Module-2
CSEN 3104
Lecture 15

Dr. Debranjan Sarkar

Interconnection Networks

Inter-PE Communication

- The design of the architecture of an interconnection network for an SIMD machine is based on
  - Operation modes
  - Control strategies
  - Switching methodologies
  - Network topologies

Operation Modes of Interconnection Network

- Synchronous mode
  - To establish synchronous communication path for
    - Data manipulating function
    - Data instruction broadcast
- Asynchronous mode
  - To have asynchronous communication when
    - Connection requests are issued dynamically
- Combined mode
  - To facilitate both synchronous and asynchronous processing
- SIMD machines work in synchronous operation mode, where lock-step operations among all PEs are enforced

Control Strategy

- Interconnection network consists of
  - A number of switching elements, and
  - Interconnecting links
- Interconnection functions are realized by properly setting control of the switching elements
- Two types of control
  - Centralized control -> the control setting function is managed by a centralized controller
  - Distributed control -> the control setting function is managed by the individual switching element
- SIMD machines have interconnection networks with centralized control on all switching elements by the control unit

Switching methodology

- Circuit switching
  - Physical path is actually established between a source and a destination
  - Much more suitable for bulk data transmission
- Packet switching
  - Data is put in a packet and routed through the interconnection network
  - No physical connection is established
  - More efficient for many short data messages
- Integrated switching
  - Includes the capabilities of both circuit switching and packet switching
- SIMD interconnection networks assume circuit switching

Network topology

- A network is depicted by a graph in which nodes represent switching points and edges represent communication links
- Static topology
  - Links between two processors are passive
  - Dedicated buses cannot be reconfigured for direct connections to other processors
- Dynamic topology
  - Can be reconfigured by setting the network's active switching elements
- SIMD interconnection networks are classified into two categories, based on network topologies:
  - Static networks
  - Dynamic networks

Static Networks

- Topologies in the static networks are classified according to the dimensions required for the layout
    - One-dimensional (Linear array)
    - Two dimensional (Ring, star, tree, mesh and systolic array)
    - Three dimensional (Completely connected, Chordal ring, 3-cube, 3-cube-connected cycle)
    - Hypercube
- Show the diagrams

Dynamic Networks

- Single stage dynamic network (show figure)
  - N number of input selectors (IS) and N output selectors (OS)
  - Each IS is essentially a 1-to-D demultiplexer ($1 \leq D \leq N$)
  - Each OS is essentially an M-to-1 multiplexer ($1 \leq M \leq N$)
  - To establish a desired path, different control signals are applied to all IS and OS selectors
  - Data items may have to recirculate through the single stage several times to reach the final destination
  - The higher is the hardware connectivity, the less is the number of recirculations required
  - Crossbar-switching network is a single-stage n/w with D=M=N
  - Crossbar is an extreme case in which only one circulation is required for any path

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

Thank you

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

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

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

- 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 $0 \le k \le r-1$ 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 $a_{n-1}a_{n-2}$ .....$a_{i+1}a_i{}'a_{i-1}$ ..... a_1a_0 (for $0 \le i \le n-1$)
- Similarly, each OS_T with $T = t_{n-1}t_{n-2}$ ..... t_1t_0 gets its inputs from ISs whose addresses are $t_{n-1}t_{n-2}$ .....$t_{i+1}t_i{}'t_{i-1}$ ..... t_1t_0 (for $0 \le i \le n-1$)
- The n routing functions are given by:

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

- 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

Module-2
CSEN 3104
Lecture 17

Dr. Debranjan Sarkar

Interconnection Networks

Shuffle-Exchange and Omega Networks

- Based on two routing functions --- Shuffle (S) and Exchange (E)
- Let A = a_{n-1} …. a_1a_0 be the address of a Processing Element (PE)
- The Shuffle function is given by

  $S(a_{n-1} \ldots a_1a_0) = a_{n-2} \ldots a_1a_0\, a_{n-1}$ where $0 \le A \le (N-1)$ and $n = \log_2 N$
- Corresponds to cyclic shifting of the bits in A to the left for 1 bit position
- Show figure for perfect shuffle
- This action corresponds to perfect shuffling a deck of N cards
- The inverse perfect shuffle does the opposite to restore ordering (Show figure)
- Corresponds to cyclic shifting of the bits in A to the right for 1 bit position
- The Exchange function is given by

