

遅延ばらつきを考慮したデータパス合成に関する基礎的考察

井上 恵介[†] 金子 峰雄[†] 岩垣 剛[†]

[†] 北陸先端科学技術大学院大学 情報科学研究科 〒923-1292 石川県能美市旭台 1-1

E-mail: †{k-inoue,mkaneko,iwagaki}@jaist.ac.jp

あらまし 半導体プロセスの微細化に伴い、寄生素子の影響増大、電圧の微弱化が進み、雑音や遅延ばらつきの問題が顕在化してきている。本研究では、遅延ばらつきの下で正しく動作する回路を、有限な任意の遅延ばらつきの下で正しく動作するクロック周期が存在する回路と捉え、そのレジスタ転送レベルの設計条件を明らかにする。また、これに基づいて遅延ばらつきを考慮したレジスタ数最小を保証するレジスタ割り当て手法を示す。

キーワード 高位合成, 遅延変動, レジスタ割り当て

A Basic Study on Datapath Synthesis Considering Delay Variation

Keisuke INOUE[†], Mineo KANEKO[†], and Tsuyoshi IWAGAKI[†]

[†] School of Information Science, Japan Advanced Institute of Science and Technology

1-1 Asahidai, Nomi, Ishikawa 923-1292, Japan

E-mail: †{k-inoue,mkaneko,iwagaki}@jaist.ac.jp

Abstract With the advance of process technologies, the device feature size of a VLSI becomes smaller. As a result, its switching delay is made smaller and its operation speed becomes higher. On the other hand, delay variation, which is due to process variation, temperature variation, power supply noise, coupling noise, etc., becomes relatively larger, and it causes a serious problem. This is because it may force a VLSI to violate the timing constraint. This paper discusses a datapath with a certain class of the robustness to the delay variation. This robustness guarantees that, under any random delay variations, there always exists a clock frequency with which the datapath operates correctly. In this paper, as a first step, we focus on register assignment in designing such a datapath. From a practical point of view, it is important to consider optimizing a datapath in terms of clock frequency, floorplan, power dissipation, etc. However, it is still an open problem, and it will be tackled in future.

Key words High-Level Synthesis, Delay Variation, Register Allocation, Register Assignment

1. ま え が き

半導体プロセスの微細化に伴い、トランジスタや配線などの形状のばらつきが顕著になってきている。その結果、回路を伝搬する信号の遅延ばらつきが増大している。また、動作時の変動(電源ノイズやクロストークなど)によって生じる遅延の動的変動の問題も増大している。遅延ばらつきへの対応策として現在では主に、ばらつきを見越したマージンをとる設計を行っている。しかし、微細化とともに遅延に対するマージンが相対的に大きくなってきている。結果としてマージンをとることが難しく、仕様通りに動作しない回路が今後増大していくことが懸念される。

この問題に対して、これまで様々な研究が行われている。マージンの幅を抑えるために、ばらつきの正確な見積りを目指す研究が行われている[1][2]。一方、回路の設計段階において対

策を講ずる必要性も高まっている。このような研究として、遅延制御素子を使用する研究が行われている[3]。

本研究では、遅延ばらつきの下で正しく動作する回路を、有限な任意の遅延ばらつきの下で正しく動作するクロック周期が存在する回路と捉え、そのレジスタ転送レベルの設計条件を明らかにする。これに基づいて、入力データフローグラフが無閉路有向グラフのとき、遅延ばらつきを考慮したレジスタ数最小を保証するレジスタ割り当て手法を示す。入力データフローグラフが有向閉路を含むとき(ループ計算のパイプライン実行)、レジスタ数最小化問題がNP困難であることを示し、ヒューリスティックによる解法を示す。

本稿は以下のように構成される。第2章で遅延ばらつきを考慮した設計問題について述べる。第3章で構造的遅延変動耐性を持つためのレジスタ割り当てアルゴリズムについて述べる。第4章でループパイプライン実行における構造的遅延変動耐性

を持つためのレジスタ割り当てアルゴリズムについて述べる。最後に第5章でまとめる。

2. 遅延ばらつきを考慮した設計の問題

2.1 回路のRTL記述

本研究では回路の動作記述としてデータフローグラフを入力とし、レジスタ転送レベル(RTL: Register Transfer Level)の回路を出力する問題を扱う。回路のRTL記述とは、回路の動作をレジスタやメモリ間でのデータ転送と演算器による演算で記述したものである[4]。

