Journal of Advances in Computer Research
Quarterly pISSN: 2345-606x eISSN: 2345-6078
Sari Branch, Islamic Azad University, Sari, I.R.Iran
(Vol. 10, No. 3, August 2019), Pages: 31-39
www.jacr.iausari.ac.ir

Low Complexity Converter for the Moduli Set $\{2^n+1, 2^n-1, 2^n\}$ in Two-Part Residue Number System

Shiva TaghipourEivazi✉
Department of Computer Engineering, Tabriz Branch, Islamic Azad University, Tabriz, Iran
taghipour@iaut.ac.ir

Received: 2019/06/01; Accepted: 2019/11/11

Abstract

Residue Number System is a kind of numerical systems that uses the remainder of division in several different moduli. Conversion of a number to smaller ones and carrying out parallel calculations on these numbers will increase the speed of the arithmetic operations in this system. However, the main factor that affects performance of system is hardware complexity of reverse converter. Reverse converters convert the resulted remainders to the conventional number system. In this paper an area efficient reverse converter is proposed for moduli set $\{2^n+1, 2^n-1, 2^n\}$ based on two-part RNS and mixed radix conversion algorithm. Selecting appropriate order of modulus and using well-known lemmas, leads to reduce the complexity of the proposed converter comparing to previous designs. To have an accurate comparison, both unit gate model and simulation in Xilinx 13.1 FPGA are used in this paper. The results of comparison indicate that the novel proposed reverse converter has improved the time complexity and area, while having almost same delay.

Keywords: *Computer Architecture, High-Speed Arithmetic operations, Parallel Processing, R/B Converter, VLSI*

1. Introduction

By referring[1], the residue number system (RNS) is a non-weighted system. RNS is based on a moduli set $\{m_1, m_2, ..., m_n\}$ that consists of a set of some positive integers named modulus. A binary number X in this system can be represented as $< x_1, x_2, ..., x_n >$ where $x_i = X \bmod m_i$. It is worth to note that the propagation of carry bits is completely omitted among different moduli in this system. Due to this advantage of RNS, this system can be used to reduce the complexity of calculations in many applications such as image encryption [2].

The effectiveness of RNS depends on many factors such as choosing the set of modulus. On the other hand, selection of a moduli set itself, is related to the simplicity and the speed of converter circuits[3].

To improve the efficiency of RNS, two-part representation is introduced in[4]–[6], which is based on dividing the number (X) to two separated parts with different moduli, as described in section 2. In section 3, the proposed reverse converter is presented for moduli set $\{2^n+1, 2^n-1, 2^n\}$ using Mixed-radix Conversion (MRC) and two-part RNS. The

hardware implementation of proposed converter is described in section 4. The performance of novel converter and previous circuits are compared in section 5 and finally, in section 6 the conclusion is presented.

2. Two-Part RNS

Considering an RNS system with moduli set $\{2^n+1, 2^n-1, 2^n\}$, the two-part (hybrid) RNS representation for this moduli set has been introduced previously in[4], [5] which are combined form of RNS and binary system. Due to[4] , in two-part representation, least significant *n* bits of *X*, is considered in binary system and then 2*n* bits are left for RNS system, thus the set of moduli set is reduced to $\{2^n+1, 2^n-1\}$ in RNS, thus according to (1) there are 3*n* bits in the dynamic range (DR) of this moduli set in this representation format.

$$M = \prod_{i=1}^{n} m_i = (2^n+1)\times(2^n-1)\times 2^n = 2^{3n} - 2^n \quad (1)$$

Previously two reverse converter have been introduced for this set based on two-part RNS in [4]and[5]. The reverse converter in[4], is based on Chinese Reminder Theorem (CRT) [7] and the converter in[5] is implemented using MRC[8], [9]. Two-part RNS is used in[6] to implement a reverse converter for moduli set $\{2^n-1, 2^{n+1}-1, 2^n\}$.

Recently a reverse converter is proposed for moduli set $\{2^{n+p}, 2^n-1, 2^n+1\}$ [10]with flexible $DR = 3n+p, 1 \le p < n$ for usual RNS, also in [11] a reverse converter is proposed for moduli set $\{2^{n+1}-1, 2^n-1, 2^n\}$, both of these converters are based on CRT and designed for usual RNS. Two reverse converters for moduli set $\{2^{n-1}-1, 2^n-1, 2^{n+k}\}$ are proposed in [12] that use MRC to design two different converters, $0 \le k \le n-2$ and $n-2 \le k \le n$ in usual RNS. Two reverse converters are presented for three moduli set $\{2^{2n+1}-1, 2^{2n}, 2^n-1\}$ in usual RNS [13], both of these converters are implemented via MRC and two-level RNS.

