Linear-Size Constant-Depth Polylog-Threshold Circuits

Prabhakar Ragde*
University of Waterloo

Avi Wigderson†
Hebrew University

Keywords: threshold circuits, AC^0, constant-depth circuits, approximate compaction.

Abstract: We present a simple explicit construction giving unbounded fan-in circuits with $o(n)$ gates and depth $O(r)$ for the threshold function of n variables when the threshold is at most $(\log n)^r$, for any integer $r > 0$. This improves a result of Ajtai and Ben-Or, who showed the existence of circuits of size $n^{O(1)}$. This is the highest threshold for which polynomial-size, constant-depth circuits are possible.

1. Introduction. The class AC^0 (functions computable by polynomial-size constant-depth unbounded fan-in circuits) has been the subject of much interest. In the interests of efficient computation, it is natural to consider which functions are in LC^0, that is, computable by constant-depth unbounded fan-in circuits with a linear number of gates. An even more restrictive class is WLC^0, in which only a linear number of wires is allowed. As an example of a WLC^0 function, consider the function $f(x_1, x_2, \ldots x_n) = (y_1, y_2, \ldots y_n)$, where $y_i = 1$ if and only if x_i is the input bit of lowest index with value 1 ([FRW,CFL]). Addition of two n-bit numbers has been shown to be in LC^0 but not WLC^0 [CFL]; there is no 1-output function known to separate AC^0 from LC^0 or LC^0 from WLC^0. In this paper, we will consider the size of a circuit to be the number of gates.

The threshold function Th_k^n has value 1 if and only if at least k of the n variables have value 1. As with all symmetric functions, these can be computed by bounded fan-in circuits of size $O(n)$ and depth $O(\log n)$ [MP]. Chandra, Stockmeyer, and Vishkin [CSV] give a construction for unbounded fan-in circuits which achieves $2^{O(n^{1/d})}$ gates and wires and depth $O(d)$, for any $d > 2$; Hastad [Ha] has proved that, up to the constant hidden in the O-notation, this is the smallest size achievable for $k = n/2$.

This does not rule out the possibility that for smaller values of k, smaller circuits may exist; in fact, it is easy to construct circuits of size $O(n^k)$ for Th_k^n. Threshold-2 is in WLC^0, a fact which can be seen by noting that $Th_2^n(x_1, x_2, \ldots, x_n) = \vee_{i=1}^n (x_i \wedge \bar{y}_i)$, where the bits

* Supported by an operating grant from the National Sciences and Engineering Research Council of Canada.

† Partially supported by American-Israeli Binational Science Foundation Grant No. 87-00082.

y_i are described in the first paragraph of this section. Ajtai and Ben-Or [AB] demonstrate a general technique for converting probabilistic AC^0 circuits into deterministic AC^0 circuits, and use this technique to show that Th_k^n is in AC^0, provided that $k = \log^{O(1)} n$. Their method proves the existence of circuits of size $O(n^r)$ and depth $O(r)$, where $k \leq (\log n)^r$. Uniform polynomial-size AC^0 circuits for polylog threshold functions were described by Denenberg, Gurevich, and Shelah [DGS]. They were not interested in optimizing size, and their constructions do not compete with subsequent polynomial-size constructions such as in Fagin, Klawe, Pippenger, and Stockmeyer [FKPS]. Note that Hastad's result [Ha] implies that if $k = \log^{\omega(1)} n$, Th_k^n is not in AC^0.

It is natural to conjecture that polylog threshold functions are candidates for separating AC^0 and LC^0; this paper shows such a conjecture to be false, by giving a simple uniform construction of circuits of size $o(n)$ and depth $O(r)$ for Th_k^n when $k \leq (\log n)^r$. Unfortunately, this construction does not place polylog threshold functions in WLC^0, since $O(nlog^{4r+1}n)$ wires are used. It was not previously known that linear-sized circuits were possible; indeed this is somewhat surprising, since even the fact that these functions are in AC^0 is non-trivial. Independently, Wegener, Wurm and Yi [WWY] proved a result similar to ours, using a similar method; their construction can be made monotone.

2. The Circuit. The first part of the circuit performs what we may call approximate compaction: it either finds that there are more than k 1's on the input wires, or compresses the 1's onto at most $n' = k^4$ wires. Then, since n' is now small relative to n, the construction of [CSV] can be applied.

