

## **Implementation Modified Counters using Fredkin & Feynman Logic Gates**

Hawrra H.Abbas Al-Rubiae  
Computer engineering. The University of Kerbala

### **Abstract :**

In recent years, reversible logic has emerged as a promising computing paradigm having application in low power CMOS, quantum computing, nanotechnology, and optical computing. The classical set of gates such as AND, OR, and EXOR are not reversible. In this paper, different type of counters designed using reversible Fredkin and Feynman gates (tested using electronic workbench v5.12). It is demonstrated that the counter architectures designed using the proposed gate are much better and optimized, compared to their existing counterparts in literature; in terms of number of reversible gates and garbage outputs. Thus, this paper provides the initial threshold to building of more complex system which can execute more complicated operations using reversible logic.

Keywords— Reversible Logic, counters, logic circuits.

### **1. INTRODUCTION**

This section provides an effective background of reversible logic with its definition and the motivation behind it.

#### **1-1. Definitions**

Researchers like Landauer have shown that for irreversible logic computations, each bit of information lost generates  $kT \ln 2$  joules of heat energy, where  $k$  is Boltzmann's constant and  $T$  the absolute temperature at which computation is performed [1]. Bennett showed that  $kT \ln 2$  energy dissipation would not occur, if a computation is carried out in a reversible way [2], since the amount of energy dissipated in a system bears a direct relationship to the number of bits erased during computation. Furthermore, voltage-coded logic signals have energy of  $E_{sig} = \frac{1}{2}CV^2$ , and this energy gets dissipated whenever switching occurs in conventional (irreversible) logic implemented in modern CMOS technology. It has been shown that reversible logic helps in saving this energy using charge recovery process [6]. Reversible circuits are those circuits that do not lose information. Reversible computation in a system can be performed only when the system comprises of reversible gates. These circuits can generate unique output vector from each input vector, and vice versa, that is, there is a one-to-one mapping between input and output vectors. Thus, an  $N \times N$  reversible gate can be represented as

$$I_v = (I_1, I_2, I_3, I_4, \dots, I_N)$$

$$O_v = (O_1, O_2, O_3, \dots, O_N)$$

Where  $I_v$  and  $O_v$  represent the input and output vectors respectively. Classical logic gates are irreversible since input vector states cannot be uniquely reconstructed from the output vector states. There are a number of existing reversible gates such as Fredkin gate [3,4,5], Toffoli Gate (TG) [3, 4] and the New Gate (NG) [6]. As the Moore's law continues to hold, the processing power doubles every 18 months. The current irreversible technologies will dissipate a lot of heat and can reduce the life of the circuit. The reversible logic operations do not erase (lose) reversible logic is likely to be in demand in high speed power aware circuits. Reversible circuits are of high interest in low-power CMOS design, optical computing, nanotechnology and quantum computing.

The most prominent application of reversible logic lies in quantum computers [13]. A quantum computer will be viewed as a quantum network (or a family of quantum networks) composed of quantum logic gates; each gate performing an elementary unitary operation on one, two or more two-state quantum systems called qubits. Each qubit represents an elementary unit of information; corresponding to the classical bit values 0 and 1. Any unitary operation is reversible and hence quantum networks effecting elementary arithmetic operations such as addition, multiplication and exponentiation cannot be directly deduced from their classical Boolean counterparts (classical logic gates such as AND or OR are clearly irreversible). Thus, quantum arithmetic must be built from reversible logical components [13]. The synthesis of reversible logic differs significantly from traditional irreversible logic synthesis approaches as fan-outs are not permitted in the reversible logic synthesis.

Outputs from one gate are used as inputs to the next gate without fan-out of more than one. This results in a high degree of interdependence among gates. Furthermore, reversible logic synthesis of sequential logic differs from combinational logic in that the output of the logic device is dependent not only on the present inputs to the device, but also on past inputs; i.e., the output of a sequential logic

device depends on its present internal state and the present inputs. The design of complex system will require sequential circuits based on Flip Flops. This paper provides the initial threshold to building of more complex system having sequential circuits and which can execute more complicated operations using quantum computers. The reversible circuits form the basic building block of quantum computers as all quantum operations are reversible. The important reversible gates used for reversible logic synthesis are Feynman Gate, New Gate and Fredkin gate.

