

プログラマブル順序回路素子(ピコ・プロセッサ)の一提案†

姜 哲 熙** 富 永 英 義**

あるデジタルシステムの制御構造をプログラマブルロジックにより設計しようとした場合、設計の対象となる回路がある程度複雑で、その規模が小さくないときは市販のビットスライス形シーケンサを用いてコスト/パフォーマンスを上げることができる。しかし、比較的小規模で、高速な制御回路を対象とした場合は、ビットスライス形シーケンサとその周辺回路がコストの面で大きな負担となる。そのうえ汎用性を持たせた構造そのものから起因する問題として、設計が煩雑になることと冗長な機能を多く含むことなどが考えられ、結果的にプログラマブルロジックを用いて制御構造を設計したことがシステム全体のコスト/パフォーマンスを悪くすることが十分考えられる。

本稿では、このような問題を十分考慮したうえ設計した汎用プログラマブル順序回路素子であるピコ・プロセッサについて論じる。ピコ・プロセッサは、1) 高速でプログラムを実行可、2) プログラミングが容易、3) 単純な構造、4) すぐれた可用性などの特徴を有するシーケンサである。本稿の後半では、ピコ・プロセッサを CRT ディスプレイの制御部に用いたときのシステム設計例を示している。最後に、市販のビットスライス形シーケンサとピコ・プロセッサとを比較しながら考察を行っている。

1. ま え が き

μ -CPU の出現は、あらゆるデジタルシステムの従来の機能回路(ワイヤードロジック)部分をプログラマブル論理により実現する道を開いた。特に、ビットスライス(以後、BS と略す)形シーケンサの出現は、比較的高速で複雑な制御構造を容易に実現することを可能にした。

しかし、これらの μ -CPU は (BS 形シーケンサも含む)、主に小形電子計算機の制御回路部を意識して設計されたものであるので、**小形で高速な*論理回路**をこれらの素子で設計しようとする、次のような問題に直面する。

- i) 複雑なプログラム構造とハードウェア構造により設計が煩雑になる。
- ii) 汎用性の機能が冗長になり、むだなハード量を導入する結果となる。
- iii) 経済的でなくなる。

最近の μ -CPU は、LSI 技術の進歩に伴い、より高集積化され、ミニ・コンピュータの CPU に匹敵するものまで開発されつつあるが、上で指摘した問題の解決にはならず、むしろより根本的な解決策が望まれる。

一方、ワンチップ・マイクロコンピュータはこのような問題を解決するために開発されたものの一つであ

ると考えられ、低速な制御回路の設計には非常に有効である。ところが、これも高速制御回路の設計には実行速度が遅い理由からほとんど用いられない。

制御回路をモジュール化して実現する試みは、 μ -CPU、シーケンサのほかにも種々あるが^{1)~6)}、最も実用化されているものとして PLA がある^{7), 8)}。PLA を用いた際、シーケンス制御を行うことが不便であったが、これを解消するため考察された F/F 内蔵形の PLA に関する報告が Horninger⁹⁾、Wood¹⁰⁾ らにより行われた。しかし、設計に際しては、PLA の内部構造のために、積和の論理式を求めた上で状態 F/F の割付けを行うことが必要であり、見通しの良い手順で設計を行うには不向きな面をもっている。

以上のように、**小形で高速な制御論理**を有する論理回路を設計するのに適した素子は少ない。よって、研究あるいは開発時(少量かつ多種であったり、機能が頻繁に修正されるようなシステム)の論理回路(特に、順序回路)の設計が迅速に、見通し良くできる汎用のプログラマブル順序回路素子の出現が待たれる。

本稿では、このようなことを十分考慮に入れて設計した汎用プログラマブル順序回路素子として、**ピコ・プロセッサ**を提案し、そのアーキテクチャ、動作原理等について述べ、応用例も示す。最後に、BS 形シーケンサとピコ・プロセッサとを比較しながら考察を行う。

2. 問題設定

一般のデジタル制御回路は制御構造とデータ構造に分離して考えることができる¹⁴⁾。例えば、図 1(a)

† A Proposal for a Programmable Sequencer, *picco*-Processor by CHUL-HEE KANG and HIDEYOSHI TOMINAGA (School of Science and Engineering, WASEDA University).

