

修士論文

題目

単一磁束量子論理回路の故障診断のための
タイミング故障シミュレーションの高速化

指導教員

高木 一義 教授

2022年

三重大学大学院 工学研究科 情報工学専攻
コンピュータアーキテクチャ研究室

渡邊 裕生 (420M513)

単一磁束量子論理回路の故障診断のためのタイミ ング故障シミュレーションの高速化

渡邊 裕生

内容梗概

半導体回路に代わる高速・低消費電力の論理回路として、近年単一磁束量子回路（RSFQ）論理回路の研究が活発に行われている。将来的に半導体集積回路と同様に大規模化が見込まれており、これまでに、RSFQ回路を用いたマイクロプロセッサ等の演算回路の設計・試作がされている。

ゲート等の素子間の情報伝達は、半導体集積回路の場合、電位の高低により行われるが、RSFQ回路では、パルスを用いて行われる。加えて、RSFQ回路では、論理ゲートはクロック入力を持っておりゲートがクロックに同期して動作する。このクロックパルス入力が定めるクロックサイクル内にデータ入力ポートへパルスが到着した場合、入力された論理値は1、到着しない場合、論理値は0となる。

半導体集積回路と同様に、RSFQ回路でも製造された回路の機能が正しく実装されているかどうかテストされる。ここでテストは、テストパターンと呼ばれる信号を回路に入力し、その応答を解析することで行われる。近年の集積回路製造において、大規模な回路に対する高品質なテストが要求されており、可能な限り少ないテストパターンで、発生しうる多くの故障を検出・診断すべきである。このことから、そのようなパターンを生成可能なテストパターン生成手法を計算機で実行し生成する。

RSFQ回路では、スイッチング速度がピコ秒レベルと高速であるので、配線のばらつきや動作環境の影響が相対的に大きい。そのため、ゲートの入力においてパルスの到着順が変わることにより設計とは異なるとゲートの機能を持つことが高い頻度で発生する。先行研究において、半導体回路にはないこの「ゲートへのパルスの到着順が設計と異なること」をタイミング故障と定義しており、そのテストパターン生成手法を提案している。生成過程で、回路に発生するタイミング故障を再現する故障シミュレーションを、膨大な回数実行する必要があるが、故障シミュレーションの実行は計算コストが高いため、大規模な回路のテストパターン生成には最適化が必須である。

本研究では、RSFQ回路の故障診断のためのタイミング故障シミュレーションを高速化し、テストパターン生成の実用性向上を目指す。本報告で

は、SFQ 回路の単一ゲートにおけるパルスの到着順が 1 箇所入れ替わる故障を考える。

故障シミュレーションでは、正常な回路の動作と故障した回路の動作との両方をシミュレーションする。その際、本手法では両回路の動作の類似性に着目し、故障回路のシミュレーションでは故障が影響し得る領域のゲートのみをシミュレートし、重複した計算を削減する。また、RSFQ 回路ではゲートがクロック入力を持つため、組み合わせ回路はパイプライン動作を持つ。この性質から、各クロックサイクルでテストパターン生成に関連のないステージが存在する。本手法では、これらのステージに属するゲートのシミュレーションを省略し高速化を行う。

本手法の効果を評価するため、既存の組み合わせ回路データセットを用いて速度比較実験及び提案手法を用いたテストパターン生成の生成時間の測定を行った。本手法は未改良の方法と比較して大幅な速度向上を達成し、実験で用いた回路データにおいて現実的な時間でパターン生成が可能であることが示せた。

Acceleration of Timing Fault Simulation for Fault Diagnosis of Single-Flux-Quantum Logic Circuits

Hiroki Watanabe

Abstract

In recent years, single-flux quantum circuits (RSFQ) have been actively studied as an alternative to CMOS circuits for high-speed and low-power logic circuits. SFQ circuits are expected to follow the same trajectory as the evolution of semiconductor integrated circuits, with circuits becoming larger and larger in the future, and microprocessors and other arithmetic circuits based on RSFQ circuits have been designed and prototyped so far.

In the case of semiconductor integrated circuits, information is transmitted between gates and other elements by means of high and low electric potentials, but in RSFQ circuits, information is transmitted by means of voltage pulses. In addition, in RSFQ circuits, logic gates have clock inputs, and the gates operate in synchronization with the clock. If the pulse arrives at the data input port within the clock cycle specified by the clock pulse input, the input logic value is 1. If the pulse does not arrive, the logic value is 0.

As with CMOS integrated circuits, RSFQ circuits are also tested to ensure that the functions of the manufactured circuit are implemented correctly. The test is done by inputting a set of signals called a test pattern into the circuit and analyzing the response. In recent years, high quality testing of large-scale circuits is required in integrated circuit manufacturing, and as many possible failures as possible should be detected and diagnosed with as few test patterns as possible. For this reason, a test pattern generation method that can generate such patterns is implemented and generated by a computer.

In the RSFQ circuit, since the switching speed is as fast as picosecond level, the effects of wiring variation and operating environment are relatively large. Therefore, it is highly likely that the gate functions differently from the design due to the change in the order of arrival of the

pulses at the gate input. In a previous study, we defined a timing fault as “a difference in the order of arrival of pulses to the gate from the design”, which is not found in semiconductor circuits, and proposed a method for generating test patterns. In the process of generating the test pattern, it is necessary to run a large number of fault simulations to reproduce the timing faults that occur in the circuit, but since running fault simulations is computationally expensive, optimization is essential for generating test patterns for large-scale circuits.

This research aims to improve the practicality of test pattern generation by accelerating the simulation of timing faults for fault diagnosis in RSFQ circuits. In this report, we consider a fault in which the order of arrival of pulses in a single gate of an SFQ circuit is switched at one point.

In the fault simulation, both the normal circuit operation and the operation of the fault circuit are simulated. In this method, we focus on the similarity of the behavior of the two circuits, and in the simulation of the fault circuit, only the gates in the region that can be affected by the fault are simulated to reduce duplicate calculations. In addition, since the gates of the RSFQ circuit have clock inputs, the combined circuit has pipeline operation. Due to this nature, there are stages in each clock cycle that are not related to the test pattern generation. In this method, the simulation of the gates belonging to these stages is omitted to improve the speed.

In order to evaluate the effectiveness of the proposed method, we conducted speed comparison experiments and measured the generation time of test pattern generation using the proposed method on an existing combinational circuit data set. The proposed method achieved a significant speedup compared to the naive method, and it was shown that the patterning generation could be done in a realistic time for the circuit data used in the experiments.

目次

1	はじめに	1
2	準備	3
2.1	単一磁束量子回路	3
2.2	論理表現	5
2.3	クロッキング方式	6
2.4	回路記述と論理シミュレーション	6
2.5	タイミング故障	8
2.6	回路テストとテストパターン生成	10
3	関連研究	11
3.1	同時故障シミュレーション	11
3.2	単一磁束量子回路のタイミング検証と遅延テストの自動テストパターン生成手法	11
3.3	故障検出のためのタイミング故障シミュレーションの高速化	12
4	提案手法	13
4.1	単一磁束量子回路におけるタイミング故障のテストパターン生成手法	13
4.1.1	概要	13
4.1.2	先行研究で用いられる故障シミュレーション手法	14
4.2	タイミング故障シミュレーションの高速化	16
4.2.1	概要	16
4.2.2	ファンアウトコーンのみを計算する手法	17
4.2.3	パイプライン動作の特性を利用する高速化手法	19
4.2.4	両手法を併用した高速化	21
5	性能評価	21
5.1	評価方法と使用する回路データ	21
5.2	小さい回路での評価	22
5.3	大きい回路での評価	24
6	まとめ	25
	謝辞	25

目 次

2.1	ジョセフソン接合	4
2.2	ジョセフソン接合の回路	5
2.3	サンプル RSFQ 回路	7
2.4	正常な AND ゲートの動作	9
2.5	タイミング故障を起こした AND ゲートの動作	9
4.6	故障シミュレーションでの仮定故障集合の分割	15
4.7	故障影響範囲	18
4.8	クロックサイクル 3 における回路の論理値分布	20
5.9	回路変換の例	22

