

2470 → 2485A → 4295C
2446
2486A → B
九月()
2508 → 4320 → 2509 → 4322
P.H.C.F.

8桁 & XEリ卓費用 LSI
μPθ940C 設計報告

IEL - 3745
昭和 49 年 10 月 11 日
集積回路(株) 2 回路技術部
山口

資源不足が騒がれ、種々の物価が急上昇している昨今、異常にも卓費用 MOS LSI は 円切り下げ、さらには 本年中と予想される製品輸入自由化による 外国製品攻勢 に対処せねばならぬところから、大幅な価格引き下げが行なわれている。このため NEC においても 本年 4 月中旬より 高級に横ばいあるが、価格が 1000 円を割ってまで 生産販売可能な 超高性能卓費用 LSI の設計を開始した。μPθ940C がそれであり、7 月末、お1回試作が完了し、その機能確認が為されたので、設計内容について報告する。

[成果概略]

- PL 板サイズ 3.78 x 3.80 mm
- Pin 個数 2ピン 35~40 個, 3ピン 100~120 個
- 消費電力 標準使用時 25mW

[設計担当者]

- 2回路技術部
システム回路、レイアウト設計 ... 山口
モジュール、70-pin による 錫めっき ...
ROM 検出用 フォトマスク作成、原図チェック ...
- 製品設計関係
No. 15 実用性の評価 ...
ファンクションブロック設計 ...

設計基準及び仕様

① デバイス

- Pチャンネル・マルチゲート 型 MOS (1チャンネル入1回)
- No.15 設計ツール

従来マルチゲート型構造では、ホロン肉厚10μを基準としたNo.4、No.5 ツールの使用していたが、ホロン拡散プロセス変更により可能となったホロン肉厚7.5μを基準としたNo.15設計ツールを使用している。(TEL-3270, 3581 製設. 参照)

この巻、従来の設計ツールを使用した場合に比し、自動コア面積が12.5~15%程度小さくなる事がある。

UPD940CはこのNo.15ツールを採用した最初の量産品種であるので、充分なコア設計検討及び評価を行なう必要がある。

- 内部コア・レジスタ、表示素子(蛍光表示管と主対照と可変)ノリ付、内部自動コア論理回路、キ入力アルファ回路を含んだ完全1チップ。28ピン・プラスチック・モールド

② 卓電仕様

- 総動小数点 8桁 × 20
- 上位8桁優先表示 (標準記憶(10³), エラー、型番表示用(桁を落つ))
- 8桁7.5桁算、四則、微算、パーセント計算、ルート計算、π置数
- その他、四則定数計算、四則くり返し計算、割引割増計算等、標準的仕様については全て満足している。
- 消費電力標準値 25mW
- 算数式 ← 加算機式、表示方式 (8桁表示 ← 8桁表示、型番表示、エラー表示等)、卓電仕様 年の変更に対しては、ROM部メモリ256の仕替え要する所がある。25700・70000α方式を添えている。
- その他細かい仕様については、本報告の裏面等とご参照願います。

[2] システム 論理 回路設計

卓電のように小規模なシステムになると システム 論理 回路設計 向の
 キットは 殆んど なくなり 1つの ひとまた 設計 分野 がある と言って 良し。
 さらに キット 準備 を 縮小 しよう と する ならば レイアウト 設計 との 対応 パック
 は 緊密 な もの を 無くて は なる はず。 回路 設計 終了 の 際 に レイアウト に
 対応 する 十分 の 配線 が 成され ると 同等 に レイアウト 設計 終了 において は
 レイアウト し たい 様 に 回路 設計 の 変更 を 行 える 必要 が ある。 す、LSI
 検査 用 テスト パターン と ランダム ビット パターン を 作る 場合には パターン 数 が
 多 く し かも 故障 検出 率 の 高い もの と する 意 義 がある テスト 装置 の 設定、
 テスト 方法 も 考慮 する 事 が 必要 とな っ て くる。

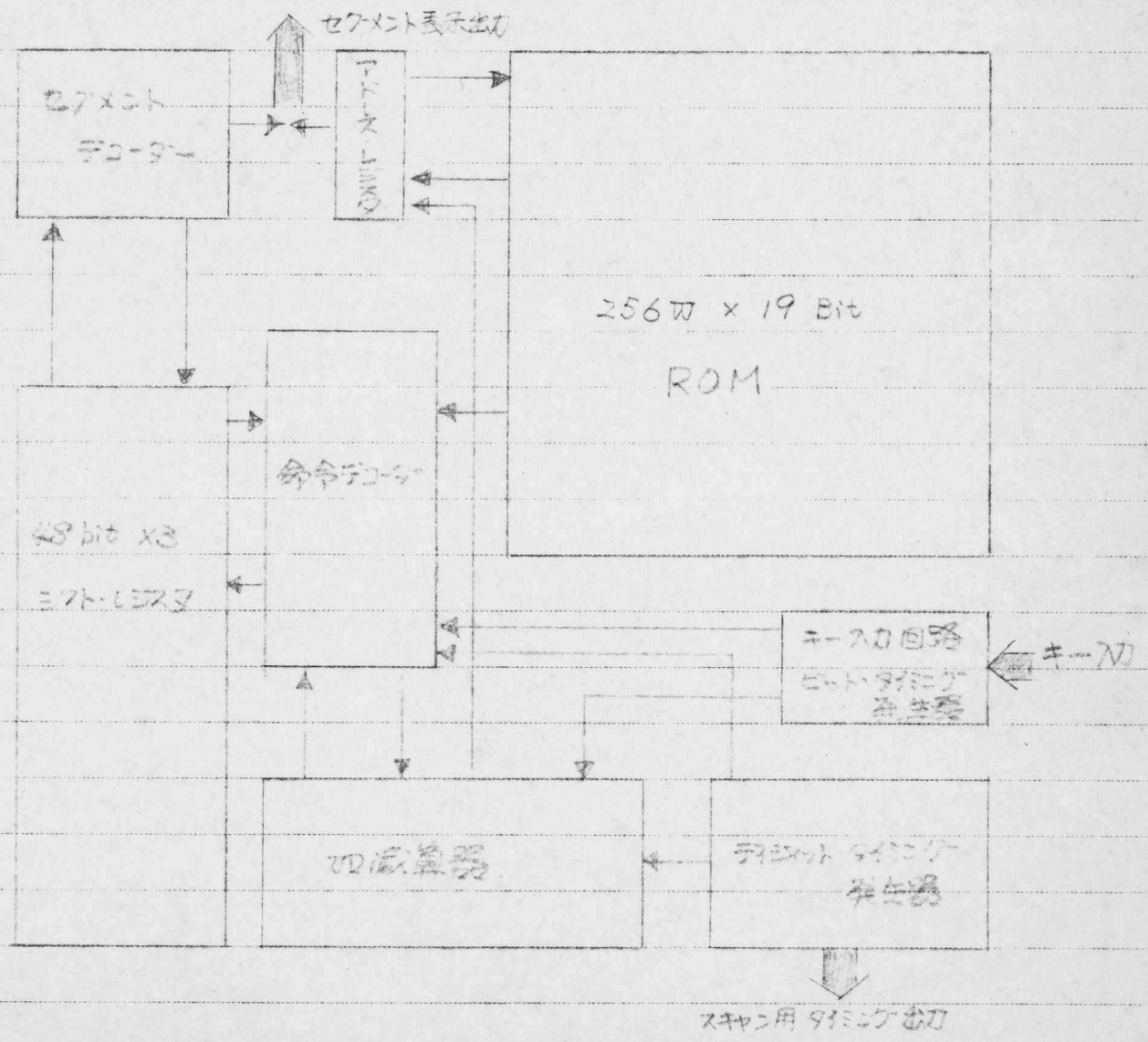


図1. システム 構成

ROM部を除いた回路図。MPD940C, 941C, 942C のフロッピー、命令コード - 番号を意味に付ける。

① レジスタ構成

X, Y, Z 各々 48 ビットのソフトレジスタ 3本からなり、XReg... 表示、演算
YReg... 定数格納、ZReg... DP表示、演算 の概略構成を記す。

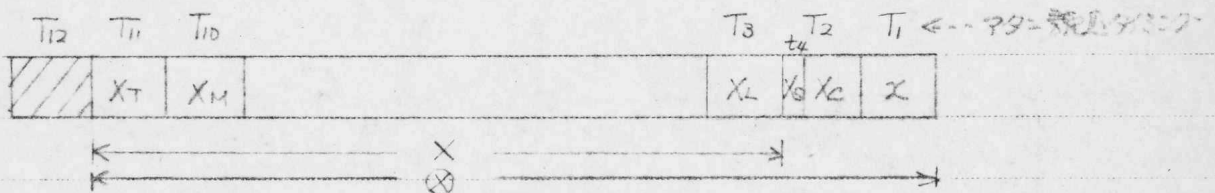


図2. レジスタ内容 RW, 7ビット・ニーモック

X	小数乗数データ 4ビット	X _M	演算結果8桁目データ 4ビット
X _c	演算制御、定数用1桁目データ	X _T	演算結果9桁目データ 4ビット
X _s	正負符号データ 1ビット	X	演算結果全桁データ 36ビット
X _L	演算数値1桁目データ 4ビット	⊗	レジスタ全データ 48ビット

従来、NEC 標準品種において用いられていた、キー信号エンコーダ、
マイクロ制御用カウンタ、ホールド回路、演算制御エンティティ等
等のハードウェアは、全て、上記 X_c 3ビットのソフト的処理において可能
となり、ゲート総数が激減した。

T₂ のレジスタ内容は使用せず常に '0' にクリアされている。その理由は
T₂ データを 4ビットの値を有ったシリアルデータとして取り扱うと、その処理
結果を T₁ データに出力され、そのときには既に ROM のアドレスが更新
されており、結果をレジスタ内に読み込む事が不能になる場合が生ずる
ためである。これを避ける為には、レジスタへの読み込みゲートに供給する
ROM 出力を、4ビット同時に遅らせる事は無いのであるが、5桁目、6桁目の
増設及びキー入出力方式との相違もあり、行なわれている。

② ROM

256 アドレス 19 ビット メモリ規模 4864 ビット 3本並

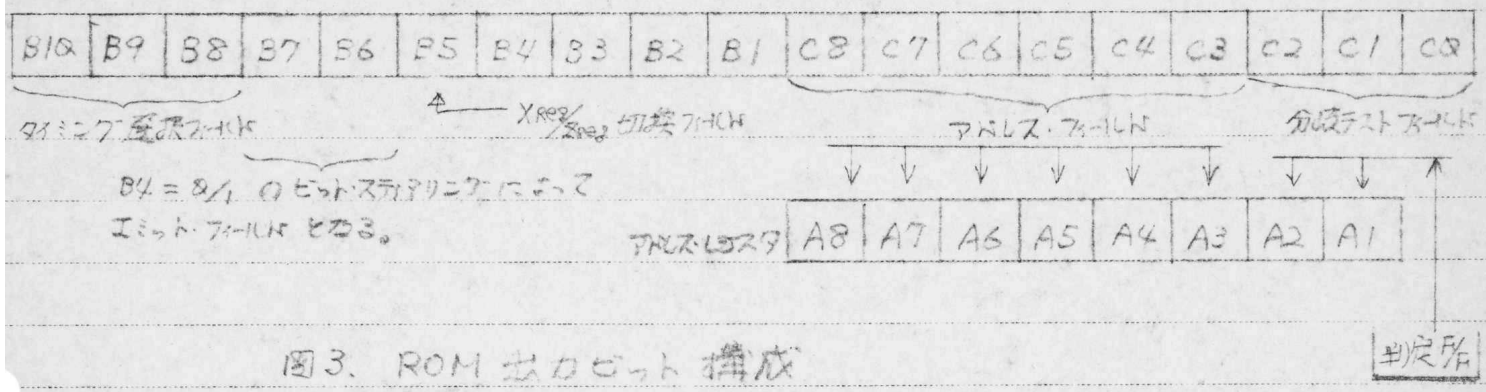


図3. ROM 出力ビット構成

(a) ROM アドレス選択法

uPD940C で行なっている機能の直接指定 と エントロル ROM の規模の大ききもの に使用される ステア・カウンタ (シーケンス・コントロール カウンタ, あるいは アドレス・カウンタ) による指定 の二種に大別される。

I. 直接指定

(i) 完全直接指定

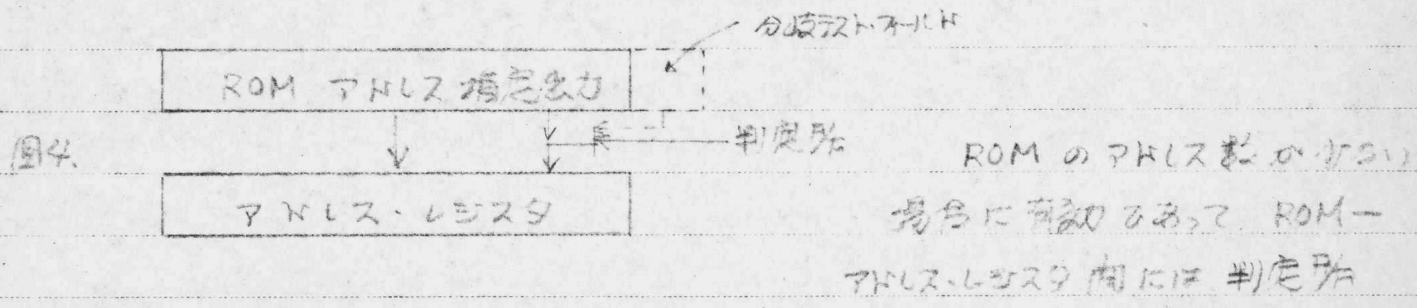


図4.

ROM のアドレス数にかかわらず ROM-アドレス・レジスタ内には 判断信号内容を制御するゲートが存在するだけならラッチゲート数は少ないが アドレス指定出力は アドレス・レジスタのビット数全てを必要とし, ROM 出力ビット数が増える。アドレス割当に対する制限は全くなく, ROM を 100% 無駄なく使用する事ができる。

(ii) バージ内直接指定

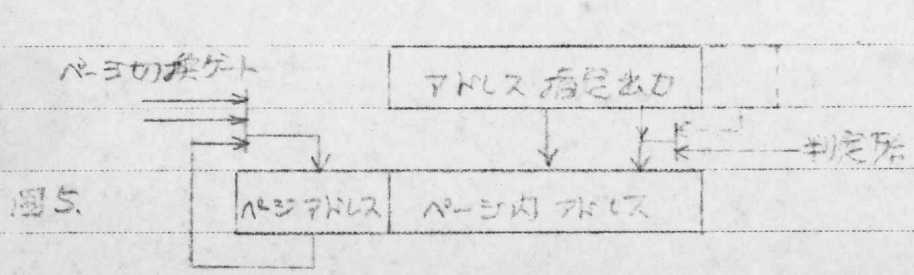


図5.

バージョンを 64 ~ 256 アドレス程度に割振り, アドレス指定出力ビット数を減らすこととするのである。バージョン切替の条件を取る際のラッチゲートを

必要とし、さらに、パーティアドレスリストとパーティ内アドレスリストとは異なる読みクロックを必要とする。

NEC 従来品種では、さらに指定出力ビット数を 1 ビット減らした方式を取っていた。(下図)

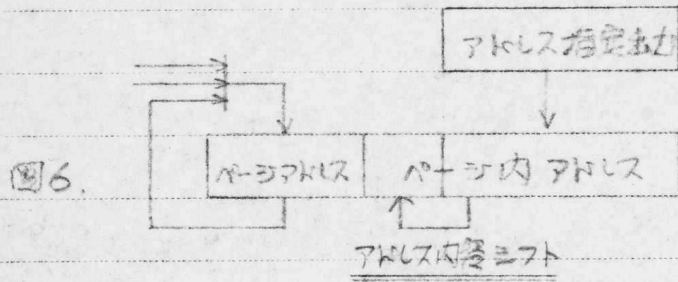


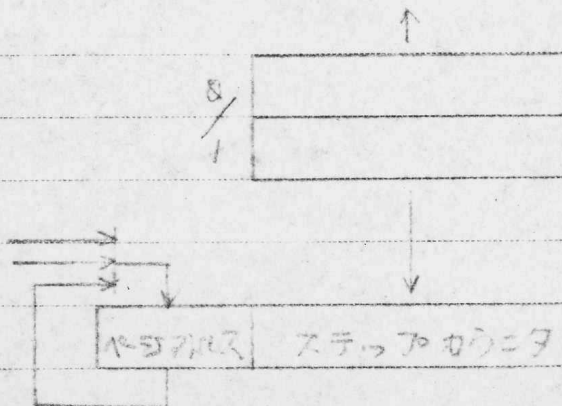
図6.

この様にすると、パーティ内でのアドレス指定についても、判段が生じてくる。

カレントパーティ以外のジャンプの場合は、パーティ切換ゲートを操作せねばならない。

II. ステップカウンタ指定

+1 可能なステップカウンタ (+1 だけでなくも逐次アドレスステップ可能なものもある) により、アドレス指定を行えるものも、直接指定の場合と同様にシステム規模により、完全指定を可能とするものと、パーティ内でのみの指定を可能とするものがある。アドレス指定のみに使われる ROM 出口を必要としなくなり、ROM 出力ビット数の減少する反面、アドレスステップがシーケンシャルに行われる、任意のアドレスにジャンプするには、無条件ジャンプ命令を常に付する必要がある。アドレス消費量が増す傾向を取る。さらにプログラム・テックを頻繁に行われる場合は、ROM アドレスに十分な余裕を確保しておかねばならない。ROM の有効な活用が、ROM 規模の小振りシステムの時、直接指定方式に代り、小さくなる可能性がある。



通常は、イニシャル、あるいは他の制御アドレスとして、アドレス・ロード命令の時のみアドレス指定部として用いる

図7.

