THE FUTURE OF COMPUTING: WILL IT BE QUANTUM COMPUTING[1]?

By Jean-François Lavignon[2]

November 2020

INTRODUCTION

Since the beginning of mankind and the arithmetic invention, increasing our capacity to compute has been one of the objectives of humans. We have just witnessed an unprecedent exponential growth of this capacity in the last 50 years with the development of the CMOS[3] technology. For almost the same investment, you can today perform 30 million more operations than in 1970. We can speculate that this growth with your ability to digitalize information has been the engine of the acceleration of innovation observed during the 1990-2020 period.

However, this technology will not continue to deliver the same increase of our computing capacity. This paper aims to study new options that could take over and especially to look at the potential of quantum computing that has been heralded as the next generation of computing.

To position the new options, we first go back to the relationship between computing and mathematical concepts and then give a quick overview of the current computing technology. New potential computing technologies are suggested before a more in-depth analysis of the quantum computing ideas is presented with both the theoretical models and the current achievements. This leads us to propose a framework to assess the potential of new computing options. In conclusion, we propose some research priorities to sustain the growth of our computing capacity that could be essential to drive future innovations.

ACTUAL COMPUTING VERSUS MATHEMATICAL CONCEPTS

Before discussing the status and future of computing, we may ask the question why humankind has been so concerned to compute? Without conducting a historical survey of the development of sciences and technologies, it is obvious that the observation of nature and the will to understand its behavior has led us to develop very strong abstractions and instruments to represent, to describe, to predict and/or to control it.

For this purpose, we have developed mathematical concepts. One of the first enables us to enumerate things and so to count them. The integer numbers represent this concept and with them come some basic operations as addition and multiplication and then the need to compute. It is interesting to observe that this abstraction

[1] This work is part of the EXDCI-2 project that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 800957. However, it reflects only the author's view and does not express the position of the European Commission nor of the EXDCI-2 project.

[2] Founder of Technology Strategy and former chairman of ETP4HPC (2013-2016)

[3] Complementary metal oxide semi-conductor

coming from a natural need is by itself more powerful than nature as integer numbers are infinity while nature is not in our current understanding.

To represent continuous systems, real numbers have been invented. Again, this concept is much stronger than what can be represented by natural means. Real number have infinite precision while to represent them, the computers use, most of the time, 64 bits. In the case of IEEE754 binary64 this means for example that for very big numbers only a small fraction of the integers is represented and the minimal non-zero number is ~5 10^{-324}. So our technical means to deal with the concept of numbers are far from capturing all the complexity of the abstraction invented by the human brain. We see here a first limitation of any computing solution in representing the number infinity and continuity concepts.

To continue to illustrate that the human brain can deal with concepts that are very powerful, let us take the example of a very single object as a 20x20 binary matrix. We can reason about this set of objects, we can use some of the instances to represent information (see QR code), we can link one object with a computable function…

Nevertheless, we will never be able to represent all the instances. Actually, there are 2^{400} (i.e. more than 10^{120}) grid instances that is more than atoms in the observable universe estimated to be in the range of 10^{80}-10^{85} (www.uniservetoday.com).

As soon as combinatorics is introduced, our current technologies to store and compute data find their limits very quickly.

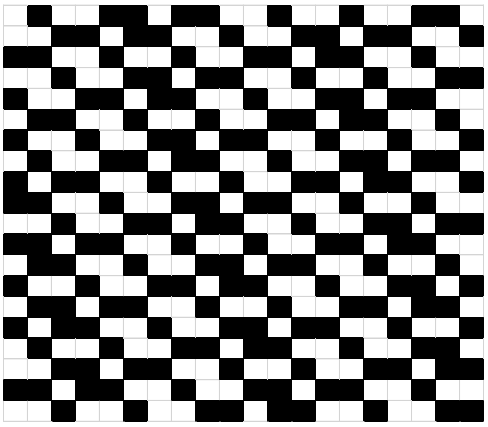

Figure 1 : there are more instances of a 20x20 binary matrix than atoms in the observable universe

We see a second limitation for any computing system either in space or time to deal with combinatorial phenomena that are nevertheless easy to access by human brain.

To explore further the relationship between mathematical concepts and computers, another important concept is the one of formal languages. Formal languages are central to define what can be computed and to program computers. Formal languages are defined as a subset of the sequences of elements belonging to an alphabet (most of the time, a formal grammar (i.e. rules to form expressions of the language) defines the expressions included in the language). As you can map the alphabet to strings of zeros and ones, formal languages can be mapped to binary languages that can be processed by computers.

The decision problems like the traveling salesman problem can be described and encoded using a formal language (a set of cities and a set of distances between some pairs of the cities). The length of the description can become a measure of the size of the problem. This concept is at the heart of complexity theory which analyzes what can be computed and with what resources in terms of space and time.

The definition of what can be computed has been formalized by Alan Turing (Turing, 1937). Alan Turing has defined the Turing machine that has been used as a yardstick to compare different computing models or systems. A Turing machine consists of

- a tape (possibly infinite) with cells,
- a head that is positioned on a cell of the tape,
- a finite set of symbols (what are written on the tape's cell),
- a finite set of states and a transition table.

The transition table, given the current state and the symbol read by the head, determines either to erase the symbol or to write another one, to move the head one cell left or right or to stay, to next state (any element of

the state set). The transitions iterate until you reach a state that belong to a subset of states defined as *final states*. Turing Machine is a computing model that has the capability to compute logical algorithms and to process decision problems. Classical computers can implement Turing machines and on the reverse Turing machines do any computation achievable on a classical computer. So, this Turing Machine model can be implemented by an actual computing system.

