Quantum Braitenberg Vehicles

Report

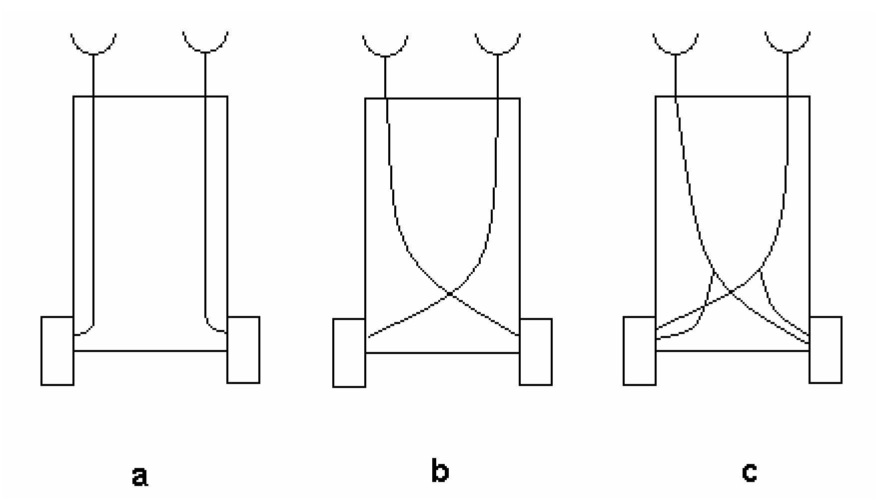
Submitted by
Indudhar Devanath

Braitenberg Vehicles

Valentino Braitenberg wrote a revolutionary book called "Vehicles, experiments in synthetic psychology" (Publisher: Cambridge, Mass. MIT Press). In his book he describes a series of thought experiments. In these experiments, he shows how simple systems (the vehicles) can display complex life-like behaviors far beyond those which would be expected from the simple structure of their 'brains'. He describes the a law called the "law of uphill analysis and downhill invention". Here he explains that it is far easier to create machines that exhibit complex behavior than it is to try and build the structures from the behavioral observations. By connecting simple motors to sensors, crossing wires and making some of them inhibitory, we can construct simple robots that could show fear, aggression, love, affection, and other feelings.