The approximate compaction is achieved through the use of perfect hash functions, as described in the following theorem.

Theorem 1. *For any k, there exists an unbounded fan-in circuit of size $o(n)$ and depth $O(1)$ which has n inputs $x_1, x_2, \ldots, x_n$ and $k^4 + 1$ outputs $z, y_1, y_2, \ldots y_{k^4}$ such that for any input, $z = 0$ if and only if $\sum_{i=1}^{n} x_i = \sum_{i=1}^{k^4} y_i$.*

Proof: It suffices to consider $k \leq n^{1/4}$. For clarity, we first describe an algorithm running on a CRCW PRAM with n processors; if several processors simultaneously write into a cell, an arbitrary processor succeeds. We also assume $m = \sqrt{n}$ is an integer; this assumption is easily removed. The algorithm first divides the input bits into m groups of size m and takes the OR of each group. It then attempts to map the m group bits into space k^2. This is accomplished by means of a set of perfect hash functions: that is, a set $\{f_\alpha \mid \alpha = 1, 2, \ldots s\}$, where each f_α maps $\{1, 2, \ldots n\}$ into $\{1, 2, \ldots k^2\}\}$, such that for each subset S of $\{1, 2, \ldots, n\}$ of size at most k, there exists a value of α such that $f_\alpha|_S$ is one-to-one. Mehlhorn [Mehl] gives a probabilistic argument that in our case proves the existence of a family of perfect hash functions of size $s = O(k \log n)$; if uniformity is a consideration, he also gives a deterministic construction of a family of size $s = O(k^2 \log n)$.

The algorithm devotes m processors to each possible value of α. The m processors each take one group bit and attempt to map it; if a collision is detected, that value of α can be discarded. In $O(1)$ time, then, the algorithm either finds a value of α that achieves the

one-to-one mapping, or finds that no such value exists (in which case there are more than k 1's in the group bits and thus in the input). This value of α suffices to map the original input bits into k^2 blocks of size $\sqrt{n}$, or space $k^2\sqrt{n}$. The process is now repeated, with the groups being of size k^2 in the second phase, resulting in compression of the original input bits into k^2 blocks of size k^2, or total space k^4.

For ease in precisely describing the circuit implementation of this algorithm, let us rename the input bits $x_1, x_2, \ldots, x_n$ as $\{x_{i,j}\}$, where $x_{i,j}$ is the jth bit within group i. The circuit forms the bits $a_i = \vee_{j=1}^{m} x_{i,j}$ for $i = 1, 2, \ldots, m$; a_i is the ith group bit. To test whether a particular hash function f_α is good or not, we need threshold-2 circuits; a WLC^0 construction has already been described. Computing $b_{\alpha,h} = Th_2(\{a_i \mid f_\alpha(i) = h)$ (for $\alpha = 1, 2, \ldots, s$, $h = 1, 2, \ldots, k^2$) determines whether function f_α causes a collision in block h when the group bits are hashed to space k^2. Since for any fixed α, each a_i is an input to exactly one such threshold circuit, and the number of wires in the circuit is linear in the number of inputs, the total number of wires used for fixed α is $O(m)$. Thus $O(ms)$ wires are used here. Computing $c_\alpha = \vee_{h=1}^{k^2} b_{\alpha,h}$ for $\alpha = 1, 2, \ldots, s$ determines if function f_α is bad. The situation where all functions are bad is captured by $z_1 = \wedge_{\alpha=1}^{s} c_\alpha$.

At this point there may be several good hash functions; one is selected by computing $d_\alpha = (\wedge_{\beta=1}^{i-1} c_\beta) \wedge \bar{c}_\alpha$. The bit d_i is 1 if and only if function i is selected. Finally, the first phase of the compression is completed by computing the ith bit of the new block h, namely $f_{h,i} = \vee_{\alpha=1}^{s} d_\alpha \wedge \vee\{x_{j,i} \mid \exists j, f_\alpha(j) = h\}$ for $i = 1, 2, \ldots, m$,$h = 1, 2, \ldots, k^2$.. This last computation takes a total of $O(k^2ms)$ gates and $O(k^2m^2s)$ wires.