Now, let look at the concept of Non-Deterministic Turing Machine (NTM). This computing model is a Turing machine but, in some states, instead of having only one possible transition (i.e. write a symbol, move in one direction and enter in the next state) it can have several of these transitions. It can branch in some given states and then follow several paths of computation. This NTM can be simulated on a classical computer but the combinatory nature of the model will quickly increase the time to generate all the potential paths. Most of the computations will become intractable on a classical computer. Nevertheless, the human brain can understand the NTM behavior, reason about it and conjecture that a NTM can solve problems in polynomial time that are no computable in polynomial time using a Deterministic Turing Machine. The NTM is a computing model that cannot be effectively implemented but that is important for complexity theory. Other discussions related to computing models can be found in (Hopcroft, et al., 1967)). We see here a third limitation of computing systems that will not be able to implement all the computing models that have been created by the human brain.

From this overview, we see that mathematical concepts are much more powerful than any actual computing systems which have limitations in their ability to deal with infinity, continuous precision, combinatorial phenomena and some computing models.

COMPUTING TODAY

As said previously, humankind has put lot of effort in developing computing (or information processing[4]) systems to help to understand, predict and control natural systems. Being schematic, all these efforts has converged to:

- one way to represent information: binary representation,
- one technology: CMOS[5],
- one architecture: von Neumann.

Between all the potential options, these ones have been so successful and have concentrated so much investment that they achieve an overwhelming domination. Today, these choices result in assets of technology developments and software that sustain all the economy and society. Any other option would need tremendous investment to reach the same level of maturity and to build such a wealthy ecosystem.

The binary representation already used in vacuum tube computers has been reinforced by the invention of the transistor technology that can control the switch between two states *on* and *off* (i.e. 0 and 1) The implementation of transistor using the CMOS technology has led to the emergence of these two winners. The progress of the CMOS technology is so far the only industrial story that has provided an exponential growth[6] of a factor 2 every

[4] Here, we do not make a difference between computing and information processing; perhaps we should as it may lead to additional perspectives but it is not the purpose of this paper.

[5] Complementary metal oxide semi-conductor

[6] Growth of the number of transistors in an integrated circuit

2 years during almost fifty years. This factor of more than 30 million over 50 years is difficult to sense[7] as an integrated circuit remain a package with pins and inside a piece of silicon.

The von Neumann architecture has been introduced in 1945 (Neumann, 1945). This architecture[8] includes a processing unit, a control unit, memory to store data and instructions and input/output mechanisms. This architecture offers a great flexibility by decoupling the control and the data on which the operations are performed. It has established a standardized vision where different contributing technologies have been able to prosper (processor, memory, interface bus, storage system, programming language, compiler, application).

This architecture is the basis of almost all the processors that have been at the heart of our computing capacity. Recently, the CPUs (Computing Processor Unit) has been completed by specialized units as GPUs (Graphical Processor Unit) to increase the number of operations performed by energy unit and the density. Initially designed for image processing, GPUs can now compute over different kind of data ranging from 64 bits floating point numbers to binary numbers. Their architecture is still inspired by the von Neumann one but with less control units, reduced instruction set, more computing units and simplified data paths. So, with the same number of transistors and the same energy their provide more "compute" than CPUs.

The analysis of the current two most powerful computing systems[9] gives us more insight on the current status of computing.

Fugaku computer: vector and fast memory

Figure 2 : The Fugaku system at RIKEN (Japan)

Fugaku is the new top500 system installed in Japan beginning of 2020. A more detailed description can be found in (Dongarra, 2020). The system is composed of more than 150,000 nodes interconnected by a fast interconnect (Tofu 6D torus architecture) with an intermediate storage using NVM[10]. Each node has one processor with 48 compute cores and 4 on package HBM2[11] for a total of 32 GB of memory and a bandwidth of 1024 GB/s. The computing power is delivered by the Fujitsu A64FX processor which implements the ARM v8.2V architecture with a Scalable Vector Extension. Each processor has two 512 bits wide vector pipelines.

---

[7] If car's engines would have made such progress, with a liter of gas your car would be able to make almost 10,000 round trips around the earth ; another comparison would be that in 1970 you live in a 1 square meter room and that in 2020 your property is a 30 storeys building, each floor being a 1km by 1 km surface.

[8] In this paper we do not introduce the differences between the von Neumann and Harvard architectures although a more precise computer history analysis should make the distinction.

[9] Based on the top500 ranking

[10] Non Volatile Memory

[11] High Bandwidth Memory generation 2

The peak performance[12] in boost mode is 537 Pflops for a power consumption of 28MW. Fugaku has achieved an efficiency of more than 80.9% on the Linpack benchmark (i.e. 415 Pflops on part of the overall system) and of 2.8% on the HPCG[13] benchmark (i.e. 13 Pflops) designed to exercise computational and data access patterns that more closely match a broad set of important applications.

To achieve this performance, Fugaku relies on two main features: large vector instructions and excellent memory bandwidth. However, if your application is not able to exhibit large vector operations you will not take full advantage of this system. Again, like for the GPU, the additional computing power comes from specialized instructions that have not the same general usage than in the past. The memory bandwidth is delivered by the HBM technology much more efficient than the DDR-SDRAM[14] technology. However, this technology cannot offer the same capacity of storage. So, Fugaku has only 32GB of memory per node for 48 cores that could be low for some HPC applications. In summary, Fugaku, to deliver more performance than past systems, had to make trade-off and has chosen more bandwidth and less capacity for memory and specialized vector instructions (which are however easier to use than the tensor instructions[15] of the GPU).

Summit computer: GPU acceleration

The Summit supercomputer is the current most powerful system using the GPU technology.

It is the aggregation of 4,608 nodes using a fast interconnect to communicate and exchange data. It delivers a peak performance of 200 Pflops (peta (10^{15}) floating point operations per second) for a power consumption of 13 MW (mega (10^{6}) watts). The global memory of the system is over 10 PB (peta (10^{15}) bytes).

