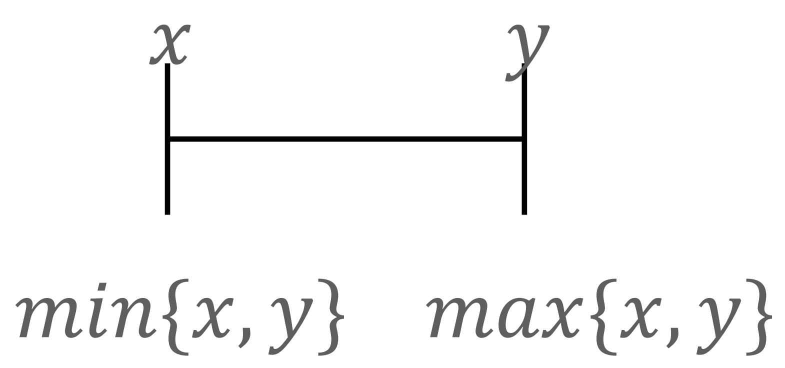
Sorting Networks

Bitonic Sort

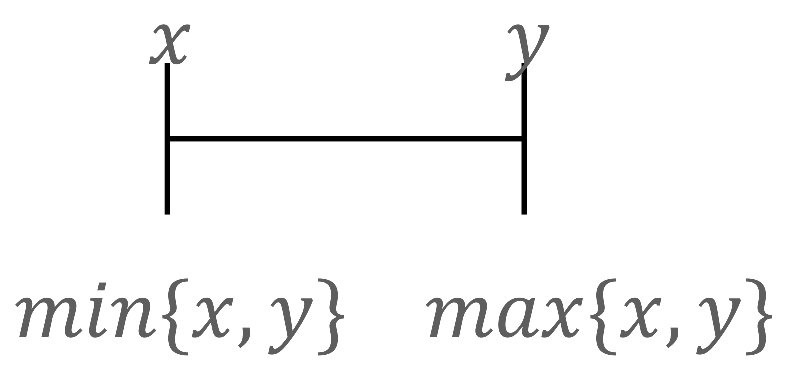
Comparator Networks

Circuits with comparators as gates

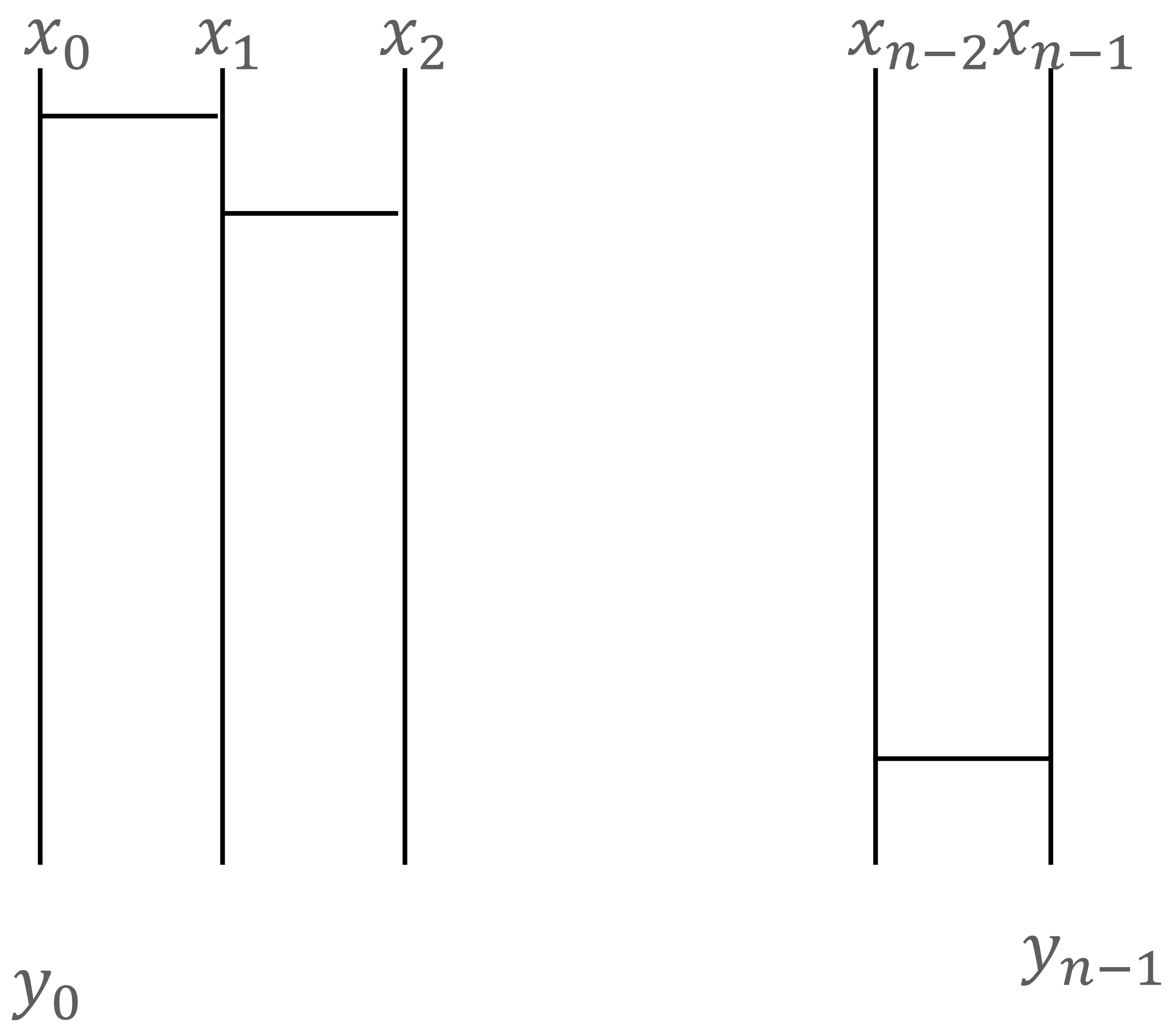


Comparator Networks

Circuits with comparators as gates



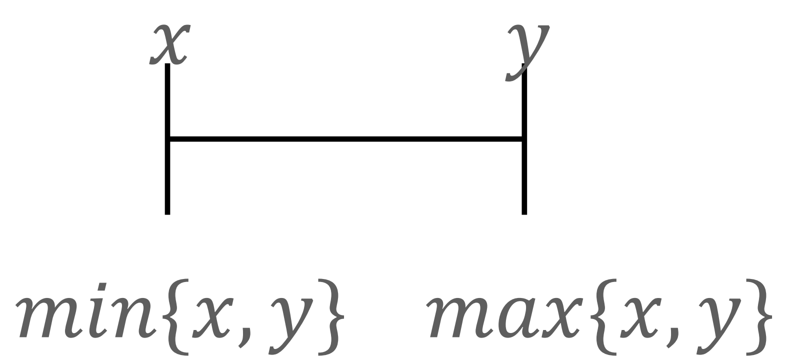
Computing the maximum: n-max



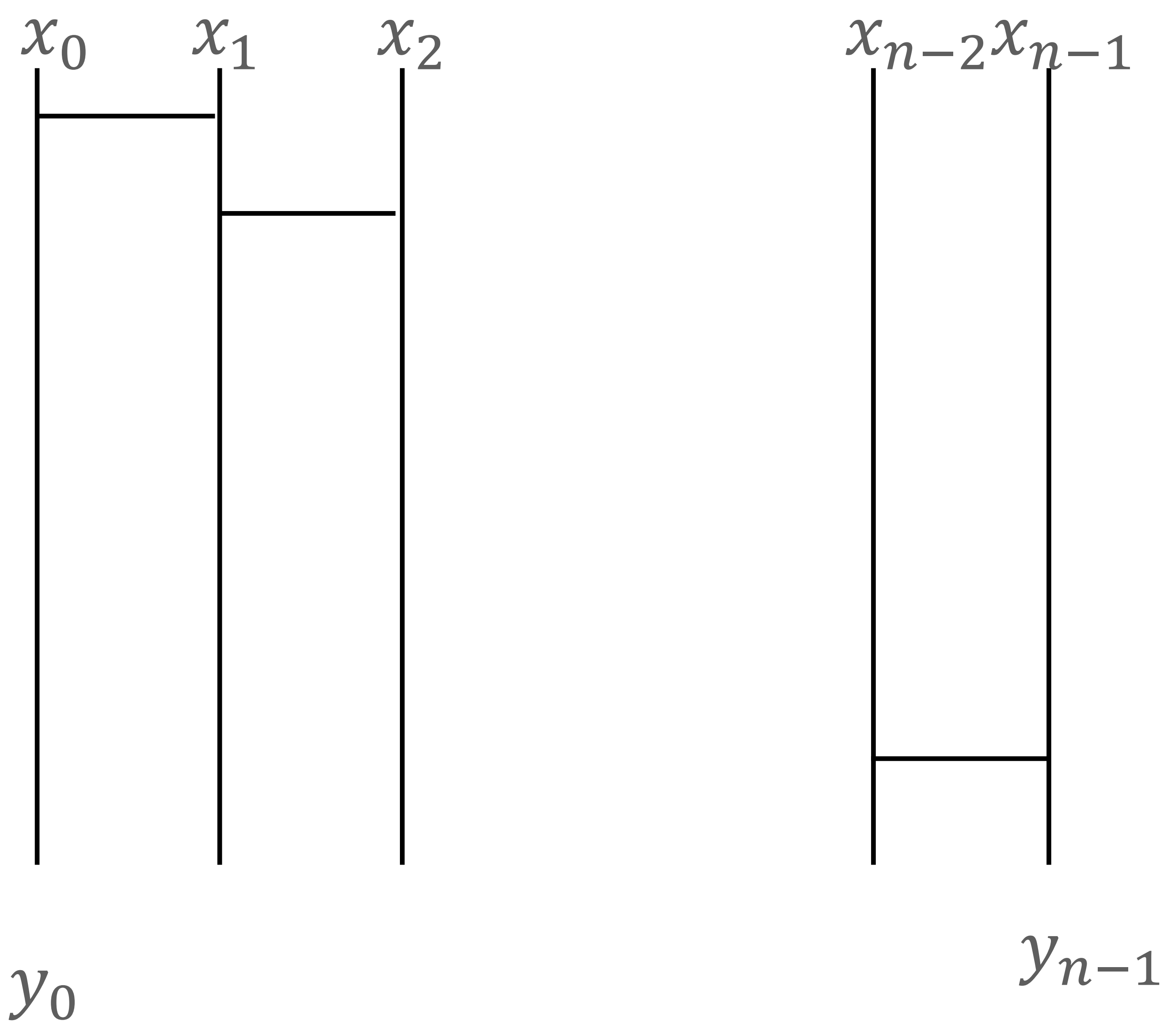
$$y_i = max\{x_j | j \leq i\}$$

naive sorting net $n - sort$

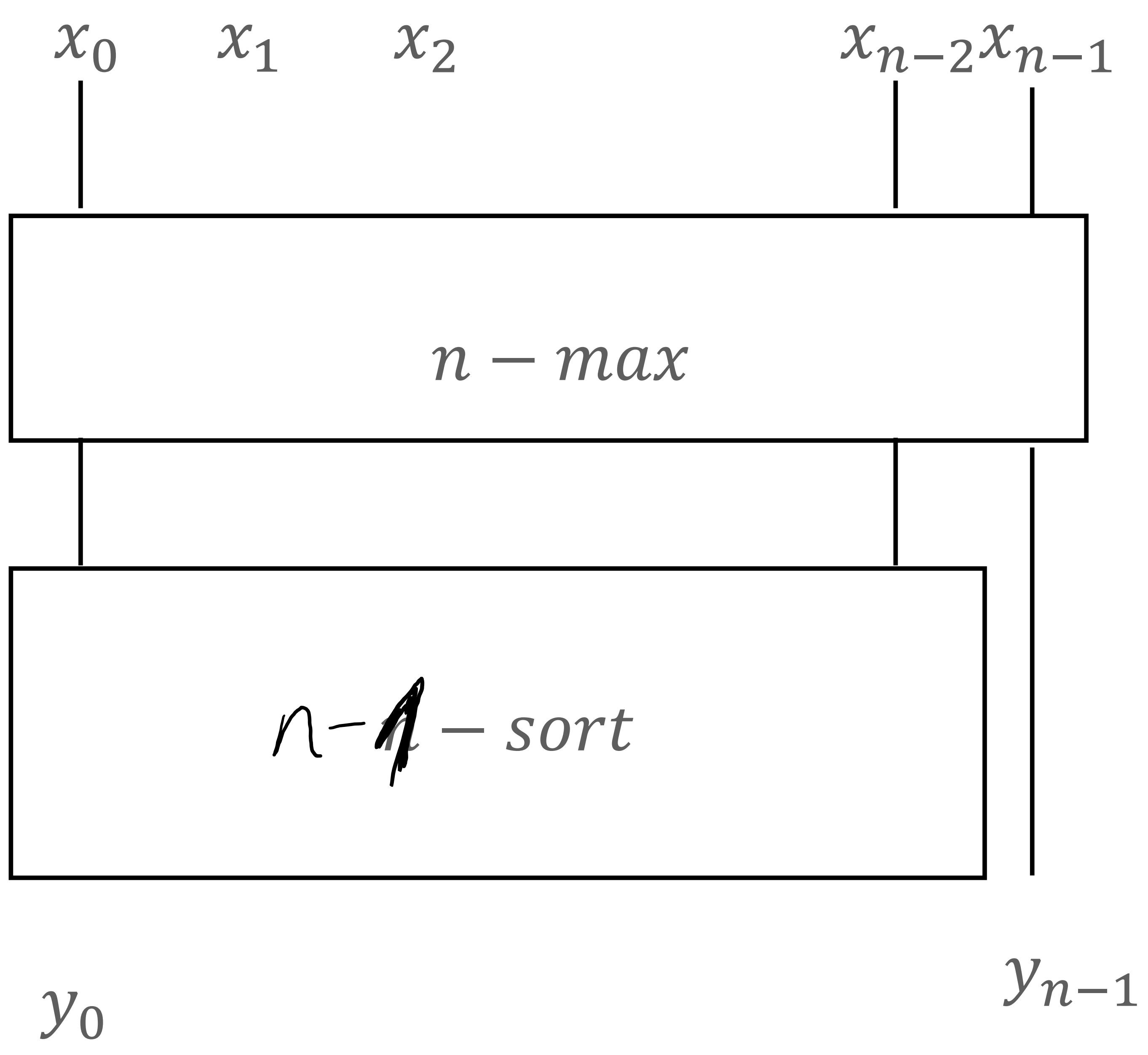
Circuits with comparators as gates



Computing the maximum: n-max



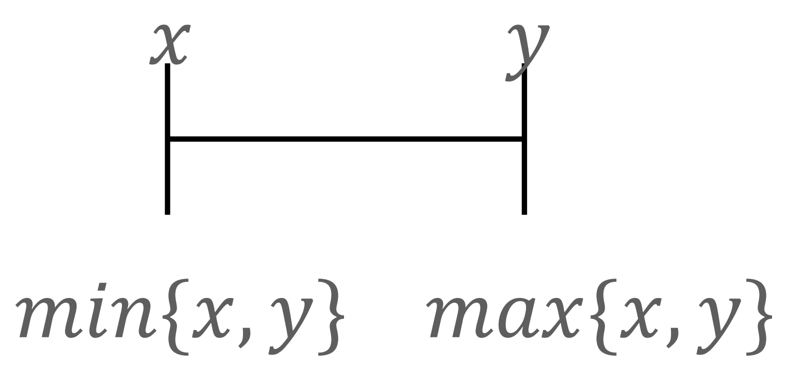
$y_i = max\{x_j | j \leq i\}$



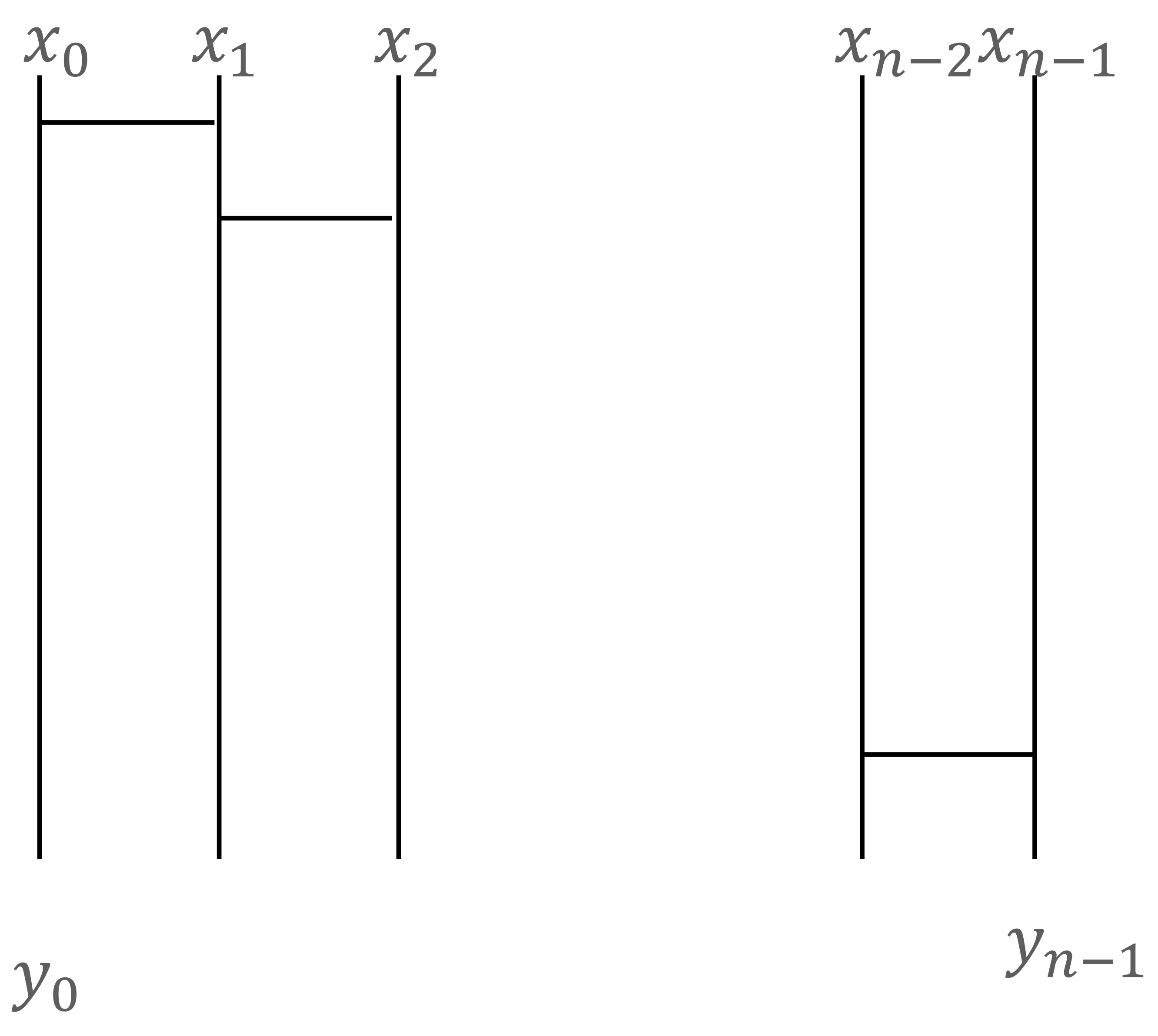
$2 - sort$: comparator

naive sorting net $n - sort$

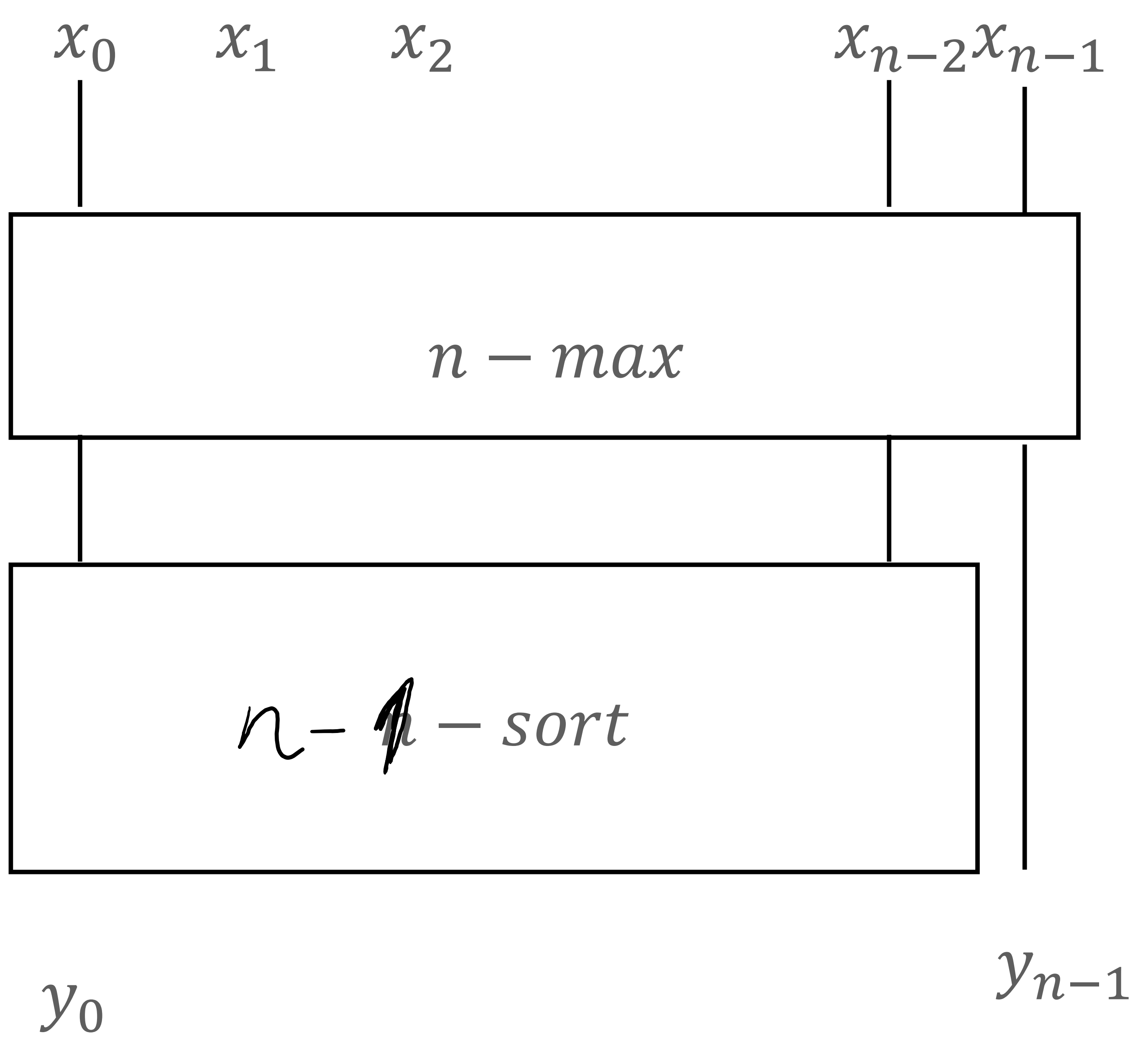
Circuits with comparators as gates



Computing the maximum: n-max



$y_i = max\{x_j | j \leq i\}$



$2 - sort$: comparator

cost = # comparators

$$c(2) = 1$$
$$c(n) = n - 1 + c(n-1)$$

$$\boxed{c(n) = O(n^2)}$$

0-1-principle

Lemma 1. *If a comparitor network N transforms input $x = x[0 : n-1]$ into output $y = y[0 : n-1]$ and*

$$f : \mathbb{N}_0 \to \mathbb{N}_0$$

is monotonous, then N transforms input

$$f(x) = (f(x_0), \ldots, f(x_{n-1}))$$

into output

$$f(y) = (f(y_0), \ldots, f(y_{n-1}))$$

0-1-principle

Lemma 1. *If a comparitor network N transforms input* $x = x[0:n-1]$ *into output* $y = y[0:n-1]$ *and*

$$f : \mathbb{N}_0 \to \mathbb{N}_0$$

is monotonous, then N transforms input

$$f(x) = (f(x_0), \ldots, f(x_{n-1}))$$

into output

$$f(y) = (f(y_0), \ldots, f(y_{n-1}))$$

By induction on depth of comparators c: if comparator c with network input x computes the pair of outputs

$$c(x) = (u, v)$$

then it computes with network input $f(x)$ the pair of outputs

$$c(f(x)) = (f(u), f(v)).$$

Details: exercise.

0-1-principle

Lemma 1. *If a comparitor network N transforms input* $x = x[0:n-1]$ *into output* $y = y[0:n-1]$ *and*

$$f : \mathbb{N}_0 \to \mathbb{N}_0$$

is monotonous, then N transforms input

$$f(x) = (f(x_0), \ldots, f(x_{n-1}))$$

into output

$$f(y) = (f(y_0), \ldots, f(y_{n-1}))$$

By induction on depth of comparators c: if comparator c with network input x computes the pair of outputs

$$c(x) = (u, v)$$

then it computes with network input $f(x)$ the pair of outputs

$$c(f(x)) = (f(u), f(v)).$$

Details: exercise.

Lemma 2. *If a sorting network N sorts all inputs*

$$x \in \mathbb{B}^n$$

then it sorts all inputs

$$y \in \mathbb{N}_0^n$$

0-1-principle

Lemma 1. *If a comparitor network N transforms input $x = x[0 : n-1]$ into output $y = y[0 : n-1]$ and*

$$f : \mathbb{N}_0 \to \mathbb{N}_0$$

is monotonous, then N transforms input

$$f(x) = (f(x_0), \dots, f(x_{n-1}))$$

into output

$$f(y) = (f(y_0), \dots, f(y_{n-1}))$$

