高機能ディスプレイ端末のためのグラフィック・プロセッサの開発およびその評価†

小林芳樹†† 高藤政雄††
浜田長晴†† 平沢宏太郎††
川本幸雄†† 小中清司††† 桑原 洋†††

グラフィック・ディスプレイ端末は，近年の LSI 技術を背景として発展し，広く用いられるようになってきた．我々は，端末の機能の高度化(インテリジェント化)，使いやすさおよび高速応答性を追求したカラーグラフィック・ディスプレイを開発し，その評価を行った．

カラーグラフィックとしてラスタ走査形 CRT が広く用いられている．このタイプのグラフィック・ディスプレイの問題の1つに，大容量の画像メモリが必要なことがあったが近年の 16 k, 64 k ビットメモリの出現により，それほど大きな問題ではなくなってきている．もう1つの問題として，図形表示のため大量の絵素を発生しなければならないことがあるが，これに対して我々はバイポーラのビットスライス LSI を用いた図形処理専用のグラフィック・プロセッサを開発し，高速化を図ることにした．

本稿では，まず開発したシステムの構成および特徴的な仮想画面，図形操作機能等を紹介し，次にグラフィック・プロセッサの構成とその評価について述べる．本プロセッサは，その図式処理をすべてマイクロプログラムで実行するものであるが，特に直線のドット展開処理の高速化に適した構成をとった．これを評価した結果，通常のマイクロプログラム制御に対し 1.5〜1.7 倍の直線描画速度を達成しており，効果の大きいことが明らかとなった．本稿では，さらに高速化を図るための構成についても考察を加えている．

1. まえがき

グラフィック・ディスプレイ端末は，近年の LSI 技術を背景として発展し，広く用いられるようになってきた．我々は，端末の技術動向である端末の機能の高度化(インテリジェント化)，使いやすさおよび高速応答を追求したカラーグラフィック・ディスプレイを開発し，その評価を行った．

カラーグラフィックとしてラスタ走査形 CRT が広く用いられている．その図形表示方式には，Run-length Encoding 法[1]，Cell Organized Encoding 法[2),3)] Real-time Scan Conversion 法[4),5)] 等もあるが，CRT 画面の絵素に対応してその色情報を記憶する画像メモリを持つ方式が一般的であり，我々も本方式を採用した．本方式の問題の1つに大容量のメモリが必要なことがあったが，近年の 16 K, 64 K ビットメモリの出現によりそれほど大きな問題ではなくなってきている．もう1つの問題として，図形表示のために大量の絵素を発生しなければならないことがあるが，これに対して我々はバイポーラのビットスライス LSI を用いた図形描画処理専用プロセッサを開発し，高速化を図ることにした．

本稿では，まず開発したシステムの構成および特徴的な仮想画面，図形操作機能等を紹介し，次にグラフィック・プロセッサの構成とその評価について述べる．グラフィック・プロセッサはすべてマイクロプログラム制御としたものであるが，特に直線のドット展開処理の高速化に適した構成をとった．本稿では基本図形である直線の描画処理の分析および評価を行い，さらに高速化を図るための構成について考察を加えた．なお，直線，円等の表示アルゴリズムを扱った文献[6)〜9)]，マイクロプログラム制御による直線表示を扱った文献[10)]はあるが，本稿のように高機能グラフィックにおけるグラフィック・プロセッサを全般的に分析し評価した文献は少ない．

2. システムの概要

2.1 システム構成

† Development and Evaluation of a Graphic Processor for an Intelligent Display Terminal by YOSHIKI KOBAYASHI, MASAO TAKATOO, NAGAHARU HAMADA, KOOTARO HIRASAWA, YUKIO KAWAMOTO (Hitachi Research Laboratory, Hitachi Ltd.), KIYOSHI KONAKA and HIROSHI KUWAHARA (Omika Works, Hitachi Ltd.).

†† (株)日立製作所日立研究所
††† (株)日立製作所大みか工場

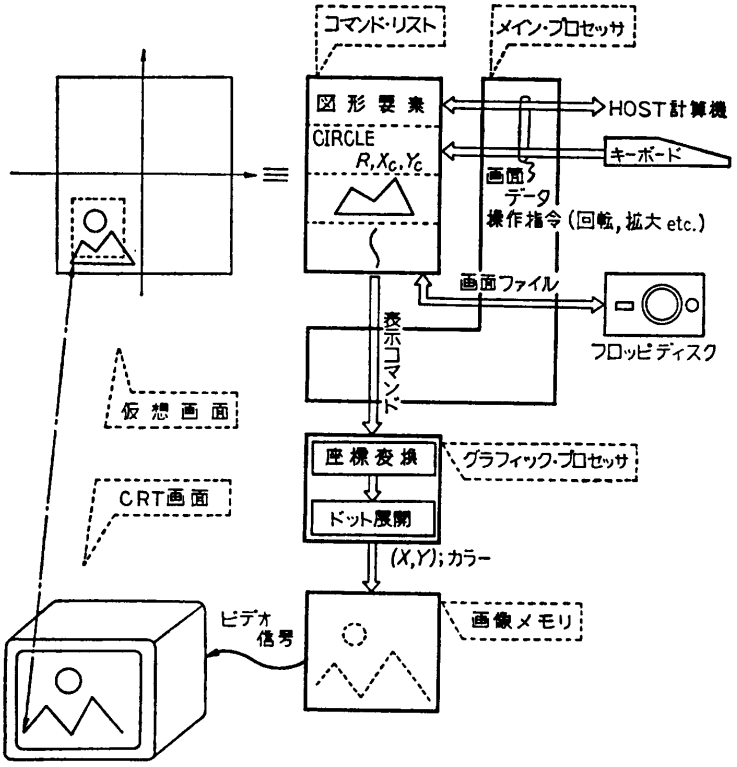


図 1 グラフィック・ディスプレイ端末のシステム構成
Fig. 1 Block diagram of a graphic display terminal.

