# 2J−8　評価基づく探索肢選択による再設計を含むアルゴリズムレベル回路設計

徐　行倹　　石塚　満

東京大学　生産技術研究所

## １．まえがき

高品質なＶＬＳＩ製品を短期間で設計することのできる知的な設計支援システムが強く望まれている。ＶＬＳＩ回路設計システムのレベルによって、アルゴリズムレベル設計システムや、ＲＴＬ（Register Transfer Level）設計システムや、ロジックレベル設計システムなどの設計システムがある。アルゴリズム総合システムでは、ＣやＰＡＳＣＡＬのようなハイレベルの言語で、回路の動作仕様を書く。ハードウェアの知識を持っていない人でも回路の設計ができる。こうような設計システムは、回路の処理速度を目標として開発したものが多く、できるだけ高速な回路をだすために、いろいろな優先度関数や最適化の技術がよく使われている。但し、そうような技術を用いると、回路面積の制御が難しくなり、回路の処理速度と面積の制約は同時に柔軟な制御ができない。仮説推論の考え方に基づき、制約条件を満たすような再設計を容易にした設計システムについて述べる。

## ２．仮説推論を用いた設計方法

アルゴリズムレベル設計システムでは[2][3]、ＣやＰＡＳＣＡＬのようなハイレベルの言語で、回路の動作仕様を書く。回路の仕様として処理速度と面積を制約するために、優先度関数や設計した回路の最適化の技術を使うが、優先度関数に依存して結果が異なる。優先度関数は、あらかじめ設定されているが多いので、回路によって速度と面積を制御することが難しかった。この不十分点に対して、仮説推論を用いた設計方法により回路の処理速度と面積の制約に関して同時に柔軟な制御ができるという可能性を持つ。

アルゴリズムレベル設計システムの問題点の原因は、一つの設計結果が出てきたら、制約に対する再設計を行わないことにある。仮説推論を用いた設計システムでは[1][4]、無矛盾性の制約条件をチェックしながら仮説を合成していくので、再設計プロセスが組み込まれている。この再設計は制約条件を満たすまで繰り返される。このような仮説推論の考え方を取り入れて、アルゴリズムレベル設計として、以下の手法の開発を行っている。

【ｓｔｅｐ１】：入力仕様書を Compile して、Data Flow Graph と Control Flow Graphを作る。

【ｓｔｅｐ２】：普通の Algorithm Synthesis 技術で、仕様書に書いたのと同じ動作をする回路を設計する。このとき、回路の処理速度と面積制約は考慮せず、Data Flow GraphとControl Flow Graphから最もカスケードな回路を設計する。最もカスケードな回路とは、全く並列に動作せず、面積が一番小、時間が一番遅い回路である。

【ｓｔｅｐ３】：仮説推論を用いた再設計。この時に、ｓｔｅｐ２の結果を出発点にして修正（仮説の入れ替え）を行い、設計ゴールを最も満足する設計結果を導く。回路の設計は、図１のような木構造で階層的に記述できる。木の根は設計したい回路で、木の葉は回路の基本ブロックで、中間ノードはサブブロックである。もし、制約条件を満たさなければ、ｉ番ノードの部分は取り替えされて、新しい仮説とする。ｉ番ノードの可能な取り替えを全て試みたら、上のｄ番ノードへバックトラックする。このように、制約条件が満たされるまで、制約条件をチェックしたり、バックトラックして部品を取り替えたりして設計を行う。

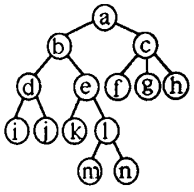



図１回路設計の木構造

図２の例はＣ言語で書いたKalman Filterの設計仕様の一部で、図３はこの仕様書に対するData Flow GraphとControl Flow Graph (Step1)で、図４(a)は初期設計の結果(Step2)である。Step3の結果は回路の時間と面積制約によって、いろいろな回路があり、図４(b)はその一例である。

この手法で設計すると、速度や面積の制約条件を満たす結果が保証される。ただし、バックトラックを含む、設計時間が多くかかてしまう。特に、回路の規模が大きい時には膨大な時間となってしまう。

