International Journal of Advance Research in Science and Engineering
Volume No.06, Issue No. 09, September 2017
www.ijarse.com
IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

High Efficient Sign detector for Residue Number System

Meer Amer Ali[1], D. Divya Reddy[2]

[1] M.Tech (DSCE), [2]Assistant Professor (ECE), Sri Visvesvaraya Institute of Technology & Science, Chowderpally, Devarkadra, Mahabubnagar

ABSTRACT

The moduli set {2n − 1, 2n, 2n + 1} has been commonly used in residue number system (RNS)-based calculations. Its sign extraction problem, even though fundamentally important in magnitude comparison and other difficult algorithms in RNS, has received considerably less attention than its scaling and reverse conversion problems. This paper presents a new algorithm for the design of a fast adder-based sign detector. The circuit is greatly simplified by minimizing the dynamic range to eliminate large modulo operations with the help of the new Chinese remainder theorem. Our synthesis results show that the proposed design outperforms all the existing adder-based sign detectors reported for this moduli set in area and speed.

Keywords: Chinese remainder theorem (CRT), computer arithmetic, residue number system (RNS).

I. NTRODUCTION

Residue number system (RNS) is gaining enhancing popularity in the VLSI implementation of application-specific digital signal processors (DSPs). This is in part due to its ability to accelerate and to decrease the power consumptions of crucial and frequently used data path operations by subword-level parallelism and modularity, and in part due to the ease of realizing modulo operations using the moduli of the forms 2^n and $2^n \pm 1$. Modular $2^n \pm 1$. arithmetic properties have been exploited with arithmetic structures, such as diminished-1, sparse carry chain, Kogge–Stone adder, and so on, to decrease the implementation complexity of modulo addition, subtraction, and multiplication for these special moduli to an extent that is comparable with their two's complement number system counterparts. These improvements have given rise to the extensive use and continual successes in developing the balanced three moduli set $\{2^n - 1, 2^n, 2^n + 1\}$ for the implementation of many new and existing DSP algorithms, including fast Fourier transform, discrete wavelet transform, finite and infinite impulse response filters, and digital image processing. In fact, the difficulties associated with the implementation of nonmodular operations, such as scaling and reverse conversion from residue-to-binary representation, have largely been resolved for this three moduli set.

Even though the hardware efficiency of its individual residue arithmetic operations, as well as its forward and reverse converters, some fundamental operations, such as sign detection, magnitude comparison, and overflow detection, for this moduli set remain slow and expensive. These operations are complicated to parallelize as they need the combination of multiple residue values to calcilate. To reduce the computational complexity, lookup tables are often used to store the precalculated orthogonal projections of the numbers of interest. Unfortunately, memory-based ways are difficult to pipeline. The size and number of lookup tables, as well as their access time also grow with the size of the moduli. Among these operations, sign detection is an example of a less focused

problem for this moduli set. Despite its importance as a preprocessing operation and an integral component of other intermodulo operations like magnitude comparison and overflow detection, only a handful of solutions are found in the literature.

A general theorem for sign detection in residue domain is presented, where the magnitude of an integer is first decoded from its residue representations by converting the residues into its equivalent binary representation to find the halfway point of the dynamic range. The large modulo operation in the reverse conversion is decreased by the mixed radix conversion (MRC) and to a modulo-two sum by using the fractional binary representation. Both the implementations are based on ROMs, which suffer from the aforementioned deficiencies. The most recent RNS sign detectors were designed for $\{2^n - 1, 2^n, 2^{n+1} - 1\}$. Although very efficient standalone, its use is limited as the efficient reverse converter, and scaler needed for a complete system implementation has not been reported for this new moduli set. The only adder-based sign detection circuits for $\{2n - 1, 2n, 2n + 1\}$ identify the sign by either full or partial reverse conversion into the binary domain using the new Chinese remainder theorem, or through the mixed radix coefficients.

In this paper, an alternative efficient sign detection algorithm for $\{2^n - 1, 2^n, 2^n + 1\}$ is proposed. The proposed technique develops the new CRT-I to greatly simplify the $2^{2n} - 1$ scaling of residue representation into addends that can be readily achieveed by circular left shifted residues at no logic cost. The reduced dynamic range enables the sign of an integer to be calculated directly from the most significant bit (MSB) of the scaled residues with a heavily strippeddown version of a reverse converter. Another benefit of our proposed sign detector is that it can also be used as a scaler.

