

마이크로 컴퓨터에 의한  
Fast Walsh Transform에 관한 연구

\*  
유상진 오민환 제영무 최승욱 안두수  
동자연 수자원공사 충주 공대 삼근관 대학교 전기공학과

Realization of Fast Walsh Transform by using a micro-computer

\*  
S. J. Yoo M. H. Oh Y. M. Chai S. W. Choi D. S. Ahn  
KIER KOWACO Chungju Tech Sung Kyun Kwan University

Abstract

In recent years, aided by the power and capability of digital computation, the techniques of Walsh Transform have been exploited for applications in communication and signal processing. This paper presents an approach of FWT by using a 16-bit word-length micro-computer. This FWT implements an in-place decimation-in-sequence algorithm which improves processing speed and memory storage. Several examples illustrate the process and demonstrate the power spectrum of FWT and that of FFT for the waveforms

1. 서론

종전에는 Fourier함수가 전기계통의 R,L,C 회로 및 기타 선형회로의 신호분석용으로 많이 이용되었지만, 급진적인 반도체소자 개발에 따른 스위칭회로의 본격적인 출현으로 신호처리 분석을 직교비정현(Orthogonal-Nonsinusoidal)함수인 Harr, Walsh 그리고 Radamacher함수를 활용하게 되었다.[1]

이런 직교비정현함수들은 현재 레이더시스템, 벨리비전, 분광학(Spectroscopy), 패턴인식, 데이터전송 및 압축분야에 널리 적용되고있는 실정이다. 특히 이런 함수들중 신호처리시 함수의 음으로 해결하여야 할 사항을 Walsh함수에서는 +1과 -1의 단순한 가감산으로 계산이 가능하므로 이를 이용한 이용분야가 증대될 전망이다.[2]

2. Walsh함수의 이론적해석

2.1. Walsh함수

월쉬함수는 그림1과같이 직교 정방형(Orthogonal Rectangular)형태의 크기가 +1과 -1인 값을 가지며, 순환관계식(Recursive Relation)을 이용하여 표현하면 다음과 같다.[3]

$$WAL(2j+p, t) = (-1)^{j(p/2)} \{WAL[j, 2t] + (-1)^{jp} WAL[j, 2(t-1/2)]\}$$

----- (1)

여기서 구간  $0 \leq t < 1$ 에서는  $p=0$  또는  $1$ ,  $j=0, 1, 2, \dots$ 이 그러므로  $0 \leq t < 1$ 에서는  $WAL(0, t)=1$   $t > 1, t < 1$ 에서는  $WAL(0, t)=0$ 이 된다.

그림에서 Walsh함수는

- 1) Sin과 Cosine함수의 특성을 가지며
- 2) n차항의 함수는 n-1개의 영교차(zero crossing)를 하며

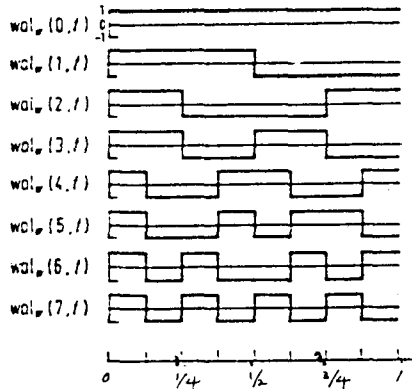


그림1. Walsh함수로 전개한 Walsh함수(N=8)

- 3) 각 함수들은 구간의 중심점을 기준으로 하여 우함수 또는 기함수의 형태를 가지며 같은 크기의 자형형태를 취한다.

2.2 Walsh 함수의 변환

만일 구간 (0,1)에서 신호 x(t)를 walsh급수로 표시하면 아래와 같다.

$$x(t) = a_0 + a_1 wal(1, t) + a_2 wal(2, t) + \dots \quad (2)$$

여기서 계수  $a_n$ 은

$$a_n = \int_0^1 x(t) wal(n, t) dt \quad (3)$$

이라고 한다.

상기 식으로부터 2개의 변환식을 구하면

$$x(t) = \sum_{n=0}^{\infty} x(n) wal(n, t) \quad (4)$$

$$X(n) = \int_0^1 x(t) wal(n, t) dt \quad (5)$$

이 된다.