2.2 セットアップ条件とホールド条件

図1の回路において、演算 O_a の結果 a をレジスタ Reg 1 に書き込み、そのデータを用いて演算 O_b (演算 O_b は演算器 FU_A に割り当てられているとする) を行い、結果 b をレジスタ Reg 2 に書き込む場合のタイミングを説明する。ここで、クロック周期を t_c 、レジスタ間の最大遅延を d_{max} 、レジスタ間の最小遅延を d_{min} 、演算 O_i の出力結果をレジスタに書くクロックに同期した離散的なタイミング(以下、単にタイミング)を $\sigma(i) \in \mathbb{Z}$ とする。なお、本稿では意図的なタイミングスキュー導入はないものとする。

Reg 2 に書き込み信号が到着する時刻は演算 O_a の結果が Reg 1 に書き込まれた時刻から d_{max} 経過した時刻以降でなければならない。これをセットアップ条件という。

$$\sigma(a) \cdot t_c + d_{max} \leq \sigma(b) \cdot t_c \dots \text{セットアップ条件}$$

一般にレジスタは複数のデータによって共有される。Reg 1 に、 O_a の結果が書き込まれた以降に演算 $O_{a'}$ の出力データ a' が書き込まれる場合、Reg 2 に書き込み信号が到着する時刻はこの a' の書き込み時刻から d_{min} 経過した時刻より早くなければならない。これをホールド条件という。

$$\sigma(b) \cdot t_c < \sigma(a') \cdot t_c + d_{min} \dots \text{ホールド条件}$$

セットアップ条件とホールド条件の二つの制約条件を全ての演算について満たす回路が正しく動作する回路である。

遅延ばらつきの下で正しく動作する回路は、遅延ばらつきを考慮したセットアップ条件とホールド条件を全ての演算について満たす回路である。なお、本稿ではクロック信号のレジスタへの到着ばらつきとデータバス上の信号の遅延ばらつきを扱う。データの書き込みタイミング $\sigma(i)$ に対してその到着時刻変動量を $\Delta_{i_{min}}, \Delta_{i_{max}}$ とする。ただし、 $\Delta_{i_{min}}, \Delta_{i_{max}}$ は一つのレジスタにおけるビット毎のばらつき中の最も早く到着するものの変動量と最も遅く到着するものの変動量をそれぞれ表すものとする。一方、 d_{max}, d_{min} の遅延変動値をそれぞれ $\Delta_{d_{max}}, \Delta_{d_{min}}$ とする。遅延ばらつきの下でのセットアップ条件とホールド条件は以下のように書くことができる。

$$\sigma(a) \cdot t_c + \Delta_{a_{max}} + d_{max} + \Delta_{d_{max}} \leq \sigma(b) \cdot t_c + \Delta_{b_{min}}$$

$$\sigma(b) \cdot t_c + \Delta_{b_{max}} < \sigma(a') \cdot t_c + \Delta_{a'_{min}} + d_{min} + \Delta_{d_{min}}$$

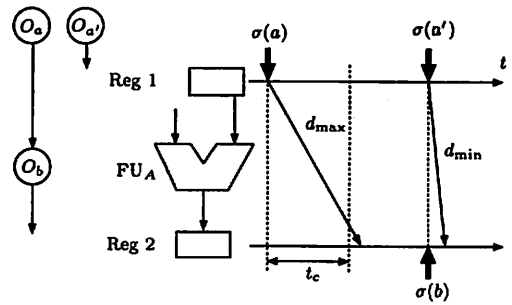


図1 セットアップ条件とホールド条件

2.3 構造的遅延変動耐性(タイプ I)

正しく動作する回路は、ある指定された遅延ばらつきモデルの下で全ての演算についてセットアップ条件とホールド条件を満足することが要求される。一方で、上位階層での設計において、遅延のばらつき、動的変動を正確に見積もるのは非常に難しいと考えられる。ここで我々は回路が機能として正しく動くこととクロック周波数に代表される性能を一旦切り離して考えることにする。すなわち、機能として正しく動くことを、「構造的遅延変動耐性」を持つと言い、次のように定義する。

