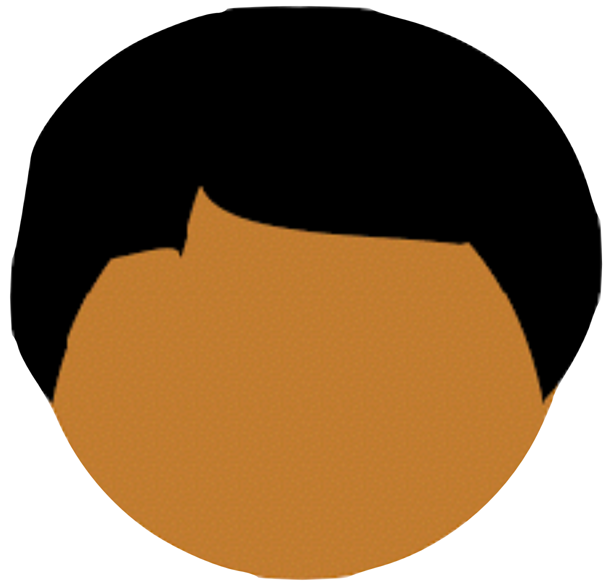
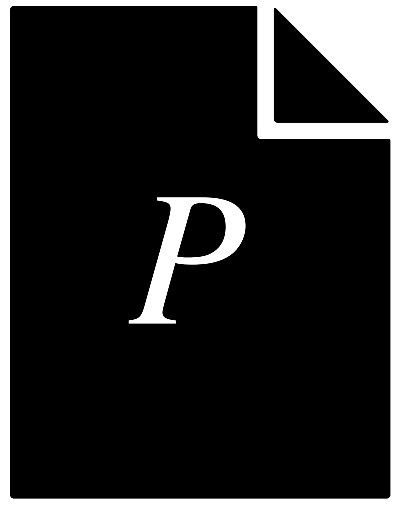
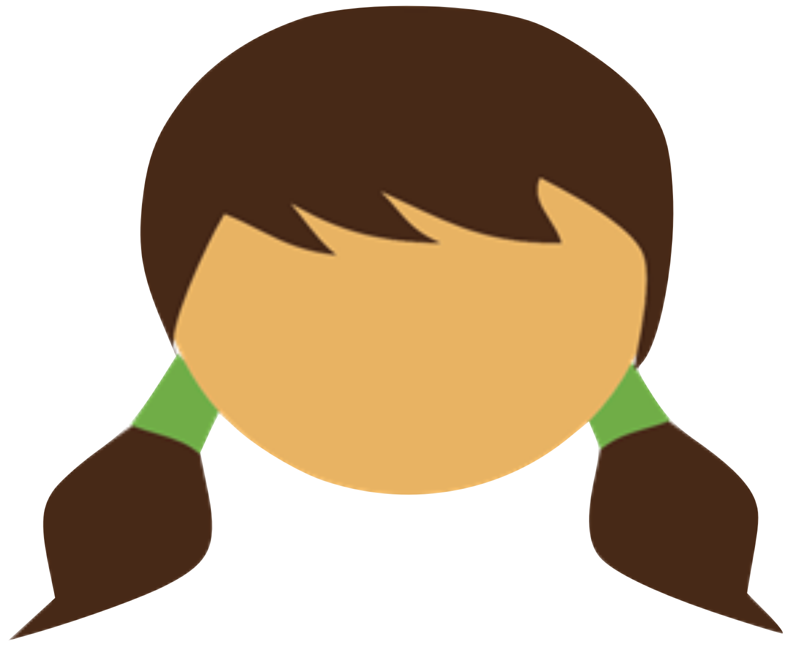
Efficient Arithmetic in Garbled Circuits

David Heath
University of Illinois Urbana-Champaign

Garbled Circuits







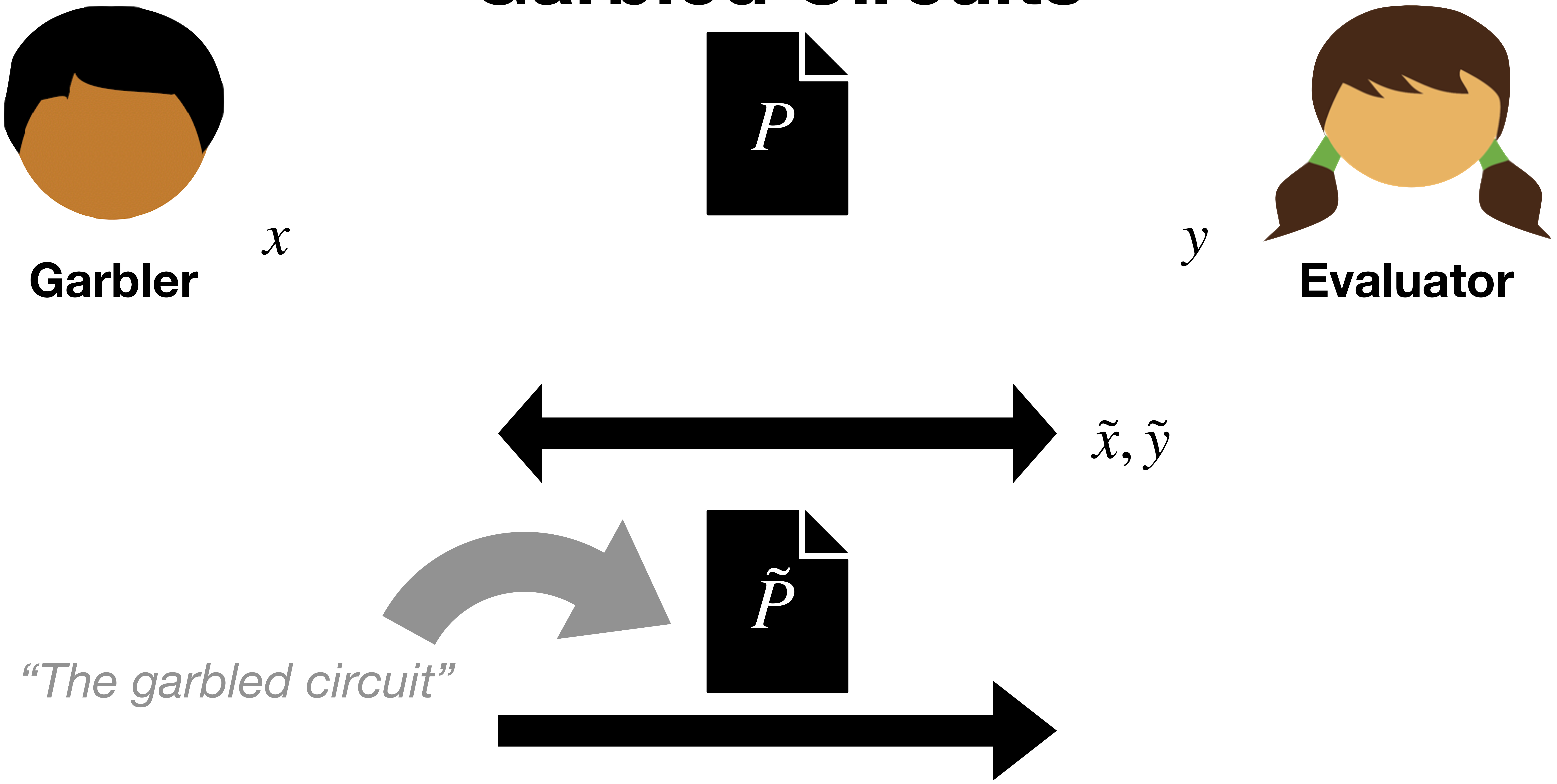
x

Garbler

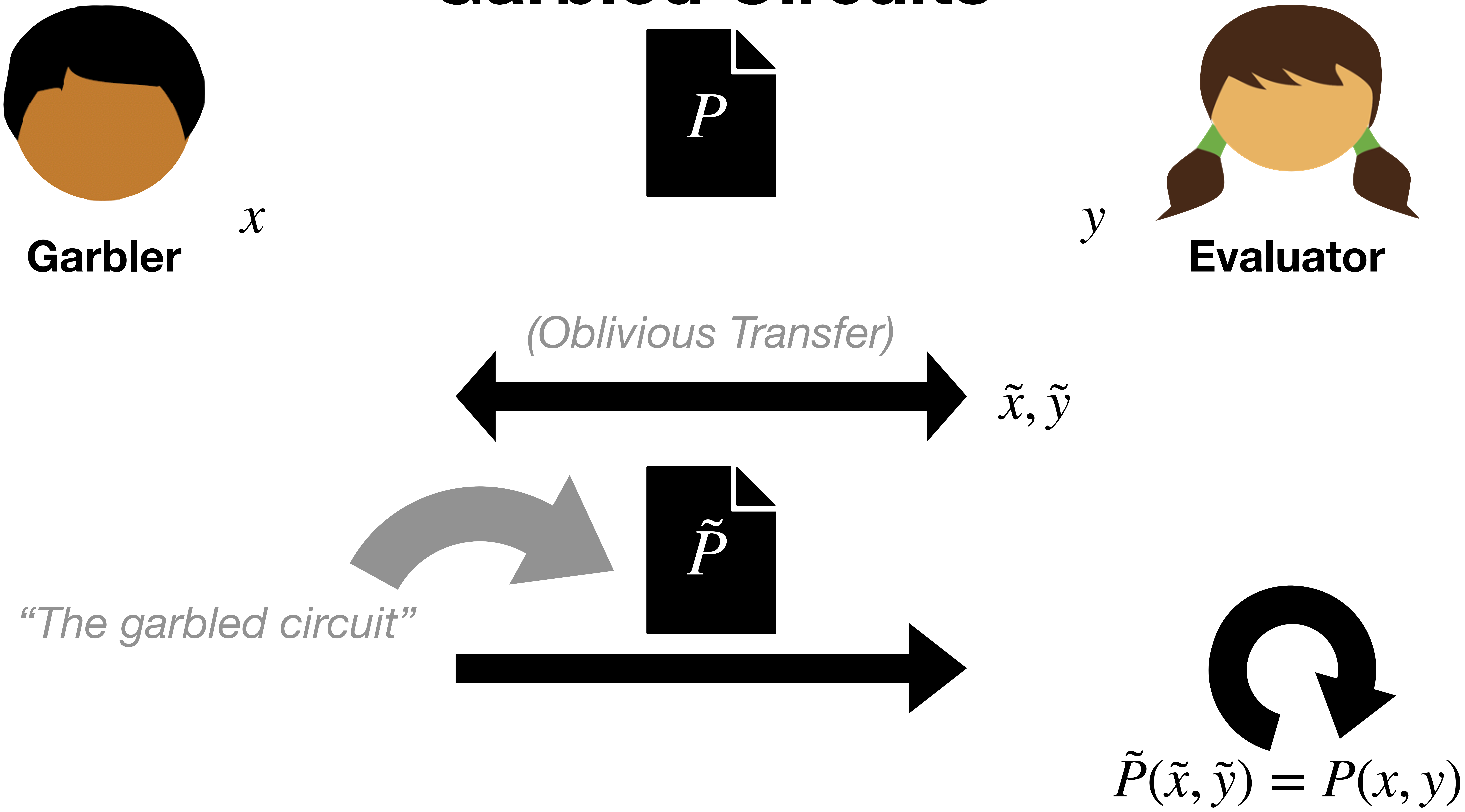
y

Evaluator

Garbled Circuits

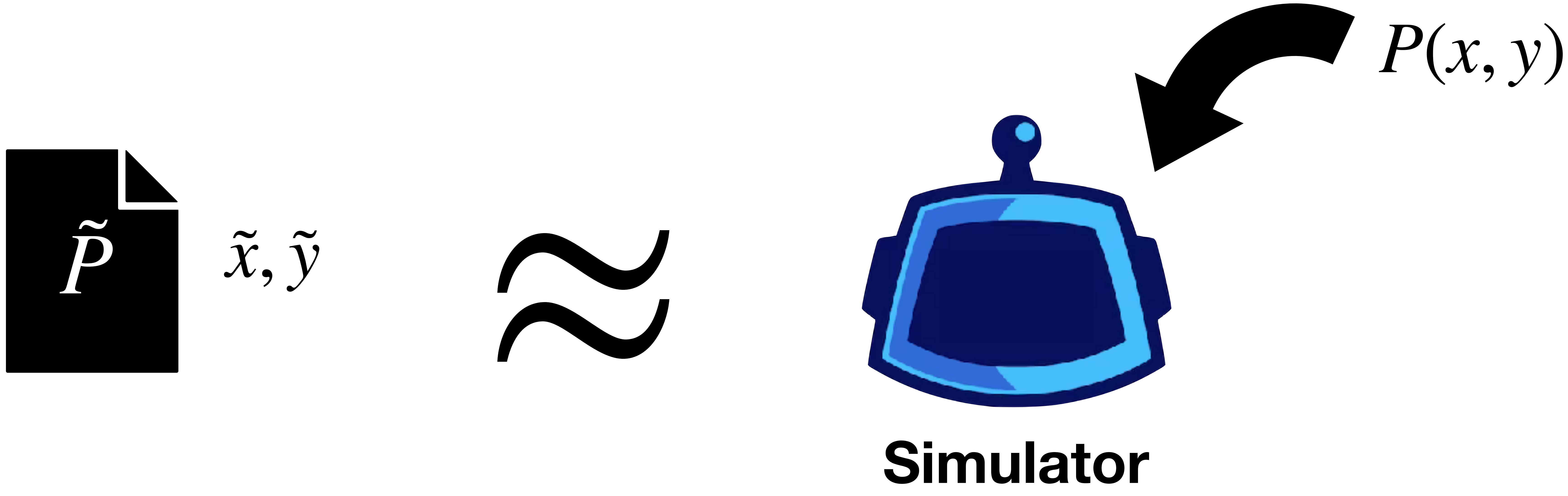


Garbled Circuits

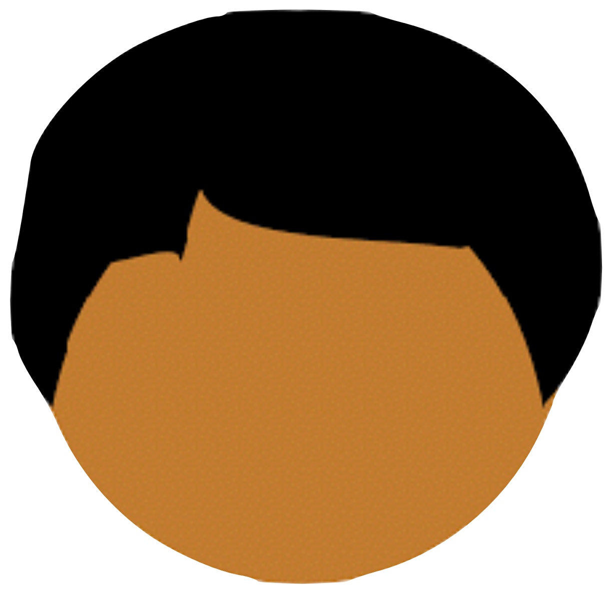


Garbled Circuits

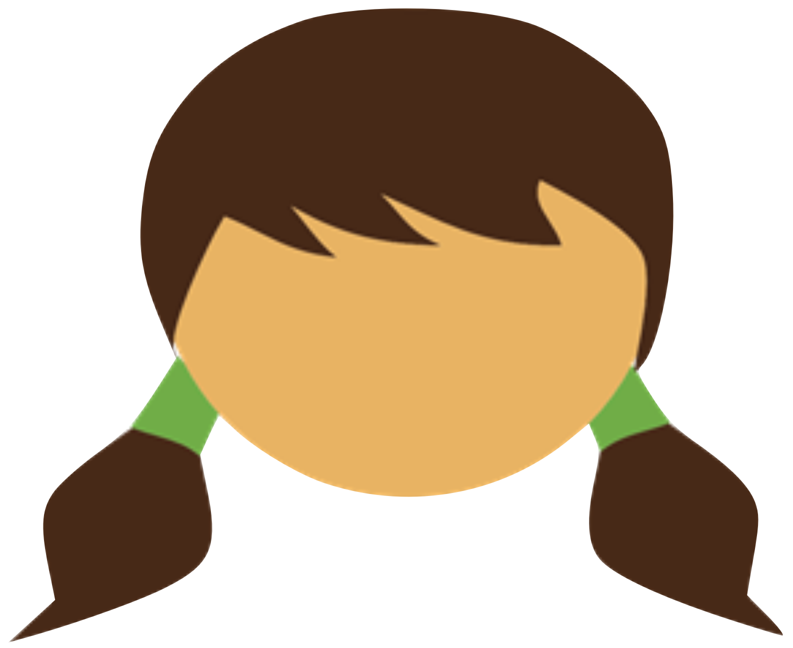
Privacy:



