LOWER BOUNDS ON CIRCUITS

WRACH 2025

Thursday 24th April, 2025

Ahmed Alharbi, Charles Bouillaguet

LIP6 - Sorbonne Université

Motivation

SCIENCES SORBONNE UNIVERSITÉ

IACR Transactions on Cryptographic Hardware and Embedded Systems
ISSN XXXX-XXXX, Vol. 0, No. 0, pp. 1–40. DOI:XXXXXXXX

Compact Circuits for Efficient Möbius Transform

Subhadeep Banik[1], and Francesco Regazzoni[2,1]

[1] Universita della Svizzera Italiana, Lugano, Switzerland, subhadeep.banik@usi.ch
[2] University of Amsterdam, Amsterdam, Netherlands, f.regazzoni@uva.nl

Abstract. The Möbius transform is a linear circuit used to compute the evaluations of a Boolean function over all points on its input domain. The operation is very useful in finding the solution of a system of polynomial equations over $GF(2)$ for obvious reasons. However the operation, although linear, needs exponential number of logic operations (around $n \cdot 2^{n-1}$ bit xors) for an n-variable Boolean function. As such, the

Motivation

SCIENCES SORBONNE UNIVERSITÉ

dimensions then the Area(A) and Time(T) product of the FFT is subject to a lower bound $AT^2 \geq N^2$ (where $N = 2^n$) [Tho79]. Because the circuit has to hold the N input data, this implies that $AT \geq N^{3/2}$. This lower-bound is based on communication complexity, i.e. the fact that "wires take space". In this paper we begin with two circuits for Möbius Transform i.e. **Expmob1**/**Expmob2** that take $AT^2 = N \cdot \log(N)$ and $N \cdot (\log(N))^2$ respectively.

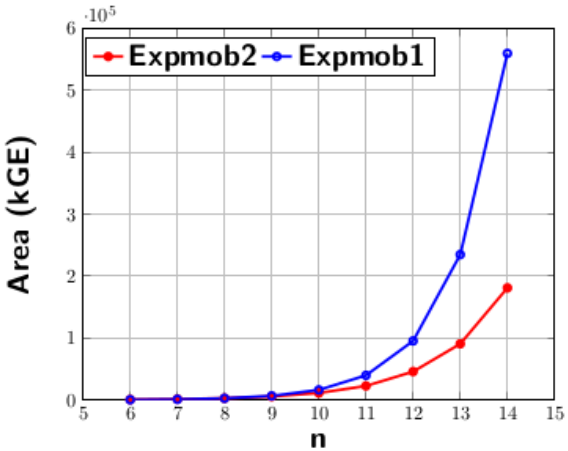
Motivation

SCIENCES SORBONNE UNIVERSITÉ

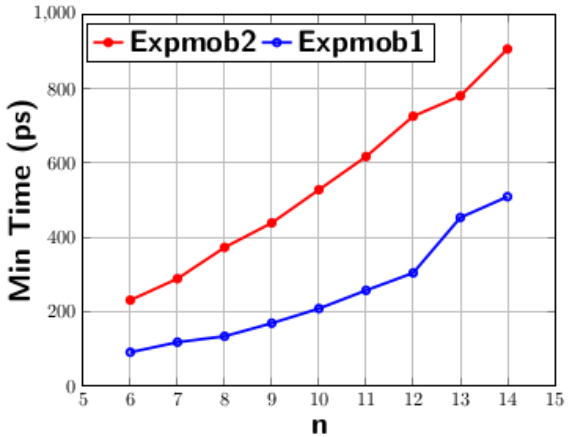
Expmob1 This architecture implements the circuit in Figure 2 as a single unrolled circuit, i.e. it implements all the n butterfly stages as dedicated circuits sequentially. Consider $\textbf{one}_i(x) : \{0,1\}^{n-1} \to \{0,1\}^n$ to be the function that inserts a 1 in the i-th MSB position of x, and $\textbf{zero}_i(x)$ to be a function that inserts a 0 in the same position, i.e $\textbf{one}_0(1001) = 1\ 1101$ and $\textbf{zero}_0(1001) = 0\ 1001$ etc. Note that there are a total of 2^{n-1} butterfly operations in each of the n stages. In the i-th stage (for $0 \leq i \leq n-1$), the j-th butterfly takes as input the bits in the position $\textbf{zero}_i(j)$ and $\textbf{one}_i(j)$ for all $0 \leq j \leq 2^{n-1} - 1$. This requires a total of $n \cdot 2^{n-1}$ number of 2-input xor gates in total. However such a circuit is able to compute the transformation in a single cycle.

Motivation

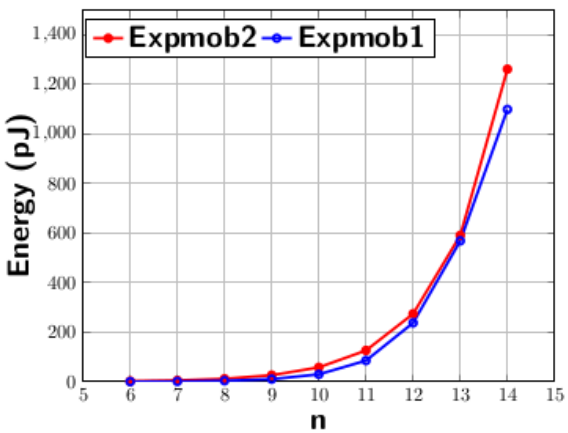
Compact Circuits for Efficient Möbius Transform

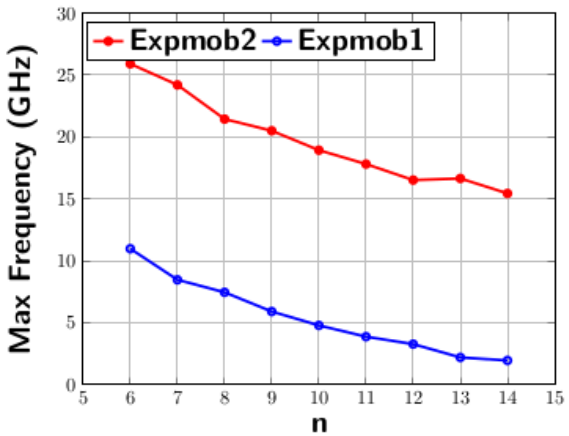


(a) Area



(b) Time



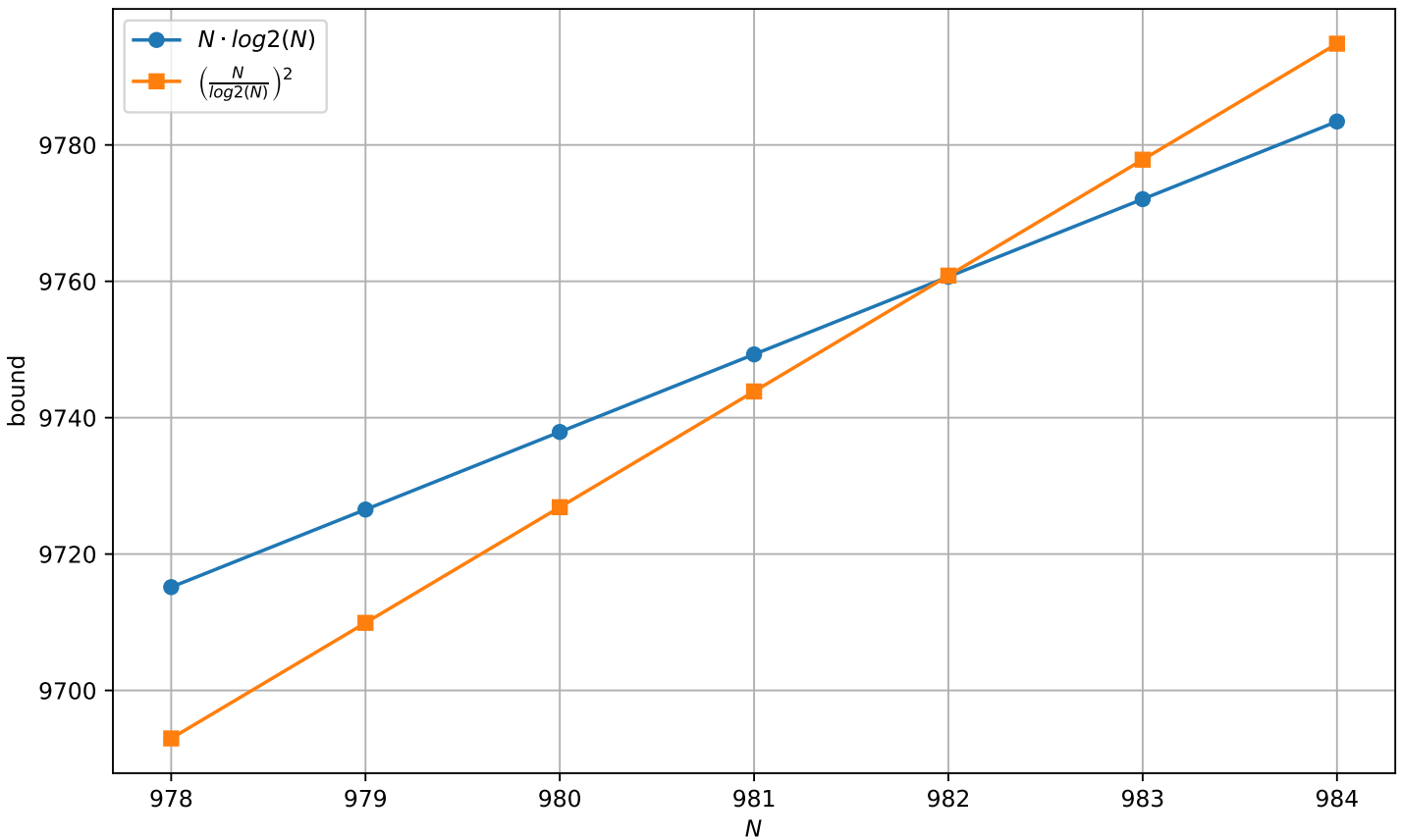


Bound from Comunication Complexity

SCIENCES SORBONNE UNIVERSITÉ

Communication complexity helped us to improve the AT^2 bound from

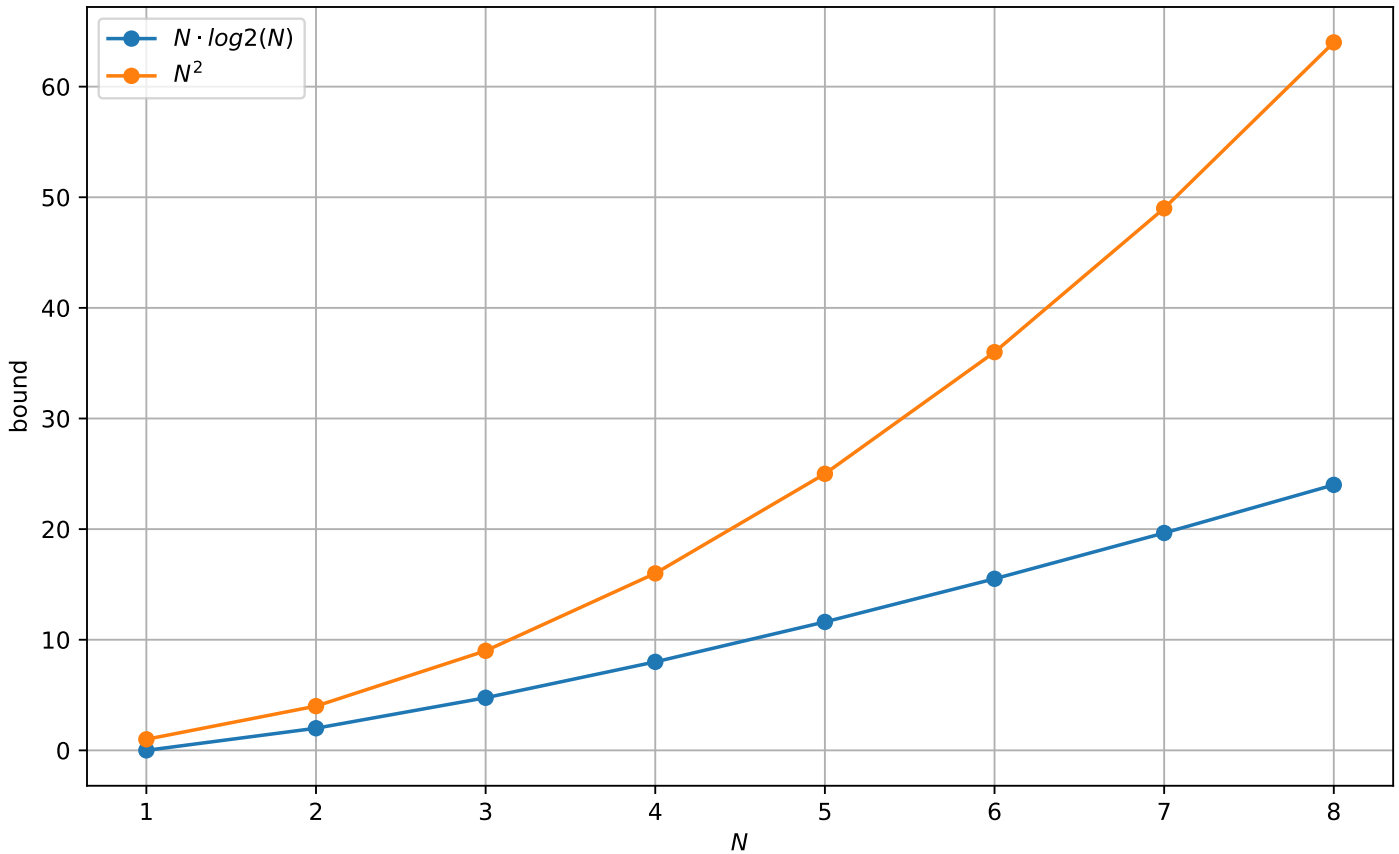
$$\Omega\left(N \cdot log_2(N)\right) \rightarrow \Omega\left(\frac{N^2}{log_2(N)^2}\right)$$



Bound from Communication Complexity

In most common practical cases, AT^2 improved from

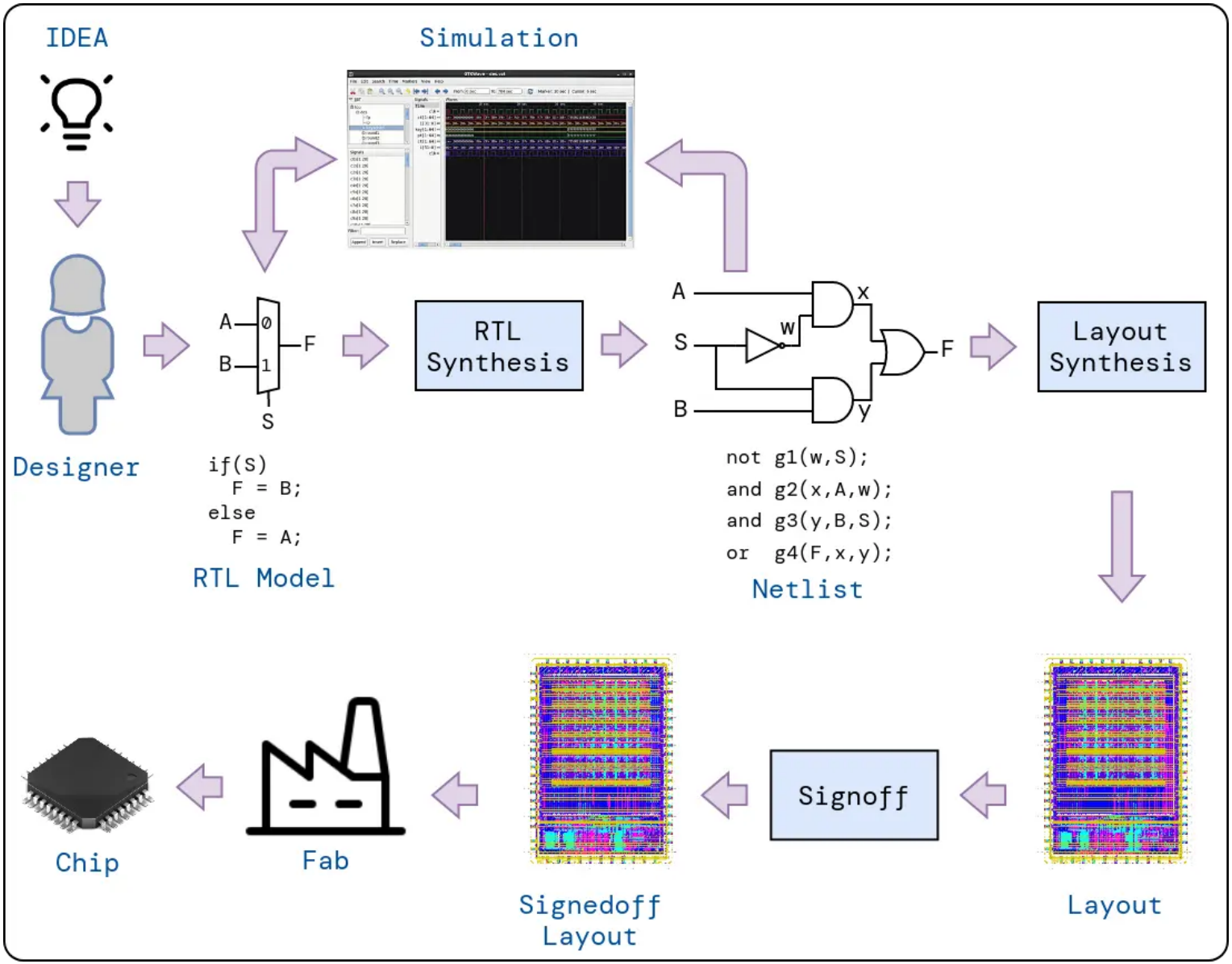
$$\Omega\left(N \cdot log_2(N)\right) \rightarrow \Omega\left(N^2\right)$$