構造的遅延変動耐性：

任意の(あるいは、ある制約を受けた)有限な遅延ばらつき・変動に対してセットアップ条件、ホールド条件を満足するクロック周波数が存在する(ような回路)。 □

セットアップ条件より、

$$\Delta_{a_{max}} + d_{max} + \Delta_{d_{max}} - \Delta_{b_{min}} \leq (\sigma(b) - \sigma(a)) \cdot t_c$$

が成り立つ。任意の有限な遅延ばらつきに対してこのセットアップ条件を満たすクロック周期が存在するための必要十分条件は、

$$\text{C1: 演算 } O_a \text{ の結果 } a \text{ を使って演算 } O_b \text{ が結果 } b \text{ を生成するとき, } \sigma(b) - \sigma(a) \geq 1$$

である。これは演算 O_b の結果を書き込むタイミングが演算 O_a を書き込むタイミングより1ステップ以上遅くなければならないことを意味している。

一方、ホールド条件は、

$$(\sigma(b) - \sigma(a')) \cdot t_c < d_{min} + \Delta_{d_{min}} + \Delta_{a'_{min}} - \Delta_{b_{max}}$$

と書ける。任意の有限な遅延ばらつきに対してホールド条件を満たすクロック周期が存在するための条件は、

$$\text{C2: } b \text{ を生成する演算 } O_b \text{ の入力データ } a' \text{ によって上書きされるとき, } \sigma(b) - \sigma(a') \leq -1$$

となる。C1, C2 を満たす回路を構造的遅延変動耐性(タイプ I)を持つ(回路)と呼ぶことにする。

2.4 構造的遅延変動耐性(タイプ II)

一つのレジスタにおける書き込み信号到着のビット毎のばらつきが十分に小さいとき、ホールド条件を満たすクロック周期が存在するための条件は次のように書き換えられる。

C2'-1: $a' \neq b$ の場合 (入力データ a を保持しているレジスタに出力データ以外のデータ a' を書き込む場合) $\sigma(b) - \sigma(a') \leq -1$

C2'-2: $a' = b$ の場合 入力データ a を保持しているレジスタに出力データ b を書き戻す場合) b の書き込みのための制御信号と a' の書き込み制御信号は同じものであり, $\sigma(b) - \sigma(a') = 0$

C2'-2 は $\Delta_{a',min} - \Delta_{b,max} \approx 0$ を前提としたものであり, $\sigma(b) - \sigma(a') = 0$ の下での詳細なホールド条件 $0 < d_{min} + \Delta_{d,min} + \Delta_{a',min} - \Delta_{b,max}$ が満たされていることを仮定している. C1, C2'-1, C2'-2 を満たす回路を構造的遅延変動耐性 (タイプ II) を持つ回路と呼ぶことにする.

2.5 順序制約付構造的遅延変動耐性

ホールド条件

$$(\sigma(b) - \sigma(a')) \cdot t_c < d_{min} + \Delta_{d,min} + \Delta_{a',min} - \Delta_{b,max}$$

において, もし $\Delta_{a',min} > \Delta_{b,max}$ なる関係があれば, $d_{max} + \Delta_{d,min} + \Delta_{a',min} - \Delta_{b,max} > 0$ が保証され, ホールド条件を満たすクロック周期が存在する条件を

$$\sigma(b) - \sigma(a') \leq 0$$

と緩和することができる. すなわち, 先節 C2'-1 に代わり,

C2'-1': $a' \neq b$ について, $\Delta_{a',min} > \Delta_{b,max}$ かつ $\sigma(b) - \sigma(a') \leq 0$ または, $\sigma(b) - \sigma(a') \leq -1$

とすることができる. C1, C2'-1', C2'-2 を満たす回路を順序制約付構造的遅延変動耐性を持つ回路と呼ぶことにする.

3. 構造的遅延変動耐性 (タイプ II) を持つためのレジスタ割り当てアルゴリズム

ここでは, 入力データフローグラフを無閉路有向グラフとし, 構造的遅延変動耐性 (タイプ II) を持つレジスタ割り当てについて議論する.

3.1 従来のレジスタ割り当てアルゴリズム

データが生成される時刻から, そのデータが最後に使用される時刻までの時間的な区間のことをデータのライフタイムと呼ぶ. 従来のレジスタ割り当て手法におけるデータのレジスタ共有の条件はライフタイムの重なりがないことである. 具体的なレジスタ割り当て手法としてレジスタ数最小を保証するレフトエッジ法が知られている [5].

