새로운 이중 색인 사상에 의한 다차원 DFT의 파이프라인 구조 개발

論 文
56-4-23

A New Two-Level Index Mapping Scheme for Pipelined Implementation of Multidimensional DFT

庾 盛 郁†
(Sungwook Yu)

Abstract - This paper presents a new index mapping method for DFT (Discrete Fourier Transform) and its application to multidimensional DFT. Unlike conventional index mapping methods such as DIT (Decimation in Time) or DIF (Decimation in Frequency) algorithms, the proposed method is based on two levels of decomposition and it can be very efficiently used for implementing multidimensional DFT as well as 1-dimensional DFT. The proposed pipelined architecture for multidimensional DFT is very flexible so that it can lead to the best tradeoff between performance and hardware requirements. Also, it can be easily extended to higher dimensional DFTs since the number of CEs (Computational Elements) and DCs (Delay Commutators) increase only linearly with the dimension. Various implementation options based on different radices and different pipelining depths will be presented.

Key Words : Multidimensional DFT, FFT, Index Mapping, Pipelined Architecture

1. 서 론

DFT (Discrete Fourier Transform)는 디지털 비디오/오디오 방송, WLAN, DSL, OFDM 등등 수많은 통신 및 신호처리 분야에서 매우 널리 사용되고 있다. 이에 따라 지난 수십 년 동안 DFT를 빠르고 효율적으로 계산하는 FFT (Fast Fourier Transform) 알고리즘과 그러한 FFT 알고리즘을 효율적으로 구현하는 아키텍처 개발에 많은 연구가 있어 왔다. 또한, 다차원 DFT 역시 영상 처리 등의 응용 분야에서 널리 쓰이고 있으며 다차원 DFT에 대한 빠른 알고리즘과 효율적인 VLSI 구현 방법에 대해서도 많은 연구가 이루어지고 있다[1-3].

FFT 알고리즘은 원래의 1차원 색인을 분해하여 2차원 또는 다차원의 색인으로 재구성하는 방법을 사용하며 가장 널리 쓰이는 색인 사상 (index mapping) 기법은 DIT (Decimation in Time) 또는 DIF (Decimation in Frequency) 알고리즘이다 [4-5]. 본 논문에서는 기존의 DIT나 DIF와는 다른 이중 색인 사상 기법을 제안하며 이를 바탕으로 한 FFT의 파이프라인 구조가 기존의 색인 사상 기법을 이용한 파이프라인 구조와 어떻게 다른지 비교, 설명한다. 또한 제안된 이중 색인 사상 기법이 어떻게 다차원 DFT의 파이프라인 아키텍처에 효과적으로 사용할 수 있는지 설명한다. 제안된 파이프라인 구조의 radix와 파이프라인 깊이를 조정함으로써 최고의 성능과 최소의 하드웨어 사이에서 최적의 조합을 선택할 수 있다. 또한 차수의 증가에 따라 요구되는 하드웨어가 기하급수적으로 증가하지 않고 선형으로만 증가하기 때문에 높은 차수의 다차원 DFT 구현에도 쉽게 적용할 수 있는 장점이 있다.

본 논문의 구성은 다음과 같다. 우선 2장에서는 기존의 DIT나 DIF 등의 색인 사상 기법과 그를 바탕으로 한 FFT의 파이프라인 구조에 대하여 설명한다. 3장에서는 제안된 이중 색인 사상 기법을 기존의 방법과 비교하고 각각의 색인 사상 기법을 사용했을 때의 FFT의 파이프라인 구조가 어떻게 다른지 비교, 설명한다. 또한 제안된 색인 사상 기법을 다차원 DFT의 구현에 효율적으로 적용하는 방법을 설명하며 2차 색인 분해 과정에서의 radix 변화에 따른 파이프라인 구조의 성능과 필요 하드웨어를 분석한다. 마지막으로 4장에서는 본 논문의 요약 및 결론이 제시된다.

2. FFT 알고리즘의 파이프라인 구조

1차원 M-포인트 DFT는 다음의 식으로 정의된다.

$$X(k)=\sum_{n=0}^{M-1}x(n)W_M^{nk} \quad 0\le k\le M-1 \tag{1}$$

여기서 $W_M=\exp(-j2\pi/M)$ 이다.

