Limited Carry-Propagate Multiply-Accumulate Unit Design for Reconfigurable Systems

Ugur Cini[1], Gokhan Kocyigit[1]
[1]*Department of Electrical Electronics Engineering, Trakya University, 22180 Edirne, Turkey*
ugurcini@trakya.edu.tr

***Abstract*—Counter and compressor arrays are frequently employed in multiplier design to efficiently reduce partial products in VLSI design. On the other hand, in reconfigurable systems, fast carry chains boost the performance of carry-propagate adders. So that, in reconfigurable systems, to save logic element area, counter and compressor trees are not employed as much since they require more area than carry-propagate scheme. In this work, carry-propagate multi-operand adders are employed in smaller blocks and the outputs are merged using double carry-save encoding to increase performance in reconfigurable systems. Hence, a more compact structure is achieved, compared to full redundant partial product reduction scheme providing comparable speed performance with counter array based carry-save structure. To show the effectiveness of the implementation, fused multiply-accumulate (MAC) units are designed for various bit-widths. The structure is implemented on Altera™ Stratix III and Cyclone III FPGAs and the results show that, using least depth of pipeline, the throughput is better than regular carry-propagate and fully redundant carry-save reduction schemes.**

***Index Terms*—Multiply-accumulate unit; multi-operand adder; redundant numbers; carry-save arithmetic; FPGA arithmetic.**

I. INTRODUCTION

Multiplication and multiply-accumulate operations are most frequently used blocks in digital signal processing [1]–[4]. Speed and area optimization of these blocks is crucial in VLSI and reconfigurable system design. Usually, the performance bottleneck in a DSP system is the multiply-accumulate (MAC) unit performance. For high performance multiplier and MAC unit design, counter and compressor trees are mostly employed in VLSI design. However, it is not always the case in reconfigurable system design. The reason is that, in contemporary reconfigurable systems (such as Altera and Xilinx FPGA families) fast carry chains [5]–[8] are available, which provides high performance in carry-propagate adder schemes. As stated in [8], contemporary FPGA logic elements are configured in two modes as logic mode and arithmetic mode. When using Altera FPGAs, arithmetic mode is automatically detected whenever the Quartus II design compiler is used for synthesis if adder structure is available in the design. As a result, carry chains are active in arithmetic mode. However, it is always the case that carry-propagate addition delay is linear with operand bit size. So that, carry-propagate scheme becomes inefficient if the bit size of the operands are very large.

In [8], various optimization methods are applied to implement various carry-save operators for partial product reduction and multi-operand addition. Although optimization over carry-save adder schemes are implemented, the area requirement is still higher than carry propagate scheme. In [8] and [9] it is reported that (6, 3) counter arrays give best performance result for the reduction of partial products and multi-operand addition input operand reduction, whenever 6-input LUT structures are implemented. In [9], register-to-register delays for various reduction schemes are analysed, which gives (6, 3) reduction gives best performance result. However, area requirement is always higher than carry propagate schemes.

In this work, multiple multi-operand carry-propagate adder blocks are implemented for the design on multiply-accumulate (MAC) units. General representation of a MAC unit is depicted in Fig. 1. MAC unit consists of a multiplier followed by an adder, where in some applications multiply and add operations are merged for high performance which is named as fused multiply-add units. Partial product reduction scheme of a multiplier and a fused multiply-add unit has a diamond shape structure [10]. The centre of the partial product reduction scheme has higher bit density compared to both ends of the partial product reduction scheme. In this work, the partial product scheme is divided into four blocks having approximately equivalent delays. So that carry-propagate adder delay is divided into equivalent-delay sub-blocks. After partitioning, the outputs of the four multi-operand addition blocks are merged into a double carry-save structure. So that, carry-propagation delay is avoided in the merging operation at the end.

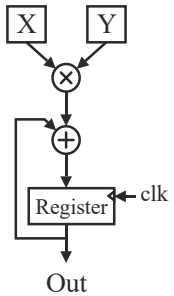

Fig. 1. General representation of MAC unit.

Manuscript received 26 September, 2016; accepted 14 January, 2017.

