

Design Alternatives for Parallel Saturating Multioperand Adders

Pablo I. Balzola, Michael J. Schulte, and Jie Ruan
EECS Dept, Lehigh University
Bethlehem, PA 18015

John Glossner and Erdem Hokenek
Sandbridge Technologies
White Plains, NY 10601

Abstract

Parallel saturating multioperand adders significantly improve the performance of GSM speech coders by giving compilers and assembly language programmers the ability to parallelize loops containing saturating dot products, while maintaining GSM compliant results. This paper presents four designs for parallel saturating multioperand adders. These designs have at most one carry-propagate adder on their critical delay path, yet produce the same results that would be obtained if the additions were performed serially with saturation after each addition. The four parallel designs offer tradeoffs in terms of area, worst case delay, and dot product latency. Compared to a 5-input serial design, the 5-input parallel designs have delays up to 3.51 times shorter.

1. Introduction

Most digital signal processors (DSPs) provide support for two's complement saturating arithmetic. With saturating arithmetic, results that overflow are saturated to the most positive or most negative number [10]. Since saturating arithmetic operations are not associative, changing the order of the operations can lead to incorrect results.

Global System for Mobile (GSM) communication speech coders and other DSP applications frequently perform saturating dot products on long vectors with saturation after each arithmetic operation [4]. The GSM standards require that the results produced by GSM compliant speech coders be bit-per-bit identical to the results produced when the operations are performed serially [11]. This ensures the numerical integrity of GSM speech coders and keeps compatibility across a variety of platforms. However, it severely limits the performance of DSPs with several parallel arithmetic units, since saturating arithmetic operations must be performed serially to maintain GSM compliance.

To overcome this limitation, a technique is presented in [13] for designing processors that perform saturating dot products. This approach is shown in Figure 1. In the first cy-

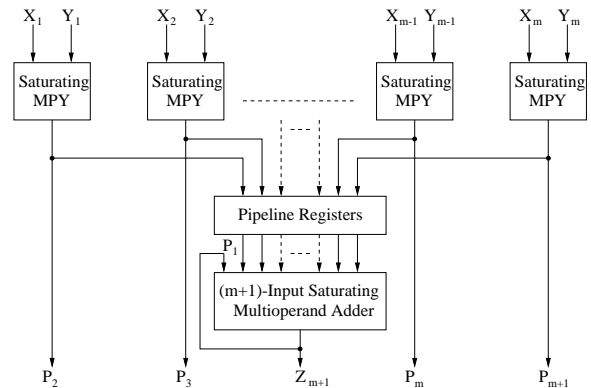


Figure 1. Parallel Saturating Arithmetic Units

cle, m saturating multipliers compute n -bit saturated products, P_2 to P_{m+1} . In the next cycle, an $(m + 1)$ -input n -bit parallel saturating multioperand adder (SMA) combines the outputs from each of the multiply units with an accumulator/feedback value P_1 , and m new saturated products are computed. The parallel SMA is designed so that it only has one fast carry-propagate adder (CPA), on the critical delay path, yet produces the same result as performing the additions serially with saturation after each addition. By pipelining the design shown in Figure 1, m additional elements of a saturating dot product can be generated and added every cycle.

This paper presents four designs for parallel SMAs, which are based on the optimized parallel SMA presented in [13]. These designs differ based on the format of the feedback operand, P_1 , and whether or not internal pipeline registers are used to reduce the delay of the feedback path. Section 2 gives an overview of serial SMA designs. Section 3 presents alternative parallel SMA designs, and discusses their hardware requirements and worst case delay paths. Section 4 provides area and delay estimates for 5-input parallel SMAs. Section 5 gives our conclusions. The notation used in this paper is as follows: upper case variables denote n -bit signals, lower case variables denote 1-bit signals, and variables starting with s denote the sign of the variable.

