Investigating the VLSI Characterization of Parallel Signed Multipliers for RNS Applications using FPGAs

By Mr. Pradeep. N, Mr. S. Elango, Dr. P. Sampath & Ms. Gayathri

Bannari Amman Institute of Technology, India

Abstract- Signed multiplication is a complex arithmetic operation, which is reflected in its relatively high signal propagation delay, high power dissipation, and large area requirement. High reliability applications such as Cryptography, Residue Number System (RNS) and Digital Signal Processing (DSP)’s effective performance is mainly depend on its arithmetic circuit’s performance. Trend of using Residue Number System (RNS) instead of Constrain over-whelming Binary representation is promising technique in VLSI Systems and Multiplier is the basic building block of such systems. In this paper we have considered signed Modified Baugh Wooley Multiplier and Modified Booth Encoding (MBE) Multiplier logic for analysis and synthesized on best suited application platform. Analysis has taken account of Delay, Number of Logic Element requirements; Number of Signal Transition for particular sample input and its Power Consumption were analyzed for both Modified Baugh Wooley Multiplier and Modified Booth Encoding Multiplier.

Keywords: baugh wooley multiplier, modified booth encoding (mbe), computer arithmetic, signed multiplier, verilog hdl, xilinx ise, altera quartus.

GJCST-A Classification : B.7.1

Strictly as per the compliance and regulations of:
Investigating the VLSI Characterization of Parallel Signed Multipliers for RNS Applications using FPGAs

Mr. Pradeep N °, Mr. S. Elango °, Dr. P. Sampath ° & Ms. Gayathri°

Abstract- Signed multiplication is a complex arithmetic operation, which is reflected in its relatively high signal propagation delay, high power dissipation, and large area requirement. High reliability applications such as Cryptography, Residue Number System (RNS) and Digital Signal Processing (DSP)'s effective performance is mainly depend on its arithmetic circuit's performance. Trend of using Residue Number System (RNS) instead of Constrain overwhelming Binary representation is promising technique in VLSI Systems and Multiplier is the basic building block of such systems. In this paper we have considered signed Modified Baugh Wooley Multiplier and Modified Booth Encoding (MBE) Multiplier logic for analysis and synthesized on best suited application platform. Analysis has taken account of Delay, Number of Logic Element requirements; Number of Signal Transition for particular sample input and its Power Consumption were analyzed for both Modified Baugh Wooley Multiplier and Modified Booth Encoding Multiplier. Analysis of Multiplier is described in Verilog HDL and Simulated using two different simulators namely Xilinx ISIM and Altera Quartus II. Then for comparative study, both multipliers are synthesized with Xilinx Virtex 7 XCV2000T-2FLG1925 and Altera Cyclone II EP2C35F672C6 and same parameter as discussed above are also evaluated. Booth Recoding provides overall advent of 9.691% in terms of area and approximately 43 % in terms of Delay compared to Modified Baugh Wooley Multiplier implemented using FPGA Technology.

Keywords: baugh wooley multiplier, modified booth encoding (mbe), computer arithmetic, signed multiplier, verilog hdl, xilinx ise, altera quartus.

I. INTRODUCTION

Multiplication is a most generally used operation in wide computing systems. In fact multiplication is nothing but addition since, multiplicand adds to itself multiplier number of times, gives the multiplication value between multiplier and multiplicand. But considering the fact that this kind of implementation really takes huge hardware resources and the circuit operates at utterly low speed. In order to address this so many ideas have been presented so far for the last three decades. Each one is aimed at improvement according to the requirement. One may be aimed at high clock speeds and another may be aimed for low power or less area occupation. Either way ultimate job is to come up with an efficient architecture which can address three constraints of VLSI speed, area, and power. Among three constrains, speed is the vital one which requires more attention. If we observe closely multiplication operation involves two steps one is producing partial products and adding these partial products [3].

Thus, the speed of a multiplier hardly depends on how fast generate the partial products and how fast we can add them together. Since the multipliers have a significant impact on the performance of the entire system, many high performance algorithms and architectures have been proposed [1-12]. The very high speed and dedicated multipliers are used in pipeline and vector computers.