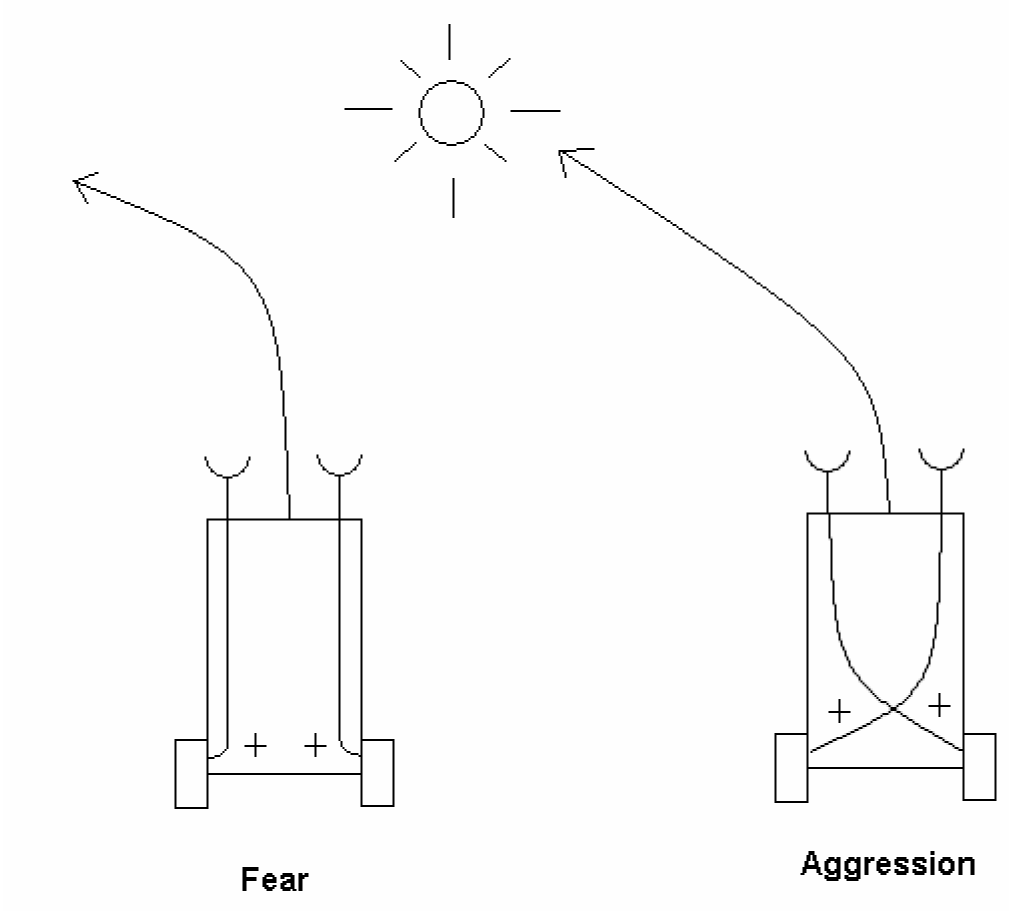
Love, Fear and Aggression

The vehicle has two sensors and two motors, right and left. The vehicle can be controlled by the way the sensors are connected to the motors. Braitenberg defines three different basic ways we could possibly connect the two sensors to the two motors.



a) Each sensor connected to the motor on the same side.
b) Each sensor connected to the motor on the opposite side.
c) Both sensors connected to both the motors.

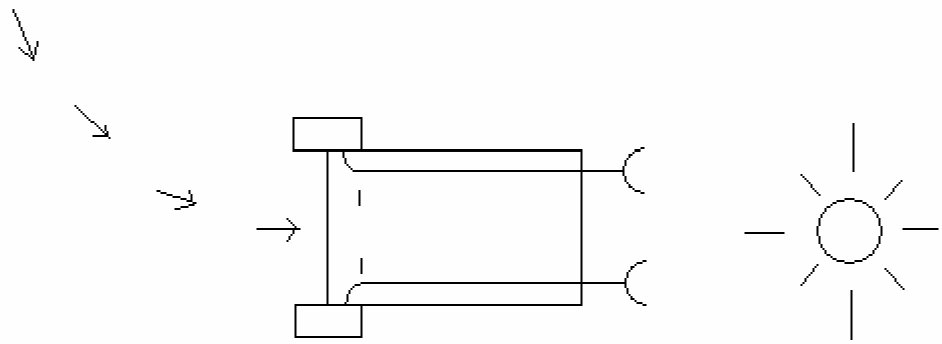
Type (a) vehicle will spend more time in places where there is less of the stuff that excites its sensors and will speed up when it is exposed to higher concentrations. If the source of the light (for light sensors) is directly ahead, the vehicle may hit the source unless it is deflected from its course. If the source is to one side, one of the sensors, the one nearer to the source, is excited more than the other. The corresponding motor turns faster. As a consequence, the vehicle will turn away from the source. Turning away from the source is illustrated with the following figure.



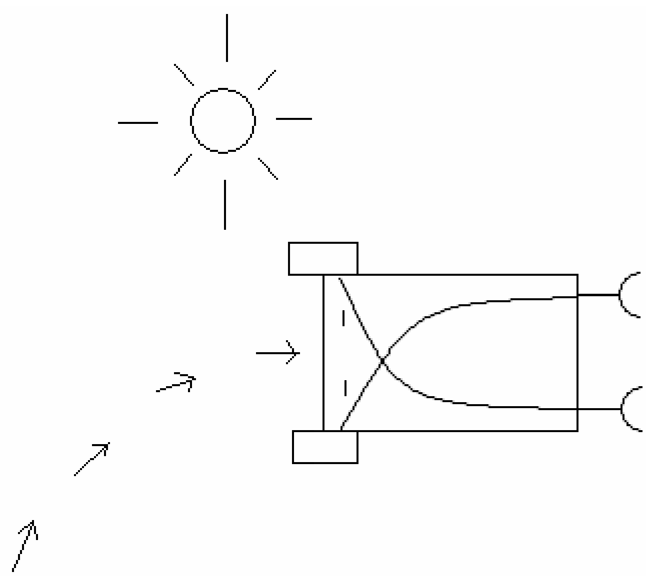
We can observe another type of vehicle, type (b) vehicle with positive motor connection. No change if the light is straight ahead, a similar reaction as seen in type (a). If it is to a side, then we observe the change. Here, the vehicle will turn towards the source and eventually hit it. As long as the vehicle stays in the vicinity of the source, no matten how it stumbles and hesitates, it will hit the source frontally, in the end.

