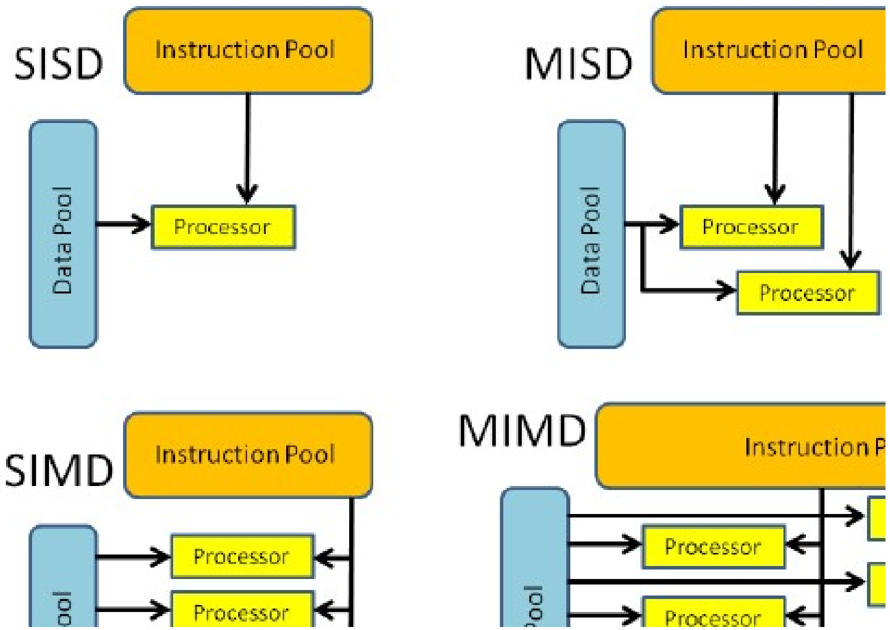
§9 Parallel Algorithms

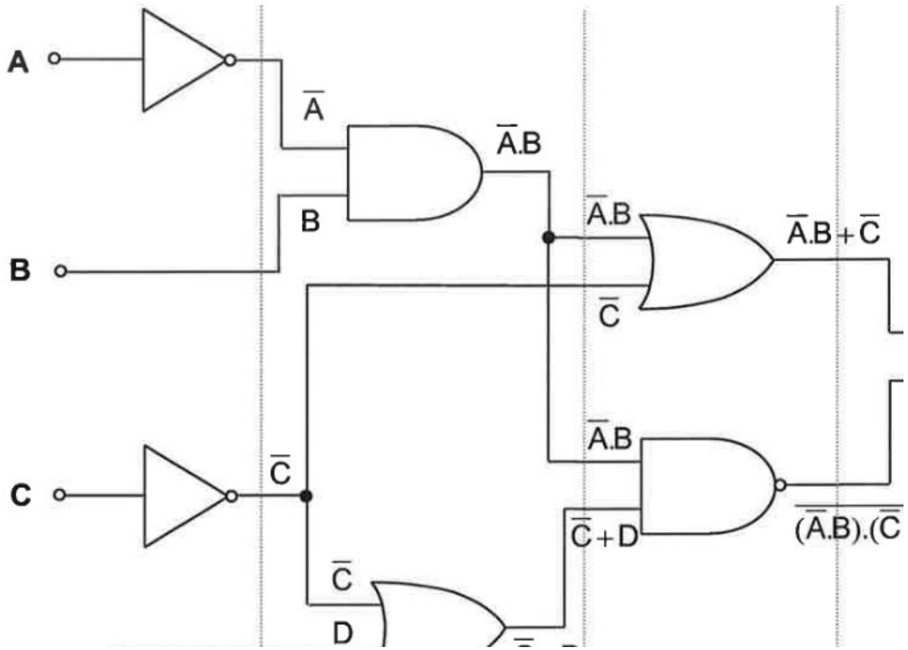
- Classification
- Parallel Prefix
- Graph Reachability
- Carry-Lookahead
- Sorting Networks

Classification

- *fine*-grained parallelism vs. *distributed* computing
- SISD/MISD/SIMD/MIMD
- PRAMs: CRCW/CREW/EREW

Here: <u>algorithms</u>, not programs ("*abstraction*")





Primitives **and** ***Cost*** **Measures**

Size = #binary *gates*

Depth = "parallel runtime"

Span

Alternative **Size**

=#cores/#CPUs/#PCs,

→ Communication **Volume,**

Work **:= Span · #CPUs**

Here: *Circuits*

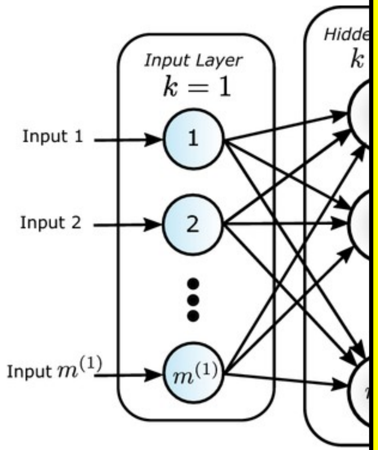
Brent's Principle:

Work, Size **≥** ***Time***

(seq.opt.) **Time**

Speedup := Time/Depth

Overhead := Size/Time



Lemma:

