

Implementation of RISC Processor Using Radix-4 & Radix-8 Booth Encoded Multi-Modulus Multipliers

V.Sandhya

PG Scholar in VLSI Design,
Department of ECE,

Dhanekula Institute of Engineering and Technology,
Ganguru, Krishna Dist., Andhra Pradesh, India.

K.Lakshmi Sowjanya

Assistant Professor,
Department of ECE,

Dhanekula Institute of Engineering and Technology,
Ganguru, Krishna Dist., Andhra Pradesh, India.

ABSTRACT:

Novel multi-modulus designs capable of performing the desired modulo operation for more than one modulus in Residue Number System (RNS) are explored to lower the hardware overhead of residue multiplication. Two multi-modulus multipliers that reuse the hardware resources amongst the modulo 2^n-1 , modulo 2^n and modulo 2^n+1 multipliers by virtue of their analogous number theoretical properties are proposed. The former employs the radix-4 Booth encoding algorithm and the latter employs the radix-8 Booth encoding algorithm. The proposed radix-4 and radix-8 Booth encoded multi-modulus multipliers, the modulo-reduced products for the moduli $2^n-1, 2^n$ and 2^n+1 are computed successively. The RNS technique is used to reduce the number of partial products. This project work deals with the one of the applications. i.e., basic RISC Processor. The algorithm is modeled in Verilog HDL and the RTL code for the modulo multiplier is synthesized using cadence RTL compiler where the design is targeted for 180nm and 45nm TSMC technology with proper constraints in terms of area and power. The layout is targeted by Cadence SOC Encounter. One of the Applications. i.e. RISC Processor.

INDEX TERMS:

Booth Multiplier, RISC Processor, RNS System.

I. INTRODUCTION:

Moduli choices of the forms $\{2^n-1, 2^n, 2^n+1\}$ and $\{2^n, 2^{n-1}, 2^{n-1}-1\}$ have received significant attention because they offer very efficient circuits when considering the area x time² product and efficient converters from and

to the binary system. Therefore, designing efficient modulo 2^n-1 multipliers is an interesting issue. RNS is an unconventional number representation that is widely employed in addition-multiplication intensive applications like digital filters, FFT/DFT and cryptography. MMA is classified into fixed and variable multi-modulus architectures. Fixed multi-modulus Architecture (FMA) performs the modulo operation with respect to multiple moduli simultaneously, thereby maintaining parallelism among the moduli. However, in Variable Multi-modulus Architecture (VMA), modulo operations are performed serially, resulting in greater hardware savings.

II. RADIX-4 BOOTH ENCODED MULTI-MODULUS MULTIPLIER:

The radix-4 booth encoded digit is formatted using three bits: a sign bit and one hot encoded magnitude bits, and the proposed multi-modulus radix-4 booth encoder using radix-4 booth encoder (BE2) slices and one mux3 block is shown in fig.1 Booth encoding technique is applied to multi-modulus multiplier modulo $(2^n-1, 2^n, 2^n+1)$ operations as the basis. The Radix-4 multiplication.

- The multiplier, Y in two's complement form can be written as in
- $Y = -Y_{n-1}2^{n-1} + \sum_{i=0}^{n-2} Y_i 2^i$; $0 < i < n-2$
- It can be written as
- $Y = (-2Y_{2i+1} + Y_{2i} + Y_{2i-1})2^{2i}$; $0 < i < n-2$
- Radix-4 Booth recoding encodes multiplier bits into $[-2, 2]$.

- Radix-8 Booth recoding encodes multiplier bits into [-4,4].

The radix-4 Booth encoded digit is formatted using three bits: a sign bit and one-hot encoded magnitude bits, $m1i$ and $m2i$. The proposed multi-modulus radix-4 Booth encoder using radix-4 Booth Encoder(BE2) slices and one MUX3 block. The proposed multi-modulus radix-4 Booth encoder using $N/2$ radix-4 Booth Encoder(BE2) slices and one MUX3 block. ($N=4$) The input corresponding to the modulus $2^n-1, 2^n$ and 2^n+1 is selected when is "00," "01," "10," respectively.