식 (1)의 M이 어떤 정수 r의 거듭제곱 형태일 경우, 다양한 색인 사상 방법이 존재하며 그 중 가장 널리 알려진 방법 중 하나가 DIT (Decimation in Time) 기법이다 [4-5]. 가령, $M=r^4$일 경우, 다음의 식으로 주어지는 DIT 색인 사상 방법을 적용하면

† 교신저자, 正會員 : 中央大學 電子電氣工學部 助教授 · 工博
E-mail : sungwook@cau.ac.kr
接受日字 : 2007年 1月 30日
最終完了 : 2007年 3月 8日

$$\begin{cases} n = rn_C + n_d & 0 \le n_C \le r^3 - 1,\ 0 \le n_d \le r-1 \\ k = k_C + r^3 k_d & 0 \le k_C \le r^3 - 1,\ 0 \le k_d \le r-1 \end{cases}$$
$$\begin{cases} n_C = rn_B + n_c & 0 \le n_B \le r^2 - 1,\ 0 \le n_c \le r-1 \\ k_C = k_B + r^2 k_c & 0 \le k_B \le r^2 - 1,\ 0 \le k_c \le r-1 \end{cases}$$
$$\begin{cases} n_B = rn_a + n_b & 0 \le n_a \le r-1,\ 0 \le n_b \le r-1 \\ k_B = k_a + rk_b & 0 \le k_a \le r-1,\ 0 \le k_b \le r-1 \end{cases} \quad (2)$$

식 (1)은 다음의 식으로 변환된다.

$$X(k_a + rk_b + r^2 k_c + r^3 k_d) = \sum_{n_d=0}^{r-1}\sum_{n_c=0}^{r-1}\sum_{n_b=0}^{r-1}\sum_{n_a=0}^{r-1} x(r^3 n_a + r^2 n_b + rn_c + n_d) W_r^{n_a k_a} W_{r^2}^{n_b k_a} W_r^{n_b k_b} W_{r^3}^{n_c (k_a + rk_b)} W_r^{n_c k_c} W_{r^4}^{n_d (k_a + rk_b + r^2 k_c)} W_r^{n_d k_d} \quad (3)$$

그림 1은 r=4, M=256인 경우에 DIT 기법을 사용하는 M-포인트 1차원 FFT의 파이프라인 구조를 보여준다[6-7]. 그림의 파이프라인 구조는 $\log_r M$ 개의 CE (Computational Element)와 $(\log_r M)-1$ 개의 DC (Delay Commutator)로 구성되어 있다. 각각의 CE는 r-포인트 버터플라이 계산과 twiddle 인자 곱셈을 수행하며 DC는 적절한 입력 데이터가 다음 단의 CE로 전달될 수 있도록 데이터의 흐름을 재조정하는 역할을 한다. 그림 2는 r=4인 경우의 CE와 DC의 내부 구조를 나타낸다. 그림의 파이프라인 구조에서 ROM은 FFT 계산에 필요한 twiddle 인자를 저장하는데 그림 1의 경우, 식 (3)에 있는 twiddle 인자인 $W_{r^2}^{n_b k_a}$, $W_{r^3}^{n_c (k_a + rk_b)}$, $W_{r^4}^{n_d (k_a + rk_b + r^2 k_c)}$를 저장하는 역할을 한다. 이 때, ROM의 높이는 k_a, $(k_a + rk_b)$, $(k_a + rk_b + r^2 k_c)$의 범위에 의해 결정되므로 r=4인 그림 1의 경우, ROM의 높이는 각각 4, 16, 64가 된다. 한편, ROM의 폭은 n_b, n_c, n_d의 범위에 의해 결정되므로 그 폭은 원래 r(=4)이어야 한다. 그러나 n_b, n_c, n_d의 값이 0일 경우 (즉, 그림 2(a)의 4개의 데이터 선들 중에서 제일 위의 선에 해당), twiddle 인자의 값이 항상 1이 되어 그 twiddle 인자와의 곱셈은 생략할 수 있으므로 필요한 곱셈기의 수는 r-1(=3)이 되며 따라서 ROM의 폭 역시 r-1(=3)이 된다.

