Cripto Hardware com FPGA

como ir de do 0 ao 1

Luckas Farias *(Judocka)*

luckas.judocka@gmail.com

22 de Outubro de 2016

Cronograma

- Motivação
- Contextualização
- Criptografia
- Aritmética sobre Corpos Finitos
- Hardware!
- AES
- Módulos em Corpos Binários
- Precauções
- Ataques conhecidos e vulnerabilidades
- Extras

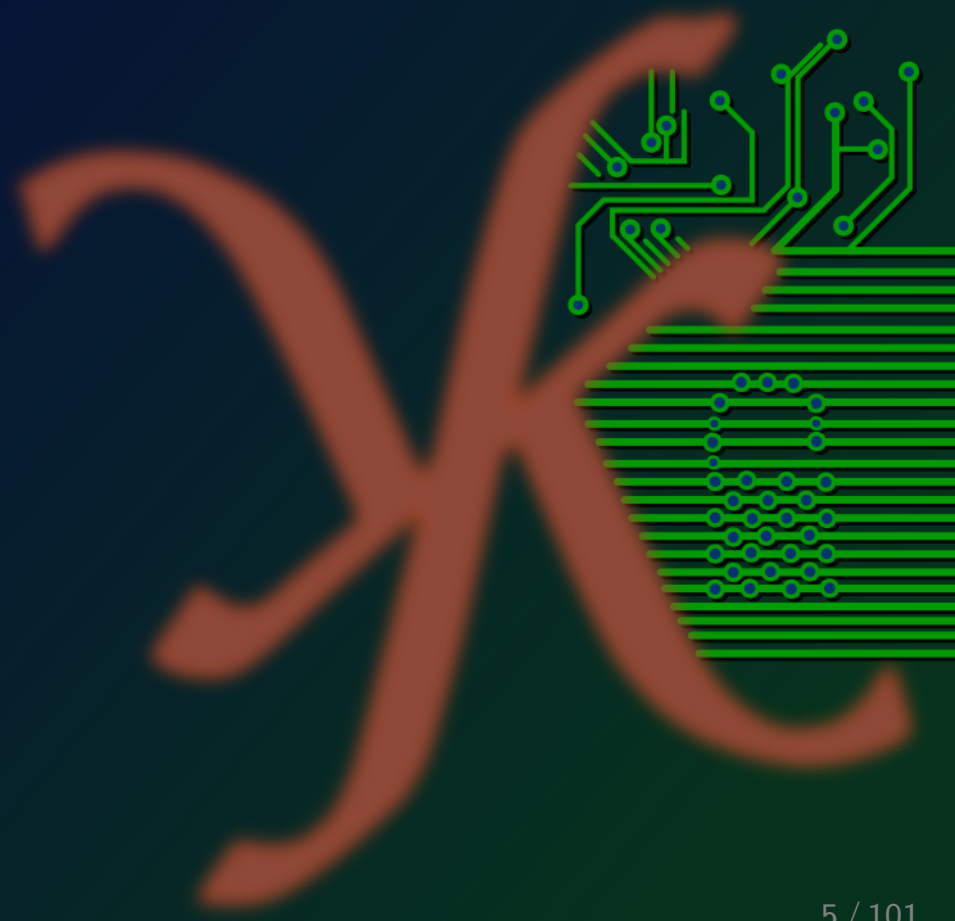
Luckas, quem é você?

- Conhecido como Judocka
- Formado em Ciência da Computação - UEL
- Mestrado em Engenharia da Computação - USP
- Professor Universitario - UAM
- Escoteiro
- Membro IEEE
  - Young Professionals
  - Mentor para ramos estudantis
- Geek em eventos:
  - Latinoware
  - CPBR
  - BSideSP
  - CryptoRave
  - Mind The Sec
  - RoadSec
- Engenheiro de coração

Sumário

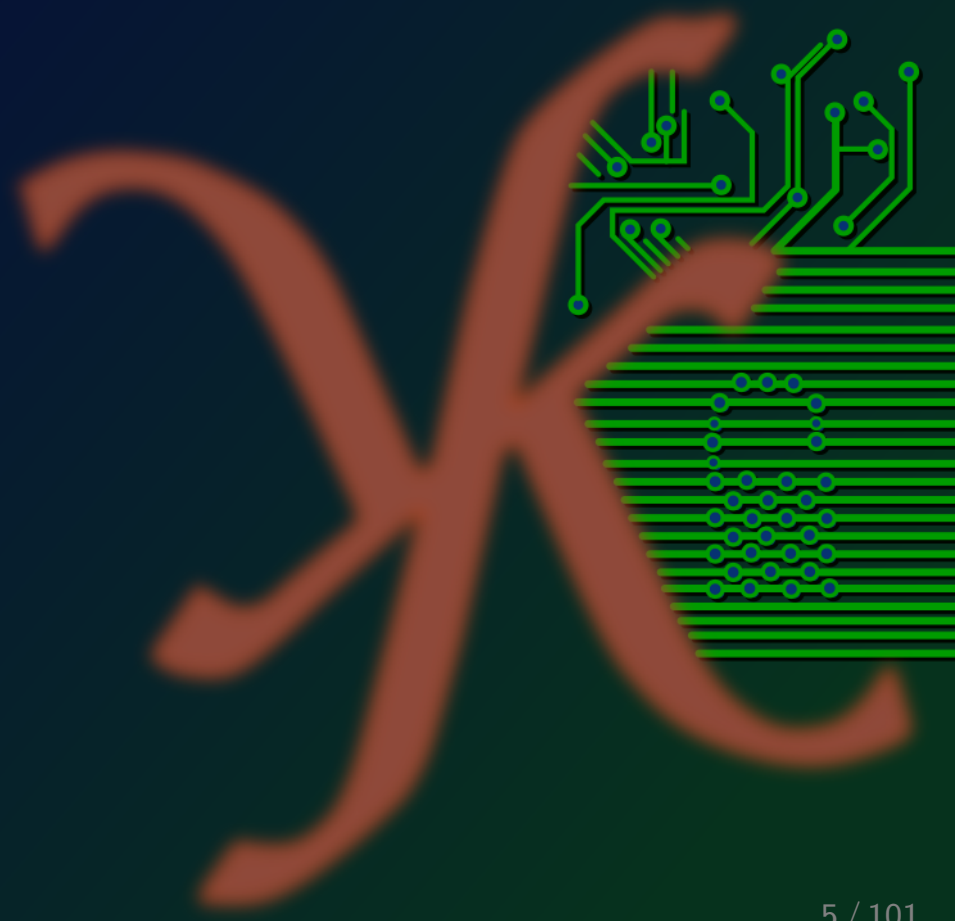
Custos da criptografia?

- Tempo?



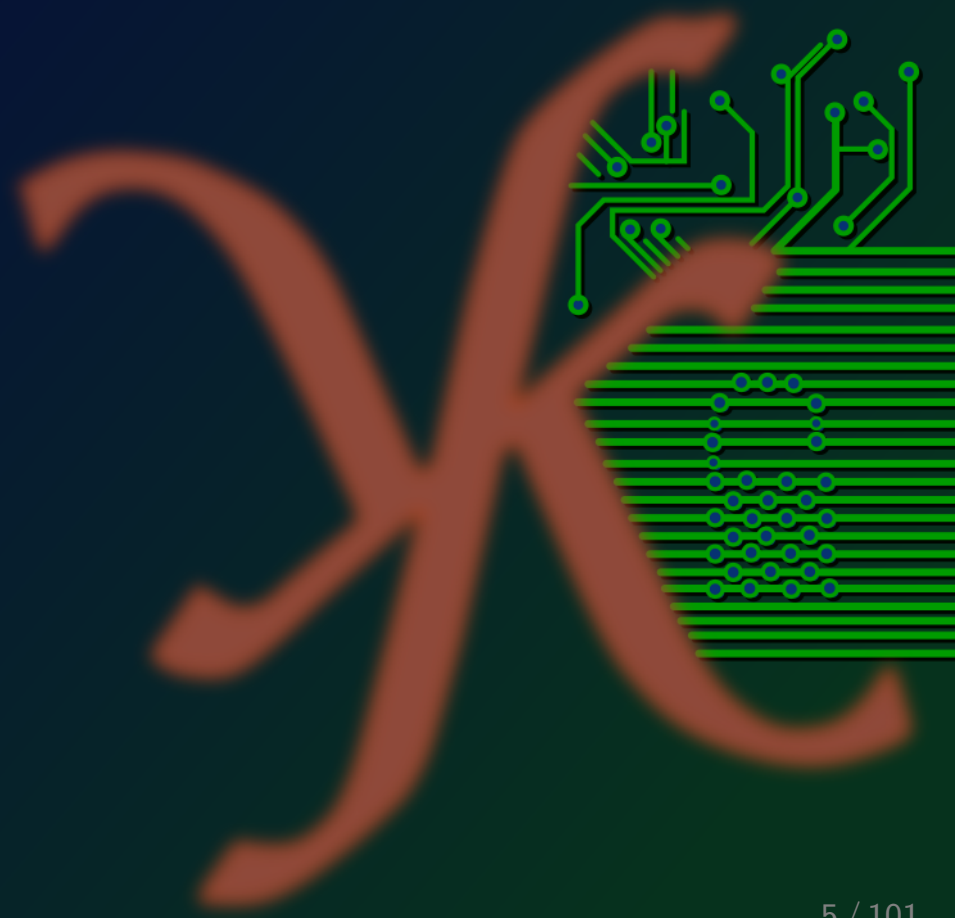
Custos da criptografia?

- Tempo?
- Processamento?



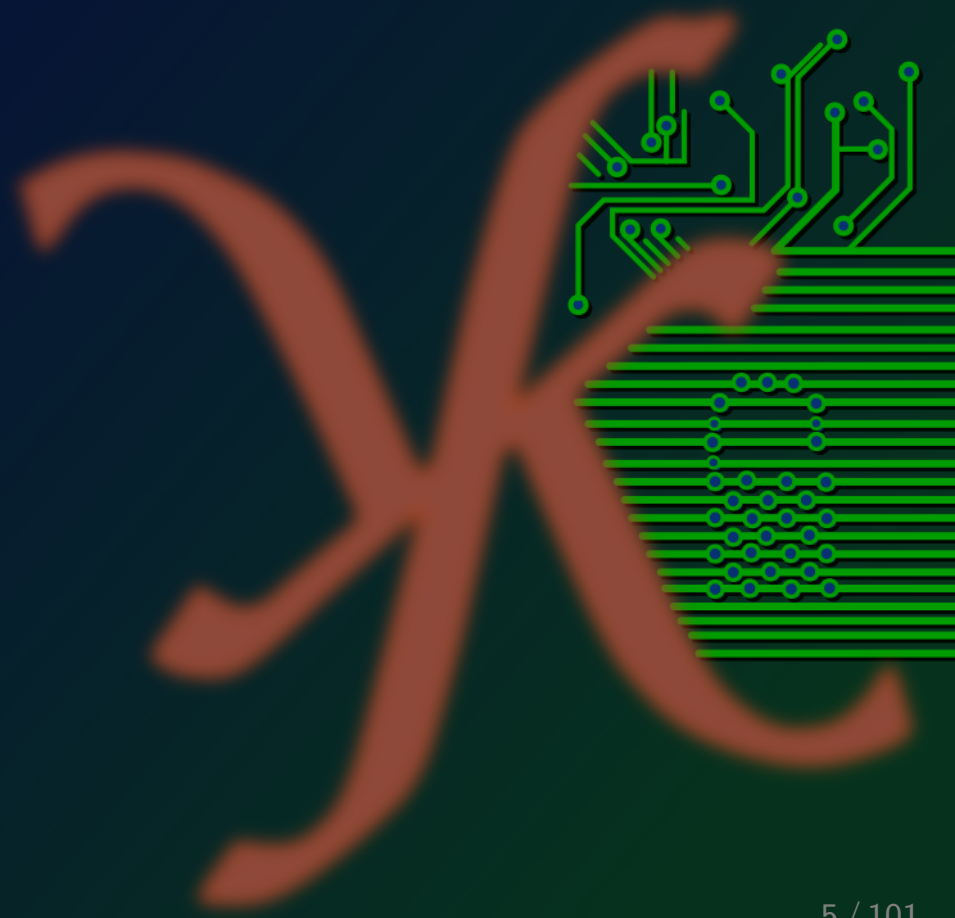
Custos da criptografia?

- Tempo?
- Processamento?
- Conhecimento?



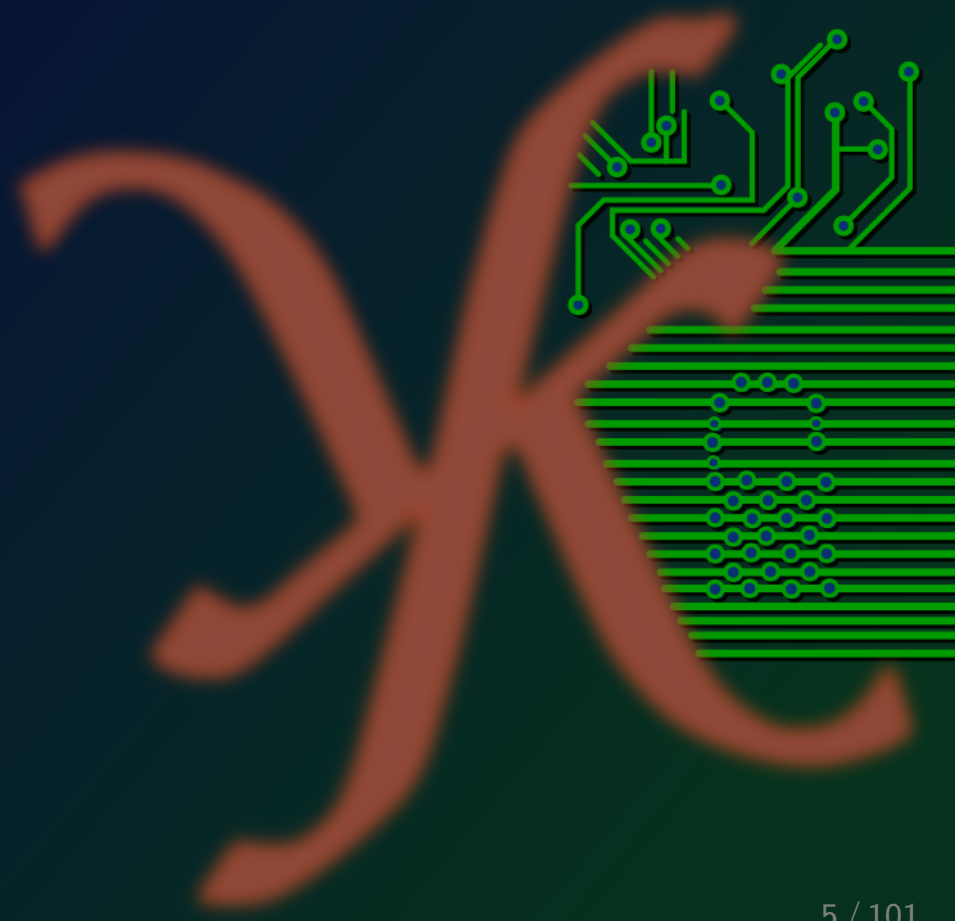
Custos da criptografia?

- Tempo?
- Processamento?
- Conhecimento?
- Otimizações?



Custos da criptografia?

- Tempo?
- Processamento?
- Conhecimento?
- Otimizações?
- Um pouco mais real...

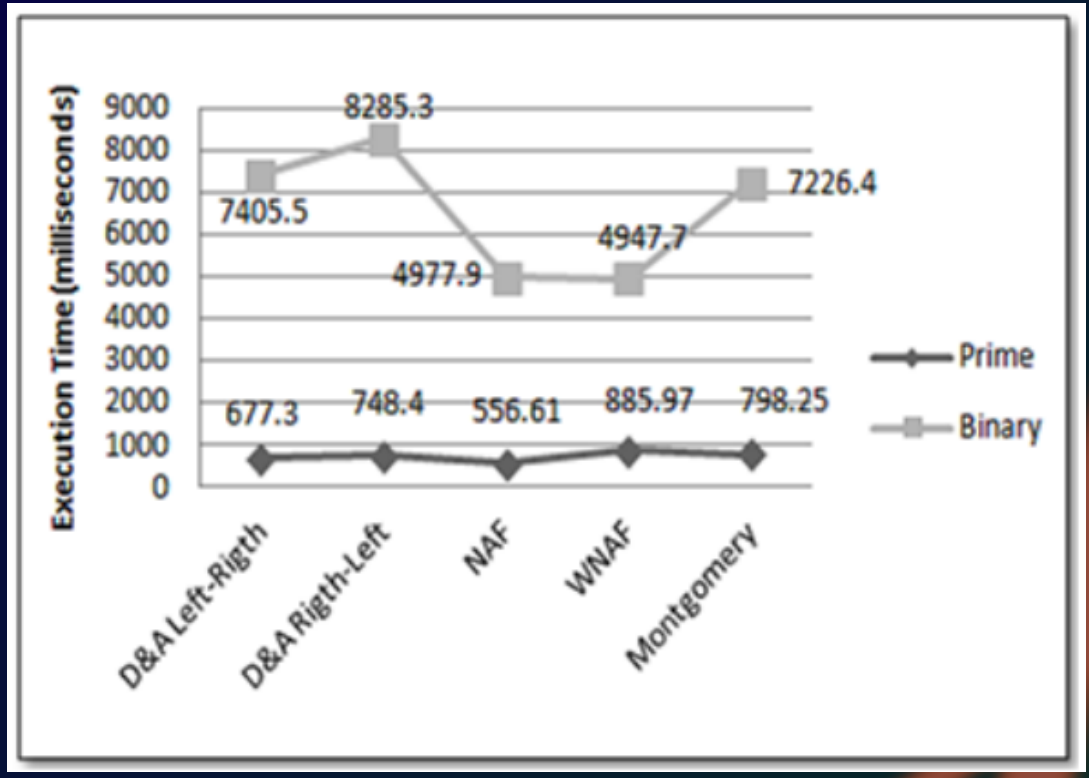


MICAz (ATMega128)

Corpo	Trabalho	Tempo de execução (s)
Binário	[Malan et al. 2004]	34
	[Yan and Shi 2006]	13.9
	[Eberle et al. 2005]	4.14
	[Szczechowiak et al. 2008]	2.16
	[Seo et al. 2008]	1.14
Primo	[Wang and Li 2006]	1.35
	[Szczechowiak et al. 2008]	1.27
	[Gura et al. 2004]	0.81
	[Uhsadel et al. 2007]	0.76
	[Großschädl 2006]	0.745

Artigo: *"Implementação eficiente de criptografia de curvas elipticas em sensores sem fio"*

ARM processor at 600MHz

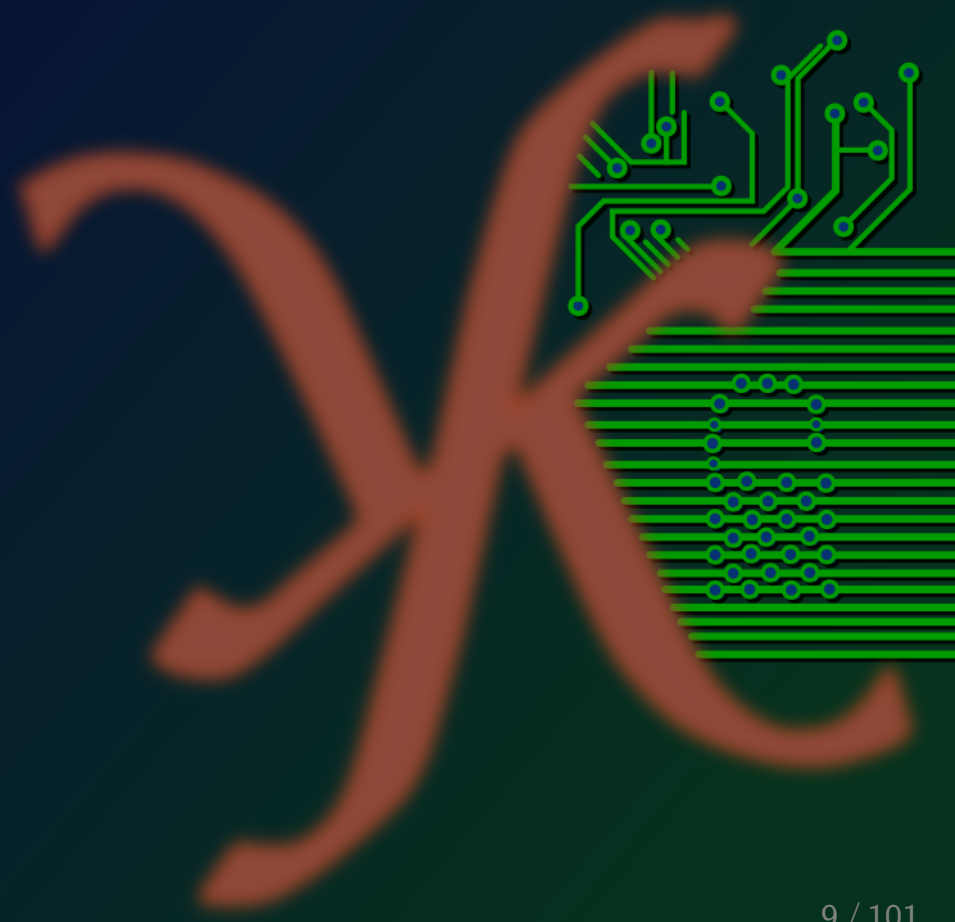


Artigo: *"A Performance Comparison of Elliptic Curve Scalar Multiplication Algorithms on Smartphones"*

Soluções da Literatura

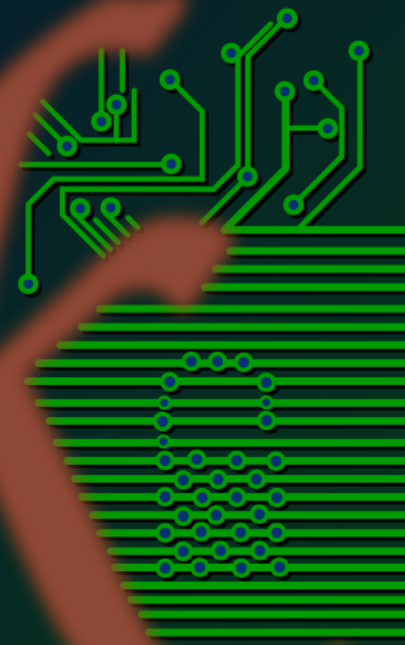
Board	Curve	Cycles	Time
8-bit processor Atmega 128	163-bit ECC	111.183.513	13.9 segundos
Dispositivos MICAz (ATMega128)	sect163k1	7.022.289	0.95 segundos
Dispositivos MICAz (ATMega128)	sect163k1	5.100.400	0.69 segundos
Texas Instrument MSP430	ECC 160-bit key	5.400.000	
8-bit processor Atmega 128	ECC 163-bit key	3.930.000	(GLV)
8-bit processor Atmega 128	ECC 163-bit key	5.945.000	(Montgomery)

Criptografia em Hardware



Criptografia em Hardware

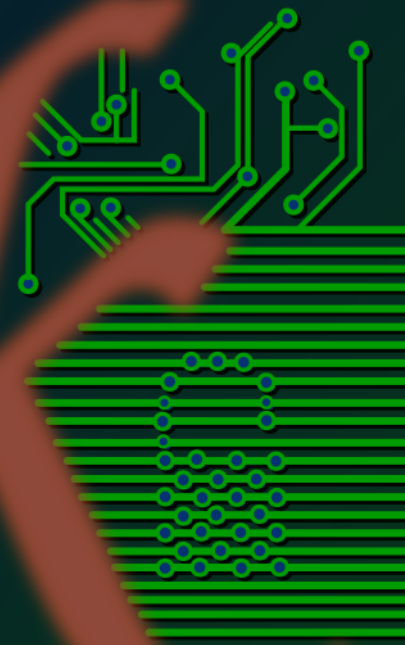
FPGA



Custo de operações em ECC

Eberle

NIST curves	Curvas genéricas
F_{163}	F_{163}
62% Multiplicação	81% Multiplicação

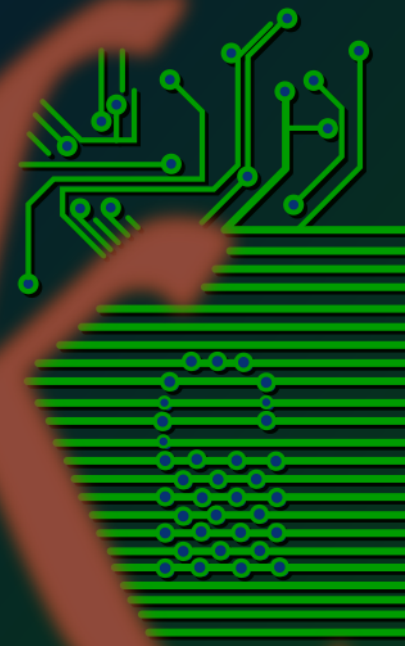


Visualmente

Paralelização da Arquitetura

Eberle

NIST curves	Curvas genéricas
F_{163}	F_{163}
62% Multiplicação	81% Multiplicação
36% com Hardware	20% com Hardware
103.33ns	1,356.11ns



