IJRD International Journal of Research & Development Organisation

ISSN-4567-7860

JOURNAL OF ELECTRICAL AND ELECTRONINS ENGINEERING

Investigation on Signed Modulo Multipliers for RNS Applications using FPGAs.

Mr. Pradeep. N[1] , Ms. K. Gayathri[2]
Bannari Amman Institute of Technology, India

Abstract–The RNS has been considered as an interesting theoretical topic for researchers in recent years because of its ability to reduce the Hardware complexity compared to other counter parts such as binary number system which has direct proportionality between width(input bitlength) and hardware requirement. Its importance stems from the absence of carry propagation between its arithmetic units. This facilitates the realization of high-speed, low-power arithmetic. The modulo 2^n multiplier is assumed to be simplest among the special module set, is taken into consideration and analysis have been made as a comparative prescriptive with normal modified booth multiplier under Radix-4(2^2).Investigation have been made on the VLSI constrain by forcing the same input among the Ordinary Modified booth Encoder Multiplier and modulo 2^n multiplier and comparative performance parameter are analyzed, and on the conclusion obtained ,future recommendations have been made for multi-modulo multiplier.

Author [1,2] : PG Student, ECE Dept, Bannari Amman Institute of Technology, Sathyamangalam.
e-mail: bpradeepn@gmail.com,k.gayathri344@gmail.com

I. Introduction

Long word-length integer multiplication is widely known to be the bottleneck operation in many Hardware implementation based application such as Digital signal processing and cryptography. Residue Number System (RNS) has emerged as a promising alternative number representation for the design of faster and low power multipliers owing to its merit to distribute a long integer multiplication into several shorter and parallel modulo multiplications[4]. This advantage is of paramount importance in embedded processors, especially those found in portable devices, for which power consumption is the most critical aspect of the design.[3] However, the overhead introduced by the data conversion circuits discourages the use of RNS at the applications. Even though Arithmetic operations like addition, subtraction, multiplication, squaring and exponentiation when implemented in RNS can achieve high speed of operation compared to decimal or binary system.

Rest of the paper is organized as section (ii) briefs the operation of ordinary booth encoder multiplier, section(iii) comprises RNS Basics, and Modulo 2^n Residue Generation and multiplication are discussed in section (iv), Result analysis are done in section (v) , conclusion and future scope are exploited in section (vi).

II. Modified Booth Encoder Multiplier

The modified-Booth algorithm is extensively used for high-speed multiplier circuits. Once, when array multipliers were used, the reduced number of generated partial products significantly improved multiplier performance. Modified Booth Multiplier is one of the different techniques for signed multiplication. This multiplier architecture is based on Radix 4(2^2) Booth multiplier.

Multiplication consists of three steps: 1) the first step to generate the partial products;

2) the second step to add the generated partial products until the last two rows are remained;

3) the third step to compute the final multiplication results by adding the last two rows.

The modified Booth algorithm reduces the number of partial products by half in the first step. We used the modified Booth encoding (MBE) scheme proposed in [booth paper]. It is known as the most efficient Booth encoding and decoding scheme. To multiply X by Y using the modified Booth algorithm starts from grouping Y by three bits and encoding into one of {-2, -1, 0, 1, Table (1) presents the truth table of the new encoding scheme. The encoder generates one, two, and Z2i signals by encoding the three bi-signals.

2}.

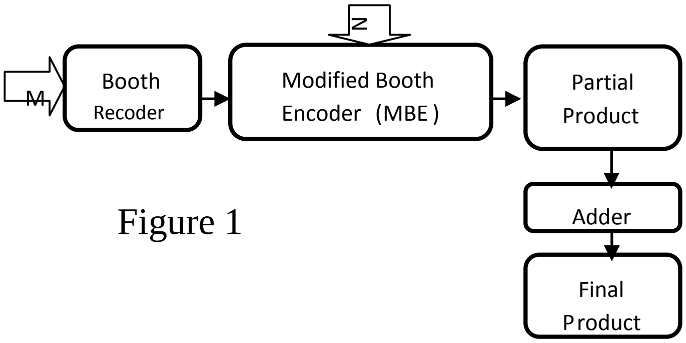
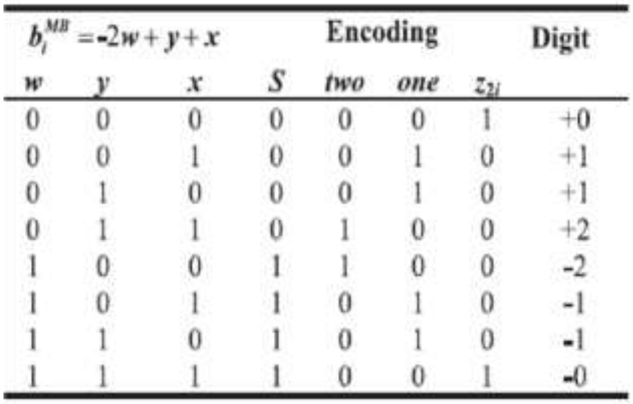


