재귀 구조에 기반한 FIR 디지털 필터의 설계

Design of a Recursive Structure-based FIR Digital Filter

이 재 진*, David Tien**, 송 기 용*

Jae-Jin Lee*, David Tien**, Gi-Yong Song*

요 약

본 논문은 Top-Down 설계방식에서 동작 레블과 로직 레블에서 동일한 구조를 가지는 새로운 FIR 디지털 필터의 설계 방법론을 제안한다. 제안되는 설계 방법론은 승산이 컨벌루션-carrying의 형태로 표현되어지며, 이것은 결과적으로 로직 레블의 승산을 동작 레벨의 컨벌루션과 같은 구조로 구현할 수 있다는 연구에 기반한 것이다. 재귀 구조에 기반한 FIR 디지털 필터의 예를 보이기 위해 본 논문에서는 L 개의 탭을 가지는 전치형과 시스톨릭 FIR 필터의 구현에 대해 기술한다. 제안된 FIR 디지털 필터는 하나의 컨벌루션 구조의 재귀적 사용과 2개의 1-비트 입/출력 포트만으로 구성될 수 있으므로 매우 규칙적이고 간결한 구조를 가진다.

Abstract

This paper proposes a new digital filter implementation which adopts an identical structure at both behavioral and logic level in top-down design. This methodology is based on the observation that multiplication is a form of convolution and carrying, and therefore multiplication is implemented with the same structure as that of a convolution in a recursive manner at the logic level. ·

In order to demonstrate a recursive structure-based FIR digital filter, we select L-tap transposed and systolic FIR filters, and implement them to have a single structure. The proposed filter design becomes regular and modular because of the recursive adoption of a single structure for convolutions, and is very compact in that it needs only two 1-bit I/O ports in addition to significant improvement on hardware complexity without time penalty on the output sequence.

Key words : FIR filter, recursive-structure, convoluation, carrying

I. Introduction

A new method which gives a different way to formulate structure for realizing digital filters has been introduced by Pled and Liu [1]. This new formulation is particularly focused on not another rearrangement of multipliers, adders, and registers, but a new description where the fundamental operation of convolution and multiplication are mixed-hence distributed arithmetic [2]. The first application of these ideas to signal processing was introduced in [3]. The most obvious practical form of realization using a table-look up of stored precalculated partial products was presented in [4-5].

This paper proposes a new digital filter with recursive structure. This is based on the ideas that multiplication is a form of convolution-carrying, and therefore a convolution is regarded again as a bit-level convolution and carrying instead of multiplication and summation.

To perform the digital filtering operation in real

*Chungbuk National University, Research Institute for Computer and Information Communication

**Charles Sturt University Bathurst, NSW, 2975, Australia

접수 일자 : 2004. 3. 9 수정 완료 : 2004. 4. 12

논문 번호 : 2004-1-9

time, high-speed computing hardware is often necessary. While systems organized for processing bit-parallel data have been used extensively in computer hardware, the bit-serial approach to digital filter realization is efficient in digital signal processing applications where data bits are available in bit string form such as telephone voice and data signals.

To present a recursive structure-based bit-level FIR digital filter, we select L-tap transposed and systolic FIR filters as representatives, and implement them to have a single structure.

The two proposed bit-level FIR filters are modeled and simulated in RT level using VHDL, then synthesized using Synopsys design compiler based on Hynix 0.35㎛ cell library. Compared to the conventional transposed or systolic FIR filter with word-level data flow, the proposed bit-level implementation brings about significant improvement on hardware complexity without time penalty on the output sequence, and becomes very compact in that it needs only two 1-bit ports; one port for input and one port for output.

II. FIR Digital Filters

Digital filters are typically used to modify or alter the attributes of a signal in the time or frequency domain. An FIR with constant coefficients is a linear time-invariant (LTI) digital filter. The output sequence, $y(n)$, of an FIR of order or length L, to an input sequence, $x(n)$, is given by the finite version of convolution sum as follows [6]:

$$y(n)=x(n)*f(n)=\sum_{k=0}^{L-1}x(k)f(n-k) \qquad (1)$$

where $f(0)\neq 0$ through $f(L-1)\neq 0$ are the filter's L coefficients. For LTI systems it is sometimes more convenient to express (1) in the z-domain with

$$Y(z)=F(z)X(z) \qquad (2)$$

where $F(z)$ is a the FIR's transfer function defined in the z-domain by

$$F(z)=\sum_{k=0}^{L-1}f(k)z^{-k} \qquad (3)$$

A. Transposed FIR Filter

The Lth order Transposed FIR filter is graphically interpreted in Fig. 1 for L=4.