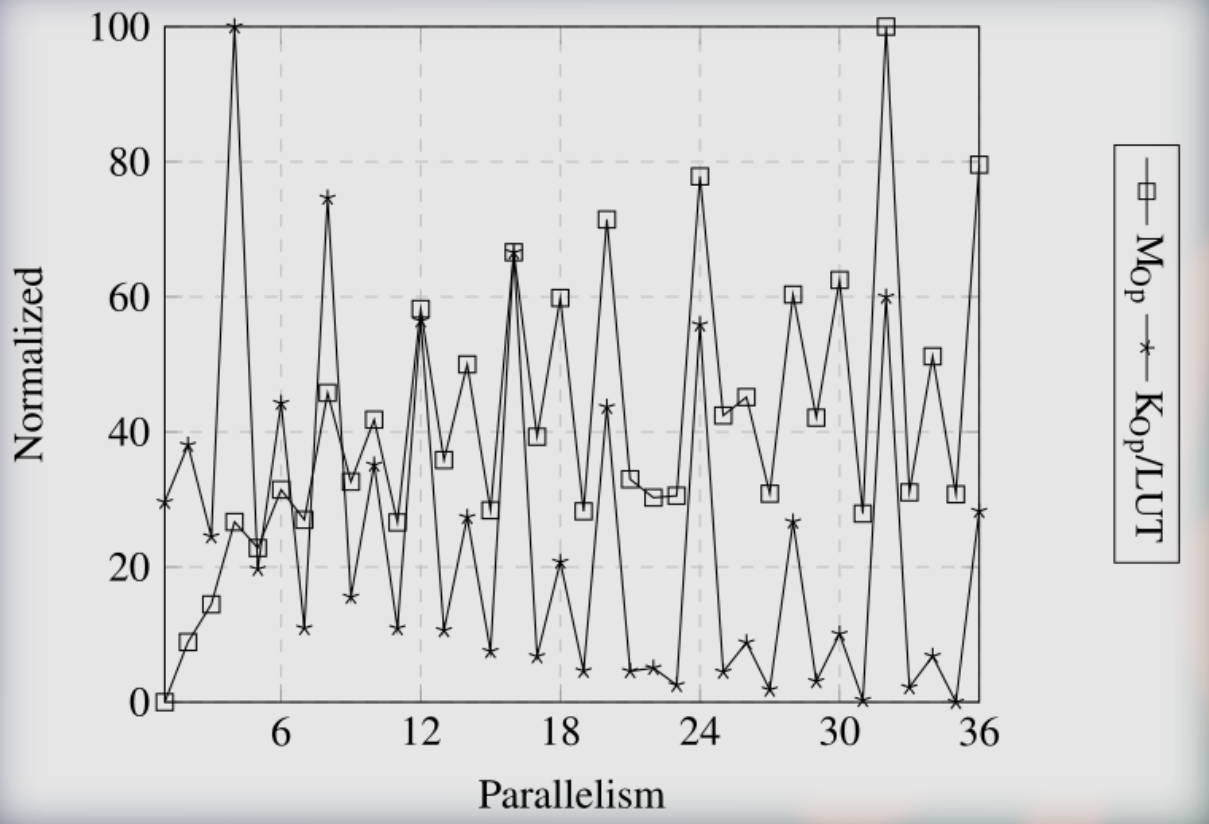
Visualmente

- Eficiência de 156,25% em curvas NIST
- Eficiência de 125,00% em curvas genéricas

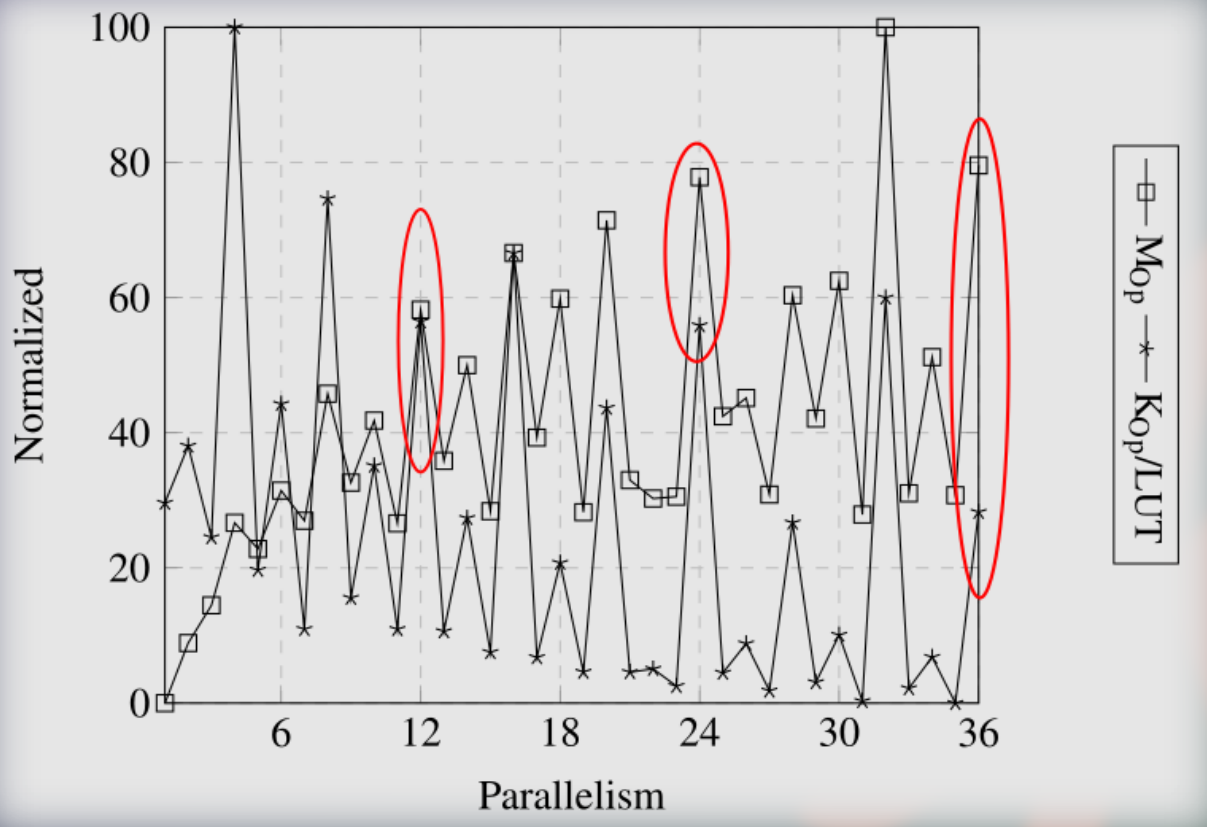
Alguns resultados de hardware

Design	Device Family	m	R/N	Slices	Clock (MHz)	Time (μs)
SIG-ECPM	Virtex-II	113	N	10686	108.3	61
	"	163	N	18079	90.2	106
	"	193	N	19250	90.2	139
	"	233	N	23020	73.6	227
Bednara [2]	Virtex	191	R	n.a.	50	2270
Gura [5][6]	Virtex-E	163	N	n.a.	66.4	143
	"	193	N	n.a.	66.4	187
	"	233	N	n.a.	66.4	225
Eberle [5]	Virtex-II	163	R	n.a.	66.4	300
	"	193	R	n.a.	66.4	420
	"	233	R	n.a.	66.4	510
Nguyen [11]	Virtex-II	233	R	13180	n.a.	2979
Orlando [12]	Virtex-E	167	R	n.a.	76.7	210

Estudo de Paralelismo F_{512}



Estudo de Paralelismo F_{512}



Exemplo

Board x Field	82	106	178	226
Zynq7020	29,371.65	19,495.66	7,336.41	5,233.57
Virtex 7 690T	29,371.65	17,626.98	6,623.95	4,501.74
DE2i-150	10,978.32	7,280.19	2,941.38	2,422.57

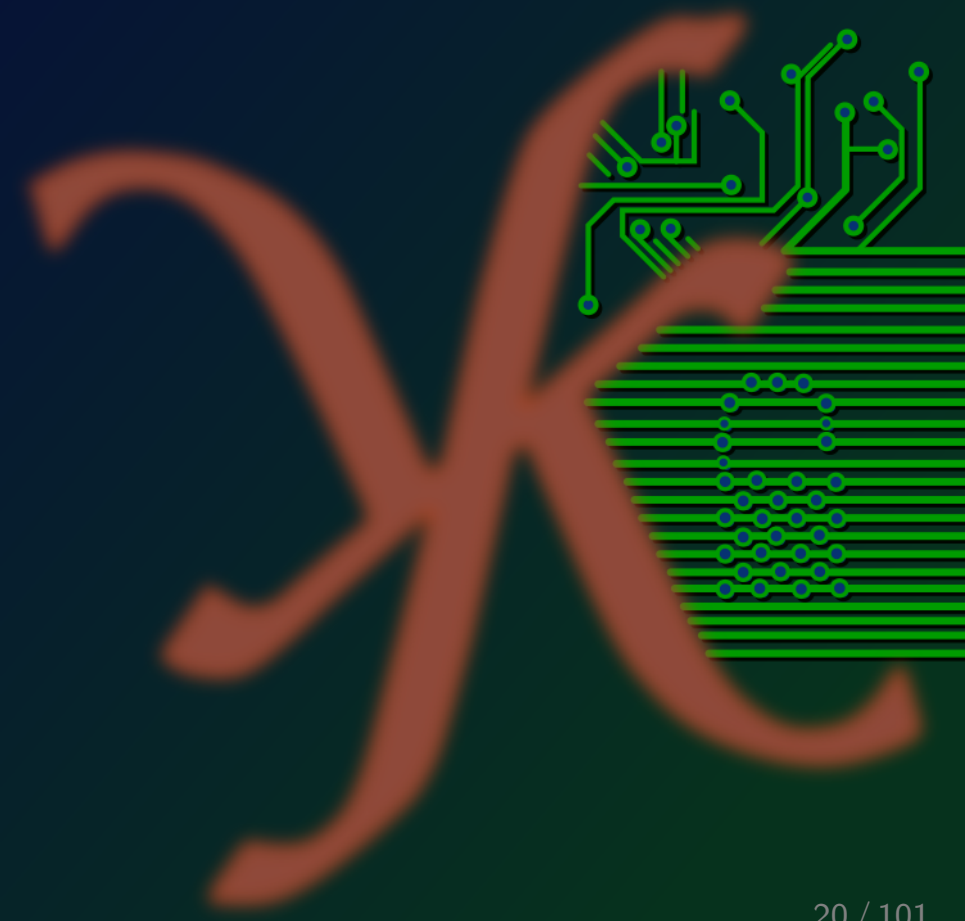
Board x Field	346	466	562
Zynq7020	1,875.18	1,462.80	984.93
Virtex 7 690T	1,636.20	1,266.70	810.26
DE2i-150	1,167.79	729.71	437.80

Software

Hardware

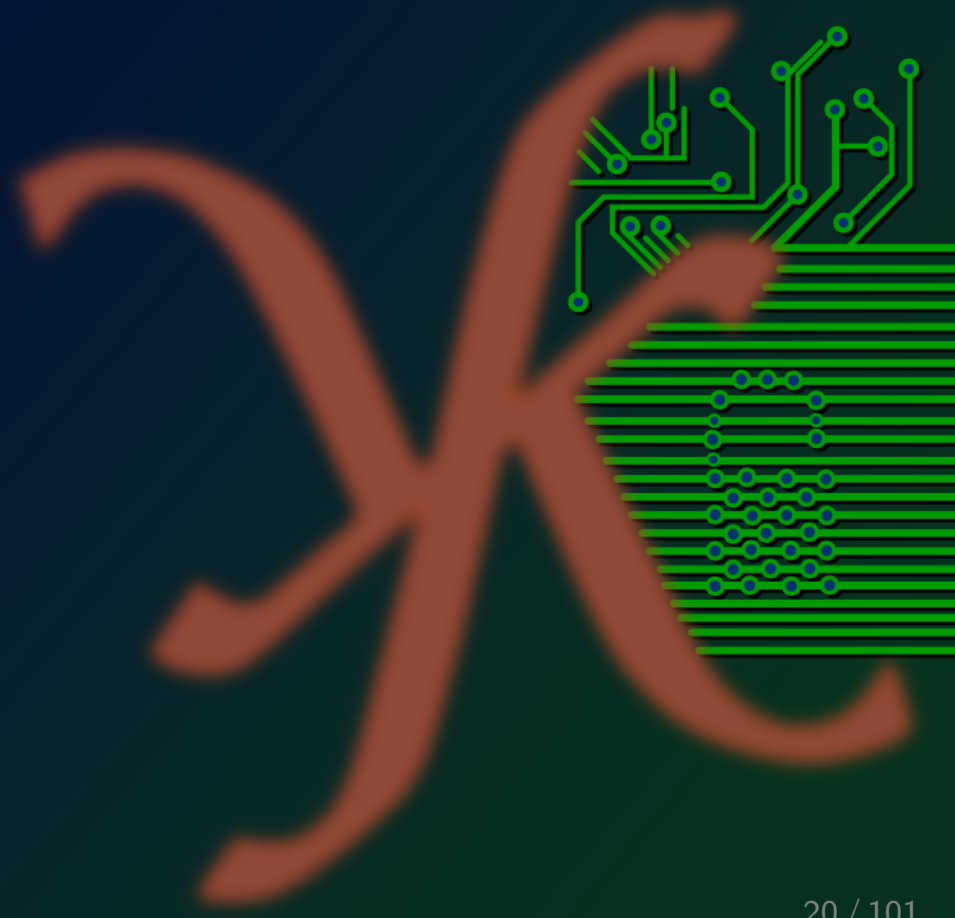
Perguntas rápidas

- Segurança?



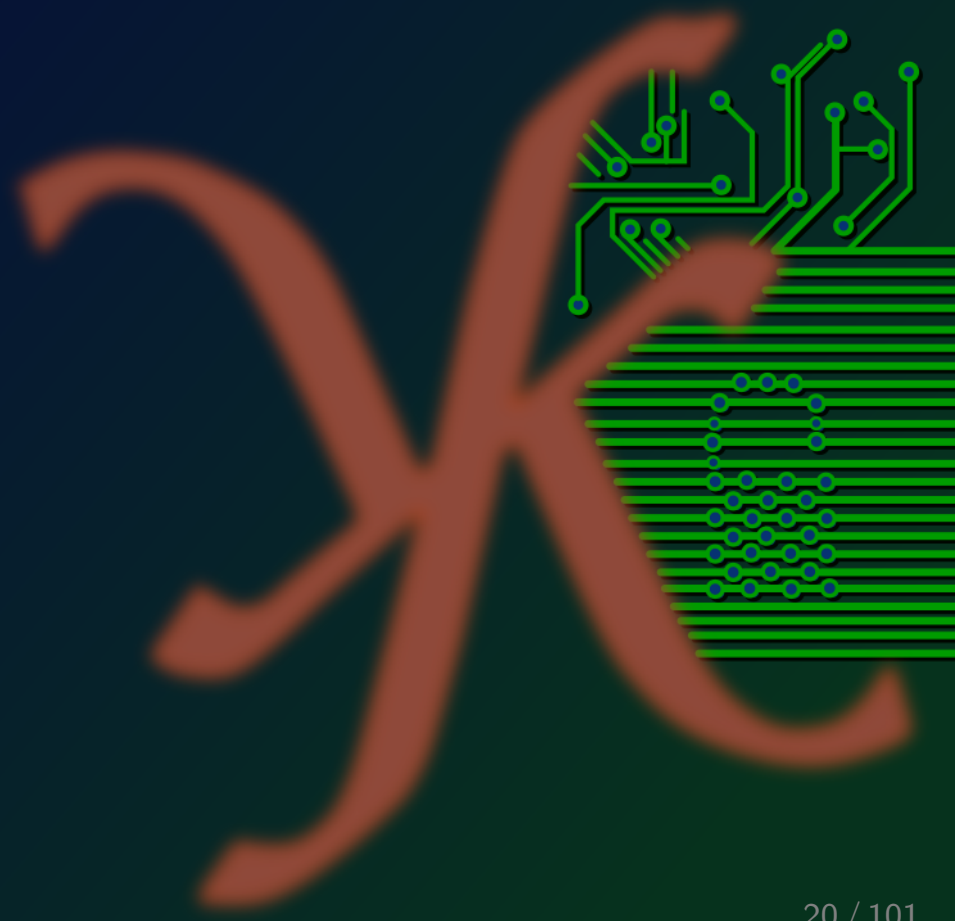
Perguntas rápidas

- Segurança?
- Criptografia?



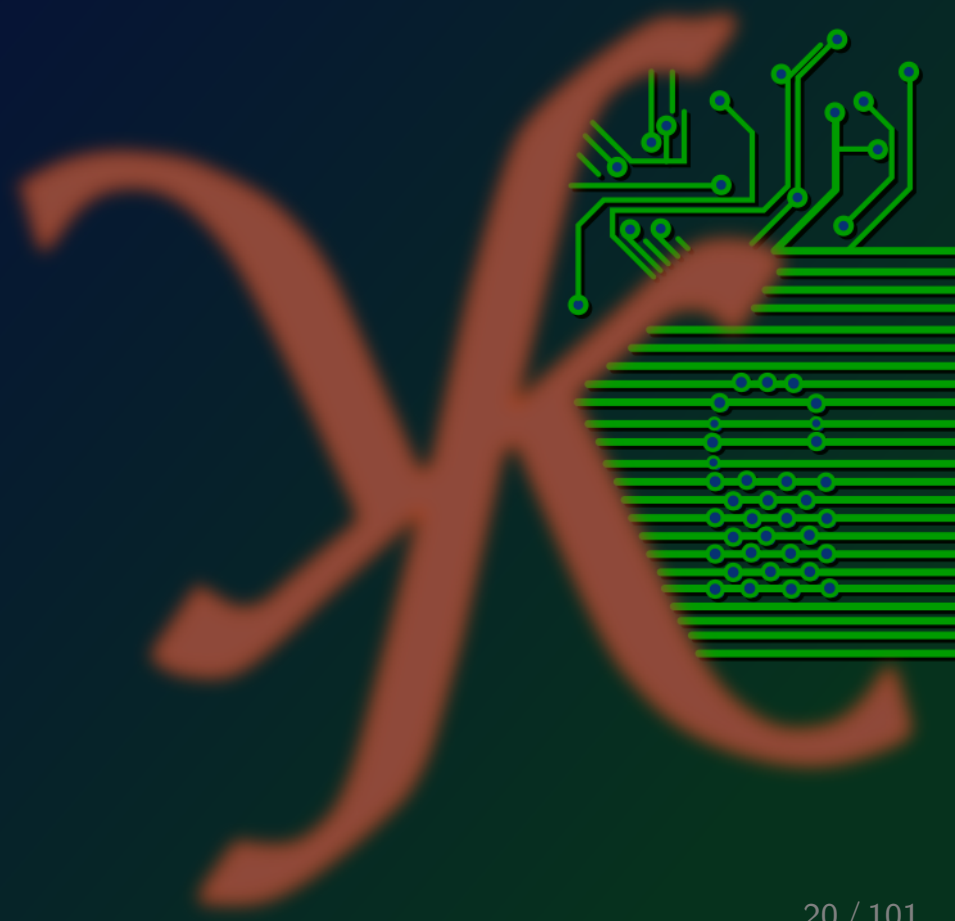
Perguntas rápidas

- Segurança?
- Criptografia?
- Simétrica e assimétrica?



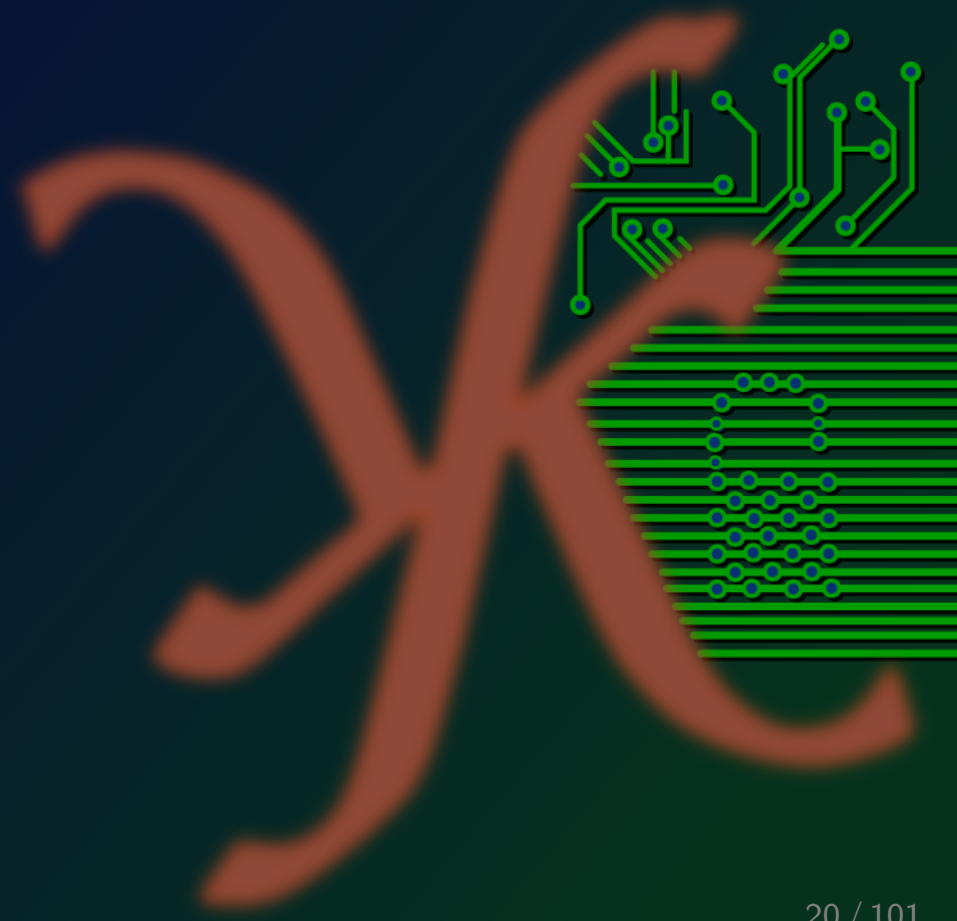
Perguntas rápidas

- Segurança?
- Criptografia?
- Simétrica e assimétrica?
- AES?



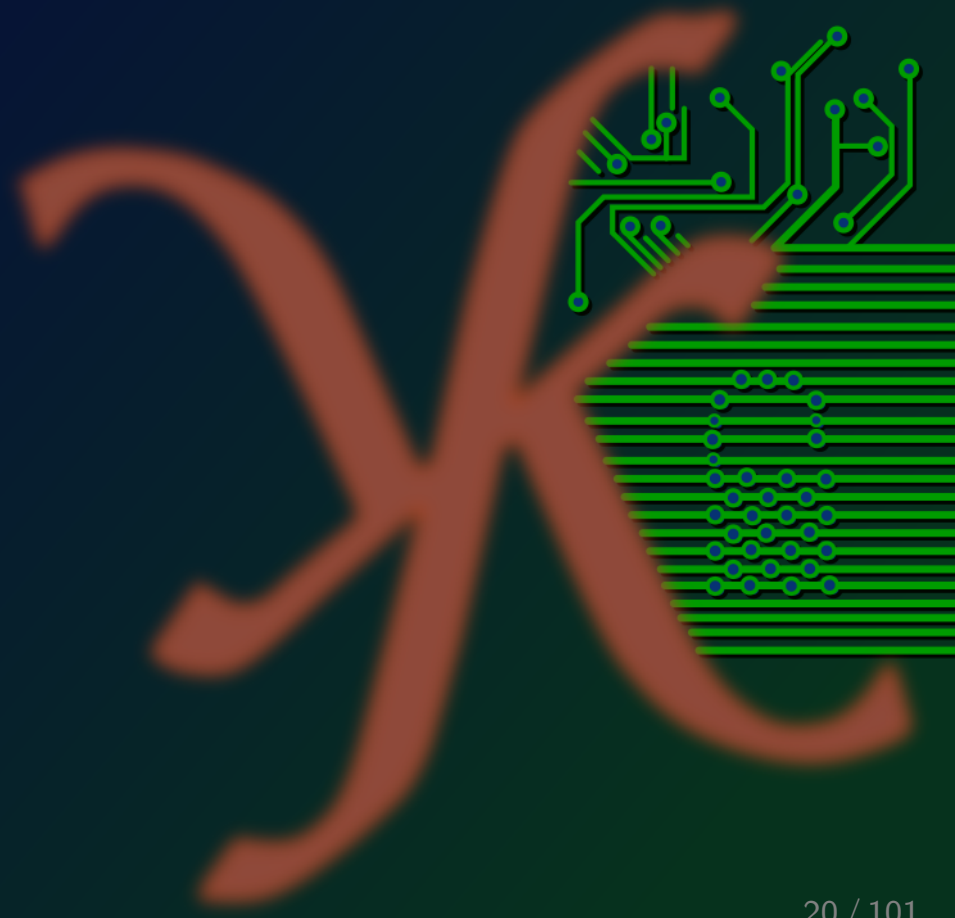
Perguntas rápidas

- Segurança?
- Criptografia?
- Simétrica e assimétrica?
- AES?
- RSA?



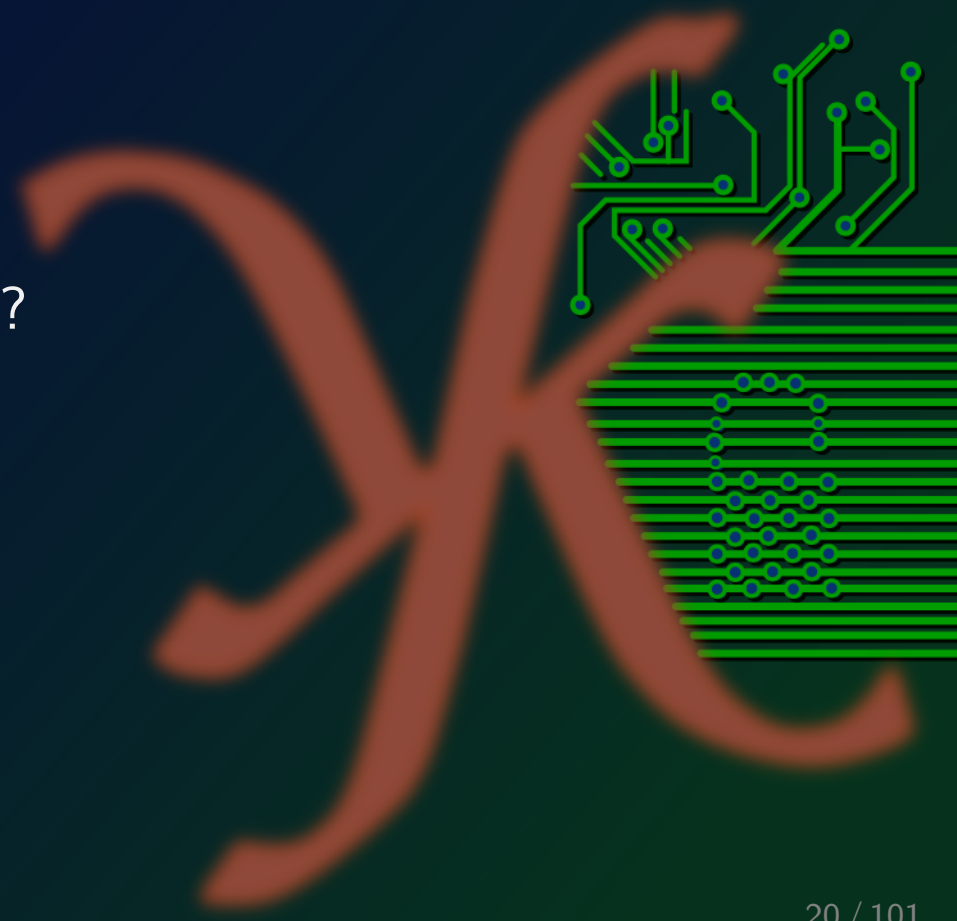
Perguntas rápidas

- Segurança?
- Criptografia?
- Simétrica e assimétrica?
- AES?
- RSA?
- Curvas Elipticas?



