논문 2010-47IE-4-3

# 회전체 진동 데이터 획득을 위한 효율적인 FPGA 로직 설계

## ( Efficient FPGA Logic Design for Rotatory Vibration Data Acquisition )

이 정 석*, 유 등 열**

( Jung-Suk Lee and Deung-Ryeol Ryu )

### 요 약

본 논문은 회전체의 진동 데이터를 효율적으로 획득하기 위해 데이터 획득 시스템을 설계하였다. 데이터획득 장치는 필터와 증폭기로 구성한 아날로그 로직과 ADC와 DSP, FPGA, FIFO 메모리를 갖고 있는 디지털로직으로 구성하였다. 센서로부터 획득한 회전체의 진동신호는 아날로그 로직을 통과하여 FPGA에 의해 제어되고, 그 신호는 ADC를 통해 변환되고 FIFO 메모리에 저장하였다. 디지털 신호 처리는 FPGA 제어어의해서 FIFO 메모리에 들어온 데이터를 이용하여 DPS에서 신호처리를 수행할 수 있도록 구성하였다. 회전체 진동을 진단 및 분석하기 위한 진동 요소는 데이터 신호로서 실수 변환, Peak to Peak, 평균 값 산출, GAP, 디지털 필터, FFT 등을 DSP에서 처리하고 설정된 이벤트를 추적하며, 그 결과 값을 도출하여 조기 경보 시스템을 구축하였다. 모든 신호처리 과정 및 이벤트 추적은 여러 분석 단계 의해서 처리 시간이 소요되며, 특정 이벤트에 따라 처리 소요 시간에도 변동이 발생한다. 데이터 획득 및 처리는 연속적으로 실시간 분석을 수행해야 하지만, DSP에서는 입력된 신호를 처리하는 동안에 입력된 이후의 데이터에서 다음 입력처리 시간동안 획득한 데이터는 처리 될 수 없고, 특히 다수의 채널에서는 더 많은 데이터 손실이 일어날 수 있다. 따라서 본 논문에서는 데이터 손실이 적고 빠른 처리를 위하여 DPS와 FPGA을 효과적인 사용하였고, 이러한 여러 분석 단계 신호처리에서 발생되는 시간을 최소한으로 줄일 수 있는 방법으로 DSP에서 처리되는 신호단계 중 일부를 FPGA에서 처리할 수 있도록 설계 하였고, 그리고 단일의 신호 처리에 의해 수행되는 분석 단계를 병렬 처리로 데이터를 실시간으로 처리하였다. 그 결과로 DSP 만으로 구성된 신호처리 보다 DSP와 FPGA로 구성된 시스템이 훨씬 빠르고 안정된 신호 처리 방법을 제시하였다.

### Abstract

This paper is designed the efficient Data Acquisition System for an vibration of rotatory machines. The Data Acquisition System is consist of the analog logic having signal filer and amplifier, and digital logic with ADC, DSP, FPGA and FIFO memory. The vibration signal of rotatory machines acquired from sensors is controlled by the FPGA device through the analog logic and is saved to FIFO memory being converted analog to digital signal. The digital signal process is performed by the DSP using the vibration data in FIFO memory. The vibration factor of the rotatory machinery analysis and diagnosis is defined the RMS, Peak to Peak, average, GAP, FFT of vibration data and digital filtering by DSP, and is need to follow as being happened the event of vibration and make an application to an warning system. It takes time to process the several analysis step of all vibration data and the event follow, also special event. It should be continuously performed the data acquisition and the process, however during processing the input signal the DSP can not be performed to the acquisited data after then, also it will be lose the data at several channel. Therefore it is that the system uses efficiently the DSP and FPGA devices for reducing the data lose, it design to process a part of the signal data to FPGA from DSP in order to minimize the process time, and a process to parallel process system, as a result of design system it propose to method of faster process and more efficient data acquisition system by using DSP and FPGA than signal DSP system.

**Keywords :** rotatory machine, vibration, data acquisition, FPGA, DSP

* 정회원, 인하공업전문대학 메카트로닉스과
(Department of Mechatronics, Inha Technical College)
** 정회원, 광운대학교 제어계측공학과
(Department of Control & Instrumentation, Kwangwoon University)
※ 이 논문은 2009학년도 인하공업전문대학 교내연구비지원에 의하여 연구되었음.
접수일자: 2010년9월29일, 수정완료일: 2010년12월7일