Outline

Chip Design

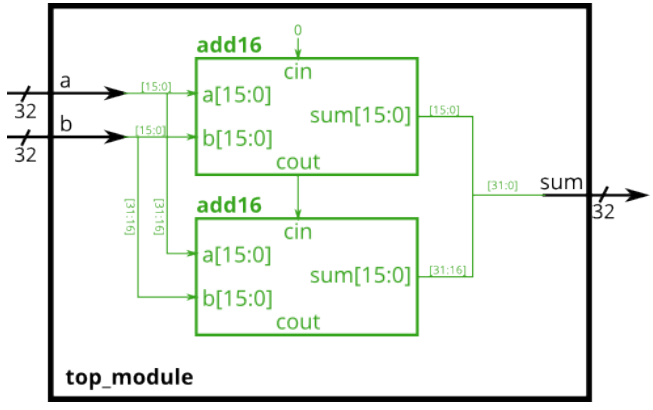
Overview



https://openlane2.readthedocs.io/en/latest/getting_started/newcomers/index.html

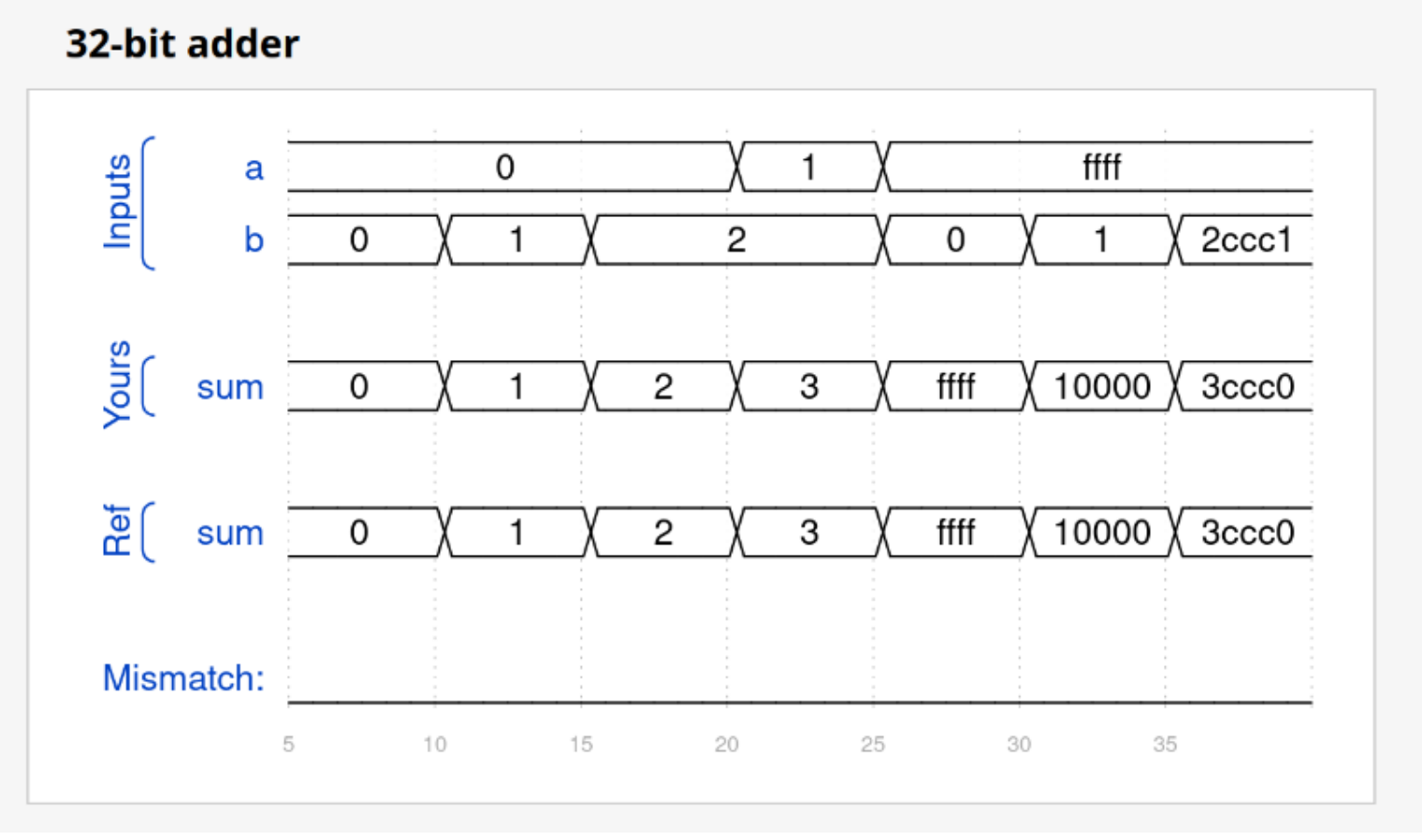
RTL Model: Verilog

```
1  module add32(
2      input [31:0] a,
3      input [31:0] b,
4      output [31:0] sum
5  );
6      wire carry;
7      wire [15:0]  sum_lo, sum_hi;
8
9      add16 a0 (a[15:0], b[15:0],  1'b0, sum_lo, carry);
10     add16 a1 (a[31:16], b[31:16], carry, sum_hi, 1'b0);
11
12     assign sum = {sum_hi, sum_lo};
13
14 endmodule
15
```



hdlbits.01xz.net

RTL Model: Test Bench (Simulation)



https://hdlbits.01xz.net/ using INTEL's Quartus

Process Design Kit (PDK)

SCIENCES SORBONNE UNIVERSITÉ

Defines Available gates, design rules, number of layers, etc.

Process Design Kit (PDK)

SCIENCES SORBONNE UNIVERSITÉ

Defines Available gates, design rules, number of layers, etc.

Feature	**Sky130**	**IHP SG13G2**
Technology Node	130nm	130nm
Designer	SkyWater Technology	Leibniz Institute for High Performance Microelectronics

Process Design Kit (PDK)

SCIENCES SORBONNE UNIVERSITÉ

Defines Available gates, design rules, number of layers, etc.

Feature	**Sky130**	**IHP SG13G2**
Technology Node	130nm	130nm
Designer	SkyWater Technology	Leibniz Institute for High Performance Microelectronics
Standard Cells	428	79

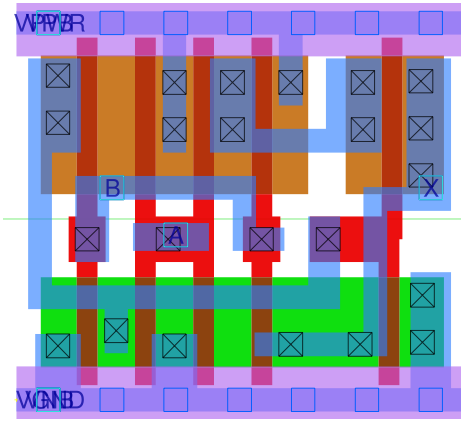
Process Design Kit (PDK)

SCIENCES SORBONNE UNIVERSITÉ

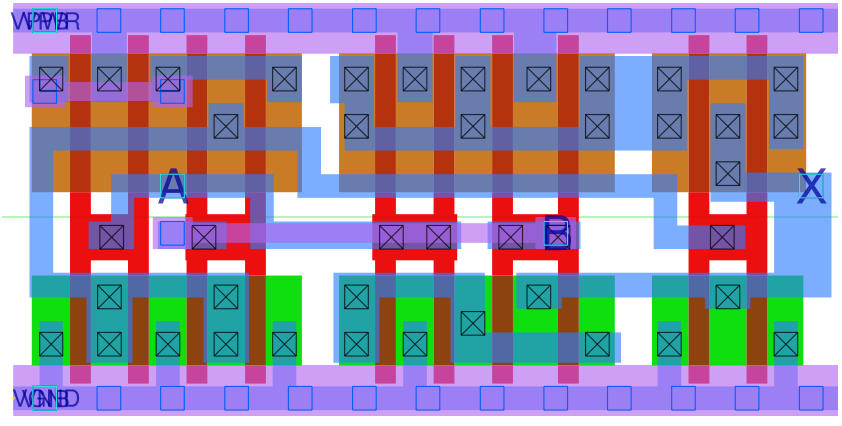
Defines Available gates, design rules, number of layers, etc.

Feature	**Sky130**	**IHP SG13G2**
Technology Node	130nm	130nm
Designer	SkyWater Technology	Leibniz Institute for High Performance Microelectronics
Standard Cells	428	79
Metal Layers	5	5–6

e.g. xor cell in `sky130_fd_sc_hd`



`xor2_1`



`xor2_2`

Same logic but with different drive strength.

RTL Synthesis (Netlist): Yosys

SCIENCES SORBONNE UNIVERSITÉ

RTL Synthesis (Netlist): Yosys

SCIENCES SORBONNE UNIVERSITÉ

```
        === expmob2 ===
        IHP PDK (Modified)
Number of cells:     599
   and21nor_x0         1
   and2_x1            35
   and3_x1            25
   and4_x1            18
   dff_x1             97
   inv_x0            114
   mux2_x1            1
   nand2_x0           49
   nand3_x0           6
   nand4_x0           5
   nexor2_x0          55
   ...
   nor3_x0            63
```

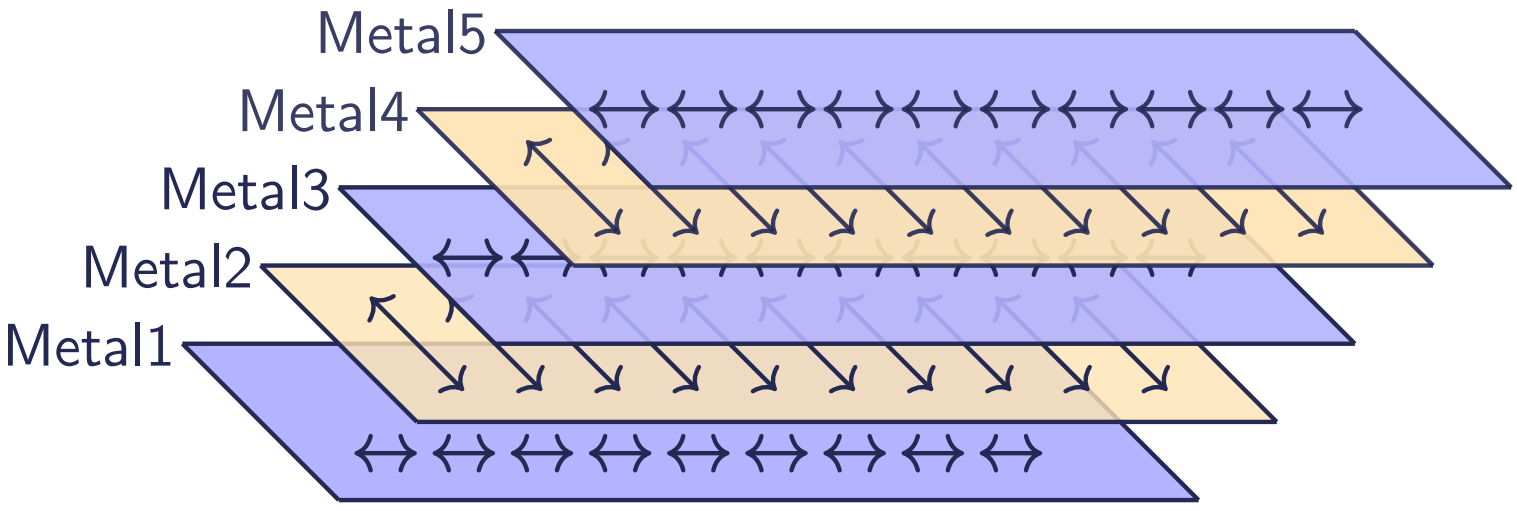
```
        === expmob2 ===
        Sky130 PDK
Number of cells:    388
  a221o_2           63
  a31o_2             3
  and2_2             8
  and2b_2           64
  and3_2            12
  and4_2            19
  buf_2             32
  or2_2              1
  or4_2              8
  or4bb_2            1
  ...
  xnor2_2            3
  xor2_2            40
```

RTL Synthesis (Netlist): Yosys

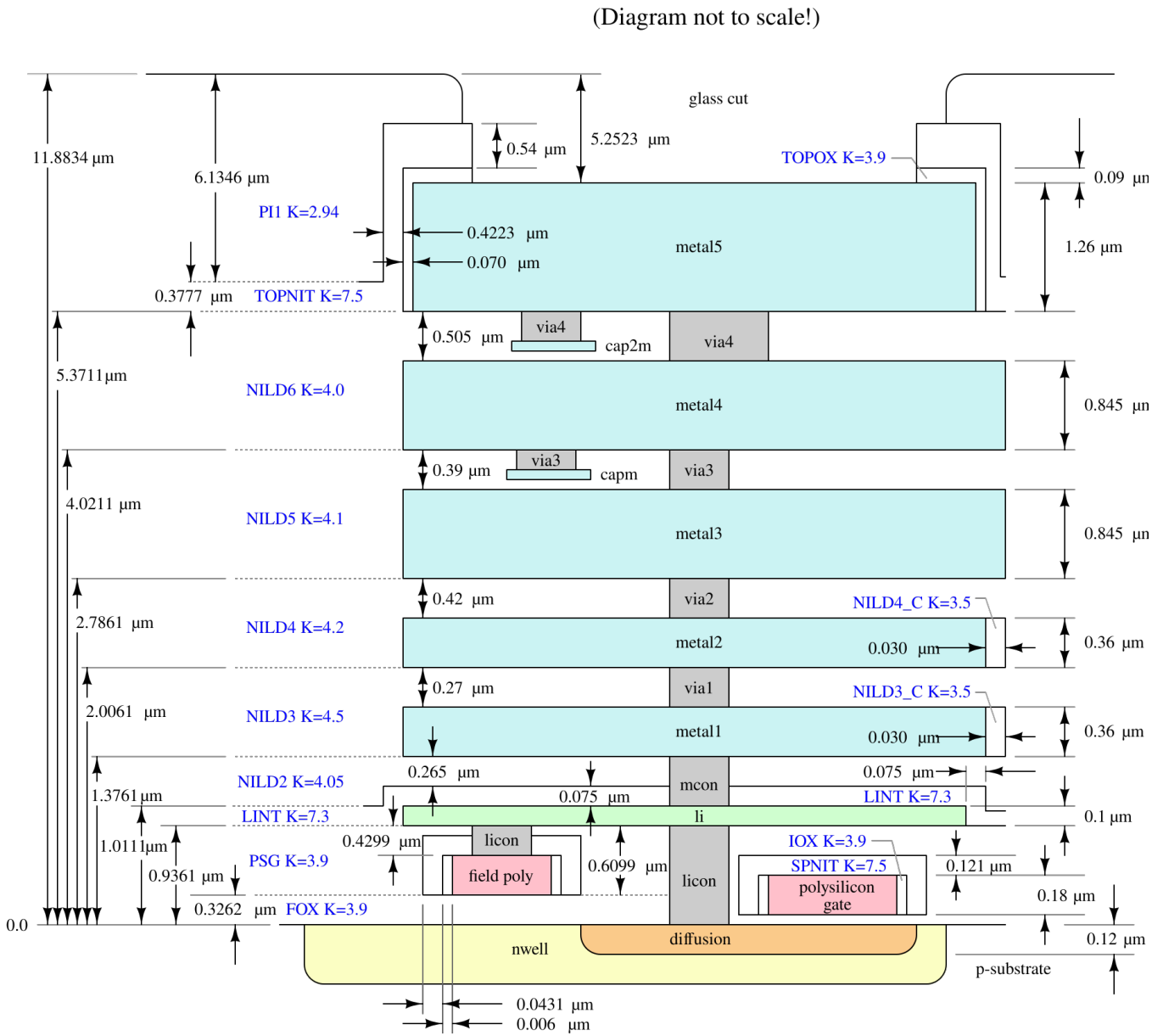
SCIENCES SORBONNE UNIVERSITÉ

```
1  /* Generated by Yosys 0.46 (git sha1 e97731b9dda91fa5fa53ed87df7c34163ba59a41,
2  module expmob1(inputs, outputs);
3    wire _00_;
4     ...
5    wire _17_;
6    input [0:15] inputs;
7    wire [0:15] inputs;
8    output [0:15] outputs;
9    wire [0:15] outputs;
10   sky130_fd_sc_hd__xnor2_2 _18_ (
11     .A(inputs[8]),
12     .B(inputs[0]),
13     .Y(_00_)
14   );
15   sky130_fd_sc_hd__inv_2 _19_ (
16     .A(_00_),
17     .Y(outputs[8])
18   );
```

