Chapter 3

Matrix Multiplication using Hypercube Networks

3.1 Objectives:

- At the end of the lecture, the learner would be able to:
  - Understand the definition of k-dimensional hypercube.
  - Determine when two nodes in a hypercube network interact with each other.
  - Compute the sufficient number of routing steps to broadcast the initial values
  - Analyze how Bit Complement funciton is used to compute the product of two matrices.

3.2 Cube Connected Network

Now let us start with the definition of a cube connected network.

Definition 1. *Cube Connected Network: It is also called as a binary n-cube network consists of* 2^k *nodes that forms a k-dimensional hypercube. The nodes are named from 0 to* 2^k*-1. Two nodes* n_1 *and* n_2 *are said to be adjacent iff their labels differ in exactly one bit position.*

Example of a Hypercube Network

3.2.1 Some Important Features of Hypercubes

1. The diameter of a hypercube with 2^k nodes = k
2. The bisection width of a hypercube with 2^k nodes = 2^{k-1}
3. Number of edges=k
4. BitComplement(n,k): It is used to complement the kth bit position of an integer n. Example:BitComplement(4,1)=BitComp
5. nCUBE is building systems based on hypercube topology.
6. Connection Machine-200(CM-200) are connected in a hypercube.

Figure 3.1: A Four Dimensional Hypercube

Now let us analyze how to route messages in a hypercube.
Two nodes whose name differ in exactly one bit position are always linked by edges.
For example 0101 will be connected with 0001,1101,0111 and 0100. Using this one can identify the shortest path between source and destination. The shortest path to reach 0011 from 0101 is via 0001.

3.2.2 Matrix Multiplication on the HyperCube

- Let us consider the following specification for a hypercube interconnection network
  - Let N= 2^k be the total number of processors. Let it be P_0,P_1,etc upto P_{N-1}.
  - Let j and j^b be two integers ranging from 0 to N-2 whose binary representation differ only in position b, where b ranges from 0 to k-1.
  - Let us compute the product of the following two matrices.

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

$$B = \begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix}$$

  - Initial Step 1: Let the elements of A and B are arranged in a hypercube as shown in Figure 2.
  - Step 2: The nodes labelled 000,001,010 and 011 will send their data to its neighbours labelled 100,101,110 and 111 respectively. That is A(0,j,k) and B(0,j,k) will send the data to the Processing Element(i,j,k). This is shown in Figure 3.
  - Step 3: The processing elements with identical first and third bits will send the elements of A to their neighbours. That is 101 will send to 100, 000 will send to 001, 010 will send to 011 and 111 will send to 110 respectively. This is shown in Figure 4.
  - Step 4: The Processing elements with identical first and second bits will send the elements of B to their neighbours. That is 110 will send to 100, 111 will send to 101, 000 will send to 010 and 001 will send to 011 respectively. This is shown in Figure 5.

