

## NEC $\mu$ PD7220/72120 Related Public Documents

- [ISSCC](#) (International Solid-State Circuit Conference)
- [\$\mu\$ PD7220A User's Manual](#)
- Nikkei Electronics Magazine ( [\$\mu\$ PD7220](#))
- Transistor Gijutsu (Technology) Magazine ( [\$\mu\$ PD7220](#))
- Transistor Gijutsu (Technology) Magazine ( [\$\mu\$ PD7220A](#))
- Nikkei Electronics Magazine ( [\$\mu\$ PD72120](#))
- [\$\mu\$ PD72120 User's Manual](#)

Go to <https://www.oguchi-rd.com/LSI%20products.php> to get more detailed NEC  $\mu$ PD7220/72120 related information such as;

"Logic Schematics", "Design Notes", "Evaluation Board Schematics", "Evaluation Software", "Silicon Die Photos", "Newspaper", "Magazine", and so forth.

Go to <https://www.oguchi-rd.com/patents.php> to get patent information including NEC  $\mu$ PD7220/72120 related patents.

# $\mu$ PD7220A

## GDCユーザーズ・マニュアル



**NEC** 日本電気株式会社



# 第1章 概 説

$\mu$ PD7220A (グラフィック・ディスプレイ・コントローラ 以下GDCと略)は $\mu$ PD7220を強化改良したもので、ラスタ走査型CRT(陰極線管)に文字・図形を表示するためのLSIです。同期信号発生や文字表示制御機能のほかに、高速グラフィック描画機能や大容量映像メモリ制御機能などをもち、簡単な文字表示制御から高級なフル・グラフィック描画/表示制御や画像表示制御のほか、表示とは無関係な用途として、大容量メモリ制御を必要とする分野に応用することが出来ます。

なお、GDCには $\mu$ PD7220A、 $\mu$ PD7220A-1と $\mu$ PD7220A-2の3種類の製品がありますが、スピードの違いだけで機能的には同一です。

一般のグラフィック装置は、次のようにその制御機能を区分することが出来ます。

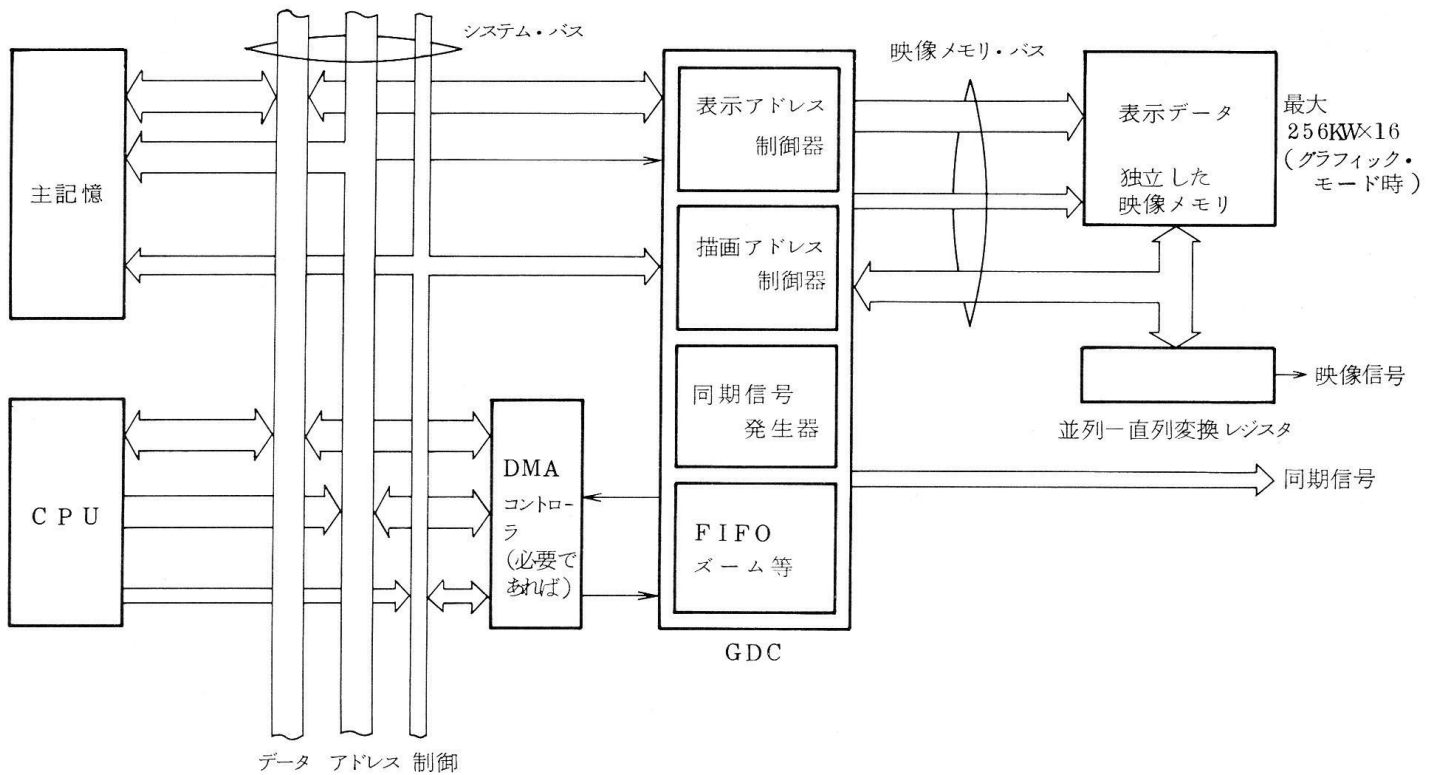
- (1) デジタイザや、あらかじめ作成されている座標リストなどから座標データを取り出したり、ソフトウェアによつて、CPUが座標を発生する。……………〔座標入力〕
- (2) 座標データや動作モード切替情報などを基にして変換処理を行い、よりマイクロなコマンド/パラメータを作成する。……………〔描画前処理〕
- (3) コマンド/パラメータを基にして描画アドレス計算を行う。……………〔描画処理〕
- (4) 算出したアドレスを映像メモリに供給し、データの修正変更動作を行う。……………〔描画実行〕
- (5) 常に、表示アドレス演算をし、算出したアドレスを映像メモリに供給して、表示データを読み出す。……………〔表示〕
- (6) 常に、同期信号を発生し、モニタ・テレビに供給すると共に、表示/描画のタイミング制御をする。……………〔同期信号発生〕

以上の(1)から(6)の機能のうち(3)、(4)、(5)、(6)の機能をGDCは有しており、これらの処理を実行中のとき、GDCはCPUの関与を一切必要としません。

CPUは、(2)の動作、即ち、GDCが解釈できるレベルのコマンド/パラメータの作成や他機器との通信制御などを行い、(3)、(4)の映像メモリに対するドット単位での描画動作はGDCが実行します。(2)、(3)、(4)の動作を並列して同時に実行できるため、ビデオRAM方式やラインバッファ方式と比較して装置の性能が大幅に向上します。

この並列動作を可能とし円滑化するため、CPUが制御するシステム・バスとGDCが制御する映像メモリ・バスとは図1-1のように分離されています。このため、CPUで作成したコマンド/パラメータは、表示/描画のタイミングとは無関係に、随時、GDCに送出できます。さらに、GDCは16個のFIFOを内蔵しているので、GDC内部回路が描画などのために専有されているときであっても、次の描画のために送出されるコマンド/パラメータを受け付けることが出来ます。FIFOに常に描画用コマンド/パラメータが蓄積されている状態にしておけば、GDCはほとんど休むことなく、描画を連続して行います。

図 1-1 システム構成図



CPUとのインタフェースおよび映像メモリとのインタフェースは、図1-2に示すように実現できます。GDCに供給するクロック周波数は、1水平走査期間内の表示時間と、そのときの映像メモリのアクセス回数とによって決定します。映像メモリとしてダイナミックRAMを使用するときに必要な行列アドレス選択信号(RAS, CAS)を発生するための基準タイミング信号( $\overline{RAS}$ ), リフレッシュ・アドレス,  $\overline{DBIN}$ 信号等は、GDCが発生します。

(以下の記述において、描画とは映像メモリ内容の修正変更動作を意味し、表示とは映像メモリ内容の読み出しを意味します。)

GDCは、①描画サイクル、②表示サイクル、③リフレッシュ・サイクルの3種の映像メモリ・アクセス・サイクルを持ち、各種信号は図1-3のようなタイミングで発生します。描画サイクル時には、常にRead/Modify/Write(R/M/W)が実行されます。ダイナミックRAM制御モードでは、HSYNCが“1”のときダイナミックRAMに対するリフレッシュ動作を起すため、リフレッシュ・サイクルに入ります。

GDC内部のブロックを図1-4に示します。CPUから与えられるコマンド/パラメータは、一旦、16語×9ビットのFIFOに格納されます。描画関係の内部回路が動作中でなければ、蓄積されているFIFO内容は読み出され、コマンド制御ROMによってコマンドの解釈がなされ、パラメータは各パラメータ・レジスタへ転送されます。同期信号発生部では、フロント・ポーチ、バック・ポーチを持った水平/垂直同期信号や表示区間信号などを、表示アドレス制御部では表示アドレスやリフレッシュ・アドレスを発生します。描画アドレス制御部では、描画アルゴリズムに従って回路が動作することにより、ドットアドレス・レベルまでの描画アドレスを算出します。映像メモリ・データ制御部ではドット単位での修正変更を行います。

GDCは後述するSYNCコマンドによって、次の3種の動作モードのうちの一つを選択します。

- ① グラフィック・モード
- ② 文字／グラフィック混在モード
- ③ 文字モード

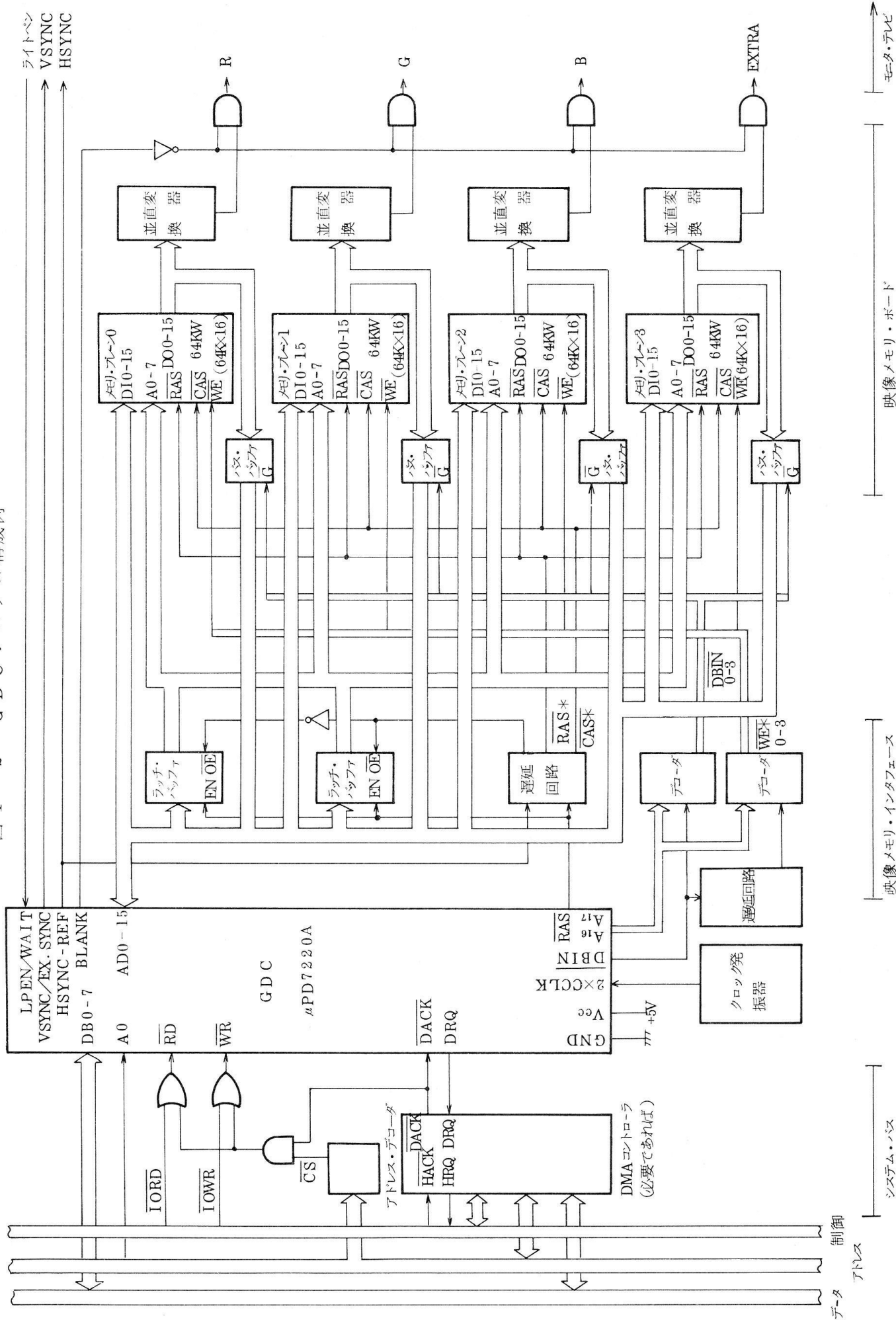
グラフィック・モードでは、最大256K語×16ビットの映像メモリを制御でき、高級なグラフィック装置に適用します。

文字／グラフィック混在モードでは、グラフィック専用としても、また、文字表示専用としても使用でき、グラフィックと文字表示を同一画面上で分割表示することも可能な万能型動作モードで、最大64K語×16ビットの映像メモリを制御できます。512×512以下のような総ドット数で64K語以下の解像度のグラフィック制御のとき、このモードを選択すると有効です。グラフィック・モード時の場合に比して2倍の周波数のクロックを与えることが出来、描画速度が向上します。文字表示専用として用いたとき、外付のライン・カウンタを使用する必要があります。

文字モードでは文字表示専用装置に適用します。最大8K語×13ビットの映像メモリが制御できます。

以下、グラフィック・モードおよび文字／グラフィック混在モードにおいてグラフィックを指定した場合をグラフィック制御、文字モードおよび文字／グラフィック混在モードにおいて文字を指定した場合を文字制御とします。

図 1 - 2 G D C システム構成例



データ  
アドレス  
制御

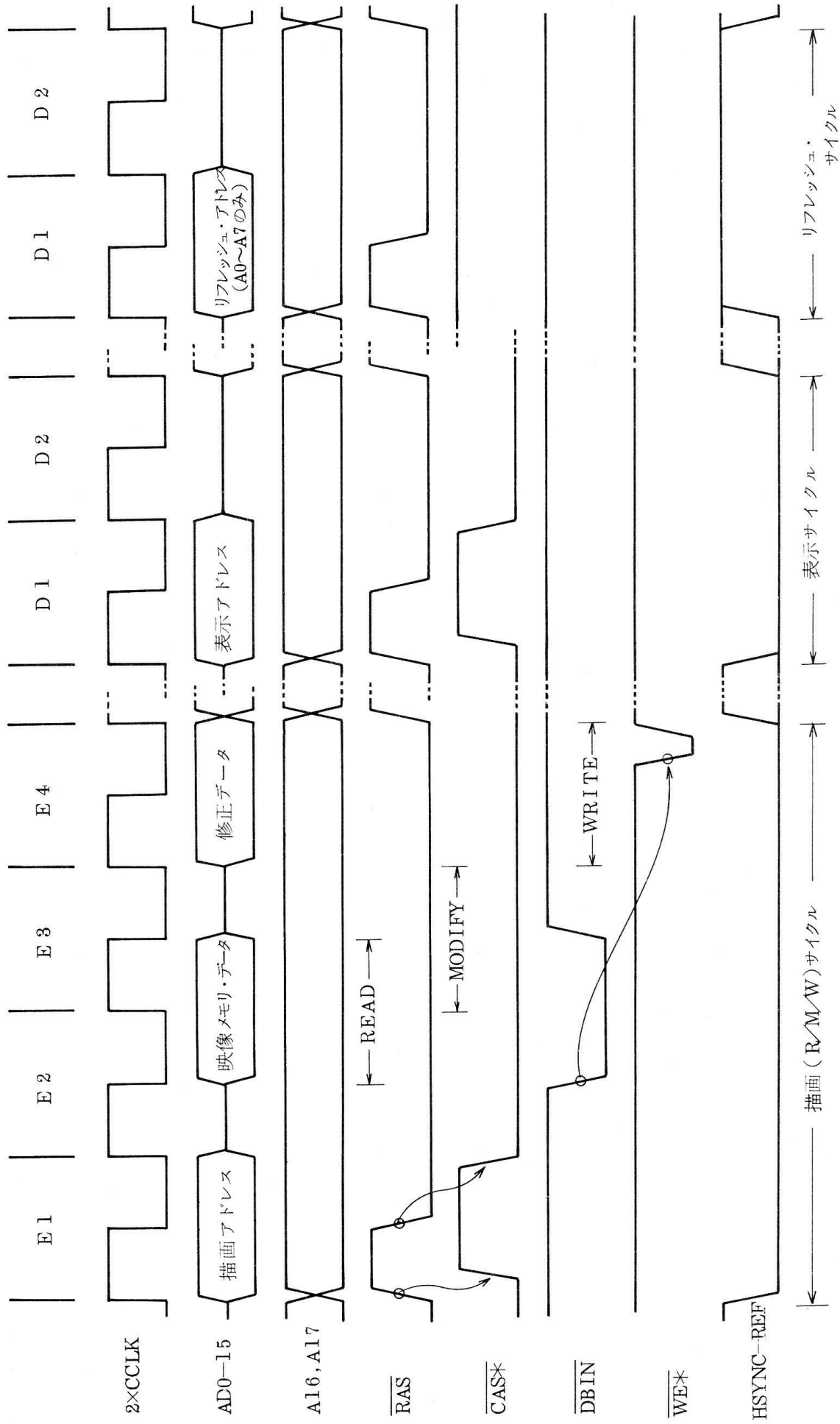
システム・バス  
インタフェース

映像メモリ・インタフェース

映像メモリ・ボード

モニタ・テレビ

図 1 - 3 G D C タイミング



表示およびリフレッシュを除く。  
映像メモリ・アクセスはすべて描画サイクルとなります。

備考 \*：外付け回路によって作成した信号を示す。



## 1.1 特 徴

### (1) 高速グラフィック描画機能

直線，四辺形，円弧，グラフィック文字描画を500ns/ドット（ $\mu$ PD7220A-2：クロック8MHz時）で描画します。この500nsの間に，次に描画すべきドットの描画アドレス計算も同時に実行しており，その結果を基にして次の500nsで次のドットの描画を行います。従って，500ns/ドットの速さで連続的に描画を実行します。さらに，映像メモリ内容を4種のドット修正モード（REPLACE，COMPLEMENT，CLEAR，SET）のうちコマンドによって設定した1モードに従って，修正を加えることができます。またパターン・レジスタの設定により直線，四辺形，円弧描画時に実線，破線，一点鎖線などの任意の線種を選択できます。グラフィック文字描画では，8×8ドット構成の任意なドットパターンを設定することができます。矩形領域の任意パターンによる塗りつぶしが可能です。なお，文字モードにおいても直線，四辺形，円等の描画指定が可能です。

### (2) 主記憶とは独立した大容量メモリ直接制御

最大256K語×16ビット（グラフィック・モード時）の大容量メモリに対して，アドレス/データ/制御の各信号を供給し，メモリ内容のRead/Modify/Writeの動作を直接実行します。メモリが周期的にリフレッシュを必要とするダイナミックRAMの場合には，リフレッシュに必要な信号を出力します。主記憶のメモリ・マップに乗らない独立したメモリを制御するため，バスの衝突に注意を払う必要がありません。また，主記憶からの制限を受けないために拡張性に富み，メモリは用途に応じて任意な構成とすることが出来ます。たとえば，2048×2048 1枚プレーンによるモノクローム表示や，1024×1024 4枚プレーンによる16色表示など種々に選択できます。

### (3) 16×9ビットの入出力FIFO内蔵

コマンド/パラメータを16個まで蓄積し，GDCの描画速度とCPUのコマンド/パラメータ作成速度の調節をします。後述されるREADなどの読み出しコマンド実行後は，映像メモリ内容または内部レジスタ内容を一時的に格納します。

### (4) メモリーメモリ間DMA転送機能

映像メモリと主記憶間のDMA転送を4クロック/バイト（以下特にことわらない限りクロックは2×CCLKをさす）で連続的にバースト転送実行できます。映像メモリに対する書き込み/読み出しの両方向とも可能であり，実時間で，バイト/ワードの変換を行います。また，アドレスが連続した1次元領域に対するDMAだけでなく，アドレス不連続部を含む2次元矩形領域に対する選択的DMAも可能です。

### (5) 外部同期機能

GDCを同期信号のマスタあるいはスレーブとして設定し用いることが出来ます。スレーブ設定をすると外部からの同期信号を受け付け，同期信号発生器を初期化できます。従って，1個のシステム内で，GDCを多数個互いに同期が取れた状態で並列動作させることが出来ます。更に，他の同期信号発生器を持つ映像機器との同期を取ることも可能です。

### (6) 描画タイミングの選択機能

表示期間中以外に描画を行うモード（フラッシュレス・モード）と表示期間中にも描画を行うモード（フラッシュ・モード）の選択が出来ます。



(7) 表示領域と映像メモリの水平方向の大きさ定義

表示領域と映像メモリの水平方向の大きさ定義をそれぞれ独立して設定できます。従って、映像メモリの一部だけを表示させることが出来ます。

(8) 拡大表示／拡大描画，パニング，スクロール

1 から 16 までの整数倍での拡大表示およびグラフィック文字描画における拡大描画が可能です。表示開始アドレスを変更するだけで、垂直方向ライン（ドット）単位、水平方向ワード単位でのスクロールやパニング（全方向へのスクロール）が可能です。簡単な外付回路を使用すると、水平方向ドット単位スクロールも実現できます。

その他、以下に示すような特徴を有しています。

(9) 自動カーサ・シフト（文字制御時）

左から右への水平方向だけでなく、右から左へ、または垂直方向などにも、1文字入力後、自動的にカーサ位置が移動します。

(10) 1文字単位でのアトリビュート設定（文字制御時）

表示反転、点滅などの文字属性（アトリビュート）を領域単位の設定ではなく1文字毎に設定できます。

(11) 256文字／行，100行／画面

(12) インタレース／ノン・インタレース走査の選択可

(13) プログラマブルな同期信号設定，カーサ形状

(14) 最大クロック周波数

μPD7220A ..... 6MHz（描画 660ns／ドット）

μPD7220A-1 ..... 7MHz（描画 580ns／ドット）

μPD7220A-2 ..... 8MHz（描画 500ns／ドット）

(15) NチャンネルMOS，+5V単電源，40ピンDIP

## 1.2 μPD7220AとμPD7220の違い

μPD7220AはμPD7220を強化改良したものです。以下にμPD7220Aの強化点をμPD7220と対比して説明します。

μPD7220A	μPD7220																																																																								
① リセット・コマンド																																																																									
<p>リセット・コマンドが3種類になり、初期化動作の他に、次の指示を行います。</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th></th> <th style="text-align: center;">表示の 停止/開始</th> <th style="text-align: center;">外部同期信号の受 け付け停止/開始</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">RESET 1</td> <td style="text-align: center;">停 止</td> <td style="text-align: center;">開 始</td> </tr> <tr> <td style="text-align: center;">RESET 2</td> <td style="text-align: center;">停 止</td> <td style="text-align: center;">停 止</td> </tr> <tr> <td style="text-align: center;">RESET 3</td> <td style="text-align: center;">開 始</td> <td style="text-align: center;">停 止</td> </tr> </tbody> </table> <p>RESET 1 コマンドはμPD7220のRESETコマンドとコードおよび指示内容は同じです。 RESET 2 コマンドは外部同期動作を実行せずに初期化動作のみを行いたいときに使用します。 RESET 3 コマンドは表示を消さないで、初期化動作のみを行いたいときに使用します。</p>		表示の 停止/開始	外部同期信号の受 け付け停止/開始	RESET 1	停 止	開 始	RESET 2	停 止	停 止	RESET 3	開 始	停 止	<p>RESETコマンドは初期化動作、表示の停止、および外部同期信号の受け付け開始を指示します。</p>																																																												
	表示の 停止/開始	外部同期信号の受 け付け停止/開始																																																																							
RESET 1	停 止	開 始																																																																							
RESET 2	停 止	停 止																																																																							
RESET 3	開 始	停 止																																																																							
② ストップ・コマンド																																																																									
<p>ストップ・コマンドが2種類になり、表示の停止の他に、次の指示を行います。</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th></th> <th style="text-align: center;">外部同期信号の受 け付け停止/開始</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">STOP 1</td> <td style="text-align: center;">開 始</td> </tr> <tr> <td style="text-align: center;">STOP 2</td> <td style="text-align: center;">停 止</td> </tr> </tbody> </table> <p>STOP 1 コマンドはμPD7220のSTOPコマンドとコードおよび指示内容は同じです。 STOP 2 コマンドは外部同期動作を実行せずに表示のみを消したいときに使用します。</p>		外部同期信号の受 け付け停止/開始	STOP 1	開 始	STOP 2	停 止	<p>STOPコマンドは表示の停止、および外部同期信号の受け付け開始を指示します。</p>																																																																		
	外部同期信号の受 け付け停止/開始																																																																								
STOP 1	開 始																																																																								
STOP 2	停 止																																																																								
③ PITCHレジスタ																																																																									
<table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th colspan="4">SYNC</th> <th colspan="4">P 5</th> </tr> <tr> <th>DB7</th><th>DB6</th><th>DB5</th><th>DB4</th><th>DB3</th><th>DB2</th><th>DB1</th><th>DB0</th> </tr> </thead> <tbody> <tr> <td>DS</td><td>PH</td><td colspan="6" style="text-align: center;">← HBP →</td> </tr> </tbody> </table> <p>PITCHレジスタを1ビット増やし、1行中の文字数の定義を0文字(00H)から511文字(1FFH)まで設定できます。</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>P<sub>H</sub></th> <th>P<sub>L</sub></th> <th>文字数/行</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0 0 0 0 0 0 0 0</td> <td style="text-align: center;">0</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0 0 0 0 0 0 0 1</td> <td style="text-align: center;">1</td> </tr> <tr> <td style="text-align: center;">⋮</td> <td style="text-align: center;">⋮</td> <td style="text-align: center;">⋮</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1 1 1 1 1 1 1 1</td> <td style="text-align: center;">255</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0 0 0 0 0 0 0 0</td> <td style="text-align: center;">256</td> </tr> <tr> <td style="text-align: center;">⋮</td> <td style="text-align: center;">⋮</td> <td style="text-align: center;">⋮</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">1 1 1 1 1 1 1 1</td> <td style="text-align: center;">511</td> </tr> </tbody> </table>	SYNC				P 5				DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	DS	PH	← HBP →						P <sub>H</sub>	P <sub>L</sub>	文字数/行	0	0 0 0 0 0 0 0 0	0	0	0 0 0 0 0 0 0 1	1	⋮	⋮	⋮	0	1 1 1 1 1 1 1 1	255	1	0 0 0 0 0 0 0 0	256	⋮	⋮	⋮	1	1 1 1 1 1 1 1 1	511	<table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th colspan="4">SYNC</th> <th colspan="4">P 5</th> </tr> <tr> <th>DB7</th><th>DB6</th><th>DB5</th><th>DB4</th><th>DB3</th><th>DB2</th><th>DB1</th><th>DB0</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td><td style="text-align: center;">0</td><td colspan="6" style="text-align: center;">← HBP →</td> </tr> </tbody> </table> <p>1行中の文字数の定義は0文字(00H)から255文字(FFH)まで設定できます。</p>	SYNC				P 5				DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	0	0	← HBP →					
SYNC				P 5																																																																					
DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0																																																																		
DS	PH	← HBP →																																																																							
P <sub>H</sub>	P <sub>L</sub>	文字数/行																																																																							
0	0 0 0 0 0 0 0 0	0																																																																							
0	0 0 0 0 0 0 0 1	1																																																																							
⋮	⋮	⋮																																																																							
0	1 1 1 1 1 1 1 1	255																																																																							
1	0 0 0 0 0 0 0 0	256																																																																							
⋮	⋮	⋮																																																																							
1	1 1 1 1 1 1 1 1	511																																																																							
SYNC				P 5																																																																					
DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0																																																																		
0	0	← HBP →																																																																							

$\mu$ PD7220A	$\mu$ PD7220																																																																																																
④ 描画実行中の表示アドレス出力																																																																																																	
<table border="1" style="width: 100%; text-align: center;"> <tr> <td colspan="4">SYNC</td> <td colspan="4">P 5</td> </tr> <tr> <td>DB7</td><td>DB6</td><td>DB5</td><td>DB4</td><td>DB3</td><td>DB2</td><td>DB1</td><td>DB0</td> </tr> <tr> <td>DS</td><td>PH</td><td colspan="4">← HBP →</td><td colspan="2"></td> </tr> </table> <p>DSビットの設定によりLPEN/WAIT入力端子を4クロック以上Hレベルにすると、<math>\mu</math>PD7220Aが描画実行中であっても、その描画アドレス出力を一時停止させ、表示アドレスを出力します。</p> <p>DS</p> <p>0：描画の一時停止不可（<math>\mu</math>PD7220と同じ）</p> <p>1：描画の一時停止可</p>	SYNC				P 5				DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	DS	PH	← HBP →						<table border="1" style="width: 100%; text-align: center;"> <tr> <td colspan="4">SYNC</td> <td colspan="4">P 5</td> </tr> <tr> <td>DB7</td><td>DB6</td><td>DB5</td><td>DB4</td><td>DB3</td><td>DB2</td><td>DB1</td><td>DB0</td> </tr> <tr> <td>0</td><td>0</td><td colspan="4">← HBP →</td><td colspan="2"></td> </tr> </table> <p>描画実行の一時停止機能はありません。</p>	SYNC				P 5				DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	0	0	← HBP →																																																					
SYNC				P 5																																																																																													
DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0																																																																																										
DS	PH	← HBP →																																																																																															
SYNC				P 5																																																																																													
DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0																																																																																										
0	0	← HBP →																																																																																															
⑤ VBLANKステータス出力																																																																																																	
<table border="1" style="width: 100%; text-align: center;"> <tr> <td colspan="4">SYNC</td> <td colspan="4">P 6</td> </tr> <tr> <td>DB7</td><td>DB6</td><td>DB5</td><td>DB4</td><td>DB3</td><td>DB2</td><td>DB1</td><td>DB0</td> </tr> <tr> <td>VH</td><td>VL</td><td colspan="4">← VFP →</td><td colspan="2"></td> </tr> </table> <p>VHビットの設定により、DB6から出力されるステータスを垂直消去時間（VBLANK）と水平消去時間（HBLANK）のいずれか選択できます。</p> <p>VH</p> <p>0：HBLANKステータス（<math>\mu</math>PD7220と同じ）</p> <p>1：VBLANKステータス</p>	SYNC				P 6				DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	VH	VL	← VFP →						<table border="1" style="width: 100%; text-align: center;"> <tr> <td colspan="4">SYNC</td> <td colspan="4">P 6</td> </tr> <tr> <td>DB7</td><td>DB6</td><td>DB5</td><td>DB4</td><td>DB3</td><td>DB2</td><td>DB1</td><td>DB0</td> </tr> <tr> <td>0</td><td>0</td><td colspan="4">← VFP →</td><td colspan="2"></td> </tr> </table> <p>DB6からは水平消去時間（HBLANK）を示すステータスが出力されます。</p>	SYNC				P 6				DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	0	0	← VFP →																																																					
SYNC				P 6																																																																																													
DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0																																																																																										
VH	VL	← VFP →																																																																																															
SYNC				P 6																																																																																													
DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0																																																																																										
0	0	← VFP →																																																																																															
⑥ インタレース・シュリンク時の1フレームあたりの表示本数																																																																																																	
<table border="1" style="width: 100%; text-align: center;"> <tr> <td colspan="4">SYNC</td> <td colspan="4">P 6</td> </tr> <tr> <td>DB7</td><td>DB6</td><td>DB5</td><td>DB4</td><td>DB3</td><td>DB2</td><td>DB1</td><td>DB0</td> </tr> <tr> <td>VH</td><td>VL</td><td colspan="4">← VFP →</td><td colspan="2"></td> </tr> </table> <p>VLビットの設定により、インタレース・シュリンク時の1フレームあたりの表示本数を定義できます。</p> <p>VL</p> <p>0：奇数本（<math>\mu</math>PD7220と同じ）</p> <p>1：偶数本</p>	SYNC				P 6				DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	VH	VL	← VFP →						<table border="1" style="width: 100%; text-align: center;"> <tr> <td colspan="4">SYNC</td> <td colspan="4">P 6</td> </tr> <tr> <td>DB7</td><td>DB6</td><td>DB5</td><td>DB4</td><td>DB3</td><td>DB2</td><td>DB1</td><td>DB0</td> </tr> <tr> <td>0</td><td>0</td><td colspan="4">← VFP →</td><td colspan="2"></td> </tr> </table> <p>インタレース・シュリンク時の1フレームあたりの表示本数は奇数本です。</p>	SYNC				P 6				DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	0	0	← VFP →																																																					
SYNC				P 6																																																																																													
DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0																																																																																										
VH	VL	← VFP →																																																																																															
SYNC				P 6																																																																																													
DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0																																																																																										
0	0	← VFP →																																																																																															
⑦ WRITEコマンドによる映像メモリへの書き込み																																																																																																	
<p>文字／グラフィック混在モード</p> <table border="1" style="width: 100%; text-align: center;"> <tr> <td colspan="4">CSRW</td> <td colspan="4">P 3</td> </tr> <tr> <td>DB7</td><td>DB6</td><td>DB5</td><td>DB4</td><td>DB3</td><td>DB2</td><td>DB1</td><td>DB0</td> </tr> <tr> <td colspan="4">← dAD →</td><td>WG</td><td>0</td><td>0</td><td>0</td> </tr> </table> <p>グラフィック・モード</p> <table border="1" style="width: 100%; text-align: center;"> <tr> <td colspan="4">CSRW</td> <td colspan="4">P 3</td> </tr> <tr> <td>DB7</td><td>DB6</td><td>DB5</td><td>DB4</td><td>DB3</td><td>DB2</td><td>DB1</td><td>DB0</td> </tr> <tr> <td colspan="4">← dAD →</td><td>WG</td><td>0</td><td colspan="2">EAD<sub>H</sub></td> </tr> </table> <p>WGビットの設定により、グラフィック・モード時または文字／グラフィック混在モードでDGD=1の時でも、文字モードと同様に、WRITEコマンドによりパラメータ設定値をそのままの形で映像メモリに書き込むことができます。</p> <p>WG</p> <p>0：書き込み不可能（<math>\mu</math>PD7220と同じ）</p> <p>1：書き込み可能</p>	CSRW				P 3				DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	← dAD →				WG	0	0	0	CSRW				P 3				DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	← dAD →				WG	0	EAD <sub>H</sub>		<p>文字／グラフィック混在モード</p> <table border="1" style="width: 100%; text-align: center;"> <tr> <td colspan="4">CSRW</td> <td colspan="4">P 3</td> </tr> <tr> <td>DB7</td><td>DB6</td><td>DB5</td><td>DB4</td><td>DB3</td><td>DB2</td><td>DB1</td><td>DB0</td> </tr> <tr> <td colspan="4">← dAD →</td><td>0</td><td>0</td><td>0</td><td>0</td> </tr> </table> <p>グラフィック・モード</p> <table border="1" style="width: 100%; text-align: center;"> <tr> <td colspan="4">CSRW</td> <td colspan="4">P 3</td> </tr> <tr> <td>DB7</td><td>DB6</td><td>DB5</td><td>DB4</td><td>DB3</td><td>DB2</td><td>DB1</td><td>DB0</td> </tr> <tr> <td colspan="4">← dAD →</td><td>0</td><td>0</td><td colspan="2">EAD<sub>H</sub></td> </tr> </table> <p>グラフィック・モード時または文字／グラフィック混在モードでDGD=1の時には、文字モードのようにWRITEコマンドによって、映像メモリに書き込むことはできません。</p>	CSRW				P 3				DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	← dAD →				0	0	0	0	CSRW				P 3				DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	← dAD →				0	0	EAD <sub>H</sub>	
CSRW				P 3																																																																																													
DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0																																																																																										
← dAD →				WG	0	0	0																																																																																										
CSRW				P 3																																																																																													
DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0																																																																																										
← dAD →				WG	0	EAD <sub>H</sub>																																																																																											
CSRW				P 3																																																																																													
DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0																																																																																										
← dAD →				0	0	0	0																																																																																										
CSRW				P 3																																																																																													
DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0																																																																																										
← dAD →				0	0	EAD <sub>H</sub>																																																																																											

$\mu$ PD7220A	$\mu$ PD7220																																																
⑧ スレーブ時の外部同期信号 (EX, SYNC) の受け付け																																																	
<table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th colspan="4">CSRFORM</th> <th colspan="4">P 1</th> </tr> <tr> <th>DB7</th><th>DB6</th><th>DB5</th><th>DB4</th><th>DB3</th><th>DB2</th><th>DB1</th><th>DB0</th> </tr> </thead> <tbody> <tr> <td>CS</td><td>CE</td><td>0</td><td colspan="4" style="text-align: center;">← L / R →</td><td></td> </tr> </tbody> </table> <p>CEビットの設定により、内部同期信号発生回路を初期化する条件を定義できます。</p> <p>CE</p> <p>0 : 表示停止中のEX, SYNC信号の立ち下がり (<math>\mu</math>PD7220と同じ)</p> <p>1 : EX, SYNCの立ち下がり</p>	CSRFORM				P 1				DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	CS	CE	0	← L / R →					<table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th colspan="4">CSRFORM</th> <th colspan="4">P 1</th> </tr> <tr> <th>DB7</th><th>DB6</th><th>DB5</th><th>DB4</th><th>DB3</th><th>DB2</th><th>DB1</th><th>DB0</th> </tr> </thead> <tbody> <tr> <td>CS</td><td>0</td><td>0</td><td colspan="4" style="text-align: center;">← L / R →</td><td></td> </tr> </tbody> </table> <p>表示を停止しているときのみ、EX, SYNC信号の立ち下がりによって内部同期信号発生回路を初期化します。</p>	CSRFORM				P 1				DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	CS	0	0	← L / R →				
CSRFORM				P 1																																													
DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0																																										
CS	CE	0	← L / R →																																														
CSRFORM				P 1																																													
DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0																																										
CS	0	0	← L / R →																																														
⑨ 文字表示におけるカーサ位置検出																																																	
表示アドレス16ビット全てについて行います。	表示アドレスの下位13ビットについてのみ行っています。																																																
⑩ 文字表示におけるカーサ形状定義																																																	
これに関する制約条件はありません。	画面の1行目にカーサを表示する場合には $CFI \neq L/R$ とする必要があります。																																																
⑪ 表示アドレス出力A 16, A 17																																																	
描画サイクル中のE 4 タイミングでも不定とならず、スタティック信号を出力します。	描画サイクル中のE 4 タイミングで不定となります。																																																
⑫ 拡大表示時のRAS信号																																																	
拡大表示時のRAS信号のHレベルが、HBLANK時の立ち上がり時に、通常時に比べ、 $\frac{1}{2}$ クロックだけ伸びる現象は起きません。	拡大表示時のRAS信号のHレベルが、HBLANK時の立ち上がり時に、通常時に比べ、 $\frac{1}{2}$ クロックだけ伸びます。																																																
⑬ 表示アドレスのインクリメント																																																	
1画面終了後のHFP (画面右下隅) においても、また垂直帰線消去期間においても、表示アドレスのインクリメントを行います。	1画面終了後のHFP (画面右下隅)、また、垂直帰線消去期間においては、表示アドレスのインクリメントは行いません。																																																
⑭ 最大クロック周波数																																																	
8MHz ----- 描画500 ns / ドット ( $\mu$ PD7220A-2)	5.5MHz ----- 描画720ns / ドット ( $\mu$ PD7220 2)																																																

## 1.3 略称一覧

本マニュアルで使用する略称とその意味を以下に示します。

略 称	意 味	略 号	意 味
AD	アドレス・データ	KW	キロ語
BLANK	表示消去信号	LAD	ライトペン・アドレス
BLINK	表示点滅信号	LC	ライン・カウンタ
CAS	列アドレス選択信号	L/F	ライン数/画面
CCLK	文字クロック	LPEN	ライトペン
CFI	カーサ表示終了	L/R	ライン数/行
CPU	中央処理ユニット	M	メガ(10 <sup>6</sup> )
C/R	文字数/行	MOD	ドット修正モード
CRT	陰極線管	P	ピッチ
CSR	カーサ	PTN	パターン
CST	カーサ表示開始	RA	内蔵RAMアドレス
DACK	DMAアクリッジ信号	RAM	ランダム・アクセス・メモリ
DAD	表示アドレス	RAS	行アドレス選択信号
dAD	ドット・アドレス	RD	読み出し信号
DB	データ・バス	REF	リフレッシュ
DBIN	データ・バス・イン信号	REM	剰余部
DI	データ入力端子	R/M/W	リード/モディファイ/ライト
DIR	方向	SAD	表示開始アドレス
D1	描画用パラメータ	SL	走査線数
D2	描画用パラメータ	SYNC	同期
D	描画用パラメータ	TEXT	グラフィック文字
DC	描画用パラメータ	TV	テレビジョン
DM	描画用パラメータ	VC	垂直カウンタ
DMA	ダイレクト・メモリ・アクセス	VBP	垂直バック・ポーチ
DO	データ出力端子	VS	垂直同期レジスタ
DRAM	ダイナミックRAM	VFP	垂直フロント・ポーチ
DRQ	DMA要求	VSYNC	垂直同期信号
EAD	描画実行アドレス	WE	ライト・イネーブル信号
FIFO	First-in First-out	WR	書き込み信号
GDC	グラフィック・ディスプレイ・コントローラ	WLH	ワード/下位バイト/上位バイト
H	水平期間	XM	X座標最大値
HBP	水平バック・ポーチ	YM	Y座標最大値
HC	水平カウンタ	ZR	拡大表示
HFP	水平フロント・ポーチ	ZW	拡大描画
HS	水平同期レジスタ		
HSYNC	水平同期信号		

# 第2章 端子

SYNC コマンドによって定義する表示/描画動作モード設定に従って端子機能は次のように変化します。

端子番号	グラフィック・モード	文字モード	文字/グラフィック混在モード
35	AD13 (I/O)	LC0 (O)	AD13 (I/O)
36	AD14 (I/O)	LC1 (O)	AD14 (I/O)
37	AD15 (I/O)	LC2 (O)	AD15 (I/O)
38	A16 (O)	LC3 (O)	$\overline{AT.BLINK-CLC}$ (O)
39	A17 (O)	CSR-LC4 (O)	CSR-IMAGE (O)

MASTER/SLAVE コマンドによる動作設定に従って、端子機能は次のように変化します。

マスタ動作	スレーブ動作
VSYNC (O)	EX.SYNC (I)

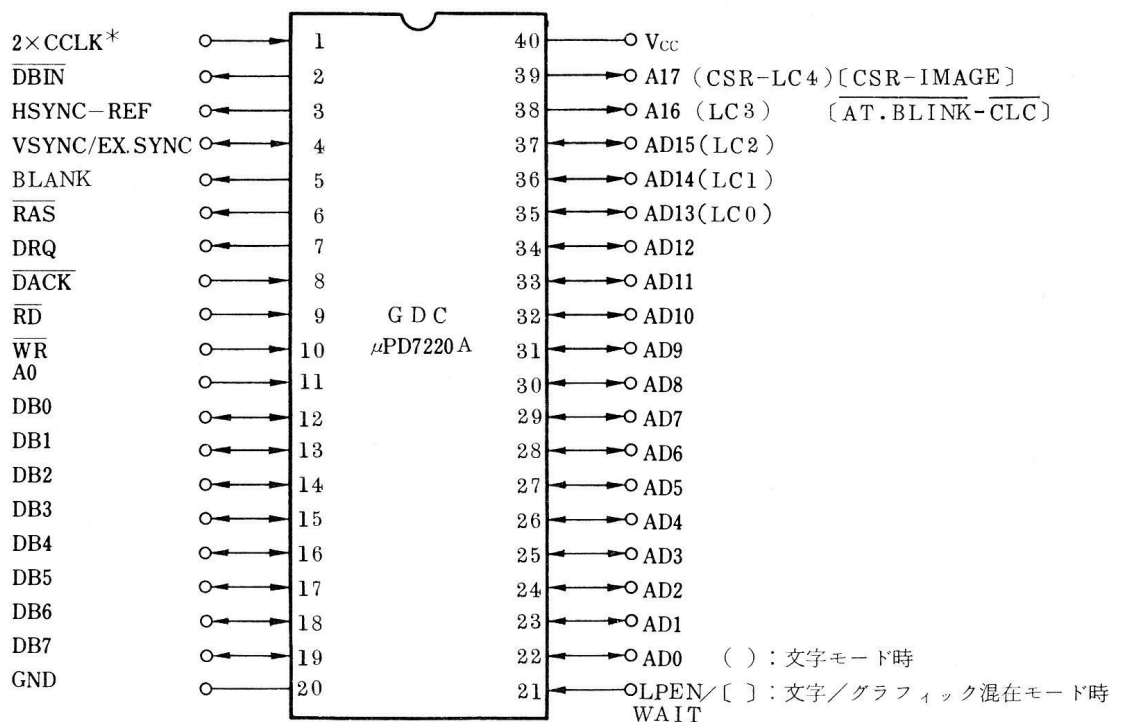
括弧内は、それぞれ (I/O) : 入出力兼用端子

(O) : 出力端子

(I) : 入力端子

を意味します。

## 2.1 端子接続



\* : この端子にはプルアップが必要です。

## 2.2 端子機能

### (1) 電源，グランド端子

#### (i) Vcc

+5Vの電源を接続します。

#### (ii) GND

0Vの接地用端子です。

### (2) クロック端子

#### (i) 2×CLK (2×Character Clock)

単相クロックを供給します。

クロックの周波数は，水平方向の表示文字数（表示ドット数）と水平表示時間との関係で決まります。また，この端子のみTTLコンパチブルではありませんので，必要に応じてVccへプルアップしてください。

### (3) システム・バス接続用端子

#### (i) DB0～DB7 (CPU Data Bus 0～7)

双方向性のデータ・バスで8ビットまたは16ビットの標準的なマイクロプロセッサに接続します。

#### (ii) $\overline{RD}$ (CPU Read Strobe)

CPUがGDCからデータまたはステータス・フラグを読み出すとき，Lレベル信号を供給します。

#### (iii) $\overline{WR}$ (CPU Write Strobe)

CPUがGDCにコマンドまたはパラメータを書き込むとき，Lレベル信号を供給します。

#### (iv) A0 (CPU Address Bus 0)

通常，CPUのアドレス・ライン最下位ビットが接続され， $\overline{RD}$ ， $\overline{WR}$ 信号との組み合わせにより，以下のように，データ・バス信号の種類を選択します。

A0	$\overline{RD}$	$\overline{WR}$	機 能
0	0	1	GDCステータス・フラグの読み出し
1	0	1	データ(GDC内FIFO)の読み出し
0	1	0	パラメータの書き込み
1	1	0	コマンドの書き込み

GDCはCS(チップ・セレクト)端子を持っていませんので，外部回路によってCSおよび $\overline{DACK}$ の信号を $\overline{RD}$ および $\overline{WR}$ 信号に組み合わせる必要があります。

また，上表のような機能を得るには後述の $\overline{DACK}$ はHレベルでなければなりません。



#### (4) DMA制御用端子

##### (i) DRQ (DMA Request)

DMA要求出力で、DMAコントローラのDRQ入力に接続します。DMA RまたはDMA Wコマンドの実行によって活性化されます。

##### (ii) $\overline{DACK}$ (DMA Acknowledge)

DMA転送中であることを示す入力で、DMAコントローラの $\overline{DACK}$ 出力が接続されます。DMAによる転送バイト数のカウントや転送開始のタイミング指示用として使用します。プルアップ抵抗が内蔵されています。

#### (5) 映像メモリ制御用端子

以下、表示動作モードによって端子機能が変わる端子については、別個に分けて説明します。

##### (i) AD0~AD15 (Display Memory Address/Data Bus 0~15)

双方向性のアドレス/データ・バスで、映像メモリに接続します。アドレスとデータが時分割されていますので、RAS出力のHレベル期間にアドレスを外部回路にラッチします。また、ダイナミックRAM制御モードのリフレッシュ・サイクル時には、リフレッシュ・アドレスが下位8ビットに出力されます。

##### (ii) A16, A17 (Display Memory Address 16, 17)

グラフィック・モード時、アドレス上位2ビットが出力され、映像メモリに接続します。

##### (iii) $\overline{RAS}$ (Row Address Strobe)

この信号には以下の3種の用途があります。

- ① アドレス多重化を行っている16ピン・ダイナミックRAMに対する $\overline{RAS}$ 、 $\overline{CAS}$ 信号を作るための基本タイミング信号。
- ② Hレベル時、AD0~AD15に対するアドレス・ラッチ・タイミング信号。
- ③ 映像メモリのアドレス・サイクルの弁別用タイミング信号。

より具体的には、映像メモリから並列に読み出された映像データを直列信号に変換するレジスタのLOADクロック発生タイミング選択用として使用します。

##### (iv) $\overline{DBIN}$ (Data Bus In)

映像メモリに対するRead/Modify/Write実行時にのみ出力され、映像メモリの出力を映像メモリのデータ・バスに乗せるために使用します。この信号を外部回路によって遅延させ、映像メモリに対する書き込みタイミング信号(WE)としても使用します。

#### (6) 同期信号発生用端子

##### (i) HSYNC-REF (Horizontal SYNC-Refresh Timing)

水平同期信号としてモニタ・テレビに接続します。ダイナミックRAM制御モードを選択した場合には、リフレッシュ・アドレスがAD0~AD7に出力されていることを示すタイミング信号としても使用されます。

##### (ii) VSYNC (Vertical SYNC)

GDCがマスタ動作時に、垂直同期信号としてモニタ・テレビに接続します。プル

アップ抵抗が内蔵されています。

(iii) EX. SYNC ( External SYNC )

GDCがスレーブ動作時に、外部同期入力端子として使用します。この信号の "H" → "L" への変化時に、GDC内部の同期信号発生回路が初期化されます。プルアップ抵抗が内蔵されています。

(7) 表示制御用端子

(i) BLANK ( Blanking Signal )

表示消去信号として使用します。以下の状態のときに出力されます。

- ① 水平帰線または垂直帰線区間
- ② RESET, STOPなどの表示停止コマンド実行時からSTARTコマンド実行時までの区間
- ③ 映像メモリに対するR/M/W実行区間 ( 描画サイクル )

(ii) LPEN ( Light-pen Strobe )

ライトペンが光入力を検出したときHレベルの信号を入力します。そのときにGDC内にラッチされた表示アドレスをLPENコマンドによって、CPUは読み出すことができます。プルダウン抵抗が内蔵されています。

(iii) WAIT ( Drawing-wait )

描画停止動作の設定されている場合、最低4クロック間Hレベルとなる信号を入力すると、GDCは描画実行中であっても、その描画を一時停止して表示アドレスを出力します。プルダウン抵抗が内蔵されています。

(iv) LC0~LC3 ( Line Count 0~3 )

文字モード時、GDCに内蔵されている5ビットのライン・カウンタのうち下位4ビットが出力されます。文字発生用ROMのアドレスに接続します。

(v) CSR-LC4 ( Cursor-Line Count 4 )

文字モード時、カーサ表示出力と内蔵ライン・カウンタの最上位ビットを時分割して出力します。表示期間中にはカーサ表示信号を出力します。LC4の信号はHSYNC出力信号の立ち下がり時に外部フリップ・フロップに記憶させます。

(iv)  $\overline{\text{AT.BLINK}}-\overline{\text{CLC}}$  ( Attribute Blink-Clear Line Counter )

文字/グラフィック混在モード時、プリンキングする文字属性に供給するプリンキング・タイミング信号と外部回路によって構成するライン・カウンタに対するクリア信号を時分割して出力します。表示期間中にはプリンキング・タイミング信号を出力します。外付ライン・カウンタはHSYNC-REF出力信号の立ち下がり時にカウントすると同時に、 $\overline{\text{CLC}}$ 信号の抽出を行います。

(vi) CSR-IMAGE ( Cursor-Image )

文字/グラフィック混在モード時、カーサ表示出力とイメージ信号出力を時分割して出力します。表示期間中にはカーサ表示信号を出力します。イメージ信号は、SCROLLコマンドにより設定されたIMビットの状態を外部表示制御回路に指示する目的の信号で、BLANK出力信号の立ち下がり時に、外部フリップ・フロップに記憶させます。

# 第3章 コマンド

GDCには、以下に示す24種のコマンドが用意されています。

## 動作制御コマンド

コマンド	動作内容	C/P	コ ー ド							
			DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
RESET 1	初期化動作、表示の停止、外部同期信号受け付け開始の指示	C	0	0	0	0	0	0	0	0
RESET 2	初期化動作、表示の停止、外部同期信号の受け付け停止の指示	C	0	0	0	0	0	0	0	1
RESET 3	初期化動作、表示の開始、外部同期信号の受け付け停止の指示	C	0	0	0	0	1	0	0	1
SYNC	動作モード、同期信号波形の定義	C P1 P2 P3 P4 P5 P6 P7 P8	0 0	0 0	0 CHR	0 F	1 I	1 D	1 G	DE S
			<p>Timing diagram for SYNC command:</p> <ul style="list-style-type: none"> <li>P2: C/R (Control/Reset)</li> <li>P3: VS<sub>L</sub> (Vertical Sync Low)</li> <li>P3: HS (Horizontal Sync)</li> <li>P4: HFP (Horizontal Front Porch)</li> <li>P4: VS<sub>H</sub> (Vertical Sync High)</li> <li>P5: DS (Data Start), P<sub>H</sub> (Horizontal Sync)</li> <li>P5: HBP (Horizontal Back Porch)</li> <li>P6: VH (Vertical Sync High), VL (Vertical Sync Low)</li> <li>P6: VFP (Vertical Front Porch)</li> <li>P7: L/F<sub>L</sub> (Line/Front Porch Low)</li> <li>P7: L/F<sub>H</sub> (Line/Front Porch High)</li> <li>P8: VBP (Vertical Back Porch)</li> <li>P8: L/F<sub>H</sub> (Line/Front Porch High)</li> </ul>							
MASTER SLAVE	マスタ動作、スレーブ動作の選択	C	0	1	1	0	1	1	1	M

## 表示制御コマンド

コマンド	動作内容	C/P	コ ー ド							
			DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
START <sup>(1)</sup>	表示の開始の指示	C	0	1	1	0	1	0	1	1
			0	0	0	0	1	1	0	1
STOP1	表示の停止, 外部同期信号の受け付け開始の指示	C	0	0	0	0	1	1	0	0
STOP2	表示の停止, 外部同期信号の受け付け停止の指示	C	0	0	0	0	0	1	0	1
ZOOM	拡大表示係数, 拡大描画係数の設定	C	0	1	0	0	0	1	1	0
		P1	← ZR →				← ZW →			
SCROLL	表示開始アドレス, 表示領域の設定	C	0	1	1	1	← RA →			
		P1	内蔵データRAMのフォーマットに従ってパラメータを設定							
		P16								
CSRFORM	文字表示時のカーサ形状等の設定	C	0	1	0	0	1	0	1	1
		P1	CS	CE	0	← L/R →				
		P2	← BL <sub>L</sub> →		BD	← CST →				
		P3	← CFI →				← BL <sub>H</sub> →			
PITCH	映像メモリ水平方向ワード数の設定	C	0	1	0	0	0	1	1	1
		P1	← PL →							
LPEN <sup>(2)</sup>	ライトペン・アドレスの読み出し指示	C	1	1	0	0	0	0	0	0
			← LAD <sub>L</sub> →							
			← LAD <sub>M</sub> →							
			×	×	×	×	×	×	← LAD <sub>H</sub> →	

