

DAC 報告 - 論理関係を中心に -

藤田 昌宏
富士通研究所

1992年6月8-12日に米国アナハイムで開催された第29回 DACに参加する機会を得た。
この会議での発表の中から、論理関係を中心に報告する。

A Report of DAC
Masahiro FUJITA
FUJITSU LABORATORIES LTD.

This report briefly introduces the papers in logic synthesis and verification presented at IEEE/ACM 29th Design Automation Conference which was held on June 8-12 at Anaheim, CA, USA.

1. 概観

第29回となるDesign Automation Conferenceは、CAD関係の最も人の集まる国際会議であるが、その理由は何と言っても、CADベンダーを中心とした展示である。展示、それによくできたチュートリアルによって活気づいていると感じられ、テクニカルセッションは、相対的に活気がなかった。この理由の1つには、一般に大きな会議では議論がしにくいという面もあるが、もう一方で、ほとんどの良い論文の最初の発表は既にワークショップなどで行なわれており、普段活発な質問をする聴衆は既にその内容を知っているという場合が多いことも否めない。また、全体的に従来手法のちょっとした改良という話が、少なくとも論理関係では多かった。

下に、論理関係のセッションの一覧を示す。

分野	セッション番号	セッション名
高位レベル合成：	7	Scheduling and Allocation
	11	Synthesis Systems and Representation
	14	System-Level Synthesis
	18	Allocation & Binding
	39	Advances in High-Level Synthesis
論理合成：	3	Two Level Logic Synthesis
	15	Performance Issues in Logic Synthesis
	23	FPGA Synthesis
	27	Multi-Level Logic Synthesis
	30	Technology Mapping in Logic Synthesis
	35	Sequential Logic Synthesis
論理検証：	38	Automated Approaches to Formal Verification of Hardware
タイミング解析：	25	Timing Optimization and Verification
	34	Path Delay Analysis
テスト	2	Test Generation
	10	Delay-Fault Testing
	17	High-Level Test Generation

全体的に、論理合成関係の研究に昔ほどの勢いが感じられない。これは最近の傾向で、'85年ごろから盛んになった論理合成の研究もようやく実用レベルに達して、研究としては落ち着いてきたようである。言い換えると、ブレークスルーと感じさせられるような結果は容易なことでは得られない程度には、熟してきている。もちろん課題は今でも種々残されており、今後も地道な研究は必要かつ重要であるが、他の手法を圧倒するような強力な新手法という形になるには、相当な進歩が必要なものが多い。このようななかでも、フィールドプログラマブルゲートアレイ (FPGA) の合成や、動作やタイミングの最適化/解析に関する研究は、現在も盛んに行なわれ、発表されている。また、2、3年前に比べてテストやレイアウトの研究が活発になってきているようである。特にテスト関係では、順序回路のテスト生成と遅延故障に対するテスト生成に関する研究が活発に行なわれており、この状況はしばらく続くようである（これは、決め手となるような手法はまだ確立されていないとも取れる）。レイアウトに関しては、タイミングを考慮したフロアプラン/配置手法、多層チャンネル配線手法、敷詰め型ゲートアレイの配置/配線手法等に多くの発表があった。

2. 論理関係の興味を引く論文

Implicit and Incremental Computation of Primes and Essential Primes of Boolean Functions, by O. Coudert and J.C. Madre (BULL)

これは、2分決定グラフを利用して積項の集合を表現する手法を新たに開発し、積和形論理式から主項と必須項をすべて求めることに応用した報告である。2分決定グラフを利用しているため、従来手法では取り扱えなかったような多数の主項を含む論理式も扱えている。この論文はかなりインパクトのあるものであり、既に他の会議で発表済みでもあるため、既にこの論文で導入された手法を元にした積和形論理式最適化手法の研究が他の研究者によって進められており、従来に比べかなり大きな論理式の最適化にも成功している。

Specification Partitioning for System Design, by F. Vahid and D.D. Gajski (UC Irvine)

これは、高位レベル合成を行なう前にチップ分割などを行なう手法を述べたものであり、高位レベル合成の実際の運用では極めて重要になると考えられるテーマである。特に動作記述自体を分割しようとしている点が新しく興味を引くが、紹介された手法の本当の評価はこれからである。

An Improved Synthesis Algorithms for Multiplexor-Based PGA's, by R. Murgai, R.K. Brayton, and A.L. Sangiovanni-Vincentelli (UCB)