UPD940C では、ROM アドレスが 256 アドレスと少ない事、アドレッシング、変更における アドレス割当操作の直帰化による工数削減、ROM 出力ビット減少化の反面、増加する 24 ビットの レイアウト上の集積率の面でも有意味が無く、むしろ、負荷 MOS、コネクタ数の増加(2)等の観点から、完全直帰指定方式を採っている。レイアウトしやすい回路設計を行なう事は、チップ面積減少に直帰的に繋がり、回路設計上の一つの眼目であると思われる。

(b) ROM 出力

詳細な命令については、巻末命令コード表を参照。

- XReg と ZReg を命令処理において同等の位置割込に置く事、即ち、 $x+1 \rightarrow x$ という命令が可能であれば、 $z+1 \rightarrow z$ という命令も可能である様にしている。
- 1, 2, 4, 8 の定数データを出力ビットによって発生させ、そのデータによる加減算、置数、判定の全てが実行できる。
- 判定旗セット条件は (1) 進条件キャリー、フロー (2) テンビット単位のフロー、フロー (3) キー入力発生有り、の 3 条件であり、判定旗は 1 個しか存在しない。この判定旗の状態は常に監視されて 10 の有無を 7-11H ROM 出力 C0 若しくは C1, C2 の状態によって判定をお互無視したり、分岐先アドレスを種々に変更したりする事ができる。

(50A-55 参照)

以上が UPD940C における論理設計思想である。以下、実際例について、専断のみを抜粋する。

- XReg/ZReg 処理切替は $B5 = \%$ で行なう
- Add/Sub 切替は $B1 = \%$ で行なう。
- 定数発生切替は $B4 = \%$ で行なう、そのコードについては B6, B7 によって発生させる。
- レジスタ操作命令は $B3 = 0, B4 = 0$ のとき有効となる。
- レジスタ間演算命令は $B3 = 1, B4 = 0$ のとき有効となる ^{10H/16H}

ほとめると次 A-2 図 8. の如くとなる。

		B4	B3	
レジスタ操作命令	////	0	0	符号化設計の方式としては ROM 出力ビットと論理ゲートとが 1対1に対応した直接制御方式であるが、広汎な命令を持つ場合には、出力ビット数が非現実的に大きくなってしまふ事、実際の命令では、全てのビットの關聯が排他的に成るべく、かなり多くの冗長度が必要以 ROM メモリー部を効率的に使用する事は言い難い事
レジスタ演算命令		0	1	
定数発生、定数データに おける演算、遷移、判定命令		1	0	
		1	1	

図8. 命令コード区分概略

等より、MPD940C のレジスタ- を最適論理設計において縮小化した、最小符号化方式を採っている。卓見として有効な命令の選別をも含んだ論理設計の出来物ゆへに、ROM 及びそれに付随する命令デコーダ- の大きさが決定され、ひいてはチップ面積が決定されると言っても言い過ぎではない。

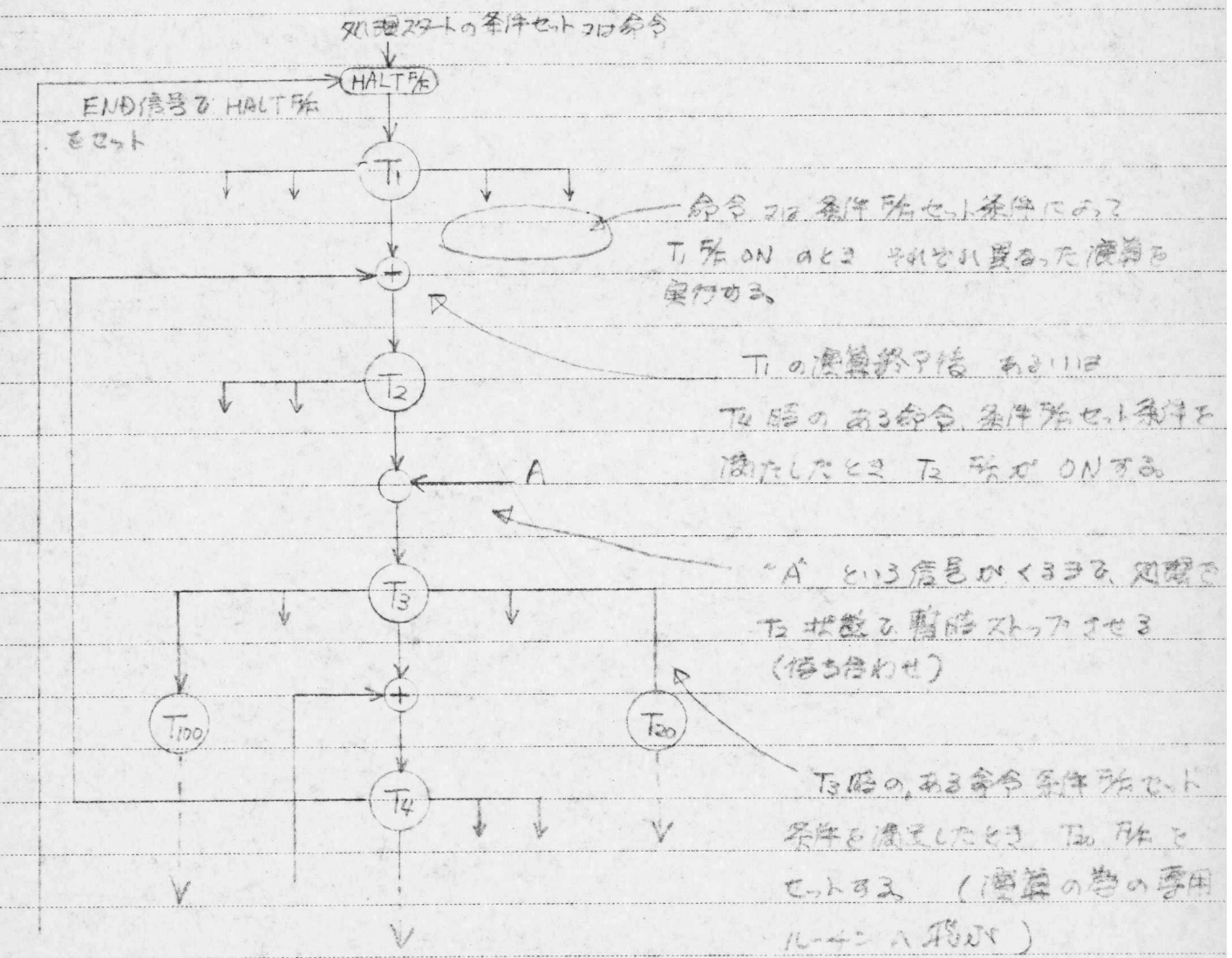
MPD940C で使用できる命令は豊富にあり、有効なものはかりを選別して13行のメモリー用としては、せつたくすぎる感がないわけでは無い。

(C) マイクロ・プログラム方式とワザ-ド・ロッキング方式

MPD940C では、コントロール ROM 内に書き込まれたマイクロ命令を、1マシンサイクル（メモ・レジスタを共用しているのち、1サイクル・タイム = 48クロック・サイクル）毎に読み出し、演算処理をする。マイクロ・プログラム方式を採用している。（但し、マイクロ命令として固定されたものを、さらに用途に応じて有効なものとする為、広張変更する事は、不可能である。）この為、仕様変更に対しては、マイクロ・プログラムの書き直しによる ROM メモリー部トランジスタの位置変更のみを済み、自由度の大きい設計が可能としている。

ワイヤード ロック方式には 1ワード毎にシフトするレジスタ (遅延線) により 状態を変化させて処理を遂げる初期の段階の方式と 状態を数ビットのレジスタに記憶させ 条件所の変化により 状態レジスタ内容を変更して処理を遂げる方式 (ワイヤードロック方式のマルチステップに類似しているが本質的に異なる。) の 2種あるように思われる。

(1) シフトレジスタ (遅延線) を用いた方式例



終了 (END)

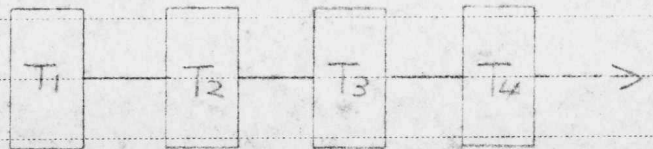
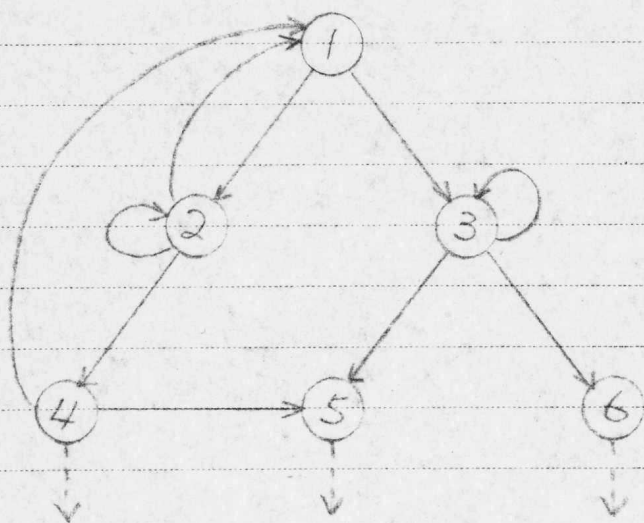


図9. ワイヤードロック方式例(1)

(ii) 状態リスト・コード方式例

状態数を16とし、リストを4ビット使用した場合



(i)と同様に各状態において行われる演算は命令、条件等により条件によって各々異なる。一般には「 $x-1$, $x+1$ 」等の類似した演算とす。

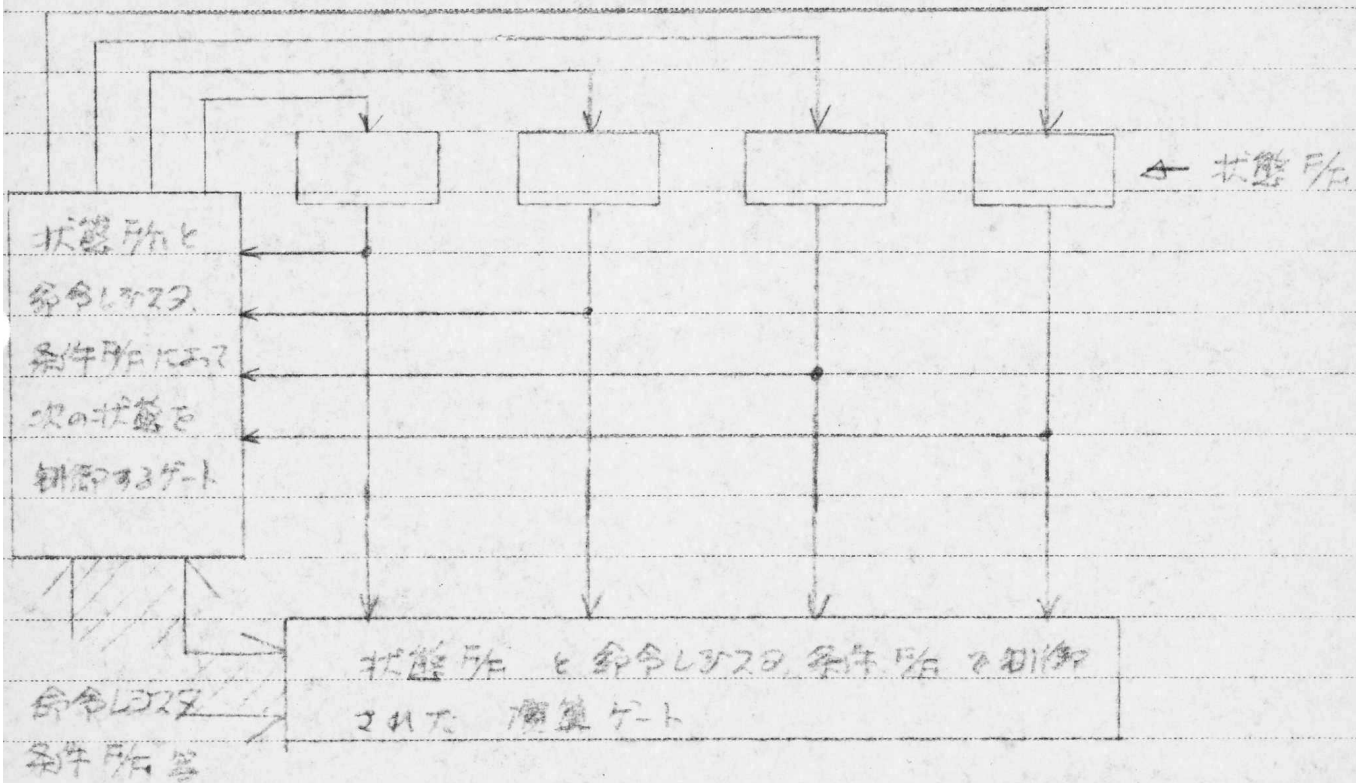


図10. ワザ-4.0のコード方式例(2)

(ii) の方式の場合、各状態における命令を多岐にするものとする。各状態のなかで、シフトレジスタ（遅延線）を用い、順次、命令を変更したり、条件所による制御されるゲートを増したりする手段を用いる。

又、ワイアードロシフ方式では、フローチャート作成、命令設定終了後にフローチャートに沿った、シーケンスコントロールのハードウェアを設計せねばならず、設計工数の増大、設計自動化による設計負担の軽減、設計変更の非容易性という欠陥がでてくる。

マイクロプログラム方式とワイアードロシフ方式との根本的な相違はシーケンスコントロールにあると見てよく、この相違がシステム全体に大きな差を生じさせている。（各方式の得失については文献参照）

(4) シェア卓電設計方式

ステージ I（キー入力受け付け、コンディショナルな制御、HALT/START 判断）とステージ II（演算命令実行）にコントロール ROM 内容が大きく区別される。ステージ I では (i) 例にあるような完全なワイアードロシフ方式の回路が組まれている。ステージ II では (ii) 例にある方式に (i) を加味した。これも、ワイアードロシフ方式を採っているが条件所による制御は少なく、判定所による状態変化が多い（演算結果による判定条件は非常に多い）ので、マイクロプログラム方式に近似しているものである。但し、状態数が少なく、且つ一つの状態のなかで (i) 例にあるような順序制御による固定された（この命令に変更を許さず、(ii) 例下段に常に分岐命令を添えてきて、固定命令に若干融通性を与えている。）命令を実行させるので、演算ループ内での仕様変更による細かい操作を ROM のみで実行する事は、分割がた難しい。さらに卓電において仕様変更の対照となるものは、殆んど、ステージ I の部分あるいは表示方式であるので、コンディショナル所を多用したシェアワイアードロシフ方式では、所の増設配線部の変更を行わねば、仕様変更は 100% 無理である。

東光ビジネスマシン(商品名ユニックス)の設計方式はチップ方式
以上に、ワイアードロッキング化されており、又、3年前、東京電子応用研究所
との協力を、開発した3としたMPD270も、この方式であったと思わ
れる。

ワイアードロッキング方式は、個別半導体、SSI 時代には、使用トランジスタ数
が少なりという事で、優位性を持っていたが、卓上システムが1チップ化
され、ワイアードロッキングが格納されるROMの面積を小さく出来る様
になった現在は、卓上仕様変更の各段にわたって出来なくとも良いと
言う次元にまで立ち戻ったとしても、マイクロプロセッサ方式の方が
全般的に'可'であるという評価を下しても差支るまいと思う。

但し、同じ、マイクロプロセッサ方式を採る場合であっても、その用途が
汎用のものであるか、卓上という特殊分野に絞られるのかによって、
そのアーキテクチャには当然、変化があり、どの程度までマイクロ細分化
した命令とするか、どの程度まで、固定化されたハードウェアを用い
るかに、その最適な各協賛を見出す事が大なる問題となる。

(C) キー入出力回路、表示回路

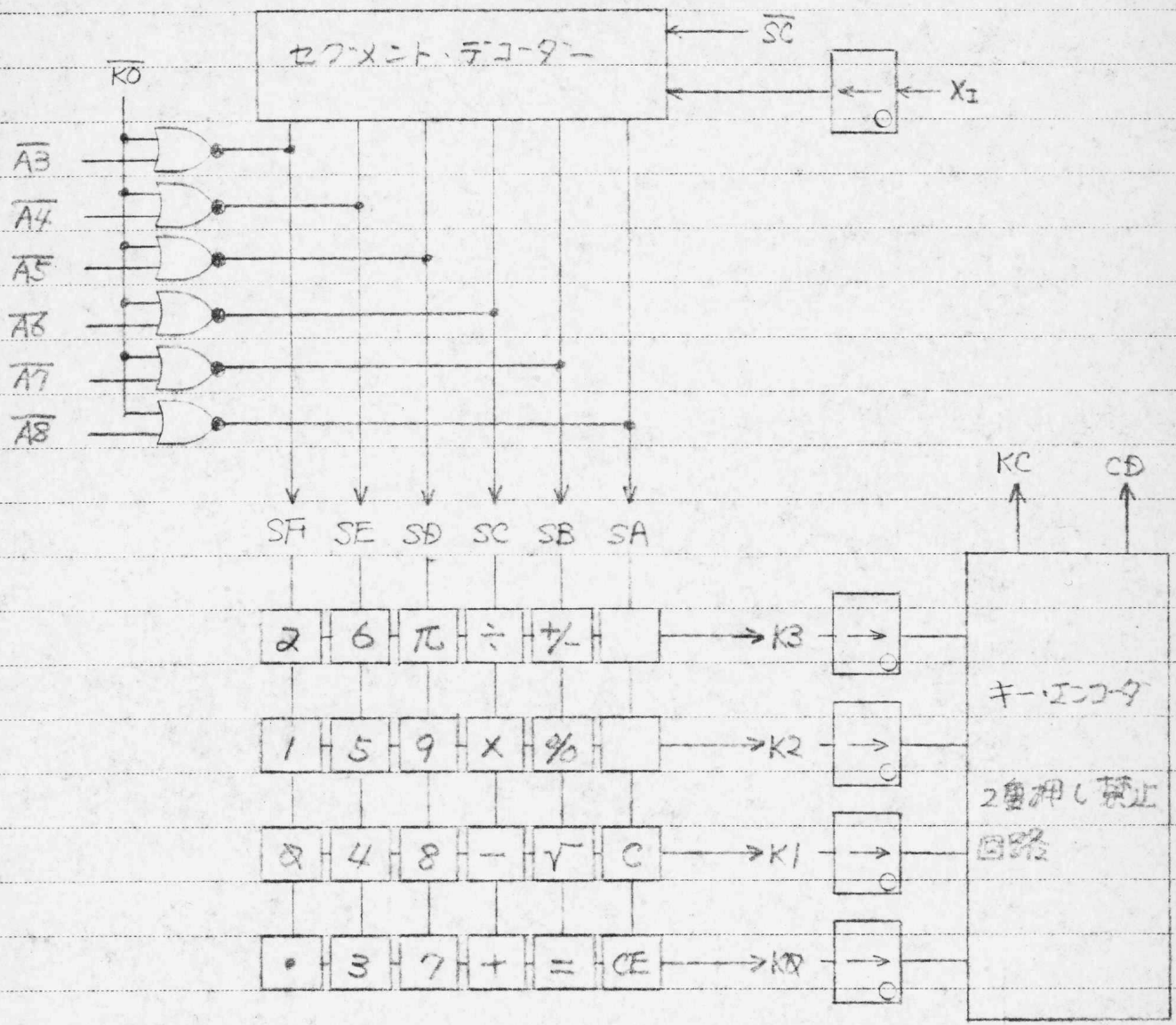


図11. キー入出力回路及びキー配置

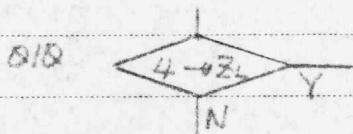
従来、キー入の供給信号として、ヒート・タイミング出力あるいはデジタリ・タイミング出力を用いていたが、ヒート・タイミング出力を用いると出力端子数が4本増加する事、キー入出力間での遅れを補償する事、ノイズに弱い事、外付素子の増加する事、キーの2重押しに対する保護をLSI内部で行う事等の難かしい事等利点は無い。さらにデジタリ・タイミング出力を用いる場合、表示管・スクリーン出力を兼用出来る事、好都合である。エンコードする為、キー入力ラッチ、4ビット・カウンタ(不発動の事を示す)の1つだけにして、シフト的に見ると大きくなる。)を必要とし

