

高速画像データ転送機能を有するCRTコントローラ

出口雅士, 東 幸哉, 若林直樹, 神山祐史, 前信 潔, 宇屋 優, 永富和保, 小沢純雄
(松下電器)

1. まえがき

近年、OA化の進展に伴ないパーソナルコンピュータやワードプロセッサ等がめざましく普及してきている。これらの機器では、従来文字情報を扱ってきたが、マンマシンインタフェースの向上を目的として、ビットマップディスプレイを用いて文字や図形を画面上で容易に編集できる機能が要求されるようになってきた。この要求に応える方法としてCRTC (CRT Controller) のインテリジェント化がはかられているが、これらのLSIはビットマップディスプレイ装置のフレームメモリ (Frame Memory 以下FMと記す) 中での画像編集という点に重点がおかれていた。¹⁾

ところが、半導体技術の進歩によるメモリの高集積化・低価格化が進んだことと、スキャナ等によるイメージ処理が普及してきたことにより、FMとは独立に表示原画を蓄えておくイメージメモリ (Image Memory 以下IMと記す) を備えたシステム構成²⁾が強く求められるようになってきた。著者らは、このような要望に応えるディスプレイプロセッサ (以下DPUと記す) を開発した。

DPUは、IM、FMに対する高速処理を可能とするため、バス帯域の利用効率を向上させること、画像データのビット処理を内部に取り込みパイプライン処理することにより、高速なBit Block Transfer^{3),4)}を実現したLSIである。上記メモリ相互間・メモリ内の矩形領域の画像データを高速に転送することによりマルチウィンドウ処理をサポートする。また、書換え可能制御記憶を内蔵することにより多様な画像データ処理の実現を可能としたことが大きな特徴である。

本稿ではDPUの機能と内部構成、及び書換え可能制御記憶の応用例について述べる。

2. 特徴

DPUはマルチウィンドウ表示をサポートする高速データ転送機能と柔軟なCRT制御機能を実現した専用LSIである。概略仕様を表1に示す。

2.1 各種機能説明

(a) 高速データ転送機能

8種のラスタ演算モードを用いて、最高 400 nsec/16ドットの速度転送が可能である。また転

表1 概略仕様

パッケージ	84ピン プラスチックパッケージ (フラット)
プロセス	CMOS 2 μm 2層A1
トランジスタ数	85000 トランジスタ
動作周波数	最大 20MHz
電源電圧	5V単一
命令セット	33 命令
メモリサイクル (FM)	200 ns (20MHz時) 同期式
(IM)	最小 400 ns (20MHz時) 非同期式
メモリ空間	(FM) 2 MB (カーソル領域含む)
(IM)	16 MB
表示画素数	最大 4096ドット×4096ドット
転送制御	(速度) IM→FM 最小 400ns
20MHz時 16ドット転送	IM→IM 最小 1200ns
	FM→FM 600ns
	FM→IM 最小 800ns
	(倍率) 1/n, n倍 (n=1~16, 整数)
(単位)	最小 1ドット
(スラ演算)	8種類
CRT制御	スキャンモード 3種類、外部同期可
カーソル制御	最大255ドット×255ドット任意形状、マスク可
描画機能	水平・垂直直線発生 (線種登録方式)
	ドット描画 (8近傍相対位置指定可)

送時の拡大 (1~16倍)、縮小 (1~1/16倍) 転送の指定もできる。

(b) 書換え可能な制御記憶の内蔵

内部ハードウェア制御のためのマイクロプログラムを格納する制御記憶 (768ワード) のうち一部 (64ワード) をRAMで実現したことによりユーザー定義マイクロプログラムが可能である。

(c) 外部回路を必要としない任意形状カーソル表示

最大255ドット×255ドットの任意形状カーソルを登録しておき、フレームメモリに直接書込むことにより表示する。また、高品質カーソル表示のために背景データのマスク機能も有する。

(d) 2MBのFM空間とは独立に16MBの高線密度での原画像データを格納できるIM空間に直接アクセス可能。

(e) CRT表示制御機能

3種のスキャンモード（インタレース、ノンインタレース、インタレースシュリンク）が設定できる。また、フレームメモリ読出し表示アドレスの発生とアクセス間隔に対し柔軟な設定ができる。

