NELSON MANDELA UNIVERSITY
The 7th Biennial African School of Fundamental Physics and Applications
28 November - 9 December 2022

Quantum Computing

Lecture 1 : Introduction to Quantum Information
Lecture 2 : Quantum Mechanics for Quantum Information
Lecture 3 : Quantum Gates and Quantum Circuits
Lecture 4 : Quantum Computing Algorithms

Mourad Telmini
mourad.telmini@fst.utm.tn

Department of Physics
Faculty of Science of Tunis
University of Tunis El Manar

5 December 2022

Acknowledgements

- This series of lectures is based on the course of Quantum Information I'm giving at University of Tunis El Manar (Faculty of Science of Tunis, Departement of Physics) for 2nd year (M2) master candidates in Nanophysics and Nanotechnology.

Acknowledgements

- This series of lectures is based on the course of Quantum Information I'm giving at University of Tunis El Manar (Faculty of Science of Tunis, Departement of Physics) for 2nd year (M2) master candidates in Nanophysics and Nanotechnology.

- It is an updated version of the lectures given at the 2020 edition of ASP.

Acknowledgements

- This series of lectures is based on the course of Quantum Information I'm giving at University of Tunis El Manar (Faculty of Science of Tunis, Departement of Physics) for 2nd year (M2) master candidates in Nanophysics and Nanotechnology.

كلية العلوم بتونس
FACULTE DES SCIENCES DE TUNIS

- It is an updated version of the lectures given at the 2020 edition of ASP.
- Most of graphics, especially those of quantum circuits are taken from IBM Qiskit website www.qiskit.org

Qiskit

Context

The Nobel Prize in Physics 2022

Ill. Niklas Elmehed © Nobel Prize Outreach

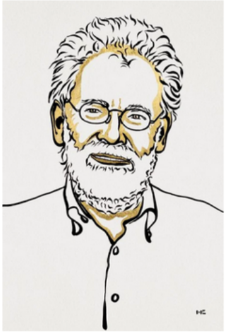
Alain Aspect

Prize share: 1/3

Ill. Niklas Elmehed © Nobel Prize Outreach

John F. Clauser

Prize share: 1/3



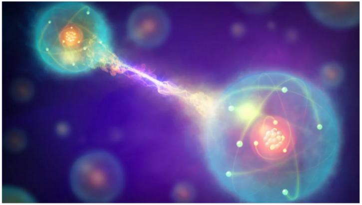
Ill. Niklas Elmehed © Nobel Prize Outreach

Anton Zeilinger

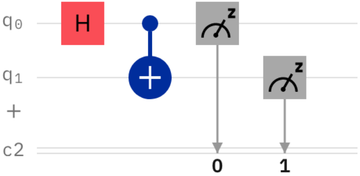
Prize share: 1/3

Nobel Prize in Physics 2022

The Nobel Prize in Physics 2022 was awarded jointly to Alain Aspect, John F. Clauser and Anton Zeilinger "for experiments with **entangled photons**, establishing the violation of Bell inequalities and pioneering **quantum information science**"



(Image credit: MARK GARLICK/SCIENCE PHOTO LIBRARY via Getty Images)



Quantum computing landscape 2022

source : The Quantum Insider September 2022

IBM Osprey Quantum Processor 2022

IBM

IBM **Newsroom** Home **Latest News** Press Releases Media Contacts Media Center Leadership Global Newsrooms Subscribe

IBM Unveils 400 Qubit-Plus Quantum Processor and Next-Generation IBM Quantum System Two

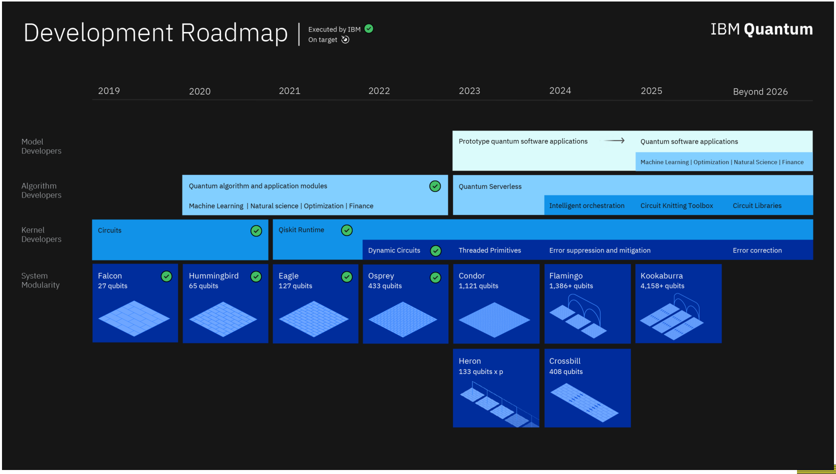
Company Outlines Path Towards Quantum-Centric Supercomputing with New Hardware, Software, and System Breakthrough

Nov 9, 2022

Dario Gil, Jay Gambetta and Jerry Chow holding the new 433 qubit 'IBM Osprey' processor

FACULTE DES SCIENCES DE TUNIS

IBM development roadmap 2022



Quantum computing to test entanglement

MIT News
ON CAMPUS AND AROUND THE WORLD

SUBSCRIBE | BROWSE | SEARCH NEWS

MIT researchers use quantum computing to observe entanglement

Researchers at the Center for Theoretical Physics lead work on testing quantum gravity on a quantum processor.

Julia C. Keller | School of Science
December 1, 2022

PRESS INQUIRIES

A representation of a traversable wormhole with quantum information passing through.

Image: A. Mueller/inqnet

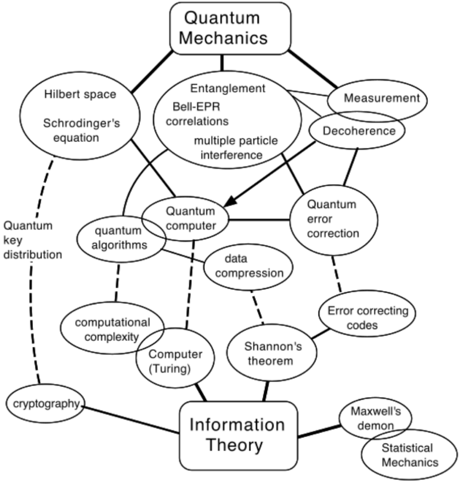
FACULTE DES SCIENCES DE TUNIS

Outline

FACULTE DES SCIENCES DE TUNIS

Outline

Quantum Mechanics and Information Theory



[1]

[1] Andrew Steane 1998 Rep. Prog. Phys. 61 117

Shannon Information Theory

Claude Shannon
1916-2001

- Defined the quantity of information produced by a source by a formula similar to the equation that defines thermodynamic entropy in physics :

$$H = -\sum_i p_i \log_2 p_i$$

Shannon Information Theory

Claude Shannon
1916-2001

- Defined the quantity of information produced by a source by a formula similar to the equation that defines thermodynamic entropy in physics :

$$H = -\sum_i p_i \log_2 p_i$$

- Analyzed the ability to send information through a communications channel, proving the existence of a maximum transmission rate that could not be exceeded (bandwidth).

Shannon Information Theory

Claude Shannon
1916-2001

- Defined the quantity of information produced by a source by a formula similar to the equation that defines thermodynamic entropy in physics :

$$H = -\sum_i p_i \log_2 p_i$$

- Analyzed the ability to send information through a communications channel, proving the existence of a maximum transmission rate that could not be exceeded (bandwidth).
- Demonstrated mathematically that even in a noisy channel with a low bandwidth, essentially perfect, error-free communication could be achieved by keeping the transmission rate within the channel's bandwidth and by using error-correcting schemes (redundancy)

Noiseless and noisy Shannon theorems

[2]

- Noiseless channel case:

Claude Shannon
1916-2001

[2] S. Barnett, Les Houches Summer School lectures 2009

Noiseless and noisy Shannon theorems

[2]

- Noiseless channel case:
  - FRCN SCHL F PHSCS HS NTRSTNG LCTRS

Claude Shannon
1916-2001

[2] S. Barnett, Les Houches Summer School lectures 2009

Noiseless and noisy Shannon theorems

[2]

Claude Shannon
1916-2001

- Noiseless channel case:
  - FRCN SCHL F PHSCS HS NTRSTNG LCTRS
  - AFRICAN SCHOOL OF PHYSICS HAS INTERESTING LECTURES

[2] S. Barnett, Les Houches Summer School lectures 2009

Noiseless and noisy Shannon theorems

[2]

Claude Shannon
1916-2001

- Noiseless channel case:
  - FRCN SCHL F PHSCS HS NTRSTNG LCTRS
  - AFRICAN SCHOOL OF PHYSICS HAS INTERESTING LECTURES
- Noisy channel case :
  - WNTM NARMQN THRS S FN

[2] S. Barnett, Les Houches Summer School lectures 2009

Noiseless and noisy Shannon theorems

2

Claude Shannon
1916-2001

- Noiseless channel case:
  - FRCN SCHL F PHSCS HS NTRSTNG LCTRS
  - AFRICAN SCHOOL OF PHYSICS HAS INTERESTING LECTURES
- Noisy channel case :
  - WNTM NARMQN THRS S FN
  - WUANTFM INAORMAQION THEORS US FUN

[2] S. Barnett, Les Houches Summer School lectures 2009

Noiseless and noisy Shannon theorems

[2]

Claude Shannon
1916-2001

- Noiseless channel case:
  - FRCN SCHL F PHSCS HS NTRSTNG LCTRS
  - AFRICAN SCHOOL OF PHYSICS HAS INTERESTING LECTURES
- Noisy channel case :
  - WNTM NARMQN THRS S FN
  - WUANTFM INAORMAQION THEORS US FUN
  - QUANTUM INFORMATION THEORY IS FUN

[2] S. Barnett, Les Houches Summer School lectures 2009

Shannon Theory

For more details about Shannon Information Theory :

From Classical to Quantum Shannon Theory

Mark M. Wilde
Hearne Institute for Theoretical Physics
Department of Physics and Astronomy
Center for Computation and Technology
Louisiana State University
Baton Rouge, Louisiana 70803, USA

July 16, 2019

Introduction to Quantum Information

- The computers that we use every day, process information according to the rules of classical binary logic.

