Module-2
CSEN 3104
Lecture 16

Dr. Debranjan Sarkar

Interconnection Networks

Mesh-connected Illiac Network

- Show figure
- Single stage recirculating network with N = 64 PEs
- In one circulation step, each PE_i is allowed to send data to any one of PE_{i+1}, PE_{i-1} , PE_{i+r} and PE_{i-r}
- In practice, N is a perfect square and r = sqrt(N). For Illiac IV, r = 8
- The interconnection network of Illiac IV is characterized by the following four routing functions:
  - $R_{+1}(i) = (i+1) \bmod N$
  - $R_{-1}(i) = (i-1) \bmod N$
  - $R_{+r}(i) = (i+r) \bmod N$
  - $R_{-r}(i) = (i-r) \bmod N$

Where $0 \leq i \leq (N-1)$

Mesh-connected Illiac Network

- In the reduced Illiac network, the outputs of IS_i are connected to the inputs of OS_j for j = i+1, i-1, i+r, and i-r
- In other words, OS_j gets its inputs from IS_i for i = j-1, j+1, j-r, and j+r respectively
- Each PE_i is directly connected to its four nearest neighbours in the mesh network
- Permutation Cycle (a b c) (d e) represents the permutation a→b, b→c, c→a and d→e, e→d in a circular fashion within each pair of parentheses
- We may write for Horizontal PEs

$$R_{+1} = (0\ 1\ 2\ \ldots\ N\text{-}2\ N\text{-}1)$$

$$R_{-1} = (N\text{-}1\ N\text{-}2\ \ldots\ 2\ 1\ 0)$$

- When the routing function is executed, data is routed as per above only if all PEs in the cycle are active

Mesh-connected Illiac Network

- For Vertical PEs, we may write

  $R_{+r} = \prod(i\ i{+}r\ i{+}2r\ \ldots.\ i{+}N{-}2r\ i{+}N{-}r)$, for r =0, 1, 2, …, r-1

  $R_{-r} = \prod(i{+}N{-}r\ i{+}N{-}2r\ \ldots.\ i{+}2r\ i{+}r\ i)$, for r =0, 1, 2, …, r-1

- In the reduced Illiac network,

  $R_{+4} = (0\ 4\ 8\ 12)\ (1\ 5\ 9\ 13)\ (2\ 6\ 10\ 14)\ (3\ 7\ 11\ 15)$

  $R_{-4} = (12\ 8\ 4\ 0)\ (13\ 9\ 5\ 1)\ (14\ 10\ 6\ 2)\ (15\ 11\ 7\ 3)$

- The cycle (2 6 10 14) in the above permutation R_{+4} will not be executed if one or more among PE_2, PE_6, PE_{10}, and PE_{14} is disabled by masking
- Generally speaking, when R_{+r} or R_{-r} is executed, data are permuted only if PE_{i+kr}, where $0 \leq k \leq r-1$ are active for each i

Mesh-connected Illiac Network

- The Illiac network is a partially connected network (Show figure)
- It is noted that
  - In one step, only 4 PEs can be reached from any PE

    PE_0 to PE_1, PE_4, PE_{12} or PE_{15}
  - In two steps, 7 PEs can be reached from any PE

    PE_0 to PE_2, PE_3, PE_5, PE_8, PE_{11}, PE_{13} or PE_{14}
  - In three steps, 11 PEs can be reached from any PE
  - Processing elements PE_6, PE_7, PE_9 or PE_{10} can be reached from PE_0 in the worst case of three steps
- It takes I steps (recirculations) to route data from PE_i to any other PE_j, in an Illiac network of size N, where I is upper-bounded by

  $$I \leq \text{sqrt}(N) - 1$$

  At most sqrt(64) – 1 = 7 steps are needed in Illiac IV to route data from any one PE to another PE
- If connectivity is increased, the upper bound can be lowered

Single Stage Dynamic Network

- Also called recirculating network
- Data items may have to recirculate through the single stage several times before reaching their final destination
- The number of recirculations needed depends on the connectivity of the network
- Generally, the higher the connectivity, the less is the number of recirculations
- The crossbar network is an extreme case where only one circulation is needed to establish any connection path
- However it is very costly. A fully connected crossbar network has a cost of $O(N^2)$ which may be prohibitive for large N
- Most recirculating networks have cost O(NlogN) or lower, which is much more cost-effective for large N

Multistage Dynamic Networks

- Many stages of interconnected switches form a multistage network
  - Described by 3 characterizing features
    - Switch box
    - Network topology
    - Control structure
  - Many switch boxes are used
  - Each switch box is basically an interchange device with 2 inputs and 2 outputs (show figure)
  - A two function switch box can be in any of the following states
    - Straight
    - Exchange
  - A four function switch box can be in any of the following states
    - Straight
    - Exchange
    - Upper broadcast
    - Lower broadcast

