

スケジュール可能範囲を考慮したクロック木合成手法

東 昌秋, 斉藤 誠, 高橋 篤司

東京工業大学 集積システム専攻

E-mail: {azuma, makoto, atushi}@ss.titech.ac.jp

各レジスタへ同時刻にクロックを分配することを前提としない準同期式回路において、レジスタ集合がいくつかの半径の小さいクラスタに分割され、クロックスケジュールがクラスタ内のレジスタのクロックタイミングが等しいという条件で与えられたとき、そのクロックスケジュールを実現するクロック木の構成法を提案する。提案手法は配線長の最小化を目的とするクラスタ内配線と、配線長および使用するバッファ数の削減を目的とするクラスタ間配線により、レジスタ間のクロックタイミングの差を満たし、バッファからそのバッファが駆動するレジスタやバッファまでのパス長を配線遅延を無視できる範囲に抑えた、低消費電力化を実現するクロック木を構成する。実験により、提案手法により得られる準同期式設計におけるクロック木は完全同期式設計に比べ、12%の低消費電力化が達成できることが確かめられ、本手法の有効性が確認された。

A Clock-Tree Routing Algorithm for Low Power Using Feasible Range of Clock Schedule

Masaaki AZUMA, Makoto SAITOH, and Atsushi TAKAHASHI

Dept. of Communications and Integrated Systems, Tokyo Inst. of Tech.

In the semi-synchronous circuit, which permits the clock to be input on different timing to each register, we propose a clock-tree routing algorithm that realizes a given clock schedule for the set of registers partitioned into several clusters. In each cluster, the clock timings of registers are assumed to be equal. We keep the path length from a buffer to each register or to each buffer driven by the buffer so small that wire delay is small enough to be ignored compared with buffer delay. The algorithm consists of two phases, routing inside the clusters and routing among the clusters. In experiments, the power of the clock-tree constructed by the algorithm is reduced by 12% compared with that of complete synchronous framework.

1 はじめに

近年 LSI の高速化と大規模化によりクロック配線の配線長が増大し、クロックが消費する電力の割合が大きくなるなど、従来の方式が限界に近づいていると言われている。そのため各レジスタへクロックを異なるタイミングで入力することを許容する設計方式、準同期式設計が提案され、その実用化へ向けた検討が進められている [1, 3, 4, 5, 6, 7, 8]。この設計方式では、各レジスタにクロックを分配するタイミング、クロックスケジュールをレジスタ間の最大・最小遅延などにより決定し、クロック木の構造、バッファ数、配線長を制御することによりクロックを各レジスタへ決定されたタイミングに到着させる。

しかし、クロック木の構築を考慮せずにクロックスケジュールを設定すると、それを実現するためのクロック木の配線長が非常に長くなる [1, 3]。そのためクロック木の配線長を短くするために、物理的に近いレジスタに近いクロックタイミングを設定する

など、レイアウトを考慮してクロックスケジュールを設定しなければならない。しかし、クロックスケジュールがクロック木構築にとって好ましいか否かをクロック木構築の前に知ることは難しい。そのため、回路が正常動作するという条件のもとで、各レジスタのクロックタイミングに範囲を持たせ、その範囲の中から適当なタイミングを選択することを許せば、よりよいクロック木を得ることが可能となる。

また、実用を考えると遅延変動時にも回路の正常動作を保証しなければならない。配線遅延を用いてスケジュールを制御すると、遅延変動時に誤動作する可能性が高くなるため、配線遅延がバッファ遅延に比べ無視できるほど小さくなるように、あるバッファからそのバッファが駆動するレジスタおよびバッファまでの配線長に制限を加え、クロック木を構築する手法が提案されている [6]。

本稿では、配線遅延を用いず、クロックタイミングの範囲を利用し、消費電力の削減を目的とするク

ロック木の構成法を提案する。

クロックスケジュールは、レジスタ集合がいくつかの半径の小さいクラスタに分割され、クラスタ内のレジスタのクロックタイミングが等しいという条件で与えられる。また各レジスタに対しスケジュール可能範囲が与えられる。クラスタの半径はクラスタに属するレジスタを囲む最小菱形領域の半径で与える。チップ内のレジスタ集合をクラスタ半径が小さいいくつかのクラスタに分割する手法は、例えばチップ領域を菱形領域に分割し各領域に属するレジスタ集合を初期クラスタとし、それらを修正する方法 [8] や、SA によりクロックタイミングを離散的に設定し、その後クラスタ分割する方法 [6] などがある。

