

探索空間を縮小する高位合成手法

Search Domain Reduction in High-Level Synthesis

増田 篤司
A. Masuda

関根 優年
M. Sekine

ビヘイビア記述から論理回路を合成する高位合成システムは、膨大な設計空間の中から解を求める問題であり、最適解を求めるのはほぼ不可能に近い。この問題を解決するために、高位合成手法についていろいろと研究されているが、できあがる回路の質が悪すぎるためにまだ実用化できていない。

このシステムでは、新たなユーザ定義の制約条件を付加してこの探索空間を制限し、そのなかでの局所解を求めることで問題を解決しようとしている。この方法により、より高品位の論理回路が短期間で設計できるようになった。

Most existing high-level synthesis systems attempt to generate a circuit from a behavioral description “out of the void,” using the entire design space as the search domain. Because of the vastness of such a search domain, it is impossible to perform more than a coarse-grain search, often resulting in inefficient designs.

Our approach ignores the designer’s knowledge of the general structure of the circuit to be synthesized. By limiting the search domain, a much finer grain search can be performed yielding a higher quality design.

1 まえがき

近年、論理 LSI の設計が複雑化してきているため、従来の論理合成システムによる論理設計より容易に設計が可能な高位合成システムが注目されている。

しかし、高位合成システムは世界中で研究されているにもかかわらず、実用化の段階には達していないのが現状である。これは、高位合成の問題が、膨大な探索空間の中から最適な論理回路を探し出す問題であり、現状の技術では高位合成される論理回路の質が悪すぎるためである。

当社では、この問題を解決するために、新たなユーザ定義の制約条件を付加し、探索空間を縮小するという方法を提案し、実用化のための研究開発を行っている。

2 高位合成システム

高位合成システムとは、図 1 に示すように、従来設計者が行っていた論理設計を自動化するもので、論理合成システムの入力記述であるレジスタ転送レベル (RTL) 記述よりも抽象度の高いビヘイビア記述を入力として、RTL 記述を自動合成するものである。

ビヘイビア記述では、論理回路の動作をプログラミング言語で記述するように逐次的に表現する。この際には、論理回路の構造的な部分や、タイミングなどはいっさい考慮せず、

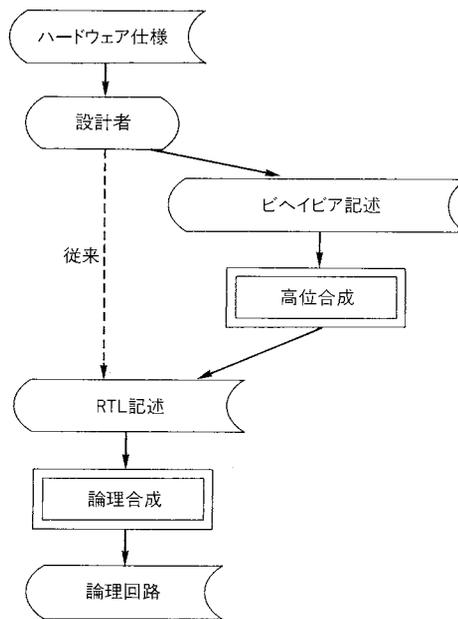


図 1. 高位合成の位置づけ 従来設計者が行っていた部分を高位合成に置き換える。

Positioning of high-level synthesis system

動作だけに着目して書き表せばよいので、容易に書き表すことができる。当社のシステムでは、このビヘイビア記述としてプログラミング言語である C++ 言語を採用しているため、C++ 記述が書ければ論理回路の設計が行える。

