Electronic Model of Human Brain using Verilog

M.Krishna[1] M. umarani[2]
1 & 2: Assistant Professor ,Department of ECE,Geethanjali College of Engineering and Technology

Abstract:
Reversible logic has become an emerging field for research. The main advantage of reversible logic is power reduction and this advantage have drawn up a significant interest in this field. The aim of the paper is to realize the decoder using Fredkin gate which is basically a reversible gate. There are many reversible logic gates i.eFredkin gate, Feynman gate, Double Feynman gate, Peres gate, New gate, Toffoli gate and many more. In the reversible logic, reversibility have a special condition which is reversible computing and it is based on the principle of bijection device with a same no of inputs and outputs which means one to one mapping. It finds its application in various fields including quantum computing, optical computing, nanotechnology, computer graphic, cryptography, digital signal processing and many more. Reversible logic is gaining importance in recent years largely due to its property of low power consumption. A comparative study in terms of the number of gates, number of constant inputs, number of garbage outputs and quantum costs is also presented. The circuit has been implemented and simulated in Xilinx.

Introduction

Two hardware description languages—Verilog and VHDL—have become established as the industry-standard starting point for large-scale digital logic synthesis [1]. But why two languages? Wouldn't one be enough? And most critically: How does the need to support more than one language affect your design flow? The answer to this riddle is critically important to every design engineer, verification engineer, and engineering manager. Come with us as we search to find the answer! Along the way we will explore the history of HDLs, examine the intricacies of your current design flows, and discuss our recommendations for the future.

In 1977, G. J. Lipovski gave this simple definition: "…a hardware description language is a variation of a programming language tuned to the overall needs of describing hardware [2]." In 1979, W. M. vanCleemput [3] listed the major applications of HDLs as:

• Description of the behavior and/or structure of a system as a means for accurately communicating design details between designers and end users.

•As the input to a system level simulator.

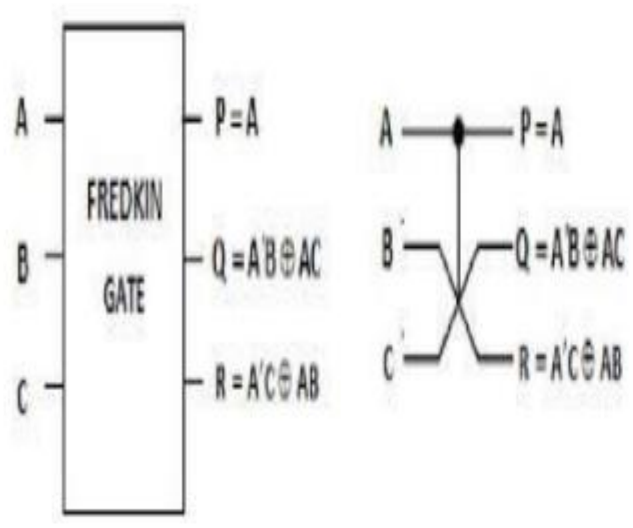
•As the input to an automatic hardware compiler.

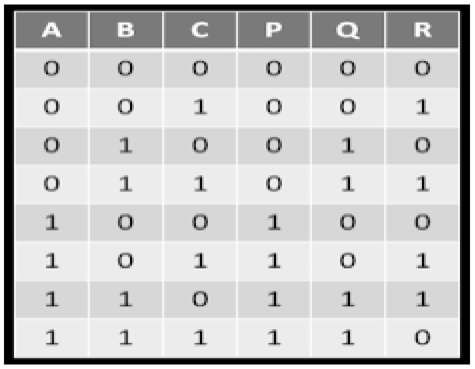
•As the input to a formal verification system. Amazingly, decades later these comments still hold true

