Degradation of Oscillation Frequency of Ring Oscillators Placed on a 90 nm FPGA

Shouhei Ishii
Department of Electronics
Kyoto Institute of Technology
Matsugasaki, Sakyo, Kyoto 606-8585
sishii@vlsi.es.kit.ac.jp

Kazutoshi Kobayashi
Department of Electronics
Kyoto Institute of Technology
Matsugasaki, Sakyo, Kyoto 606-8585
kazutoshi.kobayashi@kit.ac.jp

Abstract— We focus on degradation of FPGAs which has become dominant due to scaling and quantitatively estimate the degradation of FPGAs by NBTI. We map ring oscillators on the Cyclone II FPGAs and measure the variation of oscillation frequency. In the result, the variation of oscillation frequency is 2.46%. As for degradation of FPGAs, we measure degradation of oscillation frequency until 10,000 seconds passed at room temperature (28°C), 80°C and 100°C. As the result, degradation of oscillation frequency increases as temperature increases and degradation of about 0.1% at 10,000 seconds is observed at high temperature.

I. Introduction

Integrated circuit technology has advanced rapidly. The number of transistors on a chip has exponentially increased in these decades. As a result, the element becomes small and the number of transistors per chip increases and operation speed progresses. But the problem due to the variation in the elements has become apparent while the number of elements increase.

As the element is becoming smaller in this way, the various problems happen. The main contribution in this paper is to quantitatively measure the variation of FPGAs (Field Programmable Gate Array) and degradation of FPGAs by NBTI (Negative Bias Temperature Instability).

NBTI is the phenomenon in which threshold voltage of the device increases gradually over time by negative bias voltage being applied between the gate and the source of the pMOS[1]. When threshold voltage of the device increases, logic gates will become slower. As for degradation, the speed of degradation by NBTI is different according to the history of dynamic and static stress[2]. In [3] treating degradation of FPGAs, the measurement results on the frequency of the oscillator is shown but they do not show how to implement an array of the oscillator in commercial FPGAs. It is very difficult to verify their results. On the other hand, the proposed measurement scheme can be used by anyone who implements a circuit configuration in a commercial FPGA. It is not mandatory to know the detailed architecture of FPGAs which is not disclosed to customers. In the proposed method, it is very easy to migrate FPGAs to the latest one since the proposed method only uses commercially-available FPGA implementation tools, eg: Quartus II from Altera.

In this paper, we explain how to the variation of FPGAs in Section II. In Section III, we show the measurement results. Finally, we explain the conclusion in Section IV.

II. Measurement of the variation on FPGAs

In this section, we explain how to implement ROs (Ring Oscillators) on FPGAs to measure the variation of FPGAs, arrange ROs automatically, and measure oscillation frequency automatically.

A. Implementation of the oscillator

We show an oscillator and the circuit to divide oscillation frequency in Fig. 1, which is constructed by ROs, dividers, and input/output pins. We measure the oscillation frequency by counting the number of oscillations coming to an output pin. It is impossible to directly output a high-speed signal from ROs. Dividers are located between the output pin and ROs. Fig. 2 is the diagrammatic illustration of Fig. 1. In order to measure variations of oscillation frequency of all ROs in the whole FPGA, ROs must be implemented everywhere in the FPGA. To reduce the measurement time, DEMUX (Demultiplexer) and MUX (Multiplexer) as in Fig. 3 are also implemented to choose a measured RO in one configuration.

An RO oscillates when ENABLE is high in Fig. 3. The first contribution of this paper is to measure the variation of these ROs. We have to map ROs in an entire FPGA to obtain the variation. In the proposed method, we only use a commercially-available FPGA configuration tool, Quartus II. We can implement many ROs by increasing the number of inputs and outputs in DEMUX and MUX. But, if we increase the number of input and output, oscillation frequencies are fluctuated RO by RO because we cannot control the RO structure. We choose

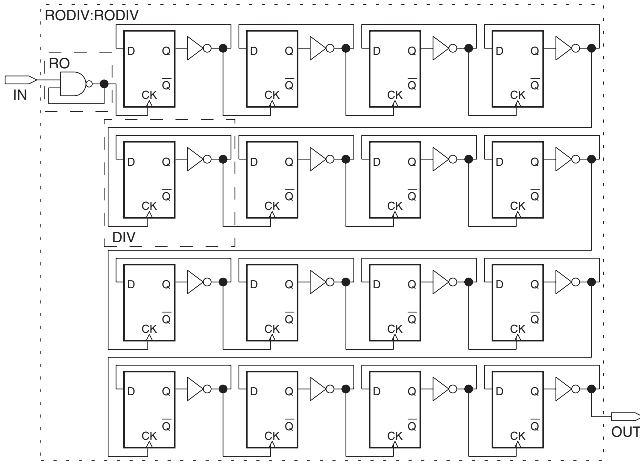

Fig. 1. The structure of the oscillator circuit and the circuit for dividing oscillation frequency

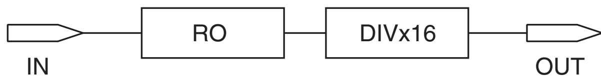

Fig. 2. The diagrammatic illustration of Fig. 1

four ROs in a configuration since only two different structures of ROs are available. Details are shown in Section III.