By induction on depth of comparators c: if comparator c with network input x computes the pair of outputs

$$c(x) = (u, v)$$

then it computes with network input $f(x)$ the pair of outputs

$$c(f(x)) = (f(u), f(v)).$$

Details: exercise.

Lemma 2. *If a sorting network N sorts all inputs*

$$x \in \mathbb{B}^n$$

then it sorts all inputs

$$y \in \mathbb{N}_0^n$$

- assume N does not sort $x \in \mathbb{N}_0^n$, i.e. N produces outputs

  $$N(x) = (x_{\sigma(0)}, \dots x_{\sigma(n-1)})$$

  and there is i such that

  $$x_i < x_j \wedge \sigma(i) > \sigma(j)$$

- define monotonous function

  $$f(a) = \begin{cases} 0 & a \leq x_i \\ 1 & a > x_i \end{cases}$$

- lemma 1 $\to$ network with input $f(x)$ produces output $(f(x_{\sigma(0)}), \dots, f(x_{\sigma(n-1)}))$. Then

  $$f(x_{\sigma(i)}) = f(x_i) = \ 0 < \ 1 = f(x_j) = f(x_{\sigma(j)})$$

- $$\sigma(i) > \sigma(j) \ \to \ 0 \text{ placed to the right of } 1$$

bitonic sequences and merge stages

def:

- for even n lower and upper half of bit strings $x \in \mathbb{B}^n$

$$x_L = x[0 : n/2-1] \ , \ x_H = x[n/2 : n-1]$$

- a sequence $x \in \mathbb{B}*$ is *bitonic* if there are constants $a, b, c \in \mathbb{N}_0$ such that

$$x = 0^a 1^b 0^c \quad \text{(upward bitonic)}$$

or

$$x = 1^a 0^b 1^c \quad \text{(downward bitonic)}$$

bitonic sequences and merge stages

def:

- for even n lower and upper half of bit strings $x \in \mathbb{B}^n$

$$x_L = x[0 : n/2-1] \; , \; x_H = x[n/2 : n-1]$$

- a sequence $x \in \mathbb{B}*$ is *bitonic* if there are constants $a, b, c \in \mathbb{N}_0$ such that

$$x = 0^a 1^b 0^c \quad \text{(upward bitonic)}$$

  or

$$x = 1^a 0^b 1^c \quad \text{(downward bitonic)}$$

- for even n an *n-bitonic merge stage* has $n/2$ comparators. With input $x \in \mathbb{B}^n$ it produces output $B(x) \in \mathbb{B}^n$ with

$$B(x)[i] = \begin{cases} \min\{x_i, x_{i+n/2}\} & i < n/2 \\ \max\{x_i, x_{i-n/2}\} & i \geq n/2 \end{cases}$$

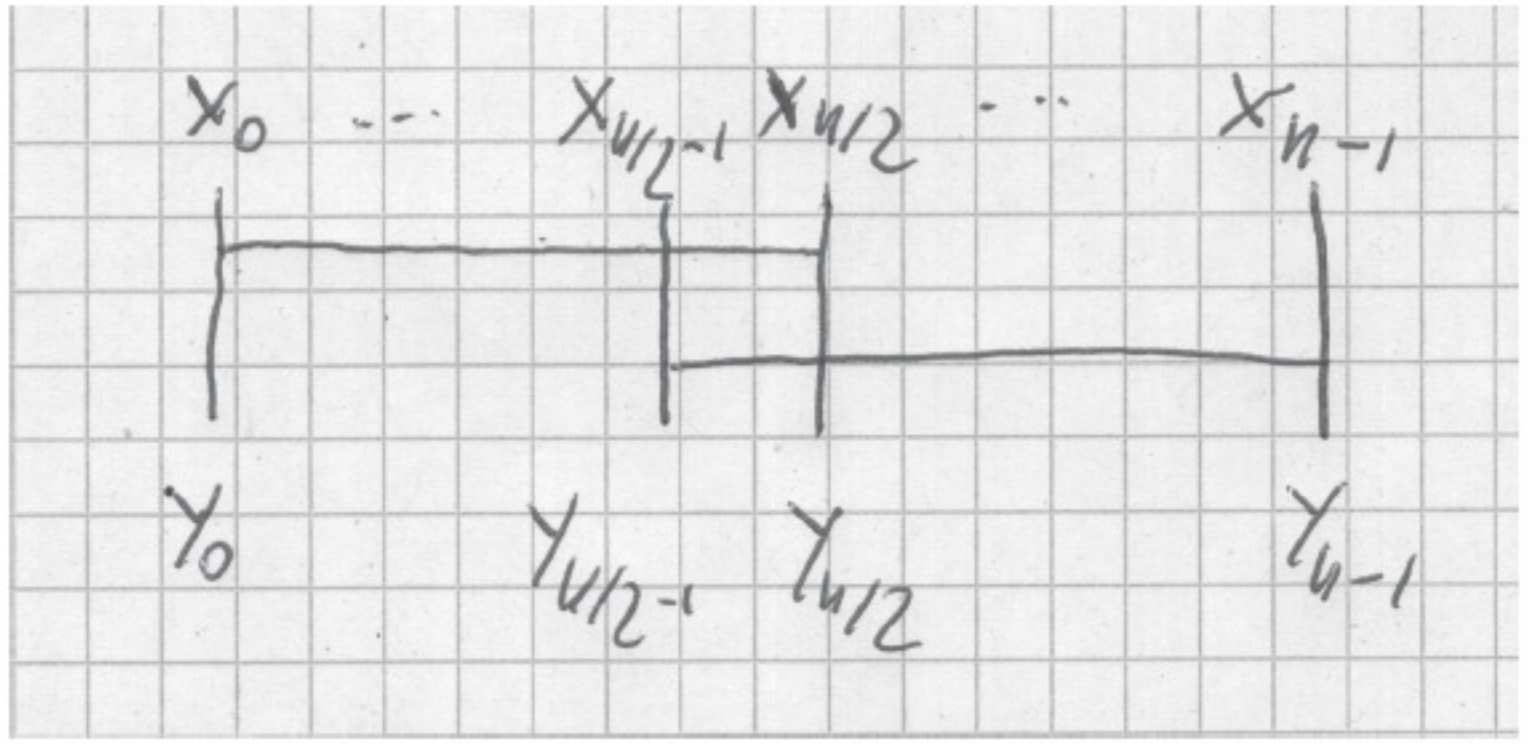


Figure 1: n-bitonic merge stage transforming input sequence x into output sequence $y = B(x)$

bitonic sequences and merge stages

def:

- for even n lower and upper half of bit strings $x \in \mathbb{B}^n$

$$x_L = x[0 : n/2-1] \ , \ x_H = x[n/2 : n-1]$$

- a sequence $x \in \mathbb{B}*$ is *bitonic* if there are constants $a, b, c \in \mathbb{N}_0$ such that

$$x = 0^a 1^b 0^c \quad \text{(upward bitonic)}$$

  or

$$x = 1^a 0^b 1^c \quad \text{(downward bitonic)}$$

- for even n an n-*bitonic merge stage* has $n/2$ comparators. With input $x \in \mathbb{B}^n$ it produces output $B(x) \in \mathbb{B}^n$ with

$$B(x)[i] = \begin{cases} \min\{x_i, x_{i+n/2}\} & i < n/2 \\ \max\{x_i, x_{i-n/2}\} & i \geq n/2 \end{cases}$$

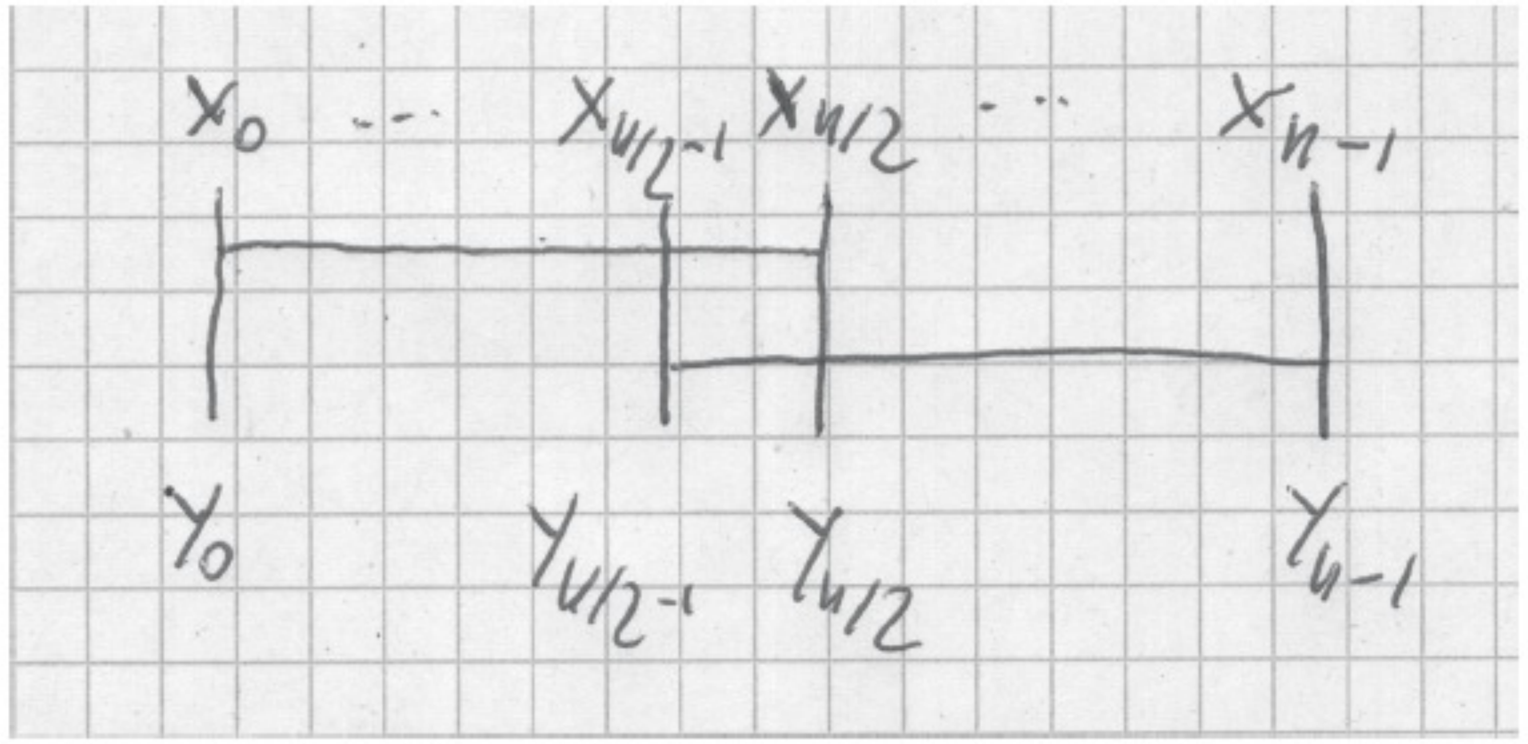


Figure 1: n-bitonic merge stage transforming input sequence x into output sequence $y = B(x)$

