Deriving Concentrators from Binary Sorters Using Half Cleaners

Tripti Jain\(^1\), Klaus Schneider\(^1\), and Ankesh Jain\(^2\)

\(^1\)University of Kaiserslautern, \(^2\)University of Ulm

**Introduction**

- distributed systems and system-on-a-chip design require efficient non-blocking interconnection networks
- simplest solution: crossbars
  - non-blocking with optimal depth \(O(\log(n))\)
  - size grows with \(O(n^2)\)
- Cross/Benes networks
  - non-blocking with size \(O(n \log(n))\) and depth \(O(\log(n))\)
  - routes are difficult to compute (having \(O(\log(n)^2/\log(n))\) depth)

still non-blocking networks of size \(O(n \log(n)^2)\) and depth \(O(\log(n)^3)\) are required for small constants \(c\)

**Implementing Binary Concentrators**

- **goal**
  - improve RBS networks using binary sorters as \(\text{Split}\) modules
- **solution**
  - replace binary sorters by more efficient concentrators
  - we present a construction of binary concentrators using two binary sorters and a half cleaner

- Modules
  - \(\text{MBS}(n)\)
  - \(\text{RBS}(n)\)
  - \(\text{MBS}(n/2)\)
  - \(\text{RBS}(n/2)\)
  - \(\text{Split}(n)\)

- Merge-Based Sorting (MBS) vs. Radix-Based Sorting (RBS)

  - MBS
    - divide-and-conquer paradigm
    - uses \(\text{Merge}\) modules to merge already sorted halves
  - RBS
    - another divide-and-conquer paradigm
    - \(\text{Split}\) modules partition inputs in two halves to be sorted

  - both paradigms lead to fat binary trees with \(O(\log(n))\) levels of \(\text{Split}/\text{Merge}\) modules

**Experimental Results Using Netlist Generators**

- **size** (original concentrators)
  - \(32\times 32\) (619)
  - \(64\times 64\) (2048)
  - \(128\times 128\) (4096)
  - \(256\times 256\) (8192)
  - \(512\times 512\) (16384)
  - \(1024\times 1024\) (32768)

- **size** (optimized concentrators)
  - \(32\times 32\) (204)
  - \(64\times 64\) (1024)
  - \(128\times 128\) (4096)
  - \(256\times 256\) (8192)
  - \(512\times 512\) (16384)
  - \(1024\times 1024\) (32768)

- **depth** (optimized concentrators)
  - \(32\times 32\) (212)
  - \(64\times 64\) (1024)
  - \(128\times 128\) (4096)
  - \(256\times 256\) (8192)
  - \(512\times 512\) (16384)
  - \(1024\times 1024\) (32768)

**Experimental Results Using Cadence RC Compiler**

- **number of cells** (original concentrators)
  - \(32\times 32\) (619)
  - \(64\times 64\) (2048)
  - \(128\times 128\) (4096)
  - \(256\times 256\) (8192)
  - \(512\times 512\) (16384)
  - \(1024\times 1024\) (32768)

- **number of cells** (optimized concentrators)
  - \(32\times 32\) (204)
  - \(64\times 64\) (1024)
  - \(128\times 128\) (4096)
  - \(256\times 256\) (8192)
  - \(512\times 512\) (16384)
  - \(1024\times 1024\) (32768)

**Optimized RBS Networks**

- binary concentrators can be obtained by two sorters and a half cleaner
- improves both the size and depth of the binary sorters
- building RBS networks with these concentrators improves the designs that would otherwise use binary sorters

**Split**

- \(\text{Split}(n)\)
- \(\text{depth}\)
- sorters are binary concentrators, but not vice versa; e.g., \(00101111\) is concentrated, but not sorted

**Merge**

- \(\text{Merge}(n)\)
- \(\text{MBS}(n)\)
- \(\text{RBS}(n)\)
- \(\text{MBS}(n/2)\)
- \(\text{RBS}(n/2)\)

**Half Cleaners Need Bitonic Sequences**

- both halves of the input sequence of half cleaners must be sorted
- concentrated input sequences are not sufficient
- see the counterexample below (halves of rightmost column (half cleaner) are concentrated, but not sorted, output is not concentrated)

**Batcher’s Bitonic Sorting Network**

- it is a MBS network introduced by Batcher in 1968
- \(\text{Merge}\) modules are also fat binary trees using half cleaners
- half cleaner with \(n\) inputs/outputs
  - a column of \(4 \times 2 \times 2\) crossbar switches
  - incoming perfect shuffle and outgoing flip shuffle permutation

**Batcher’s Bitonic Sorting Network for 8 inputs**

- \(\text{MBS}(8)\)
- \(\text{RBS}(8)\)
- \(\text{Split}(8)\)
- \(\text{MBS}(4)\)
- \(\text{RBS}(4)\)
- \(\text{Split}(4)\)

- our optimization applies to sorters, but not to concentrators, i.e., cannot be repeated

**RBS Network with Concentrators built by Optimized Bitonic Sorters**