

大規模電子計算機向け高レベル論理設計手法の確立

土屋 洋次 森田 正人 碓谷 幸夫 鶴身 栄一 森 照夫 柳田 友厚

(株) 日立製作所

論理装置が大規模・複雑化し、更にその100%LSI化が進む中で、いかに誤まりなく短期間で設計を終了するかが最大の課題となっている。

この課題を解決する一手法として、論理設計の早期段階より自動設計を適用することが考えられ、研究も広く行なわれている。

自動設計の適用範囲を広げていく際、論理設計者の従来の生活または思考形態から大きく逸脱した方法では、論理設計者が馴染まず、自動設計を急激な勢いで立上げることは不可能に近い。

そこで今回、この点を十分に考慮して図形式高レベル論理表現：FL(Function Logic Diagram)を支援し、目的とする回路系の詳細論理は、論理設計者の意図通りに効率よく、しかも確実にFLから自動生成するシステムを開発した。論理設計者は、全設計工程を通じてFLだけを意識すればよい。

ESTABLISHMENT OF HIGHER LEVEL LOGIC DESIGN FOR VERY LARGE SCALE COMPUTER

Yooji Tsuchiya Masato Morita Yukio Ikariya
Eiichi Tsurumi Teruo Mori Tomoatsu Yanagita

Kanagawa Works, Hitachi Ltd.

1 Horiyamashita, Hadano-shi, Kanagawa-ken, 259-13 Japan

23RD ACM/IEEE DESIGN AUTOMATION CONFERENCE

Session Number 20.3

1. はじめに

論理装置が大規模・複雑化する一方、LSI技術の進歩により100%LSI化した論理装置が出現している。

LSIを多用する論理装置を短期間に低価格で、しかも誤まりなく開発することが、益々重要になっている。LSIを製造してから論理不良が発生すると、再製造するのに2~3ヶ月を要することになるので、何とせよでも論理設計の早期段階で、誤まりを取り切ることが必要である(図1)。この問題解決のためDAシステムへの入力レベルを高水準化し、機能レベルの段階から設計支援する新しいDAシステムを構築した。

本DAシステムにおいては、論理設計者は、FL(Function Logic Diagram)を用いて論理設計を進める。FLは、ALDL(A Logic Description Language)と呼ばれるテキスト形言語で記述され、設計データベースに格納される。このデータベースを元に論理シミュレーションが行われる。

ディレイチェックと実装設計を進めるには、使用する回路系素子で表現された詳細論理記述が必要である。この詳細論理記述は、BLDL(Basic Logic Description Language)と呼ばれるテキスト形言語で記述され、設計データベースに格納される。BLDLは、また詳細論理図(LD)を生成するのにも用いられる。

論理生成は、ALDL記述からBLDL記述を生成する。この自動化により、高レベル論理記述を用いる論理シミュレーションと詳細論理記述を用いるディレイチェックの平行検証が可能になり、設計検証効率を大幅に向上できる。このため、論理生成は高レベル論理記述から、高速にしかも設計者の意図した通りの詳細論理を提供できなければならない。また、大形機の設計では避け難い、頻繁な設計誤まりの修正に耐える性能が要求される。

本論文では、全ての設計工程においてFLだけを用いて論理設計を進めるための設計方法論とM-68Xでの適用実績について述べる。

2. 設計方針

全ての設計工程において、FLだけを用いて設計者の意図する詳細論理を効率よくしかも確実に作るためには、機能論理から詳細論理の生成過程で以下の点が考慮されなければならない。

(1) 人手並みの論理生成性能

ディレイが厳しく、しかも設計量が膨大な大形機の設計では、詳細論理の生成結果を会話的にいちいち確認するのは、実用的ではない。詳細論理の生成結果を予測しながら機能論理を記述し、予測通りの詳細論理を自動的に得ることができれば、確実に手戻りのない設計が可能に