If the two vehicles are let loose in an environment with sufficient stimulus sources, then their characters emerge. Their characters are quite opposite. The type (a) with positive connection will become restless in their vicinity and tends to avoid them, escaping until it safely reaches a place where the influence of the source is scarcely felt. The feelings of fear displayed by this vehicle. Vehicle of type (b) with positive connection turns towards the source of light. They resolutely turns towards them and hits the source with high velocity, as if it wanted to destroy them. The aggressive feelings displayed clearly.

When we introduce some kind of inhibition to the stimulation, we observe a slightly differing behavior but very interesting behavior. It is some what relaxing and soothing type of trend in the behavior is observed.

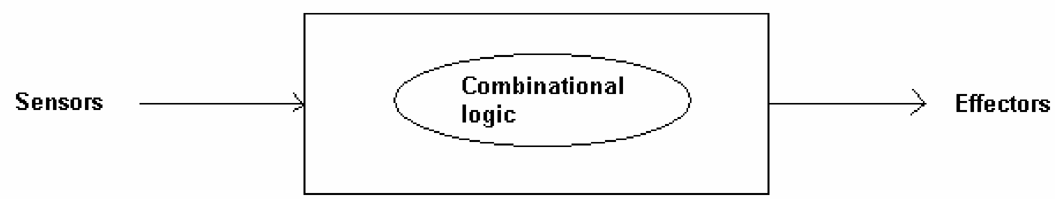


In the above example, we notice that when we switch the sensors excitation to the motors from positive excitation to negative excitation, we notice the following behavior. The negative excitation slows down the motor when the particular sensation is activated. The vehicle will spend more time in the vicinity of the source. The vehicle will orient itself towards the source and then approaches the source slowly, since the oblique course the sensor nearer to the source will slow down the motor on the same side, producing a turn toward that side. The vehicle with straight connections will come to rest facing the source. The vehicle with crossed connections for analogous reasons will come to rest facing away from the source and may not stay there very long, since a slight perturbation could cause it to drift away from the source. This would lessen the source's inhibitor influence, causing the vehicle to speed up more and more as it gets away. This behavior is illustrated in the below diagram.



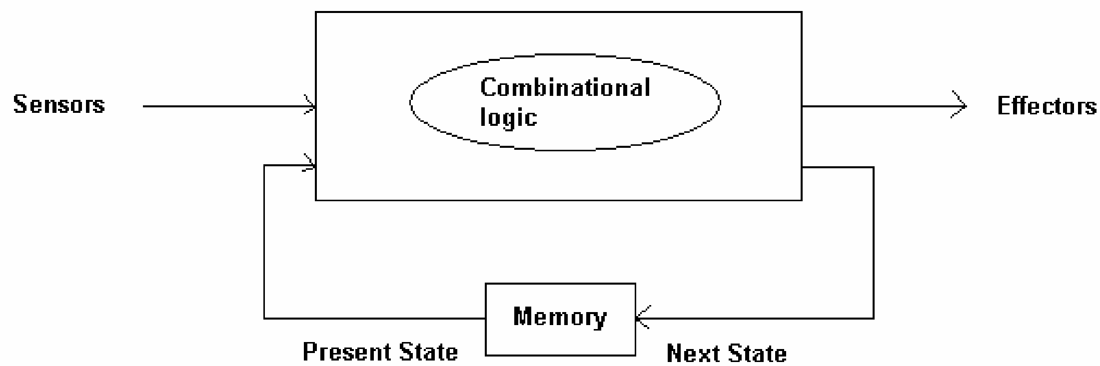
Building Braitenberg Vehicles

Type 1 Braitenberg Vehicles:



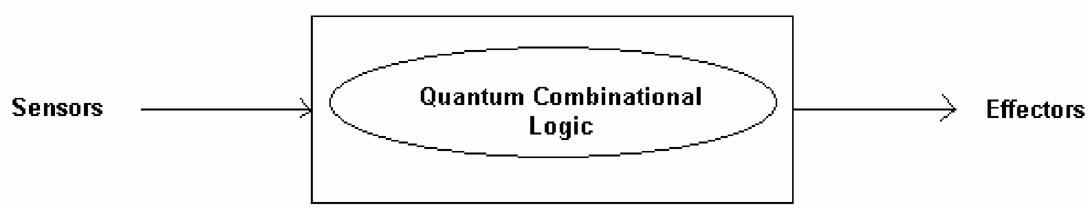
The sensors generate input to the combinational logic circuit and the output of the circuit is used to trigger the effectors of the Braitenberg Vehicle.

Type 2 Braitenberg Vehicle:



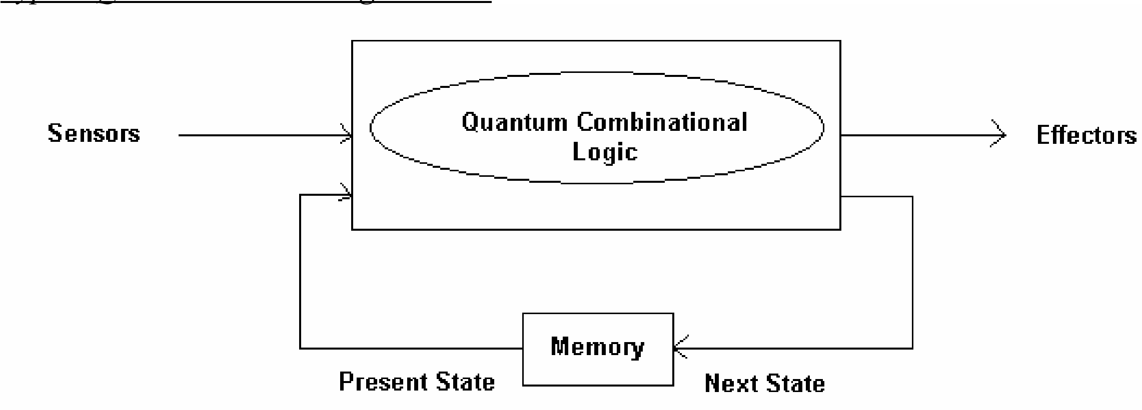
The combinational logic circuit will use sensors and present state from RAM and generate output and next state to control the effectors.

Type 3 Quantum Braitenberg Vehicle:



The sensors help generate input to the Quantum combinational logic and the output generates discrete data that help control the effectors.

Type 4 Quantum Braitenberg Vehicle:



Sensors help generate input in the form of discrete data. The logic also borrows present state input from the memory. Output generated is in the form of discrete data that will be used to trigger the effectors of the vehicles.

To start with, we will be using light sensors and servo motors that control the wheels of the robot will be our target effectors. The motion of the motors will define the behavior of each type of the vehicle. The combinational logic and Quantum logic will be designed using VHDL and will be implemented using an available XILINX FPGA chip. This chip will be mounted on the robot and will be tested for its behaviour.

Quantum Logic:

Qubits are quantum bits, derived from photons, electrons or ions. Electrons with 2 possible spin rotation +1/2 and -1/2 are represented as |01> and |1> respectively.
Wavefunction of a particle p_1 is given by $\psi = \alpha\ |0\rangle + \beta\ |1\rangle$
α and β are complex eigen values.
$|\alpha|^2$ = probability of p_1 in state |0>
$|\beta|^2$ = Probability of p_1 in state |1>

The properties of these probability are

a) $|\alpha|^2$ and $|\beta|^2$ have nonzero positive values
b) $|\alpha|^2 + |\beta|^2 = 1$
c) Particle p_1 and ψ_1 addedd to p_2 with ψ_2 we have
   $|\psi_1\psi_2\rangle = \alpha_1\alpha_2\ |00\rangle + \alpha_1\beta_2\ |01\rangle + \beta_1\alpha_2\ |10\rangle + \beta_1\beta_2\ |11\rangle$

In a quantum system n qubits represent a superposition of 2^n states. Operations over a set of qubits are defined as matrix operations. Quantum gate will be a matrix having vector of complex coefficients of the waveform as input and producing vector of complex coefficients as output.

