Unbuffered Multistage Interconnection Networks

Jon Turner
Computer Science & Engineering
Washington University

www.arl.wustl.edu/~jst
Contend and Repeat Networks

- Switch elements resolve conflicts by discarding cells
- Cells that reach their destinations are acknowledged using reverse ack path through network
- Inputs re-send cells that are not acked
- With Benes net, first $k-1$ stages distribute cells, last $k$ route using destination address
- Performance for Benes network can be estimated
  - let $p_i$—prob that cell is present at output of a stage $i$ switch; so $p_0$ is offered load, $p_{2k-1}$ is carried load and $p_{i+1} = 1 - (1 - p_i/d)^d$ for $i \geq k$ and $p_{i+1} = p_i$ for $i < k$
  - for $n=4096, d=16, p=0.4$, so need speedup of 2.5
  - analysis is optimistic; neglects correlations caused by resubmission of losers
Deflection Routing

- In deflection routing networks, cells get multiple chances to reach desired output
- Tandem banyan network
  - multiple banyans in series with outputs of each connected to output queues; cells get several chances to reach outputs
  - performance analysis similar to contend-and-repeat
  - can recirculate losers at last network if desired
Sorting Networks

- Bitonic sorter recursively merges sorted sublists
- Can switch by sorting on destination
  » additional components needed for conflict resolution
In bit-serial sorting element, sorting keys are compared bit by bit as header is received.
- Comparison starts with MSB, so first difference determines larger key.
- Single bit delay, simple circuit (about 50 gates).
- Upstream acknowledgement path used in some systems.
Batcher’s Bitonic Sorter

- Batcher’s network is constructed from subnetworks called bitonic merge networks and denoted $M_n$.
- $M_n$ has the property that given a bitonic sequence at its inputs, it will produce a sorted sequence at its outputs.
- Let $S = a_0, \ldots, a_{n-1}$ be a sequence of numbers and let $a_m$ have the smallest value of any in $S$. $S$ is bitonic if for some $i$ in $[0, n-1]$, $a_m \leq a_{m+1} \leq \cdots \leq a_{m+i}$ and $a_{m+i} \geq \cdots \geq a_{m+n-1}$, where addition is modulo $n$ (so if $S$ is graphed around the surface of a cylinder, there is one “peak” and one “valley”).
- To construct $S_m$, we take two copies of $S_{n/2}$ and connect them to a mrgc network $M_n$. 


Bitonic Merge and Sorting

- Bitonic merge network $M_n$ has same topology as banyan network $Y_{n,2}$
  - only difference is construction using sorting elements
- Since $M_n$ has $\log_2 n$ stages, $S_n$ has
  - $1 + 2 + \ldots + \log_2 n = (1/2)(\log_2 n)(1 + \log_2 n)$
    - requiring larger chip counts
      - for chips with 512 input pins and 512 outputs, need 7 ranks of chips for $n=4096$
- To use for switching, need to handle
  - multiple cells addressed to single output
  - copy congestion in multicast applications
Correctness of Merge Network

- Let $S = a_0, a_1, \ldots, a_n$ be a bitonic sequence input to $M_n$
- **Claim 1.** First stage correctly divides largest and smallest
  - because $S$ is bitonic, the $n/2$ largest values in $S$ are guaranteed to be consecutive (allowing for "wrap-around")
  - initial shuffle routes $n/2$ largest values to distinct first stage sorting elements, so these values go to lower recursive subnet
- **Claim 2.** Sequences sent to subnetworks are bitonic.
  - let $a_i, a_{i+1}, \ldots, a_{i+n/2-1}$ be sequence of largest values and note there must be some $j$ with $a_i \leq a_{i+1} \leq \cdots \leq a_{i+j}$ and $a_{i+j} \geq \cdots \geq a_{i+n/2-1}$
  - note that $a_i$ goes to input $(i \mod n/2)$ of lower subnet and subsequent values go to next consecutive inputs (with wrapping)
  - so the sequence that goes to lower subnet is bitonic
  - similar argument applies to upper subnet
- By induction, the original bitonic sequence is converted to a sorted sequence by $M_n$
Simple Sort & Route Network

- Simple components with no buffering
  - filter eliminates duplicates by comparing consecutive addresses and returns ack to inputs
  - adder computes and inserts "rank" of cells
  - concentrator uses rank as output address
  - routing network delivers to output

- Adder, concentrator, routing net all have \( \log_2 n \) stages (conc. is reverse banyan, routing net. is banyan)
Banyan Routing Property

- Use of unbuffered banyan network for routing depends on a key property of banyan networks.

- **Theorem.** Let \((x_0,y_0),\ldots,(x_r,y_r)\) be a set of input/output pairs for \(Y_{n,d}\). If \(y_0 < y_1 < \ldots < y_{r-1}\) and for \(1 \leq i \leq r-1\), 
  \(x_i = (x_{i-1} + 1) \mod n\), then the paths in \(Y_{n,d}\) joining the inputs to outputs in all pairs have no links in common.

  *Proof.* Proof by induction on number of stages. \(Y_{d,d}\) clearly satisfies statement. Consider a network with \(>1\) stage.

  Each of subnet formed when first stage is removed is a banyan net, so we need only show that the paths share no links in first stage and that the input/output pairs that must be routed in the subnets satisfy condition in the statement of the theorem.
Consider subnet \( j \); \( \ell_j = j(n/d) \) and \( h_j = \ell_j + (n/d) - 1 \) are the first and last outputs of subnet \( j \).

Let \( a \) be smallest integer such that \( \ell_j \leq \ell_P < h_j \) and let \( b \) be largest integer such that \( \ell_j \leq \ell_P < h_j \).

Only input/output pairs whose outputs are in subnet \( j \) have indices in interval \([a, b]\) implying that \( b - a + 1 \leq n/d \).

Because all input/output pairs involving subnet \( j \) appear on consecutive inputs to the network and there are at most \( n/d \) of them, they appear on inputs connected to distinct stage 1 switches. So, all can be routed to subnet \( j \) without conflict.

Also, because input/output pairs for subnet \( j \) appear on consecutive inputs to the network, they pass through consecutive stage 1 switches, which in turn connect to consecutive inputs on subnet \( j \), implying that the input/output pairs to be routed by subnet satisfy required conditions.

Argument holds for all subnets, so result follows by induction.
- First adder and concentrator compact cells
- Second adder computes fanout sum
- Copy range block computes address range and filters excess
- Copy network (banyan) does cell replication
- Routing tables assign outgoing links
- Sorting & routing block delivers cells to outputs
Limitations & Enhancements

- Poor worst-case throughput and unfairness
  » can have as few as 1 cell per cycle in worst-case
  » if number of copy network outputs is doubled, can always output \( n \) cells/cycle
  » alternatives involve rotating priorities and partial copies
- Each multicast routing table requires an entry for every multicast session - inefficient for small fanout sessions
  » also session modification requires changing \( n \) entries
- Buffers following routing table unevenly loaded
  » correct by rotating starting point of copy range computation
- Buffers introduce potential for loss-of-sequence
Improving Throughput

- Filter modified to allow two cells per output
- Routing net has two outputs for each external link; implemented as banyan pair
- Provides sufficient speedup to avoid input blocking due to head-of-line effects
- Output port processor must be modified to accept multiple cells at once