注意1 コマンド・コードは6BHまたは0DHの8ビットを使用します。

注意2 LPENコマンドの発行後LAD<sub>L</sub>, LAD<sub>M</sub>, LAD<sub>H</sub>の順にCPUが読み出すことができます。

描画制御コマンド

コマンド	動作内容	C/P	コード								
			DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
VECTW	描画に必要な各種パラメータの設定	C	0	1	0	0	1	1	0	0	
		P1	SL	R	C	T	L	← DIR →			
		P2	← DC <sub>L</sub> →								
		P3	×	DGD	← DC <sub>H</sub> →						
		P4	← D <sub>L</sub> →								
		P5	×	×	← D <sub>H</sub> →						
		P6	← D2 <sub>L</sub> →								
		P7	×	×	← D2 <sub>H</sub> →						
		P8	← D1 <sub>L</sub> →								
		P9	×	×	← D1 <sub>H</sub> →						
		P10	← DM <sub>L</sub> →								
P11	×	×	← DM <sub>H</sub> →								
VECTE	直線, 四辺形, 円弧, 1ドット描画の実行の指示	C	0	1	1	0	1	1	0	0	
TEXTW	グラフィック・テキスト・コード設定	C	0	1	1	1	1	← RA' →			
		P1	← TX8, PTN <sub>L</sub> →								
		P2	← TX7, PTN <sub>H</sub> →								
		P3	← TX6 →								
		P4	← TX5 →								
		P5	← TX4 →								
		P6	← TX3 →								
		P7	← TX2 →								
		P8	← TX1 →								
TEXTE	グラフィック・テキスト描画実行指示	C	0	1	1	0	1	0	0	0	
CSRW	描画アドレスの設定	文字モード	C	0	1	0	0	1	0	0	1
			P1	← EAD <sub>L</sub> →							
		P2	0	0	0	← EAD <sub>H</sub> →					
		文字/グラフィック混在モードで文字表示/描画	C	0	1	0	0	1	0	0	1
			P1	← EAD <sub>L</sub> →							
		P2	← EAD <sub>H</sub> →								
	文字/グラフィック混在モードでグラフィック表示/描画	C	0	1	0	0	1	0	0	1	
		P1	← EAD <sub>L</sub> →								
		P2	← EAD <sub>H</sub> →								
	P3	← dAD →				WG	0	0	0		
	グラフィック・モード	C	0	1	0	0	1	0	0	1	
		P1	← EAD <sub>L</sub> →								
P2		← EAD <sub>M</sub> →									
P3		← dAD →				WG	0	← EAD <sub>H</sub> →			
CSRR (3)	描画アドレスの読み出し指示	C	1	1	1	0	0	0	0	0	
			← EAD <sub>L</sub> →								
			← EAD <sub>M</sub> →								
			×	×	×	×	×	×	← EAD <sub>H</sub> →		
			← dAD <sub>L</sub> →								
			← dAD <sub>H</sub> →								
MASK	マスク・レジスタ値の設定	C	0	1	0	0	1	0	1	0	
		P1	← MASK <sub>L</sub> →								
		P2	← MASK <sub>H</sub> →								

注意3 CSRRコマンドの発行後EAD<sub>L</sub>, EAD<sub>M</sub>, EAD<sub>H</sub>, dAD<sub>L</sub>, dAD<sub>H</sub>の順にCPUが読み出すことができます。

映像メモリ制御コマンド

コマンド	動作内容	C/P	コ ー ド							
			DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
WRITE	パラメータの映像メモリへの書き込み準備	C P1 ⋮	0	0	1	←WLH→		0	←MOD→	
READ	映像メモリ・データの読み出し指示	C	1	0	1	←WLH→		0	←MOD→	
DMAW	映像メモリへのDMA転送開始の指示	C	0	0	1	←WLH→		1	←MOD→	
DMAR	映像メモリからのDMA転送開始の指示	C	1	0	1	←WLH→		1	←MOD→	

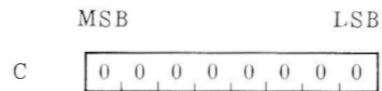
以下，コマンドあるいはパラメータ・コードを示す記号は次の略号を示します。

C : コマンド

P : パラメータ

### 3.1 動作制御コマンド

#### 3.1.1 RESET1コマンド



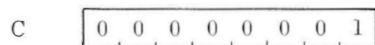
次のような初期化動作を行います。

- ① 内部基本タイミング発生回路，同期信号発生回路，ベクタ・カウンタ回路のクリア。
- ② F I F Oアドレスのクリア，F I F O書き込みモード設定。
- ③ 描画動作，D M A転送動作の停止。
- ④ 表示の停止。
- ⑤ 外部同期信号の受け付け開始

G D CはRESET入力端子を持っていないので，初期化動作はRESET1コマンドを送出することによって実行します。RESET1コマンドがG D C内に書き込まれると， $\overline{WR}$ 入力信号の立ち上がりから2クロック後までの間に初期化動作を終了します。

また，RESET1コマンドに続いてS Y N Cコマンドを実行する場合は，S Y N Cコマンドのコマンド・コード(C)を省略することができます。即ち，RESET1コマンドを実行すると，初期化動作を行うと共にコマンドがF I F Oに蓄積され，続いてパラメータ(P 1 ~ P 8)を送出するとG D CはS Y N Cコマンドとして実行します。

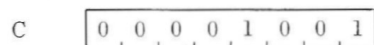
#### 3.1.2 RESET2コマンド



次の点を除き，RESET1コマンドと同様の初期化動作を行います。

- ⑤ 外部同期信号の受け付け停止

#### 3.1.3 RESET3コマンド



次の点を除き，RESET1コマンドと同様の初期化動作を行います。

- ④ 表示の開始
- ⑤ 外部同期信号の受け付け停止



### 3.1.4 SYNCコマンド

C 0 0 0 0 1 1 1 DE

P 1 0 0 CHR F I D G S

P 2 C/R

P 3 VSL HS

P 4 HFP VSH

P 5 DS PH HBP

P 6 VH VL VFP

P 7 L/FL

P 8 VBP L/FLH

#### (1) 表示動作モード等の定義

##### ① DE (Display Enable)

表示の開始/停止を定義します。

DE	機 能
0	表示を停止する
1	表示を開始する

##### ② CHR, G

次のような組合せで表示/描画動作モードを定義します。

CHR	G	モ ー ド
0	0	文字/グラフィック混在モード
1	0	文字モード
0	1	グラフィック・モード
1	1	設定できません

##### ③ F

描画タイミングを選択します。

F	機 能
0	フラッシュ描画
1	フラッシュレス描画

フラッシュレス描画をすると、表示領域とリフレッシュ期間を除いた期間に描画を行うように、自動的に描画タイミングを制御します。このため表示領域ではCRT面上にフラッシ

ユを発生しません(図3-2参照)。

スタティックRAM制御モード( $D=0$ )を指定した場合にはHSYNC期間(図3-1のHS)においても描画を実行します。

表示領域かどうかの判定はブランキング信号をもとにして行っていますので、STOP1またはSTOP2などの表示消去コマンドによって全面ブランキング状態にして描画を行ったときは、フラッシュレス描画選択時であってもリフレッシュ期間を除いた全期間にわたって描画を実行することができます。

フラッシュ描画モードを実行したときは表示期間においても描画を実行します。このためCRT面上にフラッシュ(ちらつき)が発生しますが、描画時間はフラッシュレス描画モード選択時と比較して約5倍速めることができます。描画実行時にはフラッシュによる目の疲労を軽減するため、ブランキング信号が出力され表示を消去します。

図3-1 表示タイミング

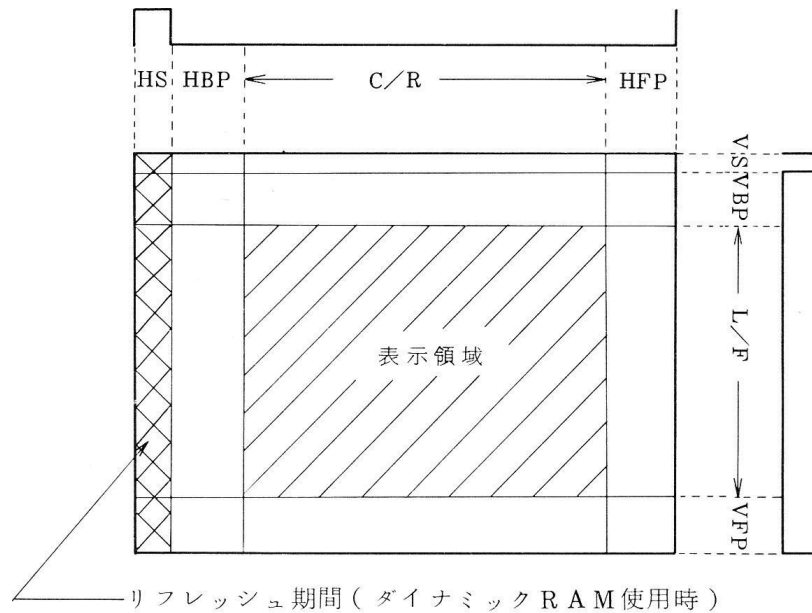
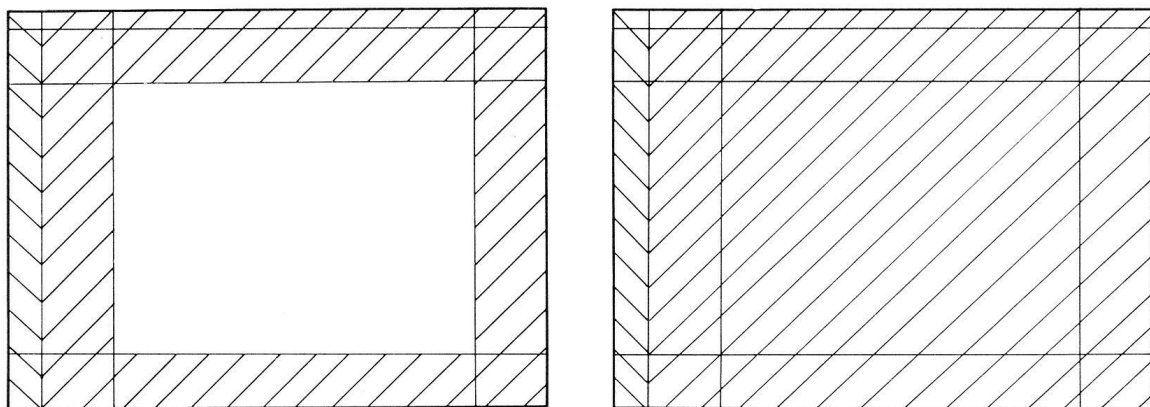


図3-2 描画可能期間



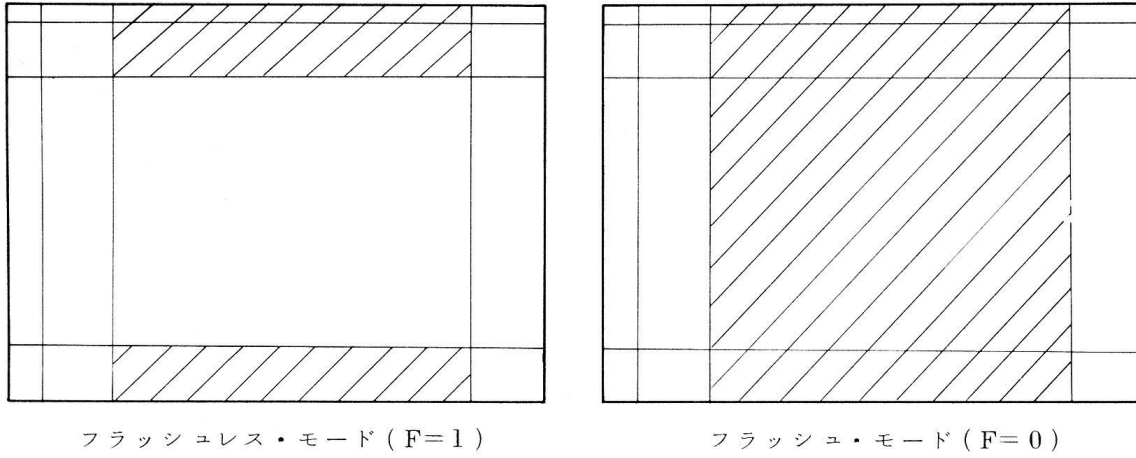
フラッシュレス・モード ( $F=1$ )

フラッシュ・モード ( $F=0$ )

▨ : ダイナミックRAM制御モード選択時 ( $D=1$ ) の描画可能期間

▨ : スタティックRAM制御モード選択時 ( $D=0$ ) に追加される描画可能期間

図 3-3 DMA 要求発生タイミング



フラッシュレス・モード (F=1)

フラッシュ・モード (F=0)

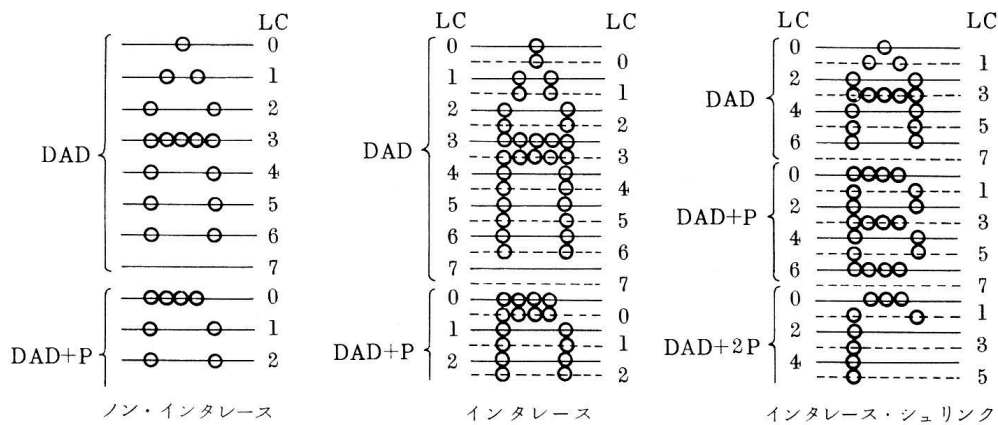
④ I, S

インタレース走査の有無およびライン・カウンタや表示アドレスの進み方の形態を設定します。

I	S	機能
0	0	ノン・インタレース
0	1	設定できません
1	0	インタレース
1	1	インタレース・シュリンク

走査線の軌跡およびライン・カウンタの進み方は図 3-4 のようになります。

図 3-4 走査線とライン・カウンタの関係



インタレースおよびインタレース・シュリンクの場合において、

破線 : 第 1 フィールドの走査線の軌跡

実線 : 第 2 フィールドの走査線の軌跡

を示します。

DAD, DAD+Pなどは, 表示アドレスの進み方を示します.

\* 4.8 インタレース時のL/Fの設定法を参照

なお, 文字/グラフィック混在モードで画面分割を行い, かつ文字表示とグラフィック表示を混在させた場合には, インタレース・シュリンク・モードの使用はできません. 従って, このような場合にはインタレースもしくはノン・インタレース・モードを使用してください.

⑤ D

使用する映像メモリ素子の種別からリフレッシュ動作の有無を定義します.

D	機能
0	スタティックRAM制御モード, リフレッシュ動作無し
1	ダイナミックRAM制御モード, リフレッシュ動作有り

リフレッシュ動作は2クロックで1アドレスのリフレッシュが行われますので, SYNCコマンドのHSパラメータで設定した文字の数だけリフレッシュされます.

\* 6.3 メモリ・インタフェースを参照

⑥ PH

1行中の文字数を格納するPITCHレジスタの9ビット目の値を定義します(詳細については3.2.6 PITCHコマンドを参照してください).

⑦ DS

LPEN/WAIT端子への4クロック以上のHレベル入力により, GDCが描画動作中であっても, 描画アドレスの出力を一時停止して表示アドレスを出力するか否かを定義します.

DS	機能
0	描画動作の一時停止無し
1	描画動作の一時停止有り

⑧ VH

DB6端子から読み出すステータスを定義します.

VH	機能
0	水平消去信号(HBLANK)
1	垂直消去信号(VBLANK)

⑨ VL

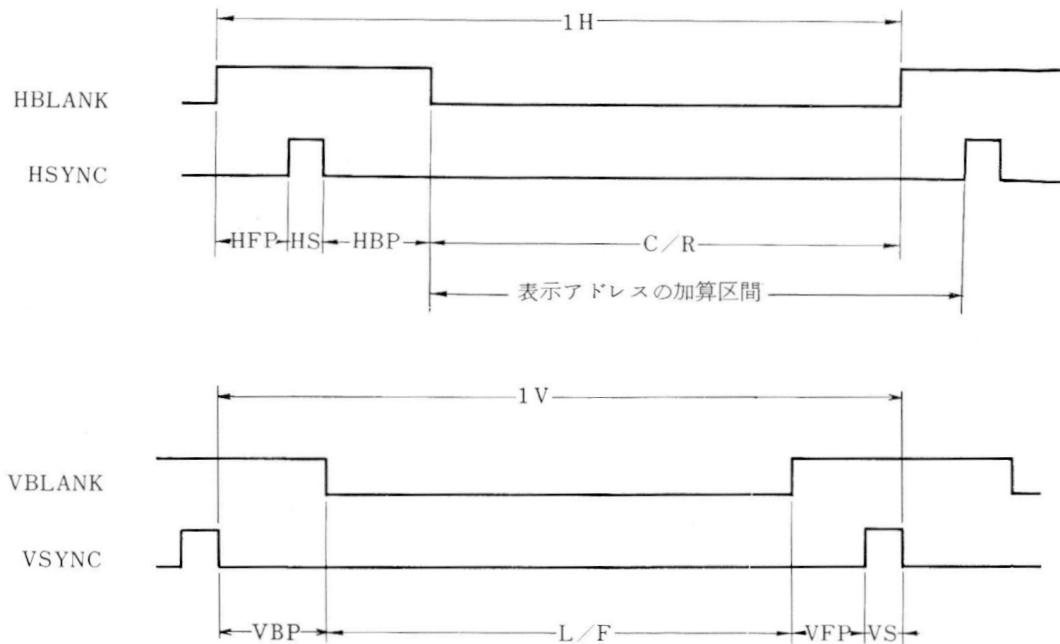
インタレース・シュリンク設定時の1フレームあたりの表示本数を定義します.

VL	機能
0	奇数本
1	偶数本

(2) 水平/垂直同期信号および表示/非表示区間の定義

図 3-5 に示す波形の値を定義します。

図 3-5 同期信号と表示区間



⑩ C/R ( Characters/Row )

1 行当りの表示文字数を設定し、水平方向の表示区間を定義します。

C/R	文字数
00000000	2
00000010	4
00000100	6
00000110	8
.....	.....
11111010	252
11111100	254
11111110	256

注意 1 文字表示区間とは 2 クロック期間を意味します。

C/R としては " 2 " から " 256 " までの偶数を設定します。奇数は設定できません。

この場合、後述する PITCH コマンドによって設定される映像メモリの横幅 P とは  $P \geq C/R$  の関係にします。ただし、後述の SCROLL コマンドの IM ビットまたは DAD + 2 ビットが 1 になっている場合には、 $P \geq 2 \times C/R$  となるように設定します。

また、このパラメータを設定すると PITCH も同値に設定されますので、PITCH コマンドは SYNC コマンド後に設定してください。

⑪ H S (Horizontal SYNC)

水平同期信号の幅を定義します。

H S	文 字 数
0 0 0 0 0	1
0 0 0 0 1	2
0 0 0 1 0	3
0 0 0 1 1	4
-----	-----
1 1 1 0 1	30
1 1 1 1 0	31
1 1 1 1 1	32

1文字とは $2 \times \text{CCLK}$ の2クロック区間を示します。

注意 使用状態により次の制約があります。

- |   |                               |
|---|-------------------------------|
| (a) D R A Mに対するリフレッシュを必要とする場合               | リフレッシュ周期を満足する<br>値であり、かつ2文字以上 |
| (b) インタレース・モードを設定した場合                       | 3文字以上                         |
| (c) インタレース・シュリンク・モードにおいて<br>1行／ラインの表示を行う場合  |                               |
| (i) A 1 6 , A 1 7のアドレス線を表示のために使<br>用する場合    | 5文字以上                         |
| (ii) A 1 6 , A 1 7のアドレス線を表示のためには<br>使用しない場合 | 4文字以上                         |

⑫ HFP (Horizontal Front Porch)

CRT管面右方の非表示区間を定義します。

HFP	文字数
000000	1*
000001	2
000010	3
000011	4
.....	.....
111101	62
111110	63
111111	64

\*: 設定できません。ただし、表示以外の目的に使用する場合を除きます。

1文字とは $2 \times \text{CCLK}$ の2クロック区間を示します。

注意 使用状態により次の制約があります。

- |                                   |  |
|-----------------------------------|--|
| (a) 通常の表示を行う場合                    | 2文字以上  |
| (b) スレーブ・モードを設定した場合               | {CE=0 4文字以上<br>CE=1 5文字以上  |
| (c) ライトペンを使用する場合                  | n+1文字以上 n: 外部ライトペン・インタフェース回路における遅延と表示アドレスが出力されてからその内容がCRT上に表示されるまでの遅延の和を文字数換算したもの。 |
| (d) DMA転送を行う場合                    | 3文字以上  |
| (e) インタレース・シュリンクのモードにおいて拡大表示を行う場合 | 3文字以上  |

⑬ HBP (Horizontal Back Porch)

CRT管面左方の非表示区間を定義します。

HBP	文字数
000000	1
000001	2
000010	3
000011	4
.....	.....
111101	62
111110	63
111111	64

1文字とは $2 \times \text{CCLK}$ の2クロック区間を示します。

注意 使用状態により次の制約があります。

- |   |       |
|---|-------|
| (a) 通常使用時   | 3文字以上 |
| (b) 画面分割を行う場合であって   |       |
| (i) インタレース・シュリンクのモードで使用する<br>とき   | 5文字以上 |
| (ii) 画面分割を行った各表示領域を設定する<br>SCROLLコマンドのIMビットが異なるか、<br>または、DAD+2ビットが異なるとき | 5文字以上 |



⑭ V S ( Vertical SYNC )

垂直同期信号の幅を定義します。

VS <sub>H</sub>	VS <sub>L</sub>	ライン数
0 0 0 0 0		設定できません
0 0 0 0 1		1
0 0 0 1 0		2
0 0 0 1 1		3
.....		.....
1 1 1 0 1		29
1 1 1 1 0		30
1 1 1 1 1		31

ライン数とは水平走査線数を示しますので、1ラインの時間=1H(図3-5参照)となります。

⑮ V F P ( Vertical Front Porch )

CRT管面下方の非表示区間を定義します。

VFP	ライン数
0 0 0 0 0 0	設定できません
0 0 0 0 0 1	1
0 0 0 0 1 0	2
0 0 0 0 1 1	3
.....	.....
1 1 1 1 0 1	61
1 1 1 1 1 0	62
1 1 1 1 1 1	63

ライン数とは水平走査線数を示します。

インタレース・シュリンクのモードにおいて、VL=1としたときは偶数ライン数を設定してください。

⑯ V B P ( Vertical Back Porch )

CRT管面上方の非表示区間を定義します。

VBP	ライン数
0 0 0 0 0 0	設定できません
0 0 0 0 0 1	1
0 0 0 0 1 0	2
0 0 0 0 1 1	3
.....	.....
1 1 1 1 0 1	61
1 1 1 1 1 0	62
1 1 1 1 1 1	63

ライン数とは水平走査線数を示します。

⑰ L/F (Lines/Field)

1画面当たり(インタレースの場合は第1フィールド)\*の表示ライン数を設定し、垂直方向の表示区間を定義します。

L/F <sub>H</sub>	L/F <sub>L</sub>	ライン数
0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	1024
0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 1	1
0 0 0 0 0 0 0 0 0 1	0 0 0 0 0 0 0 0 0 1	2
0 0 0 0 0 0 0 0 0 1	0 0 0 0 0 0 0 0 1 1	3
.....	.....	.....
1 1 1 1 1 1 1 1 0 1	1 1 1 1 1 1 1 1 0 1	1021
1 1 1 1 1 1 1 1 1 0	1 1 1 1 1 1 1 1 1 0	1022
1 1 1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1 1 1	1023

ライン数とは水平走査線数を示します。

インタレース・シュリンクのモードにおいて、VL=0としたときは偶数ライン数を設定してください。

\* 4.8 インタレース時のL/Fの設定を参照

以上のように、水平同期関係については1文字表示区間(2クロック)単位で、垂直同期関係については1ライン(1H:1水平走査期間)単位で任意な値に設定することが出来ます。

### 3.1.5 MASTER/SLAVE コマンド

C 

0	1	1	0	1	1	1	M
---	---	---	---	---	---	---	---

マスタ動作かスレーブ動作の選択をします。

M	機 能
0	スレーブ動作
1	マスタ動作

\* 4.5 外部同期機能を参照。

## 3.2 表示制御コマンド

### 3.2.1 STARTコマンド

C 

0	1	1	0	1	0	1	1
---	---	---	---	---	---	---	---

または

C 

0	0	0	0	1	1	0	1
---	---	---	---	---	---	---	---

表示区間中のBLANK出力をLレベルとして表示開始の指示をします。上記2種のコマンドは同一の機能を持っています。第2のコマンドは、最下位ビットの変更のみで表示の開始と停止を指定できるためSTOP1コマンドと対にして使用するとき有用です。

STARTコマンドは表示途中においても実行されます。ただし、BLANK出力信号がHレベルの場合にはSCROLLコマンドによる画面制御は行われませんので、任意の時刻にこのコマンドを実行させますと、最大1フィールド分画面が乱れます。このためSTARTコマンドの実行はVBLANK中に行うとよいでしょう。

### 3.2.2 STOP1コマンド

C 

0	0	0	0	1	1	0	0
---	---	---	---	---	---	---	---

BLANK出力信号を常時、Hレベルとし、表示停止および外部同期信号の受け付け開始の指示をします。

### 3.2.3 STOP2コマンド

C 

0	0	0	0	0	1	0	1
---	---	---	---	---	---	---	---

BLANK出力信号を常時、Hレベルとし、表示停止および外部同期信号の受け付け停止の指示をします。

### 3.2.4 ZOOMコマンド

C 

0	1	0	0	0	1	1	0
---	---	---	---	---	---	---	---

P 

←	ZR	→	←	ZW	→
---	----	---	---	----	---

表示時の拡大係数(ZR)およびグラフィック文字描画時の拡大係数(ZW)の値を設定します。

Z R / Z W	倍 数
0 0 0 0	1
0 0 0 1	2
0 0 1 0	3
-----	-----
1 1 0 1	14
1 1 1 0	15
1 1 1 1	16

\* 4.4 拡大機能を参照

### 3.2.5 SCROLLコマンド

表示開始アドレスおよび画面分割表示領域の大きさを示すライン数を主に設定します。

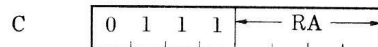
コマンド・コード下位4ビットの" R A "は内蔵データRAMのアドレスを示し、第1パラメータを格納するアドレスを与えます。パラメータの入力毎に、RAMアドレスは自動的にインクリメントされますので、連続したアドレスに格納するパラメータを与える場合には、コマンドの送出は1度だけで可能です。また" R A "を任意の値に設定できるので、内蔵RAMの任意なアドレスから書き換えることができます。

このSCROLLコマンドによって画面分割表示領域をダイナミックに変化させることができますが、表示領域中でSCROLLコマンドを使用しますと、画面が乱れる場合があります。従って、SCROLLコマンドの実行は、STARTコマンドと同様に、VBLANK中に行うとよいでしょう。

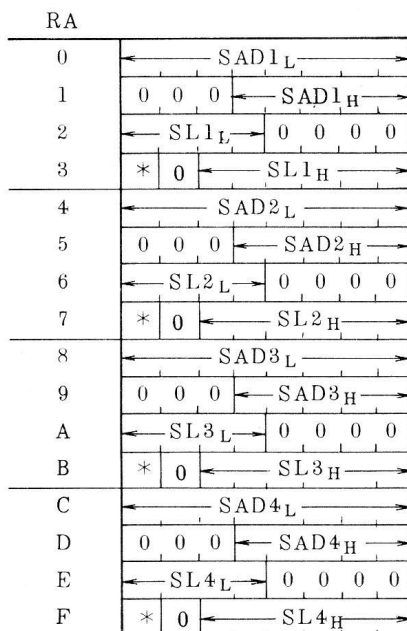
\* 4.2.2 スクロール，画面分割の方法を参照

#### (1) 文字モードの場合

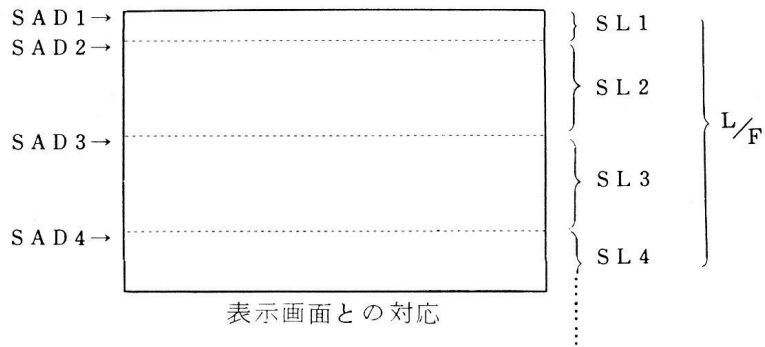
4種に画面を分割可能です。



内蔵RAMマップ



\* : D A D + 2



ただし SAD : 表示開始アドレス

SL : 画面分割表示領域の大きさを示すライン数で、  
SLの総和はL/Fに等しいか、より大きくしてください。

SL	ライン数
0 0 0 0 0 0 0 0 0 0	1024
0 0 0 0 0 0 0 0 0 1	1
0 0 0 0 0 0 0 0 1 0	2
-----	----
-----	----
1 1 1 1 1 1 1 1 1 0	1022
1 1 1 1 1 1 1 1 1 1	1023

ライン数とは水平走査線数を示します。

グラフィック描画を行う場合には、データRAMの上位8バイト(08~0FH番地)が、  
実線、破線等の線種設定や、グラフィック文字描画用のドット構成情報として使用されます  
ので、画面分割数は最大2となります。

① DAD+2

表示アドレスのインクリメント形態を定義します(図3-6参照)。

DAD+2	機 能
0	"1"によるインクリメント。 DAD+1 → DAD
1	"2"によるインクリメント。 DAD+2 → DAD(以下DAD+2 モードと略)

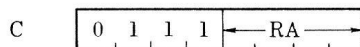
DAD : Display Address

この場合、後述のPITCHコマンドにより設定されるピッチPの値の設定には注意が必要  
です。

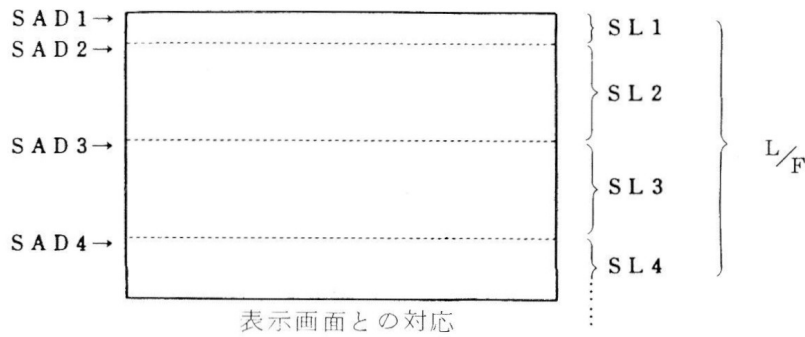
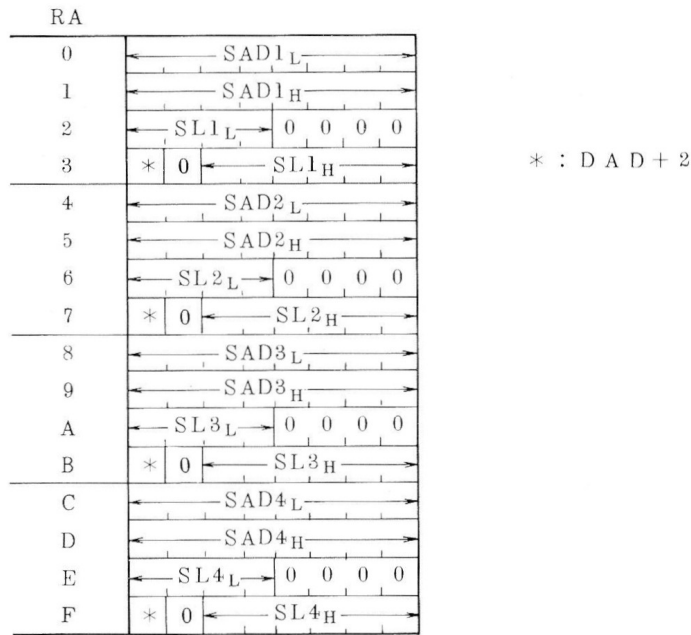
また、DAD+2モードにおけるCSR出力のタイミングは図3-6のようになります。

(2) 文字/グラフィック混在モードで文字表示のみを行う場合

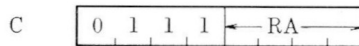
4種に画面を分割可能です。



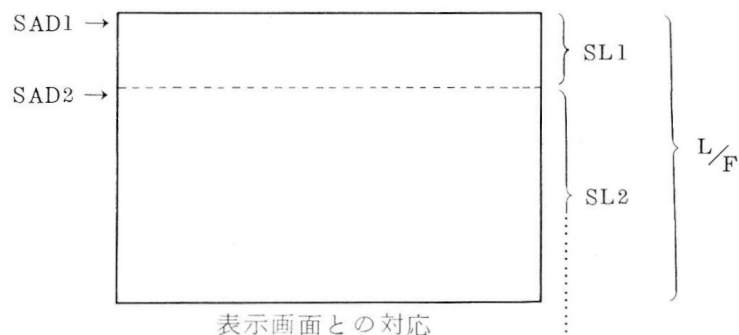
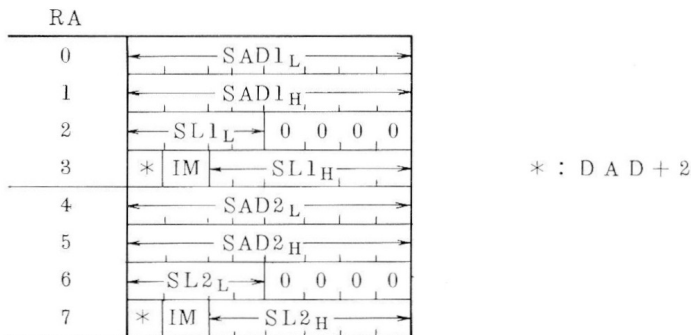
内蔵RAMマップ



- (3) 文字/グラフィック混在モードでグラフィック表示/描画動作を行う場合  
画面分割数は2種となります。



内蔵RAMマップ



① I M

表示アドレスのインクリメントするタイミングを定義します(図3-6参照)。

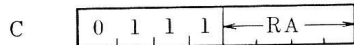
I M	表示制御	L/R制御
0	2クロックに1回表示アドレスをインクリメントします。	CSRFORM コマンドの設定値がそのまま使用されます。
1	4クロックに1回表示アドレスをインクリメントします(アドレス・サイクルは2クロックですが、同じアドレスが2度出力されます)。また、表示アドレスの計算に使用するピッチPはPITCHコマンドによる設定値の1/2になりますので、設定値は実際のPITCHの文字数の2倍の値としてください。	CSRFORM コマンドの設定値にかかわらずL/R=0を設定した場合と同じ動作をします。 (ただし、コマンドによって設定されたL/Rレジスタの内容は変化しません)。

このため文字モードまたは文字/グラフィック混在モードの文字領域においては、I M=1を設定することができません。

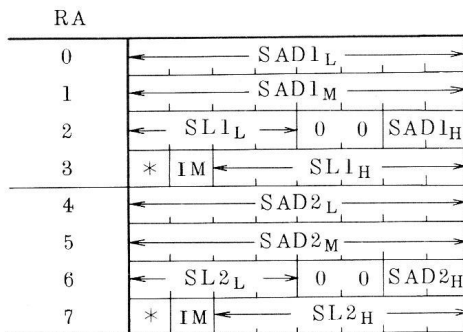
また、文字/グラフィック混在モードで使用した場合には、この設定に基づく切り換え制御信号がCSR-IMAGE 端子より出力されます。

(4) グラフィック・モードの場合

画面分割数は2種となります。



内蔵RAMマップ



\* : D A D + 2

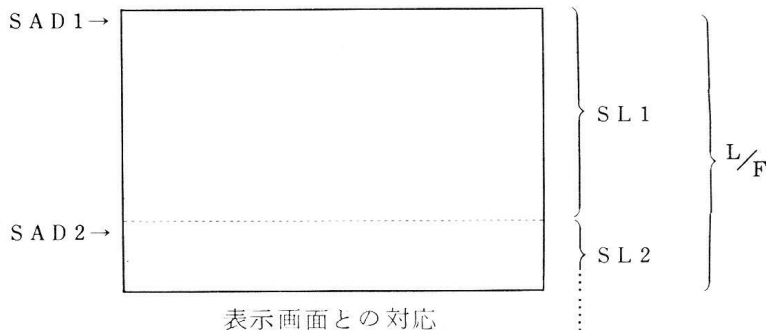
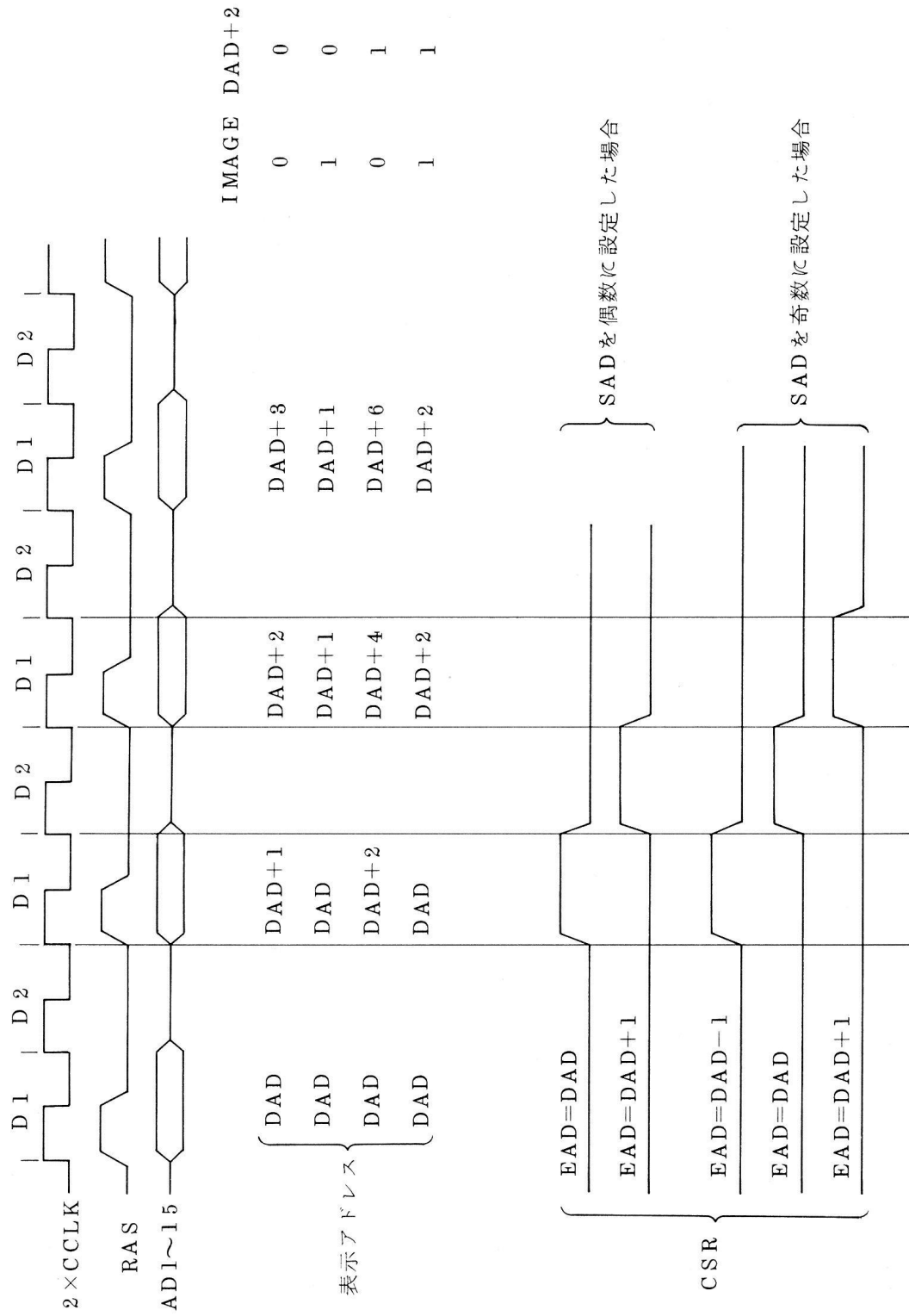


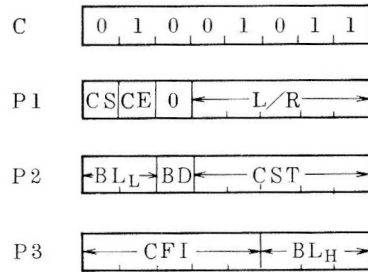
図 3-6 表示アドレスの変化と DAD+2 モードにおける CSR 出力信号





### 3.2.6 CSRFORMコマンド

ライン・カウントの上限値および文字表示時のカーサ表示形態を定義します。



① L/R ( Lines/Row )

1行中の表示ライン数を定義します。

文字/グラフィック混在モードにおいて画面を分割して文字表示とグラフィック表示を混在させるような場合には，文字表示時の "L/R" 値を設定します ( SCROLL コマンドにおいて IM=1 を設定すると CSRFORM コマンドの設定にかかわらず自動的に1行が1ライン構成 ( L/R=0 ) となります)。

グラフィック・モードにおいても，L/Rを設定する必要がありますが，通常の場合には L/R=0を設定してください。 また垂直方向のみ拡大表示しようとする場合には ( 拡大係数 -1 ) の値を L/R に設定してください。ただし，この場合も IM=1 とすると L/R=0 を設定した場合と同じ動作をします。

L/R	ライン数
0 0 0 0 0	1
0 0 0 0 1	2
0 0 0 1 0	3
0 0 0 1 1	4
.....	.....
1 1 1 0 1	30
1 1 1 1 0	31
1 1 1 1 1	32

② C E

スレーブ時の外部同期信号 ( EX. SYNC ) の受け付け条件を定義します。EX. SYNC の立ち下がり で外部同期動作を起動します。

C E	機 能
0	外部同期信号受け付け開始の指示がなされているとき
1	無条件

マスタ設定をした G D C に対しては，C E = 0 を設定します。

③ C S ( Display Cursor )

文字表示時のカーサ表示の有無を定義します。

C S	機 能
0	カーサ表示無し。
1	カーサ表示有り。

( G D C が制御可能なメモリ空間は、文字モードで 8 K ワード、文字／グラフィック混在モードで 6 4 K ワードとなっています。 )

④ B D ( Blinking Disable )

文字表示時のカーサ表示ブリンクの有無を定義します。文字／グラフィック混在モードにおける文字属性用ブリンク・タイミング信号出力 (  $\overline{\text{AT.BLINK}}$  ) には影響を与えません。

B D	機 能
0	ブリンク有り。
1	ブリンク無し。CS=1 のとき常時点灯。

⑤ B L ( Blinking Rate )

文字表示時のカーサ点滅周期および、文字／グラフィック混在モードにおける文字属性用タイミング信号の点滅周期を定義します。

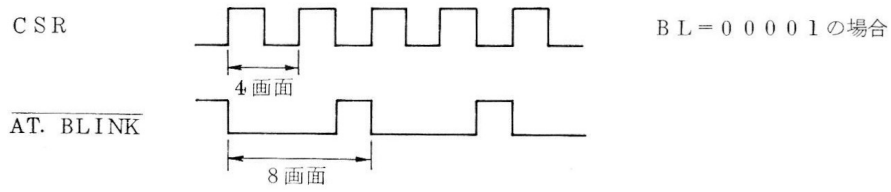
BL <sub>H</sub>	BL <sub>L</sub>	画 面			
0	0	0	0	0	設定できません
0	0	0	0	1	4
0	0	0	1	0	8
0	0	0	1	1	12
0	0	1	0	0	16
0	0	1	0	1	20
0	0	1	1	0	24
.....					.....
1	1	1	0	1	116
1	1	1	1	0	120
1	1	1	1	1	124

ブリンクの有無にかかわらず B L は設定してください ( B L = 0 は未定義のため使用不可 )。

なお、グラフィック・モード、文字／グラフィック混在モードでは CSRFORM コマンドに付随したパラメータは P 3 まで設定してください。

文字／グラフィック混在モードで G D C を使用したとき、ブリンク・アトリビュートのブリンク・タイミング信号が  $\overline{\text{AT.BLINK}}-\overline{\text{CLC}}$  出力から表示領域中に出力されます。点滅周期は B L 設定に依存し、B L 設定値の 2 倍周期で、表示デューティは 3 対 1

です。



注意 この場合の1画面とは1V(図3-5参照)期間をさします。

⑥ CST (Cursor Start)

文字表示時のカーサ表示開始ラインの値を定義します。

CST	ライン 値
00000	1
00001	2
.....	.....
11110	31
11111	32

ライン値とはLC+1でラインの本数を示します。

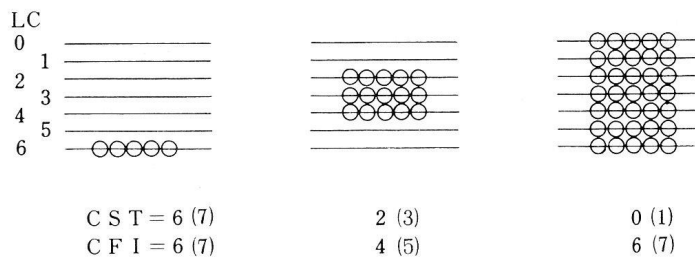
⑦ CFI (Cursor Finish)

文字表示時のカーサ表示終了ラインの値を定義します。

CFI	ライン 値
00000	1
00001	2
.....	.....
11110	31
11111	32

ライン値とはLC+1でラインの本数を示します。

CSTとの組合せでカーサ形状を決定します。アンダライン・カーサ、ブロック・カーサのほか、特殊なカーサ形状を選択できます。



### 3.2.7 PITCH コマンド

映像メモリ横方向のアドレス数を定義します。CRT面上の表示可能域の横幅はC/Rによって定義し、映像メモリの横幅はこのコマンドによって定義します。これにより自由度の高いスクロール操作や、表示領域外への描画が可能になります。

C 

0	1	0	0	0	1	1	1
---	---	---	---	---	---	---	---

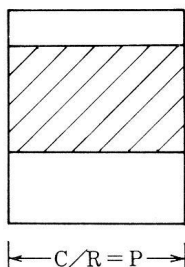
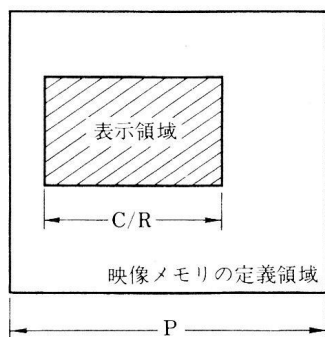
P1 

← PL →							
--------	--	--	--	--	--	--	--

PH	PL	文字数/行
0	0	0
0	1	1
0	2	2
⋮	⋮	⋮
0	254	254
0	255	255
1	256	256
1	257	257
⋮	⋮	⋮
1	510	510
1	511	511

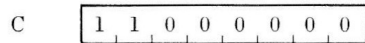
P<sub>H</sub>はSYNCコマンド、パラメータ5のP<sub>H</sub>ビットに設定します。

SCROLLコマンドに付随したパラメータ中のIMを"1"として使用する場合、PITCHの値は実際のPITCHの文字数の2倍の値を設定します。また、このコマンドはSYNCコマンドの後に設定してください。

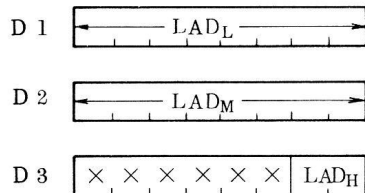


### 3.2.8 LPENコマンド

ライトペンによって得られた光検出信号がLPEN入力端子に与えられたときの表示アドレス(DAD)に対応するライトペン・アドレス(LAD)を読み出します。LIGHT PEN DETECTステータス・フラグが能動的になったことを確認してコマンドを発行します。



CPUは次の順番でLAD値を読み出すことが出来ますが、ドット・アドレス(dAD)を読み出すことはできません。

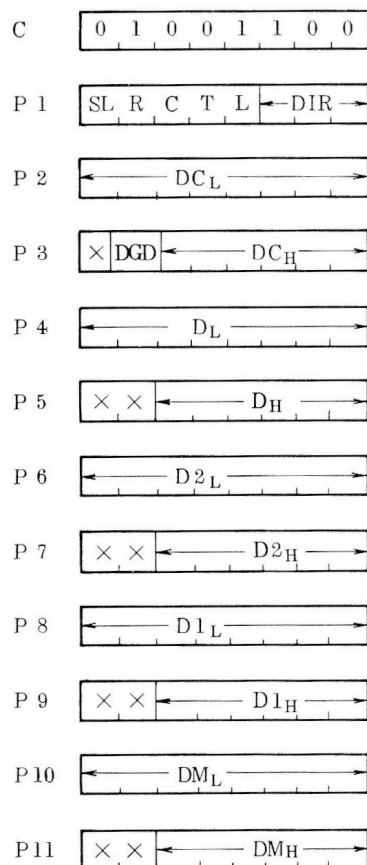


なお、RESET1, RESET2, RESET3 コマンドの実行後、またはこのコマンド実行によって、3バイトのアドレス・データ(LAD)がFIFOにセットされた後、LIGHT PEN DETECTステータス・フラグはリセットされます。

### 3.3 描画制御コマンド

#### 3.3.1 VECTWコマンド

描画方向、描画種類、14ビット構成の描画用パラメータ・レジスタDC, D, D2, D1, DMの値などを定義します。パラメータが負の値になる場合には、2の補数によって設定します。



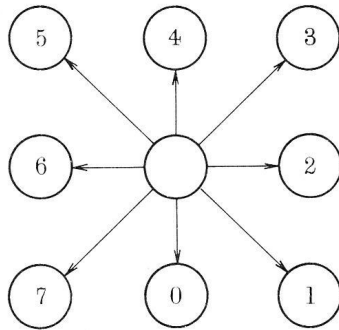
① S L , R , C , T , L

以下のように，描画種類を設定します。

S L	R	C	T	L	
0	0	0	0	0	○ 文字モード時の通常描画時 ○ 1ドット描画時 ○ READ/WRITEコマンド実行時 ○ D M A 転送時
0	0	0	0	1	○ 直線描画時
0	0	0	1	0	○ 傾斜しないグラフィック文字描画時
0	0	1	0	0	○ 円および弧描画時
0	1	0	0	0	○ 四辺形描画時
1	0	0	1	0	○ 傾斜したグラフィック文字描画時

② D I R ( Direction )

次のように描画方向を定義します。



READ, WRITEコマンドによる映像メモリの読み出し／書き込みは図の描画方向によって方向を定義します。

また，直線，四辺形，円弧，グラフィック文字などのグラフィック描画の場合には次頁の表を基にして描画方向を定義します。直線の場合，例えば傾きが0度から45度未満のとき，方向として"2"を設定します。円弧描画の場合は，1／8弧描画を描画単位としているため，どの領域に含まれる1／8弧であるかを示すため描画方向を定義します。他の場合には，1回目の描画基線方向を示します。

G D Cはこの描画方向定義を基にしてG D C内において描画アドレス計算を行います。

DIR	直線 <sup>(1)</sup>	円弧	グラフィック文字	傾斜グラフィック文字	四辺形	D M A <sup>(2)</sup>
0						
1						
2						
3						
4						
5						
6						
7						

- : 描画始点
- : 描画方向
- △ : 描画終了時のEAD指示点
- ➔ : 第1方向
- ⇨ : 第2方向
- : 定義域
- : 描画域

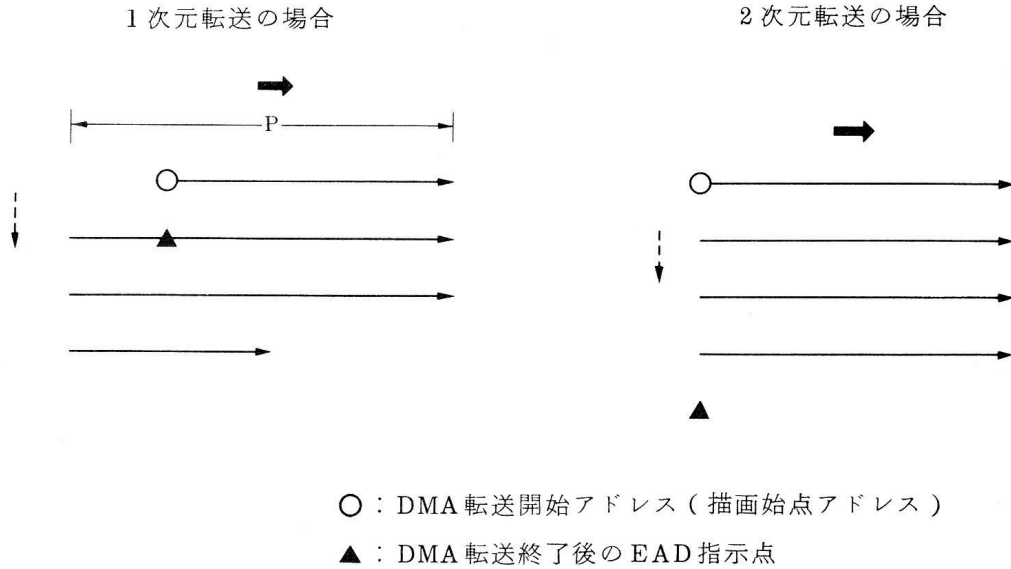
注意 1. 直線のDIR定義中破線矢印方向は書けません。

また、描画終了時のEAD指示点は描画した直線の延長にきます。

注意 2. DMA 転送終了後の EAD 指示点は 1 次元転送 / 2 次元転送および DMAW / DMAR により、次のように異なります。

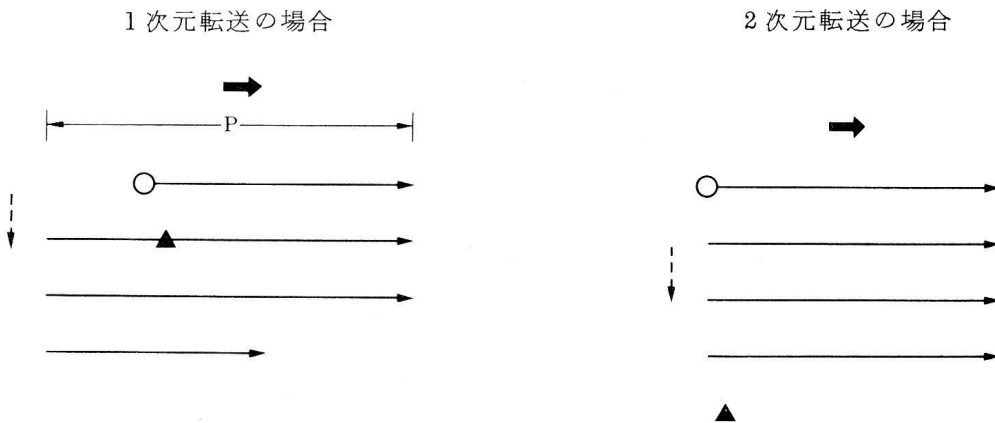
(i) DMAW の場合

1 次元の場合 (第 2 方向 = 1) は DMA 転送開始アドレス, 2 次元の場合は DMA 転送最終ラインの開始アドレスから, 各々第 2 方向に 1 だけ移動したアドレスになります。



(ii) DMAR の場合

1 次元の場合は DMA 転送開始アドレス, 2 次元の場合は DMA 転送最終ラインの開始アドレスから, 各々第 2 方向へ 1 だけ移動したアドレスに, さらに第 1 方向へ 1 だけ移動したアドレスとなります。





③ D G D ( Dynamic Graphics Drawing )

描画時のアドレスの進み方を定義します。

D G D	機 能
0	水平方向以外に描画アドレスが進行する場合には，±Pのオフセット・アドレスが加わります。
1	水平方向以外に描画アドレスが進行する場合には，±P/2のオフセット・アドレスが加わります。

P : PITCHコマンドにより設定するパラメータ

SCROLLコマンドにおいてIM=1とした領域では，必ずDGD=1としてください。

④ D C , D , D 2 , D 1 , D M

この各レジスタはRESET1，RESET2あるいはRESET3コマンドの実行時，または描画終了時に初期値に設定されます。また，負数は2の補数によって設定します。

	D C	D	D 2	D 1	D M
初 期 値	0	8	8	-1	-1
直 線	$ \Delta X $	$2 \Delta Y  -  \Delta X $	$2 \Delta Y  - 2 \Delta X $	$2 \Delta Y $	-
円	$(r/\sqrt{2})\uparrow$	$r-1$	$2(r-1)$	-1	0
弧	N	$r-1$	$2(r-1)$	-1	M
四 辺 形	3	A*	B*	-1	A*
8×8ドット グラフィックス文字	7	-	-	-	-
8×8ドット以外の グラフィックス文字， 塗りつぶし	B**	A**	-	-	-
DMA	B***	A***	$(A^{***}/2)^{(1)}$	-	-
READ/WRITE	D*	-	-	-	-
1文字 1ドット	-	-	-	-	-

- (2)  
 $\Delta X$  : X座標変位  
 $\Delta Y$  : Y座標変位  
r : 半径  
N : 描画総ドット数  
M : マスキング・ドット数  
 $\uparrow$  : 切り上げ  
A\* : 第1描画方向変位数
- B\* : 第2描画方向変位数  
A\*\* : 第1描画方向ドット数  
B\*\* : 第2描画方向ドット数-1  
A\*\*\* : 第1DMA方向バイト数<sup>(3)</sup>  
B\*\*\* : 第2DMA方向バイト数<sup>(3)</sup>  
D\* : R/Wバイト数/ワード数<sup>(3)</sup>  
- : 設定禁止パラメータ

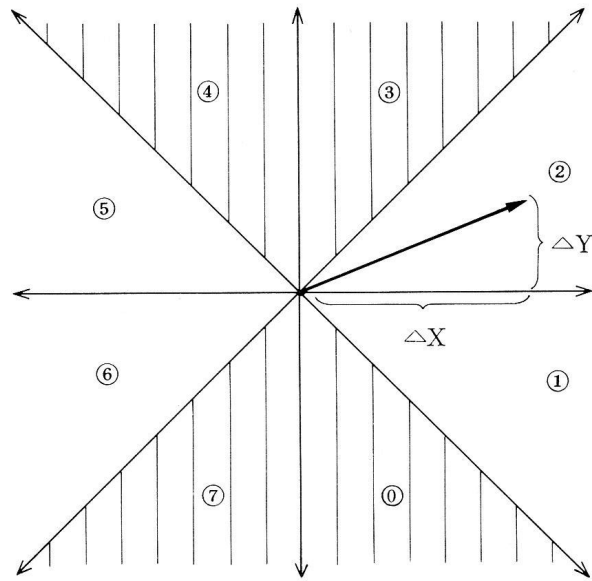
注意 1. DMAR時のワード転送の場合にのみ設定(他の場合は設定禁止)。

注意 2. Y軸方向に±45°の領域(DIRとして0,3,4,7の方向)に対して直線を描画する場合には $\Delta X$ と $\Delta Y$ の値を交換します。

注意 3. バイト/ワード指定およびDMAR/DMAWの指定により若干異なります(詳細については次頁以降を参照してください)。

## 描画パラメータの設定法

### (1) 直線描画

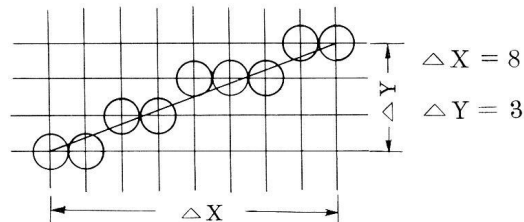


直線描画の場合は、直線の傾きによってパラメータの求め方が異なります。

上図においての " 1 " , " 2 " , " 5 " , " 6 " の範囲にある場合は、描画開始点と描画終了点間の X 座標変位 (  $\Delta X$  ) , Y 座標変位 (  $\Delta Y$  ) を算出します。この  $\Delta X$  ,  $\Delta Y$  を基に、整数減算と左シフトによってパラメータ (  $DC$  ,  $D$  ,  $D2$  ,  $D1$  ) を求めます。

直線の傾きが " 0 " , " 3 " , " 4 " , " 7 " の範囲にある場合は、算出した  $\Delta X$  と  $\Delta Y$  を交換 (  $\Delta X \leftrightarrow \Delta Y$  ) し、その結果よりパラメータの値を求めます。

#### 設定例



注意 1. 直線は 1 辺が 1 ドットで構成される四辺形の一つとも考えられますが、GDC では直線を四辺形描画によって代用することはできません。VECTW コマンドで  $D = 0$  または  $D2 = 0$  と設定しますと、16K ドット分の直線描画を行います。

注意 2. 直線描画時の VECTW のパラメータの省略

直線描画の際に発行する VECTW のパラメータは通常  $DC$  ,  $D$  ,  $D2$  ,  $D1$  までの 9 バイト分のパラメータを必要としますが、水平、垂直、 $45^\circ$  方向の直線描画の場合には  $DC$  と  $D$  までの 5 バイト分のパラメータを設定するだけで、描画実行することが可能です。また、その場合も  $D$  パラメータは  $DIR$  によって決まっていますので、 $DC$  パラメータのみ演算で求めればよく、CPU の負担が大幅に削減されます。

① 算出したDIR値が偶数の場合

DC : 総描画ドット数

D : 3 F F F H

D2, D1パラメータの設定は不要です.

