OPTIMAL DEPTH, VERY SMALL SIZE CIRCUITS FOR SYMMETRIC FUNCTIONS IN AC^0

Johan Hastad, Royal Institute of Technology
10044 Stockholm, Sweden

and

Ingo Wegener*, Norbert Wurm and Sang-Zin Yi
FB Informatik, LS II, Univ. Dortmund,
Postfach 500 500, 4600 Dortmund 50, Fed. Rep. of Germany

Abstract
It is well-known which symmetric Boolean functions can be computed by constant depth, polynomial size, unbounded fan-in circuits, i.e. which are contained in the complexity class AC^0. This result is sharpened. Symmetric Boolean functions in AC^0 can be computed by unbounded fan-in circuits with the following properties. If the optimal depth of AC^0-circuits is d, the depth is at most $d+2$, the number of wires is almost linear, namely $n \log^{O(1)} n$, and the number of gates is subpolynomial (but superpolylogarithmic), namely $2^{O(\log^{\delta} n)}$ for some $\delta < 1$.

* * Supported in part by DFG grants No. We 1066/2-1 and Me 872/1-2

1. Introduction

Symmetric functions form an important subclass of Boolean functions including all kinds of counting functions. A Boolean function $f : \{0,1\}^n \to \{0,1\}$ is called symmetric if $f(x_1, \ldots, x_n)$ depends on the input only via $x_1 + \ldots + x_n$, the number of ones in the input. Hence, symmetric functions f can be described by value vectors $v(f) = (v_0, \ldots, v_n)$ where v_i is the output of f on inputs with exactly i ones.

It is a classical result of Boolean complexity theory that all symmetric Boolean functions are contained in NC^1. They can even be computed by fan-in 2 circuits of logarithmic depth and linear size. We are interested in the more massive parallel, unbounded fan-in circuits. AC^0 is the class of Boolean functions computable by unbounded fan-in circuits of constant depth and polynomial size. The following theorem by Moran (1987) and Brustmann and Wegener (1987) is based on the lower bounds of Boppana (1984) and Hastad (1986) and the upper bounds due to Ajtai and Ben-Or (1984), Denenberg, Gurevich and Shelah (1986) and Fagin, Klawe, Pippenger and Stockmeyer (1985).

Theorem 1: A sequence of symmetric Boolean functions $f = (f_n)$ with value vectors $v(f_n) = (v_0^n, \ldots, v_n^n)$ is contained in AC^0 iff $v_{g(n)}^n = \ldots = v_{n-g(n)}^n$ for some polylogarithmic function g, i.e. $g(n) = O(\log^k n)$ for some k.

This theorem does not answer whether symmetric functions in AC^0 can be computed by AC^0-circuits which are efficient compared with the well-known logarithmic depth, linear size, fan-in 2 circuits for symmetric functions. A similar question has been asked for adders. The carry-look-ahead method leads to an unbounded fan-in circuit with optimal depth 3 (adders of depth 2 need exponential size), $\Theta(n^2)$ gates and $\Theta(n^3)$ wires. But these adders cannot compete with the well-known fan-in 2 adders of linear size and logarithmic depth. The problem of determining the complexity of unbounded fan-in constant depth adders has been solved by Chandra, Fortune and Lipton (1983) for the upper bounds and by Dolev, Dwork, Pippenger and Wigderson (1983) for the lower bounds. For any recursive function $g(n)$ where $g(n) \to \infty$ as $n \to \infty$ there are constant depth adders of size $O(ng(n))$ but there do not exist linear size adders of constant depth. The results of this paper go into the same direction.

It is proved that symmetric Boolean functions in AC^0 can be computed by unbounded fan-in circuits with constant depth, an almost linear number of $n \log^{O(1)} n$ wires and a subpolynomial but superlogarithmic number of $2^{O(\log^\delta n)}$ gates for some $\delta < 1$. We improve the best known upper bounds for the depth of AC^0-circuits by a factor of approximately 2. If the optimal depth of AC^0-circuits is d, the depth of our circuits is at most $d + 2$.

In Section 2 we review some known results and discuss some lower bounds. In Section 3 we reformulate the method of Denenberg, Gurevich and Shelah (1986) on which our circuit design presented in Section 4 is based.