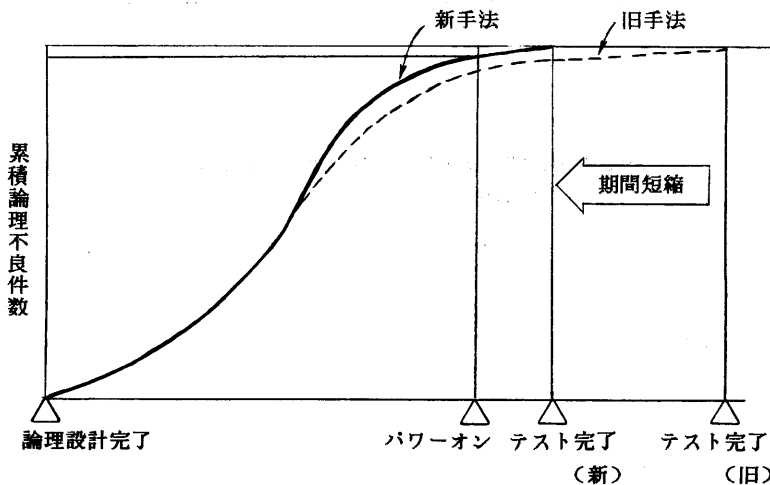


図1. 論理不良の抽出課程

なる。このためには、設計者の意図した詳細論理を確実に生成するだけの性能が要求される。

設計者は、機能論理設計を行うとき、構造表現図面であるFL記法のシンボルすなわちモジュールを思考単位としている。モジュールとモジュールの接続関係については、信号名を与えることで設計者の意図を明示している。このため、論理生成の目標は、設計者の思考単位であるモジュール単位に、機能論理から人手並みの詳細論理を生成することである。

人手並みの論理生成を実現するために考慮すべき項目として、物量制約を満足するためのゲート数、ディレイを抑制するためのゲート段数、消費電力などの電気的制約またはLSIの内外セル区分といった物理的制約からくる使用部品種がある。これら諸条件を満足するためには設計者の思考仮定を定式化せねばならない。もし、定式化できないものがあれば、生成のための設計補助情報としてFL記法に設計者の意図の表示をできるようにしなければならない。また、逆に、生成のルールを確立してこれを公開すれば、設計者は詳細論理を予測しながら機能論理設計を行うことができる。この場合、この生成ルールは人手に比べて孫色がなく、設計者に受け入れられるものでなければならない。

(2) 読み易く高密度なLDの提供

ディレイチェックによる設計検証、LSIやプリント基板の実装設計を行うには、目的のテクノロジーの素子で表現された、読み易く高密度なLDが提供されなければならない。

見易さは、高度の主観的メジャーであるため完全な解は期待できない。また、図面の密度は見易さと深く関連しており、単に高密度であればよいというものではない。

読み易く高密度の図面を得るには、ページ間クロス最小化、関連論理素子のグループ化、時系列又は、機能別の論理の流れ(ページ出現順序)といった、論理設計者の経験則が十分生かされねばならない。

(3) 機能論理と詳細論理の等価性保証

論理設計者の思考を機能論理に止め、詳細論理を元に目的物を正しく製造するためには、機能論理と詳細論理の等価性が常に保証されなけ

ればならない。機能論理から詳細論理が自動生成された後、等価性が保証されなくなるのは、機能論理又は詳細論理の一方が人手で更新される場合である。この人手更新と同期して、両者の等価性をチェックまたは、自動的に更新する手段が提供されなければならない。

(4) 処理の高速化

大形機の設計を考えると、(1)~(3)の処理は、可能な限り高速に処理されなければならない。

設計検証は、機能レベルの論理シミュレーションによる論理検証と、詳細論理レベルのディレイチェックによるタイミング検証の2つがある。これらの検証は、全く独立に行われるわけではなく、設計工程の進度に合せて平行して行われる。機能論理の論理検証と詳細論理のタイミング検証を等価論理で確実に行わせるためには、機能論理から詳細論理を高速に生成する手段が必要である。この生成の処理高速化によって、論理検証とタイミング検証の両立性が保証される。