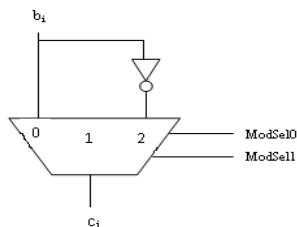


Fig :1.MUX3

III. MULTI-MODULUS PARTIAL PRODUCT ADDITION FOR RADIX-4 BOOTH ENCODING

In Radix-4 it has three stages. Partial product addition is the final stage. The first two stages are Multi-Modulus Partial Product Generation stage and Bias generation stage. A multi-modulus addition can be implemented using a MUX3 in the carry feedback path that selects from 0, c_{out} . The multi-modulus addition of partial products in a CSA tree and a parallel-prefix two-operand adders is illustrated. The parallel-prefix adder is constructed from the preprocessing(PP), prefix and post-processing blocks and the implementation of these blocks is shown fig2. The number of MUX3 blocks needed for multi-modulus partial product addition is $n/2+2$.

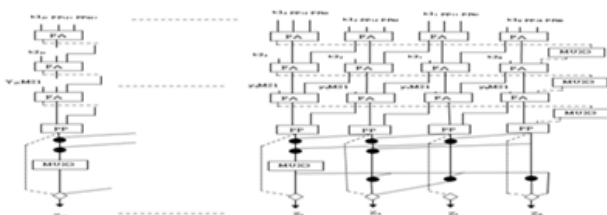


Fig. 2.Multi-modulus partial product addition for radix- 4

IV. MULTI-MODULUS PARTIAL PRODUCT ADDITION FOR RADIX-8 BOOTH ENCODING

In Radix-8 it has four stages. Partial product addition is the final stage. The first three stages are Multi-Modulus Partial Product Generation stage, Hard Multiple Generation stage and Bias generation stage. The most straightforward implementation of a final stage adder for two n-bit operands is a ripple carry adder, which requires a full adders(FAs). The carry-out of the i th FA is connected to the carry-in of the $(i+1)$ th FA. The proposed multi-modulus partial product addition for radix-8 Booth Encoding is shown in Fig.3. A multi-modulus addition can be implemented using a MUX3 in the carry feedback path that selects from 0 or c_{out} . The multi-modulus addition for partial products in a CSA tree and a parallel-prefix two-operand adder. The parallel-prefix adder is constructed from the preprocessing (PP), prefix and post-processing blocks and the implementation of these blocks. Number of MUX3 blocks needed for multi-modulus partial product addition is $n/3+5$.

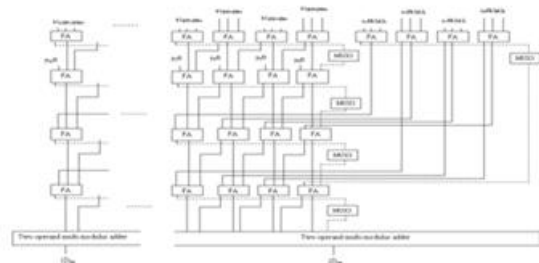


Fig:-3. Multi-modulus partial product addition for radix-8

V. RISC PROCESSOR:

The architecture of 64 bit RISC processor comprises of Control unit, general purpose register, ALU, Barrel shifter, universal shift register and accumulator. The control unit consists of two registers i.e. instruction register and instruction decoder. Instruction and data are fetched sequentially in order to reduce the latency in the machine cycle. Pipeline structure has been incorporated that further utilizes three execution cycle fetch, decode and execute. This pipeline structure helps in enhancing the speed of operation.

In fetch cycle, instruction and relevant data are inferred from the memory while in decode cycle, instruction and data drawn from the memory are bifurcated to activate component and data path for execution and in the execution cycle instruction is executed, data is manipulated and result is stored in the accumulator. The control unit accepts the op code and generate the signal that triggers the components and data path to work accordingly and perform the desired function. The control unit has two instruction decoders. These two decoders decode the instruction bits and direct the signal to either into ALU, universal shift register or barrel shift rotator. The operands are received from register A or register B. Upon receiving the operands from registers and the decoded instruction bits arithmetic and logical unit perform arithmetic and logical functions. Universal shift register and barrel shift rotator receives the input from register A and depending upon the decoded information perform the desired operation of either shifting or rotation and the result is stored in the accumulator register. Modules are the building blocks of a Processor.



