Non Redundant Radix-4 Signed Digit Encoding DSP Accelerator

A. Krishna Veni[1] P. Sayanna[2]
[1]M. Tech. Student [2]Associate Professor
[1,2]Department of Electronics & Communication Engineering
[1,2]Sudheer Reddy College of Engineering and Technology (W), Bardipur, Nizamabad, Telangana India

Abstract— DSP accelerator has very prominent role in digital signal processing applications. Acceleration mainly performed in the application specific integrated circuit devices. DSP accelerator has perform various arithmetic operations to improve the performance of the Acceleration proposed technique is implemented i.e. NR4SD.i.e.Non Redundant Radix-4signed digit encoding technique, it has high performance compared with the modified booth algorithm and also it has lass decoding time for data transmission. In this paper the complete design of DSP accelerator is designed with NR4SD technique to improve the area, de-lay and power consumption. It has implemented with the verilog HDL and synthesize and simulated with the XILINX 14.5 ISE standard.

Key words: NR4SD, Pre-Encoded Multiplier, FCU, Carry Save Form, Modified Booth Encoding

I. INTRODUCTION

Modern embedded systems target high-end application do-mains requiring economical implementations of computationally intensive digital signal process (DSP) functions. The incorporation of heterogeneity through specialized hardware accelerators improves performance and reduces energy consumption. Although application-specific integrated circuits (ASICs) kind the perfect acceleration solution in terms of performance and power, their inflexibility leads to magnified atomic number 14 quality, as multiple instantiated ASICsare required to accelerate varied kernels. Several researchers have proposed the utilization of domain-specific coarse-grained reconfigurable accelerators so as to extend ASICs' flexibility while not significantly compromising their performance.

Highance versatile data paths have been projected to expeditiously map primitive or enchained operations found within the initial data-flow graph (DFG) of a kernel. The templates of complicated enchained operations area unit either extracted directly from the kernel's DFG or laid out in a predefined activity templet library. Style selections on the accelerator's datapath highly impact its potency. Existing works on coarse-grained reconfigurable datapaths in the main exploit architecture-level optimizations, e.g., magnified instruction level correspondence (ILP).

The domain specific design generation algorithms of vary the type and variety of computation units achieving a custom-made design structure. In, versatile architectures were projected exploiting ILP and operation chaining. Recently, Ansaloni et al. Adopted aggressive operation chaining to change the computation of entire subexpressions mistreatment multiple ALUs with heterogeneous arithmetic options. The said reconfigurable architecture sex clude arithmetic optimizations throughout the field of study synthesis and consider them solely at the interior circuit structure of primitive components, e.g., adders, through-out the logic synthesis.

However, research activities have shown that the arithmetic optimizations at higher abstraction levels than the structural circuit one considerably impact on the datapath performance. In, timing-driven optimizations supported carry-save (CS) arithmetic were performed at the post-Register Transfer Level (RTL) style stage. In, common subexpression elimination in metal computations is employed to optimize linear DSP circuits. Verma et al. developed transformation techniques on the application's DFG to maximize.

The utilization of metal arithmetic previous the actual datapath synthesis. The said metal improvement approaches target inflexible datapath, i.e., ASIC, implementations. Recently, Xydis et al. projected a versatile design combining the ILP and pipelining techniques with the CS-aware operation chaining. However, all the said solutions feature an inherent limitation, i.e., metal improvement is finite to merging only additions / subtractions.

A metal to binary conversion is inserted before every operation that differs from addition/subtraction, e.g., multiplication, thus, allocating multiple metal to binary conversions that heavily degrades performance owing to long carry propagations. In this transient, we have a tendency to propose a superior field of study theme for the synthesis of versatile hardware DSP accelerators by combining optimization techniques from each the design and arithmetic levels of abstraction. We have a tendency to introduce a versatile datapath design that exploits metal optimized templates of enchained operations. The projected architecture includes versatile procedure units (FCUs), which enable the execution of an outsized set of operation templates found in DSP kernels.

