Parallel Hardware Merge Sorter

Wei Song[1,2], Dirk Koch[1], Mikel Lujan[1] and Jim Garside[1]

1. School of Computer Science, the University of Manchester
Manchester M13 9PL United Kingdom
2. Computer Laboratory, the University of Cambridge
Cambridge CB3 0FD United Kingdom

ws327@cam.ac.uk; {dirk.koch, mikel.lujan, james.garside}@manchester.ac.uk

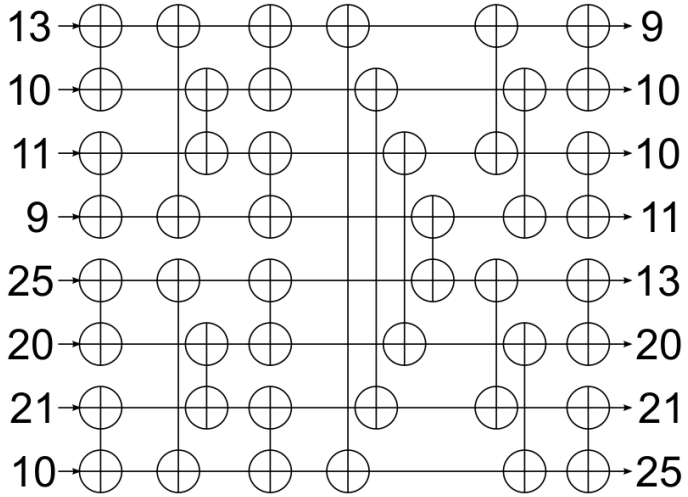
FCCM 2016 2nd May 2016

Hardware Sorter

- Sorting algorithm is important
  - One of the most used algorithms
  - Database, search engine, MapReduce, *et al.*
  - Highly data dependent
- Hardware sorting accelerator
  - Unknown random data distribution
  - cache < data set < memory
  - No good solution until very recent

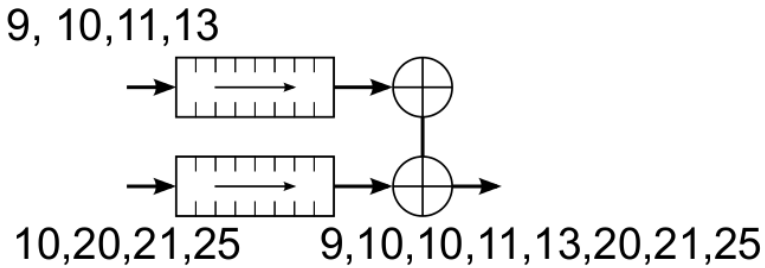
Type of Sorters

Parallel (Sorting Network)