2.2 コマンド体系

コマンド体系の特徴は次に示す2項目である。

(a) イメージメモリとフレームメモリの一貫性

描画コマンドが対象とするメモリ空間、転送コマンドの転送元（Source）空間、転送先（Destination）空間としてIM, FMのいずれも指定できる。すなわち、各コマンドはソース空間とデスティネーション空間の物理的属性とは無関係に適用できる。

(b) X-Y座標管理

全てのコマンドは、XY座標を用いて定義しているため、メモリ（IM, FM）の物理アドレスを意識することなくユーザはプログラム開発ができる。

次に、コマンドの使用例として矩形領域データの転送方法を手順を追って説明する。（図1）

① ソース空間、デスティネーション空間の定義（DFSRCS, DFDSTSコマンド）

空間の物理的属性（IM, FM）、座標原点の物理アドレス（SORG, DORG）、転送領域の幅（SWDT, DWDT）を規定する。

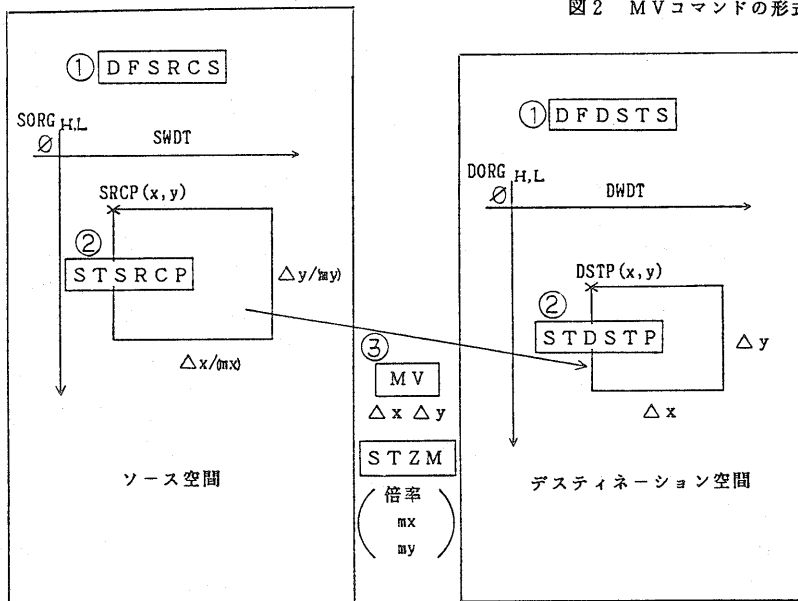


図1 矩形領域データの転送方法

② ソースポインタ、デスティネーションポインタの設定（STSRCP, STDSTPコマンド）

転送制御の基準となるポインタをX-Y座標で規定する。

③ 矩形領域転送コマンドの実行（MVコマンド）

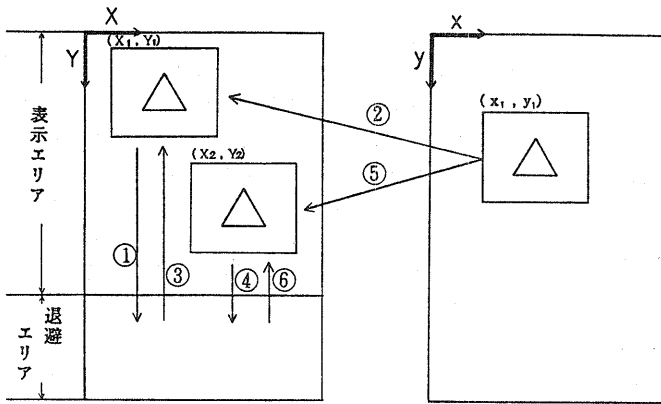
MVコマンドの形式を図2に示す。DIRフィールドは転送走査シーケンス（4種）制御用であり、Bフィールドはイメージメモリへの連続バス使用制御用である。ZフィールドはSTZMコマンドにより設定された倍率で拡大/縮小転送を行うときに指定し、ROPフィールドはラスタ演算モード（8種）の指定を行うものである。

