

고속 디지털 데이터를 위한 FDE의 구현

심재욱, 김종훈
승실대학교

The Implementation of Fractional Delay Element for High Speed Digital Data

Jaewook Shim, Chonghoon Kim

School of Electronic Engineering, Soongsil University, Seoul, Korea

E-mail : jacouk@amcs.ssu.ac.kr

Abstract

현재 우리가 사용하고 있는 대부분의 시스템들은 대용량의 데이터를 송수신하고 있다. 대용량의 데이터를 전송하는 방법에는 여러방법이 있으나 한정되어 있는 대역폭을 사용하여 전송하기 위한 방법으로는 고속 전송을 사용한다. 많은 양의 데이터를 고속으로 전송을 하다 보면 여러가지 원인으로 인해 발생하는 지연에 대한 보정이 어려워 지게 된다. 이런 문제를 해결할 수 있는 방법중에 한가지가 바로 FDE(Fractional Delay Element)이다.

FDE는 1Clock 이하의 지연을 주는 소자로써 클럭 단위의 보정의 문제점을 해결한 것이다. 시스템 클럭을 고속으로 동작시키기에는 소자의 문제점이 있으나 FDE를 사용하면 시스템 클럭을 변화 없이 지연 보정을 할 수 있다.

본 논문에서는 VHDL 코딩과 FPGA를 사용하여 FDE를 구현 하였다. FDE의 중요한 역할을 하는 FDF(Fractional Delay Filter)를 VHDL로 코딩을 하였다.

되고 있다. 그 이유는 FD를 사용하지 않는 통신 시스템을 사용하고 있다고 가정하였을 때, 시스템 클럭보다 빠른 데이터가 전송이 되어야 할 경우 무선이란 특수성에서 발생하는 여러 가지 지연에 대한 보정이 어려워진다. 그로 인해 통신 시스템, 특히 수신단의 모든 구성을 바꿔야 하는 경우가 발생할 수도 있다. 그러나 ADC나 DAC와 같은 디지털 소자의 발전이 통신 시장에서 원하는 속도에 맞추어 발전을 하지 못하는 상황이기 때문에 교체가 어려워지는 경우가 있다. 이런 경우 FD가 가능하다면 통신 시스템의 교체없이 고속의 데이터를 처리할 수 있다.

그림 1은 본 논문에서 FDE를 구현하고자 하는 위치를 나타낸것으로 안테나를 통해 입력된 RF(Radio Frequency) 신호를 Modem으로 보내기 위해서는 Interface적인 역할을 하는 IF(Intermediate Frequency)를 사용하게 되는데 이부분에 Polyphase 방식의 Interpolation 필터를 이용하여 FD를 구현하는 것을 소개하고 FPGA를 사용하여 FDE를 구현 하였다.

I. 서론

FD(Fraction Delay)는 한 샘플주기당 여러 개의 지연을 시키는 것으로 이것에 관한 연구는 많이 이루어졌다. 그 중에서 많은 부분을 차지하고 있는 것이 디지털 필터, 즉 FIR을 사용하는 방법으로 고속의 발전을 하고 있는 무선 통신에서는 FD에 관한 것이 중요하게 인식

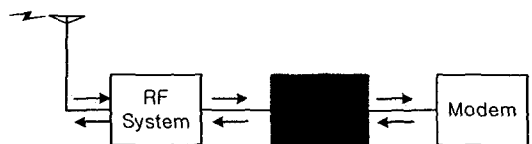


그림 1. FDE의 구현 위치

본 논문의 구성은 다음과 같다. 2절은 FDE의 중심적인 역할을 하는 FDF에 대한 기본 원리 및 구현 방안

에 대하여 설명한다. 3 절은 FDF 를 통해 FDE 를 구현 하기 위한 실험 구성도에 관해 서술한다. 4 절은 실험 구성도를 통해 검증하고자 하는 FDE 의 성능의 결과이고 마지막으로 5 절에서 결론을 맺는다.

II FDF 의 기본 원리 및 구현.

2.1 FDF 의 기본 원리

그림 2 은 FDF 의 기본 원리를 간략히 도식화한 것이다.

