SSI/MSI 論理より LSI 論理への論理の再構成手法†

田中 千代治†† 中村 俊一郎††
村井 真一†† 寺井 正幸†† 樹下 行三†††

論理装置のカスタム LSI 化に際し，既存の SSI/MSI の論理を LSI 論理に自動的に再構成することは設計ミスや LSI のデバッグの削除など開発期間の短縮に非常に有効である．本文では，元の TTL-SSI/MSI の NAND 系論理を NOR 系のマスタスライス LSI 論理に再構成する一般的な手法を述べ，この手法に基づくプログラムを作成し，ある機種に適用して非常に有効であることを実証した．

本手法は，再構成された論理を設計者が容易に理解できるよう元の論理回路構成を維持して再構成を行う方式であり，このため，各 SSI/MSI ごとに LSI で実現可能なライブラリを準備し，ライブラリによる置き換えを行い，その後，不要な端子や素子の削除，この置き換えにより生じる多くのインバータのリダクション等論理の簡単化を行う方法とした．

本手法では不要端子・素子の削除およびインバータリダクションが重要な部分であり，このため，これらのアルゴリズムを求めた．各種論理の適用結果，本文で述べる条件では，統計的に，元の回路の総ゲート数±20% 程度が LSI 論理のゲート数となることが解った．

1. まえがき

論理装置の LSI 化による利点は小型化，性能向上信頼性向上，価格低減などであるが，一方，従来の MSI, SSI による装置に比較し開発費の増大，LSI のデバッグの困難化，開発・製造期間の長期化などの短所を持っている．特に，LSI 計算機を実現するためには多品種の LSI を短期間に開発し，また，LSI 計算機のデバッグを効率良く行うことが必須であり，このため，LSI の自動設計システムの確立，能率のよい計算機デバッグ手法の開発が不可欠となっている．LSI 計算機の実現の方法として

1）既存の MSI, SSI で製品化された装置を LSI 化する．

2）MSI, SSI を使用してプロトタイプを試作し，デバッグ完了後 LSI 化を行う．

3）各種の論理シミュレータ[1)~3)]を用いて装置をデバッグし LSI 化を行う．

の3つの方法が考えられる．

上記において，1）の場合には LSI としての分割が考慮されていないこと，2）の場合にはプロトタイプを試作するために多大の費用と期間を必要とすること，3）の場合には，完全なデバッグという意味ではまだ不十分であり，特に大規模論理回路の場合には LSI にバグを持ち越す危険性が大きいこと，等の短所がある．しかしながら，1）の場合，装置はすでに開発を完了しているので，ほかの方法に較べて短期間に LSI 化を達成することが可能であり，これらの論理の変換または再構成を自動的に行えるなら短期間で大幅性能向上，小形化が達成可能となる．

上記1），2）に着目した研究は比較的少ないが元の論理から論理機能ブロックを見つけ出して LSI の論理素子に置き換える方法[4)]，ゲート単位に LSI の論理素子に置き換え以後簡単化を行う方法[5)~7)]等が報告されている．前者のようにブロック単位で LSI ゲートに置き換えていくだけの方法では，NAND 系から NOR 系へ変換する場合のようにインバータが多く発生する場合にこれを吸収できないという欠点がある．また後者の場合変換の過程で元の論理機能ブロックのまとまりを維持するのがむずかしく，変換前後の論理回路の対応がつけにくいこと，このため，クロックの同時性やクリティカルパスの検証が困難であること，また，LSI の種類により簡単化の規則が異なるため汎用性に乏しいという欠点があった．今回，これらの点を解決するため

（i）MSI 等の論理機能ブロックのまとまりを変

† Logic Reorganization from SSI/MSI Logic to LSI Logic by CHIYOJI TANAKA, SHUNICHIRO NAKAMURA, SHINICHI MURAI, MASAYUKI TERAI (Headquarters-Research and Development, Mitsubishi Electric Co.) and KOZO KINOSHITA (Faculty of Integrated Arts and Sciences, Hiroshima University).

†† 三菱電機(株)開発本部
††† 広島大学総合科学部

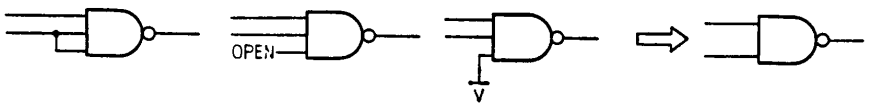

図 1 不要な入力端子の削除
Fig. 1 Elimination of redundant input pins.

換後も維持することにより，元の論理回路と変換後の論理回路の対応を付けるようにする．

（ii） インバータリダクションとマクロ内リダクションにより変換後の論理を最適化する．

（iii） 設計者は，プログラムの変換アルゴリズムを理解し，クリティカルパスやクロック等検証すべき個所をあらかじめ抽出し，元の論理と対応づけされた変換後の自動作図された図面上で検証を行う．また，必要により，タイミングを考慮した論理シミュレータで論理全体の検証をすることを目標として実用的な手法を考案した．これをプログラム化し，実用化した結果，設計人工の削減や開発期間の短縮，設計の不用意なミスの削減などに非常に役立つことがわかった．

