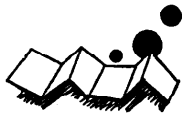


解説



VLSI における回路設計方式—PLA を中心に†

笹 尾 勤††

1. はじめに

VLSI 向き論理設計手法については、すぐれた教科書^{1)~6)}や解説^{7),8)}があり、基本的な事項はすでに述べられている。

VLSI では、数十万個のトランジスタを正しく接続し（論理設計）、信号遅延や雑音などを十分考慮し、要求水準を満足し（回路設計）、プロセスから決まる設計規則を犯すことなく配線を正しく行う（パターン設計）必要がある。これらの設計を誤りなく行うには、自動設計システムを駆使することが必須となる。また、製造時にこれらのトランジスタがすべて正しく動作していることを確認することが不可欠である。マイクロプロセッサなどの複雑な回路では、すべての機能を高々数百個の入出力ピンを用いて検査するのは、容易なことではない。結局、VLSI 向き設計手法とは、自動設計システムを駆使でき、かつ検査容易な設計法ということになる。

ROM, RAM, PLA (Programmable Logic Array) を多用し、単純化した設計基準を用いれば、設計を単純化できる。そのため、多品種少量の VLSI の開発には、規格化された設計手法が定着している。

PLA は規則的構造を有するため、論理設計や検査が簡単で、設計から検査に至るまでの一連の処理の自動化が容易である。ただし、ランダムロジックに比べ、面積や性能の点で劣るとされている。しかし、VLSI の集積度がますます増大するにあたり、設計期間、デバッグ、仕様変更への柔軟な対応などの問題を考慮すると、PLA の導入は必須となる。VLSI のコストの3分の1は検査のコストである。マイクロプロセッサは大量生産され多様のユーザに種々の形で用いられるため、性能と価格が重要である。性能を下げずに PLA を利用する技術、検査のコストを減らす技術

が開発されている。

第2章では、PLA を用いた設計法を述べる。第3章では、検査容易化設計手法について述べる。第4章では、最近発表された32ビット・マイクロプロセッサ (ATT WE 32100, Motorola MC 68020, IBM Micro/370, Berkeley SOAR, INTEL 80386) での PLA 設計例、及び検査容易化設計例を紹介する。

2. PLA を用いた設計法

2.1 PLA とは

組み合わせ論理回路は、真理値表を直接表現した ROM (読み出し専用メモリ) で実現できる。しかし、 n 変数関数の実現には 2^n 語必要であるため、 n が大きくなると実用的ではない。

通常、制御回路などは、機能を真理値表で表現するよりも、論理式で表現したほうがはるかに簡単になる。真理値表のかわりに論理式を用いたのが PLA である。たとえば、表-1 の真理値表を PLA で実現するには、 f_0 と f_1 を表現する論理和形

$$f_0 = x_1 x_2 \vee x_1 \bar{x}_2 x_3$$

$$f_1 = \bar{x}_1 \bar{x}_2 \bar{x}_3 \vee x_1 \bar{x}_2 x_3$$

を求めればよい。図-1 は、表-1 を実現する PLA であるが、AND アレイの各積項線（垂直線）は、左からそれぞれ積項 $x_1 x_2$, $x_1 \bar{x}_2 x_3$, $\bar{x}_1 \bar{x}_2 \bar{x}_3$ を実現している。また、OR アレイでは、各積項の論理和を実現している。この PLA で、第2番目の積項は、 f_0 と f_1

表-1 真理値表

x_1	x_2	x_3	f_0	f_1
0	0	0	0	1
0	0	1	0	0
0	1	0	0	0
0	1	1	0	0
1	0	0	0	0
1	0	1	1	1
1	1	0	1	0
1	1	1	1	0

† Logic Design Method for VLSI—Design and Test for Programmable Logic Arrays by Tsutomu SASAO (Department of Electronic Engineering, Osaka University).

†† 大阪大学工学部電子工学科

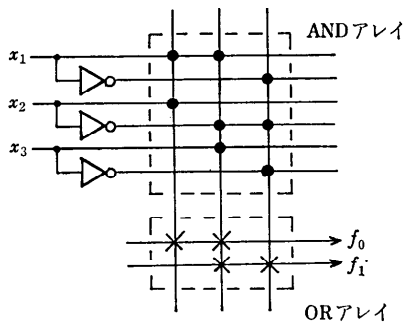


図-1 表-1の関数を実現する PLA

の両方の出力に接続していることに注意されたい。 f_0 と f_1 を別々に最小化すると積項は 4 個必要である。これからわかるように、PLA を最小化するには、すべての出力を同時に考慮した多出力関数の簡単化が必要である。

PLA は、規則的構造を有するため、次のような特長がある。

- 1) 論理設計が比較的容易である。
- 2) レイアウトが簡単である。
- 3) 論理変更が容易であり、変更を行っても周辺回路に与える影響が少ない。

このため、PLA では設計から検査までの処理をほとんど自動化でき、また、論理設計の段階でチップ面積の推定ができる。

2.2 PLA の縮小

2.2.1 論理式の簡単化⁹⁾

従来、論理式の積項数の最小化は、入力数が 10 を越えると計算時間が大幅に増え、大型計算機を用いても最小化は困難であると考えられていた¹⁰⁾。しかし、ここ数年間の研究によって、入力数が 20 程度以下なら、積項数の最小化（最小性の保証がある）が可能になっている^{20), 21)}。また、入力数が 50~100、出力数が 50~100 程度の論理式の簡単化（必ずしも最小性の保証はない）が実行時間内で行えるようになった^{11)~19)}。

