New adaptable three-moduli set \(\{2^{n+k}, 2^n - 1, 2^{n-1} - 1\}\) for residue number system-based finite impulse response implementation

Ming-Hwa Sheu\(^{(a)}\), Siang-Min Siao\(^1\), Yin-Tsung Hwang\(^2\), Chi-Chia Sun\(^3\), and You-Ping Lin\(^1\)

\(^1\) Department of Electronic Engineering, National Yunlin University of Science & Technology, 123 University Road, Section 3, Douliou, Yunlin 64002, Taiwan, R.O.C.
\(^2\) Department of Electrical Engineering, National Chung Hsing University, 145 Xingda Rd., South Dist., Taichung 402, Taiwan R.O.C.
\(^3\) Department of Electrical Engineering, National Formosa University, No.64, Wunhua Rd., Huwei Township, Yunlin 632, Taiwan, R. O. C.

\(^{(a)}\) sheumh@yuntech.edu.tw

Abstract: This paper presents a new adaptable three-moduli set, \(\{2^{n+k}, 2^n - 1, 2^{n-1} - 1\}\). It has three advantages for designing residue number system (RNS)-based digital signal processor (DSP) systems. First, it does not include a \((2^n + 1)\)-type modulo, thus providing a high-speed and low-cost system design. Second, three parallel DSP module channels achieve more efficient delay-balancing, thereby enhancing the system operation speed. Third, the set possesses an adaptable \(2^{n+k}\) modulo for avoiding over-ratio problems and reducing the system hardware overheads. Through the implementation of a mixed-radix conversion concept, an efficient converter was derived for the proposed adaptable moduli set. For system evaluation and comparison, the proposed adaptable and related moduli sets were used to implement a 16-tap RNS-based finite impulse response module that contains a forward converter, FIR module, and reverse converter. Based on TSMC 90-nm CMOS process technology, all implementations were synthesized to obtain layout results for a performance comparison. The design derived using the proposed moduli set achieved a 12%–46% \(\text{AD}^2\) (area × delay\(^2\)) saving compared with those derived using other moduli sets or binary number systems.

Keywords: adaptable moduli set, mixed-radix conversion (MRC), reverse converter, residue number system (RNS), RNS-based finite impulse response

Classification: Integrated circuits

References

[1] B. Parhami: *Computer Arithmetic: Algorithms and Hardware Designs* (Oxford University Press, 2010).
1 Introduction

The residue number system (RNS) [1] is an integer number system that can support parallel carry-free operations that increase the arithmetic unit speed. Because of these features, an RNS can be used in high-speed digital signal processing (DSP) applications such as FIR filter [2] and error detection [3]. In general, the moduli set selection directly affects the performance of RNS-based DSP system including forward converter, DSP module, and reverse converter. From the literature, \(\{2^n - 1, 2^n, 2^{n+1}\}\) [4] is a well-known three-moduli set that can provide a fast reverse converter, but it is a fixed moduli set that can cause dynamic range (DR) over-ratio problems [5], resulting in hardware overdesign. To overcome this
problem, many adaptable moduli sets, such as \( \{2^n + 1, 2^{n+k}, 2^n - 1\} \) [6], \( \{2^k, 2^n + 1, 2^n - 1, 2^n + 1\} \), and \( \{2^k, 2^n + 1, 2^n - 1, 2^n - 1\} \) [7], have been presented. They use the two parameters \( n \) and \( k \) to adjust their DRs. However, these moduli sets and [4] can lead to channel delay imbalance problems that degrade system operation time. Furthermore, these sets involve the \( 2^n + 1 \) modulo, which can increase the complexity of arithmetic architecture and latency. For further analysis, \( 2^n - 1 \), \( 2^n \), and \( 2^n + 1 \) moduli are used to implement a multiplier and accumulator (MAC) [8] with the three respective sets. Table I lists their delay and area overheads for comparison. The \( 2^n + 1 \) modulo always requires the highest hardware cost and causes the longest delay. The area and delay overheads are up to 35.6\% and 45.2\%, respectively. Therefore, previous studies have proposed several \( (2^n + 1) \)-free moduli sets, such as \( \{2^n, 2^n - 1, 2^{n+1} - 1\} \), \( \{2^n, 2^n - 1, 2^n - 1\} \) [9], \( \{2^n, 2^n - 1, 2^{n-1} - 1\} \) [10], and \( \{2^n - 1, 2^n, 2^{2n+1} - 1\} \) [11], but they are fixed and difficult to implement in addressing the DR over-ratio problem.