3. 論理生成システムの概要

3.1 モジュールベース論理生成

図2に概要を示す。機能記述を入力とし、詳細記述を生成するとともに、LDのための論理シンボルの位置決めも行っている。これにより機能論理を用いず、直接詳細論理を設計して詳細記述を行った場合と、全く同じ記述を提供することができる。

入力となる機能論理は、モジュールと呼ぶボックスのネットワークで表現される。そのモジュールの表現方法には3通りある。

- ・汎用モジュール：
ブール式または真理値表で組合せ論理を表現したもの
主に制御系の記述に使われる。
- ・標準モジュール：
演算器、マルチプレクサ、ラッチ等のマクロ回路を表現したもの。
主にデータバスの構造記述に使われる。
1つまたは複数の基本モジュールの組合

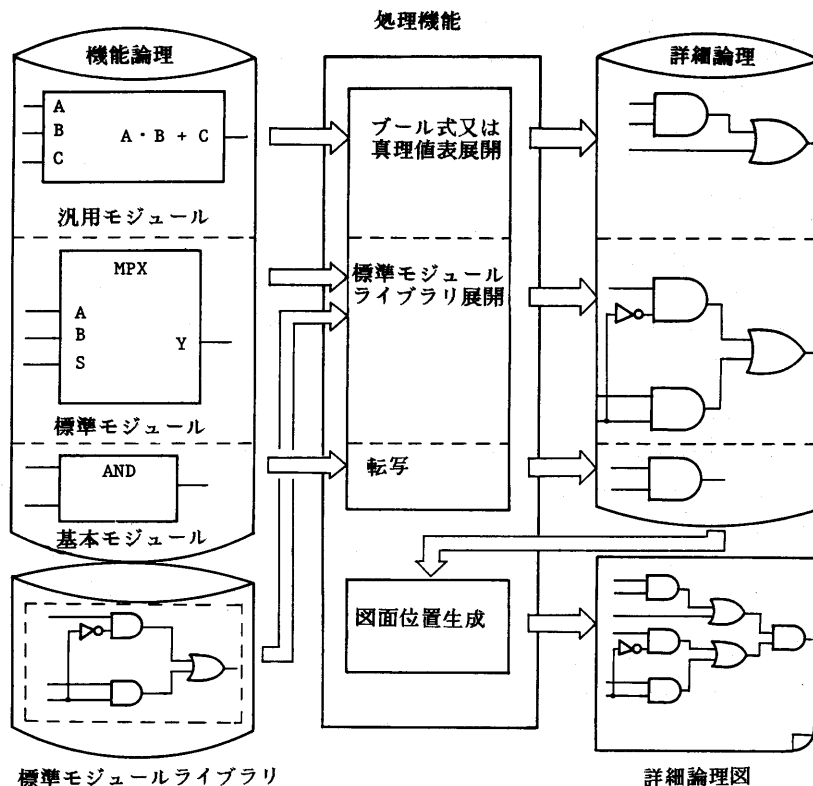


図2. 論理生成の概要

せ論理であり、標準モジュールライブラリに予め定義する。

・基本モジュール

論理設計の最小単位となる構成要素。

詳細論理の構成要素でもあり、テクノロジー制約の厳しい部分に使われる。

以上3通りのモジュール表現に対応して、記述特性を考慮して、3通りの生成機能を用意した。

(1) 汎用モジュールからの生成

FLだけを用いて論理設計を進めるためにはブール式または、真理値表の記述レベルから、詳細論理が予測できなければならない。このため、入出力信号の極性と、ブール式または真理値表の演算順序・変数の数などで、目的回路系の素子へのマッピングを一意的に行うアルゴリズムを用いている。[1]

