논문 2008-45SD-8-5

곱셈기를 사용하지 않은 고속 FIR 필터를 위한 부분 항 덧셈 방법

(The Method of Addition Subexpression for High-Speed Multiplierless FIR Filters)

김 용 은*

(Yong-Eun Kim)

요 약

곱셈기를 사용하지 않는 FIR필터는 Common Subexpression 알고리즘을 이용하여 덧셈만으로 필터를 구현한다. 따라서 곱셈기를 이용한 필터 보다 적은 면적으로 필터를 구현할 수 있다. 그런데 덧셈에서 발생하는 캐리 리플로 인하여 필터 연산시간이 길어지는 단점이 있다. 본 논문에서는 CSE방식의 FIR 필터에서 부분 항을 더할 때 최종 덧셈이 수행되는 곳까지 더해지는 부분 항을 2줄로 유지하여 덧셈의 캐리 리플을 피하여 필터의 부분 항 덧셈 시간을 단축 시켰다. 제안한 알고리즘을 증명하기 위해 논문에서 주어진 예제를 이용하여 FIR 필터의 부분 항 덧셈 회로를 설계하여 하이닉스 0.18라이브러리로 합성한 결과 기존 파이프라인을 사용한 설계 방법 보다 면적, 속도에서 53.2%, 57.9%의 이득 있음을 알 수 있다.

Abstract

Multiplierless FIR filters can be designed by only adders using Common Subexpression algorithm. It has small area compared with filter which using multipliers. But it has long operation time because of carry ripple from the adder. In this paper, when the subexpressions are added in multiplier less filters, the number of subexpressions maintains 2 until final addition to avoid carry ripple of the addition, so the subexpression addition time of the filter can be reduced. To verify proposed method, subexpression adder circuit of the FIR filter is designed using given example of paper. The designed filter was synthesized using Hynix 0.18um process. By Synopsys simulation results, it is shown that by the proposed method, area, propagation delay time can be reduced up to 53.2%, 57.9% compared with conventional design method which using pipeline.

Keywords: Multiplierless Filter, Common Subexpression, Carry ripple

I. 서 론

FIR 디지털 필터는 디지털 시스템의 많은 곳에 사용된FIR 디지털 필터는 디지털 시스템의 많은 곳에 사용된다. 일반적으로 곱셈기를 이용하여 FIR 필터를 설계하는 경우 canonic signed digit (CSD) 곱셈기를 이용한다. 곱셈기를 사용하는 경우 고속으로 동작하나 면적과 파워 소비가 많다. 파워와 면적을 줄이기 위해 곱셈기를 사용하지 않고 필터를 설계하는 경우 Common Subexpression Elimination (CSE) 알고리즘을 이용한다. 작은 면적이 이점인 CSE 방식 FIR 필터를 효율적으로 구현하기 위해 여러 가지 기법들이 연구되었다[1~4]. 그러나 CSE 알고리즘을 이용하는 경우 지연시간이 긴 단점이 있다.

본 논문에서는 CSE 알고리즘을 이용하여 FIR 필터를 설계하는 경우 긴 지연시간을 단축시키기 위해 부분 항을 최종 덧셈이 이루어지는 부분까지 2줄 씩 유지하며 더하는 방식으로 부분 항 덧셈 시간을 단축시켜 파

* 학생회원, 전북대학교 전자정보공학부
(Div. of Electronic & Information Engineering Chonbuk University)
접수일자: 2008년3월6일, 수정완료일: 2008년7월28일

이프라인을 하지 않아도 고속으로 필터를 동작시킬 수 있는 방법을 제안하였다. Ⅱ에서는 기존 CSE 알고리즘을 이용한 FIR 설계 방법, Ⅲ에서는 제안한 부분 항 덧셈 방법, Ⅳ에서는 기존의 방식과 제안한 방식으로 FIR 필터를 설계하여 시뮬레이션한 결과에 대해서 설명하고, Ⅴ에서 결론을 맺는다.

