Optimal-Depth Threshold Circuits for Multiplication and Related Problems

Chi-Hsiang Yeh
Dept. of Electrical & Computer Engineering
Queen's University
Kingston, Ontario, Canada, K7K 3N6

E.A. Varvarigos, B. Parhami, and H. Lee
Dept. of Electrical & Computer Engineering
University of California
Santa Barbara, CA 93106-9560, USA

Abstract

Multiplication is one of the most fundamental operations in arithmetic and algebraic computations. In this paper, we present depth-optimal circuits for performing multiplication, multioperand addition, and symmetric function evaluation with small size and restricted fan-in. In particular, we show that the product of two n-bit numbers can be computed using a unit-weight threshold circuit of fan-in k, depth $3\log_k n + \frac{\log_2 d}{\log_2(1+\sqrt{5})-1} + o(\log_k n + \log d) + O(1)$, and edge complexity $O(n^{2+1/d}\log(d+1))$, for any integer $d > 0$. All the circuits proposed in this paper have constant depth when $\log_k n$ is a constant and are depth-optimal within small constant factors for any fan-in k.

1 Introduction

Threshold circuits constitute a powerful computational model for arithmetic and other computations [12, 13, 16, 20]. A *linear threshold function* is defined as a Boolean function

$$\mathrm{Sgn}(F(X)) = \begin{cases} 1 & \text{if } F(X) \geq 0; \\ 0 & \text{if } F(X) < 0, \end{cases}$$

where $X = (x_1, \ldots, x_k) \in \{0,1\}^k$ is the vector of input variables, and $F(X) = \sum_{i=1}^{k} w_i x_i + w_0$. The scalars $w_i, i = 1, 2, \ldots, k$, are called the *weights*, and w_0 is called the *bias* of the threshold function. A *threshold circuit* is defined as a computational network that consists of an acyclic interconnection of threshold gates, each of which computes a linear threshold function [10, 13]. The *depth* of a circuit is defined as the length of (i.e., the number of nodes on) the longest path from any input to any output node of the circuit, while the *fan-in* of a circuit is defined as the largest fan-in among all the gates contained in it. The *edge complexity* and the *gate complexity* of a circuit are defined as the number of edges and the number of gates in the circuit, respectively.

In this paper, we propose unit-weight threshold circuits to perform iterated addition and multiplication and to evaluate general symmetric functions. The circuits we propose have depths that are considerably smaller than those in [3, 13, 14]. In particular, we present a unit-weight threshold circuit to compute the sum of m n-bit numbers, which has depth approximately equal to $2\log_k m + \log_k n + 1.44\log_2 d$, edge complexity $O(nm^{1+1/d}\log(d+1))$, and fan-in k, for any positive integer d. The parameter d can be selected to obtain the desired tradeoff between depth and edge complexity of the circuit. The depth of our iterated addition circuit is optimal within a factor of $1 + o(1)$ when $\log_2 d = o(\log_k n)$ and $\log_k m = o(\log_k n)$, and is optimal within a factor of $1.5 + o(1)$ when $\log_2 d = o(\log_k n)$ and $\log_k m \approx \log_k n$.

We derive a unit-weight threshold circuit to compute the product of two n-bit integers, which has depth approximately equal to $3\log_k n + 1.44\log_2 d$, edge complexity $O(n^{2+1/d}\log(d+1))$, and fan-in k. The depth of this multiplication circuit is optimal within a factor of $1.5 + o(1)$ for any circuit based on the grade-school method (i.e., bit-matrix reduction) and is optimal within a factor of $3 + o(1)$ from a trivial lower bound, assuming $\log_2 d = o(\log_k m)$. Siu et al [14, 13] have given multiplication threshold circuits of restricted fan-in, which require depth $(7\log_2(d+1)+4)\log_k n + o(\log d \log_k n) + O(1)$, edge complexity $O(\sqrt{\frac{\log k}{d}} n^2 k^{\frac{1}{d}})$, and fan-in k for any integer $d \geq 1$. Our multiplication circuit improves on the results in [14] by reducing the required depth by a factor of 3.67 asymptotically for circuits of fan-in k when $\log_k n$ is not a constant and by a factor of 4.86 asymptotically for circuits with similar fan-in and edge complexity when d is large. We also show that any symmetric function of n inputs can be evaluated using a

