DESIGN OF AN EFFICIENT REVERSIBLE LOGIC BASED BIDIRECTIONAL BARREL SHIFTER

O. ANJANEYULU, T. PRADEEP1 & C.V.KRISHNA REDDY2

1KITs,Warangal, Andhra Pradesh, India
2KSN Inst.of Technology, Nellore,A.P,India
E-mail : anjaneyulu_o@yahoo.com, Pradeep.thumma@yahoo.com, cvkreddy2@gmail.com

Abstract- Embedded digital signal processors and general purpose processors will use barrel shifters to manipulate data. This paper will present the design of the barrel shifter that performs logical shift right, arithmetic shift right, rotate right, logical shift left, arithmetic shift left, and rotate left operations. The main objective of the upcoming designs is to increase the performance without proportional increase in power consumption. In this regard reversible logic has become most popular technology in the field of low power computing, optical computing, quantum computing and other computing technologies. Rotating and data shifting are required in many operations such as logical and arithmetic operations, indexing and address decoding etc. Hence barrel shifters which can shift and rotate multiple bits in a single cycle have become a common choice of design for high speed applications. The design has been done using reversible fredkin and feynman gates. In the design the 2:1 mux can be implemented by fredkin gate which reduce quantum cost, number of ancilla bits and number of garbage outputs. The feynman gate will remove the fanout. By comparing the quantum cost, number of ancilla bits and number of garbage outputs the design is evaluated.

Keywords- barrel shifters, quantum cost, ancilla bits, verilog.

I. INTRODUCTION

Rotating and shifting data is required in several applications including variable-length coding, arithmetic operations, and bit-indexing. Consequently, barrel shifters, which are capable of shifting or rotating data in a single cycle, are commonly found in both digital signal processors and general purpose processors. In reversible system information is not erased. Thus in reversible gates number of inputs and outputs are equal which means that the input stage can always be retained from the output stage. If a bit is erased in an irreversible circuit then it will dissipate kTln2 joules of heat energy where k is the Boltzmann’s constant and T is the absolute temperature of environment [4]. There won’t be dissipation of kTln2 joules of heat energy if the operations are performed in reversible manner based on reversible logic circuits [3]. Based on this observation, Bennett [3] showed, for a reversible computer the heat dissipation is exactly kTln1 which is logically zero. Thus reversible computation is a highly potential field for upcoming low power/high performance computing. Reversible logic also has the applications in emerging nanotechnologies such as quantum dot cellular automata, quantum computing, optical computing and low power computing, etc.

The constraints involved in designing reversible circuits using reversible gates are:

a. The fan-out of every signal is equal to one.

b. Loops are not permitted in a strictly reversible system.
II. REVERSIBLE GATES

A Reversible Gate is an n-input, n-output (denoted by \( n \times n \)) circuit. To maintain the reversibility property of reversible logic gates several dummy output signals are needed to be produced in order to equal the number of input to that of output. These signals are commonly known as Garbage Outputs. For example, for reversible Exclusive-OR operation Feynman gates are used which produce an extra dummy output along with its principal output signal to preserve reversibility. The quantum cost of reversible gate is equal to the number of 1x1 and 2x2 reversible gates needed to design a 3x3 reversible gate. The quantum cost of all 1x1 and 2x2 reversible gates are considered as unity [18], [7], [2]. The 3x3 reversible gates are designed from 1x1 NOT gate, and 2x2 reversible gates such as Controlled-V and Controlled-V+ (V is a square-root of NOT gate and \( V^+ \) is its hermitian), the Feynman gate which is also known as Controlled NOT gate.

A NOT gate is 1x1 gate represented as shown in Fig. 1. Its quantum cost is unity since it is a 1x1 gate.

![Fig. 1. NOT GATE](image)

The input vector, \( I_v \) and output vector, \( O_v \) for \( 2 \times 2 \) Feynman Gate (FE) is defined as follows: \( I_v = (A, B) \) and \( O_v = (P = A \land Q = A \lor B) \). Feynman gates are typically used as copying gates. If \( I_v = (A, B=0) \) then \( O_v = (P = A \land Q = A) \). Fanout is not allowed in reversible logic. Feynman gate is helpful in this regard as it can be used for copying the signal by which it avoids the fanout problem as shown in Fig.2(c).

