Thanks to my parents without witch care you would of not had these papers to read.
Advancing in history in our days is more important that advancing in geography.
I'm a Tunisian computer engineer with a Canadian diploma who do not have access to journals.
The following references are from the internet.
I did not succeed publishing prior versions of this work in a journal.

References (public data):
[Quantum Entanglements, used the rules encoding]
http://www.youtube.com/watch?v=0Eeuqh9QfNI.
[with spacial thanks to the professor in the following series of videos]
http://www.youtube.com/playlist?list=PL004010FEA702502F&feature=em-share_playlist_user
Wikipedia.org introduced this problem to me.

Title

Wires replacing transistors enabling light speed.

Background of the Invention.

I tried as a computer engineer to respond to the following question :
If it is easy to check that a solution to a problem is correct, is it also easy to solve the problem?
Formulated by Stephen Cook and Leonid Levin in 1971.
My main help is a decision I took in front of my self to no longer lye.
It stabilized logic and increased my performance.
It is incompatible with hacking and crime in general.

Brief Summary of the Invention.

It is about trying to make super calculators and circuitry with any conductor (or fiber optics). It avoids or reduces rare earth resources usage of classical circuitry.
It simplifies witch would help simulations and optimization of complex problems. Verified optimums results are known in economy for being efficient for solving money crisis.

Brief description of the several views of the drawing.

D0 : Circuitry showing how if constrained to {xor, xnor} (verification) doors verification is equal to implementation.
D1 : Circuitry showing how to connect 3 verification doors together in order to have a 3 input output pins door. Any 2 pins can be chosen to be inputs and the third would be the output.
D2 : Circuitry showing how to make an approximation of {and,or} doors.
D3 : 3 wires making an xor door (the idea is more in the encoding).
D4 : A mechanical system demonstrating how to format the inputs correctly for xor "processors or circuitry". Classical circuitry using transistors could also be used. the idea is more on the encoding.

Detailed Description of the Invention.

The number of binary logical doors of I=inputs and O=outputs are 2^(O*2^I) logical doors.
For verifying one of those logical doors we can use a logical door of I+O inputs and 1 output.. The verification logical doors are 2^(1*2^(I+O)) logical doors .
Verification logical doors are not equal to logical doors.
And there for : P is not equal to NP unless we use verification only to implement.
The verification function that tells whether 2 digits are the same or not is the following :

input0	input1	output
0	0	1
0	1	0
1	0	0
1	1	1

Witch is known in classical computer engineering as an xnor.
If we combine the 3 columns 3!=6 combinations the function would stay the same.
Please notice that there is also the xor that flips the 0 and 1 and do have the same characteristics.
xor can be made with xnor :
xor(i0,i1) = xnor(0,xnor(i0,i1))
So P equals NP if we constrain, focus and capitalize effort on implementing any using xnors only.
See : D0.

At the beginning I could make with xnors only 2^I programs. The choice of the maker of the program is equal to the choice of the user. From a decay I could pick only 1 possibility the rest were determinations of that choice. By a decay I meant a usage of a digit.

Examples (tables):

B	C	D	1	S	
0	0	0	1	1	odd then using 1
0	0	1	1	0	even then using D , 1
0	1	0	1	0	even then using C , 1
1	0	0	1	0	even then using B , 1
0	0	0	1	0	even then using A , 1

S = D xor C xor B xor A xor 1

A	B	C	D	S	
0	0	0	1	0	even then not using
0	0	1	0	1	odd then using C
0	1	0	0	1	odd then using B
1	0	0	0	1	odd then using A

S = C xor B xor A

B	C	D	1	S	
0	0	1	1	0	even then using D , 1
0	1	1	1	1	odd then using C, D , 1
1	1	0	1	1	odd then using B, C, 1
0	0	1	1	1	odd then using A, D , 1

S = D xor C xor B xor A xor 1

Many thinks I could not make. I was stuck and that was for too long.

The possibility of sorting in (n*log2(n)), sorting column of digits per column of digits realization with xors allowed me having an idea. 3 pins common to 3 xors as follows:

pin0=xor0(pin1,pin2)

pin1=xor1(pin0,pin2)

pin2=xor2(pin0,pin1)

Was not posing a short circuit problem due to the combinatorial characteristics of the xor door. As a consequence entering 2 pins would determine the third one and pins would be inputs and outputs. Using this special door I'm naming xors we could create universal doors .A nand as an example, defined as pin0=nand(pin1,pin2) would be :

pin0=xors(1,0)

pin0=xors(1,pin1)

pin0=xors(1,pin2)

“None lying democracy can govern logic.”

programs made with xors could run both ways where there is not much difference between inputs and outputs. For proper functioning of those, both sides should be of the same size (choice,possibilities,digits) .See : D1.

