NOVEL PROGRAMMABLE FIR FILTER BASED ON HIGHER RADIX RECODING FOR LOW-POWER AND HIGH-PERFORMANCE APPLICATIONS

Hsiao-Yun Chen and Shyh-Jye Jou

Department of Electronics Engineering, National Chiao Tung University
1001 Ta-Hsueh Road, Hsinchu, Taiwan 30050, R.O.C.

ABSTRACT

This paper proposes the novel programmable digital finite impulse response (FIR) filters for low-power and high-performance applications. The architecture is based on the higher radix recoding scheme which specifically targets the reduction of power consumption by decreasing partial products and precomputation sharing in each partial product. We extend higher radix recoding scheme with secondary radix recoding to further improve performance by reducing the propagation delay of precomputing odd multiples of the multiplicand. A 10-tap programmable FIR filter based on proposed schemes was design in TSMC 0.18-μm technology. The performance and power consumption of the proposed schemes can improve about 40.9~65.5% and 28.3~50.2% over existing designs.

Index Terms—programmable, FIR filter, high radix, Booth recoding, precomputation sharing

1. INTRODUCTION

Due to the rapid growth in communication and multimedia applications, the low-power and high-performance digital signal processing (DSP) systems are demanded progressively. Finite impulse response (FIR) filter is one of the most widely used in DSP. High performance programmable FIR filters are frequently used to perform adaptive pulse shaping and signal equalization on the received data in real time. Besides, complexity reduction of FIR filter implementations has been of particular interest as lower computational complexity leads to high performance as well as low power design. Canonical signed digit (CSD) and common subexpression elimination are widely used in FIR filter designs with low complexity applications. However, those works that depend heavily on the coding and sharing of the FIR coefficients is difficult to accomplish in real time and lead to limit the designs of FIR filter with fixed coefficients.

This paper proposes a novel programmable FIR filter based on higher radix recoding for high performance and low power applications. We employ the higher radix recoding [1-3] to reduce the number of partial products and reduce power consumption by precomputation sharing in each partial product generation. However, the propagation delay of precomputing multiples is increasing by increasing higher radices. We can extend recoding methodology by using secondary radix recoding schemes [3] to reduce the number and propagation delay of odd multiples requirement for very high radix recoding.

The remainder of this paper is organized as follows. Section 2 presents the algorithm reformulations of FIR filter with higher radix recoding and secondary radix recoding. Section 3 describes the proposed programmable FIR filters based on higher radix recoding and secondary radix recoding. Section 4 shows the design results and comparisons of programmable FIR filters. Finally, Section 5 is the conclusions.

This work was supported by the MediaTek Inc. and the National Science Council of Taiwan, R.O.C. under Grant *NSC94-2220-E-009-035*.

2. ALGORITHM REFORMULATIONS

The reduction in partial products obtained by the well-known modified Booth multiplier recoding is offset by the need to precompute odd multiples of the multiplicand as inputs to each partial product generation. We propose to use higher radix recoding to decrease the number of partial products and reduce power consumption by precomputation sharing in each partial product generation. However, increasing radices to reduce the number of partial products will increase the propagation delay of precomputing multiples of the multiplicand. We propose to extend recoding methodology by secondary radix recoding schemes to reduce the number of odd multiples required for very high radix recoding.

2.1. Higher Radix Recoding

A $w \times w-$bit multiplication $X \cdot C$ in its simplest form is implemented by the generation of w partial products. We denote C and X in binary number representation.

$$X = \sum_{i=0}^{w-1} x[i] \cdot 2^i \quad (1)$$

$$C = \sum_{i=0}^{w-1} c[i] \cdot 2^i \quad (2)$$

The product computation is based on the sum:

$$X \cdot C = \sum_{i=0}^{w-1} X \cdot c[i] \cdot 2^i \quad (3)$$

with the partial product, $P_i = X \cdot c[i] \cdot 2^i$. The factor 2^i is realized by a shift of the multiplicand X, and the factor $c[i] \in \{1, 0\}$ is employed by a partial product generation to select either $X \cdot 2^i$ or zero as i-th partial product in the w-term sum. The main approach to decrease the number of

partial products in (3) is representing the multiplier C in a higher radix. The multiplier C is recoded to a $p=\lceil (w+1)/r \rceil$ - digit minimally redundant (Booth) radix representation

