

修士論文

タイミング制約を緩和するクロッキング方式

A Clocking Scheme with Relaxed Timing
Constraints

指導教員 五島 正裕 准教授

東京大学大学院 情報理工学系研究科
電子情報学専攻

学籍番号 48-076421

樽井 翔

概要

本論文は、従来のクロッキング方式よりタイミング制約の緩いクロッキング方式を提案する。まず、近年深刻化しているプロセス微細化に伴うプロセスバラツキの問題について述べ、バラツキが原因となって起こりうるタイミング・フォールトの説明をする。次に、既存のタイミング・フォールト対策技術とクロッキング方式の特徴や性質について述べる。提案手法では、2相ラッチをベースとすることでタイム・ボローイングの性質を持たせる。また、ラッチ間のパスをショート・パスとクリティカル・パスに分離し、セレクタで出力を制御することで、回路の遅延制約を最大1.5サイクルまで許容し、正しい動作を保証する。その結果、既存の技術より制約が緩くなることを示す。

目次

第1章	はじめに	1
1.1	背景	1
1.2	目的	1
1.3	構成	2
第2章	タイミング・フォールト	3
2.1	設計/製造とタイミング・フォールト	3
2.2	微細化とバラつき	3
2.3	実際に近い遅延に基づく設計/製造手法	5
2.4	DVFS	5
第3章	タイミング・フォールト対策技術	7
3.1	タイミング・フォールト検出	7
3.2	タイミング・フォールト予報	7
3.3	タイミング・フォールト隠蔽	8
第4章	既存のクロッキング方式	10
4.1	単相ラッチ方式	10
4.2	単相フリップ・フロップ方式	12
4.3	2相ラッチ方式	14
第5章	提案手法	16
5.1	提案手法の構成	16
5.1.1	パスの分離	16
5.1.2	クリティカル・パスの活性化	17
5.1.3	ラッチとセレクトの配置	18
5.2	提案手法の動作	19
5.3	タイミング・フォールト耐性	20
5.4	低周波数での動作	20
第6章	提案手法の妥当性	25
6.1	タイミング制約の比較	25

第7章 おわりに	37
7.1 まとめ	37
7.2 今後の課題	37
発表文献	39

目 次

2.1	タイミング・フォールトの例	4
3.1	RAZOR	8
3.2	カナリア・フリップフロップ	9
4.1	D ラッチの回路図	10
4.2	単相ラッチ方式	11
4.3	D-フリップ・フロップの回路図	12
4.4	単相フリップ・フロップ方式	13
4.5	単相フリップ・フロップと2相ラッチ	14
4.6	2相ラッチ	15
5.1	パスの分離	17
5.2	クリティカル・パスの活性化	18
5.3	ラッチとセレクトの配置	19
5.4	ラッチ・セレクトの状態	19
5.5	提案手法の動作	22
5.6	低周波数での動作	23
5.7	クリティカル・パスの後のラッチの enable 信号	23
5.8	低周波数時のクリティカル・パスの後のラッチ	24
6.1	提案手法の動作例	27
6.2	タイミング制約の比較	36

表 目 次

6.1 複数段にまたがったタイミング制約	26
--------------------------------	----

第1章 はじめに

1.1 背景

今日、コンピュータ・システムは社会の隅々にまで浸透しており、その中核となるマイクロプロセッサには非常に高い信頼性が求められている。

マイクロプロセッサの信頼性を脅かす要因のうち、今後深刻となると予想されるものの一つに、タイミング・フォールトがある。タイミング・フォールトは、回路遅延の動的な変化によって、信号のタイミングに齟齬が生じ、設計者の想定外の動作が生じる過渡故障である。

従来、タイミング・フォールトは設計/製造時に対処すべき問題であって、出荷後は熱暴走のようなケースを除き、ほとんど問題とならなかった。しかし、次章で詳しく述べるように、今後はLSIの製造プロセス微細化によって、設計/製造時の対処のみでは著しく非効率になると予想されている。

そこで有効と考えられるのが、タイミング・フォールトを動的に検出/回復するハードウェア手法である。そのような動的なタイミング・フォールト対策技術としては、RAZOR[2, 1, 4]やカナリア・フリップフロップ[8, 9]が既に提案されている。

これらの技術は、オリジナルの回路に予備の回路を並列につなげ、出力を比較をすることでタイミング・フォールトの検出を行っている。

1.2 目的

本研究では、信号を受け入れる時間を長くすることで最大遅延制約を緩和し、遅延バラツキの大きな回路に対しても正常な動作を保證するクロッキング方式を提案する。記憶素子間の回路をショート・パスとクリティカル・パスに分離し、2相ラッチをベースとして出力を制御する。通常は1サイクル以上遅れた信号はタイミング・フォールトとなっていたが、パスを分離することで最大で1.5サイクルの遅延まで許容できるようになった。また、遅延が大きくなっても、2相ラッチ・システムの特徴であるタイム・ボローイングによって遅延を補償することができる。

1.3 構成

本論文の構成は以下の通りである．まず，第2章では，タイミング・フォールト発生の背景やその特徴についてまとめる．第3章で，既存のタイミング・フォールト検出技術について述べ，タイミング・フォールトを動的に検出/回復する手法が今後有利になると考えられる理由について議論する．第4章では，既存のクロッキング方式について述べ，それぞれのタイミング制約や問題点について説明する．第5章では，提案手法であるタイミング制約の緩いクロッキング方式について説明する．第6章で，提案手法の挙動や有効性について述べ，最後に，第7章で，本論文のまとめと今後の課題を述べる．

第2章 タイミング・フォールト

タイミング・フォールトとは、回路の遅延の増減により、設計時の想定外の挙動を起こす過渡故障である。同期回路における典型的なタイミング・フォールトは、遅延の増加によってロジックの出力がクロック期間内に安定せず、前のサイクルの出力がサンプリングされることである。図 2.1 に、そのような例を示す。この例では、クロックの立ち上がり時に bit1' ~ bit5' の値が取り込まれるべきであるが、bit1 と bit4 を生成するロジックの遅延が何らかの原因で増加し、前のサイクルの値が取り込まれている。従来は、このようなタイミング・フォールトは設計時に遅延の最悪条件を見積もることで回避している。

2.1 設計/製造とタイミング・フォールト

従来の設計/製造は、タイミング・フォールトを起こさないために、以下のように進められる：

1. 回路遅延が設計に対して変動する要因は、主に、製造バラつき (Process)、電源電圧 (Voltage)、温度 (Temperature)、の 3 つである。設計時には、それぞれについて許容範囲を設定する。例えば、電源電圧は $1.10\text{V} \pm 5\%$ 、温度は $0^{\circ}\text{C} \sim 85^{\circ}\text{C}$ 、そして、製造バラつきについては、「ロジックは 3σ 、メモリは 6σ 以内」といった具合である。
2. その上で、すべての動作条件 —— 特に最悪条件において、正しく動作するように設計する。
3. それでも、製造バラつきが許容範囲外であるようなチップが製造されることは原理的に避けられない。そのようなチップは出荷検査によって取り除かれる。

したがって、出荷後には、電源電圧と温度が設定された許容範囲内に収まっている限り、タイミング・フォールトは（ほとんど）発生しないことになる。

2.2 微細化とバラつき

しかし近年では、LSI の微細化に伴い、同じチップ内でも回路の性能が大きくバラつくことが問題となりつつある [6]。

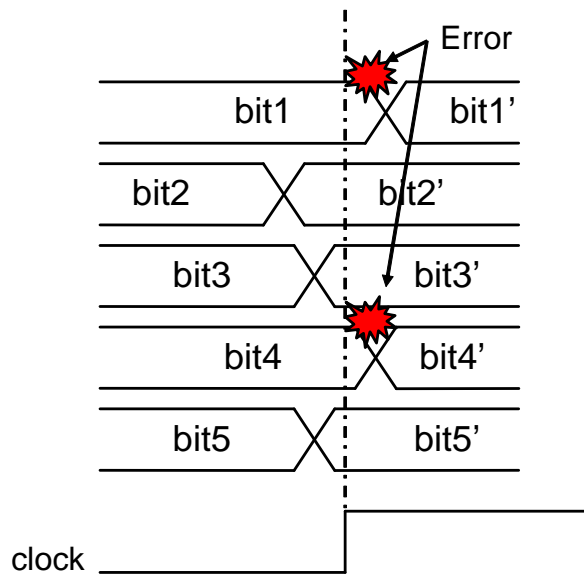


図 2.1: タイミング・フォールトの例

チップ内バラつきとの原因としては，具体的には以下が挙げられる：

製造時のバラつき

1. 露光精度の低下，エッチングや平坦化などの工作精度の低下による各加工寸法のバラつき
2. 不純物密度分布のバラつき

動作時の消費電流に起因するバラつき

1. IR ドロップによって生じる電源電圧のバラつき
2. 温度分布のバラつき

これらの要因により，トランジスタの閾値電圧や ON/OFF 抵抗値，配線の抵抗値がバラつき，回路の遅延と消費電流量がバラつくことになる．また，その結果，動作時の消費電流量のバラつきがさらに増大する負のフィードバックが存在する．

これらは，加工寸法が原子のサイズに近づいたことによるものであるので，悪化傾向を抑えることは原理的に難しい．

2.3 実際に近い遅延に基づく設計/製造手法

バラつきが増大していくと、前述した最悪値に基づく設計/製造手法は悲観的になりすぎることになる。そのため、設計/製造段階で、より実際に近い遅延に基づく手法が提案されている。

設計時の手法としては、SSTA (Statistical Static Timing Analysis)[3] が導入されつつある。パスを構成するトランジスタ、配線の遅延がすべて最悪であるような確率は高くはない。SSTA では、バラつきを統計的に取り扱うことで、悲観的になりすぎない見積もりが可能になる。

また、レイアウト後に、チップ内の温度分布や電源電圧分布を精密にシミュレートする手法なども提案されている。

さらに、製造後に実際の遅延を測定して、クロック・スキューを調整することなども提案されている。この方法では、歩留まりの向上が見込めるほか、見積もりではない、その個体が持つ実際の遅延に基づいて、最高動作周波数、最低電圧での動作が可能になる。

2.4 DVFS

前節の手法は、DVFS (Dynamic Voltage and Frequency Scaling) 技術と併用することによって、大きな効果を発揮する。具体的には以下のようにすればよい：

1. 電源電圧を一定とし、動作周波数を徐々に上げる（あるいは、動作周波数を一定とし、電源電圧を徐々に下げる）。
2. タイミング・フォールトが発生したら、動作周波数（あるいは、電源電圧）を元に戻し、回復を行う。
3. 以降、これを繰り返す。

このようにすることによって、以下のような、2 段階の効果が得られる：

1. 前述した出荷時調整と同様、見積もりではない、その個体の実際の遅延に基づいて、最高動作周波数、最低電圧での動作が可能になる。
2. 実際には、チップ内のクリティカル・パスのすべてが毎サイクル活性化されるわけではない。たとえば加算器のキャリー生成回路では、実際にキャリーが長い距離を伝播しなければ、非常に短い時間で演算を終えることができる

したがって、上述のような手法を用いれば、チップ全体のクリティカル・パスではなく、実際に活性化されたクリティカル・パス——いわば、動

的なクリティカル・パスによって決まる最高動作周波数，最低電圧での動作が可能になるのである．クリティカル・パスが活性化されるかどうかはロジックの入力に依存するので，前述した PVT のバラつきに加えて，入力 (Input) のバラつきにまで対応できると言ってもよい．

このように，タイミング・フォールトを検出/回復する技術は，前述した実際に近い遅延に基づく設計/製造技術の延長線上にあるととらえることができる．

第3章 タイミング・フォールト対策技術

2.4節で述べた DVFS 技術は、それだけでも省電力の効果を得られるが、タイミング・フォールト対策技術と組み合わせることでプロセッサの性能を限界まで引き上げることができる。以下でタイミング・フォールト対策への様々なアプローチを紹介する。

3.1 タイミング・フォールト検出

これら設計/製造時の手法に加えて、動作時にタイミング・フォールトを検出する手法が提案されている。たとえば、Austin らが提案している RAZOR や、佐藤らが提案しているカナリア・フリップフロップ [8, 9] などがある。

RAZOR では、図 3.1 のように遅れたクロックで動作するシャドウ・ラッチを元のラッチと並列に配置し、出力を比較することでタイミング・フォールトを検出する。すると、何らかの原因で信号が遅れてきたときに、元のラッチには誤った値が取り込まれ、シャドウ・ラッチはクロックが遅れているために正しい値が取り込まれる。二つのラッチの出力を比較した結果、値が異なるときにはタイミング・フォールトが発生したと見なし、命令を再実行することで回復を行う [5]。

3.2 タイミング・フォールト予報

カナリア・フリップフロップでは、図 3.2 のように、本来のフリップ・フロップと並列に配置したカナリア・フリップフロップへの入力に遅延を挟み、出力を比較することでタイミング・フォールトを予報する。入力に遅延を入れることで元のフリップ・フロップより先にカナリア・フリップフロップでタイミング・フォールトが発生するため、正常な動作を続けたままタイミング・フォールトの予報が可能となる。しかし、タイミング・フォールトが発生したのではなく発生しそうであることを検出するため、カナリア・フリップフロップがない時と比べて信号が速く到達しなければならなくなり、タイミング制約が厳しくなる。