0-7803-5700-0/99/$10.00

unit-weight threshold circuit of fan-in k, depth approximately equal to $2\log_k n + 1.44\log_2 d$, and edge complexity $O(n^{1+1/d}\log(d+1))$. The depth of this circuit is optimal within a factor of $3+o(1)$ for the given fan-in when $\log_2 d = o(\log_k n)$. The depth of our circuit for symmetric function evaluation is smaller than the depth of the corresponding circuit given in [3] by a factor of approximately 4.17 for similar edge complexity and fan-in $k = n$.

2 Iterated Addition

The addition of two operands is the most frequently encountered operation in computer arithmetic units. We can show that addition can be performed using an AND-OR circuit that has almost linear edge complexity and is depth-optimal within a factor of $1+o(1)$ when $\log_k n$ is not a constant.

Theorem 2.1
The sum of two n-bit integers can be computed using an AND-OR circuit of depth $\log_k n + o(\log_k n) + O(1)$, *edge complexity* $O(d^2 n(\log^{\overbrace{*\cdots*}^{d-1}} n)^2)$, *and fan-in k, for any positive integer* $d = o(\log_k n) + O(1)$.

In what follows, we focus on the (m,n) iterated (multi-operand) addition problem, which is the problem of computing the sum of m integers, each of which consists of n bits. A related problem is the (m,n) sum-reduction problem, where we want to produce two integers whose sum is equal to the sum of the original m n-bit numbers. Both problems have been considered extensively in the literature, and many constructions have been proposed to solve them [9, 18, 11, 17].

2.1 The (m,n) Sum-Reduction Problem

Given n one-bit numbers, an $(n, \lceil\log_2(n+1)\rceil)$-*counter* is a circuit that produces the $\lceil\log_2(n+1)\rceil$-bit binary representation of the sum of the n bits [15]. Parallel counters are important in our constructions, since they are used as subblocks in the circuits that we will propose for the sum-reduction problem.

Lemma 2.2 *An* $(n, \lceil\log_2(n+1)\rceil)$-*counter can be constructed using a unit-weight threshold circuit of depth* 2, *edge complexity* $n^2 + O(n)$, *and fan-in n.*

We are now in a position to present circuits for the sum reduction problem. Using the techniques developed in [7, 9], we can reduce the number of operands that have to be added from $m = pr$ to $p\lceil\log_2(r+1)\rceil$ by using $(r, \lceil\log_2(r+1)\rceil)$-counters. This reduction will be used repeatedly to reduce the number of operands. Note that the larger the ratio $\frac{r}{\lceil\log_2(r+1)\rceil}$ can be made, given the constraints on the fan-in of the circuits, the faster we will be able to perform the sum-reduction operation.

We define the function $f(t)$ as the unique integer $x \le t$ that satisfies the condition

$$\begin{cases} \frac{y}{\lceil\log_2(y+1)\rceil} \le \frac{x}{\lceil\log_2(x+1)\rceil} & \text{for all } y \in [x,t], \\ \frac{y}{\lceil\log_2(y+1)\rceil} < \frac{x}{\lceil\log_2(x+1)\rceil} & \text{for all } y < x, \end{cases}$$

In other words, $r = f(t)$ achieves the largest possible value for $\frac{r}{\lceil\log_2(r+1)\rceil}$ for any integer $r \le t$. If multiple values of x maximize the ratio $\frac{x}{\lceil\log_2(x+1)\rceil}$, then r is the smallest among them. The following lemma will be useful in our analysis.

Lemma 2.3 *The* (m,n) *sum-reduction problem can be executed using a unit-weight threshold circuit of depth* 2, *edge complexity* $O(nm^3/\log m)$, *and fan-in* $g(m) = m \cdot 2^{\lceil\log_2(m-1)\rceil}$.

The following theorem supplies a tradeoff scheme between depth and edge complexity in the (m,n) sum-reduction problem. It also gives flexibility in choosing the fan-in of the gates used, which is not the case in Lemma 2.3. The main idea of the following theorem is to use Lemma 2.2 repeatedly to reduce the number of operands to a small number, and then use Lemma 2.3 to obtain the final result.

