

모바일 내장형 시스템을 위한 듀얼-포트SDRAM의 성능 평가 및 최적화

(Performance Evaluation and Optimization of Dual-Port SDRAM Architecture for Mobile Embedded Systems)

양 회 석 [†] 김 성 찬 ^{**}
(Hoeseok Yang) (Sungchan Kim)

박 해 우 [†] 김 진 우 [†]
(Hae-woo Park) (Jinwoo Kim)

하 순 회 ^{***}
(Soonhoi Ha)

요 약 최근 듀얼-프로세서 기반의 모바일 내장형 시스템을 위한 듀얼-포트SDRAM이 발표 되었다. 이는 단일 메모리 칩이 두 프로세서의 로컬 메모리와 공유 메모리 역할을 모두 담당하므로 공유 메모리를 위하여 추가의 SRAM 메모리를 사용하는 기존의 구조에 비해 더 간단한 통신 구조이다. 양 포트로부터의 동시적인 접근에서의 상호

배타성을 보장하기 위하여 모든 공유 메모리 접근에는 특수한 동기화 기법이 수반되어야 하는데 이는 잠재적인 성능 악화의 원인이 된다. 이 논문에서는 이러한 동기화 비용을 고려하여 듀얼-포트SDRAM 구조의 성능을 평가하고, 주 응용의 통신 특성을 고려하여 최적화한 락우선권 기법과 정적복사 기법을 제안한다. 더 나아가, 공유 뱅크를 여러 블록으로 나눔으로써 서로 다른 블록들에 대한 동시적인 접근을 가능케 하여 성능을 개선하도록 한다. 가상 프로토타이핑 환경에서 수행된 실험은 이러한 최적화 기법들이 기본 듀얼-포트SDRAM 구조에 비하여 20-50%의 성능 향상을 가져옴을 보여준다.

키워드 : 메모리 구조, 듀얼-포트SDRAM, 모바일 내장형 시스템

Abstract Recently dual-port SDRAM (DPSDRAM) architecture tailored for dual-processor based mobile embedded systems has been announced where a single memory chip plays the role of the local memories and the shared memory for both processors. In order to maintain memory consistency from simultaneous accesses of both ports, every access to the shared memory should be protected by a synchronization mechanism, which can result in substantial access latency. We propose two optimization techniques by exploiting the communication patterns of target applications: lock-priority scheme and static-copy scheme. Further, by dividing the shared bank into multiple blocks, we allow simultaneous accesses to different blocks thus achieve considerable performance gain. Experiments on a virtual prototyping system show a promising result - we could achieve about 20-50% performance gain compared to the base DPSDRAM architecture.

Key words : Memory Architecture, Dual-port SDRAM, Mobile Embedded System

1. 서 론

최근의 모바일 내장형 시스템은 오디오, 비디오, 3D 게임 등 더욱 강력한 계산 능력을 요구하는 다양한 멀티미디어 응용을 지원한다. 가장 전형적인 구조는 기본적인 콜-프로세싱과 모뎀 기능을 수행하는 모뎀프로세서와 멀티미디어와 같이 계산량이 많은 응용을 수행하는 응용프로세서로 이루어진 듀얼-프로세서 시스템이다. 일반적으로 응용 프로세서는 여러 코어가 하나의 칩에 집적된 형태를 지니기도 하지만, 이 논문에서는 하나의 코어를 가진 것으로 가정한다.

그림 1은 다양한 듀얼-프로세서 구조를 도시한 것이다. 이중 하나는 주변장치 버스나 범용 입출력 포트(GPIO)와 같이 기존에 존재하는 통신 미디어를 재활용하는 것이다. 그림 1의 (a)가 바로 이러한 경우인데, 두 프로세서가 LCD 버스를 공유하여 통신하는 구조로 실

· 본 연구는 BK21 프로젝트, 과학기술부 도약연구지원사업(R17-2007-086-01001-0)에 의해 지원되었다. 또한 서울대학교 컴퓨터신기술연구소와 IDEC은 본 연구에 필요한 기자재들을 지원해 주었다.