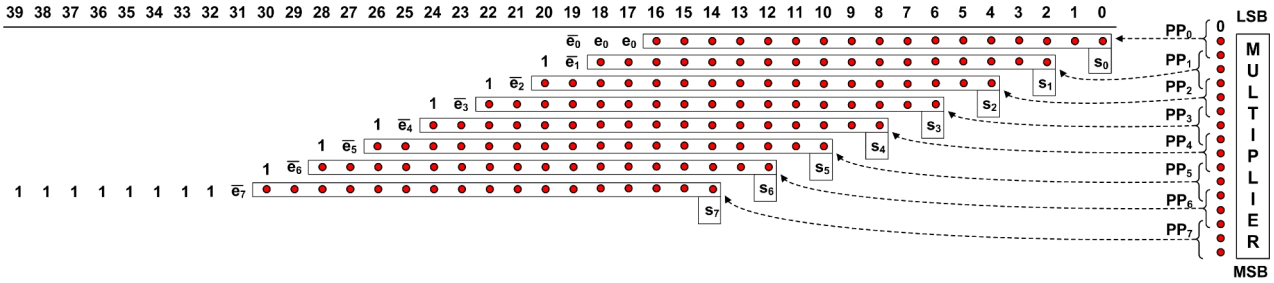

Fig. 2. Modified Booth encoding scheme with sign extension to 40-bits.

As a result, a carry-propagate and double carry-save hybrid structure is achieved having low area requirement and fast carry-logic advantage of the carry propagate scheme and carry-free output encoding of double carry-save structure. The multi-operand addition blocks can also be pipelined in the structure to further improve the performance of the multiply accumulate unit. The output encoding of the MAC unit is kept in redundant form at the output. However, it is easily converted to conventional binary form using a standard ternary adder after the MAC operation is completed, which requires an extra clock delay only. The structure is tested for various sizes of MAC units as 16×16 bit and 32 × 32 bit inputs with 40-bit and 72-bit outputs, respectively. Larger output digit extension avoids overflow for recursive MAC operations. The results are compared with full carry-propagate addition based MAC units, hardware multiplier based MAC units, and fully redundant carry-save based MAC units. The proposed scheme provides best throughput performance with average area requirement.

II. HYBRID MAC ARCHITECTURE

As stated before, full carry-save partial product reduction has extensive area requirement. So that, carry-propagate multi-operand adders are employed for the partial product reduction scheme. For the generation of partial products in the multiplication phase, modified Booth encoding scheme is employed. In Fig. 2, 16 × 16-bit multiplier input Booth encoding scheme with 40-bit sign extension is depicted. The Booth encoding scheme is explained in [9], [10]. Here, also sign extension is employed for recursive multiply-accumulate operations in order to avoid overflow quickly.

Ripple carry adder (RCA) arithmetic provides high performance in most of the FPGA systems due to the fact that fast carry chains in the fabric boosts the performance. However, as the operand bit sizes increase, the structure becomes inefficient, since RCA adder has linear delay with bit size. Table I shows ripple carry adder with fast carry logic, (4, 2) compressor array and (6, 3) counter array delay and area requirements.

TABLE I. VARIOUS ADDER OPERATOR DELAYS FOR STRATIX III.

	RCA (fast-carry logic)		**(4,2) compressor**		**(6,3) counter**	
Bit-width	Delay (ns)	Area (LUT)	Delay (ns)	Area (LUT)	Delay (ns)	Area (LUT)
16-bit	1.25	16	1.12	48	0.40	48
24-bit	1.90	24	1.16	72	0.39	72
32-bit	2.60	32	1.15	96	0.40	96
64-bit	5.20	64	1.16	192	0.40	192

In [9], a fully redundant MAC unit with (6, 3) scheme is proposed. In [9], delay of each reduction operator is given as register-to-register delay. Here, in Table I, revised delay table is given as combinational delay blocks of each reduction operator. As shown in Table I, (6, 3) counter array provides best partial product reduction performance for the Stratix III FPGAs, which is 6-input LUT based structure. However, whenever area is a consideration, fast carry logic enabled ripple carry adder (RCA) should be selected. It is interesting that, (6, 3) counter array performance is much better than (4, 2) compressor reduction scheme for the 6-input LUT based structures, as Altera's Stratix III is selected.

In this paper, double carry save [11], [12] output encoding based MAC unit is proposed; by employing carry propagate sub-block implementation. The multiply and accumulate operations are merged under the same reduction scheme, i.e. fused multiply-add operation is performed. Partial product scheme after Booth encoding is shown in Fig. 3, where partial products are fed into four separate multi-operand adder blocks. The partial products and the accumulate output are fed-back from the output is also shown in Fig. 3. Here, the operands to be added up, is divided into four approximately equivalent-delay multi-operand addition blocks. The reduction using smaller length multi-operand addition blocks would be faster than a unified adder block.

