

논문 2023-60-11-8

면적 효율적인 OFDM/CDMA 통신용 부분 병렬 FWHT 프로세서

(Area-efficient Partially-parallel FWHT Processor for OFDM/CDMA Communication)

황 지 우*, 황 용 택*, 김 민 수*, 유 호 영**

(Jiwoo Hwnag, Youngtaek Hwang, Minsu Kim, and Hoyoung Yoo[Ⓞ])

요 약

본 논문에서는 낮은 하드웨어 복잡도와 기존 방식 대비 직관적인 설계 도출이 가능하며, 면적 효율적인 Orthogonal Frequency Division Multiplexing (OFDM)/Code Division Multiple Access (CDMA) 통신용 Fast Walsh Hadamard Transform (FWHT) 프로세서 설계 기법을 제안한다. 기존 설계 방법론은 제한된 하드웨어 자원 속에서 하드웨어 복잡도와 연산 처리량 간의 trade-off 문제를 적절하게 해결하지 못하였다. 본 논문에서는 해당 문제 해결을 위한 부분 병렬 FWHT 프로세서 설계 기법을 제안한다. 제안하는 부분 병렬 FWHT 프로세서는 folding transformation 기법을 활용하여 데이터 재정렬 과정을 보다 효율적으로 진행하며, 기존 설계 기법과 비교하여 더욱 직관적인 구조 도출을 가능하게 한다. 본 논문에서는 또한 CMOS 180nm 공정에서 합성된 결과를 제시하고, 이를 통해 제안하는 부분 병렬 FWHT 프로세서 구조가 완전 병렬 구조 대비 88% 적은 면적을 차지하며, 직렬 구조와 비교하여 405% 높은 처리량을 보여주는 것을 확인하였다. 추가적으로, 제안하는 설계 기법이 다양한 하드웨어 환경에서도 유연하며 효율적인 성능을 발휘함을 확인하였다. 이를 통해 OFDM 및 CDMA 기반의 무선 통신 시스템의 성능 향상에 기여할 수 있음을 보여준다.

Abstract

In this paper, we propose a Fast Walsh Hadamard Transform (FWHT) processor design method for Orthogonal Frequency Division Multiplexing (OFDM)/Code Division Multiple Access (CDMA) communication. This method enables intuitive design deduction compared to conventional methods and offers both area efficiency and low hardware complexity. Existing design methodologies have not been able to solve the trade-off problem between hardware complexity and computational throughput in limited hardware resources. We propose a partially parallel FWHT processor design technique to solve this problem. The proposed partially parallel FWHT processor utilizes the folding transformation technique to perform the data rearrangement process more efficiently and enables a more intuitive structure derivation compared to existing design techniques. This paper also presents the results synthesized in the CMOS 180nm process, and confirms that the proposed partially parallel FWHT processor structure occupies 88% less area than the fully parallel structure and shows 405% higher throughput than the serial structure. Additionally, it was confirmed that the proposed design technique shows flexible and efficient performance in various hardware environments. This shows that it can contribute to improving the performance of OFDM and CDMA based wireless communication systems.

Keywords : Fast walsh hadamard transform, Folding transformation, OFDM processor, CDMA processor

*학생회원, **정회원, 충남대학교 전자공학과(Dept. of Electronics Engineering, Chungnam National University)

Ⓞ Corresponding Author(E-mail : hyyoo@cnu.ac.kr)

※ This work was supported by the Brain Korea 21 Plus, supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2022R1A5A8026986), supported by the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (2022-0-01170, PIM 반도체 설계연구센터), and the EDA tool was supported by the IC Design Education Center(IDEC), Korea