Theorem 2.4 *The* (m,n) *sum-reduction problem can be solved using a unit-weight threshold circuit of depth*

$$2\log_k m + \frac{\log_2 d}{\log_2(1+\sqrt{5})-1} + o(\log_k m + \log d) + O(1)$$

$$\approx 2\log_k m + 1.44\log_2 d,$$

edge complexity $O(nm^{1+1/d}\log(d+1))$, *and fan-in k, for any positive integer d.*

Proof: We denote the i^{th} Fibonacci number by $F(i) = F(i-1) + F(i-2)$, where $F(2) = F(1) = 1$, and let $r_i = f\left(m^{\frac{F(i)}{d}}\right)$. The (m,n) sum-reduction operation can be performed in three phases:

- Phase 1: This phase is subdivided into q stages, where q is the smallest integer satisfying $m^{\frac{F(q+1)}{d}} > k$. At stage i, $i = 1,2,\ldots,q$, we use $(r_i, \lceil\log_2(r_i+1)\rceil)$-counters to reduce the number of operands that

have to be added. Since an output bit of a counter at stage i is a linear combination of at most r_i edges (Lemma 2.2) a gate in the counter has fan-in at most r_i (see Lemma 2.2), where $r_i r_{i-1} < k$ for $i \leq q-1$. As a result, we can merge the second layer of the counters at stage $i-1$ with the first layer of the counters at stage i for $i = 2,3,4,\ldots,q-1$, without exceeding the available fan-in k. Therefore, a threshold circuit of depth $q+2$ suffices for Phase 1.

After the reduction in the number of operands achieved in Phase 1, we can use counters of fan-in $f(k)$ to continue reducing the number of operands, which requires edge complexity $o(nm^{1+\frac{1}{d}})$ per stage.

- Phase 2: In this phase we use $(f(k), \lceil \log_2(f(k)+1) \rceil)$-counters to continue reducing the number of operands for another x stages until there are only

$$m_{q+x} \leq \min\left(O(m^{\frac{d+1}{4d}} \log m), g^{-1}(k)\right) \quad (1)$$

operands left, where the function $g^{-1}(k)$ is the inverse of the function $g(m)$ defined in Lemma 2.3.

- Phase 3: At the beginning of phase 3 we are left with m_{q+x} operands that have to be added. The sum-reduction problem can now be solved using Lemma 2.3.

From Lemma 2.2, Phase 1 can be executed using a unit-weight threshold circuit of depth $q+2$. Also, since the circuit at each stage in Phase 1 has edge complexity $O(nm^{1+\frac{1}{d}})$, the total edge complexity for Phase 1 is $O(nm^{1+\frac{1}{d}} \log(d+1))$. From Lemma 2.2, the x stages of Phase 2 have depth equal to $2x$, fan-in equal to $f(k)$, and edge complexity $o(nm^{1+1/d})$. The depth required to implement Phase 3 is equal to 2. We denote by m_i the number of operands left after stage i of Phase 2 that have to be added to obtain the result. It can be seen that an upper bound on m_i, for $i > q$, is given by

$$m_i \leq \left\lceil \frac{m_{i-1}}{f(k)} \right\rceil \cdot \lceil \log_2(f(k)+1) \rceil. \quad (2)$$

Since from Eq. (1) the number m_{q+x} of operands left after Phase 2 is $O(m^{\frac{d+1}{4d}} \log m)$, Phase 3 has edge complexity at most $O(nm^{1+\frac{1}{d}})$ from Lemma 2.3. Since m_{q+x} is no more than $g^{-1}(k)$, the fan-in of the circuit that implements Phase 3 is at most equal to $g(g^{-1}(k)) = k$ from Lemma 2.3. Since the depth of each stage in Phases 2 and 3 is equal to two, the threshold circuit constructed above for the (m,n) reduction problem has depth $q+2x+4$, fan-in no more than k, and edge complexity $O(nm^{1+1/d} \log(d+1))$.

