트하게 된다. 브로드캐스트 신호를 받은 모든 프로세서는 절전모드에서 깨어나 레이스 컨디션 (Race Condition)에 들어가게 되고 다시 락 권한 쓰기를 시도하게 된다.^[6] 그림 3은 위의 과정을 나타낸 블록 다이어그램이다.

브로드캐스트 방식의 이벤트 드리븐 메커니즘에서는 스핀-락과 전력 소모를 줄이기 위해 락 쓰기에 실패하였을 시 절전 모드에 들어가게 된다. 그리고 절전 모드에 들어간 프로세서들은 락 권한을 가진 프로세서가 작업이 끝날 때 이벤트 신호를 브로드캐스팅 함으로써 다시 깨어나게 된다. 그러나 모든 프로세서를 깨우게 되므로 락 권한을 얻기 위한 레이스 컨디션에 들어가게 되고 락 권한 획득에 실패한 프로세서는 다시 절전 모드에 들어가 이벤트 신호를 기다리게 된다. 이 때 프로그램들의 수행 시간이 짧아 잦은 모드 전환이 생길 경우 이는 Shut

down 또는 Wake up 상태에서 트랜지션 과워를 증가시켜 파워 오버헤드를 야기시킨다.

나. 제안하는 애니캐스트 이벤트 드리븐 방법

기존의 브로드캐스트 방식의 이벤트 드리븐 모델은 수행 모드와 절전 모드의 모드 변환으로 인한 전력 오버헤드가 발생했다. 필요 없는 모드 변화가 많을 시 전력 소모가 커지게 된다. 따라서 본 절에서는 이를 최소화 하기 위한 방법으로 애니캐스트 방식의 이벤트 드리븐 모델을 제안한다.

(1) 애니캐스트 이벤트 드리븐 모델의 알고리즘

그림 4는 애니캐스트 이벤트 드리븐 모델의 알고리즘이다. 애니캐스트 방식의 이벤트 드리븐 모델도 브로드캐스트 방식과 마찬가지로 SHI (Snoop Hit Invalidate)나 SHR (Snoop Hit Read) 발생 시 락 쓰기를 시도하고, 락 권한 획득에 실패하면 락 쓰기에 성공할 때까지 반복하여 스누스를 한다. 반복되는 폴링 동작을 줄여주기 위하여 프로세서가 락 권한 획득에 실패하게 될 시 해당 프로세서는 모든 동작을 멈추고 WFE(Wait for Event) 또는 WFI(Wait for Interrupt) 상태가 되어 절전모드에 들어가게 된다.

그리고 여기서 브로드캐스트 모델과의 차이점이 생긴다. 현재 락 권한을 가진 프로세서는 작업을 마치고 동시

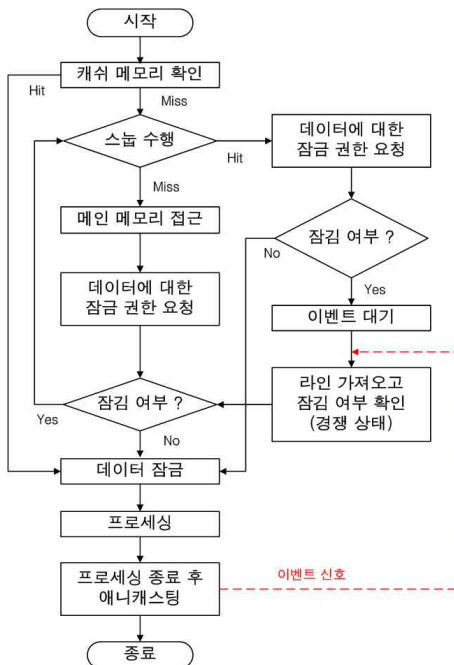


그림 4. 애니캐스트 방식의 이벤트 드리븐 방법
Fig. 4. Anycast Event Driven Mechanism.

에 noti피케이션 센터로 신호를 보내고 noti피케이션 센터는 이를 특정 프로세서로 애니캐스트하게 된다. 특정 프로세서는 FIFO (First In First Out) 또는 LRU (Least Recently Used) 알고리즘에 의해 핫-스팟 또는 콜드-스팟을 구하여 정해진 알고리즘에 따라 핫-스팟을 쉬도록 만들어주거나 콜드-스팟을 지속적으로 모드변환이 없도록 절전 상태로 남겨둔다. 그리고 애니캐스트 신호를 받은 프로세서는 절전모드에서 깨어나 다시 락 권한 쓰기를 시도하게 된다.