Perguntas rápidas

- Segurança?
- Criptografia?
- Simétrica e assimétrica?
- AES?
- RSA?
- Curvas Elipticas?
- Curvas Elipticas Criptograficas?

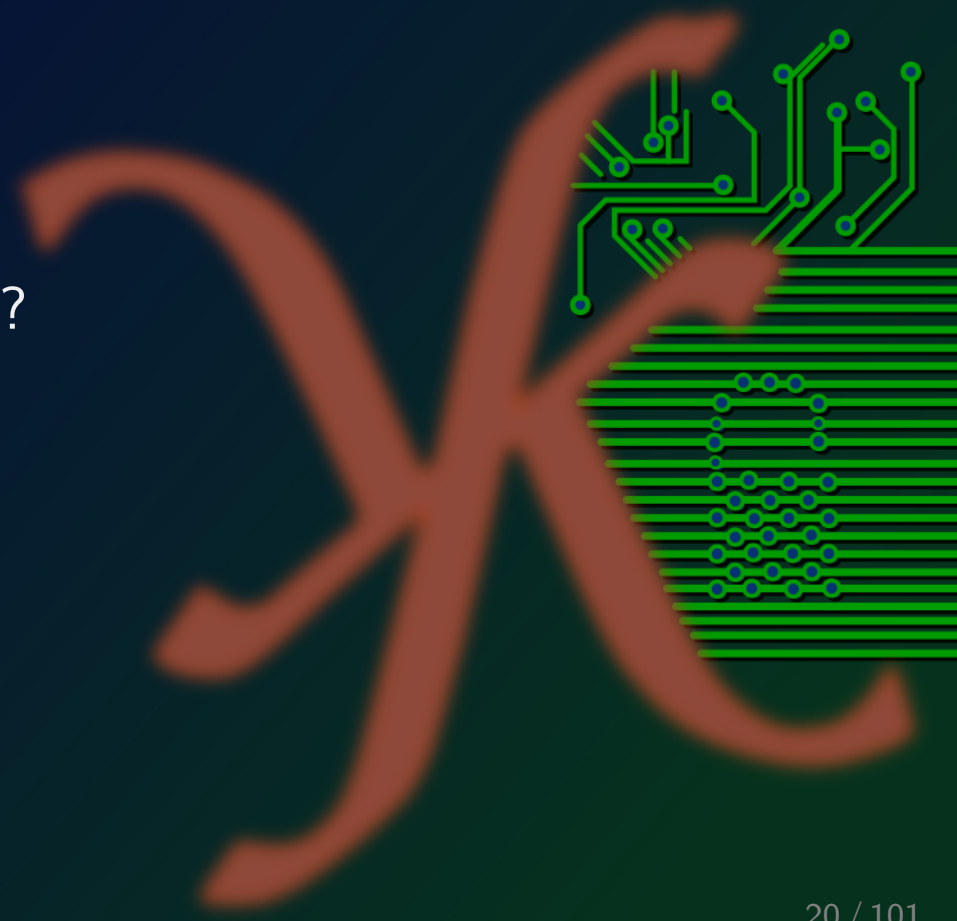


Perguntas rápidas

- Segurança?
- Criptografia?
- Simétrica e assimétrica?
- AES?
- RSA?
- Curvas Elipticas?
- Curvas Elipticas Criptograficas?
- Hardware?

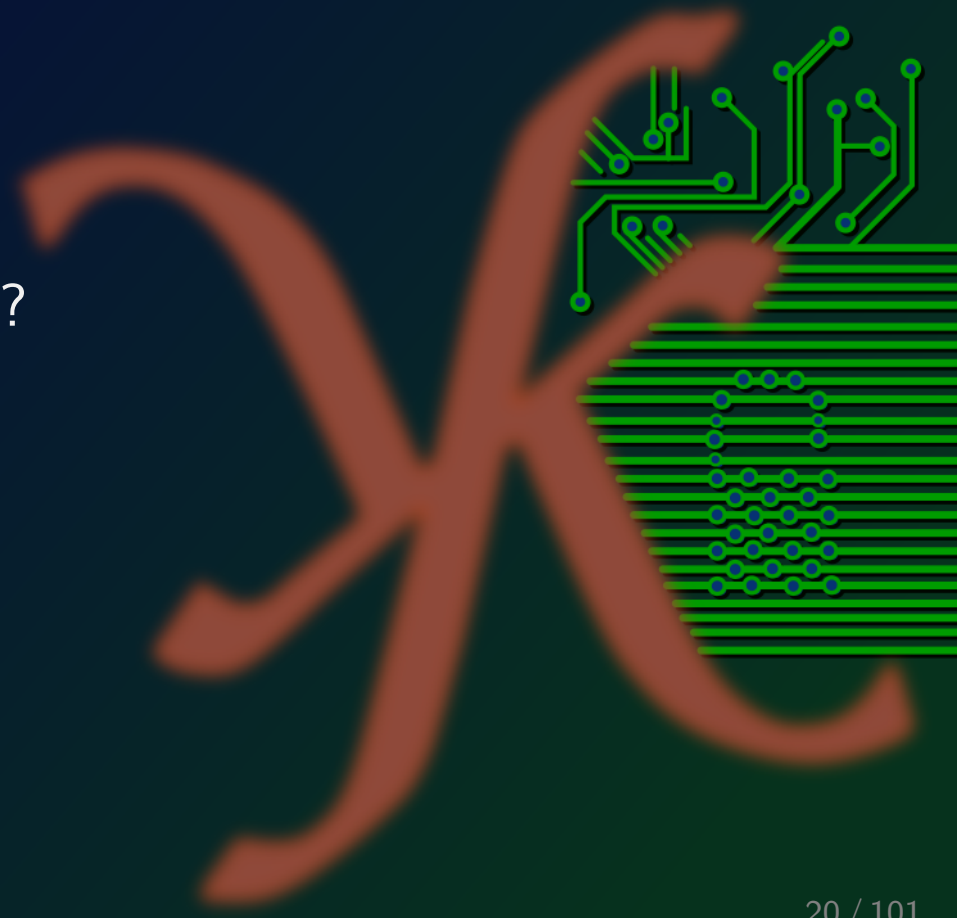
Perguntas rápidas

- Segurança?
- Criptografia?
- Simétrica e assimétrica?
- AES?
- RSA?
- Curvas Elipticas?
- Curvas Elipticas Criptograficas?
- Hardware?
- Processadores?



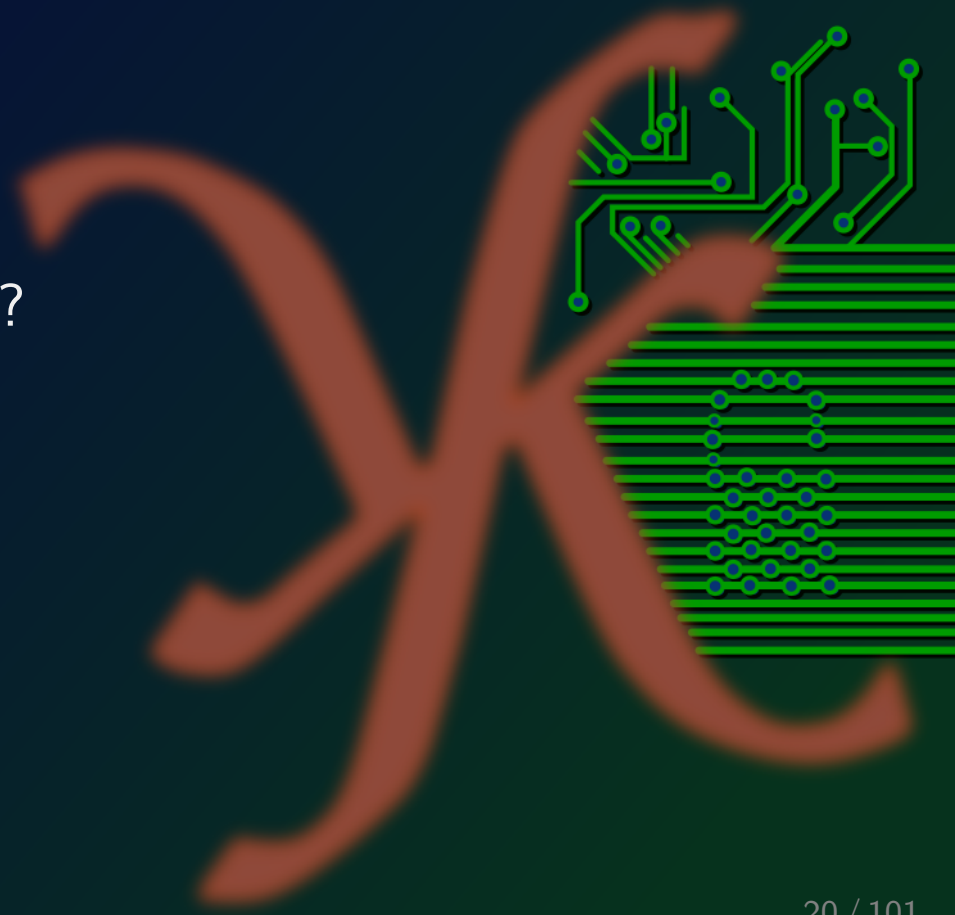
Perguntas rápidas

- Segurança?
- Criptografia?
- Simétrica e assimétrica?
- AES?
- RSA?
- Curvas Elipticas?
- Curvas Elipticas Criptograficas?
- Hardware?
- Processadores?
- Co-Processadores?



Perguntas rápidas

- Segurança?
- Criptografia?
- Simétrica e assimétrica?
- AES?
- RSA?
- Curvas Elipticas?
- Curvas Elipticas Criptograficas?
- Hardware?
- Processadores?
- Co-Processadores?
- FPGA?



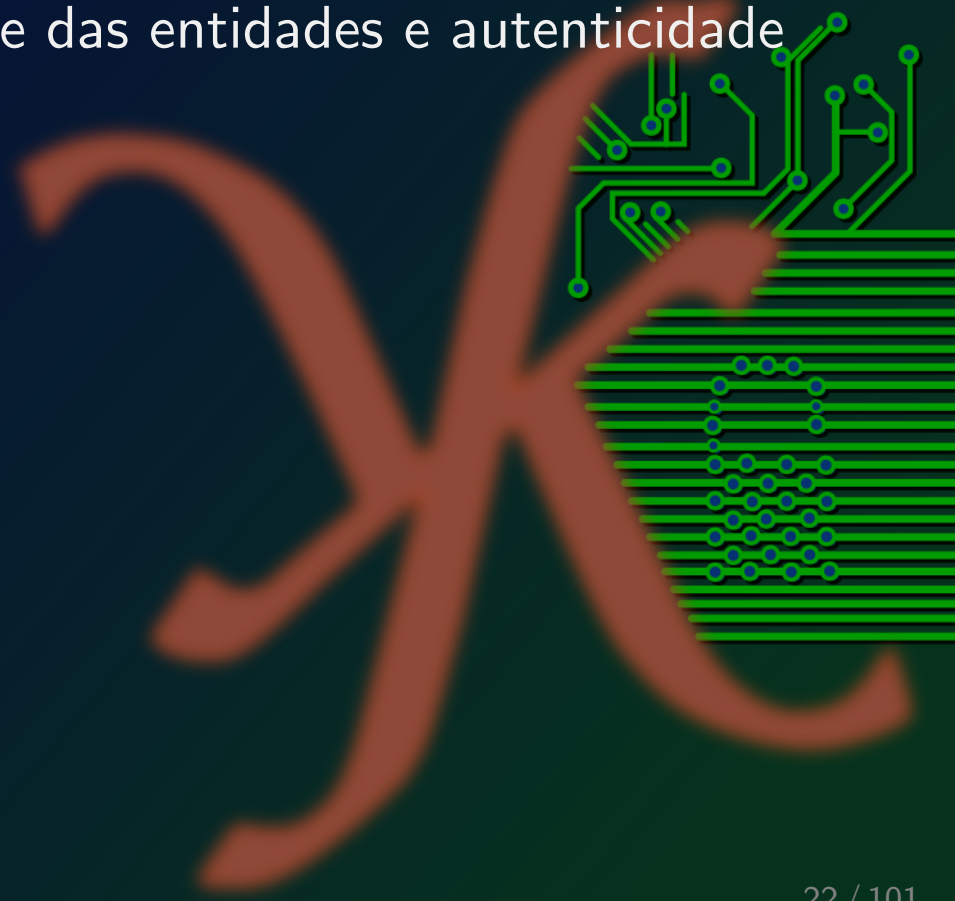
Perguntas rápidas

- Segurança?
- Criptografia?
- Simétrica e assimétrica?
- AES?
- RSA?
- Curvas Elipticas?
- Curvas Elipticas Criptograficas?
- Hardware?
- Processadores?
- Co-Processadores?
- FPGA?
- E agora?

Sumário

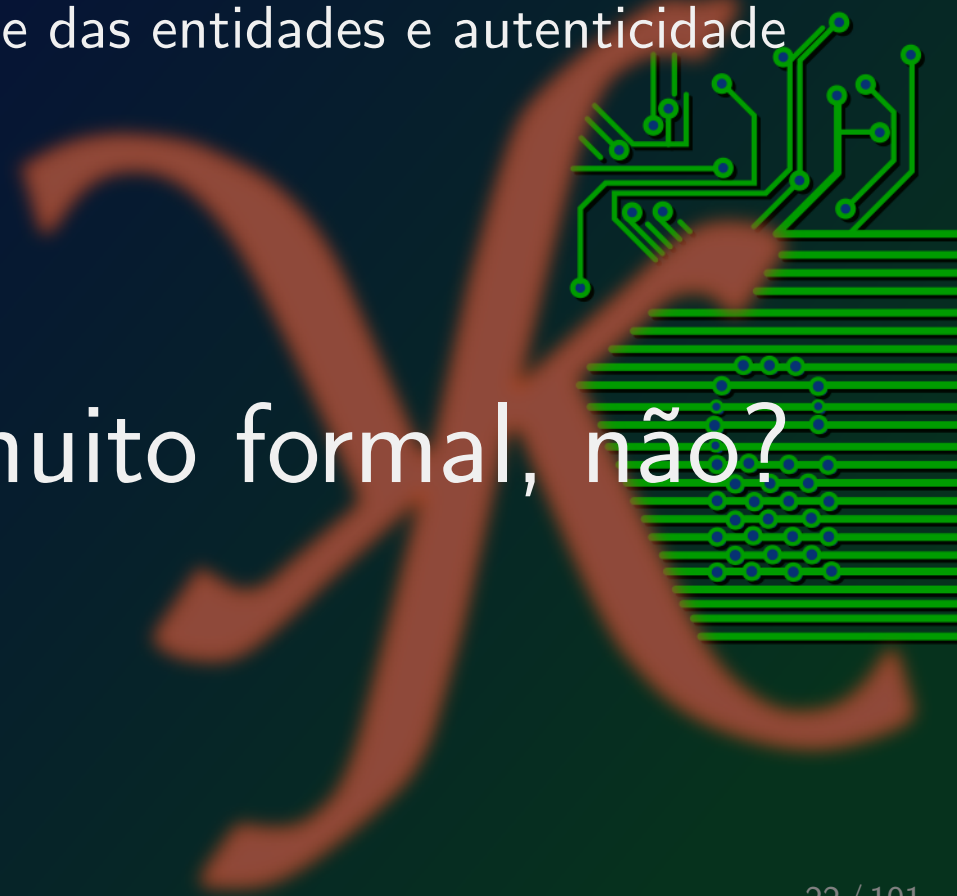
O que é Criptografia?

Criptografia é o estudo de técnicas matemáticas que visam a segurança da informação, por meio das quais é possível obter-se a integridade dos dados, autenticidade das entidades e autenticidade da informação.



O que é Criptografia?

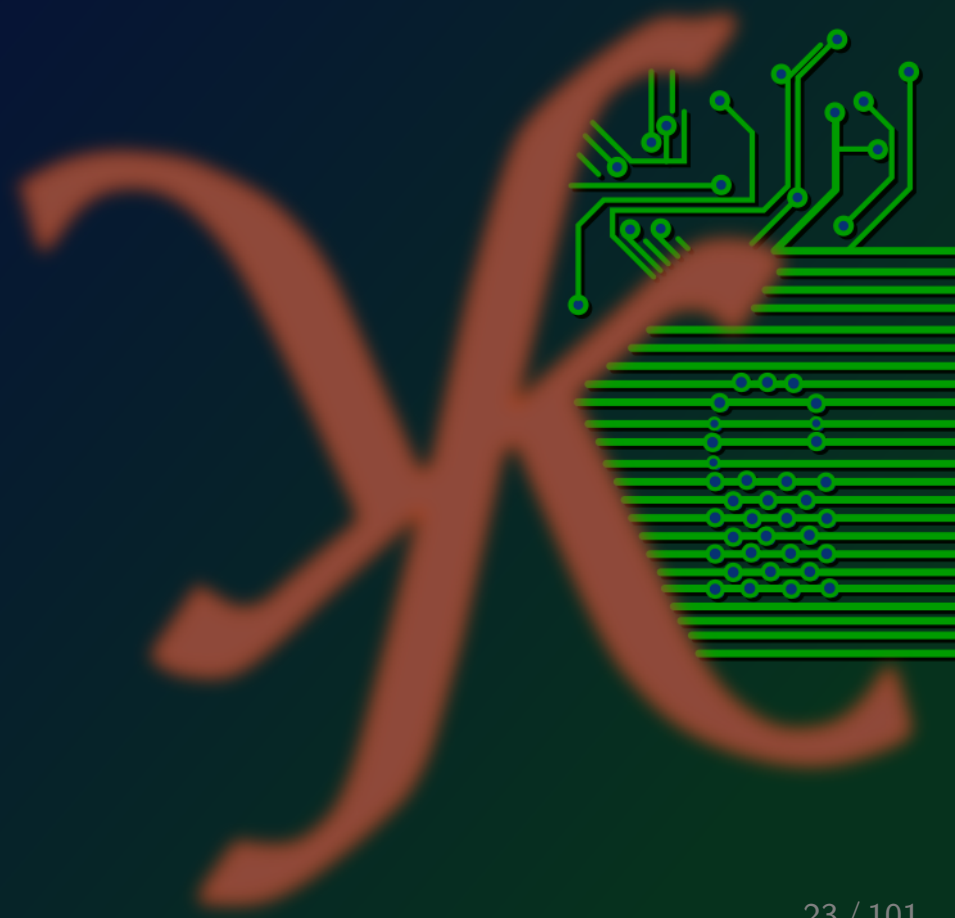
Criptografia é o estudo de técnicas matemáticas que visam a segurança da informação, por meio das quais é possível obter-se a integridade dos dados, autenticidade das entidades e autenticidade da informação.



OK.... Mas isso é muito formal, não?

Por quê usar criptografia?

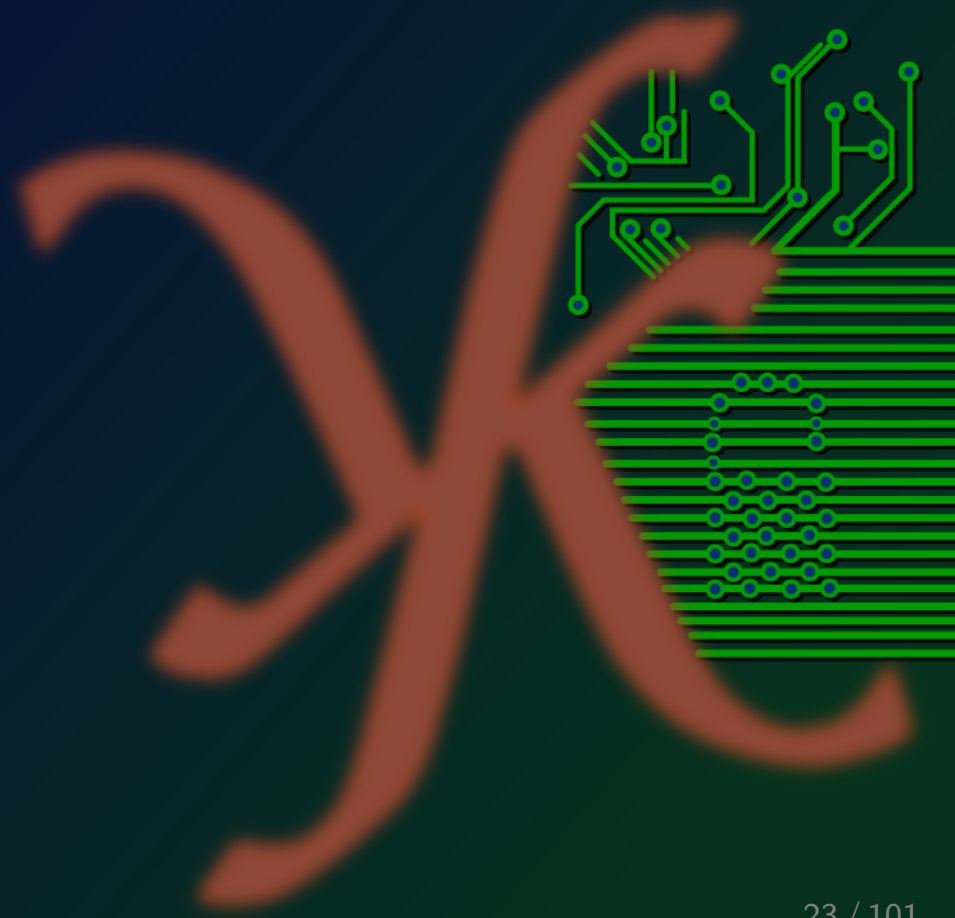
Seria por “status”?



Por quê usar criptografia?

Seria por “status”?

... por modinha?

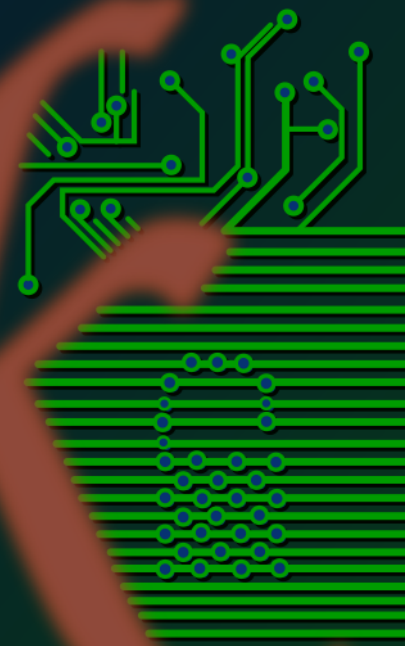


Por quê usar criptografia?

Seria por “status”?

... por modinha?

Para o Trump não ler meus e-mails?



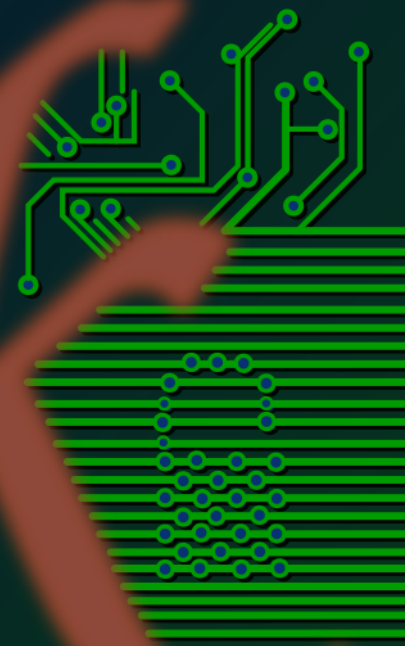
Por quê usar criptografia?

Seria por “status”?

... por modinha?

Para o Trump não ler meus e-mails?

Para minhas fotos “secretas” não vazarem?



Por quê usar criptografia?

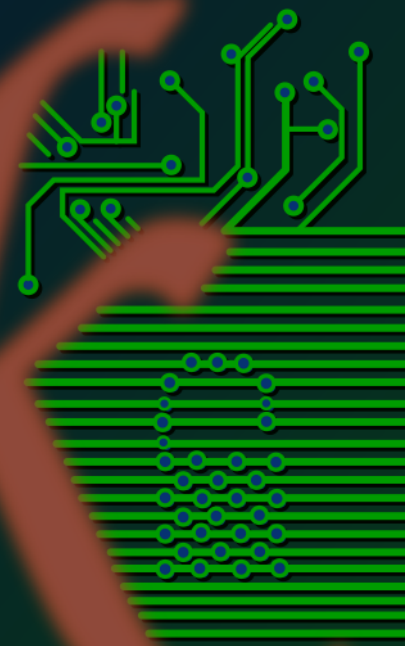
Seria por “status”?

