

ランダム走査機能を有するラスター走査型計算機 グラフィック表示装置の装置シミュレータによる実現†

平 石 裕 実^{††} 矢 島 脩 三^{††}

近年の計算機の利用の広がりと共に、人間—機械系のインタフェースとしての計算機グラフィック表示装置の占める位置は益々重要となっており、より高度な機能を有するグラフィック表示装置の出現が待望されている。本論文では、従来のランダム走査型グラフィック表示装置とラスター走査型グラフィック表示装置の得失を考慮して両者の特徴を兼ね備えたランダム走査機能を有する新たなラスター走査型計算機グラフィック表示装置の方式を提案する。次に、本装置の主要ハードウェアである表示制御部と座標演算部に対して装置シミュレータプログラムを記述しこれを計算機本体の方にインプリメントして利用することにより、速度は遅くなるもののその高度な機能にもかかわらず画像メモリを中心として少量のハードウェアで本装置を実現できることが分かったので、この方式について詳述する。さらに、本装置のいくつかの利用例をその装置シミュレータプログラムとの関連において述べ、試作した本装置の効率について述べる。

装置シミュレータの利用により、本装置をすべてハードウェアで実現するのに要する 3,150 個の IC の内約 1,300 個が節約でき、またその速度も通常の線図形表示が 1 秒程度で行え、これによりさらに多くの分野で従来のラスター走査型では困難であったライトペンによる図形指示機能を利用してより高度な計算機グラフィック表示装置の利用が開けるものと思われる。

1. ま え が き

近年の計算機の広がりに伴い、人間—機械系のインタフェースとして計算機グラフィック表示装置の占める位置は益々重要となっており、CAD や画像処理を始めとするさまざまな分野においてより高度な機能を持つグラフィック表示装置の出現が待望されている。

従来のグラフィック表示装置としては大別してランダム走査型グラフィック表示装置とラスター走査型グラフィック表示装置とがある。ランダム走査型のものはその歴史も古くソフトウェアの蓄積が豊富であり、図形表示の際に電子ビームの走査が表示図形に対応しているためライトペンで指示された図形の同定が容易であるといった特徴を持つ反面、ちらつきのため複雑な図形や面図形の表示が困難であるなどの欠点がある。一方、ラスター走査型のものは近年の集積回路技術の発展により大容量メモリが低価格で実現できるようになるのに伴い、画像メモリとテレビモニタを利用して種々開発されている^{1),2)}。これらのラスター走査型グラフィック表示装置は面図形や複雑な図形、カラー

図形をちらつきなく表示できるが、電子ビームの走査が表示図形と独立に行われるためライトペンで指示された図形の同定が困難であり、ライトペンを用いた会話的利用を防げる一因となっている。

本論文では、まずこの両方式の持つ特徴を兼ね備えた高度の機能、すなわち、ランダム走査機能を有する新たなラスター走査型計算機グラフィック表示装置の方式を提案する。次に、本装置の表示制御部と座標演算部などの主要ハードウェアをシミュレートする装置シミュレータプログラムを計算機本体の方にインプリメントして利用することにより、速度は遅くなるもののその高度な機能にもかかわらず画像メモリを中心として少量のハードウェアで本装置を実現する方式について詳述する。さらに、本装置のいくつかの利用例をその装置シミュレータプログラムとの関連において述べ、本論文の終章では試作した本装置の効率について述べる。本装置の実現によりさらに多くの分野で従来のラスター走査型では困難であったライトペンによる図形の同定機能などを利用してより高度な計算機グラフィック表示装置の利用が開けるものと思われる。

2. ランダム走査機能を有するラスター走査型計算機グラフィック表示装置の方式

2.1 装置の機能と特徴

従来のラスター走査方式とランダム走査方式の持つ

† Realization of a Raster-Scan Computer Graphic Display Device with Random-Scan Functions Using a Device-Simulator by HIROMI HIRAIISHI and SHUZO YAJIMA (Department of Information Science, Faculty of Engineering, Kyoto University).