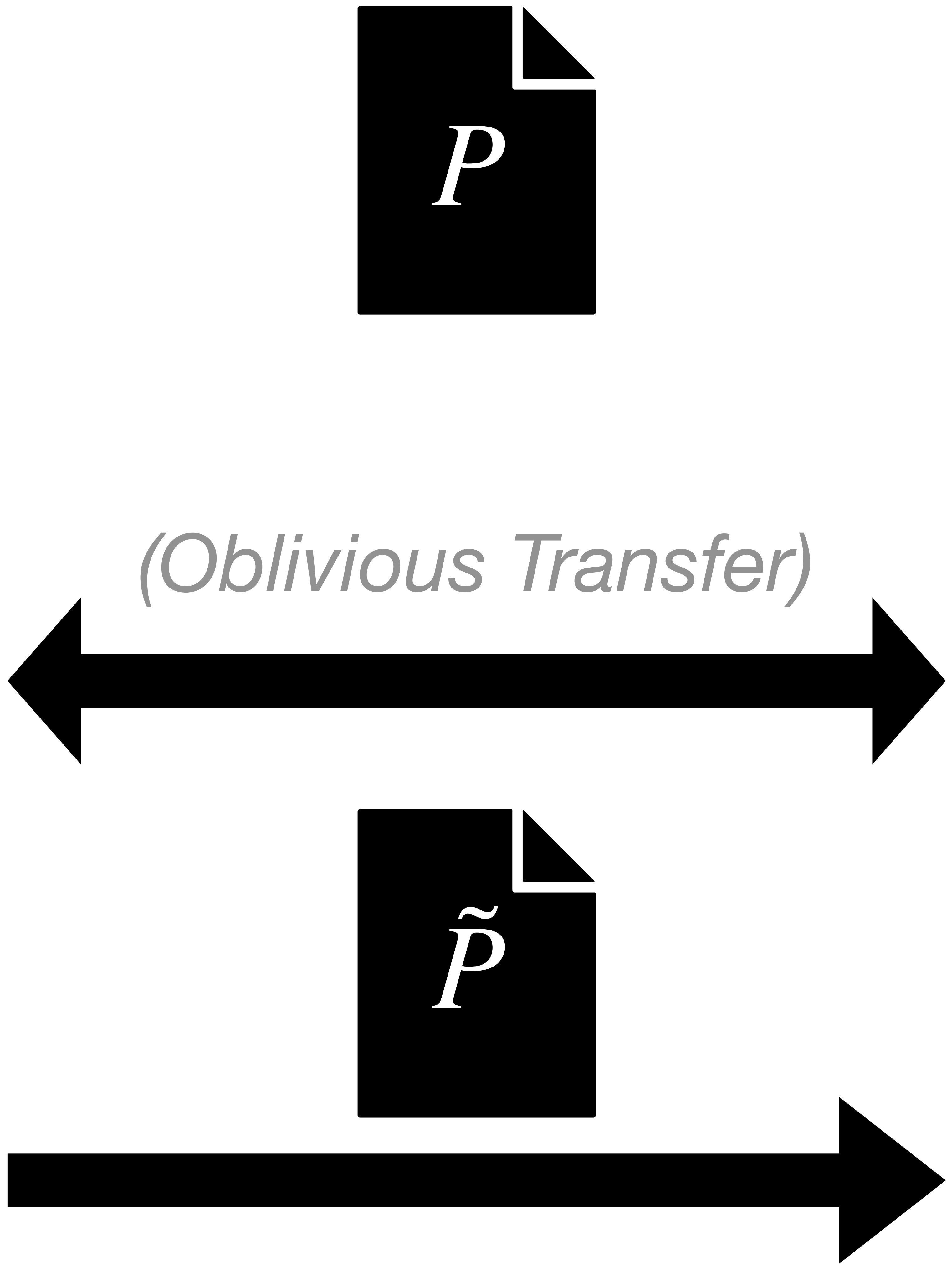
Garbled Circuits



Garbler x



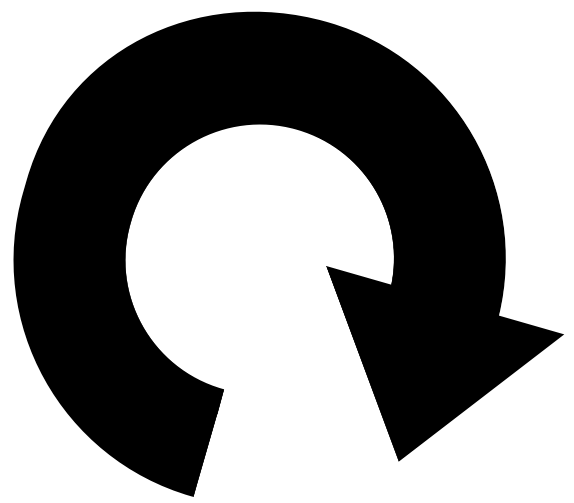
y Evaluator



$\tilde{x}, \tilde{y}$

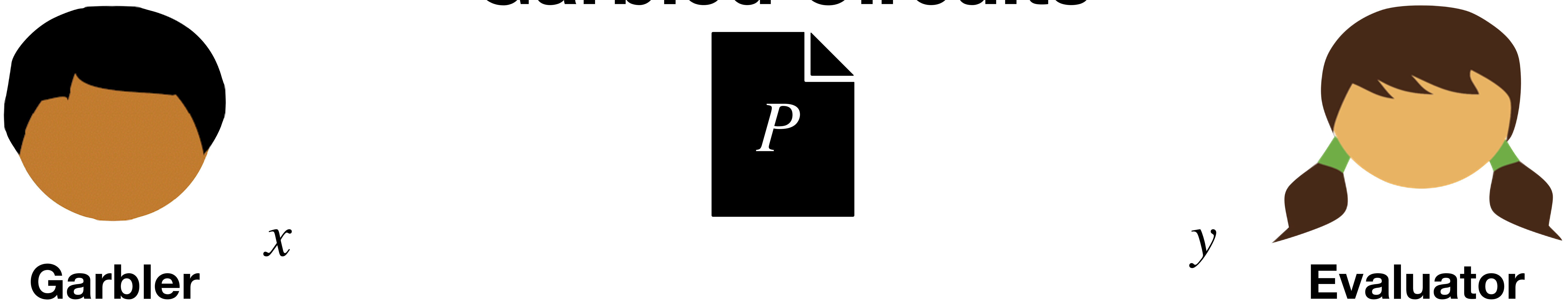
constant round protocols ✓

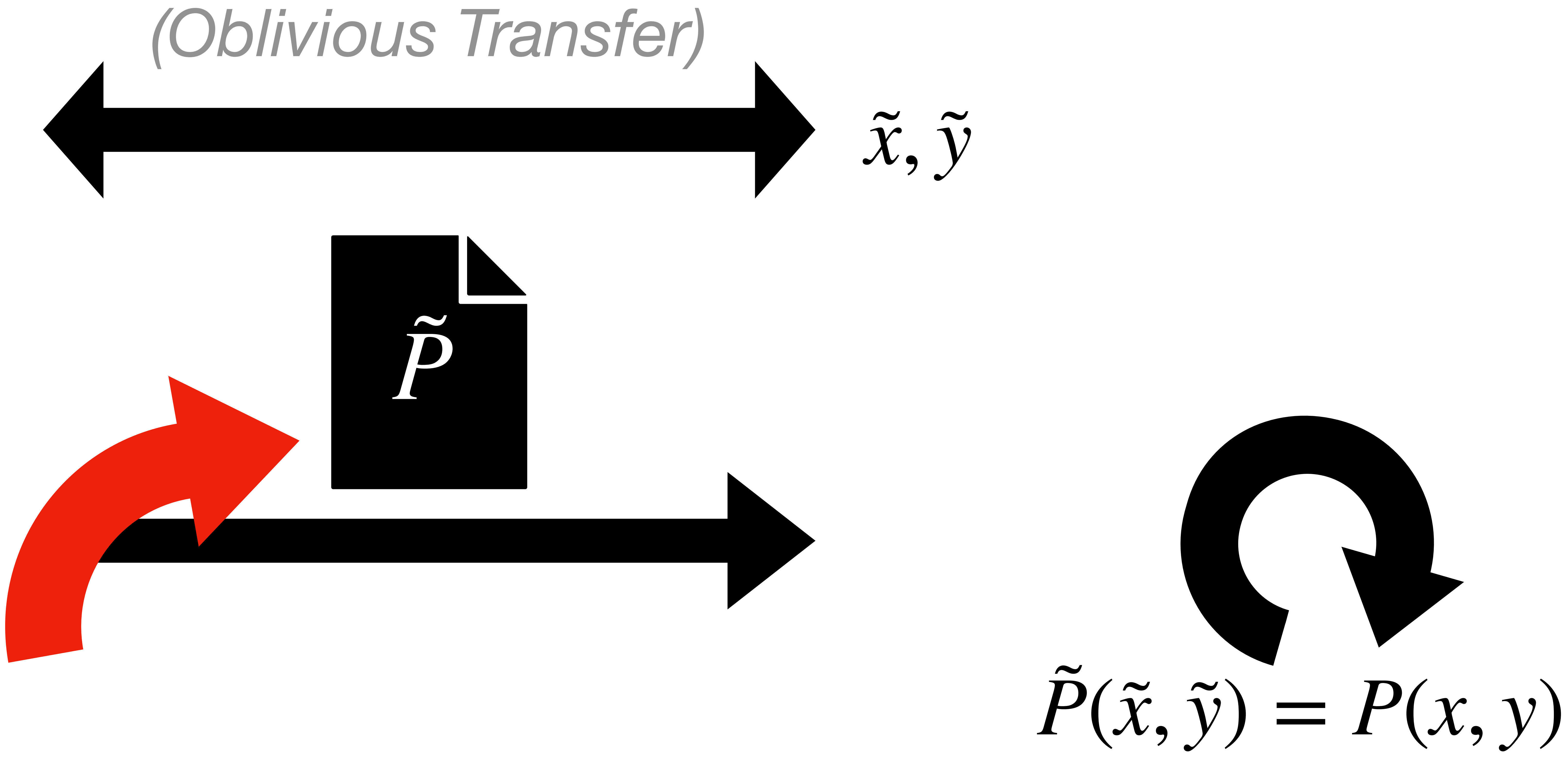
fast, symmetric-key primitives ✓



$$\tilde{P}(\tilde{x}, \tilde{y}) = P(x, y)$$

Garbled Circuits



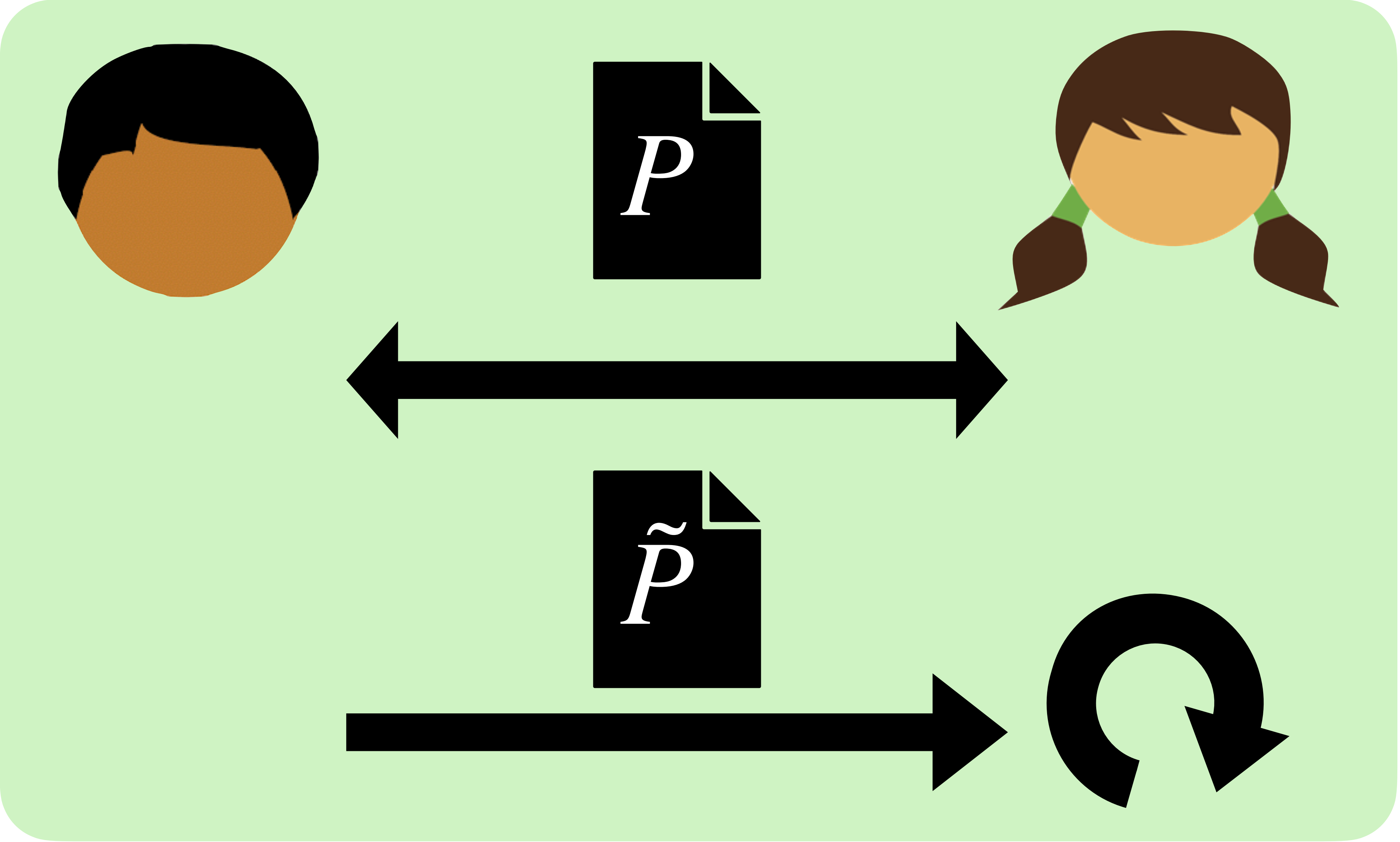


constant round protocols ✓

fast, symmetric-key primitives ✓

high bandwidth consumption ✗

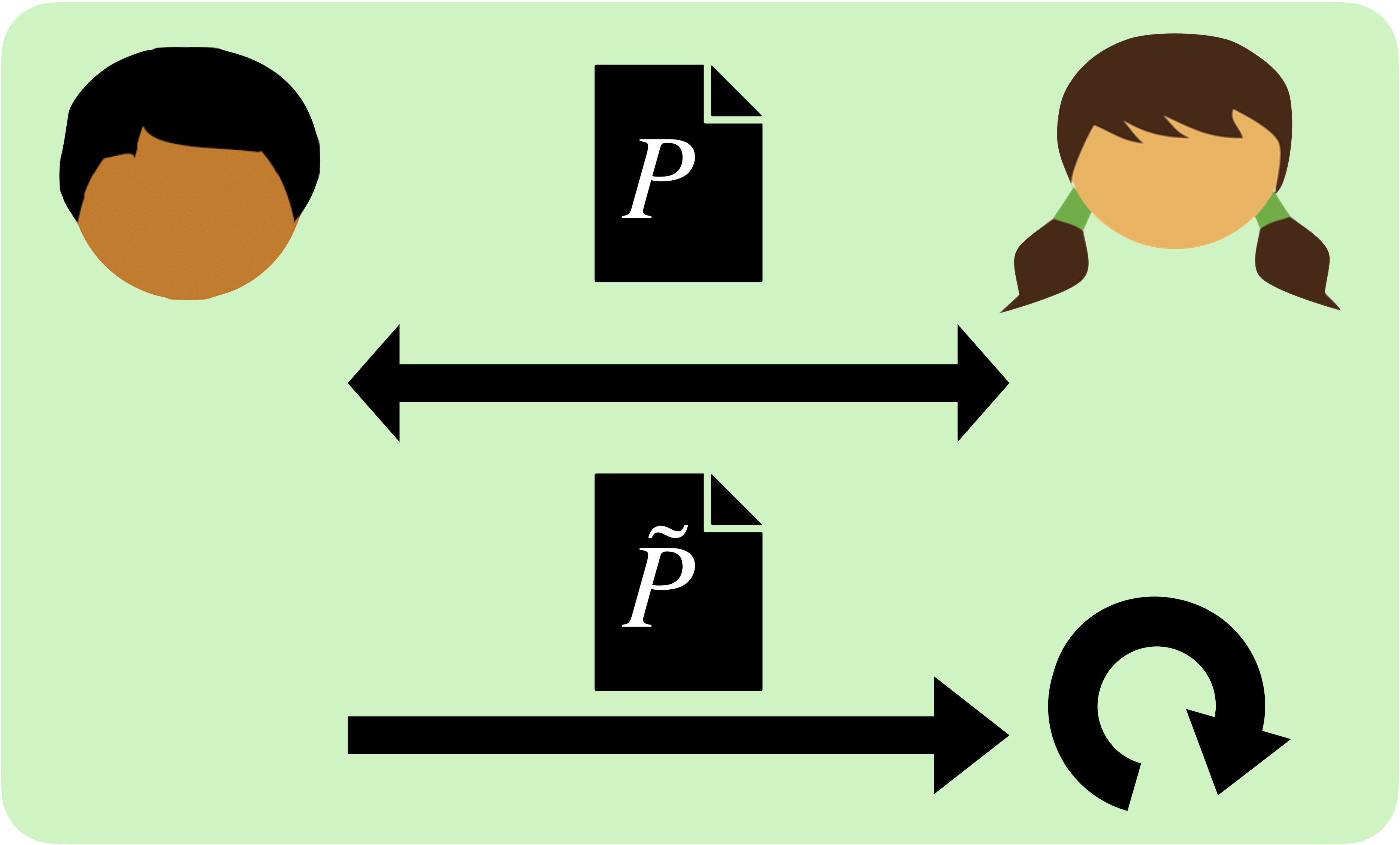
Arithmetic Garbling



Traditionally, P is a Boolean circuit

Garbling of n-gate Boolean circuit has size $O(n \cdot \lambda)$ bits

Arithmetic Garbling



Traditionally, P is a Boolean circuit

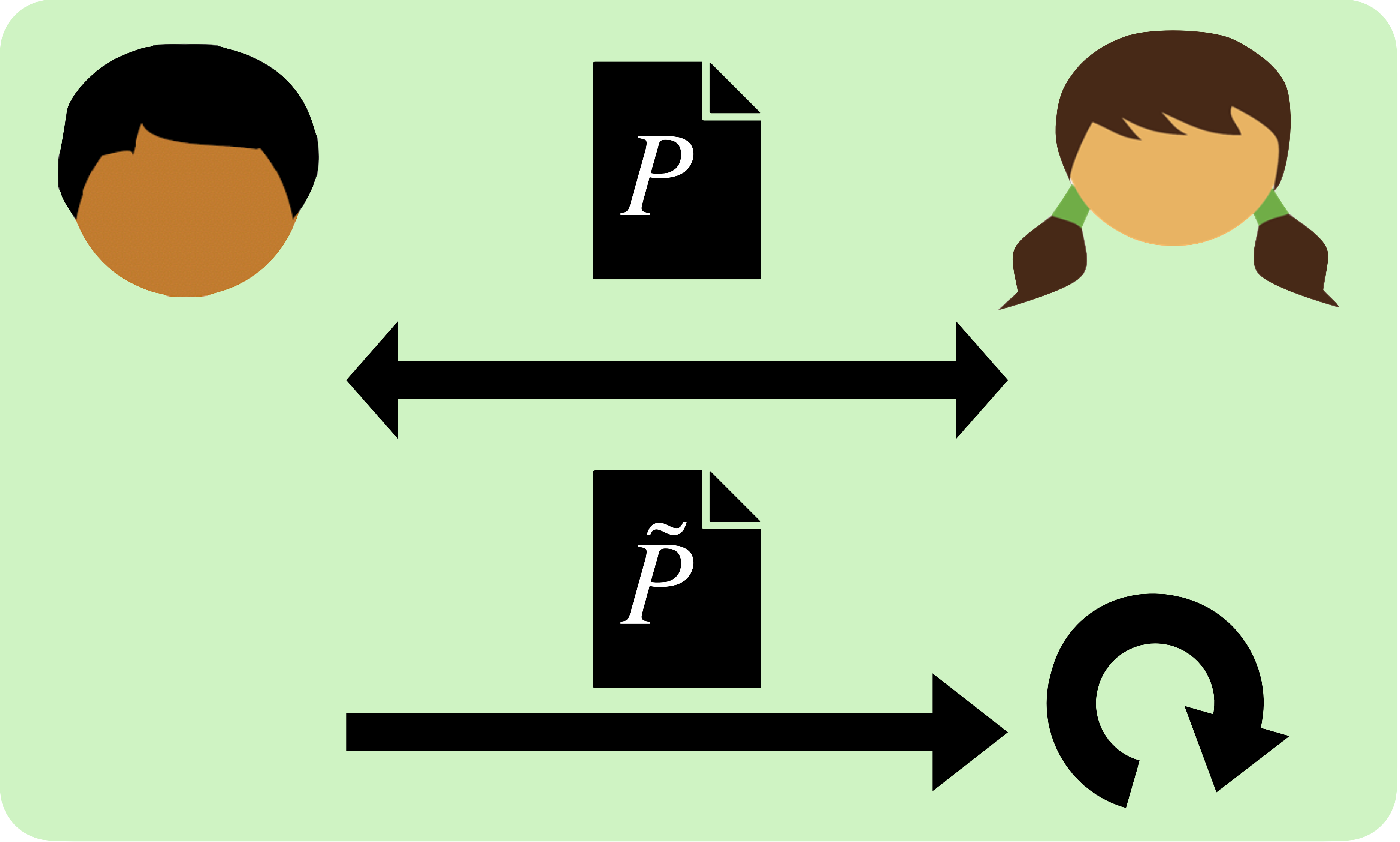
Garbling of n-gate Boolean circuit
has size $O(n \cdot \lambda)$ bits