## Ⅰ. 서 론

원자력, 화력, 수력 발전 설비와 화학 설비를 이루는 대규모 플랜트에서의 회전체 진동은 설비의 신뢰성 및 안전성을 예측하거나 검지하는 척도로 중요한 데이터이다.[1-2] 이러한 진동 데이터의 정확한 획득 및 분석 과정이야 말로 시스템 유지보수 측면에서 중요한 대상으로 여겨진다. 진동 데이터의 실시간 획득 및 자동 분석과정을 위한 시스템의 도입을 위해 많은 연구 및 개발이 진행되어 왔으며, 수많은 제품들이 생산 설비에 사용되어 조기 경보 시스템을 구축해 시스템의 안전성 및 신뢰성을 높여 생산성을 향상시켜왔다.[3-5] 이러한 회전체의 진동 데이터를 획득하기 위해 시스템은 아날로그와 디지털 로직으로 구성되었으며, 아날로그 로직은 아날로그 필터와 증폭기로 구성되고 디지털 로직은 ADC와 DSP, FPGA, FIFO 메모리로 구성되었다. 아날로그 로직을 통과한 회전체 신호는 FPGA에 의해 제어되는 ADC를 통해 입력되며, 이 데이터는 FIFO 메모리에 저장된다. 디지털 신호처리에 필요한 일정한 양의 데이터를 FIFO 메모리에 넣은 후, FPGA에서는 DSP에 신호를 보내 DSP에서 FIFO 메모리에 들어온 데이터를 이용하여 신호처리를 수행할 수 있도록 구성하였다.[5-6] DSP에서 처리되는 데이터 신호처리로는 실수 변환, Peak to Peak, 평균 값 산출, GAP, 디지털 필터, FFT 등을 처리하여 설정된 이벤트를 추적하거나 결과 값을 도출하여 조기 경보 시스템을 구축하였다. 모든 신호처리 과정 및 이벤트 추적은 DSP에서 이루어졌으며, 한 번의 신호처리 과정에서 여러 분석 단계가 이루어져서 각 분석 단계 마다 처리에 시간이 소요되며, 특정 이벤트에 따라 처리 소요 시간에도 변동이 발생한다. 또한 모든 데이터에 대한 실시간 분석 및 연속적인 분석이 이루지기 원하지만 현실적으로 불가능한 측면이 존재한다. 데이터는 계속해서 입력 받지만 DSP에서의 신호처리 기간 동안 입력된 데이터는 과거의 데이터로 다음 입력처리에서는 제외될 수 있으며, 물리적 메모리의 한계로 인해 모든 데이터를 저장 할 수도 없을 것이다. 특히 한 채널이 아닌 다수의 채널에서의 데이터를 획득하여 분석이 이루어지는 과정에서 각 채널당 하나의 DSP가 존재해야만 이러한 불연속 분석 시간차를 조금이나마 줄일 수 있다. 하지만 단 하나의 DSP만을 사용하여 여러 채널로부터의 입력 받은 데이터를 처리 시 여러 DSP를 사용하는 효과를 볼 수 있다면, 설계적인 측면과 경제적인 측면에서 많은 비용 절감 효과가 발생할 수 있다.[6~9] 본 연구에서는 이러한 여러 분석 단계에 의해 필연적으로 발생되는 시간을 최소한으로 줄일 수 있는 방법으로 DSP에서 수행되는 신호처리 단계 중 일부를 FPGA에서 처리할 수 있도록 함으로써 단일의 소프트웨어 처리에 의해 수행되는 분석 단계를 병렬 처리로 수행하여 실시간 데이터 처리를 수행하여, DSP만으로 구성된 신호처리보다 빠르고 안정된 신호처리 방법을 제시하려고 한다.

## Ⅱ. 본 론

### 1. DSP(Digital Signal Processor)와 ADC (Analog Digital Convertor)의 일반적인 설계

회전체 진동 획득 장치의 일반적인 시스템은 단일 채널로서 센서로부터 획득되는 신호를 그림 1과 같은 구조로 되어 있다. 센서로부터 들어오는 신호는 필터를 거쳐서 ADC에서 신호를 디지털로 변환시키고, 변환된 데이터는 DSP에서 필요한 데이터로 가공되며, ADC는 DSP에서 제어되고, DSP는 입력된 데이터를 RAM에 저장하고 저장된 일정량의 데이터를 가지고 신호처리를 수행한다. 이때 ADC의 샘플링 속도가 100Kbps일 경우 일정량의 연속된 데이터를 얻기 위해서 DSP는 정확히 10μS 이내에 처리 되어야 하며, 메모리에 쓰고 읽는 동작 시간도 고려하여야 한다. 만일 단일 함수 처리 시간

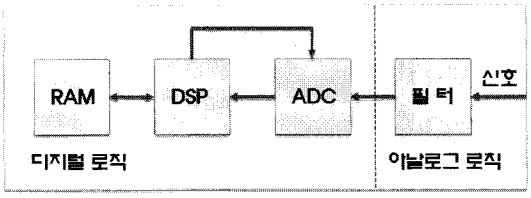


그림 1. 단일 DSP/ADC 구성
Fig. 1. Single DSP/ADC System.

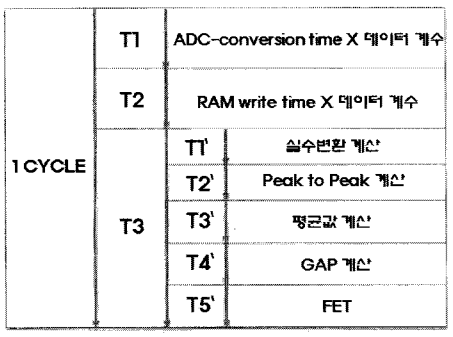


그림 2. 단일 DPS/ADC 구성 처리 시간
Fig. 2. Single DSP/ADC System Process Time.

이 10μS 이상인 루틴을 수행한다면 연속적인 데이터를 획득할 수 없고, 다만 데이터를 획득 시 다른 명령어 루틴을 수행하지 않을 경우에는 연속적인 데이터 획득이 가능하다. 또한 얻어진 512/1024/2048개의 데이터를 이용하여 신호처리를 수행하고, 신호처리가 모두 완료된 이후 다시 ADC 데이터를 저장 할 수 있다.

다중 채널에서 동시에 데이터를 획득하여 처리할 경우에는 그림 2와 같은 형태로 구성할 수 있으며, 한 개의 DSP는 4개의 ADC를 제어하여 획득한 데이터를 메모리에 저장한다. ADC를 제어하는 방식에 따라 제어 시간이 결정되고, 폴링 방식으로 4개의 ADC 데이터를 획득하여 메모리에 저장하는 방식인 경우에는 정확히 일정시간마다 ADC 데이터를 얻어야 하므로, 타이머로 프로그램을 구성 하여야 하며, 이 경우 인터럽트에 소요되는 처리 시간과 데이터 획득 및 저장에 소요되는 처리 시간이 한 채널당 2.5μS 이내에서 수행되어야 한다. 그리고 한 개의 DSP가 그림3과 같이 4개의 센서로부터 들어오는 신호를 처리하게 되며, 이에 따른 연산 소요시간은 4배가 된다.

그림 4는 단일 DSP에서 4개의 ADC를 처리하기 위

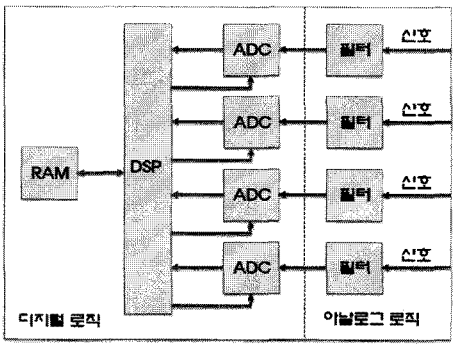


그림 3. 단일 DSP 다중 ADC 구성
Fig. 3. Single DSP Multi ADC System.

