

FFT에 기반한 병렬 디지털 신호처리시스템의 성능분석

박 준 석*, 전 창 호*, 박 성 주*, 이 동 호*, 오 원 천**, 한 기 택**
(Joon Seok Park*, Chang Ho Jeon*, Sung Ju Park*, Dong Ho Lee*,
Won Chun Oh**, Ki Teak Han**)

요 약

본 논문에서는 방대한 양의 데이터를 실시간으로 처리하기 위한 병렬 디지털 신호처리시스템을 제안하고 성능을 분석한다. 병렬 디지털 신호처리시스템의 성능분석은 FFT를 대상으로 하여 FFT의 알고리즘에 기반한 성능분석, FFT 소스코드에 기반한 성능분석, 그리고 확률적 분석에 의한 성능분석 등 세 가지의 서로 다른 분석방법을 사용하였다. 성능분석 결과, FFT를 대상으로한 버스 기반의 시스템에서는 네 개의 프로세서로 구성된 단일 보드 시스템이 가장 우수한 성능을 보였다. 이것은 FFT처리 프로세서간 또는 보드간에 데이터 교환이 많은 응용 프로그램에서는 통신부담이 증가하여 시스템의 전체 성능을 저하시키기 때문이다. 그러나 보드간 연결에 사용하는 버스의 수가 늘어남에 따라 시스템의 성능도 선형적으로 증가함을 알 수 있었다.

ABSTRACT

This paper concerns performance of a parallel digital signal processing system. The performance of the system is analyzed in terms of CPU cycles required for 1024-point FFT computation. The number of cycles is estimated in three different approaches; FFT algorithm-based, assembly level source code-based, and probability-based.

The results of analysis indicate that on a bus-based system the best performance for FFT is achieved with a single board. Because in some applications like FFT, where frequent data exchanges among processors occur, the number of communication cycles increases as the number of boards. It is observed that inter-board communication degrades overall system performance for the FFT computation. Also shown is that linear increase in performance can be obtained if multiple buses are employed.

I. 서 론

반도체 기술이 발달함에 따라 디지털 신호처리나 영상 처리 등에 특별히 우수한 성능을 나타내는 DSP칩들이 많이 개발되었다. 그러나 단일 DSP칩 만으로는 방대한 데이터를 실시간으로 처리하기가 힘들다. 따라서 최근에는 처리율과 신뢰성 및 확장성을 높이기 위해 DSP칩을 다중으로 연결한 병렬시스템 구조가 DSP 응용분야에 많이 사용되고 있다¹⁾. Paisley대학에서는 transputer를 기반으로 한 음성신호 처리시스템을 개발하였고¹⁾ 영상처리 분야에서는 TMS320C25들을 병렬로 연결하여 영상처리부를 구성한 DSP 병렬시스템이 Oakland 대학에서 개발되었으며²⁾ 일본의 NTT에서는 2048*2048 해상도의 영상(SHD image: Super High Definition image)을 처리하기 위한 병렬시스템이 개발된바 있다³⁾. 또한 터키의 이스탄

불 공과대학에서는 TMS320C25에 기반한 교육용 병렬시스템을 개발하였고⁴⁾ 시스템과 응용프로그램의 개발 초기 단계에서 미리 원형(prototype)을 설계하여 개발시간과 비용을 절감시키기 위한 병렬 DSP 시스템이 개발되기도 하였다⁵⁾. 이처럼 디지털 신호처리 응용분야에서는 고속으로 데이터를 처리하기 위하여 병렬로 시스템을 구성하고 있는 추세이다. 더욱이 대잠 소나체계와 같이 엄격한 실시간 처리를 보장해야 하는 응용분야에서는 병렬 시스템의 필요성은 한층 크다고 할 수 있다.

본 논문에서는 상용화된 DSP칩을 이용하여 고속의 신호처리 응용분야에 적합한 병렬 디지털 신호처리시스템을 제안하고, 신호처리 응용분야의 대표적인 연산으로서 대량의 산술연산을 필요로 하는 FFT에 대하여 제안된 시스템의 성능을 분석한다.

병렬 신호처리시스템의 제안 및 성능분석에 관한 본 논문은 다음과 같이 구성되어 있다. 먼저 2절에서는 일반적인 병렬시스템의 구조와 동작방식을 간단히 기술하고, 상용으로 개발된 많은 DSP칩 중에서 현재 가장 많이 사

* 한양대학교

** 국방과학연구소

접수일자: 1998년 7월 22일

용되는 TI사의 TMS320C40에 대해서 살펴본다. 3절에서는 TMS320C40을 사용하여 신호처리 응용분야에 뛰어난 성능을 발휘할 수 있도록 구성된 병렬 디지털 신호처리 시스템을 제안하고 4절에서는 제안한 시스템의 성능을 분석하기 위하여 사용하는 FFT연산의 병렬처리 기법에 대해서 기술한다. 5절에서는 제안한 시스템의 성능을 다양한 방법으로 분석하고 6절에서 결론을 맺는다.

II. 병렬시스템 구조 및 TMS320C40

병렬시스템 구조는 주어진 문제를 보다 빠르고 정확하게 수행하기 위해 여러개의 연산소자를 사용하여 작업의 동시실행을 가능하게 하는 컴퓨터시스템 구조이다. 병렬처리를 위한 시스템의 구조는 보통 Flynn의 분류법에 따라 SIMD(Single Instruction, Multiple Data stream)와 MIMD(Multiple Instruction, Multiple Data stream)로 대분한다⁶⁾.