② 算出したDIR値が奇数の場合

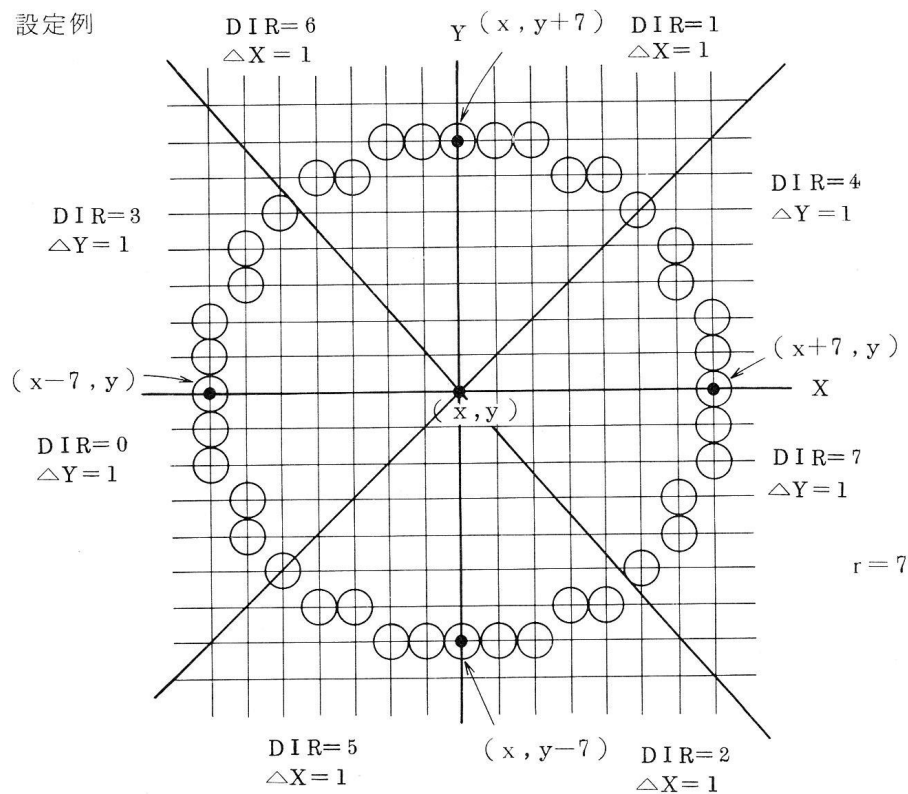
DC : 総描画ドット数

D : 0

D2, D1パラメータの設定は不要です.

## (2) 円描画

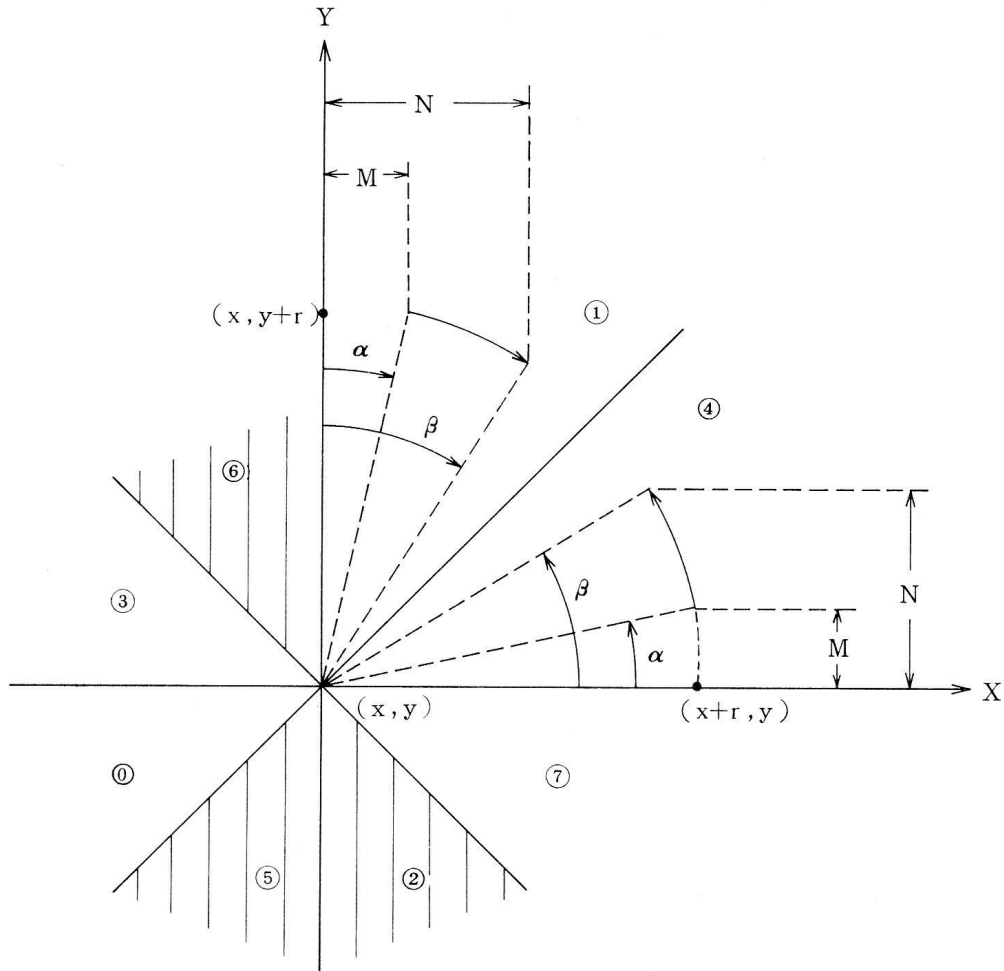
1/8弧を8回描画することによって全円を描画します. 描画開始点は, 中心から半径  $r$  だけ離れた, X, Y軸上の点に取ります. 中心座標  $(x, y)$ , 半径  $r$  のとき  $(x+r, y)$ ,  $(x, y+r)$ ,  $(x-r, y)$ ,  $(x, y-r)$  の四点から, それぞれ2回の1/8弧描画を行います. このとき描画方向(DIR)は描画毎に“+5”, または“-3”することにより求めると描画処理が簡単になります.



円, または弧描画の場合, X軸を描画開始点とする1/8弧の描画(DIR=0, 3, 4, 7)のときは, Y方向に±1ドットずつ変化しており, またY軸を描画開始点とする1/8弧の描画(DIR=1, 2, 5, 6)の場合は, X方向に±1ドットを取るように動作します.

### (3) 弧描画

描画開始点からのマスクング・ドット数 (M) を DM に、マスクング・ドット数を含む総描画ドット数 (N) を DC に設定します。マスクング区間には、描画アドレスは更新されますが、ドット修正動作は起こしません。



図において、中心 (x, y)、描画開始点 (x+r, y) または (x, y+r)、描画開始軸から見た開始角  $\alpha$ 、終了角  $\beta$  の弧の M および N は

$$M = r * \sin(\alpha) \quad , \quad N = r * \sin(\beta)$$

となります。

これは考え方を換えれば描画したい弧が前述したように、その DIR により X 軸方向または Y 軸方向へ ±1 ドットずつ動くということから、その弧を X 軸または Y 軸へ投影したときのドット数と考えてもよいからです。

弧が、" 0 "、" 3 "、" 4 "、" 7 " の範囲にある場合の描画開始軸は " X 軸 "、" 1 "、" 2 "、" 5 "、" 6 " の場合は " Y 軸 " となります。

(4) 四辺形描画

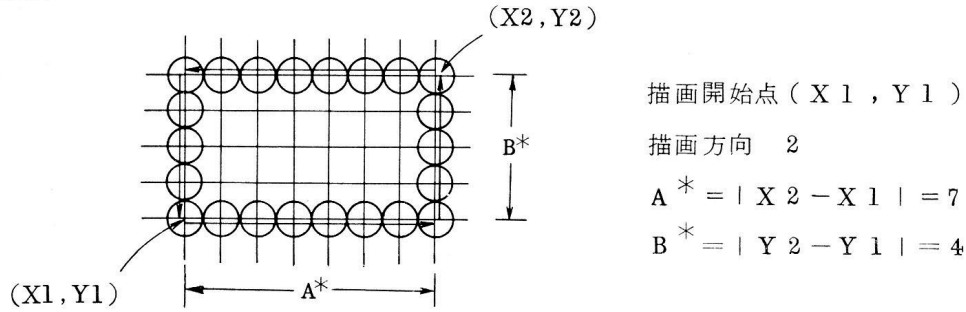
(i) 四辺形描画

対角線上の2点間の座標によって四辺形を定義したとき，2点間のX座標変位 ( $\Delta X$ )，Y座標変位 ( $\Delta Y$ ) を求めます。

描画方向 (DIR) = 2, 6 の時  $A^* = |\Delta X|$   $B^* = |\Delta Y|$

描画方向 (DIR) = 0, 4 の時  $A^* = |\Delta Y|$   $B^* = |\Delta X|$

設定例



(ii) 描画角度が45度に傾いた四辺形描画

描画開始軸を含む対角線上の2点間の座標によって四辺形を定義したとき，2点間のX，Y座標変位 ( $\Delta X$ ， $\Delta Y$ ) を求めます。

長辺を第1描画方向とした時  $A^* = (|\Delta X| + |\Delta Y|) / 2$

$B^* = ||\Delta X| - |\Delta Y|| / 2$

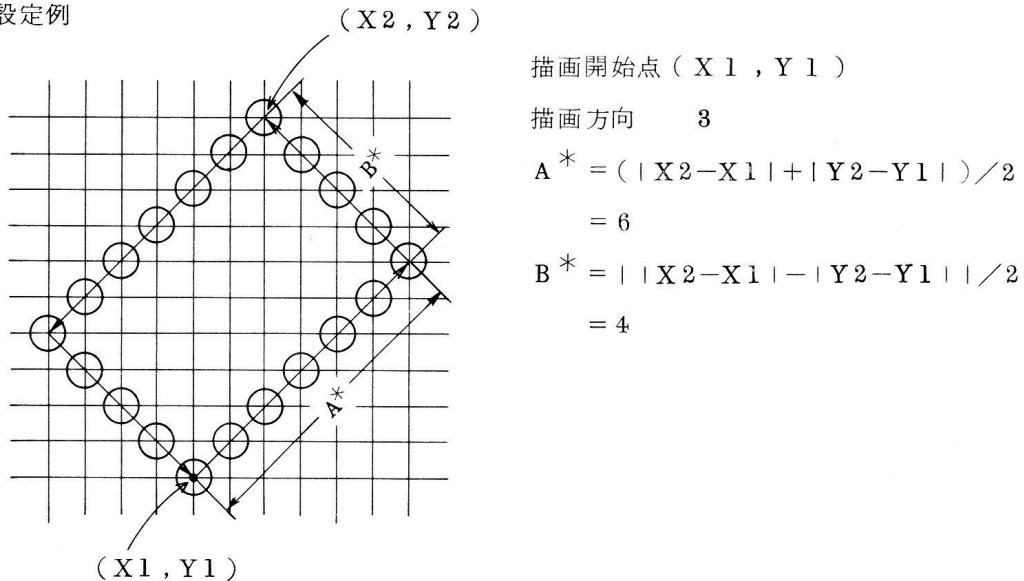
短辺を第1描画方向とした時  $A^* = ||\Delta X| - |\Delta Y|| / 2$

$B^* = (|\Delta X| + |\Delta Y|) / 2$

正方形の時  $A^* = B^* = (|\Delta X| + |\Delta Y|) / 2$

これも円るときと同様X軸およびY軸への投影と考えれば，上式を求めることができます。たとえば下の例ではY軸方向への投影を考えると， $\Delta Y = B^* + A^*$ ，X軸方向への投影を考えると $\Delta X = A^* - B^*$ となります。

設定例



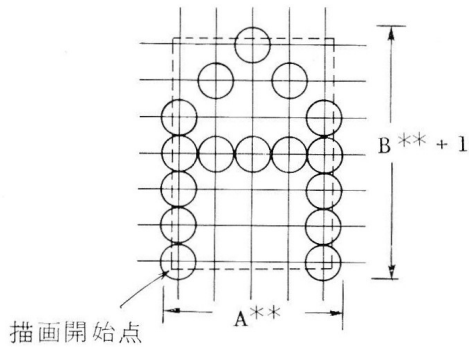
### (5) グラフィック文字描画

8×8ドット構成のグラフィック文字描画の場合には、DC=7を設定するだけで、Dの設定は特に行う必要はありません。

8×8ドット以外の場合には、DCおよびDに各々の描画方向に対するドット数を設定する必要があります。

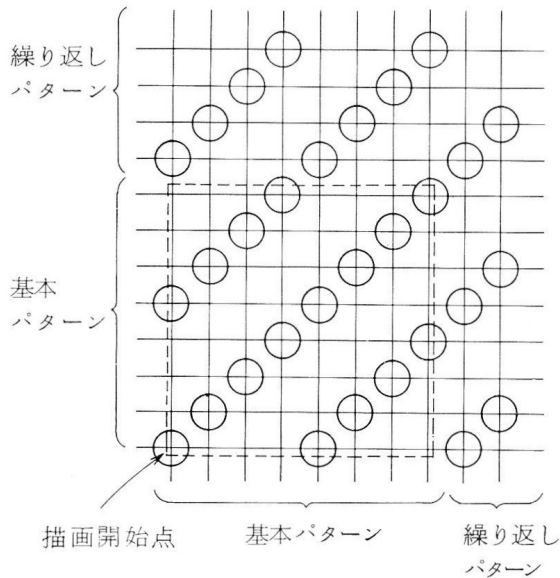
設定例

(i) 方向=2 横5ドット，縦7ドット ( $A^{**}=5$ ， $B^{**}=6$ )



RA		
9	0 0 1 0 0 × × ×	TX 7
A	0 1 0 1 0 × × ×	TX 6
B	1 0 0 0 1 × × ×	TX 5
C	1 1 1 1 1 × × ×	TX 4
D	1 0 0 0 1 × × ×	TX 3
E	1 0 0 0 1 × × ×	TX 2
F	1 0 0 0 1 × × ×	TX 1
	LSB	MSB

(ii) 方向=2 横10ドット，縦12ドット ( $A^{**}=10$ ， $B^{**}=11$ )



RA		
8	0 0 0 1 0 0 0 1	TX 8
9	0 0 1 0 0 0 1 0	TX 7
A	0 1 0 0 0 1 0 0	TX 6
B	1 0 0 0 1 0 0 0	TX 5
C	0 0 0 1 0 0 0 1	TX 4
D	0 0 1 0 0 0 1 0	TX 3
E	0 1 0 0 0 1 0 0	TX 2
F	1 0 0 0 1 0 0 0	TX 1
	LSB	MSB

### (6) DMA 転送

ワード単位での転送を行う場合には、転送バイト数は偶数でなければなりません。奇数となる場合には、最終転送をバイト単位転送とする必要があります。これを怠ると、GDCはハング・アップします（ハング・アップの解除はRESET1，RESET2またはRESET3コマンドによって行うことができます）。

VECTW コマンドによって設定した転送方向を第1DMA方向とし、もう一方の方向を第2DMA方向とすると各パラメータは次のような値に設定します。

どのようなDMA転送モードの場合でも、常に

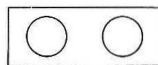
$$DC = \text{第2の方向のブロック数} - 1$$

となります。ブロック数とはワード転送のときにはワード数、バイト転送のときにはバイト数を示します。

<p><b>DMAW:WORD</b></p> <p>D = 第1の方向の転送バイト数 - 1 (但し, Dは奇数であること)</p> <p>D2 = 設定不要</p>
<p><b>DMAR:WORD</b></p> <p>D = 第1の方向の転送バイト数 - 2 (但し, Dは偶数であること)</p> <p>D2 = D / 2</p>
<p><b>DMAW および DMAR の上位/下位バイト指定</b></p> <p>D = 第1の方向の転送バイト数 - 1</p> <p>D2 = 設定不要</p>

転送順および描画パラメータの設定は次のようになります。

図において



は映像メモリの1アドレスを示し

左側が下位バイト, 右側の○が上位バイト, ○内の数字は転送順を示します。

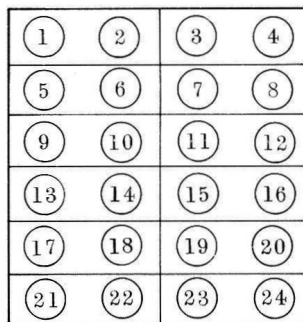
矩形領域に対する選択的DMA転送では無い1次元のDMAの場合にはDC = 0とします。

設定例

ワード転送のときの1ブロックは2バイトで第1描画方向, 第2描画方向ともブロックを単位としてバイト換算します。

(i) ワードDMA転送

第1方向 →

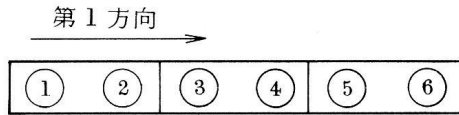


$$DC = 5, DIR = 2 \begin{cases} D = 3 & (\text{DMAW}) \\ D = 2 \quad D2 = 1 & (\text{DMAR}) \end{cases}$$

第1方向 ↑

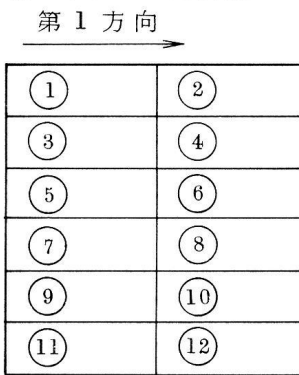
3	4	7	8	11	12	15	16	19	20	23	24
1	2	5	6	9	10	13	14	17	18	21	22

$$DC = 5, DIR = 4 \quad \begin{cases} D = 3 & (DMAW) \\ D = 2 \quad D2 = 1 & (DMAR) \end{cases}$$

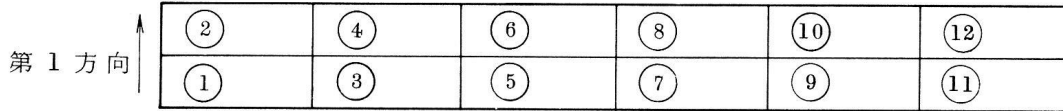


$$DC = 0, DIR = 2 \quad \begin{cases} D = 5 & (DMAW) \\ D = 4 \quad D2 = 2 & (DMAR) \end{cases}$$

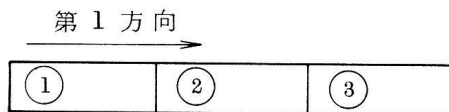
(ii) 下位バイトDMA転送



$$DC = 5, D = 1, DIR = 2$$

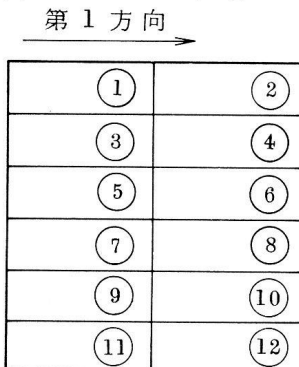


$$DC = 5, D = 1, DIR = 4$$



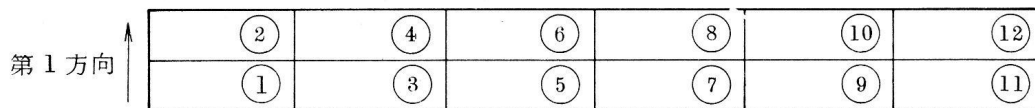
$$DC = 0, D = 2, DIR = 2$$

(iii) 上位バイトDMA転送



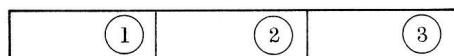
$$DC = 5, D = 1, DIR = 2$$





DC = 5, D = 1, DIR = 4

第1方向 →



DC = 0, D = 2, DIR = 2

DMA転送を行う場合に使用するDMAコントローラは、1バイト転送毎にDMAアクノリッジ信号 ( $\overline{DACK}$ ) を GDC に送り返す必要があります。GDC は、 $\overline{DACK}$  信号の立ち下がりタイミングを検出して、1バイトのDMA転送処理を開始します。

DMA転送は、バイトまたはワード単位で実行されます。他の描画のように、ドット単位での描画(転送)は実行できません。

### (7) READ, WRITE コマンド

DC に値を設定するとWRITEコマンド後のパラメータで設定したコードを、連続的に映像メモリに書き込むことが出来ます。また、READコマンド時には、映像メモリ内容を連続的にFIFOへ蓄積し、読み出すことが出来るようになりますが、READ, WRITEともDMAのように、2次元の領域をとることはできません。

次のような値をDCに設定します。

WRITE:WORD DC = 書き込みワード数 - 1
WRITE:LB, WRITE:HB DC = 書き込みバイト数 - 1
READ:WORD DC = 読み出しワード数
READ:LB, READ:HB DC = 読み出しバイト数

この場合もDMA転送時と同様、バイトまたはワード単位で実行されます。

WRITE:WORDコマンド後、奇数バイトのパラメータ送出をした後、他のコマンド送出をすると、最終転送パラメータに対する書き込み動作は起しません。偶数バイト目のパラメータによって映像メモリへの書き込み動作を起動するからです。このとき、DMA転送時のようにGDCがハング・アップすることはありません。

### (8) 1ドット描画

1ドット描画の場合、VECTWコマンドは描画の種類 (SL=R=C=T=L=0) とDIRのみ設定してください。あとのパラメータは不要です。また、描画するデータ "0" または、"1" は前もってPTNLのLSBに設定します。描画開始はVECTEコマンドで行います。

### 3.3.2 VECTE コマンド

直線，四辺形，円弧，1ドット描画の実行開始を指示します。

データRAMに格納されている線種データを参照しながら描画します。

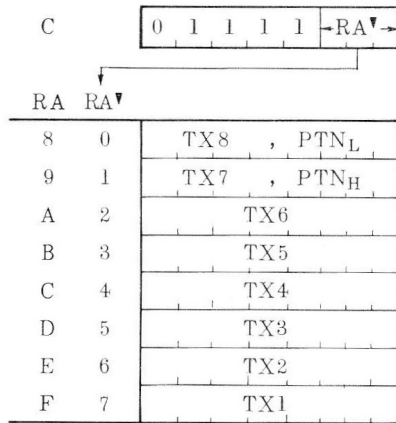
C     

0	1	1	0	1	1	0	0
---	---	---	---	---	---	---	---

### 3.3.3 TEXTW コマンド

実線，破線などの線種データ (PTN) または，グラフィック文字描画時に参照されるドット構成データ (TEXT1～8) を設定します。なお，パラメータの入力ごとにRA'は自動的にインクリメントされます。また，RA'は任意の値(0～7)に設定できますから，内蔵RAMの任意のアドレスから書き換えができます。連続したアドレスに格納するパラメータを与える場合には，コマンドの送出は最初の一回だけでできます。

これらのデータを参照しない文字制御の場合には，このコマンドは使用されません。



パラメータはGDC内蔵データRAMの上位8バイトに格納されます。

\* 4.3.1 線種，グラフィック文字ドット情報の設定を参照。

### 3.3.4 TEXTE コマンド

グラフィックス文字描画の実行開始を指示します。

データRAMに格納されているドット構成データを参照しながら描画します。

C     

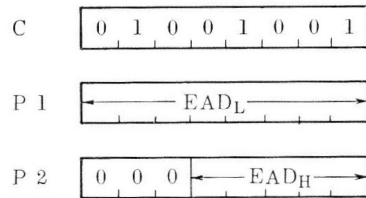
0	1	1	0	1	0	0	0
---	---	---	---	---	---	---	---

### 3.3.5 CSRWコマンド

描画実行ワード・アドレス (EAD) および描画実行ドット・アドレス (dAD) を設定します。

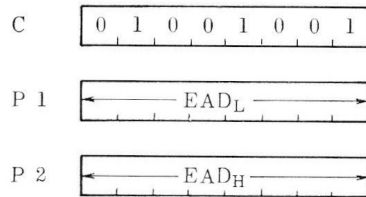
このコマンドでは文字制御時にはカーサ位置を，グラフィック制御時には描画開始点を定義するのに使用します。

(1) 文字モード

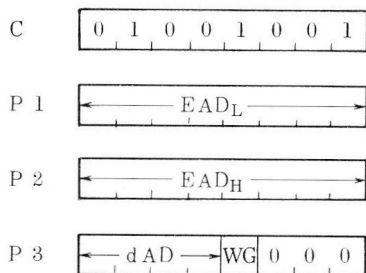


文字制御時，C S R Wコマンドの第1パラメータ解釈時から第2パラメータの解釈時までの間，G D Cは自動的にカーサ表示を消去します。

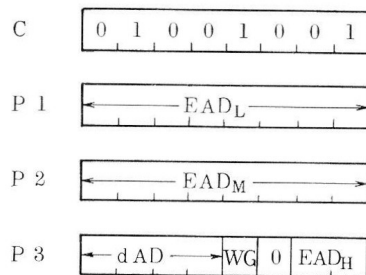
(2) 文字／グラフィック混在モードで文字表示のみを行うとき



(3) 文字／グラフィック混在モードでグラフィック表示／描画動作を行うとき



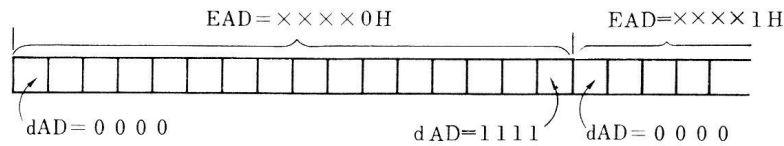
(4) グラフィック・モード



① dAD

最大16ビットのデータを格納するワード・アドレス内のドット位置を示す用途で用いられます。dADレジスタはMASKレジスタと共用しているため，その値はMASKコマンドによっても任意に設定することが出来ます。

ドット・アドレスは表示時，下位アドレスから上位アドレスの方向にスキャンされるように番地づけされています。



② WG

グラフィック・モード時または文字／グラフィック混在モード時においてDGD=1のとき，WRITE コマンドによる映像メモリへのデータ書き込み動作を定義します。

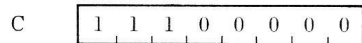
WG	機能
0	データの最下位ビットのみが書き込みデータとして参照
1	データをそのままの形で書き込む

WG=1に設定したとき，dAD=0に設定するとパラメータ解釈時間が節減できます。

※ 3.4.1 WRITE コマンド参照

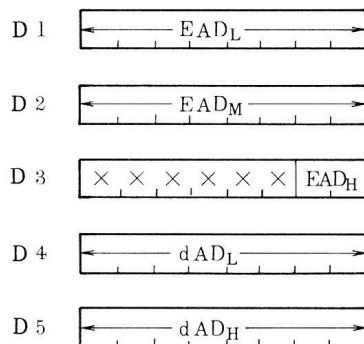
### 3.3.6 CSRR コマンド

描画実行ワード・アドレス (EAD) およびドット・アドレス (dAD) を読み出すための実行コマンドです。



このコマンドが実行されるとFIFOの内容がクリアされると同時に，FIFOの入出力方向がGDCからデータを読み出せる方向になり，CPUは次の順番でアドレスEADおよびdADを読み出すことができます。

また，アドレスの読み出しを中止したいときはFIFOの状態によらず，何らかのコマンドを書き込めばFIFOの内容がクリアされ，FIFOの方向がCPUからGDCにコマンド／パラメータを与えられる方向となって書き込んだコマンドが受け付けられます。

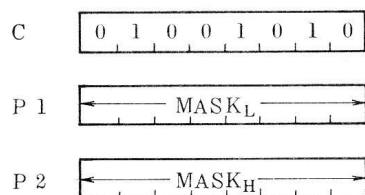


読み出されるドット・アドレス (dAD) は16ビットに展開された値であり，グラフィック描画終了後は，通常，いずれかのビットのみが"1"でほかのビットは"0"となっています。

描画を終了したとき，EAD，dADは最終描画を行ったアドレスの次に描画を行うべきアドレスの内容を保持しています。従って，この読み出しコマンドを用いることにより1つの描画（例えば弧描画）に続けて第2の描画（例えば，弧描画終了点より弧の中心点への直線描画）を行うことができます。

### 3.3.7 MASKコマンド

16ビットのdAD/MASKレジスタ(ドット・アドレス/マスク用レジスタ)の値を設定します。



文字制御におけるビット・マスク用としての用途と、ドット・アドレスを16ビットに展開した形で、直接、設定したり、多色の同時描画(4.1.1 1語を8/4/2/1ビットにする方法参照)で使用します。

MASKコマンドにより"1"としたビットのみ描画の影響を受けます。

前述したCSRWコマンドにおけるドット・アドレス(dAD)設定とMASKコマンドにおけるマスク・レジスタの設定との違いは次のようになっています。CSRWコマンドではドット・アドレス(dAD)は4ビットのエンコードされた形で与えられます。GDCは与えられた4ビットのデータを内部で自動的に、いずれか1ビットが"1"で他は"0"である16ビットのデータに展開した上、dAD/MASKレジスタに格納します。MASKコマンドでは16ビットのデータをそのままの形でdAD/MASKレジスタに設定します。

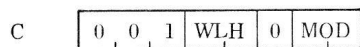
従って、CSRWコマンドdADの設定では16ビット中1ビットしか"1"となりませんが、MASKコマンドでは16ビット中任意のビットを"1"にすることができます。

なお、グラフィック・モードではMASKコマンドは特殊な場合(メモリのクリア[5.3 映像メモリ・クリア・フローチャート参照]、READコマンド使用時[3.4.2 READコマンド参照]等)、および多色の同時描画(1ワードを16ビット以外で使用する方法)を除いて使用しません。

## 3.4 映像メモリ制御コマンド

### 3.4.1 WRITE コマンド

ドット修正モードの設定や、このコマンドに続くパラメータを映像メモリに書き込むための準備をします。



ドット修正モードのみを変更したい場合には、パラメータを付随させずにこのコマンドのみを送出し、MODレジスタの内容を書き換えます。また、連続したアドレスに異なったデータを書き込むような場合には、1度コマンドを書き込んだ後パラメータのみを連続して送出することで、簡単に実現することが可能です。そのときに描画アドレスは自動的にインクリメントされます。このとき、WRITEコマンドの発行に先立って、書き込みたいブロック数-1の値と書き込み開始アドレス(EAD)の値をVECTWコマンド、およびCSRWコマンドによって設定してください(描画の

種類は  $S=L=R=C=T=L=0$  です )。

WRITEコマンドによる描画実行時、映像メモリ描画サイクルで動作しますが、MASKレジスタの内容により任意のビットをマスクすることが可能です。また、このWRITEコマンドの下位2ビットによるMODの設定により、MODレジスタの内容を変更するとともに設定されたドット修正モードで描画サイクルを実行します。

### ① WLH

ワード、下位バイト、上位バイトの選択をします。

WLH	機 能
0 0	ワード転送
0 1	設定できません
1 0	下位バイト転送
1 1	上位バイト転送

### ② MOD

ドット修正モードの選択をします。

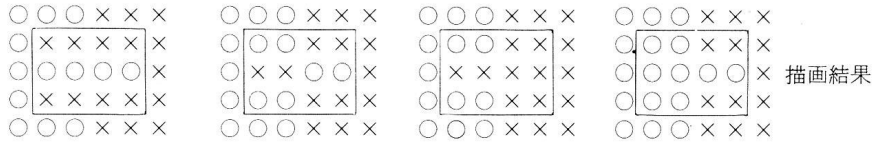
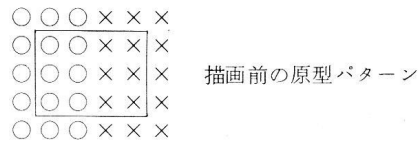
MOD	機 能
0 0	R E P L A C E
0 1	C O M P L E M E N T
1 0	C L E A R
1 1	S E T

各々の修正モードでは次図のような動作を行います。

REPLACE では、描画前の映像メモリ内容には無関係に、新しく設定した描画パターンが描画されます。

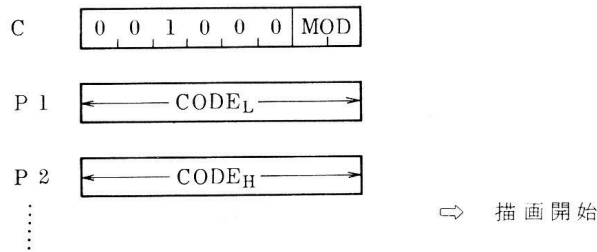
ほかの3種のモードでは、描画パターンが"1"であるドットに対してのみ、1/0反転、(COMPLEMENT)、0設定(CLEAR)、1設定(SET)の修正を加え、"0"であるドット位置に対応する映像メモリ内容は、単に、GDC内を循環して映像メモリに書き込みされます。

1ブロックを形成するパラメータの入力(ワード設定時:2バイト、下位または上位バイト設定時:1バイト)毎に、映像メモリに対するパラメータ・コードの書き込み動作を起します。映像メモリのアドレスは描画のたびにVECTWコマンドのDIRの値に従って更新されますので、パラメータを連続的に送出すると描画を連続的に実行します。

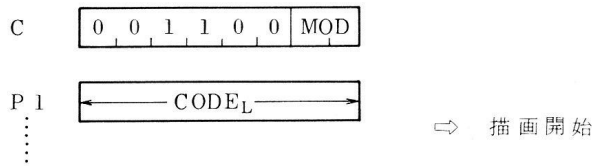


MOD=            0 0                    0 1                    1 0                    1 1  
                   (REPLACE)        (COMPLEMENT)        (CLEAR)                (SET)

(i) ワード単位書き込み

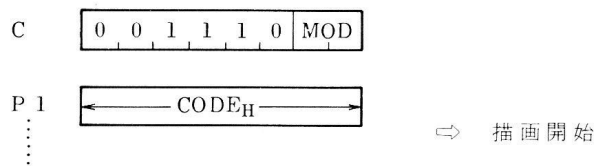


(ii) 下位バイト書き込み



上位バイトは自動的に "00" となります。

(iii) 上位バイト書き込み



下位バイトは自動的に "00" となります。

前述のようにWRITE コマンドは2つの動きを持っています。第1にドット修正モードを定義すること、第2に書き込みデータを定義することです。以下、各動作モードでのWRITEコマンドの使用法について述べます。

(1) 文字モード

キャラクタ・コードあるいはアトリビュート・コードを映像メモリに書き込む場合、または

MODレジスタの内容を変えてドット修正モードを変更する場合に使用します。

文字モードのときには、WRITEコマンドに付随したパラメータにより設定されたパターン・データ全体が一度に書き込みデータとして参照されます。例えばMASKレジスタには"0FFFFH"、ドット修正モードはREPLACEとした場合、パラメータとして5555Hが与えられていればEADで指定されたアドレスにそのデータが書き込まれます。

## (2) 文字／グラフィック混在モード

VECTWコマンドのパラメータ中で定義されるDGDビットの値により動作が異なります。

(i) DGD = 0

文字モードと同様な動作

(ii) DGD = 1

グラフィック・モードと同様な動作

## (3) グラフィック・モード

主としてMODレジスタの内容を変えてドット修正モードを変更する場合に使用します。

CSRWコマンドのパラメータ中で定義されるWGビットの値により動作が異なります。

(i) WG = 0

グラフィック・モードのときは、WRITEコマンドに付随したパラメータにより設定されたパターン・データの最下位が書き込みデータとして参照されます。このため、文字モードのようにWRITEコマンドによって映像メモリ上に任意のパターンを書くことはできません。

\* 5.1.6 グラフィック・モードでの1ワード描画参照。

(ii) WG = 1

文字モード時と同様に、WRITEコマンドによりパラメータ設定値をそのままの形で映像メモリに書き込みます。

## 3.4.2 READコマンド

映像メモリ内容を読み出すための実行コマンドです。このコマンドに先立って読み出したい映像メモリのブロック数と読み出し開始アドレス(EAD)をVECTWコマンド、およびCSRWコマンドで各々設定してください(描画の種類はSL=R=C=T=L=0です)。

C     

1	0	1	WLH	0	MOD
---	---	---	-----	---	-----

① WLH

WRITEコマンドと同じ。

② MOD

WRITEコマンドと同じ。

このコマンド実行時、映像メモリは描画サイクルで動作します。

1ブロックを形成するデータ読み出し(ワード設定時：2バイト、下位または上位バイト設定時：1バイト)ごとに、映像メモリのアドレスはVECTWコマンドのDIRの値に従って更新されますので、データを連続的に読み出すことができ、その値はFIFOに書き込まれます。このREADコマンド実行による描画サイクルでは、映像メモリから読み出して

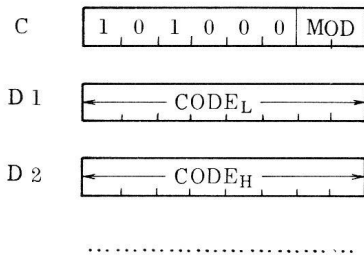


きたデータを単純に再書き込みするだけです。従って、MASK レジスタによるマスクの影響は全く受けません(ただし、グラフィック・モードでREADコマンドを使用する場合にはMASKレジスタを"0FFFFH"とする必要があります)。また、コマンドの下位2ビットによるMODの設定は単にMODレジスタの内容を書き換えるだけであり、READコマンドによる描画サイクルに対しては何らの影響も与えません。

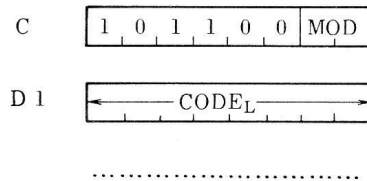
このコマンドが実行されますと、FIFOの内容がクリアされると同時にFIFOの方向がGDCからデータを読み出せる方向になり、CPUは次の順番で映像メモリ内容を読み出すことができます。

また、映像メモリの読み出しを中止したいときにはFIFOの状態によらず何かのコマンドを書き込めば、FIFOの内容はクリアされ、FIFOの方向がCPUからGDCにコマンド/パラメータを与える方向になって、書き込んだコマンドが受け付けられます。

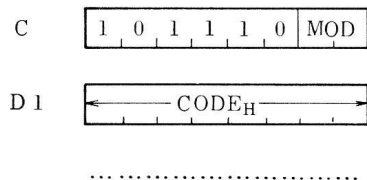
(i) ワード単位読み出し



(ii) 下位バイト読み出し



(iii) 上位バイト読み出し



### 3.4.3 DMAWコマンド

CPUが制御する主記憶からGDCを経由して映像メモリへデータをDMA転送するためのDMA実行指示コマンドです。GDCは、コマンド実行後、DRQ出力端子を能動的にします。

MASKレジスタおよびMODレジスタとの組み合わせによるこのコマンドの使用方法はWRITEコマンドと同様です。

なお、このコマンドの発行に先立ち、DMAによる転送領域とDMA転送開始アドレス(EAD)をVECTWコマンド、およびCSRWコマンドにより各々設定してください。

C 

0	0	1	WLH	1	MOD
---	---	---	-----	---	-----

① WLH

WRITE コマンドと同じ。

② MOD

WRITE コマンドと同じ。

(i) ワード単位DMA書き込み

C 

0	0	1	0	0	1	MOD
---	---	---	---	---	---	-----

下位バイト，上位バイトの順に主記憶から転送された後，映像メモリへの書き込み動作を起します。

(ii) 下位バイトDMA書き込み

C 

0	0	1	1	0	1	MOD
---	---	---	---	---	---	-----

下位バイトが転送された後，映像メモリへの書き込み動作を起します。

上位バイトは不定となりますので，これを禁止するにはMASKコマンドによって行ないます。

(iii) 上位バイトDMA書き込み

C 

0	0	1	1	1	1	MOD
---	---	---	---	---	---	-----

上位バイトが転送された後，映像メモリへの書き込み動作を起します。

下位バイトは不定となりますので，これを禁止するにはMASKコマンドによって行ないます。

このコマンドによるDMA転送の中断はRESET1，RESET2またはRESET3コマンドによって実行します。

### 3.4.4 DMARコマンド

映像メモリからGDCを経由してCPUが制御する主記憶へデータをDMA転送するためのDMA実行指示コマンドです。GDCは，コマンド実行後，DRQ出力端子を能動的にします。

MASKコマンドおよびMODレジスタとの組み合わせによるこのコマンドの使用方法はREADコマンドと同様です(ただし，グラフィック・モードであっても，READコマンドとは異なりMASKレジスタを"0FFFFH"とする必要はありません)。

C 

1	0	1	WLH	1	MOD
---	---	---	-----	---	-----

① WLH

READ コマンドと同じ。

② MOD

READ コマンドと同じ。

### 3.5 各種コマンド，パラメータの解釈に必要な時間

GDCのFIFOに各種コマンド，パラメータを書き込んでからそれらがFIFOから読み出された後，解釈されるまでに必要な時間は表3-1のようになっています。

表3-1 各種コマンド，パラメータの解釈に必要な時間（クロック数）

コマンド名	コマンド解釈	パラメータ解釈*1	備考
SYNC	6	2	
RESET1 RESET2 RESET3	6	2	
MASTER/SLAVE	12	—	
START	12	—	6BHのコマンド
	6	—	0DHのコマンド
STOP1 STOP2	6	—	
ZOOM	10	2	
SCROLL	10	4	
CRSFORM	10	2	
PITCH	10	2	
LPEN	12	—	
VECTW	10	2	
VECTE	18	—	
TEXTW	10	4	
TEXTE	16	—	
CSRW	10	P1, P2:各2 P3:4~64	*2
CSRR	14	—	
MASK	10	2	
WRITE(Word)	12	P1:2 P2:4	
WRITE(High Byte)	12	8	
WRITE(Low Byte)	12	8	
DMAW	WRITEと同じ		
READ(Word)	14	—	
READ(High Byte)	12	—	
READ(Low Byte)	14	—	
DMAR	READと同じ		

\*1. パラメータの解釈時間は1パラメータあたりのクロック数

\*2. P3以降はdADによります(dAD=0,1,2,……,F→4,8,12……64クロック)

注意 コマンド，パラメータの解釈中に1画面が終了した場合に12クロック，画面分割が起きた場合には10クロックが追加されます。

## 第4章 機能説明

### 4.1 CPUインタフェース

A0,  $\overline{RD}$ ,  $\overline{WR}$  信号の組み合わせにより, GDCはデータ・バスを介してCPUとの間で以下に示す情報の入出力を行います。A0は, 通常, システム・アドレス・バスの最下位ラインに接続します。また, A0は $\overline{DACK}$ がHレベルの場合のみ有効となります。

$\overline{DACK}$	A0	$\overline{RD}$	$\overline{WR}$	機能
1	0	0	1	GDCステータス・フラグの読み出し
1	1	0	1	データ(GDC内FIFO)の読み出し
1	0	1	0	パラメータの書き込み
1	1	1	0	コマンドの書き込み
0	×			GDCのDMA転送(DMAR, DMAWコマンド実行時)

×: Don't care

## 4.1.1 ステータス・フラグ

GDCには、次に示す8種のステータス・フラグが用意されています。

$A0=0$ ,  $\overline{RD}=0$ ,  $\overline{WR}=1$ のときGDCはDB0-7にステータス・フラグを出力しますので、CPUはその情報からGDCの状態を判断して種々の動作を起こすことができます。GDCとCPUの間にコマンド/パラメータの受け渡しがないとき、外付け回路によってステータス・フラグ読み出し状態にする信号を与えると、DB0-7からステータス・フラグを静的な信号として取り出すことができます。

表4-1 GDCステータス・フラグ

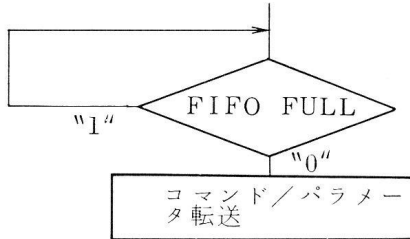
フラグ名称	読み出し端子	機能
DATA READY	DB0	GDCがREADなどの読み出しコマンド実行後、読み出しデータが、読み出し可能な状態になったことを示します。
FIFO FULL	DB1	FIFOがデータで満たされたことを示します。
FIFO EMPTY	DB2	FIFOが空であることを示します。
DRAWING	DB3	GDCが描画中であることを示します。 テキスト描画以外の描画時には描画開始から終了までこのフラグは"1"になっていますが、テキスト描画時には内蔵RAMからGDC内部のレジスタにその内容が転送されるたびにこのフラグはいったん"0"となります。また、DMAR/DMAW/READ/WRITEの各コマンド実行中はGDCが描画サイクルにあるときだけ"1"となります。
DMA EXECUTE	DB4	DMA転送を続行中であることを示します。 DMA転送の開始から終了まで"1"になります。
VERTICAL SYNC	DB5	垂直同期信号(VSYNC)が発生していることを示します。
HORIZONTAL BLANK/ VERTICAL BLANK	DB6	VHビット(SYNCコマンド, P6, ビット7)が"0"のとき、水平消去信号(HBLANK)が発生していることを示します。またVHビットが"1"のとき、垂直消去信号(VBLANK)が発生していることを示します。
LIGHT PEN DETECT	DB7	ライトペン信号によるアドレスの検出が成されたことを示します。 RESET1, RESET2あるいはRESET3コマンドの発行、またはLPENコマンドの発行後3バイトのデータ(LAD)がFIFOにセットされた後、リセットされます。

## 4.1.2 コマンド/パラメータ

コマンドの場合は  $A0=1, \overline{RD}=1, \overline{WR}=0$  として、またパラメータの場合は  $A0=0, \overline{RD}=1, \overline{WR}=0$  として、DB0-7にコマンド/パラメータを与えるとGDCに書き込まれます。

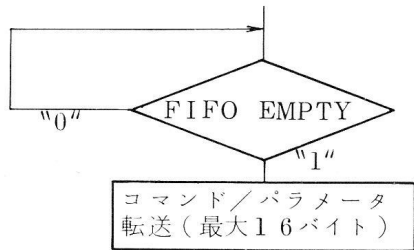
コマンド/パラメータの書き込み手順を次に示します。

- (1) FIFO FULL="0"であることを書き込み前に検出してコマンド/パラメータをGDCに送出します。



描画を行うときのように一連のコマンド/パラメータを送る必要がある場合に、このような方法を用いると描画処理が長くなってしまふことが多くなります。しかし、後述の初期化ルーチンのような場合には描画サイクルの動作を行うようなコマンドの送出はありませんので(メモリ・クリアのルーチンを除く)、GDCとCPUのクロックにもよりますが、FIFO FULLになる可能性は低くなります。

- (2) FIFO EMPTY="1"であることを書き込み前に検出して、最大16バイトまでのコマンド/パラメータをGDCに連続的に送出します。



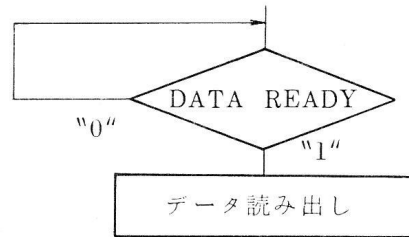
描画を行うときのように、一連のコマンド/パラメータを送る必要がある場合には、送るべきコマンド群の先頭で一度FIFO EMPTYを確認した後、16バイト分のコマンド/パラメータを無条件に送ります。17バイト目以降はFIFO FULLの確認を行いながら書き込むと描画処理に要する時間が短縮されます。

このとき、主記憶に格納しておくコマンド/パラメータを主記憶→GDC間DMA転送によって連続転送することも出来ます。ただし、この場合のDMA転送にはGDCのDRQと $\overline{DACK}$ 端子は使用できません( $\overline{DACK}=1$ )。また、A0の入力が必要となります。

### 4.1.3 データ

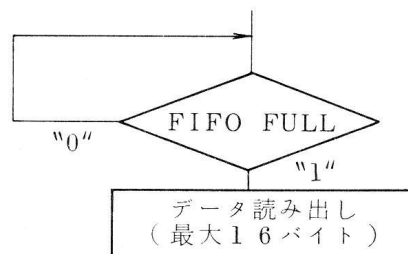
$A0=1, \overline{RD}=0, \overline{WR}=1$  のとき, GDCはDB0-7上にFIFO内のデータを読み出します。

- (1) DATA READY="1"であることを読み出し前に検出してデータをGDCから読み出します。



CSRR, LPEN コマンドによる読み出し時には所要数のデータ読み出しを連続して行うことが出来ます。READコマンドによる読み出し時には1バイト読み出し毎にDATA READYフラグ="1"を検出する必要があります。しかし、16バイトFIFO内にデータが連続的蓄積されるため、CPU側の読み出し速度がGDCの映像メモリ・データ読み出し速度よりも遅い場合には、このフラグのチェックは最初の1回だけですみます。

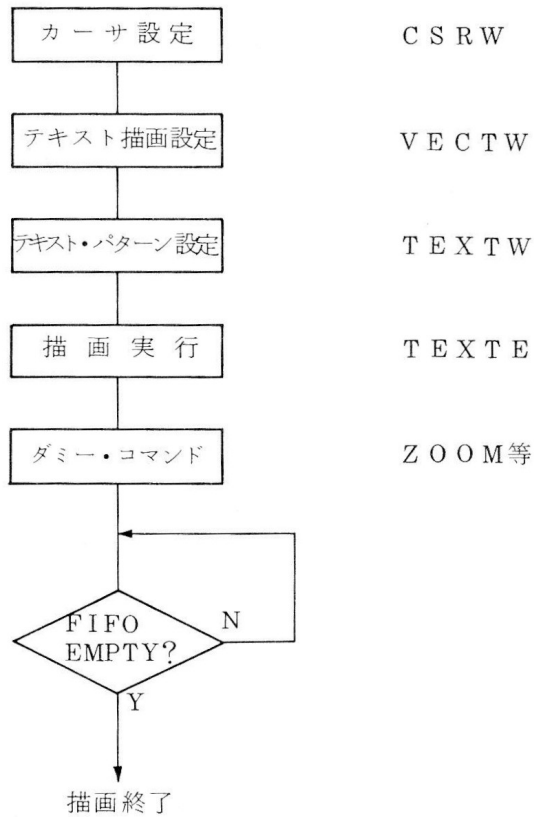
- (2) FIFO FULL="1"であることを読み出し前に検出し最大16バイトのデータを連続的に読み出します。



ただし、CSRR, LPENコマンドによる読み出し時にはFIFO FULLフラグが"1"となることはありません。この方法はREADコマンド時に使用することが出来ますが、読み出しバイト数が16のときにはFIFO FULL=1となることはありませんので、注意してください。GDC→主記憶間DMA転送によって連続読み出しを行っても構いませんが、GDCのDRQと $\overline{DACK}$ 端子は使用できません( $\overline{DACK}=1$ )。また、A0の入力が必要となります。

### 4.1.4 描画終了の検出

通常の描画終了の検出はGDCステータス・フラグのDRAWINGフラグ(DB3)をチェックすることにより行えます。しかし、TEXT描画の場合には、前述のように内蔵RAMからGDC内部のレジスタにその内容が転送されるたびに、DRAWINGフラグはいったん"0"となります。このような場合において、描画終了を検出するためにはDRAWINGフラグは用いずに以下のような方法をとります。すなわち、テキスト描画を実行させた後に、GDCが描画サイクルの動作を行わないようなダミーのコマンド(例えばZOOMコマンド等)をGDCに書き込みます。その後、FIFO EMPTYフラグをチェックすることにより、描画が終了したかどうかを検出することができます。



#### 4.1.5 FIFOの動作

FIFOはRESET1,RESET2またはRESET3コマンド実行後,CPU→GDCの方向,すなわちコマンド/パラメータを受け付ける方向になっています.しかし,READ,CSRR,LPENのいずれかのコマンドが実行されますと,FIFOの内容(実行待ちのコマンド/パラメータ)がクリアされ,同時にGDC→CPU,すなわちデータの読み出しの方向に変化します.従って,READ,CSRR,LPENの各コマンドのあとにコマンド/パラメータを続けて送出した場合には,その実行が無視されることがあります.また,データの読み出し方向にあるGDCに前記3種類以外のコマンドを与えますと,さきほどとは逆にFIFOの内容がクリアされ,同時にCPU→GDCの方向となってこのコマンドが受け付けられます.

