

準同期式回路における遅延最適化による クロック高速化

森下 和明 高橋 篤司 梶谷 洋司

東京工業大学 電気・電子工学科
〒152 東京都目黒区大岡山 2-12-1

各レジスタにおいて、クロック周期は同じであるがクロックの入るタイミングに時差を設けた回路を準同期式回路という。準同期式回路では、クロック周期の決定にレジスタ間の信号遅延の最大値と最小値が重要な意味をもち、レジスタ間の遅延が大きい方が高速化が可能である場合がある。本稿では、レジスタ間の信号遅延を解析し、遅延が小さすぎるために回路の高速化のネックになっているレジスタ間を特定する手法を示し、回路の高速化の手法の一つとして、線形計画法を用いた遅延素子挿入による準同期式回路の高速化手法を示す。またベンチマーク回路を用いた実験により、最大で 18.6% の高速化を達成した。

Clock-period minimization by delay optimization on the semi-synchronous circuit

Kazuaki MORISHITA , Atsushi TAKAHASHI and Yoji KAJITANI

Department of Electrical and Electronic Engineering, Tokyo Institute of Technology
2-12-1 Ookayama, Meguro-ku, Tokyo 152, Japan

The circuit on which the clock arrival time to each register is controlled is called *semi-synchronous circuit*. In deciding the clock-period, the maximum and minimum delay between registers is critical. In some cases, the semi-synchronous circuit can be faster by increasing the delay between some registers. Our proposing algorithm consists of two phases; the first is to find registers between which the delay decides the minimum clock-period, the second is to insert buffers between registers in order to make the semi-synchronous circuit faster. Experimental results on LGSynth91 benchmarks show that these techniques achieve maximum reduction of 18.6% compared with the semi-synchronous circuits in which buffers are not inserted.

1 序論

近年、集積回路設計において、回路機能の多様化や高速化などの高度な要求を満たすために、高速な同期式回路を実現する新たな方法が求められている。

同期式回路におけるクロックは、その性質上すべてのレジスタに同時に到達することが要求される。しかし、現実には、設計、製造、仕様に関する様々な要

因のために、クロックスキューと呼ばれる到達時刻のバラツキが生じる [3, 5, 7]。クロックスキューは、クロック周期短縮の妨げになるばかりでなく、回路の誤動作の原因にもなる。回路の微細化、大規模化、クロック周期の短縮に伴い、クロックスキューの性能に与える影響が相対的に大きくなり、クロックスキューを抑える配線技術に関心が高まった。

近年、ゼロスキューを目指した配線経路、バッファ

挿入、配線幅などに関する研究 [4, 6, 8] が相次いで発表され、成果が徐々に実際の回路に用いられている。

しかし、ゼロスキュー配線が達成されたとしてもクロック分配配線の総長や消費電力の増大、レジスタが同時に動作することによる消費電力の変動などがより重要な問題点として残る。

これらの問題点は、同一周期のクロックがすべてのレジスタへ同時に到着することを前提とする(完全)同期式回路の欠点といえる。大域的クロックを用いることの利点を失わず、先述した欠点を補う方法として、同一のクロックが、必ずしも同時ではなくレジスタへ到着することを前提とする準同期式回路が [1] で提案された。その回路でクロックの到着時刻の差を積極的に利用するならば、回路の信頼性を確保しながらさらなる性能向上が可能である。

本稿では、この準同期式回路において、遅延素子の挿入によるクロックの高速化手法を提案する。

従来の完全同期式回路では、クロック周期の下限はレジスタ間の最大パス遅延に依存する。したがって、クロック周期を小さくする為に、すべてのレジスタ間で信号遅延が最大であるパスを探索し、そのパスの配線長を短くするような配置改善手法、素子遅延や配線遅延を減少させる配線幅やトランジスタのサイジング手法が用いられる。またレイアウト上での改善が困難な場合には、論理合成まで遡っての最適化がなされる。

準同期式回路においては、高橋ら [1, 2] により、回路の信号遅延から以下のようにして最小クロック周期が求められる。まず、制約グラフをレジスタを点としレジスタ a から b に最小遅延 w_{min} 、最大遅延 w_{max} で信号が伝達される時、 a から b へ向かう重さ w_{min} を持つ D-枝、 b から a へ向かう重さ $-w_{max}$ を持つ Z-枝と呼ばれる 2 種類の有向枝を付加することにより作成する。2 種類の枝 D-枝、Z-枝をもつグラフにおいて、ある有向閉路の枝重みの総和をその閉路に含まれる Z-枝の本数で割った値をその閉路の Z-枝平均重みと呼ぶ。また、グラフ中の全ての有向閉路についての Z-枝平均重みの最小値を MCZ(Minimum Cycle Z-mean) と呼ぶ。このとき回路の最小クロック周期 T は制約グラフの $-(MCZ)$ となる。これを求める効率的なアルゴリズムも提案されている。