B. Creation of the program for automatizing places

In Section II.B and II.C, we explain how to automate places and measurement to measure the variation of FPGAs.

If we can automatically assign the location of ROs, we can measure the variation of FPGAs in a short time. So we created the program for automatic places to map on FPGAs. As for places of ROs, we use Quartus II from Altera. It is possible to assign ROs in any location by changing the contents of qsf (Quartus II Settings File),

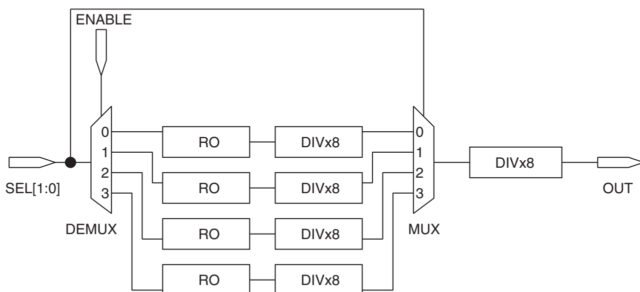

Fig. 3. The circuit added DEMUX, MUX, and ENABLE in Fig. 2

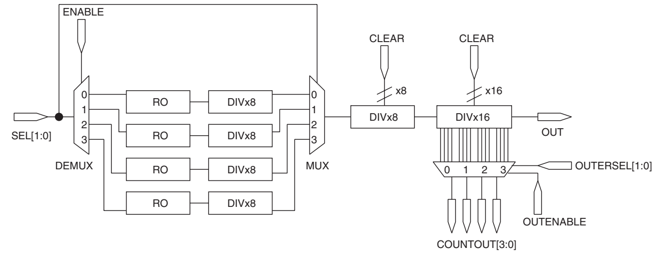

Fig. 4. The circuit diagram to control FPGA by using FT245RL

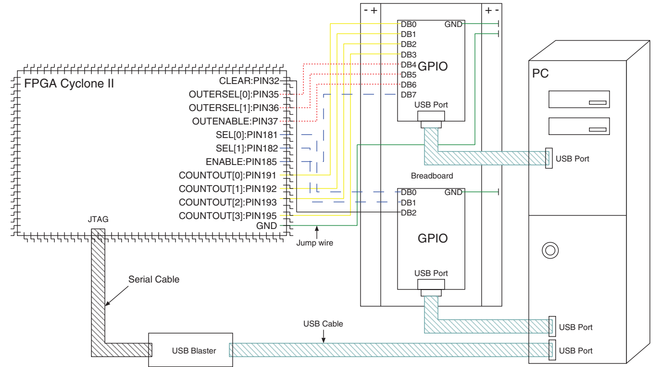

Fig. 5. The diagrammatic illustration of FPGA, GPIO, and PC

which is generated after we completed places and routes.

We can generate sof (SRAM Object File), which is used to download the circuit structure on FPGAs from the modified qsf and the circuit described in VHDL or Verilog. We can generate these qsf by automatically rewriting the contents in qsf with Perl.

C. Creation of the program for automatizing measurement

We use FT245RL (USB parallel conversion module) to control FPGAs and measure oscillation frequency with counters. The bit widths of dividers is changed to 32 bit to count up to $2^{32} = 4,294,967,296 \simeq 4.3$ billion oscillations. Actually, oscillation frequency output from ROs is about 3 GHz. When oscillation time is one second, the count value is 3 billion and does not cause the overflow. GPIO pins receive output of the upper 16 bit. The GPIO receives 4 bit at a time by using DEMUX and MUX because of limitation of GPIO pins. We show the circuit diagram to control FPGA by using FT245RL in Fig. 4. As in Fig. 4, the selector (OUTERSEL), ENABLE (OUTENABLE), and RESET (CLEAR) are used to receive 4 bit at a time in Fig. 3. CLEAR is used to reset the counters. GPIO pins are connected to input and output ports of FPGAs as in Fig. 5.