最近問題になっているのは、多値入力二値出力関数の論理式の簡単化である。多値論理式の簡単化は、次に述べる、2ビットデコーダ付 PLA の簡単化、PLA を用いた順序回路の最適符号割当、PLA の分解のほかに、多出力関数の簡単化、多段論理回路の合成、マイクロプログラムの最適化に利用できる。多値論理式簡単化プログラムとしては、MINI、MINI-II、ESPRESSO-MV があり、最小化プログラムとしては、QM⁹⁾、ESPRESSO-EXACT²¹⁾ がある。

2.2.2 デコーダ

通常の AND-OR 形 PLA の入力インバータの部分に 2 ビットデコーダに置き換えると、AND アレイの部分が小さくなる。特に、実現すべき関数が対称性をもつ場合、アレイ部分は大幅に小さくなる。2 ビットデコーダ付 PLA の場合、入力変数を二つずつ対にするが、変数の組み合わせの仕方によって、必要な積項数が大きく変化する²²⁾。デコーダ付 PLA の簡単化には、多値論理式簡単化プログラムを用いる。入力変数の割当には、ヒューリスティック法が開発されている²³⁾。制御回路の場合、入力変数の割当を考慮した 2 ビットデコーダ付 PLA は、通常の PLA に比べ積項数が 2 割程度少ない²²⁾。ただし、デコーダの部分に余分のハードウェアが必要である。

2.2.3 出力位相最適化

通常、PLA の出力部には、バッファを付ける。バッファとしては、反転型と非反転型があるが、チップ面積や遅延時間はいずれの型でも同じである。たとえば、

$$f = x_1x_3 \vee x_1x_4 \vee x_2x_3 \vee x_2x_4$$

という関数をそのまま実現すれば積項は 4 個必要である。しかし、 f のかわりに $\bar{f}$ を実現すれば、

$$\bar{f} = \bar{x}_1\bar{x}_2 \vee \bar{x}_3\bar{x}_4$$

となり、積項は 2 個で十分である。一般に、出力数 m の PLA で積項数が最小になるように各関数の出力極性を選択する問題を出力位相最適化問題（あるいは、出力極性選択問題）という。この問題は、 2^m 個のすべての組み合わせに対して、多出力の論理式の簡単化を実行すればよいが、 m が大きい場合には、計算時間がかかりすぎ、実用的ではない。

比較的良好な解を短時間内にみつけるヒューリスティック法が開発されている²²⁾。その手法を用いると、通常、算術演算回路の場合、約 10%、制御回路の場合約 5% 積項数を削減できる。しかし、PLA によっては、30% 近く削減できる場合もある (SOAR の項参照)。

2.3 PLA の畳み込みと分解

PLA が大きくなるにつれて、アレイのドント・ケアの部分が増え、チップの利用率が悪くなる。PLA の積項線数が増えると遅延時間が増す。また、信号線が多数の負荷を駆動するので、消費電力も大きくなる。これらの問題を解決する方法として、畳み込みと分解がある。

2.3.1 畳み込み

畳み込みは、PLA の入力線を左右から加え、積項線を

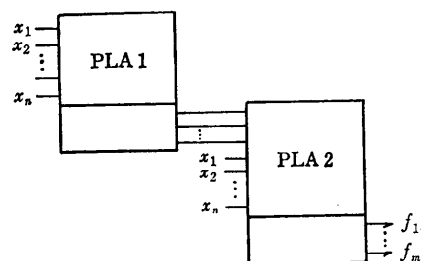
適当に入れ換え、不要な部分を除去し、出力線を左右から取り出すことによって PLA のサイズを小さくする手法である。畳込みアルゴリズムは種々考案されている^{24)~26)}。畳込み PLA は、通常の PLA に比べ、アレイのサイズが半分程度になるといわれているが²⁴⁾、ほとんど小さくならない PLA もある。畳込みを行うと入力線や出力線の方法や順序がバラバラになる。畳込みで PLA のサイズが小さくなくても、配線部分が複雑になり、全体の面積がもとの PLA より大きくなることもある。畳込み PLA では、論理の変更は困難であり、将来、修正が予測される部分には使えない。また、配線部分の容量の予測が困難であり、MOS 回路などで、動作速度が負荷の容量に大きく依存するような場合には、安心して使えない。このように、畳込み技術には、今のところ実用上の問題点が多く、表-2 のマイクロプロセッサでは使用されていない。

2.3.2 分解 (分割)

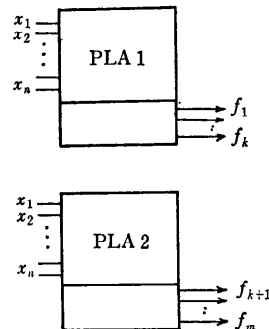
PLA の分解は、畳込みよりも単純であり、うまく実行すると、畳込みよりもはるかに良い結果が得られる。

分解とは、出力の論理関数の性質を詳しく解析して大規模な関数をいくつかの小さな PLA で実現する技術である³⁰⁾。これによく似た技術として分割がある。分割では、PLA のパターン (論理式) のみに注目し、論理関数としては取り扱わない。

分解には、図-2 に示すように、直列分解と並列分解がある。直列分解は、多段論理回路に対応する。多段論理回路にすると、アレイの総面積が減ることが多い。ただし、段数や配線部分が増えるため、遅延時間が問題となる。並列分解は、出力数の多い PLA で有効である。制御用 PLA の場合、PLA が大きくなると、特定の変数集合のみに依存する出力関数が多くなる。そのため、並列分解を行った PLA の入力には、もとの入力変数の一部分のみを加えれば十分であることが多い。出力数が多い場合、積項数の総和をほとんど増やさずに PLA を分解できる (SOAR の項参照)。PLA をうまく分解すれば、分解して得られた小さな PLA の面積の総和は、もとの PLA に比べ、かなり小さくなる。また、積項線数が減り、負荷が小さくなるため、動作速度も改善できる。小さな PLA は、割り当てられたチップ領域に配置するのも容易である。ただし、配線部分は複雑になる。分解のもう一つの効用は、検査の容易化である。PLA を分解すると PLA の入力数が減る。このため、検査の複雑度が減る³⁰⁾。



