The Entropy of Traces in Parallel Computation

Rajit Manohar
School of Electrical Engineering
Cornell University
Ithaca, NY 14853

Abstract— **The following problem arises in the context of parallel computation: how many bits of information are required to specify any one element from an arbitrary (non-empty) k-subset of a set? We characterize optimal coding techniques for this problem. We calculate the asymptotic behavior of the amount of information necessary, and construct an algorithm that specifies an element from a subset in an optimal manner.**

Keywords— **Coding; Parallel computation; Energy; Traces**

In this note we discuss a problem in information theory that arises in the context of parallel computation. A parallel computation can be represented by a subset of the set of all traces, where a trace is a sequence of symbols [1]. A symbol indicates a change in voltage on a particular wire. A trace indicates the order in which a sequence of voltage changes occur, and a set of traces is used to describe all permissible orders of execution. For example, consider a one-bit full adder, a circuit with two inputs a and b, and two outputs s (sum) and c (carry). We use a_t to denote a high voltage value on input a and a_f to denote a low voltage value on input a. The operation of adding two zeros can be depicted by the set of traces $\{a_f b_f c_f s_f,\ a_f b_f s_f c_f,\ b_f a_f c_f s_f,\ b_f a_f s_f c_f\}$. The specification of a computation is given by a set of permissible traces.

Suppose we have a computation that is described by such a trace set. Each trace in the set describes a particular implementation of the computation. Choosing a particular trace corresponds to restricting the amount of concurrency in the system. For example, the single trace $a_f b_f s_f c_f$ describes the operation of adding two zeros to produce a zero, but prevents the circuit from concurrently computing the sum and carry, since such a circuit could also produce the trace $a_f b_f c_f s_f$.

Each trace in the trace set has some probability of being chosen for execution. This probability distribution is dependent on the context in which the VLSI system is used and can, in general, be arbitrary. The entropy of the specification is the entropy of this probability distribution function.

A VLSI computation is a non-terminating iterative system, and the entropy of its specification is related to the energy it dissipates. If $C(S)$ is the energy dissipated by the VLSI implementation of specification S and $H(S)$ is the entropy of the input/output behavior of S, Theorem 3.5 from [2] shows that $H(S) \leq C(S) \leq H(S) + K(S)$, where $K(S)$ is a number that depends on the length of an iteration of the computation. The rationale for this result is that the energy dissipated corresponds to the cost of selecting a particular trace for execution.

The research described in this report was sponsored by the Advanced Research Projects Agency and monitored by the Office of Army Research.

The behavior of a VLSI system can also be restricted by its environment. The environment restriction is specified by a set of permissible traces. We consider environments that are described by arbitrary subsets of fixed cardinality (the cardinality restriction corresponds to restricting the amount of concurrency). The VLSI system must pick a trace that is permitted by its environment.

When a computation is specified using trace sets, each trace from the trace set is a valid implementation of the computation. Therefore, we are interested in the information necessary to specify a single element from the environment trace set. Abstracting away from computations and traces, we can state the problem as follows:

> *Let $\mathcal{U}$ be a fixed set of cardinality N. Let S be an arbitrarily chosen non-empty subset of $\mathcal{U}$ of fixed cardinality k. How many bits of information are necessary to identify any one element from S?*

Without loss of generality, we can assume that the set $\mathcal{U}$ consists of the first N natural numbers.

Lemma 1: Given $\mathcal{U}$, a fixed set with cardinality N and any $S \subseteq \mathcal{U}$ with fixed cardinality $k > 0$, the information necessary to identify any one element from S lies between $\lg(N-k+1)$ and $\lceil \lg(N-k+1) \rceil$ bits.

Proof: To show that $\lceil \lg(N-k+1) \rceil$ bits is an upper bound, consider the following scheme for achieving this bound:

> Method **M0**. Delete the first $k-1$ elements from $\mathcal{U}$ to obtain $\mathcal{U}'$. Clearly, no matter which S we pick, there will be *some* element from S that is contained in $\mathcal{U}'$. Encode this element using $\lceil \lg(N-k+1) \rceil$ bits of information.

To show that $\lg(N-k+1)$ bits is a lower bound, suppose there is an algorithm that takes as input some number of bits—say $\lg h$—of information and produces some element from S as a result. Without loss of generality, we can assume that this algorithm is deterministic in its input. If it were not, then there are many outcomes all of which must be contained in S, and we can canonically pick one of the possible outcomes. In other words, we can assume that the algorithm can be represented as a table of size h which encodes the mapping from the input string to elements of $\mathcal{U}$. If the table contains fewer than $N-k+1$ entries, we can choose a subset of $\mathcal{U}$ that has no representative in this table. So, the table must have at least $N-k+1$ entries, concluding the proof. ■

Let the distribution of the k-subsets of $\mathcal{U}$ be given by $\Pr(\cdot)$. We use $\Pr(\overline{a}b)$ to denote the probability that a randomly chosen subset contains element b and does not contain element a. The following theorem characterizes optimal coding techniques (by which we mean coding techniques that minimize entropy) for determining an element from a subset of $\mathcal{U}$.