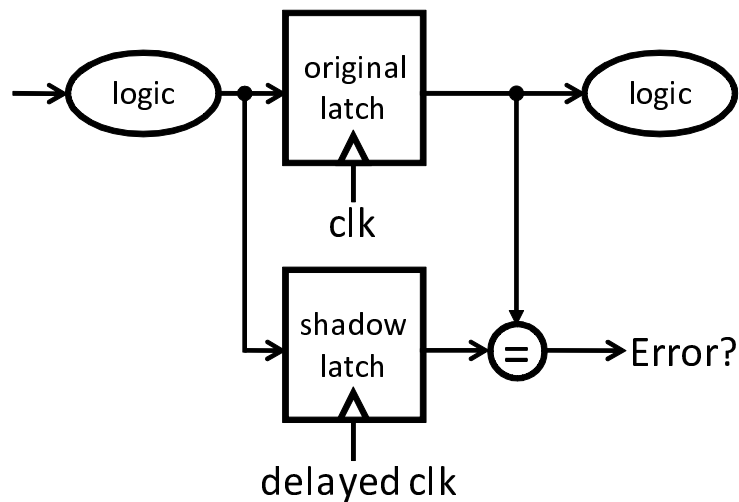


図 3.1: RAZOR

3.3 タイミング・フォールト隠蔽

遅延補償フリップ・フロップとは、単相フリップ・フロップで間に合わなかった信号について、多少の遅延は許容しフリップ・フロップに取り入れるというものである [7]。フリップ・フロップが入力を遮断する瞬間からタイミング・マージンを設け、その間に信号が到着した場合はフリップ・フロップのスイッチを切り替え、入力をフリップ・フロップ内へと受け入れる。このような動作をすることで、タイミング・フォールトとなるはずである信号を救済することができる。また、2相ラッチ・システムのタイム・ボローイングのように、タイミング・マージン内に信号が届いたフリップ・フロップの次の回路でクリティカル・パスを通らなければ、前段の遅延を補償することができる、という利点がある。

しかし、遅延によってタイミング・フォールトとなるはずだった信号と、ショート・パスを通ったことによってタイミング・マージン内に到達してしまった次のフェーズの信号が、同じ時間に同じフリップ・フロップに到達する可能性がある。この二つの信号を区別することはできないため、このような事態を避けるようなタイミング制約を設定しなければならない。

以上のことから、最小遅延制約はタイミング・マージン以上、最大遅延制約は1サイクル+タイミング・マージン以下となる。

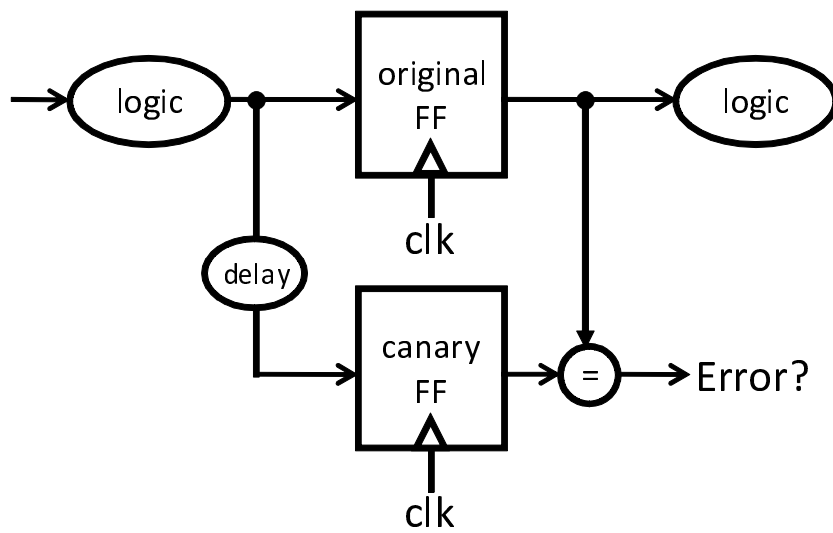


図 3.2: カナリア・フリップフロップ

第4章 既存のクロッキング方式

ラッチとフリップ・フロップのどちらを用いるか、どのようなクロックを分配するかといった、同期式順序回路の構成をクロッキング方式という [10]。クロッキング方式によって、最小サイクル・タイムなど、順序回路の時間的な設計制約の厳しさが決定される。時間的な設計制約とは、回路の遅延の最小値への制約である最小遅延制約と、最大値への制約である最大遅延制約のことを指す。クロッキング方式を考えるに当たって、本来はセットアップ・タイム、ホールド・タイム、クロック・スキューも考慮に入れなければならないが、簡単化のためにここでは考えない。

ラッチとフリップ・フロップの構成、ラッチ/フリップ・フロップのどちらを使うか、またクロックの使い方によってどのように動作が異なるかについて以下で説明する。

4.1 単相ラッチ方式

ラッチをすべて同じ位相で動作させるクロッキング方式を単相ラッチ方式という。ラッチにはDラッチを用いる。このラッチは図4.1のような回路で構成されており、インバータのループで1bitを記憶し、パス・ゲートによって信号の遮断/記憶・出力を行う。

実際にラッチを使って回路を設計した場合の信号の伝わる様子を図4.2に示す。縦軸が時間を表し、横軸が回路中を信号の変化が伝わっていく様子を表し

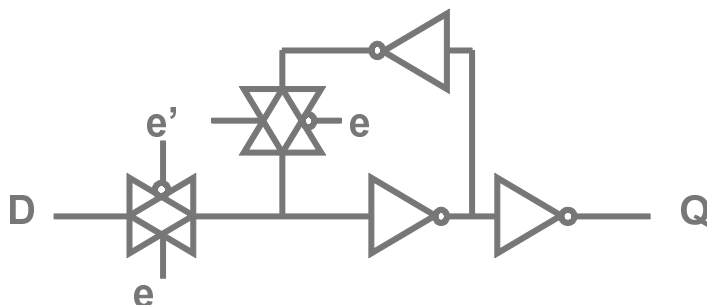


図 4.1: D ラッチの回路図

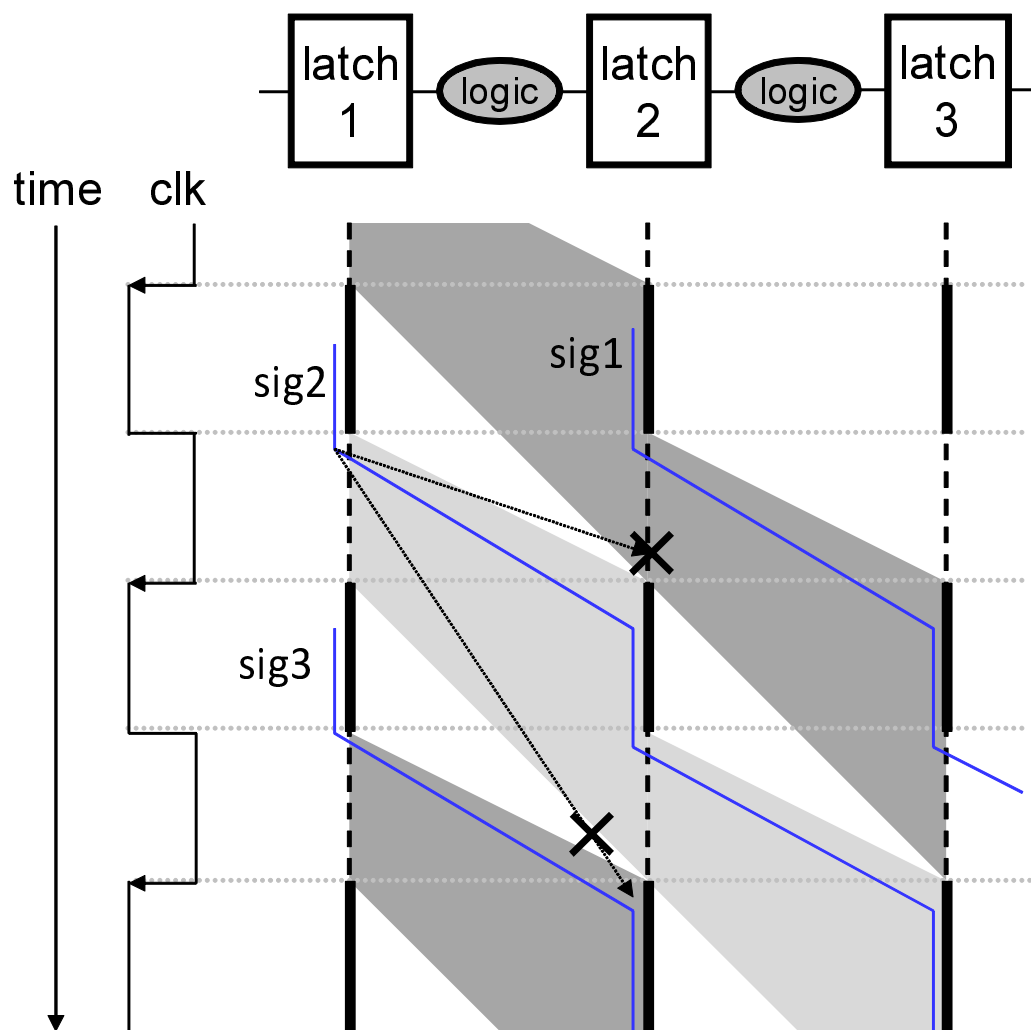


図 4.2: 単相ラッチ方式

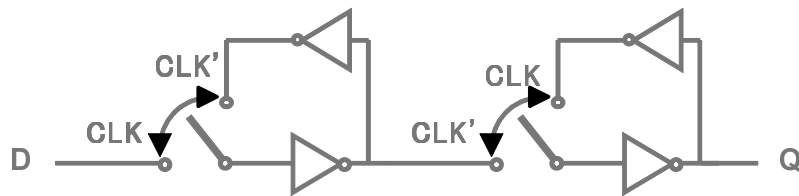


図 4.3: D-フリップ・フロップの回路図

ている。ラッチの下にある太線はラッチが信号を遮断している状態を、点線は信号を透過的に通過させている状態を表している。

図の中に典型的な信号の伝達を示す。ラッチ 1 からラッチ 2 へ信号が伝わる時、前後のクロックで伝わるべき信号と区別するために、ラッチ間の回路に最小遅延制約と最大遅延制約が生じる。sig2 が sig1 と混ざらないためには、ラッチが透過的である時間に到達してはいけないため、0.5 サイクル以上かからなければならないという最小遅延制約を満たさなければならない。sig3 と混ざらないためには、ラッチが遮断される前に到達する必要がある、クリティカル・パスを通過する時間が 1.5 サイクル以内でなければならないという最大遅延制約を満たす必要がある。ここで言っている「信号が混ざる」とは、あるサイクルで伝わっている信号 (sig2) が早く/遅く到着するために、前/後のサイクルで伝わっている信号 (sig1/sig3) に影響を与えることを指す。

4.2 単相フリップ・フロップ方式

フリップ・フロップをすべて同じ位相で動作させるクロッキング方式を単相フリップ・フロップ・システムという。フリップ・フロップは、逆の位相で動作する D ラッチを二つ直列につないだ D-フリップ・フロップを用い、回路は図 4.3 のようになる。

D ラッチの二つのパス・ゲートがスイッチの役目を果たしているため、図 4.3 の回路図ではスイッチとしている。D ラッチと同様に 1bit 記憶する機能はあるが、逆相のラッチを直列につなぐことでどちらかのラッチが常に遮断をしているため、透過性がなくなっている。

フリップ・フロップを用いた場合の信号の様子は図 4.4 のようになり、フリップ・フロップの下の点線の左の太線は一つ目のラッチ (マスタ・ラッチ) の遮断、右の太線は二つ目のラッチ (スレーブ・ラッチ) の遮断を表している。透過性がなくなったことで、sig2 が速く到達しても sig1 と混ざることはないため、最小遅延制約はない。最大遅延制約は単相ラッチと同様の考え方で、次のサイクルの信号と混ざらないために 1 サイクル以内となる。

