

ROM による論理回路設計法

不破 泰* 中村八束**

(昭和61年5月26日受理)

A Method of Designing Logic Circuits Using ROM

Yasushi FUWA and Yatsuka NAKAMURA

In view of the fact that the conventional method of designing logic circuits which uses many gate ICs raises a number of problems, a new method has been developed which uses ROMs and its application and evaluation are described. The problems include increase in the price and space of ICs, complexity of circuits, difficulty in modifying circuits, large inventory of ICs, and occurrence of hazards. The new method consists of allocation of inputs and outputs of a truth table to address and data pins of a ROM, respectively, and of writing of the truth table into the ROM; addition of latch circuits to the ROM may form synchronous sequential circuits. The application of the new method is exemplified by a logic circuit and a synchronous sequential circuit, with evaluations commented. A comparison from various points of view is made between the new and conventional methods.

1. 緒 言

論理回路は通常ゲート IC を組み合わせて設計する。しかし複雑な論理回路において多くのゲート IC を使った場合、IC のコストやスペースが増大すること、回路や配線が複雑になること、仕様の変更による設計変更が容易に行えないこと、保守用に用意しておかなければならない IC が多種になること、各 IC の伝搬遅延時間にバラツキがありひげ等があらわれやすいこと等の問題が生じてくる。カスタム LSI の利用は、費用、開発期間等問題が多い。近年 PAL¹⁾(Programmable Array Logic)等ユーザーの手元でプログラム可能なゲート IC が普及してきて、変更の容易さ、配線の単純さの面では大きな進歩が見られている。しかしコストの面ではこれからであるし、又実現できる回路の複雑さにも制限があり、回路設計は必ずしも容易になったとはいえない。ここでは ROM が最近低価格化かつ大容量化したことに着目し、ROM を中心とした組み合わせ論理回路と、更に

* 情報工学教室助手

** 情報工学教室教授

同期式順序回路の設計を提案し、その具体例を示し、従来の方法との比較を行う。

2. ROM による組み合わせ論理回路

組み合わせ論理回路の設計は真理値表やカルノー図から出発することが多いが、ROM を用いる場合は真理値表の作成のみで回路設計が終了する。というのは、アドレスバス線の組み合わせが回路の入力であり、ROM の記憶された情報の内容が回路の出力であるからである。その概念ブロック図は図1のようになる。64Kbit の ROM であれば13本のアドレスバス線を持ち、8本のデータバス線を持っているので、13入力の8出力以下のどんな複雑な論理回路にも対応できる。

例えば図2の真理値表について考えてみると、8Kbit の ROM を使いアドレスバス線 A3 を入力Aに、A2 を入力Bに、A1 を入力Cに、A0 を入力Dに割り当てる。またデータバス線D1 を出力Eに、D0 を出力Fに割り当てる。そして ROM にこの真理値表に従い図3の様なデータを書き込む。あとは基板にこの ROM を実装し、利用しないアドレスバス線(A4~A9)をグランドに接続すれば論理回路が完成する。この場合アドレスバス線10本の内4本しか利用しておらずデータバス8本の内2本しか利用していないため、ROM のメモリーセル全体の0.4%ほどしか使わないことになる。

この論理回路は8ケ位の TTL で実現できる。従ってその価格は EPROM 1ケの価格とほぼ同程度である。図2の真理値表はこの ROM にとっては最も単純な部類に属し、これより複雑になればなるほど、ROM を用いた場合の優位性が増してくる。

論理回路の遅延時間は1ゲートあたり TTL の場合約 30nsec として、3段あるいは4段に組むと90~120nsec になる。これに対し PROM は 150~250nsec 程であるから、2倍程度の遅延時間を要することになる。高速なバイポーラの PROM を用いると遅延は 25~60nsec 程になるが価格は約2倍になる。

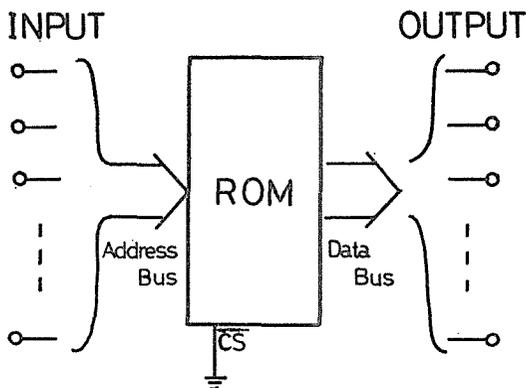


Fig. 1 Block diagram of a logic circuit using a ROM.