To find the depth of the circuit, we need to compute the numbers of stages q and x required for Phases 1 and 2, respectively. Since $r_{q+1} = f\left(m^{\frac{F(q+1)}{d}}\right) > k$ and $F(q+1) = (\phi^{q+1} - \hat{\phi}^{q+1})/\sqrt{5}$ [8], we have

$$\frac{1}{\sqrt{5}}(\phi^{q+1} - \hat{\phi}^{q+1}) > \frac{d \log_2 k}{\log_2 m},$$

where $\phi = (1+\sqrt{5})/2$ and $\hat{\phi} = (1-\sqrt{5})/2$. Therefore, the value of q is given by

$$q = \frac{\log_2 d - (\log_2 \log_2 m - \log_2 \log_2 k)}{\log_2 \phi} + o(\log d + \log\log k).$$

We can also show that

$$x \leq \left\lceil \frac{\log_2 m}{\log_2 k - \log_2 \lceil \log_2 (k+1) \rceil} \right\rceil + O(1),$$

and the result follows. The details are omitted in this paper. □

The depth of our circuit is smaller than the depth of the circuit given in [3] when the fan-in $k = m$ (the results in [3] were developed only for the case $k = m$), and is smaller than that given in [3] by a factor of 4.17 asymptotically. Theorem 2.4 provides a way to trade off depth for edge complexity with any restricted fan-in k not exceeding m. Such flexibility is also provided in the (n,n) sum-reduction circuit presented in [14, 13], which requires depth $7\log_2(d+1)\log_k n + o(\log d \log_k n) + O(1)$, edge complexity $O(\sqrt{\frac{\log k}{d}} n^2 k^{\frac{1}{d}})$, and fan-in k for any integer $d \geq 1$. Theorem 2.4 improves on the results in [14, 13], by reducing the required depth by a factor of about 3.5 asymptotically for the same fan-in k (and $d = 1$) when $\log_k n$ is not a constant and by a factor of about 4.86 asymptotically for circuits of similar size and edge complexity when d is large.

2.2 Iterated Addition

In this subsection, we turn our attention to the iterated addition problem, which is the problem of computing the sum of m n-bit integers.

Theorem 2.5 *The sum of m n-bit integers can be computed using a unit-weight threshold circuit of depth*

$$2\log_k m + \log_k n + \frac{\log_2 d}{\log_2(1+\sqrt{5}) - 1}$$

$$+o(\log_k m+\log_k n+\log d)+O(1)$$
$$\approx 2\log_k m+\log_k n+1.44\log_2 d,$$

edge complexity $O(nm^{1+1/d}\log(d+1))$, *and fan-in* k, *for any positive integer* d.

Proof: We first use the (m,n) sum-reduction circuit of Theorem 2.4 to reduce the number of operands from m to two, and then compute the sum of the two numbers using the adder of Theorem 2.1. □

A trivial lower bound on the depth required to perform iterated addition is $\log_k m+\log_k n$ since there are mn input bits. The depth of our iterated addition circuit is optimal within a factor of $1+o(1)$ when $\log_2 d=o(\log_k n)$ and $\log_2 m=o(\log n)$, and is optimal within a factor of $1.5+o(1)$ when $\log_2 d=o(\log_k n)$ and $\log_2 m\approx\log_2 n$.

3 Symmetric Functions

A Boolean function f is said to be *symmetric* if $f(x_1,\ldots,x_n)=f(x_{\pi(1)},\ldots,x_{\pi(n)})$ for any permutation $(x_{\pi(1)},\ldots,x_{\pi(n)})$ of $(x_1,\ldots,x_n)$. An important property of symmetric Boolean functions is that they are completely specified by the number of ones in their inputs (that is, by the sum $\sum_{i=1}^n x_i$). Therefore, the threshold circuits for iterated addition lead to efficient circuits for evaluating general symmetric functions.

Lemma 3.1 *Any Boolean function of* n *inputs can be evaluated by an AND-OR circuit of depth* $\frac{n}{\log_2 k}+o(\frac{n}{\log k})+O(1)$, *edge complexity* $2^n+o(2^n)$, *and fan-in* k.