Desirable to garble **arithmetic circuits**

E.g., privacy-preserving machine learning

Garbling arithmetic gates was a challenge

Arithmetic Garbling

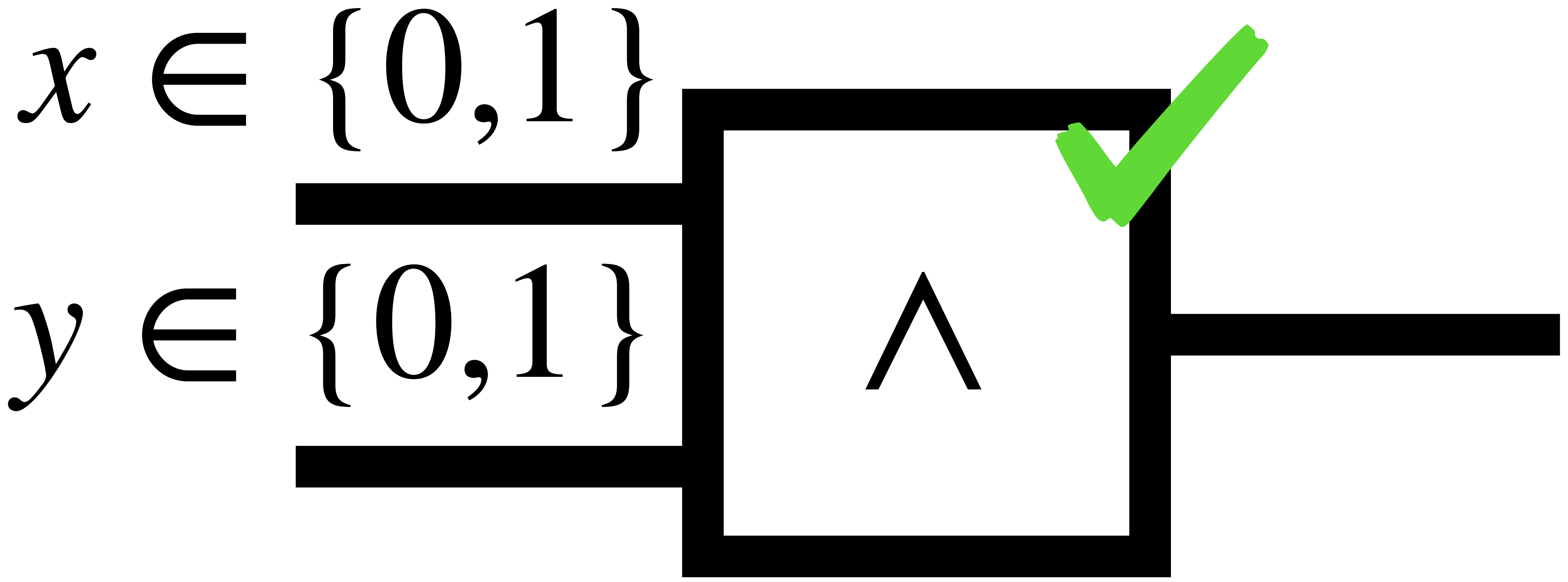


Traditionally, P is a Boolean circuit

Garbling of n-gate Boolean circuit has size $O(n \cdot \lambda)$ bits

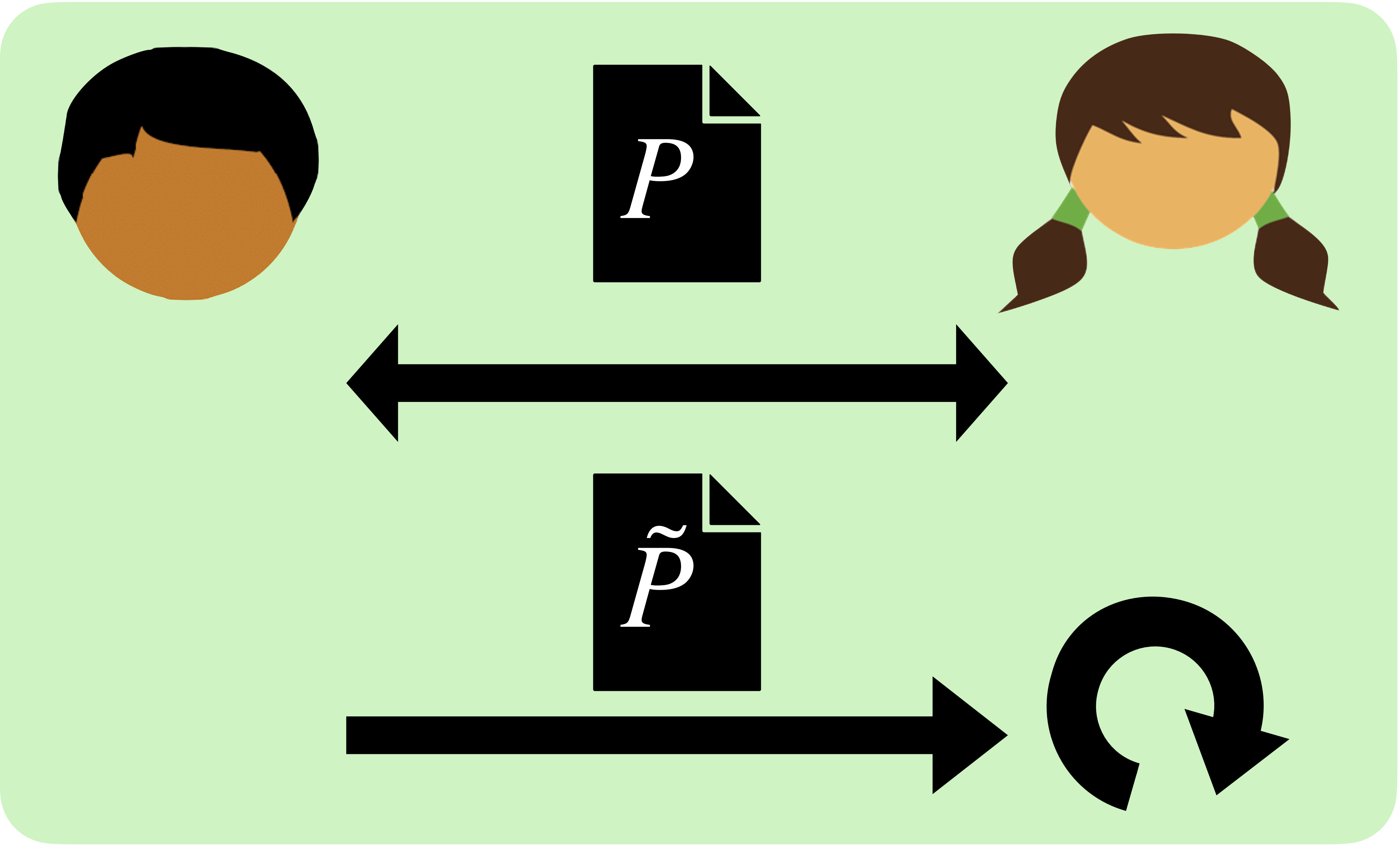
Desirable to garble **arithmetic circuits**

E.g., privacy-preserving machine learning



Garbling arithmetic gates was a challenge

Arithmetic Garbling



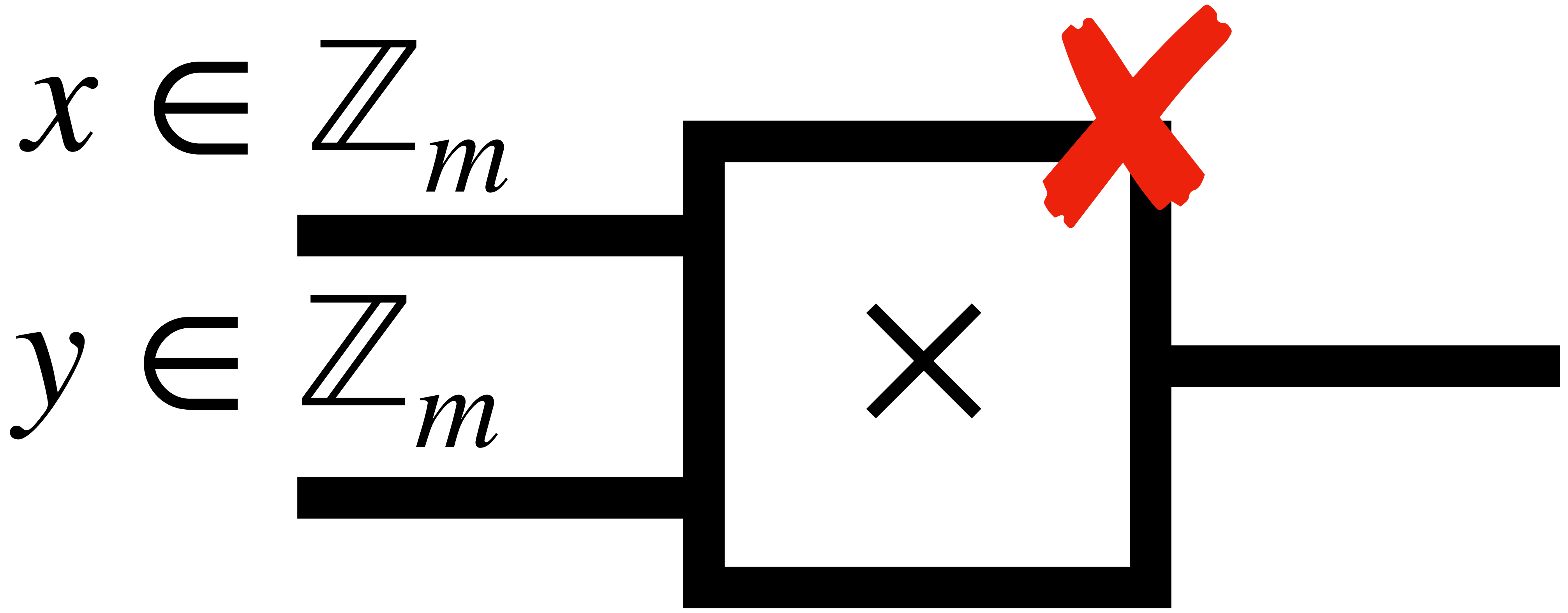
Traditionally, P is a Boolean circuit

Garbling of n-gate Boolean circuit has size $O(n \cdot \lambda)$ bits

Desirable to garble **arithmetic circuits**

E.g., privacy-preserving machine learning

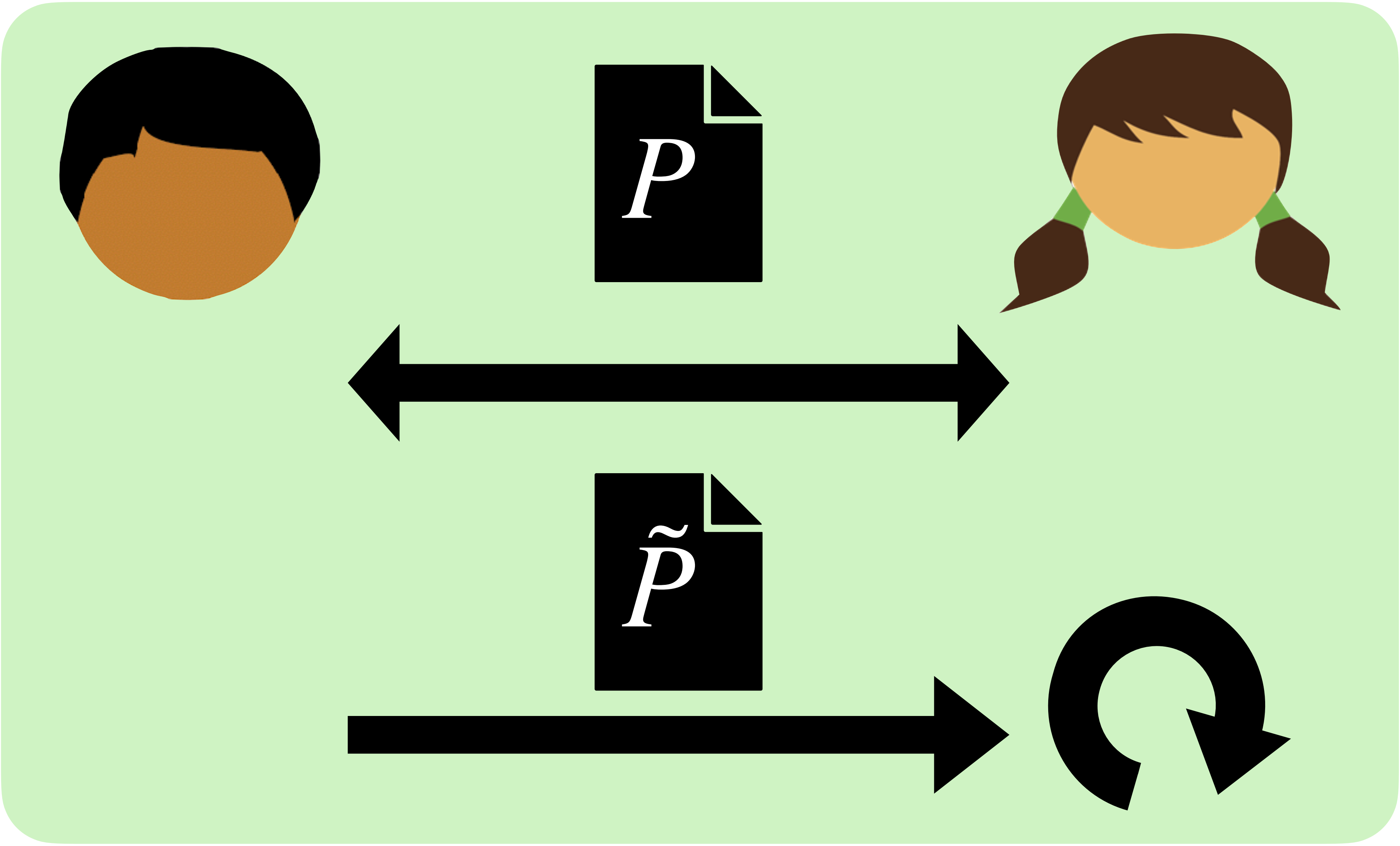
Garbling arithmetic gates was a challenge



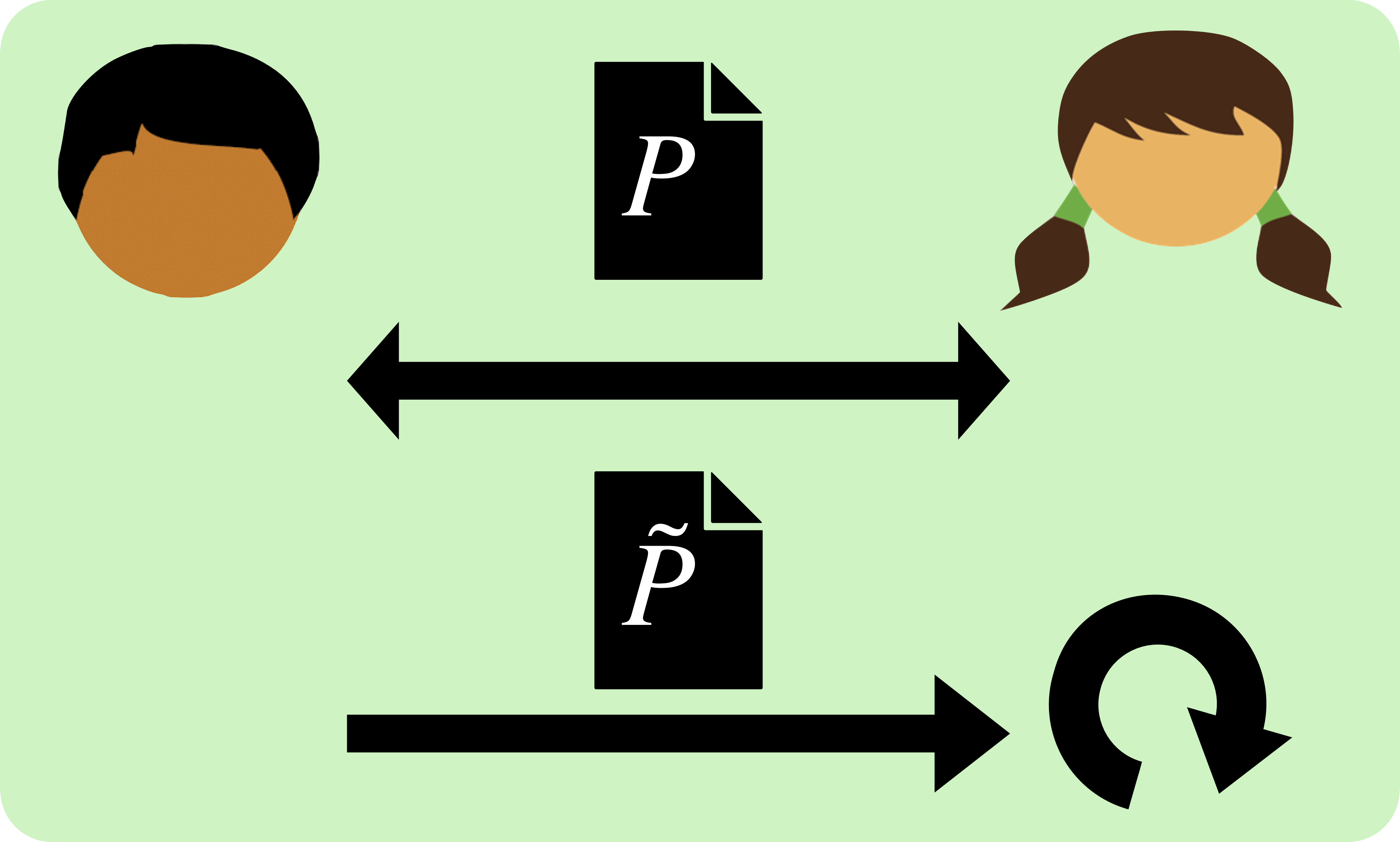
This Work

Consider: P is an n-gate arithmetic circuit over ℓ-bit integers

Goal: Generate small encoding $\tilde{P}$



This Work



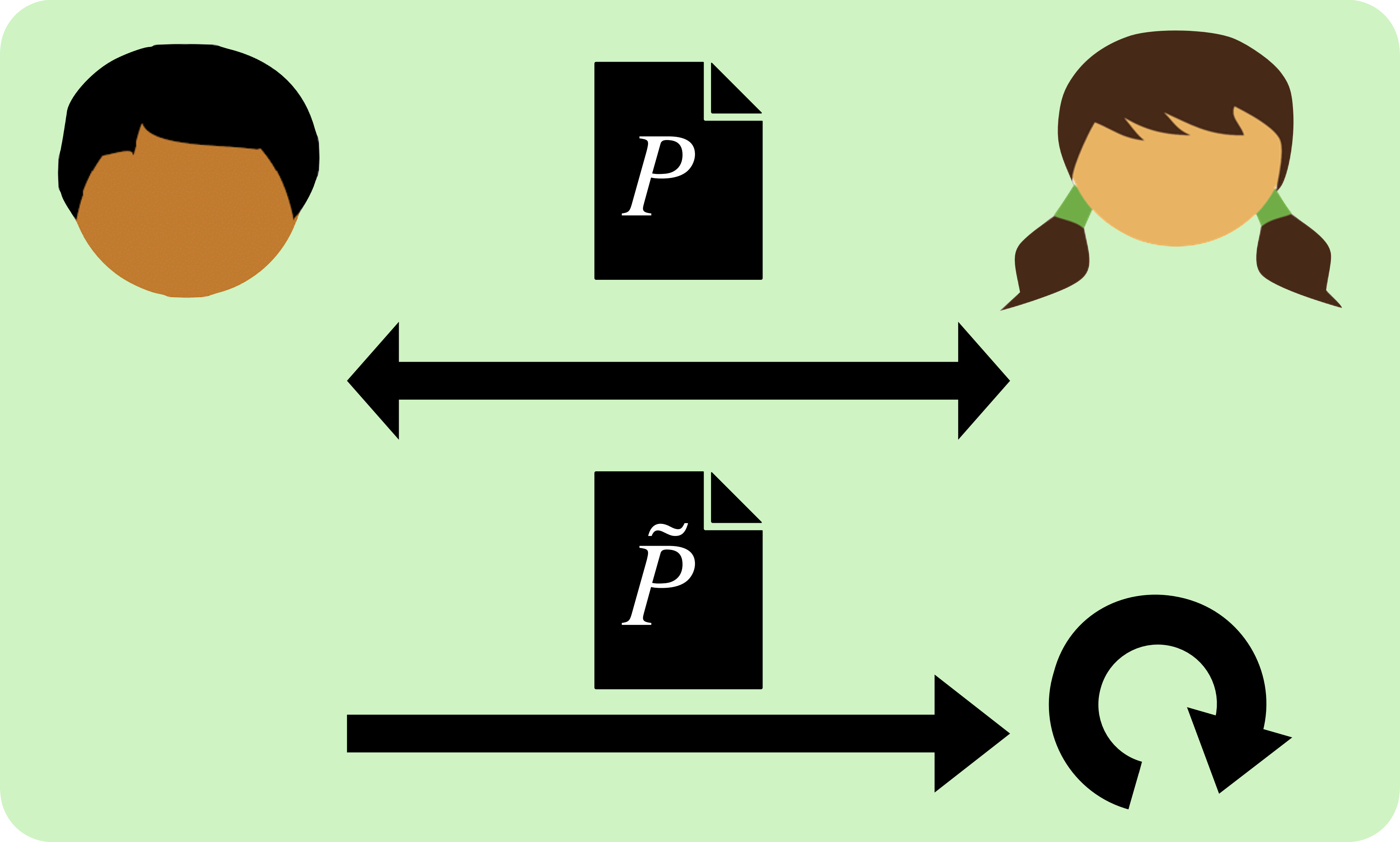
Consider: P is an n-gate arithmetic circuit over ℓ-bit integers

Goal: Generate small encoding $\tilde{P}$

Main Result: $\tilde{P}$ is $O(n \cdot \ell \cdot \lambda)$ bits long

- λ is a computational security parameter (e.g. 128)
- Assumes circular-correlation robust hashes (common in practical symmetric-key GC)

This Work



Consider: P is an n-gate arithmetic circuit over ℓ-bit integers

Goal: Generate small encoding $\tilde{P}$

Main Result: $\tilde{P}$ is $O(n \cdot \ell \cdot \lambda)$ bits long

Surprise Factor: ℓ-bit multiplication at cost $O(\ell \cdot \lambda)$

- λ is a computational security parameter (e.g. 128)
- Assumes circular-correlation robust hashes (common in practical symmetric-key GC)

	$+$	$\times$	Domain
Schoolbook	$O(\ell \cdot \lambda)$	$O(\ell^2 \cdot \lambda)$	
Karatsuba	$O(\ell \cdot \lambda)$	$O(\ell^{1.585} \cdot \lambda)$	

	+	×	Domain
Schoolbook	$O(\ell \cdot \lambda)$	$O(\ell^2 \cdot \lambda)$	
Karatsuba	$O(\ell \cdot \lambda)$	$O(\ell^{1.585} \cdot \lambda)$	
[BMR16]	0	$O(2^\ell \cdot \lambda)$	

	$+$	$\times$	Domain
Schoolbook	$O(\ell \cdot \lambda)$	$O(\ell^2 \cdot \lambda)$	any $\mathbb{Z}_m$
Karatsuba	$O(\ell \cdot \lambda)$	$O(\ell^{1.585} \cdot \lambda)$	any $\mathbb{Z}_m$
[BMR16]	0	$O(2^\ell \cdot \lambda)$	prime $\mathbb{Z}_p$

	$+$	$\times$	Domain
Schoolbook	$O(\ell \cdot \lambda)$	$O(\ell^2 \cdot \lambda)$	any $\mathbb{Z}_m$
Karatsuba	$O(\ell \cdot \lambda)$	$O(\ell^{1.585} \cdot \lambda)$	any $\mathbb{Z}_m$
[BMR16]	0	$O(2^\ell \cdot \lambda)$	prime $\mathbb{Z}_p$
[BMR16] + CRT	0	$O((\ell^2/\log \ell) \cdot \lambda)$	CRT friendly $\mathbb{Z}_N$

	$+$	$\times$	Domain
Schoolbook	$O(\ell \cdot \lambda)$	$O(\ell^2 \cdot \lambda)$	any $\mathbb{Z}_m$
Karatsuba	$O(\ell \cdot \lambda)$	$O(\ell^{1.585} \cdot \lambda)$	any $\mathbb{Z}_m$
[BMR16]	0	$O(2^\ell \cdot \lambda)$	prime $\mathbb{Z}_p$
[BMR16] + CRT	0	$O((\ell^2/\log \ell) \cdot \lambda)$	CRT friendly $\mathbb{Z}_N$
Boolean CRT	$O(\ell \cdot \lambda)$	$O(\ell \log \ell \cdot \lambda)$	CRT friendly $\mathbb{Z}_N$