3.2 レジスタ共有の条件

データのレジスタ共有のためには, C2'-1, C2'-2 の条件を満足する必要がある.

C2'-1: $a' \neq b$ の場合 $\sigma(b) - \sigma(a') \leq 1$ が成り立たなければならない. すなわち, b の入力データ a と他のデータ a' がレジスタ共有できるのは $\sigma(a')$ が $\sigma(b)$ より 1 ステップ以上離れている場合に限られる (図 2).

C2'-2: $a' = b$ の場合 入力として使ったデータ a が保持され

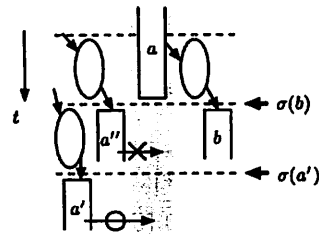


図 2 ライフタイム終了直後のステップでのデータの書き込みの可能性 (C2'-1). 矩形はデータのライフタイム, だ円は演算, 矢印はデータ依存関係をそれぞれ表す. 網囲みはデータ a が割り当てられているレジスタを表す.

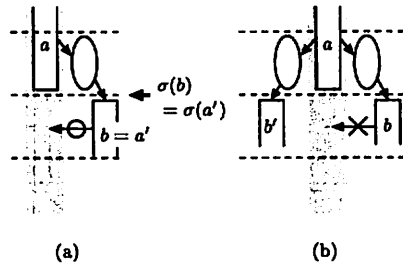


図 3 ライフタイム終了直後のステップでのデータ書き込みの可能性 (C2'-2). (a) a と b はレジスタ共有可能. (b) a と b はレジスタ共有不可能.

ているレジスタに, 出力結果 b を書き込むことは許される (図 3-a). ただし, 同じ入力データ a を使う演算が他にもあって, b の書き込みと同じタイミングでその出力結果 b' を書き込むとき, b は a とレジスタを共有できない (図 3-b). これは b' にとっての b が図 2 における b にとっての a' に相当するからである.

3.3 レジスタ割り当てアルゴリズム

以下に, 条件 C2'-1, C2'-2 を考慮したレジスタ割り当てアルゴリズムを示す.

- Step 1 データのレジスタ共有の条件の中で C2'-2 の条件を満たし, 前のデータの直後に書き込めるデータを, 前のデータと同一のレジスタを使うものと定め, これら二つのライフタイムをマージして一つのライフタイムとする.
- Step 2 マージしてできたライフタイム, およびマージされずに残っているライフタイムのすべてについて最終ステップに 1 ステップ加えたものを新たにライフタイムとする.
- Step 3 Step 2 で得られたライフタイムの集合に対して, レフトエッジ法を適用する.

図 4 は小さな入力を例にとって, 各ステップの具体例を示している. 図 5, 図 6 は入力インスタンスとして, fifth-order wave digital elliptic filter ベンチマーク [4] を用いた場合の従来のレジスタ割り当て例と構造的遅延変動耐性 (タイプ II) のためのレジスタ割り当て例を示している. なお, 入力を非巡回有向グラ

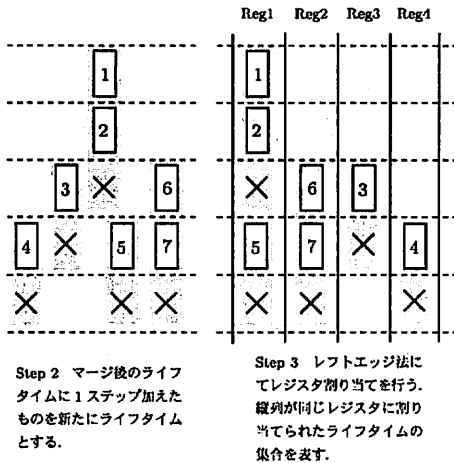
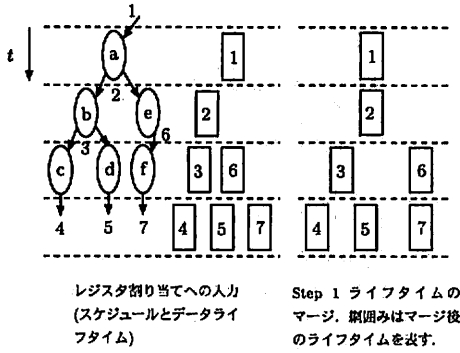