** 早稲田大学理工学部

* 数十チップの SSI か MSI により構成される大きさを、1 マシンサイクル約 100~200 ns で実行できるシーケンサを用いれば処理できる論理回路。

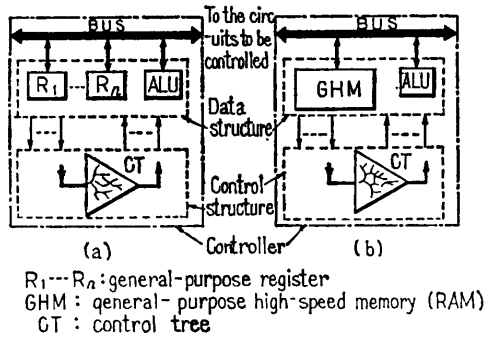


図 1 一般的な制御回路の構成
 Fig. 1 A certain controller.

のように、データ構造としてはレジスタ群が対応し、制御構造としてはレジスタ間のゲートを制御するためのタイミング信号を作る順序回路が対応する。現状では、図 1(b)のように、データ構造として汎用高速メモリ素子を適用することにより、大容量化、高速化することが可能であるが、制御構造としては前章で述べたように適切なものが少ない。

基本制御構造をモジュール化する場合、従来の論理構造をそのまま LSI 化する方法がある。この場合、モジュールが専用となり、ゲート/ピン比からも集積度の制約を招くので好ましくない。一方、既存の(μCPU や BS 形シーケンサ等の) 高集積素子を利用した場合には、汎用性は得られるが、素子の構造と実行速度により、高機能化、経済化への制約が生じる。

したがって、高速性、汎用性、可用性等がうまく設定された汎用プログラマブル順序回路が実現できれば、高速制御回路のモジュール化された構成が可能であり、見通しの良い設計を行うことができる。

素子の汎用性と可用性を高めるため、蓄積プログラムによる制御法を取った場合、プログラム・ステップの実行回数により所望する機能の実行速度が影響を受ける。

すなわち、データをより高速に処理するためには、サイクルタイムを速くするかプログラムステップ数を減らすことが必要である。このことは、1 サイクルでの処理機能を増やすことによりステップ数を減少させれば、実行速度を上げることと同等な効果を得ることを意味するものである。

以上述べてきた問題意識より、高速制御回路用の汎用プログラマブル順序回路素子には、次のようなアーキテクチャが要求される。

i) 高速性

- 1 命令は 1 マシンサイクルで実行する。(処理量/サイクルを増やすことと同等の効果)
- (集積密度) × (遅延速度) の少ない高速デバイス技術を導入する。(例えば、SOS* 等)

ii) 汎用性

- 蓄積プログラムによる制御方式をとる。
- マイクロプログラミング法を取り入れ、コントロールツリーと出力データをプログラマブルに変更できるようにする。

iii) 単純構造化

- 制御機能として必要な最小の命令数にする。
- 機能的には、外部条件の入力端子とデータの出力端子のみを備える。
- メモリ構造のような反復的論理構造を多用する。

iv) 可用性

- 外部との同期のためにウェイト機能を有する。(複数のモジュールを構成するとき有効となる)
- 最小レベルのサブルーチン機能とインタラプト機能を有する。
- リセット機能を有する。

次章では、以上のようなアーキテクチャのもとに考案された汎用プログラマブル順序回路素子であるピコ・プロセッサ (以下、*p-p* と略す) について詳しく述べる。

2. ピコ・プロセッサ

3.1 概要

p-p は図 2 に示すように、内部マイクロプログラムを記憶するメモリを持ち、外部回路の状態を取り入れ、条件 F/F をテストする機能を有する。さらに、外部回路との同期をとるとき、*p-p* を複数個動作させるときに有効なウェイト機能を持つ。

外部からの条件が満たされたときは、実行速度の高

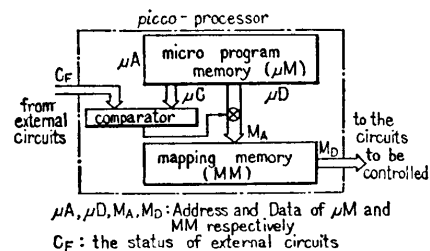


図 2 ピコ・プロセッサの基本構成
 Fig. 2 The configuration of *picco*-processor.