Any circuit of *binary* gates, depending on N inputs, has

- size ≥ N-1 and
- depth ≥ $\log(N)$

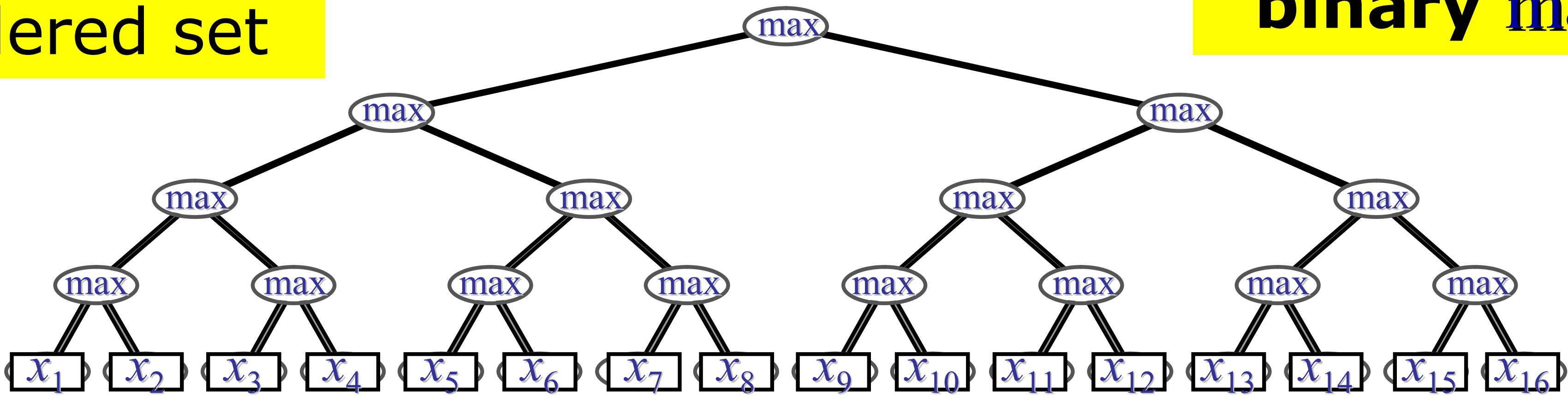
Parallel N-ary Maximum

Size = $O(N)$ binary op.s **Depth** = $O(\log N)$

$(X,\leq)$ totally ordered set

optimal!

Primitive op: **binary** max



Brent: ***Size* ≥ *Time***

(ignore *magnitude* of operands...)

(seq.opt.) **Time** = $O(N)$

Speedup = Time/Depth = $O(N/\log N)$

Overhead = Size/Time = $O(1)$

$(x_1,\ldots,x_N) \to \max_{1\leq n\leq N} x_n$

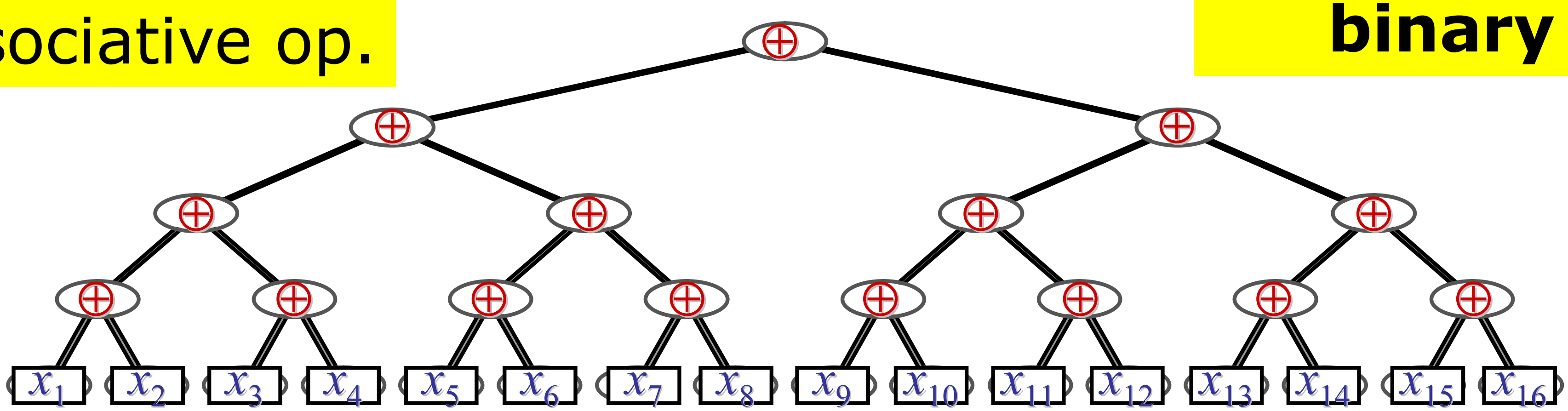
Parallel N-ary *associative* $\oplus$

Size = $O(N)$ binary op.s

Depth = $O(\log N)$

$(X,\oplus)$ set with associative op.

Primitive op: **binary** $\oplus$



$(x_1,\ldots,x_N) \to \bigoplus_{1\le n\le N} x_n$

(seq.opt.) **Time** = $O(N)$

Speedup = Time/Depth = $O(N/\log N)$

Overhead = Size/Time = $O(1)$

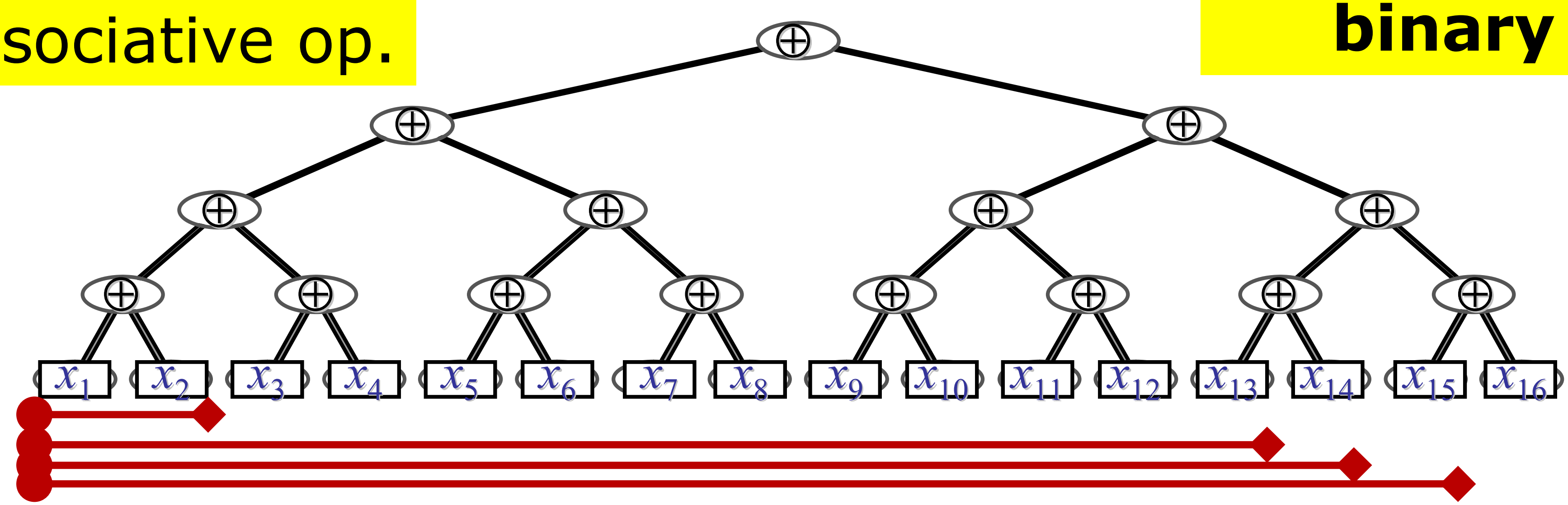
(Generalized) Prefix Sum / *Scan*

Size = $O(N^2)$ binary op.s

Depth = $O(\log N)$

$(X,\oplus)$ set with associative op.