In his 1940 master's thesis, Claude Shannon introduced a method for describing circuit behavior using Boolean algebra, manipulating these equations into their simplest form, and then synthesizing the corresponding relay switching circuits [4]. In a 1946 report, John von Neumann used a symbolic notation to describe the operations (i.e., instruction set) of the IAS computer [5]. Irving S. Reed in 1952 proved that Boolean algebraic equations can be physically realized as electronic circuits, and recommended "…in the initial synthesis [design] of a digital computer it is desirable to concentrate one's attention on the abstract model of the digital computer1 [6]." By 1956, Reed extended this work into what he called a “register transfer language” to describe the design of a digital computer [7]. This is arguably the first true HDL, as it included the concepts of timing and clocks. Research and development into hardware description languages continued through the 1960s. A 1974 survey listed over fifty HDLs from all over the world2 [8].

In this era of technology and advancement power consumption has become an important factor of consideration. In this paper we have reduced the power consumption of 4 to 16 decoder by using reversible logic. Reversible logic finds its application in quantum computing, nanotechnology, low power VLSI. In the irreversible logic for the loss of each bit of information [1] proposed that KTlog2 joules of energy is dissipated, where K is Boltzmann’s constant and T represents temperature. This amount of heat is very small in simple circuits but it becomes large in complex circuits. Bennett [2] describes that if all the computation is carried out in reversible manner, the power dissipation due to loss of bit can be avoided. Reversible logic involves the use of reversible gates which have same number of inputs and outputs and they can be made to run in backward direction also. Each input in the circuit is associated with some energy. If a bit is lost that is number of bits at the output are less as compared to the inputs ,then energy associated with the corresponding bit is dissipated in the form of heat. Since in reversible circuits no bit loss is there hence ideally in reversible circuits no power dissipation occurs. But practically some power dissipation do occur, which is much less than the conventional logic. The extra output used in order to make inputs and outputs equal are called garbage outputs. The circuit should be designed in such a manner so as to keep these outputs minimum. However in certain reversible circuits constant inputs are also used. These constant inputs are set either to logic 1 or logic 0 depending upon the operation of the circuit. Reversible gates differ from the conventional logic gates in terms of above two factors.

A. **Fredkin Gate:** Figure 1 shows the diagram of 3*3 Fredkin gate with A, B, C as inputs and P,Q,R as outputs . Here output “P‟ is the garbage output that is it is required only for the purpose of obtaining reversibility.





A	B	C	P	Q	R
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	0

Fig.1 3x3 FREDKIN GATE

It is an n-input n-output logic function in which there is a one-to-one correspondence between the inputs and the outputs. Because of this bijective mapping the input vector can be uniquely determined from the output vector [3]. That is inputs can be calculated with the help of outputs also. Whereas such a feature is not present in conventional B

B logic gates

In conventional logic gates we may have same output for more than one combination of inputs. Various reversible logic gates have been proposed till date, but few of them are described below

B. Feynman Gate: Let Iv and Ov be the input and output vector of a 2*2 Feynman gate (FG) [4,5] respectively, where Iv= (A,B) and Ov = (P=A, Q=A⨁B). The block diagram for 2*2 Feynman gate is shown in Fig.3.3. (Quantum Cost = 1)

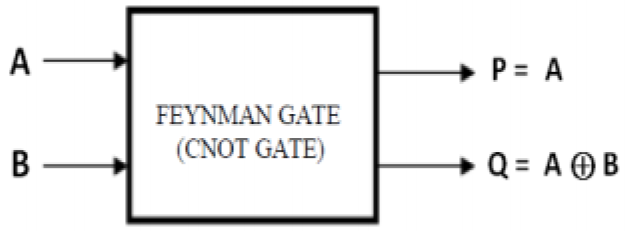


Fig.2 2x2 FEYNMAN GATE (CNOT GATE)

C. **Toffoli Gate:** Let Iv and Ov be the input and output vector of a 3*3 Toffoli Gate (TG) [6,7] respectively, where Iv =(A, B, C) and Ov=(P=A, Q=B, R=AB⨁ C). Fig.3.7 shows the 3*3 Toffoli gate. (Quantum Cost = 5)