このことから、制約グラフの MCZ を大きくするような信号遅延の変更が可能であれば、最小クロック周期を短縮できることが分かる。すなわち、各枝の重みを大きくするように信号パス遅延制御を行えば良い。Z-枝は負の重み、D-枝は正の重みであることから、最大遅延は小さく、最小遅延は大きくすれば良い。このことは、準同期式回路を高速化させるには、従来のように信号遅延を小さくするばかりではなく、特定のレジスタ間では逆に信号遅延を大きくした方が最小クロック周期をより小さくできる場合があることを意味する。

本稿では、D-枝に対応する信号パス上に遅延素子

を挿入することにより、最小クロック周期を短縮する手法を提案する。また、実験した 9 つの回路においては最大 18.6% の高速化に成功した。

本稿の構成は次のようになる。第 2 節で準同期式回路における最小クロック周期の計算方法を紹介する。第 3 節ではクロック周期の下限を決めている制約グラフ上の閉路に含まれる枝集合を特定するアルゴリズムを示す。このことから、クロック高速化のためにはどのレジスタ間の信号遅延を最適化すればよいかの判断が可能となり、準同期式回路においても従来の完全同期式回路のようにクロック周期の下限の原因を解明し、それを回路の設計にフィードバックすることが可能になる。第 4 節では信号遅延を大きくした方がクロック周期を小さくできる場合について、遅延素子をレジスタ間に挿入することにより準同期式回路の高速化をはかる手法を提示する。第 5 節では実験結果を示し、第 6 節で結論を述べる。

2 準同期式回路における最小クロック周期

準同期式回路における最小クロック周期の計算方法 **CMCZ(Computation of the Minimum Cycle Z-mean)** は [1, 2] に記述されているが、ここで簡単に紹介する。

はじめにレジスタ間の遅延が与えられた回路から制約グラフ G_0 を作成する (図 1 参照)。

回路の最小クロック周期 T は、制約グラフの MCZ を B としたとき、

$$T = -B$$

と与えられる。この最小クロック周期 T を求める効率的なアルゴリズムは [1, 2] で説明されているがここでは省略する。

図 1 の例では $a-b-d-a$ の閉路の Z-枝平均重みは -9 であり、全ての閉路の中で最小である。この回路の最小クロック周期は 9 である。

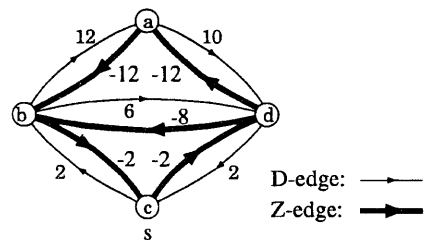


図 1: 制約グラフ G_0

準同期式回路において、あるレジスタを基準レジスタとし、そのクロックの到着時刻を基準時刻とす

る。基準時刻に対して他のレジスタにおけるクロックの到着時刻の進みを正としてこの値をクロック時差と呼ぶ。各レジスタにおけるクロック時差を決定することをクロックスケジューリングと呼ぶ。

G_0 において、全ての Z-枝の重みにクロック周期 T の重みを加えたグラフを G_T とすると、各レジスタにおけるクロック時差は G_T において、基準レジスタからそのレジスタに対応する点までの最小パス重みとして与えることができる。

3 クロック周期を決定する閉路

従来の完全同期式回路では最小クロック周期を決定する条件はレジスタ間の最大遅延であった。しかし、準同期式回路においては最小クロック周期は MCZ により決定される。この節では閉路重みが MCZ である閉路 (MCZ 閉路) に含まれる全ての枝からなる集合を求める手法を示す。ここで得られた枝集合に対応するレジスタ間の信号遅延を最適化することにより、クロック周期を小さくすることが可能となる。

まず、クロック周期を特定する閉路に含まれる枝を列挙するための基本原理として以下の定理を示す。

定理 1 最小クロック周期を達成するクロックスケジューリングにおいて、制約グラフの MCZ 閉路に含まれる枝の両端点のクロック時差の差は、その枝の重みと等しい。¹