- Block operation
- High sorting rate
- Limited data set

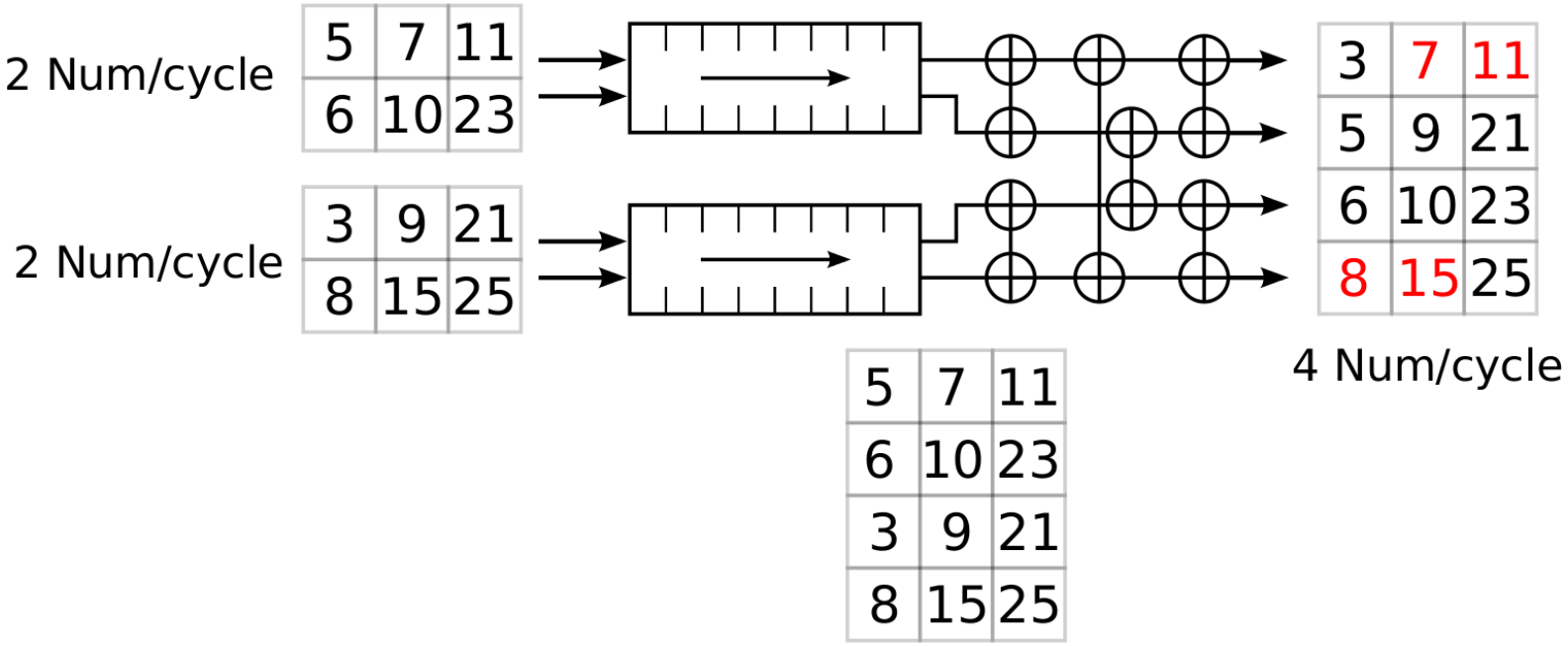
Sequential Sorter



- Stream operation
- Low sorting rate (1 number/cycle)
- Unlimited data set

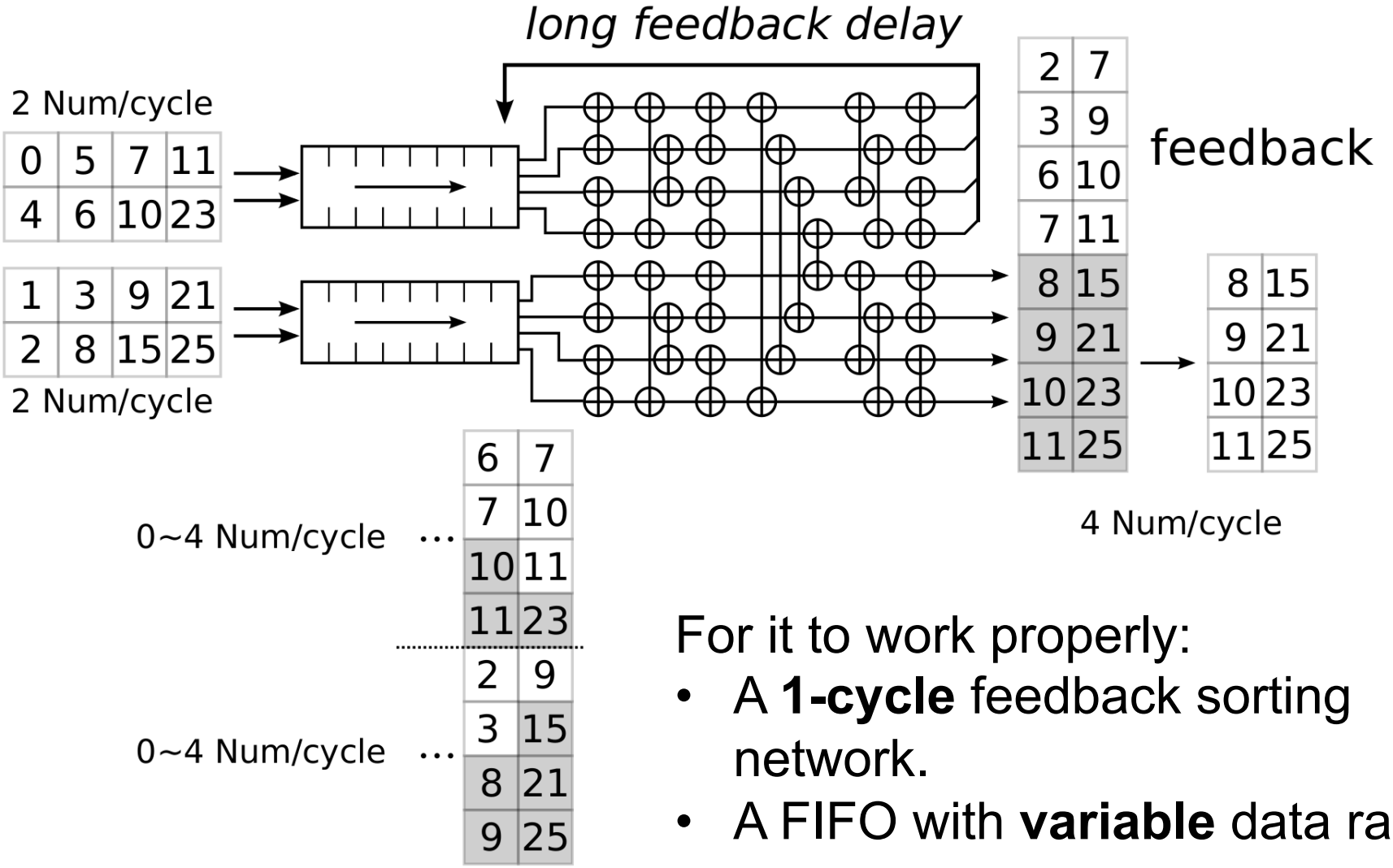
Can we combine them?

Naïve Parallel Merger



Does NOT work because data may distribute unevenly!

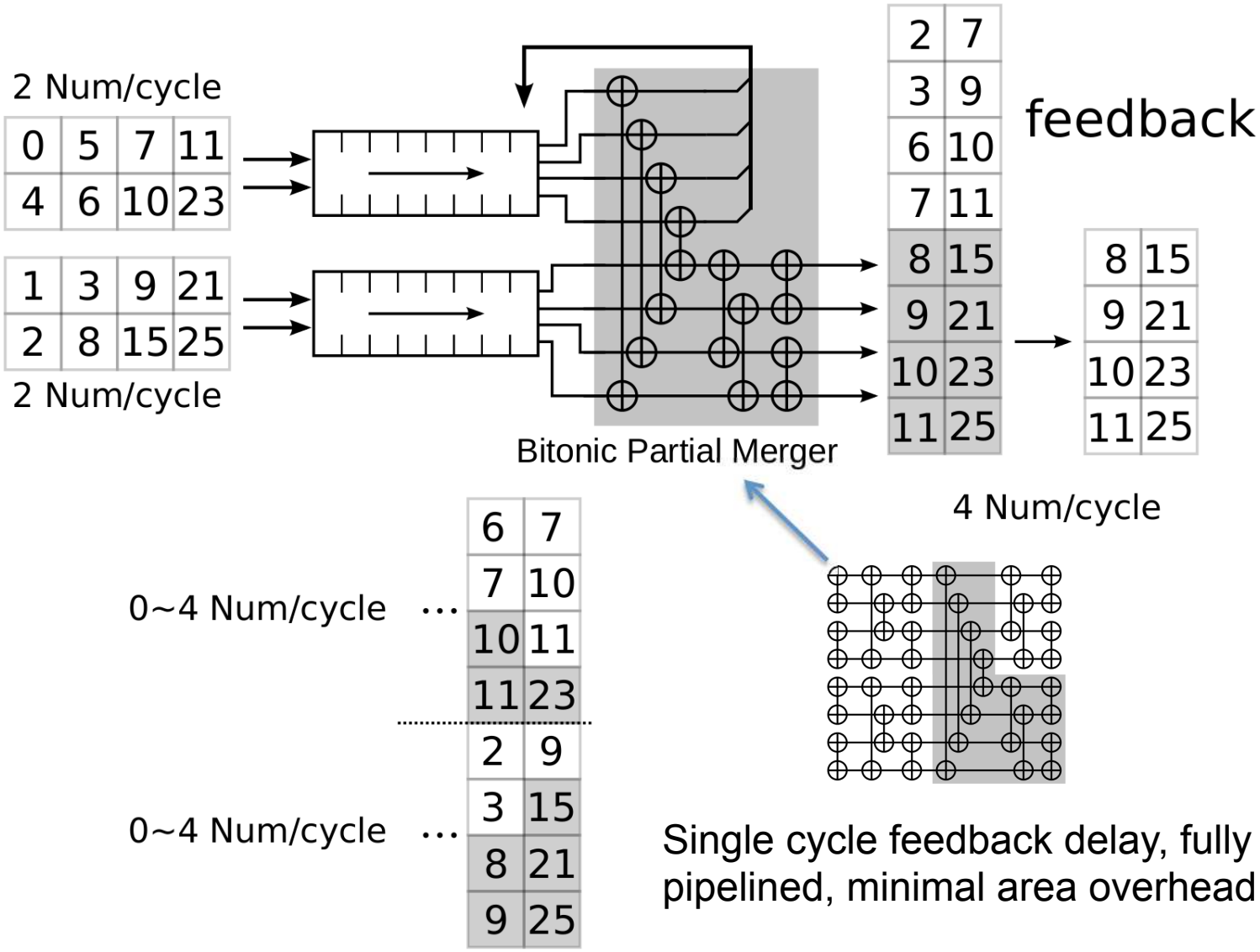
Naïve Parallel Merger



For it to work properly:

