Design of A High-Order FIR Filter Based on RNS Algorithm

Wei Wang, Shuangqiao Li [a] and Ting Chen

Chongqing University of Posts and Telecommunications, Chongqing 400065, China;

[a]shuangqiao_li@sina.com

Abstract

This paper presents a 128-tap finite impulse response (FIR) filter based on the residue number system (RNS) with the five moduli set $\{2^n-1, 2^n, 2^n+1, 2^{n-1}-1, 2^{n+1}-1\}$. The binary-to-residue conversion is achieved using a purely combinational logic circuit based on a Wallace tree structure. Compared with the normal structure, the multiply-accumulate (MAC) unit in this paper incorporates the addition into the partial product adding, which reduces the use of a modulo adder and further reduces the delay. In addition, by moderating the intermediate result of the carry-save adder (CSA), the increasing of the bit width caused by the addition is avoided, thereby the complexity of the operation is simplified. The architecture is designed and implemented in the FPGA XC5VFX70T device. The synthesis results show that the filter has a delay of 3.55 ns and a power consumption of 2585 mw. The proposed FIR filter consumes less hardware than other designs simultaneously.

Keywords

FIR filter; RNS; forward converter; MAC.

1. Introduction

In recent years, more and more complicated real-time dedicated digital signal processing functions have been embedded in mobile electronic devices. As an alternative digital system, the Residue number System (RNS) is even more advantageous in mapping complex DSP algorithms to dedicated hardware accelerators. Addition, subtraction and multiplication of longer bits can be decomposed into smaller carry-less modes for parallel processing. By introducing sub-level parallel processing, the digital system architecture using the RNS algorithm can sustain significant reductions in supply voltage over a given delay range. The reduction of the power and voltage can reduce the power consumption to the square root of the original value, so that the multi-voltage and multi-threshold voltage reduction technologies can be more effectively used to further reduce the static power consumption and the dynamic power consumption of the system [1,2]. The RNS algorithm has been used in the design of digital system architectures such as IPSP [3,4].

In particular, the application of RNS algorithm in high-speed FIR filter design has attracted people's attention [3]. As a basic component of DSP, FIR filter is easy to realize, stable and linear phase, so it is widely used in many DSP fields such as speech and image processing, noise reduction and pulse formation. Delay is a very important parameter that affects the performance of the filter. As a multiply-intensive device, binary adders and multipliers are affected by the delay of the carry chain, which makes FIR filters in the traditional two's complement system (TCS) difficult to achieve the double requirement of real-time and accuracy [5].

RNS, as a kind of non-authorized system, can divide a large bit-wide operand into several smaller operands by a specific moduli set and make them calculated in parallel in each modulo channel with no carry propagation between them. Thus the calculation speed is improved. Because of the more

efficient implementation of addition and multiplication, RNS is also a very useful method for implementing high-speed FIR filters. In recent years, some achievements have been made in the research of RNS FIR filter. N. Stamenković designed a 32-bit RNS FIR filter in 2009 [6]. The RNS FIR filter shows a significant improvement in speed compared to the TCS FIR filter. In 2012, M. Petricca and P. Albicocco implemented a high-order RNS FIR filter with a series / parallel architecture [7], achieving less delay and smaller area than the TCS FIR filter. Most systems construct the moduli set by three components $\{2^n\text{-}1, 2^n, 2^n\text{+}1\}$ [8,9]. For some applications that require a large dynamic range, such as cryptography, the value of each mode component must be increased with the dynamic range, and the performance of the arithmetic unit in each mode channel also decreases. The three-channel moduli set no longer suitable[10]. For the system with large dynamic range, this paper choose $\{2^n-1, 2^n, 2^n+1, 2^{n-1}-1, 2^{n+1}-1\}$ as the moduli set. The forward conversion use the pure combinational logic circuit instead of the look-up table (LUT) to realize modular multiplication. Modulo 2^n-1 multiply-accumulate (MAC) circuit is applied to the sub-filter channel calculation.

Section 2 introduces the RNS theoretical background and the overall architecture of the FIR filter based on RNS. Section 3 details the design of the TCS to RNS conversion circuit and the modulo 2^n-1 multiply-accumulator. The FPGA synthesis results and comparisons are presented in Section 4. Finally, the conclusions are drawn in Section 5.