(2) 애니캐스트 이벤트 드리븐 모델의 이점

이벤트 드리븐 방식의 동기화 모델에서는 공유 데이터를 사용하려는 프로세서의 수가 많을 시에는 우선 작업 중인 프로세서의 작업이 끝날 때까지 공유 데이터를 요청하는 프로세서들은 모두 절전 모드에 들어가게 된다. 작업 중인 프로세서는 작업을 마친 경우 noti피케이션 센터에 알리게 되는데, 브로드캐스트 방식의 동기화 모델의 경우 절전 모드에 들어간 프로세서들을 구분하지 않으므로 모든 프로세서들을 깨우게 된다. 절전모드에서 깨어난 프로세서들은 다시 레이스 컨디션에 들어가 락 쓰기를 시도하며, 여기서 락 쓰기에 실패한 프로세서는 다시 절전 모드에 들어가 이벤트 신호를 기다리게 된다. 따라서 락 쓰기에 실패한 프로세서들은 필요 없는 모드 전환을 하게 되며 이 때 부가적인 전력 소모가 생기게 된다.

이에 반해 애니캐스트 방식의 이벤트 드리븐 모델에서는 noti피케이션 센터에서 작업이 끝난 프로세서로부터

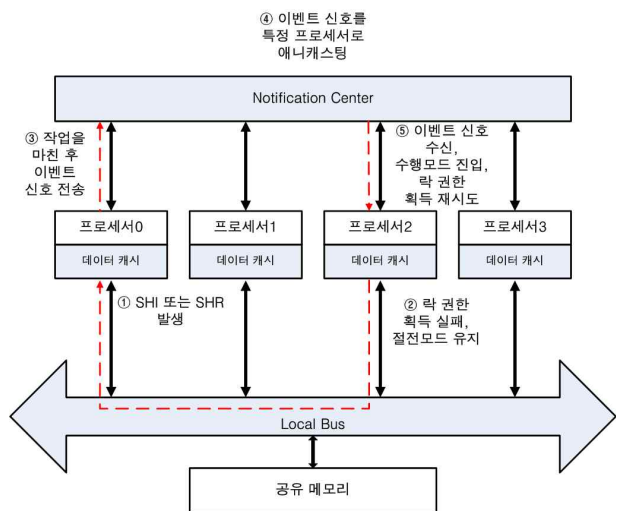


그림 5. 애니캐스트 이벤트 드리븐 방식의 블록 다이어그램
Fig. 5. Block Diagram of Broadcast Event Driven Mechanism.

터 이벤트 신호를 받아다가 특별한 알고리즘에 따라 그림 5에서와 같이 특정 프로세서로 애니캐스트 신호를 보내 깨워 작업을 수행하도록 한다. 그 밖에 이벤트 신호를 받지 못한 프로세서들은 절전 모드에서 깨어나지 않으므로 그 만큼의 모드 전환에 따른 전력 소모를 줄여줄 수 있다. 또 한 브로드 캐스트 방식의 이벤트 트리븐 모델과 마찬가지로 기존의 스누핑 기반의 프로토콜에 비해 불필요한 폴링 동작을 줄여 줌으로서 버스 대역폭이 낭비되는 것을 막아주고, 트랜지션에 의한 전력 오버헤드를 줄여준다.

III. 실험

1. 시뮬레이션 환경

앞서 소개한 두 가지 이벤트 트리븐 동기화 모델은 프로세스 동기화에 있어서의 성능을 확인하기 위해서 M5 시뮬레이터를 이용하여 구성하였다.^[7] 본 논문에서는 이벤트 트리븐 모델의 기본이 되는 스핀-락 메커니즘을 사용하였다. M5 시뮬레이터는 기본 MOESI 프로토콜 기반의 ALPHA 머신 Syscall Emulator를 수정하고 노티피케이션 센터를 추가하였다.

또한 각 모델 별 다중 프로세서의 동기화 과정을 시뮬레이션 하기 위하여 SPLASH-2 벤치마크를^[8] 수행하였다. SPLASH-2 벤치마크는 Stanford 대학에서 개발된 공유 메모리 시스템을 위한 대표적인 벤치마크 프로그램이다. SPLASH-2를 수행하기 위해서 각각의 모델에 물리어 수행하였다. 프로그램으로 Cholesky를 사용하였고 프로그램의 문제코드로는 tk29.0를 사용하였다. 벤치마크 SPLASH-2는 ALPHA 머신 타겟으로 크로스