A	B	C	D	E	F
0	0	0	0	1	0
0	0	0	1	0	1
0	0	1	0	1	1
0	0	1	1	1	0
0	1	0	0	0	1
0	1	0	1	0	0
0	1	1	0	1	1
0	1	1	1	1	0
1	0	0	0	0	0
1	0	0	1	1	0
1	0	1	0	1	0
1	0	1	1	1	0
1	1	0	0	0	0
1	1	0	1	1	1
1	1	1	0	0	0
1	1	1	1	0	0

Fig. 2 An example of a truth table.

A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	D7	D6	D5	D4	D3	D2	D1	D0	
0	0	0	0	0	0	0	0	0	0	x	x	x	x	x	x	1	0	
0	0	0	0	0	0	0	0	0	0	1	x	x	x	x	x	x	0	1
0	0	0	0	0	0	0	0	0	1	0	x	x	x	x	x	x	1	1
0	0	0	0	0	0	0	0	0	1	1	x	x	x	x	x	x	1	0
0	0	0	0	0	0	0	0	1	0	0	x	x	x	x	x	x	0	1
0	0	0	0	0	0	0	0	1	0	1	x	x	x	x	x	x	0	0
0	0	0	0	0	0	0	0	1	1	0	x	x	x	x	x	x	1	1
0	0	0	0	0	0	0	0	1	1	1	x	x	x	x	x	x	1	0
0	0	0	0	0	0	0	1	0	0	0	x	x	x	x	x	x	0	0
0	0	0	0	0	0	0	1	0	0	1	x	x	x	x	x	x	1	0
0	0	0	0	0	0	0	1	0	1	0	x	x	x	x	x	x	1	0
0	0	0	0	0	0	0	1	0	1	1	x	x	x	x	x	x	1	0
0	0	0	0	0	0	0	1	1	0	0	x	x	x	x	x	x	0	0
0	0	0	0	0	0	0	1	1	0	1	x	x	x	x	x	x	1	1
0	0	0	0	0	0	0	1	1	1	0	x	x	x	x	x	x	0	0
0	0	0	0	0	0	1	0	0	0	0	x	x	x	x	x	x	x	x
0	0	0	0	0	1	0	0	0	0	1	x	x	x	x	x	x	x	x
1	1	1	1	1	1	1	1	1	1		x	x	x	x	x	x	x	x

x:Don't Care

Fig. 3 Memory map of the ROM for Fig. 2.

3. 組み合わせ論理回路の例

2章で述べた ROM を用いた組み合わせ論理回路の例として10進数 (BCDコード) の加減算器を考える。1個のROM で1桁の演算を行うとすれば、10進数の各桁は4bit で表

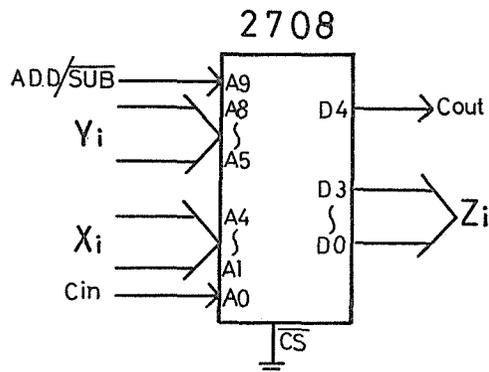


Fig. 4 Block diagram of a BCD adder-subtractor using a ROM.

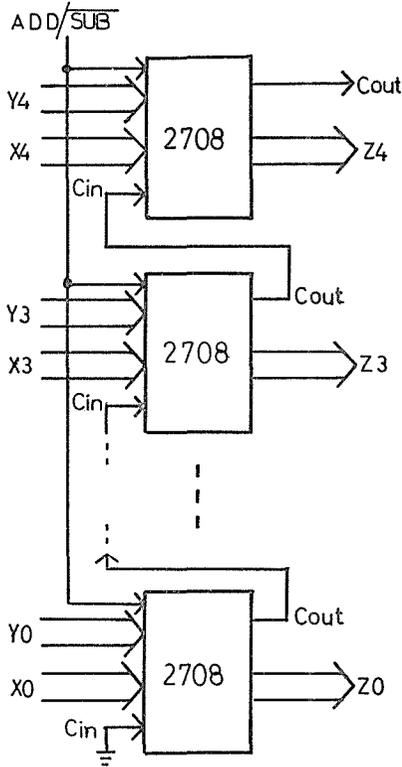


Fig. 5 Block diagram of a 5-digit BCD adder-subtractor.