2. Known results and simple lower bounds

Many of the known results are stated only for threshold functions which form some kind of basis for all symmetric functions. The threshold function T_k^n on n variables computes 1 exactly on those inputs with at least k ones, i.e. $v_l(T_k^n) = 1$ iff $l \geq k$. $NT_k^n := \neg T_{k+1}^n$, is the corresponding negative threshold function, $v_l(NT_k^n) = 1$ iff $l \leq k$. $E_k^n := T_k^n \wedge NT_k^n$ is called exactly function, $v_l(E_k^n) = 1$ iff $l = k$. A symmetric function f is obviously the disjunction of all E_k^n where $v_k(f) = 1$.

We have already mentioned that the upper bounds of Theorem 1 have been proved independently in three papers. Different methods have been used. For T_k^n where $k = \lfloor \log^m n \rfloor$ Ajtai und Ben-Or (1984) proved the existence of circuits of depth $2m + 3$ and size $\Theta(n^{2m+4} \log^{m+1} n)$. The circuit whose existence has been proved by Fagin, Klawe, Pippenger and Stockmeyer (1985) also has depth $2m + 3$ but the size is larger, namely $n^{\Theta(m^2)}$. Denenberg, Gurevich and Shelah (1986) could even construct AC^0-circuits. They were only interested in the qualitative result that the circuits are AC^0-circuits. A direct implementation leads to circuits which are less efficient with respect to depth and size compared with the other circuits. Our circuit design uses the main idea of Denenberg, Gurevich and Shelah (1986) (see also Mayr (1985)), the efficient coding of the cardinality of small subsets of $\{0, \ldots, n-1\}$ by short 0-1-vectors, see Lemma 1. Since we work directly with circuits and do not use the notation of logics, we are not concerned with the size of the "universe". This simplifies our approach. Furthermore, we present an iterative circuit design and use some implementation tricks. This leads to the uniform design of monotone circuits for T_k^n, where $k = \lfloor \log^m n \rfloor$, with the following characteristics. The depth is $\lfloor m \rfloor + 3$ (m must not be an integer), the number of gates is $g = 2^{O(\log^{m/(\lfloor m \rfloor + 1)} n \log \log n)}$ and, hence, $o(n^\alpha)$ for all $\alpha > 0$, and the number of wires equals $O(n \log^{2m+2} n)$. By using some more wires, namely $O(ng)$ wires, we can decrease the depth to $\lfloor m \rfloor + 2$, which is optimal, if m is not an integer.

Independently from our approach and with some other methods Newman, Ragde and Wigderson (1990) have worked in the same direction. They have designed uniformly circuits of depth $O(m)$, approximately $4m$, number of gates $O(n)$ and number of wires $O(n \log^{2m} n)$. Because of their use of hash functions the circuit is not monotone. This design beats our bounds only for the number of wires and is worse else.

For constant k better results are possible. Friedman (1984) has investigated the formula size of threshold functions. Using his methods the following theorem can be proved in a straightforward way.

Theorem 2: For constant k, T_k^n can be computed by unbounded fan-in circuits of depth 3 with $O(\log n)$ gates and $O(n \log n)$ wires.

In order to appreciate the new upper bounds we discuss some lower bounds. Each Boolean

function can be computed with depth 2 by its DNF. But what is the minimal depth of circuits with polynomial size ? And what is the minimal size if the depth is bounded or even unbounded ? The following lower bound has been proved by Boppana (1984) for monotone circuits and by Hastad (1986) in the general form.

Theorem 3: Polynomial size, unbounded fan-in circuits for T_k^n where $k = \lfloor \log^m n \rfloor$ and $m \in \mathbb{R}^+$ have depth $d \geq \lceil m \rceil + 1$.

There are not too many small size lower bounds on the number of gates and wires of unbounded fan-in, unbounded depth circuits for functions in AC^0 or NC^1. Hromkovic (1985) proved some lower bounds by a communication complexity approach and Wegener (1990) adapted the elimination method for proving lower bounds on the complexity of the parity function. For threshold functions we only know the following simple lower bounds.

Proposition 1: Unbounded fan-in circuits for T_k^n and $1 \leq k \leq n/2$ need at least n wires and k gates.