FACULTE DES SCIENCES DE TUNIS

Introduction to Quantum Information

- The computers that we use every day, process information according to the rules of classical binary logic.
- The hardware part operates according to the rules of quantum mechanics (semiconductors, transistors, etc.), but the quantum properties at the fundamental scale (superposition, entanglement, non-locality, etc.) are not fully exploited.

Introduction to Quantum Information

- The computers that we use every day, process information according to the rules of classical binary logic.
- The hardware part operates according to the rules of quantum mechanics (semiconductors, transistors, etc.), but the quantum properties at the fundamental scale (superposition, entanglement, non-locality, etc.) are not fully exploited.
- The purpose of Quantum Information Theory is precisely to take advantage of these properties in order to perform tasks which are impossible to realize with classical computers.

Introduction to Quantum Information

- The computers that we use every day, process information according to the rules of classical binary logic.
- The hardware part operates according to the rules of quantum mechanics (semiconductors, transistors, etc.), but the quantum properties at the fundamental scale (superposition, entanglement, non-locality, etc.) are not fully exploited.
- The purpose of Quantum Information Theory is precisely to take advantage of these properties in order to perform tasks which are impossible to realize with classical computers.
- The most known applications of quantum information are Quantum Computing, with the focus on the physical implementation of a universal quantum computer, and Quantum Cryptography for the secure transmission of information.

Elements of binary logic

- All information processed by computers (numbers, texts, images, videos, ...) is encoded using the binary system.

Elements of binary logic

- All information processed by computers (numbers, texts, images, videos, ...) is encoded using the binary system.
- Each elementary character has a specific binary code in the form of a sequence of 0 and 1, called bits.

Elements of binary logic

- All information processed by computers (numbers, texts, images, videos, ...) is encoded using the binary system.
- Each elementary character has a specific binary code in the form of a sequence of 0 and 1, called bits.
- The simplest example is that of integer numbers $n \in \mathbb{N}$.

Elements of binary logic

- All information processed by computers (numbers, texts, images, videos, ...) is encoded using the binary system.
- Each elementary character has a specific binary code in the form of a sequence of 0 and 1, called bits.
- The simplest example is that of integer numbers $n \in \mathbb{N}$.
- Each integer n is written on binary basis as the sum of terms to the power of 2. For example:

$$17 = 16 + 1 = 2^4 + 1 = 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$

Elements of binary logic

- All information processed by computers (numbers, texts, images, videos, ...) is encoded using the binary system.
- Each elementary character has a specific binary code in the form of a sequence of 0 and 1, called bits.
- The simplest example is that of integer numbers $n \in \mathbb{N}$.
- Each integer n is written on binary basis as the sum of terms to the power of 2. For example:

$$17 = 16 + 1 = 2^4 + 1 = 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$

- The binary code of the number 17 is the sequence of bits 0 or 1 which are the coefficients multiplying the power of 2 in the previous writing:

$$17 = 10001$$

where we see that we needed 5 bits to encode this number.

Elements of classical binary logic

- But generally speaking, how many bits do you need to encode a given number?

Elements of classical binary logic

- But generally speaking, how many bits do you need to encode a given number?
- Or, given a certain number of bits n what is the largest integer that can be encoded?

Elements of classical binary logic

- But generally speaking, how many bits do you need to encode a given number?
- Or, given a certain number of bits n what is the largest integer that can be encoded?
- With n bits, which form what is called a register, we can encode 2^n numbers ranging from 0 to $2^n - 1$.

Elements of classical binary logic

- But generally speaking, how many bits do you need to encode a given number?
- Or, given a certain number of bits n what is the largest integer that can be encoded?
- With n bits, which form what is called a register, we can encode 2^n numbers ranging from 0 to $2^n - 1$.
- For example with 3 bits, we can code 8 numbers: the integers going from 0 to 7.

Integer	Binary code
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Elements of Binary Algebra

- To encode the integers 8 and 9 and thus be able to encode all the digits of the decimal system, we need at least 4 bits:

Integer	Binary Code	Integer	Binary Code
0	0000	8	1000
1	0001	9	1001
2	0010	-	1010
3	0011	-	1011
4	0100	-	1100
5	0101	-	1101
6	0110	-	1110
7	0111	-	1111

Elements of Binary Algebra

- To encode the integers 8 and 9 and thus be able to encode all the digits of the decimal system, we need at least 4 bits:

Integer	Binary Code	Integer	Binary Code
0	0000	8	1000
1	0001	9	1001
2	0010	-	1010
3	0011	-	1011
4	0100	-	1100
5	0101	-	1101
6	0110	-	1110
7	0111	-	1111

- We see that for the first 8 numbers we have added a 0 to the left.

Elements of Binary Algebra

- To encode the integers 8 and 9 and thus be able to encode all the digits of the decimal system, we need at least 4 bits:

Integer	Binary Code	Integer	Binary Code
0	0000	8	1000
1	0001	9	1001
2	0010	-	1010
3	0011	-	1011
4	0100	-	1100
5	0101	-	1101
6	0110	-	1110
7	0111	-	1111

- We see that for the first 8 numbers we have added a 0 to the left.
- In addition, the 4-bit register makes it possible to encode 6 other symbols.

hexadecimal System

- The hexadecimal system is the one where these symbols are A, B, C, D, E, F respectively:

Integer	Binary Code	Integer	Binary Code
0	0000	8	1000
1	0001	9	1001
2	0010	A	1010
3	0011	B	1011
4	0100	C	1100
5	0101	D	1101
6	0110	E	1110
7	0111	F	1111

hexadecimal System

- The hexadecimal system is the one where these symbols are A, B, C, D, E, F respectively:

Integer	Binary Code	Integer	Binary Code
0	0000	8	1000
1	0001	9	1001
2	0010	A	1010
3	0011	B	1011
4	0100	C	1100
5	0101	D	1101
6	0110	E	1110
7	0111	F	1111

- Here the symbols A, B, C, D, E, F do not represent letters of the alphabet, but numerical values in the hexadecimal system corresponding respectively to the integers 10, 11, 12, 13, 14 and 15.

ASCII Code

- To encode letters and main symbols as well, the ASCII coding system (American Standard Code for Information Interchange) has been implemented.

ASCII Code

- To encode letters and main symbols as well, the ASCII coding system (American Standard Code for Information Interchange) has been implemented.
- Based on a 7-bit register (then extended to 8 bits also called *byte*), it allows 128 (then 256) symbols to be encoded, covering most of the needs to process any type of information whether in the form of usual numbers, letters or symbols.

ASCII Code

- To encode letters and main symbols as well, the ASCII coding system (American Standard Code for Information Interchange) has been implemented.
- Based on a 7-bit register (then extended to 8 bits also called *byte*), it allows 128 (then 256) symbols to be encoded, covering most of the needs to process any type of information whether in the form of usual numbers, letters or symbols.
- For example, the ASCII code for the number zero is 0110000, or 30 in decimal and 48 in hexadecimal!

ASCII Code

- To encode letters and main symbols as well, the ASCII coding system (American Standard Code for Information Interchange) has been implemented.
- Based on a 7-bit register (then extended to 8 bits also called *byte*), it allows 128 (then 256) symbols to be encoded, covering most of the needs to process any type of information whether in the form of usual numbers, letters or symbols.
- For example, the ASCII code for the number zero is 0110000, or 30 in decimal and 48 in hexadecimal!
- the letter a corresponds to the binary code 1100001, which is that of the number 97 (decimal) or 61 (hexadecimal).