To reduce the complexity of reverse converter for moduli set $\{2^n+1, 2^n-1, 2^n\}$, a novel converter is proposed in this paper which is based on MRC. According to selecting appropriate order of modulus and using well-known lemmas, the complexity of circuit is reduced.

3. Proposed Reverse Converter for moduli set $\{2^n+1, 2^n-1, 2^n\}$ in two-part RNS

Because of using two-part representation, a reverse converter for the moduli set $\{2^n+1, 2^n-1\} = \{m_1, m_2\}$ is presented in this part to compute $X(a)$. This number is obtained with the corresponding residue representation $< x_1, x_2 >$ using MRC, according to (2). Finally x_3 is concatenated to $X(a)$ as *n*-bit low order bits similar to[5].

$$X(a) = v_1 + v_2 m_1 \quad (2)$$

Where

$$v_1 = x_1 \quad (3)$$

$$v_2 = |(x_2 - x_1) \times |m_1^{-1}|_{m_2}|_{m_2} \quad (4)$$

Note that $|m_1^{-1}|_{m_2}$ is the multiplicative inverse of m_1 modulus m_2. According to (2)-(4), we have:

$$X(a) = x_1 + \underbrace{\left((x_2 - x_1) \times |(2^n+1)^{-1}|_{2^n-1}\right)_{2^n-1}}_{S} \times (2^n+1) \quad (5)$$

It is possible to simplify the equations and find the multiplicative inverse using the following lemmas [13].

Lemma 1. $|a \circ b|_m = ||a|_m \circ |b|_m|_m$

Lemma 2. The residue of a negative residue number (-v) in moduli 2^n-1 is the one's complement of v, where $0 \le v < 2^n - 1$.

Lemma 3. The multiplication of a residue number v by 2^p in moduli 2^n-1 is performed by p bit circular left shift, where p is a natural number.

To calculate S, using Lemma 1 we have:

$$\begin{aligned} S &= \left((x_2 - x_1) \times |(2^n+1)^{-1}|_{2^n-1}\right)_{2^n-1} = \left((x_2 - x_1) \times 2^{n-1}\right)_{2^n-1} \\ &= |\underbrace{\left(x_2 \times 2^{n-1}\right)_{2^n-1}}_{S_1} \underbrace{-\left(x_1 \times 2^{n-1}\right)_{2^n-1}}_{S_2}|_{2^n-1} = |S_1 + S_2|_{2^n-1} \end{aligned} \quad (6)$$

S_1 is calculated using lemma 3.

$$S_1 = \left(\underset{nbit}{x_2 \times 2^{n-1}} \right)_{2^n-1} = \underbrace{x_{2_0} x_{2_{n-1}} x_{2_{n-2}} .... x_{2_1}}_{nbit} \quad (7)$$

x_1 has n+1 bits, therefore to use Lemmas 2 and 3, first we consider that x_{1_n} is zero, therefore:

$$S_2^{/} = \left(\underset{n+1bit}{-x_1 \times 2^{n-1}} \right)_{2^n-1} = \left(\underset{nbit}{-(0\, x_1 \times 2^{n-1})} \right)_{2^n-1} = \underbrace{\bar{x}_{1_0} \bar{x}_{1_{n-1}} \bar{x}_{1_{n-2}} .... \bar{x}_{1_1}}_{nbit} \quad (8)$$

If x_{1_n} be one, considering the form of first moduli, all low order bits will be zero, therefore:

$$S_2^{//} = \left(\underset{n+1bit}{-x_1 \times 2^{n-1}} \right)_{2^n-1} = \left(\underset{n}{-(100...0 \times 2^{n-1})} \right)_{2^n-1} = 0\underbrace{111....1}_{n-1} \quad (9)$$

Considering (8) and (9), S_2 will be computed as (10).

$$S_2 = \underbrace{(\bar{x}_{1_0} \oplus x_{1_n}) \bar{x}_{1_{n-1}} \bar{x}_{1_{n-2}} .... \bar{x}_{1_1}}_{nbit} \quad (10)$$

According to (5), to calculate $X(a)$ we have:

$$\underset{n+1}{X(a)} = \underset{nbit}{x_1} + \underset{nbit}{S} \times (2^n+1) = \underset{2nbit}{S \,||\, S} + \underset{nbit}{x_1} \quad (11)$$

Where || denotes concatenation operation.