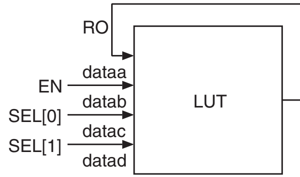

Fig. 6. dataa RO

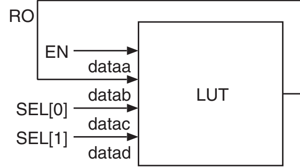

Fig. 7. datab RO

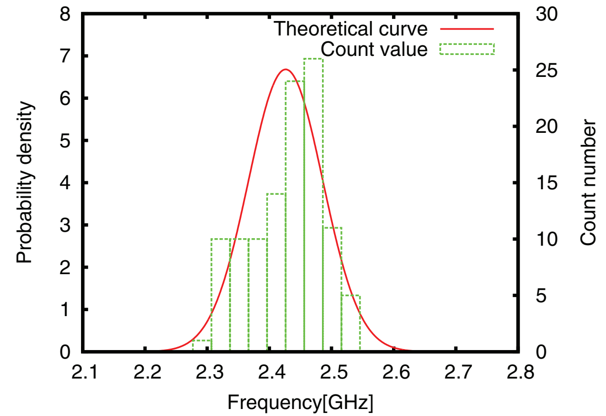

Fig. 8. The normal distribution curve and the number of counts of oscillation frequency in the interval

TABLE I
MEASUREMENT RESULTS OF THE AVERAGE OSCILLATION FREQUENCY AND THE STANDARD DEVIATION

	Average	Standard deviations
RO	2.43 GHz	2.46%

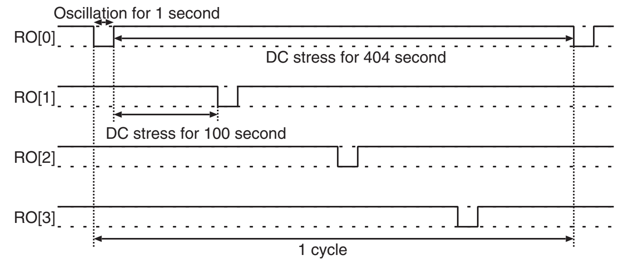

Fig. 9. The time chart for a cycle

III. RESULTS AND CONSIDERATION

In this section, we discuss results of variations of ROs in an entire FPGA and degradation of ROs placed in a specific region in an FPGA.

A. *Measurement results on the variation of FPGAs*

The circuit which we use to measure the variation is Fig. 4. Measurements are repeated for ten times to increase accuracy of measures. It takes 112 minutes to complete the measurement in an FPGA. The variation of oscillation frequency on FPGAs includes these two components. One is the variation of FPGAs itself and the variation by the measurement error. To remove the measurement error, each RO is measured by 10 times.

In addition to that, the structure of ROs are classified to two types as shown in Fig. 6 and Fig. 7. We call the case that ROs is the route of dataa dataa RO and that ROs is the route of datab datab RO. The value of oscillation frequency is 1.3 ∼ 1.8 GHz in the case of dataa and 2.3 ∼2.5 GHz in the case of datab. The oscillation frequency of dataa is much more fluctuated than that of datab. We consider that the fluctuations in dataa is mainly caused by some unknown factors besides variations. We compute variations of oscillation frequency from datab ROs. TABLE I shows measurement results of the average oscillation frequency and the variance.

We show the distribution of the measured oscillation frequencies in Fig. 8. The measured frequencies are very close to the normal distribution.

B. *Measurement results on degradation of FPGAs by NBTI*

We use the ROs with MUXs and DEMUXs in Fig. 4 to measure degradation of FPGAs. When measuring variations, ROs are mapped in an entire FPGA. But the locations of ROs are fixed when measuring degradations. The degradation caused by NBTI is accelerated when the DC stress is applied. It means that some pMOSs are always stressed by negative bias, but other pMOSs are always without stress. The DC stress can be given to ROs to stop oscillation. When measuring the degradations after the DC stress, ROs are oscillated. During oscillation, ROs are exposed in the AC stress in which all pMOSs are alternately stressed and unstressed. During the degradation measurement, we stop oscillation for the DC stress in 100 seconds, and then RO are oscillated for a second. We show the measurement scheme in Fig. 9. Measurement time is about 10,000 seconds at a specific RO location. In order to obtain the degradation tendencies of the ROs, ROs are placed in 28 different locations.

