A Low Power Multiplication and dividing Technique using Radix

Raj Kumar Agarwal[1] Dheeraj Jain[2]
[1]Research Scholar [2]Head of Department
[2]Department of Electronics & Communication engineering
[2]Institute of Technology and management, Bhiliwara

Abstract— The general objective of this paper is to develop methods to reduce the power consumption of arithmetic modules, while maintaining the delay unchanged and keeping the increase in the area to a minimum. Here we illustrate some techniques for a radix-4 divider realized in 0:6_m CMOS technology and compare it with radix-8. We obtained a power consumption reduction of 35% with respect to the standard implementation. The techniques used here should be applicable to a variety of arithmetic modules which have similar characteristics.

Key words: Radix-4, Radix-8.

I. INTRODUCTION

The general objective of our work is to develop methods to reduce the power consumption of arithmetic modules. We attempt to reduce the power while maintaining the delay unchanged and keeping the increase in the area to a minimum. Since the dynamic power dissipation in CMOS cells is proportional to the switching frequency and to the output load [1], we reduce the number of transitions and the capacitance (by using lower-drive cells when available).

The implementations of the radix-4 divider were done using the COMPASS design environment [2] and the Passport 0:6_m standard cell library [3]. The structural model was obtained by manually decomposing the behavioral model into functional blocks. Some of those blocks were synthesized by COMPASS from behavioral models to gate networks, others were manually implemented with gates. The power estimation has been carried out with PET [4], a power evaluation tool which computes the power dissipated in a circuit from the netlist extracted from the layout, the standard cell library characteristics, and the results of a logic-level simulation run on a suitable number of test vectors.

Using techniques such as switching-o_ not active blocks, retiming the recurrence, equalizing the paths to reduce glitches, using gates with lower drive capability, and changing the redundant representation, we obtained a power consumption reduction of 35% with respect to the standard implementation.

II. RADIX-4 IMPLEMENTATION

The low-power implementation of a radix-4 divider is directly derived from that presented in [4]. On account of that implementation a low-power convert and round unit as described in [5] is added. Furthermore the circuit was redesigned with a new library (same feature size of 0:6_m, but 3 metal layers) and the results are slightly different.

The energy-per-division in the standard implementation, optimized for minimum latency, is 45.4 nJ. The critical path is 8.3 ns, allowing a clock frequency of 120 MHz. The time to perform the division is tR4 =250 ns. The energy-per-division dissipated in the convert-and- round unit is reduced from 12 nJ to 3.7 nJ, and the overall energy-per-division is 27 nJ, while the area is 1.2 mm2. It was not possible for us to implement dual voltage because our library does not provide low-voltage cells. We roughly estimate that the energy-per-division of an implementation with dual-voltage is 14.3 nJ. The power reduction with respect to the basic divider is about 70%

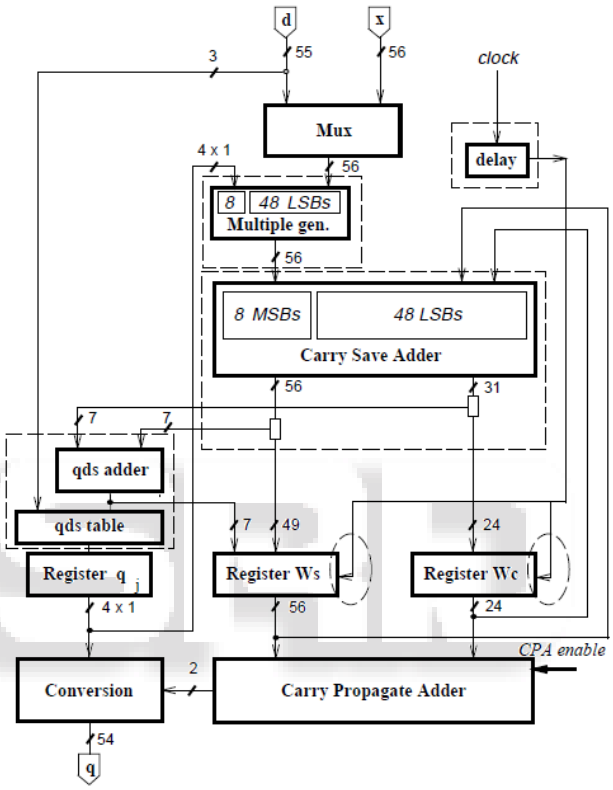

Fig. 1: Implemenattion of radix-4

III. RADIX-8 IMPLEMENTATION

For the radix-8 divider the quotient digit set is in [-7; 7].
In this the quotient digit is split into two parts

Q_h with weight 4 and Q_l with weight 1 (see [3]) and the digit set of each part is reduced to {-2;- 1; 0; 1; 2} in order to avoid the implementation of a complicated multiple generator.