**1-2. BASIC REVERSIBLE GATES**

There are a number of existing reversible gates in literature. Fredkin [3, 4, 5] and Feynman gates [3] are used to construct the reversible PLA.

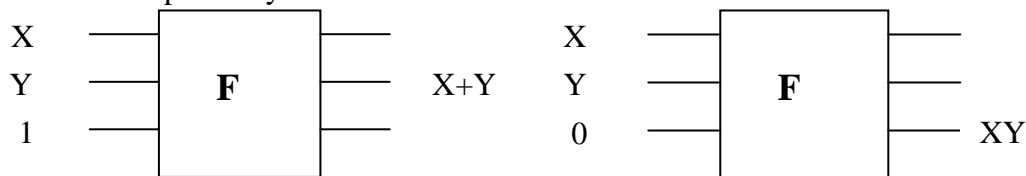
Fredkin gate [3, 4, 5], is a (3\*3) conservative reversible gate originally introduced by Petri [4,5]. It is called 3\*3 gate because it has three inputs and three outputs. The input triple (x1, x2, x3) associates with its output triple (y1, y2, y3) as follow

$$y1 = x1$$

$$y2 = (-x1 \wedge x2) \vee (x1 \wedge x3)$$

$$y3 = (x1 \wedge x2) \vee (-x1 \wedge x3)$$

Figure 1.a and Figure 1.b show the implementation of the Fredkin gate as OR and AND functions respectively.

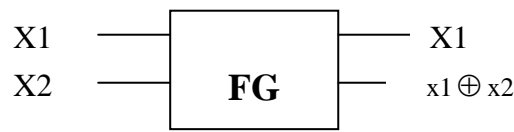


**Figure 1. Fredkin Gate as (a) OR Function (b) AND Function**

Feynman gate [3] is a 2\*2 one-through reversible gate shown in Figure 2. It is called 2\*2 gate because it has 2 inputs and 2 outputs. One through gate means that one input variable is also the output. The input double (x1,x2) associates with its output double (y1,y2) as follows.

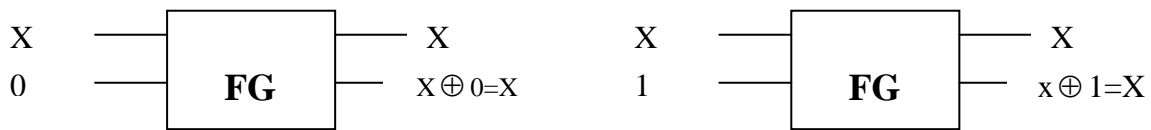
$$y1 = x1$$

$$y2 = x1 \oplus x2$$



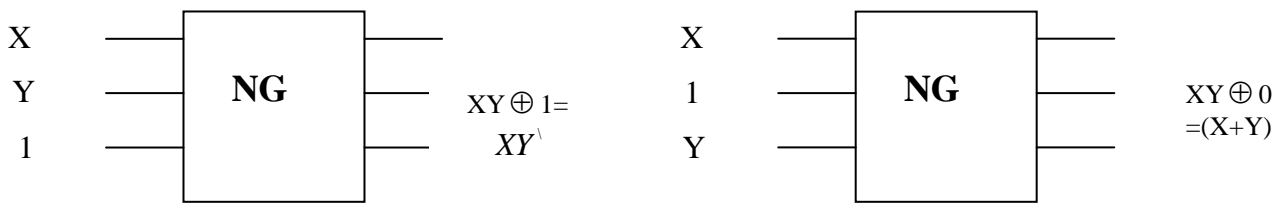
**Figure 2. Feynman Gate**

Figure 3a shows the implementation of Feynman gate for copying the input and Figure 3b shows the implementation of it for generating the complement of the input.



**Figure 3. Feynman gate as (a) Copier (b) Complementer**

To design the NAND and NOR function using New Gate as shown in figure (4a) and figure (4b) respectively.



