Design for Delay Measurement Aimed at Detecting Small Delay Defects on Global Routing Resources in FPGA**

Kazuteru NAMBA†, Member, Nobuhide TAKASHINA†∗, Nonmember, and Hideo ITO†, Fellow

SUMMARY Small delay defects can cause serious issues such as very short lifetime in the recent VLSI devices. Delay measurement is useful to detect small delay defects in manufacturing testing. This paper presents a design for delay measurement to detect small delay defects on global routing resources, such as double, hex and long lines, in a Xilinx Virtex 4 based FPGA. This paper also shows a measurement method using the proposed design. The proposed measurement method is based on an existing one for SoC using delay value measurement circuit (DVMC). The proposed measurement modifies the construction of configurable logic blocks (CLBs) and utilizes an on-chip DVMC newly added. The number of configurations required by the proposed measurement is 60, which is comparable to that required by stuck-at fault testing for global routing resources in FPGAs. The area overhead is low for general FPGAs, in which the area of routing resources is much larger than that of the other elements such as CLBs. The area of every modified CLB is 7% larger than an original CLB, and the area of the on-chip DVMC is 22% as large as that of an original CLB. For recent FPGAs, we can estimate that the area overhead is approximately 2% or less of the FPGAs.

key words: small delay defects, delay measurement, DVMC (delay value measurement circuit), FPGA (field programmable gate array), global routing resource

1. Introduction

Small delay defects can come to be a serious issue in the recent high-density VLSI devices [1]. This is because they are caused by unintended reasons such as resistive-short, resistive-open and resistive-via. Thus they can lead to vulnerability to transistor aging bringing about very short lifetime even though they do not affect system operations just after manufacturing. Small delay defects are desired to be detected in manufacturing testing. However, they are not detected by traditional manufacturing testing such as stuck-at fault, transition and path delay fault testing. To detect small delay defects, some previous works presented testing using delay measurements for systems on a chip (SoCs) [2]–[4]. In these, path delay times are measured. If a path delay time is beyond a specified time such as the three-sigma limit, even if it is not beyond the clock period, it is regarded as faulty. The measurement used in [2] utilizes a variety of test clock frequencies, all of which are higher than the system clock frequency. However, measurements using a variety of test clock frequencies take a long time because this measures a path many times changing test clock frequencies. The authors’ group proposed a delay measurement using an embedded delay value measurement circuit (DVMC) [3], [4]. This measures a path delay time at one time, and thus a time required for measurements is shorter than that using a variety of test clock frequencies. While there are other classes of delay measurements, these are not suited for this purpose. For example, measurements using a Vernier delay line (VDL) are well-known [5], [6]. They are useful for several purposes such as analysis of process variation, the results of which are available for improvement of fabrication process and enhancement of manufacturing yield. However, in use of manufacturing testing, they require very large number of redundant lines to feed the input and output signals of measured paths.

Small delay defects have also serious negative impacts on field programmable gate arrays (FPGAs), and thus they should be detected in manufacturing testing. In addition, we can improve manufacturing yield by using delay measurements. This is because we can change circuit structure on FPGAs after manufacturing, considering actual path delay times [7]. Therefore, delay measurements detecting small delay defects occurring on FPGAs are strongly desired. There are some previous works providing delay measurements for elements in FPGAs such as routing resources and look up tables (LUTs) primarily aiming to analyze process variation [8]–[12]. The papers [8]–[10] proposed measurements using ring oscillators. These methods measure the delay time of looped paths. However, they do not measure the delay time of paths not looped, which are actually used. The measurements presented in [11], [12] use a test clock generator, which generates test clocks with various frequencies. However, such measurements with a variety of test clock frequencies take much longer time than measurements with a DVMC as mentioned above.

This paper presents a design for delay measurement aimed at detecting small delay defects occurring on global routing resources on FPGAs, and a delay measurement method using the proposed design. The proposed design and measurement target a Xilinx Virtex 4 based FPGA [13], [14]; which has double, hex and long lines as global routing resources. This work is a class of design for test, in which some features are added to devices at design phase [15]. The proposed measurement is based on an existing measurement using a DVMC [3].

This paper is organized as follows: Section 2 intro-
duces the existing measurement [3] and the target FPGA. The proposed design and measurement are explained in Sect. 3. Sections 4 and 5 present the evaluation and conclusion, respectively.