Fig.4: Architecture of RISC Processor

VI.SIMULATION RESULTS:

If A=X, B=Y, depending on the selection lines the operations are performed. If Modsel0=0, Modsel1=0 then it performs 2^n-1 operation, Modsel0=0, Modsel1=1 then it performs 2^n operation and if Modsel0=1, Modsel1=0 then it performs 2^n+1 operation. The operation is same for all.

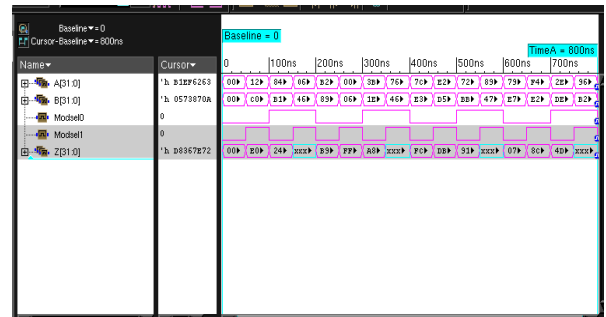


Fig.5:- Simulation Result for Radix4 32-bit

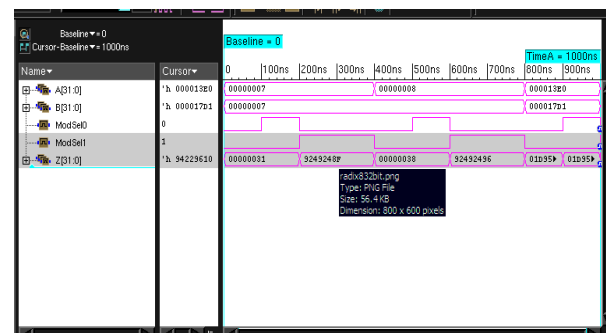


Fig.6:- Simulation Result for Radix8 32-bit

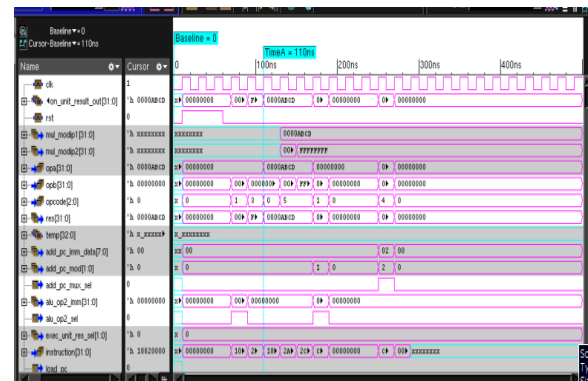


Fig.7:- Simulation Result for RISC Radix4 32-bit

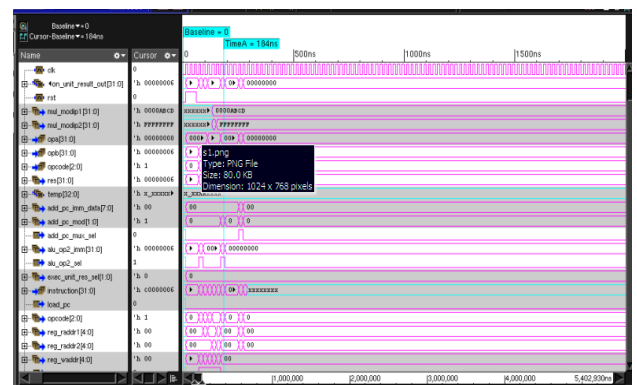


Fig.8:- Simulation Result for RISC Radix8 32-bit

VII. PERFORMANCE COMPARISON:

Design	Area		Power(μ w)		Delay(ns)	
	45	180	45	180	45	180
Technology(nm)	45	180	1.12	1.64	0.12	0.17
Radix-4(4-Bit)	49	53	10.38	11.50	0.12	0.17
Radix-4(32-Bit)	394	426	5.615	14.57	2.23	0.54
Radix-8(4-Bit)	402	422	316.1	14093.7	3.73	13.45
Radix-8(32-Bit)	8487	85668	9.05	225.612	0.44	1.99
RISCRadix-4(32Bit)	584	6204	10.39	276.67	0.44	2.12
RISCRadix-8(32Bit)	770	7950				