## ３．仮説推論を用いた再設計の高速化

通常の仮説推論では、推論の途中で矛盾な仮説の組合せを早期除くことにより高速化が図れる。しかし、回路設計では、制約条件は回路の速度や面積なので、回路の完成まで、制約を満たすかどうかが分からない。従って、深いバックトラックとなり推論が遅くなる。選択肢が一意に決まらない場合は、その時点で最も適当と思われるものを選択することになるが、これは部分的にみて適当ということで、全体でみると良い選択でない可能性がある。

このような問題点を避けて回路の設計を高速化することを目標として、初期設計の結果に基づいて、グローバルに、最適な選択肢を一意的に決めて再設計する次のような方式を開発している。回路の時間制約と面積制約に対して、探索木の各可能な探索の評価を行い、このうちで設計ゴールに一番近い結果を選んで設計する。ここでいう設計ゴールに一番近い結果とは、文献[1]と同様に、重み付けされた制約を最も多く満たす解を意味する。

以下では、制約条件違反に対する色々な変更方法に基づく評価・再設計の高速化について述べる。

### （３−１）時間の探索木

```
i = 15;
do {
        j = 15;
        tempi = 0;
        do {
                tempi = tempi + Anew[i@j]*x[j];
                if ( j >= 12 )
                        tempi = tempi + k[i@j]*y[j];
                j = j - 1;
        } while ( j >= 0 );
        x[i] = x[i] + 255*tempi;
        i = i - 1;
} while ( i >= 0 );
```

図２ Kalman Filter仕様の一部

---

An Algorithmic Level LSI Design including Redesign by Selecting Search Branch based on Evaluation

Xing-jian Xu　Mitsuru Ishitsuka ( Univ. of Tokyo )

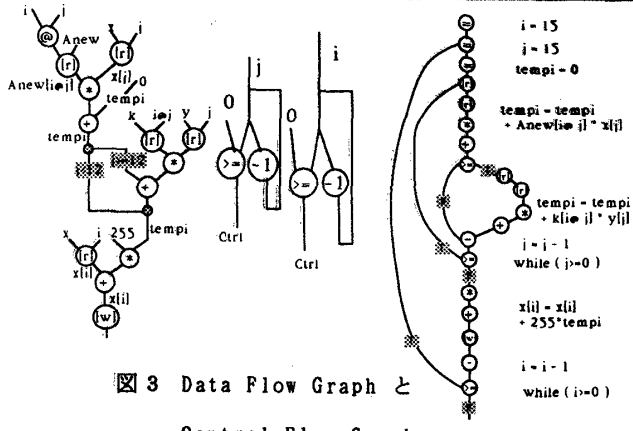



図３ Data Flow Graph と Control Flow Graph

図３のData Flow GraphとControl Flow Graphから、各操作が動作する回数が求められる。これから図５(a)のような回路の時間探索木が作られる。木の根は設計したい回路、木の葉は回路の各操作、中間のノードは操作のサブブロックである。ここで、白い円（○）はこのノードの時間数としてこのノードの下の選択肢の中で一番大きい時間数を選ぶことを意味する。黒い円（●）のノードの時間数はそのノードの下の各選択肢の時間数の合計とする。そうすると、探索木中の各ノードの時間数が分かる。ここでいう時間数は一回回路が動作するときに、各ノードにかかるクロック数である。各ノードはそれぞれの時間数と面積というデータが持っている。

（３－２）探索木の評価による、結果を探す

図５(b)の時間探索木は最もカスケードな回路から作られたので、これは回路面積が一番小、処理速度が一番遅い回路である。設計目標に対して、この設計からこの回路の処理速度を速くしなければならない。速度を速くすることは時間探索木のどちらの葉を削除することができるかという問題に等しい。

先ず、探索木の各葉を削除できるかどうかをチェックする。もし、これができれば、速度がどのぐらい速くなるかと面積がどのぐらいの大きさになるかの評価を行う。例として、図５(a)探索木の Memory Data Fetch 操作([r])葉を評価してみる。

ここで、二つの連続した Memory Data Fetch 操作がある。しかし、Bus と Memory Controller は一つしかない。従って、これが２つクロックかかる。もし、一つのBusとMemory Controllerをそれぞれ二つの小さなBusとMemory Controllerに分ければ、この二つの Memory Data Fetch 操作を並列に実行することができるので、１クロックで済む。これに対する回路と時間探索木はそれぞれ図４(a)と図５(b)になる。結果として、時間的に、320クロック( 19.80 % )速くなる。同様にして、どのくらい面積が大きくなるかという評価データも得られる。

