# POWER-DELAY TRADEOFFS FOR RADIX-4 AND RADIX-8 DIVIDERS

Alberto Nannarelli and Tomas Lang Department of Electrical & Computer Engineering University of California, Irvine, California 92697 alberto@ece.uci.edu, tomas@ece.uci.edu

# Abstract

The use of higher radices in division reduces the number of iterations to complete the operation, but increases the complexity of the circuit. In this paper we explore the influence of the radix on the power dissipation of a floating-point divider and the power-delay tradeoffs. We compare the performance and the energy consumption per operation for a radix-4 and a radix-8 divider, realized in CMOS technology. A reduction of about 40% in the energy consumption is obtained for both radices (about 70% if low-voltage gates, for dual voltage implementation, are available). Also the results show that the radix-8 divider is about 20% faster and the energy dissipated to perform a division is about the same, with respect to the radix-4.

## 1 Introduction

Tradeoffs between area and delay have always been something designers of ICs had to deal with. Recently, with the advent of portable electronics and the increased density on chip, also power dissipation started to play an important role in the design process and power consumption constraints cannot be neglected any longer. Our work is focused on the evaluation of different schemes in the design of low-power floating-point dividers and their power-delay tradeoffs. The division algorithm implemented is the modified SRT for radices greater than 2 [3]. It is known that the use of higher radices reduces the number of cycles required to complete the operation, but increases the complexity of the circuit. In this paper we study the influence of the radix on the power dissipation. More specifically, we compare the performance and the energy consumption per operation for a radix-4 and a radix-8 divider.

A number of well-known techniques for the reduction of power [7, 8, 2] are used along with some techniques specific to the division algorithm [5].

The low-power implementation of the radix-4 divider is directly derived from that presented in [4]. With respect to the techniques presented there we have added a lowpower convert-and-round unit, considered the use of dual voltage, and partitioned and disabled the quotient-digit selection function.

The units are implemented using the Passport  $0.6\mu m$  CMOS standard cell library [1]. This library does not provide low-drive (or low-power) cells for all the logic gates, and this limits the design choices. Furthermore, because of the use of automatic floorplanning, we loose the control on the placement of cells, and this reflects on variations in the interconnection capacitance, among different layouts, that sometimes hide the benefits of the technique applied (e.g path equalization).

Results show that is possible to get a reduction of about 40% in the energy-per-division for both radices (about 70% if low-voltage gates, for dual voltage implementation, are available). Also the radix-8 divider is about 20% faster and the energy dissipated to perform a division is about the same, with respect to the radix-4.

#### 2 Algorithm and Metrics

The division algorithm, described in detail in [3], is implemented by the residual recurrence

$$w[j+1] = rw[j] - q_{j+1}d$$
  $j = 0, 1, ..., m$ 

with initial value w[0] = x, where r is the radix, x the dividend, d the divisor, and  $q_{j+1}$  the quotient digit at the *j*-th iteration. The quotient is  $q = \sum_{j=1}^{m} q_j r^{-j}$  where m is the number of iterations needed to produce the n bits of the result (53 for IEEE double-precision).

The quotient digit is in signed-digit representation and is determined, at each iteration, by the selection function

$$u_{j+1} = SEL(d_{\delta}, (rw[j])_t)$$

where the divisor is truncated after  $\delta$  fractional bits and rw[j] after t fractional bits. The residual w[j] is stored in carry-save representation ( $w_S$  and  $w_C$ ).

The radix-4 divider requires 30 clock cycles to produce the quotient: 28 iterations to produce the necessary quotient digits plus additional cycles for the initialization of the operands and the final rounding. Similarly, the number of cycles required for the radix-8 divider is 20.

The performance metric is the time elapsed per operation which is

$$t_{div} = T_{cycle} \times (no. of cycles)$$

In order to have a measure of the power dissipated independent of the frequency, we measure the energy-perdivision, that is computed as  $E_{div} = \int_{t_{div}} vi \ dt = \sum_{i=1}^{N} E_i$  where N is the number of cells in the circuit and  $E_i$  is the energy dissipated in the *i*th-cell during  $t_{div}$ .

#### 3 Power Reduction and Implementations

Design techniques and modifications of the algorithm are applied to the standard implementation of the divider to obtain a reduction in the power dissipation (i.e. in the energy-perdivision). The modifications are done with the constraint of not deteriorating the timing of the circuit. The following techniques have already been presented in [4]:

- Switching-off blocks which are not active during several cycles.
- Retiming the recurrence to reduce 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).
- Changing the redundant representation to reduce the number of flip-flops.

With respect to [4], the path-equalization technique has been abandoned because, due to the automatic floorplanning of the layout, we could not control the interconnection delay and the improvement obtained was very small. The following modifications are described in [5]:

- Use of a lower voltage for  $V_{DD}$  in those cells not in the critical path.
- The on-the-fly convert-and-round algorithm is modified to eliminate shift-registers and reduce the number of flip-flops.