Theorem 3.2 *Any symmetric function of* n *inputs can be evaluated using a unit-weight threshold circuit of depth*

$$3\log_k n+\frac{\log_2 d}{\log_2(1+\sqrt{5})-1}+o(\log_k n+\log d)+O(1)$$
$$\approx 3\log_k n+1.44\log_2 d,$$

edge complexity $O(n^{1+1/d}\log(d+1))$, *and fan-in* k, *for any positive integer* d.

Proof: We can evaluate a symmetric function in two phases:

- Phase 1: We use Theorems 2.4 and 2.1 to find the sum $\sum_{i=1}^n x_i$ of the inputs. Note that the output value of the symmetric function is completely determined by $\sum_{i=1}^n x_i$.
- Phase 2: The symmetric function can be viewed as a Boolean function of $\lceil\log_2(n+1)\rceil$ variables (namely, the bits in the binary representation of $\sum_{i=1}^n x_i$ computed in Phase 1). Therefore, we can use Lemma 3.1 to find the desired result.

□

The depth of our circuit for $k=n$ is smaller than the depth of the circuit given in [3] by a factor of 4.17 asymptotically. Theorem 2.4 also provides a mechanism to trade off depth for edge complexity with any restricted fan-in k not exceeding m. The depth of our circuit for symmetric function evaluation is optimal within a factor of $3+o(1)$ from the trivial lower bound $\log_k n$ when $\log_2 d=o(\log_k n)$.

4 Multiplication

The results obtained in Section 2 for iterated addition give rise to a fast and edge-efficient multiplier that uses threshold gates of restricted fan-in, as described in the following theorem.

Theorem 4.1 *The product of two* n*-bit numbers can be computed using a unit-weight threshold circuit of depth*

$$3\log_k n+\frac{\log_2 d}{\log_2(1+\sqrt{5})-1}+o(\log_k n+\log d)+O(1)$$
$$\approx 3\log_k n+1.44\log_2 d,$$

edge complexity $O(n^{2+1/d}\log(d+1))$, *and fan-in* k, *for any positive integer* d.

Proof: Let $X=(x_{n-1},\ldots,x_1,x_0)_2$ and $Y=(y_{n-1},\ldots,y_1,y_0)_2$ be the two integers to be multiplied. We will transform the problem into the problem of finding the sum of n n-bit numbers by means of bit-matrix reduction (i.e., the grade-school method). The binary numbers

$$p_{j,i+j}\stackrel{\text{def}}{=}x_i\wedge y_j=\mathrm{Sgn}(x_i+y_j-2),$$

for $i=0,1,\ldots,n-1$, $j=0,1,\ldots,n-1$, can be computed using a unit-weight threshold circuit of depth one. We then have

$$X\cdot Y=\sum_{j=0}^{n-1}P_j, \tag{3}$$

where $P_j=(p_{j,\lceil\log_2 m_i\rceil+j-1},\ldots,p_{j,j},\overbrace{0,\ldots,0}^{j})_2$, for $j=0,1,\ldots,n-1$. The construction is completed by observing that the summation in Eq. (3) corresponds to an $(n,2n-1)$ iterated addition, and using Theorem 2.5. □

Siu et al [13, 14] have given multiplication threshold circuits of restricted fan-in, which require depth $(7\log_2(d+1)+4)\log_k n + o(\log d \log_k n) + O(1)$, edge complexity $O(\sqrt{\frac{\log k}{d}} n^2 k^{\frac{1}{d}})$, and fan-in k for any integer $d \geq 1$. Theorem 4.1 improves on the results in [14], by reducing the required depth by a factor of 3.67 asymptotically for circuits of fan-in k (and $d = 1$) when $\log_k n$ is not a constant and by a factor of 4.86 asymptotically for circuits with similar fan-in and edge complexity when d is large.

The depth of our circuit for multiplication is optimal within a factor of $3 + o(1)$ from a trivial lower bound $\log_k 2n$ when $\log_2 d = o(\log_k n)$. Since any multiplication circuit based on bit-matrix reduction has n^2 intermediate values, each of which may affect the most significant bit of the product, the depth of our circuit is optimal within a factor of $1.5 + o(1)$ from the lower bound for any multiplication circuit using bit-matrix reduction.