Lemma 3. *If $x \in \mathbb{B}^n$ is bitonic, then*

- *either $B(x)_L$ is bitonic and $B(x)_H = 1^{n/2}$*
- *or $B(x)_L = 0^{n/2}$ and $B(x)_H$ is bitonic*

bitonic sequences and merge stages

def:

- for even n lower and upper half of bit strings $x \in \mathbb{B}^n$

$$x_L = x[0 : n/2-1]\ ,\ x_H = x[n/2 : n-1]$$

- a sequence $x \in \mathbb{B}*$ is *bitonic* if there are constants $a, b, c \in \mathbb{N}_0$ such that

$$x = 0^a 1^b 0^c \quad \text{(upward bitonic)}$$

or

$$x = 1^a 0^b 1^c \quad \text{(downward bitonic)}$$

- for even n an *n-bitonic merge stage* has $n/2$ comparators. With input $x \in \mathbb{B}^n$ it produces output $B(x) \in \mathbb{B}^n$ with

$$B(x)[i] = \begin{cases} \min\{x_i, x_{i+n/2}\} & i < n/2 \\ \max\{x_i, x_{i-n/2}\} & i \geq n/2 \end{cases}$$

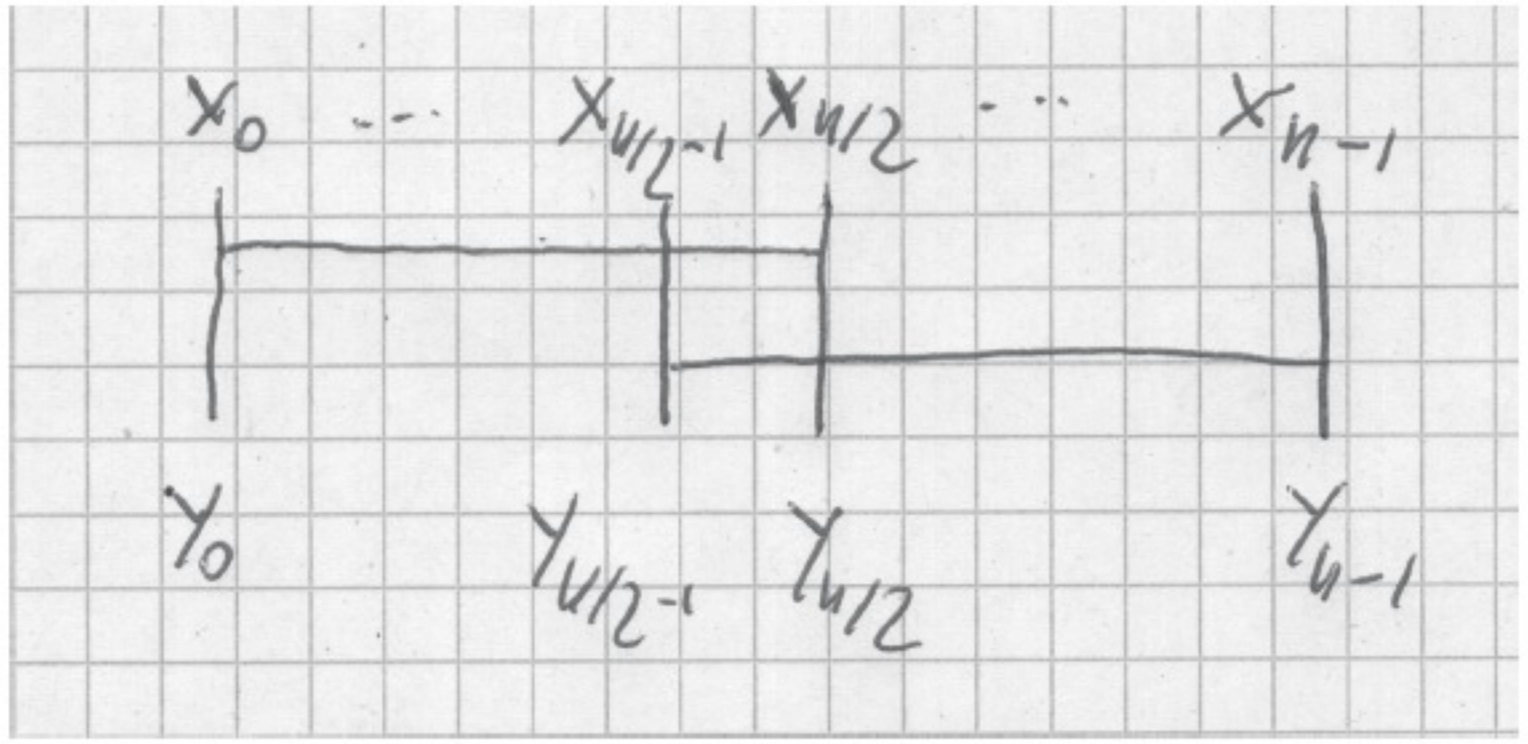


Figure 1: n-bitonic merge stage transforming input sequence x into output sequence $y = B(x)$

Lemma 3. *If $x \in \mathbb{B}^n$ is bitonic, then*

- *either $B(x)_L$ is bitonic and $B(x)_H = 1^{n/2}$*
- *or $B(x)_L = 0^{n/2}$ and $B(x)_H$ is bitonic*

Proof. Cases in obvious case split:

- upward or downward bitonic?
- more ones or more zeros or equal numbers?
- $a+b \leq n/2$? If $a+b > n/2$ consider region $\{i \mid x[0] = x[i] = x[i+n/2]\}$

cases usually not worked out in the literature.

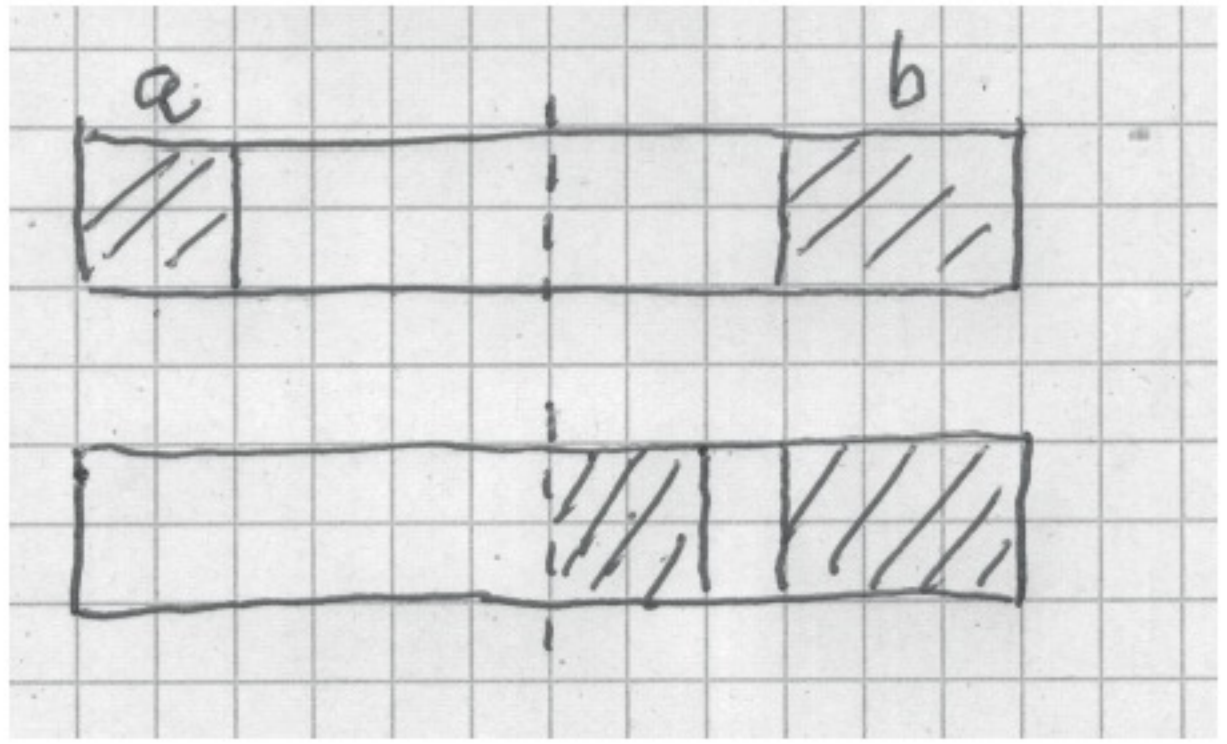


Figure 2: Striped regions are ones. Case: downward bitonic, more zeros than ones, $a+b < n/2$. $y_L = 0^{n/2}$ and y_H is (downward) bitonic.

modulo computation and cyclic shifts

reminder: for $a, b \in \mathbb{Z}$

- congruence relation modulo $k \in \mathbb{N}$

$$a \equiv b \bmod k \leftrightarrow \exists k \in \mathbb{Z}.\ a - b = z \cdot k$$

- the function $(x \bmod k)$

$$x = (a \bmod k) \leftrightarrow x \equiv a \bmod k \wedge x \in [0 : k-1]$$

modulo computation and cyclic shifts

reminder: for $a, b \in \mathbb{Z}$

- congruence relation modulo $k \in \mathbb{N}$

$$a \equiv b \bmod k \;\leftrightarrow \exists k \in \mathbb{Z}.\; a - b = z \cdot k$$

- the function $(x \bmod k)$

$$x = (a \bmod k) \leftrightarrow x \equiv a \bmod k \;\wedge\; x \in [0 : k-1]$$

Lemma 4. *For even k*

1.

$$a \equiv b \bmod k \rightarrow a \equiv b \bmod k/2$$

2.

$$x = (a \bmod k) \;\wedge\; x \in [0 : k/2 - 1] \;\rightarrow\; x = (a \bmod (k/2))$$

Proof. exercise

modulo computation and cyclic shifts

reminder: for $a, b \in \mathbb{Z}$

- congruence relation modulo $k \in \mathbb{N}$

$$a \equiv b \bmod k \leftrightarrow \exists k \in \mathbb{Z}.\ a - b = z \cdot k$$

- the function $(x \bmod k)$

$$x = (a \bmod k) \leftrightarrow x \equiv a \bmod k \ \wedge\ x \in [0 : k-1]$$

Lemma 4. *For even k*

1.

$$a \equiv b \bmod k \rightarrow a \equiv b \bmod k/2$$

2.

$$x = (a \bmod k) \ \wedge\ x \in [0 : k/2 - 1] \ \rightarrow\ x = (a \bmod (k/2))$$

Proof. exercise

cyclic shifts

For shift operand $x = x[0 : n-1] \in \mathbb{B}^n$ and shift distance $c \in [0 : n-1|$

- cyclic right shifts

$$src(x, c) = x[n-c : n-1] \circ x[0 : n-c-1]$$

Bit i is shifted to position $i + c \bmod n$:

$$x[i] = src(x, c)[i + c \bmod n]$$

- cyclic left shifts

$$slc(x, c) = x[c : n-1] \circ x[0 : c-1]$$

Bit i is shifted to position $i - c \bmod n$:

$$x[i] = slc(x, c)[i - c \bmod n]$$

bitonic merge stage with pre-shift and post-shift

Lemma 5. *You can compute $B(x)$ by*

- *preshifting x*

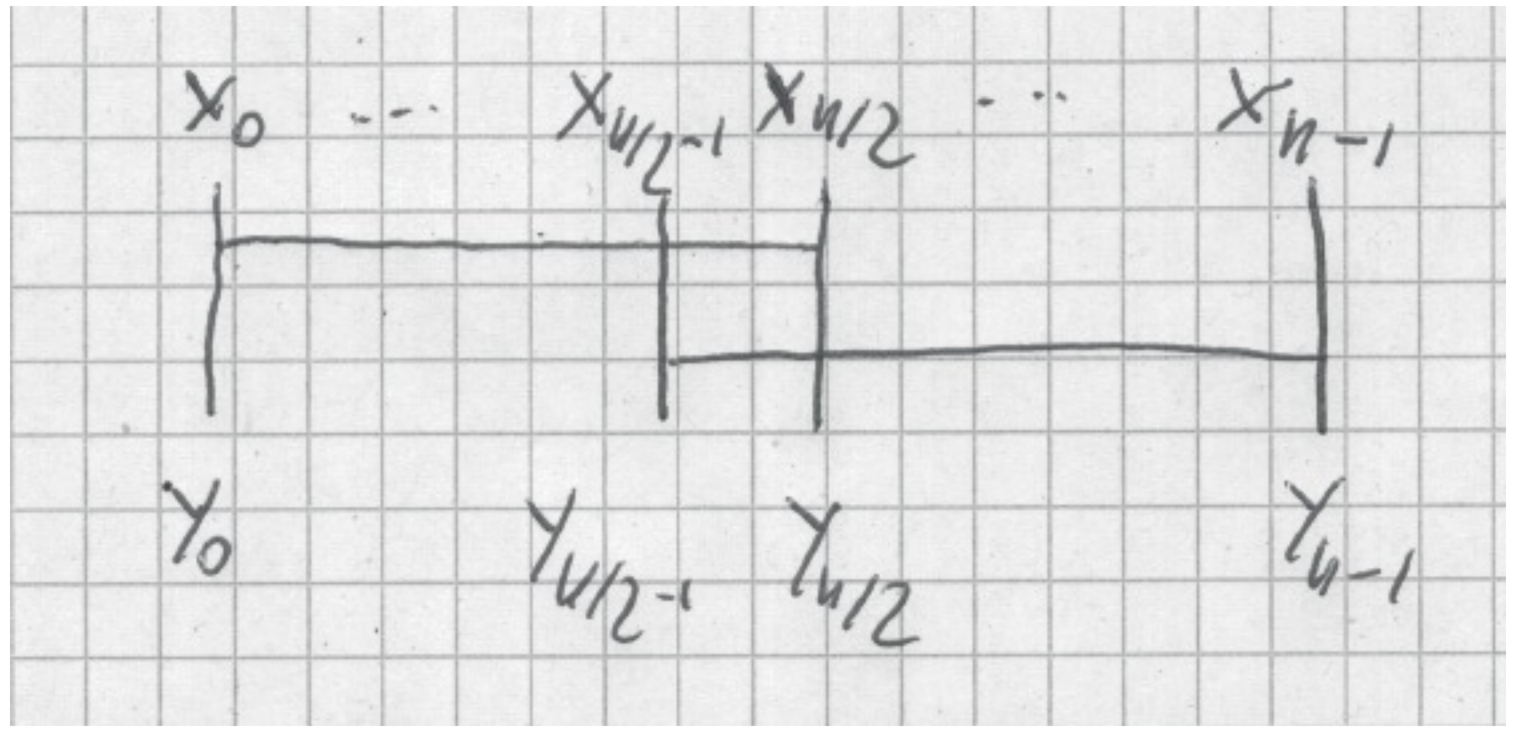
$$y = src(x, c)$$

- *applying the bitonic sorting stage to the shifted operand*

$$z = B(y)$$

- *shift upper and lower half back (operands in $\mathbb{B}^{n/2}$)*

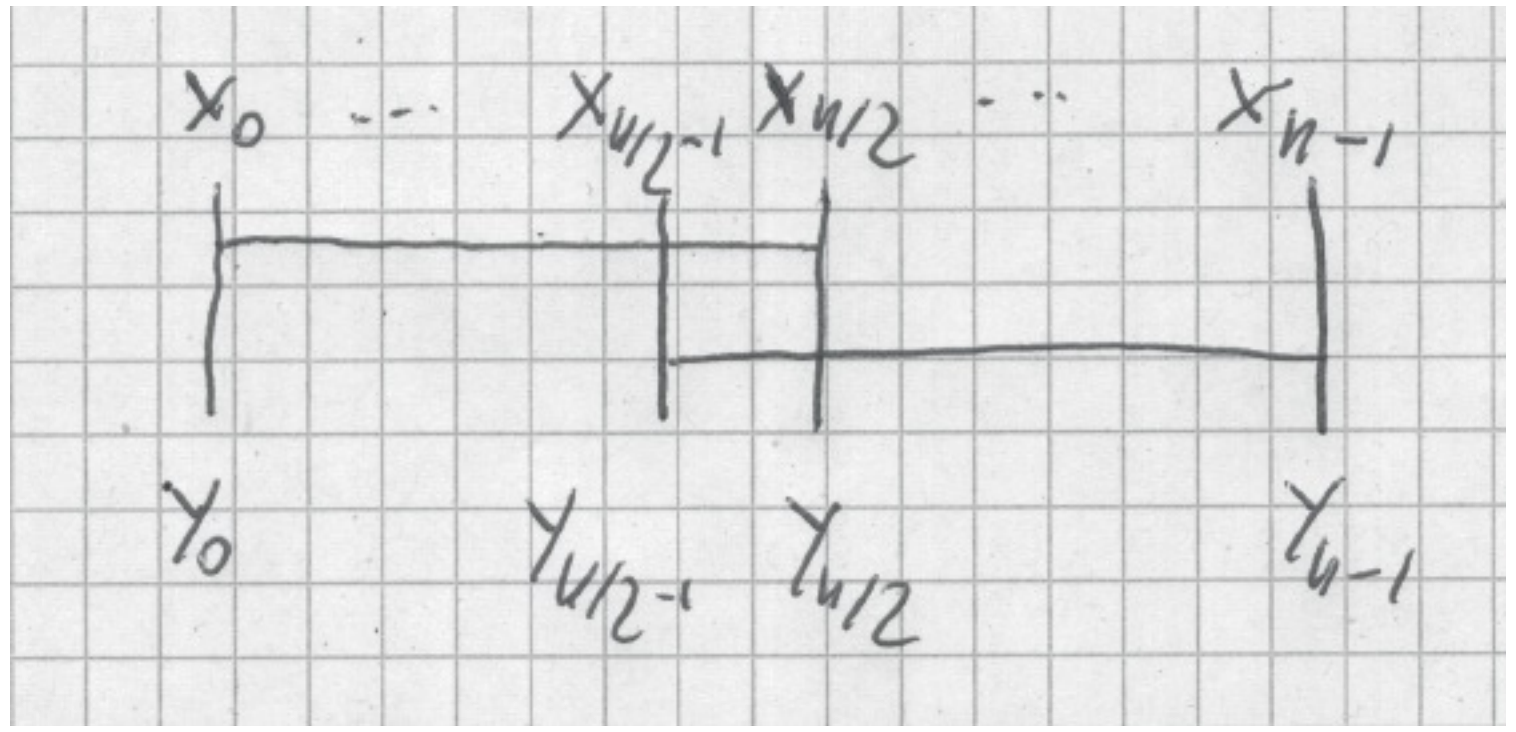
$$B(x) = slc(B(y)_L, c \bmod n/2) \circ slc(B(y)_H, c \bmod n/2)$$



bitonic merge stage with pre-shift and post-shift

Lemma 5. *You can compute $B(x)$ by*

- *preshifting x*
$$y = src(x,c)$$
- *applying the bitonic sorting stage to the shifted operand*
$$z = B(y)$$
- *shift upper and lower half back (operands in $\mathbb{B}^{n/2}$)*
$$B(x) = slc(B(y)_L, c \bmod n/2) \circ slc(B(y)_H, c \bmod n/2)$$