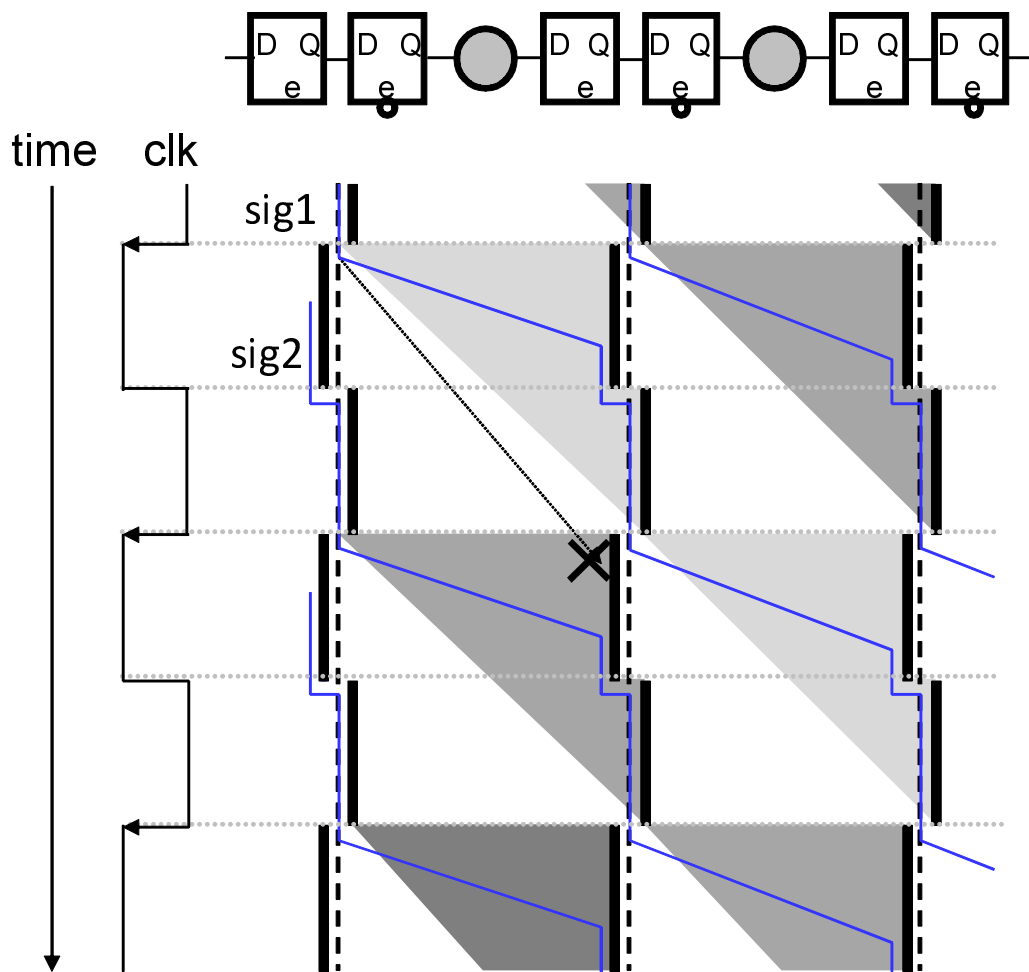


図 4.4: 単相フリップ・フロップ方式

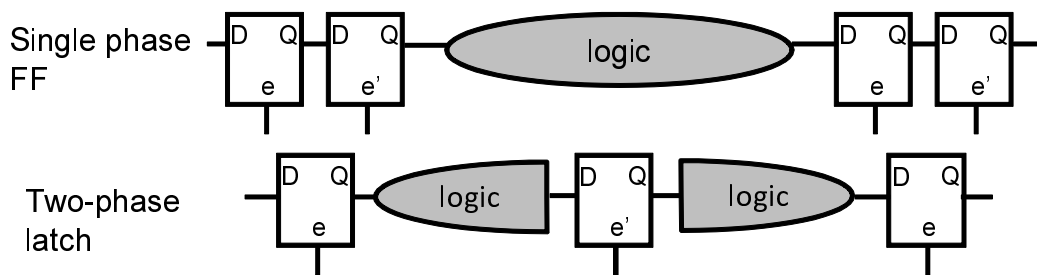


図 4.5: 単相フリップ・フロップと2相ラッチ

4.3 2相ラッチ方式

逆の位相で動作するラッチを交互に配置したクロッキング方式を2相ラッチ・システムという。単相フリップ・フロップも逆相のラッチを直列につないだものであるが、2相ラッチはこの二つのラッチを分離し、フリップ・フロップ間の回路の中間にラッチを配置したものと同一のものである。そのため、図4.5で示すようにラッチ間の回路の長さは単相フリップ・フロップの場合の約半分になる。

2相ラッチの動作の様子を表したものが図4.6である。ラッチ一つ一つは透過性があるが、逆相のラッチを交互に配置することで、sig2がどれだけ早く到着してもsig1と混ざることはないため、最小遅延に対する制約はない。最大遅延制約はsig2とsig3が混ざらないよう1サイクル以内となり、単相フリップ・フロップと同じであるが、フリップ・フロップ間の回路よりラッチ間の回路の方が短いことを考えると、2相ラッチの方が制約が緩いといえる。

また、2相ラッチの特徴にタイム・ボローイングというものがある。最大遅延制約は1サイクルであると述べたが、各ラッチは0.5サイクル毎に開閉するため、基本的には0.5サイクル程度で信号が到達しなければならない。しかし、図4.6の丸で囲った部分のように、0.5サイクル以上かかって到達した場合でも、次の回路で短いパスの結果が採用されれば、何も問題なく動作する。短いパスの結果が採用されるとは、長いパスが活性化されずに短いパスの結果によって最終的な出力が決定される場合を指す。このように前後のラッチで時間の貸し借りをできるために、最大で1サイクルの遅延が許されることになる。また、単相フリップ・フロップ・システムではフリップ・フロップ間の回路ごとに遅延制約を満たす必要があるが、2相ラッチ・システムでは0.5サイクルを超えた遅延は蓄積され、合計の遅延が1サイクルを超えなければよい、という性質がある。

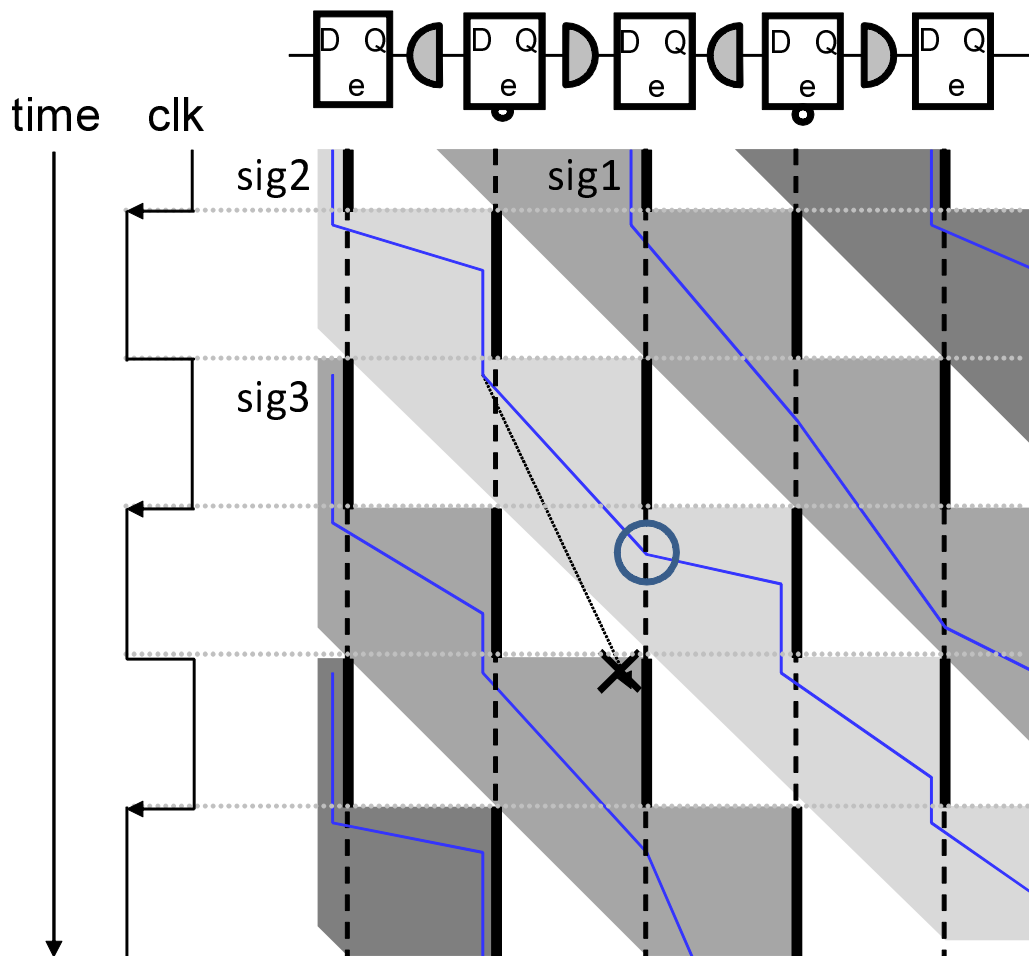


図 4.6: 2相ラッチ

第5章 提案手法

第4章で述べたように，既存のクロッキング方式にはそれぞれ長所，短所がある．そこで，本研究では遅延補償フリップ・フロップのようにタイミング・フォールト耐性を持ち，かつ既存のクロッキング方式より時間を有効活用できるクロッキング方式を提案する．

5.1 提案手法の構成

ラッチ間の回路をショート・パスとクリティカル・パスに分離し，セレクタで次の回路への出力を制御することで，タイミング制約の緩和・正しい動作の保証をする．本節ではパスの分離とその効果・原理，ラッチとセレクタの配置について述べる．

5.1.1 パスの分離

ラッチ間の回路をショート・パスとクリティカル・パスに分離し，ショート・パスのみの結果，ショート・パスとクリティカル・パスの合流後の結果を別々に扱えるようにする．ラッチ間の回路は，あるラッチの出力から分岐し様々な処理を経て最終的には一つの線に合流し，次のラッチの入力となる．この最初の分岐から最後の合流までの回路を，通過にかかる時間の閾値を0.5サイクルとして分離する．パスを分離することで，クロックに間に合わない信号（クリティカル・パスを通過してきた信号）と早く到達した信号（ショート・パスを通過してきた信号）を区別することができるため，前者を通過させ後者を遮断することが可能になる．クリティカル・パスを通過してきた信号については遮断をする必要がなくなるため，ショート・パスにのみラッチを挿入する（図5.1）．また，回路によっては分離することが困難なものがあるが，この場合は分離せずに通常の2相ラッチと同じように扱うことで，既存のクロッキング方式に劣ることはない．

上述したパスの分離条件から，ショート・パスのタイミング制約は0~0.5サイクルとなる．クリティカル・パスのタイミング制約は，最小遅延制約が0.5サイクル以上，最大遅延制約が最小遅延制約 + 1サイクル = 1.5サイクルとなる．遅延の幅が1サイクルを超えると，あるサイクルの最大遅延制約ギリギリ

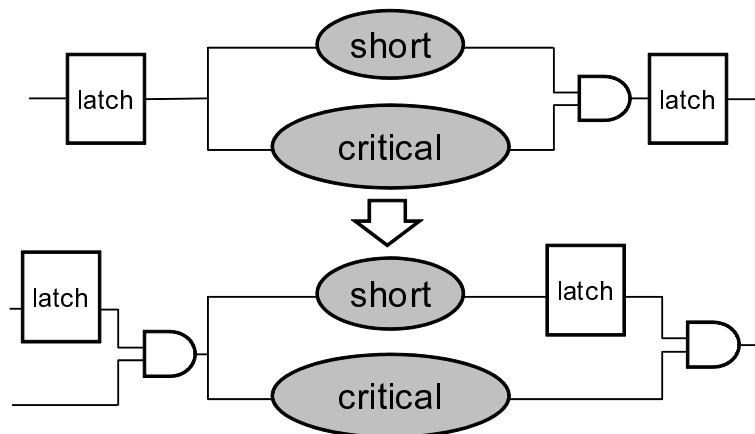


図 5.1: パスの分離

の信号と次のサイクルの最小遅延制約ギリギリの信号が混ざる可能性があるため、最大遅延制約は最小遅延制約 + 1 サイクル以内でなければならない。

5.1.2 クリティカル・パスの活性化

ショート・パスとクリティカル・パスが分岐合流したときに、最終的な出力にクリティカル・パスの結果が必要な場合と必要でない場合があり、必要な場合を「クリティカル・パスが活性化される」という。たとえばショート・パスとクリティカル・パスの合流が AND ゲートであった場合について考える (図 5.2 初期状態)。あるフェーズのショート・パスの結果が 0 であるときには、クリティカル・パスの結果にかかわらず合流後の結果も常に 0 であるため、クリティカル・パスの結果は必要なくクリティカル・パスは活性化されていない (図 5.2(1))。あるフェーズのショート・パスの結果が 1 であるときには、クリティカル・パスの結果が到着しないと最終的な合流結果がわからないため、このような場合をクリティカル・パスが活性化されているという (図 5.2(2))。AND ゲート以外の合流素子においても、同様の考え方でクリティカル・パスが活性化されない場合を導くことができる。

このクリティカル・パスの活性化の考えを応用することで、より効率的な信号の伝達を行うことができる。再度ショート・パスとクリティカル・パスの合流が AND ゲートであった場合について考える。あるフェーズのショート・パス、クリティカル・パスの結果が (0, 0) であるとする。クロックが切り替わると、次のフェーズの信号が到着次第上書きされていくが、新しく届いたショート・パスの結果が 1 であってもクリティカル・パスの結果には 0 が残っているため、合流後の結果は 0 のままとなる。この場合にショート・パスの結果が反映されていないため、信号が早く到着したことによるメリットが失われている。