SIMD와 MIMD 같은 병렬시스템을 설계하는데 있어서 가장 중요한 문제 중의 하나는 프로세서간 또는 프로세서와 메모리 모듈들을 어떻게 연결하느냐하는 문제이다. 많이 사용되고 있는 상호연결망으로는 다중버스, crossbar 스위치, 하이퍼큐브, 다단계 연결망 등이 있으며^{7), 8)} 제안된 시스템에서는 버스를 사용한다.

일반적인 병렬시스템에 사용되는 처리장치는 범용으로 개발된 상용 마이크로프로세서이다. 그러나 신호처리와 같이 대단위의 산술연산 및 데이터의 실시간 처리가 필요한 분야에서는 상용 마이크로프로세서는 그 요구조건을 만족시키기가 어렵다. 따라서 신호처리 분야의 이러한 특성을 고려할 때, 신호처리를 병렬시스템의 처리장치로는 DSP칩을 사용하는 것이 바람직하다.

현재 많은 DSP칩들이 개발되어 있지만 그 중에서도 뛰어난 성능뿐만 아니라 상용 시스템에서 널리 사용되고 있어 개발환경이 우수한 TI사의 TMS320C40을 본 논문에서 제안하고자 하는 병렬시스템의 처리장치로 채택하였다.

TI사의 TMS320C40은 병렬처리를 지원하는 32-bit 부동소수점 DSP칩^{9), 10)}으로 40MIPS/80MFLOPS의 데이터 처리능력을 가지고 있다. TMS320C40이 병렬처리를 지원하기 위한 구조로는 프로세서와 프로세서간에 쌍방향의 데이터를 전송할 수 있는 6개의 통신포트를 가지고 있으며 각각의 포트는 20 Mbytes/s의 전송율을 가진다. 그리고 메모리에 대한 데이터 입출력과 산술연산을 동시에 처리하기 위한 6개의 채널을 가진 DMA 보조프로세서가 있다. 또 TMS320C40은 두개의 독립된 외부 버스를 가지고 있어서 서로 다른 메모리에 동시에 접근할 수 있다. 이렇게 TMS320C40의 병렬처리 능력은 DSP칩의 통신포트, 버스, 그리고 DMA 보조프로세서 등을 통하여 가능하게 된다.

또한 CPU를 구성하는 모듈로는 단일 사이클 동안 IEEE 부동소수점 변환과 나눗셈 및 2중근의 역산을 할 수 있는 ALU와 단일 사이클안에 40-bit 곱셈연산을 할

수 있는 승산기가 있으며 circular 및 bit-reversed address를 생성할 수 있는 두개의 ARAU 회로와 32개의 레지스터들로 구성되어 있다.

III. 병렬 디지털 신호처리시스템 설계

그림 1은 본 논문에서 제안하는 병렬 디지털 신호처리 시스템의 전체적인 구성도이다. 전체 시스템은 각각의 기능에 따라 신호처리부, 공용메모리부, 데이터 입출력부, 그리고 중앙제어부 등 크게 네 개의 subsystem으로 구성되어 있으며 전반적인 동작은 다음과 같이 이루어진다. 먼저 센서를 통해 들어온 입력 데이터는 데이터 입출력부에서 필요한 전처리과정을 통하여 적절한 형태로 변환되어 공용메모리부로 전송된다. 신호처리부에서는 공용메모리부와 중앙제어부로부터 제공받은 데이터와 명령문을 고속으로 처리한다. 각 subsystem 상호간의 동작은 중앙제어부에 의해 제어된다. 시스템 전체의 공용메모리는 보드간 또는 서로 다른 기능적 모듈간의 통신과 데이터 교환에 사용된다.

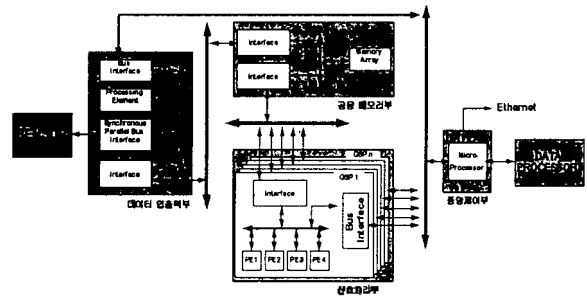


그림 1. 병렬 디지털 신호처리시스템의 구성도
Fig. 1. An organization of parallel digital signal processing system.

그림 2는 병렬 디지털 신호처리시스템의 여러 subsystem들 중 가장 핵심적인 부분인 신호처리부의 구성도이다. 그림에서처럼 하드웨어의 전체적인 구조는 보드 단위로 확장이 가능하다. 각 보드에는 4개의 TMS320C40이 통신포트를 통하여 완전연결망 형태로 서로 연결되어 있다.

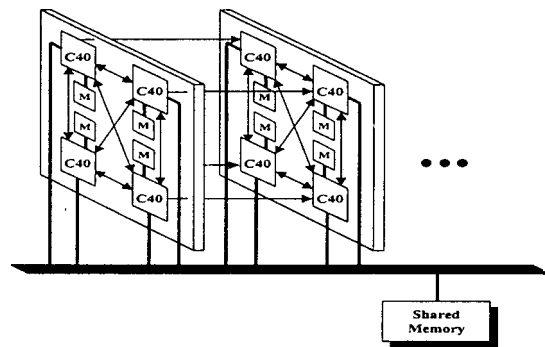


그림 2. 신호처리부의 하드웨어 구성도
Fig. 2. Block diagram of signal processing subsystem.

