IJCSN - International Journal of Computer Science and Network, Volume 8, Issue 6, December 2019
ISSN (Online) : 2277-5420
www.IJCSN.org
Impact Factor: 1.5

Wires, P=NP in Applied Logic

Rashad Zaguia

Computer engineer ,University Of Ottawa,
Ksar Helal, Monastir 5070, Tunisia

Abstract

- Before this work:

The field of computer engineering was about using ('nand' and 'nor') universal logic gates to make processors and programs. Some of the field of electric engineering was about realizing those universal gates using transistors. Some of the field of chemical engineering was about using rare earth mines as semi conductors to make those transistors.

- After this work:

A wire f(x)=x is proven to be a universal function. The field of computer engineering does no longer require the electric engineering field to make transistors. Wires are sufficient. The field of chemical engineering can still help increasing the speed by allowing a miniaturization of conductors.

Keywords – *First order logic, logic gates, Turing machine, Foundation of computer architecture, using logic gates to program, networks of wires to replace transistors.*

1. Introduction

1.1 General context

We require faster and better approaches to optimization, in order to be able to solve our problems given our limited resources planet.

A TSP of cities [1] is easier than a TSP of fewer points on a circuit of a cpu [2]. [3].

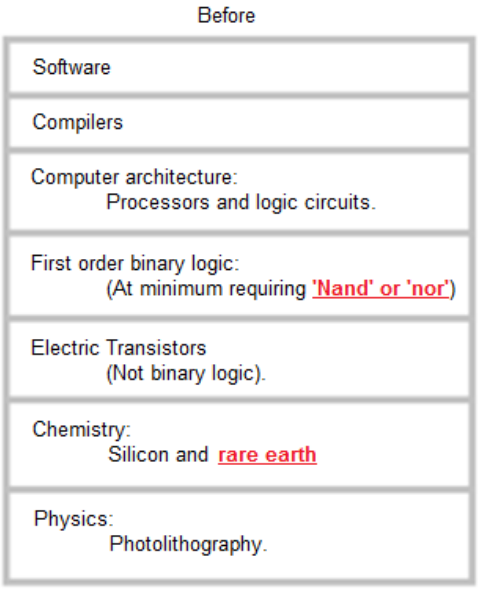

Fig. 1

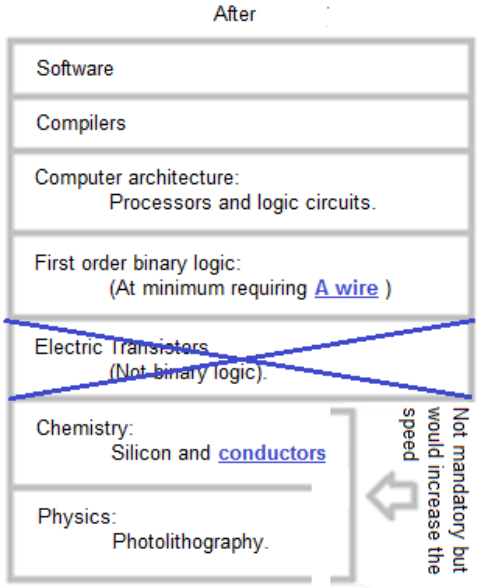

Fig. 2

1.2 Interest

This is interesting for, and would benefit

- Countries at fight or suffering management problems because of rare-earth mining.
- Processors and integrated circuits makers.
- Computer engineers (by providing more dexterity and capability).

- The industry of nano-scale.

1.3 Background (State of the Art)

1.3.1 Equivalence between algorithms, state machines and logic gates.

An algorithm has instructions and variables.
Variables together are at a state.
Inputs would possibly change that state and generate an output.
Algorithms and state machines are equivalent.
State Machines can be reduced to logical gates.

