

TMS320C80 프로세서에 적합한 Multimedia OS 사양에 관한 연구

장 석 우 · 박 인 규
홍익대학교 대학원 전기제어공학과
121-791 서울 마포구 상수동 72-1

A Study on specifications of Multimedia OS fitting with TMS320C80 Processor

Seock-woo Jang · In-gyu Park
Dept. of Electrical & Control Eng.,
Graduate School, Hong Ik University
72-1 Sangsu-dong Mapo-gu Seoul 121-791

0. 요약

본 논문은 Multimedia video processor로써 멀티미디어 데이터, 특히 비디오데이터 처리에 적합하도록 구성되어 있는 프로세서이다. TMS320C80에 Multimedia OS를 사용할 경우에 효율성의 타당성을 검증하기 위하여 영상 처리를 3가지 방법으로 수행시켜 그 결과를 비교한다. 채택한 영상처리로 DCT와 2차 Laplacian을 채택하였고 이를 적용하는 방법은 첫째, 일반적인 순차적으로 수행하는 방법과 둘째 기본으로 제공되는 kernel의 역할을 따르는 방법, 셋째로 OS모델을 따르는 경우의 방법으로 연산한다.

이 결과 첫번째 두번째 세번째 경우의 순서로 효율이 높은 결과를 얻었다. 이는 구현 방법이 복잡한 응용에 사용되어질 경우, OS모델이 우수할 것임을 반증한다.

이와 같은 결과를 토대로 TMS320C80에 적합한 task managing 부분의 OS kernel model을 제시한다.

1. 서론

TMS320C80은 칩내부에 RISC processor와 DSP processor를 4개 내장하고 있는 멀티프로세서이다. 이는 DSP의 단순 연산 능력과 RISC processor의 복잡한 algorithm 구현의 쉬운편을 동시에 구현하기 위해 이루어진 구조

이다. TMS320C80은 내부 커널을 제공하여 분리된 여러 개의 프로세서가 칩의 외부 또는 내부의 하드웨어에 접근하고자 할때 우선 순위를 결정하여 주게 된다. TMS320C80내부에 제공되는 커널은 RISC 프로세서에서 동작하며 DSP 프로세서나 RISC 내부의 다른 태스크 등을 관리하도록 되어 있다. 그러나 이 커널은 다른 프로세서에서 쓰이는 UNIX나 WIN95등의 일반 목적 OS 커널의 개념이 아니며 단순히 다중 태스크와 사용자 실행 모듈을 이어주는 역할을 하고 있는 것이다.

멀티미디어 데이터의 압축이나 복원 등에는 매우 많은 연산이 필요하면서도 구현 방법이 매우 복잡하고 가변적이다. TMS320C80은 이와같이 복잡하고 연산량이 많은 프로세싱에 적합한 구조로 되어 있다. TI사의 권고 사항에는 TMS320C80 프로세서의 기본 커널구조는 이 프로세서를 많이 사용한 multi processor 시스템에서도 시스템의 구조를 최적으로 이용할 수 있다고 주장한다. 그러나 이 커널은 다중 태스크와 사용자 실행 모듈을 이어주는 기본적인 틀만을 제공하는 것이지 OS의 측면을 적극적으로 지원하지는 않는다.

OS의 대표적인 예인 UNIX는 많은 사용자가 하나의 하드웨어를 사용하는 것을 목표로 하기 때문에 멀티태스킹 측면에서의 kernel이 매우 발달되어 있다. 또 현재 많이 활용되는 Windows나 Mac System 등과 같은 퍼

스런 컴퓨터들도 발달된 멀티태스킹 구조를 가지고 있다. 또한 linux OS 사용자가 많아진다는 것은 고급의 OS에 대한 필요성이 높아지고 있음을 의미한다.

이렇게 general purpose OS와 마찬가지로 realtime OS 또한 여러방면으로 발달하고는 있지만 Multimedia OS에 대해서는 그 경제와 활용도가 모호하기 때문에 적용되고 있는 사례는 드물다.