提案手法はクロック木をクラスタ内配線とクラスタ間配線の 2 段階で構築する。クラスタ内配線はバッファから各レジスタへのパス長を制約とし、配線長の最小化を目的とするスタイナー木により構成する。クラスタ間配線は、クロックソースを根とし各クラスタを葉とする 2 分木を低消費電力を達成しつつ、スケジュール可能範囲をできる限り広く保つように、ボトムアップに構築する。

この方法により完全同期式設計に比べて、17% の高速化を達成するクロックスケジュールを実現するクロック木を構築した。得られた木の 1 周期当りの消費電力は完全同期式設計のクロック木に対して 12% 減少した。

2 準備

あるバッファからそのバッファが駆動するレジスタおよびバッファまでの配線が短ければ、配線遅延はバッファ遅延に比べ無視できるほど小さい。配線遅延を無視できる最大の配線長を L_{lim} とする。

各バッファの遅延 D は、そのバッファが駆動する容量を C とすると、バッファに固有の定数 D_i, D_t および定数 A を用いて、 $D = A \times (D_i + D_t \times C)$ で表される。また各バッファに対して駆動可能な限界の容量 (C_{lim}) が与えられる。

各バッファの消費電力 P はバッファに固有の定数 P_i, P_t により、 $P = P_i + P_t \times C$ と見積もる。構成されたクロック木の消費電力は、バッファの消費電力の和により見積もる。

一般にサイズの大いバッファは、遅延が小さく限界駆動容量が大いだが、消費電力も大きくなる。

問題は以下のように定式化される。

入力：各レジスタの位置とスケジュール可能範囲、および所属するクラスタ

出力：クロック木

制約：クロックスケジュールの実現、バッファの駆動容量の制約、バッファからのパス長の制約

目標：消費電力の最小化

同じクラスタに属するレジスタのスケジュール可能範囲は共通部分を持つ。この共通な範囲をそのクラスタのスケジュール可能範囲と呼ぶ。

クロック木の生成はクラスタ内配線とクラスタ間配線の 2 段階のアルゴリズムにより行う。クラスタ内配線は各クラスタを 1 つのバッファで駆動することを前提にし、そのバッファから各レジスタまでのパス長を L_{lim} 以下に抑えることにより、クラスタ内のレジスタへのクロックの到着時刻を同一にする。

クラスタ間配線はクロックソースから各クラスタまでの配線を、クラスタ間のスケジュール差を満たすように、バッファを適宜挿入することにより行う。

クラスタ間配線に用いるバッファは一種類のみとする。これ以降 $D_i, D_t, P_i, P_t, C_{lim}$ はこのバッファに対する値とし、このバッファの入力容量を C_{in} とする。

3 クラスタ内配線

クラスタ内配線は総配線長および根から葉までのパス長を共に小さく抑えることを目的とした矩形スタイナー木の構成アルゴリズムである CRBST により行う [2, 9]。その際、重心に最も近いレジスタを根とし他のレジスタを葉とする。CRBST ではパラメータ k の値を変化させることにより、総配線長とパス長のどちらを重視するかを変えることができる。 k の値が 0 に近いほど総配線長が重視され、1 に近いほどパス長が重視される。根から葉までのマンハッタン距離の最大値を L_{max} とすると、CRBST で構成されるスタイナー木上での根から葉までのパス長が L_{max}/k 以下に抑えられることが知られている。そこで、 k の値を L_{max}/L_{lim} とすることにより、根から葉までのパス長を L_{lim} 以下とし、配線遅延が無視できる範囲内に抑える。ただし $L_{max} \leq L_{lim}$ であるとする。

クラスタ内配線の容量が C_{lim} よりも大きいクラスタに対しては、根の位置にそのクラスタを駆動できるバッファの中でサイズが最小のものを挿入する。クラスタ内配線の容量はクラスタ内のレジスタの入力容量と配線容量の和である。クラスタ内配線の容量が C_{lim} 以下のクラスタに対しては、クラスタ内配線ではバッファを挿入を行わず、クラスタ間配線時にバッファ挿入位置を決定する。