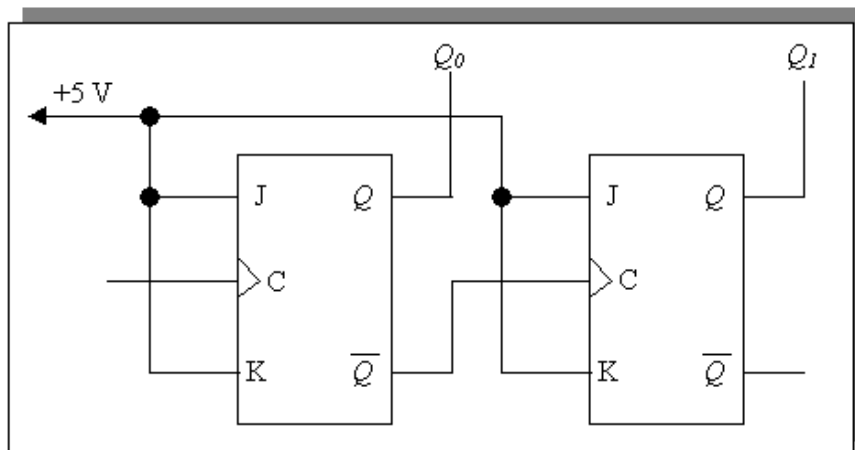
**Figure 4. New gate as (a) NAND gate (b) NOR gate**

**2. Counters theory**

An important application of flip-flops is in the design digital counters. These devices generate binary numbers in a specified count sequence when triggered by an incoming clock waveform. On each trigger, the counter advances to the next number in the sequence. After reaching the final state in the sequence, the counter then recycles. Counters may be used to count up or down, to cycle through memory addresses in microprocessors applications, to generate waveforms of particular patterns and frequencies, and to activate other logic circuits in a complex process [7].

**2-1. Asynchronous counter**

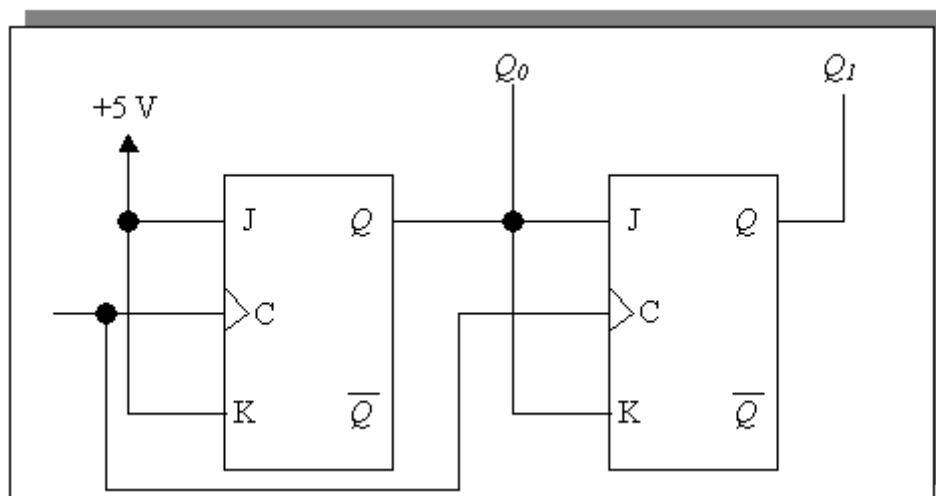
Asynchronous counters do not have a common clock that controls all the flip-flop stages. The control clock is input to the first stage, or the LSB stage of the counter. The clock for each stage subsequent is obtained from the flip-flop from the prior stage. The 2-bit counter shown in Figure (5) [7,8].