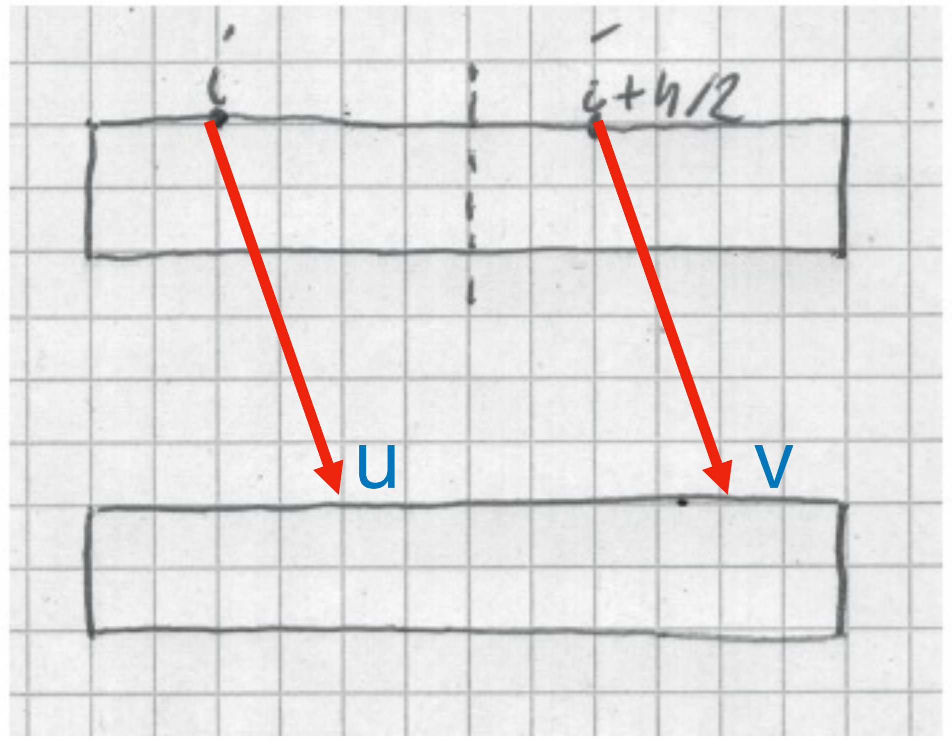
For $i < n/2$ we track pairs $x_i, x_{i+n/2}$ of inputs, which are compared in a sorting stage. They keep distance $n/2$. One ends up in the upper half, the other in the lower half of y

Lemma 6. *Let*

$$\begin{aligned} u &= \min\{i+c \bmod n, i+n/2+c \bmod n) \\ v &= \max\{i+c \bmod n, i+n/2+c \bmod n) \end{aligned}$$

Then

$$\begin{aligned} u &= i+c \bmod (n/2) \\ v &= n/2+u \end{aligned}$$



bitonic merge stage with pre-shift and post-shift

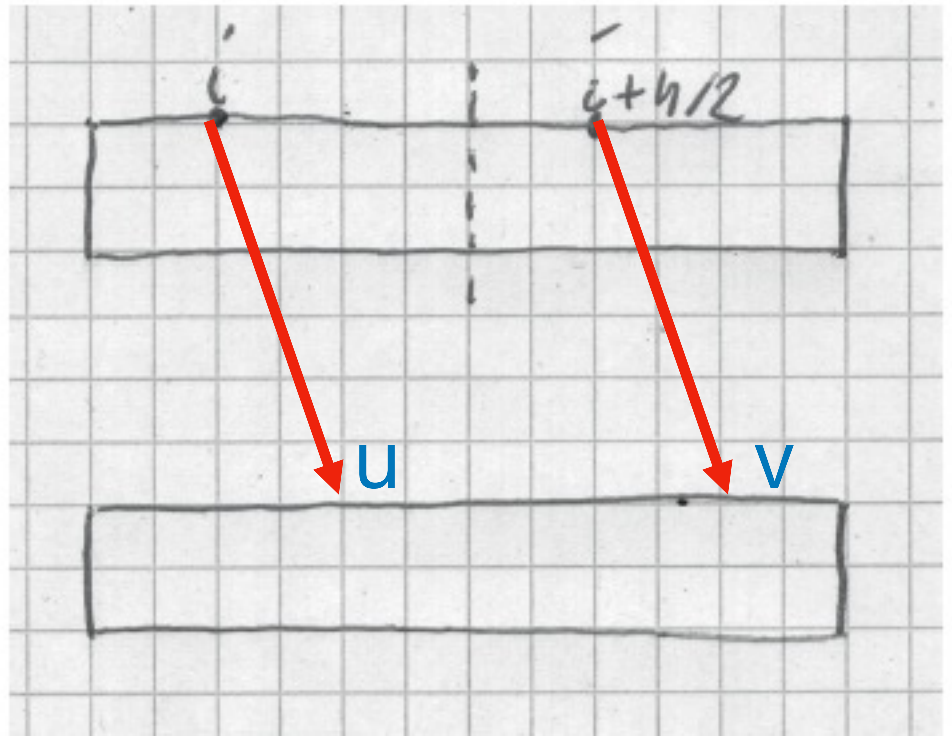
For $i < n/2$ we track pairs $x_i, x_{i+n/2}$ of inputs, which are compared in a sorting stage. They keep distance $n/2$. One ends up in the upper half, the other in the lower half of y

Lemma 6. *Let*

$$\begin{aligned} u &= \min\{i+c \bmod n, i+n/2+c \bmod n) \\ v &= \max\{i+c \bmod n, i+n/2+c \bmod n) \end{aligned}$$

Then

$$\begin{aligned} u &= i+c \bmod (n/2) \\ v &= n/2+u \end{aligned}$$



case split on position of $i+c$.

- $i+c \in [0 : n/2-1]$. Then $i+n/2+c \in [n/2 : n-1]$.

$$\begin{aligned} i+c \quad \bmod n &= i+c \in [0 : n/2-1] \\ i+c+n/2 \bmod n &= i+c+n/2 \in [n/2 : n-1] \end{aligned}$$

$$\begin{aligned} u &= i+c \\ &= i+c \quad \bmod (n/2) \\ v &= i+n/2+c \\ &= n/2+u \end{aligned}$$

bitonic merge stage with pre-shift and post-shift

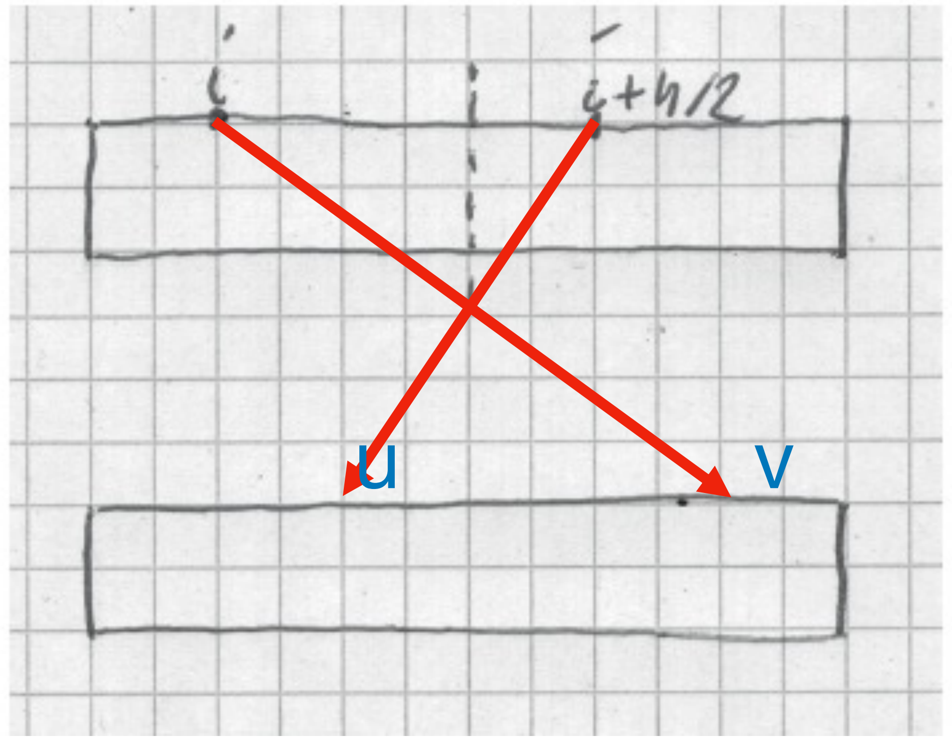
For $i < n/2$ we track pairs $x_i, x_{i+n/2}$ of inputs, which are compared in a sorting stage. They keep distance $n/2$. One ends up in the upper half, the other in the lower half of y

Lemma 6. *Let*

$$\begin{aligned} u &= \min\{i+c \bmod n, i+n/2+c \bmod n\} \\ v &= \max\{i+c \bmod n, i+n/2+c \bmod n\} \end{aligned}$$

Then

$$\begin{aligned} u &= i+c \bmod (n/2) \\ v &= n/2+u \end{aligned}$$



- $i+c \in [n/2 : n-1]$. Then $i+n/2+c \in [n-1 : 3n/2-1]$.

$$\begin{aligned} i+c \quad \bmod n &= i+c \in [n/2 : n-1] \\ i+c+n/2 \bmod n &= i+c+n/2-n \in [0 : n/2-1] \end{aligned}$$

$$\begin{aligned} u &= i+c-n/2 \\ &= i+c \quad \bmod (n/2) \\ v &= i+c \\ &= n/2+u \end{aligned}$$

bitonic merge stage with pre-shift and post-shift

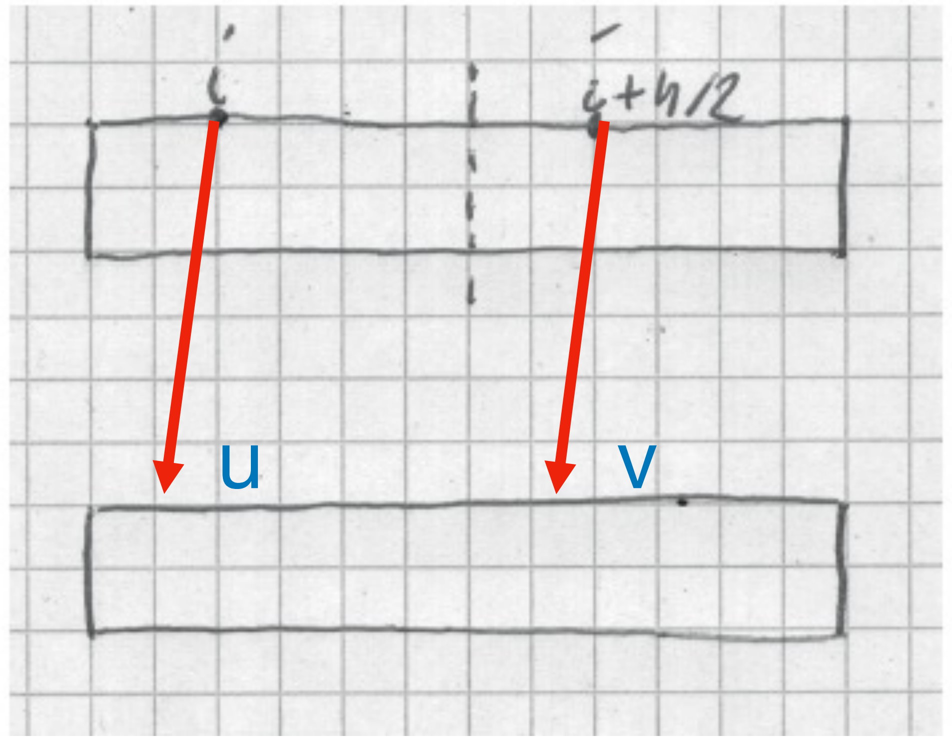
For $i < n/2$ we track pairs $x_i, x_{i+n/2}$ of inputs, which are compared in a sorting stage. They keep distance $n/2$. One ends up in the upper half, the other in the lower half of y

Lemma 6. *Let*

$$\begin{aligned} u &= \min\{i+c \bmod n, i+n/2+c \bmod n) \\ v &= \max\{i+c \bmod n, i+n/2+c \bmod n) \end{aligned}$$

Then

$$\begin{aligned} u &= i+c \bmod (n/2) \\ v &= n/2+u \end{aligned}$$



- $i+c \in [n : 3n/2-1]$. Then $i+n/2+c \in [2n-1 : 3n/2-1]$.

$$\begin{aligned} i+c \quad \bmod n &= i+c-n \in [0 : n/2-1] \\ i+c+n/2 \bmod n &= i+c+n/2-n \in [n/2 : n-1] \end{aligned}$$

$$\begin{aligned} u &= i+c-n \\ &= i+c \quad \bmod (n/2) \\ v &= i-n/2+c \\ &= n/2+u \end{aligned}$$

bitonic merge stage with pre-shift and post-shift

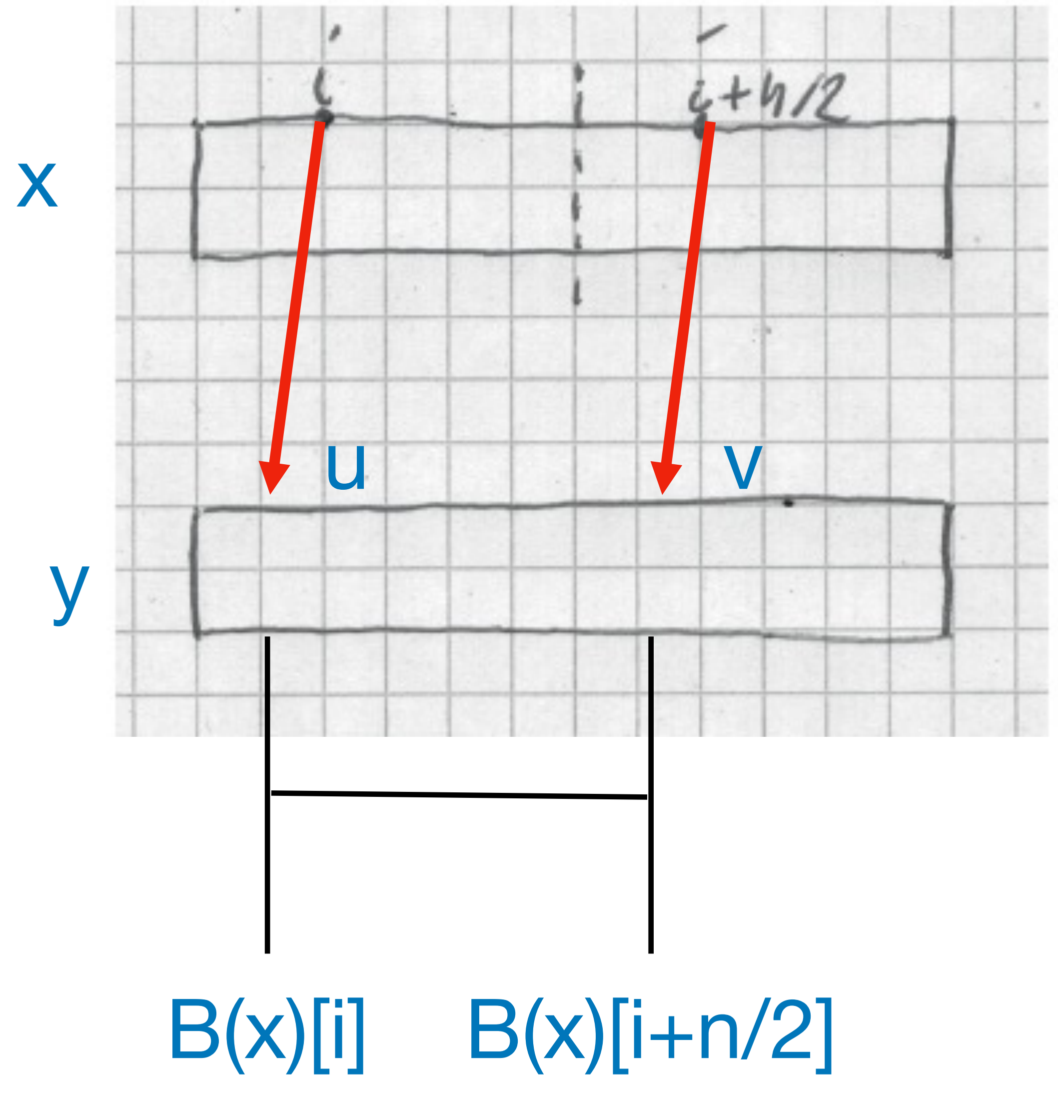
For $i < n/2$ we track pairs $x_i, x_{i+n/2}$ of inputs, which are compared in a sorting stage. They keep distance $n/2$. One ends up in the upper half, the other in the lower half of y

Lemma 6. *Let*

$$\begin{aligned} u &= \min\{i+c \bmod n, i+n/2+c \bmod n) \\ v &= \max\{i+c \bmod n, i+n/2+c \bmod n) \end{aligned}$$

Then

$$\begin{aligned} u &= i+c \bmod (n/2) \\ v &= n/2+u \end{aligned}$$



Because the same sequence elements as in input sequence x get compared in sequence y we get for $i < n/2$:

$$\begin{aligned} B(y)[u] &= \min\{y[u], y[u+n/2]\} \\ &= \min\{y[i+c \bmod n], y[i+n/2+c \bmod n]\} \\ &= \min\{x[i], x[i+n/2]\} \\ &= B(x)[i] \\ B(y)[v] &= \max\{y[u], y[u+n/2]\} \\ &= \max\{y[i+c \bmod n], y[i+n/2+c \bmod n]\} \\ &= \max\{x[i], x[i+n/2]\} \\ &= B(x)[i+n/2] \text{ similarly} \end{aligned}$$