Each node is composed of 2 CPU that are complemented by 6 GPU to provide more floating point operations.

To reach the peak performance of such system, you would have to generate for each clock cycle in the range of 70,000,000 operations of the form *d=a+b*c* (*a,b,c* and *d* being 64b floating point numbers).

If your application does not match exactly this requirement, you will reach only a fraction of the peak performance (e.g. if you have only multiplications, you will operate at most at half the peak performance).

Figure 3 : The Summit supercomputer installed at Oak Ridge

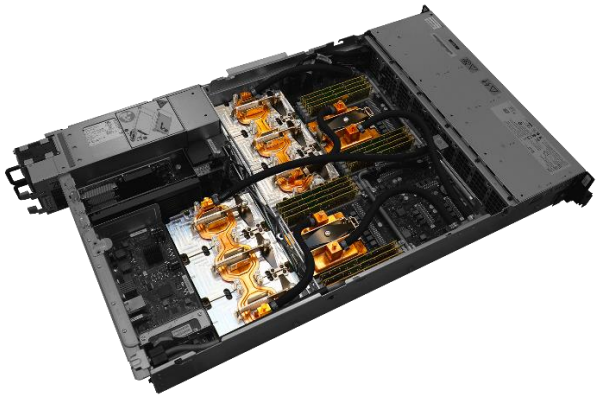

Figure 4 : One node of Summit : 2CPU + 6 GPU

[12] Peak performance is the theoretical performance with the assumption that all processing units perform the instruction delivering the greater number of operations per second.

[13] High Performance Conjugate Gradients

[14] Dual Data Rate Synchronous Dynamic Random Access Memory

[15] See the GPU presentation in next section.

On the very compute intensive and well-suited Linpack[16] benchmark, Summit has reached 148.6 for 10 MW of power. On the HPCG the performance of Summit is 2.9 Pflops. These figures represent respectively 74% and 1.4% of the peak performance of the Summit computer which is lower than the Fugaku system with 80.9% and 2.8%. In term of energy efficiency Fugaku delivers the same performance as Summit on Linpack 14.7 Gflops/W but almost twice the performance of Summit on HPCG at 0.46 Gflops/W.

Besides this difficulty to approach the peak performance, the current computing technology faces other challenges. Looking at the evolution of the GPU can help to illustrate them:

Architecture Name	Kepler	Pascal	Volta	Ampere
GPU identification	GK180	GP100	GV100	A100
Year	2014	2016	2017	2020
CMOS manufacturing process	28 nm	16 nm FinFET+	12 nm FFN	7 nm
GPU Die Size	551 mm²	610 mm²	815 mm²	826 mm²
Transistors	7.1 billions	15.3 billions	21.1 billions	54 billions
TDP watts	235	300	300	400
GPU Boost Clock MHz	875	1480	1530	1410
Peak FP32 TFLOPS	5	10,6	15,7	19,5
W/Peak FP32 TFLOPS	47,0	28,3	19,1	20,5
Peak FP64 TFLOPS	1,7	5,3	7,8	9,7
W/Peak FP64 TFLOPS = MW/Exaflops	138,2	56,6	38,5	41,2
Peak Tensor TFLOPS (FP16 matrix operati	NA	NA	125	312
W/Peak Tensor TFLOPS	NA	NA	2,4	1,3
Peak Tensor TFLOPS (FP64 matrix operati	NA	NA	NA	19,5
W/Peak Tensor TFLOPS	NA	NA	NA	20,5

Table 1 : Nvidia GPU evolution

From this table, we see that between Kepler and Pascal an important increase of computing power and energy efficiency has been achieved. This is not true at the same extent for the latest generations. To provide more flops, new specialized operations (call tensor cores by Nvidia) have been introduced. These operations come from the need of machine learning applications and are matrix multiply and accumulation of the form $D=A+B*C$ where A,B,C and D are matrix (of size 4x4 for the Volta architecture and 8x4x8 for the Ampere architecture). For application developers, it was already tricky to use the fused multiply add ($d=a+b*c$) of former CPU and GPU architectures. The tensor operations make the task even harder and we will see a decrease of the fraction of peak performance achieved by applications.

We can also see (line W/ peak FP64 Tflops) that the energy efficient has not progressed between Volta and Ampere architectures (it has even worsened). Only the introduction of more complex operator (sensor core) help to get a better energy efficient.

Near future

Looking at CMOS evolution we still see several generations coming (7nm in production, 5nm in preparation and 3nm in research). It is acknowledged that these designations of CMOS technology generations are now only a marketing name and are not, as in the past, representing the minimum gate length of a transistor (Wong, et al., 2020). Nevertheless, CMOS technology will continue to evolve in the coming years and new transistor designs

[16] See https://www.top500.org

plus other enhancements will still enable the growth of transistor density. But the energy efficient will not be improved to the same extent. It is already the case since the end of the Dennard scaling (i.e. for every CMOS technology generation, the transistor density doubles, the circuit becomes 40% faster, and power consumption (with twice the number of transistors) stays the same) around 2005. So, we clearly see the limitation of CMOS to provide more computing in the same energy budget.

To overcome this problem, the first option is to look for other technologies that can switch bit in a more efficient way than CMOS. An analysis of some of the options can be found in (Nikonov, et al., 2013).Today, no clear path has yet emerged as a good candidate. Moreover, building an industrial ecosystem that will equal the one around CMOS will need tremendous resources and will take a long time.

To continue to compute more, another option is to develop new architectures that would be more efficient than the traditional von Neumann one. As explained, von Neumann architecture is very flexible but has an important energy cost to handle the control of the execution and the data movement. In current processor, if you look at the energy budget to perform an addition, less than 5% is used for the actual operation, the rest being energy to access to the instruct, to control the execution, to manage registers. This architecture had also to develop a complex and costly (in terms of transistors and energy to operate) cache hierarchy to hide the latency to fetch data from the memory.