本文では，NAND 系 MSI, SSI で作られた論理回路を NOR 系のマスタスライス LSI で実現可能な論理に再構成する一般的な手法とこれに必要なアルゴリズムを求め，これに基づく自動作図，論理シミュレータを含む自動設計システムの構成およびその適用結果について述べる．

2. 論理再構成の手法

2.1 問題の定義

対象となる論理回路をマスタスライス LSI で実現可能な論理に再構成することとは，対象論理回路の論理を変えることなく，LSI のファンアウト条件や素子の遅延時間を考慮し，対象論理回路をできるだけ少ない LSI の基本素子に置き換えることである．

本章では次の条件で論理を再構成する方法について述べる．

（1） 対象論理回路を MSI, SSI で構成されるプリント配線板とする．

（2） 再構成の方式を，MSI, SSI に対応した LSI で実現可能な素子で構成されたライブラリを準備し，逐一置き換えていくライブラリ方式とする．

（3） MSI, SSI を NAND 系 TTL とし，LSI の基本素子を NOR ゲートまたは NOR ゲートおよび OR ゲートとし，ワイヤード OR や AND も LSI の種類により可能とする．

2.2 手　法

論理の再構成の手順とその機能は次の通りである．

（1） 論理の LSI への分割

対象となる 1 ないし複数のプリント配線板の論理を 1 つのまとまりとし，次の操作を行う．

（a） 抵抗，コンデンサなどの非論理素子を削除する．

（b） 人手指定により，ROM, RAM 等のゲート論理への置き換えまたは外出しを行う．同様に人手指定により，必要に応じて回路変更を行うと共に，以後その部分に対しプログラムによる変更を禁止する．

（c） 通常，MSI, SSI を使って論理を構成する場合，図 1 に示すように個別ゲートに余分の入力端子があり，論理的に固定されている場合があるので，この余分な入力を削除する．

（d） 以上の操作により対象論理回路を LSI 論理に変換可能とし，この論理に対して LSI への分割を行う．

（a）～（d）の操作により 1 つの LSI へ変換されるべき論理が抽出される．

（2） ファンアウト調整

元の論理回路上で信号のファンアウト数が LSI で許される制限値を超えているものに対し，図 2 に示すようにインバータまたは 1 入力 OR を挿入しファンアウト数を制限値以内におさめる．このとき，フリップフロップや後述するマクロ素子の入出力端子に関しては，ライブラリに登録されているファンイン／ファンアウト数が参照され，ファンアウト調整が行われる．

（3） ライブラリによる置き換えと LSI マクロ展開

以上の操作により対象論理回路は，SSI 対応の個別ゲート，フリップフロップおよび MSI 対応のマクロ

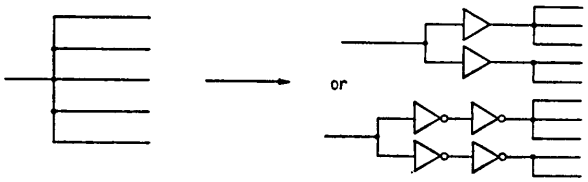

図 2 ファンアウト調整
Fig. 2 Adjustment of fan out.

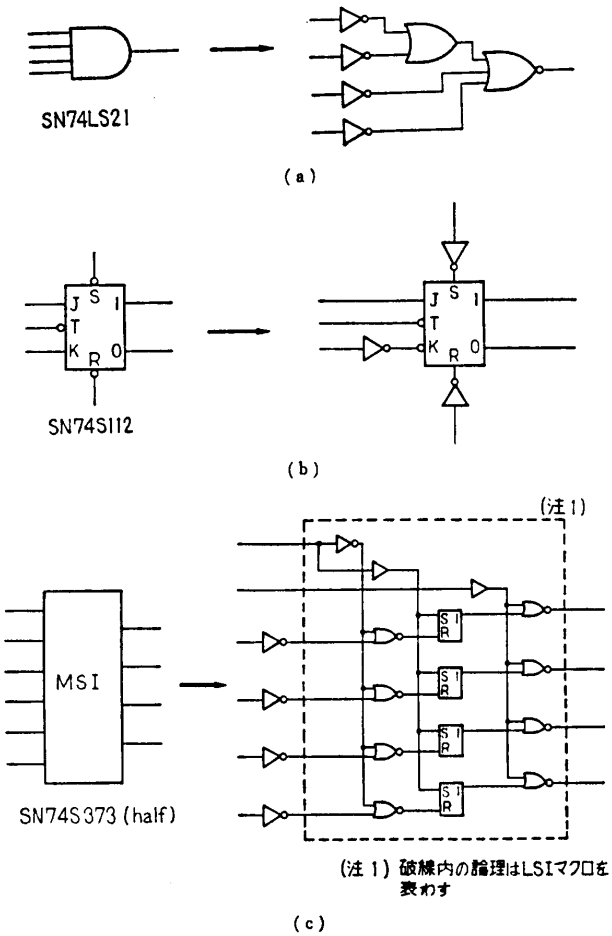

図 3 ライブラリによる置き換えと LSI マクロ展開
Fig. 3 Replacement by library and LSI macro expansion.

