Booth Modified RNS Multiplier in RNS to Binary Code Converator Using {2P+1 2P, 2p-1}

P. Charantej
MTECH (VLSI) Student,
Sense Department, VIT University,
Vellore, Tamil Nadu
charantej.peteti@gmail.com

R. Dhanabal
Assistant Professor (Senior Grade)
VLSI Division, SENSE,
VIT University, Vellore, Tamil Nadu
rdhanabal@vit.ac.in

SS Kerur
-

Harish Kittu
-

Abstract: *A RNS reverse convertor moduli set {2p+1,2p,2p-1} is proposed in this paper. Chinese Remainder Theorem is simplified to get a reverse converter that uses mod-{2p-1} operations. The proposed architecture reduces the burden of explicit use of moduli operation in conversion process and we prove that theoretically speaking it outperforms state of the art equivalent converters. In order to restrict the range we makes use of radix-8 booth modified rns multiplier in the proposed converter on cyclone2 FPGA. When compare to other convertors, this architecture saves power, area, delay and cost .*

Index Terms: *Code converters, field-programmable gate arrays (FPGAs), residue arithmetic,Soc encounter,cadence.*

I. INTRODUCTION

The alluring carry free property of residue Number systems (Rnss) gives space for Rns executions in a mixture of specific high-execution digital signal processing (Dsp) requisitions. Rns is generally connected also and increase overwhelmed Dsp requisitions, for example digital separating and convolutions [1]. Moduli determination and information change are the two most essential issues that verify the Rns equipment execution and may confine the use of Rns in Dsp requisitions [2].The most popular moduli set is {2p+1, 2p,2p-1} in RNS scaling. The advantage of using the {2p+1,2p, 2p-1} over {2^p+1, 2^p, 2^p–1} is briefly narrated in [3]-[5]. In spite of the fact that numerous information convertors had proposed dependent upon either the Chinese remainder theorem (CRT) [5]–[8], or on the mixed radix conversion (MRC) [9]. The unpredictable and moderate modulo-operation is major issue in Crt. verify the Rns equipment execution and may confine the usage of Rns in Dsp requisitions[2]. The complex and slow modulo-operation is major problem in CRT. In this paper, a novel reverse converter for the moduli set {2p+1, 2p,2p-1} is proposed. First, when compared to converters, using at mod-(2p-1) (2p+1) and mod-(2p) (2p-1) operations ,used in [3] and [6] the CRT is simplified to obtain a reverse converter that utilizes only mod- (2p-1). Next, we show a low many-sided quality execution that does not require the explicit utilization of the modulo operation in the conversion transform as it is ordinarily the case in the conventional Crt furthermore some other state of the workmanship identical converters

II. PROPOSED ALGORITHM

From the hypothesis, we quickly revaluate rejecting evidence the accompanying hypothesis, which has been displayed in [6] before presenting our methodology.

Theorem 1: Given the Rns number with appreciation to the moduli set (u_1, u_2, u_3) is in the form of {2p-1,2p, 2p+1},the decimal likeness this Rns number is registered for (u_1, + u_3) even and odd, separately, as accompanies [6]

$$U=-m_2u_1+m_1|\frac{m_2}{2}(u_1+u_3)-m_3u_2|m_2m_3 \quad \ldots (1)$$

$$U=-m_2u_1+m_1|\frac{m_2}{2}m_3(u_1+u_3)-m_3u_2|m_2m_3 \quad \ldots (2)$$

we have to simplify (1) and (2) in order to obtain a convertor that only uses modulo (2p-1).