In this design. $x(n)$ and $y(n)$ move word-serially, so wide data registers, N-bit for $x(n)$ and N+M+K-bit for $y(n)$ each, are needed to accommodate the data of varying length, where input sequence, $x(n)$, and coefficient, $f(i)$, is N and M-bit wide each. These data registers are not shown in Fig. 1 for simplicity. For a filter length L, K guard bits for arithmetic operation must be provided, where K=log$_2$$L$. For a filter of length 4 with 4-bit unsigned input and coefficient, the adder width must be 4+4+log$_2$4=10.

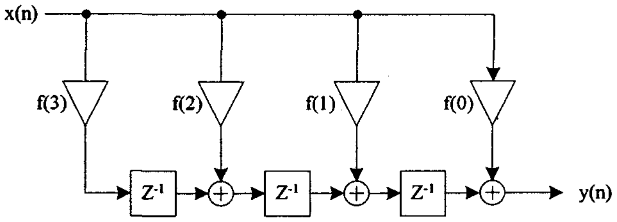

Fig. 1. Transposed FIR Filter

B. Systolic FIR filter

The locally recursive algorithms for the FIR filters was developed by Kung [7], and its systolic array [7-8] for FIR filters is shown in Fig. 2(a) for L=4. The systolic array consists of identical locally-connected processing elements, or cells, as depicted in Fig. 2.

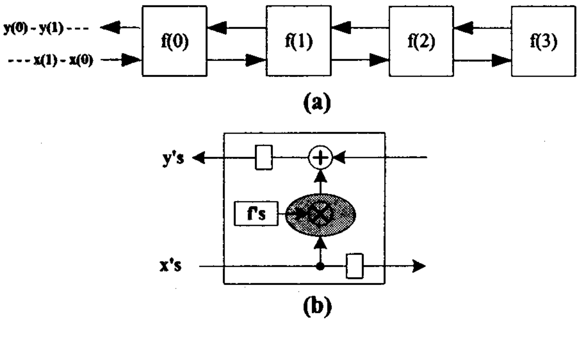

Fig. 2. (a) Systolic FIR Filter (b) Cell

In this design, $x(n)$ and $y(n)$ move between cells word-serially, so wide data registers, N-bit for $x(n)$ and N+M+K-bit for $y(n)$ each, are also needed to accommodate the data of varying length as for the transposed FIR filter. Each cell contains a multiplier and an adder as shown in Fig. 2(b).

Ⅲ. Multiplication as Convolution -Carrying

Consider the multiplication of two N-bit unsigned integers X and F:

$$X=x_{n-1}x_{n-2}\cdot\cdot\cdot x_0=\sum_{i=0}^{N-1}x_i2^i$$

$$F=f_{n-1}f_{n-2}\cdot\cdot\cdot f_0=\sum_{j=0}^{N-1}f_j2^j \tag{4}$$

$$P=XF=\sum_{i=0}^{N-1}x_i2^i\sum_{j=0}^{N-1}f_j2^j=\sum_{i=0}\sum_{j=0}x_if_j2^{i+j}$$

Now change of coordinates : $k=j+i$

$$P=\sum_{k=0}^{2N-1}\sum_{i=0}^{N-1}x_if_{k-i}2^k \tag{5}$$

According to the above relation, (5), the product $P=XF$ could be calculated from the convolution of the X and F. This relation is true for any base and has been suggested for high-speed multiplication of large integers [9].

In evaluating (4), there exist two sets of indices. Consider the following example for N=2

	f_1	f_0
	x_1	x_0
	f_1x_0	f_0x_0
f_1x_1	f_0x_1	
p_2	p_1	p_0

Evaluating (4) first over j generates the partial products ($2f_1x_0+f_0x_0$) and ($2f_1x_1+f_0x_1$) which are shifted (multiplied by 2) and added in the second sum over i. Alternatively, expression (5) is summed over i first which is convolution, then over k. This gives partial products as (f_0x_0), ($f_1x_0+f_0x_1$), and (f_1x_1), which are the columns in the partial product array of the example. The sum over k converts the convolution into a binary form. In other words, multiplication can be interpreted as convolution plus carrying. If it were not for the carries, the sum over k would not be necessary.

The direct implementation of the binary convolution of two bit streams [2] involving a concatenation of zeros, segmenting or sectioning for the process of carries by a overlap and add procedure would result in even larger combinational logic than an array multiplier.