素子となり，これらの素子に対して次のライブラリによる置き換えおよびマクロ素子の展開を行う．

（a） 個別ゲートを LSI 上で使用可能なゲートに置き換える．

（b） フリップフロップを LSI 上で使用可能なフリップフロップとインバータに置き換える．

（c） MSI 対応のマクロ素子を LSI で実現可能なマクロ素子* とインバータに置き換える．

（d） この LSI マクロを LSI の基本素子に展開する．

これらの例を図 3 (a), (b), (c)に示す．

上記(b), (c)によりマクロ素子が原形に近い形で保存され，図面上で一つの箱として書くことが可能となる．また一般に，NAND, AND で構成される論理を NOR, OR で再構成するには NAND を NOR に，AND を OR に置き換え，すべての外部端子にインバータを付ければよいが，これらのインバータはここで外出しされ後述のアルゴリズムによりリダクションが行われる．

以上の操作により，元の MSI, SSI 論理は LSI の設計条件を満たして LSI で実現可能な回路に置き換えられたことになる．以下で不要な素子や端子および論理の簡単化を行う．

（4） LSI マクロ内リダクション

一般に，LSI マクロ素子は汎用的な素子として用意されているため実際の論理回路の中で使われるときその機能をすべて使用するとは限らない．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の削除を行う．この操作を LSI マクロ内リダクションと称しこのアルゴリズムを 2.3 で述べる．

（5） インバータ・リダクション

ライブラリによる置き換えにより多くのインバータが発生するが，ここではファンアウトの制限を守って論理を変えることなくインバータを必要最小限にすることを行う．このアルゴリズムを 2.3 で述べる．

（6） ゲートの圧縮

LSI 上で NOR ゲートおよび OR ゲートが使用可能な場合，(4) の LSI マクロ内リダクションによりゲートの入力端子数が減り，その素子がインバータとなる場合や図 4 に示すように次段のゲートに吸収可能となる場合がある．また，LSI マクロ展開によりインバータが挿入され NOR-INV, OR-INV が生じることがあり，これらは OR, NOR と等価となる．

このように LSI マクロ内リダクションおよびインバータ・リダクションの後，上記操作を行うことをゲートの圧縮と称しここでこの操作を行う．

（7） 入出力バッファの挿入

最後に LSI の外部入出力信号に対してバッファ回

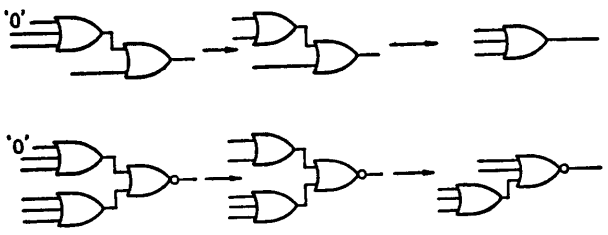

図 4 ゲートの圧縮
Fig. 4 Compression of gates.

* 以後このマクロ素子を LSI マクロと呼ぶ．

路を挿入し，論理再構成操作を終了する．

この一連の操作により元の論理構造を保存しながらLSIの論理に再構成されたこととなり，この（1）～（7）までの操作をこの順序で行うことにより，さらにLSI基本素子を削減する余地はなくなる．

2.3 アルゴリズム

（1） インバータ・リダクション・アルゴリズム

前述したようにNAND系の論理からNOR系の論理へ置き換えを行う場合，新たに多くのインバータが発生するが，これらを元からあったインバータをも含めて最小のインバータ数の回路網に置き換える操作が必要である．このため，**図5**に示すようなインバータのみにより構成される回路部分を抽出する．この抽出された回路に対するインバータのリダクション・アルゴリズムは次のようになる．

〔定義1〕 インバータの入出力をショートすると1つの信号となる論理回路を準信号系と呼ぶ．このとき，準信号系への入力となる点を始点，出力となる点を終点と呼ぶ．

〔定義2〕 準信号系に対し，ファンアウト条件を無視し，正相および逆相の信号から必要なファンアウトを取出した論理回路を準信号系の標準形と呼ぶ．

ここで，正相とは，始点と同じ論理値，逆相とは，始点と逆の論理値をとることを意味する．

準信号系とその始点・終点の例を図5に，これに対する準信号系の標準形を**図6**に示す．

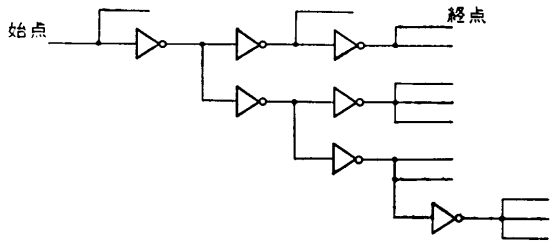

図5 準信号系
Fig. 5 Quasi-signal.

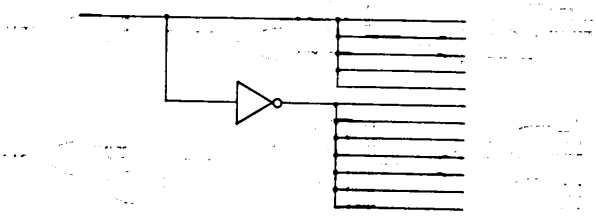

図6 準信号系の標準形
Fig. 6 A normalized form of the quasi-signal.