Residue Number System (RNS) reduces the delay of carry propagation, thus offering significant speed up over the conventional binary system. This characteristic is advantageous when repetitive arithmetic operations on long operands have to be performed. RNS has been adopted in the design of Digital Signal Processors (DSP). The low power consumption of RNS compared to conventional arithmetic circuits for the implementation of Finite Impulse Response (FIR) filters inspired lot of work against it. Therefore, RNS may be an interesting candidate for building processing circuits in deep submicron technologies.

The rest of the paper is organized as: Section-II describes Baugh-Wooley Multiplication Section-III provides deep understanding about Modified Booth Encoding techniques. Comparative results and its analysis are exploited in Section-IV and Finally Conclusion of the paper illustrated in Section –V.

II. BAUGH WOOLEY MULTIPLIER

The Baugh-Wooley multiplication is one of the efficient methods to handle the sign bits and this approach has been developed in order to design regular multipliers[2], suited for 2’s complement numbers.

Let us consider two n-bit signed numbers, X (Multiplicand) and Y (Multiplier), to be multiplied...
\[ X = -x_{n-1}2^{n-1} + \sum_{i=0}^{n-2} x_i 2^i \]  
\[ Y = -y_{n-1}2^{n-1} + \sum_{i=0}^{n-2} y_i 2^i \]

where the \( x_i \)'s and \( y_i \)'s are the bits in \( X \) and \( Y \), respectively, and \( x_{n-1} \) and \( y_{n-1} \) are the sign bits.

The product, \( P = X \times Y \), is then given by the following equation:

\[
P = \left(-x_{n-1}2^{n-1} + \sum_{i=0}^{n-2} x_i 2^i\right) \left(-y_{n-1}2^{n-1} + \sum_{j=0}^{n-2} y_j 2^j\right)
= x_{n-1}y_{n-1}2^{2n-2} + \sum_{i=0}^{n-2} \sum_{j=0}^{n-2} x_i y_j 2^{i+j}
-2^{n-1}\sum_{i=0}^{n-2} x_i y_{n-1} 2^i - 2^{n-1}\sum_{j=0}^{n-2} y_{n-1} y_j 2^j
\]

The final product can be obtained by subtracting the last two positive terms from the first two terms.

Instead of pursuing subtraction operation, it is possible to obtain the 2's complement of the last two terms and add all terms to get the final product.

The final product (3), \( P = X \times Y \) becomes:

\[
P = x_{n-1}y_{n-1}2^{2n-2} + \sum_{i=0}^{n-2} x_i 2^i \sum_{j=0}^{n-2} y_j 2^j
+2^{2n-1}\sum_{i=0}^{n-2} x_i y_{n-1} 2^i + 2^{n-1}\sum_{j=0}^{n-2} y_{n-1} y_j 2^j
-2^{2n-1} + 2^n
\]

Simple 4x4 Baugh-wooley multiplication is exhibited in figure 1.

The modified-Booth algorithm [1] is more preferred and extensively used for high-speed multiplier circuits. Modified Booth Multiplier is one of the different techniques for signed multiplication This multiplier architecture is based on Radix 4(2^2) Booth multiplier. In order to improve the architecture, we have made 2 enhancements as in [14]. The first is to use efficient Wen-Chang’s Modified Booth Encoder (MBE) since it is proved as the fastest scheme to generate a partial product.

\( a) \) Algorithm of the Modified Booth Multiplier

Booth Multiplication consists of three[10-14] steps:

1. The first step to generate the partial products;
2. The second step to add the generated partial products until the last two rows are remained;
3. The third step to compute the final multiplication results by adding the last two rows.

The modified Booth algorithm reduces the number of partial products by half in the first step. We
used the modified Booth encoding (MBE) scheme proposed in [1]. It is known as the most efficient Booth encoding and decoding scheme. To multiply M by N using the modified Booth algorithm starts from grouping N by three bits and encoding into one of \{ -2, -1, 0, 1, 2 \}. Figure 3 exhibit the general architecture of MBE.