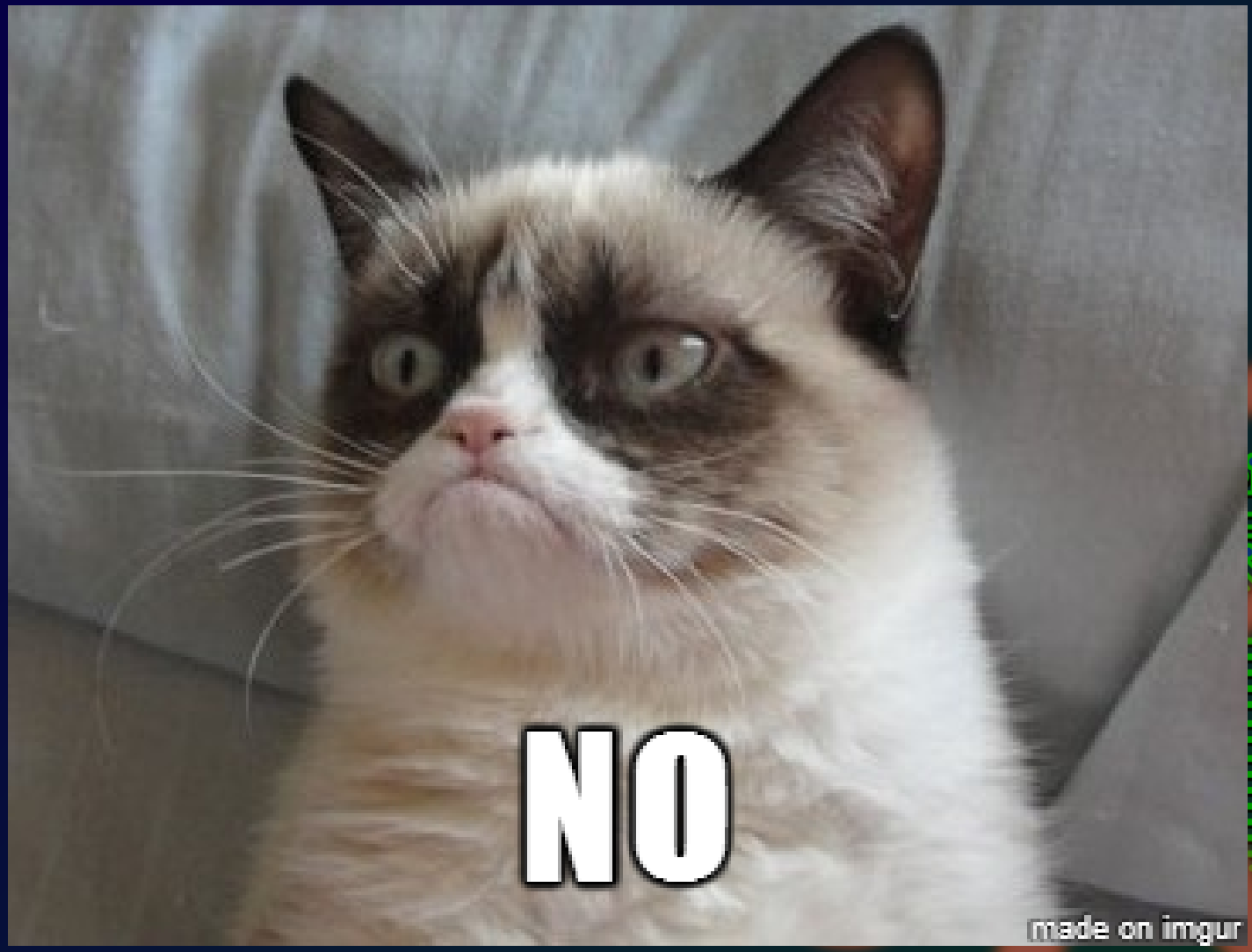
... por modinha?

Para o Trump não ler meus e-mails?

Para minhas fotos “secretas” não vazarem?

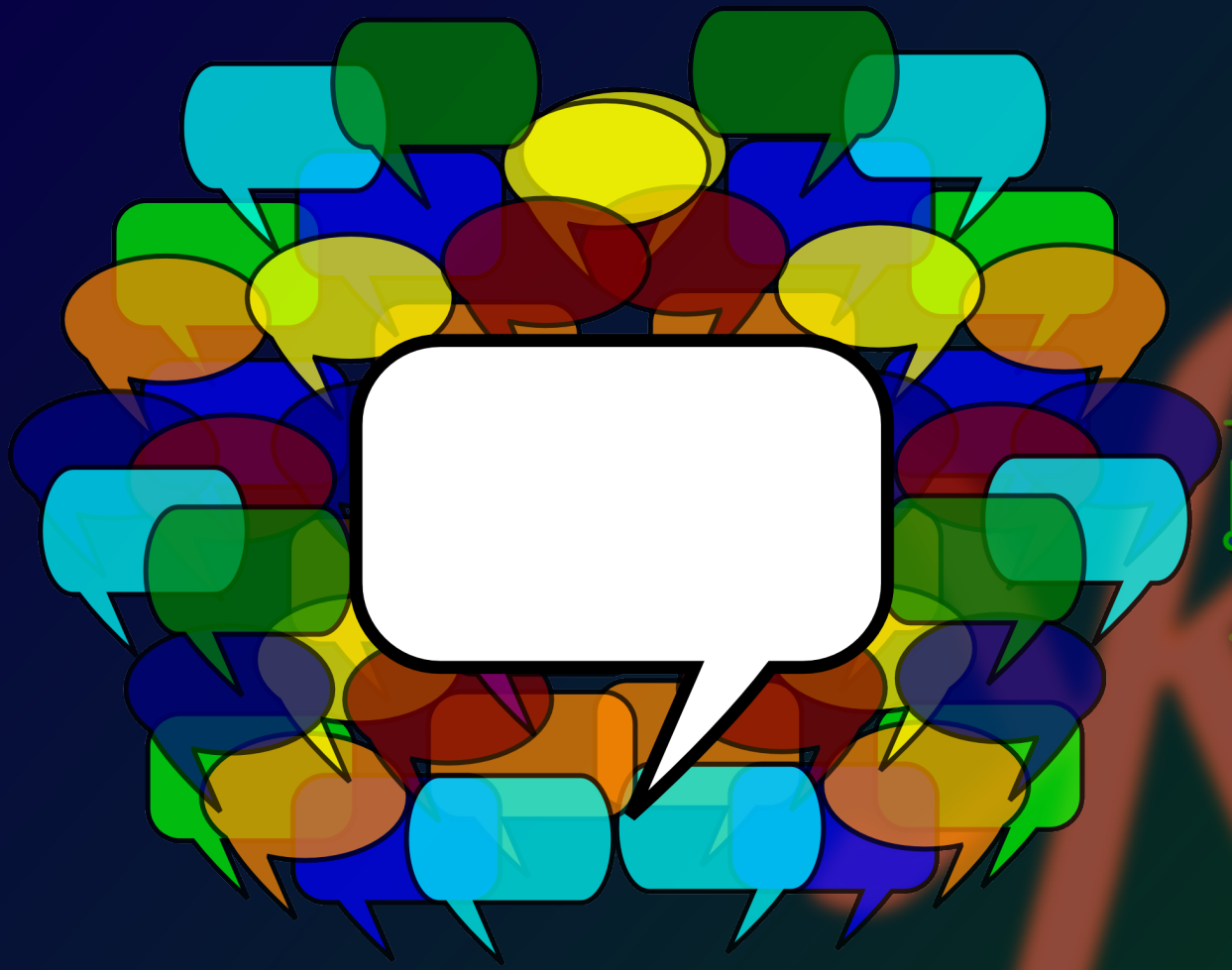
Será mesmo que é para isso?




NO
made on imgur

Why not?

Necessidade de comunicação!



Todos PRECISAM se comunicar

[1]

[1] Cinderela, Disney, 22 de Maio de 1950

Há quem quer ler a mensagem...

[2]

...Mesmo essa não sendo destinada a ele...

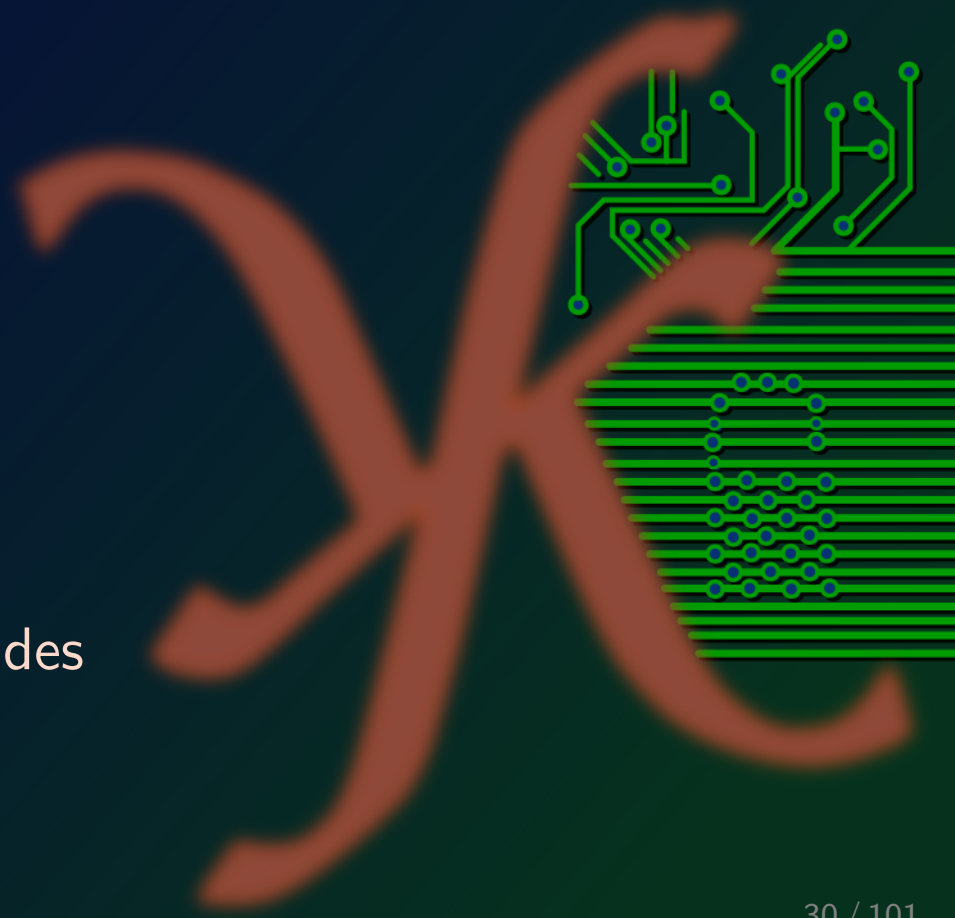
[2] Malévola em "A bela adormecida", Disney, 06 de Fevereiro de 1959

E com a perda da informação...

...As coisas geralmente não acabam bem.

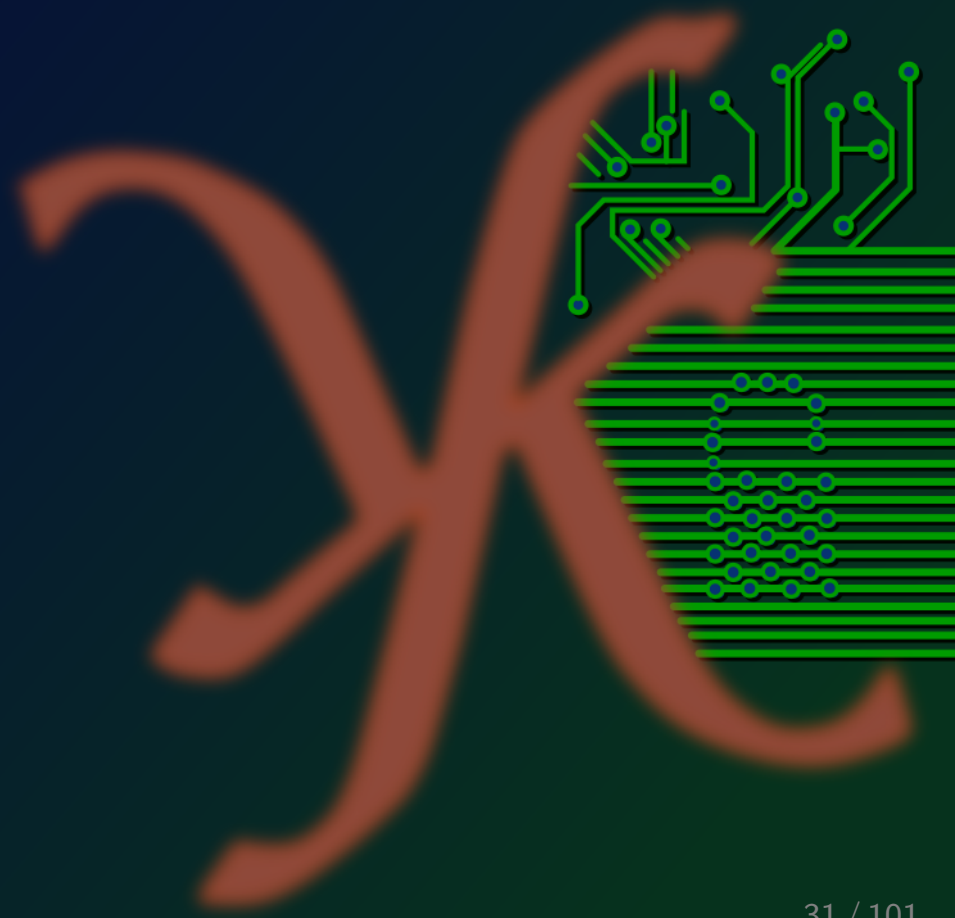
[3] "A bela adormecida", Viktor Mikhailovich Vasnetsov

Sumário



Tipos de criptografia

Existem 2 grupos em que algoritmos criptográficos são divididos

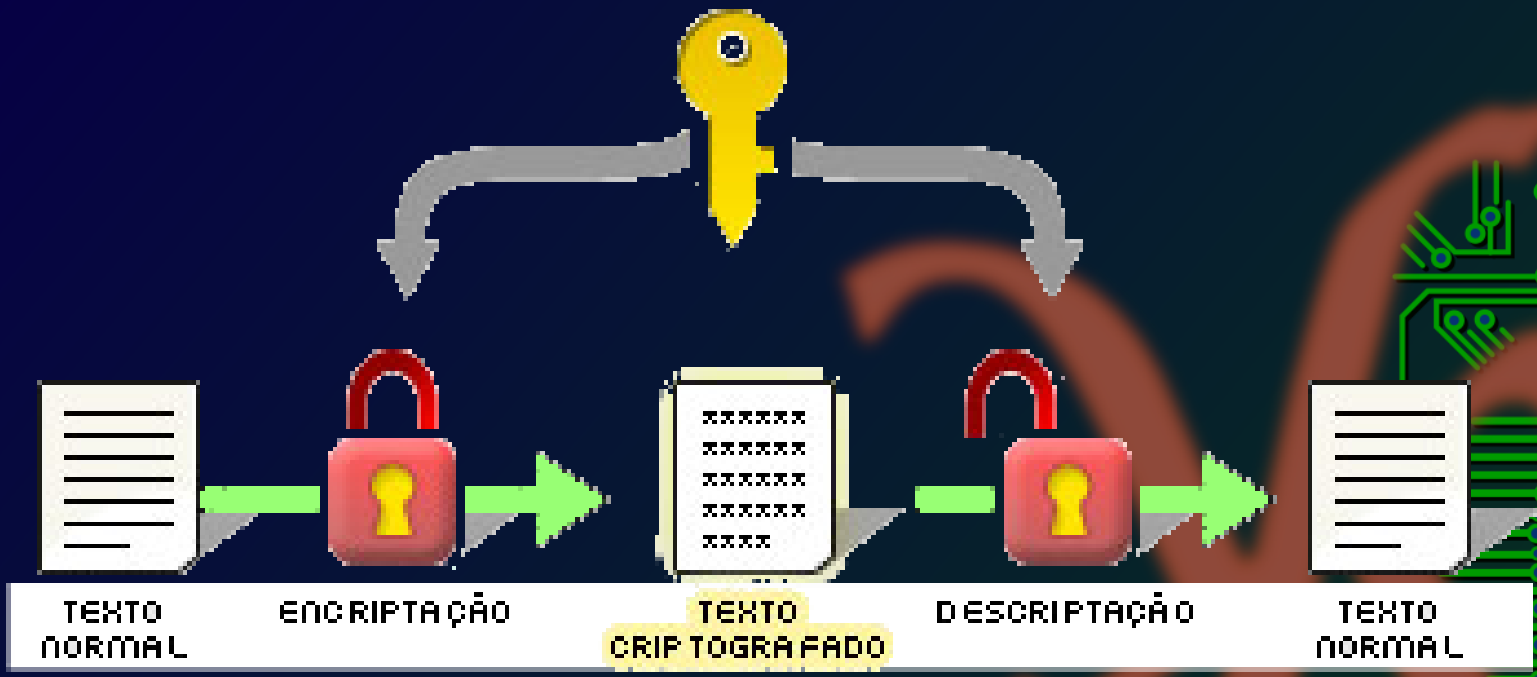


Tipos de criptografia

Existem 2 grupos em que algoritmos criptográficos são divididos

- Criptografia **Simétrica**
- Criptografia **Assimétrica**

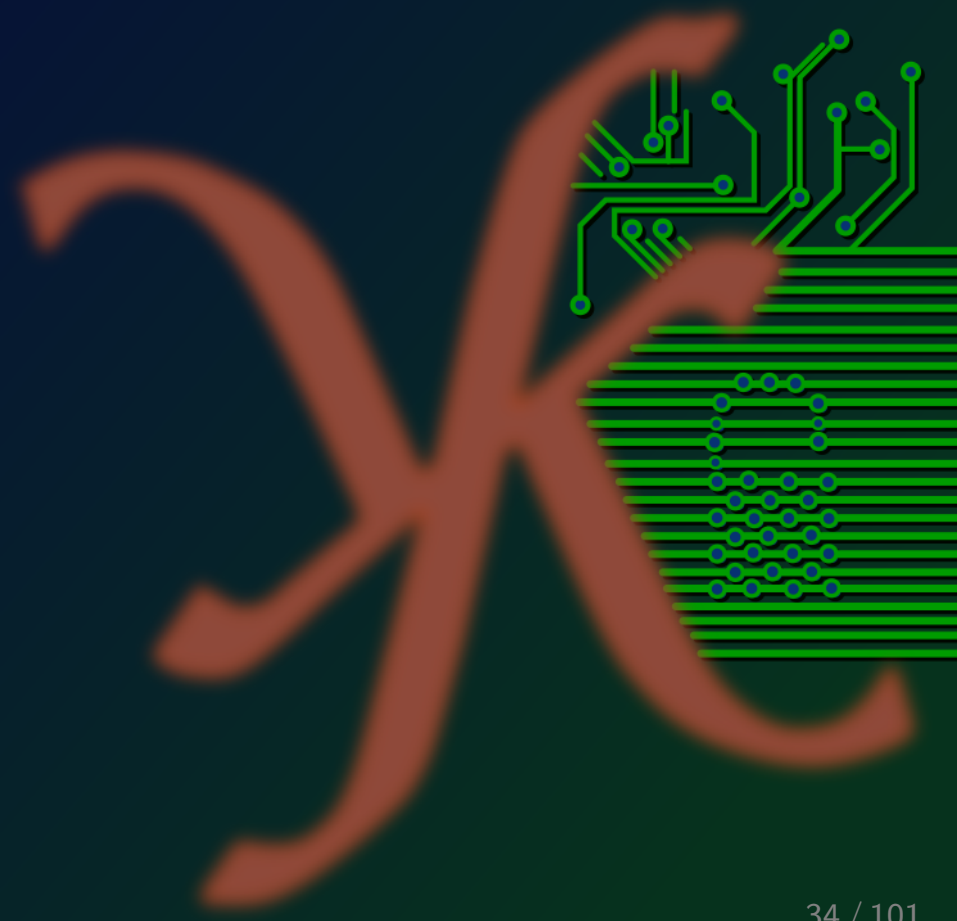
Cifras simétricas



A chave que fecha abre

Exemplos

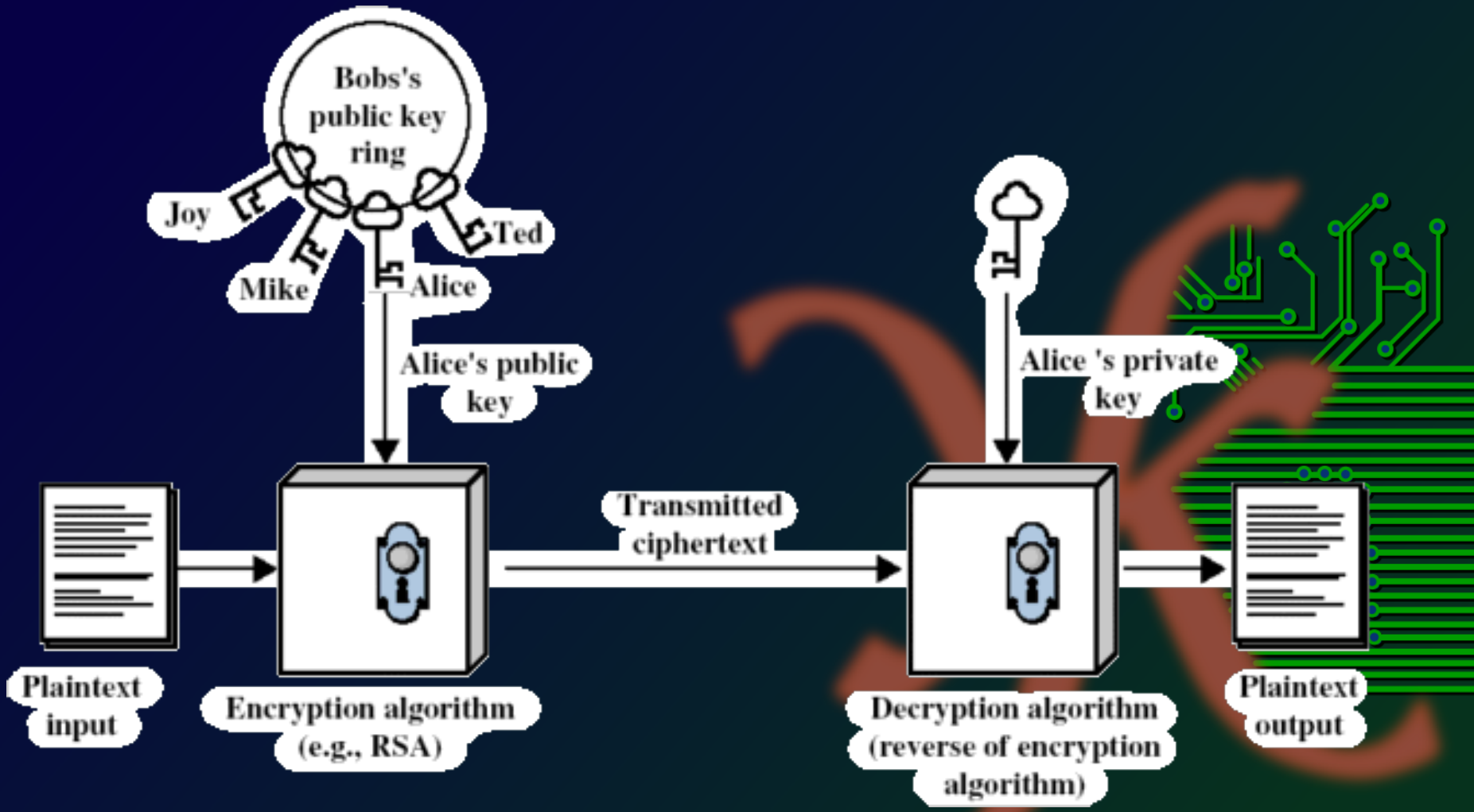
- Enigma
- DES
- AES



Onde normalmente são usados?

- Proteger arquivos em disco (TrueCrypt, EFS, etc)
- Proteção de mídias tais como DVD (css, aacs, etc)
- Transmissão eficiente de dados (desde que se tenha uma chave secreta compartilhada pelos envolvidos)
- ETC...

Criptografia Assimétrica



Quem fecha não abre

Exemplos

- Diffie–Hellman
- RSA
- Curvas Elípticas Criptográficas \o/
- Pós-quantica \o/\o/

Diffie-Hellman

Secreto	Público	Calcula	Envia
a	p, g		p, g $\rightarrow$
a	p, g, A	$g^a mod(p) = A$	A $\rightarrow$
a	p, g, A		$\leftarrow$ B
a, **s**	p, g, A, B	$B^a mod(p) = s$	

Envia	Calcula	Público	Secreto
p, g $\rightarrow$			b
A $\rightarrow$		p, g	b
$\leftarrow$ B	$g^b mod(p) = B$	p, g, A, B	b
	$A^b mod(p) = s$	p, g, A, B	b, **s**

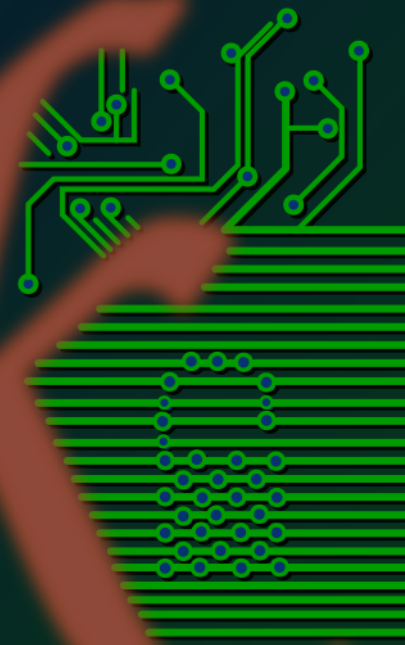
Ponto inicial...

