Intrinsic **ID SRAM PUF Technology & Solutions: Physically Unclonable Function**

Intrinsic ID delivers strong, device-unique data security and authentication solutions for the connected world. These authentication solutions are based on Intrinsic ID's patented SRAM (Static Random Access Memory) Physical Unclonable Function or SRAM PUF technology.
Using this technology, security keys and unique identifiers can be extracted from the innate characteristics of each semiconductor. Similar to biometrics measures, these identifiers cannot be cloned, guessed, stolen or shared. Keys are generated only when required and don't remain stored on the system, hence providing the highest level of protection.
Our SRAM PUF-based security solutions are very suitable for applications such as secure key generation and storage, device authentication, flexible key provisioning and chip asset management. They can be used to secure payments, to protect highly sensitive data, for anti-counterfeiting and anti-cloning, to prevent identity theft, piracy of media content and software apps, software reverse engineering, and more.
Intrinsic ID's security solutions are available as hard and soft Intellectual Property (IP) and are used by companies who want a proven, easy and cost-efficient way to provide a solid trust base within their devices and applications.

From <https://www.intrinsic-id.com/sram-puf-technology-solutions/>

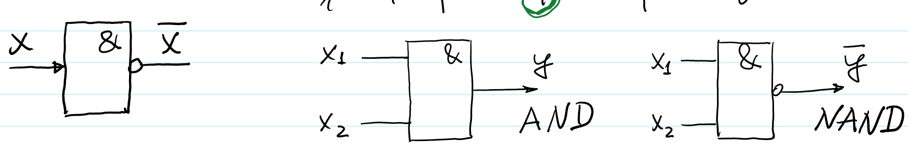
Principle of logical gates work

Logical Inversion, AND, NAND : $x \in \{0,1\}$; $x_1, x_2 \in \{0,1\}^2$

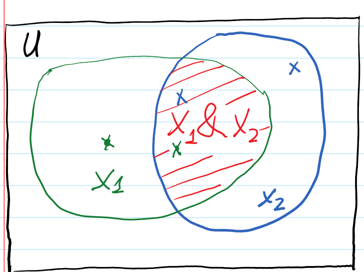
x	$\overline{x}$
0	1
1	0

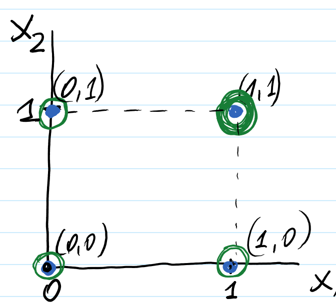
x_1	x_2	$y = x_1 \& x_2$	$\overline{y} = \overline{x_1 \& x_2} = \overline{Q}$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

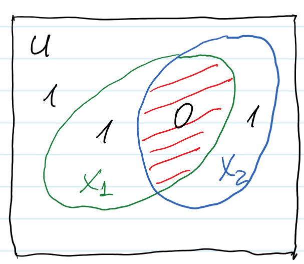
NAND allows to realiaze all Boolean functions