さらに 2重押し防止回路に至っては、格許の対応となる程、大きな回路を必要とする。

UPD940C では、ROM アドレス・ビットラインをデコードせず（通常コンピュータでは 入出力機器を、アドレス出力を割当てる為、外付回路として、アドレス・デコーダを接つ。）直接、セグメント端子に出力している。この方式は、キーの数が 24キー と少ない為には可能（卓電においては、24~28 キー程度あれば、充分。）なのはあるが、キー数が増加すると、不能となる。

図12. の様に 数値表示セグメント出力とキー供給の為のアドレス出力を時分割して出力する。この制御の為に \overline{KO} , \overline{SC} 信号を用いる。セグメント出力信号は、演算中には出力されない、表示中であれば数値表示、キー供給 11個の用途のものもあると出力される。しかしながら、キー信号有無を判定するアドレスは、 $B4=B8=B2=1$ の ROM出力が出力している場合のみに限られる。巻末フローチャート No.1 において、003番地 ~ 100番地 に至るまでのフローは OFFチャタリング防止、ONチャタリング防止をも含んだ キー関係フローであるが、キー有無の判定を行なうアドレスは、374, 377, 201, 004, 010, 020, 040, 100 の各番地であり、キーが押された場合は、キー・エンコードデータとその命令を示される定数データとを重ねて、指定されたレジスタ（キー入出力タイミングの関係を Xc, Xs, Xd のタイミングに限定される）に読み込まれる。

(例) ⑤ キーを押したとき



010番地にくると ⑤ キーに供給している SE出力端子に信号が現われ、キーを押している間は、K2入力が "H" となり、2重押しである場合は、判定所

をセット。キー・エンコードデータ "2" と "4" とを重ねて "6" を ZL に読み込む。

キー・エンコードは $K0 "H" \rightarrow "0"$, $K1 "H" \rightarrow "1"$, $K2 "H" \rightarrow "2"$, $K3 "H" \rightarrow "3"$ を出力する極めて簡単な回路を使用している。

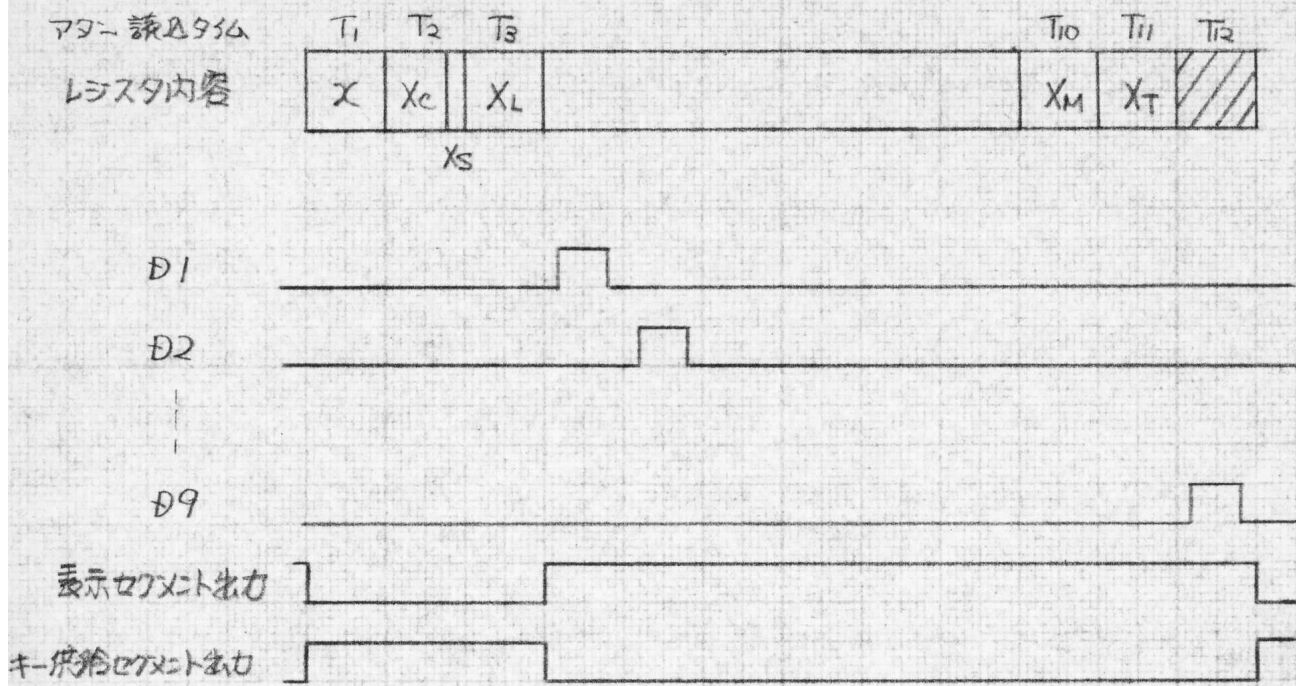


図12. セグメント出力時分割

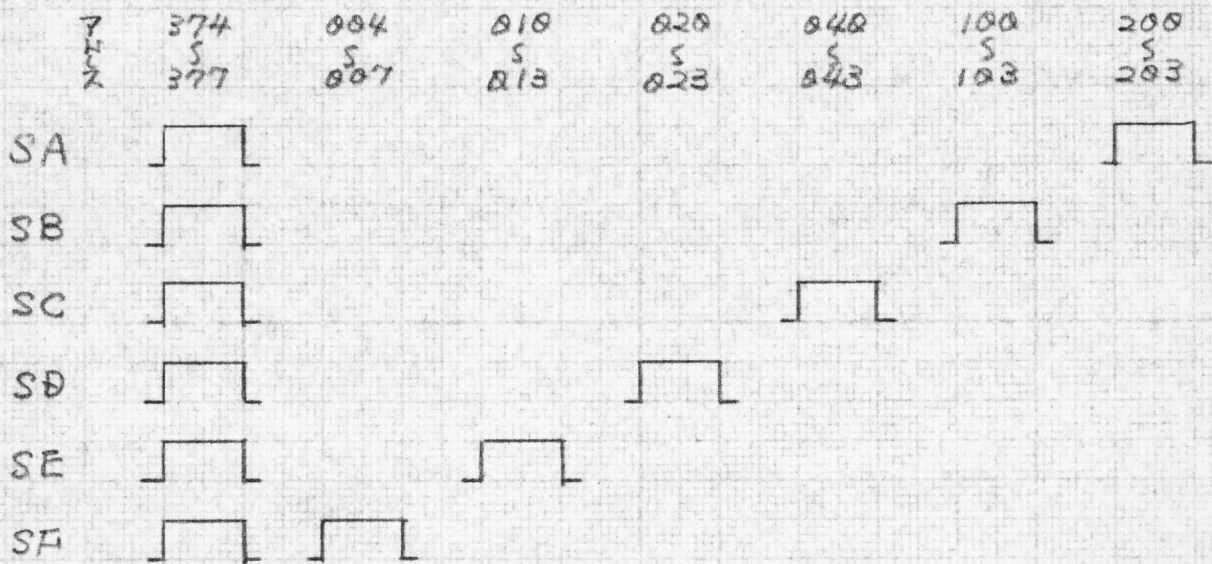


図13. マルス - キー供給信号

(1) OFF チャタリング, ON チャタリング防止

キーを離したときに生じるチャタリングによる誤動作, キーの押形割れ (バウンス) あるいは, キーを押したときに生じるチャタリングによる誤動作 (キーを連続打した現象を呈する) を全て防止するソフト的処理を行っている。 フロート No.1 003~002番地が OFF チャタリング防止 フロート, 377~031番地が ON チャタリング防止 フロートである。 チャタリングを防止するには, 全キーのうち, 1つ 又はそれ以上のキーが押されているか, さらに, 押されている (又はいる) 時間は何 msec であるか, という事を判断, 計数する必要がある。

キーが押されているか否かの判断は, 図13. にある様に, アドレスを 374~377番地の 1つだけかに選択する事によって, 全てのキー供給線を "H" とし, 1アドレスにて判断可能としている。(これがこのキー入出力方式の回路的な意味での大きな特徴である。) 時間計数については, 汎用レジスタ X_c の 3ビットを用いてソフト的に処理をし, 専用のカウンタは用いていない。この計数は Φ_1, Φ_2 フロックを基準時間発生器としたものがあり, 通常コンピュータで行なわれている水晶発振器又はワンショットマルチを基準時間発生器とした (システムフロックは高周波であるので直接を用いる事が出来ない為) WAIT・ルーチンであり, 特徴のあるものである。

キー有無判定と, 計数に同じ X_c を用い +1, +2 を行っている為, キーの重りによってチャタリング防止時間が異なる。これを避ける対策はあるが, チャタリング防止時間はいくらにすれば良いか, 定まった説がなく, ある程度以上の時間 (バウンス... >10 msec, チャタリング... >4 msec) を保証する事は良い事, フロック発振器を内蔵している為, 発振周波数は大きく変動する事などから, この事は "可" としている。又, チャタリング防止時間については, ソフト的処理をしている為, ユーザーの要求において自由にその時間をアダプツする事ができる。

(ii) 表示回路

演算中には、セグメント出力、デシマル出力とともに“L”レベルに固定し表示をマスクする。さらに表示中であつても、右詰め表示形式があるので、整数部の不要な“0”はマスク(リーディング・ゼロ・カット)し、マスクされている表示桁については、桁信号を出力しない回路を組んでいる。又、ダイナミック表示をさせている為、セグメント信号と桁信号との間に相対的な遅れがあると、桁向セグメント表示のコントラストが悪くなるフリップキが起るが、これを防止する為、桁信号のみに、 t_{ON} 、 t_{OFF} の間のみ、カット信号を挿入している。

DP表示信号は、 Σ Reg. 出力から得ているが、表示中に CPU マターを使用している事、論理ゲート数、レイアウト的に優利な事、検査用テストパターン作成上、非常に有効である事等の理由に依るものであり、その表示処理についてもソフト的に処理している。

(iii) 表示/リセット命令

従来 NEC 標準 LSI では、演算処理が終了すると必ず、000番地に復帰し、表示を行なう形式を採っていたが、MPD940Cでは従来ハードウェアを行なっていたものをソフト化に置換する事を基本方針としている為(ラッチ、ロコックゲートを極力ROMに集約する。)従来形式は必然的に採れない。又、キー供給信号として、アハス・ビット・ラインを流用しているの、アハスの冗長組合せにより、表示/演算の切換を行なう事は難かしい。その為、表示/リセットというハードウェアを必要としている。(ラッチ形式の所は、これ以外に、判定/リセットのみしか存在しない。)

セツト命令 --- $\Sigma \leftrightarrow \Sigma$

本質的には NOOP であり、B6~B10出力ビットは、どの様なものでも可

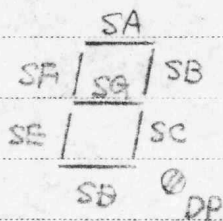
リセット命令 --- Xcち

キー・スタート後、キー・識別判定に用いられる命令であるので流用する。

B4=1, B3=0, B2=0, B1=1 であり、他のビットは、どの様なものでも可

表示素子は、7セグメント・テミナルポイント付 蛍光表示管を主対象とする。

セグ	0	1	2	3	4	5	6	7	8	9	10,11	12	13	14	15
SA	○		○	○		○	○	○	○	○		○	○		
SB	○	○	○	○	○			○	○	○					
SC	○	○		○	○	○	○	○	○	○					
SD	○		○	○		○	○		○	○		○	○		
SE	○		○				○		○			○	○		
SF	○				○	○	○	○	○	○		○	○		
SG			○	○	○	○	○		○	○		○		○	
字 #3	□	□	□	□	□	□	□	□	□	□		□	□		



表示管のラジヤの規格

セグメント出力 $V_{OH} = 1.0V$ $I_o = 1mA$

テミナル出力 $V_{OH} = 2.0V$ $I_o = 3mA$

図 14. セグメント表示

(iii) キー関係フロー

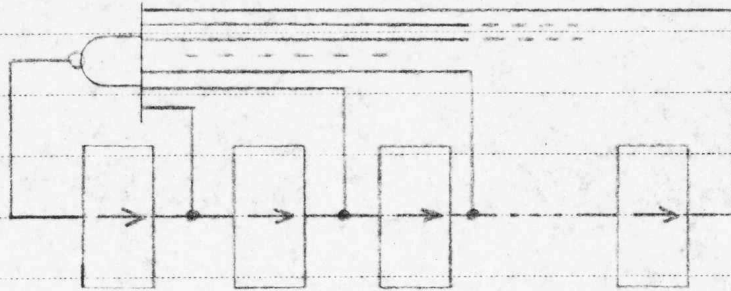
電源 ON 後、自動クリアがかかり、レジスタクリア、表示処理をリフト始に行る。377 ← 030 番地間を行来まし、キー入力待ちタイミ、ストップ状態にある。キーが押されると 377 → 031 → 200 番地で ON キョリニク計数を行る。201, 004, 010, 020, 040, 100 番地の11がけがよりスタートし、演算終了後、未だキーが押されなければ 374 ← 003 番地へ、ストップ状態となる。キーが離されると、374 → 002 → 376 番地の OFF キョリニク計数フローを通り、377 ← 030 に戻る。キー有無の判別は、2ワード毎に随時行るわける。

○ バウンス防止時間 20ワード (キーの如何に相ら可)

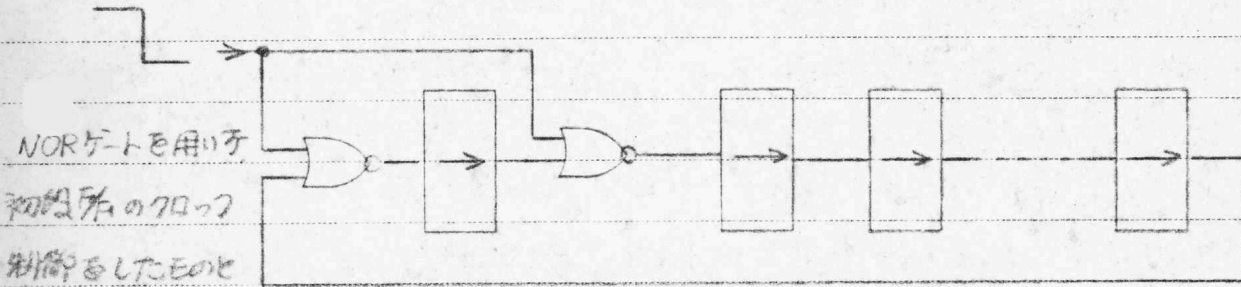
○ ON キョリニク防止時間 6 ~ 9ワード

(d) タイミング・ジェネレータ

タイミング・ジェネレータの種々の回路を次に示す。

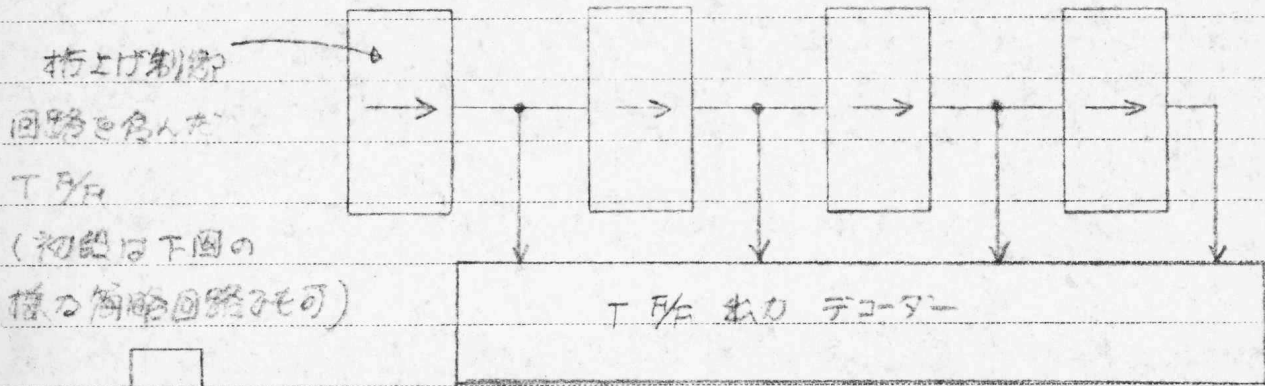


(A)



