

商用FPGAのばらつきとBTIによる経年劣化

石井 翔平[†] 小林 和淑^{†,††}

[†] 京都工芸繊維大学大学院工芸科学研究科 〒606-8585 京都市左京区松ヶ崎御所海道町
^{††} JST, CREST

あらまし 本稿では微細化に伴い顕著になってきたFPGAのばらつきや経年劣化に関する問題に着目し、BTIによるFPGAの経年劣化を定量的に評価し、ばらつきと経年劣化の関連性を示す。90nmと60nmプロセスFPGAの経年劣化の測定を行うために、FPGA上にリングオシレータをマッピングして、その発振周波数の標準偏差を実測した。FPGAの経年劣化に関する測定では温度を室温(25°C)、80°Cに保ち、30,000秒経過までの発振周波数のばらつきを測定した。

キーワード BTI、FPGA、ばらつき、経年劣化

Variations and BTI-induced Aging Degradation on Commercial FPGAs

Shouhei ISHII[†] and Kazutoshi KOBAYASHI^{†,††}

[†] Graduate School of Science & Technology, Kyoto Institute of Technology
^{††} JST, CREST

Abstract In this paper, we focus on problems concerning variations and degradation on FPGAs which has become dominant due to scaling and quantitatively estimate the degradation of FPGAs by BTI. We show the relationship between variation and degradation. To measure degradation of 90nm and 60nm FPGAs, we map ring oscillators on FPGAs and measure standard deviation of oscillation frequency. As for degradation of FPGAs, we measure variations of oscillation frequency for 30,000 seconds at the room temperature (25°C) or 80°C.

Key words BTI, FPGA, Variation, Degradation

1. 序 論

集積回路技術は飛躍的に進歩し、1つのチップに搭載されるトランジスタの数は、ここ十年で急激に増加した。その結果、素子が微細化して集積度が増し、動作も高速化した。一方、素子数が増えたことによる素子のばらつきの問題が顕在化してきた。素子が微細化することにより、ばらつきは増大し、経年劣化の影響も受けやすくなる[1]。

このように微細化が進むにつれて、様々な問題が発生しているが、本論文では特にFPGA(Field Programmable Gate Array)のばらつきとFPGAのBTI(Bias Temperature Instability)による経年劣化を実測により定量的に求めることを目的とする。BTIとは、pMOSのゲートソース間に負バイアスまたはnMOSのゲートソース間に正バイアスが印加されることで、時間の経過とともにデバイスの閾値電圧が大きくなっていく現象である[2]。閾値電圧が大きくなることで、トランジスタの速度は遅くなる。劣化に関して、動的(Dynamic)に劣化させた場合と、静的(Static)に劣化させた場合で、BTIによる劣化の速度は異なっている[3]。

本論文ではAlteraの90nmFPGA Cyclone IIと60nm

FPGA Cyclone IVを用いてFPGAのばらつきや経年劣化を評価し、ばらつきと経年劣化の相関を調べる。2節でFPGAのばらつきを求める方法について、3節ではFPGAの経年劣化を測定するための方法について述べる。4節では2節、3節で得られた結果を示し、それに対する考察を行い、5節で結論を述べる。

2. FPGAのばらつきの測定

ここでは、測定に用いたFPGAについて、FPGAのばらつきを求めるために必要なRO(Ring Oscillator)をFPGAに実装する方法、測定回路の配置を自動化する方法、発振周波数の測定を自動化する方法について述べる。

2.1 測定に用いたFPGA

本論文で用いたFPGAはAlteraの90nmFPGA Cyclone IIと60nmFPGA Cyclone IVである。FPGAとはプログラム可能なロジックデバイスであり、製造後も購入者が再設計可能な集積回路である。このFPGAの内部構造は図1のようになっており、Logic ArrayはLogic Array Block(LAB)をアレイ状に並べたものであり、LABは論理回路を構成するブロックとなっている。LABの内部はLogic Element(Logic Cellとも