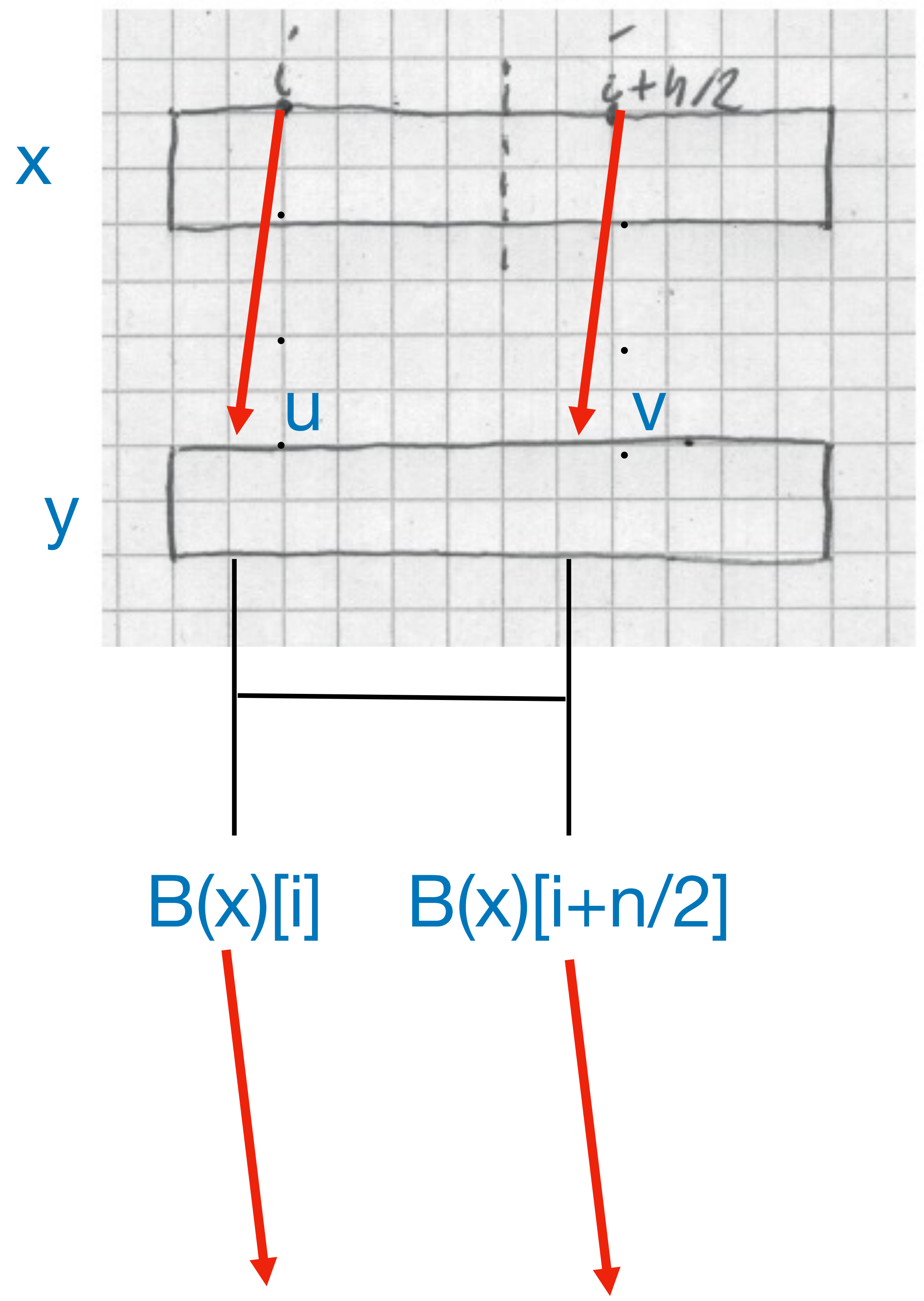
For $i < n/2$ we track pairs $x_i, x_{i+n/2}$ of inputs, which are compared in a sorting stage. They keep distance $n/2$. One ends up in the upper half, the other in the lower half of y

Lemma 6. *Let*

$$\begin{aligned} u &= \min\{i+c \bmod n, i+n/2+c \bmod n) \\ v &= \max\{i+c \bmod n, i+n/2+c \bmod n) \end{aligned}$$

Then

$$\begin{aligned} u &= i+c \bmod (n/2) \\ v &= n/2+u \end{aligned}$$



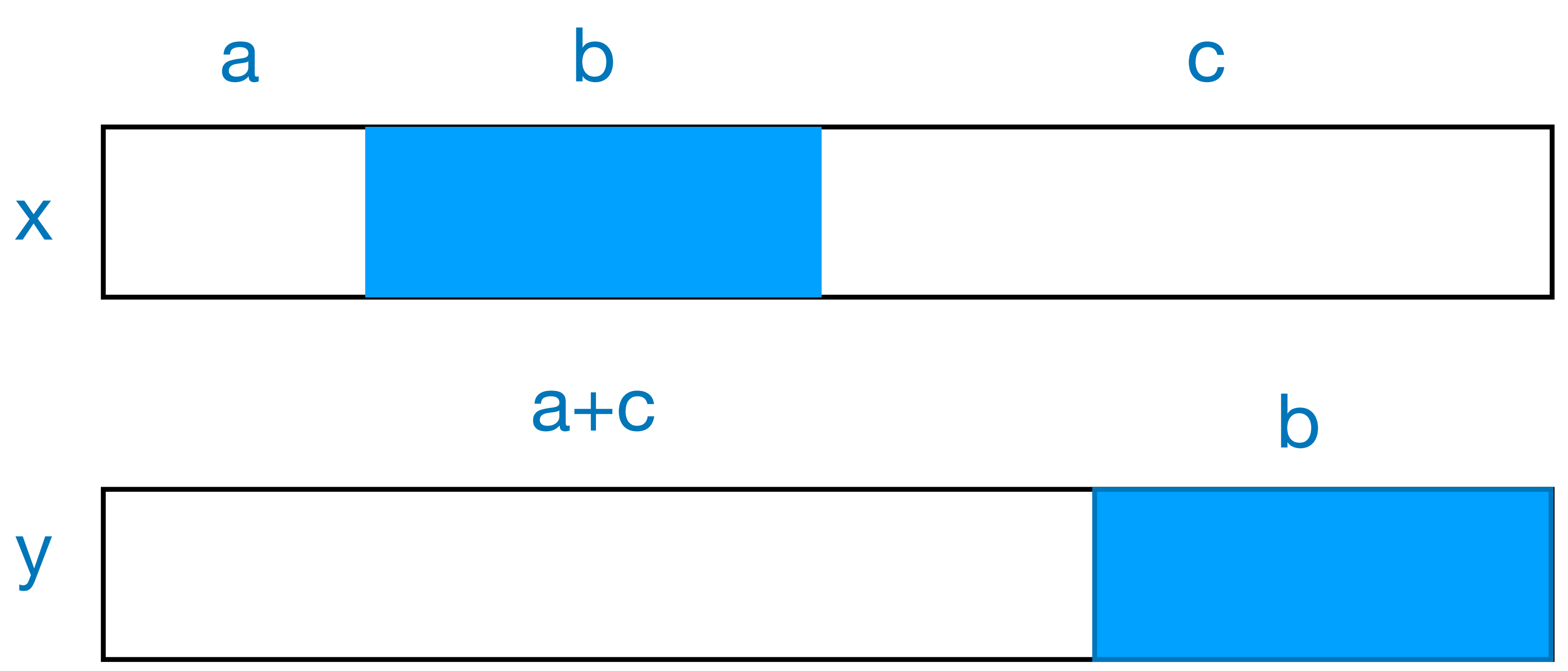
Because the same sequence elements as in input sequence x get compared in sequence y we get for $i < n/2$:

$$\begin{aligned} B(y)[u] &= \min\{y[u], y[u+n/2]\} \\ &= \min\{y[i+c \bmod n], y[i+n/2+c \bmod n]\} \\ &= \min\{x[i], x[i+n/2]\} \\ &= B(x)[i] \\ B(y)[v] &= \max\{y[u], y[u+n/2]\} \\ &= \max\{y[i+c \bmod n], y[i+n/2+c \bmod n]\} \\ &= \max\{x[i], x[i+n/2]\} \\ &= B(x)[i+n/2] \text{ similarly} \end{aligned}$$

Shiftling cyclically left by distance c gives for $i < n/2$:

$$\begin{aligned} slc(B(y)_L, c)[i] &= B(y)_L[i+c \bmod (n/2)] \\ &= B(y)_L[u] \\ &= B(x)[i] \\ slc(B(y)_H, c)[i] &= B(y)_H[i+c \bmod (n/2)] \\ &= B(y)_H[u] \\ &= B(y)[n/2+u] \\ &= B(y)[v] \\ &= B(x)[i+n/2] \end{aligned}$$

bitonic merge stage with pre-shift and post-shift



Lemma 3. *If $x \in \mathbb{B}^n$ is bitonic, then*

- *either $B(x)_L$ is bitonic and $B(x)_H = 1^{n/2}$*
- *or $B(x)_L = 0^{n/2}$ and $B(x)_H$ is bitonic*

proof of lemma 3:

Shift input x such that all ones are ate the right border

- if x is upward bitonic shift cyclically right by c

$$y = src(x, c)$$

- if x is downward bitonic shift cyclically left by a which is the same as a cyclical right shift by $n - a$

$$y = src(x, n - a)$$

bitonic merge stage with pre-shift and post-shift

Lemma 3. *If $x \in \mathbb{B}^n$ is bitonic, then*

- *either $B(x)_L$ is bitonic and $B(x)_H = 1^{n/2}$*
- *or $B(x)_L = 0^{n/2}$ and $B(x)_H$ is bitonic*

proof of lemma 3:

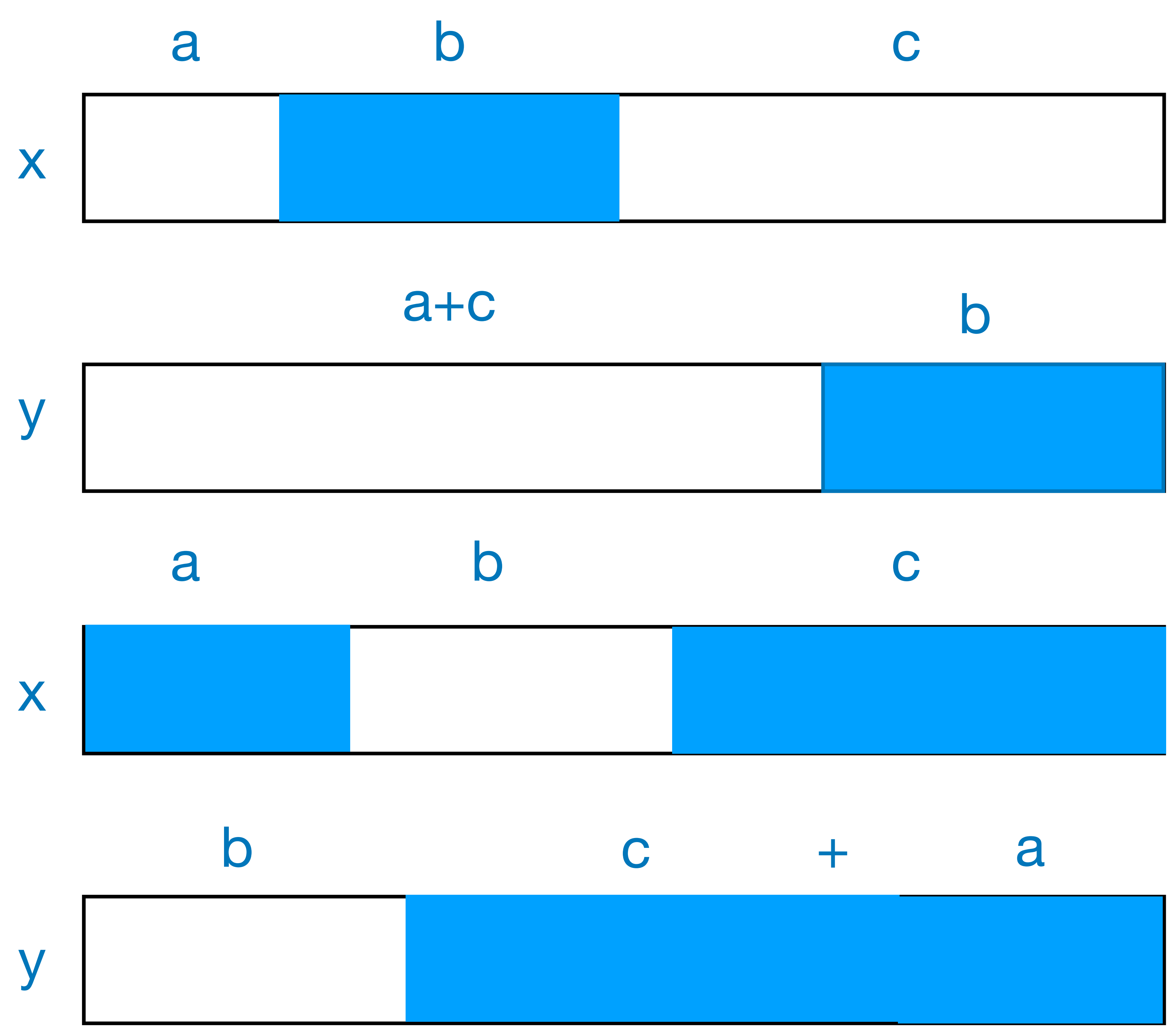
Shift input x such that all ones are ate the right border

- if x is upward bitonic shift cyclically right by c

$$y = src(x, c)$$

- if x is downward bitonic shift cyclically left by a which is the same as a cyclical right shift by $n - a$

$$y = src(x, n - a)$$



Lemma 3. *If $x \in \mathbb{B}^n$ is bitonic, then*

- *either $B(x)_L$ is bitonic and $B(x)_H = 1^{n/2}$*
- *or $B(x)_L = 0^{n/2}$ and $B(x)_H$ is bitonic*

proof of lemma 3:

Shift input x such that all ones are ate the right border

- if x is upward bitonic shift cyclically right by c

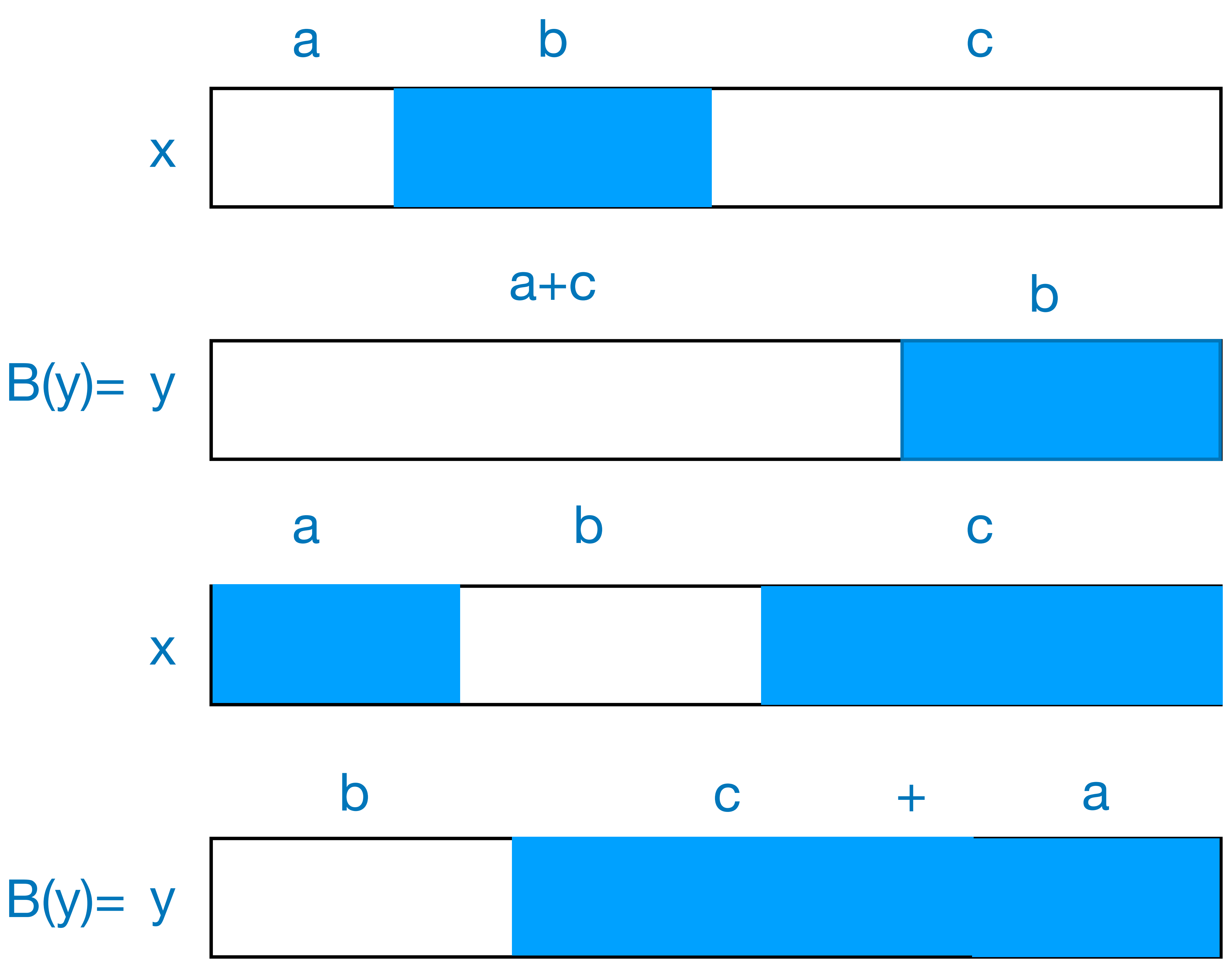
$$y = src(x, c)$$

- if x is downward bitonic shift cyclically left by a which is the same as a cyclical right shift by $n - a$