表 目 次

5.1	評価に用いた回路の基本的なパラメータ	23
5.2	c17での測定	23
5.3	c432での測定	24

1 はじめに

昨今の発展を続ける情報化社会は、半導体集積回路技術によって創造されているといっても過言ではない。半導体デジタル回路のプロセッサ製造技術は特に進化しており、動作クロック周波数は 3GHz を超え、集積度も 1 チップ当たり 40 億トランジスタに達している。更に加えてマルチチップ構成等の実装技術によって、多くのコアが搭載でき大量のデータを高速に処理できるプロセッサが安価に製造されている。半導体集積回路技術は素子のサイズを小さくすればするほど高性能になるというスケーリング則にしたがって進化してきたが、そのスケーリング則も限界が見えつつあり、集積度に依存した性能向上は近い未来に限界を迎えることが広く懸念されている。また、大規模な計算システムを行うデータセンタなどでは、維持のため消費電力が運用コストの大きな割合を占めており重大な問題となっている [1]。

そのため近年、次世代の集積回路として超伝送単一磁束量子 (Rapid Single-Flux-Quantum: RSFQ) 回路が注目されている。RSFQ 回路は数十 GHz に及ぶ高速動作が可能であり、半導体集積回路と比較して数百倍以上の電力効率を持つ。現在、RSFQ 回路を用いた様々な演算器やプロセッサの設計と試作が行われている段階であり、配置配線の自動化や新たな設計記述方式等の研究も行われ、将来的に半導体集積回路と同様に大規模化が見込まれる。

しかし、RSFQ 回路は半導体集積回路とは異なる原理に基づいて動作することから、異なるタイプの故障に注目する必要がある。そのような故障に対して現在は主に手動によるテスト作業が行われているが、回路の規模に応じた多大な時間を要するため、大規模な回路においては別の手段を模索する必要がある。このような経緯から、SFQ 回路特有の故障に対するテストの自動化によるし検証コストの抑制が求められる。

一般に論理回路のテストでは、製造された回路に入力パターンを印加し、それに対応した出力を観測し、設計で期待される出力と比較を行う。この際に印加されるパターンをテストパターンと呼ぶ。手動でのテストパターン作成によるテストが行われる場合があるが、確実に故障を検出できるかどうかかわからず、回路規模に応じた多大な時間が必要である。半導体集積回路においては、発生する故障を故障モデルとして抽象化してとらえ、そのモデルに基づき自動でテストパターン生成をする手法が開発されている。しかし、半導体集積回路と SFQ 回路では動作原理や動作速度、超伝導下での動作環境など様々な条件が異なることから、SFQ 回路固有のテ

ストパタン生成技術が必要である。

SFQ 回路のゲートはパルス論理に基づいて動作する。ゲートはすべてクロック入力を持ち、隣接する 2 つのクロックパルスが表すクロックサイクル期間内にパルスが到着したか否かで論理値を表す。このような動作原理に加え、高速な動作により、パルスの時間的な歪みは故障の原因になる。製造時のゲート特性のばらつきや動作環境によりゲートへのパルスの到着時刻は変動しうる。パルスの到着時刻が変化すると設計に比べゲートへのパルス到着順序が変わり、ゲートの機能が損なわれることがある。これをタイミング故障という。

テストパタン生成手法は多くの場合その処理が主に 2 つの部分に分かれる。シミュレーションベース処理とアルゴリズムベース処理である。シミュレーションベース処理では、回路に対して入力されるパタンをランダム生成し、そのパタンがどれほどの故障を検出できるかを故障シミュレーションにより算出する。有効なパタンを選択していくことで効率的なテストパタン列を生成する事ができる。シミュレーションベースではランダムに生成したパタンを用いるため、その方法では検出が困難な故障も存在する。そのような故障に対してアルゴリズムベース処理では、故障の影響の伝播と値の代入を繰り返し、確実に故障を検出するパタンを生成する。

タイミング故障のテストパタン生成も同様に故障シミュレーションを用いる。この故障シミュレーションは、回路が大規模になればなるほど多くの故障についてシミュレーションが必要になり、シミュレーション 1 回あたりの計算量も多くなる。その結果、計算時間が現実的な範囲では収まらず実用性を失ってしまう。この問題を解決するため、半導体集積論理回路上での故障シミュレーションを効率化する手法は、既に多くの研究で改善が繰り返されており、実用化されている。しかし、RSFQ 論理回路と半導体集積論理回路では故障シミュレーションの際に用いられる論理シミュレーション手法が異なる。よって、RSFQ 論理回路専用の故障シミュレーション手法が必要となる。大規模な RSFQ 論理回路に対してテストパタン生成が現実的な時間内で行うことができれば、今後の RSFQ 大規模回路設計において、必要な時間的コストを削減することができる。

本研究における貢献は、RSFQ 回路と論理シミュレーション手法の特性を踏まえた故障シミュレーションの高速化を行うことにより、現実的な計算時間で故障診断のためのタイミング故障のテストパタン生成を可

能とした点である。

本論文の構成は以下の通りである。第 2 章では、RSFQ 回路の特性や RSFQ 回路固有の故障、及び、回路のテストについて述べる。第 3 章では、従来の半導体集積回路のテストにおいて一般的な故障シミュレーション高速化手法と、RSFQ 回路における既存のテストパターン生成手法を、故障検出のための高速化手法について、本研究の関連研究として挙げる。第 4 章では、先行研究のテストパターン生成手法について説明と問題点の分析を行い、それを踏まえ提案手法の説明を行う。第 5 章では、公開されたデータセットと、実際に提案手法及びそれを導入したテストパターン生成手法を実装したプログラムを用いて、評価、考察を行う。最後に、本論文のまとめ、今後の課題を第 6 章で述べる。

2 準備

2.1 単一磁束量子回路

電気抵抗が存在する物質中では、電子がある一定距離を移動すると不純物や結晶格子に衝突する。その際、本来持っている量子力学的な波としての性質を失い、粒子として振る舞う。半導体集積回路は電子の粒子的振る舞いを制御することで動作している。常温下での動作を前提とする半導体集積回路において、信号が伝搬する配線の伝導体には電気抵抗が存在し、電子の移動を阻害している。一方、超伝導状態の物質中は電気抵抗は完全にゼロとなるので、高速に動作できるようになる。

単一磁束量子回路 (Single-Flux-Quantum: RSFQ 回路) はそのような超伝導の性質を利用した回路の一種である。この RSFQ 回路は図 2.1 で示すように、能動素子としてジョセフソン接合を挿入した構造と超伝導体のリングからなる回路である。

ジョセフソン接合 (Josephson Junction: JJ) は 2 つの超伝導体を絶縁体の薄膜を挟んだ接合のことである。この接合ではあるしきい電流より小さい電流を電圧降下無しで通電し、より大きな電流は流しにくいという直流ジョセフソン効果という現象が発生する。このジョセフソン効果を用いて、しきい電流値や通電する電流量を操作することにより、通電・非通電を制御することができる。現在の集積回路の主流となる接合方法はニオブとアルミ酸化膜を用いた Nb/AlO_x/Nb 接合であり、これを利用

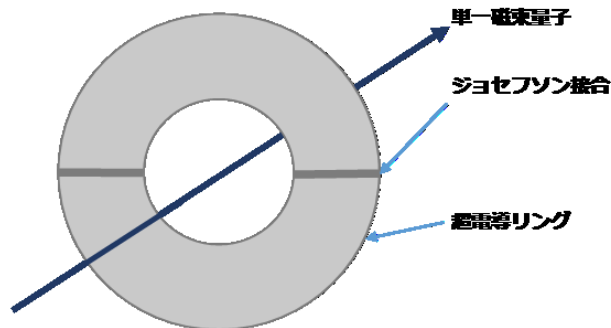


図 2.1: ジョセフソン接合

したアドバンスドプロセスや SFQ^{5ee} という製造テクノロジーにより、プロセッサ試作も行われている [3][4]。

超伝導体のリングは単一磁束回路において情報を保持するために利用される。超伝導体をリング状に形成しその中に電流を流した場合、リングを貫く磁束は $h/2e$ (h はプランク定数、 e は電子一つ当たりの電荷) の整数倍の値しかとることはできない。この現象は磁束の量子化と呼ばれ、このリングを貫く磁束自体を磁束量子と呼ぶ。RSFQ 回路においては超伝導するリング内に磁束がある状態を“1”、ない状態を“0”として扱う。