(a) 直列分解



(b) 並列分解

図-2 PLA の分解

また、組み込み検査への適用も容易となる。このように、PLA の分解 (分割) 技術はきわめて有効であるため、表-2 の多くのマイクロプロセッサの設計で利用されている。

2.4 PLA を用いた順序回路

VLSI システムは、データパス部と制御部に分割できる。制御部は、通常、有限状態機械 (FSM Finite State Machine) として実現する。FSM は、組み合わせ回路部とレジスタ部から構成できる。組み合わせ回路部はランダムロジック、PLA、あるいは ROM で実現する。このうち、性能、消費電力、面積、及び設計時間のトレードオフを考慮すると PLA が最も使いやすい。

2.4.1 FSM の分解

FSM は、並列に動作するいくつかの小さな FSM に分解する。こうすると、各 FSM の組み合わせ回路部が簡単になり、レジスタ部の負荷が減る。ただし、あまり小さく分解しすぎると、冗長な部分が増えたり、配線部分が増えたりする。

2.4.2 状態割当

FSM の各状態に 2 進符号を割り当てることを状態割当²⁷⁾という。回路が最も簡単になるような状態割当を求める問題を最適状態割当問題という。ランダムロジックの場合、この問題はきわめて難しい²⁸⁾。

PLA の場合は、符号長と積項数のみで PLA の複雑度が決まるので、次のようにすれば比較的良い解が見つかる^{29), 30)}。

1) FSM の状態数を S とするとき、現在の状態と次の状態の関係を S 値の変数 (1 ホットコード) を用いて表現する。

2) 上の論理式を実現する S 値の PLA を考え、多値の論理式簡単化プログラムで簡単化する。

3) 簡単化した多値の論理式を参考にして、なるべく論理式の積項数が増えないように、 S 値の変数部を 2 値の符号で置き換える。

この方法では、必ずしも最適な状態割当が求まる保証はなく、符号長も最短になるわけではない。しかし、最初に PLA の積項数を最小化するので、積項数の少ない割り当てが得られ、チップ面積が小さくなることが多い。

2.5 PLA における諸問題

2.5.1 ダイナミック PLA

ダイナミック PLA の消費電力は少ない。ただし、信号線と電源線上に重畳したノイズに強い影響を受けるため、PLA を設計する際、いくつかの制約が生ずる。チップ上のクロック発生器は、容量負荷が大きく、シャープな波形が要求されることから、設計は非常に難しい。システムクロックの周波数は、スタティック PLA の場合の 2 倍になる。

ダイナミック PLA では、充電と放電時に大電流が流れるので、チップの性能は、ピーク電流と、電流の変化分に制限されることが多い。第二メタル層を用いた場合、容量カップリングのために浮動節点に電圧降下が生ずるが、これは、非常に複雑な問題である。大域的配線用に、PLA の上層にある第二メタル層を用いる場合、PLA の設計時には、ノイズの環境は予測できない。ダイナミック PLA のノイズは、BELLMAC-32 A の設計時に大きな問題となった³⁷⁾。

2.5.2 スタティック PLA

スタティック PLA の性能は、消費電力で制限される。Micro/370 では、電力遅延積を改善するために、特殊な回路構成をとっている⁴²⁾。また、大きな PLA は、分解することにより、性能を改善する。

3. 検査容易化設計

ゲート数が数千程度までの回路は、ATPG (Automatic Test Pattern Generator) と呼ばれる自動検査入力生成プログラムで発生した検査入力で検査でき

る。ゲート数が数万以上になると、ATPG は時間がかかりすぎるので、機能検査を行う。機能検査では、すべての基本的な機能 (及び性能) を満足するかどうかを調べる。そのため、すべての故障を検出できるという保証はない。

VLSI の場合、トランジスタ数が非常に多いので、従来の方法での検査は不可能である。そのため VLSI 設計の最初から、検査を十分考慮しておくことが必須である。以下では、最近の VLSI で用いられている検査容易化設計のいくつかの方法について述べる。

3.1 分 割

マイクロプロセッサでは、回路はバス構造で設計されており、論理ブロックの大部分はバスに接続している。通常、バスは、外部からアクセスできるので、バスを通して LSI の小さなブロックを別々に検査できる。さらに検査を容易にするため、通常はバスに直接接続していない論理ブロックも、検査時には、直接バスに接続し、自由に読み書きできるようにしたり、マルチプレクサを用いてさらに回路を分割したりする。MC 68020, 80386, WE 32100 などのプロセッサでは、PLA, ROM, RAM, レジスタ, キャッシュなどは検査時に直接外部から制御できるようになっている。そのほか、ある特定の信号を加えると、すべての出力端子がハイ・インピーダンスとなり、チップをボードから取りはずしたのと同じ状態にできるチップもある (WE 32100, 80386)。この機能により、VLSI のチップ内部のみならず、ボードの周辺回路の検査も容易となる。

3.2 検査用マイクロコード

検査時には特別なマイクロコード命令や、検査入力を生成する ROM を用いて、マイクロプロセッサを検査する³¹⁾。検査用プログラムをすべて ROM 中に内蔵すると、チップ面積が大きくなりすぎるので、ROM 中にはブートストラップ・ローダのみ内蔵し、検査時に検査用プログラムをシリアル I/O ポートから内蔵 RAM に読み込み、自己検査を行うチップもある (モトローラ社 MC 68 HC 11)³⁹⁾。

