Optimal Bounds for Decision Problems on the CRCW PRAM

PAUL BEAME

University of Washington, Seattle, Washington

AND

JOHAN HASTAD

Royal Institute of Technology, Stockholm, Sweden

Abstract. Optimal $\Omega(\log n/\log\log n)$ lower bounds on the time for CRCW PRAMs with polynomially bounded numbers of processors or memory cells to compute parity and a number of related problems are proven. A strict time hierarchy of explicit Boolean functions of n bits on such machines that holds up to $O(\log n/\log\log n)$ time is also exhibited. That is, for every time bound T within this range a function is exhibited that can be easily computed using polynomial resources in time T but requires more than polynomial resources to be computed in time $T-1$. Finally, it is shown that almost all Boolean functions of n bits require $\log n - \log\log n + \Omega(1)$ time when the number of processors is at most polynomial in n. The bounds do not place restrictions on the uniformity of the algorithms nor on the instruction sets of the machines.

Categories and Subject Descriptors: F.1.2 **[Computation by Abstract Devices]**: Modes of Computation—*parallelism*; F.1.3 **[Computation by Abstract Devices]**: Complexity Classes—*complexity hierarchies, relations among complexity measures*; F.2.3 **[Analysis of Algorithms and Problem Complexity]**: *Trade-offs among complexity classes*

General Terms: Theory, Verification

Additional Key Words and Phrases: Concurrent-write, lower bounds, parallel random-access machines, parity, sorting

1. *Introduction*

One of the most widely used models of parallel computation is the parallel random access machine (PRAM). In this model any processor can access any memory location at a given time-step. The most powerful form of the PRAM, the CRCW PRAM, in which both concurrent read and concurrent write accesses are allowed, has received particular attention both from designers of algorithms and from those

The work of P. Beame was supported by a University of Toronto Open Fellowship and by National Science Foundation grant PYI-25800. The work of J. Hastad was supported by an IBM Postdoctoral Fellowship and supported in part by NSF grant DCR MCS-85-09905.

This research was done while P. Beame was at the University of Toronto and while both authors were at the Massachusetts Institute of Technology.

Authors' present addresses: P. Beame, Computer Science Department, FR-35, University of Washington, Seattle, Washington 98195; J. Hastad, Royal Institute of Technology, Stockholm, S-100-44, Sweden.

studying the limitations of parallel machine computation. Despite the significant interest, the only nontrivial lower bounds for decision problems on CRCW PRAMs that do not have drastic restrictions placed on either their processor and memory resources or on the instruction sets of their processors are due independently to Beame [3] and to Li and Yesha [13]. The lower bounds are for parity and related problems and are far from optimal. In both of these bounds no restriction is placed on the instruction set of the processors, no limitation is placed on how much information a single memory location may store, and the resources allowed are only polynomially bounded. We call a machine with these properties an *abstract* or *ideal PRAM.*

In this very general setting we prove the first optimal bound for any nontrivial decision problem on the CRCW PRAM by showing a time lower bound of $\Omega(\log n/\log\log n)$ for parity that matches the known upper bound. This lower bound holds even in the cases when only one of the two resources, processors or memory cells, is bounded by a polynomial in the input size. Because parity constant-depth reduces to a large number of problems, this $\Omega(\log n/\log\log n)$-time lower bound for the CRCW PRAM applies to a wide variety of interesting functions that include sorting or adding n bits, as well as multiplying two n-bit integers.

Also, by looking at the so-called "Sipser" functions, which are defined by circuits, we obtain a very sharp time hierarchy for CRCW PRAMs of polynomial-bounded resources. That is, for every time bound $T(n)$ at most $\log n/(3\log\log n) - \Theta(\log n/(\log\log n)^2$ we exhibit a family of functions which is computable in time bound T with n processors and memory cells, but which cannot be computed just one step faster by any machine with a polynomial bound on the number of processors even with no bound on the number of memory cells. A similar separation holds for machines with a polynomial bound on the number of memory cells even without a bound on the number of processors.

The proofs of both these results follow lines similar to the proofs in [2] and [3] and involve new lemmas that generalize the key lemmas used in Hastad's unbounded fan-in circuit lower bounds [10] and [11].

We also prove a tight $\Theta(\log n)$ lower bound on the time to compute almost all n-bit Boolean functions on CRCW PRAMs with polynomial numbers of processors.

A preliminary version of these results appeared in [5]. Many of these results also form a part of the first author's Ph.D. dissertation [4].

2. *History of the Problem*

Much of the lower bound work for CRCW PRAMs has been based on their close relationship to unbounded fan-in circuits. These were defined by Furst et al. [9] largely as a tool for trying to get an oracle to separate the polynomial-time hierarchy from PSPACE. Stockmeyer and Vishkin [15] showed that simple CRCW PRAMs can simulate unbounded fan-in circuits with essentially the same number of processors as the circuit size and the same time as the circuit depth. In fact, by restricting the instruction set of the CRCW PRAM to a limited set that includes addition, comparison, indirect addressing and a few related instructions, Stockmeyer and Vishkin also showed that unbounded fan-in circuits can easily simulate restricted CRCW PRAMs. The size of the resulting circuit is polynomial in the number of processors multiplied by the time and its depth is only a constant factor larger than the time. Using the latter result and a $\Omega(\log^* n)$ lower bound of Furst et al. [9] on the depth of polynomial size unbounded fan-in circuits computing parity,

Stockmeyer and Vishkin [15] obtained lower bounds for this restricted form of CRCW PRAM.

Because disjunctive normal-form formulas are unbounded fan-in circuits of depth two it follows that all Boolean functions may be computed in two steps using exponential resources on the CRCW PRAM. However, it is not reasonable to be using exponentially many processors and memory cells. With polynomial resource bounds, CRCW PRAMs can compute any function with formula size $n^{O(1)}$ in time $O(\log n/\log\log n)$, using an algorithm based on an upper bound of size $n^{O(1)}$ and depth $O(\log n/\log\log n)$ for unbounded fan-in circuits given by Chandra et al. [7].

Since Stockmeyer and Vishkin's paper, the lower bounds for unbounded fan-in circuits have been significantly improved. Ajtai, extending [1], and L. Babai (private communication) derived $\Omega(\sqrt{\log n})$ depth lower bounds for polynomial size circuits computing parity. Yao [16] markedly improved these results by showing truly exponential size lower bounds for circuits of constant depth but this improvement did not increase the depth lower bound beyond $\Omega(\sqrt{\log n})$. Finally, Hastad [10] using some techniques similar to those used by Yao, obtained an $\Omega(\log n/\log\log n)$ depth lower bound for polynomial-size circuits computing parity, which matches the bound from the algorithm of Chandra et al. However, the CRCW PRAM lower bounds that follow using Stockmeyer and Vishkin's simulation are still not entirely satisfactory since the bounds rely in an essential way on the specific restriction that is placed on the instruction set. Some operations that are prohibited in this model seem to be perfectly reasonable ones.

Abstract CRCW PRAMs can be shown to be much more powerful than these restricted machines; because of their equivalence with unbounded fan-in circuits, restricted CRCW PRAMs with polynomially many processors require exponential time to compute almost all Boolean functions whereas an abstract PRAM only takes $O(\log n)$ time without even using its power of concurrent reads or writes. Nevertheless, for certain specific functions we shall see that, by using direct techniques, lower bounds as strong as those derived for these restricted CRCW machines can be obtained for the most powerful model of CRCW PRAM.

By applying and modifying the techniques of [9], Beame [2] derived the first nontrivial lower bound that applies to the CRCW PRAM model described here. He showed that any CRCW PRAM computing the parity function with $n^{O(1)}$ memory cells and an unbounded number of processors requires time $\Omega(\sqrt{\log\log n})$. Later, using the main lemma in [10], Beame [3] obtained the following: any CRCW PRAM that computes the parity function with $n^{O(1)}$ processors (in fact with as many as $n^{2^{\delta\sqrt{\log n}}}$ processors for some $\delta > 0$) and unbounded memory requires time $\Omega(\sqrt{\log n})$. With the same techniques, an $\Omega(\sqrt{\log n})$ lower bound is easily shown for common-write CRCW PRAMs (for definitions, see Section 3) that have no bound on the number of processors but have a bound of $O(n^{2^{\delta\sqrt{\log n}}})$ on the number of cells for some $\delta > 0$.

It was shown by B. Chor (private communication) and Li and Yesha [13] that a simulation of abstract CRCW PRAMs by unbounded fan-in circuits can be combined directly with Hastad's circuit lower bound to obtain the $\Omega(\sqrt{\log n})$ lower bound. However, this simulation does not yield the above lower bound for the common-write model with an unbounded number of processors. The simulation states that any CRCW PRAM solving a decision problem on n Boolean inputs using $p(n)$ processors and $T(n)$ time can be simulated by an unbounded fan-in circuit of size $p(n)^{2^{T(n)+O(1)}}$ and depth $O(T(n))$.

Beame [3] and Li and Yesha [13] have also independently shown optimal bounds on the time needed by CRCW PRAMs to compute functions whose many-bit

output is required to appear in a single memory cell. However, as was noted in [3], such an output requirement is somewhat artificial and the lower bounds disappear if each bit of the output is allowed to appear in a separate memory cell.

3. *Definitions and Preliminaries*

Definition. A *CRCW PRAM* is a shared memory machine with processors $P_1, \ldots, P_{p(n)}$ which communicate through memory cells $C_1, \ldots, C_{c(n)}$. The values of the input variables $x_1, \ldots, x_n$ are initially stored in the first n cells of memory $C_1, \ldots, C_n$, respectively. Initially all cells other than the input cells contain the value 0. The output of the machine is the value in the cell C_1 at termination.

Before each step t, processor P_i is in state q_i^t. At time step t, depending on q_i^t, processor P_i reads some cell C_j of shared memory, then, depending on the contents, (C_j), and q_i^t, assumes a new state q_i^{t+1} and depending on this state, writes a value $v = v(q_i^{t+1})$ into some cell.

When several processors are attempting to write into a single cell at the same time step the one that succeeds will be the lowest numbered processor. (A CRCW PRAM is defined to be a *common-write* machine if, whenever several processors are attempting to write into the same cell at a given time step, they all try to write the same value.)

The CRCW PRAM defined above has been called the *PRIORITY CRCW PRAM* and is the most powerful version of CRCW PRAM normally considered. Thus lower bounds for this model will apply to any standard model of CRCW PRAM.

In studying the progress of CRCW PRAM computations, what is important is the set of inputs which lead to a given value in a memory cell or a given state of a processor at a particular time step. The computation then may be viewed as operating not on actual values so much as on the partitions associated with them.

Definition. Let M be a CRCW PRAM. For any processor P_i the *processor partition*, $P(M, i, t)$, of the input set at time step t is defined so that two inputs are in the same equivalence class of $P(M, i, t)$ if and only if they lead to the same state of processor P_i at the end of time step t.

For any cell C_j the *cell partition*, $C(M, j, t)$, of the input set at time t is defined so that two inputs are in the same equivalence class of $C(M, j, t)$ if and only if they lead to the same contents of cell C_j at the end of time step t.

At time 0, the cell partitions for the first n memory cells have exactly two equivalence classes, one consisting of those inputs for which the value of the variable in the cell is 0, the other consisting of those inputs for which the value of that variable is 1. Initially all other processor and cell partitions have only one equivalence class consisting of all the inputs.

We now look at a measure of the complexity of partitions that was used in [2] and [3] to prove lower bounds for CRCW PRAMs.

Definition. Let f be a Boolean function defined on a set $I \subseteq \{0, 1\}^n$. A Boolean formula F *represents* f on I if the inputs $x \in I$ satisfy F exactly when $f(x) = 1$. Let the *maximum clause length* of a DNF formula F be the maximum number of literals in any clause of F. The (*Boolean*) *degree* of f on I, $\delta(f)$, is the smallest maximum clause length of all disjunctive normal form (DNF) formulas representing f on I. We extend this definition to sets of functions $\mathcal{F}$ by letting $\delta(\mathcal{F}) = \max_{f \in \mathcal{F}} \delta(f)$.

The terminology of degree is derived from the standard way of writing a formula with the Boolean $\vee$ as addition and the Boolean $\wedge$ as multiplication and then viewing the resulting formula as a polynomial. This should not be confused with the degree of a polynomial in the finite field of two elements where the exclusive-OR rather than the $\vee$ is the appropriate additive operation.

In the notation of many lower bound proofs for monotone formulas, we could define the prime implicants and prime clauses of a Boolean function f. (Prime clauses are essentially prime implicants of $\bar{f}$.) These have been described as minterms and maxterms, respectively, in the notation used by Yao [16] or Hastad [10]. Observe that the degree of a function and the length of its longest minterm or maxterm may differ because its longest minterm may be longer than the longest clause in an optimal DNF formula representing it. Consider the function f defined by the DNF formula $x_1x_2x_3 + \bar{x}_1x_4x_5$. It has a minterm $x_2x_3x_4x_5$, which is larger than $\delta(f)$.

Definition. Let A be a partition of a set $I \subseteq \{0, 1\}^n$. Define the *degree* of A, $\delta(A)$, to be $\delta(\mathcal{F}_A)$ on I where $\mathcal{F}_A$ is the set of characteristic functions of the equivalence classes of A in I.

The major proof technique of the lower bounds for parity on unbounded fan-in circuits is the use of restrictions to set some of the input bits. Using restrictions permits a simplified description of the results of computations but does not drastically reduce the difficulty of the function being computed. The main idea behind using them is that, although apparently complex operations like the OR of n bits are computed in one step, by setting relatively few inputs to 0 or 1 the results of these operations are simple. In the case of the OR of n bits, setting a single input to 1 makes it trivial.