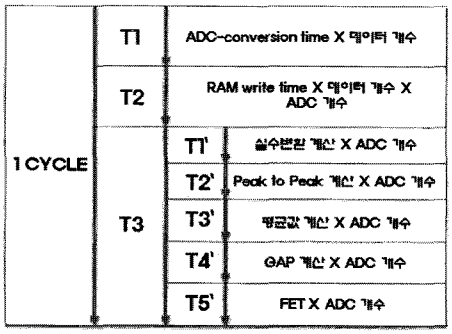


그림 4. 단일 DSP에서 ADC 구성 처리 시간
Fig. 4. ADC Process Time of Single DSP Multi ADC System.

한 처리 절차를 나타낸 것이다. 전체를 한 사이클로 구분하며, 한 사이클은 원하는 데이터 수에 대한 ADC의 컨버젼 시간(T1), ADC에서 획득된 데이터를 원하는 메모리에 저장하는데 걸리는 시간(T2), 연산 처리 시간(T3)로 구분된다. 연산 처리 시간은 실수변환 시간(T1′), Peak to Peak 계산 시간(T2′), 평균값 계산 시간(T3′), GAP 계산 시간(T4′), FET 처리 시간(T5′)로 구성된다.

한 개의 DSP로 처리 할 경우 전체 신호처리에 걸리는 시간이 ADC의 개수만큼 증가하므로, 실시간 진동 감시 및 분석을 요하는 시스템에서는 빠른 연산 처리가 요구된다.

그리고 그림 5와 같은 센서 신호를 각각 획득할 수 있도록 다중 DSP/ADC 구성을 사용하면 처리 시간을 단축시킬 수 있다. 따라서 그림 1과 같이 설계를 될 때에는 센서로부터 연속적인 데이터를 획득하여 그 데이터를 저장하여야 하는데, 만일 임의의 인터럽트 발생으로 인한 데이터를 연속적으로 하지 못하는 경우가 발생하게 된다.

이러한 문제는 DSP가 ADC 제어에 대한 항상 책임이 있기 때문이다.

DSP는 데이터의 획득 주기마다 ADC를 구동하여야 하며, 획득된 데이터를 FIFO에 써넣는 동작도 ADC에

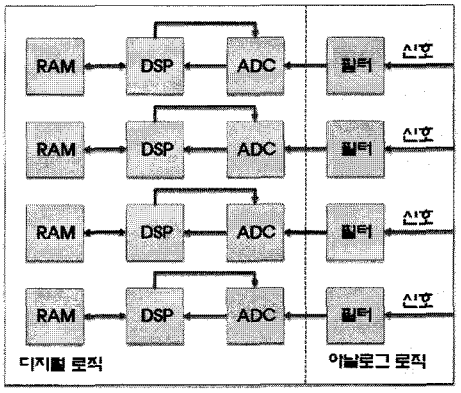


그림 5. 다중 DSP/ADC 구성
Fig. 5. Multi DSP/ADC System.

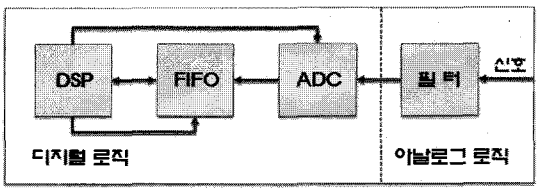


그림 6. 단일 DSP/FIFO/ADC 구성
Fig. 6. Single DSP/FIFO/ADC System.

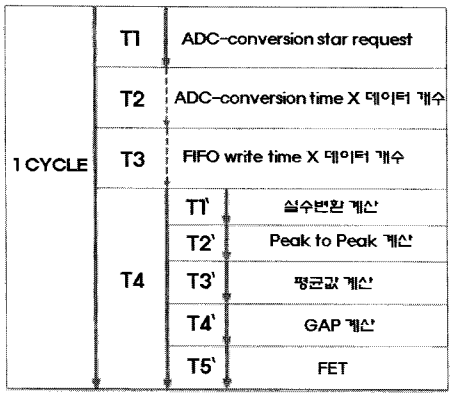


그림 7. 단일 DSP/FIFO/ADC 구성 처리 시간
Fig. 7. DSP/FIFO/ADC Process Time.

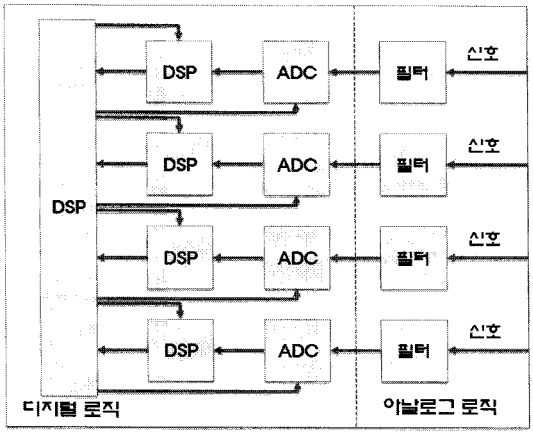


그림 8. 단일 DSP 다중 FIFO/ADC 구성
Fig. 8. Single DSP Multi FIFO/ADC System.

서 처리하게 된다. 즉 10us 마다 이러한 작업은 DSP에 의해 처리되어야 하며, 처리 시 인터럽트 및 다른 수행에 의해 중단될 수 있어서 ADC의 처음 시작만 요청 후 일정 주기 마다 계속해서 데이터를 FIFO로 송출할 수 있는 ADC를 구성하여야 한다. 데이터를 획득하는 동안 DSP는 다른 루틴을 처리 할 수 있으며, FIFO가 FULL 인 경우 이를 인지하여 DSP는 신호처리를 수행하여야 한다. 신호처리 완료 후 FIFO를 비우고, 다시 ADC를 동작시켜 데이터를 얻도록 한다. 연속적인 데이터 획득과 처리 시간의 단축을 위하여 단일의 DSP와 다수의 ADC를 사용하는 방법에서 더 나아가 한 채널당 하나의 DSP를 사용하여 처리 시간을 단축할 수 있도록 그림 6, 7과 같이 구성된다. 그림 8과 9에서는 입력 신호 한 채널당 DSP, ADC가 독립적으로 구성된다.