される為各桁毎に入力される被演算数、演算数で4 bit づつ、下位の桁から繰り上がりまたは繰り下がりにより1 bit、他に加算を行うか減算を行うかの切り換えに1 bit 必要であるから、ROM のアドレスバス線は合計10本必要となり、8 Kbit の EPROM が利用できることになる。この EPROM のブロック図を図4に、これを用いて作成した5桁の10進加減算器のブロック図を図5に示す。

この EPROM の遅延時間は 250nsec であるので、5桁の加減算に要する時間は $1.25\mu\text{sec}$ となる。汎用の中型計算機において5桁の10進加減算がおおよそ $3\mu\text{sec} \sim 20\mu\text{sec}$ 程度であるので EPROM を用いることにより簡単な回路で高速の演算器を容易に短期間で作成することが出来ることになる。又、加減算以外でも \sin , \cos 関数等の演算器も同様にして簡単に作成することができる。

4. ROM による同期式順序回路

ここでは、ROM を用いた同期式順序

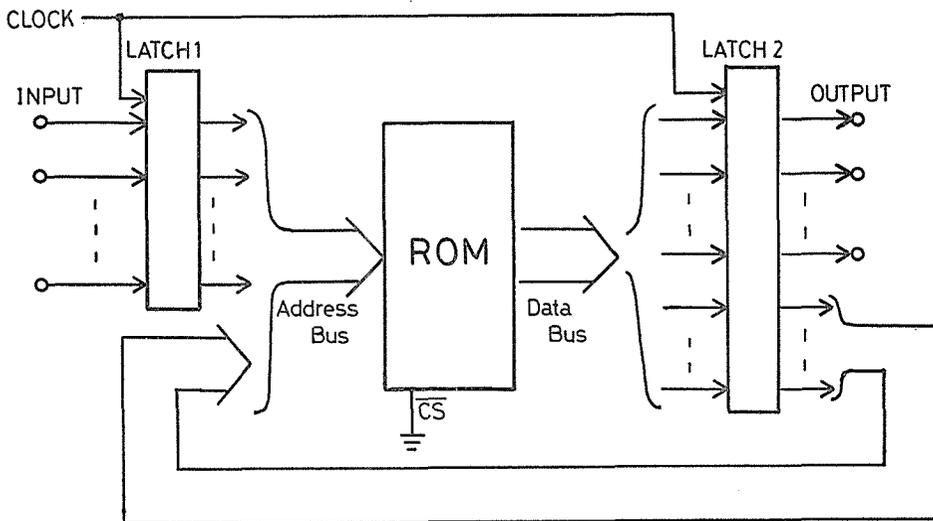


Fig. 6 Block diagram of a synchronous sequential circuit using a ROM.

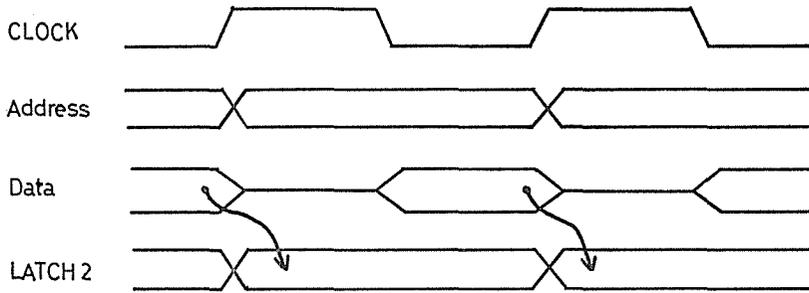


Fig. 7 Timing diagram for Fig. 6.

回路の可能性について述べる。

図 6 に ROM を用いた同期式順序回路のブロック図を示す。ラッチ 1 は入力信号を保持するためのものでラッチ 2 は順序回路の状態値と出力を保持するためのものである。このクロックには ROM の読み出し遅延時間より長い周期のパルスを加える。この回路の動作を図 7 のタイミングチャートで説明する。クロックの立ち上がりで入力信号はラッチ 1 に保持され、ROM のアドレスバス端子に加えられる。又、現在の順序回路の状態値はラッチ 2 に保持されアドレスバス端子に加えられる。そして ROM の読み出し遅延時間後に ROM より次の状態と出力信号が出され、それをラッチ 2 が次のクロックの立ち上がりで保持し、状態値は ROM にフィードバックする。ラッチ 1 は ROM の読み出しの動作中にアドレス信号が変化するのを防いでいる。