$$C = \sum_{i=0}^{p-1} d_i \cdot (2^r)^i \quad (4)$$

in a higher radix $\beta=2^r$, where $d_i \in \{-2^{r-1},...,2^{r-1}\}$. The representation of the product is

$$X \cdot C = \sum_{i=0}^{p-1} X \cdot d_i \cdot (2^r)^i. \quad (5)$$

Each partial product has expanded factorization

$$X \cdot d_i \cdot (2^r)^i = X \cdot [(-1)^s \cdot \alpha_i \cdot 2^e] \cdot 2^{ir} = (-1)^s \cdot \{\alpha_i \cdot X\} \cdot 2^{e+ir} \quad (6)$$

where $\alpha_i \in \{0,1,3,...,2^{r-1}-1\}$. The 2^{r-2} odd multiples of the multiplicand $\{\alpha_i \cdot X\}$ serve as primitive digits set to be conditionally selected, complemented by $(-1)^s$, and shifted by 2^{e+ir} to form the partial products to be accumulated.

2.2. Secondary Radix Recoding

The radices greater than eight will increase the number of required odd multiples and therefore increase the complexity and delay of the carry propagate adder to pre-compute the odd multiples. We extend higher radix (radix $\beta=2^r, r \geq 5$) recoding by the secondary radix recoding to express the sum of partial products and reduce the number of required odd multiples. As show in Eq. (4), each Booth digit value $-2^{r-1} \leq d_i \leq 2^{r-1}$ is recoded to n-digit value ($2 \leq n \leq 4$) in secondary radix recoding scheme. For example n=2 and the secondary radix (modulus) λ, we recoded the digit d_i to be

$$d_i = d_{1,i} \cdot \lambda + d_{0,i} \quad (7)$$

where digits $d_{1,i}$ and $d_{0,i}$ are chosen from a balanced complete residue system modulo λ forming the secondary radix digit set D_λ. The residue digit sets of the form

$$D_\lambda = \{0, \pm d_1, \pm d_2, ..., \pm d_{(\lambda-1)/2}\} \quad (8)$$

are termed balanced complete residue systems when every integer m $(0 \leq m \leq \lambda - 1)$ is congruent to one member of D_λ. Since complements and shifts can be employed to increase the range of digit values, all nonzero digits of D_λ should be of the form $\pm\delta 2^i$, where δ is a member of $\{0,1\}$, $\{0,1,3\}$ or $\{0,1,3,5\}$. For n=2, Eq. (4) can be expressed as

$$C = \lambda \cdot \sum_{i=0}^{p-1} d_{1,i} \cdot (2^r)^i + \sum_{i=0}^{p-1} d_{0,i} \cdot (2^r)^i \quad (9)$$

where $d_{1,i}, d_{0,i} \in D_\lambda$. Eq. (5) can be expressed as

$$X \cdot C = \lambda \cdot \sum_{i=0}^{p-1} X \cdot d_{1,i} \cdot (2^r)^i + \sum_{i=0}^{p-1} X \cdot d_{0,i} \cdot (2^r)^i. \quad (10)$$

The value of λ should be an odd and prime for the radix $\beta=2^r$ and must be of the order of an n-th root of β. The λ is better to be implemented by 2 or 3 nonzero bits to allow multiplication by λ to be a shift and add / subtract. The n-digit secondary radix values must include all integers in the high radix Booth digit range $[-2^{r-1}, 2^{r-1}]$.

2.3. Algorithm Reformulation for FIR Filter

Considering an N-tap FIR filter with input sequence X_n, output sequence Y_n, and coefficients C_i, without loss of generality we can express the FIR reformulation of high radix $\beta=2^r$ recoding as

$$\begin{aligned} Y_n &= \sum_{i=0}^{N-1} C_i \cdot X_{n-i} = \sum_{i=0}^{N-1} \left[\sum_{j=0}^{p-1} X_{n-i} \cdot d_{i,j} \cdot (2^r)^j \right] \\ &= \sum_{i=0}^{N-1} \left[\sum_{j=0}^{p-1} (-1)^{s_{i,j}} \cdot \{\alpha_{i,j} \cdot X_{n-i}\} \cdot 2^{jr+e_{i,j}} \right] \end{aligned} \quad (11)$$

where $d_{i,j} \in \{-2^{r-1},...,2^{r-1}\}$ and $\alpha_{i,j} \in \{0,1,3,...,2^{r-1}-1\}$. We can extend the FIR reformulation of high radix $\beta=2^r$ with secondary radix recoding as