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

## 3.1 Radix-4 Implementation

The low-power implementation of a radix-4 divider is directly derived from that presented in [4]. With respect to that implementation we added here a low-power convertand-round unit as described in [5]. Furthermore the circuit was re-laid-out with a new library (same feature size of  $0.6\mu m$ , 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  $t_{R4} = 250 ns$ .

The energy-per-division dissipated in the convert-andround 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  $mm^2$ .

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%.

## 3.2 Radix-8 Implementation

For the radix-8 divider the quotient digit set is in [-7, 7]. In this case, to avoid the implementation of a complicated multiple generator, 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  $\{-2, -1, 0, 1, 2\}$ .

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.

Table 1 shows the energy-per-division dissipated in the blocks. Entry "std" refers to the standard implementation, optimized for speed, entry "l-p" is the low-power implementation, and "d-v" is the estimate of a possible dual-voltage implementation. Values marked \* include level shifters.

blocks	$\operatorname{st} d$	l-p	d-v
control	0.6	0.6	0.6
clk tree	0.4	0.4	0.4
mux	1.4	0.2	0.1
mul. gen. H	3.1	1.8	0.7
CSA H	4.4	4.2	1.5
mul. gen. L	2.6	1.7	0.6
CSA L	6.0	5.3	1.9
sel. func.	3.6	4.0	4.0
register wc	4.2	1.2	0.4
register ws	4.2	4.0	*1.4
register qL	-	0.2	0.2
register qH	-	0.2	0.2
total recur. $[nJ]$	30.5	23.8	12.0
SZD	3.8	1.0	1.0
C&R unit	13.4	2.8	$^{*}1.0$
total C&R $[nJ]$	17.2	3.8	2.0
Total divider $[nJ]$	47.7	27.6	14.0
Ratio	1.00	0.59	0.30
Area $[mm^2]$	2.2	1.8	-

Table 1: Energy-per-division for radix-8

The energy-per-division in the low-power implementation of the radix-8 divider is 27.6 nJ, and the area of the unit is 1.8  $mm^2$ .

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

#### 4 Comparison between the Radix-4 and the Radix-8 Divider

Table 2 summarizes the characteristics of the two dividers. The energy-per-division is split into the contribution of the recurrence and that of the conversion and rounding.

The table shows, for both the radix-4 and the radix-8 dividers, a reduction in the energy-per-division of about 40% in the low-power implementation. An implementation with dual voltage will show a reduction of about 70% for both radices. The speed-up 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,



Figure 1: Standard implementation of radix-8 division

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.

	radix-4			radix-8		
	$\operatorname{std}$	l-p	d-v	$\operatorname{st} d$	l-p	d-v
$E_{div}$ rec. $[nJ]$	27.7	21.6	11.4	30.5	23.8	12.0
$E_{div}$ conv. $[nJ]$	17.7	5.4	2.9	17.2	3.8	2.0
$E_{div}$ total $[nJ]$	45.4	27.0	14.3	47.7	27.6	14.0
Ratio	1.00	0.59	0.31	1.00	0.59	0.30
$T_{cycle}$ [ns]	8.3	8.3	8.3	10.7	10.7	10.7
$t_{div}$ [ns]	250	250	250	214	214	214
Area $[mm^2]$	1.4	1.2	-	2.2	1.8	-

Table 2: Summary

# Acknowledgment

This work was partially supported by NSF grant MIP 9314172 and by State of California and Sun Microsystems, through UC MICRO 97-084.



Figure 2: Low-power implementation of radix-8 divider

#### References

- Compass Design Automation. Passport 0.6-Micron, 3-Volt, High-Performance Standard Cell Library. Compass Design Automation, Inc., 1994.
- [2] A. P. Chandrakasan and R. W. Brodersen. Low Power Digital CMOS Design. Kluwer Academic Publishers, 1995.
- [3] M.D. Ergegovac and T. Lang. Division and Square Root: Digit-Recurrence Algorithms and Implementations. Kluwer Academic Publisher, 1994.
- [4] A. Nannarelli and T. Lang. Low-power radix-4 divider. Proc. of International Symposium on Low Power Electronics and Design, pages 205-208, Aug. 1996.
- [5] A. Nannarelli and T. Lang. Low-Power Divider. Technical Report, Sep. 1997. Available at http://www.eng.uci.edu/numlab/archive/pub/nl97p03/.
- [6] A. Nannarelli and T. Lang. Low-Power Radix-8 Divider. Technical Report, Mar. 1998. Available at http://www.eng.uci.edu/numlab/archive/pub/nl98p02/.
- [7] W. Nebel and J. Mermet editors. Low Power Design in Deep Submicron Electronics. Kluwer Academic Publishers, 1997.
- [8] J. M. Rabaey, M. Pedram, et al. Low Power Design Methodologies. Kluwer Academic Publishers, 1996.