Proof: Obviously, at least one wire has to leave each variable x_i. If x_i enters an $\vee$-gate or $\bar{x}_i$ enters an $\wedge$-gate, we can eliminate this gate for $x_i = 1$. Otherwise one gate can be eliminated for $x_i = 0$. This procedure can be repeated at least k times before T_k^n is replaced by a constant function. □

3. The method of Denenberg, Gurevich and Shelah

We reformulate the method of Denenberg, Gurevich and Shelah (1986) in a generalized form. We use a representation supporting our circuit design. The method is based on a number theoretic theorem allowing a succint coding of the cardinality of small subsets of $\{0, \ldots, n-1\}$. Let $\mathrm{res}(i,j) := (i \bmod j) \in \{0, \ldots, j-1\}$.

Lemma 1: Let $L := L(n)$. For large n one can choose for each small subset S of $\{0, \ldots, n-1\}$, i.e. $|S| \leq L$, some number $u < L^2 \log n$ such that $\mathrm{res}(i,u) \neq \mathrm{res}(j,u)$ for all different $i, j \in S$.

Proof: The proof relies on the prime number theorem in the following form.

$$\lim_{x \to \infty} \psi(x)/x = 1 \text{ for } \psi(x) := \sum_{p \text{ prime}, p^k \leq x < p^{k+1}} ln\ p^k.$$

Let $S \subseteq \{0, \ldots, n-1\}$ be some small set, i.e. $|S| \leq L$. Let u be the smallest number such that $\mathrm{res}(i,u) \neq \mathrm{res}(j,u)$ for all different $i, j \in S$. For large n, either $u < L^2 \log n$ or $\psi(u-1) > (u-1) \ln 2$. We prove that $\psi(u-1) > (u-1) \ln 2$ implies $u < L^2 \log n$ and, hence, we prove the lemma.

Let a be the least common multiple of all $|i-j|$, where $i,j \in S$ and $i \neq j$, and let b be the product of all p^k where p is prime and $p^k \leq u-1 < p^{k+1}$. By definition of u, $\mathrm{res}(i,p^k) = \mathrm{res}(j,p^k)$ for some different $i,j \in S$. Hence, $|i-j|$ is a multiple of p^k. This implies that p^k and also b divides a. Hence, $b \leq a$. Also, $a < n^{L(L-1)/2}$, since, by assumption, S has at most $\binom{L}{2}$ subsets $\{i,j\}$ where $i \neq j$ and $|i-j| < n$ for $i,j \in S$. By definition $\ln b = \psi(u-1)$.

Combining all our inequalities we have $(u-1)\ln 2 < \psi(u-1) = \ln b \leq \ln a < L^2 \ln n$ or $u < L^2 \log n$. □

This lemma can be applied several times. For the second application n is replaced by $n' := L^2 \log n$ and S is replaced by $S' := \{i | i = \mathrm{res}(j,u) \text{ for some } j \in S\}$. It should be emphasized that $u = u(S)$ depends on S. For our circuit design this repeated application of the lemma is only of limited use. In order to keep the depth of the circuit small, we work with $L = \log^\delta n$ for some $\delta < 1$ but $\delta \approx 1$. After the first application of the lemma u can be estimated by $\log^{1+2\delta} n$ and the upper bound for u does not become smaller than $(\log^{2\delta} n)\log\log n$.

We shall see that large L corresponds to large size and small depth, and small L corresponds to small size and large depth. For the size of the circuit the function L^L is important. If $L = O(\log^\delta n)$ for some $\delta < 1$, $L^L = 2^{O((\log^\delta n)\log\log n)} = o(n^\alpha)$ for all $\alpha > 0$. If $L = \Omega(\log n)$, L^L grows superpolynomial.

4. The construction of small depth, small size circuits

We start with a simple but important subcircuit.

Lemma 2: NT_1^n can be computed by a circuit of depth 3 with $3\lceil \log n \rceil + 1$ gates and $(n+3)\lceil \log n \rceil$ wires. The output gate of the circuit is an $\wedge$-gate.