Definition. A *restriction* π *on* $K \subseteq \{1, \ldots, n\}$ is a function $\pi\colon K \to \{0, 1, *\}$ where:

$$\pi(i) = \begin{cases} 1 & \text{means } x_i \text{ is set to 1,} \\ 0 & \text{means } x_i \text{ is set to 0,} \\ * & \text{means } x_i \text{ is unset.} \end{cases}$$

We define the results of applying a restriction π to a partition, $A\restriction_\pi$, a function, $f\restriction_\pi$, a Boolean formula, $F\restriction_\pi$, a circuit, $C\restriction_\pi$, as well as sets of these objects, $\mathcal{F}\restriction_\pi$ etc., in the natural way. If σ and τ are restrictions, then $\sigma\tau$ is a restriction that is the result of applying σ first and then applying τ. For any $K \subseteq \{1, \ldots, n\}$ define $Proj[K]$ to be the set of restrictions that assign 0 or 1 exactly to the inputs in K.

Definition. If a circuit D is $C\restriction_\pi$ for some restriction π, then we say that C *contains* D and the gates of C that remain undetermined in D will be said to take on the value $*$ in C when π is applied.

In several places we need the following simple observation.

LEMMA 3.1. *Let A be a partition of a set $I \subseteq \{0, 1\}^n$. For every $K \subseteq \{1, \ldots, n\}$, there exists a restriction $\sigma \in Proj[K]$ such that $\delta(A) \leq |K| + \delta(A\restriction_\sigma)$.*

PROOF. For each $\sigma \in Proj[K]$ let $\mathcal{F}_\sigma$ be a set of DNF formulas that represent the characteristic functions of the equivalence classes in $A\restriction_\sigma$ and that have maximum clause length bounded by $\delta(A\restriction_\sigma)$. To each clause of every formula in $\mathcal{F}_\sigma$, append the conjunctive clause C_σ which is true exactly on those inputs in $\{0, 1\}^n$ that agree with σ, to obtain a set of formulas $\tilde{\mathcal{F}}_\sigma$. By construction, the

formulas in $\tilde{\mathcal{F}}_\sigma$ have maximum clause length bounded by $\delta(A\Gamma_\sigma) + |K|$. Clearly the set of inputs that satisfy any formula in $\tilde{\mathcal{F}}_\sigma$ is contained in a single equivalence class of $A\Gamma_\sigma$ and therefore is contained in a single equivalence class of A. Furthermore, every input in I satisfies some formula in $\tilde{\mathcal{F}}_\sigma$ for an appropriate $\sigma \in Proj[K]$. Thus each class in A is a union of sets of inputs that satisfy formulas in some $\tilde{\mathcal{F}}_\sigma$ and so can be represented by a DNF formula with maximum clause length bounded by

$$\max_{\sigma \in Proj[K]} \delta(A\Gamma_\sigma) + |K|.$$

The lemma follows by the definition of the degree of A. □

The hard part in showing that restrictions simplify the results of CRCW PRAM computations is naturally the very powerful concurrent write operation since the read operation is simply the interaction of individual processors with single cells. It will be useful to define an abstraction of this operation in order to be able to describe conveniently the actions of restrictions on the new cell partitions that result from the concurrent writes. It also will turn out that, in describing the effects of restrictions on the processor partitions, we use a special case of this abstraction.

Definition. We say that an input $x \in \{0, 1\}^n$ *satisfies* a Boolean function F: $\{0, 1\}^n \to \{0, 1\}$ if $F(x) = 1$. We say that x *falsifies* F if $F(x) = 0$.

Definition. A *graded set of Boolean functions* is a set $\mathcal{G}$ of Boolean functions such that each $F \in \mathcal{G}$ has an associated positive integer *grade*, $\gamma(F)$ (or has grade $= \infty$) and no two functions of a given grade are simultaneously satisfiable.

Definition. For any graded set of Boolean functions, $\mathcal{G}$, *the partition determined by* $\mathcal{G}$, $\langle \mathcal{G} \rangle$, on $\{0, 1\}^n$ is the partition such that $x, y \in \{0, 1\}^n$ are in the same equivalence class if and only if:

(a) x and y both satisfy some function $F \in \mathcal{G}$, and x and y both falsify all $F' \in \mathcal{G}$ with $\gamma(F') < \gamma(F)$, or
(b) x and y both falsify all functions $F \in \mathcal{G}$.

Let us check that this is an equivalence relation. The reflexivity and symmetry of the relation above are obvious. The transitivity is a simple consequence of the fact that the definition of a graded set of functions excludes the possibility that two functions of a given grade are simultaneously satisfiable. For technical reasons the following straightforward lemma is convenient.

LEMMA 3.2. *Let $\mathcal{G}$ be a graded set of Boolean functions. If π is a restriction, then $\langle \mathcal{G} \rangle \Gamma_\pi$ is the same partition as $\langle \mathcal{G}\Gamma_\pi \rangle$ on $\{0, 1\}^n\Gamma_\pi$.*

We note that in the obvious way the above definitions can be carried over easily to Boolean formulas that represent the Boolean functions described. Observe that if $\mathcal{F}$ represents $\mathcal{G}$ on $\{0, 1\}^n\Gamma_\pi$, then $\langle \mathcal{F} \rangle \Gamma_\pi = \langle \mathcal{G} \rangle \Gamma_\pi$. Also, the notion of degree applies to graded sets of Boolean functions since it is defined for sets of functions. It is easy to see that a graded set of Boolean functions $\mathcal{G}$ can be represented on $\{0, 1\}^n\Gamma_\pi$ by a graded set of DNF formulas $\mathcal{F}$, each with maximum clause length bounded by $\delta(\mathcal{G}\Gamma_\pi)$.

Definition. Let M be a CRCW PRAM. Define $\mathcal{G}(M, j, t)$ to be the graded set of Boolean functions as follows:

(i) For each positive integer i, the functions of grade i in $\mathcal{G}(M, j, t)$ are the characteristic functions of those equivalence classes in $P(M, i, t)$ on which P_i writes into cell C_j during time step t.
(ii) The functions of grade ∞ in $\mathcal{G}(M, j, t)$ are all the characteristic functions of the equivalence classes in $C(M, j, t-1)$.

LEMMA 3.3. *Let M be a CRCW PRAM. $\langle \mathcal{G}(M, j, t) \rangle$ is a refinement of $C(M, j, t)$ on $\{0, 1\}^n$.*

PROOF. The way in which a partition is determined by a graded set of functions imitates the priority write operation of the CRCW PRAM. Condition (b) in the definition of the partition determined by a graded set of function cannot occur here since every input satisfies the characteristic function of some equivalence class in $C(M, j, t-1)$. Condition (a) in this definition corresponds to one of two cases. Either the input causes processor P_i to write and P_i is guaranteed to succeed since no lower-numbered processor attempts to write, or no processor writes and thus the previous value in the cell remains (we view this as the cell writing its old value back to itself). Note that, if the processors always write everything they know along with their processor id when they write, the two considered partitions are equal. □

The general method we employ for showing lower bounds on CRCW PRAM computations for decision problems is as follows. We show that after certain restrictions (which set more inputs as time progresses) are applied to the inputs, the processor and cell partitions have only small degree relative to the degree required to solve the problems. In using restrictions to obtain our lower bounds, we must maintain a balance between the amount of simplification that a restriction achieves and the number of inputs it sets.

4. *Tight Lower Bounds for Parity*

THEOREM 4.1. *If M is a CRCW PRAM that computes the parity function in time $T = T(n)$, then for sufficiently large n*

(*a*) *the total hardware $h(n) = p(n) + c(n)$ must be at least $2^{\lfloor (1/24)n^{1/T} - 2 \rfloor}$,*
(*b*) *the number of processors $p(n)$ must be at least $2^{\lfloor (1/96)n^{1/T} - 2 \rfloor}$ even if the number of memory cells is infinite, and*
(*c*) *the number of memory cells $c(n)$ must be at least $2^{\lfloor (1/12)(n/T!)^{1/T} - 2 \rfloor}$ even if the number of processors is infinite.*

For the proofs of each of the parts of this theorem we define restrictions π_t for each step t of the computation such that after step t and after π_t is applied, the cell (and processor) partitions all have degree less than the number of unset variables. The lower bound follows since setting variables of parity just leaves a smaller parity function (or its negation) and any representation of parity in DNF has clauses that depend on *all* the unset variables.

In order to prove the existence of restrictions that satisfy these properties we need an appropriate probability space from which to choose restrictions. This distribution was introduced by Furst et al. [9] and has been used in several subsequent lower-bound proofs for unbounded fan-in circuits.

Definition. Let $K \subseteq \{1, \ldots, n\}$. Define R_p^K to be a probability space of restrictions on K where, for a random ρ chosen from R_p^K, independently for each $i \in K$, $\rho(i)$ is $*$ with probability p and $\rho(i)$ is 0 or 1 with equal probability $(1 - p)/2$.

The outline above is now carried out by proving two lemmas. The first tells us that many variables remain unset and the second tells us that the degrees of the partitions do not increase.

LEMMA 4.1. *Let $L \subseteq \{1, \ldots, n\}$ and $0 < p < 1$ such that $p(1 - p)|L|$ is at least m_0 for some absolute constant m_0. Choose ρ at random from R_p^L. The probability that ρ leaves at least $p|L|$ inputs unset is greater than $\frac{1}{3}$.*

PROOF. The number of unset inputs is given by the binomial distribution on $|L|$ with expected value $p|L|$. A version of the Demoivre–Laplace limit theorem, Bollobás [6, Theorem 6(ii), page 13] implies that if $p(1 - p)|L|$ grows with $|L|$ then, as $|L|$ increases, the probability that at least $p|L| + 1$ inputs are unset approaches $\frac{1}{2}$. Thus for some finite value m_0, if $p(1 - p)|L|$ is at least m_0 this probability will certainly exceed $\frac{1}{3}$. □

LEMMA 4.2. *Let M be a CRCW PRAM just prior to a read or write operation, all of whose processor and cell partitions have degree at most $r \geq 1$ with variables from $\{x_i\}_{i \in L}$. Let A be either an existing processor or cell partition of M or a new cell partition resulting from a concurrent write of M. Choose ρ at random from R_p^L. For $s > 0$ we have*

$$Pr[\delta(A\lceil_\rho) \geq s] < (6pr)^s.$$

Using Lemma 3.3 we obtain Lemma 4.2 from the following lemma, which is the key generalization of the main lemma of Hastad [10], by noting that if β satisfies $(\beta^{-1}x + 1)^r = 2$ then $\beta \leq xr/\ln 2 < 3xr/2$.

LEMMA 4.3. *Let $\mathcal{G}$ be a graded set of DNF formulas on inputs $\{x_i\}_{i \in L}$ with maximum clause length bounded by $r \geq 1$ where $L \subseteq \{1, \ldots, n\}$. Let F be an arbitrary function on $\{0, 1\}^n$. Let ρ be a random restriction chosen from R_p^L. Then, if $\langle \mathcal{G}\lceil_\rho \rangle$ is the partition determined by $\mathcal{G}\lceil_\rho$, for $s \geq 0$ we have*

$$Pr[\delta(\langle \mathcal{G}\lceil_\rho \rangle) \geq s \mid F\lceil_\rho = 0] \leq \beta^s,$$

where $\beta > 0$ satisfies

$$\left(\frac{4p}{\beta(1 + p)} + 1\right)^r = 2.$$

PROOF. We first note that we only need to consider finite graded sets of formulas (i.e., $|\mathcal{G}|$ is finite). This follows since there are only a finite number of different input strings and so only a finite number of ways in which some formula in $\mathcal{G}$ can be satisfied and all smaller ones falsified. Also, it is trivial to see that the lemma holds for $s = 0$ or $\beta \geq 1$ so we can assume that $s > 0$ and $\beta < 1$.

The rest of the proof proceeds by induction on the total number of clauses in the formulas in $\mathcal{G}$. The intuitive idea is that as we work along the clauses one by one: if ρ falsifies a particular clause, then we are left with essentially the same problem as before; if ρ does not falsify the clause then, given the fact that it does not, it is much more likely that ρ satisfies the clause (and thus ensures that the remaining partition has only one class) than ρ leaves any input in the clause unset.

In this proof for readability we write $\delta\langle \mathcal{G} \rangle$ instead of $\delta(\langle \mathcal{G} \rangle)$.

Base Case. There are no clauses in the formulas in $\mathscr{F}$. In this case the formulas are all identically 0 and so all inputs are equivalent with respect to $\mathscr{F}$. Thus the partition determined by $\mathscr{F}\lceil_\rho$ consists of a single class so $\delta\langle\mathscr{F}\lceil_\rho\rangle = 0$ and the lemma holds for $\mathscr{F}$.