또 보드들은 공유버스와 통신포트를 통하여 이중으로 연결된다. 그리고 각 DSP는 다른 프로세서에 전달될 필요가 없는 지역데이터의 처리나 프로그램의 저장에 사용하는 지역메모리를 가지고 있다.

하나의 보드를 4개의 TMS320C40로 구성한 것은 신호 처리 분야에서 수행하는 여러 가지 연산의 특성상 프로세서의 개수는 2의 멱승이 가장 적합하고, TMS320C40이 6개의 통신포트를 가지고 있기 때문에 8개 이상의 프로세서를 한 보드에 장착할 경우 완전연결망 형태로 프로세서들을 연결할 수가 없기 때문이다. 또한 2개의 프로세서로 한 보드를 구성할 경우, 사용되지 못하는 통신포트의 수가 늘어나 자원의 활용도 측면에서 비효율적인 뿐만 아니라 보드당 얻을 수 있는 성능이 미약하여 실시간 처리가 힘들기 때문이다.

많은 상용 DSP칩 중에서 TI사의 TMS320C40을 제안한 시스템의 연산소자로 선택한 이유를 간단히 요약하면 첫째로, TMS320C40의 뛰어난 부동소수점 연산 능력 때문이다. TMS320C40은 초당 6천만 번의 부동소수점 연산을 수행할 수 있으며 또한 한 사이클 안에 곱셈을 수행할 수 있는 능력을 가지고 있다. 이것은 부동소수점 곱셈 연산이 많은 신호처리의 특성에 가장 중요한 요소라고 할 수 있다. 둘째로, TMS320C40은 6개의 통신포트를 가지고 있어 동일한 보드내의 다른 DSP 칩은 물론 이웃한 다른 보드와의 직접적인 연결과 통신기능을 제공하기 때문에 병렬시스템을 구현하기가 아주 용이하다. 셋째로, TMS320C40은 칩 내부에 6 개의 DMA 채널을 가지고 있어 메모리와의 빠른 데이터 전송이 가능하며, 초당 100MByte의 대역폭을 가지는 두 개의 버스(global, local)를 통하여 외부메모리와의 다양하게 연결할 수 있다. 이외에도 TMS320C40은 칩 내부에 프로그램 메모리와 데이터 메모리를 가지고 있다¹¹⁾.

IV. FFT의 병렬처리

이 절에서는 여러 종류의 FFT중 1-D FFT의 병렬처리에 대해서 간략히 살펴보기로 한다.

FFT의 종류는 radix와 신호를 재배열하는 과정에 따라 분류되는데, 입력신호의 재배열 과정이 연산의 처음일 경우(Decimation-in-Time)와 FFT 연산이 끝난 후에 결과 신호를 재배열하는 경우(Decimation-in-Frequency)에 따라서 Radix-2 DIT(DIF) FFT, Radix-3 DIT(DIF) FFT등으로 나뉜다¹²⁾.

그림 3은 16-포인트 radix-2 DIT FFT 알고리즘에 대한 흐름도이다¹³⁾. 이와 같은 16-포인트 FFT를 4개의 프로세서를 이용하여 병렬로 연산하기 위해서는 우선 그림 4.와 같이 순차적으로 들어오는 신호 $x(i)$ 를 재배열하는 작업이 필요하다.

그림 4에서와 같이 16개의 이산입력 신호는 각각의 프로세서에 4개씩 입력되게 되며 프로세서의 수가 p 이고 N 개의 FFT를 연산할 때 프로세서 i 가 입력받는 데이터는 $i+j*p$ 가 된다. 여기서 $0 \leq j \leq (N/p), 0 \leq i < p$ 이다.

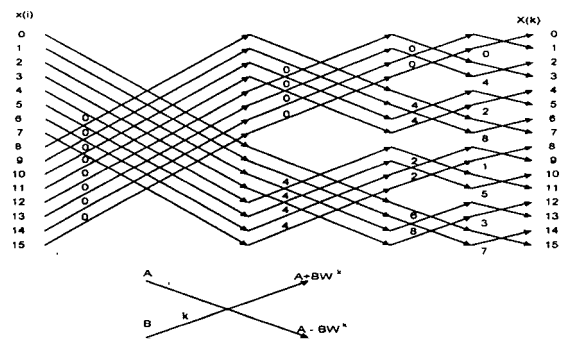


그림 3. 16포인트 DIT FFT의 flow chart
Fig. 3. Flow chart of 16-point DIT FFT.

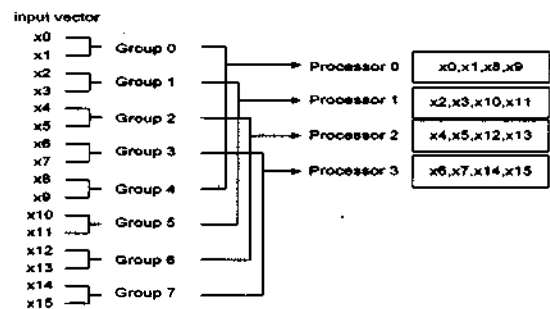


그림 4. FFT연산을 위한 입력 신호의 재배열
Fig. 4. Reordering of input signals for FFT computation.