入力信号の数を n とし、状態値を表すのに必要なビット数を m とすると、ROM のアド

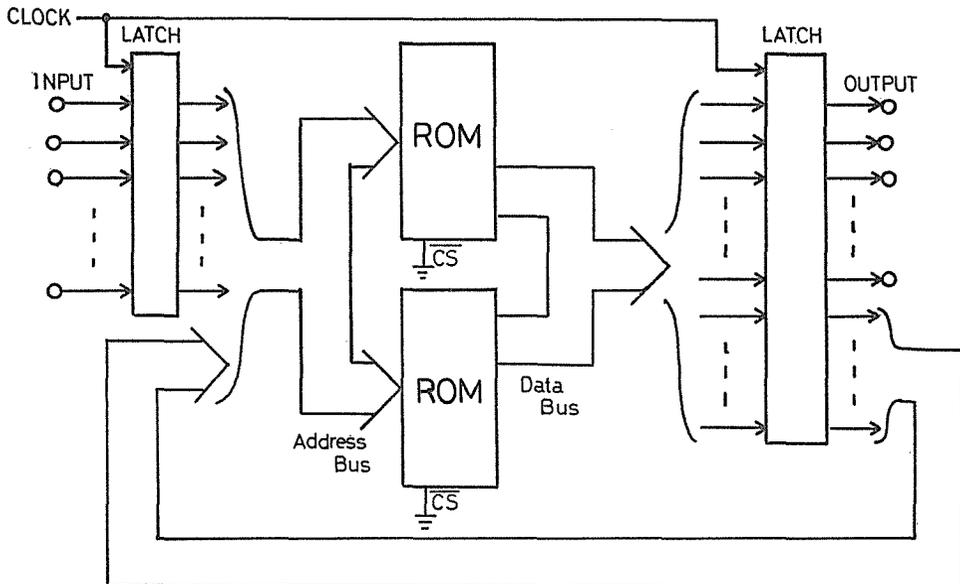


Fig. 8 A synchronous sequential circuit with many outputs.

レスバス線の本数は $n+m$ 以上でなくてはならない。又、出力線の数を k とすると、ROM のデータバス線の本数は $k+m$ 以上でなくてはならない。アドレスバス線の本数は ROM の容量によって決り、 $n+m$ の値が増えれば更に容量の大きな ROM を使うことになる。現在の PROM ではこの本数が16本のものまで市販されている。データバス線の本数は現在の大半の ROM が4本か8本である。 $k+m$ の値がこれ以上であれば、同じ ROM を複数並列に並べればよい(図8参照)。

ROM を用いた同期式順序回路の設計は、通常の順序回路の設計²⁾と同様に、まず状態遷移図を作成する(例として図9の状態遷移図を作成したとする)。そして状態割り当てを行い、真理値表を作成する(図10)。通常の順序回路の設計ではこのあとこの真理値表からカルノー図等を書き、状態値 $S1^+$, $S2^+$, $S3^+$ の論理式と出力 O の論理式を求めて回路を作成するが、ROM を用いる場合は、この真理値表の $A, B, S1, S2, S3$ をアドレスバス信号の $A4 \sim A0$ にそれぞれ割り当て、又 $S1^+, S2^+, S3^+, O$ をデータバス線の $D3 \sim D0$ にそれぞれ割り当て、組み合わせ論理回路と同様にこの真理値表をそのままデータとする ROM を作成する。そして、ラッチ IC を付けて図11の回路を作成すれば、順序回路は完成する。なおこの図において状態を初期化するためにアドレスバス線の $A5$ を用いているが($A5 = 1$ のときの ROM のデータを全て0にしている)、ラッチにクリア端子があればこれを使ってもよい。

この回路を TTL IC で作成すると、IC の数は20個程度必要になる。そのため価格的には EPROM を用いたほうが安価となる。また遅延時間の短いバイポーラ ROM を用いても安価となる。動作速度は回路に加えるクロックによって決まるが、EPROM を使った場合 EPROM の遅延時間を 250nsec とするとクロックの周波数は 4 MHz, また遅延時間

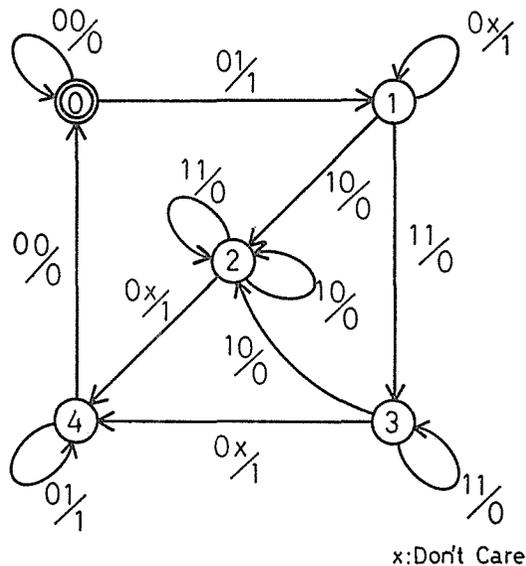


Fig. 9 An example of a state transition diagram.