![Figure 3](image)

Table 1: Modified Booth Encoder Logic [1]

<table>
<thead>
<tr>
<th>(b_3)</th>
<th>(b_2)</th>
<th>(b_1)</th>
<th>Operation</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Add 0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>A</td>
<td>Add Multiplicand</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>A</td>
<td>Add Multiplicand</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2A</td>
<td>Two times Add Multiplicand</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>-2A</td>
<td>2's Complement of</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Multiplicand and Add 2 times.</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>-A</td>
<td>2's Complement of</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>multiplicand and Add</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>-A</td>
<td>2's Complement of</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>multiplicand and Add</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>Add 0</td>
</tr>
</tbody>
</table>

In this case, the multiplicand is offset one bit to the left to enter into the adder while for the low-order multiplicand position a 0 is added. Each time the partial product is shifted two bit positions to the right and the sign is extended to the left.

During each add-shift cycle, different versions of the multiplicand are added to the new partial product depends on the equation derived from the bit-pair recoding table above.

Here are some examples for understanding:

**Example 1:** For One negative and One positive number.
Consider \(-3 \times 5\)

**Step-1:** Binary conversion and 2's complement

\[
\begin{align*}
3 & \Rightarrow 011 \\
1's \ comp & \Rightarrow 100 \\
-3 & \Rightarrow 1101 (+) \\
\end{align*}
\]

**Step-2:** Multiplication by Modified booth recoding

\[
\begin{align*}
1101 \times 0101(0) & \rightarrow 111101 \\
& \rightarrow 1101 \\
& \rightarrow 110001 \\
1's \ comp & \Rightarrow 001110 \\
& \Rightarrow 11 (+) \\
& \Rightarrow 001111 (+) \\
& \Rightarrow -15 \\
\end{align*}
\]

**Example 2:** For Both Negative Numbers.
Consider \(-3 \times -4\)

**Step-1:** Binary conversion and 2's complement

\[
\begin{align*}
-3 & \Rightarrow 1101 \quad 4 \Rightarrow 0100 \\
1's \ comp & \Rightarrow 1101 \quad 1 (+) \\
& \Rightarrow 001110 \\
& \Rightarrow 11 (+) \\
& \Rightarrow -16 \\
\end{align*}
\]

**Step-2:** Multiplication by Modified booth recoding

\[
\begin{align*}
1101 \times 0101(0) & \rightarrow 111101 \\
& \rightarrow 1101 \\
& \rightarrow 110001 \\
1's \ comp & \Rightarrow 001110 \\
& \Rightarrow 11 (+) \\
& \Rightarrow 001111 (+) \\
& \Rightarrow -15 \\
\end{align*}
\]
Once the partial products are generated then the addition process is very similar to the array multiplier.

IV. RESULTS AND ANALYSIS

The Multiplier were taken for analysis was described using structural Verilog HDL and synthesized to produce a gate level net list using two different synthesizer namely Xilinx ISE Design Suite 14.3, Altera Quartus II 12.0 with reference to Virtex7 XCV2000T-2FLG1925 and Cyclone II EP2C35F672C6 FPGA respectively. The multipliers were simulated and analyzed at different strengths such as 4 x 4, 8 x 8, 16 x16, 32 x 32 and 64 x 64 as shown below in table [2-4].

a) Area Analysis

In FPGA based design, Area requirement of the design is proportional to logic utilization i.e in Xilinx - Number of Slice LUTs Required and in Altera its Number of Logic Elements Required. For 16 x 16 bit strength Booth Consume 20.5% lesser area than Baugh-Wooley Multiplier.

b) Delay Analysis

In FPGA based Design, EDA tools having inbuilt capability to predict the Delay of the design. In Xilinx - Timing Analyzer Tool and in Altera Time Quest Timing Analyzer Tool were used for delay analyze. Various Delay analysis shows Modified Booth has about 43% performance efficient over Baugh-Wooley.

c) Power Analysis

Power Evaluation of the design done at various levels such as Total Thermal power Dissipation (mW-milli Watt's), Core Dynamic Thermal power Dissipation (mW), core static Thermal power Dissipation (mW), I/O Thermal Power Dissipation(mW). Among the various power levels dynamic power varies with design to design it decides the efficient architecture.