3.3 スキャンバス

スキャンバス³²⁾を用いると、順序回路を組み合わせ回路として検査できる。検査に要する時間は、主としてスキャンバスの段数によって決まる。スキャンバスを使用するためには、適当な検査入力を発生する必要があり、また、その応答を観測する装置も必要となる。スキャンバスを用いて、チップをいくつかの領域に分割する方法もある (Micro/370)。

スキャンパスを使うとチップ面積がかなり増える。そのため、チップ価格が重要な場合には、スキャンパスが使えない場合もある。

3.4 組み込み検査

スキャンパス法では、VLSI と VLSI テスタ間の検査入力の転送に相当な時間を要する。VLSI テスタは非常に高価なので、大量生産するマイクロプロセッサなどでは、VLSI 中に検査入力生成器と判定回路を組み込み、検査費用を安くしている。これを**組み込み検査機構**という。米国の一部メーカでは、これを組み込み自己検査 (BIST: Built-In Self Test) と呼んでいる。組み込み検査機構を導入すると、電源オン時に、チップの検査を自動的に行えるので、フィールドでの信頼度も上がる。図-3 に示すように、組み込み検査機構は、系列生成部と判定部から成り立っている。系列生成部は、2進カウンタや LFSR (Linear Feedback Shift Register) で構成する。判定部には、応答出力系列の圧縮回路があり、通常、カウンタや LFSR を用いる。このうち、LFSR を用いて応答出力系列を圧縮する方法を**シグネチャ解析法**という。

68020 や 80386 の PLA や ROM は、組み込み検査機構を用いて検査している。ROM や PLA を網羅的な検査パターンを用いて検査するためには、検査時間の点から、入力数は 20 程度以下にする必要がある。

4. マイクロプロセッサと PLA

PLA は、設計が容易であるが、動作速度やチップ面積の点でランダムロジックに劣るため、多品種少量の LSI に適するとされてきた。しかし、最近の 32 ビット・マイクロプロセッサのような大量生産品種の VLSI でも、PLA を多用している。PLA の特長は、第 2 章でも述べたが、32 ビットプロセッサで特に重要なのは、**設計変更の容易さ**である。32 ビットプロセッサはきわめて複雑で、多数のグループが協力して設計を行う。誤りの修正や機能の改善のため、マイクロプログラム部や制御部は何度も変更する。そのため、設

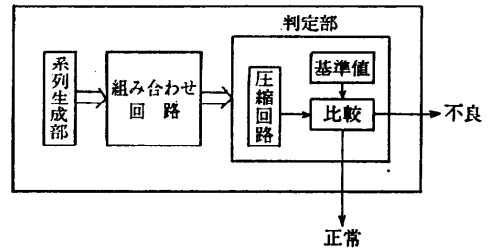


図-8 組み込み検査

計変更ができないような、構成法は使いものにならない。マイクロプロセッサのチップ面積の 80% は、ROM, RAM, レジスタファイルなどの規則的な部分が占めるが、これらの回路の設計には、それほど時間はかからない。時間のかかるのは、制御回路であり、全設計時間の 80% が費やされる。マイクロプロセッサで PLA を多用している箇所は、設計に時間がかかり、設計変更の多いマイクロプログラム部³¹⁾と制御部である。ALU など、速度を特に重視する部分や、設計がかたまり、将来、設計変更の必要性のない部分は PLA 以外の設計法を用いる。

32 ビット・マイクロプロセッサでは、価格と性能のほかに、信頼度も重要になる。組み込み検査機構を用いて信頼度を上げたプロセッサも現われている。本章では、最近発表された 32 ビットプロセッサ (表-2) を例にとって、PLA を用いた設計法、及び検査容易化設計法を紹介する。

4.1 ATT WE 32100^{35)~38)}

WE 32100 は、ATT が開発した 32 ビットプロセッサであり、UNIX 上の C 言語を効率よく実行する。このプロセッサは、BELLMAC と呼ばれ、BELLMAC 32, 32A', 32A, 32B と改良されてきた。図-4 に WE 32100 のチップ写真を示す。ここで、チップの上部は大部分 PLA で、下部は RALU (Register Arithmetic Logic Unit) 部である。制御回路の大部分は 8 個の PLA で実現している。PLA のレイアウトは機能記述レベルの高級言語から自動的に生成した。

表-2 ビットマイクロプロセッサと PLA

プロセッサ名	設 計	Tr 数	PLA 数	テクノロジー	ピン数	動作周波数 (MHz)
WE 32100	ATT	180 k	8 (表-3)	1.75 μ CMOS	108	20~25
MC 68020	Motorola	200 k	6 (表-4)	2 μ HCMOS	114	16.67
Micro/370	IBM	200 k	2+42	2 μ NMOS	171	10~18
SOAR	Berkeley	35.7 k	11 (表-5)	4 μ NMOS	84	2.5*
80386	INTEL	285 k	3 (表-7)	1.5 μ CHMOS	132	16

* サイクルタイム

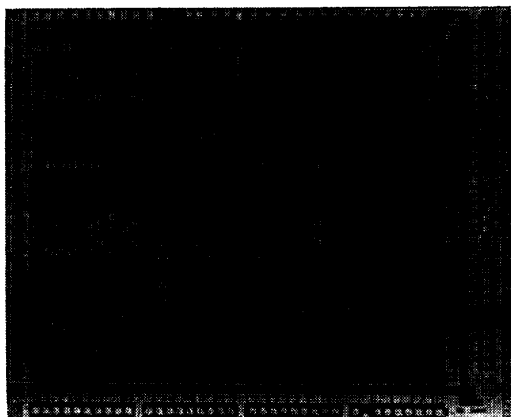


図-4 WE 32100 のチップ写真

RALU の部分は、PLA では遅いので、ゲートマトリックス法³⁸⁾で生成した。ゲートマトリックス法はシンボリック設計手法の一種であり、人手で会話的に設計する必要があるが、設計基準を変更しても再びコンパイルすれば新しいレイアウトが生成できる。32ビットプロセッサ程度の複雑度になると、最初の設計で完全なチップを得ることは非常に困難で、いろいろな不都合をなくしたり、性能を改善したりするため、何回もバージョンアップする。