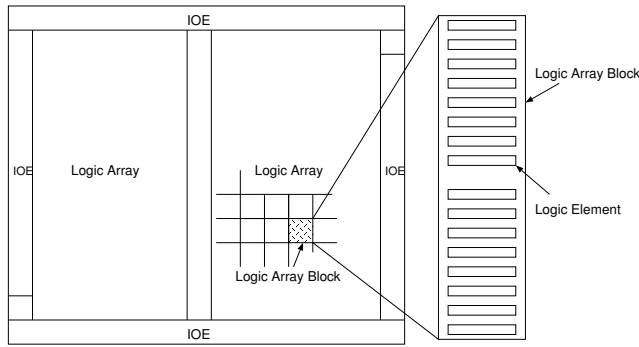


図 1 FPGA の内部構造

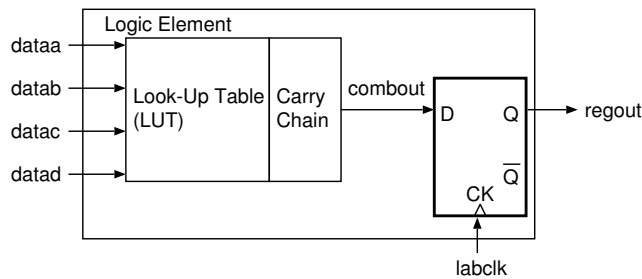


図 2 LE の内部構造

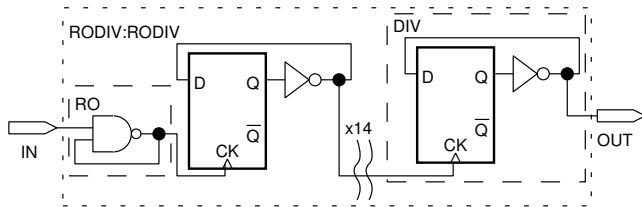


図 3 発振回路と発振周波数を分周する回路

呼ばれる)で構成され、LE は組み合わせ回路と順序回路を構成する FPGA の最小単位となっている。Cyclone II と Cyclone IV の LE の構造の概略図を次の図 2 に示す。LE の数は 90nm FPGA が 288 個、60nm FPGA が 1539 個となっている。

2.2 発振回路の作成

発振回路と発振周波数を分周するための回路の構造を図 3 に示す。図 3 は RO と分周器 (DIV:Divider) と入出力ピンで構成され、発振させたものを分周器を用いて分周し、その発振周波数を調べる。分周させて周波数を落として測定することで、ノイズの影響の少ない正確な発振周波数を求めることが出来る。図 3 が今回測定を行う回路ではあるが、1 つの RO ではなく複数の RO をマッピングした方が一度にまとめて測定出来るため、図 4 のような DEMUX (Demultiplexer) と MUX (Multiplexer) を用いて、多数の発振器を切り替える。これにより、1 回の Configuration (FPGA に実装する回路構造データをダウンロードすること) で、多数の発振器の発振周波数が測定出来る。

図 4 中の Enable が High の時、RO が発振する。図 4 では、DEMUX を使うことにより、出力数を増やして多数の発振器の発振周波数が測定出来る。MUX は DEMUX によって複数出力されたものを 1 つにまとめるものである。セレクタを切り替

