

FPGA 実装を想定した束データ方式による 非同期式回路のフロアプラン手法の検討

齋藤 寛^{†1} 濱田 尚 宏^{†1}
米田 友 洋^{†2} 南 谷 崇^{†3}

束データ方式による非同期式回路において性能を決める要因は、遅延素子を含む制御回路の遅延である。したがって、実装の対象となる FPGA 上での制御回路の遅延を概算し、入力から出力までの制御回路の遅延の総和が最小となるようフロアプランを行う。

A Study of Floorplanning for Asynchronous Circuits with Bundled-data Implementation on FPGAs

HIROSHI SAITO,^{†1} NAOHIRO HAMADA,^{†1}
TOMOHIRO YONEDA^{†2} and TAKASHI NANYA^{†3}

A factor to decide the performance of asynchronous circuits with bundled-data implementation is the delay of the control circuit including delay elements. This paper shows a floorplan method for bundled-data implementation so that the sum of control delays from inputs to outputs is minimized while estimating the control delays on the target FPGA.

1. はじめに

回路全体をグローバルクロック信号で制御する同期式回路は、集積回路の微細化技術の向上において、以下の問題に直面する。ひとつは、配線遅延によるクロックスキューである。

クロックスキューが大きいとデータ転送の際、同期の失敗確率が増加する。次に、消費電力の問題である。集積度が高い回路に対して周波数の高いクロック信号を供給すると、クロックツリーにおける消費電力が大きくなる。更には、電磁放射の問題である。クロック信号に同期して回路が一斉に動作するため、集積度が高くなると電磁放射も大きくなる。

回路の各部分をローカルなハンドシェイク信号で制御する非同期式回路は、クロック信号がないので上で挙げた問題は同期式回路ほど深刻とはならない。また、実装手段にもよるが、平均遅延動作、環境変動に対する耐性といった利点がある。しかしながら、クロック信号がない分、同期式回路と比べ非同期式回路の設計は困難である。同期式回路の場合、全ての演算が同じタイミングで動作するので、演算が実行されるクロックサイクルとクロックサイクル時間のみを意識すればよいが、非同期式回路ではクロック信号がなく全ての演算が異なるタイミングで動作するので、演算毎に動作タイミングを考慮する必要がある。また、同期式回路の場合、ハザードといった予期せぬ信号遷移が起こったとしても、次のクロックサイクルまでに正しい値が生成されれば問題にはならないが、非同期式回路の場合、ハザードはそのまま伝搬されてしまう可能性がある。その為、ハザードのない回路、もしくはハザードが伝搬されない回路を実現する必要がある。

本研究の目的は、FPGA 実装を対象とした束データ方式による非同期式回路のフロアプラン手法の提案と評価である。束データ方式とは非同期式回路のデータエンコーディング手法の1つである。N ビットのデータを N 本の信号線と要求・応答信号の束で表し、演算の完了タイミングを要求信号線上にのった遅延素子によって保証する。フロアプランとは回路モジュールの配置を決め、レイアウト設計のための配置制約を生成することである。FPGA は再構成可能デバイスの1つで、近年では、プリンタやネットワーク機器関係などで利用が普及している。また、ツールも無償で提供され、回路モデルを HDL などと与えればプログラミングまで自動で行ってくれるため、開発コストを抑えることもできる。

FPGA に束データ方式による非同期式回路を実現するためにはまず、遅延素子を含む制御論理に最適化の制限をかける必要がある。遅延素子が最適化されてしまうと所望のタイミングを保証できなくなる。また、制御論理を分割するような最適化は、新たなハザードが起こる可能性がある。次に、性能のよい回路を得るためにはフロアプランが重要である。しかしながら、FPGA の開発ツールで性能のよい束データ方式による非同期式回路を生成することは難しい。こうしたツールは、そもそも同期式回路を対象としているため、同期式回路で効果のある最適化を行う。例えば、レジスタ間最大遅延を最小化し、クロックサイクルタイムを短くするといった方法である。一方、束データ方式による非同期式回路では、要求、応答信号の生成や遅延素子といった制御回路の遅延が性能を決定する。したがって、性能のよい回路を生成するためには、制御回路の遅延を考慮したうえでフロアプランを行い、生成された配置制約を用いて配置配線を行うことが重要である。