Received ; August 7, 2023

Revised ; September 6, 2023

Accepted ; September 25, 2023

I. 서 론

무선 통신 시스템에서 데이터 전송률이 더 높아짐과 함께 한정적인 스펙트럼 문제로 인해 Orthogonal Frequency Division Multiplexing (OFDM)과 Code Division Multiple Access (CDMA)와 같은 다중접속 방식이 모바일 통신 시스템에서 활용되고 있다. 오늘날 OFDM 과 CDMA를 포함한 통신 시스템 외에도 다양한 의료 산업 및 방위 산업 분야에서 디지털 신호 처리를 광범위하게 활용하고 있다. 일반적으로 이러한 목적을 위해 널리 사용되는 범용 알고리즘으로는 Fast Fourier Transform (FFT)이 있다. 하지만, 하드웨어 상에서 FFT 연산을 수행하는 것은 상대적으로 많은 연산량과 높은 하드웨어 복잡도를 요구한다. 따라서, 하드웨어 자원이 제한된 환경에서의 디지털 하드웨어 기반 신호 처리에는 제약이 있다. Fast Walsh Hadamard Transform (FWHT)은 곱셈 연산을 포함하지 않으며 덧셈과 뺄셈만으로 연산을 수행하여 이러한 복잡도 문제를 해결할 수 있다. FWHT는 FFT의 일반화된 모델로서 2^n 개의 실수에 대하여 직교, 대칭 그리고 선형적인 연산을 수행하며 FFT에 사용되는 분할 정복 알고리즘인 Cooley-Tukey 알고리즘을 기반으로 한 butterfly unit을 활용한다. 따라서, FFT와 유사한 데이터 흐름을 가지고 있다. 일반적인 FWHT 설계 기법은 과거의 FFT 설계 기법에서 동일하게 가졌던 하드웨어 복잡도와 처리 속도 사이의 trade-off 문제를 가지고 있다. 하지만, FFT의 하드웨어 구조는 50년 이상 연구되었기 때문에 연구된 안정적인 FFT 구조를 FWHT에 적용할 수 있다^[1-3]. 본 논문에서는 FWHT 연산 처리 단계별 데이터 재정렬 순열을 분석하여 OFDM/CDMA 통신 시스템 및 다양한 산업 분야에서 활용할 수 있는 면적 효율적인 부분 병렬 FWHT 프로세서 구조를 제안한다.

II. 연구 배경

CDMA 시스템에서 동시에 전송되는 채널들을 구분하기 위해 서로 직교성을 가지는 직교 확산 코드를 이용한다. 기지국(Base station)에서는 전송할 데이터 수열과 직교 확산 코드의 곱셈 연산을 통해 생성한 확산 신호를 전송한다. 마찬가지로 이동국(Mobile station)에서는 수신한 확산 신호에 동일한 직교 확산 코드와의 곱셈 과정을 거쳐 확산 신호에 숨어있는 데이터 수열을

복구한다^[4, 5]. 보편적인 CDMA 시스템에서 직교 확산 코드로 Hadamard 행렬을 기반으로 생성한 Walsh 코드를 사용한다^[6, 7]. 추가적으로, OFDM 시스템에서도 Walsh 코드의 높은 직교성을 활용한다. Walsh 코드는 Walsh Hadamard Transform (WHT)을 통해 디코딩할 수 있으며 WHT의 고속 알고리즘인 FWHT을 사용하면 더 낮은 디코딩 복잡도를 가질 수 있다^[8].

1. Hadamard matrix

Hadamard 행렬은 1과 -1만을 가지며 구성하는 모든 행들이 서로 직교하고 모든 열 또한 같은 특징을 가지는 정방 행렬이다. 따라서, 식 (1)과 같이 직교, 대칭 및 선형성을 가진다.

$$H_N = H_N^T = H_N^{-1} = H_N^*, \quad (1)$$

H_N 은 $N = 2^n$ 을 만족하는 $N \times N$ Hadamard 행렬, H_N^T 는 전치행렬, H_N^{-1} 은 역 행렬, H_N^* 는 conjugate 연산이다. 2×2 Hadamard 행렬은 핵심 행렬(Core Matrix)로 식 (2)와 같다.

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \quad (2)$$

Hadamard 행렬의 확장은 핵심 행렬 H_2 에 kronecker product를 연속적으로 적용하여 수행할 수 있다. 즉, Hadamard 행렬의 재귀적 관계식은 식 (3)과 같으며 \otimes 는 kronecker product를 의미한다. 식 (3)에 의해 확장된 4×4 Hadamard 행렬식은 식 (4)와 같다.

$$H_{2N} = H_2 \otimes H_N = \begin{bmatrix} H_N & H_N \\ H_N & -H_N \end{bmatrix}, \quad (3)$$

$$H_4 = H_2 \otimes H_2 = \begin{bmatrix} H_2 & H_2 \\ H_2 & -H_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}. \quad (4)$$

2. Fast Walsh Hadamard Transform

N 차 입력 벡터 x_N 에 대한 Hadamard 변환식은 식 (5)와 같다.

$$X_N = \frac{1}{N} H_N x_N. \quad (5)$$

Hadamard 행렬의 각 요소는 1 또는 -1로만 구성되었기 때문에 식 (5)에 주어진 곱셈 연산은 WHT의 고속 알고리즘인 FWHT에서 디지털 순차 구조로 구현된 일련의 덧셈, 뺄셈 연산으로 수행할 수 있다. FWHT 알