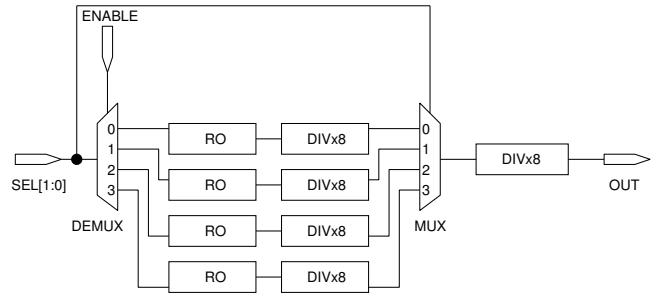


図 4 2 ビット入力切替器での DEMUX → RO → DIVx8 → MUX → DIVx8 の回路図

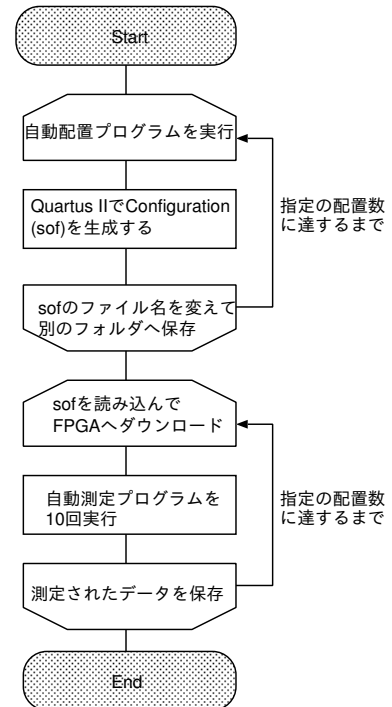


図 5 自動化のフローチャート

えることで、多数の発振器の発振周波数を求めることが出来、マッピングする場所を変えて、発振周波数の値を比較することで、FPGA のばらつきを求めることが出来る。

ばらつきを一度にたくさん調べたい場合は、入力切替器のビット数を増やして一度に同時に測定出来る発振器の数を増やせばよい。しかし、入力のビットを増やすと回路の構造が大きく変わってしまい、それが測定に悪い影響を及ぼす。そこで本研究では 2 ビットの入力切替器を用いて、4 つの発振器の発振周波数を調べ、その発振器の配置を自動で変えることで FPGA のばらつきを調べることにした。実際に多数の発振器を様々な場所にマッピングして得られた発振周波数の測定結果と考察については 4 節で述べる。

2.3 配置と測定の自動化プログラム作成

ここでは、FPGA のばらつきを求めるために必要な配置と測定の自動化について述べる。自動化を行うためのフローチャートを図 5 に示す。このような手順で、FPGA のばらつきを自動で求める。配置の自動化についての詳細は 2.3.1 節、測定の自動化についての詳細は 2.3.2 節に示す。

2.3.1 配置の自動化

発振器の配置場所を指定し、短時間で FPGA のばらつきを求めするために FPGA にマッピングを行う自動配置プログラムを作成した。発振器などの配置には Altera 社の Quartus II を使い、配置配線を行った際に生成される qsf (Quartus II Settings File) の中身を変更することで、配置したい場所を自由に指定出来る。

qsf と Verilog あるいは VHDL で記述した回路で、FPGA に回路構造データをダウンロードするための sof (SRAM Object File) が生成可能である。本研究では qsf を Perl で自動的に書き換え、これをバッチ化して多数の qsf、sof を生成した。

2.3.2 測定の自動化

sof が自動的に生成出来るようになったので、測定の自動化も行う。これには GPIO (General Purpose Input/Output) とである FT245RL (USB パラレル変換モジュール) を用いる。GPIO により FPGA を制御して、カウンタを使って周波数を測定し、出力された周波数の値を PC にまとめて自動的に取り込む。

発振器から出力される発振周波数は 3GHz 程度であるから、1秒間の発振時間だと、30億回のカウントとなる。桁溢れを起こさないためには、図4の後段のカウンタ(分周器)を24段に増やす。これにより前段(1~8段目)のDIVを加えても32段あるため、 $2^{32} = 4,294,967,296 \approx 43$ 億回程度カウント出来る。その後段24段のカウンタの出力を取り出して、FPGA からGPIOピンへ送る。ピン数の関係上ここでは一番後ろ(最も分周しているところ)から16ビット分カウンタの出力を取り出した。取り込むカウンタの数を増やすことによって、発振周波数の精度が増し、その精度は8ビット分カウンタの出力を取り出す時0.391%だが、16ビットでは0.00153%まで上がる。

16ビット分のカウンタをまとめて取り出すにはピンが足りないので、セクタを使って4ビットずつ取り出した。2ビットの取り込むカウンタを切り替えるセクタと1ビットのENABLEを用いて、4ビットずつ取り込む。これに元々あったROを切り替える2ビットの入力切替器と1ビットのENABLEを加え、新たにD-FFをリセットするためにRESETを加えれば、GPIOピンは $7 + 2 + 1 + 1 = 11$ ピンで済む。D-FFにリセットを付けるのは、カウンタの値を出力した後に、リセットしないと前の値がD-FFに残ってしまうためである。

測定を自動で行うために、図4の回路にカウンタを増やして、新たにOUTERSEL、OUTENABLE、COUNTOUT、CLEARを追加した回路図を図6に示す。入出力ピンに新たにOUTERSEL、OUTENABLE、COUNTOUT、CLEARなどを加えているが、これはFPGAから見てそれぞれ、カウンタを4ビットずつ取り込むための入力切替器、その入力切替器のENABLE、出力されるカウンタ値、D-FFのリセットとなっている。GPIOピンとFPGAの入出力ポートをジャンプワイヤで接続して、カウンタの値をやり取りする。

