Algebraic study of Glitches in Circuits

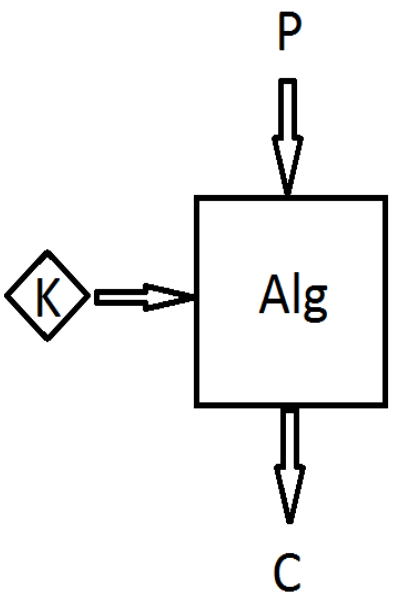
Molteni Maria Chiara

University of Trento and STMicroelectronics

16/11/2016

Cryptographic Algorithm

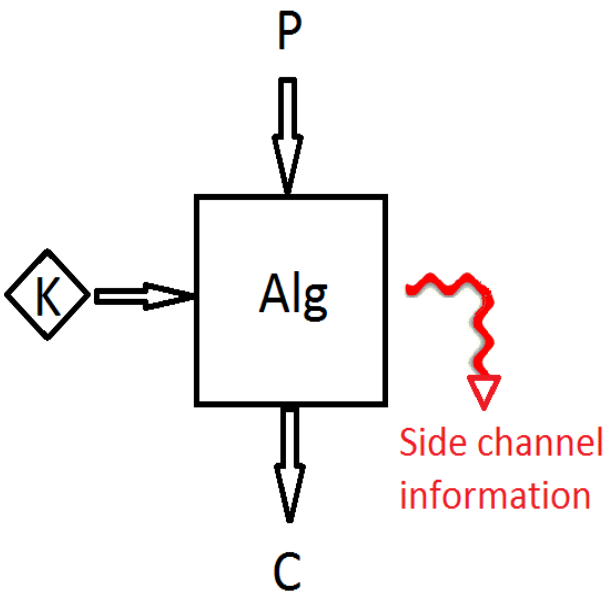
- P = Plaintext
- C = Ciphertext
- K = Key, unknown by the attacker



In a higher abstraction level, a **mathematical secure algorithm** denies attacker any knowledge about the key K, also if she knows algorithm operations.

Cryptographic Algorithm

- P = Plaintext
- C = Ciphertext
- K = Key, unknown by the attacker



Side-channel information are recovered from the encryption device while it execute the encryption operations (power consumption, electromagnetic radiation, execution time).

↓

These sources can carry information about the key, also if the algorithm is mathematically secure, and an attacker might try to exploit this link with a **side-channel attack**.

Glitches

Power consumption can be caused by glitches.

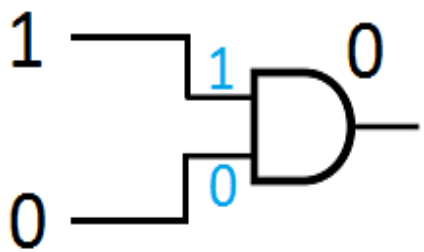
A **glitch** is a fast and unwanted change at the output of a gate in a circuit due to different time propagation of the inputs.

Glitches

Power consumption can be caused by glitches.

A **glitch** is a fast and unwanted change at the output of a gate in a circuit due to different time propagation of the inputs.

Example

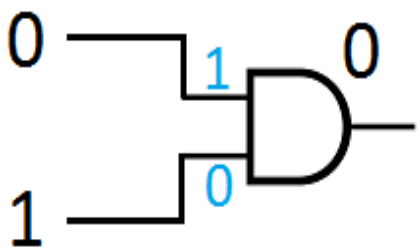


Glitches

Power consumption can be caused by glitches.

A **glitch** is a fast and unwanted change at the output of a gate in a circuit due to different time propagation of the inputs.

Example

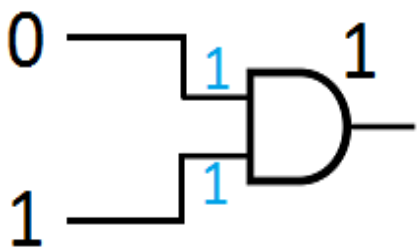


Glitches

Power consumption can be caused by glitches.

A **glitch** is a fast and unwanted change at the output of a gate in a circuit due to different time propagation of the inputs.

Example

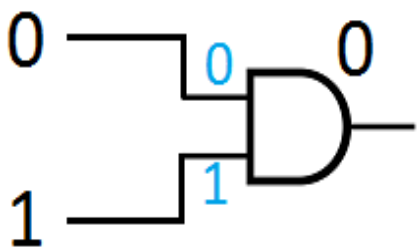


Glitches

Power consumption can be caused by glitches.

A **glitch** is a fast and unwanted change at the output of a gate in a circuit due to different time propagation of the inputs.

Example



STATIC HAZARD

Transients

In a circuit, bit changes can be represented by **transients**.

A **transient** is a bitstring t without any repetition of zeros and ones:

$$t = a_1 a_2 ... a_n \text{ with } a_i \in \mathbb{Z}_2 \qquad \forall i \in \{1, 2, ..., n\}$$

s.t. $a_i \neq a_{i+1} \qquad \forall i \in \{1, 2, ..., n-1\}$

Transients

In a circuit, bit changes can be represented by **transients**.

A **transient** is a bitstring t without any repetition of zeros and ones:

$$t = a_1 a_2 ... a_n \text{ with } a_i \in \mathbb{Z}_2 \qquad \forall i \in \{1, 2, ..., n\}$$

s.t. $a_i \neq a_{i+1} \qquad \forall i \in \{1, 2, ..., n-1\}$

Example

Transients

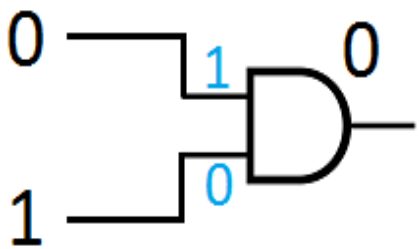
In a circuit, bit changes can be represented by **transients**.

