International Journal of Soft Computing 4 (3): 127-130, 2009
ISSN: 1816-9503

A Memoryless MRC Technique for RNS-to-Binary Conversion Using The Moduli Set (2^n, 2^n-1, 2^{n-1}-1)

Kazeem Alagbe Gbolagade
Department of Applied Maths and Computer Science,
University for Development Studies, Navrongo, Ghana

Abstract: In this study, we investigate Residue Number System (RNS) to binary conversion, which is an important issue concerning the utilization of RNS numbers in Digital Signal Processing (DSP) applications. We present a Mixed Radix Conversion (MRC) technique for efficient RNS to binary conversion. First, we show that the computation of the required multiplicative inverses can be eliminated. Next, we propose an adder based RNS to binary converter, which requires mod-(2^n-1) or mod-(2^{n-1}-1) instead of mod-(2^n) (2^{n-1}-1) required by other state of the art Chinese Remainder Theorem (CRT) based equivalent converters. The proposed converter outperforms CRT based equivalent state of the art converters in terms of both speed and area. Consequently, due to the fact that our scheme operates on smaller magnitude operands, it results in less complex adders, which potentially results in faster implementation.

Key words: Residue number system, reverse conversion, chinese remainder theorem, mixed radix conversion, moduli selection, multiplicative inverses

INTRODUCTION

There is no ordering significance between the digits in Residue Number System (RNS). The result of any operation such as addition, subtraction or multiplication depends solely on the corresponding digits of its operands. This parallel property accounts for the inherent carry free property of the RNS, which makes RNS to be very useful in addition and multiplication dominated Digital Signal Processing (DSP) applications such as digital filtering, correlation, convolution, fast fourier transform and image processing. However, the following are the disadvantages of RNS: magnitude comparison, overflow detection, sign detection, moduli selection and data conversion. For RNS advantages not to be nullified by its disadvantages, an effective data converter is required. The research on residue to binary conversion is based on either the Chinese Remainder Theorem (CRT), (Van Vu, 1985; Elleithy and Bayoumi, 1992; Ahmad and Hoda, 1998; Wang, 1998; Ahmad *et al.*, 2003; Amir and Keivan, 2007; Gbolagade and Cotofana, 2008) or Mixed Radix Conversion (MRC) (Huang, 1983; Chakraborti *et al.*, 1986; Yassine, 1991, 1999; Yassine and Moore, 1991). CRT is desirable because the computation can be parallelized while MRC is by its very nature a sequential process. However, many RNS-Binary converters are based on MRC due to the complex and slow modulo-M operation (M being the system dynamic range thus a rather large constant) required by CRT. The major problem with the MRC is that the computations of the MR digits is done in a serial manner and requires a large number of arithmetic operations. In this study, we present a memoryless reverse converter. First, we show that the computation of the multiplicative inverses can be eliminated. Next, we employ shifting property to eliminate the involved multipliers. The proposed converter outperforms equivalent CRT based reverse converter in terms of both area and speed.

MATERIALS AND METHODS

RNS is defined in terms of a set of relatively prime integers $(m_i)_{i=1,n}$ such that gcd $(m_i, m_j) = 1$ for $i \neq j$, where gcd means greatest common divisor of m_i and m_j. For such a system

$$M = \prod_{i=1}^{n} m_i$$

is the dynamic range and any integer $X \in (0, M\text{-}1)$ can be uniquely represented as $X = (x_1, x_2, x_3, \ldots, x_n)$, where, $x_i = |X|m_i$, $0 \le x_i < m_i$. We note here in this study, that we use $x_i = |X|_{mi}$ to denote the X mod m_i operation and the operator $\otimes$ to represent the operation of addition, subtraction and multiplication. Given any 2 integer numbers K and L RNS represented by $K = (k_1, k2, k3, \ldots, k_n)$ and $L = (l_1, l_2, l_3, \ldots, l_n)$, respectively. $W = K \otimes L$, can be calculated as W = (w_1, w_2, w_3, …, w_n), where, $w_i = |k_i \otimes l_i|_{mi}$, for i = 1 to n. This actually means that the complexity of the calculation of the ⊗ operation is determined by the number of bits required to represent the residues and not by the one required to represent the input operands. Using MRC, the reverse conversion can be formulated as follows (Szabo and Tanaka, 1967). Suppose that an n-digit RNS number X = (x_1, x_2, x_3, …, x_n) with the set of relatively prime integer moduli set (m_1, m_2, m_3, …, m_n), determine the Mixed Radix Digits (MRD) (a_1, a_2, a_3, …, a_n) such that the Eq. 1 holds true:

$$X = a_1 + a_2m_1 + a_3m_1m_2 + \ldots + a_nm_1m_2m_3\ldots m_{n-1} \quad (1)$$

The MRD can be computed as follows Yassine and Moore (1991):

$$\begin{aligned}
a_1 &= x_1 \\
a_2 &= \left|(x_2 - a_1)\left|m_1^{-1}\right|_{m_2}\right|_{m_2} \\
a_3 &= \left|((x_3 - a_1)\left|m_1^{-1}\right|_{m_3} - a_2)\left|m_2^{-1}\right|_{m_3}\right|_{m_3} \qquad (2) \\
&\ldots \\
&\ldots \\
a_n &= \left|((\ldots(x_n - a_1)\left|m_1^{-1}\right|_{m_n} - a_2)\left|m_2^{-1}\right|_{m_n} - \ldots - a_{n-1})\left|m_{n-1}^{-1}\right|_{m_n}\right|_{m_n}
\end{aligned}$$

Apart from parallelizing the MRC technique, the usage of special moduli sets such as (2^n+1, 2^n, 2^n - 1), (2^n, 2^n - 1, 2^{n-1} - 1), (2n + 1, 2n, 2n - 1) etc., results in a further simplification of the MRC technique. In the following study, we simplify the MRC technique using the moduli set (2^n, 2^n - 12^{n-1} - 1).

PROPOSED ALGORITHM

Given the RNS number (x_1, x_2, x_3) with respect to the moduli set (m_1, m_2, m_3) in the form (2^n, 2^n-1, 2^{n-1}-1), the proposed algorithm computes the binary equivalent of this RNS number using MRC technique. First, we demonstrate that the computation of the multiplicative inverses can be eliminated for this moduli set. Next, we present a memory less MRC technique.

Theorem I: Given the moduli set (2^n, 2^n-12^{n-1}-1) with m_1 = 2^n, m_2 = 2^n-1, m_3 = 2^{n-1}-1, the following hold true:

$$\left|m_1^{-1}\right|_{m_2} = 1 \quad (3)$$

$$\left|m_2^{-1}\right|_{m_3} = 1 \quad (4)$$

$$\left|m_1^{-1}\right|_{m_3} = 2^{n-2} \quad (5)$$

Proof: If it can be demonstrated that $|m_1X_1|_{m2}$ = 1, then 1 is the multiplicative inverse of m_1 with respect to m_2.

$$\left|m_1X1\right|_{m_2} = \left|2^nX1\right|_{2^n-1} = \left|2^n\right|_{2^n-1} = 1$$

thus, Eq. 3 holds true.

In the same way, if it can be shown that $|m_2X_1|_{m3}$ = 1, then 1 is the multiplicative inverse of m_2 with respect to m_3. $|(2^n - 1)X1|_{2^{n-1}-1}$ putting y = 2^n, we obtain

$$\left|y-1\right|_{\frac{y}{2}-1} = \left|\left|y\right|_{\frac{y}{2}-1} - \left|1\right|_{\frac{y}{2}-1}\right|_{\frac{y}{2}-1} = \left|2-1\right|_{\frac{y}{2}-1} = 1$$

thus, Eq. 4 holds true.