In this paper, the observation that a multiplication can also be interpreted in terms of convolution-carrying is examined from the aspect of implementation, and as a result, the multiplier is implemented succinctly as a convolution-carrying logic with a series of a combination of 2-input AND gate and 1-bit serial adder in a structure adopted recursively from the upper-level.

The multiplier of the transposed form in Fig. 1, for example, will be implemented with a series of a combination of 2-input AND gate and 1-bit serial adder as shown in Fig. 3, adopting the original structure for convolution recursively. The details will be explained in the following section.

In each type of implementation, the fundamental structure for convolution will be adopted recursively, so the entire structure becomes regular and modular.

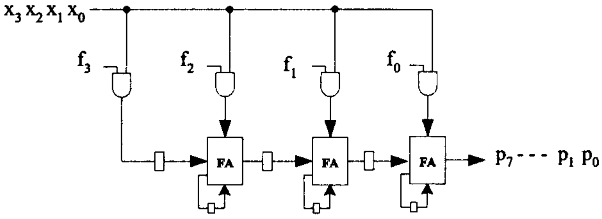

Fig. 3. Convolution-Carrying Multiplier

To analyze the performance of three 16x16 multipliers; array multiplier, convolution-carrying multiplier in Fig. 3, and systolic multiplier proposed in [10] which actually is another type of convolution-carrying multiplier based on systolic structure, each one is designed with VHDL [11-12] and implemented on XCS40 with approximate 784 CLBs and 224 IOBs [13]. Implementation reports of each design are listed in Table 1.

Table 1. Implementation reports

16x16 Mutliplier	Array	Convolution-Carrying	Systolic
CLB	208	24	33
Avg. connection delay(ns)	4.499	1.464	1.336
Worst clock cycle (ns)	19.772	4.388	3.708

Ⅳ. Recursive Structure-based Bit-level FIR filters

A. Design of a Transposed FIR Filter

For coefficient sequence of four M-bit integers, $f(i)$, the proposed implementation of a recursive structure-based bit-level transposed FIR filter is shown in Fig. 4. The multipliers in Fig. 4(a) is implemented again as convolution and carrying as shown in Fig. 4(b), and this brings recursive structure on the implementation of a transposed FIR filter at logic-level.

Note that the N-bit input sequence, $x(n)$, enters the array from left in bit-serial manner and output sequence, $y(n)$, emerges from right without intermission in a bit-serial manner as well. New types of delays are adopted between FAs to synchronize the bit data flow to guarantee the generation and summation of the product terms according to recurrence relation of the convolution sum; delay for $y(n)$ is $N+M+K$ unit delay where K is guard bits. The $N+M+K$ unit delay on the path for output sequence flow is implemented with a shift register of the same width as shown in Fig. 4(a). The bit-level transposed FIR filter in Fig. 4 needs only two 1-bit ports; one port for input and one port for output.

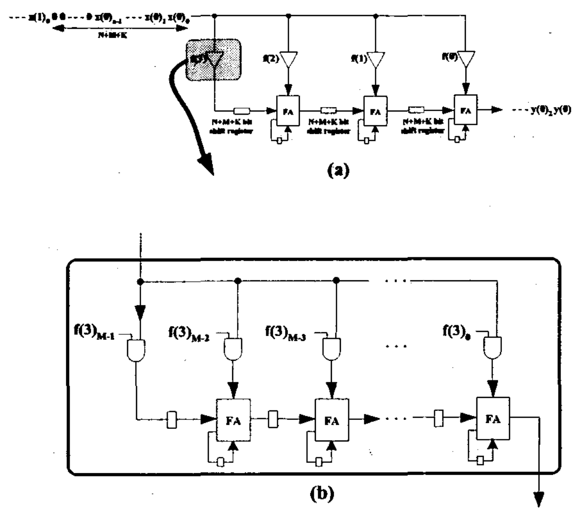

Fig. 4. (a) Bit-level transposed FIR filter
(b) Convolution-Carrying multiplier

B. Design of a Systolic FIR Filter

The implementation of a recursive structure-based bit-level systolic FIR filter is shown in Fig. 5. The multiplier in the systolic array in Fig. 2(b) is implemented recursively with the same structure as original systolic array as shown in Fig. 5(b).

Note that the N-bit input sequence, $x(n)$, enters the array through the left cell in bit-serial manner and output sequence, $y(n)$, emerges from the left cell in a bit-serial manner as well.