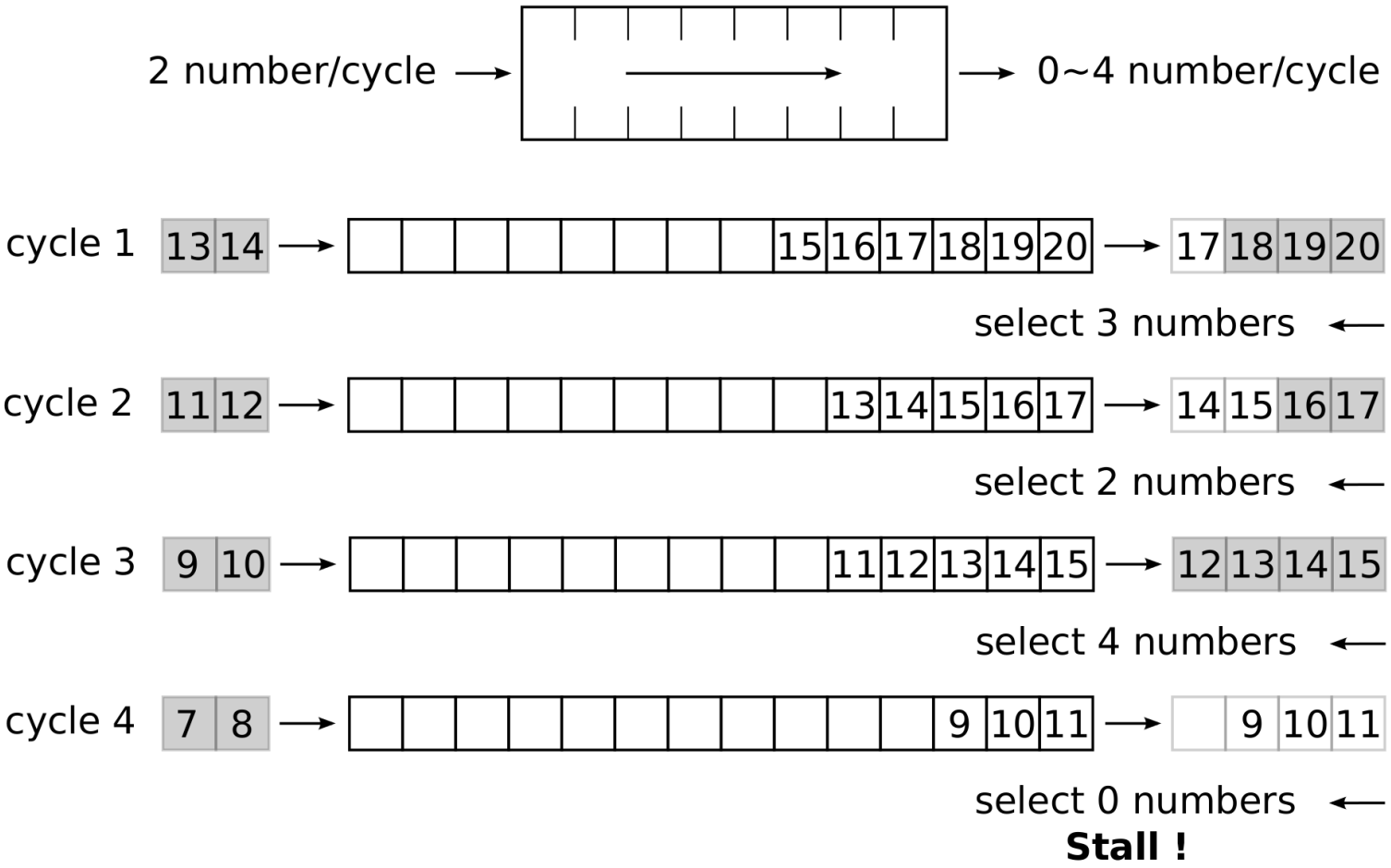
- A **1-cycle** feedback sorting network.
- A FIFO with **variable** data rate.

Fast Feedback using Bitonic Merger



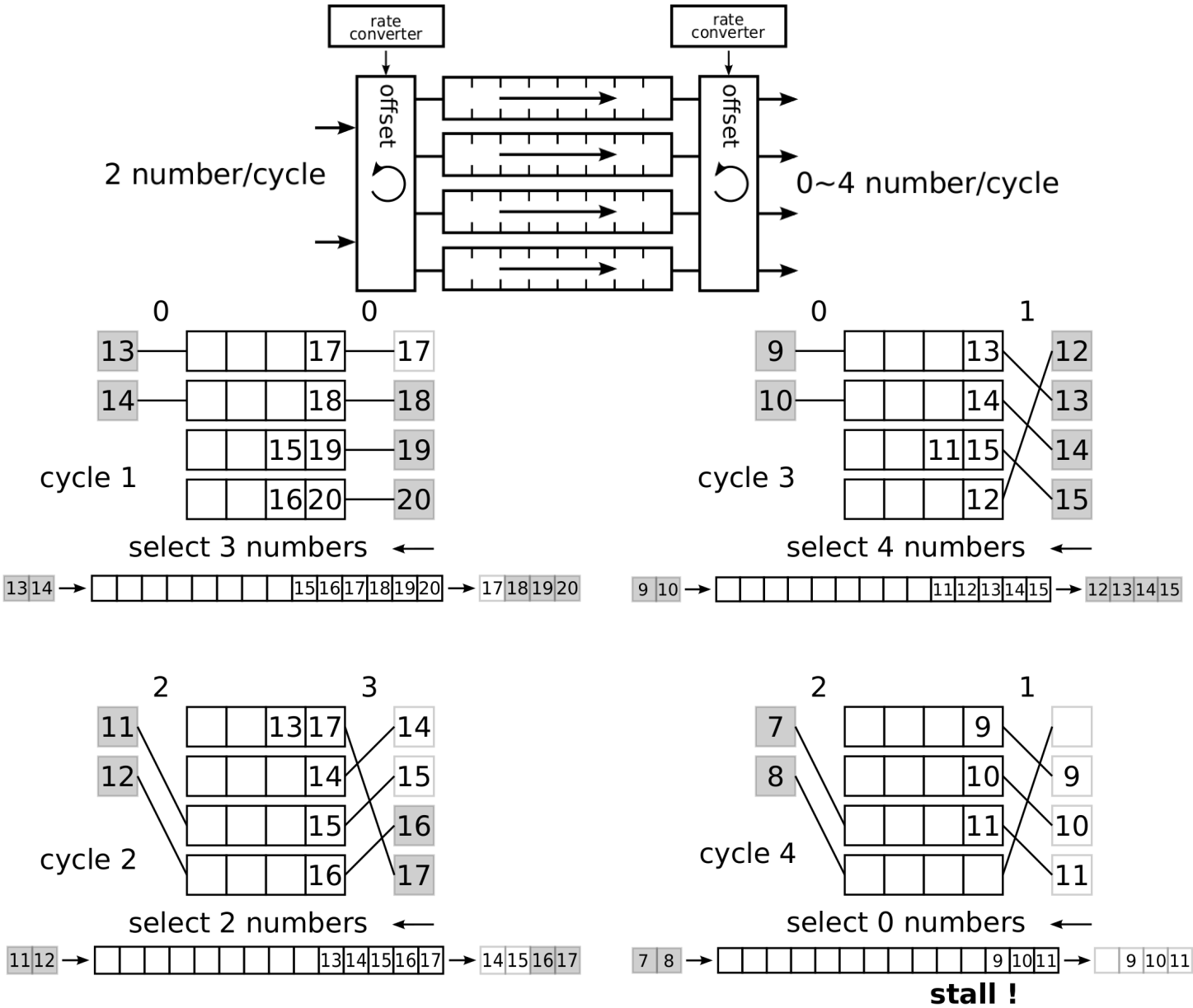
Single cycle feedback delay, fully pipelined, minimal area overhead.

