

4N-1

データ転送路の自動割付

関川 浩 名古屋 彰 小栗 清 中村 行宏
NTT情報通信処理研究所

1. はじめに

アルゴリズムレベルの動作記述からハードウェアを自動合成する場合、演算器やレジスタ等のリソースの割付と、リソース間の接続構成の最適化が重要となる。リソースの割付の最適化については、いくつかの手法が提案されている[1]-[4]。LSIのチップ上でセクタやバスの占める面積を考慮すれば、リソース間のデータ転送路の削減についても同時に重視する必要がある。

そこで、本稿では、与えられたリソース割付情報から、データ転送路割付の準最適解を自動的に求めるアルゴリズムを提案する。

データ転送路の最適化は、時分割的な共用化も図り得るという点で、通常の論理関数の最適化とは異なるアプローチをとる必要がある。

2. 問題の設定と解決の方針

本稿では、セクタの総入力線数の削減を目標としてセクタの割付を行う。具体的には、「どの転送元からどの転送先へデータ転送があるか」(空間的な情報)、「そのうちで同時に起こるものは何か」(時間的な情報)というリソース間の転送情報が与えられているものとし、できる限り転送路を共用できるようにセクタを自動割付する。その際、データ転送の遅延時間の増大と問題の複雑化を防ぐため、セクタの使用は高々二段に制限する。さらに、問題の簡単化のため、セクタの割付を次のように二段階に分けて行うことにする。

第一段階は、各データ転送路に高々一個のセクタしか入れない場合、すなわち、セクタを共用し得る転送先のグループ分けを行う段階である。第一段階で他の転送先とセクタの共用ができなかった転送先についてのみ、第二段階での、データ転送路に二個のセクタを入れてよい場合を考える。そして、第一段階での最小値および最小値を与えるグループ分けを求め、しかる後に、そのグループ分けに対して第二段階での最小値を求めることを目標にする。各段階ともヒューリスティックなアルゴリズムを適用することにする。

3. アルゴリズム

3.1 第一段階

第一段階では、セクタを共用し得る転送先のグループ分けを求める。そのため、グループ分けを木に配置し、以下の手順によりその一部を探索する。木の構造を図1に示す。

- (1)他の転送先とセクタを共用しにくい順に転送先に順序をつける。
- (2)根の位置にセクタの共用がまったくないグループ分けを置く。その際、グループに番号をつけて、 i 番目のグループに i 番目の転送先を入れるものとする。
- (3)一般に、各節の子には、その節に隣接するグループ分けのうち、セクタの総入力線数が減少しているもののみを置く。ただし、二つのグループ分けが隣接して

いるとは、一方のグループ分けにおいて、ある一つの転送先をそれが属するグループから他のグループに移したものが、グループの順番まで込めてもう一方のグループ分けになっている場合をいうものとする。その際、グループの数をそろえるため、空なグループも残しておく。

- (4)根を最小元とし、各節には根からその節にいたるまでの転送先の移動の履歴から決まる順序を入れ、その順序にしたがって木を探索する。ただし、各レベルごとに今まで探索した範囲での最小値を記憶しておき、もし、ある節のセクタの総入力線数がそのレベルの最小値以上であれば、その節より下の探索は打ち切る。

以上の手順により求められたグループ分けを第一段階での準最適解として採用する。なお、上記の手順(3)により、この木にはすべてのグループ分けが含まれているとは限らない。しかし、セクタの総入力線数の最小値を与えるグループ分けが少なくとも一つ含まれていることが示される。

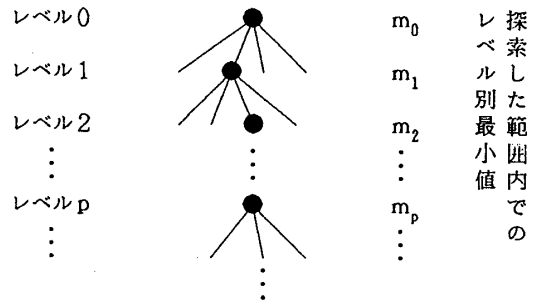


図1 木の構造

3.2 第二段階

第二段階では、第一段階でセクタの共用ができずに単独で残ってしまった転送先へのデータ転送路のみを対象とする。

データ転送路にセクタを二段入れる場合、セクタの総入力線数が減少するのは、転送元に近いセクタが次のいずれかの条件を満たす場合に限られる。

- (1)入力線数が2以上で出力線数が3以上。
- (2)入力線数が3以上で出力線数が2以上。

そこで、次の条件を満たす転送先の集合を作っておく。

- 条件1 転送先3個以上からなる集合でそのすべての転送先にデータ転送がある転送元は2個以上。
条件2 転送先2個からなる集合でその両方の転送先にデータ転送のある転送元は3個以上。

転送元には、適当に順序をつけておく。第二段階では条件1、2を満たす転送先の集合に対し、そのすべてにデータ転送のある転送元を、同一のグループに属する転送元は同一のセクタにつなぎ得るようにグループ分けを行う。その手順は、転送元の順序にしたがい、以下の通りを行う。

- (1) 最初、グループは一つであり（1番目のグループと呼ぶ）、順序が一番早い転送元のみが属している。
- (2) 順序が i 番目 ($i \geq 2$) の転送元をできるだけ順序が早いグループに入れる。
- (3) 入れられるグループがないときは、順序が最後のグループの後ろに新しいグループを作ってそこに入れる。