2. RNS Algorithm and FIR Filter Architecture

The RNS is defined by moduli set, which consists of n pairwise relatively prime integers $\{m_0, m_1, \ldots, m_{n-1}\}$, and $M = \prod_{i=0}^{n-1} m_i$ is the dynamic range. A number X within the dynamic range can be represented by the list of its residues x_i with respect to the moduli m_i defined in the moduli set. The RNS representation of X can be written based on the congruence:

$$|X|_{m_i} = x_i \tag{1}$$

Where, x_i is calculated using:

$$x_i = \begin{cases} X \bmod m_i, & X \geq 0 \\ (m_i - X) \bmod m_i, & X < 0 \end{cases} \tag{2}$$

In RNS, any number in range [0, M-1] has a unique set of residues. What's more, the addition, subtraction and multiplication operation can be performed entirely on the residue representation of the operands.

Let the RNS representations of X and Y be given by $\{x_0, x_1, \ldots, x_{n-1}\}$ and $\{y_0, y_1, \ldots, y_{n-1}\}$ respectively. Then

$$|X \bullet Y|_M = \{|x_0 \bullet y_0|_{m_0}, |x_1 \bullet y_1|_{m_1}, \ldots, |x_{n-1} \bullet y_{n-1}|_{m_{n-1}}\} \tag{3}$$

where the operation $\bullet$ can be either addition, subtraction or multiplication.

The choice of the moduli set affects the complexity, power consumption, and delay of the RNS. There are two commonly used moduli set: 1) mosuli set for arbitrary modulo values; and 2) special moduli set with exponentials of power of 2, eg $2^n \pm 1$, 2^n. The two types of moduli set have their own advantages: the moduli set of arbitrary modulus performs well on balance and degree of freedom; the modular addition and multiplication under special moduli set is simpler, which can simplify the computation unit and converter design. Therefore, most systems are designed with special moduli set $\{2^n\text{-}1, 2^n, 2^n\text{+}1\}$. With the continuous increase of the dynamic range, the ordinary 3-channel special moduli set can't meet the needs of practical application. This paper choose the five moduli set $\{2^n-1, 2^n, 2^n+1, 2^{n-1}-1, 2^{n+1}-1\}$ [10]. Each of the components in the moduli set has the form of 2^n and $2^n \pm 1$, and the dynamic range is expanded to 5n-1 bits.

The MAC units of modulo 2^n+1 channel are realized by diminished-1 method [11]. The input X of n+1 bits is expressed as $x_z\boldsymbol{X}^*$, where x_z is the zero detection bit, and $\boldsymbol{X}^*$ is the representation of X-1 with n bits. The modular multiplication consists of partial product generation, a carry-save adder (CSA) tree

and a modulo 2^n+1 adder [12]. Modulo 2^n+1 adder with diminished-1 method can be implemented by the parallel prefix structure [13]:

$$\langle X'+Y'+1\rangle_{2^n+1}=\left\langle X'+Y'+\overline{C_{\text{out}}}\right\rangle_{2^n} \tag{4}$$

$$(G_i^*,P_i^*)=\begin{cases}\overline{(G_{n-1},P_{n-1})} & i=-1\\ (G_i,P_i)\circ\overline{(G_{n-1},P_{n-1})} & 0\le i\le n-2\end{cases} \tag{5}$$

where $\circ$ is the prefix operator. In order to further reduce the delay, the input is directly converted from binary to the residues of the diminished-1 form through the TCS to RNS conversion circuit and calculated in the sub-filtering channel. The operation of adding by 1 is realized in the last tap.

The difference equation of an N-order FIR filter is:

$$y(n)=\sum_{k=0}^{N-1}a_k x(n-k) \tag{6}$$

where x is the input of the filter, a_k and is the coefficient, $0\le k\le N-1$. y is the output of the filter.

The modulo set is {7, 15, 16, 17, 31} when n = 4, and the structure of the transposed FIR filter corresponding to it is shown in Figure 1.

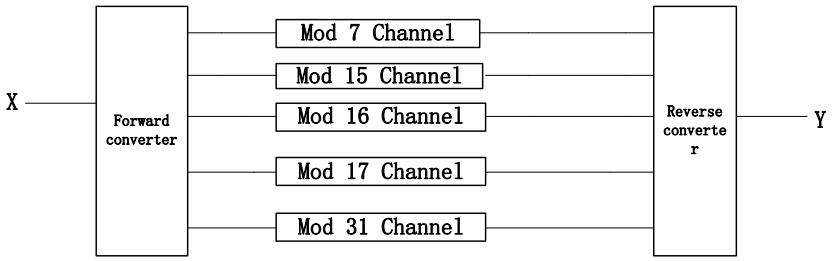

Fig. 1 RNS-based FIR filter architecture.