	$+$	$\times$	Domain
Schoolbook	$O(\ell \cdot \lambda)$	$O(\ell^2 \cdot \lambda)$	any $\mathbb{Z}_m$
Karatsuba	$O(\ell \cdot \lambda)$	$O(\ell^{1.585} \cdot \lambda)$	any $\mathbb{Z}_m$
[BMR16]	0	$O(2^\ell \cdot \lambda)$	prime $\mathbb{Z}_p$
[BMR16] + CRT	0	$O((\ell^2/\log \ell) \cdot \lambda)$	CRT friendly $\mathbb{Z}_N$
Boolean CRT	$O(\ell \cdot \lambda)$	$O(\ell \log \ell \cdot \lambda)$	CRT friendly $\mathbb{Z}_N$

[AIK11] and [BLL+23] also garble arithmetic, but require public-key cryptography

	$+$	$\times$	Domain
This Work	$O(\ell \cdot \lambda)$	$O(\ell \cdot \lambda)$	any $\mathbb{Z}_m$
Schoolbook	$O(\ell \cdot \lambda)$	$O(\ell^2 \cdot \lambda)$	any $\mathbb{Z}_m$
Karatsuba	$O(\ell \cdot \lambda)$	$O(\ell^{1.585} \cdot \lambda)$	any $\mathbb{Z}_m$
[BMR16]	0	$O(2^\ell \cdot \lambda)$	prime $\mathbb{Z}_p$
[BMR16] + CRT	0	$O((\ell^2/\log \ell) \cdot \lambda)$	CRT friendly $\mathbb{Z}_N$
Boolean CRT	$O(\ell \cdot \lambda)$	$O(\ell \log \ell \cdot \lambda)$	CRT friendly $\mathbb{Z}_N$

[AIK11] and [BLL+23] also garble arithmetic, but require public-key cryptography

This Work

Goal

Integer
Arithmetic

This Work

Goal

Long Integer Arithmetic

Short Integer Arithmetic

Chinese Remainder Theorem

This Work

Goal

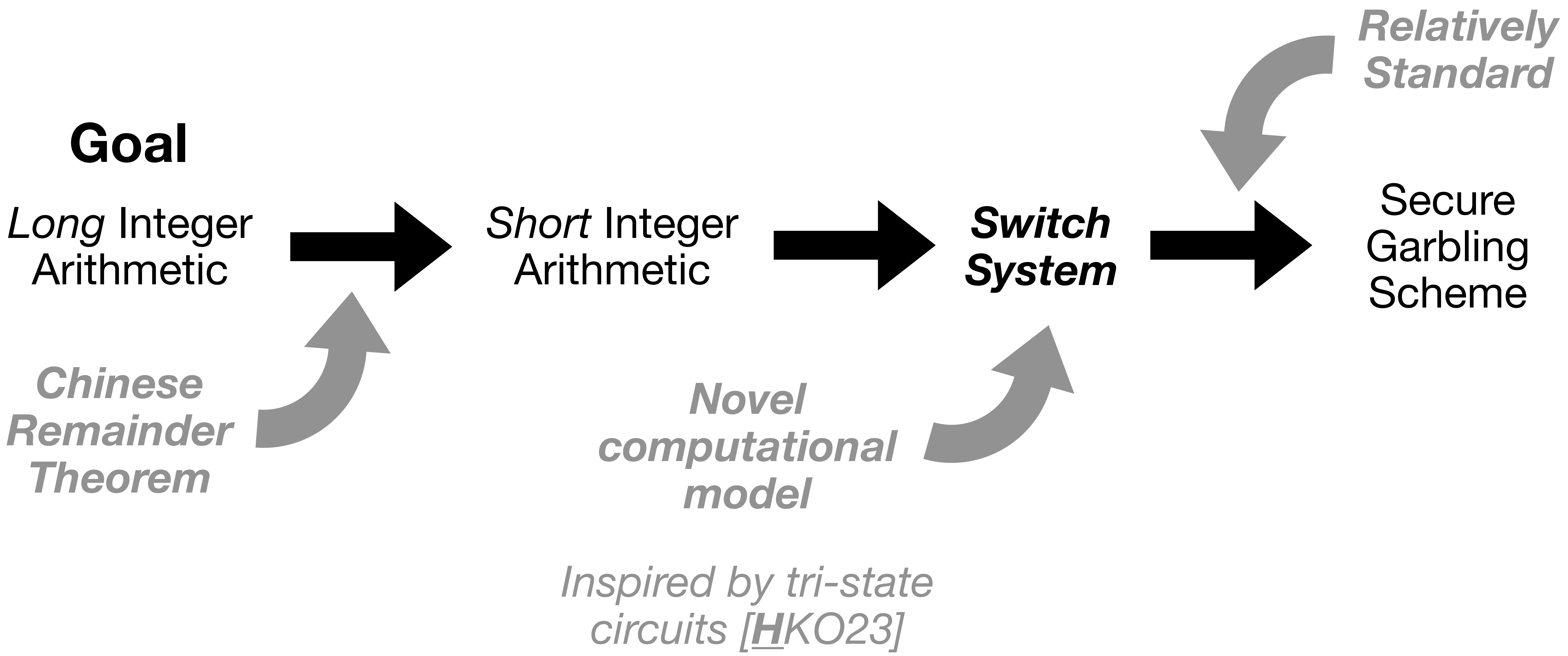
Long Integer Arithmetic → *Short* Integer Arithmetic → ***Switch System***

Chinese Remainder Theorem

Novel computational model

*Inspired by tri-state circuits [**H**KO23]*

This Work


Goal
Long Integer Arithmetic
Chinese Remainder Theorem
Short Integer Arithmetic
Novel computational model
Inspired by tri-state circuits [**H**KO23]
Switch System
Relatively Standard
Secure Garbling Scheme

This Work

Goal

Long Integer Arithmetic

Chinese Remainder Theorem

Today's Focus

Short Integer Arithmetic

Switch System

Novel computational model

*Inspired by tri-state circuits [<u>**H**</u>KO23]*

Relatively Standard

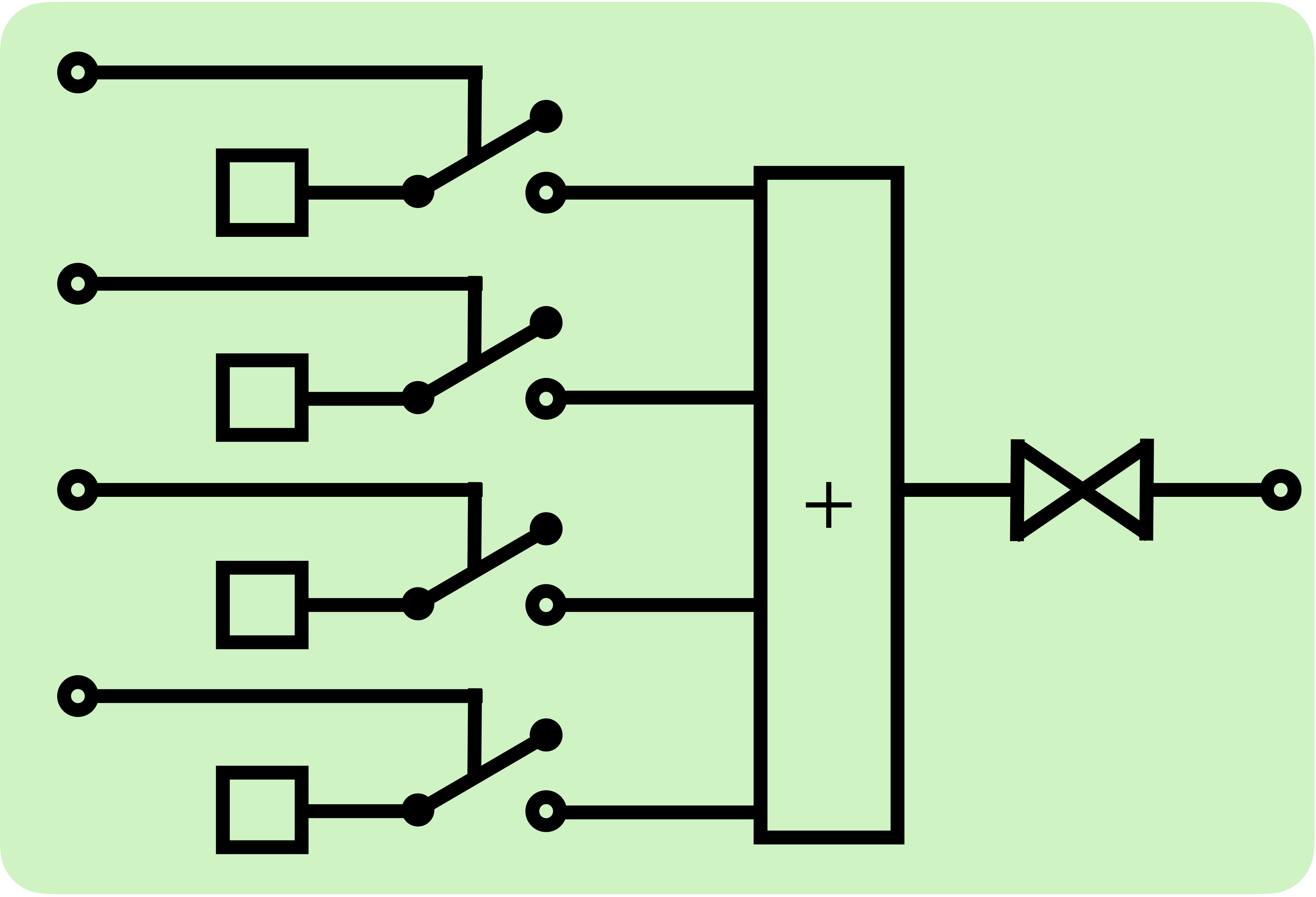
Secure Garbling Scheme

Switch Systems

Alternative to Boolean Circuits

Relatively Straightforward to Garble

Models Computation as a **Constraint System** that the evaluator will solve



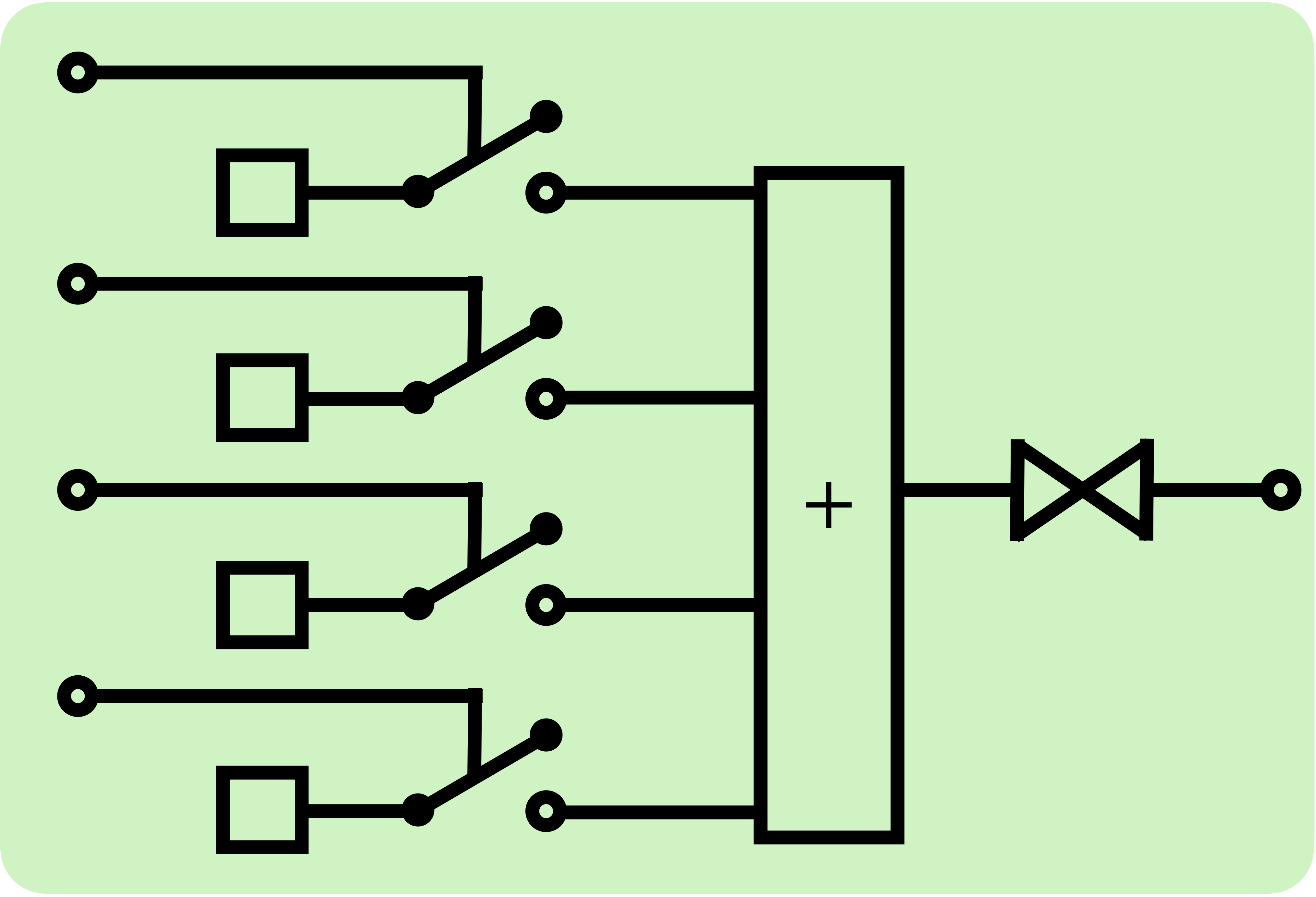
See Paper

Switch Systems

Alternative to Boolean Circuits

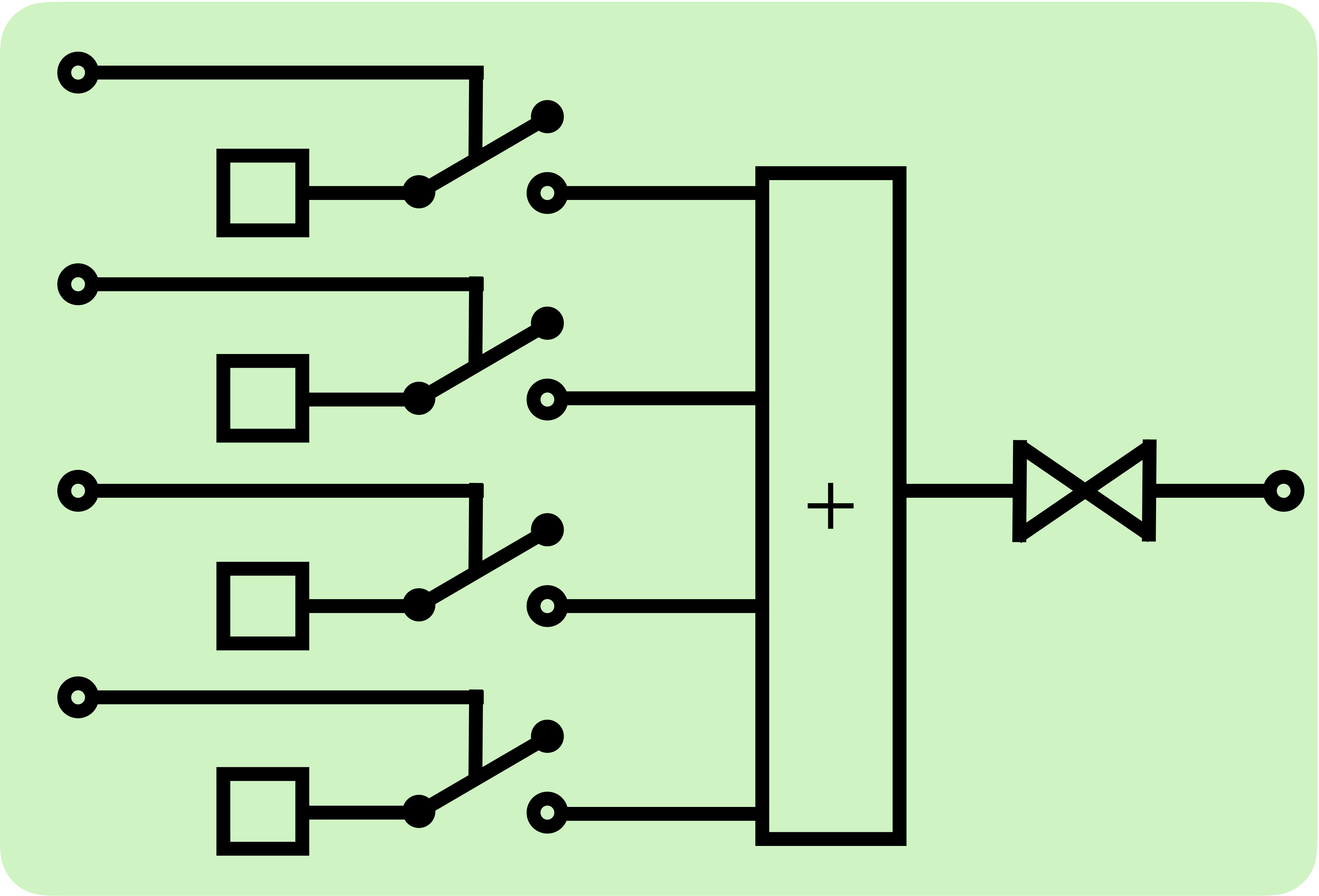
Relatively Straightforward to Garble

Models Computation as a **Constraint System** that the evaluator will solve



See Paper

Switch Systems



Alternative to Boolean Circuits

Relatively Straightforward to Garble

Models Computation as a **Constraint System** that the evaluator will solve

See Paper

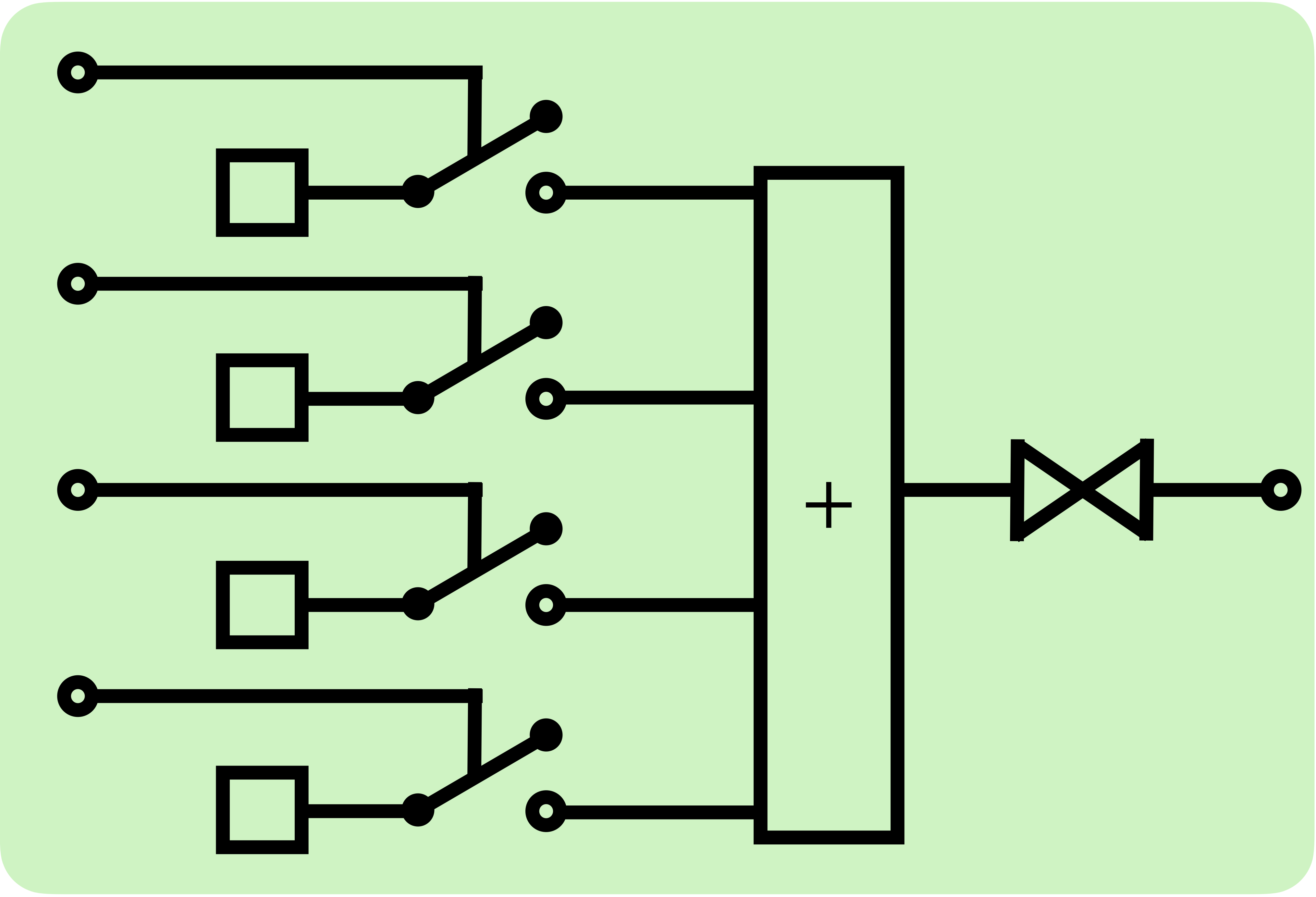
Switch Systems

Alternative to Boolean Circuits

Relatively Straightforward to Garble

Models Computation as a **Constraint System** that the evaluator will solve

Captures much of the state-of-the-art in symmetric-key garbling

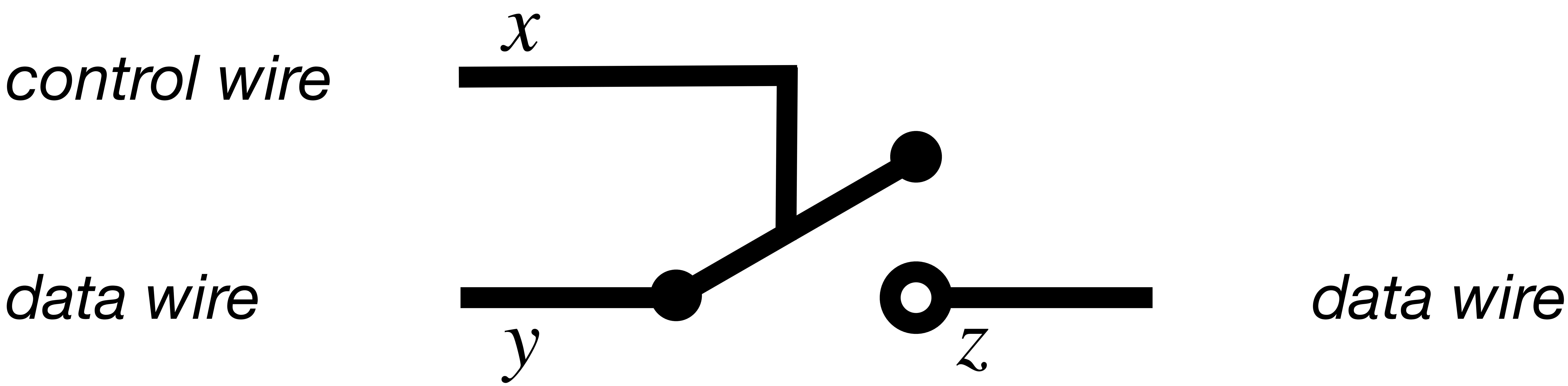


See Paper

Free XOR, Half-AND Gates, Garbled RAM, One-Hot Garbling, ***Arithmetic Computations***