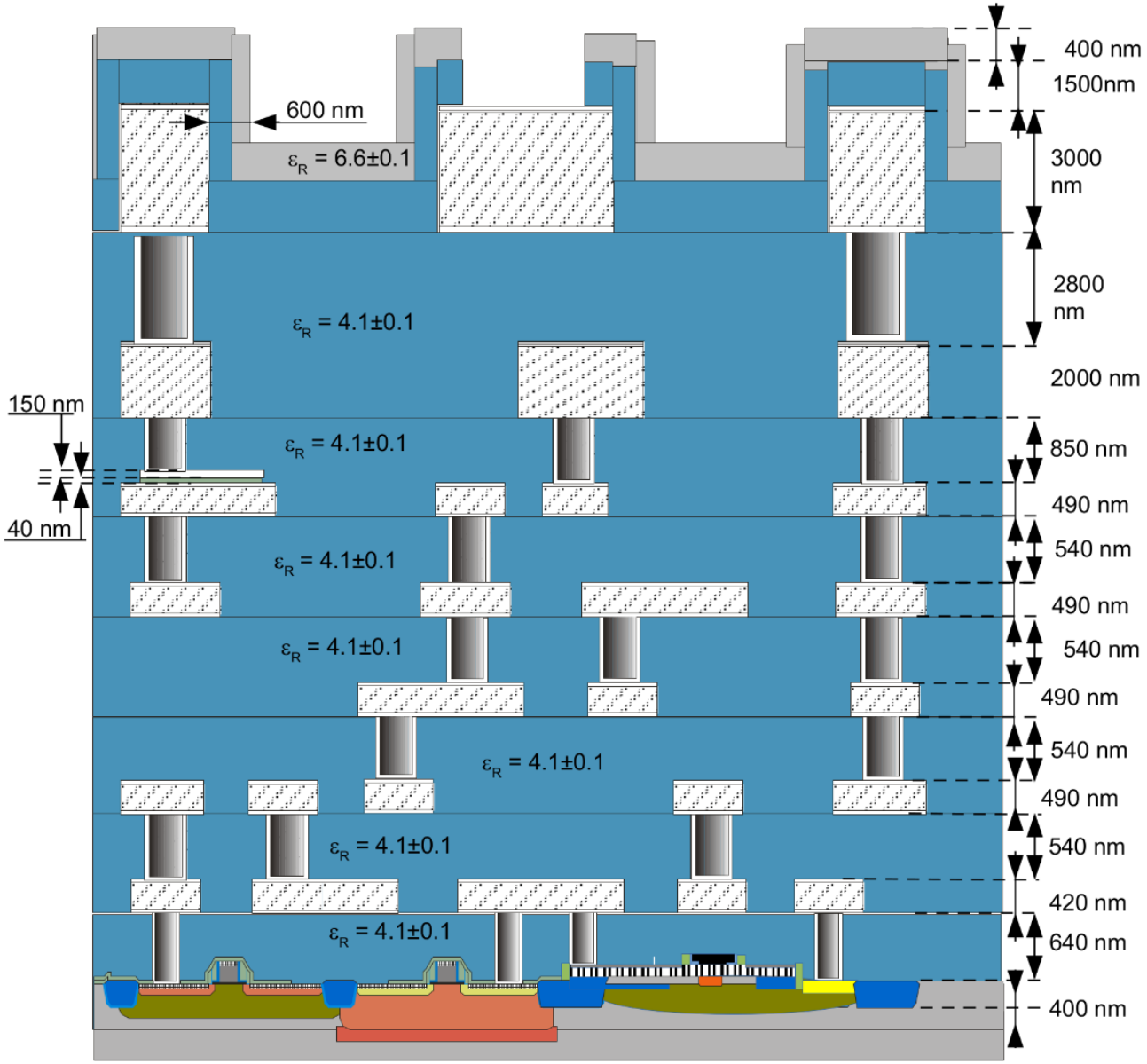
Routing Direction: e.g. Sky130



sky130: Cross section



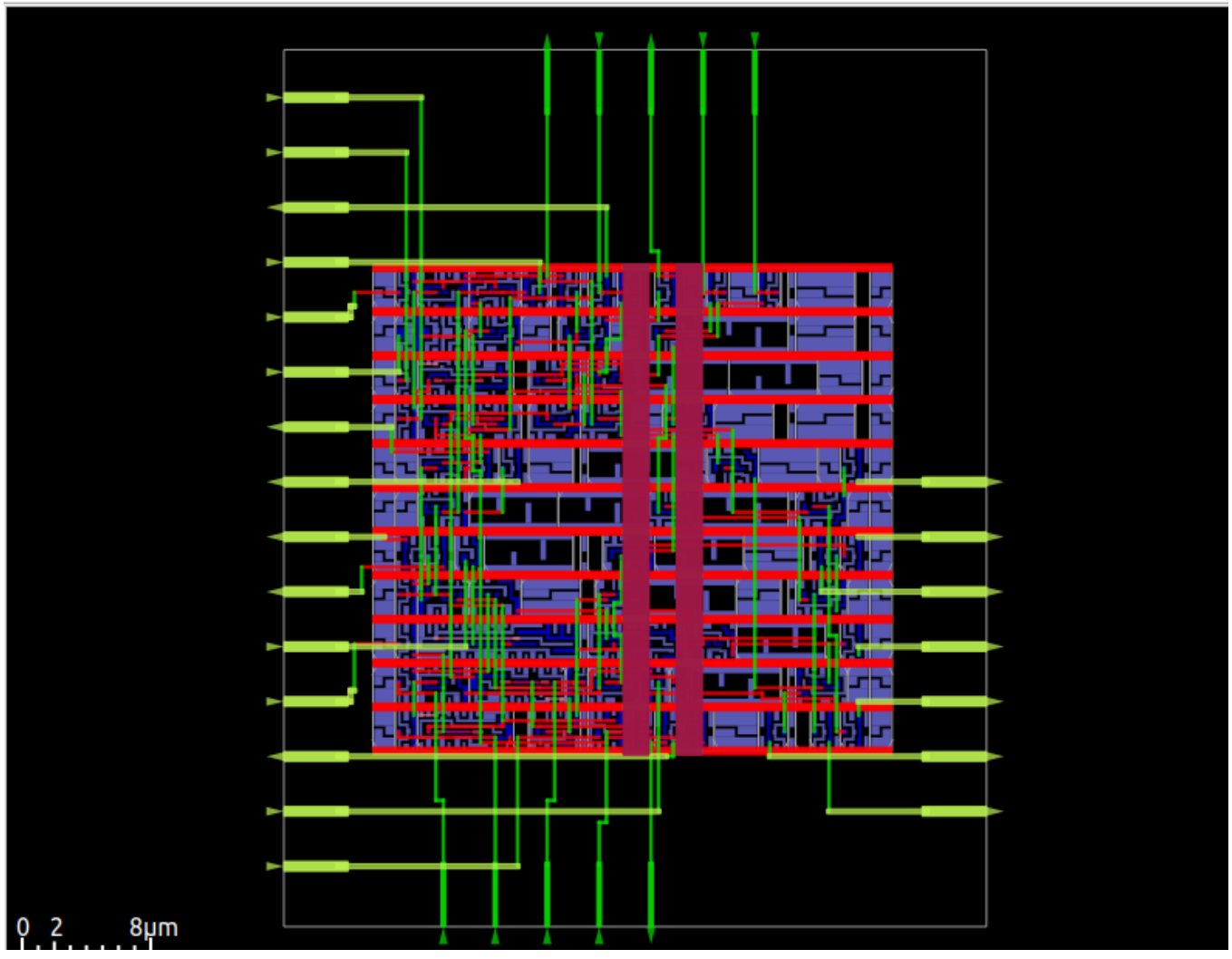
IHP: Cross section



More rules

- minimum wire width
- minimum distance between two wires (pitch)
- Components have to be on grid of 0.005 μm

Place and Route (PnR) or Layout Synthesis



`expmob1_n4 16 bits`

Comparison

Counting gates `expmob2_n14`

- Runtime `4m 44s`

Comparison

Counting gates `expmob2_n14`

- Runtime `4m 44s`
- Closer to circuit complexity

Comparison

Counting gates `expmob2_n14`

- Runtime `4m 44s`
- Closer to circuit complexity

PnR Layout `expmob2_n14`

- Runtime `19h 32m 37s`

Comparison

Counting gates `expmob2_n14`

- Runtime `4m 44s`
- Closer to circuit complexity

PnR Layout `expmob2_n14`

- Runtime `19h 32m 37s`

Gates Area	Wiring Area	Core Area
1222 μm^2	328μm^2	

`expmob2_n5`

Comparison

Counting gates `expmob2_n14`

- Runtime `4m 44s`
- Closer to circuit complexity

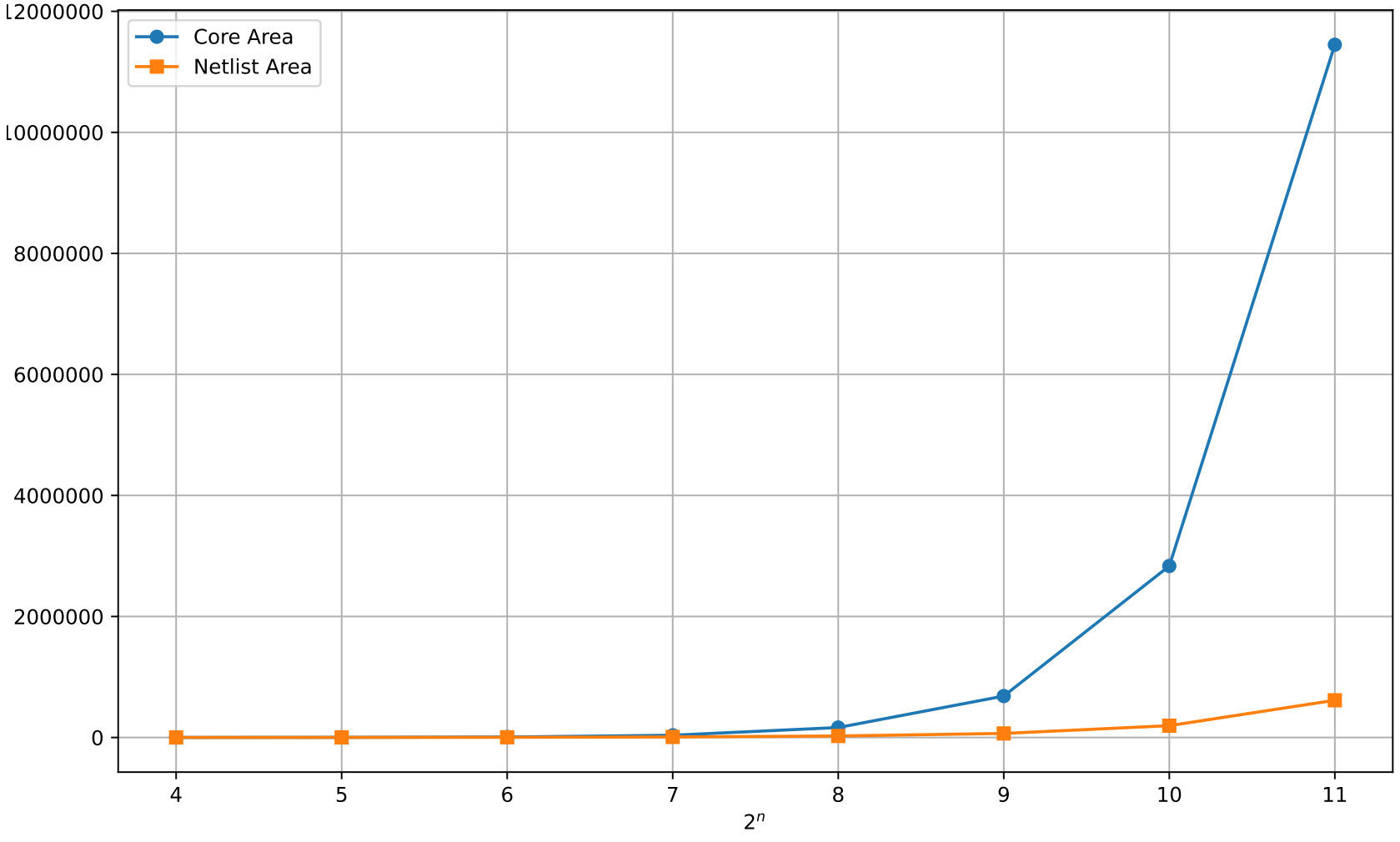
PnR Layout `expmob2_n14`

- Runtime `19h 32m 37s`

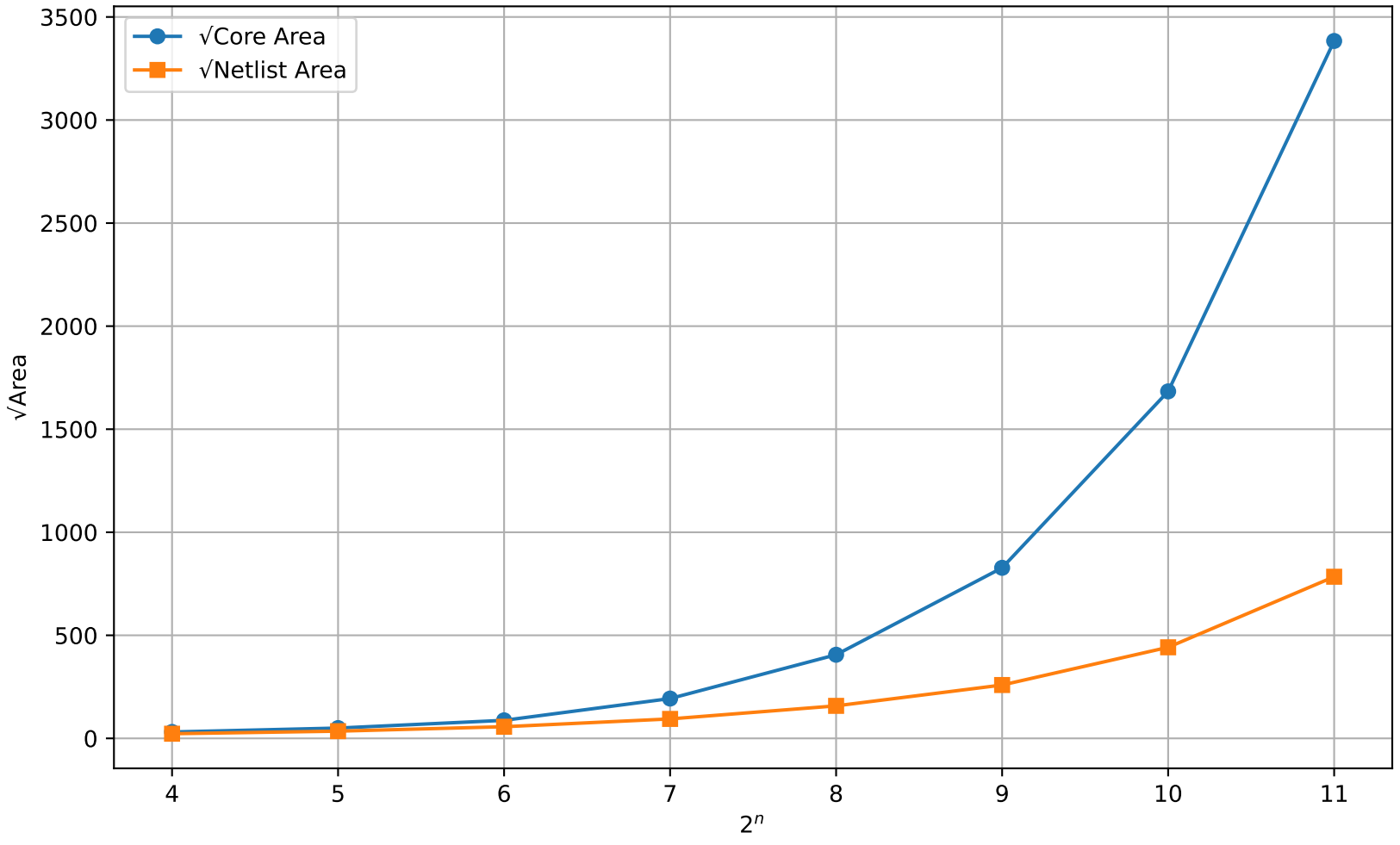
Gates Area	Wiring Area	Core Area
1222 μm^2	328μm^2	2499μm^2

`expmob2_n5`

Netlist Area vs. Layout area



Netlist Area vs. Layout area



Communication Complexity

Two-way Communication Complexity

$$f : X_A \times X_B \to Y_A \times Y_B, \quad |X_A| = |X_B|$$

Alice	**Bob**
Knows X_A	Knows X_B
Wishes to compute Y_A	Wishes to know Y_B
$\xrightarrow{m_0}$	
$\xleftarrow{m_1}$	
$\dots$	
computes y_a	computes y_b

