Application of Reversible Logic Approach in 16 bit Arithmetic Logic Unit

Ankita Verma¹, Rita Jain²
¹M.tech student, Electronics and Communication Dept.
²Head of Electronics and Communication Dept.
Lakshmi Narain College of Technology, Bhopal, India

Abstract—In modern Era of circuit designing, complexity of circuit increases day by day. Hence power dissipation plays important role in designing of any digital circuit. In earlier many approaches were used to reduce power dissipation. Reversible logic design can also be used for same objective. This approach gaining importance day by day. Arithmetic logic unit is very important part of central processing unit. So it must be fast in term of computations and should dissipate less power. Here a technique is discussed for designing arithmetic and logic unit with the use of reversible gates. Modules are designed using VHDL. Synthesis and simulation is carried out on Xilinx plan ahead 14.4.

Keywords—VLSI, ALU, Reversible gates, Constant input, Garbage output.

I. INTRODUCTION

In modern VLSI (very large scale integration) system power dissipation is very high due to rapid switching of internal signals. The complexity of VLSI circuits increases with each year due to packing more and more logic elements into smaller volumes. Hence power dissipation become the main area of concern in VLSI design. Sources of power dissipation can be given as the leakage current, which is primarily determined by the fabrication technology, consists of two components: Reverse bias current in the parasitic diodes formed between source and drain diffusions and the bulk region in a MOS transistor. The sub threshold current that arises from the inversion charge that exists at the gate voltages below the threshold voltage, the standby current which is the DC current drawn continuously from Vdd to ground. The capacitance current which flows to charge and discharge capacitive loads during logic changes. The short-circuit (rush-through) current which is due to the DC path between the supply rails during output transitions. In Conventional logic circuits, the heat is generated due to the loss of information during computation. In order to prevent this loss of information the conventional circuits are modeled using reversible logic. Landauer [1961], shows that for conventional logic computations, each bit of information lost, generates kT log2 joules of heat energy, where k is Boltzmann’s constant and T the absolute temperature at which computation is performed [1]. A logic which do not dissipate called reversible. Bennett [6] showed zero energy dissipation would be possible only if the network consists of reversible gates. An irreversible computer can always be made reversible by having it save all the information it would otherwise throw away. Thus reversibility will become an essential property in future circuit design Bennett showed that kT log2 energy dissipation would not occur, if a computation is carried out in a reversible way [2]. This paper is organized as follows: Section II gives literature review. Section III gives the brief introduction about the reversible logic gates. Section IV describes the method. Section V shows the synthesis and simulation results. Section VI shows the conclusion.

II. LITERATURE REVIEW

Landauer, Rolf. [1] R Landauer’s showed, amount of heat generation due to loss of bit is kT log2, and this value is approximately 2.8*10^-21 joule, which is small but not negligible. Bennett, Charles H [2] showed that this heat dissipation due to information loss can be avoided if the circuit is designed using reversible logic gates. No amount of heat would be dissipated from the system as long as the system was able to return to its initial state from its final state. E. Fredkin, T Toffoli, Feynman, Peres [3] [4] [5] proposed reversible logic gates. N. Srinivasa Rao, P. Satyanarayana [6]. They proposed a new gate SSG, and its applications such as half adder, half subtractor, full adder and full subtractor. Himanshu Thapliyal and M B Srinivas [7]. They proposed new gate and showed that their gate is better in terms of number of reversible gates and garbage outputs. Lekshmi Viswanath, ponn M [8] designed an arithmetic and logic unit with reversible gates and compared their results with conventional unit.

III METHOD USED

Arithmetic and logical unit is a basic part of central processing unit. It performs various functions. Arithmetic operators perform arithmetic functions like addition, subtraction and multiplication. Logical operators performs logical functions such as logical AND, logical OR, logical NOT, logical XOR etc. while designing this unit some criteria should be followed such as speed should be more, should use less area and should
dissipates less power. All mentioned criteria are important for efficient designing. In designing of arithmetic logic unit, most common approach is usage of multiplexer, it selects input and respective operation will perform. In this, multiplexer selects input for every operation, but at the time of computation all modules take part in computation, it uses more number of logic gate at this time. Hence, power associated with those gates also plays very important role. So here main approach is to reduced gate count at the time of computation. For designing of arithmetic and logical unit, a reconfigurable decoder approach is used. A 3 to 1 decoder circuit is used which act as a selection unit for every module. This decoder enables the individual modules. At a time only one module is activated. Other modules will remain deactivated. By this approach we can reduce number of gate counts during computation. This approach can be understood by this diagram.