以上示した ① ~ ③ の一連のコマンド実行により、IM-IM, FM-FM, IM-FM間での高速なデータ転送が実行できる。さらに、この高速データ転送機能を用いることにより、図3に示すようにマルチウィンドウ表示に必要な次の処理が高速化できる。

- ・ウィンドウの開設、移動、消去
- ・ウィンドウ内データのビットスクロール
- ・ウィンドウ内データの拡大、縮小

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	0	1	Z	0	0	0	0	DIR	B	ROP				0
												△ X			
												△ Y			

図2 MVコマンドの形式



(FM)

(IM)

<ウィンドウの開設>

- ①. 下絵データの退避
- ②. ウィンドウデータの転送

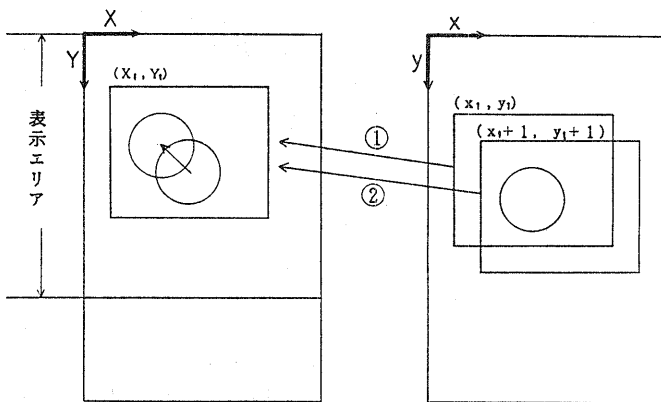
<ウィンドウの移動>

- ③. 下絵のデータの復帰
- ④. 下絵データの退避
- ⑤. ウィンドウデータの転送

<ウィンドウの消去>

- ⑥. 下絵データの復帰

*注) 退避エリアをFM上にとった場合

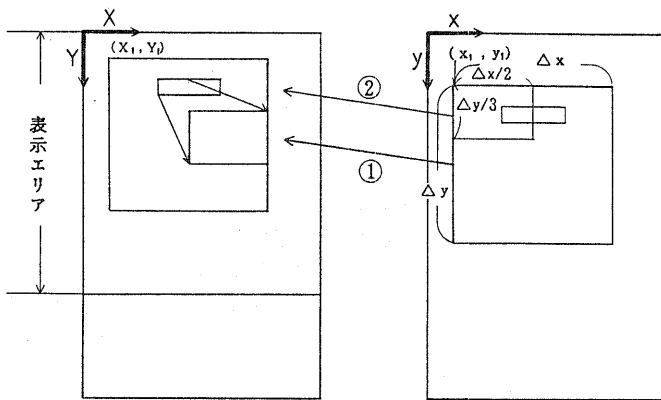


(FM)

(IM)

<ウィンドウ内データの
ビットスクロール>

- ①. ウィンドウデータの転送
- ②. ソースポイントを1ビットシフトした
ウィンドウデータの転送



(FM)

(IM)

<ウィンドウ内データの拡大>

- ①. ウィンドウデータの転送
倍率 $\begin{pmatrix} X-1 \text{ 倍} \\ Y-1 \text{ 倍} \end{pmatrix}$
- ②. ウィンドウデータの転送
倍率 $\begin{pmatrix} X-2 \text{ 倍} \\ Y-3 \text{ 倍} \end{pmatrix}$

図3 マルチウィンドウ表示例

3. システム構成

図4にDPUを用いたシステム構成を示す。画像を扱うためにはメモリ階層の考えが必要で、

- ・大容量・中速の原画を蓄えておくべきIM(16MB)
- ・小容量・高速の表示データのためのFM(2MB)

の2種類のメモリを考え、DPUはこれらをそれぞれ独立に処理・制御できるようにしている。

IMが接続されるIMバスには、画像処理プロセッサ、光ディスク、イメージ・スキャナなど原画を発生・処理する装置が接続されることを想定し、コマンド・パラメータを入力するホストバスとは独立させている。これらの構成はmaxモードと呼んでいる。

その他にホストバスとIMのデータバスとがマルチプレクスされるminモードがある。この構成は、ホストプロセッサが画像処理プロセッサとしても動作する場合に使用される。