**Figure (5) A two-bit asynchronous counter**

**2-2. Synchronous counters**

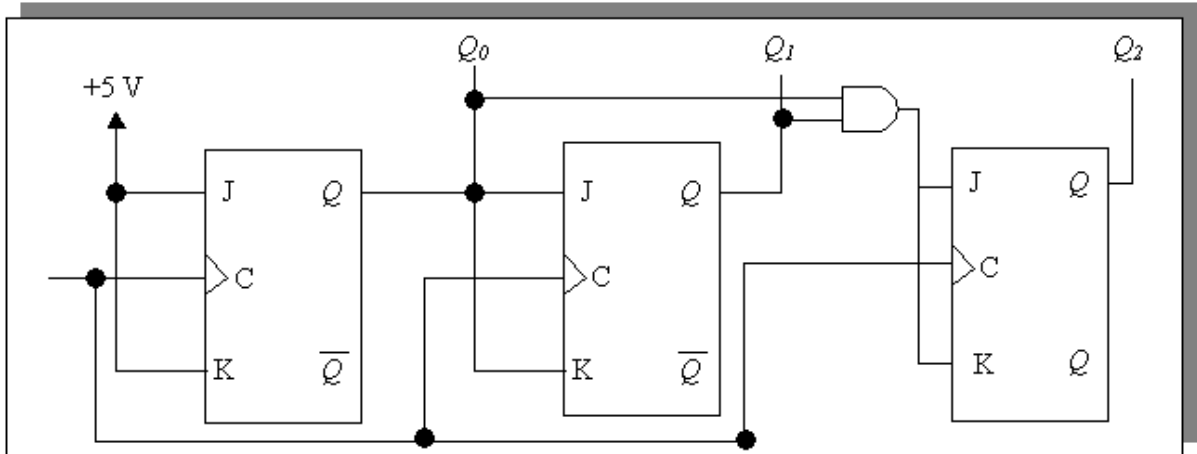
1.Synchronous digital counters have a common clock which results in all the flip-flops being triggered simultaneously. Consequently, there are no cumulative delays that result because the clock signal must ripple through the stages as in the asynchronous counters. Synchronous counters can be designed to count up and down in numerical order. In addition, they may be used to produce count sequences of non-consecutive numbers. The count sequence produced by synchronous counters is not dependent on the trigger characteristics of the flip-flops that comprise the count stages. The count sequence is achieved by applying the required logic function into the flip-flops figure (6) [7,8,9].



**Figure (6) A two-bit synchronous counter**

Figure (7) shows A three-bit synchronous counter where

- $J_2 = Q_1 Q_0$
- $K_2 = Q_1 Q_0$
- $J_1 = Q_0$
- $K_1 = Q_0$
- $J_0 = 1$
- $K_0 = 1$

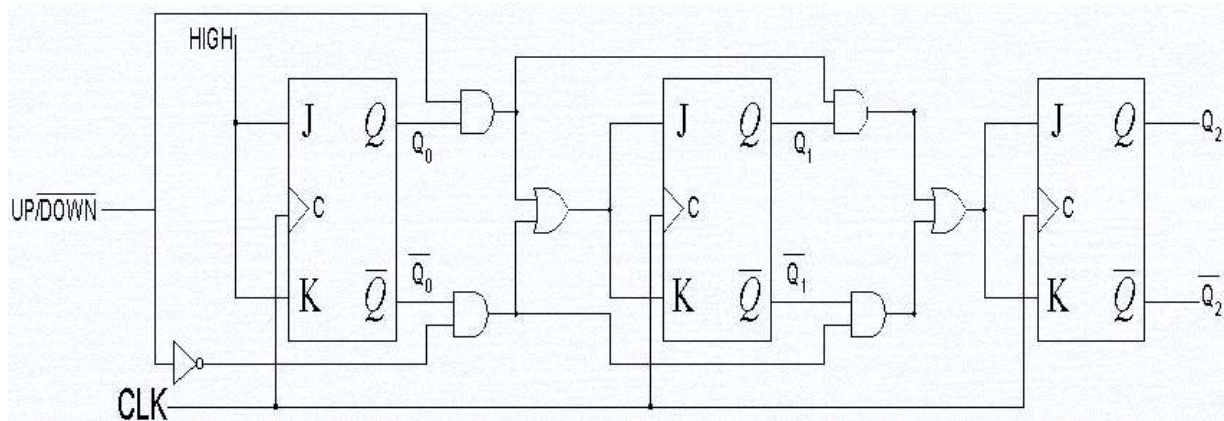


**Figure (7) A three-bit synchronous counter**

**2-3. Bidirectional counters**

Bidirectional counters, also referred to as UP/DOWN counters, are capable of progressing in either direction through any given count sequence. Recall that in general, bidirectional counters can be reversed at any point in their count sequence. Figure (8) 3-bit binary bidirectional counter [7,8,9].