Induction Step. Assume that the lemma holds for all graded sets of formulas $\mathscr{F}'$ with fewer clauses than the formula of $\mathscr{F}$. Let F_1 be a formula in $\mathscr{F}$ that has lowest grade among those formulas in $\mathscr{F}$ that are not identically 0; let C_1 be a clause of F_1. We can analyze the probability by considering separately the cases in which ρ does or does not force clause C_1 to be 0. The failure probability, the probability that $\delta\langle\mathscr{F}\lceil_\rho\rangle \geq s$, is an average of the failure probabilities in these two cases. Thus

$$\Pr[\delta\langle\mathscr{F}\lceil_\rho\rangle \geq s \mid F\lceil_\rho = 0] \leq \max(\Pr[\delta\langle\mathscr{F}\lceil_\rho\rangle \geq s \mid F\lceil_\rho = 0 \wedge C_1\lceil_\rho = 0], \Pr[\delta\langle\mathscr{F}\lceil_\rho\rangle \geq s \mid F\lceil_\rho = 0 \wedge C_1\lceil_\rho \neq 0]).$$

The first term in the maximum is $\Pr[\delta\langle\mathscr{F}\lceil_\rho\rangle \geq s \mid (F \vee C_1)\lceil_\rho = 0]$. Let $\tilde{F}_1$ be F_1 with clause C_1 removed; thus $F_1 = C_1 \vee \tilde{F}_1$ and $\tilde{F}_1 \neq F_1$. Let $\tilde{\mathscr{F}}$ be the same as $\mathscr{F}$ with formula F_1 replaced by $\tilde{F}_1$. In this case $C_1\lceil_\rho = 0$ so $F_1\lceil_\rho = \tilde{F}_1\lceil_\rho$ and thus $\langle\mathscr{F}\lceil_\rho\rangle = \langle\tilde{\mathscr{F}}\lceil_\rho\rangle$. In other words, when $C_1\lceil_\rho = 0$, the lemma requires a bound on $\Pr[\delta\langle\tilde{\mathscr{F}}\lceil_\rho\rangle \geq s \mid (F \vee C_1)\lceil_\rho = 0]$. Since $\tilde{\mathscr{F}}$ has one fewer clause than $\mathscr{F}$ does, the inductive hypothesis implies that this probability is at most β^s.

The estimation of the second term in the maximum is more difficult. Let $T \subseteq L$ be the set of variables appearing in clause C_1. By hypothesis $|T| \leq r$. Let ρ_T be the restriction of ρ to the variables in T. The condition that $C_1\lceil_\rho \neq 0$ is equivalent to the condition that $C_1\lceil_{\rho_T} \neq 0$. We analyze the cases based on the subset Y of the variables in T to which ρ_T assigns $*$; we use the notation $*(\rho_T) = Y$ to denote the event that the variables in T which are assigned $*$ by ρ_T are exactly those in Y. Then

$$\Pr[\delta\langle\mathscr{F}\lceil_\rho\rangle \geq s \mid F\lceil_\rho = 0 \wedge C_1\lceil_{\rho_T} \neq 0] = \sum_{Y\subseteq T} \Pr[\delta\langle\mathscr{F}\lceil_\rho\rangle \geq s \wedge *(\rho_T) = Y \mid F\lceil_\rho = 0 \wedge C_1\lceil_{\rho_T} \neq 0]. \quad (1)$$

Consider the case in which $Y = \phi$. Then ρ_T sets every variable in T so the value of C_1 is forced by ρ_T. But since we already know that $C_1\lceil_{\rho_T} \neq 0$ we must have $C_1\lceil_{\rho_T} = 1$. In this case every input satisfies $F_1\lceil_\rho$ and since F_1 has lowest grade we know that all inputs are equivalent with respect to the $\langle\mathscr{F}\lceil_\rho\rangle$. It follows that $\delta\langle\mathscr{F}\lceil_\rho\rangle = 0$ so the term corresponding to $Y = \phi$ has probability 0. The sum in (1) then becomes

$$\begin{aligned}\Pr[\delta\langle\mathscr{F}\lceil_\rho\rangle \geq s \mid F\lceil_\rho = 0 \wedge C_1\lceil_{\rho_T} \neq 0] &= \sum_{Y\subseteq T, Y\neq\phi} \Pr[\delta\langle\mathscr{F}\lceil_\rho\rangle \geq s \wedge *(\rho_T) = Y \mid F\lceil_\rho = 0 \wedge C_1\lceil_{\rho_T} \neq 0] \\ &= \sum_{Y\subseteq T, Y\neq\phi} \{\Pr[\delta\langle\mathscr{F}\lceil_\rho\rangle \geq s \mid F\lceil_\rho = 0 \wedge C_1\lceil_{\rho_T} \neq 0 \wedge *(\rho_T) = Y] \\ &\qquad \times \Pr[*(\rho_T) = Y \mid F\lceil_\rho = 0 \wedge C_1\lceil_{\rho_T} \neq 0]\} \quad (2)\end{aligned}$$

by simple conditional probability.

We tackle the latter term in each of these products first. If we let $\rho_T(Y) = *$ denote the event that every variable in Y is unset by ρ_T then elementary probability yields

$$\Pr[*(\rho_T) = Y \mid F\lceil_\rho = 0 \wedge C_1\lceil_{\rho_T} \neq 0] \leq \Pr[\rho_T(Y) = * \mid F\lceil_\rho = 0 \wedge C_1\lceil_{\rho_T} \neq 0].$$

Then following Hastad [10, Lemma 3, page 12], we have

$$\Pr[\rho_T(Y) = * \mid F\lceil_\rho = 0 \wedge C_1\lceil_{\rho_T} \neq 0] \leq \left(\frac{2p}{1+p}\right)^{|Y|}.$$

Now we look at the first term in each product in (2). The condition that $C_1\lceil_{\rho_T} \neq 0 \wedge *(\rho_T) = Y$ exactly specifies ρ_T (which is $\rho\lceil_T$) since it means that every variable in $T \backslash Y$ is set to 0 or 1 in the way that does not force the value of C_1 to 0 and that every variable in Y is set to $*$. We let F' be $F \vee G$ where $G\lceil_\rho = 0$ if and only if ρ sets the variables in $T \backslash Y$ in the unique way that does not force clause C_1 to 0. Thus

$$\begin{aligned}\Pr[\delta\langle \mathscr{G}\lceil_\rho\rangle \geq s \mid F\lceil_\rho = 0 \wedge C_1\lceil_{\rho_T} \neq 0 \wedge *(\rho_T) = Y] \\ = \Pr[\delta\langle \mathscr{G}\lceil_\rho\rangle \geq s \mid F'\lceil_\rho = 0 \wedge *(\rho_T) = Y].\end{aligned}$$

Now, the condition $*(\rho_T) = Y$ means that the variables in Y are unset by ρ and that the variables in $T \backslash Y$ are all set by ρ. The latter part of this condition is implied by the condition $F'\lceil_\rho = 0$. Thus we do not change the events by rewriting the probability as

$$\Pr[\delta\langle \mathscr{G}\lceil_\rho\rangle \geq s \mid F'\lceil_\rho = 0 \wedge *(\rho_Y) = Y],$$

where ρ_Y is ρ restricted to the variables in Y. The condition $*(\rho_Y) = Y$ means that every variable in Y is unset by ρ.

If $|Y| \leq s$, then, by Lemmas 3.1 and 3.2,

$$\begin{aligned}&\Pr[\delta\langle \mathscr{G}\lceil_\rho\rangle \geq s \mid F'\lceil_\rho = 0 \wedge *(\rho_Y) = Y] \\ &\quad \leq \Pr[\exists \sigma \in Proj[Y], \delta\langle(\mathscr{G}\lceil_\sigma)\lceil_\rho\rangle \geq s - |Y| \mid F'\lceil_\rho = 0 \wedge *(\rho_Y) = Y] \\ &\quad \leq \sum_{\sigma \in Proj[Y]} \Pr[\delta\langle(\mathscr{G}\lceil_\sigma)\lceil_\rho\rangle \geq s - |Y| \mid F'\lceil_\rho = 0 \wedge *(\rho_Y) = Y] \\ &\quad = \sum_{\sigma \in Proj[Y]} \Pr[\delta\langle(\mathscr{G}\lceil_\sigma)\lceil_{\rho'}\rangle \geq s - |Y| \mid F'\lceil_{\rho'} = 0 \wedge *(\rho_Y) = Y], \qquad (3)\end{aligned}$$

where ρ' is the restriction of ρ to set $L' = L \backslash Y$. This last equality holds because ρ' sets exactly the same inputs that ρ does.

Because the probabilities on L' are independent of those on Y, the condition on ρ_Y does not affect the probabilities for ρ' so it can be eliminated without changing the probabilities in (3). Furthermore, because the probabilities on L' for ρ chosen at random from R_p^L are the same as those for a ρ' chosen from $R_p^{L'}$, the sum in (3) is equivalent to

$$\sum_{\sigma \in Proj[Y]} \Pr[\delta\langle(\mathscr{G}\lceil_\sigma)\lceil_{\rho'}\rangle \geq s - |Y| \mid F'\lceil_{\rho'} = 0], \qquad (4)$$

where ρ' is a restriction chosen at random from $R_q^{L'}$.

Since σ sets all inputs in Y and $F'\lceil_{\rho'} = 0$ we know that $\sigma\rho'$ sets all the inputs in T and thus forces the value of C_1. If $C_1\lceil_{\sigma\rho'} = 1$, then all inputs in $\langle(\mathscr{G}\lceil_\sigma)\lceil_{\rho'}\rangle$ are equivalent and thus $\delta\langle(\mathscr{G}\lceil_\sigma)\lceil_\rho\rangle = 0 \leq s - |Y|$. Otherwise $C_1\lceil_{\sigma\rho'} = 0$ and then $\langle(\mathscr{G}\lceil_\sigma)\lceil_{\rho'}\rangle = \langle(\tilde{\mathscr{G}}\lceil_\sigma)\lceil_{\rho'}\rangle$ since $\tilde{F}_1\lceil_{\sigma\rho'} = F_1\lceil_{\sigma\rho'}$. Thus the sum in (4) is equivalent to

$$\sum_{\sigma \in Proj[Y]} \Pr[\delta\langle(\tilde{\mathscr{G}}\lceil_\sigma)\lceil_{\rho'}\rangle \geq s - |Y| \mid F'\lceil_{\rho'} = 0].$$

Because $\tilde{\mathscr{G}}\lceil_\sigma$ has strictly fewer clauses than $\mathscr{G}$ and because it only has input variables from L' we can apply the inductive hypothesis to bound the probabilities in each

term in this sum by $\beta^{s-|Y|}$. For each Y the number of terms in the above sum is at most $|Proj[Y]| = 2^{|Y|}$ so we obtain a total bound of $2^{|Y|}\beta^{s-|Y|}$.

If $|Y| > s$, then we simply make the pessimistic assumption of failure, that is, that the degree of the resulting partition is too large. Since $\beta < 1$ and $s - |Y| < 0$ we certainly have $1 < 2^{|Y|}\beta^{s-|Y|}$. Thus

$$\Pr[\delta\langle\mathscr{G}\Gamma_\rho\rangle \geq s \mid F\Gamma_\rho = 0 \wedge C_1\Gamma_{\rho_T} \neq 0 \wedge *(\rho_T) = Y]$$

is at most $2^{|Y|}\beta^{s-|Y|}$.

Finally, substituting these bounds in (2) we obtain a total failure probability of at most

$$\begin{aligned}
&\sum_{Y\subseteq T, Y\neq\phi} \left(\frac{2p}{1+p}\right)^{|Y|} 2^{|Y|}\beta^{s-|Y|} \\
&\quad = \beta^s \sum_{i=1}^{|T|} \binom{|T|}{i} \left[\frac{4p}{\beta(1+p)}\right]^i \\
&\quad = \beta^s\left[\left(\frac{4p}{\beta(1+p)} + 1\right)^{|T|} - 1\right] \\
&\quad \leq \beta^s\left[\left(\frac{4p}{\beta(1+p)} + 1\right)^{r} - 1\right] \\
&\quad = \beta^s
\end{aligned}$$

using the definition of β. Thus the lemma holds for $\mathscr{G}$ and by induction we have proved the lemma. □

PROOF OF THEOREM 4.1. Let π_0 leave every input unset. We define restrictions $\pi_1, \pi_2, \ldots$ so that $\pi_{t+1} = \pi_t\rho_{t+1}$ and ρ_{t+1} is a restriction defined on K_t where K_t is the set of inputs unset by π_t.

Part (a). Recall that we wish to show that any CRCW PRAM computing parity in T steps requires total hardware, $h(n)$, at least $2^{[(1/24)n^{1/T}-2]}$.

CLAIM. *Let $s = \log 4h(n)$. For $t \geq 1$ we can choose π_t so that*

$$|K_t| \geq \frac{1}{12}\, n(24s)^{-(t-1)},$$

$$\max_i\, \delta(P(M, i, t)\Gamma_{\pi_t}) \leq s,$$

$$\max_j\, \delta(C(M, j, t)\Gamma_{\pi_t}) \leq s.$$

First we see how this claim implies the desired result. In order to compute parity in T steps, the degree of the partition in the first cell must be equal to the number of unset bits, that is, $\delta(C(M, 0, T)\Gamma_{\pi_T}) = |K_T|$. Then the claim implies that $s \geq (1/12)n(24s)^{-(T-1)}$ or equivalently $(24s)^T \geq 2n \geq n$. Solving this for s and substituting $s = \log 4h(n)$ yields $24 \log 4h(n) \geq n^{1/T}$ or $h(n) \geq 2^{(1/24)n^{1/T}-2}$.