* Silicon-On-Sapphire

F function (2 bits)	Wait (1 bit)	Condition (n bits)	X (operand) (m bits)	
mnemonic	b ₁₁ b ₁₀ b ₉ b ₈ b ₇ b ₆ b ₅ b ₄ b ₃ b ₂ b ₁ b ₀	function		
OUT	0 0	W C C C C	X X X X X	OUTPUT
OTR	0 1	W C C C C	X X X X X	OUTPUT & RETURN
JMP	1 0	W C C C C	X X X X X	JUMP
JSR	1 1	W C C C C	X X X X X	SUBROUTINE JUMP

(CCCC)=(0000): non-condition

図3 ピコ・プロセッサの命令形式
Fig. 3 Instruction sets (12 bit-word).

速化を計るため、1マシンサイクルで、マイクロプログラムに従った情報をマッピングメモリに写像し、そのデータを外部に出力する。

μP. の命令セットは、OUT, JMP, OTR (output and return), JSR (jump to subroutine) から成る。ただし、この四つの命令にはそれぞれ無条件と、条件付きの場合のウェイト指定の有無の3通りがあるので、実際は12個となる。

3.2 機能

μP. はプログラム制御と入出力だけを行い、演算機能は外部にオプションで付けるハードウェアにまかせる。

命令形式は図3のようになっている。Fは操作部で、命令の機能を2ビットで四つまで指定することができる。Wのウェイト指定部は1ビット構成で、指定された場合は条件が満たされるまでプログラムの進行が停止される。条件指定部Cは、外部回路の状態を保持している条件F/Fを指定するものである。nビット構成の場合、2ⁿ個の条件F/Fをテストすることができる。Xはオペランドであり、マッピングメモリのアドレス指定に使われる。マッピングメモリにはジャンプ先アドレス、または出力されるデータが記憶されている。

図4に各命令の詳細について説明する流れ図を示す。

3.3 ハードウェア

μP. は三相のクロックに同期して動作するマイクロプログラムメモリ、マッピングメモリ、条件判定部、スタック等から成り、マイクロプログラミング法^{12),13)}による制御を行う。

図5、図6に示した、12ビット語長のμP. のクロックタイミング図とブロック図を参照しながら、μP. の内部回路動作

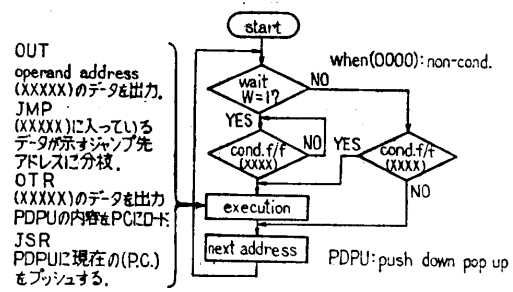
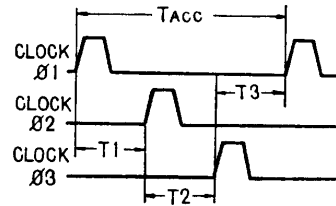


図4 各命令の流れ図
Fig. 4 Flow of instructions.



Tacc: 1命令サイクルタイム

図5 ピコ・プロセッサのクロックタイミング図
Fig. 5 Timing diagram of picco-processor.

について紹介する。

リセットによりクリアされたプログラムカウンタ(PC)の内容は、φ1の立ち上がりでマイクロプログラムメモリをアドレッシングする。読み出された命令のOPは、φ2の立ち上がりで命令解読のためDECに入れられ、四つの命令に分類される。C/F/Fの4ビットは、外部からの状態情報によってセットされたレジスタと比較され、一致した場合はオペランドイネーブル(OE)を出す。このOEは、マッピングメモリをアドレッシングするOPRの5ビットを制御する信号とな

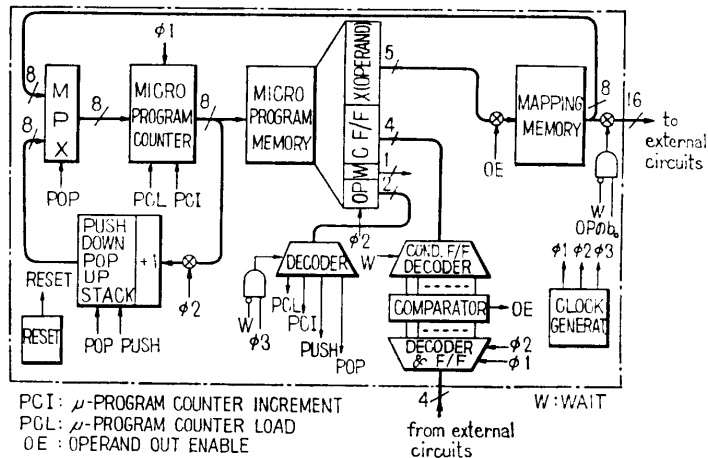


図6 12ビット語長のピコ・プロセッサの構成図
Fig. 6 Hardware configuration of 12 bits-word picco-processor.

