

ADSP-2105를 이용한 범용 DSP보드의 제작 및 이를 이용한 실시간 FFT 분석기의 구현

조철희* 서주하** 권호열** 경익주*
* 강원대학교 전자공학과 ** 강원대학교 컴퓨터공학과

Implementation of a General Purpose DSP board using the ADSP-2105
Digital Signal Processor and its application to a real-time FFT analyzer

Chul-hee Jo* Hwa-ju Seo** Ho-yeol Kwon** Ik-joo Chung*

* Dept. of Electronics, KangWon National University

** Dept. of Computer, KangWon National University

요 약

디지털 신호를 처리하기 위해 특별히 제작된 ADSP-2105(Anlong digital signal processor)는 빠른 Fixed-point 연산과 Harvard-architecture로 구조화됐기 때문에 빠른 수행연산을 할 수가 있다. 본 논문은 이 DSP 프로세서를 이용해 음성신호의 실시간 FFT 분석에 관한 방법을 소개한다. 실시간 FFT분석기로서의 DSP보드는 크게 음성신호를 받는 입력부분과 FFT를 계산하는 FFT부분으로 나뉘어지는데, 입력부분은 AD1849로 8KHz로 데이터를 샘플링해 받게 되었고, FFT부분은 실제로 DSP가 FFT를 수행하는 부분으로 되어있다. 실시간 처리를 구현하기 위해 입력부분은 두개의 뱅크로 만들어 한 뱅크에서 음성신호를 받아들이는 동안에 다른 뱅크에서는 FFT를 계산하도록 되어있어서, DSP 보드는 항상 음성신호를 샘플링할 수 있는 상태를 유지할 수 있다. 그리고 FFT 처리부는 빠른 처리로 음성신호를 샘플링할 뱅크가 채워지기 전에 실행되게 프로그램되어 있어 실제적으로 모든 음성데이터를 FFT하게 되어 있다.

1. 서 론

Digital Signal Processing(DSP)의 고속연산을 필요로 하는 응용분야에 특별히 설계된 ADSP-2105는 single-chip processor로 기능화되었다. 특히, 프로그램과 데이터 라인이 분리되어 서로간의 트래픽을 줄인 Modified Harvard-architecture로 설계되었다. 또 데이터 어드레스를 관리하는 DAG(Data Address

Generator)은 Circular buffer, Automatic address modification, 그리고 FFT 에 적합한 bit-reversal 기능들이 있다. 이런 기능들을 이용해 빠른 FFT를 프로 그래밍할 수 있었다.[1]

실시간 FFT를 구현하기위해 음성데이터를 두개의 뱅크로 만들어서 한 뱅크마다 FFT가 수행이 되는데, 한 뱅크가 음성데이터를 받는동안에 다른 뱅크에서는 FFT를 수행한다. 이런 구조로 DSP보드는 항상 음성데이터를 받아들일 준비가 되어있고 FFT를 수행할수가있다. 프로세서 측면에서 볼때 음성데이터를 받는 시간간격이 상당한 시간이기 때문에 이 시간을 이용해 다른 뱅크의 FFT를 수행할 수 있고, 문제는 한 뱅크에서의 FFT를 끝마친 시간이 다른 뱅크에서의 음성데이터가 채워지기 전에 이루어져야 된다. 본 논문에서는 실시간 FFT를 구현하기위한 FFT의 계산시간과 한 뱅크에서의 샘플링된 시간을 비교해볼이의 타당성을 제시할것이다.

2절에서는 DSP보드를 간단히 설명하겠고, 3절에서는 실시간 FFT를 구현한 알고리즘에 대해 설명하겠다. 끝으로 4절에서는 실험적인 결과로 DSP보드에서의 FFT 계산시간을 구해 실시간이 가능한 이유를 보여줄것이다.

2. DSP 보드

DSP보드와 PC간의 인터페이스를 블록도로 간단히 살펴보면, DSP보드에는 음성신호를 받는 입력부분과 FFT를 수행하는 처리부분으로 나뉘어져있고, PC에서 FFT 결과를 복사하는 부분으로 되어있다. 먼저 DSP 보드에 관하여 살펴보고, DSP보드와 PC간의 인터페이

스 구조를 설명하겠다.

DSP보드는 그림 1에서 보듯이 음성신호를 받는 입력부분과 FFT를 수행하는 처리부분으로 되어있다.