4. Hardware Implementation

The hardware implementation of proposed reverse converter is shown in Figure 1. As it is illustrated in this figure, an operand propagation unit (OPU1) is used to generate S_1 and S_2. According to (7) and (10), this unit involves one exclusive OR (XOR) and *n* inverter gates, also the delay of this unit is equal to one XOR and one inverter. According to (6), to generate S, S_1 and S_2 are added using a modular adder 2^n-1. Thus, an *n*-bit carry propagate adder with end around carry (CPA with EAC) should be used [4]. Note that the delay of CPA with EAC is twice of the delay of a regular CPA, while having same hardware. Therefore to increase speed, the modularity of adder in (6) is discarded and will be postponed to $X(a)$ calculation in (13). Therefore a usual *n*-bit CPA is used in this part to generate *n*-bit S' and carry out (Cout), thus it consists of one half adder (HA) and *n*-1 full adders (FA). Similar to [3] we have:

$$S = S' + Cout \tag{12}$$

According to (12), (11) can be rewritten as follows:

$$X(a) = \underset{nbit}{x_1} + \underbrace{S' + Cout \,\|\, S' + Cout}_{2nbit} = \left(+ \underbrace{\underbrace{S'}_{nbit}\underbrace{S'}_{nbit}}_{} + \underbrace{\underbrace{000...000}_{n-1bit}\underbrace{x_1}_{n+1bit}}_{} + \underbrace{\underbrace{Cout}_{1bit}\underbrace{Cout}_{1bit}}_{} \right) = \left(+ \underbrace{S_3}_{2nbit} + \underbrace{S_4}_{2nbit} + \underbrace{\underbrace{0...0\,Cout}_{n-1bit}\underbrace{0...0\,Cout}_{n-1bit}}_{} \right) \tag{13}$$

According to (13), OPU2 is used to generate S_3 and S_4. Considering (13), this unit does not require any hardware. To calculate $X(a)$ a modified adder is used. Due to the constant zero bits in S_4 and the place value of *Cout* and the form of x_1, the modified adder is consists of (*n*+1) FA's in least significant bits (LSB) part and (*n*-1) HA's to calculate most significant bits (MSB) bits.

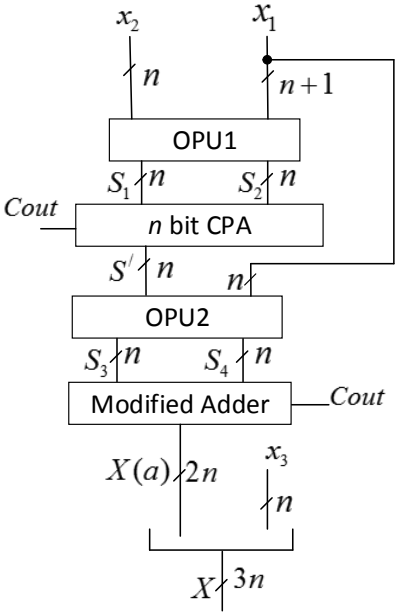

Figure 1. Proposed reverse converter

The following example illustrate the mentioned process.

Example:

Consider the moduli set $\{2^n+1, 2^n-1, 2^n\}$ with n =3 in two-part RNS. Therefore the moduli set is {9, 7, 8} and given residues are < 8, 6, 7 >, we have $x_1=(1000)_2$, $x_2=(110)_2$ and $x_3=(111)_2$.

According to Figure 1 and considering (7) and (10), $S_1=011$, $S_2=011$. S' and *Cout* will be computed as $011+011=\underset{C_{out}}{0}\ \underset{S'}{110}$.

Considering (13), by using OPU2 $S_3=110110$ and $S_4=001000$ are obtained. Therefore we have $X(a)=110110+001000+000000=111110$.

Finally $X=111110\|111=503$ is obtained.

5. Performance Evaluation

In this part a comparison between the proposed reverse converter and the previous converters is done. To have a fair comparison both unit gate model [3], [14] and simulation is performed. Table 1 displays the required hardware and delay of the proposed converter in details.

