

# 마이크로프로세서에 의한 WALSH 함수 발생기 구현

안 두수\* 박 준훈<sup>o</sup> 이 명규\*\* 김 중부\*\*\*

\* 성균관대학교 전기공학과 <sup>o</sup> 충주산업대학교 제어계측공학과  
 \*\* 경성대학교 전기공학과 \*\*\* 인덕전문대학 전자과

## A Microprocessor Based Design of Walsh Function Generator

D.S. Ahn\* J.H. Park<sup>o</sup> M.K. Lee\*\* J.B. Kim\*\*\*

\* Dept. of Electrical Eng., Sung Kyun Kwan Univ.  
<sup>o</sup> Dept. of Control & Instrumentation Eng., Chung Ju Nat'l Univ.  
 \*\* Dept. of Electrical Eng., Kyung Sung Univ.  
 \*\*\* Dept. of Electronic Eng., In Duk Institute of Design

### ABSTRACT

Walsh function and transform are important analytical tools for control theory and signal processing and have wide applications in those fields, especially in the field of digital communications. Therefore there is a need for a Walsh function generator in order to realize certain applications. And a number of different designs are known. But design and implementation of such a generator through hardware logic may give rise to orthogonality error.

To develop Walsh function generator which gets rid of orthogonality error, this paper presents a microprocessor based design and implementation method.

### 1. 서론

최근 Walsh 함수를 포함한 직교 함수(orthogonal function)의 응용분야가 확대되고 있는데 그 하나는 시스템의 해석, 간단화, 최적제어 설계등의 제어이론 연구 분야에의 이용이며 다른 하나는 데이터 전송, 신호처리, 영상처리 연구 분야등에서 직교변환을 이용하는 것이다. 특히 실시간 처리에 의한 신호처리나 디지털 통신 시스템의 설계등에서 Walsh 함수 및 Walsh 변환은 폭넓게 이용되고 있으며 이들 시스템의 구현을 위해선 Walsh 함수 발생기의 구현이 요구된다. 이와 관련해서 Besslich는 1973년 배열 발생기 방법에 의해 플립플롭과 주변 디지털 논리회로로 구성된 Walsh 함수 발생기의 구현 방법을 제시한바 있으며 이후 이와 유사한 여러 방법이 연구 제시 되었다. [1,2] 그러나 이러한 하드웨어 회로에 근거한 방법은 대체로 신호 전달 지연 시간, 잡음, 동기신호 동작의 차이등에 의해 직교성 오차(orthogonality error)가 발생하는 단점을 지닌다. [3]

따라서 본 연구에서는 소프트웨어에 근거한 방법으로써 Walsh 함수 발생을 위한 알고리즘을 제시하고 이를 마이크로프로세서와 몇가지 부가적인 주변회로를 이용해서 구현했다. 여기서 제시된 방법은 앞서의 직교성 오차를 최소화하고 완전한 Walsh 함수의 발생을 실현할수 있으며 확장성이 뛰어난 장점을 지닌다.

### 2. Walsh 함수

Walsh는 1923년 불완전 집합의 Rademacher 함수로부터 구형과 형태의 완전한 직교함수 집합을 이루는 Walsh 함수라고 하는 직교함수를 얻어 냈다. [3]

즉, 그림 2-1은 영점교차(zero crossing) 수의 증가에 따른 배열 순서를 지닌 Walsh 함수로 일반적으로 신호처리등에 이용되는 푸우리에 급수(Fourier series)에서와 같이 정현 대칭과 여현 대칭의 특성을 갖는 함수로 나뉘어짐을 볼수 있다.

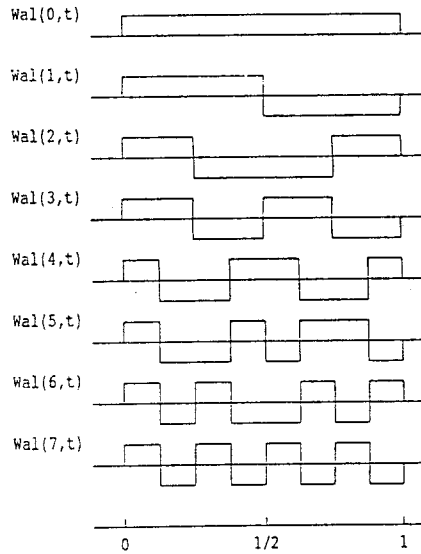


그림 2-1 Walsh 함수

Fig. 2-1 Walsh function(sequence ordered)

한편 그림 2-1에서 시구간을 N=8로 일정 등분하여 샘플링 처리하면 식 2-1과 같이 N=8에 대한 이산(discrete) Walsh 함수에 대한 행렬식을 얻을수 있다.