表-3 に BELLMAC 32A で使用した 8 個の PLA を示す。これらの PLA はダイナミック CMOS の NOR-NOR で実現している。2.5 で述べたように、クロックパルスの立ち上がり時にすべての寄生容量中の電荷が放電するため、グラウンドライン上に大きなノイズが生ずる。そのため、ノイズを減らすいくつかの工夫がしてある。

WE 32100 では、検査を容易化するため、内部の ROM は、外部から直接アクセスできるようになっている。また、ボードレベルの検査を容易化するため、特定の入力を加えると、チップの出力がすべてハイ・インピーダンスになるよう設計されている。

4.2 Motorola MC 68020^{39), 40)}

Motorola 社のプロセッサは、6つの大きな PLA を用いている。

図-5 に示すように、MC 68020 は EU (実行ユニット) と、EU 及びキャッシュを制御するマイクロマシン部から成り立っている。IRB, IRC, IRD の三つの 16 ビットレジスタは、エントリ PLA 部中にある 4 つの PLA (A1, A2, A5, 及び A6: 表-4 参照) を駆動している。これらの PLA の出力は、マイクロコード用 ROM のアドレス (11 ビット) を表現する。

表-3 BELLMAC 32A の PLA

PLA	入 力	出 力	積 項
1	50	67	190
2	30	27	120
3	27	54	181
4	54	61	134
5	30	37	153
6	24	13	44
7	12	21	58
8	25	12	42

表-4 M 68020 の PLA

PLA	入 力	出 力	積 項
A1	16	11	240
A2	16	11	85
A5	22	11	105
A6	11	11	10

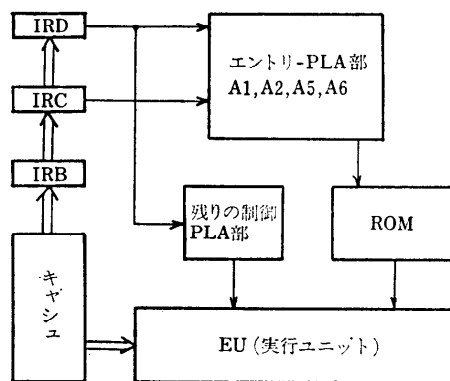


図-5 MC 68020 のマイクロマシン

“残りの制御 PLA 部”には、二つの大きな PLA があり、二つの PLA の出力数は約 150 である。そのほか、小さな PLA がいくつかあるが、それらの出力数は約 50 である。

MC 68020 の検査には、チップを小さなブロックに分割するという基本方針を用いている。PLA, ROM 及びキャッシュの検査は、検査用マイクロコード、バス及びマルチプレクサを用いてほかの部分から分離し、独立に行う。A1 と A2 の PLA は、二つの方法で検査できる。一つは、テストからの検査ベクトルの注入であり、もう一つは、組み込み検査機構である。組み込み検査時には、EU のレジスタを用いて網羅的な検査入力を生成する。A5 と A6 の PLA の入力は、IRC と IRD のビットを結合したものであり、これらの PLA を網羅的方法で検査するには時間がかかりすぎ

る。A1, A2, A5, A6 の PLA の出力数はすべて 11 で、これらが NMA (the next microcode address) バスに接続している。これをそのまま観測するのは大変なので、16 ビットのシグネチャレジスタを用いてデータ圧縮する。残りの制御 PLA は、データ圧縮を行わず、マルチプレクサを用いてバスから直接出力値を観測する。

ROM は、検査時に入力ピンからアクセスできる。11 入力なので、網羅的検査を行う。

レジスタはすべて 32 ビットであり、32 ビットのデータ・アドレスバスを用いてアクセスできる。EU 部と制御部は、機能検査を用いて調べる。

MC 68020 は大量生産するチップなので、チップ面積は小さいほどよい。検査用に付加した論理回路は全面積の 3% 以下であり、また、検査用マイクロコードは、全マイクロコードの 2% 以下である。

4.3 IBM Micro/370^{(1), (2)}

Micro/370 は、IBM ワトソン研究所が開発した 32 ビットプロセッサであり、システム/370 (大型コンピュータ) の命令のうちの 102 個の命令を直接実行する。図-6 に Micro/370 のブロックダイアグラムを示す。命令デコード部は二つの PLA を使っている。制御部は数個の PLA を用いて実現している。

バス制御部の状態数は 42 個あり、回路を簡単にするため、並列に動作する 5 個の有有限状態順序機械に分解してある。バス制御部をスタンダードセル法で設計してみたり、大きな PLA を 2~3 個用いて設計したり試みたが、期待する密度、性能、及び消費電力が得られなかった。また、ランダムロジックは、設計の変更が困難であるため使えなかった。そのため、最初、大きな PLA で実現し、PLA の分割と配置を会話的に何度も繰り返して、最適化した。最終的に得られた回路は 42 個の PLA から成り立ち、きわめて密度の高い回路となった。このほか、計算実行部は、形状が