Theorem 2:

Given the Rns number with appreciation to the moduli set (u_1, u_2, u_3) is in the form of {2p-1, 2p, 2p+1}, the decimal likeness this Rns number is registered for (u_1,+u_3) even as accompanies :

$$U = y; \text{ for } y \geq 0; U = y + M \text{ for } y<0; \quad \ldots (3)$$

$$\text{Where } y=m_2(u_2-u_1)+u_2+m_1m_2|\frac{u_1+u_3}{2}-u_2|m_2m_3 \quad \ldots (4)$$

Proof: to prove this theorem we have to use lemma in [10]

$$|am_1|m_1m_2=m_1|a|m_2 \quad \ldots (5)$$

To be more precise (1) producing negative result, U is given as

$$U=|-m_2u_1+m_1|m_2\ \frac{u_1+u_3}{2}-m_3u_2|m_2m_3|M$$

put $m_3=m_2-1$ and apply (5)

we get

$$y=|m_2(u_2-u_1)+u_2+m_1m_2|\ \frac{u_1+u_3}{2}-u_2|m_3|M \quad \ldots (6)$$

Comparison (6) is the general declaration of (4), substantial for both positive what's more negative. The following phase of the confirmation is to show that at most one curative expansion is needed for the count of the mod-M. We show that by acknowledging the best worth one may get in (6).

***Most positive value*:** The following must hold true to get most positive value in (6):

$$|\frac{u_1+u_3}{2}-u_2|m_3 = m_3-1,\ u_1=m_1-1,\ u_2=1,\ u_3=m_3-1$$

Substituting all these values in (6), we get

$$U=|M-2+m_1m_2+2m_2+1|M \quad \ldots (7)$$

Since $0<M-2m_1m_2+2m_2+1<M$, no corrective addition of M is required for obtaining correct result. Here, for y<0, this must hold true $|\frac{u_1+u_3}{2}\ u_2|m_2=0$; $u_1>u_2$; So here one corrective addition is required for computing correct result which can be done as:

Most negative value:To obtain most negative in (6) put

$$u_1=m_1-1;\ u_2=0;\ |\frac{u_1+u_3}{2}-u_2|,3=0;$$

$$U=|-m_1m_2+1|M \quad \ldots (8)$$

since $0<M-2m_1m_2+1<M$, only one corrective addition of M is required for obtaining correct result if y <0. Thus given that $M=m_1m_2m_3$; in case y<0 the exact result looks like:

$$U=m_2(u_2-u_1)+u_2+m_1m_2|\frac{u_1+u_3}{2}-u_2|m_3 \quad \ldots (9)$$

Theorem 3: Given the Rns number with appreciation to the moduli set (u_1, u_2, u_3) is in the form of {2p-1,2p,2p+1},the decimal likeness this Rns number is registered for (u_1,+u_3) odd, as accompanies [6]

$$\{U = y; \text{ for } y\geq 0; U = y + M \text{ for } y<0; \quad \ldots (10)$$

Where

$$y=m_2(u_2-u_1)+u_2+m_1m_2|\frac{m_3}{2}+\frac{u_1+u_3}{2}-u_2|m_3 \quad \ldots (11)$$

Proof: To be more accurate (2) will rarely produce negative result, U is given as

$$U=|-m_2u_1+m_1\frac{m_2+m_3}{2}+\frac{m_3}{2}(u_1+u_3)=m_3u_2|m_3|_M$$

Put =-1 and apply (5) we get

$$U=|m_2(u_2-u_1)+u_2+m_1.m_2|\frac{u_1+u_3}{2}-u_2|m_3|_M \quad \ldots (12)$$

the above equation is a simple form of $|y|_M$. Just as said above one corrective addition is required for the calculation of mod-M.This illustration is as follows Most positive value:The following must hold true to get most positive value in (12)

$$|\frac{m_3}{2}+\frac{u_1+u_3}{2}-u_2|m_3=m_3=1,\ u_1=1,\ u_2=1,\ u_3=m_3-1.$$

substituting all these values in (12), we get

$$U = |M - m_1m_2 + 1|_M \quad \ldots (13)$$

Since $0<M-2m_1m_2+2m_2+1<M$, only one corrective addition of M is required .

Once more, for y<0; the following must hold true:

$|\frac{m_3}{2} + \frac{u_1+u_3}{2} - u_2|_{m3}=0$; $u_1=1$; $u_2=0$; and $u_3=m_3-1$.