3. 이중 색인 사상 기법과 다차원 DFT에의 적용

앞 장의 DIT 또는 DIF 알고리즘과는 달리 제안된 색인 사상 구조는 2단계의 색인 분해 과정을 거쳐 FFT 알고리즘을 얻는다. 가령, $M = N^2 = r^4$일 경우, 다음의 1단계 색인 사상을 사용하면

$$\begin{cases} n = Nn_0 + n_1 & 0 \le n_0 \le N-1,\ 0 \le n_1 \le N-1 \\ k = k_0 + Nk_1 & 0 \le k_0 \le N-1,\ 0 \le k_1 \le N-1 \end{cases} \quad (4)$$

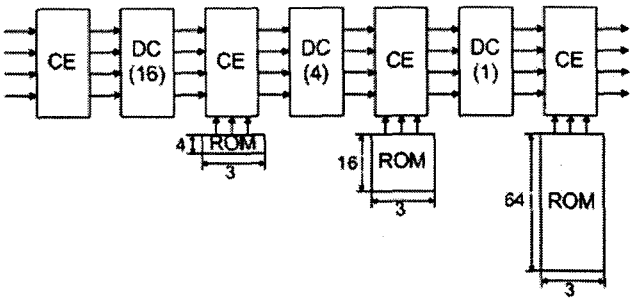
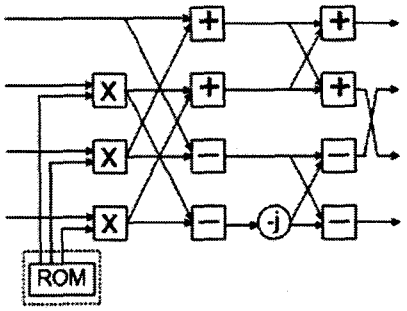
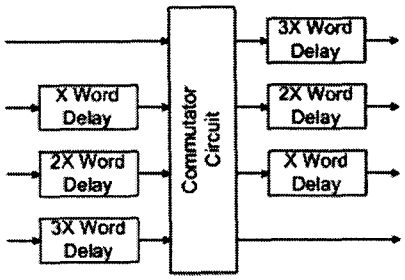


그림 1 DIT 방식을 사용한 256-포인트 Radix-4 파이프라인 FFT 아키텍처

Fig. 1 Radix-4 Pipelined Architecture for 256-Point 1-D FFT by DIT



(a) Computational Element



(b) Delay Commutator

그림 2 Radix-4 파이프라인 FFT 프로세서의 구성 요소

Fig. 2 Radix-4 Pipelined FFT Processor Compenents

식 (1)은 다음의 식으로 변환된다.

$$X(k_0 + Nk_1) = \sum_{n_1=0}^{N-1}\sum_{n_0=0}^{N-1} x(Nn_0 + n_1) W_N^{n_0 k_0} W_{N^2}^{n_1 k_0} W_N^{n_1 k_1} \quad (5)$$

그 다음, 아래의 2단계 색인 사상을 적용하면

$$\begin{cases} n_0 = rn_a + n_b & 0 \le n_a, n_b \le r-1 \\ n_1 = rn_c + n_d & 0 \le n_c, n_d \le r-1 \end{cases}$$
$$\begin{cases} k_0 = k_a + rk_b & 0 \le k_a, k_b \le r-1 \\ k_1 = k_c + rk_d & 0 \le k_c, k_d \le r-1 \end{cases} \quad (6)$$

식 (5)는 결국 다음의 식으로 변환된다.

$$X(k_a + rk_b + r^2k_c + r^3k_d) = \sum_{n_d=0}^{r-1}\sum_{n_c=0}^{r-1}\sum_{n_b=0}^{r-1}\sum_{n_a=0}^{r-1} x(r^3n_a + r^2n_b + rn_c + n_d) W_r^{n_a k_a} W_{r^2}^{n_b k_a} W_r^{n_b k_b} W_{r^4}^{(rn_c+n_d)(k_a+rk_b)} W_r^{n_c k_c} W_{r^2}^{n_d k_c} W_r^{n_d k_d} \quad (7)$$