ASCII Code

- To encode letters and main symbols as well, the ASCII coding system (American Standard Code for Information Interchange) has been implemented.
- Based on a 7-bit register (then extended to 8 bits also called *byte*), it allows 128 (then 256) symbols to be encoded, covering most of the needs to process any type of information whether in the form of usual numbers, letters or symbols.
- For example, the ASCII code for the number zero is 0110000, or 30 in decimal and 48 in hexadecimal!
- the letter a corresponds to the binary code 1100001, which is that of the number 97 (decimal) or 61 (hexadecimal).
- See Appendix 1 for more details

Classical Operations

- Given a register of n bits (for example one byte), what operations can we do with it?

0	1	1	0	0	0	0	1

Classical Operations

- Given a register of n bits (for example one byte), what operations can we do with it?

0	1	1	0	0	0	0	1

- The value of one or more bits can be changed, which makes it possible to scan the entire field of possible codes.

Classical Operations

- Given a register of n bits (for example one byte), what operations can we do with it?

0	1	1	0	0	0	0	1

- The value of one or more bits can be changed, which makes it possible to scan the entire field of possible codes.
- Any other operation, like swapping two bits, ultimately comes down to the first operation.

Classical Operations

- Given a register of n bits (for example one byte), what operations can we do with it?

0	1	1	0	0	0	0	1

- The value of one or more bits can be changed, which makes it possible to scan the entire field of possible codes.
- Any other operation, like swapping two bits, ultimately comes down to the first operation.
- So the basic one-bit operations are:

Classical Operations

- Given a register of n bits (for example one byte), what operations can we do with it?

0	1	1	0	0	0	0	1

- The value of one or more bits can be changed, which makes it possible to scan the entire field of possible codes.
- Any other operation, like swapping two bits, ultimately comes down to the first operation.
- So the basic one-bit operations are:
  - Identity: trivial operation that leaves a bit with its initial value: $0 \longrightarrow 0; 1 \longrightarrow 1$

Classical Operations

- Given a register of n bits (for example one byte), what operations can we do with it?

0	1	1	0	0	0	0	1

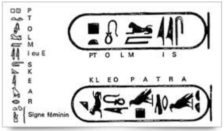
- The value of one or more bits can be changed, which makes it possible to scan the entire field of possible codes.
- Any other operation, like swapping two bits, ultimately comes down to the first operation.
- So the basic one-bit operations are:
  - Identity: trivial operation that leaves a bit with its initial value: $0 \longrightarrow 0; 1 \longrightarrow 1$
  - Inversion: which changes the value of the bit : $0 \longrightarrow 1; 1 \longrightarrow 0$

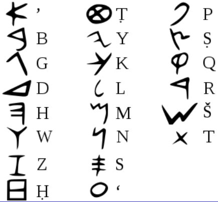
Information storage

- In information processing, you have to be able to write this information on a medium and be able to read it in order to use it.
- Since the beginning of mankind, this need has been implemented on physical media that have evolved throughout history.
- The prehistoric man painted on the walls of caves sketchy drawings that today we call cave paintings (hands, animals, human beings, ...). Even today, we can "read" this information and understand it.

Evolution of information media

- More evolved civilizations have used other techniques (such as engraving) and other supports (stone, marble, ...) for different representations.
- The complexity of life has prompted mankind to invent information encoding systems that allow it to express ideas in a more efficient manner.
- Thus, we created two fundamental tools: numbers and letters (or cryptograms)
- Here is for example the Phoenician alphabet, of which the Punic alphabet (that of Carthage) is a variant. Note the analogies with the current Latin alphabet.





History of number systems

- The creation of figures met a practical need from the appearance of the first centers of civilization. It was necessary to count all kinds of things and quantify them (people, head of cattle, food, ...) and do basic operations such as adding two numbers.

History of number systems

- The creation of figures met a practical need from the appearance of the first centers of civilization. It was necessary to count all kinds of things and quantify them (people, head of cattle, food, ...) and do basic operations such as adding two numbers.
- Several systems have been invented over time and in different regions, but to make a short cut, the system that has emerged as the standard is the decimal system, no doubt thanks to its anthropogenic origin (counting with the fingers).

History of number systems

- The creation of figures met a practical need from the appearance of the first centers of civilization. It was necessary to count all kinds of things and quantify them (people, head of cattle, food, ...) and do basic operations such as adding two numbers.
- Several systems have been invented over time and in different regions, but to make a short cut, the system that has emerged as the standard is the decimal system, no doubt thanks to its anthropogenic origin (counting with the fingers).
- This is the system that we use today all over the world.

History of number systems

- The creation of figures met a practical need from the appearance of the first centers of civilization. It was necessary to count all kinds of things and quantify them (people, head of cattle, food, ...) and do basic operations such as adding two numbers.
- Several systems have been invented over time and in different regions, but to make a short cut, the system that has emerged as the standard is the decimal system, no doubt thanks to its anthropogenic origin (counting with the fingers).
- This is the system that we use today all over the world.
- Extended to real numbers and augmented by scientific notation (mantissa and exponent), the decimal system satisfies almost all of our needs for expressing any quantity used in a conventional system of units, such as the international SI system. For example $h = 6.62607015 \times 10^{-34} J.s$ (exact value since May 20, 2019).

Number systems

Many systems have been used by peoples and at different times.

- The unary system (base 1): Counting with a succession of sticks, possibly grouped by 5, 10 or other.

Number systems

Many systems have been used by peoples and at different times.

- The unary system (base 1): Counting with a succession of sticks, possibly grouped by 5, 10 or other.
- The quinary system (base 5): of which traces remain until the 20th century in African languages, but also, partially, in Chuvash, Suzhou, Roman and Mayan notations. The name of the numbers 6, 7, 8 and 9 in many languages testify to this quinary system: they are said to be 5 + 1, 5 + 2, 5 + 3 and 5 + 4 in Wolof (language of the Niger-Congolese family) , in Khmer (Austro-Asiatic language), in Nahuatl (Uto-Aztec language), and, in many Austronesian languages such as Lote or Ngadha (in partial form). The quinary base appears as a sub-base of the decimal base and the vigesimal base.

Number systems

Many systems have been used by peoples and at different times.

- The unary system (base 1): Counting with a succession of sticks, possibly grouped by 5, 10 or other.
- The quinary system (base 5): of which traces remain until the 20th century in African languages, but also, partially, in Chuvash, Suzhou, Roman and Mayan notations. The name of the numbers 6, 7, 8 and 9 in many languages testify to this quinary system: they are said to be 5 + 1, 5 + 2, 5 + 3 and 5 + 4 in Wolof (language of the Niger-Congolese family) , in Khmer (Austro-Asiatic language), in Nahuatl (Uto-Aztec language), and, in many Austronesian languages such as Lote or Ngadha (in partial form). The quinary base appears as a sub-base of the decimal base and the vigesimal base.
- See Appendix 2 for other number systems (binary, ternary, senary, octal, nonary, decimal, duodecimal, hexadecimal, vigecimal, sexagesimal)

Decimal system

- The main strength of the decimal system is the simple and efficient way to give meaning to the position of a given digit in the decimal writing of a number (units, tens, hundreds, thousands, ...).

Decimal system

- The main strength of the decimal system is the simple and efficient way to give meaning to the position of a given digit in the decimal writing of a number (units, tens, hundreds, thousands, ...).
- This of course follows from the development on the decimal basis.

$$2022 = 2 \times 10^3 + 0 \times 10^2 + 2 \times 10^1 + 2 \times 10^0$$

Decimal system

- The main strength of the decimal system is the simple and efficient way to give meaning to the position of a given digit in the decimal writing of a number (units, tens, hundreds, thousands, ...).
- This of course follows from the development on the decimal basis.

$$2022 = 2 \times 10^3 + 0 \times 10^2 + 2 \times 10^1 + 2 \times 10^0$$

- The other strong point is the ease of performing the operation of adding two numbers, with the introduction of the carry-over rule.

Decimal system

- The main strength of the decimal system is the simple and efficient way to give meaning to the position of a given digit in the decimal writing of a number (units, tens, hundreds, thousands, ...).
- This of course follows from the development on the decimal basis.

$$2022 = 2 \times 10^3 + 0 \times 10^2 + 2 \times 10^1 + 2 \times 10^0$$

- The other strong point is the ease of performing the operation of adding two numbers, with the introduction of the carry-over rule.
- The operation of multiplying two numbers is more complicated, but its rules are simple and its manual realization, for reasonable numbers remains within reach.

Decimal System

- However, as soon as the numbers get large, the multiplication becomes very difficult to accomplish in a reasonable time, while the addition is always domesticable.

