

TMS320C40 을 이용한 소나 신호처리시스템의 성능분석

박광철, 문병표, 전창호, 박성주, 이동호

Performance Analysis of a Sonar Signal Processing System using TMS320C40

Kwang-Chul Park, Byoung-Pyo Moon, Chang-Ho Jeon, Sung-Ju Park, Dong-Ho Lee
Dept. Of Electrical and Computer Engineering, Hanyang Univ

Abstract

소나 시스템과같이 방대한 양의 연산을 요구하는 고속 신호처리를 구현하기 위해서는 상용 DSP 칩의 병렬 처리 방법은 필요 불가결하다. 본 논문에서는 TI사의 TMS320C40 을 이용한 병렬 신호 처리 시스템을 소개한다. TI사의 TMS320C40 을 이용한 소나 시스템 신호처리부의 기본 모델을 제시하고, TI에서 제공하는 FFT 구현 소스의 분석을 통한 연산의 수학적인 모델을 재시하고 이를 근거로 제안된 모델의 성능을 분석하였다.

I 개요

1970 년말 NMOS 기술을 이용한 고정 소수점 방식의 DSP 칩이 처음으로 개발된 이래로 DSP 칩의 시장은 눈에 띄는 발전을 거듭해 왔다.[1] 이러한 기술 개발의 성과로 상용 DSP 칩의 성능 또한 괄목상대한 성장을 했다는 것도 사실이다. 그러나 이러한 DSP 칩의 성능의 발전에도 불구하고 아직까지 레이더, 영상 신호 처리(image signal processing), 소나(sonar)같은 고속의 신호처리를 요구하는 분야에서는 몇 개의 상용 DSP 칩으로 실시간 신호 처리를 할 수 있는 시스템을 구축하는 것은 불가능하다. 이를 만족시키기 위해서는 DSP 칩의 병렬 처리가 필수적 요소가 되며 시스템의 병렬 처리 기법은 지금까지 많은 연구가 수행되었다.[2]-[3] 시스템의 병렬 처리에 관한 연구는 레이더나 영상 신호처리에 집중되어 있으며 소나 시스템의 병렬 처리에 관한 연구는 지금까지 그 연구 내용이 국내적으로 미비하며, 더욱이 소나의 분야 자체 특성상 연구 내용의 공개는 어려운 실정이다.

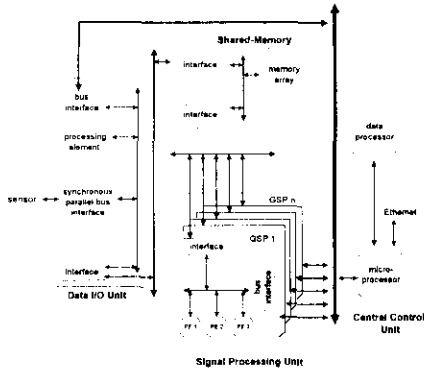
본 연구에서는 TI사의 TMS320C40 을 이용하여 소나 시스템 신호처리부의 기본 모델을 제안하였다. 제안된 모델은 소나 체계에서 사용하는 여러 가지 연산의 특성, 시스템의 확장성,

TMS320C40 의 특성들을 고려하였다. 소나 알고리즘은 여러가지의 DSP 이론을 필요로 하는데 그 중에서도 FFT 연산이 가장 많은 비중을 차지한다.[4] 따라서 제안된 모델은 FFT 의 구현을 고려하여 성능을 검증하였으며 이는 TI사에서 제공하는 'Parallel Processing with the TMS320C4X'의 Source Code 의 분석을 통해 수학적인 모델을 구하고 이를 근거로 하였다.

본 논문의 전개는 다음과 같다. II 장에서는 소나 신호처리 시스템의 구성 및 소나 시스템 신호처리의 구조 모델을 제안하였다. III 장에서는 FFT 의 병렬 처리 방법을 소개하고 소스코드의 모델링 방법을 설명하였다. IV 장에서는 제안된 모델의 Simulation 을 소개하고, V 장에서는 결론을 맺는다.