3 高位合成システムの処理の流れ

当社の高位合成システムは、図2に示すような処理フローに沿って論理回路を合成する。

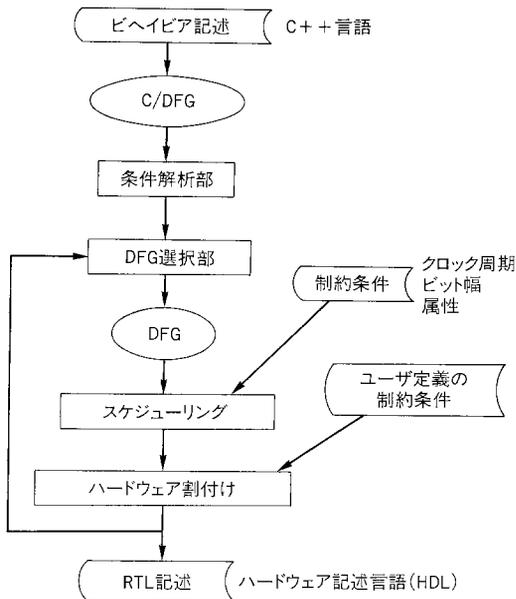


図2. 高位合成の処理フロー 当社の高位合成システムは、このフローに沿って内部処理を行っている。

Flow of high-level synthesis system

以下に処理フローに沿って簡単に説明する。

3.1 条件ごとの部分合成

まず、入力されたビヘイビア記述(C++記述)を、制御とデータの流れを表すコントロール/データフローグラフ(C/DFG)に変換する。

一般の高位合成システムでは、このC/DFGを基にハードウェアの設計をしていくが、制御を含んだままで高位合成を行うのは非常に複雑なアルゴリズムを必要とするので、このシステムでは図3に示したように、C/DFGをさらに条件ごとに分割して条件を含まないDFGの集合にし、各DFGを基にハードウェア設計をして、最後にそれらを一つにまとめるといった手法を採っている。このように一つの問題を複数の部分問題にすることで、ハードウェア設計の複雑度の軽減を図っているのがこのシステムの一つの特長である。この、C/DFGをDFGに分割するために、すべての制約条件の組合せを求める条件解析部と、各条件の組合せ下で動作する範囲を抽出するDFG選択部とを備えている。

そしてこの分割されたDFGごとに、スケジューリング、ハードウェア割当てを行う。

3.2 スケジューリング

スケジューリングとは、DFG上の演算をどのクロックサイ

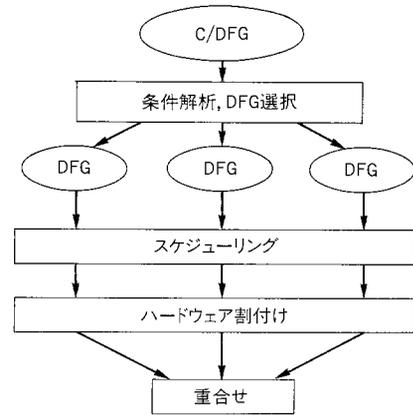


図3. 条件ごとの部分合成法 C/DFGをDFGに分割した後の処理の流れ。

Factoring of C/DFG into individual DFGs

クルで実行するかを決定する処理である。スケジューリングの前処理として、入出力ピンのビット幅から有効けた数を考慮して各演算のビット幅を計算し、それを基に仮想的な遅延時間を計算する。その後、その演算を実行すべきクロックサイクルに割付けを行う。この際に与えるクロック周期を変更することにより、合成される論理回路をそのクロック周期に合った実行形態にすることができる。つまり、同じ動作仕様であれば、目的とする論理回路のクロック周期や、データのビット幅に対応させて、これらの制約条件を変更するだけで容易に設計のやり直しができることになる。

スケジューリングの実施例を図4に示す。図4(a)はスケジューリングを行うDFGであり、データの流れだけを表している。図4(b)のスケジューリングされたDFGでは、それに加えて各演算を実行するクロックサイクルが表されている。図中のST1~ST6はクロックサイクルを表しており、そのサイクルで実行される演算が右に書かれている。

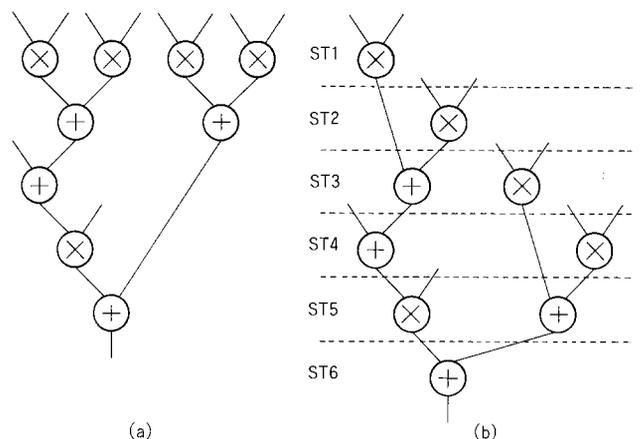


図4. スケジューリング例 (a)スケジューリング対象のDFG, (b)スケジューリング後のDFG

Example of scheduling

3.3 ハードウェア割付け

ハードウェア割付けとは、スケジューリングされたDFGの演算を実際にその演算を実行するハードウェア素子に割り付ける処理である。この際には、与えられた設計制約（ハードウェア素子の個数や内部結線の本数など）を満たす最適な割付けを行う必要がある。