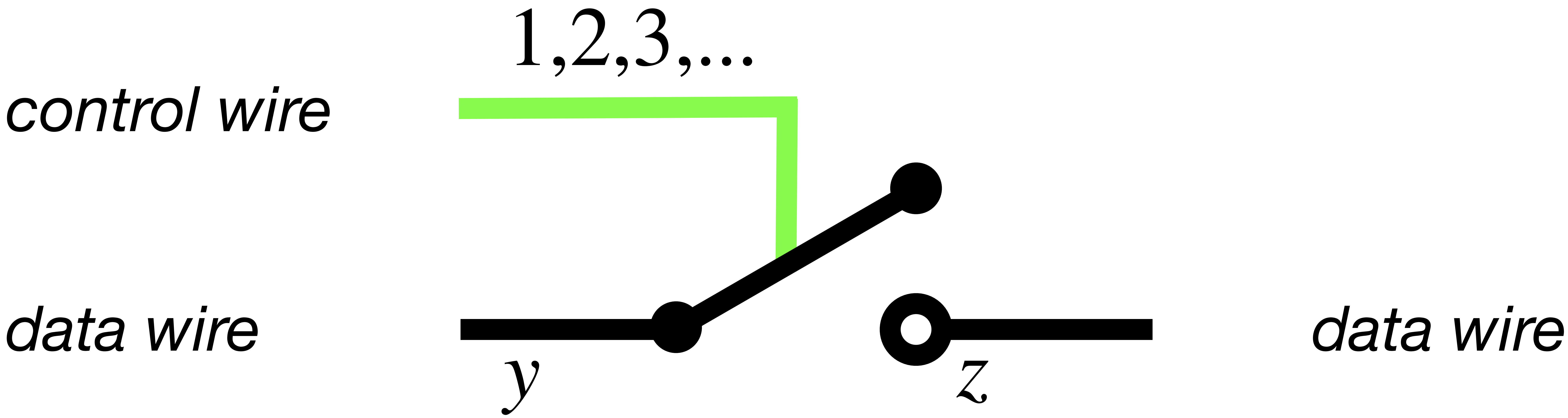
Switch Systems

Switch



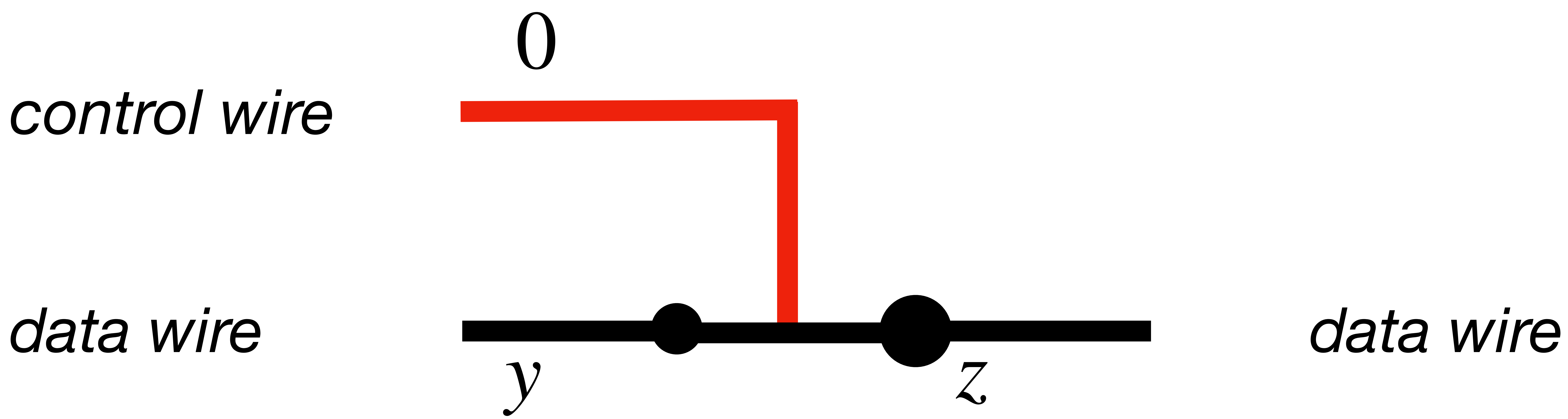
Switch Systems

Switch



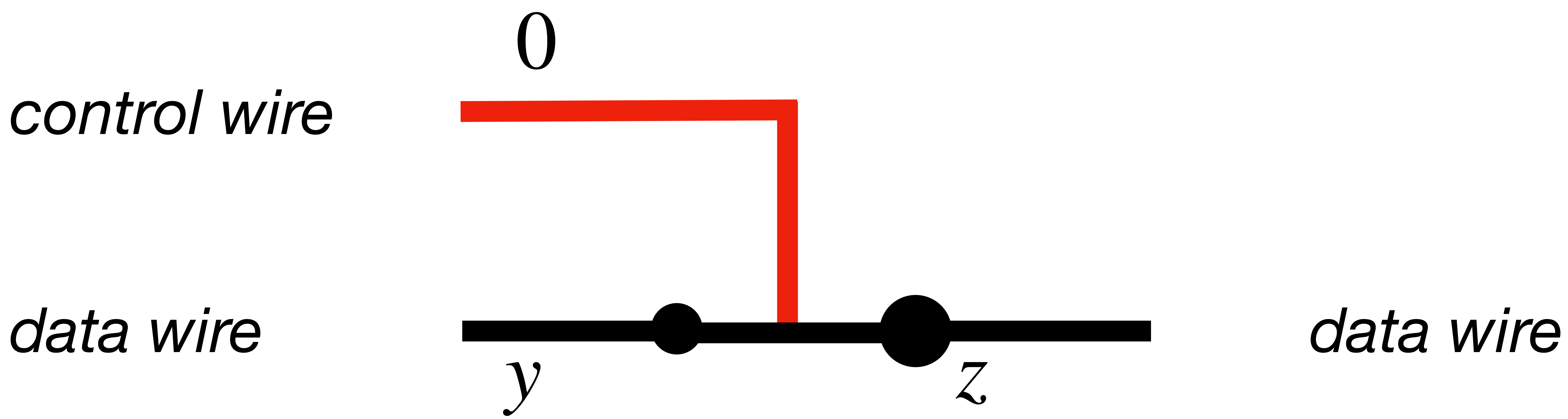
Switch Systems

Switch



Switch Systems

Switch



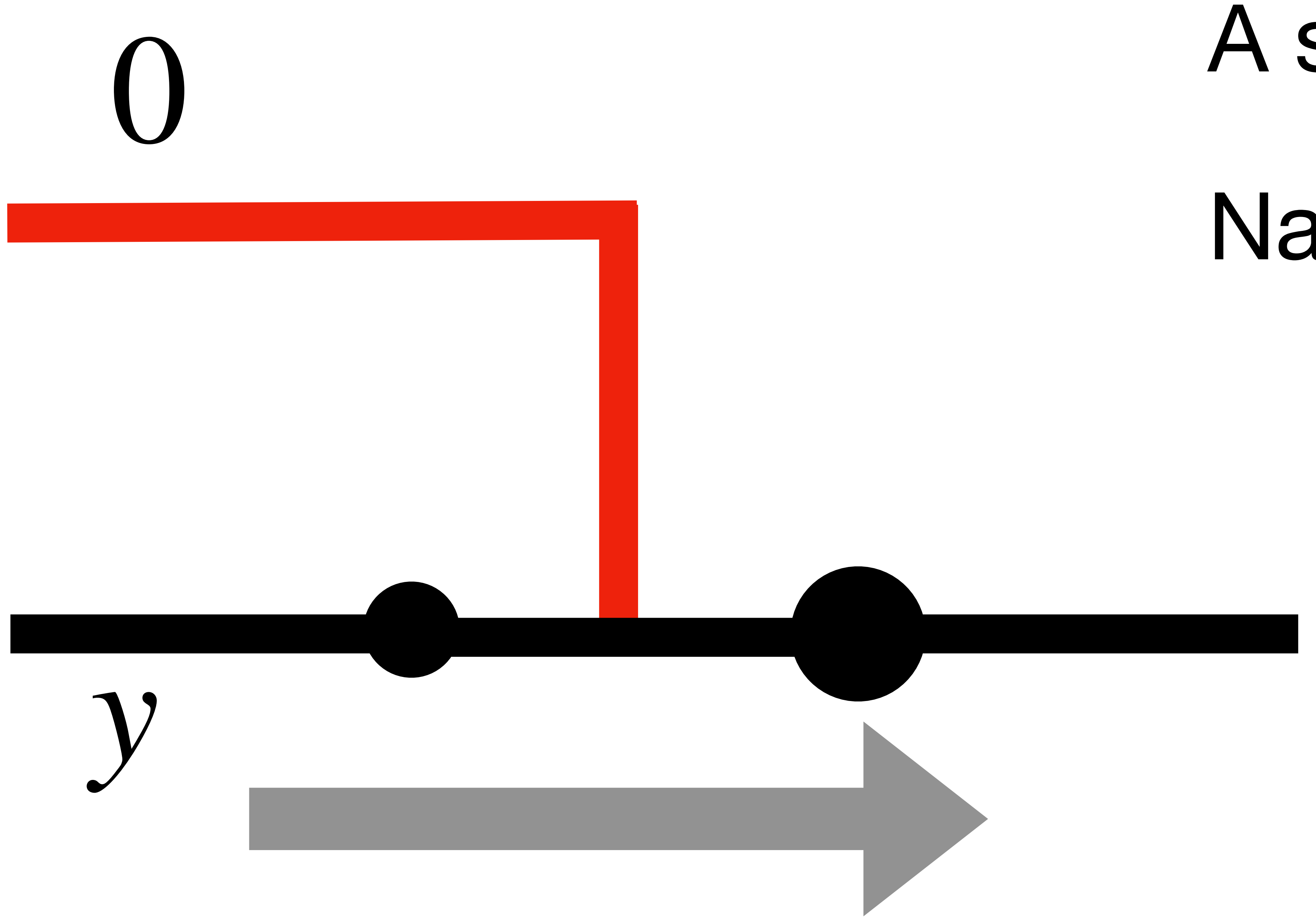
$$x = 0 \implies y = z$$

Switch Systems

Switch

A switch enforces a *constraint*

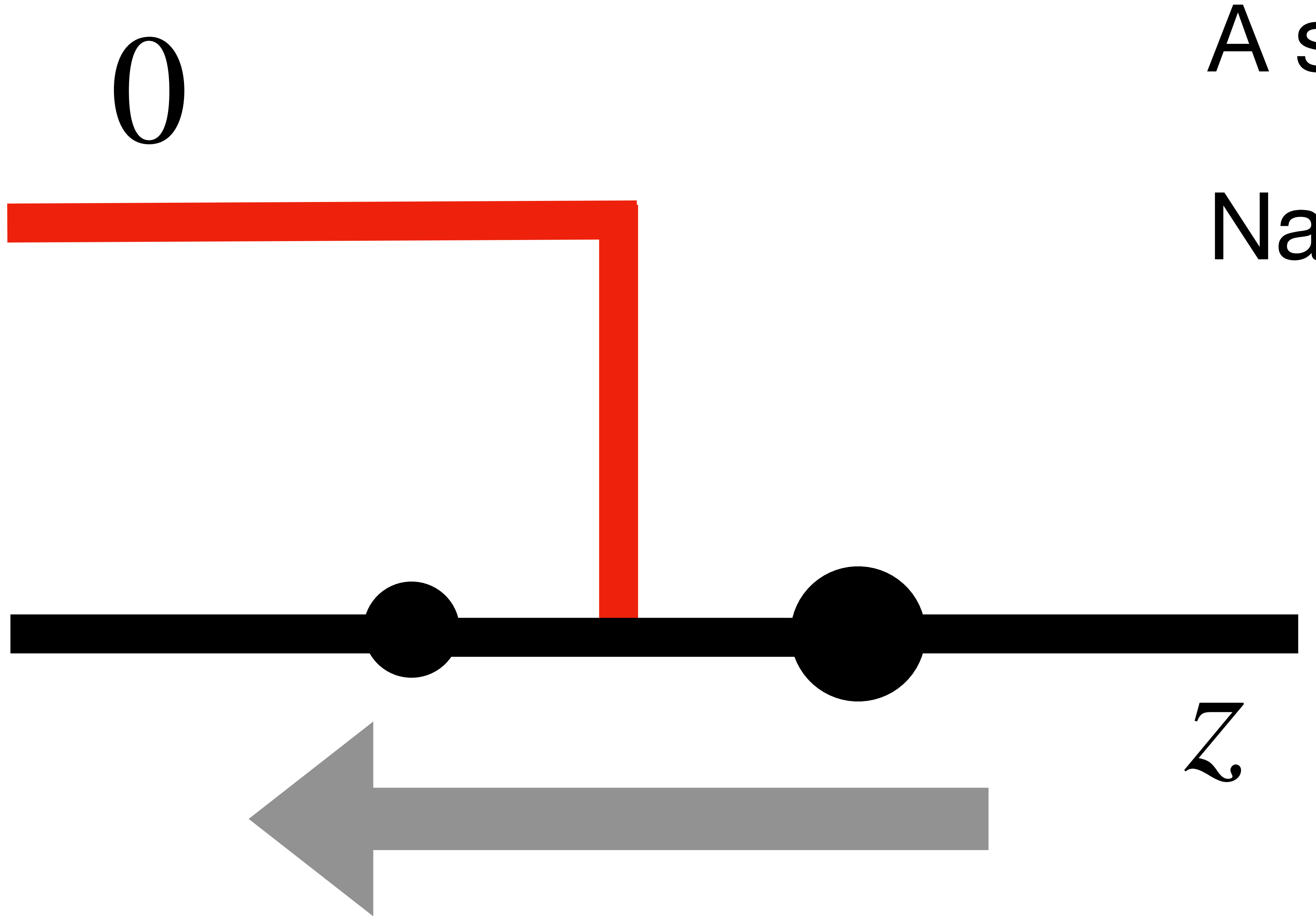
Namely, it is *bidirectional*



$$x = 0 \implies y = z$$

Switch Systems

Switch



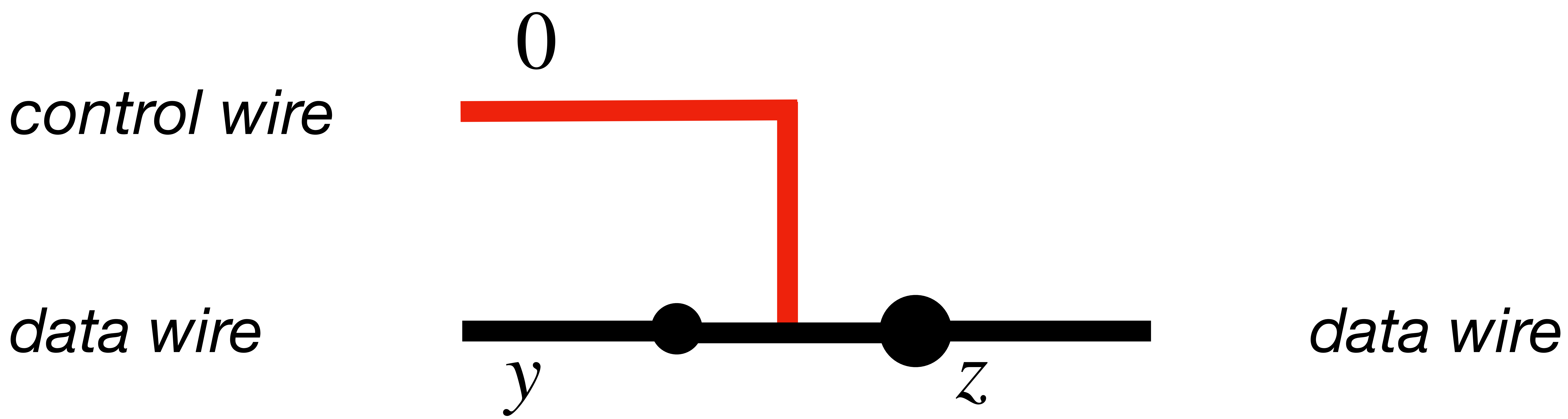
A switch enforces a *constraint*

Namely, it is *bidirectional*

$$x = 0 \implies y = z$$

Switch Systems

Switch



$$x = 0 \implies y = z$$

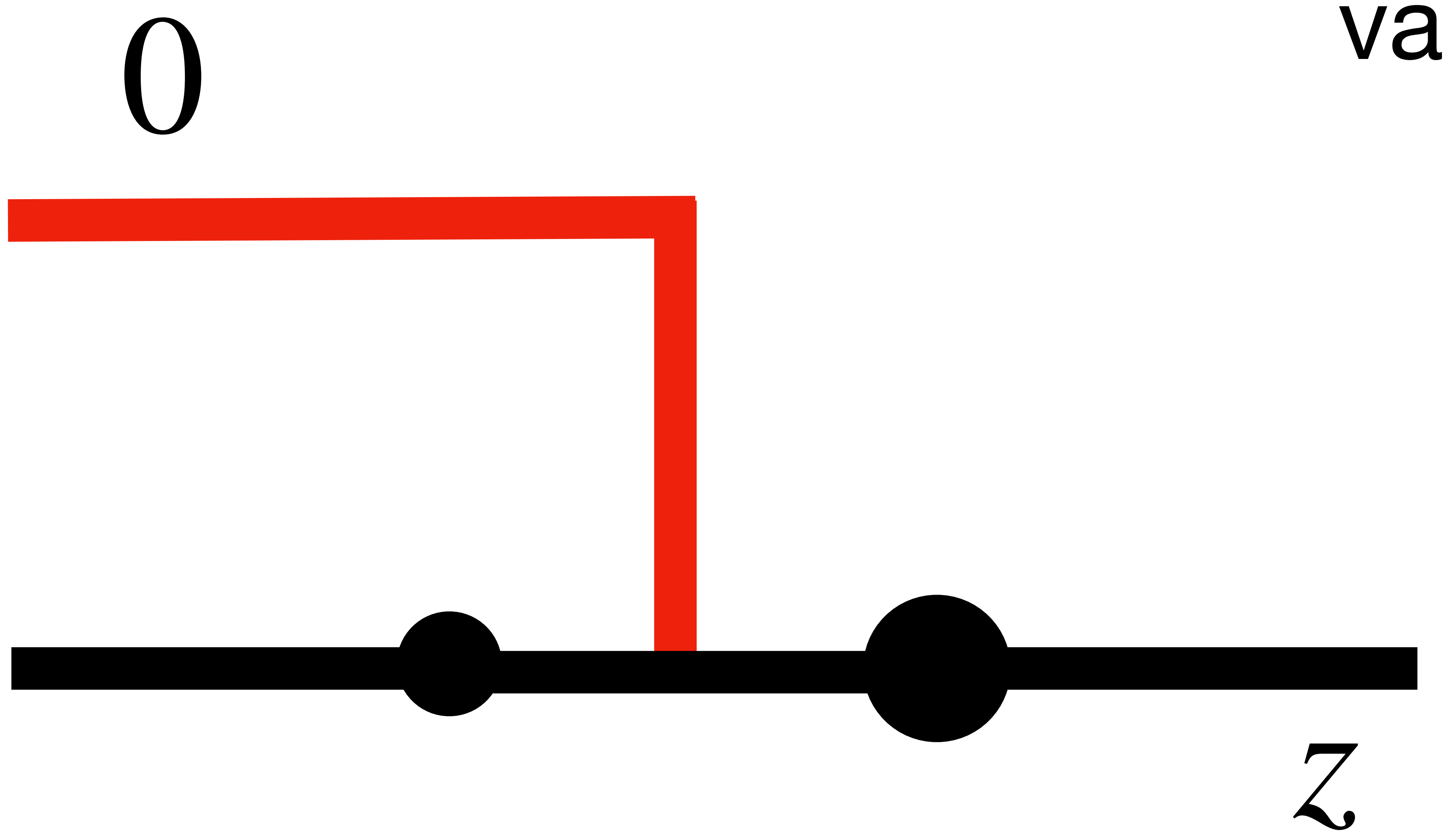
Insight: Garbler chooses one key per value per wire.