We now show the claim by induction on t:

Base Case. At time 0 the processor partitions all consist of a single class and for each cell C_j, $C(M, j, 0)$ is a partition that depends on at most one input bit so $\delta(C(M, j, 0)) \leq 1$. After the read in step 1, each processor P_i reads one memory cell so the new state of the processor depends only on one input bit and

$\delta(P(M, i, 1)) \leq 1$. Let $q = \frac{1}{12}$. By Lemma 4.2 if we choose a random ρ from $R_q^{K_0}$, then

$$\Pr[\delta(C(M, j, 1)\lceil_\rho) \geq s] < (6q)^s = 2^{-s} = 2^{-\log 4h(n)} = \frac{1}{4h(n)}.$$

Since there are $c(n)$ memory cells, the probability that there exists a cell whose partition has degree larger than s, after ρ is applied, is bounded above by $c(n)/4h(n) \leq \frac{1}{4}$. Thus the probability that $\max_j \delta(C(M, j, 1)\lceil_\rho) \leq s$ is at least $\frac{3}{4}$. Since $s \geq 1$ we already know that $\max_i \delta(P(M, i, 1)) \leq s$ even before π_1 is applied. Also, by Lemma 4.1, for n sufficiently large, the number of variables in K_0 left unset by ρ is at least the expected value of $n/12$ with probability at least $\frac{1}{3}$. Thus the probability that ρ satisfies all these conditions is strictly positive and we can let $\pi_1 = \rho_1$ be one of the restrictions for which all the conditions hold. The base case follows.

Induction Step. Let $t \geq 1$. Assume the claim holds for t. We shall show that it holds for $t + 1$. During the $(t + 1)$st step of the machine, each processor first reads some cell based on its current state and based on the value read it changes to a new state. Thus, for each i, the cell C_j which processor P_i reads depends only on the equivalence class in $P(M, i, t)$ containing the input. Also, this equivalence class and the equivalence class in $C(M, j, t)$ containing the input determines the new state of the processor. Therefore each equivalence class in $P(M, i, t + 1)$ is an intersection of an equivalence class in $P(M, i, t)$ and one in $C(M, j, t)$ for some j. Then

$$\begin{aligned}\delta(P(M, i, t + 1)\lceil_{\pi_t}) &\leq \delta(P(M, i, t)\lceil_{\pi_t}) + \max_j \delta(C(M, j, t)\lceil_{\pi_t}) \\ &\leq s + s = 2s. \end{aligned} \tag{5}$$

We now must choose a restriction ρ_{t+1} that reduces this upper bound by half and handles the new cell partitions resulting from the write phase of the $(t + 1)$st step while not setting too many inputs. We show that such a ρ_{t+1} exists by choosing ρ at random from $R_q^{K_t}$ where $q = 1/(24s)$ and proving that the probability that such a ρ fails to have these properties is strictly less than 1.

Consider a particular memory cell C_j. By Lemma 4.2, if we choose a ρ at random from $R_q^{K_t}$ with $q = 1/(24s)$, we have

$$\begin{aligned}\Pr[\delta((C(M, j, t + 1)\lceil_{\pi_t})\lceil_\rho) \geq s] &< (12qs)^s \\ &= 2^{-s} = 2^{-\log 4h(n)} = \frac{1}{4h(n)}.\end{aligned}$$

Therefore we see that for a ρ chosen at random from $R_q^{K_t}$,

$$\Pr\left[\max_j \delta(C(M, j, t + 1)\lceil_{\pi_t\rho}) \geq s\right] < \frac{c(n)}{4h(n)}. \tag{6}$$

For each processor P_i we already know that $\delta(P(M, i, t + 1)\lceil_{\pi_t}) \leq 2s$. Since $P(M, i, t + 1)\lceil_{\pi_t}$ depends only on the inputs in K_t, by Lemma 4.2 we have

$$\begin{aligned}\Pr[\delta((P(M, i, t + 1)\lceil_{\pi_t})\lceil_\rho) \geq s] &< (12qs)^s \\ &= 2^{-s} = 2^{-\log 4h(n)} = \frac{1}{4h(n)}.\end{aligned}$$

Taking the maximum over all processors,

$$\Pr\left[\max_i \delta(P(M, i, t+1)\Gamma_{\pi_t\rho}) \geq s\right] < \frac{p(n)}{4h(n)}. \tag{7}$$

Therefore, putting (6) and (7) together, we see that with probability at least $\frac{3}{4}$, $\max_i \delta(P(M, i, t+1)\Gamma_{\pi_t\rho}) \leq s$ and $\max_j \delta(C(M, j, t+1)\Gamma_{\pi_t\rho}) \leq s$. Since we only apply this argument with $|K_t| q > s \geq \log 4n$ and since $1 - q > \frac{1}{2}$, by Lemma 4.1, with probability at least $\frac{1}{3}$, for sufficiently large n, the number of variables in K_t left unset by ρ is at least the expected value of $|K_t|q = |K_t|(24s)^{-1}$, which is at least $\frac{1}{12}n(24s)^{-t}$ by the inductive hypothesis. Thus the total failure probability is strictly less than 1 and we can let ρ_{t+1} be one of the restrictions for which all the conditions hold and the claim follows for $t+1$. By induction the claim for part (a) is proved. □

Part (b). We want to show that any CRCW PRAM computing parity in T steps requires a number of processors, $p(n)$, at least $2^{[(1/96)n^{1/T}-2]}$.

CLAIM. *Let* $s = \log 4p(n)$. *For* $t \geq 1$, *we can choose* π_t *so that*

$$|K_t| \geq \frac{1}{48} n(96s)^{-(t-1)},$$

$$\max_i \delta(P(M, i, t)\Gamma_{\pi_t}) \leq s,$$

$$\max_j \delta(C(M, j, t)\Gamma_{\pi_t}) \leq s.$$

First, we see how this claim implies the desired result. As in part (a) in order to compute parity in T steps, it is necessary that $\delta(C(M, 0, T)\Gamma_{\pi_T}) = |K_T|$. Then the claim implies that $s \geq \frac{1}{48}n(96s)^{-(T-1)}$ or equivalently $(96s)^T \geq 2n \geq n$. Solving this for s and substituting $s = \log 4p(n)$ yields $p(n) \geq 2^{(1/96)n^{1/T}-2}$.

We now show the claim by induction on t:

Base Case. As in the base case in part (a) we have $\max_j \delta(C(M, j, 0)) = 1$ and $\max_i \delta(P(M, i, 1)) \leq 1 \leq s$ so we only have to bound the degrees of the new cell partitions. Also, letting $q = \frac{1}{48}$, choosing ρ at random from $R_q^{K_0}$, and using a similar argument to the base case in part (a), we have for each j,

$$\Pr[\delta(C(M, j, 1)\Gamma_\rho) \geq s] < (6q)^s = 8^{-s} = 2^{-2s-s} = \frac{2^{-2s-2}}{p(n)}.$$

However, since each processor knows only one input bit, there are only two different cells that each processor could possibly write into on any input. So, at most $2p(n)$ cells could ever have been written into after one step. Thus the probability that $\max_j \delta((C(M, j, 1)\Gamma_\rho) \leq s$ is very close to 1. Also, by Lemma 4.1, for n sufficiently large, ρ leaves at least the expected $n/48$ inputs unset with probability at least $\frac{1}{3}$. Thus the probability that ρ satisfies all the conditions is strictly positive and we can let $\pi_1 = \rho_1$ be one of the restrictions for which all the conditions hold. The base case follows.

Induction Step. Let $t \geq 1$. Assume the claim holds for t. We shall show that it holds for $t+1$. Since the actual number of cells has no effect on the degrees of the

partitions resulting from reads and state transitions, as in part (a):

$$\delta(P(M, i, t+1)\Gamma_{\pi_t}) \leq \delta(P(M, i, t)\Gamma_{\pi_t}) + \max_j \delta(C(M, j, t)\Gamma_{\pi_t}) \leq s + s = 2s. \tag{8}$$

Again we must find a restriction ρ_{t+1} that keeps the degrees of the processor and cell partitions small but does not set too many bits. As in part (a), we show that such a ρ_{t+1} exists by choosing ρ at random from $R_q^{K_t}$ for appropriate q and prove that the probability that such a ρ fails to have these properties is strictly less than 1. The added complication is that we do not have an a priori bound on the number of memory cells for which ρ has to keep $\delta(C(M, j, t+1)\Gamma_{\pi_t\rho}) \leq s$. The reason why this does not hurt us is that, by the inductive hypothesis, any memory cell C_j that is not written into on any input in $\{0, 1\}^n\Gamma_{\pi_t}$ already satisfies $\delta(C(M, j, t+1)\Gamma_{\pi_t}) \leq s$.

For each memory cell C_j that is written into by some processor on an input in $\{0, 1\}^n\Gamma_{\pi_t}$, using the same reasoning as in part (a), we have

$$\Pr[\delta(C(M, j, t+1)\Gamma_{\pi_t\rho}) \geq s] < (12qs)^s. \tag{9}$$

Also as in part (a), for each processor P_i,

$$\Pr[\delta(P(M, i, t+1)\Gamma_{\pi_t\rho}) \geq s] < (12qs)^s. \tag{10}$$

Equation (8) implies that, for inputs in $\{0, 1\}^n\Gamma_{\pi_t}$, the classes in the new state partition of each processor have characteristic functions represented by DNF formulas with maximum clause length bounded by $2s$. Since a DNF clause of length $\leq 2s$ is satisfied by a fraction of at least $1/2^{2s}$ of the possible inputs, each class in the partition $P(M, i, t+1)\Gamma_{\pi_t}$ consists of a fraction of at least $1/2^{2s}$ of the possible inputs. This means that, for inputs in $\{0, 1\}^n\Gamma_{\pi_t}$, each processor can only be in one of 2^{2s} states and therefore can write into at most 2^{2s} different cells. Therefore the total number of cells for which ρ must work is at most $2^{2s}p(n)$.

Let $q = 1/(96s)$. The argument above means that (9) must be applied in at most $2^{2s}p(n)$ places and (10) must be applied in $p(n)$ places. Thus the total probability that either $\max_i \delta(P(M, i, t+1)\Gamma_{\pi_t\rho}) \geq s$ or $\max_j \delta(C(M, j, t+1)\Gamma_{\pi_t\rho}) \geq s$ is bounded by

$$\begin{aligned}(2^{2s}+1)p(n)(12qs)^s &= (2^{2s}+1)p(n)\left(\frac{1}{8}\right)^s \\ &= (2^{2s}+1)2^{-2s}p(n)2^{-s} \\ &= (1+2^{-2s})p(n)2^{-\log 4p(n)} = \frac{1}{4}(1+2^{-2s}).\end{aligned}$$

Also by Lemma 4.1, using the same reasoning as in part (a), with probability at least $\frac{1}{3}$, for sufficiently large n, the number of variables in K_t left unset by ρ is at least the expected value of $|K_t|q = |K_t|(96s)^{-1}$, which is $\geq \frac{1}{48}n(96s)^{-t}$ by the inductive hypothesis. Thus the total failure probability is strictly less than 1 and we can let ρ_{t+1} be one of the restrictions for which all the conditions hold and the claim follows for $t+1$. By induction the claim for part (b) is proved. □

Part (c). We want to show that any CRCW PRAM computing parity in T steps requires a number of memory cells, $c(n)$, at least $2^{[(1/12)(n/T!)^{1/T}-2]}$.

CLAIM. *Let* $s = \log 4c(n)$. *For* $t \geq 1$ *we can choose* π_t *so that*

$$|K_t| \geq \frac{(1/12)n(12s)^{-(t-1)}}{t!},$$

$$\max_i \delta(P(M, i, t)\lceil_{\pi_t}) \leq st,$$

$$\max_j \delta(C(M, j, t)\lceil_{\pi_t}) \leq s.$$

First we see how this claim implies the desired result. As before, in order to compute parity in T steps, it is necessary that $\delta(C(M, 0, T)\lceil_{\pi_T}) = |K_T|$. Then the claim implies that $s \geq \frac{1}{12}n(12s)^{-(T-1)}/T!$ or equivalently $(12s)^T \geq n/T!$. Solving this for s and substituting $s = \log 4c(n)$ yields $c(n) \geq 2^{(1/12)(n/T!)^{1/T}-2}$.

We now show the claim by induction on t:

Base Case. As in the base case in part (a), $|K_0| = n$ and

$$\delta(P(M, i, 0)) = 0,$$

$$\delta(C(M, j, 0)) \leq 1,$$

$$\delta(P(M, j, 1)) \leq 1 < s.$$

Also, letting $q = \frac{1}{12}$, choosing ρ at random from $R_q^{K_0}$, just as in the base case in part (a), we have for each j,

$$\Pr[\delta(C(M, j, 1)\lceil_\rho) \geq s] < (6q)^s = 2^{-s} = 2^{-\log 4c(n)} = \frac{1}{4c(n)}.$$

Since there are $c(n)$ cells for which ρ must work the probability that $\max_j \delta((C(M, j, 1)\lceil_\rho) \leq s$ is at least $\frac{3}{4}$. Also, applying Lemma 4.1, for sufficiently large n, ρ leaves at least the expected number of $n/12$ inputs unset with probability at least $\frac{1}{3}$. Thus the probability that ρ satisfies all the conditions is strictly positive and we can let $\pi_1 = \rho_1$ be one of the restrictions for which all the conditions hold. The base case follows.

Induction Step. Let $t \geq 1$. Assume the claim holds for t. We shall show that it holds for $t + 1$. By the same reasoning as that leading to eq. (5) it is clear that the new processor partitions resulting from reads and state transitions satisfy:

$$\begin{aligned}\delta(P(M, i, t+1)\lceil_{\pi_t}) &\leq \delta(P(M, i, t)\lceil_{\pi_t}) + \max_j \delta(C(M, j, t)\lceil_{\pi_t}) \\ &\leq st + s = s(t+1). \end{aligned} \tag{11}$$

Thus, even before ρ_{t+1} is applied, the processor partitions satisfy the conditions required.