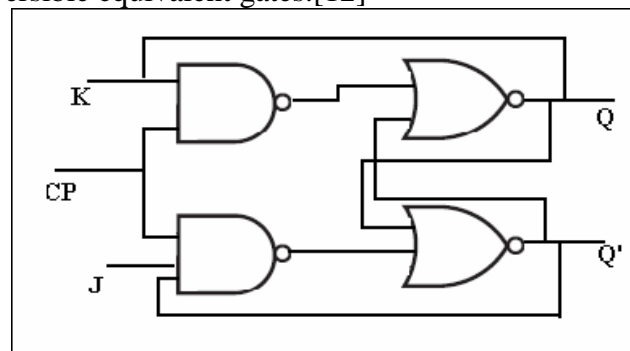
$\bar{Y} = DOWN \quad \& \quad Y = UP$



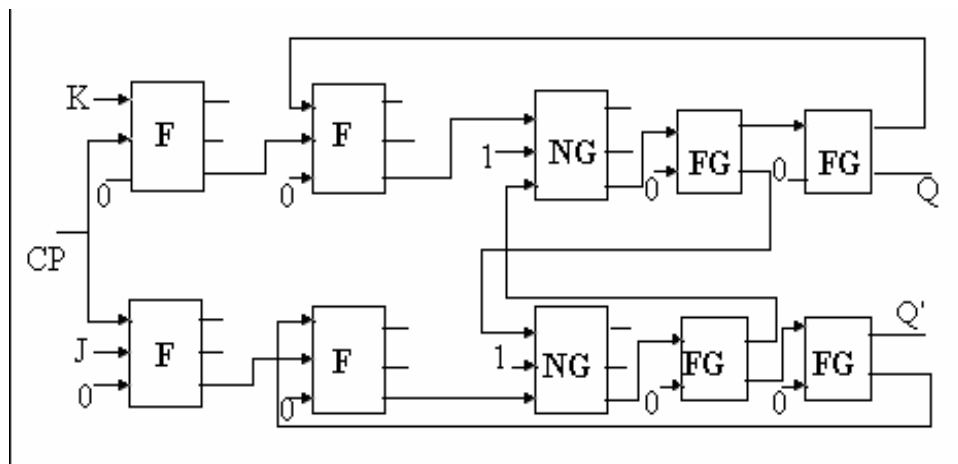
**Figure (8) Bidirectional counters**

**3.Proposed Reversible J-K flip flop using Fredkin Gate**

A JK flip flop can be considered as a refinement of the RS flip flop since the indeterminate state of the RS type is defined in the JK type. The JK flip flop switches to its complement state, when inputs are applied to both J and K simultaneously. Figure (9) shows the JK flip flop designed from conventional irreversible gates. Figure (10 ) shows the JK flip flop designed from the reversible equivalent gates.[12]