Alice:

- Seleciona-se a=6, p=23 e g=5
- Envia p e g
- Calcula-se $A = g^a mod(p)$
  $A = 5^6 mod(23) = 15.625 mod(23) = 8$
- Envia A

Bob:

- Seleciona-se b=15
- Recebe p, g e A
- Calcula-se $B = g^b mod(p)$
  $B = 5^{15} mod(23) = 30.517.578.125 mod(23) = 19$
- Envia B



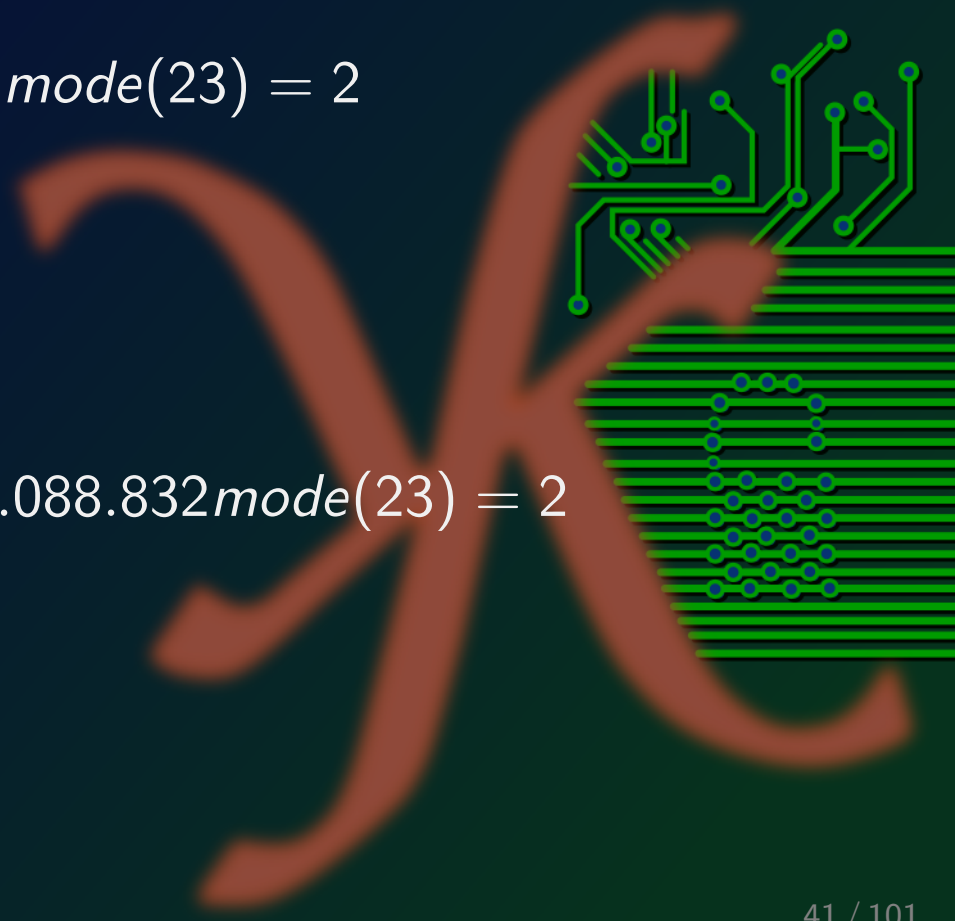
Segredo compartilhado

Alice:

- Calcula $s = B^a mod(p)$
  $s = 19^6 mod(23) = 47.045.881 mode(23) = 2$

Bob:

- Calcula $s = A^b mod(p)$
  $s = 8^{15} mod(23) = 35.184.372.088.832 mode(23) = 2$



Quem usa?

- SSH
- Bitcoin
- HTTPS
- TODOS (deveriam)

Sumário

Aritmética sobre corpos Finitos

Alguns corpos são melhores para uma implementação eficiente

- Corpos Primos
- Corpos Binários

Vamos complicar um pouco?

Mas o que é um Corpo?

A definição de corpo segundo Menezes é:

Consist of a set F together with two operation, addition $(+)$ and multiplication $(.)$

Mas o que é um Corpo?

A definição de corpo segundo Menezes é:

Consist of a set F together with two operation, addition $(+)$ and multiplication $(.)$

Esse corpos possui operações básicas:

- $(F, +)$ é um grupo **abeliano** sobre adição, cuja identidade é 0.
- $(F - \{0\}, .)$ é um grupo **abeliano** sobre a multiplicação, cuja identidade é 1.

Mas o que é um Corpo?

A definição de corpo segundo Menezes é:

Consist of a set F together with two operation, addition $(+)$ and multiplication $(.)$

Esse corpos possui operações básicas:

- $(F, +)$ é um grupo **abeliano** sobre adição, cuja identidade é 0.
- $(F - \{0\}, .)$ é um grupo **abeliano** sobre a multiplicação, cuja identidade é 1.
- A lei distributiva:
  $(a + b).c = a.c + a.b$ é valida para $a, b, c \in F$

Corpo Primo

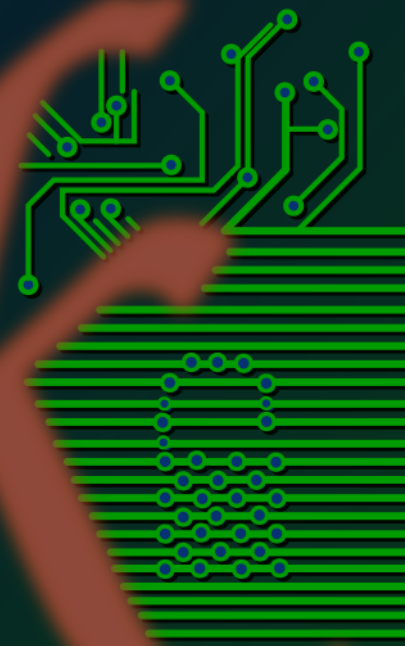
- Possui um numero primo como modulo.
- Corpo F_{23}:
  - $1223 \bmod 23 = 4$
  - 911 mor $23 = 14$
  - $17 \bmod 23 = 17$
- Secp256k1 $(2^{256} - 0x1000003D1)$:
  FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
  FFFFFFFF FFFFFFFF FFFFFFFE FFFFFC2F

Corpo Binário

Corpos finitos da ordem 2^m são denominados **corpos binários** (ou *Corpos Finitos de característica-2*).

Um corpo finito binário é importante saber quem é o **polinômio irredutível** $f(z)$.

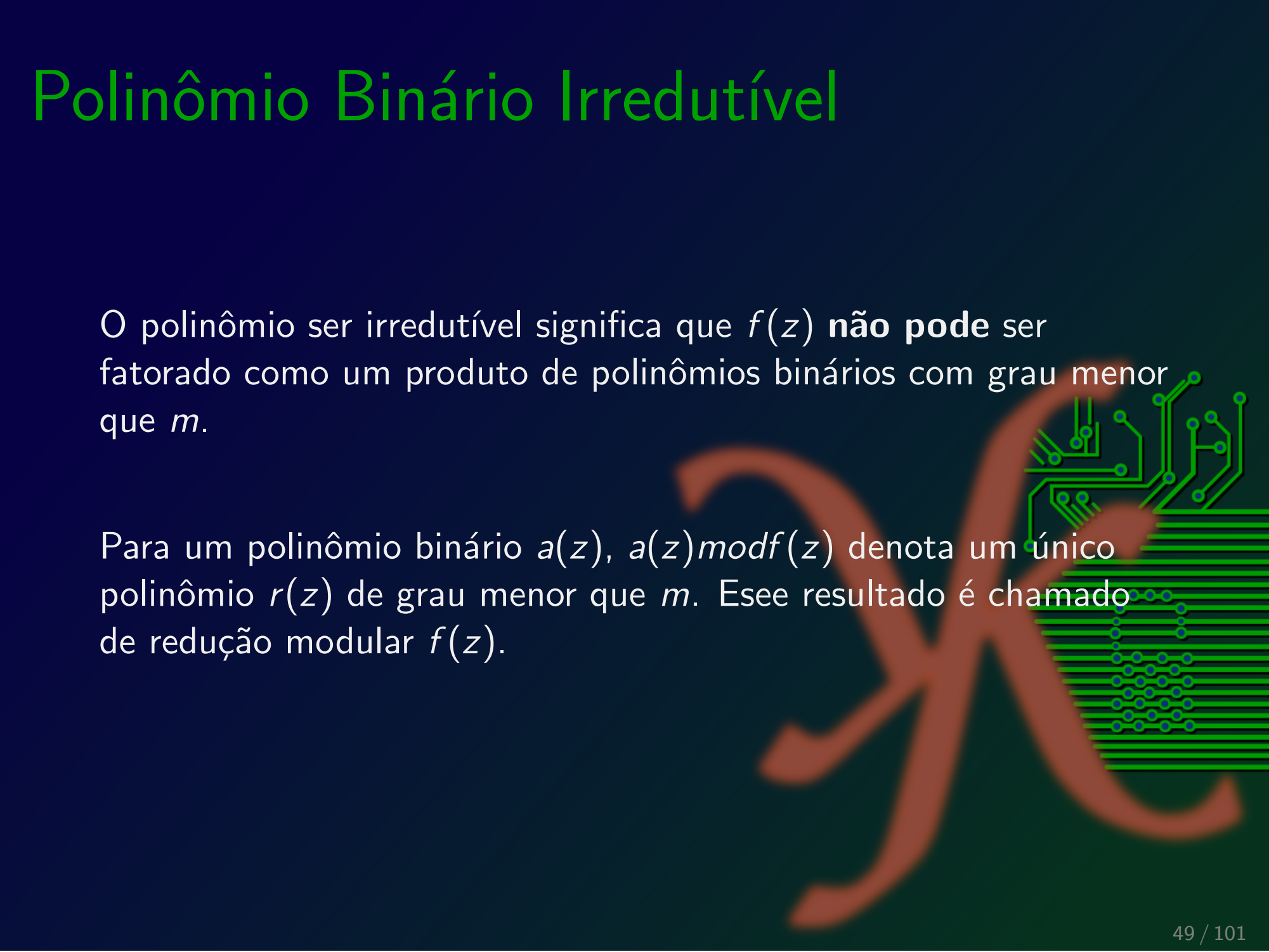
AES aplica-se sobre um corpo de 8 bits
Curvas Elipticas sobre corpos > 300 bits



Polinômio Binário Irredutível

O polinômio ser irredutível significa que $f(z)$ **não pode** ser fatorado como um produto de polinômios binários com grau menor que m.

Para um polinômio binário $a(z)$, $a(z) mod f(z)$ denota um único polinômio $r(z)$ de grau menor que m. Esee resultado é chamado de redução modular $f(z)$.



Exemplo no corpo F_{2^4}

Em F_{2^4} o polinômio irredutível é $f(z) = z^4 + z^1 + z^0$.

Exemplos de operações aritméticas:

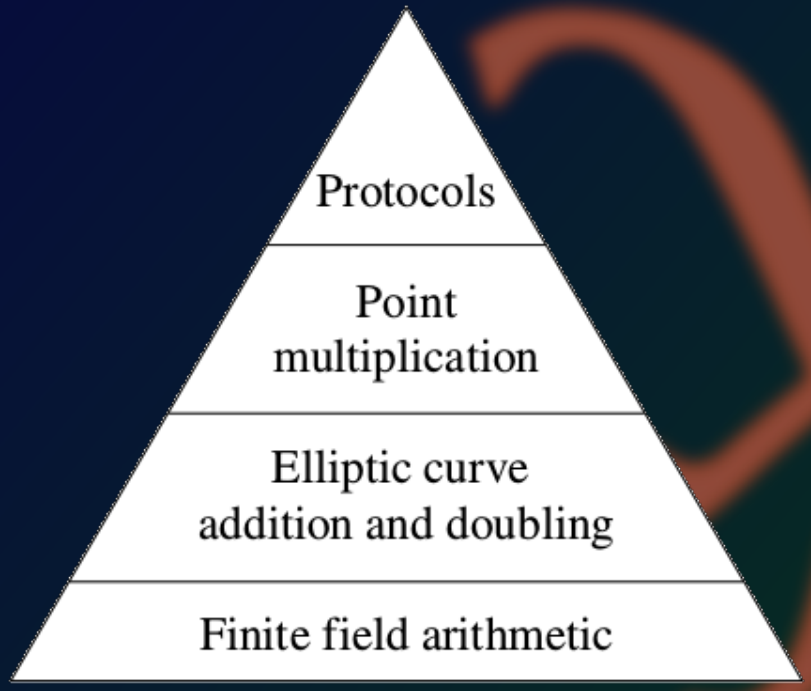
- Adição: $(z^3 + z^2 + 1) + (z^2 + z + 1) = z^3 + z$
- Subtração: $(z^3 + z^2 + 1) - (z^2 + z + 1) = z^3 + z$
- Multiplicação: $(z^3 + z^2 + 1).(z^2 + z + 1) = z^5 + z + 1$
  $= z.z^4 + z + 1 = z(z + 1) + z + 1 = z^2 + z + z + 1 = z^2 + 1$
- Inversão: $(z^3 + z^2 + 1)^{-1} = z^2$
  Sabendo que $(z^3 + z^2 + 1).z^2 mod(z^4 + z + 1) = 1$

Sumário

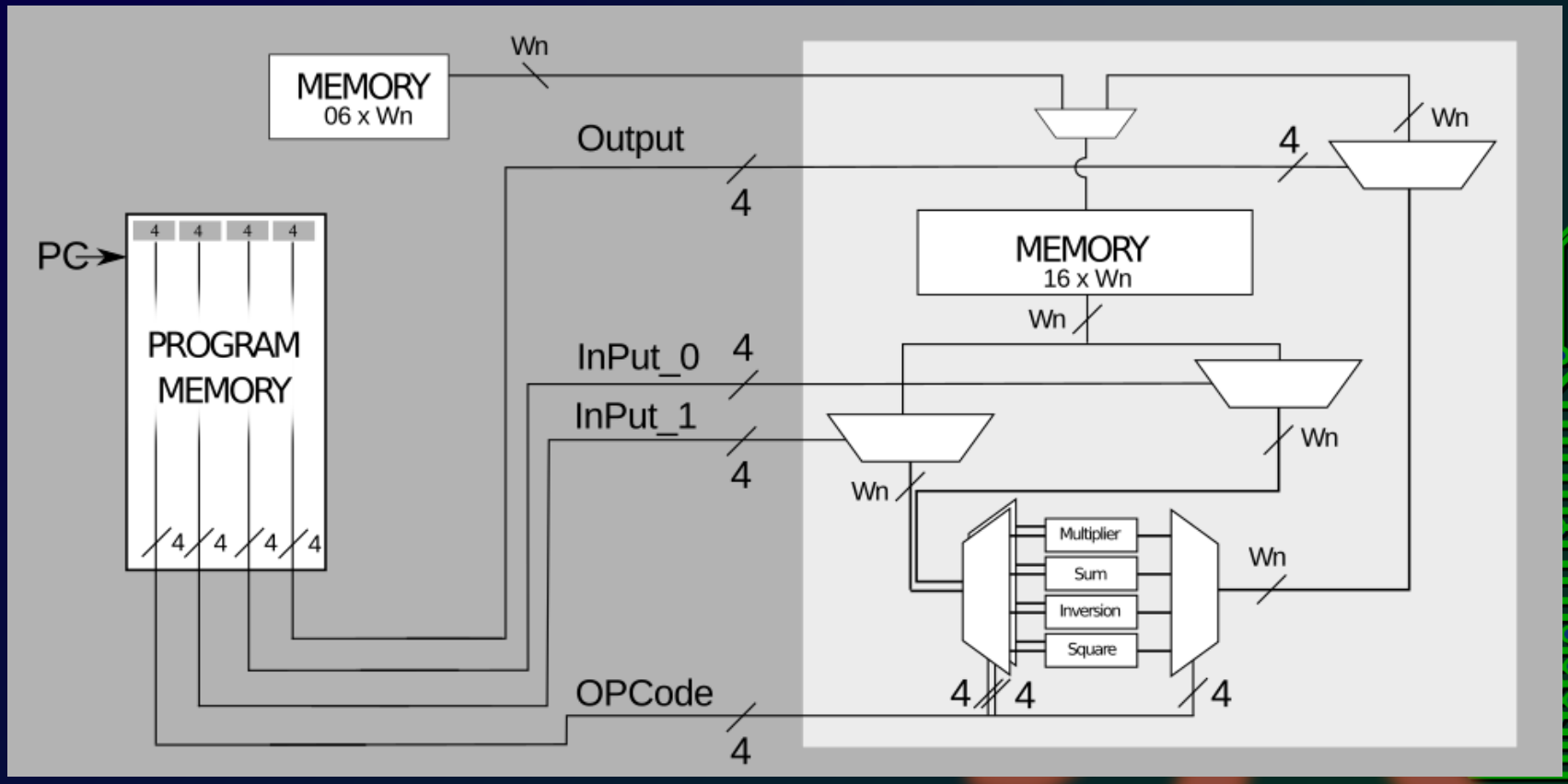
Design Modular

Por exemplo, a implementação de uma assinatura feita totalmente em hardware deve considerar que a **única entrada** no dispositivo é **a mensagem** a ser assinada e a **única saída** é a **assinatura** desta mensagem.

O processo de implementação pode ser representado como:



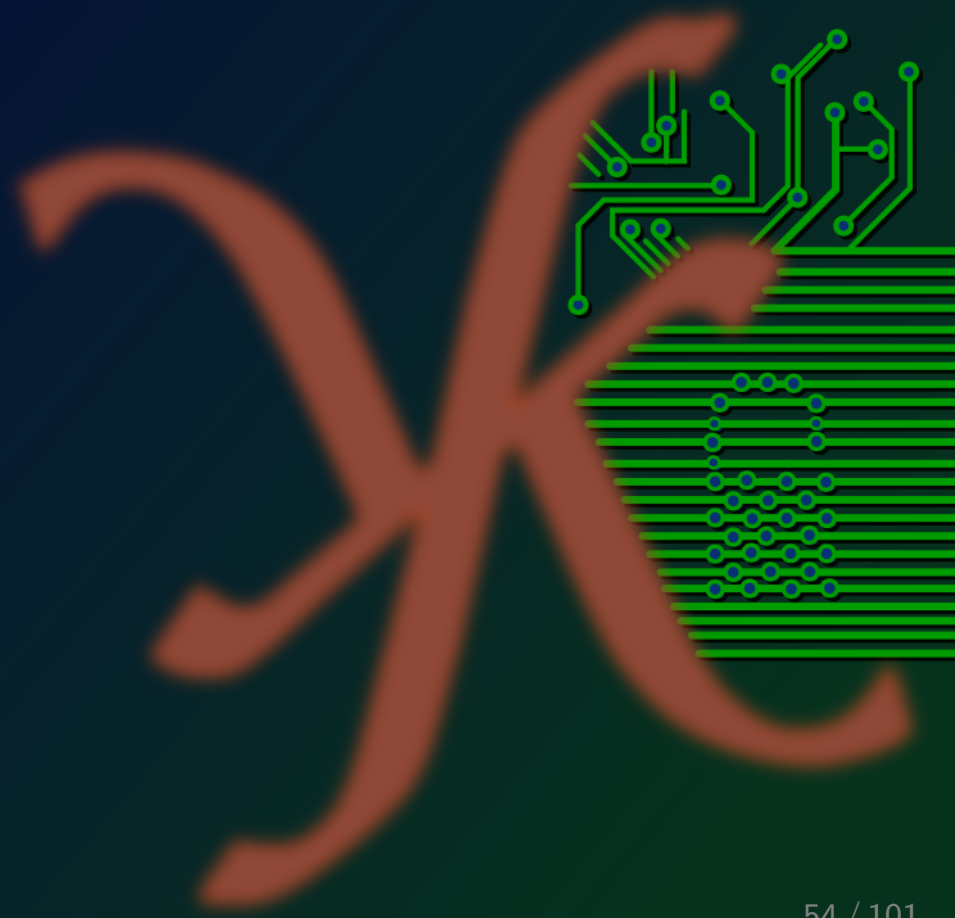
Arquitetura para um Co-Processor



Por que Hardware?

Critérios pelo qual se opta pelo Hardware:

- **Custo** é sempre um fator significante



Por que Hardware?

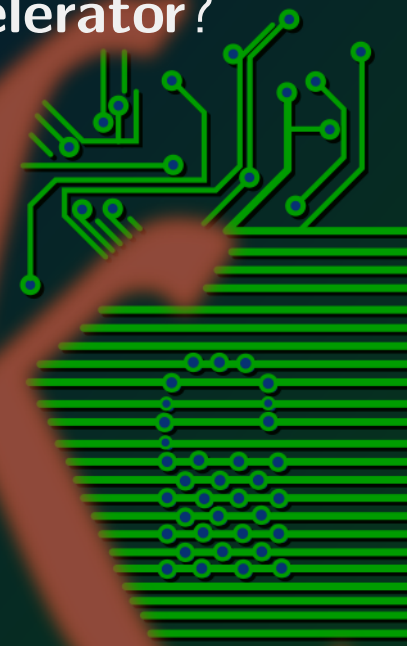
Critérios pelo qual se opta pelo Hardware:

- **Custo** é sempre um fator significante
- Existe um bom argumento para usar **hardware accelerator**?

Por que Hardware?

Critérios pelo qual se opta pelo Hardware:

- **Custo** é sempre um fator significante
- Existe um bom argumento para usar **hardware accelerator**?
- **Throughput**: Qual a velocidade desejada?



Por que Hardware?

Critérios pelo qual se opta pelo Hardware:

- **Custo** é sempre um fator significante
- Existe um bom argumento para usar **hardware accelerator**?
- **Throughput**: Qual a velocidade desejada?
- **Flexibilidade**: Precisa ter a habilidade de comportar mais funcionalidades?

Por que Hardware?

Critérios pelo qual se opta pelo Hardware:

- **Custo** é sempre um fator significante
- Existe um bom argumento para usar **hardware accelerator**?
- **Throughput**: Qual a velocidade desejada?
- **Flexibilidade**: Precisa ter a habilidade de comportar mais funcionalidades?
- **Consumo de Energia**: É ou não uma preocupação?

Por que Hardware?

Critérios pelo qual se opta pelo Hardware:

- **Custo** é sempre um fator significante
- Existe um bom argumento para usar **hardware accelerator**?
- **Throughput**: Qual a velocidade desejada?
- **Flexibilidade**: Precisa ter a habilidade de comportar mais funcionalidades?
- **Consumo de Energia**: É ou não uma preocupação?
- **Segurança**: Countermeasures attacks based on timing, power analysis, and electromagnetic radiation precisam ser considerados.

Por que Hardware?

Critérios pelo qual se opta pelo Hardware:

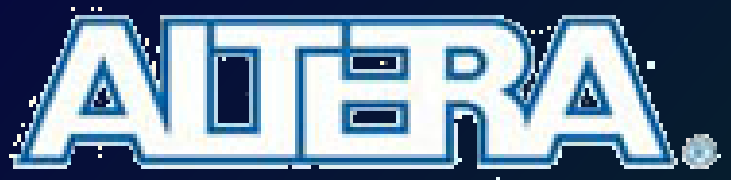
- **Custo** é sempre um fator significante
- Existe um bom argumento para usar **hardware accelerator**?
- **Throughput**: Qual a velocidade desejada?
- **Flexibilidade**: Precisa ter a habilidade de comportar mais funcionalidades?
- **Consumo de Energia**: É ou não uma preocupação?
- **Segurança**: Countermeasures attacks based on timing, power analysis, and electromagnetic radiation precisam ser considerados.
- **Plataforma de desenvolvimento:** Qual FPGA?

