タイミング制約を考慮した分割手法†

6 L－1

南 淳一朗　小出 哲士　若林 真一
広島大学 工学部

1 まえがき

近年のVLSI技術の進歩により，回路はますます大規模化の傾向にある．大規模回路を１つのチップに実現することは大変困難であるため，与えられた回路を複数に分割し，複数のチップを用いて回路を実現する必要がある．これまで様々な回路分割手法が研究されている [1] が，近年の回路の動作速度の高速化によって分割の際にタイミング制約を考慮することが必要となってきた．従来のタイミング制約を扱った分割問題の遅延モデルとしては，単位遅延モデル [2] などがある．文献 [2] では異なるチップを接続する配線のみに遅延を設けており，チップ内のゲートの遅延を全く考慮にいれていないため，遅延の見積りが正確ではない．一方 general delay model は (1) 各ゲートにそれぞれの遅延を設け，(2) チップ内の配線遅延は無視し，(3) 異なるチップ間をまたぐ配線にある一定の遅延を与えるモデルであり，単位遅延モデルより正確に遅延を見積もることが可能である．文献 [3] などでは，この general delay model を用い，制約をI/Oピン数と面積とし，入力から出力までの最大遅延が最小となるような分割を求めるアルゴリズムを提案している．しかし，これらの従来手法は組合せ回路の入力から出力までの遅延時間の最小化のみに着目しており，順序回路を直接扱うことはできない．そこで，本稿では general delay model を用い，順序回路のレジスタ間の遅延制約を考慮した回路分割手法を提案する．

2 タイミング制約を考慮した回路分割

本研究で取り扱うグラフモデルは，回路素子をノード，ネットを枝で表したハイパーグラフ $G=(V,E)$ で，各ノードは面積 $a(v_i)$ と遅延 $d(v_i)$ を持つ．配線遅延モデルは general delay model を用い，カットされる枝には一定の配線遅延 $d_c \gg d(v_i)$ を設ける．配線遅延制約は全てのレジスタ間に設ける．レジスタ $r_i, r_j \in V$ 間のパス上に存在するノード集合を P_{ij}，パスのカット回数を $ncut(P_{ij})$ とすると，r_i, r_j 間の配線遅延制約は

$$\sum_{v_i \in P_{ij}} d(v_i) + d_c \times ncut(P_{ij}) < D$$

となる．また，面積制約は

$$\sum_{\Phi(v_i)\in V_j} a(v_i) \leq \frac{\alpha \times \sum_{v_i \in V} a(v_i)}{2} \quad (j=A,B,\ \alpha>1)$$

となる．図１は general delay model を用いて r_1, r_2 間の配線遅延を示したものである．

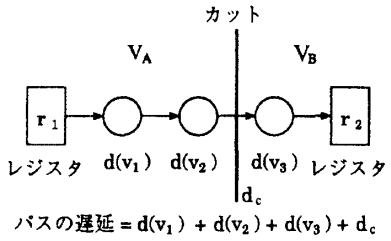

図 1 general delay model

【タイミング制約を考慮した回路２分割問題】

入力	$G=(V,E)$
出力	$\Phi : V \rightarrow \{V_A, V_B\}$ $(V = V_A \cup V_B, V_A \cap V_B = \phi)$
目的関数	カット数の最小化
制約	$\sum_{v_i \in P_{ij}} d(v_i) + d_c \times ncut(P_{ij}) < D$ $\sum_{\Phi(v_i)=j} a(v_i) \leq \frac{\alpha \times \sum_{i=1}^{n} a(v_i)}{2}$ $(j=\{V_A,V_B\}, \alpha>1)$

3 提案アルゴリズム

提案手法は大きく分けて次の２つのフェーズからなる．

フェーズ１：タイミング制約を考慮したクラスタリング
フェーズ２：FM法による分割

本研究では大規模回路を扱うため，実用的な時間で分割を得るにはノード数を減少させる必要がある．ノード数を減少させるための方法としてクラスタリング [1] が知られている．本研究で扱うモデルはレジスタ間に配線遅延制約を設けているため，クラスタリングの際，レジスタ間のパス長を考慮する必要がある．以下に，配線遅延制約を考慮したクラスタリングアルゴリズムを提案する．

3.1 フェーズ1:タイミング制約を考慮したクラスタリング

フェーズ１では，ノード数を減少させるために，以下の２つのステップでクラスタリングを行なう．

ステップ１：配線遅延制約に基づくクラスタリング
ステップ２：３つの制約を考慮したクラスタリング

ステップ１ではレジスタ間のパスがカットされると必ず配線遅延制約違反が生じるパス上のノードをクラスタリングする (図2)．

次にステップ２では，与えられたハイパーグラフの全ての枝 $e \in E$ に

$$info(e) = (plen(e), ncp_1(e), ncp_2(e), ncp_m(e))$$

の４項組の値を与える．ここで，$plen(e)$ はその枝の属するレジスタ間のパス長の最大値 (すなわち，パスの

†"A timing-driven circuit partitioning method", Jun'ichiro MINAMI, Tetsushi KOIDE, Shin'ichi WAKABAYASHI, Faculty of Engineering, Hiroshima University. e-mail:{south, koide, wakaba}@ecs.hiroshima-u ac.jp

カットされうる回数), $ncp_i(e)$ は枝 e の属するレジスタ間のパスで i 回までカットできるパスの数. ただし, $ncp_m(e_i)$ は 3 回以上カット可能なパスの数を表すものとする. (図 3).

