

OpenMP를 활용한 TI 다중코어 DSP기반의 고속 FFT 처리부 개발

남경호 · 오우진

금오공과대학교

Development of High speed FFT system using OpenMP on TI multicore DSP

Kyungho Nam, Woojin Oh

Kumoh National University of Technology

E-mail : wjoh@kumoh.ac.kr

요 약

신호처리 시스템에서 FFT는 많이 사용되고 있으며, 고속화를 위하여 많은 연구가 진행되어 왔다. FFT는 통신, 영상처리, 레이더 등 많은 영역에서 직접 또는 변형되어 많이 활용되고 있으나 실시간 처리 속도 한계와 가격의 문제로 FFT 길이가 제한되는 경우가 많다.

본 연구에서는 TI사의 고속 DSP인 8 core의 TMS320C6678에 OpenMP 병렬처리 기법으로 FFT를 구현한 결과를 제시한다. 속도 개선을 위한 다양한 병렬처리 방안에 대하여 단일 FFT의 길이별 성능과 다중 FFT를 처리하기 위한 방안을 제안하였다. 이러한 OpenMP기반의 FFT는 DSP간 hyperlink 연결로 다수의 DSP로 병렬처리로 성능 개선이 가능하며, 본 연구에서는 16 core로 확장하여 그 성능이 30% 내외 개선되는 것을 보였다. 본 연구 결과는 초 고속 신호처리가 요구되는 의료영상, 초고해상도 영상처리, 고정밀 레이더 등에 활용이 가능할 것이다.

키워드

다중코어 DSP, OpenMP, Hyperlink, FFT

1. 서 론

디지털 신호처리의 기본인 FFT(Fast Fourier Transform)은 디지털 통신, 영상처리 및 방송, 레이더 등 다양한 분야에서 활용되고 있다. 고속 FFT를 구현하기 위해서 많은 경우에 FPGA, SOC, ASIC 등의 HW 방식으로 접근하고 있다. 그러나 알고리즘 개발과정이나 Cognitive Radio, 소량의 특수 장비에서는 소프트웨어 적인 접근이 필요한 경우가 있다.

본 논문에서는 다중 코어를 갖는 DSP 환경에서 2D FFT를 병렬처리 OpenMP API로 구현하기 위한 방안을 검토하고 그 성능을 비교하였다. 사용한 DSP는 TI사의 TMS320C6678로 8 core를 지원하며, 다수의 DSP를 연결하여 확장인 가능한 특징이 있다. FFT는 N-by-N의 2D에 대하여 N=256~2048까지 시험하였다.

TI사의 C6678은 8 core를 갖는 다중코어 DSP로 각각의 core는 32K L1 Cache, 512K L2 메모리를 갖고 있으며 모든 core가 공유하는 shared memory로 4MB SRAM이 있다. core간 연결버스는 Teranet로 연결되어 최고 50GBaud까지 전송이 가능하며 그림 1과 같이 외부의 다른 DSP와

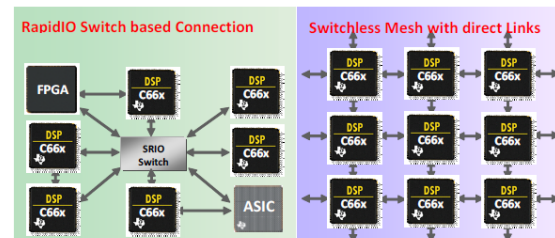


그림1. 다중 DSP의 연결 구조 예 [1]
(switch 방식, mesh 방식)

II. TI 다중 코어 DSP와 OpenMP

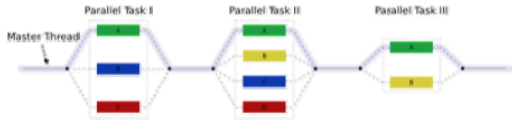


그림 2. 병렬처리 모델[2]

Hyperlink로 연결하여 성능 확장이 가능하다. [1]
 OpenMP는 공유 메모리에 기반한 다중 처리 프로그래밍을 위한 표준 API로 개발된 것으로 현재 Ver 4.0까지 제공되고 있으며, TI의 DSP는 Ver 3.0을 지원한다. Ver 4.0은 이중 core 간의 다중처리가 추가 된 것이므로 동일 core 로 구성된 C6678 DSP에서는 성능차이가 없을 것으로 판단된다. OpenMP는 프로그램 지시자(directives)로 구현되어 기존에 작성된 프로그램에 쉽게 적용이 가능한 장점이 있으며, time scheduling, dynamic load 관리 등을 지원한다. [2]

OpenMP의 병렬처리 과정은 그림 2과 같이 master thread에서 필요에 따라 병렬 처리를 수행하는 fork-join 방식이 일반적으로 사용된다. 사용자는 병렬 처리 영역과, thread 수를 지시하면 컴파일러가 부하에 따라 자동적으로 병렬화를 구현한다. C언어의 반복문에 대한 간단한 예는 다음과 같다.

```
#pragma omp parallel for
for (int i=0; i<n; i++)
    c[i] = a[i] + b[i];
```

III. 2D FFT 구현 및 성능비교

논문 [3]에서는 특정길이에 대한 FFT 알고리즘을 개선하여 다중코어에 적합하도록 개선하였으나, 본 논문에서는 DSP의 OpenMP 성능을 확인하기 위한 것이므로 TI에서 제공되는 표준 FFT 라이브러리를 그대로 사용하였다.

