



技術解説

# プログラマブル・ロジック・デバイス

笹尾 勤\*

## 1. PLDとは

最近、プログラマブル・ロジック・デバイス (Programmable Logic Device: PLD) という技術が雑誌等で話題になっている。これは、論理設計者が現場でパーソナルコンピュータと専用の書き込み器を使って、自由にプログラムできる論理素子の総称であり、図1・1に示すように PROM (Programmable Read Only Memory), FPLA (Field Programmable Logic Array), 及び PAL (Programmable Array Logic) などに分類できる。

たとえば、表1・1の真理値表に示す論理関数を、従来のように NAND あるいは NOR ゲートを用いて構成すると、図1・2のように

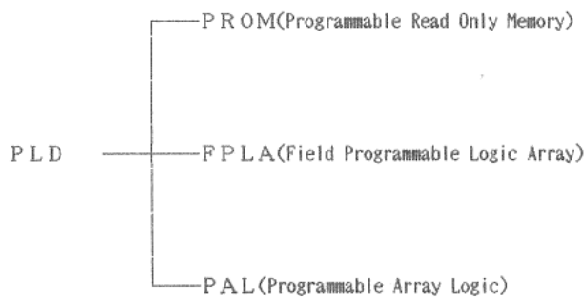


図1.1 論理構成による PLD の分類

表1.1 全加算器の真理値表

$x_2$	$x_1$	$x_0$	$f_1$	$f_0$
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

なる。この回路は、NOR ゲートの個数が最小になるように設計したものであるが、非常に込み入っている (これをランダム論理回路という)。

真理値表が与えられたとき、ゲート数が最小であるようなランダム論理回路を自動的に求める実用的方法は知られておらず、現在の技術では、大型計算機を用いても困難である。ふつうランダム論理回路は人手により設計されているが、人手設計は時間がかかり、誤りも生じやすい。