Proof: We assume that $n = 2^k$, otherwise one may consider NT_1^N where $N = 2^{\lceil \log n \rceil}$ and may replace $N-n$ inputs by zeros. NT_1^n is the conjunction of all prime clauses $\bar{x}_i \vee \bar{x}_j$, $i \neq j$. We use a so-called separating system to compute these prime clauses. By $(\bar{x}_0 \wedge \ldots \wedge \bar{x}_{n/2-1}) \vee (\bar{x}_{n/2} \wedge \ldots \wedge \bar{x}_{n-1})$ we compute with 3 gates and $n+2$ wires the conjunction of all prime clauses $\bar{x}_i \vee \bar{x}_j$ where the first bit of the k-bit number i equals 0 while the first bit of j equals 1. The same can be done for the other $k-1$ bits of the numbers $0,\ldots,n-1$. Finally, the k outputs of the $\vee$-gates are combined by an $\wedge$-gate. □

If we like to decrease the depth to 2, $\Theta(n^2)$ gates and wires are necessary and sufficient.

We are interested in the design of efficient circuits for symmetric functions in AC^0. Since f and $\neg f$ have the same complexity, it is by Theorem 1 sufficient to consider symmetric functions f where $v_k(f) = 1$ only for some k where k or $n - k$ is bounded by a polylogarithmic function. Furthermore, by duality, we may restrict ourselves to functions f where $v_k(f) = 1$ only for some k where k is bounded by a polylogarithmic function.

Let $L = L(n)$ be a function specified later and let f be a symmetric function where $v_k(f) = 1$ only for some $k \leq L$.

The first step of our circuit design is an application of the coding lemma. For all $u \in \{1, \ldots, U\}$ where $U := \lfloor L^2 \log n \rfloor$ we compute in parallel the following information:

$$y^u = (y^u_o, \ldots, y^u_{u-1}), \bar{y}^u = (\bar{y}^u_o, \ldots, \bar{y}^u_{u-1}), c^u.$$

Here $y^u_j := 1$ iff $x_i = 1$ for some i where $\mathrm{res}(i, u) = j$, $\bar{y}^u_j$ is the negation of y^u_j, and c^u is the so-called validity bit, i.e. $c^u := 1$ iff for all $j \in \{0, \ldots, u-1\}$ there is at most one i such that $x_i = 1$ and $\mathrm{res}(i, u) = j$.

Altogether we have replaced the n inputs by less than $2U^2 + U$ "new inputs". If $c^u = 1$, y^u contains the same number of ones as x. Since symmetric functions depend only on the number of ones in the input and not on their positions, y^u is in this case a valid encoding of the input. By Lemma 1 there exists at least one valid encoding, if $f(x) = 1$.

The computation of $y^u, \bar{y}^u$ and c^u is easy. Let $A(j, u)$ be the set of all i where $\mathrm{res}(i, u) = j$. Then y^u_j is the disjunction of all x_i, $i \in A(j, u)$, and $\bar{y}^u_j$ is the conjunction of all $\bar{x}_i$, $i \in A(j, u)$. y^u and $\bar{y}^u$ can be computed in depth 1 with $2n$ wires and $2u$ gates. The validity bit c^u is the conjunction of $b^u_o, \ldots, b^u_{u-1}$ where b^u_j is the NT_1-function for the variables $x_i, i \in A(j, u)$. Since the NT_1-circuits of Lemma 2 compute the output at an $\wedge$-gate, the conjunction for the computation of c^u can be combined with these $\wedge$-gates. Let $n_j := |A(j, u)|$. Then $n_o + \ldots + n_{u-1} = n$. Hence, c^u can be computed in depth 3 (last gate is an $\wedge$-gate) with

$$3 \sum_{0 \leq j \leq u-1} \lceil \log n_j \rceil + 1 = O(U \log n)$$

gates and

$$\sum_{0 \leq j \leq u-1} (n_j + 3) \lceil \log n_j \rceil = O(n \log n)$$

wires.
Hence, all $y^u, \bar{y}^u$ and c^u can be computed in depth 3 with $O(U^2 \log n)$ gates and $O(nU \log n)$ wires.

