한국통신학회 1986년도 춘계학술발표회 논문집(1986.5.24)

RADIX-2 BUTTERFLY 연산회로의 설계

°최 병 윤, 신 경 욱, 유 종 근, 임 충 빈, 김 봉 열, 이 문 기
연세대학교 전자공학과

Design of radix-2 Butterfly Arithmetic Unit.

B.Y. Choi. K.W.Shin. J.K. Yoo. C.B. Lim. B.R. Kim. M.K. Lee
Electronics Engineering Dept., Yonsei University, Seoul, Korea.

A high performance Butterfly Arithmetic Unit for FFT processor using two adders is proposed in this papers, which is Based on the distributed and merged arithmetic. Due to simple and easy architecture to implement, this proposed processor is well suited to systolic FFT processor.

Simulation was performed using YSLOG (Yonsei logic simulator) on IBM AT computer, to verify logic. By using 3um double Metal CMOS technology, Butterfly arithmetic have been achieved in 1.2 usec.

1. 서론

디지탈 신호처리는 통신, 레이다, 소나, 지진파해석등 광범위한 분야에서 널리 응용되고 있다.
따라서 실시간 처리가 불가피하게 되어, 최근에 SYSTOLIC과 WAVEFRONT 알고리즘을 사용하는 연구가 주목을 받고 있다. (1) (10)
본 논문에서는 FFT를 SYSTOLIC 방식으로 수행할 때 필요한 BUTTERFLY 연산회로를 설계했다.
BUTTERFLY 연산회로 설계시, DISTRIBUTED 연산과 MERGED 연산을 사용하여, 승산기 〈MULTIPLIER〉가 없이 2개의 가산기만으로 고속으로 DIT〈DECIMATION-IN-TIME〉 BUTTERFLY 연산을 수행하는 회로를 제시한다.
전체 동작을 확인하기 위해, 논리 시뮬레이터인 YSLOG를 이용해서 IBM-AT 에서 논리 시뮬레이션을 수행하였다.

2. FFT 알고리즘 〈2〉 〈5〉

표본수가 N인 경우, DFT는 다음과 같이 표현된다.

$$X(k) = \sum_{n=0}^{N-1} x(n)\, e^{-j(2\pi/N)nk} = \sum_{n=0}^{N-1} x(n)\, W^{nk} \qquad n,k=0,1,2,..,N-1$$

여기서 $W^{nk} = e^{-j(2\pi/N)nk}$는 회절인자 〈TWIDDLE FACTOR〉이다.
DFT는 N 의 복소수 곱셈과 N〈N-1〉 복소수 덧셈이 필요하다.
FFT는 DFT를 효율적으로 계산하는 알고리즘으로 DFT를 $\log_2 N$ STAGE로 분할하고, 각 STAGE는 $\frac{N}{2}$ 복소수 곱셈 〈COMPLEX MULTIPLICATION〉을 포함한다.
그러므로 전체 필요한 곱셈횟수는 $(\frac{N}{2})\log_2 N$ 이다.
〈그림 1〉과 〈그림 2〉은 N=8인 경우, FFT의 연산과 DIT BUTTERFLY 연산 흐름이며, 이때 BUTTERFLY 연산식은,

$$X = A + B \cdot W^k \quad \text{----- 〈1〉}$$
$$Y = A - B \cdot W^k \quad \text{- - - (2)}$$

와 같다.
윗식의 모든 복소수 항을 실수 항과 허수항으로 표시하여 연산회수를 계산해 보면, 4번의 실수곱셈과 6번의 실수 덧셈이 필요함을 알수있다. 또 FFT 연산의 최종 결과는 BIT-REVERSED 형식으로 얻어지므로, 적당한 데이타 정돈 〈 SCRAMBLING〉 과정이 필요하다.

3. DISTRIBUTED 연산과 MERGED 연산

디지탈 신호처리 회로를 집적화하는데 가장 큰 문제점이 승산기 이다.

DISTRIBUTED 연산과 MERGED 연산을 결합해서 사용함으로써, 승산기가 없이 2개의 가산기 만으로 수행되는 고정 소숫점 BUTTERFLY 연산 알고리즘을 제시한다.