Again, if it can be proved that $|m_1X2^{n-2}|_{m2}$ = 1, then 2^{n-2} is the multiplicative inverse of m_1 with respect to m_3. $|2^n(2^{n-2})|_{2^{n-1}-1}$, putting y = 2^n, we obtain

$$\left|2^{2n-2}\right|_{\frac{y}{2}-1} = \left|\frac{y^2}{4}\right|_{\frac{y}{2}-1} = 1$$

thus, Eq. 5 holds true.

Next, we assume a moduli set of length 3 for Eq. 2 and by substituting Eq. 3-5, we obtain the following expressions:

$$X = a_1 + a_2m_1 + a_3m_1m_2 \quad (6)$$

where:

$$\begin{aligned}
a_1 &= x_1 \\
a_2 &= \left|(x_2 - a_1)\right|_{m_2} \qquad (7) \\
a_3 &= \left|(2^{n-2}(x_3 - a_1) - a_2)\right|_{m_3}
\end{aligned}$$

In order to clarify this algorithm, let us convert $(7, 1, 0)_{RNS (8|7|3)}$ to decimal.

$$\begin{aligned}
a_1 &= 7 \\
a_2 &= \left|(1-7)\right|_7 = 1 \\
a_3 &= \left|2(0-7) - 1\right|_3 = 0
\end{aligned}$$

Hence,

$$X = 7+1X8+0X8X7 = 15$$

as it should.

In order to further reduce the hardware complexity, we use the following properties Amir and Keivan (2007):

- Modulo (2^p-1) multiplication of a residue number by 2^q, where, p and q are positive integers, is equivalent to q bit circular left shifting
- Modulo (2^p-1) of a negative number is accomplished by subtracting this number from (2^p-1). This is equivalent to taking one's compliment of the number

If the residues x_1, x_2 and x_3 have binary representations as Eq. 8-10:

$$x_1 = (x_{1,\,n-1}x_{1,\,n-2}\dots x_{1,\,1}x_{1,\,0}) \tag{8}$$

$$x_2 = (x_{2,\,n-2}x_{2,\,n-3}\dots x_{2,\,1}x_{2,\,0}) \tag{9}$$

$$x_3 = (x_{3,\,n-3}x_{3,\,n-4}\dots x_{3,\,1}x_{3,\,0}) \tag{10}$$

The multiplier in a_3 in Eq. 7 and of course all the multipliers in Eq. 6 can be eliminated by using the above properties. We start by showing the binary representation of a_1 and a_2.

$$a_1 = (x_{1,\,n-1}x_{1,\,n-2}\dots x_{1,\,1}x_{1,\,0})$$
$$a_2 = |u_1 + u_2|_{2^n-1}$$

where:

$$u_1 = (x_{2,\,n-2}x_{2,\,n-3}\dots x_{2,\,1}x_{2,\,0})$$
$$u_2 = -\,(x_{1,\,n-1}x_{1,\,n-2}\dots x_{1,\,1}x_{1,\,0})$$

Also, a_3 in Eq. 7 can be written as:

$$a_3 = |u_3 + u_4|_{2^n-1}$$

Using the properties stated above, we have

$$\begin{aligned} u_3 &= \left|2^{n-2}x_3\right|_{2^{n-1}-1} \\ &= \left|2^{n-2}(\underline{0}\underline{x_{3,n-3}x_{3,n-4\dots}x_{3,1}x_{3,0}})\right|_{2^{n-1}-1} \\ &= \underline{x_{3,0}}\underline{0x_{3,n-3}x_{3,n-4}\dots x_{3,1}} \end{aligned}$$

Also,

$$\begin{aligned} u_4 &= \left|-2^{n-2}x_1\right|_{2^{n-1}-1} \\ &= \left|2^{n-2} - (\underline{x_{1,n-1}x_{1,n-2\dots}x_{1,1}x_{1,0}})\right|_{2^{n-1}-1} \\ &= \left|2^{n-2}(\overline{x}_{1,n-1}\overline{x}_{1,n-2}\dots\overline{x}_{1,1}\,\overline{x}_{1,0})\right|_{2^{n-1}-1} \\ &= \overline{x}_{1,1}\overline{x}_{1,0}\,\overline{x}_{1,n-1}\,\overline{x}_{1,n-2}\dots \end{aligned}$$

