HIGH-SPEED 64-BIT BINARY COMPARATOR USING NEW APPROACH

Anjuli, Satyajit Anand
E&CE Department, FET-MITS, Lakshmangarh, Sikar, Rajasthan (India)

ABSTRACT

High-speed 64-bit binary comparator using new approach is proposed in this brief. Comparison is most basic arithmetic operation that determines if one number is greater than, equal to, or less than the other number. Comparator is most fundamental component that performs comparison operation. This brief presents comparison of modified and existing 64-bit binary comparator designs concentrating on delay. Means some modifications are done in existing 64-bit binary comparator design to improve the speed of the circuit. Comparison between modified and existing 64-bit binary comparator designs is calculated by simulation that is performed at 90nm technology in Tanner EDA Tool.

KEYWORDS: Binary comparator, digital arithmetic, high-speed.

I. INTRODUCTION

In digital system, comparison of two numbers is an arithmetic operation that determines if one number is greater than, equal to, or less than the other number [1]. So comparator is used for this purpose. Magnitude comparator is a combinational circuit that compares two numbers, A and B, and determines their relative magnitudes (Fig.1). The outcome of comparison is specified by three binary variables that indicate whether A>B, A=B, or A<B. The circuit, for comparing two n-bit numbers, has 2n inputs & 2^n entries in the truth table. For 2-bit numbers, 4-inputs & 16-rows in the truth table, similarly, for 3-bit numbers 6-inputs & 64-rows in the truth table [1].
In recent years, high-speed and low-power device designs have emerged as principal themes in the electronic industry due to increasing demand of portable devices. This tremendous demand is due to the popularity of battery operated portable equipment such as personal computing devices, wireless communication, medical applications, etc. Demand and popularity of portable electronic devices are driving the designers to strive for higher speed, smaller power consumption, and smaller area.

The logic style used in logic gates basically influences the speed, size, power dissipation, and the wiring complexity of a circuit [2]. Circuit size depends on the number of transistors and their sizes and on the wiring complexity [3]. The wiring complexity is determined by the number of connections and their lengths. All these characteristics may vary considerably from one logic style to another, and thus proper choice of logic style is very important for circuit performance [4].

In order to differentiate both the designs existing and modified, simulations are carried out for delay and power consumption with 1 volt input voltage (and supply voltage), 30°C temperature and 50MHz frequency at 90nm technology in Tanner EDA Tool.

II. 64-BIT BINARY COMPARATOR

64-bit binary comparator compares two numbers each having 64 bits (A to B). For this arrangement, the truth table has 128 inputs and 2128 entries. By using a comparator of minimum number of bits, a comparator of maximum number of bits can be designed [5],[6],[7] with the help of tree-based structure logic [8] and also with other useful logic styles.

III. EXISTING 64-BIT BINARY COMPARATOR DESIGN

64-bit comparator in reference [8],[9],[10] represents tree-based structure which is inspired by the fact that G (generate) and P (propagate) signal can be defined for binary comparisons, similar to G (generate) and P (propagate) signals for binary additions.

Two numbers (each having 2-bits: A and B) comparison can be realized by:

\[ B_{\text{Big}} = \overline{A_1}B_1 + (A_1 \oplus B_1)(\overline{A_0}B_0) \]  

\[ EQ = (A_1 \oplus B_1)(\overline{A_0} \oplus B_0) \]  

For \( A<B \), “\( B_{\text{Big}}, EQ \)” is “1,0”. For \( A=B \), “\( B_{\text{Big}}, EQ \)” is “0,1”. Hence, for \( A>B \), “\( B_{\text{Big}}, EQ \)” is “0,0”. Where \( B_{\text{Big}} \) is defined as output A less than B (A_LT_B). A closer look at equation (1)
reveals that it is analogous to the carry signal generated in binary additions. Consider the following carry generation:

\[ C_{\text{out}} = AB + (A \oplus B) \cdot C_{\text{in}} \]
\[ = G + P \cdot C_{\text{in}} \quad (3) \]

Where A & B are binary inputs \( C_{\text{in}} \) is carry input, \( C_{\text{out}} \) is carry output, and G & P are generate & propagate signals, respectively.