Let $v = (v_o, \ldots, v_n)$ be the value vector of f. We consider the symmetric function f^u on u variables with value vector $v^u := (v_o, \ldots, v_u)$. We know that $f(x) = f^u(y^u)$, if $c^u = 1$. Furthermore, $f(x) = 0$, if $c^u = 0$ for all u. Hence,

$$f(x) = \bigvee_{1 \leq u \leq U} c^u \wedge f^u(y^u).$$

We compute $f^u(y^u)$ by its conjunctive normal form, i.e. in depth 2, from y^u and $\bar{y}^u$. Hence, $f^u(y^u)$ is computed on the third level by an $\wedge$-gate. Therefore, also $c^u \wedge f^u(y^u)$ can be computed on the third level. Finally, f is computed in depth 4.

We still have to estimate the size of the conjunctive normal forms. Since, $v_i = 0$ for $i > L$, $\binom{u}{L+1}$ clauses are sufficient to cover the inputs with more than L ones. If $v_i = 0$ and $i \leq L$, $\binom{u}{i}$ clauses are sufficient to cover the inputs with i ones. Since $\binom{U}{j} \leq U^j$, we can compute all $c^u \wedge f^u(y^u)$ from $y^u, \bar{y}^u$ and c^u with $O(U^{L+2}) = O(L^{2L+4} \log^{L+2} n)$ gates and $O(U^{L+3}) = O(L^{2L+6} \log^{L+3} n)$ wires. We have proved the following theorem.

Theorem 4: i) Symmetric functions f on n variables where $v_k(f) = 1$ only for some $k \leq L$ can be computed by an unbounded fan-in circuit of depth 4 with $O(L^{2L+4} \log^{L+2} n)$ gates and $O(L^{2L+6} \log^{L+3} n + n\ L^2 \log^2 n)$ wires.
ii) If $L = L(n) = O(\log^\delta n)$ for some $\delta < 1$, the number of gates is bounded by $2^{O(\log^\delta n \log\log n)}$ and the number of wires by $O(n \log^{2+2\delta} n)$.

We make some remarks. The number of gates (in part ii) of the theorem) is subpolynomial, i.e. $o(n^\alpha)$ for each $\alpha > 0$, but superpolylogarithmic. The upper bound is superpolynomial, if $L = \Omega(\log n)$. We leave it to the reader to discuss functions L where $L = o(\log n)$ but $L = \omega(\log^\delta n)$ for all $\delta < 1$. We also leave it to the reader to design circuits where Lemma 1 is applied more than once.

The threshold functions are of particular interest. Our circuit works for NT^n_{k-1} only with the negative variables $\bar{x}_i$. The conjunctive normal form for $f^u = NT^u_{k-1}$ consists only of the $\binom{n}{k}$ prime clauses containing only $n - k$ negative y^u-variables each. By Lemma 1, the NT_1-circuits work only with negative variables. By construction, $\bar{y}^u_j$ is the conjunction of some negative variables. Hence, we do not need the positive y^u- and x-variables. In order to compute $T^n_k = \neg NT^n_{k-1}$ we apply deMorgan's rules to this circuit and obtain a monotone circuit for T^n_k of the same size as stated in Theorem 4.

Up to now we do not have designed efficient circuits for all symmetric functions in AC^0. For the general case let us assume that $v_k = 1$ only for some $k \leq L^m$ where $L = O(\log^\delta n)$ for some $\delta < 1$ and m is a constant. For $U := \lfloor L^{2m} \log n \rfloor$ we compute as before all $y^u, \bar{y}^u$ and c^u with $O(U^2 \log n)$ gates and $O(nU \log n)$ wires. y^u and $\bar{y}^u$ are computed in depth 1 and c^u in depth 3.

Since v_k may equal 1 for $k = L^m$, the normal forms for f^u may have nonpolynomial size. We only can test y^u for up to L ones with normal forms of very small size. We do these computations for all subvectors of y^u. Afterwards we may test for L^2 ones by testing L pieces for L ones each. After m steps of this type we can test y^u for up to L^m ones. We explain these ideas now in more detail.

Let u be fixed and let $[i,j]$ denote the vector $(y_i^u, \ldots, y_j^u)$. We are interested in the following functions and their negations.