Difference between keys on data wires is equal to the hash of the zero control key

Switch Systems

Switch

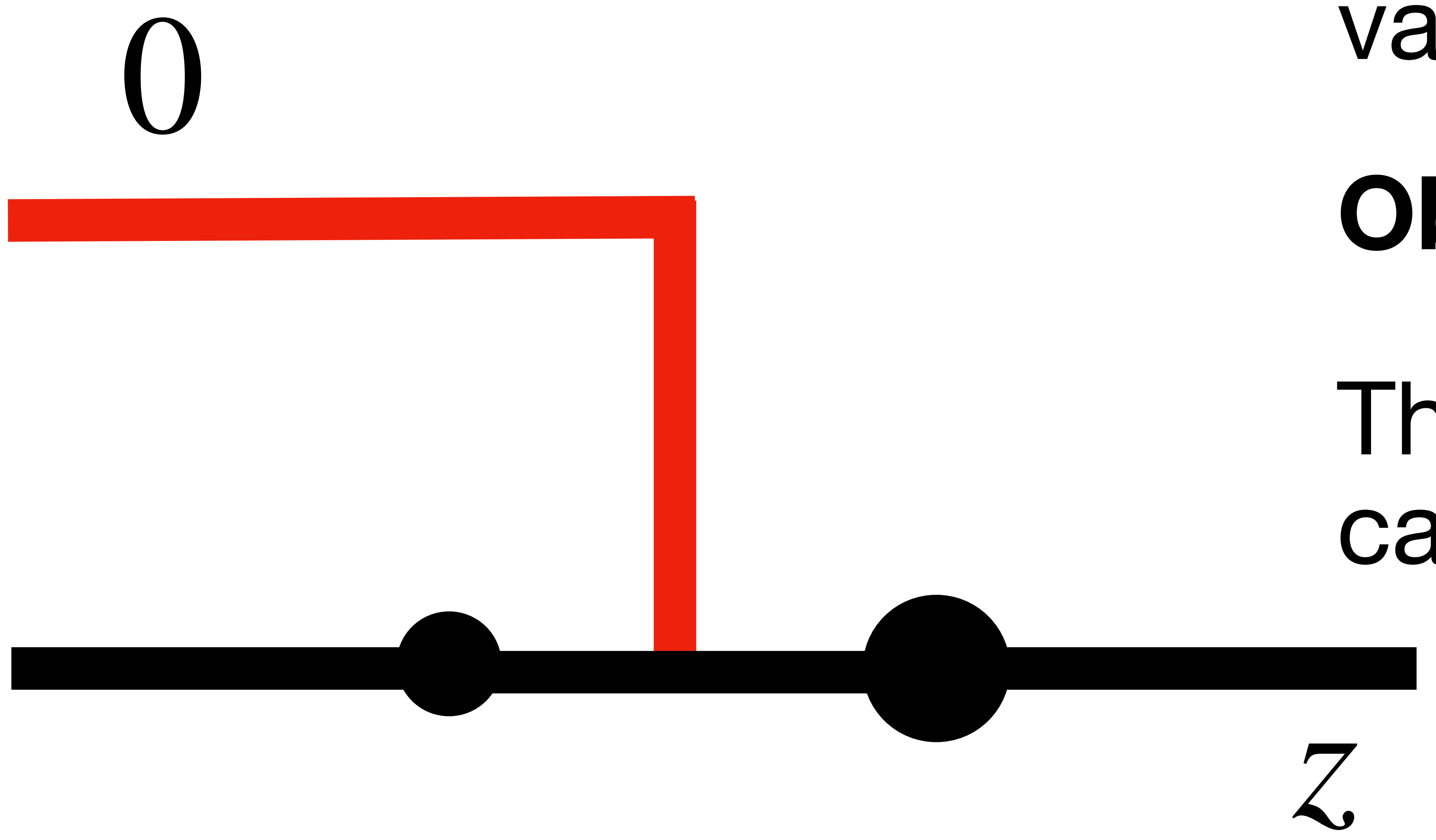
GC Evaluator will learn value of every control wire



$$x = 0 \implies y = z$$

Switch Systems

Switch



$$x = 0 \implies y = z$$

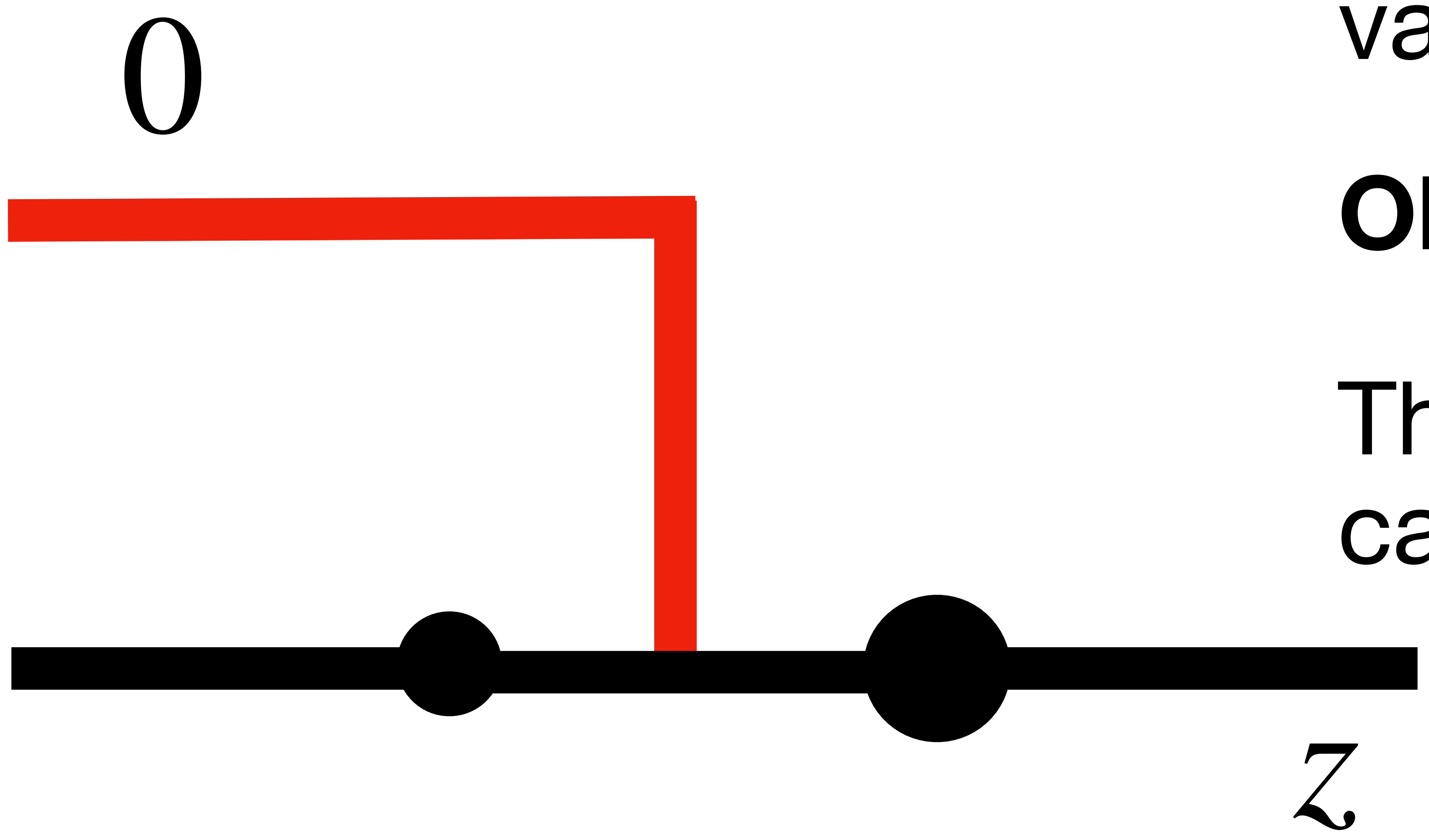
GC Evaluator will learn value of every control wire

Oblivious switch system:

The control wire values can be simulated

Switch Systems

Switch



$$x = 0 \implies y = z$$

GC Evaluator will learn value of every control wire

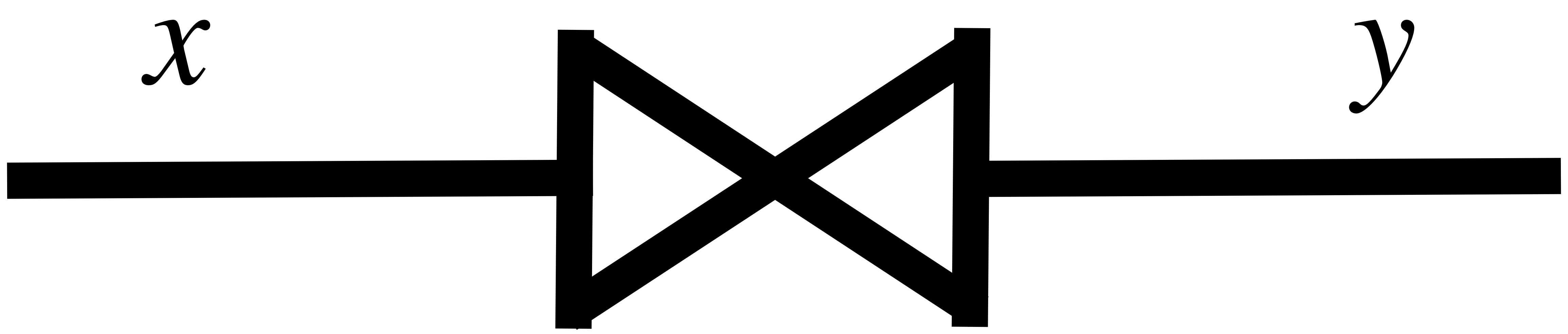
Oblivious switch system:

The control wire values can be simulated

Insight: Garbler can introduce one-time-pad masks that allow to safely reveal control values

Switch Systems

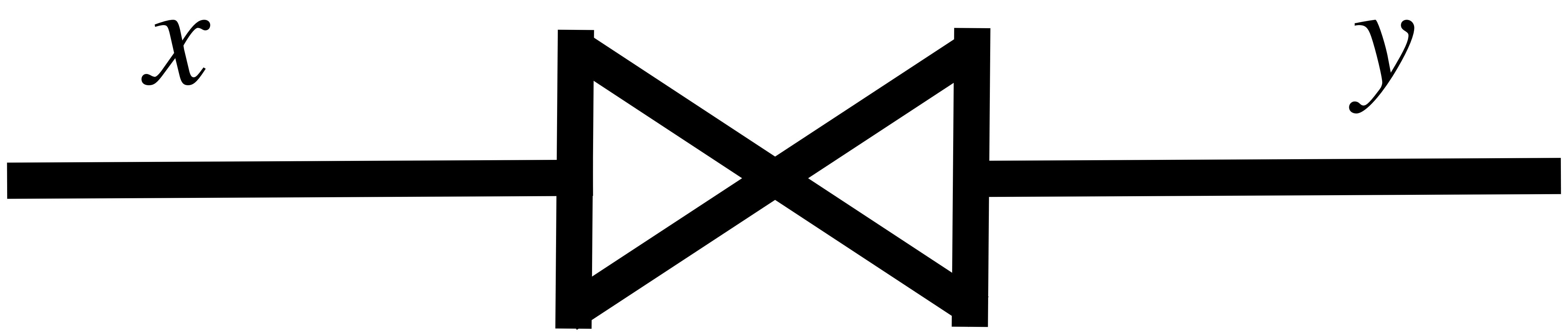
Join



$$x = y$$

Switch Systems

Join



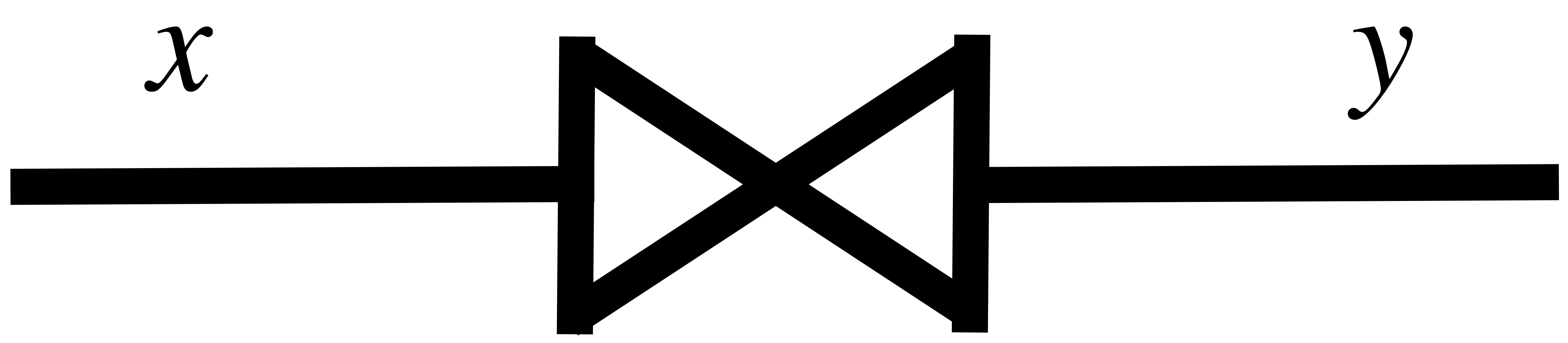
$$x = y$$

NOTE! The only gates that contribute to the size of a garbled circuit are joins!

Switch Systems

Join

Switch systems evaluate as a system of constraints, but they must be set up as a circuit



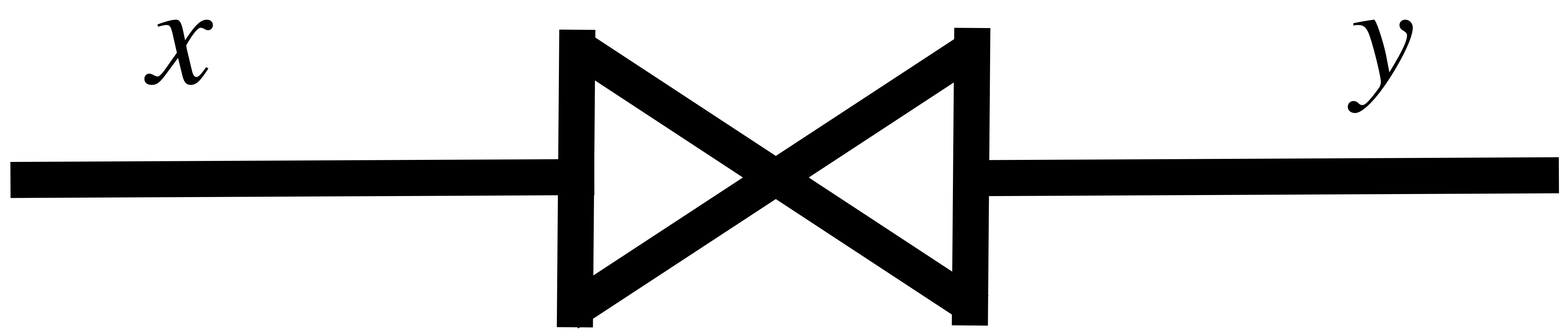
$$x = y$$

NOTE! The only gates that contribute to the size of a garbled circuit are joins!

Switch Systems

Join

Switch systems evaluate as a system of constraints, but they must be set up as a circuit



$$x = y$$

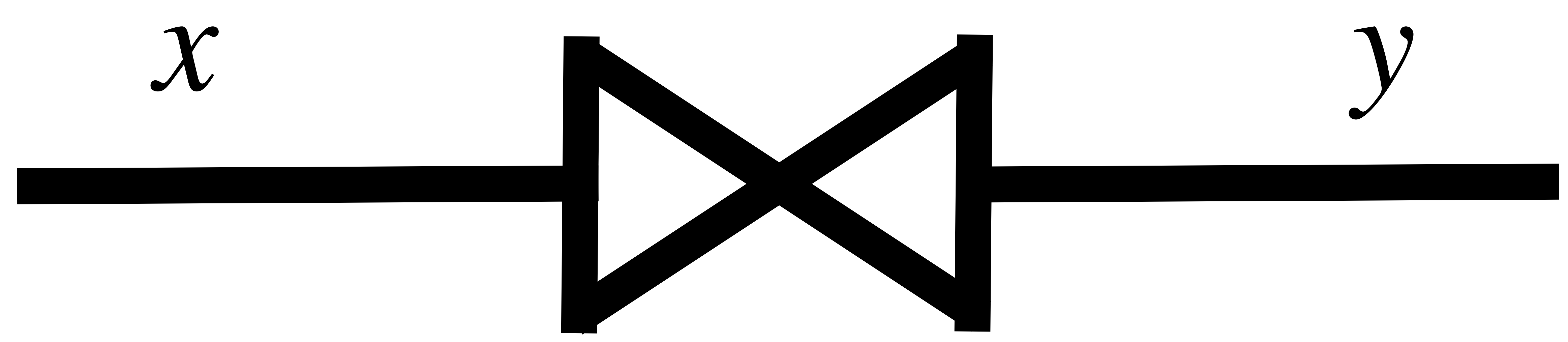
Insight: Garbler encrypts system in circuit order, evaluator solves constraints

NOTE! The only gates that contribute to the size of a garbled circuit are joins!

Switch Systems

Join

Switch systems evaluate as a system of constraints, but they must be set up as a circuit



$$x = y$$

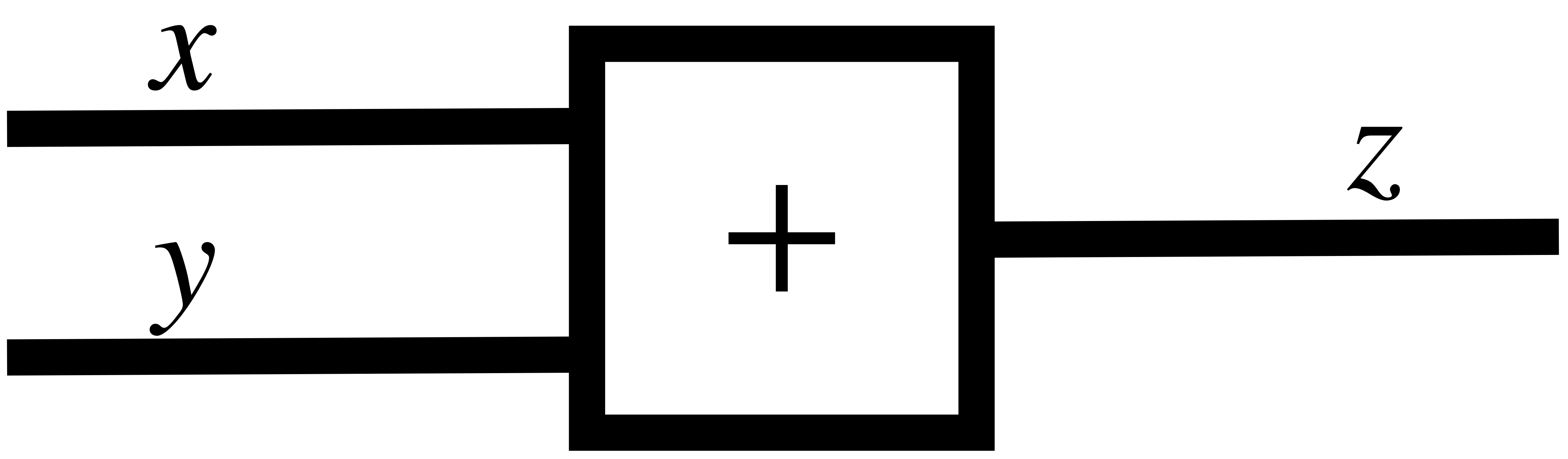
Insight: Garbler encrypts system in circuit order, evaluator solves constraints

NOTE! The only gates that contribute to the size of a garbled circuit are joins!