提案手法では、実装の対象となる FPGA の素子遅延や配線遅延を基に、束データ方式で

^{†1} 会津大学
The University of Aizu

^{†2} 国立情報学研究所
National Institute of Informatics

^{†3} 東京大学
The University of Tokyo

必要なタイミング制約を考慮しつつ、制御回路の遅延を最小化するようフロアプランを行う。また、最適化されたフロアプランを基に、遅延素子を決定する。こうしたことをシーケンスペア¹⁾とシミュレーテッドアニーリング(SA)を用いて実現し、こうした考慮がないものとフロアプラン後の性能とフロアプラン時間で比較を行う。

本稿の構成は以下のとおりである。2節では、本稿で用いる束データ方式による非同期式回路の回路モデルとFPGAについて解説する。3節では、提案手法を述べ、4節では実験結果を述べる。最後に5節で結論を述べる。

2. 準備

2.1 対象とする回路モデル

図1に、本稿で利用する束データ方式による非同期式回路の回路モデルを表す。このモデルは、データパス回路と制御回路からなる。データパス回路は、演算器(FU)、レジスタ(REG)、マルチプレクサ(MUX)、グルーロジック(g)からなる。グルーロジックは、ALUの制御信号(sel_{FU})、マルチプレクサの制御信号(sel_{MUX})、レジスタ書き込み信号(w_{REG})を生成するための論理である。一方、制御回路は制御モジュール(CTR)の集合からなる。ある制御モジュールCTR_i (0 ≤ i ≤ m-1)は、Qモジュール²⁾、2つの遅延素子(D_{sei}とD_{bhi})、グルーロジック(図3にあるような分岐判定論理)からなり、データパス回路上での演算を制御する。D_{sei}はレジスタのセットアップ時間制約を含めたレジスタの書き込みタイミングを制御し、D_{bhi}はホールド時間制約と分岐判定制約を満たすために要求される。

次に、QモジュールQ_iの動作を解説する。Qモジュールは、直前のQモジュールQ_{i-1}から入力されるin_i信号の立ち上がり遷移によって制御を開始する。演算がマルチプレクサや演算器を利用する場合、sel_{MUX}やsel_{FU}がin_iよりグルーロジックを介して生成される。in_iの立ち上がり遷移の到着後、Q_iは要求信号req_iを1にする。この値は、遅延素子D_{sei}を通り、応答信号ack_iを1としてQ_iに戻る。次に、Q_iはreq_iを0とし、ack_iが0になるのを待つ。レジスタにデータを書き込むための制御信号w_{REG}は、応答信号ack_iよりグルーロジックを介して生成される。なおデータは、ack_iの立ち下がり遷移時に、レジスタに書き込まれる。ack_iが0になった後、Q_iはout_iを1とし、制御を次のQモジュールQ_{i+1}に移す。

外部からの入力データは、最初のQモジュールQ₀によってレジスタに書き込まれ、最後のQモジュールQ_{m-1}によってレジスタに書き込まれた値が、外部への出力データとなる。Q_{m-1}でout_{m-1}を1とした後、全てのQモジュールにおけるin_iとout_iが順番に0となる。この間、データパス回路では演算は行われない。Q_{m-1}でout_{m-1}が0となった後、外部からの次の入力データがQ₀によってレジスタに書き込まれる。

以下では、信号の立ち上がり遷移をsignal+, 立ち下がり遷移をsignal-とする。

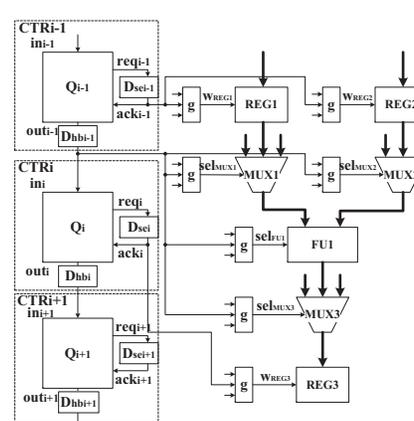


図1 回路モデル.
 Fig.1 Circuit model.

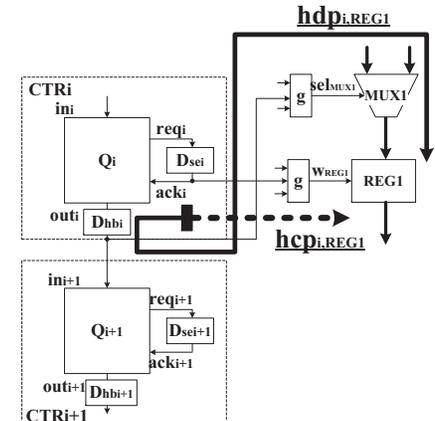


図2 hcp_{i,u}とhdp_{i,u}の例.
 Fig.2 Example of hcp_{i,u} and hdp_{i,u}.