2.4 GPIO と FPGA との接続手順

GPIO と FPGA との接続の手順を説明する。FT245RL は1個で8ビットしかGPIOピンを持たない。しかし、今回の測定

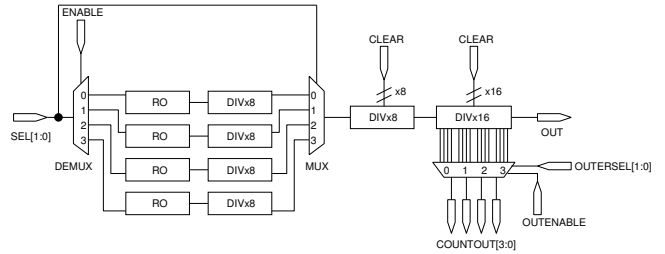


図6 FT245RLでFPGAを制御するための回路図

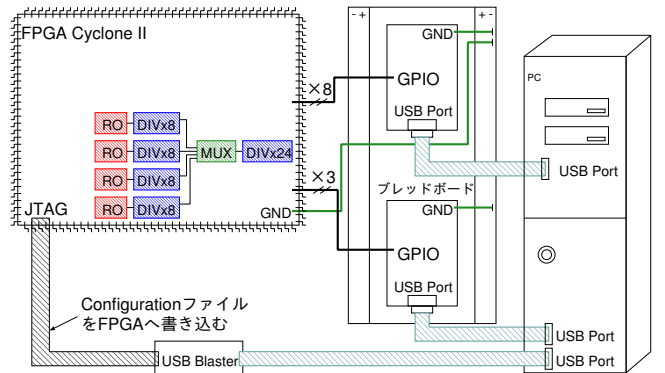


図7 FPGA、GPIO、PCの概略図

では11ビットのGPIOピンが必要となるため、これを2個用いた。

今回の測定で用いるのはGPIO Port (DB0~DB7) とGND、USB Port である。ブレッドボード上にFT245RLを配置して、8ピンのGPIO Portを使って、FPGAボードの入出力ポートとジャンプワイヤで接続する。USBケーブルでFT245RLからPCへと接続することでPCから電源供給がされて、PC上にカウンタの値を表示出来る。FT245RLとFPGAボードはそれぞれ接地しておく。FPGA、GPIO、PCとの概略図を図7に示す。FPGAのピンの役割はそれぞれ

- 発振器の切替器 (SEL[0]、SEL[1])
- セクタ ENABLE ピン (ENABLE)
- D-FF のリセット (CLEAR)
- カウンタ切替器 (OUTERSEL[0]、OUTERSEL[1])
- セクタ ENABLE ピン (OUTENABLE)
- カウンタ出力値 (COUNTOUT[0]~COUNTOUT[3])

である。

GPIO と FPGA の接続をした後、FPGA を GPIO で制御するためにプログラムを作成する。発振時間を1秒と設定すれば、桁溢れが起こらず4つのROの発振周波数を求めることが出来る。短くても桁溢れは起こらないが、GPIOを制御するWindowsの仕様上、割り込み時間の誤差が大きく、短い発振時間では発振周波数の値が不正確な値となる。

3. FPGA の BTI による経年劣化の検証

この節では、FPGA の BTI による経年劣化を調べるために、室温 (25°)、80° での経年変化による発振周波数の推移を求める方法について述べる。BTI は温度による影響が大きく、温度が高ければ高いほど、発振周波数の劣化が大きくなる。今回は

温度による影響も確認するために、室温以外に、80°で同様の測定を行った。発振を止めてDCストレスを与えて発振周波数を劣化させる実験の測定結果と考察については4節で述べる。

FPGAの経年劣化を評価するためには、ある特定の箇所に発振器を配置して一定期間発振周波数の推移を調べる。測定の際、経年劣化とばらつきとの相関を調べたいので、60nm PFGAに関しては、ばらつきの測定で最も高い発振周波数が得られた箇所と、最も低い発振周波数が得られた箇所の経年劣化を測定する。測定にはばらつきの測定で用いたGPIOの制御プログラムを利用し、恒温装置を用いて温度の調節を行いながらFPGAの経年劣化を測定する。

