INTEGRATION Letters

AT^2-optimal VLSI integer division and integer square rooting

Kurt Mehlhorn *

Fachbereich Angewandte Mathematik und Informatik, Universität des Saarlandes, 6600 Saarbrücken, Fed. Rep. Germany

Received 3 February 1984

Abstract. In this paper we exhibit AT^2-optimal integer division circuits for n-bit integers for all computation times T in the range $[\Omega((\log n)^2), O(\sqrt{n})]$ and AT^2-optimal square rooting circuits for all computation times T in the range $[\Omega((\log n)^3), O(\sqrt{n})]$.

Keywords. VLSI computation, optimal networks, integer division, square rooting.

1. Introduction

Research on efficient schemes for the basic arithmetic operations on integers, potentially suitable for direct VLSI circuit implementation, has been going on for some years. Significant progress was made and resulted in optimal circuits for integer addition and subtraction and multiplication. Optimality is defined with respect to the customary AT^2 measure of complexity, which is central to the synchronous VLSI model of computation [8,3]. Here A is the area of the chip, while T is the computation time. As is well known [1,3], any multiplication circuit for two n-bit integers (with a $2n$-bit result) must satisfy $AT^2 = \Omega(n^2)$, $A = \Omega(n)$, $T = \Omega(\log n)$. For integer multiplication the question of optimality was settled in [7,5]. They exhibited a class of designs which achieve $AT^2 = O(n^2)$ for all computation times in the range $[\Omega(\log n), O(\sqrt{n})]$. More precisely, the range

* This work was done while the author was visiting the Coordinated Science Laboratory at the University of Illinois. It was supported by the National Science Foundation under Grant MCS-81-05552 and by the Deutsche Forschungsgemeinschaft SFB 124, VLSI-Entwurf und Parallelität.

North-Holland
INTEGRATION, the VLSI journal 2 (1984) 163–167

$[\Omega((\log n)^2), O(\sqrt{n})]$ is covered in [7] and the range $[\Omega(\log n), O(n^{2/5})]$ is covered in [5].

In this letter we deal with integer division and integer square rooting. We will exhibit AT^2-optimal circuits for both problems. More specifically, we show the following theorems.

Theorem 1. *There is an AT^2-optimal, i.e., $AT^2 = O(n^2)$, integer division circuit for all T in the range $[\Omega((\log n)^2), O(\sqrt{n})]$.*

Theorem 2. *There is an AT^2-optimal, i.e., $AT^2 = O(n^2)$, integer square rooting circuit for all T in the range $[\Omega((\log n)^3, O(\sqrt{n})]$.*

The circuits which we propose are VLSI implementations of well-known algorithms based on Newton iteration. The implementations are not straightforward, however. The division algorithm is described in [6, Section 4.3.3] and the square rooting algorithm is described in [2]. With respect to the division problem it is easy to see that we can restrict our consideration to the Reciprocal Problem.

Reciprocal Problem. *Let v have the binary representation $v = (0.v_1v_2\ldots v_n)$ where $v_1 = 1$. Compute $z = xx.x\ldots x$ with n places after the radix point such that $|z - 1/v| \leqslant 2^{-n}$.*

The Square Rooting Problem is defined as follows.

Square Rooting Problem. *Let v have the binary representation $v = v(v_1v_2\ldots v_n)$. Compute $z = xx\ldots x.x\ldots x$ with $\frac{1}{2}n$ digits to the right of the radix point such that $|v - z^2| \leqslant 1$.*

The optimal division circuit is discussed in Section 2, and the optimal square rooting circuit is discussed in Section 3. We close this section with a short remark about lower bounds. For integer division the lower bound $AT^2 = \Omega(n^2)$ follows from the fact that x/y is equivalent to a shift if y is a power of two. Thus integer division is at least as difficult as shifting and the lower bound is established. For square rooting we have to work slightly harder. Suppose that we have a VLSI chip for computing square roots with area A and computation time T. Then we can clearly build a chip with the same performance which takes two numbers v and z (where v is an integer of length n and z has $\frac{1}{2}n$ binary digits after the radix point) and checks whether $|\sqrt{v} - z| < 2^{n/2}$. For the latter problem, Lipton and Sedgewick [4] have shown that $AT^2 = \Omega(n^2)$.