Ⅱ. CSE 알고리즘을 이용한 FIR 필터 설계 방법

면적이 작으면서 저 전력인 FIR 필터를 설계할 때 CSE 알고리즘을 이용한다. 대표적으로 사용되는 CSE 알고리즘은 Hartley 알고리즘이다[1, 3]. 또한 가장 적은 수의 덧셈을 사용하여 필터를 설계할 때 Bull-Horrocks modified 알고리즘(BHM)을 이용한다[6~7]. 하지만 BHM은 지연시간이 가장 긴 CSE 알고리즘이다.

필터의 계수가 8bit로 $h_0 = 155$, $h_1 = 109$, $h_2 = 93$, $h_2 = 98$와 같이 주어져 있을 때 필터의 출력은 식(1)과 같다.

$$y(n) = h_0x(n) + h_1x(n-1) + h_2x(n-2) + h_3x(n-3) \tag{1}$$

그림 1은 논문 [5]에서 사용된 전치형 FIR필터의 구조를 보여준다. 그림 2는 BHM 알고리즘을 이용하여 곱셈을 사용하지 않고 최소한의 덧셈을 이용하여 전치형 FIR필터를 설계한 구조도 이다[6~7].

그림 2에서 최대 지연시간은 덧셈을 6개의 딜레이이다. 필터 연산 속도를 높이기 위해서 고속 덧셈기를 사용할 경우 면적에 대한 오버헤드가 크므로 파이프라인을 그림 2의 점선과 같이 하게 되면 최대 지연시간은 덧셈 6개 에서 덧셈 2개의 지연시간으로 단축된다. 하지만 파이프라인하기 위해 지연 소자(D플리플롭×와이어의 비트)가 9개 필요하고 레이턴시가 증가하는 단점

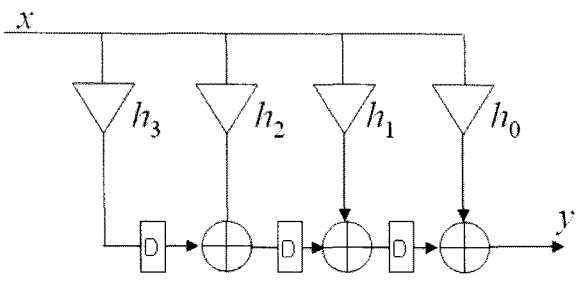

그림 1. 전치형 FIR필터의 구조
Fig. 1. Transposed FIR filter structure.

이 있다. 최대 지연시간을 예측하기 위해 그림 2에서 덧셈에서 출력된 결과를 P로 표시하였다. 입력 x가 8bit라고 할 때 $P1$은 $x + x \ll 1$의 결과이다. 오버플로우를 고려했을 때 그림 3과 같이 덧셈이 수행된다. 그림 4는 그림 3의 $P1$을 계산하기 위한 리플 캐리 덧셈

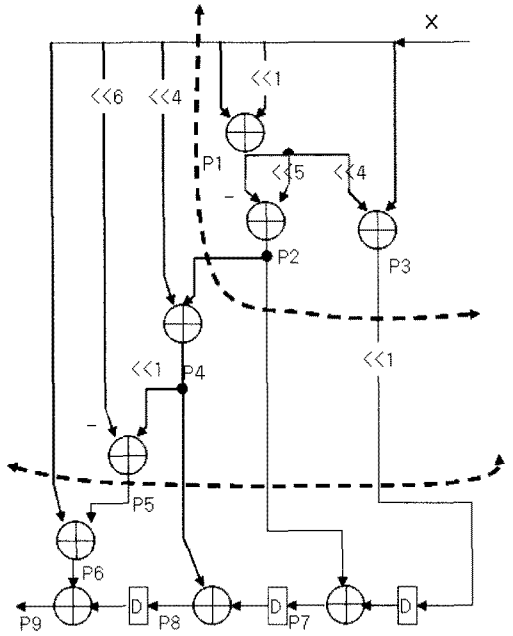

그림 2. BHM 알고리즘을 이용한 FIR 필터 구조
Fig. 2. Sturucture of FIR filter using BHM algorithm.

	x[7]	x[7]	x[7]	x[6]	x[5]	x[4]	x[3]	x[2]	x[1]	x[0]
+	x[7]	x[7]	x[6]	x[5]	x[4]	x[3]	x[2]	x[1]	x[0]	
P1[9]	P1[8]	P1[7]	P1[6]	P1[5]	P1[4]	P1[3]	P1[2]	P1[1]	P1[0]	

그림 3. 부호 확장 후 P1 연산
Fig. 3. P1 operation after sign extension.