このアルゴリズムは、設計者にとって自然なもので、設計者はこのアルゴリズムを知って、詳細論理を思い浮かべながら、ブール式または真理値表を記述することができる。

図3に、本アルゴリズムによるECLテクノロジーを用いた生成の例を示す。特徴的なことは、モジュールの入出力信号に、物理概念である極性を明示していることである。この例ではパラメータA、CとO1が負極性であり、B、DとO2が正極性である。Box内のブール式は正論理で記述してある。図3の(A)に示す同一形式のブール式でも、入出力信号の極性により(B) Dotted-ORを持つ1つの素子(C) Wired-ORを利用した2つの素子へと最適な生成結果は異なる。

本アルゴリズムは、原理的には何段のゲート生成でも可能である。しかし、ディレイが厳しく、しかも設計量が膨大な大形機の設計では、無雑作にブール式又は真理値表を書き、詳細論理の生成結果を会話的にいちいち確認すること

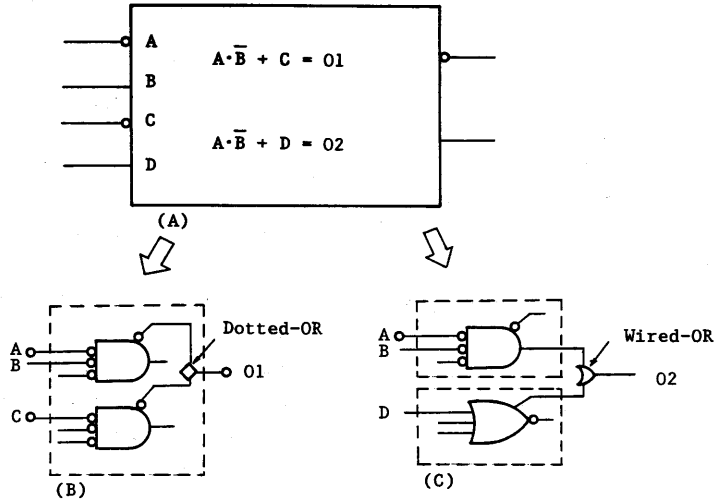


図3. 汎用モジュールからの論理生成の例

は実用的ではない。論理設計者は、入出力信号に極性を与えることで、極性伝搬を思い浮べて、ゲート数・ゲート段数を予測できる。詳細論理の生成結果を予測しながらブール式または真理値表を記述していく方法は、確実に手戻りのない設計方法である。

(2) 標準モジュールからの生成

標準モジュールは、1～数十ゲートの回路を1つの設計単位として取り扱えるようにしたものである。生成のために、各標準モジュールに対応する詳細論理を予めライブラリに登録しておく。生成時には、テクノロジーコードとマクロコードをキーにして、ライブラリを参照し、標準モジュールを詳細論理に置換える。ライブラリ方式では、更新の容易性と使い易さが考慮されねばならない。

ライブラリの内容は、FL図面上の論理シンボル定義と基本モジュールの組合せで表現された詳細論理の定義から成る。ライブラリは、FL設計データベースを作るのと全く同じ方法で作成できる。すなわち、基本モジュールで表現したい詳細論理を記述し、FL図面上の論理シンボル定義を記述すれば、それを新しい標準モジュールとしてライブラリに追加することがで

きる。このように、ライブラリは論理設計者により容易に作成できる。

設計者は、FL図面を記述するとき、標準モジュールの種別を明記しなければならない。このため、種別が多いと選択ミスを起こす可能性が高くなる。そこで、ライブラリに登録されている種類よりも、実際には多くの生成形を提供できるように、後述するゲート回路特性利用・不要ゲート削除などの交換機能を設けている。この交換機能により、使い易さが大幅に向上している。