5 Conclusion

We have proposed several threshold circuits to perform iterated addition and multiplication and to evaluate symmetric functions. Our constructions provide effective tradeoffs among edge complexity, circuit depth, and maximum fan-in through the flexibility provided in the choice of the parameters k (fan-in) and d (levels of hierarchy). Our circuits appear to be considerably more depth-efficient than the best previous circuits, assuming similar edge complexity and fan-in (or, alternatively, considerably more cost-effective for similar circuit depth). Moreover, the depths of all the circuits presented in this paper are optimal within a small constant factor with any fan-in restriction.

References

[1] E. Allender, "A note on the power of threshold circuit," *Proc. Symp. Foundations of Computer Science,* pp. 580-585, 1989.

[2] N. Alon and J. Bruck, "Explicit constructions of depth-2 majority circuits for comparison and addition," *SIAM J. Disc. Math.,* pp. 1-8, 1994.

[3] P. Beame, E. Brisson, and R. Ladner, "The complexity of computing symmetric functions using threshold circuits," *Theoretical Comput. Sci.,* pp. 253-265, 1992.

[4] R. Boppana, "Threshold functions and bounded depth monotone circuits," *ACM Symp. Theory of Computing,* pp. 475-479, 1984.

[5] J. Bruck, "Harmonic analysis of polynomial threshold functions," *SIAM J. Disc. Math.,* pp. 168-177, 1990.

[6] J. Bruck, "Computing with networks of threshold elements," Ph.D. dissertation, Dept. Electrical Engineering, Stanford Univ., 1989.

[7] L. Dadda, "Some schemes for parallel multipliers," *Alta Freq.*, pp. 349-356, 1965.

[8] Graham, R.L., D.E. Knuth, and O. Patashnik, *Concrete Mathematics: A Foundation For Computer Science,* Addison-Wesley, Menlo Park, CA, 1994.

[9] I.T. Ho and T.C. Chen, "Iterated addition by residue threshold functions and their representation by array logic," *IEEE Trans. Comput.,* Vol. C-22, pp. 762-767, 1973.

[10] I. Parberry, *Circuit Complexity and Neural Networks,* Cambridge, Mass., MIT Press, 1994.

[11] B. Parhami, *Computer Arithmetic: Algorithms and Hardware Designs*, Oxford University Press, 2000.

[12] J. Reif and S.R. Tate "On threshold circuits and polynomial computation," *SIAM J. Comput.*, Vol. 21, No. 5, pp. 896-908, 1992.

[13] K.Y. Siu, V.P. Roychowdhury, and T. Kailath, *Discrete Neural Computation, A Theoretical Foundation,* Prentice Hall, New Jersey, 1995.

[14] K.Y. Siu, V.P. Roychowdhury, and T. Kailath, "Toward massively parallel design of multipliers," *J. Parallel Distributed Computing*, No. 24, pp. 86-93, Jan. 1995.

[15] E.E. Swartzlander, "Parallel Counters," *IEEE Trans. Computers,* Vol. 29, pp. 1021-1024, 1973.

[16] S. Vassilladis, S. Contofana, and K. Bertels, "2-1 addition and related arithmetic operations with threshold logic," *IEEE Trans. Computers,* Vol. 45, no. 9, pp. 1062-1067, Sep. 1996.

[17] C.S. Wallace, "A suggestion for a fast multiplier," *IEEE Trans. Computers,* Vol. EC-13, pp. 14-17, 1964.

[18] C.-H. Yeh and B. Parhami, "Efficient pipelined multioperand adders with high throughput and low latency: designs and applications," *Proc. Asilomar Conf. Signals, Systems, and Computers,* pp. 894-898, 1996.

[19] C.-H. Yeh and E. A. Varvarigos, "New efficient majority circuits for the computation of some basic arithmetic functions," *J. Comput. Information,* pp. 114-136, 1996.

[20] C.-H. Yeh and E. A. Varvarigos, "Depth-efficient threshold circuits for multiplication and symmetric function computation," *Proc. Int'l Computing and Combinatorics Conf., LNCS,* pp. 231-240, 1996.