Detailed multiply-add operation after the Booth encoding for 16 × 16-bit input and 40-bit output is shown in Fig. 4. There exist four outputs from the multi-operand addition operations, which reside inside three output components. As redundant carry-save output scheme is composed of two binary outputs, here there exist three. So that, the output encoding scheme is equivalent to double carry-save output encoding [9]. Sign bit of as s_7 for the seventh partial product sign bit also resides inside the empty slot at the output block in proper digit level. So that 2nd multi-operand adder block and 4th multi-operand adder block are both 5-operand adders in the revised scheme which is shown in Fig. 4. In the proposed scheme, multi-operand adders can also be pipelined to improve the throughput. The result of the MAC unit is composed of three components which is named as double carry-save encoding scheme, and, it can easily converted to standard binary representation using a ternary adder as shown in Fig. 4. The output encoding scheme presented here is also named as stored-double-carry system in [11], and helps to increase performance by less logic depth in the structure. The proposed scheme is a composition of carry-propagate multi-operand adder clusters with redundant output encoding.

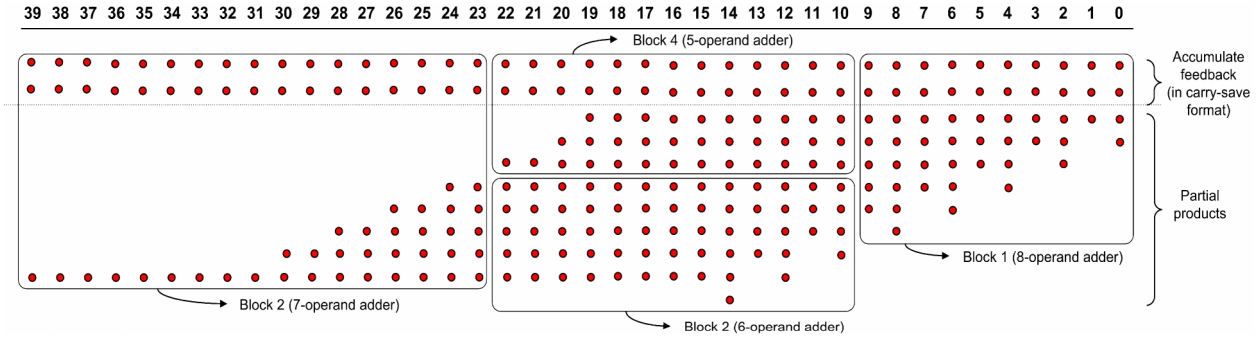

Fig. 3. Dividing the partial products into sub-blocks for multi-operand adder implementation.

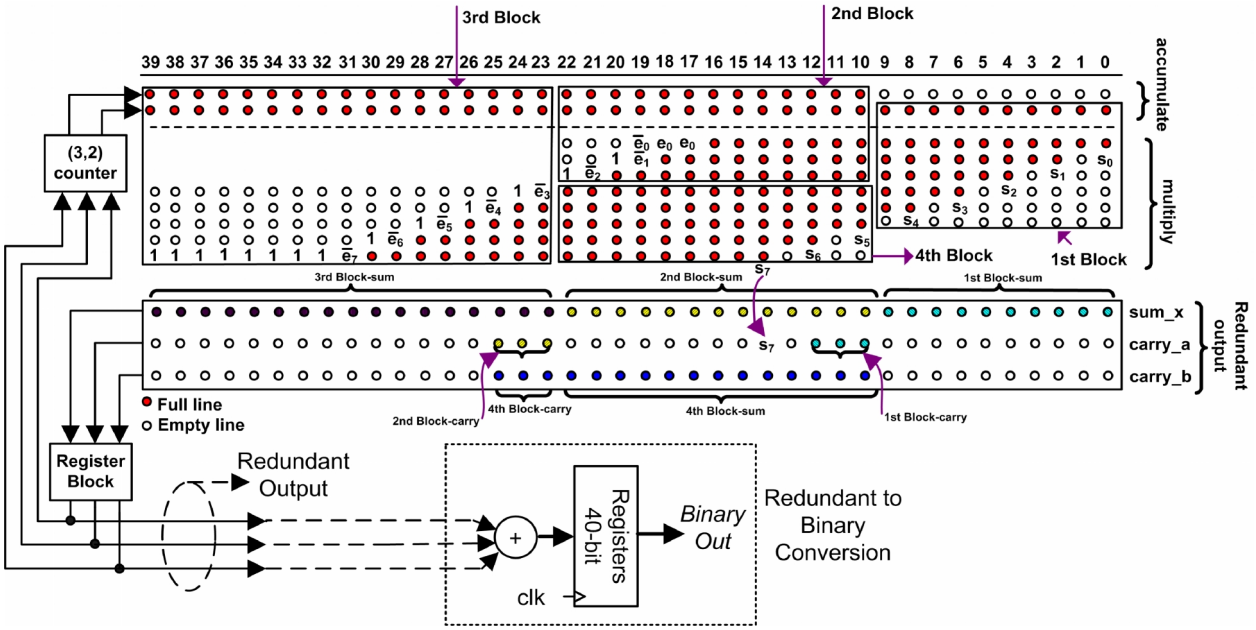

Fig. 4. Proposed multiply-accumulate architecture.