Fig. 1 Proposed approach

3.1 Adder/ sub Module

If two n bit binary numbers are to be added or subtracted then n binary full adder/subtractor should be cascaded. For designing 16 bit module, 16 binary full adder/subtractor should be cascaded. A parallel adder/subtractor is the interconnection of a number of full adder/subtractor and applying the inputs simultaneously. In this, 4 bit parallel adder/subtractor circuit is designed using a 4x4 reversible DKG gate. Fig. 2 shows the reversible DKG gate. This gate can act as an adder or subtractor depending on its control input A, when A is 0 the gate behaves as a full adder and when A is 1 the gate behaves as a subtractor. The block diagram of a 4 bit reversible adder/subtractor using DKG gate is shown below.

Condition for addition or subtraction can be given as
When A =0
P=B; \hspace{1cm} (1)
Q=C; \hspace{1cm} (2)
R= (C \oplus D) \oplus (CD); \hspace{1cm} (3)
S= (B \oplus C \oplus D); \hspace{1cm} (Sum\ of\ full\ adder) \hspace{1cm} (4)
When A =1
P=B; \hspace{1cm} (5)
Q=D'; \hspace{1cm} (6)
R= [B'(C \oplus D)] \oplus CD; \hspace{1cm} (7)
S= (B \oplus C \oplus D); \hspace{1cm} (Difference\ of\ full\ sub) \hspace{1cm} (8)

3.2 Multiplier Module

A binary multiplier is an electronic circuit used in digital electronics, such as a computer, to multiply two binary numbers. It is built using binary adders. A variety of computer arithmetic techniques can be used to implement a digital multiplier. Most techniques involve computing a set of partial products, and then summing the partial products together.

3.2.1 Implementation of Multiplier with Algorithm

The following algorithm can be used to design an 8x8 multiplier using 4x4 multiplier
1. Let A and B be two 8 bit numbers that are to be multiplied.
2. Divide A into equal halves A1 and A0 such that A0 indicates the 4bit LSB and A1 indicates the 4bit MSB. Divide B also in the same way as B0 and B1.
3. Multiply A0 and B0 using reversible 4bit multiplier. This forms the first partial product PR1. Retain the 4bit LSB of PR1.
4. Now multiply A1 and B0. This forms the partial product PR2. Then multiply A0 and B1. This forms partial product PR3.
5. Add PR2, PR3 and the 4bit MSB of PR1 using reversible parallel adder designed from TSG gate. This Forms the temporary result TR1. Retain the 4bit LSB of TR1.

6. Multiply A1 and B1 resulting in the partial product PR4. Now add PR4 and the remaining bits of TR1 (Excluding the 4bit LSB) resulting in the temporary result TR2.

7. Now concatenate TR2 with the 4bit LSB of TR1 and PR1. This forms the final 16 bit product. The same can be apply for designing higher multipliers using lower multipliers.

### 3.3 Logical unit

This unit contains of logical operators such as AND, OR, NOT, XOR, NAND.

![Fig 3 1 bit logical unit](image)

Logical block can be summarized as:

- When CD=00 (PFAG block acts as an AND gate and the output of the AND operation between A and B is obtained at S).
- When CD=01 (NAND operation between A and B is performed and the output is obtained at S).
- When CD=10 (OR operation between A and B is performed and obtained at S).
- When CD=11 (NOT operation of A is carried out and obtained at Q1).
- The XOR operation of A and B is obtained at Q irrespective of the value of C and D.

Total reversible gates used in all modules can be summarized as:

- For implementing a 4bit reversible adder/subtractor = 4 DKG gates are required.
- Hence for 16 bit reversible adder/subtractor = 16 DKG gates are required.

- For implementing 4x4 multiplier = 29 gates (16 Fredkin gates +13 TSG gates).
- For implementing 16 bit logical unit design requires a total of 32 reversible gates (16 PFAG + 16 FG).