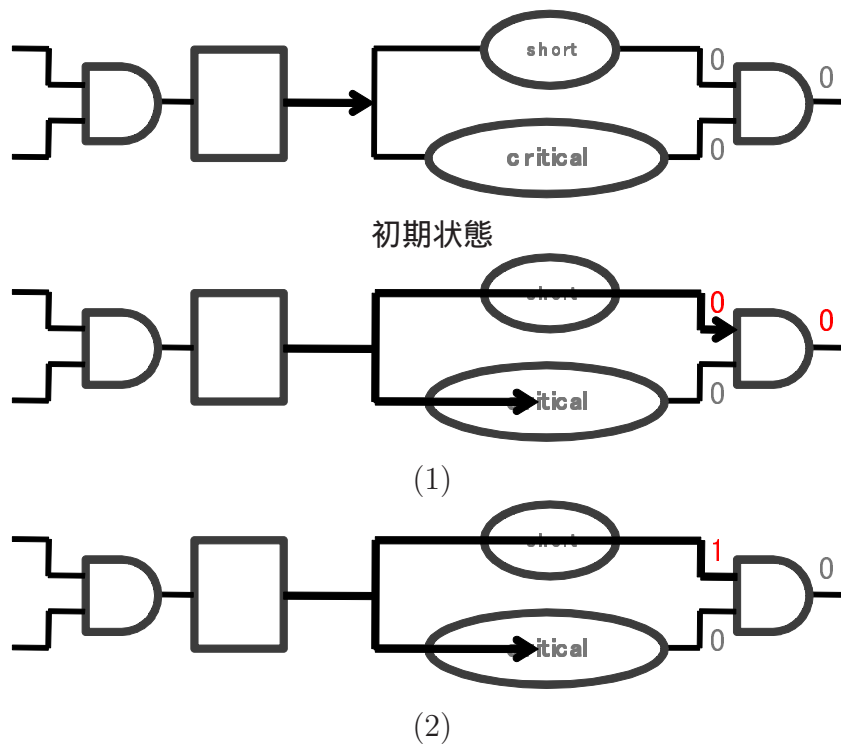


図 5.2: クリティカル・パスの活性化

このような場合には2相ラッチの性質である「早く到着した信号で、前の回路の遅延を補償」することができないため、ショート・パスの結果をクリティカル・パスの結果にかかわらず先に伝えることに意味があると言える。クリティカル・パスの信号が到着した結果、合流後の信号に変化がなければ、先に伝えたショート・パスの結果をそのまま使うことが出来、変化があった場合には合流後の結果で上書きしていくことになる。また、クリティカル・パスの結果が制約より遅かった場合には、3章で述べた検出・回復手法を用いることで対応できる。もちろん、合流がANDゲートでなくても同様の考え方でショート・パスの結果を先に伝えることには意味があると言える。

5.1.3 ラッチとセレクタの配置

ショート・パスの結果と、ショート・パスとクリティカル・パスの合流後の結果を扱うための回路構成は図 5.3 のようになる。ショート・パスの後ろにあるラッチは、通常の2相ラッチと同様の動作をする。クリティカル・パスの後ろにあるラッチは通常時は透過的で、5.4 節で述べる低周波数動作時やストール時に初めて信号を遮断する。そして、ショート・パスの結果と合流後の結果を2入力のセレクタに通す。このセレクタはクロックで制御し、ショート・パ

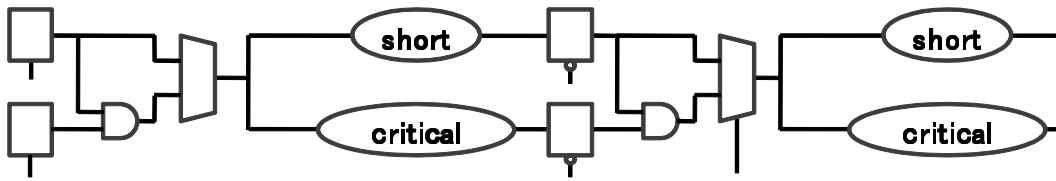


図 5.3: ラッチとセレクタの配置

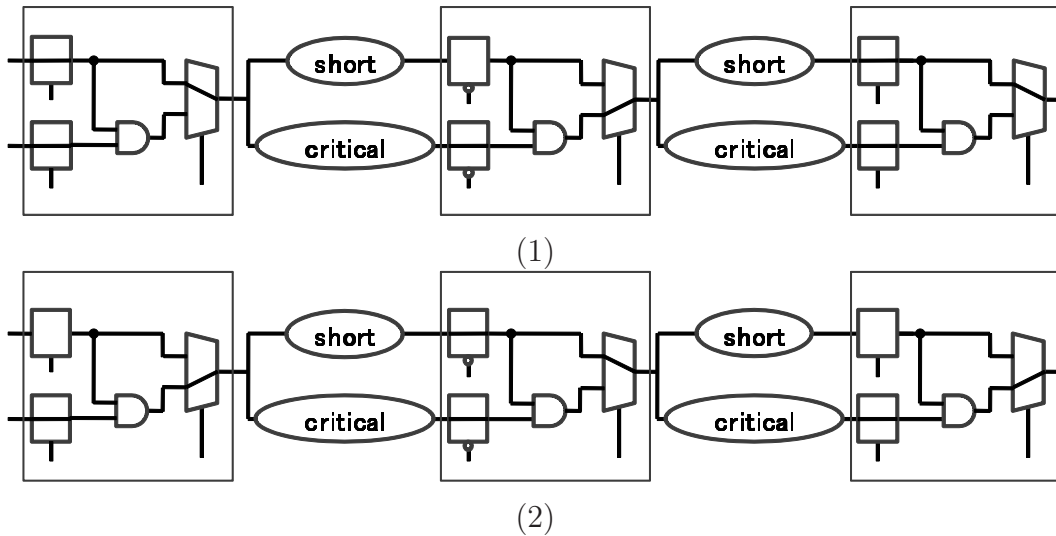


図 5.4: ラッチ・セレクタの状態

スのラッチが透過的である時間にショート・パスの結果を選び、遮断している時間に合流後の結果を選ぶ。

5.2 提案手法の動作

ショート・パスの後のラッチとその後のセレクタをクロックで制御するため、ラッチの開閉およびセレクタの状態を図 5.4 のように表すことができる。図 5.4(1) では、一つめのショート・パスの後のラッチが透過的でセレクタもショート・パスを選んでいる。二つめのラッチは遮断していて、セレクタは合流後の結果を選んでいる。図 5.4(2) では、一つめのショート・パスの後のラッチが遮断していてセレクタは合流後の結果を選んでいる。二つめのラッチは透過的でセレクタもショート・パスを選んでいる。クリティカル・パスの後のラッチは、図 5.4(1), (2) とともに常に信号が素通りしている。また、大きく四角く囲んだ部分が、通常の 2 相ラッチにおける合流部分とラッチに相当する。

図 5.4 において信号が伝わっていく様子を表したものが図 5.5 である。図中の (1), (1)' は図 5.4 の (1) に対応, (2) と (2)' は図 5.4(2) に対応している。sig2

を基準に考え、sig1 / sig2 はそれぞれ前のフェーズ / 後のフェーズで伝わる信号を表す。sig2 の赤い線はショート・パスを伝わる信号、青い線はクリティカル・パスを伝わる信号を表す。合流後の結果は、クリティカル・パスの結果が到着して初めて確定するため、以後この二つを同じものとして扱う。latch1 ~ 3 は図 5.4 で大きな四角で囲んだラッチ、合流、セレクトアを含んでいるとする。セレクトアで選択されていない側の信号 (図 5.4(1) 一つ目のセレクトアの合流後の信号、(2) 一つ目のセレクトアのショート・パスの信号… etc) は、信号が遮断されていると見ることもできるため、図 5.5 中ではラッチの下の太線で表現している。赤い太線はショート・パスの結果が遮断される時間、青い太線は合流後の結果が遮断される時間を示す。

図 5.5 のように latch1 から sig2 がショート・パスとクリティカル・パスを伝わっていくため、ショート・パスは遅延 0.5 サイクル以下というパスの分離条件から、必ず latch2 で遮断している時間に到達し、クリティカル・パスは遅延 0.5 サイクル以上 1.5 サイクル以下という条件から、(1) の間に latch2 に到着することはなく、(2) か (1)' の間に到着する。(2) の間は latch2 では合流後の結果が選択されていないために遮断され、(1)' になったときに latch3 へと伝わる。latch2 において (1) の時間は sig1 のクリティカル・パスの結果と sig2 のショート・パスの結果が到着する可能性があるが、パスの分離によってショート・パスの結果とクリティカル・パスの結果を別々に扱えるため、遅延補償フリップ・フロップのように前後のフェーズが混ざる、という問題は解決できる。そして、sig1 のクリティカル・パスの結果を latch3 へ伝え、sig2 のショート・パスの結果を遮断することができる。同様に (1)' の間は、latch2 に sig2 のクリティカル・パスの結果と sig3 のショート・パスの結果を混同することなく制御できる。

5.3 タイミング・フォールト耐性

提案手法では、パスを分離しショート・パスの結果を先に伝えることでタイミング・フォールト耐性を実現できている。パスを分離しショート・パスの結果を先に伝えられるため、

5.4 低周波数での動作

通常の周波数では正常に動作していたにも関わらず、DVFS によって周波数を下げたときに、信号が到達する時間が想定している時間より早くなることもある。低周波数動作時も、信号の伝達速度は変わらないため、クリティカル・パスを通過するのにかかる時間が最小遅延制約の 0.5 サイクル以下になることがある (図 5.6)。ショート・パスの結果はラッチで遮断する時間もクロックに

合わせて長くなるため問題ないが、クリティカル・パスの結果はラッチを素通りするため前のフェーズの信号と混ざる可能性がある。そのため、クリティカル・パスの後のラッチを図 5.7 のような回路によって制御をすることで同期させる。図中の delay は基本動作周波数の 0.5 サイクル分とすると、通常動作時はラッチが透過的な状態で、低周波数動作時は図 5.8 のように遮断する時間ができる。これによってクリティカル・パスを通過してきた信号も最小遅延制約の 0.5 サイクルまでは遅らせることができる。また、ストール時にはこのラッチを遮断することで対応可能である。