図 2.2 に示しているのは、もっとも簡単な RSFQ 回路の 1 つのジョセフソン伝送路 (Josephson Transmission Line: JTL) である。この回路は左端の端子からパルス入力を受け取り、右端の端子からパルスを出力するという機能を持っている。左端の端子はパルス入力を受け取ると、ジョセフソン効果によりリング 1 に磁束量子が発生する。リング 1 に磁束量子が発生した後、リング 1 で発生する電流がリング 2 の JJ に対してジョセフソン効果が発生させ、リング 2 に磁束量子が移る。その後、リング 2 の磁束量子に伴って現れる回転電流が右端の端子に出力されるパルスを発生させる。このようにして、超伝導体のリングを磁束量子がパルスを介して移動していくことにより、素子間の情報伝達を行う。

磁束量子が移動する際に JJ において通電・非通電を切り替えるスイッチングが発生する。消費電力の発生はこのスイッチングの瞬間のみであり、1 回あたり 10^{-19} J 程度ととても小さく、超伝導が発生する極低温という動作条件を考慮しても非常に高い電力効率を実現することができる。

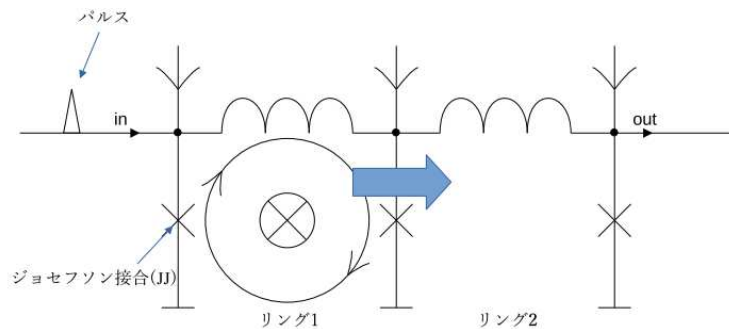


図 2.2: ジョセフソン接合の回路

また、動作速度に関しても、スイッチング速度が数ピコ秒と速く、回路の高速動作を可能にしている。

2.2 論理表現

現在一般的に使用されている半導体集積回路では、電圧の大きさによって“1”、“0”の論理値を表している。それと比較し、RSFQ回路では素子間の情報伝達の媒体として電圧パルスを用いる。したがって、パルスの到着未着で論理値を表現する原理上、論理値の決定期間を定めなければならない。このことが考慮されたパルスを用いる論理表現の方法は主に二線式とクロック同期式が存在する。

二線式は“1”を表す信号線と、“0”を表す信号線を2つ用意し、どちらにパルスが到着したかにより論理値を表現する。

クロック同期式はゲートに対してクロックパルスを供給する。そのクロックパルスはクロックサイクルを形成し、ゲートのデータ入力ポートに対して、クロックサイクル内にパルスが到着すれば“1”、しなければ“0”として論理値を表現する。

後者ではゲートがクロックパルスに同期して動作する用に設計されており、前者に比べ配線数が少なく回路面積が小さくなる。しかし、ゲートに対するクロックパルスやデータパルスの遅延を調整するタイミング設計を要求する。厳密なタイミング設計が行えれば高速な回路を設計可能だが、非常に煩雑な作業を要求する。

本報告では、RSFQ回路のデジタル設計で主に研究が進められているクロック同期式の回路のみ扱う。

2.3 クロッキング方式

先述のように、一般的に用いられるクロック同期式では各ゲートにクロックを供給する必要がある。RSFQ 回路では、高い動作周波数を前提とした回路のレイアウトを行うため、配線の遅延調整等に厳しい制約がかかる。この制約により、クロック同期式で RSFQ 回路を設計する場合、全ゲートに対して同一の時刻にクロックパルスを供給することは困難である。そのため多くの場合、ゲートごとに異なるクロックタイミングで動作するよう設計する。このゲートへのクロックパルスの供給方式には、代表的な方法としてコンカレントフロー方式、クロックフォロワーデータ方式がある。

コンカレントフロー方式では、データパルスよりもクロックパルスが先に到着するようタイミングを調整する。この方式では回路に対する 1 度のクロック入力でデータパルスが 1 ゲート進むことになる。言い換えれば、ゲート 1 つにつき 1 段分のパイプラインステージが構成される。コンカレントフローでは n 段の回路に対し 1 つの入力ベクトルを印加した場合、 $n+1$ 回のクロックパルス入力が必要である。

クロックフォロワーデータ方式では、データパルスが先に到着し、クロックパルスが遅く到着するようタイミングを調整する。この方式は、データパルスに対してクロックパルスが追従するようタイミング調整と配線を行う。これにより、1 度のクロックパルス入力によりデータパルスがすべてのゲートを通過する。

クロックフォロワーデータ方式では、1 クロックサイクル間に複数の論理ゲートを通るため、入力からその出力応答までにかかるクロックサイクルが少なく、コンカレントフロー方式に比べスループットが高くなる傾向にある。しかし、パルスの遅延設計等のタイミング制御が難しいため、それぞれの特徴を加味し回路に合わせて適切にクロッキング方式を選択していく必要がある。

2.4 回路記述と論理シミュレーション

従来の半導体集積回路と異なり、RSFQ 回路ではパルスによりゲート間の情報伝達が行われているため、クロッキング方式を指定し、パルスの到着順序制約を設計時にゲートに対して与えなければならない。したがって、半導体回路設計で使用されるハードウェア記述言語とは異なった特有の、ゲートへの入力配線にパルスの到着順位を数字で併記し、数

字が小さいほど早く到着するよう制約が定められる記述形式が提案されている。

鬼頭らが提案する RSFQ 回路の回路記述方式では、ゲートへの入力配線に対してパルスの到着順位制約を付加する構文を導入している [6]。例として図 2.3 に示す回路は以下のように記述される。

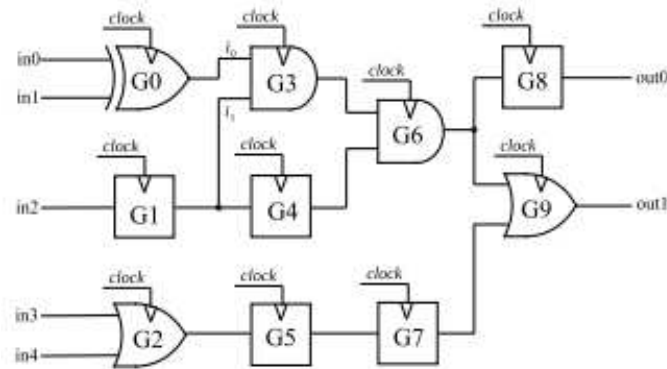


図 2.3: サンプル RSFQ 回路

$$t0 = XORG0(in0@1, in1@1, clk@0)$$

$$t1 = DFFG1(in2@1, clk@0)$$

$$t2 = ORG2(in3@1, in4@1, clk@0)$$

$$t3 = ANDG3(t0@1, t1@1, clk@0)$$

$$t4 = DFFG4(t1@1, clk@0)$$

$$t5 = DFFG5(t2@1, clk@0)$$

$$t6 = ANDG6(t3@1, t4@1, clk@0)$$

$$t7 = DFFG7(t5@1, clk@0)$$

$$out0 = DFFG8(t6@1, clk@0)$$

$$out1 = ORG9(t6@1, t7@1, clk@0)$$

この研究では新たな記述方式に伴いその方式で記述された回路の論理シミュレーション手法も提案している。

この手法では、全配線にシミュレーションされるクロックサイクル数の長さで論理値系列を割り当て、ゲートの内部状態更新とその論理値系列への値の代入を繰り返すことで回路の挙動をシミュレーションする。RSFQ回路では、ゲートがクロックポートを持ち、入力クロックパルスに同期して動作する。よって、クロックサイクルごとにゲートが異なる状態を保持することになる。このため、シミュレーションアルゴリズムでは、クロックサイクルごとに全ゲートをシミュレーションする。ゲートをシミュレーションする際、RSFQ回路の組み合わせ回路は入力側から順に出力が決定していくため、回路のトポロジーに基づき入力側からのゲートからシミュレーションする必要がある。