Area Efficient Multirate FIFO

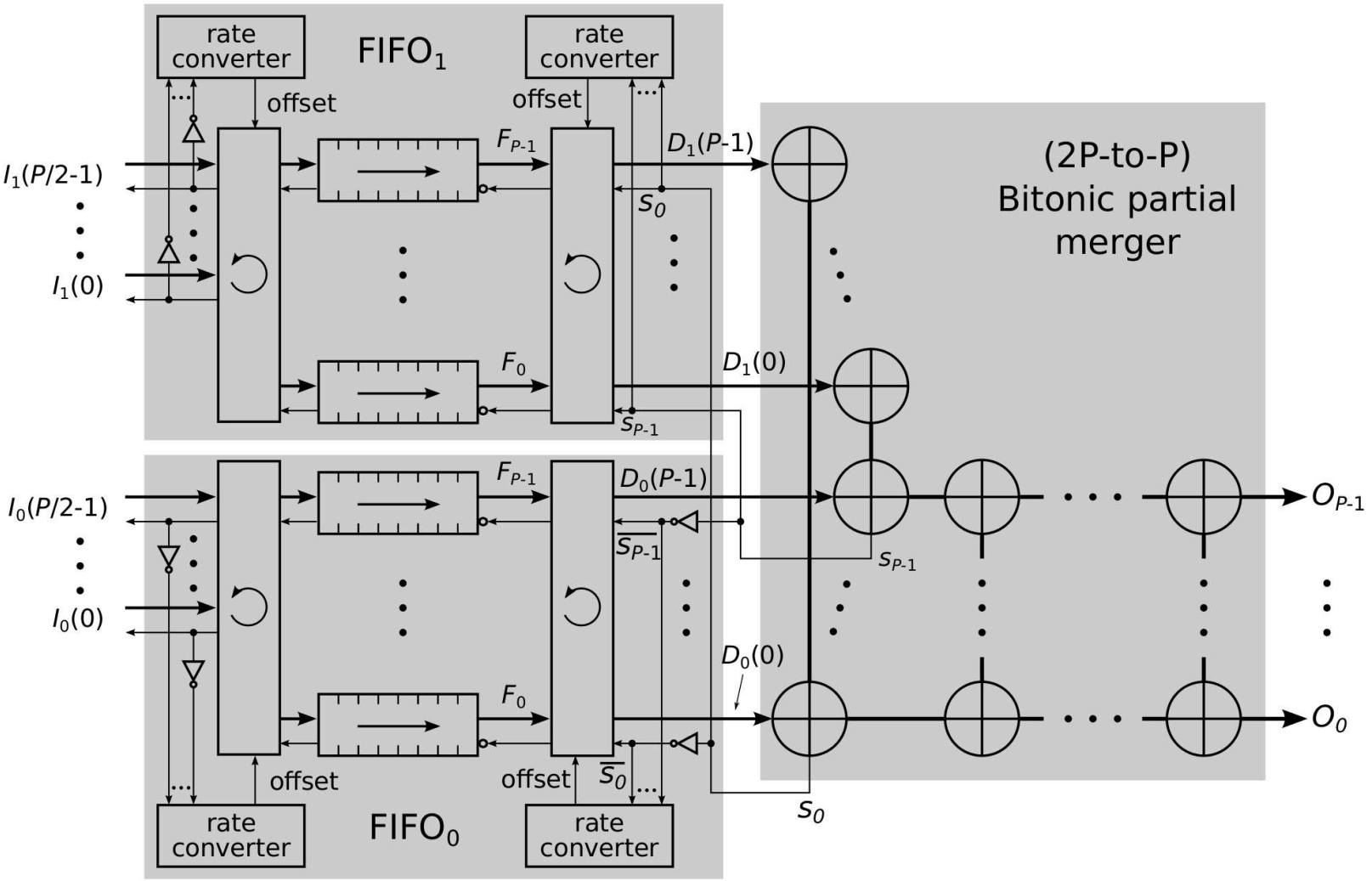


Not a standard FIFO or shifter design.

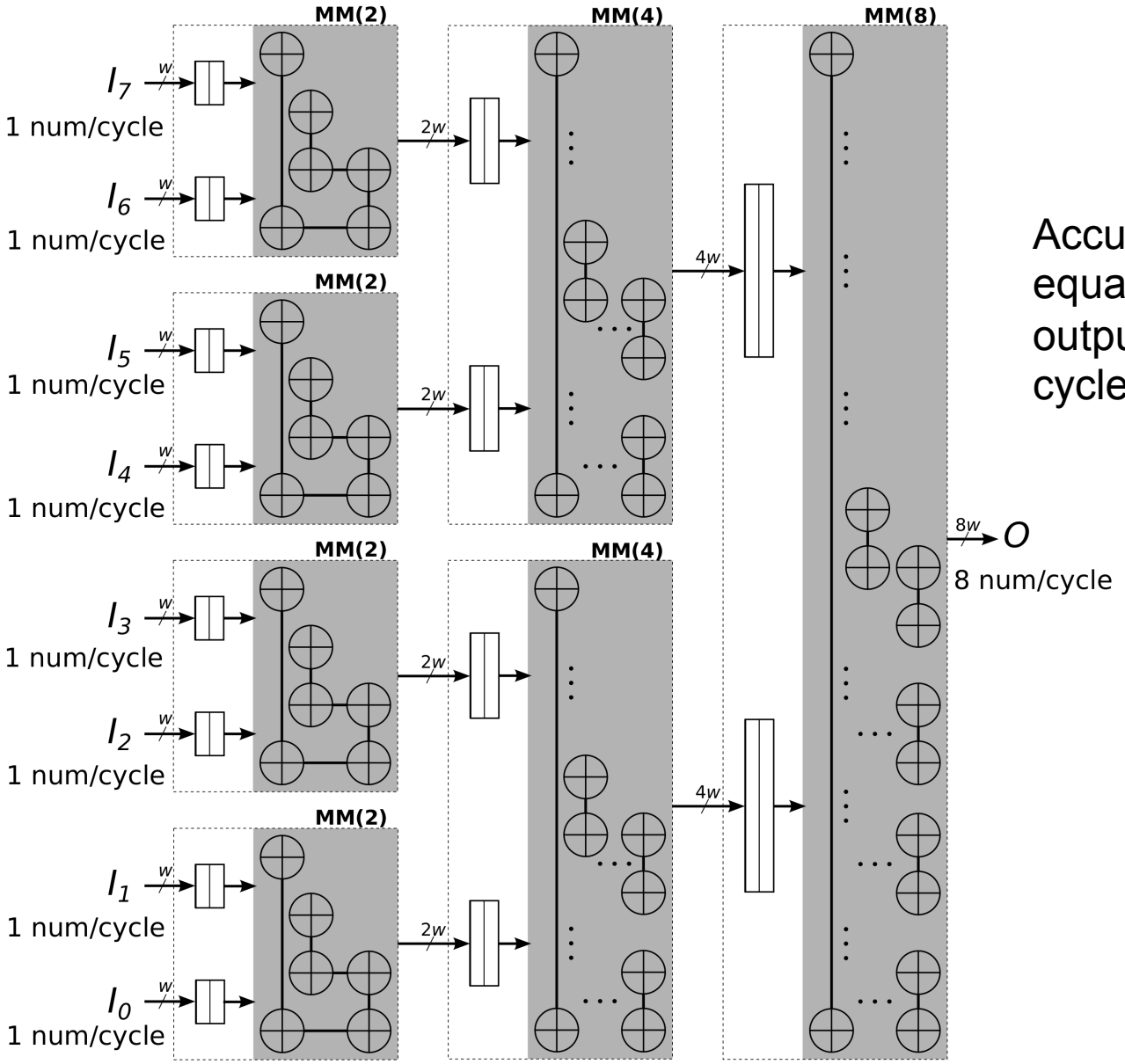
Area Efficient Multirate FIFO



Multirate Merger (Implementation)