그림 8에서는 그림 2와 같은 처리 시간이 소요되며, 그림 9에서는 그림 7과 같은 처리 시간이 소요된다. 채널이 다수로 확장되더라도 DSP에서 처리되는 시간을

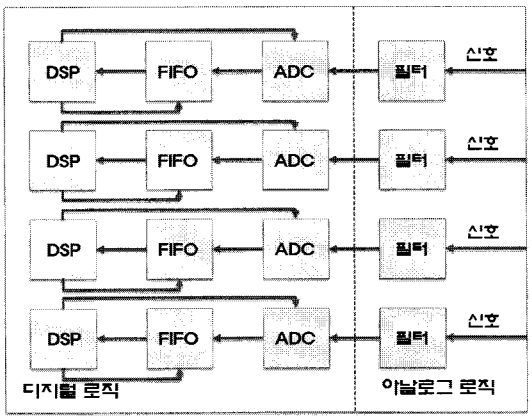


그림 9. 다중 DSP/FIFO/ADC 구성
Fig. 9. Multi DSP/FIFO/ADC System.

일정하게 유지할 수 있는 구조라 하겠다.

## 2. 회전체 진동을 획득하기 위한 DSP/ADC 설계

회전체의 진동 신호를 획득하는 방법에는 회전체의 회전 속도를 임의로 설정하여, 이 기준에 의해 신호의 값을 획득하는 방법과 회전체의 회전 속도에 비례하는 기준 신호, 즉 가변 신호에 의해 신호의 값을 획득하는 방법이 있다. 전자의 경우는 앞에서 언급한 일반적인 ADC 처리 로직에 의해 데이터를 얻을 수 있지만, 후자의 경우에는 동기 신호를 생성, 생성된 동기신호에 맞춰 데이터를 획득하여야 하기 때문에 일반적인 ADC 처리 로직으로는 정확한 데이터를 얻기 힘들다. 회전체의 회전 속도에 대한 동기신호를 생성해 주는 로직을 Keyphaser라고 하며, 이 로직에 의해 회전체의 RPM을 측정, 동기 신호를 생성해 낼 수 있다. 이 RPM 동기 호를 수신하여RPM을 측정하고, 측정된 값의 범위에 따른 ADC의 샘플링 속도를 자동으로 결정할 수 있도록 해야 한다. 하나의 Keyphaser 모듈에서는 4개의 RPM 신호를 생성하여, 각 ADC 모듈로 전송한다.

그림 10은 RPM 신호를 나타낸 것이다. RPM 신호와 동기화하여 샘플링이 이루어져야 회전체의 위상각을 계산할 수 있으며, RPM 신호를 DSP에서 입력 신호로 처

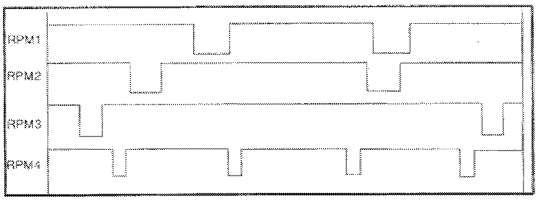


그림 10. Keyphaser에서의RPM 출력 신호
Fig. 10. RPM Output Signal of Keyphaser.

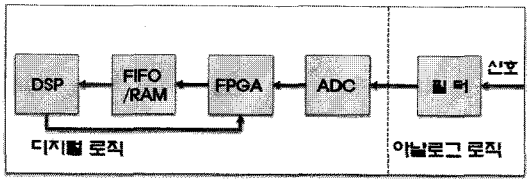


그림 11. 단일 DSP/FIFO/FPGA/ADC 구성
Fig. 11. Single DSP/FIFO/FPGA/ADC System.

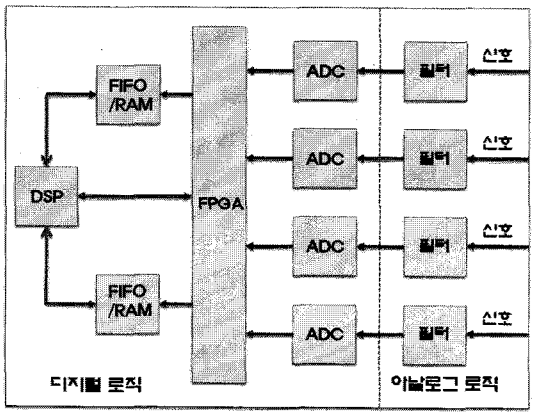


그림 12. 단일 DSP/FPGA 다중 ADC 구성
Fig. 12. Single DSP/FPGA Multi ADC System.

리하여, 동기화 신호로써 동작하면 위상각 계산에 큰 오차가 발생한다. RPM은 분당 회전수로 RPM 신호의 주기를 측정하고 1분에 대해 이 측정된 값으로 나누면 계산된다. RPM신호와 동기하여 정확한 데이터 값을 획득하기 위해서는 그림 11과 같은 로직으로 구성하여야 한다. RPM 신호는 FPGA로 입력되며, 입력되는 신호는 바로 RPM 값을 계산하고, 이 신호에 동기되어 ADC의 값을 획득한다. 그림 12, 13, 14는 다수 채널 입력으로 구성하였을 경우의 구성을 나타낸 것이다.

그림 11의 구조는 DSP와 ADC를 이용한 일반적인 설계 구성에서 FPGA를 추가한 것이다. 이는 FPGA를 이용하여 설계하면, 회로의 수정 없이 개발자가 필요한 로직을 변경, 추가하기 쉬운 설계 기술을 적용할 수 있으며, RPM이라는 동기화된 신호를 정확하게 인식하기 위하여 독립적인 수행 로직이 필요하기 때문이다.