![Fig. 2. CNOT gate, its quantum implementation and its useful properties](image)

The input and output vector for \( 3 \times 3 \) Fredkin gate (FR) [1] are defined as follows: \( I_v = (A, B, C) \) and \( O_v = (P = A, Q = A \land B \lor C \land AC \land R = A \lor C \land AB) \). Figure 3(a) shows the block diagram of a Fredkin gate. A Fredkin gate can work as 2:1 MUX, as it is able to swap its other two inputs depending on the value of its first input. The first input \( A \) works as a controlling input while the inputs \( B \) and \( C \) work as controlled inputs as shown in the Fig. 3(a). Thus when \( A=0 \) the outputs \( P \) and \( Q \) will be directly connected to inputs \( A \) and \( B \) and if \( A=1 \) the inputs \( B \) and \( C \) will be swapped resulting in the value of the outputs \( Q=C \) and \( R=B \). The quantum implementation of a Fredkin gate with a quantum cost of 5 is shown in Figure 3(b) [7]. In Fig. 3(b) each dotted rectangle is equivalent to a 2x2 Feynman gate and the quantum cost of each dotted rectangle is considered as 1 [18]. The same assumption is used for calculating the quantum cost of the Fredkin gate [7]. Thus, the quantum cost of the Fredkin gate is 5 as it consists of 2 dotted rectangle, 1 Controlled-V gate and 2 CNOT gate.

![Fig. 3. Fredkin Gate and its quantum implementation](image)

III. BI DIRECTIONAL BARREL SHIFTER

A barrel shifter is a combinational circuit which has \( n \)-input and \( n \)-output and \( m \) select lines that controls bit shift operation. A barrel shifter having \( n \) inputs and \( k \) select lines is called \((n,k)\) barrel shifter. Barrel shifter can be unidirectional allowing data to be shifted or rotated only to left (or right), or bidirectional which provides data to be rotated or shifted in both the directions. The logarithmic barrel shifter is mostly widely used among the different designs of barrel shifter, because of its simple design, less area and the elimination of the decoder circuitry. An \( n \)-bit logarithmic barrel shifter has a total of \( \log_2(n) \) stages. Each stage determines whether to shift or not to shift the input data. The stage \( k \) will shift the input \( \text{times if the control bit} \ k (where \ k = 0, 1, \ldots (\log_2(n)-1)) \) is set to 1 otherwise the input will remain unchanged. Logarithmic shifter is more efficient in terms of design as well as area but delay cost is large [11]. This paper presents the designs of reversible bidirectional arithmetic and logical barrel shifter that can perform six operations: logical right shift, arithmetic right shift, right rotate, logical left shift, arithmetic left shift and left rotate. The existing shifter is a unidirectional logarithmic shifter consists of multiplexers. A 3x3 Fredkin Gate works as simple...
(2:1) multiplexers. Feynman gates are used for producing fanouts.

The existing shifter is complex in design and requires large number of gates. As a result the total number of garbage outputs is high. Thus there is great room for improving the circuit complexity, total number of gates and garbage outputs, delay and quantum cost. For efficient designing of a reversible circuit several criteria are needed to be considered:

a. Minimize the number of gates as possible.
b. Minimize the quantum cost of the circuit.
c. Total number of garbage outputs and usage of constant inputs should be minimized.

By maintaining the above parameters and observing the previous design, a novel logarithmic Reversible Barrel Shifter has been proposed. The proposed barrel shifter is a left rotating shifter which uses Fredkin gates for reversible (2:1) multiplexing and Feynman Gates for producing fan outs. A (4, 2) logarithmic barrel shifter has been illustrated in Figure 4. The circuit uses a total of 6 Fredkin gates, 4 Feynman gates and produces 6 Garbage outputs. The Quantum cost of the circuit has also been evaluated. The calculation shows that the Quantum Cost of the proposed (4, 2) circuit is 34.

![Proposed Reversible (4, 2) Barrel Shifter](image)

In this design, the input data is represented as i7, i6, i5, i4, i3, i2, i1, i0 while the shift value is controlled by select signals represented as S2S1S0 and the output data is obtained as shown in Table II.