**Figure (9) J-K flip flop**

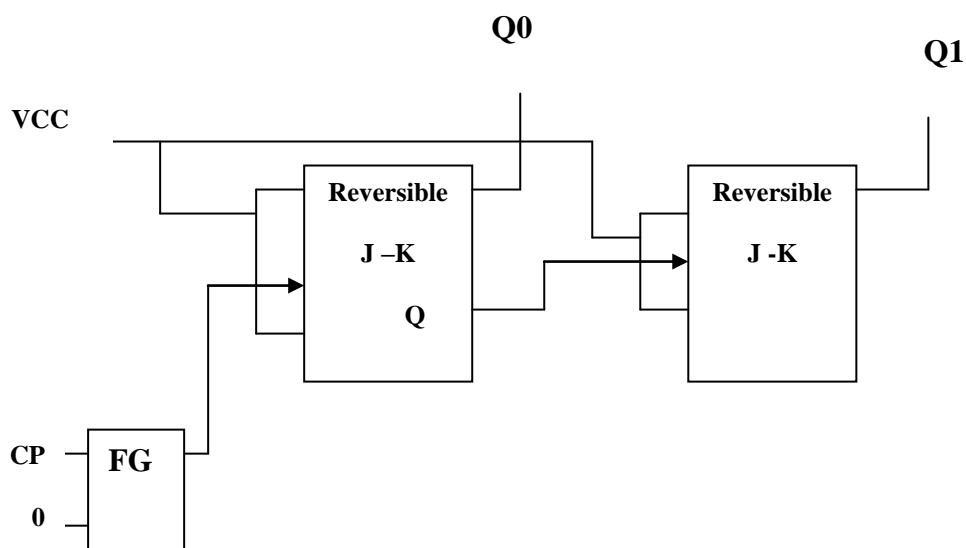


**Figure (10) The reversible J-K flip**

**Evaluation of the Proposed JK Flip Flop**

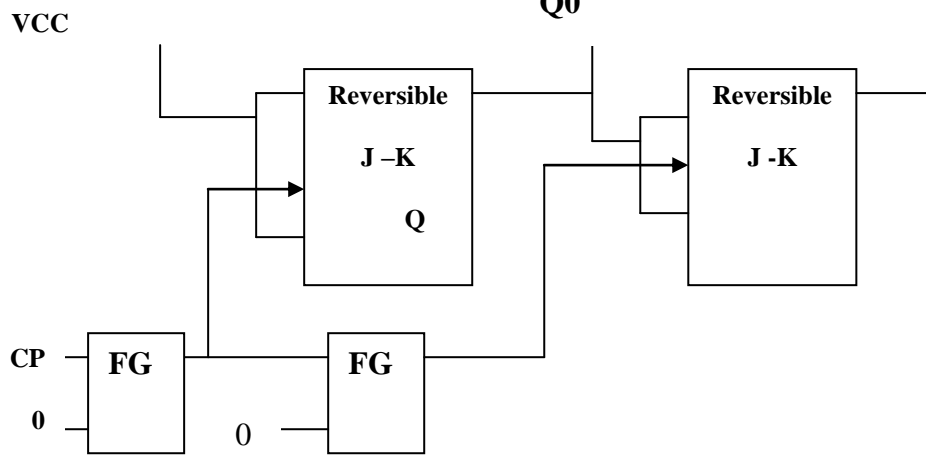
	No of gates	Garbage Outputs
Proposed Circuit	10	12
Existing One	None in literature	None in Literature

The reversible J-K flip flop is used to implement more complex reversible sequential circuits. Figure (11) shows a two-bit asynchronous counter implemented using reversible J-K flip flop