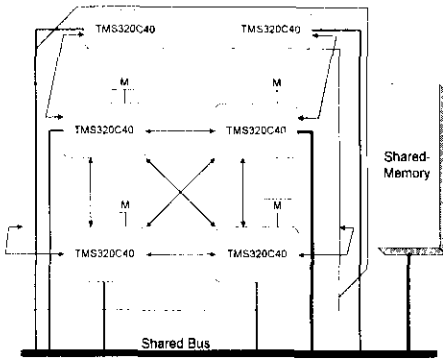
II 소나 신호처리 시스템

<그림 1>은 소나 시스템의 구현의 한가지 예이다. 전체 시스템은 신호처리부, 공용메모리부, 데이터 입출력부, 그리고 중앙제어부 등 크게 네 개의 subsystem 으로 구성되어있다. 센서를 통해 입력된 데이터는 입출력부로 전달된 다음 전처리과정을 거쳐 공용메모리부로 전송된다. 신호처리부에서는 공용메모리부와 중앙제어부로부터 제공받는 데이터와 명령문을 고속으로 처리한다. 신호처리부는 10GFLOPS 의 성능을 요구하며 각 시스템별 연결구조는 버스를 사용한다. 공용메모리부는 데이터 입출력부를 통하여 들어오는 데이터를 저장하여 신호처리부에서 처리할 수 있도록 제공해 주며 신호처리부의 임시 결과로 출력되는 데이터를 저장 하기도 한다. 데이터 입출력부는 수중 음향 신호를 받아서 공용메모리부로 전송하는 역할을 한다. 소나 시스템의 신호처리부에서 필요한 특성은 다음과 같다. 첫째, 소나 시스템의 신호처리부의 기본 구조 모듈은 직렬 및 병렬 구조로의 확장성이 있어야한다. 이는



<그림 1> 소나 시스템 구성도

소나 시스템이 방대한 양의 수중 음향 데이터를 처리해야 함은 물론, 복잡한 알고리즘을 구현할 수 있기 위함이다. 둘째, 시스템은 모듈간의 연결에 있어서 안정성을 고려한 버스 구조를 가져야 한다. 이를 위해서는 데이터 버스와 시스템 제어 버스를 따로 가지는 것이 적당하다. 이러한 사실로부터 제안된 하드웨어 구조의 개략적 구성은 <그림 2>과 같다.



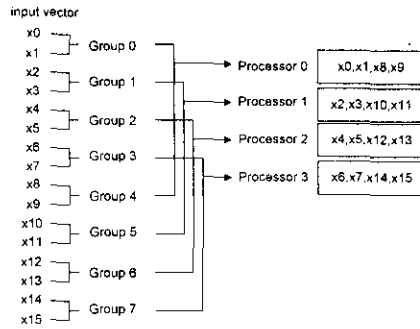
<그림 2> 제안하는 소나 시스템 신호처리부의 기본 모듈

<그림 2>에서처럼 하드웨어의 전체적인 구조는 보드 단위로 확장이 가능하다. 각 보드에는 4 개의 TMS320C40 이 communication port 를 통하여 완전연결망 형태로 서로 연결되어 있다. 또 보드들은 공유버스와 communication port 를 통하여 이종으로 연결된다. 이러한 TMS320C40 간의 communication port 를 통한 칩간의 연결구조는 앞서 살펴 보았던 시스템의 병렬 구현에 있어서 발생하게되는 연산의 중간과정에서 데이터의 교환이 필요한 FFT 연산의 특성을 고려한 구조이다. 그리고 보드간 또는 서로 다른 기능적 모듈간의 통신과 데이터 교환에 사용되는 시스템 전체의 공유메모리가 있고, 각 DSP 는 다른 프로세