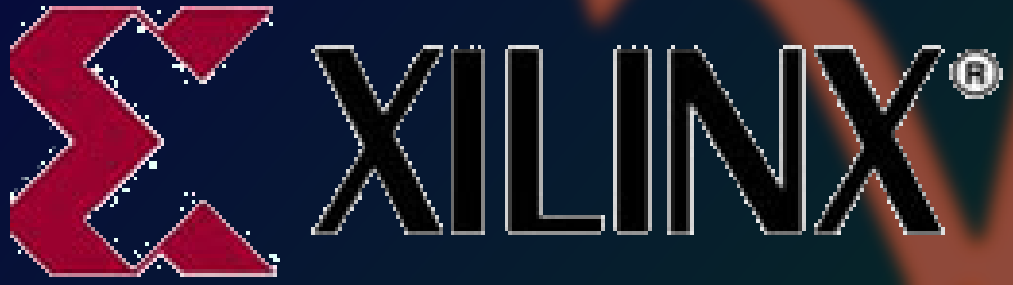
Escolha do hardware

Após estas considerações, considere os itens determinantes para a escolha entre dispositivos de baixa ou alta potência:

High-end device		Low-end device	
High priority	Low priority	High Priority	Low priority
Throughput	Cost	Cost	Throughput
Security	Power consumption	Hardware vs. software	Flexibility
Scalability	Complexity	Complexity	Algorithm agility
System architecture		Power consumption	Scalability
Implementation platform		Security	
Algorithm agility		System architecture	
Flexibility		Implementation platform	
Hardware vs. software			

Fabricantes



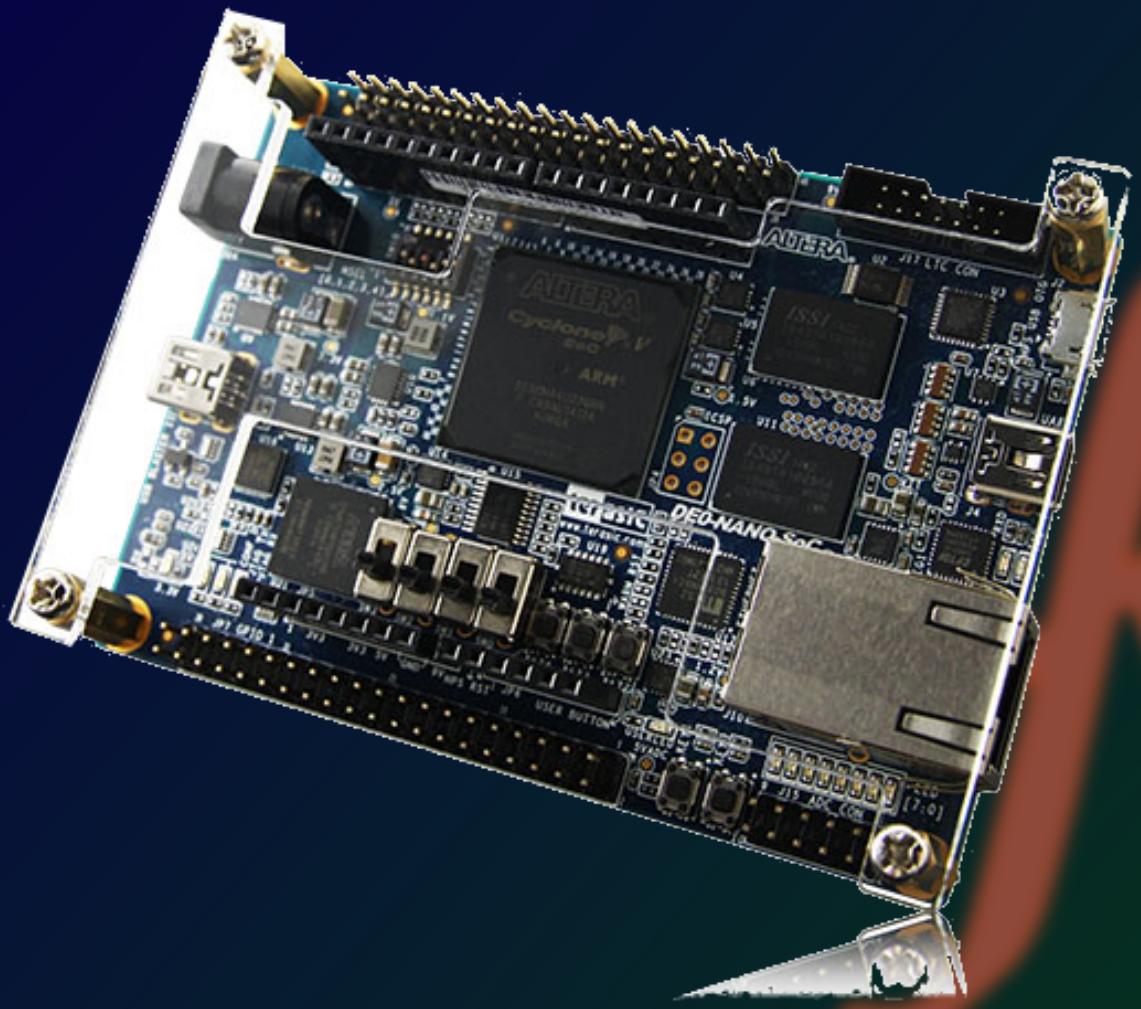


Intel FPGA

Comprou a Altera

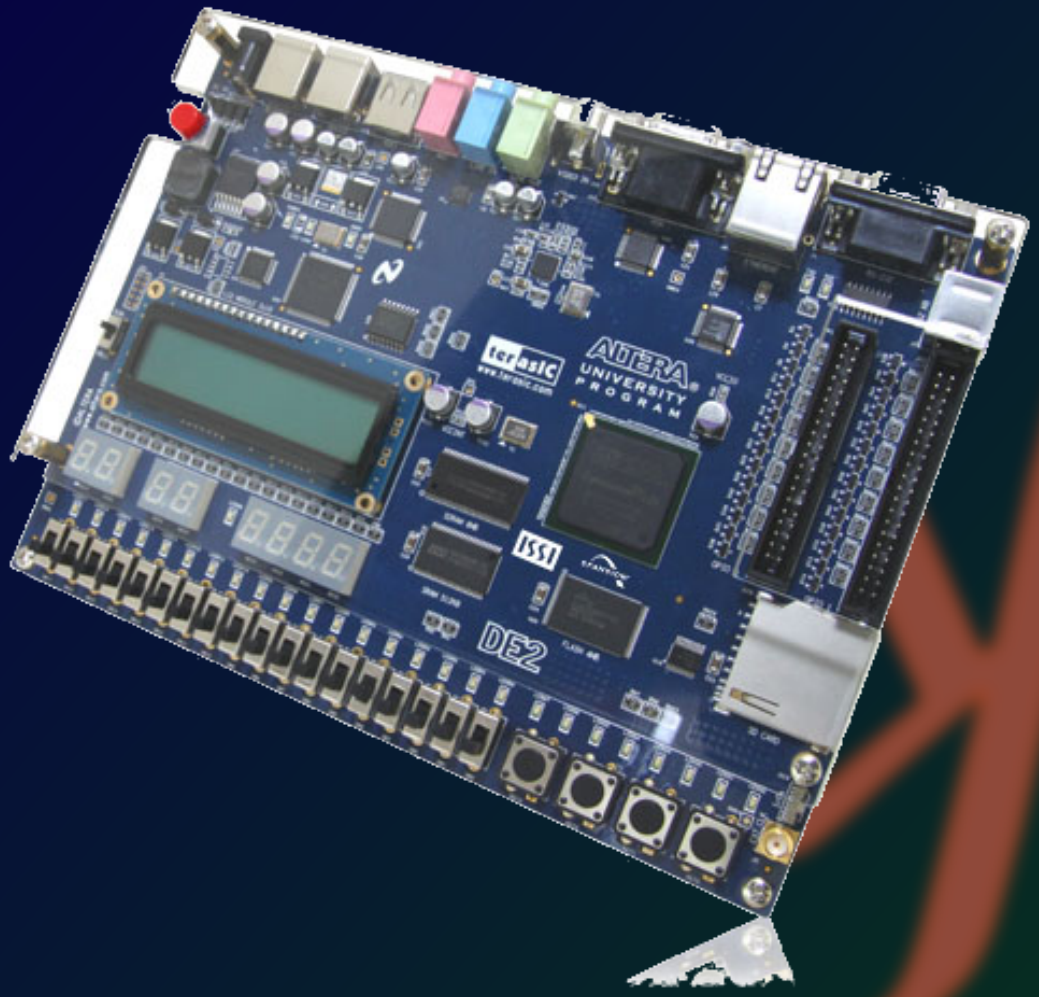
Placas legais

DE0-Nano-SoC



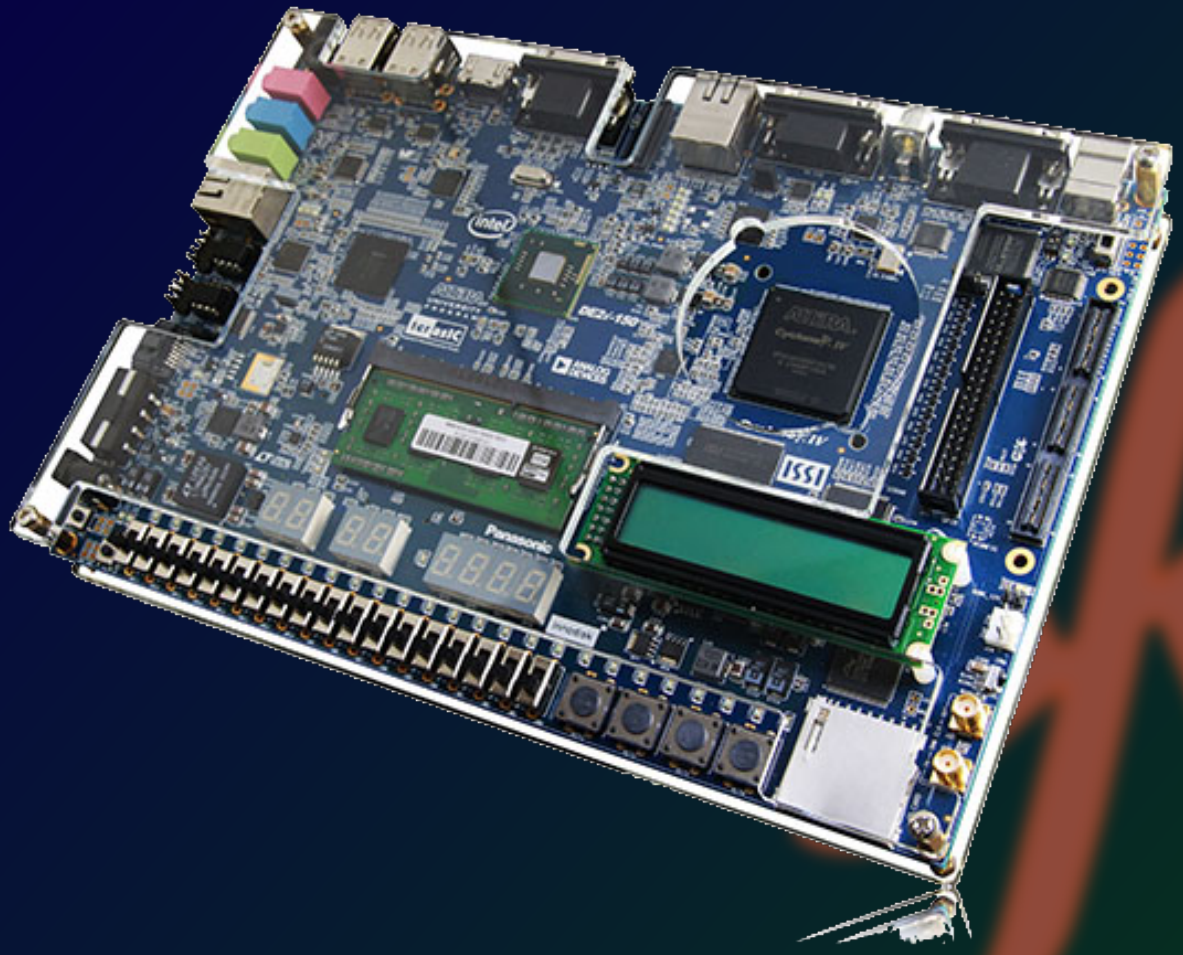
Placas legais

Altera DE2-SOC



Placas legais

Altera DE2i-150



Placas legais

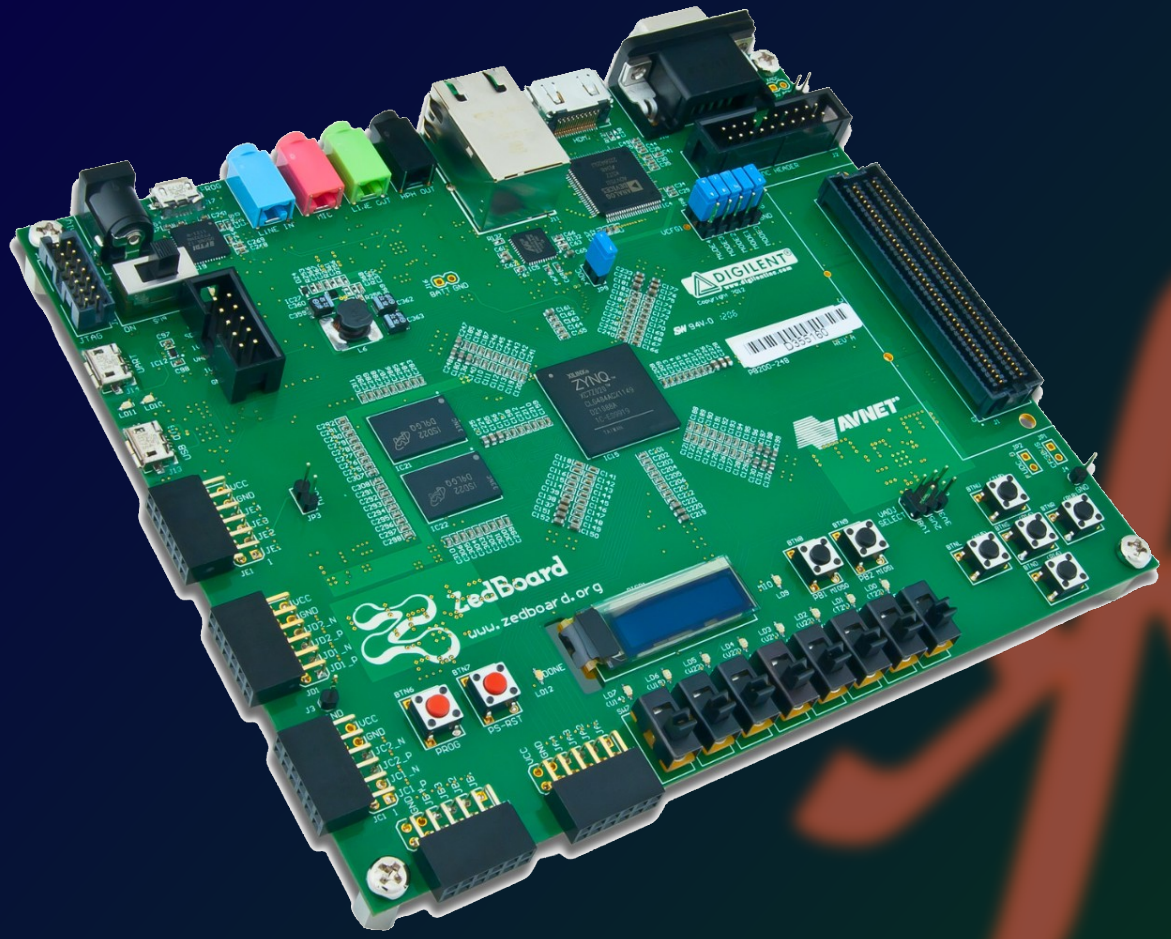
Parallella - Zynq Xilinx

Placas legais

Parallella - Zynq Xilinx

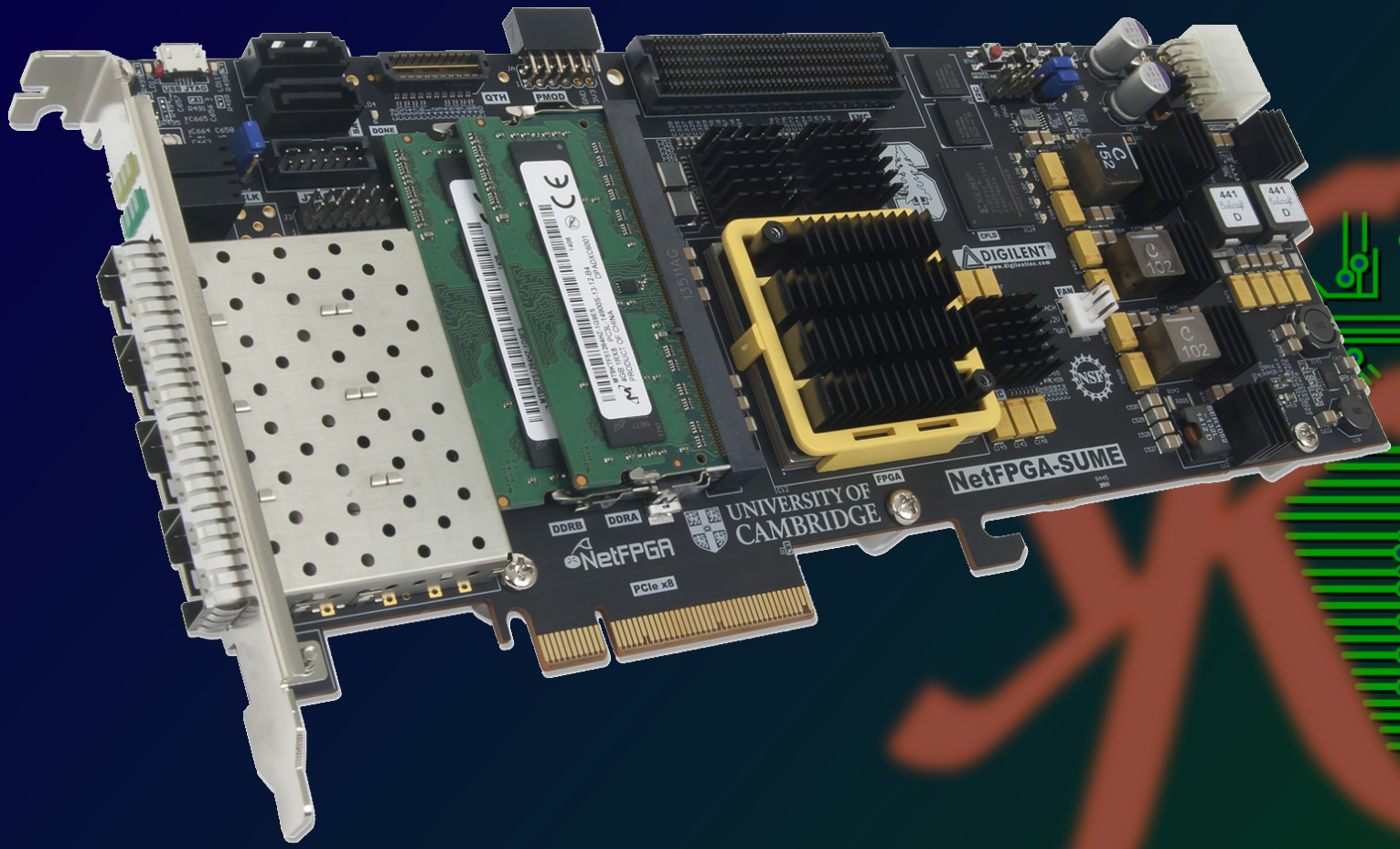
Placas legais

Zedboard - Zynq Xilinx



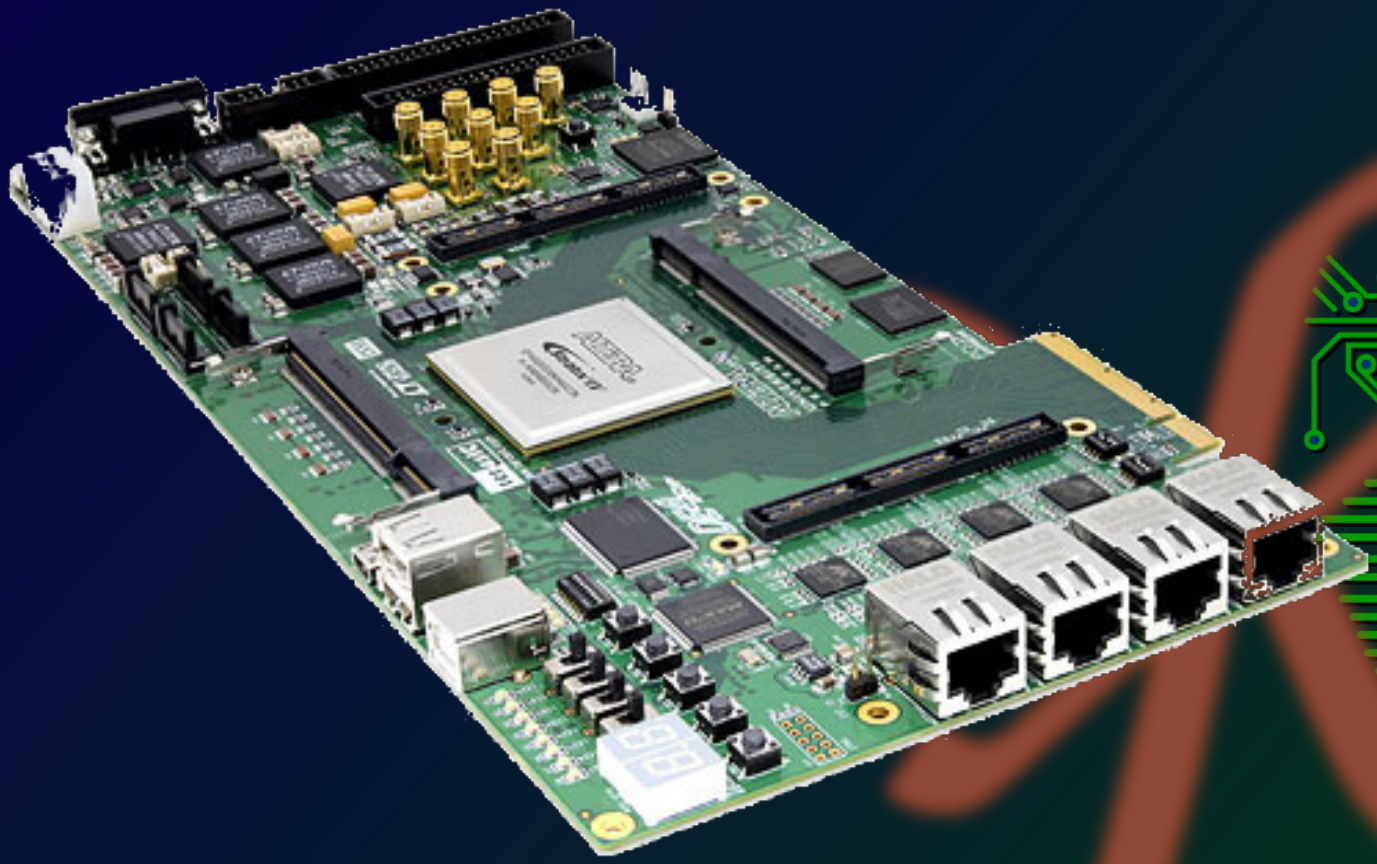
Não tão embarcado assim...

Virtex 7 - Net FPGA



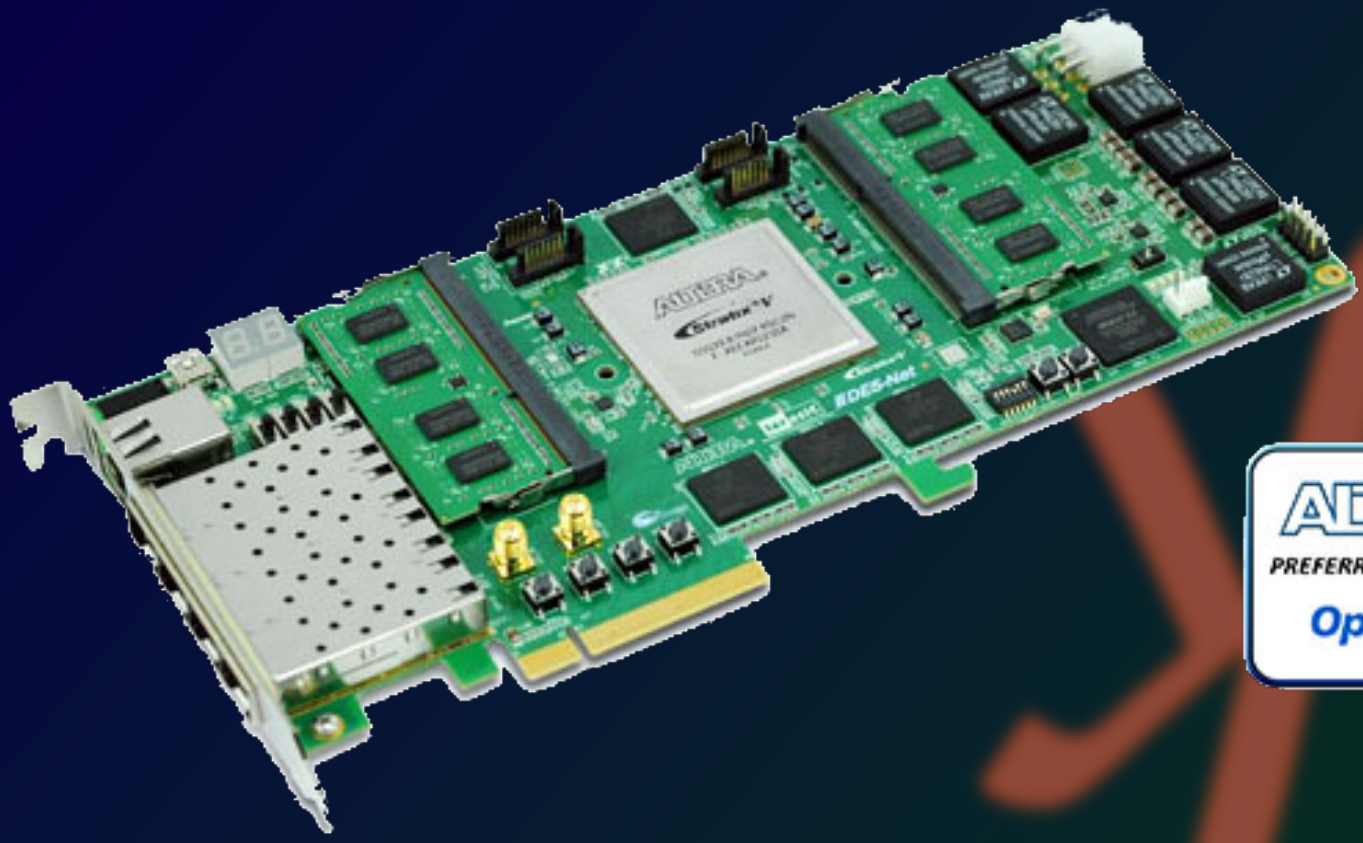
Não tão embarcado assim...