実際の実装では、パフォーマンスの観点から、ゲート記述の系列をトポロジーに基づき回路の入力に近いゲートからシミュレーションされるようソートする。また、ゲートの内部状態更新と出力の処理を行う際、クロックパルスよりパルスが早く到着する配線の論理値と遅く到着する配線の順で処理することで、コンカレントフロー方式とクロックフォロワーデータ方式で異なるゲート機能のシミュレーションを可能にする。先行研究のRSFQ回路におけるタイミング故障のテストパターン生成手法でも、この論理シミュレーション手法が使用されている。

2.5 タイミング故障

半導体集積回路におけるタイミング故障は、配線や素子の遅延が増加してフリップフロップに取り込まれる際に、許容時間を超過してしまうことで、論理的動作が変化することである。タイミング故障には、各故障箇所において信号の0から1への立ち上がりの遅延と1から0への立ち下がりの遅延がある。

一方、SFQ論理回路は半導体回路と異なりパルス論理を用いて動作するため、タイミング故障時の回路論理の振る舞いが異なったものとなる。SFQ回路においてもそのような信号の伝達が設計で与えられた制約を満たさない故障が発生するが、SFQ回路は半導体回路が異なる動作原理を持つことから、異なる故障モデル化を求められる。

SFQ回路では製造時のばらつきや極低温での動作環境等の物理的原因によりパルスの遅延がずれ、設計とは異なるパルスの到着順序となる可能性がある。コンカレントフロー方式とクロックフォロワーデータ方式でゲートへのパルスの到着順序には設計上で異なる制約が指定される。そ

のため、なんらかの原因で指定の到着順序とは異なるパルス入力があった場合、ゲートの機能が変わってしまう。[2] ではこの「ゲートへのパルスの到着順が設計と異なること」を SFQ 回路におけるタイミング故障として定義している。

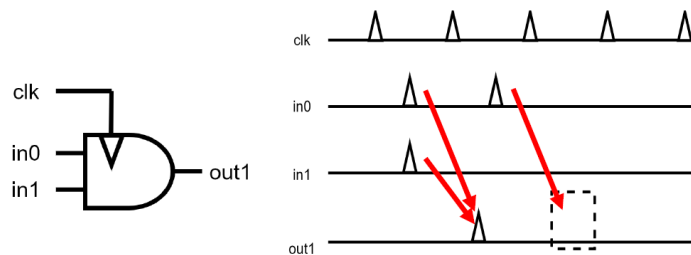


図 2.4: 正常な AND ゲートの動作

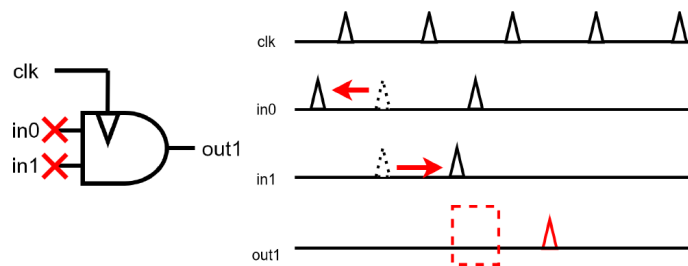


図 2.5: タイミング故障を起こした AND ゲートの動作

図 2.4 に AND ゲートの正常な動作、図 2.5 にタイミング故障が発生した時の動作を示す。図 2.4 中の回路で発生するタイミング故障で in0 が 1 クロック早着である場合を考える。。この故障により、本来出力されるはずだった AND ゲートのパルスが出力されなくなり、ゲート機能が変化してしまっている。また、in1 が 1 クロック遅延した場合、この故障により、本来出力されないはずだった AND ゲートのパルスが出力されてしまっており、同様にゲート機能が変化してしまっている。

このような 2 入力のゲートに対して発生するタイミング故障を、各入力ポートに対するパルスの到着時刻の不等式として定式化したものを以下に示す。

$$i0(t) < \mathbf{clock}(t) < in1(t) < \mathbf{clock}(t + 1), or,$$

$$in1(t) < \mathbf{clock}(t) < in0(t) < \mathbf{clock}(t + 1), or,$$

$$in0(t), in1(t) < \mathbf{clock}(t) < \mathbf{clock}(t + 1), or,$$

$$\mathbf{clock}(t) < in0(t) < \mathbf{clock}(t + 1) < in1(t), or,$$

$$\mathbf{clock}(t) < in1(t) < \mathbf{clock}(t + 1) < in0(t), or,$$

$$\mathbf{clock}(t) < \mathbf{clock}(t + 1) < in0(t), in1(t), or,$$

$$in0(t) < \mathbf{clock}(t) < \mathbf{clock}(t + 1) < in1(t), or,$$

$$in1(t) < \mathbf{clock}(t) < \mathbf{clock}(t + 1) < in0(t).$$

この不等式上における $in0(t), in1(t), clock(t)$ はクロックサイクル t におけるパルスの到着時刻を表す。図 2.5 に示すタイミング故障は 1 行目と 4 行目の太字で示された不等式によりそれぞれ早着、遅延の故障が定式化されたものである。本研究ではこのタイミング故障に対するテストパターン生成手法の改良を行う。

2.6 回路テストとテストパターン生成

回路のテストとは、回路にあるパターンを入力し、それに対する出力を観測することで、設計上期待した出力が得られるかどうかの確認をすることである。テストの際、実際に回路に入力される入力パターンをテストパターンと呼ぶ。

故障テストでは主に故障の検出と診断があり、故障検出では回路に対する故障の有無の検証、故障診断では故障箇所や故障の種類を目的としている。それぞれ、故障検出のためのテストでは、テストパターン列は短いが故障箇所の特定はできない、故障診断のためのテストでは、テストパターン列は長いが故障箇所の特定ができるという特徴を持っている。

一つのテストパターンによって検出や診断ができる故障は限定的な数でしかない。そのため複数のテストパターンを回路に入力する事が、想定される故障を全て検出・診断する際に求められる。

RSFQ 回路でのテストパタンの設定は、手動でのアドホックな方法が現在主流であるが、体系的な方法ではないため全ての故障を発見出来るとは言えない。また、大規模な回路に対して手動でのテストパターン生成は不可能である。そのため、計算機で実行可能な体系的テストパターン生

成 (Automatic Test Pattern Generation: ATPG) が必要である。半導体集積回路では、多くの研究が進んでおり、RSFQ 回路に対しても応用可能な技術も存在している。

RSFQ 回路においても半導体集積回路と同様に様々な故障原因があり、その原因の数だけ故障はモデル化されている。大規模な回路では同一モデルの故障でも発生する可能性のある箇所は多く、仮定故障は膨大な数存在する。即ち、膨大な数の仮定故障をテストする必要がある大規模回路のテストでは、テストパタンの増大が避けられない。そのテストパターン全体を回路に対して印加しそれぞれの出力を調べる必要があるため、テストパタンの巨大化はテストコストの増大に直結する。したがって、テストパターン生成手法には、可能な限り多くの故障を検出・診断可能な小さなテストパターンを生成する能力が要求されている。

3 関連研究

3.1 同時故障シミュレーション

パターン生成においては、故障を再現するシミュレーションが膨大な回数実行される。Ulrich は正常な回路と故障が発生した回路の挙動類似性に注目して故障回路のシミュレーションは正常回路とは異なる振る舞いのみをシミュレーションすることで計算量を削減する手法を提案している [7]。RSFQ 回路の故障シミュレーションでも大規模な回路では計算コストは高い。しかし RSFQ 回路では半導体集積回路とは異なる論理シミュレーションアルゴリズムを必要とすることから、それを考慮した故障シミュレーションアルゴリズムを構成が要求される。

