

インタラクティブ ビヘイビア合成環境

Interactive Behavioral Synthesis Environment

今井 浩史
H. Imai

チップ上に実装するシステムの大規模化に対応するため、ビヘイビア合成を用いた設計手法が提案されてきている。この設計手法では、システムの動作をプログラムを書く要領で記述する。ビヘイビア合成は、この記述を加工し、論理回路を生成する。これにより、従来より短期間で設計を終えることが可能となる。しかし、従来のビヘイビア合成では満足のいく回路を設計することは難しい。そこで、ビヘイビア合成の処理に設計者が介入できるインタラクティブ ビヘイビア合成環境を開発した。これにより、満足のいく回路を短期間に設計することが可能となる。

The increase in complexity of single-chip systems motivates a design methodology that includes synthesis from behavioral descriptions, or so-called behavioral synthesis. In this methodology, the behaviors of a system are designed like programs. Behavioral synthesis transfers the description into a logic circuit. This approach drastically reduces the design time.

Since most behavioral synthesis systems generate logic circuits which are inferior to those created by manual design, we propose a behavioral synthesis environment where the designer can interactively control the behavioral synthesis processes.

1 まえがき

半導体製造技術の目覚しい進歩に伴い、チップ上に実装されるシステムの大規模化が進んでいるとともに、市場への製品投入期間の短縮も求められている。このような状況の下で、論理合成ツール、レイアウトツールのようなEDA (Electronic Design Automation) ツールを利用して、システムの大規模化、設計期間短縮に対応してきた。しかし、いっそうのシステムの大規模化、設計期間短縮の要求に従来の設計手法では対応しきれなくなっている。

そこで、ビヘイビア合成を用いた新しい設計手法が提案されている。ビヘイビア合成を用いた設計では、プログラムを書く要領でシステムの動作を記述し、この記述にハードウェア化に必要な処理を加え、論理回路を生成する。これにより、大規模なシステムも容易に設計でき、設計期間も短縮できる。

しかし、この設計手法でも十分ではない。そのため、当社は米国のカリフォルニア大学アーバイン校との共同研究で、ビヘイビア合成の処理に設計者の介入が可能なインタラクティブ ビヘイビア合成環境を開発した。

2 ビヘイビア合成

ビヘイビア合成を用いた設計手法では、システムの動作をプログラムを書く要領で記述する。例えば、c言語に代表されるプログラム言語やVHDL^(注1)(Very high speed IC

```
while (x < a) {
    u1 = u - 5 × u × x × dx - 3 × y × dx;
    x1 = x + dx;
    y1 = y + u × dx;
    u = u1;
    x = x1;
    y = y1;
}
```

図1. ビヘイビア記述 ビヘイビア合成を用いた設計手法では、c言語に代表されるプログラム言語でシステムの動作をプログラムを書く要領で記述する。

Behavioral description

Hardware Description Language)などを用いて記述する(図1)。この記述をビヘイビア記述と呼ぶ。

ビヘイビア記述はシステムの動作を記述したもので、チップ上に実装するための論理素子、クロック、レイアウトといった論理回路と密接に結び付いた情報はいっさいもたない。このため、設計者はシステムの動作の設計にだけ集中することができ、システムを短期間に設計できる。ビヘイビア合成は、ビヘイビア記述にハードウェア化に必要な処理を施し、論理合成の入力であるRTL(レジスタ転送レベル)記述を生成する。RTL記述は論理素子、クロック

(注1) VHDLは、1983年、米国国防省の指導のもとに言語仕様の開発が始まり、87年に規格化された。これにより事実上世界標準となった。

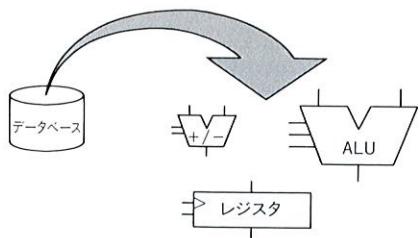


図2. アロケーション 論理回路部品のデータベースから、回路設計に必要な部品の種類と個数を決定する。この例では、加減算器を1個、ALUを1個、レジスタを1個を選択している。

Allocation

といった論理回路の情報をもった記述である。RTL記述が得られた後は、論理合成を用いた設計手法に従う。

ビヘイビア合成は主に三つの処理でビヘイビア記述からRTL記述を生成する。

(1) アロケーション この処理では、ビヘイビア記述の動作と同じ動作をするRTL記述を生成するために必要な論理回路部品の種類とそれらの個数を決定する(図2)。

(2) スケジューリング 論理回路はクロックに従って動作する。ビヘイビア記述の動作をクロックサイクルに従って実行するために、ビヘイビア記述をクロックサイクルごとに実行されるグループに分割する(図3)。この処理は、各動作のデータ依存性とアロケーションで決定された論理回路部品の種類と個数を考慮し、ビヘイビア記述中の動作の並列性を生かすように記述を分割する。アロケーションで十分な数の論理回路部品が選択されていると、ビヘイビア記述中の動作の並列性を最大限に利用することができ、回路規模は大きいが性能のよい論理回路を生成することが可能になる。