図4 レジスタ割り当て手法の具体例

フとするために遅延枝を切り開いてシステムの入力と出力に置き換えている。ハードウェア資源として加算器4個、乗算器2個を使用し、各演算のステップ数を1とした。図中の横軸はレジスタ番号を表し、縦軸は制御ステップ番号を表す。太い縦線分は、個々のデータのライフタイムを表し、ライフタイムのかたわらにある記号はそのライフタイムのデータを出力した演算の識別記号である。図6に記した×記号は、他のデータに割り当てられない箇所を表している。識別記号に下線を引いたデータは外部入力、枠囲みをしたデータは外部出力を表す。

3.4 最適性に関する定理

以下の議論において、レジスタ割り当てをデータの集合 D から正整数値で表現されたレジスタ集合への関数 A と定義する(以下、この値をレジスタ番号と呼ぶ)。このとき、提案アルゴリズムの Step 3 で行われるレフトエッジ法を用いたレジスタ割り当てはレジスタ番号が小さい順に行われていくものとする。また、データ d のライフタイムの開始時刻を $l_s(d)$ 、終了時刻を $l_e(d)$ とし、 $\mathcal{R}_A(d)$ をレジスタ割り当て A において d と同じレジスタに割り当てられ、開始時刻が $l_s(d)$ 以降であるデータの集合とする (d 自身も $\mathcal{R}_A(d)$ の要素である)。提案アルゴリズムに関して以下の定理が成り立つ。

定理1 提案アルゴリズムは、構造的遅延変動耐性(タイプII)

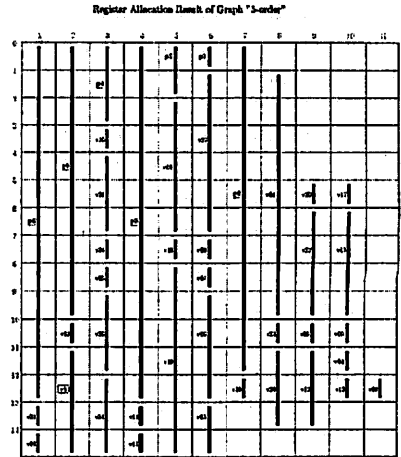


図5 従来のレジスタ割り当て手法による結果

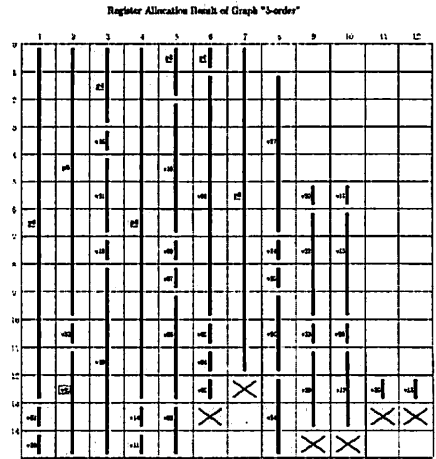


図6 遅延ばらつき考慮したレジスタ割り当て結果

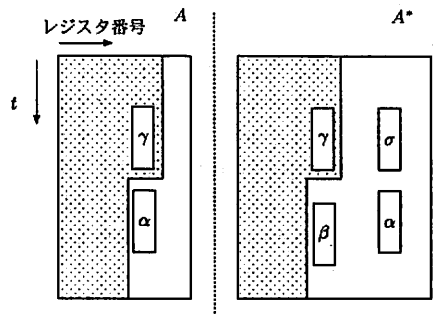


図7 A と A^* によるレジスタ割り当て。網囲みは共通部分。α, β, σ, γ はデータのライフタイム

のためのレジスタ割り当てにおいて、レジスタ数最小の解を生成する。

証明: 提案アルゴリズムで得られるレジスタ割り当てを A とする。一方、構造的遅延変動耐性(タイプII)のためのレジスタ

割り当てで、レジスタ数最小となるものを A^* とする。レジスタ数最小となる割り当てが複数存在する場合は、 A と A^* で割り当てられたデータをレジスタ番号の小さい方からライフタイムの開始時刻が早い順に比較していくとき最も多く共通するものを A^* とする。また、このとき最後に共通したデータを γ とする。最初に異なるデータのうち、 A で割り当てられたものを α 、 A^* で割り当てられたものを β とする。 A^* において、 α の直前に割り当てられているデータを σ とする (図 7)。このとき、両者の関係を三つ組 (x, y, z) で表現することができる。 x, y, z はそれぞれ 0 または 1 をとる変数であり、その値は以下のように決定されるものとする。