Decimal System

- However, as soon as the numbers get large, the multiplication becomes very difficult to accomplish in a reasonable time, while the addition is always domesticable.
- To convince yourself, try the following two operations :

$$
\begin{array}{rr}
 & 59863254681452687 \\
+ & \\
 & 15296325750265236 \\
 & \_\_\_\_\_\_\_\_\_
\end{array}
$$

Decimal System

- However, as soon as the numbers get large, the multiplication becomes very difficult to accomplish in a reasonable time, while the addition is always domesticable.
- To convince yourself, try the following two operations :

$$\begin{array}{r} 59863254681452687 \\ + \\ 15296325750265236 \\ \text{— — — — — — — — —} \end{array}$$

- and

$$\begin{array}{r} 59863254681452687 \\ \times \\ 15296325750265236 \\ \text{— — — — — — — — —} \end{array}$$

Decimal System

- However, as soon as the numbers get large, the multiplication becomes very difficult to accomplish in a reasonable time, while the addition is always domesticable.
- To convince yourself, try the following two operations :

$$\begin{array}{r} 59863254681452687 \\ + \\ 15296325750265236 \\ \text{— — — — — — — — —} \end{array}$$

- and

$$\begin{array}{r} 59863254681452687 \\ \times \\ 15296325750265236 \\ \text{— — — — — — — — —} \end{array}$$

- Start the stopwatch at the start and note the time it took you to complete each operation. Surprised ?

Back to the binary system

- The binary system, introduced by Leibnitz, takes the two basic ideas of the decimal system, namely the encoding in position in base 2, and the carry rule for the addition operation

Back to the binary system

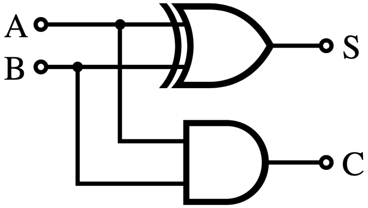
- The binary system, introduced by Leibnitz, takes the two basic ideas of the decimal system, namely the encoding in position in base 2, and the carry rule for the addition operation
- We recalled above how we code integers in binary. For addition, the rules are as follows for 2 bits:

bit 1	bit 2	sum	carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

- Thanks to the developments of electronics in the 20th century, it was possible to build physical supports (circuits) which make it possible to carry out the operations of storage (writing) and reading, as well as the various mathematical operations in binary code (we will come back to this later).

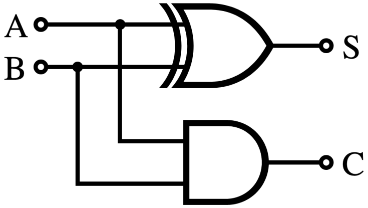
Half-adder Circuit

- The half-adder circuit is built from an XOR gate and a AND gate (we will come back to this)



Half-adder Circuit

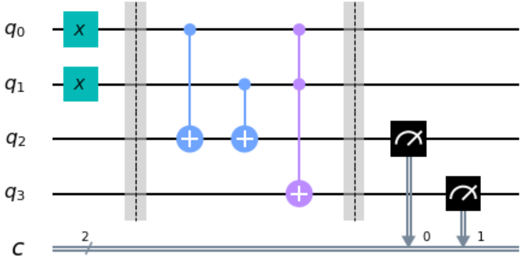
- The half-adder circuit is built from an XOR gate and a AND gate (we will come back to this)



- It takes two bits E_1 and E_2 as input and delivers 2 outputs, the sum S and the carry R,

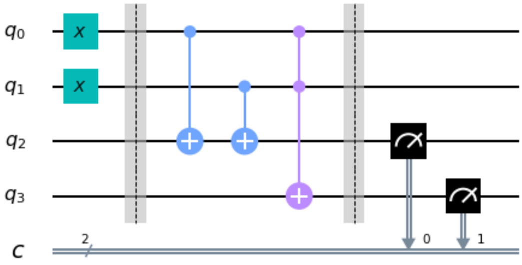
Hal-adder Quantum Circuit

- Here's a taste of what a half-adding quantum circuit looks like:



Hal-adder Quantum Circuit

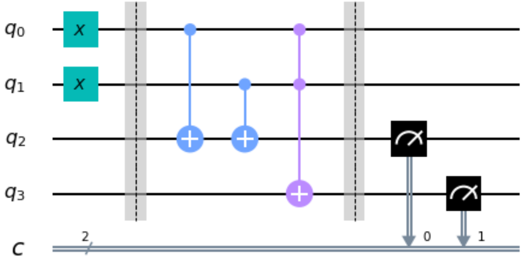
- Here's a taste of what a half-adding quantum circuit looks like:



- For now, it seems complicated, but at the end of this course, this kind of circuit will become clear.

Hal-adder Quantum Circuit

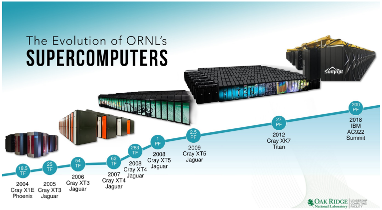
- Here's a taste of what a half-adding quantum circuit looks like:



- For now, it seems complicated, but at the end of this course, this kind of circuit will become clear.
- But already know at this level that this circuit is formed by quantum gates X, $CNOT$ and $Toffoli$!

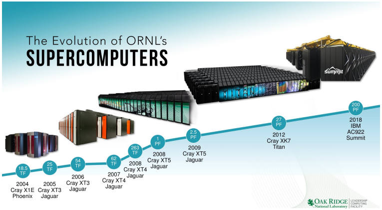
Classical Supercomputers

- This example gives a little idea about the practical implementation of a simple computation operation with logic circuits. The operation of digital calculators and computers is based on the same principle, obviously with a high degree of complexity. but the principle is the same.



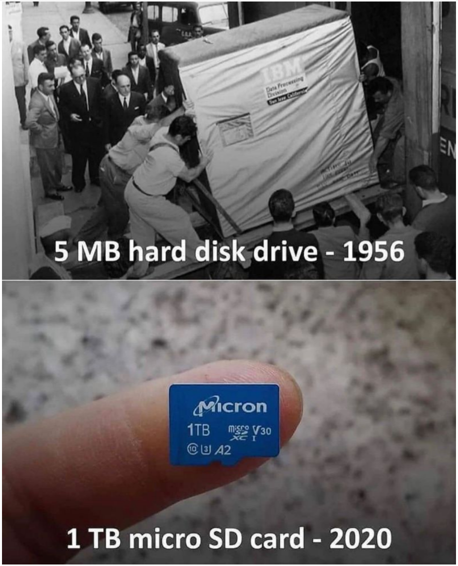
Classical Supercomputers

- This example gives a little idea about the practical implementation of a simple computation operation with logic circuits. The operation of digital calculators and computers is based on the same principle, obviously with a high degree of complexity. but the principle is the same.



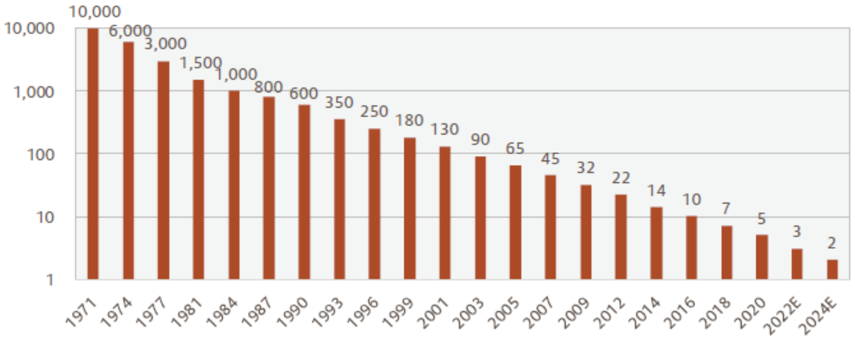
- Today there are very powerful computers like *IBM Summit*, capable of performing up to 2×10^{17} operations / second.

Evolution of storage capacities



Limits of miniaturization: Moore's Law

Figure 2: The incredible shrinking universe (device size in nm, log scale)



Source: PC Magazine, Epoch Investment Partners
Note: For reference, most atoms are 0.1 to 0.5 nm in diameter

Epoch perspectives, 11 February 2021

Limits of computing power

- Are today's computers sufficient to perform the calculations we need ?

Limits of computing power

- Are today's computers sufficient to perform the calculations we need ?
- Yes, to a certain extent.

Limits of computing power

- Are today's computers sufficient to perform the calculations we need ?
- Yes, to a certain extent.
- However, for some problems we reach a limit. For example, the factorization of large numbers, used to encrypt messages (RSA protocol) and ensure the security of communications and transactions related to e-commerce, electronic signatures, etc.

FACULTE DES SCIENCES DE TUNIS

Quantum computers

- Several projects have been launched in recent years with the aim of building the first efficient and reliable quantum computer (IBM, Google, Dwave, ...).

Quantum computers

- Several projects have been launched in recent years with the aim of building the first efficient and reliable quantum computer (IBM, Google, Dwave, ...).
- For now, this objective is still a long way off, given the constraints and obstacles, both fundamental (decoherence) and technical (cryogenics, cabling, etc.).

Quantum computers

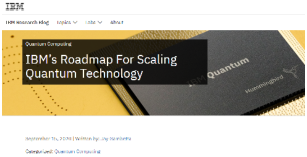
- Several projects have been launched in recent years with the aim of building the first efficient and reliable quantum computer (IBM, Google, Dwave, ...).
- For now, this objective is still a long way off, given the constraints and obstacles, both fundamental (decoherence) and technical (cryogenics, cabling, etc.).
- Some prototypes with a few dozen qubits already exist and work.