†† 京都大学工学部情報工学科

欠点を克服し両方式の特徴を兼ね備えたグラフィック表示装置の持つべき機能と特徴とを種々検討した結果、これらはおよそ次の項目と考えられる。

(1) ランダム走査機能：面図形や複雑な図形をカラー表示するためにラスター走査型高解像度カラーテレビモニタを利用し、従来のラスター走査型グラフィック表示装置では困難であったライトペンによる図形の指示機能³⁾を実現して計算機本体からはランダム走査型グラフィック表示装置として取り扱うことを可能にする。これによりランダム走査型グラフィック表示装置用のソフトウェアを利用可能にする。

(2) ライトペンの色指定機能：カラー表示の際にたとえば赤色の背景の上に描かれた緑色の直線をライトペンで指示することを可能にするためにライトペンが検出すべき色の種類を指定可能とする。

(3) 種々の利用形態に対する適応：高機能グラフィック表示装置としての利用や簡易グラフィック表示装置としての利用、また将来の多ユーザ利用など本装置の種々の利用形態を考慮して、画像メモリの分割利用により画面の精度、1画素あたりのビット数、画面枚数をプログラマブルとする。

2.2 ランダム走査機能とライトペン方式

ラスター走査型テレビモニタを利用した本装置でランダム走査型グラフィック表示装置のもつ機能を実現するにあたり(1)画面変更の際の処理(2)ライトペン機能の2点が問題となる。

(1) 画面変更の際の処理：従来のラスター走査型グラフィック表示装置では表示画面が画像メモリに記憶されているので画面変更の際に利用者は“画面消去命令”などを用いて陽に画面を消去する必要がある、ランダム走査型の場合のようにディスプレイファイルを変更するだけでは画面変更が行えない。

本装置では画像メモリ内のデータをテレビモニタに表示するために読み出すときの読み出しモードとして“蓄積モード”と“自動消去モード”の2種のモードを設けることによりこの問題点を解決している。“蓄積モード”は単に画像メモリよりテレビモニタのラスター走査に同期して読み出すモードであり、“自動消去モード”は画像メモリより画像データを読み出すと同時に同じ語に背景色を表わす色データを書き込んで自動的に画面の消去を行う。本装置は通常“蓄積モード”で動作しており、画面を表示するときにはそのディスプレイファイルを一度だけ実行すればよく、画面変更の要求があったときには、自動的に1フレームの期間

“自動消去モード”となり画面を消去した後“蓄積モード”にて新しいディスプレイファイルを一度実行する。これにより、通常の画面表示に際しては一度だけディスプレイファイルを実行すればよいので従来のランダム走査型に比べて効率がよく、また画面変更のときには画面の消去が1フレーム時間(約30 msec)で自動的に行われ利用者はディスプレイファイルを変更するだけでよくなる。

(2) ライトペン方式：先に述べたように従来のラスター走査型グラフィック表示装置では、ライトペンで指定した図形の同定が困難である。またカラー表示の場合は背景に色を持つ図形の指示も可能とする必要がある。そこで、本装置ではライトペンが光を検出した時点でその点の色がライトペンヒットを許された色であるかを判定しライトペンヒットが許された色であれば表示図形のディスプレイファイルを最初から実行して表示図形の各点の(X, Y)座標を順次作成し、これとライトペンが指示した位置の(X, Y)座標を比較して両者が一致した時点で計算機に割込みをかけることによりライトペンの色指定機能とライトペンによる図形の同定機能を実現している。

2.3 装置の構成

本装置の構成図を図1に実線で示す。表示処理装置は、表示制御部⁴⁾と座標演算部⁵⁾とよりなり、各々マイクロプログラム制御のビットスライスマイクロプロセッサであり、パイプライン的に動作させることにより処理効率の向上を計っている。

表示制御部は、計算機の主記憶装置に蓄えられた図形表示命令を読み出して解釈し、座標演算部に図形発生のために必要なデータを送るなどシステム全体の制御を行うほか、ライトペンヒット時の処理やランダム

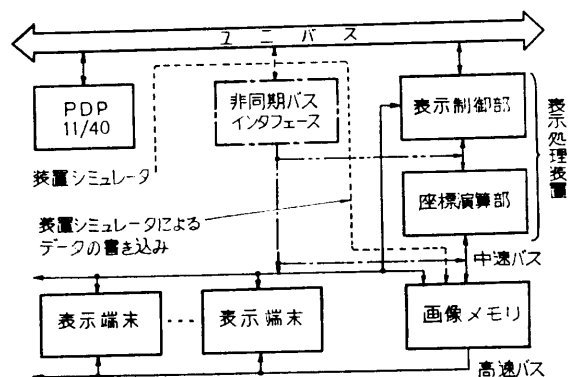


図1 システム構成図と装置シミュレータ

Fig. 1 System configuration and device-simulator.