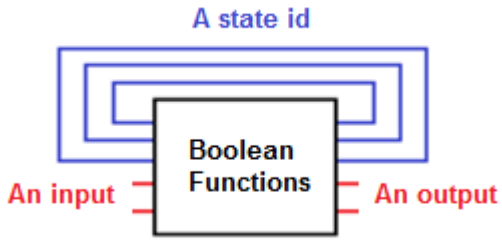

Fig. 3

1.3.2 How to make any Boolean function? [4]

A Standard Boolean function has many inputs and one output.
Here is an example:

u	i	s	a function b
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

b =
not(u) and i and s
or
u and not(i) and s
or
u and i and not(s)
or
u and i and s
basically b=011 or 101 or 110 or 111
(Function b implementation could be simplified, it is not the objective here)

THE GENERAL CONCLUSION: the set of functions {'and','or','not'} is universal (a set capable of making any function).
Universal functions are capable of making all Math.
Here is addition:
(Note: Boolean xor has a truth table and there for realizable with the universal set of functions)

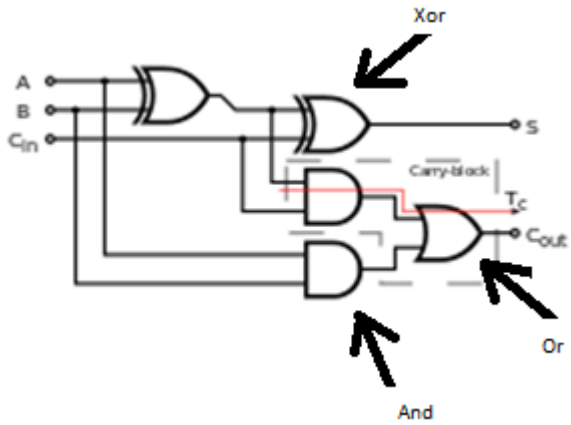

Fig. 4 A 1 bit full adder:

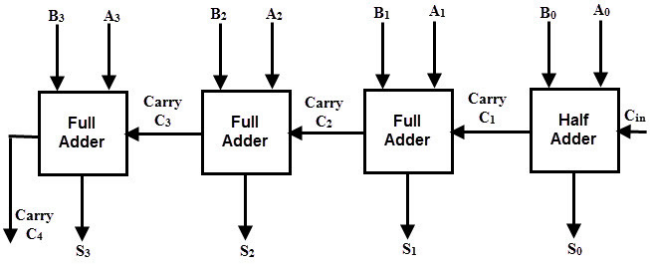

Fig. 5 A 2 numbers adder (4 digits each)

1.3.3 'Nand' gate universality proof [4]

Here is 'not and' also called 'nand' truth table:

x	y	nand
0	0	1
0	1	1
1	0	1
1	1	0

x nand x = not(x)
[not(x) is now possible using 'nand' only]
not(x nand y) = x and y
['and' is now possible using 'nand' only]
not(x) nand not(y) = x or y
['or' is now possible using 'nand' only]
a 'nand' is capable of making the set of univeral functions {'and','or','not'} and there for a 'nand' is universal.

1.3.4 'Nor' gate universality proof [4]

x nor x = not(x)
not(x nor y) = x or y
not(x) nor not(y) = x and y
a 'nor' is capable of making the set of functions {'and','or','not'} and there for a 'nor' is universal.

1.3.5 Transistors and relays [5]

To ease understanding here, let's replace fast transistors with slow relays.

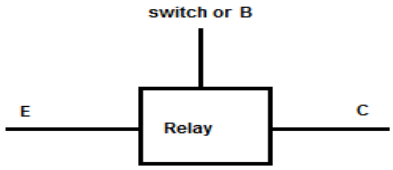

Fig. 6

If B is at '+': E gets connected to C with a magnet.
If B is at '-': E gets separated from C with a magnet.

Fig. 7

N is a wire at rest, the majority of electrons do not move in a direction or in another. It is not at 0. It is not at 1.

Fig. 8

D is a wire that is by default at 0 (if 0 is '-').
If D connects to a '+', it would be a '+', because the resistance would hold against a short circuit.

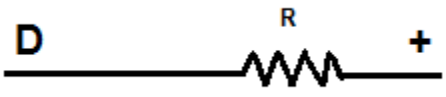

Fig. 9

D could be put by default at 1 (if 1 is '+').
If D connects to a '-', it would be a '-', because the resistance would hold against a short circuit.