証明

任意の 2 レジスタを u, v とし、 u から v への信号遅延の最大値を $w_{max}(u, v)$ 、最小値を $w_{min}(u, v)$ とする。最小クロック周期を達成するクロックスケジューリングにおいて、 u, v のクロック時差をそれぞれ $d(u), d(v)$ とすると G_T において任意の u, v について

$$\begin{aligned} d(v) - d(u) &\leq w_{min}(u, v) \text{ (D-枝の場合)} \\ d(v) - d(u) &\leq T - w_{max}(u, v) \text{ (Z-枝の場合)} \end{aligned}$$

が成り立つ [2]。

ある MCZ 閉路に含まれる枝 (u', v') の両端点のクロック時差の差と (u', v') の枝重みが等しくないとする。このとき、

$$\begin{aligned} d(v') - d(u') &< w_{min}(u', v') \text{ (D-枝の場合)} \\ d(v') - d(u') &< T - w_{max}(u', v') \text{ (Z-枝の場合)} \end{aligned}$$

となる。この MCZ 閉路に含まれるすべての枝について上記の式を列挙し辺々加えると

$$0 < \sum_{\text{D-枝}} w_{min} + \sum_{\text{Z-枝}} (T - w_{max})$$

¹逆は否である。

となる。したがって、

$$\begin{aligned} T &> - \frac{[\sum_{\text{D-枝}} w_{min} + \sum_{\text{Z-枝}} (-w_{max})]}{\text{MCZ閉路に含まれる Z-枝数}} \\ &= - \frac{\text{MCZ閉路の重み}}{\text{MCZ閉路に含まれる Z-枝数}} \\ &= -(MCZ) \end{aligned}$$

となり、

$$T = -(MCZ)$$

に矛盾する。

(証明終り)

枝の両端点のクロック時差の差とその枝の重みが等しい枝を列挙し、それらの枝によりグラフを構成したとき、有向閉路を構成する枝が MCZ 閉路に含まれる枝である。

系 1 MCZ 閉路は枝の重みとその両端点のクロック時差の差の等しい枝からなるグラフの強連結成分に含まれる。

MCZ 閉路を特定するアルゴリズム **FMCZ** (Find the cycle of the Minimum Cycle Z-mean) の概要は図 2 のようになる。Step.4 でラベル *Target* のつけられた枝が MCZ 閉路に含まれる枝になる。 G'_T から *Target* のつけられた枝以外の枝を取り除いたグラフを G_{Target} と呼ぶ。この G_{Target} 中に存在する全ての閉路が MCZ 閉路である。

FMCZ (Find the cycle of the Minimum Cycle Z-mean)

- Step 1.** アルゴリズム CMCZ により最小クロック周期 T を求める。
- Step 2.** G_T において、ソースから各点までの最小パスを求め、その値を各点のクロック時差とする。(図 3 の四角で囲まれた値)
- Step 3.** 両端点のクロック時差の差とその枝の重みが等しい場合、その枝にラベル *Candidate* をつける。(図 3 の C のついた枝)
- Step 4.** ラベル *Candidate* のつけられた枝以外を取り除いたグラフ G'_T において、強連結成分に含まれる枝にラベル *Target* をつける。(図 4 の T のついた枝)

図 2: FMCZ (Find the Minimum Cycle Z-mean)

4 遅延素子の挿入によるクロック高速化

4.1 遅延素子の挿入

最小クロック周期を小さくすることとは G_0 における MCZ を大きくすることにはほかならない。

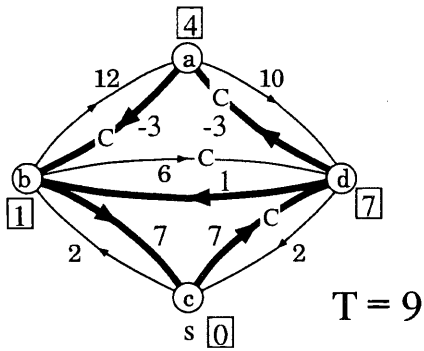


図 3: G_T

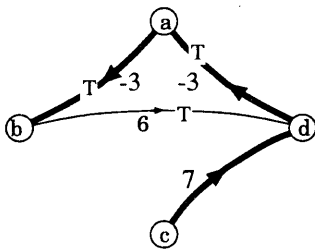


図 4: G'_T

すなわち、MCZ 閉路に含まれる枝の重みを大きくすればよい。

G_0 には Z-枝と D-枝が存在する。Z-枝の重み ($-w_{max}$) を大きくするという事は、信号遅延の最大値を下げることであり、D-枝の重み (w_{min}) を大きくするという事は、信号遅延の最小値を上げることであり、通常、最大遅延は設計段階で限界まで最小化されており、更に小さくすることは困難であると考えられる。一方、最小遅延を大きくすることは、遅延素子の挿入などにより比較的容易に実現できると考えられる。ここでは、回路を高速化するために D-枝の重みを大きくする、すなわち、信号遅延遅延の最小値を大きくすることを考える。