2.2 タイミング制約

本稿で用いる回路モデルには以下の3つのタイミング制約が存在する。

- セットアップ時間制約 - セットアップ時間制約とは、レジスタにデータが書き込まれる時より一定の時間(セットアップ時間)前に書き込まれるデータが安定しなければならないという制約である。ここで、Q_{i-1}のack_{i-1}-からQ_iによって制御されるあるレジスタREGまでのデータパスをsdp_{i,v} (0 ≤ v ≤ j-1)とする。一方、ack_{i-1}-からCTR_iを経由したREGまでの制御パスをscp_{i,v}とする。d_{sdp_{i,v}}をsdp_{i,v}における最大遅延、d_{scp_{i,v}}をscp_{i,v}における最小遅延、d_{setup}をセットアップ時間、d_{sm_i}をd_{sdp_{i,v}}に対するマージンとすると、セットアップ時間制約は以下ようになる。

$$d_{scp_i} > d_{sdp_i} + d_{setup} + d_{sm_i} \quad (1)$$

- ホールド時間制約 - レジスタにデータが書き込まれた後一定の時間(ホールド時間)、レジスタの入力は変化してはいけないという制約をホールド時間制約と呼ぶ。本稿で用いる回路モデルでは、あるレジスタREGが連続する2つのQモジュールQ_iとQ_{i+1}によってデータが書き込まれる時、ホールド時間制約違反の可能性がある。Q_iのack_i-によってレジスタにデータを書き込んでいる最中に、out_{i+}によってREGの入力にあるマルチプレクサMUXの出力が変化する為である。仮に、Q_iとQ_{i+1}によってデータが書き込まれるレジスタがk個あるとする。ack_i-からCTR_iとMUXを介してREGに至るまでのデータパスをhdp_{i,u} (0 ≤ u ≤ k-1)、ack_i-からREGの出力ま

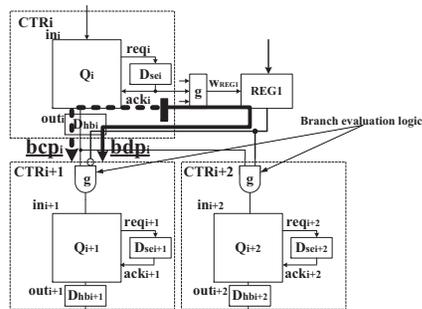


図3 $bdp_{i,w}$ と $bcp_{i,w}$ の例.
Fig. 3 Example of $bdp_{i,w}$ and $bcp_{i,w}$.

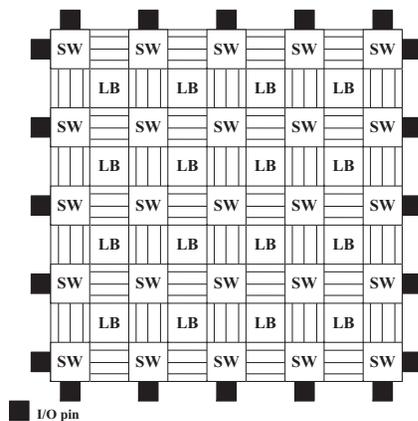


図4 FPGA.
Fig. 4 FPGA.

での制御パスを $hcp_{i,u}$ とする．図2は， $hcp_{i,u}$ と $hdp_{i,u}$ の例を表す． d_{hdp_i} を $hdp_{i,u}$ における最小遅延， d_{hcp_i} を $hcp_{i,u}$ における最大遅延， d_{hold} をホールド時間， d_{hm_i} を d_{hcp_i} に対するマージンとすると，ホールド時間制約は以下ようになる．