So, there are ways to compute more at equal technology with architectures that are adapted to an application. Several options have been developed focusing on alternative data handling and control of operation technics. To mention one of them, data flow architecture performs operation over a flow of data reducing control and data transfer overhead. A more detail analysis of new architecture options can be found in one of the EXDCI-2 reports (Lavignon, et al., 2020). It must be noticed that architecture improvements are more demanding than just waiting for a more efficient digital computing technology and only benefit to part of the applications.

PHYSICAL SYSTEMS TO COMPUTE

To overcome the limitations of digital computing implemented with CMOS chips, we might have to come back to “analog” computing and to use physical systems able to provide computation. Actually, as we compute to predict the result of the law of physics, the observation of nature governed by the laws of physics can give us the results of mathematical operations. If we set some parameters of a physical systems and measure the evolution of other parameters linked to the first ones by a mathematical equation, we get some means to solve the underlying equations. With the hypotheses that setting and measuring the parameters are faster and/or more energy efficient that using a CMOS computer and that the precision we get is adequate for our purpose, a physical system can help us to compute more.

To illustrate this idea, a very basic example would be an electric system for which you can set a current and a resistance and measure the voltage. This system is governed by the Ohm’s law and the voltage is the multiplication of the current and the resistance (U=RI). Such a physical system will provide you a multiplier. Even if common implementations of this simple system will not outperform a digital computer, this gives the perspective that physical systems can compute in a different way than digital ones.

To investigate more, let us present an experimentation done by the European project (ESCAPE) using an optical system. This optical system has been designed by a start-up company (Optalysys) and used inside the ESCAPE project focused on Energy-efficient Scalable Algorithms for Weather Prediction at Exascale. The idea was to use an optical system to compute Fourier coefficients used in spectral methods for weather prediction.

Optalysys has been investigating an optical implementation of this spectral transform. The system has been designed using standard components and its architecture is illustrated in figure 5. The fundamental idea behind this optical processor is to encode information into a laser beam by adjusting the magnitude and phase in each point of the beam.

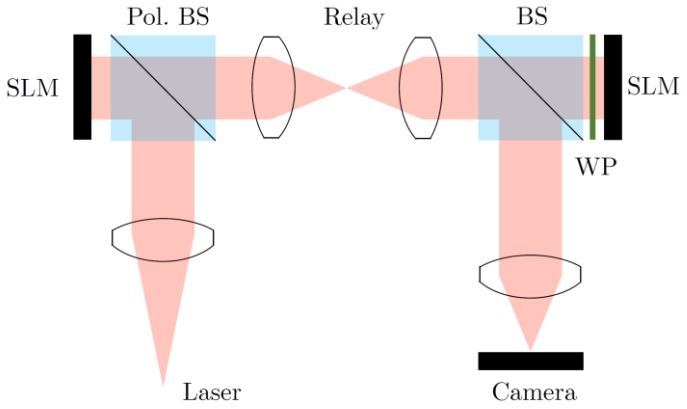

Figure 5

Illustration of the fundamental idea behind the optical processor. The laser beam is emitted on the bottom left. Two spatial light modulators (SLM) are used together to input the complex function. The system uses beam splitters (BS) and an optical relay to image one reflective SLM onto another, followed by a lens assembly which approximates an ideal thin lens and renders the optical Fourier transform on a camera sensor. The half-waveplate (WP) before the second SLM is used to rotate linearly polarised light onto the axis of SLM action (the direction in which the refractive index switches), thus causing it to act as a phase modulator.

This information becomes the Fourier transform of the initial information in the focal plane of a lens. The information can be encoded into the optical beam by using spatial light modulators (SLMs). The result of the Fourier transform can be recorded by placing a camera in the focal plane of a lens. The camera output gives the value

The system can be miniaturized in a form factor similar to a PCIe[17] board and connected to a computer through this PCIe link. The system is naturally parallel and the amount of computation will depends on the spatial light modulator resolution and of the camera used as sensor device. In the ESCAPE project, the optical device had 4 SLM with 256 levels of modulation and two cameras with a 4 K resolution (4096×3072) that can operate up to 300 Hz and the total operation power consumption of the devices is 66W. A given system will be more or less adapted to a given computational task. For example, the system used by ESCAPE can perform a convolution task (equivalent to a 2048x1536 complex to complex 2D Fourier transforms on a digital computer) with only 6.6 mJ whereas a 1920x1080 complex to real 2D Fourier transforms useful for weather prediction code will need 660 mJ. Even in the former case the energy saving compared to a digital computer is significant (on a CPU based system the energy was as least 150J). The precision of the computation is not the same as with digital computer and can be a problem (e.g. for weather forecast codes) but can serve your purpose (e.g. convolutions used in neural networks). More detail can be found on the project reports available on its web site and in the publication (Müller, et al., 2019).

This example shows that optical systems can perform computational tasks with an energy advantage over digital computers. Other physical systems can be envisioned as thermal, hydraulic, electric, electronic and photonic ones. The more promising being perhaps the optical and photonic systems for their ability to perform parallel operations and the electronic ones for being easily connected to CMOS. The investigation of physical systems has not been very active due to the tremendous success of digital computing. The upcoming limitations of digital computers may lead to a more active research and development in this field.

[17] PCI Express (Peripheral Component Interconnect Express) is a high-speed serial computer expansion bus standard.

UNIVERSAL QUANTUM COMPUTING MODEL

In comparison, quantum computing has been a very active field in the last 15 years and has been presented as the next computing technology to replace digital computing. The idea of quantum computing has been proposed in 1982 by Feynman (Feynman, 1982) and the foundation of the universal quantum computer has been described a few years later (Feynman, 1985). At first his motivation was to simulate the dynamics of quantum systems by developing quantum computers. This idea has been further developed and has led to the emergence of a model that can perform universal computation using a different paradigm than digital computers.