Parallel Merge-Tree

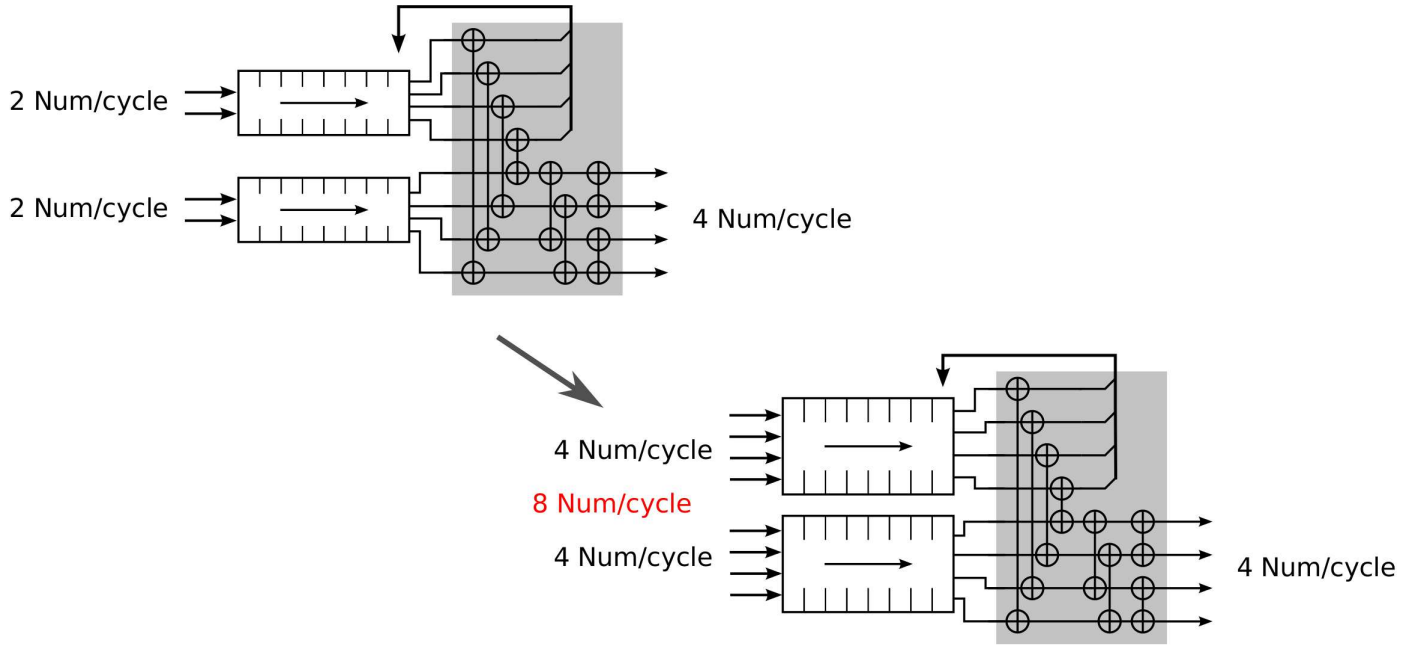


Accumulated input rate equates with the maximal output rate, both are 8 num/cycle.

Rate Mismatch

Two ways to handle rate mismatch:

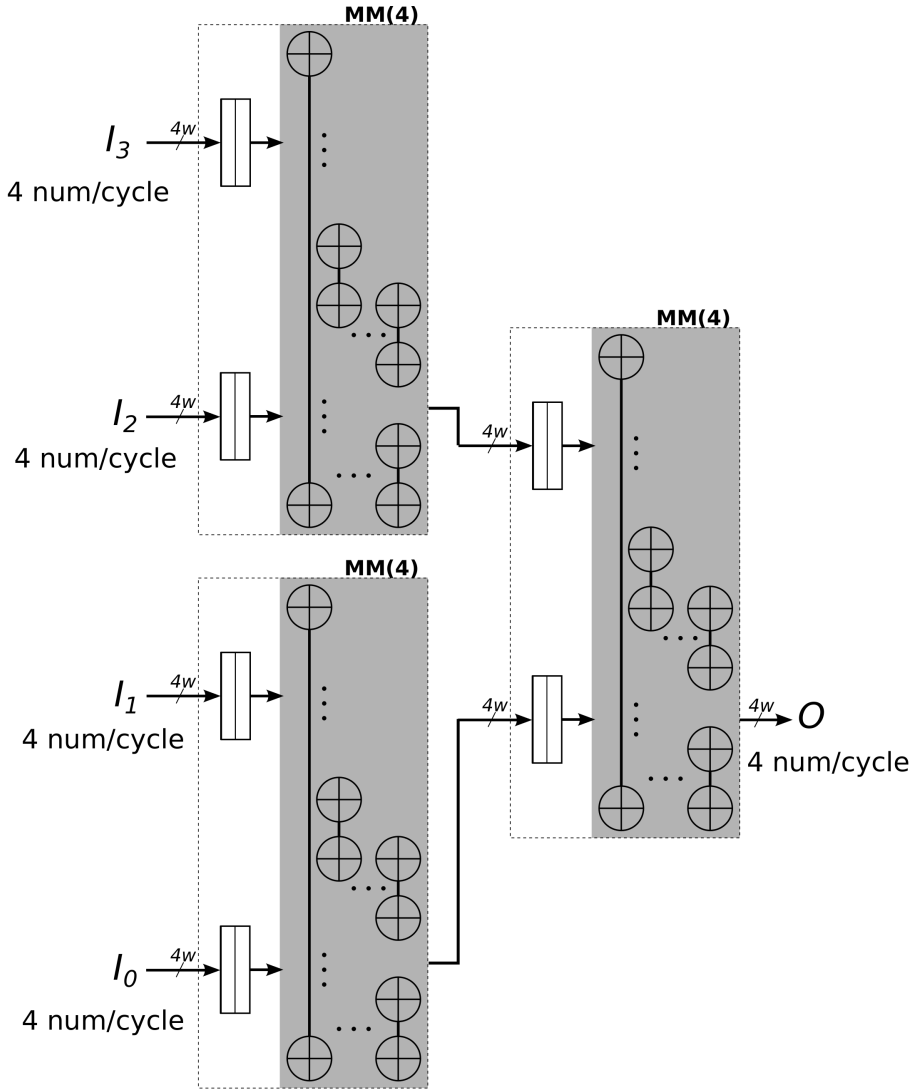
- Increase input rate to the maximal output rate in each merger (**previous work**).
  - (J. Casper and K. Olukotun. “Hardware acceleration of database operations.” FPGA 2014)



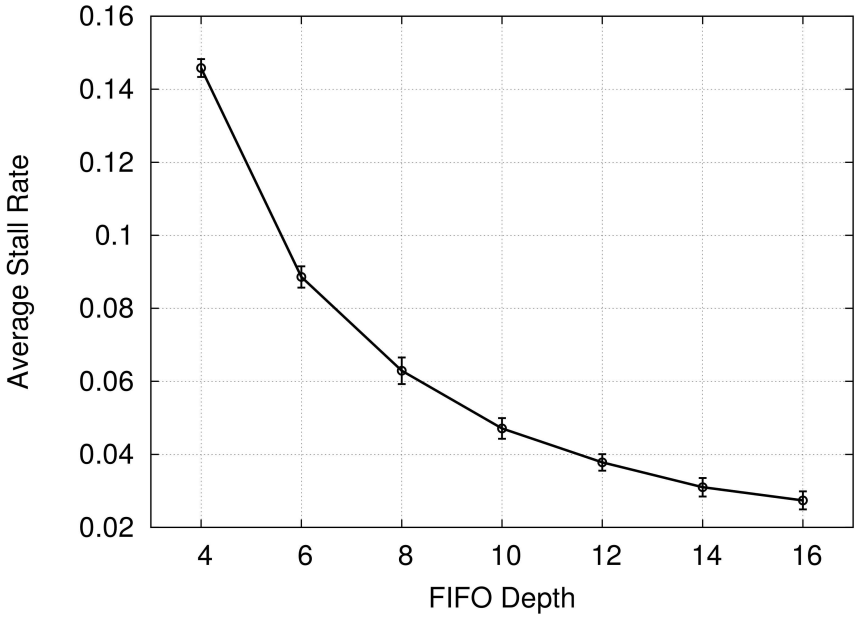
- Allow stall and use long FIFOs to reduce stall rate. (**this paper**)

Rate Mismatch

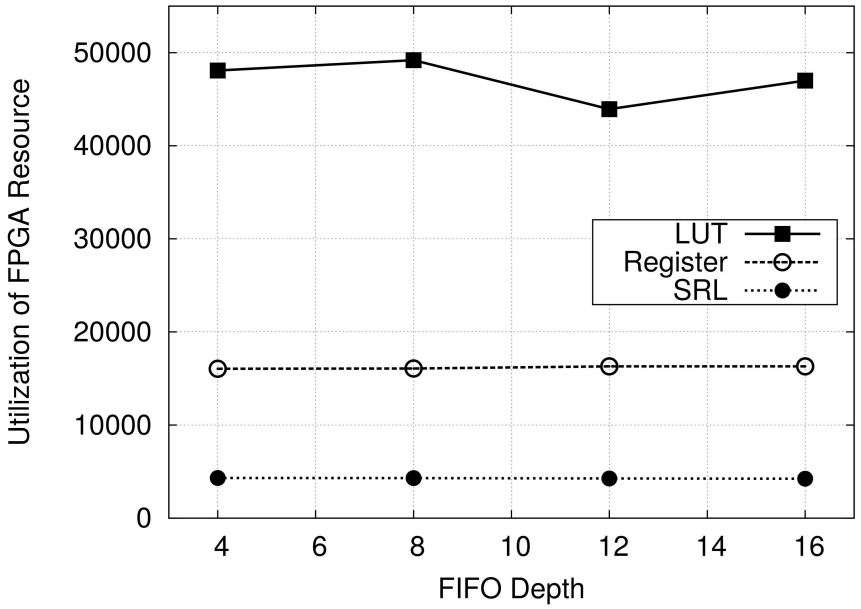
Significant increase of accumulated input rate!
For a 4 num/cycle merge tree, the accumulated input rate is **16 num/cycle**!