2. Serial SMAs

A simple approach for computing saturating dot products is to use one saturating multiplier and one 2-input saturating adder. As presented in [14], the saturating multiplier and 2-input saturating adder can be constructed by adding saturation logic to a conventional multiplier and CPA. Each cycle, the saturating multiplier computes a new saturated product, which is added to a feedback/accumulator value using the saturating adder. The saturating multiplier and adder can be combined to form a saturating multiply-accumulate (MAC) unit, with only one CPA on the critical delay path and about the same cycle time as a conventional MAC unit [14]. Although this approach allows for a relatively short cycle time, a p element dot product has a latency of approximately p cycles.

Since high-performance DSPs often have more than one multiplier or MAC unit [11, 6, 12], the latency of saturating dot products can be reduced to approximately p/m cycles by performing m saturating multiplies in parallel and then adding the saturated products and a feedback value using an $(m + 1)$ -input SMA. For example, Lucent's Voice Coding Processor [11] and DSP16000 [1] use two saturating MAC units and a 3-input parallel SMA [2] to improve the performance of GSM speech coders [3].

A simple method for designing an $(m + 1)$ -input serial SMA is to use m fast CPAs (e.g., carry-lookahead adders) and saturate the result of each addition that overflows. This method is illustrated in Figure 2 for a 5-input serial SMA. Since overflow occurs if and only if both inputs have the same sign and the output's sign differs [8], the overflow detection logic (ODL) computes

$$o_i = sz_i \cdot sp_{i+1} \cdot \overline{st_{i+1}} + \overline{sz_i} \cdot \overline{sp_{i+1}} \cdot st_{i+1} \quad (1)$$

where $sz_1 = sp_1$. The V-Gen components, in Figure 2, generate the values to which the result should saturate when overflow occurs [13]. As shown in Equation 2, the sign of only one of the inputs to the CPA, sp_{i+1} , is required to generate the saturation value

$$V_{i+1} = sp_{i+1} \overline{sp_{i+1}} \overline{sp_{i+1}} \cdots \overline{sp_{i+1}} \overline{sp_{i+1}} \quad (2)$$

When the result of the i^{th} addition overflows, $o_i = 1$ and $Z_{i+1} = V_{i+1}$; otherwise $o_i = 0$ and $Z_{i+1} = T_{i+1}$.

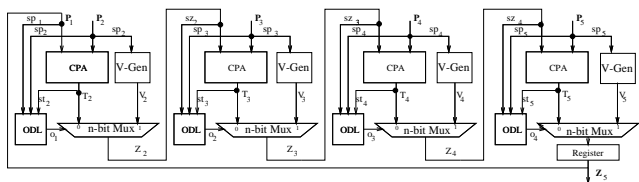


Figure 2. 5-Input Serial SMA.

An $(m + 1)$ -input serial SMA requires m CPAs, m V-Gens, m ODLs, and m n -bit multiplexers (Muxes). The critical delay path of this unit consists of m CPAs, m ODLs, and m n -bit Muxes. The component with the longest delay is the CPA.

The designs covered in the rest of this paper improve worst case delay of the serial SMA by reducing the number of CPAs on the critical delay path to at most one. Since Equation (1) is used to detect overflow, the signs of temporary sums still need to be computed. This is done using sign detection circuits (SDCs). An SDC uses logic that is similar to a fast CPA. Since only the sign of the result is required the SDC has significantly less area and delay than a fast CPA, as described in [9].

3. Parallel SMAs

In this section, four designs for parallel SMAs are presented. These designs differ based on the format of the accumulator/feedback operand, P_1 , and whether or not internal pipeline registers are used to reduce the delay of the feedback path. P_1 is either in two's complement or carry-save format. Having P_1 in carry-save format reduces the worst case delay by allowing the CPAs on the critical delay path to be replaced by SDCs and carry-save adders (CSAs). A CSA accepts three n -bit inputs and adds them to produce two n -bit outputs after only a full adder delay [8]. Adding internal pipeline registers allows values that do not depend on P_1 to be precomputed, which also reduces the worst case delay.

Parallel 5-input SMAs are used to illustrate the design alternatives, since they achieve a good tradeoff between the worst case delay path and the number of cycles needed to perform saturating dot products. Furthermore, 5-input SMAs support four parallel multipliers, which are found in current and emerging high-performance DSPs [5]. The general techniques presented in this paper can be extended to other values of m , and expressions are provided that show how the number of components and worst case delay path vary with m .