(3) バインディング ビヘイビア記述中の各動作をアロケーションで選択した論理回路部品中のどの部品で実行させるかを決定する(図4)。スケジューリングで異なるクロックサイクルに分割された動作は同じ部品で実行でき、このような部品の共有化により回路規模の増大を防ぐことができる。

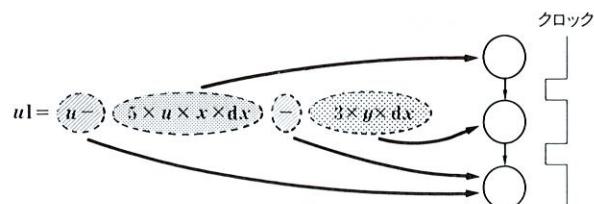


図3. スケジューリング ビヘイビア記述中の演算(掛算、減算)をクロックサイクルごとに実行されるグループに分割する。この例では、上のグループから順にクロックに従って実行される。

Scheduling

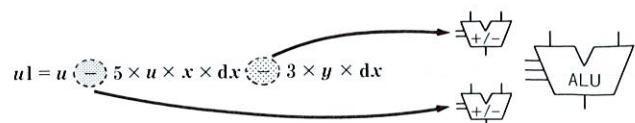


図4. バインディング ビヘイビア記述中の演算をどの回路部品で実行させるかを決定する。この例では、二つの減算がそれぞれ異なる加減算器に割り当てられている。

Binding

これら三つの処理は互いに密接に関連した処理であり、これらの処理のよしあしが生成される論理回路の規模と性能に大きな影響を与える。

3 インタラクティブビヘイビア合成環境

完全自動なビヘイビア合成では、設計者が手設計した回路と同等な回路を生成することを目指して、数多くのアルゴリズムが提案されている。しかし、現状では手設計した回路と比較して回路規模、性能の面で満足のいく結果を得るのは難しい。さらに、ビヘイビア合成自身のアルゴリズムが複雑なため生成される回路を予想することが難しく、ビヘイビア記述中の各動作と生成される回路中の部品との対応づけも不可能に近い。したがって、生成された回路のボトルネックの発見が困難で、修正も難しい。

そこで、これらの問題を解決するため、ビヘイビア合成の各処理に設計者が介入できる環境を提供し、設計者の意図する回路を生成する手法が研究されている。この手法の研究では、カリフォルニア大学アーバイン校のProf. D.D. Gajskiが世界をリードしており、当社と共同研究を行っている。

3.1 設計手法

アーバイン校で研究されているビヘイビア合成環境はISE(Interactive Synthesis Environment)⁽¹⁾と呼ばれている(図5)。ISEの特長は、アロケーション、スケジューリング、バインディングの各処理を自動でも手動でも実行できる環境であり、設計上重要な箇所は設計者みずからが手動で設計し、残りを自動で設計するといった設計手法を提供することである。

アロケーション画面で、回路部品データベースから回路部品を選択し、それらの個数を決定する(アロケーション)。ビヘイビア記述表示画面で、記述をクロックサイクルごとに実行されるグループへ分割する(スケジューリング)。手動で分割することも、クロックサイクルの値を与えて自動で分割することもできる。ビヘイビア記述表示画面とアロケーション画面を用いて、記述中の各動作を回路部品に割り当てる(バインディング)。この処理も手動または自動で行うことができる。