$$\begin{aligned} Y_n &= \sum_{i=0}^{N-1} C_i \cdot X_{n-i} \\ &= \sum_{i=0}^{N-1} \left[\lambda \cdot \sum_{j=0}^{p-1} X_{n-i} \cdot d_{1,i,j} \cdot (2^r)^j + \sum_{i=0}^{p-1} X \cdot d_{0,i,j} \cdot (2^r)^j \right] \end{aligned} \quad (12)$$

where $d_{1,i,j}, d_{0,i,j} \in D_\lambda$. Each partial product can also be expanded with factorization

$$X_{n-i} \cdot d_{\eta,i,j} \cdot (2^r)^i = (-1)^{s_{\eta,i,j}} \cdot \{\delta_{\eta,i,j} \cdot X_{n-i}\} \cdot 2^{jr+e_{\eta,i,j}} \quad (13)$$

where $\delta_{\eta,i,j} = \{1\}$, $\{1,3\}$ or $\{1,3,5\}$ and $\eta = 1$ or 0.

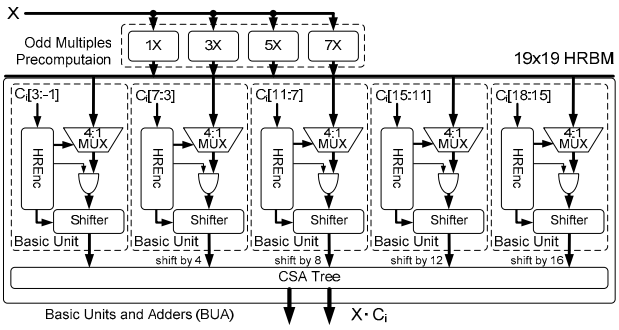

Fig. 1 Parallel architecture of 19×19 – bit radix 16 ($\beta=2^4$) HRBM.

3. PROGRAMMABLE FIR FILTER BASED ON HIGHER RADIX BOOTH RECODING

3.1. Higher Radix Booth Multiplier (HRBM)

Fig. 1 shows the parallel architecture of 19×19 – bit radix-16 $(\beta = 2^4)$ HRBM, as shown in Eq. (5) and Eq. (6). The HRBM is composed of the odd multiples precomputation, basic units, and adders. In the implementation of HRBM, the input X, coefficient C, and output $X \cdot C$ are represented in two's complement format.

The precomputations of odd multiples $\{1\times, 3\times, 5\times, 7\times\}$ for radix-16 HRBM is utilized by complement and shift operations to generate the full set of partial products {-8×, -7×, ..., 7×, 8×} for Booth radix-16 digits. Fig. 2 shows the structure of the odd multiples precomputation and shows the $5\times$ implemented by carry propagate adder (CPA). The basic unit of HRBM is composed of higher radix encoder (HREnc), 4:1 MUX, Shifter, and AND gates. Since HREnc is directly connected to coefficients, it does not lie on the critical path. Table 1 present the radix-16 recoding scheme of HREnc. 4:1 MUX is used to select the odd multiples with control signal *sel*. Shifter is composed of one 4:1 MUX to select the data shifting 0 to 3 bits with the control signal *sf*. AND gates and signal *zero* are used to deal with zero partial product output.

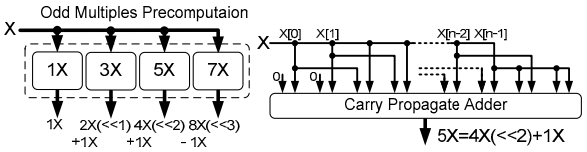

Fig. 2 Odd multiples precomputation structure and precompute $5\times$ architecture.

Table 1 Radix-16 Recoding Scheme of HREnc

Coefficient				4:1MUX	Shifter	AND	Sign		Partial Product	
$C_i[4j+3:4j-1]$				*sel*	*sf*	*zero*	*sig*			
00000		11111		00	00	0	0	1	0	-0
00001	00010	11110	11101	00	00	1	0	1	1	-1
00011	00100	11100	11011	00	01	1	0	1	2	-2
00101	00110	11010	11001	01	00	1	0	1	3	-3
00111	01000	11000	10111	00	10	1	0	1	4	-4
01001	01010	10110	10101	10	00	1	0	1	5	-5
01011	01100	10100	10011	01	10	1	0	1	6	-6
01101	01110	10010	10001	11	00	1	0	1	7	-7
01111		10000		00	11	1	0	1	8	-8

Signal *sig* decides the partial product will be add (*sig*=0) or subtract (*sig*=1). Finally, we employ the carry save adders (CSA) tree to sum the outputs of five basic units.