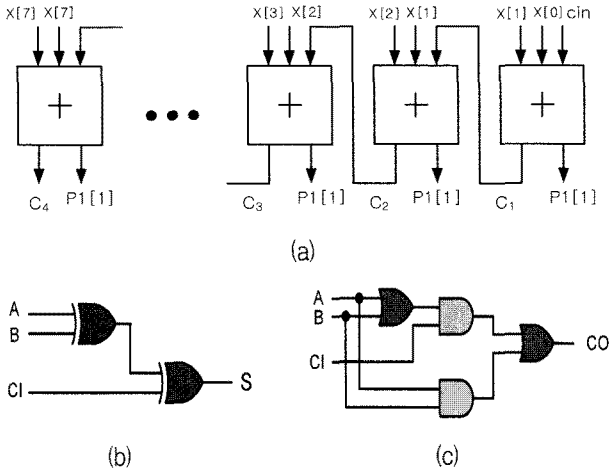

그림 4. 그림 3의 연산을 위한 리플 캐리 덧셈 회로
Fig. 4. Ripple carry addition circuit for operation of fig. 3.

표 1. 그림 2에서 각 P연산의 지연 시간과 최대지연 시간

Table 1. Delay time and maximum delay time of each P operation in fig. 2.

	지연시간(전가산기 지연 개수)
$P1$	9
$P2$	16
$P3$	11
	파이프라인
$P4$	13
$P5$	13
	파이프라인
$P6$	20
최대 지연시간	$P1+P2=25$

기 구조와 캐리 리플이 발생하는 경로를 보여주고 있다. 그림 4-(b)는 입력 3 비트의 합을 구하는 회로이고 그림 4-(c)는 캐리 발생 회로이다. 4-(c)에서 A, B는 동시에 입력되는 변수이므로 캐리전파속도는 and 게이트 1개 or 게이트 1개만큼 시간이 소요된다. 그림 4-(a)의 각 셀에는 그림 4-(b), (c)회로로 구성된다. 전가산기로 그림 2 필터를 설계할 때 최대지연 시간을 계산하기 위해 각 P의 연산시간을 표 1에 나타내었다.

Ⅲ. 제안한 부분 항 덧셈 방법

Ⅱ장의 그림 2에서 지연시간에 영향을 미치는 가장 큰 요소는 각 덧셈의 캐리 리플이다. 따라서 이러한 캐리 리플을 생기지 않도록 하게 하면 필터의 연산속도는 빨라지게 된다. 캐리리플이 생기지 않도록 하기위해서 D플리플롭과 만나는 덧셈까지 부분 항을 2줄씩 유지하며 전개하면 된다.

제안한 방법을 설명 설명하기 위해 그림 2 필터의 부분항 덧셈 과정을 예로 들도록 한다. 그림 2에서 부분항이 연산되는 부분을 부분 항 덧셈 부분 그림 6-(a), 지연 소자와 만나는 최종 덧셈 부분 그림 6-(b)로 나누었다.

그림 2의 $P1$에서는 $x+x\ll 1$이 더해져서 출력 되어야 한다. 그런데 이 때 두 벡터를 더하지 않고 다음 덧셈으로 넘기고 다음 $P2$를 구하기 위해 $P1\ll 5-P1$이 연산되고 $P3$는 $P1\ll 4-x$가 연산되어야 한다. $P1$는 2줄이므로 그림 3와 같이 $P2$는 4줄이 더해져야 된다. 또한 그림 4과 같이 $P3$는 3줄이 더해져야 된다. 이 때 그림 3과 같이 Wallce_tree 알고

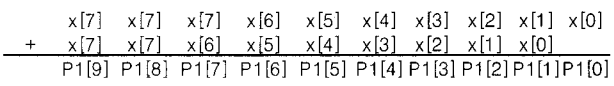

그림 3. $P2$가 2줄이 될 때까지 연산되는 과정

Fig. 3. Computed process until P2 has 2 matrices.