3.2 単一磁束量子回路のタイミング検証と遅延テストの自動テストパターン生成手法

RSFQ 回路においてパルスの到着時刻が設計から大きく変化し回路が故障する問題は非常に一般的である。[8] では製造段階で発生する回路素子の特性のばらつきが遅延特性にどれほど影響するかを分析している。分析では、ゲートを構成する JJ の面積や超伝導ループのインダクタンスのばらつきをガウス分布でモデル化し、モンテカルロシミュレーションを用いて遅延の確率分布を計算している。また、あるゲートの遅延にそ

のゲートに inputs するパルスの遅延が影響する Timing bleed についても遅延時間を分析している。それらの分析結果を踏まえこの研究では Timing bleed を考慮したテストパターン生成手法を提案している。このテストパターン生成手法は D-algorithm をベースに作られており、回路の配線に対して値の割り当てを繰り返し、故障の影響を出力まで伝搬させていくことでパターンを生成するものである。本研究では、Timing bleed については取り扱わない。

3.3 故障検出のためのタイミング故障シミュレーションの高速化

準備でも述べたように故障検出とは故障の有無を検証するテストであり、タイミング故障を検出するため、入力パタンのペアを生成することでテストパターン列を生成する手法を提案されている。

入力がペアであるのは、タイミング故障が時間的に論理動作を変えてしまうものであるためである。ペアであることによって、ゲートの一方の入力が過去または未来方向にずれたとしても、他方が決定されているため、故障した出力を確実に得られる。生成された入力パタンのペアを繰り返し故障シミュレーションに入力し、出力から検出率を検証する事でテストパターン列を得る。

大規模な回路において故障シミュレーションの試行回数が増大することが想定されることから、テストコストの削減を目的とした故障シミュレーションの最適化を提案している。検出と比較して診断では更なる故障シミュレーションの回数が求められ、テストパターン列が長くなることから故障診断のための故障シミュレーションの高速化が不可欠である。しかし、故障の有無の検証に加え故障箇所や原因の特定を行うためそれを踏まえたパターン生成を構成する必要がある。

4 提案手法

4.1 単一磁束量子回路におけるタイミング故障のテストパターン生成手法

4.1.1 概要

故障診断とは、準備で述べたように、故障シミュレーションの出力から故障箇所を一意に判別することである。目的は、単一のタイミング故障を含む対象回路の出力を基に、故障箇所と種類を一意に診断できる最短のテストパターン列を生成することである。先行研究で提案される故障診断方法は、検証したい回路データと設計者が与えた想定故障を入力し、冗長故障以外の故障を診断できるテストパターン列を出力するものである。本手法が行う手順は以下の通りである。

1. ランダムに3組みのテストパターン列を生成し、連続して回路に印加する。故障シミュレーションにより、応答不一致によって全ての仮定故障の集合の分割をする。
2. まだ選択されていない、分割された部分集合の中で、長さが最大である部分集合が最小となる3つのテストパターン列 S_0 (p_0, p_1, p_2) を選択する。
3. 一次入力パターン列 S_0 から新たなパターン列をランダムに生成する。先頭または末尾に新たなパターンを生成した遷移 (p_{-1}, p_0, p_1) (または (p_1, p_2, p_3)) を回路に入力した出力によって、1. で分割された部分集合を更に分割し、長さが最大である部分集合が最小となるパターン遷移が選択され、その際採用された遷移から p_{-1} または p_3 が S_0 の先頭または末尾に追加される。追加されて4列になったパターン列を S_1 として保持する。
4. 想定されるすべての故障を判別できれば終了。想定される故障のほとんどが判別できる場合は、手順6に進む。
5. 保持したパターン列の故障診断率を計測し設定した閾値を超えない場合3. の手順をを繰り返しパターン列を追加する。追加を繰り返しても診断率が変わらない場合、2. に戻り、新たに特定された故障がある場合は同じ処理を繰り返す。

6. 想定される故障のほとんどが特定された場合と、ステップ1およびステップ3で生成されたランダムパターンにより新たな故障が特定されない場合があるが、いずれにしても SAT ベースのアルゴリズムを用いて残りの故障のサブセットを特定し、処理を終了する。

任意の故障の組を区別するためには、想定される故障の一次出力がすべての故障の組で異なることを条件とする回路の出力が必要である。区別する故障の組が大きいと回路が非常に大きくなるため、想定する故障の数に制限を設けることを想定している。

1.2. で、故障シミュレーションを行う事で、最も有効で効率的なテストパターン遷移を選出し初期列を生成する。3. で現在の列の末尾または先頭と連続して印加するパターンを生成しているのは、現在生成済みの要素を再利用することで、テストパターン列の長さを削減するためである。故障シミュレーションを用いたパターン選択を予め設定する十分大きい故障新案率を超えるまで実行するため、大規模回路に対するテストパターン生成を行う場合、膨大な回数の故障シミュレーションが行われることとなる。次節でより詳しい計算量について故障シミュレーションの方法とともに説明する。

本報告では、各ゲートがコンカレントフロー方式またはクロックフォロデータ方式 (混成可) でクロック供給されるフィードバックを含まない回路に対して、1 サイクルずれる故障を考える。また、先行研究では故障診断率を超えた後、残りのパターン列を SAT ベースのアルゴリズムを用いて導出しているが、今回シミュレーションベースでのテストパターン生成の高速化を主としているため、そのまま終了させる処理を行っている。

4.1.2 先行研究で用いられる故障シミュレーション手法

故障検出のためのテストパターン生成では、故障シミュレーションを用いて生成したパタンの故障検出能力を測定し、効率のよいパタンの選定に利用するが、故障診断テストパターン生成では、故障シミュレーションを用いて生成したパターンによって仮定故障の集合を分割することで、効率のよいパタンの選定を行う。

この故障シミュレーションでは、入力されたパターンに対して検証対象回路の挙動を、設計通り正常に動作した場合と、故障が発生した場合両方シミュレーションし、故障の影響が回路の出力に影響するかどうかを判定する。

故障が回路の出力に影響していれば、そのパターンによりその故障を検出可能となり、出力が単一のものであるとき診断可能であるという。仮定故障集合の分割は、図 4.6 に示すように正常回路と仮定故障に対応する故障回路を用意し、その回路すべてを論理シミュレーションアルゴリズムでシミュレーションする事で行う。

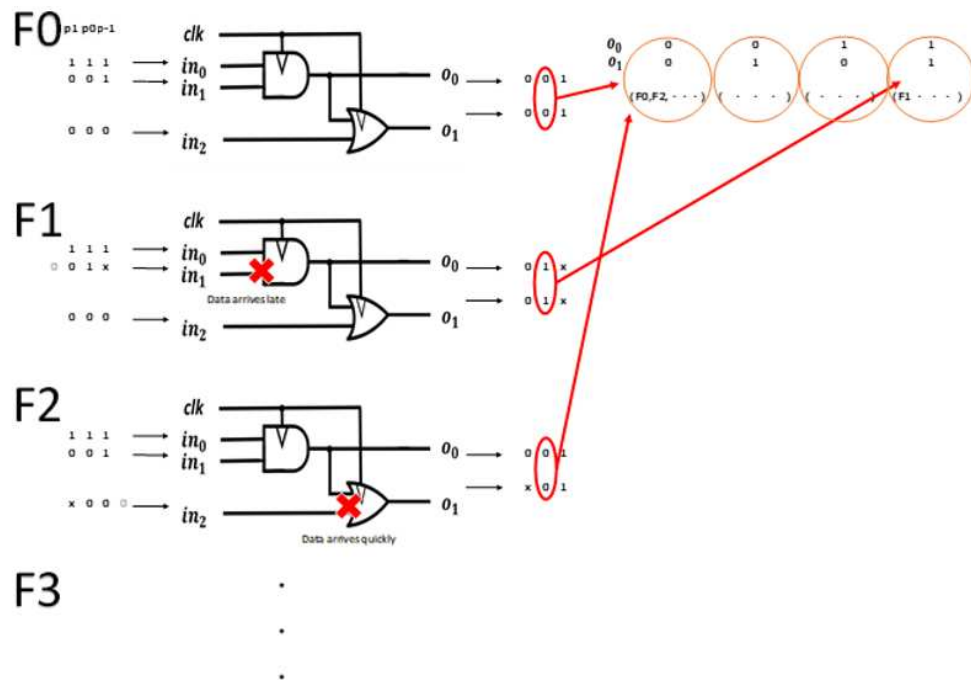


図 4.6: 故障シミュレーションでの仮定故障集合の分割

用意した正常回路 F0 に対し、それぞれの故障の起こりうる早着または遅延するタイミング故障を再現した故障回路を、故障 1 つごとに F1, F2, F3, ... のように用意する。この用意した正常回路、故障回路に対しランダムに生成した 3 組のパターン遷移を印加し、その出力を比較する。