最近では、ゲートの個数を減らすことよりも、

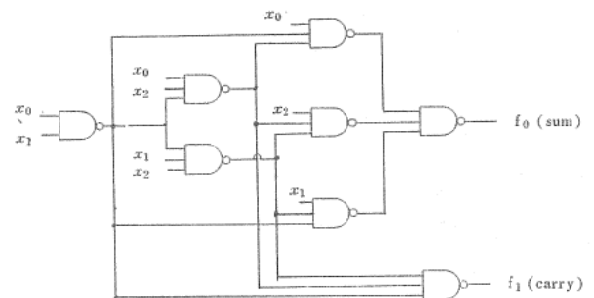


図1.2 表1.1を実現するランダム論理回路

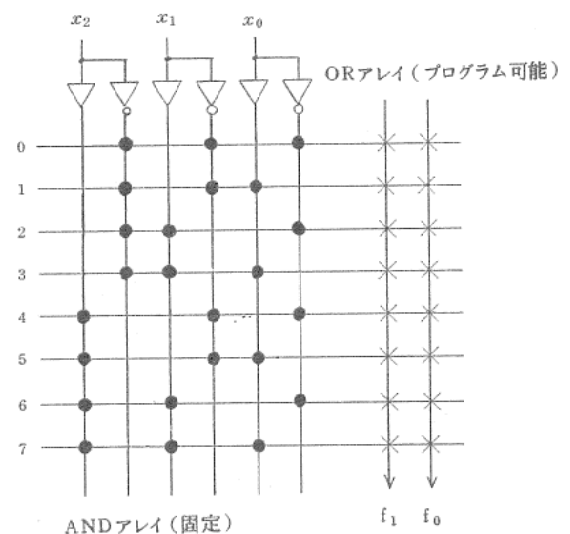


図1.3 (a) PROM

\*笹尾 勤 (Tsutomu SASAO), 大阪大学工学部, 電子工学科, 助手, 工学博士, 電子工学

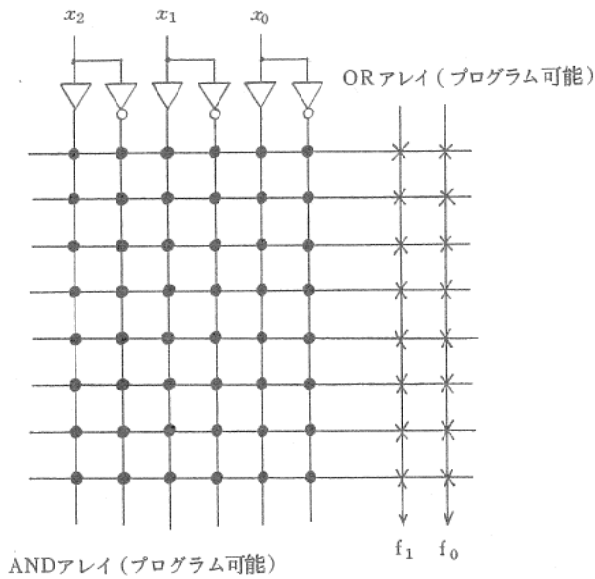


図1.3 (b) FPLA

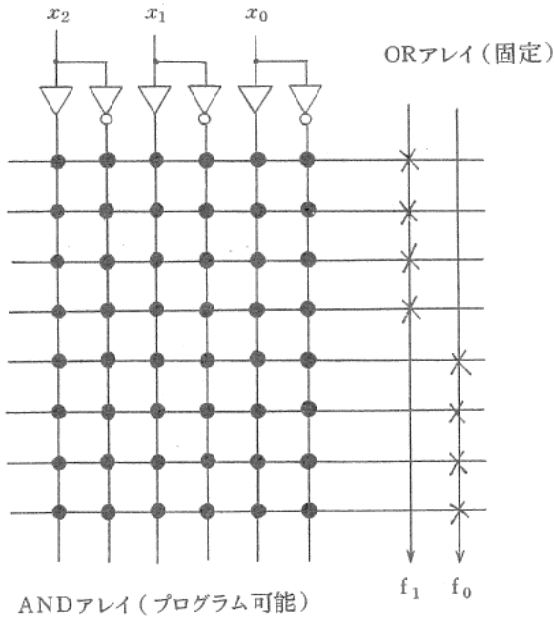


図1.3 (c) PAL

図1.3 PLD の構造

誤りのない論理回路を短期間に開発することが重要となり、図1.3に示すような規則的構造をした PLD が利用され始めた [1].

20ピンの PLD の場合、3~15個程度の MSI/SSI を一つのパッケージに置換えることができる。

## 2. PROM

図1.3 (a) に PROM の概念図を示す。PROM では、AND アレイの部分は固定され

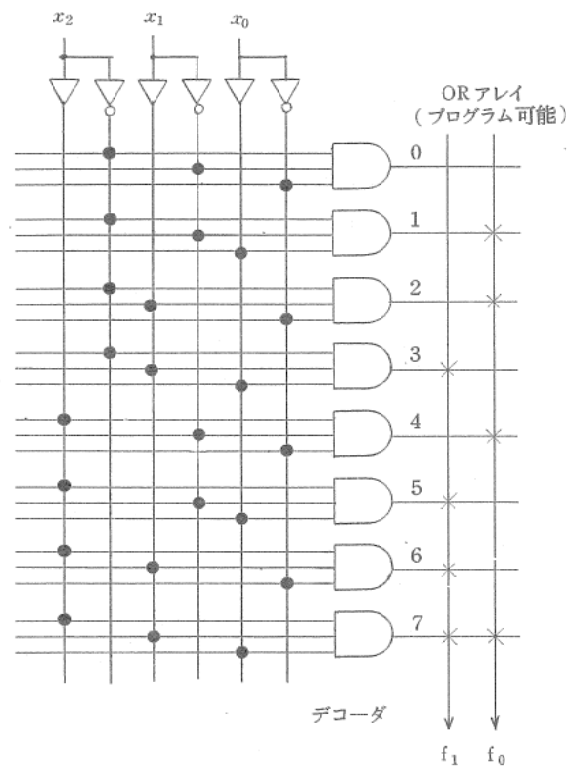


図2.1 PROM による実現

ており、デコーダとよばれている。例えば、表1.1の真理値表は、図2.1のように実現できる。PROM の場合、OR アレイのみがプログラム可能である。PROM を論理回路の代わりに使用する場合、マイクロコンピュータのプログラムやデータを貯えるための MOS 型素子は遅いので、バイポーラ型の高速の素子を用いる。PROM の論理設計は極めて簡単で、表1.1の真理値表をそのままプログラムすればよい。