開発したシステムの構成は，**図1**に示されるように次の4つの機能ブロックからなる．

（1） グラフィック・コマンドリスト……直線，円，円弧および面等のコマンドを記憶するメモリ．

（2） メインプロセッサ……ホスト計算機との通信，フロッピ・ディスクのファイル管理，コマンドリストの管理等のシステム全体を管理・制御する汎用マイクロプロセッサ．

（3） グラフィック・プロセッサ……メインプロセッサから与えられる図形コマンドを解釈し，画像メモリに図形を生成するための図形描画専用プロセッサ．

（4） 画像メモリ……CRT 表示画面の最小単位である絵素に対応した色情報を記憶する CRT 画面リフレッシュ用メモリ．

2.2 システムの特徴

本システムは，インテリジェント化，使いやすさ，高速応答を目指したもので，特にプロセス制御用としてコスト・パフォーマンスを追求したものである．本システムの中核となるものがコマンドリストである．コマンドリストの図形は，処理単位（これを図形要素と呼ぶ）で管理されており，これにより次の機能を端末で実現することができる．

（1） 仮想画面機能……物理的な CRT 画面の大きさにとらわれず，±32 K×±32 K の絵素平面に図形を描くことができ，これを任意の倍率で CRT 画面に投影できる．**図2**にこの例を示す．

（2） 図形管理および操作機能……図形要素単位に表示，消去の指示が行え，また移動，拡大，回転等の図形操作を行うことができる．

（3） サブルーチン機能……繰返して使われる図形は，サブルーチンとして登録でき，図形要素の中から表示位置，倍率を指定して呼び出すことができ，コマンドリストの容量を削減できる．

以上のように，従来ホスト計算機で行っていた機能を端末で行うことができ，ホスト計算機の負荷，通信容量を削減できる．

さらに，使いやすさを向上するため，

（1） ユーザプログラム拡張機能……端末の DTOS (Display Terminal Operating System) のグラフィック・マクロを用いることにより，ユーザ特有の機能を容易にプログラムし組込めるソフトウェア構成をとっている．

（2） 端末作画機能……端末の TIPS (Terminal Interactive Picture Generation System) により，VECTOR, CIRCLE, TEXT 等のコマンドにより画

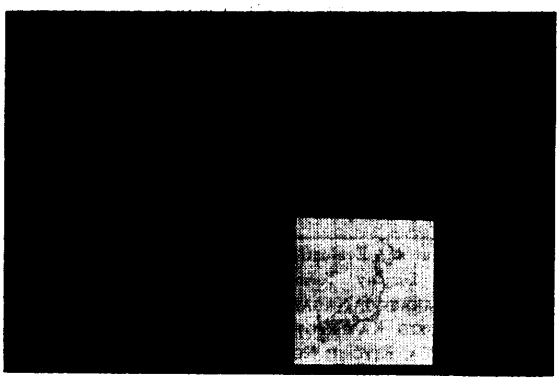


図 2 世界地図と日本の拡大表示
Fig. 2 World-wide map and scaling-up of Japan.

面を作成でき，これをフロッピ・ディスクに画面ファイルとして登録できる．
のようなユーザサポート機能の充実を図っている．

さて応答性については，プロセス制御用途では画面の切換え時間として1～2秒以内が望まれている．従来のセミグラフィック・ディスプレイを用いた場合，プロセス制御分野での一般的な表示画面の線数は（CRT画面を700×500ドットの絵素構成とする），100ドット長の直線にして高々2,000本程度であった．グラフィック・ディスプレイとしての機能向上による表示線数の増大を3倍とみると6,000本／秒の直線描画処理能力が必要となる．マイクロコンピュータのプログラムだけで直線のドット展開処理を行う場合は，2桁近く処理能力が低くなる．一方，DDA (Digital Differencial Analizer) ハードウェアによる直線発生回路では，直線の描画処理能力は十分であるが，円，面等の図形を描画する場合にはマイクロコンピュータのプログラムが介在するために処理時間がかかる．このため我々は，画像メモリの書込応答時間に追随する直線のドット展開処理能力（4.3節に示すように3ステップ，600n秒／ドット）を持つ，マイクロプログラム制御の図形処理専用グラフィック・プロセッサを開発した．画像メモリには16Kビットの RAM を用いており，CRT 画面のリフレッシュのための読み出しと直線のドット書き込みとを時分割に制御している．この時間は CRT 表示タイミングによるが1μ秒前後であり，1μ秒のとき本システムは7,000本／秒（100ドット／本）の直線描画処理能力を持つ．また，グラフィック・プロセッサとシステムを制御するメインプロセッサとの間を FIFO (First-In-First-Out) バッファを介してパイプライン処理することにより，コマンド処理のスループットの向上を図っている．

なお本システムは，プロセス制御用途に専用にしたものではなく，機能的にも処理速度の点でもそのほかの経営情報分野や CAD/CAM (Computer Aided Design/Manufacturing) 分野等の広い分野に適用できるものと考えている．このため画像メモリの構成も各種の絵素構成の CRT に対処できるように，512×512，1,024×512，1,024×1,024 の各種構成をとることができ，絵素のカラー情報も赤，緑，青の各1ビットを基本とするが，最大4ビットまで拡張可能な構成にしている．

3. グラフィック・プロセッサの構成

グラフィックプロセッサは，サイクル時間200n秒のマイクロプログラム制御のプロセッサで，そのデータ処理部は**図3**に示すように，

（1） メインプロセッサ・アダプタ……メインプロセッサからの図形コマンドを受ける部分で FIFO バッファを用いる．これにより両プロセッサが互いに非同期にパイプライン処理を行え，高速化が図られる．