To improve GC handling, reduce the number of joins!

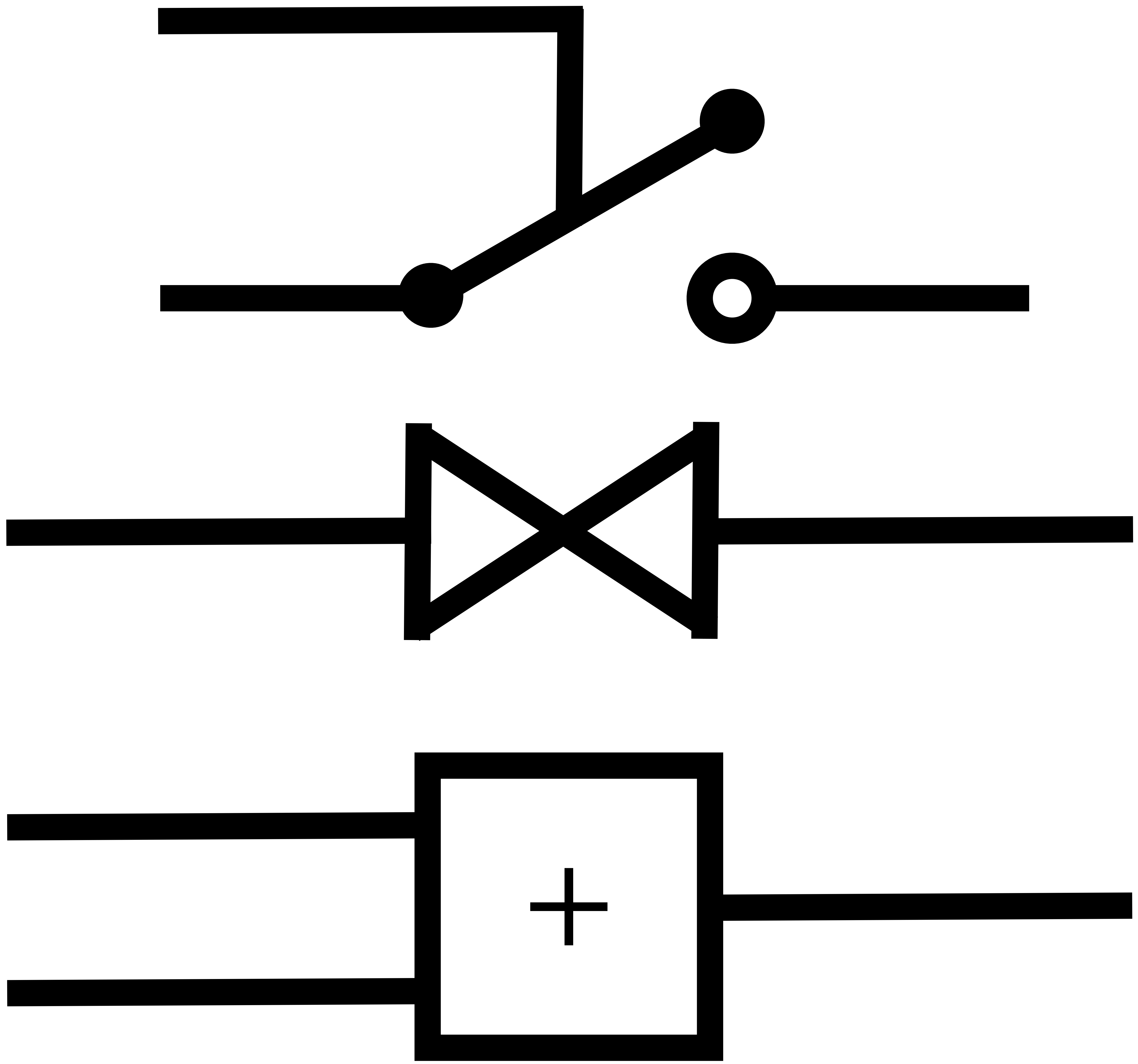
Switch Systems

ADD

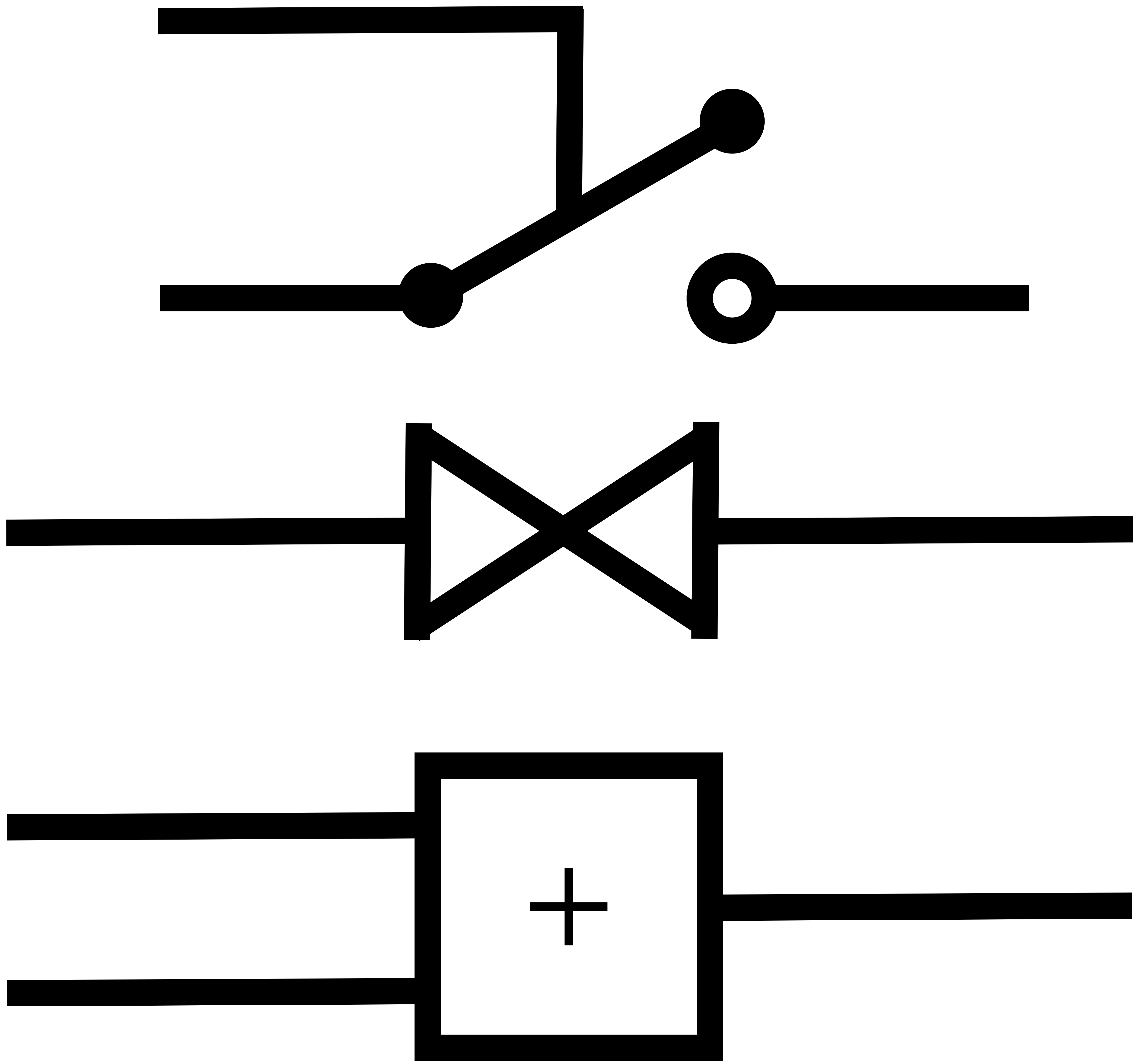


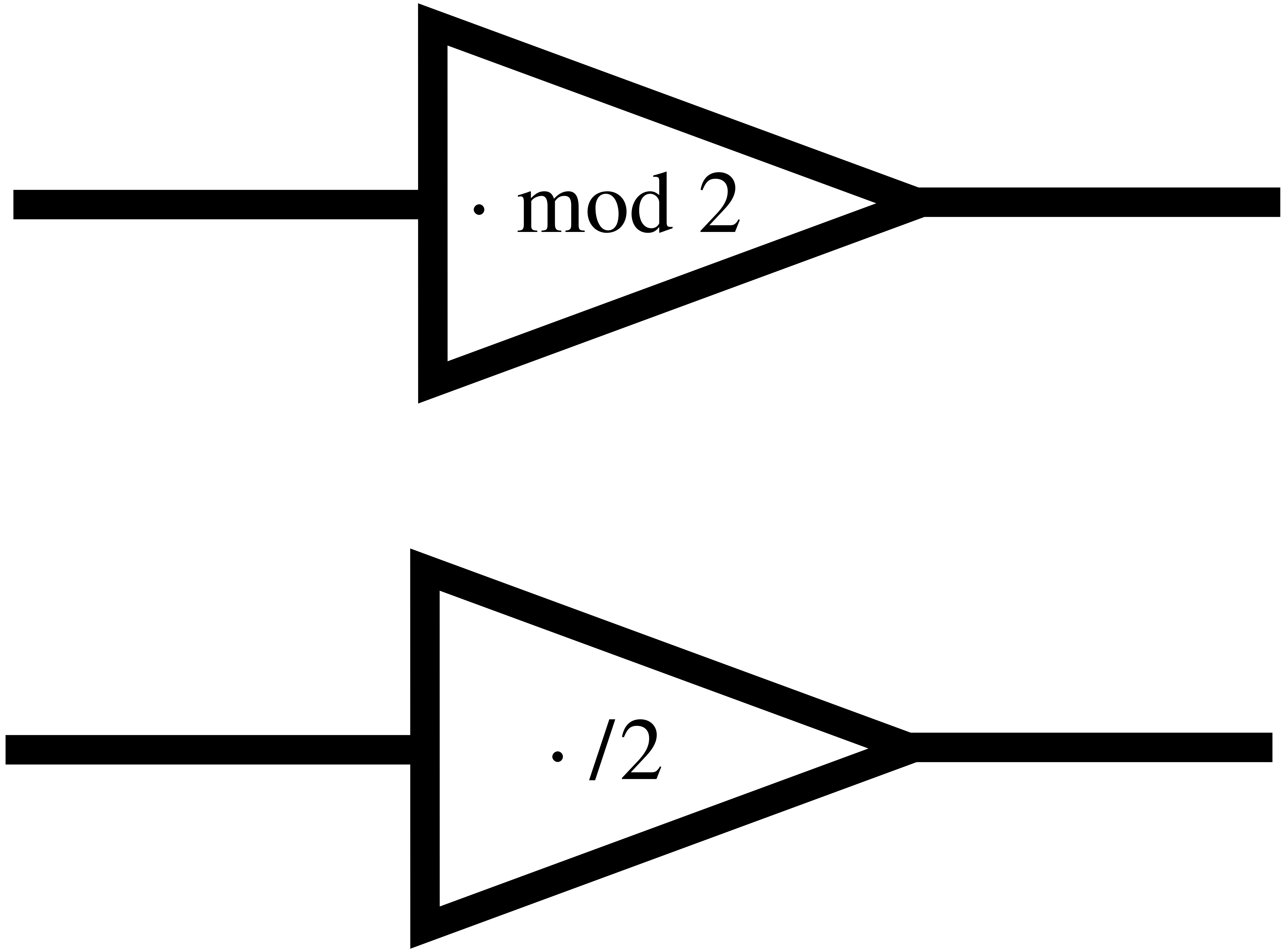
$$x + y = z$$

Switch Systems



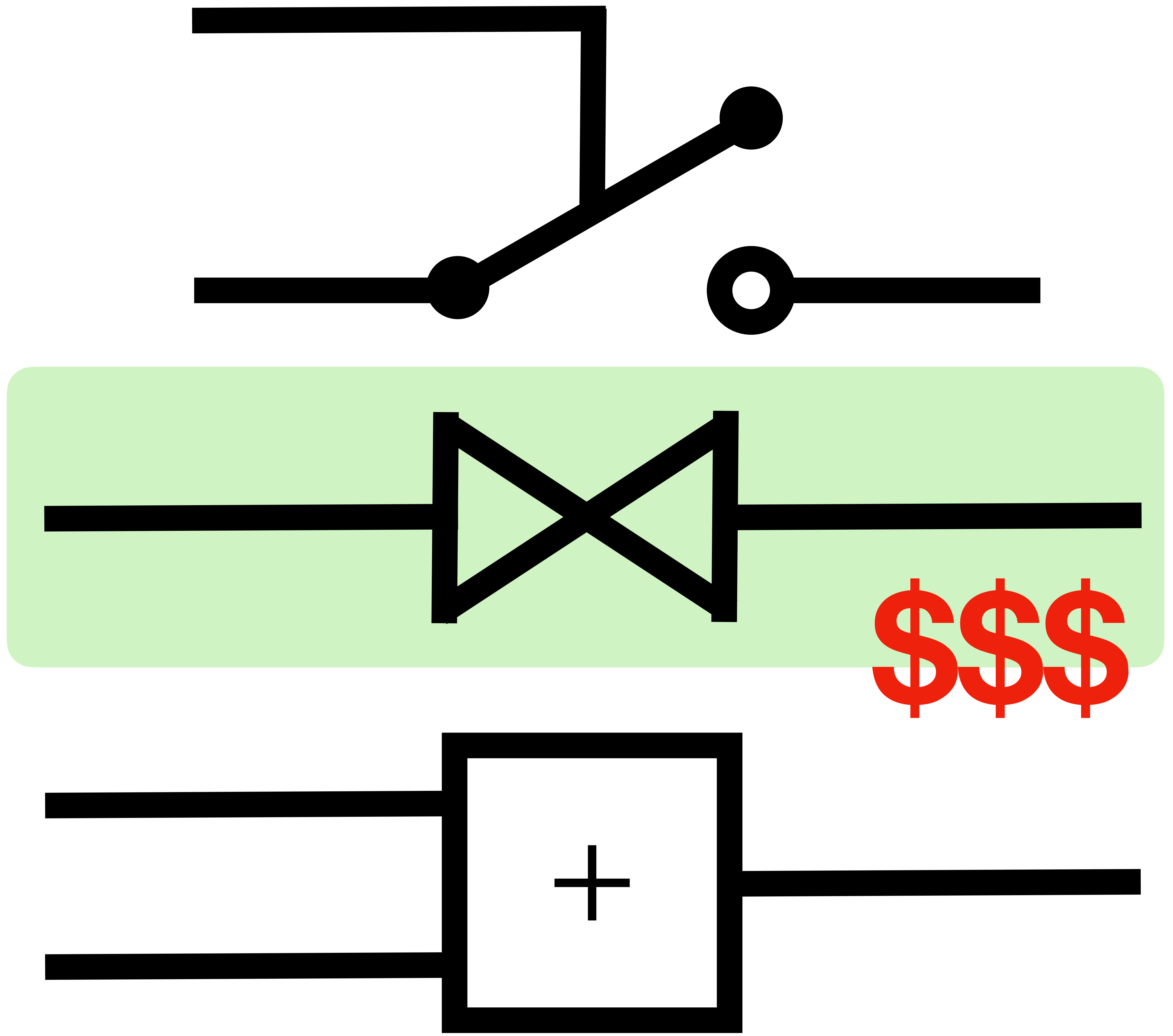
Switch Systems

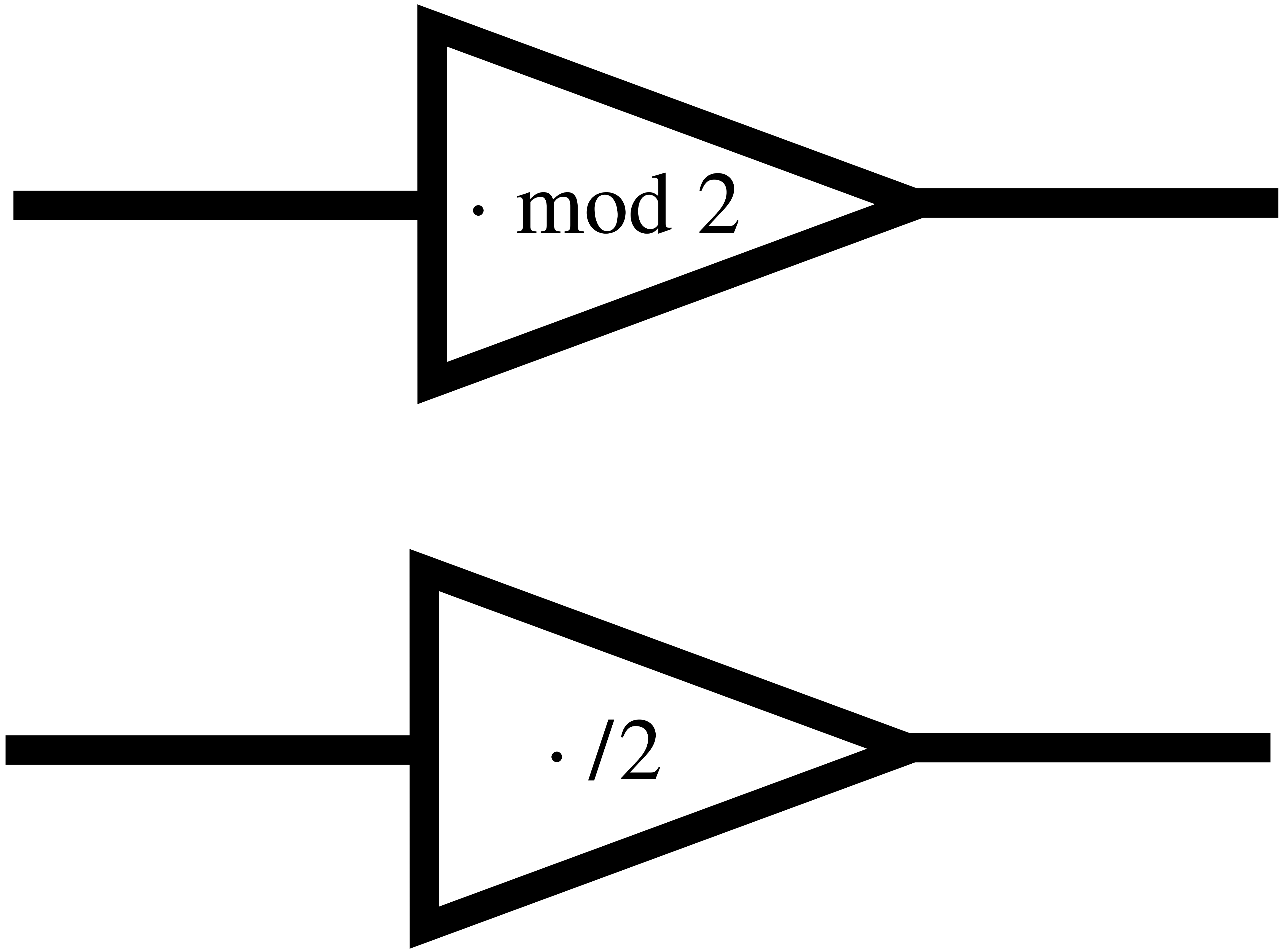




See paper

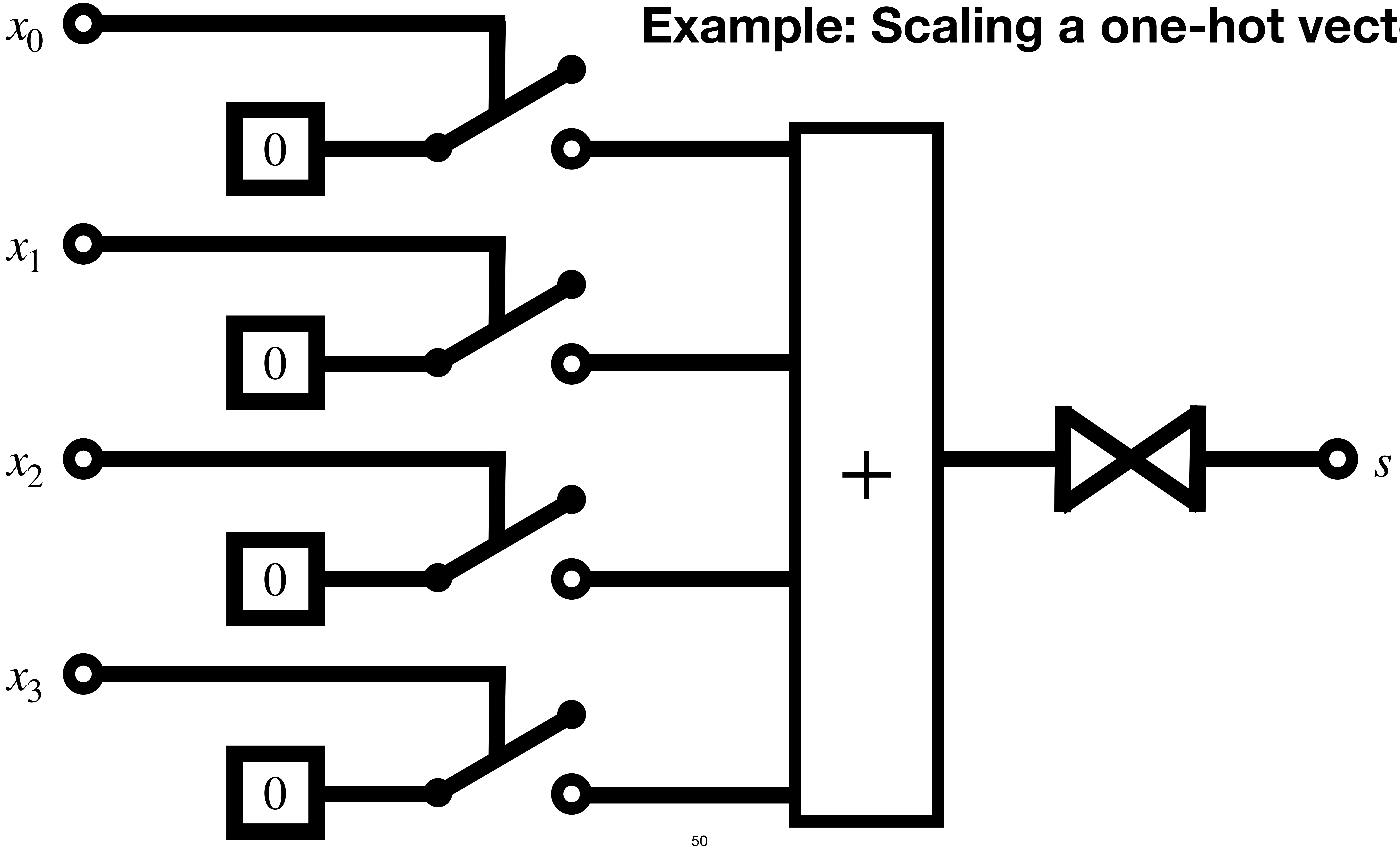
Switch Systems



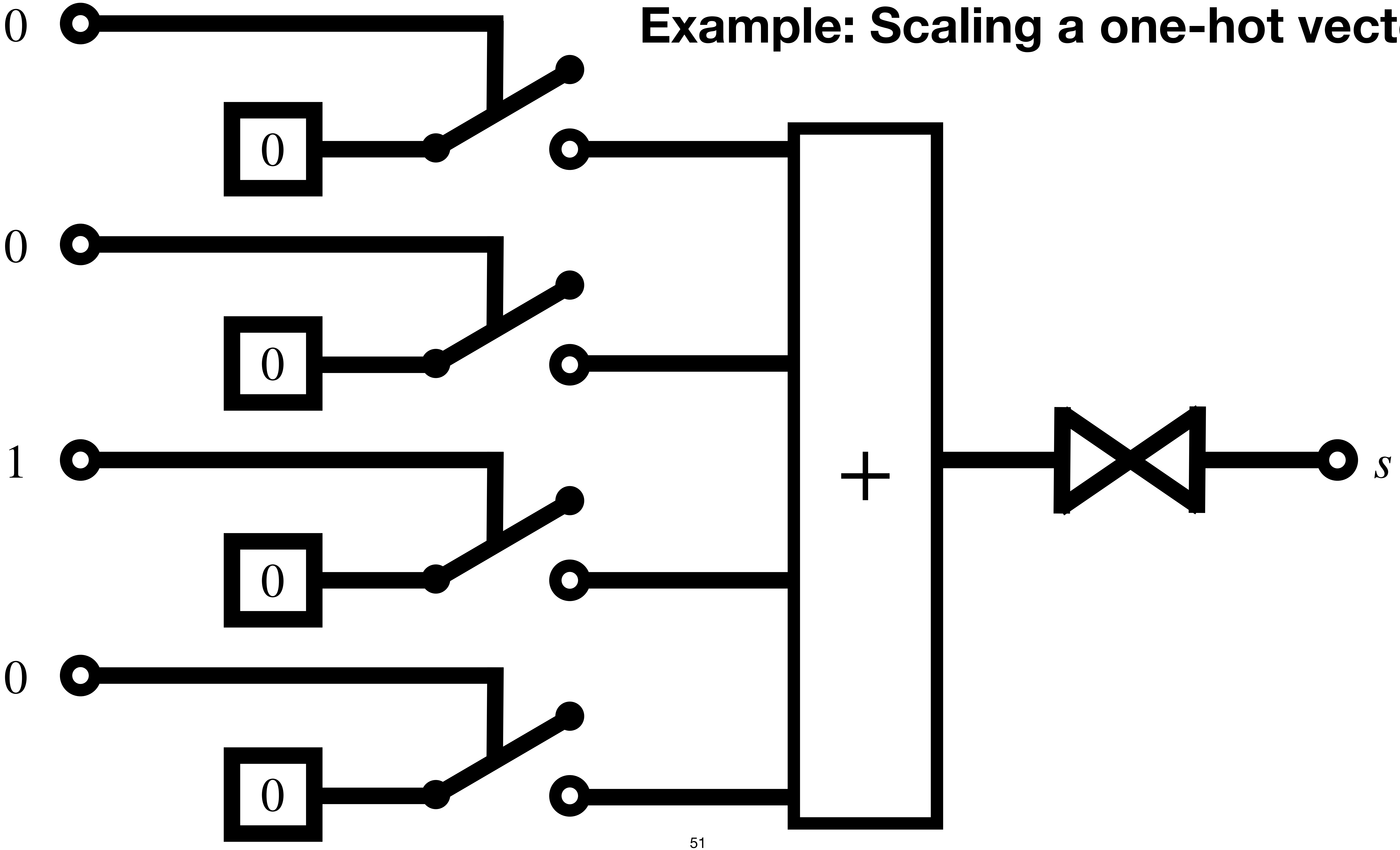


See paper

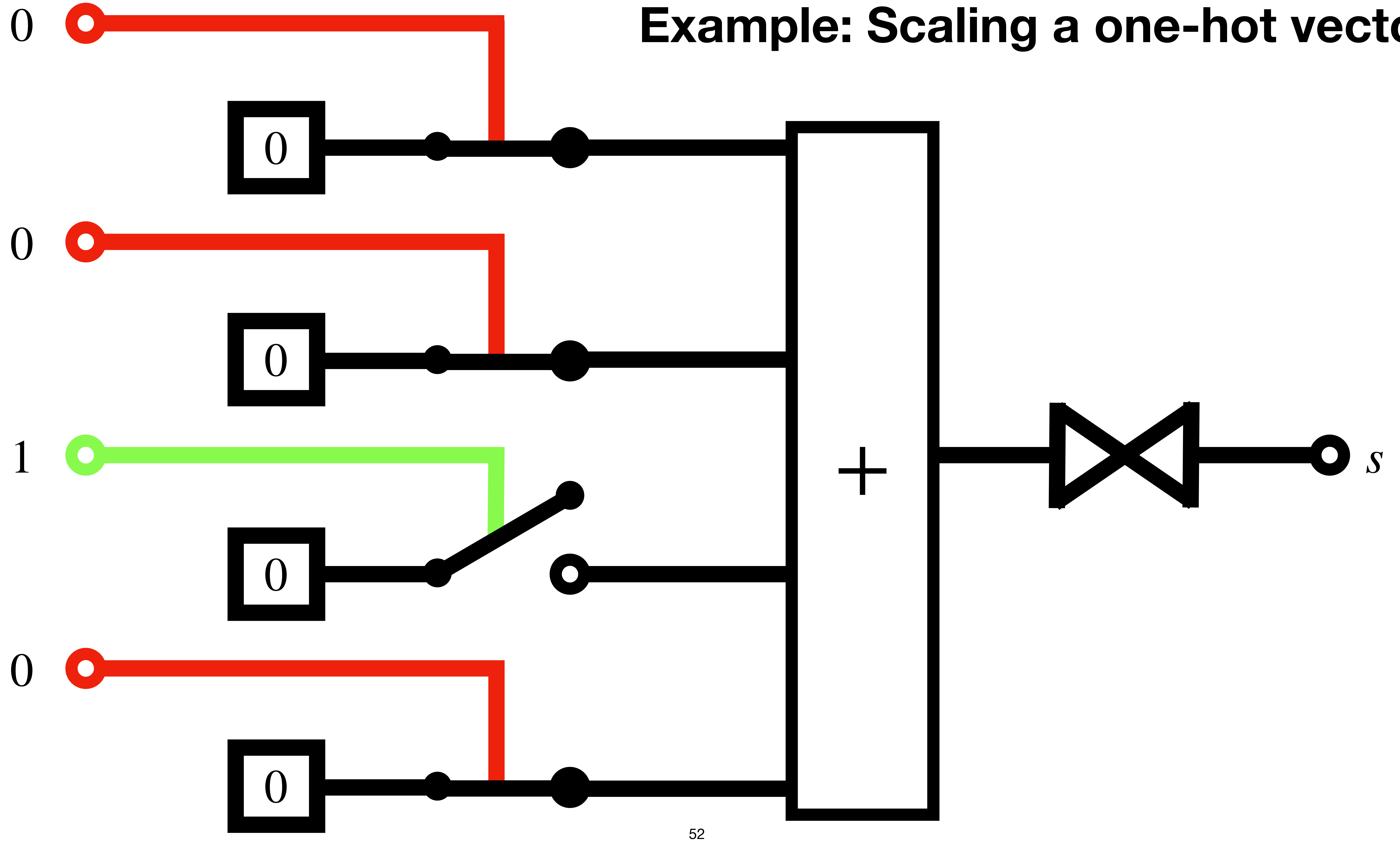
Example: Scaling a one-hot vector



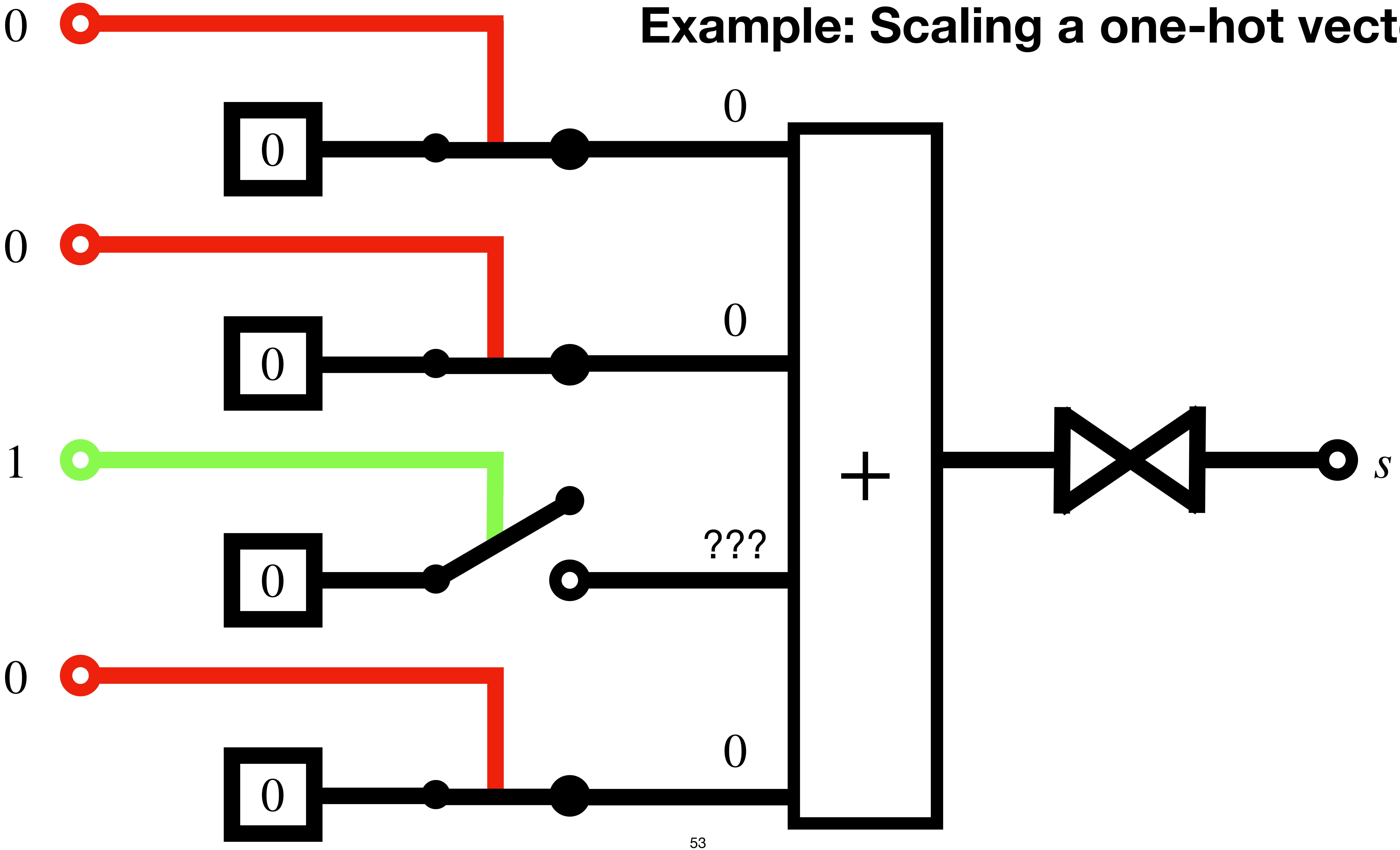
Example: Scaling a one-hot vector



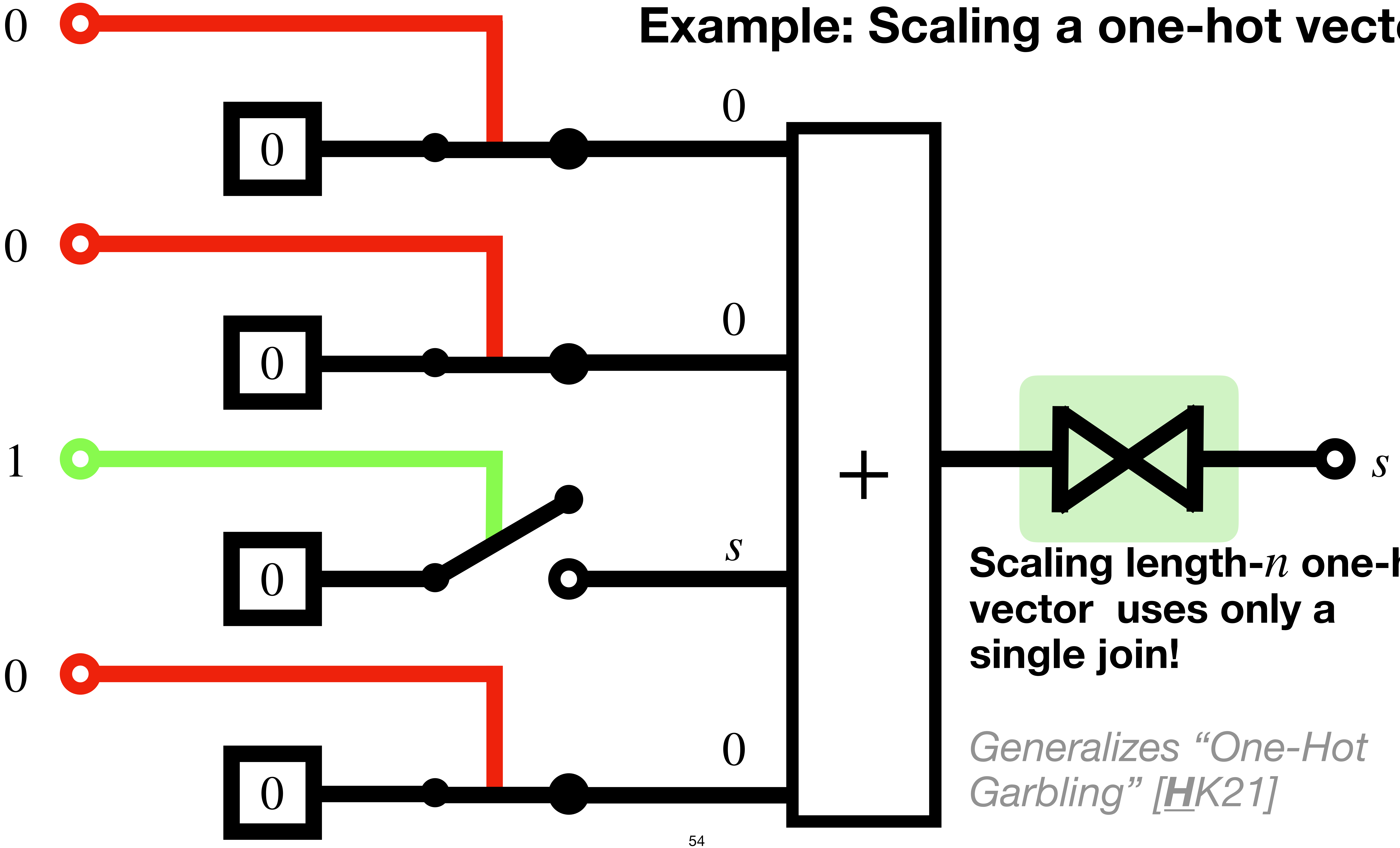
Example: Scaling a one-hot vector



Example: Scaling a one-hot vector



Example: Scaling a one-hot vector



Scaling length-n one-hot vector uses only a single join!

Generalizes "One-Hot Garbling" [HK21]

How to Multiply Short Integers

Let $x, y \in \mathbb{Z}_{2^k}$

How to Multiply Short Integers

Let $x, y \in \mathbb{Z}_{2^k}$

Suppose we have one-hot encoding of x and a single wire that encodes y

$$\begin{array}{r c} & \\ & \\ x\text{-th position} & \\ & \\ & \left[\begin{matrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{matrix}\right] \end{array}$$

For sake of example, let $x = 4$

How to Multiply Short Integers

Let $x, y \in \mathbb{Z}_{2^k}$

Suppose we have one-hot encoding of x and a single wire that encodes y

x-th position

$$\begin{bmatrix}0\\0\\0\\0\\1\\0\\0\\0\end{bmatrix} \xrightarrow{\text{scale by } y} \begin{bmatrix}0\\0\\0\\0\\y\\0\\0\\0\end{bmatrix}$$

For sake of example, let $x = 4$

How to Multiply Short Integers

Let $x, y \in \mathbb{Z}_{2^k}$

Suppose we have one-hot encoding of x and a single wire that encodes y

x-th position

$$\begin{bmatrix}0\\0\\0\\0\\1\\0\\0\\0\end{bmatrix} \xrightarrow{\text{scale by } y} \begin{bmatrix}0\\1\\2\\3\\4\\5\\6\\7\end{bmatrix}^\intercal \cdot \begin{bmatrix}0\\0\\0\\0\\y\\0\\0\\0\end{bmatrix}$$

For sake of example, let $x = 4$

Via ADD gates

How to Multiply Short Integers

Let $x, y \in \mathbb{Z}_{2^k}$