이 논문에서는 TMS320C80 프로세서 하나를 사용한 영상처리 보드에서 DCT와 2차 laplacian의 두가지 프로세싱의 효율을 비교한 커널 모델하에서 비교한다. 사용하는 커널 모델은 첫째, 일반적인 순차적으로 수행하는 방법과, 둘째, 기본으로 제공되는 kernel의 규약을 따르는 방법, 셋째로 보다 고기능의 OS모델 등의 3가지 모델을 제시하고, 상호 비교 검토한다. 이때 비교된 성능은 다른 멀티태스킹 호스트를 썼을때 본 프로세서를 얼마나 다양하고 효율적으로 사용할 수 있는가에 대한 결론을 유도한다.

2절에서는 이 연구를 수행한 하드웨어 기반과 내부구조에 대해 설명하고, 3절에서는 제시된 커널 모델과 그에 대한 실행방법에 대해, 4절에서는 그에 따른 결과 및 검토에 대해, 5절에서는 결론을 이끌어내고 향후 계획에 대해 언급한다.

2. 하드웨어 기반 및 내부 구조

사용한 TMS320C80 하드웨어는 다음과 같은 구조로 되어있다.

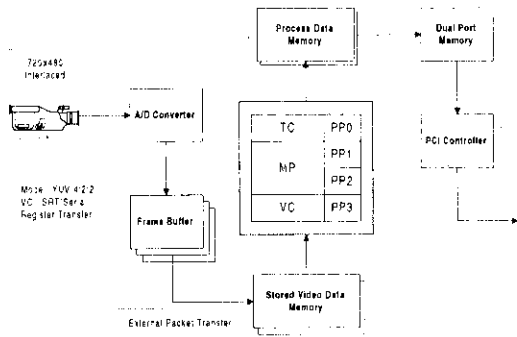


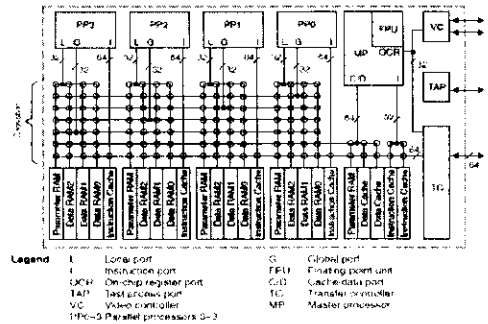
그림 1 TMS320C80 board block diagram

그림1의 구조를 간단히 설명하면 다음과 같다.

비디오 입력신호인 NTSC 신호가 A/D 컨버터를 거쳐 입력으로 들어오는 Interlaced NTSC 신호를 saa7111 (A/D converter)를 거쳐 MVP의 VC에 의해 vram으로 30필드/s로 저장하게 된다. 이때 데이터는 4:2:0 모드이며 이는 프로세싱을 위해 프로세서 내부의 TC(transfer

controller)의 관리하에 DRAM으로 저장된다.

프로세서의 내부구조는 다음 그림과 같고 이와 같은 crossbar 구조로 인해 내부의 5개 프로세서들이 서로 연결되어 돌아가게 된다. 이때 하드웨어 내부적으로 정의되어있는 base kernel에 의해 각각의 프로세서가 내부 메모리, 외부 메모리, 포트들 통해 resource에 접근을 하는 기본적인 방법을 제시한다.



그러나 본 논문에서는 모든 resource의 sharing을 manage하는 것에 관하여 논하지 않고 태스크의 실행, 즉 프로세서의 sharing에 대해서만 논의하려고 한다.

3. 커널 및 소프트웨어

이미지 프로세싱을 위해 기본적으로 수행하여야 할 루틴은 프로세서 내부의 4개의 DSP와 transfer controller, video controller등을 관리할 server routine이고, 항상 수행되어야 할 루틴은 비디오 데이터 입력부에서 비디오 데이터를 연속적으로 받아들이도록 하는 interrupt service routine이다. 이것들은 기본적으로 MP를 점유하게 된다.