Table 1. Hardware and delay of proposed backward converter

Parts	INV	FA	HA	XOR	Delay
OPU1	n	-	-	1	$1t_{INV+}1t_{XOR}$
CPA	-	n-1	1	-	$1t_{HA+}(n\text{-}1)t_{FA}$
OPU2	-	-	-	-	-
Modified Adder	-	n+1	n-1	-	$(n\text{-}1)t_{HA+}(n+1)t_{FA}$
Total	n	2n	n	1	$1t_{INV+}1t_{XOR}+(n)t_{HA+}(2n)t_{FA}$

Table 2 shows the required delay and hardware of the proposed converter and the converters in [3], [5], [6], [10]–[12] in terms of logic gates.

Table 2. Required delay and hardware in details

Converter	Hardware requirements	Conversion delay
Proposed Converter	$nA_{INV}+(2n)A_{FA}+nA_{HA}+1A_{XOR}$	$1t_{INV+}1t_{XOR}+(n)t_{HA+}(2n)t_{FA}$
[3]Area-efficient	$(5n)A_{INV}+(7n)A_{FA}+(n+1)A_{XOR/AND}+(2n+2)A_{XNOR/OR}$	$3t_{INV+}(7n+5)t_{FA}$
[3]Delay-efficient	$(5n)A_{INV}+(7n)A_{FA}+(n-1)A_{XOR/AND}+(3n+2)A_{XNOR/OR}+1A_{MUX}$	$2t_{INV+}1t_{MUX}+(6n+5)t_{FA}$
[5]	$(4n+3)A_{INV}+(2n+1)A_{FA}+(2n+2)A_{OR/AND}+(4n)A_{MUX}+1A_{XOR}$	$3t_{INV+}(2n+1)1t_{OR}+3t_{MUX+}(n)t_{HA+}(2n+1)t_{FA}+1t_{XOR}$
[6]	$(2n+1)A_{INV}+(4n)A_{FA}+(4n+4)A_{HA}+(2n+1)A_{MUX}$	$1t_{INV+}1t_{MUX}+(3)t_{HA+}(2n)t_{FA}$
[10]	$(n+P)\ A_{OR/NOR}+(4n)A_{FA}+PA_{XOR}$	$1t_{OR}+(4n+1)t_{FA}$
[11]	$(n+1)A_{INV}+(5n+3)A_{FA}+(n+1)A_{XOR/AND}$	$1t_{INV}+(6n+3)t_{FA}$
[12]	$(4n+2k-1)A_{FA}+(4n-2k-2)A_{OR/AND}+(4n-2k-4)A_{XNOR/OR}$	$(6n-1)t_{FA}$

Table 3 displays unit gate area and delay for each logical component[3].

Table 3. Unit Gate Model

Component	Unit gate Area	Unit gate Delay
INV	1	1
FA	7	4
HA	4	2
XOR	1	1
MUX_{2-1}	3	2

Table 4 illustrates the results of comparison based on unit gate model. Note that time complexity is the multiplication of unit gate area by unit gate delay.

Table 4. Comparison based on unit gate model

Converter	Unit gate area	Unit gate delay	Time complexity
Proposed Converter	$19n+2$	$10n+3$	$190n^2+77n+6$
[3]Area-efficient	$63n+3$	$28n+23$	$1764n^2+1533n+69$
[3]Delay-efficient	$66n+6$	$24n+24$	$1584n^2+1728n+144$
[5]	$32n+14$	$12n+17$	$384n^2+712n+238$
[6]	$52n+20$	$8n+9$	$416n^2+628n+180$
[10]	$30n+4p$	$16n+5$	$480n^2+150n+64np+20p$
[11]	$39n+25$	$24n+13$	$936n^2+1107n+325$
[12]	$40n+8k-19$	$24n-4$	$960n^2-616n+192kn-32k+76$

As the results indicate, the proposed reverse converter has better time complexity comparing to the previous converters. The reason of this improvement is related to using two-part representation (comparing to [3], [10]–[12]). Also because of selecting appropriate order of modulus and using well-known lemmas, the proposed converter has been improved comparing to [5], [6].