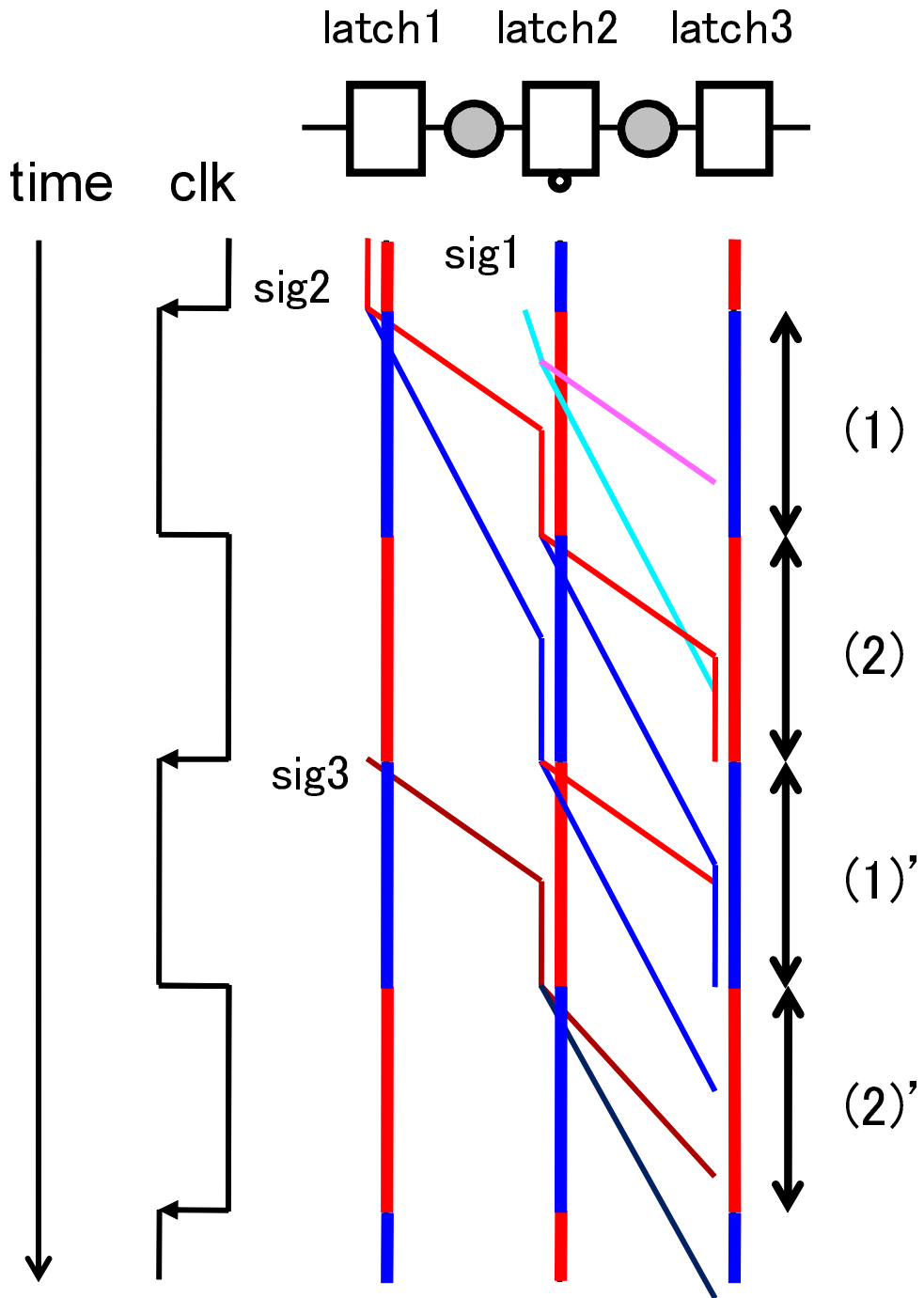


図 5.5: 提案手法の動作

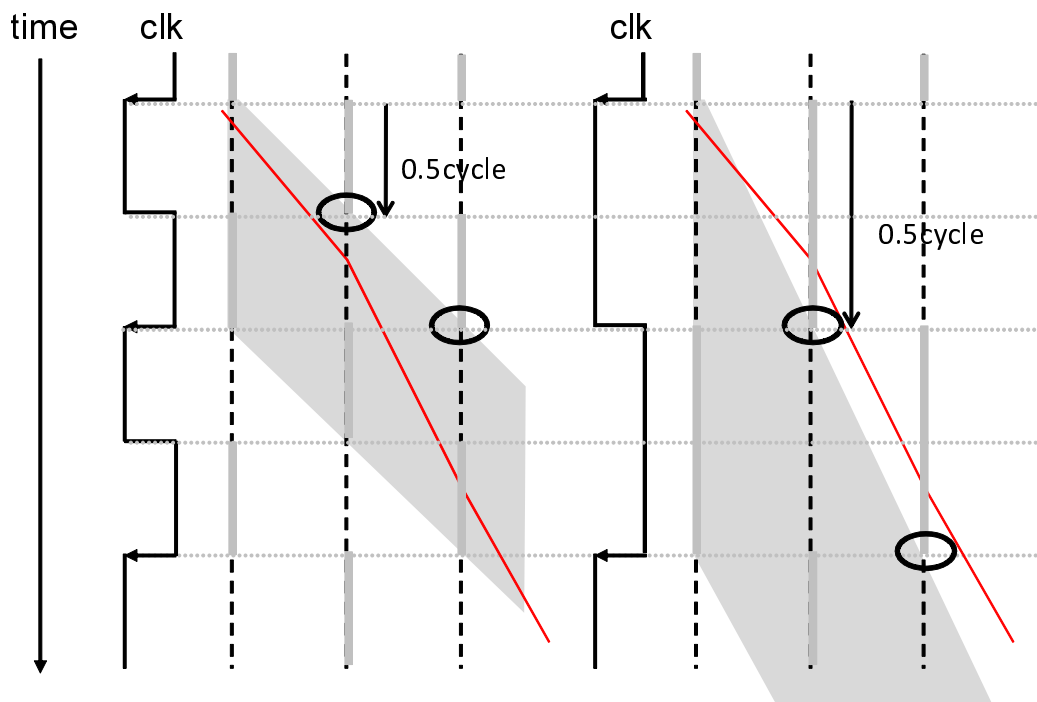


図 5.6: 低周波数での動作

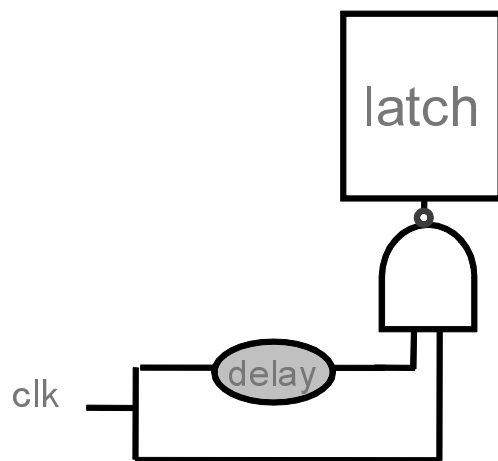


図 5.7: クリティカル・パスの後のラッチの enable 信号

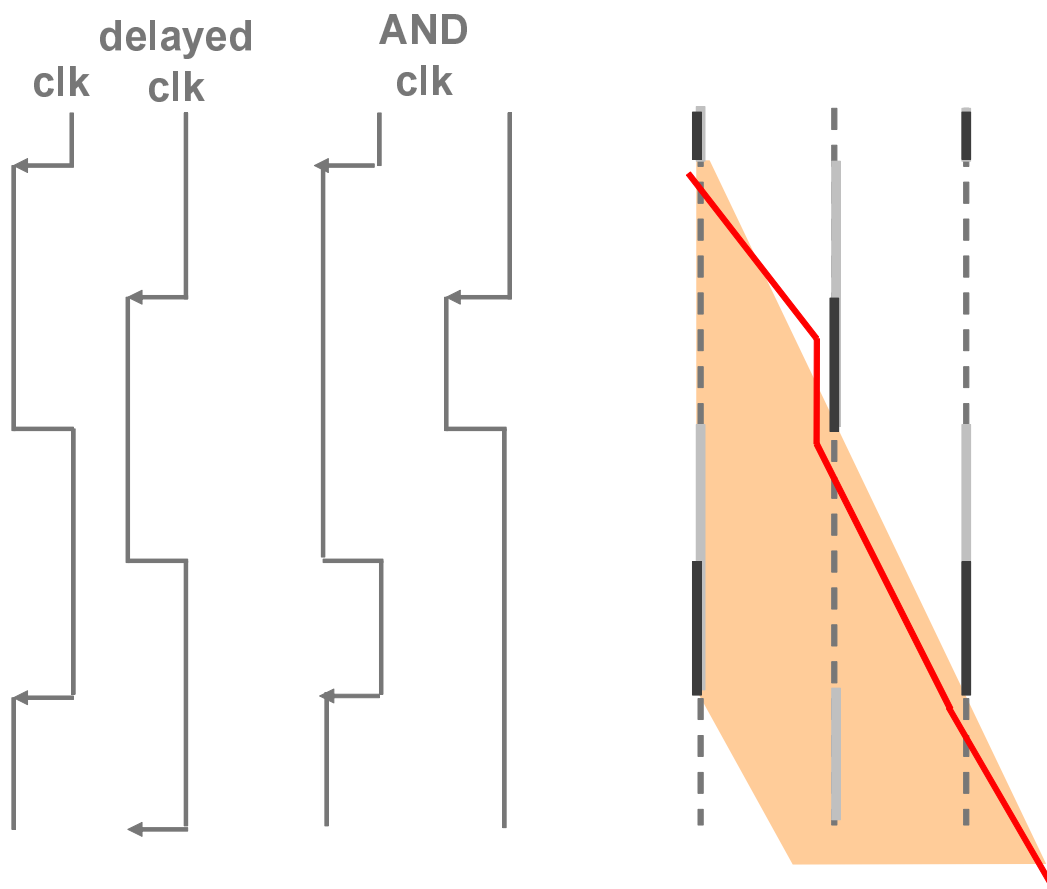


図 5.8: 低周波数時のクリティカル・パスの後のラッチ

第6章 提案手法の妥当性

5.2節では，提案手法における信号伝達の1例を説明した．以下，可能性のある全ての信号伝達のパターンについて動作を考える．パターンの場合分けは，あるラッチを信号が出発する時間，次のラッチに信号が到達する時間の組み合わせによって行った．あるラッチを信号が出発する時間とは，あるラッチに信号が到達した時間と同義であり，0.5サイクルごとに挙動が変わるため0.5サイクルごとに区切って考える．次のラッチに信号が到達する時間とは，あるラッチの次のラッチにショート・パス，クリティカル・パスの結果がいつ到達するかを表しており，これも0.5サイクルごとに挙動が異なるためそのように区切って考える．本章において，以降あるラッチはlatch1，次のラッチはlatch2，遅延は(1)と(2)の切り替わりが基準であるとする．

図6(a)～(i)が全ての場合分けを表したものである．赤い信号がショート・パス，青い信号がクリティカル・パスを通過している信号である．前のフェーズの信号は，現在考えている信号と重なりうる，クリティカル・パスを通過してきた信号が(2)の間にlatch2に到着する場合を考えればよい．同様に後のフェーズの信号は，ショート・パスを通過してきた信号が(2)'の間にlatch2に到着する場合を考えればよい．

全ての図において，前後のフェーズで信号が混ざっていないこと，先に伝えるべき信号を正しく選択していることがわかるため，タイミング制約を守りさえすれば，ラッチとセレクトによって正しく信号を伝達できていることがわかる．

6.1 タイミング制約の比較

本節で，既存のクロッキング方式およびタイミング・フォールト対策技術，提案手法のタイミング制約を比較する．図6.2が，クロックの赤い矢印の位置を遅延0サイクルとしたときの，各手法のタイミング制約の幅を矢印で表したものである．2相ラッチと提案手法は，ラッチの配置の間隔がその他のクロッキング手法の1/2であるため，各ラッチのタイミング制約を矢印で表している．RAZORとCANARYフリップ・フロップは基準となるクロッキング方式を単相フリップ・フロップ方式とした．この図から，既存のクロッキング方式は制約の幅が1サイクルで，提案手法だけ1.5サイクルの幅を持っていることがわ

表 6.1: 複数段にまたがったタイミング制約

	最小遅延制約 (1 段)	最大遅延制約 (1 段)	最小遅延制約 (N 段)	最大遅延制約 (N 段)
単相ラッチ	0.5	1.5	N-1.5	N+0.5
単相 FF	0	1	N-1	N
遅延補償 FF	TM	1+TM	N-1+TM	N+TM
2 相ラッチ	0, 0.5	1, 1.5	N-1, N-0.5	N, N+0.5
RAZOR	0(+delay)	1(+delay)	N-1(+delay)	N(+delay)
CANARY	0	1(-delay)	N-1	N(-delay)
提案手法	0, 0.5	1.5, 2	N-1, N-0.5	N+0.5, N+2

かる．これはパスを分離したことによって得られた効果であると考えられる．また，RAZOR と CANARY フリップ・フロップはそれぞれタイミング・フォールト検出・予報するための時間帯を設けているために，元の単相フリップ・フロップよりタイミング制約が厳しくなっている．

ラッチ間の回路の局所的なタイミング制約はこれまで述べてきたとおりだが，複数段ラッチをまとめてみたときに提案手法だけ性質が異なる．複数段にまたがったタイミング制約を表にまとめたものが表 6.1 である．提案手法以外は，複数段にまたがっても遅延制約の幅は 1 サイクルのままである．これは，パスの分離とクリティカル・パスの活性化の考えによって，提案手法では遅延が最悪である場合ではなく典型的な場合を想定した設計をできるためである．