Primitive op: **binary** $\oplus$



(seq.opt.) **Time** = $O(N)$

Speedup = Time/Depth = $O(N/\log N)$

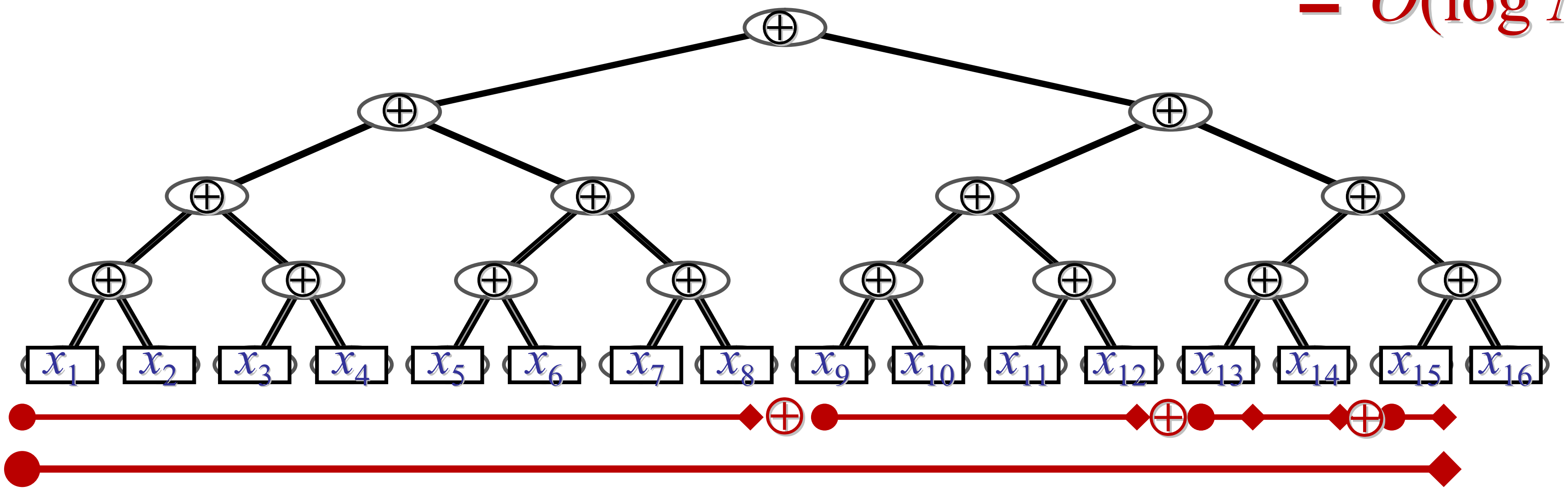
Overhead = Size/Time = $O(N)$

$(x_1,\ldots,x_N) \to \left(\bigoplus_{1\leq n\leq M} x_n : M=1\ldots N \right)$

Parallel *Prefix*

Size = $O(N)$ binary op.s
+ $O(N)$ = $O(N)$

Depth = $O(\log N)$
+ $O(\log N)$
= $O(\log N)$



(seq.opt.) **Time** = $O(N)$

Speedup = Time/Depth = $O(N/\log N)$

Overhead = Size/Time = $O(1)$

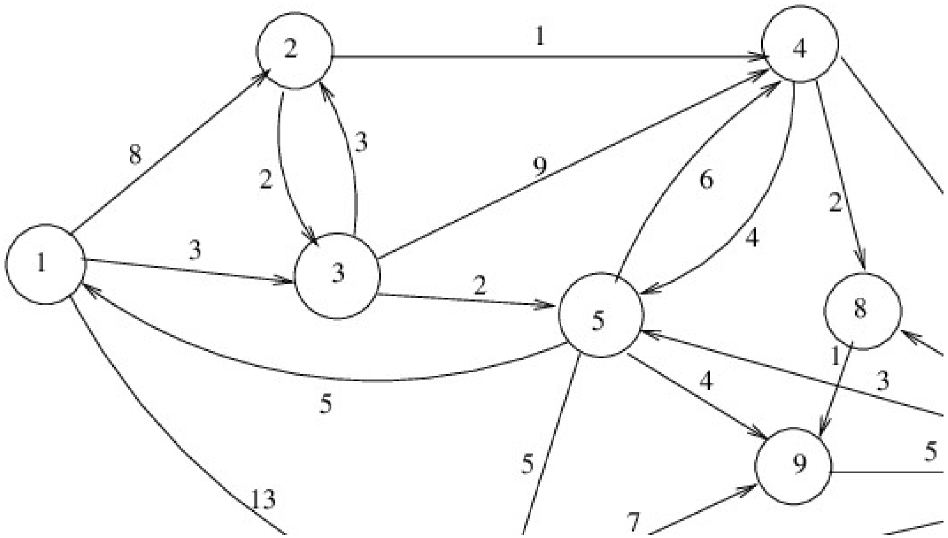
$$(x_1,\ldots,x_N) \to \left(\bigoplus_{1\le n\le M} x_n :\ M=1\ldots N\right)$$

Sequential **Graph Reachability**

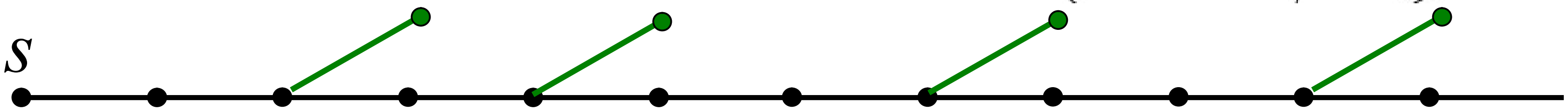
t

Input: $K,s,t\in\mathbb{N}$ and directed graph as $N\times N$
0/1 adjacency matrix A, vertices $V=\{1\ldots N\}$

Output: Is t reachable from s within distance K **?**



s



Dijkstra: Time = $O(N^2)$

Parallel Graph Reachability

Size = $O(N^3 \cdot \log K)$ **Depth** = $O(\log N \cdot \log K)$

Input: $K,s,t\in\mathbb{N}$ and directed graph as $N\times N$
0/1 adjacency matrix A, vertices $V=\{1\ldots N\}$

Output: Is t reachable from s within distance K ?