（2） 論理演算部……バイポーラのビットスライス LSI を用いて16ビットの論理演算部を構成している．ここには16語のレジスタファイルが内蔵される．

（3） データメモリ部……それぞれ4k語まで拡張できる RAM, ROM からなり，

RAM……描画制御情報，座標変換パラメータ等の各種制御変数およびユーザプログラマブルな文字フォント，面パターン

ROM……数値定数，SIN 値テーブル，直線制御テーブル，文字フォント，面パターン

が格納される．アクセスには，マイクロプログラムからの直接アクセスと，テーブル参照のためのアドレスカウンタによるアクセスの2つのモードがある．

（4） 画像メモリアダプタ……(x, y) レジスタ，色情報レジスタ等を有し，画像メモリへの書込処理とグラフィック・プロセッサにおける次の (x, y) 算出処理とをパイプライン的に処理するためのパイプラインレジスタを持つ．画像メモリに対しては，1絵素（ドット）単位の書込／読出のほか，面の描画の高速化のため16ドット（ブロック）単位の書込／読出が可能である．

から構成される．

一方，マイクロプログラムの制御部は，1語32ビットで最大32k語まで拡張可能な構成になっている．マイクロ命令は，**図4**に示すように，

① マイクロプログラムのシーケンスは，ビット0～2で制御される．このフィールドで，各種のジャンプ (INSTJ, B, BC)，サブルーチンコール (BAL)，カウンタによるループのためのリピート (RPT) 等が指定できる．条件ジャンプ (BC) 時の条件の選択は TSEL で，またジャンプアドレスは JUMP ADDR で示される．

② 16ビット論理演算部は，その演算動作を指定する ALUI，シフト入力を指定する SFTC および16語の2ポードレジスタファイルのアドレスを指定する

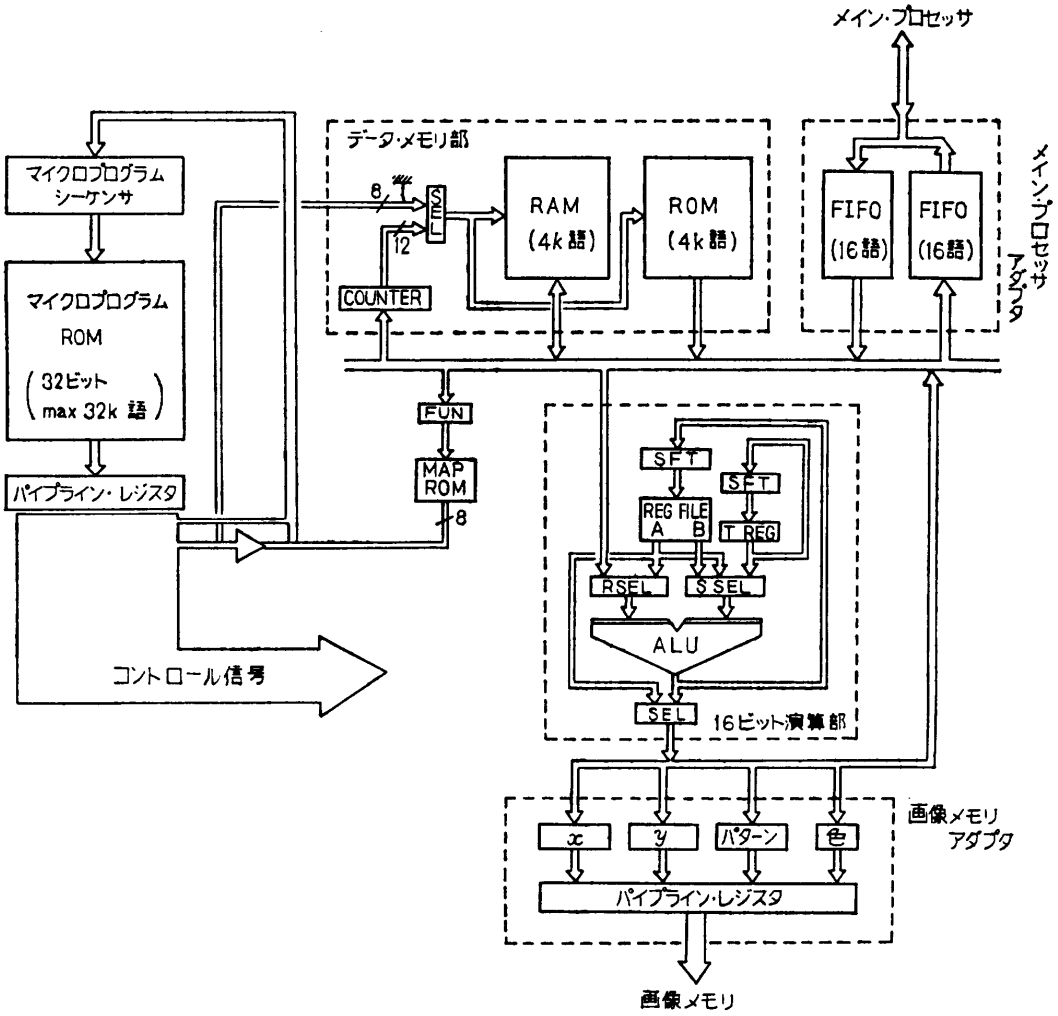


図 3 グラフィック・プロセッサのハードウェア構成

Fig. 3 Haadware structure of a graphic processor.