$$WAL(n, t) = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{pmatrix} \quad (2-1)$$

### 3. Walsh 함수 발생 알고리즘

Walsh 함수 유도에는 다양한 방법이 있으나 본 논문에서는 디지털 로직(logic)처리와 관련하여 부울합성(Boolean synthesis)에 의한 방법을 취하기로 한다. 이 경우 이산 Walsh 함수는 식 3-1과 같이 표현된다.[4]

$$WAL(n, t) = (-1)^{\alpha}, \quad \alpha = \sum_{i=1}^p (n_i \odot t_{i-1}) \quad (3-1)$$

여기서  $n = n_0, 1, 2, \dots, 2^p - 1$ 이며  $n_i$ 는  $n$ 의 2진 표현으로 식 3-2와 같다.

$$n_i = (n_p, n_{p-1}, n_{p-2}, \dots, n_2, n_1)_2 \quad (3-2)$$

또한  $\odot$ 는 모듈로 2진가산(modulo 2 addition)을 의미한다. 한편 Walsh 함수의 대칭관계에 의해 식 3-1은 식 3-4와 같이 표현된다.[3]

$$WAL(n, t) = WAL(t, n) \quad (3-3)$$

$$WAL(t, n) = (-1)^{\beta}, \quad \beta = \sum_{i=1}^p (t_i \odot t_{i-1}) n_{p-i+1} \quad (3-4)$$

따라서 이로부터 식 3-5의 관계가 성립되며 이를 식 3-1에 적용하면 식 3-1은 다시 식 3-6으로 표현된다.

$$\sum_{i=1}^p (n_i + n_{i+1}) t_{p-i+1} = \sum_{i=1}^p (t_i \odot t_{i-1}) n_{p-i+1} \quad (3-5)$$

$$WAL(n, t) = (-1)^{\sum_{i=1}^p (t_i \odot t_{i-1}) n_{p-i+1}} \quad (3-6)$$

여기서 식 3-6을 식 3-7의 그레이 부호(gray code) 함수  $g_b(i, t)$ 를 이용해서 다시 나타내면 식 3-8과 같다.[5]

$$g_b(i, t) = t_i \odot t_{i-1} \quad (3-7)$$

$$WAL(n, t) = (-1)^{\sum_{i=1}^p g_b(i, t) n_{p-i+1}} \quad (3-8)$$

이제 디지털 로직 처리를 위해 2진수 1을 -1에, 0을 +1값에 각각 치환 시키면 식 3-8로부터 식 3-9를 얻을 수 있고

$$WAL(n, t) = \prod_{i=1}^p g_b(i, t) n_{p-i+1} \quad (3-9)$$

같은 방법으로  $WAL(n, t-1)$ 은 식 3-10과 같이 된다.

$$WAL(n, t-1) = \prod_{i=1}^p g_b(i, t-1) n_{p-i+1} \quad (3-10)$$

여기서 식 3-9와 3-10에 대해 모듈로 2진가산을 취하고 이를  $NEG(n, t)$ 로 두면 식 3-11로 된다.[6] 이는 Walsh 함수 발생과 관련하여 함수 값의 논리적 반전(logical negation)을 지시하는 기능을 지닌 반전함수(negation function)로 식 3-12의 관계가 성립된다.

$$NEG(n, t) = WAL(n, t) \odot WAL(n, t-1) \quad (3-11)$$

$t=1, 2, 3, \dots, 2^p-1$

$$NEG(n, t)=1 \text{ 이면 } WAL(n, t) = \overline{WAL(n, t-1)}$$

$$NEG(n, t)=0 \text{ 이면 } WAL(n, t) = WAL(n, t-1) \quad (3-12)$$

여기서  $\overline{WAL(n, t-1)}$ 는 종전의 원래 값에 대한 논리적 반전을 의미한다. 따라서 반전함수  $NEG(n, t)$ 가 결정되면 연속적으로 Walsh 함수 발생을 위한 행렬식을 결정할 수가 있다.

그러므로 이제  $NEG(n, t)$ 를 결정하기 위해 식 3-11을 식 3-9, 3-10을 이용해서 표현하면 식 3-13과 같다

$$NEG(n, t) = \prod_{i=1}^p g_b(i, t) n_{p-i+1} \odot \prod_{i=1}^p g_b(i, t-1) n_{p-i+1} \quad (3-13)$$

다시 식 3-13을 간략히 표현하면 식 3-14와 같이 된다.

$$NEG(n, t) = \prod_{i=1}^p d(i, t) n_{p-i+1}, \quad t=1, 2, 3, \dots, 2^p-1 \quad (3-14)$$

여기서  $d(i, t)$ 는 dyadic 부호로 식 3-15와 같이  $g_b(i, t)$ 에 대한 모듈로 2진가산에 의해 얻어진다.

$$d(i, t) = g_b(i, t) \odot g_b(i, t-1) \quad (3-15)$$

따라서 표 3-1에 보인바와 같이 2진 부호로부터 그레이 부호  $g_b(i, t)$ 를 얻고 다시 이로부터  $d(i, t)$ 를 얻어 이를 이용해서 반전함수  $NEG(n, t)$ 를 결정할 수 있으며 결과적으로 원하는 Walsh 함수 발생이 가능하다.