$$y = src(x, n - a)$$

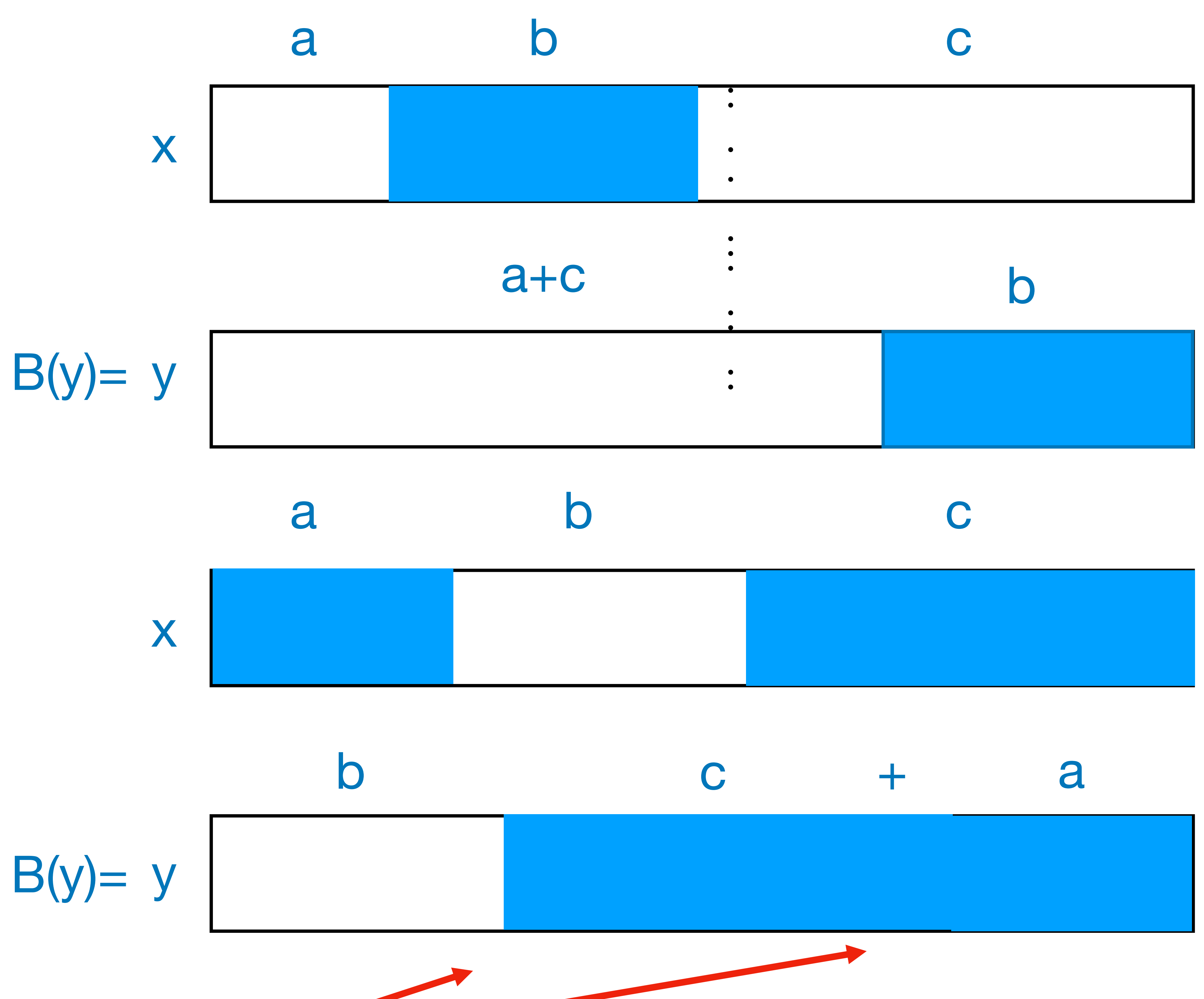
The shifted sequence y does not change when $B(\ )$ is applied

$$B(y) = y$$



Lemma 3. *If $x \in \mathbb{B}^n$ is bitonic, then*

- *either $B(x)_L$ is bitonic and $B(x)_H = 1^{n/2}$*
- *or $B(x)_L = 0^{n/2}$ and $B(x)_H$ is bitonic*



proof of lemma 3:

Shift input x such that all ones are ate the right border

- if x is upward bitonic shift cyclically right by c

$$y = src(x, c)$$

- if x is downward bitonic shift cyclically left by a which is the same as a cyclical right shift by $n - a$

$$y = src(x, n - a)$$

The shifted sequence y does not change when $B(\ )$ is applied

$$B(y) = y$$

- if the sequence has at least $n/2$ ones, then

$$B(y)_H = 1^{n/2}$$

and the lower half has for some d, e the form

$$B(y)_L = 0^d 1^e$$

Shifting $B(y)_H$ shanges nothing. Shifting $B(y)_L$ makes it bitonic

Lemma 3. *If $x \in \mathbb{B}^n$ is bitonic, then*

- *either $B(x)_L$ is bitonic and $B(x)_H = 1^{n/2}$*
- *or $B(x)_L = 0^{n/2}$ and $B(x)_H$ is bitonic*

proof of lemma 3:

Shift input x such that all ones are ate the right border

- if x is upward bitonic shift cyclically right by c

$$y = src(x, c)$$

- if x is downward bitonic shift cyclically left by a which is the same as a cyclical right shift by $n - a$

$$y = src(x, n - a)$$

The shifted sequence y does not change when $B(\ )$ is applied

$$B(y) = y$$

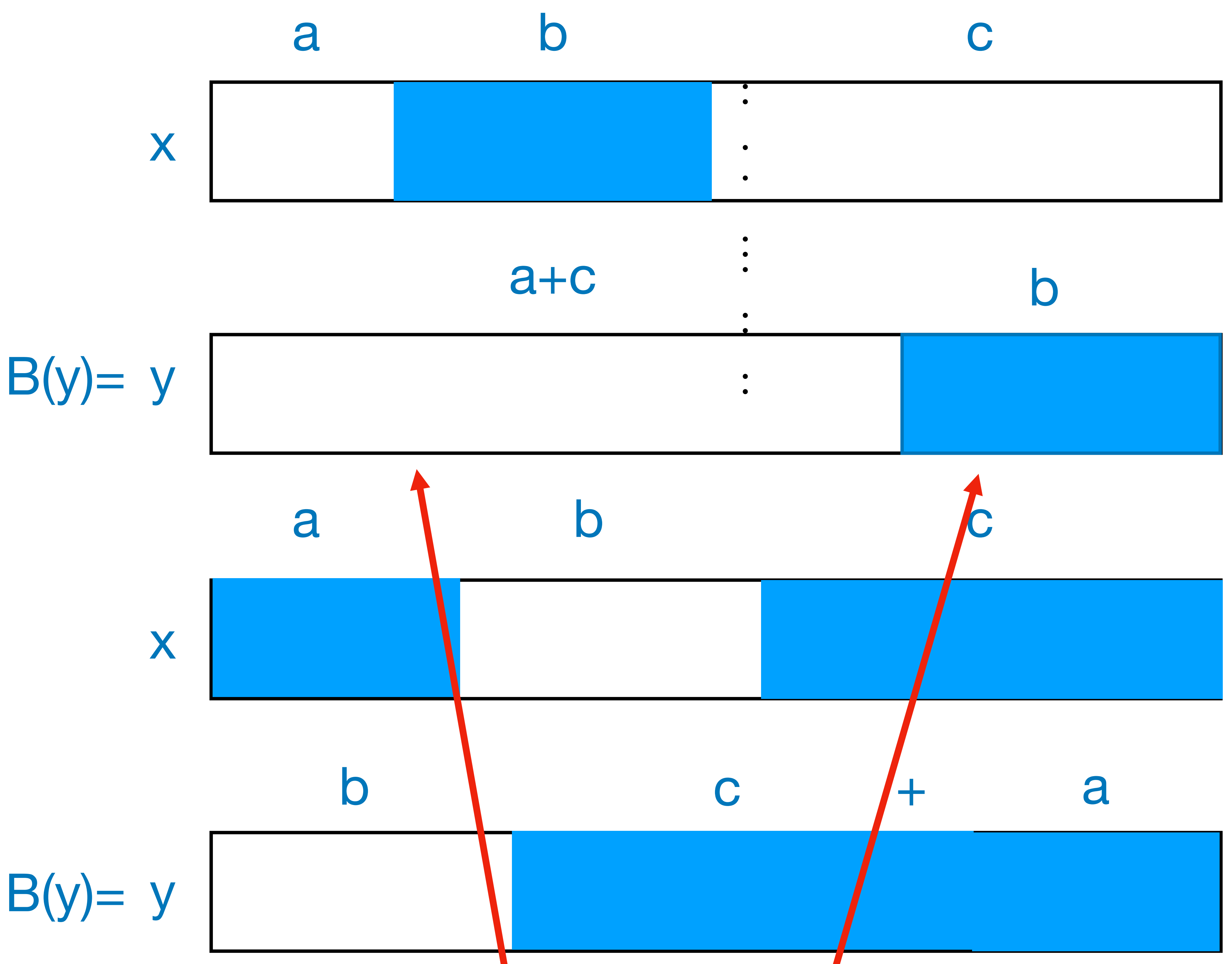
- if the sequence has at least $n/2$ ones, then

$$B(y)_H = 1^{n/2}$$

and the lower half has for some d, e the form

$$B(y)_L = 0^d 1^e$$

Shifting $B(y)_H$ shanges nothing. Shifting $B(y)_L$ makes it bitonic



- if the sequence has at least $n/2$ zeros, then

$$B(y)_L = 0^{n/2}$$

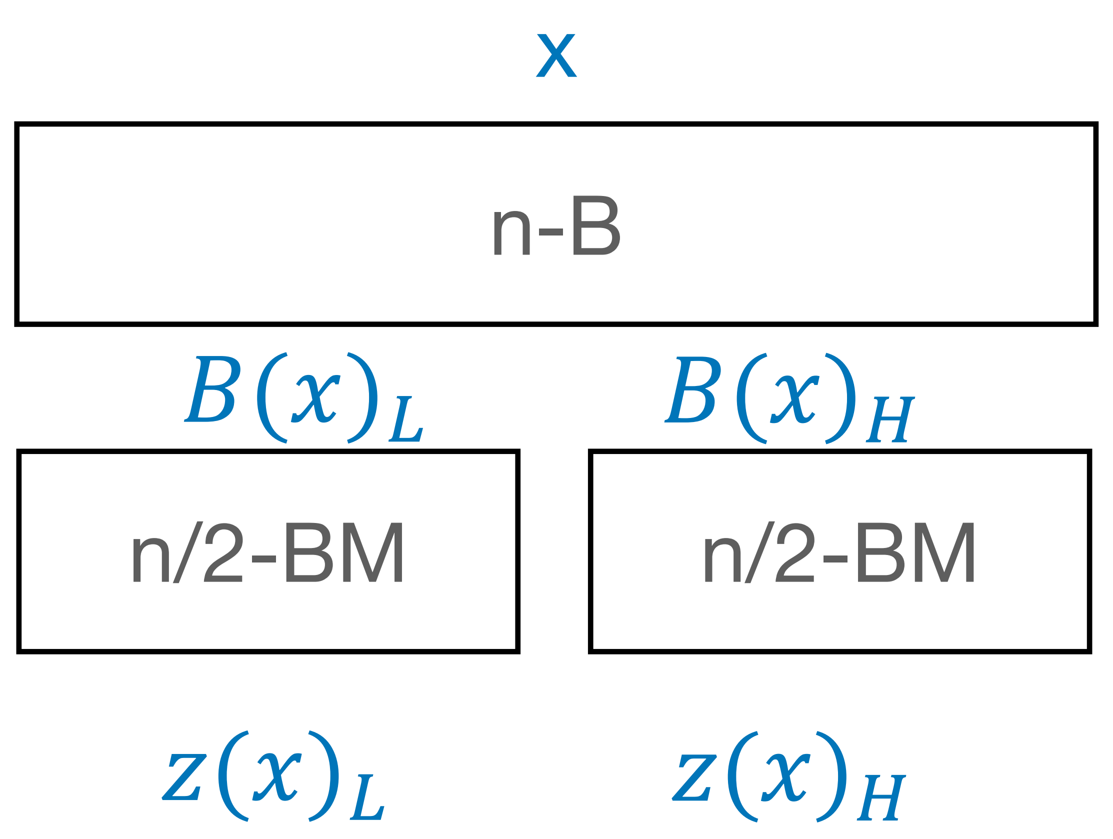
and the ~~lower~~ upper half has for some d, e the form

$$B(y)_H = 0^d 1^e$$

Shifting $B(y)_L$ shanges nothing. Shifting $B(y)_H$ makes it bitonic

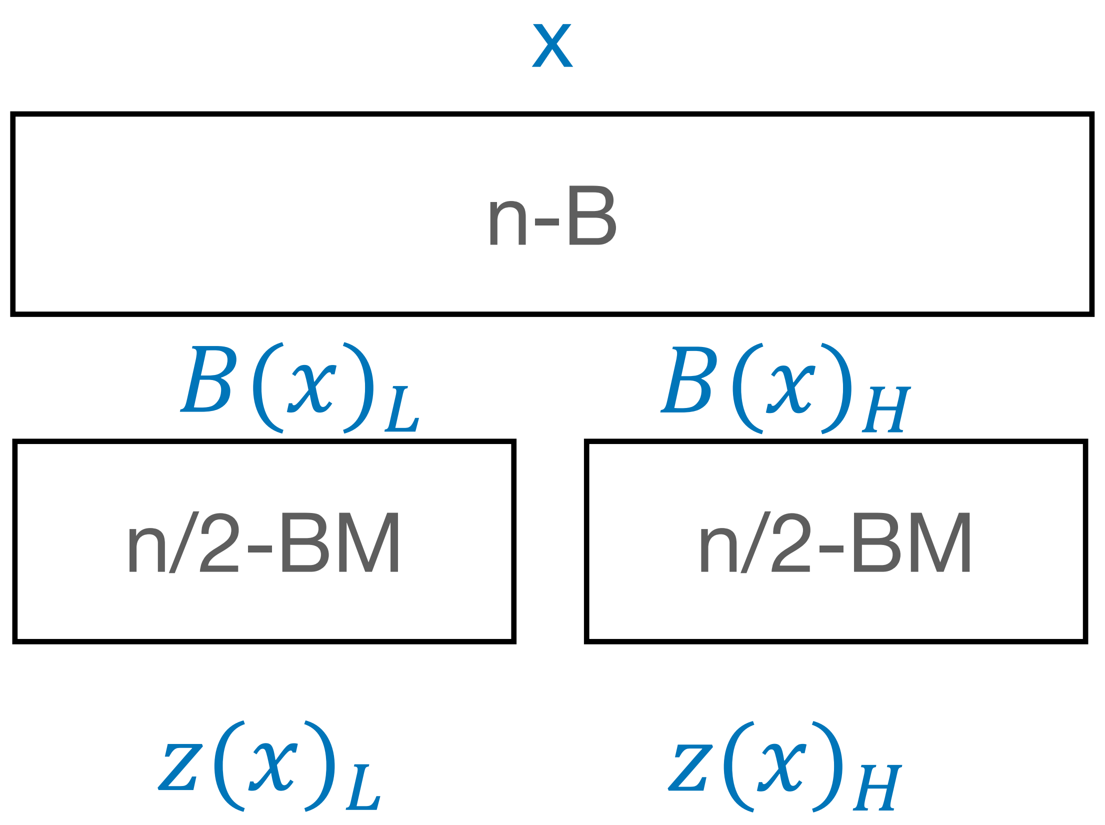
bitonic merge networks *n-BM*

- 2-BM: comparator
- n-BM:



bitonic merge networks *n-BM*

- 2-BM: comparator
- n-BM:

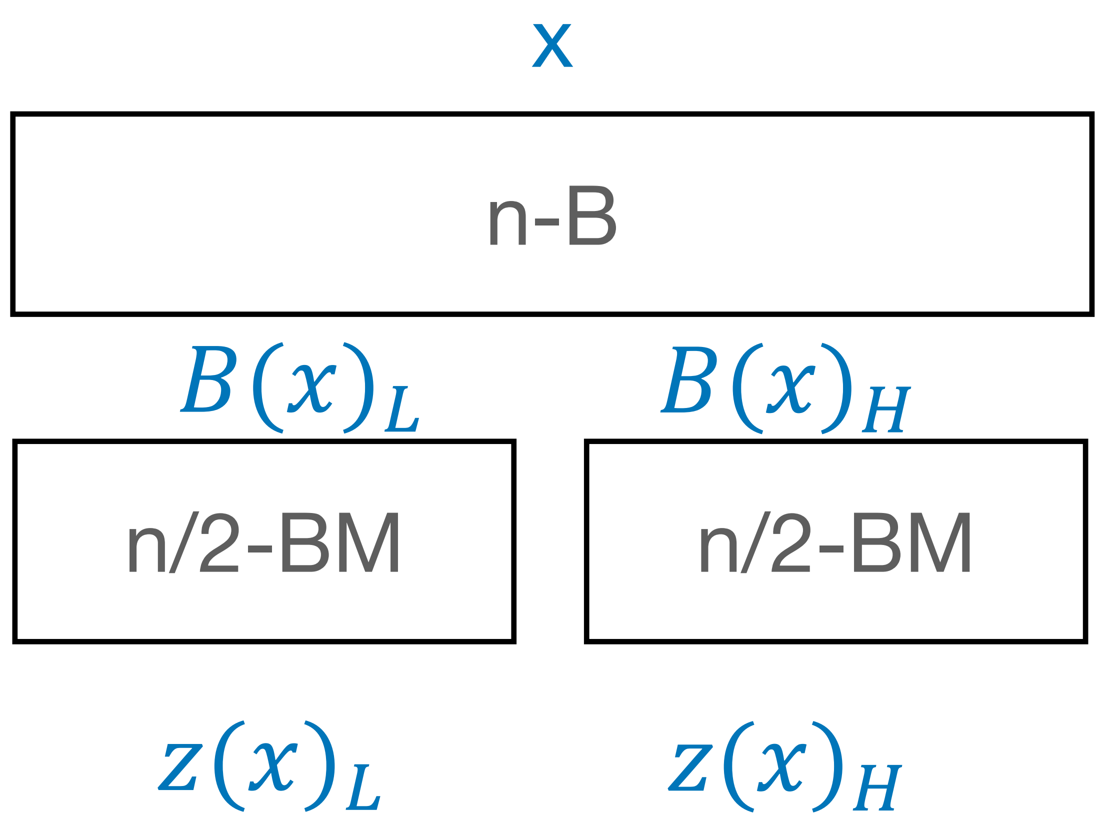


Lemma 8. *If the input $x \in \mathbb{B}^n$ of a bitonic merge network n-BM is bitonic, then its the ouput $z(x) \in \mathbb{B}^n$ is sorted.*