4. 結果と考察

この節では、2, 3節で述べた方法で測定を行い、それによって得られた結果とその結果から分かる考察を述べる。

4.1 FPGAのばらつきの測定結果

2.3.1, 2.3.2節にある方法を用いて、90nmと60nm FPGAのばらつきの測定を行った。測定に用いた回路は図6であり、測定の精度を高めるために、測定は同じ配置でも10回行った。測定箇所は90nm FPGAで448箇所、60nm FPGAで4104箇所あり、測定にかかる合計時間は90nm FPGAで約2時間、60nm FPGAで約13時間であった。

正確なFPGAの発振周波数のばらつきを得るためには、測定誤差を考慮しなければならない。通常、測定回数を増やせば増やす程、その平均値の精度が増すので、測定回数を10回に増やして測定誤差によるばらつきをなるべく小さくしている。測定毎の発振周波数のばらつきがどの程度あるのかを調べるために発振周波数の平均を求めた後、測定誤差の標準偏差も求めた。

発振周波数の値は測定誤差以外にROの発振経路によっても大きく変わるため、異なる発振経路が存在する場合はそれをなるべく取り除かなければならない。図8, 9に示すように、ROの発振経路がそれぞれ異なることがあり、便宜上dataaでROが発振する場合をdataa型、databでROが発振する場合をdatab型と呼ぶ。各型によって発振周波数がどの程度変わるかを表1に示す。表1からROの発振経路によって大きく発振周波数の値が変わることが分かった。

ROの発振経路によって発振周波数の値が大きく変わるため配線の位置をこちらで指定し、全て同じ構造とすれば、この問題は解決出来る。ROの発振経路を固定するためにはLogic Element (LE)内の論理回路をVerilog上に直接記述する。16ビットあるmaskを適切な値に指定することでLUTやFFの配線までも変更することが出来る。表1によれば、databが最も発振周波数が高いので、ROの発振経路がdatabとなるように、LUTの論理回路パターンである16ビットのLut.maskを書き換えた。その際Quartus IIが自動的に最適化を行おうとするため、全ての最適化を解除しなければならない。VerilogでLE内部を直接記述したが、60nm FPGAのみ発振経路を固定することが出来た。

最適化を解除しても一部のROには設定したLUT.maskの値が適用されずに勝手に書き換わったため、測定後ROがdatab

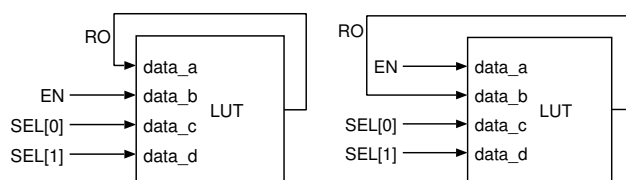


図8 dataa 型

図9 datab 型

表1 それぞれの発振経路での発振周波数の値(概算)

ROの発振経路	発振周波数の値 [GHz]
dataa(60nm FPGA)	1.8
datab(60nm FPGA)	2.5
datac(60nm FPGA)	2.2
datad(60nm FPGA)	0
dataa(90nm FPGA)	1.8
datab(90nm FPGA)	2.4
datac(90nm FPGA)	未確認
datad(90nm FPGA)	未確認

表2 90nm FPGAにおけるばらつきの測定結果

	平均 [GHz]	標準偏差 [%]
RO[0]	2.39	2.98
RO[1]	2.45	2.17
RO[2]	2.43	2.63
RO[3]	2.44	0.923

表3 60nm FPGAにおけるばらつきの測定結果

	平均 [GHz]	標準偏差 [%]
RO[0]	2.45	4.12
RO[1]	2.57	0.982
RO[2]	2.38	4.68
RO[3]	2.46	0.849

を発振経路としているもの以外は全て取り除いた。datab型以外を取り除いた結果、測定箇所は90nm FPGAで112箇所、60nm FPGAで1900箇所となった。表2, 3に90nmと60nm FPGAのばらつきの結果を示す。