위의 식의 적분항을 신속하게 처리하기 위하여 이산합(Discrete Summation)으로 변환하여 유한이산(Finite Discrete)로 월쉬변환하면 다음과 같이 표시된다.

$$X_n = \sum_{i=0}^{N-1} x_i \text{wal}(n,i) \quad n=0,1,2,\dots,N-1 \quad (6)$$

$$x_i = \sum_{n=0}^{N-1} X_n \text{wal}(n,i) \quad i=0,1,2,\dots,N-1 \quad (7)$$

이런 변환식을 N=8인 경우 signal flow diagram의 형태로 간단히 표시 하면 그림 2와 같다.

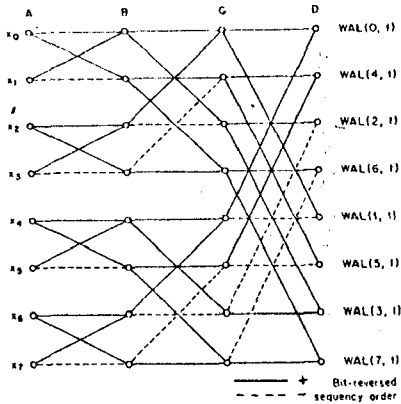


그림 2. 버터플라이형 월쉬 변환 흐름도

위의 그림은 in-place 알고리즘을 사용하였으며 복귀 +1과 -1만을 취하므로 부호로만 가산과 감산으로 계산이 가능하므로 신호 처리 속도가 비교적 빠르다. 이를 고속 월쉬 변환(FWT; Fast Walsh Transform)이라 한다. [4] 실제 식 X(n)을 구하기 위하여 FFT(Fast Fourier Transform)을 이용할 경우 Nlog N번의 가산과 곱셈 연산을 수행하여야 하지만 월쉬 함수를 이용하여 직접 변환 하려면 N번의 곱셈 연산과 N(N-1)번의 가감산 연산이 필요하다. 이에 대해 고속 월쉬 변환을 이용하면 Nlog N번의 가감산 연산만으로도 쉽게 구할 수 있다. [5][6]

그리고 월쉬 변환 함수의 power spectrum은 같은 영교차점을 갖는 월쉬함수 계수의 연산 본포로 나타내며 아래와 같이 정의된다. [7]

$$P(0) = X_c^*(0) \quad (8)$$

$$P(k) = X_c^*(k) + X_c^*(k)$$

$$P(N/2) = X_c^*(N/2)$$

여기서  $X_c$ 와  $X_s$ 는 월쉬 변환 함수 X(n)의 우수항과 기수항이 되며 스펙트럼의 포인트수는 N/2+1개가 된다.

### 2.3 FWT 알고리즘

일반적으로 어느 임의의 미분 방정식에 의해 전개될 수 있는 함수는 영교차와 순자항의 수에 의해 자수가 정하여 지지만 Walsh 함수의 계수들은

- 1) Radamacher 함수의 곱
- 2) Hadamard Matrix
- 3) Boolean 합성
- 4) 연속적인 곱(Continued Product)으로 전개될 수 있다.

이중 월쉬 함수를 연속적인 곱으로 표시 하면 다음과 같다.

$$\text{wal}(n,i) = \prod_{r=0}^{p-1} (-1)^{i_r + i_r} \quad (9)$$

여기서  $i, n = 0, 1, 2, \dots, N-1,$   
 $r = 0, 1, 2, \dots, p,$   
 $p = \log_2 N$  이며.

$$i = (i_{p-1}, i_{p-2}, \dots, i_0)$$

$$n = (n_{p-1}, n_{p-2}, \dots, n_0)$$

와 같이 이진 형태로 표시 된다.