식 (3)과 식 (7)의 비교에서 알 수 있듯이, 본 논문에서 제안하는 색인 사상 기법은 기존의 DIT 또는 DIF와는 다른 새로운 FFT 알고리즘을 유도하며 그림 3은 제안된 색인 사상 기법을 사용한 256-포인트 FFT의 파이프라인 구조를 보여준다. $256 = 16^2 = 4^4$ 이므로 이 FFT 아키텍처는 식 (7)에서 r=4인 경우에 해당한다. 그림 1의 DIT를 사용한 파이프라인 구조와 비교해 볼 때, CE와 DC 등의 전체적인 배치에 있어서는 2가지 구조가 서로 유사하나 twiddle 인자를 저장하는 ROM의 크기와 내용 등에 있어서 차이가 있음을 알 수 있다.

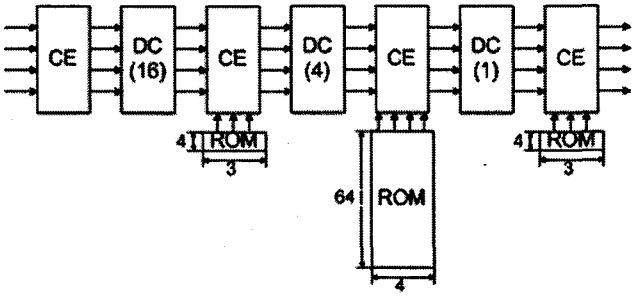


그림 3 제안된 색인 사상 기법을 사용한 256-포인트 Radix-4 파이프라인 FFT 아키텍처

Fig. 3 Radix-4 Pipelined Architecture for a 256-Point 1-D FFT by proposed index mapping scheme

제안된 2중 색인 사상 기법은 다차원 DFT의 구현에 더욱 효과적으로 적용될 수 있다. 가령, 다음의 $N \times N$-포인트 2차원 DFT를 고려해 보자.

$$X(k_0, k_1) = \sum_{n_1=0}^{N-1}\sum_{n_0=0}^{N-1} x(n_0, n_1) W_N^{n_0 k_0} W_N^{n_1 k_1} \quad (8)$$

식 (5)와 (8)의 비교를 통해, N^2-포인트 1차원 FFT의 계산 과정에서 twiddle 인자 곱셈 과정을 생략함으로써 $N \times N$-포인트 2차원 DFT를 계산할 수 있음을 알 수 있다. 이러한 성질은 2번째 레벨의 색인 분해 과정 후에도 성립하므로 식 (7)의 식에서 $W_{r^4}^{(rn_c+n_d)(k_a+rk_b)} \left(= W_{N^2}^{n_1 k_0}\right)$에 의한 twiddle 인자 곱셈 과정만 생략하면 $N \times N$-포인트 2차원 DFT를 계산할 수 있다. 그림 4(a)는 제안된 색인 사상 기법을 이용하여 16×16-포인트 2차원 DFT를 계산하는 radix-4 파이프라인 구조를 보여준다. 그림 4(a)의 파이프라인 구조에는 1번째 레벨의 색인 분해 과정에서 생성된 twiddle 인자 ROM (즉, 그림 3에서 2번째 ROM)이 없으나 첫 번째와 세 번째 ROM은 2번째 레벨의 색인 분해 과정에서 생성된 것으로서 이들은 생략할 수 없다. 결론적으로, 제안된 색인 사상 기법을 이용하면 1차원 FFT의 파이프라인 구조를 효과적으로 이용하여 2차원 DFT를 구현할 수 있음을 알 수 있다.