この操作を、まず条件1を満たす集合に対して行い、ついで、条件2を満たす集合に対して行う。ただし、条件1を満たす集合は包含関係について極大なもののみを記憶しておく。また、転送先の集合には、転送先の順序から決まる辞書式順序を入れておく。

条件1を満たす集合に対する手順

- (1) 条件1を満たす転送先の極大集合でまだ調べていないものを順序にしたがって持ってくる。すべての極大集合を調べ終わったら終了。
- (2) (1)で持ってきた極大集合を X とし、 X の部分集合で要素が3個のものうち順序が一番早いもの Y をとり、転送元のグループ分けを行う。セクタの総入力線数が減少を続ける限り、順序にしたがって X 内で Y を動かしながら転送元のグループ分けを続ける。ただし、動かすことにより Y の元の数が減るものは飛ばす。セクタの総入力線数が減少しなくなったら(3)へ。
- (3) セクタの二段使用が確定した転送元を除き(2)へ。セクタを二段使用する転送元がなければ、 X は調べ終わったことになるので(1)へ。

条件2を満たす集合に対する手順

- (1) 条件2を満たす集合を順序にしたがって持ってくる。すべての集合を調べ終わったら終了。
- (2) (1)で持ってきた集合を X とし、 X に対して転送元のグループ分けを行う。ただし、セクタを二段使用するのは、転送元三個以上のグループのみ。
- (3) (2)の結果セクタの総入力線数が減少していれば、セクタの二段使用が確定した転送元を除き(2)へ。セクタを二段使用する転送元がなければ、 X は調べ終わったことになるので(1)へ。

以上すべての操作が終了したときのセクタの割付を最終結果として採用する。

本アルゴリズムの一適用例の結果を表1、図2に示す。この例では、転送元20個、転送先17個、 $s \rightarrow d$ なるデータ転送のある転送元 s と転送先 d の対 (s, d) の数が75個、 $s \rightarrow d$ 、 $t \rightarrow e$ なるデータ転送が同時に起こる転送元 s 、 t 、転送先 d 、 e の四つ組 (s, t, d, e) の数が252という転送状況で、処理時間はワークステーション上で2、3分程度であった。

4. おわりに

本稿では、リソース割付情報から準最適データ転送路を自動割付するためのアルゴリズムを提案し、その適用例について簡単に触れた。

本アルゴリズムの適用によって、人手による割付にほぼ匹敵する最適化のレベルが得られるとともに、所要時間では大幅な短縮が期待できる。

今後は、われわれが開発している論理合成システム PARTHENON[5]へこれらの手法を組み込むことにより、ハイレベルシンセシス機能の追加を進めていきたい。

【参考文献】

- [1] M.C.McFarland, A.C.Parker, and R.Camosano, "Tutorial on High-Level Synthesis", Proc. 25th Design Automation Conf., pp.330-336(1988).
- [2] A.C.Parker, J.Pizarro, and M.Milnar, "MAHA: A Program for Data Path Synthesis", Proc. 23rd Design Automation Conf., pp.461-466(1986).
- [3] 若林, 吉村: 「複雑な制御構造をもつアルゴリズムからの高速なステートマシン合成手法」, 第37回情報処理学会全国大会論文集, pp.1729-1730(1988).
- [4] R.Camosano and W.Rosentiel, "Synthesizing Circuits From Behavioral Descriptions", IEEE Trans. CAD, Vol.8, No2, pp.171-180(1989).
- [5] Y.Nakamura, K.Oguri, A.Nagoya, and R.Nomura, "A Hierarchical Behavioral Description based CAD System", EURO ASIC90(1990).

表1 各段階での結果

	セクタ数	セクタの総入力線数
初期状態	17	75
第一段階終了時	8	60
第二段階終了時	11	55

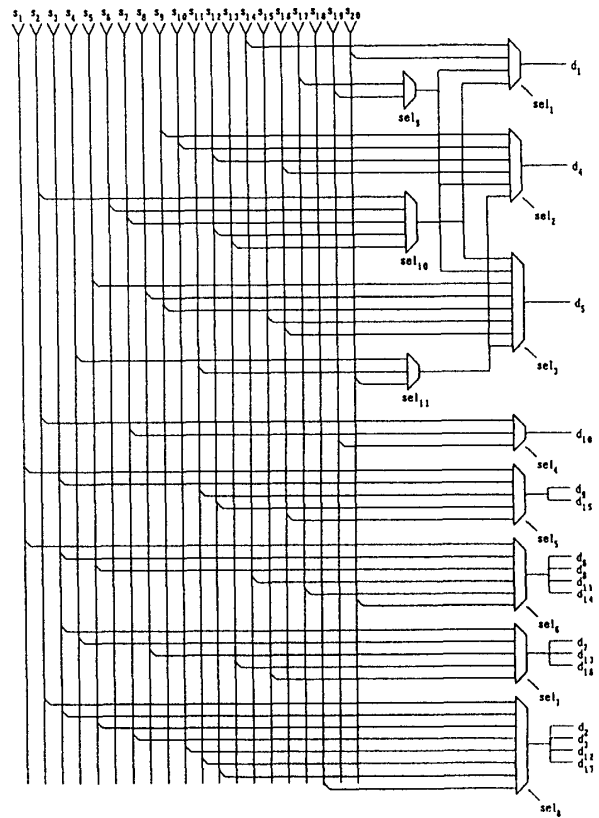


図2 最終結果