Proof by induction on n. Trivial for $n = 2$. Induction step $n/2 \to n$:
lemma 3 $\to$

bitonic merge networks *n-BM*

- 2-BM: comparator
- n-BM:



Lemma 8. *If the input $x \in \mathbb{B}^n$ of a bitonic merge network n-BM is bitonic, then its the ouput $z(x) \in \mathbb{B}^n$ is sorted.*

Proof by induction on n. Trivial for $n = 2$. Induction step $n/2 \to n$:
lemma 3 $\to$

Proof by induction on n. Trivial for $n = 2$. Induction step $n/2 \to n$:
lemma 3 $\to$

cases:

- $B(x)_L = 0^{n/2}$ and $B(x)_H$ bitonic. Then

$$z(x)_L = 0^{n/2} \text{ and } z(x)_H \text{ sorted by Ind. Hyp.}$$

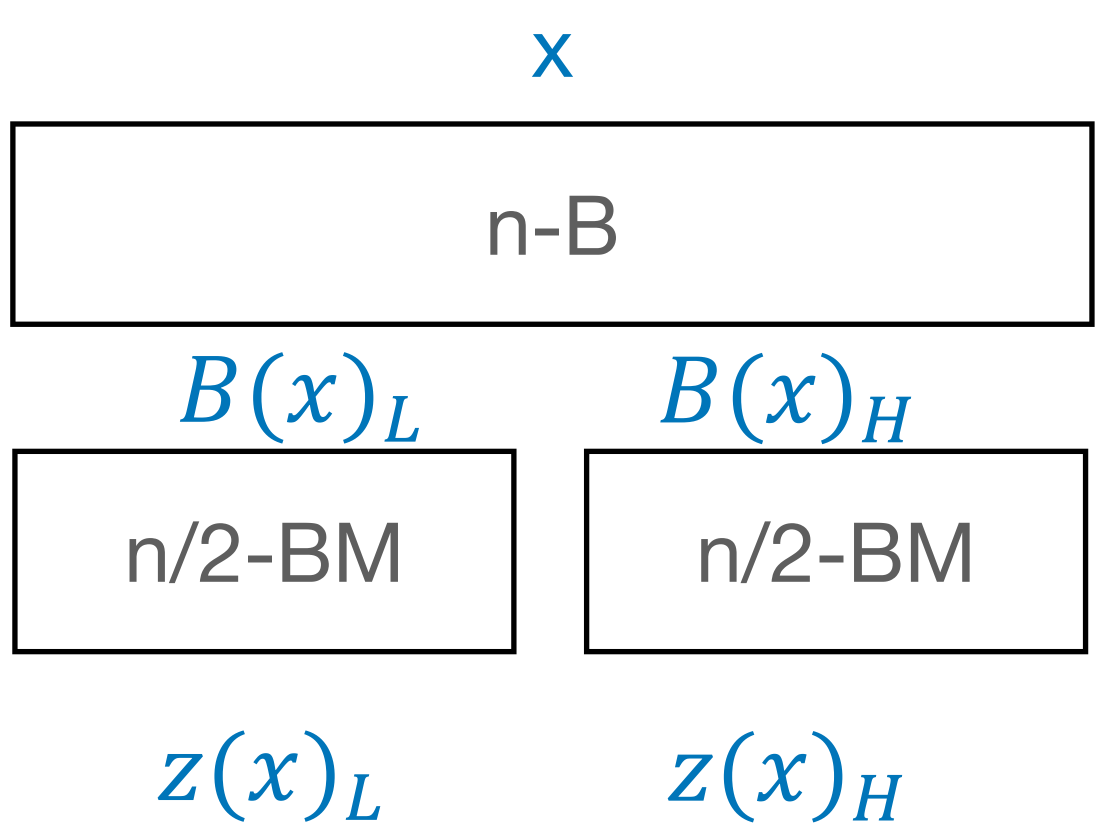
- $B(x)_H = 1^{n/2}$ and $B(x)_L$ bitonic. Then

$$z(x)_H = 1^{n/2} \text{ and } z(x)_L \text{ sorted by Ind. Hyp.}$$

In both cases $z(x)$ is sorted. lemma 3

bitonic sorter *n-BS*

- 2-BM: comparator
- n-BM:

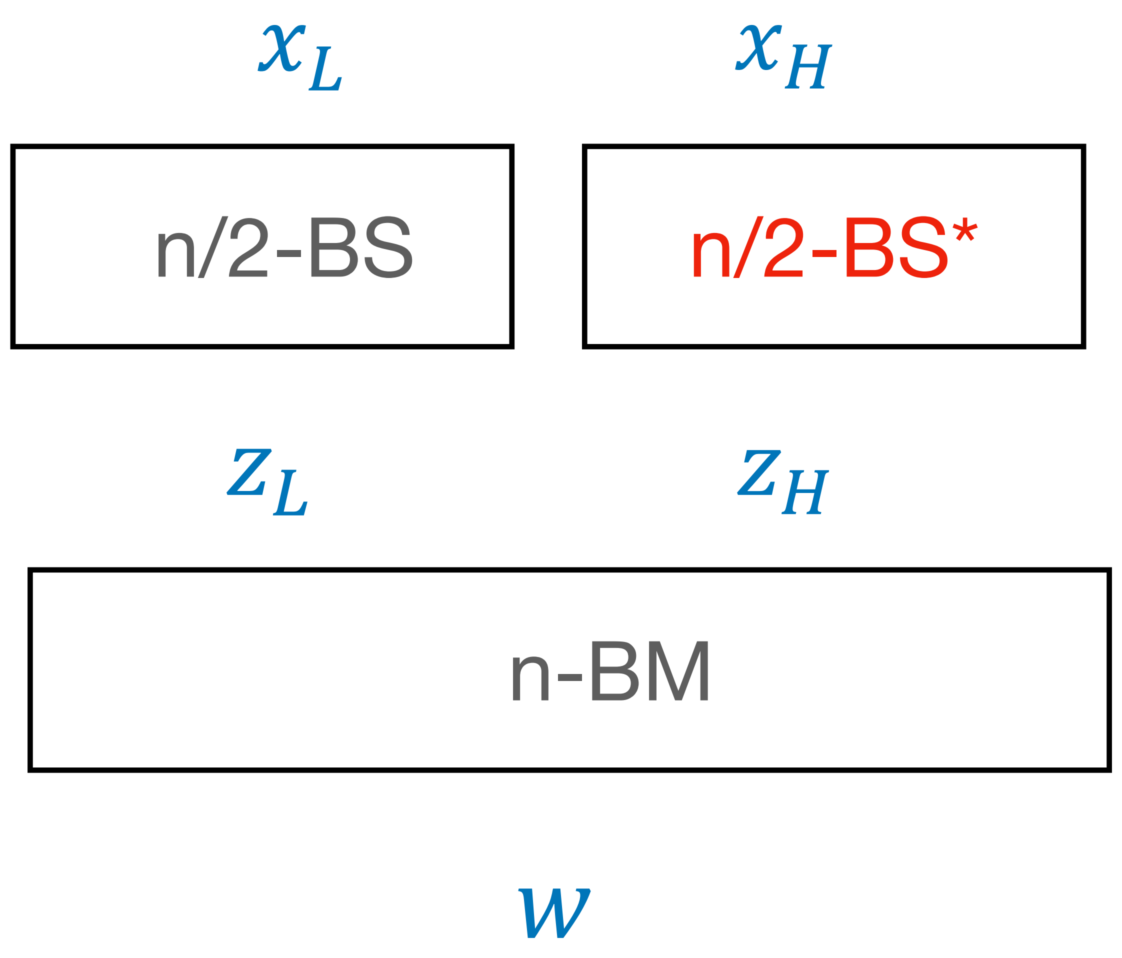


Lemma 8. *If the input $x \in \mathbb{B}^n$ of a bitonic merge network n-BM is bitonic, then its the ouput $z(x) \in \mathbb{B}^n$ is sorted.*

Proof by induction on n. Trivial for $n = 2$. Induction step $n/2 \to n$:
lemma 3 $\to$

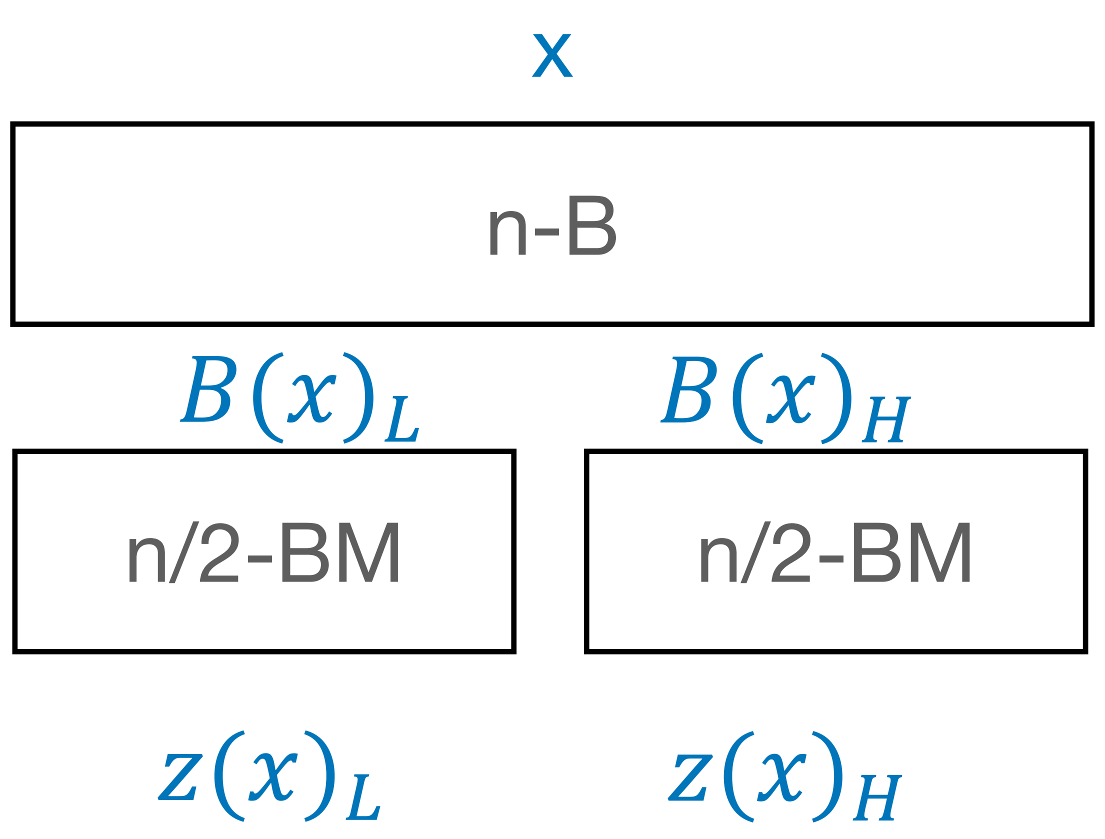
- n-BS*: reverse order of outputs in comparators

- 2-BS: comparator
- n-BS:



bitonic sorter *n-BS*

- 2-BM: comparator
- n-BM:

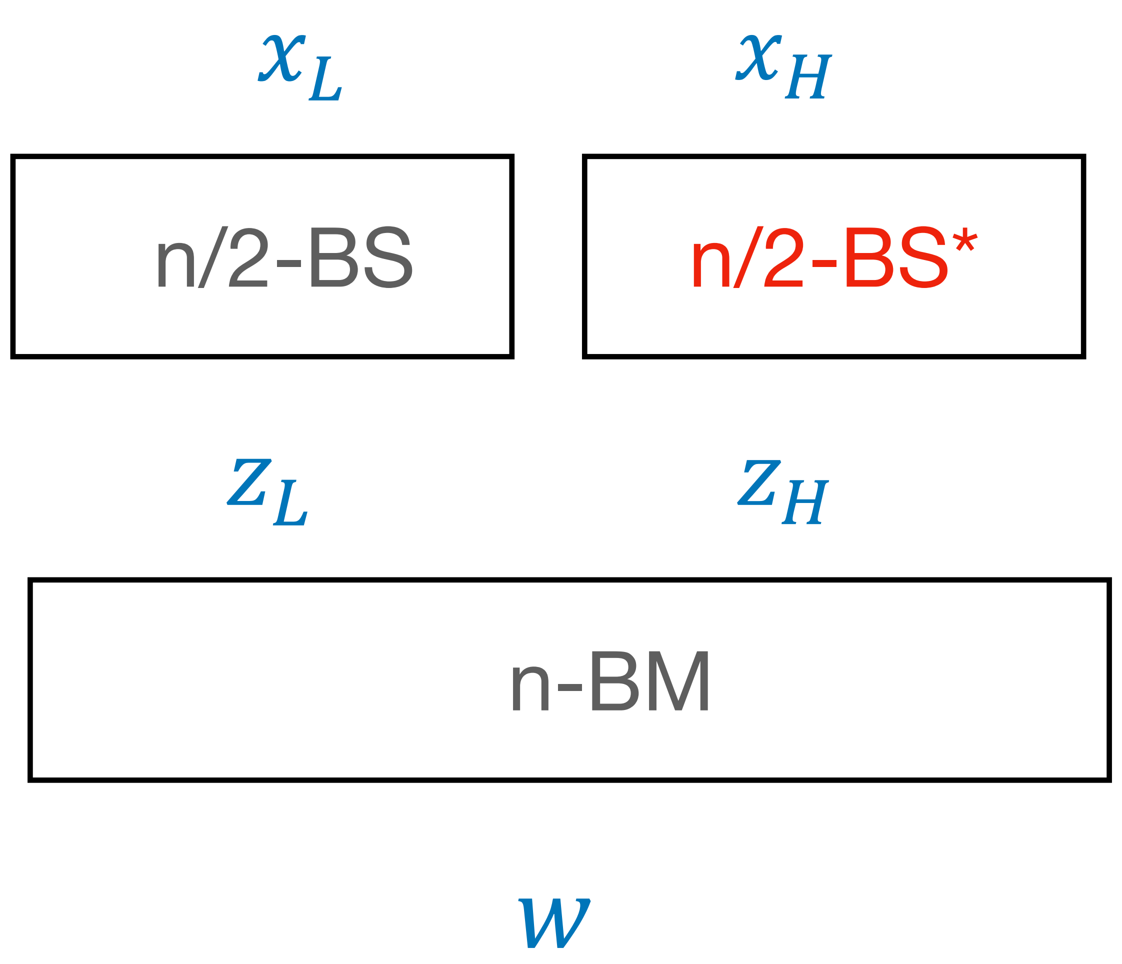


Lemma 8. *If the input $x \in \mathbb{B}^n$ of a bitonic merge network n-BM is bitonic, then its the ouput $z(x) \in \mathbb{B}^n$ is sorted.*

Proof by induction on n. Trivial for $n = 2$. Induction step $n/2 \to n$:
lemma 3 $\to$

- n-BS*: reverse order of outputs in comparators

- 2-BS: comparator
- n-BS:

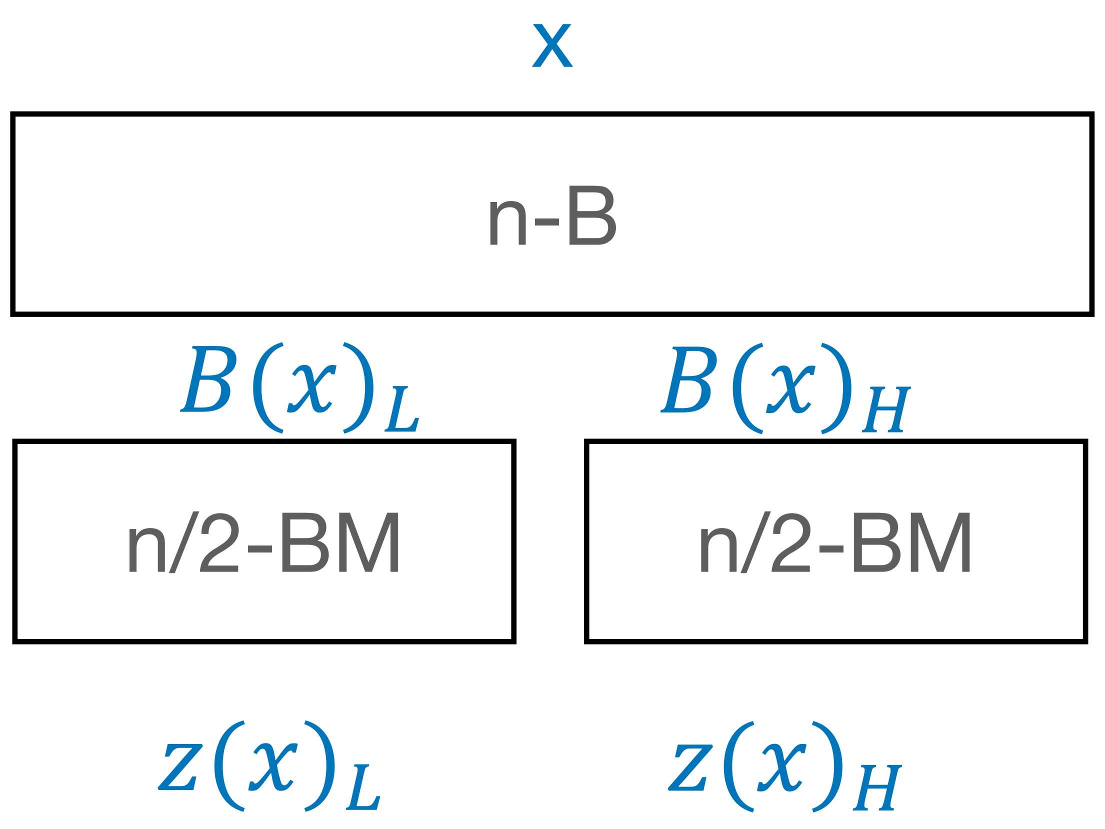


Lemma 8. *With input $x \in \mathbb{B}^n$ a bitonic sorter n-BS produces an output $w \in \mathbb{B}^n$, which is sorted.*