レジスタ間の信号遅延の最小値を大きくする為にバッファなどの遅延素子を挿入する。このとき、最小遅延が増大するような遅延素子の挿入を考えるが、最大遅延に関しては少なくとも減少することはない。

クロック周期を小さくするためには、全ての MCZ 閉路の重みが大きくなるなければならない。 G_{Target} から遅延が変化した枝を取り除いたとき、閉路が存在するならば、Z-枝平均重みが変化しない MCZ 閉路が存在することになり、クロック周期は改善され

ない。したがって、少なくとも G_{Target} から遅延が変更された枝を取り除いたとき、無閉路にならない。

もし、Z-枝のみからなる MCZ 閉路が存在したならば、遅延挿入により Z-枝の重みが減少することはない。したがって、クロック周期の改善は不可能である。以降 G_{Target} の各閉路は D-枝を含むとする。

また、通常、レジスタ a-b 間にのみ遅延素子を挿入すること (レジスタ a-b 間のみ信号遅延を変更すること) は不可能である場合が多い。a-b 間だけでなく他のレジスタ間の遅延も変化してしまう (例えば a-c 間や d-b 間の信号遅延も変化させてしまう) 場合が多い。しかし、問題の簡略化のために特定のレジスタ間のみ信号遅延を変更できるものとする。

以降、遅延素子としてバッファを用いる。

4.2 最大遅延と最小遅延が変化する場合

バッファ挿入によるレジスタ間の最大遅延と最小遅延の変化はレジスタ間の回路構造に依存する。ここでは、バッファの挿入により最小遅延が変化だけでなく、最大遅延も最小遅延と同量だけ変化するものとする。すなわち、D-枝の重みだけでなく対になっている Z-枝の重みも同様に変化する。また、遅延の変化量は挿入するバッファのサイズに比例すると仮定する。

バッファをどのレジスタ間に挿入するかは大きな問題である。ここでは問題の簡略化のために、MCZ 閉路に含まれる全ての D-枝の重みが増加するようにバッファを挿入するものとする。すなわち、それらの D-枝に対応するレジスタ間すべてにバッファを挿入する。さらに D-枝の重みの増加量は、全ての D-枝について同じであるとする。仮定より、対になっている Z-枝の重みも同様に増加する。

各閉路の Z-枝平均重みの増加量は、閉路に含まれる (重みの変化する D-枝数) - (重みの変化する Z-枝数) に比例し、Z-枝数に反比例する。したがって、全ての閉路に対し、バッファ挿入前の Z-枝平均重みと、挿入するバッファサイズに対する Z-枝平均重みの増加率を求めることにより、MCZ を最大化するようなバッファのサイズを求めれば良い。ただし、Z-枝平均重みとその増加率について、全ての有向閉路について逐一列挙する必要は無い。

(重みの変化する D-枝数) - (重みの変化する Z-枝数) と Z-枝数の等しい閉路を C_1, C_2 とする。このとき、 C_1 と C_2 の Z-枝平均重みの増加率は等しい。また、 C_1 のバッファ挿入前の Z-枝平均重みが C_2 よりも大きい場合には、いかなるバッファサイズに対しても C_1 の Z-枝平均重みは C_2 よりも大きい。したがって、(重みの変化する D-枝数) - (重みの変化する Z-枝数) と Z-枝数が等しい閉路については、その中でバッファ挿入前の Z-枝平均重みが最小の閉路のみを考えれば良い。よって、考慮すべき閉路の数は点数を n としたときに高々 $O(n^2)$ である。

バッファの挿入アルゴリズム OSBI(Optimal Size Buffer Insertion) の概略は図 5 のようになる。

OSBI(Optimal Size Buffer Insertion)

Step 1. FMCZ より MCZ 閉路に含まれる D-枝を特定する。

Step 2. Step 1. で求めた D-枝とそれに対応する Z-枝の重みが増化したときの、閉路のバッファ挿入前の Z-枝平均重みと、挿入するバッファサイズに対する Z-枝平均重みの増加率を MCZ 閉路となる可能性のある閉路に対して求める。(そのアルゴリズム CACZ(Computation of All of the Cycle Z-mean) については後述する)