II. PRELIMINARIES AND NOTATIONS

RNS is characterized by a set of N coprime numbers, called the moduli set $\{m1, m2, \ldots, mN\}$, i.e., $GCD(mi, mj) = 1 \forall i \neq j$. Any integer X can be represented by an N-tuple $(x1, x2, \ldots, xN)$ in this moduli set. Each residue x_i is the least nonnegative remainder calculated by dividing X by the modulus mi, which can be expressed mathematically as $x_i = |X|m_i$ for $i = 1, 2, \ldots, N$. The product of all moduli is called the dynamic range M, i.e., $M = \_N\ i{=}1\ m_i$. Any integer X that lies within $0 \leq X < M$ will have a unique residue representation.

An integer X within the dynamic range can be recovered from its residue representation $(x1, x2, \ldots, xN)$ by applying the CRT

$$|X = \left| \sum_{i=1}^{N} M_i \, | |M_i^{-1}|_{m_i} x_i |_{m_i} \right|_M \quad \ldots\ldots(1)$$

where $M_i = M/m_i$ and $|M{-}1_i|m_i$ is the multiplicative inverse of $|M_i|m_i$.

To represent a signed integer $\hat{X}$ in RNS, M is divided into two symmetrical half ranges for the representation of positive and negative integers. When M is even, the range of signed integers that can be definitely represented in RNS is $[-M/2, M/2 - 1]$. Correspondingly, for odd M, the range of definitely representable signed integers in RNS is $[-(M - 1)/2, (M - 1)/2]$. The signed integer $\hat{X}$ can be represented using the same residue representation as an unsigned integer X for the same moduli set. The relationship between $\hat{X}$ and X is given as follows:

$$\hat{X} = \left| \left\lfloor X + \frac{M}{2} \right\rfloor \right|_M - \left\lfloor \frac{M}{2} \right\rfloor \quad \ldots\ldots(2)$$

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

When $\hat{X} \geq 0$, the residue representation of X can be mapped to that of $\hat{X}$ in the range of $[0, M/2 - 1]$ if M is even and $[0, (M - 1)/2]$ if M is odd. In a similar way, when $\hat{X} < 0$, the residue representation of X can be mapped to that of $\hat{X}$ in the range of $[M/2, M - 1]$ if M is even and $[(M + 1)/2, M - 1]$ if M is odd. Thus, the sign of $\hat{X}$ can be identified as follows.

When M is even

$$\text{sign}(\hat{X}) = \begin{cases} 0, & \text{if } X \in [0, (M/2) - 1] \\ 1, & \text{if } X \in [M/2, M - 1] \end{cases} \quad \ldots\ldots(3)$$

When M is odd

$$\text{sign}(\hat{X}) = \begin{cases} 0, & \text{if } X \in [0, (M - 1)/2] \\ 1, & \text{if } X \in [(M + 1)/2, M - 1] \end{cases} \quad \ldots\ldots(4)$$

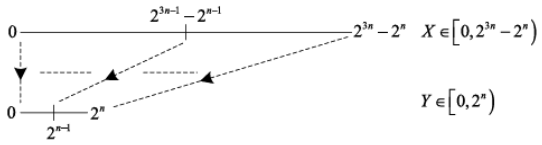

Fig. 1. Mapping of the half ranges of integer X in [0, M) to the half ranges of its scaled integer Y in [0, M').

Properties 1 and 2 are employed in order to simplify some arithmetic operations in the derivation of our proposed sign detection circuit for RNS $\{2^n - 1, 2^n, 2^n + 1\}$.

Property 1: The modulo $2^n - 1$ multiplication of an n-bit binary number x and r exponent of two is equivalent to a circular left shift (CLS) of the binary bits of x by r positions

$$|2^r x|_{2^n - 1} = \text{CLS}_n(x, r) \quad \ldots\ldots(5)$$

where CLS$n(x, r)$ represents the circular shift of an n-bit binary number x by r bits to the left.

Property 2*:* As a corollary of Property 1

$$|-2^r x|_{2^n - 1} = |2^r (2^n - 1 - x)|_{2^n - 1} = |2^r \bar{x}|_{2^n - 1} = \text{CLS}_n(\bar{x}, r) \quad \ldots\ldots(6)$$

Where $-x$ is the one's complement of integer x.