2. Efficient reciprocal computation in VLSI

We start by reviewing the well-known algorithm for reciprocal computation based on Newton iteration. As before let v have the binary representation $v = (0.v_1v_2\ldots v_n)$ with $v_1 = 1$.

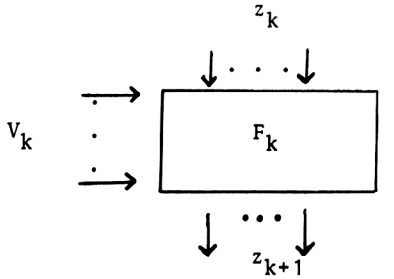
Algorithm

$z_0 \leftarrow \frac{1}{4}\lfloor 32/(4v_1 + 2v_2 + v_3)\rfloor$; $k \leftarrow 0$;

repeat comment at this point $z_k \leqslant 2$, $|z_k - 1/v| \leqslant 2^{-2^k}$
and z_k has a binary representation of the form $xx.xx\ldots x$ with $2^k + 1$ places after the radix point;
let $V_k = (0.v_1v_2\ldots v_{2^{k+1}+3})$;
$z_{k+1} \leftarrow 2z_k - V_k z_k^2 + r$ where $0 \leqslant r < 2^{-2^{k+1}-1}$ is chosen such that z_{k+1} is a multiple of $2^{-2^{k+1}} - 1$
$k \leftarrow k + 1$
until $2^k \geqslant n$ **od**

The proof of correctness can be found in [6, Section 4.3.3D]. It is important to observe that the computations in the kth iteration of the loop is done on binary numbers of $O(2^k)$ bits. The arithmetic on these numbers must be done exactly.

For the sequel we assume w.l.o.g. that $n = 2^m$ is a power of two. It follows from the results of Preparata and Vuillemin [7] and Mehlhorn and Preparata [5] that there is a VLSI circuit F_k given as follows:

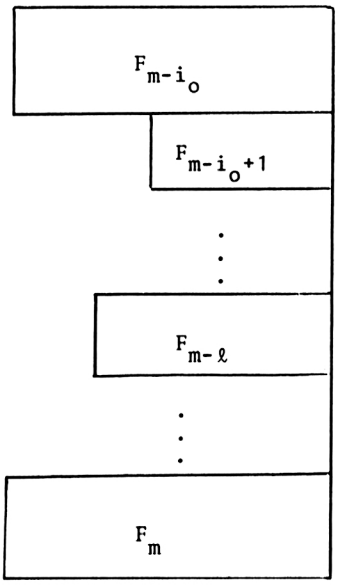


(a) F_k computes $z_{k+1} = z_k - V_k z_k^2 + r$ as given above; F_k takes input z_k at the top and input V_k at the left and produces z_{k+1} at the bottom.

(b) F_k operates in time t and has area $c(2^k/t)^2$ for some constant c. Computation time t can be chosen arbitrarily in the range $ak \leqslant t \leqslant b2^{k/2}$ for some constants a and b, $a \geqslant 1$.

We can now describe our VLSI circuit for reciprocal computation. Let T be chosen arbitrarily in the range $a(\log n)^2 \leqslant T \leqslant e\sqrt{n}$, where $e = \min\{a, b\}$. The circuit to be described operates in time $T_r = O(T)$ and has area $A_r = O(n^2/T^2)$. Thus $A_r T_r^2 = O(n^2)$.

Let $i_0 = \log((T/a)\log n)$. Then $\log\log n \leqslant i_0 \leqslant \frac{1}{2}\log n$. We realize the first $m - i_0$ iterations of the loop on a single (fast) circuit F_{m-i_0} of delay $a \log n$ which we feed back to itself. Later iteration steps have own dedicated circuits. More precisely, for iteration step $m - l$, $i_0 - 1 \geqslant l \geqslant 0$, we use circuit F_{m-l} of delay $T/2^{l/2}$. Thus total delay is $O((\log n)^2 + T) = O(T)$ since $T = \Omega((\log n)^2)$.



used for first $m - i_0$ iterations, delay $a \log n$

used for $(m - l)$th iteration, delay $T/2^{l/2}$

It remains to compute the total area of all circuits. We leave it to the reader to convince himself that the space needed for routing inputs and outputs and for routing intermediate results increases the area by at most a onstant factor.