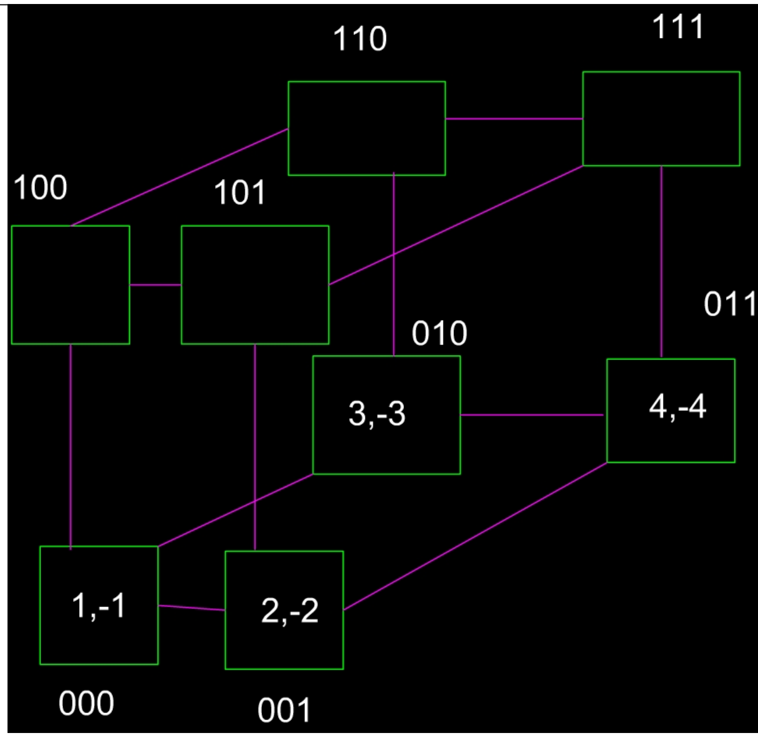

Figure 3.2: Step 1: An arrangement of matrix elements in a hypercube

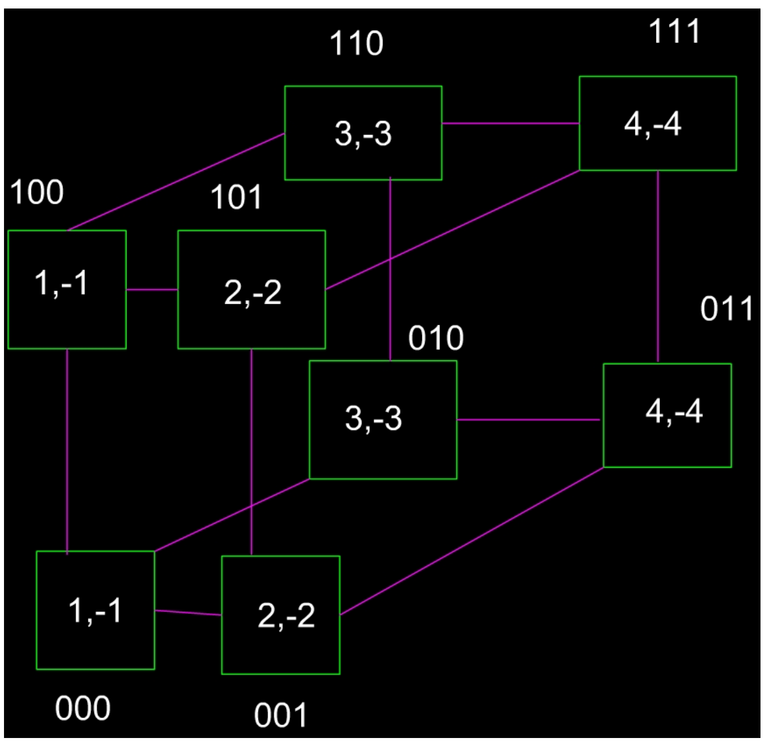

Figure 3.3: Step 2: A data routing in a hypercube

- Step 5:Each Processing Element $P_{i,j,k}$ will compute the product of its elements. This is shown in Figure 6.

- Step 6: The Processing Elements $P_{i,j,k}$ that differ in the Most significant bit will sum up their elements and the result will be stored in the Processing Element $P_{0,j,k}$. This is shown in Figure