고리즘은 FFT의 분할 정복 알고리즘인 Cooley-Tukey 알고리즘과 유사한 알고리즘이며 WHT이 가지는 $O(N^2)$ 의 연산 복잡도를 $O(N \log_2 N)$ 으로 줄일 수 있다. FWHT의 구조는 FFT의 복소수 곱셈이 포함된 butterfly unit을 덧셈기와 가산기로 구성된 butterfly unit으로 대체하여 구현할 수 있다^[9~11]. 따라서, FWHT는 사각과 형태의 변환으로 FFT와 같은 정현과 형태의 변환보다 디지털 신호 처리에서 상황에 따라 더 효율적인 연산이 가능하다. FWHT 연산은 FFT와 동일하게 Decimation In Frequency (DIF)와 Decimation In Time (DIT) 구조로 나눌 수 있다. DIF 구조는 입력 비트열이 순차일 때 출력 비트열이 비트 반전 변환되는 구조이며 DIT 구조는 DIF 구조와 대칭이 되는 구조이다. 그림 1은 입력 개수 $N = 16$ 에 대한 FWHT 연산 구조이다. 각 stage 별로 8개의 butterfly unit들이 분할 정복 알고리즘을 기반으로 더 이상 분할할 수 없을 때까지 주어진 입력 쌍의 덧셈과 뺄셈 연산을 각각 수행한다. 이때, 입력과 출력 비트열이 서로 비트 반전된 형태임을 확인할 수 있다. 따라서, FWHT 연산은 $N/2$ 개의 덧셈과 뺄셈 연산으로 구성된 각 stage 연산을 총 $\log_2 N$ 번 반복하여 수행할 수 있으며 총 $(N/2 \times \log_2 N)$ 개의 butterfly unit들이 필요하다.

III. 일반적인 FWHT 프로세서

일반적인 FWHT 프로세서는 여러 입력에 대하여 동시에 연산을 수행하는 병렬 구조와 순차적으로 들어오는 입력에 대하여 하나씩 연산을 수행하는 직렬 구조로 나뉜다. 본 논문에서는 병렬 구조로는 완전 병렬 구조와 직렬 구조로는 Single-path Delay Feedback (SDF) 구조를 예로 한다.

1. Fully-Parallel FWHT 구조

완전 병렬 구조는 그림 2와 같이 전체 입력 데이터를 대상으로 FWHT의 전체 stage 연산을 동시에 수행하는 구조이다. 각 stage는 입력 데이터 개수 N 에서 $\log_2 N$ 개의 butterfly unit들로 구성되며 각각의 stage 사이에는 데이터 재정렬을 위한 reordering network가 존재한다^[12~14]. Reordering network는 추가적인 구성 요소가 필요 없으며, 다음 stage의 butterfly unit들의 입력 쌍에 직접 연결되어 있다. 따라서, 매 클럭 주기당 N -point의 높은 연산 처리량을 가지며 직관적인 데이

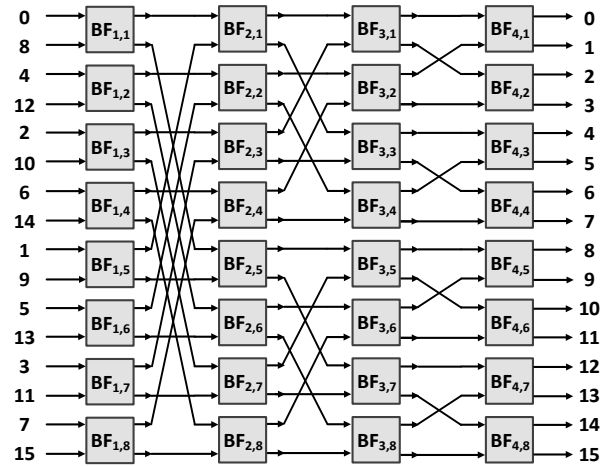


그림 1. 고속 월시 하다마드 변환 ($N = 16$)
Fig. 1. Fast walsh hadamard transform ($N = 16$).

터 흐름 분석이 가능하다. 하지만, 데이터 개수가 증가함에 따라 연산에 사용되는 butterfly unit들의 수도 선형적으로 증가하여 높은 하드웨어 복잡도를 요구한다. OFDM 및 CDMA와 같은 통신 시스템의 이동국에서는 기지국과 비교해 상대적으로 높은 휴대성, 낮은 소비전력과 계산량의 최소화를 요구한다. 따라서, 완전 병렬 구조의 FWHT 프로세서를 사용하기에는 제약이 있다.