4 クラスタ間配線

クラスタ間配線はスケジュールクロック木生成手法 (SC) [1, 3] を応用した手法を用いて行う。

4.1 アルゴリズム SC

アルゴリズム SC はクロックソースを根とする 2 分木を、与えられたクロックスケジュールを実現するように構成する。

SC はまず、ボトムアップ方式で再帰的に要素数が 1 となるまでマージ操作を実行して、クロック木のトポロジーを生成する。マージ操作は、2 つの要素 (レジスタまたは部分木) の親点を作り、親点から 2 つの要素のそれぞれの親点へ配線をする。親点は 2

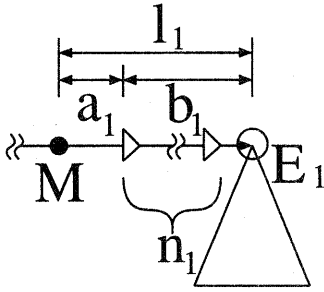


図 1: バッファが挿入された場合

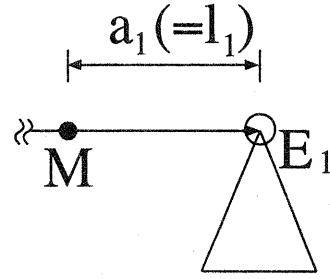


図 2: バッファが挿入されない場合

つの要素間に所望のスケジュール差を生じさせることができる点の中で、配線長が最小となる点とする。このような点の候補は線分または点となる。これを候補線分と呼ぶ。

次にトップダウン方式でクロック木の決定を行なう。このとき子の候補線分の中で、親の位置からの距離が最小となる点を選ぶ。

アルゴリズム SC では 2 つの要素のマージを配線長を重視したコスト関数により選択していたが、提案手法のコスト関数では電力を重視する。またバッファ挿入を最大配線長 L_{lim} を満足するように行う。

4.2 マージの制約

クラスタ間配線は原点をクロックソースとし、各クラスタを最初の要素として SC を実行することにより行う。各要素は親点の「位置」、「スケジュール可能範囲」、「未駆動容量」、その要素の親点の位置から次段のバッファおよびレジスタまでの最大パス長（「要駆動長」と呼ぶ）の 4 つの情報を持つ。最初の要素であるクラスタに関しては、「位置」はクラスタ内配線の根となったレジスタの位置である。クラスタ内配線においてバッファが挿入されなかったクラスタに関しては、「スケジュール可能範囲」はクラスタのスケジュール可能範囲、「未駆動容量」はクラスタ内配線の配線容量とレジスタの入力容量の和、「要駆動長」はクラスタ内配線の根から葉までのパス長の最大値となる。バッファが挿入されたクラスタに関しては、「スケジュール可能範囲」はクラスタのスケジュール可能範囲から挿入されたバッファの遅延を引いた範囲、「未駆動容量」は挿入されたバッファの入力容量、「要駆動長」は 0 となる。

親点の位置が E_1 、スケジュール可能範囲が $[S_{min1}, S_{max1}]$ 、未駆動容量が c_1 、要駆動長が e_1 である要素を考える。ある点 M から E_1 まで配線がなされ、その配線上に n_1 個のバッファが挿入されていたとする。 M から E_1 までの配線長を l_1 、 M から配線上の M に最も近いバッファまでの配線長（未駆動配線長）を a_1 、残りの配線長（既駆動配線長）を b_1

($= l_1 - a_1$) とする（図 1 参照）。また配線の単位長さ当たりの容量を c とする。このとき、 M から E_1 までの遅延時間 d_1 は、

$$d_1 = A(n_1 D_i + D_t(b_1 c + C_{in}(n_1 - 1) + c_1))$$

である。遅延時間 d_1 は、バッファ数 n_1 と既駆動配線長 b_1 により変化する。

M におけるスケジュール可能範囲は $[S_{min} - d_1, S_{max} - d_1]$ 、未駆動容量 c'_1 は $a_1 c + C_{in}$ 、要駆動長 e'_1 は a_1 となる。

M から E_1 まで配線上にバッファが挿入されていない、すなわち $n_1 = 0$ の場合（図 2 参照）、遅延時間 $d_1 = 0$ 、未駆動容量 c'_1 は $l_1 c + c_1$ 、要駆動長 e'_1 は $l_1 + e_1$ となる。