This paper first presents an adaptable and \( (2^n + 1) \)-free moduli set, \( \{2^{n+k}, 2^n - 1, 2^{n-1} - 1\} \), that was derived from a fixed moduli set [9]. Subsequently, a new reverse conversion algorithm is derived and the design of its converter architecture is presented. The RNS-based FIR system was used to evaluate the performance of the proposed adaptable moduli set. The experimental results show that the minimal delay of our system improved by up to 22\% compared with that of systems proposed in related studies [4, 6]. At a 25-MHz working frequency, the proposed design improved the power and cost savings by up to 28\% and 12\%, respectively. Compared with binary-based designs, the proposed RNS-based design achieved power and delay savings of up to 23\% and 17\%.

The remainder of this paper is organized as follows: Section II introduces a new reverse converter algorithm for the proposed moduli set. Subsequently, a low-cost

| \( n \)  | Area \( \times 10^3 \) (\( \mu \text{m}^2 \)) | Area Overhead (%) | | Delay (nS) | Delay Overhead (%) |
|--------|-----------------|------------------|----------------|----------------|------------------|
|        | \( 2^n \) | \( 2^n - 1 \) | \( 2^n + 1 \) | \( 2^n \) | \( 2^n - 1 \) | \( 2^n + 1 \) | \( 2^n \) | \( 2^n - 1 \) | \( 2^n + 1 \) |
| 3      | 30.6           | 24.1            | 37.4           | 35.6          | 2.5            | 2.4            | 2.9            | 24.1          |
| 6      | 33.9           | 27.9            | 41.3           | 32.4          | 2.6            | 2.5            | 3.2            | 21.9          |
| 9      | 37.1           | 31.7            | 45.1           | 29.7          | 3.3            | 3.1            | 4.2            | 26.2          |
| 11     | 58.6           | 51.7            | 70.9           | 27.1          | 4.6            | 4.3            | 5.9            | 27.1          |
| 14     | 64.8           | 59.2            | 78.4           | 24.5          | 4.9            | 4.7            | 7.2            | 34.7          |
| 17     | 73.7           | 69.1            | 88.9           | 22.3          | 5.2            | 4.9            | 8.2            | 40.2          |
| 19     | 86.5           | 83.1            | 104.1          | 20.2          | 5.5            | 5.4            | 9.9            | 43.8          |
| 22     | 97.7           | 95.2            | 117.4          | 18.9          | 5.8            | 5.7            | 10.4           | 45.2          |

\( * \)Modulo area overhead (%): \( \frac{\text{area}_{\text{Max}} - \text{area}_{\text{Min}}}{\text{area}_{\text{Max}}} \times 100\% \)

\( ** \)Modulo delay overhead (%): \( \frac{\text{channel}_{\text{longest}} - \text{channel}_{\text{shortest}}}{\text{channel}_{\text{longest}}} \times 100\% \)
architecture is designed and analyzed. Next, Section III presents the entire system performance evolution and compares it with that of related moduli sets. Furthermore, implementation of the proposed three-moduli set and reverse converter in an RNS-based FIR module is described. Finally, the conclusion of this study is offered.