	S1	S2	S3
①	0	0	0
②	0	0	1
③	0	1	0
④	0	1	1
⑤	1	0	0

A	B	S1	S2	S3	S1 ⁺	S2 ⁺	S3 ⁺	O
0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	1	1
0	1	0	0	1	0	0	1	1
1	0	0	0	1	0	1	0	0
1	1	0	0	1	0	1	1	0
1	1	0	1	0	0	1	0	0
1	0	0	1	0	0	1	0	0
0	0	0	1	0	1	0	0	1
0	1	0	1	0	1	0	0	1
1	1	0	1	1	0	1	1	0
1	0	0	1	1	0	1	0	0
0	0	0	1	1	1	0	0	1
0	1	0	1	1	1	0	0	1
0	1	1	0	0	1	0	0	1
0	0	1	0	0	0	0	0	0

(a)
(b)

Fig. 10 State assignment (a) and the truth table (b) for Fig. 9.

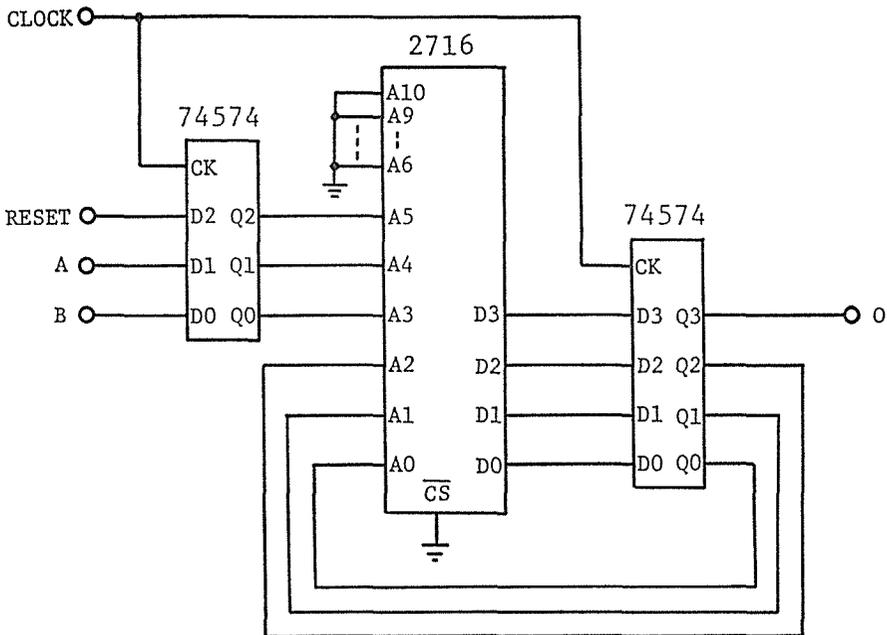


Fig. 11 A synchronous sequential circuit for Fig. 9.

35nsec のバイポーラ ROM を用いると 28.5MHz まで動作可能である。一方 TTL IC を用いると 3 段程度で組めるので、11MHz 程度までの動作となる。

順序回路にはこの同期式以外に非同期式順序回路があるが、ROM は入力アドレスバス線が 1 本でも変化すると遅延時間の間出力のデータバス線が全て不安定になるので、非同期式順序回路に ROM を用いた本方法をそのまま適用する事はできない。

5. ROMによる同期式順序回路の応用例

前章で提案した ROM による同期式順序回路の例として、非同期式 *RS-232C インターフェイス回路を設計した。図12に設計した回路のブロック図を示す。ROM を用いた回路はこの図の出力制御回路、入力制御回路の部分である。出力制御回路は CPU により

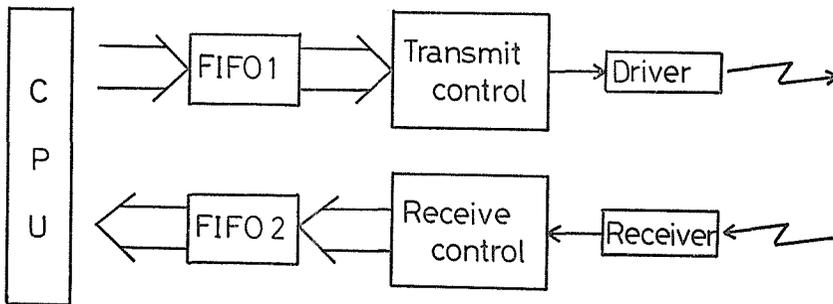


Fig. 12 Block diagram of the RS-232C interface.