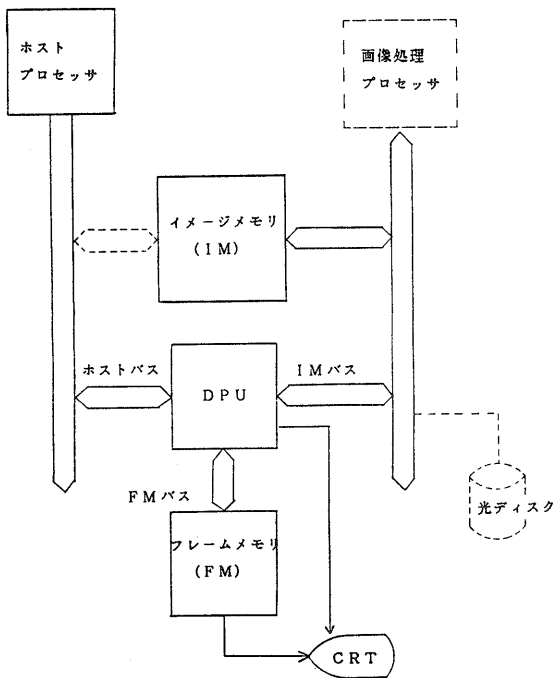


図4 システム構成例

4. 内部構成

図5にDPUの内部ブロック図を示す。内部は大きく以下の4つのブロックに分かれる。

4.1 命令解釈部

(a) ホストインターフェイス部 (HOST I/F)

ホストプロセッサのインターフェイスとして16ビット×5ワードのFIFO (First-In First-Out) メモリを内蔵している。このFIFOメモリは、ホストプロセッサからのコマンド及びパラメータの受信と、読み出し系コマンド実行時の応答パラメータを送信するために双方向構造である。

(b) マイクロ制御部 (Micro Code Gen. & Decoder)

FIFOメモリからコマンドを取り出し、解釈を行って、制御記憶 (27ビット×768ワード) 内のマイクロ命令ルーチンを実行する。制御記憶はROM領域 (704ワード) とRAM領域 (64ワード) から構成されている。RAM領域へのマイクロ命令ルーチンの格納は、IM上の指定されたアドレスを先頭として連続128ワード (2ワード=32ビットでマイクロ命令1ワード=27ビットを表現する。) に定義されたマイクロコードを読み込む事によって行う。

4.2 演算実行部

(a) 算術論理演算部 (ALU & BSH)

ホストプロセッサから与えられるパラメータを加工し内部レジスタに格納するための16ビット算術論理演算器 (ALU) に加え、転送時のビット位置合せとシフト動作強化のため、32ビットの入力データのうち連続した16ビットを切出すバレルシフト (BSH) を内蔵している。

(b) 汎用レジスタ (GR)

16ビット×32ワードの汎用レジスタを内蔵している。このうち21ワードは処理パラメータ設定用レジスタとして使用し、残りの11ワードは作業レジスタとして使用する。

4.3 表示制御部

(a) 水平・垂直同期信号発生部 (CRTC)

CRTへの同期信号を発生し、表示アドレス・リフレッシュアドレス送出タイミングを制御する。

(b) 表示アドレス・リフレッシュアドレス発生部 (CRT Adr. Gen.)

表示区間で20ビットの表示アドレスを発生する加算器と、各水平同期信号区間で9ビットのダイナミックRAMに対するリフレッシュアドレスを発生するリフレッシュカウンタを内蔵している。

4. 4 データ転送制御部

(a) IMインターフェイス (IM I/F)

IMとは23ビットの多重バスで接続し、バス・リクエストを出力しバスが使用中でなければアドレス送出を開始する非同期方式アクセスを行う。バスアクセス周期は最小8クロック、400nsec(20MHz動作時)である。バスの上位7ビットはアドレス専用バス、下位16ビットはアドレス/データの多重バスである。

(b) FMインターフェイス (FM I/F)

FMとは20ビットの多重バスで接続し、4クロック周期の同期アクセスを行う。バスアクセス周期は200nsec(20MHz動作時)である。バスの上位4ビットはアドレス専用バス、下位16ビットはアドレス/データの多重バスである。

(c) 転送アドレス発生部 (Trans. Adr. Gen.)