3.1. SMA with two's complement feedback

Design 1 uses the technique for constructing optimized parallel SMAs presented in [13]. The goal of this design is minimize area, while having only one CPA on the critical delay path. As an example of this design, a 5-input SMA with two's complement feedback (SMA-WTCF) is shown in Figure 3. An $(m + 1)$ -input SMA-WTCF computes $(m + 1)$ temporary sums, T_1 to T_{m+1} , as

$$T_i = V_i + \sum_{j=i+1}^{m+1} P_j \quad (3)$$

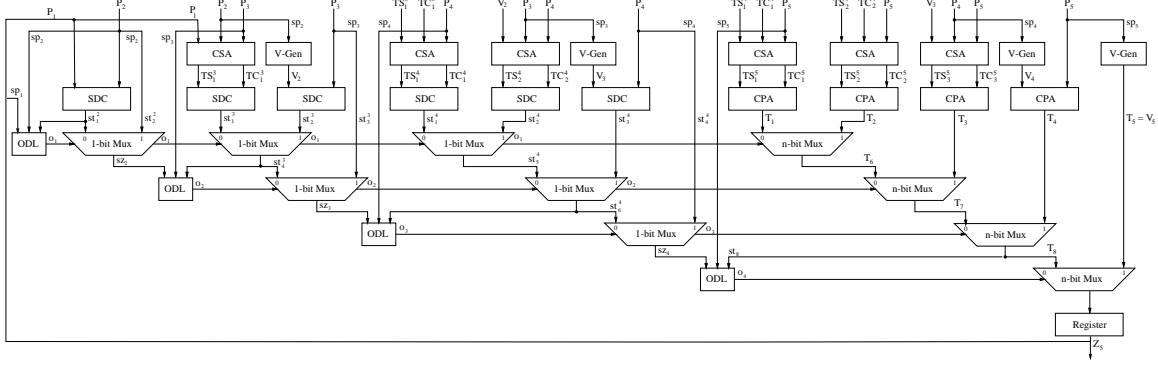


Figure 3. 5-Input SMA with Two's Complement Feedback.

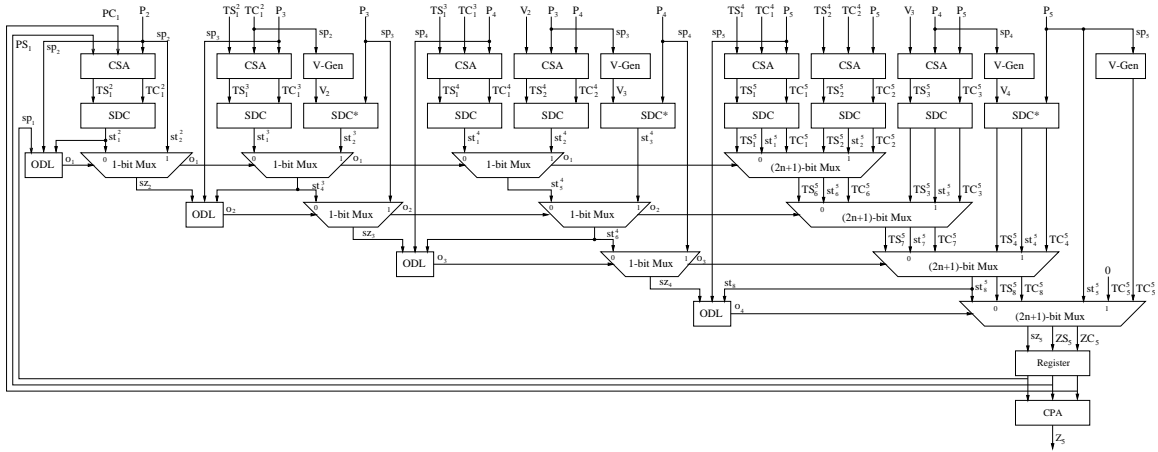


Figure 4. 5-Input SMA with Carry-Save Feedback.

where $V_1 = P_1$. T_i is the correctly saturated sum when adding P_i is the last addition to cause overflow and T_1 is the correctly saturated sum when none of the additions cause overflow. In parallel with this, the SMA uses CSAs, SDCs, V-Gens, ODLs, and 1-bit Muxes to compute the overflow bits, o_1 to o_m , where $o_i = 1$ if adding P_{i+1} causes overflow. Then, n -bit Muxes are used to select the appropriate temporary sum, based on the values of the overflow bits. Finally, the result Z_{m+1} is stored and fed back as P_1 in the next cycle.