VIII. CONCLUSION & FUTURE SCOPE:

The equivalences in operations central to modulo multiplication, i.e., modulo negation, modulo reduction of binary weight, modulo multiplication by powers-of-two, and two-operand modulo addition for the three special moduli, 2^n-1 , 2^n and 2^n+1 were demonstrated. New radix-4 and radix-8 Booth encoded modulo 2^n multipliers with architectures comparable to those of the corresponding modulo 2^n-1 and modulo 2^n+1 multipliers were introduced. With the correlation among modulo 2^n-1 , modulo 2^n and modulo 2^n+1 operations as the basis, radix-4 and radix-8 Booth encoded multi-modulus multipliers that perform modulo multiplication for the three special moduli successively were developed. In RISC Processors and DSP applications modulo multiplier is major concern in calculations. Performing the multiplication process the products need to be reduced by using RNS. The future implementation of this paper is by using some other applications we can reduce the area and delay. Future work will be added by increasing the number of instructions and make a pipelined design with less clock cycles per instruction and includes the integration of the divider block.

IX. REFERENCES

1)B. Cao,C. H.Chang, andT.Srikanthan, “An efficient reverse converter for the 4-moduli set $\{2^n-1,2^n,2^n+1,2^{2n}+1\}$ based on the new Chinese Remainder Theorem,” IEEE Trans. Circuits Syst. I, Fundam. Theory Appl., vol. 50, no. 10, pp. 1296–1303, Oct. 2003.
 2)P. V. A. Mohan and A. B. Premkumar, “RNS-to-binary converters for two four-moduli sets $\{2^n-1,2^n,2^n+1,2^{2n}-1\}$ and $\{2^n-1,2^n,2^n+1,2^{n+1}+1\}$,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 54, no. 6, pp. 1245–1254, Jun. 2007.

3)P. V. A. Mohan, “RNS-to-binary converter for a new three-moduli set $\{2^{n+1}-1,2^n,2^n-1\}$,” IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 54, no. 9, pp. 775–779, Sep. 2007.
 4)G. C. Cardarilli, A. Del Re, A. Nannarelli, and M. Re, “Residue Number System reconfigurable data path,” in Proc. IEEE Int. Symp. Circuits Syst., Scottsdale, AZ, USA, May 2002, vol. 2, pp. 756–759.
 5)W. K. Jenkins and A. J. Mansen, “Variable word length DSP using serial- by-modulus residue arithmetic,” in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process., Minneapolis, MN, USA, Apr. 1993, vol. 3, pp. 89–92.
 6)G. C. Cardarilli, A. Del Re, A. Nannarelli, and M. Re, “Residue Number System reconfigurable data path,” in Proc. IEEE Int. Symp. Circuits Syst., Scottsdale, AZ, USA, May 2002, vol. 2, pp. 756–759.
 7)R. Muralidharan and C. H. Chang, “Fixed and variable multi-modulus squarer architectures for triple moduli base of RNS,” in Proc. IEEE Int. Symp. Circuits Syst., Taipei, Taiwan, May 2009, pp. 441–444.
 8)Rupali S. Balpande and Rashmi S. Keote.2011, “Design of FPGA based Instruction Fetch & Decode Module of 32-bit RISC (MIPS) Processor, International Conference on Communication Systems and Network Technologies pp. 409 – 413 [7]
 9)Samiappa Sakthikumar, S. Salivahanan, V. S. Kanchana Bhaaskaran, —16-Bit RISC Processor Design for Convolution Applicationl, IEEE-International Conference on Recent Trends in Information Technology, ICRTIT 2011, Anna University, Chennai. June 3-5, 2011. [9]
 10)Vijay R. Wadhankar, Vaishali Tehre, — A FPGA Implementation of a RISC Processor for Computer Architecturel, National Conference on Innovative Paradigms in Engineering & Technology (NCIPET-DEC 25,2012). Proceedings published by International Journal of Computer Applications.