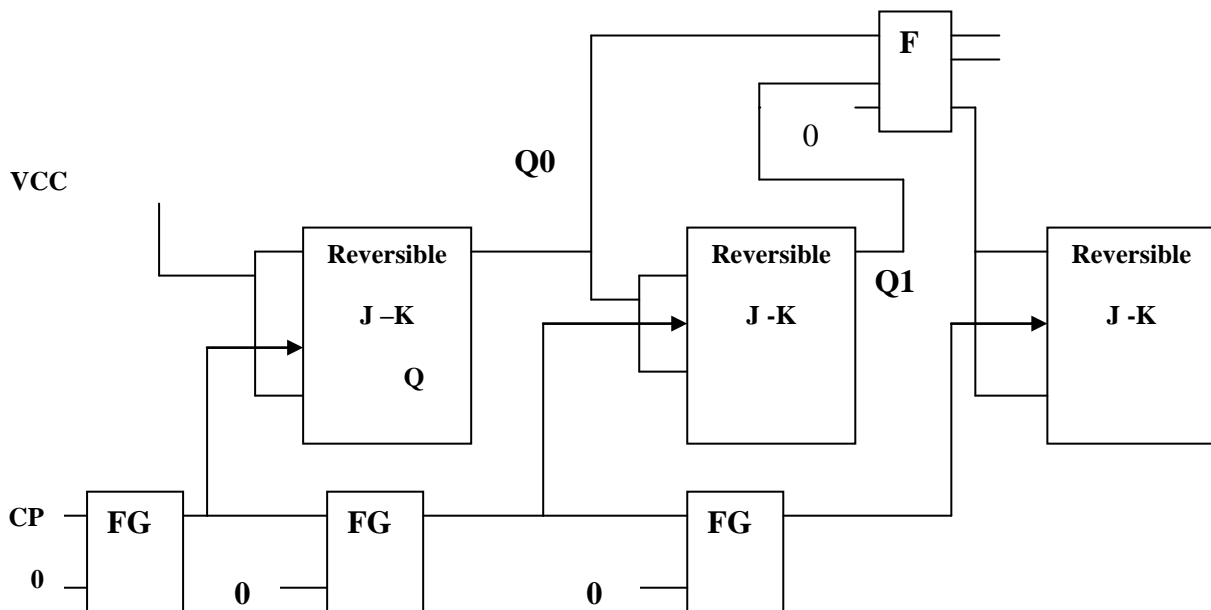


**Figure (11) A two-bit asynchronous**

Figure (12a,b) shows the above two , three bit synchronous counter implemented using J-K flip flop with common clock input (CP)