Now we must find a restriction ρ_{t+1} that keeps the degrees of the cell partitions small but does not set too many bits. As before, we show that such a ρ_{t+1} exists by choosing ρ at random from $R_q^{K_t}$ for appropriate q and prove that the probability that such a ρ fails to have these properties is strictly less than 1. This time we will have to make q depend on t since the bound on the degrees of the processor partitions is dependent on t. In particular we let $q = 1/(12s(t+1))$.

For each memory cell C_j, since the new processor partitions have degree at most $s(t+1)$ by (11) and since the old cell partitions have degree at most s, using the

same reasoning as in the previous two cases, we have

$$\Pr[\delta(C(M, j, t+1)\lceil_{\pi_t\rho}) \geq s] < (6qs(t+1))^s$$
$$= 2^{-s} = 2^{-\log 4c(n)} = \frac{1}{4c(n)}. \tag{12}$$

Since there are $c(n)$ cells, the total probability that $\max_j \delta(C(M, j, t+1)\lceil_{\pi_t\rho}) \geq s$ is at most $\frac{1}{4}$. Also, using Lemma 4.1 and the same reasoning as in part (a), with probability at least $\frac{1}{3}$, for sufficiently large n, the number of variables in K_t left unset by ρ is at least the expected value of $|K_t|q = |K_t|(12s(t+1))^{-1}$ which is $\geq \frac{1}{12}n(12s)^{-t}/(t+1)!$ by the inductive hypothesis. Thus the total failure probability is strictly less than 1 and we can let ρ_{t+1} be one of the restrictions for which all the conditions hold and the claim follows for $t+1$. By induction the claim for part (c) is proved. □

We can restate the resource trade-offs given in Theorem 4.1 in terms of the time required by practically sized CRCW PRAMs to compute the parity function:

COROLLARY 4.1. *If M is a CRCW PRAM that computes the parity function in time $T = T(n)$, then*

(a) *if the number of processors $p(n) = n^{O(1)}$, then*

$$T(n) \geq \frac{\log n}{O(1) + \log\log n} = \frac{\log n}{\log\log n} - O\left(\frac{\log n}{(\log\log n)^2}\right),$$

even if the number of memory cells is infinite, and

(b) *if the number of memory cells $c(n) = n^{O(1)}$, then*

$$T(n) \geq \frac{\log n}{O(1) + 2\log\log n} = \frac{\log n}{2\log\log n} - O\left(\frac{\log n}{(\log\log n)^2}\right),$$

even if the number of processors is infinite.

PROOF:

(a) From Theorem 1 part (b) we have $p(n) \geq 2^{[(1/96)n^{1/T(n)}-2]}$ or equivalently,

$$(96\log 4p(n))^{T(n)} \geq n.$$

Since $p(n) = n^{O(1)}$ there is a constant c_1 such that $(c_1\log n)^{T(n)} \geq n$ so $T(n) \geq \log n/(\log c_1 + \log\log n)$ as required.

(b) From Theorem 1 part (c) we have $c(n) \geq 2^{[(1/12)(n/T!)^{1/T}-2]}$ or equivalently,

$$(12\log 4c(n))^T \geq \frac{n}{T!}.$$

For $T \leq \frac{1}{2}\log n/\log\log n$, $T\log T \leq \frac{1}{2}\log n$. Thus $T! < 2^{T\log T} \leq \sqrt{n}$ and for values of T in this range we have

$$(12\log 4c(n))^{T(n)} \geq \sqrt{n}.$$

Substituting $c(n) = n^{O(1)}$ we see that there is a constant c_2 such that $(c_2\log n)^{T(n)} \geq \sqrt{n}$ so $T(n) \geq \frac{1}{2}\log n/(\log c_2 + \log\log n)$ and the corollary follows. □

A close look at the algorithm given by Chandra et al. [7] for computing functions with polynomial formula size, shows that parity can be computed by CRCW PRAMs with polynomially many processors and memory cells in time

$\log n/\log\log n - c\log n/(\log\log n)^2$, where the constant c depends on the exponent in the polynomial bound on the number of processors and cells. The only difference between our bound (a) and this one is that this constant c is smaller relative to the exponent of the polynomial that bounds the number of processors and cells than is the constant in our lower bound.

Using the constant-depth reductions given by Chandra et al. [7] and Furst et al. [9], these same tight lower bounds for parity can be obtained for a large number of functions. We assume that the reader is familiar with the definitions of most of these problems; the terminology is from [7].

COROLLARY 4.2 [7, 9]. *If M is a CRCW PRAM computing any of the following decision problems, the bounds in Corollary* 4.1 *hold*:

THRESHOLD, MAJORITY, UNDIRECTED GRAPH CONNECTIVITY, UNDIRECTED CYCLE DETECTION IN GRAPHS, BIPARTITE MATCHING, CIRCUIT VALUE PROBLEM.

The bounds in Corollary 4.1 *also hold for computing all the bits of the following function problems*:

MULTIPLICATION, SORTING, BIT SORTING, MULTIPLE-ADDITION, BIT SUM, NETWORK FLOW WITH UNARY CAPACITIES.

The MULTIPLE-ADDITION problem is just the integer addition problem discussed in [3] and [13]. This corollary shows that when the output is permitted to be represented as bits, the time complexity is $\Theta(\log n/\log\log n)$ for machines with polynomially bounded hardware. This complements the previous results which showed that, when the output is required to be in a single cell, the time complexity is $\Theta(\log n)$ for such machines.

The functions listed in this corollary are by no means all the natural functions to which our parity lower bound applies but merely a representative sample of the variety of problems involved.

5. *The Sipser Functions and a CRCW Time Hierarchy*

In [14], Sipser defined a set of functions F_k^m on m^k inputs for $k \geq 2$ that are described by alternating unbounded fan-in circuits of depth k and size $O(m^k)$. He obtained a strict hierarchy of polynomial-size unbounded fan-in circuits by showing that these functions required more than polynomial-size circuits of depth $k-1$. Sipser's function F_k^m was described by an alternating tree of depth k of $\wedge$ and $\vee$ gates with an $\wedge$ at the root, with fan-in m at every level, and with distinct inputs at every leaf. We modify it somewhat by defining f_k^m to be a function having fan-in $a_k = \lceil\sqrt{\frac{1}{2}mk\log m}\rceil$ from the leaves, and fan-in m everywhere else. The resulting function has $n = m^{k-1}\lceil\sqrt{\frac{1}{2}mk\log m}\rceil < m^{k-1/2}\sqrt{k\log m}$ inputs in total. Note that we even have an f_1^m that is merely an $\wedge$ of $a_1 = \lceil\sqrt{\frac{1}{2}m\log m}\rceil$ distinct variables. f_k^m can be easily computed in k steps by a CRCW PRAM with n processors and memory cells that simulates its defining circuit.

THEOREM 5.1. *If M is a CRCW PRAM that computes the function f_T^m of n inputs in time $T-1$, then for m sufficiently large*

(*a*) *the total hardware $h(n) = p(n) + c(n)$ must be at least $2^{[(1/27)(n^{1/2T}/\sqrt{2\log n})-2]}$,*
(*b*) *the number of processors $p(n)$ must be at least $2^{[(1/108)(n^{1/2T}/\sqrt{2\log n})-2]}$ even if the number of memory cells is infinite, and*
(*c*) *the number of memory cells $c(n)$ must be at least $2^{[(1/147)(n^{1/2T}/\sqrt{2\log n})-2]}$ even if the number of processors is infinite.*

To prove this theorem we define restrictions π_t for each step t of the computation just as we did for parity such that after step t and after π_t is applied, the cell (and processor) partitions all have degree less than m and yet the function to be computed is f^m_{T-t}.

Parity is a very nice function that treats 0 and 1 equally, so it is possible to use restrictions from the probability distribution R^L_p and leave the parity function unchanged in character. The functions f^m_k, which we have just described, treat 0 and 1 very differently, depending upon whether k is even or odd. Also, they are not symmetric so that treating all variables equally and independently as R^L_p does is inappropriate. The functions f^m_k do have some symmetry: inputs that appear at leaves that are joined to the same bottom level gate are symmetric with respect to each other. (We call such a set of inputs a *block*.) Also, blocks that fan in to the same second level gate are symmetric with respect to each other. These symmetries motivated the following restrictions of Hastad [11]:

Definition. Let $L \subseteq \{1, \ldots, n\}$ and let $\mathscr{L} = \{L_i\}_{i=1}^{l}$ be a partition of L into blocks. Define $R^+_{q,\mathscr{L}}$ to be a probability space of restrictions on L where for a random ρ chosen from $R^+_{q,\mathscr{L}}$ and independently for every $i \in \{1, \ldots, l\}$,

(1) A parameter s_i is chosen such that $\begin{cases} \Pr[s_i = *] = q \\ \Pr[s_i = 0] = 1 - q \end{cases}$

(2) Independently for each $j \in L_i$, $\begin{cases} \Pr[\rho(j) = s_i] = q \\ \Pr[\rho(j) = 1] = 1 - q \end{cases}$

Similarly $R^-_{q,\mathscr{L}}$ is a probability space of restrictions defined as above except that the positions of 1 and 0 are reversed.

Note that restrictions from $R^+_{q,\mathscr{L}}$ never assign $*$ and 0 to inputs from the same block and restrictions from $R^-_{q,\mathscr{L}}$ never assign $*$ and 1 to the same block. The restrictions from $R^+_{q,\mathscr{L}}$ are likely to set most inputs to 1 and will be used for f^m_k when the bottom level gates are $\wedge$; the restrictions from $R^-_{q,\mathscr{L}}$ are likely to set most inputs to 0 and will be used for f^m_{km} when the bottom level gates are $\vee$.

Definition. For a restriction ρ chosen from $R^+_{q,\mathscr{L}}$ let $g^+(\rho)$ be the restriction that agrees with ρ everywhere ρ sets inputs and that assigns 1 to all but the variable of least index in each block that is given a $*$ by ρ. Similarly, for a restriction ρ chosen from $R^-_{q,\mathscr{L}}$ let $g^-(\rho)$ be the restriction that agrees with ρ everywhere ρ sets inputs and that assigns 0 to all but the variable of least index in each block that is given a $*$ by ρ.

The definitions of g^+ and g^- are intended to be cleaned up versions of the original restrictions. The idea of this lower bound is that when f^m_k has ρ applied to it, there is a copy of f^m_{k-1} sitting inside it. In this process most of the bottom level gates will end up with more than one variable and to keep degrees small while retaining the copy of f^m_{k-1} it will be essential to apply the g^+ and g^- to reduce these to one variable.

As in the case of the parity function we need two lemmas, one making sure that the function we are trying to compute remains complicated after a restriction and one that controls the degree of the partitions.

For the first lemma we see how a restriction modifies f^m_k by looking at how it acts on the circuit for f^m_k.

LEMMA 5.1. *Let* $k \geq 2$, $\frac{1}{4} > q \geq \sqrt{(2(k-\frac{1}{2})\log m)/m}$, *and let* $\mathcal{L} = \{L_i\}_{i=1}^{l}$ *be the partition of the input set of* f_k^m *into blocks that are the sets of inputs which fan into each of its bottom level gates.*

(i) *If* k *is odd then, for* ρ *chosen at random from* $R_{q,\mathcal{L}}^{+}$, *the circuit that defines* $f_k^m \lceil_{g^+(\rho)}$ *contains a circuit that defines* f_{k-1}^m *with probability at least* $\frac{2}{3}$ *for all* $m \geq 36$.

(ii) *If* k *is even then, for* ρ *chosen at random from* $R_{q,\mathcal{L}}^{-}$, *the circuit that defines* $f_k^m \lceil_{g^-(\rho)}$ *contains a circuit that defines* f_{k-1}^m *with probability at least* $\frac{2}{3}$ *for all* $m \geq 36$.

PROOF. Suppose k is odd; the case when k is even is analogous.

CLAIM I. *With probability at least* $\frac{5}{6}$ *the* $\wedge$ *gate corresponding to block* L_i *takes the value* s_i *(as defined for* ρ*) for every* $i = 1, \ldots, l$.

For each block L_i, the only case in which the corresponding $\wedge$ gate does not take the value s_i is when ρ assigns 1 to all the variables in L_i. This probability is bounded by

$$(1-q)^{|L_i|} = [(1-q)^{1/q}]^{q|L_i|} < \exp(-qa_k)$$
$$\leq \exp\left(-\left(k-\frac{1}{2}\right)\log_2 m\right) < m^{-(k-1/2)} \leq \frac{1}{6}\, m^{-(k-1)}$$

for $m \geq 36$. Since there are exactly m^{k-1} blocks the claim follows. We now assume that all bottom level $\wedge$ gates take on their s_i value.

CLAIM II. *With probability at least* $\frac{5}{6}$ *there will be at least*

$$a_{k-1} = \lceil \sqrt{\tfrac{1}{2}m(k-1)\log m} \rceil$$

inputs to each $\vee$ *gate at the second level from the bottom that are assigned* $*$ *by* $g^+(\rho)$.