(3) 基本モジュールからの生成

基本モジュールからの生成では、単にコピーするだけでよい。

3.2 ゲート論理特性の利用方式

論理生成を実用的なものにするためには、ゲート数・ゲート段数の最小化だけでなく、ファンアウト・消費電力等の物理的諸条件を満足しなければならない。このため、機能論理記述より、自動または人手指示により、所定の物理的条件を認識し、ゲートの回路特性(駆動能力、ディレイ等)を有効利用して、条件を満足する論理生成を行っている。

同一論理機能で回路特性の異なるゲートとして、以下に一部の例を示す。

- ・内外ゲート：
 - LSIの内部で閉じるゲートと、LSIの入出力ピンに接続されるゲート
- ・パワー制御ゲート：
 - ファンアウト数を多くとるために駆動能力を高めたゲート
- ・スピード制御ゲート：
 - ゲートディレイは大きい、消費電力を低く抑えたゲート

このゲート回路特性の利用を汎用モジュールと標準モジュールの生成過程で考慮している。汎用モジュールの生成過程では、はじめから、当該回路特性のゲートを用いてゲート論理を生成する。

標準モジュールでは、標準的な回路特性を持つノーマルゲートにより、ゲート論理の基本形を定義してライブラリに格納しておく。このた

め、物理条件が満たされない場合、不適切なゲートを確認し、これらのゲートを回路特性の異なるゲートと置換する。具体例を図4に示す。図4では、LSIの内部ゲートを用いてゲート論理がライブラリに定義されていて、生成時に1ゲートが外部ゲートに置換される例である。

3.3 不要ゲートの削除方式

標準モジュールは、汎用的なマクロ回路として用意されているため、実際に使われるときにその機能を全て使うとは限らない。このため、これらの機能の一部を使用する場合、その不要部分を削除し、実装エリアの有効活用を図らなければならない。

標準モジュールの未使用出力端子に着目し、それに影響するモジュール内のゲートを削除する例を図5に示す。出力端子からの不要ゲート削除は未使用出力端子から信号をバックトレースして行われる。その処理手順を以下に示す。