- $p_l[i,j]$, where $1 \leq l \leq m$ and $0 \leq i \leq j \leq u-1$, computes 1 iff $[i,j]$ contains exactly L^l ones.
- $q_l[i,j]$, where $1 \leq l \leq m$ and $0 \leq i \leq j \leq u-1$, computes 1 iff $[i,j]$ contains less than L^l ones.
- $r_l[i,a]$, where $1 \leq l \leq m$, $0 \leq i \leq u-1$ and $0 \leq a \leq L^l + 1$, computes 1 iff $[i,u-1]$ contains exactly a ones.
- $s_l[i,a]$, where $1 \leq l \leq m$, $0 \leq i \leq u-1$ and $0 \leq a \leq L^l + 1$, computes 1 iff $[i,u-1]$ contains less than a ones.
- $t[i,j]$, where $1 \leq i \leq j \leq u-1$, computes 1 iff $[i,j]$ contains no one.

The computation of t and $\bar{t}$ is simple and can be done on level 2.

The functions for $l = 1$ have simple conjunctive and disjunctive normal forms. Each of the $O(U^2)$ functions $p_1, \bar{p}_1, q_1, \bar{q}_1, r_1, \bar{r}_1, s_1$ and $\bar{s}_1$ has at most

$$\max\left\{\binom{U}{L}, \binom{U}{L-1} + \binom{U}{L+1}, \binom{U}{a}, \binom{U}{a-1} + \binom{U}{a+1}\right\} = O(U^{L+2})$$

prime implicants. All these functions are computed on level 3 by conjunctive normal forms with $O(U^{L+4})$ gates and $O(U^{L+5})$ wires.

We describe how we compute the functions $p_l, \bar{p}_l, \ldots, s_l, \bar{s}_l$ by disjunctive forms using the functions $p_{l-1}, \ldots, \bar{s}_{l-1}, t, \bar{t}$. Then we know by deMorgans's rules also conjunctive forms of the same complexity for these functions.

For even l we use the disjunctive forms and for odd l the conjunctive ones. Then the first level of stage l is of the same type as the second level of stage $l-1$ and these two levels can be merged.

$p_l[i,j] = 1$ iff there exists a partition of $[i,j]$ into L pieces each containing exactly L^{l-1} ones. This leads to a disjunctive form for p_l with at most $\binom{U}{L-1} + 1$ gates.

$\bar{p}_l[i,j] = 1$ iff $[i,j]$ contains less than L^l ones or more than L^l ones. This is equivalent to the following statement. There exists a partition of $[i,j]$ into $k \in \{1, \ldots, L\}$ pieces such that

the first $k-1$ pieces contain exactly L^{l-1} ones each and the last piece contains less than L^{l-1} ones or there exists a partition of $[i,j]$ into $L+1$ pieces such that first L pieces contain exactly L^{l-1} ones each and the last piece is not the constant 0-vector. Hence, $O(L\binom{U}{L+1})$ gates are sufficient.

$q_l[i,j] = 1$ iff there exists a partition of $[i,j]$ into $k \in \{1,\ldots,L\}$ pieces such that the first $k-1$ pieces contain L^{l-1} ones each and the last piece contains less than L^{l-1} ones. Hence, $O(L\binom{U}{L})$ gates are sufficient.

$\bar{q}_l[i,j] = 1$ iff there exists partition of $[i,j]$ into $L+1$ pieces such that the first L pieces contain L^{l-1} ones each. Hence, $O(\binom{U}{L+1})$ gates are sufficient.

The computation of the functions $r_l, \bar{r}_l, s_l$ and $\bar{s}_l$ can be performed in a similar way.

Finally, we like to compute the function f where $v_k(f) = 1$ only for some $k \leq L^m$. For even m the last level is an $\vee$-level and we compute f as the disjunction of all $r_m[0,k]$ where $v_k(f) = 1$. The negation of f is the disjunction of all $r_m[0,k]$ where $k \leq L^m$ and $v_k(f) = 0$ and the function $\bar{s}_m[0, L^m+1]$. Hence, in all cases it is possible to integrate the computation of f into the computation of the functions of stage m.

Up to now we have considered the computation of f on the input $(y^u, \bar{y}^u)$ under the assumption that $c^u = 1$. We still have to eliminate invalid computations. In any case we have negations only at the inputs. If the last level is an $\vee$-level, we feed all gates on level 3, an $\wedge$-level, with the appropriate c^u. Furthermore, we combine the outputs for the different u on the last level by a disjunction which does not increase the depth. Invalid computations feed 0 into this output disjunction. If all computations are invalid, we compute 0 which is also the correct result. If the last level is an $\wedge$-level, we feed all gates on level 4, an $\vee$-level with the appropriate $\bar{c}^u$. Furthermore, we combine the outputs for the different u on the last level by a conjunction. Invalid computations feed 1 into this output conjunction. In order to obtain the correct result also in the case where all u are invalid, we also feed the disjunction of all c^u into the output conjunction.