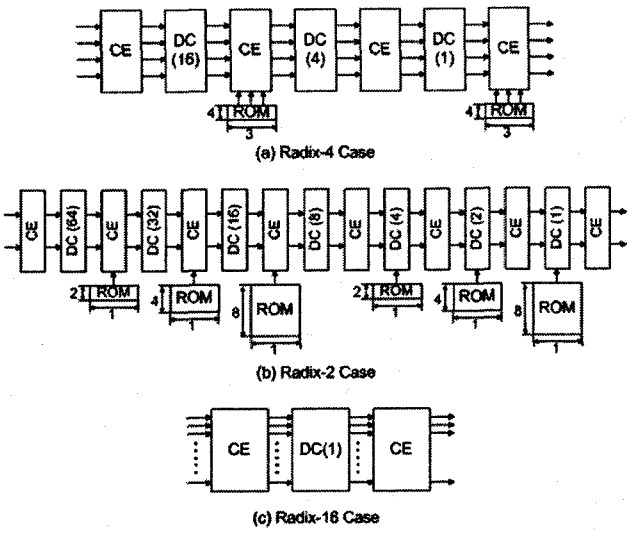


그림 4 16×16 2차원 DFT를 위한 파이프라인 구조

Fig. 4 Pipelined Architecture for 16×16 2-D DFT

제안된 2중 색인 사상 기법의 또 다른 장점은 주어진 응용분야의 요구조건에 맞춰 DFT 회로의 성능과 사이즈를 적절하게 조절할 수 있다는 점이다. 그림 4(b)는 16×16-포인트 2차원 DFT를 계산하는 radix-2 파이프라인 구조를 보여준다. 그림 4(a)의 radix-4의 경우에 비해 radix-2 파이프라인 구조는 더 많은 수의 Computational Element와 Delay Commutator를 요구하나 각각의 CE와 DC의 복잡도가 radix-4의 경우에 비해 훨씬 간단하기 때문에 전체적으로 회로의 복잡도 및 회로가 차지하는 면적은 작아지게 된다. 그러나 radix-4 파이프라인 구조에서는 매 clock마다 4개의 데이터가 출력되는 반면, radix-2 파이프라인 구조에서는 매 clock마다 2개의 데이터가 출력되므로 회로의 성능 면에서는 radix-4의 경우가 더욱 유리하다고 할 수 있다. 반대로 그림 4(c)에 나타난 radix-16의 경우, CE와 DC의 개수는 radix-4의 경우보다 훨씬 작으나 각각의 CE와 DC의 복잡도가 매우 높기 때문에 전체 회로의 복잡도는 더 커지게 된다. 그러나 radix-16 파이프라인 구조의 경우, 매 클럭마다 16개의 데이터가 출력되므로 성능 면에서는 가장 유리하다고 할 수 있다. 사실, radix-16 파이프라인 구조는 2번째 레벨에서의 색인 분해 과정을 생략한 형태로서 이는 2차원 DFT를 구현하기 위해 널리 사용되는 row-column 방법의 특수한 형태로 볼 수 있다. 따라서 그림 4(c)에 나타난 2개의 CE 사이의 DC는 16×16 행렬의 행과 열을 바꾸는 전치(transposition)의 역할을 하게 된다.

표 1 16×16 2차원 DFT 파이프라인 구조의 성능과 복잡도

Table 1 Performance and Hardware Requirements for 16×16 2-D DFT

구분	throughput (매 클럭당 출력 데이터의 수)	클럭의 주기	필요한 레지스터의 수	필요한 CE (DC)의 수	필요한 곱셈기의 수	필요한 덧셈기의 수	ROM의 크기
Radix-16 Architecture	16	$2t_{mul}+4t_{add}$	240+16×2	2 (1)	16	128	
		t_{mul}	240+16×12				
		$\approx t_{add}$	240+16×16				
Radix-4 Architecture	4	$t_{mul}+2t_{add}$	252+4×4	4 (3)	6	32	24
		t_{mul}	252+4×10				
		$\approx t_{add}$	252+4×12				
Radix-2 Architecture	2	$t_{mul}+t_{add}$	254+2×8	8 (7)	6	16	28
		t_{mul}	254+2×14				
		$\approx t_{add}$	254+2×20				

