Chapter 3

Architecture of NETRA

This chapter contains a detailed description of the architecture of NETRA. The first four sections describe the components of NETRA, their functions, capabilities and features. The last section critically examines the architecture in view of the computational requirements for IVS developed in the previous chapter.

Figure 3.1 shows the architecture of "NETRA," which is a recursively defined hierarchical multiprocessor system and provides distributed as well as shared memory environment. The architecture consists of the following components:

1. A large number (1000 - 10000) of Processing Elements (PEs), organized into clusters of 16 to 64 PEs each.
2. A tree of Distributing-and-Scheduling-Processors (DSPs) that make up the task distribution and control structure of the multiprocessor.
3. A parallel pipelined shared Global Memory and a Global Interconnection that links the PEs and DSPs to the Global Memory.

3.1. Processor Clusters

The clusters consist of 16 to 64 PEs, each with its own program and data memory. Each PE is a general purpose processor with a high speed floating point capability. They form a layer below the DSP-tree, with a leaf DSP associated with each cluster. PEs within a cluster also share a common data memory. The PEs, the DSP associated with the cluster, and the shared memory are connected together with a crossbar switch. The crossbar switch permits point-to-point communications as well as selective broadcast by the DSP or any of the PEs. Figure 3.2 shows the cluster organization. A 4x4 crossbar is shown as an example of the implementation of the crossbar switch. The crossbar design consists of pass transistors connecting the input and output data lines. The switches are controlled by control bits indicating the connection pattern. If a processor or DSP needs to broadcast, then all the
control bits in its row are made one. In order to connect processor $P_i$ to processor $P_j$, control bit (i,j) is set to one, and the rest of the control bits in row i and column j are off.

Clusters can operate in SIMD mode, systolic mode, or MIMD mode. In an SIMD mode, PEs in a cluster execute identical instruction streams from private memories in a lock-step fashion. In systolic mode, PEs repetitively execute an instruction or set of instructions on data streams from one or more PEs. In both cases, communication between PEs is synchronous. In MIMD