る。Wの1ビットはプログラムカウンタインクリメント(PCI)をマスクすることにより、プログラムの進行を制御する。PCロード(PCL)はOUT命令を除いた三つの命令が解読されたときに、φ3のタイミングで出される。OUT, OTR命令が解読されたときは、ウェイト指定をうけてないかぎり、直ちに出力ゲートは開かれ、φ3でPCIが送出される。POP, PUSH信号は、それぞれOTRとJSR命令がデコードされたときに発せられる。すなわち、インクリメント(+1)にφ2でゲートされて入れられた(PC)は“+1”され、φ3で出されるPUSH信号によりプッシュダウンポップアップスタック(PDPU)にストアされる。また、(PDPU)はφ3のPOP信号に制御され、PCLによりPCに復帰される。

3.4 チップ構成

μ-p. をチップ化する際に、ピン数の制限により処理能力が限られてしまう。図3に示した図から、μ-p. の順序回路としての能力について次のように検討する。すなわち、マッピングメモリの大きさは順序回路で取り得る独立なF/Fの個数 M_D bitとそれが出現するパターン数 2^{M_A} により決められる。さらに、マイクロプログラムメモリについては、シーケンスの長さがアドレスの数 $2^{\mu A}$ に対応し、語長は外部からのテスト条件の数 $2^{C_F}(=2^{\mu_e})$ と M_A を規定するものとなる。したがって、μ-p. のチップ化に際しては、処理能力とピン数との関係を十分検討したうえで行う必要がある。

表1は、40ピンLSIにしたときの16ビット命令語長のμ-p. の処理能力を示す一例である。

表1 40ピンのLSIにしたときのピコ・プロセッサのピン割当表
Table 1 A pin-allocation for picco-processor on 40-pin LSI chip.

input pins (TEST terms)	output pins (DATA terms)	miscellany (V _{cc} , GND, CS, PROG, RST, etc.)
5 pins (32 terms)	24 pins (24 terms)	11 pins

where, CS: chip select, PROG: control for programming (only field type), RST: reset.

ータ構造を示すものであり、例えば、リターンアドレスレジスタ (AR)

現在ディスプレイしている行の先頭の文字が記憶されたキャラクタバッファメモリ (CBM) のアドレスをその行の走査が終るまで保持するレジスタ
アドレスカウンタ (A1)

ARより渡されたアドレスを順次アップしながら、CBMをアドレッシングするカウンタ

インデックスアドレスレジスタ (IA)

3ビット構成のアップカウンタであり、ディスプレイするラインのアドレスを指定するカウンタ ($\langle 000 \rangle_2$ のときIA₀の信号が出る。) などのようなものがある。

一方、ディスプレイの制御構造は、図8に示す制御の流れにより表現される。これは、そのままμ-p. のマイクロプログラムに対応するものであり、図10は試作したディスプレイ制御のμ-p. によるマイクロプログラムリストである。

ところで、プログラマブルな論理素子を用いて、ラインバッファを持たないCRTディスプレイの制御構

4. ピコ・プロセッサによる制御回路の設計例

前章で述べたμ-p. の論理機能を実際に確めるため、μ-p. の試作を行った。μ-p. は高速デバイス技術によりLSI化されることが本来のねらいであるが、大学における試作では汎用のショットキーTTLのSSI, MSIを用いてパッケージを組立て、チップ機能を模擬した。