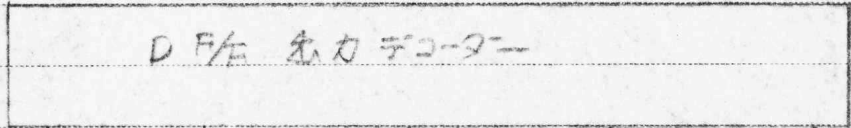
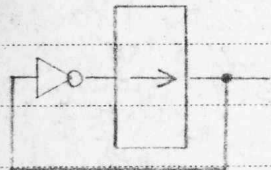
NORゲートを用いた
初段からのクロック
制御をしたEのセ
等価。

(B)



桁上げ制御
回路を含んだ
T/F
(初段は下図の
様な簡略回路でも可)

(C)



(D)

図15. タイミング・ジェネレータ各種回路

④ 回路は旧来 シャープカスタム品に用いられていたもので、NEC
は新標準品 MPD281C, MPD282D 用の タイミングジェネレーター
MPD262C にも使用されている。

⑤ 回路は カシオカスタム品 MPD1749b に用いられ、さらに NEC
MPD284C, MPD234C シリーズ品 に使用されているが NOR2 入力
波形として、急峻な波形のものを入力すると、正常な発振をさせる
確率が小さくなるという欠陥を持っている。

⑥ 回路は 東芝カスタム品、栄光カスタム品、最近のカシオカスタム品に
用いられている標準的な回路である。

⑦ 回路は、ROM を用いた バイナリ・カウンタ方式であり、NEC,
MPD274C, MPD277C, MPD735C に使用されている。D 出力
デコーダー、74HC157 出力を持った ROM と D 出力 とを構成され
小さい面積で作る事ができる。

MPD940C では、⑥回路と⑦回路を融合させ、1段発展させた
回路を使用している。

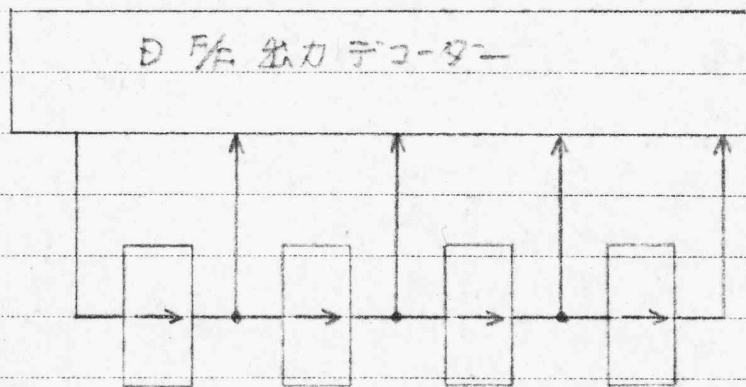


図 16. MPD940C タイミング・ジェネレーター (フロント)

この様になると、ROM 側の 74HC157 が 1本だけ減り面積的
に有利になる。

ヒート・タイム・ジェネレーター としては出力が 4 種と少ないので、④図の
回路を使用している。

(e) フォーマット, 演算符号

AAA $\boxed{\leftarrow}$ BBBB

AAA はアドレスを示し, $(A_8, A_7), (A_6, A_5, A_4), (A_3, A_2, A_1)$ に従って
オプティカル表記をしている。

BBBB は 40ビットを示し, $(B_{10}, B_9, B_8), (B_7, B_6), (B_5, B_4), (B_3, B_2, B_1)$ に
従って, オプティカル表記をしている。

次のアドレス指定ビットについては, 記載していない。

No.1 \rightarrow キー入力 フォーNo.2 \rightarrow 置数 フォーNo.3 \rightarrow 数値左詰め, B/W 演算前処理 フォーNo.4 \rightarrow 加減算 フォーNo.5 \rightarrow 乗除算 フォーNo.6 \rightarrow 演算後処理, B/W 表示処理 フォーNo.7 \rightarrow π 置数 B/W ルート演算 フォー

(i) 命令使用法抜粋

- \leftarrow 実質的に NOOP であるので、アドレス指定部のみを
使用した無条件ジャンプ命令として用いる。
- \leftarrow 実質的に NOOP。表示左セット命令として用いる。
- $X \geq 10$ マチーの 10進演算モードにしてレジスタ・データに
入れている数値により、可能としているが、“9”以下の数値データ
(-, L, E, カタリスト・モードは 10以上) を左詰めする際の
検出判断に用いる。
- $X < C$ $C=1$ としたときは $X=0$ の判定とし、 $C=4$ としたときは
 $X < 4$ の判定をする事ができる。
- $X + C$ この命令のときのみ、レジスタ単位のキャリー、ホロー判定
はなく、無条件のキャリー判定を行なう。演算制御
プログラムのビット判定に主として用いる。表示左セット命令
と異なる。

- $X \rightarrow RS$
 $X \rightarrow C \rightarrow X$
 $X \rightarrow LS$
 $X \rightarrow C \rightarrow X$

数値データの RS LS を行うときには小数点データの -1, +1 を行なう場合が多いので、1ワード毎に上記命令を同時に実行させる。さらに 'C' とは 1, 2, 4, 8 の定数の任意に得られ、演算のシフトとしては、 X, X_C, X_8 のいずれでも良い事から、演算回数制御を同時に行なう事も可能である。
- $\textcircled{C} \rightarrow X$

前のレジスタ状態を変化させるに 'C' を重複、置数する命令である。

(例) 2001 \textcircled{C} のみセットされているとき、回キーが押された \textcircled{C} フラグをさらに立てたとき $200C$ とすきと 2011 となる。
- $C \rightarrow X$

前のレジスタ状態をクリアして、改めて 'C' を置数する命令である。

表示部がセットされている状態では、 $\textcircled{C} \rightarrow X, C \rightarrow X$ は、キー信号有無の判別命令ともなる。

(1) 演算時間

従来より、マイクロプロセッサ方式の卓電は、演算時間を短縮する様なハードウェアを使用せず、1ステップ毎に命令を読み出し、実行していく為、演算時間が長いとされているが、演算スピードを犠牲にして、ハードウェアを少なくしたハードウェア方式卓電よりは、MPD940Cの演算スピードは高速度である。MPD940Cは、数値表示処理、DP表示処理を、ソフトウェアで行なっているにも拘らず、NEC 8桁標準品 MPD271C, MPD274C, MPD277C のいずれよりも、演算速度は速くされている。

次ページ 図17 参照

UPD 940C クロック周波数一演算時間

mSec

700

600

500

400

300

200

100

20

30

40

50

60

70

80

90

100

CLOCK KHZ

- 1---99999999 √
- 2---99999999 ÷ 0.000001 %
- 3---0.0000001 × 99999999 %
- 4---2 ÷ 3 =
- 5---2 × 3 =
- 6---2 - 3 =
- 7---2 + 3 =
- 8---置数, D.P.

1

2,3

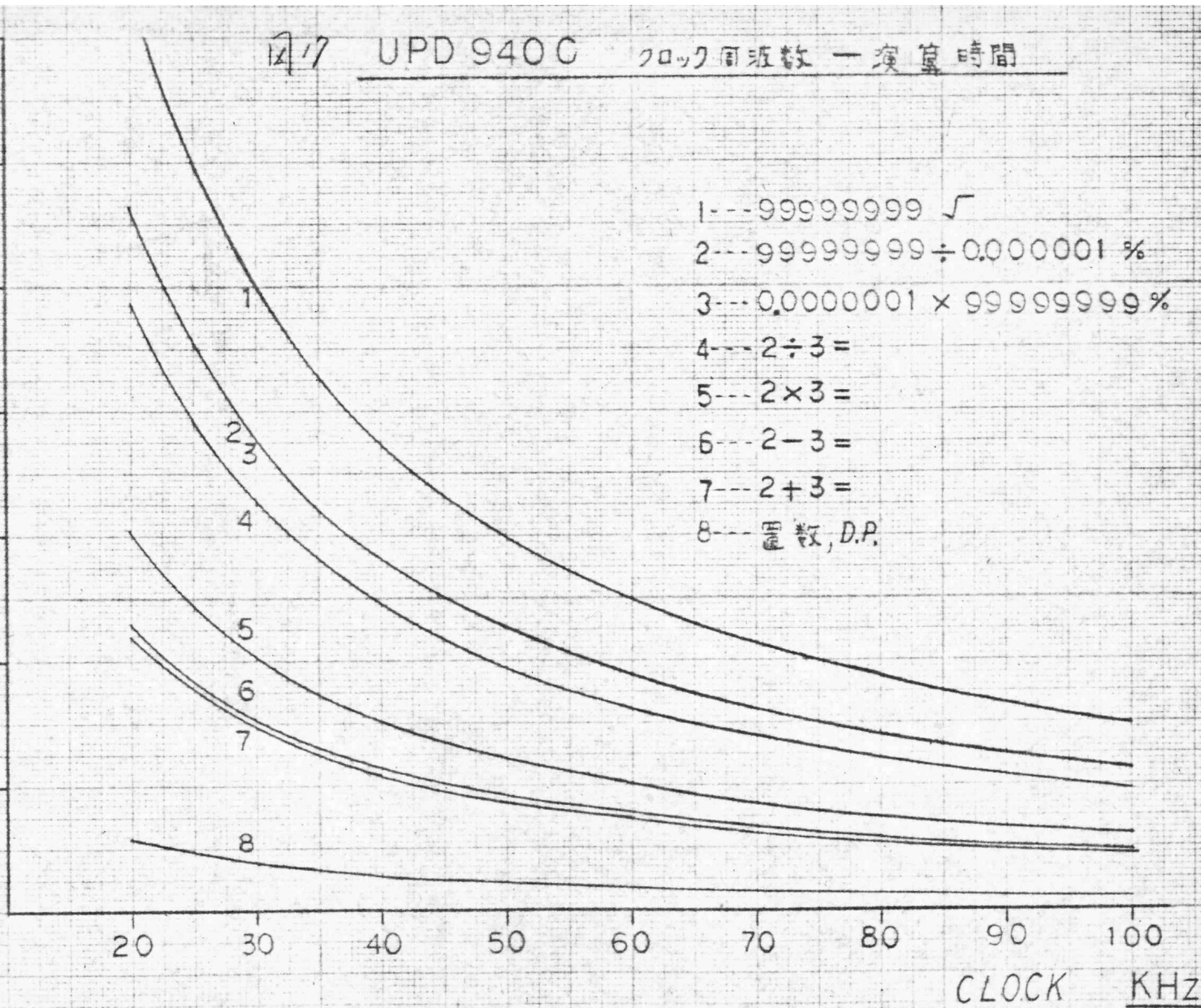
4

5

6

7

8



(F) 総括

(i) 動作周波数

加減乗除計算の演算時間 t_{op} は、 $f=40\text{ KHz}$ としたとき 0.3 sec以内
に収まる。従って最高動作周波数は 100 KHz 程度あれば
充分であるので、過剰な回路構成を取る必要性は無い。
従来、NEC 標準品では、ROM アドレス・レジスタの読み込みクロックを
 T_{last} 中の t_3 に置く (T_{last} 中の t_3 、 t_4 のレジスタ・データは使用不可
として) 実際に ROM 出力を使用するのは T_1 中からであるの
で、その間 2 ビットの ROM 出力の遅れが生じても、阜重としては
動作する様に設計されていた。さらに アド・読み込みタビニク
信号発生回路についても、1 桁か早く発生させ、(ROM 出力変化
時 1.5 ビット余裕) その信号を 1 桁遅れを持たせ、その後で
波形整形をする回路を組んでいた。ところが実際に動作
周波数の上限を決めている原因は、クロック・バス部分、クロック
ゲートを使用しているものは、クロックゲートのレベル不良、クロック・コンテ
ィンタを内蔵しているものは、その発振上限周波数、その発振波形形造
(クロック幅、間隔)、クロック関係配線レイアウトの不備、等に
ある。又、レジスタ ROM を使用したものについては、カンブリック
デシマシーを 1 ビット毎に行なう為、ROM 出力は 1 ビット単位を
更改させ、ROM 出力に対して、数ビットの遅れ余裕を持たせる
意義は全く無い。

以上の様に、システムに対する従来の設計方針には、多少
実際と矛盾する点がある為、MPD940C では、標準周波数
から低い事もあり、上記の処置は採っていない。

MPD940C では、システムに対し、次の様な配慮をしている。
 ○ ROM アドレス・レジスタ 出力を T_{last} 中の t_1 出力し、実際に ROM
出力は T_1 中より使用する事にし、ROM 内部 (インタ 2 段、ホロン
配線が長く、負荷 MOS の L が大きいので立ち下り遅れが大きい)
での遅れについては、0.5 ビットの余裕を持たせる。さらに
立ち下り波形のなまら ROM 出力を直結ゲートに配線し、さらに
遅れを大きくさせる事を無くす為、ROM 全出力端に波形整形用

1段バッファを挿入する。(以上の措置を取れば、150KHz程度まで充分動作するというコンピュータ・シミュレーション結果を、製設・飯岡氏より得ている。)

- クロック・ジェネレータの発振形態において、 $\phi_1 - \phi_2$, $\phi_2 - \phi_1$ クロック間隔を $1.5 \mu\text{Sec}$ 程度にせよめた クロック・ジェネレータを使用する事により、クリティカルパス部における遅れ余裕を大きくする。
- その他、レイアウト配慮。

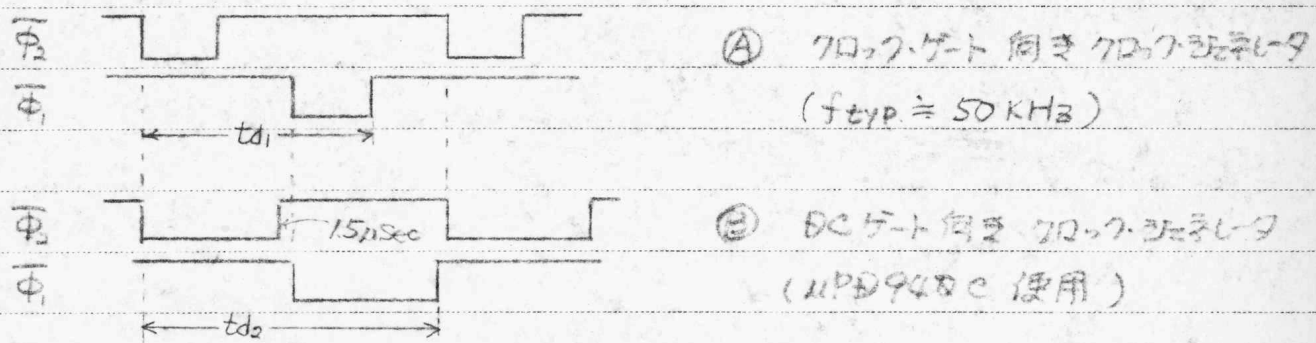


図18. 各用途向けクロック波形

信号は ϕ_2 を感化し、ゲートを通過して ϕ_1 においてレジスタに読み込まれる。従って、信号の遅れは、①の場合 td_1 、②の場合 td_2 期間に許容される。③の波形を採った クロック・ジェネレータを用いた方が遅れ余裕は大きくなる。(実際には、 ϕ_1 の信号を伝達する時間を加味せねばならぬので 図の td_1, td_2 は多少、短くなる。)

(ii) クロック・ゲートの使用に関して

クロック・ゲートの使用目的としては、消費電力低減を狙ったものと、ゲート数を減少させる目的を掲げたものがある。MPD948C においても、一部にクロック・ゲートを使用しているが、これは後者の目的を掲げる。その内容を次ページ 図19. に示す。

いかにせよ、ビット単位で、ゲートを閉鎖するデータバス都合にクロック・ゲートを使用した場合には、動作周波数上限を決定する要因となる。さらに、使用周波数が元来高く、 ϕ_2 ON 状態の時に、

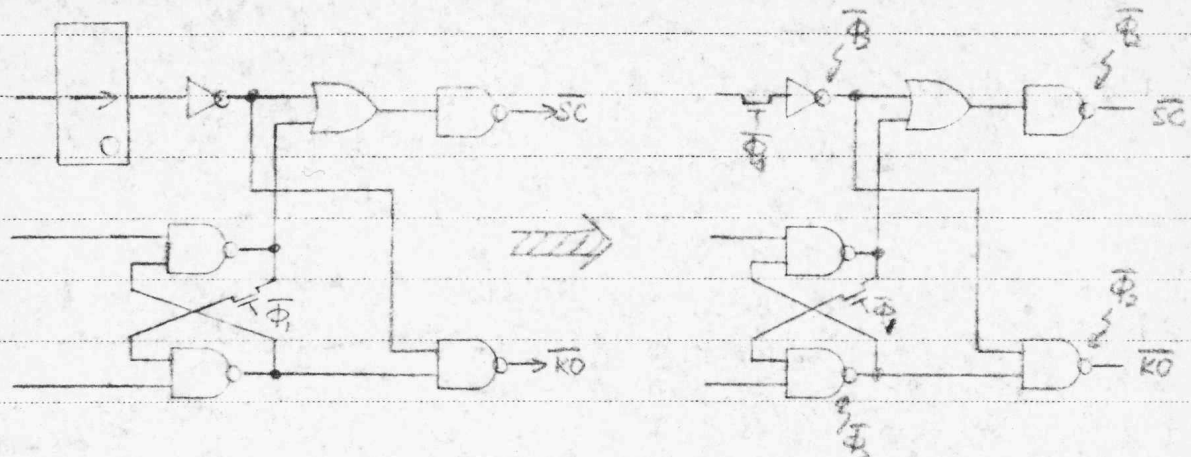


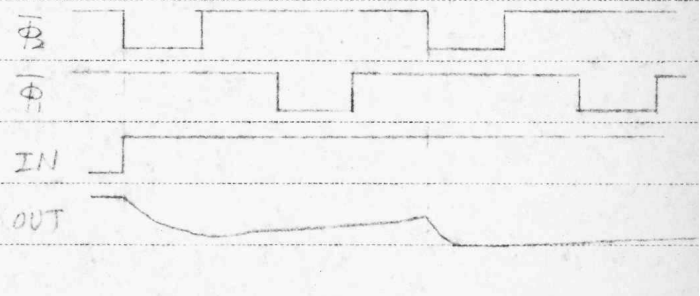
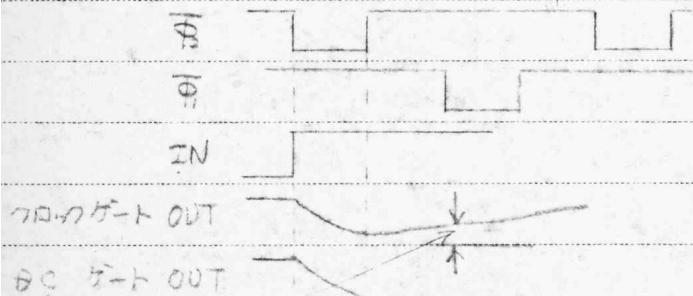
図19. 7400, 7401ゲート化によるゲート縮小 (UPD940C回路図参照)

電源電圧レベルにまで充分充電できる1つの場合には $V_{th\min}$ 値を決定する事になり、結果的に、右の7400, 7401ゲート化による消費電力削減の意図が充分果せるようになる。(図20, 参照)

入力信号がビット単位では無く、1ワード程度の長い期間、入力される場合、あるいは同時に長い期間同じレベルを保持する信号を取扱ひ、その信号のレベル変化後、数ビット程度の遅れ余裕を持つ回路には、右の7400, 7401ゲートを用いても、電源動作範囲、最高動作周波数に於ける影響はさほど大まくなり、(UPD940Cでは、図19. にある様に、この類の回路にのみ、右の7400, 7401ゲートを使用してよい。)

図20.