Figure 1

Table 1:Modified Booth Encoder Logic [1]



$b_i^{MB} = -2w+y+x$			Encoding				Digit
w	y	x	S	two	one	z_{2i}	
0	0	0	0	0	0	1	+0
0	0	1	0	0	1	0	+1
0	1	0	0	0	1	0	+1
0	1	1	0	1	0	0	+2
1	0	0	1	1	0	0	-2
1	0	1	1	0	1	0	-1
1	1	0	1	0	1	0	-1
1	1	1	1	0	0	1	-0

Here –2*multiplicand is actually the 2s complement of the multiplicand with an equivalent left shift of one bit position. Also, +2 *multiplicand is the multiplicand shifted left one bit position which is equivalent to multiplying by 2.

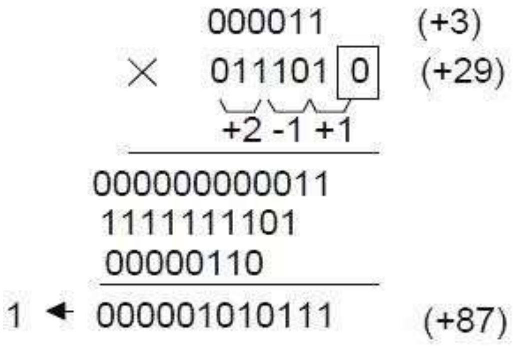
neg_i □ $y_{2i□1}$ &(y_{2i} ~ &$b_{2i□1}$) one_i □ y_{2i} ^ $y_{2i□1}$

two_i □ (($y_{2i□1}$ & y_{2i}) ~| (y_{2i} & $y_{2i□1}$ & $y_{2i□1}$))

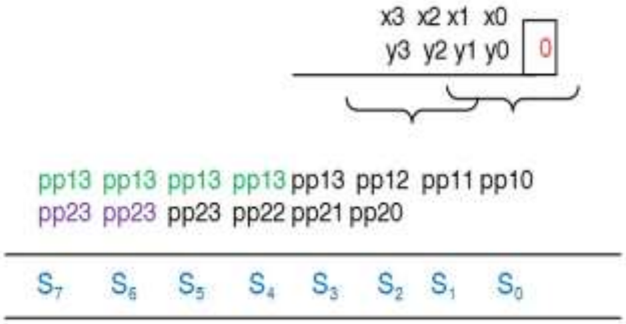
n_{xj} □ x_j ^ $y_{2i□1}$

p_{ij} □ (n_{xj} | one_i) ~ &($n_{xj□1}$ | two_i)

To enter 2*multiplicand into the adder, an (n+1)-bit adder is required. In this case, the multiplicand is offset one bit to the left to enter into the adder while for the low-order multiplicand position a 0 is added. Each time the partial product is 1] Multiplication of 2 positive nos.

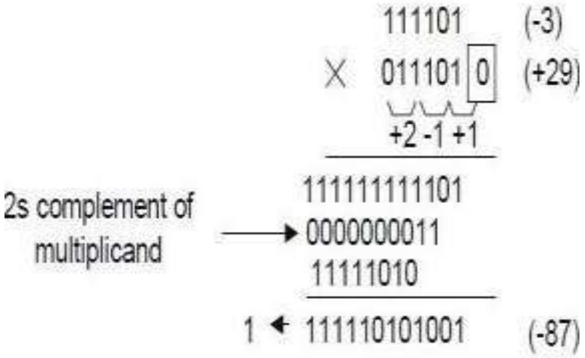


shifted two bit positions to the right and the sign is extended to the left.


x3 x2 x1 x0
y3 y2 y1 y0
0
pp13 pp13 pp13 pp13 pp13 pp12 pp11 pp10
pp23 pp23 pp23 pp22 pp21 pp20
S7 S6 S5 S4 S3 S2 S1 S0

Here are some examples for understanding:

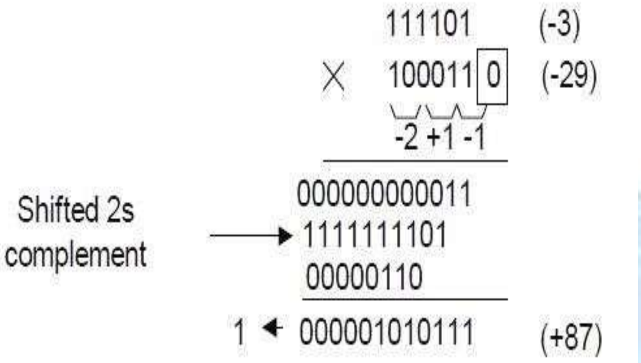
2]Multiplication of a positive and Negative nos.



III. RNS Basics

Residue Number System (RNS), an unconventional and non-weighted number representation, has emerged as a viable solution to implement long multiplications. RNS facilitates design of high speed multipliers by its virtue to decompose an integer multiplication into several small word-length and parallel modulo multiplications[RM]. Furthermore, as the modulo multiplications are independent of each other, an error in one residue channel will not be propagated to other channels. This fault tolerance offered by RNS becomes a valuable feature in deep submicron VLSI multipliers at low voltage operation.

3]Multiplication of 2 Negative nos.



The decomposition of a binary number into its residues is known as the binary-to-residue or forward conversion. Conversely, the composition of the residue back to a binary number is known as the residue-to-binary or reverse conversion. Thus, a complete

RNS multiplier consists of three components: a binary-to-residue converter, parallel modulo multipliers and a residueto-binary converter. Figure 2 briefly explains the general process that involve in the Binary to Residue conversion and the residue multiplication using Radix-2^2 Multiplication method.

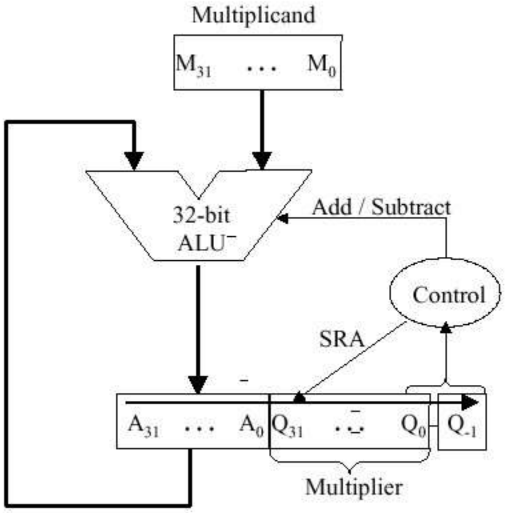


Figure 2

In general ,The RNS is defined in terms of a set of relatively prime moduli.

Consider if M={m_1,m_2,m_3,.........m_L} and GCD(m_i,m_j)=1, for i ≠ j. hence the dynamic range-DR is given by the product of the relatively prime moduli in the moduli set.

that is, DR = m_1 x m_2 x m_3 x....x m_L.

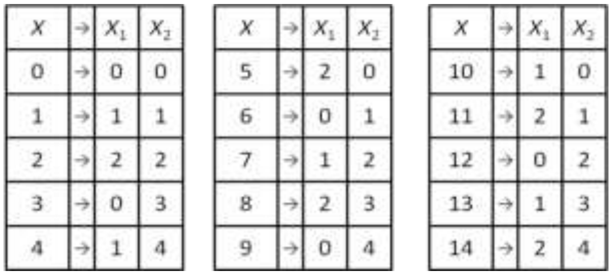
Any integer in the residue class Z_m has a unique L-tuple representation given by

X □□□□ *RNS*

(x_1,x_2,......,x_L) where

x_i=X mod m_i and is called the *i*th residue.

P□□*p*1, *p*2□□□3,5□ ***M***□3□5□15



X	→	X1	X2
0	→	0	0
1	→	1	1
2	→	2	2
3	→	0	3
4	→	1	4

X	→	X1	X2
5	→	2	0
6	→	0	1
7	→	1	2
8	→	2	3
9	→	0	4

X	→	X1	X2
10	→	1	0
11	→	2	1
12	→	0	2
13	→	1	3
14	→	2	4

IV. Modulo 2^n Residue Generation and multiplication

The residue for modulo 2^n can be successfully generated by truncating the bits, apart from the least nbit that is in simple taking only the least nbit gives the residue for modulo 2^n.