## 4.2 表示機能

### 4.2.1 表示アドレスの進み方

表示アドレスは

- (1) ノン・インタレース/インタレース/インタレース・シュリンク
- (2) 1行中のライン数(L/R)の設定
- (3) 拡大表示係数(ZR)の設定

の各々の組み合わせにより,その進み方が変化します(表4-2~4-4).



表4-2 L/R=0, ZR=0のときの表示アドレスの進み方

ノン・インタレース ( I=0, S=0 )

1	DAD	DAD+1	DAD+2	DAD+3		
2	DAD+P	DAD+P+1	DAD+P+2	DAD+P+3		
3	DAD+2P	DAD+2P+1	DAD+2P+2	DAD+2P+3		
4	DAD+3P	DAD+3P+1	DAD+3P+2	DAD+3P+3		
⋮						

インタレース ( I=1, S=0 )

1	DAD	DAD+1	DAD+2	DAD+3		
2	DAD+P	DAD+P+1	DAD+P+2	DAD+P+3		
3	DAD+2P	DAD+2P+1	DAD+2P+2	DAD+2P+3		
4	DAD+3P	DAD+3P+1	DAD+3P+2	DAD+3P+3		
⋮						

第1フィールド

1	DAD	DAD+1	DAD+2	DAD+3		
2	DAD+P	DAD+P+1	DAD+P+2	DAD+P+3		
3	DAD+2P	DAD+2P+1	DAD+2P+2	DAD+2P+3		
4	DAD+3P	DAD+3P+1	DAD+3P+2	DAD+3P+3		
⋮						

第2フィールド

インタレース・シュリンク ( I=1, S=1 )

1	DAD+P	DAD+P+1	DAD+P+2	DAD+P+3		
2	DAD+3P	DAD+3P+1	DAD+3P+2	DAD+3P+3		
3	DAD+5P	DAD+5P+1	DAD+5P+2	DAD+5P+3		
4	DAD+7P	DAD+7P+1	DAD+7P+2	DAD+7P+3		
⋮						

第1フィールド

1	DAD	DAD+1	DAD+2	DAD+3		
2	DAD+2P	DAD+2P+1	DAD+2P+2	DAD+2P+3		
3	DAD+4P	DAD+4P+1	DAD+4P+2	DAD+4P+3		
4	DAD+6P	DAD+6P+1	DAD+6P+2	DAD+6P+3		
⋮						

第2フィールド

表 4-3 L/R=2, ZR=0 のときの表示アドレスの進み方

ノン・インタレース ( I=0, S=0 )

1	DAD	DAD+1	DAD+2	DAD+3		
2	DAD	DAD+1	DAD+2	DAD+3		
3	DAD	DAD+1	DAD+2	DAD+3		
4	DAD+P	DAD+P+1	DAD+P+2	DAD+P+3		
⋮						
⋮						

インタレース ( I=1, S=0 )

1	DAD	DAD+1	DAD+2	DAD+3		
2	DAD	DAD+1	DAD+2	DAD+3		
3	DAD	DAD+1	DAD+2	DAD+3		
4	DAD+P	DAD+P+1	DAD+P+2	DAD+P+3		
⋮						
⋮						

第1フィールド

1	DAD	DAD+1	DAD+2	DAD+3		
2	DAD	DAD+1	DAD+2	DAD+3		
3	DAD	DAD+1	DAD+2	DAD+3		
4	DAD+P	DAD+P+1	DAD+P+2	DAD+P+3		
⋮						
⋮						

第2フィールド

インタレース・シュリンク ( I=1, S=1 )

1	DAD	DAD+1	DAD+2	DAD+3		
2	DAD+P	DAD+P+1	DAD+P+2	DAD+P+3		
3	DAD+P	DAD+P+1	DAD+P+2	DAD+P+3		
4	DAD+2P	DAD+2P+1	DAD+2P+2	DAD+2P+3		
⋮						
⋮						

第1フィールド

1	DAD	DAD+1	DAD+2	DAD+3		
2	DAD	DAD+1	DAD+2	DAD+3		
3	DAD+P	DAD+P+1	DAD+P+2	DAD+P+3		
4	DAD+2P	DAD+2P+1	DAD+2P+2	DAD+2P+3		
⋮						
⋮						

第2フィールド

表 4-4 L/R=0, ZR=2 のときの表示アドレスの進み方

ノン・インタレース ( I = 0 , S = 0 )

1	DAD	DAD	DAD	DAD+1		
2	DAD	DAD	DAD	DAD+1		
3	DAD	DAD	DAD	DAD+1		
4	DAD+P	DAD+P	DAD+P	DAD+P+1		
⋮						
⋮						

インタレース ( I = 1 , S = 0 )

1	DAD	DAD	DAD	DAD+1		
2	DAD	DAD	DAD	DAD+1		
3	DAD	DAD	DAD	DAD+1		
4	DAD+P	DAD+P	DAD+P	DAD+P+1		
⋮						
⋮						

第1フィールド

1	DAD	DAD	DAD	DAD+1		
2	DAD	DAD	DAD	DAD+1		
3	DAD	DAD	DAD	DAD+1		
4	DAD+P	DAD+P	DAD+P	DAD+P+1		
⋮						
⋮						

第2フィールド

インタレース・シュリンク ( I = 1 , S = 1 )

1	DAD	DAD	DAD	DAD+1		
2	DAD+P	DAD+P	DAD+P	DAD+P+1		
3	DAD+P	DAD+P	DAD+P	DAD+P+1		
4	DAD+2P	DAD+2P	DAD+2P	DAD+2P+1		
⋮						
⋮						

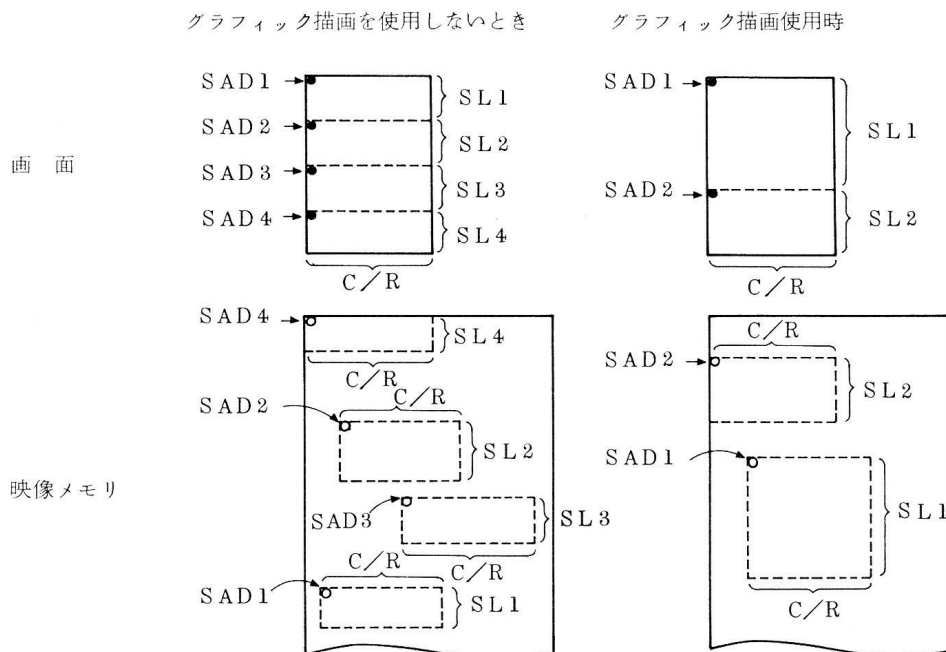
第1フィールド

1	DAD	DAD	DAD	DAD+1		
2	DAD	DAD	DAD	DAD+1		
3	DAD+P	DAD+P	DAD+P	DAD+P+1		
4	DAD+2P	DAD+2P	DAD+2P	DAD+2P+1		
⋮						
⋮						

第2フィールド

## 4.2.2 スクロール，画面分割の方法

SCROLLコマンドによって設定するSAD（表示開始アドレス），SL（表示領域を示すライン数）により，表示画面を水平方向に最大4個（文字制御時）まで分割して表示できます。画面の分割は表示アドレスに不連続点を設けることによって成されます。直線，円弧などのグラフィック描画を行う場合には，最大画面分割数は2個（グラフィック制御時）となります。



SADとして任意値を設定できますので，映像メモリの任意アドレスから表示を開始できます。

SAD 1～4，SL 1～4はSCROLLコマンドによって内蔵データRAMに一旦格納されます。表示開始時SAD 1とSL 1がRAMから読み出され表示開始アドレスとしてSAD 1が使用されます。SL 1は一水平走査終了ごとにデクリメントされ，“0”になると次の表示領域の定義をするSAD 2，SL 2がRAMから読み出され表示に使用されます。以降，上記シーケンスを繰り返し，1画面表示を終了します。

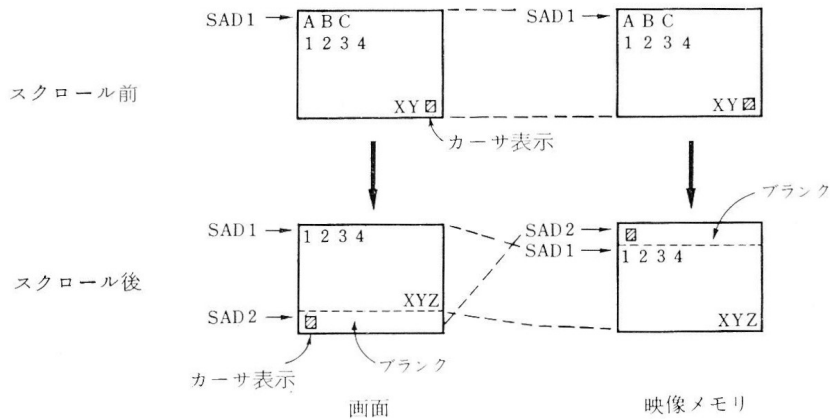
ただし，表示停止中にはこの動作は起こりませんので，そのときのアドレス出力は正常値が保証されません。従って，表示制御コマンド（特にSTARTコマンド）はVSYNC信号がHレベルの期間中に入れるようにしてください。

以上のような理由からSAD，SLの設定値変更という単純操作で多岐にわたるスクロール操作が可能になります。また画面の一部にスクロールに影響されない固定表示域を設定することもできます。

### (1) ページ内スクロール (INTRA PAGE)

画面1枚分に相当する1ページ分の映像メモリ内でスクロール動作をする形式を指します。

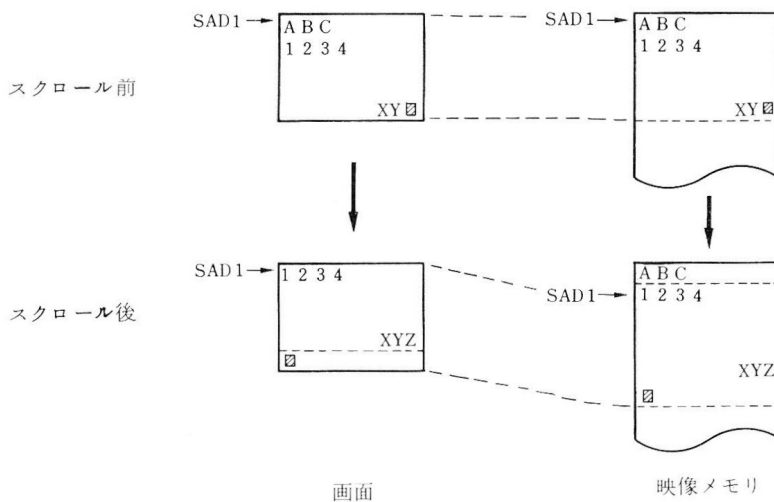
下図のようにカーサ位置が最下行または最下行最終文字位置にあるとき、LF (ライン・フィールド) 動作または1文字入力動作を行ったとき画面上では全行1行分だけ巻上げ (スクロール) を行い、最下行を抹消するスクロール操作が通常行われます。



スクロールを行う場合、スクロール条件の検出をCPU側で行った後、SAD2から始まる1行分の映像メモリを同一文字繰返し書き込み法によってクリアし、SAD1, SL1, SAD2, SL2の再設定を行うことによってスクロール操作が終了します。

### (2) ページ間スクロール (INTER PAGE)

画面1枚分を超える容量をもつ映像メモリが使用可能である場合にのみ実現できます。

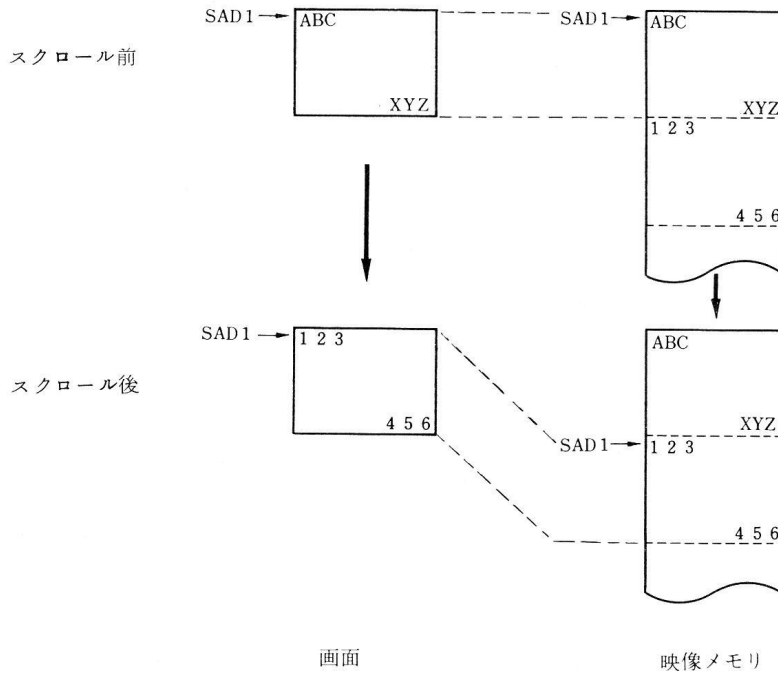


スクロール条件検出後SAD1の値を1行分移動することによって簡単にスクロール操作を実行できます。

このスクロール形式では画面上方にスクロールを行うばかりでなく、下方へのスクロールもSAD1の変更によって容易に実現できます。

### (3) ページ切換え

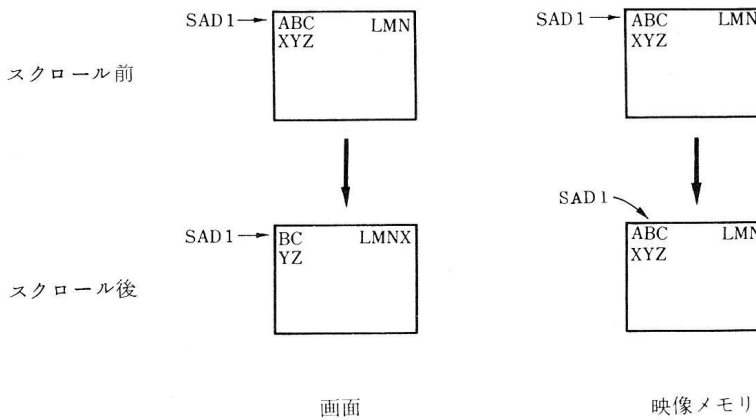
画面1枚分に相当する映像メモリを数枚分もち、それらをSADを変化させることによってページ(画面1枚)毎に切り換えて表示させることができます。



ページ間スクロールとの違いは、SAD1の変化量を1行分とするか1画面分とするかの相違だけで、CPUが管理します。

### (4) 水平方向スクロール

映像メモリの規模には無関係に、水平方向にスクロールする領域を画面上に定義します。SADの変化量を1文字単位に選択することによって実現できます。

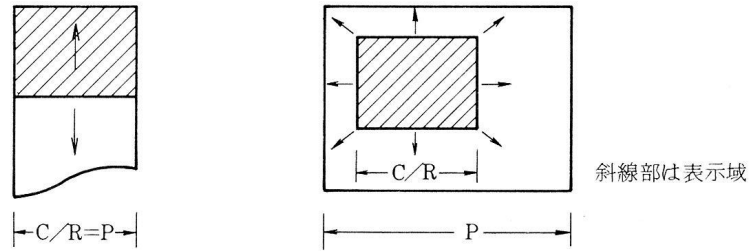


注意 本例ではピッチPとC/Rとの間に $P=C/R$ のような関係を有しています。

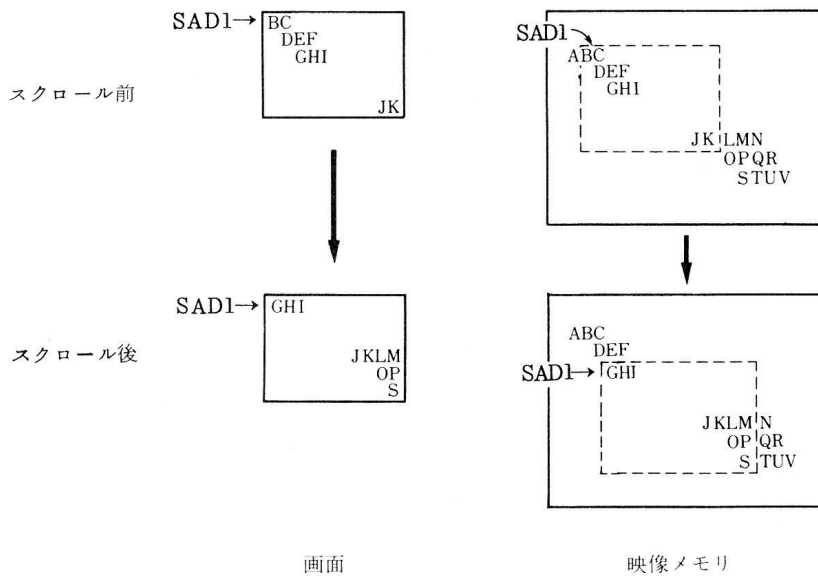
## (5) 全方向スクロール

映像メモリの規模を画面1枚分の領域に対して、水平垂直両方向に対して余裕をとって設定したときのみを実現できます。

SYNCコマンドのC/R設定による水平方向表示域定義よりも、PITCHコマンドによる映像メモリの水平方向アドレス総数定義を下図右のように大きな値に取ると全方向へのスクロール（パニング）が可能になります。



SADを変化させることにより容易に全方向スクロールを実現することができます。



文字表示時、垂直方向は1行単位、水平方向は1文字単位でスクロールできます。垂直方向の表示領域を示すSLはライン数で設定することに注意する必要があります。1行表示の途中のラインで“SL=0”となり、画面分割条件を満たすと、その行表示の途中で次行表示が途中から開始します。画面分割によってライン・カウンタの進み方は変更されません。

グラフィック表示時、垂直方向は1ライン単位（1ドット単位）、水平方向はワード単位（16ドット単位など）でスクロールできます。簡単な外付け回路を用いると水平方向についても1ドット単位スクロールが可能となります。（第6章ハードウェアを参照）。

文字表示において、簡単な外付け回路とSYNCコマンドのVBPおよびL/F値の操作によって、全画面の垂直方向1ライン単位でのスムーズ・スクロールが可能です。

(6) SCROLLコマンドに付随したSLパラメータに関して

SLパラメータは

画面分割を行わない場合  $SL1 \geq L/F$  (ノン・インタレース)

$SL1 \geq (L/F) + 1$  (インタレース)

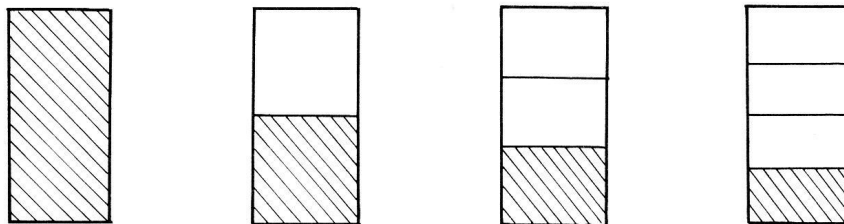
画面分割を行う場合  $\sum_{m=1}^n SLm \geq L/F$  (ノン・インタレース)

$\sum_{m=1}^n SLm \geq (L/F) + 1$  (インタレース)

を満足するように設定すればよいのですが、次のように設定しても動作上は全く支障がなく手間もかかりません。

画面分割を行わない場合  $SL1 = 3FFH$

画面分割を行う場合 最終分割領域の  $SL = 3FFH$



部分の  $SL = 3FFH$  と設定します。



## 4.3 描画機能

### 4.3.1 線種，グラフィック文字ドット情報の設定

TEXTWコマンドによって，16ビットの1/0パターンをデータRAMの8番地および9番地に記憶し，直線，円弧，四辺形描画実行コマンドVECTEを与えると，GDCは内蔵RAMにあるこの線種パターンを参照しつつ描画しますので，この1/0パターンを変えることにより任意の線種にできます。

設定例

LSB								MSB									
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	→	—————
1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	→	- - - - -
1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	→	- - - - -
1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	→	- - - - -
1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	→	—— ———
1	0	1	1	1	1	1	0	1	0	1	1	1	1	1	0	→	- - - - -
1	1	0	1	1	1	1	0	1	1	0	1	1	1	1	0	→	- - - - -
1	0	1	0	1	1	1	0	1	0	1	0	1	1	1	0	→	- - - - -

← P T N<sub>L</sub>
← P T N<sub>H</sub>

描画開始時には，必ずパターン情報のLSBから参照が開始されます。  
 パターン情報は内蔵データRAMのアドレス"8"と"9"に格納されます。

RA	MSB								LSB								
8	0	1	1	1	0	1	0	1	0	1	1	1	0	1	0	1	P T N <sub>L</sub>
9	0	1	1	1	0	1	0	1	0	1	1	1	0	1	0	1	P T N <sub>H</sub>

データRAM内容

TEXTW コマンドによって，グラフィック文字描画時に参照される8バイト構成(8×8ドット)のドット構成情報を内蔵データRAMのアドレス"8"から"F"に格納することが出来ます。

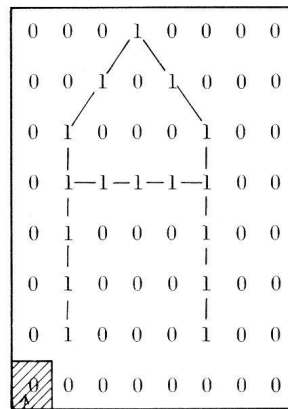
設定例

RA	LSB								MSB										
8	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	T X 8		
9	0	0	1	0	1	0	0	0	0	0	1	0	1	0	0	0	T X 7		
A	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	T X 6		
B	0	1	1	1	1	1	1	0	0	0	1	1	1	1	1	1	0	0	T X 5
C	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	T X 4		
D	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	T X 3		
E	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	T X 2		
F	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	T X 1		

8×8ドット領域に5×7ドットで表現される文字"A"のドット構成情報を例えば，上図のようにTX8からTX1まで設定します。描画時にはTX1からTX8の順に参照されます。また，線

種指定の場合と同様パターン情報のLSBから参照が開始されます。

上記設定時の描画例 (ZW=0, DIR=2)



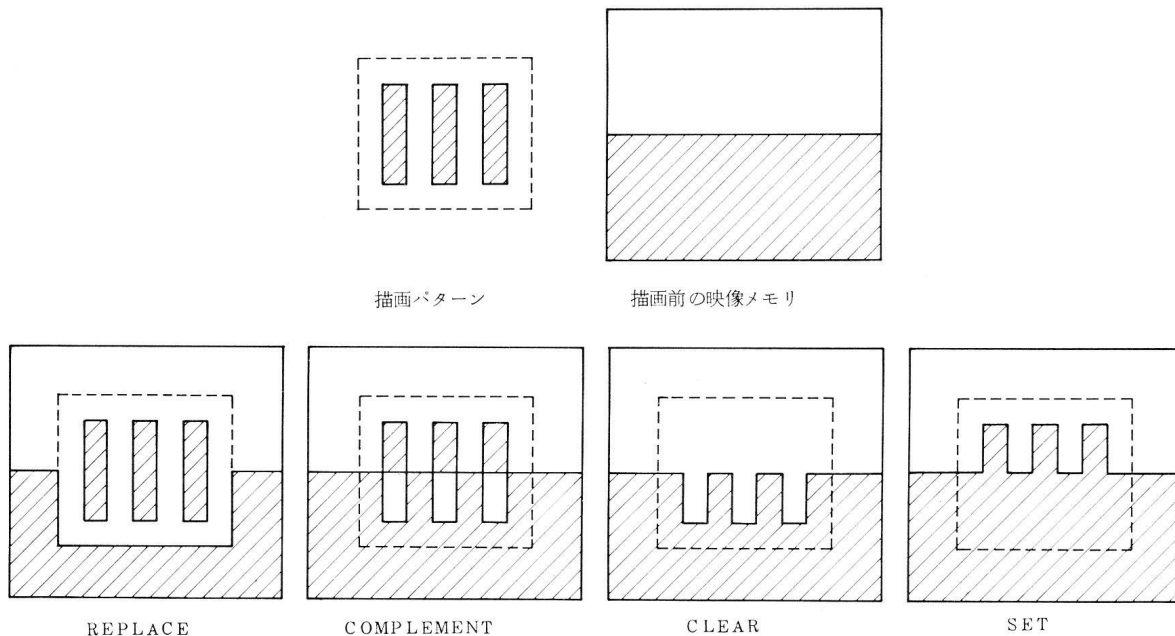
描画開始点

### 4.3.2 ドット修正機能

GDCは、REPLACE、COMPLEMENT、CLEAR、SETの4種のドット修正機能を持っています。WRITEコマンド下位2ビット(MOD)を設定することによって修正の種別を選択できます。

破線枠内のパターンをグラフィック文字描画によって映像メモリへ描画したときの例を図4-1に示します。斜線部は“1”のパターン、他は“0”のパターンを示します。REPLACE時には、描画前の映像メモリ内容とは無関係に、新しい描画パターンを描画します。他の3種の修正モードでは、描画パターンが“1”であるドットに対してのみ、COMPLEMENT指定時には描画前映像メモリ内容を反転し、CLEAR時には“0”に、SET時には“1”に各々修正を加えます。描画パターンが“0”であるドットに対応する描画前映像メモリ内容には修正を加えず、単に、GDC内を循環し、映像メモリへ同一のデータが再書き込みされます。

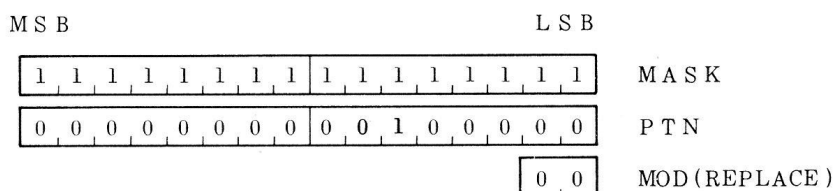
図4-1 ドット修正機能



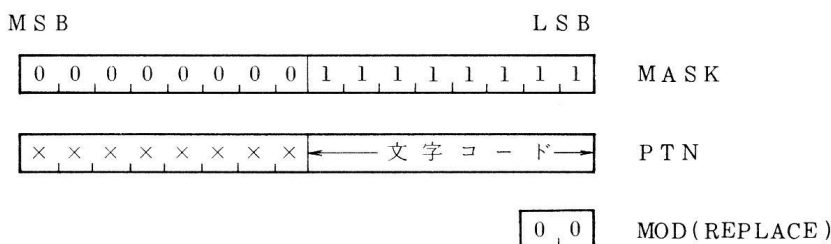
文字モードにおいてもドット修正機能を使用することが出来ます。MASKコマンドによるMASKレジスタ操作によって、文字コード領域とアトリビュート・ビット領域との判別や、ある1ビットのみのアトリビュート・ビットのみを選択し修正を加えることが出来ます。

映像メモリ1アドレスに格納される16ビットのデータのうち、上位8ビットを反転、ブリンク、高輝度などの文字属性（アトリビュート）制御用ビットとして使用し、下位8ビットを文字コード格納用として使用したとき、次のようにMASKレジスタをMASKコマンドにより、PTNレジスタをWRITEコマンドのパラメータによって設定すると各種制御を達成できます。

- (1) アトリビュート領域=0，文字コード領域=20Hにクリアするとき

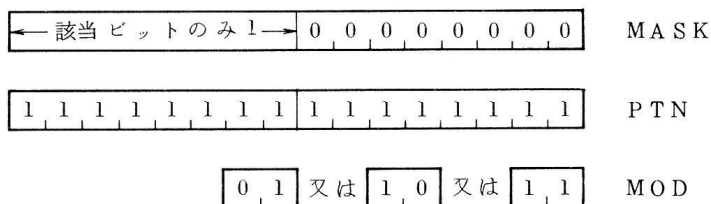


- (2) アトリビュート領域は不変で、文字コードのみを変更するとき

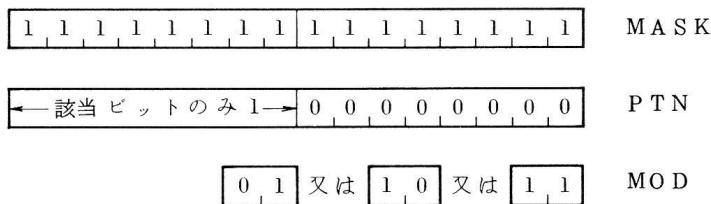


MASKレジスタで“1”のビットに対応するPTN内容が、修正モードに従って修正されます。“0”であるビットに対応する映像メモリ内容には、修正は加えられません。

- (3) 文字コード領域は不変で、アトリビュート領域の任意ビットのみを修正するとき



あるいは



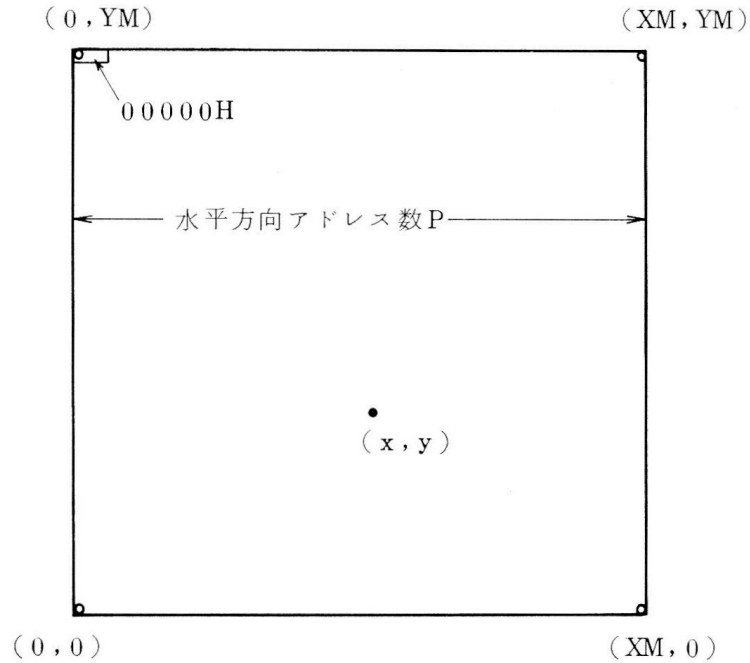
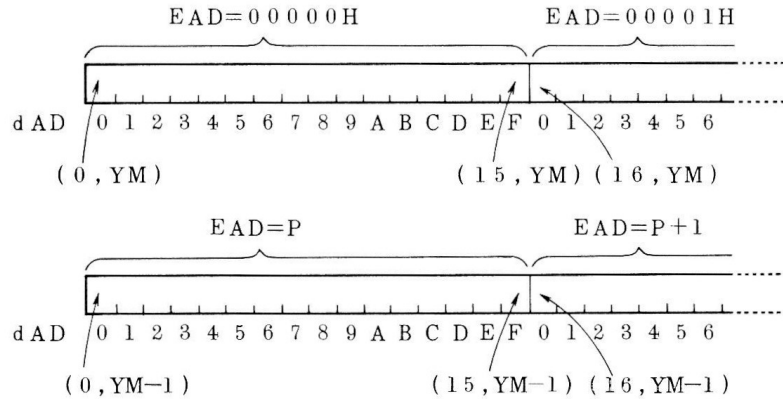
VECTWコマンドによって転送バイト／ワード数をDCにあらかじめ設定しておくこと、連続的に上記変更や修正を実行します。

### 4.3.3 座標と絶対番地との対応

CSRWコマンドによって描画開始アドレスを設定します。X，Y座標をベースとして点がアドレスされているグラフィック装置では座標→絶対番地変換を必要とします。その変換処理は簡単な加減算とシフト操作のみで実現できます。

次図に示すように1ワードを16ビット構成とし、XY座標原点(0,0)を映像メモリ左下隅に

とりX座標最大値をXM, Y座標最大値をYMとし, (0, YM)の点をワード・アドレス0番地, ドット・アドレス0番地とします. 左から右の方向にドット・アドレスが1番地ずつ増加し, ドット・アドレス16番地でワード・アドレスが1番地増加します. ワード・アドレス0および1番地付近の座標値との対応の詳細を下图に示します.



水平方向アドレス数つまりPITCH Pは

$$P = \{ (XM+1) / 16 \} \uparrow$$

ある点(x, y)のワード・アドレスEADとドット・アドレスdADは

$$EAD = P * (YM - y) + INT(x / 16)$$

$$dAD = REM(x / 16)$$

ここで↑: 切り上げ, INT: 整数部, REM: 剰余部を示します.

INT(x / 16)はxの右シフト4回, REM(x / 16)はxの下位4ビットのみの抽出によって容易に求められます. Pについても同様です. また, Pが"2のべき乗"である場合にはP \* (YM - y)は(YM - y)の左方向シフトによって求められます. また, YMを $2^n - 1$  (n: 正の整数)であるとすれば(YM - y)はyをnビットとしたときの1の補数で与えられます. これから, XM, YMは2のべき乗の値から1をひいた値とすれば, 計算が簡単になることがわかります.

## 4.3.4 描画により影響を受けるレジスタ

GDCにはプログラムできるレジスタが各種ありますが、それらは1度設定すると再度プログラムするまで前の値を保持しているものと、描画時等に自動的に変化していくものの2種類があります。

### (1) プログラムで書き換えない限り、前の値を保持しているレジスタ

- 以下のコマンドおよびパラメータにより設定するレジスタ

SYNC

MASTER/SLAVE

ZOOM

SCROLL

CSRFORM

PITCH

TEXTW

- ドット修正モードを決定するMODレジスタ

- MASKレジスタ(ただし、文字/グラフィック混在モードでDGD=1として描画を行った場合、およびグラフィック・モードにおいて描画を行った場合は変化します。)

- VECTWコマンドのパラメータで設定するSL,R,C,T,L(円,直線,四辺形,テキスト・パターンを使ったグラフィック文字等の描画種類の設定)および描画方向を決定するDIR,DGDレジスタ

- CSRWコマンドの第3パラメータで設定するWGレジスタ

### (2) 描画時に影響を受けるレジスタ

- 描画実行アドレス(EAD,dAD)

描画終了後は次の描画実行すべきアドレスとなります。

- VECTWコマンドに付随したパラメータで設定するDC,D,D2,D1,DMのレジスタ  
設定された領域の描画終了後もしくはRESET1,RESET2またはRESET3コマンド実行後はそれぞれ

$DC = 0, D = 8, D2 = 8, D1 = -1, DM = -1$

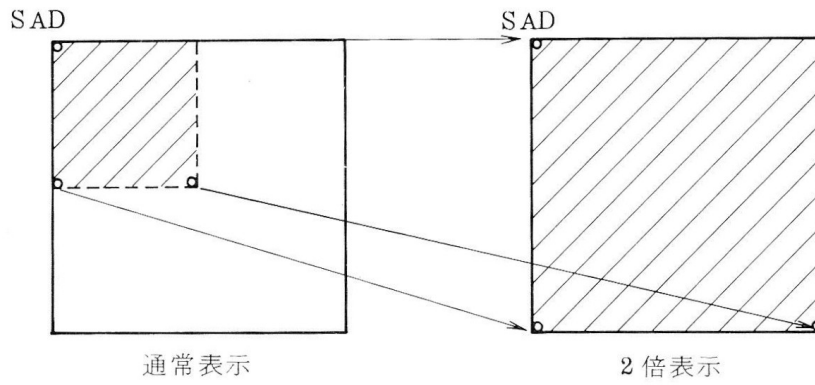
となります。

## 4.4 拡大機能

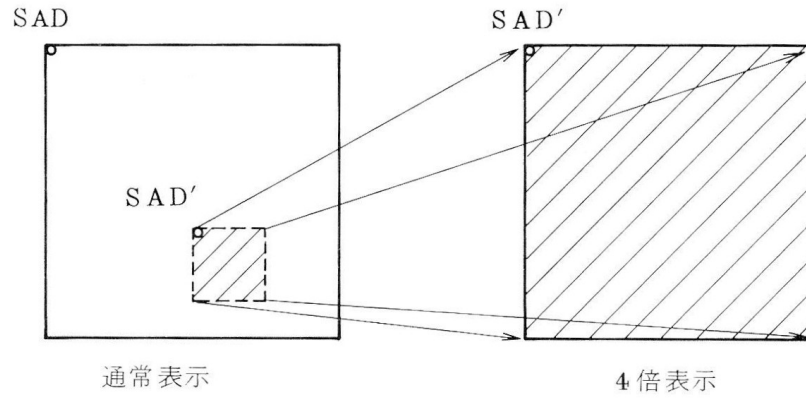
### 4.4.1 拡大表示機能

1から16までの整数倍での拡大表示が可能です。ZOOMコマンド第1パラメータ上位4ビットZRによって拡大係数を決定します。拡大係数に従って、縦横方向に等倍に拡大するようにアドレス制御されます。

SCROLLコマンドによって、表示開始アドレス(SAD)を変化させないとき、次のように拡大します。



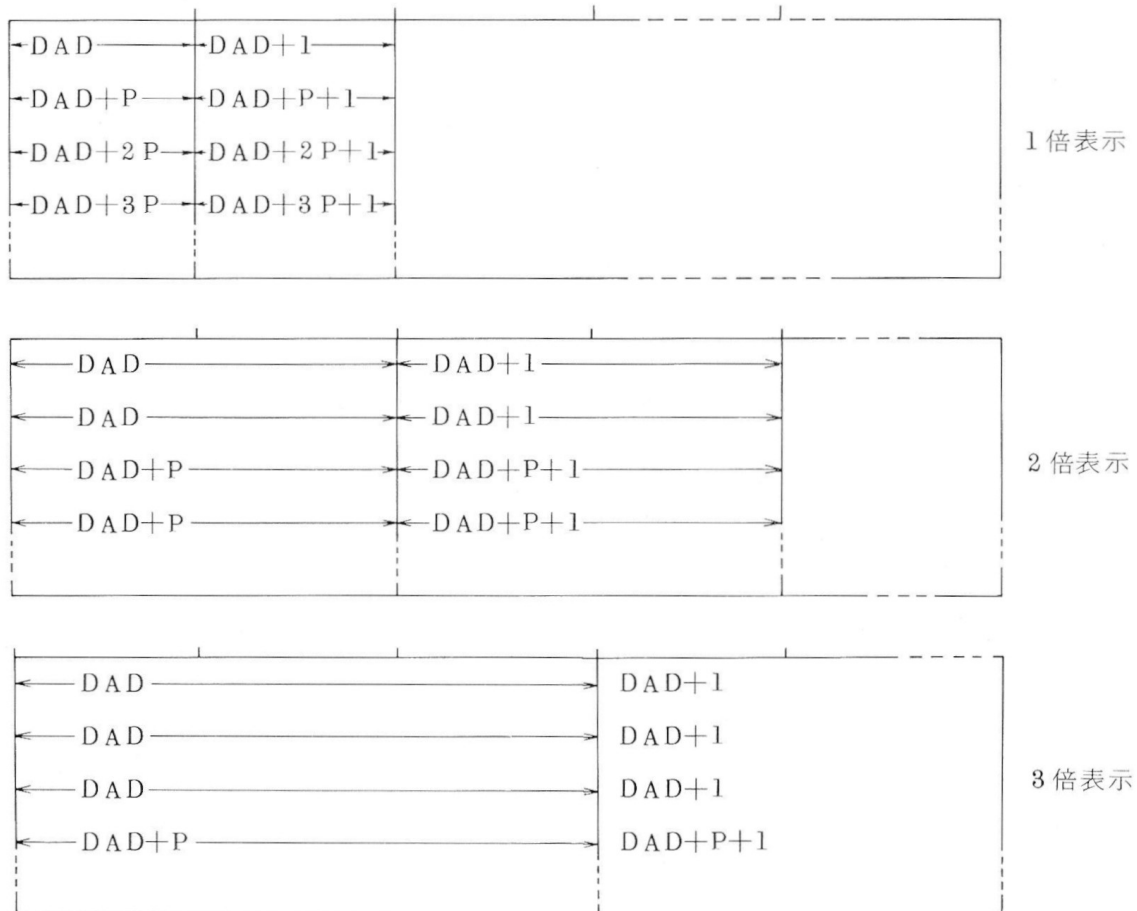
SCROLL コマンドによつて，SAD を変化させると表示画面の任意箇所を拡大表示できます。



拡大表示時，表示アドレスの周期を図 4-5 のように変更します（表示画面上）。

$L/R=0$ ，ノンインタレース時

図 4 - 5



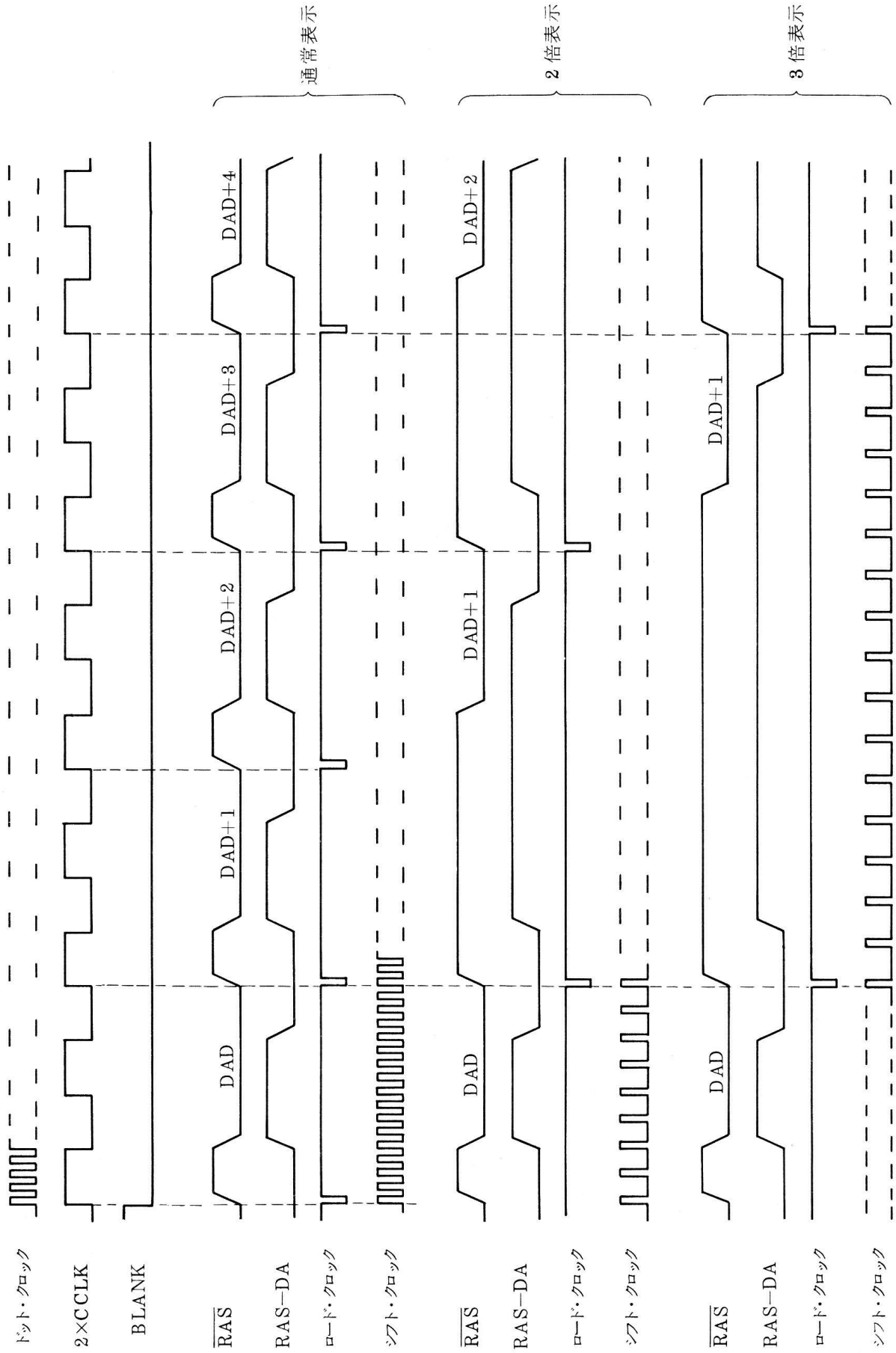
設定した拡大係数に従って、水平方向に対しては表示アドレス・サイクルを変更します。図4-5のように、通常表示時2クロックのサイクルが、2倍拡大時4クロック、3倍拡大時6クロックとなります。このとき、 $\overline{RAS}$ 出力は図4-6のように伸長区間には高レベルとなり、映像メモリ出力から映像直列信号を生成する並列-直列変換レジスタのロード・クロック発生タイミングの選択を容易にします。垂直方向に対しては図4-5のように、同一表示を行うライン数を変更します。

このように、GDCは拡大表示時、ワード・アドレス単位での表示アドレス制御のみを行いますので、並列-直列変換レジスタに供給するロード・クロックやソフト・クロックの発生周期を図4-6のように変更するには、外付け回路が必要です。

ただし、GDCからの表示アドレスの発生については図4-6のように、BLANK信号立ち下がり後の1文字分(2×CLK 2クロック)についてはロード・クロックの発生やBLANKの遅延のため拡大の有無によらず、通常のように2クロックで出力されます。

\*外付回路の具体例については、第6章ハードウェアの項を参照。

図 4 - 6 拡大表示時のタイミング

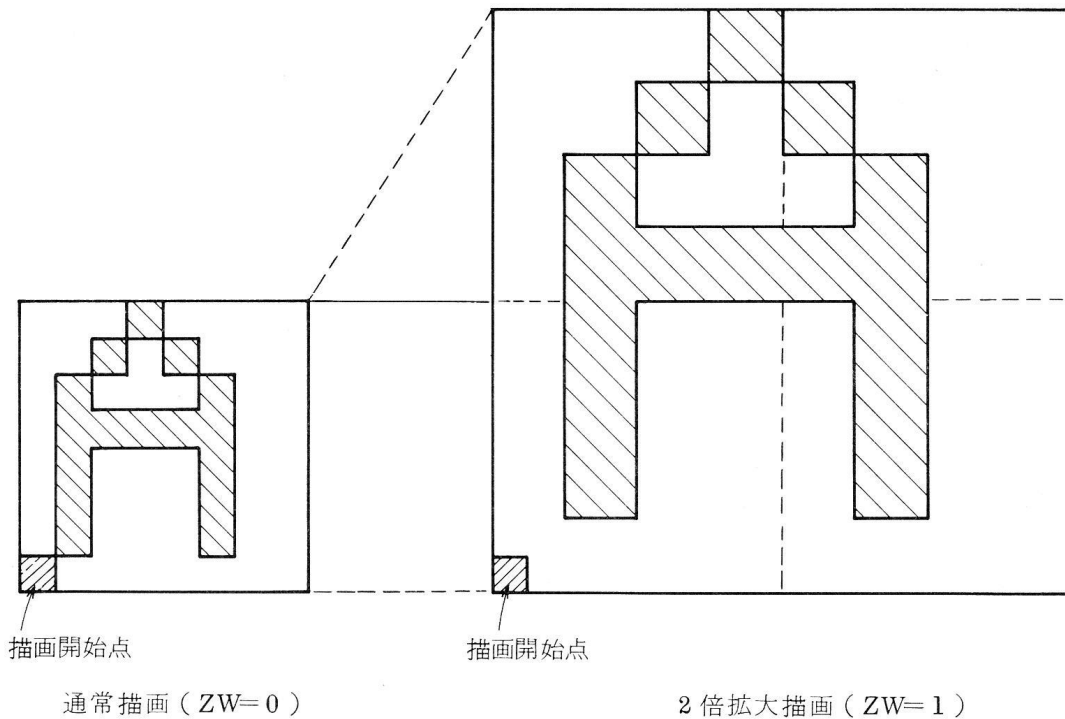


(注) RAS-DA の信号は  $\overline{\text{RAS}}$  から作ったロード・クロック発生用信号 (図 6 - 4 参照)



#### 4.4.2 拡大描画機能

グラフィック文字描画時に1から16までの整数倍での拡大描画が実行できます。ZOOMコマンドの第1パラメータ下位4ビットZWによって拡大係数を決定します。拡大係数に従って、縦横両方向に等倍に拡大描画されます。



## 4.5 外部同期機能

MASTER/SLAVE コマンドによってGDCをスレーブに設定すると、VSYNC出力端子がEX.SYNC入力端子となり外部からの同期信号の受け付けが可能となります。GDCはこのEX.SYNC入力信号の立ち下がり時に次のように同期信号発生部の初期化動作を行います（図4-7参照）。

スレーブGDCにおいて、外部からの同期信号をクロックの立ち上がり時にスレーブGDC内に取り込み、立ち下がり有無を検出した後、1クロック後にI.SYNCを発生し、同期信号発生部における諸カウンタ、レジスタの値を次のような値に初期設定します。

- ① 水平カウンタ HFP時の  $HC = 3$
- ② 垂直カウンタ VBP時の  $VC = 0$ 、第1フィールド状態

マスタとスレーブGDCのクロックは、次のような条件を満足していれば周波数が異なっても構いません。

- ① マスタGDCのVSYNC信号立ち下がり時のマスタGDCクロックと、スレーブGDCクロックの位相が合っていること。
- ② 1水平走査時間および、1フレーム（フィールド）走査時間はマスタ/スレーブ共に同一であること。

従って、マスタとしてGDCではなくテレビ・カメラなどの同期信号を発生する機器を接続することも出来ます。

CSRFORMコマンドの第3パラメータで設定するCEビットが“1”のときには、以下のコマンド発行とは無関係に、常時、外部同期動作を起動します。

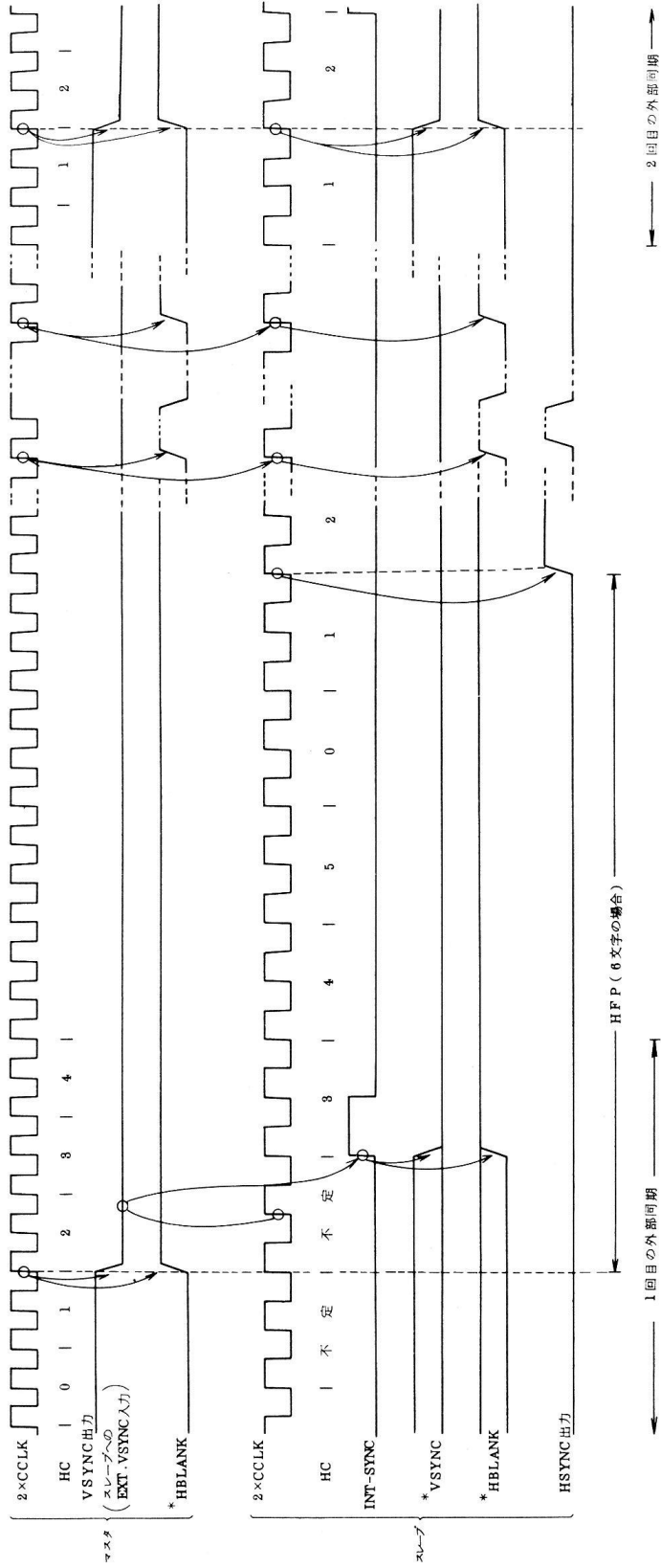
また、CEビットが“0”のときには、次の期間のみ外部同期動作を起動します。

RESET1またはSTOP1、 $DE = 0$ としたSYNCコマンド発行から

RESET2、RESET3、STOP2、 $DE = 1$ としたSYNCまたはSTARTコマンド発行までインタレース・モードが設定されている場合、マスタGDCはこの期間、第1フィールドにおけるVSYNC出力を発生します。これはスレーブGDCが、外部同期信号による初期設定時、第1フィールド状態となるためです。

このため、EX.SYNC端子に加える同期信号は第1フィールドにおけるVSYNC信号（インタレース走査時）となるように外部で第2フィールドのVSYNCを抜くようにしてください。

図 4-7 外部同期タイミミング

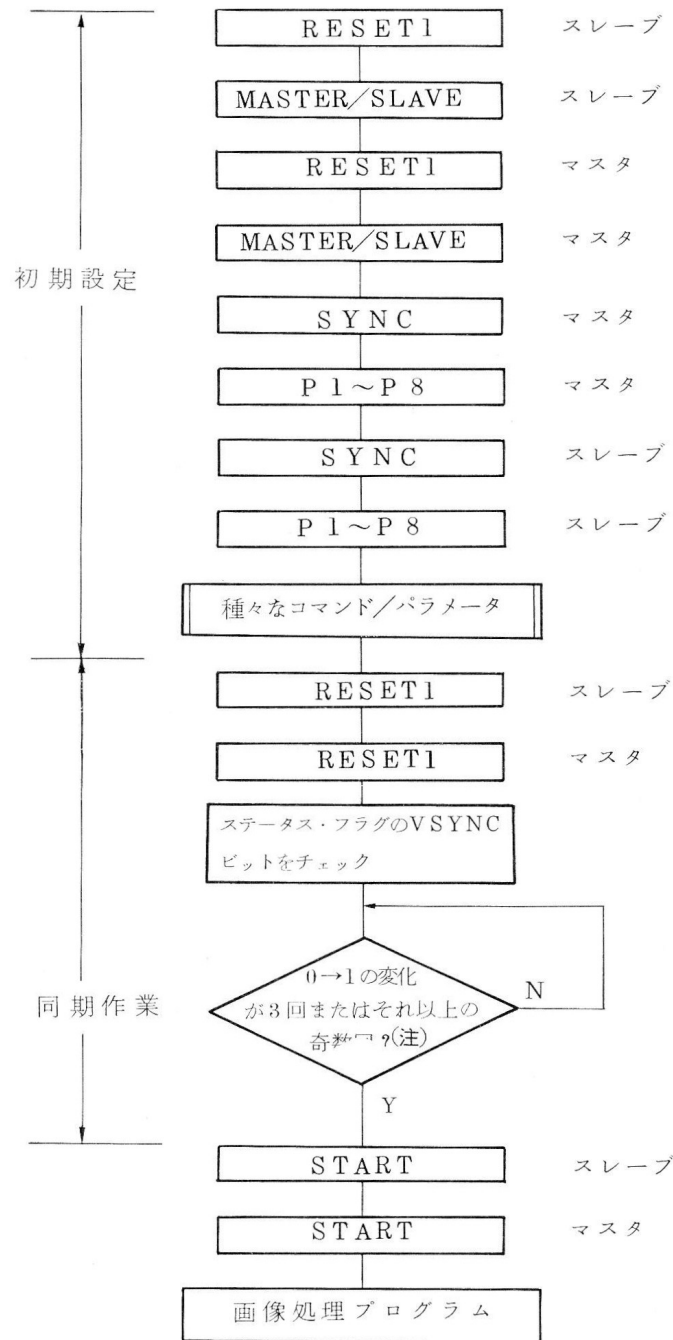


HC : GDC 内部で使用している水平同期用のカウンタで, カウント値が 1 と 2 の間で外部への出力状態が変化します.