$$d_{hdp_i} > d_{hcp_i} + d_{hold} + d_{hm_i} \quad (2)$$

- 分岐判定制約 - Q_i によってデータが書き込まれたレジスタ REG の値によって制御が分かれる場合， Q_i からの out_i より先に REG の値が CTR_{i+1} の中にある分岐判定論理に到着してしまうと，間違った方向に分岐してしまう可能性がある．今， REG の値によって l の方向に制御が分かれると仮定する． ack_i から REG を介して CTR_{i+1} の中にある分岐判定論理までのデータパスを $bdp_{i,w}$ ($0 \leq w \leq l-1$)， ack_i から分岐判定論理までの制御パスを $bcp_{i,w}$ とする．図3は $bdp_{i,w}$ と $bcp_{i,w}$ の例を表す． d_{bdp_i} を $bdp_{i,w}$ における最大遅延， d_{bcp_i} を $bcp_{i,w}$ における最小遅延， d_{bm_i} を d_{bdp_i} に対するマージンとすると，分岐判定制約は以下ようになる．

$$d_{bcp_i} > d_{bdp_i} + d_{bm_i} \quad (3)$$

2.3 FPGA

FPGA は図4のように，論理ブロック (LB)，スイッチマトリックス (SW)，入出力ピンなどから構成される．論理ブロックは，さらに複数の look up table (LUT) と呼ばれるプログラム可能な論理とフリップフロップ (FF) から構成される．スイッチマトリックスは論理

ブロックと配線をつなぐスイッチであり，入出力ピンは外部に対する入出力に利用される．FPGA は規則的な構造をしているため，遅延の見積もりが ASIC などと比べて容易である．本稿では，FPGA の遅延を以下のように分類する．

- LUT 遅延 d_{lut}
- 論理ブロック内の配線遅延 d_{wl}
- 論理ブロック間の配線遅延 d_{wg}

d_{lut} は，LUT の入力から出力までの遅延である． d_{wl} は，論理ブロック内の LUT 間の配線遅延である． d_{wg} は，あるブロックの LUT から別のブロックの LUT までの配線遅延である．

d_{lut} はデータシートから得ることができる．例えば，Vertex 4 (xc4vlx25) では，0.15ns である⁴⁾． d_{wl} や d_{wg} はデータシートに記載されていない．これらの値は，使用する配線によって異なるため，固定ではない．そこで本稿では， d_{wl} と d_{wg} の値を概算する． d_{wl} は， d_{wg} と比べ配線が短いため，遅延もあまり大きくならない．ブロック内の LUT を使うよう接続関係のある論理を同一のブロックに配置し，静的タイミング解析ツールで配線遅延を得る．この操作を，複数のブロックで行ったときに得られた平均値を d_{wl} とする．一方，論理ブロック間の配線遅延は，他の遅延と比べ，正確に見積もることが難しい．これは，ブロックからブロックにたどり着くまで複数の配線を経由するためである．本稿では，ブロック間の LUT を使うよう接続関係のある論理を異なるブロックに配置し，静的タイミング解析ツールで配線遅延を得る．この操作を複数のブロックで行い，ブロック間のマンハッタン距離とそのときの配線遅延を基に，最小二乗法を用いて一次方程式 $d_{wg}(dist)$ を生成する． $dist$ はブロックからブロックまでのマンハッタン距離である．

3. 提案手法

3.1 概要

束データ方式の場合，性能は遅延素子を含む制御回路によって定まるので，制御遅延を最小化することが性能のよい回路を生成する上で重要である．また，遅延素子は面積のオーバーヘッドとなるので，遅延素子に必要となる素子数を最小化することが重要である．

提案手法では，SA とシーケンスペアをベースに，前節で述べたタイミング制約を考慮しつつ制御回路における遅延が最適となるようフロアプランを行う．フロアプランに際し，モジュールの論理遅延の最大値と $d_{wg}(dist)$ によるモジュール間配線遅延を基に制御遅延を概算する．フロアプラン後，生成されたフロアプランに最適な遅延素子を生成する．なお，この段階でタイミング制約を考慮したとしても，配置配線後の遅延によって制約違反が起こる可能性は十分にありえる．しかしながら，フロアプランの段階でタイミング制約を考慮したうえで遅延素子を生成することによって，制約違反の数が抑えられることが期待できる．以下の節で提案手法の詳細を解説する．

3.2 入 力

提案手法の入力は、モジュールファイル、パスファイル、デバイスファイル、SA におけるパラメータファイルである。

モジュールファイルは、 REG , MUX , FU , CTR の集合からなり、各モジュールはモジュール名、素子数、遅延、アスペクト比、マージンをパラメータとして持つ。素子数はそのモジュールを FPGA に実装した時に必要となる素子数、遅延はそのモジュールを論理合成した段階での遅延、アスペクト比はモジュールの幅と高さに対する比、マージンはモジュールの面積に対するマージンを表わす。素子数、アスペクト比、マージンを基に、各モジュールのサイズと幅と高さを定める。なお、グルーロジックは必要となるモジュールに含める。