PROM の欠点は、入力変数の個数を  $n$  とすると、アレイの大きさが  $2^n$  に比例すること、及び、ランダム論理回路の代りに用いる場合、ハザードが生じることである。PROM が市場に出たのは1970~1971年頃である。

## 3. FPLA

図1.3 (b) に FPLA の概念図を示す。FPLA は、AND アレイと OR アレイから構成されており、両方のアレイがプログラム可能である。PLA は1970年頃 TI 社 (Texas Instruments Incorporated) により考案されたが、当時はマスクでプログラムを行うため、少なくとも数千個以上生産を行う製品にしか適用出来

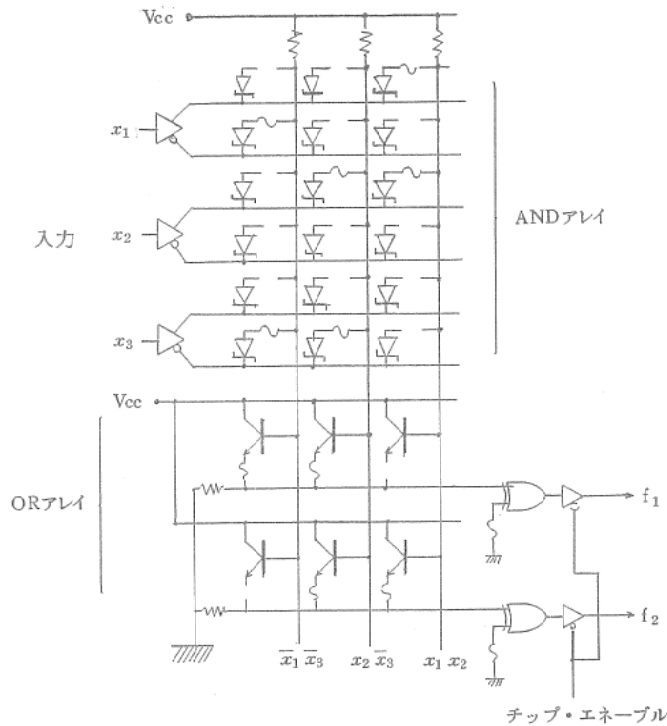


図3.1 フューズ型 FPLA の例

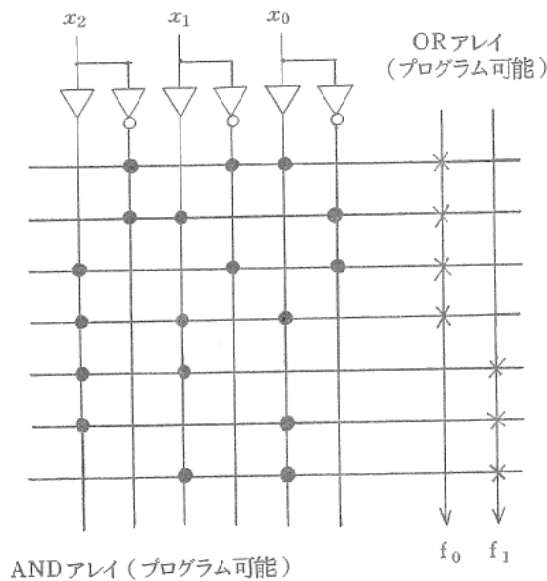


図3.2 PLA での実現

なかった。1975年になって、シグネティクス社から、PROM のヒューズ技術を用いた図3・1のような現場でプログラム出来る PLA (FPLA) が発売された。これは、従来のマスク PLA とは異なり、試作品をただ一つ製作する場合にも適用できるという画期的なものであった。たとえば、表1・1の真理値表は図3・2のような FPLA で実現できる。ただ、発売さ