INT.SYNC : GDC 内部で使用している信号であり, この信号がアクティブになったときに外部への出力 HC = 8 にプリセットされます. そのためスレーブ・モードに使用した場合には HFP を 4 文字以上に設定する必要があります.

\*HBLANK : } 1 水平期間, 1 垂直期間を定義するために記述した信号  
 \*VSYNC : }

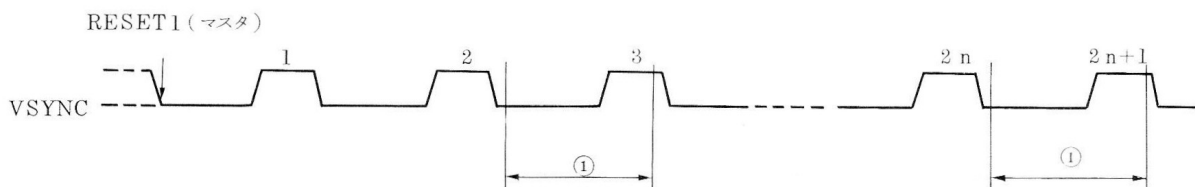
インタレース走査時，GDC以外の機器から外部同期信号を供給する場合には，第1フィールド時のVSYNC信号を抽出して供給する必要があります。マスタ，スレーブが共にGDCである場合の同期動作は，次のフロー・チャートに示すように，初期設定およびRESET1コマンド発行後，ステータス・フラグのVSYNCビットをチェックすることにより，簡単に行えます。



注意. この判定は次のようにしても同様の効果があります。

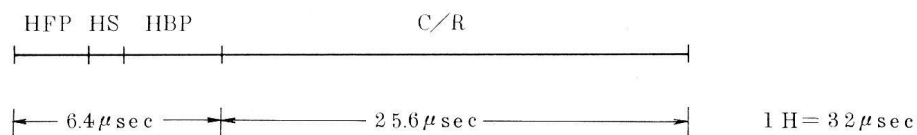
“1→0の変化が2回またはそれ以上の偶数回?”

このため，STARTコマンドを書き込むのは下図①のタイミングで行います。



## 4.6 クロック周波数の設定

GDCに与えるクロック周波数は1水平走査期間内の表示時間C/Rと、そのときの映像メモリ・アクセス数  $\left( \frac{\text{水平方向表示ドット数}}{\text{1ワード当りのビット数}} \right)$  によって決定されます。



たとえば、上図のように1水平走査時間(1H)が32μsecの高解像度モニタテレビを使用し、512ドット(16ドット×32回メモリ・アクセス)の表示を行うとき、表示時間を25.6μsecに選択すると、1回のメモリ・アクセスに要する時間は

$$25.6 / 32 = 0.8 \mu\text{sec}$$

表示サイクルはDAD+2モードを使用しない通常表示では2クロックであるので、クロック周波数は2.5MHzとなります。

種々なモニタ・テレビ、解像度のときのクロック周波数の目安を次表に示します。

モニタ・テレビジョン	15.75 kHz		23.6 kHz		31.5 kHz		63 kHz	
水平周波数(H)	15.75 kHz		23.6 kHz		31.5 kHz		63 kHz	
映像増幅器帯域幅	~15 MHz		~25 MHz		~40 MHz		~100 MHz	
グラフィック・モード (ノン・インタレース)	256×256	1024×256	384×384	1536×384	512×512	1024×512	512×1024	1024×1024
グラフィック・モード (インタレース)	256×513	1024×513	384×769	1536×769	512×1025	1024×1025	512×2049	1024×2049
文字モード (ノン・インタレース)	16文字×24行	64文字×24行	24文字×24行	96文字×24行	32文字×48行	64文字×48行	32文字×96行	64文字×96行
通常使用時の周波数	0.625MHz	2.5 MHz	1.25 MHz	5 MHz	2.5 MHz	5 MHz	5 MHz	—
DAD+2設定時の周波数	—	1.25 MHz	0.625MHz	2.5 MHz	1.25 MHz	2.5 MHz	2.5 MHz	5 MHz
IMAGE設定時の周波数	1.25MHz	5 MHz	2.5 MHz	—	5 MHz	—	—	—
ドット周波数								
6ドット/アドレス	1.875MHz	7.5 MHz	3.75 MHz	15 MHz	7.5 MHz	15 MHz	15 MHz	30 MHz
9ドット/アドレス	2.8 MHz	11.25MHz	5.625MHz	22.5 MHz	11.25MHz	22.5 MHz	22.5MHz	45 MHz
16ドット/アドレス	5 MHz	—	10 MHz	—	20 MHz	40 MHz	40 MHz	80 MHz
24ドット/アドレス	7.5 MHz	—	15 MHz	—	30 MHz	—	60 MHz	—

—は使用不可を示す。

1水平表示期間内の表示ドット数を一定とし、1回の映像メモリ・アクセスで16ドットの表示を行わせるグラフィック表示モードにおけるGDCのクロック周波数は、1回の映像メモリ・アクセスで水平表示方向8ドットのみを表示を行う文字表示モードにおける、クロック周波数と比較して1/2の周波数となります。

GDCクロック周波数を同一としたとき、水平周波数が低いモニタTVでは水平周波数が高いモニタTVと比較して横方向のドット数を多く取ることができりますが、縦方向ドット数、即ち、走査線本数は、垂直周波数を一定とすることから、少なくなります。インタレースを行う事によって走査線本

数を増加させることはできますが、高品質なモニタTVを使用しないと、表示品質が劣化します。

解像度が低くとも良い応用(256×256)の場合、GDCのクロック周波数は低くなります。即ち、描画速度との関連で述べると、解像度の高低に拘らず単位画面あたりの描画速度は同一です。ドットあたりの描画速度が問題となる場合にはIMAGEモードを選択することによってGDCのクロック周波数を倍とし、実質的な描画速度を2倍に引きあげるとともに、同期信号設定に融通性を持たせることが可能となります。

逆に、1024×1024の高い解像度を持つ画面をノン・インタレースで実現しようとする、GDCのクロック周波数が10MHz以上となり、GDCの最高動作周波数の上限を超えてしまいます。この場合、表示アドレスの増加が“2”ずつ行われる“DAD+2”の動作モードを設定し、1クロックごとにデータを取り出せるようにすることによって、GDCのクロック周波数を通常使用モードの“1/2”に落として使用することができます。

従って、通常使用モードにおけるGDCのクロック周波数が低すぎたり高すぎたりする場合であっても、“DAD+2”または“IMビット”の設定によって、GDCのクロック周波数を各々“1/2”、“2”倍に変更してGDCの動作範囲内で使用することができるようになります。

IMやDAD+2のフラグはSCROLLコマンドによって設定します。図3-6に示すように、IM=1としたとき、表示サイクルは4クロックであり通常の倍になります。DAD+2=1としたときには逆に表示サイクルでの表示アドレスの進み方が+2ずつですので、ハードウェアで各クロックにアドレスを補うことにより、表示サイクル(2クロック間)に連続した2アドレスを同時に読み出すことができます。

1語のドット数は後述のように16/8/4/2/1ドットと選択することができます。16ドット/語のときGDCのクロックを $f_{GDC}$ とすればドット構成を $16/2^n$ ( $n:0,1,2,3,4$ )とした場合の $f$ は

$$f = f_{GDC} \times 2^n$$

となります(4.11参照)。

## 4.7 映像メモリの構成

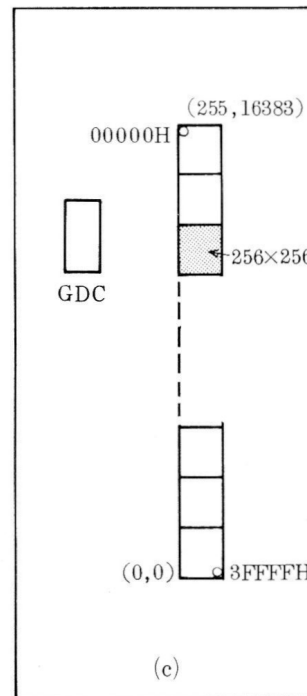
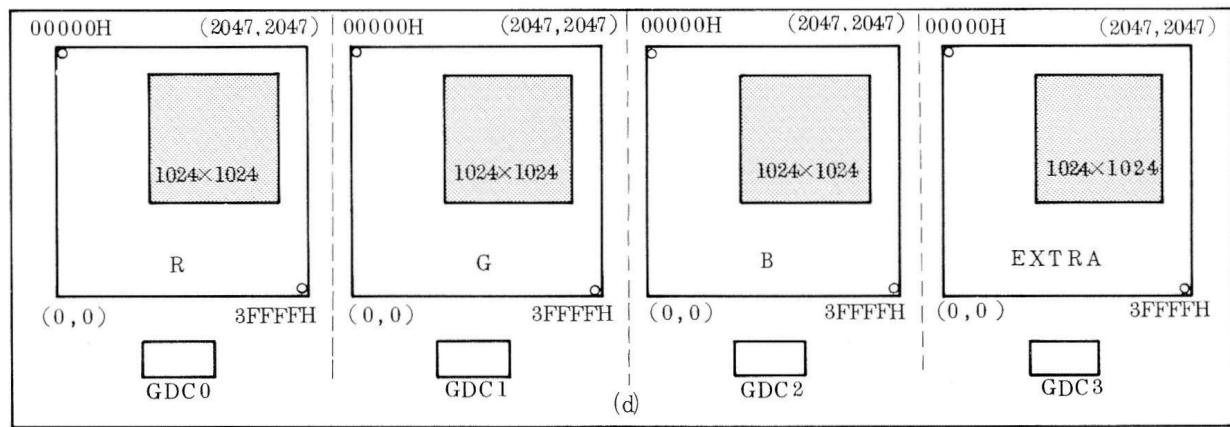
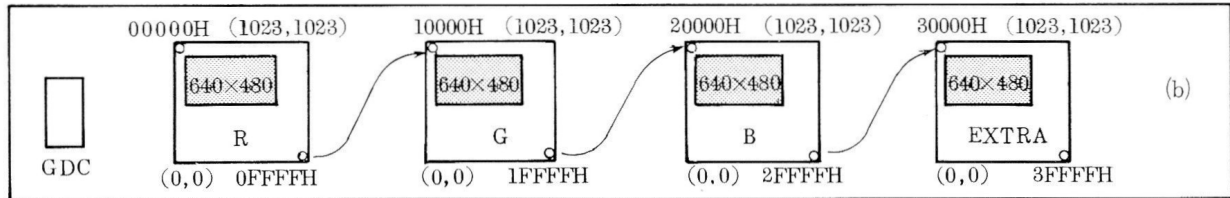
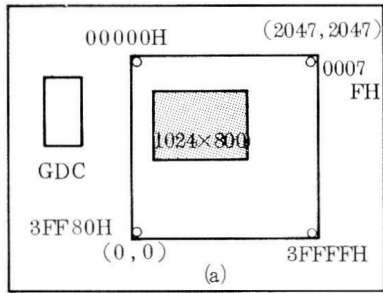
グラフィックモード時 256KW×16ビット，文字／グラフィック混在モード時 64KW×16ビット，文字モード時 8KW×13ビットの映像メモリを直接制御できます。映像メモリの水平方向アドレス数はPITCHコマンドによって511までの任意値に設定でき，それとは独立に，水平方向の表示領域をSYNCコマンドのC/R値によって設定できるので，映像メモリの構成を図4-8に示すように，種々に変化させて使用することが出来ます。

映像メモリの水平方向最大ドット数は8176ドットまで定義できます。垂直方向に関しては，使用している映像メモリ容量によって決定されます。水平方向の表示域は，C/R値が偶数でなければならないという制約から，32ドット（2文字）単位で最大4064ドットまで定義できます。

映像メモリ・プレーンを数枚用意し，描画時には，1枚のプレーンのみを選択し，表示時には，映像メモリ・プレーンから同時に映像出力を取り出し，多階調，多色表示することが出来ます。（b）は，1024×1024ドットのメモリ・プレーン4枚を使用し，640×480ドットを16色で表示するときのメモリ構成を示しています。

- (a) : 2048×2048ドット 表示域 1024×800ドット
- (b) : 1024×1024ドット 表示域 640×480ドット 16色カラー
- (c) : 256×16384ドット 表示域 256×256ドット 64枚
- (d) : 2048×2048ドット 表示域 1024×1024ドット 16色カラー GDC4個使用

図 4-8 映像メモリの構成





## 4.8 インタレース時のL/Fの設定法

GDC内の同期信号発生用カウンタは、VSYNCの立ち上がりから次のVSYNCの立ち上がりまでを1周期として計算します。従って、垂直同期関係を定義するパラメータVFP、VS、VBP、L/Fは1周期内のライン数によって設定します。

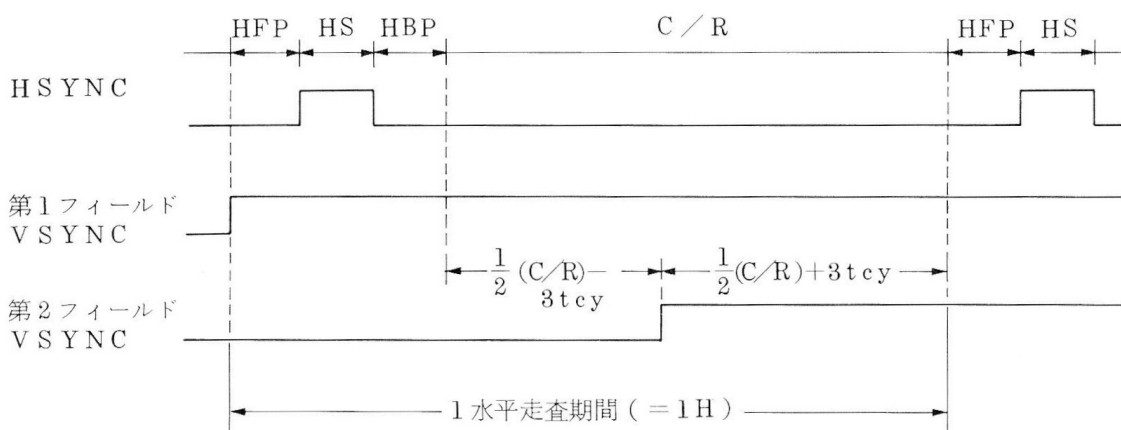
ただし、インタレース時における第1フィールドと第2フィールドでのVSYNC信号の出力は、図4-9のようになっています。

すなわち、

第1フィールド：HFPの始まりに同期してVSYNCが変化します。

第2フィールド：C/Rすなわち表示時間に相当する期間の真ん中より3クロック前に同期してVSYNCが変化します。

図4-9 インタレース時におけるVSYNC信号



このような出力タイミングを持つインタレース時のL/F設定値については次の2つの方法があります。

1 フレーム当たりの表示ライン数を奇数本としたい場合には

- ① SYNCコマンドの第6パラメータで指定するVLビット=0
- ② L/R設定値を偶数(最下位ビット=0)

とします。

1 フレーム当たりの表示ライン数を偶数本としたい場合には

- ① SYNCコマンドの第6パラメータで指定するVLビット=1
- ② VFP設定値を偶数(最下位ビット=0)

とします。

GDC内のL/Fの設定用のレジスタは10ビットですので、ノン・インタレースであれば最大1024本、表示ライン数奇数本設定時最大2045本、偶数本設定時2046本/フレームまで設定できます。

なお、表示ライン数を奇数本とした場合には、SCROLL コマンドによって送出するSLパラメータは

$$SL \geq L/F + 1$$

となるように設定してください。

1 水平走査期間内でのVSYNC発生タイミングは図4-9のようになっていますので、厳密には第1フィールドVSYNC周期は以下の値(クロック)だけ第2フィールドに比べ長くなります。

$$\begin{aligned} \text{①の場合} \quad & HFP+HS+HBP+(1/2(C/R)-3tcy)-(1/2(C/R)+3tcy) \\ & = HFP+HS+HBP-6tcy \end{aligned}$$

従って、第1フィールドと第2フィールドの長さは厳密には等しくありませんので、モニタTVの特性によってはインタレースを行った場合、フィールド間で等間隔の走査とならず、片寄った表示となることがあります。

\*第6章 インタレース時のVSYNC発生タイミングの修正についての項参照。

図4-10 VSYNCの発生タイミング(表示ライン数奇数本設定時)

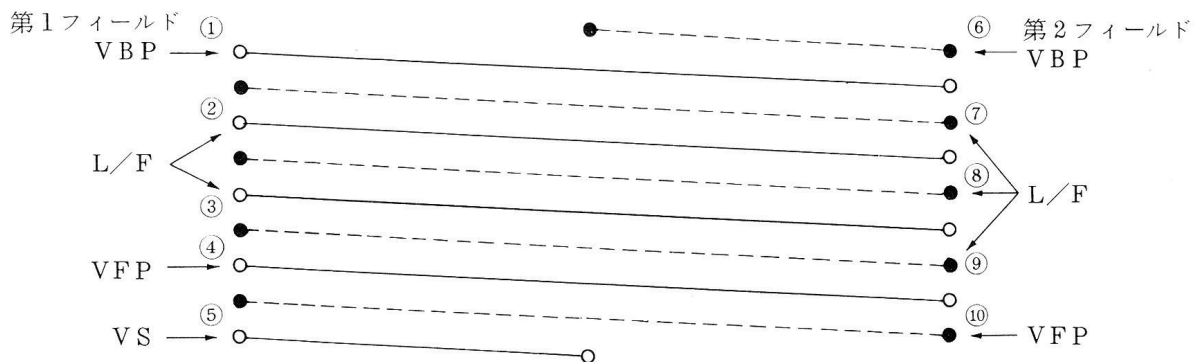
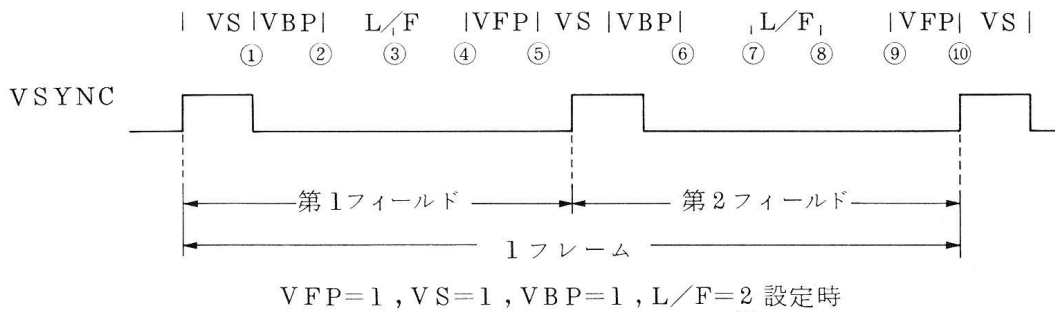
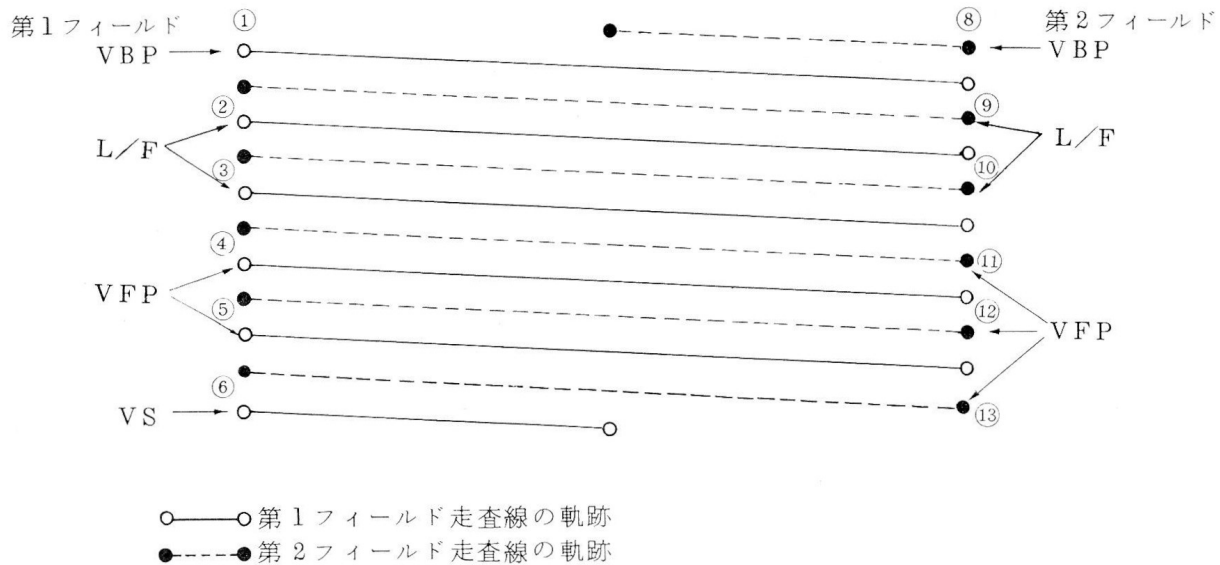
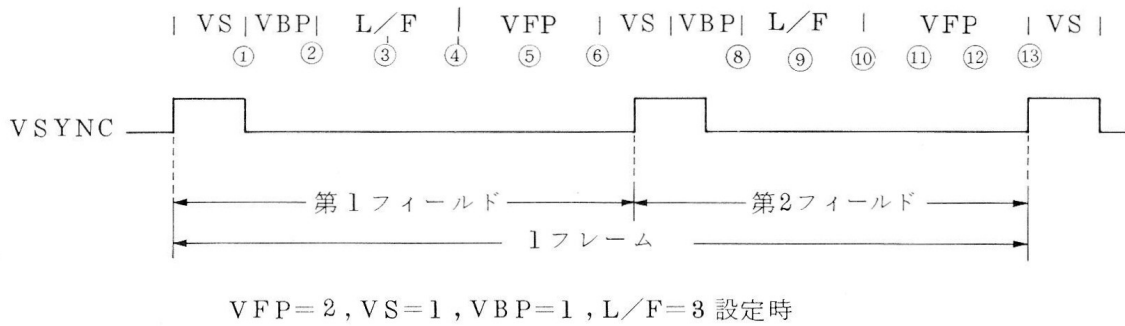


図 4-12 VSYNCの発生タイミング (表示ライン数偶数本設定時)



## 4.9 NTSC基準同期信号の発生法

GDCは、応用の種類によって水平および垂直の同期信号周期を任意に選択できる工業用モニターテレビに対するプログラマブルな同期信号発生器として使用できるほか、NTSC, PAL, SECAMのような通常のテレビジョン放送の同期信号基準ののつとつた同期信号を発生することも可能です。

さらに、GDCは外部同期端子EX.SYNCを有しているので、NTSC映像信号にGDCが制御する映像メモリから創造された映像情報を、完全に同期を取りながら、重畳する事が容易に実現できます。例えば、テレビジョン・カメラからの映像信号にGDCが制御する文字図形情報を直接、重ね合わせたり、その文字図形情報をテレビジョン・カメラを介さずに直接、ビデオ・テープ・レコーダに記憶させることができます。

以下に、日本、米国等で標準放送規格として使用されている「NTSC」を例にとって、同期信号発生用各種パラメータの設定法、および、GDCに供給する基本クロック(2×CCLK)の周波数の選定法について説明します。

NTSCは白黒/カラー間の両立性が保たれており、その副搬送波の周波数は3579545Hzです。カラーへの移行を考慮した場合にはこの副搬送波周波数を基準として、同期信号を発生することが望まれます。

副搬送波周波数( $f_{sc}$ )と水平周波数( $f_H$ )との間には、以下に示す関係があります。

$$f_{sc} = \frac{455}{2} * f_H \dots\dots\dots(1)$$

上式は  $f_{sc}$  を 3579545 Hz の 2 倍の周波数，即ち，7159090 Hz としたとき 455 回だけ  $2 * f_{sc}$  波形を計数した時間が  $1 / f_H$  であることを示しています。

ここで  $f_{sc} = 3579545$  Hz は不変とし容易に実現可能なカウンタを用いた周波数てい降器を考えます。(1)式の分子は "5" の倍数であり，分母は "2" であるので  $f_{sc}$  の "2/5" の "2" の倍数の周波数を計数用周波数として選択すれば  $f_H$  との間に整数関係が保たれます。

$f_{sc}$  を基本周波数として生成される計数用周波数を  $f_{CNT}$  とすると

$$f_{CNT} = \frac{455}{2} * f_H * \frac{2 * K}{5}$$

$$= 91 * f_H * K \dots\dots\dots(2)$$

(K は整数)

(2)式は， $f_{CNT}$  として  $f_{sc} * \frac{2 * K}{5}$  の周波数を選択したとき  $91 * K$  回の計数を行うことによつて 1 H 期間が導出されることを示しています。

GDC における計数は  $2 * CLK$  クロックの 2 クロックに 1 回だけ行われることに注意し，最高動作周波数を考慮に入れると，(2)式において "K = 2" に選択することが実用的です。

このとき

$$f_{CNT} = 182 * f_H$$

$$= f_{sc} * \frac{4}{5}$$

$$= 2863636 \text{ (Hz)}$$

この周波数を GDC のクロックとして使用すると

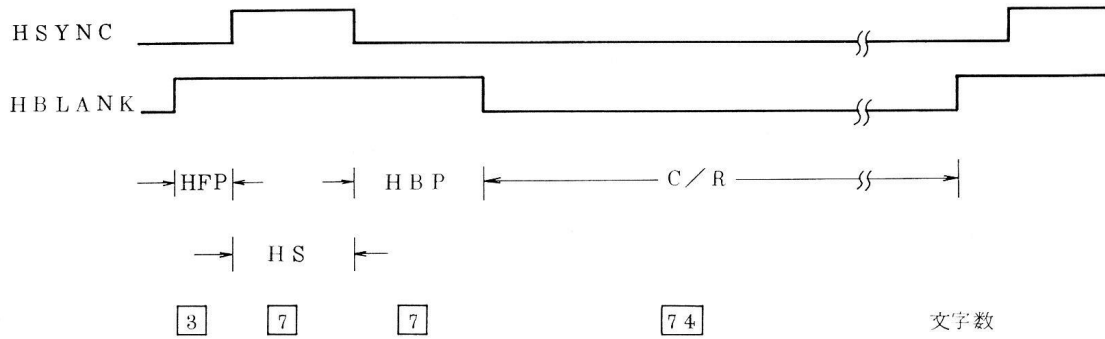
HFP+HS+HBP+C/R で表現される 1 水平期間は 91 文字となります。

1 文字区間，即ち，2 クロックは  $0.6984127 \mu s$  であるので諸元表に示される値を GDC の計数単位である "文字数" に置換すると諸元表の GDC 欄に示す値となります。

NTSC 規準のうち垂直関係，等価パルス，カラー関係を除いた各諸元は次のとおりです。

	NTSC	GDC
走査線数/フレーム	525 本	525 本
フィールド周波数	60 Hz	60 Hz
水平周波数 $f_H$	15734.26 Hz	15734.26 Hz
公称周期 H	63.5555 $\mu s$	63.5555 $\mu s$
水平ブランキング期間	10.5~11.4 $\mu s$	11.873017 $\mu s$
HBLANK		17 文字
水平同期パルス幅	4.19~5.08 $\mu s$	4.888895 $\mu s$
HSYNC		7 文字
水平フロント・ポーチ	1.27~2.22 $\mu s$	2.0952383 $\mu s$
HFP		3 文字
水平バック・ポーチ+	9.2~10.3 $\mu s$	9.777779 $\mu s$
水平同期パルス幅		14 文字

○ 水平関係

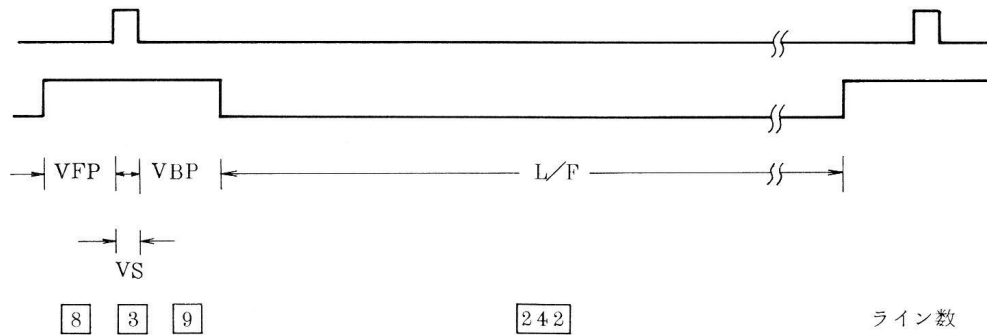


○ 垂直関係

1 フィールド内の表示ライン数を 242.5 本

垂直ブランキング期間を 20 本

垂直同期パルス幅を 3 本とすると,



I : インタレース・モード設定フラグ = " 1 "

S : シュリンク・モード設定フラグ = " 1 "

以上に示した同期信号発生用設定値は, SYNC コマンドのパラメータ送出によって GDC に設定します.

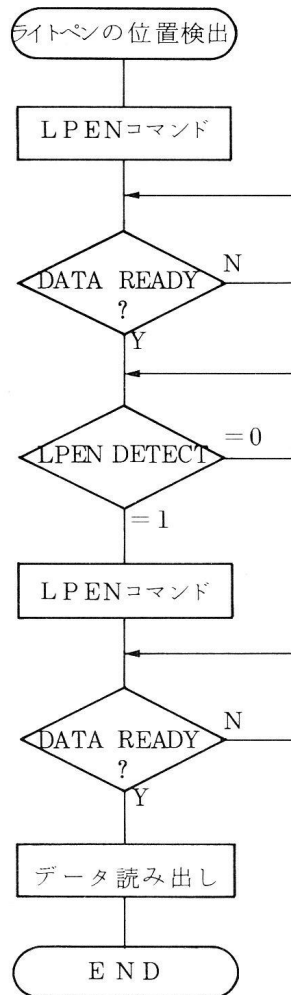
ただし, 以上の設定は水平, 垂直のフロント・ポーチ, バック・ポーチを規格に従って決定したものであって, GDC を単なる NTS C 同期信号発生器として使用する場合を想定しています. そのため, この設定状態で映像メモリからの映像信号を送出すると上下左右の映像が画面からはみ出ることに注意する必要があります.

このような表示領域の欠けを除去するには水平/垂直のフロント・ポーチおよびバック・ポーチの設定値を使用する CRT にあわせて吟味する必要があります.

## 4.10 ライトペンによる位置検出機能

L PEN 入力端子にライトペンによって検出した光信号を接続することにより, 位置検出を行うことが出来ます. 画面表示開始時から第 1 回目の L PEN 信号が入力されると, 次の表示サイクルにおいて, そのときの表示アドレスがライトペン・アドレス・レジスタ (LAD) と比較され, 一致していなければそのときの表示アドレスが LAD に転送されます. 一致していれば, ライトペン・ステー

タス・フラグ (LPEN DETECT) が "1" となります。同一画面内の走査期間において2回目以降のLPEN信号あるいは、ステータス・フラグがセットされてからリセットされるまでの間に発生するLPEN信号は全て無視しますが、LPEN DETECTフラグがリセットされるとすぐに位置検出をはじめますので、ライトペンで座標入力をする場合はLADを読む前に一度ダミーのLPENコマンドを発行して、LPEN DETECTをリセットしてからLPEN DETECTのチェックに行くようにしてください。



従って、ライトペン・ステータス・フラグが "0" (前回検出したライトペン・アドレスと一致していない) の場合であっても、ライトペン・アドレスには最新のライトペン・アドレスが入っています (ただし、LPEN信号が入力された後に、LPENコマンドを発行した場合)。

CPUは、LPEN DETECTフラグが "1" であることを検知した後、LPENコマンドによって3バイトに分割されたLADをFIFOに転送し、DATA READYフラグを "1" とすることによって読み出し可能状態とするとともに、LPEN DETECTフラグをリセットします。

このように、ライトペンによる表示アドレス位置検出では、一画面に1度だけ検出する表示アドレスが、2画面期間に亘る2回の検出において同一であることを確認する回路を内蔵していますので、部屋灯などの光入力によって誤動作することはありません。ただし、ライトペン自身のスイッチング遅れや、GDC内および、インタフェース回路部でのデジタル的な遅れを見込み、読み出されるLAD値に修正を加えるソフト的な措置が必要です。

正常な表示アドレスは表示期間 ( C / R ) およびフロント・ポーチ部 ( H F P ) に継続して出力されます。ライトペン使用の際には S Y N C コマンドの H F P の値に制限がありますので、注意してください。

## 4.1.1 1語を8/4/2/1ビットにする方法

以下のようなフローチャートに従ったプログラムにより、1語を8ビット、4ビット、2ビット、1ビット (  $16/2^n$ ビット ;  $n=0, 1, 2, 3, 4$  ) 構成にもすることが可能です。

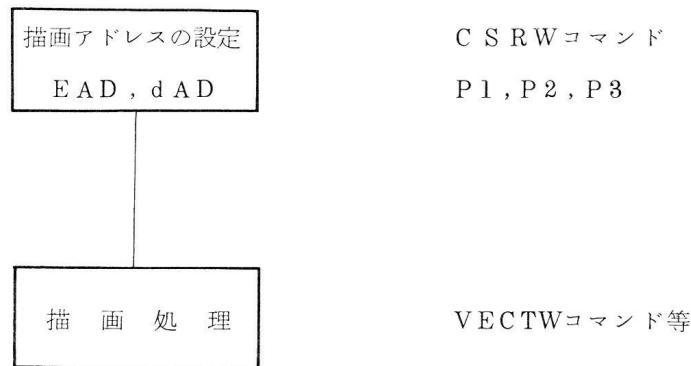
(i) 通常の場合 ( 1語16ビットの場合 )

(i) 描画アドレスの設定 ( E A D , d A D )

C S R W コマンドにより設定し、d A D は P 3 の上位4ビットにより設定します。

(ii) 描画処理

V E C T W 等の描画コマンドを送出します。



(2) 1語8ビット、4ビット、2ビット、1ビット構成の場合

(i) 描画ワード・アドレスの設定 ( E A D )

C S R W コマンドにより設定しますが、P 3 の上位4ビットで設定する d A D は次の M A S K コマンドで再設定しますので、どんな値を入れてもかまいませんが、" 0 " を設定すると、G D C 内でのパラメータ解釈時間が最小となります。

(ii) 描画ドット・アドレスの設定 ( d A D )

M A S K コマンドにより設定します。d A D と設定値の関係を表4-5に示します。なお、同じ方法により1語16ビット構成の場合にも d A D を設定することができます。

(iii) 描画処理

V E C T W 等の描画コマンドを送出します。



CSRWコマンド  
P1, P2, (P3)\*1

MASKコマンド  
P1, P2

VECTW 等

\*1 文字／グラフィック混在モードやグラフィック・モードで上位アドレスA16, A17を使用しないシステムでは, P3を省略することができます。

なお, 具体例として1語を8ビットとして1B8H番地の7ドット目から2H番地の2ドット目まで, 1点鎖線を描画した例を図4-13に示します。また, この応用として1語4ビット構成とし, 一度にR, G, B同時描画することが可能なシステム構成例を図4-14に示します。

表4-5 dADとMASKコマンドのパラメータの関係

dAD	1W=16BITS <sup>(注)</sup>		1W=8BITS		1W=4BITS		1W=2BITS		1W=1BITS	
	P1	P2	P1	P2	P1	P2	P1	P2	P1	P2
0	01H	00H	01H	01H	11H	11H	55H	55H	0FFH	0FFH
1	02H	00H	02H	02H	22H	22H	0AAH	0AAH		
2	04H	00H	04H	04H	44H	44H				
3	08H	00H	08H	08H	88H	88H				
4	10H	00H	10H	10H						
5	20H	00H	20H	20H						
6	40H	00H	40H	40H						
7	80H	00H	80H	80H						
8	00H	01H								
9	00H	02H								
10	00H	04H								
11	00H	08H								
12	00H	10H								
13	00H	20H								
14	00H	40H								
15	00H	80H								

注意. 通常はCSRWの第3パラメータで設定



図 4-13 1語8ビットの場合の描画例

(a) プログラム例

```

CSRW      : 0100 1001B(49H)
           P1 : 1011 1000B(0B8H)      ; EAD=1B8H
           P2 : 0000 0001B(01H)
MASK      : 0100 1010B(4AH)
           P1 : 0100 0000B(40H)
           P2 : 0100 0000B(40H)      ; dAD=6, lW=8BITS
WRITE     : 0010 0000B(20H)      ; MOD=REPLACE
TEXTW     : 0111 1000B(78H)
           P1 : 0110 1111B(6FH)      ; 1点鎖線
           P2 : 0110 1111B(6FH)
VECTW     : 0100 1100B(4CH)
           P1 : 0000 1011B(0BH)      ; LINE, DIR=3, X=12D, Y=10D
           P2 : 0000 1100B(0CH)      ; DC=12D(0CH)
           P3 : 0000 0000B(00H)
           P4 : 0000 1000B(08H)      ; D=8D
           P5 : 0000 0000B(00H)
           P6 : 1111 1100B(0FCH)     ; D2=-4D(3FFCH)
           P7 : 0011 1111B(3FH)
           P8 : 0001 0100B(14H)     ; D1=20D(14H)
           P9 : 0000 0000B(00H)
VECTE     : 0110 1100B(6CH)
    
```

(b) 描画例

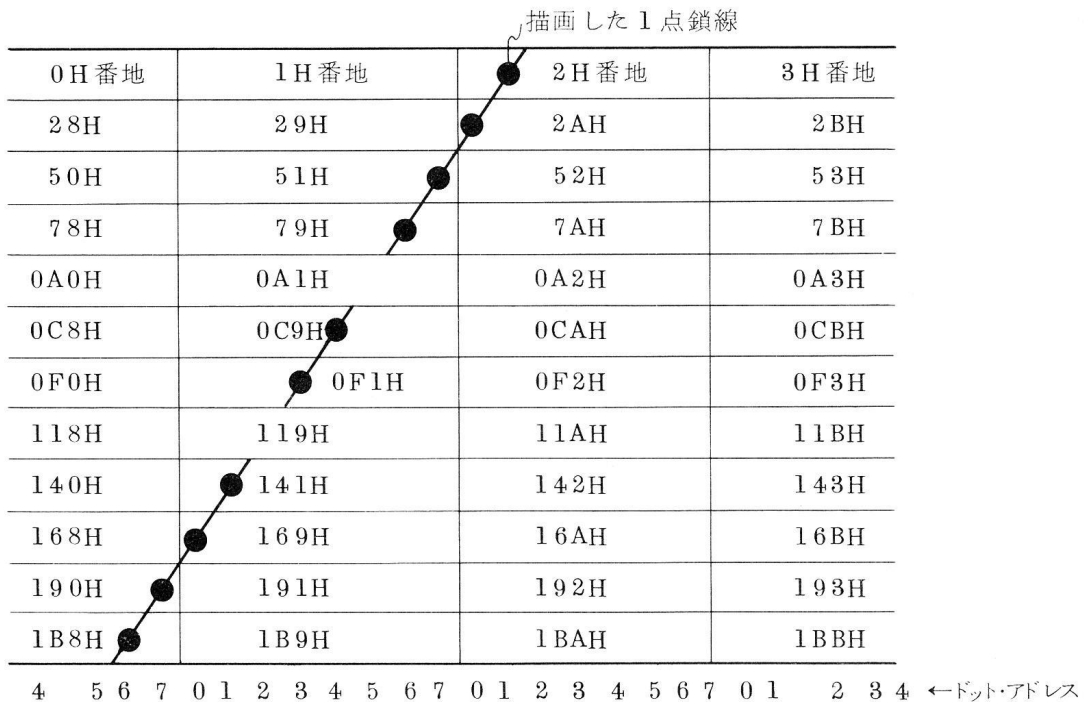
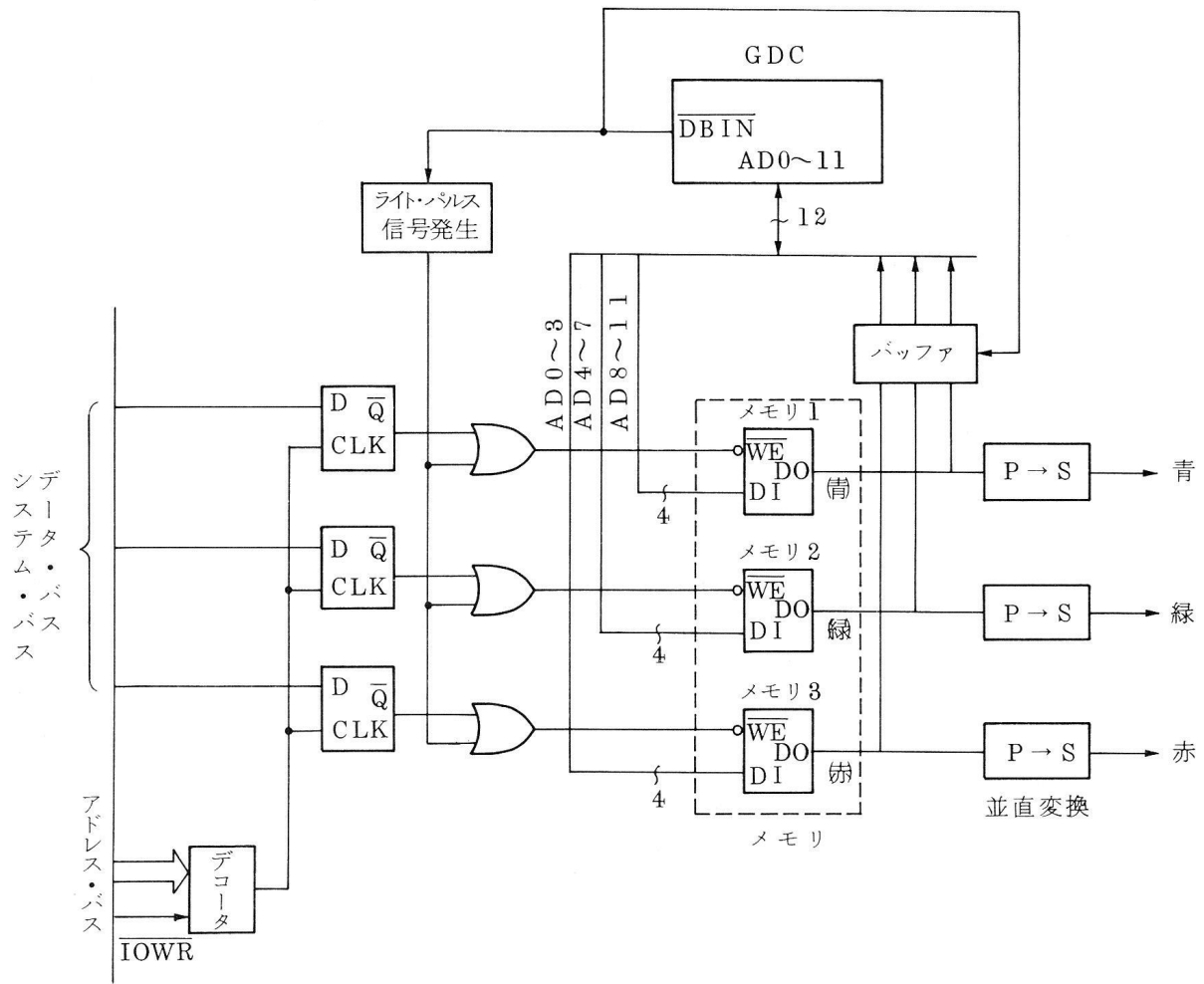
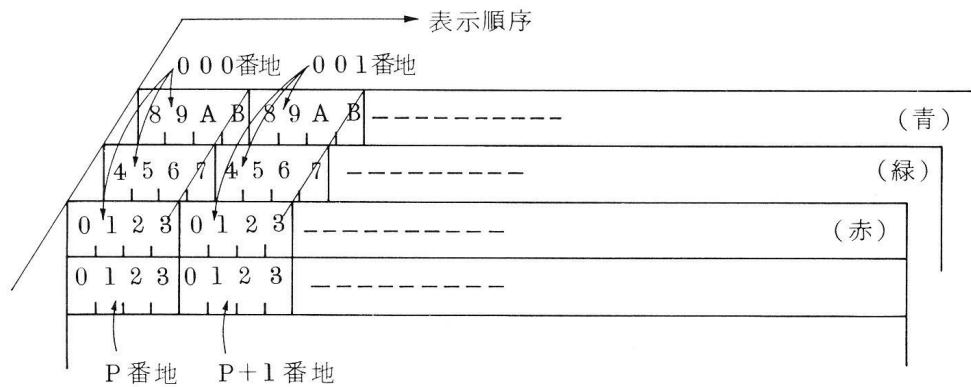


図4-14 1語4ビット構成RGB同時描画システム構成例

(a) システム・ブロック図



(b) メモリ構成



## 4.12 信号多重化出力端子の出カタイミング

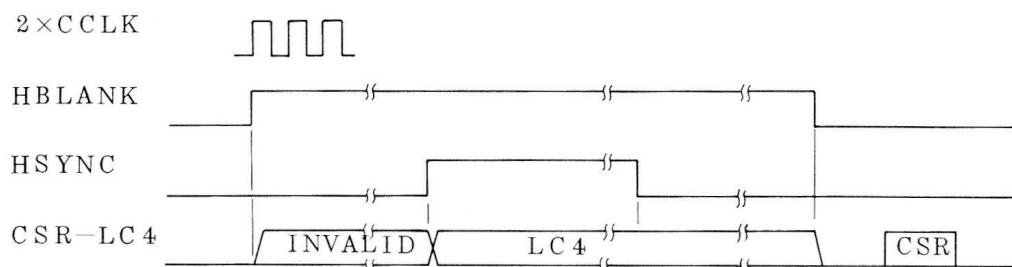
文字モードまたは文字／グラフィック混在モード時には，38ピンおよび39ピンには多重化された信号が出力されます。

端子	文字モード	文字／グラフィック混在モード
38	LC3	AT.BLINK-CLC
39	CRS-LC4	CSR-IMAGE

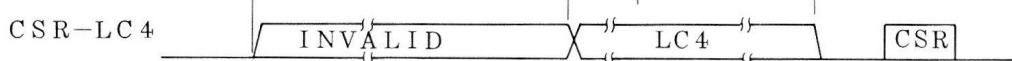
以下，各信号について詳細な出力タイミングについて記述します。

### (1) CRS-LC4 端子

#### (i) 通常時

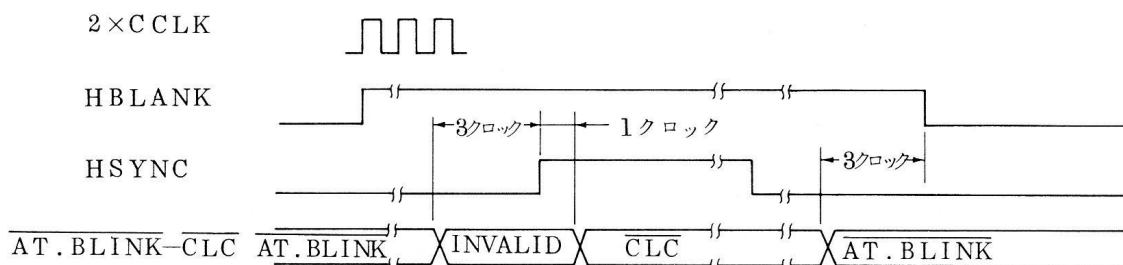


#### (ii) インタレース・シュリンク時

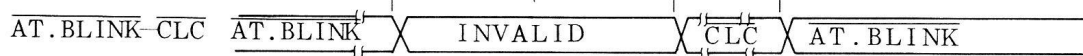


### (2) AT.BLINK-CLC 端子

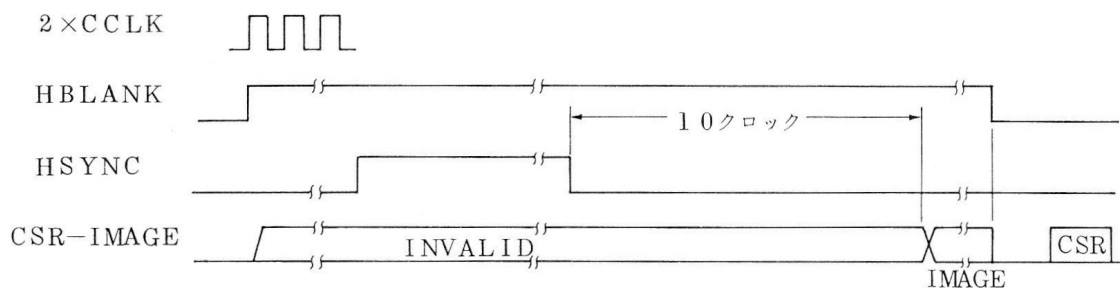
#### (i) 通常時



#### (ii) インタレース・シュリンク時



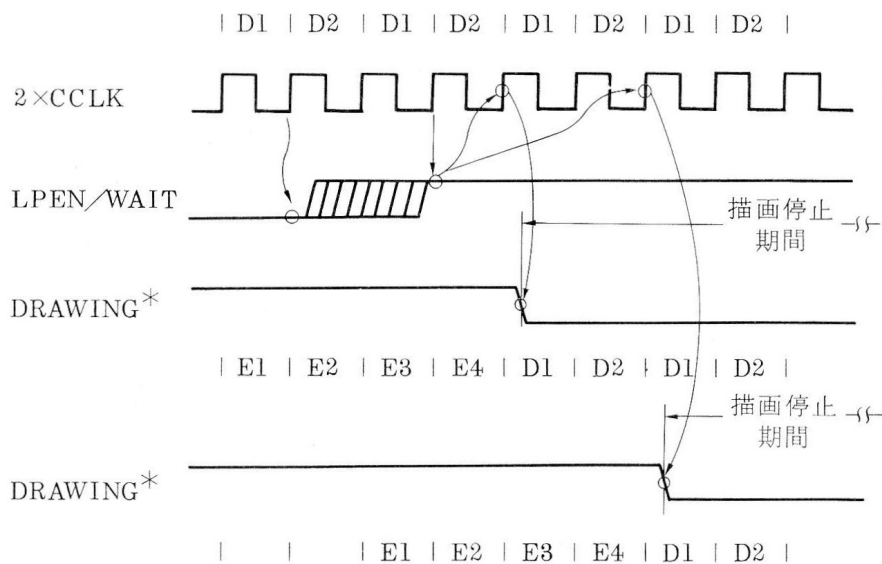
(3) CSR-IMAGE 端子 (画面分割時)



### 4.13 描画の一時停止タイミング

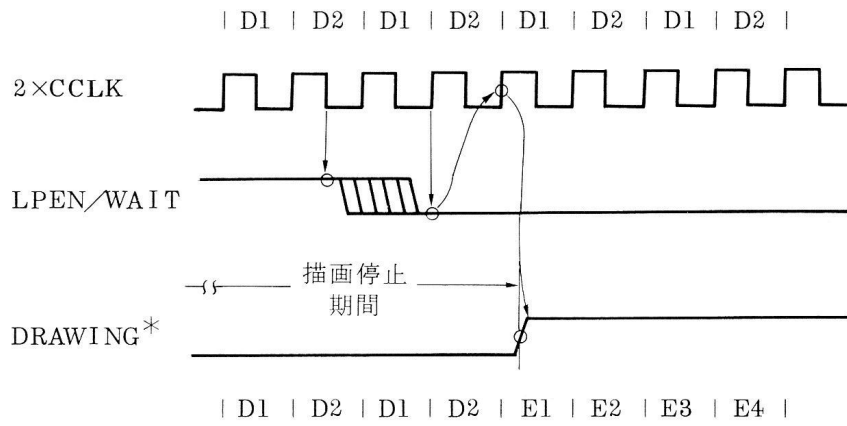
$\mu$ PD7220A では DS ビット (SYNC コマンドの第 5 パラメータの DB 7) を “1” に設定すると、LPEN/WAIT 端子へのハイ・レベル入力によって描画の一時停止を行うことができます。その際のタイミングは図 4-15, 図 4-16 に示すようになっています。