We show the degradations in oscillation frequency at a given location under the condition of room temperature , 80 °C, and 100 °C in Fig. 10. According to Fig. 10, degradation of about 30 MHz is observed compared with room

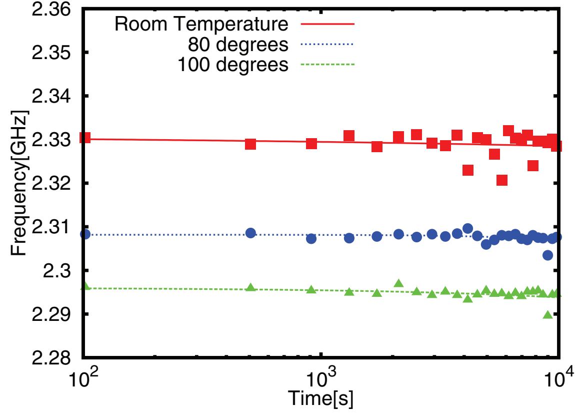

Fig. 10. Changes in oscillation frequency at a given location

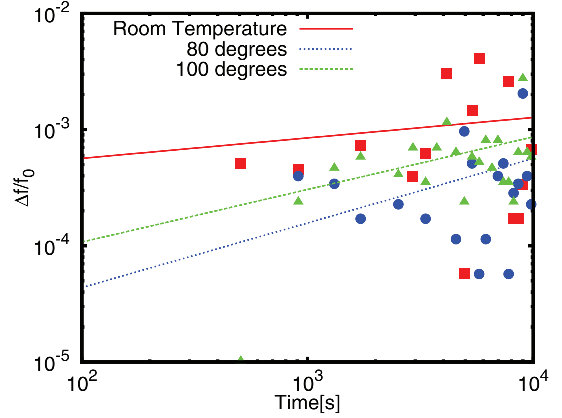

Fig. 11. Changes in $\frac{\Delta f}{f_0}$ at a given location

temperature at °C. Degradation of oscillation frequency is about 0.1% after 10,000 seconds passed.

During the stress, the threshold voltage is degraded in proportion to the nth power of time (t^n)[3] as follows.

$$\Delta V_{\text{th}} \propto at^n \tag{1}$$

ΔV_{th} is also proportional to oscillation frequency (f)[4]. Thus, the measured frequency is degraded in proportion to ΔV_{th}

$$f = -f_0 \Delta V\text{th} + f_0 = -f_0 \alpha t^n + f_0 \equiv at^n + f_0 \tag{2}$$

As n is bigger, oscillation frequency is easier to degrade. There are results that n is $1/6 \sim 1/4$ on degradation of oscillation frequency[1, 5]. By transforming (2) equation and making $\{f_0 - f\}$ $\{\Delta f\}$,

$$\frac{\Delta f}{f_0} \equiv \frac{f_0 - f}{f_0} = \alpha t^n \tag{3}$$

We show the graph describing degradation by using t as the horizontal axis and $\frac{\Delta f}{f_0}$ as the vertical axis in Fig. 11. According to Eq. (3), the $\frac{\Delta f}{f_0}$ value becomes bigger as time passes. The increased amount of $\frac{\Delta f}{f_0}$ is $10 \sim 100$ times in the most cases after 10,000 seconds passed.

C. Measurement results on degradation of FPGAs by NBTI for longer DC Stress and measurement time

In the measurement results for 10,000 seconds with the 100-sec. DC stress period as shown in Fig.10, the degradation is not significant. To confirm the degradation in much longer DC stress and measurement time, we increase time for the DC stress from 100 seconds to 1,200 seconds and measurement cycles from 25 cycles to 50 cycles. Thus the total measurement time is 60,000 seconds. The measurement results is shown in Fig. 12 and 13. According to Fig. 13, the $\frac{\Delta f}{f_0}$ value becomes bigger as time passes like Fig. 11. From Fig. 13, fluctuations of the measured frequencies become small after 10,000 seconds.

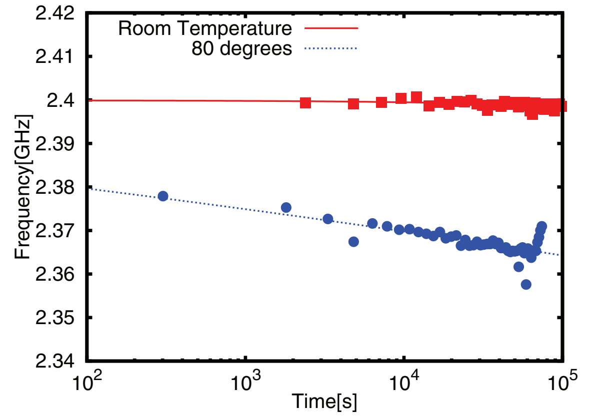

Fig. 12. Changes in oscillation frequency of ROs at a randomly selected point

As for progression of degradation, the amount of degradation in Fig. 12 is four times as at 10,000 seconds at 80 °C compared to Fig. 10. The reason is that time to stop oscillation and to add DC stress increases to 1,200 seconds. From these results, time of about $30{,}000 \sim 40{,}000$ seconds is required to obtain exact fitting waveforms. By increasing time to add DC stress from 100 seconds to 1,200 seconds, clear degradation can be observed.