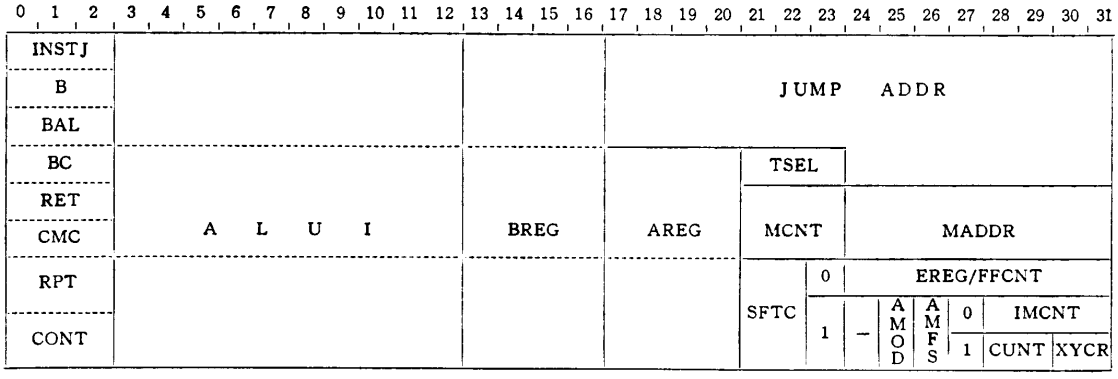


図 4 マイクロプログラム命令フォーマット

Fig. 4 Microprogram instruction format.

AREG, BREG により制御される.

③ データメモリ部は MCNT で制御され, MADR フィールドによるアクセスあるいはカウンタによるアクセスが可能である.

④ カウンタは CUNT によりロード, カウントアップおよびダウンが制御される.

⑤ 画像メモリアダプタ部は, XYCR により (x, y) レジスタへの座標値セットが制御され, IMCNT により画像メモリに対する書込/読出が制御される.

⑥ 4.3 節で述べるが, 直線のドット展開処理の高速化のために, レジスタファイルのAアドレスを特定フラグによりモディファイすることを指定できる A-MOD, そしてその特定フラグに指定の条件ビットの値をセットすることを指定できる AMFS ビットを持つ.

⑦ そのほかの外部レジスタ, フリップフロップは, EREG/FFCNT で制御される.

の各フィールドからなる垂直形マイクロプログラムである. 直線, 円, 円弧, 面, キャラクタ等の基本コマンドは, 座標変換処理やクリッピング処理も含めて約 1.7k 語で実現されている.

本プロセッサの開発に当って, 我々はできる限りハードウェアの簡略化を図るため, 座標変換やクリッピングのための特殊なハードウェア[11),12)]も, 乗, 除算ハードウェア, 直線描画ハードウェアをも設けず, すべてマイクロプログラムで実現する方針をとった. この方針のもとに, 基本図形である直線の描画の高速化のためにとった手段と処理時間の分析, およびその評価について以下の章に述べる.

4. 直線描画処理とその分析

直線の描画処理の概略フローを**図5**に示す. (I)では直線コマンドの線種や書き込み色情報等の各種パラメータのイニシャル処理を行う. (II), (III)では座標 (X, Y) を取込んで座標の絶対, 相対, インデックス相対等の修飾を行う. (IV)で仮想画面から CRT 画面への座標変換を行い, (V)で変換後の直線の CRT 画面外に出た部分を切り取るクリッピング処理を行う. (VI), (VII)は CRT 画面座標の直線を1ドットごとに展開して画像メモリに書き込むドット展開処理部で, (VI)はその前処理にあたる. 直線コマンドでは, 指定直線本数だけ(II)～(VII)を繰返す. 以下の節では特に重要な処理部(IV), (V), (VII)の処理とその分析について述べる.

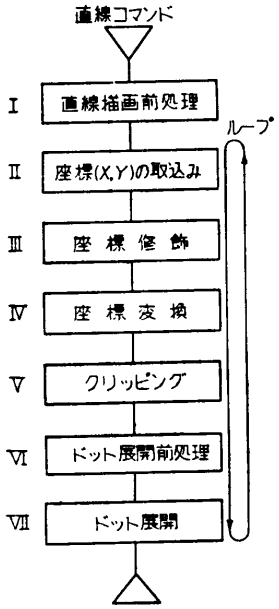


図5 直線描画処理フロー
Fig. 5 Flow chart of line generation.

4.1 座標変換処理

すでに述べたように本端末では, 最大4層までのサブルーチン機能を用いて図形要素を定義でき, この図形要素に対し仮想画面上で拡大, 回転等の操作を行える. これを**図6**に示すが, サブルーチンを含む図形要素は, 図形操作の変換 φ_E により仮想画面に写像され, 仮想画面上の図形は変換 φ_V により CRT 画面の任意の矩形エリア (これをビューポートと呼ぶ) に写像される. したがってコマンドリスト中の座標 (X, Y) は, これが第4層のサブルーチンのコマンドの場合には, 第1層～第4層のサブルーチンの図形変換を各々 $\varphi_{S1}, \varphi_{S2}, \varphi_{S3}, \varphi_{S4}$ として

$$\begin{pmatrix} x \\ y \end{pmatrix} = \varphi_V \cdot \varphi_E \cdot \varphi_{S1} \cdot \varphi_{S2} \cdot \varphi_{S3} \cdot \varphi_{S4} \begin{pmatrix} X \\ Y \end{pmatrix}$$

の座標変換が CRT 画面座標 (x, y) を求めるために必要となる. この変換処理マイクロプログラムで実行する訳であるが, 逐次変換処理を行っていたのでは大変なので

(1) 変換を一括して行えるよう一括変換 φ_T

$$\varphi_T = \varphi_T \cdot \varphi_E \cdot \varphi_{S1} \cdot \varphi_{S2} \cdot \varphi_{S3} \cdot \varphi_{S4}$$

(2) 上記変換 φ_T の状態 STS

A	B	C	D	——

A=0/1……移動: 無/有
B=0/1……拡大: 無/有
C=0/1……回転: 特殊角度/任意角度
D=0～3……回転角 0°, 90°, 180°, 270°