FFT 알고리즘은 전체적으로 $\log_2 N$ 에 해당하는 stage를 가지게 되며 데이터 분할에 의한 병렬처리 방법을 사용할 경우에는 처음 $\log_2 N - \log_2 p$ 의 stage에서는 연산의 중간 값들을 프로세서간에 교환할 필요가 없으나 $\log_2 p$ 만큼의 stage에서는 프로세서간에 중간 연산결과를 교환해야 한다. 각 stage에서 프로세서의 수 p 로 분배된 FFT의 butterfly가 수행되는 병렬처리 과정을 그림 5에 도식적으로 나타내었다. 그림에서 step 0,1은 프로세서간 중간 연산값들의 교환이 필요하지 않은 stage를 나타내며, step2와 3은 중간 연산 값들이 프로세서들 간에 교환이 필요한 경우를 나타낸다.

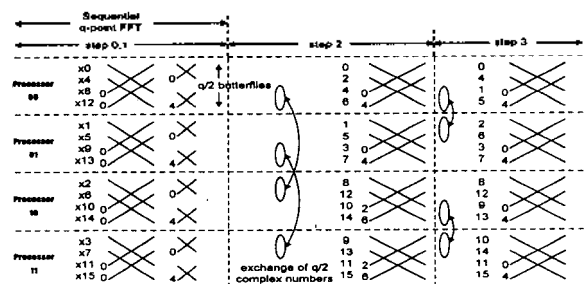


그림 5. 4개의 프로세서를 이용한 16-포인트 DIT FFT 알고리즘
Fig. 5. 16-point DIT FFT algorithm with four processors.

V. 성능분석

병렬 디지털 신호처리시스템의 성능을 분석하기 위하

여 본 논문에서는 1024-포인트 FFT연산을 수행하는데 소요되는 CPU 사이클을 세 가지의 방법으로 계산한다. 이러한 세 가지의 성능분석 방법은 FFT 알고리즘에 기반한 성능분석, 소스코드 분석에 기반한 성능분석, 그리고 확률적 분석에 기반한 성능분석이다.

보드간 통신은 버스를 통한 공유메모리를 이용하여 이루어지고, 보드 내에서의 프로세서간 통신은 통신포트를 통하여 이루어진다. 또 전체 계산량이 모든 보드에 균등하게 분배된다고 가정한다.

5.1 FFT 알고리즘에 기반한 성능분석

제한한 시스템에서 FFT연산을 수행할 경우 시스템의 성능에 주된 영향을 미치는 성능요소들로는, 메모리 접근, 프로세서간 통신, 보드간 통신, 곱셈연산, 그리고 덧셈연산 등 크게 다섯 가지가 있다. 따라서 FFT 알고리즘에 기반한 성능분석에서는 이러한 성능요소들의 연산횟수를 구하고, 각각의 단위 연산에 소요되는 CPU 사이클 수를 곱함으로써 시스템이 처리할 수 있는 속도를 sampling rate의 단위로 구한다.

성능분석을 위해서 필요한 파라미터들에 대한 정의는 다음과 같다.

- B : 보드의 수
- N : FFT의 포인트 수
- P_B : 보드당 프로세서 수

위에서 정의한 파라미터들을 이용하여 complex FFT 알고리즘이 병렬 디지털 신호처리시스템에서 수행될 경우 필요한 성능요소들의 연산횟수를 수학적으로 모델링 하면 식(1)과 같다.

$$\left(\frac{2N \log_2 N}{B P_B}\right) + \left(\frac{2N \log_2 N}{B P_B}\right) + \left(\frac{2N \log_2 N}{B P_B}\right) + \left(\frac{N \log_2 P_B}{B P_B}\right) + \left(\frac{N \log_2 B}{2}\right) \quad (1)$$

식(1)에서처럼 연산은 크게 다섯 항으로 이루어져 있는데, 각각은 곱셈연산의 횟수, 덧셈연산의 횟수, 메모리 접근의 횟수, 프로세서간 통신의 횟수, 그리고 보드간 통신의 횟수이다. 여기서 $\log_2 N$ 는 N-포인트 FFT 연산을 수행할 때 필요한 stage 수, $\log_2 B$ 는 보드간 통신이 필요한 stage 수, 그리고 $\log_2 P_B$ 는 한 보드내에서 프로세서간 통신이 필요한 stage 수이다.

곱셈, 덧셈, 메모리 접근, 프로세서간 통신, 그리고 보드간 통신 등 각 연산에 필요한 사이클 수를 C_{mul} , C_{add} , C_{mem} , C_{ipc} , 그리고 C_{ibc} 라 할 때, N-포인트 complex FFT 수행에 소요되는 전체 사이클 수는 식(2)와 같이 된다.

$$\left(\frac{2N \log_2 N}{B P_B}\right) C_{mul} + \left(\frac{2N \log_2 N}{B P_B}\right) C_{add} + \left(\frac{2N \log_2 N}{B P_B}\right) C_{mem} + \left(\frac{N \log_2 P_B}{B P_B}\right) C_{ipc} + \left(\frac{N \log_2 B}{2}\right) C_{ibc} \quad (2)$$

도출된 식(2)를 사용하여 TMS320C40으로 구성된 하드웨어 구조에서 FFT를 수행할 경우 시스템이 나타내는 최대 성능을 구할 수 있다. TMS320C40은 한 사이클 동안에 덧셈 또는 곱셈을 수행할 수 있기 때문에 식(2)로부터 실제적인 성능수치를 얻기 위해 사용한 C_{mul} 과 C_{add} 의 값은 1이 된다.