Step 3. 線形計画法等を用いて、最小クロック周期及び挿入するバッファサイズを決定する。(図 6)

Step 4. 挿入可能なバッファサイズが 0 であつたら終了。そうでなければバッファを挿入して Step 1. へ。

図 5: OSBI(Optimal Size Buffer Insertion)

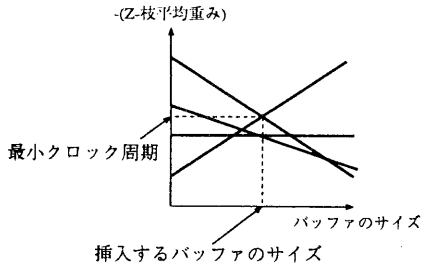


図 6: 最小クロック周期と挿入するバッファのサイズ

図 6 の各直線はある閉路について挿入するバッファのサイズとその閉路の Z-枝平均重みとの関係を表す。Step.2 の CACZ で各直線の傾きと縦軸の切片を求めている。縦軸の切片は閉路のバッファ挿入前の Z-枝平均重みを表し、傾きは挿入するバッファサイズに対する閉路の Z-枝平均重みの増加率を表す。CACZ については以降の節にて説明する。

4.3 Z-枝平均重みとその変化量

制約グラフに含まれる閉路は、重みの変化する枝を含む閉路と含まない閉路に分けられる。

BAI で用いる CACZ(Computation of All of the Cycle Z-mean) は、重みの変化する枝を含む

閉路と含まない閉路に分けてバッファ挿入前の Z-枝平均重みとその増加率を求める。

4.3.1 変化枝を含む閉路

この節では、変化枝を含む閉路のバッファ挿入前の Z-枝平均重みとその増加率を求めるアルゴリズムを示す。

この節で用いる用語を以下に定義する。

変化枝: バッファ挿入により重みの変化する D-枝を正の変化枝、Z-枝を負の変化枝と呼ぶ。

変化数 k : (正の変化枝数) - (負の変化枝数)。

$L(v, i, j, k)$: G_0 中の任意の点を始点 s としたときの、Z-枝数 i 、枝数 j 、変化数 k で s から点 v に至る walk の最小重み。

はじめに G_0 においてある変化枝 (u, v) に着目し、 G_0 から (u, v) を取り除いたグラフに対し、全ての i, j, k ($0 \leq j \leq n, 0 \leq i \leq j, -j \leq k \leq j$) について Z-枝数 i 、枝数 j 、変化数 k で v から点 u に至る walk の最小重み $L(v, i, j, k)$ を求める。

$L(v, i, j, k)$ を求めるアルゴリズム

CL(Computation of $L(v, i, j, k)$) の概略は図 7 のようになる。図式による説明が図 8 である。このアルゴリズムは、最短パスを求めるアルゴリズムの拡張である。なお、枝 (u, v) の重みを $w(u, v)$ で表している。図 8 の点線の枝が正もしくは負の変化枝である。

CL(Computation of $L(v, i, j, k)$)

s が与えられたとき、初期値 $L(s, 0, 0, 0) = 0$ とする。全ての i, j, k ($0 \leq j \leq n, 0 \leq i \leq j, -j \leq k \leq j$) と全ての枝 (u, v) について、以下を実行。

1. (u, v) が正の変化枝であるならば、

$$L(v, i, j, k) = \min(L(v, i, j, k), L(u, i, j - 1, k - 1) + w(u, v))$$
2. (u, v) が負の変化枝であるならば、

$$L(v, i, j, k) = \min(L(v, i, j, k), L(u, i - 1, j - 1, k + 1) + w(u, v))$$
3. (u, v) が D-枝であるならば、

$$L(v, i, j, k) = \min(L(v, i, j, k), L(u, i, j - 1, k) + w(u, v))$$
4. (u, v) が Z-枝であるならば、

$$L(v, i, j, k) = \min(L(v, i, j, k), L(u, i - 1, j - 1, k) + w(u, v))$$

図 7: CL(Computation of $L(v, i, j, k)$)

(u, v) が正の変化枝であるとき、Z-枝数 i 、総枝数 $j - 1$ 、変化数 $k - 1$ の v から u の walk の最小重みは $L(u, i, j - 1, k - 1)$ である。 (u, v) を含む Z-枝数 i 、総枝数 j 、変化数 k の閉路のバッファ挿入前の最小重