상기식을 식 (6)에 대입하여  $X_m$ 을 구하면

$$X_m = \sum_{i=0}^{N-1} x_i \text{WAL}(n,i)$$

$$= \sum_{i=0}^{N-1} \left[ x_i \prod_{r=0}^{p-1} (-1)^{i_r + i_r} \right]$$

$$X_m = \sum_{i_{p-1}=0}^1 \sum_{i_{p-2}=0}^1 \dots \sum_{i_0=0}^1 \left[ x(i_{p-1}, i_{p-2}, \dots, i_0) \prod_{r=0}^{p-1} (-1)^{i_r + i_r} \right]$$

$$X_m = \sum_{i_{p-1}=0}^1 \sum_{i_{p-2}=0}^1 \dots \sum_{i_0=0}^1 \left[ \prod_{r=0}^{p-1} (-1)^{i_r + i_r} \sum_{i=0}^1 (-1)^{i_r + i_r} X(i_{p-1}, i_{p-2}, \dots, i_0) \right]$$

그러므로 월쉬 계수  $A_r$ 을 FWT 알고리즘으로 계산한 경우 r번째 까지 반복 계산 하여 표시 하면 아래와 같다.

$$A_0(i_{p-1}, \dots, i_0) = x(i_{p-1}, \dots, i_0)$$

$$A_1(n_{p-1}; i_{p-1}, \dots, i_1) = \sum_{i_0=0}^1 \left[ (-1)^{i_0 + i_0} X(i_{p-1}, \dots, i_0) \right]$$

$$A_r(n_{p-1}, n_{p-2}, \dots, n_{p-r}; i_{p-1}, i_{p-2}, \dots, i_r) = \sum_{i_{r-1}=0}^1 \left[ (-1)^{i_{r-1} + i_{r-1}} \cdot A_{r-1}(n_{p-1}, n_{p-2}, \dots, n_{p-r-1}; i_{p-1}, i_{p-2}, \dots, i_{r-1}) \right]$$

여기서  $r=p$  이면  $X_m = \frac{1}{N} A_p(n_{p-1}, n_{p-2}, \dots, n_0)$  이다.

이런 유도식으로 부터  $A_r$ 을 계산 하면 bit reversed order 형태로 출력 된다.

### 3. 시뮬레이션 구성

#### 3.1 하드웨어구성

본 실험에서는 IBM-PC/AT(주 기억 장치 640K; 보조기억장치 30MB)에 각종 입력 신호를 발생하는 Function Generator(peak to peak voltage:-5..+5V; 주파수 범위 0..79KHz; 출력 파형:정현파 및 구형파) 그리고 이의 신호를 받아 컴퓨터에 전달하는 12-bit A/D를 그림 3과 같이 구성하였다.

여기서 A/D는 12비트의 충분한 분해능을 가지므로 입력 데이터의 트렁케이션 에러를 감소 시키고 고해상도의 FWT Power Spectrum을 발생함수 있다. 이때 리얼 타임의 응용에 적합하도록 월쉬 계수를 256 포인트로 선정 하였다.

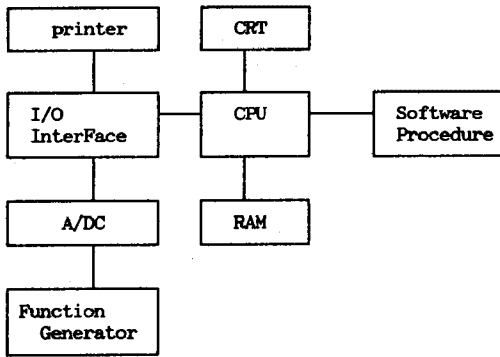


그림 3. 하드웨어 구성 개략도

그리고 ADC가 선형 동작을 한다는 가정하에 Gain = 1 인 프리 앰프를 사용하고 입력 신호의 고주파 성분을 감소시키기 위해서 입력단에 로우 패스 필터를 구성 하였다.

이와 같은 하드웨어 구성을 위하여 Function Generator의 peak to peak voltage와 주파수 범위를 A/DC의 입력전압 및 주파수 범위내에서 결정 하였고 예일리어 심 오차를 감소하기 위하여 sampling frequency를 높게 선정 하였다.

3.2 소프트웨어 구성 (S/W Procedure)

본 실험의 소프트웨어 프로시저는 다음과 같다.

1) PROGRAMMING INITIALIZE

- SAMPLING RATE (FREQUENCY) 설정
- 프리 앰프의 Gain 설정
- 입력 전압 범위 설정

2) A/DC의 출력 데이터를 Reading

3) Data Storage

4) FWT Processing

5) Power Spectrum Processing

6) Data Display

그림4는 소프트웨어에 관한 흐름도를 표시한것이다.