Multistage Networks

- Capable of connecting an arbitrary input terminal to an arbitrary output terminal
- Consists of n stages where $N = 2^n$ is the number of input and output lines
- Each stage may use N/2 switch boxes
- Each stage is connected to the next stage by at least N paths
- The interconnection patterns from stage to stage determine the network topology
- The network delay is proportional to the number of stages
- The control structure can be
  - Individual stage control
    - uses the same control signal to set all switch boxes in the same stage.
    - Requires n sets of control signals
  - Individual box control
    - Separate control signal is used to set the state of each switch box
    - High flexibility but more cost
  - Partial stage control
    - Compromise design

Cube Interconnection Networks

- Can be implemented as a
  - single-stage (or recirculating) network
  - Multistage network
- Show figure of a 3-dimensional cube
- Horizontal lines (parallel to x-axis) connect vertices (PEs) whose addresses differ in the LSB position
- Vertical lines (parallel to y-axis) connect vertices (PEs) whose addresses differ in the MSB position
- Lines parallel to z-axis connect vertices (PEs) whose addresses differ in the middle bit position

Cube Interconnection Networks

- The unit cube concept may be extended to an n-dimensional unit space, called n-cube with n-bit address for each vertex
- In the n-cube, each PE located at a corner is directly connected to its n neighbours
- The addresses of the neighbouring PEs differ in one bit position only
- The routing functions of an n-dimensional cube network are given by:

$$C_i(a_{n-1} \ldots\ldots a_1 a_0) = a_{n-1} \ldots\ldots a_{i+1} a_i' a_{i-1} \ldots\ldots a_1 a_0 \qquad \text{for } i = 0,1,2, \ldots\ldots, n-1$$

- Examples of cube network
  - Pease's binary n-cube
  - The flip network
  - Programmable switching network proposed for the Phoenix project

Cube Interconnection Networks

- In the recirculating (single stage) cube network, each IS_A (for $0 \le A \le N-1$) is connected to n OSs whose addresses are $a_{n-1}a_{n-2} \ldots\ldots a_{i+1}a_i{'}a_{i-1} \ldots\ldots a_1a_0$ (for $0 \le i \le n-1$)
- Similarly, each OS_T with $T = t_{n-1}t_{n-2} \ldots\ldots t_1t_0$ gets its inputs from ISs whose addresses are $t_{n-1}t_{n-2} \ldots\ldots t_{i+1}t_i{'}t_{i-1} \ldots\ldots t_1t_0$ (for $0 \le i \le n-1$)
- The n routing functions are given by:

$$C_i(a_{n-1}a_{n-2} \ldots\ldots a_1a_0) = a_{n-1}a_{n-2} \ldots\ldots a_{i+1}a_i{'}a_{i-1} \ldots\ldots a_1a_0 \text{ (for } 0 \le i \le n-1)$$

- Show the diagram of recirculating cube network
- C_0 = (0,1) (2,3) (4,5) (6,7)
- C_1 = (0,2) (1,3) (4,6) (5,7)
- C_2 = (0,4) (1,5) (2,6) (3,7)
- If all the 3 connecting patterns are assembled together, the 3-cube is obtained

Cube Interconnection Networks

- The same set of cube-routing functions may be implemented by a three-stage cube network (Show figure)
- Two-function (Straight and Exchange) switch boxes are used
- Stage i implements the C_i routing function
- This means the switch boxes at stage i connect an input line to the output line that differs from it only at the i^{th} bit position
- Individual box control is assumed in a multistage cube network
- Show the path between a source and a destination
- Supports up to *N* one-to-one simultaneous connections
- There may be some permutations which cannot be established
- Also supports one-to-many connections; that is, an input device can broadcast to all or a subset of the output devices (Show figure)

Cube Interconnection Networks

- We may note that the permutation (0,1) (0,2) (0,4) is performed only if the top row boxes (a, e, i) are set to exchange and the rest are set to straight
- Masking may change the data-routing patterns
- General practice is to disable all the PEs belonging to the same cycle of a permutation
- In case of P_2 = (0,4) (1,5) (2,6) (3,7), if both PE_2 and PE_6 become inactive by masking, the cycles (2,6) are removed and the cube-routing function C_2 performs only the partial permutation (0,4) (1,5) (3,7)
- If only PE_2 is disabled,
  - the above partial permutation will still be performed
  - Data in both PE_2 and PE_6 will be transferred to PE_2
  - PE_6 will not receive any data
- Masking should be carefully applied to cube networks

Thank you