パスファイルは、 CTR 毎にセットアップ時間制約、ホールド時間制約、分岐判定制約に対するパス ($sdp_{i,v}$, $scp_{i,v}$, $hdp_{i,u}$, $hcp_{i,u}$, $bdp_{i,w}$, $bcp_{i,w}$) とマージン (d_{sm_i} , d_{hm_i} , d_{bm_i}) を表す。

デバイスファイルは、対象となる FPGA の名前、最後の要素の座標、遅延からなる。遅延は前節で述べた d_{lut} , d_{wl} , $d_{wg}(dist)$ と Q モジュールにおける信号生成の遅延 d_{req+} , d_{req-} , $d_{out+}(req+, req-, out+)$ を生成するまでの遅延) からなる。

SA におけるパラメータファイルは、開始温度 $T_{initial}$, 終了温度 T_{end} , 温度毎の繰り返し回数 ite , クーリングファクタ cf からなる。

3.3 出 力

出力は配置制約ファイルと遅延素子ファイルである。配置制約ファイルは、各モジュールの配置を表わし、対象となる FPGA のベンダが定義するフォーマットで生成する。このファイルはフロアプラン後の配置配線で用いる。一方、遅延素子ファイルは、制約を満たすために必要となる遅延素子 (D_{se_i} と D_{hb_i}) 毎に Verilog HDL の形式で生成する。

3.4 フロアプラン

提案手法では、モジュール名によるシーケンスペアを任意に作成し、初期フロアプランとする。シーケンスペアはフロアプランの表現法の一つで、2 つのシーケンスにおけるモジュール名の位置によって、各モジュールの相対的な位置関係が定まる。また、モジュールのサイズをパラメータに制約グラフ¹⁾ を生成することによって、モジュールの X 座標と Y 座標が定まる。初期フロアプランの生成後、初期フロアプランのコストを概算し、SA によってフロアプランを行う。

SA では、 $T_{initial}$ から以下の方法で制御遅延が最適となるフロアプランを探索する。まず、シーケンスペアのうちの 1 つのシーケンス、もしくは 2 つのシーケンスを任意に選択し、任意の 2 つのモジュールの位置を交換する。それによってできる新しいフロアプランのコストが、これまでに探索された中で最適なものより小さい場合、そのフロアプランとその時のコストを最適なものとして保存する。仮にもし、大きくなったとしても、一定の確率で

そのフロアプランを受け入れる。ただしこの場合は、最適なフロアプランの更新は行われず、受け入れたフロアプランに対して次の交換を行うのみである。こうした作業を ite 回繰り返した後、温度 T を cf によってさます。 T が T_{end} に達した時点で探索は完了する。この時点での最適なフロアプランを出力する。

3.5 コスト関数

本稿で用いる回路モデルの制御回路の遅延 d_{ctr} は、以下のとおりである。

$$d_{ctr} = \sum_{i=0}^{m-1} (d_{ctr_i} + \max(d_{hreq_i}, d_{breq_i})) \quad (4)$$

ここで、 d_{ctr_i} は i 番目の制御モジュールの遅延、 d_{hreq_i} と d_{breq_i} はそれぞれ、 i 番目の制御モジュールでホールド時間制約を満たすために必要となる遅延と分岐制約を満たすために必要となる遅延を表す。

d_{ctr_i} は以下のように概算する。仮に ack_{i-1} より Q_i によって制御される REG までのデータパスが j 本あると仮定する。(1) で表されたセットアップ時間制約は j 本すべてで満たす必要がある。もし、1 つでも違反がある場合、セットアップ時間を満たすために必要な遅延 d_{sreq_i} を d_{ctr_i} に含める必要がある。 d_{sreq_i} は以下のように表すことができる。

$$d_{sreq_i} = \max(d_{sdp_{i,0}} + d_{setup} + d_{sm_i} - d_{scp_{i,0}}, \dots, d_{sdp_{i,j-1}} + d_{setup} + d_{sm_i} - d_{scp_{i,j-1}}, 0) \quad (5)$$