RPM의 값을 그림 10에서 보는 것과 같이 각 파형의 하강에지에서부터 다음 하강에지까지의 시간을 실시간으로 검지하여야 하는데, DSP에서 이 부분까지 같이 처리하기에는 제한 시간이내에 연산 로직을 수행하고 처리하기에 상당한 로드가 예상되며, 이로 인해 RPM의 순간적인 변화를 검지하지 못할 수도 있다. 즉 소프트웨어로 처리하기에는 신뢰성과 정확성면에서 많은 오류 및 에러를 유발할 수 있다. 그러므로 이 부분은 하드웨

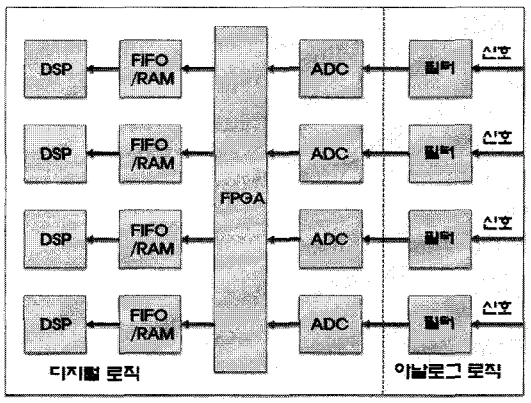


그림 13. 다중 DSP/FIFO/ADC & 단일 FPGA 구성
Fig. 13. Multi DSP/FIFO/ADC &Single FPGA System.

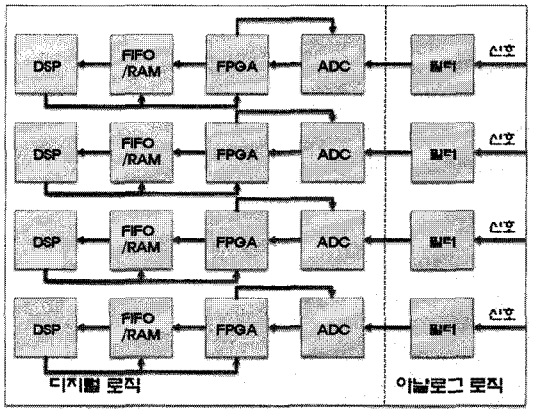


그림 14 다중 DSP/FIFO/FPGA/ADC 구성
Fig. 14. Multi DSP/FIFO/FPGA/ADC System.

어로 로직을 구성하여 처리함으로써, 신뢰성과 정확성을 높일 수 있다.

그림 12는 단일 DSP와 4개의 ADC에 FPGA를 사용하여 구성한 것이다. FPGA 에서는 각 ADC를 제어하는 로직을 가지며, DSP에서 FPGA의 내부 레지스터를 설정하여 ADC를 제어한다. ADC에서 컨버젼이 끝나면 FPGA에서는 ADC로부터 데이터를 읽어 FIFO에 저장한다. FPGA 로직은 일정 시간마다 계속해서 반복적인 기능을 수행하며, ADC로부터 획득된 데이터를 FIFO에 넣는 책임을 갖는다. DSP에서 분석에 필요한 데이터가 2개의 FIFO중 한쪽에 모아지면, FPGA는 다음 FIFO로 데이터를 써 넣는다. DSP에서 한쪽 FIFO/RAM에서 데이터를 읽어 처리하면, 다른 FIFO에 데이터를 저장하여, 데이터의 손실을 최소한으로 하기 위함이다. 이를 위해서는 DSP에서의 처리가 FIFO에 데이터가 모두 모아질 때 까지 완료해야 한다.

그림 13과 14의 구성은 그림 12에서의 구성에서 DSP가 각 채널별로 존재하는 구성이다. 또한 그림 14에서

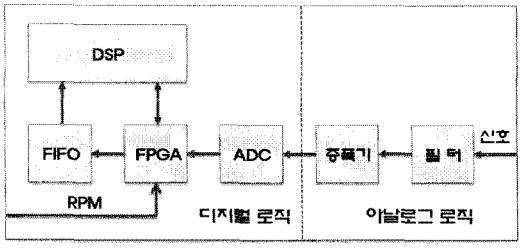


그림 15. 설계 구성도
Fig. 15. Design System.

는 FPGA가 채널별로 독립되어 존재한다. 이는 FPGA나 DSP 고장 시 전체 시스템의 고장을 막기 위함이다. 단일의 FPGA만을 사용하여 구성할 경우, 일부 내부 로직의 손상이 전체 시스템의 고장으로 이어질 수 있기 때문이다. 각 채널별로 독립된 디바이스로 구성하는 방식이 신뢰성 면에서는 우수하지만, 경제적인 면을 고려하여 시스템 설계에 반영해야 한다.

## 3. 회전체 진동 데이터 획득 및 분석 시스템 설계

그림 15는 회전체 진동 데이터 획득 및 분석 시스템을 제작하기 위한 설계 구성도를 나타낸 것이다. 회전체 기계에서 장착된 진동 센서로부터 진동신호를 받으면 아날로그 로직에 있는 필터를 통하여 데이터를 획득하게 되고, 이 데이터는 고주파 노이즈를 제거하기 위한 부분으로 진동 신호 영역을 설정할 수 있도록 설계하였고, 증폭기는 회전축에서 발생되는 최초 신호의 영역은 0V~24V, -24V~0V 영역 대에서 발생하고, 이러한 신호를 입력하기 위해서 증폭기를 사용한다. 특히 ADC가 빠른 속도록 데이터를 획득할 수 있도록 신호의 크기를 변화시켜 주는 역할을 한다. ADC변환은 아날로그 신호를 디지털 신호로 변환해 주기 위한 컨버터 역할을 하며, FPGA는 ADC를 제어하기 위한 로직과

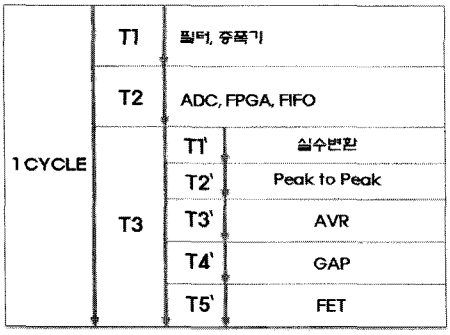


그림 16. 데이터 획득 및 분석 주기
Fig. 16. Data Acquisition & Analysis Period.