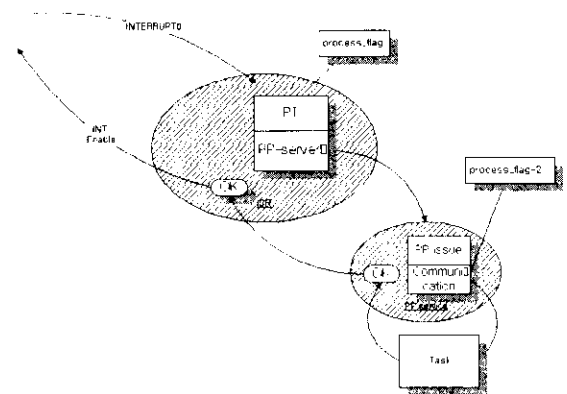


그림 3 Task 구동 block diagram

[그림 3]은 기본적으로 수행된 루틴의 블록도이다. 비교될 세가지의 다른 커널은 다음과 같이 구성한다. 첫째로, sequential한 방법으로 코딩하는 모델에서는 수행루틴을 것을 순차적으로 코딩한다. 둘째로, base kernel에 의존한 모델은 [그림 5]와 같은 구조를 갖는다.

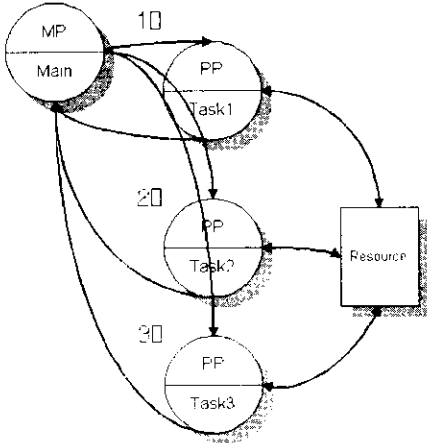


그림 4 Data flow of Base kernel

이 경우 태스크가 resource로 접근하고자 할 때에는 hardware에 의한 통제에 의해서만 움직이며 요청이 들어온 경우 바로 실행하고자 하도록 한다. 실행의 기본모듈인 task를 생성할 때 semaphore를 사용하여 생성하는 기본적인 태스크외에는 사용하지 않는다. 셋째로, OS 커널을 사용한 모델은 [그림 5]와 같은 구조를 갖는다.

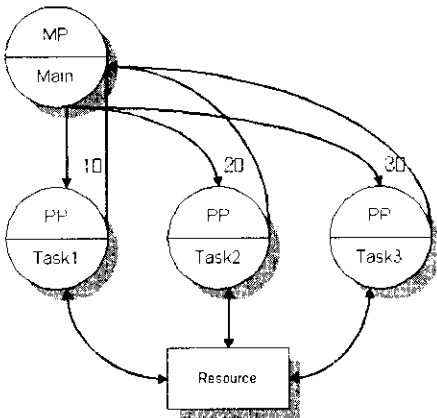


그림 5 Data flow of OS kernel

이 모델에서 태스크가 resource로 접근하고자 할때는

OS로 동작하는 server module의 제어에 의해서만 접근을 허용하게 된다. 영상처리라는 특정 목표를 갖고 있으므로 비선형으로 태스크를 스케줄링한다.

MVP 내부 커널의 TASK 구성 형식은 다음 그림과 같다. 하나의 포트에 들어가는 태스크들은 태스크 디스크 컴퓨터와 함께 물려있고 이는 순차적으로 처리되게 된다.

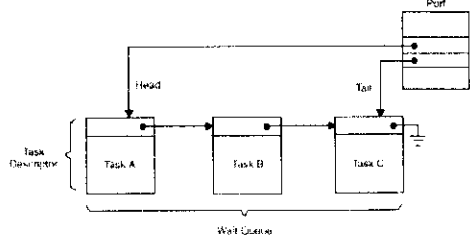


그림 6 Task Executing

DCT 연산은 [그림 7]과 같이 수행된다.

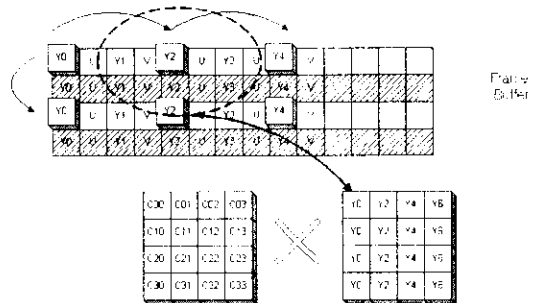


그림 7 Processing

1. 결과 및 검토

연산 결과는 연산 루틴으로 들어가는 부분부터 연산이 완전히 끝남을 알려주는 부분까지의 소요 연산 속도의 판단 기준으로 비교하였다.