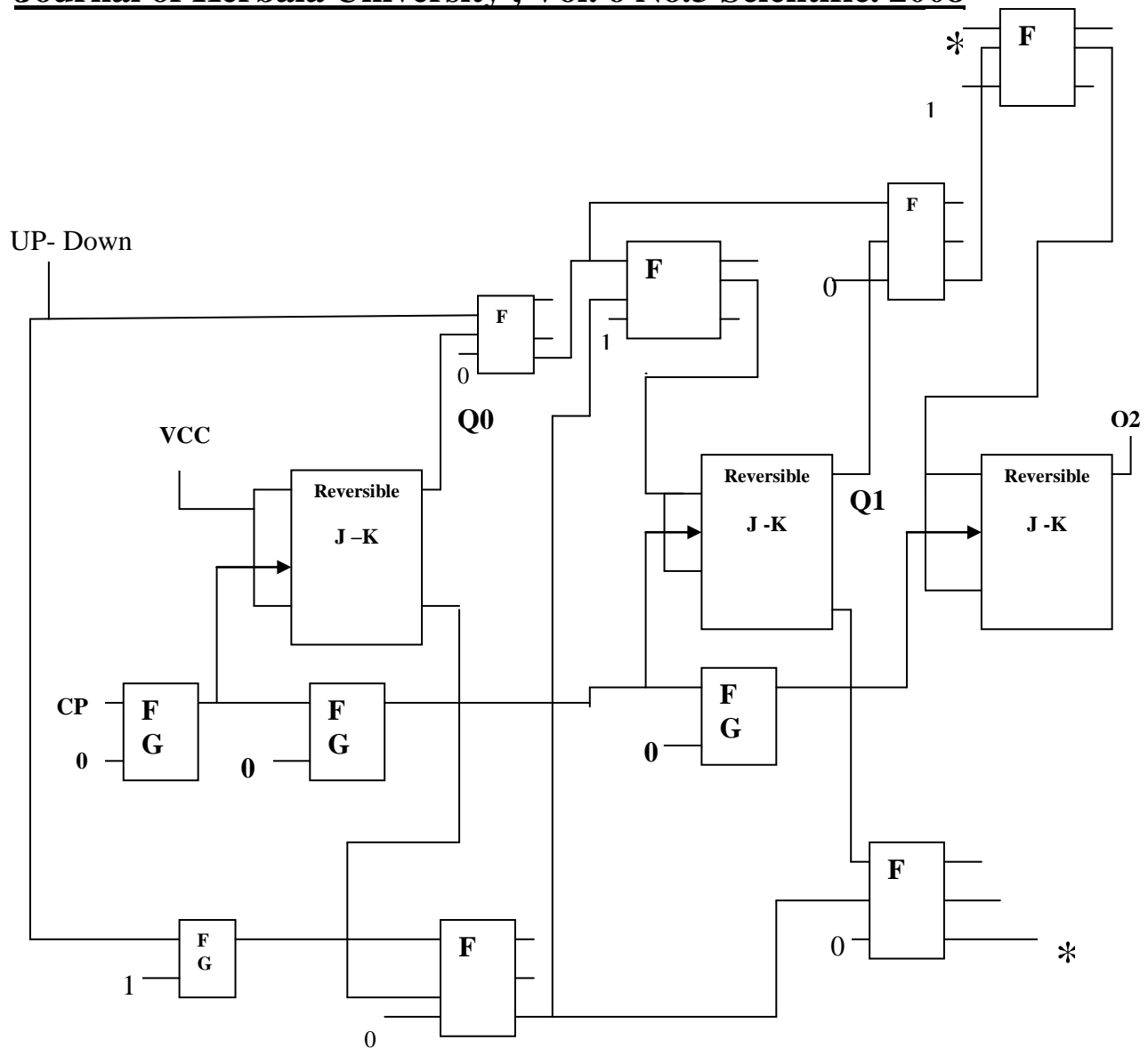


**Figure (12a) A two bit synchronous counter**



**Figure (12b) A three bit synchronous counter**

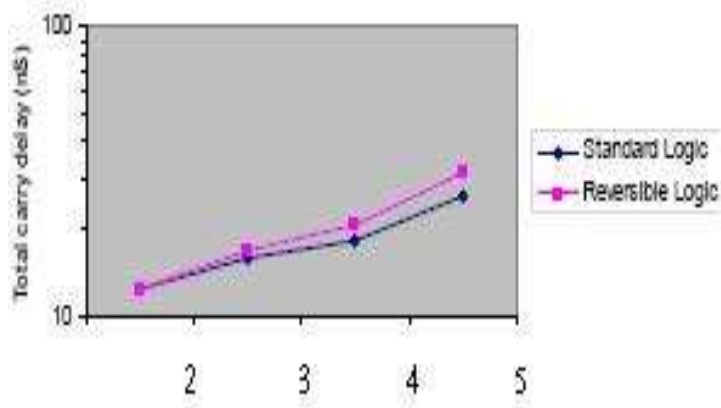
As shown that the Up-Down count is more complex counter also this counter can be built from the reversible J-K flip flop figure(13).



**Figure (13) the Reversible Up/ Down Counter**

**4. Conclusions :**

The focus of this paper is the application of the recently proposed reversible . Different type of reversible counters is also proposed in this paper. All the proposed architectures are analyzed in terms of technology independent implementations. The technology independent analysis is necessary since quantum or optical logic implementations are not available. There are a number of significant applications of reversible logics such as low power CMOS, quantum computing, nanotechnology, and optical computing. The proposed circuit can be used for designing large reversible systems. In a nutshell, the advent of reversible logic has contributed significantly in reducing the power consumption. Thus, the paper provides the initial threshold to build more complex systems which can which can execute more complicated operations. The reversible circuits designed and proposed here form the basis of the ALU of a primitive quantum CPU. The number of Fredkin gates in the reversible implementations seems to be about two times that of the implementation using classical logic gates. we have approximately compared the time delay for the reversible and standard logic implementation for different counter bit sizes, the results appear in figure bellow



**Figure (14) the approximately time delay**

**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, November 1973.

[3] E. Fredkin, T Toffoli, "Conservative Logic", International Journal of Theory. Physics, 21(1982),pp.219-253.

[4] T. Toffoli., "Reversible Computing", Tech memo MIT/LCS/TM-151, MIT Lab for Computer Science (1980).

[5] Alberto LEPORATI, Claudio ZANDRON, Giancarlo MAURI," Simulating the Fredkin Gate with Energy {Based P Systems", Journal of Universal Computer Science, Volume 10, Issue 5,pp 600-619.

[6] M. P. Frank, "Introduction to reversible computing: motivation, progress, and challenges", In Proceedings of the 2nd Conference on Computing Fron-tiers, pages 385–390, 2005.

[7] MANO, "Digital Design", Prentice/Hall International, book, 1986

[8] L.Floyd "Digital Fundamental", seven editions, Prentice Hall, book, 2000

[9] Stephen Brown and Zvonko Varnish," Fundamental of Digital Logic", Amazan.com, Book, 2001

[10] www.rfic.co.uk,"J-K flip flop"

[11] E.Fredkin, T Toffoli," Conservative Logic", International Journal of Theory Physics, 21(1982)

[12] Himanshu Thapliyal , M.B Srinivas and Mark Zwolinski ," A Beginning in the Reversible Logic Synthesis of Sequential Circuits",Centre for VLSI and Embedded System Technologies International Institute of Information Technology

[13] Vlatko Vedral, Adriano Barenco and Artur Ekert, " Quantum Networks for Elementary Arithmetic Operations", arXiv:quantph/ 9511018 v1, nov 1995.