Feasible Models of Computation: Three-Dimensionality and Energy Consumption

Peter Sanders, Roland Vollmar, Thomas Worsch

University of Karlsruhe, Department of Computer Science, 76128 Karlsruhe, Germany
Email: {sanders,vollmar,worsch}@ira.uka.de

Abstract. Starting with cellular automata as a model of parallel machines we investigate the constraints for the energy consumption of r-dimensional machines which are motivated by fundamental physical limitations for the case $r = 3$. Depending on the operations which must be considered to dissipate energy (state changes, communication), relations between the relative performance of 2-D and 3-D machines are derived.

1 Introduction

At now feasible clock speeds approaching 1 GHz, information can travel at most 30 cm during one clock cycle. Hence transmission latencies are an issue for large parallel computers and there is an interest in (physically and economically) feasible architectures with short interconnections.

Usually these issues are discussed for the case $r = 2$ because chips and printed circuit boards allow only a constant number of layers. The starting point for this paper is to look at the the general r-D case with an emphasis on the case $r = 3$ and scaling issues which are not a trivial generalization of the case $r = 2$. More specifically we will consider the cooling problem and simulations between 2-D and 3-D systems.

What kind of model is adequate here? A (conceptually) very simple model is the *grid model of circuits* [8] which describes the circuit as a regular orthogonal grid of cells. This model is adequate since inaccuracies in the production technology forbid to exploit arbitrarily accurate placement of materials. The cell model is in turn equivalent to cellular automata (CA) where the state of an automaton cell encodes the states of a fixed number of VLSI-cells. The additional feature of synchronous operation does not increase the performance of CA [5]. The energy consumption of 2-D VLSI circuits is an area of intense study. For example refer to [4] for a theoretical treatment and to the PATMOS series of workshops for practical aspects. The results for higher dimensions presented here illuminate additional aspects like the relation between speed and energy consumption. Also, our results apply to functions with a single output bit making it possible to use the formalisms developed in formal language theory.

In Sect. 2 we introduce some basic notions. In Sect. 3 it turns out that the energy consumption has to be taken into account for $r = 3$ and that this has some interesting implications for the performance of the machine. The tradeoffs between different models are considered in Sect. 4. Finally, Sect. 5 summarizes the paper. For the proofs of all propositions and a more detailed discussion of related topics readers are referred to the technical report [6].

2 Some Basic Notions

A $\mathbb{Z}^r$–CA consists of a collection of identical deterministic finite automata positioned at all grid points of the underlying integer lattice $\mathbb{Z}^r$. Q denotes the *set of states* and N the r-dimensional von Neumann neighborhood.

A *(global) configuration* of a CA is a mapping $c : \mathbb{Z}^r \to Q$. The local rule of a CA determines the new state of a cell depending on the current states of all cells belonging to its neighborhood. All cells synchronously update their state according to the local rule $\delta : Q^N \to Q$.

As an example, the recognition of formal languages by CA will be considered. Let $A \subset Q$ denote some *input alphabet* with $|A| \geq 3$ containing a symbol $\mathtt{@}$. We assume the symbols of an input $w \in A^+$ of length n to be numbered from 0 to $n-1$. Input symbols are provided in a "row major order" to a small *input cube* of cells. For $k \in \mathbb{N}_+$ denote by W_k the cube $\{(x_1, \dots, x_r) \mid \forall i : 0 \leq x_i < k\}$. Let $m = \lceil n^{1/r} \rceil$. At the beginning of a computation a cell in W_m with coordinates $(x_1, \dots, x_r)$ receives the j-th symbol of w, if $0 \leq j = \sum_{i=1}^{r} x_i m^{i-1} < n$. In the initial configuration for an input w all cells not containing an input symbol initially are in a so-called *quiescent state.* Cell $\mathbf{0}$ has to produce the result whether the input has been accepted or rejected.

The *time* complexity of a CA is defined as usual. By $W(n)$ we denote the smallest cube comprising all cells used during the computation for at least one input of size n. The *extent* of $W(n)$ is denoted by $d(n)$; the *size* of $W(n)$ is $d(n)^r$.

We will consider (variants of) two languages throughout the paper:

$$L_{\text{parity}} := \{w \mid w \in \{0,1\}^+ \text{ and } w \text{ contains an even number of 1s}\}$$
$$L_{\text{vv}} := \{v\mathtt{@}^{|v|}v \mid v \in \{0,1\}^+\}$$

For a language L let $L^{[r]} := \{w \in L \mid |w|^{1/r}/3 \text{ is an integer}\}$.

L_{parity} and L_{vv} can be accepted without using cells outside the input cube in time $\Theta(n^{1/r})$ which is asymptotically optimal.

3 Energy Consumption

Cooling of chips and computers is a crucial issue in hardware design. In two dimensions, cooling poses no limit to building larger machines since the surface of the machine grows in proportion with the number of active elements. But in three dimensions this is no longer the case.