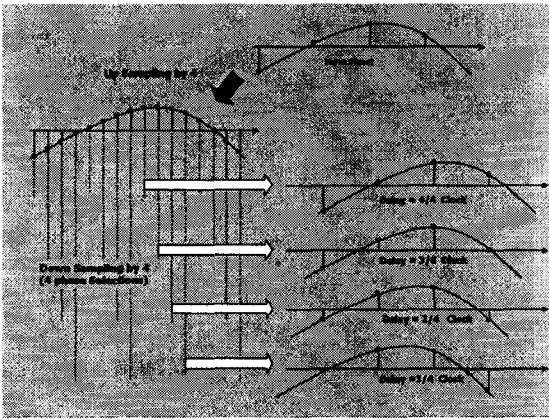


그림 2. FDF 기본 원리

그림 2 은 4 채널의 예를 들어 설명한 것으로, 입력 되는 원 신호를 4 배 Up sampling(Interpolation)을 한 다음 다시 4 배 Down sampling(Decimation)을 하면서 Down sampling 하는 위치를 선택하여 원하는 FD 를 얻는 것이다.

2.2 FDF 의 구현 방안

그림 3 은 FDF 의 구현 방안을 도식화한 것이다.

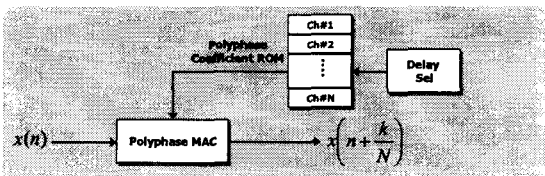


그림 3. FDF 구조

각 채널에 필요한 필터 계수를 별도의 ROM 에 저

장하고 MAC(Multiplication Accumulation)구조의 Polyphase 필터 한 개를 구현한 다음 선택적으로 각 채널에 필요한 필터 계수를 제공하여 출력하는 형태이다. 본 논문에서 FDF 의 구현은 그림 3 과 같은 구조를 사용하고 채널을 선택해 주는 역할을 하는 Delay sel 은 PCI 를 통해 컴퓨터로 제어를 한다.

2.3 FDF 의 구현

본 논문에서 구현하고자 하는 FDF 는 64 채널을 선택하고 통신 시스템을 가정하였기 때문에 기저대역을 처리하기 위해 Polyphase Lowpass Interpolation 필터를 구현하였다. 설계 사양은 그림 4 와 같다.

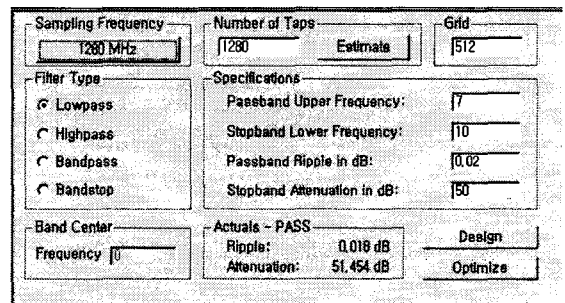


그림 4. 설계 사양

그림 4 은 ScopeFIR 이라는 프로그램을 사용하여 필터 계수를 구하기 위한것으로 시뮬레이션의 결과 값은 그림 5 이다.

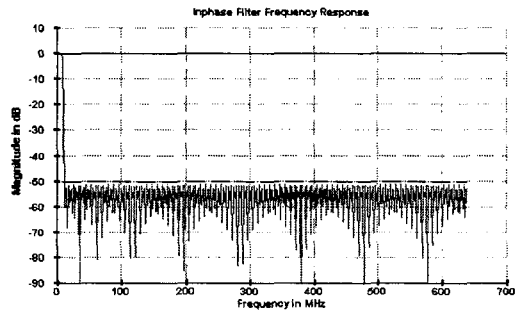


그림 5. 시뮬레이션 결과

시뮬레이션 결과값을 가지고 실제 Interpolation 필터를 구현하기 위해서는 시스템 클럭이 1280MHz 가 필요하게 되지만, Polyphase 방식을 사용하기 때문에 실제로는 필요하지 않게 된다.

그림 6 의 측정 결과는 3 번째 채널의 특성을 나타낸 것으로, 시스템 클럭이 80MHz(4x)이고 FDF 에 입력되는 표본화율 20MHz(1x)의 신호를 Polyphase MAC 구조에 PCI 를 통해 필터 계수를 입력하여 얻은 결과로 입력신호를 64 배 Interpolation 한 후 다시 64 Decimation 한 효과를 얻는 것을 확인할 수 있다.