  $E(a_{n-1} \ldots a_1a_0) = a_{n-1} \ldots a_1a_0'$
- The Exchange function exchanges the data between two PEs with adjacent addresses
- It is to be noted that $E(A) = C_0(A)$, where C_0 was the cube routing function

Shuffle-Exchange and Omega Networks

- The Shuffle-Exchange function can be implemented as
  - Single stage network
  - Multistage network
- Single Stage recirculating shuffle-exchange network (Show figure)
- Dashed lines -> Shuffle Solid lines -> Exchange
- A number of parallel algorithms can be effectively implemented by using Shuffle-Exchange function. Examples:
  - Fast Fourier Transform (FFT)
  - Polynomial Evaluation
  - Sorting
  - Matrix Transposition etc…

Multistage Omega Networks

- To implement Shuffle-Exchange functions (Show figure)
- An N X N Omega network consists of n (= $\log_2 N$) identical stages
- Perfect shuffle interconnection between two adjacent stages
- Each stage has N/2 numbers of 4-function (straight, exchange, upper broadcast and lower broadcast) switch boxes under independent box control
- The switch boxes can be repositioned without violating the perfect shuffle interconnection between stages (Show figure)
- The n-cube network has the same interconnection topology as the repositioned Omega
- However, they are different in the following two points:
  - Cube NW uses 2-function switch boxes, whereas Omega NW uses 4-function ones
  - The dataflow directions in the two NWs are opposite to each other i.e. the roles of the input-output lines are exchanged in the two networks

Routing Algorithm for Omega Network

- A source *S* (with address $s_{n-1}\ s_{n-2}\ \ldots\ s_0$) has to be connected to a certain destination *D* (with address $d_{n-1}\ d_{n-2}\ \ldots\ d_0$)
- Starting at input *S*, connect the input of the first switch [in the $(n\text{-}1)^{th}$ stage] that is connected to *S* to
  - the upper output of the switch when $d_{n-1}= 0$
  - otherwise, to the lower output
- In the same way, bit d_{n-2} determines the output of the switch located on the next stage
- This process continues until a path is established between *S* and *D*
- In general, the input of the switch on the i^{th} stage is connected to the upper output when $d_i = 0$; Otherwise, the switch is connected to the lower output
- Example: Source 2 (i.e., *S* = 010) and destination 6 (i.e., *D* =110) (Show Figure)
- In addition to one-to-one connections, the omega network also supports broadcasting
- Show Figure to explain the paths between source 2 and destinations 4,5,6 and 7

Thank you

Module-2
CSEN 3104
Lecture 18
20/08/2019

Dr. Debranjan Sarkar

Interconnection Networks

Shuffle-Exchange and Omega Networks

- Based on two routing functions --- Shuffle (S) and Exchange (E)
- Let A = a_{n-1} …. a_1a_0 be the address of a Processing Element (PE)
- The Shuffle function is given by

  $S(a_{n-1} \ldots a_1a_0) = a_{n-2} \ldots a_1a_0\, a_{n-1}$ where $0 \le A \le (N-1)$ and $n = \log_2 N$
- Corresponds to cyclic shifting of the bits in A to the left for 1 bit position
- Show figure for perfect shuffle
- This action corresponds to perfect shuffling a deck of N cards
- The inverse perfect shuffle does the opposite to restore ordering (Show figure)
- Corresponds to cyclic shifting of the bits in A to the right for 1 bit position
- The Exchange function is given by

  $E(a_{n-1} \ldots a_1a_0) = a_{n-1} \ldots a_1a_0'$
- The Exchange function exchanges the data between two PEs with adjacent addresses
- It is to be noted that $E(A) = C_0(A)$, where C_0 was the cube routing function

Shuffle-Exchange and Omega Networks

- The Shuffle-Exchange function can be implemented as
  - Single stage network
  - Multistage network
- Single Stage recirculating shuffle-exchange network (Show figure)
- Dashed lines -> Shuffle Solid lines -> Exchange
- A number of parallel algorithms can be effectively implemented by using Shuffle-Exchange function. Examples:
  - Fast Fourier Transform (FFT)
  - Polynomial Evaluation
  - Sorting
  - Matrix Transposition etc…