本章では、このμ-p. を用いてCRTキャラクタディスプレイ(以下、ディスプレイと略す。)の制御部を設計する例を示すこととする。

μ-p. による制御論理を説明する前に、μ-p. を用いたときのディスプレイのデータ構造を明らかにする必要がある。図7はμ-p. の制御構造を含めたデ

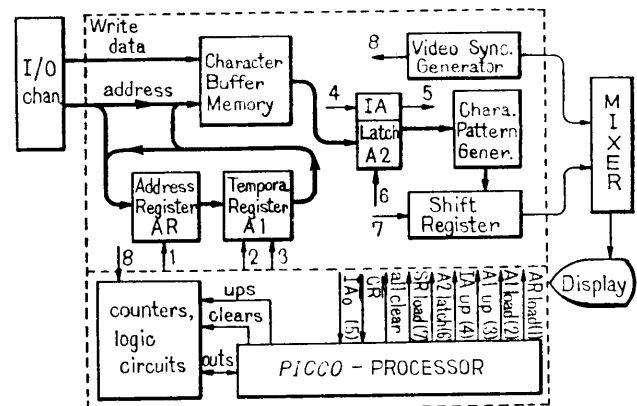


図7 ピコ・プロセッサの制御によるCRTディスプレイシステムの構成図
Fig. 7 Block-diagram of CRT display controlled by picco-processor.

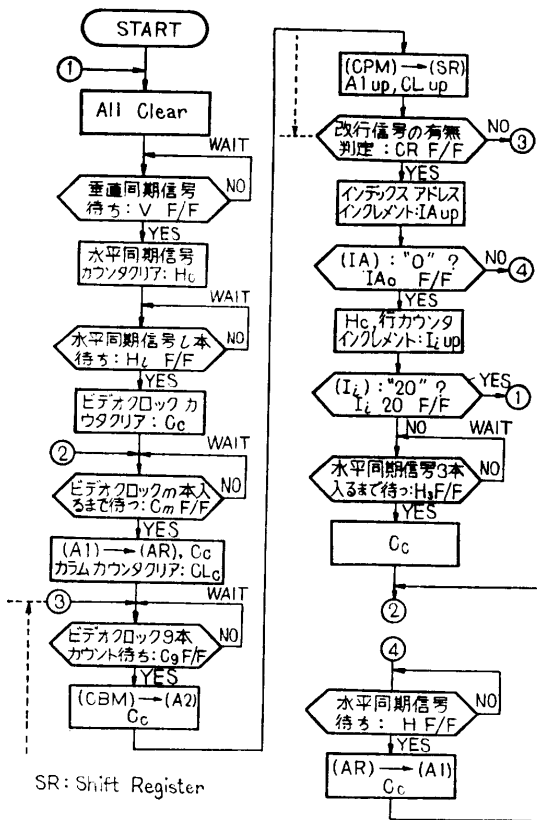


図 8 CRT ディスプレイの制御の流れ図
Fig. 8 Control sequence for a CRT display.

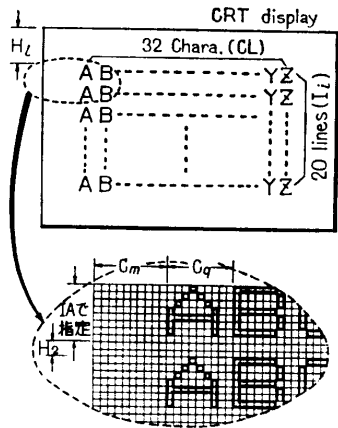


図 9 文字ディスプレイの模写図
Fig. 9 An appearance for characters displayed.

造を実現する場合¹³⁾, 次の関係式により示される時間的制限が存在する. すなわち,

$$mT_{Acc} < nT_{vc} \quad (1)$$

ただし, m : スペースを含めた 1 文字を横方向に走査する機能を実現するシーケンサのプログ

μ -Prog. mem. add.	Function	Wait	Condition	Operand
0	OUT			ALL CLEAR
1	OUTC	W	$V_{F/F}$	Hc
2	OUTC	W	$H_{lF/F}$	Cc
3	OUTC	W	$C_{mF/F}$	(A1) \rightarrow (AR), Cc, CLc
4	OUTC	W	$C_{gF/F}$	(CBM) \rightarrow (A2), Cc
5	OUT			(CPM) \rightarrow (SR)
6	JMFC		$\overline{C}R_{F/F}$	A1 up, CL up
7	OUT			4
8	JMPC		$\overline{I}A_{0F/F}$	IA up
9	OUT			13
10	JMPC		$I_{i0F/F}$	Ii up, Hc
11	OUTC	W	$H_{lF/F}$	0
12	JMP			3
13	OUTC	W	$H_{F/F}$	(AR) \rightarrow (A1), Cc
14	JMP			3

図 10 ディスプレイを制御する場合のピコ・プロセッサのマイクロプログラム
Fig. 10 *Picco*-processor's micro-program for controlling the CRT display.