FWT는 그림2와 같이 중간 과정 결과의 점검을 위하여 별도의 저장 작업을 할 필요가 없는 in-place decimation의 알고리즘을 실행 하였으며 이는 입력 데이터가 입력 되기전에 월쉬 변환 과정을 수행함으로써 프로세싱 레이트를 향상시킬 수 있다.

4. 실험 결과 및 고찰

본 실험에서 이용한 월쉬 변환은 Walsh ordering에 의한 고속 변환으로서 컴퓨터 시뮬레이션 결과에서도 알수 있듯이 FFT보다 처리 속도가 훨씬 신속 하였다. 그림5와 그림6은 실제 Function Generator에서 점현파와 구형파의 입력 신호(pv; -5..+5V 입력주파수 18Hz)를 A/DC에 인가 하였을 경우 FWT에 의한 Power Spectrum을 그림7과 8은 FFT에 의한 Power Spectrum을 각각 표시 하였다.(컴퓨터 시뮬레이션에 의한 이들의 Power Spectrum은 부록 참조) 여기서 FWT와 FFT의 선택은 일반적으로 입력 파형에 의존되고 있다. 즉 점현파의 입력 신호에서는 FWT의 Spectrum은 기준값을 중심으로

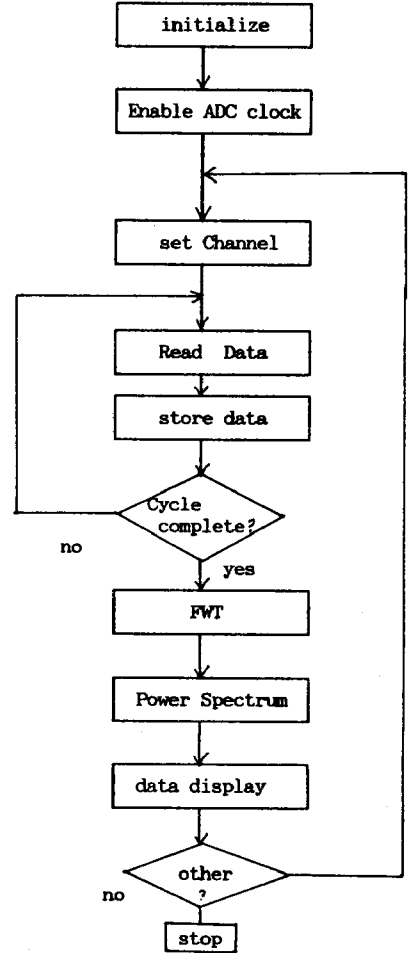


그림 4. S/W 프로그램 구성도

나머지 값들이 산재 되어 있지만 FFT의 경우는 기준값을 중심으로 단순하게 표현되며 구형파의 입력신호에서는 FWT의 Spectrum이 FFT의 Spectrum보다 더욱 단순하게 표현 되고 있다. 이는 변환 알고리즘의 전개방법에 의한 결과로서 월쉬 함수는 불연속의 CAL과 SAL함수에 의해 전개 되었지만 푸리에 함수는 연속의 사인파와 코사인 함수로 전개 되기 때문이다. 그러므로 월쉬함수는 구형파와 같은 불연속 입력 파형에 푸리에 함수는 점현파와 같은 연속 입력 파형에 각각 적당하다. 한편 실제 실험한 Power Spectrum과 컴퓨터 시뮬레이션에 의한 Power Spectrum과는 거의 유사하나 특히 구형파의 입력 신호인 경우 저주파인 18Hz 부근의 기준치는 거의 비슷하지만 저주파 대역에서의 입력 파형이 약간 slope한 현상을 발생하므로 나머지 부분에서의 차이가 있다. 이는 고속의 기계어로 프로그램밍을 하면 높은 주파수의 구형파를 잃으므로 컴퓨터 시뮬레이션에 의한 Power Spectrum과 동일한 형태를 보일 것이다.