Consider the following example for understanding
An 8 x 8 Multiplication is done in Binaryto-Residue conversion method. Here if we consider n=4,then the multiplication process will be

ordinary multiplication: 01010101 X 00110011
85 X 51
=4335 mod16
=15

Modulo Multiplication:
0101 X 0011
5 X 3
=15

V. Result analysis

In general ,area delay performance will surely get improvised while we are moving from any ordinary multiplier to binary -to-residue converted modulo multiplication (i.e simply a residue multiplication).

This paper focus mainly on comparative numerical analysis on ordinary Booth multiplier's performance against modulo multiplier's performance in terms of VLSI constrains such as Area, Delay and power consumption.

The Multiplier were taken for analysis was described using structural Verilog HDL and synthesized to produce a gate level net list using two different synthesizer namely Xilinx ISE Design Suite 14.3, Altera Quartus II 12.0 with reference to Virtex7 XCV2000T-2FLG1925 and Cyclone II EP2C35F672C6 FPGA respectively. The multipliers were simulated and analyzed at different strengths such as 8 x 8, 16 x16, 32 x 32 and 64 x 64 as shown below in table [2-4] .

a) Area Analysis

In FPGA based design, Area requirement of the design is proportional to logic utilization i.e in Xilinx - Number of Slice LUTs Required and in Altera its Number of Logic Elements Required. For 16 x 16 bit strength Modulo Booth Multiplier Consume 61.2% lesser area than ordinary Booth Multiplier.

b) Delay Analysis

In FPGA based Design, EDA tools having inbuilt capability to predict the Delay of the design. In Xilinx - Timing Analyzer Tool and in Altera Time Quest Timing Analyzer Tool were used for delay analyze. Various Delay analysis shows Modulo multiplier has about 48% performance efficient over Booth multiplier.

c) Power Analysis

Power Evaluation of the design done at various levels such as Total Thermal power Dissipation (mW-milli Watt's), Core Dynamic Thermal power Dissipation (mW), core static Thermal power Dissipation (mW), I/O Thermal Power Dissipation(mW). Among the various power levels dynamic power varies with design to design it decides the efficient architecture.

Dynamic Power Requirement of the design is decided based on number of signal transition (or) activity during simulation time. Here analysis has been made using Power Play Power Analyzer from Altera. Power Analyzer required an input file of

Signal Activities and Value Changed Dump (VCD) File to evaluate the power of the design. Here we have measure the signal activities count for 20 different Samples for 100ns simulation and the same sample is forced for other design also in order to evaluate the exact power difference between the design. power Analysis with powerplay analyzer tool for 4 x 4 bit shows 20.27% Modulo multiplier consume higher than ordinary Booth Multiplier and found consistence for all strength and the analysis done on Reference [1] found to be relevant.

Table 2 : Area and Delay analysis using Xilinx ISE

	Multiplier	No. of IOBs	Xilinx Virtex7 XCV2000T-2FLG1925

Multipliers Strength	Name		No. of Slice LUTs Required	Delay (ns)
8x8	MODULO 2^n	36	37	11.929
	BOOTH	32	96	22.151
16x16	MODULO 2^n	72	137	21.002
	BOOTH	64	354	40.870
32x32	MODULO 2^n	144	362	32.383
	BOOTH	128	1595	81.191
64x64	MODULO 2^n	288	1709	69.575
	BOOTH	256	6480	159.289

Table 3 : Area analysis using Altera Quartus-II

Multipliers Strength	Multiplier Name	No. of IOBs	Altera Cyclone II EP2C35F672C6	
			No. of Logic Elements Required	Delay (ns)
8x8	MODULO 2^n	44	54	16.620
	BOOTH	32	150	25.082
16x16	MODULO 2^n	88	209	31.838
	BOOTH	64	538	42.826
32x32	MODULO 2^n	176	541	44.190
	BOOTH	128	2,284	87.473
64x64	MODULO 2^n	352	2,377	61.214
	BOOTH	256	9,542	189.886