The depth of our circuit is $m+2$, if $m \geq 2$. The number of gates can be estimated by

$$O(L^{m+1}U^{L+4}) = 2^{O(\log^{\delta} n \log\log n)}.$$

The number of bits in all $(y^u, \bar{y}^u, c^u)$ is very small. Hence, the number of wires which enter the gates on the first two levels dominates the number of all other wires. Therefore, the number of wires can be estimated by $O(nU\log n) = O(n\log^{2m\delta+2} n)$.

We have proved the following theorem.

Theorem 5: Let $L = O(\log^{\delta} n)$ for some $\delta < 1$. Symmetric functions f on n variables, where $v_k(f) = 1$ only for some $k \leq L^m$ and $m \geq 2$ is a constant, can be computed by unbounded fan-in circuits of depth $m+2$ with $2^{O(\log^{\delta} n \log\log n)}$ gates and $O(n \log^{2m\delta+2} n)$ wires.

In order to compare our results with the results of the other papers we consider the special case of T_k^n and $k = \lfloor \log^m \rfloor$. Let $L := \log^{m/(m+1)} n$. Then $L^{m+1} = \log^m n$. We have proved the following theorem.

Theorem 6: The threshold functions T_k^n where $k = \lfloor \log^m n \rfloor$ for constant m can be computed by unbounded fan-in circuits of depth $m+3$ with $2^{O(\log^{m/(m+1)} n \log\log n)}$ gates and $O(n \log^{2m+2} n)$ wires.

In order to compare our results with the other papers we consider the special case of threshold functions. We design circuits for NT-functions working on the negative variables $\bar{x}_i$ only and apply deMorgan's rules to obtain monotone circuits for threshold functions. We consider NT_k^n. W.l.o.g. (reductions by projection) we assume that $k = L^h$ and that L is an integer.

We compute as before c^u (for all $u \leq U$) in depth 3. We also compute all $\bar{y}^u$ in depth 1. Now we are interested in the following functions.

- $p_l^*[i,j]$, where $1 \leq l \leq h$ and $0 \leq i \leq j \leq u-1$, computes 1 iff $[i,j]$ contains at most L^l ones.
- $q_l^*[i,j]$, where $1 \leq l \leq h$ and $0 \leq i \leq j \leq u-1$, computes 1 iff $[i,j]$ contains less than L^l ones.

The functions for $l = 1$ again have simple disjunctive normal forms. We describe how we compute the functions p_l^* and q_l^* by disjunctive forms (for odd l) and by conjunctive forms (for even l) using the functions p_{l-1}^* and q_{l-1}^*.

$p_l^*[i,j] = 1$ iff there exists a partition of $[i,j]$ into L pieces such that each piece contains at most L^{l-1} ones. And $p_l^*[i,j] = 1$ iff for all partitions of $[i,j]$ into L pieces the first piece contains at most L^{l-1} ones or some other piece contains less than L^{l-1} ones.

$q_l^*[i,j] = 1$ iff there exists a partition of $[i,j]$ into L pieces such that the first $L-1$ pieces contain at most L^{l-1} ones and the last piece contains less than L^{l-1} ones. And $q_l^*[i,j] = 1$ iff for all partitions of $[i,j]$ into L pieces there is some piece containing less than L^{l-1} ones.

In this way we compute correctly NT_k^n if y^u is a valid coding. If the last level is an $\vee$-level, we feed the $\wedge$-gates on level 4 by c^u and combine all outputs on the last level by

a disjunction. If the last level is an $\wedge$-level, we combine all outputs on the last level by a conjunction. If $NT_k^n(x) = 1$, the invalid codings cause no problem, since they are only underestimating the number of ones. If $NT_k^n(x) = 0$, invalid computations may feed ones into the output gate. Hence, we feed the disjunction of all c^u into the output gate.