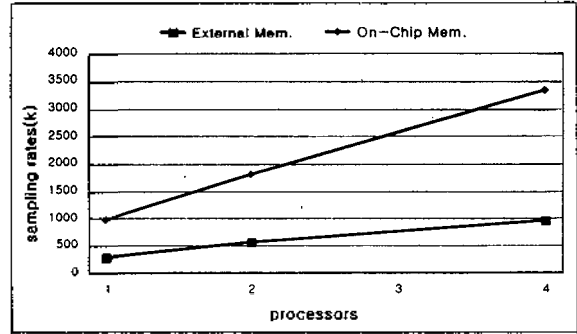


그림 6. 단일 보드에서의 FFT 알고리즘에 의한 성능분석 결과
Fig. 6. Results of FFT algorithm-based analysis for single board system.

먼저 그림 6은 단일 보드에서 프로세서의 증가에 따른 성능의 변화를 보여준다. 그림에서처럼 사용하는 메모리에 따라 증가량의 차이는 있지만 모두 프로세서의 수가 증가할수록 sampling rate가 증가하는 것을 알 수 있다. 단일 보드에서는 보드간 통신이 발생하지 않으므로 보드간 통신에 의한 성능저하는 없다.

그리고 다중 보드를 사용한 경우의 분석결과는 그림 7과 같다. 보드의 수가 증가할수록 그리고 보드간 통신에 소요되는 단위 사이클의 수가 클수록 보드간 통신에 의한 부하가 커져서 전체적인 시스템의 성능이 저하됨을 알 수 있다.

FFT의 알고리즘에 기반하여 성능분석을 한 결과를 종합해 보면, 보드의 수가 증가할수록 시스템의 전체적인 성능은 오히려 떨어짐을 알 수 있다. 이것은 보드간 통신이 시스템의 전체적인 성능에 큰 영향을 미치는 요인임을 말해 준다. 따라서 신호처리부의 설계는 되도록 보드간 통신이 적게 하여야 한다. 또한 하나의 보드를 구성하는 프로세서의 수는 4개가 가장 적합함을 확인하였다.

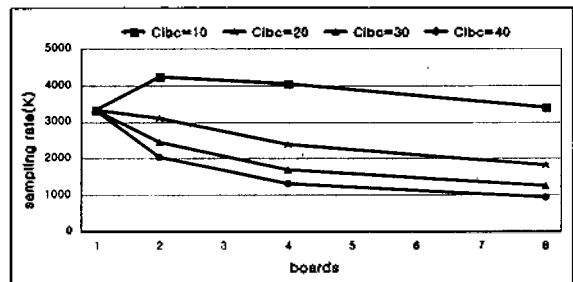


그림 7. 다중 보드에서의 FFT 알고리즘에 의한 성능분석 결과
Fig. 7. Results of FFT algorithm-based analysis for multiple board system.

5.2 FFT 소스코드 분석에 기반한 성능분석

TI사에서는 TMS320C4X의 다양한 응용과 필요한 개발 환경을 제공하기 위해 TMS320C4X를 이용한 구현 예제들을 개발하여 공개하였다. 그 중의 하나가 *Parallel Processing with the TMS320C4X¹³⁾* 이다 이 자료에는 TMS320C4X를 이용한 병렬시스템에서 FFT를 병렬로 처리하기 위한 C 언어 및 Assembly 레벨의 소스코드가 포함되어 있다. 본 논문에서는 이러한 소스코드의 분석을 통해 Radix-2 DIT FFT를 병렬처리로 구현하기 위한 연산을 모델링 함으로써 성능분석을 하였다.

명령어 사이클은 TMS320C4X User's Guide⁹⁾를 근거로 했다. 소스코드의 분석에 의하면 곱셈과 덧셈이 가장 많으며 이 두 개의 명령어는 TMS320C40이 지원하는 병렬처리 기법에 따라 한 사이클에 수행될 수 있다. 또한 FFT 알고리즘의 특성상 중간 연산결과를 프로세서간에 교환이 필요하게 된다. 이 때 TMS320C40의 통신포트를 통한 데이터 전송은 5 사이클, 메모리를 통한 데이터 전송은 16 사이클이 필요하다고 가정한다.