図 4-15 描画停止の受け付けタイミング



\*: 描画サイクルを示す内部信号

図 4-16 描画停止の解除タイミング



\*：描画サイクルを示す内部信号

## 第5章 ソフトウェア

### 5.1 描画手順

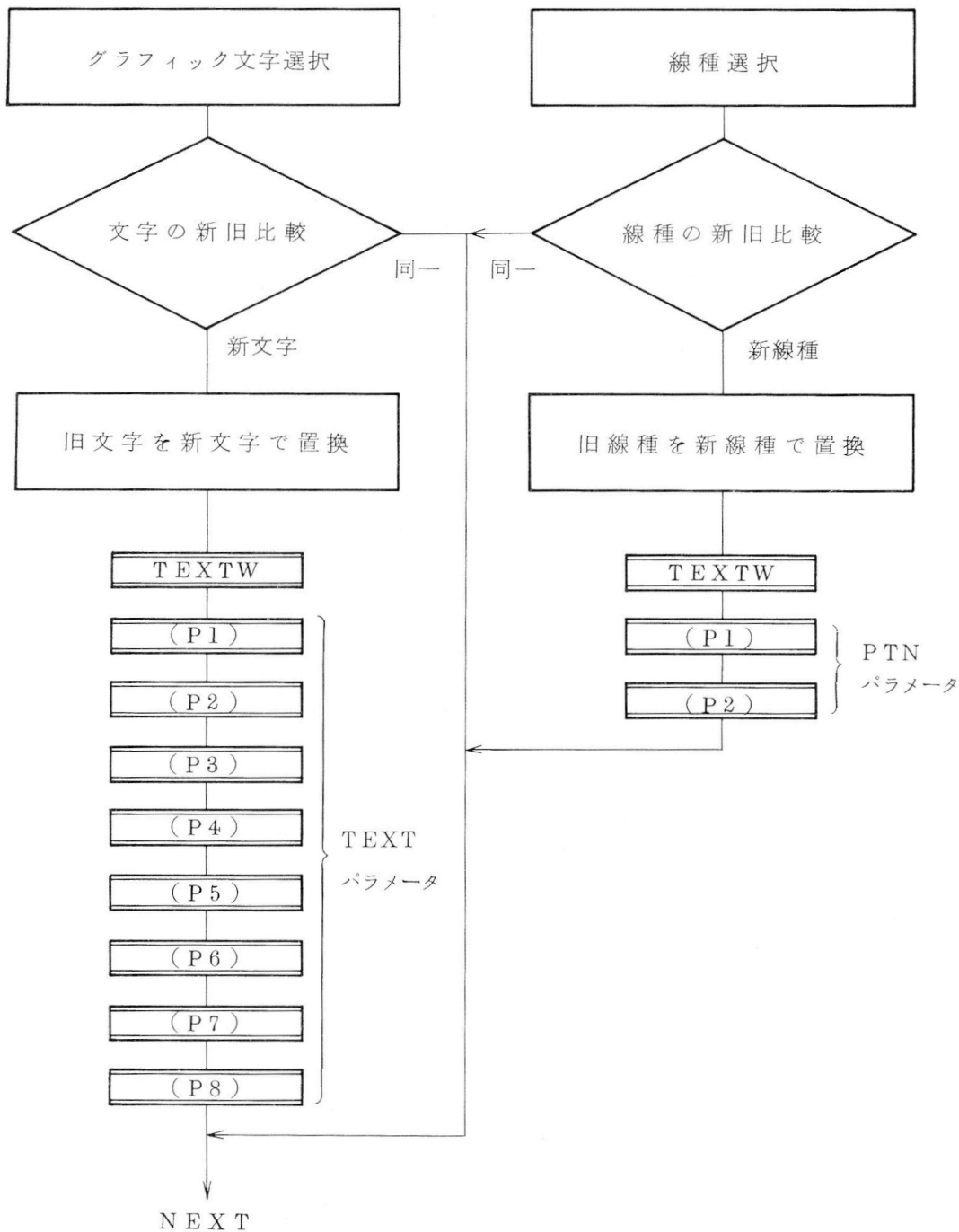
GDCに直線，円，四辺形などの描画を実行させるには，ディジタイザなどの座標発生器から得られた座標値を基にして，GDCが解釈できるレベルのコマンド／パラメータをCPUは作成する必要があります。このため概略，次のような段階を踏みます。

- (1) 実線，破線などの線種や，グラフィック文字のドット情報を選択する（TEXTWコマンド）。
- (2) REPLACEなどのドット修正モードを選択する（WRITEコマンド）。
- (3) 描画開始点を決定し描画開始絶対番地を算出する（CSRWコマンド）。
- (4) 直線，円などの描画種類や描画方向，描画パラメータを算出する（VECTWコマンド）。
- (5) 描画実行の指示をする（VECTE/TEXTEコマンド）。

(1)から(4)の選択や算出順はどのようであっても構いません。描画実行指示時に，必要な値がGDCに設定してあれば良いからです。

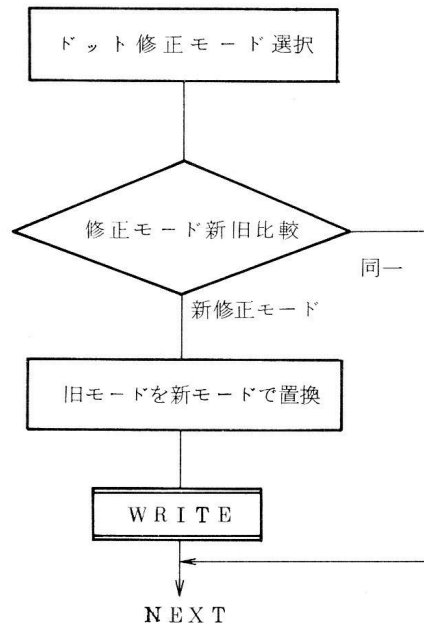
### 5.1.1 線種，ドット情報の選択

描画前にGDC内に設定されている線種やグラフィック文字のドット情報がどのような種類のものであるかを記憶しておきます。変更があったときにのみ新しい値をGDCに設定するためです。線種とドット情報とは，同一の内蔵RAM内に格納されているので，新しい線種が設定されたときには，古いドット情報は消去されてしまいますから注意が必要です。出来るだけ，GDCに送出するコマンド／パラメータ数を減らした方が，CPU処理を含めた全体的描画速度は向上します。



## 5.1.2 ドット修正モードの選択

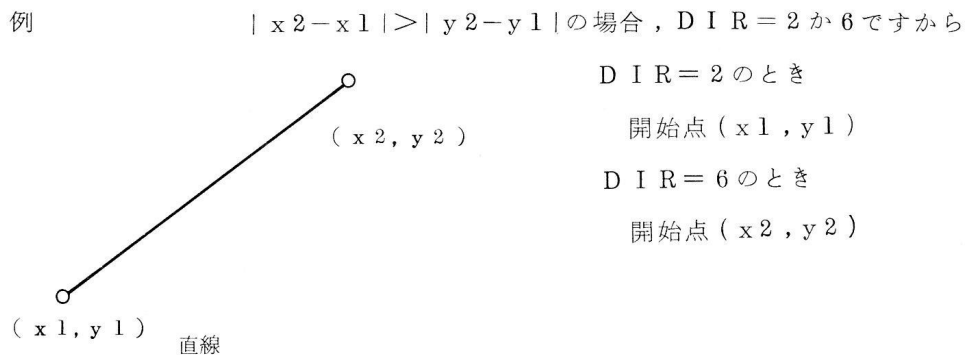
5.1.1と同様に、古いドット修正モードを記憶しておきます。



ドット修正モードはWRITEコマンドの下位2ビットMODで設定します。パラメータを付随させなければ、ドット修正モードの書き換え再設定のみを行いますからWRITEコマンドのWLHにはワード/ハイ・バイト/ロウ・バイトのいずれを選択しても構いません。

## 5.1.3 描画開始絶対番地の算出

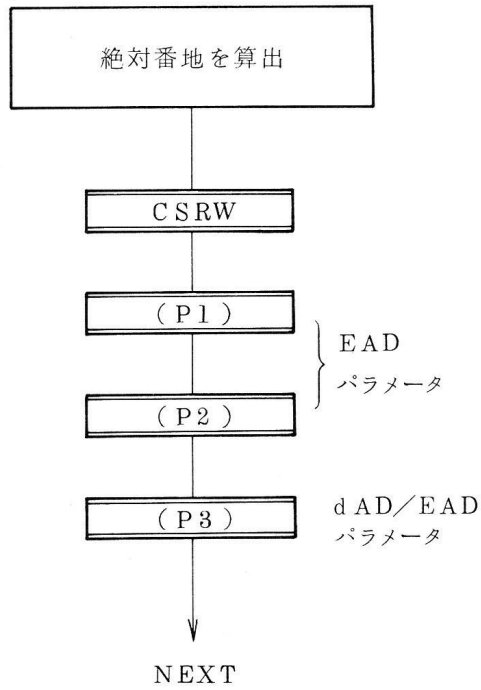
描画方向の決定と絡め、与えられた点のうち、いずれかを描画開始点として定義し、「座標と絶対番地との対応」の項(4.3.3)に述べられるように絶対番地を算出します。



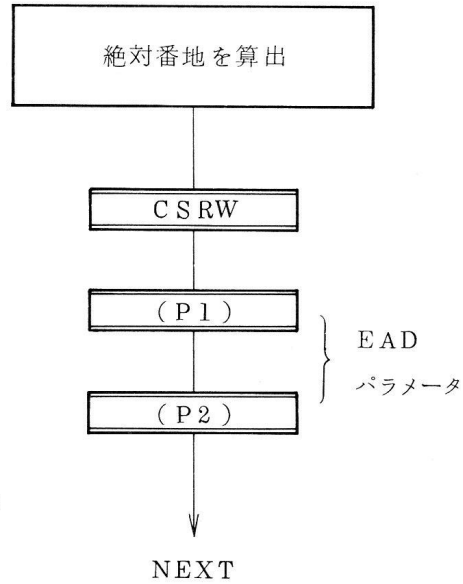
この場合、グラフィック描画時(TEXTを含む)には描画実行アドレスEADに加えてドット・アドレスdADを、文字描画時にはEADをCSRWコマンドによってGDCに設定します。



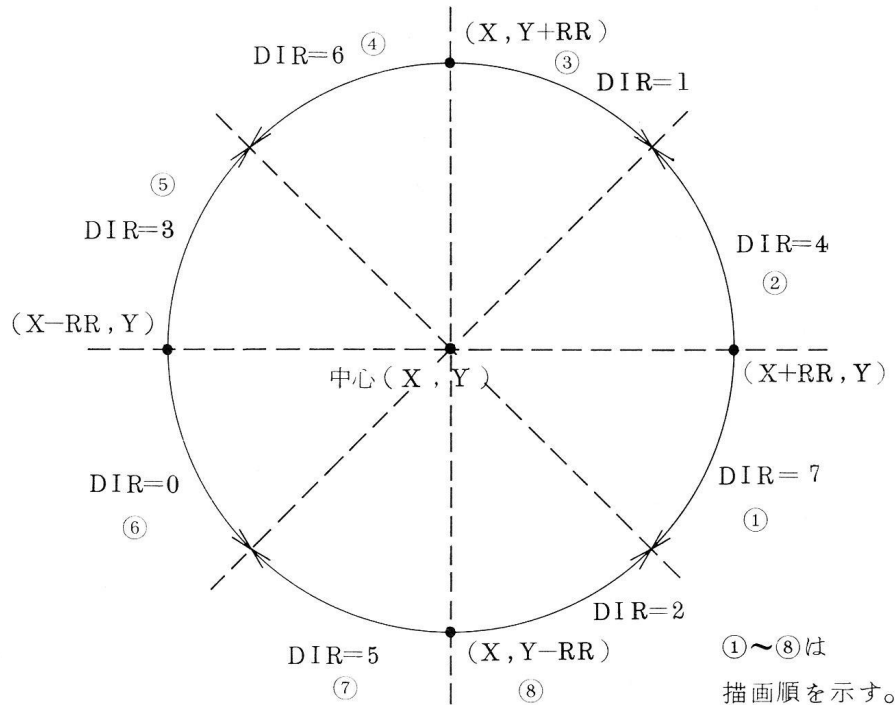
グラフィック制御時  
(ドット・アドレスの設定が必要)



文字制御時



円弧，扇形描画の場合，描画開始点は，中心の座標  $(X, Y)$  から  $X$  方向に  $\pm RR$ ， $Y$  方向に  $\pm RR$  ( $RR$  は半径) だけ離れた点に取ります。



$(X+RR, Y)$  の点から方向 "7" ( $DIR=7$ ) で描画を開始したとき，第2の  $1/8$  弧描画は，同一描画開始点  $(X+RR, Y)$  の  $DIR=4$  に取り，第3の弧描画では描画開始点を  $(X, Y+RR)$  に取り，方向を "1" とします。以下  $(X, Y-RR)$  の  $DIR=2$  の弧描画まで順に描画します。

このような順で描画すると，方向算出は "+5" または "-3" することにより容易に求めることが出来ます。

## 5.1.4 描画方向，描画パラメータの算出

### (1) 直線描画

$(X1, Y1)$ ， $(X2, Y2)$  の 2 点間の直線補間において，描画方向を " 0 " から " 3 " に限定したときの概略フローは，図 5-1 のようになります。

### (2) 四辺形描画

対角線上の 2 点  $(X1, Y1)$ ， $(X2, Y2)$  が与えられたときの四辺形および傾斜四辺形描画の概略フローは各々図 5-2，図 5-3 のようになります。

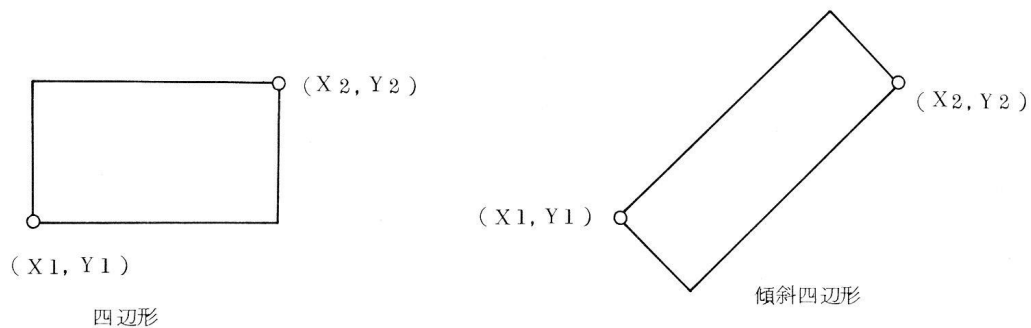
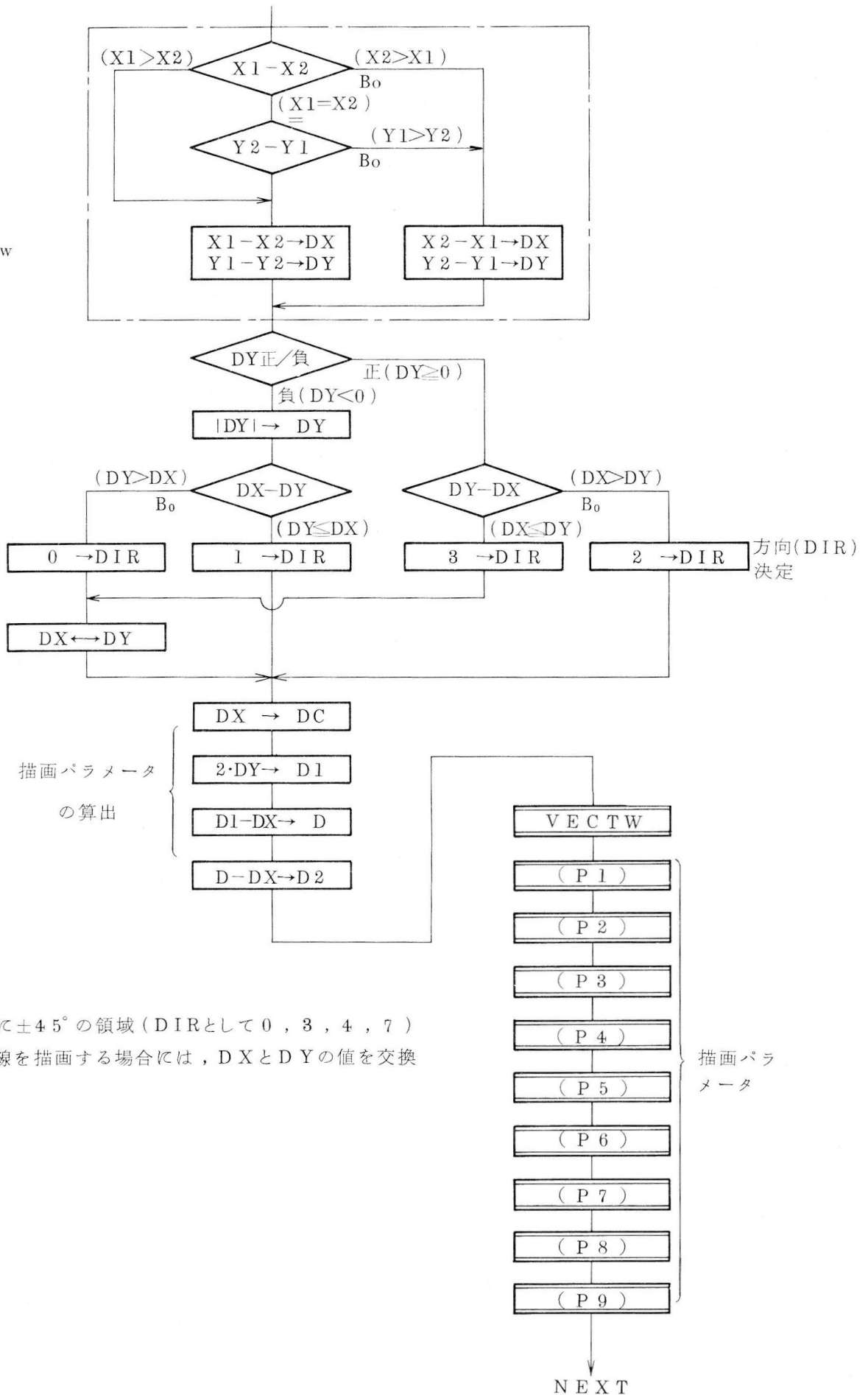


図 5 - 1 直線描画

Bo : Borrow

= : Equal



Y 軸方向に  $\pm 45^\circ$  の領域 (DIRとして 0, 3, 4, 7) に対して直線を描画する場合には, DXとDYの値を交換します.

描画パラメータ

図 5 - 2 四辺形描画

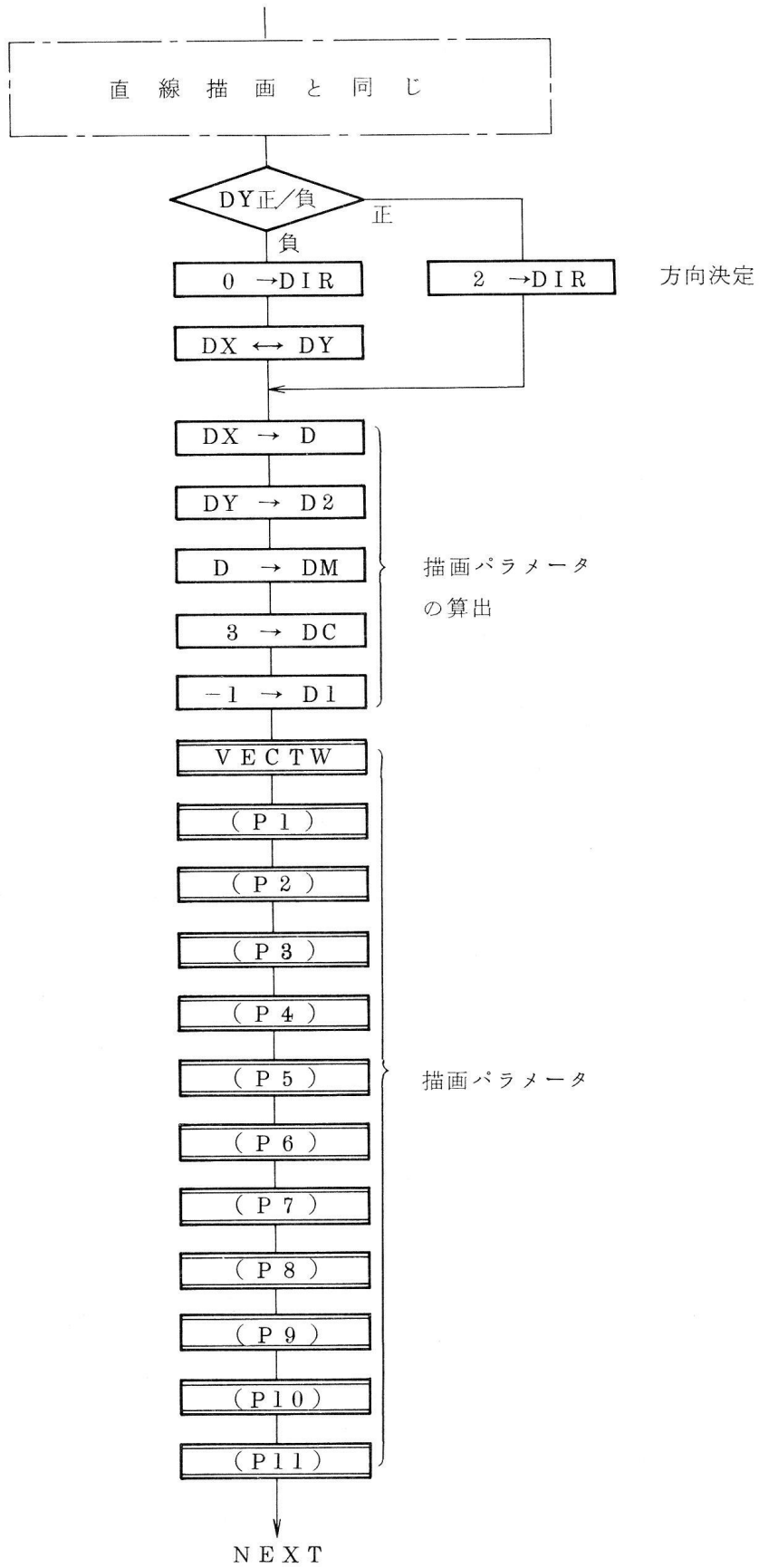
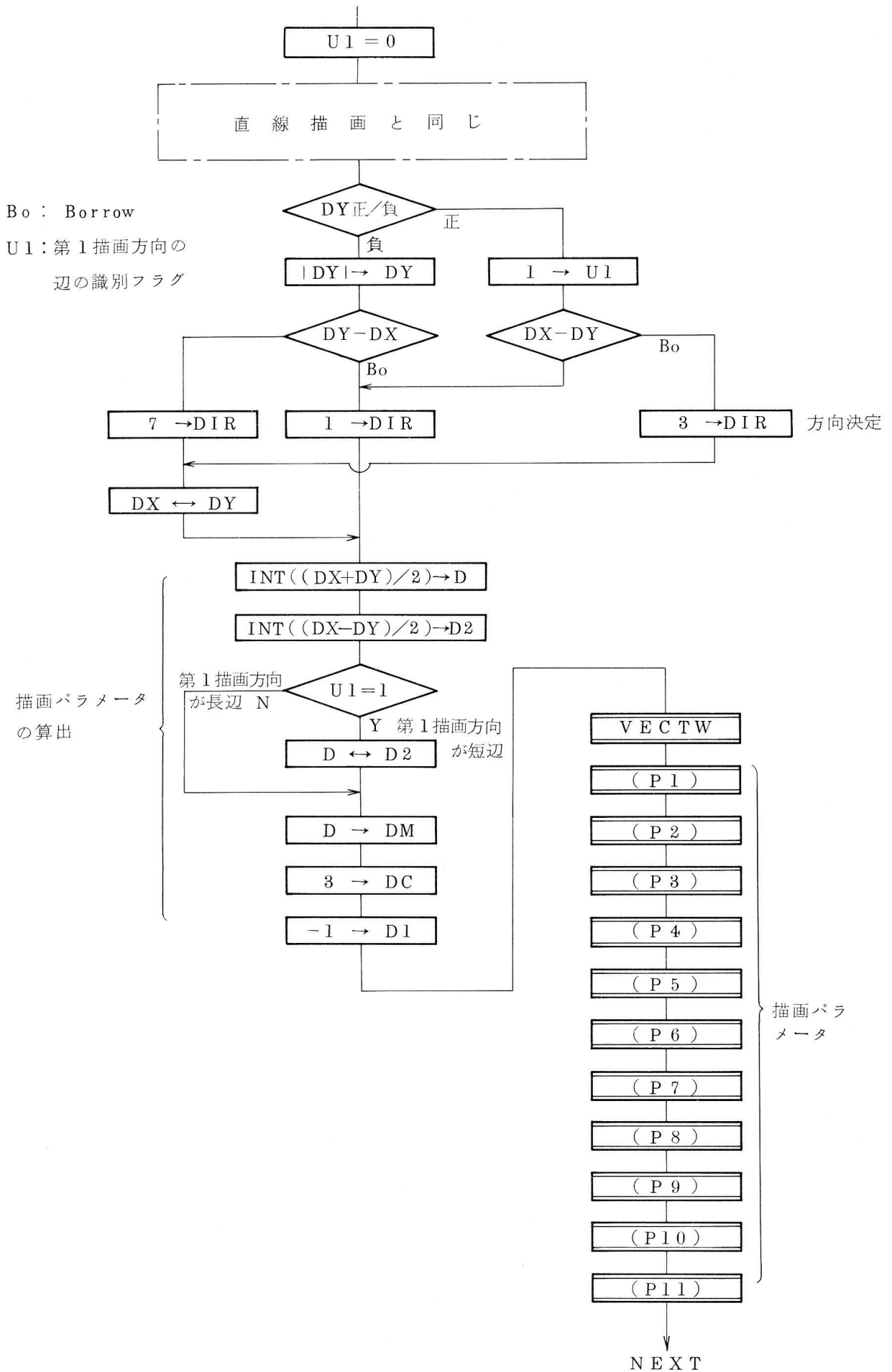


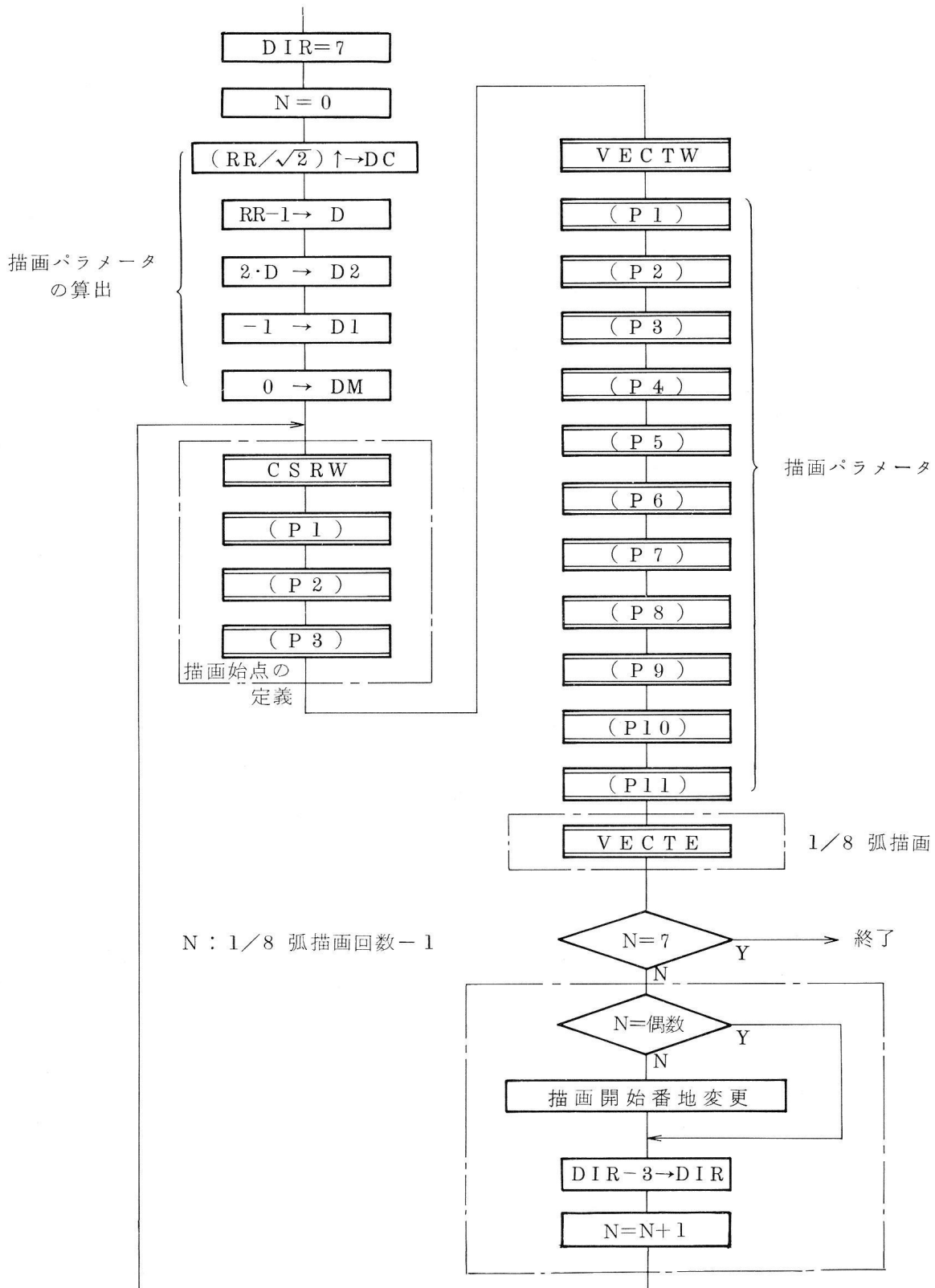
図 5 - 3 傾斜四辺形描画



(3) 円描画

中心の座標 ( X , Y ) , 半径 R R の円描画は , 1 / 8 弧を 8 回描画することにより達成します . 描画パラメータは描画開始時に算出した値を記憶しておき , 1 / 8 弧描画ごとに , その値を G D C に送じます .

図 5 - 4 円描画



(4) 弧，扇形描画

中心座標 ( X , Y ) ，半径 R R ，描画開始角 T 1 ，描画終了角 T 2 で弧を定義します。弧の場合には，両端の点から中心に直線を描画し扇形とします。円の場合には，中心との直線補間を行いません。

概略フロー中，I 1 ， I 2 は弧描画の開始および終了が，どの 1 / 8 弧領域に含まれるかを示します。次図は，I 1 ， I 2 と D I R ( 方向 ) との関連を示します。

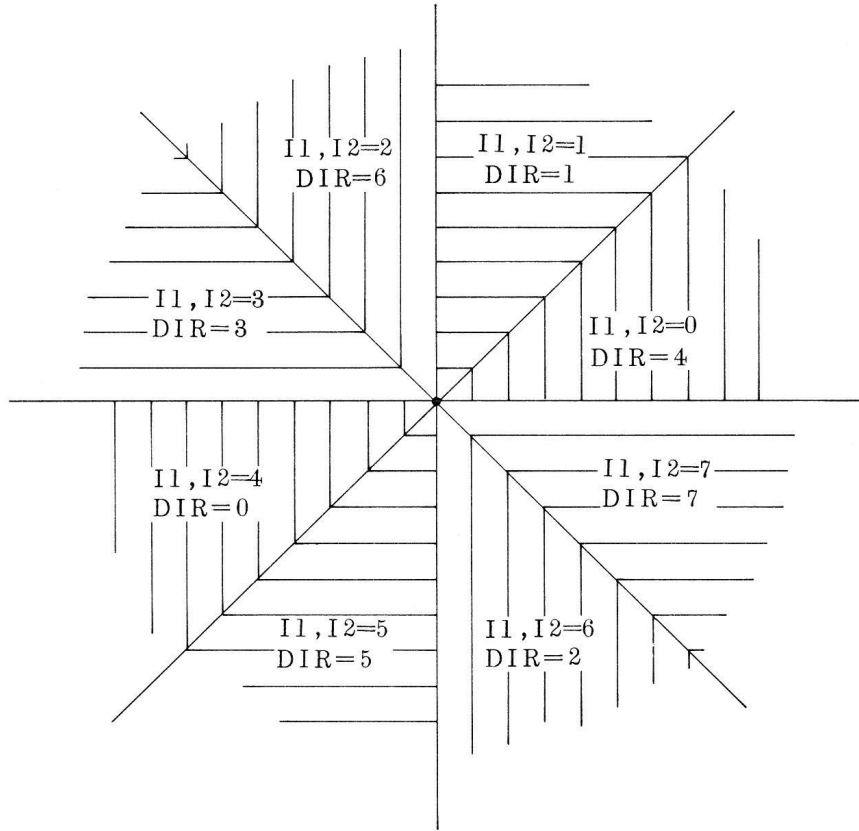


図 5 - 5 弧，扇形描画(1)

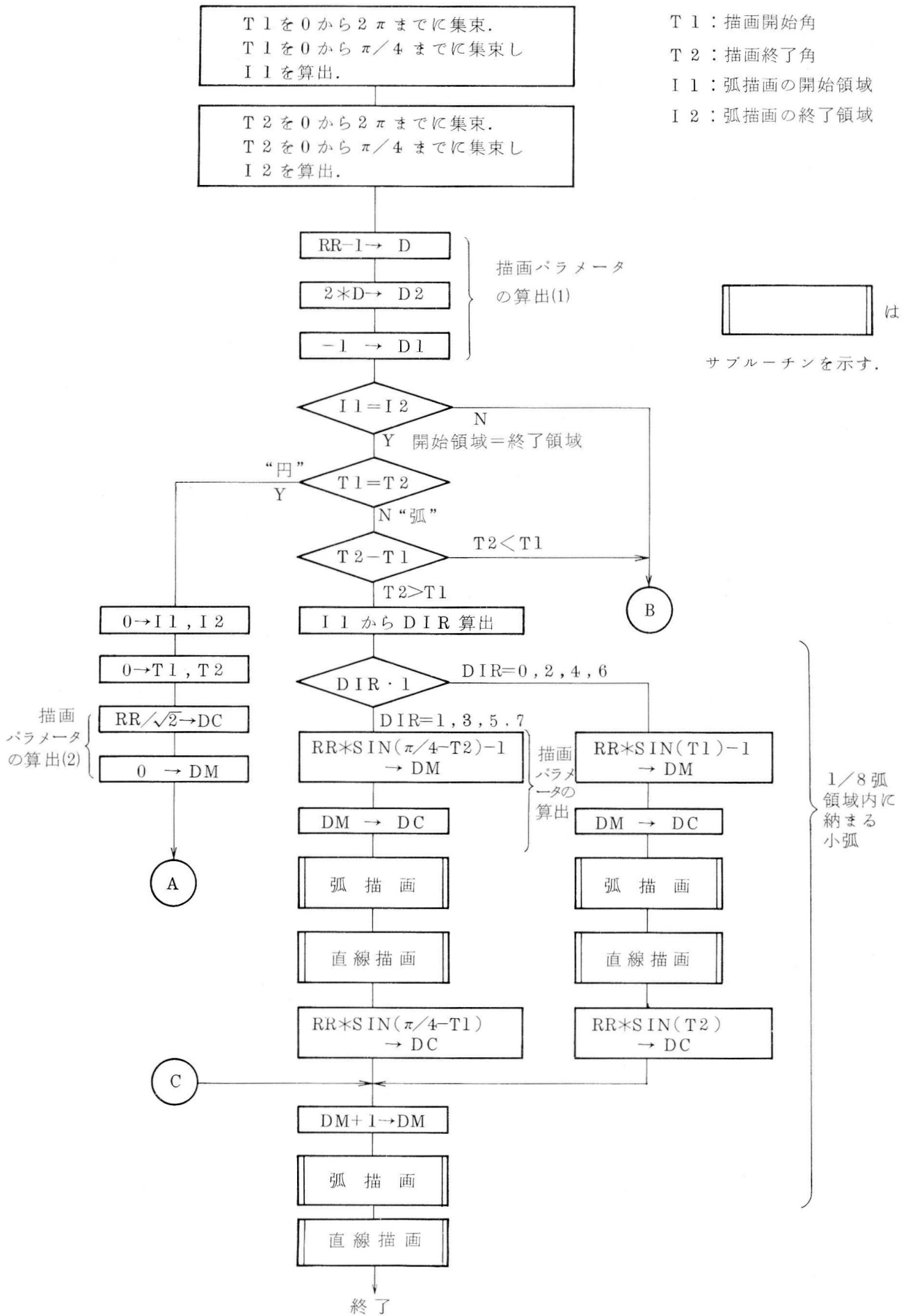




図 5 - 6 弧, 扇形描画(2)

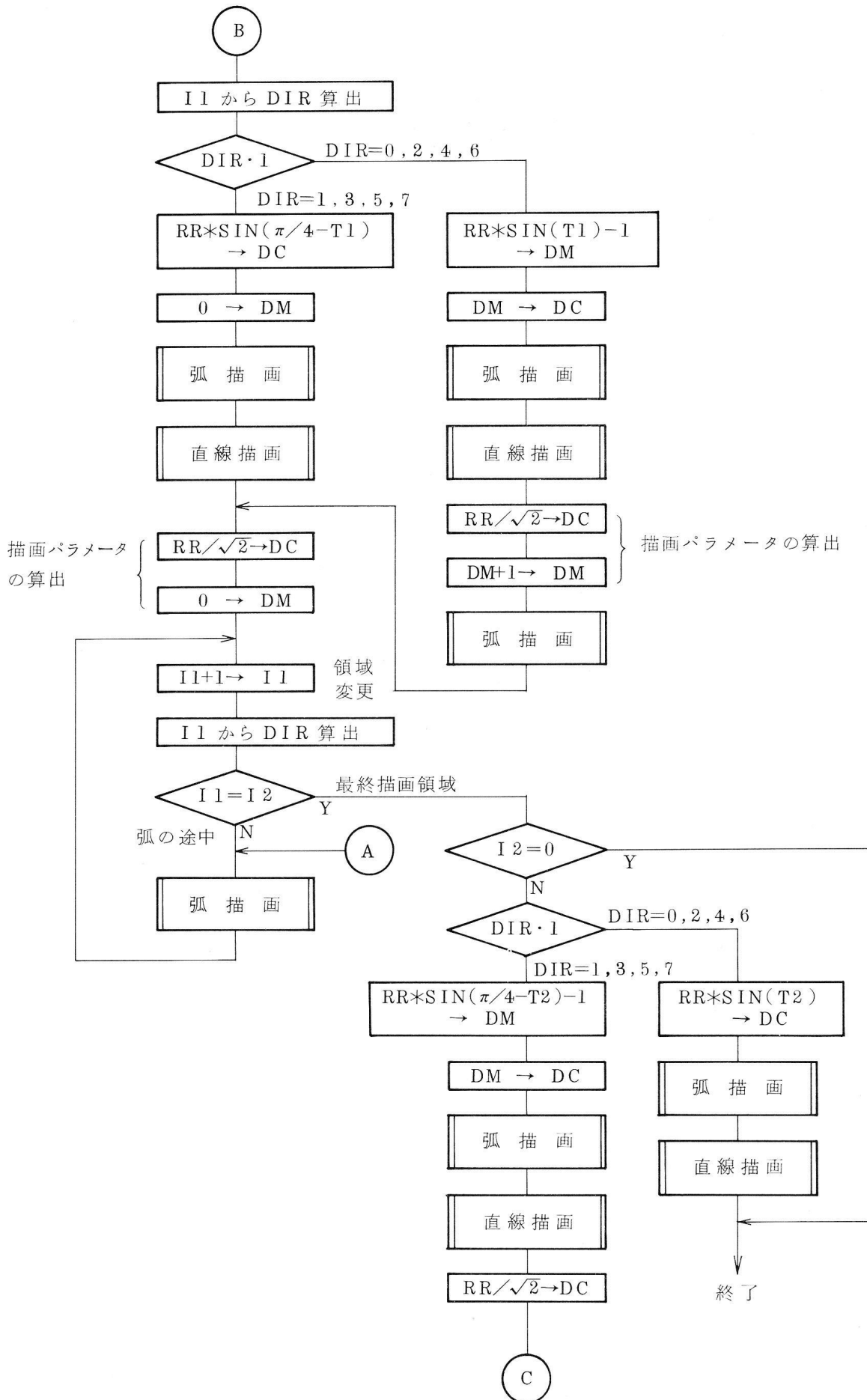
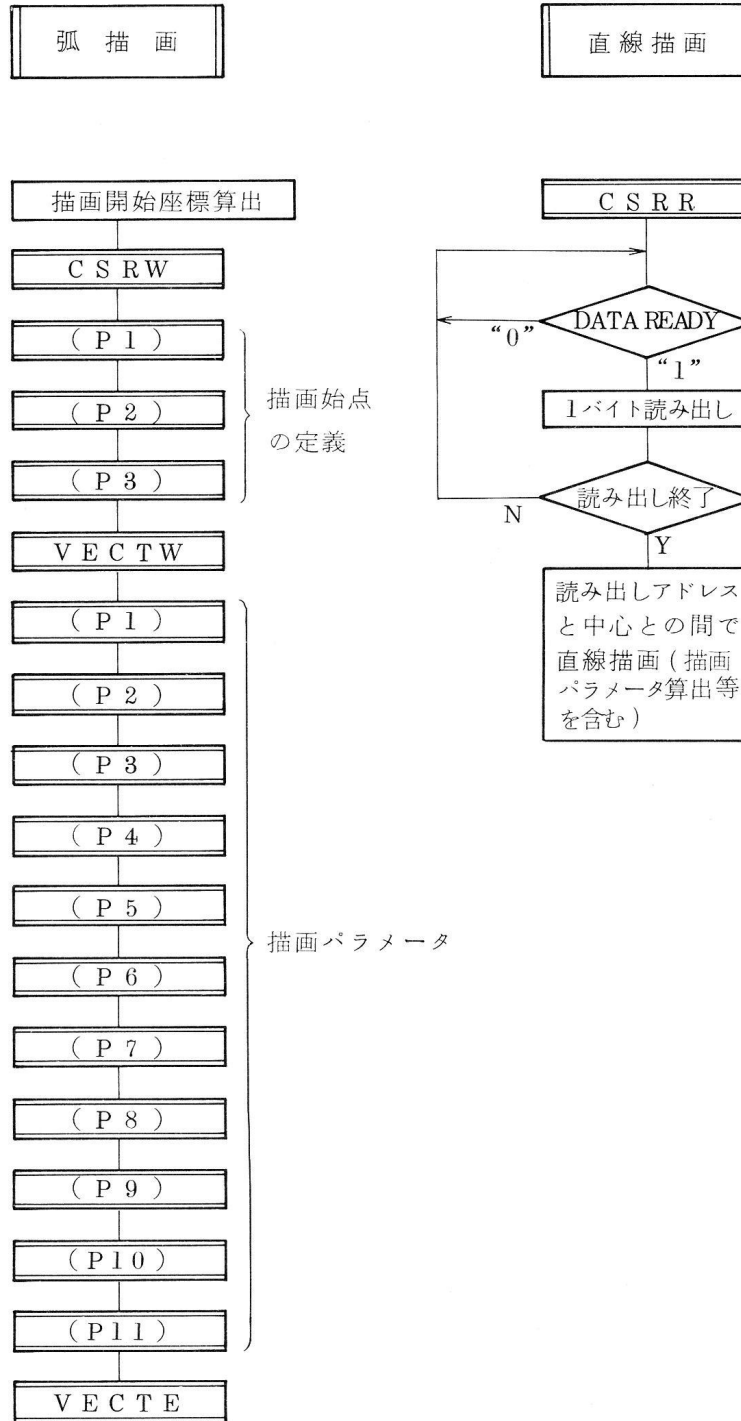


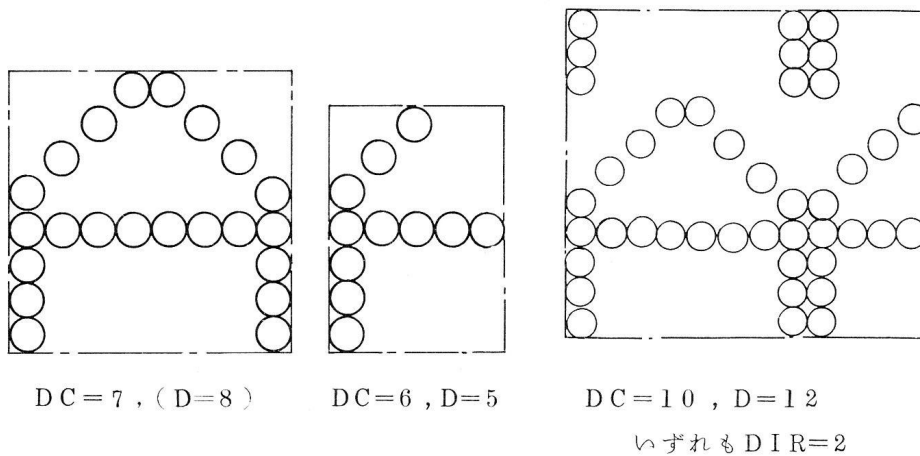
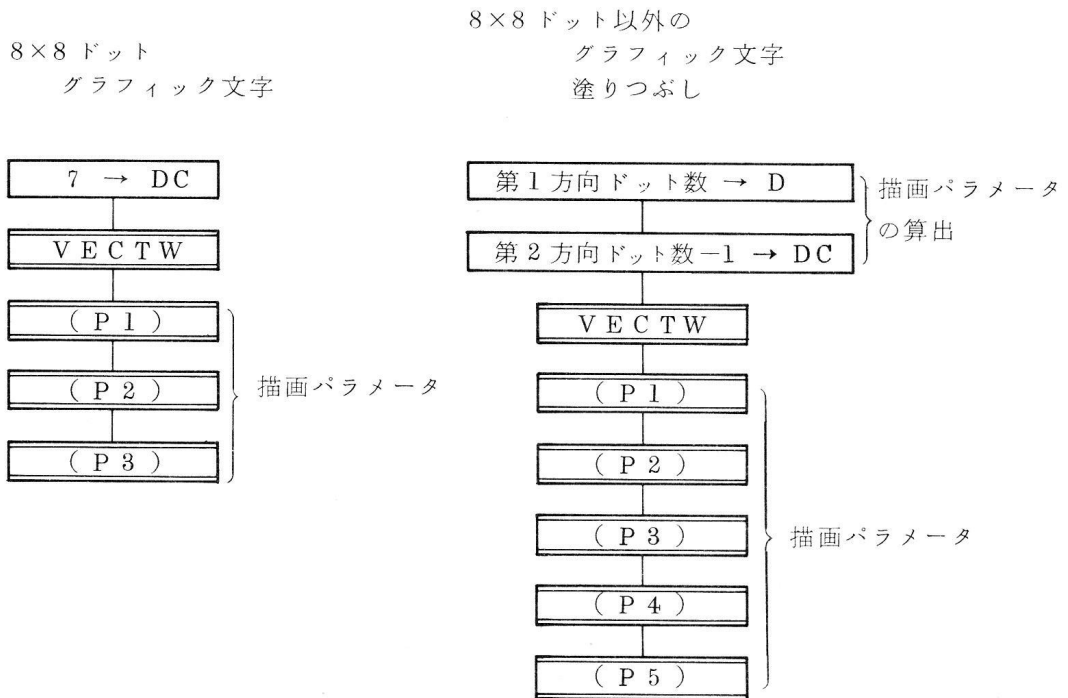
図 5 - 7 弧，扇形描画(3)



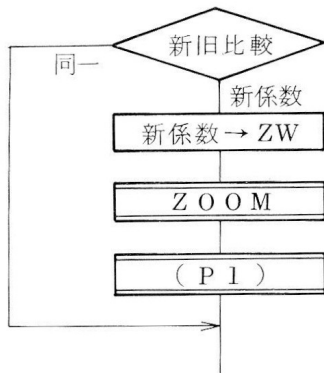


(5) グラフィック文字描画

8×8ドットのグラフィック文字描画時には“D”の設定は不要です。8×8ドットより小さいドット構成のグラフィック文字描画の場合(5×7ドットなど), VECTW コマンドにより“DC”に加えて“D”の設定が必要となりますが, TEXTW コマンドによって設定するドット情報は, 必要バイト数だけを(例えば7バイトのみ)GDCに送出するだけで構いません。8×8ドットよりも大きな領域を同一の8×8ドット構成情報によって, 任意の形状(ドット, 斜線など)で塗りつぶす場合など, 描画領域が8×8ドットでないときには“DC”および“D”を設定します。



グラフィック文字描画時には, 拡大描画が可能です。拡大係数は, 変更があったときのみGDCに設定します。



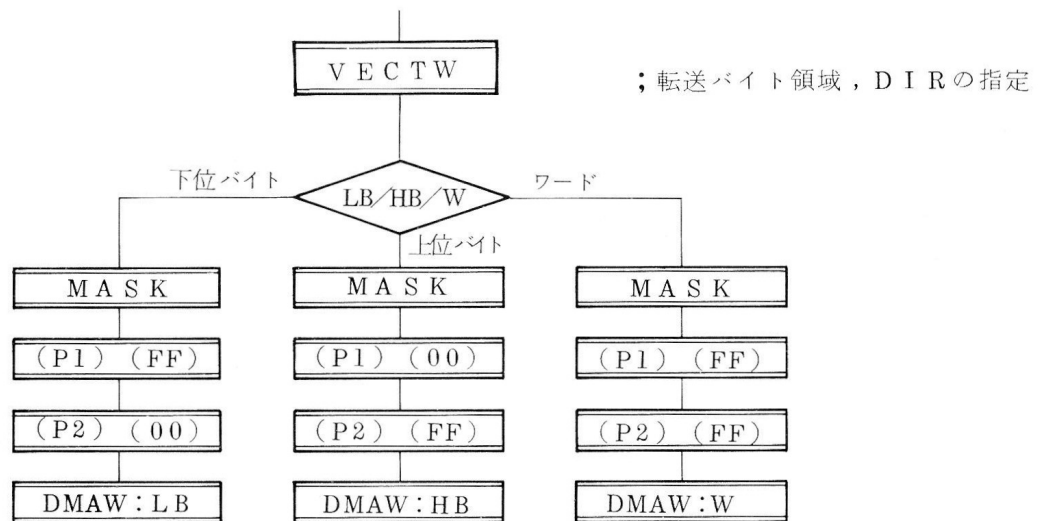
### 5.1.5 描画実行指示

直線，四辺形，円弧は VECTE コマンド，グラフィック文字は TEXTE コマンドで実行指示します。

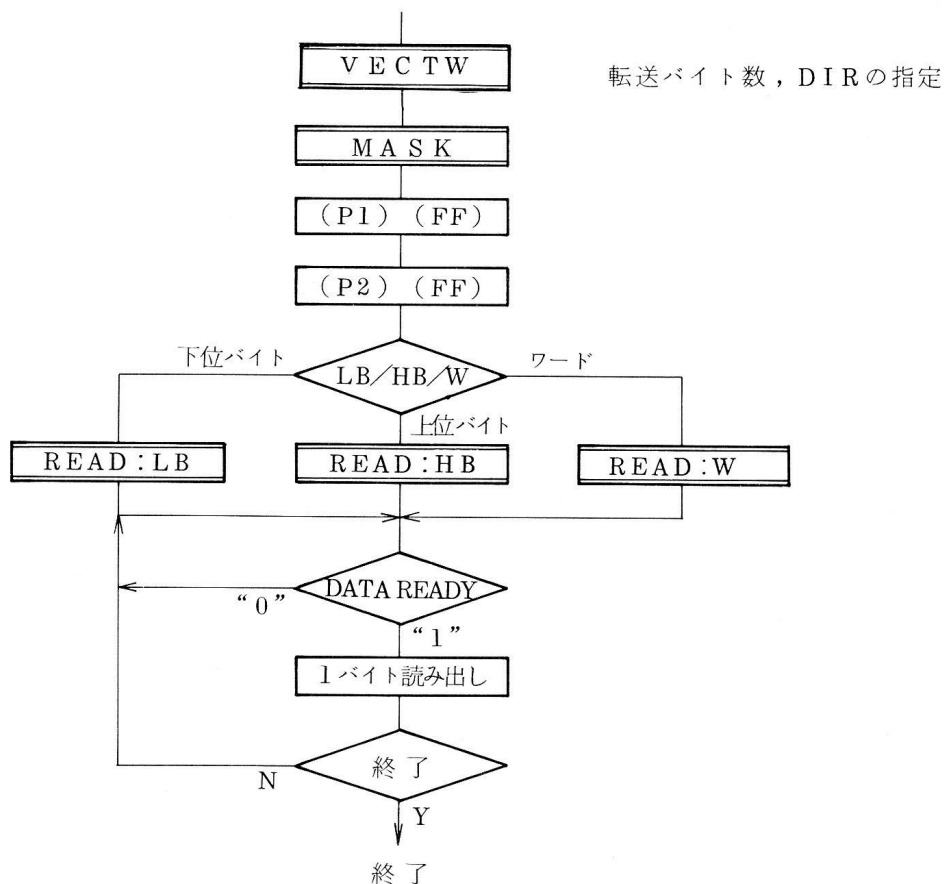
文字モードにおいて文字コードを描画（映像メモリに書き込む）するときは WRITE コマンドに続くパラメータ（文字コード）入力によって描画実行を指示します。

映像メモリ内容を主記憶へ DMA 転送するときは DMA R コマンドによって転送の開始を指示します。

主記憶内容を映像メモリへ DMA 転送するときは MASK コマンドによって，必要に応じてマスクをかけた後，DMA W コマンドによって転送の開始を指示します。

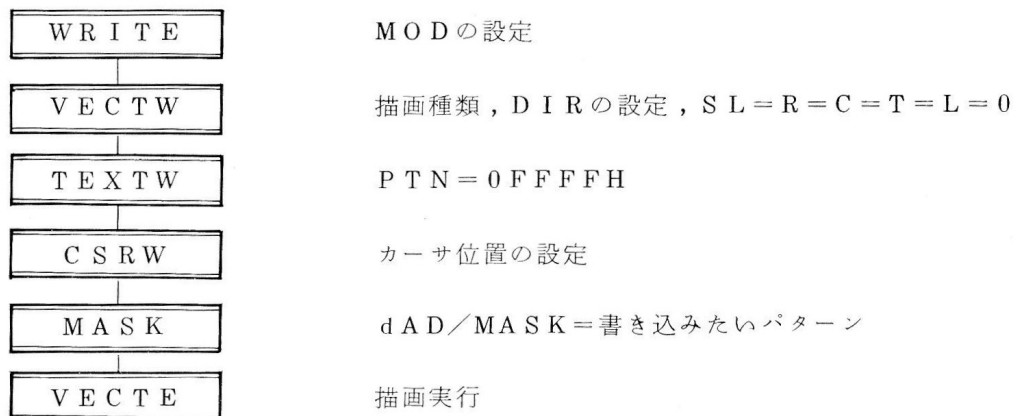


映像メモリ内容を読み出す場合、グラフィック・モード時には、ワード単位での番地変更を可とするため、MASKコマンドによって、マスク・レジスタを全て“1”とする必要があります。