<table>
<thead>
<tr>
<th>Operation performed by a (n,k) reversible bidirectional Barrel shifter</th>
<th>Control signal values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logical right shift</td>
<td>Left=0 Rot=0 Sra=0 Sla=0</td>
</tr>
<tr>
<td>Arithmetic right shift</td>
<td>Left=0 Rot=0 Sra=1 Sla=0</td>
</tr>
<tr>
<td>Rotate right</td>
<td>Left=0 Rot=1 Sra=0 Sla=0</td>
</tr>
<tr>
<td>Logical left shift</td>
<td>Left=1 Rot=0 Sra=0 Sla=0</td>
</tr>
<tr>
<td>Arithmetic left shift</td>
<td>Left=1 Rot=0 Sra=0 Sla=1</td>
</tr>
<tr>
<td>Rotate left</td>
<td>Left=1 Rot=1 Sra=0 Sla=0</td>
</tr>
</tbody>
</table>

Table II Shift and rotate operation output for k = 3

<table>
<thead>
<tr>
<th>Operation</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-bit shift right logical</td>
<td>0 0 0 a7a6a5a4a3</td>
</tr>
<tr>
<td>3-bit shift right arithmetic</td>
<td>a7a7a7a7a6a5a4a3</td>
</tr>
<tr>
<td>3-bit rotate right</td>
<td>a2a1a0a7a6a5a4a3</td>
</tr>
<tr>
<td>3-bit shift left logical</td>
<td>a4a3a2a1a00000</td>
</tr>
<tr>
<td>3-bit shift left arithmetic</td>
<td>a7a3a2a1a00000</td>
</tr>
<tr>
<td>3-bit rotate left</td>
<td>a4a3a2a1a0a7a6a5</td>
</tr>
</tbody>
</table>

The design of a reversible barrel shifter can be divided into six modules: (i) Data reversal control unit-I, (ii) Arithmetic right shift control unit, (iii) Shifter or rotation unit which consists of three sub-modules that performs Stage I, Stage II and Stage III operations, (iv) Rotation unit, (v) Arithmetic left shift control unit, (vi) Data reversal control unit-II. The reversible design of the modules of the reversible bidirectional barrel shifter along with their working are explained as follows:

1. Data Reversal Control Unit-I
In reversible barrel shifter the direction of the shift operation performed is controlled by the control signal left as shown in the Table I. The reversible bidirectional barrel shifter performs the shift operation in the left direction if the value of control signal left as 1, that is, the arithmetic left shift operation or logical left shift operation. Otherwise, the shift operation is performed in the right direction for the value of left=0, that is, arithmetic right shift operation or logical right shift operation. The data reversal control unit-I has Fredkin gates, since two outputs of the Fredkin gate can work as 2:1 MUXes.
Design of An Efficient Reversible Logic Based Bidirectional Barrel Shifter

4 Fredkin gates can be used to reverse the 8 bit input data by utilizing two outputs of the Fredkin gate as 2:1 Muxes. A left shift operation for a n bit input data by k-bit can be performed in three steps:

(i) Reverse the input data.
(ii) Perform k bit right shift operation, and
(iii) Reverse the outputs of the step (ii).

For example, for a 8-bit input data i7, i6, i5, i4, i3, i2, i1, i0 the three steps of logical left shift operation by 3 bits will be: (i) reverse i7, i6, i5, i4, i3, i2, i1, i0 to produce i0, i1, i2, i3, i4, i5, i6, i7, (ii) perform the 3 bit logical right shift operation to produce 0, 0, 0, i0, i1, i2, i3, i4, and (iii) reverse the outputs of step (ii) to yield i4, i3, i2, i1, i0, 0, 0. The data reversal control unit-I is shown in Fig. 5.

2 Arithmetic Right Shift Control Unit
The reversible right shift control unit is shown in Fig. 5. The arithmetic right shift operation is controlled by the arithmetic right shift control unit. The designing of this unit is done using a single Fredkin gate controlled by the control signal sra, and preserves the sign bit of input data. The arithmetic right shift operation is performed if the value of control signal sra = 1, otherwise it simply passes the data to the next module. Multiple copies of the sign bit are created using the Feynman gates because fanout is not allowed in reversible logic.

![Fig. 5. Proposed (8,3) reversible bidirectional barrel shifter](image)

*FE represents Feynman Gates, FR represents Fredkin gates and G represents the garbage outputs

3 Shifter Or Rotation Unit
The three stage design of the reversible shifter or rotation unit id as shown in Fig. 5. The amount of shift operation that has to be performed is done by the shifter unit in the design of reversible bidirectional barrel shifter. This unit is controlled by the control signals S2, S1 and S0. This unit can be divided into three stages. Depending on the value of control signal S2, S1 and S0, the first, second and the third stages of this unit right shifts the input data by 2², 2¹ and 2º bits respectively. All the three stages are designed using the chain of 8 Fredkin gates controlled by the control signals S2, S1 and S0. The Feynman gates are used in the design to avoid the fanout problem. The working of the three stages of the shifter unit is explained as follows:

- **Stage - I**: The first stage of shifter unit is controlled by the control signal S2 and it will shift the input data by 2²-bits. The input data is right shifted by 2²-bits if the value of control signal S2 is 1, else the input data remains unchanged. The outputs of the Stage I is passed as inputs to Stage II of the shifter unit.
- **Stage - II**: The second stage of the shifter unit is controlled by the control signal S1 and it works on the outputs of the first stage. The input data provided to the second stage is right shifted by 2¹-bits if the value of control signal S1 is 1, else the input data remains unchanged. The outputs of the Stage II is passed as inputs to Stage III of the shifter unit.
- **Stage - III**: The third stage of the shifter unit is controlled by the control signal S0. The output data generated by the stage-II is right shifted by -bits If the value of control signal S0 is 1 else the output data remains unchanged. The outputs of this stage is passed as inputs to the next module in the design of reversible bidirectional barrel shifter.

4 Rotation Unit
The rotation unit is shown in Fig. 5. The rotation operation is controlled by the rotation unit. The designing of this unit is done using a chain of 8 Fredkin gates and controlled by the control signal rot, and performs the rotation operation of input data. The rotation operation is performed if the value of control signal rot = 1, otherwise it simply passes the data to the next module.

5 Arithmetic Left Shift Control Unit
The arithmetic left shift control unit is shown in the Fig. 5. The design of the arithmetic left shift control unit and the design of the arithmetic right shift control unit are same. This control unit is controlled by the control signal sla and is responsible to perform the arithmetic left shift operation. This unit is implemented using a single Fredkin gate. This unit preserves the sign bit needed to perform the arithmetic left shift operation if the value of control signal sla = 1, else it simply passes the LSB of the shifter or rotation unit.

6 Data Reversal Control Unit II
The data reversal control unit is controlled by the control signal left. If the value of control signal left is 1, this unit reverses its input data to generate a left shifted result else it simply passes the input data to its outputs. The data reversal control unit II reverses its 8
Design of An Efficient Reversible Logic Based Bidirectional Barrel Shifter

The design of this unit is shown in Fig. 5 which is same as explained for data reversal control unit I.

The (8,3) reversible bidirectional arithmetic and logical barrel shifter uses 32 Feynman gate to copy the input data to avoid the fanout, and 41 Fredkin gates are used for arithmetic and logical bidirectional shifting and rotating. The above design of the (8,3) reversible bidirectional barrel shifter can be generalized to design a (n,k) reversible bidirectional barrel shifter.

V. PERFORMANCE ANALYSIS

To avoid the fanout problem, in the proposed design Feynman gate is used. Chains of n/2 Fredkin gates are used in data reversal unit-I and data reversal unit-II. The arithmetic right shift control unit uses one Fredkin and $2^k$-1 Feynman gates. Chain of n Fredkin gates and n Feynman gates are used in shifter or rotation unit at each stage. Rotation unit $2^k$ fredkin gates for m=0 to (k-1) for each stage. One Fredkin gate and one Feynman gate is used in arithmetic left shift control unit. Thus the total number of Fredkin gates used to design a (n,k) reversible bidirectional barrel shifter can be written as:

$$\text{FR}=\text{Number of Fredkin gates used in data reversal control unit}+ \text{Number of Fredkin gates used in arithmetic right shift control unit}+ \text{Number of Fredkin gates used in shifter or rotation unit}+ \text{Number of Fredkin gates used in arithmetic left shift control unit}$$

$$+ \text{Number of Fredkin gates used in data reversal control unit-II} = n/2 + 1 + (n+k) + \sum_{m=0}^{k-1} 2^m + 1 + n/2$$

$$= \sum_{m=0}^{k-1} 2^m + n + n*(k+1) + 2.$$ 

The total number of Feynman gates used to design a (n,k) reversible bidirectional barrel shifter is:

$$\text{FE}=\text{Number of Feynman gates required to design arithmetic right shift control unit}+ \text{Number of Feynman gates used in shifter or rotation unit}+ \text{Number of Feynman gates used in arithmetic left shift control unit}$$

$$+ (2^k - 1) + (n+k).$$

Ancilla input Bits

The Table III shows the number of ancilla bits required to design a reversible bidirectional barrel shifter for different values of n and k. $2^k+(n+k)$ Feynman gates are required to design a (n,k) reversible bidirectional barrel shifter. Each Feynman gate requires one ancilla input bit to copy the input data. Additionally, the Fredkin gate used in arithmetic right shift control unit requires one ancilla bit. Hence the total number of ancilla inputs (ANs) required to design a (n,k) reversible bidirectional arithmetic and logical barrel shifter is $\text{ANs}=2^k+(n+k)+1$. Table III shows that the total number of ancilla inputs required to design a (8,3) reversible bidirectional barrel shifter are 33 which is same as illustrated in Fig. 5.

