Proceedings of the 6th Nordic Signal Processing Symposium - NORSIG 2004
June 9 - 11, 2004
Espoo, Finland

IMPLEMENTATION OF BIT-LEVEL PIPELINED DIGIT-SERIAL MULTIPLIERS

Krister Landernäs, Johnny Holmberg

Mälardalen University
Department of Electronics
SE–721 23 Västerås, Sweden
Email: krister.landernas@mdh.se,
johnny.holmberg@mdh.se

Oscar Gustafsson

Linköping University
Department of Electrical Engineering
SE–581 83 Linköping, Sweden
Email:oscarg@isy.liu.se

ABSTRACT

In this paper, a digit-serial multiplier based on shift-accumulation (DSAAM) [1] is compared to a digit-serial/parallel multiplier ($\mathrm{S/P_{pipe}}$) [2]. Both the studied multipliers can be pipelined to an arbitrary degree and are, therefore, well suited for high-throughput implementation. In our study bit-level pipelining was considered. Neither of the multipliers have been implemented in a deep-submicron technology previously, which motivates our study. The multipliers were implemented using a $0.18\mu m$ standard cell technology, and the area, throughput and current consumption was analyzed.

It was concluded that the DSAAM can be implemented with a lower latency than the $\mathrm{S/P_{pipe}}$, leading to a higher throughput and lower area and current consumption. On average the area and current consumption of the DSAAM is 50% and 52% lower than for the $\mathrm{S/P_{pipe}}$, respectively. Furthermore, the throughput of the DSAAM is 37% higher.

1. INTRODUCTION

In digit-serial computation the bits in each word are divided into groups called digits d, and each digit is processed one at a time [3], [4]. Digit-serial computation can, therefore, be considered as a compromise between bit-serial and bit-parallel computation. By choosing the appropriate digit-size the designer can find the best area-time trade-off for the application.

Digit-serial multipliers can be derived by unfolding bit-serial multipliers [3]. However, this leads to a loop in each adder, and, hence, the multipliers cannot be pipelined arbitrarily. Recently, digit-serial multipliers that can be pipelined to the bit-level have been considered [1], [2]. These multipliers can be pipelined to an arbitrary degree and are, therefore, well suited for high-throughput implementations. Neither of the multipliers have been implemented in a deep-submicron technology previously.

In this paper, bit-level pipelining is considered, i.e., at most one full adder between the registers. The arithmetic critical path can be expressed as

$$T_{\mathrm{cp}} = T_{\mathrm{fa}} + T_{\mathrm{D}}, \tag{1}$$

where T_{fa} and T_{D} are the delay of a full adder and the D flip-flop, respectively. Note that for a $0.18\mu m$ process $T_{\mathrm{fa}} \approx T_{\mathrm{D}}$ [5].

A common component in modern digital signal processing systems are frequency-selective digital filters. When implementing recursive digital filters low-latency processing elements are desirable, since increased latency in the feedback loop will decrease the throughput of the algorithm [6], [7]. Thus, a multiplier structure that can be implemented with both with a short T_{cp} and a low latency is a good candidate when implementing recursive digital filters.

In some applications an important implementation constraint is power consumption. This is the case for example in battery powered systems. The main source of power consumption in CMOS circuits is the dynamic power dissipation caused by switching in the circuit. A first order estimate of he dynamic power dissipation can be formulated as

$$P_{\mathrm{dyn}} = \alpha f_{\mathrm{clk}} C_{\mathrm{L}} V_{\mathrm{dd}}^2, \tag{2}$$

where α is the switching activity, C_{L} is the switched capacitance, V_{dd} is the supply voltage and f_{clk} is the clock frequency [8].

The multipliers were implemented with $f_{\mathrm{x}} = 23$ and $f_{\mathrm{y}} = 12$, where f_{x} and f_{y} is the number of fractional bits of the serial data and parallel multiplicand, respectively. Five digit-size are studied ranging from $d = 2$ to $d = 12$. Only positive numbers of the parallel multiplicand is considered. Sign extension is used to handle negative data [9].

2. A PIPELINED DIGIT-SERIAL/PARALLEL MULTIPLIER

In [2], a digit-serial/parallel multiplier that can be pipelined to the bit-level was presented. We will denote this multiplier $\mathrm{S/P_{pipe}}$.

As can be seen in Fig. 1, the multiplier is fed serially while the multiplicand is parallel. Each pipe is a systolic array of full adders. The architecture of the pipe depends on the size of the multiplicand. The output from each pipe is a digit in carry-save representation. The digits are then added using 2-level *carry save adder* (CSA) nets. Finally an adder with carry recursion is used to obtain the correct digit in two's complement form. Naturally this adder must also be pipelined to the bit-level.