10回の測定での標準偏差の平均は最大でも約0.05%であり、FPGAのばらつきと比べて10倍以上小さい結果となり、測定誤差が発振周波数の値に及ぼす影響は小さいことが分かる。表2によれば、標準偏差の値においてRO[3]だけが小さい結果となっている。これは、RO[3]のみがLUT.maskの値がほとんど同じであり、ROの発振経路が固定されていたからである。RO[0]~RO[2]はmaskが回路の配置場所によって変わることが多く、回路構造が少し異なることが多かったため、標準偏差は2%以上となっている。表3も同様で、LUT.maskがほぼ固定されていたRO[1]とRO[3]に関しては1%を切っている。この結果から、LUT.maskが常に固定されている場合、90nmと60nm FPGAの標準偏差は共に1%を切っており、プロセスが進んだことによるばらつきにおける微細化の影響は確認出来なかった。

このばらつきが正規分布となっているかを確認するために、発振周波数を一定区間毎に区切り、その区間内にある発振周波数の個数をカウントした図を図10, 11に示す。図10, 11から、平均値に近づく程カウント数が多くなっているから、正規分布

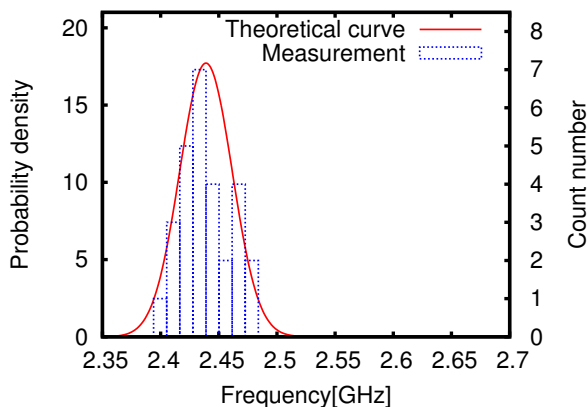


図 10 90nm FPGA における正規分布曲線と区間内の発振周波数のカウント数

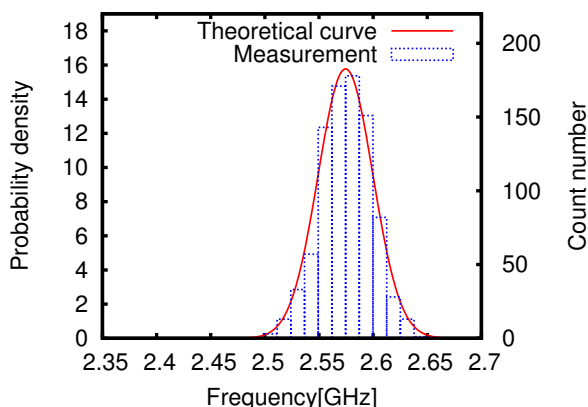


図 11 60nm FPGA における正規分布曲線と区間内の発振周波数のカウント数

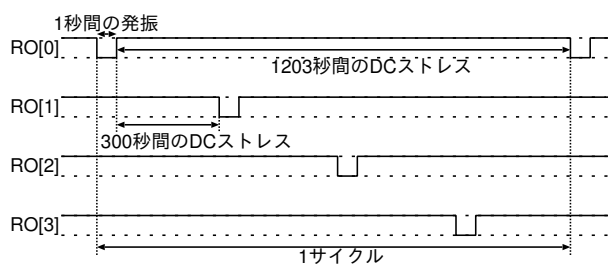


図 12 1 サイクルでのタイムチャート

の形に近いことが分かる。

4.2 FPGA の BTI による経年劣化の測定結果

3 節にある方法を用いて、FPGA の BTI による経年劣化の測定を行った。測定に用いた回路は図 6 である。1 サイクルの測定手順としては図 12 のタイムチャートのようにになっている。25 サイクル繰り返すと、 $1203 \times 25 = 30,075$ 秒 (約 8.5 時間) となる。これを 90nm と 60nm FPGA で行い、RO の発振周波数が経年で劣化するかどうかを調べた。60nm FPGA に関しては、ばらつきとの相関も調べた。

室温、80° の条件下で、90nm FPGA における発振周波数の推移を図 13 に、60nm FPGA におけるばらつき測定で最も高い発振周波数が得られた配置での発振周波数の推移を図 14、最