After comparing equations (1) & (3):

\[ G_1 = A_1B_1 \quad (4) \]
\[ \text{EQ}_1 = \overline{(A_1 \oplus B_1)} \quad (5) \]
\[ C_{\text{in}} = \overline{A_0B_0} \quad (6) \]

\( C_{\text{in}} \) can be considered as \( G_0 \). Since for static logic, equation (1) requires tall transistor stack height, hence, an encoding scheme is employed to solve this problem. For this, encoding equation is given as:

\[ G_{[i]} = \overline{A_{[i]}B_{[i]}} \quad (7) \]
\[ \text{EQ}_{[i]} = \overline{(A_{[i]} \oplus B_{[i]})} \quad (8) \]

Where \( i = 0 \ldots 63 \).

Put these two values from equations (7) & (8) in equations (1) & (2).

\[ B_{\text{Big}[2j+1: 2j]} = G_{[2j+1]} + \text{EQ}_{[2j+1]}\cdot G_{[2j]} \quad (9) \]
\[ \text{EQ}_{[2j+1: 2j]} = \text{EQ}_{[2j+1]}\cdot \text{EQ}_{[2j]} \quad (10) \]

Where \( j = 0 \ldots 31 \).

G & P signals can be further combined to form group G & P signals.

\[ B_{\text{Big}[3: 0]} = \overline{A_3B_3} + (A_3 \oplus B_3) \cdot (A_2B_2) + (A_3 \oplus B_3) \cdot (A_2 \oplus B_2) \cdot (A_1B_1) + (A_3 \oplus B_3) \cdot (A_2 \oplus B_2) \cdot (A_1 \oplus B_1) \cdot (A_0B_0) \]
\[ B_{\text{Big}[3: 0]} = \overline{A_3B_3} + (A_3 \oplus B_3) \cdot [A_2B_2 + (A_2 \oplus B_2) \cdot (A_1B_1 + (A_1 \oplus B_1) \cdot (A_0B_0))] \]
\[ B_{\text{Big}[3: 0]} = G_3 + \text{EQ}_3 \cdot \{G_2 + \text{EQ}_2 \cdot (G_1 + \text{EQ}_1 \cdot G_0)\} \]

\[ B_{\text{Big}[3: 0]} = B_{\text{Big}[3: 2]} + \text{EQ}_{[3: 2]} \cdot B_{\text{Big}[1: 0]} \quad (11) \]
\[ \text{EQ}_{[3: 0]} = \text{EQ}_{[3: 2]} \cdot \text{EQ}_{[1: 0]} \quad (12) \]

Similarly, for 64-bit comparator, \( B_{\text{Big}} \& \text{ EQ} \) can be computed as:

\[ B_{\text{Big}[63: 0]} = G_{63} + \sum_{k=0}^{62} \left( G_k \cdot \prod_{m=k+1}^{63} \text{EQ}_m \right) \quad (13) \]
\[ \text{EQ}_{[63: 0]} = \prod_{m=0}^{63} \text{EQ}_m \quad (14) \]
Fig. 2 shows 8-bit version of existing tree-based comparator structure and Fig. 3 - Fig. 5 shows corresponding circuit schematics for each logic block of each stage. Pre-encoding circuitry is aimed to minimize the number of transistors. Hence, modified pass transistor logic style is employed to reduce the number of transistors up to 9. In above 8-bit example circuitry, the first stage comparison circuit implements equations (9 & 10) for \( j = 0 \ldots 3 \), whereas the second stage generates \( B_{\text{Big}[3:0]} \), \( B_{\text{Big}[7:4]} \) and \( EQ_{[3:0]} \), \( EQ_{[7:4]} \) according to equations (11 & 12). Finally, \( B_{\text{Big}[7:0]} \) and \( EQ_{[7:0]} \) are computed in third stage according to equations (13 & 14).