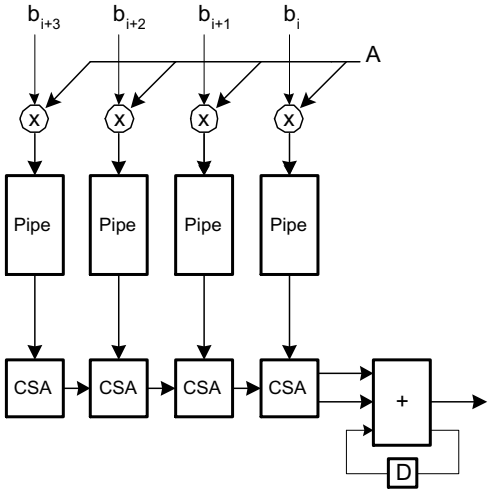

Fig. 1. $\mathrm{S/P}_{\mathrm{pipe}}$ ($d = 4$).

3. DIGIT-SERIAL/PARALLEL MULTIPLIER BASED ON SHIFT-ACCUMULATION

A well known parallel multiplier is the shift-and-add multiplier [9]. In general, the shift-and-add multiplier generates the partial products sequentially and accumulates them successively as they are generated. This approach was used to implement a digit-serial multiplier [1]. We will denote the digit-serial shift-and-add multiplier DSAAM.

The DSAAM basically consists of four parts. A partial product generator, which is implemented using AND-gates. Next, an accumulator adds the partial products which have the same significance level with carry signals from the preceding accumulation. The accumulator is implemented using CSAs. Depending on how the accumulator is implemented a CSA structure containing one or several levels is used to reduce the number of operands. Finally, a carry-propagating adder is needed. The DSAAM structure is shown in Fig. 2.

In this paper, only positive numbers are considered. The DSAAM can, however, easily be modified to handle two's complement numbers. By inverting the partial products containing the sign bit of the coefficient and setting the register

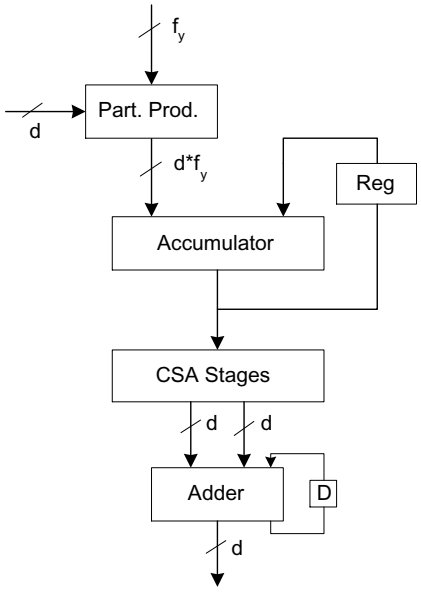

Fig. 2. DSAAM.

corresponding to that significance to one, a two's complement coefficient can be used [1].

The DSAAM will contain two feedback loops: the carry recursion in the accumulator and the the carry loop in the carry propagating adder. The longest logic path of these two loops will determine T_{cp}. In general, a good approach is to try to equalize the logic path in these two loops. Since the accumulator can be implemented with a logic path of one CSA the multiplier can be pipelined to the bit-level, provided that the carry-propagating adder also can be pipelined to the bit-level. A bit-level pipelined digit-serial adder is presented in, e.g., [13].

4. EXPERIMENTAL RESULTS

The multipliers were implemented using UMC $0.18\mu m$ standard cell technology [5]. A circuit layout was obtained using Synopsys Design Compiler [10] and Cadence Silicon Ensemble [11]. Only the core logic was considered, i.e., no pads were included in the designs. Using a SPICE netlist the average current consumption was simulated in Synopsys Nanosim [12].

As mentioned earlier, the multipliers studied are pipelined to the bit-level. This requires a number of pipelining levels to be introduced in the multiplier structure. The number depends on the digit-size and is different in the two cases. This is shown in Table 1.

The main reason that the $\mathrm{S/P}_{\mathrm{pipe}}$ has a longer latency is that the number of CSA stages increases with d. A digit-size of three results in three CSA stages, $d = 4$ will result in four CSA stages, and so on. This leads to an increase in

d	P (DSAAM)	P ($\mathrm{S/P}_{\mathrm{pipe}}$)
2	7	7
3	7	9
4	8	12
6	8	16
12	8	15

Table 1. Number of pipelining levels introduced in the multipliers.

the number of required pipelining levels as d increases. The DSAAM, however, has four CSA stages for all digit-sizes considered. Each stage requires more CSAs when increasing the digit-size, but the total number of stages is always four. This leads to a lower latency for the DSAAM compared to the $\mathrm{S/P}_{\mathrm{pipe}}$.