$d_{sdp_{i,v}}$ ($0 \leq v \leq j-1$) は、パス $sdp_{i,v}$ にあるモジュールの論理遅延とモジュール間配線遅延の和である。モジュール間配線遅延は、 $d_{wg}(dist)$ を用いて概算する。ここで、 $dist$ はモジュールの中点座標より求める。 $d_{sdp_{i,v}}$ は、Q モジュールの遅延 d_{req+} , d_{req-} , d_{out+} と CTR_{i-1} と CTR_i 間の配線遅延の和である。この配線遅延も $d_{wg}(dist)$ を用いて概算する。なお、全てのパスでセットアップ時間制約を満たす場合、 d_{sreq_i} は 0 となる。最終的に、 d_{ctr_i} は以下ようになる。

$$d_{ctr_i} = \max(d_{scp_{i,0}} + d_{sreq_i}, \dots, d_{scp_{i,j-1}} + d_{sreq_i}) \quad (6)$$

次に、 d_{hreq_i} の概算を解説する。連続する Q モジュール Q_i と Q_{i+1} によってデータが書き込まれる REG が k 個あるとする。この場合、 k 個の REG で (2) で表されたホールド時間制約を満たす必要がある。もし 1 つでも違反がある場合、ホールド時間制約を満たすために必要な遅延 d_{hreq_i} に相当する遅延素子 D_{hb_i} を CTR_i に付加する必要がある。 d_{hreq_i} は以下ようになる。

$$d_{hreq_i} = \max(d_{hcp_{i,0}} + d_{hold} + d_{hm_i} - d_{hdp_{i,0}}, \dots, d_{hcp_{i,k-1}} + d_{hold} + d_{hm_i} - d_{hdp_{i,k-1}}, 0) \quad (7)$$

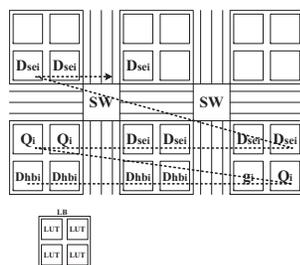


図5 制御モジュール CTR_i の配置.
Fig. 5 Placement of control module CTR_i .

d_{sreq_i} のときと同様, $d_{hcp_{i,u}}$ ($0 \leq u \leq k-1$) と $d_{hdp_{i,u}}$ は, パス $hcp_{i,u}$ と $hdp_{i,u}$ に含まれるモジュールの論理遅延と $d_{wg}(dist)$ によって概算されたモジュール間配線遅延の和である.

d_{breq_i} の概算は以下のとおりである. ある Q モジュール Q_i でレジスタに書き込まれた値を使って, 制御が l つの方向のいずれかに分岐する場合, l つの分岐で分岐判定制約を満たす必要がある. 仮にもし 1 つでも違反があれば, 分岐判定制約を満たすために必要となる遅延 d_{breq_i} に相当する遅延素子 D_{hb_i} を CTR_i に付加する必要がある. d_{breq_i} は以下のようになる.

$$d_{breq_i} = \max(d_{bdp_{i,0}} + d_{bm_i} - d_{bcp_{i,0}}, \dots, d_{bdp_{i,l-1}} + d_{bm_i} - d_{bcp_{i,l-1}}, 0) \quad (8)$$

d_{sreq_i} や d_{hreq_i} を求めたときと同様, $d_{bdp_{i,w}}$ ($0 \leq w \leq l-1$) と $d_{bcp_{i,w}}$ は, パス $bdp_{i,w}$ や $bcp_{i,w}$ に含まれるモジュールの論理遅延と $d_{wg}(dist)$ によって概算したモジュール間配線遅延の和である.

3.6 遅延素子の生成

フロアプラン後, 遅延素子の生成を行う. d_{sreq_i} の値が 0 より大きいとき, セットアップ時間制約を満たすよう遅延素子 D_{se_i} を生成する. 遅延素子は LUT で実現するので, 1LUT 追加あたりの遅延の増加を d_{incs} とすると, 必要となる LUT 数は $\lceil d_{sreq_i} / (2 * d_{incs}) \rceil$ となる. ここで, d_{incs} を $d_{lut} + d_{wl}$ とする. d_{incs} に 2 を掛ける理由は, レジスタにデータを書き込むまでに遅延素子を 2 回通るからである.

同様に, d_{hreq_i} と d_{breq_i} の最大値が 0 より大きいとき, ホールド時間制約と分岐判定制約を満たすために遅延素子 D_{hb_i} を生成する. 1LUT 追加あたりの遅延の増加を d_{inchb} とすると, 必要となる LUT 数は $\lceil \max(d_{hreq_i}, d_{breq_i}) / d_{inchb} \rceil$ となる. ここで, d_{inchb} は $d_{lut} + d_{wl}$ とする.