식 (1)에서 $P=W^{k}\cdot B$이고, N 비트 입력값은 정규화 (NORMALIZED)되어 입력된다고 하면, DISTRIBUTED 연산은 다음과 같이 구현된다. (3) (9)

$$B = Br + jBi$$
$$Br = -bro + \sum_{n=1}^{N-1} brn\, 2^{-n} \qquad ---- (3)$$
$$Bi = -bio + \sum_{n=1}^{N-1} bin\, 2^{-n}$$

식 (3)을 풀어쓰면,

$$Br = \frac{1}{2}Br - (-\frac{1}{2}Br)$$
$$= \frac{1}{2}\left[-bro + \sum_{n=1}^{N-1} brn\, 2^{-n}\right] - \frac{1}{2}\left[-\overline{bro} + \sum_{n=1}^{N-1}\overline{brn}\, 2^{-n} + 2^{-(N-1)}\right]$$
$$= \frac{1}{2}\left[-(bro-\overline{bro}) + \sum_{n=1}^{N-1}(brn-\overline{brn})2^{-n} + 2^{-(N-1)}\right] \qquad ---- (4)$$

동일한 과정으로

$$Bi = \frac{1}{2}\left[-(bio-bio) + \sum_{n=1}^{N-1}(bin-bin)2^{-n} + 2^{-(N-1)}\right] \qquad ----- (5)$$

식 (4)와 (5)를 결합하면,

$$P = (Wr+jWi)(Br+jBi)$$
$$= (WrBr-WiBi) + j(WiBr+WrBi)$$
$$= \frac{1}{2}(Wr-Wi)2^{-(N-1)} - \frac{1}{2}\left[(bro-\overline{bro})Wr-(bio-\overline{bio})Wi\right] + \frac{1}{2}\sum_{n=1}^{N-1}\left[(brn-\overline{brn})Wr-(bin-\overline{bin})Wi\right]2^{-n}$$
$$+j\left\{\frac{1}{2}(Wr+Wi)2^{-(N-1)} - \frac{1}{2}\left[(bro-\overline{bro})Wi+(bio-\overline{bio})Wr\right] + \frac{1}{2}\sum_{n=1}^{N-1}\left[(brn-\overline{brn})Wi+(bin-\overline{bin})Wr\right]2^{-n}\right\} \qquad ------ (6)$$

식 (6)을 정돈하면,

$$Pr = \frac{1}{2}(Wr-Wi)2^{-(N-1)} + \sum_{n=0}^{N-1} qrn(brn,bin)2^{-n}$$
$$Pi = \frac{1}{2}(Wr+Wi)2^{-(N-1)} + \sum_{n=0}^{N-1} qri(brn,bin)2^{-n} \qquad ---- (7)$$

qrn 과 qin 에 대한 값은 (표 1)과 같다.

(표 1)에 보는 바와 같이, Wr과 Wi대신에 $\frac{1}{2}(Wr+Wi)$와 $\frac{1}{2}(Wr-Wi)$ 가 사용됨을 알수 있다.

식(7)에 보는 바와 같이, 복소수 곱셈은 일련의 accumulate와 shift 동작으로 효율적으로 수행될 수 있음을 볼수 있다.

내적 (inner product)를 효율적으로 계산하기 위해 제안된 merged arithmetic을 적용하여 X,Y 를 구하는데, Pr 과 Pi를 계산하는데 사용된 가산기를 그대로 사용할 수 있다.

즉

$$Xr = Ar + Pr$$
$$Yr = 2Ar - Xr \qquad ---------- (8)$$
$$Xi = Ai + Pi$$
$$Yi = 2Ai - Xi$$

식(8)에서 보는바와 같이, Xr,Yr,Xi,Yi 를 계산하는데, distributed 연산시 사용된 가산기에 결합 (merge) 시킬수 있음을 볼수 있다.[6]

4. scaling과 Butterfly 제어회로.

Parseval 이론에 의해서

$$\sum_{n=0}^{N-1} x(n) = \frac{1}{N}\sum_{n=0}^{N-1}|X(k)|^2 \qquad ------ (9)$$

이므로, scaling이 필요하게 된다.

입력데이타와 회절인자가 1보다 작으므로, Overflow를 방지하기 위해서 각 stage의 데이타 입력을 2만큼 scaling시키는 AAS (automatic array scaling) 기법을 사용했다.