An $(m + 1)$ -input SMA-WTCF has m CPAs, m ODLs, m V-Gens, m n -bit Muxes, $(m^2 - m)/2$ CSAs, $(m^2 - m)/2$ SDCs, $(m^2 - m)/2$ 1-bit Muxes, and one n -bit feedback register. Its critical delay path goes through $(m - 1)$ CSAs, one CPA, m n -bit Muxes, and one ODL¹.

An alternative approach for designing parallel $(m + 1)$ -input SMAs, with a single CPA on the critical delay path,

¹The components on the critical delay path may vary depending on m , n , and implementation techniques. The critical delay paths reported in this paper are for $m = 4$ and $n = 32$, using the implementation techniques described in Section 4.

is presented in [7]. This approach differs from Design 1 in that avoids computing individual saturation values, V_i , but instead computes $(2m + 1)$ temporary sums. It also uses more complex overflow detection logic and an n -bit $(2m + 1)$ -to-1 Mux to select the final result.

3.2. SMA with carry-save feedback

With Design 1, all the components on the critical path, except the CPA, have relatively low delay. Design 2 removes this CPA from the critical path by keeping the feedback/accumulator operand and the temporary sums in carry-save format. With this format, P_1 is represented by an n -bit sum vector, PS_1 , and an n -bit carry vector PC_1 , where $P_1 = PS_1 + PC_1$.

Figure 4 shows an example of a 5-input SMA with carry-save feedback (SMA-WCSF). Compared to Design 1, this design replaces m of the CPAs by SDCs, increases the width of the m n -bit Muxes and one n -bit feedback register to $(2n + 1)$ bits, and adds a CSA before the leftmost SDC and a CPA after the feedback register. The CPAs are

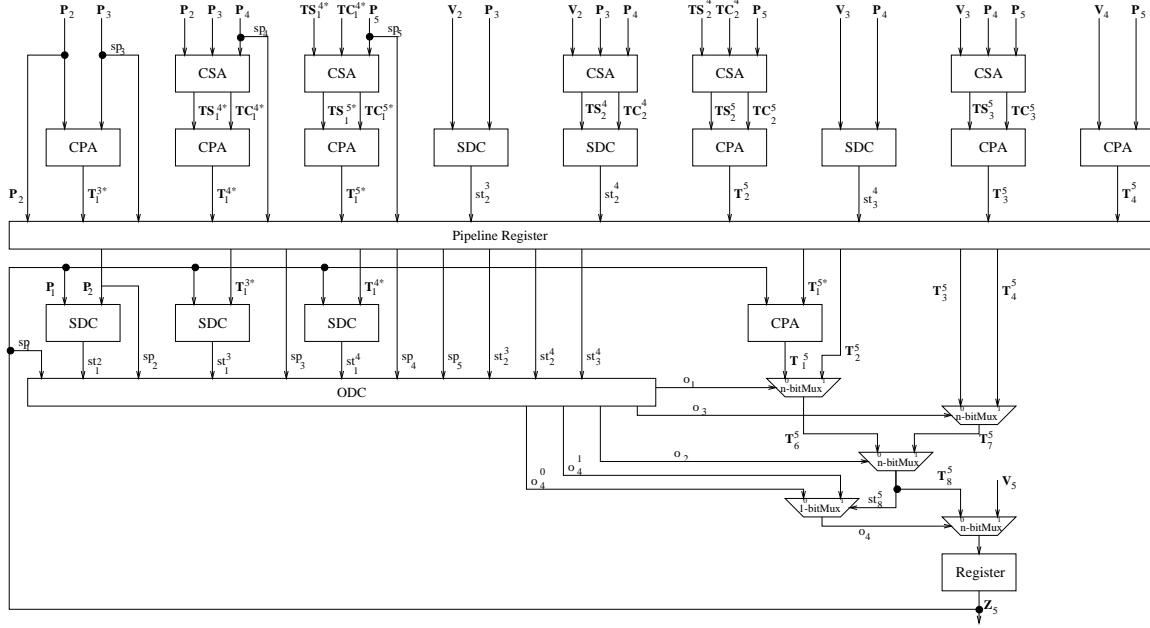


Figure 5. 5-Input Pipelined SMA with Two's Complement Feedback.