· 이 논문은 제34회 추계학술대회에서 '모바일 내장형 시스템을 위한 듀얼-포트SDRAM의 성능 평가 및 최적화'의 제목으로 발표된 논문을 확장한 것임

[†] 학생회원 : 서울대학교 전기컴퓨터공학부
dalgachi@iris.snu.ac.kr
starlet@iris.snu.ac.kr
jwkim@iris.snu.ac.kr

^{**} 정 회 원 : 서울대학교 전기컴퓨터공학부
sungchan.kim@iris.snu.ac.kr

^{***} 정 회 원 : 서울대학교 전기컴퓨터공학부 교수
sha@iris.snu.ac.kr

논문접수 : 2008년 1월 10일
심사완료 : 2008년 4월 5일

Copyright©2008 한국정보과학회 : 개인 목적이거나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 컴퓨팅의 실제 및 레터 제14권 제5호(2008.7)

제 상용화된 휴대전화에서 사용되고 있다. 이 경우 주변 장치 버스를 재활용함으로써 비용이 절감되지만, 낮은 통신 대역폭이 문제가 된다.

듀얼-포트 메모리는 이러한 구조에서 성능 향상을 위한 좋은 방안이 될 수 있다. 지금까지 멀티 프로세서 환경에서의 효율적인 프로세서간 통신을 위하여 메모리 구조 최적화[1,2]와 멀티-포트 메모리[3,4]를 중심으로 많은 연구가 이루어져왔다.

최근 삼성전자는 그림 1의 (b)에 나타난 형태인 하나의 공유 बैं크와 두 프로세서를 위한 각각의 전용 बैं크를 가지는 듀얼-포트SDRAM, OneDRAM™[5]을 발표하였다. 두 전용 बैं크는 두 프로세서 각각을 위한 로컬 메모리 역할을 담당하고, 중앙의 특수한 बैं크는 두 프로세서간 통신을 위하여 존재한다. 이는 크게 두 장점을 지니는데, 첫째는 SRAM을 공유메모리로 사용하는 것보다 더 큰 용량의 공유 영역을 제공한다는 것이고, 둘째는 단일 칩 솔루션과는 다르게, 프로세서의 종류에 구애 받지 않고 적용 가능하기 때문에 다양한 내장형 시스템에 쉽게 채용 가능하다는 것이다. 이 논문의 나머지 부분에서 OneDRAM™을 기본 DPSDRAM 구조라 칭하기로 한다.

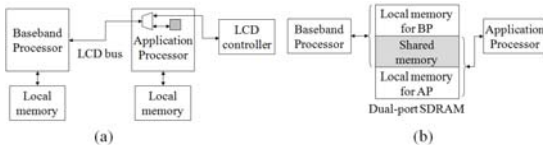


그림 1 듀얼-프로세서 구조: (a) 주변장치 버스의 재활용, (b) DPSDRAM 구조(OneDRAM™ [5])

서로 다른 두 프로세서가 같은 메모리 영역을 공유하기 위해서는 접근 시 동기화 비용을 지불하여야 한다. 이러한 동기화 부하는 자칫 DPSDRAM 구조의 성능 이득 보다 더 크게 작용하여 오히려 전체 성능을 저하시킬 수 있다. 이 논문에서는 이러한 부하를 줄이기 위하여 기본 DPSDRAM 구조와 더불어 락우선권 기법과 정적복사 기법, 그리고 공유 बैं크를 다수의 블록으로 분할하는 기법을 제안하고 평가한다.

2. 듀얼-포트 SDRAM 구조

그림 2는 기본 DPSDRAM 구조의 구성을 보여주고 있다. 모뎀 프로세서와 응용 프로세서는 두 개의 전용 बैं크와 하나의 공유 बैं크를 지니는 DPSDRAM에 연결되어 있다. 그림에서 하드웨어 세마포어는 명시적으로 외부에 도시되어있으나 실제로는 메모리 패키지 안에 포함되어 있다. 공유 메모리에 대한 물리적 접근 충돌을

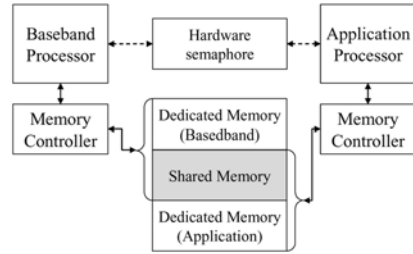


그림 2 DPSDRAM 구조의 기본 구성

방지하기 위하여 두 프로세서는 모두 공유 बैं크 접근 전에 하드웨어 세마포어를 취득하여야만 한다. 그림에서 실선은 데이터의 이동을 나타내며, 점선은 하드웨어 세마포어 명령 전달을 나타낸다.