3.2. HRBM^2 Based on Secondary Radix Recoding

Fig. 3 shows the parallel architecture of $19\times19-$bit radix-32-modulo-7 ($\beta=2^5;\lambda=7$) HRBM^2 based on secondary radix recoding, as shown in Eq. (10). The HRBM^2 is composed of the basic units and adders. In the implementation of HRBM^2, the input X, coefficient C, and output $X\cdot C$ are represented in two's complement format.

The 2-digit values of $D_7=\{0,\pm1,\pm2,\pm4\}$ cover all integers in Booth radix-32 digit range [-16,16]. The digits of D_7 can be realized by a conditional complement $(-1)^s$ and a shift 2^e, e=0~2. A $19\times19-$bit multiplier $X\cdot C$ based on secondary radix recoding to recode coefficient C. Eq. (10) can be expressed as

$$X\cdot C=7\cdot\sum_{i=0}^{3}X\cdot d_{1,i}\cdot(2^5)^i+\sum_{i=0}^{3}X\cdot d_{0,i}\cdot(2^5)^i \qquad (14)$$

where $d_{1,i},d_{0,i}\in D_7$. Using radix-32- modulo-7 recoding the $19\times19-$bit $X\cdot C$ has 4 partial products of each digit expression. Each digit expression can be extended as

$$X\cdot d_{\eta,i}\cdot(2^5)^i=(-1)^{s_{\eta,i}}\cdot\{\delta_{\eta,i}\cdot X\}\cdot 2^{5\cdot i+e_{\eta,i}} \qquad (15)$$

where $\delta_{\eta,i}=\{0,1\}$, $e_{\eta,i}=0\sim2$, and $\eta=1$ or 0.

The basic unit of HRBM^2 is composed of secondary radix encoder (HRSEnc), Shifter1, Shifter0, and AND gates. Since HRSEnc is directly connected to coefficients, it does not on the critical path. Table 2 presents the radix-32-modulo-7 recoding scheme of HRSEnc. Shifter0 is composed of one 4:1 MUX to select the data shifting 0 to 2 bits with the control signal *sf0*. Shifter1 is composed of one 2:1 MUX to select the data shifting 0 to 1 bit with the control signal *sf1*. AND gates and signal *z[1:0]* are used to deal with zero partial product of $\delta_{\eta,i}$. Signal *s[1:0]* decides the partial product will be add (*s[1]* or *s[0]*=0) or subtract(*s[1]* or *s[0]*=1). Finally, we employ the CSA trees to sum the output of each digit expression.

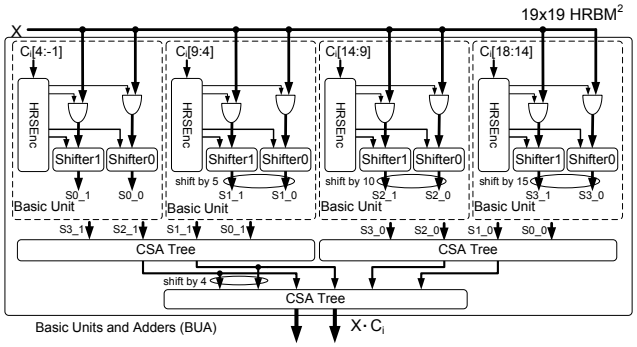

Fig. 3 Parallel architecture of $19\times19-$bit radix-32-modulo-7 ($\beta=2^5;\lambda=7$) HRBM^2.