2 Reverse conversion algorithm and hardware design

For a new moduli set, reverse conversion is the most crucial consideration. This section describes the development of the reverse conversion algorithm for the proposed three-moduli set \( \{2^{n+k}, 2^n - 1, 2^{n-1} - 1\} \) on the basis of mixed-radix conversion (MRC). The binary number \( X \) can be obtained by performing binary conversion of the residue numbers \( x_1 = \langle X \rangle_{m_1=2^{n+k}}, \ x_2 = \langle X \rangle_{m_2=2^n-1}, \) and \( x_3 = \langle X \rangle_{m_3=2^{n-1}-1} \), where \( \langle X \rangle_{m_i} \) denotes \( X \mod m_i \).

The MRC of \( (m_1, m_2, m_3) \) is given by

\[
X = v_1 + m_1 v_2 + m_1 m_2 v_3 = v_1 + m_1 (v_2 + m_2 v_3),
\]

where \( v_1 = x_1, \ v_2 = (x_2 - x_1)k_1, \ v_3 = (x_3 - x_1)k_2 - v_2k_3 \).

The terms \( k_1, \ k_2, \) and \( k_3 \) represent three multiplicative inverses: \( k_1 = \langle m_1^{-1} \rangle_{m_1} = 2^{n-k}, \ k_2 = \langle m_1^{-1} \rangle_{m_2} = 2^{n-k-2}, \) and \( k_3 = \langle m_1^{-1} \rangle_{m_3} = 1. \)

After substitution, Eq. (1) can be rewritten as follows:

\[
X = x_1 + 2^{n+k}[v_2 + (2^n - 1)v_3],
\]

where

\[
v_2 = (x_2 - x_1)2^{n-k}\!
\]

\[
v_3 = (x_3 - x_1)2^{n-k-2} - v_2\!
\]

\[
= \langle L_1 + L_2 \rangle_{2^{n-1}}.
\]

Accordingly, \( L_1 \) and \( L_2 \) can be simplified to Eqs. (5) and (6):

\[
L_1 = \left(2^{n-k}\!
\]

\[
= \langle L_1 + L_2 \rangle_{2^{n-1}}.
\]

\[
L_2 = -2^{n-k}\!
\]

\[
= \langle L_{21} + L_{22} \rangle_{2^{n-1}}.
\]

Hence, \( v_2 \) in Eq. (3) can be rewritten as

\[
v_2 = \langle L_1 + L_2 \rangle_{2^{n-1}} = v_{2,n-1} \ldots v_{2,0}.
\]

Subsequently, \( L_3, \ L_4, \) and \( L_5 \) in Eq. (4) are computed as follows:
\[ L_3 = \left( \frac{2^{n-k-2}(x_{3,n-2} \ldots x_{3,0})}{n-1} \right)_{2^{n-1}-1} = \left( \frac{2^{n-k-2}(x_{3,n-2} \ldots x_{3,k+1} x_{3,k} \ldots x_{3,0})}{n-2} \right)_{2^{n-1}-1} \]  
\[ = \frac{x_{3,k} \ldots x_{3,0} x_{3,n-2} \ldots x_{3,k+1}}{k+1} n-k-2 \]  
\[ = \left( L_{41} + L_{42} \right)_{2^{n-1}-1} \]  
\[ L_4 = \left( \frac{-2^{n-k-2}(x_{1,n+k-1} \ldots x_{1,0})}{n+k} \right)_{2^{n-1}-1} \]  
\[ = \left( \frac{-2^{n-k-2}(0 \ldots 0 x_{1,n+k-1} \ldots x_{1,n-1} + x_{1,n-2} \ldots x_{1,k+1} x_{1,k} \ldots x_{1,0})}{n-k-2} \right)_{2^{n-1}-1} \]  
\[ = \left( \frac{x_{1,n+k-1} \ldots x_{1,n-1} \ldots 1 + x_{1,k} \ldots x_{1,0} x_{1,n-2} \ldots x_{1,k+1}}{k+1} n-k-2 \right)_{2^{n-1}-1} \]  
\[ = \left( L_{41} + L_{42} \right)_{2^{n-1}-1} \]  
\[ L_5 = \left( \frac{v_{2,n-1} \ldots v_{2,0}}{n} \right)_{2^{n-1}-1} = \left( \frac{-(0 \ldots 0 v_{2,n-1} \times 2^{n-1} + v_{2,n-2} \ldots v_{2,0})}{n-1} \right)_{2^{n-1}-1} \]  
\[ = \left( \frac{1 \ldots 1 v_{2,n-1} + v_{2,n-2} \ldots v_{2,0}}{n-1} \right)_{2^{n-1}-1} = \left( L_{51} + L_{52} \right)_{2^{n-1}-1} \]  