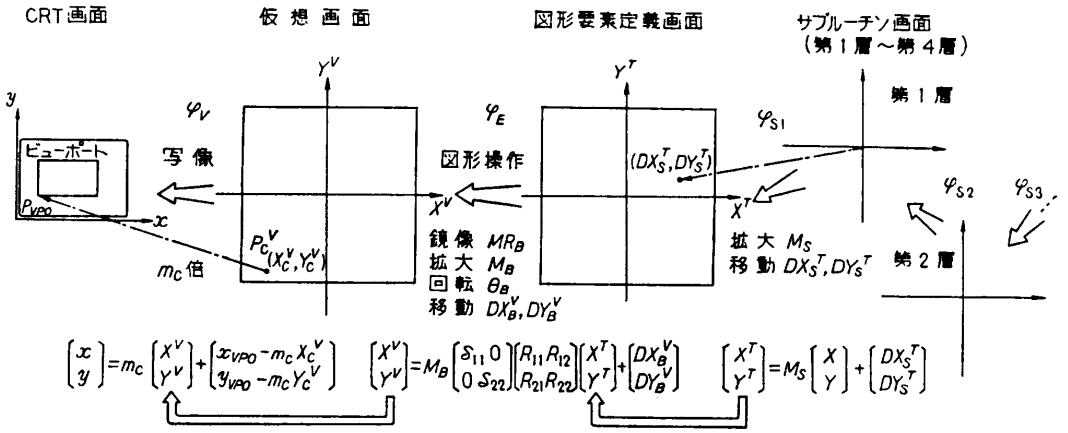


図 6 各画面間における座標変換処理

Fig. 6 Coordinate transformation between the screens.

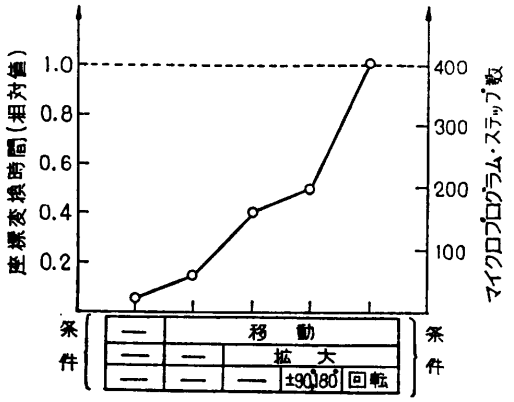


図 7 座標変換処理時間，ステップ数

Fig. 7 Processing time and dynamic steps for coordinate transformation.

を管理し，各座標変換の条件に応じて変換処理速度の向上を図った．このようにした座標変換処理の分析結果を**図7**に示す．この結果，

（1） すべての座標変換のある場合は，16ビットの乗算が12回必要であり（倍率を32ビットで管理しているため），処理時間がかかる（約400ステップ）．

（2） これに対し，移動・拡大あるいは特殊角度の回転の場合の処理時間は，（1）の0.4～0.5に軽減される．

（3） 全く変換のない場合あるいは移動のみの場合の処理時間は（1）の0.05～0.15に軽減される．

ことがわかる．

4.2 クリッピング処理

クリッピング処理は，CRT画面枠外の描画処理を禁止するもので，マイクロプログラムで実行する場合でも

方式A＝1ドットの座標(x, y)ごとに枠外か否かをチェックし，枠内の場合にのみ画像メモリに書き込む．

方式B＝直線と画面枠との交点を演算で求め，画面枠内の部分のみのドット展開処理を行う．

の2つの方式が考えられる．これらを比較すると

（1） マイクロプログラム容量……方式Bの方がAより約100～150ステップ増加する．

（2） 処理速度……**図8**に示すように場合によって異なる．直線の描画長をlとし，

t_{dA}：方式Aの1ドット処理時間

t_{dB}：方式Aの1ドット処理時間

t_0：直線描画処理のオーバヘッド時間

T_{C1}：方式Bのクリッピング時間（場合Ⅱ）

T_{C2}：方式Bのクリッピング時間（場合Ⅲ）

とする．$\Delta t=t_{dA}-t_{dB}$ とおくと

- 場合Ⅰ（クリップなし）：

方式Aの方が $\Delta t\cdot l$ 増加する

- 場合Ⅱ（クリップ1点）：$l\geqq l_{T1}$ において

方式Aの方がBより $\Delta t\cdot(l-l_{T1})$ 増加する．

ここに $l_{T1}=T_{C1}/\Delta t$

- 場合Ⅲ（クリップ2点）：$l\geqq l_{T2}$ において

方式Aの方がBより $\Delta t'\cdot(l-l_{T2})$ 増加する．

ここに $l_{T2}=T_{C2}/\Delta t'$

$$\Delta t'=\Delta t+(l_0/l)\cdot t_{dB}$$

この $\Delta t'$ は，方式Aが長さ $l=l_0+l_1$ の処理を行わざるを得ないのに対し，方式Bは長さ l_1 の処理だけでよく相対的に t_{dB} の値が小さくなるため Δt より増加

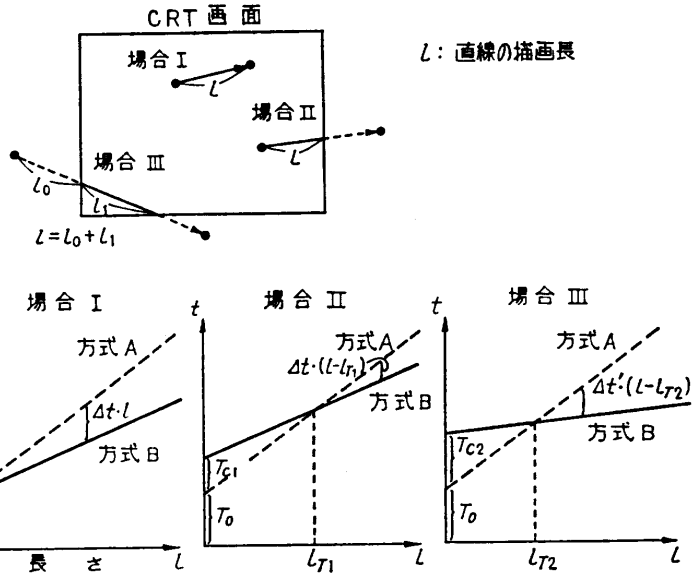


図 8 方式 A, B のクリッピング処理時間の比較

Fig. 8 Comparison of clipping time of the two methods.