A 3-D machine with extent d can have $O(d^3)$ active elements while all the dissipated energy has to be transported through the surface of some cube with surface area $\Theta(d^2)$. But the maximum allowable temperature at any point in the machine must not exceed a certain constant value, i.e., it cannot be scaled with d. Hence, in a large machine it is not feasible to consume one unit of energy in each cell in each step because eventually the machine would become overheated.

Proposition 1. *For every physically feasible computation, every subcube of the machine with extent d' and every interval of the computation of length t', no more than $\mathrm{O}(d'^3 + t'd'^2)$ units of energy may be dissipated within this space-time interval of the computation. (Analogously for the general r-dimensional case.)*

Which operations consume energy? From the laws of thermodynamics it only follows that each irreversible computation consumes at least $k_B T \ln 2$ of energy (k_B is the Boltzmann constant, T the temperature in Kelvin). In principle, universal computers can be built using only reversible gates. However, gates which actually consume very little energy are currently only gedanken experiments and/or trade speed for energy consumption, so they cannot be used for building *fast* computers. For a more detailed discussion refer to [3]. So let us start by approximating the actual energy consumption of a machine by counting the (proper) state changes of its cells. Especially for CMOS this is quite accurate.

Proposition 1 can now be reformulated in terms of the numbers of state changes [9]. The *(state) change complexity* of a CA with time complexity t is the function s where $s(n)$ is the maximum number of *proper* state changes of all cells which happen for computations on inputs of size n. Obviously $s(n) \geq t(n)$. On the other hand, a CA which fulfills Prop. 1 must have a time complexity of at least $t(n) \in \Omega\left(\frac{s(n)-d(n)^3}{d(n)^2}\right)$ if $s(n)$ changes occur in a cube of extent $d(n)$. We call such a constrained CA a $\mathbb{Z}^r$−CACE (where CE stands for "cold everywhere").

In the remainder of this section, we identify two related languages which cannot be accepted in time $\mathrm{O}(n^{1/r})$ by $\mathbb{Z}^r$−CACE due to a large change complexity in some subcube. We call a nondecreasing function $f(n)$ *almost-log* if $f(n) \in \mathrm{O}(\mathrm{Poly}(\log n))$ and $f(n) \notin \mathrm{O}(\log n)$.

Proposition 2. *A $\mathbb{Z}^r$−CA recognizing L_{vv} makes a total of $\Omega(n^{(r+1)/r}/f(n))$ state changes for every almost-log $f(n)$ in the subcube of extent $3n^{1/r}$ containing the input cube in its center (and analogously for $L_{\mathrm{vv}}^{[r]}$).*

Corollary 3. *No $\mathbb{Z}^r$−CACE can accept L_{vv} in less than $n^{2/r}/f(n)$ steps.*

On the other hand, note that as a consequence of Proposition 6 below L_{vv} can be recognized by $\mathbb{Z}^r$−CACE in time $\Theta(n^{2/r})$.

In addition, there are languages where any CACE has to be slower than a general CA due to a large change complexity in a subcube although the overall change-complexity is small. Let L_1 denote the language of all words with the following r-dimensional arrangement: The central subcube of extent $n^{1/(r+1)}$ (and size $n^{r/(r+1)}$) contains a word from $L_{\mathrm{vv}}^{[r+1]}$ and the remainder of the whole cube is filled with @ symbols everywhere. Proposition 2 implies:

Corollary 4. *A $\mathbb{Z}^r$−CA C recognizing L_1 will make a total of $\Omega(n/f(n))$ state changes in the central subcube M of the input cube of extent $3n^{1/(r+1)}$. Therefore, a $\mathbb{Z}^r$−CACE needs at least $\Omega(n^{2/(r+1)}/f(n))$ steps to accept L_1.*

We have seen that technologies which limit the change complexity considerably constrain the performance of 3-D machines. One candidate for a relaxation is communication. In terms of CA, handing information from one cell to another must involve state changes. Therefore, communicating one bit of information through a "wire" requires energy proportional to its length. But there are technologies which do have negligible energy consumption per unit of wire length, e.g., modern optical fibers. On the other hand the problem really is an issue for current CMOS technology (e.g., [2,7]).

Therefore we introduce a modification of the CAce model, denoted CAww ("with wires"): Each cell has access to unidirectional "wire"-registers for each of the $2 \cdot r$ coordinate directions. The information in these registers moves without consuming energy. However, reading or writing a wire register requires one unit of energy. On this model one can recognize L_{vv} in time $\mathrm{O}(n^{1/r})$ without violating Proposition 1.

4 Comparison of the Models

After having introduced the different models (CA, CAce, CAww) in different dimensions, we are now going to state some results about the relations between the models. Proofs can be found in [6].