2. Preliminary

2.1 Delay Measurement for SoC

Figure 1 illustrates the outline of the delay measurement circuit that the authors’ group proposed in [3]. This circuit measures the path delay time of a path under measurement \( p \). This circuit contains a DVMC and a stop signal generator (SSG), both of which are on-chip. The SSG detects changes of the input value of a predesignated FF. The DVMC measures a time lapse after the clock edge just after rising of the start signal and before the change of the stop signal.

To measure the delay time of \( p \), we measure the delay times of the following two paths, which are denoted by broken lines in Fig. 1:

- clock signal CLK \( \rightarrow \) output of FF \( i \) \( \rightarrow \) path \( p \) \( \rightarrow \) input of FF \( j \) \( \rightarrow \) SSG \( \rightarrow \) DVMC
- clock signal CLK \( \rightarrow \) output of FF \( j \) \( \rightarrow \) SSG \( \rightarrow \) DVMC

By calculating the difference of the delay times of the above two paths, we can obtain a time close to the delay time of \( p \). The difference between the obtained value and the actual delay time of \( p \) is lead by the clock tree and the lines from the FFs to the SSG.

Figure 2 illustrates the construction of the DVMC, which consists of a ring oscillator, an asynchronous counter and a register. At the clock edge just after rising of the start signal, oscillation starts in the oscillator. The asynchronous counter counts the number of oscillation. At the change of the stop signal, the register captures the state of the oscillator and the counter, which indicates the time lapse. The captured data are scanned-out and observed at an external device. The difference of measured delay times are calculated at the external device. Scheme for the scan-out is omitted in this figure, which is shown in [3].

Some measurement errors can occur on the measurement system due to several reasons such as process variations in the system. However, the errors have only a low effect on the result of manufacturing testing using the measurement system, which is discussed in [3].

2.2 Target FPGA

The proposed method targets a Xilinx Virtex 4 based FPGA [13], [14]. Figure 3 illustrates a construction of the target FPGA. The FPGA is an island-style FPGA, which has an array of switch blocks (SBs). Every SB is connected to a configurable logic block (CLB). Some SBs are connected to input/output blocks (IOBs). SBs are connected to each other through global routing resources.

Global routing resources are categorized into the following three according to the number of SBs that every line spans: double, hex and long lines. Figure 4 illustrates these
lines. Every double line connects a SB \( SB \) to two neighboring SBs \( SB_{i+1} \) and \( SB_{i+2} \). Every hex line connects SB \( SB \) to \( SB_{i+3} \) and \( SB_{i+6} \). Every long line connects \( SB \) to \( SB_{i+6} \), \( SB_{i+12} \), \( SB_{i+18} \) and \( SB_{i+24} \). Double and hex lines are unidirectional. Long lines are partially bidirectional. Specifically, long lines input from and output to \( SB \) and \( SB_{i+24} \), while they only output to \( SB_{i+6} \), \( SB_{i+12} \) and \( SB_{i+18} \). There are four directions for double and hex lines: north, east, south and west directions. Every SB can be a source, \( SB \), for ten double lines for every direction. In sum, every SB can be a source for 40 double lines. Similarly, every SB can be a source for 40 hex lines. There are two directions for long lines: vertical and horizontal directions. Every SB can be a source, i.e. \( SB_{i}/SB_{i+24} \), for five long lines for every direction, and totally for ten long lines.

Figure 5 illustrates the outline of a CLB. Every CLB has four slices. Every slice has two basic logic elements (BLEs). Figure 6 shows the construction of a BLE. Every BLE consists of a 4-input LUT, a FF and some logic gates. The FF has three inputs: a normal input, a clock enable CE and a clock signal CLK. When CE = 0, the output value of the FF does not change regardless of CLK.

3. Proposed Design for Delay Measurement

This section provides a design for delay measurement and delay measurement method using the proposed design aimed at detecting small delay defects on global routing resources in the Xilinx Virtex 4 based FPGA described in 2.2. The proposed design uses some extra circuits added for the measurement. The construction of the FPGA including the extra circuits is explained in 3.1. The proposed delay measurement for a line under measurement is also described in 3.1. To detect small delay defects in manufacturing testing, the path delay times of all global routing resources should be measured. Just like traditional testing for FPGAs, in the proposed measurement, the target FPGA is configured with specified configurations, and then some stimuli are supplied and results are observed. The specified configurations for measurement of all global routing resources are discussed in 3.2 and 3.3. In 3.2, the outline of the proposed configurations is explained. Because a configuration time for an FPGA is long, the small number of configurations is desirable. The proposed configurations are designed to provide a small number. The point for the small number is presented in 3.2. In 3.3, the detail placement and routing in the proposed configuration is presented.