Two-way Communication Complexity

$$f : X_A \times X_B \to Y_A \times Y_B, \quad |X_A| = |X_B|$$

Alice **Bob**

Knows X_A Knows X_B

Wishes to compute Y_A Wishes to know Y_B

$$\xrightarrow{\quad m_0 \quad}$$

$$\xleftarrow{\quad m_1 \quad}$$

$$\dots$$

computes y_a computes y_b

Communication complexity := minimum number of bits exchanged

Two-way Communication Complexity

$$f : X_A \times X_B \to Y_A \times Y_B, \quad |X_A| = |X_B|$$

Alice **Bob**

Knows X_A Knows X_B

Wishes to compute Y_A Wishes to know Y_B

$m_0 \longrightarrow$

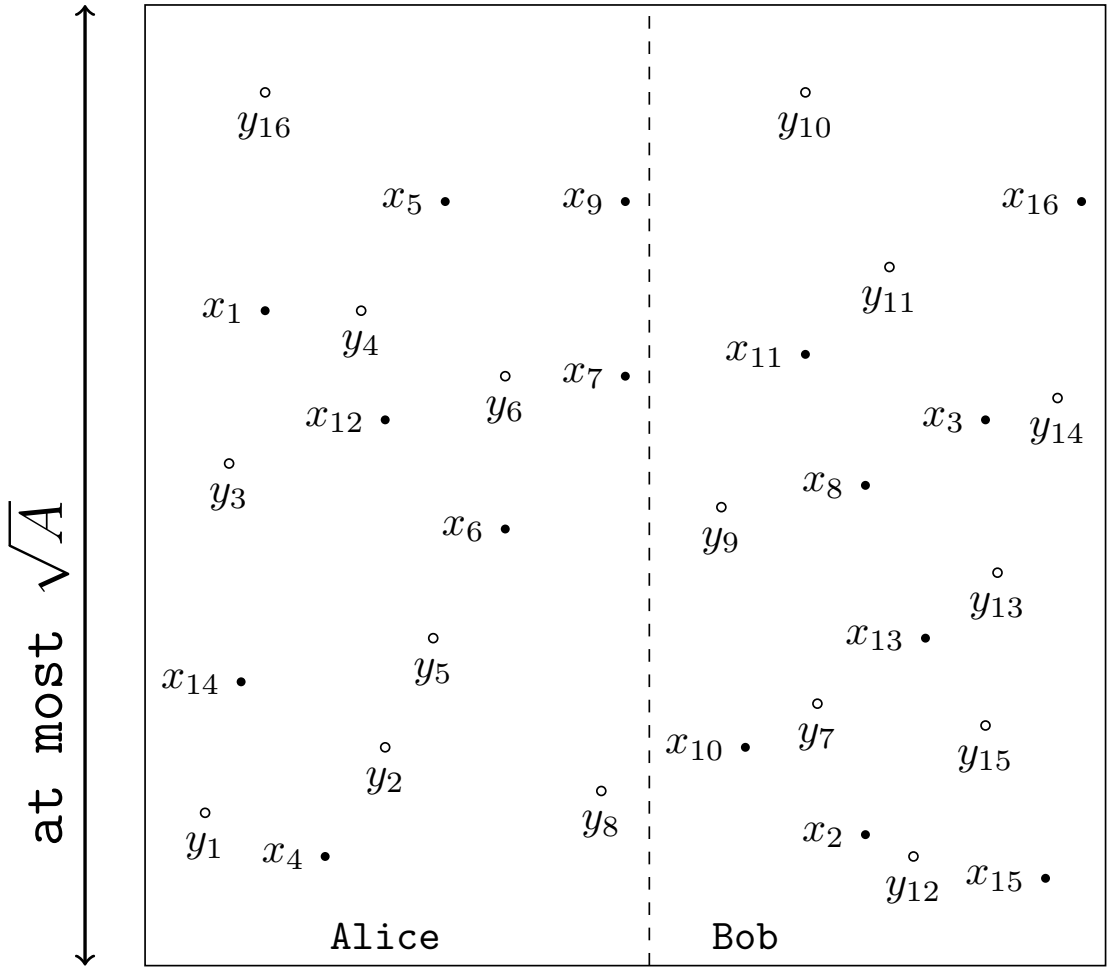
$\longleftarrow m_1$

$\ldots$

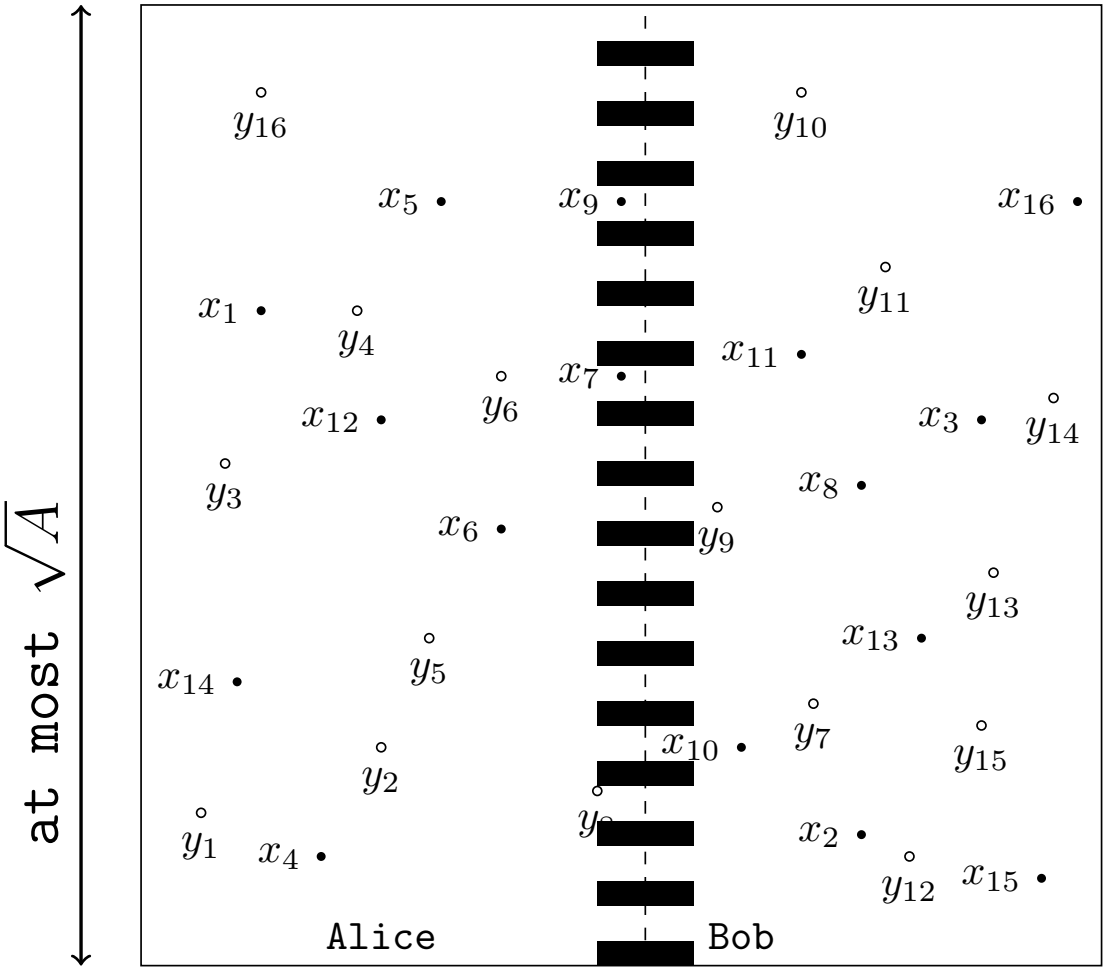
computes y_a computes y_b

Communication complexity := minimum number of bits exchanged
Both parties **don't** have to learn all the output bits.

Relationship to circuits: space separation

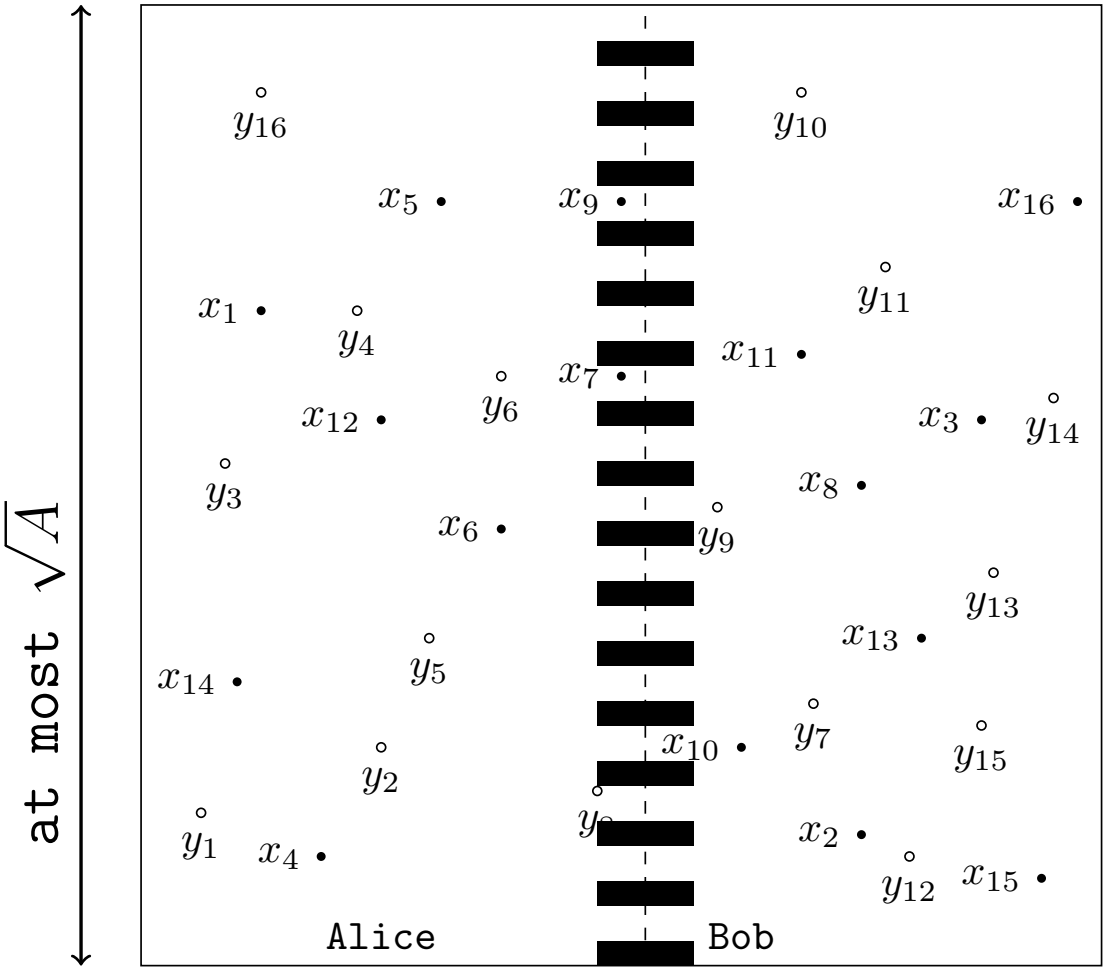


Relationship to circuits: space separation



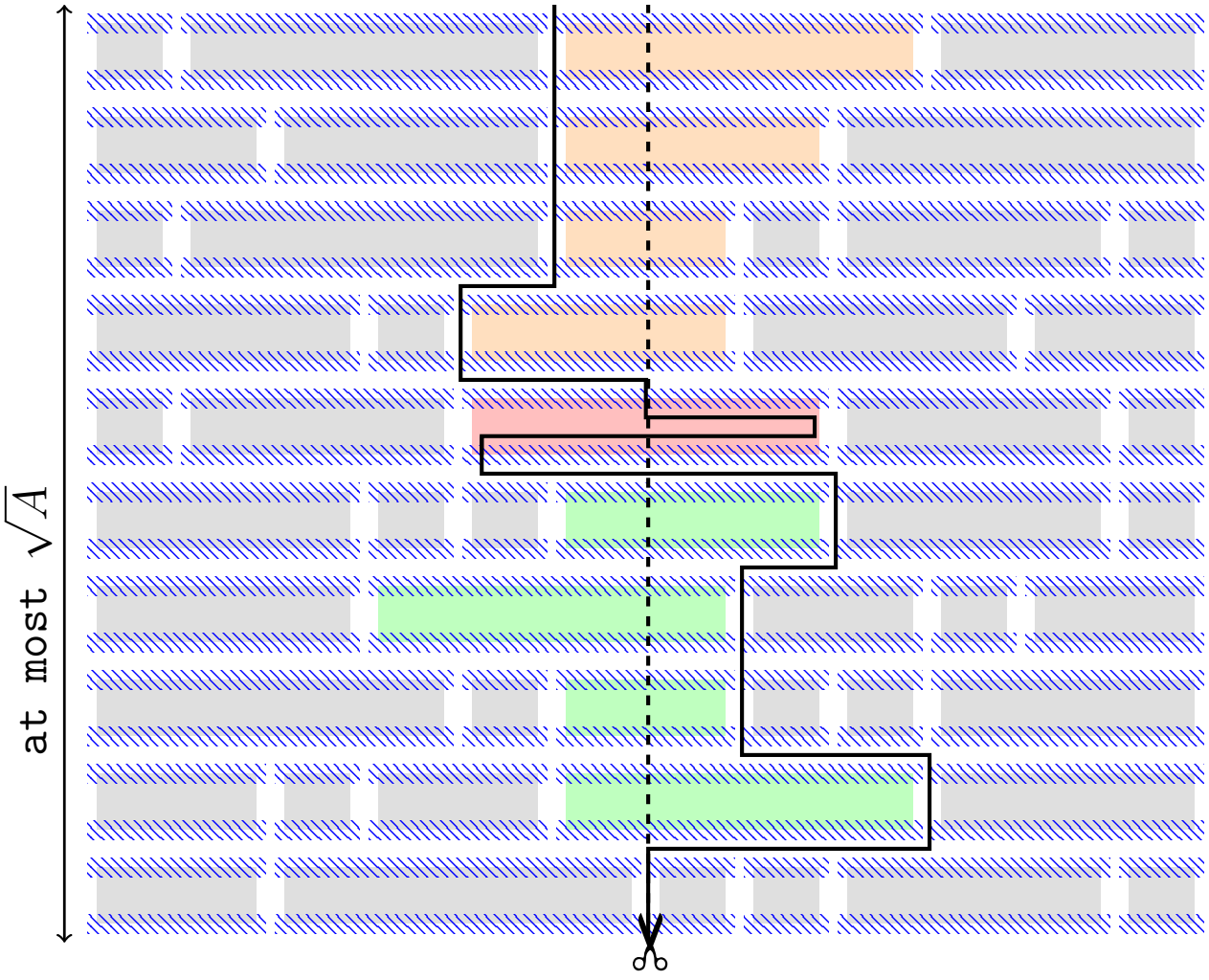
1. At most $\Omega\left(\sqrt{A}\right)$ wires cross both sections

Relationship to circuits: space separation

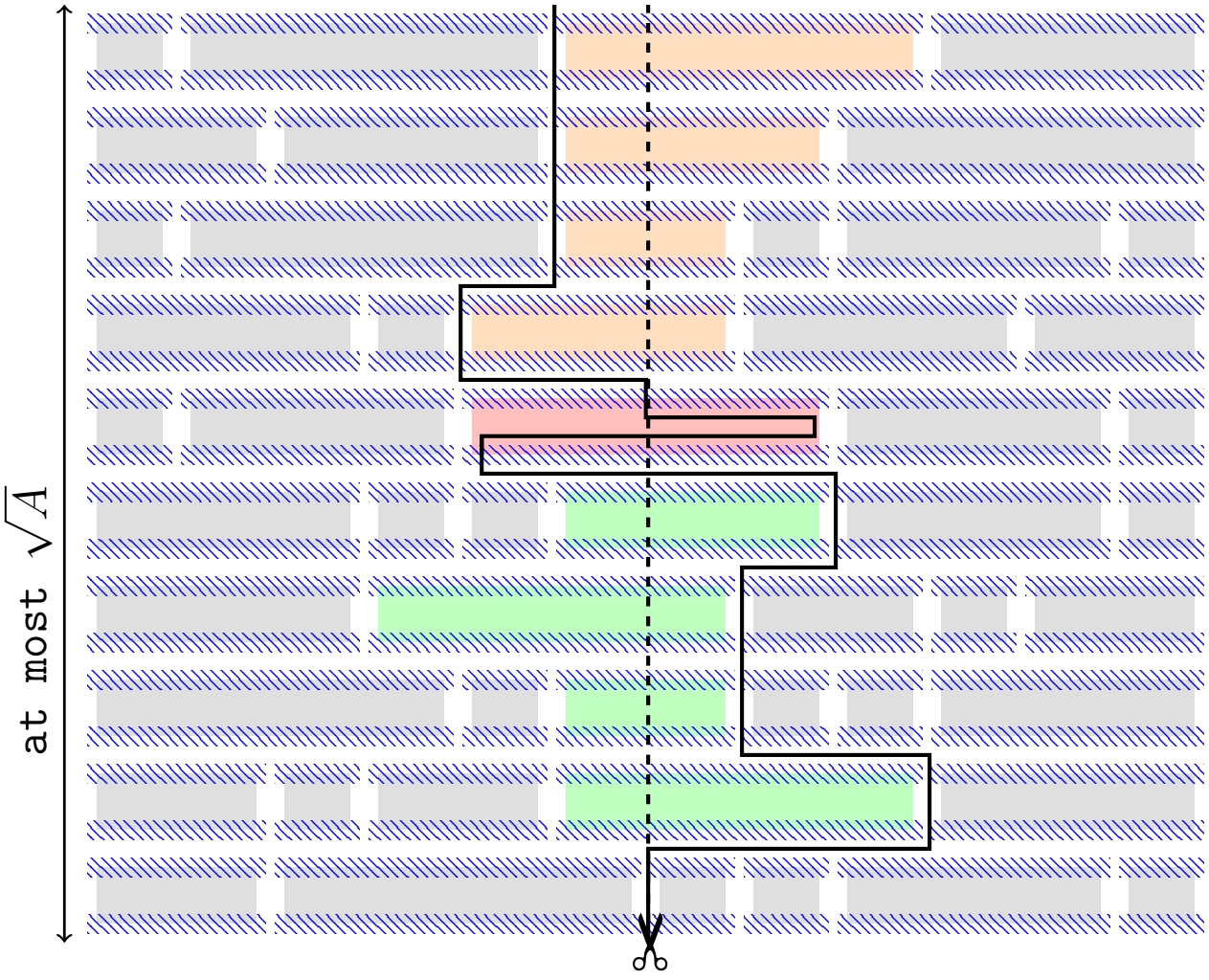


1. At most $\Omega\left(\sqrt{A}\right)$ wires cross both sections
2. After `T` `cycles`, at most $\Omega\left(\sqrt{A} \cdot T\right)$ bits are exchanged