서에 전달될 필요가 없는 지역데이터의 처리나 프로그램의 저장에 사용하는 지역메모리를 가지고 있다. 한 보드당 4 개의 TMS320C40 을 가지도록 설계한 것은 소나 체계에서 수행하는 여러 가지 연산의 특성상 프로세서의 개수는 2 의 배수가 가장 적합하고, TMS320C40 이 6 개의 communication port 를 가지고 있기 때문에 8 개의 프로세서를 한 보드에 장착할 경우 완전연결망 형태로 프로세서들을 연결할 수가 없기 때문이다. 또한 2 개의 프로세서로 한 보드를 구성할 경우 사용되지 못하는 communication port 의 수가 늘어나 자원의 활용도 측면에서 비효율적일 뿐만 아니라 보드당 얻을 수 있는 성능이 미약하여 목표로 하는 10 GFLOPS 의 성능을 얻기 위해서는 많은 수의 보드를 사용하여야 하며 그럴 경우 보드간 통신의 횟수가 증가하여 시스템 전체의 성능이 저하될 수 있기 때문이다.

III. FFT 의 병렬 처리 방법 및 소스 코드 모델링

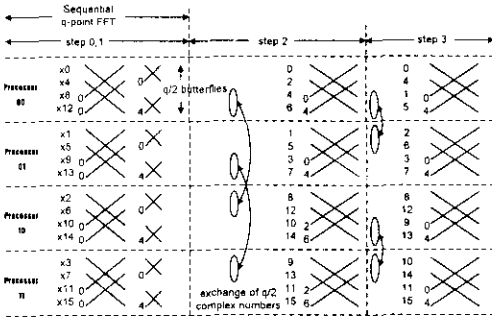
한 예로서 16-point radix-2 DIT FFT 알고리즘을 병렬로 연산하기 위해서는 <그림 3>과 같이 순차적으로 들어오는 신호 $x(i)$ 를 재배열하는 작업이 필요하다.



<그림 3> FFT 연산을 위한 입력 신호의 재배열

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

FFT 알고리즘의 특성상 $\log_2 N$ 에 해당하는 스텝을 가지게 되며 데이터 분할에 의한 병렬 처리 방법을 사용할 경우에는 처음 $\log_2^N - \log_2^p$ 의 스텝에서는 연산의 중간 값들을 프로세서간 교환할 필요가 없으나, $\log_2 p$ 만큼의 스텝에서는 프로세서간 중간 연산 결과를 교환해야하는 특성을 가지게 된다. 이러한 사실은 <그림 4>에 나타내었다. 이는 각 스텝에서 프로세서의 수 P 로 분배된 FFT 의 butterfly 를 각 프로세서가 연산하는 방식의 병렬 처리 방법이다. 그림에서 step 0,1 은 프로세서간 중간 연산 값들의 교환이 필요하



<그림 4> 4 개의 프로세서를 이용한 16-point DIT FFT 알고리즘

지않은 step 을 말해주며, step2 와 3 은 중간 연산 값들의 프로세서간 교환이 필요한 경우를 나타낸다.

본 논문에서는 TI사에서 제공하는 'Parallel Processing with the TMS320C4X'의 Source code 의 분석을 통해 연산의 실험적 모델링 구하였으며 이를 근거로 제안된 모델의 성능 분석을 하였다. Source code 의 명령어 사이클은 'TMS320C4X User's Guide'에 나와 있는 각 명령어의 사이클 수를 근거로 했다.

FFT 의 알고리즘은 특성상 중간 연산 결과를 프로세서와 프로세서간 데이터의 교환을 요구한다. 이 때 하드웨어의 관점에서 TMS320C40 의 communication port 를 통한 데이터 전송은 5 사이클이 필요하다고 가정한다.

source code 는 네 개의 어셈블리 함수들로 구성되어 있다. 첫째, cmove.asm 이다. 이 함수는 외부 메모리로부터 계산할 데이터들을 내부 RAM 에 저장하기 위한 프로그램이다. 둘째, r2dit.asm 은 <그림 4>에서 보이는 것과 같이 FFT 를 병렬 처리 연산을 실행하는 과정 중에서 프로세서와 프로세서간의 데이터 교환이 필요치 않은 스텝에서의 FFT 를 실행하는 프로그램이다. 셋째, exch_r.asm 은 communication port 를 통해서 데이터를 교환하기 위한 code 이다. bfly.asm 은 프로세서와 프로세서 사이에서 교환된 데이터를 FFT 의 butterfly 를 실행하는 code 이다.