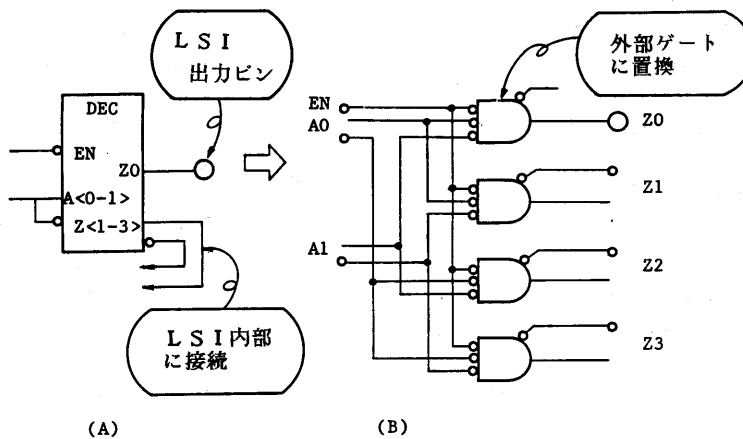


図4. 標準モジュールからのゲート回路特性利用の例

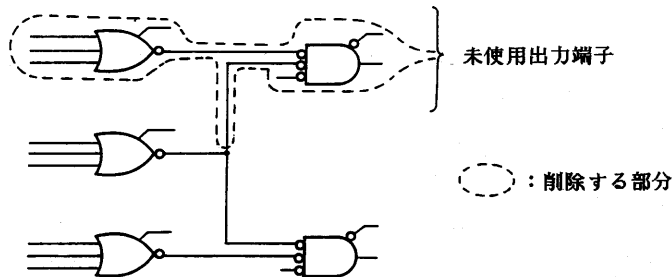


図5. 標準モジュールの不要ゲート削除の例

- (1) 全ての出力ピンが未使用であるゲートを、トレースゲートとして取出す。
- (2) トレースゲートの全ての入力ピンに対して、1ピンずつ未使用ピン扱いにしながら、次の条件(a)を判定する。条件成立時は(3)へ行く。
全入力ピンの処理が終わったら(4)へ行く。
- (a) 該入力ピンが接続されていたネットに、他にシンクゲート(入力ピン)がなくかつソースゲート(出力ピン)がある。
- (3) 該入力ピンが接続されていたネットにつながるソースゲートの出力ピンを未使用扱いとする。もし、該出力ピンを含むゲートの出力ピンが全て未使用なら、新たにトレースゲートとして登録する。
- (2)の続きの処理へ戻る。
- (4) 該トレースゲートを削除する。

以上、トレースゲートが存在する間、(1)～(4)を繰り返す。

3. 4 LDのシンボル配置方式

論理生成の結果は、BLDLで記述され、設計データベースに格納される。この設計データベースに基づき、ディレイチェックと実装設計が行われる。BLDLは、テキスト形言語であるが、論理設計者が必要に応じて詳細論理の生成結果を論理図で確認できるよう、生成過程で論理図シンボルの配置を行っている。論理シンボル間の配線は、LDのLBP(Laser Beam Printer)図面出力で別途行う。

論理図のシンボル配置の方法としては、LSIやプリント板分割と同様に、ゲート論理レベルの1階層に全部展開した後に、シート単位に区切る方法が、比較的単純な処理方法として考えられる。しかし、この方法では、論理の機能的まとまりや論理の流れといった論理設計者にとっての見易さを考慮できないため、次の方法とした。

- ・論理の機能的まとまりは、モジュールの区切りを保存する。

- ・論理の流れは、機能論理図の流れを保存する。

標準モジュールでは、ライブラリ登録時に人手で詳細論理のシンボル相対位置を定義する。汎用モジュールでは、詳細論理を生成するとき、シンボルの相対位置を自動決定する。このようにして、モジュール単位にモジュール内の詳細論理のシンボル位置を見易く、かつ高密度になるよう予め決定する。次に、機能論理図のモジュール相対位置(ページ順序、同一ページ内上下左右)を保存しながら、モジュール毎にそのモジュール内詳細論理のシンボル相対位置を絶対位置に変換していく。このようにして、あたかも辞書を索引するかの如く、詳細論理図とモジュール単位に対応のとれる詳細論理図を提供している。

4. 設計環境への統合

FLだけを用いて設計者の意図した詳細論理を間違いなく作るため、全設計工程に渡り機能論理と詳細論理の等価性を保証する手段が必要である。このため、論理比較システムと、論理生成システムの部分生成機能を開発した。[3]

論理比較システムは、生成後の詳細論理に、自動化機能を補完するため、人手で実装情報を追加・更新する場合の、ミス混入防止を目的としている。このため、機能論理と詳細論理を論理的に比較し、不一致箇所をリスト出力する。設計者は、詳細論理を更新する度に、本システムで検証を行い、誤った設計作業の進行を防止できる。

実装設計開始後の機能論理の変更に対し、再度生成をやり直したのでは、詳細論理に付加された実装情報が失われる。このため、機能論理の変更のあった部分のみ、部分的に詳細論理を再度生成し、旧詳細論理とマージする部分生成機能を設けた。部分生成機能では、機能論理の変更箇所を機能論理と詳細論理を比較することで認識しているため、特別に情報を与える必要はない。設計者は、部分生成機能を用いて作り直した詳細論理の実装情報の不足分のみ補うだけでよい。

5. 評価

論理生成システムは、超大形機M-68Xに搭載された2000ゲートのECL-LSIのすべてに適用された。

表1に生成結果の諸元、表2に処理時間を示す。

表1に示すように、FLはLDに比し、約4倍コンパクトである。また、モジュール別の生成ゲート数比率で分るように、標準モジュールの利用比率が圧倒的に高い。ゲート回路特性利