III. PROPOSED SIGN DETECTION ALGORITHM

Let (x_1, x_2, x_3) be the residue representation of an integer X with respect to the moduli set $\{m_1, m_2, m_3\} = \{2^n - 1, 2^n, 2^n + 1\}$. Since the dynamic range M of this moduli set can be factored into 2^n and $2^{2n} - 1$, the sizes of the modulo operations required for identifying the sign of $\hat{X}$ from its equivalent residue representation of X can be substantially decreased by scaling (x_1, x_2, x_3) in the residue domain by $2^{2n} - 1$. This will map thes lower half range $[0, 2^{3n-1} - 2^{n-1})$ of X to the lower half range $[0, 2^{n-1})$ of the scaled integer Y and the upper half range $[2^{3n-1} - 2^{n-1}, 2^{3n} - 2^n)$ of X to the upper half range $[2^{n-1}, 2^n)$ of Y , as shown in Fig. 1. By minimizing the dynamic range from $M = 2^{3n} - 2^{2n}$ to $M' = 2^n$, its half range can be easily identified from the MSB of the scaled integer Y. This new concept of sign detection in $\{2^n - 1, 2^n, 2^n + 1\}$ can be made very efficient provided that scaling by $2^{2n} - 1$ as well as the reverse conversion of the scaled residues into Y can be computed efficiently from the residues x_1, x_2, and x_3. As only the MSB of Y is needed for the sign detection of $\hat{X}$, a full reverse conversion from (x_1, x_2, x_3) is not needed.

To simplify the scaling by $22n - 1$ in the residue domain, the new CRT, also called CRT-I, is used to convert X into a weighted sum of its residues modulo $22n - 1$. Corresponding to CRT-I

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

$$X = x_3 + m_3|k_1(x_1 - x_3) + k_2 m_1(x_2 - x_1)|_{m_1 m_2} \quad \ldots\ldots(7)$$

where $k_1 m_3 = |1|_{m_1 m_2}$ and $k_2 m_3 m_1 = |1|_{m_2}$.

With $m_1 = 2^n - 1$, $m_2 = 2^n$, and $m_3 = 2^n + 1$, we have

$$X = x_3 + (2^n + 1)|k_1(x_1 - x_3) + k_2(2^n - 1)(x_2 - x_1)|_{2^n(2^n - 1)} \quad \ldots\ldots(8)$$

It can be proved that the multiplicative inverses of $|2^n + 1|_{2^n(2^n-1)}$ and $|2^{2n} - 1|_{2^n}$ are given by $k1 = 2^{2n-1} - (2^n - 1)$ and $k2 = -1$, respectively. These closed form expressions of $k1$ and $k2$ are proved as follows.

Proof of $k1 = 2^{2n-1} - (2^n - 1)$:

$$\begin{aligned} k_1(2^n + 1)|2^n(2^n - 1) &= |[2^{2n-1} - (2^n - 1)](2^n + 1)|2^n(2^n - 1) \\ &= |2^{2n-1}(2^n + 1) - (2^{2n} - 1)|2^n(2^n - 1) \\ &= |2^{3n-1} - 2^{2n-1} + 1|2^n(2^n - 1) \\ &= |2^{2n-1}(2^n - 1) + 1|2^n(2^n - 1) = 1. \end{aligned}$$

Proof of $k_2 = -1$:

$$|k_2(2^{2n} - 1)|_{2^n} = |-1 \times (2^{2n} - 1)|_{2^n} = |-2^{2n} + 1|_{2^n} = 1$$

Substituting the values of $k1$ and $k2$ into (8), we have

$$\begin{aligned} X &= x_3 + (2^n + 1)\left|\begin{matrix}\{2^{2n-1} - (2^n - 1)\}(x_1 - x_3) \\ - (2^n - 1)(x_2 - x_1)\end{matrix}\right|_{2^n(2^n-1)} \\ &= x_3 + (2^n + 1)\left|\begin{matrix}2^{2n-1}(x_1 - x_3) \\ - (2^n - 1)(x_2 - x_3)\end{matrix}\right|_{2^n(2^n-1)} \quad \ldots\ldots(9) \end{aligned}$$

By scaling X by $2^{2n} - 1$, the scaled integer Y can be obtained by

$$Y = \left\lfloor \frac{X}{2^{2n} - 1} \right\rfloor = \left\lfloor \frac{x_3}{2^{2n} - 1} \right\rfloor + \left\lfloor \frac{(2^n + 1)\,Z}{2^{2n} - 1} \right\rfloor \quad \ldots\ldots(10)$$

where $Z = |2^{2n-1}(x_1 - x_3) - (2^n - 1)(x_2 - x_3)|_{2^n(2^n-1)}$.