図21.



この周波差は、7400, 7401ゲートを使用時に於いて、 $D-L$ レベルが異なる。これを補償する為、 V_{cc} を上げ、等価的に電圧を上げた形を取りに事が出来る。

入力信号変化1ビット目には充分充電できるとも、2ビット以降は十分に充電できず、その結果、

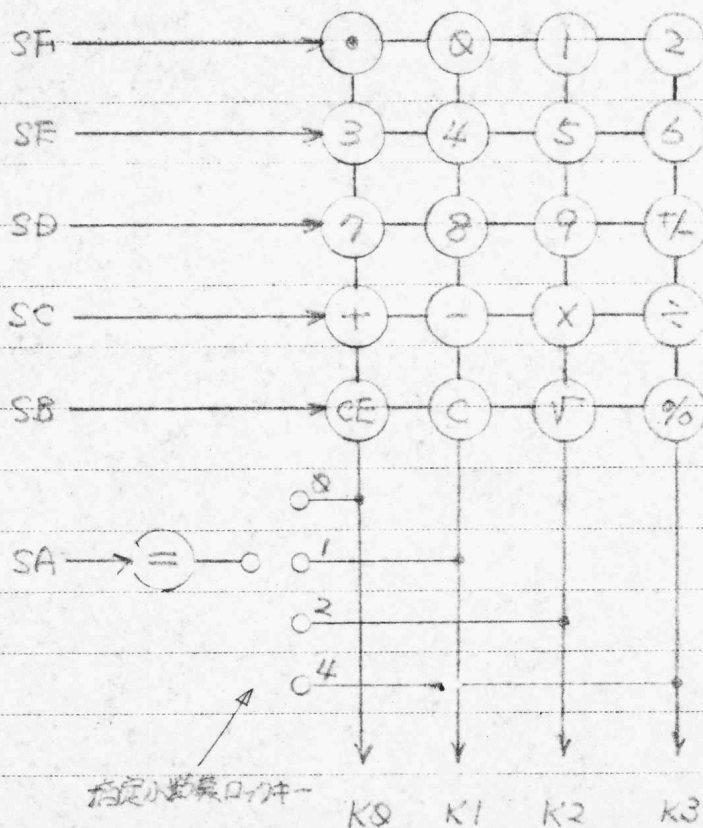
(iii) 仕様変更

前述した様に、ROMメモリー部トランジスタの位置変更のみで種々の仕様変更を容易に行なう事が出来る。

MPD940Cを基本として、スワフ・タウンとした機種として、MPD941C（四則、%計算）、MPD942C（四則、 $\sqrt{\quad}$ 計算、 π 常数）、加算機式卓電 MPD943C、シフトキー仕様を満たした、MPD921Cを相次開発し、2 開発する予定がある。

以上の機種ではロックキーを使用しているが、図22.の様にキー組配線を行なえば、キー入力を増設せずに、ロックキーを装備する事が出来る。

図 22. 指定小数点ロックキーを装備した例



[3] モックアップ (フレットボード)

回路設計者が期待した通りの回路機能を実現しているか、卓電仕様を満足する様に、ボードが作成されているか、実際に個別部品、SSIの回路を組み、検証をする為にモックアップを作製する。モックアップの調整には、システム、回路を習熟する必要がある。その習熟度の違いにより、調整に要する工数が大きく異なる。

モックアップ用回路素子として、SSIを用いる場合には、次の素子を基本として、ワイヤード・オフを多用すると、使用SSI個数、部品枚数、本数が減り、デバッグ時に生ずる回路変更の作業も大きく省ける事となる。さらに、LSI回路図面とモックアップとは、論理的に等価であるのは良いので、モックアップ縮小化の為に、LSI図面に対し、種々の論理変換を行なう事は、全く差支えない。

図23. 基本素子

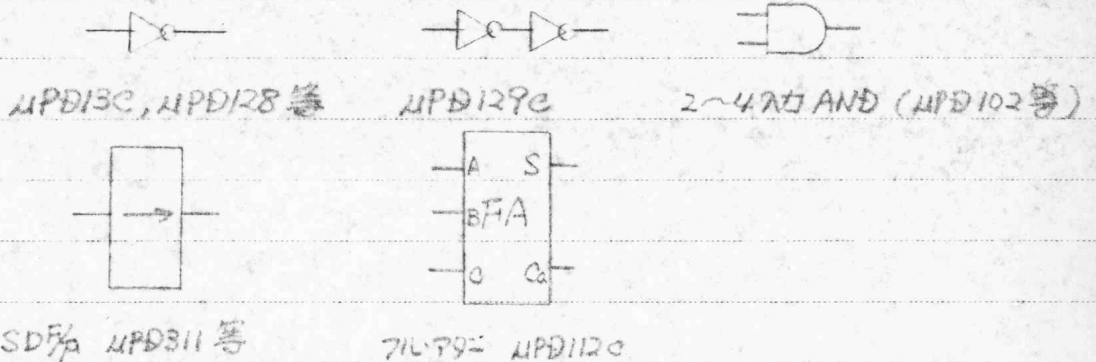


図24. LSI回路図面 ⇒ モックアップ図面 変換例

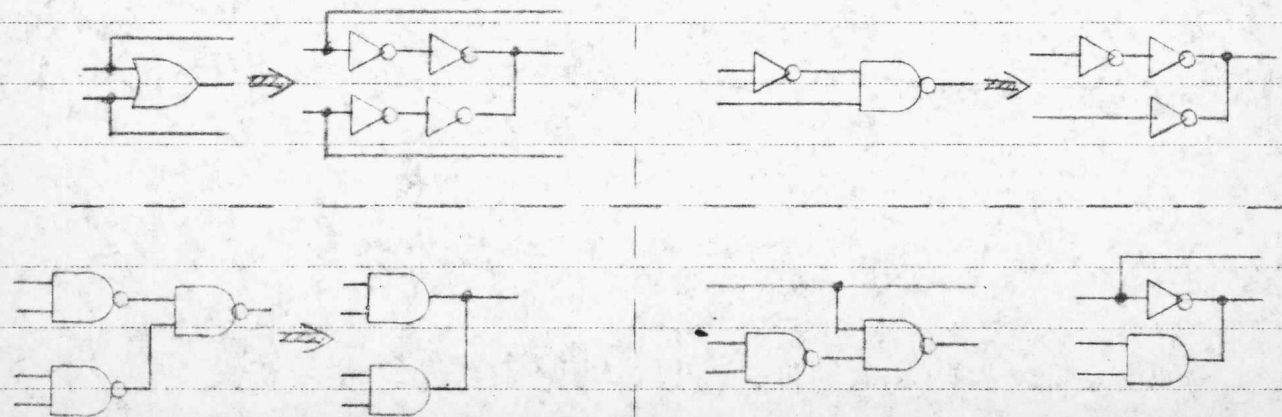
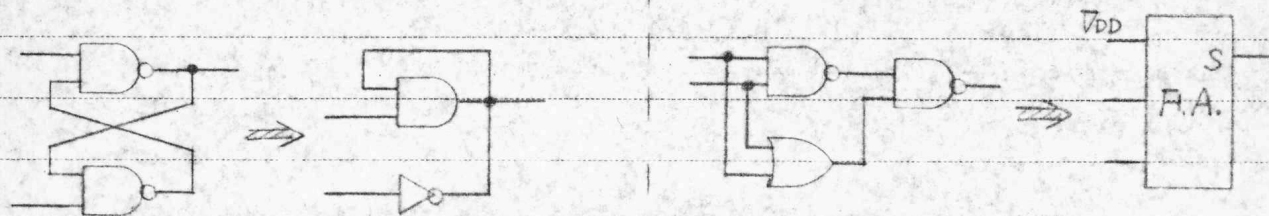


図27 続き



モックアップの調整を容易にする為、デバグ用回路を構築する。

- 演算回路、レジスタ交換の ROM から出される命令通りに動作する事を確認する為、ROM 出力コードをスイッチによって指定し、ステップキーを押す事によって、1ワード期間のみ命令が発生する様にする。
- ROM のアドレスをスイッチによって指定し、その ROM 出力ビット、及びアドレスを表示させる様にする。
- フロートポートのデバグを行なう為、ステップキーを押す事によって、1アドレス / ステップで命令を実行させ、命令を実行したアドレス、及び ROM 出力ビットを表示させる。

以上の内容を實現させる回路を構成する事によって、調整作業が能率的になり、且つ、LSI化する際の ROM コード確認が非常に容易かつ確実なものとなる。

(4) コンピュータ・シミュレーション

卓上に使用されるコンピュータ・シミュレーションには、演算、レジスタ交換、判定等の命令の内容及び、レジスタ規模を記憶させ、レジスタ・データ、ポートの入力させる事において、ポートの良否確認を行なわせるシステム・シミュレーションと、論理回路機能を記憶させ（接続情報）その入力端子に信号を入力させる事において機能確認を行なわせるロジック・シミュレーションの2種がある。ハードウェアの機能が確認された期待した命令が、實現可能となる場合において、ポートのデバグをする為

システム・シミュレーションを行なう事は有効であるが、演算機能を確認する為の演算例は数多くあり、これ等を全て、シミュレーションのみで確認する事は、電算機使用時間、確認作業工数の点からみて、実際的ではない。さらに、ロジック・シミュレーションのみを行なう事は、システム・シミュレーションの場合以上に非現実的であり、結局、モックアップを制作した方が手取り早い。モックアップを制作して、回路、演算は機種の確認を終了したのであれば、コンピュータ・シミュレーションを行なう必要は無いのであるが、ランダム・ビット・パターンやテストパターンを作成する場合には、ロジック・シミュレーションを行なう事によって、パターンを発生させる必要がある。又、検査率向上の為に特に設定されたテスト端子の機能を確認する事の為に、回路情報を作成し、ロジック・シミュレーションを実施する必要が生ずる。ロジック・シミュレーションによって、回路の欠陥を捉える事は、前述した様に難しく、テストパターンを作成していく過程で、徐々に確認をしていくには良い。そこで、卓電機能確認の為のシミュレーションは、行なう必要は無いと結論し、UPD940Cでは行なっていない。

但し、テストパターン作成用として、ロジック・シミュレーションは、不可欠、且つ有効な手段である事は、言うまでもない。

5) 検査用テストパターン

ここ3、4年の卓電用MOS LSIの急速な進歩によって、卓電の入出力、演算、制御記憶の回路機能、さらには、周辺表示デバイスのドライバまでを含め、卓電の全機能を、1つのLSIチップ内に集積できるようになった。必然的にLSIに集積されるトランジスタの数は2000~1万個(大容量ROMを持つものは1万個以上)と膨張し、これらのトランジスタ全てを接続する配線部及びトランジスタそのものの良否を判定するテストパターンの作成、あるいは、その作成に要する工数を削減し、且つ、故障検出率の高いテストパターンを得られる様にする回路設計(有効なテスト端子の設定及びシステム全体を考慮した論理回路設計)が、1チップLSIの検査設計する上での重要な1項目となってきていると言っても過言

ではない。

現在 NEC を行なわれているテストパターン形式には SSI 時代より使用されているランダムビットパターン形式と、2年前、8桁 Qメモリ 専用として開発された UPD2710 に初めて採用された実装方式テストパターンの2種がある。この各々の形式の特徴、得失は次の様なものとなる。

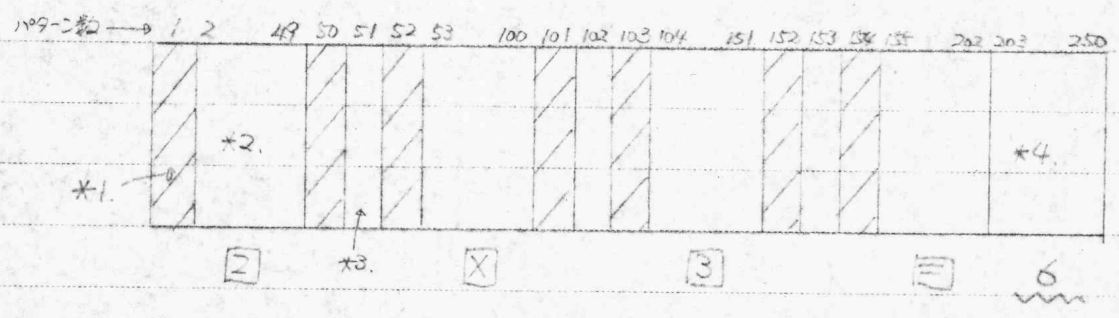
(a) パターン容量

MOS LSI 換専用 IC テスタとして用いられている MH-134C, MH-200 テスタのテストパターン記憶容量は、4096 語 (MH-134C... 1語 48ビット, MH-200... 1語 52ビット) である。使用パターン数は、この値以下に制限される。

(i) 実装方式

実装試験と同様に $2 \times 3 = 6$ の様な演算をさせ、演算結果が "6" となれば良品であると判定する形式のものがある。

キー入力に相当するパターン、演算結果を判定するまでの待S時間を指定 (即ち 同一パターンのくり返しパターン数、及びくり返し回数をパターン内に書き込まれた値により、ミニコンのマイクロ命令ループを制御する) するパターン、演算結果の出力判定をするパターン、以上の種のパターンを記憶させるだけで良く、パターン数は、細かいものとなる。



- *1. □ キーを入力させるには要するくり返しパターン数及びくり返し回数を指定したパターン (1パターン)
- *2. □ キーのキー入力に相当するパターン (この例では、1周期 48ビットタイムとする)
- *3. キーを押しているときのパターン (1パターン)
- *4. 演算結果をみるパターン。このパターンのみ良品判定を行なわせる。

図28. 実装方式テストパターン例 ($2 \times 3 = 6$)

図28. 7示す様に 250 パターン分記憶させれば良い。このまゝの形をランダム・ビット・パターンに置換すると、数万パターンを要し、実現不能である。

(ii) ランダム・ビット・パターン

1クロックサイクル毎に変化するパターンを使用するものである。その昔、前述した様に実際の演算をさせ、しかも、4096パターン以内に納める事は、不能である。実装方式では、最終的に演算が正しくできるかをみよものであるが、ランダム・ビット・パターンでは、個々のゲートに注目し、マクロ命令が正しく実行されているか、ROM出力は正しいか等、より細かい部分の良否を判定する形を取る。この為、後に記述する様に、有効なテスト端子を設定する必要があるが、テスト端子の設定が有効でないか、ゲート数が多い、且つ、ゲート間の関連が薄くマクロ命令の数多くある場合には、4096パターン以内で故障検出率を高くなる事が、できる場合が生ずる。

(b) テスト時間 に対する 故障検出率

(i) 実装方式

$2 \times 3 = 6$ の様な演算をさせると、数万クロックサイクルを必要とし、長い時間を要するが、置数、乗算ルックアップに限定されたROM、2命令しか実行できぬ為、検出率ほぼ高くなる。この為、さらに加減算、除算、ルート計算等のパターンを追加して検出率を高めようとするが、相互のパターン間で、重複する命令(既に故障検出のなされたゲート)を数多く含む事となり、テスト時間が長くなる割に、総合的な検出率は高くなる(1)。

(ii) ランダム・ビット・パターン

ROMアドレス・リスト、ROMトランジスタが正常に動作しているかどうかを検出するROM検査用テストパターン、ROMから出される、リスト交換、演算等の命令が正しく処理されているかどうかをみるランダム・ビット検査用テストパターンの2種を、用途に応じて、ビットタイム、テストタイムを縮小したりして、作成する。この為、短かいテスト時間、故障検出率の高いパターンとする事ができる。

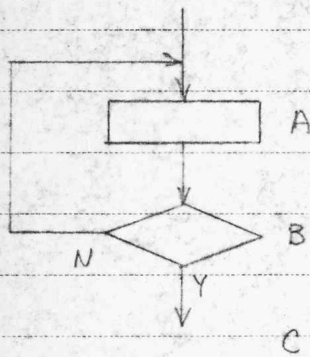


図29.