Answer: $(A^K)_{s,t}$ Bool. matrix power

depth $O(\log N)$
size $O(N^3)$

$N\times N$ *Boolean* matrix product

$$(X\cdot Y)_{I,J} = \bigvee_L X_{I,L} \wedge Y_{L,J} \qquad \forall\ I,J=1..N$$

Matrix powering $A \to A^K$
by repeated squaring $A^2, A^4, A^8, \ldots$

$O(\log K)$ repetitions

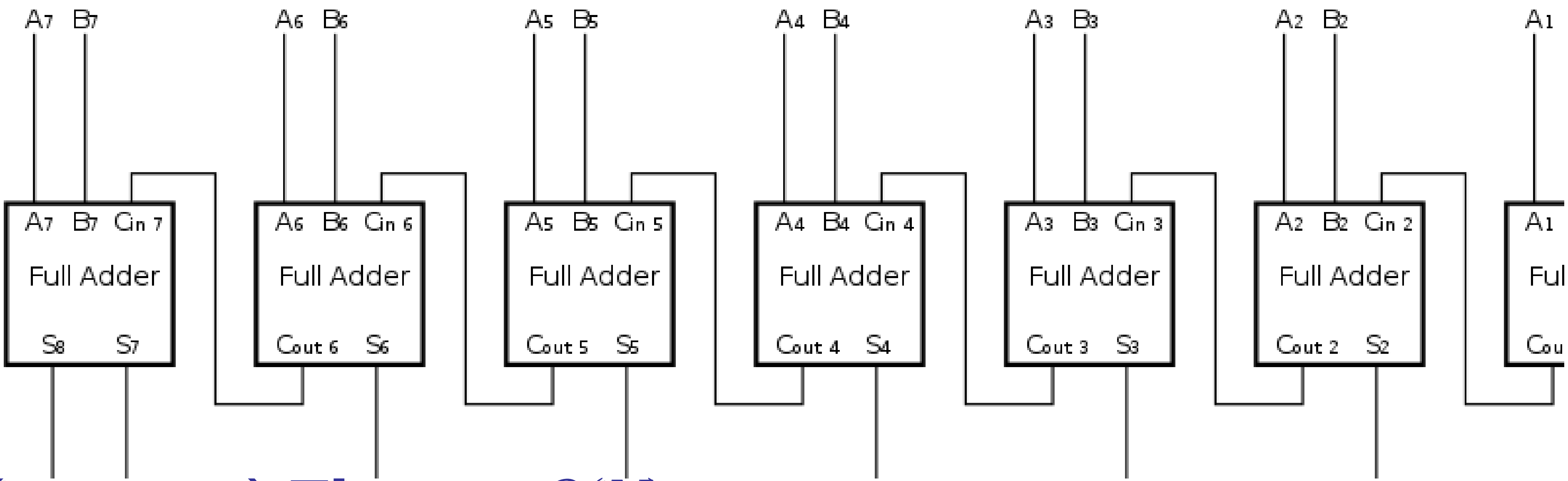
Dijkstra: Time = $O(N^2)$

Associative $\vee$

Sequential *Ripple-Carry* Adder

Input: two n-bit integers in binary,

$$A=(A_0,\ldots,A_{N-1})_2 \quad \text{and} \quad B=(B_0,\ldots,B_{N-1})_2$$



(seq.opt.) **Time** = $O(N)$

Depth = $O(N)$

Speedup = Time/Depth = $O(1)$

Output: $S:=A+B$ in binary,

$$S=(S_0,\ldots,S_{N-1},S_N)_2$$

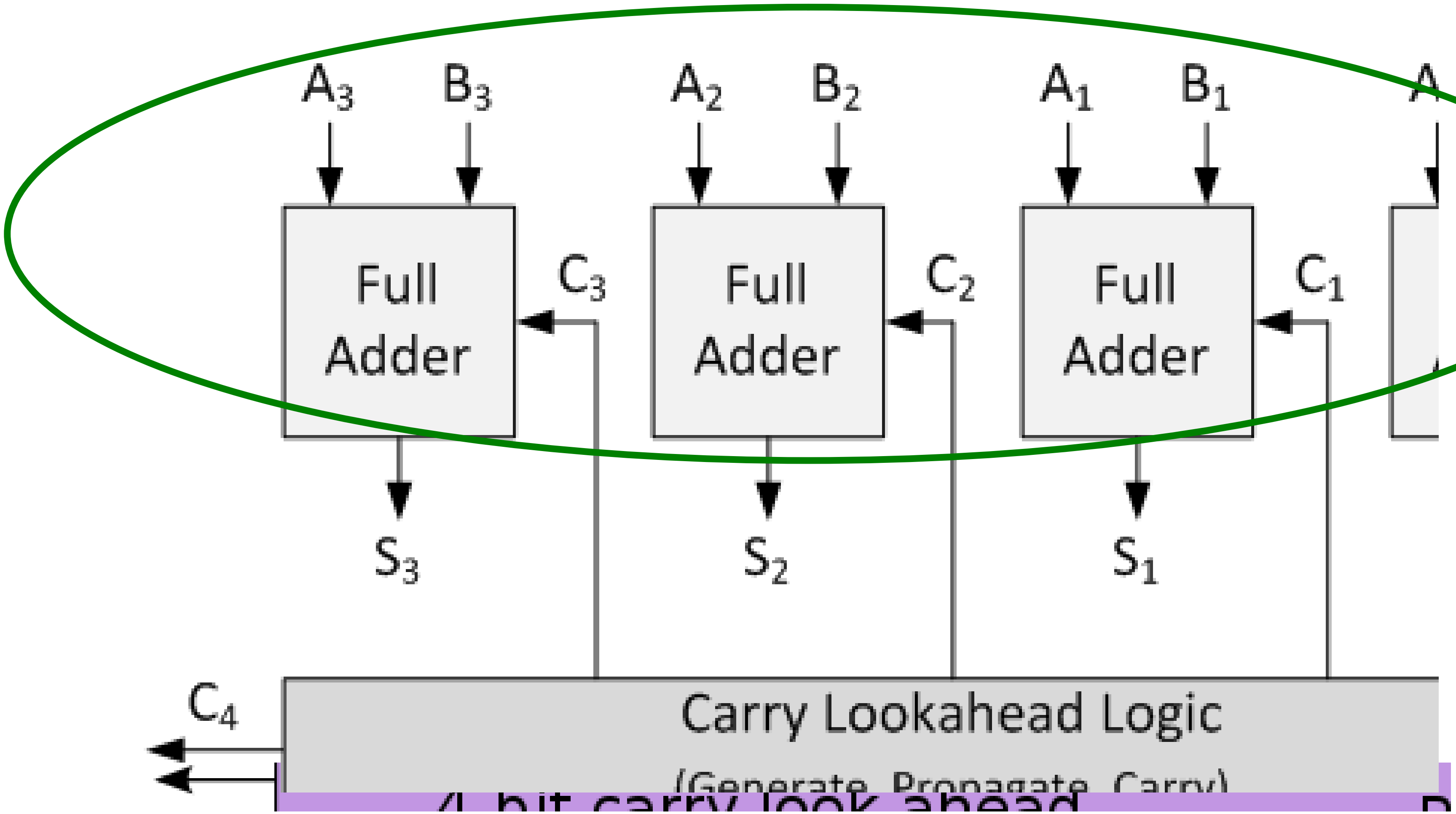
Operations:

binary $\wedge, \vee, \neg$

Carry-*Lookahead* Adder

Size = $O(N)$ binary bit op.s
+ $O(N)$ = $O(N)$

Depth = $O(\log N)$
+ $O(1)$ = $O(\log N)$



Carry Predictor / Prefix Sum

Size = $O(N)$ binary op.s ©

Depth = $O(\log N)$

$G_i := A_i \wedge B_i$ "*generate*" new carry to stage #i+1

$P_i := A_i \vee B_i$ "*propagate*" (possible) previous carry

$$G_{i+1} = G_i \vee (P_i \wedge G_{i\text{-}1}) \quad , \qquad P_{i+1} = P_i \wedge P_{i\text{-}1}$$

Goal: Calculate $(G_i : i=1\ldots N)$ in *parallel!*

Lemma: The following operation © on *pairs* of bits

$$(P'',G'') = (P'\wedge P,\ G'\vee(P'\wedge G)) \ =: \ (P',G') \text{ © } (P,G)$$

is (a) *idempotent* and (b) *associative*.

Power of Abstraction!

$(x_1,\ldots,x_N) \rightarrow (\text{©}_{1\leq n\leq M}\, x_n \ : \ M=1\ldots N)$, © associative

Parallel Sorting, *Revisited*

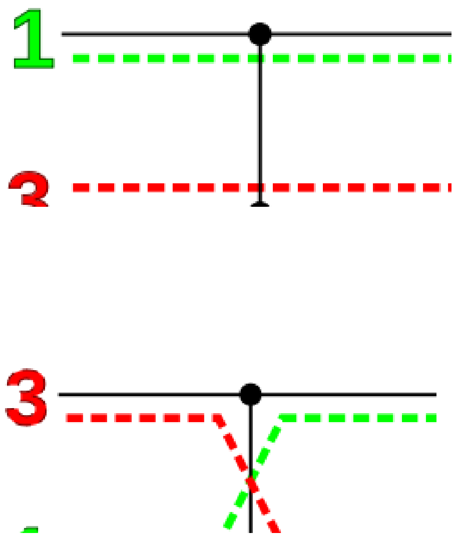
$(X,\leq)$ totally ordered set

Gate semantics ($n>m$):

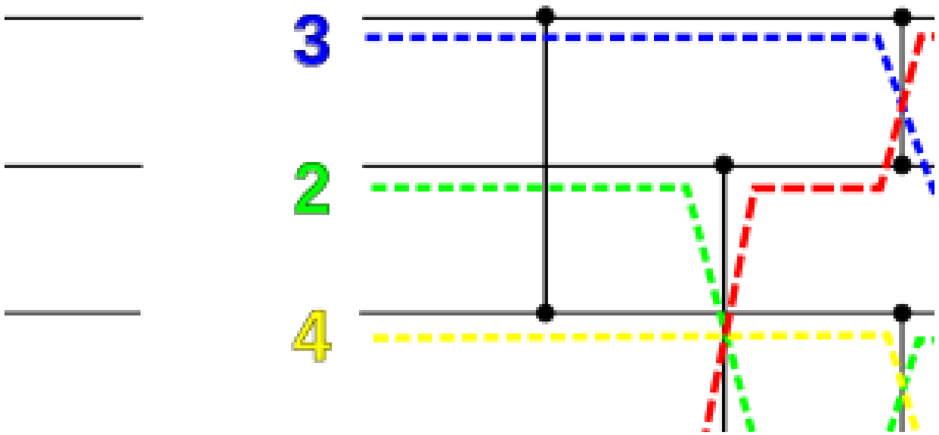
"**`If`** $x[n]<x[m]$ **`then`**

`swap` $x[n] \leftrightarrow x[m]$"

primitive/gate:



Example: *Sort4*



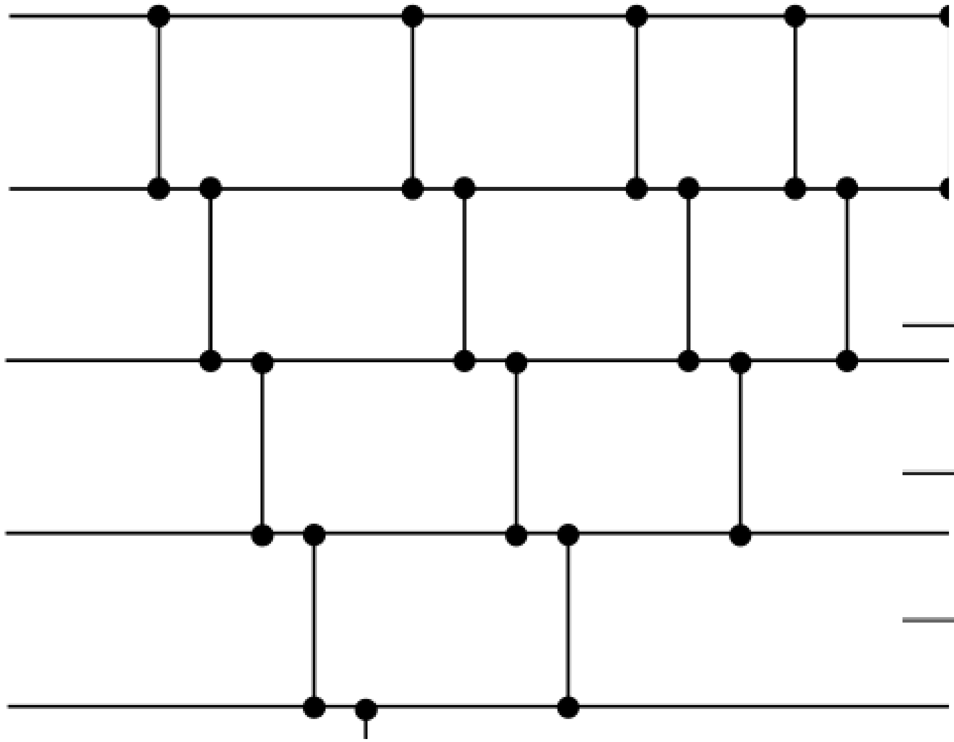
Bubble Sorting Network

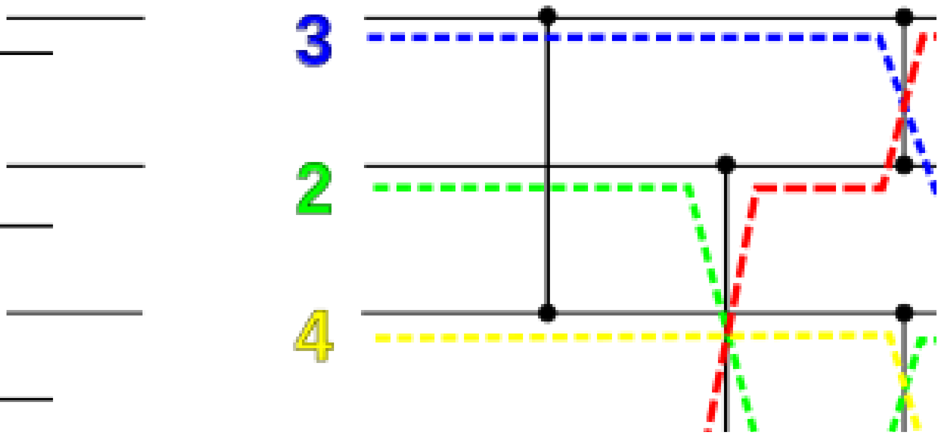
Size → opt.time $O(N \cdot \log N)$ **?** **Depth** → $O(\log N)$ **?**

(Brent's Principle)

***Bubble* Sorting Network:**

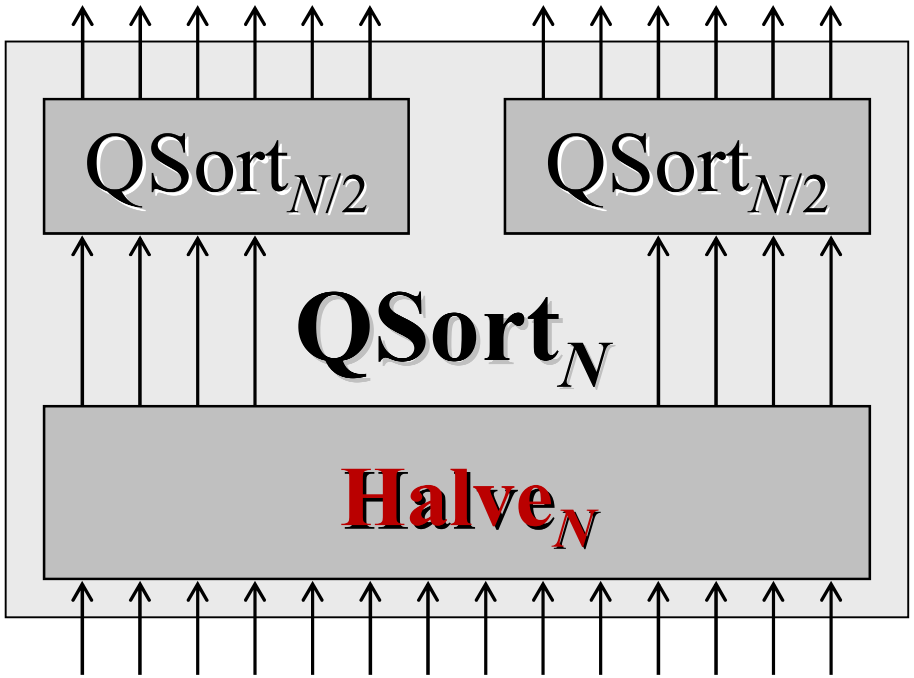
Size $O(N^2)$, Depth $O(N)$





Parallelizing *QuickSort* ?

Size → opt.time $O(N \cdot \log N)$? **Depth** → $O(\log N)$?



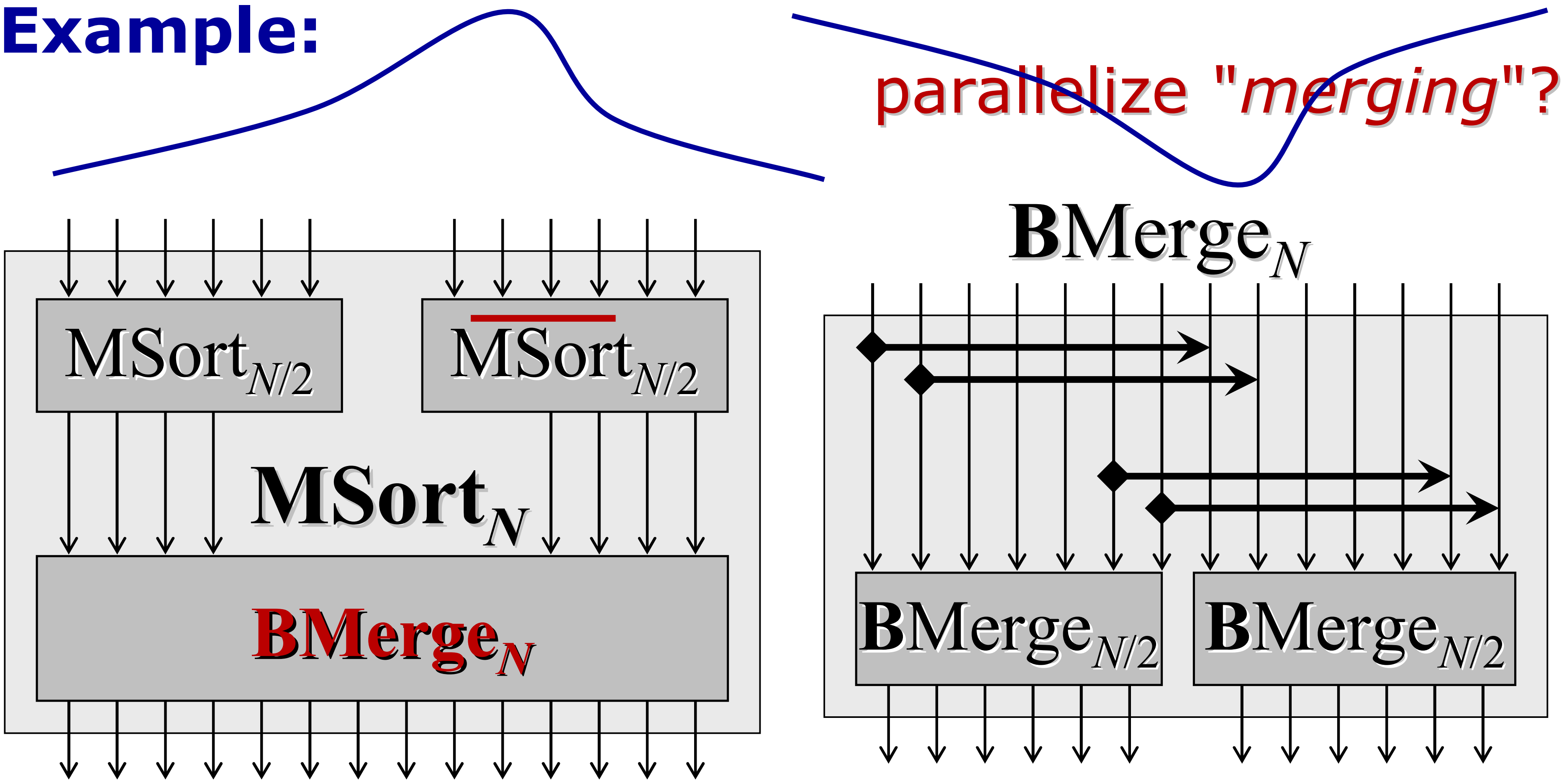
***Bubble* Sorting Network:**
Size $O(N^2)$, Depth $O(N)$