Relationship to circuits: space separation PDK

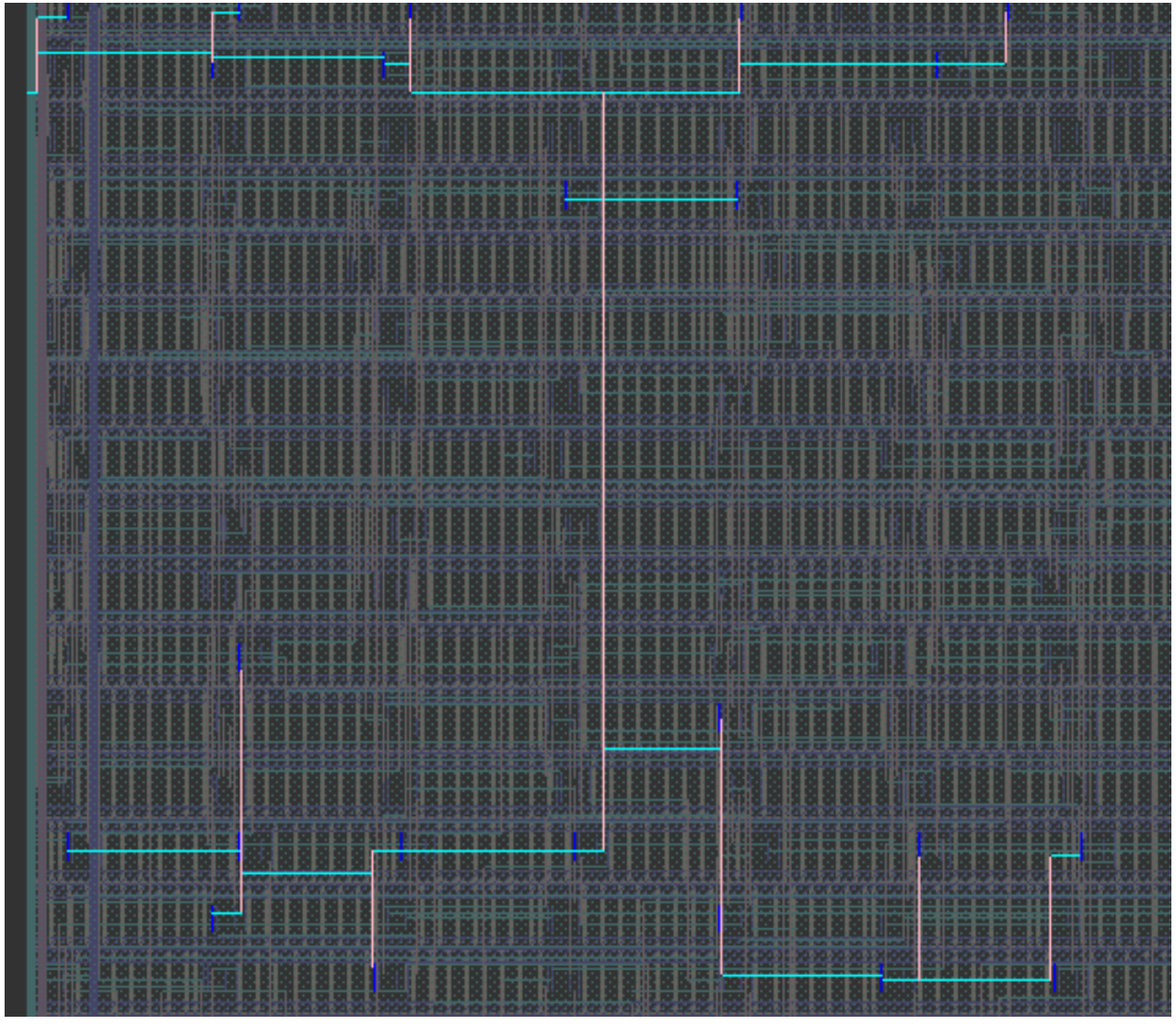


Relationship to circuits: space separation PDK



$$\sqrt{A} \cdot T \geq \Omega\left(twCC(f)\right)$$

e.g. Rotation circuit



Wires that deliver a single input

Time separation

If all input ports all present then $A \geq \Omega(n)$

Time separation

If all input ports all present then $A \geq \Omega(n)$

Assume, the function has `64-bits` inputs, but the circuit accepts only `8-bits` of the input at a time unit.

Time separation

If all input ports all present then $A \geq \Omega(n)$

Assume, the function has `64-bits` inputs, but the circuit accepts only `8-bits` of the input at a time unit.

How can we bound the area A?

One-Way Communication Complexity

How to compute $f : \{0,1\}^{2n} \to \{0,1\}^{2m}$?

One-Way Communication Complexity

How to compute $f : \{0,1\}^{2n} \to \{0,1\}^{2m}$?

$$f : X_A \times X_B \to Y_A \times Y_B, \quad |X_A| = |X_B|$$

Alice **Bob**

Knows X_A Knows X_B

b_0 ⟶

b_1 ⟶

$\dots$

computes y_b

One-Way Communication Complexity

How to compute $f : \{0,1\}^{2n} \to \{0,1\}^{2m}$?

$$f : X_A \times X_B \to Y_A \times Y_B, \quad |X_A| = |X_B|$$

Alice **Bob**

Knows X_A Knows X_B

b_0 ⟶

b_1 ⟶

$\dots$

computes y_b

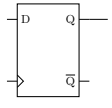
Communication complexity := minimum number of bits Alice sends

Relationship to circuits: time separation

Alice **Bob**

Simulate the circuit

for first 1/2 inputs



memory contents

$\longrightarrow$

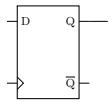
computes y_b

Relationship to circuits: time separation

Alice **Bob**

Simulate the circuit

for first 1/2 inputs



memory contents

$\longrightarrow$

computes y_b

$$A \geq \Omega\left(owCC(f)\right)$$

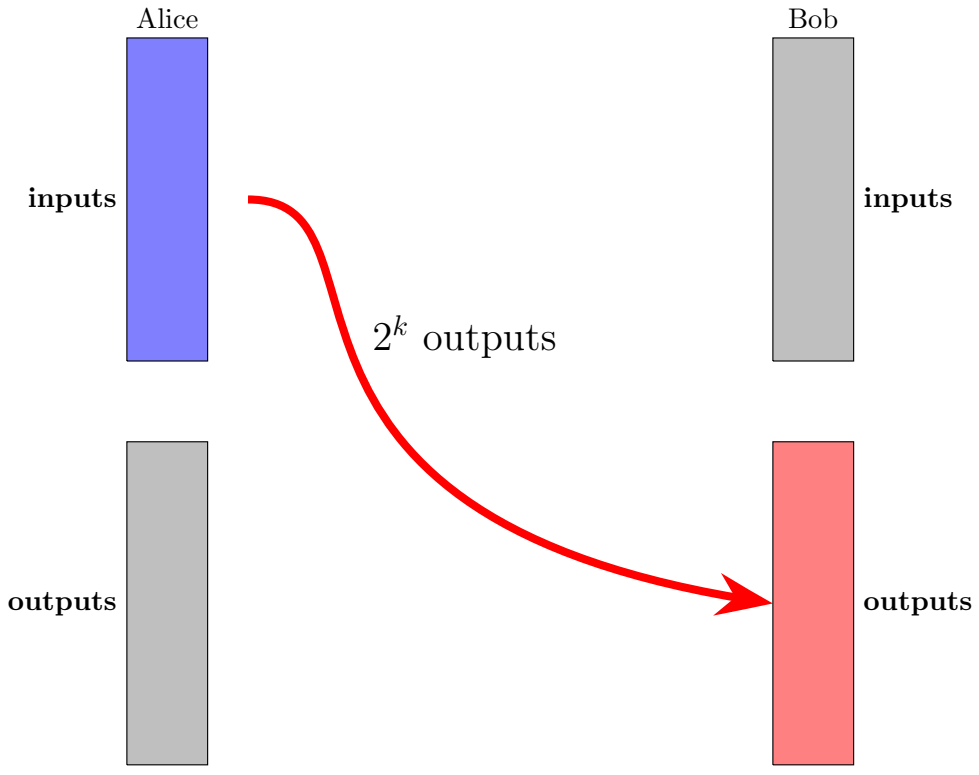
Bounding techniques

Proof techniques: Flow argument

$$f : X_A \times X_B \to Y_A \times Y_B, |X_A| = |X_B|$$

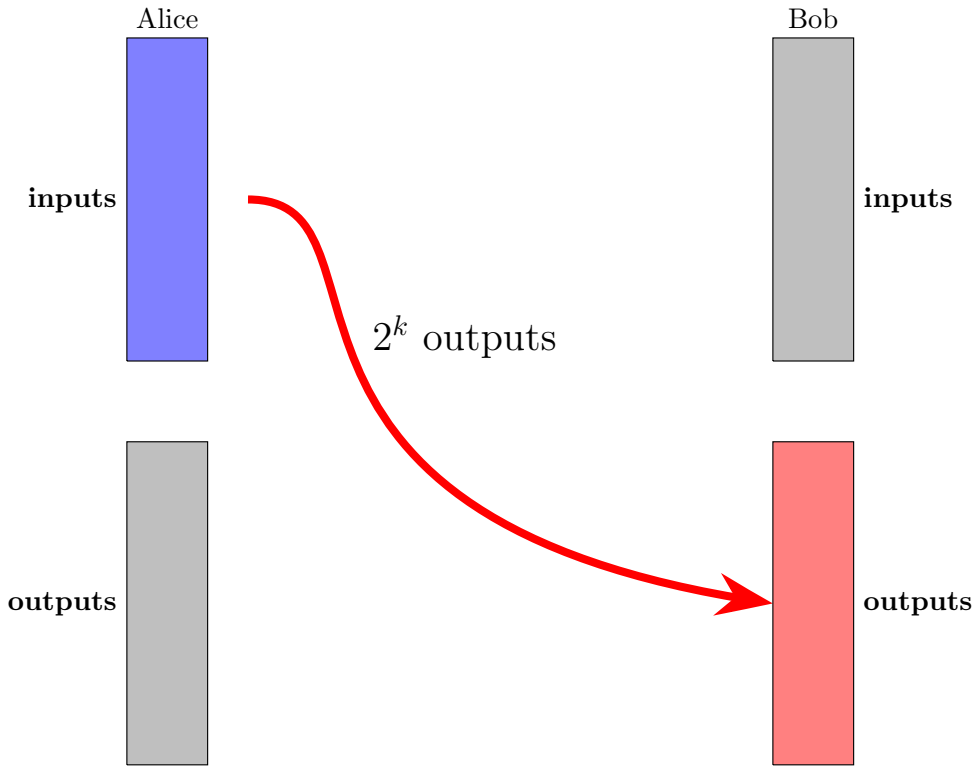
Proof techniques: Flow argument

$$f : X_A \times X_B \to Y_A \times Y_B, |X_A| = |X_B|$$



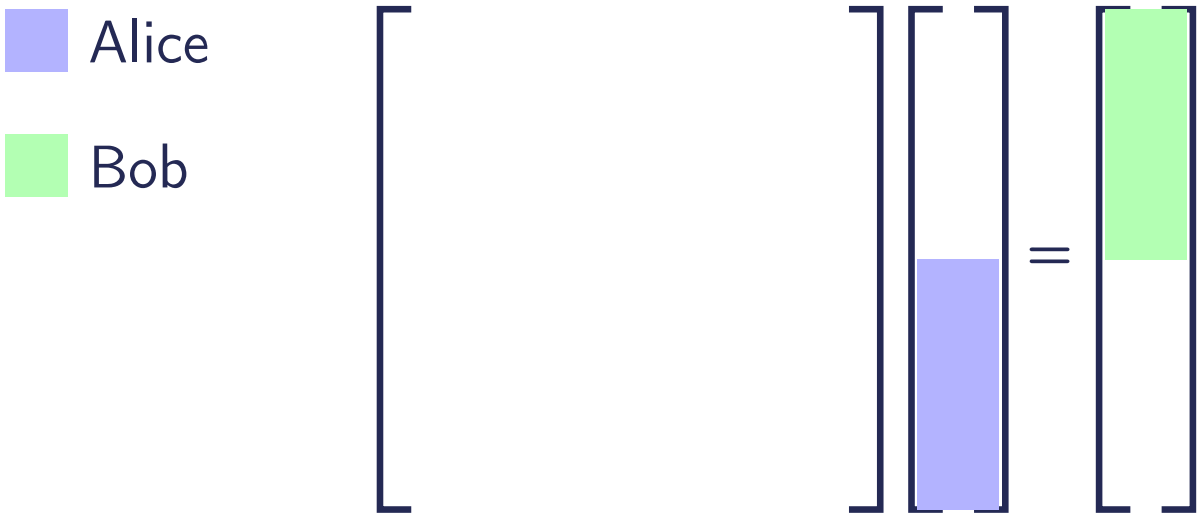
Proof techniques: Flow argument

$$f : X_A \times X_B \to Y_A \times Y_B, |X_A| = |X_B|$$

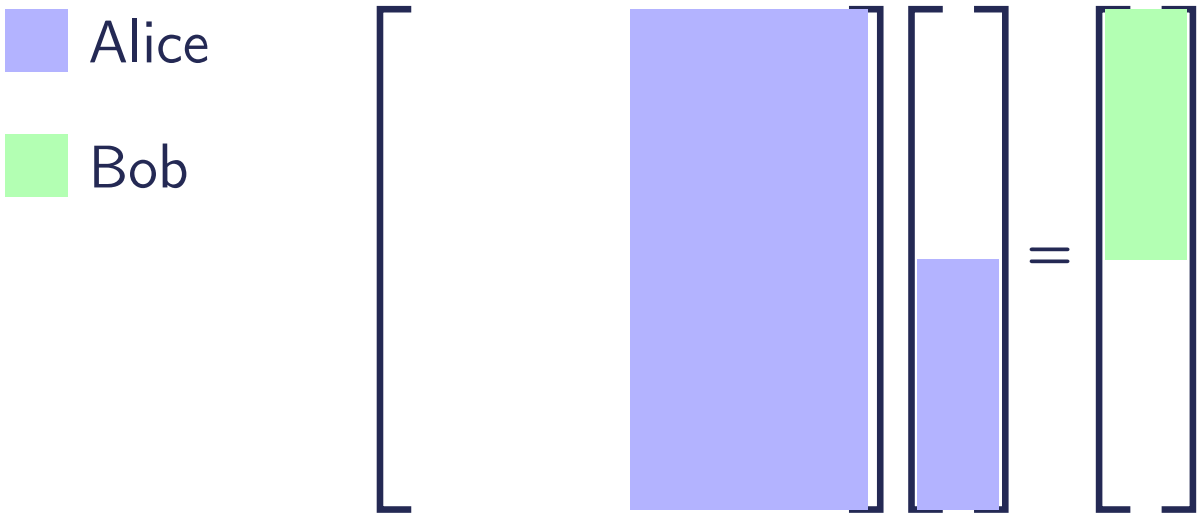


If Alice can generate 2^k different outputs in Bob's part, then she must send at least k-bits.