하드웨어 세마포어가 공유 बैं크에 대한 물리적 상호 배타 접근을 제공하지만 이는 아직 충분하지 않다. 논리적인 상호 배타 접근을 제공하기 위해서는 소프트웨어 세마포어가 제공되어야 한다. 이 소프트웨어 세마포어는 두 프로세서 간에 공유되어야 하므로, 공유 बैं크 안에 위치하여야 한다. 따라서, 공유 बैं크에 존재하는 공유 자료 구조에 대하여 접근하기 위해서는 하드웨어 세마포어는 물론, 소프트웨어 세마포어를 접근하는 부하까지 감당하여야 한다. DPSDRAM 구조의 성능 이득을 극대화하기 위해서는 이러한 동기화 부하를 최소화하는 것이 매우 중요하다.

프로세서가 많은 양의 데이터를 한번에 접근하는 경우 세마포어에 부하가 큰 접근 시간에 의해 상쇄되는 효과가 있다. 보통, 전송에서 데이터를 쓰는 프로세서와 읽는 프로세서 모두 공유 बैं크를 접근 하므로 동기화 부하는 배가된다.

3. 최적화 기법

3.1 락우선권 기법

모바일 내장형 시스템에서 모뎀 프로세서와 응용 프로세서는 통신에 있어서 서로 다른 요구사항을 지닌다. 우선, 대부분의 통신은 모뎀 프로세서에서 응용 프로세서로 단방향으로 이루어진다. 또한, 모뎀 프로세서는 빈번하지는 않지만 한번 통신이 시작되면 많은 양을 한꺼번에 전송하는 반면, 응용 프로세서는 작은 양의 데이터를 매우 빈번하게 접근한다.

모뎀 프로세서에서는 RF단의 통신 처리 등 실시간 제약이 있으므로 공유 메모리 접근을 위하여 하드웨어 세마포어를 취득하려 할 때 대기 시간을 줄이는 것이 중요하다. 반면 응용 프로세서의 경우 성능 측면에서 볼 때 하드웨어 세마포어에 접근하는 부하가 가장 중요한 요소이다. 응용 프로세서는 작은 양의 데이터를 빈번하게 접

근하므로 동기화에 대한 부하가 누적되기 때문이다.

이러한 관찰에 기반하여 응용 프로세서에 기본으로 접근 권한을 주는 락우선권 기법이 고안되었다. 모뎀 프로세서가 접근을 요구하면 시스템은 즉시 접근 권한을 모뎀 프로세서로 전환한다. 모뎀 프로세서가 전송을 끝내고 하드웨어 세마포어를 해제하면 접근 권한은 다시 응용 프로세서로 자동으로 전환된다. 따라서, 응용 프로세서는 공유 메모리 접근 시 하드웨어 세마포어를 취득하지 않아도 된다. 한편 모뎀 프로세서는 공유 메모리 접근 시, 요청 즉시 하드웨어 세마포어를 얻으므로 대기 시간이 없다. 결과적으로 제안하는 락우선권 기법은 서로 다른 두 프로세서 측의 요구사항을 모두 만족한다.

그림 3은 락우선권 기법을 구현한 락메니저가 응용 프로세서 측의 메모리 컨트롤러에 장착된 전체 시스템을 도시한 것이다. 락메니저는 두 프로세서로부터 공유 메모리 접근 요구를 입력으로 받는다.