3.1 Proposed Method for a Line under Measurement

Figure 7 illustrates the construction of the BLE in the proposed design. We can obtain this BLE by modifying the original BLE shown in Fig. 6. Specifically, an extra 2:1 multiplexer (MUX) is added at the output of the BLE. One of the inputs of the MUX is connected to the original output of the BLE, and the other is connected to the input of the FF through an extra line newly added.

Figure 8 shows the outline of the proposed measurement for a line under measurement. The paths denoted by heavy lines are measured. The line \( l \) in Fig. 8 (b) is a double, hex or long line under measurement. The proposed measurement is based on the existing measurement introduced in 2.1. Just like the existing measurement, the proposed measurement measures the delay times of two paths, a path including \( l \) and the other not including \( l \). After this, we can obtain the delay time of \( l \) (to be exact, a time close to the delay time of \( l \)) by calculating their difference. This circuit has two slices, a sender slice and a receiver slice. The sender and receiver slices are connected to \( l \) as its input and output, respectively. While every slice has two BLEs as described in 2.2, only one BLE is drawn in each slice for simplicity. This circuit has a FF, a SSG and a controller as well as the two slices, which are constructed on the FPGA. In addition, it has an on-chip DVMC. The DVMC is not constructed on the FPGA but newly added for this measurement. This measurement system uses the DVMC shown in [3].

First we discuss measurement for a rising transition \( 0 \rightarrow 1 \) on the heavy line in Fig. 8 (a). The FF in the receiver slice and the FF labeled as “transition” are initialized.
as 0 and 1; the clock enable is activated. The output value 1 of the transition FF is propagated to the input of the FF in the receiver slice through the LUT and MUX in the slice, which is denoted by a heavy broken line. At a clock edge, the value of the FF in the receiver slice changes 0 → 1, i.e., a rising transition occurs. The transition reaches the DVMC through the output of the receiver slice and the SSG. The DVMC regards this transition as a stop signal, and then it provides the lapse time from the clock edge to this reaching of the transition. We can measure the delay time for a falling transition in a similar manner.

We can make measurement of the path shown in Fig. 8(b) in a similar manner. At a clock edge, a transition occurs at the FF in the sender slice. The transition is propagated to the input of the receiver slice through the MUXes in the sender slice and the line \( l \), and then it is propagated to the DVMC through the receiver slice and the SSG. In the receiver slice, the transition passes through the LUT, the extra line and the extra MUX bypassing the FF.

3.2 Configurations

Here, the outline of the configurations for the proposed method is explained. The configurations enable us to measure all global routing resources aimed at detecting small delay defects on the resources in manufacturing testing. Because a configuration time for an FPGA is long, the small number of configurations is desirable. To achieve a small number, the configurations are designed such that many lines are measured for each configuration.

First, measurement of paths including \( l \) shown in Fig. 8(b) is discussed. In Fig. 8(b), the LUT in the receiver slice is transparent. Figure 9 illustrates the configuration of the BLE in a receiver slice. The 4-input LUT is configured as a 2:1 MUX. This receiver slice can be used for measurements of two lines the outputs of which are connected to the two normal inputs of the MUX. The two lines are measured under the same configuration, which reduces the number of
configurations. The control signal of the MUX is supplied from a controller, explained later. The remaining input of the 4-input LUT is not used.

As mentioned above, all lines that the proposed measurement targets are the three lines; i.e. double, hex and long lines. First, measurements for double lines are discussed. Figure 10 illustrates measurements of double lines. The signal lines in this figure are BUSes with the width of two bits. Every double line is connected to three SBs. One SB, SB
$_i$
, is the input and the remaining two, SB
$_{i+1}$
 and SB
$_{i+2}$
, are the outputs as shown in Fig. 4. Every CLB has four slices as shown in Fig. 5. Two of the four are configured as sender slices, and the remaining two are configured as receiver slices. Since each slice has two BLEs, one SB can be SB
$_i$
 for four double lines at a time. As explained above, every receiver slice can be connected to the outputs of two lines under measurement. Thus, every SB can be SB
$_{i+1}$
 or SB
$_{i+2}$
 for eight double lines. In sum, for every configuration, we can measure up to four double lines per one SB. Similarly, we can measure up to four hex lines per one SB with one configuration. In this, SB
$_i$
, SB
$_{i+1}$
 and SB
$_{i+2}$
 in the above explanation are replaced with SB
$_i$
, SB
$_{i+3}$
 and SB
$_{i+6}$
.