データ転送時における転送ワード数を管理するカウンタ及び転送元アドレス、転送先アドレスを発生するハードウェアより構成される。

(d) ラスタ演算部 (ROPU)

内部バス (A, B bus) とは独立に外部バス (Data bus) 部に16ビットのラスタ演算用論理演算器を設け、ALU動作と並行してラスタ演算 (演算モード8種) を行うことができる。

5. データ転送におけるパイプライン構造

図6にデータ転送における3段のパイプラインの各段階と、それに対応する内部ハードウェアのブロックを対比して示す。

(a) パイプライン動作の第一の段階では、IM又はFMからの転送元データ ① をDQレジスタ (Data Queue 2ワード) へ取込む。

(b) 第二の段階ではビット位置合わせを行う。DQレジスタ先頭データ ② とテンポラリーレジスタTRのデータ ③ の合計32ビットをバレルシフトBSHに入力し、転送先のワード境界に整合してビット位置合わせを行った16ビットデータ ④ を切出しデータバッファレジスタDBRに格納すると同時に、DQ先頭データ ② をTRに格納する。

(c) 第三の段階ではラスタ演算を行う。転送先データ ⑤ をLATCH2に格納し、LATCH1のデータ ⑥, LATCH2のデータ ⑦ をROP (Raster Operation用演算器) に入力し、ラスタ演算結果 ⑧ を転送先に書込む。

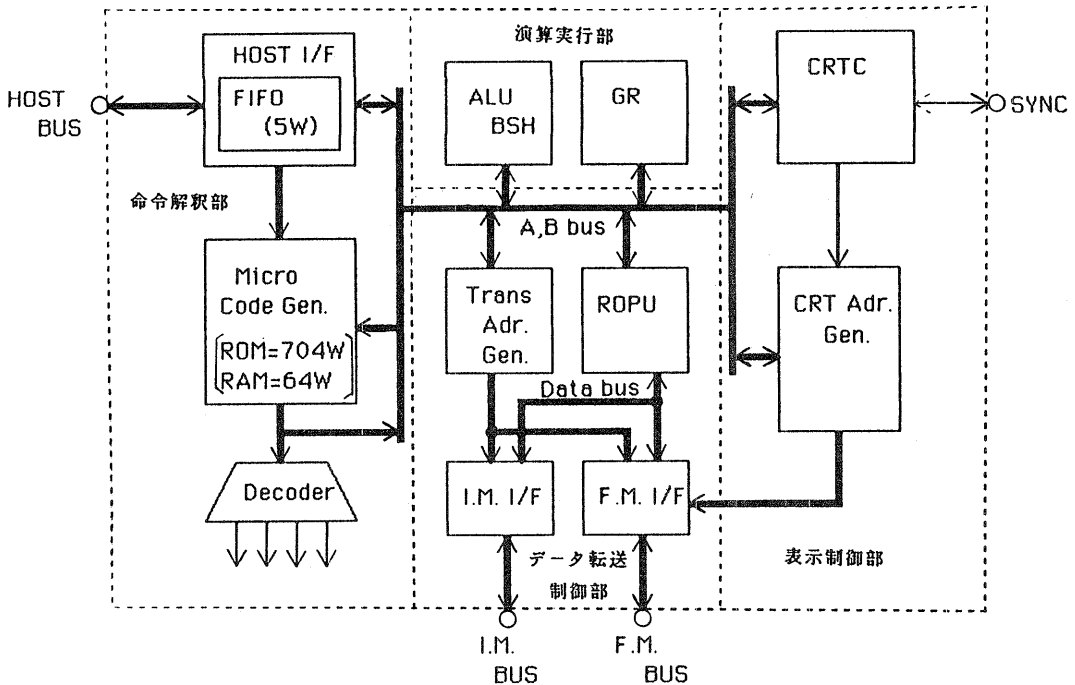


図5 内部ブロック図

転送元をIM、転送先をFMとした場合のタイミング図を図7に示す。

- ・バスアクセス周期は、前にも述べたように、最小IMアクセス周期（8クロック）はFMアクセス周期（4クロック）の2倍と設計した。そのためIMからFMへデータ転送の場合、1ワードのデータ転送あたり
 - ・IMアクセスは1回（ソースデータの読み込み）