즉 Butterfly 연산은 다음과 같이 변형된다.

$$X = \frac{1}{2}\left[A + B\cdot W^{k}\right] = \frac{1}{2}A + (\frac{1}{2}B)\cdot W^{k} \qquad --- (10)$$

$$Y = \frac{1}{2}\left[A - B\cdot W^{k}\right] = \frac{1}{2}A - (\frac{1}{2}B)\cdot W^{k}$$

Butterfly 연산자체가 점프(jump) 동작이 필요하지 않고 순차적으로 진행되므로 속도면에서 장점이 있고, 속수용 목적에 적합한, ring counter로 구성된 hardwired[4][8] 제어방식을 사용해서 설계했다.

5. Butterfly 연산회로 구조.

(그림 3)에 보는바와 같이 butterfly 연산회로가 구성된다.

연산의 속도가 가산기(adder) 부분에 의해서 좌우되므로, carry lookahead 가산기를 이용하였다.

6. 시뮬레이션과 결과 고찰.

Butterfly 연산회로에서 가장 중요한 역할을 하는 가산기를 YSS CMOS Cell Library[11] 를

를 이용하여 구성하고 논리 시뮬레이션을 행하였다. (그림 4, 5)
8bit 데이타의 경우, 12개의 명령 사이클이 필요하다.
클락주파수가 10MHZ라면, butterfly 연산시간은 1.2㎲가 된다.

기존방식과의 비교가 표 2에 보여진다.

7. 결 론

본 논문에서는 distributed 연산과 merged 연산을 결합하여 두개의 가산기로 구성된, butterfly 연산회로를 제안했다.

제안한 방식은 승산기를 사용하지 않았으므로, 기존의 4개의 승산기를 사용하는 방식에 비해 50% 이상의 면적감소를 얻을 수 있다.
연속속도는 단일 승산기를 사용하는 경우나 4개의 가산기를 사용해서 bit-serial로 distributed 연산을 하는 방식에 비해 훨씬 빠르고, 다수의 승산기를 사용하는 경우와 비슷하다는 결과를 얻을 수 있었다.
따라서 제안한 butterfly 연산회로를 wavefront 또는 systolic 신호처리회로에 바람직하다.

8. 참고문헌

(1) T. Willey, etc. "systolic implementation for deconvolution, DFT and FFT."
IEE proceedings, F. Oct. 1985 (PP. 466)

(2) J.A. Johnston.
"Parallel pipeline fast Fourier transformer"
IEE, proceedings, F. Oct. 1983 (pp. 504)

(3) I.R. MACTAGGART and M.A. Jack
"Radix-2 Butterfly Processor Using Distributed Arithmetic"
ELECTRONICS LETTERS Jun 1983 (pp. 43)

(4) Panny Cohen
"Simplified Control of FFT Hardware"
IEEE, Trans. ASSP DEC. 1976 (pp. 517)

(5) DEEPAK BHAGAT, and H.W. MERGLER
"High - performance Microprocessor-Based FFT processor"
IEEE, Trans. IESI. May 1978 (pp.102)

(6) EARL.E, SWARTZLANDER
"Merged Arithmetic"
IEEE, Trans. Computer. oct. 1980 (pp.946)

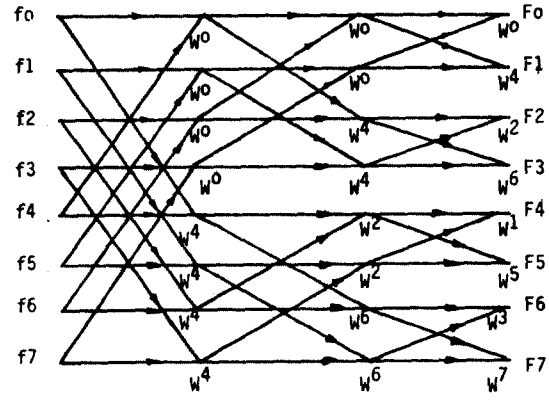
(7) ZAHEER M.ALI
"A high Speed FFT processor"
IEEE. Trans. Communication May 1978

(8) Richard W.Linderman etc.
"CUSP : A 2μm CMOS Signal Processor"
IEEE. J.Solid-State Circuit Jun 1985

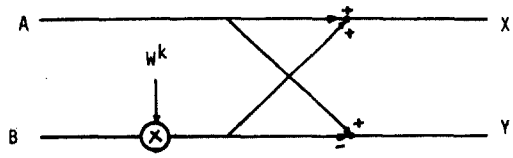
(9) Stanley A. White
"A simple FFT Butterfly Arithmetic Unit"
IEEE. Trans. CAS Apr. 1981.