As shown in Fig. 4, every long line is connected to five SBs unlike double and hex lines. In addition, SB
$_{i+24}$
 as well as SB
$_i$
 can be a source of the long line. As shown in Fig. 11, every long line provides eight paths. Two paths are measured with a configuration, and thus the eight are measured with four configurations. For every configuration, measurement is made in a similar way to that for the double and hex lines. The correspondence of SBs among measurements of long lines as well as double and hex lines is summarized in Table 1. The “long line” row has four sub-rows, which correspond to the four configurations. For example of the first row, SB
$_1$
, SB
$_{1+6}$
 and SB
$_{1+12}$
 are in the same columns as SB
$_1$
, SB
$_{1+1}$
 and SB
$_{1+2}$
 in the double line row. We can obtain the explanation for the first configuration for long lines by replacing SB
$_1$
, SB
$_{1+1}$
 and SB
$_{1+2}$
 in the above explanation for double lines to SB
$_1$
, SB
$_{1+6}$
 and SB
$_{1+12}$
.

Since measurements of paths not including $l$ shown in Fig. 8 (a) use neither sender slices nor lines under measurement, these measurements can be parallelized very freely.

3.3 Placement and Routing

(1) placement

The proposed configurations can be categorized into four classes, which Fig. 12 illustrates. Lines along horizontal (east and west) direction and ones along vertical (north and south) direction are measured under different configurations. In the proposed measurement, an FPGA is divided into two parts. One includes lines under measurement, the other is configured as a controller and so on. They alternate after measurements for the one finishes.

Figure 13 gives a detail placement for measurement of horizontal lines. Figure 12 is a brief, in which lines under measurement and controllers are clearly divided. How-
ever, they are shuffled actually. The CLBs on the lowest (zeroth) row serve as a controller for the whole of the measurement system, and control signals are sent through vertical double lines along north direction. The $(2i + 1)$-th rows $(i = 0, \ldots, n/2 - 1; n$ is the number of rows) are under measurement and consist of sender and receiver slices. The $2i$-th rows serve as controllers for the $(2i + 1)$-th rows. This system has only one DVMC, which is connected to a controller for the whole. It is noted that, while the DVMC is connected to only the lower right CLB in this figure, it must be connected to upper and left for the other three classes, which requires a margin on routing resources.

(2) routing of control signals

In (2)-(4), measurement of paths including lines under measurement is discussed. Here, routing and the operation of control signals are discussed. Only the measurement of horizontal lines shown in Fig. 13 is discussed. We can measure vertical lines in a similar manner. The proposed system can measure only one path at a time because it has only one DVMC. The controller must select lines under measurement in sequence supplying appropriate control signals. The selection summarizes in Table 2, the detail of which will be explained at the later paragraphs for the related control signals or parts.

Figure 14 illustrates routing for transition signals. To provide transition signals, two FFs in every CLB for controllers are used as transition FFs. The transition FFs in the $2i$-th row are connected to each other and ones in the $2(i - 1)$-th and $2(i + 1)$-th rows to form a daisy chain. First a transition FF in the lowest row has 1 and the other FFs have 0 to measure the line in the lowest row. Subsequently, the value is shifted to the next FF to measure the next line until the value 1 reaches at the uppermost row. Every CLB has two sender slices; the two transition FFs correspond to the two sender slices. Every slice has two BLEs, which share a transition FF.

Figure 15 shows routing for the clock enable signals.
A clock enable signal is broadcasted in every column. The clock enable signal is activated in the column which has the sender slice for the line under measurement. The two sender slices share a clock enable signal.

As mentioned in 3.2 with Fig. 9, each BLE in every receiver slice is configured as an MUX, which requires a control signal. Figure 16 illustrates how to apply the control signal, which is broadcasted. Every double line has two paths, SB\(_i\) \(\rightarrow\) SB\(_{i+1}\) and SB\(_i\) \(\rightarrow\) SB\(_{i+2}\). Transitions occur in both, and either must be selected. The control signal selects it. As shown in Fig. 10, one of the two inputs is applied from the previous CLB and the other is from the previous but one. In other words, every receiver slice works as SB\(_{i+1}\) for one and SB\(_{i+2}\) for the other; this construction allows the selection.