Suppose we have one-hot encoding of x and a single wire that encodes y

x-th position

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \xrightarrow{\text{scale by } y} \begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{bmatrix}^{\intercal} \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ y \\ 0 \\ 0 \\ 0 \end{bmatrix} = 4 \cdot y = x \cdot y$$

Single wire that encodes $x \cdot y$

Via ADD gates

For sake of example, let $x = 4$

How to Multiply Short Integers

Let $x, y \in \mathbb{Z}_{2^k}$

Suppose we have one-hot encoding of x and a single wire that encodes y

$$\begin{bmatrix}0\\0\\0\\0\\1\\0\\0\\0\end{bmatrix} \xrightarrow{\text{scale by } y} \begin{bmatrix}0\\1\\2\\3\\4\\5\\6\\7\end{bmatrix}^{\intercal} \cdot \begin{bmatrix}0\\0\\0\\0\\y\\0\\0\\0\end{bmatrix} = 4 \cdot y = x \cdot y$$

Single wire that encodes $x \cdot y$

Uses only a single join gate

How to Multiply Short Integers

Let $x, y \in \mathbb{Z}_{2^k}$

Suppose we have one-hot encoding of x and a single wire that encodes y

$$\begin{bmatrix}0\\0\\0\\0\\1\\0\\0\\0\end{bmatrix} \xrightarrow{\text{scale by } y} \begin{bmatrix}0\\1\\2\\3\\4\\5\\6\\7\end{bmatrix}^\intercal \cdot \begin{bmatrix}0\\0\\0\\0\\y\\0\\0\\0\end{bmatrix} = 4 \cdot y = x \cdot y$$

Single wire that encodes $x \cdot y$

Uses only a single join gate $\Rightarrow$ garbled encoding of size $k \cdot \lambda$ bits

How to Multiply Short Integers

$$\mathrm{hot}(x), y \mapsto x \cdot y$$

To multiply again, we need to convert single wire to one-hot encoding

This is achieved by another (complex) switch system, somewhat similar to one-hot scaling

See paper

This Work

First symmetric-key garbling scheme for arithmetic circuits that achieves linear-cost multiplication

switch systems generalize much of the garbled circuit literature

Opens possibility of new custom arithmetic garbled "gates"

See Paper For

More details on (oblivious) switch systems

How to garble switch systems

Switch system that converts between one-hot representation and arithmetic representation

Long integer handling, based on Chinese Remainder Theorem