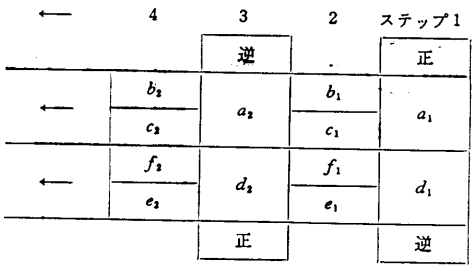

図7 漸化式による作表法
Fig. 7 A table generated by the recurrence equation.

以上の定義よりインバータ・リダクションとは，準信号系の標準形よりファンアウト条件を満足し，最小のインバータで準信号系を求めることとなる．

〔アルゴリズム〕

1 正相の終点の数を a_1，逆相の終点の数を d_1 とする準信号系が与えられ，このときのファンアウト許容数を F とすると次の漸化式により**図7**の表を作成する．

$$\left.\begin{aligned} a_i &= b_i F + c_i \\ d_i &= e_i F + f_i \end{aligned}\right\} \cdots\cdots (1)$$

$$\left.\begin{aligned} a_{i+1} &= b_i + f_i \\ d_{i+1} &= e_i + c_i \end{aligned}\right\} \cdots\cdots (2)$$

$i=1, 2, 3\cdots\cdots$

上式で c_i, f_i はそれぞれ a_i, d_i を F で割った余りである．この漸化式は，$a_i<F$ かつ $d_i<F$，または $a_i=F$ かつ $d_i=0$，または $a_i=0$ かつ $d_i=F$ のいずれかの条件が満たされた時点で終了する．

2 図7の各欄に対しステップ1より順に次の操作を行う．

1）初期値として，a_1 個の終点を a_1 の欄に，d_1 個の終点を d_1 の欄に置く．

2）a_i 個の準信号系を F 個ずつインバータでまとめ b_i 個の準信号系とし，これを b_i の欄に置く．残った c_i 個の準信号系を c_i の欄に移す．d_i の欄から同様に f_i, e_i の欄を作成する．

3）b_i の欄と f_i の欄の準信号系を a_{i+1} の欄に移す．同様に，e_i と c_i の欄の準信号系を d_{i+1} の欄に移す．

3 最終ステップにおいて，逆相側が0でないときには逆相側にインバータを付けこのインバータの入力と正相信号をまとめて始点とし，逆相側が0のときには正相側をまとめて始点とする．

以上の操作より，最終ステップの逆相側が0のとき必要なインバータの数は $\sum b_i + \sum e_i$，逆相側が0で

7	6	5	4	3	2	ステップ1
逆		正		逆		正
0	0 / 1	1	0 / 2	2	1 / 2	5
2	0 / 1	3	1 / 1	4	1 / 2	7
正		逆		正		逆

図 8 図6の標準形からの作表

Fig. 8 A table generated from the normalized quasi-signal.

ないときには $\sum b_i+\sum e_i+1$ となる（アルゴリズム終）

上記2）でF個ずつインバータをまとめるとき，a_i または d_i の欄の中からどのように選んでまとめるかという自由度があるが，インバータの段数の少ない準信号系から順に選んでいくことにより最終的にインバータ段数が平均化された準信号系が得られることとなる．

図6の準信号系の標準形に対し $F=3$ として上記操作1,2,3を行った例を**図8**，**図9**，**図10**に示す．

本アルゴリズムは準信号系の標準形に対し最小のインバータで準信号系を構成するものであることが証明される．

〔定理1〕 本アルゴリズムにより構成されるインバータ回路網は最小の数のインバータで構成される．（証明は**付録**を参照）

以上 NOR ゲートのみが使用可能な場合のインバータのリダクション・アルゴリズムについて述べたが，OR ゲートも使用可能な場合にはアルゴリズムはより簡単になる．すなわち，1入力 OR ゲートおよびインバータから成る準信号系を定義し，これを図6に示す標準形に変換し，この後，それぞれに対し許容ファンアウト数内でまとめ1入力 OR ゲートを挿入すればよい．

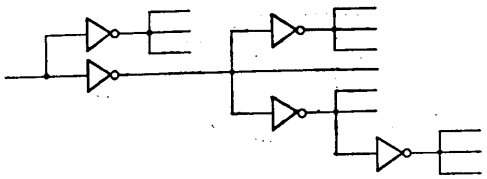

図 10 インバータリダクション後の準信号系

Fig. 10 Final result of the converted quasi-signal.

（2） LSI マクロ内リダクションアルゴリズム

本アルゴリズムは，マクロ素子の不使用出力端子に着目し，それに影響するマクロ内素子を削除する「出力端子からのリダクション」と，マクロ素子の入力端子が論理値1または0に固定されていることにより生じる冗長部分を削除する「入力端子からのリダクション」の2つの部分からなる．いずれの場合もマクロ素子の端子のところからマクロ内論理を追跡して不要部分の削除を行う．