With the help of these values (12) is in form of

$U = |-m_3+ 1|_M$ ... (14)

since 0<-+1+M<M,one corrective addition is used to get exact result.Regarding the case y<0,the correct result can be computed as follows :

$U=m_2 (u_2-u_1) +u_2+m_1m_2 (|\frac{m_3}{2}+\frac{u_1+u_3}{2} - u_2|_{ms}+m_3)$ ...(15)

III. HARDWARE PROPOSED ARCHITECTURE

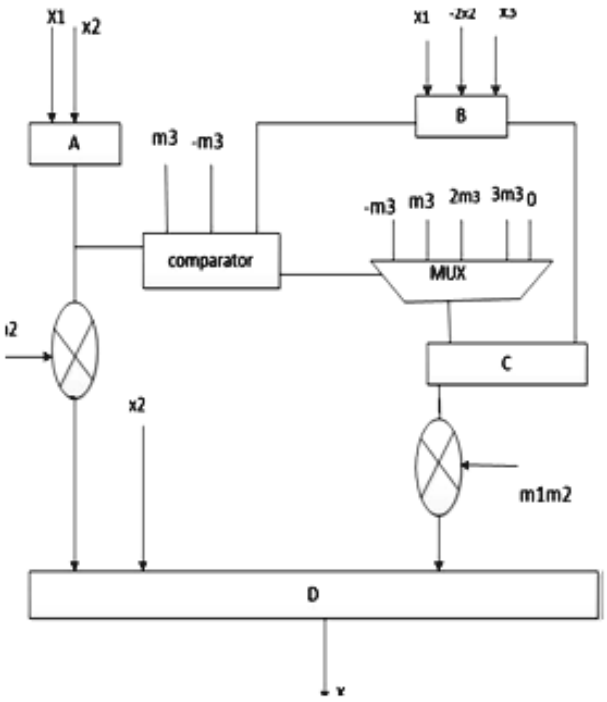


Fig. 1: Fittings structure of our proposal.

The fittings acknowledgment of the proposed plan, portrayed in Fig. 1 is dependent upon the comparisons in Hypotheses 2 and 3. The adder A output : A= (u_2-u_1) next the consequence is duplicated by m_2 to get M2A. The three-data adder B figures $(\frac{u_1+u_3}{2}-u_2)$ and might be executed as a 3:2 carry save adder carry propogate adder (Cpa). We demonstrate later in this area that just one restortive expansion or subtraction is obliged to register the modulo-m3 operation and this could be consolidated with the conceivable increases of m_3 and $\frac{m_3}{2}$.

The aforementioned operations are actualized by adder C with a selectable info. The fittings execution evacuates the divisions by moving left all the operands included in adder B and C, along these lines amplifying the two adders with one spot. At long last, the yield of adder C, without the rightmost spot to represent the past movement, is reproduced by P12=(m_2m_1) and the effect is summed together by snake D with the one from the multiplier m_2. The additional include u_2 for adder D might be implanted in the multiplier consistent with the rule of combined number-crunching, therefore D could be really executed as a standard 2:1 adder.

With help of 4 following cases we can say that no explicit use of hardware in the modulo-operation for computing

$|\frac{u_1+u_3}{2} - u2|m_3$ and $|\frac{m_3}{2} + \frac{u_1+u_3}{2} - u_2|_{m3}$;

Case 1: Here) =0 and =2p-1 and the results in the most negative value . For this case the modulus in (4) reduces to $|-u_2|_{m3}$. To perform the modulo m3operation we need to do corrective additions.

Given that $m_3+ (-u_2) = (2p-1)-(2p-1) =0$, for any positive integer n, for computing modulo we need only one corrective addition with m_3.

Case2: Here (u_1+u_3) = even and u_2 =0 and the results in the most negative value. In this case the modulus in (4) reduces to $|\frac{u_1+u_s}{2}|m_3$. To perform the modulo operation we need to do corrective additions.

Given that $m_3 (\frac{u_1+u_3}{2}) = (2p-1)-(2p+2p-2/2) = (2p-1)-(2p-1) =0$.This says sum in modulo adder cannot exceed m_3, one subtraction with is required.