Since $x_3 \in [0, 2^n]$, $x_3 < 2^{2n} - 1$. Therefore $[(x_3/2^{2n} - 1)] = 0$, and Y can be written as

$$\begin{aligned} Y &= \left\lfloor \frac{(2^n + 1)Z}{2^{2n} - 1} \right\rfloor = \left\lfloor \frac{Z}{2^n - 1} \right\rfloor \\ &= \left\lfloor \frac{|2^{2n-1}(x_1 - x_3) - (2^n - 1)(x_2 - x_3)|_{2^n(2^n-1)}}{(2^n - 1)} \right\rfloor \quad \ldots\ldots(11) \end{aligned}$$

As $[(|x|m1m2 /m1)] = |[(x/m1)]|m2$ from [11], (11) can be rewritten as

$$\begin{aligned} Y &= \left|\left\lfloor \frac{2^{2n-1}(x_1 - x_3) - (2^n - 1)(x_2 - x_3)}{(2^n - 1)} \right\rfloor\right|_{2^n} \\ &= \left|\left\lfloor \frac{2^{2n-1}(x_1 - x_3)}{2^n - 1} \right\rfloor + x_3 - x_2\right|_{2^n} \quad \ldots\ldots(12) \end{aligned}$$

Let $H = 2^{2n-1}(x_1 - x_3)$. Since $H = m[(H/m)] + |H|_m$ for any integer H and m, we have

$$H = (2^n - 1)\left\lfloor \frac{H}{2^n - 1} \right\rfloor + |H|_{2^n - 1} \quad \ldots\ldots(13)$$

Taking mod 2^n operation on both the sides of (13), we have

$$|H|_{2^n} = \left|(2^n - 1)\left\lfloor \frac{H}{2^n - 1} \right\rfloor\right|_{2^n} + ||H|_{2^n - 1}|_{2^n} \quad \ldots\ldots(14)$$

Since $|H|_2 n = |2^{2n-1}(x_1 - x_3)|_{2^n} = 0$ and $|2^n - 1|_2 n = -1$

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

$$\left\| \left\lfloor \frac{H}{2^n - 1} \right\rfloor \right\|_{2^n} = \|H|_{2^n-1}|_{2^n} \quad \ldots\ldots(15)$$

Substituting (15) into (12), we have

$$\begin{aligned} Y &= \|H|_{2^n-1} + x_3 - x_2|_{2^n} \\ &= \|2^{2n-1}(x_1 - x_3)|_{2^n-1} + x_3 - x_2|_{2^n} \quad \ldots\ldots(16) \end{aligned}$$

If $Y \in [0, 2^{n-1})$, X falls in the lower half range of M and (x_1, x_2, x_3) denotes a positive integer, i.e., $X\hat{} \geq 0$. Otherwise, if $Y \in [2^{n-1}, 2^n)$, X falls in the upper half range of M and (x_1, x_2, x_3) represents a negative integer, i.e., $X\hat{} < 0$.

3.1 HARDWARE IMPLEMENTATIONS

The residues $x1$, $x2$, and $x3$ can be denoted in a binary form as $x_1 = x_{1,n-1}x_{1,n-2} \ldots x_{1,0}$, $x_2 = x_{2,n-1}x_{2,n-2} \ldots x_{2,0}$ and $x_3 = x_{3,n}\, x_{3,n-1} \ldots x_{3,0}$, respectively, where $x_{i,j}$ represents the j th bit of the residue x_i . The binary vectors of $x1$ and $x2$ are of n bits but the binary vector of x_3 is of $n + 1$ bits. In (16), one of the terms in the modulo $2^n - 1$ sum involves the operation $|-2^{2n-1}\, x_3|_{2^n-1}$, which cannot be directly implemented by Property 2, since x_3 has $n+1$ bits. To apply the CLS property on the one's complement of x_3 as in (6), x_3 is expressed as $x_3 = 2^n x_{3,n} + x_{3,n-1}x_{3,n-2} \ldots x_{3,0}$. Since $|2^n\, x_{3,n}|_{2^n-1} = x_{3,n}$, the MSB $x_{3,n}$ of x_3 can be logically OR with $x3,0$ to form an n-bit binary vector $x'_3 = |x_3|_{2^n-1} = |x_{3,n-1}x_{3,n-2} \ldots x'_{3,0}|_{2^n-1}$, where $x'_{3,0} = x_{3,0} \vee x_{3,n}$ and $\vee$ denotes a logical OR operator. $|H|_{2^n-1}$ in (16) can then be implemented using the CLS operations of Properties 1 and 2 to obtain