Theorem 1: Any optimal method for encoding any one element from an arbitrary k-subset $S \subseteq \mathcal{U}$ corresponds to choosing an element from a fixed subset $\{k_1, \ldots, k_{N-k+1}\}$ of $\mathcal{U}$. Let the probability of picking k_i be q_{k_i} and, without loss of generality, let $q_{k_i} \geq q_{k_j}$ for $i \leq j$. Then the optimal coding technique consists of picking element $k_i \in S$ such that $k_j \notin S$ for $j = 1, 2, \ldots, i-1$, and $q_{k_i} = Pr(\overline{k_1}\overline{k_2} \cdots \overline{k_{i-1}} k_i)$.

Proof: By lemma 1, we need to consider at least $N-k+1$ elements. Assume that **M1** is some optimal encoding technique. Once again, we can assume that this algorithm is equivalent to a table $\mathcal{U}''$ that maps input strings to elements from $\mathcal{U}$. Let q_i be the probability of choosing entry i from $\mathcal{U}''$ using algorithm **M1**, and let $j = argmax_i\, q_i$. If **M1** is optimal, we claim that $q_j = \Pr(j)$. For if $q_j < \Pr(j)$, there is some subset of $\mathcal{U}$ which contains j for which **M1** chooses another element l from $\mathcal{U}''$. We can increase q_j and decrease q_l by modifying **M1** to pick j instead of l for that particular subset. We observe that if $x \geq y$, the function $(x+\epsilon)\lg\frac{1}{x+\epsilon} + (y-\epsilon)\lg\frac{1}{y-\epsilon}$ decreases with increasing ϵ (its derivative is $\lg\frac{y-\epsilon}{x+\epsilon}$ which is negative when $x \geq y$). Since $q_j \geq q_l$, increasing q_j by ϵ and decreasing q_l by ϵ reduces the entropy of the distribution—a contradiction, from which we conclude that $q_j = \Pr(j)$. Repeating this argument concludes the proof. ∎

A VLSI circuit is a non-terminating system, and therefore the trace sets are infinite in size. We consider the case when N, k, and $N-k$ are infinite. We make the simplifying assumption that the elements of the k-subset are chosen *independently*, i.e., $\Pr(ab) = \Pr(a)\Pr(b)$.[1] In this case, the probabilities $p_i = \Pr(i)$ determine the distribution Pr. Under these assumptions, the following theorem completely characterizes optimal coding schemes.

Theorem 2: Assume that set $\mathcal{U}$ is infinite, and that the elements of $S \subseteq \mathcal{U}$ are independently chosen. Let $\{k_1, k_2, k_3, \ldots\}$ be a subset of $\mathcal{U}$ that optimizes the encoding. Let the probability of picking k_i be q_{k_i} where, without loss of generality, $q_{k_i} \geq q_{k_j}$ for $i \leq j$. Then k_i is the ith most probable element from $\mathcal{U}$, and the optimal coding technique consists of picking $k_i \in S$ such that $k_j \notin S$ for $i = 1, 2, \ldots, i-1$, resulting in $q_{k_i} = p_{k_i}\prod_{j<i}(1-p_{k_j})$.

Proof: The fact that the optimal coding technique consists of picking the elements from a fixed set $\{k_1, k_2, \ldots\} \subseteq \mathcal{U}$ in order follows from Theorem 1, as do the probabilities q_{k_i}.

The entropy of the distribution q_{k_i} is given by the expression $\sum_i \left(\prod_{j<i}(1-p_{k_j})\right)\mathcal{H}(p_{k_i})$, where $\mathcal{H}(x)$ is the binary entropy function defined to be $-x\lg x - (1-x)\lg(1-x)$. If the entropy of distribution q_{k_i} is the minimum possible, then exchanging the position of elements k_i and k_{i+1} cannot reduce the entropy. Using this fact, we conclude that $\mathcal{H}(p_{k_i}) + (1-p_{k_i})\mathcal{H}(p_{k_{i+1}}) \leq \mathcal{H}(p_{k_{i+1}}) + (1-p_{k_{i+1}})\mathcal{H}(p_{k_i})$, which simplifies to the condition $\mathcal{H}(p_{k_i})/p_{k_i} \leq \mathcal{H}(p_{k_{i+1}})/p_{k_{i+1}}$. Since $\mathcal{H}(x)/x$ is decreasing on the interval $(0,1]$ (its derivative is $-\frac{1}{x^2}\lg\frac{1}{1-x}$), this condition is equivalent to $p_{k_i} \geq p_{k_{i+1}}$.