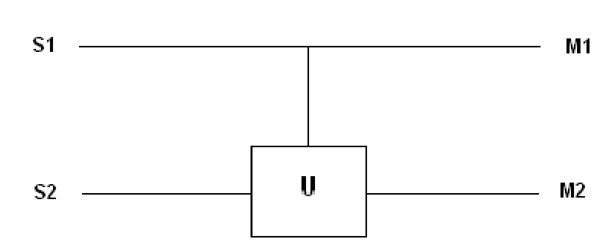
Example of a quantum gate

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} \alpha |0> \\ \beta |1> \end{bmatrix} = \begin{bmatrix} a\,\alpha |0> + b\,\beta |1> \\ c\,\alpha |0> + d\,\beta |1> \end{bmatrix}$$

a,b,c,d = Complex coefficients of the matrix indicating complex probability to transit from one state to another.

α |0> , β|1> = complex waveform coefficients to be propagated through the matrix operator.

General Purpose Controller Gate



if S1 = 0 then M2 = S2
if S1 = 1 then M2 = U (S2)

Here 'U' is the Quantum Logic that will be designed and implemented for our Quantum Braitenberg Vehicles.

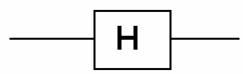
Fundamentals of Quantum Logic Gates

Quantum gates in parallel with another Quantum Gate will increase the dimensions of the quantum logic system which is represented in the matrix form. This is because the mathematical Kronecker product of Matrices is applied to the system. This Kronecker Matrix Multiplication is the one responsible for Qubit states to grow such that N bits corresponds to superposition of 2^N States where as in other digital systems N bits corresponds to 2^N distinct states.

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \otimes \begin{bmatrix} x & y \\ z & v \end{bmatrix} = \begin{bmatrix} a\begin{bmatrix} x & y \\ z & v \end{bmatrix} & b\begin{bmatrix} x & y \\ z & v \end{bmatrix} \\ c\begin{bmatrix} x & y \\ z & v \end{bmatrix} & d\begin{bmatrix} x & y \\ z & v \end{bmatrix} \end{bmatrix} = \begin{bmatrix} ax & ay & bx & by \\ az & av & bz & bv \\ cx & cy & dx & dy \\ cz & cv & dz & dv \end{bmatrix}$$

Kronecker Matrix Product

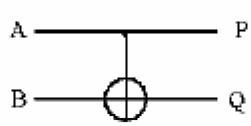
Quantum gate in series of another quantum gate will retain the dimensions of the quantum logic system.



$$= \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Hadamard Gate

Hadamard gate has a unitary matrix. Example of unitary matrix and also a permutation matrix is a Feynman gate. Permutation matrix is a matrix is a matrix which has only one ‘1’ in every row or column.

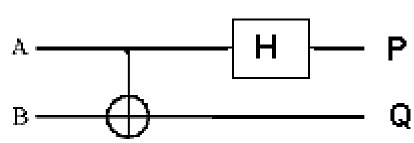


$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

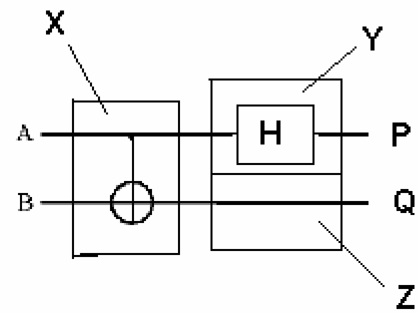
Feynman Gate

<u>Analyzing Quantum Logic Circuits</u>

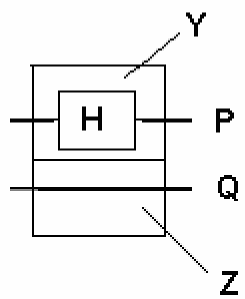
Example 1:



The above quantum circuit can be split into 3 circuits as shown below.