出力端子からのリダクションは，**図11**に示すように不使用出力端子から信号を後方追跡し，追跡の通った部分をそのたびに削除することにより行われる．す

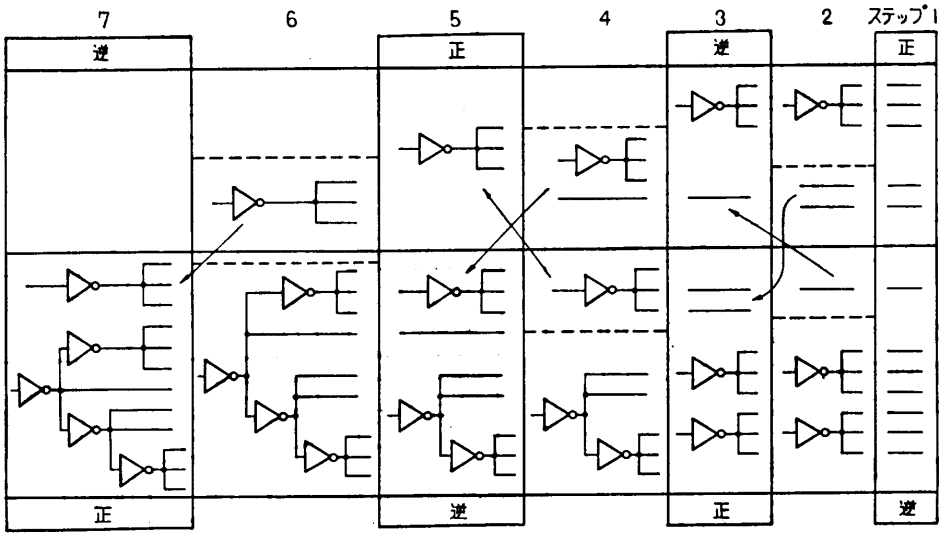

図 9 図8に対応するインバータ回路網の構成

Fig. 9 The quasi-signal construction corresponding to Fig. 8.

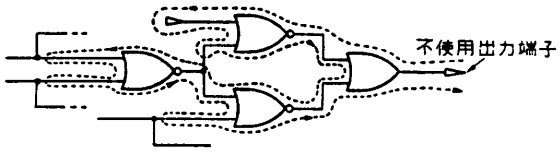

図 11 出力端子からのリダクションの例
Fig. 11 Example of reduction from unused output terminal.

なわち，信号を後方追跡し，信号が分岐していればそこで戻り，1出力の素子にぶつかったらそこをノードとし，その1つの入力端子を選んでさらに後方追跡を行い，複数出力の素子にぶつかったらその出力を削除して追跡を戻る．追跡がノードに戻ったら，未追跡の入力端子があればそれを後方追跡し，なければ追跡は戻る．追跡が元の不使用出力端子に戻った所で終了する．このように，追跡の通った素子および信号の部分を削除し，これを，そのマクロ素子のすべての不使用出力端子について行うことにより出力端子からのリダクションが行われる．

入力端子からのリダクション・アルゴリズムは，論理値が1または0に固定されているLSIマクロの入力端子から論理値1/0を保持しながら信号を前方追跡することにより行われる．すなわち，信号を追跡していき，信号が分岐していればそこをノードとしその内の1本を選んで先に進み，図12(a)のように，信号の論理値が0でNORに達すればその入力端子を削除して追跡を戻り，図12(b)のように論理値1でNORゲートに達すればそのNORゲートを削除し，その出力信号を論理値0として追跡を先に進め，同時にそのNORゲートのほかの入力端子から出力端子のリダクションのときと同様にして後方削除を行う．このように追跡を行い，その結果，追跡が元の入力端子に戻った所で終了する．

この操作をそのマクロ素子のすべての1/0固定の入力端子に行うことにより入力からのリダクションを完了する．

3. 論理再構成プログラム-LORES-

このプログラムの自動設計システム全体における位置づけは図13に示すものであり，本プログラムにより生成されたLSIに対して，論理図の自動作成，論理シミュレーション（ファンアウト，仮想配線長による伝搬遅延もシミュレートすることによりクリティカルパス等のチェックを行う．），LSIレイアウト設計などが行われる．また，本プログラムの構成は図13の枠内に示すように既存のプリント配線板の論理データベースを入力とし2.2で述べた手法による論理の再構成を行い，LSIデータベースを作成するものである．この結果得られた論理を自動作図プログラムで作画した1部を図14に示す．

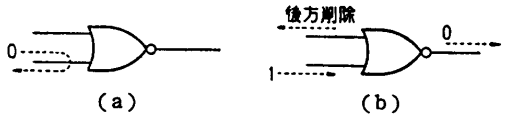

図 12 入力端子からのリダクションの例
Fig. 12 Examples of reduction from input terminal.