(3) SSG

In the proposed measurement system, the SSG works as a selector selecting the receiver for the line under measurement; only the subject transition is propagated to the DVMC. There are two classes of gates in the SSG: MUXes and OR-gates. As summarized in Table 2, some selections must be made in the SSG, and they are made with the MUXes. Some selections are made outside the SSG; the OR-gates are used for the selections. For example, clock enable signals select columns of CLBs for sender slices; columns not selected are disabled. In this, receiver slices corresponding to the columns not selected output stable 0’s. The stable 0’s are ignored at the OR-gates and only selected values are propagated to the DVMC.

Figure 17 shows the outline of the SSG for the proposed measurement. The SSG consists of sub-SSGs. The sub-SSGs on the same column are connected in series. The sub-SSGs in the lowest row are also connected in series.

Figure 18 illustrates routing for the SSG. Sub-SSGs are configured on CLBs for controllers. Every sub-SSG has four inputs and a output. In the \(i\)-th row, one input is from the previous sub-SSG in the \(2(i+1)\)-th row and the output is to the next sub-SSG in the \(2(i-1)\)-th row. Two inputs are from receiver slices and the other one is a control signal. The sub-SSG works as a selector which selects among the one from the previous sub-SSG and the two from the receiver slices sensitizing the subject transition to the next sub-SSG, the detail of which is explained in the next paragraph. Every sub-SSG utilizes one slice in each CLB. It is noted that every slice has two BLEs. Every line in Fig. 18 represents a 2-bit bus except the control signal line. The control signal line is shared by the two BLEs.

Figure 19 gives the logic of the sub-SSG. This has an MUX, the normal inputs of which are connected back to the two receiver slices. Figure 20 gives an example indicating the task of the MUX. In this example, the \(j\)-th line denoted by a heavy line is under measurement. As mentioned above with Fig. 14, transition FFs form a chain in every column; only one transition FF has 1. The \(j\)-th FF has 1 in this example. After the measurement of the \((j-1)\)-th line, the FF in the \((j-1)\)-th sender slice has 1. For a preparation of the measurement of the \(j\)-th line, the value 1 are shifted from the \((j-1)\)-th to the \(j\)-th transition FF; this figure shows the state after the preparation. At the clock edge, a rising transition 0 \(\rightarrow\) 1 occurs on the \(j\)-th line and a falling transition 1 \(\rightarrow\) 0 occurs on the \((j-1)\)-th line. However, we need propagate only the rising transition to the output of the SSG; we must ignore the falling transition. The MUX does not select the line with falling transition; the falling transition stops. In the lines other than the \((j-1)\)-th and \(j\)-th lines, no transition occurs, i.e. 0 \(\rightarrow\) 0, and thus the corresponding receiver
slices output stable 0’s, which are ignored at the OR-gate.

Figure 21 illustrates the configuration on the lowest row. Every CLB on the lowest row has a sub-SSG for the lowest row as well as a regular sub-SSG. Figure 22 gives the logic of the sub-SSG for the lowest row. The MUX selects the target BLE from two in the target slice. Its control signal is broadcasted in the lowest row as shown in Fig. 21. Receiver slices in columns which are not under measurement output stable 0’s and thus the output of the sub-SSG is sensitized to the DVMC through the OR-gate.

(4) possibility

Table 3 summarizes the number of the required vertical lines for the control signals. Every SB can be a source for ten double lines along north direction and other ten along south direction in the target FPGA, and thus the target FPGA meets this requirement by a margin. As mentioned above we need some resources for routing to DVMC; the margin can be used for it.

Besides the parts discussed above, we must place a controller generating control signals on the FPGA, which is placed on the lowest row. Here, the possibility of constructing the controller is discussed. Every CLB has four slices. As shown in Fig. 21, two out of four slices are used for the SSG. In addition, as shown in Fig. 14, two FFs are used as transition FFs. Thus we must configure the controller and the transition FFs on the two remaining slices. Table 4 shows an evaluation result for the required resources for the controller and the transition FFs. These were constructed on Virtex-4 XC4VLX15, the smallest of the Virtex-4 family. The controller was placed and routed on only two slices on the lowest row, which indicates that the construction of the controller does not obstruct application of the proposed measurement.

(5) paths not including line under measurement

We discuss placement and routing for measurements of paths not including lines under measurement. The measurement is made using one configuration explained below. Two slices in every CLB are configured as receiver slices, which are placed at the same position as the above case for paths including lines under measurement. The two slices for sender slices are not used. All routing are made in the same manner except in Figs. 14 and 15. Control signals are supplied to receiver slices while they are supplied to sender slices in these figures.