3. FIR architecture design

3.1 forward conversion.

Forward converter transforms the inputs from 2's complement format to RNS representation. The conversions of data increase the overhead of the system which need to be minimized. The five moduli set {2^n-1, 2^n, 2^n+1, 2^{n-1}-1, 2^{n+1}-1} is chosen to accelerate the speed of the Sub-filter channels. The residues of a binary number X with respect to the five moduli set is shown in (7) [10]. X is partitioned into q-number of possible blocks: B_1, B_2, ... , B_q. The bits of each block are the same as the power of 2.

$$\begin{cases}r_1=\left|B_1+B_2+\cdots+B_q\right|_{2^{n-1}-1}\\ r_2=\left|B_1+B_2+\cdots+B_q\right|_{2^n-1}\\ r_3=\textit{The least significant n bits of } X\\ r_4=\left|B_1-B_2+B_3-B_4+\cdots-B_{q-1}+B_q\right|_{2^n+1}\\ r_5=\left|B_1+B_2+\cdots+B_q\right|_{2^{n+1}-1}\end{cases} \tag{7}$$

In order to simplify the modular operation, the Wallace tree structure is adopted. Suppose a binary number $A=a_n a_{n-1}\cdots a_0=\sum_{k=0}^{n}a_k 2^k$, then A mod 2^n-1 can be expressed as follows:

$$\begin{aligned}|A|_{2^n-1}&=\left|a_n\cdot 2^n+a_{n-1}\cdot 2^{n-1}+\cdots+a_0\right|_{2^n-1}\\ &=\left|\left|a_n\cdot 2^n\right|_{2^n-1}+a_{n-1}+\cdots+a_0\right|_{2^n-1}\\ &=\left|\left|a_n(2^n-1)+a_n\right|_{2^n-1}+a_{n-1}+\cdots+a_0\right|_{2^n-1}\\ &=\left|a_n+2^{n-1}\cdot a_{n-1}+\cdots+a_0\right|_{2^n-1}\\ &=\left|2^{n-1}\cdot a_{n-1}+\cdots+(a_0+a_n)\right|_{2^n-1}\end{aligned} \tag{8}$$

The result of A mod 2^n-1 is obtained by shifting the most significant bit (MSB) of the carry to the least significant bit (LSB). Therefore, the modulo 2^n-1 conversion can be realized by the End Around Carry (EAC) Carry Save Adder (CSA) tree and a modulo 2^n-1 adder.

In this paper, modulo 2^n+1 addition is implemented by diminished-1 number system. The negative number can be converted into a positive format with formula $|-B|_{2^n+1} = |2^n+1-B|_{2^n+1}$, and all components are expressed in the form of diminished-1. The residues generation of A mod 2^n+1 is given by

$$\begin{aligned} |A|_{2^n+1} &= \left|a_n \cdot 2^n + a_{n-1} \cdot 2^{n-1} + \cdots + a_0\right|_{2^n+1} \\ &= \left|\left|a_n(2^n+1-1)\right|_{2^n+1} + a_{n-1} + \cdots + a_0\right|_{2^n+1} \\ &= \left|-a_n + a_{n-1} + \cdots + a_0\right|_{2^n+1} \\ &= \left|a_{n-1} + \cdots + (a_0 - a_n)\right|_{2^n+1} \end{aligned} \tag{9}$$

An Inverted End Around Carry (IEAC) CSA tree is used to invert the MSB and shift to the LSB, and a correcting action (COR) is needed. There are two reasons to use the COR, the first one is that the result of the IEAC CSA is 1 larger than the actual result; and the second, the final results need to be added the q. So, the COR is expressed with COR=q-S_{ISCSA}-S_0, where, the S_{ISCSA} is the number of IEAC CSA, and the S_0 is the number of zero input. The structure of Binary to RNS converter based on Wallace tree is shown in Figure 2.