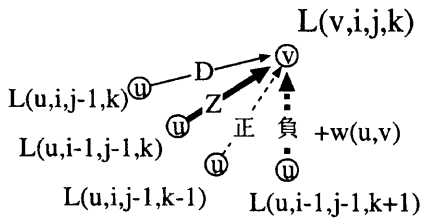


図 8: walk 重み $L(v, i, j, k)$ の求め方

みは $L(u, i, j-1, k-1) + w(u, v)$ となる。このとき、閉路の増加率は k/i である。増加率が k/i である閉路の中で、Z-枝平均重みが最小である閉路の Z-枝平均重みは、 $\min_{1 \leq j \leq n} (L(u, i, j-1, k-1) + w(u, v))$ となる。

(u, v) が負の変化枝であるとき、 (u, v) を含む Z-枝数 i 、総枝数 j 、変化数 k の閉路のバッファ挿入前の最小重みは $L(u, i-1, j-1, k+1) + w(u, v)$ となる。このとき、閉路の増加率は k/i である。増加率が k/i である閉路の中で、Z-枝平均重みが最小である閉路の Z-枝平均重みは、 $\min_{1 \leq j \leq n} (L(u, i-1, j-1, k+1) + w(u, v))$ となる。

増加率の種類は $O(n^2)$ であるが、図 6 における縦軸の切片と傾きの関係から、実際には OSBI に渡す必要のあるパラメータの数は少ない。

4.3.2 変化枝を含まない閉路について

変化枝を含まない閉路の Z-枝平均重みのバッファを挿入することによる増加率は 0 である。したがって、変化枝を含まない閉路の中で最も Z-枝平均重みの小さい閉路を探せば良い。

制約グラフ G_0 から変化枝を取り除いたグラフを \hat{G}_0 とする。変化枝を含まない閉路の中で、最も Z-枝平均重みの小さい閉路の Z-枝平均重みが \hat{G}_0 の MCZ となる。

4.3.3 CACZ(Computation of All of the Cycle Z-mean)

CACZ は図 9 のようになる。

4.4 最小遅延のみが変化する場合

前節のようにレジスタ間にバッファを挿入するとき、最大遅延と最小遅延がともに等しく変化する場合、全ての枝が Z-枝である MCZ 閉路が存在しないとしても、クロック高速化が不可能の場合がある。例えば、MCZ 閉路が、図 11 にあるように D-枝と Z-枝

CACZ(Computation of All of the Cycle Z-mean)

Step 1. G_0 の、全ての变化枝 (u, v) について以下を行う。

1. G_0 から (u, v) を削除したグラフにおいて v を始点として、 $L(u, i, j, k)$ を求める。(CL)
2. (u, v) を含む増加率が等しい閉路の中で、Z-枝平均重みが最小である閉路の Z-枝平均重みを全ての増加率に対しそれぞれ求める(図 10)。

Step 2. \hat{G}_0 の MCZ を求める。

Step 3. 全ての異なる増加率に対し、増加率とそのバッファ挿入前の最小 Z-枝平均重みを OSBI に返す。

図 9: CACZ(Computation of All of the Cycle Z-mean)

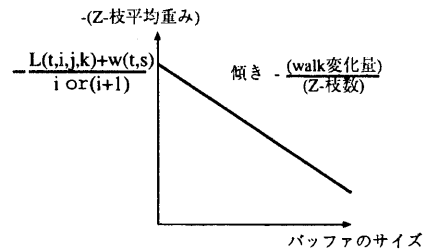


図 10: Z-枝平均重みとその変化量

それぞれ 1 本からなる閉路の場合である。このような閉路を **DZ-閉路** と呼ぶ。

最大遅延と最小遅延がともに等しく変化するという仮定のもとでは、DZ-閉路に対応するレジスタ間に遅延 D_{buf} のバッファを挿入しても、Z-枝の重みの変化分は $-D_{buf}$ 、D-枝の重みの変化分は $+D_{buf}$ となり、閉路の重みは全く変化しない。

レジスタ間の最大遅延と最小遅延の差が非常に大きい場合には、そのレジスタ間に対応する DZ-閉路が MCZ 閉路となり、最小クロック周期の下限を決定することがある。信号遅延の最大値と最小値の差は、ゲートの立ち上り立ち下がり遅延の差や、信号パスが多重パスになっていた場合のそれぞれのパスのゲート段数の差によるものである。

例えば、図 12 のような多重パスが存在したとする。あるパスを伝達する信号の遅延の最小値が w_{min} 、最大値が w_{max} のとき $[w_{min}, w_{max}]$ と表す。S からゲート E に至るパス毎の信号遅延は図にある通りで