소스코드는 네 개의 어셈블리 함수들로 구성되어 있다. 첫째, *cmove.asm*이다. 이 함수는 외부 메모리로부터 계산할 데이터들을 칩 내부메모리에 저장하기 위한 것이며 P개의 프로세서를 이용하여 N개의 데이터를 처리하기 위해서 1개의 프로세서가 실행해야할 사이클 수는 $28 + 16 * \frac{N}{P_T}$ 이다. P_T 는 연산에 사용된 모든 프로세서의 수이다. 둘째, *r2dit.asm*은 그림 5와 같이 FFT를 병렬로 처리하는 과정 중에서 프로세서와 프로세서간의 데이터 교환이 필요치 않은 stage에서 FFT를 실행하는 프로그램이다. 이 때 필요한 사이클 수는 $26 + 14.5 * \frac{N}{P_T}$ 이다. 셋째, *exch_r.asm*은 통신포트를 통해서 데이터를 교환하기 위한 코드이며 사이클 수는 $29 + 20 * \frac{N}{P_T}$ 가 된다. *bfly.asm*은 프로세서간에 교환된 데이터를 이용하여 FFT의 butterfly를 실행하는 과정에서 필요한 코드이다. 사이클 수는 $29 + 9 * \frac{N}{P_T}$ 이다. 따라서 N-포인트 FFT를 실행하기 위한 총 사이클 수는 프로세서 내부에서 실행되는 사이클 수인 *r2dit.asm*, 프로세서간 통신이 발생하는 stage에서의 사이클 수인 [*exch_r.asm*+*bfly.asm*], 그리고 보드간 통신이 발생하는 stage에서의 사이클 수인 [*cmove.asm*+*bfly.asm*]를 모두 합한 것이 된다.

소스코드 분석에 의한 방법으로 N-포인트 FFT를 제안된 시스템에서 수행할 경우 소요되는 전체 CPU 사이클을 수식으로 정리하면 식(3)과 같다.

$$(14.5 * \frac{N}{P_T} + 26) + (58 + 29 * \frac{N}{P_T}) * \log_2 P_B + (28 * P_T + 16 * N + 29 + 9 * \frac{N}{P_T}) * \log_2 B \quad (3)$$

그림 8은 1K, 2K, 4K, 8K 크기의 FFT를 수행할 때 보드 수의 증가에 따른 성능변화를 나타낸 것이다. 그림에서처럼 보드의 수가 증가함에 따라 보드간 통신 횟수가 증가하게 되고 이것은 FFT 수행에 필요한 전체 CPU 사이클의 증가를 초래한다. 따라서 보드간 통신에 의한 통신비용의 증가로 시스템의 성능이 저하됨을 알 수 있다.

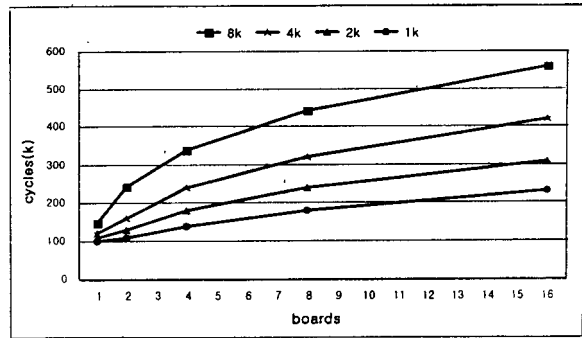


그림 8. 다중 보드에서의 FFT 소스코드 분석에 기반한 성능분석 결과
Fig. 8. Results of source code-based analysis for multiple board system.

5.3 확률적 분석에 기반한 성능분석

확률적 분석방법을 사용하여 성능분석을 할 경우 필요한 파라미터들의 정의는 표 1과 같다.

표 1. 확률적 분석방법에서 사용되는 파라미터들의 정의
Table 1. Parameters used in probability-based analysis.

BW	전체 버스 시스템의 대역폭 (bytes/sec)
BM	단위 시간당 성공적으로 액세스 된 메모리 모듈의 개수
B_S	증대지연을 포함한 버스의 물리적인 대역폭 (bytes/sec)
B	시스템의 전체 버스의 개수
B_T	시스템의 전체 보드의 개수
r	프로세서의 평균 메모리 요청 확률
M	시스템의 전체 메모리 모듈의 개수
M_S	메모리 모듈의 처리속도 (bytes/sec)
P_S	시스템의 성능
P_P	시스템의 최대 성능
C_B	한 바이트의 데이터를 처리하는데 걸린 사이클 수
O_C	하나의 시스템 사이클에 수행될 수 있는 메모리 액세스의 수
C_T	프로그램을 실행하는데 걸린 전체 사이클 수

확률적 분석에 기반한 성능분석 방법에서는 P_T 개의 프로세서, B 개의 버스, M 개의 메모리 모듈로 구성된 다중 버스 시스템에서 프로세서의 평균 메모리 요청 확률이 r 일때 전체 시스템의 버스 대역폭을 구한다.

식(4)는 j 번째 메모리가 액세스될 확률을 나타내며 식(5)는 i 개의 메모리가 액세스될 확률을 나타내고 식(6)은 액세스될 메모리 모듈의 평균 개수를 나타낸다. 여기서 전체 버스 시스템의 대역폭은 식(7)과 같이 메모리와 버스 중 더 느린 것에 의하여 결정된다. 버스 대역폭 결정에 핵심요소인 r 값은 시스템을 응용하는 분야에 따라서 다르며 FFT인 경우는 FFT 소스코드 분석을 바탕으로 식(8)과 같이 산출된다.