走査機能の実現を行う。

座標演算部は、専用の図形発生器としてバクトル発生器、文字発生器など5種類のハードウェア発生器を有し、表示制御部より送られたデータに基づき、平行移動、拡大、縮小、クリッピングなどの処理を行い、座標演算部内の発生器を制御して表示図形の各点のX、Y座標と色データを逐次発生して画像メモリに書き込む。

画像メモリは、256kバイトの容量を有し64kバイト×4ブロック構成になっており、同期式の高速バス（表示端末と接続される）と非同期式の中速バス（座標演算部と接続される）の2系統からの時分割アクセスが可能で、各ブロックは並列動作可能である。高速バスからの読み出しモードとしては、先に述べたように“蓄積モード”と“自動消去モード”とがある。この2つのモードは表示制御部により制御され、ランダム走査機能の実現のために用いられる。

表示端末はテレビモニタのラスタ走査に同期して画像データを読み出すために高速バスの制御を行うほか、ライトペンやジョイスティックなどの制御をも行う。高速バスには最大8台の表示端末を接続して画像メモリの分割使用により8枚の異なる絵を表示することができるほか、異なる2枚の絵に対して画素レベルの演算を施した結果の表示も可能である。

3. 装置シミュレータプログラムの機能と構成

3.1 装置シミュレータの機能と特徴

2章で述べたような高度の機能を持つ、ラスタ走査型計算機グラフィック表示装置をすべてハードウェアで実現するためには相当量のハードウェアが必要である。一方、動作速度は犠牲になるが、本装置の主要ハードウェアである表示制御部と座標演算部とをソフトウェア的にシミュレートする装置シミュレータ⁶⁾を作成して利用することにより、本装置の持つ高度な機能を実現できるので、この方法により本装置を試作した。

装置シミュレータは図1に示すように本体計算機PDP-11/40上で動作し、非同期バスインタフェースを介して表示端末の制御を行うとともに画像メモリに画像データを書き込む。

本装置シミュレータの機能と特徴を以下に述べる。

(1) 装置シミュレータは、計算機の主記憶装置内に作成された図形表示命令を解読実行し表示図形の各

点のX、Y座標と色データを作成し画面を表示するとともに、ライトペンの色指定や図形同定機能のシミュレーションを行う。これにより、画像メモリを中心とする少量のハードウェアで本装置を動作させ、必ずしも高速度ではないが種々のアプリケーションプログラムでライトペンを用いて本装置を会話的に利用することができる。

(2) 装置シミュレータは、ハードウェアの各ブロックに対応する形で構成し、アプリケーションプログラムが必要とする速度に応じてその一部をハードウェア化することにより、画面の表示速度やライトペンの応答速度を段階的に向上させることを可能にする。このとき、すでに開発されたアプリケーションプログラムもほとんど変更することなく動作できるようにする。

(3) システム開発においてハードウェアの完成を待たなくてもアプリケーションを開始することができるなどシステム開発の一手法としても有意義である。

3.2 非同期バスインタフェース

本表示装置は、高速バスを除いてその内部の各ブロック間の接続（表示制御部と座標演算部、座標演算部と画像メモリ（中速バス）、表示制御部と表示端末（表示端末バス）など）にはすべて非同期バス方式を採用しており、アドレス線やデータ線のビット数は異なるけれども基本的には同一仕様にて設計されている。非同期バスインタフェースは、プログラム制御のもとでこれらの非同期バスのうち任意のバスに接続できる。

表1に、計算機からアクセス可能な非同期バスインタフェース内のレジスタの種類とその意味を示す。

表1 計算機からアクセス可能な非同期バスインタフェースのレジスタ

Table 1 Meaning of registers in asynchronous bus interface.

レジスタ名	ビット	意 味
ADR	15-0	接続先バスアドレス線 <15-0>
CAR	15-14	接続先バス制御線 <1-0>
	12	ライトペンヒット座標要求
	5	高速バス制御モード
	4-0	接続先バスアドレス線 <20-16>
DBR	15-0	接続先バスデータ線 <15-0>
CSR	15	ビジーフラグ
	14-13	接続先バス選択
	11	ライトペンヒット
	2	接続先バス駆動
	1-0	接続先バスデータ線 <17-16>

CSR の接続先バス選択ビットに非同期バスインタフェースの接続先を設定しておき、ADR, CAR, DBR, CSR にアドレスビット, 制御ビット, データビット (書き込み時のみ) を設定してから CSR の接続先バス駆動ビットをたてると, 非同期バスインタフェースは接続されたバスにアクセスする。またライトペンヒット時には CSR のライトペンヒットビットが 1 になるので CAR のライトペンヒット座標要求ビットをたてることによりライトペンヒットの生じた X, Y 座標を読み出すことができる。このようにして計算機から非同期バスインタフェースを介してシステム内の各ブロックにアクセスすることができる。

