Synthesizable Time Measurement and Test Scheme for SoC Architecture

M. Amir Abas
Centre for Research and Postgraduate Studies, University Kuala Lumpur, Malaysia
drmamir@unikl.edu.my

Abstract - This paper presents a Synthesizable high-resolution time measurement and test scheme for digital System on Chip (SoC) application namely Two-Delay Interpolation Method (TDIM). The scheme is designed to measure internal timing parameters in SoC architecture such as jitter, set-up and hold time, delay faults and etc. Simulation result shows the circuit is capable to measure as low as 5 ps timing interval and the range of measurement is programmable using programmable logic core in SoC. The small size of the design makes it eminently suitable for SoC applications, particularly when the accurate measurement of small time intervals are required; a need which is increasing as operational speeds tend toward 100GHz.

Keywords: Time Measurement, Interpolation, MUTEX, Programmable

1.0 Introduction

The new paradigm of system on a chip (SoC) has changed the approach to design and testing [5]. SoC consists of several cores and mixed analog-digital circuits. Each core needs to be tested whether using on-chip or off-chip strategies. One of the categories of fault that is of concern in SoC, and which needs critical research, is timing faults. They are very difficult to measure, even by using sophisticated automatic test equipment (ATE). Timing faults require a high-resolution mechanism to test and measure the value in order to diagnose the fault accurately. In this paper, two designs of time measurement circuits (TMC) are proposed. The two circuits employ Mutual Exclusion (MUTEX) elements and delay lines as the main components in measuring timing intervals. In general the operation of the time measurement is accomplished by applying two rising edges to two chains of delay lines. The measurement begins when the first edge goes high and its rising edge propagates in the first chain of delay line followed by the second rising edge propagates in the second chain of delay line. The first delay line has different value of delay compared to the second delay line. As a result this enables a series of MUTEX to digitize the timing between the two edges during the propagation of the two edges in the delay lines.

It is an issue that ensuring a design is error-free has become virtually impossible. This may be due to design errors that are not detected during development and manufacturing or simply due to changes in the design requirements. A partial solution to this problem is to reduce the number of respins (redesigns due to errors) per design, thereby reducing the costs. Embedded programmable logic cores in SoC are increasingly being seen as an attractive way to provide post-fabrication flexibility to an integrated circuit. A programmable logic core is a flexible logic fabric that can be customized to implement any digital circuit after fabrication [7]. Before fabrication, the designer embeds a programmable fabric (consisting of many uncommitted gates and programmable interconnects between the gates). After the fabrication, the designer can then program these gates and the connections between them.

All the design procedures and the operations of the above circuits are described in detail. This paper starts with an explanation of the main components of the TMCs comprising MUTEX, TDIM, Delay Components, Code Converter and finally the proposed TMC circuit for programmable logic core. At the end of this paper an analysis of circuits is given in order to verify the performance of the measurement process.

2.0 MUTEX Arbiter

The implementation of the TMCs is initiated with the operation of a MUTEX arbiter circuit. A MUTEX is commonly used in an asynchronous design environment but in this particular context a MUTEX is introduced to compare two rising signals and determines which arrives first at the two inputs. A MUTEX circuit, shown in Figure 1.a, has two inputs a bistable comprising NAND gates (or
comprising NOR gates for active low inputs) and a metastability filter circuit (Figure 1.a). Normally the two input signals are low. When the two rising signals arrive at the inputs with a small timing difference, the MUTEX will recognize which one arrived first. If the first signal is faster than the second signal, output 1 will change to high. On the other hand if the second signal is faster than the first signal, output 2 will change to high.

![MUTEX circuit diagram](image1)

Figure 1. MUTEX circuits and timing diagram

The concept of the operation is based on the difference between two delay lines. This technique has been proposed in [4] or implementing stabilized CMOS delay line applications. However, the structure of the design was not described in detail. In our design structure, the main block comprises K cells and each cell consists of two delay elements and a MUTEX. The two delay elements are made from a pair of inverters as in the TDM circuit design. In setting the delay value, the first inverter pair of the delay element has a structure slightly smaller than the second inverter pair. As a result, the first delay element contributes a delay, $\Delta \tau$, longer than the second delay element. If the time, $\Delta \tau$, is known, each cell would delay the first signal by $\Delta \tau$ while the second signal is not delayed. The process of measuring two rising signals is started with the faster signal which must go to the longer delay element. Therefore the first signal will be delayed by $\Delta \tau$ for K number of times. Each time the test signals pass through one cell the output is tested to check the result of the difference between the two inverter pairs. A positive difference will indicate that the first test signal is faster than the second test signal, while a negative difference will define matters the other way round. Eventually the complete result of the measurement is in the form of a thermometer code.