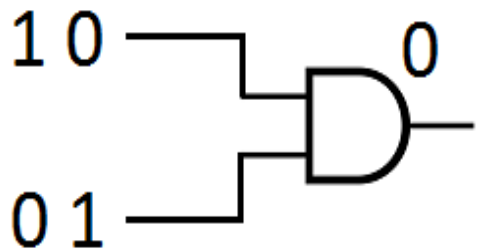
A **transient** is a bitstring t without any repetition of zeros and ones:

$$t = a_1 a_2 ... a_n \text{ with } a_i \in \mathbb{Z}_2 \qquad \forall i \in \{1, 2, ..., n\}$$

s.t. $a_i \neq a_{i+1} \qquad \forall i \in \{1, 2, ..., n-1\}$

Example





Transients

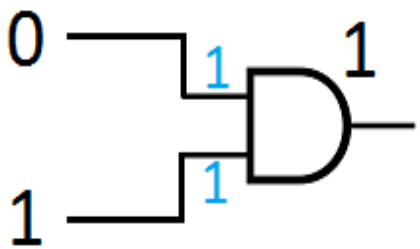
In a circuit, bit changes can be represented by **transients**.

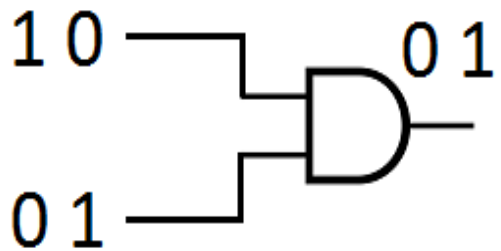
A **transient** is a bitstring t without any repetition of zeros and ones:

$$t = a_1 a_2 ... a_n \text{ with } a_i \in \mathbb{Z}_2 \qquad \forall i \in \{1, 2, ..., n\}$$

s.t. $a_i \neq a_{i+1} \qquad \forall i \in \{1, 2, ..., n-1\}$

Example





Transients

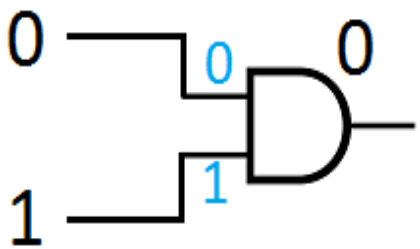
In a circuit, bit changes can be represented by **transients**.

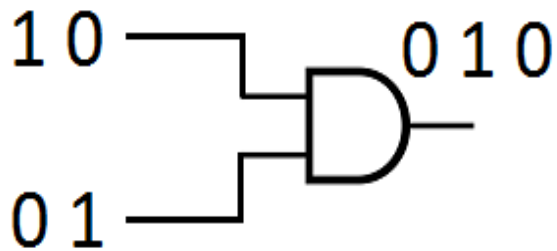
A **transient** is a bitstring t without any repetition of zeros and ones:

$$t = a_1 a_2 ... a_n \text{ with } a_i \in \mathbb{Z}_2 \qquad \forall i \in \{1, 2, ..., n\}$$

s.t. $a_i \neq a_{i+1} \qquad \forall i \in \{1, 2, ..., n-1\}$

Example





Transients

$\mathcal{T}$ is the set of transients, $e \in \mathcal{T}$ is the empty transient

Transients' properties

Given $t \in \mathcal{T}$, $t = a_1 a_2 ... a_n$:

- $\alpha(t) = a_1 \in \mathbb{Z}_2$
- $\omega(t) = a_n \in \mathbb{Z}_2$
- $u(t) = |\{i \mid a_i = 1\}| \in \mathbb{N}$
- $z(t) = |\{i \mid a_i = 0\}| \in \mathbb{N}$
- $\ell(t) = n \in \mathbb{N}$

Worst case scenario for glitches propagation

The **worst case scenario for glitches propagation** in a circuit happens when in each gate the maximum number possible of glitches occurs. This situation implies the maximum power consumption in the circuit, and this is why it is "the worst case".

Given a circuit, this situation is described using transients and operations among them.
It is possible to define 3 operations on set $\mathcal{T}$.

Operations on $\mathcal{T}$: $\boxplus$

This operation among transient is equivalent to OR the signals in the worst case.

Given $t, t' \in \mathcal{T}$:

- $t \boxplus 0 = 0 \boxplus t = t$
- $t \boxplus 1 = 1 \boxplus t = 1$
- if $\ell(t), \ell(t') > 1$, $w = t \boxplus t'$ s.t.:
  - $\alpha(w) = \alpha(t) \vee \alpha(t')$
  - $\omega(w) = \omega(t) \vee \omega(t')$
  - $z(w) = z(t) + z(t') - 1$

Example: if $t = 010$ and $t' = 1010$, then $w = 010 \boxplus 1010 = 101010$

Operations on $\mathcal{T}$: $\boxtimes$

This operation among transient is equivalent to AND the signals in the worst case.

Given $t, t' \in \mathcal{T}$:

- $t \boxtimes 0 = 0 \boxtimes t = 0$
- $t \boxtimes 1 = 1 \boxtimes t = t$
- if $\ell(t), \ell(t') > 1$, $w = t \boxtimes t'$ s.t.:
  - $\alpha(w) = \alpha(t) \wedge \alpha(t')$
  - $\omega(w) = \omega(t) \wedge \omega(t')$
  - $u(w) = u(t) + u(t') - 1$

Example: if $t = 010$ and $t = 1010$, then $w = 010 \boxtimes 1010 = 01010$

Operation on $\mathcal{T}$: $\overline{\cdot}$

This operation transform a transient in its complement, and it is equivalent to NOT the signal.

Given $t \in \mathcal{T}$, $\overline{t}$ is s.t.:

- $\alpha(\overline{t}) = \overline{\alpha(t)}$
- $\ell(\overline{t}) = \ell(t)$

Example: if $t = 0101$, then $\overline{t} = 1010$

Hazard Algebra

Considering these 3 operations on $\mathcal{T}$, we can enunciate the following theorem:

Theorem

$\mathcal{C} = (\mathcal{T}, \boxplus, \boxtimes, \overline{\cdot}, 0, 1)$ *is a* **commutative de Morgan bisemigroup**, *i.e, given* $x, y, z \in T$, *it satisfies:*

L1	$x \boxplus y = y \boxplus x$	*L1'*	$x \boxtimes y = y \boxtimes x$
L2	$x \boxplus (y \boxplus z) = (x \boxplus y) \boxplus z$	*L2'*	$x \boxtimes (y \boxtimes z) = (x \boxtimes y) \boxtimes z$
L3	$x \boxplus 1 = 1$	*L3'*	$x \boxtimes 0 = 0$
L4	$x \boxplus 0 = x$	*L4'*	$x \boxtimes 1 = x$
L5	$\overline{\overline{x}} = x$		
L6	$\overline{x \boxplus y} = \overline{x} \boxtimes \overline{y}$	*L6'*	$\overline{x \boxtimes y} = \overline{x} \boxplus \overline{y}$