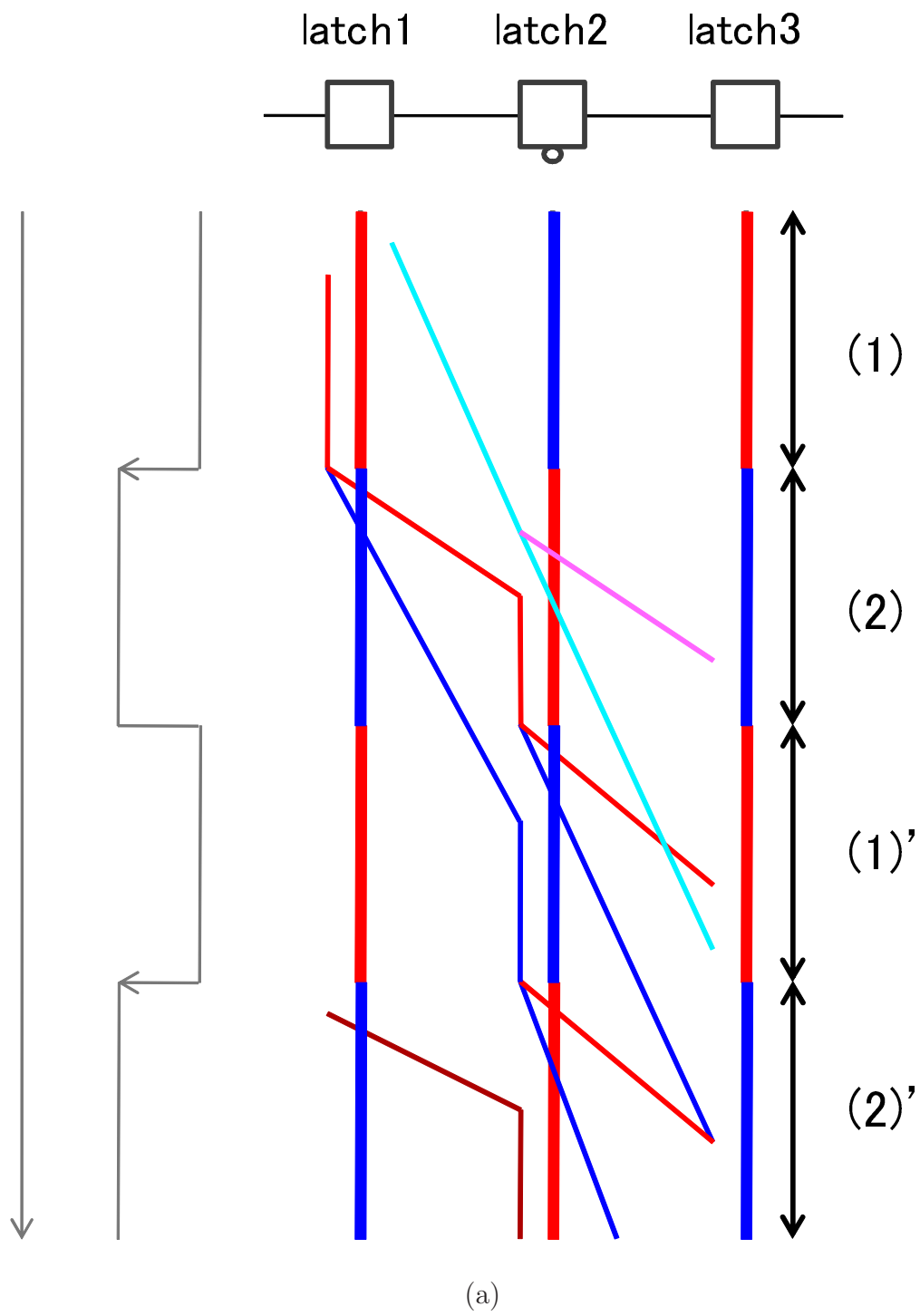
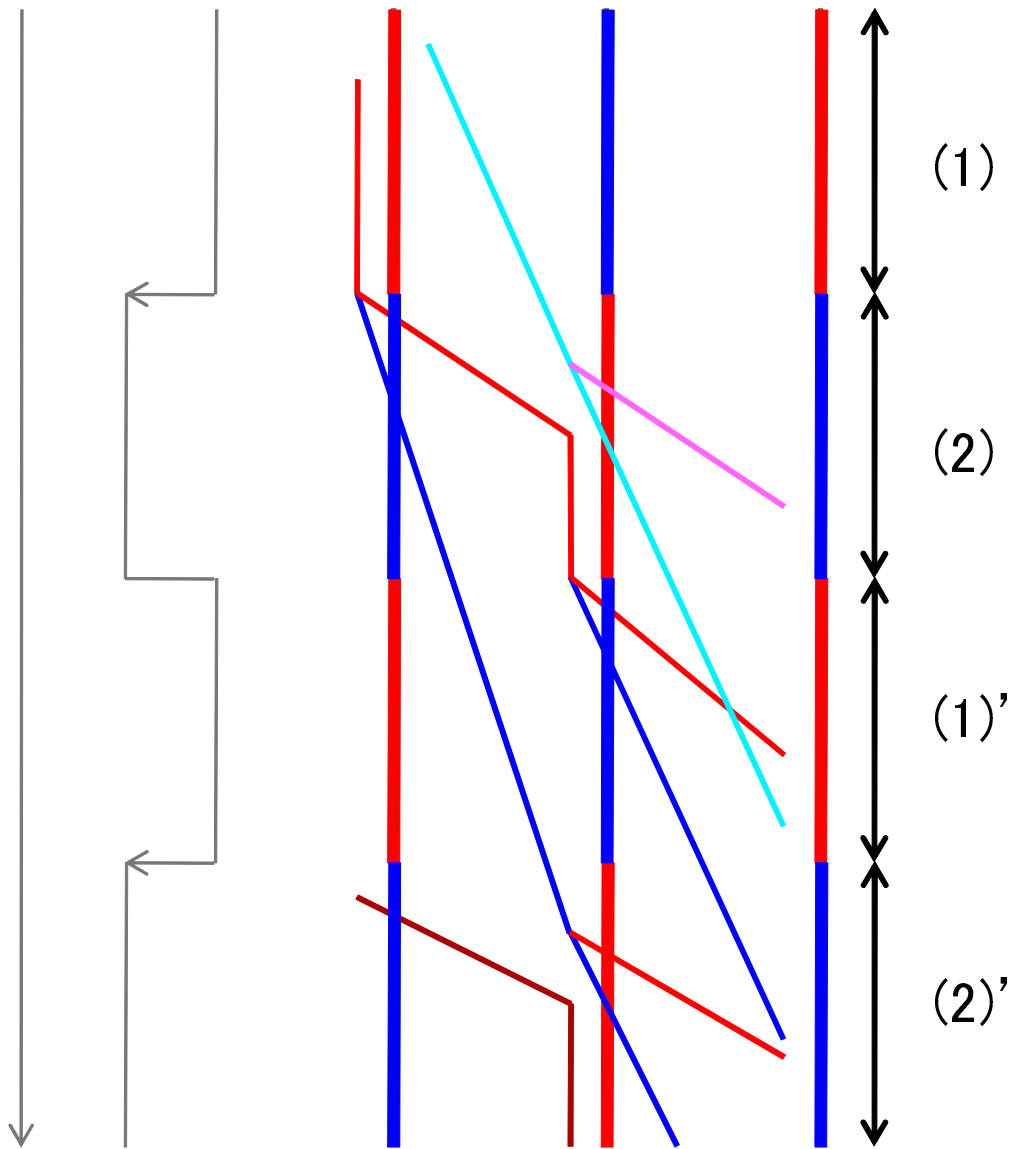
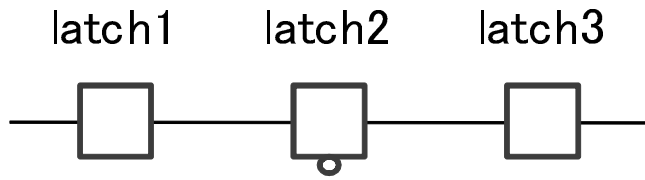
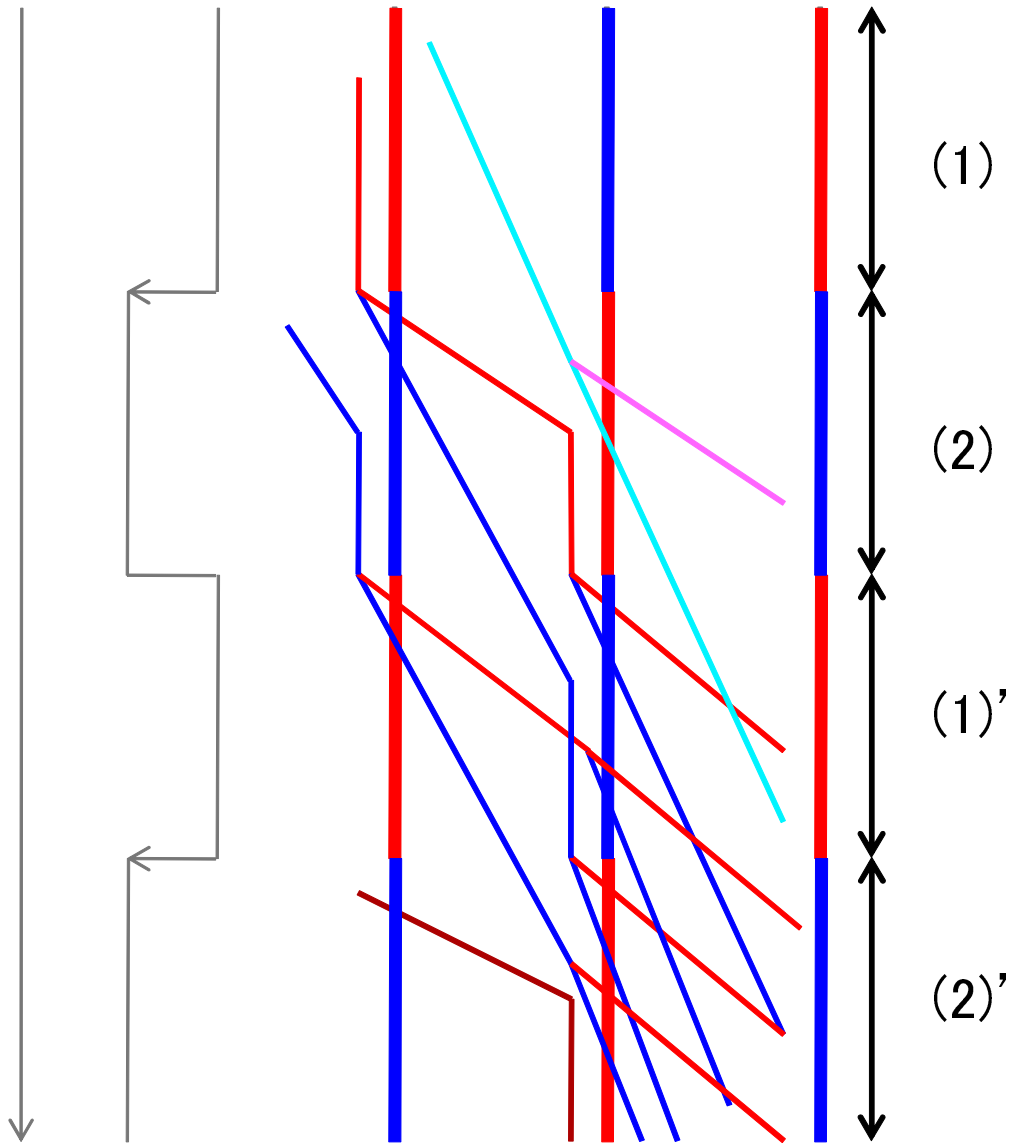
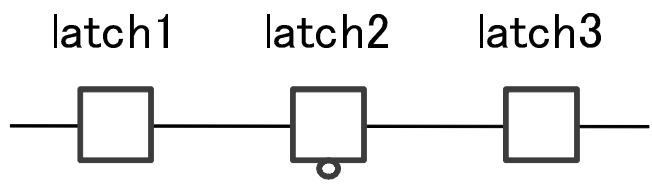


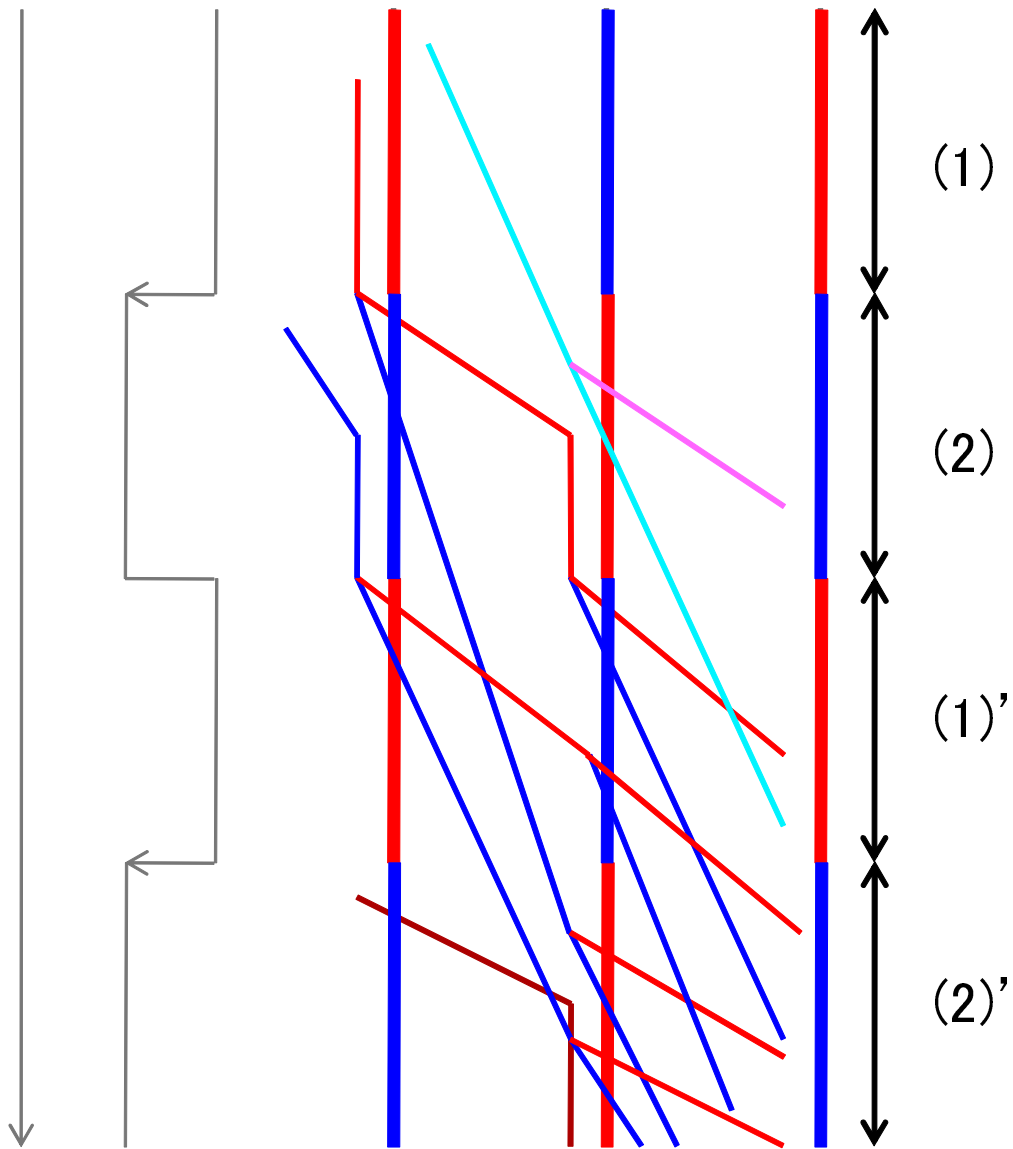
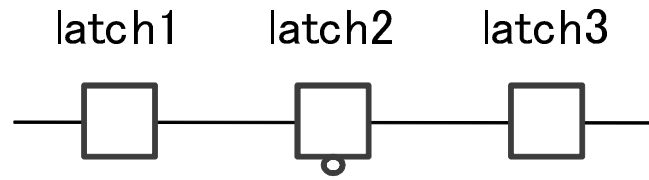
図 6.1: 提案手法の動作例