FIFO 1 に書き込まれた出力データを受け取り、指定されたデータ長、ストップビット長、の形式で 1 ビットずつドライバ回路を通して出力する。その際パリティ計算も行い、指定されたパリティ形式で出力する。入力制御回路は逆にレシーバ回路を通して入力されたシリアルデータを指定された形式に従いパラレルデータに組み立て、FIFO 2 に書き込む。ここで FIFO 1 と 2 は IDT 社の IDT7201³⁾ (512×9 ビット) を用いた。CPU はこの FIFO を介してデータの入出力を行うため、1 文字ずつ送受信する従来の RS-232C 用インターフェイス IC を用いる場合に比べて、入出力処理の負荷が軽減される。なおそれほど処理の負荷が問題にならないときは、高価な FIFO のかわりにレジスタ IC を用いて、従来の IC 同様 1 文字ずつデータのやりとりをすることもできる。

出力制御回路、入力制御回路の設計は前章で述べたとおり状態遷移図を書き、状態割り当てを行い、真理値表を作成しこれをデータとして ROM に書き込むことにより行う。図 13 にその一部として出力制御回路の状態遷移図を示す。また図 14 に出力制御回路、図 15 に入力制御回路の回路図を示す。どちらの回路も出力線の本数が多いため 2 個の ROM を並列に並べて使用した。この回路の伝送速度は、遅延時間が 200nsec の EPROM を用いた場合、出力制御回路は 5 Mbps で動作可能であり市販されているインターフェイス LSI (インテル 8251 は入出力とも非同期式では 19.2Kbps まで) に比べ 200 倍以上高速である。