- x : A において $l_e(\gamma) = l_s(\alpha)$ のとき 0, そうでないとき 1
 - y : A^* において $l_e(\gamma) = l_s(\beta)$ のとき 0, そうでないとき 1
 - z : A^* において $l_e(\sigma) = l_s(\alpha)$ のとき 0, そうでないとき 1
- (x, y, z) の各値について場合分けをして証明する。

$(x, y, z) = (0, 0, 0), (0, 0, 1)$ の場合: $l_e(\gamma) = l_s(\alpha) = l_s(\beta)$ であるが C2'-2 より、 γ は α, β とマージ不可能である。従って、このような状況は存在しない。

$(x, y, z) = (0, 1, 0), (0, 1, 1), (1, 1, 1)$ の場合: A^* において、 $\mathcal{R}(\alpha)$ の各データのレジスタ番号と $\mathcal{R}(\gamma)$ の各データのレジスタ番号を入れ替える。入れ替えた後のレジスタ割り当てを A^{**} とする。

$(x, y, z) = (1, 0, 0), (1, 1, 0)$ の場合: σ と α はマージ可能である。 A においてマージ候補が存在するにも関わらず α がどのデータともマージしないのは提案アルゴリズムに反する。従って、このような状況は存在しない。

$(x, y, z) = (1, 0, 1)$ の場合: β を出力する演算の入力データが 1 つの場合、この状況は存在しない。 β を出力する演算の入力データが複数存在する場合、提案アルゴリズムにおいて γ と β がマージされなかったのは、 β とマージ可能なデータ c が他にあって ($l_e(c) = l_s(\beta)$)、 c とマージされたからである。 $c = \sigma$ の場合、 A^* において、 $\mathcal{R}(\beta)$ の各データのレジスタ番号と $\mathcal{R}(\gamma)$ の各データのレジスタ番号を入れ替える。入れ替えた後のレジスタ割り当てを A^{**} とする。 $c \neq \sigma$ の場合、 c の直後に割り当てられたデータを e とする。 A^* において、 $\mathcal{R}(\gamma)$ の各データのレジスタ番号と $\mathcal{R}(e)$ の各データのレジスタ番号を入れ替える。さらに、 $\mathcal{R}(e)$ の各データのレジスタ番号と $\mathcal{R}(\alpha)$ の各データのレジスタ番号を入れ替える。入れ替えた後のレジスタ割り当てを A^{**} とする。

A^{**} はレジスタ数最小のレジスタ割り当てであり、 A^* より A と一つ多くの共通データを持つ割り当てである。これは A^* の選び方に反する。

4. ループパイプライン実行における構造的遅延変動耐性 (タイプ II) のためのレジスタ割り当てアルゴリズム

4.1 実行可能解が存在するための条件

ループパイプライン実行では開始時刻と終了時刻が繰り返し周期についての剰余として一致するライフタイムを持つデータ (このようなデータを円状データと呼ぶことにする) が存在する

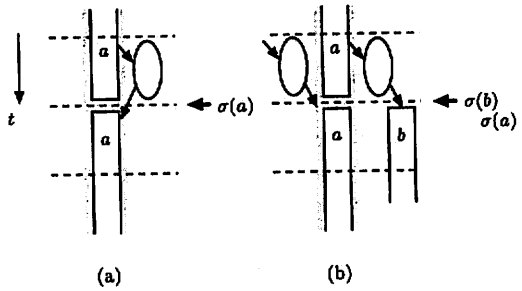


図 8 a が存在する場合の実行可能解の存在性。(a) 実行可能な例、(b) 実行不可能な例。

可能性がある。この場合、必ずしも全てのスケジュールに対して構造的遅延変動耐性 (タイプ II) を持つレジスタ割り当てが存在するとは限らない。

円状データ (データ名 a) が存在する場合: 入力として使ったデータ a が保持されているレジスタに出力結果 a を書き込むことは許される (図 8-a)。ただし、 $\sigma(a)$ と同じタイミングで a 以外のデータを出力する場合、 a は自分自身とレジスタと共有できない (図 8-b)。それは a にとっての自分自身が図 2 における b にとっての a'' に相当するからである。これは、このようなレジスタ割り当てが実行不可能であることを意味している。