본 실험에서는 TI사에서 제공하는 EVM (Evaluation Module)인 TMDXEVM6678LE를 사용하였으며, 그 특징은 다음과 같다.

- single C6678 multicore DSP
- 5128M DDR3, 64MB NAND Flash
- 1MB EEPROM for bootloader
- XDS560V2 onboard JTAG emulation

개발환경은 Code Composer 5.5에서 Omp_1.2.0.5를 적용하였으며, platform은 내부 메모리만 사용하는 ti.omp.examples.platforms.evm6678을 수정하여 대용량 데이터만 외부 DDR3에 저장하도록 설정하였다. FFT는 TI에서 최적화하여 제공하는 dsplib의 함수 중에서 32bit float로 작성된 DSPF_sp_fftSPxSP를 사용하였다. 단일 EVM보드는 1개

표 1. 데이터 메모리 위치별 수행속도 (us)

FFT 길이 (N)	Code: Shared	Shared	DDR3
	Data: Shared	DDR3	DDR3
64	0.154	0.226	0.356
128	0.250	0.269	0.507
156	0.510	0.387	1.014
512	1.398	0.849	2.433
1024	2.893	1.799	5.181

의 DSP를 갖고있어 8 core까지만 지원되므로 16 core를 구현하기 위하여 2개의 보드를 hyperlink로 연결하였다.

데이터 메모리의 저장위치에 따른 수행 속도 차이를 비교하기 위하여 단일 core에서 1D FFT 수행 결과를 표 1에 보였다. 여기서 L2는 core 별 56독립 메모리로 속도가 가장 빠른 공간이며, shared 메모리는 MSMCSRAM (Multicore Shared Memory Controller SRAM)으로 모든 core가 공유하는 공간이다. 외부 메모리인 DDR3를 사용하는 경우에는 캐시가 동작하여 속도가 개선된 것으로 보인다. 따라서 본 연구에서는 프로그램은 공유메모리에, 큰 데이터는 shared 메모리에 저장하는 설정을 사용하여 OpenMP 동작을 시험하였다.

NxN의 2D FFT의 수행 속도를 core 수에 따라 비교하였다. 2D FFT는 1D FFT를 2N회와 matrix transpose를 수행하는 것이나 본 논문에서는 matrix transpose는 제외한 측정 시간을 제시하였다. 8 core, 1024pt 2D FFT의 경우 4.76ms의 소요 시간이 측정되었었으며 이는 약 30%의 overhaed가 부가된 수준이다. transpose를 직접적으로 병렬처리하는 것은 데이터의 연관성 때문에 어렵거나 submatrix로 분할하여 다중처리에 용이하게 구현하였다.

그림 1에 EVM 보드 2개를 hyperlink로 연결하여 동작하는 사진과 code composer에서 16 core로 설정한 것을 보였다. 각 보드는 EVMBOC(EVM Breakout Card)로 서로 연결되어 있으나 본 실험과 무관하며 향후의 switch 방식으로 연결하기 위한 것이다. OpenMP의 단점 중에 하나는 loop 병렬화를 위한 초기 오버헤드가 많다는 것이나, 위의 예에서는 나타나지 않았다. 또한 core간 공유 메모리 접근에서 나타날 수 있는 충돌은 고속 내장메모리를 사용하고, 50Gbaud급의 hyperlink로 연결되므로 속도저하는 관찰되지 않았다.

IV. 향후 계획

현재 16 core에서 추가로 DSP를 연결하는 방안을 고려하였으나 EVM 보드의 Hyperlink 연결 소켓이 1개만 지원되어 더 확장하지 못했다. 현재

별도의 보드를 개발하고 있으며, SRIO Switch 등을 사용하여 시험할 예정이다. 이렇게 개발된 시스템은 FMCW 레이더의 실시간 처리에 활용될 예정이며, 복잡한 algorithm이 추가되어도 scalable한 구조로 쉽게 확장이 가능한 장점을 갖게 된다. 실제 최종 구현에서는 알고리즘 검증 후에 최적화된 DSP 수로 간략화 될 것이다.

참고문헌

- [1] Instruments, Texas. “Tms320c6678 multicore fixed and floating-point digital signal processor.” Data manual, available online at <http://www.ti.com/lit/ds/symlink/tms320c6678.pdf> (2010).
- [2] Chapman, Barbara, Gabriele Jost, and Ruud Van Der Pas. Using OpenMP: portable shared memory parallel programming. Vol. 10. MIT press, 2008.
- [3] Yang, Haofan, et al. “Efficient implementation of 3780-point FFT on a 16-core processor.” ASIC (ASICON), 2013 IEEE 10th International Conference on. IEEE, 2013.