$$Y = \|u_1 + u_2|_{2^n-1} + x_3 - x_2|_{2^n} \quad \ldots\ldots(17)$$

where

$$u_1 = |2^{2n-1}x_1|_{2^n-1} = \mathrm{CLS}_n(x_1, 2n-1) = \underbrace{x_{1,0}}_{1}\,\underbrace{x_{1,n-1} \ldots x_{1,1}}_{n-1} \quad \ldots\ldots(18)$$

$$u_2 = |2^{2n-1}\bar{x}_3|_{2^n-1} = \mathrm{CLS}_n(\bar{x}'_3, 2n-1) = \underbrace{\bar{x}'_{3,0}}_{1}\,\underbrace{\bar{x}_{3,n-1} \ldots \bar{x}_{3,1}}_{n-1} \quad \ldots\ldots(19)$$

The term $|u_1 + u_2|_{2^n-1}$ can be expressed as

$$|u_1 + u_2|_{2^n-1} = \begin{cases} |u_1 + u_2 + 1|_{2^n}, & \text{if } u_1 + u_2 \geq 2^n - 1 \\ u_1 + u_2, & \text{otherwise} \end{cases} \quad \ldots\ldots(20)$$

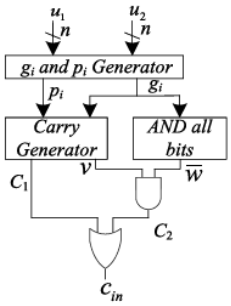

Fig. 2. Generation of carry-in signal c_{in}

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

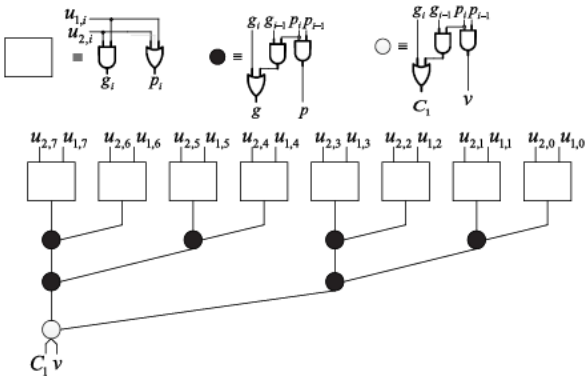

Fig. 3. Example of the generation of the carry signal C_1 and v for n = 8

Hence, $||u_1 + u_2|2^n - 1|2^n = |u_1 + u_2 + c_{in}|2^n$, where $c$in ∈ {0, 1}. As $|-x_2|2^n = 2_n - x^2 = \overline{x}_2 + 1$, (17) can be written as

$$Y = |u_1 + u_2 + c_{\text{in}} + x_3 + \bar{x}_2 + 1|_{2^n} \quad \ldots\ldots(21)$$

The generation of the carry-in signal c_{in} is shown in Fig. 2. The condition $u_1 + u_2 \geq 2^n$ is detected by $C_1 = 1$ and the signal C_1 can be generated by parallel prefix operators. As an example, the carry signal C_1 for $n = 8$ can be generated by the circuit shown in Fig. 3. The condition $u_1 + u_2 = 2^n - 1 = \{11 \ldots 11\}$n can be identified by $C_2 = 1$. C_2 is generated by $\overline{w} \wedge v$, where $w = \wedge_{i=0}^{n-1} g_i$ and $v = p_{n-1}{:}0 = \wedge_{i=0}^{n-1} p_i$, where ∧ denotes a logical AND operator. The signals g_i and $p_{n-1:0}$ have already been generated in the computation of C1. Accordingly, the condition $u_1 + u_2 \geq 2^n - 1$ for $c$in = 1 can be detected by

$$c_{\text{in}} = C_1 \vee C_2 \ldots\ldots(22)$$

The two addends, u2 and x3, in (21) can be further simplified as follows:

$$\begin{aligned}
|u_2 + x_3|_{2^n} &= ||2u_2|_{2^n} - u_2 + |x_3|_{2^n}|_{2^n} \\
&= \left| \underbrace{\bar{x}_{3,n-1}\bar{x}_{3,n-2}\ldots\bar{x}_{3,1}0}_{n} + |x_3|_{2^n} - u_2 \right|_{2^n} \\
&= \left| \underbrace{\bar{x}_{3,n-1}\bar{x}_{3,n-2}\ldots\bar{x}_{3,1}\bar{x}_{3,0}}_{n} - \bar{x}_{3,0} + |x_3|_{2^n} - u_2 \right|_{2^n} \\
&= ||\bar{x}_3|_{2^n} + |x_3|_{2^n} - u_2 - \bar{x}_{3,0}|_{2^n} \\
&= |2^n - 1 - u_2 - \bar{x}_{3,0}|_{2^n} = |\bar{u}_2 - \bar{x}_{3,0}|_{2^n} \\
&= |x'_{3,0}x_{3,n-1}x_{3,n-2}\ldots x_{3,1} - \bar{x}_{3,0}|_{2^n} \qquad \ldots\ldots(23)
\end{aligned}$$