Claim 1. *F_{m-i_0} has area* $O((n/T)^2)$.

Proof. Since $a(m - i_0) \leqslant a \log n \leqslant b2^{(m-i_0)/2}$ for sufficiently large n, we conclude that F_{m-i_0} has area at most $c(2^{m-i_0}/a \log n)^2 = c(n/T)^2$. Thus F_{m-i_0} has area $O(n^2/T^2)$. □

Claim 2. *F_{m-l}, $0 \leqslant l < i_0$, has area* $O((n^2/T^2)2^{-l})$.

Proof. Note first that $a(m - l) \leqslant T/2^{l/2} < b2^{(m-l)/2}$ for $0 \leqslant 1 < i_0$ by choice of i_0. Thus circuit F_{m-l} with delay $T/2^{l/2}$ exists and has area at most $c(2^{m-l}/(T/2^{l/2}))^2 = c(n/T)^2 2^{-l}$. □

Claims 1 and 2 together imply that our circuit has area $O(n^2/T^2)$. This completes the proof of Theorem 1.

3. Efficient square rooting

Square roots can also be computed by Newton iteration; the iteration rule is given by

$$z_{k+1} \leftarrow \tfrac{1}{2}(V_k/z_k + z_k),$$

where again z_k and V_k consist of $O(2^k)$ bits. Let F_k be a circuit which takes inputs

z_k and V_k and produces output z_{k+1}. It follows from Theorem 1 that we make circuit F_k to work in time t and area $c(2^k/t)^2$ for every t in the range $ak^2 \leqslant t \leqslant b2^{k/2}$ for some constants a, b and c. We now proceed in complete analogy to Section 2, except that we choose $i_0 = \log(T/(\log n)^2)$ in order to reflect the fact that the fastest known optimal division chip runs in time $\Theta((\log n)^2)$ instead of $\Theta(\log n)$ as for multiplication. Then total delay is $O(T + (\log n)^3) = O(T)$ since T is assumed to be at least $\Omega((\log n)^3)$. Also the area of the circuit is $O((n/T)^2)$ by the arguments above. This proves Theorem 2.

Acknowledgment

I want to thank G. Bilardy for suggesting that the construction also yields an optimal square rooting circuit and I want to thank F. Preparata for our many discussions about division.

References

[1] Abelson, H. and P. Andreae, Information transfer and area-time trade-offs for VLSI multiplication, Comm. ACM **23** (1) (1980) 20–22.

[2] Alt, H., Square rooting is as difficult as multiplication, *Computing* **21** (1979) 221–232.

[3] Brent, R.P. and H.T. Kung, The chip complexity of binary arithmetic, *J. Assoc. Comput. Mach.* **28** (28) (1981) 521–534.

[4] Lipton, R.I. and R. Sedgewick, Lower bounds for VLSI, *ACM Symp. on Theory of Computing 1981*, pp. 300–307.

[5] Mehlhorn, K. and F.P. Preparata, Area-time optimal VLSI integer multiplier with minimum computation time, Tech. Rep., Coord. Science Lab., Univ. of Illinois, 1983.

[6] Knuth, D.E., *The Art of Computer Programming, Vol. 2* (Addison-Wesley, Reading, MA, 1969).

[7] Preparata, F.P. and J. Vuillemin, Area-time optimal VLSI networks for computing integer multiplication and Discrete Fourier Transform, *Proc. of ICALP*, Haifa, Isreal (1981) pp. 29–40.

[8] Thompson, C.D., Area-time complexity of VLSI, *Proc. 11th Ann. ACM Symp. on the Theory of Computing (SIGACT)* (1979) pp. 81–88.