*これは通信方式としての非同期式の意味である。

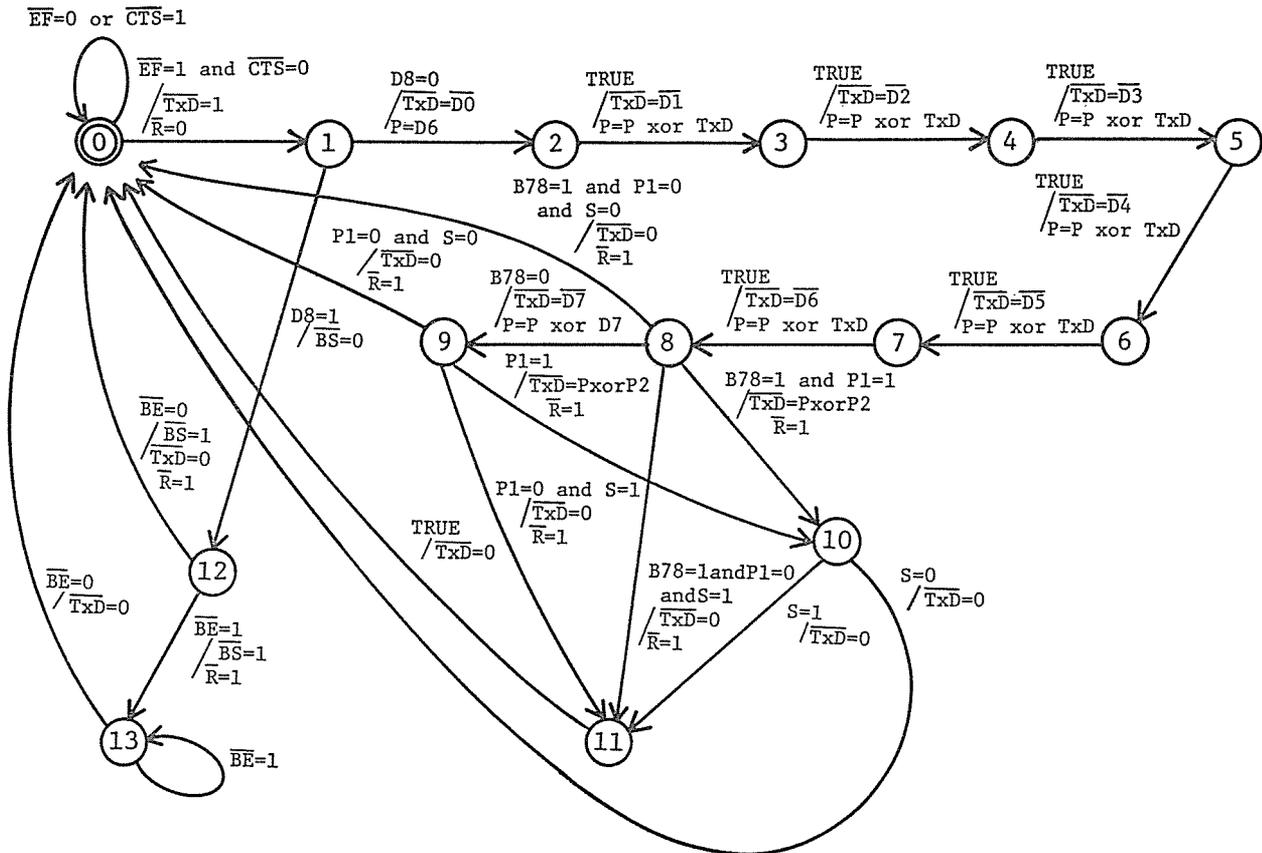


Fig. 13 A state transition diagram for transmit control.

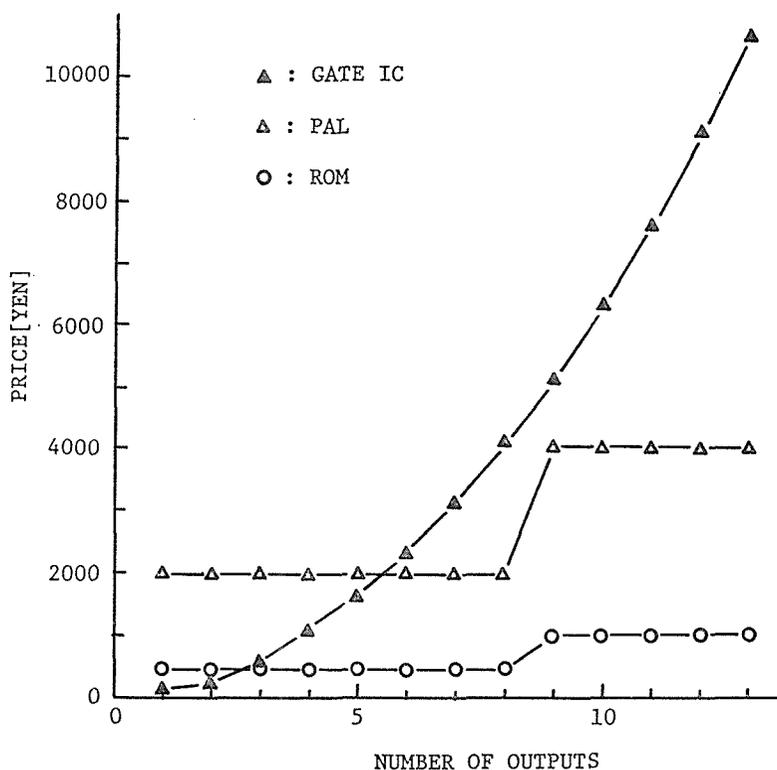


Fig. 16 Relation between the price and number of outputs.

かの目安を示す。従来の方法として通常のゲート IC を用いる方法の他に、PAL を用いる方法も取り上げ比較を行った。

図16に各方法を用いたときの出力の数と価格との関係を示す。ここで入力の数も出力の数と等しいと仮定し、また各出力の論理式中の OR の数を平均5と仮定している。ROM を用いる方法と PAL を用いる方法とでは、この場合どちらも出力の数が8以下のとき1個の ROM 又は PAL で実現できるが、出力が8以上では2個必要になる。これに対しゲート IC を用いる方法では、1個の IC は安価であるが IC 中の AND, OR ゲートが少ないため、出力が増えるにつれ使用する IC の数が多くなる。このグラフから出力の数が3以上の時は、ROM を用いる方法が最も安い事がわかる。(なおここで用いた価格は86年5月現在、一般市場で少数ロットにより購入した場合のおおよそのものである)

次に速度の比較として、図17に各方法を用いたときの論理式の複雑さと遅延時間との関係を示す。ここでは論理式の複雑さとして論理式(積和形)に含まれる OR の数を用い、また入力数は6としている。ROM を用いる方法では、遅延時間は ROM のアクセスタイムであり常に 200nsec である(最近ではアクセスタイムが 150nsec の物もでてきており、またバイポーラの ROM を用いると 100nsec 以下となる)。PAL を用いる方法では、1個の PAL の遅延時間は 40nsec と高速であるが、1個の PAL で実現できる OR の数