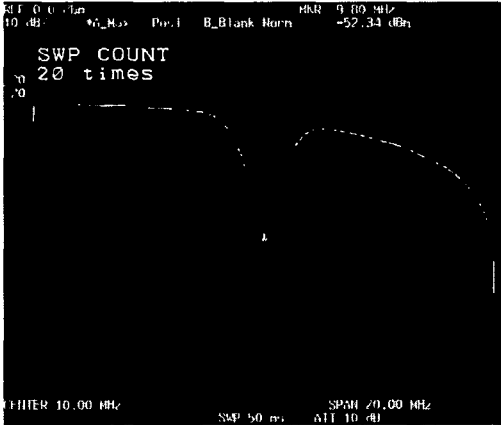


그림 6. FDF 측정 결과

III. FDE 의 구현을 위한 실험 구성도.

3.1 Test Board

그림 7 은 FDE 를 구현 및 성능 검증을 하기 위한 실험 구성도를 간략히 도식화한 것이다.

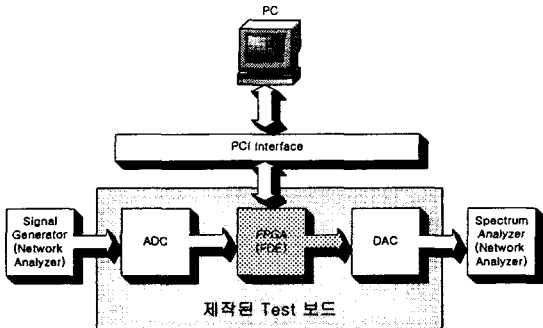


그림 7. 실험 구성도

실험 구성도에서 Signal Generator 와 Spectrum Analyzer 는 실험하기 위해 제작된 모든 구성 요소의 특성을 검증하기 위한 장비이고, 실제로는 Delay 측정을 위해 Network Analyzer 를 사용하였다.

그림 8 는 FDE 의 구현 및 검증을 위하여 제작된

Test Board 이다.

FDE 의 역할을 하는 것은 FPGA 이고 VHDL 로 Coding 하였다. ADC,DAC 는 FPGA 가 Digital 소자 이기 때문에 Digital 신호로 바꿔주는 역할을 한다.

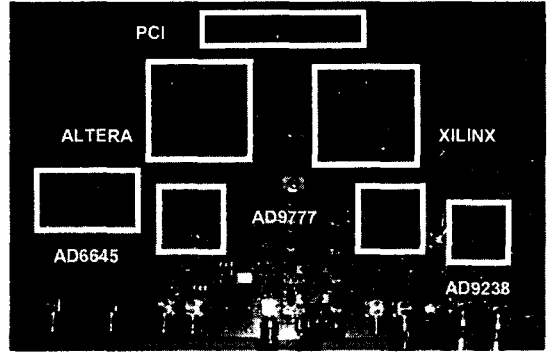


그림 8. Test Board Layout

3.2 VHDL 코딩

성능 검증을 위한 방법으로 본 논문에서는 일반적인 무선통신 시스템을 가정하고 실험을 하였기 때문에 다음과 같은 과정을 거쳐 성능 검증을 하였다.

- 1) 입력신호는 140MHz 의 단일 톤의 신호를 사용하고 20MHz 의 NCO 를 사용하여 기저대역으로 내린다.
- 2) 기저대역의 신호를 4x 에서 1x 로 표본화율을 하향 변환시켜야 한다. 이때 2 단의 Decimation 필터를 사용한다. 이때 처음 단계 즉 4x 에서 2x 로 변환시키는 것은 HBF(Half Band Filter)를 사용하고 2x 에서 1x 로 변환은 FIR(Finite Impulse Response Filter)를 사용한다.
- 3) 1x 로 하향 변환된 신호를 FDF 에 통과시킨다.
- 4) FDF 를 통과한 1x 신호를 다시 DAC 로 입력시키기 위해 4x 로 상향 변환을 시키는 것으로, 이때 사용되는 것은 2 단의 HBF 를 사용한다.
- 5) 기저대역의 신호를 다시 입력 신호와 같은 신호로 올려주는 것이다. 이때는 내릴때와 동일한 NCO 를 사용하여 Complex Modulation 을 해 준다.

이제 까지 설명한 단계를 도식화한 것이 그림 9 이

다.