Glitch-counting algorithm

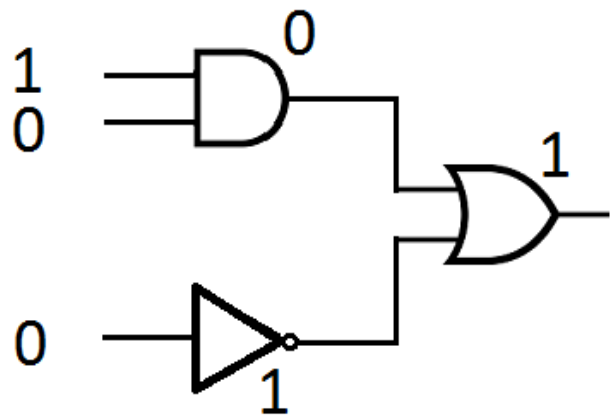
Let R a circuit. Starting from transients that represent inputs' switches, it is possible to define transients for each gate in R, which represent the worst case possible of glitches propagation.

Glitch-counting algorithm

Let R a circuit. Starting from transients that represent inputs' switches, it is possible to define transients for each gate in R, which represent the worst case possible of glitches propagation.

Example

Initial Stable State:

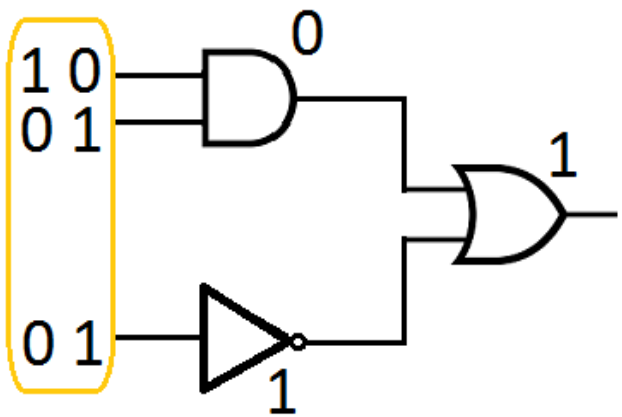


Glitch-counting algorithm

Let R a circuit. Starting from transients that represent inputs' switches, it is possible to define transients for each gate in R, which represent the worst case possible of glitches propagation.

Example

After inputs' switches, Unstable State:



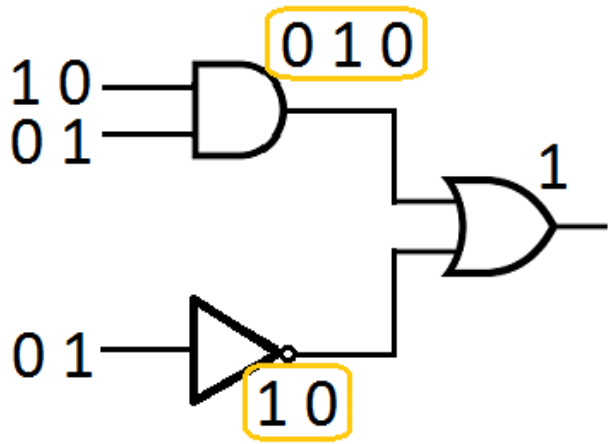
Glitch-counting algorithm

Let R a circuit. Starting from transients that represent inputs' switches, it is possible to define transients for each gate in R, which represent the worst case possible of glitches propagation.

Example

Glitches propagation (worst case):

- gates at height 1
  $10 \boxtimes 01 = 010$ and $\overline{01} = 10$



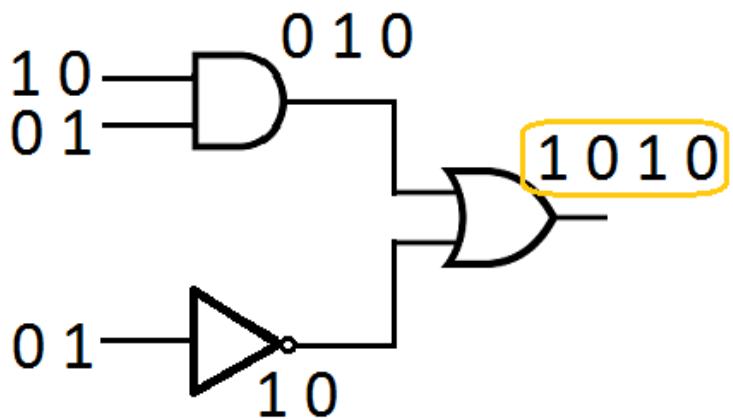
Glitch-counting algorithm

Let R a circuit. Starting from transients that represent inputs' switches, it is possible to define transients for each gate in R, which represent the worst case possible of glitches propagation.

Example

Glitches propagation (worst case):

- gates at height 1
  $10 \boxtimes 01 = 010$ and $\overline{01} = 10$
- gates at height 2
  $010 \boxplus 10 = 1010$



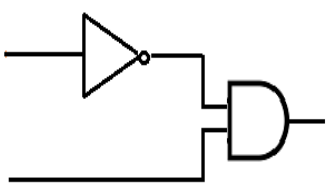
Propagation Sequences: informal definition

Propagation Sequences

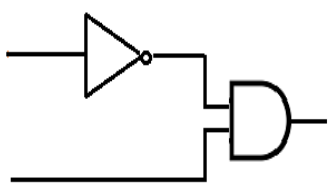
A Propagation Sequence in a circuit is a specific propagation of the signal in it, defined specifying, for each gate with 2 or more inputs, the order whereby inputs' changes have effects on the gate. Different Propagation Sequences in a circuit determine different propagation of glitches.

Example

Propagation Sequence 1



Propagation Sequence 2



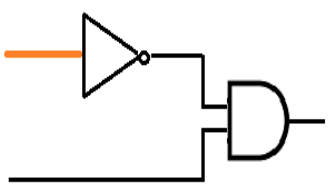
Propagation Sequences: informal definition

Propagation Sequences

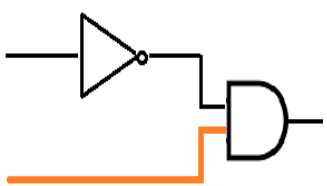
A Propagation Sequence in a circuit is a specific propagation of the signal in it, defined specifying, for each gate with 2 or more inputs, the order whereby inputs' changes have effects on the gate. Different Propagation Sequences in a circuit determine different propagation of glitches.