The redundant output encoding can be converted to standard binary form with a three-operand adder block at the output, which is also shown in Fig. 4.

III. Results and Discussions

The performance of the proposed scheme is compared to various implementations with equivalent functionality. These are, soft multiplier based MAC units with and without various pipeline stages, and hard multiplier based MAC units with and without pipeline stages. Also, a recent MAC unit design proposed in [9] is also included in the comparison scheme. 32 × 32-bit MAC unit is also developed and implemented as well and the results are shown in Fig. 5. The performance measurements are made on Altera Stratix III and Cyclone III FPGAs. According to the results, proposed scheme with a single level of pipeline is gives the best throughput results. A detailed hardware requirement analysis for 16×16-bit MAC unit with 40-bit output extension results are given in Table II. It is shown that the proposed scheme requires less area than fully redundant (6, 3) counter based implementation proposed in [9]. Redundant to binary conversion is given as +1 in clock delay for fair comparison with other implementations. The proposed scheme with a single pipeline stage provides highest throughput with 3 clock delays. The proposed scheme with 1-level pipeline has 28 % more resource requirement compared to soft multiplier with 2-level pipeline scheme. However, the proposed scheme is 57 % faster, as shown in Table II. The proposed scheme provides best performance compared to various conventional MAC unit implementations as shown in Table II.

TABLE II. COMPARISON OF 16 × 16-BIT MAC UNITS FOR STRATIX III.

Structure	Resource Usage	Speed (MHz)	Clock Delay
Soft Multiplier (no pipeline)	258 ALUT + 72 Reg.	124	1
Soft Multiplier: 1-level pipeline	259 ALUT + 104 Reg.	160	2
Soft Multiplier: 2-level pipeline	263 ALUT + 190 Reg.	210	3
Hardware Multiplier (no pipeline)	1 DSP Block + 40 ALUT + 40 Reg.	141	1
Hardware Multiplier: 1-level pipeline	1 DSP Block + 40 ALUT + 80 Reg.	261	2
(6, 3) counter based redundant MAC [9]	418 ALUT + 178 Reg.	286	1+1
Proposed (No pipeline)	336 ALUT + 101 Reg.	220	1+1
Proposed: 1-level pipeline	**336 ALUT + 279 Reg.**	**330**	**2+1**