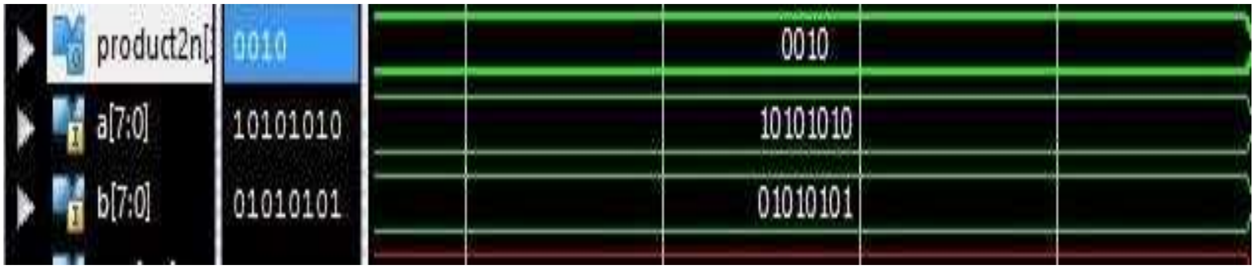
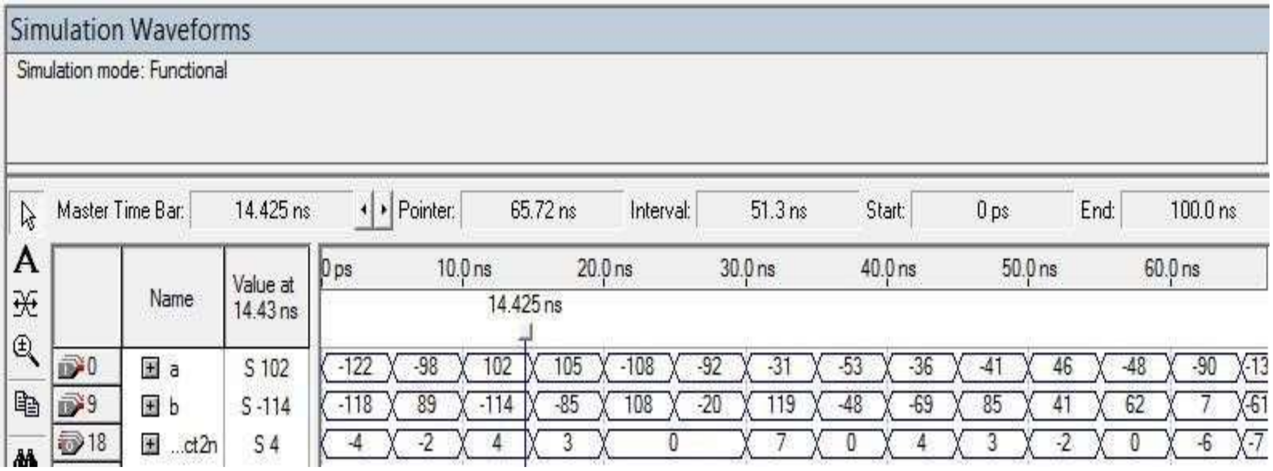


Figure 3



Simulation Waveforms

Simulation mode: Functional

Master Time Bar: 14.425 ns | Pointer: 65.72 ns | Interval: 51.3 ns | Start: 0 ps | End: 100.0 ns

	Name	Value at 14.43 ns
0	a	S 102
9	b	S -114
18	...ct2n	S 4

Figure 4

Table 4 : Power Analysis (Time interval of 100ns with 20 different samples)

Multipliers Strength	Multiplier Name	Altera Cyclone II EP2C35F672C6				
		Number Signal Transition during simulation for 100ns	Power estimation			
			Total Thermal Power Dissipation (mW)	Core Dynamic Thermal Dissipation (mW)	Core Static Thermal power Dissipation (mW)	I/O Thermal power Dissipation (mW)
8x8	MODULO 2^n	2795	267.69	2.02	80.45	185.23
	BOOTH	10291	222.68	5.31	80.30	137.07
16x16	MODULO 2^n	19877	464.73	9.06	81.13	374.54
	BOOTH	53998	350.73	21.07	80.73	248.93
32x32	MODULO 2^n	58860	668.60	24.14	81.85	562.62
	BOOTH	443814	612.41	83.08	81.65	447.68
64x64	MODULO 2^n	94547	872.34	31.53	82.03	758.78
	BOOTH	1877344	1278.88	360.30	83.24	836.34

The Xilinx Simulation result for booth- 8 x 8 bit is exhibited below in the Figure 3, and then the structure level port-map model is synthesized as Gate-level Netlist for signal Transition calculation.

VI. Conclusion

Our work has covered analysis of advanced Modified Booth multiplier architecture with Radix-4 Encoding and Modulo Multiplier at various strength such as 8 x 8, 16 x 16, 32 x 32 & 64 x 64 and the Result analysis with various VLSI Parameters like (Delay, Number of Logic Element requirements, Number of Signal Transition for particular sample input and its Power Consumption). As the Multiplier strength grows Area Curve shows a moderate difference while the delay performance of modulo multiplier compared to that of Booth Encoder Multiplier is approximately 4 times better. Modulo multiplier proves great result in all forms of VLSI constraints and works effectively with desired specification needed for highly reliable RNS application and for further optimization Multi-Modulo Residue architecture are advisably wise choice.