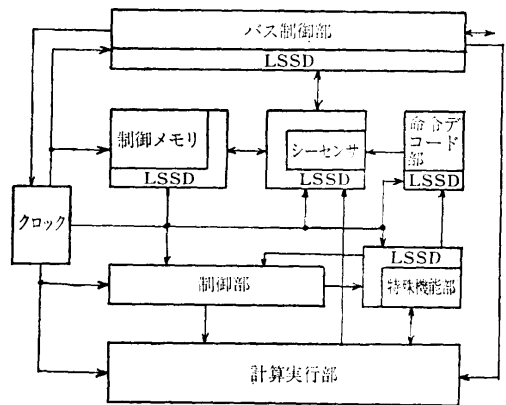


図-6 Micro/370 のブロック・ダイアグラム

規則的で、設計も固まっていたので、人手で設計した。シーケンサの部分は、PLA では遅いので自動配置配線を用いたスタンダードセル法で実現した。

Micro/370 の検査には、機能検査と LSSD を用いたストラクチャード検査の両方を用いている。制御メモリとデータバス部には、機能検査を用いている。面積や検査回路の性能を考慮すると、ストラクチャード検査は使えなかった。バス制御部、命令デコード、及びシーケンサ部は、非常に複雑なので LSSD が必須であった。LSSD を使用したためのレジスタマクロの面積増加分はかなり大きく、もとの面積の 2~3 倍以上になった。しかし、大部分の面積が配線のために費やされているので、制御回路全体での面積増加分は、10% 以下におさえられている。図-6 に示すように、LSSD レジスタはチップをバス制御部、シーケンサ・命令デコード部、制御メモリ、そして、計算実行・特殊機能部の 4 つの部分に分割している。これらの部分は、別々に検査する。Micro/370 は、ピン数が 171 本あるが、そのうち 106 本が信号用、26 本が検査用、38 本が電源とグラウンド用である。

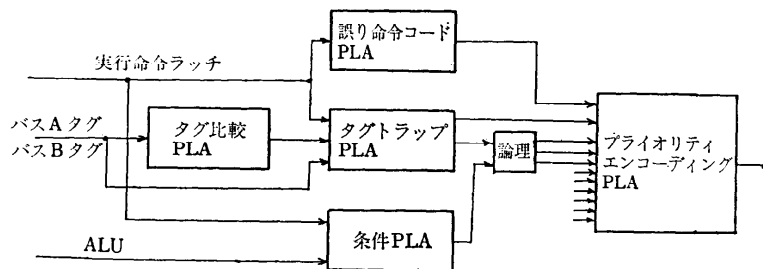


図-7 SOAR のトラップ機構

4.4 Berkeley SOAR^{(43), (44)}

SOAR (Smalltalk on a RISC) は, Smalltalk を効率よく実行する 32 ビットプロセッサである。他のプロセッサとは異なり, 大学で開発した実験用チップであり, SUN ワークステーションと外部メモリの下で動作する。

図-7 に SOAR で用いたトラップ機構の一部を示す。SOAR の制御部は, 表-5 に示す 12 個の PLA で構成している。これらの PLA は, 面積と動作速度が最適になるように分割してある。表-6 に SOAR で用いた, 命令デコーダ用 PLA の最適化例を示す。最初の設計では, この PLA は 10 入力 39 出力 152 積項であった。ESPRESSO-II C を使って簡単化したところ, 80 積項まで減った (PLA (T))。しかし, この PLA はまだ大きすぎたため, 入力数 9 出力数 16 の PLA (A) と, 入力数 9 出力数 23 の PLA (B) の二つに並列分解した。これらの PLA を別々に簡単化したところ, 積項数は, それぞれ 38 と 41 になった。実際の SOAR の LSI 中では, これらの PLA が実現されている。

しかし, さらに以下のような最適化が可能である。これらの PLA の入力中には, RESET 入力がある。RESET 入力値が 0 のとき, 4 つの出力値のみが意味をもち, 他の出力値はどうでもよい。この条件をドント・ケアとして簡単化すると, PLA (T) の積項数は 63 まで減る。さらに, 文献 12) の方法を用いて出力位相を準最適化すると PLA (T) の積項数は, 47 まで減る。このように, SOAR の PLA では, 出力位相最適化が非常に効果的であるが, 文献 12) のアルゴリズムでは, 必ずしも最適解は求まらない。実際, 出力関数をすべて否定して簡単化すると積項数は 39 まで減る。出力位相割当問題に関しては, 引き続き研究が必要である。

4.5 INTEL 80386⁽⁴⁵⁾

INTEL 80386 は, 8086 及び 80286 のオブジェクトコードを直接実行できる 32 ビット・マイクロプロセッサである。組み込み検査機構と付加検査機構を用いると, チップ中のトランジスタの 50% 以上を検査でき

表-5 SOAR の PLA

PLA	入 力	出 力	積 項
デ コ ー ド	9	16	38
	9	23	43
	7	10	19
レジスタアクセス	12	10	17
	12	7	10
	6	7	7
条 件	11	2	34
ト ラ ッ プ	8	1	8
	15	5	23
	10	2	4
	10	4	9
	8	1	8

表-6 SOAR 用 PLA の最適化

PLA	入力数	出力数	積 項 数	ドント・ケアを考慮		
				単純な簡単化	もとの関数	
					出力位相準最適化	すべての出力を否定
A	9	16	38	30	27	31
B	9	23	41	37	26	30
T	10	39	80	63	47	39