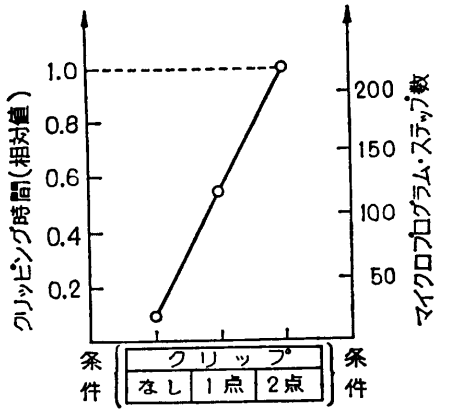


図 9 クリッピング処理時間，ステップ数

Fig. 9 Processing time and dynamic steps for clipping devider.

する．

ここで，処理時間をステップ数におきかえることにし，また方式Aの枠内チェックのための増加ステップ数，すなわち Δt を4ステップ〔直線の方向により交わる可能性のある枠の2辺と (x, y) とを比較する最小ステップをとった〕，および $l_0/l=0.5$ と仮定する．さらに，実際のクリッピング演算の処理ステップ数（図9）から $T_{C1}=120$, $T_{C2}=200$, また次節の結果から $t_{dB}=3$ が得られる．ゆえに，場合II，IIIにおいても

$l_{T1}=30$ ドット

$l_{T2}=40$ ドット

と比較的短い長さ l_{T1}, l_{T2} で，方式Aの処理時間が方式Bの処理時間を越えることがわかる．

の結果が得られた．我々は特に処理速度を重視して方式Bを採用した．この処理の分析結果を**図9**に示す．

この結果，

（1） 場合IIIのように2点でクリッピングされるときの処理時間が最も長い（約220ステップ）

（2） 場合IIのように1点でクリッピングされるときの処理時間は（1）の0.55に軽減される．

（3） 場合Iのようにクリッピングされないときは内部か否かの判定だけであり，処理時間は（1）の0.08に軽減される．

ことがわかる．

4.3 直線のドット展開処理

ここで採用した直線のドット展開アルゴリズムは，文献[8),10)]にみられるもので，**図10**を用いて簡単に説明する．

描画すべき直線を $\vec{AB}$ とし，その方向を角度 0°～45° の範囲とする．このとき，A点の次の点は右上か右横のいずれか1点であり，これは勾配 $\Delta Y/\Delta X$ と格子の中点 (1/2) とを比較することにより判定できる．すなわち，判別式 D を導入して

$$D_1=\frac{\Delta Y}{\Delta X}-\frac{1}{2} \cdots\cdots D_1 \geqq 0 \text{ ならば右上}$$

$$D_1<0 \text{ ならば右横}$$

これを第 $i-1$ 点から第 i 点について適用すると

$$D_{i-1} \geqq 0 \text{ のとき} \quad D_i=D_{i-1}+\frac{\Delta Y}{\Delta X}-1$$

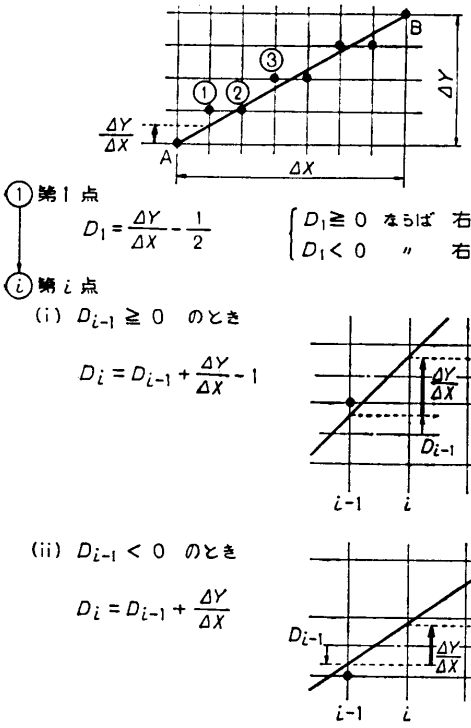


図 10 直線のドット展開アルゴリズム
Fig. 10 Algorithm of dot plotting.

$D_{i-1}<0$ のとき $D_i=D_{i-1}+\frac{\Delta Y}{\Delta X}$

が第 i 点の判別式 D_i でこの符号により

$D_i\geqq 0$ ならば右上
$D_i<0$ ならば右横

の点を選択する.

これをマイクロプログラムで実行するが, 図10において点①から点Bまでをプロットするループ処理を考えた場合, 通常のマイクロプログラム制御では**図11**(a)に示すように6ステップを要する. これ対し, (b)に示すように

(1) マイクロプログラムのループ回数を示すカウンタを設け, ループ処理の高速化を図る (このカウンタはメモリアドレスカウンタと共用する).

(2) 条件クラブの値により演算ユニット内レジスタファイルのアドレスを切換えるハードウェアを設け, 判別式Dの符号判定ステップを減らす.

ことにより, それほどハードウェアを増加させることなく直線のドット展開処理を3ステップに高速化することができた.