円状データが存在しない場合: この場合は実行可能解が存在する。

4.2 レジスタ数最小化問題の NP 困難性

ここでは、ループパイプライン実行における構造的遅延変動耐性 (タイプ II) のためのレジスタ数最小割り当て問題が NP 困難であることを示す。最適化問題に代わりループパイプライン実行の計算アルゴリズムとそのスケジュールおよび整数 K が与えられたとき、 K 個以下のレジスタで割り当てるレジスタ割り当てが存在するかという判定問題を考える。循環区間グラフの彩色問題が NP 完全なクラスに属することが証明されており [6]、この問題から本問題への多項式時間還元が存在を示すことで証明する。循環区間グラフとは、グラフの各頂点がある円の各円弧に一対一対応し、円弧 I_i と I_j が重なるときかつそのときのみ対応する頂点 v_i と v_j に辺を結ぶことで得られるグラフのことである。循環区間グラフに対応する円弧の集合を区間表現という (図 9-a)。グラフ G の彩色問題とは、 G とある正の整数 K が与えられたときに、枝で結ばれた頂点同士は同じ色で塗らないという条件の下で、 G の全ての頂点を K 色以下で彩色できるかどうかを判定する問題である。

定理 2 ループパイプライン実行における構造的遅延変動耐性 (タイプ II) のためのレジスタ数最小割り当て問題は NP 困難である。

証明: 循環区間グラフ $C = (V, E)$, $V = \{v_1, v_2, \dots, v_n\}$ に対応する区間表現を $\mathcal{I} = \{I_1, I_2, \dots, I_n\}$ とする (I_i は v_i に対応する区間)。時計回り方向に時間軸を定義することで \mathcal{I} はループパイプライン実行におけるデータの集合 $\mathcal{D} = \{d_i | i = 1, 2, \dots, n\}$

と見なすことができる (d_i は I_i に対応している). また, 端点から端点までの極小領域を 1 ステップ間隔とする (図 9-b). 複数の区間の端点が一致する場合 (この場合もこれらの区間に重なりがあると見なす) は, 区間の重なりが 1 ステップ間隔以上となるような区間表現に改める.

C を循環区間グラフの集合, \mathcal{L} を実行可能解を持つ条件を満たすループパイプラインスケジュールの集合として, 関数 $f: C \rightarrow \mathcal{L}$ を以下のように定義する.

各 d_i の終了時刻と, 開始時刻および終了時刻が繰り返し周期についての剰余として等しい円状データ a_i を追加する. d_i と a_i を入力データとし, a_i を出力データとするステップ数 1 の演算 O_i を定義する ($\sigma(O_i) = \sigma(a_i)$). 次に, 入力データを a_i とし出力データを d_i とするステップ数 1 の演算 O'_i を定義する ($\sigma(O'_i) = \sigma(d_i)$). このようにして定義された演算 O_i, O'_i の集合とその依存関係および各演算の結果の書き込み時刻は, ループパイプライン実行における実行可能解を持つための条件を満たすスケジュール $r \in \mathcal{L}$ となる. 図 9-c に例を示した. r の構成は計算複雑度 $O(n)$ で可能である.

レジスタ割り当てにおいて, 円状データは必ず一つのレジスタを使用し, 他のデータとレジスタを共有しないことに注意する. r がレジスタ数 $K+n$ 個以下でレジスタ割り当てが可能であるとき, 円状データを除くデータは K 個以下のレジスタで割り当てが可能である. 元の循環区間グラフに対して, 異なるレジスタに割り当てられたデータに対応する頂点を異なる色にし, 同じレジスタに割り当てられたデータに対応する頂点が同じ色になるように頂点を彩色すると, K 色以下で可能である. 逆に, 循環区間グラフ $c \in C$ が K 色以下で彩色可能であるとき, $f(c)$ において, 同じ色に塗られた頂点に対応するデータを同じレジスタを使用することによりレジスタ数 K 個以下で割り当て可能である. 従って f は多項式時間還元であることが示され題意が示された. ■

4.3 ヒューリスティックによるレジスタ割り当てアルゴリズム

定理 2 により, ここではヒューリスティックを用いたレジスタ割り当て手法について述べる. 入力スケジュールに対して, 頂点集合が各データに対応し, データがレジスタを共有ができないとき, かつそのときのみデータ間に枝を張るグラフを考える (このようなグラフを競合グラフという). このグラフに対し頂点彩色を行い, 同じ色で塗られた頂点に対応するデータの集合を同じレジスタに割り当てるものとする. 彩色問題は NP 完全に属する問題でありヒューリスティックによる解法が知られている [7].