2. Single-path Delay Feedback FWHT 구조

SDF 구조는 butterfly unit 연산을 위해 입력 쌍 사이 정해진 거리만큼 지연 요소를 두어 기다린 후 해당 데이터가 입력될 때 butterfly unit 연산을 수행하는 구조이다. 이때, 지연 요소는 feedback loop를 활용한 단일 경로를 사용하기 때문에 직렬화된 데이터 흐름이 형성된다. $\log_2 N$ 개의 stage들로 구성된 SDF 구조는 각 stage 별로 butterfly unit, feedback을 위한 First In First Out (FIFO) 레지스터, 데이터 치환을 위한 multiplexer로 구성된 기본 회로를 활용한다. FIFO 레지스터는 각 stage i 별로 $(N/2^i)$ 길이를 가지며 feedback loop를 통해 데이터를 지연시킨다. $N = 512$ 의 SDF 구조는 그림 3과 같이 각각 256, 128, 64, 32, 16, 8, 4, 2, 1 길이의 FIFO 레지스터를 포함하는 기본 회로 9개로 구성된다. SDF 구조는 완전 병렬 구조에 비해 상대적으로 적은 하드웨어 자원만으로도 구현이 가능하지만 feedback loop를 활용한 데이터 지연과 낮은 병렬 차원 때문에 적은 연산 처리량을 가진다. 결과적으로, 실시간에 가까운 빠른 연산을 요구하는 산업 분야, 특히 OFDM/CDMA 통신 시스템에서 이동국과

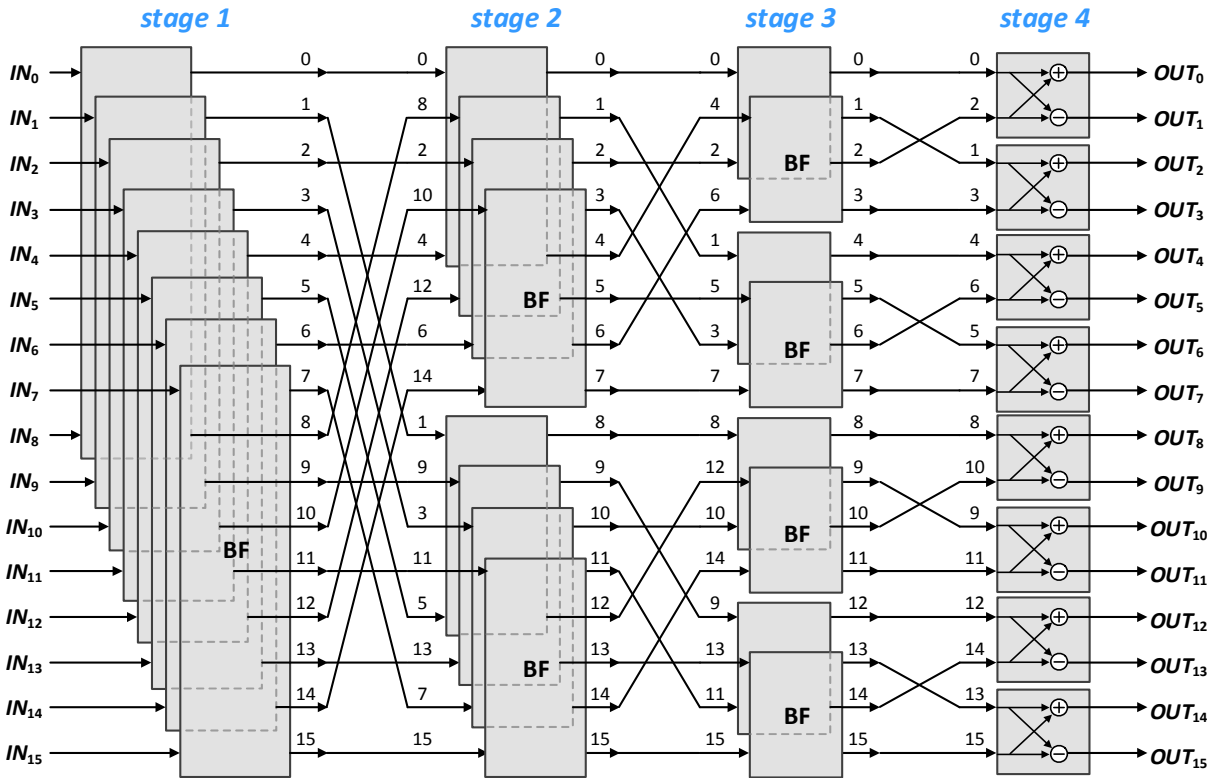


그림 2. 완전 병렬 구조의 FWHT 프로세서 ($N = 16, P = 16$)
 Fig. 2. Fully-Parallel structured FWHT processor ($N = 16, P = 16$).