replaced by SDCs, since the signs of the temporary results are still needed to detect overflow. The Muxes and feedback register increase to $(2n + 1)$ -bits to handle the $2n$ -bit carry-save format and the sign bit. The additional CSA adds PS_1 , PC_1 , and P_2 to produce $TS_1^2 + TC_1^2 = PS_1 + PC_1 + P_2$. The CPA is used to produce a two's complement result after the entire saturated dot product is computed.

An $(m + 1)$ -input SMA-WCSF has one CPA, m ODLs, m V-Gens, $m(2n + 1)$ -bit Muxes, $(m^2 - m + 2)/2$ CSAs, $(m^2 + m)/2$ SDCs, $(m^2 - m)/2$ 1-bit Muxes, and one $(2n + 1)$ -bit feedback register. Its critical delay path goes through m CSAs, one SDC, $m(2n + 1)$ -bit Muxes, and one ODL. Compared to the SMA-WTCF, the SMA-WCSF has a shorter worst case delay, but more area. It also adds one clock cycle to the latency of the entire dot product, since an additional cycle is needed to produce the final result in two's complement format using the CPA.

3.3. Pipelined SMA with two's complement feedback

The main difference between Design 1 and Design 3 is that a pipeline register is added to the SMA. Since P_1 is fed back to the SMA, it can only be used during the second pipeline stage. Otherwise, each pass through the SMA would require two cycles, instead of one. Figure 5 shows the design of a 5-input pipelined SMA with two's complement feedback (PSMA-WTCF). To simplify the figure, the V-Gens are not shown. In the first pipeline stage, only intermediate values that do not require P_1 are computed. For

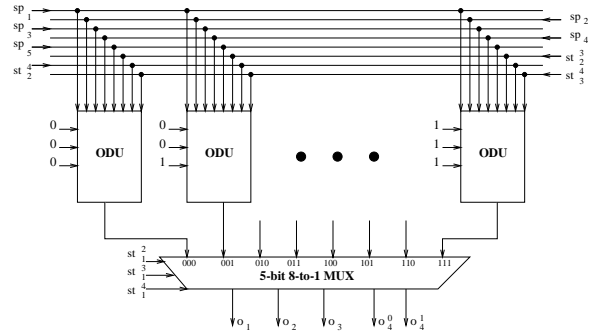


Figure 6. Overflow Detection Circuit for $m = 4$.

example, instead of computing $T_1^3 = P_1 + P_2 + P_3$, the first pipeline stage computes $T_1^{3*} = P_2 + P_3$. In the second stage, intermediate values are combined with P_1 , the overflow detection bits are computed, and the appropriate temporary sum is selected.

Design 3 uses an overflow detection circuit (ODC), which takes the sign bits of the input operands and intermediate additions and produces the overflow detection bits. The ODC calculates these bits in a similar fashion to the previous two designs, but optimizations are made to reduce the worst case delay. Since sign bits st_1^2 to st_1^m arrive later than the other sign bits, the worst case delay is reduced by using overflow detection units (ODUs) to calculate 2^{m-1} versions of the overflow bits and then using an $(m + 1)$ -bit 2^{m-1} -to-1 Mux to select the correct set of overflow bits. An

	Serial	Design 1	Design 2	Design 3	Design 4
Area (gates)	2624	10873	11460	15112	18774
Delay (ns)	28.46	12.08	10.24	8.10	8.68

Table 1. Area and Delay Estimates for 5-input SMAs.

CPA. The PSMA-WCSF requires one more cycle than the PSMA-WTCF to compute a saturated dot product, since an additional cycle is needed to produce the final result in two's complement format.