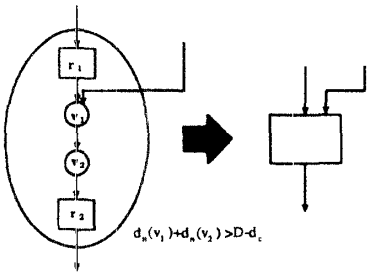

図 2 ステップ 1 のクラスタリング

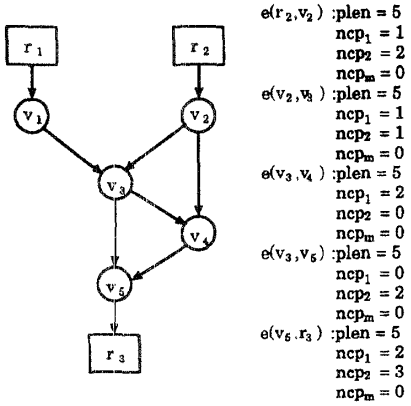

図 3 枝の情報

以下に, $ncp_1(e)$ を求めるアルゴリズムを示す.

```
calc_ncp1(e = (vi, vj)){
  while(d=レジスタまでの遅延){
  /* ここでは, vからレジスタまでの
     全てのパスの遅延を d に代入する */
      Path+=Number_Of_Path(vi, D − (d + 2dc));
  }
  return Path;
}
Number_Of_Path(vi, d){
  if (vi==レジスタ){
      if (d < 0) return 1;
      else return 0;
  }
  else{
      while(next=ノード viに入力のあるノード){
          Count +=
            Number_Of_Path(next,d − dn(vi));
      }
  return Count;
  }
}
```

$ncp_2(e)$ も $ncp_1(e)$ と同様にして求めることができる. 次にこの枝の情報を基に以下の計算式を用いて各枝 e の重み $w(e)$ を計算する.

$$w(e) = \alpha \times plen(e) \times (2 \times ncp_1(e) + ncp_2(e))$$

この枝の重み $w(e)$ を用いて枝 e に接続するノード v_i, v_j 間のコスト関数 $cost(v_i, v_j)$ を以下のように定める.

$$cost(v_i, v_j) = w(e) + \beta \times \frac{1}{a(v_i) \times a(v_j)}$$

ステップ 2 では $cost(v_i, v_j)$ の大きいペアから順にノード数が一定数になるまでクラスタリングする. そして枝の情報 $info(e)$ はノードをクラスタリングするたびに再計算する.

3.2 フェーズ 2:FM 法による分割

フェーズ 2 では, 面積制約, 配線遅延制約の下で FM 法 [4] に対してタイミング制約をゲインで取り扱うように拡張した文献 [5] の手法を適用する. タイミング制約の考慮のために, 以下に定義する $vio(e)$ を各枝に与える.

$$vio(e) = \begin{cases} 0 \text{ (その枝が属するパスが制約を満たしている)} \\ ncut(e) - \lfloor D/d_c \rfloor \quad (otherwise) \end{cases}$$

以下に, $vio(e)$ を求めるアルゴリズムについて示す. ここで, $Max\_Delay(v_i, v_j)$ は v_i から v_j までの最大遅延を返す関数である.

```
calc_vio(e = (vi, vj)){
    int Vio;
    d1 = d2 = 0;
    while(r = viまでのパスを持つ全てのレジスタ){
        d1=Max(d1,Max_Delay(r, v1));
    }
    while(r=v2からのパスを持つ全てのレジスタ){
        d2=Max(d2,Max_Delay(vj, r));
    }
    Vio=⌊((d1 + d2) − D)/dc⌋
    if (Vio<0) return Vio;
    else return 0;
}
```

FM 法では枝 e の重みは, $1 + vio(e)$ とし, ゲイン

$g(v_i)$ = (カットされる枝の重みの総和の減少度)

を求め, 文献 [5] の拡張 FM 法を適用して 2 分割を求める.

4 あとがき

本稿は general delay model 上で, タイミング制約を考慮した分割手法を提案したが, 詳細は現在検討中である. 今後, クラスタリング手法, FM 法のゲインの計算方法の改善等を行なう予定である. また, シミュレーション実験によるアルゴリズムの有効性の検証も行なう予定である.

文献

[1] C. J. Alpert and A. B. Kahng: "Recent direction in netlist partitioning: a survey", INTEGRATION the VLSI journal, pp. 1–81 (1996).

[2] R.Murgai, R.K.Brayton and A. S. Vincentelli: "On clustering for minimum delay/area", Proc. ICCAD91, pp. 6–9 (1991).

[3] H. Yang and D. F. Wong: "Circuit clustering for delay minimization under area and pin constraints", Proc. ED & TC95, pp. 65–70 (1995).

[4] C. M. Fiduccia and R. M. Mattheyses: "A linear-time heuristic for improving network partitions", Proc. ACM/IEEE DAC82, pp. 175–181 (1982).

[5] T. Koide, et al.: "A new performance driven placement method with the Elmore delay model for row based VLSIs", Proc. ASPDAC95, pp. 405–412 (1995).