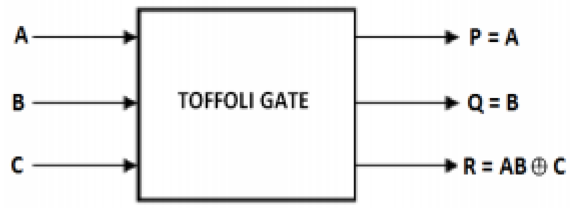


Fig.3 3x3 TOFFOLI GATE

D. **Peres Gate**: Let Iv and Ov be the input and output vector of a 3*3 Peres Gate [6,8,9] respectively, where Iv=(A,B,C) and Ov=(X=A,Y=A⨁B , Z=AB⨁C). Fig. 3.11 shows the block diagram of 3*3 Peres gate. (Qauntum Cost = 4)

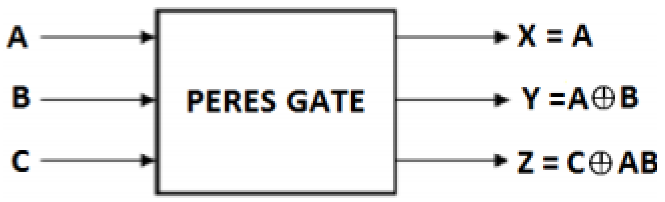


Fig.4 3x3 PERES GATE

Peres Gate is an important gate which has a low quantum cost as compared to other gates. A single Peres gate can give generate and propagate outputs when the third input C = 0. Two Peres gates can be combined to form a full adder.

E. **Double Feynman Gate:** Let Iv and Ov be the input and output vector of a 3*3 Double Feynman Gate respectively, where Iv = (A, B, C) and Ov = (P=A, Q=A⨁B, R=A⨁C). Fig.3.5 shows the 3*3 Double Feynman gate. (Quantum Cost = 2)

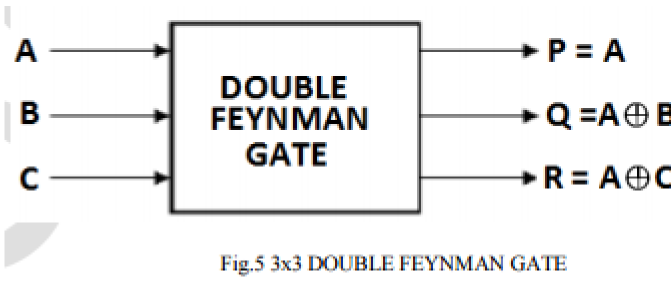


Fig.5 3x3 DOUBLE FEYNMAN GATE

Verilog was originally a proprietary verification/simulation product from Gateway Design Automation. Phil Moorby working with Chi-Lai Huang developed the language specification during the month of December 1983. Work on the logic simulator continued during 1984, and the first sale was in early 1985 [20]. Moorby's previous work with HILO-2 [21] was a major inspiration for the Verilog language, with influence also from C, Pascal, and especially occam [14]. Original goals for the language were to support logic simulation, fault simulation, and logic synthesis. The simulator supported min-max static and dynamic timing analysis [20]. By 1987, Moorby had written an even faster logic simulator called Verilog-XL, which boasted gatelevel simulation speeds approaching that of hardware accelerators. Verilog-XL was a landmark product in EDA, rapidly gaining market share in an industry where there were several competing digital simulators. In addition to greater simulation speed, Verilog-XL boasted greater design-size capacity than its competitors. It featured a single, integrated language for modeling both the design and the testbench, plus the ability for end users to extend the language using Verilog's Programming Language Interface (PLI). Furthermore, Gateway actively and successfully courted ASIC vendors to use Verilog-XL as their timing signoff "golden simulator." At the request of these endors, Gateway implemented several Verilog language enhancements to support accurate gatelevel timing, including pin-to-pin path delays and a standardized method for back-annotation [22]. Due to Verilog being a proprietary language, Gateway could easily add the features that ASIC vendors wanted, and do it quickly without the multi-year process of IEEE standardization. Although Verilog remained proprietary to Gateway, multiple companies licensed the language to use for ASIC cell simulation libraries as well as logic synthesis tools. In late 1989, Gateway was acquired by Cadence, which continued to sell Verilog-XL and other Verilog tools obtained from Gateway [27].