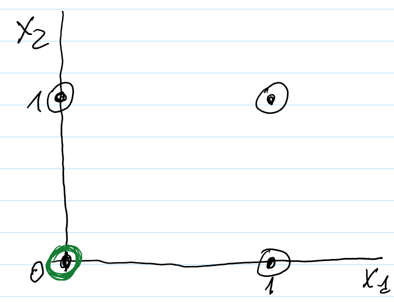


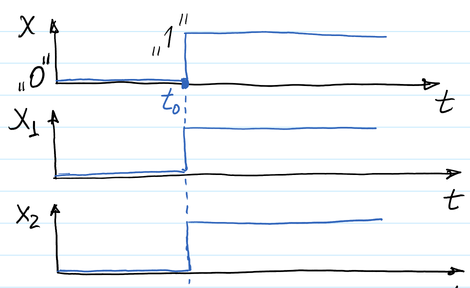
Brėž. NAND

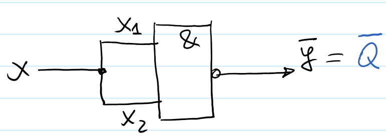




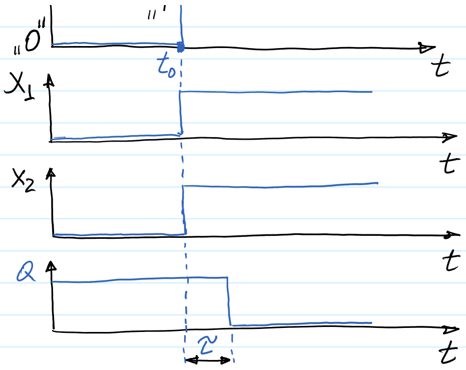


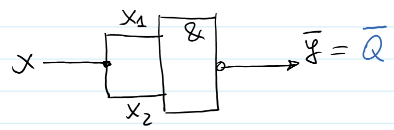






x_1	x_2	x	$\overline{y} = \overline{Q}$
0	0	0	1
1	1	1	0

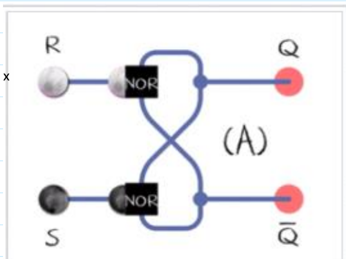




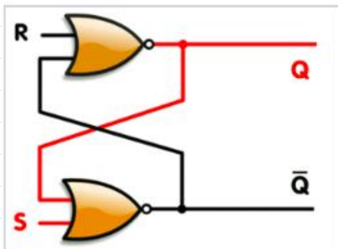
x_1	x_2	x	$\bar{y} = \bar{Q}$
0	0	0	1
1	1	1	0

Flip-flop (electronics)

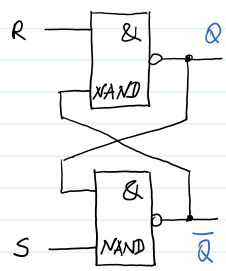
From <https://en.wikipedia.org/wiki/Flip-flop_(electronics)>



An animated SR latch. Black and white mean logical '1' and '0', respectively.
(A) S = 1, R = 0: set
(B) S = 0, R = 0: hold
(C) S = 0, R = 1: reset
(D) S = 1, R = 1: not allowed
The restricted combination (D) leads to an unstable state.

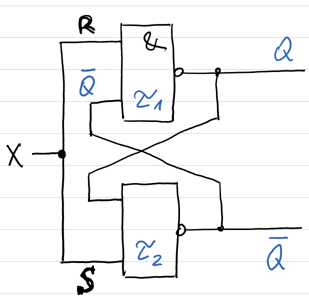


An animation of a SR latch, constructed from a pair of cross-coupled NOR gates. Red and black mean logical '1' and '0', respectively.

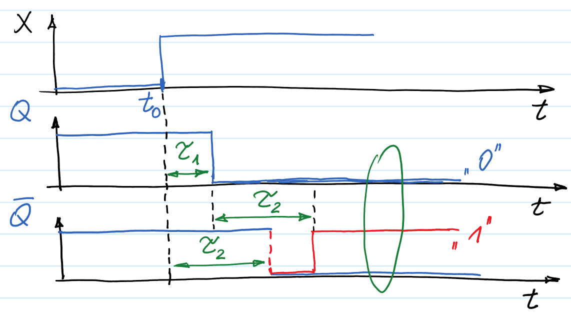


R	S	Q	$\bar{Q}$
1	0	0	1
1	1	0	1
0	1	1	0
1	1	1	0
0	0	1	1
1	1	?	?

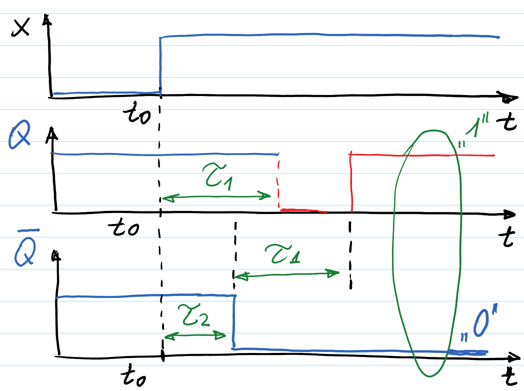
RS - Flip-flop



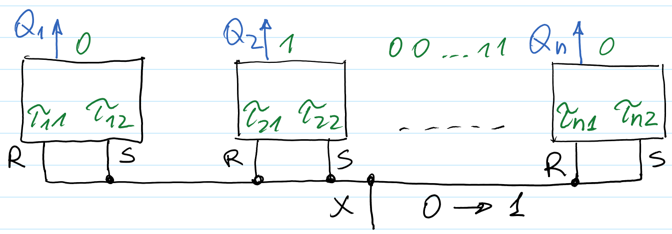
If $\tau_1 < \tau_2$



If $\tau_1 > \tau_2$



$Q_1 \uparrow 0$ $Q_2 \uparrow 1$ 0 0 .... 1 1 $Q_n \uparrow 0$ (0100 .... 11 0)

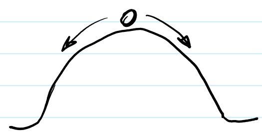


(0100....11 0)

1 cipars su n RS-trig.