Dynamic Power Requirement of the design is decided based on number of signal transition (or) activity during simulation time. Here analysis has been made using Power Play Power Analyzer from Altera. Power Analyzer required an input file of Signal Activities and Value Changed Dump (VCD) File to evaluate the power of the design. Here we have measure the signal activities count for 20 different Samples for 100ns simulation and the same sample is forced for other design also in order to evaluate the exact power difference between the design. power Analysis with powerplay analyzer tool for 4 x 4 bit shows 46.90% Modified Booth consume less than Baugh-wooley Multiplier and found consistence for all strength.

Table 2: Area analysis using Altera Quartus-II

<table>
<thead>
<tr>
<th>Multipliers Strength</th>
<th>Multiplier Name</th>
<th>No. of IOBs</th>
<th>Altera Cyclone II EP2C35F672C6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>No. of Logic Elements Required</td>
</tr>
<tr>
<td>4x4</td>
<td>BAUGH</td>
<td>16</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>BOOTH</td>
<td>16</td>
<td>28</td>
</tr>
<tr>
<td>8x8</td>
<td>BAUGH</td>
<td>32</td>
<td>164</td>
</tr>
<tr>
<td></td>
<td>BOOTH</td>
<td>32</td>
<td>150</td>
</tr>
<tr>
<td>16x16</td>
<td>BAUGH</td>
<td>64</td>
<td>698</td>
</tr>
<tr>
<td></td>
<td>BOOTH</td>
<td>64</td>
<td>538</td>
</tr>
<tr>
<td>32x32</td>
<td>BAUGH</td>
<td>128</td>
<td>2,874</td>
</tr>
<tr>
<td></td>
<td>BOOTH</td>
<td>128</td>
<td>2,284</td>
</tr>
<tr>
<td>64x64</td>
<td>BAUGH</td>
<td>256</td>
<td>10,122</td>
</tr>
<tr>
<td></td>
<td>BOOTH</td>
<td>256</td>
<td>9,542</td>
</tr>
</tbody>
</table>

Table 3: Area and Delay analysis using Xilinx ISE

<table>
<thead>
<tr>
<th>Multipliers Strength</th>
<th>Multiplier Name</th>
<th>No. of IOBs</th>
<th>Xilinx Virtex7 XCV2000T-2FLG1925</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>No. of Slice LUTs Required</td>
</tr>
<tr>
<td>4x4</td>
<td>BAUGH</td>
<td>16</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>BOOTH</td>
<td>16</td>
<td>18</td>
</tr>
<tr>
<td>8x8</td>
<td>BAUGH</td>
<td>32</td>
<td>104</td>
</tr>
<tr>
<td></td>
<td>BOOTH</td>
<td>32</td>
<td>96</td>
</tr>
<tr>
<td>16x16</td>
<td>BAUGH</td>
<td>64</td>
<td>452</td>
</tr>
<tr>
<td></td>
<td>BOOTH</td>
<td>64</td>
<td>354</td>
</tr>
<tr>
<td>32x32</td>
<td>BAUGH</td>
<td>128</td>
<td>1851</td>
</tr>
<tr>
<td></td>
<td>BOOTH</td>
<td>128</td>
<td>1595</td>
</tr>
<tr>
<td>64x64</td>
<td>BAUGH</td>
<td>256</td>
<td>7392</td>
</tr>
<tr>
<td></td>
<td>BOOTH</td>
<td>256</td>
<td>6480</td>
</tr>
</tbody>
</table>
Table 4: Power Analysis (Time interval of 100ns with 20 different samples)