II. MODIFIED BOOTH ALGORITHM

Modified booth algorithm is used for signed digit multiplication, both A and B variable consist n=nk bits and B represented as in the modified booth form that is shown in below.

$$B = \langle b_{n-1} \dots b_0 \rangle_{2's} = -b_{2k-1}2^{2k-1} + \sum_{i=0}^{2k-2} b_i 2^i$$
$$= \langle \mathbf{b}_{k-1}^{MB} \dots \mathbf{b}_0^{MB} \rangle_{MB} = \sum_{j=0}^{k-1} \mathbf{b}_j^{MB} 2^{2j}.$$

B is formed within the limits of {-2,-1,0.1,2},and

$$\mathbf{b}_j^{MB} = -2b_{2j+1} + b_{2j} + b_{2j-1},$$

So, B is divided into three digit set like b_{2j-1}, b_{2j}, b_{2j+1}, it can be denoted as signed bits depends on MSB bits, each set has individual operation that has shown in below table:

b_{2j+1}	b_{2j}	b_{2j-1}	$\mathbf{b}_j^{MB}$	s_j	one_j	two_j
0	0	0	**0**	0	0	0
0	0	1	**+1**	0	1	0
0	1	0	**+1**	0	1	0
0	1	1	**+2**	0	0	1
1	0	0	**−2**	1	0	1
1	0	1	**−1**	1	1	0
1	1	0	**−1**	1	1	0
1	1	1	**0**	1	0	0

Table 1: Modified booth encoding

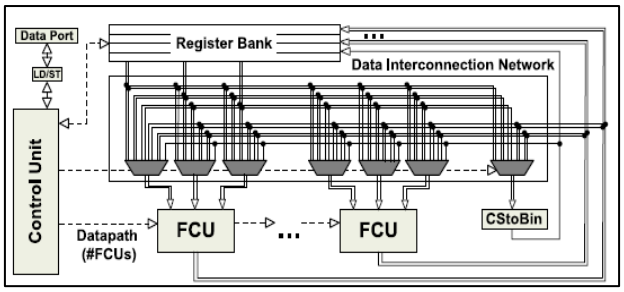

Fig. 1: Abstract form of the flexible datapat

III. PROPOSED FLEXIBLE ACCELERATOR

The projected versatile accelerator design is shown in Fig. 1. Each FCU operates directly on metal operands and produces knowledge in the same form1 for direct use of intermediate results. Each FCU operates on 16-bit operands.

Such a bit-length is adequate for the most DSP datapaths, however the field of study construct of the FCU can be foursquare custom-made for smaller or larger bit-lengths. The number of FCUs is decided at style time supported the ILP and space constraints obligatory by the designer. The CS to Bin module could be a ripple-carry adder and converts the metal kind to the two's complement one. The register bank consists of scratch registers and is used for storing intermediate results and sharing operands among the FCUs. Completely different DSP kernels (i.e., completely different register allocation and electronic communication patterns per kernel) is mapped onto the projected design mistreatment post-RTL datapath interconnection sharing techniques. The management unit drives the architecture (i.e., communication between the in-fo port and also the register bank, configuration words of the FCUs and choice signals for the multiplexers) in every clock cycle.

IV. NON REDUNDANT RADIX4 SIGNED DIGIT ENCODING

In this section Non redundant radix-4 signed digit encoding has represented, which is advanced method to the modified booth algorithm technique. Modified booth algorithm has worked for the moduli set {-2,-1,0.1,2} but this proposed technique has divided into NR4SD+ and NR4SD- with respect to {-2,-1,0.1} and {-1,0.1,2} moduli sets.

The proposed FCU is designed with the NR4SD techniques to improve the performance. The block diagram of proposed multipler with NR4SD is shown in the below figure:

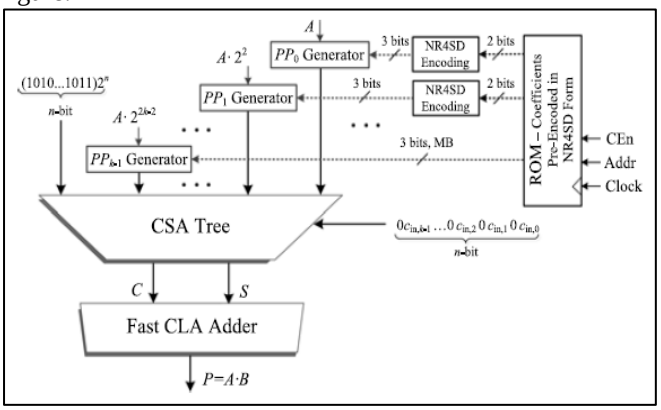

Fig. 2: Proposed FCU

The above diagram represents the proposed multiplier with NR4SD technique, in this the variable B has divided into the sets this are initially stored in the memory ROM. These bits are nothing but n^{-}_{2j+1}, n^{+}_{2j} for NR4SD+ and n^{+}_{2j+1}, n^{-}_{2j} for NR4SD-.

These implementations are shown in the below figure.