4. Evaluation

First, the number of required configurations is evaluated. In manufacturing testing of FPGAs, a test application time consists mostly of a configuration time [16], which is proportional to the size of the FPGA and the number of configurations. Thus, since the size of the FPGA is an independent parameter to the performance of the proposal, the number of configurations is a key parameter indicating the performance from the viewpoint of the test application time.

The number of configurations $C$ appears as

$$C = N_{\text{class}} \left( C_{\text{double}} + C_{\text{hex}} + C_{\text{long}} + C_{\text{notinc}} \right),$$

where

- $N_{\text{class}}$ is the number of classes of configurations, which is 4 as shown in Fig. 12,

| direction | north | south |
|-----------|-------|-------|
| Figure 14 | 2     | 0     |
| Figure 15 | 1     | 0     |
| Figure 16 | 1     | 0     |
| Figure 18 | 1     | 6     |
| total     | 5     | 6     |

| used        | available | utilization(%) |
|-------------|------------|----------------|
| FF          | 61         | 12,288         | 1              |
| LUT         | 59         | 12,288         | 1              |
| slice       | 35         | 6,144          | 1              |
**C**\textsubscript{double} is the number of configurations for double lines for every class,

- **C**\textsubscript{hex} is the number of configurations for hex lines for every class,

- **C**\textsubscript{long} is the number of configurations for long lines for every class, and

- **C**\textsubscript{notinc} is the number of configurations for paths not including \( l \) shown in Fig. 8 (a), which is 1.

We will consider for a class in which horizontal lines are under measurement. We can make the same discussion for vertical lines. For double lines, we have

\[
C_{\text{double}} = \left\lceil \frac{N_{\text{double}}}{N_{\text{testable, double}}} \right\rceil,
\]

where

- **N**\textsubscript{double} is the number of horizontal double lines the sources of which are a specified SB, and

- **N**\textsubscript{testable, double} is the number of double lines measured per a configuration such that the sources are a specified SB.

Figure 23 illustrates all horizontal double lines the source of which is SB\(_i\), and an example of four lines measured with a configuration. Measured and not measured lines are denoted by solid and broken lines. As shown in this figure, every SB can be a source for 20 double lines along horizontal (east and west) directions. Thus we have \( N_{\text{double}} = 20 \). From Fig. 10, the number of sender slices in a SB is two. Every slice has two BLEs. As a result we have \( N_{\text{testable, double}} = 4 \). In fact, four lines are measured with a configuration in Fig. 23. Consequently \( C_{\text{double}} = 5 \). Similarly \( C_{\text{hex}} = 5 \).

For long lines, we can establish that

\[
C_{\text{long}} = 2 \left\lceil \frac{N_{\text{long}}}{N_{\text{testable, long}}} \right\rceil,
\]

where

- **N**\textsubscript{long} is the number of horizontal long lines the sources of which are a specified SB, and

- **N**\textsubscript{testable, long} is the number of long lines measured per a configuration such that the sources are a specified SB.

This is similar to that for double lines except that two is multiplied. The reason why two is multiplied is as follows: Every long line is measured using four configurations as previously discussed using Table 1 and Fig. 11. However both SB\(_i\) and SB\(_{i+24}\) can be sources for a line. Thus the number of required configuration for a long line is two per a source. Since \( N_{\text{long}} = 5 \) and \( N_{\text{testable, long}} = 4 \), we have \( C_{\text{long}} = 4 \).

As a result, the number of configurations \( C \) appears as 60. A recent stuck-at fault testing for global routing resources on Virtex-4 [10] requires 51 \( \sim \) 83 test configurations. Thus, we can conclude that, by using the proposed method, we can measure path delays of all global routing resources in a comparable time to stuck-at fault testing for global routing resources. While stuck-at fault testing detects only stuck-at faults, delay fault testing detects delay faults as well as stuck-at faults [17]. Since the proposed testing targets delay faults, we can conclude that the proposed testing detects more faults with a comparable time to traditional stuck-at fault testing.

Next, test application time is compared to test configuration time in the proposed measurement, which gives evidence that the above evaluation using the number of test configurations is reasonable. Test configuration time is almost the same as the time for scan-in of configuration data (bit stream) through JTAG. The test application time consists of the following three: 1) time for initialization, 2) time for measurement and 3) time for scan-out of measurement results. We can make initialization in one clock cycle; we can measure a path in approximately one clock cycle [3]. As a result, test application time consists mostly of time for scan-out operations. Scan-in and scan-out times appear as \( NF/P \) where \( N \) is the data amount, \( F \) is the clock period and \( P \) is the number of I/O pins which serve the operation. Thus the data amount is a key parameter.