The expected number of inputs given $*$ for a single $\vee$ is the expected number of $s_i = *$ among those blocks L_i whose inputs fan-in to the $\vee$. Let $b_k = \sqrt{\frac{1}{2}m(k-\frac{1}{2})\log m}$. Since there are m blocks that fan-in to a single such $\vee$ the number of $*$'s assigned is given by a binomial distribution on m inputs with expected value $qm \geq \sqrt{2m(k-\frac{1}{2})\log m} = 2b_k \geq 2a_{k-1}$. Thus the probability that fewer than a_{k-1} inputs remain is bounded by the probability that this binomial distribution does not attain half its expected value. Applying the Chernoff bounds on the tails of the binomial distribution (e.g., see [8, page 18]), this probability is bounded above by

$$\exp\left(-m\left[\left(1-\frac{q}{2}\right)\ln\left(\frac{1-q/2}{1-q}\right) - \frac{q}{2}\ln 2\right]\right) \qquad \text{where } \ln \text{ is } \log_e.$$

Now, standard inequalities, $\ln(1-x) \geq -x/(1-x)$ and $\ln(1-x) \leq -(x + x^2/2)$, show that $(1-q/2)\ln(1-q/2) \geq -q/2$ and $-(1-q/2)\ln(1-q) \geq q(1-q^2/4)$. Thus the total bound is at most

$$\exp\left(-mq\left[1 - \frac{q^2}{4} - \frac{1+\ln 2}{2}\right]\right).$$

For $q < \frac{1}{4}$, we have $1 - q^2/4 - (1+\ln 2)/2 \geq \frac{1}{8}$, and, since $qm \geq 2b_k$, this bound is at most $e^{-b_k/4}$. Thus the probability that fewer than a_{k-1} inputs remain for at

least one of the m^{k-2} second-level gates is $\leq e^{-b_k/4}m^{k-2}$. Since $qb_k \geq (k - \frac{1}{2})\log m$, the assumption that $q < \frac{1}{4}$ implies that $b_k > 4(k - \frac{1}{2})\log m$ and it follows that we have a probability of failure of at most $m^{-3/2} < \frac{1}{6}$ for $m \geq 4$. The second claim follows.

The two claims taken together imply the lemma. □

LEMMA 5.2. *Let M be a CRCW PRAM just prior to a read or write operation, all of whose processor and cell partitions have degree at most $r \geq 1$ with variables from $\{x_i\}_{i \in L}$. Let A be either an existing processor or cell partition of M or a new cell partition resulting from a concurrent write of M. Let $\mathscr{L} = \{L_i\}_{i=1}^{l}$ be a partition of L. Choose ρ at random from $R^{+}_{q,\mathscr{L}}$ where $q \leq 1/(6r)$. For $s > 0$ we have*

$$Pr[\delta(A\lceil_{g^+(\rho)}) \geq s] < (6qr)^s.$$

The same result holds if + replaces − throughout.

This is a corollary of Lemma 3.3 and the following lemma.

LEMMA 5.3. *Let $\mathscr{G}$ be a graded set of DNF formulas on inputs $\{x_i\}_{i \in L}$ with maximum clause length bounded by $r \geq 1$ where $L \subseteq \{1, \ldots, n\}$. Let $\mathscr{L} = \{L_i\}_{i=1}^{l}$ be a partition of L. Let F be an arbitrary Boolean function on $\{0, 1\}^n$. Let ρ be a random restriction chosen from $R^{+}_{q,\mathscr{L}}$ where $q \leq 1/(6r)$. Then, if $\langle \mathscr{G}\lceil_{g^+(\rho)} \rangle$ is the partition determined by $\mathscr{G}\lceil_{g^+(\rho)}$, for $s \geq 0$ we have*

$$Pr[\delta(\langle \mathscr{G}\lceil_{g^+(\rho)} \rangle) \geq s \mid F\lceil_\rho = 0] \leq \beta^s$$

where $\beta > 0$ satisfies

$$\left(\frac{4q}{\beta(1+q)} + 1\right)^r = 2.$$

Also the same result holds if + is replaced by − throughout.

PROOF. The proof proceeds in very similar fashion to the proof of Lemma 4.3. As in that lemma we use an induction on the total number of clauses in the formulas of $\mathscr{G}$ and the base case is identical.

The idea of the proof of Lemma 4.3 was that, if ρ chosen from R_p does not falsify a clause, then it is much more likely that it satisfies it than that it leaves any input unset. This is true because of the property that, given that ρ does not set a variable to a particular value, it is more likely that it sets the variable to the other value than that it leaves the variable unset. Here, we are choosing ρ from a different distribution, $R^{+}_{q,\mathscr{L}}$ or $R^{-}_{q,\mathscr{L}}$. For a random ρ from either distribution, we still have a property that permits the proof to go through, namely, conditioned on the fact that a number of variables in a block are not set in a particular way, it is much more likely that all the variables in the block are set than that any variable in the block is unset.

Induction Step. Assume that the lemma holds for all graded sets of formulas $\mathscr{G}'$ with fewer clauses than the formulas of $\mathscr{G}$. Let F_1 be a formula in $\mathscr{G}$ that has lowest grade among those formulas in $\mathscr{G}$ that are not identically 0; let C_1 be a clause of F_1. As before we analyze the probability by considering separately the cases in which ρ does or does not force clause C_1 to be 0 and obtain

$$\begin{aligned}\Pr[\delta\langle \mathscr{G}\lceil_{g^+(\rho)} \rangle \geq s \mid F\lceil_\rho = 0] \leq \max(&\Pr[\delta\langle \mathscr{G}\lceil_{g^+(\rho)} \rangle \geq s \mid F\lceil_\rho = 0 \wedge C_1\lceil_\rho = 0], \\ &\Pr[\delta\langle \mathscr{G}\lceil_{g^+(\rho)} \rangle \geq s \mid F\lceil_\rho = 0 \wedge C_1\lceil_\rho \neq 0]).\end{aligned}$$

As in the previous proof we let $\tilde{F}_1$ be F_1 with clause C_1 removed and let $\tilde{\mathscr{G}}$ be the same as $\mathscr{G}$ with formula F_1 replaced by $\tilde{F}_1$. Again the inductive hypothesis implies that the probability in the first term is at most β^s.

Let T be the set of variables appearing in clause C_1. By hypothesis $|T| \leq r$. Let ρ_T be the restriction of ρ to the variables in T. Again we analyze the cases based on the subset of the variables in T to which ρ_T assigns $*$. However, unlike the situation in Lemma 4.3, we only separate the cases based on which blocks in $\mathscr{S}$ these unset variables belong to. This is because there will only be one unset variable in each block when $g^+(\rho)$ is applied and $g^+(\rho)$ acts independently on the blocks. We let K be the set of blocks that have variables in T and, following [11], we say that a block in K is *exposed* if it has some variable in T which is assigned $*$ by ρ. We use the notation $ex(\rho_T) = Y$ to denote the fact that the set of exposed blocks is Y. Then

$$\begin{aligned}\Pr[\delta\langle\mathscr{S}\restriction_{g^+(\rho)}\rangle \geq s \mid F\restriction_\rho = 0 \wedge C_1\restriction_{\rho_T} \neq 0] \\ = \sum_{Y\subseteq K} \Pr[\delta\langle\mathscr{S}\restriction_{g^+(\rho)}\rangle \geq s \wedge ex(\rho_T) = Y \mid F\restriction_\rho = 0 \wedge C_1\restriction_\rho \neq 0]. \quad (13)\end{aligned}$$

The probability in the case in which $Y = \phi$ is 0 since $Y = \phi$ implies that ρ sets every variable in T and, because we already know that $C_1\restriction_\rho \neq 0$, the value of C_1 is forced to 1 by ρ, making $\delta\langle\mathscr{S}\restriction_\rho\rangle = 0$. The sum in (13) then becomes

$$\begin{aligned}\Pr[\delta\langle\mathscr{S}\restriction_{g^+(\rho)}\rangle \geq s \mid F\restriction_\rho = 0 \wedge C_1\restriction_\rho \neq 0] \\ = \sum_{Y\subseteq K, Y\neq\phi} \{\Pr[\delta\langle\mathscr{S}\restriction_{g^+(\rho)}\rangle \geq s \mid F\restriction_\rho = 0 \wedge C_1\restriction_\rho \neq 0 \wedge ex(\rho_T) = Y]\} \\ \times \Pr[ex(\rho_T) = Y \mid F\restriction_\rho = 0 \wedge C_1\restriction_\rho \neq 0], \quad (14)\end{aligned}$$

by conditional probability.

As in [11] we have

CLAIM

$$\Pr[ex(\rho_T) = Y \mid F\restriction_\rho = 0 \wedge C_1\restriction_\rho \neq 0] \leq \left(\frac{2q}{1+q}\right)^{|Y|}.$$

In order to see this define R to be the set of ρ such that $ex(\rho_T) = Y \wedge F\restriction_\rho = 0 \wedge C_1\restriction_\rho \neq 0$. Also define $\bar{R}$ to be the set of ρ such that $F\restriction_\rho = 0 \wedge C_1\restriction_\rho \neq 0$ but $ex(\rho_T) \neq Y$. We define a mapping H: $R \to \bar{R}$ such that the probability of each $\bar{\rho}$ in the image of H is much larger than the sum of the probabilities of all $\rho \in R$ such that $H(\rho) = \bar{\rho}$. The bound will follow by the definition of conditional probability.

Since ρ leaves some variable in each block in Y unset, for each $L_i \in Y$ we know that $s_i = *$ in the definition of ρ. Let $P \subseteq T$ be the set of variables in the blocks of Y that appear positively in clause C_1 and let $N \subseteq T$ be those variables in the blocks of Y that appear negatively in C_1. For $\rho \in R$ define $\bar{\rho} = H(\rho)$ to be the same as ρ except that in the blocks of Y:

(i) Every variable in P to which ρ assigns $*$ is assigned 1 by $\bar{\rho}$.
(ii) Every variable not in P to which ρ assigns $*$ is assigned 0 by $\bar{\rho}$.

Since we have only changed $*$ to non-$*$ values we still have $F\restriction_{\bar{\rho}} = 0$. The changes made on the variables in C_1 only set the variables in P to 1 and variables in N to 0 so they maintain $C_1\restriction_{\bar{\rho}} \neq 0$. Lastly $ex(\bar{\rho}_T) = \phi \neq Y$ so it is clear that $\bar{\rho} = H(\rho) \in \bar{R}$. Let $S \subseteq \bar{R}$ be the image of R under the map H.

Consider $\bar{\rho} \in S$ and let $\rho \in R$ satisfy $H(\rho) = \bar{\rho}$. Let A be the set of variables in P that are assigned $*$ by ρ. We estimate the probability of ρ in terms of that of $\bar{\rho}$. For each $L_i \in Y$ we must have $s_i = *$ for ρ and it is consistent that $s_i = 0$ for $\bar{\rho}$. We will be conservative and only consider the probabilities for $\bar{\rho}$ when its $s_i = 0$ for all $L_i \in Y$. For ρ, the probability that $s_i = *$ for every L_i in Y is $q^{|Y|}$ as opposed to $(1-q)^{|Y|}$, which is the probability for $\bar{\rho}$ that $s_i = 0$ for every L_i in Y. For ρ, the probability is $q^{|A|}$ that all variables in A are assigned $*$ given that $s_i = *$ for every

L_i in Y as opposed to $(1-q)^{|A|}$ which is the probability for $\bar{\rho}$ that all variables in A are assigned 1 given that $s_i = 0$ for every L_i in Y. In the blocks of Y some variables that are assigned $*$ by ρ are assigned 0 by $\bar{\rho}$ but, given that ρ has $s_i = *$ and $\bar{\rho}$ has $s_i = 0$ for $L_i \in Y$, this does not affect the probability of ρ relative to that of $\bar{\rho}$. Finally, the variables assigned 1 by ρ are also assigned 1 by $\bar{\rho}$ and the probability of these 1's is the same whether $s_i = 0$ or $s_i = *$. Since ρ and $\bar{\rho}$ are identical in all other aspects, it follows that $\Pr[\rho] \le (q/(1-q))^{|Y|+|A|}\Pr[\bar{\rho}]$. Then we have

$$\begin{aligned}
\sum_{H(\rho)=\bar{\rho}} \Pr[\rho] &\le \sum_{A \subseteq P} \left(\frac{q}{1-q}\right)^{|Y|+|A|} \Pr[\bar{\rho}] \\
&= \left(\frac{q}{1-q}\right)^{|Y|} \Pr[\bar{\rho}] \sum_{A \subseteq P} \left(\frac{q}{1-q}\right)^{|A|} \\
&= \left(\frac{q}{1-q}\right)^{|Y|} \Pr[\bar{\rho}] \sum_{i=0}^{|P|} \binom{|P|}{i} \left(\frac{q}{1-q}\right)^{i} \\
&= \left(\frac{q}{1-q}\right)^{|Y|} \Pr[\bar{\rho}](1-q)^{-|P|}.
\end{aligned}$$

Clearly $|P| \le |T| \le r$ and by assumption $q \le 1/(6r)$, so $(1-q)^{-|P|} \le (1-q)^{-r} \le (1-1/(6r))^{-r} < 2$. Thus $\sum_{H(\rho)=\bar{\rho}} \Pr[\rho] < 2(q/(1-q))^{|Y|}\Pr[\bar{\rho}]$.