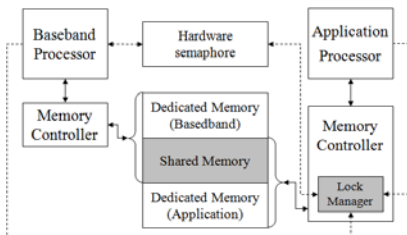


그림 3 락우선권 기법의 구조

3.2 정적복사 기법

정적복사 기법은 응용 프로세서가 같은 데이터를 여러 차례 반복적으로 접근하는 경향이 있다는 관찰에 기반하고 있다. 3D 렌더링 응용이 좋은 예이다. 모뎀 프로세서에 의하여 3D 렌더링을 위한 데이터들이 공유 बैं크에 전송된 이후에 응용 프로세서는 그 데이터들을 여러 차례 반복적으로 접근하며 3D 렌더링을 수행한다.

이 경우 반복적인 접근에서 매번 동기화 부하를 겪지 않기 위해서는 그 데이터를 로컬 메모리로 복사한 후 접근하는 것이 이득이 된다. 로컬 메모리의 경우 공유 영역과는 달리 캐시가 유효화 되어있기 때문에 접근 빈도가 높아질수록 성능의 이득은 더 커진다. 이에 모뎀 프로세서가 그림 4(a)에 보듯 응용 프로세서와 데이터 버스를 공유하는 변형된 DMA 컨트롤러를 통해 응용 프로세서의 로컬 메모리로 직접 데이터를 전송하는 정적복사 기법이 고안되었다.

이 정적복사 기법에서 공유 메모리는 오직 전송을 위한 중간 버퍼로만 사용되며, 전체 시스템은 프로세서간 통신이 “put-get” 기법[6]을 통해 이루어지는 분산 메모리 시스템으로 동작한다. 응용 프로세서는 모뎀 프로세서

에게 전송 전에 데이터를 받을 위치를 알려준다. 그 후, 모뎀 프로세서는 데이터를 응용 프로세서의 개입 없이 명시된 지점으로 전송한다. 이 기법은 특히 응용 프로세서 측에 공유 데이터를 사용하는 전용 하드웨어가 있을 경우 유용하다. 전용 하드웨어의 경우 하드웨어/소프트웨어 세마포어를 운용하는 것이 쉽지 않으므로 정적복사 기법을 통해 로컬 메모리에 데이터를 전송한 후 접근하는 것이 효율적이다. 모뎀 프로세서는 공유 बैं크의 하드웨어 세마포어를 취득하고, 데이터를 공유 बैं크에 전송 한 후, 하드웨어 세마포어를 해제한다. 그 후, 변형된 DMA 컨트롤러를 통해 공유 बैं크의 데이터를 응용 프로세서의 로컬 메모리로 복사한다.

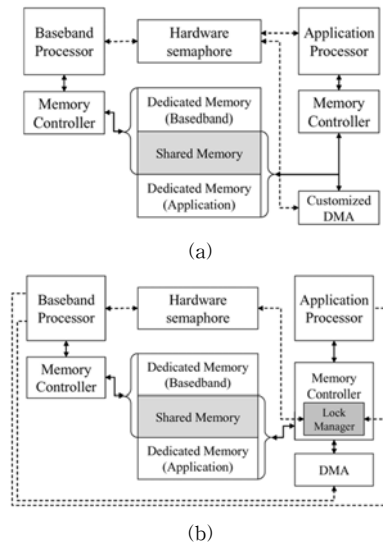


그림 4 정적복사 기법의 구조: (a) 락메니저가 있는 경우, (b) 락메니저가 없는 경우

그림 4(a)는 정적복사 기법을 위해 변형된 DMA 컨트롤러가 응용 프로세서 측에 장착된 구조를 보여준다. 변형된 DMA 컨트롤러는 일반적인 DMA 컨트롤러와는 달리 하드웨어 세마포어를 운용하여야 한다. 모뎀 프로세서의 요청으로 DMA 컨트롤러가 전송을 수행하기 전에 우선 하드웨어 세마포어를 취득하고 전송을 수행하는 동안 접근 권한을 유지하여야 한다. DMA 컨트롤러가 전송을 수행하는 동안 모뎀 프로세서는 그 공유 बैं크의 영역을 접근할 수 없다. 그림 4(b)는 락메니저가 함께 통합된 더 개선된 구조를 보여준다. 이 경우 락메니저가 DMA 컨트롤러 대신에 하드웨어 세마포어를 운용하므로 더 간단하고 일반적인 DMA 컨트롤러가 사용될 수 있다.