For $k = \lfloor \log^m n \rfloor$ (m not necessarily an integer but a constant) we choose $L = \lfloor \log^{m/(\lfloor m \rfloor + 1)} n \rfloor$ and $h = \lfloor m \rfloor + 1$. (In order to be precise we have to use the appropriate reductions.) By the design above we have proved the following theorem.

Theorem 6: The threshold functions T_k^n where $k = \lfloor \log^m n \rfloor$ for constant m can be computed by monotone unbounded fan-in circuits of depth $\lfloor m \rfloor + 3$ with $2^{O(\log^{m/(\lfloor m \rfloor + 1)} n \log\log n)}$ gates and $O(n \log^{2m+2} n)$ wires.

The depth of our circuits differs, by Theorem 3, from the lower bound for polynomial-size circuits only by 2, if m is an integer, and by 1 else.

In order to minimize the depth we can even do better. The coded variables $\bar{y}^u$ are computed by $\wedge$-gates on the first level, and the functions p_1^* and q_1^* are computed by their disjunctive normal forms. Hence, the level of the computation of $\bar{y}^u$ and the first level of the computation of p_1^* and q_1^* can be merged. This increases the number of wires to $O(ng)$ where g is the number of gates. If $m \geq 2$, the total depth is decreased to $\lfloor m \rfloor + 2$. For $1 \leq m < 2$, we like to get by with depth 3 and have to feed the disjunction of all c^u into the output gate, an $\wedge$-gate, on depth 3. In this case we compute the NT_1-functions c^u by their disjunctive normal forms in depth 2. Also in depth 2 we can compute the disjunction of all c^u. In this case we have increased the number of gates to $O(nU)$ and the number of wires to $O(n^2 U)$.

Theorem 7: The threshold functions T_k^n where $k = \lfloor \log^m n \rfloor$ can be computed by monotone unbounded fan-in circuits of depth $\lfloor m \rfloor + 2$ with polynomial size. If $m \geq 2$, the number of gates is subpolynomial and the number of wires is only by a linear factor larger.

By Theorem 3 this depth is optimal, if m is not an integer. Theorem 7 holds also for $m < 1$, since in that case T_k^n has disjunctive normal forms of polynomial size.

References

Ajtai,M. and Ben-Or, M. (1984). A theorem on probabilistic constant depth computations. 16. Symp. on Theory of Computing, 471-474.

Boppana,R. (1984). Threshold functions and bounded depth monotone circuits. 16. Symp. on Theory of Computing, 475-479.

Brustmann,B. and Wegener,I. (1987). The complexity of symmetric functions in bounded-depth circuits. Information Processing Letters 25, 217-219.

Chandra, A., Fortune,S. and Lipton,R.J. (1983). Unbounded fan-in circuits and associative functions. 15. Symp. on Theory of Computing, 52-60.

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

Dolev,D., Dwork,C., Pippenger,N.J. and Wigderson,A. (1983). Superconcentrators, generalizers and generalized connectors with limited depth. 15. Symp. on Theory of Computing, 42-51.

Fagin,R., Klawe,M.M., Pippenger,N.J. and Stockmeyer,L. (1985). Bounded-depth, polynomial-size circuits for symmetric functions. Theoretical Computer Science 36, 239-250.

Friedman,J. (1984). Constructing $O(n \log n)$ size monotone formulae for the k-th elementary symmetric polynomial of n Boolean variables. 25. Symp. on Foundations of Computer Science, 506-515.

Hastad,J. (1986). Almost optimal lower bounds for small depth circuits. 18. Symp. on Theory of Computing, 6-20.

Hromkovic,J. (1985). Linear lower bounds on unbounded fan-in Boolean circuits. Information Processing Letters 21, 71-74.

Mayr,E. (1985). Fast selection on paracomputers. 11. Int. Workshop on Graphtheoretical Concepts in Computer Science, 249-254.

Moran,S. (1987). Generalized lower bounds derived from Hastad's main lemma. Information Processing Letters 25, 383-388.

Newman,I., Ragde,P. and Wigderson,A. (1990). Perfect hashing, graph entropy and circuit complexity. Proc. 5. Structure in Complexity Theory.

Wegener,I. (1990). The complexity of the parity function in unbounded fan-in, unbounded depth circuits. To appear: Theoretical Computer Science.