If $p_l > p_{k_1}$ then by the condition just derived, $l \notin \{k_1, \ldots, k_{N-k+1}\}$. Let q'_{k_i} be the distribution obtained after replacing k_1 with l. Then, $q'_{k_1} > q_{k_1}$, and for all $i > 1$, $q'_{k_i} < q_{k_i}$. Since $q_{k_1} \geq q_{k_i}$ for all i, increasing q_{k_1} and reducing q_{k_i} for $i > 1$ reduces the entropy (see proof of Theorem 1)—a contradiction, from which we conclude that $p_{k_1} = max_i\, p_i$. Repeating this argument concludes the proof. ∎

Corollary 1: The entropy of the distribution q_{k_i} is finite if $\lim_{n\to\infty} p_{k_n} > 0$, or $\lim_{n\to\infty}\sqrt[n]{-p_{k_n}\lg p_{k_n}} < 1$.

Proof: The entropy of the distribution is given by the expression $\sum_i \left(\prod_{j<i}(1-p_{k_j})\right)\mathcal{H}(p_{k_i})$. Under the assumption that $p_{k_i} \geq p_{k_j}$ for $i \leq j$, the nth term of the infinite summation is bounded above by $(1-p_{k_n})^{n-1}\mathcal{H}(p_{k_n})$, since $(1-p_{k_n}) \geq (1-p_{k_i})$ for $i < n$. Using the nth root test, we conclude that the infinite sum converges if $\lim_{n\to\infty}(1-p_{k_n})\sqrt[n]{\mathcal{H}(p_{k_n})} < 1$, which holds if $\lim_{n\to\infty} p_{k_n} > 0$. If $p_{k_n} \to 0$ as $n \to \infty$, we can bound the limit in the nth root test by observing that $(1-p_{k_n}) \leq 1$, and that the second term of the binary entropy function is not significant in the limit of large n. The simplified limit term in the nth root test is $\sqrt[n]{-p_{k_n}\lg p_{k_n}}$. ∎

If the distribution p_{k_i} goes to zero as x^i for some constant $x < 1$, then Corollary 1 implies that the entropy of the distribution is finite.

If the input distribution is uniform, all $N-k+1$-subsets of $\mathcal{U}$ are equivalent. Transforming method **M0** to pick the smallest element from $S \cap \mathcal{U}'$ is optimal even when N is finite. In the infinite case, the uniform distribution satisfies the independence assumption, and we conclude that:

*Corollary 2: If the input distribution is uniform, transforming method **M0** to pick the smallest element from $S \cap \mathcal{U}'$ is an optimal coding scheme. Let $H(N,k)$ denote the entropy of $\mathcal{U}'$, and let $p = \frac{k}{N}$ be fixed. Then,*

$$\lim_{N,k\to\infty} H(N,k) = \frac{\mathcal{H}(p)}{p}$$

Proof: Assuming a uniform distribution of the subsets, the probability of picking the ith entry in the table as $N \to \infty$ is $p(1-p)^{i-1}$ (Theorem 2). Computing the entropy of this distribution concludes the proof. ∎

To gain some intuition about why this function represents the amount of information necessary, we make the following observations. Given a set of size N, there are $\binom{N}{pN}$ possible subsets of size $pN = k$. With $\lg\binom{N}{pN}$ bits we can specify every element from a subset. However, we are only interested in one element from the subset. Since the subset has pN elements all of which are equiprobable, we have provided pN times the amount of information necessary. For large N, $\lg\binom{N}{pN}$ is approximately $N\mathcal{H}(p)$, and if

[1] If $N-k$ is finite, the assumption of independence is not realistic. Since $S \subseteq \mathcal{U}$ has cardinality k, $\Pr(\overline{k_1}\cdots\overline{k_{N-k+1}}) = 0$, which (assuming independence) would imply that $\Pr(k_i) = 1$ for some i—making the original problem trivial.

we divide this by pN, we get $\mathcal{H}(p)/p$, which matches the asymptotic behavior of $H(N, k)$.

Acknowledgments: I am indebted to Prof. Yaser S. Abu-Mostafa for his detailed comments and suggestions, and to the anonymous referees for suggestions that improved the proofs.

References

[1] Jan L. A. van de Snepscheut. Trace Theory and VLSI Design. PhD Thesis, Eindhoven University of Technology, 1983.

[2] José A. Tierno. An Energy Complexity Model for VLSI Computations. PhD Thesis, California Institute of Technology, 1995.

Rajit Manohar Rajit Manohar received his B.S., M.S., and Ph.D. in Computer Science from the California Institute of Technology. He is currently an Assistant Professor of Electrical Engineering at Cornell University. His research interests include asynchronous VLSI design, asynchronous computer architecture, low power design, and the use of concurrency theory for the design of reliable and robust asynchronous systems.