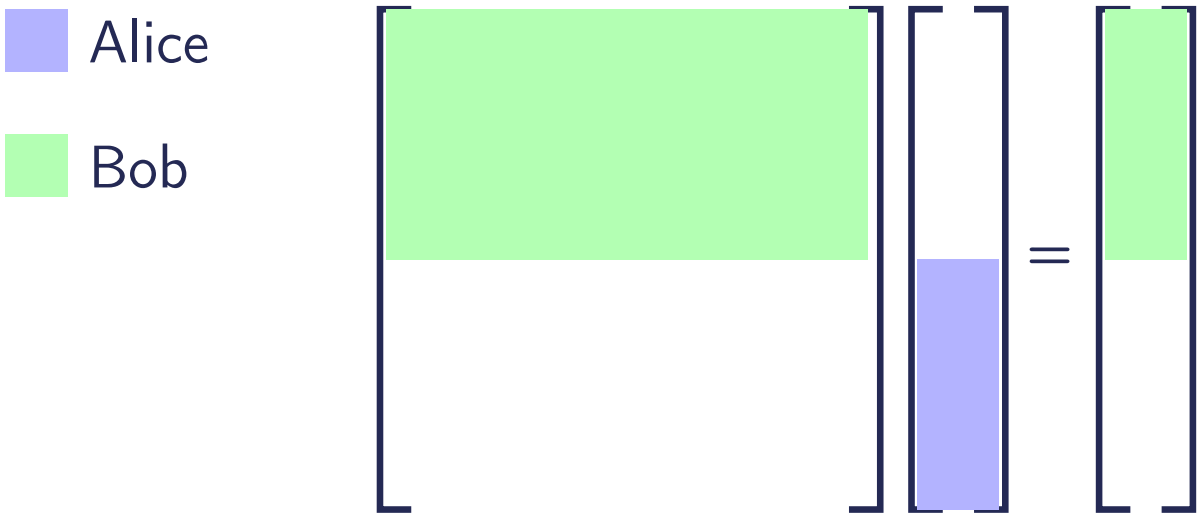
Proof techniques: Rank method $f(x) := Ax = b$



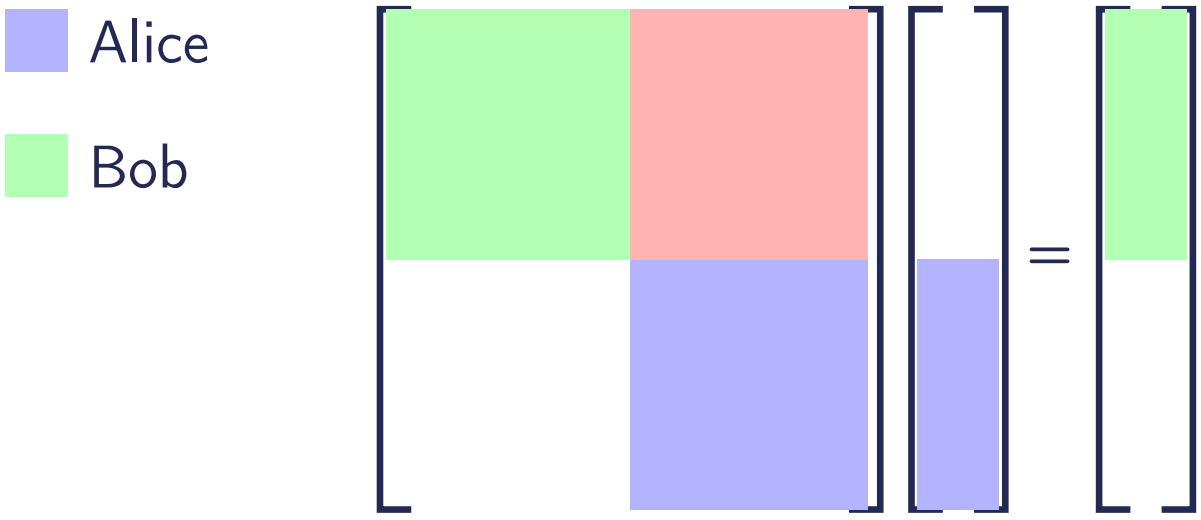
Proof techniques: Rank method $f(x) := Ax = b$



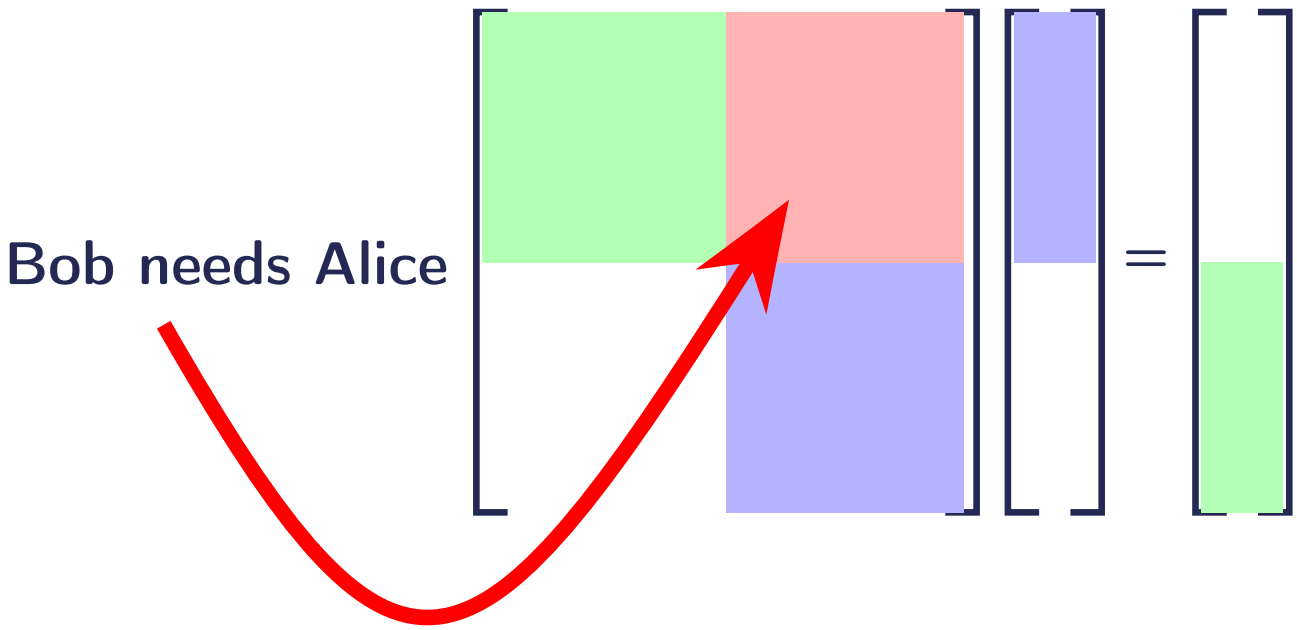
Proof techniques: Rank method $f(x) := Ax = b$



Proof techniques: Rank method $f(x) := Ax = b$



Proof techniques: Rank method $f(x) := Ax = b$



Reduce it to a binary problem

If we can use $f : \{0,1\}^{2n} \to \{0,1\}^m$ to compute $g : \{0,1\}^{2n} \to \{0,1\}$

Reduce it to a binary problem

If we can use $f : \{0,1\}^{2n} \to \{0,1\}^m$ to compute $g : \{0,1\}^{2n} \to \{0,1\}$ then

$$twCC(f) \geq twCC(g)$$

$$owCC(f) \geq owCC(g)$$

Reduce it to a binary problem

If we can use $f : \{0,1\}^{2n} \to \{0,1\}^m$ to compute $g : \{0,1\}^{2n} \to \{0,1\}$ then

$$twCC(f) \geq twCC(g)$$

$$owCC(f) \geq owCC(g)$$

Numerous techniques: *Rectangles, fooling sets, rank, etc...*

Möbius Transform

Definition

Möbius Transform

It converts truth table of a Boolean function to ANF, and vice versa.

Definition

Möbius Transform

It converts truth table of a Boolean function to ANF, and vice versa. Let $N := 2^n$

$$\begin{aligned} \text{Möbius}_N : \{0,1\}^{2^n} &\to \{0,1\}^{2^n} \\ x &\to M_n \cdot x \end{aligned}$$

Definition

Möbius Transform

It converts truth table of a Boolean function to ANF, and vice versa. Let $N := 2^n$

$$\text{Möbius}_N : \{0,1\}^{2^n} \to \{0,1\}^{2^n}$$
$$x \to M_n \cdot x$$

- $M_1 = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$

Definition

Möbius Transform

It converts truth table of a Boolean function to ANF, and vice versa. Let $N := 2^n$

$$\begin{aligned} \text{Möbius}_N : \{0,1\}^{2^n} &\to \{0,1\}^{2^n} \\ x &\to M_n \cdot x \end{aligned}$$

- $M_1 = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$
- $M_n = \begin{pmatrix} M_{n-1} & 0 \\ M_{n-1} & M_{n-1} \end{pmatrix}$

From now on, we assume there are $2N$ i.e. Möbius_{2N}

e.g. N=32

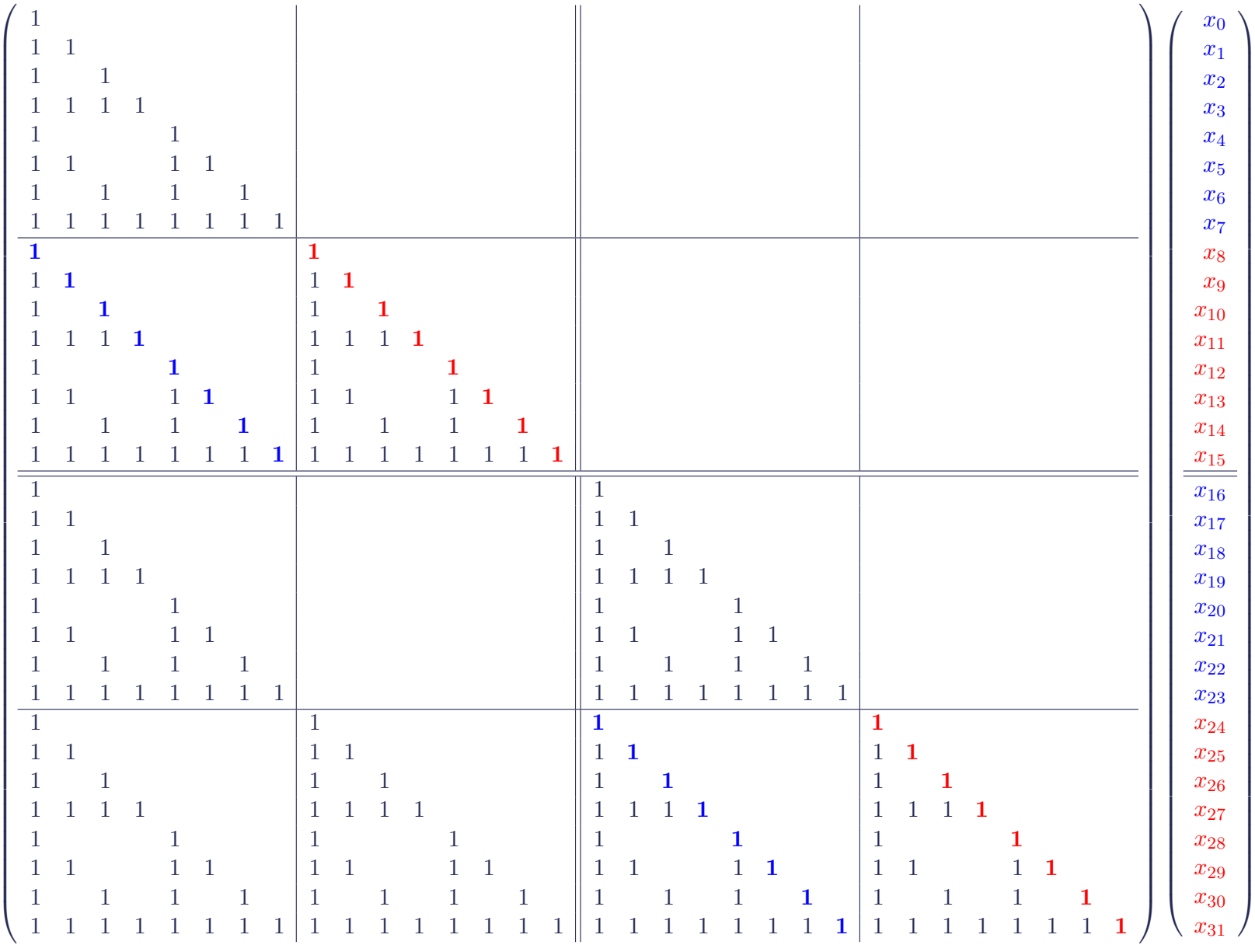
$$
\left(\begin{array}{cccccccc|cccccccc||cccccccc|cccccccc}
1 & & & & & & & & 1 & & & & & & & & & & & & & & & & & & & & \\
1 & 1 & & & & & & & 1 & 1 & & & & & & & & & & & & & & & & & & & \\
1 & & 1 & & & & & & 1 & & 1 & & & & & & & & & & & & & & & & & & \\
1 & 1 & 1 & 1 & & & & & 1 & 1 & 1 & 1 & & & & & & & & & & & & & & & & \\
1 & & & & 1 & & & & 1 & & & & 1 & & & & & & & & & & & & & & & & \\
1 & 1 & & & 1 & 1 & & & 1 & 1 & & & 1 & 1 & & & & & & & & & & & & & & \\
1 & & 1 & & 1 & & 1 & & 1 & & 1 & & 1 & & 1 & & & & & & & & & & & & & \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & & & & & & & & & & & & & \\
\hline
1 & & & & & & & & 1 & & & & & & & & 1 & & & & & & & & & & & & \\
1 & 1 & & & & & & & 1 & 1 & & & & & & & 1 & 1 & & & & & & & & & & & \\
1 & & 1 & & & & & & 1 & & 1 & & & & & & 1 & & 1 & & & & & & & & & & \\
1 & 1 & 1 & 1 & & & & & 1 & 1 & 1 & 1 & & & & & 1 & 1 & 1 & 1 & & & & & & & & & \\
1 & & & & 1 & & & & 1 & & & & 1 & & & & 1 & & & & 1 & & & & & & & & \\
1 & 1 & & & 1 & 1 & & & 1 & 1 & & & 1 & 1 & & & 1 & 1 & & & 1 & 1 & & & & & & & \\
1 & & 1 & & 1 & & 1 & & 1 & & 1 & & 1 & & 1 & & 1 & & 1 & & 1 & & 1 & & & & & \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & & & & & \\
\hline\hline
1 & & & & & & & & & & & & & & & & 1 & & & & & & & & & & & & \\
1 & 1 & & & & & & & & & & & & & & & 1 & 1 & & & & & & & & & & & \\
1 & & 1 & & & & & & & & & & & & & & 1 & & 1 & & & & & & & & & & \\
1 & 1 & 1 & 1 & & & & & & & & & & & & & 1 & 1 & 1 & 1 & & & & & & & & & \\
1 & & & & 1 & & & & & & & & & & & & 1 & & & & 1 & & & & & & & & \\
1 & 1 & & & 1 & 1 & & & & & & & & & & & 1 & 1 & & & 1 & 1 & & & & & & & \\
1 & & 1 & & 1 & & 1 & & & & & & & & & & 1 & & 1 & & 1 & & 1 & & & & & & \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & & & & & & & & & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & & & & & \\
\hline
1 & & & & & & & & 1 & & & & & & & & 1 & & & & & & & & 1 & & & & \\
1 & 1 & & & & & & & 1 & 1 & & & & & & & 1 & 1 & & & & & & & 1 & 1 & & & \\
1 & & 1 & & & & & & 1 & & 1 & & & & & & 1 & & 1 & & & & & & 1 & & 1 & & \\
1 & 1 & 1 & 1 & & & & & 1 & 1 & 1 & 1 & & & & & 1 & 1 & 1 & 1 & & & & & 1 & 1 & 1 & 1 & \\
1 & & & & 1 & & & & 1 & & & & 1 & & & & 1 & & & & 1 & & & & 1 & & & \\
1 & 1 & & & 1 & 1 & & & 1 & 1 & & & 1 & 1 & & & 1 & 1 & & & 1 & 1 & & & 1 & 1 & & \\
1 & & 1 & & 1 & & 1 & & 1 & & 1 & & 1 & & 1 & & 1 & & 1 & & 1 & & 1 & & 1 & & 1 & \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
\end{array}\right)
\left(\begin{array}{c}
x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \\ x_{13} \\ x_{14} \\ x_{15} \\ \hline\hline
x_{16} \\ x_{17} \\ x_{18} \\ x_{19} \\ x_{20} \\ x_{21} \\ x_{22} \\ x_{23} \\ x_{24} \\ x_{25} \\ x_{26} \\ x_{27} \\ x_{28} \\ x_{29} \\ x_{30} \\ x_{31}
\end{array}\right)
$$