図 4.6 のような回路 (in_0, in_1, in_2) に対してあるパターン遷移 $p_{-1}(1,1,0) \rightarrow p_0(1,0,0) \rightarrow p_1(1,0,0)$ を入力する場合を考える。この際、3 遷移を入力するのは、タイミングのズレによる出力の誤差のない出力を得る必要があるためであり、確実に 2 遷移目を取得することができる。それぞれの正常回路、故障回路について、2 遷移目に対応する出力を比較する。正常回路 F0 の 2 遷移目に対応する出力 (o_0, o_1) が (0,0) であるのに対し、故障回路 F1 の 2 遷移目に対応する出力は (1,1) であり、故障があると判別で

きる。また故障回路 F2 に関しては 2 遷移目に対応する出力が F0 と同じ (0,0) でありこの比較だけでは判別ができない。このようにそれぞれ取得した出力結果ごとに、仮定故障の集合を分割していく。分割した仮定故障の集合の中で最も要素数が大きい分割の長さを指標とし、また次のランダムに生成するパターン遷移と比較し最大の長さが最小であるものを採用する。この作業をループさせ分割を繰り返し、ある出力に相応する仮定故障集合の分割の長さが 1 になった時、その仮定故障は診断可能であるといえる。

準備の章で説明したように、単一磁束量子回路には半導体集積回路とは異なる論理シミュレーションアルゴリズムが必要である。提案されている論理シミュレーションアルゴリズムでゲート N 個を持つ回路を T クロックサイクル分シミュレーションする場合、計算量は $O(NT)$ となる。この回路に対して K 個の仮定故障を故障シミュレーションするには、正常な回路に加え、 K 個の故障を再現する回路をシミュレーションする必要がある。これにより、故障シミュレーションの計算量は $O(KNT)$ となる。

図 4.6 で示したものはあるランダムに生成された 3 組のパターンの故障シミュレーションであり、実際は、優秀なパターン遷移を発見するまで、生成されたパターンをこの再現された故障回路の数だけ繰り返さなくてはならない。大規模な回路では、仮定する故障の数も増加し再現する故障回路も増え、その回路のシミュレーション自体も時間がかかり、仮定故障も膨大な数存在するため、テストパターン生成では無視できない計算コストがかかる。また、テストパターン生成アルゴリズム自体がランダム生成を用いているため、テストパターン列の長さの削減にはランダム生成するサンプル数を増やす必要がある。サンプル数が増えることにより、故障シミュレーションを実行する必要があるため、この計算コストはテストパターンの質にも関わる。

4.2 タイミング故障シミュレーションの高速化

4.2.1 概要

前述したように、先行研究の故障診断のための単一磁束量子回路におけるタイミング故障のテストパターン生成には、膨大な回数の故障シミュレーションの実行を要するため、論理シミュレーションを用いた単純な方法では多大な計算時間がかかってしまう。そこで本論文では、計算コス

トの少ない効率的な故障シミュレーション手法を先行研究のパターン生成方法に導入する。本論文で提案する故障シミュレーション手法は、既存の論理シミュレーションのアルゴリズムを変更し、タイミング故障におけるテストパターン生成では必要のない計算を省略する2つの方法によって高速化を図る。まず一つ目は、正常回路と故障回路の挙動が故障に影響される範囲以外は同じであるためのその部分の計算について再利用を行う方法である。二つ目は、SFQ回路はゲートレベルでパイプライン化されているため、クロックサイクルごとに計算する必要がないゲートが存在する。そのゲートの計算を省略する方法である。

これらの方針での高速化手法の具体的なアルゴリズムや詳しい計算量削減の原理等を次節以降で説明する。また、それらの手法は同時に併用でき、更なる高速化が可能である。その併用方法についても、両手法の説明後に述べる。

4.2.2 ファンアウトコーンのみを計算する手法

先述の Ulrich が提案する同時故障シミュレーションで故障シミュレーションを行う際、正常回路と故障回路の計算の挙動が類似しているという特徴を利用して高速化を行っている。本手法でも同様の特徴に注目して高速化を行うが、動作や挙動の違いから、RSFQ回路では特有の論理シミュレーション手法に基づいての前処理やシミュレーション計算が必要となる。

本報告では正常回路と故障回路の挙動の類似性を利用し、正常回路でのシミュレーション結果を、故障回路でのシミュレーションの際必要な部分を使いまわす。

今回改良を図る先行研究のテストパターン生成手法におけるテスト対象の回路は、組み合わせ回路であることが前提となっている。また、正常回路と故障回路の相違点は故障の有無であり、挙動の違いがその故障が発生しているゲートの出力が影響する領域でのみ起こりうる。したがって、テストパターン生成手法において故障回路の再計算を行う領域は図4.7のように故障したゲートの出力が影響するファンアウトコーンとなる。

本手法では、故障が発生したゲートのファンアウトコーンのみ故障回路のシミュレーション計算を行い、ファンアウトコーン外のゲートは計算せず、正常回路の計算を使いまわす。加えて、前述したようにSFQ回路では独自の論理シミュレーション手法が存在し、計算するゲートの集合を回路のトポロジーに基づきソートしたリストを用意する必要がある。

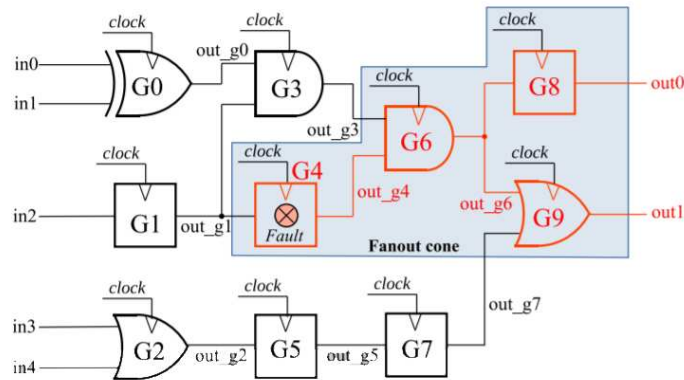


図 4.7: 故障影響範囲

よって、故障シミュレーション計算の前処理として、仮定故障毎にそれが発生するゲートのファンアウトコーンに属するゲートの集合を計算し、それを軽量な方法で並べ替える方法取る。

この前処理では、テスト対象の回路が組み合わせ回路のみであるという性質が利用できる。ゲートをノードとすることで、ゲート同士の配線を信号が伝搬する方向へ伸びる有向辺とする有向グラフと回路をとらえると、組み合わせ回路は非巡回有向グラフ (Directed Acyclic Graph: DAG) となる。あるゲートのファンアウトコーンに属するゲートの集合の計算は、ゲートのノードから到達可能なノードを列挙する問題と等価である。

回路出力から入力方向に 1 ステージずつゲートのファンアウトコーンを集合演算を用いて求めていく。あるステージの計算が終わったあと、次のステージの計算で直前に計算したファンアウトコーンを利用して計算量を削減している。

RSFQ 回路における論理シミュレーションでは、ゲートがシミュレーションされる順序が変わってしまうと、実際の回路の挙動と異なる振る舞いを計算してしまう。そのため回路のトポロジーに基づいて適切な順番でソートされたゲートの集合をアルゴリズムに与えなくてはならない。提案手法では、故障が仮定されているゲートごとにシミュレーションすべきファンアウトコーンがあり、そのファンアウトコーンそれぞれに対

してソートを行わなければならない。そこで本手法ではあるゲートが回路入力から何段目かを計算して対応辞書を構築し、その辞書を用いて大量のゲート集合を並び替える。初期化処理の終了後、故障シミュレーションを行う。