DE4 FPGA

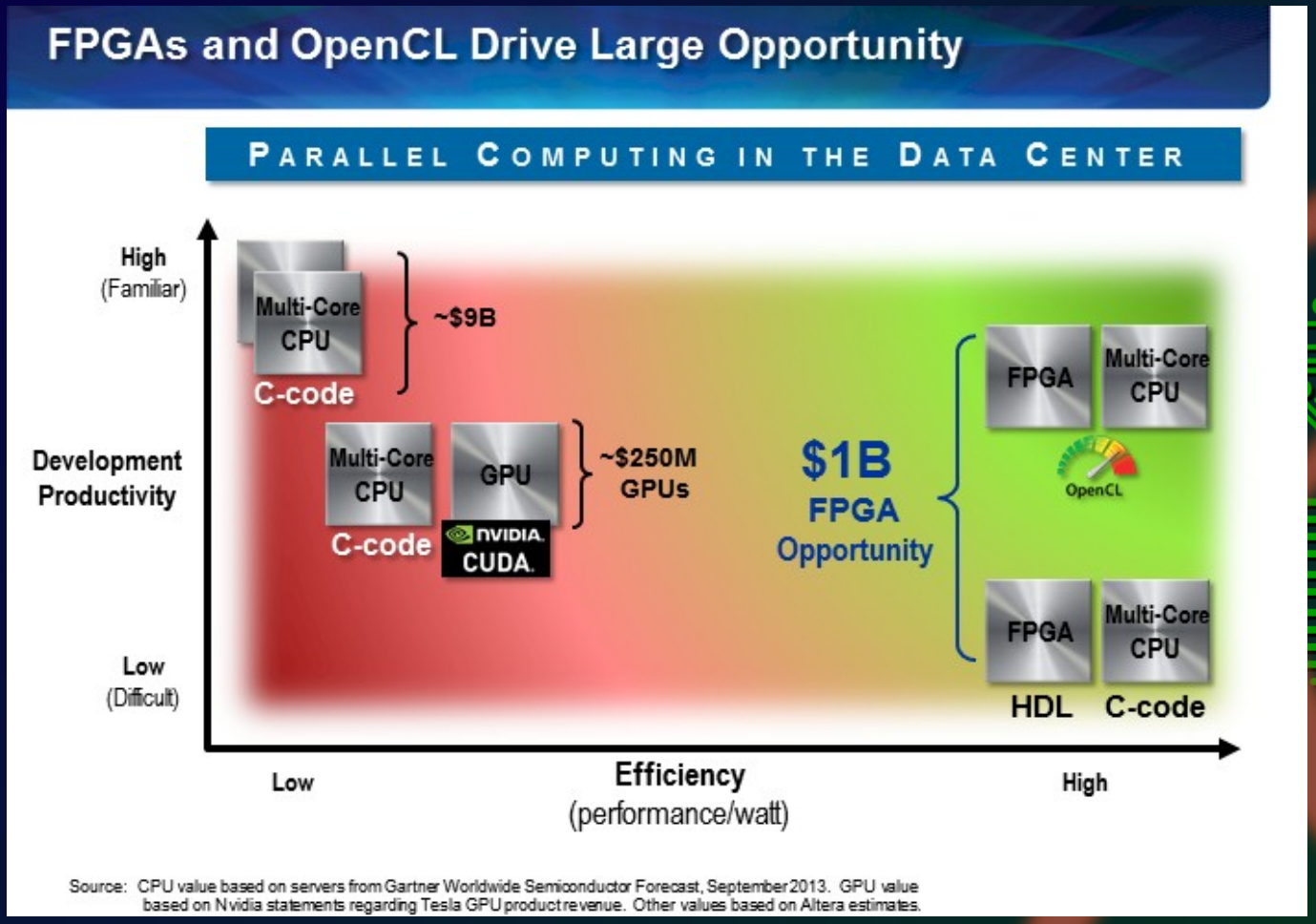


Não tão embarcado assim...

DE5 - Net FPGA



O FUTURO

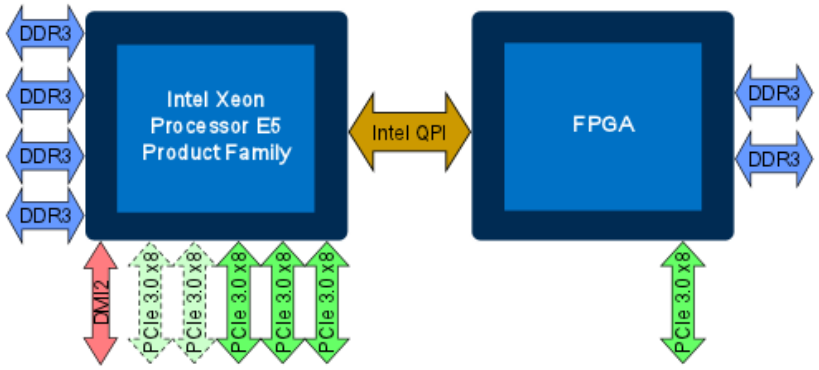

FPGAs and OpenCL Drive Large Opportunity
PARALLEL COMPUTING IN THE DATA CENTER
High
(Familiar)
Development
Productivity
Low
(Difficult)
Multi-Core CPU
C-code
~$9B
Multi-Core CPU
GPU
C-code
NVIDIA CUDA
~$250M GPUs
$1B
FPGA
Opportunity
FPGA
Multi-Core CPU
OpenCL
FPGA
Multi-Core CPU
HDL
C-code
Low
Efficiency
(performance/watt)
High
Source: CPU value based on servers from Gartner Worldwide Semiconductor Forecast, September 2013. GPU value based on Nvidia statements regarding Tesla GPU product revenue. Other values based on Altera estimates.

O FUTURO

Intel Xeon FPGA

Intel® Xeon® Processor + Field Programmable Gate Array Software Development Platform (SDP) Shipping Today

Software Development for Accelerating Workloads using Intel® Xeon® processors and coherently attached FPGA in-socket



Processor	Intel® Xeon® Processor E5
FPGA Module	Altera® Stratix® V
QPI Speed	6.4 GT/s full width (target 8.0 GT/s at full width)
Memory to FPGA Module	2 channels of DDR3 (up to 64 GB)
Expansion connector to FPGA Module	PCI Express® (PCIe) 3.0 x8 lanes - maybe used for direct I/O e.g. Ethernet
Features	Configuration Agent, Caching Agent, (optional) Memory Controller
Software	Accelerator Abstraction Layer (AAL) runtime, drivers, sample

Available as part of Intel & Altera co-sponsored Hardware Accelerator Research Program

IDF15 INTEL DEVELOPER FORUM

O FUTURO

Intel Xeon FPGA

Sumário

O que é AES?

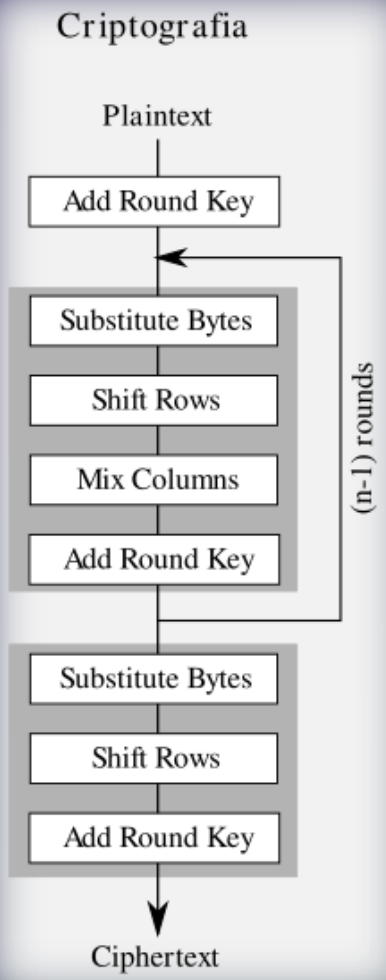
Advanced Encryption Standard

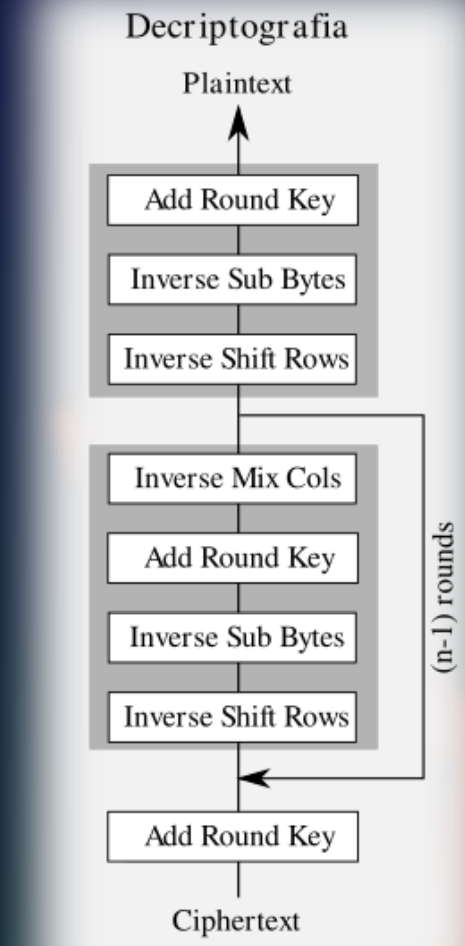
O AES é um algoritmo que usa **criptografia simétrica**, sendo capaz de codificar e decodificar **blocos** de informação de 128 bits.

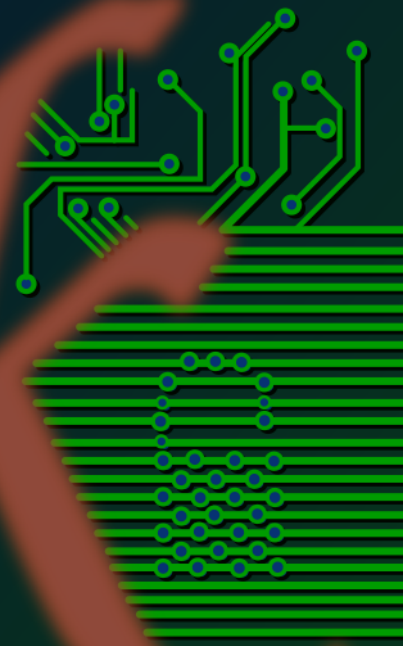
Parâmetros

Data (bits)	**Rounds**	**key size** (bits)
128	10	128
128	12	192
128	14	256

Codificador e Decodificador







Considerações sobre o AES

- Padrão Simetrico
- Robusto
- Implementação simples
- Amplamente difundido

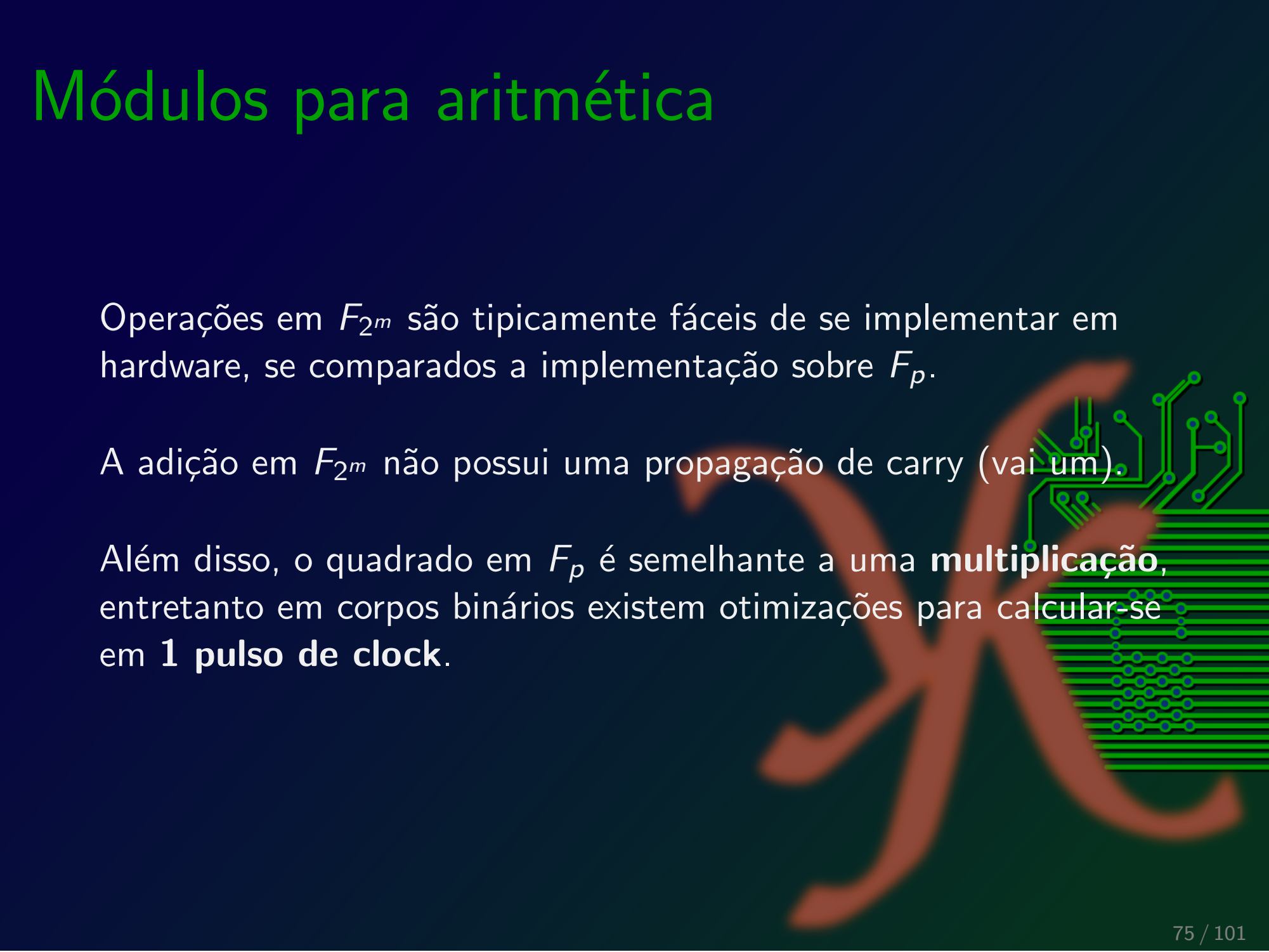
Sumário

Módulos para aritmética

Operações em F_{2^m} são tipicamente fáceis de se implementar em hardware, se comparados a implementação sobre F_p.

A adição em F_{2^m} não possui uma propagação de carry (vai um).

Além disso, o quadrado em F_p é semelhante a uma **multiplicação**, entretanto em corpos binários existem otimizações para calcular-se em **1 pulso de clock**.



Exemplo no corpo F_{2^4}

Em F_{2^4} o polinômio irredutível é $f(z) = z^4 + z^1 + z^0$.

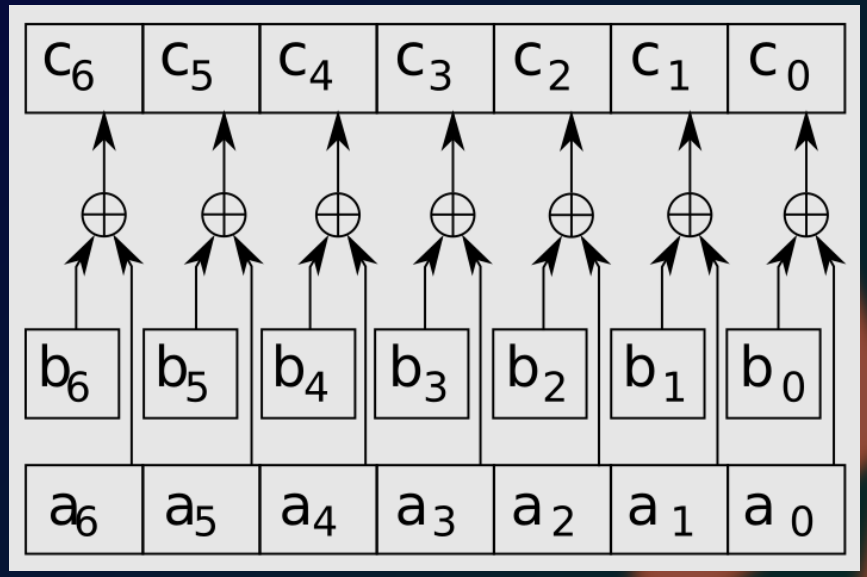
Exemplos de operações aritméticas:

- Adição: $(z^3 + z^2 + 1) \oplus (z^2 + z + 1) = z^3 + z$
- Subtração: $(z^3 + z^2 + 1) \oplus (z^2 + z + 1) = z^3 + z$
- Multiplicação: $(z^3 + z^2 + 1).(z^2 + z + 1) = z^5 + z + 1$
  $= z.z^4 + z + 1 = z(z + 1) + z + 1 = z^2 + z + z + 1 = z^2 + 1$
- Inversão: $(z^3 + z^2 + 1)^{-1} = z^2$
  Sabendo que $(z^3 + z^2 + 1).z^2 mod(z^4 + z + 1) = 1$

Adição

A adição consiste em uma operação de XOR bit a bit.

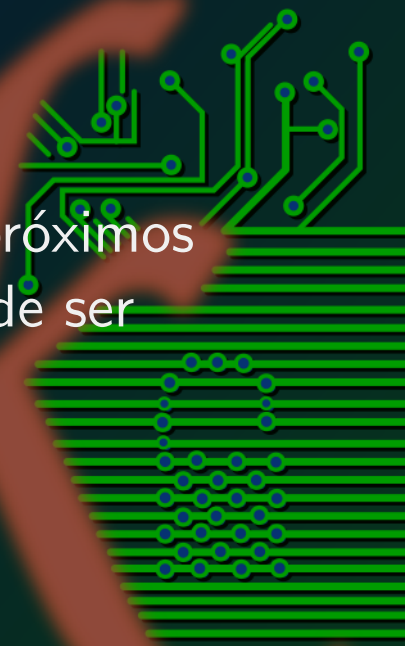
$$OUT \leftarrow U \oplus V$$



Redução

A redução pode ser executada como uma **redução arbitrária polinomial**, sendo neste caso a mesma operação sempre.

Se $f(z)$ é um trinômio, ou um pentanômio com termos próximos um dos outros, então a redução de $c(z)$ módulo $f(z)$ pode ser **processada eficientemente uma palavra por vez**.



Multiplicação

A multiplicação mais básica é a que consiste no método shift-soma:

$$a(z).b(z) = a_{m-1}.z^{m-1}.b(z) + \cdots + a_2.z^2.b(z) + a_0.z^0.b(z) + a_0.z^0.b(z)$$

Com o qual podemos usar o fato de que:

$$b(z).z = b_{m-1}.z^m + b_{m-2}.z^{m-1} + \cdots + b_2.z^3 + b_1.z^2 + b_0.z$$

$$b(z).z = b_{m-1}.r(z) + (b_{m-2}.z^{m-1} + \cdots + b_2.z^3 + b_1.z^2 + b_0.z) mod f(z)$$

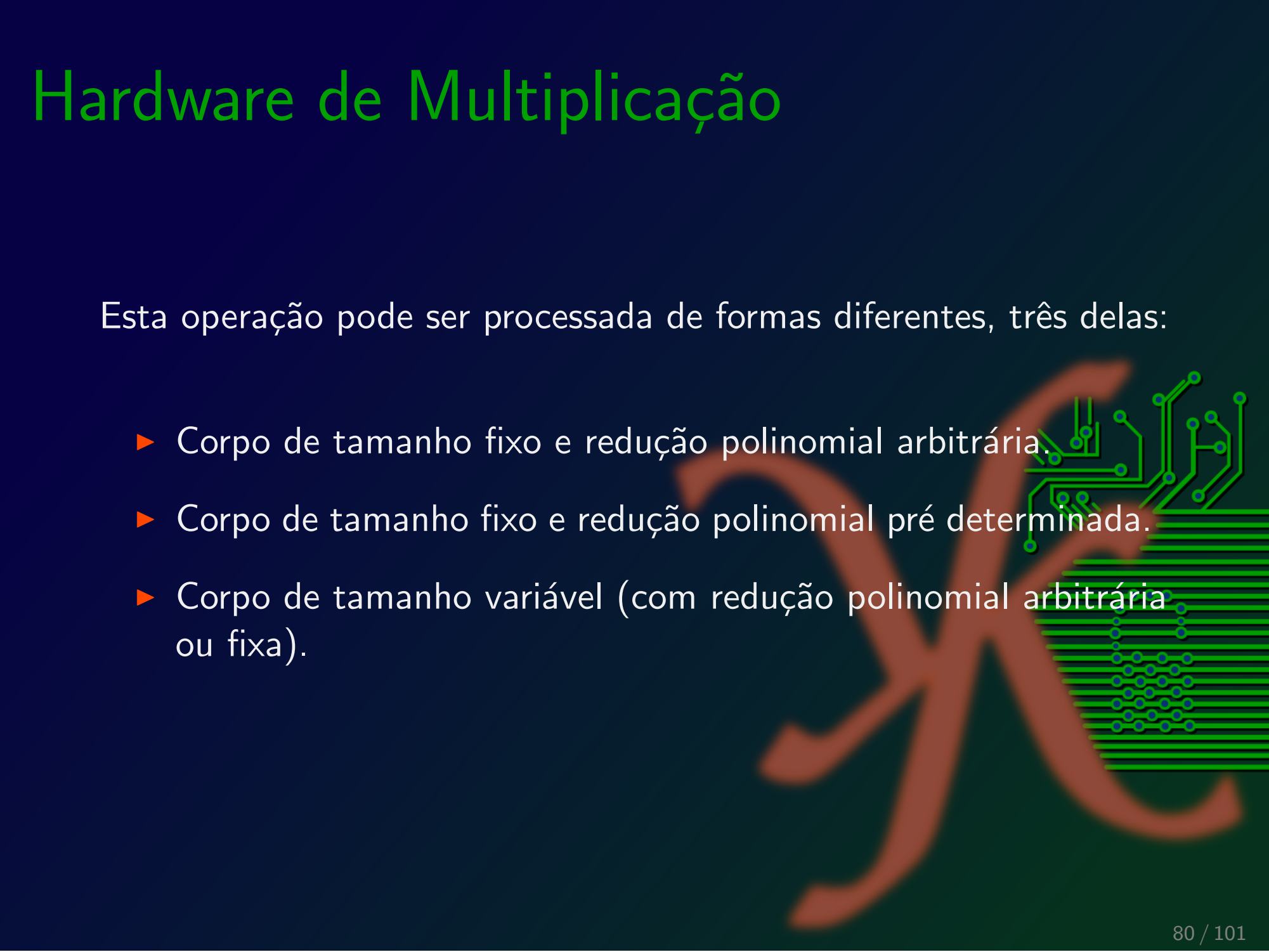
e assim calcular $b'(z).z$ em um pulso de clock.

Assim pode-se calcular $a(z).b(z)$ em m pulsos de clock.

Hardware de Multiplicação

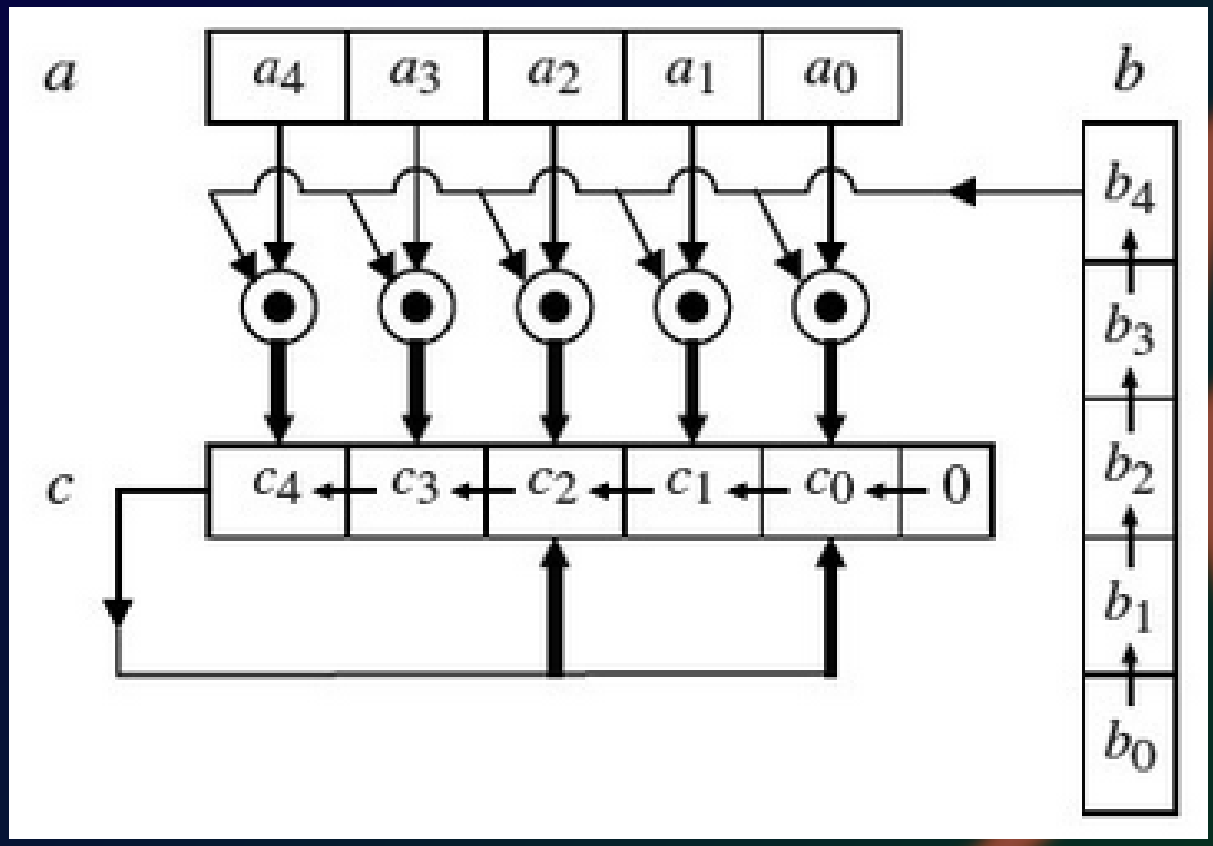
Esta operação pode ser processada de formas diferentes, três delas:

- Corpo de tamanho fixo e redução polinomial arbitrária.
- Corpo de tamanho fixo e redução polinomial pré determinada.
- Corpo de tamanho variável (com redução polinomial arbitrária ou fixa).



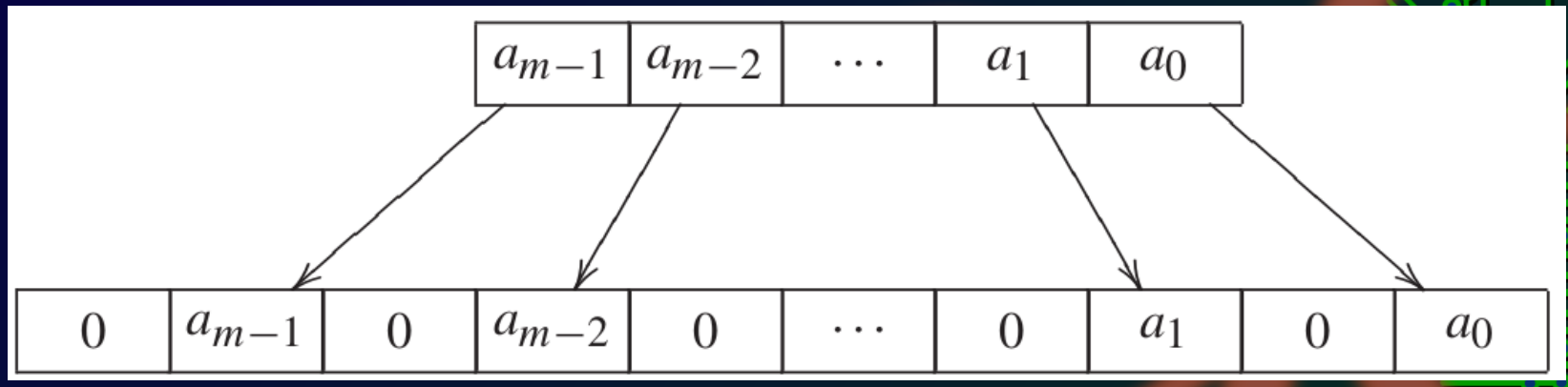
Multiplicação esquemático

Caso a redução polinomial $f(z)$ seja fixa pode-se projetar o multiplicador com um nível menor de complexidade, isso pois a redução precisa de um registrador menor.