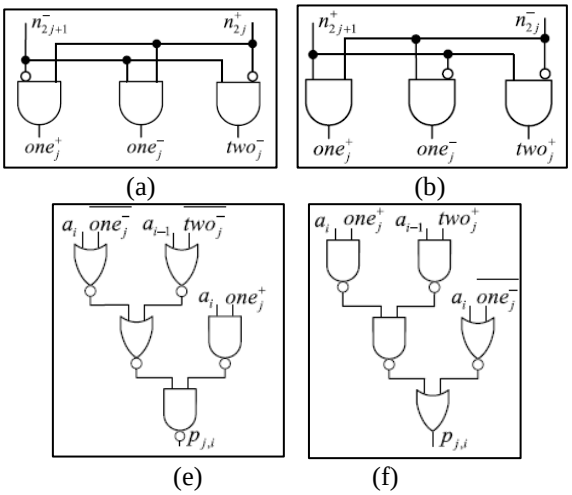

Fig. 3: NR4SD+ and NR4SD-

From the bits of n^{-}_{2j+1},n^{+}_{2j} and n^{+}_{2j+1},n^{-}_{2j} ,three moduli bits are generated one^{+}j, one^{-}j, two^{-}j and one^{+}j, one^{-}j, two^{-}j , these are shown in the above diagram. From these the required partial products PP bits are generated as shown in the below figures for NR4SD+ and NR4SD-.

V. IMPLEMENTATION RESULTS

We executed in Verilog the multiplier plans of Table 5.The PPGs for the NR4SD1, NR4SDþ multipliers (Figs. 4c and 4d, separately) contain countless since all the A bits are supplemented if there should arise an occurrence of a negative digit. In request to maintain a strategic distance from these inverters and, in this way, diminish the territory/ control/deferral of NR4SD1, NR4SDþ pre-encoded multipliers, the PPGs for the NR4SD1, NR4SDþ multipliers were outlined in view of primitive NAND and NOR entryways, and supplanted by Figs. 4e and 4f, separately. The CSA tree and CLA snake were transported in from Synopsys. DesignWare library. The ROM for the 2's supplement or preencoded coefficients is a synchronous ROM of 512 words regularly met at DSP frameworks, e.g., discourse CoDecs or sound sifting [20]. The width of every ROM relies on upon the multiplier engineering (Table 5). A limited state machine synchronized the information stream and the multiplier operation yet was not considered in the region/control figurings.

VI. SYNTHESIS AND SIMULATION RESULTS

In this paper the new design NR4SD technique is used to design the FCU unit for DSP acclelarator and its implemented with the XILINX ISE 14.5 simulation tool and implemented with Verilog HDL. The RTL diagram and simulation results are displayed below.

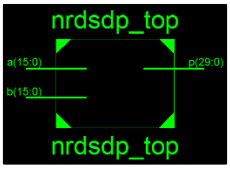

Fig. 4: Top level schematic diagram

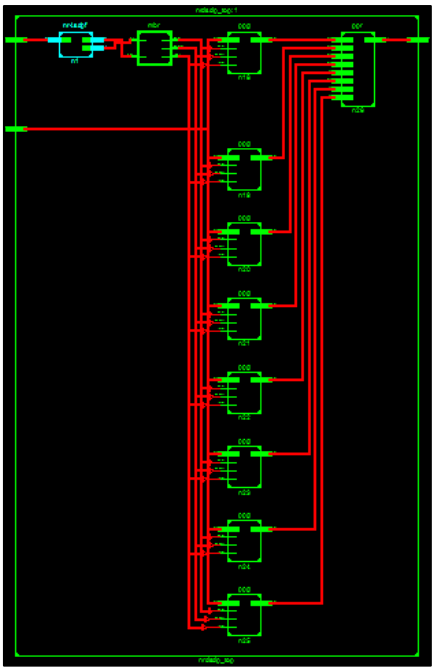

Fig. 5: Internal architectures of RTL diagram

A. Synthesis Result

Device Utilization Summary (estima	
Logic Utilization	**Used**
Number of Slice LUTs	166
Number of fully used LUT-FF pairs	0
Number of bonded IOBs	158

```
Total                      13.651ns (3.052ns logic, 10.599ns route)
                                    (22.4% logic, 77.6% route)
```

Fig. 6(a): Proposed NR4SD based FCU unit results:

Device Utilization Summary (estima	
Logic Utilization	**Used**
Number of Slice LUTs	311
Number of fully used LUT-FF pairs	0
Number of bonded IOBs	62

```
Total                      14.665ns (2.135ns logic, 12.530ns route)
                                    (14.6% logic, 85.4% route)
```

Fig. 6(b): MB based FCU unit results

Fig. 7: Simulation result

VII. CONCLUSION

In this paper it consist the design of flexible unit with non-redundant radix-4 sign encoding technique for DSP accelerator unit. It performs several arithmetic operations, the experimental results shows the proposed FCU unit has better performance compared with the MB encoding based FCU unit with the area and delay.