Quantum computers

- Several projects have been launched in recent years with the aim of building the first efficient and reliable quantum computer (IBM, Google, Dwave, ...).
- For now, this objective is still a long way off, given the constraints and obstacles, both fundamental (decoherence) and technical (cryogenics, cabling, etc.).
- Some prototypes with a few dozen qubits already exist and work.
- However, to show how fast this field is changing, here are two recent announcements (September 2020):

Dwave has announced the launch of a 5000+ qubit quantum computer:

IBM has published a roadmap announcing a 1000+ qubit computer for 2023.



Quantum Supremacy

nature

Explore our content ⌄ Journal information ⌄

nature › articles › article

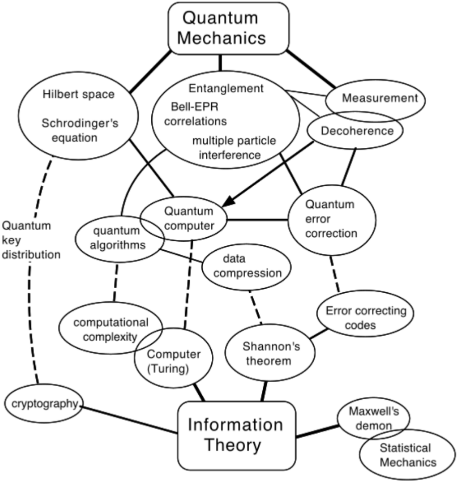
Article | Published: 23 October 2019

Quantum supremacy using a programmable superconducting processor

Frank Arute, Kunal Arya, [...] John M. Martinis

Nature **574**, 505–510(2019) | Cite this article

Quantum Mechanics and Information Theory



[3]

[3] Andrew Steane 1998 Rep. Prog. Phys. 61 117

Cryptography

- Until the 1970s, known encryption systems were based on the principle of symmetric cryptography, where the same (secret) key is used to encrypt and decrypt a message.

Cryptography

- Until the 1970s, known encryption systems were based on the principle of symmetric cryptography, where the same (secret) key is used to encrypt and decrypt a message.
- In 1978, R. Rivest, A. Shamir and L. Adleman described the first public system of asymmetric cryptography (named after their initials RSA), based on the properties of prime numbers and factorization. In such a system, two keys are used: one is used to encrypt, the other to decrypt.

Cryptography

- Until the 1970s, known encryption systems were based on the principle of symmetric cryptography, where the same (secret) key is used to encrypt and decrypt a message.
- In 1978, R. Rivest, A. Shamir and L. Adleman described the first public system of asymmetric cryptography (named after their initials RSA), based on the properties of prime numbers and factorization. In such a system, two keys are used: one is used to encrypt, the other to decrypt.
- The key used to encrypt is accompanied by a large integer, the product of two large primes kept secret (of the order of 200 digits, see RSA numbers). To calculate the decryption key, the only known method requires knowing the two prime factors.

RSA Protocole

- The security of the RSA system is based on the fact that it is easy to find two large prime numbers (using primality tests) and multiply them between them, but that it would be difficult for an attacker to find these two numbers. . This system also allows the creation of digital signatures, and has revolutionized the world of cryptography.

RSA Protocole

- The security of the RSA system is based on the fact that it is easy to find two large prime numbers (using primality tests) and multiply them between them, but that it would be difficult for an attacker to find these two numbers. . This system also allows the creation of digital signatures, and has revolutionized the world of cryptography.
- It is a protocol widely used today for the secure communication of information (telecommunications, electronic commerce, defense, etc.)

RSA Protocole

- The security of the RSA system is based on the fact that it is easy to find two large prime numbers (using primality tests) and multiply them between them, but that it would be difficult for an attacker to find these two numbers. . This system also allows the creation of digital signatures, and has revolutionized the world of cryptography.
- It is a protocol widely used today for the secure communication of information (telecommunications, electronic commerce, defense, etc.)
- A sender A (Alice) wants to send a secret message to a receiver B (Bob), without a possible spy E (Eve) intercepting it.

RSA Protocole

- The security of the RSA system is based on the fact that it is easy to find two large prime numbers (using primality tests) and multiply them between them, but that it would be difficult for an attacker to find these two numbers. . This system also allows the creation of digital signatures, and has revolutionized the world of cryptography.
- It is a protocol widely used today for the secure communication of information (telecommunications, electronic commerce, defense, etc.)
- A sender A (Alice) wants to send a secret message to a receiver B (Bob), without a possible spy E (Eve) intercepting it.
- To do this, Alice and Bob need to agree on a code that allows them and no one else to piece together the messages.

Prime numbers

- A prime number is a natural number that admits exactly two distinct positive and integer divisors. These two divisors are 1 and the number considered, since any number has as divisors 1 and itself (as shown by the equality $n = 1 \times n$)

Prime numbers

- A prime number is a natural number that admits exactly two distinct positive and integer divisors. These two divisors are 1 and the number considered, since any number has as divisors 1 and itself (as shown by the equality $n = 1 \times n$)
- the largest prime number known to date was discovered in 2018: It is the Mersenne prime number : $2^{82589933} - 1$, which has more than 24 million digits in decimal number system.

Prime numbers

- A prime number is a natural number that admits exactly two distinct positive and integer divisors. These two divisors are 1 and the number considered, since any number has as divisors 1 and itself (as shown by the equality $n = 1 \times n$)
- the largest prime number known to date was discovered in 2018: It is the Mersenne prime number : $2^{82589933} - 1$, which has more than 24 million digits in decimal number system.
- A very large number of primes are listed, including gigantic numbers of over a million digits.

Prime numbers

- A prime number is a natural number that admits exactly two distinct positive and integer divisors. These two divisors are 1 and the number considered, since any number has as divisors 1 and itself (as shown by the equality $n = 1 \times n$)
- the largest prime number known to date was discovered in 2018: It is the Mersenne prime number : $2^{82589933} - 1$, which has more than 24 million digits in decimal number system.
- A very large number of primes are listed, including gigantic numbers of over a million digits.
- It is easy to choose two as the numbers p and q and calculate N for the protocol, while the reverse operation is very difficult.

Prime numbers

- A prime number is a natural number that admits exactly two distinct positive and integer divisors. These two divisors are 1 and the number considered, since any number has as divisors 1 and itself (as shown by the equality $n = 1 \times n$)
- the largest prime number known to date was discovered in 2018: It is the Mersenne prime number : $2^{82589933} - 1$, which has more than 24 million digits in decimal number system.
- A very large number of primes are listed, including gigantic numbers of over a million digits.
- It is easy to choose two as the numbers p and q and calculate N for the protocol, while the reverse operation is very difficult.
- This allows the use of very large numbers like $RSA2048$ for public key cryptography protocols.

RSA-2048

25195908475657893494027183240048398571429282126204
03202777713783604366202070759555626401852588078440
69182906412495150821892985591491761845028084891200
72844992687392807287776735971418347270261896375014
97182469116507761337985909570009733045974880842840
17974291006424586918171951187461215151726546322822
16869987549182422433637259085141865462043576798423
38718477444792073993423658482382428119816381501067
48104516603773060562016196762561338441436038339044
14952634432190114657544454178424020924616515723350
77870774981712577246796292638635637328991215483143
81678998850404453640235273819513786365643912120103
97122822120720357

See Appendix 4 for more details about RSA numbers.

Complexity classes

- To better understand the difficulty of the factoring problem, here are some examples of mathematical problems as well as the scale laws of the number of operations n with the number of bits (or digits) as well as the complexity classes:

Problem	Operations	Class
Addition of 2 numbers of n bits	n	P
Multiplication of 2 numbers of n bits	n^2	P
FFT de n bits	$n\log(n)$	P
Factoring a number of n bits	$2^{n/2}$	NP
Travelling salesman problem (n towns)	$e^{n\log(n)}$	NPC

Complexity classes

- To better understand the difficulty of the factoring problem, here are some examples of mathematical problems as well as the scale laws of the number of operations n with the number of bits (or digits) as well as the complexity classes:

Problem	Operations	Class
Addition of 2 numbers of n bits	n	P
Multiplication of 2 numbers of n bits	n^2	P
FFT de n bits	$n\log(n)$	P
Factoring a number of n bits	$2^{n/2}$	NP
Travelling salesman problem (n towns)	$e^{n\log(n)}$	NPC

- Current computer architectures are unable to deal with complex problems due to a lack of efficient algorithms.

The second quantum revolution

- Fortunately, we are living in great times!

The second quantum revolution

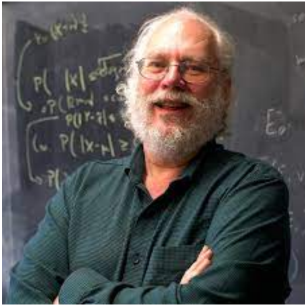
- Fortunately, we are living in great times!
- A century after the birth of quantum mechanics, we are in the midst of what is called the second quantum revolution, with as corollary a rapid development of quantum technologies, including quantum computing.