The input data should enter the array at every other unit delay to synchronize data flow, resulting in degeneration on the throughput of bit-level systolic FIR filter. This degeneration can be avoided by having the input data interleaved with two input sequence $X1$ and $X2$. The scheme of interleaving two input sequence enables bit-level systolic FIR filter to preserve the same efficiency as word-level systolic FIR filter. The proposed a recursive structure-based bit-level systolic FIR filter is also a super-systolic array [10] for FIR filter, for the cell of systolic array itself is organized as another systolic array, that is, super-systolic array. The super-systolic array in [10] is with a bit-serial systolic multiplier in each cell based on mixed type of data flow in which data are passed on bit-level in some part and word-level in other parts.

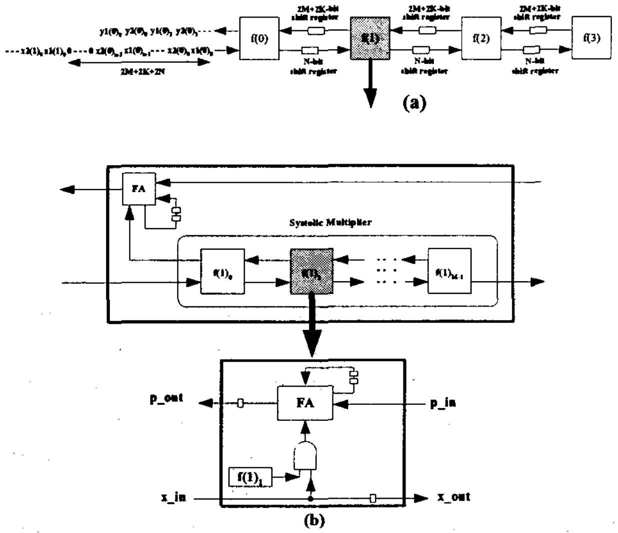

Fig. 5. (a) Bit-level systolic FIR filter
(b) Systolic multiplier

Ⅴ. Syntheis and Performance Evaluation

To verify the correctness of the proposed recursive structure-based bit-level FIR filters, a simulation is performed on each of its VHDL code. For two data streams $f(i)$'s -- F, 7, A and C -- and $x(i)$'s -- 5, 9

B and 3 -- in hexadeciaml each, the simulation results for recursuve structure-based bit-level transposed FIR filter and recursuve structure-based bit-level systolic FIR filter are shown in Fig. 6 and Fig. 7 respectively.

After simulation, each one is synthesized using Synopsys design compiler [12] based on Hynix 0.35㎛ cell library. The input sequence, $x(n)$, and coefficients, $f(i)$, of 16-bit each, and filter length L of 4 are assumed. Table 2 shows synthesis reports of conventional word-level transposed FIR filter (WTF) and a recursive structure-based bit-level transposed FIR filter (RSBTF). Table 3 shows synthesis reports of conventional word-level systolic FIR filter (WSF) and a recursive structure-based bit-level systolic FIR filter (RSBSF).

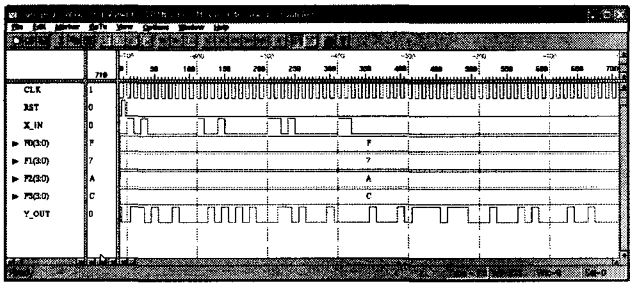

Fig. 6. Simulation wave for resursive structure-based bit-level transposed FIR filter

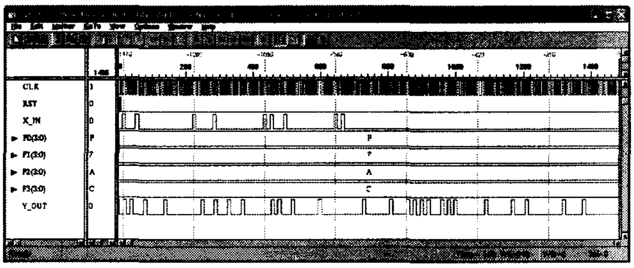

Fig. 7. Simulation wave for resursive structure-based bit-level systolic FIR filter