### 5.1.6 グラフィック・モードでの1ワード描画

グラフィック・モードでは1ドット描画ということを中心にしているため、WRITEコマンドにより任意のパターンを映像メモリに書き込むことはできません(グラフィック・モードでは描画時にパターン・レジスタの1ビットのみ参照されます)。ワード単位での任意のパターンの書き込みは以下の方法をとることにより、可能となります。

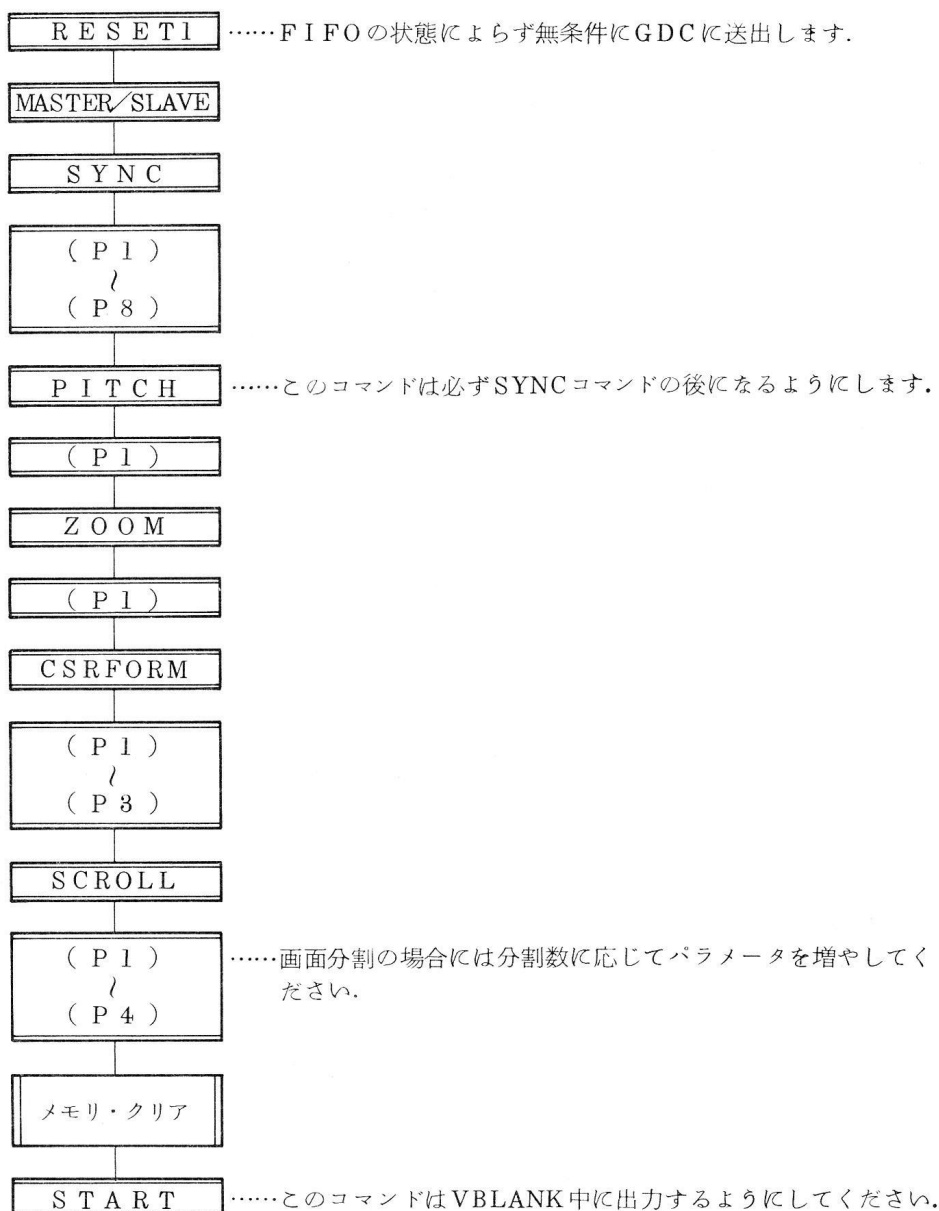


通常の描画と異なるのはTEXTWコマンドで0FFFFHを設定し、MASKコマンドにより書き込みたいパターンを設定する点です。また、VECTWコマンドで設定するSL,R,C,T,Lは全て0に設定します。

なお、描画が垂直方向(DIR=0,4)の場合にはGDCが自動的に次の描画アドレスを演算しますから、MASKコマンドとVECTEコマンドの発行の繰り返しにより連続的に描画実行できますが、それ以外の方向へ描画する場合にはアドレス演算がうまくいきませんので、カーサの再設定が必要となります。

## 5.2 初期化フローチャート

同期信号発生、表示に必要なパラメータは全て設定します。電源投入時には、出力端子の状態はRESET1,RESET2またはRESET3コマンドが発行されるまで不定です。RESET1,RESET2またはRESET3コマンドは、FIFOの状態を検出することなく無条件にGDCに送出します。



## 5.3 映像メモリ・クリア・フローチャート

WRITE コマンドを使用した文字制御時の 1 次元的クリアと、グラフィック文字描画によるグラフィック制御時の 2 次元的クリアの方法があります。1 次元的クリアでは、“DC”にクリア・ワード数を設定します。1 回のコマンド群の送出で 16KW までのメモリをクリア出来ます。16 KW 以上のメモリをクリアする場合には、CSRW コマンドによりクリアの開始番地を変えてクリア・フローを繰り返します。

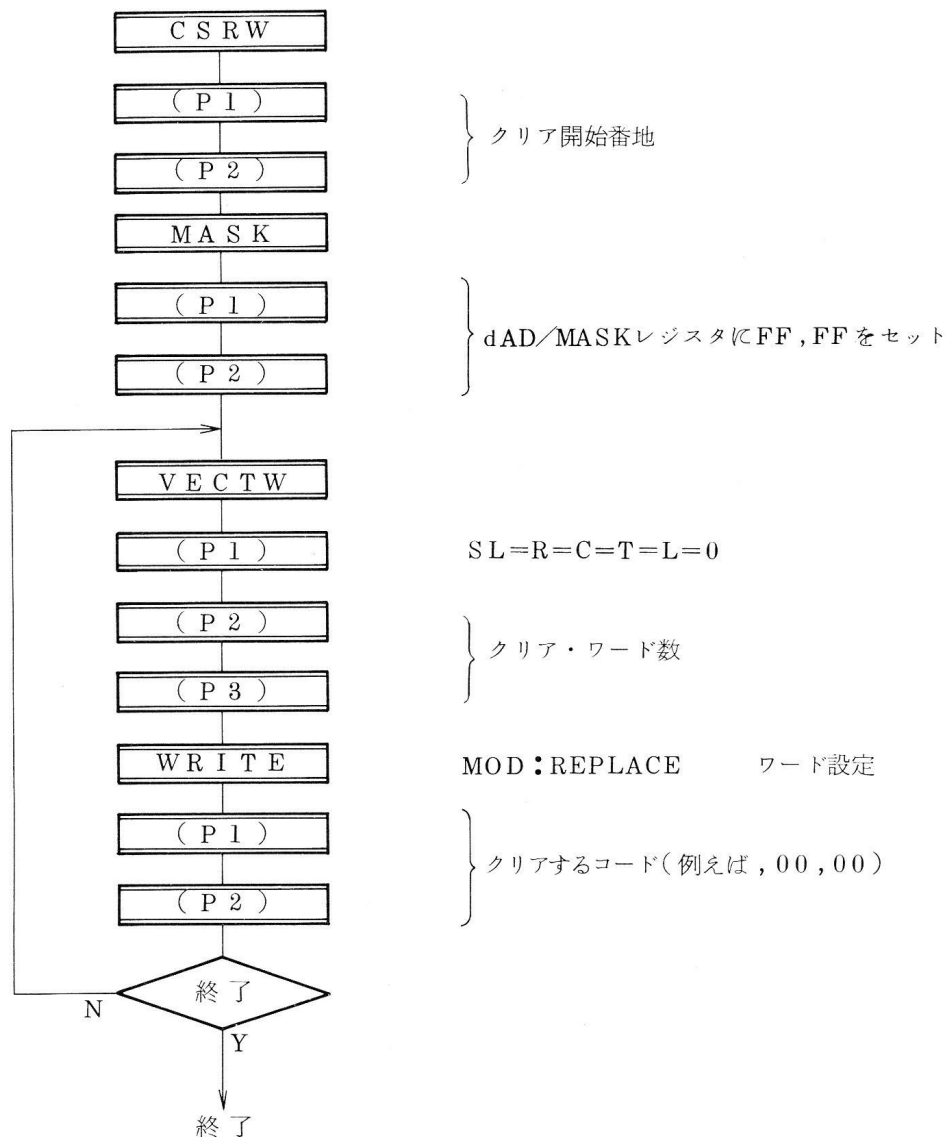
2 次元的クリアでは、第 1 と第 2 の描画方向に対するワード数を“D”および“DC”に設定します。従って、(16K×16K)ワードまでのメモリ・クリアを 1 回のコマンド群の送出によって可能とします。

いずれの場合も、MASK コマンドによってマスク・レジスタ内容を全て“1”とし、16 ビット同時描画（ワード単位描画）を実行させておりましたが、この MASK コマンドの使用方法はグラフィック・モードでは特殊なものであり、一般の描画に使用することはできません。



(1) 1 次元的クリア

クリア開始番地を“EAD”に，クリア・ワード数を“DC”に設定します．

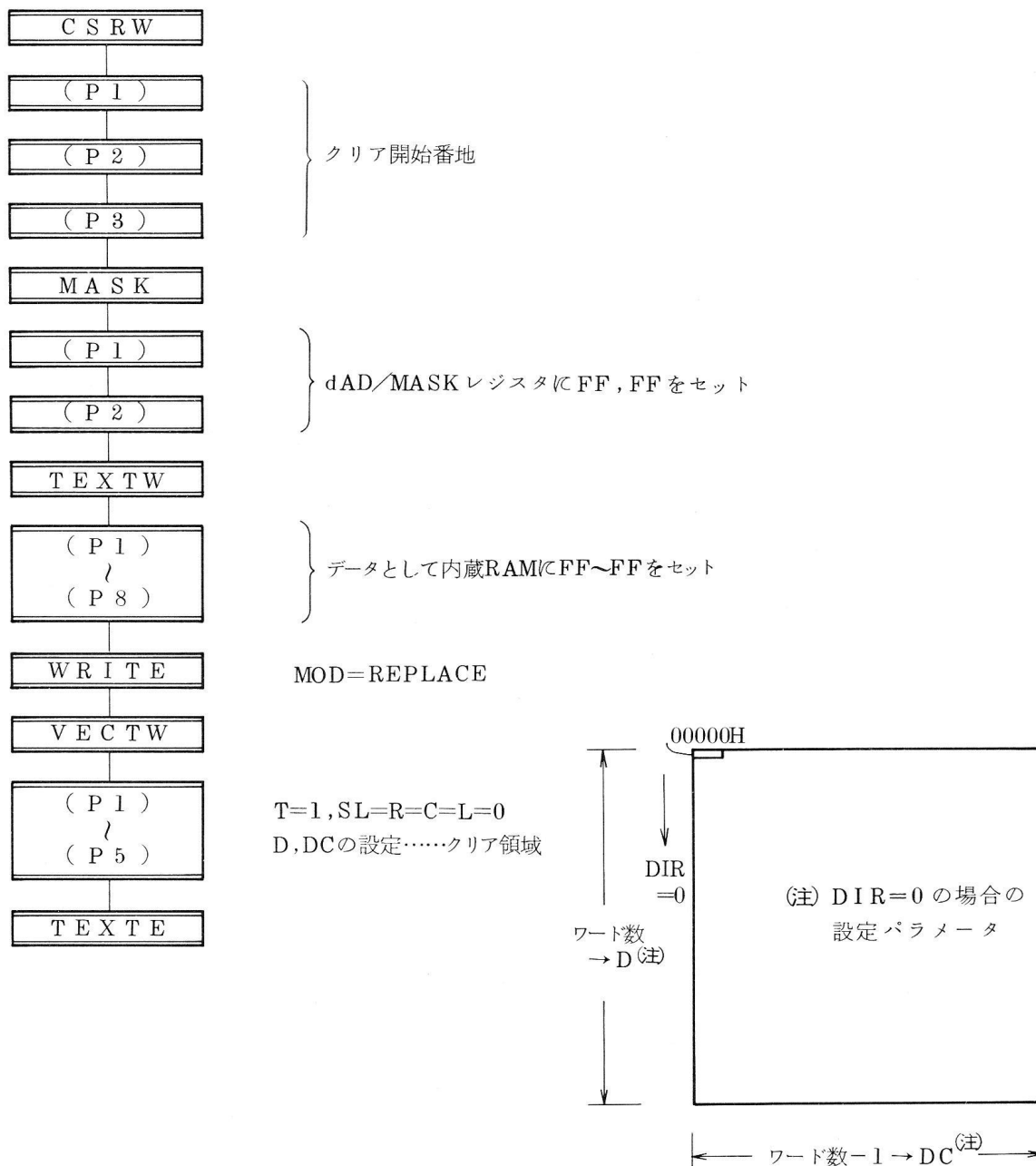


文字／グラフィック混在モードでVECTWコマンドのDGD="0"とした場合および文字モードの場合は，WRITEコマンドに付随したパラメータを任意値にすることで任意のデータにクリアすることが可能です．

注 グラフィック制御時(文字／グラフィック混在モードでDGD="1"とした場合およびグラフィック・モードの場合)は，映像メモリに書き込まれるデータは"0000H"もしくは"0FFFFH"となり，WRITEコマンドに付随したパラメータをそのまま映像メモリに書き込むことはできません．

(2) 2 次元的クリア

クリア開始番地を“EAD”に，第1の描画方向のクリア・ワード数を“D”に，第2の方向の(ワード数-1)を“DC”に設定します。



このフローチャートに従うと，映像メモリに書き込まれるデータは全て"0FFFFH"です。また，テキスト・パターンを全て"00"とすれば映像メモリに書き込まれるデータは"0000"となります。

テキスト・パターンの内容をそのまま映像メモリに書き込むような場合には，MASKコマンドは使用できません。

## 5.4 閉区間の塗りつぶし

### 5.4.1 矩形領域の塗りつぶし

矩形領域の塗りつぶしはGDCが持っているコマンドを利用し、テキスト描画により実現します。以下に例を示します。この例ではPITCHを40字分とっており、頂点が0番地の2ドット目、2番地の12ドット目、118H番地の2ドット目、11AH番地の13ドット目にあるような矩形を塗りつぶしをしています。

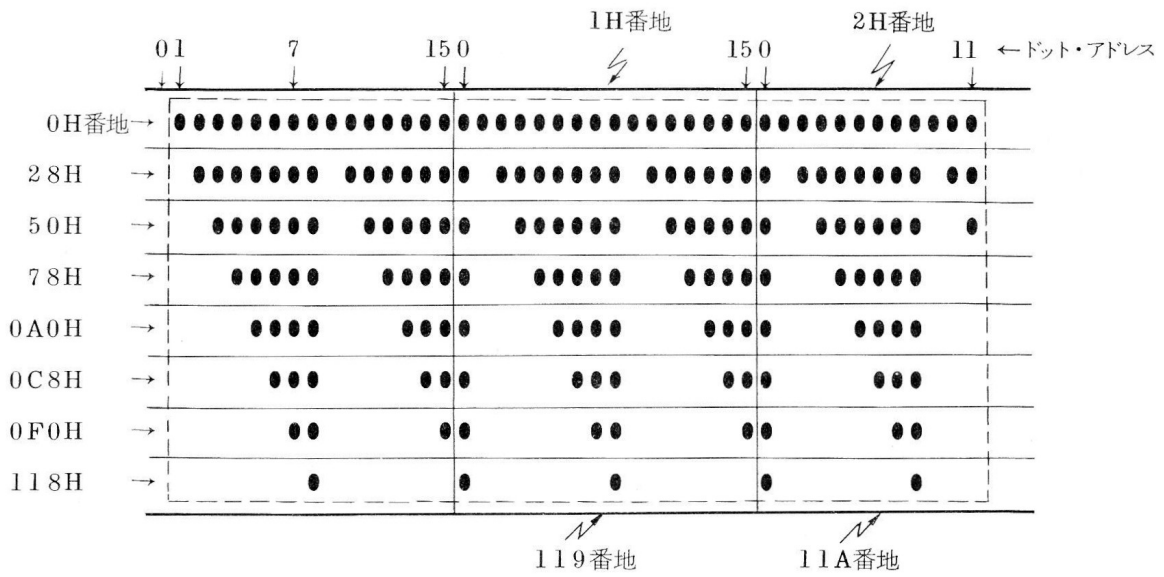
```
CSRW      : 0100 1001B(49H)
          P1 : 0001 1000B(18H) ; EAD=118H
          P2 : 0000 0001B(01H)
          P3 : 0001 0000B(10H) ; dAD=01

WRITE     : 0010 0000B(20H) ; MOD=REPLACE

TEXTW    : 0111 1000B(78H)
          P1 : 1111 1111B(0FFH) ; TEXT PATTERNS
          P2 : 1111 1110B(0FEH)
          P3 : 1111 1100B(0FCH)
          P4 : 1111 1000B(0F8H)
          P5 : 1111 0000B(0F0H)
          P6 : 1110 0000B(0E0H)
          P7 : 1100 0000B(0C0H)
          P8 : 1000 0000B(80H)

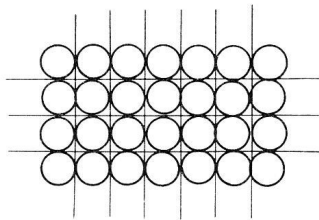
VECTW    : 0100 1100B(4CH)
          P1 : 0001 0000B(10H) ; TEXT, DIR=2
          P2 : 0000 0111B(07H) ; DC=07
          P3 : 0000 0000B(00H)
          P4 : 0010 1011B(2BH) ; D=2BH
          P5 : 0000 0000B(00H)

TEXTE    : 0110 1000B(68H)
```

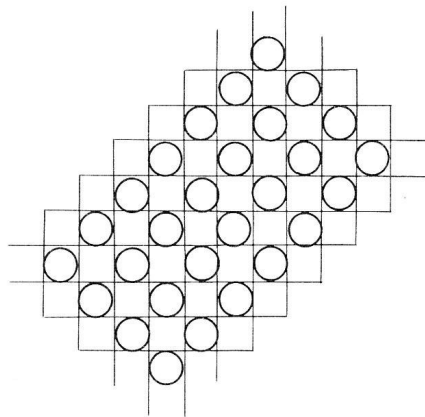


注意 矩形領域のベタ塗りつぶしにおいてD I Rの値が偶数の場合には下図(a)のようにベタ塗りになりますが、D I R値が奇数の場合には図(b)のように塗りつぶされないドットが発生します。

(a) D I Rが偶数の場合

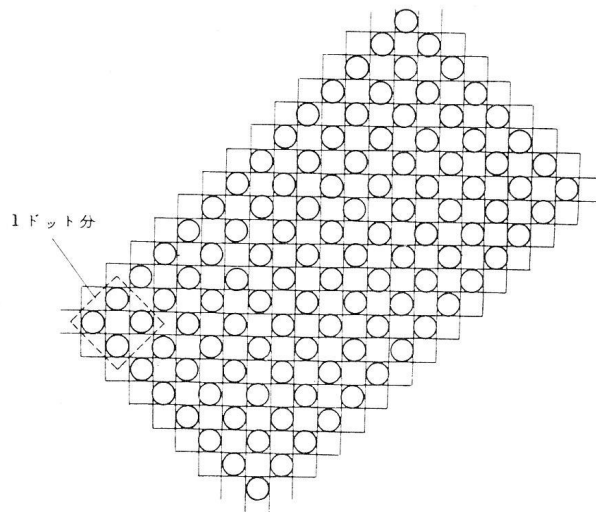


(b) D I Rが奇数の場合



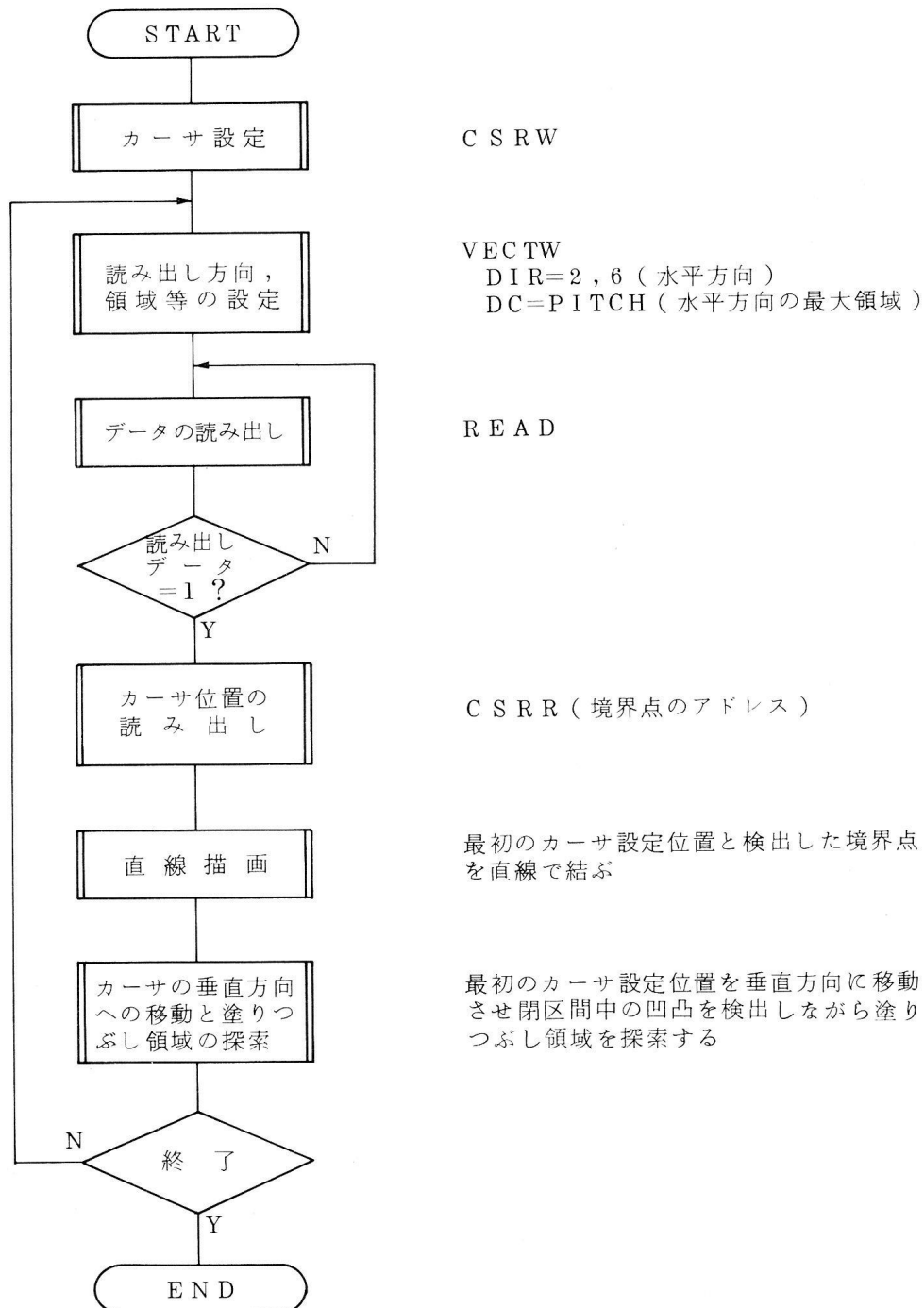
またこの場合、拡大描画係数を設定しますと(b)と同様、(c)のようになります。

(c) D I Rが奇数で拡大描画をした場合 ( ZW=1 )



## 5.4.2 任意閉区間の塗りつぶし

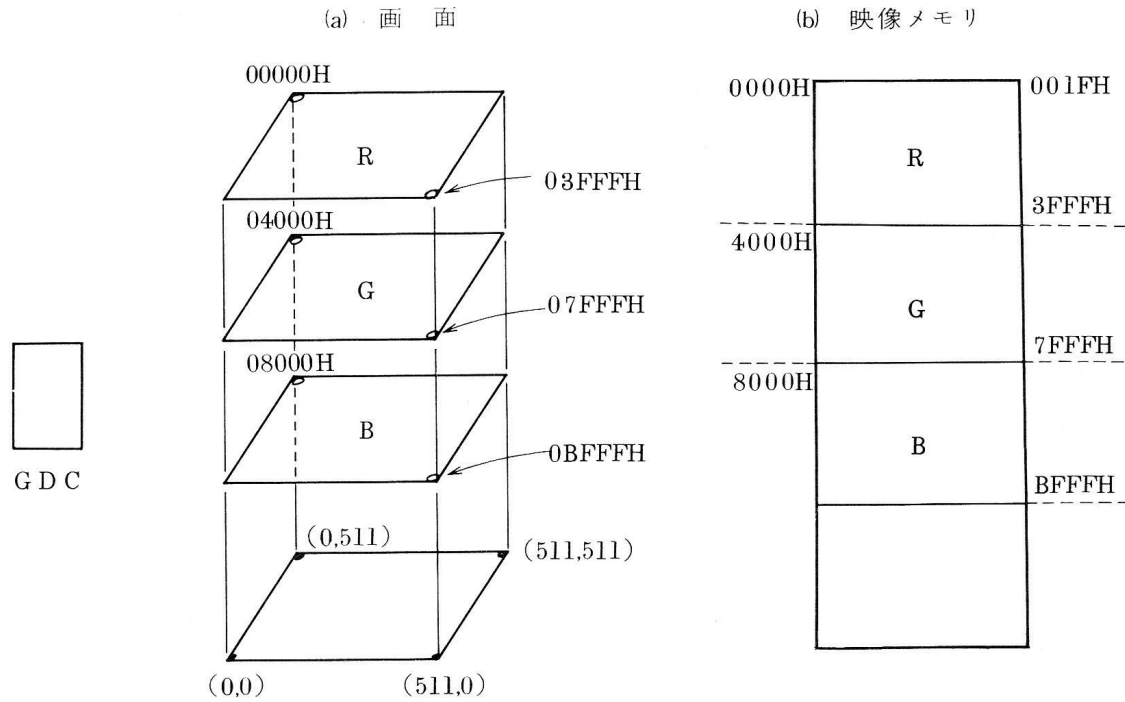
任意閉区間の塗りつぶしは、基本的には閉区間の中にカーサを設置し水平方向にカーサを移動しながらデータを読み進みデータの“0”→“1”の変化点を見つけることにより境界点を検出する境界線検出部分、最初にカーサを設置した位置から検出した境界点までを直線描画するという直線描画部分と、カーサを垂直方向に移動させて塗りつぶし領域を探索する部分の3つから構成されます。これらの3つの動作の繰り返しにより、任意閉区間の塗りつぶしを可能にします。



## 5.5 描画の実例

次のようなメモリ構成を持つ装置において、各種描画を行うとき、GDCに送出すべきコマンド/パラメータ群の実例を以下に示します。

512×512ドット，7色カラー



GDC 1個で、各メモリ・プレーンの表示/描画を行います。

## 5.5.1 映像メモリ・クリア

### (1) 一次元的クリア

コマンド	コマンド コード	パラメータ
(ZOOM)	-46-	00
(CSRW)	-49-	00 00
(MASK)	-4A-	FF FF
(VECTW)	-4C-	02 FF 3F
(WRITE:W)	-20-	20 07
(VECTW)	-4C-	02 FF 3F
(WRITE:W)	-20-	20 07
(VECTW)	-4C-	02 FF 3F
(WRITE:W)	-20-	20 07
(VECTW)	-4C-	02 FF 3F
(WRITE:W)	-20-	20 07
(START)	-6B-	

- MOD=REPLACE, WLH=ワード転送を指定

- 64KW(16KW×4回)を

下位バイト="20<sub>H</sub>"

上位バイト="07<sub>H</sub>"

にクリアする。

- "20<sub>H</sub>"はASCIIコードにおいて  
ブランク・コードを示す。

### (2) 二次元的クリア

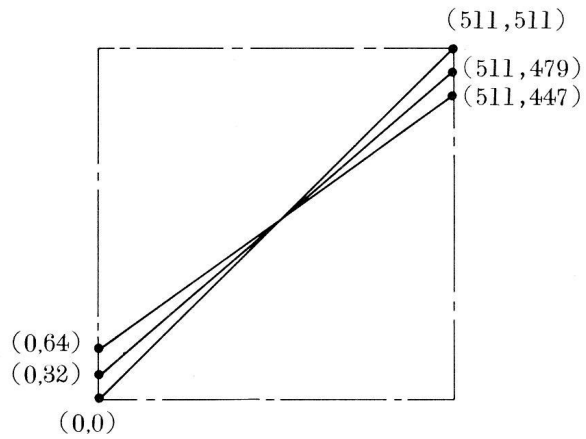
(ZOOM)	-46-	00
(CSRW)	-49-	00 00 00
(MASK)	-4A-	FF FF
(TEXTW)	-78-	FF FF FF FF FF FF FF FF
(VECTW)	-4C-	10 1F 00 00 08
(WRITE:W)	-20-	
(TEXTE)	-68-	
(START)	-6B-	

等倍指定

MOD=REPLACEに設定

## 5.5.2 直線描画(グラフィック制御時)

### (1) 実線描画



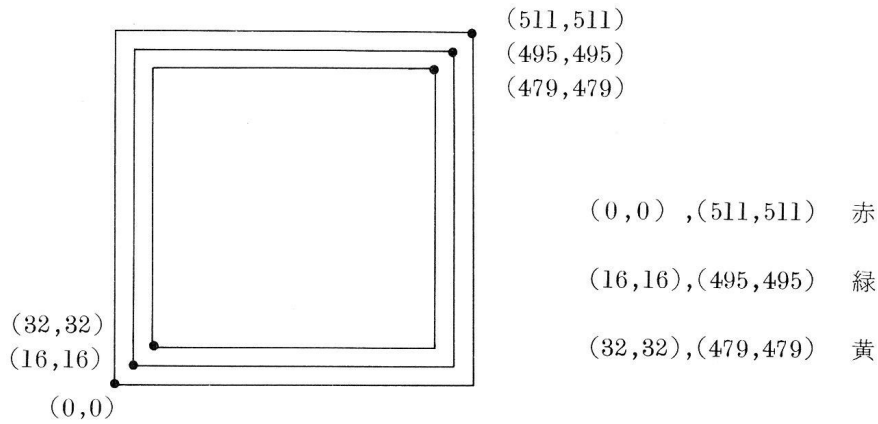
(TEXTW)	-78- FF FF	} 実線指定
(WRITE:W)	-21-	
(CSRW)	-49- E0 3F 00	} MOD: COMPLEMENT
(VECTW)	-4C- 0B FF 01 FF 01 00 00 FE 03	
(VECTE)	-6C-	
(CSRW)	-49- E0 7B 00	} 赤 (0,0)→(511,511)
(VECTW)	-4C- 0A FF 01 7F 01 80 FF 7E 03	
(VECTE)	-6C-	
(CSRW)	-49- E0 37 00	} 緑 (0,32)→(511,479)
(VECTW)	-4C- 0A FF 01 FF 00 00 FF FE 02	
(VECTE)	-6C-	
(CSRW)	-49- E0 77 00	} 黄 (0,64)→(511,447)
(VECTW)	-4C- 0A FF 01 FF 00 00 FF FE 02	
(VECTE)	-6C-	

赤色プレーンと緑色プレーンに各々, 同一の描画を実行する.





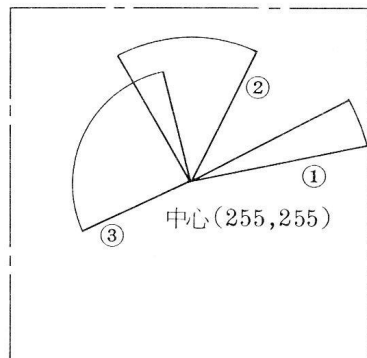
### 5.5.3 四辺形描画 (グラフィック制御時)



(TEXTW)	-78- FF FF	実線指定
(WRITE:W)	-21-	MOD:COMPLEMENT
(CSRW)	-49- E0 3F 00	赤 (0,0), (511,511)
(VECTW)	-4C- 42 03 00 FF 01 FF 01 FF FF FF 01	
(VECTE)	-6C-	
(CSRW)	-49- E1 7D 00	緑 (16,16), (495,495)
(VECTW)	-4C- 42 03 00 DF 01 DF 01 FF FF DF 01	
(VECTE)	-6C-	
(CSRW)	-49- E2 3B 00	黄 (32,32), (479,479)
(VECTW)	-4C- 42 03 00 BF 01 BF 01 FF FF BF 01	
(VECTE)	-6C-	



### 5.5.5 扇形描画(グラフィック制御時)



①: 半径 255  
 開始角 0.2 (ラジアン)  
 終了角 0.5 (ラジアン)  
 赤

②: 半径 223  
 開始角 1.0 (ラジアン)  
 終了角 2.1 (ラジアン)  
 緑

③: 半径 191  
 開始角 1.8 (ラジアン)  
 終了角 3.7 (ラジアン)  
 黄

(TEXTW)	-78- FF FF	実線指定	
(WRITE:W)	-20-	MOD:REPLACE	
(CSRW)	-49- 1F 20 E0	}	注 弧(マスク)
(VECTW)	-4C- 24 31 00 FE 00 FC 01 FF FF 32 00		
(VECTE)	-6C-		
(CSRR)	-E0- (BF 19 00 00 02)*	}	①
(CSRW)	-49- 0F 20 F0		
(VECTW)	-4C- 0A FA 00 6C FF 72 FE 66 00		
(VECTE)	-6C-		
(CSRW)	-49- 1F 20 E0	}	弧
(VECTW)	-4C- 24 7A 00 FE 00 FC 01 FF FF 32 00		
(VECTE)	-6C-		
(CSRR)	-E0- (BD 10 00 00 40)*	}	直線
(CSRW)	-49- 0F 20 F0		
(VECTW)	-4C- 0A DF 00 17 00 38 FF F6 00		
(VECTE)	-6C-		
(CSRW)	-49- 2F 44 F0	}	弧
(VECTW)	-4C- 21 D0 FF DE 00 BC 01 FF FF 00 00		
(VECTE)	-6C-		
(CSRR)	-E0- (B7 48 00 00 01)*	}	直線
(CSRW)	-49- 0F 60 F0		
(VECTW)	-4C- 0B BB 00 37 00 7C FF F2 00		
(VECTE)	-6C-		
(CSRW)	-49- 2F 44 F0	}	弧
(VECTW)	-4C- 26 70 00 DE 00 BC 01 FF FF 00 00		
(VECTE)	-6C-		
(CSRR)	-E0- (08 48 00 00 20)*	}	直線
(CSRW)	-49- 08 48 D0		
(VECTW)	-4C- 08 C0 00 24 00 64 FF E4 00		
(VECTE)	-6C-		
(CSRW)	-49- 2F 08 F0	}	** 弧(マスク) 注
(VECTW)	-4C- 26 29 00 BE 00 7C 01 FF FF 2B 00		
(VECTE)	-6C-		
(CSRR)	-E0- (CD 08 00 20 00)*	}	** 直線
(CSRW)	-49- CD 08 50		
(VECTW)	-4C- 08 BA 00 9A FF E0 FE 54 00		
(VECTE)	-6C-		
(CSRW)	-49- CD 48 50		
(VECTW)	-4C- 08 BA 00 9A FF E0 FE 54 00		
(VECTE)	-6C-		

(CSRW)	-49-	2F 08 F0	}	**	}
(VECTW)	-4C-	2E 87 00 BE 00 7C 01 FF FF 2B 00			
(VECTE)	-6C-				
(CSRW)	-49-	2F 48 F0	}	弧	}
(VECTW)	-4C-	2E 87 00 BE 00 7C 01 FF FF 2B 00			
(VECTE)	-6C-				
(CSRW)	-49-	04 20 00	}	**	}
(VECTW)	-4C-	23 87 00 BE 00 7C 01 FF FF 00 00			
(VECTE)	-6C-				
(CSRW)	-49-	04 E0 00	}	弧	}
(VECTW)	-4C-	23 87 00 BE 00 7C 01 FF FF 00 00			
(VECTE)	-6C-				
(CSRW)	-49-	04 20 00	}	**	}
(VECTW)	-4C-	20 65 00 BE 00 7C 01 FF FF 00 00			
(VECTE)	-6C-				
(CSRW)	-49-	04 E0 00	}	弧	}
(VECTW)	-4C-	20 65 00 BE 00 7C 01 FF FF 00 00			
(VECTE)	-6C-				
(CSRR)	-E0-	(C5 6C 00 00 40)*	}	**	}
(CSRW)	-49-	C5 2C E0			
(VECTW)	-4C-	0A A1 00 2B 00 8A FF CC 00			
(VECTE)	-6C-		}	直線	}
(CSRW)	-49-	C5 6C E0			
(VECTW)	-4C-	0A A1 00 2B 00 8A FF CC 00			
(VECTE)	-6C-				

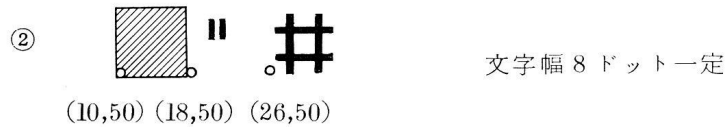
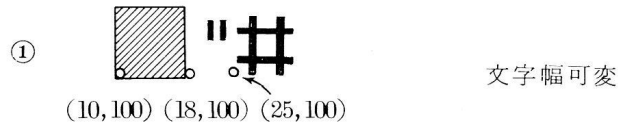
\* CSRRコマンドのあとに書いてある()内のは読み出しアドレス・データを示します。

\*\* ]: Rプレーン(赤)      ]: Gプレーン(緑)

注意. 弧(マスク)とは扇形の一方の線分(描画開始角側)を書くための座標データを得るために  
描画(マスクのため実際には描画しない)させたものです。

## 5.5.6 グラフィック文字描画（グラフィック制御時）

### (1) 文字幅変更



下の例では，文字幅は異なりますが，CSRWコマンドによる描画開始点をオーバーラップさせて， $8 \times 8$  ドット領域に対する描画をしています。

(WRITE:W)	-20-		
(TEXTW)	-78-	FF	FF
(CSRW)	-49-	60	73 A0
(VECTW)	-4C-	12	07 00
(TEXTE)	-68-		
(CSRW)	-49-	A0	79 A0
(VECTW)	-4C-	12	07 00
(TEXTE)	-68-		
(TEXTW)	-78-	00	22 22 22 00 00 00 00
(CSRW)	-49-	61	73 20
(VECTW)	-4C-	12	07 00
(TEXTE)	-68-		
(CSRW)	-49-	A1	79 20
(VECTW)	-4C-	12	07 00
(TEXTE)	-68-		
(TEXTW)	-78-	00	44 44 FE 44 FE 44 44
(CSRW)	-49-	61	73 90
(VECTW)	-4C-	12	07 00
(TEXTE)	-68-		
(CSRW)	-49-	A1	79 A0
(VECTW)	-4C-	12	07 00
(TEXTE)	-68-		

①

②

①

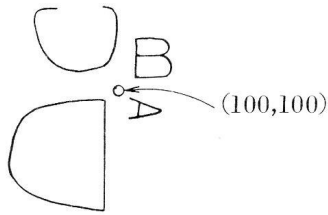
②

①

②

MOD: REPLACE

(2) 拡大グラフィック文字描画



- A : DIR=0 ZW=0 赤
- B : DIR=2 ZW=1 緑
- C : DIR=4 ZW=2 黄
- D : DIR=6 ZW=3 青

(ZOOM)	-46- 00	}	A	1 倍 COMPLEMENTモード			
(WRITE:W)	-21-						
(TEXTW)	-78- 00 38 44 82 82 FE 82 82						
(CSRW)	-49- 66 33 40						
(VECTW)	-4C- 10 07 00						
(TEXTE)	-68-						
(ZOOM)	-46- 01	}	B	2 倍			
(TEXTW)	-78- 00 7E 82 82 7E 82 82 7E						
(CSRW)	-49- 66 73 40						
(VECTW)	-4C- 12 07 00						
(TEXTE)	-68-						
(ZOOM)	-46- 02	}	C	3 倍			
(TEXTW)	-78- 00 7C 82 02 02 02 82 7C						
(CSRW)	-49- 66 33 40						
(VECTW)	-4C- 14 07 00						
(TEXTE)	-68-						
(CSRW)	-49- 66 73 40						
(VECTW)	-4C- 14 07 00						
(TEXTE)	-68-						
(ZOOM)	-46- 03				}	D	4 倍
(TEXTW)	-78- 00 3E 42 82 82 82 42 3E						
(CSRW)	-49- 66 B3 40						
(VECTW)	-4C- 16 07 00						
(TEXTE)	-68-						

## 5.5.7 文字コードの書き込み(文字制御時)

上位/下位バイト共に“00”から“FF”の同一コードを書き込みます。

---

```
(VECTW)          -4C- 02
(MASK)           -4A- FF FF
(WRITE:W)       -20- 00 00 01 01 02 02 03 03 04 04 05 05 06 06 07 07 08 08 09
09 0A 0A 0B 0B 0C 0C 0D 0D 0E 0E 0F 0F 10 10 11 11 12 12 13 13 14 14 15 15 16 16
17 17 18 18 19 19 1A 1A 1B 1B 1C 1C 1D 1D 1E 1E 1F 1F 20 20 21 21 22 22 23 23 2
4 24 25 25 26 26 27 27 28 28 29 29 2A 2A 2B 2B 2C 2C 2D 2D 2E 2E 2F 2F 30 30 31
31 32 32 33 33 34 34 35 35 36 36 37 37 38 38 39 39 3A 3A 3B 3B 3C 3C 3D 3D 3E 3E
3F 3F
(WRITE:W)       -20- 40 40 41 41 42 42 43 43 44 44 45 45 46 46 47 47 48 48 49
49 4A 4A 4B 4B 4C 4C 4D 4D 4E 4E 4F 4F 50 50 51 51 52 52 53 53 54 54 55 55 56 56
57 57 58 58 59 59 5A 5A 5B 5B 5C 5C 5D 5D 5E 5E 5F 5F 60 60 61 61 62 62 63 63 6
4 64 65 65 66 66 67 67 68 68 69 69 6A 6A 6B 6B 6C 6C 6D 6D 6E 6E 6F 6F 70 70 71
71 72 72 73 73 74 74 75 75 76 76 77 77 78 78 79 79 7A 7A 7B 7B 7C 7C 7D 7D 7E 7E
7F 7F
(WRITE:W)       -20- 80 80 81 81 82 82 83 83 84 84 85 85 86 86 87 87 88 88 89
89 8A 8A 8B 8B 8C 8C 8D 8D 8E 8E 8F 8F 90 90 91 91 92 92 93 93 94 94 95 95 96 96
97 97 98 98 99 99 9A 9A 9B 9B 9C 9C 9D 9D 9E 9E 9F 9F A0 A0 A1 A1 A2 A2 A3 A3 A
4 A4 A5 A5 A6 A6 A7 A7 A8 A8 A9 A9 AA AA AB AB AC AC AD AD AE AE AF AF B0 B0 B1
B1 B2 B2 B3 B3 B4 B4 B5 B5 B6 B6 B7 B7 B8 B8 B9 B9 BA BA BB BB BC BC BD BD BE BE
BF BF
(WRITE:W)       -20- C0 C0 C1 C1 C2 C2 C3 C3 C4 C4 C5 C5 C6 C6 C7 C7 C8 C8 C9
C9 CA CA CB CB CC CC CD CD CE CE CF CF D0 D0 D1 D1 D2 D2 D3 D3 D4 D4 D5 D5 D6 D6
D7 D7 D8 D8 D9 D9 DA DA DB DB DC DC DD DD DE DE DF DF E0 E0 E1 E1 E2 E2 E3 E3 E
4 E4 E5 E5 E6 E6 E7 E7 E8 E8 E9 E9 EA EA EB EB EC EC ED ED EE EE EF EF F0 F0 F1
F1 F2 F2 F3 F3 F4 F4 F5 F5 F6 F6 F7 F7 F8 F8 F9 F9 FA FA FB FB FC FC FD FD FE FE
FF FF
```

---









# 第6章 ハードウェア

注. この章にある回路またはブロック図は動作の考え方を示すために記載したもので、部品の偏差や温度特性を考慮した量産設計を対象としたものではありません。また、GDCに許容されるすべての周波数について動作を保証するものでもありませんので、量産の場合には充分御検討のうえ、御使用くださるようお願いいたします。

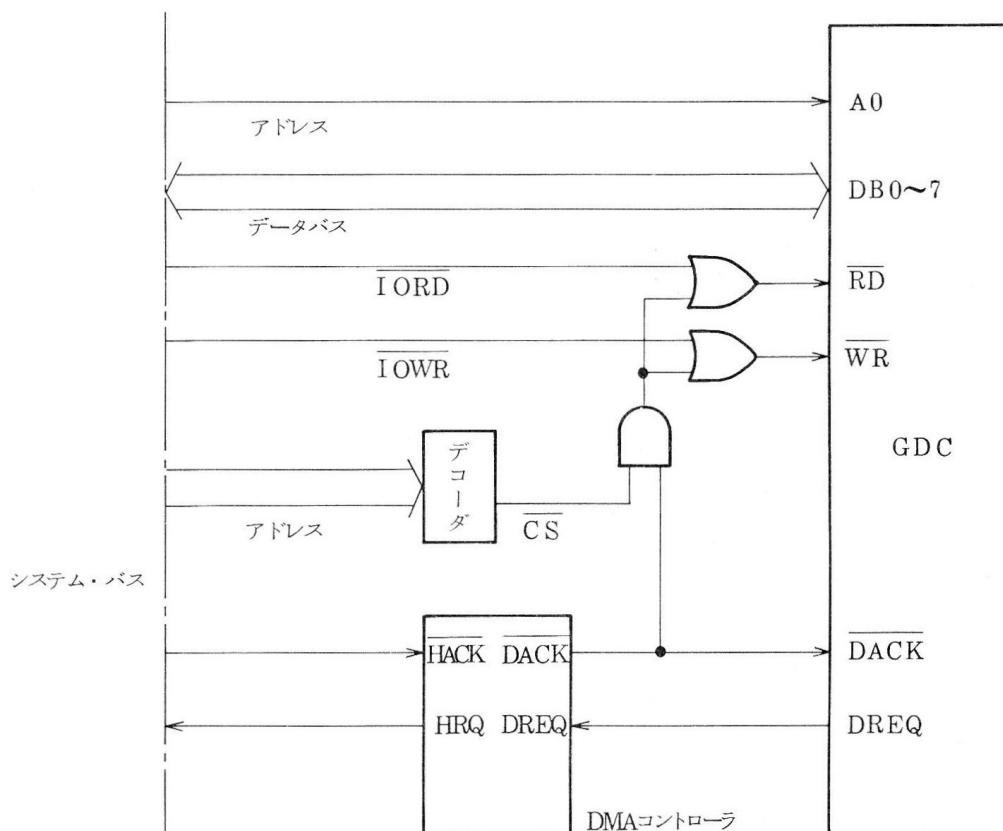
なお、記載回路例に関する特許、その他第三者の工業所有権につきましては、弊社ではその責を負いかねますので、あらかじめ御了承ください。

## 6.1 CPUインタフェース

GDCはチップ選択入力端子(CS)を持っていないため、外付け回路によって $\overline{RD}$ 、 $\overline{WR}$ 信号に組み合わせて使用します。

DMA要求出力(DREQ)およびDMA承認入力(DACK)は $\mu$ PD8257などのDMAコントローラに接続します。DMAコントローラは、映像メモリー主記憶間のデータ転送をDMAによって送受したいときに使用します。また、DMA転送を行わないときには、 $\overline{DACK}$ はHレベルにしてください。

図6-1 CPUインタフェース

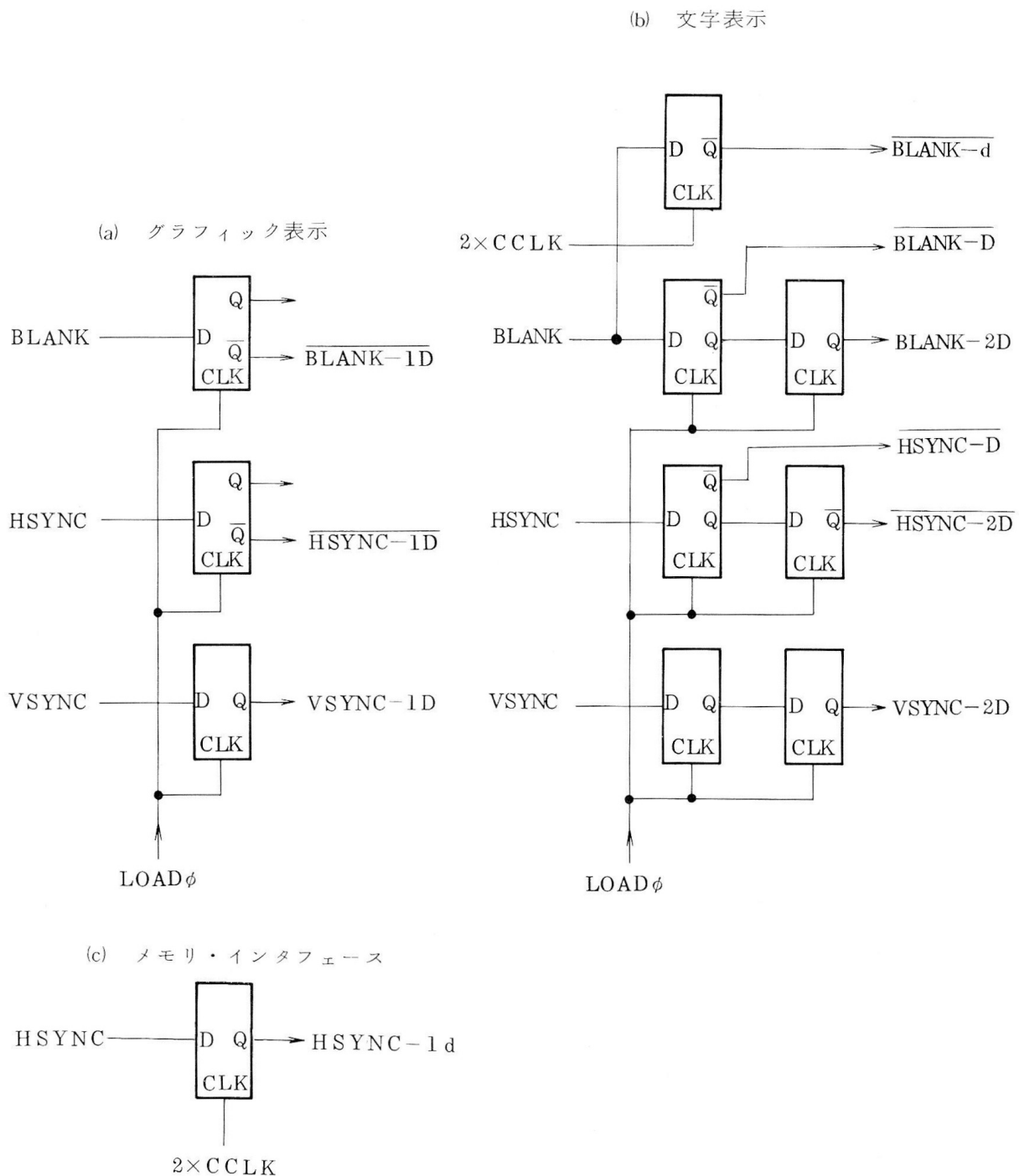


## 6.2 同期信号遅延回路

GDCから出力されるHSYNC, VSYNC, BLANKの各ビデオ関係信号出力は、映像メモリのアドレス信号やカーサ表示信号と同一のタイミングで出力されます。グラフィック表示の場合、映像メモリのデータ読み出しに2クロック(1表示サイクル)を要します。従って、BLANK信号は表示信号とタイミングを合わせるため1表示サイクルの遅延をさせる必要があります。

文字表示の場合には、映像メモリの出力をさらに文字発生用ROMに接続しドット情報を読み出すため、さらに1表示サイクルの読み出し余裕を持たせますので、文字表示時には、BLANK信号を2表示サイクルの遅延をさせる必要があります。

図6-2 同期信号遅延回路例

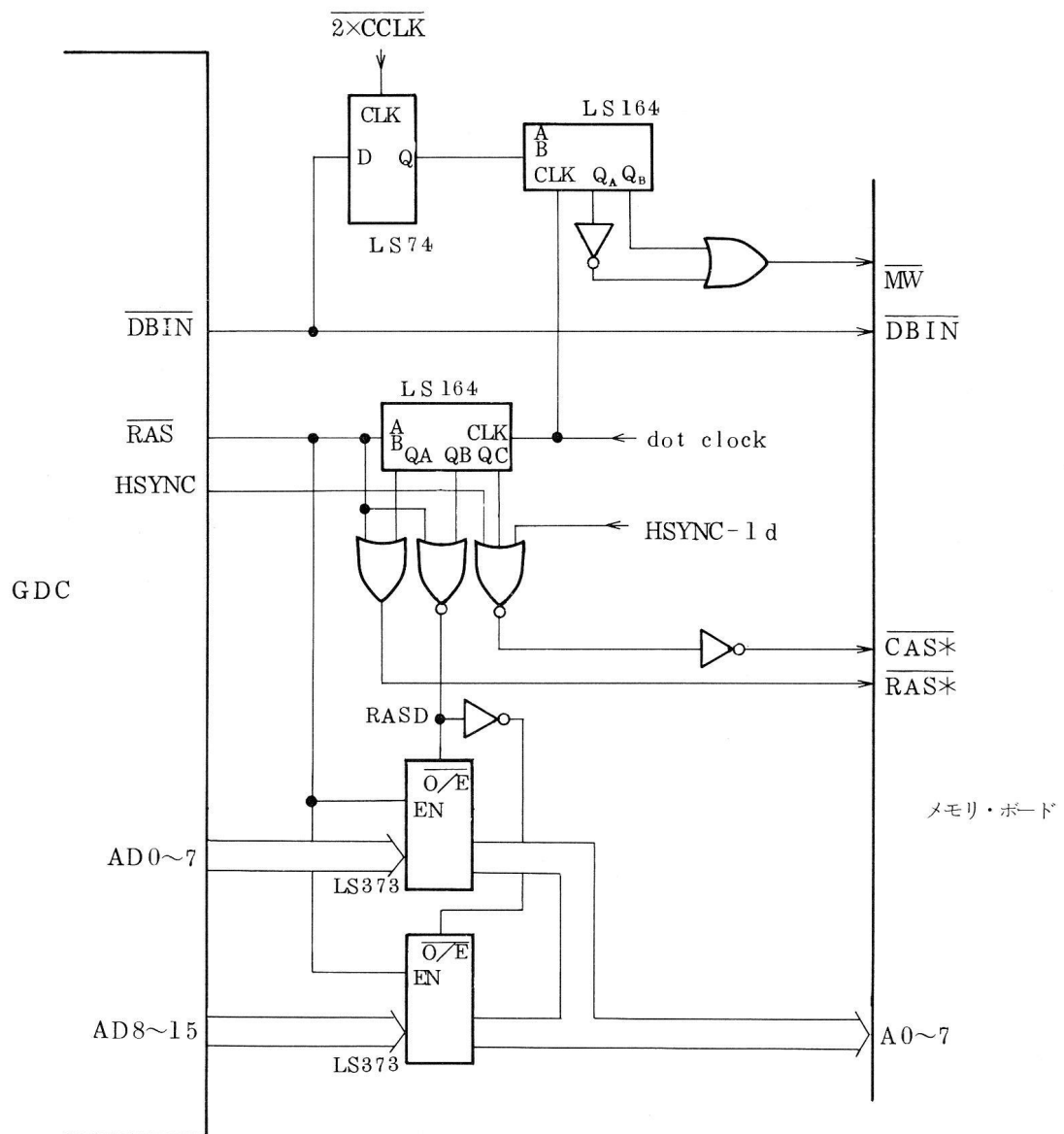


### 6.3 メモリ・インタフェース

GDCが出力する $\overline{RAS}$ 信号を基にして、16ピンDRAMに供給する $\overline{RAS^*}$ 、 $\overline{CAS^*}$ (行/列アドレス選択信号)の作成、アドレス・ラッチ回路のラッチ・イネーブル信号、出力選択タイミング信号として使用します。

また、GDCが出力する $\overline{DBIN}$ 信号は、映像メモリ内容を映像メモリ側データ・バスに乗せるタイミング信号として、また外付け回路によって遅延させ、映像メモリ書き込み信号( $\overline{MW}$ )を作成します。

図 6-3 メモリ・インタフェース



HSYNCおよびHSYNC-1d信号によって、RASオンリー・リフレッシュが行えます。また、リフレッシュ時(HSYNC="1")には下位8ビットのアドレスのみが、映像メモリに与えられるようにします。

## 6.4 ドット・クロック, GDCクロック発生回路

ドット・クロックの周波数は1水平走査期間内の表示時間と, そのときの映像メモリのアクセス回数によって決定されます. グラフィック・モード時には, 1表示期間2クロックで16ドットを表示するため, ドット・クロック8クロック周期のGDCクロックを作成します. また, 表示期間の最終タイミングで, 映像メモリ出力を並列-直列変換レジスタにロードするクロックが必要です. このクロックを発生するとき,  $\overline{RAS}$  信号によって, 2クロックの表示期間の最終であることの判別を行います.