Quadrado de um polinômio

A representação binária de $a(z)^2$ é obtida inserindo 0 entre os valores dos bits (posições ímpares do polinômio).



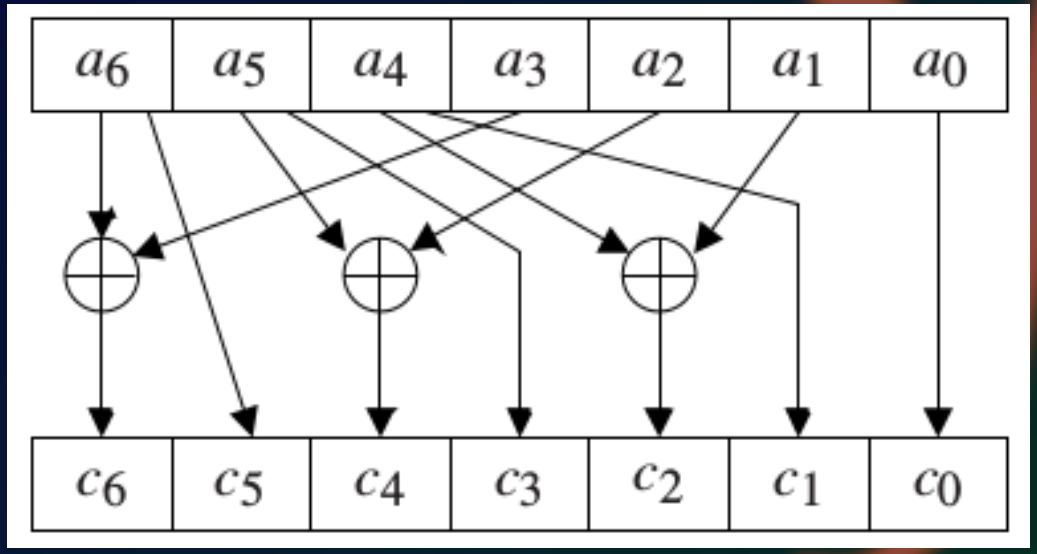
Exemplo de Hardware

Se considerarmos o corpo F_{2^7} cujo polinômio irredutível seja $f(z) = z^7 + z + 1$, temos que:

$$c = a^2$$

$$c = a_6 z^{12} + a_5 z^{10} + a_4 z^8 + a_3 z^6 + a_2 z^4 + a_1 z^2 + a_0 z^0$$

$$c = (a_6 + a_3) z^6 + a_6 z^5 + (a_5 + a_2) z^4 + a_5 z^3 + (a_4 + a_1) z^2 + a_4 z + a_0$$



Inversão

A operação **mais** difícil de se implementar em hardware do corpo é a inversão.

Existem dois tipos de algoritmos para se calcular a inversa de um polinômio:

- Aqueles que se baseiam no algoritmo de Euclides estendido (e suas variantes).
- Aqueles que usam a multiplicação do corpo para obter o resultado.

Inversão

Algoritmo de Euclides estendido

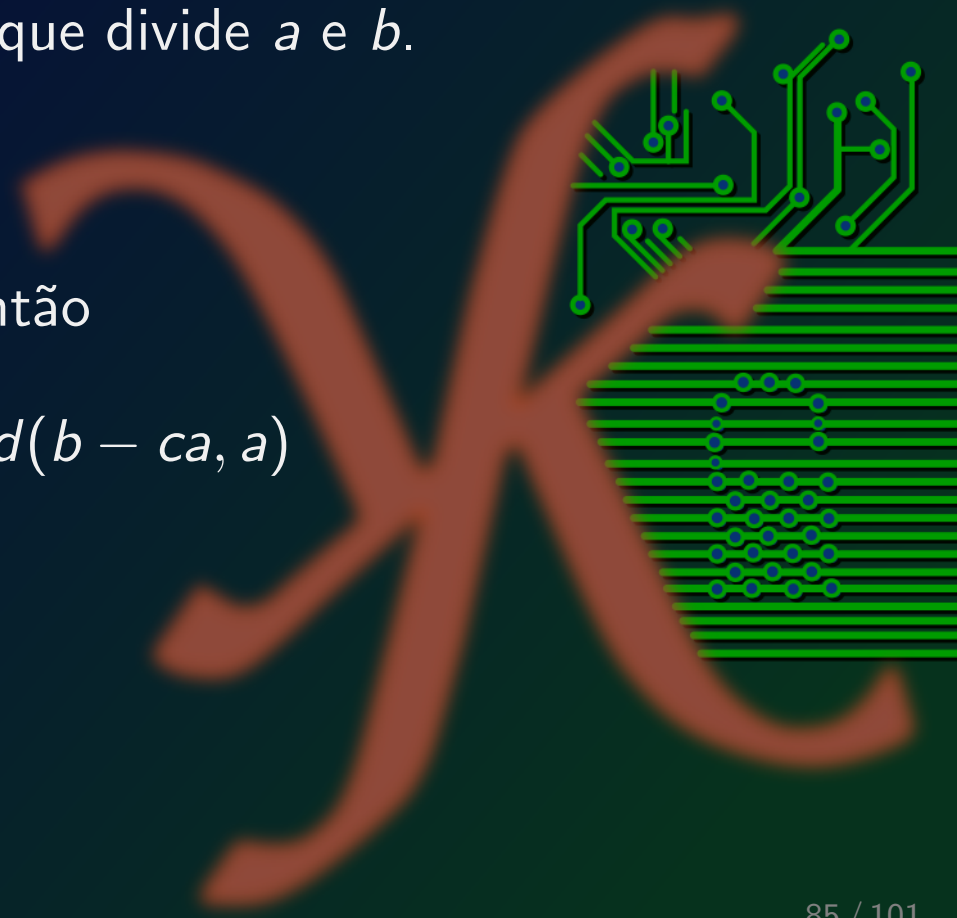
Sejam a e b polinômios binários, não mutuamente 0. O *Maior Divisor Comum / greatest common divisor* (gcd) de a e b é um polinômio binário d do maior grau que divide a e b.

Teorema

Sejam a e b polinômios binários, então

$$gcd(a, b) = gcd(b - ca, a)$$

para todo polinômio c.



A Inversa pela multiplicação

Sendo a um elemento não nulo de F_{2^m}, a inversão pela multiplicação usa o fato de que:

$$a^{-1} = a^{2^m-2}$$

Assim $2^m - 2 = \sum_{i=1}^{m-1} 2^i$, e com isso se tem que:

$$a^{-1} = a^{\sum_{i=1}^{m-1} 2^i} = \prod_{i=1}^{m-1} a^{2^i}$$

Com isso a^{-1} pode ser computado em $m-1$ quadrados e $m-2$ multiplicações.

Meio complicado, né?

Sumário

Como se proteger de ataques?

Considerando protocolos sem backdors, os ataques sobre maneiras de implementar recaem sobre:

- Ataques de medida de tempo
- Ataques de consumo de potência
- "Ladrões de galinha"

Ataques de medida

- O algoritmo depende dos elementos
- Em busca de “otimizações” como multiplicar por 0000...0010
- Equações não completas
- Depende de “conferências”

Soluções

- Usar equações completas
- Não usar sistemas que necessitam de checagem
- Mesmo parecendo burro, faça com que sempre tenha o mesmo tempo

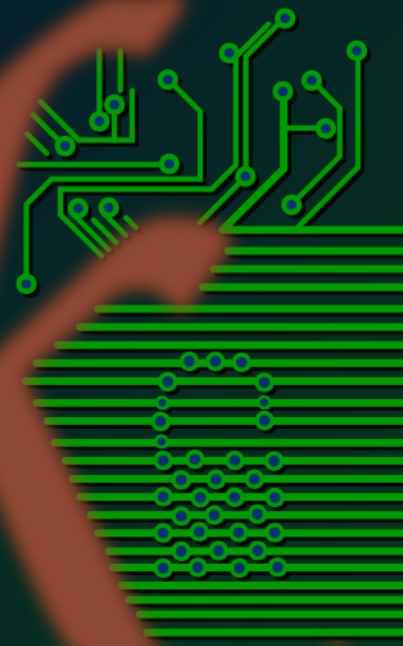
Perspectivas futuras

- Curvas Elípticas (mais) Robustas
- Pós-quântica

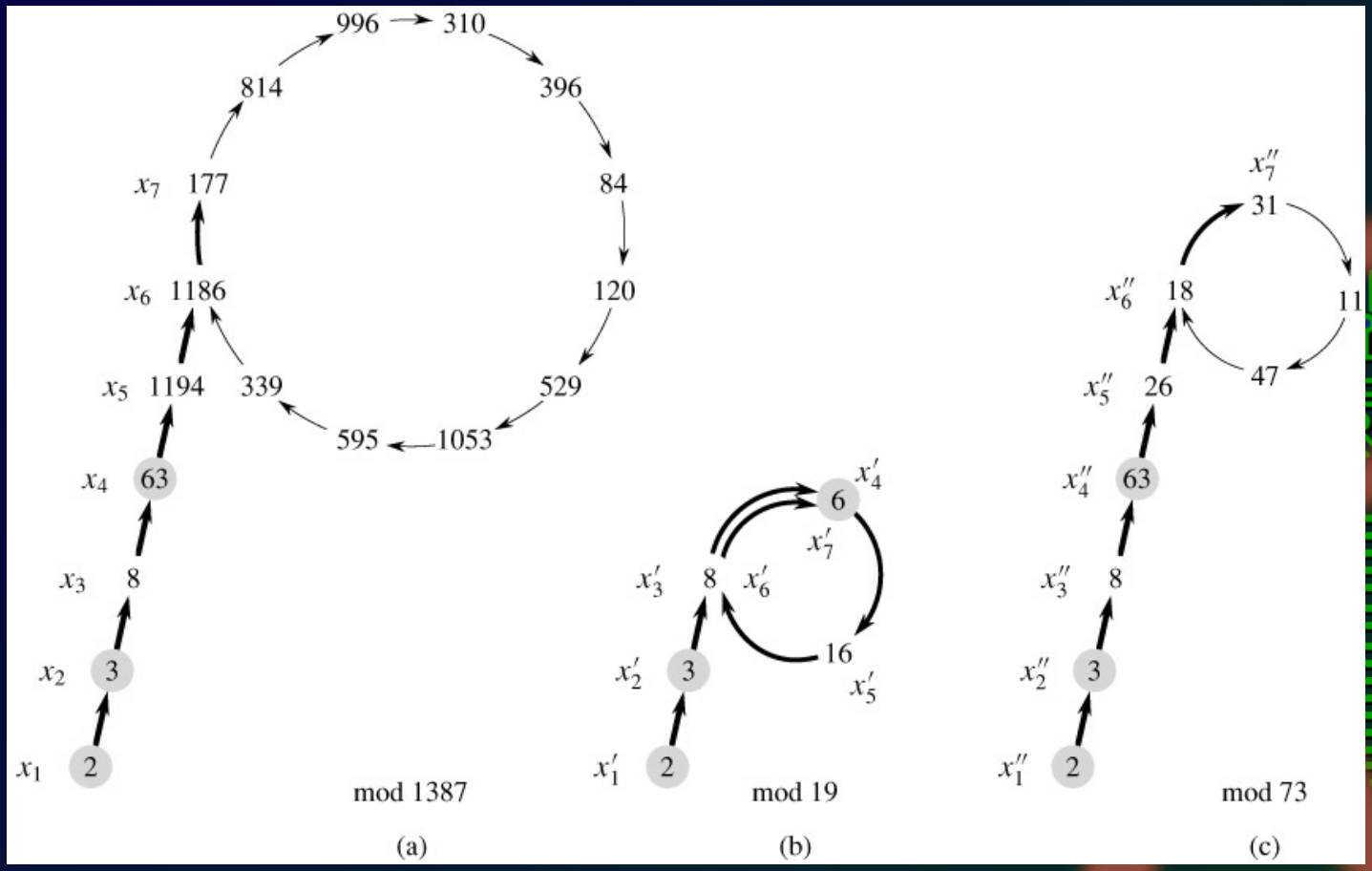
Sumário

Pollard Rho

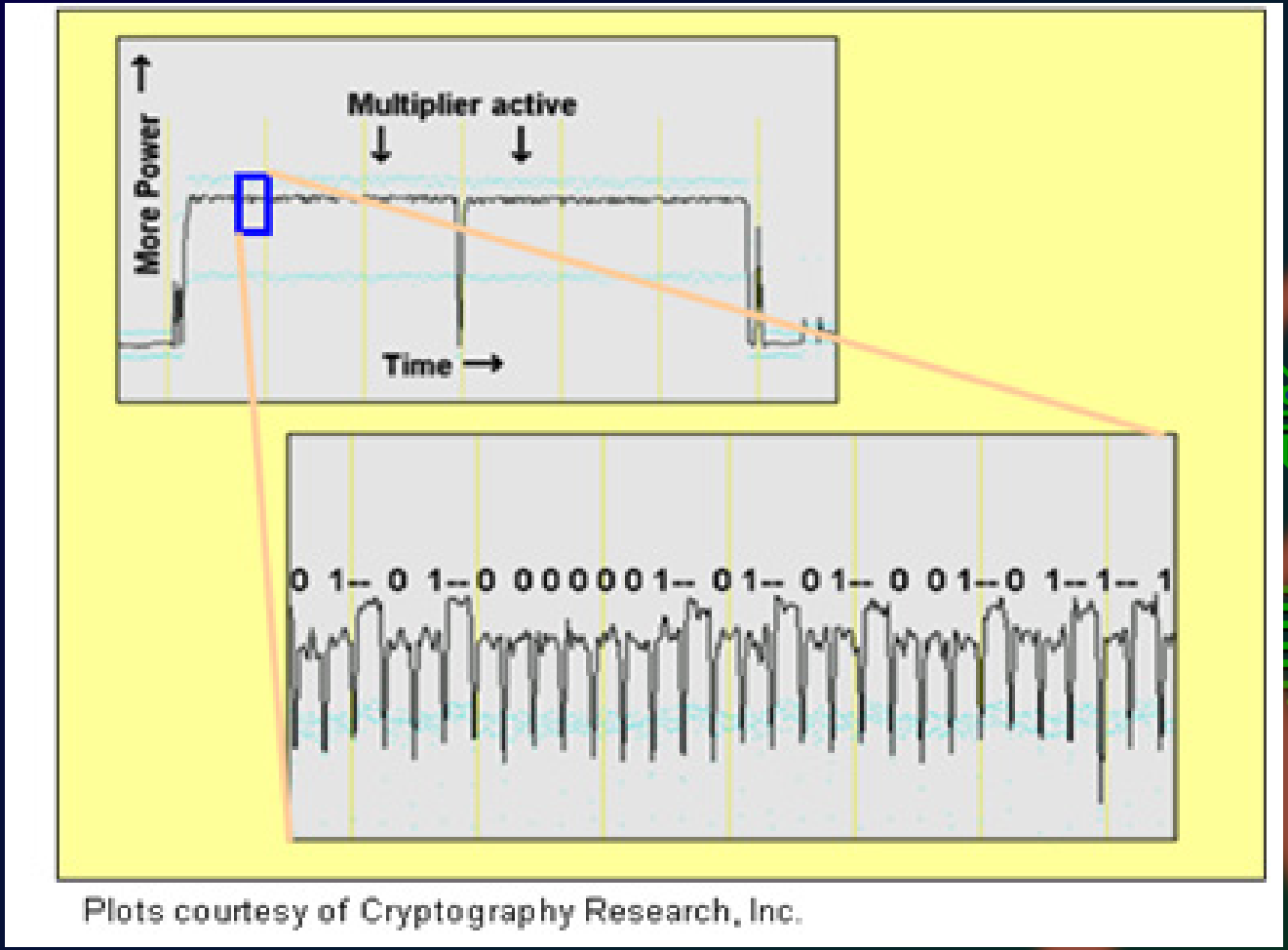
```
POLLARD-RHO(n)
 1  i = 1
 2  x_1 = RANDOM(0, n − 1)
 3  y = x_1
 4  k = 2
 5  while TRUE
 6      i = i + 1
 7      x_i = (x_{i−1}^2 − 1) mod n
 8      d = gcd(y − x_i, n)
 9      if d ≠ 1 and d ≠ n
10          print d
11      if i == k
12          y = x_i
13          k = 2k
```



Pollard Rho

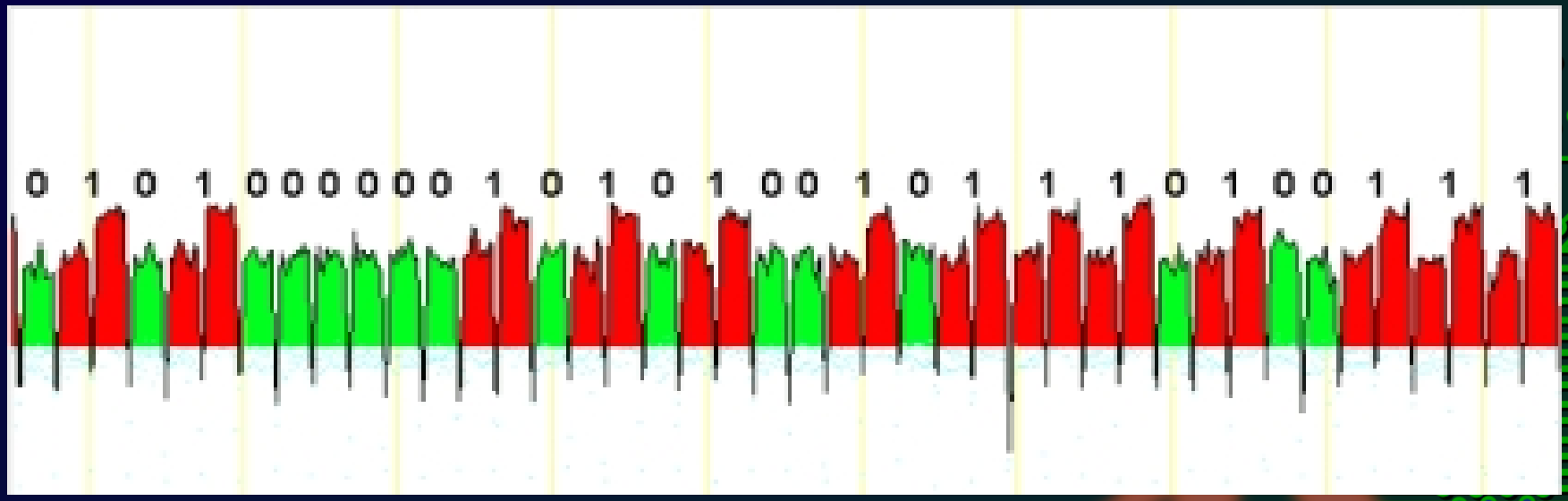

996
310
396
84
120
529
1053
595
339
1194
1186
177
814
x_1 2
x_2 3
x_3 8
x_4 63
x_5
x_6
x_7
mod 1387
(a)
x'_1 2
x'_2 3
x'_3 8
x'_4 6
x'_5 16
x'_6
x'_7
mod 19
(b)
x''_1 2
x''_2 3
x''_3 8
x''_4 63
x''_5 26
x''_6 18
x''_7 31
11
47
mod 73
(c)

Side channel attack

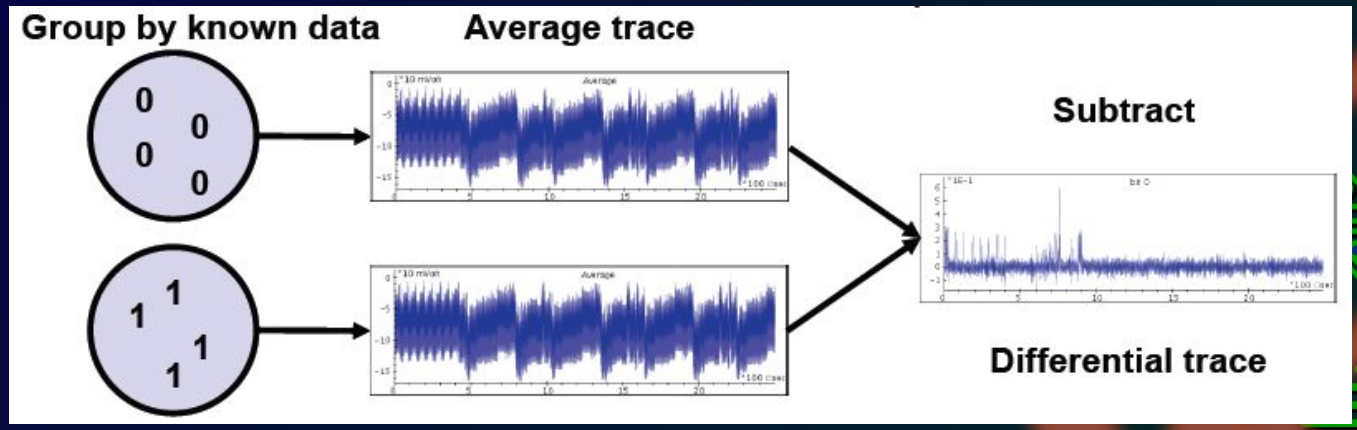


Plots courtesy of Cryptography Research, Inc.

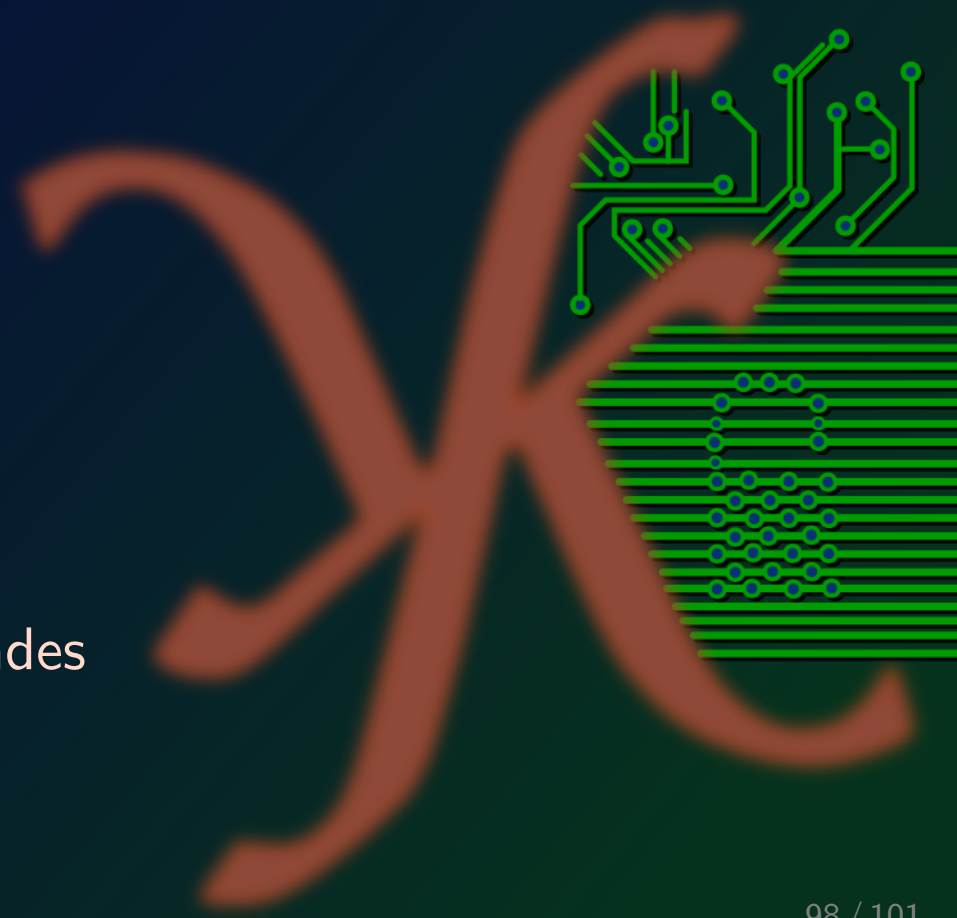
Side channel attack



Side channel attack

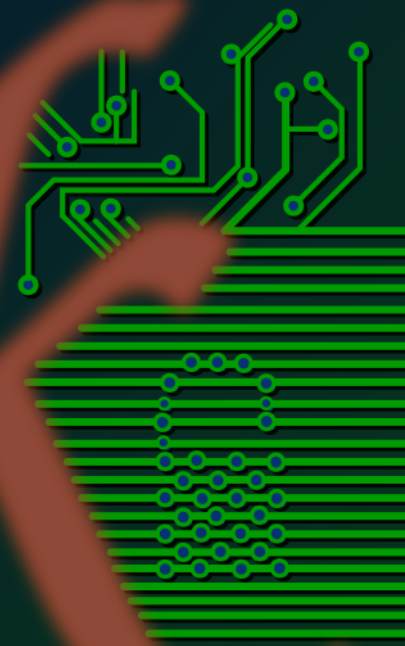


Sumário



Ataque de força bruta

- Todas as precauções aqui descrevem proteção
- Se meu objetivo é atacar posso descarta-las
- Mesma arquitetura, só alterar o programa
- Eficiência de no melhor caso 2x e no pior caso 1x



Muitas dúvidas?

Obrigado pela atenção

E-mail: **luckas.judocka@gmail.com**