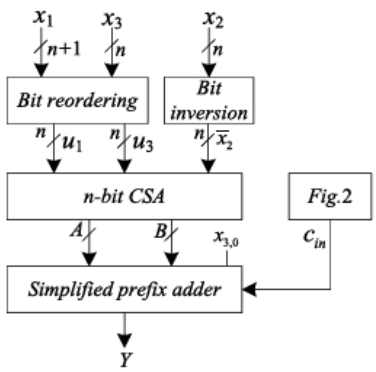

Fig. 4. Proposed sign detection architecture for {$2^n - 1$, 2^n, 2^n+1}

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

When $x_{3,n} = 0$, $x'_{3,0} = x_{3,0} \vee 0 = x_{3,0}$. Then $|u_2 + x_3|_{2^n} = |x_{3,0}x_{3,n-1}x_{3,n-2}\ldots x_{3,1} - \bar{x}_{3,0}|_{2^n}$ (24)

When $x_{3,n} = 1$, since $x_3 \in [0, 2n]$, $x_{3,n-1}x_{3,n-2}\ldots x_{3,0} = 00\ldots0$. Hence, $x'_{3,0} = x_{3,0} \vee x_{3,n} = 1$ and

$$\begin{aligned} |u_2 + x_3|_{2^n} &= \left|\underbrace{100\ldots0}_{n} - 1\right|_{2^n} = \underbrace{011\ldots1}_{n} \\ &= \left|\underbrace{x_{3,0}x_{3,n}x_{3,n}\ldots x_{3,n}}_{n} - \bar{x}_{3,0} + x_{3,n}\right|_{2^n} \\ &= \left|\underbrace{x_{3,n}00\ldots0}_{n} - \bar{x}_{3,0}\right|_{2^n}. \end{aligned} \quad \ldots\ldots(25)$$

To satisfy both (24) and (25)

$$|u_2 + x_3|_{2^n} = |u_3 - \bar{x}_{3,0}|_{2^n} \quad \ldots\ldots(26)$$

where the n-bit binary vector u_3 is given by

$$u_3 = (x_{3,0} \vee x_{3,n})x_{3,n-1}x_{3,n-2}\ldots x_{3,1} \quad \ldots\ldots(27)$$

Substituting (26) into (21), we have

$$Y = |u_1 + u_3 + \bar{x}_2 + c_{\text{in}} + 1 - \bar{x}_{3,0}|_{2^n} \quad \ldots\ldots(28)$$

If $x_{3,0} = 1$, $1 - \overline{x}_{3,0} = 1$, and if $x_{3,0} = 0$, $1 - \overline{x}_{3,0} = 0$. Hence, the term $1 - \overline{x}_{3,0}$ in (28) can be replaced by $x_{3,0}$ and

$$Y = |u_1 + u_3 + \bar{x}_2 + c_{\text{in}} + x_{3,0}|_{2^n} \quad \ldots\ldots(29)$$

The sign of $\hat{X}$ can be identified by the MSB of Y. An n-bit carry save adder (CSA) can be used to add the three n-bit operands, u_1, u_3, and $\overline{x}_2$, to produce an n-bit sum $A = a_{n-1}a_{n-2}\ldots a_0$ and an n-bit carry vector $B = b_{n-1}b_{n-2}\ldots b_1b_0$. Due to the modulo $2n$ addition, the final carry output bit bn of the CSA need not be generated. As $b0 = 0$, it can be replaced by $x3,0$ of (29) before the MSB of Y is calculated by a simplified parallel prefix adder of A and B with the input carry bit $c_{-1} = c_{\text{in}}$. The prefix adder is simplified by keeping only the carry generation network for the computation of carry signal c_{n-1}, from which the sign of $\hat{X}$ can be identified by $\text{sign}(\hat{X}) = a_{n-1} \oplus b_{n-1} \oplus c_{n-1}$. The architecture of the proposed sign detector is shown in Fig. 4, where the circuit diagram for the simplified prefix adder is depicted in Fig. 5 for $n = 8$.