표 1은 16×16-포인트 2차원 DFT를 서로 다른 radix를 사용하여 파이프라인 구조로 구현했을 때의 성능과 파이프라인 구조에 필요한 하드웨어의 복잡도를 보여준다. 일반적으로, CE 내에서 요구되는 클럭의 주기가 DC 내에서 요구되는 클럭의 주기보다 훨씬 크므로 CE 내부를 적절히 파이프라인 구조로 재구성함으로써 더욱 효과적인 DFT 파이프라인 구조를 만들 수 있다. 표 1에서는 각각의 radix에 대해서로 다른 3가지의 파이프라인 옵션을 보여준다. 가령, 각 radix의 클럭 주기에서 첫 번째 칸은 CE 내부를 전혀 파이프라인 구조로 바꾸지 않은 경우에 해당한다. 즉, CE 전체를 하나의 클럭 주기 내에서 계산하고 그 최종 결과값을 레지스터에 저장하는 방식이다. 이 경우, 회로 복잡도는 작은 편이나 클럭 주기가 다른 경우에 비해 매우 커지는 단점이 있다. 두 번째 칸은 CE 내에 있는 모든 덧셈기와 곱셈기의 끝에 레지스터를 추가하는 방식이다. 이 경우, 전체 레지스터의 수가 증가하여 회로 복잡도가 약간 증가하나 첫 번째 경우에 비해 클럭 주기가 많이 감소하므로 대개의 경우, 첫 번째 방법보다는 더 좋은 구현 방법이 된다. 세 번째 칸은 각각의 곱셈기의 내부를 파이프라인 구조로 바꾼 경우에 해당한다. 가령, $t_{mul} \approx 2 \times t_{add}$ 를 가정할 경우, 각각의 곱셈기를 2개의 파이프라인 스테이지로 분할함으로써 좀 더 효과적인 파이프라인 구조를 만들 수 있다. 물론, 각각의 덧셈기까지도 파이프라인 구조로 만들고 곱셈기의 파이프라인 구조를 더욱 심화하여 더 세부적인 파이프라인 구조로 만들 수도 있다[8]. 그러나, 이 경우에는 덧셈기 및 곱셈기 내부의 레지스터의 수가 너무 많이 증가하여 전체 회로 복잡도에 크게 영향을 미칠 수 있으므로 주의를 기울여야 한다.

앞서 언급한 바와 같이, 제안된 2중 색인 사상 기법은 3차원 이상의 다차원 DFT의 구현에도 쉽게 확장, 적용될 수 있다. 가령, 식 (1)의 M-포인트 1차원 DFT에서 $M = N^3$ 이고 $N = r^3$ 일 경우, 다음의 식으로 주어지는 1단계 색인 사상을 적용하면

$$\begin{cases} n = Nn_z + n_2 & 0 \le n_z \le N^2-1,\ 0 \le n_2 \le N-1 \\ k = k_z + N^2 k_2 & 0 \le k_z \le N^2-1,\ 0 \le k_2 \le N-1 \end{cases}$$
$$\begin{cases} n_z = Nn_0 + n_1 & 0 \le n_0 \le N-1,\ 0 \le n_1 \le N-1 \\ k_z = k_0 + Nk_1 & 0 \le k_0 \le N-1,\ 0 \le k_2 \le N-1 \end{cases} \quad (9)$$

식 (1)은 다음의 식으로 변환된다.

$$X(k_0 + Nk_1 + N^2k_2) = \sum_{n_2=0}^{N-1}\sum_{n_1=0}^{N-1}\sum_{n_0=0}^{N-1} x(N^2n_0 + Nn_1 + n_2) W_N^{n_0k_0} W_{N^2}^{n_1k_0} W_N^{n_1k_1} W_{N^3}^{n_2(k_0+Nk_1)} W_N^{n_2k_2} \quad (10)$$

이제 다음과 같이 주어지는 2단계 색인 사상을 적용하면

$$\begin{cases} n_0 = rn_{0z} + n_{02} \\ k_0 = k_{0z} + r^2k_{02} \end{cases} \quad \begin{cases} n_{0z} = rn_{00} + n_{01} \\ k_{0z} = k_{00} + rk_{01} \end{cases}$$
$$\begin{cases} n_1 = rn_{1z} + n_{12} \\ k_1 = k_{1z} + r^2k_{12} \end{cases} \quad \begin{cases} n_{1z} = rn_{10} + n_{11} \\ k_{1z} = k_{10} + rk_{11} \end{cases}$$
$$\begin{cases} n_2 = rn_{2z} + n_{22} \\ k_2 = k_{2z} + r^2k_{22} \end{cases} \quad \begin{cases} n_{2z} = rn_{20} + n_{21} \\ k_{2z} = k_{20} + rk_{21} \end{cases} \quad (11)$$