ファンアウトコーンのみ計算する故障回路のシミュレーションでは回路への入力系列、正常回路の配線集合、クロックサイクル数、假定故障集合、假定故障に対応するファンアウトコーンの辞書を受け取り、全配線の論理値系列を出力する。ここでいう假定故障とは故障が発生するゲート、そのゲートのどの配線に対して故障が発生するか、早着遅延どちらの故障が発生するかの情報を持つ三つ組みである。正常回路の配線集合を与える理由は、故障回路をシミュレーションする際、ファンアウトコーン以外の領域は正常回路のシミュレーション結果を再利用できるため、その配線の論理値を指定するために与えている。

4.2.3 パイプライン動作の特性を利用する高速化手法

提案手法の2つ目はパイプライン特性を利用した高速化である。ここではSFQ回路のパイプライン特性と論理シミュレーションにおけるパターン生成と無関係なゲートのシミュレーションについて説明する。また、それを踏まえたシミュレーション計算削減手法を説明する。コンカレントフロックロッキング方式ですべてのゲートが実装されているSFQ回路は1クロックに1ゲートずつ信号が伝搬する。この特性から、SFQ回路の組み合わせ回路は複数のパイプラインステージからなるとみなせる。パイプラインの特性から、入力は複数のクロックサイクルを経て出力に到達する。この際、入力が出力に到達するまでのすべてクロックサイクルにおいて限られたパイプラインステージだけがその入力に関連する信号を保持している。テストパターン生成では、2つのパターンをパターンの遷移として回路に入力して場合の故障シミュレーションを行う。図4.8に回路に対してパターン遷移(0, 0, 1, 1, 1)→(0, 0, 0, 0, 1)→(0, 0, 1, 1, 1)を入力した場合のクロックサイクル3における論理値の分布を示す。

セクション2.4で説明したSFQ回路における論理シミュレーション手法では、全クロックサイクルで全ゲートに関するシミュレーションを行う。よって、パターン遷移に関連した論理値を保持していないステージをもシミュレーションしていることとなる。

図4.8では、入力パターンの遷移に伴う論理値を赤色で示した。入力パターンは3クロックサイクルを経て、論理値(outg0, outg1, outg2, outg3,

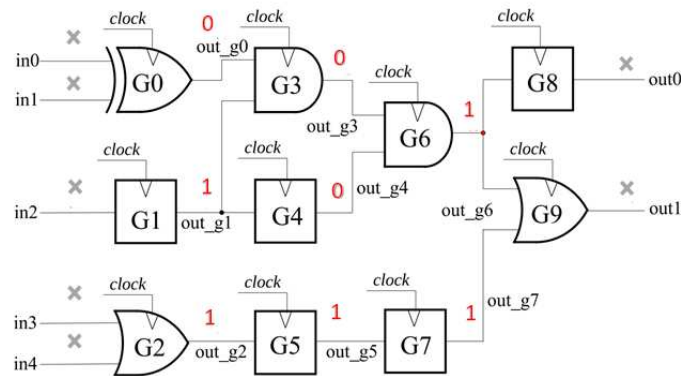


図 4.8: クロックサイクル 3 における回路の論理値分布

outg4, outg5, outg6, outg7) = (0, 1, 1, 0, 1, 1) になっている。このとき、図のようにパイプラインステージ 2, 3, 4 だけが入力遷移に関連した論理値を保持している。SFQ 回路では 1 クロックあたり 1 ゲートが信号を伝搬するので、クロックサイクル 4 では、ステージ 3, 4, 5 が関連する論理値を保持する。1 クロック進んだクロックサイクル 4 では、ステージ 3, 4, 5 が関連する論理値を保持する。関連する論理値を保持しないステージは、パターン生成に無関係である。したがって、本手法ではパターン遷移に関係しないステージの処理を、高速化のために省略する。RSFQ 回路の実装とそれぞれのクロッキング方式がわかれば、各クロックサイクルでどのステージをシミュレーションすべきか計算できる。コンカレントフロークロッキング方式では、1 クロックごとに 1 ステージ信号が伝搬するため、クロックサイクルが増加するごとにシミュレーションすべきステージが 1 ステージずつ前進していく。

回路全体がクロックフォロデータクロッキング方式で実装されていた場合、1 クロックサイクルで出力まで入力が伝搬するため、1 クロックサイクル目でシミュレーションすべきステージは全ステージである。また、2 つのクロッキング方式が混在する回路の場合、コンカレントフロークロッキング方式の回路はゲートごとにステージを区切り、クロックフォロデータクロッキング方式のゲートは連続して存在する場合に 1 ステージとしてまとめて扱う必要がある。

4.2.4 両手法を併用した高速化

先述した2つの方針に基づく故障シミュレーションの高速化は併用することが出来る。ここでは併用する際の前処理とシミュレーションについて説明する。

両手法を併用する場合、パイプラインステージからファンアウトコーンに属するどのゲートを計算すべきか軽量な方法で特定できる必要がある。そこで、パイプライン特性を利用した高速手法を用いる際のゲートとステージの対応辞書を用いて、ファンアウトコーンごとにそれに属するゲートをステージで分類する。

両手法を併用した故障シミュレーションでは、回路への入力系列、クロックサイクル数、回路記述、さらに仮定故障ごとのファンアウトコーンのゲートをステージごとに分類した集合の系列の辞書を入力して、全配線の論理値系列を出力する。この方法により各仮定故障に対するシミュレーションでは故障が影響する範囲内に含まれるパターン生成において入力されるパターンの遷移に関連するゲートの振る舞いのみシミュレーションされる。

5 性能評価

5.1 評価方法と使用する回路データ

本章では、提案手法を導入した故障診断用テストパターン生成手法を、パターン生成時間の観点から評価する。これらの評価では、実際に本手法をプログラムとして実装し、評価用データセットを入力し、それぞれ実行時間を測定した。

評価プログラムはPythonで記述し、CPUにAMD Ryzen 5 3500U Processor、メインメモリに8GBのRAMを搭載したコンピュータ上で動作させた。回路データの解析には、パーサ生成ツールAntlr4で生成したパーサを使用する。

評価に使用したデータは、テストパターン生成、タイミング解析、テクノロジーマッピングなどのデジタル設計分野の評価で頻繁に使用されている組合せ回路データセットISCAS'85である。ISCAS'85は、ストライクパターン生成、タイミング解析、テクノロジーマッピングのデジタル設計分野で評価に頻繁に使用されるデータセットである。このデータセットには、加算器、乗算器、ALUなどの回路データが含まれている。

このような計算機システムで使用される論理関数の回路データが含まれているため、実用的なシステムを設計するのと同じような感覚で評価することができる。しかし、ISCAS'85 自体は CMOS 集積回路用のデータセットであり、SFQ 回路には対応していない。そこで本研究では、ISCAS'85 の一部の回路データを [6] で提案された回路記述形式に変換し、評価に用いることにした。

(@具体的な変換方法について説明する。RSFQ 回路ではクロックパルスに同期してゲートが動作するため、CMOS 回路で実装された組み合わせ回路の論理的機能を保存しつつ SFQ 回路に変換するには、データパルスが伝搬していくタイミングをあわせる必要がある。タイミングをあわせる方法として、D-flipflop を適切に回路に挿入する方法がある。図 5.9 に変換の例を示す。もとの回路データにおいて回路入力 in2 は直接 G2 に入力されている。SFQ 回路においては in2 の直後に D-flipflop 回路を挿入し、G2 に入力される他のパルスとのクロックサイクルを合わせることで論理的機能を保存する。その他変換の際に、n 入力をもつ論理ゲートを 2 入力のゲートに分解したり、各ゲートに対してコンカレントフロックロッキング形式でクロックを供給する等の処理を施す。)

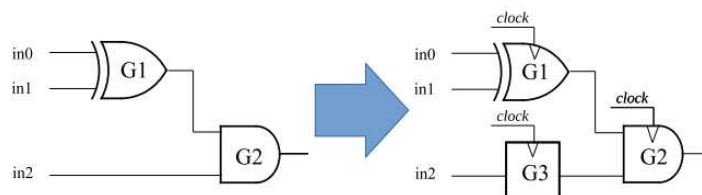


図 5.9: 回路変換の例

両評価に用いた ISCAS'85 の回路データは c17、c432、c499、c880 である。各回路の変換後のゲート数やパイプライン段数等の数値を表 5.1 に示す。

5.2 小さい回路での評価

本節では先行研究のテストパターン生成手法に高速化手法を導入し、パターン生成にかかる時間を測定することで、提案手法が実用性向上に寄与したかを評価する。

表 5.1: 評価に用いた回路の基本的なパラメータ

データラベル	ゲート数	仮定故障数	パイプライン段数
c17	9	30	2
c432	1287	2926	31

表 5.2: c17での測定

Base	Pipeline	Fanout	Both
12.54[sec.]	11.66[sec.]	11.50[sec.]	10.75[sec.]