この配線による消費電力は、

$$n_1 P_i + P_t(l_1 c + v_1 + n_1 C_{in})$$

とする。

同様に M から、スケジュール可能範囲が $[S_{min2}, S_{max2}]$ 、未駆動容量が c_2 、要駆動長が e_2 である別の要素の親点 E_2 までの配線がなされていたとする。その配線上に n_2 個のバッファが挿入され、 M から E_2 までの配線長を l_2 、未駆動配線長を a_2 、既駆動配線長を $b_2 (= l_2 - a_2)$ 、 M から E_2 までの遅延時間を d_2 、 M から E_2 までの配線から見積もられる M の未駆動容量を c'_2 、要駆動長 e'_2 とする。

このとき M のスケジュール可能範囲は、両側から計算される M のスケジュール可能範囲の共通の範囲

$$[\max(S_{min1} - d_1, S_{min2} - d_2), \min(S_{max1} - d_1, S_{max2} - d_2)]$$

となり、未駆動容量は $c'_1 + c'_2$ 、要駆動長は $\max(e'_1, e'_2)$ となる。

M のスケジュール可能範囲が存在するためには、

$$\max(S_{min1} - d_1, S_{min2} - d_2) \leq \min(S_{max1} - d_1, S_{max2} - d_2)$$

を満たす必要がある。この条件を満たすためには、

$$S_{min1} - d_1 \leq S_{max2} - d_2$$

$$S_{min2} - d_2 \leq S_{max1} - d_1$$

となる必要がある。したがって d_1 と d_2 の間に次のような関係があれば、 M のスケジュール可能範囲が

存在する.

$$S \min_1 - S \max_2 \leq d_1 - d_2 \leq S \max_1 - S \min_2$$

クロック木において M の末駆動容量は C_{lim} 以下でなければならない. 要駆動長は L_{lim} 以下でなければならない. したがって

- $c'_1 + c'_2 \leq C_{lim}$
- $\max(e'_1, e'_2) \leq L_{lim}$

でなければならない. $n_1 > 0, n_2 > 0$ の場合, この2つの条件を満たすための必要条件は,

$$a_1 + a_2 \leq \min(2L_{lim}, (C_{lim} - 2C_{in})/c) = Z$$

である.

また最大駆動容量と最大駆動長のバッファの制約から, b_1 は次の2つの条件を満たす必要がある.

- $b_1 c + c_1 + C_{in}(n_1 - 1) \leq n_1 C_{lim}$
- $b_1 + e_1 \leq n_1 L_{lim}$

したがって

$$b_1 \leq \min((n_1 C_{lim} - C_{in}(n_1 - 1) - c_1)/c, n_1 L_{lim} - e_1) = X$$

でなければならない. b_2 に関しても同様に容量と駆動長の制約を満たす必要がある.

$$b_2 \leq \min((n_2 C_{lim} - C_{in}(n_2 - 1) - c_2)/c, n_2 L_{lim} - e_2) = Y$$

SC ではマージする際のコスト (マージングコストと呼ぶ) が最小となる要素のペアに対してマージを実行するが, 本稿では2つの要素は以下のようにマージするものとして, マージングコストとしてマージする際に必要なバッファ数と配線長から見積もられる消費電力を与える.

4.3 2要素のマージ

2つの要素のマージは

- (1) 消費電力の最小化
- (2) マージにより生成される新たな要素のスケジュールの範囲の最大化
- (3) マージにより生成される新たな要素の要駆動長の最小化

を目標として行う. このときバッファの制約条件を満たす必要がある.

制約条件を満たす解を得るために, マージに使用するバッファ数を0から順に1ずつ増やす. 各バッファ数で, バッファをマージする2つの要素への配線へ振り分ける全ての場合を考える. バッファのそれぞれの振り分け方に対するマージは, バッファの制約を満たし目標を達成する b_1, b_2, a_1, a_2 を定めることにより実行される.