装置シミュレータはこの非同期バスインタフェースを用いて表示端末, 画像メモリに必要なデータの読み書きを行うことにより表示制御部と座標演算部をシミュレートしているが, 図 1 において座標演算部をハードウェア化したときには, 非同期バスインタフェースを図の二点鎖線で示すように表示制御部と座標演算部とを接続するバスに結合して, 装置シミュレータの座標演算部をシミュレートする部分を除去することにより本装置の効率を段階的に向上させることができる。

3.3 ソフトウェア構成

装置シミュレータはアセンブリ言語で記述したが, 図 2 にそのソフトウェア構成を示す。図において, ライトペン処理部と表示命令の読み出し解読部とが表示制御部のシミュレータであり, 基本図形発生部が座標演算部のシミュレータである。

装置シミュレータは, その処理効率を考慮して表示制御部と座標演算部のマイクロプログラムをマイクロ命令レベルでシミュレートするものではなく, それらのマイクロプログラムが実現する処理の流れ図に基づき機能レベルでシミュレートしている。図 3 に図形表示の際の装置シミュレータの流れ図を示す。

3.4 ライトペン処理部

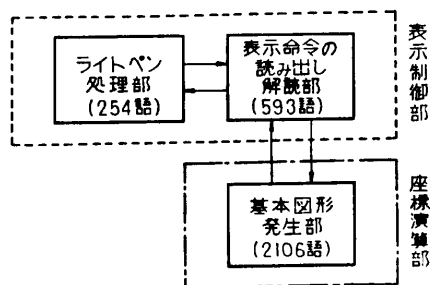


図 2 装置シミュレータの構成

Fig. 2 Structure of device-simulator.

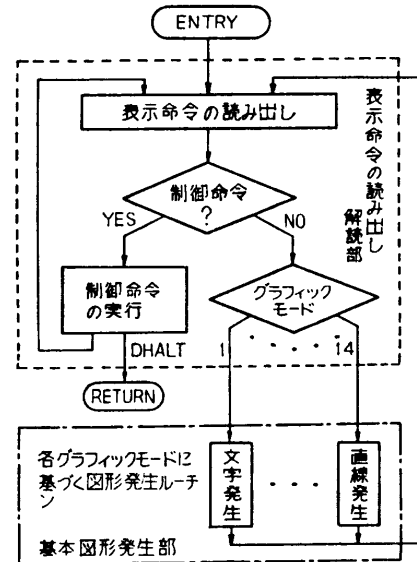


図 3 装置シミュレータの図形表示動作

Fig. 3 Flowchart of display routine in device-simulator.

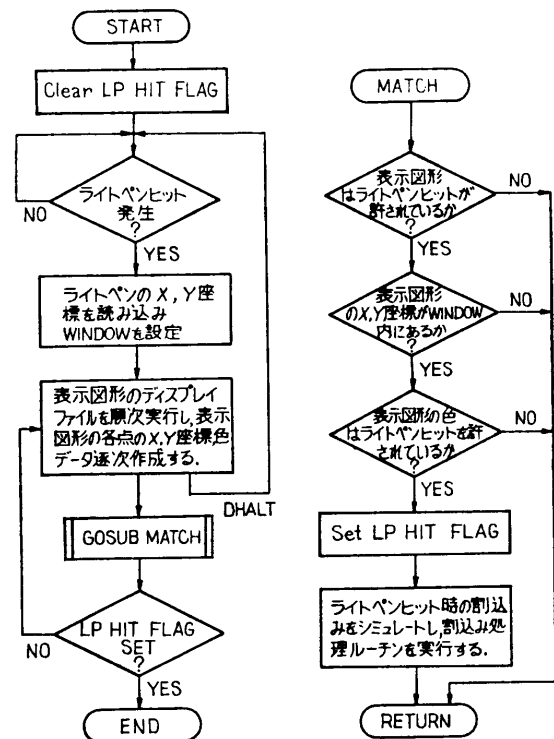


図 4 ライトペン処理の流れ図

Fig. 4 Flowchart of light pen hit simulation.