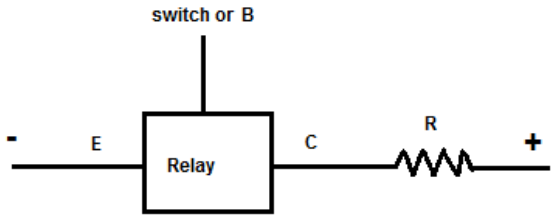

Fig. 10

If B is '-' then C is '+'
If B is '+' then C is '-'
C = Not(B)
Relays and transistors are similar.

1.3.6 How to use transistors to make universal logic gates [4]

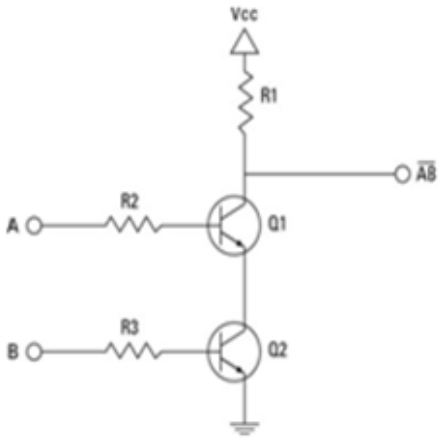

Fig. 11

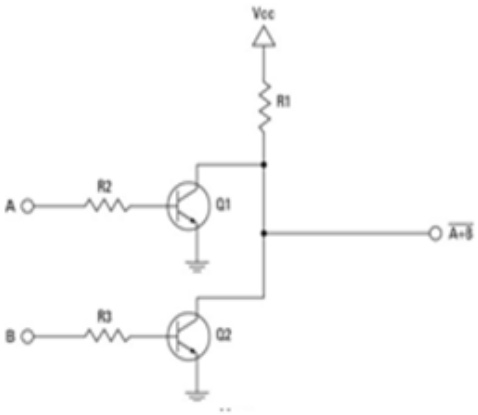

Fig. 12

1.4 Related Work

There exists many ways to make a 'nand' a previously known smallest universal logic gate.
The gate 'nor' was also is the same category.

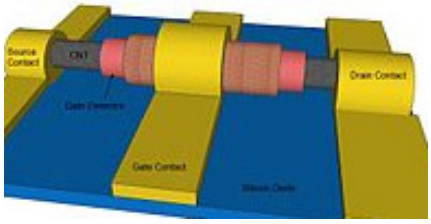

Fig. 13 CNTFET [8] Transistors made with Carbon

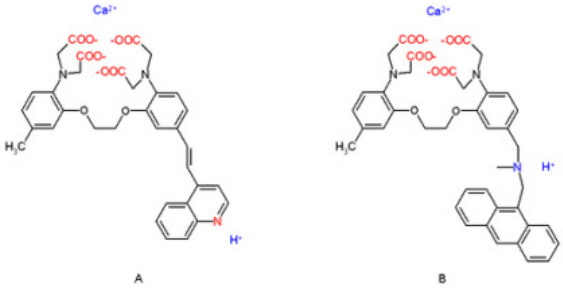

Fig. 14 Molecular logic gates [9]

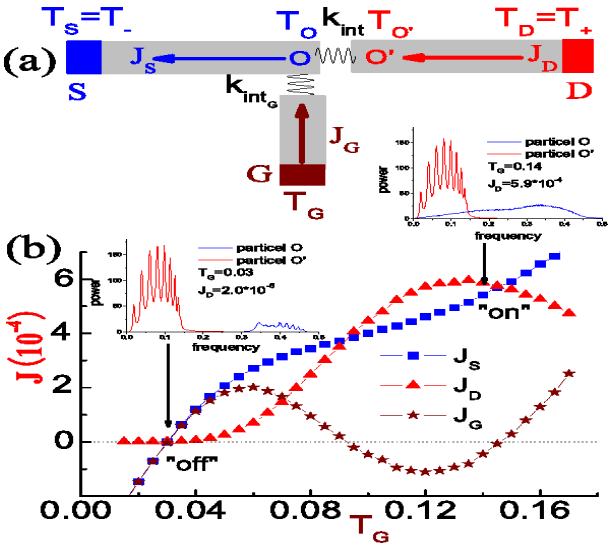

Fig. 15 Thermal logic gates [10]