The conditional probability we wish to estimate is

$$\begin{aligned}
\frac{\sum_{\rho \in R} \Pr[\rho]}{\sum_{\bar{\rho} \in \bar{R}} \Pr[\bar{\rho}] + \sum_{\rho \in R} \Pr[\rho]} &\le \frac{\sum_{\rho \in R} \Pr[\rho]}{\sum_{\bar{\rho} \in S} \Pr[\bar{\rho}] + \sum_{\rho \in R} \Pr[\rho]} \\
&= \frac{\sum_{\bar{\rho} \in S} \sum_{H(\rho)=\bar{\rho}} \Pr[\rho]}{\sum_{\bar{\rho} \in S} \Pr[\bar{\rho}] + \sum_{\bar{\rho} \in S} \sum_{H(\rho)=\bar{\rho}} \Pr[\rho]} \\
&\le \frac{\sum_{\bar{\rho} \in S} 2(q/(1-q))^{|Y|}\Pr[\bar{\rho}]}{\sum_{\bar{\rho} \in S} [1 + 2(q/(1-q))^{|Y|}] \Pr[\bar{\rho}]} \\
&= \frac{(2q)^{|Y|}}{2^{|Y|-1}(1-q)^{|Y|} + (2q)^{|Y|}}.
\end{aligned}$$

Now, for $|Y| = 1$, the denominator equals $1+q$ and for $|Y| \ge 2$, $2^{|Y|-1}(1-q)^{|Y|} \ge \sqrt{2}^{|Y|}(1-q)^{|Y|} > (1+q)^{|Y|}$ since $q \le 1/(6r) \le \frac{1}{6}$. Thus the bound of the claim follows.

Now we look at the first term in each product in (14). The condition that $ex(\rho_T) = Y \wedge C_1\Gamma_\rho \neq 0$ determines ρ on every variable in T which is not in a block in Y. Thus we can let F' be $F \vee G$ where $G\Gamma_\rho = 0$ if and only if ρ sets the variables in T which are not in a block in Y in the unique way that does not force C_1 to 0. Unlike the situation in the proof of Lemma 4.3, the condition $G\Gamma_\rho = 0$ is not sufficient to ensure that $C_1\Gamma_\rho \neq 0$. This is because there may be variables in T that are in the blocks of Y whose value ρ sets. Because of the condition that every block in Y is exposed, any variable that ρ sets, which is in a block of Y, must be set to 1. Recall the notation P and N for the variables in the blocks of Y which respectively appear positively (negatively) in clause C_1. Observe that setting the variables in P to 1 cannot force C_1 to 0, but that setting any variable in N to 1 guarantees that C_1 is forced to 0. Then, letting ρ_Y be the restriction of ρ to the variables in the blocks of Y and using the notation $*(\rho_Y)$ for the set of variables in the blocks of Y that are

unset by ρ (ρ_Y), we have

$$\Pr[\delta\langle\mathscr{G}\lceil_{g^+(\rho)}\rangle \geq s \mid F\lceil_\rho = 0 \wedge C_1\lceil_\rho \neq 0 \wedge ex(\rho_T) = Y] = \Pr[\delta\langle\mathscr{G}\lceil_{g^+(\rho)}\rangle \geq s \mid F'\lceil_\rho = 0 \wedge ex(\rho_T) = Y \wedge N \subseteq *(\rho_Y)].$$

In order to get rid of the uncertainty about ρ's behavior on the blocks of Y we take a worst case over all possible behaviors of ρ on these blocks. This behavior is exactly captured by the set $*(\rho_Y)$. We already know that $N \subseteq *(\rho_Y)$ and that $b(T \cap *(\rho_Y)) = Y$ where b is the function that, given a set of variables, produces the set of blocks in which those variables appear. It follows that

$$\Pr[\delta\langle\mathscr{G}\lceil_{g^+(\rho)}\rangle \geq s \mid F'\lceil_\rho = 0 \wedge ex(\rho_T) = Y \wedge N \subseteq *(\rho_Y)] \leq \max_{V \supseteq N, b(V \cap T) = Y} \Pr[\delta\langle\mathscr{G}\lceil_{g^+(\rho)}\rangle \geq s \mid F'\lceil_\rho = 0 \wedge ex(\rho_T) = Y \wedge *(\rho_Y) = V].$$

Recall that the condition $ex(\rho_T) = Y$ means that the blocks in Y have variables in T that are unset by ρ and that all the variables in T that are not in blocks in Y are set by ρ. The latter part of the condition is implied by the condition $F'\lceil_\rho = 0$ and the former part of the condition is implied by the condition $*(\rho_Y) = V$ since we know that $b(V \cap T) = Y$. Thus we do not change the events by eliminating $ex(\rho_T) = Y$ to get

$$\max_{V \supseteq N, b(V \cap T) = Y} \Pr[\delta\langle\mathscr{G}\lceil_{g^+(\rho)}\rangle \geq s \mid F'\lceil_\rho = 0 \wedge *(\rho_Y) = V].$$

Suppose that $|Y| \leq s$. Let τ_1 be the restriction which sets all the variables in the blocks of Y to 1 except those in V. The condition $*(\rho_Y) = V$ implies that $(F'\lceil_{\tau_1})\lceil_\rho = F'\lceil_\rho$ since τ_1 agrees with ρ on all the inputs it sets. Also, the definition of $g^+(\rho)$ guarantees that, for each block in Y, at most one variable is unset and that all others are set to 1. By definition, this set of unset variables is completely determined by the set of variables that ρ_Y leaves unset. Let V^+ denote the set of variables in the blocks of Y that are unset by $g^+(\rho)$ (given that $*(\rho_Y) = V$) and let τ_2 be the restriction, which sets all the variables in the blocks of Y to 1 except those in V^+. It is clear that τ_2 agrees with $g^+(\rho)$ on all the inputs it sets. Then by Lemmas 3.1 and 3.2

$$\begin{aligned}
&\Pr[\delta\langle\mathscr{G}\lceil_{g^+(\rho)}\rangle \geq s \mid F'\lceil_\rho = 0 \wedge *(\rho_Y) = V] \\
&\quad \leq \sum_{\sigma \in \mathrm{Proj}[V^+]} \Pr[\delta\langle(\mathscr{G}\lceil_\sigma)\lceil_{g^+(\rho)}\rangle \geq s - |Y| \mid F'\lceil_\rho = 0 \wedge *(\rho_Y) = V] \\
&\quad = \sum_{\sigma \in \mathrm{Proj}[V^+]} \Pr[\delta\langle(\mathscr{G}\lceil_{\sigma\tau_2})\lceil_{g^+(\rho)}\rangle \geq s - |Y| \mid (F'\lceil_{\tau_1})\lceil_\rho = 0 \wedge *(\rho_Y) = V] \\
&\quad = \sum_{\sigma \in \mathrm{Proj}[V^+]} \Pr[\delta\langle(\mathscr{G}\lceil_{\sigma\tau_2})\lceil_{g^+(\rho')}\rangle \geq s - |Y| \mid (F'\lceil_{\tau_1})\lceil_{\rho'} = 0 \wedge *(\rho_Y) = V], \qquad (15)
\end{aligned}$$

where ρ' is the restriction of ρ to the blocks of $\mathscr{L}'$ and $\mathscr{L}'$ is $\mathscr{L}$ with the blocks of Y removed. To see this last equality we note that $g^+(\rho) = g^+(\rho_Y)g^+(\rho')$, $\rho = \rho_Y\rho'$, and the condition $*(\rho_Y) = V$ implies that $\tau_1 = \rho_Y$ and $\tau_2 = g^+(\rho_Y)$.

Because the probabilities over $\mathscr{L}'$ are independent of those over Y, the condition on ρ_Y does not affect the probabilities for ρ', so it can be eliminated without changing the probabilities in (15). Furthermore, because the probabilities on $\mathscr{L}'$ for ρ chosen from $R^+_{q,\mathscr{L}'}$ are the same as those for ρ' chosen from $R^+_{q,\mathscr{L}'}$, the sum in (15) is equivalent to

$$\sum_{\sigma \in \mathrm{Proj}[V^+]} \Pr[\delta\langle(\mathscr{G}\lceil_{\sigma\tau_2})\lceil_{g^+(\rho')}\rangle \geq s - |Y| \mid (F'\lceil_{\tau_1})\lceil_{\rho'} = 0], \qquad (16)$$

where ρ' is a restriction chosen at random from $R^+_{q,\mathscr{L}'}$.

By definition, $\sigma\tau_2$ sets all the variables in the blocks of Y and thus the condition $(F'\restriction_{\tau_1})\restriction_{\rho'} = 0$ guarantees that $\sigma\tau_2\rho'$ sets all the inputs in T and thus forces the value of C_1. If $C_1\restriction_{\sigma\tau_2\rho'} = 1$, then all inputs in $\langle(\mathscr{G}\restriction_{\sigma\tau_2})\restriction_{g^+(\rho')}\rangle$ are equivalent and thus $\delta\langle(\mathscr{G}\restriction_{\sigma\tau_2})\restriction_{g^+(\rho')}\rangle = 0 \le s - |Y|$. Otherwise $C_1\restriction_{\sigma\tau_2\rho'} = 0$ and then $\langle(\mathscr{G}\restriction_{\sigma\tau_2})\restriction_{g^+(\rho')}\rangle = \langle(\tilde{\mathscr{G}}\restriction_{\sigma\tau_2})\restriction_{g^+(\rho')}\rangle$ since $\tilde{F}_1\restriction_{\sigma\tau_2\rho'} = F_1\restriction_{\sigma\tau_2\rho'}$. Thus the sum in (16) is bounded by

$$\sum_{\sigma\in\mathrm{Proj}[V^+]} \Pr[\delta\langle(\tilde{\mathscr{G}}\restriction_{\sigma\tau_2})\restriction_{g^+(\rho')}\rangle \ge s - |Y| \mid (F'\restriction_{\tau_1})\restriction_{\rho'} = 0].$$

Because $(\tilde{\mathscr{G}}\restriction_{\sigma\tau_2})$ has strictly fewer clauses than $\mathscr{G}$ and because it only has inputs that appear in the blocks in $\mathscr{L}'$, we can apply the inductive hypothesis to bound the probabilities in each term in this sum by $\beta^{s-|Y|}$. For each Y the number of terms in the above sum is at most $|\mathrm{Proj}[V^+]| = 2^{|Y|}$ so we obtain a total bound of $2^{|Y|}\beta^{s-|Y|}$.

If $|Y| > s$ then we make the pessimistic assumption of failure. Since $\beta < 1$ and $s - |Y| < 0$ we have

$$\Pr[\delta\langle\mathscr{G}\restriction_{g^+(\rho)}\rangle \ge s \mid F\restriction_{\rho} = 0 \wedge C_1\restriction_{\rho_T} \ne 0 \wedge ex(\rho_T) = Y] \le 1 < 2^{|Y|}\beta^{s-|Y|}.$$

Finally, substituting these bounds in (14) we obtain a total failure probability of at most

$$\sum_{Y\subseteq K, Y\ne\phi} \left(\frac{2q}{1+q}\right)^{|Y|} 2^{|Y|}\beta^{s-|Y|} = \beta^s \sum_{i=1}^{|K|} \binom{|K|}{i} \left[\frac{4q}{\beta(1+q)}\right]^i \le \beta^s$$

using the same reasoning as before. Thus the lemma holds for $\mathscr{G}$ and by induction we have proved the lemma. Essentially the same argument with the roles of 0 and 1 reversed holds for $R^-_{q,\mathscr{L}}$ since the only real effect of reversing them in the restrictions is caused by the signs of the literals in the clauses of $\mathscr{G}$. □

PROOF OF THEOREM 5.1. Let π_0 leave every input unset. We define restrictions $\pi_1, \pi_2 \cdots$ so that $\pi_{t+1} = \pi_t\rho_{t+1}$ and ρ_{t+1} is a restriction defined on the set of inputs unset by π_t.

Part (a). Recall that we want to show that any CRCW PRAM computing f^m_T in $T-1$ steps requires total hardware, $h(n)$, at least $2^{[(1/27)(n^{1/2T}/\sqrt{2\log n})-2]}$.

CLAIM. *Let $s = \log 4h(n)$. For $t \ge 0$ and $24s \le \sqrt{m/(2\log n)}$ we can choose π_t so that $f^m_T\restriction_{\pi_t}$ is a copy of f^m_{T-t} and*

$$\max_i \delta(P(M, i, t)\restriction_{\pi_t}) \le s,$$

$$\max_j \delta(C(M, j, t)\restriction_{\pi_t}) \le s.$$

First we see how this claim implies the desired result. Observe that $\sqrt{T}m^T > n \ge m^{T-1/2}$ by definition of f^m_T so $\sqrt{m} > (n/\sqrt{T})^{1/(2T)} > \frac{8}{9}n^{1/2T}$. Thus, if we assume that $27\log 4h(n) \le n^{1/2T}/\sqrt{2\log n}$, then $24s \le \sqrt{m/(2\log n)}$ so we can apply the claim. Then, in order to compute f^m_T in $T-1$ steps, the claim requires that the first cell contain f^m_1 after π_{T-1} is applied. Obviously $\delta(f^m_1) = a_1 = \lceil\sqrt{\frac{1}{2}m\log m}\rceil$. Thus the degree of the partition in the first cell must be equal to a_1, so that $\delta(C(M, 1, T)\restriction_{\pi_{T-1}}) \ge a_1$, but this is impossible since $s < a_1$. Part (a) follows.

We now show the claim by induction on t:

Base Case. At time 0, the processor partitions all consists of a single class and for each cell C_j, $C(M, j, 0)$ is a partition that depends on at most one input bit so $\delta(C(m, j, 0)) \le 1$. Thus the claim holds initially.