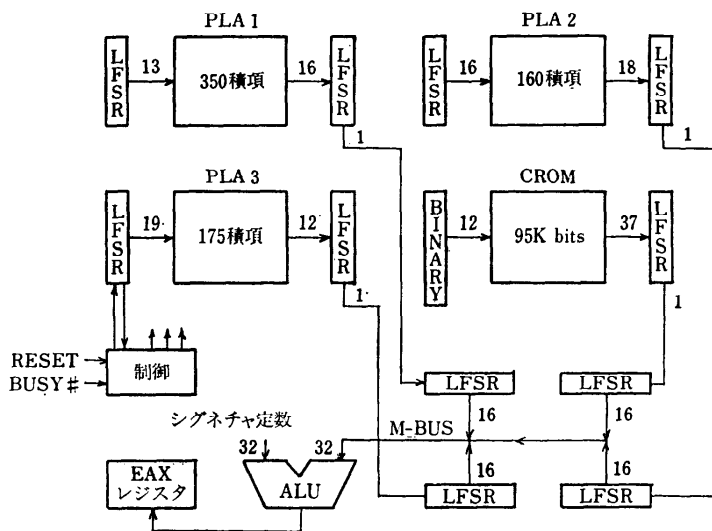


図-8 80386 の組み込み検査

る。INTEL ではこの組み込み検査機構を BIST (組み込み自己検査), 付加検査機構を検査用フック (hook) と呼んでいる。

図-8 に組み込み検査機構を示す。三つの大きな

表-7 80386 の PLA

PLA	入 力	出 力	積 項
1	13	16	350
2	16	18	160
3	19	12	175

PLA がある (表-7)。検査中は、通常の PLA 入力を切り離し、LFSR が AND アレイに 2^n-1 個の擬似乱数を加える。PLA の OR 出力部には、各出力を並列に入力する LFSR があり、網羅的な入力に対する計算結果を蓄積する。CROM (マイクロプログラム用) の入力部には 2 進カウンタが接続している。検査中は、このカウンタが 2^{12} 個のすべての組み合わせを計数する。リセット中に BUSY # ピンを低電位にすると、組み込み検査を開始する。検査が始まると、デバイスは、512k サイクルの自己検査を行う。PLA 3 の入力数が最大であり、19 入力であるから $2^{19}=512k$ となる。これは、16 MHz 動作時で約 33ms かかる。自己検査は、PLA 3 がもとの値に戻ったときに終了する。次に、マイクロコード命令を実行し、二つの 32 ビット蓄積レジスタのシグネチャとチップ内部に記憶してある正しいシグネチャとの EXOR をとり、EAX レジスタに入れる。デバイスが正常の場合には、EAX レジスタの値はすべて 0 となる。EAX レジスタの内容は、ユーザが読めるので、電源投入時にプロセッサの検査ができる。

付加検査機構を用いると、ほとんどのレジスタを直接読み書きできる。これらのレジスタは通常のレジスタのほかに、マイクロプログラム専用のレジスタも含まれている。また、すべての入出力ピンを電氣的に切り離す機構がある。CLK 2 (外部クロック) を入力したまま HOLD/HALA シーケンスを実行するとすべての出力がハイインピーダンスとなり、すべての入力が無視される。これは、マイクロプロセッサの周辺回路を検査する際、有効である。

5. あとがき

本論文では、最近の 32 ビット・マイクロプロセッサを例にとって、PLA 設計および検査容易化設計について紹介した。VLSI では、ROM・RAM・PLA を多用した設計法が、定着した技術となっている。PLA は論理設計・変更・検査が容易であるため、今後とも VLSI 設計の基本要素として重要な地位を占めるだろう。

参 考 文 献

一 般

- 1) Mead, C. and Conway, L.: Introduction to VLSI Systems, Addison-Wesley Publishing Co., p. 396 (1980).
- 2) Muroga, S.: VLSI System Design, John Wiley & Sons, p. 489 (1982).
- 3) Weste, N.H.E. and Eshraghian, K.: Principles of CMOS VLSI Design: A Systems Perspective, Addison-Wesley Publishing Co., p. 531 (1985).
- 4) 可児, 川西, 船津: 超 LSI CAD の基礎, オーム社, p. 225 (1983).
- 5) 渡辺, 浅田, 可児, 大附: VLSI の設計 I, 岩波書店, p. 275 (1985).
- 6) 樹下, 浅田, 唐津: VLSI の設計 II, 岩波書店 p. 313 (1985).
- 7) 安浦寛人: 論理回路の設計技術 [Ⅲ]: VLSI 向き論理設計, 電子通信学会誌, Vol. 67, No. 1, pp. 78-83 (1984).
- 8) 須藤, 中島, 吉村: ハードウェアアルゴリズムと VLSI 設計, 情報処理, Vol. 26, No. 6, pp. 593-602 (1986).

論理式簡単化

- 9) 笹尾 勲: PLA の作り方・使い方, 日刊工業新聞社, p. 213 (1986).
- 10) 石川, 笹尾, 寺田: 論理式最小化の一手法とその適用限界について, 電子通信学会論文誌 (D), Vol. J 65-D, No. 6, pp. 797-854 (1982).
- 11) Hong, S. J., Cain, R. G. and Ostapko, D. L.: MINI: A Heuristic Approach for Logic Minimization, IBM J. Res. Dev, pp. 443-458 (1974).
- 12) Sasao, T.: Input Variable Assignment and Output Phase Optimization of PLA's, IEEE Trans. Comput., Vol. C-33, No. 10, pp. 879-894 (1984).
- 13) Brayton, R. K., Hachtel, G. D., McMullen, C. T. and Sangiovanni-Vincentelli, A. L. M.: Logic Minimization Algorithms for VLSI Synthesis, Kluwer Academic Publishers (1984).
- 14) Rudell, R. L. and Sangiovanni-Vincentelli, A. L. M.: Espresso-MV: Algorithms for Multiple Valued Logic Minimization, Custom Integrated Circuits Conference, pp. 230-234 (May 1985).
- 15) Brown, D. W.: A State-machine Synthesizer-SMS, 18th DAC, pp. 301-304 (1981).
- 16) De Micheli, G., Hofmann, M., Newton, A. R. and Sangiovanni-Vincentelli, A. R. M.: A Design System for PLA-based Digital Circuits, Advances in Computer-Aided Engineering Design, Vol. 1, JAI Press, pp. 285-364 (1985).
- 17) Poretta, A., Santomauro, M. and Somenzi,

F.: TAU: A Fast Heuristic Logic Minimizer, ICCAD-84 pp. 206-208 (1984).