Case 3: $(u_1+u_3)= 1$ and $= 2p -1$.Then modulus from (11) reduces to $|\frac{m_s}{2} + (\frac{1}{2}) -u_2|_{m3}$.

In this case $\frac{m_s}{2} + \frac{1}{2} - u_2$ is always negative that $m_3 \frac{m_s}{2} + \frac{1}{2} + -u_2=2p-1-\frac{1}{2}+\frac{1}{2}-2p+1= p>0$, for any integer p. For computing the modulo, we need one corrective addition with .

Case4: (u_1+u_3) is odd, $u_2= 0$.The modulus from (11) reduces to$|\frac{m_s}{2} + \frac{u_1+u_s}{2} m_s$. given that $(2m_3)-((\frac{m_s}{2})+\frac{u_1+u_s}{2}) = 2(2p-1)-((\frac{2p-1}{2}) +(((2p+2p-2)-1)/ 2))=4p-2-3p+2=p>0$, for any positive integer. For computing the modulo we need one

corrective subtraction with m_3: so with one most add or subtract we implement modulo m_3.

In the following, adding of ($\frac{m_s}{2}$) the term is actually delayed and appended to the correction step required for the modulo-m3 operation without any delay overhead.Thus, we removed $\frac{m_s}{2}$ as adder C input and rechecked the four correction cases detailed above.

if (u_1+u_3) *is even*,the term ($\frac{m_s}{2}$)is not included in calculation and the correction performed as usual.

If (u_1+u_3)=*odd* , the provisional sum at the output of adder B is $\frac{u_1+u_s}{2}$ - u_2 instead of $\frac{m_s}{2}$ + ($\frac{u_1+u_s}{2}$) - u_2 . Then the provisional sum is smaller with $\frac{m_s}{2}$ than it should actually be.

Taking into consideration the correction rules changes as follows:

1. (u_1+u_3) = *even*

if provisional sum >0 add m_3.

if provisional sum is $\geq m_3$ then subtract m_3; otherwise do nothing;

2. (u_1+u_3) = odd

if provisional sum is < - ($\frac{3m_s}{2}$). add ($\frac{3m_s}{2}$); if provisional sum is > $\frac{m_s}{2}$ then subtract $\frac{m_s}{2}$;

otherwise add $\frac{m_s}{2}$;

As indicated by (9) and (15), no need of explicit implementation for modulo-M. Here corrective addition of $m_3$is done before multiplication of m_1m_2.

The correction is done when the following conditions are true:

$u_1>u_2$

$\{| \frac{u_1+u_s}{2} - u_2|_{m3} = 0$; (u_1+u_3) even ...(16)

$\{| \frac{m_s}{2} + \frac{u_1+u_s}{2} - u_2|_{m3} = 0$;

(u_1+u_3) old ...(17)

By revising the correction rules we can combine modulo-m3operations as follows:

1. *if . (u_1+u_2) =even*

if provisional sum < 0 add m_3; (17) become true when the provisional sum is equal to, but on observing Case 1that (16) is false, hence no need of extra m_3 addition;

- if provisional sum =0 and (16) is true (the sign bit of adderA is 1) add ;
- otherwise do nothing;

(17) become true when the provisional sum = m_3 and but on observing Case 2 all these happens when u_2=0 and u_1=2p, hence (16) become true.Hence for modulo m_3 adder cancelling of the required addition on behalf of previous m_3 corrective subtraction.

2. if . (u_1+u_3) = odd

if provisional sum < ($\frac{m_s}{2}$) add $\frac{3m_s}{2}$;

from case 3 the provisional sum is

$\frac{1}{2}$ $-u_2$ <- $\frac{3m_s}{2}$, so (17) is false;

therefore no extra addition of is required to add.

if provisional sum < ($\frac{m_s}{2}$) add $\frac{m_s}{2}$;

(17) become true when provisional sum = $\frac{m_s}{2}$.