입력부분은 편의상 BANK0, BANK1으로 나뉘어 BANK0가 샘플링된 신호를 받아들이는 동안에 BANK1은 FFT를 수행하도록 프로그래밍되어있고, 반대로 BANK1이 음성신호를 받아들이는 사이에 BANK0는 FFT를 수행한다. 특히 CODEC용으로 제작된 AD1849은 인터럽트에 의해 데이터를 읽어오기때문에, 입력부분은 항상 데이터를 받아들이기 준비가 되며, 동시에 매 블록마다 FFT 결과를 수행시킴수가 있다. 하드웨어적인 구성은 AD1849와 DSP가 시리얼로 연결되어있어 샘플링된 데이터를 받게되었고 그 데이터는 메모리에 저장되어진다.[2]

처리부분은 한 블록에 들어있는 샘플링된 데이터들을 FFT 해서 결과를 메모리에 저장되는데, FFT를 실행할 조건은 한 블록의 음성데이터가 채워질 때이다.

지금까지 설명한 DSP보드의 입력부분과 FFT 처리부분은 DSP 보드의 메모리에서 이루어지고, 다음은 메모리에 들어있는 FFT 결과값을 PC에게 빠르게 전송하는 부분을 설명하겠다.

PC와 DSP보드 사이를 빠르게 전송하는 방법으로 포트를 사용해 데이터를 교환하는대신 DSP보드의 메모리를 직접 어드레싱해서 데이터를 주고받는다. 이런 구조로 되어있어 간단히 FFT 결과를 메모리에서 메모리로 복사하면 된다. 그림 2는 메모리와 PC사이의 인터페이스 구조를 나타낸 것인데, 프로세서와 연결된 메모리는 PC가 제어하도록 설계되었다. 즉, 메모리에 저장되어있는 FFT결과를 PC가 복사하기위해 우선 프로세서와 메모리사이를 분리시키고나서 PC가 데이터를 복사하게되어있다. 하지만 프로세서와 메모리사이를 분리시키기위해 프로세서를 중지시키게 설계했고, 매 번의 FFT 결과를 PC가 복사할때마다 잠깐동안의 프로세스는 중단이 된다. 이런 방법으로인해 프로그램의 흐름에 지연현상이 야기되지만 FFT를 계산시간에 비교해 적은 시간이기 때문에 고려할 문제는 아니다.

3. 실시간 FFT의 구현

앞절에서 설명한 각부분을 종합해 실시간 FFT 구현 방법에 대해서 자세히 설명해보겠다.

그림 3은 그림 1의 입력부분, FFT 처리부분, 그리고 그림 2의 PC와의 인터페이스 구조를 종합해서 도시화한 것이다. 앞서 설명한대로 메모리에는 BANK0, BANK1, FFT 결과가 저장되고, PC와 연결된 스위치들을 조절해가며 FFT결과를 얻을수있다. 동작흐름을 보면, 스위치 A가 BANK0에 연결시키면, 스위치 B가 BANK1과 연결돼 BANK1을 FFT 하고 스위치 D는 PC와 메모리 사이를 분리시키고 스위치 C는 DSP 프로세서와 메모리를 분리시켜 DSP 프로세서가 메모리를 사용하게된다. FFT처리부가 종결이되면, 프로세서는 PC에게 종결신호를 보내 FFT 결과를 복사하라는 메시지를 알리고, 이 신호를 받은 PC는 메모리에 들어있는 FFT결과를 복사하고, 스위치 A를 BANK1, 스위치 B를 FFT 처리부와 연결시키고 스위치 C를 DSP 프로세서와 메모리를 연결시키고 스위치 D를 PC와 메모리를 분리해 FFT를 수행하게된다. BANK0, BANK1을 번갈아 가며 수행하기때문에 밖에서 볼때 항상 음성신호를 샘플링하는 것을 알수있으며, 실시간 FFT를 수행할수가 있다.

위의 과정에서 PC가 메모리를 사용하기위해 프로세서를 중지시켜 하드웨어적으로 분리시키며 다시 프로세서가 메모리와 연결될때 프로그램의 흐름은 계속된다.

위에서 설명한 부분에서 FFT 종결신호는 PC가 주기적으로 프로세서를 중지시키면서 메모리에 들어있는 내용으로 종결신호를 참조해 프로세스의 흐름을 지연시키게 되어있지만 실험적으로 이에 대한 문제는 크지 않다.

본 논문에 사용된 FFT는 DIT-FFT 알고리즘으로 프로그래밍되었으며 다음은 N-Point FFT를 수행할 사이클수를 계산한것이다.