4. 実 施 例

本プログラムは，TTL-MSI/SSI論理を数種類のマ

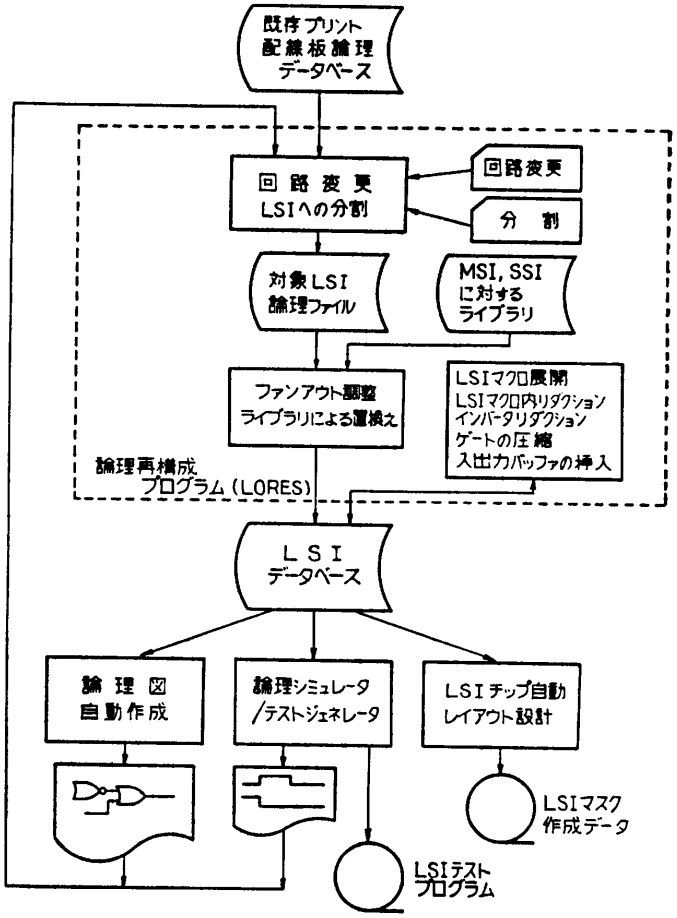

図 13 論理再構成プログラム (LORES) と関連 CAD プログラム
Fig. 13 Logic reorganization program (LORES) and its related CAD programs.

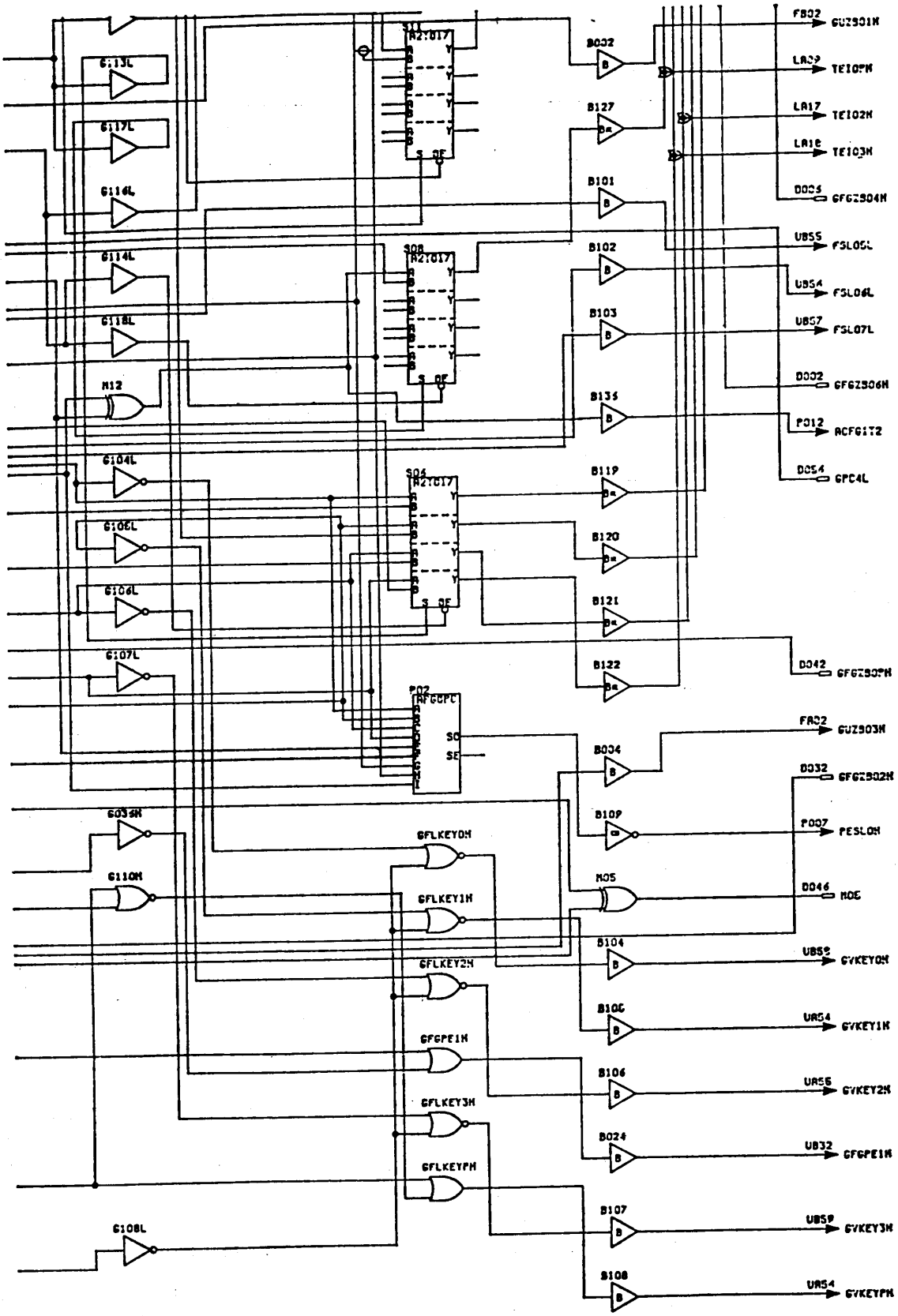

図 14 論理図作図プログラムで作った論理図

Fig. 14 An Output of automatic logic diagram generation program.

表 1 再構成の実施例
Table 1 Some results of practical use.