れた当初は、まだ FPLA は高価で、遅延時間も大きく (50ns)、また論理設計も従来の TTL ゲートに馴染んだ技術者には取付きにくかったため、それほど普及しなかった。

PLA の論理設計は、形式的に述べると、“与えられた真理値表を実現する AND-OR 二段多出力論理回路の設計”という古典的問題となる。これは、古くから研究されている基本的問題の一つであるが、ここ数年間のめざましい研究成果により、入力変数が100程度までなら、大型計算機を使用すればほぼ最適解に近いものが、実用時間内に求まるようになった[2]。また、市販の FPLA に収まる程度の問題 ( $n=20$ 程度) なら、16ビットのパーソナルコンピュータで簡単化できるようになっている。そして安価な論理設計支援プログラムも市販され始めたため、FPLA の設計は誰でも行えるようになってつつある。

#### 4. PAL

図1・3 (c) に PAL の概念図を示す。PAL はプログラム可能な AND アレイと、固定された OR アレイから構成されている。この素子は 1978年に MMI 社 (Monolithic Memories

Inc.) から発売された。この素子は FPLA で OR アレイを固定したものであるが、OR アレイを固定したため、FPLA より高速で安価になった。また、使い方が TTL ゲートと似ている

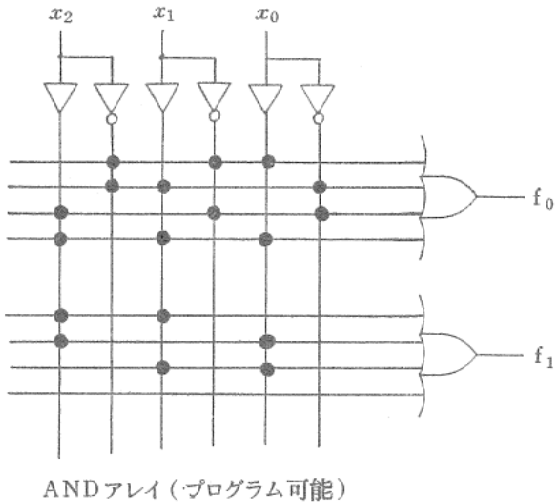


図 4.1 PAL での実現

ため、多くの技術者に取っ付きやすく、またパーソナルコンピュータ上で動く論理設計サポートプログラム PALASM (PAL Assembler: パラズム) を PAL の発売と同時に公開したため、PAL は短期間内に多くの人々に用いられるようになった。特に、米国某社のミニコンには PAL が大量に使用されたという。当初、MMI 社はシグネティクス社と同じ FPLA を発売する予定で、そのデータシートまで用意し

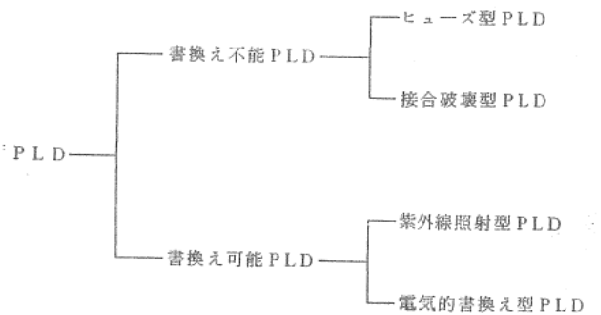


図 5.1 プログラム法による PLD の分類

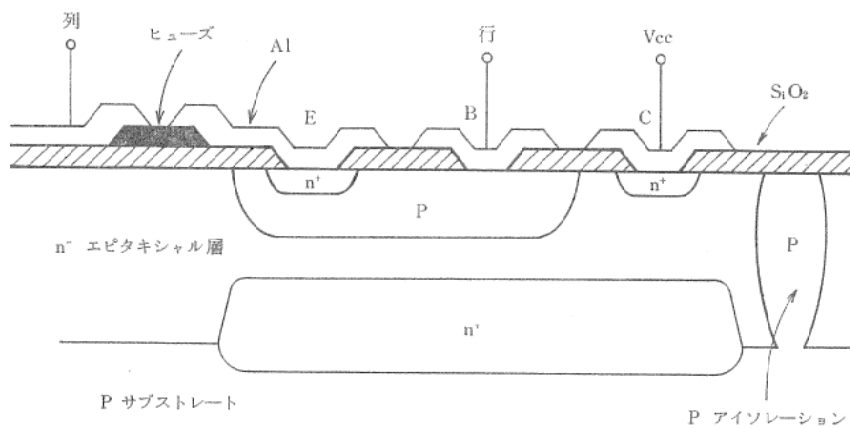
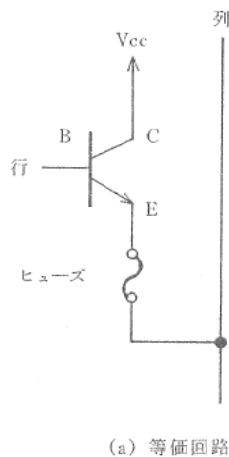