Hardware implementation of the proposed residue to binary converter for the moduli set (2^n+1, 2^n, 2^{n-1}-1) is based on the computation of u_1, u_2, u_3 and u_4. It is made up of 3n-bit full adders together with modulo (2^n-1) and (2^{n-1}-1) adders which can be implemented using different methods. Using n-bit and (n-2)-bit one's compliment adders, the performance of the proposed converter can be significantly improved but this is left as a subject of future investigation.

RESULTS AND DISCUSSION

The proposed converter is a novel converter, which is dedicated to the moduli set (2^n, 2^n-1, 2^{n-1}-1). A converter, which is better than equivalent existing converters has been presented by Ahmad and Hoda (1998). Thus, we compare our proposal with this best state of the art equivalent converter. The converter presented by Ahmad and Hoda (1998) requires 1 adder, 3 subtractors and 1 comparator, whereas our proposal requires only three adders. In terms of area, our proposal is better than the best state of the art converter. Delay wise, the proposed converter requires (t_n + t_{n-1}) whereas the converter given by Ahmad and Hoda (1998) requires ($2t_{n-1}$ + t_{2n-1}). This implies that the proposal is faster than the best state of the art converter. Figure 1 depicts the implementation, the converter in Ahmad and Hoda (1998), while Fig. 2 shows the implementation of the proposed converter. Table 1 and 2, respectively, show the hardware requirements and performance comparison of this proposal and the one in Ahmad and Hoda (1998). Clearly, it can be seen that this proposal is better in terms of area and delay than the known best state of the art equivalent converter.

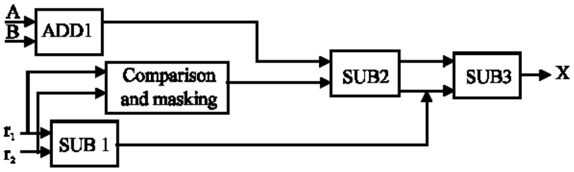

Fig. 1: Hardware implementation of Ahmad and Hoda (1998)

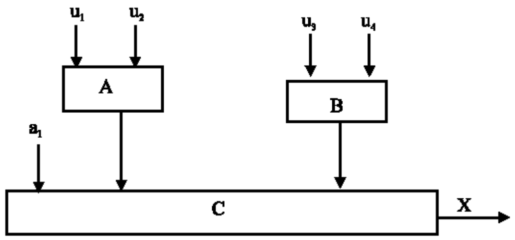

Fig. 2: Hardware implementation of the proposed converter

Table 1: Comparison of hardware requirements

Converter	No. adders	Total No. full adders
Ahmad and Hoda (1998)	5	6n-4
Proposed converter	3	3n

Table 2: Comparison of delay

Converter	Delay
Ahmad and Hoda (1998)	$2t_{n-1} + t_{2n-1}$
Proposed converter	$t_n + t_{n-1}$

CONCLUSION

In this study, we investigated RNS to binary conversion, which is an important issue concerning the utilization of RNS numbers in DSP applications. We presented an MRC technique for efficient RNS to binary conversion. First, we demonstrated that the computation of the required multiplicative inverses can be eliminated. Next, we proposed an adder based RNS to binary converter, which requires only 3, 2-1 adders and mod- $(2^n$-1) and mod-$(2^{n-1}$-1) instead of mod- (2^n) $(2^{n-1}$-1) required by other state of the art CRT based equivalent converters. Also, in terms of delay, this proposal requires $(t_n + t_{n-1})$. The proposed converter outperforms CRT based equivalent state of the art converters in terms of both speed and area. Consequently, due to the fact that our scheme operates on smaller magnitude operands, it results in less complex adders, which potentially results in faster implementation. The performance of the proposed converter can be significantly improved using n-bit and (n-2)-bit one's compliment adders but this is left as a subject of future investigation.