For the second phase of the compression, we switch the roles of the indices (that is, we consider k^2 groups, each of r bits) and repeat the previous development. Letting $x'_{i,j} = f_{j,i}$, we have: $a'_i = \vee_{j=1}^{m} x'_{i,j}$ for $i = 1, 2, \ldots, m$; $b'_{\alpha,h} = Th_2(\{a'_i \mid f_\alpha(i) = h)$; $c'_\alpha = \vee_{h=1}^{k^2} b'_{\alpha,h}$; $z_2 = \wedge_{\alpha=1}^{s} c'_\alpha$; $d'_\alpha = (\wedge_{\beta=1}^{i-1} c_\beta) \wedge \bar{c}'_\alpha$; $f'_{h,i} = \vee_{\alpha=1}^{s} d'_\alpha \wedge \vee\{x'_{j,i} \mid \exists j, f_\alpha(j) = h\}$. Renaming the $\{f'_{h,i}\}$ as $y_1, y_2, \ldots y_{k^4}$ and letting $z = z_1 \vee z_2$ concludes the formal description of the circuit. ■

Theorem 3. *If $k \leq (\log n)^d$ for some integer $d > 0$, then Th_k^n is in LC^0.*

Proof: We first use the approximate compression circuit to compress the input bits onto k^4 wires. The construction in [CSV] gives a circuit for $Th_k^{k^4}$ of depth $O(d)$ with at most most $2^{c(\log^{4d} n)^{1/4(d+1)}}$ or $o(n)$ gates and wires. Connecting the outputs $y_1, y_2, \ldots y_{k^4}$ of the approximate compression circuit to the inputs of the $Th_k^{k^4}$ circuit and taking the conjunction of the output of the $Th_k^{k^4}$ circuit with z yields a Th_k^n circuit with $o(n)$ gates and $O(k^4 n \log n)$ wires. ■

The non-monotonicity of the circuit comes partly from the need to select a unique hash function at each level. Wegener, Wurm, and Yi [WWY], independently working in the more general setting of all symmetric functions within AC^0, construct monotone circuits which in the case of threshold functions are of the same size as our construction. Further applications of approximate compaction to the problems of approximate counting and exact compaction are given in [Rag].

Acknowledgements. We would like to thank Ilan Newman for useful discussions. We would also like to thank the anonymous referee for pointing out an error in our original proof of Theorem 1, and for many stylistic suggestions.

References

[AB] M. Ajtai and M.Ben-Or, A theorem on probabilistic constant depth computation, Proc. 16th ACM STOC (1984), 471-474.

[CFL] A. Chandra, S. Fortune, and R. Lipton, Unbounded fan-in circuits and associative functions, J. Comp. Syst. Sci. 30 (1986), 222-234.

[CSV] A. Chandra, L. Stockmeyer, and U. Vishkin, Constant depth reducibility, SIAM J. Comp. 13 (1984), 423-429.

[DGS] L. Denenberg, Y. Gurevich, and S. Shelah, Definability by constant-depth polynomial-size circuits, Information and Control 70 (1986), 216-240.

[FKPS] R. Fagin, M.M. Klawe, N.J. Pippenger, and L. Stockmeyer, Bounded-depth, polynomial-size circuits for symmetric functions, Theor. Comp. Sci 16 (1985), 239-250.

[FRW] F.E. Fich, P. Ragde, and A. Wigderson, Relations among concurrent-write PRAMs (preliminary version), Proc. 3rd ACM PODC (1984), 179-189.

[Ha] J. Hastad, Almost optimal lower bounds for small depth circuits, Proc. 18th ACM STOC, 1986, 6-20.

[Mehl] K. Mehlhorn, On the program size of perfect and universal hash functions, Proc. 23rd IEEE FOCS (1982), 170-175.

[MP] D.E. Muller and F.P. Preparata, Bounds to complexities of networks for sorting and for switching, J. ACM 22 (1975), 195-201.

[Rag] P. Ragde, The parallel simplicity of compaction and chaining, Proc. 17th ICALP (1990).

[WWY] I. Wegener, N. Wurm, and S.-Z. Yi, Symmetric functions in AC^0 can be computed in constant depth with very small size. Universität Dortmund, Forschungsbericht Nr. 326, 1989, also in Proc. 15th MFCS, 1990, pp.523-9.