4.0 TDIM Circuit Design

The design process of the TDIM begins with the evaluation strategy for the main component, which is the delay line structure. Different sizes of delay line are simulated to get the fine resolution of the cell. Next two design strategies are proposed, which are a 32-bit design structure and a synthesizable design structure. 32-bit design structure is a standard TDIM for any digital circuit while synthesizable design structure could be applied to a programmable IC package such as a FPGA, but offers less resolution time measurement. Finally, the following section describes the simulation work for a 16-bit design structure using 0.18 $\mu$m CMOS technology. Several characteristics of the TDIM are studied in order to analyze its time measurement accuracy, to check the consistency of the output against the effects of external parameter variations such as temperature, voltage supply and imperfections in design layout, and to get an estimation of size for a design layout.
5.0 Delay Resolution of the Inverter Circuit

The design of the CMOS inverter is one the most important issues in the context of this application. The delay difference of two inverter pairs will determine the resolution of the TMC. An investigation has been carried out to analyze the accuracy of the delay for different sizes of transistor width in the inverter circuit structure. Figure 3.a shows the basic structure of one inverter pair. The first inverter in the circuit is used as a target gate setting the delay difference while the structure for the second inverter is maintained. Five different transistor widths were used in the investigation as depicted in Figure 3.b and the simulation outputs of the respective transistor widths are shown in Figure 4. It is clearly shown that the delay is increased when the size of the inverter is reduced. Table 1 shows the delay value for every size being tested. The delay differences between two inverter pairs were calculated by subtracting the delay for one inverter pair from that of the next inverter pair. The results show that the delay difference could reach as low as 1ps. This is significant in increasing the resolution of the TMC.

6.0 32-Bit TDIM Design

Figure 5 shows the proposed block diagram of a 32-bit TDIM. According to the analysis of the inverter pair delay structure, the sizes of \(15\lambda/2\lambda, 5\lambda/2\lambda\) and \(12\lambda/2\lambda, 4\lambda/2\lambda\) were chosen because the delay (5ps) is much closer to the minimum range requirement of the TMC at \(\leq 5ps\) [6].

Due to the measurement being based on successive operations, the 32-bit output would be in the form of a thermometer code. The completion time of one measurement is equal to the total number of inverters.
Hence for 32-bit operation it would be $32 \times 2\tau = 64\tau$ which is equivalent to 2.56ns if the propagation delay of inverter for a 0.18 \mu m CMOS process is for example 40ps. A coding converter is needed to convert the thermometer code into binary code. Finally the capture register would capture the result of the time measurement for the next analysis operation.

The operation of the TMC can be summarized as follows:

1. Activate reset signal for all the capture registers. This should be done for every measurement that needs to be carried out.
2. Calibrate the TMC to determine the resolution of one cell. An averaging technique can be introduced to determine the resolution of one cell.
3. Activate the test signals. The first rising test signal must be received by the first input of the TMC (IN1).
4. The coding converter circuit will convert the thermometer code to equivalent binary code.
5. The capture registers will read the results from the coding converter.
6. Finally the captured result can be read in parallel or can be scanned out depending on the needs of the application.

7.0 Thermometer to Binary Code Converter

In the implementations of the TDIM designs, generate the results of time measurement in thermometer code format. A thermometer to binary code converter (TBCC) is required to convert the result to binary equivalent. The circuit of the converter was designed using combinational logic gates as shown in Figure 6.
8.0 Synthesizable Design Structure

As shown in Figure 7, NAND and NOT gates are used to implement the delay element. The NAND gate should have a higher propagation delay time than the NOT gate. For the detector element, two types of MUTEX are used which are made from NAND or NOR gates. The two MUTEXes are placed alternately as the test signals are inverted every time they pass through each delay element. The size of one cell comprises 18 transistors which can be easily constructed using programmable logic core in SoC.

9.0 Simulation Analysis

The block diagram of Figure 5 was simulated using a Pspice simulator. All the dimensions and transistor sizes as proposed in Table 1 were used. However, the number of cells has been reduced to 16 from 32 due to the limitation in the number of variables in the simulation process. The circuit of the thermometer to binary code converter as described in section 7 was implemented in this simulation. Several timing intervals were applied to analyze the performance of the TDIM. Two examples of the simulation results, Example 1 and Example 2, show the comparison of two results between thermometer code generated and the prediction obtained from the input timing interval over the resolution of the TDIM. In the thermometer code, the first MUTEX output is not considered as no delay is applied to the test signals. In the circuit design the purpose of the first MUTEX output is to show that the first test signal is faster than the second test signals.