The second quantum revolution

- Fortunately, we are living in great times!
- A century after the birth of quantum mechanics, we are in the midst of what is called the second quantum revolution, with as corollary a rapid development of quantum technologies, including quantum computing.
- Quantum information gives hope, and it is reasonable to think that we will be able to solve the problem of factorization of large numbers and many others within a reasonable period of time.

The second quantum revolution

- Fortunately, we are living in great times!
- A century after the birth of quantum mechanics, we are in the midst of what is called the second quantum revolution, with as corollary a rapid development of quantum technologies, including quantum computing.
- Quantum information gives hope, and it is reasonable to think that we will be able to solve the problem of factorization of large numbers and many others within a reasonable period of time.
- A major breakthrough was made in 1994 by Peter Shor, who developed a quantum factoring algorithm.

- 

Peter Shor
(ICTP Dirac medal 2017)

Shor's algorithm

- Without going into details at this level (See lecture 3), and assuming the existence of a perfect quantum computer, the algorithm developed by Peter Shor promises to factor a number of 500 digits, which should take more than the age of the universe on a current processor, in just 2 seconds !!!

Shor's algorithm

- Without going into details at this level (See lecture 3), and assuming the existence of a perfect quantum computer, the algorithm developed by Peter Shor promises to factor a number of 500 digits, which should take more than the age of the universe on a current processor, in just 2 seconds !!!
- Physicists made a first rough estimate for RSA-2048 and found that with a quantum computer formed of 10,000 logical qubits and 10 million physical (superconducting) qubits, spaced 1 cm apart for the wiring, which would cost "only" 100 billion USD, and using a modest electric power of 10 MW, would get the job done in 16 hours !!! (J. Preskill 2012)

Conclusions of Lecture 1

- Brief review of Shannon Information Theory

Conclusions of Lecture 1

- Brief review of Shannon Information Theory
- Reminder of binary logic and operations

Conclusions of Lecture 1

- Brief review of Shannon Information Theory
- Reminder of binary logic and operations
- Classical cryptography (RSA protocol)

Conclusions of Lecture 1

- Brief review of Shannon Information Theory
- Reminder of binary logic and operations
- Classical cryptography (RSA protocol)
- Second Quantum revoltution and Birth of Quantum Information

Conclusions of Lecture 1

- Brief review of Shannon Information Theory
- Reminder of binary logic and operations
- Classical cryptography (RSA protocol)
- Second Quantum revoltution and Birth of Quantum Information
- You can go through the appendices for a complement of lecture 1

Conclusions of Lecture 1

- Brief review of Shannon Information Theory
- Reminder of binary logic and operations
- Classical cryptography (RSA protocol)
- Second Quantum revoltution and Birth of Quantum Information
- You can go through the appendices for a complement of lecture 1
- Next lecture : Quantum mechanics for Quantum Information

Appendix 1 : ASCII Code (7-bits version)

- *Command characters (0–31 and 127)*: Control characters are non-printable characters. They are used to send commands to the computer or printer. Many of these characters are no longer used today.

Appendix 1 : ASCII Code (7-bits version)

- *Command characters (0–31 and 127)*: Control characters are non-printable characters. They are used to send commands to the computer or printer. Many of these characters are no longer used today.
- *Special characters (32–47 / 58–64 / 91–96 / 123–126)*: Include all printable characters that are not letters or numbers, such as punctuation marks or technical and mathematical characters. ASCII also includes the blank character, which is considered a non-visible but printable character.

Appendix 1 : ASCII Code (7-bits version)

- *Command characters (0–31 and 127)*: Control characters are non-printable characters. They are used to send commands to the computer or printer. Many of these characters are no longer used today.
- *Special characters (32–47 / 58–64 / 91–96 / 123–126)*: Include all printable characters that are not letters or numbers, such as punctuation marks or technical and mathematical characters. ASCII also includes the blank character, which is considered a non-visible but printable character.
- *Numbers (30–39)*: the ten digits from 0 to 9.

Appendix 1 : ASCII Code (7-bits version)

- *Command characters (0–31 and 127)*: Control characters are non-printable characters. They are used to send commands to the computer or printer. Many of these characters are no longer used today.
- *Special characters (32–47 / 58–64 / 91–96 / 123–126)*: Include all printable characters that are not letters or numbers, such as punctuation marks or technical and mathematical characters. ASCII also includes the blank character, which is considered a non-visible but printable character.
- *Numbers (30–39)*: the ten digits from 0 to 9.
- *Letters (65–90 / 97–122)*: divided into two blocks, upper and lower case.

Appendix 2 : Number systems

Many systems have been used by peoples and at different times.

- The unary system (base 1): Counting with a succession of sticks, possibly grouped by 5, 10 or other.

Appendix 2 : Number systems

Many systems have been used by peoples and at different times.

- The unary system (base 1): Counting with a succession of sticks, possibly grouped by 5, 10 or other.
- The binary system (base 2): used in languages of South America and Oceania is similar to the number system used in computer science (1 current is on, 0 current is not on)

Appendix 2 : Number systems

Many systems have been used by peoples and at different times.

- The unary system (base 1): Counting with a succession of sticks, possibly grouped by 5, 10 or other.
- The binary system (base 2): used in languages of South America and Oceania is similar to the number system used in computer science (1 current is on, 0 current is not on)
- A ternary system (base 3) :

Appendix 2 : Number systems

Many systems have been used by peoples and at different times.

- The unary system (base 1): Counting with a succession of sticks, possibly grouped by 5, 10 or other.
- The binary system (base 2): used in languages of South America and Oceania is similar to the number system used in computer science (1 current is on, 0 current is not on)
- A ternary system (base 3) :
- The quinary system (base 5): of which traces remain until the 20th century in African languages, but also, partially, in Chuvash, Suzhou, Roman and Mayan notations. The name of the numbers 6, 7, 8 and 9 in many languages testify to this quinary system: they are said to be 5 + 1, 5 + 2, 5 + 3 and 5 + 4 in Wolof (language of the Niger-Congolese family) , in Khmer (Austro-Asiatic language), in Nahuatl (Uto-Aztec language), and, in many Austronesian languages such as Lote or Ngadha (in partial form). The quinary base appears as a sub-base of the decimal base and the vigesimal base.

Appendix 2 : Number Systems

- The senary system (base 6): is used in the Ndom and Kómnzo languages of Papua New Guinea, as well as in dice. It uses six digits from 0 to 5, finger counts by "multiples of three" bases, the most convenient.

Appendix 2 : Number Systems

- The senary system (base 6): is used in the Ndom and Kómnzo languages of Papua New Guinea, as well as in dice. It uses six digits from 0 to 5, finger counts by "multiples of three" bases, the most convenient.
- octal system (base 8): is used in the northern pame language, in Mexico, and in the yuki language, in California, as well as in computer science.

Appendix 2 : Number Systems

- The senary system (base 6): is used in the Ndom and Kómnzo languages of Papua New Guinea, as well as in dice. It uses six digits from 0 to 5, finger counts by "multiples of three" bases, the most convenient.
- octal system (base 8): is used in the northern pame language, in Mexico, and in the yuki language, in California, as well as in computer science.
- The nonary system (base 9): is opposite to hexadecimal, "power of three" bases. It includes two ternary digits in one digit.

Appendix 2 : Number Systems

- The senary system (base 6): is used in the Ndom and Kómnzo languages of Papua New Guinea, as well as in dice. It uses six digits from 0 to 5, finger counts by "multiples of three" bases, the most convenient.
- octal system (base 8): is used in the northern pame language, in Mexico, and in the yuki language, in California, as well as in computer science.
- The nonary system (base 9): is opposite to hexadecimal, "power of three" bases. It includes two ternary digits in one digit.
- The decimal system (base 10): has been used by many civilizations, such as the Chinese from early times, and, probably, the Proto-Indo-Europeans. Today, it is by far the most widespread.

Appendix 2 : Number Systems

- The senary system (base 6): is used in the Ndom and Kómnzo languages of Papua New Guinea, as well as in dice. It uses six digits from 0 to 5, finger counts by "multiples of three" bases, the most convenient.
- octal system (base 8): is used in the northern pame language, in Mexico, and in the yuki language, in California, as well as in computer science.
- The nonary system (base 9): is opposite to hexadecimal, "power of three" bases. It includes two ternary digits in one digit.
- The decimal system (base 10): has been used by many civilizations, such as the Chinese from early times, and, probably, the Proto-Indo-Europeans. Today, it is by far the most widespread.
- The duodecimal system (base 12): is used in Nepal by the Chepang people. It is found, because of its advantages in terms of divisibility (by 2, 3, 4, 6), for a certain number of currencies and current account units in Europe in the Middle Ages, partially in Anglo-Saxon countries. in the imperial system of units, and in commerce. It is also used to count months, hours, flowers, oysters and eggs.