Recall: *QuickSort* requires smart "halving" **!**

Parallelizing *Mergesort* ?

Example:

parallelize "*merging*"?



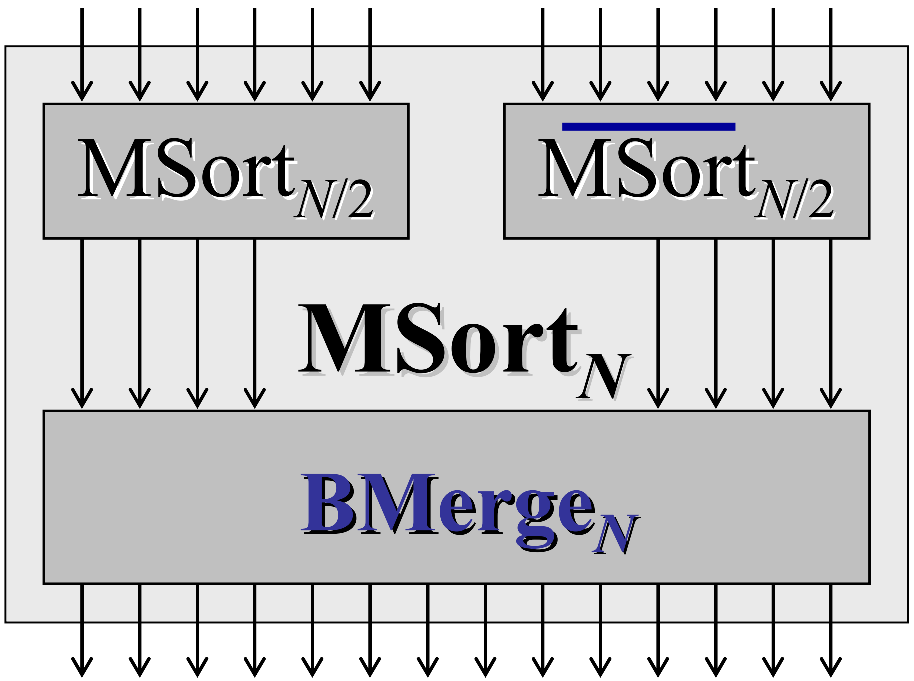
or *cyclic shift*

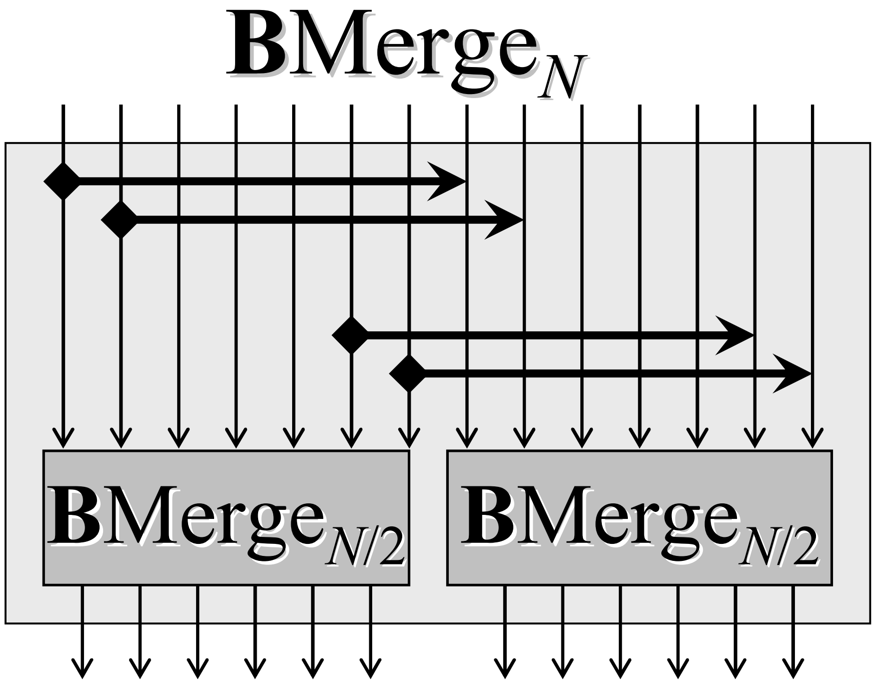
Def: Call $x[1\ldots N]$ ***bitonic*** if, for some $M \leq N$,
$x[1\ldots M]$ non-*de*creasing & $x[M+1\ldots N]$ non-*in*creasing