4. Synthesis Results

Designs for each of the 5-input SMAs presented in this paper were modeled in VHDL and synthesized using LSI Logic's 0.6 micron LCA300K gate array library and the Leonardo synthesis tool from Exemplar. The CPAs and SDCs were implemented using fast carry-lookahead logic. Area and delay estimates from the synthesis tool are shown in Table 1 for a nominal voltage of 3.3 Volts and temperature of 25° C. Of the parallel SMAs, Design 1 has the least area and Design 3 has the least delay. As explained in Section 3.4, Design 4 has a slightly longer delay than Design 3. This is because the ODC component has a longer delay than the CPA (or CSA/SDC pair) and the fan-out of the overflow detection signals is larger for Design 4 than for Design 3. For other implementations, Design 4 is expected to have less delay than Design 3. Compared to the serial SMA, the four parallel SMAs have between 4.14 and 7.14 times more area and between 2.36 and 3.51 times less worst case delay. For many implementations, the long worst case delay of the serial SMA would limit the processor clock rate. Because of the feedback path, the serial SMA cannot be pipelined effectively.

5. Conclusions

This paper covers four design alternatives for parallel SMAs that perform multiple additions with saturation, yet have at most one CPA on the critical delay path. These designs are ideal for loops that perform saturating dot products, such as those found in GSM speech coders. The designs offer various tradeoffs in terms of area, worst case delay, and dot product latency.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. CCR-9703421.

References

- [1] M. Alidina et al. DSP16000: A High Performance, Low-Power Dual-MAC DSP Core for Communications Applications. In *IEEE Custom Integrated Circuits Conference*, pages 119–122, 1998.
- [2] M. M. Alidina and L. R. Tate. Hierarchical Carry-Select, Three-Input Saturation, United States Patent, No. 5,889,689, March 1999.
- [3] J. Du, G. Warner, E. Vallow, and T. Hollenbach. High-Performance DSPs: Using DSP16000 for GSM EFR Speech Coding. *IEEE Signal Processing Magazine*, 17:16–26, March 2000.
- [4] European Telecommunication Standards Institute. Digital Cellular Telecommunications System: ANSI-C Code for the GSM Enhanced Full Rate (EFR) Speech Code (GSM 06.53), March 1997. ETS 300 724.
- [5] J. Eyre and J. Bier. The Evolution of DSP Processors. *IEEE Signal Processing Magazine*, pages 46–51, March 2000.
- [6] J. Fridman and Z. Greenfield. The TigerSHARC DSP Architecture. *IEEE Micro*, 20(1):66–76, 2000.
- [7] R. K. Kolagotla and H. R. Srinivas. Multioperand Addition with Intermediate Saturation, United States Patent, No. 6,182,105-B1, January 2001.
- [8] I. Koren. *Computer Arithmetic Algorithms*. Brookside Court Publishers, 1998.
- [9] T. Lang and J. D. Bruguera. Multilevel Reverse-Carry Computation for Comparison and for Sign and Overflow Detection. In *1999 IEEE International Conference on Computer Design*, pages 73–79, October 1999.
- [10] P. Lapsley. *DSP Processor Fundamentals: Architectures and Features*. IEEE Press, 1997.
- [11] M. K. Prasad, P. D'Arcy, A. Gupta, M. S. Diamondstein, and H. R. Srinivas. Half-Rate GSM Vocoder Implementation on a Dual MAC Digital Signal Processor. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 619–622, 1997.
- [12] Z. RozenShein et al. Star*Core 100 - A Scalable, Compilable, High-Performance Architecture for DSP Applications. In *Proceedings of the International Conference of Signal Processing Applications and Technology*, 1999.
- [13] M. J. Schulte, P. I. Balzola, J. Ruan, and J. Glossner. Parallel Saturating Multioperand Adders. In *Proceedings of the International Conference on Compilers, Architectures and Synthesis for Embedded Systems*, pages 172–179, November 2000.
- [14] N. Yadav, M. Schulte, and J. Glossner. Parallel Saturating Fractional Arithmetic Units. In *9th Great Lakes Symposium on VLSI*, pages 214–217, March 1999.