正常回路と故障回路を論理シミュレーションするのみの単純な方法と提案手法両方をプログラムとして実装し、ISCAS'85の回路データを入力して、実行時間を計測する。

先に、テストパターン生成の評価プログラム上で行った実装上の留意点について簡単に述べる。故障シミュレーションを用いたパターン生成では、繰り返し各仮定故障に対して故障シミュレーションを行うが、その際、前回のイテレーションで診断が可能となった仮定故障の故障シミュレーションは行わない。

テストパターン生成の評価プログラムを終了する診断閾値は1.00とし、ランダム生成されるパターンは各イテレーションで100個というようにパラメータを設定しプログラム実行する。

作成したプログラムにISCAS'85の回路データのc17を入力して各10回実行した平均実行時間を表5.2に示す。表のデータラベルは、改良を加えていない方法(Base)、パイプライン特性を利用する方法(Pipeline)、ファンアウトコーンのみ故障回路をシミュレーションする方法(Fanout)、提案手法の両手法を併用したもの(Both)を表している。

今回、改良を加えていない生成手法、ファンアウトコーンのみを計算する手法、パイプライン動作の特性を利用する高速化手法と併用手法のどの生成においても、故障診断率1を満たす全ての仮定故障を診断できるテストパターンを生成している。用意した回路データの中で最も小さな回路であるc17に対しては、シミュレーション回数が少ないことから初期化等のオーバーヘッドが提案手法による計算量削減を上回ることが

表 5.3: c432 での測定

Base	Both
130623.11[sec.]	4314.72[sec.]

懸念されていたが、結果速度が本手法により 1.16 倍の高速化に成功していることがわかる。また、c17 ではパイプライン段数が少ないことから、Pipeline 手法が Fanout 手法より時間がかかっているが、パイプライン段数が増加することでより高速化の効果がでる事が予想される。

今測定では、イテレーションごとにランダムに生成するパターン数を 100 に設定し計測を行ったが、10、20、とパターン数を減らした際も同様に故障診断率 1 を満たすパターン列が生成できた。

5.3 大きい回路での評価

本節では、大きい回路に対しての評価方法とその結果、考察について述べる。

c432 に対し診断閾値を 0.01 とし、ランダム生成されるパターンは各イテレーションで 10 個というようにパラメータを設定しプログラム実行する。改良を加えていない手法と併用手法での各測定時間を表 5.3 に示す。

c432 に対する提案手法を導入しないテストパターン生成に比べ、併用手法を導入した生成では故障診断率 0.01 を満たすパターンを約 30 倍もの速さで生成可能となった。

c17 での計測結果と比較して、大幅な高速化がなされているのは回路規模の増加やそれに伴うパイプライン段数の増加によって、省くことのできる計算量が、改良前の手法の時の計算量と比較しても格段に増加しているためである。また故障診断率を満たすために繰り返される試行回数も仮定故障が多い分格段に増加している。

c432 に対する併用手法でのパターン生成を、ランダム生成されるパターンは各イテレーションで 10 個、診断閾値を 1 とした際のパターン生成時間は 32237. 秒であった。表 5.3 で示すように、提案手法を導入しないテストパターン生成での故障診断率 0.01 を満たすパターンの生成に 36.28 時間かかっていることから、故障診断率 1 を満たすパターンの生成時間の 8.95 時間は驚異的な高速化であり、またこれは現実的なパターン生成時間であると

いえる。これは故障診断率を 0.01 で設定した上記の測定と比較してもわかるように、併用手法によって大幅な時間短縮に成功している。計算を使いまわす手法であることから、後半になるほど計算が少なくなり高速化の効果が表れやすい。一度のシミュレーションでの計算量は回路の規模が大きくなるにつれ増加し、シミュレーションの回数も増えるため大規模回路では故障診断用テストパターン生成の更なる高速化が期待される。

6 まとめ

本論文では、タイミング故障のテストパターン生成において、処理時間のうち大きな割合を占める故障シミュレーションの高速化手法を故障診断に対し適用した。故障診断におけるタイミング故障のテストパターン生成において、処理時間の大きな割合を占める故障シミュレーションにおいて、故障によって影響を受けるゲートのファンアウトコーンや RSFQ 回路のパイプライン特性に着目し、削減した。

本手法は、故障が導入されるゲートのファンアウトコーンと SFQ 回路のパイプライン特性を考慮して、パターン生成に関わらないゲートのシミュレーションを削減し、不要な計算を省くことで高速化を図った。

本手法の評価では半導体集積回路のテストパターン生成やタイミング解析に用いられているものを変換することで使用した。したがって、更なる正確な評価を取得するには、SFQ 回路用に専用化された回路を用いて設計されたプロセッサ等のデータを用いて評価する必要がある。また、本手法は複数の故障を対象とした故障診断や、クロックフォローデータ方式を用いて設計された回路については対応してない。今後の課題としては、これらの追加評価及び手法の改良が挙げられる。

謝辞

本研究を進めるにあたり、日々温かいご指導をいただきました高木一義先生に深く感謝致します。コロナ禍でのリモート作業環境において、Slack でのコミュニケーションをとって頂き大変助かりました。また、京都大学情報学研究所教授の高木直史先生、中京大学情報工学科講師の鬼頭信貴先生には、国際学会での発表に関して、大変有用なアドバイスをいただきました。誠にありがとうございました。また、同じくテーマ変更の

相談に乗っていただき、研究室での生活等を通してご支援いただきました深澤祐樹先生にも深く感謝申し上げます。

参考文献

- [1] Beloglazov, Anton, Jemal Abawajy, and Rajkumar Buyya. “Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing.” *Future generation computer systems* 28.5 (2012): 755-768.
- [2] K. Takagi, M. Ono, N. Kito, N. Takagi. Test Pattern Generation for Timing Faults in Rapid Single-Flux-Quantum Circuits Proc. 22nd Workshop on Synthesis And System Integration Mixed Information technologies (SASIMI 2019), R3-16, 2019- 10-22.
- [3] Nagasawa, Shuichi, et al. “Nb 9-layer fabrication process for superconducting largescale SFQ circuits and its process evaluation.” *IE-ICE Transactions on Electronics* 97.3 (2014): 132-140.
- [4] Tolpygo, Sergey K., et al. “Advanced fabrication processes for superconducting very large-scale integrated circuits.” *IEEE Transactions on Applied Superconductivity* 26.3 (2016): 1-10.
- [5] S. Nakamura, K. Takagi, N. Kito, N. Takagi, “A Timing Fault Model and an Efficient Timing Fault Simulation Method for Rapid Single-Flux-Quantum Logic Circuits,” *Journal of Physics: Conference Series*, 1975(012026), pp.1-8, 2021.
- [6] Kito, Nobutaka, Kazuyoshi Takagi, and Naofumi Takagi. “Timing-aware description methods and gate-level simulation of single flux quantum logic circuits.” 17th Workshop on Synthesis And System Integration of Mixed Information technologies (SASIMI2010), R3-5. 2012.
- [7] Ulrich, Ernst G., and Thomas Baker. “The concurrent simulation of nearly identical digital networks.” *Papers on Twenty-five years of electronic design automation*. 1988. 318-323.

- [8] Wang, Fangzhou, and Sandeep Gupta. “Timing verification for rapid single-flux-quantum (SFQ) logic: New paradigm and models.” 2019 IEEE International Superconductive Electronics Conference (ISEC). IEEE, 2019.