最近どの学会でも活発に発表されているフィールドプログラマブルゲートアレイ用の合成に関する論文の1つである。発表者は、UCBにおいてFPGA用合成に関する研究の中心となっている人であり、合成に関する理解は深い。一般に、FPGAの合成では、各セルが実現できる論理関数の種類が極めて多いことが多く、結果的に従来のようなライブラリを個々にもち、回路パターンでマッピングするアプローチは適切ではなく、論理関数を考慮したものを使うことが望ましい。本論文では、対象を絞ることにより、この論理関数を考慮するマッピングを高速に行なう手法を提示し、その有効性を実験により示している。

Efficient Sum-to-One Subsets Algorithm for Large Optimization, by C.K. Chen and M. Fujita (Fujitsu)

著者が従来から開発している、回路の構造を大きく変えられる回路変換を重視した多段論理回路最適化手法における、新しい回路の再構成手法の提示し、その有効性を実験により示している。多段論理回路最適化はここ数年の進歩が激しかったが、ここにきて少し飽和してきている。これからさらに進歩するためには、従来に比べBoolean Relationなどかなり難しい問題を効率良く解かなければならず、当面は地味な研究になる可能性が高い。

Recurrence Equations and the Optimization of Synchronous Logic Circuits, by M. Damiani and G. De Micheli (Stanford)

その地味な研究をしているのがこの論文である。順序回路を最適化するためには、ドントケアの表現としては従来最も広いと考えられていたBoolean Relationでも完全にはドントケアを表現できず、さらに工夫が必要なことを示し、その形式化と解法例を述べている。まだ、問題が形式化できた程度の段階なので、評価はこれからであるが、問題が難しいだけにすぐ広まるという手法でもない。

A New Model for Improving Symbolic Product Machine Traversal, by G. Cabodi, P. Camurati, F. Corno, S. Gai, P. Pronetto, and S. Reorda (Politecnico di Torino)

これは、2つの順序回路の比較を行なうような検証におけるちょっとしたアイデアに関する発表である。2つの順序回路の間で状態割付の対応が取れば、実は組み合わせ回路部分の比較のみで検証できる。本論文は、状態割付の対応が完全には分からないが、部分的には分かっている場合、それを利用することで、検証時間を数倍高速化できることを述べている。通常2つの回路を比較する場合には、両者はまったく異なる構造をしていることはあまりないと考えられ、状態割当の対応が部分的には分かるとする仮定はかな

り現実的である。

Inductive Verification of Iterative Systems, by J.K. Rho and F. Somenzi (Colorado Univ.)

繰り返し構造を持つハードウェアは多いが、従来の検証手法はこの繰り返し構造をうまく利用できていない。繰り返し構造は、帰納法的な証明がうまくいきそうであるが、それを実際に示した論文であり、今後この種の研究は活発化すると考えられる。

3. チュートリアル

DACの1日間のチュートリアルは全部で9つあり、その内の"A Practical Introduction To Formal Hardware Verification"に参加したので、簡単にまとめておく。

まず、Carnegie Mellon大学のProf. Clarkeと彼の学生が形式的検証、特にモデルチェッキングと呼ばれる手法の基礎的な説明を行なったあとで、実際の設計に適用した例がインテル、ヒューレットパッカード、シーメンス、富士通の各研究者から報告された。また、少し異なる手法についても、京都大学とスタンフォード大学から説明があった。検証の実例に関しては個々の例によっていろいろと違いはあるが、まとめると次のようになる。

- ・特に設計者の意図しないような条件の時に活性化されるバグの発見に有効である。例えばマルチプロセッサ間のバス制御を行なうハードウェアにおいてデッドロックを引き起こす条件があるか、という検証に対して設計者が反例を見つけだすことは非常に難しい。また、複数の有限状態機械が互いに通信しあうようなモデルでは可能な状態の組み合わせを列挙することは不可能に近い。
- ・実際の回路をそのまま検証できることはまれで、何らかの手段で元の回路をモデル化して簡単化したものを検証している。具体的にはデータのビット長を短くするとか、割り込み処理部を削る等している。
- ・効率を考えると設計者自身が検証を行なうべきである（検証を行なう人が設計者から一々情報をもらって検証するよりも）。ただし、現時点ではハードウェアの設計者にTemporal Logicで仕様を記述させることには無理があると思われる。形式的検証を実際の設計支援に導入して行くためにはこの点を解決しなければならない。

結論として、設計者が実際に使う上では問題点はいくつか残されているが、さまざまな工夫で実際の設計の検証に役立つ場合が少なからずあることが実証されたことも事実である。なお、このチュートリアルはほぼ満員の盛況であり、日本と比べその関心の高さに驚いた面もあった。