The final carry propagating adder in both multipliers must be a digit-serial adder that can be pipelined to the bit-level. In [13], a digit-serial hybrid adder was presented. It was shown that the adder can be pipelined to the bit-level with low latency. This adder was utilized when implementing the multipliers in this paper.

The core area of the two multipliers studied is shown in Fig. 3. For small digit-sizes, e.g., $d = 2$ and $d = 3$ the area of the multipliers are comparable. For higher digit-sizes the core area of the DSAAM is much smaller than the corresponding area for the $\mathrm{S/P}_{\mathrm{pipe}}$ multiplier. The reason for this is that the latter requires much more pipelining registers. In fact, for $d = 12$ the area of the combinational logic is about the same in the two cases.

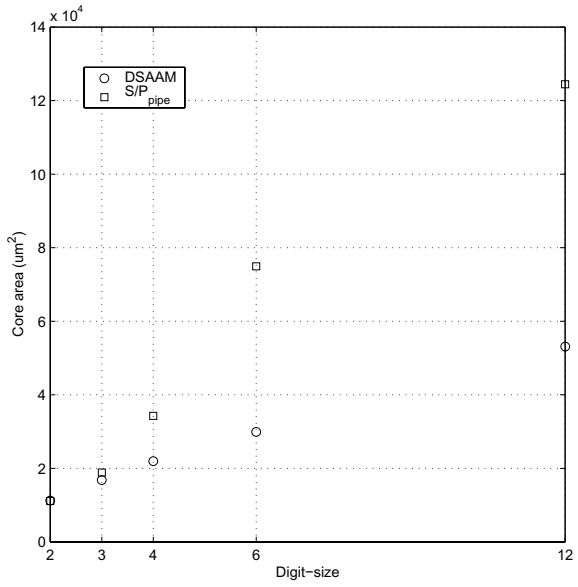

Fig. 3. Core area of the DSAAM and $\mathrm{S/P}_{\mathrm{pipe}}$.

The throughput of the multipliers can be calculated as

$$\frac{1}{T_{\mathrm{s}}} = \frac{d}{[d(P-1) + f_{\mathrm{x}} + f_{\mathrm{y}} + 1]\, T_{\mathrm{cp}}}, \qquad (3)$$

where P is the number of pipelining levels introduced in the multiplier. In order to simplify the control scheme of the multiplier one reset signal for all registers is assumed in (3). The value of P can be found in Table 1.

The throughput of the two multipliers studied is shown in Fig. 4. In the case where $d = 2$ the throughput of the

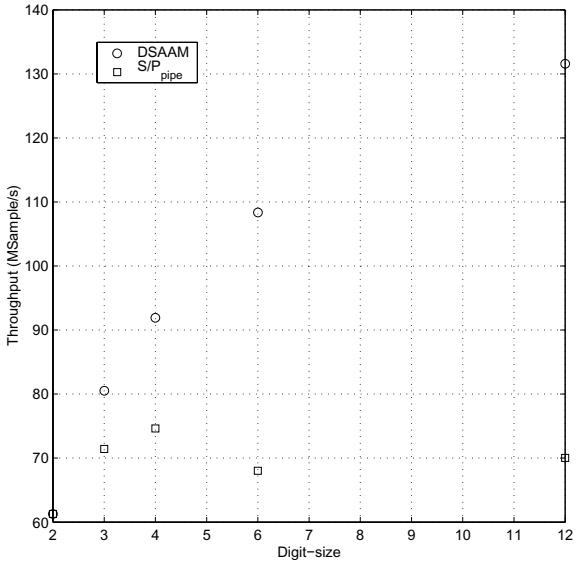

Fig. 4. Throughput of the DSAAM and $\mathrm{S/P}_{\mathrm{pipe}}$.

two multipliers is identical. For higher digit-sizes the long latency of the $\mathrm{S/P}_{\mathrm{pipe}}$ degrades the performance of the multiplier.

The current consumption of the two multipliers studied is shown in Fig. 5. Since the clock frequency is similar for both multipliers, it is fair to assume that the switching frequency is about the same. This would suggest that the number of nodes switching will have an impact on the current consumption (2). It is likely that a correlation between the number of switching nodes and the number of logic gates implemented exist. Comparing the area of the multipliers (Fig. 3) and the current consumption (Fig. 5) it becomes quite clear that similarities can be found, which would support our theory.