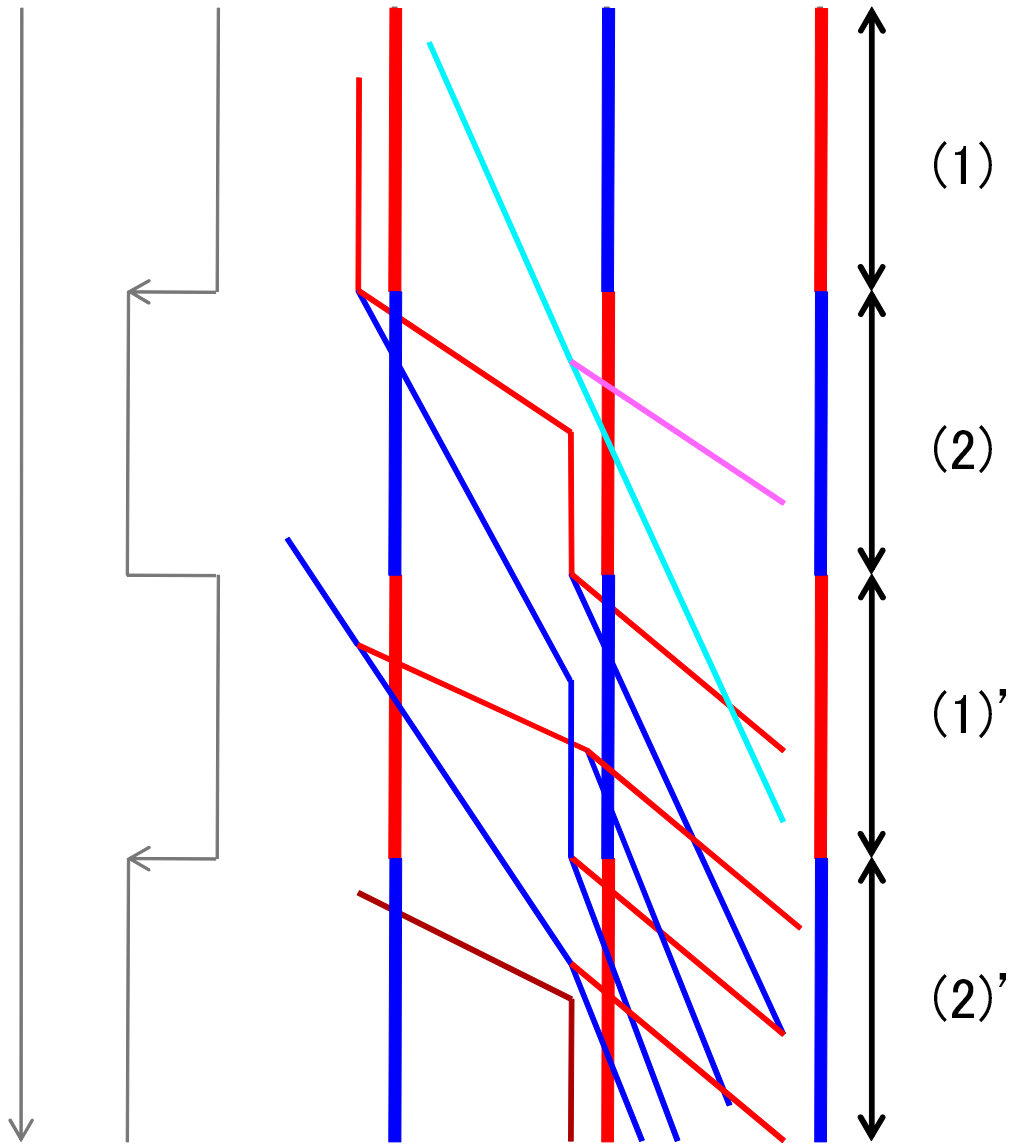
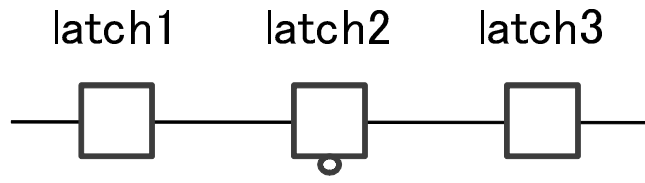
(b)



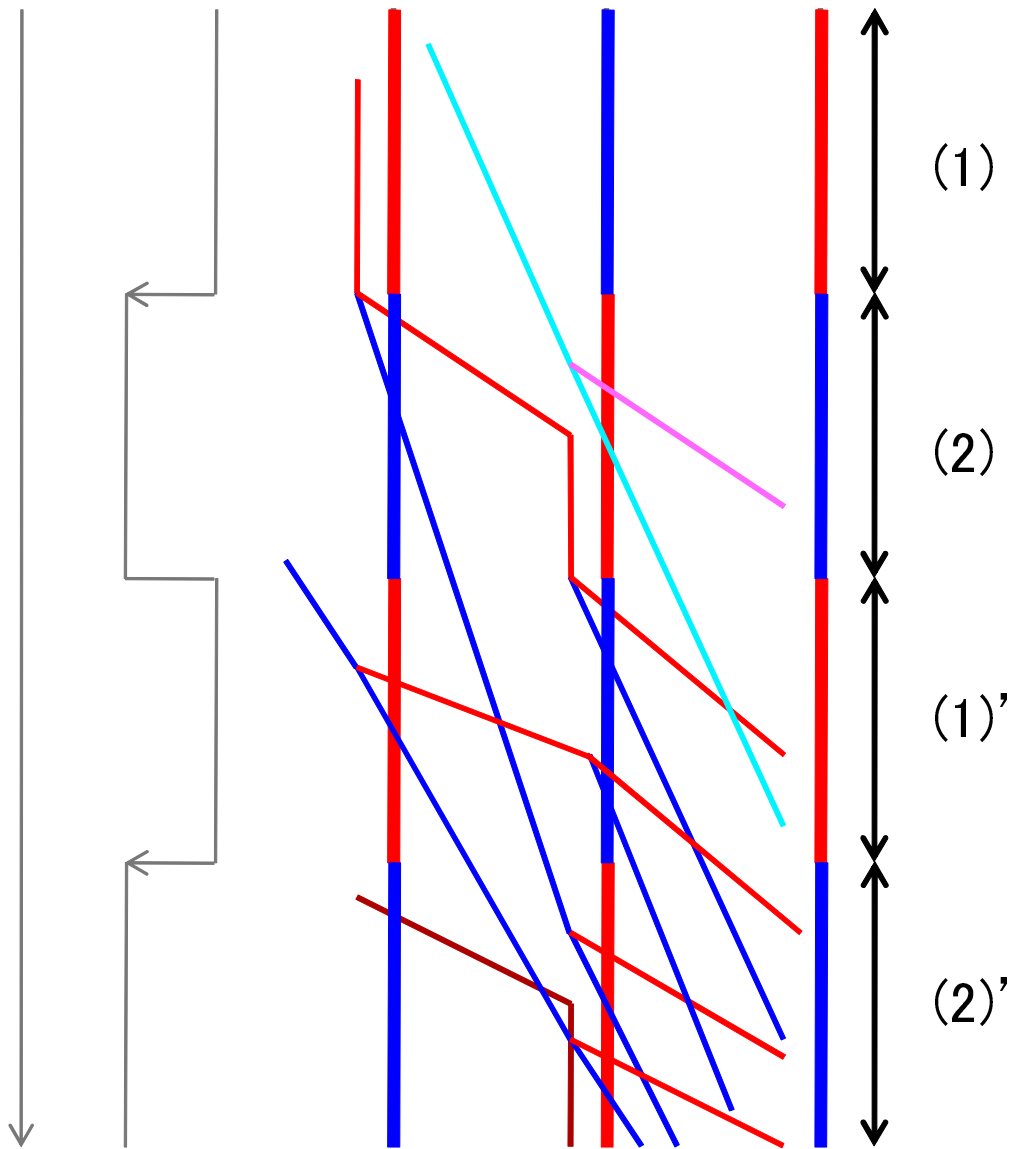
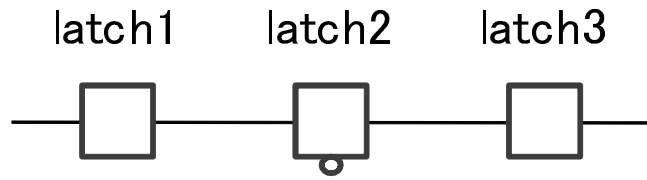
(c)



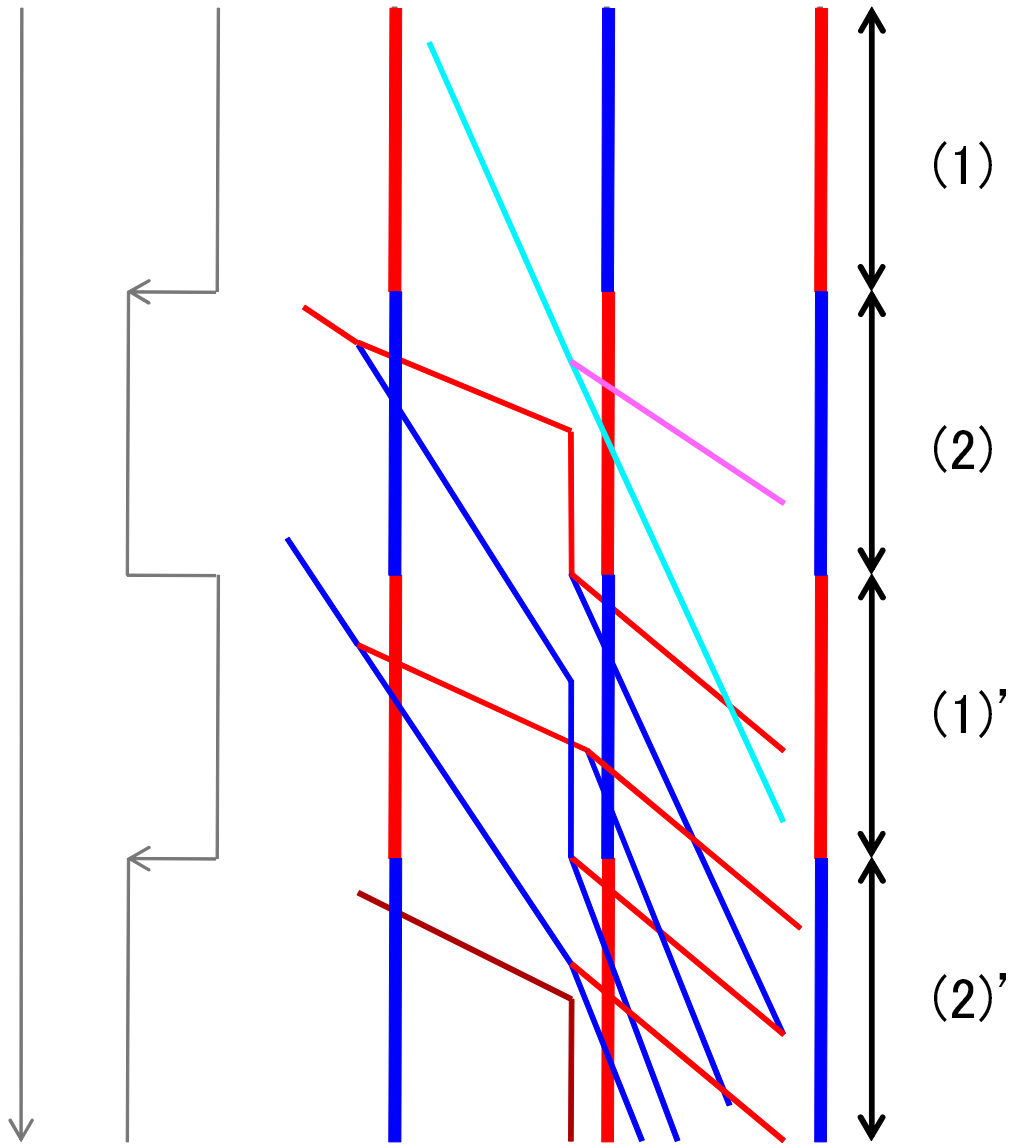
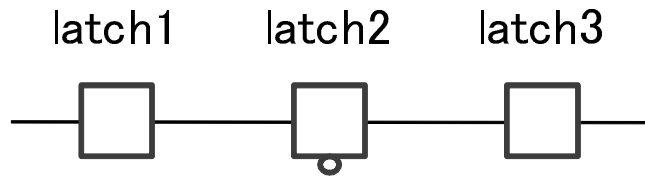
(d)