図 4 にライトペン処理の流れ図を示す。ライトペン入力要求があると, まずライトペンが光を検出するのを待って, その X, Y 座標を読み込み, ライトペン

の遅延時間を補正して適当な大きさの領域を設定する。次に、表示命令の読み出し解読部と基本図形発生部とを用いて表示図形の各点の X , Y 座標と色データを順次発生し、その図形要素がライトペンヒットを許可されており、かつ X , Y 座標が設定された領域内にあり、しかもその図形の色がライトペンヒットを許された色であれば、LP HIT FLAG をセットして、ハードウェアにおけるライトペンヒット時の割込みをシミュレートして割込み処理ルーチンを実行する。表示図形のディスプレイファイルを最後まで (DHALT) 実行しても上述の 3 つの条件を満たす点がないときは、利用者がライトペンで指示した図形要素に対するライトペンヒットが禁止されている場合なので、新たにライトペンが光を検出するまで待つ。

ライトペンの割込み処理ルーチンでは、通常その時点の表示命令やその主記憶装置上の番地、図形につけられた名前、 X , Y 座標などの情報を利用者の指定したバッファ領域へ転送する。このようにして、本装置シミュレータではライトペンの色指定機能とライトペンで指示された図形の同定機能を実現している。

4. 装置シミュレータと利用者プログラムとの関係

4.1 アセンブリ言語による図形表示

アセンブリ言語プログラムにより装置シミュレータを利用して図形表示を行う場合の例を図 5 に示す。装置シミュレータのエントリアドレス DFINTP とディスプレイプログラムカウンタ DPC をグローバルとして宣言しておき、表示図形のディスプレイファイルを DFILE 番地以降に作成し、その開始番地 DFILE を DPC に設定してサブルーチンの形で装置シミュレータにリンクすることにより図形表示を行う。このようにして作成したアプリケーションプログラムは、表示制御部と座標演算部をハードウェア化したときには、図 6 に示すように、図 5 のグローバル宣言部分と、“JSR PC, DFINTP” ステートメントを取り除き、表示制御装置内のディスプレイプログラムカウンタのユニバスアドレスをシンボル DPC として定義してやるだけでほかの部分は変更することなく利用することができる。

4.2 ユーザ BASIC による図形表示

高級言語プログラムで図形表示を行う場合、通常はグラフィック表示装置をサポートする GSP (Graphic Subroutine Package) を開発したり、言語を拡張して

```

Globl DPC, DFINTP
MOV  # DFILE, DPC
JSR  PC, DFINTP

```

装置シミュレータ内のディスプレイプログラムカウンタの番地 DPC と開始番地 DFINTP をグローバルとして宣言
ディスプレイファイルの開始番地を DPC にセットして装置シミュレータサブルーチンを呼ぶ

DFILE: ディスプレイファイルを格納

図 5 アセンブリ言語プログラムによる装置シミュレータの利用

Fig. 5 Assembly language program using device-simulator.

```

DPC=172000
MOV  # DFILE, DPC

```

ディスプレイプログラムカウンタのハードウェアアドレスを DPC として定義
ディスプレイプログラムカウンタにディスプレイファイルの開始番地をセットすることにより図形表示を行う

DFILE: ディスプレイファイルを格納

図 6 ハードウェア作成後のアセンブリ言語プログラム

Fig. 6 Assembly language program when the hardware is completed.

図形表示のための命令を組み込んだりする方法が用いられるが、これらの方法は、一般に作業量が多く、また GSP や図形表示命令の拡張や変更が容易でないといった欠点がある。そこでわれわれは、できるだけ簡単に高級言語プログラムで図形表示を行えるようにするために、マルチジョブをサポートする OS のもとで 1 つのジョブとしてグラフィック表示装置をサポートするソフトウェアを動作させ、高級言語で書かれたプログラムをほかのジョブとして動作させ、両者の間で通信を行いながら図形表示を行わせる方法が有効であると考え、実際にフォーランドジョブ (F-JOB) とバックランドジョブ (B-JOB) の 2 つのジョブをサポートする PDP-11/40 の OS RT-11 のもとで多ユーザ BASIC (MU-BASIC) に図形表示機能を追加した。

図 7 に MU-BASIC により装置シミュレータを利用して図形表示を行うときのソフトウェア構成を示す。利用者の BASIC プログラムは RT-11 の F-JOB として動作する。B-JOB は、FB 通信制御部、装置シミュレータ、ディスプレイファイル用バッファ領域より構成される。FB 通信制御部は、表示端末の初期化のほかに BASIC プログラムとの間で主記憶装置内に設けられた SYNCWD を中心とする通信用バッファ領域を介して通信を行い、装置シミュレータを用いて図形表示やライトペン処理を行う (図 8 参照)。

BASIC プログラムより SYNCWD を介して装置シ

RT 11 OS	
B-JOB	F-JOB (約 4.8k 語)
FB 通信制御部 (約 0.3k 語)	MU-BASIC インタプリタ
装置シミュレータ (約 3k 語)	(約 11.8k 語)
ディスプレイ ファイル用 バッファ領域 (約 1.4k 語)	BASIC プログラム領域 (約 7.5k 語)

図 7 MU-BASIC による装置シミュレータの利用

Fig. 7 Software structure when using device-simulator in MU-BASIC.