なお上記(1), (2)により, 座標変換処理, クリッピング処理で重要となる乗, 除算処理も高速になって

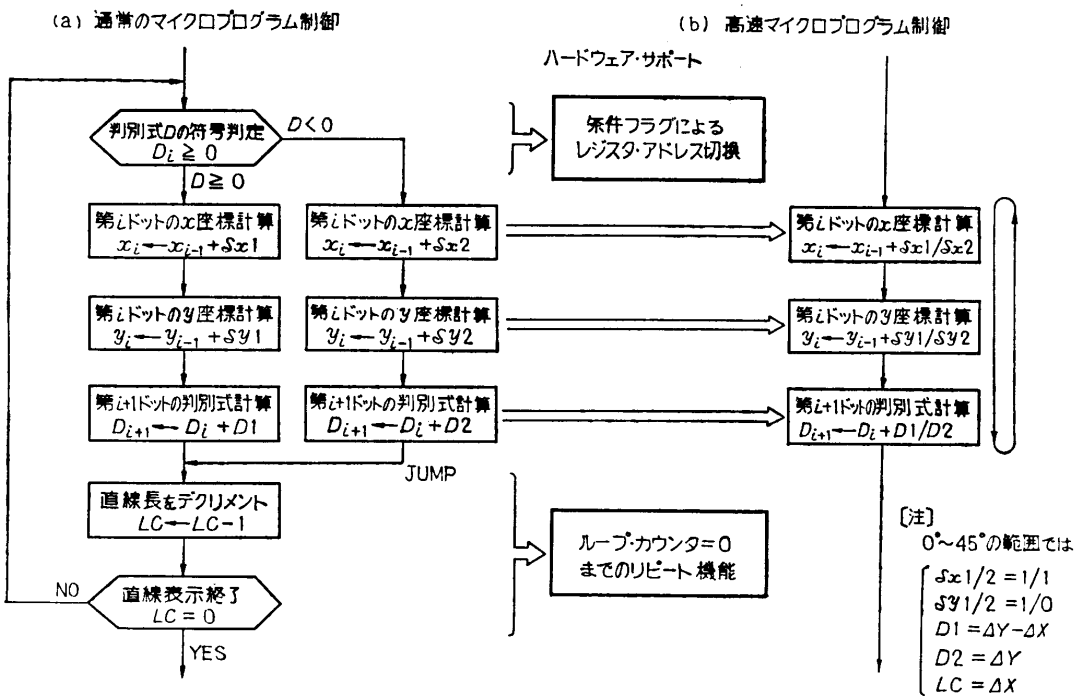


図 11 直線のドット展開処理の高速マイクロプログラム制御
Fig. 11 High-Speed microprogrammed control of dot plotting.

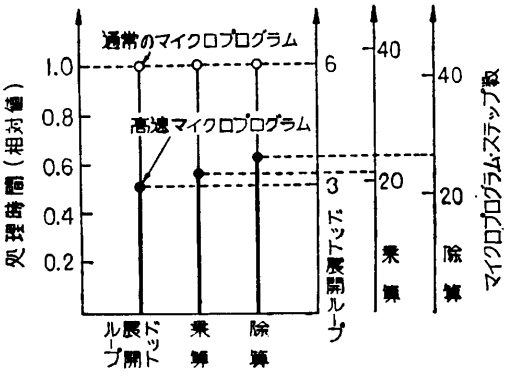


図 12 高速マイクロプログラム制御の効果

Fig. 12 Effect of high-speed microprogrammed control.

いる．たとえば16ビット乗算の場合，乗数をシフトしてそのシフトビットの値により積として被乗数を加算するか否かの処理を16回繰返して積を求める．これが上記(1)，(2)によりマイクロプログラム1ステップでループ (16回) でき，通常の2〜4ステップに対して高速化が図られる．

以上の高速マイクロプログラム制御を採用した効果を**図12**に示す．この結果から，通常のマイクロプログラム制御に対して，

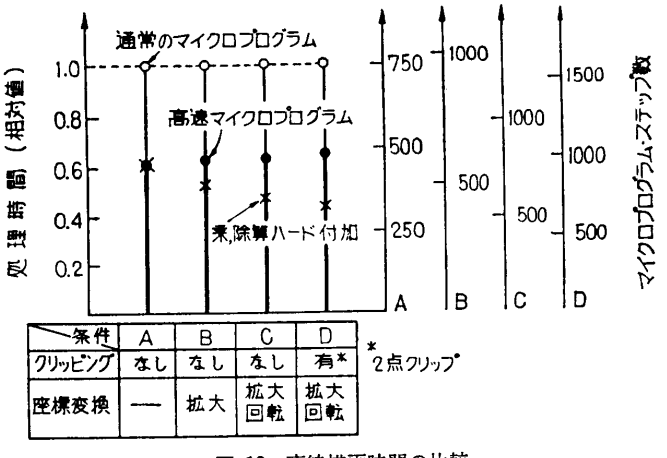
(1) ドット展開処理のループは6ステップから3ステップと1/2に短縮される．

(2) 16ビットの乗，除算処理時間も，0.55〜0.6に短縮される．

と効果の大きいことがわかる．

5. 直線描画処理の評価

前章で分析した結果から，クリッピング，座標変換



条件	A	B	C	D
クリッピング	なし	なし	なし	有*
座標変換	—	拡大	拡大 回転	拡大 回転

*2点クリップ

図 13 直線描画時間の比較

Fig. 13 Comparison of line generating time.

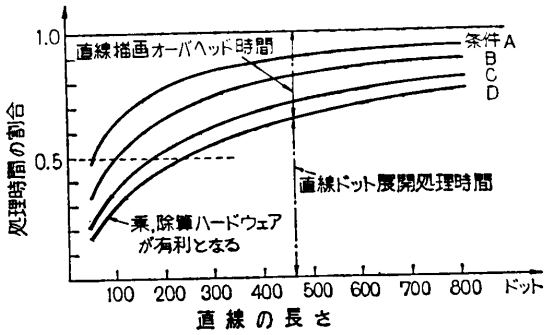


図 14 ドット展開処理時間の割合

Fig. 14 Ratio of dot plotting time.