This description of this universal quantum computing model is inspired by (Rieffel, 2000). In quantum computing the first concept is to use quantum elements called qubits that have two quantum states noted $|0>$ or $|1>$. Each qubit can be in a superposition of these states noted $\alpha|0> + \beta|1>$ where α and β are two complex numbers verifying $|\alpha|^2 + |\beta|^2 = 1$. If you measure the state of the qubit you will find $|0>$ with probability $|\alpha|^2$ and $|1>$ with probalility $|\beta|^2$. A system with *n* qubits can in a superposition of 2^n states and so provide a space of states that grows exponentially.

The second concept is that you can operate on a qubit or a set of qubits transformations which is an evolution of the quantum state of this qubit or this set of qubits. According to quantum mechanics this transformation (called a "gate") is a reversible and unitary operator. The gates can be used to build quantum system that are entangled meaning that the measurement of one part of the system gives you information on the state of the other part.

As an example, let us take a 2 qubits system where at first the two qubits are in the state $|0>$. First you can apply a gate (called Walsh-Hadamard) that puts the first qubit in a superposition of the two states $1/\sqrt{2}\,(|0> + |1>)$. Then you can apply a *cnot*[18] gate that is a 2 qubits operation permuting the state of the second qubit if the first qubit is in state $|1>$. The resulting state of the 2 qubits is $1/\sqrt{2}\,(|00\rangle + |11\rangle)$. If you measure one qubit of this system you will find either $|0>$ or $|1>$ with equal probability of 1/2. If you then measure the second one you will find the same state with probability 1. The two qubits are entangled.

These two concepts are very powerful and Deutsch has shown (Deutsch, 1985)that it is possible to construct reversible quantum gates for any classically computable function. In fact, it is possible to conceive of a universal quantum Turing machine (Bernstein, et al., 1997). So, this model is universal and an arbitrary classical function *f* with m input and k output bits can be implemented on a quantum computer. We can assume the existence of a quantum gate array U_f that implements *f*. U_f is a m + k qubit transformation of the form:

$U_f : |x, y\rangle \rightarrow |x, y \oplus f(x)\rangle$ where $\oplus$ denotes the bitwise exclusive-or, x the state of m qubits and y the states of k qubits

Quantum gate arrays U_f , defined in this way, are unitary for any function *f* since $f(x) \oplus f(x) = 0$ we have $U_f U_f = I$. To compute $f(x)$ we apply U_f to $|x>$ tensored with k zeros : $|x, 0\rangle$ to get $|x, f(x)\rangle$.

So, if we have a m + k qubits system we can start with all qubits in the $|0>$ state. In a first step, all the first m qubits can be put in a superposition of states by applying a Walsh-Hadamard gate. Then we apply the U_f transformation on the m+k qubits. The resulting system represents potentially all the value of *f* over the 2^m inputs. But even if the system represents all the *f* values, if we measure the qubits we will only get the information about one value of *f*. To take advantage of the power of quantum computing, you need to find a clever way to

[18] *Cnot* or « connected not » transforms a qubits depending on the state of a controlling qubit

transform the knowledge of f over all inputs into something that you can output from the system. Two main methods have been proposed to extract the information:

- Detection of periodic patterns with quantum Fourier transform (e.g. the Shor's algorithm (Shor, 1994)). After a quantum Fourier transform, some qubits corresponding to the periodic patterns have larger amplitudes and can be measured, revealing the relevant information.
- Amplitude amplification (e.g. the Grover's algorithm (Grover, 1996)). The idea is to transform the state in such a way that values of interest have a larger amplitude and therefore have a higher probability of being measured.

With these constructions, qubits, gates and extraction methods, we have a computing model capable to compute all the classical computable function and a path to extract some relevant information. The question is then how to implement such a model?

Before looking at technologies to implement qubits and gates, it must be observed that, during all the operations, the quantum system needs to stay "coherent" to keep its quantum properties. If the system loses this coherence, the superpositions and the entanglements that have been created will be lost. As small interactions with the environment may cause decoherence of the quantum system, you have a difficult trade-off to isolate as much as possible the quantum system and being able to control and measure it. Keeping the coherence of a quantum system during its operation is more and more difficult as its size and the number of gates applied to it increase. So, it is a hard problem to develop quantum systems of significant size.

The implementation of qubits has also specific problems. Several technologies[19] are being used or studied to implement qubits. They have different figures of merit but all face a noise issue. To claim universal computing capability, you would need error free qubits or "logical qubit". Real systems are actually noisy and the precision of the qubit for any technology is a hard issue. Approaches to correct the noise and to get "logical qubits" have been proposed but there are incredible costly[20] for the size of the system compared to classical computing error correction scheme. Today no technology can provide qubits that match the logical behavior of the quantum computing theory.

The implementation of the gates is also a question. The universal capability of the model gives you the existence of a transformation for any computable function but how to implement this transformation? It has been showed that any transformation can be implemented using a set of basic gates. One of the minimal set of gates is formed by unitary transformations over each individual qubit and *cnot* gates between any pair of qubits (Barenco, et al., 1995). The good news is that you have minimal set of gates. One of the bad news is that, in a n qubits system, you need $O(n^2)$ *cnot* gates to have universal computing capability. It means that the implementation needs to provide interaction between any pair of qubits even the ones that will be far from each other. The current implementation even for tens of qubits have not tackle this problem and provide only *cnot* gates between neighboring qubits. Any actual implementation will also face the problem of precision of the unitary transformation. In the theoretical model, the qubit can be set in a complex number combination of its 2 states. In real systems, the precision will be limited.