迂回配線のないマージ可能な解が得られたとき, それ以上バッファ数を増やさずに終了する. 配線長 l' の迂回配線を含むマージ可能な解が得られたとする. 2つの要素のマージにおける消費電力は, バッファ数および迂回配線長により変化する. バッファ数を1つ増加させたとき, 消費電力はバッファの増加により $P_i + P_i C_{in}$ 増加し, 迂回配線の減少により高々

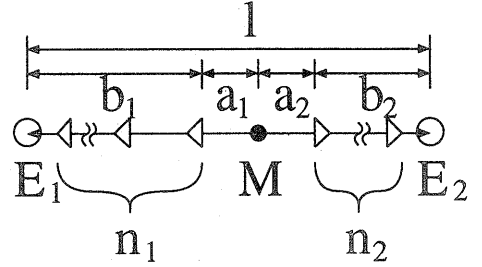


図 3: $n_1 > 0, n_2 > 0$ の場合

$P_i l' c$ だけ減少する. したがって, $P_i + P_i C_{in} > P_i l' c$ であるとき, バッファ数を増加させても消費電力が減少することはないため, それ以上バッファ数を増やさず終了する. 得られたマージ可能な解の中から, 目標を達成する解をマージ候補とする.

E_1, E_2 間のマンハッタン距離を l とする. それぞれの要素のスケジュールの目標値 s_1, s_2 として, スケジュール可能範囲の中での中間値を与える. この目標値のスケジュール差を達成できたとき2つの要素のマージにより生成される新たな要素のスケジュール可能範囲の幅が最大となることが保証されている. 目標値のスケジュール差が達成できるように b_1, b_2 を定める. 目標値のスケジュール差が達成できない場合には, できる限りその目標値のスケジュール差に近づけるように b_1, b_2 を定めることにより, マージにより生成される要素のスケジュール可能範囲の幅の最大化が達成される.

n_1, n_2 の値により E_1, E_2 のマージを以下のように場合分けして考える. (1) $n_1 > 0, n_2 > 0$, (2) $n_1 = 0, n_2 = 0$, (3) $n_1 > 0, n_2 = 0$ または $n_1 = 0, n_2 > 0$

4.3.1 $n_1 > 0, n_2 > 0$ の場合

b_1 の制約 $b_1 \leq X$, b_2 の制約 $b_2 \leq Y$, および $a_1 + a_2$ の制約 $a_1 + a_2 \leq Z$ の3制約を満たす必要がある. $X + Y + Z < l$ であるときには3制約を満たすことができないので実現不可能であり, このバッファ挿入の仕方は考慮しない. 以下では $X + Y + Z \geq l$ の場合を考える.

2つの要素の目標スケジュールの差を満たすとき, $s_1 - d_1 = s_2 - d_2$ が成り立つ. この式から2つの要素の目標スケジュールの差を実現するための b_2 と b_1 の差が求まる.

$$b_2 - b_1 = \frac{s_2 - s_1}{AD_{tc}} - \frac{D_i(n_2 - n_1)}{D_{tc}} - \frac{c_2 - c_1 + C_{in}(n_2 - n_1)}{c} = W$$

この W の値により以下のように場合分けする.

	クロック周期 [ps]	総配線長 [μm]	バッファ数	消費電力 [$\mu\text{W}/\text{MHz}$]
提案手法	9543	29493	80	55.4
完全同期	11569	25947	84	62.8

表 1: 実験結果

1. $0 \leq W \leq l$ の場合

3 制約および $b_1 + b_2 + a_1 + a_2 = l$ および生成される要素のスケジュールに関する制約,
 $S \min_1 - S \max_2 \leq d_1 - d_2 \leq S \max_1 - S \min_2$
を満す $b_1, b_2, a_1 + a_2$ の中で, スケジュール可能範囲を最大化するために, $|b_2 - b_1 - W|$ が最小となる $b_1, b_2, a_1 + a_2$ を選び, その中で次段の要駆動長を最小化するために $a_1 + a_2$ を最小とする $b_1, b_2, a_1 + a_2$ を求める. そのような $b_1, b_2, a_1 + a_2$ が存在すればマージ可能であり, 次段の要駆動長を最小化するために $a_1 = a_2 = (l - b_1 - b_2)/2$ とする. そのような $b_1, b_2, a_1 + a_2$ が存在しなければ, そのバッファの振り分け方ではマージ不可能である.

2. $W > l$ の場合

3 制約および $b_1 + b_2 + a_1 + a_2 \geq l$ および
 $S \min_1 - S \max_2 \leq d_1 - d_2 \leq S \max_1 - S \min_2$
を満す $b_1, b_2, a_1 + a_2$ の中で, 消費電力の最小化のために迂回配線が最小となる, すなわち $b_1 + b_2 + a_1 + a_2$ が最小となる $b_1, b_2, a_1 + a_2$ を選び, スケジュール可能範囲を最大化するために, $|b_2 - b_1 - W|$ を最小とする $b_1, b_2, a_1 + a_2$ を求める. そのような $b_1, b_2, a_1 + a_2$ が存在すればマージ可能であり, 次段の要駆動長を最小化するために $a_1 = a_2 = \max((l - b_1 - b_2)/2, 0)$ とする. そのような $b_1, b_2, a_1 + a_2$ が存在しなければ, そのバッファの振り分け方ではマージ不可能である.