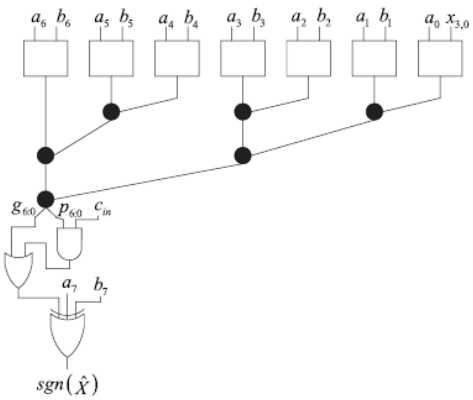

Fig. 5. Simplified prefix adder for n = 8

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

u_1		0	1	1	1	1
u_3		0	1	1	0	1
$\overline{x}_2$	+	1	1	0	0	0
A		1	1	0	1	0
B		1	1	0	1	$0 = x_{3,0}$
A		1	1	0	1	0
B		1	1	0	1	0
c_{in}						1
Y		1	0	1	0	1

Fig. 6. Computation of *Y* for Example 1

Example 1: For $n = 5$, $\{m_1, m_2, m_3\} = \{31, 32, 33\}$, $M = 31 \times 32 \times 33 = 32\,736$, and $M/2 = 16\,368$. The signed integer $\hat{X} = -11\,161$ can be denoted by the residue representation $(x_1, x_2, x_3) = (30, 7, 26)$ according to the unsigned integer $X = 21\,575$ in the same moduli set. The binary representation of the residues are $x_1 = 111102$, $x_2 = 001112$, and $x_3 = 0110102$. According to (18), (19), and (27), $u_1 = 011112$, $u_2 = 100102$, and $u_3 = 011012$. Also, $x_{3,0} = 0$. Since $u_1 + u_2 = 01111 + 10010 = 33 > 32$, $C1 = 1$. Since 33 _= 31, $C2 = 0$. According to (22), $c_{in} = C_1 \vee C_2 = 1$. The calculation of Y in (29) is illustrated in Fig. 6. Since MSB of $Y = 1$, the integer $\hat{X}$ represented by (30, 7, 26) is negative.

IV. SYNTHESIS AND SIMULATION RESULTS

The proposed razor latch is designed with the XILINX ISE 14.5 simulation tool and implemented with Verilog HDL. The RTL diagram and simulation results are displayed below.

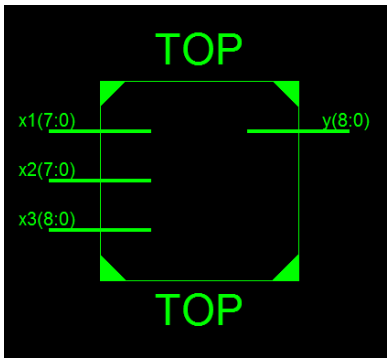

Fig: Top level schematic diagram

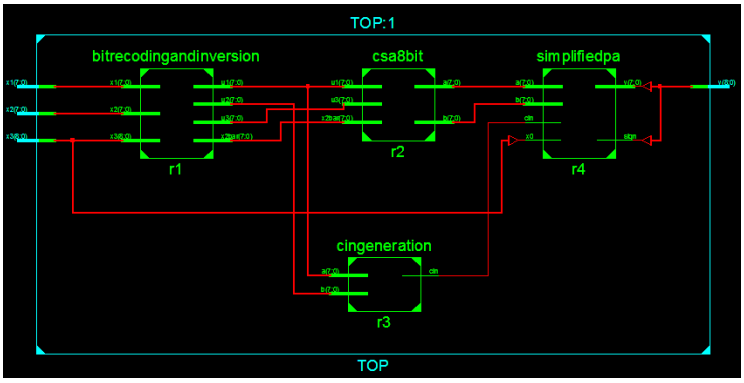

Fig: Internal architectures of RTL diagram

TOP Project Status			
Project File:	signdetector.xise	**Parser Errors:**	X 1 Error
Module Name:	TOP	**Implementation State:**	Synthesized
Target Device:	xc7z010-2clg400	**• Errors:**	No Errors
Product Version:	ISE 14.5	**• Warnings:**	2 Warnings (2 new)
Design Goal:	Balanced	**• Routing Results:**	
Design Strategy:	Xilinx Default (unlocked)	**• Timing Constraints:**	
Environment:	System Settings	**• Final Timing Score:**	