![Figure: 2 Tree-Diagram of 8-Bit Binary Comparator](image)

Stage 0\(^{th}\) is implemented using modified pass transistor logic style giving output in actual form, Stage 1\(^{st}\) is implemented using CMOS logic style giving output in inverse form, Stage 2\(^{nd}\) is also implemented using CMOS logic style but giving output in actual form.

64-bit comparator is here designed by using 7 stages (from 0\(^{th}\) to 6\(^{th}\)). In stage 0\(^{th}\), modified pass transistor logic style circuitry (as in Fig. 3) is employed to produce “less than” & “equal to” outputs. An output of stage 0\(^{th}\) act as inputs of stage 1\(^{st}\) in stage 1\(^{st}\), CMOS circuitry (as in Fig. 4) is employed to produce inverse inputs for stage 2\(^{nd}\). In stage 2\(^{nd}\), again CMOS circuitry (as in Fig. 5) is employed to produce actual inputs for stage 3\(^{rd}\). Now, according to tree structure given in Fig. 2, again circuitry of stage 1\(^{st}\) is used for stage 3\(^{rd}\). Similarly, for stage 4\(^{th}\), circuitry of stage 2\(^{nd}\) is employed. For stage 5\(^{th}\) circuitry of stage 1\(^{st}\) is employed. For stage 6\(^{th}\) circuitry of stage 2\(^{nd}\) is employed. Description of this design is given in tabular form in Table I.

![Figure: 3 Schematic of Stage 0\(^{th}\) of Existing 64-Bit Binary Comparator](image)
According to [8] existing design is having two outputs (“A less than B” & “A equal to B”). This research work also represents here two outputs but they are “A less than B” and “A greater than B”. Means “A greater than B” output is here calculated in place of “An equal to B” output. For this arrangement, an extra circuitry of NOR gate (which is shown in Fig. 6) is included at the end of schematic of existing 64-bit binary comparator design. Outputs of “A less than B” & “A equal to B” are given to two inputs of NOR gate that produces “A greater than B” output. Existing design requires 1210 transistor count for 64-bit binary comparator. Accordingly schematic of Existing 64-bit binary comparator is drawn and shown in Fig. 7.
According to input bit stream, waveforms of existing 64-bit binary comparator are obtained and shown in Fig. 8. Waveforms show that only one output is high (“1”) at a time. When both the outputs “less than” & “greater than” (A_LT_B & A_GT_B) are low (“0”), then waveforms represent that “equal to” output is high (A_EQU_B is “1”) at that time. Simulation results for this design are given in Table III – Table V for conclusion.

IV. MODIFIED 64-BIT BINARY COMPARATOR DESIGN

Some modifications have been done for two basic stages (1st and 2nd) in existing 64-bit binary comparator design [8] to improve the speed of the circuit. For this design, stage 0th is same as existing 64-bit comparator design & implemented using modified pass transistor logic style (Fig. 3) giving output in actual manner. Stage 1st is also implemented using modified pass transistor logic style (MPTL) giving output in actual manner as in Fig. 9.
idea behind PTL (pass transistor logic) is to use purely NMOS pass transistors network for logic operation [5]. The basic difference of pass-transistor logic style compared to the CMOS logic style is that the source side of the logic transistor networks is connected to some input signals instead of the power lines. In this design style, transistors act as switch to pass logic levels from input to output [4]. But purely NMOS pass transistors network does not provide full output voltage swing. Due to this reason modified pass transistor logic style (MPTL) have been used for stage 0\textsuperscript{th} and stage 1\textsuperscript{st}. MPTL means extra PMOS circuitry is used in pass transistor logic style circuitry to pass logic high (“1”) from input to output. Stage 2\textsuperscript{nd} is same as stage 1\textsuperscript{st} of existing 64-bit comparator design & implemented using CMOS logic style but giving output in inverse manner as in Fig. 10. Description of this design is given in tabular form in Table II.