Table 2. Synthesis reports for transposed FIR filter

FIR Filter	WTF	RSBTF
Combinatorial area (2-input NAND)	8884.5	757.5
Noncombinatorial area (2-input NAND)	714	1645
Net interconnect area (2-input NAND)	17.317	6.7
Total area (2-input NAND)	9615.817	2409.2
Critical path delay (ns)	35.81	0.99

Table 3. Synthesis reports for systolic FIR filter

FIR Filter	WSF	RSBSF
Combinatorial area (2-input NAND)	8887.5	850.5
Noncombinatorial area (2-input NAND)	1050	2947
Net interconnect area (2-input NAND)	17.86	9.49
Total area (2-input NAND)	9955.36	3806.99
Critical path delay (ns)	35.41	0.99

Ⅵ. Conclusions

Based on the observation that multiplication is a form of a convolution and carrying, this paper proposes a new digital filter implementation methodology which adopts an identical structure at both behavioral and logic level in top-down design, and demonstrates implementations of L-tap recursive structure-based transposed FIR filter and systolic FIR filter.

Results we got through synthesizing the recursive structure-based bit-level transposed FIR filter and systolic FIR filters show that the proposed implementation is very compact in that it needs only two I/O ports in addition to significant improvement on hardware complexity without time penalty on the output sequence.

Acknowledgements

This work was done as a part of Information & Communication fundamental Technology Research Program supported by Ministry of Information & Communication in republic of Korea

References

[1] A.Peled and Bi.Liu "A new hardware realization of digital filters," *IEEE Trans. Acoust., Speech, Signal Proc.*, vol. ASSP-22, pp. 456-462, 1974.

[2] C.Sidney Burrus, "Digital Filter Structures Described by Distributed Arithmetic," *IEEE Trans. On Circuits and Systems*, vol. CAS-24, pp 674-680, 1977

[3] E.Anderson, "A digital filter implemented in parallel form," *Symp. On Digital Filtering*, Imlerial College, London, 1971

[4] A.Croisier, D.J.Esteban, M.E. Levilion, and V.Riso, "Digital filter for PCM encoded signals," *U.S. Patent 3 777130*, 1973

[5] A.Peled and B.Liu "A new hardware realization of digital filters," *IEEE Arden House Workshop on Digital Signal Processing*, 1974

[6] U.Meyer-Baese, *Digital Signal Processing with Field Programmable Gate Arrays*, Springer, 2001.

[7] S.Y.Kung, *VLSI Array Processors*, Prentice Hall, 1988.

[8] H.T.Kung, "Why Systolic Architectures?," *Computer* Vol.15, No.1, pp.37-46, January 1982.

[9] A.Schonhage and V.Strasser, "Fast multiplication of large numbers," *Computer*, vol. 7, pp281-292, 1971

[10] G.Y.Song and J.J.Lee, "Implementation of the Super Systolic Array for Convolution," *ASP-DAC 2003*, pp. 491-494, Jan. 2003.

[11] Y.C.Hsu, K.F.Tsai, J.T.Liu and E.S.Lin, *VHDL Modeling for Digital Design Synthesis*, Kluwer Academic Publishers, 1995.

[12] K.C. Chang, *Digital Systems Design with VHDL and Synthesis*, IEEE Computer Society Press, 1999.

[13] *The Programmable Logic Data Book*. Xilinx, Inc. 1995.

Jae-Jin Lee
received the B.S. and M.S. degrees in computer engineering from Chungbuk National University in 2000 and 2003, respectively.
He is currently working towards Ph. D. degree on computer engineering.
His research interests include the areas of VLSI design, design automation and computer architecture.

David Tien
received M.S. and Ph.D. degrees in Computer Science, Pure Mathematics and Electrical Engineering from Harbin, Chinese Science Academy, the Ohio State University, USA and the University of Sydney, Australia, respectively.
Prior to joining the School of Information Technology, Charles Sturt University in 2000, he had 20 years' experience in research and teaching at the University of Sydney, Ohio State University and Singapore.
Currently, his major research interests are artificial intelligence, image and signal processing, telecommunication, education theory, and biomedical engineering. During the past 15 years, David served as the Secretary, Treasurer and Chairman of the IEEE Singapore Section, MDC Chairman of Tele -Communication Society, Region 10, and is currently the Treasurer of NSW Section. David also serves as a member of the Charles Sturt University Senate.

Gi-Yong Song
received the B.S. and M.S. degrees in electronic engineering from Seoul National University in 1978 and 1980, respectively and the Ph.D. degree from University of Louisiana in 1995.

He is currently a professor in the School of Electrical and Computer Engineering.
His research interests include the areas of VLSI design, design automation, and computer architecture.