These equations are resubstituted into Eq. (4), and \( v_3 \) can be represented as
\[ v_3 = \left( L_3 + L_{41} + L_{42} + L_{51} + L_{52} \right)_{2^{n-1}-1} \]  
\[ = \frac{v_{3,n-2} \ldots v_{3,0}}{n-1} \]  
\[ (11) \]

Finally, \( X \) can be calculated as
\[ X = x_1 + 2^{n+k}[v_2 + 2^n v_3 - v_3] \]  
\[ = x_1 + 2^{n+k}(P - v_3) \]  
\[ (12) \]

where
\[ P = v_2 + 2^n v_3 = \frac{v_{3,n-2} \ldots v_{3,0} v_{2,n-1} \ldots v_{2,0}}{n-1} \]  
\[ (13) \]
\[ v_3 = \frac{v_{3,n-2} \ldots v_{3,0}}{n-1} \]  
\[ (14) \]

Derived from this conversion algorithm, the hardware architecture of the reverse converter for \( \{2^{n+k}, 2^{n-1} - 1, 2^n - 1\} \) is shown in Fig. 1. Its function blocks were designed on the basis of Eqs. (7), (11), and (12). First, the operand preparation unit (OPU 1) is a bit organizer that simply performs permutations by using Eqs. (5) and (6) to generate three results. Subsequently, the results are summed by one \( (n) \)-bit carry-save adder (CSA 1) with an end-around carry (EAC) and one \( (n) \)-bit carry-propagate adder (CPA 1) with the EAC to obtain \( v_2 \). According to Eqs. (8)–(10), \( x_1, x_3, \) and \( v_2 \) are inputted to OPU 2, which is also a bit organizer, to produce five outputs. Subsequently, three \( (n-1) \)-bit CSAs (CSA 2, CSA 3, and CSA 4) with an EAC, one modulo \( (2^{n-1} - 1) \) adder, and one \( (n-1) \)-bit CPA 2 with an EAC are
used to obtain the summation $v_3$. After OPU 3 performs the operations in Eqs. (13) and (14), a $(2n - 1)$-bit CPA 3 is used to complete $(P - v_3)$, and $x_1$ is then cascaded to output the final $X$. The total hardware cost of the proposed reverse converter is listed in Table II.

![Diagram of the proposed converter architecture](image)

### Fig. 1. Proposed converter architecture for $\{2^{n+k}, 2^n - 1, 2^{n-1} - 1\}$

### Table II. Hardware cost for the proposed converter

| Moduli set | Delay | Hardware cost |
|------------|-------|---------------|
| Proposed with $(2^n + 1)$ free $\{2^{n+k}, 2^n - 1, 2^{n-1} - 1\}$ | $(6n + 1)FA + 3NOT$ | $(4n + 2k - 2)FA + (4n - 2k - 4)HA + (4n + 2k - 1)HA$ |
3 Moduli set evaluation and experimental results

For the moduli set evaluation, a 16-tap FIR filter was implemented using the proposed moduli set, \(\{2^n - 1, 2^n, 2^n + 1\}\) [4], \(\{2^n + 1, 2^{n+k}, 2^n - 1\}\) [6], and a binary number system. The architecture of the binary-based FIR [13] does not include a forward or reverse converter. The architecture of the RNS-based FIR is shown in Fig. 2, which has three-stage blocks. The forward converter is implemented on the basis of combinational logic [12]. The design of the FIR module with a mod function was derived from [13]. The reverse converter depends on the design complexity of the moduli set.