Language Wars in the Early 1990s

By the late 1980s the stage was set for what became known as the "language wars," pitting Verilog versus VHDL. VHDL became popular with many users and EDA tool vendors because it was an IEEE standard and thus could be used with no royalty cost.
simulator vendor used a different scheme [25]. Synopsys licensed the Verilog language from Gateway (later Cadence), and in 1988 introduced its Logic Compiler synthesis tool (soon renamed Design Compiler) [26]. By 1989, eight ASIC vendors supported Design Compiler. By the summer of 1991, 27 ASIC vendors supported the Synopsys synthesis tool, with 20 using Design Compiler at their own design centers [27]. Despite its early use of Verilog, Synopsys became a strong advocate for VHDL, perhaps because they offered a VHDL simulator product but no Verilog simulator. In 1990, Cadence released the Verilog language to a newly formed nonprofit organization called Open Verilog International (OVI). The language definition entered the public domain and became available to any vendor [22]. Starting in 1992, OVI began sponsoring the annual International Verilog Conference (IVC).

CONCLUSION

Also US military suppliers were required to document their designs using VHDL. However, early VHDL tools revealed inconsistencies and ambiguities in the 1987 standard, and in 1991 the IEEE was forced to issue an unusual "Standards Interpretation" document [23]. Different vendors had dissimilar interpretations of the standard, and this became a source of frustration for early VHDL users [19]. In 1988 the 1st VHDL Users Group meeting was held as a "birds-of-a-feather" session at DAC (Design Automation Conference). In the fall of 1988, an independent VHDL Users Group meeting was held, and semiannual meetings continued afterwards.

In 1991, the organization VHDL International (VI) was founded and the conference was renamed VIUF (VHDL International Users Forum) [24]. Meanwhile, Verilog continued to gain popularity with ASIC vendors who settled on Verilog-XL as their "golden simulator" for timing verification (i.e., signoff). Verilog-XL offered a standard timing back-annotation procedure, whereas
VHDL specified no such standard, thus each VHDL

In this paper, we have presented a technique to design Verilog code for 4 to 16 reversible decoder using Fredkin gate. We have seen that there is a power reduction in the circuit when it is implemented in the reversible logic. Some other reversible gates can be used in design process of reversible decoder apart from fredkin gate. However the circuit consists of 4 inputs,15 constant inputs,10 garbage outputs, 16 outputs. Hence the power dissipation is compared with the irreversible logic gates
.

REFERENCES

[1]. R.Landauer, "Irreversibility and Heat Generation in the Computational Process", IBM Journal of Research and Development, 5,pp. 183-191,1961.
[2]. C.H.Bennett, "Logical Reversibility of Computation", IBM J.Research and Development, pp.525-532, November1973.

[3]. Nagamani A N, Jayashree H V,H R Bhagyalakshmi,” Novel Low Power Comparator Design using Reversible Logic Gates” Vol. 2 No. 4 Aug -Sep 2011.

[4]. Milburn, Gerard.j., The Feynman processor perseus books 1998 [

5]. Feynman R., 1985. Quantum mechanical computers, Optics News, 11: 11-20.

[6]. E. fredkin, T. Toffoli, “Conservative Logic”, International Journal of Theory of Physics,21, 1982, pp 219-253

[7]. Toffoli T., 1980. Reversible computing, Tech Memo MIT/LCS/TM-151, MIT Lab for Computer Science.

[8]. P.D. Picton, “ Fredkin gates as the basic for comparison of different logic designs”, Synthesis and optimization of logic system, London, VK, 1994.

[9]. Peres, A. 1985. Reversible logic and quantum computers. Physical Review A, 32: 3266-3276