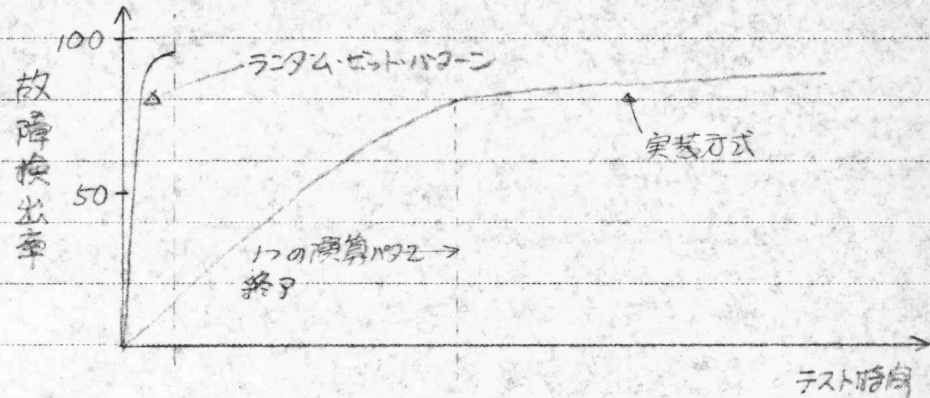


図30. テスト時間対故障検出率概念図

図29の様な回路を挿く72-は卓電72-チャートには随所に出てくるが、
 実装方式では、A→B向を実際に判定所がセットするまで回り続ける。
 7. 17- μ s = 48ビット (例) という時間を短縮する手段を取らなければ、
 この回路を実際に8回回ったとすると、この72-から抜け出すまでに、
 768クロックサイクルを要する。しかも、この回路を使用される
 ゲートの数に換出率があるとしたら、この期間、検出率ゼロという事と
 なり効率の悪いパターンという事になる。ランダムビットパターンでは、
 ROM用パターンで、A→B→A→B→Cとアドレス制御し、さらに、A、Bを
 実行されるマイクロ命令をランダムロジック用パターンで実行、検出すれば、
 長く、短かく、効率の高いパターンとなる。(アドレスステップを1パターン・1ステップ
 とする事であれば、上例は5パターンで済み、さらに効率が増す。)

(C) パターン作成

(i) 実装方式

テストパターンをデバッグするには、最終的には良品サンプルが無いと
 行かないが、テストパターン作成の時期的に差が出る可能性がある。又、
 パターンを紙テープ化する際、マニピュルを行なわねばならぬ等、家内工業的
 な感あり。

(ii) ランダムビットパターン

ロジックシミュレーション結果によって発生させるが、持続情報を知って、パターンの
 デバッグに多くの工数、電算機使用時間を必要とする。

(d) クリアパターン

(i) 実装方式

通常実装状態と全く同じ形式を取れば良い。

(ii) ランダム・ビット・パターン

クリアパターンに7112も1ビットタム=1パターンの割り子、パターンが消費されるので、実装検出率に無関係なクリアパターンを、あまり長いパターンとする事は出来ない。電源をONさせたときLSI内部のレジスタ、等の状態は1/2とどちらかに倒れ、その倒れ方は不定である。コンピュータ・シミュレータの結果より作成したテストパターンは、その初期状態を1, 0と55%かに限定して作られているのでこの初期状態が、デバイスの状態と異なる時デバイス出力とテストパターン予想出力が一致しなくなる場合が生ずる。この為テストパターンを走らせる前に全ての1/0の状態を1又は0にクリア(初期値決定)する必要がある。ランダム・ビット・パターン形式ではこのクリアパターン(Don't care pattern)がテストパターン(care pattern)の前に付随している。

(e) テストパターンに起因する歩留り低下 (ランダム・ビット・パターンに7112)

(i) クリアパターンの不良

クリアパターンにおいて1箇所でも、初期値決定のなされる1/0の存在すると前述した様に良品を不良と判定する場合がある。品種によってはクリアパターンが不良であっても、レイアウト的要素によって1/2の倒れ方がどちらか一方に片寄り長期的に大きな歩留り低下とはならず、クリアパターン不良を発見できる場合がある。

(ii) 論理回路に関連するもの

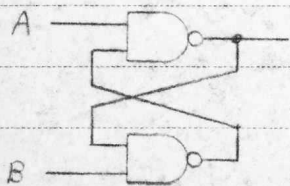


図31.

実装状態ではA, B入力が同時に'L'→'H'になる事は無いとして設計をしても、タイミングを縮小してテストパターンを作ったとき、'L'→'H'の状態が起る場合があり、その出力が不定となり、その出力状態によっては良品を不良とする事となる。A, B入力信号にヒゲが生じる場合も同様である。

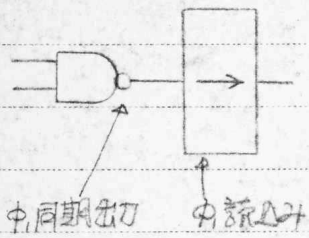


図 32.

左記の様な回路でも実際状態では誤動作せぬとしても、テストパターン作成上、都合が悪い。データ読込みに対して、中、クロック幅のデータ遅れしか許容できない事。さらに、レジスタがレジスタ構成の場合、 $L \rightarrow H$ の遅れが、少くともあれば、データが逃げ、誤動作し、良品であっても、テストパターンが通らぬという不可解な現象を呈する。

以上の様にランダムビットパターン形式の場合には、論理回路設計段階よりテストパターンを考慮し、パターン作成には、相部に与える庄重さを必要とし、設計者、パターン作成者にとって、大きな負担にゐらぬわけである。

[MPD9400 シリーズ品のテストパターン]

主に ROM 以外のランダムゲートの故障を検出する事を目的としたパターン (パターン名 A940) と ROM アドレスレジスタ、デコーダ、メモリ部の故障を検出する事を目的としたパターン (パターン名 B940) の二種作成する。

テスト用入力端子として 2 ピン (CH1, CH2) 用意しており、次の様な用途に用いられる。

- ① 48 ビットの容量を持つデータレジスタを、各の 24 ビットにダイナミックに明確に区別し、各 24 ビット毎に独立の演算をさせる。(A940, CH1)
- ② 演算中であっても、Xレジスタ全桁をセグメント・デコーダを介して出力させ、Yレジスタ全桁をシリアルに、DP 出力端子より出力させる。(A940, CH2)
- ③ アドレス・ステアを 70-チャートに依り、任意に行なわせる為、判定出力を反転させる事もできる。(A940, B940, CH2)
- ④ ビット・タイム・ジェネレータ、ディジットタイム・ジェネレータのクリアをする。(A940, B940, CH1)
- ⑤ ディジット・クロック、ワード・クロック制御を行なう。(B940, CH1)

アドレス・ステアに対して、変化をつける方法として ③ の方法と 70-チャートを使用しない場合の空きアドレスを、アドレス・ジェネレータ用として使用する 2 種の

方法をとっている。

(1) ROM 検査用テストパターン (B948)

ROM 関係に使用されているトランジスタは約 3000 個ある。ROM 出力 19 ビットのうち 35 次アドレス指定出力 9 ビットを除く、ゲート制御出力 10 ビットを全てセグメント、あるいは ティンパット出力端子より並列に出力させ、⑤ のテスト端子操作によって、1~2 パターン = 1 アドレスの割合いるパターンを作成している。この為、700 パターン程度の短かいパターンで、100% 近い検査率をあげることができる。

(2) ランダム・ゲート検査用テストパターン (A948)

前ページ ① にある様に、1ワードタイムを $\frac{1}{2}$ に縮小させる事ができるが、レジスタ内容から、アドレスを処理したデータを選択する際の ティンパット信号即ち $T_1 \rightarrow X$, $T_2 \rightarrow X_c, X_s$, $T_3 \rightarrow X_L$, $T_0 \rightarrow X_H$, $T_1 \rightarrow X_T$ 及び、ワードクロック発生用の T_2 は消去短縮することはできない。従って、 T_3 及びタイムに CHT テスト端子に アドレッシング信号を 入れている事によって $T_4 \sim T_9$ までの数値データ部分を短縮し (ティンパットタイムは $T_1 \rightarrow T_2 \rightarrow T_3 \rightarrow T_0 \rightarrow T_1 \rightarrow T_2$ の如くなる順序で発生する。) 検査率を落とす事なく、パターン数を短縮している。パターン数を短縮した分だけ、故障検査率を高める事ができる。従来、セグメント・データ部の故障については、 $\frac{2}{3}$ 程度しか検査されていなかったが、 $\mu PD948$ では 0~15 の全ての数値をレジスタに入れている事によって、100% の故障検査を可能にしている。

テストパターンでは本質的に検査し得ない回路が 2箇所あり、実装不良の主因となるものと思われる。

- ① 自動クリア論理回路 ... 自動クリアがつかない故障
- ② クロック発生回路 ... 自走発振しない故障

[6] レイアウト設計

UPD940 のレイアウト図の概略は、3層システム構成図とはほぼ同一である。このレイアウト設計におけるパレット・サイズは最終的に決定される。UPD940 は、No.15 フロア設計標準による第1回レイアウト量産品であるが、回路設計例よりかなり異なる要求を製造に出したため、短期間に設計を完了する等、大きな協力を得た。

No.15 フロアでは、ホロニ間隔が従来の 10μm より 7.5μm と半減が出来る為、有効パレット・サイズを小さく出来る事が出来る。以下、その量産例を掲げよう (次ページ)

① マトリクス方式レイアウト

横方向のみ縮小でき、負荷 MOS 2μm² 10mm ピッチの 8.5mm ピッチとなる。負荷 MOS 縦方向は、コンタクト・ホロニとの間隔分のみ小さく出来るが、縮小に大きな留意はしない。

② ROM

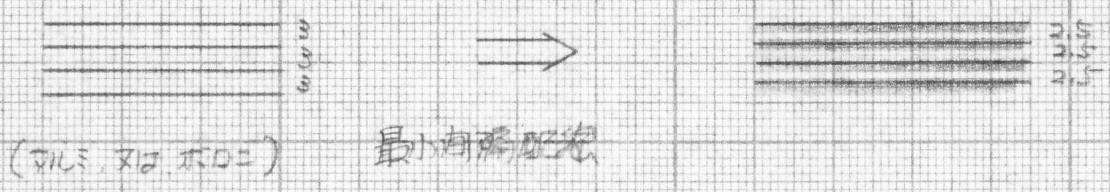
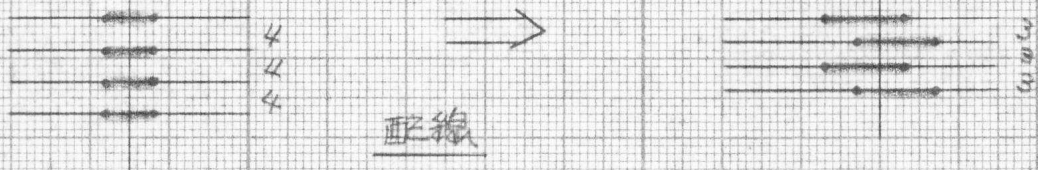
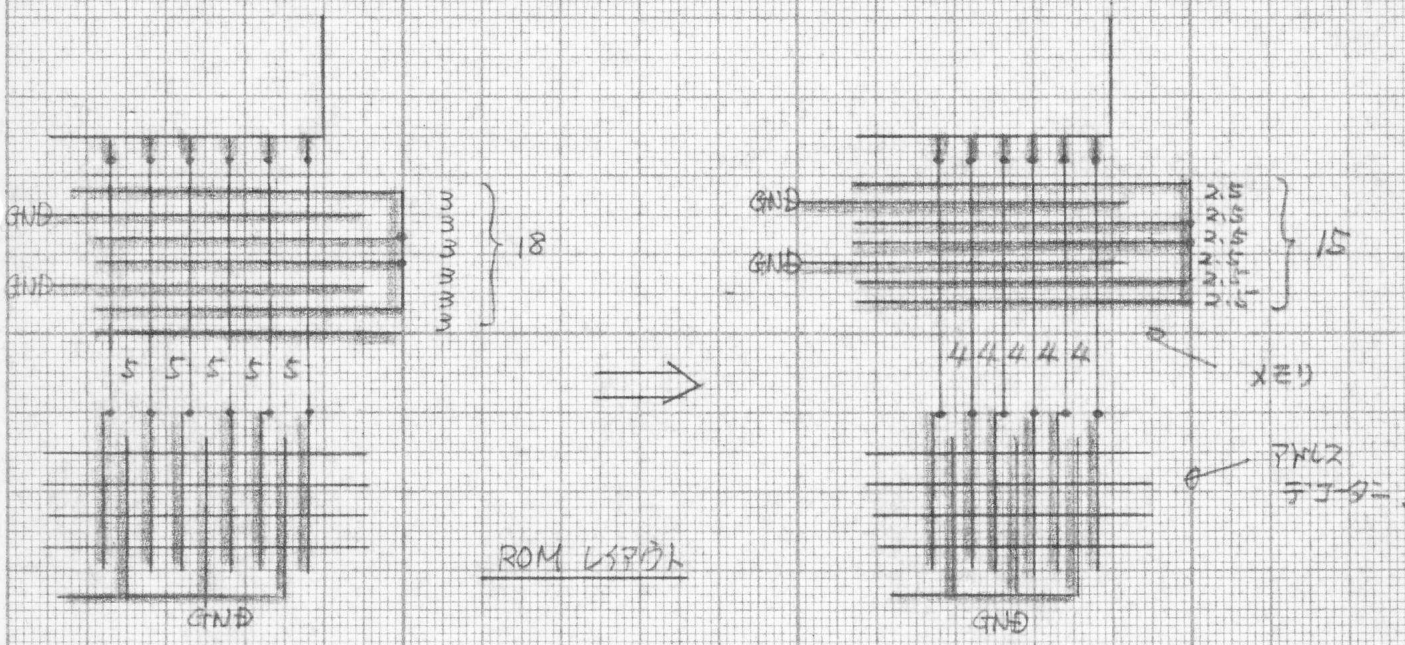
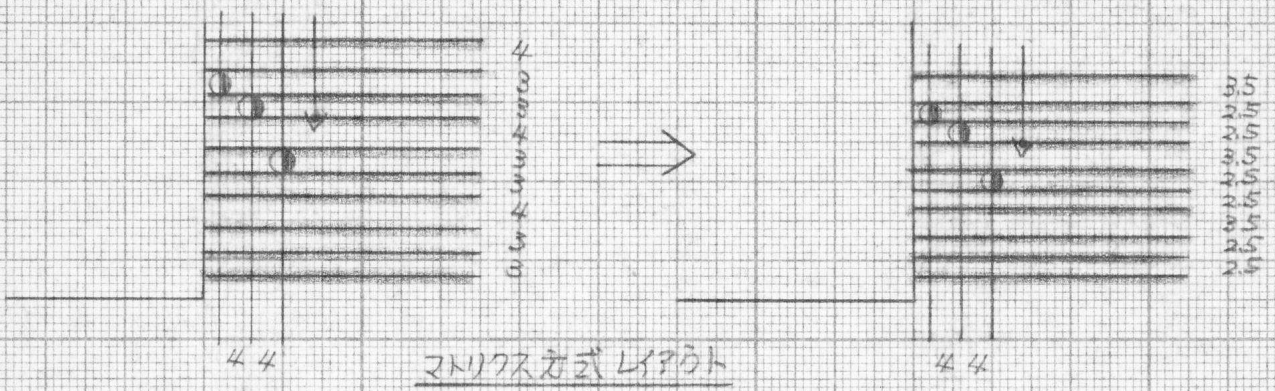
アドレスデコーダとメモリ部との関連もあり、2方向とも縮小出来る。行アドレス出力 64本、列アドレス出力 4本、ROM出力 19ビットの容量を持つ UPD940 の ROM を No.5 レイアウトで設計すると (5×64), (18×19+α)。No.15 になると (4×64), (15×19+α) となり横方向 64mm^{*} 縦方向 57mm^{*} 縮小出来る。No.15 レイアウトを採用した場合、ROM における面積縮小率が他の回路に比して非常に大きく、UPD940 の ROM 集約化という基本設計思想と合致し、パレット・サイズ極小化を可能とした。(××200μm²程度)

③ 配線

一部の限られた箇所のみ使用されるので、全般的にみれば配線部での面積縮小は微々たるものである。

その他、ファンクション・フロアについても横方向のみ縮小される。

No.15 レイアウトにおいて、200倍図面上で「半目盛り」は常に存在し、原図デジタイズの能率が落ちる事が考えられるが、その為にフロアの重ね合わせをしなすという対策をたてても殆んど効果が無い事は明白である。



No.4, 5 LSPOT

No.15. LSPOT

図33. No.4, 5 と No.15 比較

(i) μ PD940C のレイアウト

レイアウト上の留意点について列挙する。

- ① V_{CC} の位置を 'L' レベルよりも 'H' レベルのマージンがゆるいのでマトリクス・レイアウトを用いる事もあり、 V_{DD} よりも GND を優先してアルミで配線する事。
- ② フロック配線については、極力アルミを使用し、ホロンを用いる場合は、抵抗が小さくなる様に配慮する。
- ③ 内部領域から、外部端子への配線に与める面積が大きくなる様にレイアウトする。(パレット外縁部への配線は、内部配線に比し、大きな面積を必要とする。)
- ④ 従来 ROM を使用した品種では、ROM メモリー部に対して図Ⅱのアドレス・レジスタ、デコーダ配線の部分の、ROM 全体に与める面積が大きく、且つ、レイアウト上での「まとまり」がなかった。回路設計段階より、この点を配慮している。
- ⑤ マトリクス方式レイアウトを主とし、ファンクション・ブロックは、7440カ・パス部となる加算器及びその周辺回路に使用する。
- ⑥ エクスパンダ・ゲートを活用し、そのゲートとしては、厚膜トランジスタ・ブロックを使用し、一般回路レイアウトにおいてできた空隙部に配置する。

レイアウト設計は設計の総まとめ的の意味合いが濃く、システム、回路を設計し、それに熟知した者が行う事が全ての点で好ましい。

8桁卓電 シェア EL-8000 (¥10900 宣伝中) に使用されている日立製 LSI TD3623 について比較してみる。

	パレットサイズ	消費電力	デバイス	端子数	演算仕様
HD3623	4.94 × 4.96 _{mm}	約 30mW	Pch. Si ゲート 8	40ピン	可
μ PD940C	3.98 × 3.80 _{mm}	25mW	Pch. Al ゲート 8	28ピン	優

図 34.