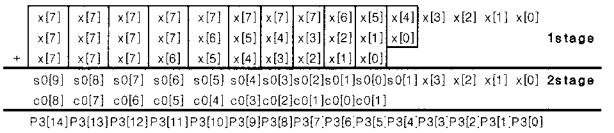

그림 4. $P3$가 2줄이 될 때까지 연산되는 과정

Fig. 4. Computed process until P3 has 2 matrices.

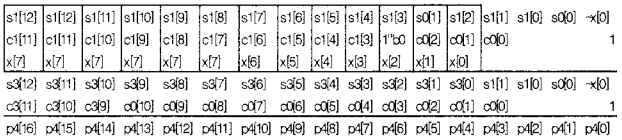

그림 5. $P4$가 2줄이 될 때가지 연산되는 과정

Fig. 5. Computed process until P4 has 2 matrices.

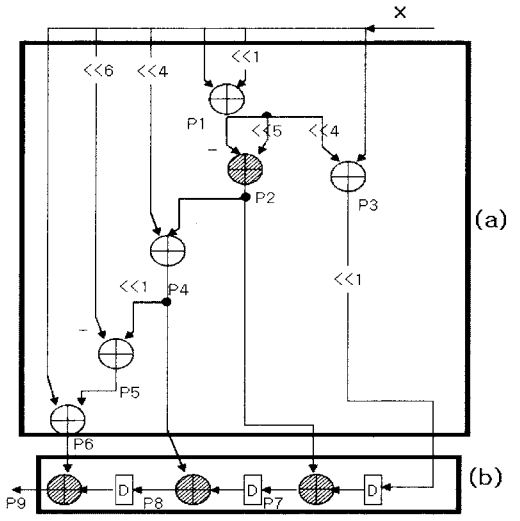

그림 6. 부분 항 4줄이 발생하는 경우

Fig. 6. The case that subexpression generates 4 matrices.

리즘[8]을 이용하여 네모 박스로 변수들을 묶어 전가산기를 이용하여 2번 전개하면 그림 3과 같이 $P2$, $P3$가 2줄이 된다. 그림 3과 같은 연산과정에서는 캐리 리플이 발생하지 않고 단지 전가산기 2개 연산시간만 필요하다. 이 과정을 수행 후 $P2$는 다시 2줄이 되게 된다. $P2$ 값이 계산된 후 $P4$ 값을 계산하기 위해서 $P2+x\ll 4$를 연산해야 한다. $P4$를 2줄로 만들어 주기 위해서는 $P2$가 2줄 x는 1줄 즉 3줄이므로 그림 5와 같이 세로로 3개씩 묶어 더하면 된다. 이 때 지연시간은 1개의 전가산기 계산시간만 필요하다. 이러한 방

표 2. 제안한 방법을 이용한 P 지연시간과 최대지연 시간
Table 2. Delay time and maximum delay time of each P operation using proposed method.

	지연시간(전가산기 지연 개수)
$P1$	2
$P2$	3
$P3$	2
$P4$	2
$P5$	2
$P6$	3
최대 지연시간	$P1+P2+P4+P5+P6=12$

식으로 최종 덧셈영역 그림 6-(b)까지 2줄을 유지하면서 부분 항을 연산한 후 D 플리플롭과 만나는 최종 덧셈 영역 그림 6-(b)($P3, P7, P8, P9$)에서 리플 캐리 덧셈기를 사용하면 $P3, P7, P8, P9$에서만 캐리 리플이 발생하므로 부분 항 덧셈 부분 그림 6-(a)에서는 리플이 발생하지 않으므로 필터의 고속 연산 가능하다.

부분 항 2줄을 만들기 위해서 그림 3과 같이 2번 전개가 필요한 부분은 그림 6에서 빗금으로 표시한 부분과 같이 부분 항 2줄짜리가 서로 만나는 곳과 최종 지연소자에서 부분 항이 만나는 경우이다. 이 경우 4줄이 연산되어야하므로 그림 3과 같이 2번의 연산이 필요하다. 따라서 전가산기 셀이 2배 더 많아지게 되지만 고속 연산을 위해 파이프라인을 한 오버헤드보다 작다.