Reducing Stalls using FIFO



Stall rate vs. FIFO depth

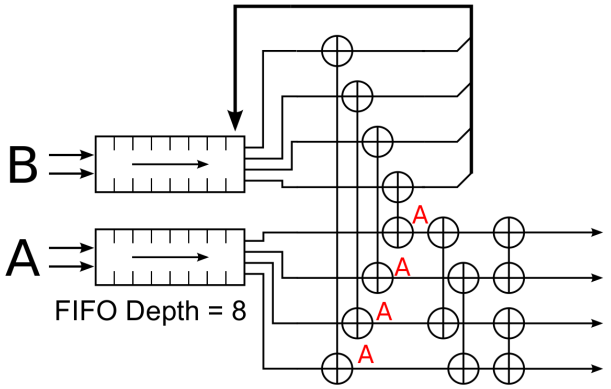


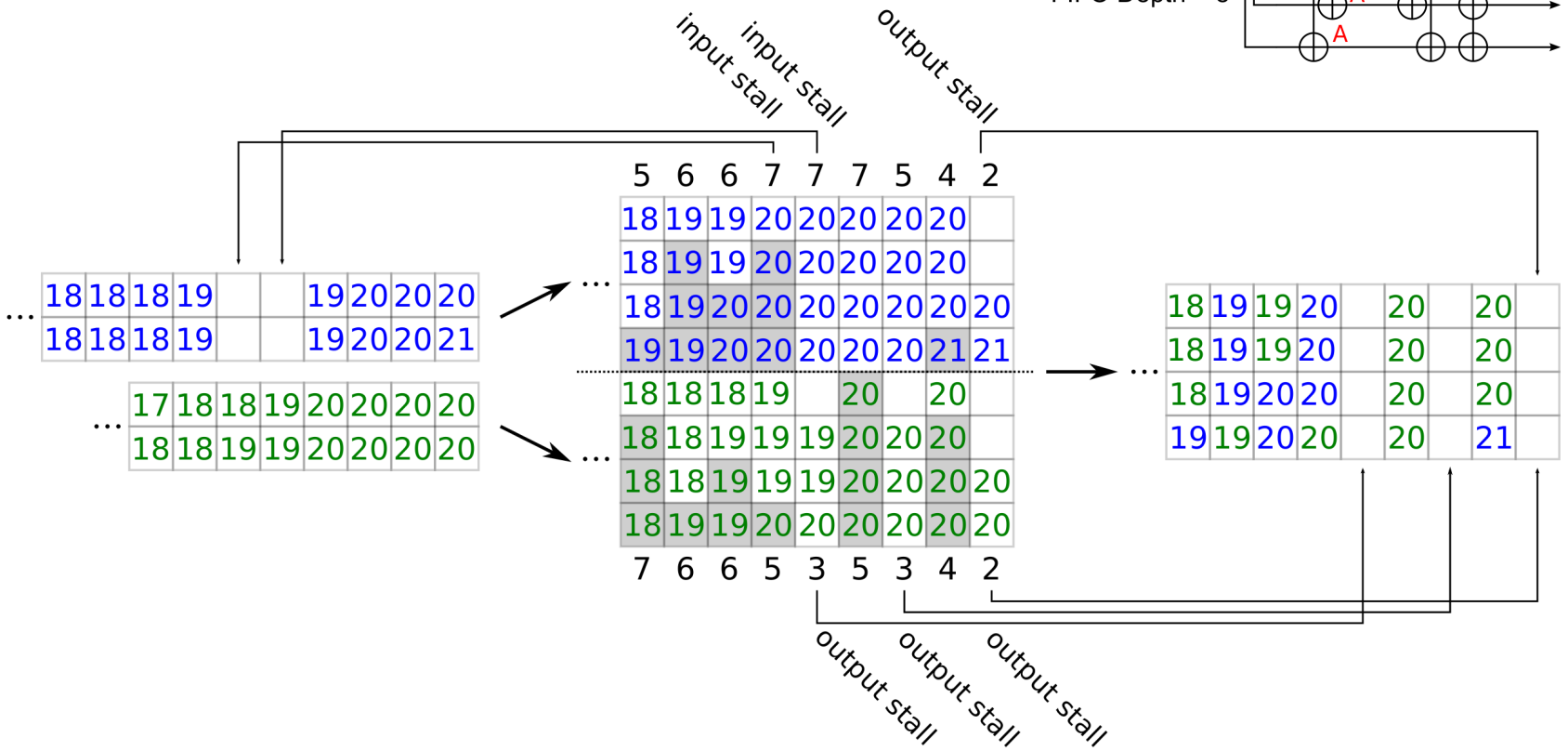
FPGA area overhead

A single SRL is a 16-bit shifter.
There is no extra cost to increase depth to 16.

Sorting Skewed Data

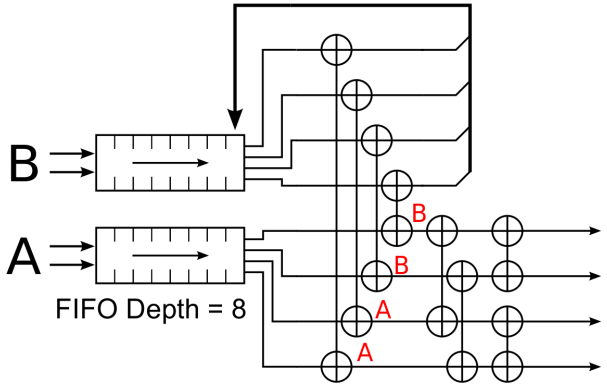
Skewed data: data with uneven distribution, causing duplicated data.