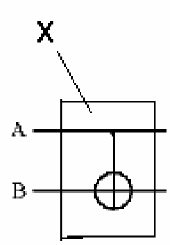
Here gate X (Feynman gate) is in series with gates H (Hadamard gate) and Z (Wire) which are themselves in parallel.



$$\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix} \parallel \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$$

$$\begin{bmatrix} 1\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} & 1\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} \\ 1\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} & -1\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} \end{bmatrix} = \begin{bmatrix}1 & 0 & 1 & 0\\ 0 & 1 & 0 & 1\\ 1 & 0 & -1 & 0\\ 0 & 1 & 0 & -1\end{bmatrix}$$

Kronecker Product Rule **M1**

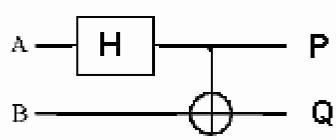


$$= \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{bmatrix}$$

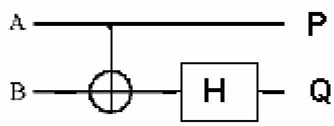
M2

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{M2} \times \underbrace{\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix}}_{M1} = \underbrace{\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & -1 \\ 1 & 0 & -1 & 0 \end{bmatrix}}_{M3}$$

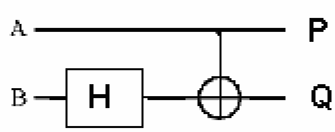
From this result we note that if the input is ‘00’ the output will be either ‘00’ or ‘10’. If the output is connected to servo motors then the vehicle would move either backwards or towards right. Similarly if the gates are re arranged as follows, the results are seen accordingly.



$$= \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 \end{bmatrix}$$

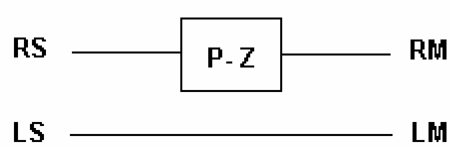


$$= \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$



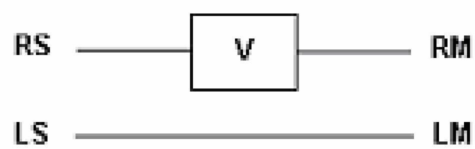
$$= \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

Pauli-Z gate example



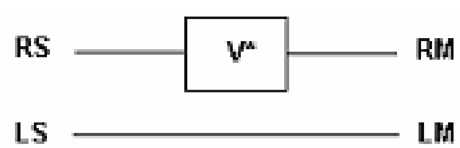
$$\underset{\text{Pauli - Z}}{\begin{bmatrix} 1 & 1 \\ 0 & -1 \end{bmatrix}} \text{ Kronecker Product } \underset{\text{Unitary}}{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

V Gate (Root of Not Gate) Example



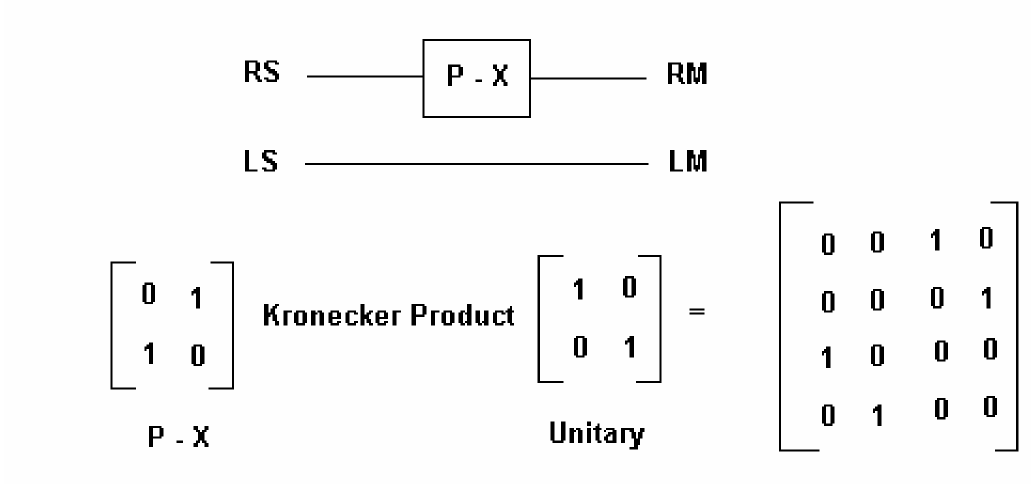
$$\frac{1+j}{2}\underset{\text{Pauli - Z}}{\begin{bmatrix} 1 & --j \\ --j & 1 \end{bmatrix}} \text{ Kronecker Product } \underset{\text{Unitary}}{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}} = \frac{1+j}{2}\begin{bmatrix} 1 & 0 & --j & 0 \\ 0 & 1 & 0 & --j \\ --j & 0 & 1 & 0 \\ 0 & --j & 0 & 1 \end{bmatrix}$$