図 5.2 ヒューズ型素子の構造

ていたが、急きょ OR アレイ固定の PAL に変更したのは、経営者の優れた判断といえる。

PAL では、OR アレイが固定しているため、FPLA ほど設計の自由度がない。そのため、多数のファミリ素子を用意してその欠点を補っている。PAL の論理設計は、FPLA より簡単であり、各出力関数を別々に単純化すればよい。通常のもは、パーソナルコンピュータ上で設計できる。表 1・1 の関数は図 4・1 の PAL で実現できる。

### 5. プログラム法による PLD の分類

図 5・1 は、PLD をプログラム方式から分類したものである。このうち、現在よく用いられているのは書換え不能な素子で、ヒューズ型素子 (図 5・2) と接合破壊型素子 (図 5・3) に分類できる。ヒューズ型素子の場合、製品出荷時は各交点にヒューズが集積されており電氣的に接続している。プログラム時に、交点に大電流を流すことによってヒューズを溶断し

個別化を行う。一方、接合破壊型素子では、製品出荷時は各交点に逆接合したダイオードが集積されており、電氣的に分離している。プログラム時に、交点に高電圧を加えることにより、一方のダイオードを短絡し、個別化を行う。この場合、個別化を行うと交点は電氣的に接続する。

### 6. PLD 設計システム

PLD を製品開発に使用している大きな企業 (例えば、ATT, GE, IBM, NEC, NTT, RCA 等) では、その設計システムを自社で開発し、社内で使用している。これらのシステムでは、高価な計算機を使用し、ソフトウェアは非公開である。そのため、そういうシステムを持たないグループは PLD を使用するのが困難であった。しかし、ここ数年の間に、安価で高性能な 16 ビットパーソナルコンピュータ、および標準 OS (MS-DOS) 上で動作する PLD 設計用ソフトウェア/PLD 書き込み器が市販され