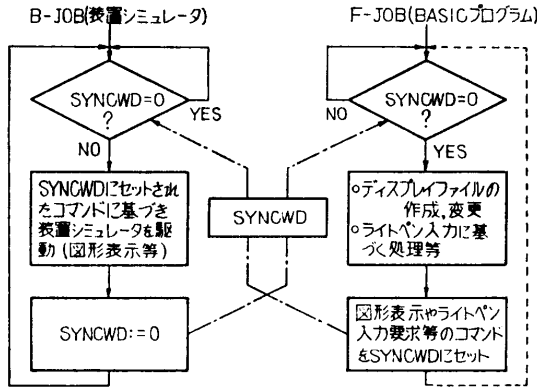


図 8 BASIC プログラムと装置シミュレータの通信

Fig. 8 Communication between BASIC program and device-simulator.

ミュレータに与えることのできる主な命令としては、

- (1) 画像メモリの消去 (画面消去)
- (2) 図形表示
- (3) ライトペンの入力要求

の3種類がある。

5. 効率の評価

本章では、本装置をすべてハードウェア化した場合と装置シミュレータによる場合との効率比較を行う。

5.1 ハードウェア化したときの効率

ハードウェア化したときの本装置の効率評価を行うために、まず画像メモリに一点分の色データを書き込むのに必要な時間 t_p について考察する。 t_p は、画像メモリの分割使用の状態、すなわち表示画面の精度、1画素あたりのビット数、表示画面数により異なるので、ここでは、最もよく使用され则认为られる 512×512、8ビット/画素、表示画面 1枚の場合について考える。

画像メモリは4ブロック構成になっており、高速バスからはテレビモニタのラスター走査に同期してブロック単位で8語並列にアクセスされる (水平走査1回につき 32 回)。あるブロックに対して高速バスと中速バスからのアクセスが競合した場合には、中速バスからのアクセスが待たされる。中速バスからの1回のアクセスに対して高速バスからのアクセスとの競合が生じる確率を p 、そのときの平均待時間を t_{ch} 、競合のために生ずる平均待時間を t_{WH} とすると

$$p = \frac{t_{MI} \times 32}{t_H} \times \frac{1}{4} \times \frac{N_V}{N_F} + \frac{t_{MI}' \times 4}{t_H}$$

$$t_{ch} = \frac{t_{MI}}{2} \times \frac{1}{p} \times \frac{t_{MI} \times 32}{t_H} \times \frac{1}{4} \times \frac{N_V}{N_F} + \frac{t_{MI}'}{2} \times \frac{1}{p} \times \frac{t_{MI}' \times 4}{t_H}$$

$$t_{WH} = p \times t_{ch}$$

ただし t_H : 水平走査時間 63.5 μ sec

t_{MI} : 高速バスからのアクセスによる中速バスアクセス禁止時間

1,050 n sec (蓄積モード)

1,210 n sec (自動消去モード)

t_{MI}' : 画像メモリのダイナミックメモリをリフレッシュ (水平帰線消去期間に全ブロックに対し4回アクセスする) するときの中速バスアクセス禁止時間

1,050 n sec

N_F : 1フレームの走査線数 525 本

N_V : 1フレームの可視領域の走査線数 480 本

蓄積モードと自動消去モードの場合の t_{WH} の値は

$$t_{WH} = \begin{cases} 98 \text{ n sec (蓄積モード)} \\ 120 \text{ n sec (自動消去モード)} \end{cases}$$

となる。

次に水平方向・垂直方向・水平軸と $\pm 45^\circ$ の方向の4本の直線を描く場合の一点あたりに要する平均時間 t_p を求める。簡単のために高速バスとの競合は考慮しない。 t_d を中速バス上で前の読み書きサイクルが終了した後次のサイクルが開始するまでの遅延時間とし、 t_a を画像メモリのアクセス時間、 $t_a + t_b$ を画像メモリのサイクル時間とすると (ただし $t_b < t_a + 2 \times t_d$ とする)

$$t_p = \frac{49}{64} \times (t_a + t_d) + \frac{15}{64} \times \max(t_a + t_b, t_a + t_d)$$