ADC에서 얻은 데이터를 FIFO에 쓰는 메모리 컨트롤 로직을 일정한 데이터로 FIFO에 써 넣은 후, 이를 DSP에게 신호를 전송하는 로직으로 구성하였다. 또한 DSP는 신호처리를 위한 프로세서로 입력한 신호를 전압형태인 실수로 변환하며, 실효값, 최대 진폭(Peak to Peak), 평균 값, GAP 전압, 디지털 필터 FIR, 고속푸리에 변환(FFT) 등을 소프트웨어로 처리하고, 그 처리 결과를 메모리에 전송한다. 그림 16은 그림 15에 나타낸 시스템에서 수행되는 전 과정을 시간의 흐름에 따라 도식화한 것이다.

회전체의 신호를 획득하여 분석하기 위해 필요한 물리적인 시간의 흐름을 나타내었다. (T1+T2) << T3로 나타낼 수 있듯이 전체 과정에서 T3과정의 시간 소요가 제일 크게 나타난다. 물론 빠른 DSP를 사용한다면 이러한 시간은 어느 정도 극복할 수도 있겠지만, 시스템 요구사항이 높아질수록 이를 극복하기 위한 다른 방법이 고안되어야 한다. 본 연구에서는 이러한 T3에 소요되는 시간을 줄일 수 있는 방법으로 FPGA를 사용하여 DSP에서 수행되는 연산 로직 일부를 FPGA가 처리할 수 있도록 했고, T3의 시간을 대폭 줄일 수 있도록 FPGA를 설계하는데 목적이 있다. 물론 기존 시스템에서도 FPGA를 사용하여 실시간 처리를 위한 로직을 구성하였으나, 이는 T3에서의 시간을 줄이는 것이 목적이 아니라 단순히 연속적인 데이터를 FIFO에 넣는 것을 목적으로 하고 있다. 그림17은 시스템의 FPGA 내부 구성을 도식화한 것이다. 본 연구에서는 이를 그림 18과 같이 구성하여 FPGA를 설계하였다. 그림 17에서 보이는 구성은 FPGA의 내부 구성이다. FPGA내부 구성은 외부 장치와의 인터페이스를 하기 위한 구조로 되어야 한다.

FPGA는 기본적으로 DSP와 FIFO, ADC와 연결되어야 하여 데이터를 송수신한다. DSP는 일반적으로 데이터 버스, 어드레스 버스, 컨트롤러 신호의 버스 방식을

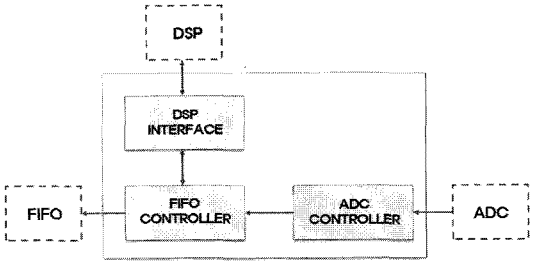


그림 17. 일반적인 FPGA 내부구성
Fig. 17. General FPGA Internal System.

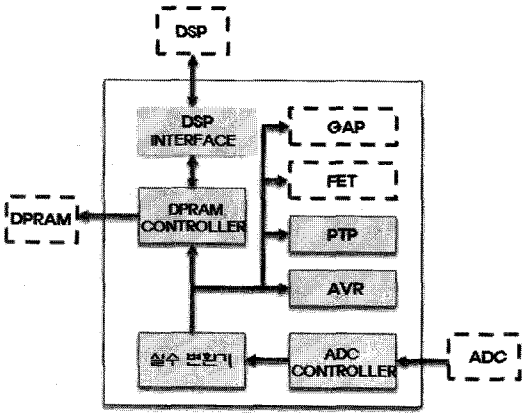


그림 18. FPGA 내부 설계 구성
Fig. 18. FPGA Internal Design System.

사용한다. FIFO와 ADC는 경우에 따라 데이터 버스 신호 방식과, 시리얼 통신 방식으로 데이터를 송수신 할 수 있다. 따라서 그림 17에서는 각 디바이스와 인터페이스 하기 위한 FPGA 내부 로직 구성을 정의하였다. ADC와 인터페이스를 담당하는 ADC 컨트롤러와 FIFO와 인터페이스 하기 위한 FIFO 컨트롤러, DSP와 인터페이스 하기 위한 DSP 인터페이스로 구성하였다. FPGA의 외부인터페이스와 별도로 내부 인터페이스와 로직은 그림 18과 같이 구성하였다.

그림 18에서 보이는 FPGA 내부 로직이 활성화되기 위해서는 일단 DSP에서 FPGA 내부 레지스터의 값을 설정하고 이 값에 따라 ADC 컨트롤러가 작동시켰다. ADC 컨트롤러는 설정된 값에 따라 일정 주기마다 ADC 컨버젼 값을 획득하고, 획득된 값은 실수 변환기로 전송되며, 실수 변환기에서는 설정된 로직에 따라 ADC의 2진 값을 실수 값으로 변환한다. 실수 변환기에서 변환된 값들은 평균을 계산하기 위한 로직인 AVG, 최고값과 최저값의 차이를 구하기 위한 PTP 로직 등으로 전달하고, 그 데이터를 저장하기 위해 DPRAM 컨트롤러에도 전송된다.

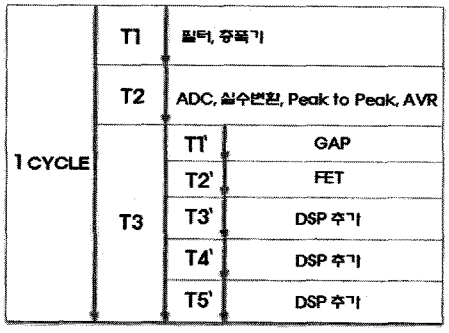


그림 19. 향상된 데이터 획득 및 분석 주기
Fig. 19. Improved Data Acquisition & Analysis period.

그림 18과 같이 FPGA를 구성하면, 기존 DSP에서의 프로그램 처리 시간을 단축할 수 있다. 그림 19는 최종 결과로 기존 시스템에서 1 사이클에 소요되는 데이터 획득 및 분석시간을 현저하게 줄일 수 있으며, DSP에서는 제한된 시간 이내에 다른 수행을 처리할 수 있다. 즉 DSP에서 수행하던 연산을 일부 FPGA에서 수행하도록 FPGA에 연산로직을 설계함으로써 가능해 진다.