Induction Step. Let $t \geq 0$. Assume the claim holds for t. We shall show that it holds for $t+1$. In a manner analogous to Theorem 4.1, we choose a restriction $\rho_{t+1} = g^+(\rho)$ for a ρ chosen at random from $R^+_{q,\mathscr{S}}$ if $T-t$ is odd (or $g^-(\rho)$ for ρ chosen from $R^-_{q,\mathscr{S}}$ if $T-t$ is even) where $q = 1/(24s)$ and $\mathscr{S}$ is the partition corresponding to the blocks of f^m_{T-t}. By the same reasoning as part (a) of the proof of that theorem (using Lemma 5.2 instead of Lemma 4.2) we see that this choice of q is sufficient to keep the degree of the processor and cell partitions bounded by s with probability at least $\frac{3}{4}$. It now remains to show that ρ_{t+1} leaves a copy of f^m_{T-t-1} inside $f^m_T\lceil_{\rho_t+1}$. The condition $24s \leq \sqrt{m/(2\log n)}$ along with $n \geq m^{T-1/2}$ implies that $q > \sqrt{(2(T-\frac{1}{2})\log m)/m} \geq \sqrt{(2(T-\frac{1}{2}-t)\log m)/m}$, which is the condition required in Lemma 5.1 for $f^m_{T-t}\lceil_{\rho_{t+1}}$ to contain a copy of f^m_{T-t-1} with probability at least $\frac{2}{3}$ for m sufficiently large. Thus the requirements on ρ_{t+1} are satisfied with probability strictly greater than 0 so we choose one of these successful ρ_{t+1}. Without loss of generality we can allow ρ_{t+1} to set more inputs so that the remaining function is f^m_{T-t-1}. By induction the claim for part (a) is proved. □

Part (b). We want to show that any CRCW PRAM computing f^m_T in time $T-1$ requires a number of processors, $p(n)$, at least $2^{[(1/108)/(n^{1/2T}/\sqrt{2\log n})-2]}$.

CLAIM. *Let $s = \log 4p(n)$. For $t \geq 0$ and $96s \leq \sqrt{m/(2\log n)}$ we can choose π_t so that $f^m_T\lceil_{\pi_t}$ is a copy of f^m_{T-t} and*

$$\max_i \delta(P(M, i, t)\lceil_{\pi_t}) \leq s,$$
$$\max_j \delta(C(M, j, t)\lceil_{\pi_t}) \leq s.$$

This claim implies (b) in the same manner as in part (a) above.

We now show the claim by induction on t:

Base Case. The base case is identical to that in part (a).

Induction Step. The induction step for (b) follows by making the same modifications to the proof of Theorem 4.1 part (b) as we made above to the proof of Theorem 4.1 part (a). We note that as in part (b) of Theorem 4.1 we must choose a probability $q = 1/(96s)$ instead of $1/(24s)$, and this is the reason for the difference in bounds from part (a). □

Part (c). This time we want to show that any CRCW PRAM computing f^m_T in time $T-1$ requires a number of memory cells, $c(n)$, at least $2^{[(1/14T)(n^{1/2T}/\sqrt{2\log n})-2]}$.

CLAIM. *Let $s = \log 4c(n)$. For $t \geq 0$, $t < T$ and $12sT \leq \sqrt{m/(2\log n)}$, we can choose π_t so that $f^m_T\lceil_{\pi_t}$ is a copy of f^m_{T-t} and*

$$\max_i \delta(P(M, i, t)\lceil_{\pi_t}) \leq st,$$
$$\max_j \delta(C(M, i, t)\lceil_{\pi_t}) \leq s.$$

This claim implies (c) in the same manner as in part (a) and (b) above.

We now show the claim by induction on t:

Base Case. This follows since $P(M, i, t)$ has only one class and thus has degree 0 and $C(M, j, t)$ has degree 1 as before.

Induction Step. The induction step for (c) follows by making the same modifications to the proof of Theorem 4.1 part (c) as we made above to the proof of Theorem 4.1 parts (a) and (b). We note that as in part (c) of Theorem 4.1 we

choose a probability $q = 1/(12s(t+1))$. In order to show that ρ_{t+1} leaves a copy of f^m_{T-t-1} inside $f^m_{T-t}\Gamma_{\rho_{t+1}}$ we need both the conditions $12sT \le \sqrt{m/(2\log n)}$ and $t < T$ to imply that $q > \sqrt{(2(T-\frac{1}{2})\log m)/m} \ge \sqrt{(2(T-\frac{1}{2}-t)\log m)/m}$. Then the induction for part (c) follows as before. □

As it stands, our functions f^m_T are defined only for certain numbers of inputs depending on T and m; call this number $\nu_{T,m}$. Let T be a function of n. We extend our functions to all numbers of inputs by defining $f_{T(\cdot)}$ on n inputs to be $f^m_{T(n)}$ computed on the first $\nu_{T(n),m}$ inputs, when m is the largest index such that $\nu_{T(n),m} \le n$. We now can restate the resources required to reduce the time for computing $f_{T(\cdot)}$ on machines with reasonable resource bounds.

COROLLARY 5.1

(*a*) *For any function T such that*

$$T(n) = \frac{\log n}{3\log\log n} - \omega\left(\frac{\log n}{(\log\log n)^2}\right),$$

there is a function $f_{T(\cdot)}$ of n inputs that can be computed on a CRCW PRAM with n processors and memory cells in time $T(n)$ but cannot be computed by any CRCW PRAM with a polynomially bounded number of processors, $p(n) = n^{O(1)}$, running in time $T(n) - 1$.

(*b*) *For any function T such that*

$$T(n) = \frac{\log n}{5\log\log n} - \omega\left(\frac{\log n}{(\log\log n)^2}\right)$$

there is a function $f_{T(\cdot)}$ of n inputs that can be computed on a CRCW PRAM with n processors and memory cells in time $T(n)$ but any CRCW PRAM computing it in time $T(n) - 1$ requires both the number of memory cells and the number of processors to exceed any polynomial in n.

PROOF. A straightforward calculation shows that, for functions T in this range, $f_{T(\cdot)}$ uses all but an $o(\log^{-2} n)$ fraction of its inputs, so there is no significant error introduced by assuming that $n = \nu_{T(n),m}$.

(a) Using Theorem 5.1 part (b) we see that to compute f^m_T in $T - 1$ steps requires $p(n) \ge 2^{[(1/108)(n^{1/2T}/\sqrt{2\log n})-2]}$ or equivalently that

$$[108(\log 4p(n))\sqrt{2\log n}]^{2T} \ge n.$$

For $p(n) = n^{O(1)}$ the inequality implies that there is a constant c_1 such that $(c_1\log n)^{3T} \ge n$ so $T \ge \frac{1}{3}\log n/(c_2 + \log\log n)$. This is just $T \ge \frac{1}{3}\log n/\log\log n - c_3\log n/(\log\log n)^2$ for some constant c_3. This is contradicted by the conditions of the claim so case (a) follows.

(b) Using Theorem 5.1 part (c) we see that to compute f^m_T in $T - 1$ steps requires $c(n) \ge 2^{[(1/14T)(n^{1/2T}/\sqrt{2\log n})-2]}$ or equivalently that

$$[14T(\log 4c(n))\sqrt{2\log n}]^{2T} \ge n.$$

For $T \le \frac{1}{5}\log n/\log\log n$, $2T\log T \le \frac{2}{5}\log n$. Thus $T^{2T} \le n^{2/5}$ and for values of T in this range we have

$$[14(\log 4c(n))\sqrt{2\log n}]^{2T} \ge n^{3/5}.$$

For $c(n) = n^{O(1)}$ the inequality implies that there is a constant c_1 such that $(c_1\log n)^{3T} \ge n^{3/5}$ so $T \ge \frac{1}{5}\log n/(c_2 + \log\log n)$ and the cell-restricted case follows as did part (a). Combining this with part (a) yields the desired result. □

This implies that the class of functions that can be computed in time-bound $T(\cdot) - 1$ on machines with reasonable bounds is strictly contained in the class of functions that can be computed in time $T(\cdot)$. This yields a strict time hierarchy among CRCW PRAMs.

6. *Almost All Boolean Functions*

If we do not worry about showing that specific functions are hard to compute, we can get larger lower bounds on the time complexity of Boolean functions than those in the previous sections by considering the class of almost all Boolean functions.

LEMMA 6.1. *Almost all Boolean functions require unbounded fan-in circuit size* $\Omega(2^{n/2})$.

PROOF. To see this, use the following argument by W. L. Ruzzo (private communication): Without loss of generality the negations can be pushed to the inputs by De Morgan's laws so we assume free access to inputs and their negations. The number of unbounded fan-in in circuits with s gates is then just $2^{s(s+2n+1)}$ since each gate can be described by its operation (either $\wedge$ or $\vee$) and by the subset of the inputs and gates to which it is attached. Since there are 2^{2^n} Boolean functions of n inputs, it easy to see that most functions require size $\Omega(2^{n/2})$. □

Using the simulation of CRCW PRAMs by circuits given by Li and Yesha [13] and B. Chor (private communications) (cf. Section 2) along with this lemma yields:

THEOREM 6.1. *Almost all Boolean functions of n inputs require time $\log n - \log\log p(n) + \Omega(1)$ on a CRCW PRAM with $p(n)$ processors.*

PROOF. Substituting directly in the simulation we see that any CRCW PRAM taking at most $\log n - \log\log p(n) - \omega(1)$ time can be simulated by an unbounded fan-in circuit of size $o(2^{n/2})$. But by Lemma 6.1, almost all Boolean functions of n inputs require unbounded fan-in circuits of size $\Omega(2^{n/2})$. The theorem follows immediately. □

Because of the upper bound in [3] of $\log n - \log\log[p(n)/n] + O(1)$ for computing any function on Boolean inputs, this bound is nearly optimal. This optimality suggests that no general improvement in the simulation of CRCW PRAMs by unbounded fan-in circuits is likely to be obtained.

7. *Directions for Further Research*

The $\Omega(\log n/\log\log n)$ time lower bounds for computing specific Boolean functions given in Sections 4 and 5 are tantalizingly close to the $\log n - \log\log n + \Theta(1)$ time bounds for almost all Boolean functions on CRCW PRAMs with a polynomial number of processors. However, finding a specific problem in NP for which we can close this gap appears to be a formadiable though fundamentally interesting task. This is because the work of Chandra et al. [7] and Stockmeyer and Vishkin [15] implies that such a problem would not be in NC^1 (would not have $O(\log n)$ depth combinatorial circuits).

ACKNOWLEDGMENTS. Many thanks to Steve Cook and Charlie Rackoff for their encouragement and their detailed reading and careful criticisms of drafts of this work. Thanks also to Al Borodin and Faith Fich for their many helpful comments and suggestions.

REFERENCES

1. AJTAI, M. Σ_1^1-Formulae on finite structures. *Ann. Pure Appl. Logic 24*, 1983, 1–48.
2. BEAME, P. W. *Lower Bounds for Very Powerful Parallel Machines.* Manuscript, 1985.
3. BEAME, P. W. Limits on the power of concurrent-write parallel machines. *Inf. Computation 76*, 1 (1988), 13–28.
4. BEAME, P. W. *Lower Bounds in Parallal Machine Computation.* Ph.D. Dissertation, TR 198/87. Univ. Toronto, Toronto, Ont., Canada, 1986.
5. BEAME, P. W., AND HASTAD, J. Optimal bounds for decision problems on the CRCW PRAM. In *Proceedings of the 19th ACM Symposium on Theory of Computing* (New York, N.Y., May 25–27). ACM, New York, 1987, pp. 83–93.
6. BOLLOBÁS, B. *Random Graphs.* Academic Press, Orlando, Fla., 1985.
7. CHANDRA, A. K., STOCKMEYER, L. J., AND VISHKIN, U. Constant depth reducibility, *SIAM J. Comput. 13*, 2 (1984), 423–439.
8. ERDÖS, P., AND SPENCER, J. *Probabilistic Methods in Combinatorics.* Academic Press, Orlando, Fla., 1974.
9. FURST, M., SAXE, J. B., AND SIPSER, M. Parity, circuits, and the polynomial time hierarchy, *Math. Syst. Theory 17*, 1 (1984), 13–28.
10. HASTAD, J. Almost optimal lower bounds for small depth circuits. In *Proceedings of the 18th Annual Symposium on Theory of Computing* (Berkeley, Calif., May 28–30). ACM, New York, 1986, pp. 6–20.
11. HASTAD, J. *Computational Limitations for Small Depth Circuits.* MIT Press, Cambridge, Mass., 1987.
12. KUCERA, L. Parallel computation and conflicts in memory access. *Inf. Proc. Lett. 14*, 2 (1982), 93–96.
13. LI, M., AND YESHA, Y. New lower bounds for parallel computation. *J. ACM 36*, 3 (July 1989), 671–680.
14. SIPSER, M. Borel sets and circuit complexity. In *Proceedings of the 15th Annual ACM Symposium on the Theory of Computing* (Boston, Mass., Apr. 25–27). ACM, New York, 1983, pp. 61–69.
15. STOCKMEYER, L. J., AND VISHKIN, U. Simulation of parallel random access machines by circuits. *SIAM J. Comput. 13*, 2 (1984), 404–422.
16. YAO, A. C. Separating the polynomial-time hierarchy by oracles: Part I. In *Proceedings of the 26th IEEE Foundations of Computer Science.* IEEE, New York, 1985, pp. 1–10.

RECEIVED AUGUST 1987; REVISED NOVEMBER 1988; ACCEPTED NOVEMBER 1988