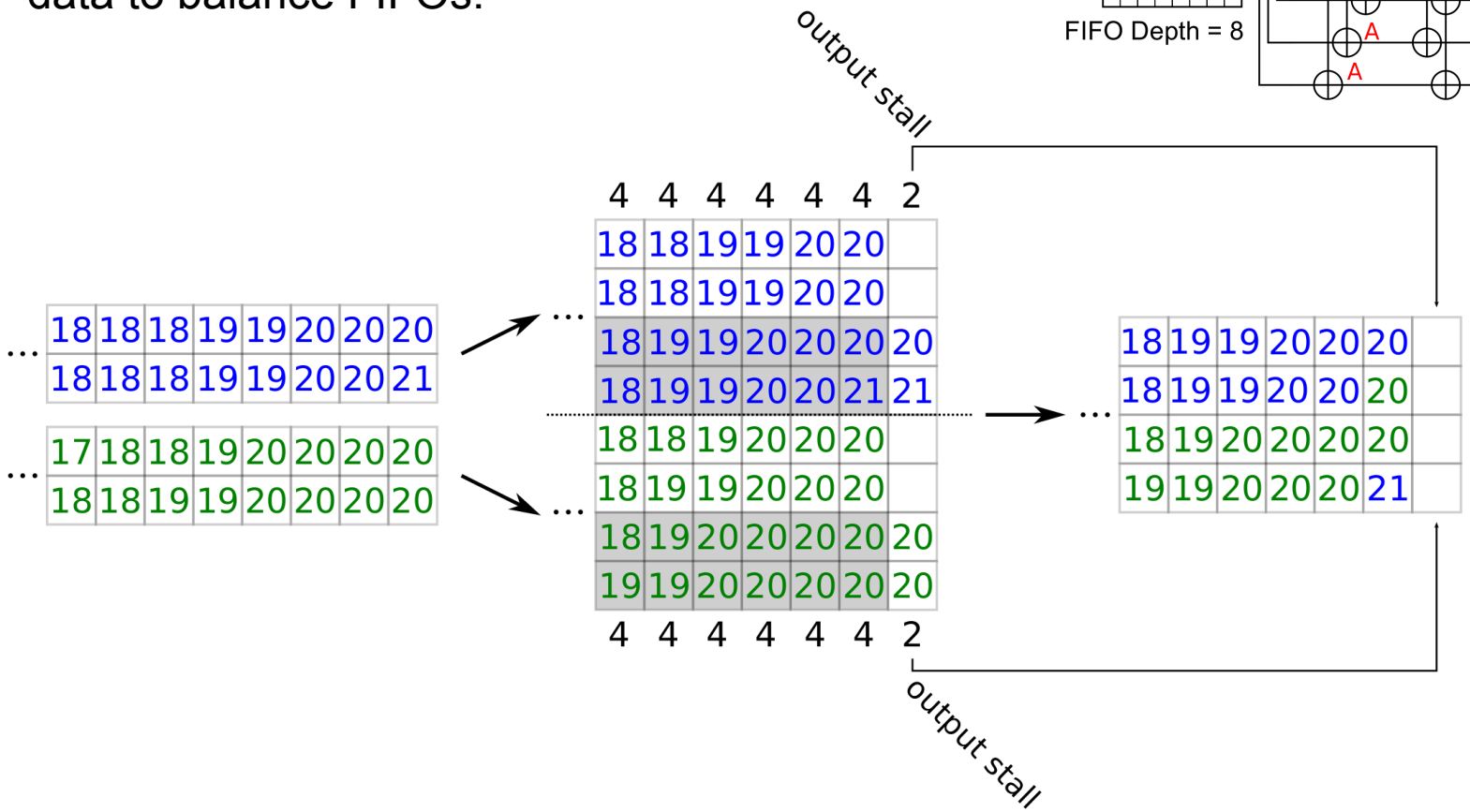




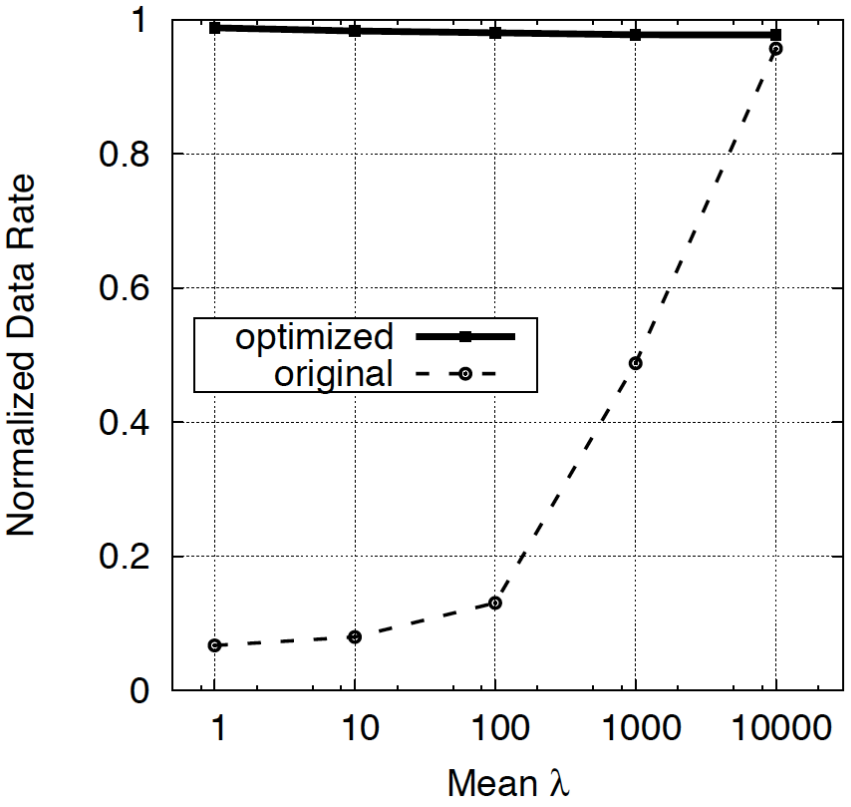
Sorting Skewed Data

Statically configure comparators to choose the even distribution. When possible, use duplicated data to balance FIFOs.