Example

Propagation Sequence 1



Propagation Sequence 2



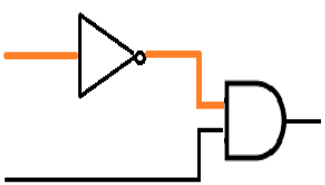
Propagation Sequences: informal definition

Propagation Sequences

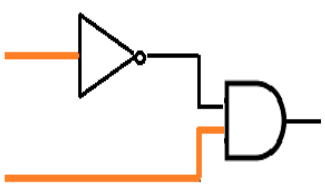
A Propagation Sequence in a circuit is a specific propagation of the signal in it, defined specifying, for each gate with 2 or more inputs, the order whereby inputs' changes have effects on the gate. Different Propagation Sequences in a circuit determine different propagation of glitches.

Example

Propagation Sequence 1



Propagation Sequence 2



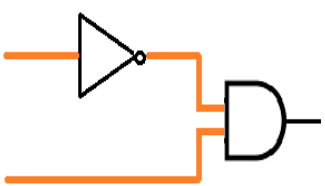
Propagation Sequences: informal definition

Propagation Sequences

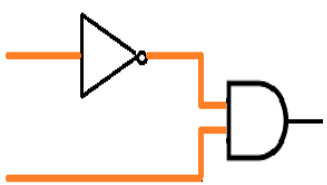
A Propagation Sequence in a circuit is a specific propagation of the signal in it, defined specifying, for each gate with 2 or more inputs, the order whereby inputs' changes have effects on the gate. Different Propagation Sequences in a circuit determine different propagation of glitches.

Example

Propagation Sequence 1



Propagation Sequence 2

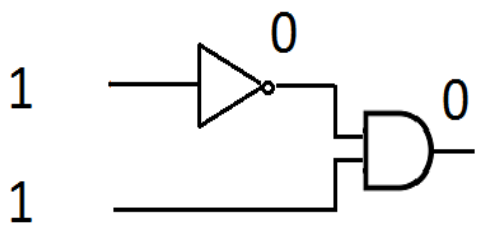


Glitch-counting algorithm for PS

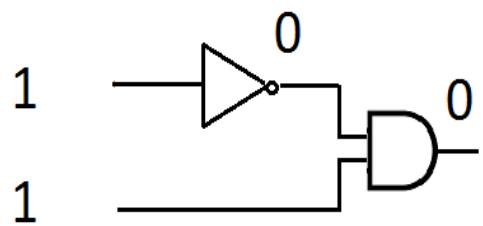
Given a circuit with some input transients and defined a propagation sequence on it, transients for each gate can be computed with an algorithm similar to the Glitch-counting algorithm.

Example

Propagation Sequence 1



Propagation Sequence 2

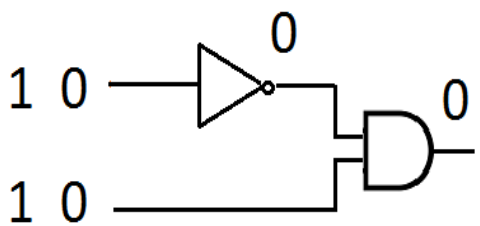


Glitch-counting algorithm for PS

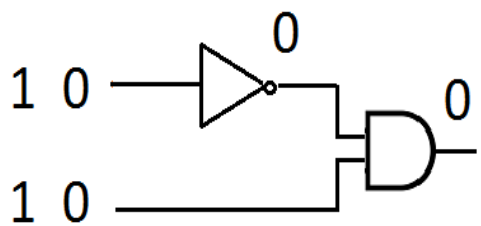
Given a circuit with some input transients and defined a propagation sequence on it, transients for each gate can be computed with an algorithm similar to the Glitch-counting algorithm.

Example

Propagation Sequence 1



Propagation Sequence 2

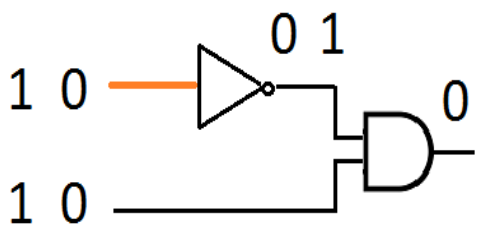


Glitch-counting algorithm for PS

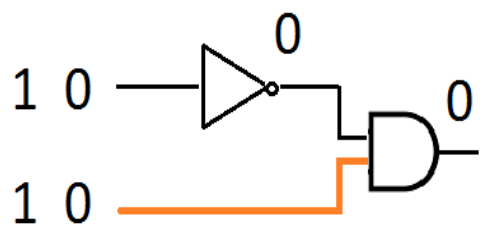
Given a circuit with some input transients and defined a propagation sequence on it, transients for each gate can be computed with an algorithm similar to the Glitch-counting algorithm.

Example

Propagation Sequence 1



Propagation Sequence 2

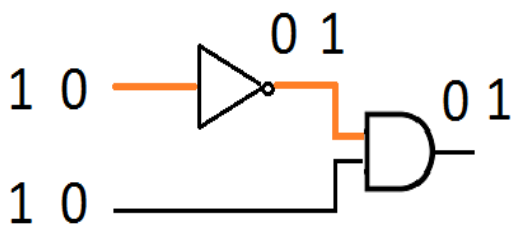


Glitch-counting algorithm for PS

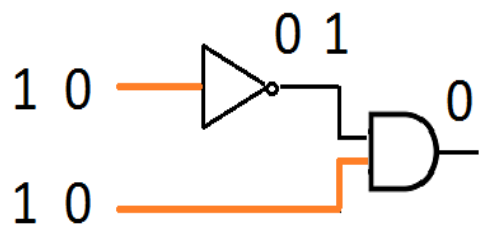
Given a circuit with some input transients and defined a propagation sequence on it, transients for each gate can be computed with an algorithm similar to the Glitch-counting algorithm.