$\tau_{21} \approx \tau_{22}$ $\qquad \tau_{i1} \approx \tau_{i2}$ $\qquad \tau_{j1} \approx \tau_{j2}$

$\tau_{21} < \tau_{22} \rightarrow \tau_{21} > \tau_{22}$ $\qquad \tau_{i1} > \tau_{i2} \rightarrow \tau_{i1} < \tau_{i2}$ $\qquad$ - - - -



About 15% of RS flip-flops are unstable due to $\tau_{i1} \approx \tau_{i2}$ and are sensible to randon changing of environment conditions.

Every transition of X from 0 → 1 will have 15% errors

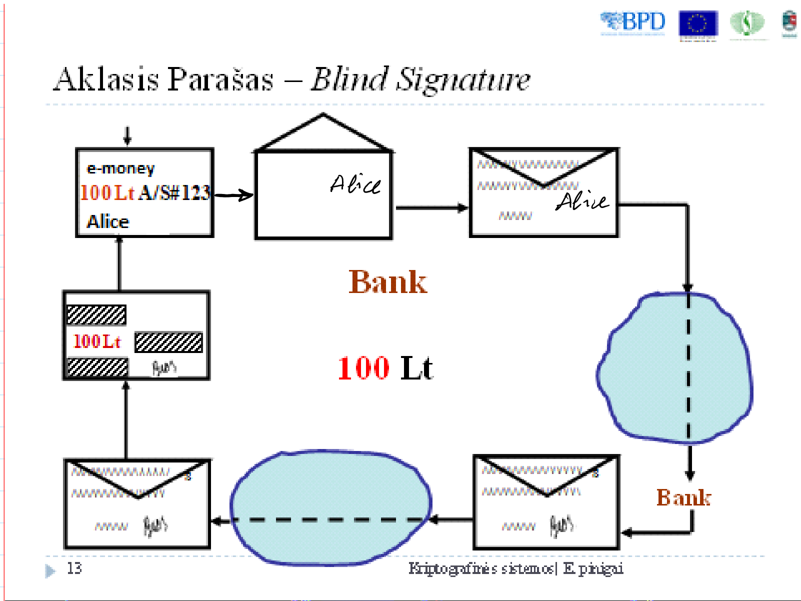
Solution is to apply Error Correcting Codes (ECC): to correct ~ 25% errors.

It is recomended to use 320 – 400 bits PUF.

Using 400 bits PUF we have 25% of correct bits values, i.e. 300 bits.

How many different PUFs can be produced and distributed?

$N_{PUFs} = 2^{300} \approx 10^{100}$; The number of Planet population will be soon 8 Mlrd $= 8 \cdot 10^{9}$.



Chaum e-money system

e-coin

RSA cryptosystem

B: $p, q \leftarrow$ genprime

$n = p \cdot q$

$\phi = (p-1)\cdot(q-1)$ $\quad PuK = (n, e)$

$\left.\begin{array}{l} e = 2^{16}+1 \\ d = e^{-1} \bmod \phi \end{array}\right\} \Rightarrow \begin{array}{l} e\,d = 1 \bmod \phi \\ PrK = d \end{array}$

If $e = 2^{16}+1$ – it is prime

1) $1 < e < \phi$

2) $gcd(e, \phi) = 1$ since e is prime

>> d = mulinv(e, fy) $\quad$ % fy = ϕ

Since numbers e and d are presented in exponent, then exponent value is computed mod ϕ according to Euler theorem:

If $gcd(z, n) = 1 \Rightarrow z^{\phi} \bmod n = 1$

Any computations performed in the exponent are computed mod ϕ:

$$z^{e\cdot d} \bmod n = z^{e\cdot d \bmod \phi} \bmod n = z^{1} \bmod n = z$$

if $z < n$

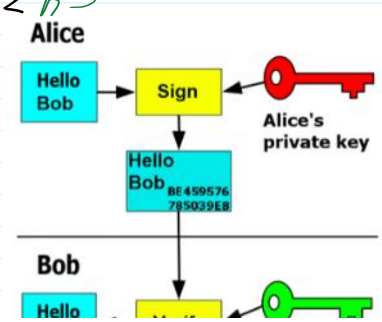
RSA signature creation:

On message M encoded by decimal number $m < n$.