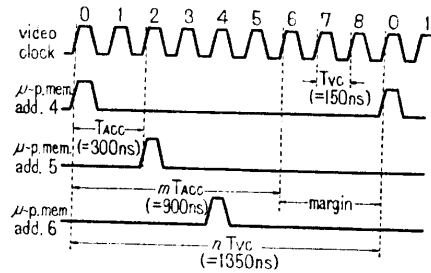


図 11 プログラム制御のボトルネックとなる部分のタイミング図
Fig. 11 Timing diagram of program sequence controlled by *picco*-processor critically.

ラムステップ数.
 n : スペースを含めた 1 文字の横のドット数.

T_{Acc} : シーケンサのマシンサイクル (ns).
 T_{vc} : ビデオクロックの周期 (ns).

である. 図 8 のコントロールフローの中で, この制御条件が満足されなければならない部分を破線の矢印で示した. 一文字の横を 9 ドットとし, $T_{vc} > 100$ ns とすれば, 図 11 から T_{Acc} を 300 ns にしても良いことがわかる. したがって, 試作した p - p の T_{Acc} と T_{vc} をそれぞれ 300 ns, 150 ns にし, 命令語長は 12 ビット構成 (図 3 参照) にした.

試作した p - p の規模は SSI, MSI の数にメモリ・チップ数を単純に加えると 70 チップとなった. 同様にして得られたディスプレイの制御部のチップ数は同期信号発生部とメモリ (キャラクタバッファとキャラ

クタパターン ROM) を除いて 28 チップであった。

5. 考 察

前章で述べたように、*p-p* は順序回路の設計を単純なプログラム論理により容易に実現できる素子であった。本章では、*p-p* の特徴をより明確にするため、BS 形シーケンサと *p-p* との比較、考察を行う。

BS 形シーケンサを用いて *p-p* と同等な機能をもつ回路を構成するためには、図 12 に示すように、BS 形シーケンサの他にマイクロ・プログラム・メモリ、ネクスト・アドレス制御部、条件判定部が必要となる。すなわち、BS 形シーケンサはマイクロ・プログラム・メモリのアドレスを決定するロジックのみを有する構造になっている。これはどのような制御構造にも適応できる柔軟な構造を有しているといえるが、その反面いつも周辺回路といっしょに使わねば、シーケンサとしての働きができないということも意味するものである。したがって、対象とする回路がある程度複雑である場合には、BS 形シーケンサを用いてコスト/パフォーマンスをあげることができる。現状では、小形で高速な回路の設計はワイヤードロジックに依るしかない。なぜならば、BS 形シーケンサを用いてこの回路を設計したとき、これにより制御される被制御回路のハードウェア量 (コスト) が BS 形シーケンサを含めた周辺回路に比べてあまり大きくないからである。

ところで、これを解決するために考えられたものが *p-p* であると前にも述べてあるが、次にその長・短所をあげてみよう。

長所としては、

- a. プログラミングが非常に簡単に行えること (四つの基本命令しかない)。
- b. 構造が簡単であること (基本的に条件入力端と出力端しかない)。

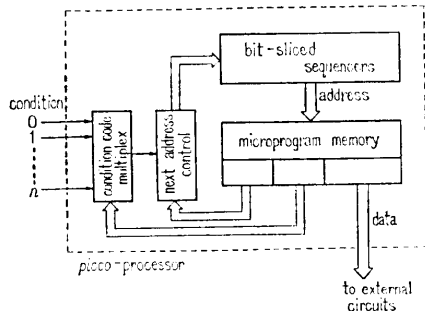


図 12 ピコ・プロセッサとビットスライス形シーケンサとの比較

Fig. 12 Picco-processor vs. bit-sliced sequencers.

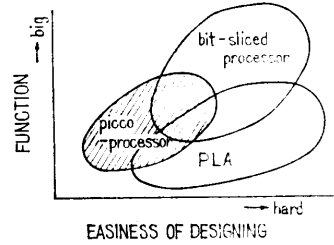


図 13 ワイヤードロジックをプログラマブルロジックに置き換える場合、その設計の対象となる回路機能の大きさと設計し易さとの関係の概念図

Fig. 13 Conceptual relations with function and easiness of designing when programmable logic elements are used to design a certain hardwired logic.