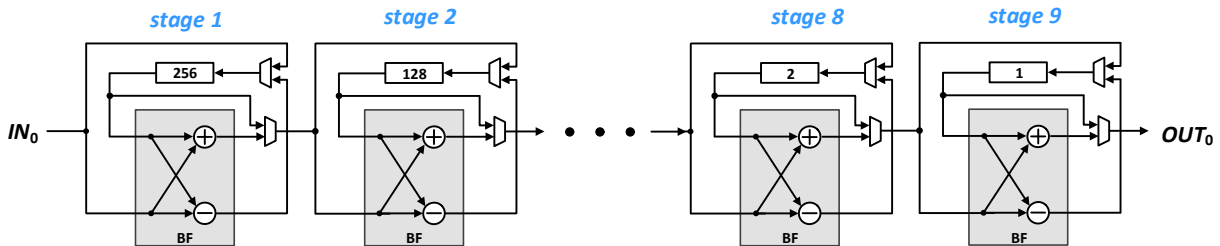


그림 3. 단일 경로 지연 피드백 구조의 FWHT 프로세서 ($N = 512, P = 1$)
 Fig. 3. Single-path Delay Feedback structured FWHT processor ($N = 512, P = 1$).

같은 경우의 활용에는 제한이 있다.

IV. 제안하는 FWHT 프로세서

완전 병렬 구조는 SDF 구조와 비교하여 높은 연산 처리량을 달성하지만, 그에 따른 하드웨어 복잡도 증가는 불가피하다. 반대로 SDF 구조는 완전 병렬 구조와 비교하여 적은 하드웨어 복잡도로 구현이 가능하지만 낮은 연산 처리량을 달성한다. 따라서, 본 논문에서는 folding transformation^[15, 16] 기법을 활용하여 데이터 스트림 순서를 재배열하는 과정을 통해 연산 처리 속도와 하드웨어 복잡도 사이 trade-off 문제를 해결하는 부분 병렬 FWHT 프로세서 구조를 제안한다. 제안하는

설계 기법과 구조를 명확하게 설명하기 위해 FWHT 연산의 입력 데이터 수 $N = 512$, 병렬 인자 $P = 8$ 의 부분 병렬 FWHT 구조를 예시로 한다.

1. 제안하는 부분 병렬 FWHT 알고리즘

제안하는 부분 병렬 FWHT 프로세서의 알고리즘은 folding Transformation을 기반으로 한다. Folding transformation은 하드웨어 자원 절약에 중점을 둔 방법 중 하나로 시간 다중화(Time-Multiplexing)를 기반으로 한다. Folding transformation 기법에서 folding(ordering) set을 결정하는 것은 중요한 설계 요소 중 하나이다. 제안하는 부분 병렬 구조의 FWHT에서 folding set은 각 stage 별 butterfly unit들의 입력