Example

Propagation Sequence 1



Propagation Sequence 2

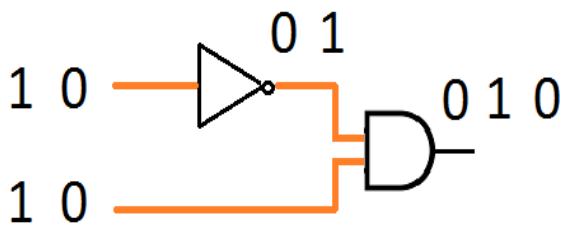


Glitch-counting algorithm for PS

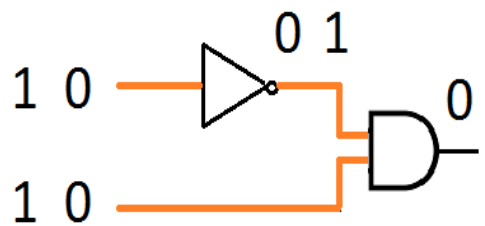
Given a circuit with some input transients and defined a propagation sequence on it, transients for each gate can be computed with an algorithm similar to the Glitch-counting algorithm.

Example

Propagation Sequence 1



Propagation Sequence 2



Considering different Propagation Sequences in a circuit, it is possible to observe different outputs' behaviours for the same gates.

Combinations

Combination

Given a circuit with n inputs, we call *combination* the string $a_1 a_2 ... a_n$, such that $a_i \in \{1, ..., n\}$ and $a_i \neq a_j \ \forall i \neq j$ and $1 \leq i, j \leq n$.

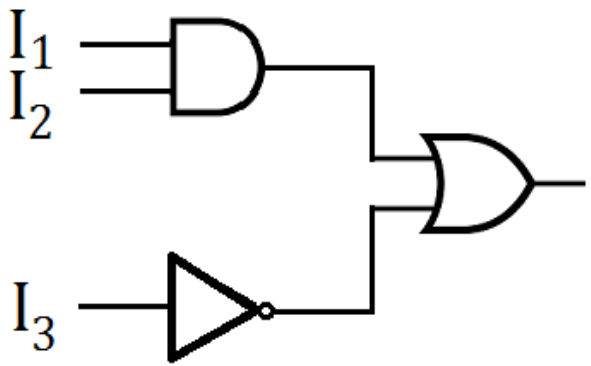
Each combination represents the order that switches of circuit's inputs have effect on gate producing circuit's output.

Proposition

Let n the number of inputs, Cb the set of all combinations on n inputs and S_n the symmetric group on $X = \{1, ..., n\}$. Then there is a bijective map:

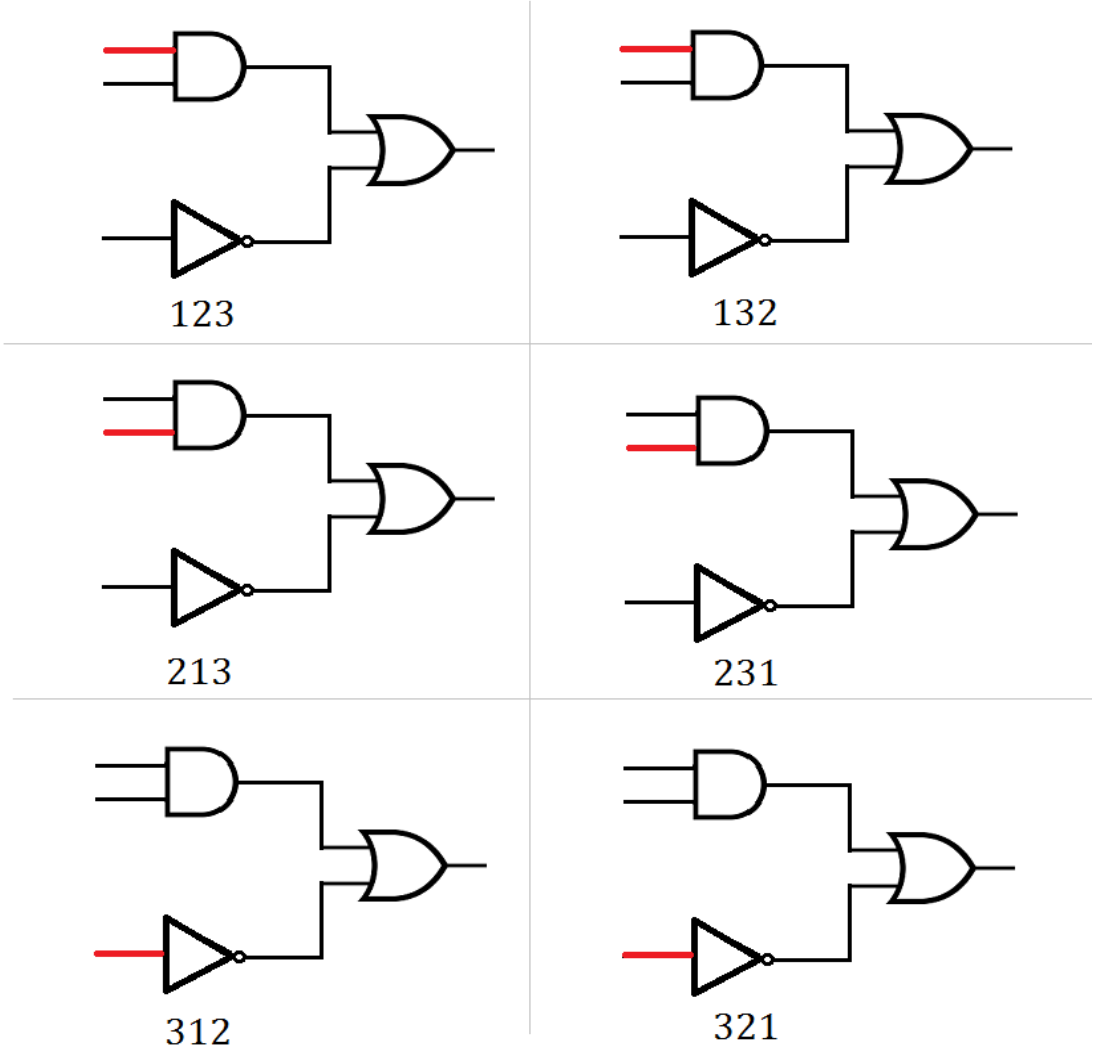
$$\iota \colon S_n \longrightarrow Cb$$

Combinations: example

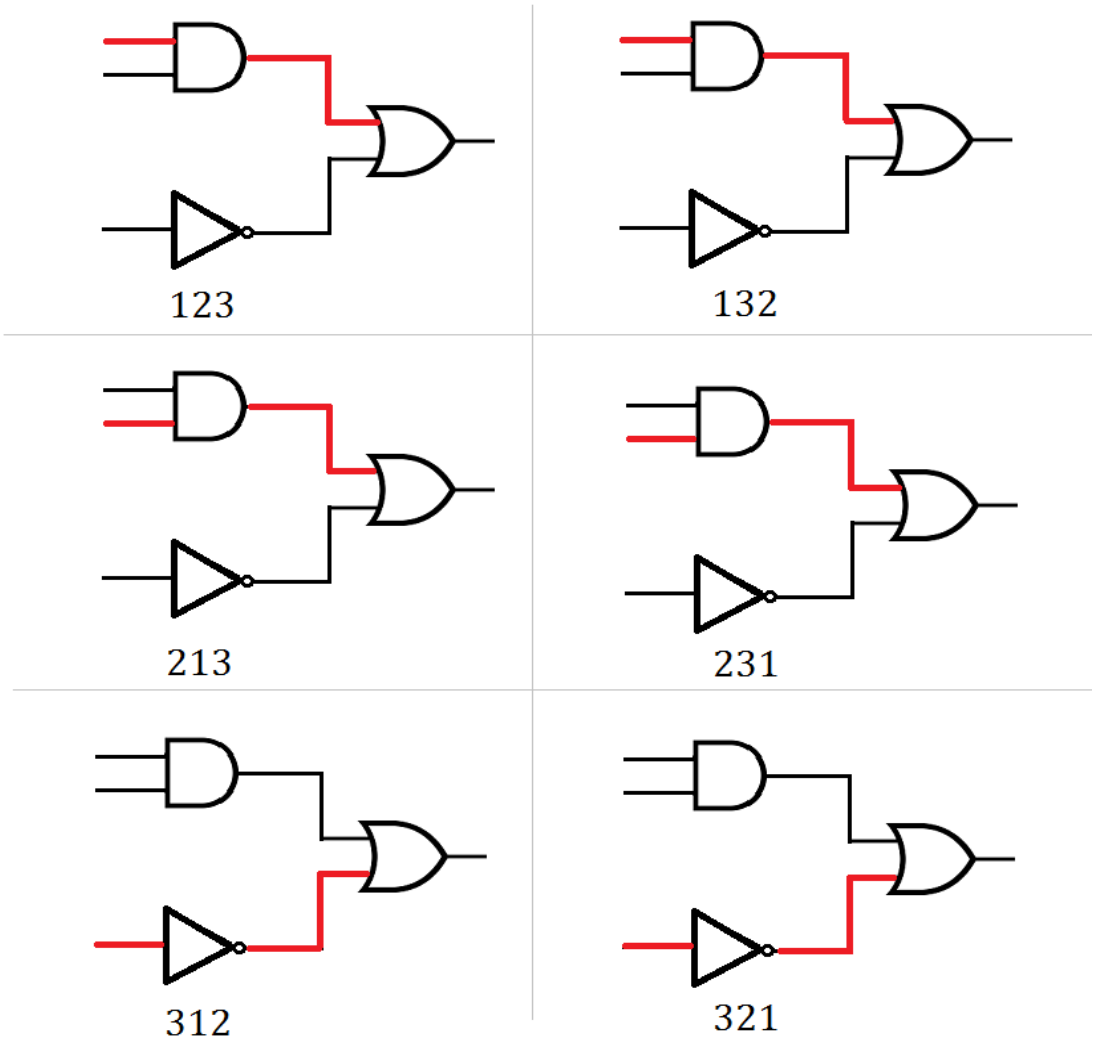


This circuit has 3 inputs, and the combinations are $3! = 6$.