- ・FMアクセスは2回（更新前のデータの読み込み、ラスタ演算後の更新データの書き込み）となり、IMアクセス時間とFMアクセス時間とが等しくなる。
 - ・外部バスにラスタ演算器を直結したことで、FMのアドレス出力の間でラスタ演算が行える。
- 以上の2点より、外部バスアクセス時間のみでデータ転送が行え、最小400nsec/16ドットの転送時間を実現している。

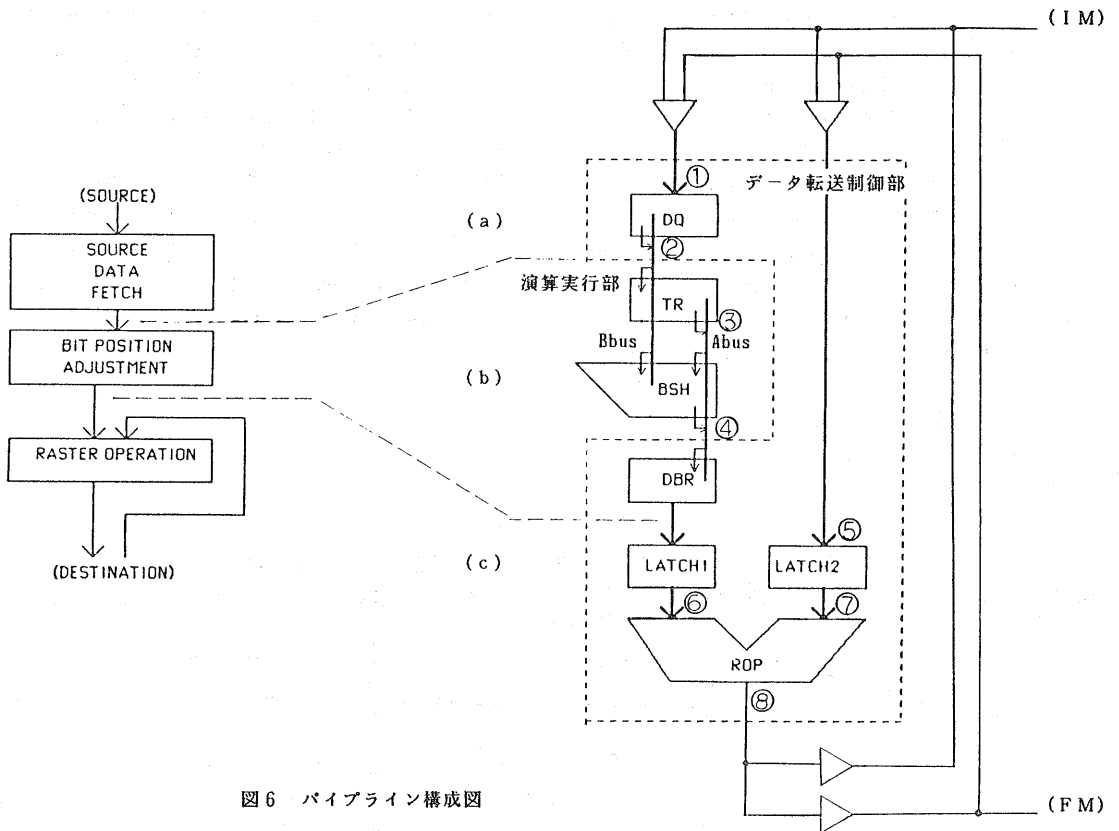


図6 パイプライン構成図

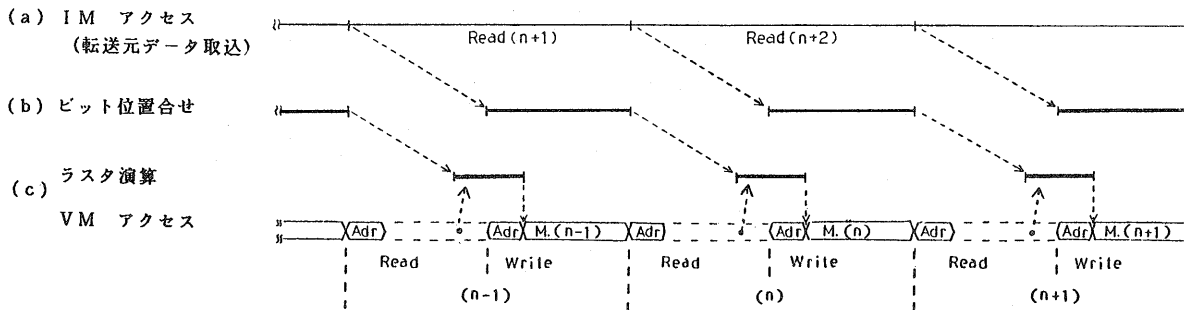


図7 パイプラインタイミングチャート

6. 書換え可能制御記憶

従来、CPUや専用コントローラ等のLSIでは、制御記憶をマスクレイアウト上パターン効率のよいROMで実現していた。DPUは、メインフレームやミニコンで使用されているダイナミックマイクロプログラミング⁵⁾を採用し、制御記憶の一部をRAMとして内蔵した。既にCPUなどでは、LSI外部に制御記憶を設け柔軟性を得ようとする試みは多く存在する。⁶⁾しかし、DPUにおいては高速化を目的として書換え制御記憶のオンチップ化を考えた。

この書換え可能制御記憶を、マイクロ診断とユーザ定義マイクロプログラムに用いている。

6.1 マイクロ診断

チップ製造時の機能検査として、コマンドレベルの機能検査に加えて、マイクロ診断用テストプログラムを書換え可能制御記憶にロードし、実行することにより、マイクロ命令レベルでチップ内部の詳細な機能検査が可能となり、チップ検査の精度が向上している。

6.2 ユーザ定義マイクロプログラム

DPUのコマンドとして制御記憶のROM領域に常駐している描画コマンドは、点描画や水平・垂直線描画などの基本的なものにとどめた。

これは、グラフィックシステムには一つの機能を実現するためだけでも多くの属性が要求され、それらをすべてROMでサポートするのはROM容量の限界により不可能と判断したためである。そのため制御記憶の一部(64ワード)のRAM化を試み、ユーザ定義マイクロプログラムをロードするためのコマンドと実行開始コマンドを備えた。