Here, the data amount of configuration data and measurement results are compared. For example of Virtex-4 XC4VLX15, the size of bit-stream is approximately 581 KB, and thus the amount of test configuration data is approximately 278 M bit. The amount of measurement re-
sults appears as

\[ N_{\text{transition}}(N_{\text{line}} + N_{\text{class}}N_{\text{receiver}}N_{\text{BLE}})D, \]

where
- \( N_{\text{transition}} \) is the number of directions for every transition, which is two, i.e., rising and falling,
- \( N_{\text{line}} \) is the number of target lines on an FPGA (for lines which have multiple paths, the number of the paths are counted),
- \( N_{\text{class}} \) is the number of classes of configurations, which is four as shown in Fig. 12,
- \( N_{\text{receiver}} \) is the number of receiver slices for every class of configurations,
- \( N_{\text{BLE}} \) is the number of BLEs in every slice, which is two, and
- \( D \) is the data size of a measurement result per a line.

The terms \( N_{\text{line}} \) and \( N_{\text{class}}N_{\text{receiver}}N_{\text{BLE}} \) are for measurements of paths including and not including the line under measurement, respectively. The number \( N_{\text{line}} \) is 184 K for an FPGA with 6,144 slices, the same as Virtex-4 XC4VLX15. The number \( N_{\text{receiver}} \) is 1,536 for the FPGA. By setting \( D = 5 \) bits, we can achieve enough high accuracy for small delay test. Thus the amount of measurement results is approximately 2 M bit. Therefore, we can conclude that test application time is only approximately \( 10^{-2} \) as long as test configuration time, and thus the above evaluation using the number of test configurations is reasonable.

The proposed design needs two modifications of circuits. One is the modification of BLEs in CLBs as shown in Fig. 7, and the other is adding of an on-chip DVMC. We must design an FPGA considering area increase for the two when we use the proposed method. Table 5 shows the area ratio of the modified CLB and the on-chip DVMC to the original CLB, which are described in Verilog-HDL and compiled with Synopsys Design Compiler in an industrial technology. The evaluation uses the DVMC illustrated in [3]. The area of the DVMC depends on its time resolution, which is set to 0.1 ns in this evaluation. In general FPGAs, the area of routing resources is much larger than that of the other elements; [18], [19] said that it is more than 65% of the total area of a modern FPGA. Thus, the area overhead of the proposed measurement, which is smaller than the original CLBs, can be regarded as small. For example, we assume that the area of the original CLBs is 35% of the total area of an FPGA, and then use of the modified CLBs increases the total area by 1.02 times. For example of an FPGA with 6,144 slices, the same as Virtex-4 XC4VLX15, the area of the DVMC is less than 0.01% of the total area.

| Table 5 | Area (ratio). |
|---------|--------------|
| original CLB | modified CLB | DVMC |
| 1.00     | 1.07         | 0.22 |

5. Conclusion

This paper has presented a design for delay measurement aimed at detecting small delay defects occurring on global routing resources in FPGAs. It also shows a measurement method using the proposed design, which is based on an existing measurement for SoC using a DVMC suitable for use in manufacturing testing. The proposed design uses modified BLEs and an on-chip DVMC. The number of configurations required by the proposed method is 60, which is comparable to that required by stuck-at fault testing for global routing resources in FPGAs. The area overhead is low for general FPGAs, in which the area of routing resources is much larger than that of the other elements such as CLBs. The area of the modified CLB is 7% larger than an original CLB, and the area of the on-chip DVMC is 22% as large as an original CLB. For recent FPGAs, we can estimate that the area overhead is approximately 2% or less of the FPGAs.

The proposed measurement targets only global routing resources. The future work includes the development of measurements for other parts such as CLBs and SBs.

Acknowledgements

This work was supported by VLSI Design and Education Center (VDEC), the University of Tokyo in collaboration with Synopsys, Inc.