To have an accurate comparison, FPGA simulation is done. The proposed converter in this paper and previously presented converters have been described in VHDL, with Xilinx ISE 13.1. Virtex5 is used and Device XCVLX220 ff1760-2 is chosen. Because of the different DR for moduli sets [3], [6], [10]–[12], to have a fair comparison the

converters have been synthesized to cover same DR's. In this case, for a chosen DR, the value of *n* is obtained. The results are shown in table 5.

Table 5. The simulation results

DR(bit)	Converter	Area (slice)	Delay(ns)	Time complexity
32	Proposed Converter, n=11	27	7.895	213.165
	[3]Area-efficient, n=7	99	17.690	1751.310
	[3]Delay-efficient, n=7	111	15.941	1769.451
	[5], n=11	42	9.315	391.23
	[6], n=11	49	8.302	406.798
	[10],n=11,k=1	89	17.677	1573.253
	[11], n=11	105	20.843	2188.515
	[12], n=11,k=1	138	21.069	2907.522
64	Proposed Converter, n=22	58	12.024	697.392
	[3]Area-efficient, n=13	185	26.931	4982.235
	[3]Delay-efficient, n=13	201	24.237	4871.637
	[5], n=22	144	12.536	1805.184
	[6], n=22	189	11.827	2235.303
	[10],n=22,k=1	290	40.562	11762.98
	[11], n=22	208	38.194	7944.352
	[12], n=22,k=1	299	39.504	11811.696
128	Proposed Converter, n=43	118	19.796	2335.928
	[3]Area-efficient, n=24	345	47.747	16472.715
	[3]Delay-efficient, n=24	378	39.430	14904.540
	[5], n=43	280	15.01	4202.8
	[6], n=43	452	12.674	5728.648
	[10],n=43,k=1	357	57.466	20515.362
	[11], n=43	410	69.631	28548.71
	[12], n=43,k=1	598	74.735	44615.584
256	Proposed Converter, n=86	233	34.271	7985.143
	[3]Area-efficient, n=52	749	98.639	73880.611
	[3]Delay-efficient, n=52	824	72.243	59528.232
	[5], n=86	390	22.775	8882.25
	[6], n=86	981	21.025	20625.525
	[10],n=86,k=1	720	111.219	80077.68
	[11], n=86	819	134.388	110063.772
	[12], n=86,k=1	1210	149.581	180993.01

As the results of simulation show, the proposed reverse converter has better area and time complexity comparing to all previously presented converters. The simplicity of proposed circuit, using two-part RNS and applying well known lemmas are the reasons of this improvement. Figure 2 is the RTL of proposed reverse converter for n=86.