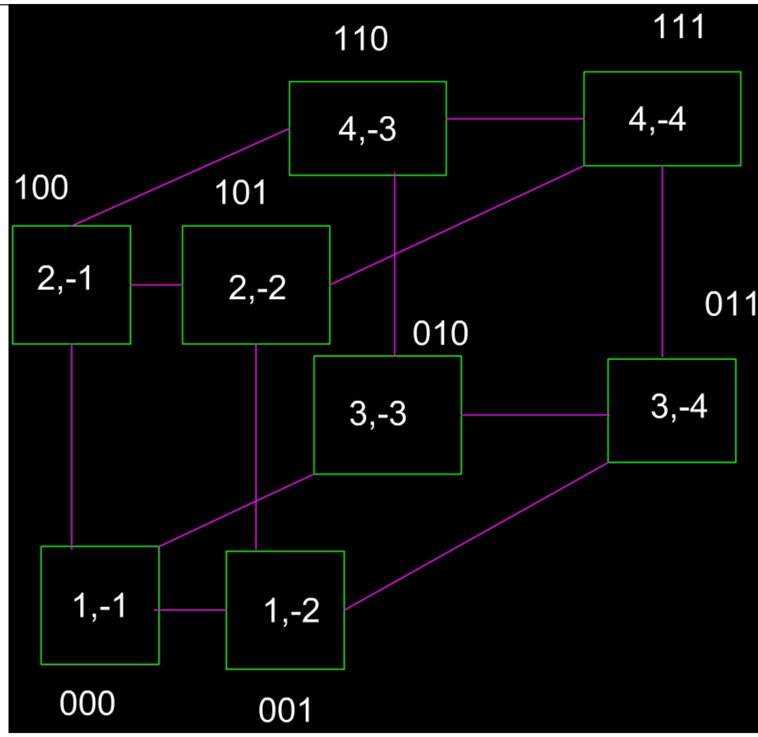

Figure 3.4: Step 3: A data routing in a hypercube

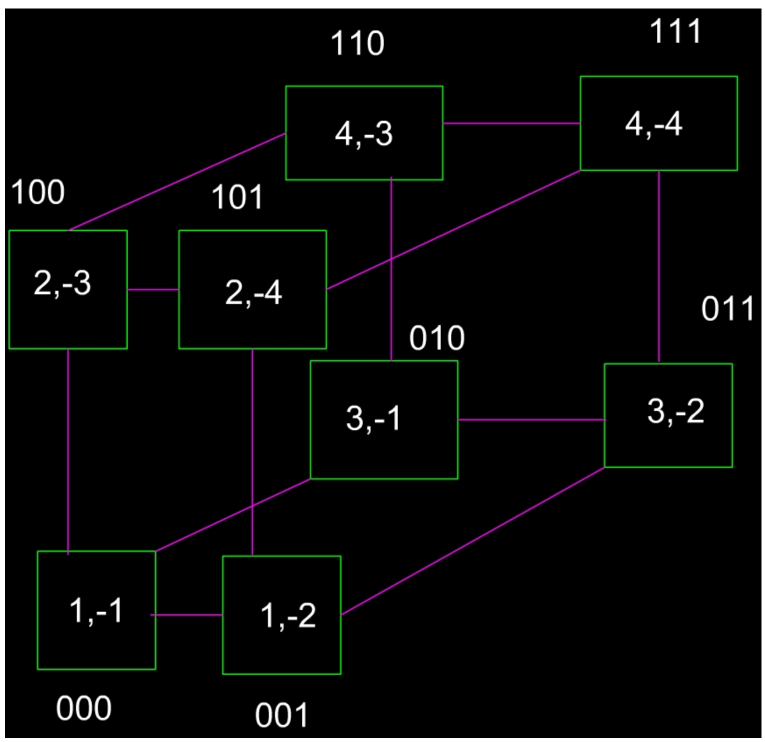

Figure 3.5: Step 4: A data routing in a hypercube

7.

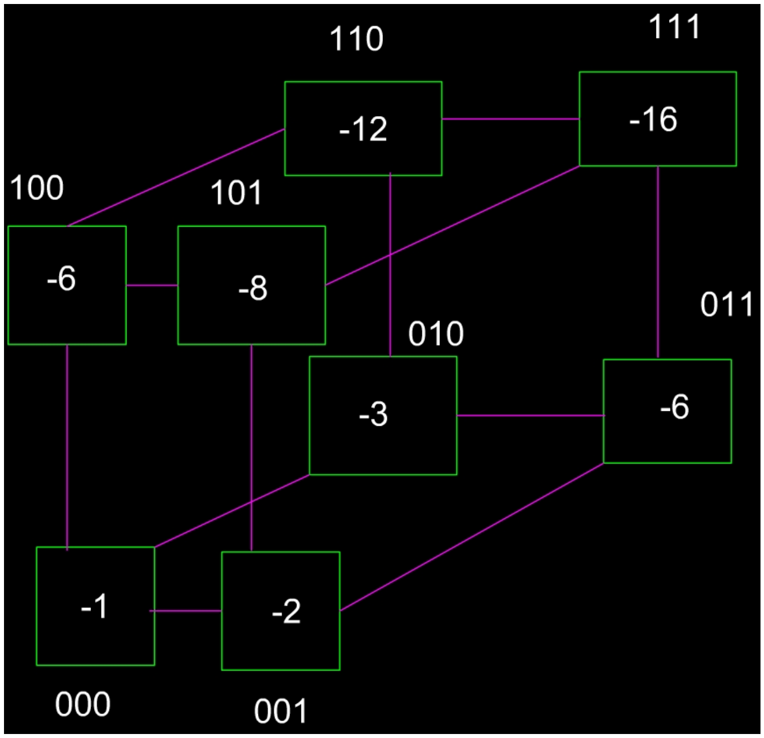

Figure 3.6: Step 5: Computation of the product

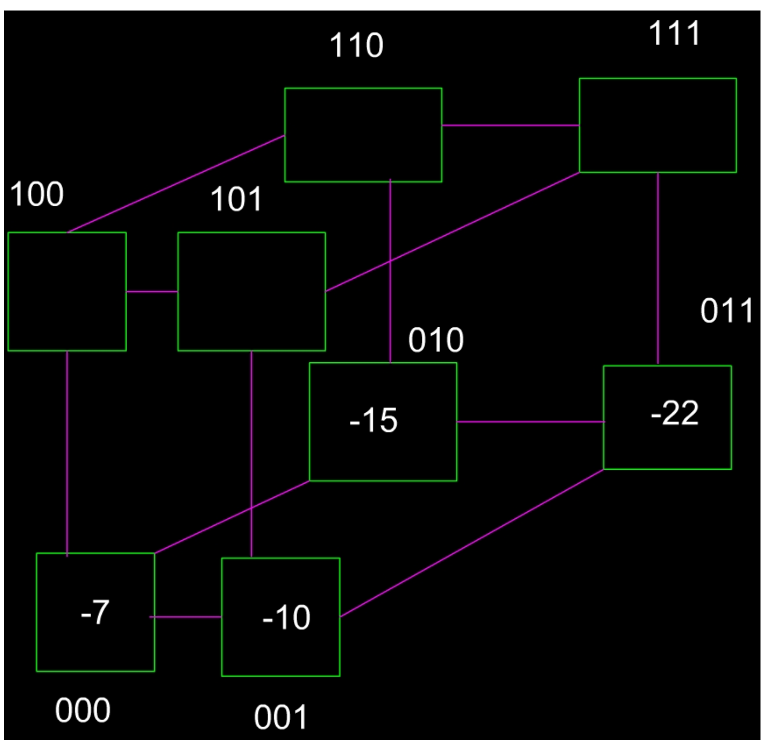

Figure 3.7: Step 6: Summing up of the elements