Appendix 2 : Number Systems

- The hexadecimal system (base 16):, very commonly used in electronics as well as in computer science.

Appendix 2 : Number Systems

- The hexadecimal system (base 16):, very commonly used in electronics as well as in computer science.
- The vigesimal system (or vicesimal, base 20): exists in Bhutan in the Dzongkha language, and was in use among the Aztecs and, although irregular, for the Mayan numeration. It is found again, because of its advantages in terms of divisibility (by 2, 4, 5, 10). Some believe that it was also used by the Gauls or by the Basques in the early days, but it is not really known whether their numeration had a decimal or vigesimal character.

Appendix 2 : Number Systems

- The hexadecimal system (base 16):, very commonly used in electronics as well as in computer science.
- The vigesimal system (or vicesimal, base 20): exists in Bhutan in the Dzongkha language, and was in use among the Aztecs and, although irregular, for the Mayan numeration. It is found again, because of its advantages in terms of divisibility (by 2, 4, 5, 10). Some believe that it was also used by the Gauls or by the Basques in the early days, but it is not really known whether their numeration had a decimal or vigesimal character.
- The sexagesimal system (base 60): was used for Babylonian numeration, as well as by Indians and Arabs in trigonometry. It is currently used in measuring time and angles.

Appendix 3 : RSA Protocole

- Bob chooses two prime numbers p and q, with $N = pq$, and a number c having no common divisor with the product $(p-1)(q-1)$.

Appendix 3 : RSA Protocole

- Bob chooses two prime numbers p and q, with $N = pq$, and a number c having no common divisor with the product $(p-1)(q-1)$.
- He calculates d which is the inverse of c for the multiplication modulo $(p-1)(q-1)$

$$cd = 1 \mod (p-1)(q-1)$$

Appendix 3 : RSA Protocole

- Bob chooses two prime numbers p and q, with $N = pq$, and a number c having no common divisor with the product $(p-1)(q-1)$.
- He calculates d which is the inverse of c for the multiplication modulo $(p-1)(q-1)$

$$cd = 1 \mod (p-1)(q-1)$$

- He sends Alice the numbers N and c (but not p and q separately!) To Alice by an insecure channel (email or sms for example).

Appendix 3 : RSA Protocole

- Bob chooses two prime numbers p and q, with $N = pq$, and a number c having no common divisor with the product $(p-1)(q-1)$.
- He calculates d which is the inverse of c for the multiplication modulo $(p-1)(q-1)$
$$cd = 1 \mod (p-1)(q-1)$$
- He sends Alice the numbers N and c (but not p and q separately!) To Alice by an insecure channel (email or sms for example).
- Alice wants to send Bob an encoded message, represented by a number a $a < N$ (if the message is too long, Alice can segment it into several sub messages).

Appendix 3 : RSA Protocole

- Bob chooses two prime numbers p and q, with $N = pq$, and a number c having no common divisor with the product $(p-1)(q-1)$.
- He calculates d which is the inverse of c for the multiplication modulo $(p-1)(q-1)$

$$cd = 1 \mod (p-1)(q-1)$$

- He sends Alice the numbers N and c (but not p and q separately!) To Alice by an insecure channel (email or sms for example).
- Alice wants to send Bob an encoded message, represented by a number a $a < N$ (if the message is too long, Alice can segment it into several sub messages).
- She then calculates $b = a^c \mod N$, and sends b to Bob.

Appendix 3 : RSA Protocole

- Bob chooses two prime numbers p and q, with $N = pq$, and a number c having no common divisor with the product $(p-1)(q-1)$.
- He calculates d which is the inverse of c for the multiplication modulo $(p-1)(q-1)$
$$cd = 1 \mod (p-1)(q-1)$$
- He sends Alice the numbers N and c (but not p and q separately!) To Alice by an insecure channel (email or sms for example).
- Alice wants to send Bob an encoded message, represented by a number a $a < N$ (if the message is too long, Alice can segment it into several sub messages).
- She then calculates $b = a^c \mod N$, and sends b to Bob.
- When Bob receives the message he calculates $b^d \mod N = a$, which allows him to find Alice's message.

Appendix 3 : RSA Protocole

- The fact that the result is precisely a, that is to say Alice's original message, is a result of number theory (Fermat's Little Theorem).

Appendix 3 : RSA Protocole

- The fact that the result is precisely a, that is to say Alice's original message, is a result of number theory (Fermat's Little Theorem).
- So, in the end, Bob decoded the message and reconstructed the original message a sent by Alice.

Appendix 3 : RSA Protocole

- The fact that the result is precisely a, that is to say Alice's original message, is a result of number theory (Fermat's Little Theorem).
- So, in the end, Bob decoded the message and reconstructed the original message a sent by Alice.
- The key to the success of the RSA protocol is the great difficulty of factoring N and finding the two prime numbers p and q, as soon as N is large enough.

Appendix 3 : Fermat's Little Theorem

- Statement 1: "Let p be a prime number, and a be a prime integer with p. Then a^{p-1} has for remainder 1 in the division by p. "

Appendix 3 : Fermat's Little Theorem

- Statement 1: "Let p be a prime number, and a be a prime integer with p. Then a^{p-1} has for remainder 1 in the division by p. "
- Statement 2: "Let p be a prime number, and $1 \leq a \leq p-1$. Then p divide $a^{p-1} - 1$."

Appendix 3 : Fermat's Little Theorem

- Statement 1: "Let p be a prime number, and a be a prime integer with p. Then a^{p-1} has for remainder 1 in the division by p. "
- Statement 2: "Let p be a prime number, and $1 \leq a \leq p-1$. Then p divide $a^{p-1} - 1$."
- This theorem conjectured by Fermat in 1640, was later demonstrated by Leibniz and Euler.

Appendix 3 : Fermat's Little Theorem

- Statement 1: "Let p be a prime number, and a be a prime integer with p. Then a^{p-1} has for remainder 1 in the division by p. "
- Statement 2: "Let p be a prime number, and $1 \leq a \leq p-1$. Then p divide $a^{p-1} - 1$."
- This theorem conjectured by Fermat in 1640, was later demonstrated by Leibniz and Euler.
- The proof uses Gauss's theorem in arithmetic.

Appendix 3 : Fermat's Little Theorem

- Statement 1: "Let p be a prime number, and a be a prime integer with p. Then a^{p-1} has for remainder 1 in the division by p. "
- Statement 2: "Let p be a prime number, and $1 \leq a \leq p-1$. Then p divide $a^{p-1}-1$."
- This theorem conjectured by Fermat in 1640, was later demonstrated by Leibniz and Euler.
- The proof uses Gauss's theorem in arithmetic.
- Fermat's great theorem, renamed Fermat-Wiles Theorem: stated around 1640 by Fermat, demonstrated in 1995 by Wiles (Prix Abel 2016).

Appendix 3 : Fermat's Little Theorem

- Statement 1: "Let p be a prime number, and a be a prime integer with p. Then a^{p-1} has for remainder 1 in the division by p. "
- Statement 2: "Let p be a prime number, and $1 \leq a \leq p-1$. Then p divide $a^{p-1} - 1$."
- This theorem conjectured by Fermat in 1640, was later demonstrated by Leibniz and Euler.
- The proof uses Gauss's theorem in arithmetic.
- Fermat's great theorem, renamed Fermat-Wiles Theorem: stated around 1640 by Fermat, demonstrated in 1995 by Wiles (Prix Abel 2016).
- Statement: "The equation $x^n + y^n = z^n$ where $x, y, z \in \mathbb{N}$ and $n \geq 3$, does not admit solutions other than $(0, 0, 0)$.

Appendix 3 : RSA Protocole

- Let's go back to the RSA protocol by illustrating it with an example where Bob chooses:

$$N = 21, p = 3, q = 7, (p-1)(q-1) = 12$$

Appendix 3 : RSA Protocole

- Let's go back to the RSA protocol by illustrating it with an example where Bob chooses:

$$N = 21, p = 3, q = 7, (p-1)(q-1) = 12$$

- Bob chooses $c = 5$, which has no common divisor with 12, and sends the two numbers $N = 21$ and $c = 5$ (we will verify that $d = 5$) to Alice by public route.