用と不要ゲート削除の比率は10%以下であるが、生成を実用レベルのものにするには、必須である。対人手性能比は、ゲートレベルで人手設計と比べたものであり、1.3%のゲートが自動と人手で違っている。

この1.3%の違いの例を図6に示す。このように、複合ゲートの使い方に差があった。しかし、ゲート数・ゲート段数は全く同じであり、実質的には、100%人手と同じであった。

表2に示すように、処理時間は実用範囲に収めることができた。

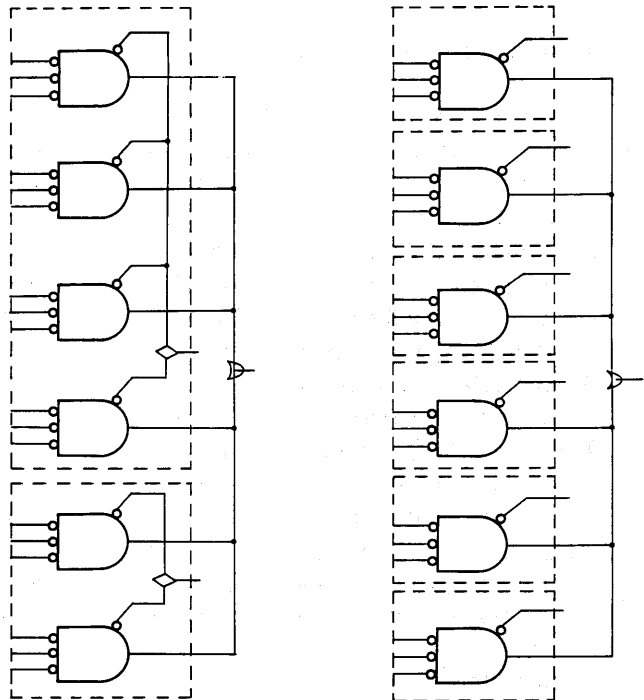
表1. 生成結果の諸元

図面枚数比		モジュール別生成ゲート数比率			ゲート回路 特性利用率	不要ゲート 削除率	性能比 (対人手)
FL	LD	汎用	標準	基本			
		%	%	%	%	%	%
1	4	19	79	2	9	4	98.7

表2. 処理時間

(M-200H換算)

論理生成		論理比較
全部生成	部分生成	
秒	秒	秒
55	143	88



(A) 人手設計

(B) 自動生成

図6. 自動生成結果の人手設計と異なる例

6. 結言

機能論理から製品品質レベルの詳細論理を自動生成するシステムを開発し、M-68Xの開発に100%適用した。

全ての設計工程において、FLだけを用いて設計者の意図する詳細論理を効率よく、しかも確実に生成する方法の開発に成功した。このために、人手並みの論理生成技術と、機能論理と詳細論理の等価性の保証技術を確立した。

今後、今回の開発経験を生かし、より高位概念からの生成を目指す予定である。

7. 謝辞

本研究の推進に当り、終始適切な御助言、御支援を頂いた大野泰廣博士、久保隆重氏、泉千賀彦氏、および三善正之氏に感謝致します。

8. 参考文献

- [1] T.Shinsha et al., "POLARIS ; Polarity Propagation Algorithm for Combinational Logic Synthesis", 21st DA Conf., 1984, pp.322-328
- [2] Y.Ohno et al., "Principles of Design Automation System for Very Large Scale Computer Design ", 23rd DA Conf., 1986.
- [3] T. Shinsha et al., "Incremental Logic Synthesis Through Gate Logic Structure Identification", 23rd DA Conf., 1986.
- [4] M.Miyoshi et al., "An Extensive Logic Simulation Method of Very Large Scale Computer Design", 23rd DA Conf., 1986.
- [5] R.Toyoshima et al., "An Effective Delay Analysis System for A Large Scale Computer Design", 23rd DA Conf., 1986.