As an equality tool

SCIENCES SORBONNE UNIVERSITÉ

$$
\begin{pmatrix}
1 \\
1&1 \\
1&&1 \\
1&1&1&1 \\
1&&&&1 \\
1&1&&&1&1 \\
1&&1&&1&&1 \\
1&1&1&1&1&1&1&1 \\
1&&&&&&&&1 \\
1&1&&&&&&&1&1 \\
1&&1&&&&&&1&&1 \\
1&1&1&1&&&&&1&1&1&1 \\
1&&&&1&&&&1&&&&1 \\
1&1&&&1&1&&&1&1&&&1&1 \\
1&&1&&1&&1&&1&&1&&1&&1 \\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\
1&&&&&&&&&&&&&&&&1 \\
1&1&&&&&&&&&&&&&&&1&1 \\
1&&1&&&&&&&&&&&&&&1&&1 \\
1&1&1&1&&&&&&&&&&&&&1&1&1&1 \\
1&&&&1&&&&&&&&&&&&1&&&&1 \\
1&1&&&1&1&&&&&&&&&&&1&1&&&1&1 \\
1&&1&&1&&1&&&&&&&&&&1&&1&&1&&1 \\
1&1&1&1&1&1&1&1&&&&&&&&&1&1&1&1&1&1&1&1 \\
1&&&&&&&&1&&&&&&&&1&&&&&&&&1 \\
1&1&&&&&&&1&1&&&&&&&1&1&&&&&&&1&1 \\
1&&1&&&&&&1&&1&&&&&&1&&1&&&&&&1&&1 \\
1&1&1&1&&&&&1&1&1&1&&&&&1&1&1&1&&&&&1&1&1&1 \\
1&&&&1&&&&1&&&&1&&&&1&&&&1&&&&1&&&&1 \\
1&1&&&1&1&&&1&1&&&1&1&&&1&1&&&1&1&&&1&1&&&1&1 \\
1&&1&&1&&1&&1&&1&&1&&1&&1&&1&&1&&1&&1&&1&&1&&1 \\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1
\end{pmatrix}
\begin{pmatrix}
x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \\ x_{13} \\ x_{14} \\ x_{15} \\ x_{16} \\ x_{17} \\ x_{18} \\ x_{19} \\ x_{20} \\ x_{21} \\ x_{22} \\ x_{23} \\ x_{24} \\ x_{25} \\ x_{26} \\ x_{27} \\ x_{28} \\ x_{29} \\ x_{30} \\ x_{31}
\end{pmatrix}
$$

As an equality tool



Reflections: twCC for Möbius over 2N bits

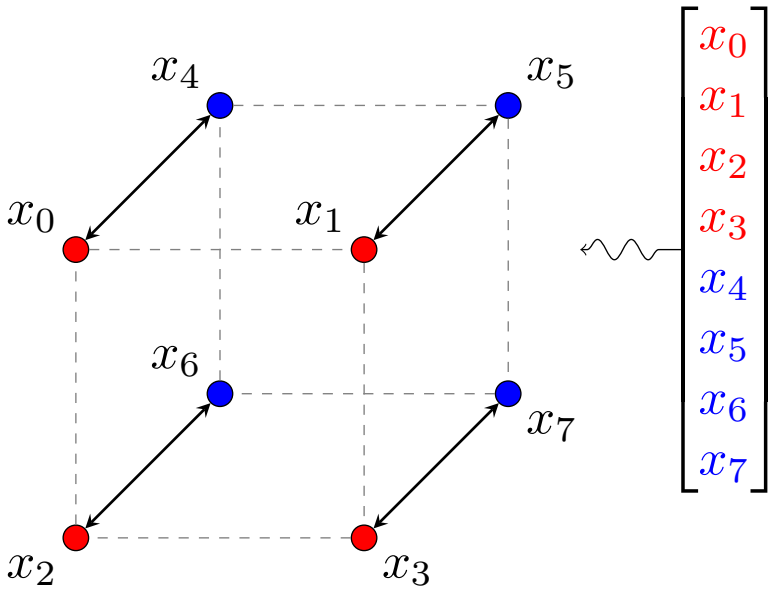
- We know $twCC\left(\mathsf{eq}_{2N}\right) \geq N$

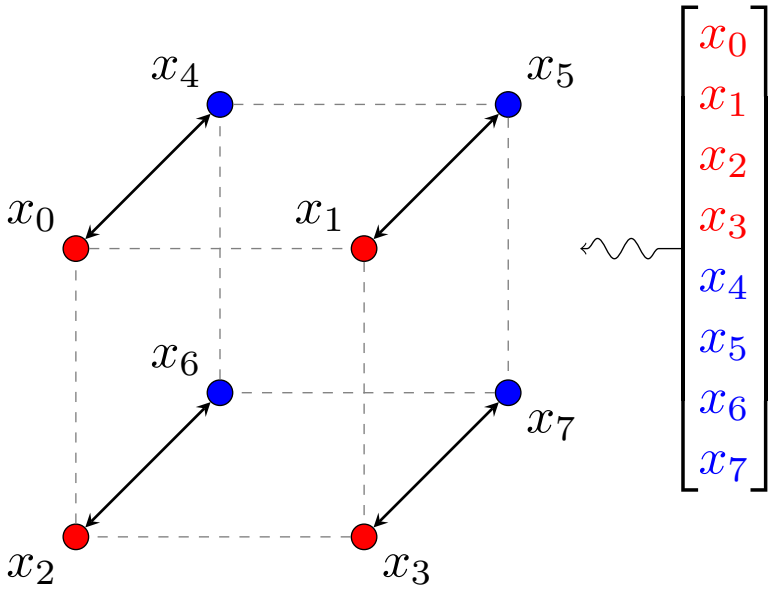
Reflections: twCC for Möbius over 2N bits

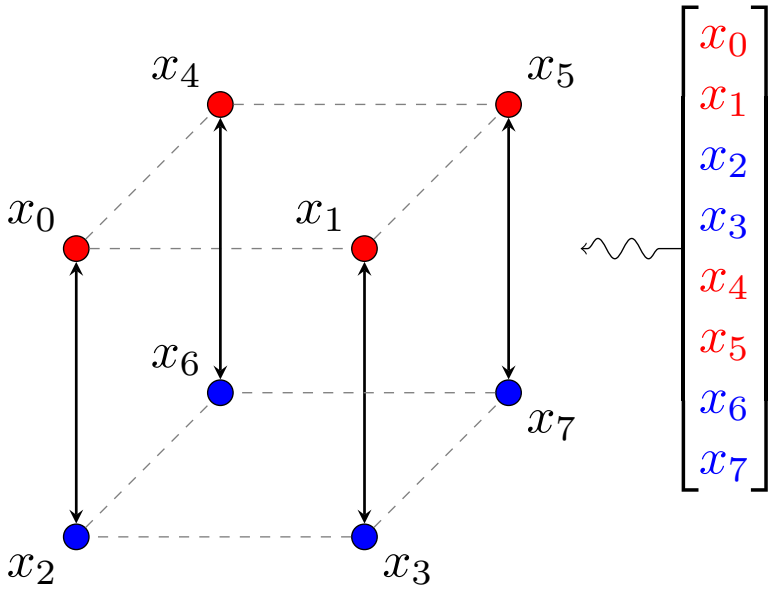
- We know $twCC\left(\mathsf{eq}_{2N}\right) \geq N$
- $twCC\left(\mathsf{M\ddot{o}bius}_{2N}\right)$ requires to study all **balanced** input partitions X_A, X_B. i.e.
  $\mathsf{M\ddot{o}bius}_{2N} : X_A \times X_B \to Y_A \times Y_B, |X_A| = |X_B| = N$

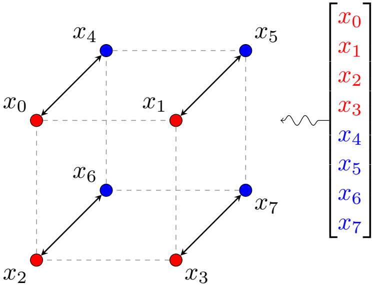
Reflections: twCC for Möbius over 2N bits

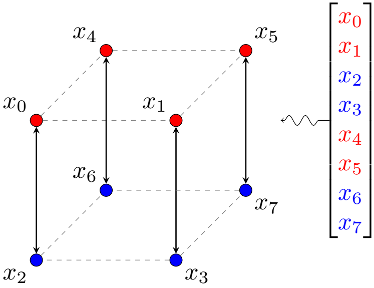
- We know $twCC\left(\mathsf{eq}_{2N}\right) \geq N$
- $twCC\left(\mathsf{M\ddot{o}bius}_{2N}\right)$ requires to study all **balanced** input partitions X_A, X_B. i.e.
  $\mathsf{M\ddot{o}bius}_{2N} : X_A \times X_B \to Y_A \times Y_B, |X_A| = |X_B| = N$
- Our reduction $\mathsf{M\ddot{o}bius}_{2N} \to \mathsf{eq}_{2N}$ works only on $2^N \cdot log_2\left(2N\right)$ partitions, but there are $\binom{2N}{N}$ partitions.

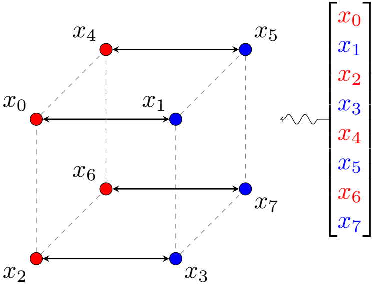

x_4
x_5
x_0
x_1
x_6
x_7
x_2
x_3
x_0
x_1
x_2
x_3
x_4
x_5
x_6
x_7

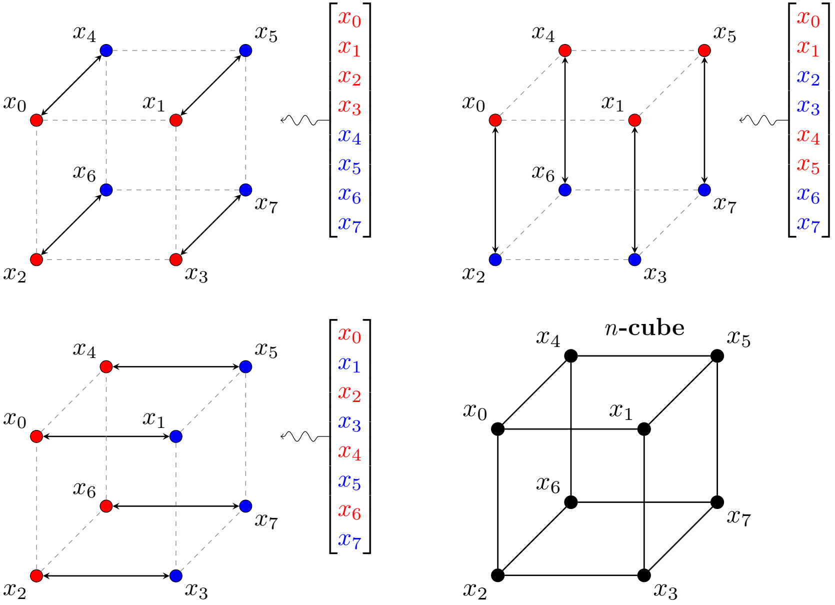

x_4
x_5
x_0
x_1
x_6
x_7
x_2
x_3
x_0 x_1 x_2 x_3 x_4 x_5 x_6 x_7


x_4
x_5
x_0
x_1
x_6
x_7
x_2
x_3
x_0 x_1 x_2 x_3 x_4 x_5 x_6 x_7


x_0 x_1 x_2 x_3 x_4 x_5 x_6 x_7
x_0 x_1 x_2 x_3 x_4 x_5 x_6 x_7

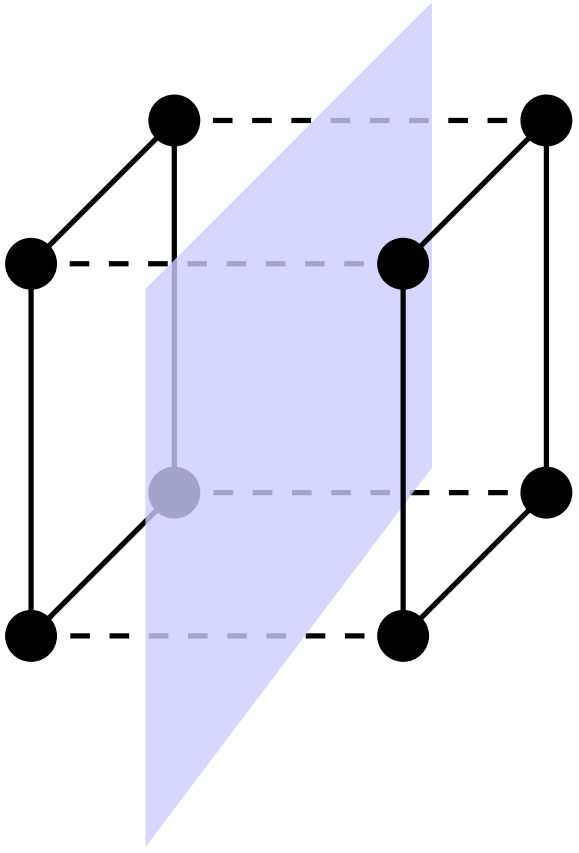

x_0 x_1 x_2 x_3 x_4 x_5 x_6 x_7
x_0 x_1 x_2 x_3 x_4 x_5 x_6 x_7


x_0 x_1 x_2 x_3 x_4 x_5 x_6 x_7
x_0 x_1 x_2 x_3 x_4 x_5 x_6 x_7


x_0
x_1
x_2
x_3
x_4
x_5
x_6
x_7
n-cube

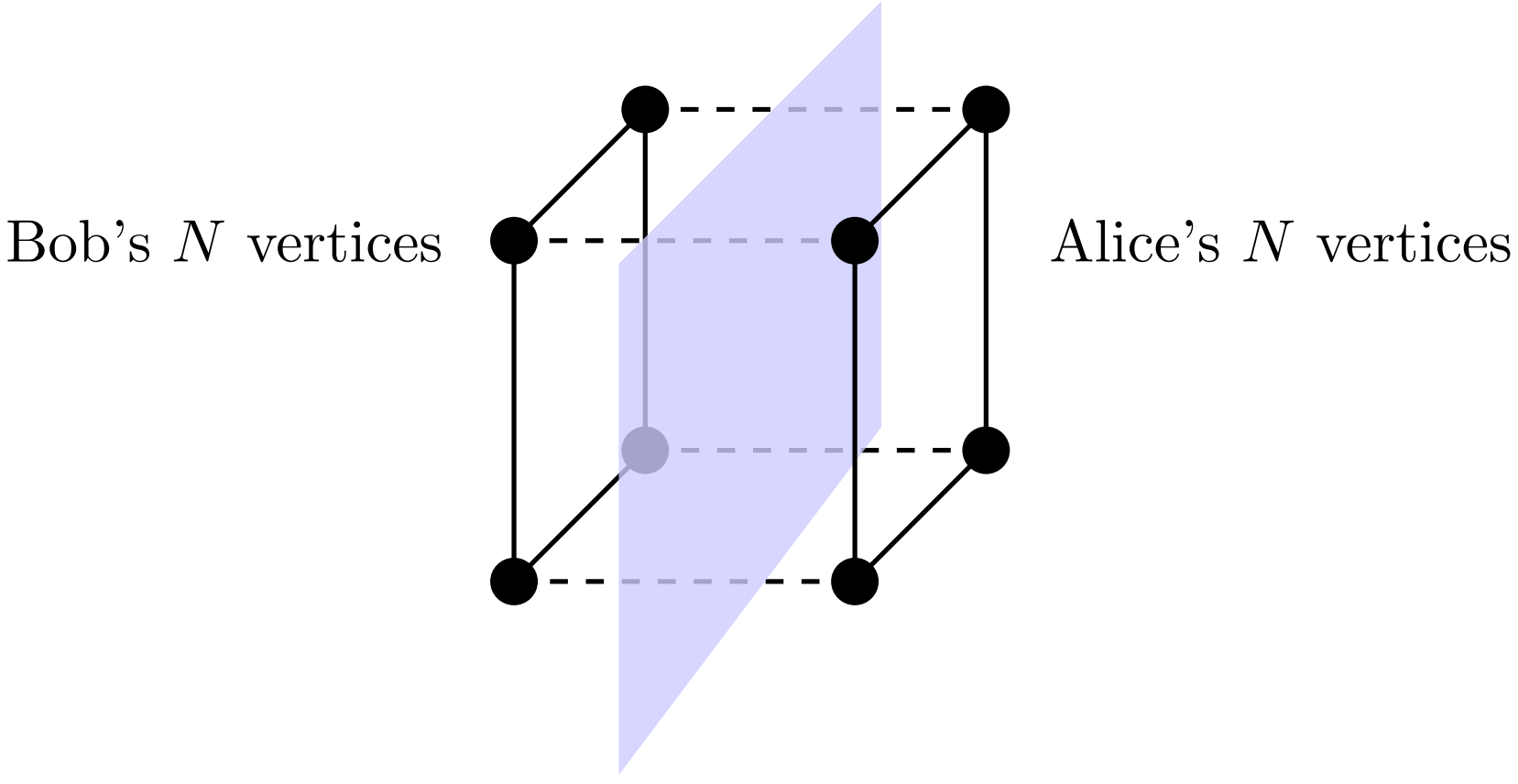
Isoperimetric Inequality: $n+1$-cube, $2N$ vertices