Appendix 3 : RSA Protocole

- Let's go back to the RSA protocol by illustrating it with an example where Bob chooses:

$$N = 21, p = 3, q = 7, (p-1)(q-1) = 12$$

- Bob chooses $c = 5$, which has no common divisor with 12, and sends the two numbers $N = 21$ and $c = 5$ (we will verify that $d = 5$) to Alice by public route.
- Having received the key, Alice decides to send the message $a = 4$ to Bob. She calculates

$$a^c = 4^5 = 1024 = 21 \times 48 + 16; 4^5 = 16 \mod 21$$

Appendix 3 : RSA Protocole

- Let's go back to the RSA protocol by illustrating it with an example where Bob chooses:

$$N = 21, p = 3, q = 7, (p-1)(q-1) = 12$$

- Bob chooses $c = 5$, which has no common divisor with 12, and sends the two numbers $N = 21$ and $c = 5$ (we will verify that $d = 5$) to Alice by public route.
- Having received the key, Alice decides to send the message $a = 4$ to Bob. She calculates

$$a^c = 4^5 = 1024 = 21 \times 48 + 16; 4^5 = 16 \mod 21$$

- Alice sends Bob the number $b = 16$, which he must decode to find the original message $a = 4$

Appendix 3 : RSA Protocole

- Let's go back to the RSA protocol by illustrating it with an example where Bob chooses:

$$N = 21, p = 3, q = 7, (p-1)(q-1) = 12$$

- Bob chooses $c = 5$, which has no common divisor with 12, and sends the two numbers $N = 21$ and $c = 5$ (we will verify that $d = 5$) to Alice by public route.
- Having received the key, Alice decides to send the message $a = 4$ to Bob. She calculates

$$a^c = 4^5 = 1024 = 21 \times 48 + 16; 4^5 = 16 \mod 21$$

- Alice sends Bob the number $b = 16$, which he must decode to find the original message $a = 4$
- Pour cela, Bob calcule

$$b^5 = 16^5 = 49932 \times 21 + 4 \text{ then} 16^5 = 4 \mod 21$$

Appendix 3 : RSA Protocole

- Let's go back to the RSA protocol by illustrating it with an example where Bob chooses:

$$N = 21, p = 3, q = 7, (p-1)(q-1) = 12$$

- Bob chooses $c = 5$, which has no common divisor with 12, and sends the two numbers $N = 21$ and $c = 5$ (we will verify that $d = 5$) to Alice by public route.
- Having received the key, Alice decides to send the message $a = 4$ to Bob. She calculates

$$a^c = 4^5 = 1024 = 21 \times 48 + 16; 4^5 = 16 \mod 21$$

- Alice sends Bob the number $b = 16$, which he must decode to find the original message $a = 4$
- Pour cela, Bob calcule

$$b^5 = 16^5 = 49932 \times 21 + 4 \;\text{ then} 16^5 = 4 \mod 21$$

- Finally Bob extracts the message $a = 4$.

Appendix 3 : RSA protocol (Factoring)

- Bob sent Alice the key (N, c) by public way (on the Internet for example). So this key is accessible to everyone, including the possible spy on Eve.

Appendix 3 : RSA protocol (Factoring)

- Bob sent Alice the key (N, c) by public way (on the Internet for example). So this key is accessible to everyone, including the possible spy on Eve.
- So why can't Eve reconstruct the message $a = 4$ from the coded message $b = 16$ sent by Alice?

Appendix 3 : RSA protocol (Factoring)

- Bob sent Alice the key (N, c) by public way (on the Internet for example). So this key is accessible to everyone, including the possible spy on Eve.
- So why can't Eve reconstruct the message $a = 4$ from the coded message $b = 16$ sent by Alice?
- To do this, Eve needs d, and therefore the prime numbers p and q whose product gives N.

Appendix 3 : RSA protocol (Factoring)

- Bob sent Alice the key (N, c) by public way (on the Internet for example). So this key is accessible to everyone, including the possible spy on Eve.
- So why can't Eve reconstruct the message $a = 4$ from the coded message $b = 16$ sent by Alice?
- To do this, Eve needs d, and therefore the prime numbers p and q whose product gives N.
- the question is therefore: Is it that difficult to find p and q knowing N?

Appendix 3 : RSA protocol (Factoring)

- Bob sent Alice the key (N, c) by public way (on the Internet for example). So this key is accessible to everyone, including the possible spy on Eve.
- So why can't Eve reconstruct the message $a = 4$ from the coded message $b = 16$ sent by Alice?
- To do this, Eve needs d, and therefore the prime numbers p and q whose product gives N.
- the question is therefore: Is it that difficult to find p and q knowing N?
- the answer is yes, it is a really difficult problem, as soon as N is big enough.

Appendix 3 : RSA protocol (Factoring)

- Bob sent Alice the key (N, c) by public way (on the Internet for example). So this key is accessible to everyone, including the possible spy on Eve.
- So why can't Eve reconstruct the message $a = 4$ from the coded message $b = 16$ sent by Alice?
- To do this, Eve needs d, and therefore the prime numbers p and q whose product gives N.
- the question is therefore: Is it that difficult to find p and q knowing N?
- the answer is yes, it is a really difficult problem, as soon as N is big enough.
- Obviously, in the example chosen $N = 21$, the factorization $21 = 3 \times 7$ is trivial, but the situation gets complicated very quickly when N becomes large.

Appendix 4 : RSA numbers

- RSA-200 : made up of 200 digits in decimal

  27997833911221327870829467638722601621070446786955
  42853756000992932612840010760934567105295536085606
  18223519109513657886371059544820065767750985805576
  13579098734950144178863178946295187237869221823983

FACULTE DES SCIENCES DE TUNIS

Appendix 4 : RSA numbers

- RSA-200 : made up of 200 digits in decimal

  27997833911221327870829467638722601621070446786955
  42853756000992932612840010760934567105295536085606
  18223519109513657886371059544820065767750985805576
  13579098734950144178863178946295187237869221823983

- This is one of the largest numbers that it has been possible to factorize (2005 : F. Bahr, M. Boehm, J. Franke, and T. Kleinjung)

Appendix 4 : RSA numbers

- RSA-200 : made up of 200 digits in decimal

27997833911221327870829467638722601621070446786955
42853756000992932612840010760934567105295536085606
18223519109513657886371059544820065767750985805576
13579098734950144178863178946295187237869221823983

- This is one of the largest numbers that it has been possible to factorize (2005 : F. Bahr, M. Boehm, J. Franke, and T. Kleinjung)
- The calculation carried out on a network of computers required a CPU calculation time equivalent to 75 years on an Opetron processor running at 2.2GHz.

Appendix 4 : RSA numbers

- The current record, dating from 2009, for the largest factored number (RSA-768): formed by 232 digits in decimal

$$
\begin{aligned}
&12301866845301177551304949583849627207728535695953\\
&34792197322452151726400507263657518745202199786469\\
&38995647494277406384592519255732630345373154826850\\
&79170261221429134616704292143116022212404792747377\\
&\hfill 94080665351419597459856902143413
\end{aligned}
$$

- The calculation carried out on a network of computers required approximately two years of calculation, that is to say a CPU calculation time equivalent to 2000 years on an Opetron processor running at 2.2GHz.

Appendix 4 : RSA-2048

25195908475657893494027183240048398571429282126204
03202777713783604366202070759555626401852588078440
69182906412495150821892985591491761845028084891200
72844992687392807287776735971418347270261896375014
97182469116507761337985909570009733045974880842840
17974291006424586918171951187461215151726546322822
16869987549182422433637259085141865462043576798423
38718477444792073993423658482382428119816381501067
48104516603773060562016196762561338441436038339044
14952634432190114657544454178424020924616515723350
77870774981712577246796292638635637328991215483143
81678998850404453640235273819513786365643912120103
97122822120720357

Appendix 4 : RSA challenge

- The RSA-2048 number formed of 617 digits, or 2048 binary digits.

Appendix 4 : RSA challenge

- The RSA-2048 number formed of 617 digits, or 2048 binary digits.
- There was a challenge launched with a prize of USD 200,000 (canceled in 2007).

Appendix 4 : RSA challenge

- The RSA-2048 number formed of 617 digits, or 2048 binary digits.
- There was a challenge launched with a prize of USD 200,000 (canceled in 2007).
- This is an extraordinarily difficult problem and it is highly unlikely that this challenge will be met quickly, unless there is a major technological breakthrough.

Appendix 4 : RSA challenge

- The RSA-2048 number formed of 617 digits, or 2048 binary digits.
- There was a challenge launched with a prize of USD 200,000 (canceled in 2007).
- This is an extraordinarily difficult problem and it is highly unlikely that this challenge will be met quickly, unless there is a major technological breakthrough.
- With current technology, it is estimated that the time required, on a single processor, to factor a number of "only" 500 digits would be greater than the age of the universe !!!

Appendix 4 : RSA challenge

- The RSA-2048 number formed of 617 digits, or 2048 binary digits.
- There was a challenge launched with a prize of USD 200,000 (canceled in 2007).
- This is an extraordinarily difficult problem and it is highly unlikely that this challenge will be met quickly, unless there is a major technological breakthrough.
- With current technology, it is estimated that the time required, on a single processor, to factor a number of "only" 500 digits would be greater than the age of the universe !!!
- This time can be reduced by resorting to parallelization. If we accept a calculation period of 10 years, we would have to use a cluster of computers that would cover the surface of Tunisia several times, which would cost 10^{18} USD and would require an electric power of 10^{12} megawatt, which would exhaust all the world's fossil fuel resources in one day !!! (J. Preskill 2012)

Outline

Outline

Outline