5. まとめ

遅延ばらつきの下で正しく動作するクロック周期が存在するデータバス回路について検討し, 回路がこのような性質を有するための十分条件を明らかにした. また, 構造的遅延変動耐性を持つためのレジスタ数最小となるレジスタ割り当てアルゴリズムを提案した. 今後の課題として, 順序制約付構造的遅延変動耐性に対するレジスタ割り当て (レジスタの遅延順序決定を

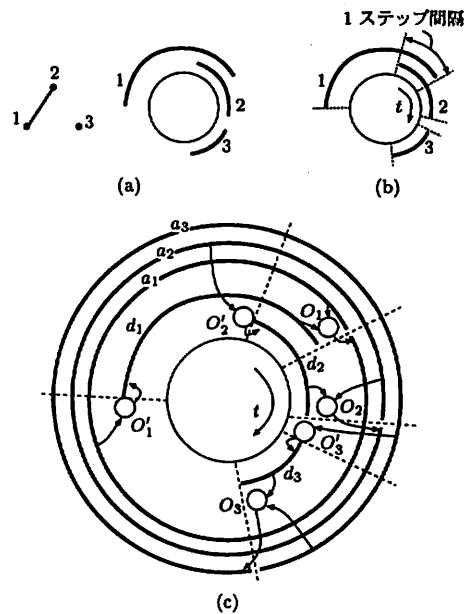


図 9 循環区間グラフのインスタンスと問題の変換. (a) 循環区間グラフの例とその区間表現. (b) 1 ステップ間隔の説明. 破線間は 1 ステップ間隔を表す. (c) レジスタ割り当て判定問題への入力インスタンスへの変換. 太線分は区間 (データのライフタイム) を表し, 円は演算を表す. 矢印は演算と出力データの依存関係を表す.

含む) 手法. 遅延変動が存在する下で正しく動作する RTL 回路 (スケジュール, 資源割り当て, フロアプラン) の特性と性能の最適化がある.

文 献

- [1] M. Conti, G.-F. D. Betta, S. Orcioni, G. Soncini, C. Turchetti, and N. Zorti, "Test structure for mismatch characterization of MOS transistors in subthreshold regime", *Proc. of IEEE International Conference on Microelectronic Test Structures*, pp. 173-178, vol. 10.
- [2] S. Matsumoto, H. J. Mattausch, S. Ooshiro, Y. Tatsumi, M. Miura-Mattausch, S. Kumashiro, T. Yamaguchi, K. Yamashita, and N. Nakayama, "Test-Circuit-Based Extraction of Inter- and Intra-Chip MOSFET-Performance Variations for Analog-Design Reliability", *Proc. of IEEE Custom Integrated Circuits Conference*, pp. 582-585, 2001. 1997.
- [3] M. Murakawa, E. Takahashi, T. Susa, and T. Higuchi, "Post-Fabrication Clock Timing Adjustment for Digital LSI's with Generic Algorithms Ensuring Timing Margins", *Report of MIRAI Project*, 2004.
- [4] P. Michel, U. Lauther, and P. Duzy, "The Synthesis Approach to Digital System Design", *Kluwer Academic Publishers*, 1992.
- [5] F. J. Kurdahi, and A. C. Parker, "Real: A Program for Register Allocation", *Proc. of the 24th Design Automation Conference*, pp. 210-215, 1987.
- [6] D. L. Springer, and D. E. Thomas, "Exploiting the Special Structure of Conflict and Compatibility Graphs in High-Level Synthesis", *Proc. of ICCAD*, pp. 254-257, 1990.
- [7] F. Chow, and J. Hennessy, "The Priority-based Coloring Approach to Register Allocation", *ACM TOPLAS*, 12(4), pp. 501-536, 1990.