The standard implementation, shown in Figure 1, has a critical path of 10.7 ns corresponding to a maximum clock frequency of 93 MHz. The time to perform the operation is T_{R8} =214 ns, and its energy-per-division is 47.7 nJ.

The low-power implementation, described in detail in [6], is obtained by retiming the recurrence, changing to radix-8 the LSBs in the carry-save adder, and by disabling the SZD unit during the recurrence steps (Figure 2). By implementing the modified convert-and-round algorithm, we reduce the number of flip-flops in the convert-and-round unit from 171 to 81.

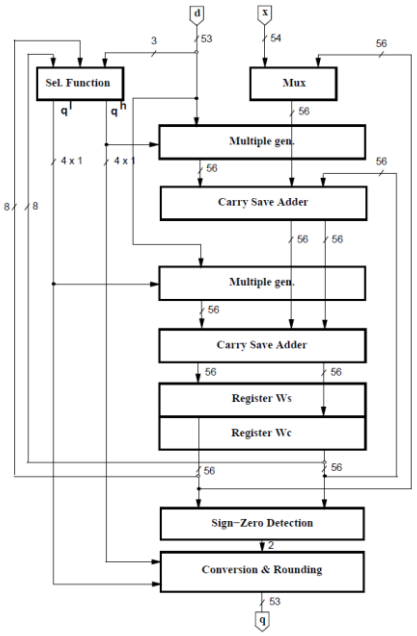

Fig. 2: Implementation of radix-8

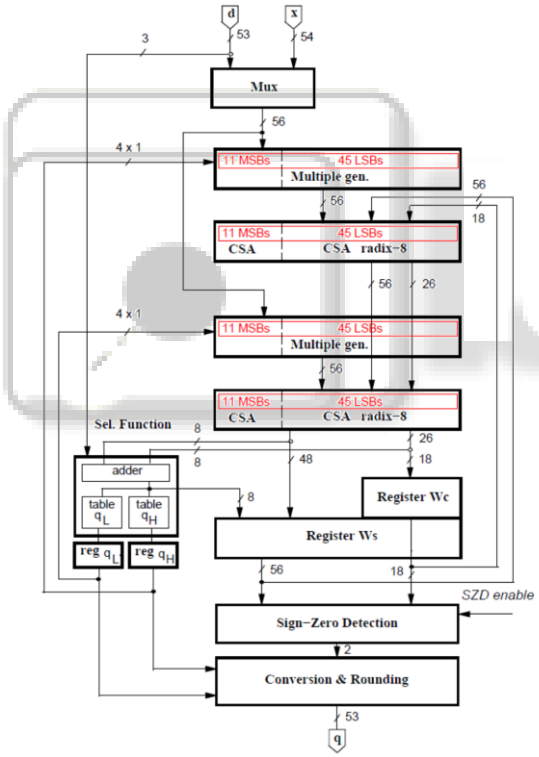

Fig. 3: Low power implementation of radix-8

We estimated that by implementing the unit with dual voltage the energy-per-division becomes 14.0 nJ, corresponding to a reduction of 70%.

IV. Power Reduction and Implementations

In order to obtain a reduction in the power dissipation (i.e. in the energy-per-division) various design techniques and modifications of the algorithm are applied to the standard implementation of the divider .The modifications are done without deteriorating the timing of the circuit. The various techniques have already been presented in [4]:

- Switching-off not active blocks which are not active during several cycles.
- Retiming the recurrence for reducing the number of spurious transitions and limit the critical path to a few bits in the recurrence allowing the rest of the bits to be redesigned for low-power (e.g. using slower cells).
- Altering the redundant representation to minimize the number of flip-flops.

