Information Processing Letters 16 (1983) 1-2
North-Holland Publishing Company

24 January 1983

INFINITE CUBE-CONNECTED CYCLES *

Friedhelm MEYER AUF DER HEIDE

Johann Wolfgang Goethe-Universität Frankfurt, Fachbereich Informatik, 6000 Frankfurt a.M., W. Germany

Communicated by H.R. Wiehle
Received 3 June 1982
Revised 1 October 1982

An infinite network for parallel computation is presented which can for every k become partitioned in cube-connected cycles-networks of size $2^{2^k}2^k$ [1]. This construction extends a result from [2], where finite such networks are constructed. This infinite network is useful for simplifying the structure and improving the efficiency of the general purpose parallel computer shown in [3].

Keyword: Parallel computation

Introduction

Preparata and Vuillemin introduced in [1] the cube-connected cycles (CCC), a network for parallel computation, which can simulate a lot of important parallel computations as sorting, permuting and fast Fourier transformation with a constant loss of time but with a saving of processors by a factor proportional to the depth of the network being simulated: log n for permuting and fast Fourier transformation, log n^2 for sorting.

Galil and Paul [3] used the CCC's to construct an infinite general purpose parallel computer which needs O(t log p^2) steps and O(p) processors to simulate an infinite parallel computer which started with some input, executes t steps and uses p of its processors.

The key of this simulation is the use of a 'post office', i.e., a network in which the communication between the processors of M is simulated. This 'post office' has to be an infinite sorting network on which for every p it is possible to sort p items written in the first p input-processors of the network in O(log p^2) steps using O(p) processors. In [3] this is realized by connecting CCC's of all sizes by trees. The items have to be transported to the CCC of suitable size in order to be sorted.

We shall now describe an infinite network with the following properties:

(1) Each processor can communicate with 4 other processors.

(2) For every k the network can become partitioned in CCC's of size $2^{2^k}2^k$. (In fact these CCC's contain still some more edges and form the network constructed in [2].)

(3) For $k' < k$ each CCC in the network of size $2^{2^k}2^k$ can be partitioned in CCC's of size $2^{2^{k'}}2^{k'}$.

This network allows us to construct a very simple 'post office' for the general purpose computer from [3]. Additionally we save more than a half of the processors used for a simulation and the time necessary to transport the items to the suitable CCC.

Construction of the infinite cube-connected cycles

The CCC-network has $2^{2^k}2^k$ vertices and degree 3. We use a description of this graph tailored to our construction. For a positive integer k let I_k be

* The Publisher is glad to announce that this article is the 1000th paper to appear in *Information Processing Letters* since its birth in February 1971.

the set

$$\{-2^{k-1},\ldots,-2,\,-1,\,1,\,2,\ldots,2^{k-1}\}.$$

Then the vertex set of G_k is $\{0, 1\}^{I_k} \times I_k$.

The edge set consists of cycle-edges and cube-edges.

Two vertices $(\bar{a}, b)$ and $(\bar{a}', b')$, $b \geqslant b'$ are joint by a cycle-edge if $\bar{a} = \bar{a}'$ and $b = b' + 1$ or $b = 1$, $b' = -1$ or $b = 2^{k-1}$, $b' = -2^{k-1}$.

They are joint by a cube-edge, if $b = b'$ and $\bar{a}$ and $\bar{a}'$ differ exactly at the b*th* position.

A CCC network is obtained by attaching processors to the vertices of the graph. The edges then determine which processors may communicate in one step.

Our infinite network has the following infinite graph G: Let A be the set of all positive and negative integers except zero, and B the subset of $\{0, 1\}^A$ of those sequences which only contain a finite number of 1's. Then the vertex set V of G is $B \times A$. (Clearly V is countable.)

The cycle-edges and cube-edges are defined as above for G_k, except the cycle-edges with $b = 2^{k-1}$ and $b' = -2^{k-1}$, which do not appear as cycle-edges here.

Additionally we now insert new edges, the cross-edges, between vertices $(\bar{a}, b)$ and $(\bar{a}', b')$, if $\bar{a} = \bar{a}'$ and $b = -b'$.

Fig. 1 shows the vertices of G for some fixed $\bar{a} \in B$ with cycle- and cross-edges.

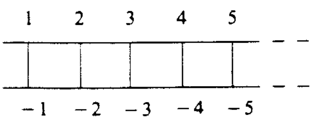

Fig. 1.

Now for some positive integer x let D_x^k be the set

$$\{-x,\,-x-1,\ldots,-x-2^{k-1}+1\} \cup$$
$$\cup\,\{x,\,x+1,\ldots,x+2^{k-1}-1\}.$$

Let $\bar{c} = (c_i)_{i \in A} \in B$ be a sequence with $c_i = 0$ for all $i \notin D_x^k$. Then consider the subgraph $G(\bar{c}, x, k)$ of G with vertex set

$$\{(\bar{a}, b) \in V,\ a_i = c_i \text{ for all } i \notin D_x^k,\ b \in D_x^k\}.$$

If we now remove all cross-edges between vertices $(\bar{a}, b)$, $(\bar{a}, -b)$ in this graph with $b \notin \{x, x + 2^{k-1} - 1\}$, we obviously obtain a graph isomorphic to G_k. Now let $x = 2^{k-1}p$ for some non-negative integer p. Then the vertices of the graphs $G(\bar{c}, x, k)$ with some x of the above form and all $\bar{c}$ as above partition G into isomorphic copies of G_k. If we do not remove the cross-edges, the above vertex-sets define graphs as constructed in [2], but with a new representation of their vertices. The third property also follows directly from this partition.

References

[1] F.P. Preparata and J. Vuillemin, The cube-connected cycles: A versatile network for parallel computation, Proc. 20th IEEE-FOCS, Puerto Rico (1979) pp. 140–147.

[2] F. Meyer auf der Heide, Efficiency of universal parallel computers, Intern. Bericht des Fachbereichs Informatik der Johann Wolfgang Goethe-Universität Frankfurt; Acta Inform. (1982) submitted.

[3] Z. Galil and W.J. Paul, A theory of complexity of parallel computation, Proc. 13th ACM-STOC, Milwaukee (1981) pp. 247–262.