| Table III Ancilla Inputs In (N,K) Reversible Bidirectional Barrel Shifter |
|-------------------|-----|-----|-----|-----|-----|
| n/k               | n=4 | n=8 | n=16| n=32| n=64|
| K=2              | 13  | 21  | 37  | 69  | 133 |
| K=3              | 33  | 57  | 105 | 201 |     |
| K=4              | 81  | 145 | 273 |     |     |
| K=5              |     | 193 | 353 |     |     |
| K=6              |     | 449 |     |     |     |

Quantum Cost

Table IV shows the quantum cost for a reversible bidirectional barrel shifter for different n and k values. The number of Feynman and Fredkin gates used will decide the quantum cost of (n,k) reversible bidirectional barrel shifter. The quantum cost of the Feynman gate is considered as one, while the quantum cost of the Fredkin gate is considered as five. Hence the quantum cost of the proposed design of (n,k) reversible bidirectional barrel shifter can be calculated as $\text{QC}=5*(\sum_{m=0}^{k-1} 2^m+n*(k+1)+2)+2^k+(n*k)$

The quantum cost of a (8,3) reversible bidirectional barrel shifter shown in Fig. 5 is 237.

| Table IV Quantum Cost Of (N,K) Reversible Bidirectional Barrel Shifter |
|--------------------------|-----|-----|-----|-----|-----|
| n/k                     | n=4 | n=8 | n=16| n=32| n=64|
| K=2                    | 137 | 165 | 301 | 573 | 1117|
| K=3                    | 237 | 421 | 789 | 1525|     |
| K=4                    | 565 | 1029| 1957|     |     |
| K=5                    | 1317| 2437|     |     |     |
| K=6                    | 3013|     |     |     |     |

Garbage Outputs

For different reversible bidirectional barrel shifter designs the number of garbage outputs produced is as shown in Table V. In the table, n is the number of input data bits and k represents the shift value. In the design of (n,k) reversible bidirectional barrel shifter the shifter unit can be designed in k stages and each stage consists of the chain of n Fredkin gates to perform the shift operation. Each Fredkin gate in the chain of n Fredkin gates produces at least one garbage output except the last Fredkin gate which produces two garbage outputs. Two garbage outputs are produced by Fredkin gate which is used in the design.
of arithmetic left shift control unit and arithmetic right shift control unit. One garbage output is produced by last Fredkin gate of the data reversal control unit-II as the control signal left cannot be utilized further. Hence the number of garbage outputs (GOs) required to design a (n,k) reversible bidirectional arithmetic and logical shifter can be written as $\text{GOs} = k(n + 1) + 6 + \sum_{m=0}^{k-1} 2^m$. In (8,3) reversible bidirectional barrel shifter design the number of garbage outputs produced in Fig. 5 are 40 which is equal to the result in Table V.

Table V
| Garbage Outputs In (N,K) Reversible Bidirectional Barrel Shifter |
|-------------------------|------------------|------------------|------------------|------------------|
| n/k         | n=4 | n=8 | n=16 | n=32 | n=64 |
| K=2        | 19  | 27  | 43   | 75   | 139  |
| K=3        | 40  | 64  | 112  | 208  |
| K=4        | 89  | 153 | 281  |
| K=5        | 202 | 362 |
| K=6        |     |     | 459  |

CONCLUSIONS

In this paper An Efficient Design of Reversible Logic Based Bidirectional Barrel Shifter has been proposed. The design of the proposed bidirectional shifter is done using Fredkin gates and Feynman gates. The number of garbage outputs, the number of ancilla inputs and the quantum cost of the (n,k) reversible bidirectional barrel shifter increase more rapidly by varying n and keeping k as a constant compared to the designs in which n is kept as a constant while k is varied. The functional verification of the proposed design of the reversible barrel shifters are performed through simulations using the Verilog HDL flow for reversible circuits. The design of bidirectional barrel shifter is evaluated in terms of garbage outputs, ancilla inputs and the quantum cost. The proposed design of reversible bidirectional barrel shifter can perform logical right shifting, arithmetic right shifting, rotating right, logical left shifting, arithmetic left shifting and rotating left operations.

REFERENCES