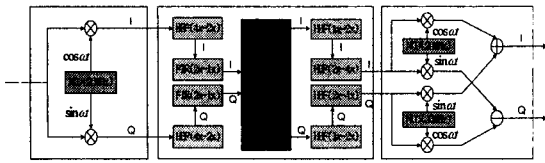


그림 9. 성능 검증을 위한 블럭도

IV. FDE 실험 결과

3 절에서 서술한 방법으로 FDF 와 다른 블럭을 VHDL 로 코딩을 후 FPGA 를 사용하여 FDE 를 구현 후 Delay 를 측정 하였다.

그림 10 은 140MHz 0dB 의 신호를 Network Analyzer 를 사용하여 FDE 의 Delay 와 Magnitude 를 측정함으로써 그림에서 위의 것이 Delay 이다.

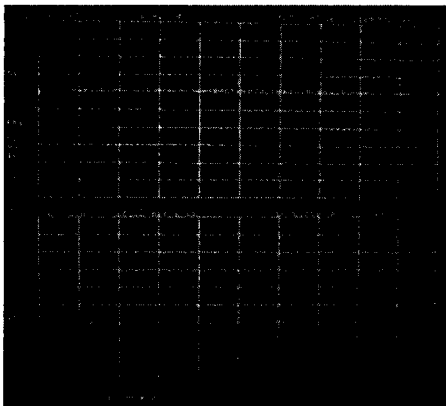


그림 10. 채널 1의 측정 결과

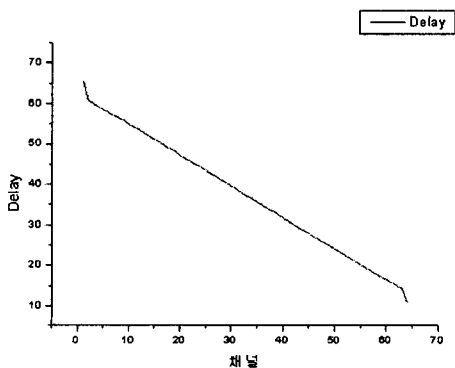


그림 11. 각 채널의 Delay 측정 결과

그림 11 은 각 채널 별로 Delay 를 측정한 것으로 이때의 측정 주파수는 140MHz 로 동일하고 입력신호는 50MHz 부터 400MHz 로 Sweep 을 하도록 하였으며 입력 신호의 Power 는 0dB 로 하였다. Sweep time 은 500ms 이다.

그림 11 에서 세로축은 Delay 를 나타내는 것으로 단위는 ns 이다. 본 논문에서는 시스템 클럭이 80MHz (12.5ns)이지만 실제 구현된 FDF 는 1x 인 50ns 이다. 측정된 Delay 를 살펴 보면 50ns 안에 선형적으로 각 채널 이 Delay 된 것을 확인할 수 있다. 그러나 채널 1 과 채널 64 는 비선형적인데, 이것은 그림 10 에서 보는 것과 같이 Ripple 의 원인으로 생각된다.

V. 결론.

본 논문에서는 무선 통신 시스템을 중심으로 Polyphase 방식의 Interpolation 필터를 이용하여 FDE 를 구현해 보고 실제 하드웨어를 통해 정확한 Delay 를 측정하였다.

제작된 하드웨어는 일반적으로 사용이 가능한 FPGA, ADC, DAC 들을 사용하여 제작이 되었기 때문에 일반적으로 사용이 가능하다고 사료된다. 더구나 FDE 의 구현 방법이 외부적으로 쉽게 제어가 가능할 뿐 아니라 필터 계수의 변화에 따라 채널 수의 변화가 용이하기 때문에 데이터의 속도 변화에 따른 변동이 가능하다. 이런 이유에서 기존의 여러 방식보다 월등히 효율적이라고 할 수 있다.

참고 문헌

- [1] Ronald E. Crochiere Lawrence R. Rabiner, "Multirate Digital Signal Processing." : Prentice-Hall signal processing series Alan Oppenheim.
- [2] N.J.Fliege, "Multirate Digital Signal Processing." : JOHN WILEY & SONS.
- [3] Alan V.Oppenheim, Ronald W.Schafer with John R.Buck, "Discrete-Time Signal Processing." : Prentice Hall Signal Processing Series.
- [4] William J. Dally, John W. Poulton, "Digital Systems Engineering." : Cambridge University Press.