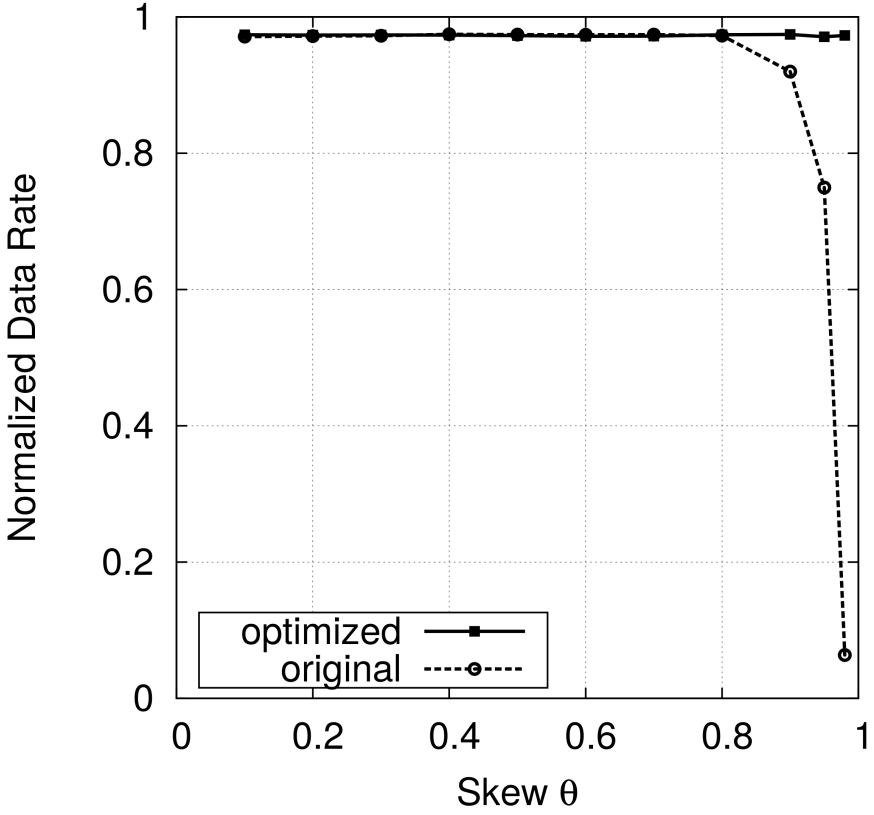




Sorting Skewed Data



Poisson distribution

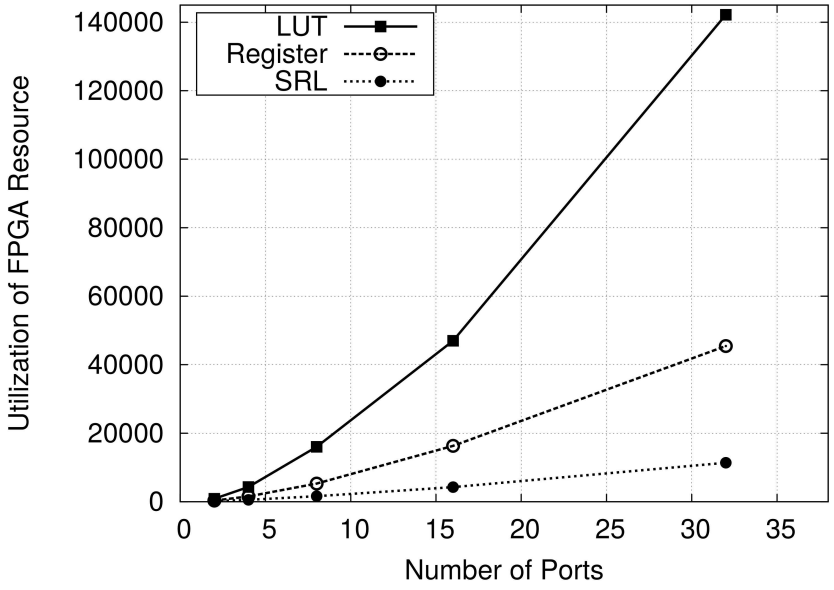


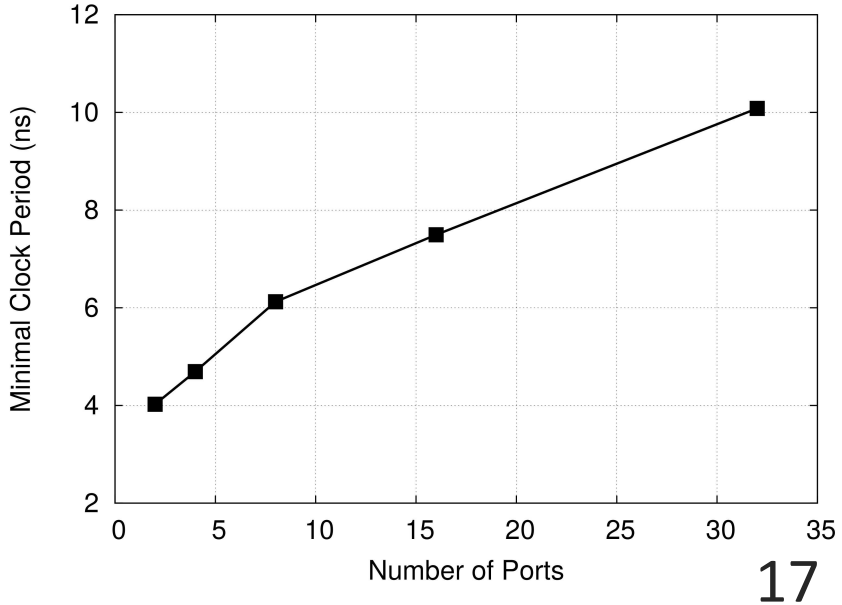
Zipfan distribution

Scalability Analysis

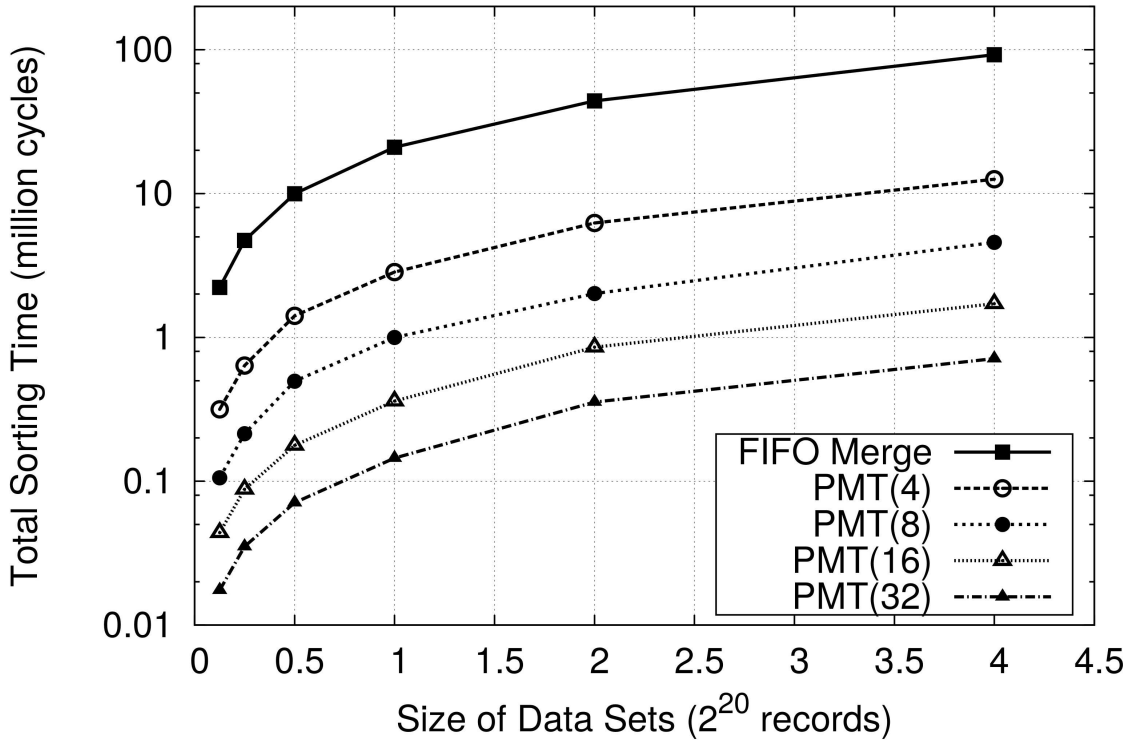
Parallel Merge-Tree in a Virtex-7 XC7VX485T FPGA

Ports	Period (ns)	Frequency (MHz)	Register	LUT	SRL	Stall Rate	Data Rate (Number/cycle)	Data Rate (Gb/s)
2	4.02	248.5	328 (0.05%)	853 (0.28%)	132	1.75%	1.97	31.3
4	4.69	213.1	1534 (0.25%)	4278 (1.41%)	528	2.16%	3.91	53.4
8	6.12	163.3	5287 (0.87%)	16016 (5.28%)	1608	2.56%	7.80	81.5
16	7.50	133.4	16299 (2.68%)	47001 (15.48%)	4238	2.74%	15.56	132.9
32	10.08	99.2	45445 (7.48%)	142179 (46.83%)	11379	2.98%	31.05	197.1





Sorting Time Estimation



$$t_{\text{PMT(P)}} \sim (1 - \bar{r}_{\text{stall}}) \cdot \frac{N}{P} \log_P N$$

$$= \frac{(1 - \bar{r}_{\text{stall}})}{P \log P} \cdot N \log N$$

$t_{\text{FIFOMerge}}$

160x Speed up (PMT32)

Related Sorters (Rough Estimation)

Sorter	Type	Device	Clk Freq	Rate	Time est (1G N)	Description
Parallel Merge Module (1989) [1]	Parallel merger	ASIC	N/A	N/A	N/A	No stall, single stage full comparator.
High BW Sort Merge Unit (2014) [2]	Parallel merger	Xilinx Virtex-6	200M	6 num/cycle 4 seq/pass ?	2.5G cycle 12.5 sec	No stall. Need 5-port memory support, 6.14x memory BW.
Parallel Merge-Tree (2016) [This]	**Parallel merger**	**Xilinx Virtex-7**	**99M**	**32 num/cycle 32 seq/pass**	**0.19G cycle 1.88 sec**	**Allow stall, 1x memory BW.**
FPGASort (2011) [3]	Sequential merger	Xilinx Virtex-5	252M	1 num/cycle 2 seq/pass	30G cycle 2 min	
Tree-merger (2011) [3]	Sequential merger	Xilinx Virtex-5	273M	1 num/cycle 102 seq/pass	4.5G cycle 16.5 sec	

[1] G. S. Liu and H. H. Chen. "Parallel merge module for combining sorted lists." *IEE Proceedings,* 1989.
[2] J. Casper and K. Olukotun. "Hardware acceleration of database operations." in *Proc. of FPGA*, 2014.
[3] D. Koch and J. Torresen. "FPGASort: a high performance sorting architecture exploiting run-time reconfiguration on FPGAs for large problem sorting." In *Proc. of FPGA*, 2011.

Conclusion and Future Works

- A scalable parallel merge-tree
  - Merge multiple sorted sequences
  - High sorting speed-up
    - High sorting rate
    - Merging multiple sequences in each pass
  - Scalable area and speed performance
    - Allowing stall
  - Strong tolerance to skewed data

- Future work
  - Full sorting system with connection to host computer
  - Data scheduling and compression over PCIe