$$P[M_j] = 1 - \left(1 - \frac{r}{M}\right)^{P_T} \quad (4)$$

$$f(i) = {}_M C_i \cdot P[M_j] \cdot (1 - P[M_j])^{M-i} \quad (5)$$

$$BM = \sum_{i=1}^B i \cdot f(i) + \sum_{i=B+1}^M B \cdot f(i) \quad (6)$$

IF BS > MS THEN

$$BW \text{ (bytws/sec)} = MS \cdot BM$$

ELSE

$$BW \text{ (bytws/sec)} = BS \cdot BM$$

$r =$

$$\frac{P_T \left(28 + 16 \frac{N}{P_T} \right) \log_2 B_T}{\left(14.5 \frac{N}{P_T} + 26 \right) + \left(58 + 29 \frac{N}{P_T} \right) \log_2 P_B + \left(29 + 9 \frac{N}{P_T} + 28 P_T + 16 N \right) \log_2 B_T} \quad (8)$$

$$P_S = (1-r)P_P \cdot de + C_B \cdot BW \cdot O_C \quad (9)$$

$$de = \frac{(C_T - C_{ibc}) - \frac{4}{5} C_{ibc}}{C_T - C_{ibc}} \quad (10)$$

산출된 r 값에 따라서 전체 시스템의 성능 P_S 는 프로세서 내부 처리 성능과 버스를 통한 성능값의 합으로 식(9)와 같이 정리되며 프로세서간의 통신에 기인한 성능저하는 식(10)과 같이 고려하였다.

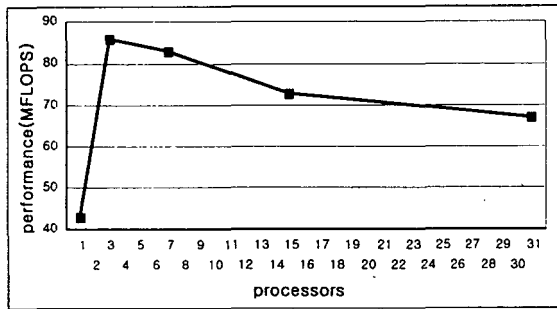


그림 9. 단일 보드에서의 확률적 분석에 의한 성능분석 결과
Fig. 9. Results of probability-based analysis for single board system.

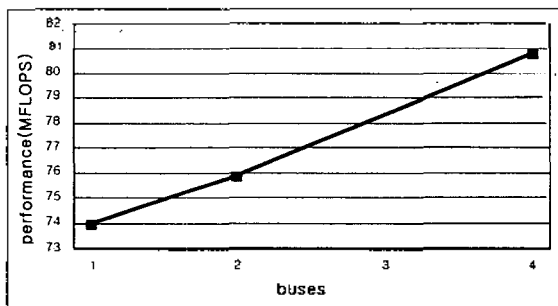


그림 10. 다중 버스를 사용할 경우의 성능분석 결과
Fig. 10. Results of probability-based analysis for multiple bus system.

이러한 수학적 모델링 및 평균메모리요청 확률계산에 근거한 시스템 전체의 처리속도는 그림 9 및 그림 10과 같다. 그림 9는 1개의 메모리 모듈 및 1개의 버스로 구성된 시스템에서 1024-포인트 FFT를 수행할 경우 프로세서의 개수에 따른 처리속도의 변화를 나타낸다. 프로세서가 2개에서 4개로 증가될 때는 속도가 증가하지만 프로세서가 더욱 증가함에 따라 보드간 통신량의 증대로 전

체 시스템의 속도는 감소함을 알 수 있다. 그림 10은 16개의 프로세서와 4개의 메모리로 구성된 시스템에서 1024-포인트 FFT 수행시 버스 개수가 증가함에 따라서 전체 시스템의 처리속도는 선형적으로 증가함을 보여 준다.

5.4 성능분석 결과

FFT 알고리즘에 기반한 성능분석, FFT 소스코드 분석에 기반한 성능분석, 그리고 확률적 분석에 의한 성능분석 등 세 가지의 성능분석 방법을 사용하여 본 논문에서 제안한 병렬 디지털 신호처리시스템의 성능을 분석한 결과 서로 근접한 결과를 얻었으며 이것은 병렬 디지털 신호처리시스템의 성능분석이 타당함을 보여준다. 먼저 그림 6과 같이 단일보드 내에서는 프로세서의 수가 증가함에 따라 전체적으로 시스템의 성능이 향상됨을 알 수 있었다. 그러나 그림 9에서처럼 프로세서의 수가 4개 보다 많을 경우 TMS320C40이 가지고 있는 6개의 통신포트로는 모든 프로세서들을 완전연결망 형태로 연결할 수 없으며 이것은 프로세서간 통신지연을 급격히 증가시키게 되고 따라서 시스템의 전체적인 성능은 저하하게 된다.

또한 그림 7과 그림 8에서 알 수 있는 바와 같이 보드의 수가 증가함에 따라 보드간 통신을 위해서 사용되는 버스의 병목현상으로 인하여 시스템의 전체적인 성능은 저하된다. 그러나 그림 10에서처럼 버스의 수를 증가시켜 주변 보드간 통신에 필요한 지연시간이 감소하게 되어 시스템의 전체적인 성능은 향상된다.

이와 같은 결과로부터 하나의 보드는 TMS320C40의 중요한 자원인 통신포트를 최대한으로 활용하고, 프로세서의 증가에 선형적인 성능향상을 얻을 수 있도록 4개의 프로세서로 구성하는 것이 가장 적합함을 알 수 있다. 이것은 본 논문에서 제안한 병렬 디지털 신호처리시스템의 신호처리부 구성이 타당함을 입증해 준다. 그리고 보드간 연결은 보드간 통신지연을 줄일 수 있도록 다중 버스를 사용하여야 함도 알 수 있다.