이는 디버거에서 지원하는 실행 클럭수를 알려주는 benchmark running을 이용하였고 결과는 다음 그림과 같다.

[그림 8]에서 sequential execution, base kernel execution, OS kernel execution으로 연산하여 비교한 표를 보였으나 이는 부분적으로 linear하게 나타낸 표라서 차이가 많은 것 처럼 보이지만 사실상은 1000클럭 이내의 차이이므로 큰 차이는 아니다. 연산에 필요한 클럭수는 sequential execution, basic kernel execution, OS kernel execution의 순으로 나타났다. 단 이 수치는 디버

거에서 제공하는 커맨드에 따른다.

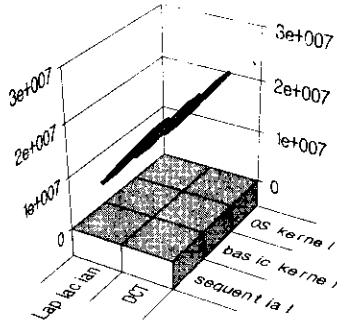


그림 8 Data chart

이러한 결과로 부터 가장 우수한 모델로 선정된 OS kernel은 basic kernel 구조를 활용하는 틀 안에서 구성되며 구조는 [그림 3]과 같다.

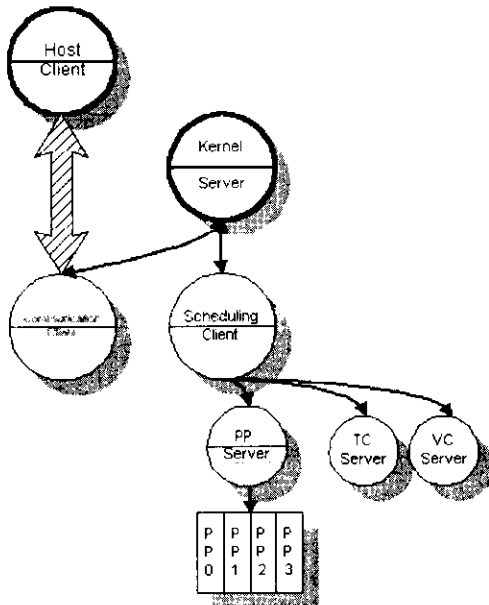


그림 9 OS structure

프로세서의 태스크를 실행하기 위한 스케줄링 콜라이언트 모듈과 호스트와의 통신 태스크를 목표로 하는 구조로 만들며 이 구조의 내부로는 base kernel을 이용한다.

5. 결론 및 향후 연구 과제

본 논문에서는 TMS320C80 멀티 프로세서의 작동을 극대화할 수 있는 멀티미디어 OS 커널의 효율성을 검증하였다. TMS320C80은 기본 하드웨어 커널이 프로세서 활용에 적합하게 구성되어 있는 고성능의 프로세서이다.

이를 활용하기 위한 OS 커널을 만든다면 호스트와의 통신 콜라이언트와 구별되는 프로세싱 모듈과 배너저 콜라이언트를 제어하는 메인 서버로 구성하는 것이 적합하다. 또 내/외부적인 port 등을 통한 메모리/버스 등의 접근은 제공되는 기본 base kernel을 이용하는 것만으로도 최적화가 된다.

TMS320C80 프로세서는 멀티미디어의 특성 및 구조적 활용에 적합하게 되어 있다. 향후 이러한 구조를 활용하여 개별 모듈로써 임의의 호스트에 적합하게 동작시킬 수 있는 OS 코어를 만들어 활용하는 것이 바람직한 것이다. 더 나아가 여러개의 TMS320C80 프로세서를 구동하는 커널에 관한 연구도 수행되어야 할 것이다.

6. 참고 문헌

- [1] 민용식,허기택 "시스템 프로그래밍" 대림 출판사 1993
- [2] RAO.HWANG Techniques & Standards For Image, video & Audio Coding 1996
- [3] TI TMS320C80 User's Guide 1996
- [4] 이분호,염재훈 "C언어를 이용한 영상신호 처리" 대영사