3. $-l \leq W < 0$ の場合

$b_1 = -W, b_2 = 0$ として 1. の場合と同様にマージする.

4. $W < -l$ の場合

$b_1 = -W, b_2 = 0$ として 2. の場合と同様にマージする.

4.3.2 $n_1 = 0, n_2 = 0$ の場合

このときマージ可能なのは, 2つの要素のスケジュール可能範囲に共通の範囲が存在し, $l + e_1 + e_2 \leq 2L_{lim}$ および $lc + c_1 + c_2 \leq C_{lim}$ を満す場合である. マージ可能な場合は次段の要駆動長を最小化するため, $a_1 + e_1$ と $a_2 + e_2$ の差が最小となるように a_1, a_2 を決める.

4.3.3 $n_1 = 0, n_2 > 0$ または $n_1 > 0, n_2 = 0$ の場合

$n_1 = 0, n_2 > 0$ の場合について説明する.

(1) の場合と同様に $b_1, b_2, a_1 + a_2$ を定める. ただし, $n_1 = 0$ のとき $b_1 = 0$ であり, b_1 に関する制約は考える必要がない. また $a_1 + a_2$ に関する制約は
 $a_1 + a_2 \leq \min(2L_{lim} - e_1, (C_{lim} - C_{in} - c_1)/c)$
となる. a_1, a_2 は次段の要駆動長を最小化するため, $a_1 + e_1$ と a_2 の差が最小となるように決める.

5 実験結果

このアルゴリズムを $728 \times 710[\mu\text{m}]$ の領域にレイアウトされたレジスタ数 888 の回路を 40 個のクラスタに分割したものに對して, 下のような条件のもとで実行した.

$$L_{lim} = 300[\mu\text{m}]$$

$$P_t = 3.24[\mu\text{W}/(\text{MHz} \cdot \text{pF})]$$

$$c = 0.0002[\text{pF}/\mu\text{m}]$$

$$A = 2.17$$

クラスタ分割はチップ領域を菱形領域に分割し, 各領域内のレジスタ集合をそれぞれ初期クラスタとし, 次にそのクラスタ分割での最小クロック周期およびスケジュールを求め, 高速化の妨げとなるレジスタを隣接クラスタに移動させることを繰り返し, 最終的なクラスタ分割とクロックスケジュールを決定するアルゴリズム CBCS[8] により行った. 使用できるバッファとして 7 種類のバッファを考え, クラスタ間配線に用いるバッファは, 3 番目にサイズの小さいバッファを選んだ.

完全同期式設計に比べて 17% の高速化を達成するクロックスケジュールに對して, 提案手法によりクロック木を構築した. その結果, 得られたクロック木の 1 周期当りの消費電力は $55.4[\mu\text{W}/\text{MHz}]$ であり. 完全同期式設計におけるクロック木の 1 周期当りの消費電力, $62.8[\mu\text{W}/\text{MHz}]$ に對して 12% 減少させた. 表 1 に提案手法が構築するクロック木と完全同期式設計におけるクロック木の比較を示す. また提案手法が構築したクロック木を図 4 に示す.

6 終りに

準同期式回路において, レジスタ集合がいくつかの半径の小さいクラスタに分割され, クロックスケジュールがクラスタ内のレジスタのクロックタイミングが等しいという条件で与えられたとき, そのクロックスケジュールを実現するクロック木の構成法を提案した.