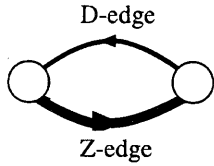


図 11: DZ-閉路

ある。このとき $W_{min}^1 \leq W_{min}^2$, $W_{max}^1 \leq W_{max}^2$ であるならば, 図 13 にあるように, A-E 間にバッファを挿入することにより, このレジスタ間の最大遅延を変化させずに最小遅延のみを増大させることができる。すなわち, DZ-閉路において, Z-枝の重みを変化させることなく, D-枝の重みのみを増大させることができる。したがって, DZ-閉路の重みは増大し, クロック周期を短縮できる可能性がある。このとき, 挿入可能なバッファ遅延の最大値は,

$$D_{buf} = \begin{cases} W_{max}^2 - W_{max}^1 & (W_{max}^2 - W_{max}^1 \leq W_{min}^2) \\ W_{min}^2 - W_{min}^1 & (others) \end{cases}$$

となる。

通常, レジスタ間にバッファを挿入した場合, 他のレジスタ間の信号遅延に影響を与えることに注意しなければならない。しかし, 本稿では問題の単純化のために「あるレジスタ間にバッファを挿入しても, 他のレジスタ間の信号遅延に影響は与えない。」という仮定を設ける。

MCZ 閉路である DZ-閉路が存在し, その DZ 閉路に対応するレジスタ間の最大遅延を増大させることなくバッファの挿入が可能であるとき, 挿入可能な最大のバッファをそのレジスタ間の多重パス上に挿入する。この挿入を MCZ 閉路である DZ-閉路が存在する限り繰り返す手法を **DZBI(DZ-cycle Buffer Insetion)** とする。

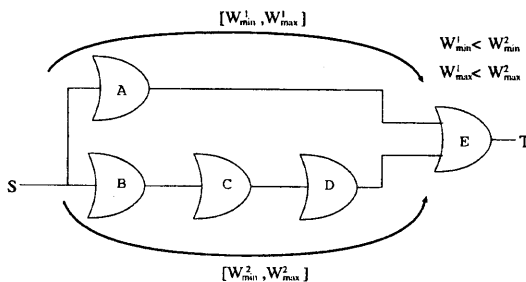


図 12: 多重パスの例

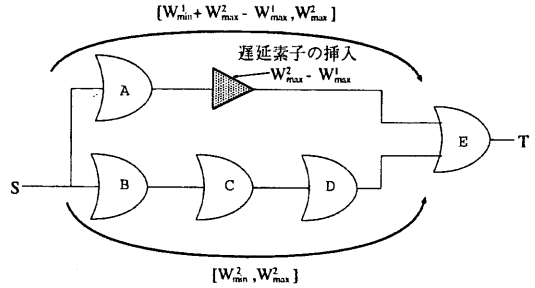


図 13: 多重パスにバッファを挿入する

5 実験結果

用いたベンチマーク回路は MCNC ベンチマーク回路の LGSynth91 である。各回路の情報および CMCZ[2] を用いて求めた準同期式回路における最小クロック周期を表 1 に示す。表中の # Regs は回路中のレジスタ数, # IO は入出力端子数であり, 回路規模の目安を与える。 G_0 及び G_T にはこれらのレジスタに対応する点に加え, IO を一つにまとめた仮想的なレジスタに対応する点が含まれる。Max delay は回路中のレジスタ間の遅延の最大値であり, 従来の完全同期式回路におけるクロック周期の下限である。本実験では信号パス上に存在するゲートの最大遅延を足し合わせた値をレジスタ間の最大遅延としている。最小遅延についても同様である。Min clock は遅延の最適化を行わない場合の準同期式回路におけるクロック周期の下限である。最大遅延(完全同期式回路のクロック周期の下限)に対するクロック周期の減少率が red. で示されている。

表 refcircuitDATA 中の回路に対し OSBI 及び DZBI を施した場合の実験結果を表 2 に示す。OSBI clock は OSBI でバッファを挿入した場合の最小クロック周期であり, 挿入したバッファの遅延時間を Buffer size に示す。DZBI clock は DZBI でバッファを挿入した場合の最小クロック周期を示す。Min clock から OSBI clock 及び DZBI clock への減少率が red. に示す。

表中で — となっている項目は, それぞれ回路が OSBI 及び DZBI で改善できなかったことを示す。OSBI で改善できなかった回路は, DZ-閉路が MCZ 閉路となっていた。DZBI で改善できなかった回路は, DZ-閉路が MCZ 閉路でなかったか, DZ-閉路に多重パスが存在しない為に改善できなかった。