項目 / LSI番号	元の回路の論理量									LSI上のゲート数	計算時間 (cpu time 分)	LSIの端子数
	ゲート	インバータ	EOR	FF	MSI内部 ゲート	MSI内部 インバータ	MSI内部 EOR	MSI内部 FF	ゲート(1)総数			
1	94	82	5	2	251	237	0	0	698	800	53.5	116
2	29	15	0	20	175	98	0	4	485	307	34.3	76
3	101	56	7	0	468	119	32	0	900	897	57.7	107
4	48	35	4	0	290	146	0	16	647	517	44.2	116
5	103	29	0	0	202	133	0	38	733	651	57.5	116
6	27	3	0	13	35	12	0	24	336	430	29.0	71
7	27	42	1	10	241	148	0	12	604	584	36.1	100
8	25	32	0	0	234	102	0	0	393	335	20.5	115

(1) FF (フリップフロップ) を平均7ゲート，EOR (Exclusive OR) を4ゲートとして計算した．

スタスライス LSI の論理に再構成するために使用されている．1つの実施例は900ゲートの ECL マスタスライス LSI への再構成であり，この場合の基本素子は 3NOR/3OR でワイヤード OR が可能な構成になっている．この場合，MSI がライブラリに対応し，ライブラリとして約110種，LSI マクロが約60種登録されていて，すでに数十種の LSI 設計に使用されている．この結果の例を**表1**に示す．表1において，元の回路の論理量の欄で，ゲートは1入力のゲートから13入力のゲートまですべてを1個と数え，また MSI 内部の論理量は MSI のゲートの使用数から計算している．LSI のゲート数は LSI の総ゲート数であり，その中でフリップフロップは5～9ゲートで構成されている．表1やほかの例より，元の回路の総ゲート数と LSI の総ゲート数の関係は，多入力ゲートの数，MSI 上での不使用回路の程度，各信号のファンアウト数などにより異なり，このため，LSI の種類により異なるが，総ゲート数の ±20% 程度が LSI 上でのゲート数となることが判った．

なお，プログラムは FORTRAN でかかれ，プログラムサイズはソースカードで約18,000枚で，使用計算機は MELCOM-COSMO 700 であり，計算機時間は元のデータベースの検索から LSI データベースに最終的に登録するまでの時間である．

再構成後のゲート数の観点からこの結果とゲートレベルの置き換え簡単化方式[5]の結果を比較すると，LSI の種類が異なるため直接比較することができないが以下の理由により同種の LSI においてそのゲート数に大きな差がないことが解った．すなわち，本方式ではランダムロジック部(LSI マクロ外の個別ゲート)でワイヤードゲートに置き換え可能な部分を通常のゲートに置き換え，ゲート数を増す要因となるが，一般にランダムロジック部のゲートの全入力信号が無分岐信号であることは希でありワイヤードゲートに置き換え可能な確率は非常に少ない．一方 LSI マクロ内の部分に関しては，本方式の場合人手により交換が行われるのでより少ないゲート数となる可能性がある．

5. むすび

既存の MSI/SSI で作られた論理回路を LSI で実現できる論理回路に再構成する手順を求め，そのプログラムシステムと使用例を示した．

本方式は，元の論理の構成を維持することにより自動化により理解不能となる欠点を除去し設計者が容易に理解できるようにした所に特長がある．

既存論理の LSI 化をさらに自動化するためには分割の自動化があり，プログラムを試作し[5]実験を行っているが人手と比較しまだまだ能率が悪く，また，機能を無視して機械的に分割するため実用化には至っていない．このため，最小の人手介入で能率のよい分割方式を開発することが今後の課題である．

最後に，本研究において種々のご援助，ご指導をいただいた大阪大学工学部尾崎弘教授および藤原秀雄博士に深く謝意を表す．また，本研究の機会を与えて下さり，常にご鞭撻下さった三菱電機(株)開発本部首藤勝計算機研究部長に感謝する．

参考文献

1) 稲垣，槌田，酒井，矢島：LSI を含んだ論理システムに適した論理シミュレータ SIM/D 信学技

報 Vol. 79, No. 248 EC 79-73.
2) Kawata, N., Saito, T., Maruyama, F. and Uehara, T.: Design and Verification of Large-Scale Computers by using DDL 16th DA Conference pp. 360-366.
3) Ohno, Y., Miyoshi, M. and Sato, K.: Logic Verification System for Very Large Computers using LSI 16th DA Conference pp. 367-374.
4) 永谷, 和田, 上田: LSI ゲート構成自動化の一手法 情報処理学会 設計自動化研究会 夏季シンポジウム資料 (1976-8).
5) 中村, 村井, 田中, 寺井, 藤原, 樹下: 論理回路の LSI セルへの分割変換 信学技報 Vol. 77, No. 93 EC 77-19.
6) 寺井, 樹下, 中村, 村井, 田中: 既存の論理回路の NOR 回路への変換の一方法について 昭和52年電子通信学会総合全国大会.
7) Nakamura, S., Murai, S., Tanaka, C., Terai M., Fujiwara, H. and Kinoshita, K.: LORES-Logic Reorganization System 15th DA Conference pp. 250-258.