Results :Xors with the help of wires connection function can make any program.

Simple watching people can act on them along with watchers will line.

As verifier influences verified.
Elections organizers should allow voters to verify their votes in terms of quantity and content.
What to keep in mind is that buying votes is more difficult then changing the end result.
Here is an example of implementation
The address of the voter determines the bureau where he or she is going to vote.
If he or she want to change the location of voting the ID showing the person address must be updated.
At voting location 3 powers must be present often are an executive worker a legislative worker and a judiciary worker .

First the executive : Verifies the ID and gives the last water bill of its address.

Second the legislative : Has 2 bags of numbers 2 copies of each number.

Each bag have numbers of distinct format.
One format is for IDs and the other is for water bills.
Picks 2 numbers from each bag writes the number of the address ID the number of the water
Bill on 2 copies he or she keeps and returns the new legislative numbers of ID and water bill with it's amount to the voter.

Third the judiciary :

Has 2 bags of numbers 2 copies of each number.
Each bag have numbers of distinct format.
One format is for IDs and the other is for water bills.
Picks 2 numbers from each bag Glues the legislative number of the address ID and the Legislative number of the water bill on 2 copies he or she keeps and returns the new judiciary
Numbers of ID and water bill with it's amount to the voter.
Voter votes in secret puts the 2 judiciary numbers on the envelope shows them to the judiciary worker and puts the envelope in ballot.
He or she can copy by hand the judiciary numbers before getting out.
The voter at voting location can verify the following :
The number of votes each power counted.
How many voted from her or his address seeing the judiciary number of the address.
The total of water bills in that bureau.
The total of bills in all bureaus.

Her or his vote and In case of a mistake the 3 powers can be gathered to fix it and that is for a limited delay until the Destruction of all links kept by powers between numbers.
The same technique of verification can be applied to the revenue of the nations or organization to verify the total of money.

Ideas for using this technology :
"Assignment optimization problem" solution grid has constraints:
In a line there can be only one 1, and same on a column.
Entering a 1 at a place would modify the feasible in a deterministic way to a new feasible.
It is not the case for entering 0.
Having a grid and entering the assignment data from small to big from a column of weakest digits to the next if the encountered value is 1 may be optimal.
Entering all the data together and request triple xor doors to vote may also be an approach. This could be extended to other optimization problems.

3 functions can be made without transistors, simple wires instead.
The 3 functions {and,or,not} can make any function and there for a computer.
(note : this was the first approximation and sharply require a repeater after several layers of doors unlike the previously demonstrated)
See: D2.

Let : L be a luminosity.

0 be the set of even multiples of L {0L, 2L, 4L, 6L, …}

1 be the set of odd multiples of L {1L, 3L, 5L, 7L, …}

Then: 0 mixed with 0 would give 0

0 mixed with 1 would give 1

1 mixed with 0 would give 1

1 mixed with 1 would give 0

Witch is an xor that can run at the speed of photons or electrons. (light verification)

See: D3.

It is a bit hard to make 2 distinct inputs fall in different decays while counting.(See: table above)

So a solution is to do that at inputs and here is an example of that (simplified):

See: D4.

Claims:

What I claim is :

1) Constraining to Verification as an approach to capitalize effort, optimize and simplify. Remarque : this claim do not extend to capitalizing effort, optimizing or simplifying with other methods.

2) The xors door made with 3 xor doors.

3) That elections organizers allowing people verifying their secret voting results instead of asking them to believe trusted people. Remarque : This claim is with will in the opposite direction then making it impossible for some to buy. Remarque : “Secret voting” is making it a bit difficult to reveal a person voting decision to another without consent.

4) Public allowed to verify state country or organization revenue while keeping source secret, protecting privacy.

5) D2 : The wires circuitry showing how to make an approximation of {and,or} doors. Remarque : a lamp could be replaced by a resistance.

6) D3 : 3 wires making an xor door (the idea is more an idea of a set encoding to make a logical door with wires).

7) D4 : A mechanical system demonstrating how to format the inputs correctly for xor "processors or circuitry". The idea is more of encoding many inputs properly for xors since input. For example classical circuitry using transistors could also be used.

Abstract of the disclosure.

Verification constraint as a mean of simplification and optimization.
Verification Function simplification to wires.
Verifiability as a state corruption reduction.
Majority vote function, approximation function or data formatting function, as a supplement to verification function, allowing access to universal logic possibilities.