식 (10)은 다음의 식으로 변환된다.

$$\begin{aligned} &X(k_{00} + rk_{01} + r^2k_{02} + r^3k_{10} + r^4k_{11} + r^5k_{12} + r^6k_{20} + r^7k_{21} + r^8k_{22}) \\ &= \sum_{n_{22}=0}^{r-1}\sum_{n_{21}=0}^{r-1}\sum_{n_{20}=0}^{r-1}\sum_{n_{12}=0}^{r-1}\sum_{n_{11}=0}^{r-1}\sum_{n_{10}=0}^{r-1}\sum_{n_{02}=0}^{r-1}\sum_{n_{01}=0}^{r-1}\sum_{n_{00}=0}^{r-1} \\ &x(r^8n_{00} + r^7n_{01} + r^6n_{02} + r^5n_{10} + r^4n_{11} + r^3n_{12} + r^2n_{20} + rn_{21} + n_{22}) \\ &W_r^{n_{00}k_{00}} W_{r^2}^{n_{01}k_{00}} W_r^{n_{01}k_{01}} W_{r^3}^{n_{02}(k_{00}+rk_{01})} W_r^{n_{02}k_{02}} W_{r^6}^{(r^2n_{10}+rn_{11}+n_{12})(k_{00}+rk_{01}+r^2k_{02})} \\ &W_r^{n_{10}k_{10}} W_{r^2}^{n_{11}k_{10}} W_r^{n_{11}k_{11}} W_{r^3}^{n_{12}(k_{10}+rk_{11})} W_r^{n_{12}k_{12}} \\ &W_{r^9}^{(r^2n_{20}+rn_{21}+n_{22})(k_{00}+rk_{01}+r^2k_{02}+r^3k_{10}+r^4k_{11}+r^5k_{12})} \\ &W_r^{n_{20}k_{20}} W_{r^2}^{n_{21}k_{20}} W_r^{n_{21}k_{21}} W_{r^3}^{n_{22}(k_{20}+rk_{21})} W_r^{n_{22}k_{22}} \end{aligned} \quad (12)$$

2차원 DFT의 경우와 마찬가지로 식 (12)의 FFT 공식은 3차원 DFT를 파이프라인 구조로 구현할 때 유용하게 사용될 수 있다. 즉, 식 (12)에서 $W_{r^6}^{(r^2n_{10}+rn_{11}+n_{12})(k_{00}+rk_{01}+r^2k_{02})}$ 과 $W_{r^9}^{(r^2n_{20}+rn_{21}+n_{22})(k_{00}+rk_{01}+r^2k_{02}+r^3k_{10}+r^4k_{11}+r^5k_{12})}$ 에 의한 twiddle 인자 곱셈을 생략함으로써 다음의 식으로 주어지는 $N \times N \times N$-포인

트 3차원 DFT의 구현에 이용할 수 있다.

$$X(k_0,k_1,k_2)=\sum_{n_2=0}^{N-1}\sum_{n_1=0}^{N-1}\sum_{n_0=0}^{N-1}x(n_0,n_1,n_2)\,W_N^{n_0k_0}W_N^{n_1k_1}W_N^{n_2k_2} \quad (13)$$

이 때, 2차원 DFT의 경우와 마찬가지로 2단계의 색인 분해 과정에서 radix의 조정이 가능하며 이러한 radix의 선택과 전체 아키텍처의 파이프라인 깊이의 조정에 의해 3차원 DFT 회로의 성능과 면적을 적절하게 조절할 수 있다. 또한, 이미 개발된 다양한 radix의 버터플라이를 그대로 CE의 구현에 사용할 수 있으며 DC 역시 재사용이 가능하므로[9-11] 제안된 파이프라인 아키텍처는 구현이 매우 용이하고 3차원 이상의 고차원 DFT에도 특별한 제약 없이 쉽게 확장하여 적용할 수 있다.