付 録 定理1の証明

〔定義〕 準信号系において正相出力を持つインバータを正相インバータ，逆相出力を持つインバータを逆相インバータと呼ぶ．

〔証明〕 ある準信号系において，始点のファンアウト数を f_0，正相インバータの数を N_P，i 番目の正相インバータのファンアウト数を f_{Pi}，逆相インバータの数を N_N，i 番目の逆相インバータのファンアウト数を f_{Ni}，インバータの総数を $N=N_P+N_N$，正相終点の数を T_P，逆相終点の数を T_N とする．このとき，

$$T_P=f_0+\sum_{i=1}^{N_P} f_{Pi}-N_N \cdots\cdots(1)$$

$$T_N=\sum_{i=1}^{N_N} f_{Ni}-N_P \cdots\cdots(2)$$

が成り立つ．本アルゴリズムにより構成されるインバータ回路網（これをAとする）において許容ファンアウト数F未満のファンアウト数を持つ可能性のある点は，始点と始点に接続される逆相インバータの内の1つの出力のみである（このインバータを第1番目の逆相インバータとする）．したがってAにおいて（1）（2）式より

$$T_P=f_0+N_P\cdot F-N_N \cdots\cdots(3)$$

$$T_N=f_{N1}+(N_N-1)\cdot F-N_P \cdots\cdots(4)$$

が成立する．ここでAと等価で $N'<N$ なる N' 個のインバータよりなる回路網Bが存在するとすると

$$f_0+N_P\cdot F-N_N=f_0'+\sum_{i=1}^{N_P'} f_{Pi}'-N_N' \cdots(5)$$

$$f_{N1}+(N_N-1)\cdot F-N_P=f_{N1}'+\sum_{i=2}^{N_N'} f_{Ni}'-N_P' \cdots\cdots(6)$$

となる．一方，$N=N_P+N_N>N'=N_P'+N_N'$ より

$$N_P-N_P'>N_N'-N_N \cdots\cdots(7)$$

$$N_N-N_N'>N_P'-N_P \cdots\cdots(8)$$

（5）式より

$$0=(f_0-f_0')+(N_P\cdot F-\sum_{i=1}^{N_P'} f_{Pi}')-(N_N-N_N')$$

$$\geqq(f_0-f_0')+F(N_P-N_P')-(N_N-N_N') \cdots\cdots(9)$$

$$>(f_0-f_0')+(F+1)(N_N'-N_N)$$

（$\because$（7）式を代入）

$$\therefore \quad \frac{f_0'-f_0}{F+1}>N_N'-N_N$$

$1\leqq f_0', f_0\leqq F$ であるから $\frac{f_0'-f_0}{F+1}<1$，また N_N'，N_N は自然数であるから $N_N\geqq N_N'$

同様に（6），（8）式より $N_P\geqq N_P'$ となる．

今 $N_N=N_N'$ とすると $N>N'$ より $N_P>N_P'$，また（5）式より $0=(f_0-f_0')+(N_P\cdot F-\sum_{i=1}^{N_P'} f_{Pi}')\geqq(f_0-f_0')+F(N_P-N_P')\geqq 1$ となり矛盾する．したがって

$$N_N>N_N' \cdots\cdots(10)$$

同様にして

$$N_P>N_P' \cdots\cdots(11)$$

となる．ここで

$$g=(f_{N1}-f_{N1}')+\{(N_N-1)\cdot F-\sum_{i=2}^{N_N'} f_{Ni}'\}-(N_P-N_P')$$

とすると

$$g\geqq(f_{N1}-f_{N1}')+F(N_N-N_N')-(N_P-N_P')$$

（9）式より

$$N_N-N_N'\geqq(f_0-f_0')+F(N_P-N_P')$$

これを代入して

$$g\geqq(f_{N1}-f_{N1}')+F(f_0-f_0')+(F^2-1)(N_P-N_P')$$

ここで，$1\leqq f_0, f_0', f_{N1}, f_{N1}'\leqq F$ なので

$$(f_{N1}-f_{N1}')+F(f_0-f_0')\geqq 1-F^2$$

また $N_P>N_P'$ なので，$(F^2-1)(N_P-N_P')\geqq F^2-1$

したがって $g=0$ となるためには少なくとも，$f_0=1$，$f_0'=F$，$f_{N1}=1$，$f_{N1}'=F$，$N_P=N_P'+1$ でなければならない．ところで本アルゴリズムでは $f_0=f_{N1}=1$ となるのは $T_P=0, T_N=1$ の場合だけである．

($\because$ $f_0=f_{N1}=1$ となるのは本アルゴリズムの漸化式の最終ステップ（ステップnとする）で正相側（a_i または d_i）$=0$, 逆相側（d_i または a_i）$=1$ となる場合であるが，$n\neq 1 (n\geqq 3)$ のとき，これはステップ $n-2$ で正相側$=F$, 逆相側$=0$ の終了条件が成り立つことになるからあり得ない.)

$T_P=0$, $T_N=1$ のとき，明らかに $f_{N1}'=F$ とはなり得ないから矛盾する．また $T_P=0, T_N=1$ 以外の場合，$f_0=f_{N1}=1$ となることはないから矛盾する．したがって，本アルゴリズムにより構成されるインバータ回路網は最小数のインバータで構成される．（証明終り）

（昭和55年5月19日受付）
（昭和55年10月23日採録）