(10) 신경욱 등
"집적화된 FFT 연산용 systolic Array의 설계"
추계 학술 발표 논문회집. 1985. 11.

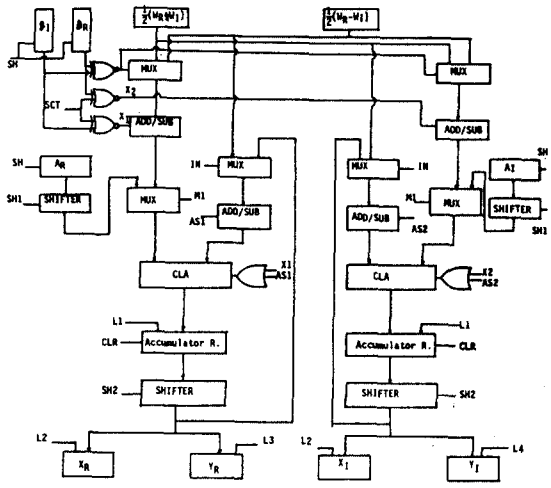
(11) M.K. Lee, et. al.,
"Development of VLSI Design Methodology,"
Jour. of the Engineering Research Institute Yonsei Univ., Vol. 17, No. 2, 1985.



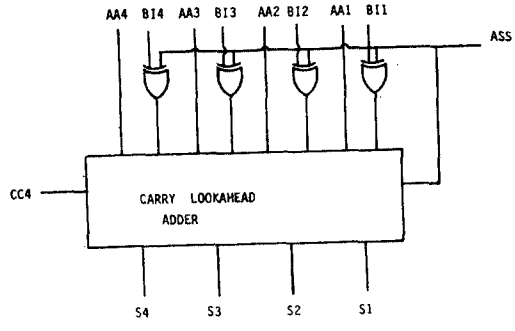
(그림 1) 연산 흐름도.



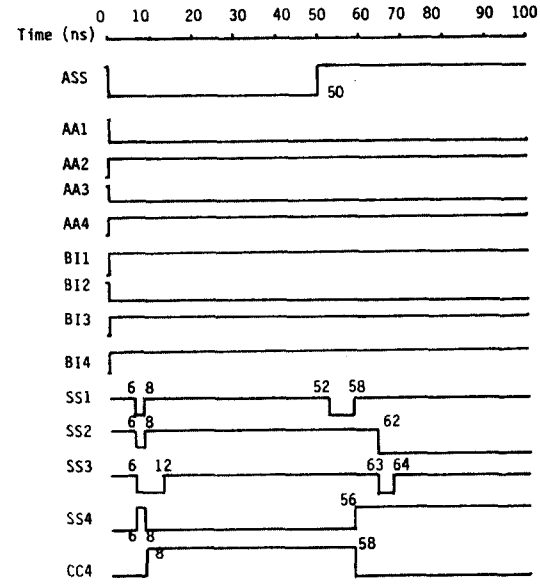
(그림 2) Butterfly연산 흐름도



(그림 3) 전체 Butterfly 연산회로



(그림 4) Carry lookahead adder/subtractor



(그림5) CLA의 Simulation 결과

brn	bin	qrn (n≠0)	qin (n≠0)
0	0	-(Wr-Wi)/2	-(Wr+Wi)/2
0	1	-(Wr+Wi)/2	+(Wr-Wi)/2
1	0	+(Wr+Wi)/2	-(Wr-Wi)/2
1	1	+(Wr-Wi)/2	+(Wr+Wi)/2

(n=0 인 경우 부호가 반대임)

(표 1) 데이타 선택표

	단일 승산기 사용	다수의 승산기 사용	제안된 방식
비 트 수	16 비트	20 비트	8 비트
주파수	5MHZ	50 MHZ	10 MHZ
연산시간	5.27 us	1.04 us	1.2 us

(표 2) Butterfly 연산시간 비교