According to [4], the path-equalization technique has been abandoned because , on account of having the automatic floorplanning of the layout, the interconnection delay can't be controlled and very small improvement was obtained

The following modifications are described in [5]:

- Utilization of lower voltage for V_{dd} in those cells not in the Critical path.
- To eliminate shift-registers and reduce the number of flip-flops, the on-the-fly convert-and-round algorithm is modified

Moreover, since the quotient-digit selection is a function of some bits of the divisor having fixed for the whole division operation, it is convenient to decompose the function into sub functions and to enable only the sub function corresponding to the actual value of the divisor. This is especially convenient for higher radices, because the quotient-digit selection is more complex and therefore is responsible for a significant portion of the energy.

V. Result of Comparison between the Radix-4 and the Radix-8 Divider

Table 1 summarizes the characteristics of the two dividers.

	Radix-4	**Radix-8**
0-delay transitions	9.1	8.8
actual transitions	13.1	11.2
no. of transistors	20.1	21.1
Area (mm2)	1.51	1.49

Table 1: Transitions and Area

The energy-per-division is divided into the contribution of the recurrence and that of the conversion and rounding. According to table for both the radix-4 and the radix- 8 dividers, there is a 40% reduction in the energy-per-division in the low-power implementation. Dual voltage implementation will show a reduction of about 70% for both radices. The speed for the radix-8 over the radix-4 is about 17%, while the increase in the energy-per-division is less than 2% in the low-power implementation. In the dual voltage implementation, our estimate indicates that the energy-per-division in the radix-8 is even smaller than in the radix-4. However, the radix-8 has a larger energy-per-cycle, 1.38 nJ, compared to the radix-4, 0.9 nJ.

Furthermore, we notice that in the low-power implementation the area is reduced. This is due to the reduction in the number of flip-flops both in the recurrence (change of redundant representation) and in the conversion-and-round unit (elimination of some flip-flops). The increase in area from radix-4 to radix-8 (about 50%) does

not reflect on the energy dissipated to complete an operation, that is almost the same.

The radix-4 divider is smaller, but it is slower and consumes almost the same amount of energy. It is up to the designer to decide which of the two implementation to use according to the constraints of the design.

REFERENCES

[1] N. H. E. Weste and K. Eshraghian. Principles of CMOS VLSI Design. Addison Wesley Publishing Company, 2nd edition, 1993.

[2] Compass Design Automation. User Manuals for COMPASS VLSI. Compass Design Automation, Inc., 1992.

[3] Compass Design Automation. Passport - 0.6- Micron, 3-Volt, High-Performance Standard Cell Library. Compass Design Automation, Inc., 1994.

[4] A. Nannarelli. Power evaluation tool: modeling and implementation. COMPASS User's Group Conference, Apr 1996.

[5] M.D. Ergegovac and T. Lang. Division and Square Root: Digit-Recurrence Algorithms and Implementations. Kluwer Academic Publisher, 1994.

[6] Ian T. Foster, Designing and Building Parallel Programs, Addison-Wesley,1995.

[7] Michael J. Flynn and Stuart F. Oberman, Advanced Computer Arithmetic Design, Wiley-Interscience, New York, New York, 2001.

[8] Alan Daly, William Marnane, Tim Kerins, and Emanuel Popovici, Fast Modular Division for Application in ECC on Reconfigurable Logic, Proc. of the 13th International Conference on Field-Programmable Logic and Applications (2003).

[9] A. De Vora, M. Ley, E. Ofner, and H. Gr¨unbacher, A High-Speed Radix 4 Hardware Divider For ASIC's, Tagungsband Mikroelektronik (2003).

[10] David L. Harris, Stuart F. Oberman, and Mark A. Horowitz, SRT Division Architectures and Implementations, IEEE Symposium on Computer Arithmetic (1997).

[11] David W. Matula and Alex Fit-Florea, Prescaled Integer Division, IEEE Symposium on Computer Arithmetic (2003).

[12] James E. Stine and Michael J. Schulte, A Combined Interval and Floating- Point Divider, Proceedings of the 32nd Asilomar Conference on Signals, Systems, and Computers (1998).

[13] H. P. Sharangpani and M. L. Barton, Statistical Analysis of Floating Point Flaw in the Pentium Processor, Technical Report (1994).

[14] Robert Uang, Christian Bourd´e, and Tim Bagwell, A SiGe 23 GHz Fractional-N Frequency Synthesizer, Technical Report (2004).

[15] Roman Lysecky and Frank Vahid, A Study of the Speedups and Competitiveness of FPGA Soft Processor Cores using Dynamic Hardware/ Software Partitioning, Proceedings of the Design, Automation and Test, Volume 1 (2005).