Following this u_1=u3-$2u_2$=$2m_3$ $\Rightarrow$ u_1-u_2=$2m_3$-u_3+u_2;

Since $2m_3$-u_3> 0 (16) also become true, then correction will become

$\frac{m_s}{2}$ + m_3 $\frac{m_s}{2}$ = . otherwise add ($\frac{m_s}{2}$).

From these computations there is an incredible decrease of modulo operation with assistance of a solitary remedial addition or subtraction there by diminishing the disambiguation quality of convertor.

IV. RESULTS

45nm Technology

The top module is FINAL.v

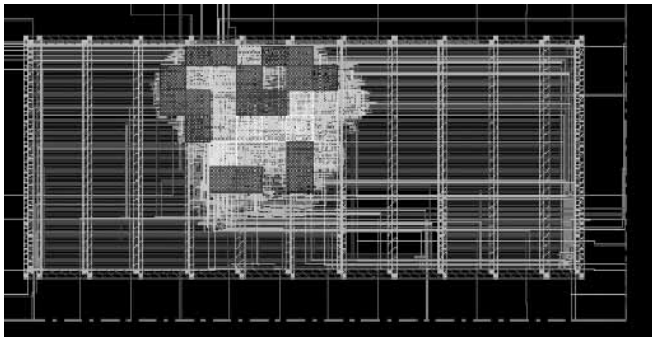
# of cells=2898

Dyanimic power=1.30mw

Total area=4904

Slacktime=54.3ns

V. LAYOUT VIEW



VI. MODIFICATIONS AND CONCLUSION

In this proposed architecture we make use of radix-8 RNS booth modified multiplier [11]with help of beamount smith modulo adder[12].By using booth modified multiplier the total partial products generated are reduced to 3.When compare to all other converters in [6][3] this proposed convertor uses an efficient way of reversion conversion process in area, complexity. when compared to the usage of mod-(2p+1)(2p-1)(2p) operations in [3][6] this proposed architectur used the mod- (2p-1) by simplyifing the traditional CRT.Then we implement the corrective addition m3 instead of corrective addition of M.hence, we implement a lowcomplexity architecture which uses simple addition or subtraction for reducing the overhead of the explicit modulo operation.

REFERENCES

[1]. 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.

[2]. W. Wang, M. Swamy, M. Ahmad, and Y. Wang,, "A study of the residue-to-binary converters for the three-moduli sets," Feb. 2003.

[3]. M. Ahmad, Y. Wang, and M. Swamy "Residue-to-binary number converters for three moduli sets," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 46, no. 7, pp. 180–183, Feb. 1999.

[4]. A. Premkumar, "An RNS to binary converter in moduli set," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 39, no. 7, pp. 480–482, Jul. 1992.

[5]. A. Premkumar, "An RNS to binary converter in a three moduli set with common factors," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 42, no. 4, pp. 298–301, Apr. 1995.

[6]. K. Gbolagade and S. Cotofana, "An efficient RNS to binary converter using the moduli set," XXIII Conf. Des. Circuits Integr. Syst. (DCIS), Grenoble, France, Nov. 2008.

[7]. K. Gbolagade and S, "A residue to binary converter for the moduli set," Cotofana, in Proc. 42nd Asilomar Conf. Signals, Syst., Comput. (ACSSC), Oct. 2008, pp. 1785–1789.

[8]. B. Vinnakota and V. Rao, "Fast conversion techniques for binary-residue number systems,"IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 41, no. 12, pp. 927–929, Dec. 1994.

[9] . M. Akkal and P. Siy, "A new mixed radix conversion algorithm MRCII," J. Syst. Arch., vol. 53, pp. 577–586, 2007.

[10]. Ramya Muralidharan, Radix-8 Booth Encoded Modulo 2n-1 Multipliers With Adaptive Delay for High Dynamic Range Residue Number System.

[11]. Beaumont-Smith, Cheng-Chew Lim, "Parallel Prefix- Adder Design", IEEE, 2001

[12]. Y. Wang, "Residue-to-binary converters based on new Chinese remainder theorems," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 47, no. 3, pp. 197–205, March 2000.

❐