となる。

高速バスとの競合を考慮した場合、話は複雑になる

が、中速バスから連続的にデータを書き込むときの1点を書くのに要する平均時間 t_p' は

$$t_p' \approx t_p + t_{WH}$$

であると考えられる。ここで、画像メモリの速度として $t_a = 220$ nsec, $t_b = 260$ nsec (蓄積モード), 420 nsec (自動消去モード) 座標演算部のベクトル発生器の速度を考慮して $t_a = 160$ nsec とすると

$$t_p' \approx \begin{cases} 500 \text{ nsec (蓄積モード)} \\ 560 \text{ nsec (自動消去モード)} \end{cases}$$

となる。これより画面全体を面図形の長方形として塗りつぶすのに約 130~150 msec を要することが分かる。

次に本装置全体の効率をベクトル発生の場合について考察する。ベクトルモードの場合、表示制御部のマイクロプログラムは9ステップ、座標演算部のマイクロプログラムのステップ数は3である。表示命令の読み出し、表示制御部、座標演算部はパイプライン動作を行うので、長さ6以上のベクトルについては、ほぼ t_p' の時間ごとにベクトルの各点を発生させることができるものと思われる。図9に、32 msec の間に発生することのできるベクトルの本数とベクトル長との関係を、本装置の場合と DEC 社のランダム走査型グラフィック表示装置 VT 11 の場合⁷⁾について示す。なお図中の点線のグラフは、本装置の最大長ベクトルの長さを VT 11 のそれと等しく 1,024 と考えた場合を示している。

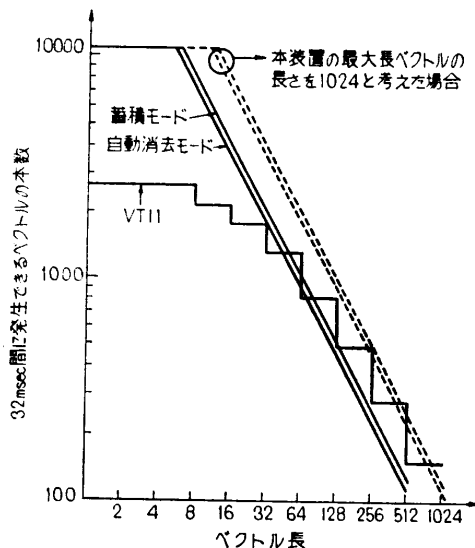


図9 32 ミリ秒間に発生可能なベクトルの本数とベクトル長の関係

Fig. 9 Relation between maximum number of vectors per 32 msec and vector magnitude.

表2 装置シミュレータの効率

Table 2 Efficiency of device-simulator.

	基本ループ	一点発生するのに要する時間 (実測)
長方形	124	155 μ sec
ベクトル	154	195 μ sec

表3 ハードウェアと装置シミュレータの効率比較

Table 3 Comparison of performance between hardware and device-simulator.

表示制御部と座標演算部の実現方法	図形発生速度			ソフトウェア量/ハードウェア量
	長方形	ベクトル	比	
装置シミュレータ	155 μ sec/点	195 μ sec/点	275/390	5,906 バイト (内フォントメモリ 1,920 バイト) (1,500 ステップ内フォントメモリ 200 ステップ)
ハードウェア	500~560 nsec/点			1
				IC 数 1,310 個 (内 SSI 810 個, MSI 370 個 LSI (メモリ素子も含む) 130 個) ゲートレベル約 16,700 ゲート+14k バイト (メモリ素子)

5.2 装置シミュレータの効率

表2に装置シミュレータの長方形発生部とベクトル発生部の基本ループにおいて1点を発生するのに必要な PDP-11/40 のメモリサイクル数と発生時間を示す。文字発生の場合は、1文字につき光らせる点の数を N とすると、1文字発生に必要なメモリサイクル数は $(885 + 122 \times N)$ サイクル

である。192 種の 9×7 ドットの文字に対する N の平均値は約 16 なので1文字の発生に約 3.6 msec かかり1秒間に約 280 文字表示できる。これは 2,400 bps のキャラクタディスプレイ相当の速度であり十分実用に耐えると考えられる。実際に BASIC で1画面塗りつぶすのに約 50 秒、図1の本装置構成図 (各ブロック内と UNIBUS 部分を塗りつぶす) の表示に約 12 秒、1,920 文字の表示に約 10 秒かかる。

5.3 装置シミュレータとハードウェアの比較

表3に、装置シミュレータによる本装置の効率と、表示制御部と座標演算部をハードウェア化したときの本装置の効率の比較を示す。

図形発生速度は、ハードウェアによる場合は1点あたり 500~560 nsec を要し、一方、装置シミュレータによる場合は1点あたりに必要な時間は 155 μ sec~195 μ sec であるので、表示制御部と座標演算部のハードウェア化により 275~390 倍の速度向上が実現される。