V* Gate (Inverse of Root of Not Gate) example:

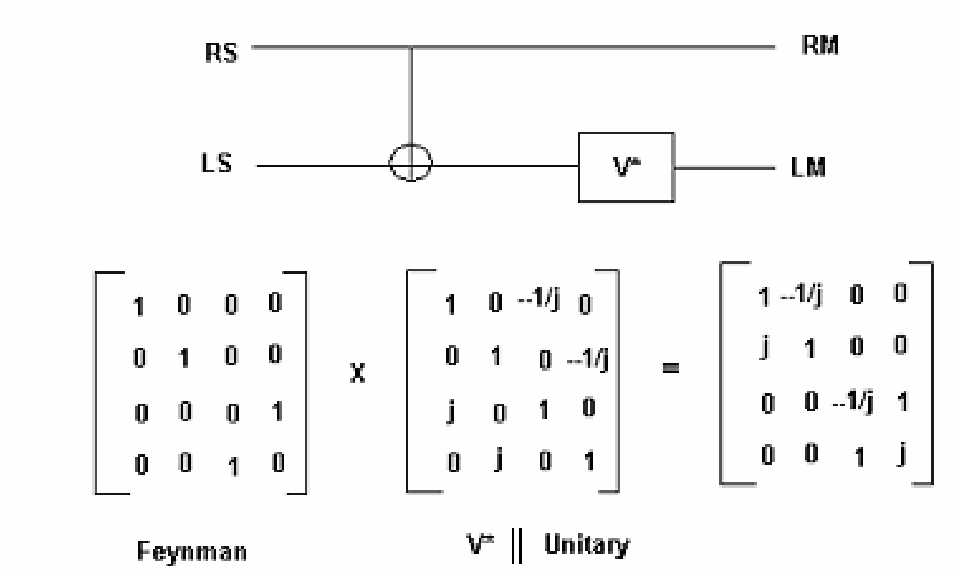


$$\frac{1}{1+j}\underset{\text{V*}}{\begin{bmatrix} 1 & --1/j \\ j & 1 \end{bmatrix}} \text{ Kronecker Product } \underset{\text{Unitary}}{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}} = \frac{1}{1+j}\begin{bmatrix} 1 & 0 & --1/j & 0 \\ 0 & 1 & 0 & --1/j \\ j & 0 & 1 & 0 \\ 0 & j & 0 & 1 \end{bmatrix}$$