References References References

1. Pradeep.N, S.Elango, Dr.P.Sampath and K.Gayathri "Investigating the VLSI Characterization of Parallel Signed Multipliers for RNS Applications using PGAs" Global Journal of Computer Science and Technology: A Hardware & Computation Volume 15 Issue 1
Version 1.0 Year 2015

Modulo multiplier 8 x 8 bit simulation result on Altera Quartus-II is illustrated in the Figure 4, and then synthesis summary is depicted in table 2,3 and 4.

2. K. N. Vijeyakumar, Dr. V. Sumathy and S. Elango "VLSI Implementation of Area-Efficient Truncated Modified Booth Multiplier for Signal Processing Applications" The Arabian Journal for Science and Engineering, Volume. 39, No.11, 7795-7806, 2014.

3. Data Conversion in Residue Number System A thesis submitted to McGill University in partial fulfillment of the requirements for the degree of Master of Engineering. © 2011 Omar Abdelfattah

4. R. Muralidharan and C. H. Chang, "Hard multiple generator for higher radix modulo multiplication," in Proceedings 12th International Symposium. Integrated Circuits, Singapore, 546–549, 2009.

5. A. Dandapat, S. Ghosal, P. Sarkar, D. Mukhopadhyay (2009), "A 1.2- ns16×16-Bit Binary Multiplier Using. High Speed Compressors", International Journal of Electrical, Computer, and Systems Engineering,234-239, 2009.

6. Kiat-seng yeo and Kaushik Roy, "Low-Voltage, Low-Power, VLSI Subsystems Tata MC-Graw Hill.

7. S. L. Freeny, "Special-purpose hardware for digital filtering," Proceedings, .IEEE, 63-4- 633–647 1975.

8. C. S. Wallace, "A suggestion for parallel multipliers," IEEE Transaction on Electronic and Computer, 13-1-14–17,1964.

9. O. Hasan and S. Kort, "Automated formal synthesis of Wallace tree multipliers,"in Proceedings 50th Midwest Symposium Circuits and System, 2007.

10. J. Fadavi-Ardekani, "M × N booth encoded multiplier generator using optimized Wallace trees," IEEE Transaction. on Very Large Scale Integration.(VLSI) System,1-2-120–125,1993.

11. F. Elguibaly, “A fast parallel multiplier-accumulator using the modified Booth algorithm,” IEEE Transaction. Circuits System. II, Analog Digitial. Signal Process., 479-902–908, 2000.

12. K. Choi and M. Song, “Design of a high performance 32 × 32-bit multiplier with a novel sign select Booth encoder,” in Proceedings on IEEE International. Symposium on Circuits System, 2-701–704, 2001.

13. Y. E. Kim, J. O. Yoon, K. J. Cho, J. G. Chung, S. I. Cho, and S. S. Choi, “Efficient design of modified Booth multipliers for predetermined coefficients,” in Proceedings on IEEE International. Symposium on Circuits and Systems, 2717–2720, 2006.

14. W.-C. Yeh and C.-W. Jen, “High-speed booth encoded parallel multiplier design,” IEEE Transactions on Computers, 49-7-.692–701, 2000.

15. J.-Y. Kang and J.-L. Gaudiot, “A simple high-speed multiplier design,” IEEE Transactions on Computers ,5510-1253–1258,2006.

16. O. Salomon, J.-M. Green, and H. Klar, “General algorithms for a simplified addition of 2’s complement numbers,” IEEE Journal on Solid-State Circuits,30-7-839–844, 1995.

17. E. de Angel and E. E. Swartzlander, Jr., “Low power parallel multipliers,”in Workshop VLSI Signal Process. IX, 199–208, 1996.

18. N. S. Szabo and R. I. Tanaka, Residue Arithmetic and its Application to Computer Technology. New York: McGraw-Hill, 1967.

19. M.A. Soderstrand et al., Residue Number System Arithmetic Modern Applications in Digital Signal Processing. IEEE Press 1986.

20. Paliouras and T. Stouraitis, “Novel High-Radix Residue Number System Multipliers and Adders,” Proceedings IEEE International Symposium on Circuits and Systems, pp. 451-454, 1999.