3.7 配置制約の生成

フロアプランの決定後, ターゲットとなる FPGA のベンダが提供する制約ファイルフォー

表 1 Vertex 4 上での遅延.
Table 1 Delays on Vertex 4.

name	delay (ns)
d_{lut}	0.15
d_{wl}	0.15
d_{wls}	0.21
$d_{wg}(dist)$	$0.04 * dist + 0.39$
d_{req+}	0.43
d_{req-}	0.43
d_{out+}	0.47

表 2 SA のパラメータ.
Table 2 Parameters for SA.

name	value
$T_{initial}$	1
T_{end}	0.001
ite	1000
cf	0.9

表 3 性能評価.

Table 3 Performance evaluation.

回路	モジュール (制御)	SC/HC/BC	レジスタ間遅延最小		提案手法	
			性能 [ns]	時間 [s]	性能 [ns]	時間 [s]
diffeq	37 (8)	25/1/1	48.65	131.55	45.50	216.52
isqrt	40 (9)	30/2/3	52.91	251.18	43.75	443.77
orddif	43 (14)	32/5/0	117.91	223.60	111.47	362.38
ewf	68 (18)	74/8/0	134.87	480.67	120.71	756.47

表 4 面積評価.

Table 4 Area evaluation.

回路	SC/HC/BC	遅延素子 (D_{se_i}/D_{hb_i}) [lut]	全体 [lut]
diffeq	25/1/1	18 (16/2)	568
isqrt	30/2/3	20 (16/4)	361
orddif	32/5/0	79 (79/0)	1342
ewf	74/8/0	58 (58/0)	2245

マットに従って, 配置制約を生成する. なお, 制御モジュール CTR_i は以下のように配置する. 左下より右上に, 始めにホールド時間制約と分岐判定制約を満たすために必要な遅延素子 D_{hb_i} を, 次に分岐判定論理などのグルーロジック, Q モジュール, セットアップ時間を満たすために必要な遅延素子 D_{se_i} の順で配置する (図 5).

4. 実験

実験では, 4 つの回路 diffeq, usqrt, ordif, ewf の論理設計に対して, 提案手法を適用し, 概算性能とフロアプラン時間を評価する. また, 提案手法のようにフロアプラン時に束データ方式の特徴を考えず, レジスタ間遅延の和が最小となるようフロアプランを行い, 得られたフロアプランにたいしてタイミング制約の考慮と遅延素子の生成を行ったもの (レジスタ間遅延最小と呼ぶ) と比較する. 更に, フロアプラン後に挿入された遅延素子の LUT

数が回路全体の LUT 数と比べどの程度になるのかを示す．実験のため提案手法を Java で実装し，Core2Duo プロセッサと 2G のメモリを持った Window マシンで実験を行った．

実験で用いる FPGA は，Xilinx 社の Vertex 4 (xc4vfx25)⁴⁾ である．Vertex 4 では，1 つの論理ブロック内に 4 つのスライスがある．また，1 つのスライスには 2 つの LUT がある．このため実験では，論理ブロック内の配線をスライス間の配線 d_{wls} と LUT 間の配線 d_{wll} に分類する．これらの値は遅延素子の生成時に用いられるが，前節で説明したとおり，制御モジュール CTR に含まれる遅延素子や Q モジュールなどは，規則的に配置されるため，座標に応じて d_{wls} と d_{wll} を使い分ける．また，FPGA の遅延として，データシートと Xilinx ISE に備わっている静的タイミング解析ツールより， d_{lut} ， d_{wls} ， d_{wll} ， d_{req+} ， d_{req-} ， d_{out+} を求める． $d_{wg}(dist)$ は，2 節で述べたように，論理ブロック間のマンハッタン距離 ($dist$) とその時の配線遅延を基に最小二乗法によって生成する．これらを表 1 に示す．

回路の各モジュールを Xilinx ISE11.1⁵⁾ を用いて論理合成し，遅延と面積 (スライス数) を得る．実際の面積に 1.5 倍か 2 倍した値をフロアプランにおける各モジュールの面積とする．これは論理合成以降で行われる最適化や入出力バッファの挿入などで論理合成時に概算された値とずれが生ずる可能性があるからである．なお，モジュールに外部入出力が含まれない場合は 1.5 倍，含まれる場合は 2 倍とする．アスペクト比は必要以上にスライスを使わないようモジュール毎に適宜設定する．次に，制御モジュール毎にセットアップ時間制約，ホールド時間制約，分岐判定制約の確認に必要なパスを設定する．これらの制約に対するマージン (d_{sm_i} ， d_{hm_i} ， d_{bm_i}) は 0 とする．