今後の課題としてはこのクロック木の構成法をふまえたクロックスケジュールの決定法や, バッファ

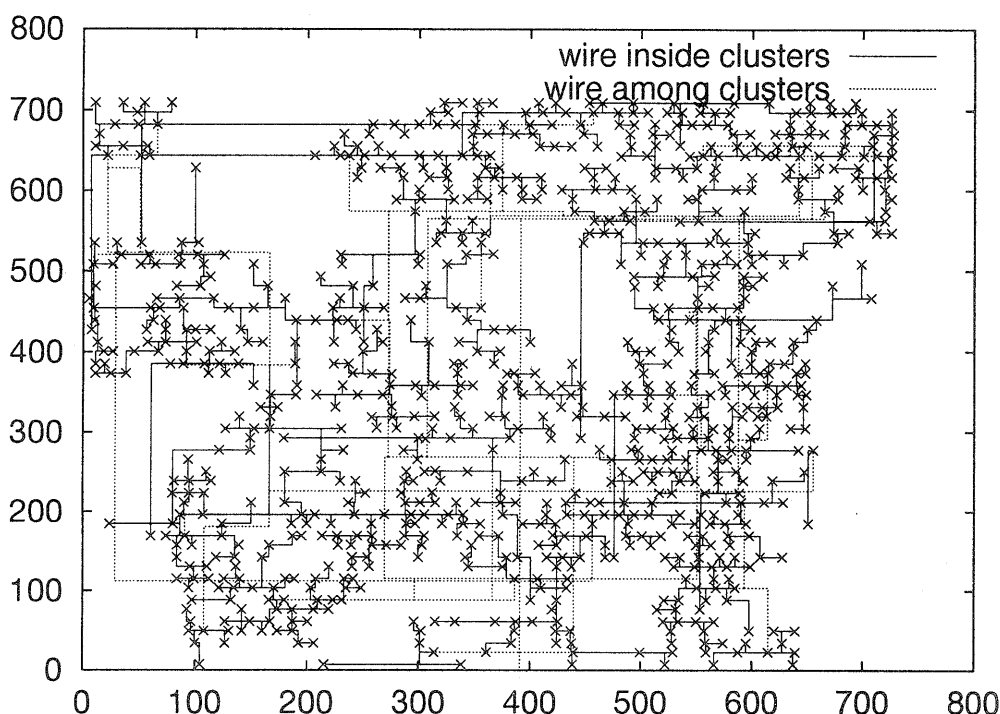


図 4: クロック木

の選択方法の改良, 遅延変動時に誤動作を起こさないためのクロック木の構成法などが挙げられる。

謝辞

本研究を進めるにあたり, 終始適切な御助言と暖かいご指導を下された東京工業大学 梶谷洋司教授, 議論に参加頂いた松下電器産業 豊永昌彦氏, 黒川圭一氏, 安井卓也氏に深く感謝します。本研究は CAD21 プロジェクトの一部である。

参考文献

- [1] K. Inoue, W. Takahashi, A. Takahashi, and Y. Kajitani. Schedule-clock-tree routing for semi-synchronous circuits. *IEICE Transactions on Fundamentals*, Vol. E82-A, No. 11, pp. 2431–2439, 1999.
- [2] H. Mitsubayashi, A. Takahashi, and Y. Kajitani. Cost-radius balanced spanning/steiner trees. *IEICE Transactions on Fundamentals*, Vol. E80-A, No. 4, pp. 689–694, 1997.
- [3] A. Takahashi, K. Inoue, and Y. Kajitani. Clock-tree routing realizing a clock-schedule for semi-synchronous circuits. In *Proc. 1997 ICCAD*, pp. 260–265, 1997.
- [4] A. Takahashi and Y. Kajitani. Performance and reliability driven clock scheduling of sequential logic circuits. In *Proc. ASP-DAC '97*, pp. 37–42, 1997.
- [5] A. Takahashi, W. Takahashi, and Y. Kajitani. Clock-routing driven layout methodology for semi-synchronous circuit design. In *Proc. TAU '97*, pp. 63–66, 1997.
- [6] M. Toyonaga, K. Kurokawa, T. Yasui, and A. Takahashi. A practical clock tree synthesis for semi-synchronous circuits. In *Proc. ISPD '00*, pp. 159–164, 2000.
- [7] T. Yoda and A. Takahashi. Clock period minimization of semi-synchronous circuits by gate-level delay insertion. *IEICE Transactions on Fundamentals*, Vol. E82-A, No. 11, pp. 2383–2389, 1999.
- [8] 斎藤誠, 高橋篤司. レイアウトを考慮したクラスタ分割によるクロックスケジューリング手法. 情報処理学会 DA シンポジウム 2000, pp. 39–42, 2000.
- [9] 三林秀樹, 高橋篤司, 梶谷洋司. Cost-radius balanced plane steiner tree. DA 97-85 (97-103), 情報処理学会研究報告, 1997.