VI. 결론 및 향후계획

본 논문에서는 상용화된 DSP칩을 이용하여 고속의 신호처리 응용분야에 사용할 수 있는 병렬 디지털 신호처리시스템을 제안하였다. 그리고 제안된 시스템의 성능을 분석하기 위하여 신호처리 응용분야의 대표적인 연산이며 대량의 산출연산을 필요로 하는 FFT를 대상으로 하여 알고리즘에 의한 방법, 소스코드 분석에 의한 방법, 그리고 확률적 분석에 의한 방법 등 세 가지의 성능분석 방법을 사용하였다. 성능분석 결과 제안된 시스템은 대장 소나체계와 같이 실시간 처리가 요구되는 응용분야의 요구조건을 충분히 만족시키는 성능을 나타냄을 알 수 있었다.

그리고 시스템의 성능에 가장 큰 영향을 미치는 보드간 통신지연을 최소화 할 수 있는 고속버스 및 다중버스의 사용과 TMS320C6x와 같은 차세대 고성능 DSP칩을 이용하여 시스템을 구체적으로 설계하고 그 시스템의 성능을 평가하는 연구를 계속 수행중에 있다.

참 고 문 헌

1. A. J. Anderson, "Selection Criteria in the Development of a Multiple Processor Based DSP System," *Journal of Microcomputer Application*, pp. 327-345, 1992.
2. P. V R Raja and S. Ganesan, "An SIMD Multiple DSP Microprocessor System for Image Processing," *Microprocessors and Microsystems*, pp. 493-501, Nov. 1991.
3. T. Sawabe, T. Fujii, H. Nakada, N. Ohta, and S. Ono, "A 15GFLOPS Parallel DSP System for Super High Definition Image Processing," *IEICE Trans. Fundamentals*, Vol.39, No.7, pp. 786-792, Jul. 1992.
4. F. Kurugollu, H. Palaz, H. Gumuskaya, E. Harmanci and B. Orencik, "Advanced Educational Parallel DSP System Based on TMS320C25 Processors," *Microprocessors and Microsystems*, Vol. 19, No. 3, pp. 147-156, Apr. 1995.
5. M. Engels, R. Lauwereins and J. Peperstraete, "Rapid Prototyping for DSP Systems with Multiprocessors," *IEEE Design & Test of Computer*, pp. 52-62, Jun. 1991.
6. K. Hwang, *Advanced Computer Architecture: Parallelism, Scalability, Programmability*, McGraw-Hill, 1993.
7. T. Feng, "A Survey of Interconnection Networks," *IEEE Computer*, Vol. 14, No. 12, pp. 12-27, Dec. 1981.
8. L. N. Bhuyan, Q. Yang and D. P. Agrawal, "Performance of Multiprocessor Interconnection Networks," *IEEE Computers*, Vol. 22, No. 2, pp. 25-37, Feb. 1989.
9. *TMS320C4X User's Guide*, Texas Instruments, 1996.
10. *TMS320C3X/4X Optimizing C Compiler*, Texas Instruments, 1996.
11. R. O. Nelsen, *Sonar Signal Processing*, Artech House, 1991.
12. W. W. Smith, J. M. Smith, *Handbook of Real-Time Fast Fourier Transform*, IEEE Press, 1995.
13. R. M. Piedra, *Parallel Processing with the TMS320C4X*, Texas Instruments, pp. 97-133, 1993.

▲전 창 호(Chang Ho Jeon) 1949년 12월 20일생



1977년 : 한양대학교 전자공학과(공학사)
 1982년 : Cornell University(공학석사)
 1986년 : Cornell University(공학박사)
 1986년~1989년 : 성균관대학교 전기공학과 조교수
 1989년~현재 : 한양대학교 전자컴퓨터공학부 교수

▲박 성 주(Sung Joo Park) 1957년 9월 26일생



1983년 : 한양대학교 전자공학과(공학사)
 1988년 : University of Massachusetts(공학석사)
 1992년 : University of Massachusetts(공학박사)
 1992년~1994년 : IBM Microelectronics(N.Y. USA) 연구 Staff

1995년~현재 : 한양대학교 전자컴퓨터공학부 조교수

▲이 동 호(Dong Ho Lee) 1962년 3월 2일생



1986년 : 한양대학교 전자공학과(공학사)
 1988년 : The University of Texas at Austin(공학석사)
 1991년 : The University of Texas at Austin(공학박사)
 1991년~1994년 : LG전자 중앙연구소 선임연구원

1994년~현재 : 한양대학교 전자컴퓨터공학부 조교수

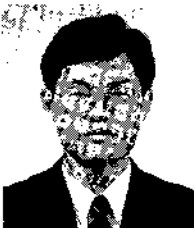
▲오 원 천(Won Tcheen Oh)

한국음향학회지 제14권 3호 참조

▲한 기 택(Ki Teak Han)

1982년 : 동국대학교 전자공학과(공학사)
 1993년 : 부산대학교 전자공학과(공학석사)
 1982년~현재 : 국방과학연구소 선임연구원

▲박 준 석(Joon Seok Park) 1969년 5월 23일생



1993년 2월 : 한양대학교 전자계산학과(공학사)
 1995년 2월 : 한양대학교 대학원 전자계산학과(공학석사)
 1995년 3월~현재 : 한양대학교 대학원 전자계산학과 박사과정