5. 결론

마이크로 컴퓨터의 보급으로 FWT는 금후 여러분야에 보급될 것이며 특히 실시간 응용에서는 FFT에 비해 FWT가 처리 속도가 우수하다. 이번 실험을 통하여 S/W Procedure를 Pascal로 프로그래밍 하였기 때문에 Sampling Frequency가 한정되어 있다. 그러므로

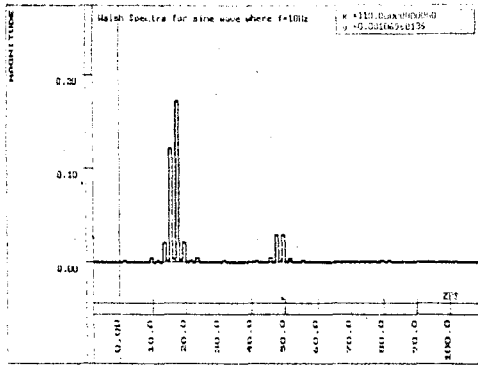


그림 5. FWT를 이용한 power spectrum (sine wave)

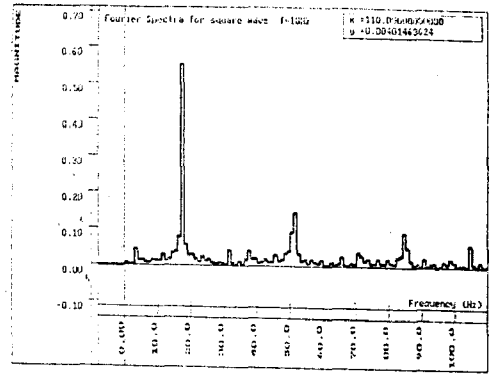


그림 8. FFT를 이용한 power spectrum (square wave)

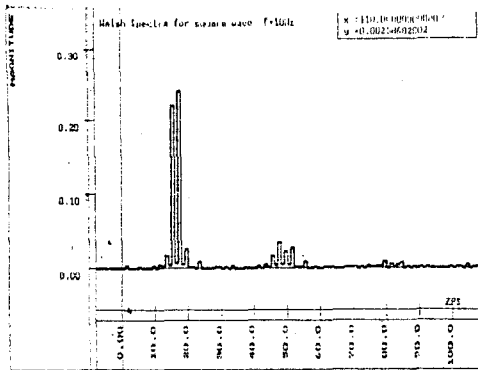


그림 6. FWT를 이용한 power spectrum (square wave)

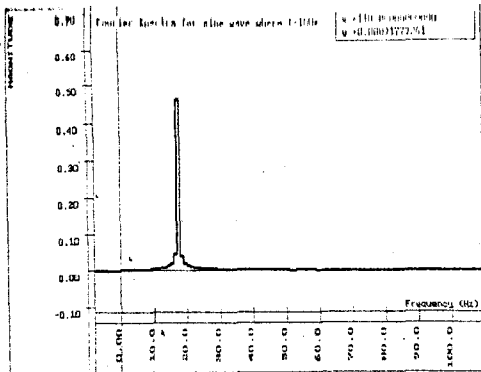


그림 7. FFT를 이용한 power spectrum (sine wave)

전대역에 걸쳐 sampling할수 있도록 추후 High Level 언어 대신 기계어를 활용하고 A/DC에 DMA(Direct Memory Access)기능을 부가하여 처리 속도를 증진 시키고 다중 채널로의 이용이 연구 되어야 할것이다.

참고 문헌

1. P.Stavroulakis and S.Tzafestas "Walsh series approach to observer and filter design in optimal control system" Int.J.Control Vol.26 No.5 pp721-727,1977
2. P.Sharma.S.I.Ahson and J.Henry "MicroProcessor Implimentation of fast walsh hadamard transform for calculation of symmetrical components" Proceedings of the IEEE,Vol 76,No 10,pp1385-1388 1975
3. Henning F.Harmuth,"Transmission of information by orthogonal functions" 2nd Edition,spinger-verlag NEW YORK Heidelberg Berlin,pp 4-28,1972
4. J.W.Manz,"A sequency ordered fast walsh transform",IEEE AU-20,pp 204-205 1972
5. 배종일,"본포 정수계의 파라미터 추정을 위한 월쉬함수 접근" 성균관 대학교 박사학위 청구 논문 pp 17-23,1989
6. Alan V.oppenheim/Ronald W.Shafer "Digital Signal Processing" Prentice-Hall,INC,Englewood cliffs,New Jersey, pp284-287,1975
7. N.Ahmed , K.R.Rao "Orthogonal transforms for digital signal processing" Springer-Verlag,Berlin,Heidelberg, New York ,pp117-119,1975