REFERENCES

[1] G. W. Reitwiesner, "Binary arithmetic," Adv. Comput., vol. 1, pp. 231–308, 1960.
[2] K. K. Parhi, VLSI Digital Signal Processing Systems: Design and Implementation, Hoboken, NJ, USA: Wiley, 2007.
[3] Y.-E. Kim, K.-J. Cho, J.-G. Chung, and X. Huang, "CSD-based programmable multiplier design for predetermined coefficient groups," IEICE Trans. Fundam. Electron. Commun. Comput. Sci., vol. 93, no. 1, pp. 324–326, 2010.
[4] O. Macsorley, "High-speed arithmetic in binary computers," Proc. IRE, vol. 49, no. 1, pp. 67–91, Jan. 1961.
[5] W.-C. Yeh and C.-W. Jen, "High-speed booth encoded parallel multiplier design," IEEE Trans. Comput., vol. 49, no. 7, pp. 692–701, Jul. 2000.
[6] Z. Huang, "High-level optimization techniques for low-power multiplier design," Ph.D. dissertation, Dept. Comput. Sci., Univ. California, Los Angeles, CA, USA, 2003.
[7] Z. Huang and M. Ercegovac, "High-performance low-power left-to-right array multiplier design," IEEE Trans. Comput., vol. 54, no. 3, pp. 272–283, Mar. 2005.
[8] Y.-E. Kim, K.-J. Cho, and J.-G. Chung, "Low power small area modified booth multiplier design for predetermined coefficients," IEICE Trans. Fundam. Electron. Commun. Comput. Sci., vol. E90-A, no. 3, pp. 694–697, Mar. 2007.
[9] C. Wang, W.-S. Gan, C. C. Jong, and J. Luo, "A low-cost 256-point FFT processor for portable speech and audio applications," in Proc. Int. Symp. Integr. Circuits, Sep. 2007, pp. 81–84.
[10] A. Jacobson, D. Truong, and B. Baas, "The design of a reconfigurable continuous- flow mixed-radix FFT processor," in Proc. IEEE Int. Symp. Circuits Syst., May 2009, pp. 1133–1136.
[11] Y. T. Han, J. S. Koh, and S. H. Kwon, "Synthesis filter for mpeg-2 audio decoder," Patent US 5 812 979, Sep. 1998.
[12] M. Kolluru, "Audio decoder core constants rom optimization," Patent US6108633, Aug. 2000.
[13] H.-Y. Lin, Y.-C. Chao, C.-H. Chen, B.-D. Liu, and J.-F. Yang, "Combined 2-d transform and quantization architectures for h.264 video coders," in Proc. IEEE Int. Symp. Circuits Syst., May. 2005, vol. 2, pp. 1802–1805.
[14] G. Pastuszak, "A high-performance architecture of the double-mode binary coder for h.264.avc," IEEE Trans. Circuits Syst. Video Technol., vol. 18, no. 7, pp. 949–960, Jul. 2008.
[15] J. Park, K. Muhammad, and K. Roy, "High-performance fir filter design based on sharing multiplication," IEEE Trans. Very Large Scale Integr. Syst., vol. 11, no. 2, pp. 244–253, Apr. 2003.
[16] K.-S. Chong, B.-H. Gwee, and J. S. Chang, "A 16-channel low-power nonuniform spaced filter bank core

for digital hearing aids," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 53, no. 9, pp. 853–857, Sep. 2006.
[17] B. Paul, S. Fujita, and M. Okajima, "Rom-based logic (RBL) design: A lowpower 16 bit multiplier," IEEE J. Solid-State Circuits, vol. 44, no. 11, pp. 2935–2942, Nov. 2009.
[18] M. D. Ercegovac and T. Lang, "Multiplication," in Digital Arithmetic. San Francisco, CA, USA: Morgan Kaufmann, 2004, pp. 181–245.
[19] N. Weste and D. Harris, CMOS VLSI Design: A Circuits and Systems Perspective, 4th ed. Reading, MA, USA: Addison-Wesley, 2010.