bitonic sorter *n-BS*

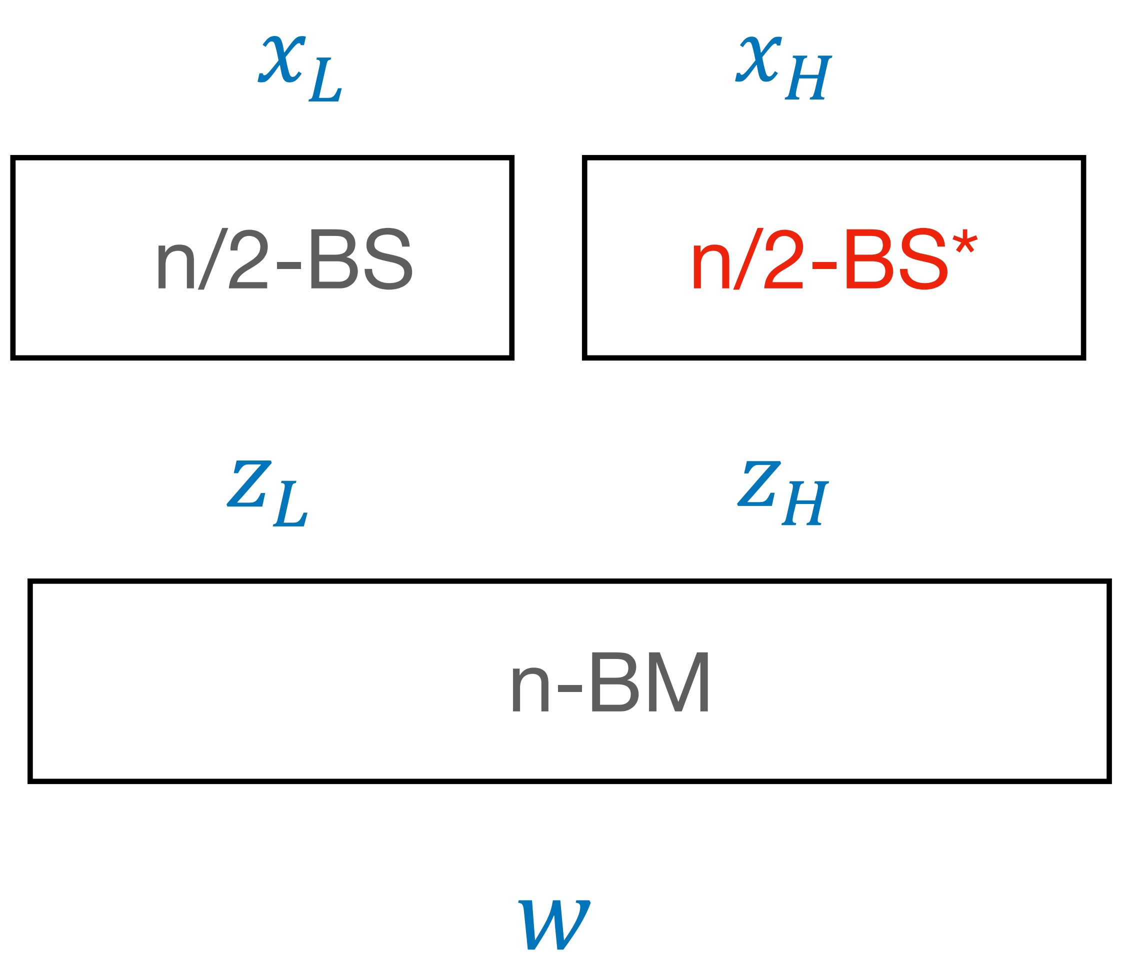
- n-BS*: reverse order of outputs in comparators

- 2-BM: comparator
- n-BM:



Lemma 8. *If the input $x \in \mathbb{B}^n$ of a bitonic merge network n-BM is bitonic, then its the ouput $z(x) \in \mathbb{B}^n$ is sorted.*

Proof by induction on n. Trivial for $n = 2$. Induction step $n/2 \to n$:
lemma 3 $\to$



Lemma 8. *With input $x \in \mathbb{B}^n$ a bitonic sorter n-BS produces an output $w \in \mathbb{B}^n$, which is sorted.*

Proof by induction on n. Trivial for $n = 2$. Induction step $n/2 \to n$:

- Induction hypothesis $\to$:

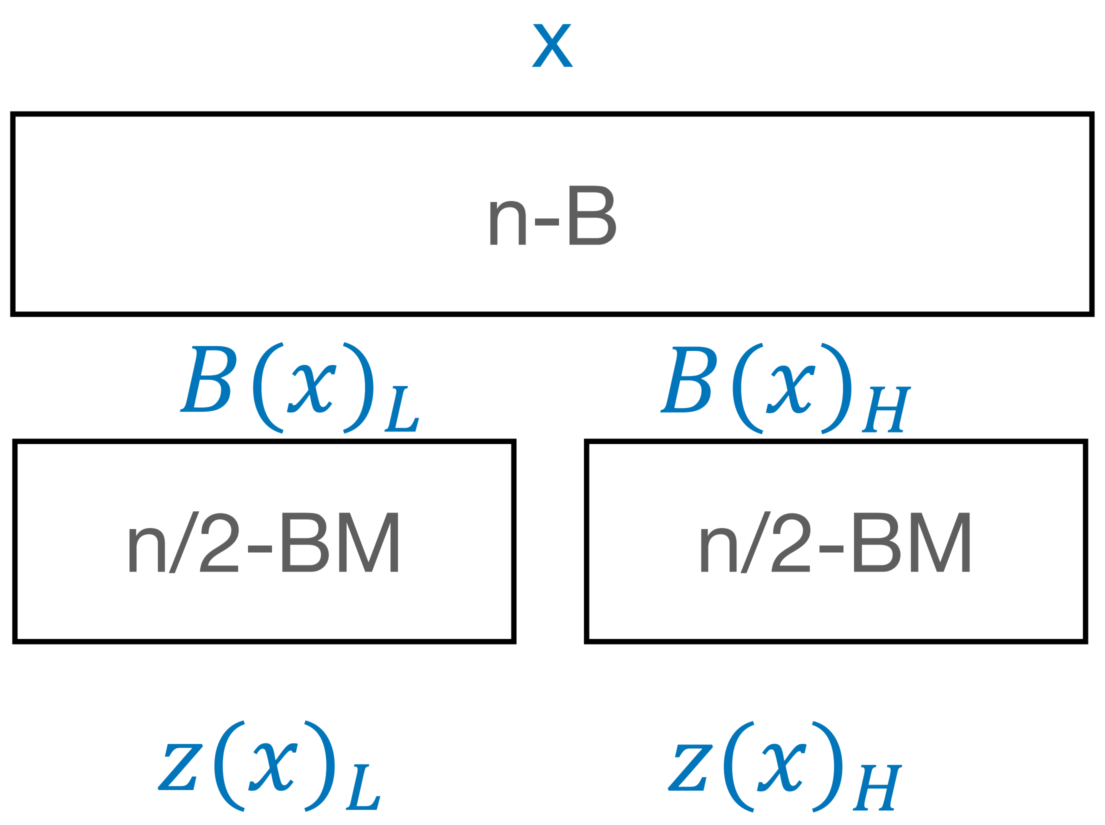
  z_l sorted in increasing order and z_H sorted in decreasing order.

- $z_L \circ z_H$ is bitonic.
- lemma 7 $\to$:

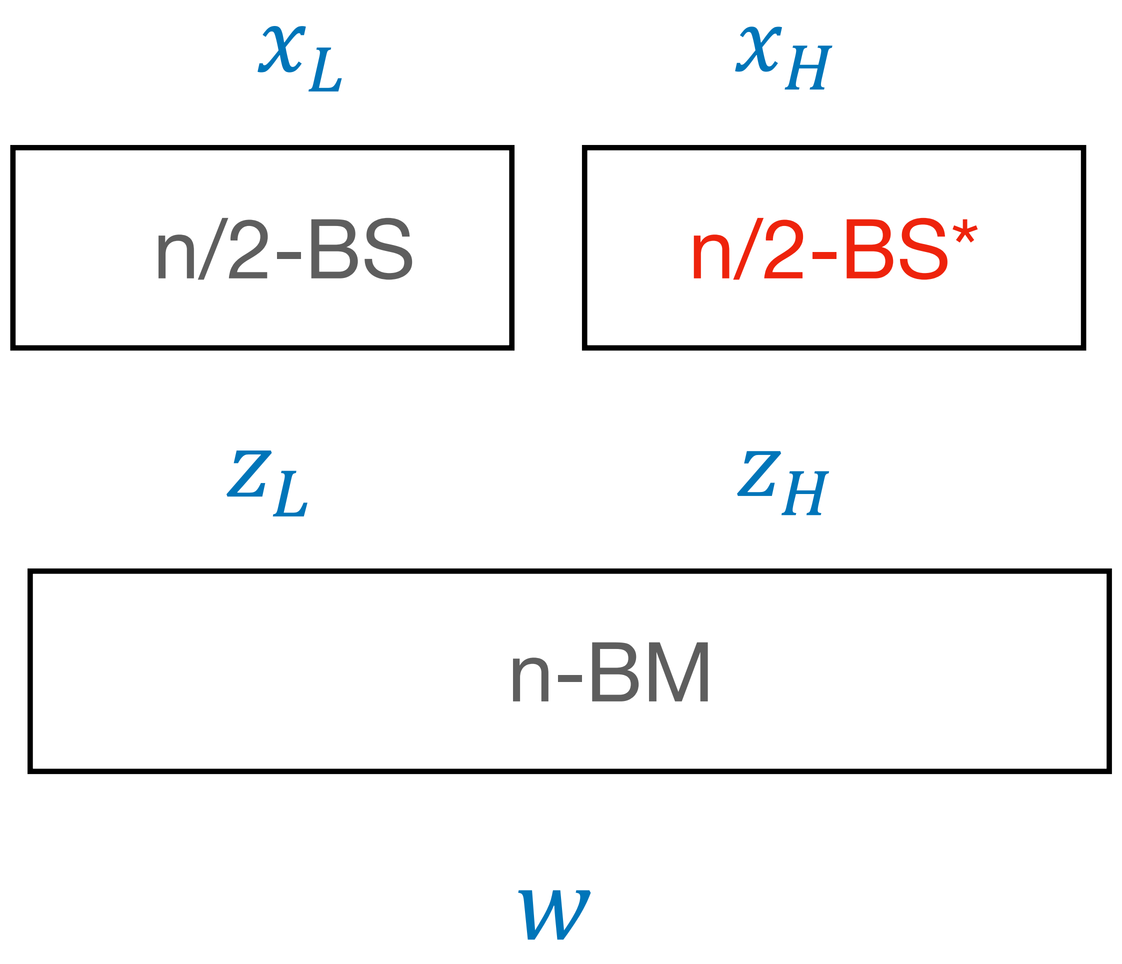
  output w is sorted.

cost and delay

- 2-BM: comparator
- n-BM:



- n-BS*: reverse order of outputs in comparators



- bitonic merger

delay $O(\log n)$

$$\begin{cases} d_M(2) = 1 \\ d_M(n) = 1 + d_M(n/2) \end{cases}$$

cost $O(n \log n)$

$$\begin{cases} c_M(2) = 1 \\ c_M(n) = 2c_M(n/2) + n/2 \end{cases}$$

„Exercise 1!"

- bitonic sorter

„Exer. 1" $O(\log^2(n)) \longrightarrow$

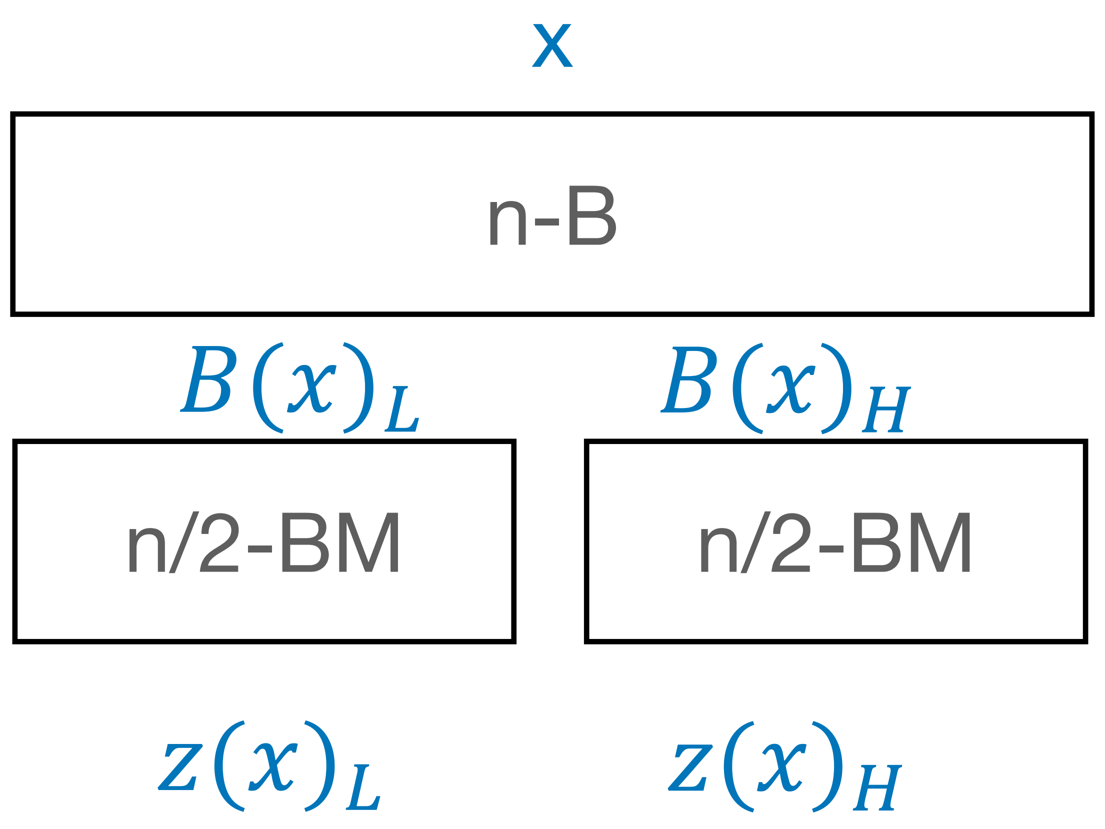
$$\begin{cases} d_S(2) = 1 \\ d_S(n) = d_S(n/2) + d_M(n) \end{cases}$$

$O(n \log^2 n) \longrightarrow$

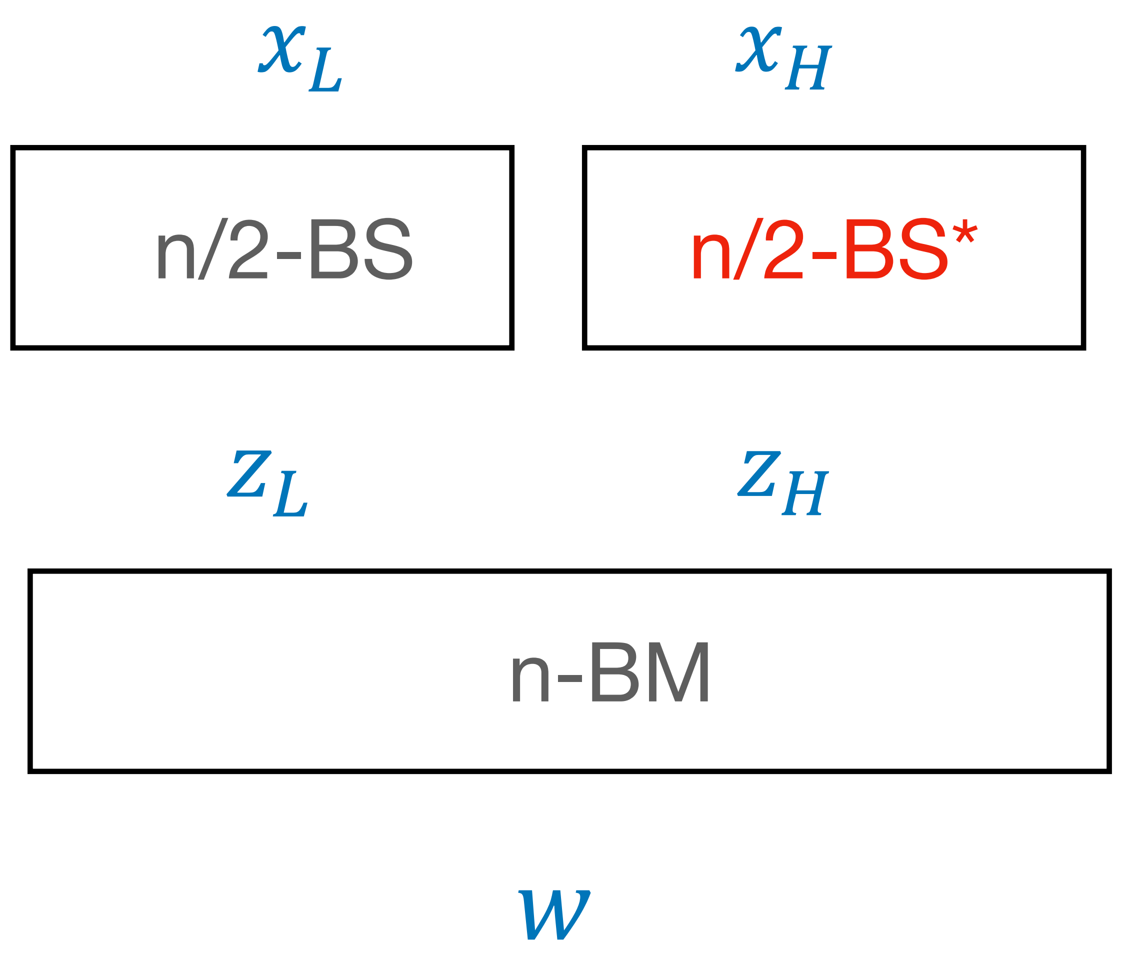
$$\begin{cases} c_S(2) = 1 \\ c_S(n) = 2c_S(n/2) + c_M(n) \end{cases}$$

cost and delay

- 2-BM: comparator
- n-BM:



- n-BS*: reverse order of outputs in comparators



- bitonic merger

$$
\begin{aligned}
d_M(2) &= 1\\
d_M(n) &= 1 + d_M(n/2)\\
c_M(2) &= 1\\
c_M(n) &= 2c_M(n/2) + n/2
\end{aligned}
$$

- bitonic sorter

$$
\begin{aligned}
d_S(2) &= 1\\
d_S(n) &= d_S(n/2) + d_M(n)\\
c_S(2) &= 1\\
c_S(n) &= 2c_S(n/2) + c_M(n)
\end{aligned}
$$

solution: exercise sheet 1