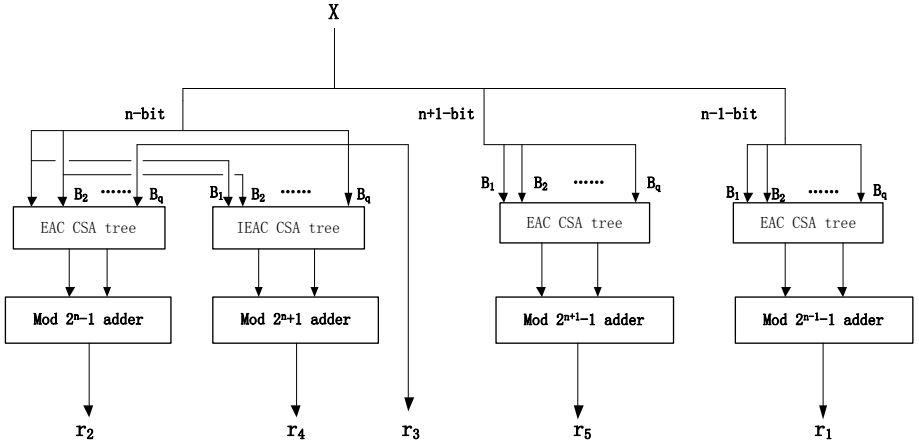

Fig. 2 The Binary to RNS converter architecture based on {2^n-1, 2^n, 2^n+1, 2^{n-1}-1, 2^{n+1}-1}

3.2 Modulo 2^n-1 multiply-accumulate circuit.

In the tap of the modulo 2^n-1 sub-filter channel, a modulo 2^n-1 multiplier cascaded modulo 2^n-1 adder is needed to achieve the adding operation of the input and coefficient products. Modulo 2^n-1 multiplier is mainly divided into three parts: 1) generate partial product; 2) summation of partial products; 3) addition using modulo adder. The partial product can be generated by the AND gate array. The product of the partial products is summed up by the Wallace tree structure. Finally, the output of the Wallace tree is summed by modulo 2^n-1 adders.

Using (8), we can simplify the generation of partial products and obtain the partial products in the form of residues directly. Suppose two 4-bit numbers x, y are the input of the modulo 2^4-1 multiplier, the partial product of which is shown in Figure 3.

2^6	2^5	2^4	2^3	2^2	2^1	2^0	
			x_3y_0	x_2y_0	x_1y_0	x_0y_0	$=pp_0$
		x_3y_1	x_2y_1	x_1y_1	x_0y_1	x_3y_1	$=pp_1$
	x_3y_2	x_2y_2	x_1y_2	x_0y_2	x_3y_2	x_2y_2	$=pp_2$
x_3y_3	x_2y_3	x_1y_3	x_0y_3	x_3y_3	x_2y_3	x_1y_3	$=pp_3$

Fig. 3 Partial products of modular 24-1 multiplier

Summation of partial products uses the EAC CSA tree structure in Section 2.1 to simplify the modulo operation. The last stage of the multiplier is a modulo adder whose output is the input to the

accumulator. The other input is the output of the previous tap. In order to further improve the speed, this paper put the accumulate operations into the EAC CSA tree, then the entire multiply-accumulate structure contains only one modulo adder. The structure of modulo 2^n-1 multiply-accumulate operations is shown in Figure 4.

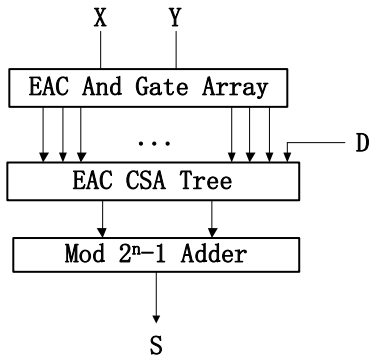

Fig. 4 The structure of modulo 2^n-1 multiply-accumulate operation

4. Results Analysis and Discussion

A 128-tap RNS FIR filter is designed with input and coefficient bit widths of 16 bits and 12 bits respectively. The filter is synthesized using a Xilinx Virtex-5 device. The comparison of area, power and delay of RNS DWT filter banks with previously published results has been summarized in following.

Table 1 Comparison of RNS and TCS DWT filter bank

	[7]	[14]	This work
Taps	128	4	128
Delay/ns	3.95	3.64	3.55
Power/mW	3 144	-	2 585

It can be seen from Table 1 that the 128-tap RNS FIR filter designed in this paper is better than the 4-tap RNS FIR filter based on LUT in [14] for delay. At the same number of taps, the proposed filter reduces latency by 10% compared to [7] and reduces power consumption by 17%. Although the number of taps is significantly reduced by using the serial / parallel structure in [7], the normal operation unit used in [7] increases the overall operation time. In this paper, we use the modified CSA to construct the Wallace tree structure and omit the accumulator in the tap structure, which further improves the speed of forward conversion and modulo 2^n-1 multiply-add operation and obtains better delay performance.