위의 결과를 이용하여 B 개의 보드와 P 개의 프로세서를 이용하여 N-point FFT 를 실행하기 위한 총 사이클 수는 프로세서 내부에서 실행되는 intra-processor phase 의 사이클 수 r2dit.asm 와 프로세서와 프로세서 사이에서 실행되는 inter-processor phase 의 사이클 수 {exch_r.asm + bfly.asm}와 보드와 보드 사이에서 실행되는 inter-board phase 사이클 수 {cmove.asm + bfly.asm} 를 모두 합한 결과가 된다.<표 1>에는 FFT phase 당 사이클 수를 정리하였다.

<표 1> FFT 스텝당 CYCLE 수

구분	CYCLE 수
Intra Processor Phase	$26 + 14.5 * \frac{N}{P}$
Inter Processor Phase	$\{(29 + 20 * \frac{N}{P}) + (29 + 9 * \frac{N}{P})\} * \log_2^{PB}$
Inter Board Phase	$\{(28 + D * \frac{N}{P}) * P + (29 + 9 * \frac{N}{P})\} * \log_2^B$

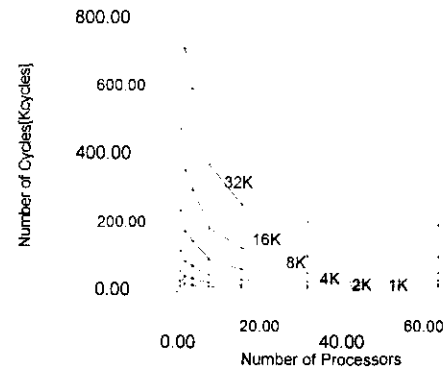
여기서 \log_2^{PB} 는 inter-processor phase 에서의 스텝 수이고 \log_2^B 는 inter-board-phase 에서의 스텝 수이다. N 과 P 와 B 는 연산에 사용된 데이터 수, 프로세서의 수, 보드수를 의미하며, PB 는 한 보드내에서의 프로세서 수이다. D 는 메모리를 통하여 하나의 데이터를 전송할 때 필요한 사이클 수를 의미한다. 이러한 결과를 수식으로 정리하면 다음과 같이 된다.

$$(14.5 * \frac{N}{P} + 26) + (58 + 29 * \frac{N}{P}) * \log_2^{PB} \tag{식 1}$$

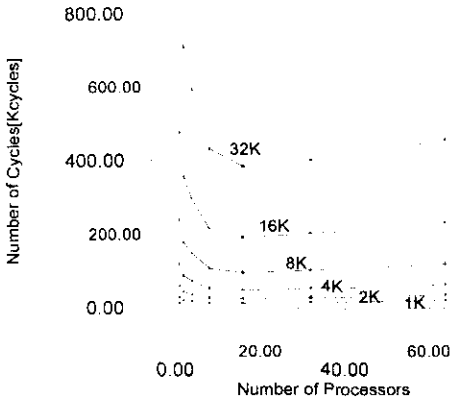
$$+ (28 * P + D * N + 29 * 9 * \frac{N}{P}) * \log_2^B$$

V 분석 결과

<식 1>을 이용하여 제안된 모델에 적용한 결과를 살펴보기로 한다. <그림 5>와 <그림 6>은 D 가 각각 1 사이클, 3 사이클이라고 가정 했을 때의 결과이다. 프로세서의 수가 증가함에 따라 연산에 필요한 총 사이클의 수가 전체적으로 감소함을 볼 수 있으며 이는 적절하게 데이터의 병렬 처리가 실행됨을 의미한다.