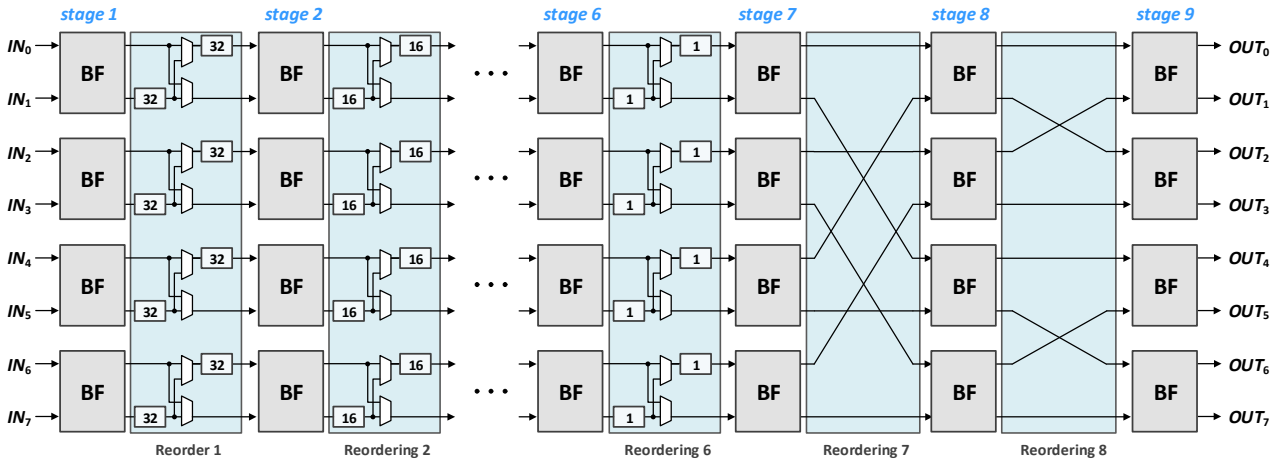


그림 4. 부분 병렬 구조의 FWHT 프로세서 ($N = 512, P = 8$)
 Fig. 4. Partially-Parallel structured FWHT processor ($N = 512, P = 8$).

쌍을 의미한다. 본 논문에서 제안하는 FWHT 구조의 경우 입력 순서가 순차적이기 때문에 다음 stage 입력은 연속된 순서로 입력되어야 한다. 따라서, butterfly unit의 출력을 재배열하는 과정이 필요하다. butterfly unit들의 출력 쌍은 입력 개수 $N = 2^n$ 일 때, stage s 에서 $(n - s)$ 비트만 반전되어 나타나는 특성이 존재한다. 반전되어 나타나는 순서를 연속된 순서로 재배열하기 위해 반전되는 비트의 크기와 병렬 인자의 크기를 고려한 지연 소자가 필요하다. 효율적인 butterfly unit들의 입력 쌍, 즉 folding set을 구성하기 위해 그림 5와 같은 기본 회로를 활용하여 각 stage 사이에 데이터 재배열 회로를 구현하였다. 제안하는 구조의 기본 회로는 병렬 차원 사이 데이터 교환에 필요한 지연 시간을 반영한 길이의 FIFO 레지스터 한 쌍과 데이터 치환을 위한 multiplexer 한 쌍을 포함한다.

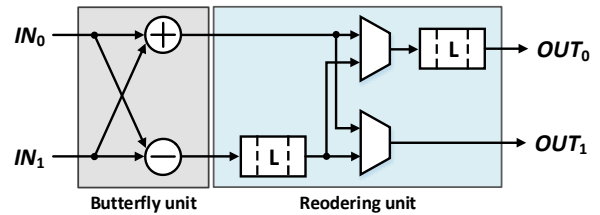


그림 5. 부분 병렬 FWHT 구조의 기본 회로
 Fig. 5. Basic circuit of Partially-Parallel FWHT structure.

여 대칭적으로 구현할 수 있다. 해당 설계 방법은 포괄적인 구조로서 모든 병렬 차원에 모두 적용할 수 있다. 따라서, 다양한 설계 환경에 따라 병렬 인자를 선택하여 부분 병렬 구조의 FWHT 프로세서를 설계한다면 가장 효율적인 FWHT 프로세서를 설계할 수 있다.

V. 실험 결과

2. 제안하는 부분 병렬 FWHT 구조

그림 4는 $N = 512, P = 8$ 인 부분 병렬 FWHT 프로세서 구조이다. 기본적으로 병렬 인자 P 에 해당하는 부분 병렬 FWHT 프로세서는 한 클럭당 P 개의 데이터가 병렬로 처리되기 때문에 각 stage 별로 $(P/2)$ 개의 butterfly unit들과 데이터 재배열을 위한 전체 $(N - P)$ 길이의 FIFO 레지스터들로 구성된다. 해당 구조는 각 stage 사이에 존재하는 데이터 재배열 회로에 지연 요소 낭비를 최소화하기 위해 feed back 구조가 아닌 feed forward 구조를 활용하였다^[17, 18]. 또한, 데이터 지연이 더 이상 필요하지 않은 부분은 직접 연결하였다. 제안하는 부분 병렬 구조는 입력 데이터의 개수 N 과 병렬 인자 P 에 따라 해당 설계 방법을 적용하

표 1은 제안하는 부분 병렬 FWHT 구조와 일반적으로 사용되는 FWHT 구조 사이의 하드웨어 리소스, 처리 속도, 처리량, 하드웨어 효율성 측면에서 정량적인 비교를 보여준다. 표 1을 통해 완전 병렬 구조, SDF 구조, 제안하는 부분 병렬 구조의 장점을 명확하게 확인할 수 있다. 제안하는 부분 병렬 구조는 완전 병렬 구조, SDF 구조와 비교하여 현실적인 처리량을 제공하면서 유연한 하드웨어 복잡도를 제공할 수 있다.