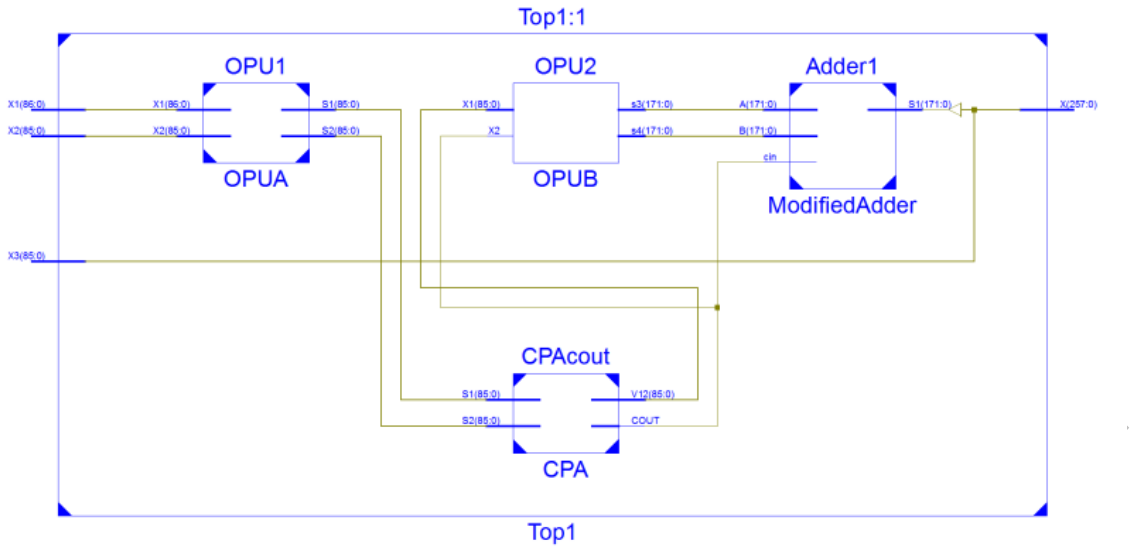

Figure 2. RTL of proposed reverse converter for n=86

6. Conclusion

In this paper a novel parallel RNS reverse converter is proposed for the moduli set $\{2^n+1, 2^n-1, 2^n\}$. By using two-part technique and postponing the modularity operation in CPA with EAC, the complexity of proposed converter has been improved significantly compared to previous similar converters. Using less area in proposed reverse converter leads to consume less power and cost and because this step is necessary for all applications of RNS, the improvement of reverse converter, leads to decrease the complexity and cost of RNS system.

References

[1] K. Navi, A. S. Molahosseini, and M. Esmaeildoust, "How to teach residue number system to computer scientists and engineers," *IEEE Trans. Educ.*, vol. 54, no. 1, pp. 156–163, 2011.

[2] A. A. Abbasi, R. Hosseini, and M. Mazinani, "A Novel Image Encryption Model Based on Hybridization of Genetic Algorithm , Chaos Theory and Lattice Map," *J. Advances Comput. Research*, vol. 9, no. 4, pp. 129–144, 2018.

[3] N. SalehiTabrizi and S. TaghipourEivazi, "Designing Efficient Two-Level Reverse Converters for Moduli Set { $2^{2n+1}-1$, 2^{2n}, $2^{n}-1$ }," *Circuits, Syst. Signal Process.*, vol. 37, no. 9, pp. 4162–4180, 2018.

[4] M. Jameii, Sh. Taghipoureivazi and M. Azad, "Using both Binary and Residue Representations for Achieving Fast Converters in RNS," *J. Advances Comput. Research*, pp. 91–104, 2011.

[5] S. Taghipoureivazi, M. Hosseinzadeh, and A. Habibizadnovin, "Efficient RNS converter via two-part RNS," *J. Circuits, Syst. Comput.*, vol. 24, no. 1, pp. 1–12, 2015.

[6] S. TaghipourEivazi, "Efficient Reverse Converter for New Moduli Set," *J. Advances Comput. Research*, pp. 87–94, 2017.

[7] K. M. Ibrahim and S. N. Saloum, "An efficient residue to binary converter design," *IEEE Trans. Circuits Syst.*, vol. 35, no. 9, pp. 1156–1158, 1988.

[8] A. Omondi and B. Premkumar, *Residue Number Systems: Theory and Implementation*. London, UK, UK: Imperial College Press, 2007.

[9] N. S. Szabó and R. I. Tanaka, *Residue arithmetic and its applications to computer technology*. McGraw-Hill, 1967.

[10] A. Hiasat, "A Residue-to-Binary Converter with an Adjustable Structure for an Extended RNS Three-Moduli Set," *J. Circuits, Syst. Comput.*, vol. 28, no. 08, p. 1950126, Aug. 2018.

[11] A. Hiasat, "An Efficient Reverse Converter for the Three-Moduli Set (2^{n+1}-1, 2^n, 2^n-1)," *IEEE Trans. Circuits Syst. II Express Briefs*, vol. 64, no. 8, pp. 962–966, 2017.

[12] M. V. N. M. Latha, R. R. Rachh, and P. V. A. Mohan, "RNS-to-Binary Converters for a Three-Moduli Set $\{2^{n-1}-1, 2^n-1, 2^{n+k}\}$," *IETE J. Educ.*, vol. 58, no. 1, pp. 20–28, Jan. 2017.

[13] M. R. Noorimehr, M. Hosseinzadeh, and K. Navi, "Efficient Reverse Converters for 4-Moduli Sets $\{2^{2n-1}$-1, 2^n, 2^n+1, 2^n-1$\}$ and $\{2^{2n-1}$-1, 2^{2n-1}, 2^n+1, 2^n-1$\}$ Based on CRTs Algorithm," *Circuits, Syst. Signal Process.*, vol. 33, no. 10, pp. 3145–3163, 2014.

[14] S. T. Eivazi, M. Hosseinzadeh, and O. Mirmotahari, "Fully parallel comparator for the moduli set $\{2^n, 2^n$-1, 2^n+ 1$\}$," *IEICE Electron. Express*, vol. 8, no. 12, pp. 897–901, 2011.