FFT에 사용될 레지스터 초기화	10
총 Stage수(Floating Point Scaling 포함)	$\log_2 N (6N + 27)$
총 groups 수	$9N - 9$
총 butterfly 수	$12N - 12$
	(식 1)

DIT-FFT 알고리즘에서 FFT를 수행하기에 앞서 변수들의 초기화작업에 10 사이클, fixed연산에 대한 예러보정과 총 Stage에 $\log_2 N (6N + 27)$ 사이클, 그리고 N-Point DIT-FFT에대해 총 groups, 총 butterfly의 사이클이 각각 $9N - 9, 12N - 12$ 이다. 그래서 총 사이클 수는

$\log_2 N (6N + 27) + 21N - 11$ 이다.

4. 실험 결과

본 논문의 실시간 FFT의 성능을 알아보기위해 음성 신호는 AD1849로 8KHz로 샘플링했으며, 프로세서는 ADSP-2105, 10MHz로 실험했다. DSP 보드의 프로그램은 어셈블러로 프로그래밍되었으며 DSP 보드를 제어할 프로그램은 C로 프로그래밍되어있다. 표 1은 식 1을 근거로 각 N-Point FFT에대해 FFT 계산시간과 한 बैं크를 채울 시간을 비교한것이다. 이 표에서 볼때 모든 N-Point FFT에대해서 한 बैं크에 샘플링될 시간보다 FFT 계산시간이 적어 실시간이 가능하다는 것을 알수있다.

N-Point	총 Sampling	
	시간(ms)	FFT 계산 시간(ms)
2	0.025	0.0007
4	0.050	0.0018
8	0.100	0.0038
16	0.200	0.0082
32	0.400	0.0176
64	0.800	0.0379
128	1.600	0.0824
256	3.200	0.1787
512	6.400	0.3863
1024	12.800	0.8320
2048	25.600	1.7846

표 1. FFT계산시간과 한 बैं크에서의 샘플링시간비교

5. 결론

DSP보드가 실시간 FFT를 수행하지만 매번의 FFT 결과를 시각화할 때 다음의 बैं크를 FFT 하기전에 마쳐야 다음의 FFT로 재빨리 수행할 수 있을것이다. 그러나, 시각화의 처리시간이 FFT 소비시간보다 많은 시간이 소비되므로 FFT 결과를 디스크에 저장시킨후 분석을 요하는 것이 나올것이다. 이에 대한 해결책으로 PC가 FFT의 종결신호를 알기위해 메모리를 주기적으로 체크하지 않고 인터럽트로 처리해 PC의 부담을 줄여 시각화하는 시간을 늘릴수있으며, 각 बैं크마다의 FFT화하는 것보다는 시각화가 끝나는대로 FFT 결과를 화면에 보여줘도 될것이다.

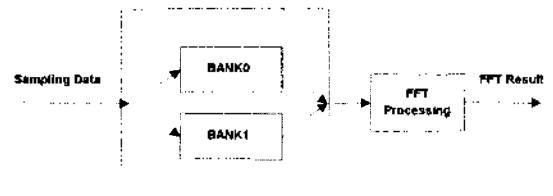


그림 1. DSP보드의 입력부분과 출력부분의 블록도

참 고 문 헌

- [1] DIGITAL SIGNAL PROCESSING LABORATORY USING THE ADSP-2101 MICROPROCESSOR, ANALOG DEVICES, PRENTICE HALL.
- [2] Serial-Port 16-Bit SoundPort Stereo Codec, AD1849, ANALOG DEVICES.
- [3] ADSP-21XX Coprocessor/Development Kit, Saddle Point Systems.



그림 2. DSP보드와 PC사이의 인터페이스

ADSP-2105를 이용한 범용 DSP 보드의 제작 및 이를 이용한 실시간 FFT 분석기의 구현

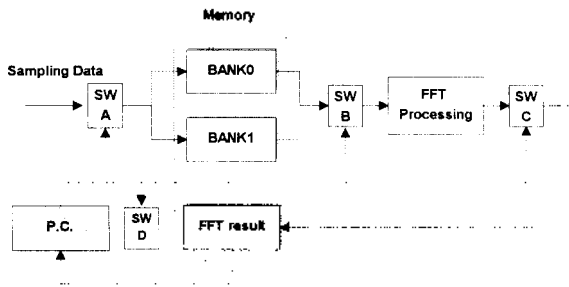


그림 3. 전체적인 블럭도