更にこのようにして実現された制御記憶の汎用化により、アプリケーションプログラムに依存する各種の描画ルーチンのうちボトルネックとなる部分を任意に抽出しファームウェア化することも可能となった。

(a) ユーザ定義マイクロプログラムでの実現例

- ・ 図形発生：(任意角度の)直線描画、円(弧)描画、楕円(弧)描画など
- ・ 図形変換：回転、鏡像、(任意倍率の)拡大・縮小など
- ・ その他：MH符号圧縮・伸長、閉領域塗りつぶし、輪郭追跡、文字フォント展開など

(b) 処理ステップ数評価

閉領域塗りつぶしを例にDPUとCPUの処理ステップ数の比較を行ったものを表2に示す。

これは、4bit/pixelのFM上で塗りつぶすワード数をWとした時の1ライン当りの処理ステップ数である。

表2 閉領域塗りつぶしの場合のDPU・CPU処理ステップ数

処理プロセッサ 処理内容		DPU	CPU (MNI617)
		塗りつぶし	
塗りつぶし	境界点サーチ	45W+115	200W+150
	1ライン塗りつぶし	4W	30W
計		49W+115	230W+150

W=塗りつぶしワード数/1ライン

これらの結果からステップの効率化度Eを考えると、

$$E = \frac{(230W+150) \times L}{(49W+115) \times L + \text{ロード実行ステップ数}}$$

$$E = \frac{(230W+150) \times L}{(49W+115) \times L + \text{ロード実行ステップ数}}$$

L:ライン数

ロード実行ステップ数は、403ステップなので塗りつぶし領域を例えば640×480とした場合、E≒4.5となる。

(c) グラフィックシステムとの整合性

既存のグラフィックシステムの一部にDPUのような専用LSIを組込む場合、第一に問題となるのがその整合性である。一般にソフトウェアの変更には多くの労力が必要となり、問題によっては変更不可能な場合がある。