- 18) Bartholomeus, M. and De Man, H.: PRESTOL-II: Yet Another Logic Minimizer for Programmed Logic Arrays, ISCAS-85, pp. 226-229 (1985).
- 19) Sasao, T.: HART: A Hardware for Logic Minimization and Verification, ICCD-85, pp. 713-718 (1985).
- 20) Dagenais M.R., Agarwal, V.K. and Rumin, N.C.: The McBoole Logic Minimizer, 22-nd DAC, pp. 667-673 (1985).
- 21) Rudell, R.L. and Sangiovanni-Vincentelli, A.M.: Exact Minimization of Multiple-valued Functions for PLA Optimization, ICCAD-86, pp. 352-355 (1986).

デコーダ付 PLA

- 22) Sasao, T.: Multiple-valued Decomposition of Generalized Boolean Functions and the Complexity of Programmable Logic Arrays, IEEE Trans. Comput., Vol. C-30, No. 9, pp. 635-643 (1981).
- 23) 石川, 笹尾, 寺田: デコーダ付プログラマブル・ロジック・アレイの変数割当の一手法, 電子通信学会論文誌 (D), Vol. J 65-D, No. 5, pp. 590-597 (1982).

畳込み PLA

- 24) Hachtel, G.D., Newton, A.R. and Sangiovanni-Vincentelli, A.L.M.: An Algorithm for Optimal PLA folding, IEEE Trans on CAD, Vol. CAD-1, No. 2, pp. 63-77 (1982).
- 25) De Micheli G. and Sangiovanni-Vincentelli, A.L.M.: Multiple Constrained Folding of Programmable Logic Arrays: Theory and Applications, IEEE Trans. CAD, Vol. CAD-2, No. 3, pp. 151-167 (1982).
- 26) Egan J.R. and Liu C.L.: Bipartite Folding and Partitioning of a PLA, IEEE Trans. CAD, Vol. CAD-3, No. 3, pp. 191-199 (1984).

状態割当

- 27) 南谷 崇: 論理合成, 情報処理, Vol. 25, No. 10, pp. 1071-1077 (1984).
- 28) 当麻, 内藤, 南谷: 順序機械, 岩波書店, p. 272 (1982).
- 29) De Micheli, G., Brayton, R.K. and Sangiovanni-Vincentelli, A.L.M.: Optimal State Assignment for Finite State Machines, IEEE Trans. CAD, Vol. CAD-4, No. 3, pp. 269-285 (1985).
- 30) De Micheli, G.: Symbolic Design of Combinational and Sequential Logic Circuits Implemented by Two-level Logic Macros, IEEE Trans. CAD, Vol. CAD-5, No. 4, pp. 597-616 (1986).

検査容易化設計

- 31) 迫田, 石原: VLSI におけるマイクロプログラムの設計支援, 情報処理, Vol. 25, No. 10, pp. 1041-1047 (1984).
- 32) 樹下行三: テストと CAD, 情報処理, Vol. 25, No. 10, pp. 1112-1118 (1984).
- 33) 小嶋, 増井, 奥村: マイクロプロセッサのテスト手法とテスト容易化設計, 情報処理, Vol. 25, No. 10, pp. 1125-1130 (1984).
- 34) 笹尾 勲: 論理関数分解の一般化とその応用, 文部省科学研究補助金総合研究 (A), しこつこ研究会資料 (1986).

WE 32100

- 35) Wadsack, R.L.: Design Verification and Testing of the WE 32100 CPUs, IEEE Design and Test, pp. 66-75 (1984).
- 36) Jacobs, H.: Verification of a Second-generation 32-bit Microprocessor, IEEE Computer, Vol. 19, No. 4, pp. 64-70 (1986).
- 37) Law, H-F.S. and Shoji, M.: PLA Design for the BELLMAC-32 A Microprocessor, ICCD-82, pp. 161-164 (1982).
- 38) Lopez A.D. and Law, H-F.S.: A Dense Gate Matrix Layout Method for MOS VLSI, IEEE J. Solid-State Circuits, Vol. SC-15, pp. 736-740 (1980).

MC 68020

- 39) Daniels R.G. and Bruce, W.C.: Built-in Self Trends in Motorola Microprocessors, IEEE Design and Test, pp. 64-71 (1985).
- 40) Kuban, J. and Salick, J.: Testability Features of the MC 68020, Proc. IEEE Inter. Test Conf., pp. 821-826 (1984).

Micro/370

- 41) Chao, H.H., Ong, S., Lewis, K. and Tang, J.Y.: Design Optimization for Control Circuits Implemented with PLAs, ICCD-86, pp. 526-529 (1986).
- 42) Chao, H.H., Ong, S., Tsai, M.Y., Shih, F-H.W., Hou, J.C.L., Lewis, K., Tang, J-Y. F., Trempel, C.A., Hadsell, R.W., Yu, N.W., Greier, P.F., Franch, R.L., McCormick, P.E., Davis, Jr, C.V., Diamond, A.L. and Medve, T.J.: Micro/370: A 32-bit Single-chip Microprocessor, IEEE Journal of Solid-State Circuits, Vol. SC-21, No. 5, pp. 733-740 (1986).

SOAR

- 43) Pendleton, J.M., Kong, S.I., Brown, E.W., Dunlap, F., Marino, C., Ungar, D.M., Patterson, D.A. and Hodges, D.A.: A 32-bit Microprocessor for Smalltalk, IEEE Journal of Solid-State Circuits, Vol. SC-21, No. 5, pp. 741-749 (1986).
- 44) Newton, A.R.: Techniques for Logic Synthesis, VLSI-85, pp. 33-48, IFIP (1986).

80386

- 45) Gelsinger, P.P.: Built In Self Test of the 80386, ICCD-86, pp. 169-173 (1986).

(昭和 62 年 2 月 5 日受付)