![Figure:9 Schematic of Stage 1\textsuperscript{st} of Modified 64-Bit Binary Comparator](image)

![Figure: 10 Schematic of Stage 2\textsuperscript{nd} of Modified 64-Bit Binary Comparator](image)

Existing 64-bit binary comparator design [8] follows tree-based structure from 2-bit to 64-bit circuitry. But modified design follows tree-based structure from 2-bit to 8-bit circuitry only. After 8-bit to 64-bit circuitry, modified design follow simple logic structure in place of tree-based structure. In modified design, both the outputs of eight (from 0\textsuperscript{th} to 7\textsuperscript{th}) 8-bit comparators are given to 8\textsuperscript{th} 8-bit comparator to produce final outputs (“less than” and “greater than”). A less than B outputs of 0\textsuperscript{th} to 7\textsuperscript{th} 8-bit comparators are given to $A_{0:7}$ inputs of 8\textsuperscript{th} 8-bit comparator. A equal to B outputs of 0\textsuperscript{th} to 7\textsuperscript{th} 8-bit comparators are given to $B_{0:7}$ inputs of 8\textsuperscript{th} 8-bit comparator that produces final outputs.
Since output of 8-bit comparators are obtained in inverse form. So, at the end of schematic design of modified 64-bit comparator two inverters (Fig. 11) are required to produce actual form of output waveform. This design requires 1732 transistor count for 64-bit comparator. Schematic (using instances of each section) of modified 64-bit binary comparator design is drawn and shown in Fig. 12.

![Figure 12: Schematic of Modified 64-Bit Binary Comparator](image)

**Figure: 12** Schematic of Modified 64-Bit Binary Comparator

![Figure 13: Waveforms of Modified 64-Bit Binary Comparator](image)

**Figure: 13** Waveforms of Modified 64-Bit Binary Comparator
According to input bit stream, waveforms of modified 64-bit binary comparator are obtained and shown in Fig. 13. Input bit stream for modified design is same as in existing design of 64-bit comparator. Output waveforms of modified design produce same position of 1,s and 0,s as in waveforms of existing design for each input bits. Waveforms show that only one output is high (“1”) at a time. When both the outputs “less than” & “greater than” (A_LT_B & A_GT_B) are low (“0”), then waveforms represent that “equal to” output is high (A_EQU_B is “1”) at that time. Simulation results for modified 64-bit binary comparator design are given in tabular form in Table III – Table V.

V. SIMULATION AND COMPARISON

After simulation of both the designs final results are obtained for delay and power consumption and are shown in Table III – Table V. Simulations have been carried out at 90nm technology in Tanner EDA Tool.

Table I. Description of Existing 64-Bit Binary Comparator design

<table>
<thead>
<tr>
<th>Detail</th>
<th>Stage 0th</th>
<th>Stage 1st</th>
<th>Stage 2nd</th>
<th>Transistor Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design</td>
<td>Using MPTL Style</td>
<td>Using CMOS Style</td>
<td>Using CMOS Style</td>
<td>1210</td>
</tr>
<tr>
<td>Nature of output</td>
<td>Actual</td>
<td>Inverse</td>
<td>Actual</td>
<td></td>
</tr>
</tbody>
</table>

Table II. Description of Modified 64-Bit Binary Comparator Design

<table>
<thead>
<tr>
<th>Detail</th>
<th>Stage 0th</th>
<th>Stage 1st</th>
<th>Stage 2nd</th>
<th>Transistor Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design</td>
<td>Same as Existing</td>
<td>Using MPTL Style</td>
<td>Same as stage 1st of Existing</td>
<td>1732</td>
</tr>
<tr>
<td>Nature of output</td>
<td>Actual</td>
<td>Actual</td>
<td>Inverse</td>
<td></td>
</tr>
</tbody>
</table>