정적복사 기법은 공유 메모리에서 응용 프로세서의 로컬 메모리로 추가적인 전송을 수반하기 때문에 데이

타 전송 시간이 전보다 더 길어지는 단점이 있다. 그러나, 이러한 부하는 응용 프로세서의 계산 시간이 DMA의 데이터 전송 시간과 겹치는 경우 숨겨질 수 있다. 이는 실험을 통해 증명될 것이다.

3.3 공유 बैं크의 블록 분할

연속된 데이터 전송 사이의 병렬성을 극대화시키기 위하여 공유 बैं크의 영역을 여러 개의 블록으로 나눌 수 있다. 이 다수의 블록에 대하여 동시적인 접근을 허용함으로써 각 블록은 각각 하드웨어 세마포어를 별개로 가져야 한다.

블록들의 동시적 접근은 연속된 데이터 전송에서 각각이 서로 시간상으로 겹칠 수 있도록 하여 데이터 전송 성능을 높인다. 이는 잘 알려진 생산자-소비자 예제에서의 더블 버퍼링 기법의 기본 원리이다. 즉, 응용 프로세서(소비자)의 데이터 읽기는 모뎀 프로세서(생산자)의 데이터 쓰기와 시간 상으로 겹칠 수 있다. 정적복사 기법에서도 다수의 블록을 차례로 임시 버퍼로 사용함으로써, 연속된 정적복사 전송이 서로 시간상으로 겹칠 수 있다.

4. 실험 및 검증

시스템의 환경을 다음과 같이 가정하였다. 모뎀 프로세서와 응용 프로세서의 동작 클럭은 각각 250MHz, 500MHz로 설정하였다. 모든 메모리 인터페이스와 DPSDRAM은 모두 100MHz에서 동작함을 가정하였다. 그림 1(a)의 기존 시스템을 위하여 두 개의 오프-칩 SDRAM과 LCD 버스를 가지는 시스템을 모델하였는데, 이 경우 LCD 버스의 클럭은 20MHz로 가정하였다.

4.1 최적화 기법이 적용된 DPSDRAM의 검증

첫 실험은 기본 DPSDRAM과 3장에서 제안된 여러 최적화 기법의 효용성을 검증해보기 위한 것이다. 그림 5는 다양한 DPSDRAM 구조의 성능 비교를 보여주고 있다. 'A'는 기본 DPSDRAM 구조, 'B', 'C', 'D'는 각각 정적복사, 락우선권, 두 기법 모두 적용된 DPSDRAM 구조의 성능을 나타낸다. 알파벳 뒤에 제시된 숫자는 응용 프로세서가 전송된 후에 같은 데이터 셋을 반복적으로 접근한 회수를 나타낸다.

실험은 실제 LCD 버스 구조 기반의 3세대 휴대전화용 듀얼-프로세서 구조에서 추출해낸 전송 패턴을 기반으로 수행되었다. 모뎀 프로세서는 1-10KB의 데이터를 공유 메모리에 한번에 전송하고, 응용 프로세서는 그 데이터를 32B 단위로 접근하며 전체 전송량은 256 KB로 설정하였다.

LCD 버스 구조는 낮은 대역폭으로 인해 기본 DPS-DRAM에 비해 약 4배 이상 느린 성능을 나타내므로, LCD 버스 구조의 실험결과는 지면 관계상 생략하도록 한다. 비반복적인 접근에 관해서는 정적복사, 락우선권

