A MIMD type highly parallel processor comprising 4096 processing elements (PEs) with a nearest neighbor mesh connection is studied. The system realizes more than 100MB/S initial data transfer capability by multi-layering PE arrays, transmitting data from each upper layer PE to dependent lower layer PEs simultaneously. This configuration reduces the maximum internode distance and the inter-PE data transfer delay by relaying inter-PE data via upper layer PEs. High speed inter-PE synchronizations, for instance, synchronization of all PEs and local synchronization within any layer, have been realized (less than one microsecond for all PEs). A small scale system with 256 PEs is now under fabrication. Each PE consists of a 16-bit micro-processor, DRAMs and two newly developed types of LSIs. The size of a PE is 9cm x 6cm x 3cm.

2. System Architecture

In order to cope with the inherent data transfer problems, we adopted a hierarchical PE array structure similar to the EGPA [8], namely a large scale array with a small scale array above it. The number of PEs in the smaller array is approximately the square root of the number in the large one. By making use of the small array to accomplish data transfer to and from the large PE array and between PEs in the array, realization of a data transfer rate that matches the large array's processing power and reduction of the inter-PE data transfer delay are attempted. Hence, even if there is an increase in the number of PEs, a high performance system that is not limited by data transfer capability can be realized.

2.1 System Configuration

In the HAP, multiple users are assumed in order to make full use of the processing power of all the PEs. It is most properly used as the back-end processor for a number of user computers. Figure 1 shows the system configuration of the HAP. It consists of a PE array, a control PE array and a system management processor that are hierarchically coupled, and a data I/O mechanism.
1) Configurations and Roles of Each Block

The PE array consists of a maximum of 4096 (64 x 64) PEs, and it can execute parallel tasks. The control PE array (cPE array) has a maximum of 64 (8 x 8) control PEs (cPE). Together with the data I/O mechanism, it performs the input-output of data to the PE array. It also relays the inter-PE data transfers. Besides these, hierarchical parallel tasks can also be executed. A system management processor (SMP), using a general purpose computer, controls the whole system. The data I/O mechanism performs the input-output of data to user computers and cPES, and also data buffering.

2) Physical Connection among PEs

The PEs inside the PE array are physically NNM and torus-connected in consideration of the total data transfer capability/hardware quantity for connection between PEs and easy expansion to a physical structure. Therefore, each PE is connected with its four nearest neighbor (north, south, east, and west) PEs. Although torus-connection slightly increases the inter-PE wiring length, it has the merit of reducing the maximum internode distance to half, compared to that of only an NNM, and it is used. The cPE array also has the same connection scheme as the PE array. In inter-layer connection, such as the connection between PEs and cPES, called lower PEs and upper PEs, respectively, a bus is used to reduce the amount of connecting hardware. The connection between the cPE array and the SMP is the similar.

2.2 Data Transfer

In the operation of a parallel processor, the program and the required initial data are supplied to the PEs in the first phase. Then, tasks expanded in parallel are executed in the PEs with necessary data transfer between the PEs. Finally, the processed data are collected by the user computer. That is, there are usually four types of data transfer in a parallel processor, namely, program load, initial data supply, inter-PE data transfer and result collecting. Since system performance is evaluated by the total processing time, which includes these data transfers, improvement in its transfer capabilities increases the value of a parallel processor. In the HAP, rapid data transfer is realized through the following approach:

1) Program Load

Load distribution, instead of function distribution, is used to improve the performance in highly parallel processors. The programs for all PEs are basically identical, so they are broadcast to all cPES and PEs by the SMP. Moreover, independent program load to each PE is also possible, in consideration of occasions when the programs are different for part of the PEs. These are realized through memory accesses (write operations) to the lower PEs or PE by the upper PE.

2) Initial Data Supply and Result Collecting

Since the data to be processed by each PE is different, it is transferred from the SMP in serial fashion. This is a potential bottleneck in system performance. Therefore, in the HAP, data elements are supplied in parallel through the cPES. Specifically, the initial data buffered in the data I/O mechanism is transferred simultaneously to all the cPES, then each cPE transfers it to the PEs that are coupled to it. Result collecting is basically similar, except the direction of data flow is reversed. These data transfers are DMA transfers by the cPES.

3) Inter-PE Data Transfer

A parallel processor with NNM connection is very suitable for processing problems where inter-PE data transfer occurs locally or regularly (e.g., solving Poisson's equation by the ODD-EVEN SOR method) [4][9]. However, it is not suitable for problems where transfers occur irregularly between PEs with large internode distances, e.g., logic simulation. This is because the maximum internode distance is still as large as N, where N equals the number of PEs, in this kind of parallel processor even if torus connection is employed together with the NNM; therefore, much time is