この問題は、膨大な探索空間の中の一つの最適解を求めるものであり、非常に困難な問題である。この解決方法の一つとして、高位合成システムが合成する論理回路のアーキテクチャを限定し、探索空間を縮小させるという方法が考えられる。

しかし、合成するアーキテクチャを想定したハードウェア割付けのアルゴリズムにした場合は、想定外のアーキテクチャの論理回路を合成することは困難になってしまう。

そこでこのシステムでは、アーキテクチャを限定する情報をユーザが与え、その情報に基づいてハードウェア割付けを行うという手法を採用している。

このユーザ定義のアーキテクチャ情報は、単にCPUとかDSP (Digital Signal Processor) といった情報ばかりではなく、さらに詳細な情報を与えることができるような構造をしている。この情報に基づき、対象となる論理回路を限定することで、探索空間の大幅な縮小が可能となる。

さらに、このユーザ定義情報を変更すれば、図5に示すように、合成される論理回路のアーキテクチャも容易に変更可能となる。

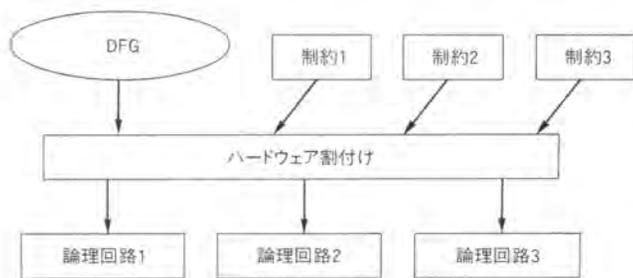


図5. 制約変更例 設計制約を1から3のように変更することにより、同一のDFGから合成される論理回路を1から3のように変更できる。
Example of changing constraints

3.4 重合せ処理

最後に各条件の組合せごとに高位合成された論理回路の重合せを行う。

各条件の組合せの下で動作するハードウェアはすでに決まっているので、そのハードウェアの動作条件としてそれらの論理和をとったものを与える。

このようにして最終的な論理回路が高位合成される。

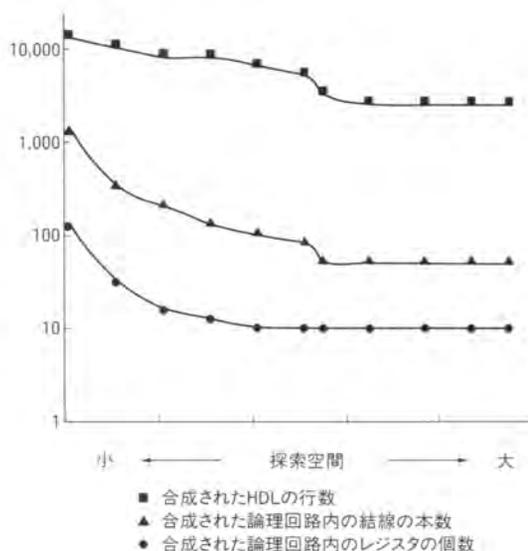


図6. 実施例 探索空間を変化させた場合に合成された論理回路を示す。

Example of application of system

4 実施例

このシステムを適用して高位合成を行った一例を図6に示す。図6中の横軸は探索空間の広さを表している。

この結果から、同じ動作の論理回路を合成するにしても、制約条件を変更して探索空間の領域を変化させると、探索空間が広いほうが良い論理回路が合成できていることがわかる。しかし、探索空間が広ければその分合成にかかる時間は大きくなる。ユーザはこのトレードオフを考慮し制約条件を変化させ、合成される論理回路の質を制御し、目的（面積，速度）に合った論理回路を設計することになる。

5 あとがき

当社で研究開発を行っている高位合成システムの概要を述べた。このシステムによれば、論理設計にかかる設計者の負担軽減につながるばかりでなく、論理設計にかかる期間の大幅な短縮が期待できる。今後実用化に向けてよりいっそうの努力を進めていく。



増田 篤司 Atsushi Masuda

1988年入社。大規模論理LSI設計用CADの研究・開発に従事。現在、研究開発センター ULSI 研究所研究主務。
ULSI Research Labs.



関根 優年 Masatoshi Sekine

1976年入社。バイポーラメモリ、64KビットCRAM、VLCAD機能設計CADの研究・開発に従事。現在、研究開発センター ULSI 研究所研究主務。
ULSI Research Labs.