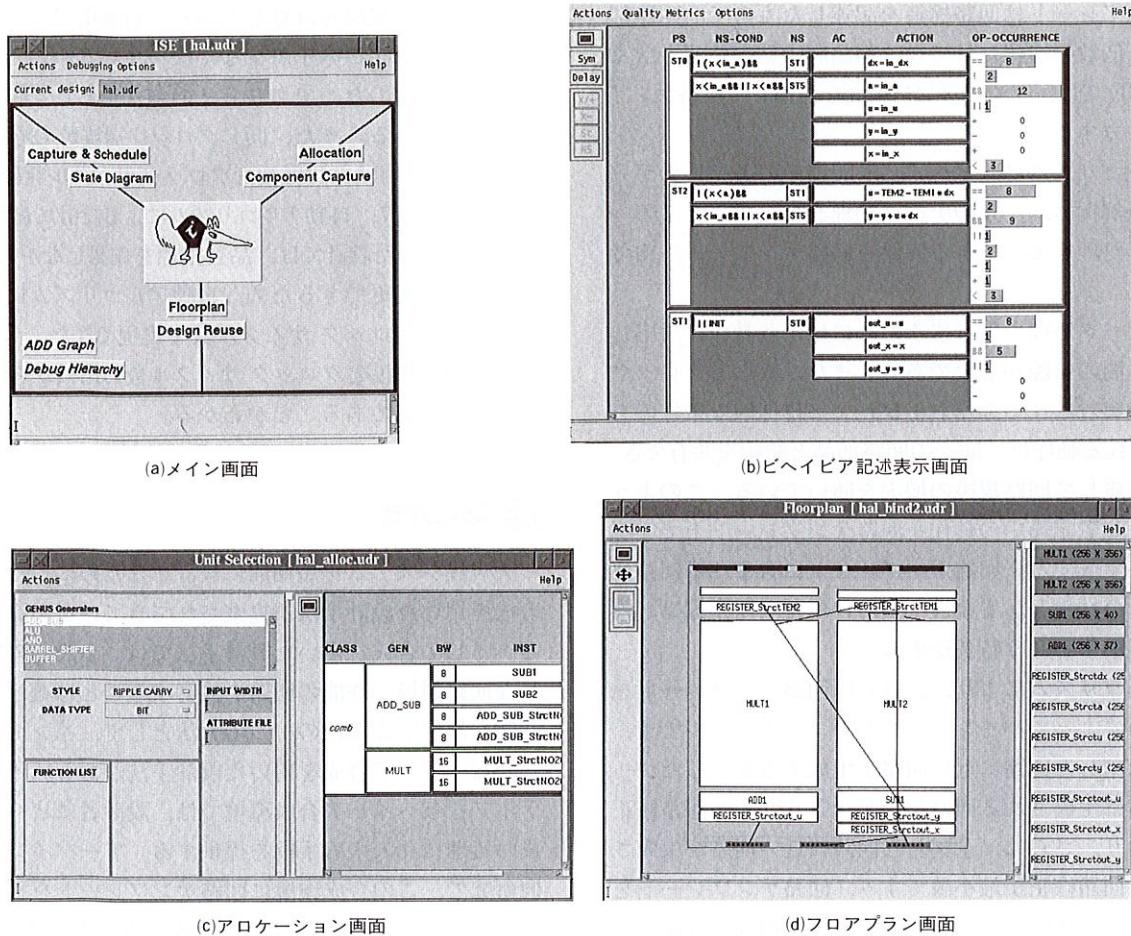


図5. ISE メイン画面、ビヘイビア記述表示画面、アロケーション画面、フロアプラン画面を示す。これらの画面を使用し、アロケーション、スケジューリング、バインディングをインタラクティブに実行する。フロアプラン画面には、設計された回路が表示される。

Examples of interactive synthesis environment (ISE) displays

手動でバインディングを行う際、いまバインディングしようとしている動作に対して、どの回路部品でその動作を実行させると回路規模が小さくなると期待できるか、その回路部品の候補を示すことができる。このように、ISEは設計のヒントとなる情報を提供する機構をもっている。生成された回路がフロアプラン画面に表示される。

さらに、ISEはビヘイビア記述中に各動作と生成された回路中の部品との対応付けの機構を提供している。また、生成された回路の規模、効率よい動作の指標となるクロックサイクルごとの動作の実行時間とクロックとの比といったさまざまな品質指標を提示する機構をもつ。これらの機構により、設計者は回路のボトルネックを容易に発見でき、適切な回路修正が可能となる。

このように、手動と自動をミックスした設計手法により、手設計した回路と同等な品質の回路を短期間で設計することができる。

3.2 バインディングアルゴリズム

前節で説明した設計手法を提供するためには、まったく

新しいタイプのアルゴリズムが必要となる。そこで、アーバイン校との共同研究として当社独自のバインディングアルゴリズム⁽²⁾の組込みを行った。このバインディングアルゴリズムの特長は、設計者が意図する回路構造をテンプレートとして与え、この回路テンプレートを参照してバインディングすることである(図6)。

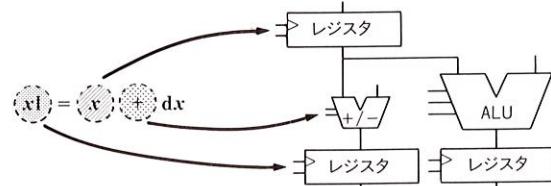


図6. 回路テンプレートを利用したバインディングアルゴリズム ビヘイビア記述中の一連の動作が回路テンプレートの回路部品、結線で実行できるようにバインディングする。この例では、レジスタと加減算器間の結線を利用するように、変数 x , $x1$, 加算がバインディングされている。