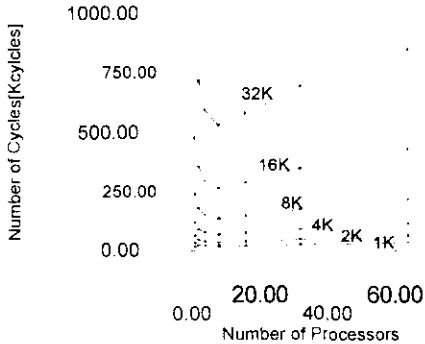


<그림 5> FFT 성능 분석 결과 (D=1 cycle)

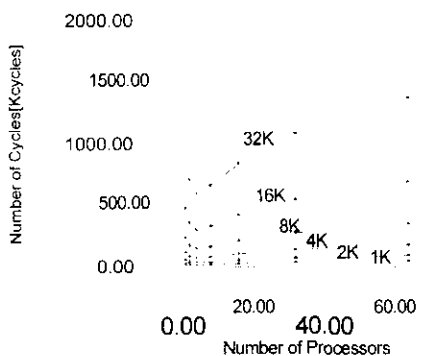


<그림 6> FFT 성능 분석 결과 (D=3 cycle)

<그림 5>와 <그림 6>에서 프로세서의 수가 2 일 때 급격한 싸이클수의 증가를 보여주고 있는데 이는 Inter Processor Phase에서의 데이터 전송에 요구되는 싸이클의 증가때문이다. <그림 7>과 <그림 8>은 각각 D가 6, 10 싸이클 일때의 결과이다. D가 증가함에 따라 프로세서의 수가 증가할 지라도 프로세서간의 데이터 통신에 필요한 싸이클이 증가하여 병렬 처리 시스템의 성능은 저하됨을 볼 수 있다.



<그림 7> FFT 성능 분석 결과 (D=6 cycle)



<그림 8> FFT 성능 분석 결과 (D=10 cycle)

<표 2> 4K-point FFT 실행 결과(D=10)

프로세서 수	Total Cycle	Intra Processor Phase (%)	Inter Processor Phase (%)	Inter Board Phase(%)
1	59418	59418 (100%)	0 (0%)	0 (0%)
2	89172	29722 (33.3%)	59450 (66.6%)	0 (0%)
4	74382	14874 (20%)	59508 (80%)	0 (0%)
8	83083	7450 (8.9%)	29812 (35.8%)	45821 (55.1%)
16	106184	3738 (3.5%)	14964 (14.1%)	87482 (82.3%)
32	138533	1882 (1.4%)	7540 (5.4%)	129111 (93.1%)
64	178210	954 (0.5%)	3828 (2.1%)	173428 (97.3%)

<표 2>는 4K-point FFT 실행 결과를 나타내주는 표이다. 표에서 보이는 바와 같이 프로세서의 수가 증가함에 따라서 연산을 실행하기 위한 inter processor phase 와 inter board phase 의 싸이클 수가 계속 증가되는 것을 보여주고 있다. 이것은 point 가 일정한 FFT 연산을 할 경우, 단순히 프로세서의 증가는 프로세서간 또는 보드간의 연산 비중이 커지기 때문에 오히려 시스템의 성능을 저하시키는 결과를 초래한다는 것을 보여준다.

VI. 결론

본 논문에서는 소나 시스템 신호처리부의 모델을 TI사의 TMS320C40 을 이용하여 제시하였다. TI사가 제공하는 'TMS320C4X User's Guide'와 'Parallel Processing with the TMS320C4X'의 source code 에 근거하여 FFT 구현을 위한 요소 연산에 대한 수학적인 모델을 구하여 제안된 신호처리부 모델의 성능을 분석하였다. 일정한 포인트의 FFT 연산의 경우 단순한 보드 수의 증가를 통해서 보드와 보드 사이의 데이터 전송에 필요한 싸이클의 증가로 인하여 시스템의 성능을 증가시킬 수 없다는 것이 판명되었다. 이러한 사실은 시스템의 구현에 있어서 시스템 버스의 데이터 병목현상이 전체 시스템의 성능을 좌우하는 중요한 요소라는 것을 말한다.

참고 문헌

[1] 金在錫, "고속 DSP 기술 동향," 전자공학회지 제 22 권 제 2 호, 1995, pp. 78-88
 [2] 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, Jul. 1992, pp. 786-792
 [3] L. N. Bhuyan, Q. Yang, and D. P. Agrawal, "Performance of Multiprocessor Interconnection Networks," IEEE Computers, Feb. 1989, pp. 25-37
 [4] R. O. Nelsen, Sonar Signal Processing, Artech House, 1991, p231-248