<table>
<thead>
<tr>
<th>Multipliers Strength</th>
<th>Multiplier Name</th>
<th>Number Signal Transition during simulation for 100ns</th>
<th>Total Thermal Power Dissipation (mW)</th>
<th>Core Dynamic Thermal Dissipation (mW)</th>
<th>Core Static Thermal power Dissipation (mW)</th>
<th>I/O Thermal power Dissipation (mW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4x4</td>
<td>BAUGH</td>
<td>1857</td>
<td>169.92</td>
<td>1.13</td>
<td>80.12</td>
<td>86.67</td>
</tr>
<tr>
<td></td>
<td>BOOTH</td>
<td>986</td>
<td>166.13</td>
<td>1.01</td>
<td>80.01</td>
<td>86.59</td>
</tr>
<tr>
<td>8x8</td>
<td>BAUGH</td>
<td>20911</td>
<td>223.47</td>
<td>4.81</td>
<td>80.30</td>
<td>138.36</td>
</tr>
<tr>
<td></td>
<td>BOOTH</td>
<td>10291</td>
<td>223.39</td>
<td>5.28</td>
<td>80.30</td>
<td>138.30</td>
</tr>
<tr>
<td>16x16</td>
<td>BAUGH</td>
<td>498261</td>
<td>351.24</td>
<td>27.12</td>
<td>80.74</td>
<td>243.39</td>
</tr>
<tr>
<td></td>
<td>BOOTH</td>
<td>51942</td>
<td>345.25</td>
<td>19.86</td>
<td>80.72</td>
<td>244.67</td>
</tr>
<tr>
<td>32x32</td>
<td>BAUGH</td>
<td>9606019</td>
<td>642.20</td>
<td>115.05</td>
<td>81.75</td>
<td>445.40</td>
</tr>
<tr>
<td></td>
<td>BOOTH</td>
<td>469336</td>
<td>601.67</td>
<td>82.31</td>
<td>81.61</td>
<td>437.74</td>
</tr>
<tr>
<td>64x64</td>
<td>BAUGH</td>
<td>19212038</td>
<td>1302.34</td>
<td>331.53</td>
<td>83.13</td>
<td>887.68</td>
</tr>
<tr>
<td></td>
<td>BOOTH</td>
<td>1877344</td>
<td>1278.88</td>
<td>360.30</td>
<td>83.24</td>
<td>836.34</td>
</tr>
</tbody>
</table>

Figure 5: The Xilinx Simulation result for booth-32 x 32 bit is exhibited below in the Figure 5, and then the structure level port-map model is synthesized as Gate-level Netlist for signal Transition calculation. Modified Booth's 64 x 64 bit simulation result on Altera Quartus-II is illustrated in the Figure 6, and then synthesis summary is depicted in Figure 7-11.
The Figure 7-plot graph Xilinx Area-Multiplier strength versus No. of LUT’s, figure 8- Graph plot for Xilinx Delay-Multiplier strength versus delay time (ns). Figure 9-plot for Altera Area-Multiplier strength versus No. of LUT’s figure and 10-Altera Delay-Multiplier strength versus delay time (ns).and finally figure 11 Graph plot for Altera Powerplay power-strength versus power dissipation (mW).

![Figure 7: Xilinx Area in No. LUT's](image1)

![Figure 8: Xilinx Delay in Nano sec's.](image2)

![Figure 9: Altera-Area in No. Logic Elements](image3)

![Figure 10: Altera-Delay in Nano sec's.](image4)

![Figure 11: Altera-Power Dissipation in mW](image5)

**V. Conclusion**

Our work has covered analysis of advanced signed multiplier architecture such as Baugh Wooley Multiplier and Modified Booth Encoder (MBE) Multiplier at various strength such as 4 x 4, 8 x 8, 16 x 16, 32 x 32 & 64 x 64 and the Result analysis with various VLSI Parameters like (Delay, Number of Logic Element requirements, Number of Signal Transition for particular sample input and its Power Consumption). As the Multiplier strength grows Area Curve shows a moderate difference while the delay performance of booth compared to that of Baugh wooley is approximately 4 times better (i.e., for 32 x 32 Baugh wooley needs ~325 ns while booth complete it with ~90 ns). Modified Booth proves great result in all forms of VLSI constraints and works effectively with desired specification needed for highly reliable RNS application and for further optimization Multi-Modulo Residue architecture are advisably wise choice. Thus Signed Booth multiplier performs superior than state of art multiplier and its efficiency can be utilized for further optimization of Multi-Modulo Residue architecture for all modulus in special moduli set.
References Références Referencias


This page is intentionally left blank