### 3.1 실수 변환기 설계

ADC 컨트롤러에 의해 출력된 데이터는 24bit raw 데이터로 그 값은 표 2에 의해서 전압 값으로 환산되어야 한다. 이때의 입력 값은 실수연산을 위해서 floating point 32bit로 변환한 후 전압 값으로 환산하였다. 표 2에 의해 수식 (1) 도출하여 전압 값으로 환산하였으며, 실수의 변환은 ALTERA사의 라이브러리인 FP_ CNVT _ITF를 사용하였다.

표 1. ADS1274의 입력 값 대한 이상적인 출력 값
Table 1. Ideal Output Data on the Input of ADS1274.

| 입력 신호 $V_{IN}$(AINP - AINN) | 출력 코드(2) |
|---|---|
| $\geq +V_{REF}$ | 7FFFFFh |
| $\frac{+V_{REF}}{2^{23}-1}$ | 000001h |
| 0 | 000000h |
| $\frac{-V_{REF}}{2^{23}-1}$ | FFFFFFh |
| $\leq -V_{REF}\left(\frac{2^{23}}{2^{23}-1}\right)$ | 800000h |

$$ADC\ voltage(floating\ point) = \frac{2500(floating\ point)}{0x7fffff}(FP\_CNVT\_ITF) \quad (1)$$

(1) $V_{IN}$은 입력전압을 나타낸다.
(2) AINP는 양의 전압 값을 나타낸다.
(3) AINN은 음의 전압 값을 나타낸다.
(4) Vref은 기준접압을 의미하며, +5V
(5) 외란, INL, offset, 이득 에러는 배제한다.

### 3.2 최대/최소 전압 값 획득 로직 설계

실시간으로 입력되는 raw 데이터에 대해 임의의 시작점에서부터 설정된 총 데이터 중에서 최대와 최소값을 판별하는 로직이다. 기존의 DSP에서 수행될 시에는 총 데이터를 메모리에 저장해 놓은 상태에서 각각의 데

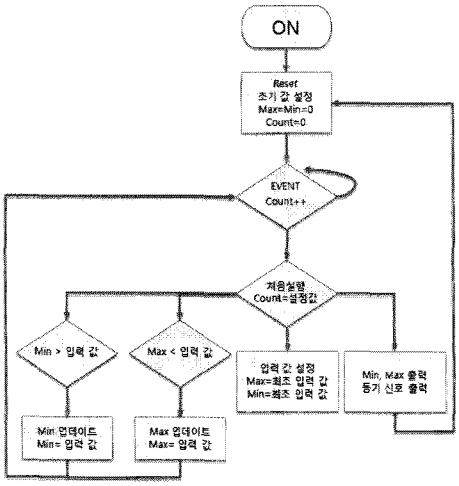


그림 20. 최대/최소값의 로직 순서도
Fig. 20. Logic Flow Chart of Max/Min Value.

이터를 비교 판단하였기 때문에 이를 처리하는 시간이 소요된다. 하지만 실시간으로 입력되는 데이터 하나하나에 대해서 이 로직을 실시간으로 실행하기 때문에 총 데이터에서 마지막 데이터를 입력 받은 후 수 Ms 이내에 최대/최소값을 출력한다. 그림 20은 최대/최소를 판별하기 위한 순서도이며, FPGA 내부에서 로직으로 구성되어 있다.

### 3.3 평균값 산출을 위한 로직 설계

입력된 모든 데이터를 한 번에 더하고 평균을 구하는 것보다 일단 데이터가 입력될 때마다 합산을 하고, 설정된 데이터 개수만큼 데이터가 모아지면, 최종적으로 총 데이터 수로 나누어 평균을 구할 수 있도록 로직을 구성한다. 그림 21은 이러한 로직의 순서도를 나타낸다.

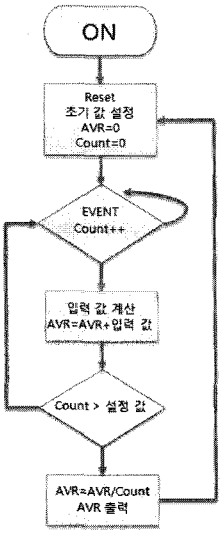


그림 21. 평균 계산을 위한 순서도
Fig. 21. Flow Chart for Average Value Calculating.

### 3.4 실효치 계산

실효치를 계산하는 과정은 평균을 내는 과정과 유사하다. 단지 입력된 데이터가 일단 제곱한 후 더해진다는 것과 마지막 합산의 평균 후 제곱근 연산자에 의해 계산되어야 한다는 것이다. 그림 22는 실효치 계산을 위한 순서도를 나타낸다.

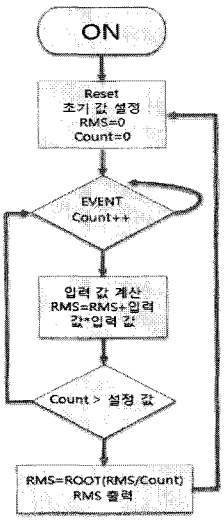


그림 22. 실효치 계산 순서도
Fig. 22. Flow Chart of RMS.

## Ⅲ. FPGA 설계 검증 및 시뮬레이션

본 시뮬레이션은 회전체 기계에 장착되어있는 변위 센서로부터 들어오는 신호를 100khz로 샘프링하여 데이터를 DSP에서 처리하였고, 그 함수의 알고리즘을 수행하는데 걸리는 시간을 측정한 결과는 다음 표 2와 같다. 실험은 데이터 셋의 크기를 변경하여 2회 실시 하였으며, 데이터 셋은 512개와 2048개 이다.