Binding algorithm with circuit template

回路テンプレートは回路構造を記述したもので、回路部品の種類と個数の情報だけでなく部品間の結線情報も含んでいる。ISEでは、フロアプラン画面を利用して回路テンプレートを入力する。

このアルゴリズムは、ビヘイビア記述中の各動作のデータ依存性を解析し、一連のデータ処理を抽出する。この一連のデータ処理を実行できる回路を回路テンプレート内で探索する。

一連のデータ処理を実行するためには、各動作に対応した回路部品間の結線が必要である。また、異なるクロックサイクルで実行される動作および互いに排他的な条件のもとで実行される動作は、同一の回路部品と結線を共有でき、この点を考慮して回路規模の増大を防いでいる。このように、ビヘイビア記述中の動作を回路テンプレート上で実現し、回路を生成する。回路テンプレートに含まれる回路部品では不十分な場合、必要最小限だけ新たな回路部品と結線が付加されるという特長がある。

このアルゴリズムにより、設計者は回路テンプレートを与えることによってバインディングをコントロールでき、設計者の意図した回路に近い回路が生成できる。したがって、設計者は生成される回路を予測できるため、回路修正が容易であり、さらに、意図した回路に近い回路が生成されるため、回路修正が最小限ですみ、回路テンプレートを与えない場合と比べていっそうの設計期間短縮が可能となつた。

ISEを用いた設計とビヘイビア合成の自動化アルゴリズムによる設計との比較のため、SRA (Square-Root Approximation)⁽³⁾の設計を例として取り上げる。ISEによる設計の詳細は文献(1)に説明されており、自動化アルゴリズムとしてビヘイビア記述中の各動作を実行可能なクロックサイクルで直ちに実行するようにスケジューリングするアルゴリズム、回路部品をできるかぎり共有するバインディングアルゴリズムを仮定している。

設計結果を表1に示す。この表では、回路のデータ処理部分の機能素子、記憶素子、接続素子だけを比較している。各素子の大きさはあらかじめ与えられており、それらの値

表1. ISEとビヘイビア合成の自動化アルゴリズムによる設計の比較
Example of design

	ISE	自動ビヘイビア合成
機能素子 (μm^2)	588,764	720,092
記憶素子 (μm^2)	298,944	149,472
接続素子 (μm^2)	207,648	237,600
合計 (μm^2)	1,095,356	1,107,164
クロック サイクル数	7	8

を用いて回路規模を計算している。自動化アルゴリズムによる設計では記憶素子が少なくなっているが、これは記憶素子の共有化のためであり、その副作用として接続素子が多くなっている。また、同じクロック周波数のもとでSRAが動作を完了するのに必要なクロックサイクル数は、ISEによる設計では7、自動アルゴリズムによる設計では8である。

ISEを用いた設計では、品質指標を確認しながらスケジューリングを変更できるため、自動アルゴリズムによる設計より少ないクロックサイクル数を実現できた。データ処理部分の回路規模とクロックサイクル数の比較からISEによる設計が有効であることがわかる。

4 あとがき

大規模システムを短期間に設計するために、ビヘイビア合成を用いた設計手法が提案されてきている。完全自動なビヘイビア合成を用いた設計手法では、手設計の回路と同等な回路規模、性能の回路を設計することは難しい。この問題を解決するための一つの方法として、インタラクティブなビヘイビア合成を用いた設計手法がある。インタラクティブなビヘイビア合成環境では、設計者にビヘイビア合成の処理に介入する手段を提供する。アーバイン校との共同研究で、この合成環境に回路テンプレートを利用するバインディングアルゴリズムを組み込み、インタラクティブなビヘイビア合成環境における設計期間短縮の見通しがついた。

謝 辞

この研究はカリフォルニア大学アーバイン校との共同研究で行った研究結果であり、ご指導・ご協力いただいた同大学の Prof. D.D. Gajski, Dr. V. Chaiyakul, Dr. T. Hadley, Dr. H.-P. Juan の各位に感謝する次第である。

文 献

- (1) D.D. Gajski, et al: Interactive Behavioral Synthesis, Proceedings of the Synthesis and System Integration of Mixed Technologies, SASIMI'96, pp. 117-124 (1996)
- (2) 増田篤司, 他: 探索空間を縮小する高位合成手法, 東芝レビュー, 50, 6, pp.465-467 (1995)
- (3) D.D. Gajski: Principles of Digital Design, Prentice Hall (1997)



今井 浩史 Hiroshi Imai, Ph.D.

マイクロエレクトロニクス技術研究所 システムLSI技術研究所主務、学術博。大規模論理LSI設計用CADの研究・開発に従事。情報処理学会会員。
Microelectronics Engineering Lab.