Even if we may be able to implement a minimal set of gates, the number of basic operations (i.e. either a unitary transformation over 1 qubit or a *cnot* between two qubits) to implement any quantum transformation is another question. (Barenco, et al., 1995)has shown that, in a n-qubit system, the transformation which is the multiplication of the last qubit by a unitary transformation conditioned by the n-1 other qubits can be implemented by $O(n^2)$ basic operations. A more general U_f unitary transformation over the n-qubits systems can

[19] Among them trapped ions and superconducting circuits technologies have led to the largest systems.

[20] For physical qubit error rate of 10^{-3}, you would need more than 15,000 physical qubits to implement one logical qubit.

be constructed using a finite number $O(n^3 4^n)$ of one or two qubits gates. This latest result shows that there is no evidence that any interesting transformation can be computed in a tractable time by a quantum computer with the minimal set of gates. The move to a minimal set of gates, has introduced a combinatorial effect on the number of gates to perform a transformation that leads to a time issue.

The coherence of quantum system, precision of qubits, basic set of gates implementation and number of gates to achieve a given transformation are open questions to implement the universal quantum computing model. These questions are acknowledged by the physicists since many years and one good reference is the DoE report of 2004 (Dr. Richard Hughes, 2004). The five Di Vincenzo criteria[21] highlighted in this report, can still be used to analyze the progresses made and to look at the current roadblocks. A more recent report on Quantum Computing issued by the US National Academy of Sciences (Horowitz, et al., 2019), shows that the same hard problems are still on the table.

Will we have a universal quantum computer of significant use (equivalent of around 4,000 accurate qubits[22]) in 10, 30, 100 years or never[23]? My view is that we cannot today answer this question. Any implementation path would need significant breakthroughs to advance toward this goal. Until we see some progress for at least one of the current roadblocks, it is better not to give an answer to this question.

The adiabatic quantum computational model (based on a controlled evolution of a quantum system Hamiltonian)[24] has been proven to have the same potential as the universal quantum computing model (Aharonov, et al., 2008). But it also faced some similar issues to be implemented: control of the Hamiltonian and coherence time compare to time to perform the Hamiltonian transition. Again, significant breakthroughs would be needed to get a full implementation of this model.

CURRENT QUANTUM COMPUTING SYSTEMS

After the analysis of the universal quantum computing model, it is worth to look at the current situation regarding efforts to implement quantum computers. The current systems are defined as NISQ – Noisy Intermediate-Scale Quantum (see (Preskill, 2018)). To illustrate the level of achievement and the usage that can be done, we present two different approaches the one from Google and the one from D-WAVE.

[21] These five criteria are :

- A scalable physical system with well-characterized qubits
- The ability to initialize the state of the qubits to a simple fiducial state
- Long (relative) decoherence times, much longer than the gate operation time
- A universal set of quantum gates
- A qubit-specific measurement capability

[22] The Shor algorithm would need in the range of 4,000 logical qubits to factorize a 2048 bits number used by RSA

[23] Not all the computing models can be implemented as shown with the Non-Deterministic Turing Machine (NTM). It might also be the case for the Universal quantum computing model.

[24] The presentation of the D-WAVE approach in the next section gives some more information about the basis of adiabatic quantum computation

Google system

The Google system that has be used to announce quantum supremacy is described in (Arute, et al., 2019). This system is based on superconducting qubits and Google has designed a chip (the Sycamore processor) with 53 qubits layout in a rectangular array, each connected to its four nearest neighbors with couplers. The circuit can be calibrated and benchmarked to decrease the error rates. After the calibration process, if you make several measurements (few million measurements in the case of the Sycamore chip and the sequence of gates used in the experiment) you can get a trusted result.

When you apply to this circuit a sequence of gates (some are one qubit gates and others two qubit gates) and measure all the qubits you get a bitstring (of 53 either 0 or 1 bits). For a given sequence of gates if you repeat this sequence you will generate a set of bitstrings. Owing to quantum interference, the probability distribution of the bitstrings is not uniform, some bitstrings are much more likely to occur than others. If you use the Sycamore circuit, you can get access to the probability distribution of the set of bitstrings for a given sequence of gates.

Computing such a probability distribution is obviously out of reach of a classical computer as soon as the number of qubits or/and the number of gates are increased. You face a combinatorial problem that we have seen is intractable with classical computers even for problems that seem small to our understanding.

So Google has demonstrated that you can built a quantum system that can be controlled, calibrated, benchmarked, observed and that have a size and reliable enough operation that make it impossible to simulate with a classical computer. It is a remarkable engineering achievement but what does it really means for computing? If you are interested in observing the probability distribution of a given sequence of gates the circuit is useful but for other tasks you currently have not path to take advantage of it. The system can be viewed as a "physical system" not simulable by a classical computer with some control on it (the sequence of gates) and the ability to observe a probabilistic result. However, if your computing problem does not exactly match that, you cannot find any help from the Sycamore system.

D-WAVE system

The D-WAVE system is described in (McGeoch, et al., 2019). This system is based on qubits that represent the spin (-1 / +1) of a lattice of superconducting circuits. Some of the qubits interact by pair through a coupler. The number of qubits and the topology of the coupling is specific to the chip generation of the D-Wave machine[25]. This system uses quantum annealing to find the ground state of a Hamiltonian that is of the form:

$$H_{final} = H(E,V) = \sum_{i,j \in V} J_{i,j}\sigma_i\sigma_j + \sum_{i \in E} K_i\,\sigma_i$$

(*E* is the set of qubits, *V* the set of pairs of qubits that are coupled, $J_{i,j}$ the coupling coefficient, K_i the field applied to qubit *i* and σ_i the value of the qubit *i*).