4. 결 론

이 논문에서는 FFT를 구현하기 위한 새로운 색인 사상 기법을 제안하였다. 기존의 DIT (Decimation in Time) 또는 DIF (Decimation in Frequency) 기법과는 달리 제안된 색인 사상 기법은 2중 분해 구조로 되어 있어서 FFT의 구현 뿐만 아니라 다차원 DFT의 구현에 더욱 효과적으로 쓰일 수 있다. 제안된 색인 사상 기법을 사용한 다차원 DFT의 파이프라인 구조는 3차원 이상의 높은 차수에도 적용될 수 있으며 구현에 필요한 회로의 복잡도가 DFT의 차수에 기하급수적으로 증가하지 않고 단지 선형으로만 증가하기 때문에 사실상 다차원 DFT의 차수에 제약을 받지 않는다. 또한, 이미 개발된 다양한 radix-2, 4, 8의 버터플라이 회로를 그대로 CE 회로에 사용할 수 있으므로 구현이 용이하고 2번째 레벨의 색인 분해 과정에 사용되는 radix의 적절한 선택 및 파이프라인 깊이의 적절한 조정을 통해 전체 DFT 파이프라인 구조의 성능과 복잡도를 쉽게 조정할 수 있다.

감사의 글

본 연구는 2005년도 중앙대학교 학술연구비 지원에 의하여 수행되었습니다. 연구비 지원에 감사드립니다.

참 고 문 헌

[1] T. Sansaloni, A. Perez-Pascual, V. Torres and J. Valls, "Efficient Pipeline FFT Processors for WLAN MIMO-OFDM Systems," Electronics Letters, Vol. 41, pp. 1043-1044, 2005.

[2] S. G. Johnson and M. Frigo, "A Modified Split-Radix FFT With Fewer Arithmetic Operations," IEEE Transactions on Signal Processing, Vol. 55, pp. 111-119, 2007.

[3] S. Bouguezel, M. O. Ahmad, and M. N. S. Swamy, "New radix-(2×2×2)/(4×4×4) and radix-(2×2×2)/(4×4×4) DIF FFT algorithms for 3-D DFT," IEEE Transactions on Circuits and Systems, Vol. 53, pp. 306-315, 2006.

[4] J. W. Cooley and J. W. Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series," Mathematics of Computation, Vol. 19, pp. 297-301, 1965.

[5] C. S. Burrus, "Index Mappings for Multidimensional Formulation of the DFT and Convolution," IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. 25, pp. 239-242, 1977.

[6] B. Gold and T. Bially, "Parallelism in Fast Fourier Transform Hardware," IEEE Transactions on Audio and Electronics, Vol. AU-21, pp. 5-16, 1973.

[7] L. R. Rabiner and B. Gold, Theory and Applications of Digital Signal Processing, Ch. 10, Englewood Cliffs, NJ: Prentice-Hall, 1975.

[8] L. Dadda and V. Piuri, "Pipelined Adders," IEEE Transactions on Computers, Vol. 45, pp. 348-356, 1996.

[9] J. D. Bruguera and T. Lang, "Implementation of the FFT Butterfly with Redundant Arithmetic," IEEE Transactions on Circuits and Systems II, Vol. 43, pp. 717-723, 1996.

[10] E. E. Swartzlander, Jr., W. K. W. Young and S. J. Joseph, "A Radix 4 Delay Commutator for Fast Fourier Transform Processor Implementation," IEEE Journal of Solid-State Circuits, Vol. 19, pp. 702-709, 1984.

[11] V. Szwarc, L. Desormeaux, W. Wong, C. Yeung, C. H. Chan and T. A. Kwasniewski, "A Chip Set for Pipeline and Parallel Pipeline FFT Architectures," Journal of VLSI Signal Processing, Vol. 8, pp. 253-265, 1994.

저 자 소 개

유 성 욱 (庾 盛 郁)

1992. 2 서울대학교 전기공학과 졸업
1996. 12 UT Austin ECE 석사
2000. 5 UT Austin ECE 박사
2000. 8 - 2004. 2 Intel Corp.
2004. 4 - 2005. 2 삼성 시스템 LSI.
2005. 3 - 현재 중앙대학교 전자전기공학부 조교수
Tel : (02) 820-5740
E-mail : sungwook@cau.ac.kr