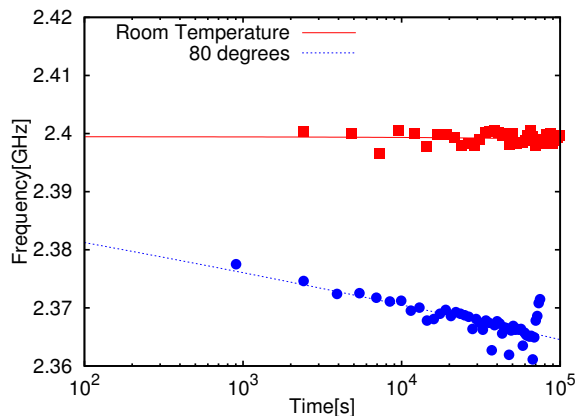


図 13 90nm FPGA での発振周波数の推移

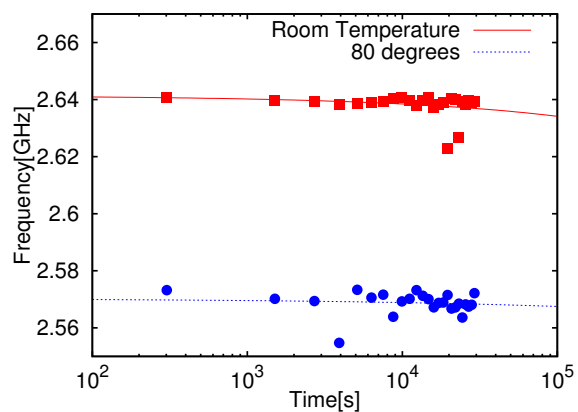


図 14 60nm FPGA で最も高い発振周波数が得られた配置での発振周波数の推移

表 4 発振周波数の劣化率

	30,000s [%]	10 年 [%]
90nm FPGA (室温)	0.00825	1.22
90nm FPGA (80°C)	3.51	4.63
60nm FPGA (最高:室温)	0.142	6.37
60nm FPGA (最高:80°C)	0.0567	1.12
60nm FPGA (最低:室温)	0.0250	1.59
60nm FPGA (最低:80°C)	0.0704	0.611

も低い発振周波数が得られた配置での発振周波数の推移を図 15 に示す。図 13~15 から、発振周波数の劣化率が求められるので、30,000 秒と 10 年経過時点 (推定) における発振周波数の劣化率を表 4 に示す。本来は微細化が進んでいる 60nm FPGA の方が経年劣化が起こりやすいため、60nm FPGA の方が 90nm FPGA よりも劣化率が大きくなるはずだが、表 4 によれば、60nm FPGA は 1% 前後となっている (60nm FPGA (最高:室温) は Fitting が歪んでおり正確ではない)。この結果から、微細化による経年劣化の影響を確認することが出来なかった。

4.3 劣化を表す関係式

しきい値電圧 ΔV_{th} は t^n に比例 [4] するので、

$$\Delta V_{th} \propto at^n \quad (1)$$

と書ける。発振周波数 f としきい値電圧 ΔV_{th} も比例関係 [5] があるので、 f は

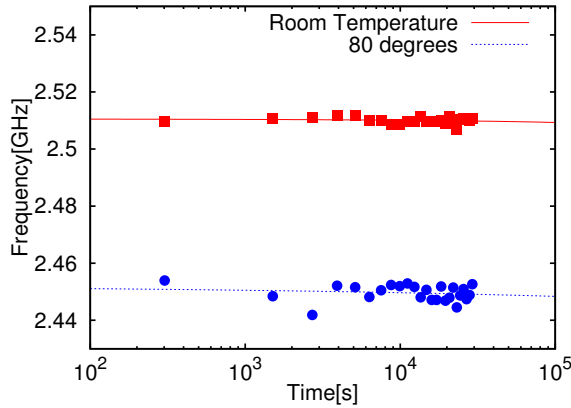


図 15 60nm FPGA で最も低い発振周波数が得られた配置での発振周波数の推移

表 5 (2) 式の a, n, f_0 の値

測定条件	a	n	f_0
90nm FPGA(室温)	-1.11×10^{-6}	0.521	2.40
90nm FPGA(80°C)	-6.30×10^{-2}	0.0301	2.45
60nm FPGA(最高:室温)	-8.17×10^{-5}	0.390	2.64
60nm FPGA(最高:80°C)	-1.01×10^{-4}	0.290	2.57
60nm FPGA(最低:室温)	-8.60×10^{-6}	0.432	2.51
60nm FPGA(最低:80°C)	-4.86×10^{-4}	0.180	2.45