The adiabatic quantum computation operates starting from a simple Hamiltonian H_{init} for which the ground state is known and can be used as the initial state of the quantum system. It then "slowly" evolves toward the H_{final} by changing the Hamiltonian. A simplify representation of the process between time 0 and *T* (end of the adiabatic computation) would be to apply at *t* the Hamiltonian:

[25] The D-WAVE2000Q has around 2000 qubits and a topology called Chimera that is a mix of 4 by 4 qubits coupling and of interactions between groups of 8 qubits. The newly announced Advantage quantum system has around 5,000 qubits and more connectivity.

$$H(t) = A(t)H_{init} + B(t)\, H_{final}$$

With $A(0) \gg B(0)$ and $A(T) \ll B(T)$

At the end of the process, the value of the qubits can be read and should give you the ground state of H_{final}. However, the physical limitations of the D-WAVE system or interference from external noise can have an adverse effect on the calculation. This means that the D-WAVE system does not guarantee to return a ground state solution every time; thus, it is prudent and cost-effective to repeat the annealing many times for each input.

This D-WAVE system is a superb engineering achievement. Nevertheless, it illustrates the limitations of current quantum system:

- Even if it is based on quantum annealing thought to be less affected by noise than gate-based quantum computing, the noise impact is not yet mastered;
- The topology of the connections is limited and leads to mapping problems and to the reduction of the number of qubits effectively used;
- No theoretical results exist about the speed-up that the system can provide compared to classical computing methods;
- In practice no demonstration of real applications that find a clear benefit have yet been achieved. Some problems designed to be “difficult” for classical computing and “adapted” for quantum annealing are still better solved by classical approaches (Mandrà, et al., 2018).

The D-WAVE systems will continue to evolve with increased number of qubits and new interconnection topology (Boothby, et al., 2019), nevertheless some of the limitations will remain.

Summary

With these two examples, we see that the NISQ era have given birth to remarkable engineering achievements with quantum systems that can be controlled and measured. However, they are still far away from the error free quantum computing models with no known paths to reach this level. One of the limitations is the topology of the interaction between the qubits that is limited and specific to each hardware system. This reduces the flexibility of the quantum system and makes the mapping of your problem a complex task. You must adapt to the physical features of the quantum device rather than you get a system designed to solve some computing tasks that are used in applications.

These quantum systems are similar to the physical system concept presented in this paper with the difference that the underlying hardware uses quantum phenomena. This could be an advantage to simulate quantum systems as atoms or molecules and could lead to quantum simulators. However, they should not be considered superior over other non-quantum physical systems because inheriting from the universal quantum computing model a superiority over other computing models. Actually, none of the implemented quantum systems has a theoretical foundation proving that they can universally compute or/and in a more powerful way than classical computing and so they should be only judged on what these physical systems can compute.

HOW TO ASSESS FUTURE COMPUTING OPTIONS

The previous analysis of current computing progresses, new physical systems or quantum systems shows that none of the new options proposes a complete replacement of the current digital computers. They can be candidates to accelerate some tasks but not a universal solution. So, the integration of these new options with the current technology will be a central topic for their adoption. Therefore, to assess the value of a new option,

we believe it is important to focus from the beginning on how the interaction with standard computer will be designed. We see several criteria to consider.

The first criterion is the level of integration with the current computing technology. This level can be inside the die, inside the package, inside the board, inside the server or at system level. This would mean different integration technologies ranging from hybrid silicon process, 2.5/3D integration, coherent communication protocol or communication software stack. This criterion will also have a strong impact on the new software elements that must be developed to enable the new technology.

The second criterion is the "value proposal" of the new technology. This includes several elements as the computational task performed by the new option, its frequency of operation, its communication bandwidth with the hosting system, its advantage over CMOS (energy or time). These elements will serve to assess the performance of the new technology in operation and to analyze how it outperform CMOS for some computational tasks.

The third criterion is the maturity level of the new technology. The expected time to get a reliable device is important to assess how the new technology will impact computing. The maturity of optical systems, photonic solutions, memristor technologies, quantum systems is very different. If we invest in these technologies, we can expect to integrate them in production system at different time horizons.

These three criteria give the vision on the path for a new technology to contribute to the future of computing. It enables to prioritize the research according to your objectives that can also be divers (in terms of risk you want to take and of expected horizons).

In addition, it is important to integrate in the research of new computing solutions an application vision. As the new technologies will only cooperate with standard computing solutions, the application point of view will give the complete picture of the mandatory integration. Focusing on application will help to identify and develop the software to support the new technology. It will also provide a more accurate evaluation of the advantages in terms of time, energy or cost to compute. This complete application vision will have to guide the research on future computing technologies.

We must acknowledge that the path for new computing options to reach production level will encounter severe pitfalls. First, they are less generic that the current computing solutions and so the domain of application will be smaller. Finding a field with sufficient attraction will be difficult for most of the potential options. Second, the CMOS technology has raised the threshold so high that most of the options will need several generations to be really in position to outperform it. This means a long-term commitment for the funding of the first generations that will not be successful on the market. Finding a self-sustainable funding model will be a strong challenge for all the new computing technologies.

CONCLUSION

This paper does not present any new scientific knowledge. However, it looks at some new options to compute and gives a framework to analyze and compare these options that could be useful to assess the potential of different research paths.

One of the objectives of this paper was to analyze the tremendous interest generated by quantum computing which has generated very positive communication and being presented as the next computing generation. It is worth to notice that this communication comes most of the time from commercial or political sources that lack the expertise of physicists. The physicists are surprise of such enthusiasm (Preskill, 2018) but this is not their job to discourage this tremendous interest for their field. It is the responsibility of IT[26] people to make a realistic analysis of the potential of quantum computing and they have to be careful not to apply an analogy with the transistor history in the quantum field that faces different implementation challenges.