	7007名	使用個数	2電源 消費MOS数	1電源 消費MOS数	総消費MOS数	7007名	使用個数	
	NAD1	3	3		6	1倍消費MOS	198	
	" 2	3	3		9	1/2倍 "	10	
組	" 3	2	エズパシキ 使用有 1		7	単 1倍消費MOS	840	
み	" 4	1			5	体 ROMXEN-部	2270	使用トランスジスタ数
合	NOR2	2	2		6	ト 2倍消費MOS	19	5491個
せ	GA2	2	4		10	ラ 3倍 "	2	
フ	" 5	6	6		24	ニ 4倍 "	116	
ン	" 6	2	2		10	ジ 6倍 "	4	
ク	COCG	7	14		49	タ 8倍 "	5	
シ	DBFF	23	エズパシキ 45		137	ナ		
ソ	SDFE	6	18		54	ロ		
ノ	SDFE1	8	16		56	リ		
ト	DYNM	4	4		12	ル		
チ	CLGN	2		4	14			
リ	CGOS	1	11		23			
	CGDE	1	11		24			
	CGDR-S	2		18	32			
	ROMBO	19	38		76			
	ACI	1	2	3	14			
	BUFO1	8	8		24			
	BUFO2	9		9	27			
	BUFI3	2	4		8			
	KEYIB	5	10		25			
	トランス レジスタ	140		280	1584			

レジスタ、レジスタ、7007レジスタ、入出力レジスタ、自動クリア論理回路7007 (ACI) を
除いた、残存する論理ゲート数は、365ゲートである。

図. 35. 使用MOSトランジスタ一覧

〔7〕 今後の展望

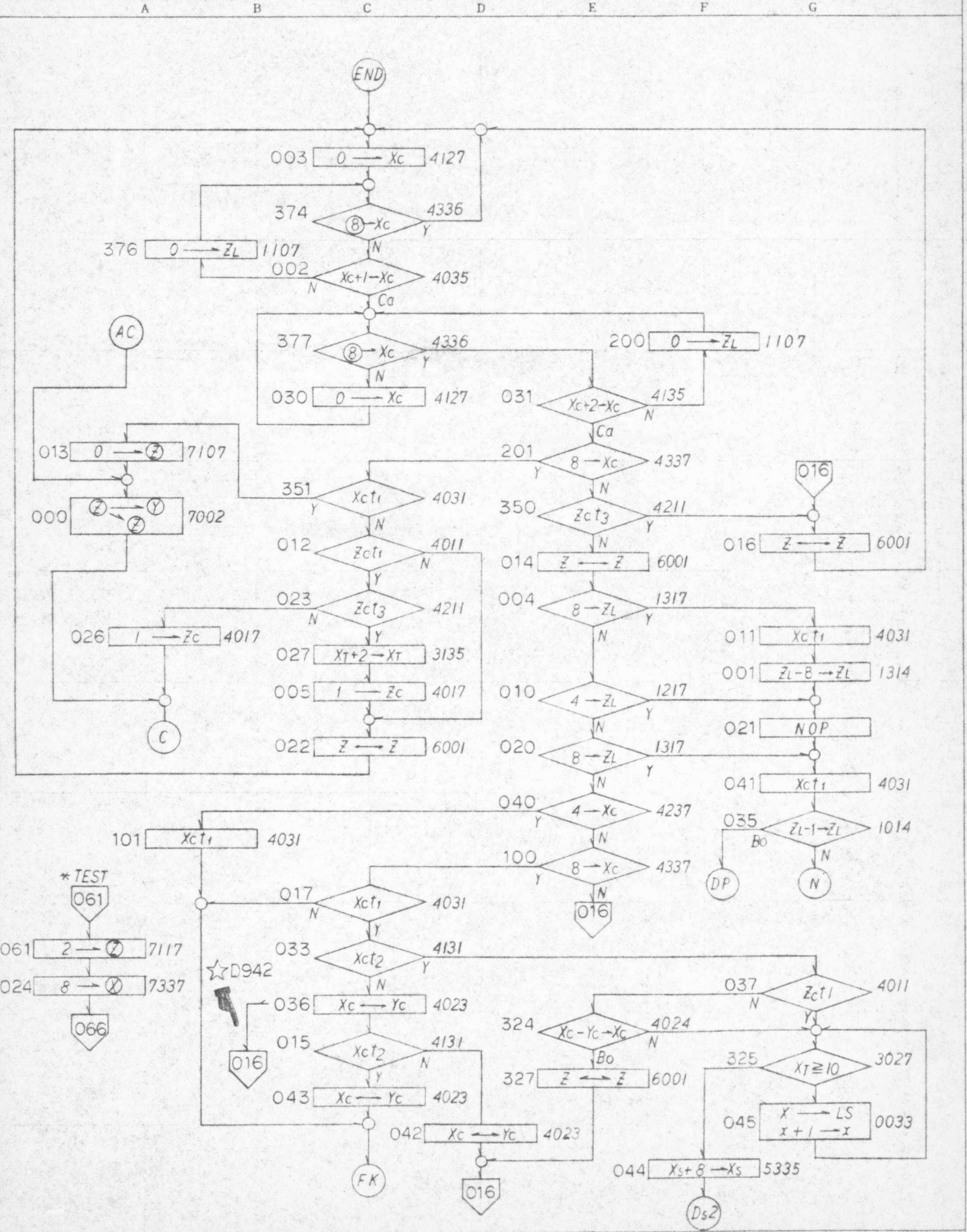
販売価格が 1000円を割っても採算がとれる卓電用LSIを生産できる様になり、卓電設計生産技術も飽和点に近づいた感があり、今後は守値を維持しつつ、より高機能を演算仕様を求める方向に進む以外にない。 従来、生産者主義、シェア拡大主義を押し通して来たが今後、不況が深刻化し、需要も停滞すると考えられ、この際、知識集約化をはかり、ポスト卓電、マイクロコンピュータ等について、思考する時期に来ているものと思われる。

今後の卓電の進む方向について列挙してみる。

- ① 表示フォーマットの向上、又は高耐圧化
- ② 演算時間短縮 (スイッチング向上、メモリサイクル可変、ROM-RAM化)
- ③ 外付 R、C、D 減少化
- ④ F/D 化、CMOS 化

以上の内容について日立(株)のLSIは既に手掛けており、その点では、NECは数歩後退していると言える。多少リスクはあるが、新味あるものを思考していく必要がある。

タイトル	年	月	日	版	承認	査閲	担当	年	月	日	版	承認	査閲	担当	登録番号
UPD 940 C 941 C 942 C NO.1	49.	6.	20	1											参照番号
	49.	6.	25	2											作成者
	49.	7.	6	3											
	49.	7.	15	4											

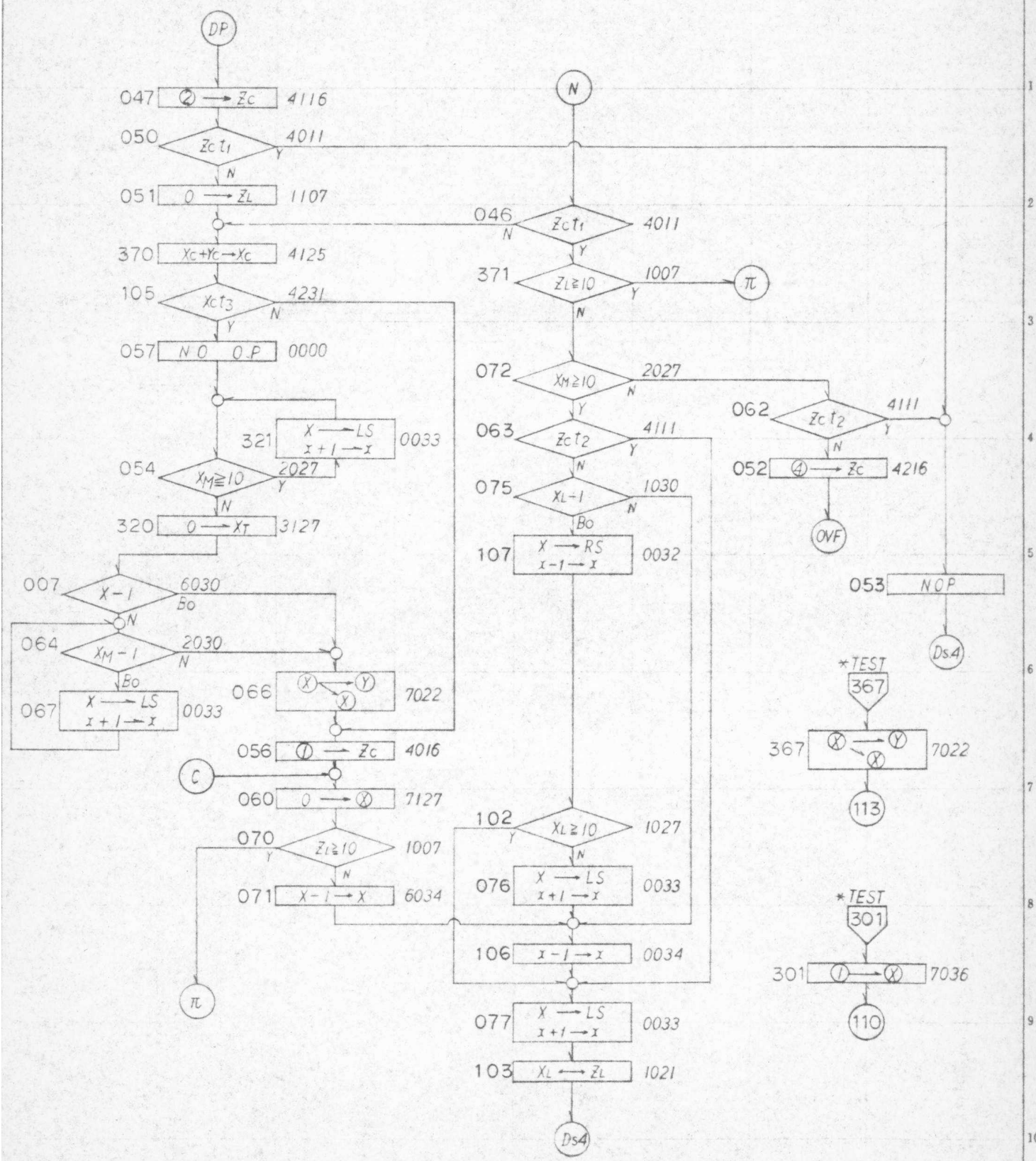


フローチャート

タイトル	年	月	日	版	承認	査閲	担当	年	月	日	版	承認	査閲	担当	登録番号
UPD 940 C															
															参照番号
															作成者

NO.2

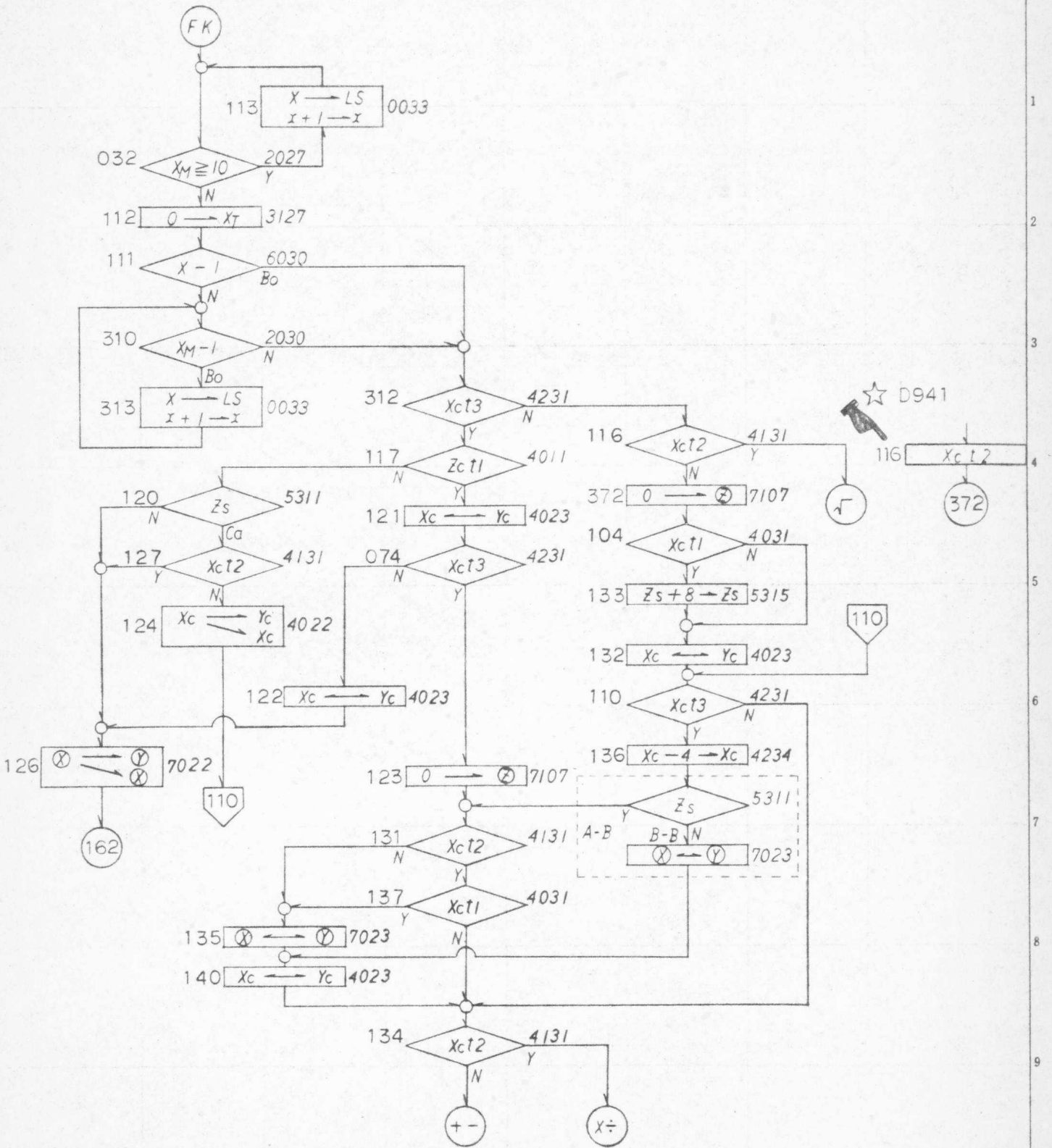
A B C D E F G



フローチャート

イトル UPD 940 C NO.3	年	月	日	版	承認	査閲	担当	年	月	日	版	承認	査閲	担当	登録番号
															参照番号
															作成者