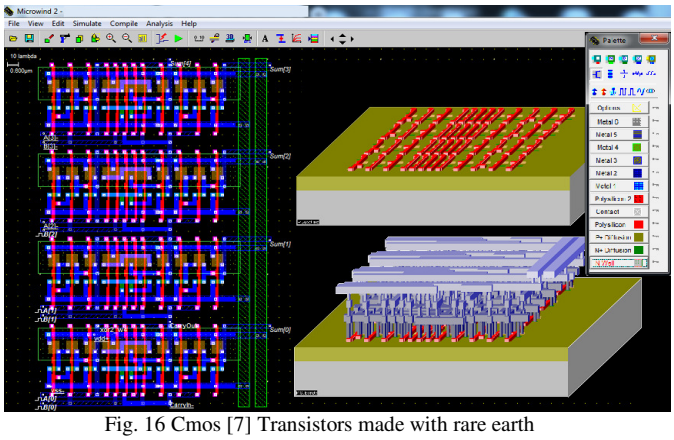

Fig. 16 Cmos [7] Transistors made with rare earth

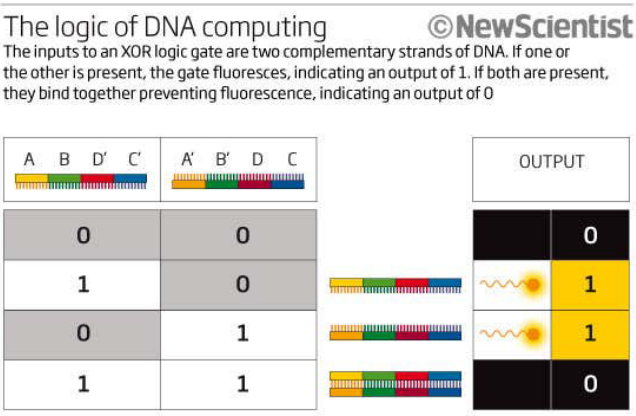

Fig. 17 DNA logic gates [11]

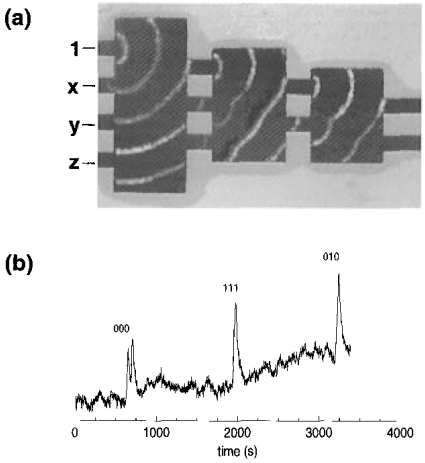

Figure 1. (a) Three-layer chemical wave network representing an

Fig. 18 Chemical wave logic gates [12]

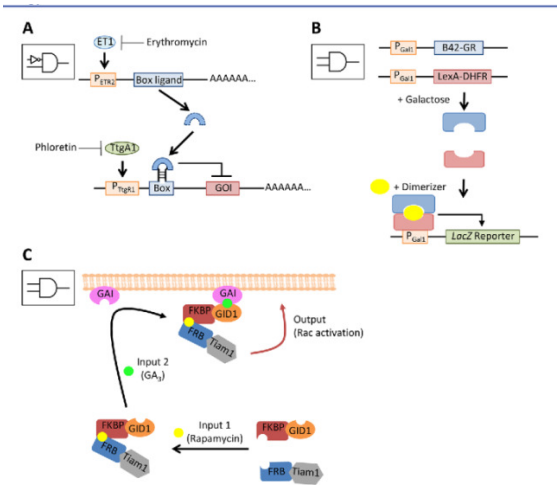

Fig. 19 Biomolecule-Based Logic Gates [13]

1.5 Distinction form the related work

This paper proposes a simpler and faster way to make logic gates.
It is by using simple wires (conductors).
Proving that f(x) = x is universal.