Table 2 Radix-32-modulo-7 Recoding Scheme of HRSEnc

Coefficient				Shift		AND		Sign		Partial Product			
$C_i[5j+4:5j-1]$				*sf1/sf0*		*z[1:0]*		*s[1:0]*		Radix-32		Modulo-7	
000000		1111111		0	00	0	0	0	1	0	-0	[0,0]	[0,0]
000001	000010	111110	111101	0	00	0	1	0	1	1	-1	[0,1]	[0,-1]
000011	000100	111100	111011	0	01	0	1	0	1	2	-2	[0,2]	[0,-2]
000101	000110	111010	111001	0	10	1	1	0	1	3	-3	[1,-4]	[-1,4]
000111	001000	111000	110111	0	10	0	1	0	1	4	-4	[0,4]	[0,-4]
001001	001010	110110	110101	0	01	1	1	0	1	5	-5	[1,-2]	[-1,-2]
001011	001100	110100	110011	0	10	1	1	0	1	6	-6	[1,-1]	[-1,1]
001101	001110	110010	110001	0	00	1	0	0	1	7	-7	[1,0]	[-1,0]
001111	010000	110000	101111	0	00	1	1	0	1	8	-8	[1,1]	[-1,-1]
010001	010010	101110	101101	0	01	1	1	0	1	9	-9	[1,2]	[-1,-2]
010011	010100	101100	101011	1	10	1	1	0	1	10	-10	[2,-4]	[-2,4]
010101	010110	101010	101001	1	10	1	1	0	1	11	-11	[1,4]	[-1,-4]
010111	011000	101000	100111	1	01	1	1	0	1	12	-12	[2,-2]	[-2,2]
011001	011010	100110	100101	1	00	1	1	0	1	13	-13	[2,-1]	[-2,1]
011011	011100	100100	100011	1	00	1	0	0	1	14	-14	[2,0]	[-2,0]
011101	011110	100010	100001	1	00	1	1	0	1	15	-15	[2,1]	[-2,-1]
011111		100000		1	01	1	1	0	1	16	-16	[2,2]	[-2,-2]

3.3. Programmable FIR Filter Architectures

We implement the 10-tap programmable FIR filters based on HRBM and HRBM^2 respectively. For high performance, the transposed direct form architecture with CSA summation is used to implement the FIR filter. Fig. 4 shows the FIR filter using radix-16 HRBM consists of one odd multiples precomputation and ten basic units and adders (BUAs) of HRBM. The HRBM scheme efficiently removes the redundant computations by sharing the odd multiples pre-computation in FIR filter operation, which leads to low-power design. For further improvement of high performance, Fig. 5 shows the FIR filter using radix-32-modulo-7 HRBM^2 only consists of ten BUAs of HRBM^2. Since the odd multiple of radix-32-modulo-7 HRBM^2 is only {1}, it efficiently eliminates the propagation delay of precomputing multiples.

4. RESULTS AND COMPARSIONS

Two 10-tap programmable FIR filters using proposed $19\times$ 19-bit HRBM and HRBM^2 are synthesized by TSMC 0.18 μm CMOS technology. We also implement the 10-tap FIR filters using $19\times19-$bit carry save array multiplier (CSAM),

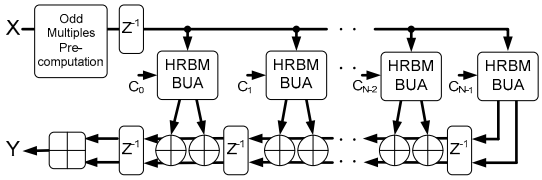

Fig. 4 Programmable FIR filter using HRBM.

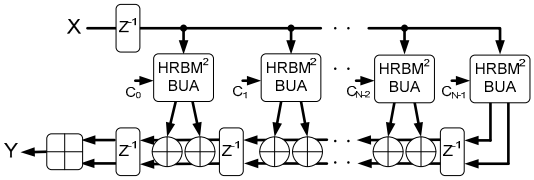

Fig. 5 Programmable FIR filter using HRBM^2.

Table 3 Design Results of 10-tap Programmable FIR Filter

	For Low Power Design				
	HRBM	HRBM^2	CSHM	WTM	CSAM
Clock cycle (ns)	5.8	5.8	5.8	5.8	5.8
Area (gates)	36455	45582	47261	39501	60298
Power@172 MHz (mW)	68.32	77.71	126.00	108.38	137.10
	For High Speed Design				
	HRBM	HRBM^2	CSHM	WTM	CSAM
Clock cycle (ns)	2.6	2.0	4.8	4.4	5.8
Area (gates)	37634	47193	59101	51070	60298
Power@172 MHz (mW)	69.68	78.55	160.31	142.78	137.10

Wallace tree multiplier (WTM), and computation sharing multiplier (CSHM) [4] for comparisons. WTM and CSAM are two widely used multipliers. Since the tree structure of partial products summation, WTM has better performance than CSAM. CSHM also has the concept of computation sharing for low complexity FIR filter design. We design the 10-tap FIR filter using CSHM as proposed in [4]. Since the filter architecture in [4] has one pipeline stage, we also implement the filters using different multipliers with one pipeline stage. Table 3 shows the clock cycle, area and power of the filters using different multipliers for low power or high speed design constraints. The power results shown in Table 3 are measured by the clock rate of 172 MHz (clock cycle=5.8 ns) for low power or high speed design constraints.