$$f = -f_0 a t^n + f_0 \equiv a t^n + f_0 \quad (2)$$

で表せられ、 n が大きければ大きい程、劣化しやすい。発振周波数の経年劣化ではこの n の値は $1/6 \sim 1/4$ という結果が多い[2], [6]。 t^n の n の値は、表 5 のように示された。表 5 から、 n の値は LE を記述した 60nm FPGA の方が、 $1/6 \sim 1/4$ の範囲に近い結果となった。fitting した時に得られた n の値の標準偏差も 60nm FPGA の方が 1000 倍以上小さくなっている。

5. 結 論

本稿では FPGA の BTI による経年劣化の測定を行った。用いた FPGA デバイスは Altera の 90nm プロセスの Cyclone II と 60nm プロセスの Cyclone IV である。ばらつきの測定では、配置と測定の自動化を行った後、発振器の発振周波数を測定した。FPGA のばらつきの測定には、測定誤差や RO の発振経路の違いによる発振周波数の変動の問題があった。

測定誤差によるばらつきは、測定回数を増やして平均を取ることで、なるべく測定誤差によるばらつきを小さくした。RO の発振経路に関しては、コンパイル時に出力されるネットリストを参照し、LUT の論理回路パターンである 16 ビットの LUT_mask を調べて、Verilog 内に記述して LUT_mask を固定した。LUT_mask が固定できない RO では、異なる LUT_mask が出力されたものの中で遅い発振経路の方を取り除いて、RO の発振経路の問題を解決した。発振周波数の値に与える影響を FPGA のばらつき以外でなるべく小さくした後、発振周波数の標準偏差を求めたところ、その値は 90nm FPGA で最大約 3%、60nm FPGA で最大約 4.7% となった。LUT_mask の固定が上手く行った箇所では 90nm と 60nm FPGA の両方とも標

準偏差は 1% 以下に抑えられた。

FPGA の経年劣化の測定では、温度によって RO の発振周波数にどの程度影響を与えるか評価するため、温度を室温 (25°C)、80°C に保ち、30,000 秒経過までの発振周波数を求めた。発振周波数の経年劣化は 60nm FPGA では温度が高い方が劣化率が小さい結果となっており、高温の方が劣化率が大きいことを示すことが出来なかった。プロセスの微細化による経年劣化の影響も実証出来なかった。これら 2 つが示せなかった原因は経年劣化の測定時間が少ない点と同じ配置における経年劣化の測定回数が少ないからであった。経年劣化の測定結果としては本意な結果に終わったが、経年劣化を表す式である $f = a t^n + f_0$ の n 値では理想的な値が得られた。

ばらつきと経年劣化の相関に関しては 10 年後の経年劣化 (推定) から、60nm FPGA で最も高い発振周波数が得られた配置の方が劣化率が大きくなっている。しかし、最も高い発振周波数が得られた配置の方が室温での n 値においては小さくなっているため、相関は分からなかった。これも高い周波数が得られる配置と低い周波数が得られる配置を 1 箇所ではなく、複数箇所抽出して測定を行い、相関を調べる必要がある。

今後はプロセスの微細化による経年劣化の影響、ばらつきと経年劣化の相関を調べ、余裕があれば、より微細なプロセス (28nm, 22nm) でも同様の測定を行い、それらが発振周波数の経年劣化にどの程度影響を与えるかを調べたいと考えている。

謝 辞

研究を進めるために様々な指導やアドバイスを下さった数内美智太郎さんには深く感謝致します。FPGA の経年劣化を測定するために、恒温装置を貸して頂いた京大の方々、様々なご協力を頂いた小林研究室の皆様にも感謝しています。

文 献

- [1] A. W. Strong, E. Y. Wu, R.-P. Vollertsen, J. S. G. LaRosa, I. Stewart E. Rauch and T. D. Sullivan: "Reliability Wearout Mechanisms in Advanced CMOS Technologies", A JOHN WILEY & SONS, INC. (2009).
- [2] 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).
- [3] 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).
- [4] 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. Wawrzyniek), ACM, pp. 229-238 (2010).
- [5] 松波 弘之, 吉本 昌広: "半導体デバイス", 共立出版株式会社 (2000).
- [6] 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).