2. Statement of the problem

If the set of gates {'xnor', xor} is verification and the set of gates {'nand','nor'} is implementation can we make implementation with verification only ?
Can we make implementation with wires only? (Verification can make a wire)

3. Methodology

As we were capable of making verification with implementation but not the reverse, what if we constrain our self to verification only to focus our effort?

4. Proof of concept

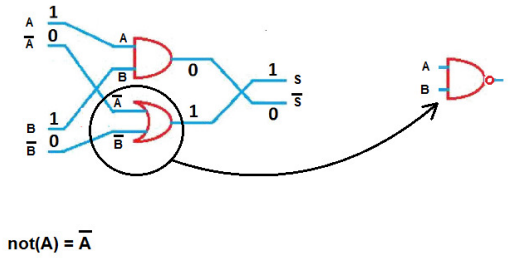

Fig. 20

Not(a) or not(b) is equivalent to A nand B
This figure demonstrates how to make a 'nand' with {'or', 'and'}

5. Implementation

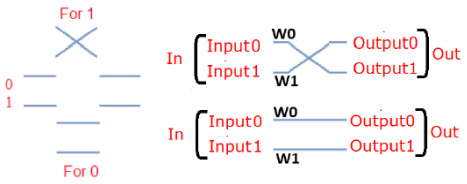

Fig. 21

Replacing 0 with the couple (0,1) and 1 with the couple (1,0).
Allowed having a not by simply flipping two wires.

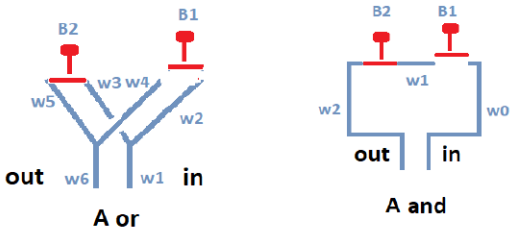

Fig. 22

Emitting a signal at w1 would return to w6 if one of the buttons is pressed. Emitting a signal at w0 would return to w2 if both buttons are pressed.

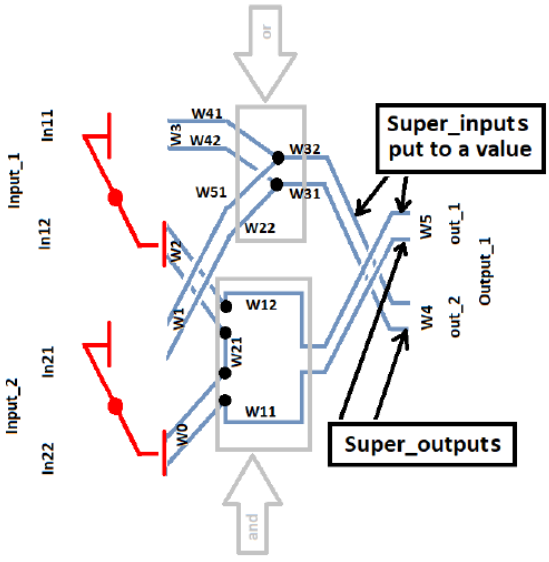

Fig. 23

This is a 'nand' made with wires only composed with the described above.
Emitting a 5 volts at w12 and w32 would have a 5 volts at w11 or w31 but not at both.
The possible outputs would be a 5v and a 0v or a 0v and a 5v. It is a gate that composes well with its self.

6. Analysis and Results

6.1 Example and demonstration

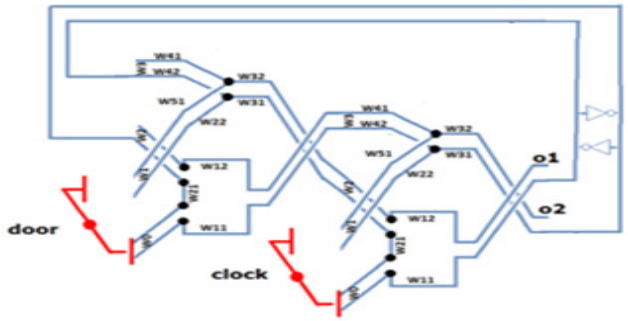

Fig. 24

Here is a simple Alarm-system program written with logic gates, example:

If the clock is on watching time and the door opens the alarm goes on even if the door recloses.