Since WTM- and CSAM-based architectures do not use computation sharing and perform redundant computations for all taps, the FIR filter using HRBM shows better results in terms of speed and power. CSHM-based architecture requires eight odd multiples precomputations, {1, 3,..., 15}, that are two times of odd multiples requirement of HRBM-based architecture. The basic unit of CSHM is more complicated than the BUA of HRBM. Besides, HRBM-based architecture uses CSA summation to efficiently reduce critical path delay. Therefore, HRBM-based architecture has better speed and power results than CSHM-based architecture. Fig. 6 shows normalized power, area, and clock cycle of different multiplier-based architectures. The FIR filter using HRBM has 45.8%, 40.9%, and 55.2% performance improvement over FIR filter using CSHM, WTM, and CSAM. HRBM-based architecture has 45.7%, 37.0%, and

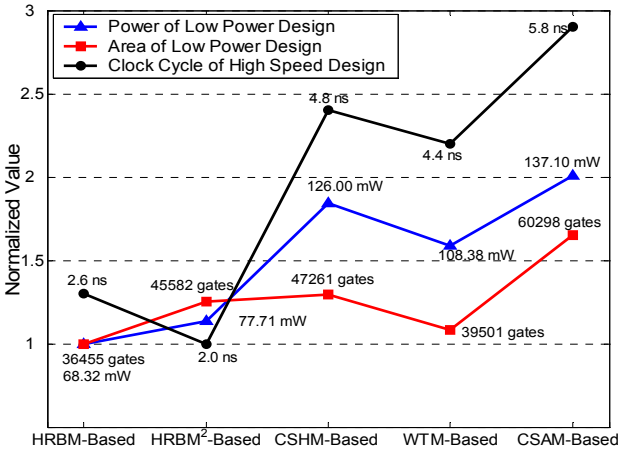

Fig. 6 Design results comparison of different multiplier-based.

50.2% power reduction over the FIR filter based on CSHM, WTM, and CSAM. Furthermore, HRBM^2-based architecture efficiently eliminates the propagation delay of precomputing multiples. In terms of performance and power consumption, the HRBM^2 scheme has 54.5~65.5% and 28.3~43.3% improvement with respect to the FIR filter based on CSHM, WTM, and CSAM.

5. CONCLUSIONS

We present novel programmable digital FIR filters for low-power and high-performance applications. The programmable FIR filters using $19\times19-$bit HRBM and HRBM^2 are design by TSMC 0.18 μm CMOS technology. We also design $19\times19-$bit WTM-, CSAM-, and CSHM-based architectures of FIR filters for comparisons. Using higher radix recoding to decreasing partial products and precomputation sharing in each partial product, HRBM-based architecture has better results in terms of performance and power consumption that are about 40.9~55.2% and 37.0~50.2% improvement over other designs. Extending higher radix recoding scheme with secondary radix recoding can further improve performance by reducing the propagation delay of precomputing odd multiples. Thus, HRBM^2-based architecture can improve performance about 54.5~65.5% and power consumption about 28.3~43.3% over other designs.

11. REFERENCES

[1] C.N. Lyu and D.W. Matula, "Redundant Binary Booth Recoding," *Proc. 12th IEEE int'l Symp. Computer Arithmetic*, pp. 50-57, 1995.

[2] P.E. Madrid, B. Millar, and E.E. Swartzlander, "Modified Booth Algorithm for High Radix Multiplication," *Proc. IEEE Computer Design Conf.*, pp. 118-121, 1992

[3] P.M. Seidel, L.D. McFearin, and D.W. Matula, "Secondary Radix Recoding for Higher Radix Multipliers," *IEEE Trans. Computers*, vol. 54, no. 2, pp. 111-123, Feb. 2005.

[4] J. Park, W. Jeong, H. Mahmoodi-Meimand, Y. Wang, H. Choo, and K. Roy, "Computation Sharing Programmable FIR Filter for Low-Power and High-Performance Applications," *IEEE J. Solid-State Circuits*, vol. 39, no. 2, pp. 348-357, Feb. 2004.