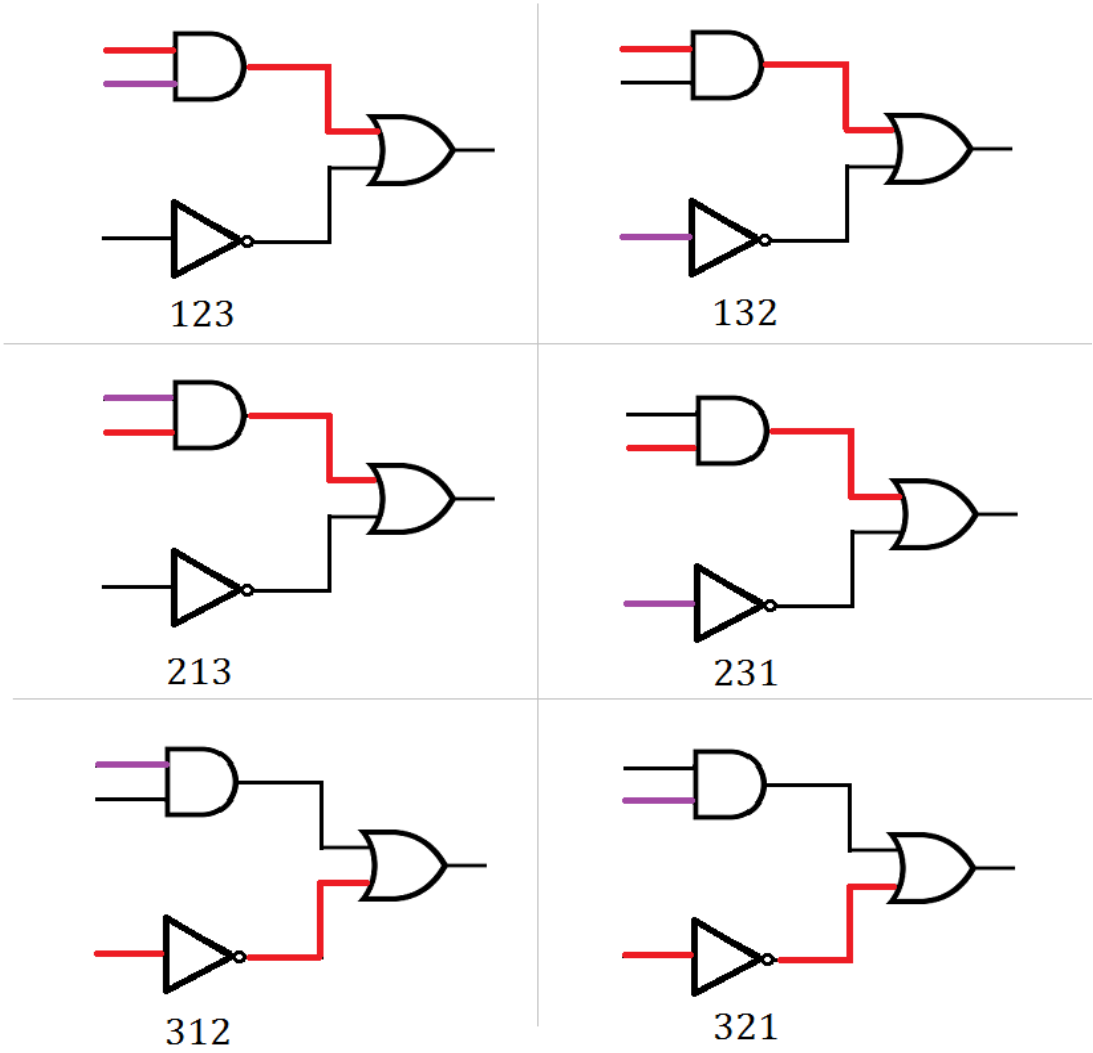
Combinations: example



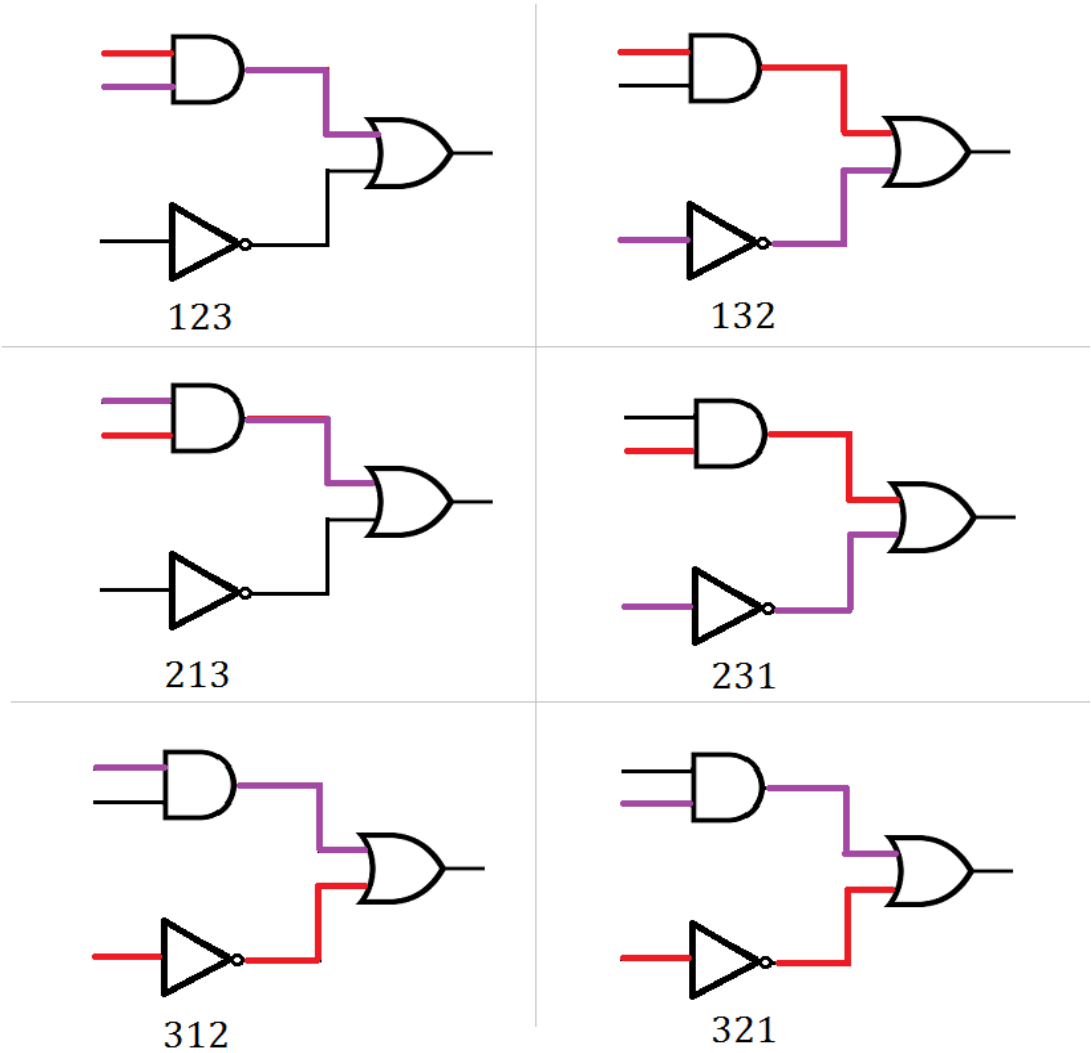
Combinations: example



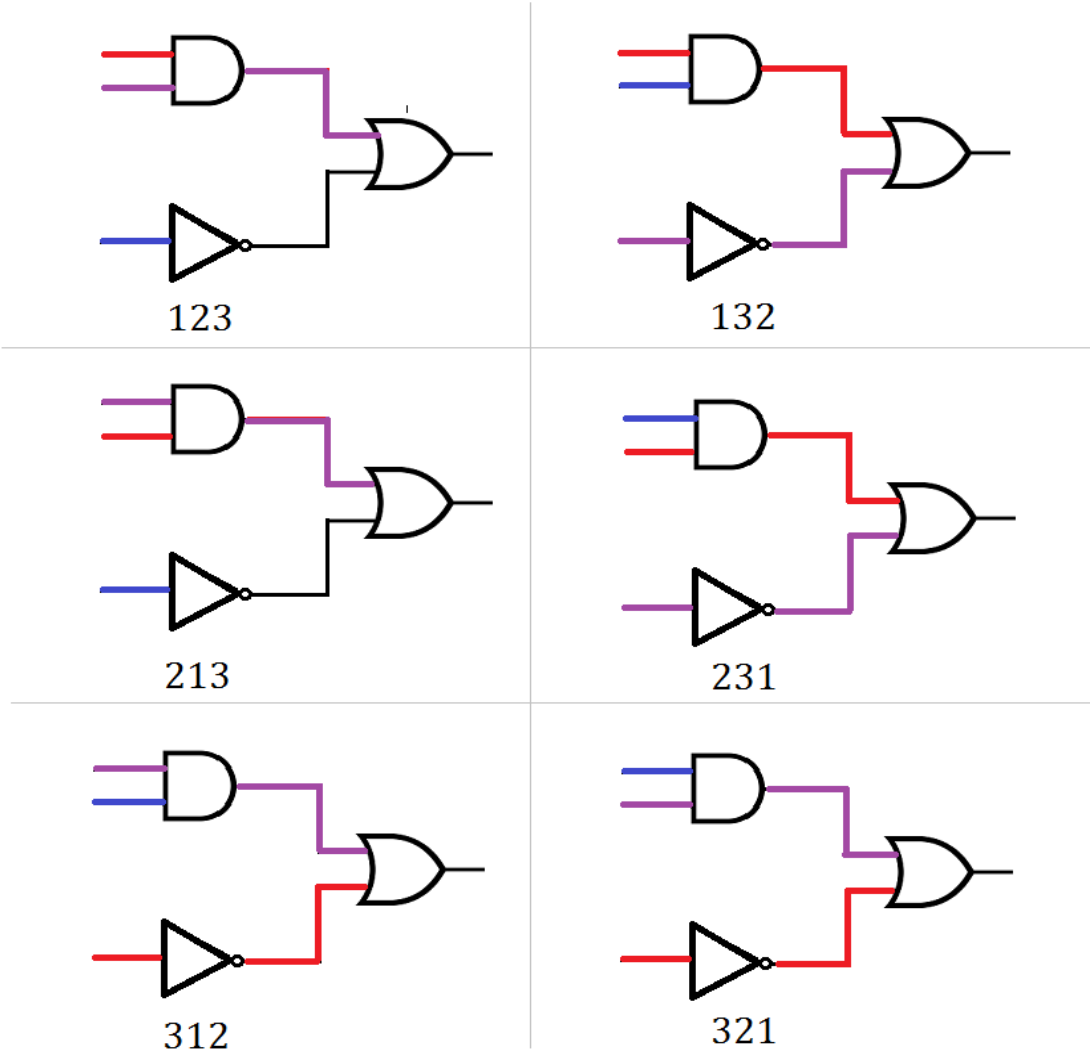
Combinations: example



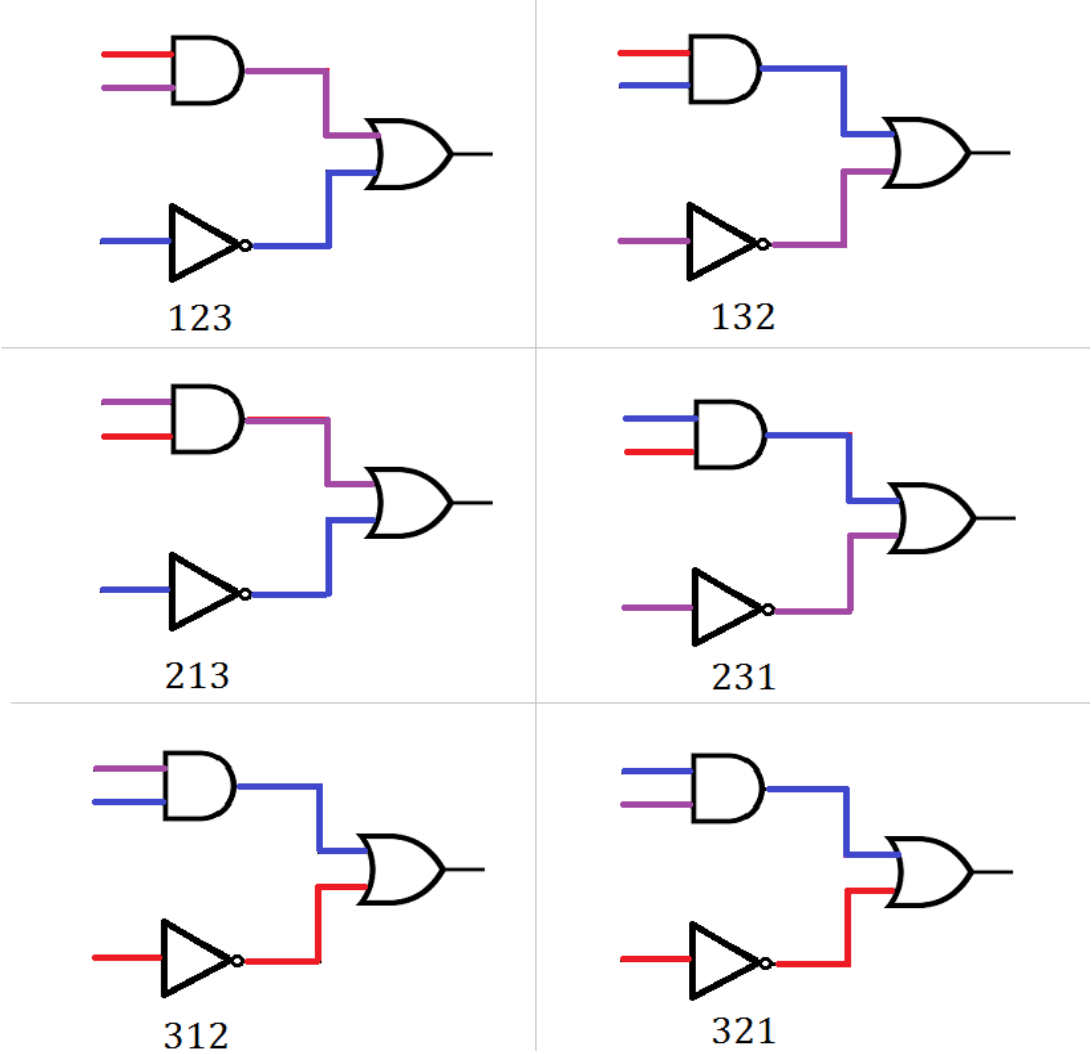
Combinations: example



Combinations: example



Combinations: example



Equivalent relation

An **equivalent relation** on Cb can be defined:

Given $c_1 \in Cb$ and $c_2 \in Cb$, c_1 is equivalent to c_2 if and only if for all sets of transients in input that describe possible switches, transient in the output gate of the circuit is always the same. Equivalent relation is denoted by $\sim_{Cb}$.

Propagation Sequences set

Let R a circuit with n inputs, and Cb the set of all combinations on it. Quotient set

$$\frac{Cb}{\sim_{Cb}}$$

is the **set of all propagation sequences in the circuit**.

A propagation sequence is an equivalent class in $\frac{Cb}{\sim_{Cb}}$, which contains all combinations on the circuit that describe always the same behaviour in glitches propagation.

Conclusion

The aim of these studies is to develop a methodology to define if a cryptographic algorithm is secure against side-channel attacks or if it has some fatal leakages caused by glitches.

- **Hazard algebra** and **LP model** consider only the worst case of glitches propagation, and then define if there can be some fatal leakages caused by glitches;
- **Propagation Sequences** describe a more realist situation, and better quantify the amount of critical leakages in a circuit.