Table 2 Comparison of FPGA resource utilization

	[7]	This work
Slice/%	59	38
LUT/%	38	31
FFs/%	29	6

Table 2 shows the comparison between this article and [7] in FPGA resource utilization. The forward converter and the multiplier used in [7] contain multiple LUTs. But this paper uses pure combinational logic to achieve forward conversion and multiply-accumulate operations. Although the number of taps is much larger than that in [7], the hardware resources utilization is reduced.

5. Conclusion

In this paper, a 128-tap transposed RNS FIR filter is designed based on the five moduli set {2^n-1, 2^n, 2^n+1, 2^{n-1}-1, 2^{n+1}-1}. In the TCS to RNS conversion circuit, a Wallace tree is used to improve the

speed of the operation. The output of each CSA is residue form, which avoiding the inconvenience of the adding of the bit width to the final modulo addition. The forward converter circuit is a pure combinational logic circuit, which greatly saves the hardware resources required for large bit wide data conversion. At the same time, the modulo accumulate operation is merged into the partial product summation of the modulo multiplication. The Wallace tree structure and a modulo 2^n-1 adder are used to carry out the multiply-accumulate operation, so that one level accumulator is omitted, and then the speed is further increased and the resource consumption is reduced . The synthesis results show that, compared to the [7], the FIR filter designed in this paper is with 10% less delay and 17% less power consumption. The utilization of Slice is also reduced by 35% .

Acknowledgements

I would like to express my sincere appreciation to my supervisor, Professor Wang, for his academic guidance and personal support through every phase of this dissertation, and also for his encouraging my accomplishment of the whole research. Without his great patience and helpful suggestions, the completion of this dissertation would not have been possible.

References

[1] C.H. Chang, J.Y.S. Low: Simple, Fast, and Exact RNS Scaler for the Three-Moduli Set, IEEE Trans Circ & Syst I: Regu Pap, Vol. 58 (2011) No. 11, p.2686-2697.
[2] G C. Cardarilli, A D. Re, A. Nannarelli, et al. Low power and low leakage implementation of RNS FIR filters, 39th ACSSC. Pacific Grove, (2005), p.1620-1624.
[3] R. Conway, J. Nelson: Improved RNS FIR filter architectures, IEEE Trans Circ & Syst II: Expr Bri, Vol. 51 (2004) No. 1, p.26-28.
[4] G L. Bernocchi, G C. Cardarilli, A D. Re, et al. Low-power adaptive filter based on RNS components, IEEE Int Symp Circ & Syst, (2007), p.3211-3214.
[5] A. Omondi, B. Premkumar: Residue number systems: theory and implementation (Imperial College Press, UK 2007), p.41-55.
[6] N. Stamenkovic: Digital FIR filter architecture based on the residue number system, Facta universitatis-series: Electronics & Energetics, Vol. 22 (2009) No. 1, p.125-140.
[7] M. Petricca, P. Albicocco, G C. Cardarilli, et al. Power efficient design of parallel/serial FIR filters in RNS, 46th ACSSC. Pacific Grove, (2012), p.1015-1019.
[8] G C. Cardarilli, A. Nannarelli and M.Re: Residue number system for low-power DSP applications, 41th ACSSC. Pacific Grove, (2007), p.1412-1416.
[9] A. Mirshekari, M.Mosleh: Hardware implementation of a fast FIR filter with residue number system, 2nd ICIMA. Wuhan, (2010), p. 312-315.
[10] D. Boruah, M. Saikia: A pure combinational logic gate based forward converter for new five moduli set RNS, Second ICACCE. Dehradun, (2015), p.301-307.
[11] C. Efstathiou, H T. Vergos and D. Nikolos: Handling zero in diminished-one modulo 2^n+1 adders, Int J Elec, Vol. 90 (2003) No. 2, p.133-144.
[12] C. Efstathiou, I. Voyiatzis: On the diminished-1 modulo 2^n+1 fused multiply-add units, 6th Int Conf Des & Technol Integr Syst Nano Era, (2011), p.1-5.
[13] G. Jaberipur, S. Nejati: Balanced minimal latency RNS addition for moduli set $\{2^n-1, 2^n, 2^n+1\}$, 18th Int Conf Syst, Signals & Image Process, (2011), p.1-7.
[14] M D. Namdev, M R. Patel: An efficient adaptive RNS filter design on high speed FPGA, Int J Modern Engineer & Managem Resear, Vol. 2 (2015) No.3, p.41-45.