A NEW MODEL FOR LARGE MEMORIES

R. FRANCESE (2) - L. PAGLI (3) - D. PARENTE (2) - M. TALAMO (4)

ABSTRACT - In this paper a new Memory Model of Computation (CMM) is introduced. In CMM, a RAM processor accesses a memory of \( x \) cells in \( \log x \) time. In fact, the usual assumption of the RAM model, that all memory cells are accessed in constant time, becomes impractical as \( x \) increases.

With a very simple modification of the boolean circuits of the memory, CMM makes it possible to access in constant time, a memory cell consecutive to another already accessed cell. Problems of size \( n \) requiring time \( T(n) \) in the RAM model can be solved in CMM with a multiplicative factor \( O(\log x) \) in time complexity. Ad hoc algorithms are instead designed for other basic problems such as searching and sorting.

1. Introduction

In the classical RAM model of computation, the cost of a random access to the memory is generally assumed as independent of the memory size and no limitations are posed on the number of memory cells (registers). The assumption of constant access time is realistic if we consider a fast memory of fixed size, such as a random access memory, and a very large amount of slower (secondary) memory. However, the enormous evolution of the technology of integrated cir-

---

Received 31 March 1992.

(1) This work has been supported in part by the C.N.R. project «Sistemi Informatici e Calcolo Parallelo».

(2) Dipartimento di Informatica e Applicazioni, Università di Salerno, 84081 Baronissi, Salerno, Italy.

(3) Dipartimento di Informatica, Università di Pisa, Corso Italia 40, 56100 Pisa, Italy.

(4) Dipartimento di Informatica e Sistemistica, Università di Roma La Sapienza, Via Salaria 113, 00136 Roma, Italy.
circuits makes it possible to design random access memories of larger and larger size and it is no more possible, over a certain size, considering the access time to be constant.

From VLSI theory, we know that the logic gates have limited fan in - fan out and a constant time access becomes unacceptable as the memory size increases [6]. The access time log x seems to be correct, for a memory of size x, if we consider the above limitations.

Some recent papers have addressed this problem from a computational complexity point of view, by defining alternative models with different access time cost functions. In the models proposed in [1,2,3], the main memory is composed of two or more levels hierarchically organized, (e.g. cache and main memory), and an access to location i requires time log i or other cost functions such as f(i) = x^a, 0 < a ≤ 1. The overheads for the access time can be optimized by bringing data into fast memory and by using them several times before returning them in slower memory.

In [2] the model allows a «block transfer» (BT) where a memory block of size t, starting at location i, can be copied sequentially in time f(i)+t. Efficient sequential algorithms are designed for the models in [1,2] while parallel algorithms are considered in [3].

A different approach is taken in [5] where the memory is a single bank of size x and the extra access time is log x for all memory cells; pipelining is also allowed in the access channels (LPM model).

In our work, we follow this last approach for the memory organization; we introduce a Computational Memory Model, CMM, where the access time is either log x or requires constant time, if the memory cell is adjacent (preceding) to the one already accessed. Our approach is different from the «BT-model» [2] since both the memory organization and the access cost function does not depend on the memory address. It also differs from the one proposed in [5] since the constant access is realized in a very simple way by slightly modifying the boolean circuits of the memory, and does not require complex synchronizations to allow pipeline technics.

From a theoretical point of view, we shall see that our model, in the proposed realization, is a restriction of the LPM: pipelining is only allowed to consecutive cells. As a consequence we have that lower bounds valid for the LPM also hold here, while algorithms designed for the CMM (upper bounds) can be used in the LPM, with obvious modifications.

Complexity results found in [1,2] are not valid for CMM. In our model we assume that the number of internal registers of the RAM processor is fixed and