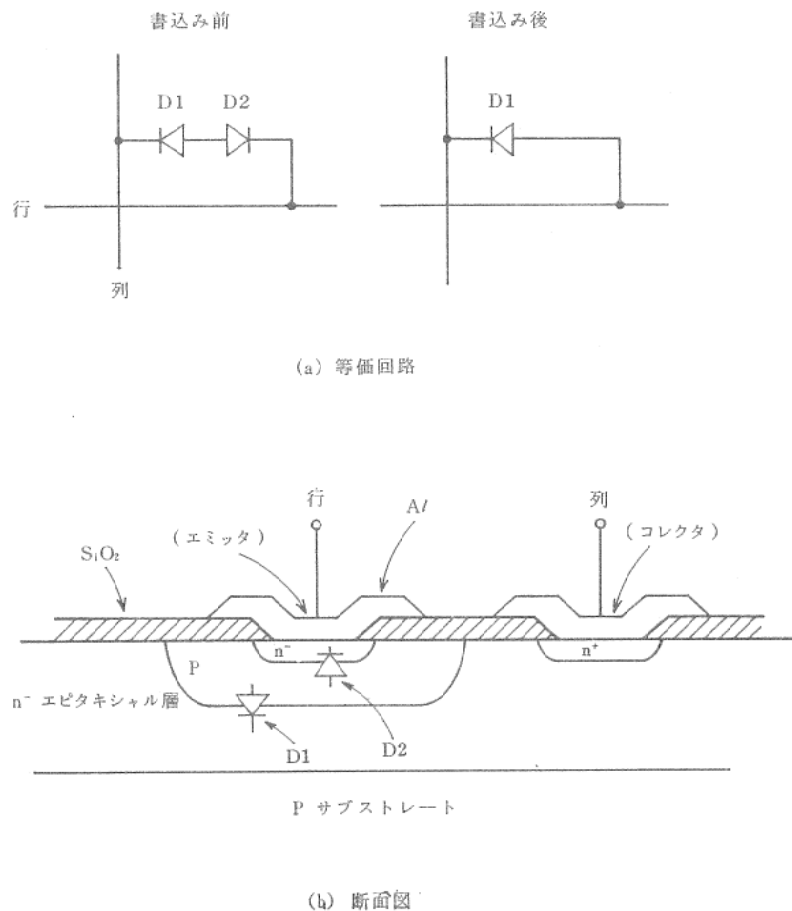


図 5.3 接合破壊型素子の構造

始めた。これらのシステムを用いると、個人や小さなグループでも PLD を用いた製品開発ができる。

PLD を用いて効果的に論理設計を行うには、以下に示すようなパーソナルコンピュータを用いた設計開発システムが必要である。

- 1) 16ビットパーソナルコンピュータ (MS-DOS が動作可能なもの)
- 2) フロッピーディスク・ドライブ
- 3) プリンタ
- 4) PLD 設計用ソフトウェア
- 5) PLD 書込み器

このうち、1)～3) は手持ちのものを使用すればよい。4) のソフトウェアのうち、PAL 設計専用のソフトウェア (PALASM:パラズム) は MMI 社が公開しているので、ディスク代程度で入手できる。また、RPOM, FPLA, PAL のすべてが設計でき、論理式簡単化プログラムの附属したものは、現在30万円程度で販売されている。5) の書込み器は、PAL や FPLA 専用の簡易型は10万円以下で入手できるが、汎用のものは50～200万円程度である。

### 7. PLD設計ソフトウェアの概要

市販の PLD 設計ソフトウェアは、以下の機能を備えている。

A. 記述された論理の PLD への変換  
設計すべき論理回路の記述方法として、

- 1) 論理回路図 (ネットリスト)
- 2) 論理式
- 3) 真理値表
- 4) 代数式
- 5) 状態遷移表 (ステートマシン)

などがある。

例えば、以下に示すような二つの2ビットの数の加算を行う論理回路を考えてみよう。

$$\begin{array}{r}
 \phantom{+)} \quad x_1 \quad x_0 \\
 +) \quad y_1 \quad y_0 \\
 \hline
 z_2 \quad z_1 \quad z_0 \text{--- (sum)} \\
 \phantom{z_2} \quad c_1 \quad c_0 \text{--- (carry)}
 \end{array}$$

論理回路図による入力

全加算器 (full-adder: 表 1・1) をモジュールと考え、図 7・1 の記号で表すと、2ビット

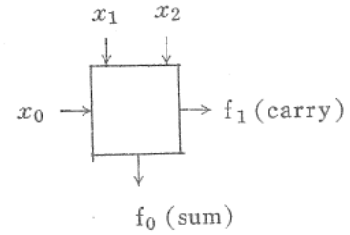


図 7.1 全加算器

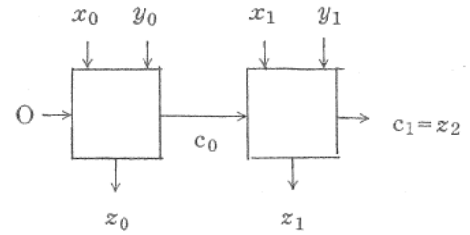


図 7.2 2ビット加算器

ト加算器は、図 7・2 のように実現できる。この方法は、既存の回路を PLD 化する場合はもちろんのこと、新規に設計する場合でも、従来の SSI/MSI を用いた設計に馴染んだ設計者には極めて分かりやすい。