- c. 1 チップにシーケンサに必要な全機能を集約したこと (図 12 の破線部分)。

などがあり、小形で高速な回路でも *p-p* を用いれば、コスト・パフォーマンスを落さずに、十分ストアロジックによる設計が可能となる。

短所になる要因は、ほとんど長所の c に起因するものであり、例えば

- a. チップのゲート・ピン比によりアーキテクチャに制約が生じること
 - b. 高集積化と高速な処理速度が要求されるので、素子密度とゲート遅延との積を減らす高速デバイス技術により、LSI 化されなければならないこと
- などがある。

図 13 に、設計の対象となる回路の機能の大きさと設計し易さとの関係を PLA, BS 形シーケンサ, *p-p* について図示した。

6. む す び

本稿では、今までワイヤードロジックで組むのが一般的であった小形で高速な論理回路を、コスト/パフォーマンスを下げずに、プログラマブルロジックにより設計できる汎用の順序回路素子について論じた。

今後、*p-p* を LSI 化するとき起る諸問題——高速デバイス技術の選定、チップのピン数制限を考慮した適切な命令語長の決定、メモリの書き込み法の決定、インタラプト機能の追加方法等——についてさらに検討する必要がある。

謝辞 日ごろ、ご指導、ご鞭達いただいている平山博教授に深甚な謝意を表します。また、本研究を進めるに当りきっかけをいただいた東光(株)の山中稔博士に深謝します。

なお、本研究は相川健(現東芝)君の修士論文から

手がけられたものである。

ご親切に、根気よく不備な点をご指摘、ご訂正して下さった未知の査読者に深謝します。

参 考 文 献

- 1) Torode, J. Q., Kehl, T. H. : The Logic Machine, IEEE Trans. Comput., Vol. C-23, No. 11, pp. 1164-1169 (Nov. 1974).
- 2) Clark, W. A. : Macromodular Computer System, in proc. 1967 SJCC, pp. 337-401.
- 3) Bell, C. G., et al. : The Description and Use of Register Transfer Modules (RTM's), IEEE Trans. Comput., Vol. C-21, No. 5, pp. 495-500 (May 1972).
- 4) Parker, A. C., Siewiorek, D. P. : Educational and Industrial Applications of Register Transfer Modules, 2nd EUROMICRO, pp. 221-230 (Sept. 1976).
- 5) Lipovski, G. J. : The Architecture of a Simple, Effective Control Processor, *ibid.*, pp. 7-18 (Sept. 1976).
- 6) Friedman, A. D., Simoncini, L. : The Effect of LSI Technology on the Theory of Modular Computer Design, *Computer*, pp. 60-67 (July 1978).
- 7) Logue, J. C., et al. : Hardware Implementation of a Small System in Programmable Logic Arrays, IBM J. Res. Dev., Vol. 19, No. 2, pp. 110-119 (March 1975).
- 8) Lipp, H. M. : Array Logic, 2nd EUROMICRO, pp. 57-64 (Sept. 1976).
- 9) Horninger, K. : A High-Speed ESFI SOS Programmable Logic Array with an MNOS Version, IEEE J. Solid-state Circuits, Vol. SC-10, No. 5, pp. 331-336 (Oct. 1975).
- 10) Wood, R. A. : High-Speed Dynamic Programmable Logic Array Chip, IBM J. Res. Dev., Vol. 19, No. 4, pp. 379-383 (July 1975).
- 11) Boulaye, G. G. : Microprogramming, pp. 34-110, John Wiley & Sons, New York (1975).
- 12) Husson, S. S. : Microprogramming Principles and Practices, pp. 22-76, Prentice-Hall, New York (1970).
- 13) Gray, M. T. : Microprocessors in CRT Terminals Applications, *Computer*, pp. 53-59 (Oct. 1975).
- 15) 富永, 小原, 平山: 改版電子計算機, pp. 217-226, コロナ社, 東京 (1977).
- 16) 姜, 富永, 平山: $\mu\mu$ -CPU (ピコ・プロセッサ) の提案とその応用, 信学会計算機研資 EC 76-65, pp. 25-34 (1976).

(昭和53年8月25日受付)

(昭和54年4月19日採録)