Batcher's Bitonic Sorter

Size = $O(N\cdot\log^2 N)$ gates → opt.time $O(N\cdot\log N)$ **?**

Depth = $O(\log^2 N)$ par.time → $O(\log N)$ **?**

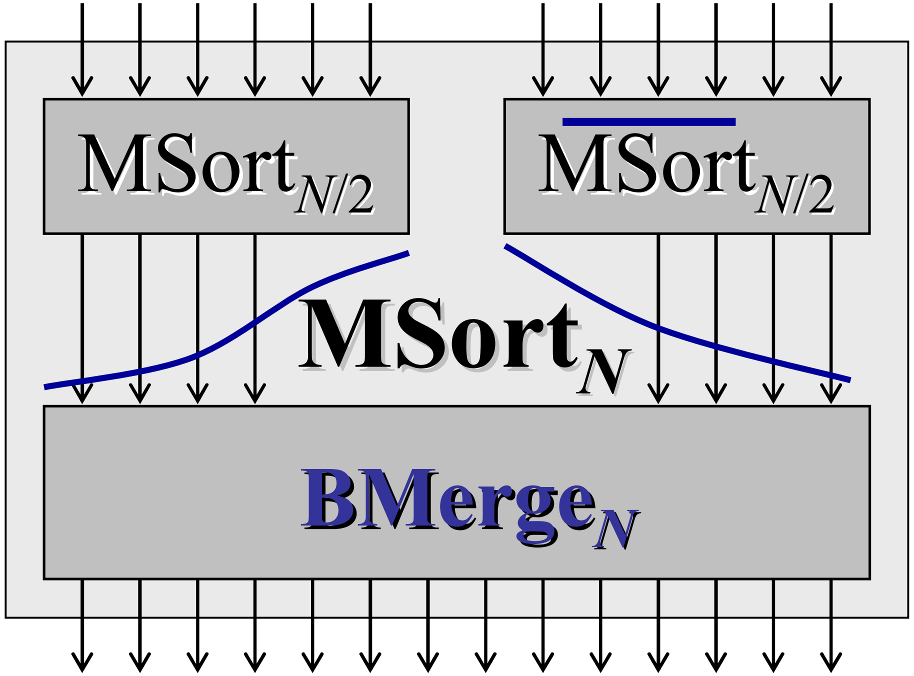


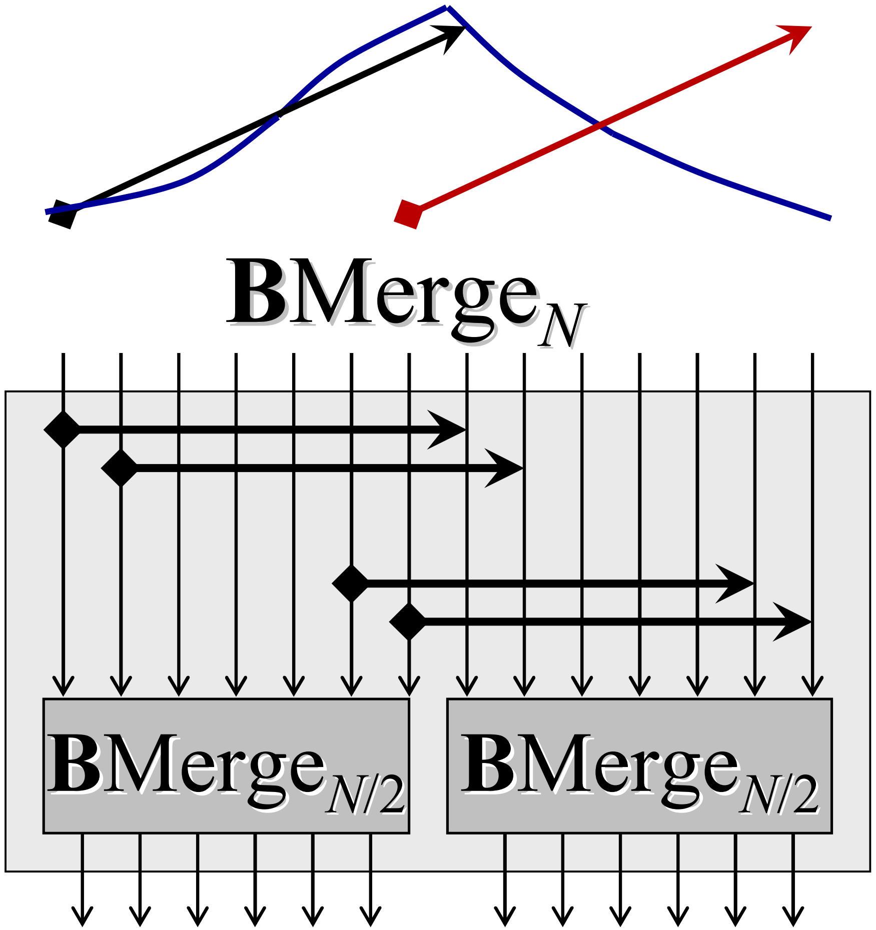


DepthMSort(N) = **1**·DepthMSort($N/2$)+DepthBMerge(N)

DepthBMerge(N) = $O(\mathbf{1})$+**1**·DepthBMerge($N/2$) = $O(\log N)$

Correctness of *Batcher*





Def: Call $x[1...N]$ ***bitonic*** if, for some $M \leq N$, (or *cyclic shift*)

$x[1...M]$ non-*de*creasing & $x[M+1...N]$ non-*in*creasing

AKS Sorting Network

Batcher's Bitonic Sorting Network

Size = $O(N\cdot\log^2 N)$ gates → opt.time $O(N\cdot\log N)$ **?**

Depth = $O(\log^2 N)$ par.time → $O(\log N)$ **?**

Ajtai, Komlós, Szemerédi (1983):

Sorting network of

Size $O(N\cdot\log N)$

Depth $O(\log N)$

§9 Summary

- Classification
- Parallel Prefix
- Graph Reachability
- Carry-Lookahead
- Sorting Networks

Mini-Quiz / Homework

a) Draw the circuit for ***Parallel Prefix*** of Size $=O(N)$ and Depth $=O(\log N)$ when $N=16$.

b) Schematically sketch the circuit for ***Graph Reachability*** of Size $=O(N^3 \cdot \log N)$ and Depth $=O(\log^2 N)$

Mini-Quiz / Homework

Prove the Lemma!

Hint: First recall and write down the definitions of "*idempotent*" and "*associative*"

Lemma: The following operation © on *pairs* of bits

$$(P'',G'') = (P' \wedge P,\ G' \vee (P' \wedge G)) \quad =: \ (P',G') \text{ © } (P,G)$$

is (a) idempotent and (b) associative.