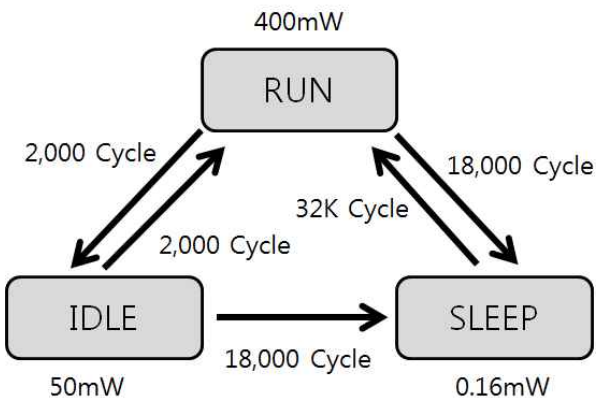


그림 6. SA-1100의 상태 변화에 따른 전력 소모량
Fig. 6. The Power Consumption of SA-1100.

표 1. 시뮬레이션 시스템 구성
Table 1. Construct of Simulation.

시스템	구성
Processor	ALPHA 4-core 1GHz
L1 Cache	32kB
L2 Cache	256kB
L1 Latency	1ns
L2 Latency	10ns

컴파일하여 구성된 시뮬레이터에 연결하였다.

이렇게 구성된 시뮬레이션 환경을 이용하여 본 논문에서 제시된 기존의 브로드캐스트 방식의 이벤트 트리븐 모델 및 제안하는 애니캐스트 방식의 이벤트 트리븐 모델의 시뮬레이션을 수행 하였다. 표 1은 시뮬레이션 환경 구성표이다.

2. 벤치마크 테스트 수행시간 측정

다음은 벤치마크 테스트 수행 결과 및 시간에 대한 설명이다. 표 3은 각각의 메커니즘 모델에서 SPLASH2 Cholesky 벤치마크 프로그램을 수행하였을 시 락 쓰기 시도 횟수 및 락 쓰기 실패 횟수이다. 애니캐스트 방식을 사용하였을 때 기존의 스핀락 구조에 비하여 97.7%, 브로드캐스트 방식에 비해 37.5% 락 쓰기 시도 횟수를 줄일 수 있었다. 그 중 락 권한 획득에 성공하여 실제로 작업을 수행한 경우는 125회이고 그 이외에는 불필요한 폴링이다.

스핀-락 메커니즘의 경우 11,702회 락 쓰기 시도를 실패하였다. 이에 비해 이벤트 트리븐 동기화 방식에서는 불필요한 폴링 횟수를 많은 량 줄일 수 있다. 브로드캐스트 방식의 경우 불필요한 폴링 횟수를 296회로 줄여 주었고, 애니캐스트 방식의 경우 138회로 줄였다.

표 2. 각 메커니즘의 락 쓰기 시도 횟수 및 실패 횟수

Table 2. Number of Trying to Lock and Failure to Lock.

	락 쓰기 성공 횟수	락 쓰기 실패 횟수
스핀-락	11,827 회	11,702 회
브로드캐스트	421 회	296 회
애니캐스트	263 회	138 회

3. 측정된 수행 시간에 따른 전력 소모량 예상

본 소절에서는 앞에서 측정된 시뮬레이션 수행 시간

표 3. 측정된 수행 시간에 따른 예상 평균 전력 소모량

Table 3. Power Consumption of Simulation.

	스핀-락	브로드캐스트	애니캐스트
총 수행 시간 (x4)	20,012x10 ⁶ (80,048x10 ⁶)	20,020x10 ⁶ (80,080x10 ⁶)	20,021x10 ⁶ (80,084x10 ⁶)
WFE 시간	-	43,344x10 ⁶	46,519 x10 ⁶
평균전력 소모량	652.97	580.31	553.23

에 따른 전력 소모량을 측정해 본다. 코어의 수행 모드와 절전 모드에서의 전력 소모량은 SA-1100을 기본 모델로 사용하였다.^[5] 그림 6는 전력 모델로 사용한 SA-1100의 각 상태에서의 전력 소모량 및 모드 전환에 걸리는 시간이다. SA-1100 플랫폼의 경우 수행모드에서의 평균 소모 전력은 400mW이고 휴식모드에서는 50mW, 그리고 절전 모드에서는 0.16mW이다.

표 3은 벤치마크가 수행된 총시간 및 WFE 시간 그리고 이로 인해 예상되는 평균 전력 소모량이다. 시뮬레이션 과정에서의 전력 소모량은 스핀-락 모델의 경우 평균652.97mW, 브로드캐스트 모델 580.31mW였고, 제안하는 애니캐스트 방식의 모델은 553.23mW 였다. 결과적으로는 애니캐스트 방식의 이벤트 드리븐 메커니즘은 스핀-락에 비해 15.3% 정도의 에너지 절감을 브로드캐스트 방식의 이벤트 드리븐 메커니즘에 비해서 4.7%의 에너지 절감을 얻을 수 있었다.