### IV RESULTS

#### 4.1 Power analysis:
Power analysis is done on xilinx planahead 14.4

<table>
<thead>
<tr>
<th>Name</th>
<th>Power (W)</th>
<th>IO Standard</th>
<th>Signal Rate</th>
<th>% High</th>
<th>Clock (MHz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carry</td>
<td>0.00000</td>
<td>0.010000</td>
<td>1.000000</td>
<td>0.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>A_0_BUF</td>
<td>0.00002</td>
<td>0.010000</td>
<td>1.100000</td>
<td>0.52</td>
<td>1.0000</td>
</tr>
<tr>
<td>B_1_BUF</td>
<td>0.00002</td>
<td>0.010000</td>
<td>1.100000</td>
<td>0.52</td>
<td>1.0000</td>
</tr>
<tr>
<td>C_1_BUF</td>
<td>0.00455</td>
<td>0.100000</td>
<td>1.200000</td>
<td>0.67</td>
<td>1.0000</td>
</tr>
<tr>
<td>Command</td>
<td>0.00001</td>
<td>0.010000</td>
<td>0.000000</td>
<td>0.0000</td>
<td>1.0000</td>
</tr>
</tbody>
</table>

Power dissipation due to i/o's

<table>
<thead>
<tr>
<th>Name</th>
<th>Power (W)</th>
<th>IO Standard</th>
<th>Signal Rate</th>
<th>% High</th>
<th>Clock (MHz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carry</td>
<td>0.000001</td>
<td>0.010000</td>
<td>1.000000</td>
<td>0.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>A_0_BUF</td>
<td>0.00002</td>
<td>0.010000</td>
<td>1.100000</td>
<td>0.52</td>
<td>1.0000</td>
</tr>
<tr>
<td>B_1_BUF</td>
<td>0.00002</td>
<td>0.010000</td>
<td>1.100000</td>
<td>0.52</td>
<td>1.0000</td>
</tr>
<tr>
<td>C_1_BUF</td>
<td>0.00455</td>
<td>0.100000</td>
<td>1.200000</td>
<td>0.67</td>
<td>1.0000</td>
</tr>
<tr>
<td>Command</td>
<td>0.00001</td>
<td>0.010000</td>
<td>0.000000</td>
<td>0.0000</td>
<td>1.0000</td>
</tr>
</tbody>
</table>

Power dissipation due to data

<table>
<thead>
<tr>
<th>Name</th>
<th>Power (W)</th>
<th>IO Standard</th>
<th>Signal Rate</th>
<th>% High</th>
<th>Clock (MHz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carry</td>
<td>0.000001</td>
<td>0.010000</td>
<td>1.000000</td>
<td>0.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>A_0_BUF</td>
<td>0.00002</td>
<td>0.010000</td>
<td>1.100000</td>
<td>0.52</td>
<td>1.0000</td>
</tr>
<tr>
<td>B_1_BUF</td>
<td>0.00002</td>
<td>0.010000</td>
<td>1.100000</td>
<td>0.52</td>
<td>1.0000</td>
</tr>
<tr>
<td>C_1_BUF</td>
<td>0.00455</td>
<td>0.100000</td>
<td>1.200000</td>
<td>0.67</td>
<td>1.0000</td>
</tr>
<tr>
<td>Command</td>
<td>0.00001</td>
<td>0.010000</td>
<td>0.000000</td>
<td>0.0000</td>
<td>1.0000</td>
</tr>
</tbody>
</table>

Power dissipation due to logic
Above analysis can be summarized as:

Table 1 power distribution

<table>
<thead>
<tr>
<th>Power dissipation in (m watts)</th>
<th>This paper</th>
<th>Reference paper</th>
</tr>
</thead>
<tbody>
<tr>
<td>IO power</td>
<td>4.60</td>
<td>6.75</td>
</tr>
<tr>
<td>Signal power</td>
<td>0.32</td>
<td>0.30</td>
</tr>
<tr>
<td>Logic power</td>
<td>0.10</td>
<td>0.07</td>
</tr>
<tr>
<td>Total power</td>
<td>5.02 m watt</td>
<td>7.12 m watt</td>
</tr>
</tbody>
</table>

4.2 Area analysis (in terms of number of gates)

Table 2 Number of gates analysis

<table>
<thead>
<tr>
<th>Logic units</th>
<th>This paper</th>
<th>Reference paper</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 bit XORs2</td>
<td>496</td>
<td>768</td>
</tr>
<tr>
<td>1bit XORs3</td>
<td>16</td>
<td>80</td>
</tr>
</tbody>
</table>

V CONCLUSION

It can be concluded by above result that power dissipation can be reduced up to by some extent by using reversible logic approach. With the use of this approach there is reduction in gate count that is 40 %, thus area requirement gets reduced for developing same logic. With the reduction in gate count 30% power is reduced. In future, other operations can be implemented using this approach.

REFERENCES