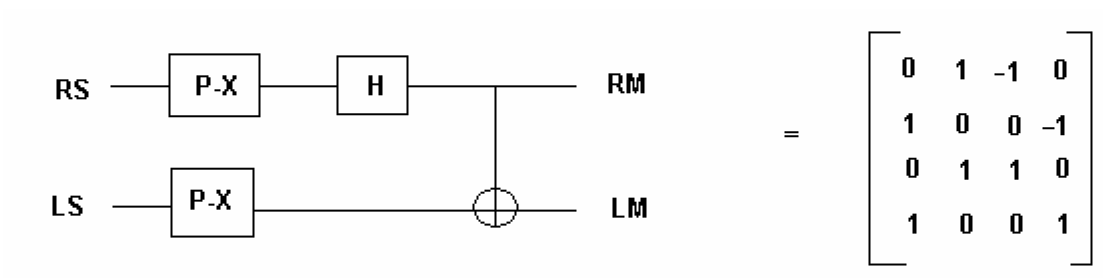
Pauli-X gate example



Combination of Feynman and V* Gate

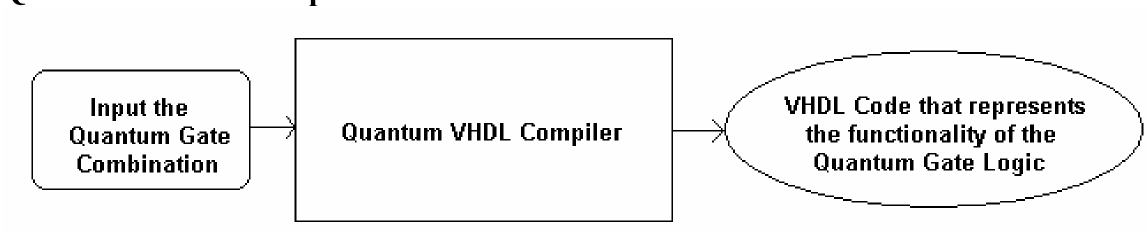


Combination of Hadamard, Pauli-X and Feynman gate



By using these combination of gates, we could build a Quantum logic that we can use to implement on the Braitenberg Vehicle. After the choice of the gates, then it is input into a compiler which then converts the Quantum Logic into VHDL code that can be directly downloaded into the FPGA style circuits to generate its equivalent Binary Logic which behaves exactly like that of the designed Quantum Logic.

Quantum VHDL Compiler



This compiler is one of a kind which can generate a VHDL code for any kind of the Quantum circuit that is given to the compiler. The compiler is build using LISP programming. Lisp is a general-purpose programming language and an AI language interactive Common Lisp programs are easy to test (interactive) easy to maintain (depending on programming style) portable across hardware/OS platforms and implementations (there is a standard for the language and the library functions) Common Lisp provides clear syntax, carefully designed semantics, several data types (numbers, strings, arrays, lists, characters, symbols, structures, streams etc.) runtime typing (the programmer need not bother about type declarations, but he gets notified on type violations.), many generic functions (88 arithmetic functions for all kinds of numbers (integers, ratios, floating point numbers, complex numbers), 44 search/filter/sort functions for lists, arrays and strings automatic memory management (garbage collection) packaging of programs into modules an object system, generic functions with powerful method combination **(Common Lisp was the first ANSI standard object oriented programming language.)** and DSB macros (every programmer can make his own language extensions). Common Lisp is well suited to large programming projects and explorative programming.

The language has a dynamic semantics which distinguishes it from languages such as C and Ada. It features automatic memory management, an interactive incremental development environment, a module system, a large number of powerful data structures, a large standard library of useful functions, a sophisticated object system supporting multiple inheritance and generic functions, an exception system, user-defined types and a macro system which allows programmers to extend the language. Eric Raymond has written an essay called "How to Become a Hacker," in which he explains that Lisp is worth learning for the profound enlightenment experience you will have when you finally get it; that experience will make you a better programmer for the rest of your days, even if you never actually use Lisp itself a lot.

Uses of Building such amazing Quantum Braitenberg Vehicles

1) They can be used to solve maze games
2) They can be tested for their performance in the Robot Soccer Competitions
3) The Quantum circuits could be implemented into public entertainment robots
4) They can also be used to build commercially viable Robot Pets.

Reference:

1) **Martin Lukac, Mikhail Pivtoraiko, and Marek Perkowski,** ``Genetic Algorithms for Quantum and Reversible Logic Synthesis, accepted to *Artificial Intelligence Review Journal,* Special Issue on Artificial Intelligence in Logic Design, S. Yanushkevich guest editor, 2003.
2) **Goran Negovetic, Martin Lukac, Marek Perkowski, Andrzej Buller,** ``Evolving Quantum Circuits and an FPGA-based Quantum Computing Emulator," *Proceedings of Symposium on Boolean Problems.* September, 2002, Freiberg, Germany
3) **Valentino Braitenberg**, "*Vehicles, experiments in synthetic psychology*" Publisher: Cambridge, Mass. : MIT Press, 1984
4) Reference for **LISP Programming**:
    a. http://www.nist.gov/lispix/doc/lispix/lisp-new.htm
    b. http://www.cons.org/cmucl
    c. http://www-formal.stanford.edu/jmc/history/lisp/lisp.html