제안하는 부분 병렬 구조의 FWHT 프로세서는 하드웨어 설계를 위한 Register Transfer Level (RTL)은 Verilog HDL로 기술하고 CMOS 180nm 공정을 활용하였다. 각 프로세서는 최대 동작 주파수를 고려해 각 동작 주파수 80MHz, 400MHz, 200MHz로 합성을

표 1. 하드웨어 복잡도 비교
Table 1. Hardware complexity comparison.

	Fully-Parallel	SDF	Partially-Parallel
Butterflies	$(N/2)\log_2 N$	$\log_2 N$	$(P/2)\log_2 N$
Registers	0	$N-1$	$((N/P)-1)P$
Delays	-	$N-1$	$N-P$
Latency	1	$N-1$	$(N/P)-1$
Throughput	N	1	P
Utilization	100%	50%	100%

표 2. 합성 결과 ($N = 512$)
Table 2. Synthesis result ($N = 512$).

	Fully-Parallel	SDF	Partially-Parallel
Operating Frequency [MHz]	80	400	200
Critical path delay [ns]	11.41	1.76	3.91
Gate count [#NAND]	660K	74K	78.4K
Latency [ns]	12.5	1277.5	315
Throughput [Gbps]	40.9	0.4	1.62

진행하였다. 표 2의 합성 결과로부터 제안하는 구조가 입력의 부분 병렬화와 데이터 재배열 회로를 효율적으로 구현함으로써 완전 병렬 구조 대비 88% 적은 면적을 달성하였으며 SDF 구조 대비 405% 높은 처리량을 달성하였다. 비록 표 2는 제안하는 부분 병렬 구조를 병렬 인자 $P = 8$ 에 대해서만 합성 결과를 제시하고 있지만 표 1을 통해 다른 N 과 P 에 대해서도 하드웨어 복잡도와 처리 속도 측면에서 효율적인 FWHT 프로세서를 설계할 수 있음을 유추할 수 있다. 따라서, 제안하는 부분 병렬 FWHT 프로세서는 다양한 설계 요건을 요구하는 OFDM/CDMA 통신의 이동국에서 유용하게 활용할 수 있다.

VI. 결 론

본 논문에서는 OFDM/CDMA 통신에서 활용 가능한 부분 병렬 FWHT 구조를 제안하였다. 제안하는 구조를 도출하기 위해 각 stage 별 데이터 재배열 과정이 갖는 규칙성을 파악하고 일반화된 데이터 재배열 회로를 제

시하였다. 이를 통해 일반적으로 활용되는 완전 병렬 구조의 높은 하드웨어 복잡도 문제와 SDF 구조의 낮은 연산 처리량 문제를 해결하였다. 제안하는 구조 검증을 위해 진행한 CMOS 180nm 공정 합성 결과 기존 구조 대비 높은 하드웨어 복잡도 개선을 이루었다. 결과적으로 제안하는 구조는 직관적인 설계 기법을 통해 FWHT 연산의 입력 개수 N 및 병렬 인자 P 의 변화에도 유연하게 최적화가 가능한 FWHT 프로세서 설계가 가능하다. 따라서, 다양한 설계 환경에서 순차적인 FWHT 출력을 필요로 하는 OFDM 통신 또는 Walsh 코드 기반의 CDMA 통신용 System on Chip (SoC) 및 Application Specific Integrated Circuit (ASIC)에 본 논문에서 제안하는 FWHT 프로세서를 활용할 수 있을 것이다.

REFERENCES

- [1] S. Choi, Y. Shin, K. Lim and H. Yoo, "Efficient Partially-parallel NTT Processor for Lattice-based Post-quantum Cryptography," *JOURNAL OF SEMICONDUCTOR TECHNOLOGY AND SCIENCE*, vol. 22, no. 6, pp. 459-474, Dec. 2022.
- [2] M. Ayinala, M. J. Brown, and K. K. Parhi, "Pipelined parallel FFT architectures via folding Transformation," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 20, no. 6, pp. 1068-1081, Jun. 2012.
- [3] Liu, Jinqi, et al. "Pipelined architecture for a radix-2 fast Walsh-Hadamard-Fourier transform algorithm." *IEEE Transactions on Circuits and Systems II: Express Briefs* 62.11 (2015), pp. 1083-1087.
- [4] M. Noshad and M. Brandt-Pearce, "Hadamard coded modulation: An alternative to OFDM for wireless optical communications," *2014 IEEE Global Communications Conference, Austin, TX, USA, 2014*, pp. 2102-2107.
- [5] A. Amira, et al. "An FPGA implementation of Walsh-Hadamard transforms for signal processing." *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings*. Vol. 2, 2001.
- [6] A. Garcia, Manjarres, et al. "Parallel-Pipeline Fast Walsh-Hadamard Transform Implementation Using HLS." *2021 International Conference on Field-Programmable*