$Sign(PrK = d, m) = \sigma = m^{d} \bmod n$.

RSA signature verification:

$Ver(PuK = (e, n), \sigma) = \sigma^{e} \bmod n = m$

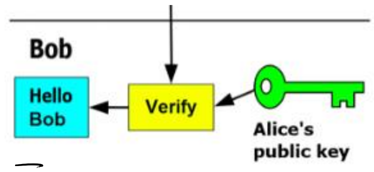


RSA signature verification:

$Ver(PuK=(e,n), \sigma) = \sigma^e \bmod n = m.$

Correctness: $\sigma^e \bmod n = (m^d)^e \bmod n = m^{\overbrace{de \bmod \phi}^{=1}} \bmod n =$

$= m \bmod n \xlongequal[\text{if } m<n]{} m$



A: $PrK_A = d_A$ $\quad$ $PuK_A = (n_A, e)$ $\qquad \xleftarrow{PuK=(n,e)}$ $\qquad B$: $PrK = d$, $\quad PuK = (n, e)$.

A: $m = 100$; is masking value m:

$t \leftarrow randi$; $1 < t < n$: $gcd(t, n) = 1 \Rightarrow \exists! \; t^{-1} \bmod n$.

$m' = m \cdot t^e \bmod n \xrightarrow{\quad m' \quad}$ $\quad B$:

$Ver(PuK=(n,e), \sigma', m') = m' \xleftarrow{\quad \sigma' \quad}$ $\quad Sign(PrK=d, m') = \sigma'$

$\sigma' = (m')^d \bmod n =$

$= (m \cdot t^e)^d \bmod n =$

$= m^d \cdot t^{\overbrace{ed \bmod \phi}^{\to 1}} \bmod n =$

$\xleftarrow{\sigma' = m^d \cdot t \bmod n}$ $\quad = m^d \cdot t \bmod n$

A: unmasks signed m'

$(\sigma')^e \bmod n = ((m')^d)^e \bmod n = (m')^{\overbrace{ed \bmod \phi}^{=1}} \bmod n =$

$= m' \bmod n \xlongequal[\text{if } m'<n]{} m' \Rightarrow$ Signature is valid. = True

A: wants to find a valid signature σ of B on $m = 100$:

$\sigma = m^d \bmod n$

A extracts (unmasks) $m^d \bmod n = \sigma$ from σ':

$\sigma' \cdot t^{-1} \bmod n \longrightarrow$ if $gcd(t, n) = 1 \Rightarrow t^{-1} \bmod n$ exists.

$\sigma' \cdot t^{-1} \bmod n = m^d \cdot \cancel{t} \cdot \cancel{t^{-1}} \bmod n = m^d \bmod n = \sigma.$

But $m^d \bmod n$ – is a B's signature on the actual amount of money $m = 100$.

$\sigma = m^d \bmod n$. $\qquad$ $PuK = (n, e)$ B's

A: (m, σ) $\qquad (m, \sigma)$ $\qquad V$: verifies is B's signature

$\sigma = m^d \bmod n.$ $PuK=(n,e)$ B's

A: (m, σ) $\xrightarrow[\text{to the Vendor } V]{(m,\sigma)}$ V: verifies is B's signature on the money amount $m = 100$ is true

$Ver(PuK=(n,e), \sigma, m) = True$

$\sigma^e \bmod n = (m^d)^e \bmod n = m^{de \bmod \phi} \bmod n = m \bmod n = m$ if $m < n$

E-coin properties.

1.**Anonimity**.

2.**Untraceability**.

3.**Double-spending prevention**.

4.**Divisibility**.

Chaum

Divisible coins (e-money) are growing is size.

A: $\xrightarrow{(m,\sigma), AD_1}$ V_1 $\xrightarrow{(m,\sigma), AD_1, AD_2}$ V_2 ———

$\underbrace{(m,\sigma), AD_1, AD_2, AD_3}_{\text{growing in size}}$ $\rightarrow$ V_3 - - - - - -

Property: the only customer **Alice** can create and is responsible for Random Identification String - RIS during the Withdrawal protocol.

Questions:

1.Is it possible for **Alice** to modify e-coin $\prod$.

1.How vendor **Victor** can cheat against **Bank** and how it is prevented?

E-coin properties.

1.**Anonimity**.

2.**Untraceability**.

3.**Double-spending prevention**.

4.**Divisibility**.

International Association for Cryptographic Research - IACR Barcelona, 2008, announced results:

1.Divisible e-money can be trully anonymous.

2.Divisible and trully anonymous e-money grow in size during their transfers.