표 3-1  $p=3$ 일때  $t, g_b(i, t), d(i, t)$ 의 관계  
Table 3-1 The relation between  $t, g_b(i, t)$  and  $d(i, t)$  for  $p=3$

n	t			$g_b(i, t)$			$d(i, t)$		
	$t_1$	$t_2$	$t_3$	$i=3$	$i=2$	$i=1$	$i=3$	$i=2$	$i=1$
0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0	0	1
2	0	1	0	0	1	1	0	1	0
3	0	1	1	0	1	0	0	0	1
4	1	0	0	1	1	0	1	0	0
5	1	0	1	1	1	1	0	0	1
6	1	1	0	1	0	1	0	1	0
7	1	1	1	1	0	0	0	0	1

### 4. 하드웨어 구현

일반적인 Walsh 함수 발생기의 구현은 그림 4-1에 보인바와 같이 배열 발생기(array generator) 방법이 이용된다. 이는 플립플롭(flip-flop) 및 논리 게이트를 배열 방법으로 적절히 구성, Rademacher 함수의 연속적 곱에 의해 Walsh 함수를 유도해서 발생시키는 방법으로써 이 경우 신호전달 지연, 잡음, 동기신호 동작의 차이등으로 인해 직교성 오차를 갖게되 완전히하고 순수한 Walsh 함수를 발생시키기 어렵다.

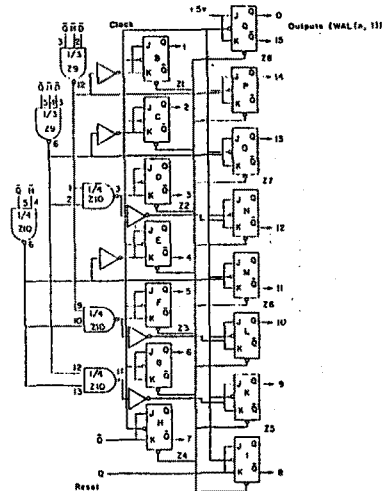


그림 4-1 Besslich 배열 발생기  
Fig. 4-1 Besslich array generator

따라서 직교성 오차를 제거한 Walsh 함수 발생기를 구현하기 위해 앞서 제시된 Walsh 함수 발생 알고리즘을 마이크로프로세서를 이용한 소프트웨어 기법을 적용해서 실현하고 이를 하드웨어적으로 구현한다. 즉, 하드웨어 구현을 위해 마이크로프로세서는 Zilog사의 Z80A를 사용했고 여기에 시스템 구성을 위해 EPROM(2764), SRAM(6116), i8255(입출력 포트) 및 기타 부가회로를 연결했다. 이에 대한 간략한 회로 구성도(block diagram)를 그림 4-2에 보였다.

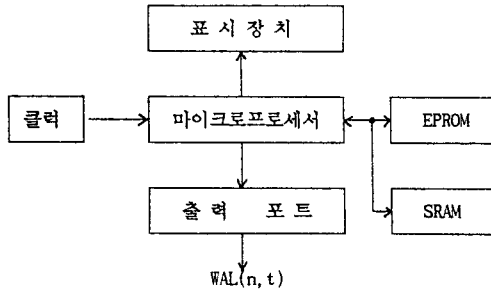


그림 4-2 마이크로프로세서를 이용한 Walsh 함수 발생기 구성도

Fig. 4-2 Block diagram for Walsh function generator using microprocessor

다음으로 본 논문에서 제시된 알고리즘의 프로그래밍 처리를 위해 Z80 어셈블리 언어를 이용했으며 이의 흐름도는 그림 4-3과 같다. 흐름도에서 보인 바와 같이 알고리즘은 각 단계의 순차적인 연속 동작에 의해 처리되며 각 단계는 기본적으로 부호 발생 처리를 위해 하향 계수기(down counter)에 의한 순환 반복구조를 갖는다. 따라서 하향 계수기 값은 부호 발생처리 회수에 관계되며 이는 곧 발생 시 키고자 하는 Walsh 함수의 차수(order number) 결정과 직결된다.