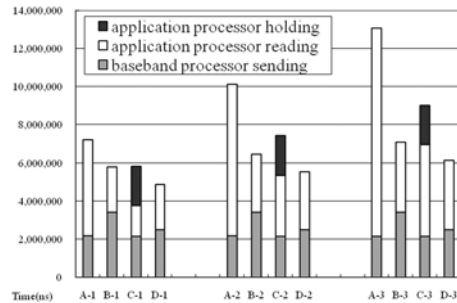


그림 5 다양한 접근 횟수에 따른 DPSDRAM 구조들의 성능 비교

기법 모두 기본 구조에 비해 약 20%의 성능 향상을 보여준다. 이러한 성능 향상은 응용 프로세서 측의 데이터 읽기 과정에서의 수행 시간 단축으로 인한 것이다. 정적복사 기법의 경우 모뎀 프로세서에 의한 데이터 송신 시간은 DMA 때문에 증가하는 것을 볼 수 있다. 락우선권 기법에서 응용 프로세서에서 데이터를 읽어가는 부분의 약 56%가 접근 권한 획득이 지연되어 대기 하는 시간에 할애되었음을 확인할 수 있다. 정적복사와 락우선권 기법을 모두 적용한 경우 최고의 성능 향상을 보여주는데 이는 기본 DPSDRAM을 대상으로 비교하면 약 33% 정도이다. 이는 정적복사와 락우선권을 동시에 사용하면 락우선권 기법만 사용했던 경우와 비교할 때 더 이상 접근 권한 대기에 지연되는 시간이 없기 때문이다.

응용 프로세서에서 같은 데이터 셋을 반복적으로 접근할 때 정적복사의 효용은 더욱 커진다. 정적복사 기법의 이득을 검증하기 위하여 실험에서 모뎀 프로세서에 의한 데이터 송신 이후, 응용 프로세서가 데이터를 접근하는 회수를 늘려보았다. 기본 DPSDRAM 구조와 락우선권만을 사용하는 경우는 상대적으로 낮은 확장성을 보임을 확인할 수 있다. 특히 기본 DPSDRAM의 경우 데이터 접근 회수가 커짐에 따라 누적되는 동기화 부하가 늘어나 성능이 매우 악화된다. 락우선권 기법만 적용한 경우와 정적복사 기법만 적용한 경우의 성능차이도 접근 회수가 늘어남에 따라 더 커진다. 정적복사 기법의 경우 추가의 DMA 전송 부하가 있지만 전송 완료된 데이터가 로컬 영역에 존재하므로 캐시 효과를 얻을 수 있어 데이터 접근 회수가 증가함에 따라 이를 별충하는 더 큰 이득을 얻는 것으로 나타났다. 이는 데이터 접근 회수의 확장성 측면에서 정적복사 기법이 락우선권 기법보다 우수함을 의미한다. 정적복사와 락우선권 기법을 동시에 사용하는 경우 정적복사에서의 동기화 부하가 락우선권 기법에 의해 줄어들기 때문에 3번의 접근의 경우 기본 DPSDRAM 구조보다 최고 53%까지의 성능 향상 효과를 보였다.

4.2 공유 बैं크 내 블록 분할의 검증

두 번째 실험은 공유 बैं크를 다수의 블록으로 나누는 구조 최적화의 효용성을 검증하기 위한 것이다. 더블버퍼링 기법을 사용하여 서로 다른 블록들 간에 동시적 접근을 허용함으로써 추가의 성능 향상을 얻을 수 있다. 즉, 모뎀 프로세서가 하나의 공유 बैं크 블록에 데이터 셋을 쓰는 동안 응용 프로세서는 다른 블록에 이미 전송된 데이터 셋을 읽을 수 있다.

여기서는 기본 DPSDRAM 구조를 공유 बैं크가 네 개의 블록으로 나뉘어져 있고 어떤 최적화 기법도 사용되지 않은 경우로 칭한다. 그림 6에서 'A'와 'B'는 각자 이 기본 DPSDRAM 구조와 이 구조에 더블버퍼링을 적용한 경우를 나타낸다. 'C'와 'D'는 락우선권 기법과 정적복사 기법이 적용된 경우를 나타내고, 'E'는 정적복사 기법에 더블버퍼링 기법이 추가된 경우를 나타낸다. 'F'는 모든 기법(락우선권, 정적복사, 더블버퍼링)이 동시에 적용된 결과를 나타낸다.

공유 बैं크를 다수의 블록으로 나누는 것의 효과는 매우 명백히 나타난다. 기본 구조에 더블버퍼링 기법만 적용한 경우 기본 구조에 비하여 17-30%의 성능 향상을 보여준다. 서로 다른 블록을 동시에 접근하는 것을 허용함으로써 두 프로세서 간 공유 메모리 접근의 충돌은 줄어들었으나, 이에 의한 성능 향상은 접근 회수가 늘어남에 따라 줄어든다.