Quantum computing has indeed a very strong Science Appeal for computer scientists compared to other research options:

- New field that proposes a completely different paradigm for IT and that challenges our usual vision of physics;
- A strong mathematical framework;
- New research topics to map applications onto the quantum systems.

However, the progress of quantum computing during the last fifteen years had no impact on our ability to compute more (Horowitz, et al., 2019). From our analysis, it is not because we are funding quantum computing research that we will have more powerful computer in the coming years. The time horizon of pay-off for computing of research on quantum technology is still unknown. If we do not want our ability to compute to stall, we need to research in other domains than quantum computing too.

Computing (and more globally data processing) has been a key element to power the evolution of your society. The exponential growth of computing capacity brought by digital CMOS for 50 years (1970-2020) has been one of the reasons of the acceleration of the innovation we have witnessed during this period. This mainstream computing technology (i.e. digital CMOS) will stall soon. If increasing the computing/processing capability beyond the level reached at that time is of importance for mankind, we must research alternatives. The investment should be made with a fair assessment of the potential of the different candidates and the author's view is that we should not put all the eggs in the quantum basket. We must also consider research on new materials, new architectures and physical systems. Even if for all these paths a self-sustainable R&D scheme will be difficult to reach, it is the only way to provide more computing capability in the future.

[26] Information Technology

ACKNOWLEGMENT

The reviewers of this paper, François Bodin, Stéphane Requena and Ingolf Wittmann have provided valuable comments to improve its quality. Nevertheless, this does not mean that the reviewers share the conclusions presented here.

REFERENCES

Aharonov Dorit [et al.] Adiabatic Quantum Computation is Equivalent to Standard Quantum Computation [Article] // SIAM Journal of Computing. - 2008. - Issue 37 : Vol. Vol. 37.

Arute Frank [et al.] Quantum supremacy using a programmable [Article] // Nature. - 2019.

Barenco Adriano [et al.] Elementary gates for quantum computation [Article] // Physical review. - 1995. - Vol. V1.

Bernstein E. et Vazirani U.V. Quantum complexity theory [Article] // Society for Industrial and Applied Mathematics Journal on Computing . - 1997. - 5, 1411–1473 : Vol. 26.

Boothby Kelly [et al.] Next-Generation-Topology-of-DW-Quantum-Processors [En ligne] // D-WAVE. - 2019. - https://www.dwavesys.com/sites/default/files/14-1026A-C_Next-Generation-Topology-of-DW-Quantum-Processors.pdf.

Deutsch Quantum theory, the Church-Turing principle and the universal quantum computer. [Article] // Proceedings of the Royal Society of London Ser. A A400, 97–117. - 1985. - Ser. A A400, 97–117.

Dongarra Jack Report on the Fujitsu Fugaku system [Rapport]. - [s.l.] : University of Tennessee, Knoxville, 2020.

Dr. Richard Hughes A quantum information science and technology roadmap [En ligne]. - 2004. - https://qist.lanl.gov/pdfs/qc_roadmap.pdf.

ESCAPE ESCAPE project [En ligne] // ESCAPE project. - http://www.hpc-escape.eu/.

Feynman Richard Quantum Mechanical Computers [Article] // Foundations of Physics. - 1985. - 6 : Vol. v21.

Feynman Richard Simulating Physics with computer [Article] // International Journal of Theoretical Physics. - 1982. - Vol. 21.

Grover L. K. A fast quantum mechanical algorithm for database search [Conférence] // Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing. - 1996.

Hopcroft J.E. et Ullman J.D. Nonerasing stack automata [Article] // Journal of Computer and System Sciences. - 1967. - Vol. 1. - pp. 166-186.

Horowitz Mark et Grumbling Emiy QUANTUM COMPUTING, Progress and Prospects [Rapport]. - [s.l.] : National Academy of Sciences, 2019.

Lavignon Jean-François et Duranton Marc Trends and potential synergies between electronics, photonics and HPC [Rapport]. - [s.l.] : https://exdci.eu/resources/public-deliverables, 2020.

Mandrà Salvatore et Katzgraber Helmut A deceptive step towards quantum speedup detection [Article] // Quantum Science and Technolology. - 2018.

McGeoch Catherine [et al.] Practical Annealing-Based Quantum Computing [Article] // IEEE Computer Magazine. - 2019. - Vol. vol. 52.

Müller Andreas [et al.] The ESCAPE project: Energy-efficient Scalable Algorithms for Weather Prediction at Exascale [Article] // Geoscientific Model Development. - 2019. - 4425–4441 : Vol. 12.

Neumann von First Draft of a Report on the EDVAC [Rapport]. - 1945.

Nikonov Dmitri et Young Ian Overview of Beyond-CMOS Devices and a Uniform Methodology for Their benchmarking [Article] // Proceedings of the IEEE. - 2013.

Optalysys Optalysys [En ligne]. - https://www.optalysys.com/.

Preskill John Quantum Computing in the NISQ era and beyond [Article] // Quantum. - 2018. - Vol. v2.

Rieffel Eleanor An Introduction to Quantum Computing for Non-Physicists [Article] // ACM Computing Surveys. - 2000. - Vol. 32.

Shor P. W. Algorithms for quantum computation: Discrete log and factoring [Conférence] // Proceedings of the 35th Annual Symposium on Foundations of Computer Science. - 1994.

Turing Alan On Computable Numbers, with an Application to the Entscheidungsproblem [Article] // Proceedings of the London Mathematical Society. - 1937. - Vol. vol. 43, p. 544-546.

Wong Philip [et al.] A density metric for semiconductor technology [Journal]. - [s.l.] : Proceedings of IEEE, 2020. - 4 : Vol. vol 108.

www.uniservetoday.com https://www.universetoday.com/36302/atoms-in-the-universe/ [En ligne].

The EXDCI-2 project receives funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 800957.