Our study show, not only, that low-latency digit-serial multipliers can be implemented with a higher throughput. When implementing digit-serial multipliers requiring a high degree of pipelining, a low latency also leads to considerable less current consumption. On average the area and current consumption of the DSAAM is 50% and 52% lower than for the $\mathrm{S/P}_{\mathrm{pipe}}$, respectively. Furthermore, the through-

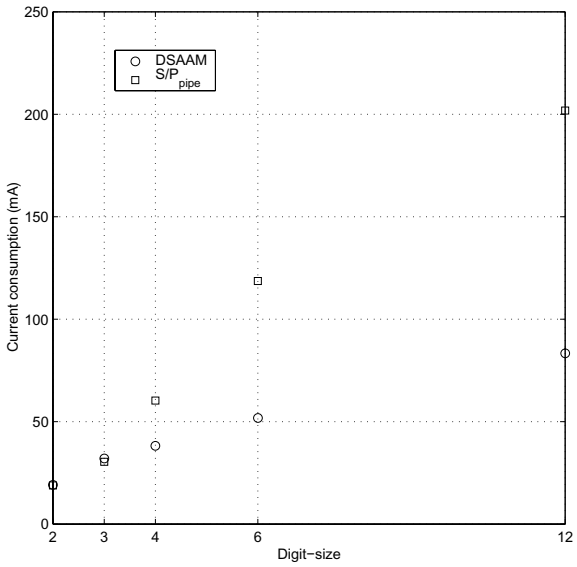

Fig. 5. Current consumption of the DSAAM and $\mathrm{S/P}_{\mathrm{pipe}}$.

put of the former multiplier is 37% higher than the throughput of the latter.

5. CONCLUSIONS

In this paper, a digit-serial multiplier based on shift-accumulation (DSAAM) was compared to a digit-serial/parallel multiplier $\mathrm{S/P}_{\mathrm{pipe}}$. Both the studied multipliers can be pipelined to an arbitrary degree and are, therefore, well suited for high-throughput implementation. In our study bit-level pipelining was considered. The multipliers were implemented using a $0.18\mu m$ standard cell technology, and the area, throughput and current consumption was compared.

It was concluded that the DSAAM can be implemented with a lower latency than the $\mathrm{S/P}_{\mathrm{pipe}}$, leading to a higher throughput and lower area and current consumption. On average the area and current consumption of the DSAAM is 50% and 52% lower than for the $\mathrm{S/P}_{\mathrm{pipe}}$, respectively. Furthermore, the throughput of the former multiplier is 37% higher than the throughput of the latter.

6. REFERENCES

[1] O. Gustafsson and L. Wanhammar, "Bit-level pipelinable general and fixed coefficient digit-serial/parallel multipliers based on shift-accumulation," in *Proc. 9th International Conference on Electronics, Circuits and Systems, 2002*, Vol. 2, pp. 493–496, 15-18 September 2002.

[2] O. Nibouche, A. Bouridane, M. Nibouche, and D. Crookes, "A new pipelined digit serial-parallel multiplier," in *Proc. IEEE Int. Symp. on Circuits and Systems*, Vol. 1, pp. 12–15, Geneva, Switzerland, May 2000.

[3] K. K. Parhi, "A systematic approach for design of digit-serial signal processing architectures," *IEEE Trans. on Circuits and Systems*, Vol. 38, No. 4, pp. 358–375, April 1991.

[4] R. Hartley, and P. Corbett, "Digit-serial processing techniques," *Proc. IEEE Trans. on Circuits and Systems*, Vol. 37, pp. 707–719, June 1990.

[5] High Performance 0.18μ Standard cell library data book, Rev.2.1, Virtual Silicon Technology, Inc., Sunnyvale, CA, January 2001.

[6] M. Renfors and Y. Neuvo, "The maximal sample rate of digital filters under hardware speed constraints," *IEEE Trans. on Circuits and Systems*, CAS-28, No. 3, pp. 196–202, March 1981.

[7] P. Israsena and S. Summerfield, "Bit-level retiming of high-speed digital recursive filters," in *2002 Asia-Pacific Conference on Circuits and Systems, 2002. APCCAS '02*, Vol. 2, pp. 19–24, October 2002.

[8] A. P. Chandrakasan and R. W. Brodersen "Minimizing power consumption in digital CMOS circuits," *Proc. IEEE*, Vol. 83, No. 4, pp. 498–523, 1995.

[9] L. Wanhammar, *DSP Integrated Circuits,* Academic Press, 1998.

[10] Design Compiler Reference Manual, v1999.10, Synopsys 1999.

[11] Envisia Silicon Ensemble Place-and-Route Reference, Product version 5.3, Cadence Design Systems, Inc., November 1999.

[12] Nanosim User Guide, Release 2002.03, Synopsys 2002.

[13] K. Landernäs, J. Holmberg, and M. Vesterbacka, "A high-speed low-latency digit-serial hybrid adder," presented at *IEEE Int. Symp. on Circuits and Systems*, ISCAS 2004.