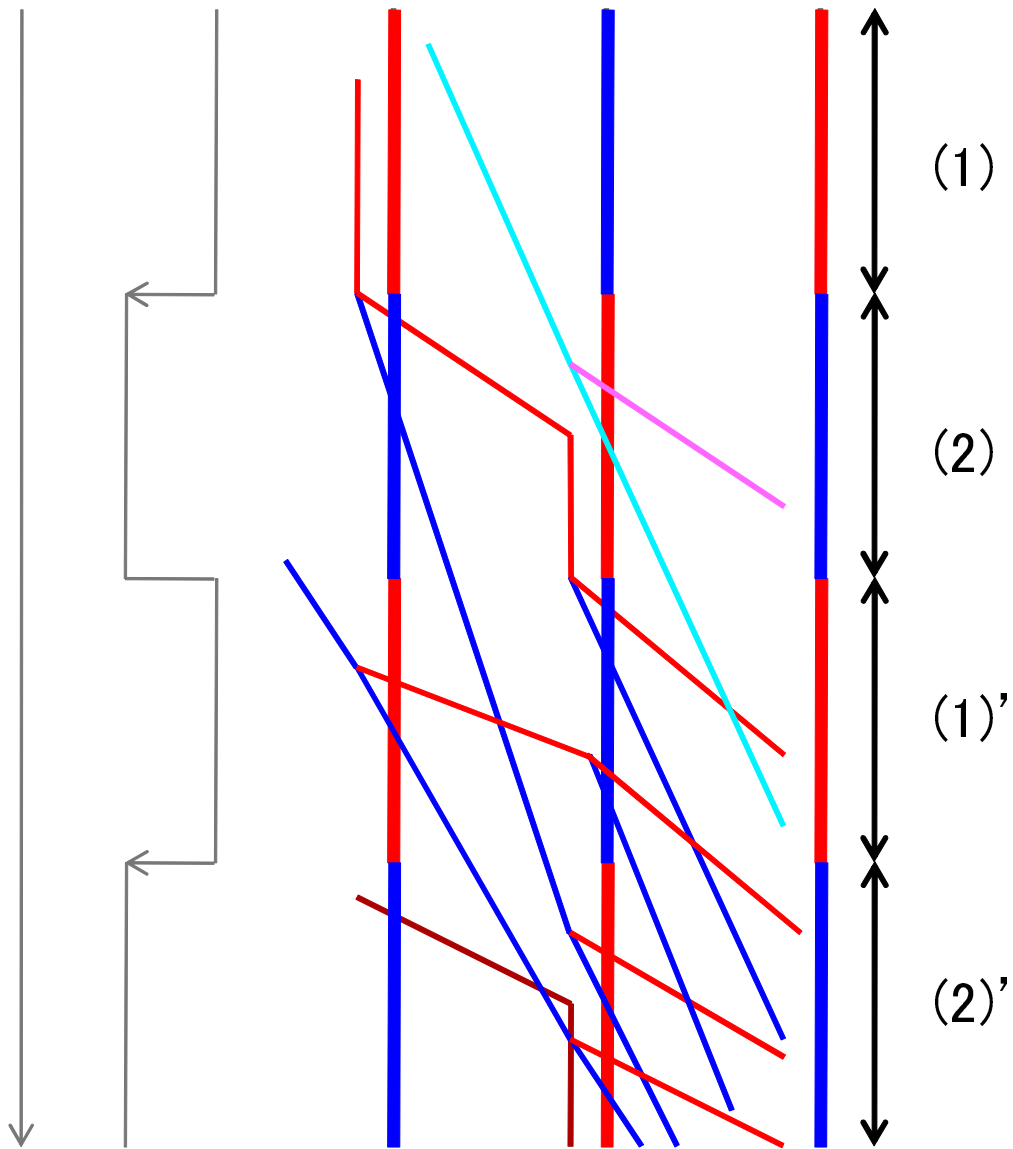
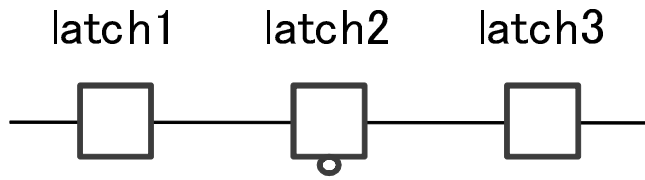
(e)



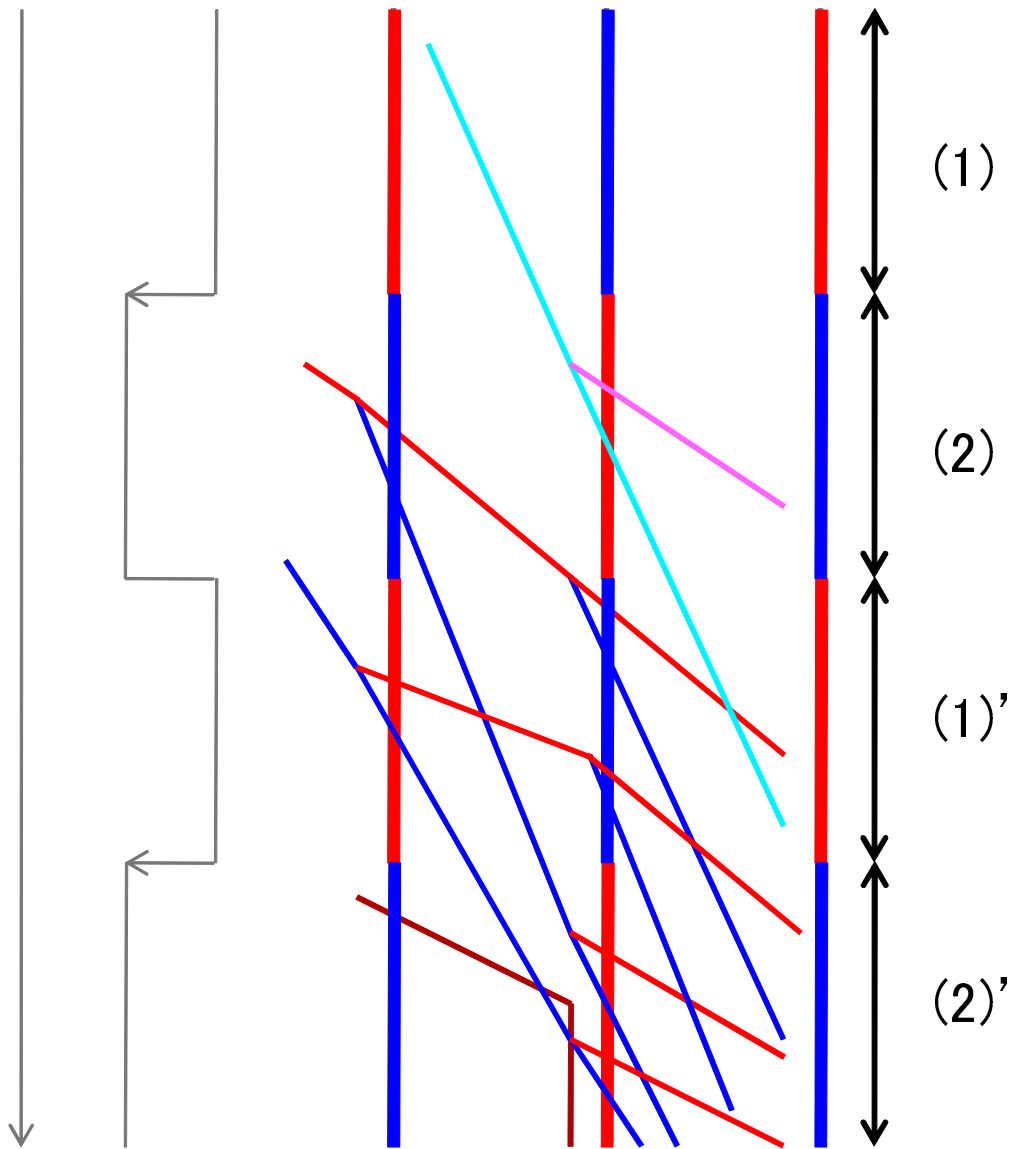
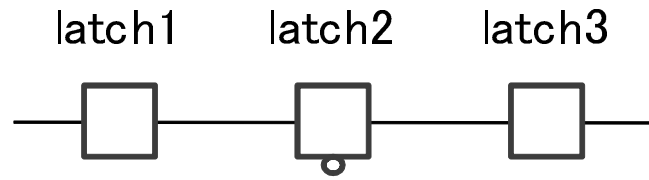
(f)



(g)



(h)



(i)

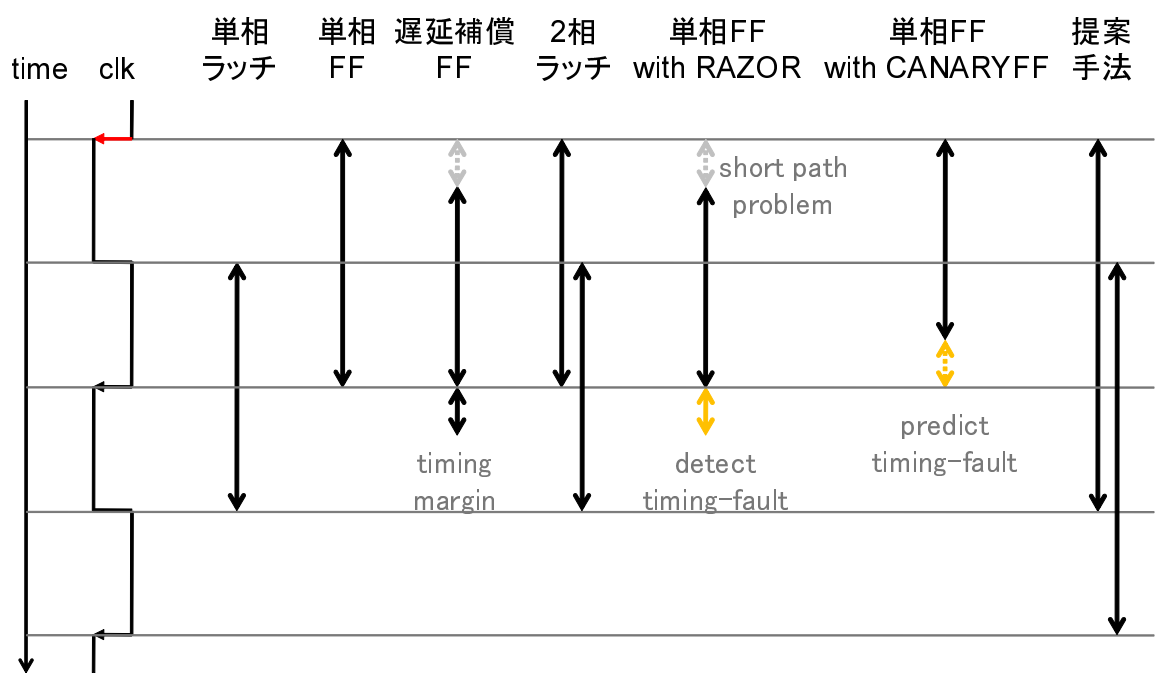


図 6.2: タイミング制約の比較

第7章 おわりに

7.1 まとめ

本論文では、年々深刻化していくプロセスバラつきやタイミング・フォールトへの解決策として、タイミング制約の緩いクロッキング方式を提案した。2相ラッチ間の回路を、ショート・パスとクリティカル・パスに分離することで前後のフェーズの信号の混在を防ぐ..そしてショート・パスの結果と、ショート・パスとクリティカル・パスの合流後の結果をセレクタで制御することで、誤った信号の伝達を防ぐ。

7.2 今後の課題

現段階では評価を取れていないため、今後、シミュレーションによってハードウェアコストや動作可能周波数の評価を取り、実現可能性について考察する必要がある。そして、FPGA 上に実装し動作を確認することにも意味があると考えられる。

また、より効率的な信号の利用法を模索し、プロセスバラつきの問題を解決する手法を提案していきたい。

参考文献

- [1] T. Austin, D. Blaauw, T. Mudge, and K. Flautner. Making Typical Silicon Matter with Razor. *IEEE Computer*, 2004.
- [2] D. Ernst, N.S. Kim, S. Das, S. Pant, T. Pham, R. Rao, C. Ziesler, D. Blaauw, T. Austin, and T. Mudge. Razor: A Low-Power Pipeline Based on Circuit-Level Timing Speculation. *Int. Symp. on Microarchitecture*, 2003.
- [3] S. Mukhopadhyay, H. Mahmoodi, and K. Roy. Modeling of Failure Probability and Statistical Design of SRAM Array for Yield Enhancement in Nanoscaled CMOS. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 24, No. 12, 2005.
- [4] S.Das, D.Roberts, S.Lee, S.Pant, D.Blaauw, T.Austin, K.Flautner, T.Mudge. A Self-Tuning DVS Processor using Delay-Error Detection and Correction. *IEEE Journal of Solid-State Circuits (JSSC)*, April 2006.
- [5] 杉本健, 入江英嗣, 五島正裕, 坂井修一. タイミングエラー耐性を持つスーパースカラプロセッサ. 並列/分散/協調処理に関する「佐賀」サマー・ワークショップ, 2008.
- [6] 岡田健一. 集積回路における性能ばらつき解析に関する研究. 京都大学博士論文, 2003.
- [7] 広瀬健一郎. 動作時信号遷移監視に基づく遅延補償フリップフロップの研究. 東京大学修士論文, 2008.
- [8] 佐藤寿倫. カナリア・フリップフロップを利用する省電力マイクロプロセッサの評価. 先進的計算基盤シンポジウム SACSIS, pp. 227–234, 2007.
- [9] 佐藤寿倫, 国武勇次. ばらつき耐性を持つカナリアFFを利用したデザインマージン削減による省電力化. 情報処理学会論文誌, pp. 2029–2042, 2008.
- [10] 五島正裕. デジタル回路. 数理工学社, 2007.

発表文献

主著論文

- 脆弱性検出のための静的値範囲解析
樽井 翔，勝沼 聡，入江 英嗣，五島 正裕，坂井 修一
電子情報通信学会技術研究報告 CPSY2007-22, pp. 95-100 (2007)
- タイミングフォールト耐性を持つクロッキング方式
樽井 翔，塩谷 亮太，五島 正裕，坂井 修一
電子情報通信学会研究報告 CPSY2008-14, pp. 25-30 (2008)

謝辞

非常に多くの方々から多大なご指導，ご協力，励ましを頂き，本論文を完成させることができました．この場を借りて，感謝の意を表したいと思います．本研究を進めるにあたり，坂井修一教授，五島正裕准教授から，大変多くのご指導を頂きました．ここに深く感謝の意を表します．

塩谷亮太氏をはじめ，研究室のメンバーの皆様には，ミーティングにおける議論を通して，貴重なご意見を頂きました．

八木原春水さん，伊世知代さん，長谷部環さんには，研究室における設備の導入や各種事務手続きなど，研究室で過ごすための様々なご支援を頂きました．