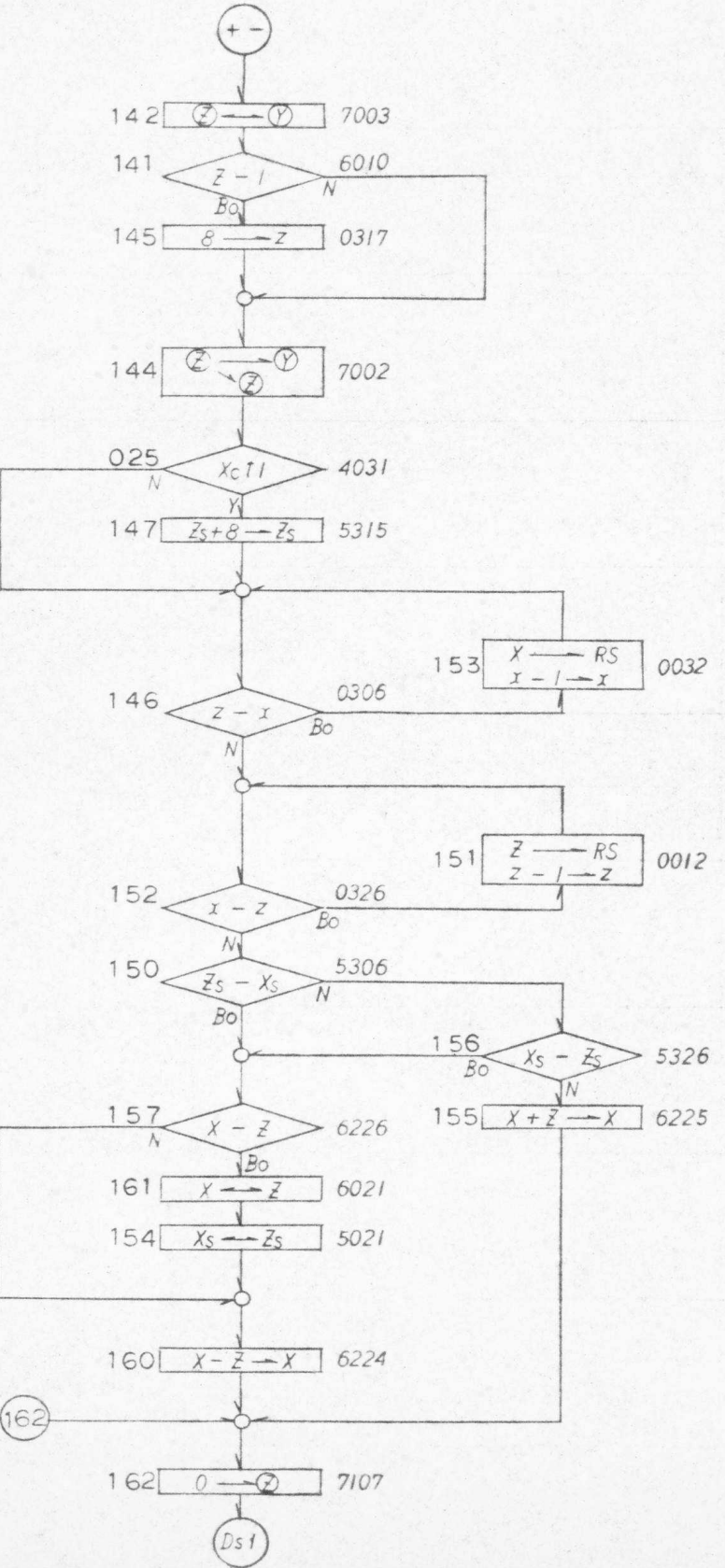
A B C D E F G



1
2
3
4
5
6
7
8
9
10

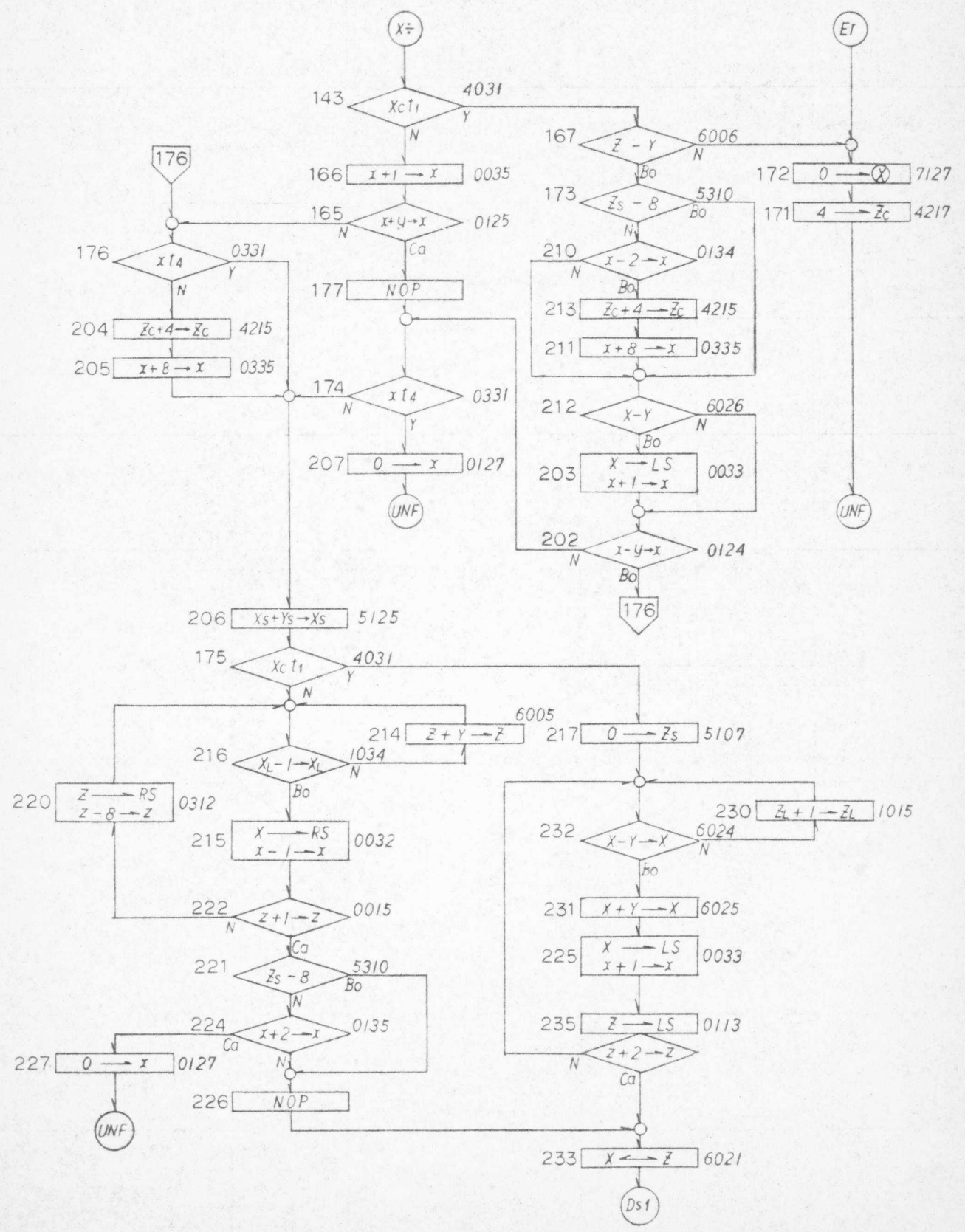
フローチャート

タイトル	年	月	日	版	承認	査閲	担当	年	月	日	版	承認	査閲	担当	登録番号
UPD 940 C															
															参照番号
															作成者



フローチャート

イトル UPD 940 C NO.5	年	月	日	版	承認	査閲	担当	年	月	日	版	承認	査閲	担当	登録番号
															参照番号
															作成者
A	B	C	D	E	F	G									

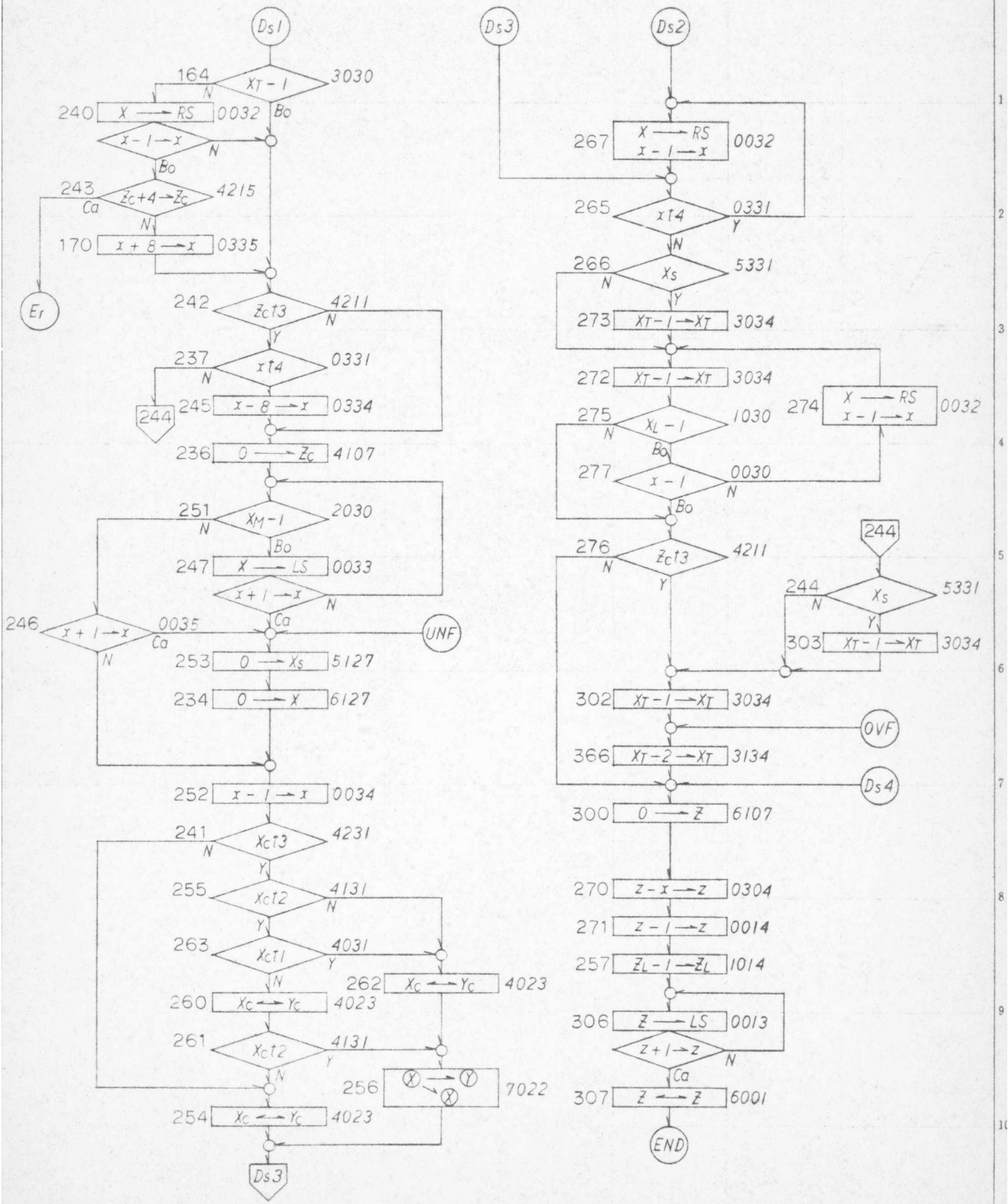


フローチャート

タイトル	年 月 日 版	承認 査閲 担当	年 月 日 版	承認 査閲 担当	登録番号
UPD 940 C					
					参照番号
					作成者

NO.6

A B C D E F G



フローチャート

イトル

UPD 940 C

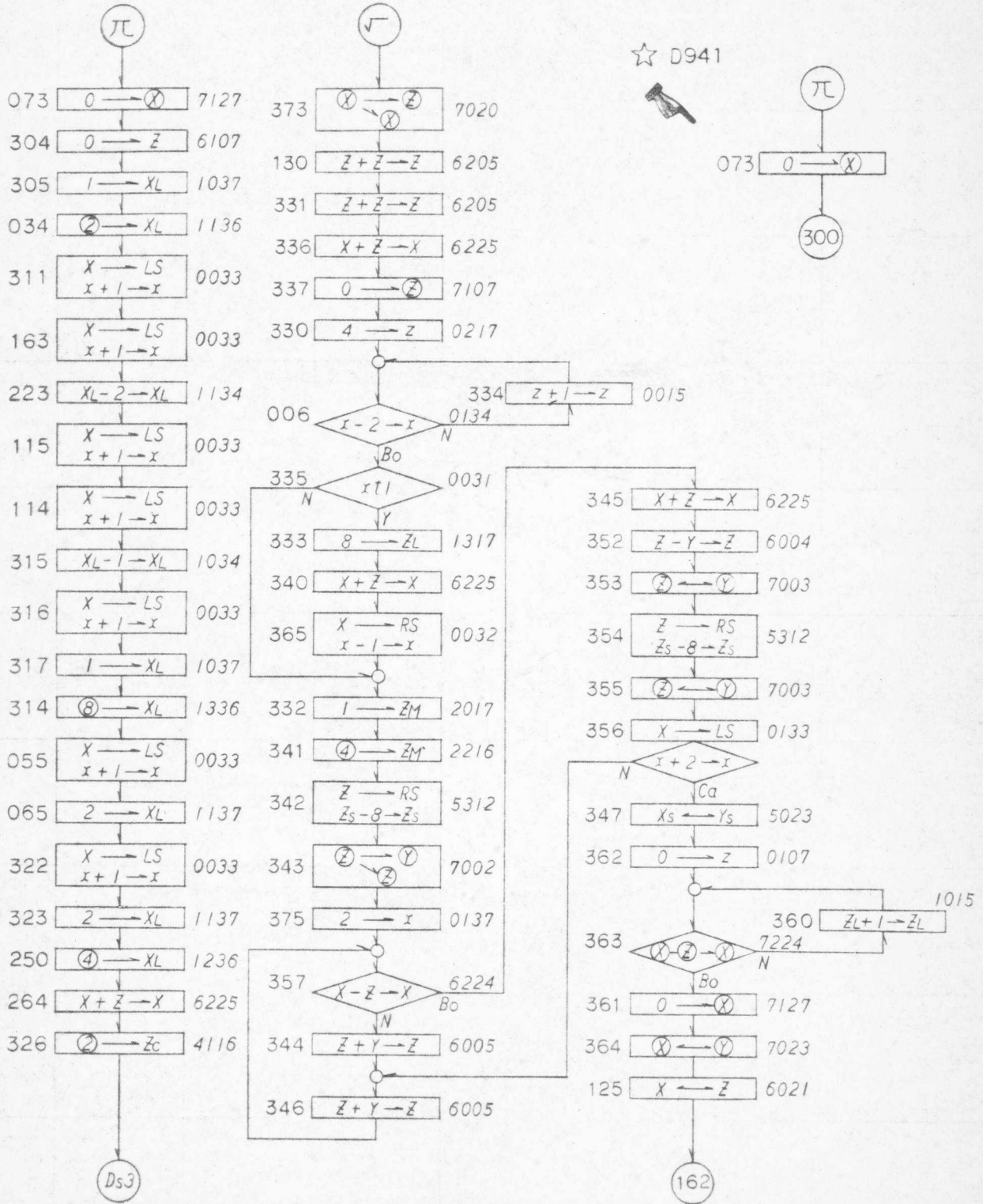
NO.7

年	月	日	版	承認	査閲	担当	年	月	日	版	承認	査閲	担当	登録番号
---	---	---	---	----	----	----	---	---	---	---	----	----	----	------

参照番号

作成者

A	B	C	D	E	F	G
---	---	---	---	---	---	---



					AD1		AD2		Zir		Yir		Zor		Xor																	
5	4	3	2	1	7 6	7 6	7 6	7 6	X1a	X0a	code	code	ZI	YI	X1a	ZI	X1a	YI	Zo	E	Xo	E										
					$\bar{z} \rightarrow \bar{z}$				○	X	X	X	X	X	○	X	X	○	X	○	○	X	X									
					$x \rightarrow \bar{z}$				○	X	X	X	X	X	○	X	X	○	○	X	X	○	○									
					$z \leftrightarrow z$				X	X	X	X	○	X	○	X	X	○	X	○	○	○	X	X								
					$x \leftrightarrow z$				X	X	X	X	○	X	○	X	X	○	○	X	X	○	○									
					$\bar{z} \rightarrow Y$				○	X	X	X	X	X	X	○	○	X	X	○	○	○	X	X								
					$x \rightarrow Y$				○	X	X	X	X	X	X	○	○	X	○	X	X	○	○									
					$z \leftrightarrow Y$				X	X	X	X	X	○	X	○	○	X	X	○	○	○	X	X								
					$x \leftrightarrow Y$				X	X	X	X	X	○	X	○	○	X	○	X	X	○	○									
					$\bar{z}-Y \rightarrow \bar{z}$	$\bar{z}-Y \rightarrow \bar{z}$	$\bar{z}-X \rightarrow \bar{z}$	$\bar{z}-X \rightarrow \bar{z}$	○	X	X	X	X	○	X	○	X	○	X	○	○	○	○	X	X							
					$x-Y \rightarrow x$	$x-Y \rightarrow x$	$x-Z \rightarrow x$	$x-Z \rightarrow x$	○	X	X	X	X	○	X	○	X	○	○	X	X	○	○	○	X	X						
					$\bar{z}+Y \rightarrow \bar{z}$	$\bar{z}+Y \rightarrow \bar{z}$	$\bar{z}+Z \rightarrow \bar{z}$	$\bar{z}+Z \rightarrow \bar{z}$	○	X	X	X	X	○	X	○	X	○	X	○	○	○	○	○	X	X						
					$x+Y \rightarrow x$	$x+Y \rightarrow x$	$x+Z \rightarrow x$	$x+Z \rightarrow x$	○	X	X	X	X	○	X	○	X	○	○	○	X	X	○	○	○	X	X					
					$z-Y$	$z-Y$	$z-x$	$z-x$	○	X	X	X	X	○	X	○	X	○	○	○	X	○	○	○	X	X						
					$x-Y$	$x-Y$	$x-Z$	$x-Z$	○	X	X	X	X	○	X	○	X	○	○	○	X	○	○	○	X	X						
					$z \geq 10$	$0 \rightarrow z$	$z \geq 10$	$0 \rightarrow z$	○	X	X	X	X	X	X	○	X	○	X	○	○	○	○	○	○	X	X					
					$x \geq 10$	$0 \rightarrow x$	$x \geq 10$	$0 \rightarrow x$	○	X	X	X	X	X	X	○	X	○	○	○	X	X	○	○	○	○	X	X				
					$z-C$	$C=1$	$C=2$	$C=4$	○	X	X	○	X	X	X	○	X	○	○	○	X	○	○	○	○	○	X	X				
					$x-C$				○	X	X	○	X	X	X	○	X	○	○	○	X	○	○	○	○	○	○	X	X			
					$z+C$				○	X	X	○	X	X	X	○	X	○	○	○	X	○	○	○	○	○	○	X	X			
					$x+C$				○	X	X	○	X	X	X	○	X	○	○	○	X	○	○	○	○	○	○	○	X	X		
					$\bar{z} \rightarrow RS, z-C \rightarrow \bar{z}$				$\bar{T}_3 \sim \bar{T}_1$	X	X	○	X	X	X	○	X	○	○	○	X	○	○	○	○	○	○	○	X	X		
					$x \rightarrow RS, x-C \rightarrow x$					X	X	○	X	X	X	○	X	○	○	○	X	○	○	○	○	○	○	○	○	X	X	
					$z \rightarrow LS, z+C \rightarrow \bar{z}$				$T_2 \sim T_1$	X	○	○	X	X	X	○	X	○	○	X	○	○	○	○	○	○	○	○	○	X	X	
					$x \rightarrow LS, x+C \rightarrow x$					X	○	○	X	X	X	○	X	○	○	○	X	○	○	○	○	○	○	○	○	X	X	
					$z-C \rightarrow z$				○	X	X	○	X	X	X	○	X	○	○	X	○	○	○	○	○	○	○	○	○	X	X	
					$x-C \rightarrow x$				○	X	X	○	X	X	X	○	X	○	○	○	X	○	○	○	○	○	○	○	○	X	X	
					$z+C \rightarrow \bar{z}$				○	X	X	○	X	X	X	○	X	○	○	X	○	○	○	○	○	○	○	○	○	X	X	
					$x+C \rightarrow x$				○	X	X	○	X	X	X	○	X	○	○	○	X	○	○	○	○	○	○	○	○	X	X	
					$0 \rightarrow \bar{z}$				○	X	○	X	X	X	X	○	X	○	○	X	○	○	○	○	○	○	○	○	○	○	X	X
					$0 \rightarrow x$				○	X	○	X	X	X	X	○	X	○	○	○	X	○	○	○	○	○	○	○	○	○	X	X
					$C \rightarrow \bar{z}$				X	X	○	X	X	X	X	○	X	○	○	X	○	○	○	○	○	○	○	○	○	○	X	X
					$C \rightarrow x$	↓	↓	↓	↓	X	X	○	X	X	X	○	X	○	○	○	X	○	○	○	○	○	○	○	○	○	X	X

○ ... 0, ○ ... 1, E6=1

E7=1

49.4.27,

10	9	8		1	2	3	4	←→	9	10	11	12	
			0										X
			1										X _L
			2										X _m
			3										X _T
			4										X _{C₄}
			5										X _S
			6										X
			7										(X)

7	6	
		1 → code
		2 → code
		4 → code
		8 → code

- ◎ 234 のとき Key code を変更
- ◎ Key 信号は表示中のみに出る

- ⑩ ... 3, 4, 6
- ㊦ ... 1