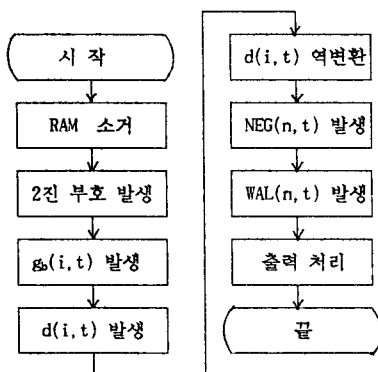


그림 4-3 마이크로프로세서를 이용한 Walsh 함수 발생기 흐름도

Fig. 4-3 Flowchart for Walsh function generator using microprocessor

표 4-1에는 하향계수 값을 8(B register=08H)로 초기화해서 처리된 Walsh 함수 발생의 결과를 보였다.

표 4-1 반전함수 NEG(n, t)에 의한 N=8인 Walsh 함수발생 Table 4-1 Derivation of the Walsh functions from NEG(n, t) for N=8

n	WAL(n, t)	WAL*(n, t)	NEG(n, t)
0	1 1 1 1 1 1 1 1	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
1	1 1 1 1 -1 -1 -1 -1	0 0 0 0 1 1 1 1	0 0 0 0 1 0 0 0
2	1 1 -1 -1 -1 -1 1 1	0 0 1 1 1 1 0 0	0 0 1 0 0 0 1 0
3	1 1 -1 -1 1 1 -1 -1	0 0 1 1 0 0 1 1	0 0 1 0 1 0 1 0
4	1 -1 -1 1 1 -1 -1 1	0 1 1 0 0 1 1 0	0 1 0 1 0 1 0 1
5	1 -1 -1 -1 1 1 1 -1	0 1 1 0 1 0 0 1	0 1 0 1 1 1 0 1
6	-1 -1 1 -1 -1 -1 1 1	0 1 0 1 1 0 1 0	0 1 1 0 1 1 1 1
7	-1 -1 1 -1 1 -1 -1 -1	0 1 0 1 0 1 0 1	0 1 1 1 1 1 1 1

주) 위 표에서 WAL\*(n, t)는 WAL(n, t)가 0, 1로 치환된 것을 의미함

## 5. 검토 및 결론

본 논문에서는 소프트웨어에 근거해서 마이크로프로세서를 이용한 Walsh 함수 발생기 구현 방법에 대해 다루었다. 이를 위해 본 논문은 일반적인 Walsh 함수 유도 방법인 Rademacher 함수의 연속적 곱으로 부터의 유도 방법과는 달리 부울합성(Boolean synthesis)으로 부터의 유도에 기초한 방법을 이용했다. 이에 따라 2진 부호로 부터 그레이 부호  $g(i, t)$  및 dyadic 부호  $d(i, t)$ 를 구하고 그로 부터 반전함수인 NEG(n, t)를 얻어 이를 이용해서 Walsh 함수를 발생토록 했다. 또한 이를 실제 하드웨어로 구현하기 위해 기존의 배열 발생기 방법이 아닌 마이크로프로세서를 중심으로 한 시스템을 구성, 소프트웨어적으로 Walsh 함수 발생 처리 알고리즘의 구현이 가능토록 했다. 본 논문에서 제시된 마이크로프로세서에 의한 Walsh 함수 발생기의 구현 방법은 기존의 방법이 지닌 직교성 오차를 제거 함으로써 완전하고 순수한 Walsh 함수를 발생시킬수가 있었으며 동시에 발생기 구성 및 동작에 있어서 소프트웨어적으로 Walsh 함수 발생 차수의 조정이 가능한 점등 그 유용성과 확장성이 뛰어난 장점을 지닌다.

아울러 향후 다른 직교함수에 본 논문에서 제시된 기법을 적용시켜 범용적인 직교함수 발생기 구현에 대한 지속적인 연구가 기대된다.

## 참고 문헌

- [1] P.W.Besslich, "Walsh function generators for minimum orthogonality error", IEEE Trans. Electromag. Compat. EMC-19, 1973, pp. 407-410
- [2] R.Kitai, "Walsh function array generator:A comparison", IEEE Trans. Electromag. Compat. EMC-21, 1979, pp. 153-154
- [3] K.G.Beauchamp, "Application of Walsh and Related Functions", Academic Press, 1984
- [4] N.Ahmed and K.R.Rao, "Orthogonal Transforms for Digital Signal Processing", Springer-Verlag, 1975
- [5] C.K.Yuen, "Walsh function generator using Gray code", in Proc. Symp. Application of Walsh Function, 1973, pp. 284-289
- [6] Li Yi-Hua, "Design of a Walsh function generator", IEEE Trans. Electromag. Compat., EMC-29, 1987, pp. 83-86