For no language a $\mathbb{Z}^r$–CA has to be slower than a $\mathbb{Z}^r$–CAww or a $\mathbb{Z}^r$–CAce. From Corollary 3 and the remark above it follows that in some cases they are faster, i.e., the restriction of energy consumption implies a restriction of the computational power of CA for fixed time bounds:

Proposition 5 (Restricted vs. unrestricted CA). *There are problems for which $\mathbb{Z}^r$–CA are faster than $\mathbb{Z}^r$– CAce by a factor of $\Omega(n^{2/r}/f(n))$ for every almost-log $f(n)$. The same holds for $\mathbb{Z}^r$–CAww instead of $\mathbb{Z}^r$–CA.*

On the other hand each CA can be transformed into an equivalent (but slower) CAce. Let $\mathbb{Z}^r$–CA–Ext–Time(d,t) denote the family of languages recognized by $\mathbb{Z}^r$–CA with extent at most $\mathrm{O}(d)$ and time complexity at most $\mathrm{O}(t)$; for $\mathbb{Z}^r$–CAce a similar notation will be used.

Proposition 6 (Reduction of state changes per step).
$\mathbb{Z}^r\text{–CA–Ext–Time}(d,t) \subseteq \mathbb{Z}^r\text{–CAce–Ext–Time}(d, d \cdot t)$.

Even for unrestricted CA one can ask what price one has to pay when changing the dimensionality (see also [1]).

Proposition 7 (Change of dimension).
If d is space-constructible in time t:

1. $\mathbb{Z}^r\text{–CA–Ext–Time}(d,t) \subseteq \mathbb{Z}^{r-1}\text{–CA–Ext–Time}(d^{r/(r-1)}, d^{1/r(r-1)}t)$
2. $\mathbb{Z}^{r-1}\text{–CA–Ext–Time}(d,t) \subseteq \mathbb{Z}^r\text{–CA–Ext–Time}(d^{(r-1)/r}, t)$

Algorithms for L_{parity} and L_{vv}, Cor. 3 and Prop. 7.1 and Prop. 6 verify the following statements, where L_{vv} may be used for deducing the tightness facts:

Proposition 8 (Dimensionalities vs. restrictions). *There are problems for which $\mathbb{Z}^r$–CACE are faster than $\mathbb{Z}^{r-1}$–CA by a factor of $n^{1/(r(r-1))}$ which is tight. On the other hand, there are problems for which $\mathbb{Z}^{r-1}$–CA are faster than $\mathbb{Z}^r$– CACE by a factor of $n^{1/(r(r-1))}/f(n)$ which is tight up to the $f(n)$ factor.*

In other words, e.g., for L_{vv} the time saved by going from (feasible) 2-dimensional CA to (feasible) 3-dimensional CA due to the smaller extent gets more than lost because of the restriction on the number of state changes.

5 Conclusions

This paper discusses some present and future problems of parallel machine design using the CA model. For classical irreversible computing (having CMOS technology in mind), change complexity elegantly models the energy consumption and mirrors the amount of information transmission. In the light of this model, the third space-dimension turns out to play an important role. In a sense, even machines traditionally thought as two-dimensional require the third dimension for cooling and architectures exploiting the third dimension for additional purposes must not increase the energy consumption per volume by more than a constant factor. If we have a technology where the energy consumption of information transmission does not grow with wire length there is still a lot of freedom. For example, a moderately coarse grained machine with n processors and $\Omega(n^{1/2})$ memory cells per processor can be economically equipped with a full multistage butterfly using 3-D wiring.

Interesting future work in this direction includes different wire models, scaling properties of free space optical interconnects and more detailed models for memory cells and memory hierarchies.

References

1. A.-C. Achilles, M. Kutrib, and T. Worsch. On relations between arrays of processing elements of different dimensionality. In R. Vollmar, W. Erhard, and V. Jossifov, editors, *Parcella '96*, pages 13–20. Akademie Verlag, 1996.
2. A. Aggarwal, A. K. Chandra, and P. Raghavan. Energy consumption in VLSI circuits (preliminary version). In *STOC '88*, pages 205–216, Chicago, 1988.
3. R. P. Feynman. *Feynman Lectures on Computation.* Addison Wesley, 1996.
4. T. Lengauer. VLSI theory. In *Handbook of Theoretical Computer Science*, volume A: Algorithms and Complexity, chapter 16, pages 835–868. Elsevier, 1990.
5. K. Nakamura. Asynchronous cellular automata and their computational ability. *Systems, Computers, Controls*, 5:58–66, 1974.
6. P. Sanders, R. Vollmar, and T. Worsch. Feasible models of computation: Three-dimensionality and energy consumption. TR 2/97, U. Karlsruhe, CS Dept., 1997.
7. J. Smit and J. A. Huisken. On the energy complexity of the FFT. In *PATMOS '95*, pages 119–132, 1995.
8. J. D. Ullman. *Computational Aspects of VLSI.* Computer Science Press, 1984.
9. R. Vollmar. Some remarks about the "efficiency" of polyautomata. *International Journal of Theoretical Physics*, 21:1007–1015, 1982.