REFERENCES

Amir, S.M. and N. Keivan, 2007. New arithmetic residue to binary converters. Int. J. Comput. Sci. Eng. Syst., 1 (4): 295-299. http://bit.kuas.edu.tw/~ijcses/v1/n4.

Ahmad, A.H. and S.A. Hoda, 1998. Residue-to-Binary Arithmetic Converter for the Moduli Set $(2^k, 2^k\text{-}1, 2^{k-1}\text{-}1)$. In: IEEE Transaction on Circuits and System-II (IEEE trans. II): Anal. Digital Signal Proc., 45: 204-209. DOI: 10.1109/82.661651. Acession Number: 5861252. http://ieeexplore.ieee.org/xpls/abs_all.jsp.

Ahmad, M.O., W. Wang and M.N.S. Swammy, 2003. A study of the residue-to-binary converters for 3 moduli sets in IEEE Transaction on Circuits and System-I (IEEE trans. I): Fundamental Theory and Applications, 50 (2): 235-243. DOI: 10.1109/TCS1.2002.808191. http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=01183647.

Chakraborti, N.B., J.S. Soudararajan and A.L.N. Reddy, 1986. An implementation of mixed-radix conversion for residue number applications. IEEE Trans. Comput, C-35: 245-253. DOI: 10.1109/TC1986.1676829. http://portal.acm.org/citation.cfm?id=6432.6445.

Elleithy, K. and M. Bayoumi, 1992. Fast and flexible architectures for RNS arithmetic decoding. In: IEEE Transaction on Circuits and System-II (IEEE trans. II), CAS-28: 226-235. Accession No.: 4182322. DOI: 10.1109/82.136572. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=136572.

Gbolagade, K.A. and S.D. Cotofana, 2008. Residue Number System Operands to decimal conversion for 3-moduli set. Proceedings of 51st IEEE Midwest Symposium on Circuits and Systems, Knoxville, USA., pp: 791-794. ISBN: 978-1-4244-2166-4. Accession No.: 10.1109/MWSCAS.2008.4616918. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4616918.

Huang, C.H., 1983. A fully parallel mixed-radix conversion algorithm for residue number applications. IEEE Trans. Comput., C-32: 398-402. DOI: 10.1109/TC1983.1676242. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1676242.

Szabo, N. and R. Tanaka, 1967. Residue Arithmetic and its application to Computer Technology, 1234567890 MP72106987. Card No. 66-15186 62659, New York: McGraw-Hill.

Van Vu, T., 1985. Efficient implementation of Chinese remainder theorem for sign detection and residue decoding. In: IEEE. Trans. Comput., C-34: 646-651. DOI:10.1109/TC.1985.1676602.http://portal.acm.org/citation.cfm?id=3907.

Wang, Y., 1998. New Chinese Remainder Theorem. In; Proc. 32nd Asilomar Conference on Signals, Systems and Computers, California, USA, pp: 165-171. ISBN: 0-7803-5148-7. Accession No.: 6319360. DOI: 10.1109/ACSSC.1998.750847. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=750847&fromcon.

Yassine, H.M., 1999. Fast Arithmetic based on Residue Number System Architectures. IEEE International Symposium on Circuits and Systems (ISCAS 91), ISBN: 0-7803-0050-5. Accession No.: 4244590. DOI: 10.1109/ISCAS.1991.176163. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=176163, pp. 2947-2950.

Yassine, H.M., 1991. Matrix Mixed Radix Conversion for RNS Arithmetic Architectures, 34th Midwest Symposium on Circuits and Systems, California USA., pp: 273-278. ISBN: 0-7803-0620-1. Accession No.: 43-94806. DOI: 10.1109/MWSCAS.1991.252046. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=252046.

Yassine, H.M. and W.R. Moore, 1991. Improved mixed-radix conversion for residue number architectures. IEE Proceedings G: Circuits. Devices and Systems, California, USA., 138 (1): 120-124. Accession No.: 3860412. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=87822.