For a fair comparison, all designs with RNS-based and binary-based FIRs were implemented using Verilog code and synthesized using TSMC 90-nm CMOS processor technology. Fig. 3 shows a comparison of the minimal system delays. RNS-based designs include a three-stage delay distribution. The bottom of the figure shows the optimal values of parameters \(n\) and \(k\) under DR requirements. The two adaptable RNS-based FIR designs attained a shorter delay than did the fixed RNS- and binary-based FIR designs. Moreover, the proposed adaptable design was faster than the other adaptable work [6], because the proposed moduli set without a \(2^n + 1\) channel achieved a shorter delay in the forward converter and FIR module. For an overall system comparison, the proposed system achieved an up to 22% saving compared with those proposed in [4] and [6] and the binary-based design. In addition, the channel delay balance of the parallel FIR modules was analyzed; Table III lists the fastest and slowest module channels, and Fig. 4 shows the corresponding delay results. The proposed moduli set achieved more efficient balance with a 16%–26% delay overhead.

Under the same frequency (25 MHz) and area constraints, Fig. 5(a) depicts a synthesized area comparison, showing that the FIR module requires at least 97% of the layout area. The proposed FIR module has the lower hardware cost, with a 12% area saving compared with the designs in [4] and [6]. In general, the hardware cost of the RNS-based FIR is higher than that of the binary-based FIR. However, the proposed design has an area overhead of only 10.7% at most. Fig. 5(b) shows a power comparison. Similarly, the FIR module requires the most system power. The proposed module attained the lowest power consumption and achieved a power saving of up to 28%. Regarding the \(AD^2\) (area \(\times\) delay\(^2\)) comparison, Fig. 6 shows that the proposed system can achieve a 12%–46% saving. Finally, the design rules...
and layout versus the schematic were confirmed. The 64-bit chip layout result is depicted in Fig. 7.

![Figure 7](image1.png)

**Fig. 7.** Minimum delay of the stage and system delay comparison

| Moduli set | Fastest channel | Slowest channel |
|------------|-----------------|-----------------|
| $\{2^n, 2^n - 1, 2^n - 1\}$ | $2^{(n-1)} - 1$ | $2^{(n+k)}$ |
| $[6] \{2^n + 1, 2^n + k, 2^n - 1\}$ | $2^n - 1$ | $2^n + 1$ |

![Table III. Fastest and slowest module channels](image2.png)

**Fig. 4.** Fastest and slowest channel delay distribution for the balance analysis

![Image](image3.png)

(a) System area comparison and three-stage area distribution

(b) System power comparison and three-stage power distribution

**Fig. 5.** RNS- and binary-based FIR system synthesis results
4 Conclusion

This paper presents a new adaptable three-moduli set \( \{2^n+k, 2^n−1, 2^n−1−1\} \) and its converter design. The proposed moduli set improved the channel delay imbalanced and avoided \((2^n+1)\) modulo defects. The set was employed to implement a RNS-based FIR system. The layout synthesis results show that it improved the area, delay, and power savings compared with those of related three-moduli sets. Furthermore, compared with binary-based designs, the proposed design also improved the delay and power savings. Overall, the proposed design achieved a 12%–46% \(AD^2\) saving compared with the RNS- and binary-based FIR systems. Finally, a 64-bit VLSI chip layout for the proposed design was implemented and demonstrated. The proposed moduli set is suitable for RNS-based DSP system designs.

Acknowledgments

This study was partially supported by the National Science Council, Taiwan, R.O.C. (MOST 104-2220-E-224-001, and 104-2221-E-224-040). The very-large-scale integration design tools were provided by the National Chip Implementation Center.