The fitted values n in t^n are shown in Table II. The fitted values of n are fluctuated because the measured values are fluctuated so much especially just after the measurement was started. Actually, the contribution of n is small on the measurement in the short period. The fitted lines are not considerably changed by changing n.

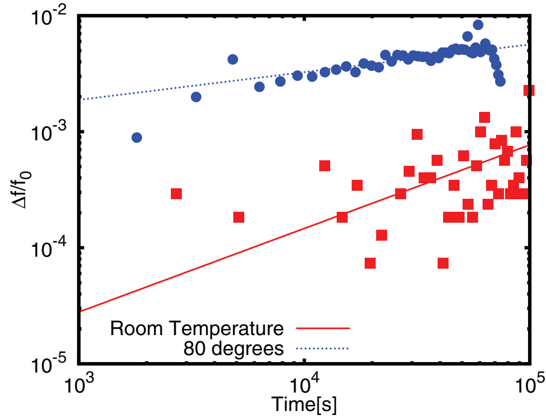

Fig. 13. Changes in $\frac{\Delta f}{f_0}$ of ROs at a randomly selected point

TABLE II
VALUE OF a, n, *and* f_0 IN (2) EQUATION

	a	n	f_0
Room Temperature	-4.48×10^{-4}	0.194	2.33
80°C	-6.59×10^{-10}	1.59	2.31
100°C	-1.24×10^{-5}	0.565	2.30
Room Temperature at a particular location	-4.14×10^{-6}	0.513	2.40
80°C at a particular location	-0.0506	0.0336	2.44

IV. CONCLUSION

In this paper, we measured degradation of FPGAs by NBTI. FPGA devices are Cyclone II from Altera fabricated in a 90 nm process. As for measurement of the variation, we implement arrays of oscillators by configuring four oscillators by one configuration and changing the location of the oscillator to generate the configuration and program it automatically. The measured standard deviation of the oscillated frequency is 2.46%.

As for degradation of FPGAs, first, oscillation frequency is measured until 10,000 second with 100 sec. DC stress interval by changing ambient temperature to the room temperature (28°C), 80°C and 100°C, respectively. Degradation of FPGAs by NBTI is around 0.1% after 10,000 stress time.

By applying longer 1,200-sec DC stress period and 60,000-sec total measurement time, clear degradations can be measured. But the fitted n values of t^n are fluctuated so much caused by the fluctuation of the measured frequencies after the measurement was started.

ACKNOWLEDGMENT

I express my deep appreciation to Michitaro Yabuuchi, who gave me some advices to research degradation of FPGAs. I also express my appreciation to people at Kyoto University who lent me the constant-temperature unit when I measured degradation of FPGAs and people at the Kobayashi laboratory who cooperated with me.

REFERENCES

[1] K. Kang, K. Kim, A. E. Islam, M. A. Alam and K. Roy: "Characterization and estimation of circuit reliability degradation under NBTI using on-line IDDQ measurement", Proceedings of the 44th annual Design Automation Conference, DAC '07, New York, NY, USA, ACM, pp. 358–363 (2007).

[2] W. Wang, S. Yang, S. Bhardwaj, S. Vrudhula, F. Liu and Y. Cao: "The Impact of NBTI Effect on Combinational Circuit: Modeling, Simulation, and Analysis", Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, **18**, 2, pp. 173–183 (2010).

[3] E. A. Stott, J. S. Wong, N. P. Sedcole and P. Y. K. Cheung: "Degradation in FPGAs: Measurement and Modelling", FPGA (Eds. by P. Y. K. Cheung and J. Wawrzynek), ACM, pp. 229–238 (2010).

[4] H. Matsunami and M. Yoshimoto: "Semiconductor Devices", Kyoritsu Shuppan Co., Ltd. (2000).

[5] K. Ramakrishnan, S. Suresh, N. Vijaykrishnan and M. Irwin: "Impact of NBTI on FPGAs", VLSI Design, 2007. Held jointly with 6th International Conference on Embedded Systems., 20th International Conference on, pp. 717–722 (2007).