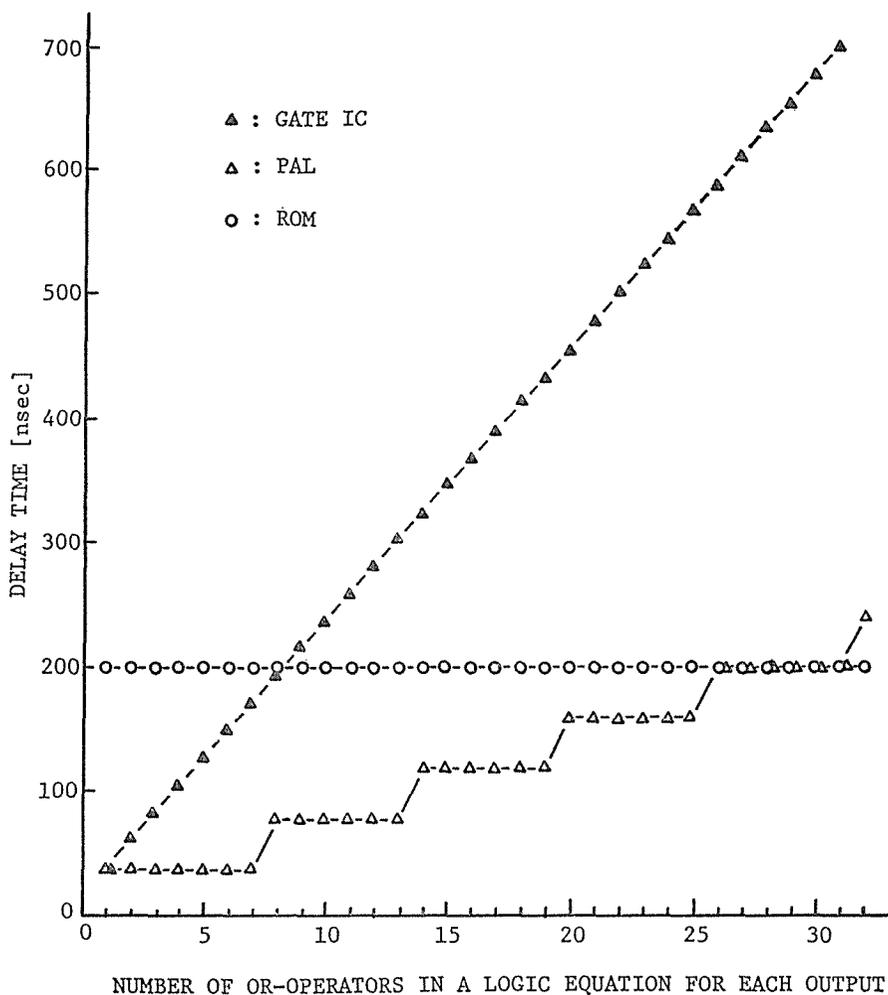


Fig. 17 Relation between the delay time and number of or-operators in a logic equation for each output.

に制限があり（多い物でも 8 個程度）OR の数が増えるにつれ PAL を何段か組み合わせなければならず速度は低下する。ゲート IC を用いた方法でも OR の数が少ないときは高速であるが、OR が増えるに従い何段もの IC を組み合わせる事になり速度は低下する。このグラフからは ROM を用いる方法は OR の数が 9 個以上のときはゲート IC より高速となり、PAL を用いる方法は高速であるが OR の数が増えるにつれ ROM とあまりかわらなくなることがわかる。

この 2 つのグラフより、複雑な論理式の場合とあまり高速性が問われない場合は、ROM を用いる方法が有望だと言える。

7. 結 語

論理回路の設計における様々な問題点を解決するため、ROM を用いた回路設計を提案し、従来の TTL IC を用いた場合と比較し、設計の容易さ、変更のしやすさなどがすぐれ、また価格、速度についても有利であることを示した。また、ラッチ回路 IC と組み合わせ、同期式順序回路も ROM を用いて設計できる事を示し、論理回路の場合と同様の利点がある事を確かめた。とくに、順序回路については、その例として RS-232C インターフェース回路を設計し、CPU にとり処理の負担が少なく、従来のインターフェース IC と比べ高速動作が可能な回路が、本方法により簡単に設計できる事を確かめた。非同期式順序回路に本方法をそのまま用いる事は出来ないという欠点もあるが、今後 ROM が更に大容量化、高速化、低価格化するのに伴い、本方法はますます有効な手法となると思われる。

参 考 文 献

- 1) Monolithic Memories, Inc.: PAL Handbook, (1984).
- 2) J. HARTMANIS and R. E. STEARNS: Algebraic Structure Theory of Sequential Machines, Prentice-Hall, Inc. Englewood Cliffs, N. J. (1966).
- 3) Integrated Device Technology, Inc.: HIGH SPEED CMOS PRODUCTS, pp. 65-75 (1985).