Table 2: Truth table of 16-bit thermometer to binary code converter

<table>
<thead>
<tr>
<th>Thermometer Code</th>
<th>Binary Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1</td>
<td>0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1</td>
<td>0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1</td>
<td>0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1</td>
<td>0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1</td>
<td>0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0</td>
</tr>
<tr>
<td>1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1</td>
<td>1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
</tr>
</tbody>
</table>

Fig. 6: Thermometer to binary converter

Fig. 7: Proposed synthesizable design structure

Table 3: Propagation delay time of test signals through each cell

<table>
<thead>
<tr>
<th>Test Sig.</th>
<th>Cell Delay (ps)</th>
<th>Difference (ps)</th>
<th>MUTEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>72.8</td>
<td>1 69.2</td>
<td>3.4</td>
<td>1</td>
</tr>
<tr>
<td>64.2</td>
<td>2 69.2</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>59.6</td>
<td>3 59.6</td>
<td>4.6</td>
<td>1</td>
</tr>
<tr>
<td>54.5</td>
<td>4 54.5</td>
<td>5.1</td>
<td>1</td>
</tr>
<tr>
<td>49.9</td>
<td>5 49.9</td>
<td>4.6</td>
<td>1</td>
</tr>
<tr>
<td>44.9</td>
<td>6 44.9</td>
<td>5</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 2: Truth table of 16-bit thermometer to binary code converter
resolution of 5ps is successfully maintained by each cell which is closely matched with the expected results. The histogram analysis of Figure 8 shows a very low standard deviation of 0.65 for the average resolution of 5.01ps.

![Histogram analysis of cells in TDIM](image)

Fig. 8: Histogram analysis of cells in TDIM

In order to check the consistency of the outputs in the next simulation stage, three parameters were introduced which are strongly related to the output operations. The first parameter is the imperfections during fabrication. The NMOS transistors in cells 5 and 11 were randomly chosen to have imperfection defects by increasing their width with 10% than normal size. Figure 11 shows the results of the simulation. By using the same timing interval of 72ps as in Example 1, the thermometer code generated remained at 14. This means that the increase of 10% has minor effect on the average TMC outputs.

**Example 1.**

Input timing test 42ps. Resolution/cell = 5ps. Output = 1111111111111100 (Decimal=14)

Expected Output = \[
\frac{actual\ time}{resolution} = \frac{42\ ps}{5\ ps} = 8
\]

![Example 1 simulation result](image)

Fig. 9: Example 1 simulation result

**Example 2**

Input Timing Test 72ps. Resolution/cell = 5ps. Output = 1111111111111100 (Decimal =14)

Expected Output = \[
\frac{actual\ time}{resolution} = \frac{72\ ps}{5\ ps} = 14
\]

![Example 2 simulation result](image)

Fig. 10: Example 2 simulation result

The next simulation stage was to introduce temperature variations to the TDIM. The temperature range was varied from 20°C to 47°C. The results shows that the thermometer code
remained at 14 except at temperature 20°C which produce 15 as depicted in Figure 12. Figure 13 shows similar simulation but using a 50ps time interval. The outputs were probed from “thermometer to binary code” converter.

![Simulation results with voltage supply variations](image)

**Fig. 13:** Simulation results with (a) room temperature at 27°C, (b) extreme temperature at 47°C

Two different temperatures, 20°C and 47°C, were used and the results give the same responses as the first investigation. The TDIM generates binary code one bit higher at temperature 20°C (Figure 13.a) than at temperature 47°C (Figure 13.b). The last stage of the simulation was to investigate the effect of power supply variations to the TDIM. The supply voltage was varied for the range of 1.4V to 2.2V.

The results shown in Table 4 indicating that the thermometer generated are affected by the changes of the voltage supply. This is certainly due to the propagation delay of gate which is increased when the voltage supply is reduced. As a result the resolution of each cell of TDIM becomes bigger when it is supplied with low voltage supply. Therefore it is important to keep the voltage supply within the specification range for this particular application.

![Timing input = 72ps](image)

Table 4: Simulation results with voltage supply variations from 1.4V to 2.2V

<table>
<thead>
<tr>
<th>Vdc</th>
<th>Output (T. Code)</th>
<th>Binary Equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.4</td>
<td>11111111100000</td>
<td>01011</td>
</tr>
<tr>
<td>1.6</td>
<td>11111111110000</td>
<td>01010</td>
</tr>
<tr>
<td>1.8</td>
<td>11111111111000</td>
<td>01110</td>
</tr>
<tr>
<td>2</td>
<td>11111111111110</td>
<td>10000</td>
</tr>
<tr>
<td>2.2</td>
<td>11111111111111</td>
<td>10000</td>
</tr>
</tbody>
</table>

*Timing input = 72ps

10.0 Conclusion

In this paper, we have described the concept of the time measurement circuit which is effective for measuring internal timing parameters in SoC architecture. There are two options in implementation stage that could be embedded closely at test point during fabrication or post-fabrication using programmable logic core in SoC which is carried out after the fabrication process. The synthesizable feature of the circuit is very useful for certain extent particularly during trial run stage. This could avoid respins issue which often cost huge of dollars.

References