の条件

A クリッピング〔なし〕，座標変換〔1：1 対応〕

B 〃 〔なし〕，〃 〔拡大〕

C 〃 〔なし〕，〃 〔拡大，回転〕

D 〃 〔2点〕，〃 〔拡大，回転〕

において，100ドットの長さの直線描画処理時間を

I 通常のマイクロプログラム制御 (○)

II 高速化 (4.3節) マイクロプログラム制御 (●)

III 乗，除算ハードウェアをIIに付加 (×)

で対比したものを**図13**に示す．これから

(1) 本システムの高速化マイクロプログラム制御は通常のものに比して，いずれの条件A〜Dにおいても約0.6〜0.65に処理時間を短縮できる．これは処理速度にすると1.5〜1.7倍の高速化になり，4.3節の高速化の効果が大きい．

(2) 本システムに乗，除算ハードウェアを付加した場合は，条件A，Bではあまり効果はない．条件Dでは通常のマイクロプログラム制御の0.43となり，さらに0.2程度の処理時間の短縮が期待できる．

ことがかわる．

さて，本グラフィック・プロセッサをさらに高速化するためには，上記(2)の乗，除算ハードウェアの付加のほかに，あるいはそれに加えて直線のドット展開処理とそのほかの座標変換処理等とを別々のプロセッサにしパイプライン処理を行う構成が考えられる．このため本プロセッサの直線描画処理時間のドット展開処理とそれ以外の処理（オーバヘッド）との時間比をまとめた．これを**図14**に示す．これから

(3) 条件A，B，C，Dでは，それぞれ直線長が約60，100，180，250ドットで負荷がバランスし，それ以上の長さではドット展開処

理の負荷が大きくなり，それ以下の長さではオーバヘッド処理の負荷が大きくなる．

（4） 条件 C, D では，直線長が約 90，120 ドット以上の場合はパイプライン処理にすることの方が，以下の場合は乗，除算ハードウェアを付加することの方が効果が大きくなる．

ことがわかる．

6. むすび

端末機能の高速化（仮想画面機能，図形操作機能等），使いやすさおよび高速応答を追求したカラーグラフィック・ディスプレイを開発した．本稿は，そのグラフィック・プロセッサの構成および直線描画処理の分析，評価について述べたものである．

グラフィック・プロセッサは，座標変換処理，クリッピング処理，ドット展開処理を含めた図形コマンド処理をすべてマイクロプログラム制御で行うようにしたものであるが，特に直線のドット展開処理の高速化に適した構成をとった．直線描画速度を分析，評価した結果，通常のマイクロプログラム制御に対して 1.5～1.7 倍に高速化でき，効果が大きいことを確認した．

また，さらに高速化を図るために，次の 2 つの構成について考察し，

（1） 乗，除算ハードウェアを付加した構成は，短い直線（120 ドット以下）でかつ座標変換処理，クリッピング処理が行われる場合に効果が大きい．

（2） ドット展開処理とそのほかの座標変換処理等とをパイプラインで処理する構成は，（1）とは逆に，座標変換処理を行わない場合，あるいは直線が長い場合にその効果が大きい．

ことを明らかにした．

なお本稿では直線描画処理の分析，評価だけに限定したが，別報にて円，円弧，面，キャラクタ等の描画処理の分析，評価を報告する予定である．

最後に，本研究の遂行にあたり日頃ご指導いただく日立製作所大みか工場森田副工場長，同日立研究所高砂所長およびグラフィック・プロセッサの製作および性能評価データの収集をして頂いた同日立研究所第82研究室奥山良幸氏，(株)日立エンジニアリングシステムエンジニアリング部柿爪勇二氏に深謝いたします．

参考文献

1) Laws, B. A.: A Gray-scale Graphic Processor using Run-length Encoding, Proc. of the conf. on Computer Graphics, Pattern Recognition and Data Structure, pp. 7-10 (1975).
2) Jordan, Jr., B. W. et al.: A Cell Organized Raster Display for Line Drawings, Communications of the ACM, Vol. 17, No. 2, pp. 70-77 (1974).
3) Barret, R. C.: Scan Conversion Algorithms for a Cell Organized Raster Display, Communications of the ACM, Vol. 17, No. 3, pp. 157-163 (1974).
4) Cheek, T. B.: High Performance Color Raster-scan Display without using Bit Maps, SID 78 Digest, pp. 80-81 (1978).
5) 安田 洋：電子計算機用カラー図形出力装置，NHK 技術月報，Vol. 18, No. 2, pp. 56-61 (昭 50).
6) Suenaga, Y. et al.: A High-Speed Algorithm for the Generation of Straight Lines and Circular Arcs, IE³ Transactions on Computer, Vol. C-28, No. 10, pp. 728-736 (1979).
7) Maxwell, P. C. et al.: The Generation of Polygons Representing Circles, Ellipses and Hyperbolas, Computer Graphics and Image Processing, Vol. 8, No. 10, pp. 84-93 (1979).
8) Bresenham, J. E.: Algorithm for computer control of a digital plotter, IBM System Journal, Vol. 4, No. 1, pp. 25-30 (1965).
9) 釜江尚彦他：図形のドット表示，電子通信学会論文誌 (A), Vol. 56-A, No. 7, p. 401 (昭 48).
10) Blazek, O. et al.: Raster Scan Graphics with Zoom and Pan, Hewlett-Packard Journal, pp. 6-12 (1978).
11) Sutherland, I. E.: A clipping devider, FJCC, pp. 765-775 (1968).
12) Hagan, T. G. et al.: The adage graphics Terminal, FJCC, pp. 747-755 (1968).

(昭和 55 年 4 月 3 日受付)
(昭和 55 年 10 月 23 日採録)