これらの実験結果によれば OSBI が適用可能な場合には劇的なクロック高速化が可能であり, 実験では 18.6% もの高速化に成功している。また多くの回路においては OSBI が適用不可能であっても DZBI により数% の高速化が可能であることが示された。

Circuit name	# Regs (# IO)	# of Gates	Max delay	Min clock	red. [%]	Circuit name	Min clock	OSBI clock	Buffer size	DZBI clock	red. [%]
s298	23 (9)	75	4.94	3.89	21.2	s298	3.89	—	—	3.84	1.3
s344	15(20)	101	9.55	8.89	6.9	s344	8.89	7.24	0.887	—	18.6
s349	15(20)	104	9.55	8.89	6.9	s349	8.89	7.24	0.887	—	18.6
s382	21(9)	99	5.78	4.01	30.6	s382	4.01	—	—	3.90	2.7
s444	21(9)	119	6.43	4.72	26.6	s444	4.72	—	—	—	0
s526	21(9)	141	4.93	3.89	21.1	s526	3.89	—	—	3.84	1.3
s526n	21(9)	140	4.93	3.89	21.1	s526n	3.89	—	—	3.84	1.3
s1488	6(27)	550	10.34	9.81	5.1	s1488	9.81	—	—	9.68	1.3
s1494	6(27)	558	10.75	10.21	5.0	s1494	10.21	—	—	9.81	3.9

表 1: 回路情報と最小クロック周期 ([1][2])

表 2: OSBI 及び DZBI を適用した場合

プロジェクトの一部である。

6 結論

準同期式回路において最小クロック周期を決定する要因を求めることを可能にした。また、遅延素子を挿入することで信号遅延の最小値を増加させるアルゴリズムを提示し、レジスタ間の信号遅延の最小値がクロック周期の下限を決める要因となっている場合について、遅延素子を挿入することで回路の高速化させる本手法の有用性を実験により示し、準同期式回路を最大 18.6% 高速化させることに成功した。

本研究における実験は、遅延時間の見積りに単純な手法を用いている。このためレジスタ間の最大遅延及び最小遅延の差が必要以上に大きく見積もられている。正確な見積もりを行えば、本手法によってクロック周期は更に改善されると思われる。したがって、配線遅延なども考慮した上で、より正確な最大遅延と最小遅延の見積もり手法の開発が必要である。

本実験では MCZ 閉路に含まれる全ての D-枝に対応するレジスタ間にバッファを挿入したが、バッファを挿入するレジスタ間の選択にも工夫が必要である。

また、本手法では、レジスタ間に遅延素子を挿入した場合、他のレジスタ間の遅延に影響を与えない、という仮定を設けた。この仮定はゲートを複製することで実現できるが、面積の増大等による遅延時間の変化が伴い現実的ではない。バッファの挿入による各レジスタ間の遅延の変化を正確に反映させた上で、挿入する遅延素子の大きさと最小クロック周期を求める手法が必要である。

謝辞

本研究を進めるにあたり真剣に討論して頂いた梶谷研究室諸賢に深謝する。なお、本研究は CAD21 プ

参考文献

- [1] A. Takahashi, M. Furuya, and Y. Kajitani, "Clock Period Minimization by Clock Skew Control," in *IEICE Technical Report VDL95-42*, pp. 85–92, 1995
- [2] A. Takahashi and Y. Kajitani, "Performance and Reliability Driven Clock Scheduling of Sequential Logic Circuits", to appear in *ASP-DAC '97*
- [3] R. B. Deokar and S. S. Sapatnekar, "A graph-theoretic approach to clock skew optimization," in *Proc. ISCAS '94*, pp. 1:407–410, 1994.
- [4] M. Edahiro, "A clustering-based optimization algorithm in zero-skew routings," in *Proc. 30th DAC*, pp. 612–616, 1993.
- [5] J. P. Fishburn, "Clock skew optimization," *IEEE Trans. on Computers*, vol. 39, no. 7, pp. 945–951, 1990.
- [6] J. L. Neves and E. G. Friedman, "Circuit synthesis of clock distribution networks based on non-zero clock skew," in *Proc. ISCAS '94*, pp. 4:175–178, 1994.
- [7] T. G. Szymanski, "Computing optimal clock schedules," in *Proc. 29th DAC*, pp. 399–404, 1992.
- [8] R. S. Tsay, "Exact zero skew," in *Proc. 1991 ICCAD*, pp. 336–339, 1991.