論理式による入力

2ビット加算器を論理式で表現すると、

$$z_0 = x_0 \vee y_0$$

$$c_0 = x_0 \cdot y_0$$

$$z_1 = x_1 \oplus y_1 \oplus c_0$$

$$z_2 = c_1 = x_1 \cdot y_1 \vee c_0 \cdot (x_1 \vee y_1)$$

となる。ここで、 $c_0$ 、 $c_1$  は中間変数である真理値表

表 7・1 は 2ビット加算器の真理値表である。真理値表による記述は、符号変換器、数表

表 7.1 2ビット加算器

入力				出力		
$x_1$	$x_0$	$y_1$	$y_0$	$z_2$	$z_1$	$z_0$
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	1	0
0	0	1	1	0	1	1
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	0	1	1
0	1	1	1	1	0	0
1	0	0	0	0	1	0
1	0	0	1	0	1	1
1	0	1	0	1	0	0
1	0	1	1	1	0	1
1	1	0	0	0	1	1
1	1	0	1	1	0	0
1	1	1	0	1	0	1
1	1	1	1	1	1	0

などに便利である。

代数式で表現すると

$$Z = [z_2, z_1, z_0]$$

$$X = [x_2, x_1, x_0]$$

$$Y = [y_2, y_1, y_0]$$

$$Z = X + Y$$

となる。このように設計すべき回路が代数式で表現できる場合には、この記述法は極めて簡潔である。

#### B. 論理式簡単化

従来は人手で行っていたので時間がかかり、誤りも多かったが、コンピュータを用いれば、短時間に誤りなく設計できる。

#### C. 論理シミュレーション

これは、記述した論理が設計者の意図する通り動作するかを確認する操作であり、記述誤りや設計誤りの多くが検出できる。また、デバイス書込み時に行うヒューズの検査入力や動作試験のための検査入力も生成する。

#### D. ドキュメンテーション

論理設計の記述、ヒューズパターン、ピン配置図などの設計資料を自動的に生成する。E.

#### E. フューズパターン生成

各種 PLD 書込み器に適合するフォーマットを指定できる。

### 8. PLDの応用例

PLD は、多品種少量生産品の製作に特に有効である。ここでは、著者が研究用に試作した PLA 簡単化用並列プロセッサについて紹介する。先に述べたように、市販の FPLA 程度の大きさの PLD ならばパーソナルコンピュータで簡単化できる。しかし、VLSI で使用する大規模 PLA は、計算時間や、メモリ容量の点からパーソナルコンピュータでは、簡単化は困難である。著者は、パーソナルコンピュータ上で

大規模 PLA の簡単化を可能にするため、PLA 簡単化アルゴリズムのうち、最も時間のかかる部分を並列演算ハードウェアで実行し、計算時間を短縮する方法を考案した [3]。その部分をランダム回路で設計すると、2入力ゲートが 1000個以上必要であるが、同じ機能を FPLA を用いて設計しなおすと、20ピン FPLA 43個と若干のバッファ用 IC で構成できた。システムは、23cm×16cm のボード1枚の上に搭載でき、きわめて小型になった。ただし、発熱が大きい (30w以上) ため冷却用ファンを2個用いた。システム完成後数カ月して、機能追加のために回路の変更を行ったが、その際、基板の配線は変更の必要がなく、FPLA を一個焼きなおすだけ (半日) で作業が終った。

以上のように、PLD は試作用には最適の素子であり、PLD を用いると、パッケージ数、配線、ハンダ付が大幅に減り、回路もパーソナルコンピュータのファイル上で管理出来るため、開発、保守の時間が大幅に削減できる。

### 9. おわりに

以上、PLD の歴史、論理構成、デバイス、および設計用サポートツールについて手短かに述べたが、本稿が読者のお仕事に少しでもお役に立てば幸いである。

#### 参考文献

- 1) 笹尾 勤, "PLA の使い方・作りかた" 日刊工業新聞社 (近刊).
- 2) R.K. Brayton et al, "Logic minimization algorithms for VLSI synthesis," Kluwer Academic publishers, Boston, 1984.
- 3) T. Sasao, "HART: A hardware for logic minimization and verification," International Conference on Computer Design: VLSI in Computers, Oct. 7~10, 1985.