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice LUTs	33	17600	0%
Number of fully used LUT-FF pairs	0	33	0%
Number of bonded IOBs	34	100	34%

Fig: Synthesis report

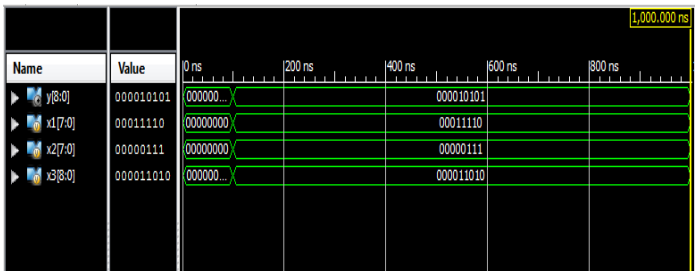

Fig: Simulation result

VI. CONCLUSION

In this paper, an efficient fast sign detection algorithm for the residue number system moduli set $\{2^n-1, 2^n, 2^n+1\}$ is presented. The proposed algorithm which allows parallel implementation and include modulo 2n additions. Based on existing sign detection algorithm, an efficient sign detection algorithm is proposed. The sign detection unit can be implemented using one carry save adder, one comparator and one prefix adder. Here efficiency achieved is better than other algorithm for sign detection. Adder based sign detectorwas designed by the Verilog HDL synthesized in Xilinx ISE 14.5.

VII. FUTURE SCOPE

Arithmetic designs are developed and holded with these proposed methods of reverse converters in RNS formulation. Finally with these conditional procure positions of adders obtain the eventual levels of area, power and performance.

REFERENCES

[1] R. Muralidharan and C.-H. Chang, "Area-power efficient modulo $2n-1$ and modulo $2n+1$ multipliers for $\{2n-1, 2n, 2n+1\}$ based RNS," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 59, no. 10, pp. 2263–2274, Oct. 2012.

[2] H. T. Vergos and G. Dimitrakopoulos, "On modulo $2n+1$ adder design," *IEEE Trans. Comput.*, vol. 61, no. 2, pp. 173–186, Feb. 2012.

[3] H. T. Vergos, "A family of area-time efficient modulo 2*n* +1 adders," in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI*, Lixouri, Kefalonia, Greece, Jul. 2010, pp. 442–443.

[4] L.-S. Didier and L. Jaulmes, "Fast modulo 2*n*−1 and 2*n*+1 adder using carry-chain on FPGA," in *Proc. Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, USA, Nov. 2013, pp. 1155–1159.

[5] N. S. Szabó and R. I. Tanaka, *Residue Arithmetic and Its Applications to Computer Technology*. New York, NY, USA: McGraw-Hill, 1967.

[6] R. Conway and J. Nelson, "Improved RNS FIR filter architectures," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 51, no. 1, pp. 26–28, Jan. 2004.

[7] E. Vassalos, D. Bakalis, and H. T. Vergos, "RNS assisted image filtering and edge detection," in *Proc. IEEE 18th Int. Conf. Digit. Signal Process.*, Fira, Santorini, Greece, Jul. 2013, pp. 1–6.

[8] J.-C. Bajard, L.-S. Didier, and T. Hilaire, "ρ-direct form transposed and residue number systems for filter implementations," in *Proc. IEEE 54th Int. Midwest Symp. Circuits Syst.*, Seoul, Korea, Aug. 2011, pp. 1–4.

[9] Y. Liu and E. M.-K. Lai, "Design and implementation of an RNS-based 2-D DWT processor," *IEEE Trans. Consum. Electron.*, vol. 50, no. 1, pp. 376–385, Feb. 2004.

[10] T. F. Tay, C.-H. Chang, and J. Y. S. Low, "Efficient VLSI implementation of 2*n* scaling of signed integer in RNS {2*n*−1, 2*n*, 2*n*+1}," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 21, no. 10, pp. 1936–1940, Oct. 2013.

AUTHOR DETAILS

	**MEER AMER ALI**, pursuing M.Tech (DSCE) from Sri Visvesvaraya Institute Of Technology & Science, Chowderpally (Vill), Devarkadra (Mdl), Mahabubnagar (Dist), TS, INDIA.
	**D. DIVYA REDDY**, working as Assistant professor (ECE) from Sri Visvesvaraya Institute Of Technology & Science, Chowderpally (Vill), Devarkadra (Mdl), Mahabu nagar (Dist), TS, INDIA.