Multistage Omega Networks

- To implement Shuffle-Exchange functions (Show figure)
- An N X N Omega network consists of n (= $\log_2 N$) identical stages
- Perfect shuffle interconnection between two adjacent stages
- Each stage has N/2 numbers of 4-function (straight, exchange, upper broadcast and lower broadcast) switch boxes under independent box control
- The switch boxes can be repositioned without violating the perfect shuffle interconnection between stages (Show figure)
- The n-cube network has the same interconnection topology as the repositioned Omega
- However, they are different in the following two points:
  - Cube NW uses 2-function switch boxes, whereas Omega NW uses 4-function ones
  - The dataflow directions in the two NWs are opposite to each other i.e. the roles of the input-output lines are exchanged in the two networks

Routing Algorithm for Omega Network

- A source *S* (with address $s_{n-1}\ s_{n-2}\ \ldots\ s_0$) has to be connected to a certain destination *D* (with address $d_{n-1}\ d_{n-2}\ \ldots\ d_0$)
- Starting at input *S*, connect the input of the first switch [in the $(n\text{-}1)^{th}$ stage] that is connected to *S* to
  - the upper output of the switch when $d_{n-1}= 0$
  - otherwise, to the lower output
- In the same way, bit d_{n-2} determines the output of the switch located on the next stage
- This process continues until a path is established between *S* and *D*
- In general, the input of the switch on the i^{th} stage is connected to the upper output when $d_i = 0$; Otherwise, the switch is connected to the lower output
- Example: Source 2 (i.e., *S* = 010) and destination 6 (i.e., *D* =110) (Show Figure)
- In addition to one-to-one connections, the omega network also supports broadcasting
- Show Figure to explain the paths between source 2 and destinations 4,5,6 and 7

Omega Network (Blocking)

- Omega network is a blocking network
- Because some permutations cannot be established by the network
- For example, a permutation that requires
  - source 3 to be connected to destination 1, and
  - source 7 to be connected to destination 0
- This cannot be established (Show figure)
- However, such permutations can be established in several passes through the network
- For example, when node 3 is connected to node 1, node 7 can be connected to node 0 through node 4
- That is, node 7 sends its packet to node 4, and then node 4 sends the packet to node 0
- Therefore, we can connect node 3 to node 1 in one pass and node 7 to node 0 in two passes

Delta Network

- Recapitulation of Floor Function and Ceiling Function
- Floor(x) = Greatest integer ≤ x Floor(2.4) = 2
- Ceil(x) = Least integer ≥ x Ceil(2.4) = 3
- Mathematical definition of q-shuffle of qc objects (denoted by S_{q*c}):
- $S_{q*c}(i) = (qi + Floor(i/c)) \bmod qc$ for $0 \le i \le qc-1$
- Alternatively, $S_{q*c}(i) = qi \bmod (qc-1)$ for $0 \le i < qc-1$
  $= i$ for $i = qc-1$
- Show diagram of a 4-shuffle of 12 indices viz. S_{4*3}
- 2-shuffle is basically the well known perfect shuffle, discussed earlier

Construction of a^n X b^n Delta Network

- An a^n X b^n delta network has a^n sources and b^n destinations
- There are n stages consisting of a X b crossbar modules
- a-shuffle is used as the link pattern between every two consecutive stages
- Numbering of the stages is done as 1, 2, …., n starting at the source side
- a^{n-1} crossbar modules are required in the first stage
- The first stage has $a^{n-1}b$ output terminals and so the second stage must have $a^{n-1}b$ input terminals
- So stage 2 requires $a^{n-2}b$ crossbar modules
- The i^{th} stage has $a^{n-i}b^{i-1}$ crossbar modules of size a X b
- Thus the total number of a X b crossbar modules required in an a^n X b^n delta network can be found as:

$(a^n - b^n)/ (a – b)$ for $a \neq b$

and, $nb^{n-1} = na^{n-1}$ for $a = b$

Construction of a^n X b^n Delta Network

- The stages are interconnected in such a fashion that there exists a unique path of constant length from any source to any destination
- The path is digit controlled such that a crossbar module connects an input to one of its b outputs depending on a single base-b digit taken from the destination address
- If the destination D is expressed in a base-b system as $(d_{n-1}d_{n-2} \ldots d_1d_0)_b$, where $D = d_0b^0 + d_1b^1 + \ldots\ldots + d_{n-1}b^{n-1}$ and $0 \leq d_i < b$, then the base-b digit d_i controls the crossbar modules of stage (n-i)
- No input or output terminal of any crossbar module is left unconnected
- Show the diagram of 4^2 X 3^2, 2^3 X 2^3 and a^n X b^n delta network

Thank you