IV. 결 론

최근 임베디스 시스템에서는 프로세서의 개수를 증가시키는 추세이며 이에 따라 동기화로부터 생기는 오버헤드가 더 증가하게 된다. 동기화 과정의 오버헤드를 줄이기 위하여 많은 기술들이 제안되었고, 기존의 방식에서 생기는 통신 및 전력 오버헤드를 줄이기 위하여 이벤트 방식의 동기화 메커니즘들이 제안이 되었었다. 기존의 이벤트 방식의 동기화 메커니즘으로는 ARM사의 Cortex-A9 아키텍처가 있지만 잦은 모드 전환으로 인해 트랜지션 전력 소모를 야기시킨다. 그러므로 본 논문에서는 애니캐스트 이벤트 드리븐 동기화 방법을 이용하여 트랜지션 파워 소모를 줄일 수 있는 방법을 제안하였다.

결과적으로는 애니캐스트 방식의 이벤트 드리븐 방

법은 스핀-락에 비해 15.3% 정도의 에너지 절감을 브로드캐스트 방식의 이벤트 드리븐 메커니즘에 비해서 4.7%의 에너지 절감을 얻을 수 있었다. 따라서 본 논문에서 제안하는 애니캐스트 방식의 이벤트 드리븐 메커니즘은 전력 감소 효과가 있고 공유 메모리 다중 프로세서 구조에서 프로세서 동기화로 인한 오버헤드를 줄일 수 있는 것을 확인 할 수 있었다.

참 고 문 헌

- [1] 리처드 로(Richard Low), "마이크로 프로세서 동향과 멀티 코어, 메모리 및 전원 관리," Freescale Semiconductor Report, 2007.
- [2] Monchiero. M, Palermo. G, Silvano. C, Villa. O, "Efficient Synchronization for Embedded On-Chip Multiprocessors," IEEE Trans. on Very Large Scale Integration Systems, Vol. 14, no. 10, pp.1049-1062, Oct. 2006.
- [3] Shaoshan. Liu, Jean-Luc Gaudiot, "Synchronization Mechanisms on Modern Multi-core Architecture," Advances in Computer Systems Architecture Lecture Notes in Computer Science, Vol. 4679, pp-290-303, 2007.
- [4] Hennessy. J.L, Patterson. D.A, "Computer Architecture - A Quantitative Approach," Morgan Kaufmann, SanFrancisco, 2006.
- [5] Luca Benini, Alessandro Bogliolo, and Giovanni De Micheli "A Survey of Design Techniques for System-Level Dynamic Power Management," IEEE Trans. on Very Large Scale Integration Systems, Vol. 8, no. 3, pp.299-316, JUNE 2000.
- [6] John Goodacre, "Low Power Processing: Use Only the Power Needed to Get the Job Done," on ARM Technical Report, Dec. 2008.
- [7] Binkert N.L., Dreslinski R.G., Hsu L.R., Lim, K.T., Saidu A.G., Reinhardt S.K., "The M5 Simulator: Modeling Networked Systems," on Micro, Vol. 26, no. 4, pp.52-60, Aug. 2006.
- [8] S. Woo, M. Ohara, E. Torrie, J.P. Singh, A. Gupta. "The SPLASH-2 programs: Characterization and methodological considerations," in Proc. of the 22th International Symposium on Computer Architecture, Vol. 23 no. 2, Italy, June 1995.

저 자 소 개



박 영 진(학생회원)
2009년 중앙대학교 전자전기
공학부 학사 졸업.
2011년 연세대학교 전기전자
공학과 석사 졸업.
2011년~현재 삼성전자 연구원.

<주관심분야 : 마이크로 프로세서, 컴퓨터 아키텍처, 메모리구조>



정 하 영(학생회원)
2003년 중앙대학교 전자전기
공학부 학사 졸업.
2005년 연세대학교 전기전자
공학과 석사 졸업.
2010년 연세대학교 전기전자
공학과 박사과정.

2010년~현재 삼성전자 책임연구원.
<주관심분야 : 마이크로 프로세서, ASIC, SoC>



이 용 석(정회원)
1973년 연세대학교 전자공학과
학사 졸업.
1977년 University of Michigan
Electrical Engineering
석사 졸업.
1981년 University of Michigan
Electrical Engineering
박사 졸업.

1933년~현재 연세대학교 전기전자공학과 교수.
<주관심분야 : 마이크로프로세서 설계, VLSI
설계, DSP 프로세서 설계, 고성능 연산기 설계>