Bob's N vertices



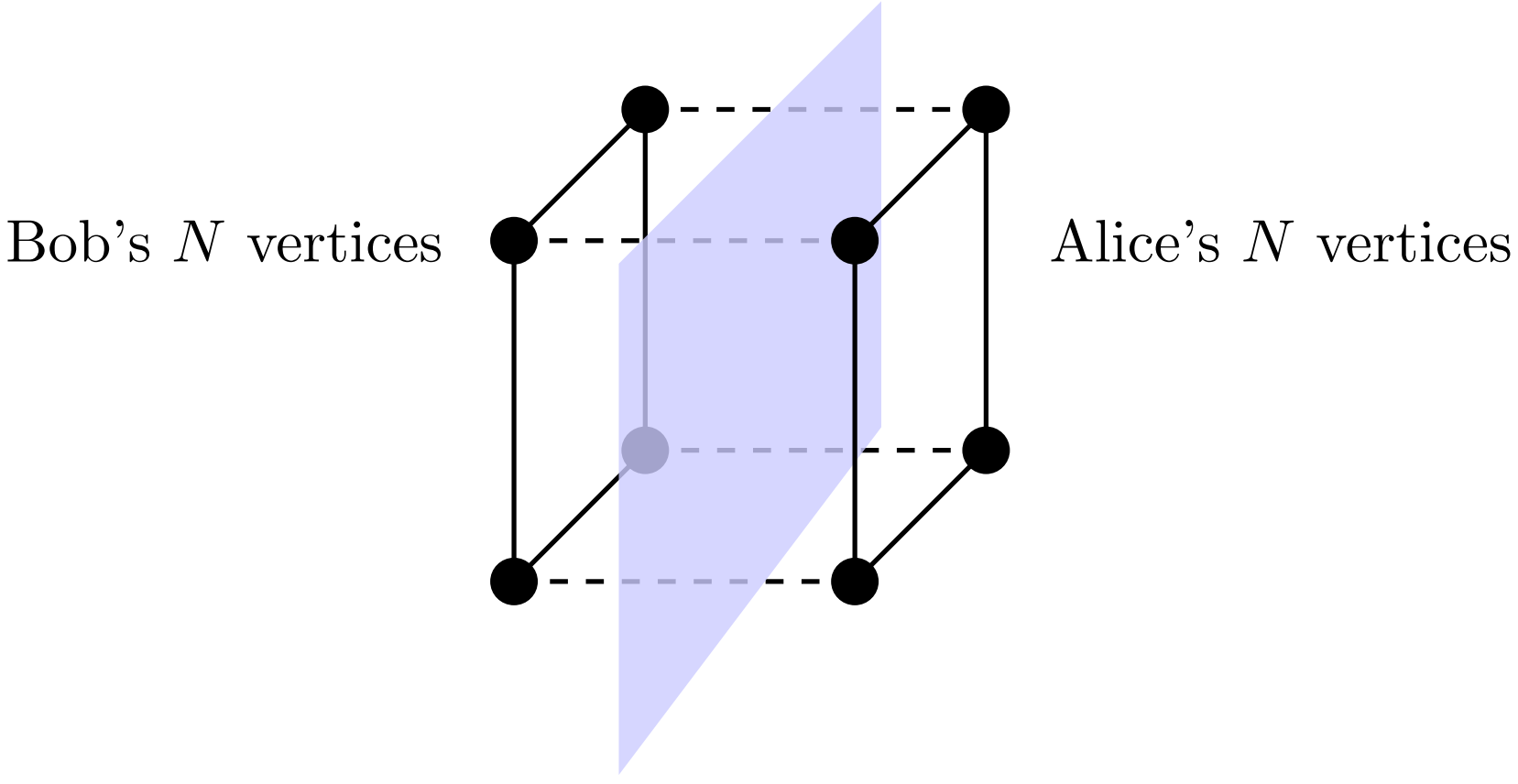
Alice's N vertices

Isoperimetric Inequality: $n+1$-cube, $2N$ vertices



∂X_A is the set of edges between Alice and Bob, thus

Isoperimetric Inequality: $n+1$-cube, $2N$ vertices



∂X_A is the set of edges between Alice and Bob, thus
$\partial X_A \geq log_2(\frac{2N}{N}) \cdot N = N$ edges between them

- $N \cdot log_2\,(2N)$ comparisons using the $log_2\,(2N)$ canonical partitions.

- $N \cdot log_2(2N)$ comparisons using the $log_2(2N)$ canonical partitions.
- Any balanced partition shares at least $\frac{N}{log_2(2N)}$ edges with **one** canonical partition.

- $N \cdot log_2(2N)$ comparisons using the $log_2(2N)$ canonical partitions.
- Any balanced partition shares at least $\frac{N}{log_2(2N)}$ edges with **one** canonical partition.
- $twCC(\text{Möbius}_{2N}) \geq \frac{N}{log_2(2N)}$ (General case)

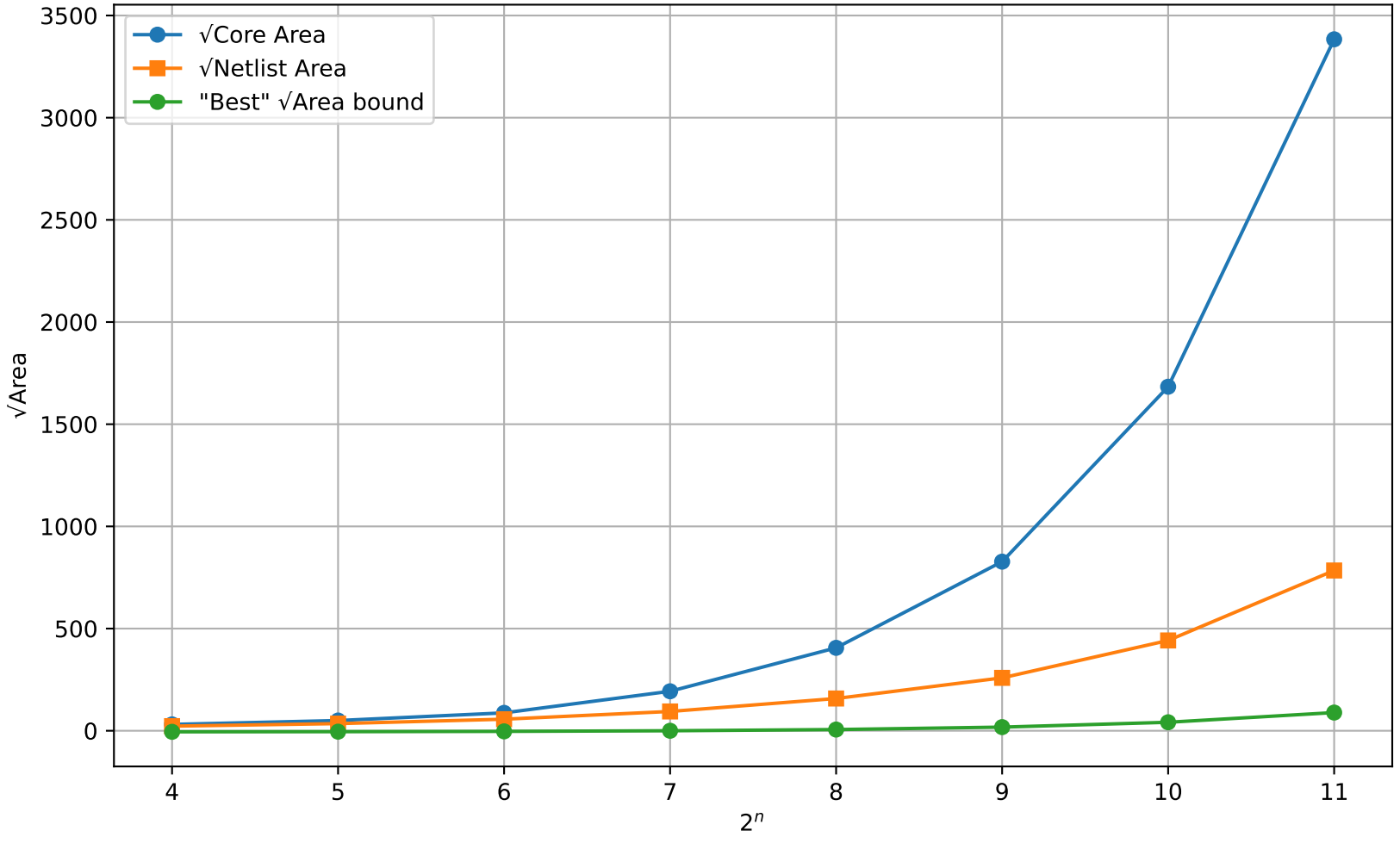
- $N \cdot log_2(2N)$ comparisons using the $log_2(2N)$ canonical partitions.
- Any balanced partition shares at least $\frac{N}{log_2(2N)}$ edges with **one** canonical partition.
- $twCC(\text{Möbius}_{2N}) \geq \frac{N}{log_2(2N)}$ (General case)
- $twCC(\text{Möbius}_{2N}) \geq N$ (Canonical Partitions)

- $N \cdot log_2(2N)$ comparisons using the $log_2(2N)$ canonical partitions.
- Any balanced partition shares at least $\frac{N}{log_2(2N)}$ edges with **one** canonical partition.
- $twCC(\text{Möbius}_{2N}) \geq \frac{N}{log_2(2N)}$ (General case)
- $twCC(\text{Möbius}_{2N}) \geq N$ (Canonical Partitions)

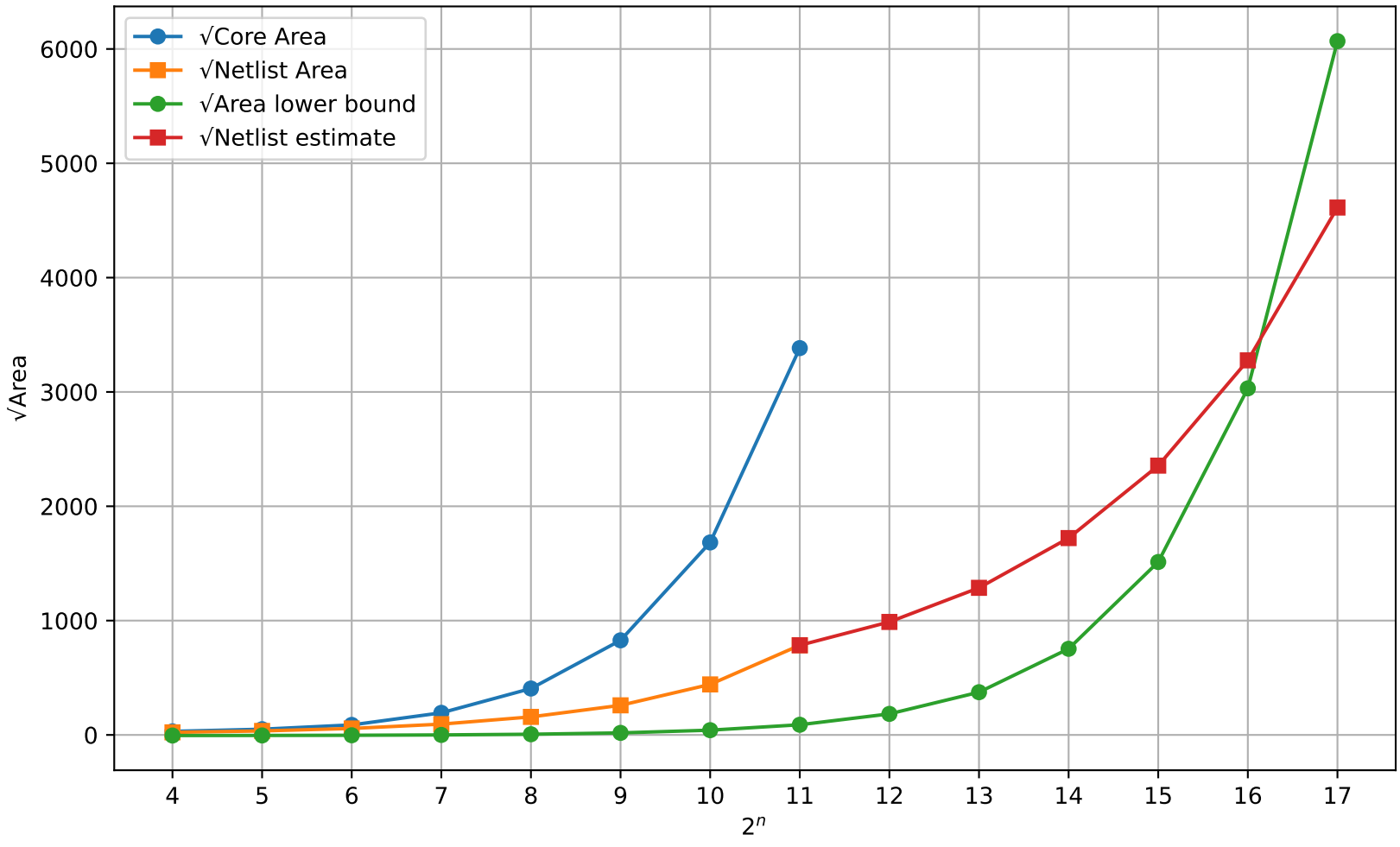
General Case	Canonical Partitions
$\sqrt{A} \cdot T \geq \Omega\left(\frac{N}{log_2(2N)}\right)$	$\sqrt{A} \cdot T \geq \Omega(N)$

Comparison

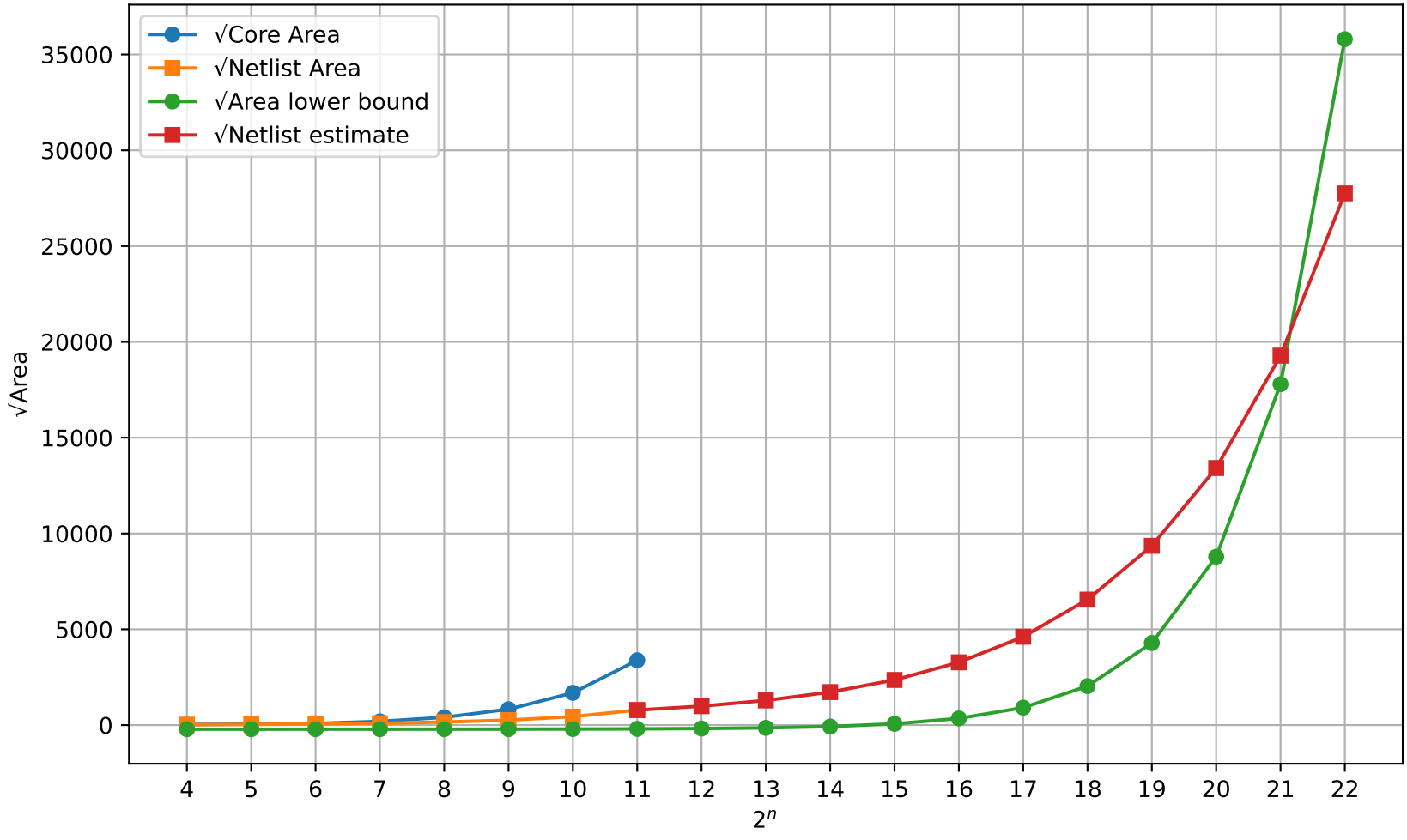
A moment of truth: Canonical partition



Not so gloomy: if inputs were aligned (smaller cut)



Removing assumption on inputs alignment but still canonical



Trugarez!