図8は、CPUで行っていた塗りつぶし処理の各機能を、DPUに順次置換えていった過程で処理時間の低減化を図った結果を示す。測定条件として垂直方向480ラインの矩形領域(4bit/pixel)を塗りつぶす場合で、水平方向の塗りつぶしワード数を変数とした。

①は、全てをCPUで行った場合の処理時間であり、②は、ユーザマイクロプログラムにより行った処理時間で、

・境界点サーチ…各ラインのX座標を求める
(64W)

・塗りつぶし…1ワード単位の塗りつぶし
(64W)

処理時間
(秒)

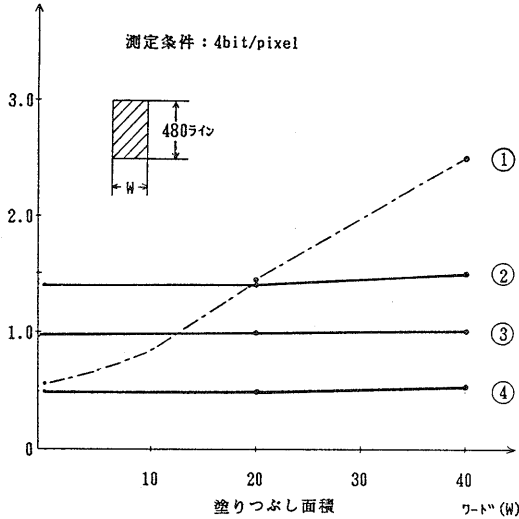


図3 ユーザマイクロプログラムの処理時間

の2種のプログラムで行った。この場合は、各ライン毎に境界点サーチ、塗りつぶしのプログラムをロードする必要がある。③は、②のプログラムを次のように改良した処理時間である。

- ・境界点サーチ... 全ての境界点を発見しメモリに格納する
- ・塗りつぶし... サーチしたすべての境界点の間を塗りつぶす

ここで、各プログラムのロードは、各処理につき1度のみである。④は、③に加えCPUのオーバーヘッドを軽減したもので、DPUへのコマンドの与え方に工夫を加えた場合である。

以上の応用例に示したように、DPUはディスプレイ制御専用プロセッサでありながら、書換え可能制御記憶を内蔵しているため、ユーザのシステムに固有の多様なアプリケーションにも柔軟に対応することができる可能性を有している。

7. あとがき

本稿では、大容量イメージメモリ、および表示用フレームメモリに対するデータ転送処理機能をもつCRTコントローラLSIについて、その機能と構造およびデータ転送制御方式について述べた。

本LSIの特徴は、独立な3系統のバスを利用することにより、イメージメモリ、フレームメモリとのデータ転送およびホストプロセッサとの通信を効率よく行うことを可能としたことである。

また、マイクロプログラムを格納する制御記憶の一部を書換え可能とすることにより、多様なグラフィック処理の基本コマンドを柔軟に実装することができ、従来の機能固定の専用LSIにない大きな特徴を出すことができた。

終わりに、本LSIの開発は、設計、製造、評価の各段階で多くの部署の方々の協力を得て行われたものであり、当社半導体研究センター、松下電子工業(株)、松下電通(株)の関係各位に深く感謝致します。

参考文献

- 1) Hideo Maejima et al.
: VLSI for High Performance Graphic Control Which Utilizes Multi-Processor Architecture, IEEE ICCD'84 (1984)
- 2) 坂村 健: 高機能ワークステーションのアーキテクチャ, 情報処理, Vol.25, NO.2(1984)
- 3) 鈴木則久: 高性能ワークステーションについて, 情報処理, Vol.25, NO.2(1984)
- 4) Newman, W.M. and Sproull, R.F.
: Principles of Interactive Computer Graphics, McGraw-Hill (1979)
- 5) 萩原 宏: マイクロプログラミング, 産業図書 (1977)
- 6) 木下常雄 他: 16ビットSOSマイクロプロセッサT88000, 信学技報EC81-30 (1981)