装置シミュレータのソフトウェア量は、文字発生用のフォントメモリ 1,920 バイトを含めて 5,906 バイトであり、またステップ数は 1,500 ステップである。一方、ハードウェア量としては、表示制御部と座標演算部の論理設計を行った結果、IC 数として SSI 810

個, MSI 370 個, LSI 130 個で合計 1,310 個となった。これをゲートレベルと記憶素子の容量に換算すると約 16,700 ゲート+14k バイトとなる。これより、フロントメモリに対応する部分を除いて考えると装置シミュレータのアセンブリ言語 1 ステップ (平均 3 バイト) がハードウェアにおける IC 1 個に対応しており、また装置シミュレータソフトウェアの 1 バイトは、ハードウェアで約 4.2 ゲート+3 バイトに相当している。

6. む す び

本論文では、従来のランダム走査型とラスター走査型の両者の特徴を兼ね備えたランダム走査機能を有する新たなラスター走査型計算機グラフィック表示装置の方式を提案し、その主要ハードウェアに対するソフトウェア装置シミュレータの利用により速度は必ずしも速いとはいえないが高度の機能を有する本装置を画像メモリを中心として少量のハードウェアで構成できることを示した。装置シミュレータの速度はハードウェアによる場合に比べて約 275~390 倍遅いが、5.2 節で述べたように 1,920 文字の表示や図 1 の装置構成図の表示が約 10 秒程度で行える性能を持つので、多くの分野で高度の機能を有する計算機グラフィック表示装置の利用が開けるものと思われる。現在、実際に本装置を、会話型図形編集システム GEST⁸⁾のもとで学会・講演会用のカラースライド作成に用いており、また漢字のデータ圧縮の研究⁹⁾などにも利用している。さらに本装置の表示処理装置のハードウェア化を進めることにより本装置の効率も段階的に向上し、より高速性を必要とするアプリケーションにおいても本装置の利用が可能となるほか、すでに開発したアプリケーションプログラムも変更することなくその効率が向上する。このように装置シミュレータの利用は、アプリケーションプログラムが必要とする速度に応じたシステムをハードウェア量との兼ね合いで段階的に構成することができ、またシステム開発においてハードウェアの完成を待たなくてもアプリケーションプログラムの開発が可能になるなどシステム開発の一手法としても有効であろう。

謝辞 本装置の開発にあたり貴重なご意見・ご討論をたまわった本学上林弥彦助教授、稲垣耕作博士、また本システムの論理設計やハードウェア開発などにご協力頂いた川久保和雄氏 (現在国鉄勤務)、増田一彦氏 (現在気象庁勤務)、桑原禎司氏、嶋田弘僧氏に深謝します。なお本研究は一部文部省科学研究費ならびに倉田奨励金による。

参 考 文 献

- 1) Baskett, F. and Shustek, L.: The Design of a Low Cost Video Graphics Terminal, SIGGRAPH-ACM, Vol. 10, No. 2, pp. 235-240 (1976).
- 2) Myers, W.: Interactive Computer Graphics: Flying High-Part I, IEEE Computer, Vol. 12, No. 7, pp. 8-17 (1979).
- 3) Newman, W.M. and Sproull, R.F.: Principles of Interactive Computer Graphics, McGraw Hill (1973).
- 4) 平石, 増田, 矢島: ランダム走査機能を有するカラーテレビ走査型計算機グラフィックスにおける表示制御装置の設計, 電子通信学会技術報告, EC 77-76 (1978).
- 5) Hiraishi, H., Kawakubo, K. and Yajima, S.: The Design of a Microprogrammed Graphic Display Processor with Pipeline Organization, Proc. 3rd EUROMICRO Symp., pp. 66-73 (1977).
- 6) 平石, 矢島: テレビ走査型計算機グラフィック表示装置の開発における装置シミュレータを利用する手法, 電子通信学会技術報告, IE 78-17 (1978).
- 7) "GT 44 User's Guide," Digital Equipment Corporation (1973).
- 8) 平石, 青木, 矢島: テレビ走査型計算機グラフィックスを利用したスライド作成用図形編集システム GEST, 電子通信学会技術報告, EC 78-64 (1979).
- 9) 矢島, Goodsell, J.L., 市田, 平石: Data Compression of Kanji Character Patterns Digitized on the Hexagonal Mesh, 電子通信学会技術報告, EC 78-63 (1979).

(昭和 54 年 8 月 30 日受付)

(昭和 55 年 9 月 18 日採録)