SA に対するパラメータの設定は，表 2 に示すとおりである．この場合，一度 SA を実行すると，約 66,000 回任意のモジュールのペアの配置が交換される．しかしながら，これは探索空間のほんの一部にすぎない．そのため，ここでは回路毎に 10 回 SA を実行したときの平均値で性能評価を行う．

性能評価を表 3 に示す．表 3 のうち，モジュールはレジスタ，マルチプレクサ，演算器，制御モジュールの総数を表わす．括弧の値は制御モジュール数である．SC，HC，BC はそれぞれ，回路全体におけるセットアップ時間制約，ホールド時間制約，分岐判定制約の個数を表わす．性能は回路の概算性能，時間はフロアプラン時間を表わす．レジスタ間遅延最小は，入力から出力までのレジスタ間最大遅延の総和を最小化することを目的にフロアプランを行った結果を表わし，提案手法は提案手法によるフロアプランの結果を表わす．

表 3 より，提案手法を用いた方が性能のよい回路をえることができる．その差は，制御モジュールの数とホールド時間制約と分岐判定制約を満たすために必要となる遅延 D_{hb_i} の挿入に依存する．前者は orddif と ewf より伺える．これらは共に D_{hb_i} が挿入されず，制御モジュールの多い ewf のほうが，差が大きい．後者は diffeq や isqrt と orddif より伺える．diffeq や isqrt では D_{hb_i} が挿入されるため，挿入のない orddif より差が大きい．フロアプラン時間に関しては，ベンチマークによるが約 1.6 倍から 1.8 倍の時間がかかる．これはコ

スト関数の計算の際，タイミング制約を満たすかを確認するためである．

次に，フロアプラン後に挿入された遅延素子の回路全体の面積に対する影響を表 4 に示す．この評価は回路毎に SA を 10 回実行した中で，性能が最適なものに対して行う．遅延素子は遅延素子に必要となる LUT 数を表し，括弧はそのうち D_{se_i} に必要となる LUT 数と D_{hb_i} に必要となる LUT 数を表す．全体は，回路全体で必要となる LUT 数を表し，ISE を用いて配置配線が終わったときに得られた値を表す．

表 4 より，遅延素子に必要となる LUT 数は，回路全体の数パーセント程度で済むことがわかる．これは，フロアプランの段階でタイミング制約を考慮した上で制御回路の遅延を最小化したことによる．特に，ホールド時間制約はフロアプランの段階で満たさ，分岐判定制約のある diffeq と isqrt に関してのみ D_{hb_i} が挿入された．

5. 結 論

本稿では，FPGA 実装を対象とした束データ方式による非同期式回路のフロアプラン手法を提案した．提案手法は，束データ方式で必要となるタイミング制約を考慮しながら，制御回路の遅延が最小となるようにフロアプランを行う．実験の結果より，フロアプランのときに束データ方式の特徴を考慮しなかったものより性能の良い回路を得ることができることを示した．今後は，規模の大きな回路への適用，配置配線後の回路との比較，パイプライン回路を扱うための拡張を行うつもりである．

謝 辞

本研究は科学技術振興事業団 (JST) の戦略的基礎研究推進事業 (CREST) の助成を受けたのもである．

参 考 文 献

- 1) H.Murata, K.Fujiyoshi, S.Nakatake, and Y.Kajitani, "VLSI Module Placement Based on Rectangle-Packing by the Sequence-Pair", *IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems*, 15(12):1518-1524, 1996.
- 2) F.U.Rosenberger, C.E.Molnar, T.J.Chaney, and T-P.Fang, "Q-Modules: Internally Clocked Delay-Insensitive Modules", *IEEE Transaction of Computer*, vol.C-37, no.9, pp.1005-1018, 1988.
- 3) N.Sherwani, "Algorithms for VLSI Physical Design Automation", KAP, 1995.
- 4) XILINX Inc, "Vertex-4 FPGA", <http://www.xilinx.com>
- 5) XILINX Inc, "ISE Foundation 11.1i", <http://www.xilinx.com>