References

[1] M. Tehranipoor and N. Ahmed, Nanometer Technology Designs: High-quality Delay Tests, Springer, 2007.
[2] K. Noguchi, K. Nose, T. Ono, and M. Mizuno, “A small-delay defect detection technique for dependable LSIs,” Proc. IEEE Symp. VLSI Circuits, pp.64–65, June 2008.
[3] T. Tanabe, K. Katoh, K. Namba, and H. Ito, “A delay measurement for VLSI circuit by subtraction,” IEICE Trans. Inf. & Syst. (Japanese Edition), vol.393-D, no.4, pp.460–468, April 2010.
[4] K. Katoh, K. Namba, and H. Ito, “A low area on-chip delay measurement system using embedded delay measurement circuit,” Proc. IEEE Asian Test Symp., pp.343–348, Dec. 2010.
[5] R. Datta, A. Sebastine, A. Raghunathan, and J.A. Abraham, “On-chip delay measurement for silicon debug,” Proc. ACM Great Lakes Symp. VLSI, pp.145–148, May 2004.
[6] M.C. Tsai, C.H. Cheng, and C.M. Yang, “An all-digital high-precision built-in delay time measurement circuit,” Proc IEEE VLSI Test Symp., pp.249–254, April 2008.
[7] A. Kumar and M. Anis, “FPGA design for timing yield under process variations,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol.18, no.3, pp.423–435, March 2010.
[8] P. Sedcole and P.Y.K. Cheung, “Within-die delay variability in 90nm FPGAs and beyond,” Proc. IEEE Int’l Conf. Field-Programmable Technol., pp.97–104, Dec. 2006.
[9] E. Bergeron, M. Feeley, M.A. Daigneault, and J.P. David, “Using dynamic reconfiguration to implement high-resolution programmable delays on an FPGA,” Proc. IEEE Northeast Workshop Circuits Syst. TAIAS Conf., pp.265–268, June 2008.
[10] T. Tuan, A. Lesca, C. Kingsley, and S. Trimberger, “Analysis of within-die process variation in 65nm FPGAs,” Proc. IEEE Int’l Symp. Qual. Electron. Des., pp.1–5, March 2011.
[11] J.S.J. Wong, P. Sedcole, and P.Y.K. Cheung, “Self-characterization of combinational circuit delays in FPGAs,” Proc. IEEE Int’l Conf. Field-Programmable Technol., pp.17–23, Dec. 2007.
[12] J.S.J. Wong and P.Y.K. Cheung, “Improved delay measurement method in FPGA based on transition probability,” Proc.
ACM/SIGDA Int'l Symp. Field Programmable Gate Arrays, pp.163–172, Feb. 2011.

[13] J. Yao, B. Dixon, C. Stroud, and V. Nelson, “System-level built-in self-test of global routing resources in Virtex-4 FPGAs,” Proc. IEEE Southeastern Symp. Syst. Theory, pp.29–32, March 2009.

[14] J. Yao, Built-In Self-Test of Global Routing Resources in FPGAs, LAP Lambert Academic Publishing, 2011.

[15] L.T. Wang, C.W. Wu, and X. Wen, VLSI test principles and architectures: design for testability, Morgan Kaufmann, 2006.

[16] L.T. Wang, C.E. Stroud, and N.A. Touba, System-on-Chip Test Architectures: Nanometer Design for Testability, Morgan Kaufmann, 2007.

[17] A. Krstic and K.T. Cheng. Delay fault testing for VLSI circuits, Kluwer Academic Publishers, 1998.

[18] A. Gayasen, N. Vijaykrishnan, and M.J. Irwin, “Exploring technology alternatives for nano-scale FPGA interconnects,” Proc. Annu. Des. Autom. Conf., pp.921–926, 2005.

[19] M.H. Moaiyeri, A. Jahanian, and K. Navi, “Comparative performance evaluation of large FPGAs with CNFET- and CMOS-based switches in nanoscale,” Nano-Micro Lett., vol.3, no.3, pp.178–188, 2011.

Kazuteru Namba received B.E., M.E. and Ph.D. from Tokyo Institute of Technology in 1997, 1999 and 2002, respectively. He joined Chiba University in 2002. He is currently an Assistant Professor of Graduate School of Advanced Integration Science, Chiba University. His current research interests include dependable computing. He is a member of the IEEE and the IPSJ.

Nobuhide Takashina received B.E. and M.E. from Chiba University in 2010 and 2012, respectively. He joined Mitsubishi Electric Corporation in 2012.

Hideo Ito was born in Chiba, Japan, on June 1, 1946. He received the B.E. degree from Chiba University in 1969 and the D.E. degree from Tokyo Institute of Technology in 1984. He joined Nippon Electric Co. Ltd. in 1969 and Kisarazu Technical College in 1971. He had been a member of Chiba University from 1973 until March 2012. He is currently a Professor Emeritus of Graduate School of Advanced Integration Science. He had been a member of the IEEE and the IPSJ.