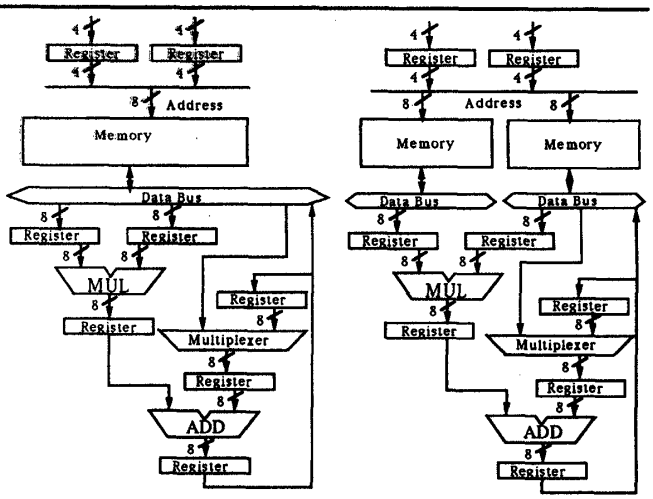
これは回路の再設計する方法うちの一つで、別の再設計方法も色々がある。これと同様に、全て可能な再設計する方法を並列で評価して、一番良い方法を選ぶことにより、バックトラックを回避できる。この手法で、柔軟な制約の充足と設計時間の速さを得ることができる。

４．むすび

以上紹介した設計方法は以下の問題点がある：

（ａ）時間の探索木を作る時に、ループの回数を計算するが、ループ回数の計算ができない場合のある。この場合、ループ回数は近似計算される。仕様を書くときに、注意すればよいが、不便なことである；

（ｂ）初期設計した回路に基づいて回路を再設計するが、最終結果は構造的に初期設計した回路に似ている。




(a)初期設計結果(Step1)　(b)制約を満たすような再設計結果

図４　回路の設計結果

同じ機能を持つ構造が異なる回路を探すことができない；

現在、ここに記したアルゴリズムレベル設計システムの作成を進めている。

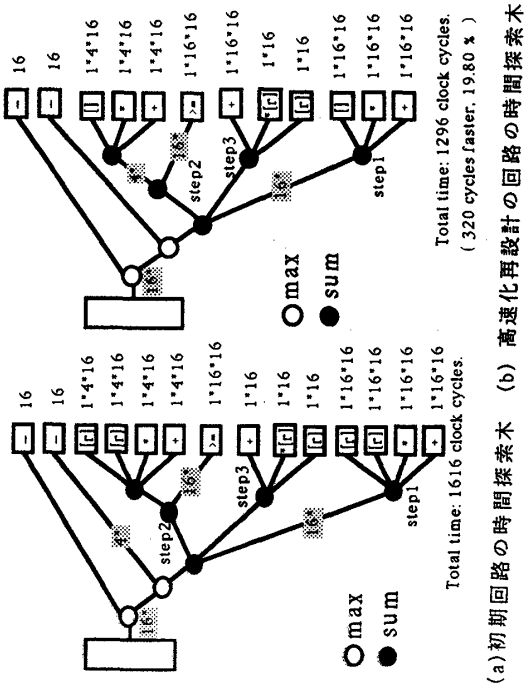
参　考　文　献

[1]牧野、石塚：制約評価機構付き仮説推論システムとその回路ブロック設計への応用、人工知能学会、Vol.5 No.5 pp640-648 (Sept. 1990)

[2]D.E. Thomas E.D. Lagnese R.A. Walker J.A. Nestor J.V. Rajan R.L.Blackburn:Algorithmic and Register-Transfer Level Synthesis: The System Architect's Workbench, Kluwer Academic Publishers, 1990

[3]Machael C.McParlend, SJ, Alice C.Parker, Raul Camposano: Totorial on High-level Synthesis. 25th AMC/IEEE Design Automation Conference. pp330-337, 1988

[4]丸山、角山、松永、簑田、沢田、川戸：評価・再設計機構を備えた論理設計支援システム、電子情報通信学会論文誌 Vol.J72-A No.8 pp1172-1180 (1989.8)




(a)初期回路の時間探索木　(b) 高速化再設計の回路の時間探索木

図５　時間探索木