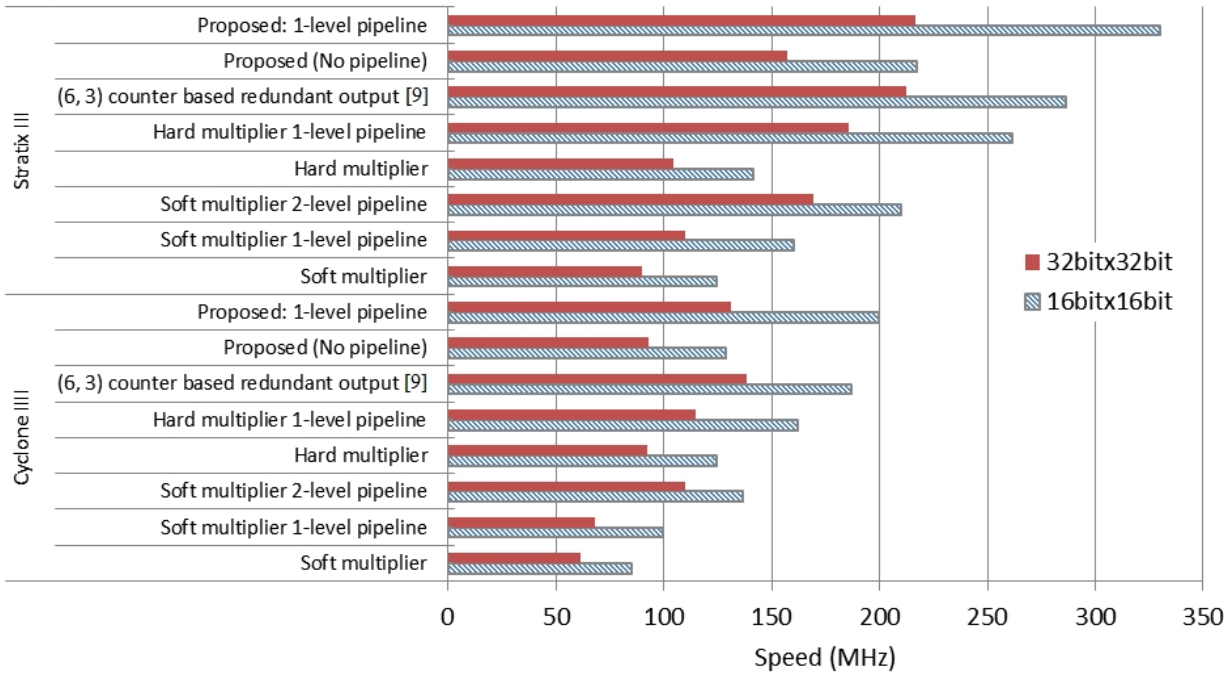

Fig. 5. Performance measurements of various MAC unit architectures.

Although area and delay is a trade-off in the compared designs, a high amount of parallelism and pipeline also limits the performance of the system. The proposed system provides a good trade-off point for high performance without excessive register and LUT increase.

IV. CONCLUSIONS

In this paper, carry-propagate multi-operand adder based sub-blocks with redundant output encoding architecture is developed. The hybrid design employs both carry-propagate adders and redundant output encoding which requires less area than full carry-save architecture [9], and have highest throughput compared to various MAC unit implementations. The hybrid structure is composed of a modified Booth encoding stage followed by clustered multi-operand adder blocks. The proposed structure provides low logic depth providing fast multiply-add operation. The proposed hybrid structure is advantageous whenever high performance is desired. The hybrid structure is also scalable such that larger size MAC units can be synthesized using similar design strategy, which is an advantage over hard multiplier based MAC units.

REFERENCES

[1] K. Parhi, *VLSI Digital Signal Processing Systems*. John Wiley & Sons, 1999, pp. 10–25.

[2] M. Y. Zulfikar, S. A. Abbasi, A. R. M. Alamoud, "FPGA based Walsh and inverse Walsh transforms for signal processing", *Elektronika ir Electrotechnika*, vol. 18, no. 8, 2012. [Online]. Available: http://dx.doi.org/10.5755/j01.eee.18.8.2601

[3] A. S. N. Mokhtar, M. B. I. Raez, M. Marufuzzaman, M. A. M. Ali, "Hardware implementation of a high speed inverse Park transformation using CORDIC and PLL for FOC brushless servo drive", *Elektronika ir Electrotechnika*, vol. 19, no. 3, 2013. [Online]. Available: http://dx.doi.org/10.5755/j01.eee.19.3.1267

[4] T. Tuncer, "Implementation of duplicate TRNG on FPGA by using two different randomness source", *Elektronika ir Electrotechnika*, vol. 21, no. 4, 2015. [Online]. Available: http://dx.doi.org/10.5755/j01.eee.21.4.12779

[5] Xilinx Inc., "Virtex-6 family overview", *Xilinx Datasheet DS105*, 2012, pp. 1–11.

[6] *Stratix III Device Handbook.* Altera Corp., 2011, ch. 2.

[7] *Cyclone III Device Handbook*. Altera Corp., 2012, ch. 2.

[8] H. Parandeh-Afshar, A. Neogy, P. Brisk, P. Ienne, "Compressor tree synthesis on commercial high performance FPGAs", *ACM Trans. Reconfigurable Technology and Systems*, vol. 4, 2011. [Online]. Available: http://dx.doi.org /10.1145/2068716.2068725

[9] U. Cini, O. Kurt, "A MAC unit with double carry-save scheme suitable for 6-input LUT based reconfigurable systems", in *IEEE Proc. Int. Conf. Electronics, Circuits, and Systems (ICECS 2015)*, Cairo, 2015, pp. 649–652. [Online]. Available: https://doi.org/10.1109/ICECS.2015.7440400

[10] M. D. Ercegovac, T. Lang, *Digital arithmetic*. Morgan Kaufmann, 2003, ch. 4.

[11] B. Parhami, "Generalized signed-digit number systems: a unifying framework for redundant number representations", *IEEE Trans. Computers*, vol. 39, no. 1, pp. 89–98, 1990. [Online]. Available: https://doi.org/10.1109/12.46283

[12] U. Cini, M. Aktan, A. Morgul, "An alternative carry-save arithmetic for new generation field programmable gate arrays", *Turk J Elec Eng & Comp Sci*., vol. 24, pp. 435–447, 2016. [Online]. Available: https://doi.org/10.3906/elk-1306-184