락우선권 기법을 더블버퍼링과 같이 적용한 경우('C') 더블버퍼링 기법만 적용한 경우에 비해 25-40% 향상되고, 락우선권 기법만 사용한 경우에 비해서는 약 30% 향상된다. 여기서 전 실험에서 확인된 락우선권의 낮은 확장성에 주목할 만 한데, 락우선권 기법이 더블버퍼링 기법과 동시에 적용되는 경우 성능이 정적복사와 락우선권을 동시에 사용했던 경우에 근접한 정도로 향상되는 것을 확인할 수 있다.

정적복사 기법에서도 더블버퍼링 기법을 활용하는 경우 이득을 볼 수 있다. 공유 बैं크의 여러 블록을 정적복사 기법의 임시 버퍼로 사용하는 경우 연속된 데이터

전송의 시간이 시간 상으로 겹칠 수 있기 때문이다. 그림 6의 'E'에서 더블버퍼링 기법을 사용한 경우 정적복사 기법만을 사용한 것('D')에 비해 모뎀 프로세서의 송신 시간은 15%, 전체 성능은 8% 향상 시킨 것을 확인할 수 있다. 그러나, 더블버퍼링 기법이 정적복사, 락우선권 기법 모두와 같이 사용된 'F'의 경우 추가의 성능 향상이 없었다. 이는 응용 프로세서의 로컬 메모리 접근과 DMA의 전송이 시간 상으로 중첩되어 락우선권의 효과가 상쇄되기 때문인 것으로 분석된다.

5. 결론

이 논문에서는 모바일 내장형 시스템을 위한 DPS-DRAM 구조의 성능을 평가하였다. 또한, 무시할 수 없는 동기화 부하를 줄이기 위하여 락우선권 기법, 정적복사 기법, 공유 बैं크 분할의 세 최적화 기법을 제안하였다. DPSDRAM 구조의 효용성은 여러 실험을 통해서 검증 되었다. 기본 DPSDRAM 구조는 실제 전송 패턴을 고려한 실험에서 기존의 주변장치(LCD) 버스를 사용한 구조에 비해 4배 빠른 성능을 보여주었다. 더 나아가 제안한 세 가지 최적화 기법들을 적용함으로써 기본 DPSDRAM 구조에서 20-50%의 성능 향상을 얻을 수 있었다.

참고 문헌

- [1] T. V. Meeuwen et al, "System-level interconnect architecture exploration for custom memory organizations," in Proceedings of International Symposium on System Synthesis, pp. 13-18, Sep. 2001.
- [2] F. Gharsalli et al, "Unifying memory and processor wrapper architecture in multiprocessor SoC design," in Proc. ISSS, pp. 26-31, Oct. 2002.
- [3] K. Patel, E. Macii, and M. Poncino, "Synthesis of partitioned shared memory architectures for energy-efficient multi-processor SoC," in Proc. DATE, pp. 10700-10701, Feb. 2004.
- [4] Integrated Device Technology, "Dual port memory simplifies wireless base station design," Application Note, AN-409, Jan. 2004.
- [5] Samsung Electronics Inc, "Fusion Memory Solution OneDRAM™" http://www.samsung.com/PressCenter/PressRelease/PressRelease.asp?seq=20061213_0000306480.
- [6] K. Hayashi et al, "API000+: Architectural support of PUT/GET interface for parallelizing compiler," in Proc. ASPLOS, pp. 196-207, Oct. 1994.

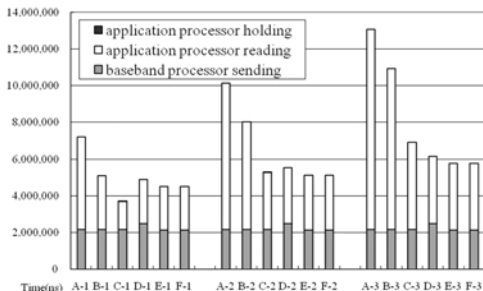


그림 6 공유 बैं크 분할 시 DPSDRAM 구조들의 성능 비교