- Technology (ICFPT)*, 2021.
- [7] S. K. Bahl, "Design and prototyping a Fast Hadamard Transformer for WCDMA," *14th IEEE International Workshop on Rapid Systems Prototyping*, 2003. Proceedings., San Diego, CA, USA, 2003, pp. 134-140.
- [8] Amira, A., and A. Bouridane. "An FPGA based accelerator for discrete Hartley and fast Hadamard transforms." *2003 46th Midwest Symposium on Circuits and Systems*. Vol. 2. IEEE, 2003, pp. 860-863
- [9] AT Shaheen, "Design and Simulation of parallel CDMA System Based on 3D-Hadamard Transform," *Journal of Engineering Sciences*, vol. 6, no. 4, pp. 54-69, Dec. 2013.
- [10] P. Mazumder, S. Chandra, S. Rana, M. Mukhopadhyay and M. K. Naskar, "Parallel Hardware Implementation of Walsh Hadamard Transform," *Journal of Scientific & Industrial Research*, vol. 81, no. 7, pp. 748-753, July 2022
- [11] P. K. Meher and J. C. Patra, "Fully-pipelined efficient architectures for FPGA realization of discrete Hadamard transform," *2008 International Conference on Application-Specific Systems, Architectures and Processors, Leuven, Belgium*, 2008, pp. 43-48.
- [12] S. Hafizullah, M. S. S. V. Srikrishna Manideep, V. Sharma, P. Nath, A. Naugarhiya and S. Verma, "An Efficient Hardware Implementation of Walsh Hadamard Transform for JPEG XR," *2018 15th IEEE India Council International Conference (INDICON), Coimbatore, India*, 2018, pp. 1-4.
- [13] S. S. U. Qadri et al., "Hardware implementation of fast-sequence ordered complex hadamard transform," *2014 IEEE 10th International Colloquium on Signal Processing and its Applications, Kuala Lumpur, Malaysia*, 2014, pp. 106-110.
- [14] G. Bi, A. Aung and B. P. Ng, "Pipelined Hardware Structure for Sequence-Ordered Complex Hadamard Transform," *IEEE Signal Processing Letters*, vol. 15, pp. 401-404, 2008.
- [15] H. Yoo and I. -C. Park, "Partially Parallel Encoder Architecture for Long Polar Codes," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 62, no. 3, pp. 306-310, March 2015.
- [16] M. Dali, R. M. Gibson, A. Amira, A. Guessoum and N. Ramzan, "An efficient MIMO-OFDM radix-2 Single-Path Delay Feedback FFT implementation on FPGA," *2015 NASA/ESA Conference on Adaptive Hardware and Systems (AHS), Montreal, QC, Canada*, 2015, pp. 1-7.
- [17] A. Amira and S. Chandrasekaran, "Power Modeling and Efficient FPGA Implementation of FHT for Signal Processing," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 15, no. 3, pp. 286-295, March 2007.
- [18] Reynoso-Godinez, R., et al. "FPGA-based parallel process for Walsh-Hadamard transform." *Electronics Letters* 56.20 (2020), pp. 1039-1041.

저 자 소 개



황 지 우(학생회원)
2023년 충남대학교 전자공학과
학사 졸업.
2023년~현재 충남대학교
전자공학과 석사과정.

<주관심분야: GNSS 수신기 하드웨어, GNSS 시
물레이터>



황 용 택(학생회원)
2023년 충남대학교 전자공학과
학사 졸업.
2023년~현재 충남대학교
전자공학과 석사과정.

<주관심분야: GNSS 수신기 하드웨어>



김 민 수(학생회원)
2021년 군산대학교 전자공학과
학사 졸업.
2023년 충남대학교 전자공학과
석사 졸업.
2023년~현재 LIG NEX1 연구원.

<주관심분야: GNSS 수신기 하드웨어>



유 호 영(정회원)
2010년 연세대학교
전기전자공학부
학사 졸업.
2012년 KAIST 전자공학과
석사 졸업.
2016년 KAIST 전자공학과
박사 졸업.

2016년 삼성전자 메모리사업부 책임 연구원.
2016년~2020년 충남대학교 전자공학과 조교수.
2020년~현재 충남대학교 전자공학과 부교수.
<주관심분야: Error Correction Code, FPGA역공
학, FPGA 플랫폼, PIM, GNSS 수신기 하드웨어>