실험 결과는 512개의 데이터셋이 있을 경우가 2048의

표 2. DSP에서 처리된 소요시간
Table 2. Process Time of the DSP.

| 기능 | 함수명 | 512 | 2048 |
|---|---|---|---|
| Rms | aibCalcRms | 0.3ms | 1ms |
| PTP | aibCalcPp | 0.38ms | 1.37ms |
| Avr | aibCalcAvr | 0.26ms | 0.8ms |
| MAX | aibCalcMax | 0.24ms | 0.7ms |
| MIN | aibCalcMin | 0.24ms | 0.7ms |
| 총 소요시간 (ms) | | 1.42ms | 4.57ms |

표 3. 각 함수의 코드길이 따른 수행 시간
Table 3. The Process Time based on the Code Length of Each Function.

| 함수명 | 코드길이 [Word] | 수행시간 [Cycle] | 수행시간 |
|---|---|---|---|
| aibCalcRms | 452 | 467,684 | 1ms |
| aibCalcAvr | 376 | 192,056 | 0.84ms |
| aibCalcPp | 420 | 619,819 | 1.37ms |
| aibCalcMax | 356 | 188,258 | 0.79ms |
| aibCalcMin | 356 | 188,817 | 0.79ms |
| 총 소요시간 (ms) | 1,960 | 1,656,634 | 4.79ms |

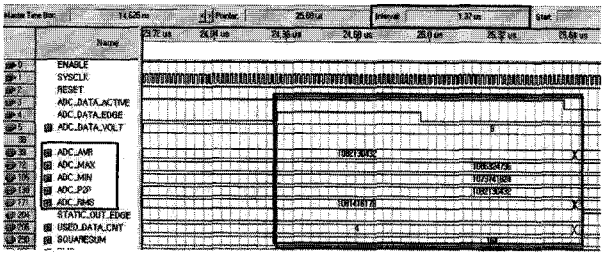

그림 23. Avr/MAX/MIN/PTP/RMS 출력 파형
Fig. 23. The Out Signal of Avr/Max/MIM/PTP/RMS.

경우보다 시간이 적게 소요된다. 전체적으로 DSP에서 소요되는 시간은 비슷하게 걸리는 것으로 결과가 나왔으며, 그중에서 Peak to Peak값이 더 많은 시간이 소요되고 있다. 각 함수를 수행하는 코드에 따라 데이터 셋의 개수를 2048을 기준으로 실행하여 실험하였고, 그 결과 각 함수의 소요되는 시간은 표 3과 같이 총 소요시간이 4.79ms가 된다.

그림 23은 실수로 변환된 전압의 Peak-to-Peak, 실효치, 평균값, 최대, 최소값들이 계산되는 과정에서의 결과 파형을 나타내고 있다. 즉 전체의 실수로 변환된 함수들의 처리시간이 병렬로 동시에 이루어지면서 전체 소요시간은 1.37usec 이다.

## Ⅳ. 결 론

본 연구에서는 회전체 진동데이터 획득에 대해서 DSP와 ADC를 이용하여 설계하고자 할 때, 고속으로 연속적인 데이터를 효율적으로 얻기 위한 방법을 제시하고자 하였다. 이를 위해 FPGA를 사용하여 DSP에서 처리해야 할 소프트웨어 연산 과정을 하드웨어 로직으로 구현하여 신뢰성을 높이기 위한 방법을 모색하였다. DSP에서 ADC를 제어하고, ADC에서 얻어진 2진 데이터를 실수로 변환하며, 각 연산과정에 필요한 소프트웨어 프로그램을 수행하던 과정을 하드웨어 로직으로 설계하여 DSP에서 처리하던 과정을 FPGA에서 실시간으로 처리하도록 하여 DSP의 처리되는 시간이 4.79msec 시간을 1.37usec으로 대폭 줄였으며, 이로 인해 DSP에서는 새로운 로직을 더 수행할 수 있는 시간을 확보할 수 있었다. 이러한 것을 가능하게 한 것은 FPGA를 사용하여 설계를 다양한 방면으로 모색할 수 있었기 때문이다. FPGA는 고정된 형태가 아닌, 프로그램 가능한 반도체 소자이기 때문에 능동적인 하드웨어 개발이 가능한 소자이며, 본 연구에서는 이 같은 점을 적극 활용하여 회전체 진동의 실시간 데이터를 획득하여 보다 진동 감시자 및 분석가들이 실시간으로 데이터를 효율적으로 처리할 수 있도록 하드웨어적으로 방법을 제시하였다.

## 참 고 문 헌


[1] Condition-based Maintenance and Machine Diagnostics Williams, Davies / Drakes 1994

[2] B.K.N. RAO Handbook of Condition Monitoring, 1996

[3] 공호선외, 통합 기계 상태 모니터링 기반기술, 한국과학기술원 2001-12-31

[4] Study on Probability Distribution and Criterion of Vibration Data for Condition Diagnosis of Rotating Machinery, Mitoma, Journal of the Society of Plant Engineers Japan, 2007, v.19 no.2 pp.106-113

[5] Monitoring and vibrational diagnostic of rotating machinery in power plants, Power Station Maintenance - Profitability Through Reliability, 1998. (Conf. Publ. No. 452)

[6] Zhang, A. H. Monitoring and Diagnostic Technique for Mechatronic Equipment , 1995.

[7] Digital Signal Processing and Applications with the C6713 and C6416 DSK By Rulph Chassaing ISBN 0-471-69007-4 Copyright © 2005 by John Wiley & Sons, Inc.

[8] J. W. Cooley, The structure of FFT and convolution algorithms, from a tutorial, *IEEE 1990 International Conference on Acoustics, Speech, and Signal Processing*, Apr. 1990.

[9] 이정석외 2, 회전체 진동 데이터의 AC/DC 성분 데이터 획득 및 분석 장치, 디지털산업정보학회, Dec. 2009.

## 저 자 소 개


**이 정 석(정회원)**

1985년 광운대학교 전기공학과 학사 졸업
1990년 광운대학교 전기공학과 석사 졸업
2001년 광운대학교 제어계측공학과 박사졸업
1990년~1997년 국방과학연구소 선임연구원
2002년~현재 인하공업전문대학 메카트로닉스과 부교수.
<주관심분야 :제어계측, 자동화 설비, 회전체 진동>

**유 등 열(정회원)**

2002년 광운대학교 제어계측공학과 학사 졸업
2001년~현재 광운대학교 제어계측공학과 석박사 과정
<주관심분야 :제어계측, 자동화 설비, 반도체 설계, 임베디드 OS>