If the clock is shut down the alarm goes off.

Opening the box where the clock is, would take time for any-person.

Clock: 0 access prohibited, 1 access permitted.

Door: 0 closed, 1 open.

Cyrene: 0 silent, 1 making noise.

The program would be:
(Cyrene nor Door) nor Clock = Cyrene

The Alarm example has memory.
It files back the Cyrene status to the logic gates structure.
One returned bit of memory requires two traditional 'not' gates that use only 1 transistor each.

A 'not' with resistances only (long wires) is also presented.
Current encoding and initialization set the room door and the clock of the alarm, to zeros.

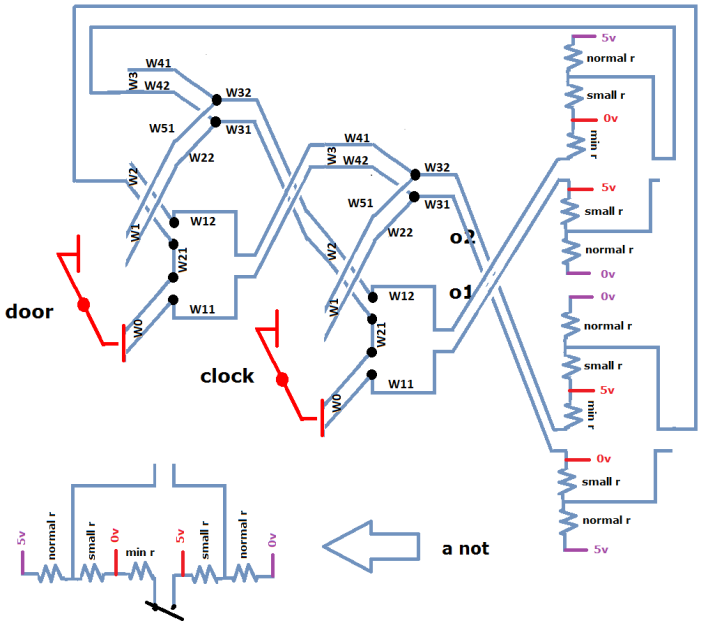

Fig. 25

(Notice: at the beginning the red generator decides the value).

7. Conclusion

In this paper a wire as a function F(x) = x is proven universal.
Conductors are cheaper and faster than semi-conductors gives advantage to wires.
A wire could always transmit light, which would allow us to compute at real light speed.
Any function that can make a wire is universal.
f(x) = x = x xor 0
['xor' is capable of making a wire, 'xor' is universal].

Considering section "2 statement of the problem"
P is equal to NP

Over long term it means better delivery, better infrastructure, better resources usage for much less encryption and online banking. Simple payments on delivery are a solution.

There is no conflict of interest.

References

[1]
http://www.math.uwaterloo.ca/tsp/gallery/itours/usa13509.html

[2] http://www.math.uwaterloo.ca/tsp/gallery/itours/pcb3038.html
[3] William Cook, reported in a video after his work on Concorde (a TSP solver)
[4] Techniques from before 1949, since: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.186.6171&rep=rep1&type=pdf
[5] Transistor : 1948-2019, The first bipolar junction transistors were invented by Bell Labs' William Shockley, which applied for patent (2,569,347) on June 26, 1948.
[6] http://www.claymath.org/millennium-problems/p-vs-npproblem
[7] Cmos logic gates, http://www.microwind.org
[8] CNTFET logic gates, https://www.elsevier.es/en-revista-journal-applied-researchtechnology-jart-81-articulo-design-evaluation-energy-efficientcarbon-nanotube-S1665642317300378
[9] Molecular logic gates, https://science.sciencemag.org/content/285/5426/391
[10] DNA logic gates, https://pubs.acs.org/doi/abs/10.1021/ja047628k
[11] Thermal Logic Gates, https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.99.177208
[12] Chemical wave logic gates, https://pubs.acs.org/doi/abs/10.1021/jp961209v
[13] Biomolecule-Based Logic Gates, https://pubs.acs.org/doi/abs/10.1021/sb3001112