Ⅷ. 실험결과

표 3은 Ⅱ절의 그림 2와 같은 FIR필터의 입력이 16비트일 때 리플 덧셈기를 이용하여 설계한 결과와 그림 2와 같이 점선대로 파이프라인하여 설계하였을 때와 제안한 방법을 이용하여 설계하였을 때 하이닉스/매그너칩 0.18공정 라이브러리를 이용하여 합성한 결과를 보여주고 있다. 표 4는 표 3과 같은 방법으로 입력을 16비트로 확장하였을 때 결과이다. 표 3에서 파이프라인 된 필터보다 속도가 57.9% 빠르고 면적은 53.2% 더 작음을 알 수 있다. 하지만 최대 동적파워는 파이프라인 된 필터보다 16%정도 큼을 알 수 있다.

표 3. 시뮬레이션 결과
Table 3. Simulation results.

비트	구분	면적(cell)	최대 동적 파워(mW)	속도(ns)
16	파이프라인된 필터	18639.148	11.63	12.1
	제안한 부분 항 덧셈 방식 필터	8731.84	13.46	5.1

Ⅸ. 결 론

계수가 고정되어 있는 경우 CSE 알고리즘을 이용하여 필터를 설계하는 것이 효율적이다. 하지만 CSE는 부분 항을 더할 때 발생하는 캐리 리플로 인하여 지연시간이 길다는 단점이 있었다. 이러한 단점을 해결하기 위해 고속 덧셈기를 이용하거나 파이프라인을 하였다. 본 논문에서는 부분 항 2줄로 유지하면서 캐리 리플을 피하였다. 따라서 최종 지연소자 연결된 덧셈기에 도달할 때까지 부분 항을 더하는데 부분 항이 4줄 일때는 전가산기 2개의 개산시간 부분 항이 3줄일 때는 전가산기 1개의 연산시간만을 필요로 하므로 필터의 고속 연산가능하게 하였다. 제안한 방법은 더해질 부분 항의 비트수가 클 때 즉 필터 입력 x의 입력 비트수가 클 때 더욱 효율적임을 예측할 수 있다.

참 고 문 헌

[1] R. Hartley, "Optimization of canonic signed digit multipliers for filter design," in Proc. IEEE int. Symp. Circuits and Systems, Singapore, June 1991, pp. 1992-1995.

[2] M. Potkonjak et al., "Multiple constant multiplication: Efficient and versatile freamework and algorithms for exploring common subexpression elimination," *IEEE Trans. Computer-Aided Design*, vol. 15, no. 2, pp. 151-165, Feb. 1996.

[3] R. Hartley, "Subexpression sharing in filters using canonic signed digit multipliers," IEEE int. Symp. Circuits and Syst. II, vol. 43, Oct. 1996.

[4] M. Mehendale, S. D. Sherlekar, and G. Vekantesh, "Systhesis of multiplierless FIR filters with minimum number of additions, " *in Proc. 1995, IEEE/ACMInt. Conf. computer-Aided Design*, Los Alamitos, CA, 1995, pp. 1595, pp. 668-671.

[5] Marcos Martinez-Peiro, Eduardo I. Boemo, and Lars Wanhammar, "Design of High-Speed Multiplierless Filter Using a Nonrecursive Signed Common Subexpression Algorithm, " *IEEE Transactions on circuits and Systems II,*

2002, vol. 49, pp. 196-203.
[6] A. Dempster and M. D. Macleod, "Constant integer multiplication using minimum adders," *Proc. Inst. Elec. Eng. Circuits and systems*, vol. 141, no. 5, pp.407-413, Oct. 1994.
[7] A. Dempster and M. D. Macleod, "Use of minimum adder multiplier blocks in FIR digital filters," *IEEE Trans. Circuits Syst. II*, vol. 42, pp569-557, Sept. 1995.
[8] C. S. Wallace, "Suggestion for a fast multiplier," *IEEE Transactions on Electronic Computers*, vol. EC-13, pp. 14-17, 1964.

저 자 소 개

김 용 은(학생회원)
2005년 전북대학교 전자공학과 학사 졸업
2007년 전북대학교 정보통신 공학과 석사 졸업
2007년~현재 전북대학교 전자 정보공학부 박사
<주관심분야 : 통신, 신호처리, 반도체>