Table III. Simulation Data with 1volt Input Voltage

<table>
<thead>
<tr>
<th>Design</th>
<th>Power Consumption (watt)</th>
<th>Delay Time (second)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>t_A_LT_B</td>
</tr>
<tr>
<td>Existing</td>
<td>9.0675e-6</td>
<td>4.4240e-9</td>
</tr>
<tr>
<td>Modified</td>
<td>1.3577e-5</td>
<td>4.2155e-9</td>
</tr>
</tbody>
</table>

Table IV. Simulation Data with 30°C Temperature

<table>
<thead>
<tr>
<th>Design</th>
<th>Power Consumption (watt)</th>
<th>Delay Time (second)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>t_A_LT_B</td>
</tr>
<tr>
<td>Existing</td>
<td>9.2485e-6</td>
<td>4.4187e-9</td>
</tr>
<tr>
<td>Modified</td>
<td>1.3833e-5</td>
<td>4.2144e-9</td>
</tr>
</tbody>
</table>
Table V. Simulation Data with 50MHz Frequency

<table>
<thead>
<tr>
<th>Design</th>
<th>Power Consumption (watt)</th>
<th>Delay Time (second)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$t_{A_{LT,B}}$</td>
</tr>
<tr>
<td>Existing</td>
<td>9.2765e-6</td>
<td>4.4240e-9</td>
</tr>
<tr>
<td>Modified</td>
<td>1.3566e-5</td>
<td>4.2156e-9</td>
</tr>
</tbody>
</table>

After simulation of both the designs final results are obtained for delay and power consumption with 1 volt input voltage. Delay comparison of modified and existing 64-bit comparator designs is shown in Fig. 14 & Fig. 15. Simulated data for these graphs is given in Table III.

The graphs shown in Fig. 14 & Fig. 15 reveal that delay of modified 64-bit comparator design at 1 volt input voltage is remarkably reduced than existing 64-bit comparator design. In Fig. 14, delay is reduced 4.7%. In Fig. 15, delay is reduced 1.6%.

After simulation of both the designs final results are obtained for delay and power consumption with 30°C temperature. Simulation with temperature has been done at 1 volt input voltage. Delay comparison of modified and existing 64-bit comparator designs is shown in Fig. 16 & Fig. 17. Simulated data for these graphs is given in Table IV.
The graphs shown in Fig. 16 & Fig. 17 reveal that delay of modified 64-bit comparator design at 30°C temperature is remarkably reduced than existing 64-bit comparator design. In Fig. 16, delay is reduced 4.6%. In Fig. 17, delay is reduced 1.6%. After simulation of both the designs final results are obtained for delay and power consumption with 50MHz frequency. Simulation with frequency has been done at 1 volt input voltage. Delay comparison of modified and existing 64-bit comparator designs is shown in Fig. 18 & Fig. 19. Simulated data for these graphs is given in Table V.

The graphs shown in Fig. 18 & Fig. 19 reveal that delay of modified 64-bit comparator design at 50MHz frequency is remarkably reduced than existing 64-bit comparator design. In Fig. 18, delay is reduced 4.7%. In Fig. 19, delay is reduced 1.6%.

**VI. CONCLUSION**

In modified design, at 1 volt input voltage delay for output “A less than B” ($t_{A,LT,B}$) is reduced 4.7% and delay for output “A greater than B” ($t_{A,GT,B}$) is reduced 1.6% in comparison to existing design. Similarly, at 30°C temperature delay for output “A less than B” ($t_{A,LT,B}$) is reduced 4.6% and delay for output “A greater than B” ($t_{A,GT,B}$) is reduced 1.6%. And also at 50MHz frequency delay for output “A less than B” ($t_{A,LT,B}$) is reduced 4.7% and delay for output “A greater than B” ($t_{A,GT,B}$) is reduced 1.6% in comparison to existing design. Hence, superiority of modified design is maintained for temperature and frequency also. All of the reduction in delay is obtained after sacrificing power consumption and transistor count. But still modified design gives better result (for delay) than existing design. Therefore, modified 64-bit binary comparator design can be better option for high-speed applications.

**REFERENCES**