図6-4 ドット・クロック, GDCクロック発生回路例

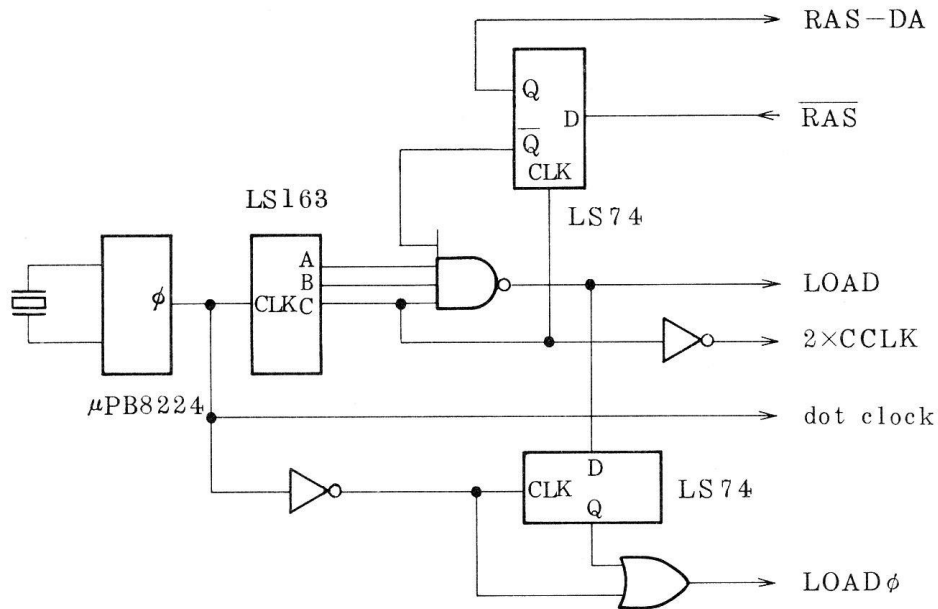
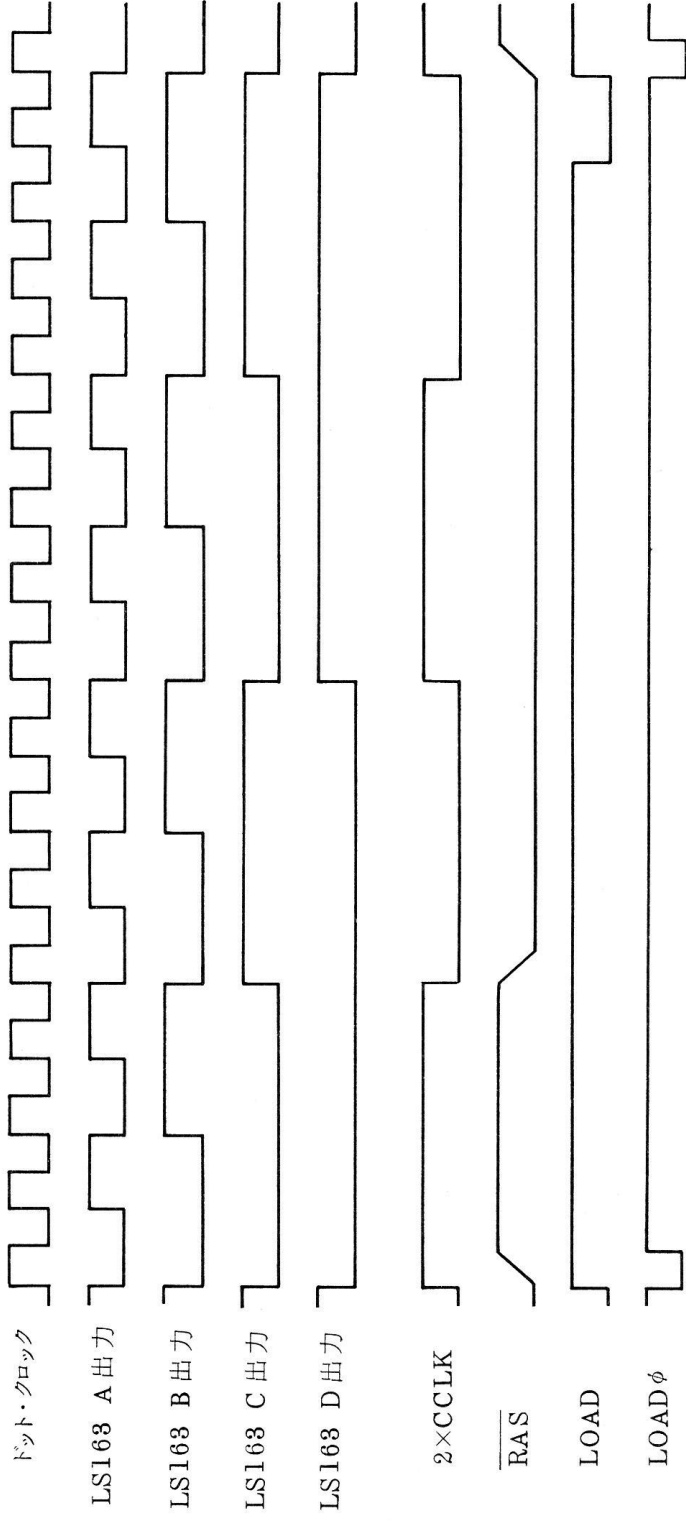


図 6-5 ドット・クロック、GDDCクロック・タイミング

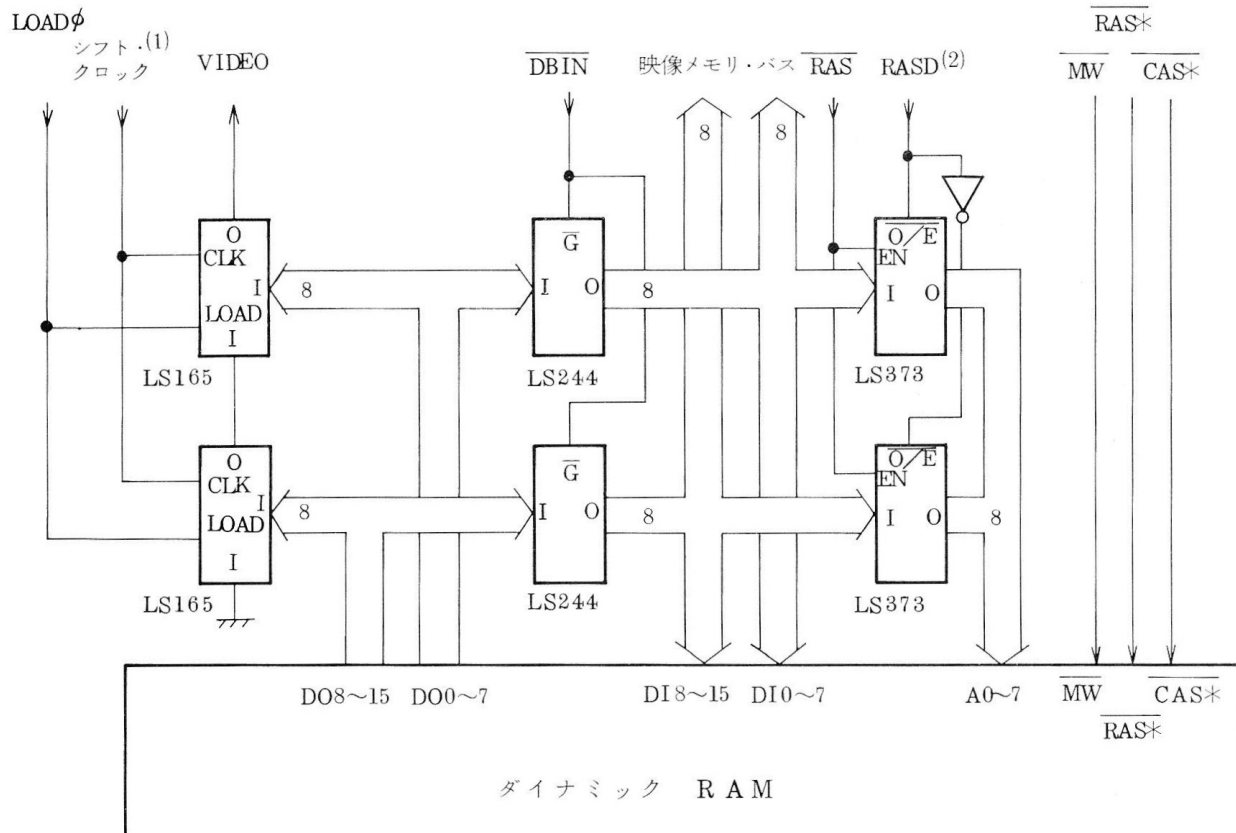




## 6.5 映像メモリ・ボード回路

映像メモリとしてD I（データ入力），D O（データ出力），を持つ一般的な16ピン・ダイナミックRAMを使用した例を示します。多色化する場合には，各メモリ・プレーン毎に，並列-直列変換レジスタを付随させます。また，1語を4ビットにして3色同時描画をする場合には，図4-14（第4章 1語を8/4/2/1ビットにする方法）を参照してください。

図6-6 映像メモリ・ボード



注意1 拡大表示をしない場合には dot clock になります。

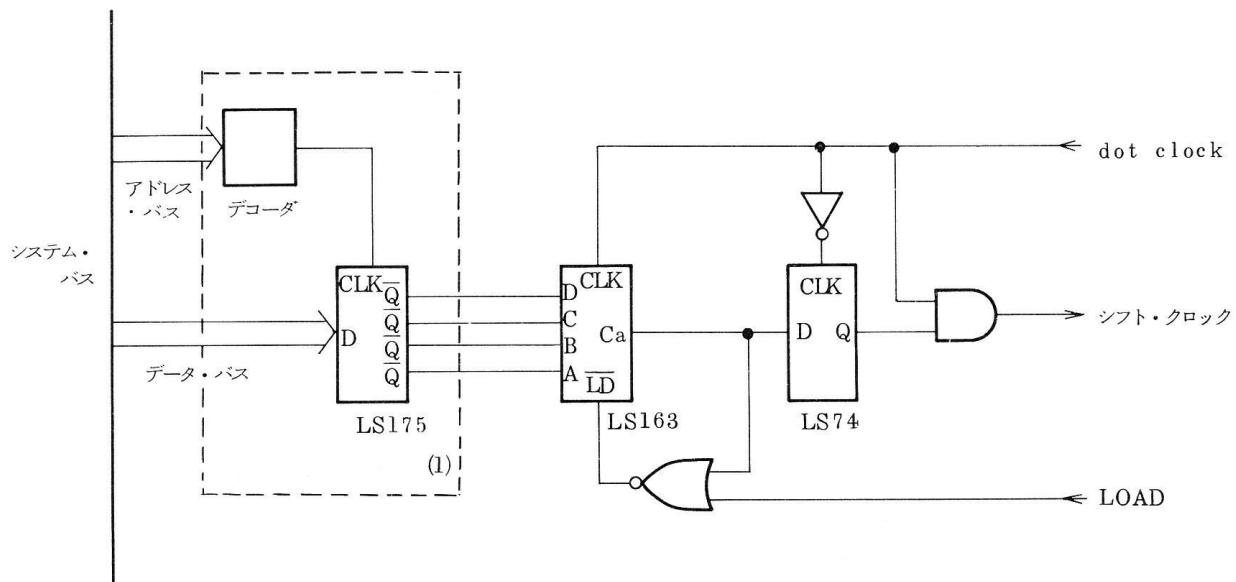
注意2 図6-3参照

## 6.6 拡大表示用回路

拡大表示を行う場合には表示アドレスの制御と、並列→直列変換器に供給するロード・クロックおよびシフト・クロックの周期を拡大係数に従って変化させる必要があります。表示アドレス制御については、GDC内で拡大係数に合わせて表示アドレスを発生していますので、ロード・クロックとシフト・クロックについて外部回路で制御します。

ロード・クロックの周期の制御は、GDCが拡大表示期間に、 $\overline{RAS}$ 出力信号を“1”レベルに保っていますので、図6-4のような回路を用いますと特に外付け回路を必要としませんが、シフト・クロックの周期制御は、図6-7のように拡大係数を記憶する4ビットのフリップフロップと、4ビット・バイナリ・カウンタによってシフト・クロックを間引くことにより行えます。図中、LOAD信号はシフト・クロックとLOAD $\phi$ との同期をとるための回路です。

図6-7 拡大表示用回路例



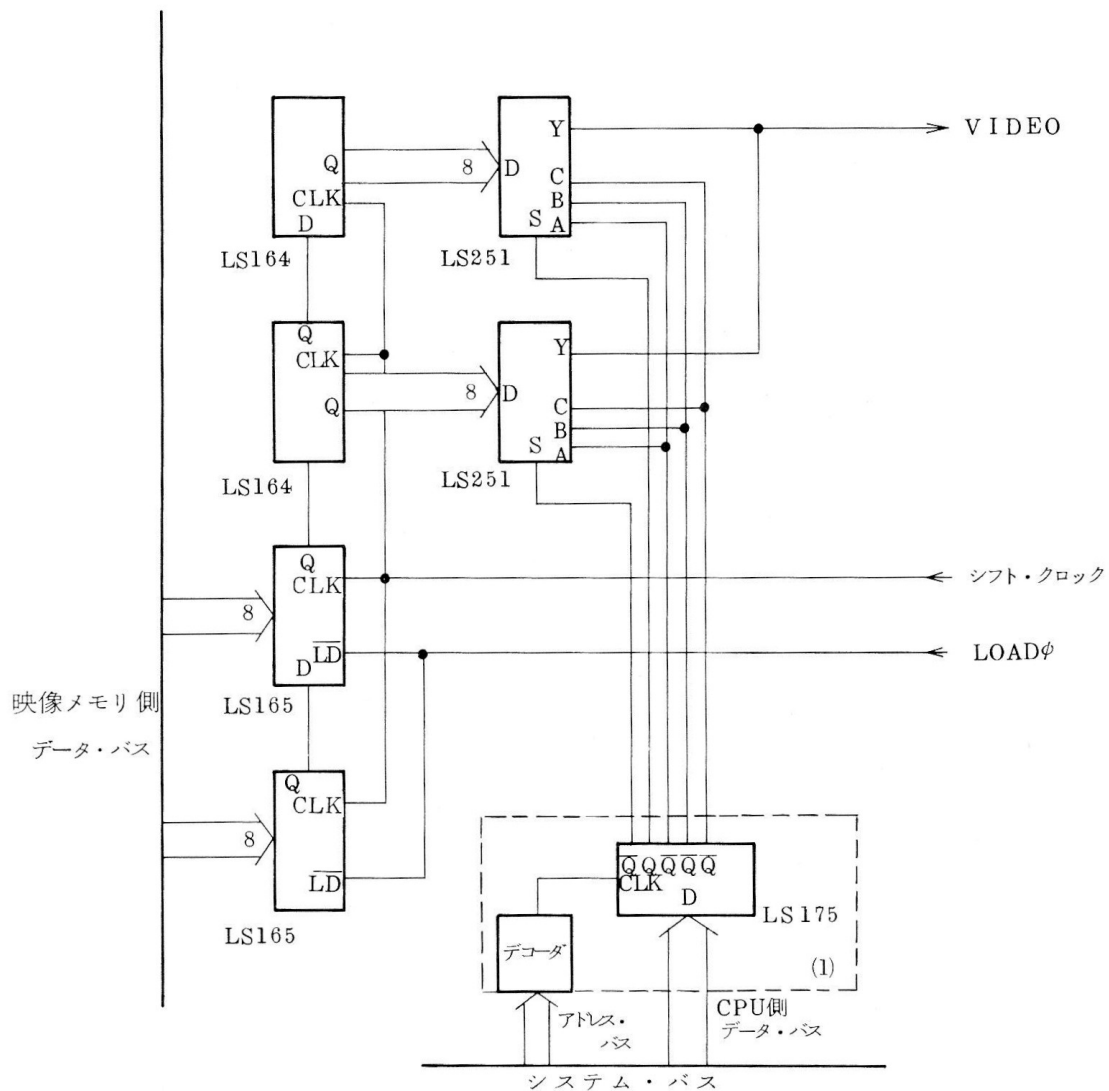
注意1 この回路はシステム・バス上にあるCPUが直接制御します。

なお、このフリップ・フロップに与えるデータは、ZOOMコマンドのZRと同じものです。

## 6.7 水平方向ドット単位スクロール用回路

図6-8のように、ドット・シフト量を記憶する4ビットのフリップ・フロップと、その内容に従って、ドット遅延を生じさせる16ビットのシフト・レジスタの信号出力端子を選択する1/8セレクタ2個によって構成される回路を付加すると、水平方向へのドット単位でのスムーズなスクロールが可能となります。垂直方向については、GDC自身が1ライン(ドット)単位でのスクロールを制御します。なお、この場合にはBLANK信号を2表示サイクル(グラフィック表示の場合の1サイクル+ドット・スクロールのための1サイクル)遅延させる必要があります。

図6-8 水平方向ドット単位スクロール用回路例



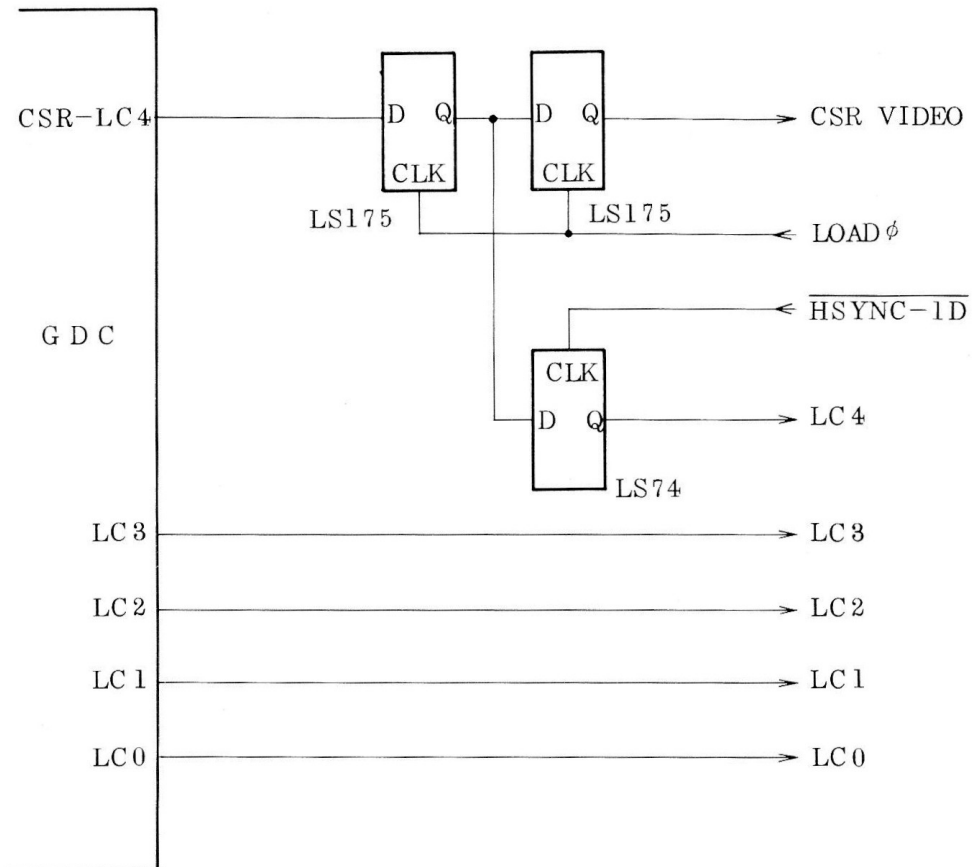
注意1 この回路はシステム・バス上にあるCPUが直接制御します。



## 6.8 カーサ出力, ライン・カウント出力取り出し回路(文字モード)

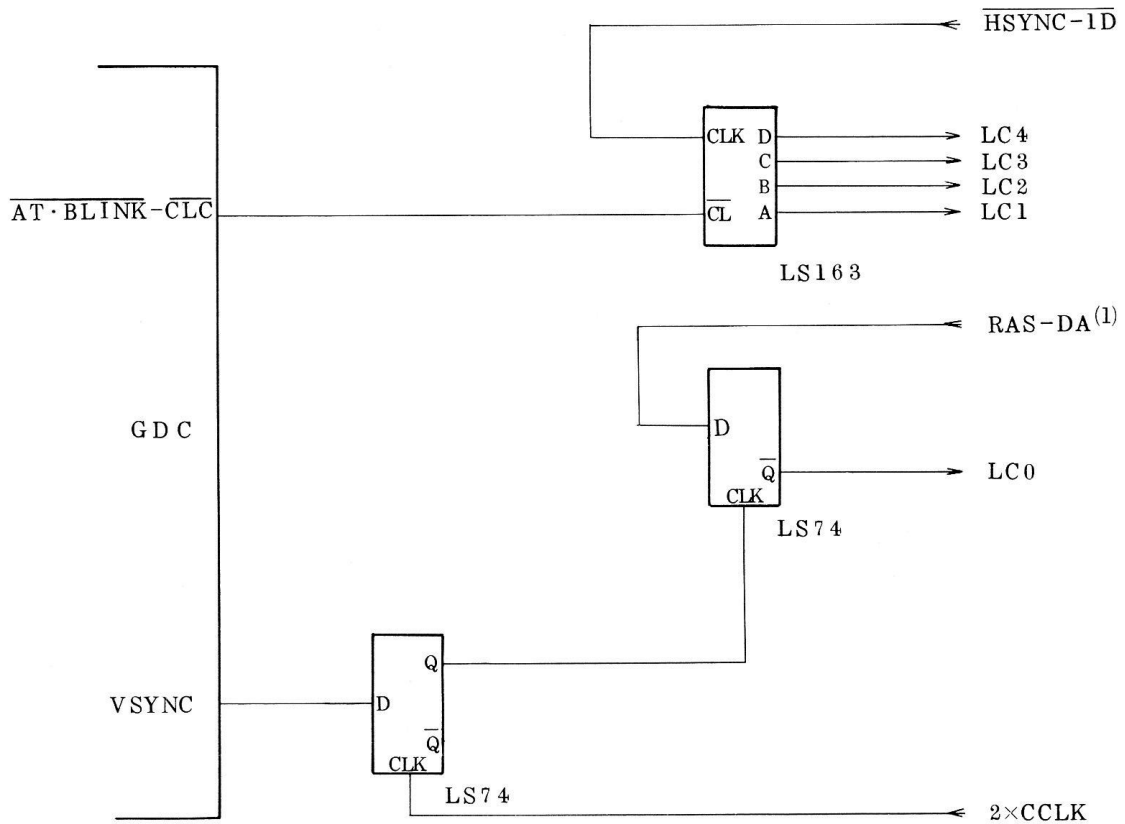
本モードでは, GDCが内蔵している5ビットのライン・カウンタを直接使用します. ライン・カウンタの5ビット目は, CSR-LC4端子から時分割されて出力されますので, HSYNC信号の立ち上がり時に外付フリップフロップに取り込みます.

図6-10 カーサ出力等の取り出し回路例(文字モード)



文字／グラフィック混在モードにおいては文字表示を行うために外部ラインカウンタを付加します。また、インタレース・シュリンクの走査モードを使用するときには、下図のように外付ラインカウンタを構成する必要があります。ただし、ライン数（CSRFORMのL/Rで指定）は偶数でなければなりません。

図3-2に示されるように、インタレース・シュリンク時には、ラインカウンタを“+2”し、第1フィールド時の初期値は“1”、第2フィールド時の初期値は“0”としますが、第1フィールドのVSYNCの信号変化は、タイミング“D1”に同期し、第2フィールドでは“D2”に同期していますので、これから第1フィールドと第2フィールドの判別を行います。

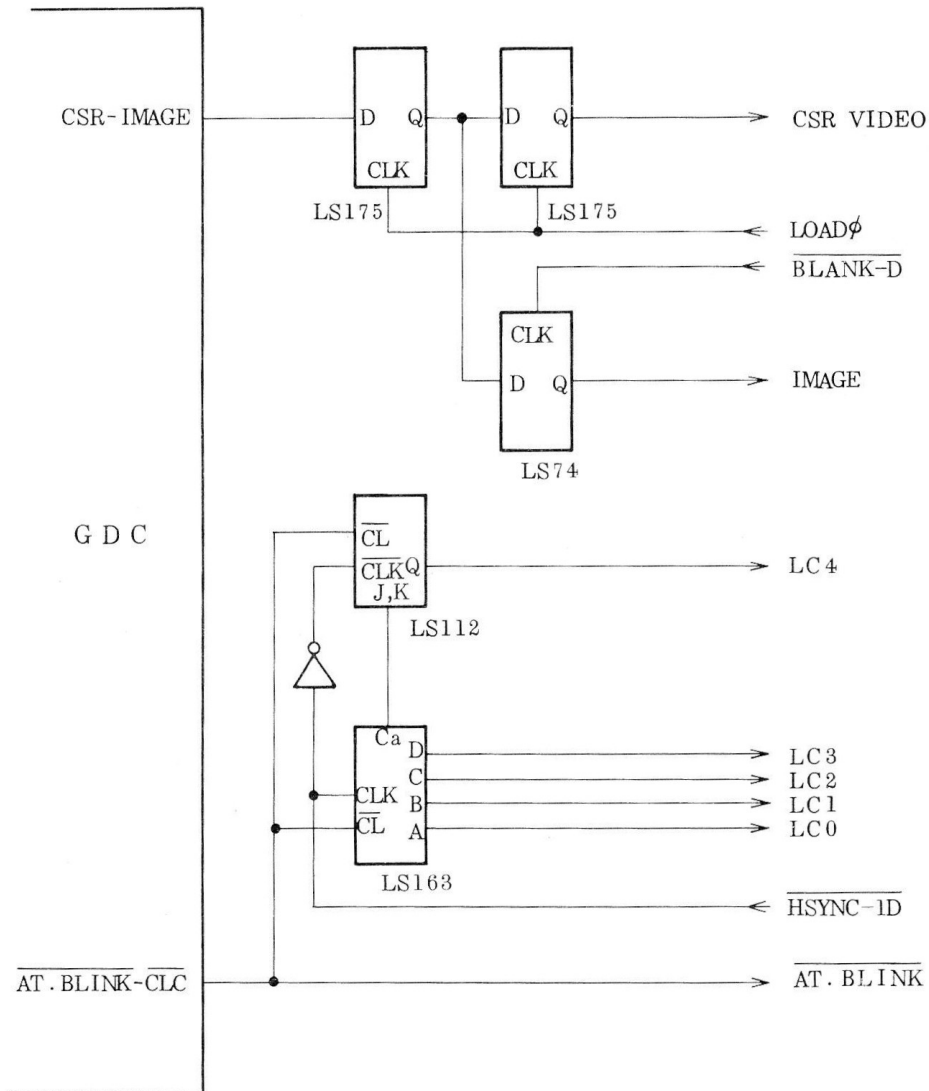


注意1 図6-4参照

## 6.9 カーサ出力等の取り出し回路(文字／グラフィック混在モード)

外付け回路によって構成するライン・カウンタを使用します。カウント・パルスとしてHSYNC信号，クリア信号として $\overline{CLC}$  (クリア・ライン・カウンタ) 信号を使用します。グラフィック(イメージ)表示領域と文字表示領域とを区分する信号(IMAGE)は，CSR-IMAGE 出力端子から時分割されて出力され，外付け回路によってBLANK信号の立ち下がり時に記憶します。

図 6-11 カーサ出力等の取り出し回路例(文字／グラフィック混在モード)

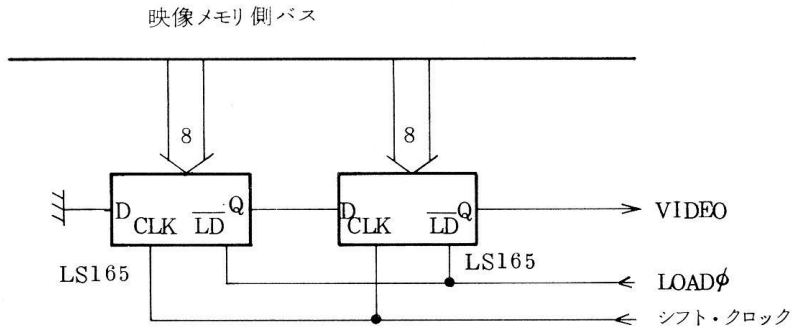


## 6.10 映像信号取り出し回路

### (1) グラフィック表示時

映像メモリの出力を並列-直列変換レジスタに接続するだけで、映像直列信号が得られます。

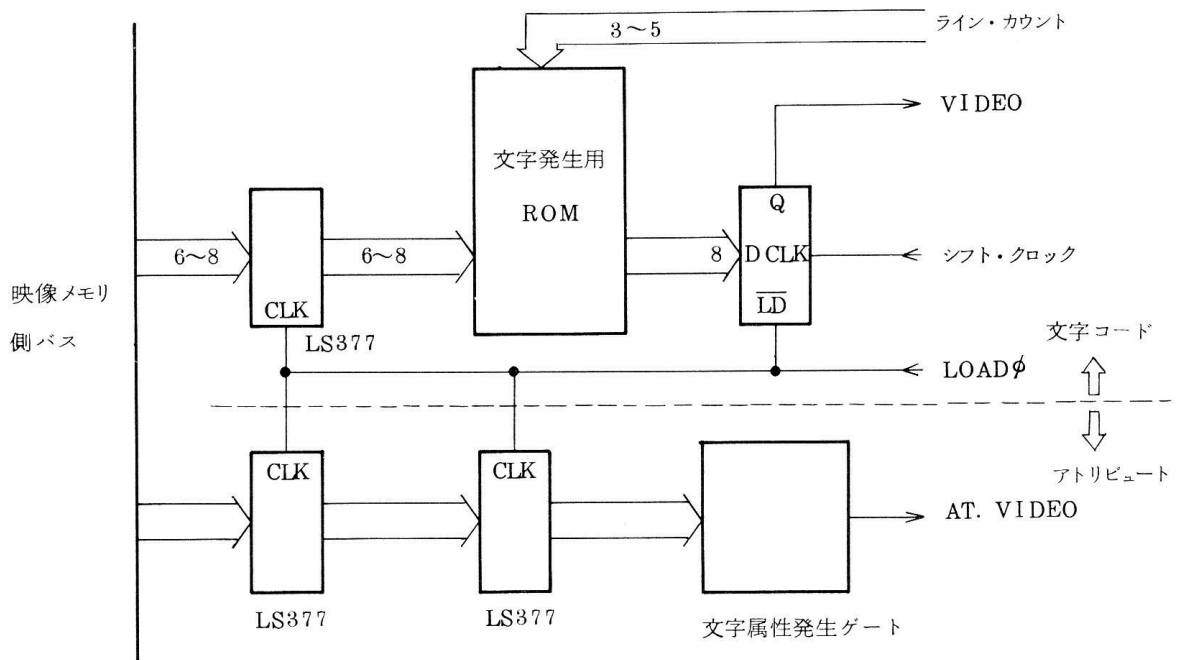
図 6-12 映像信号取り出し回路例(グラフィック表示時)



### (2) 文字表示時

映像メモリ出力をラッチした後、文字発生用ROMのアドレスにライン・カウント信号と共に接続し、ROM出力を並列-直列変換レジスタに接続し映像直列信号を取り出します。文字属性信号はフリップ・フロップに記憶し文字属性決定ゲートに接続します。

図 6-13 映像信号取り出し回路例(文字表示時)





### (3) 文字／グラフィック混在表示時

文字表示とグラフィック表示とを1個のGDCで表示画面を水平方向に分割して、同時表示することができます。SCROLLコマンドのIMビット＝“1”としたときには、1表示サイクルが、文字表示時の2クロックから、4クロックになり初めの2クロックで下位バイトを次の2クロックで上位バイトを並列一直列変換器に接続するようにします(図6-15, 図6-16参照)。

図6-14 映像信号取り出し回路例(文字／グラフィック混在表示時)

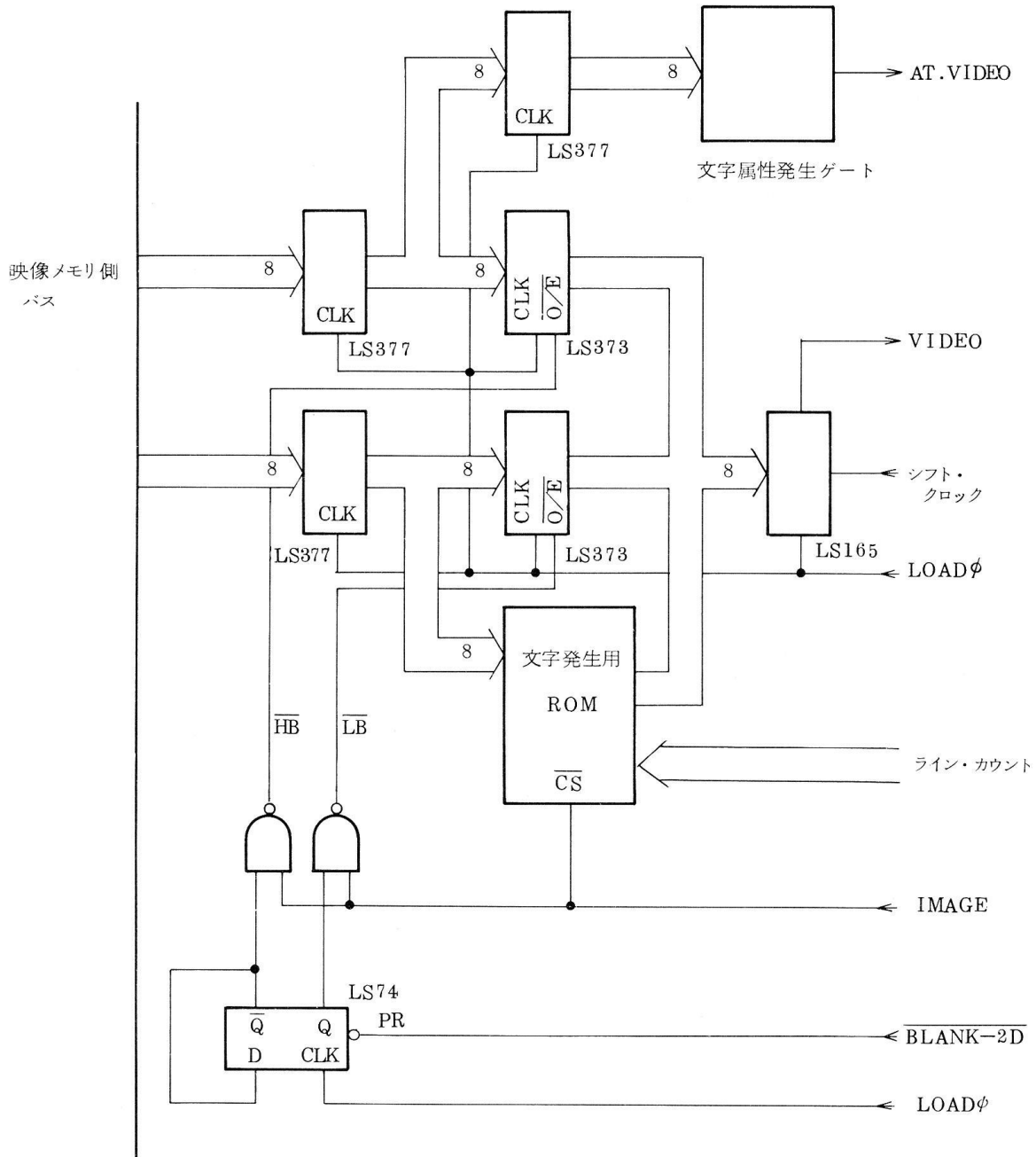


図 6-1-5 IM=1, 文字/グラフィック混在モードのタイミング

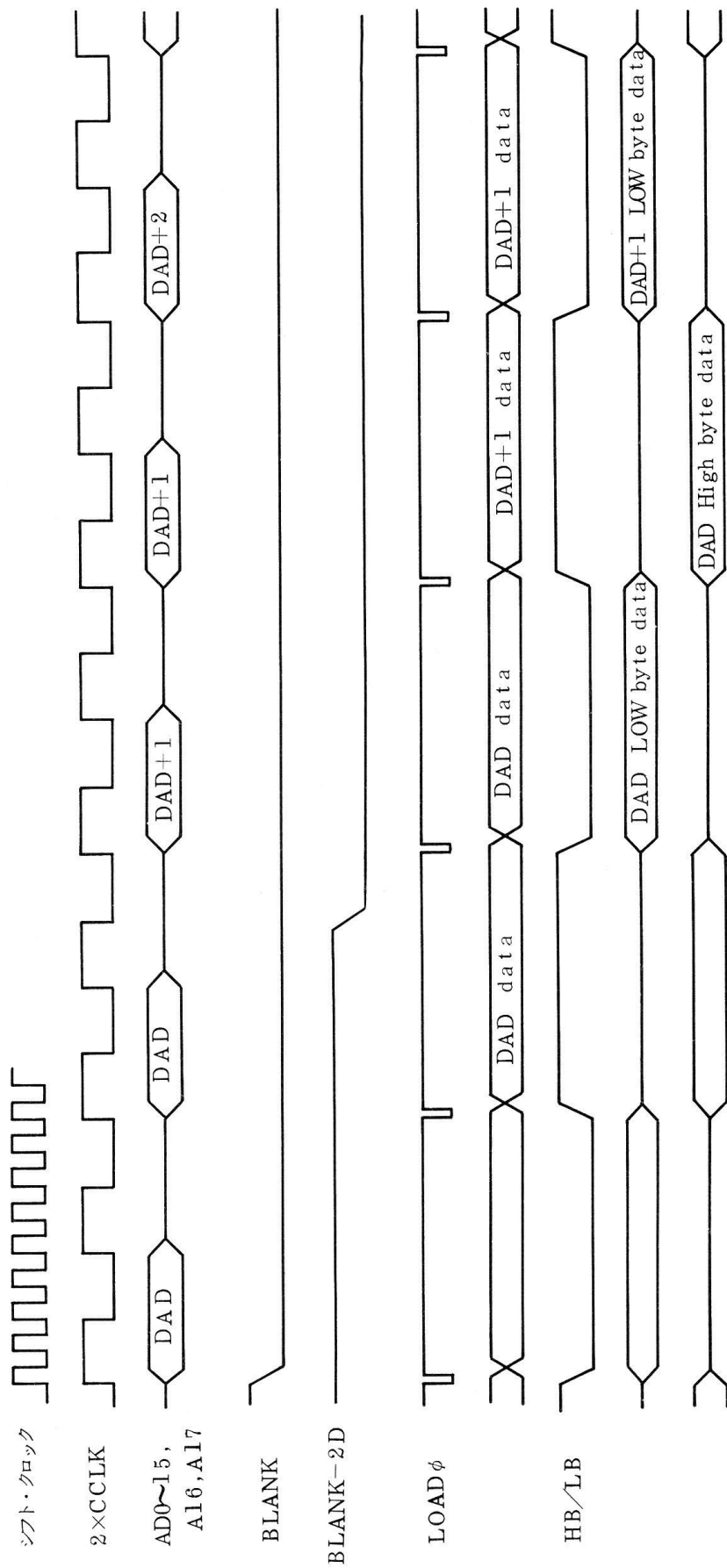
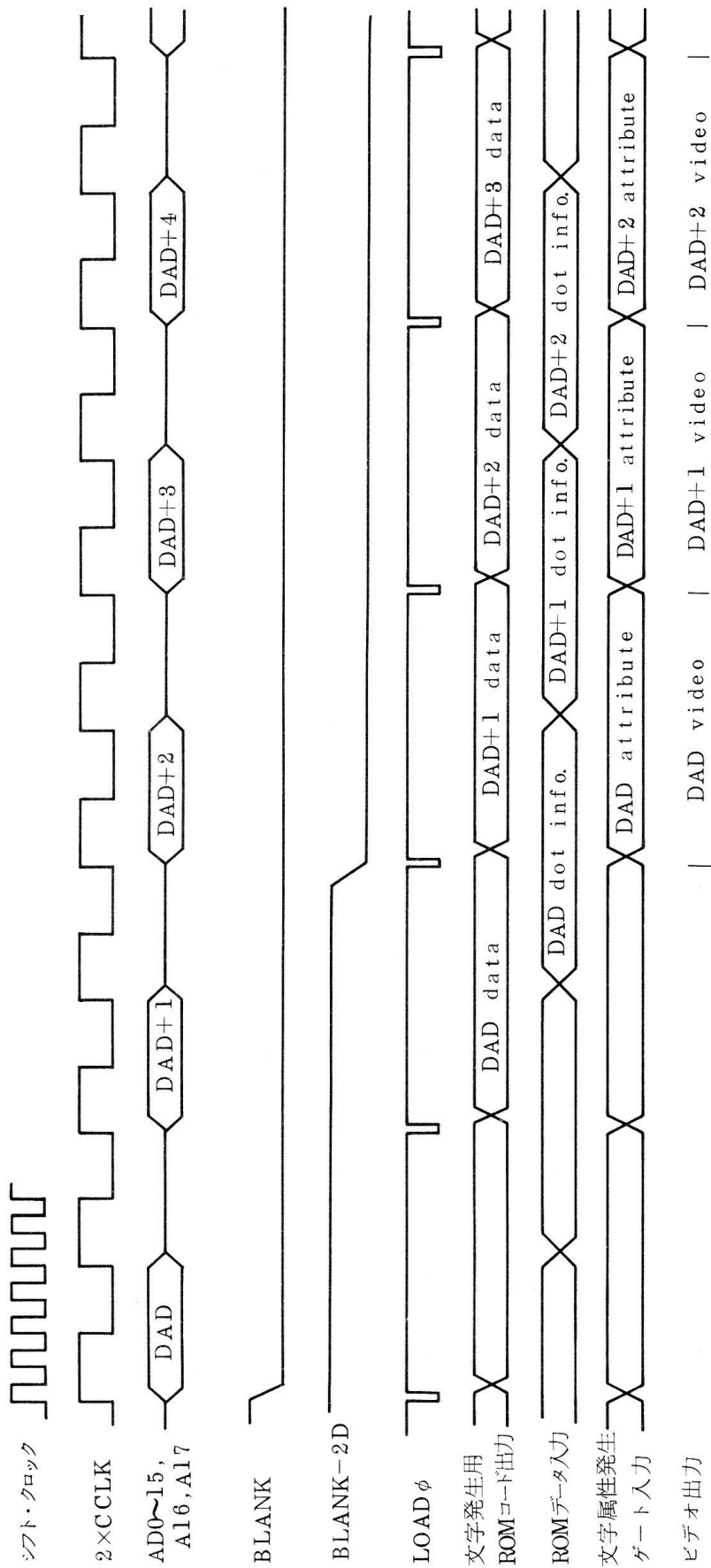


図 6-16 IM=0, 文字/グラフィック混在モードのタイミング



## 6.11 文字表示における垂直方向へのスムーズ・スクロール用回路

図6-17に示すように、スクロール・ライン数を記憶するフリップ・フロップとその内容によりフランク信号を操作する回路と表示開始アドレス等を制御するソフトウェアを付加することにより、文字表示においても簡単に垂直方向へのスムーズ・スクロールを実現することができます。

### (1) ソフトウェア

SYNCコマンドで設定したVBPからスクロールするライン数を減じてVBPを再設定するとともに、SCROLLコマンドで設定した表示ライン数SLにスクロールするライン数を加算しSLを再設定します。

$$\text{つまり } VBP = VBP - N$$

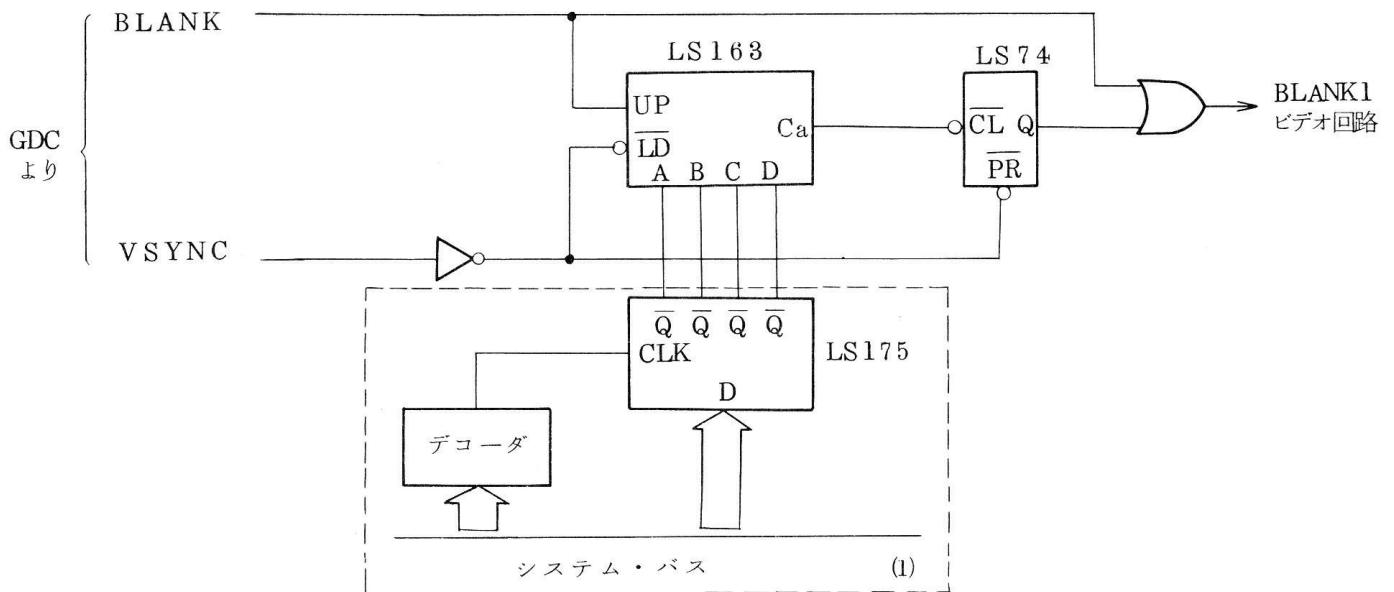
$$SL = SL + N$$

ただし、N：スクロール・ライン数

となります。

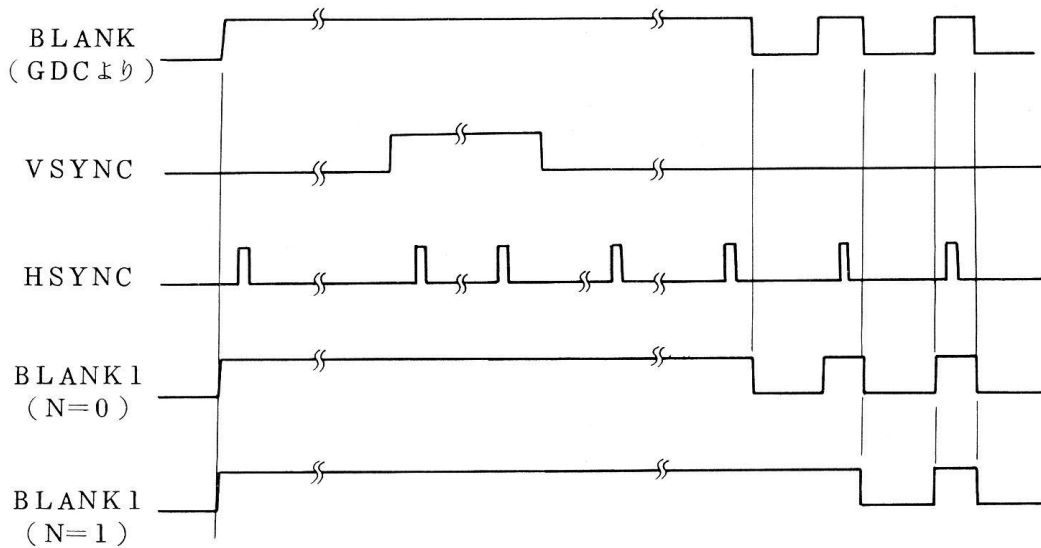
### (2) ハードウェア

図6-17 垂直方向へのスムーズ・スクロール回路（文字表示時）



注意] この回路はシステム・バス上にあるCPUで直接制御します。このときフリップ・フロップにセットする値はスクロールしたいライン数にします。

この回路はVBPのときにGDCから出力されるBLANK信号を74LS175にセットされた値により制御する回路です。

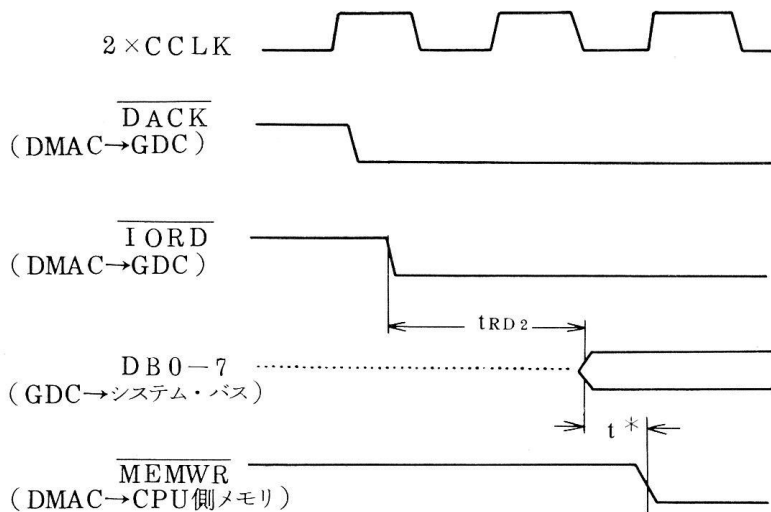


## 6.12 DMAコントローラ・インタフェース回路

### 6.12.1 システム・メモリに対するライト・データのセットアップ時間について

DMA転送を行う場合にはGDC側から出力される転送データのセットアップ時間に注意する必要があります。図6-18に示すようにDMAコントローラからGDCに対して $\overline{DACK}$ 信号が出力され、 $\overline{IORD}$ 信号が出力されます。GDCからの転送データはこの $\overline{IORD}$ 信号から $t_{RD2}$ の遅延で出力されます。次に、DMAコントローラはシステム・メモリに対して $\overline{MEMWR}$ 信号を出力しシステム・メモリはその制御信号を使用してGDCから出力されたデータを書き込みます。このときにDMAコントローラから出力される $\overline{MEMWR}$ 信号に対するデータのセットアップ時間( $t^*$ )がシステム・メモリのデータ・セットアップ時間を満足する必要があります。

図6-18 ライト・データのセットアップ時間



$\overline{\text{MEMWR}}$  信号に対するデータ・セットアップ時間がシステム・メモリの規格を満足しないような場合には、次のような方法によって  $\overline{\text{MEMWR}}$  信号に対するデータ・セットアップ時間を満足させることができます。

- (i) システム・メモリに  $\overline{\text{MEMWR}}$  信号の前縁（立ち下がり）でデータを書き込む方式のメモリを使用する場合には、DMA コントローラのクロック周波数 ( $f_{\text{CLK(DMAC)}}$ ) と GDC のクロック周波数 ( $f_{\text{CLK(GDC)}}$ ) との間に

$$f_{\text{CLK(DMAC)}} \leq f_{\text{CLK(GDC)}}$$

の関係を満たすようにすれば、 $\overline{\text{MEMWR}}$  信号の前縁に対するデータ・セットアップ時間を満足することができます（つまり、DMA コントローラのクロック周波数と GDC のクロック周波数と等しいか、あるいは遅くします）。

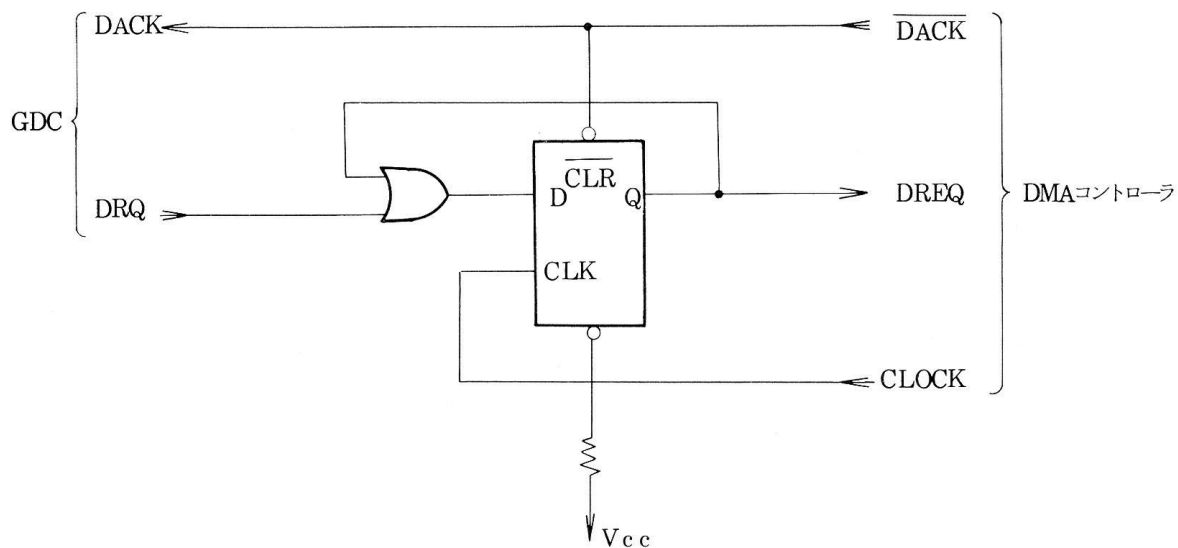
- (ii) システム・メモリに  $\overline{\text{MEMWR}}$  信号の後縁（立ち上がり）でデータを書き込む方式のメモリを使用するような場合（例えば、スタティック RAM を使用する場合には）は DMA コントローラに WAIT を挿入することにより  $\overline{\text{MEMWR}}$  信号の後縁に対するデータ・セットアップ時間を満足することができます。

## 6.12.2 DREQ 信号の保持

GDC は DMA 転送要求コマンドが入力されると DMA 要求信号 (DRQ) をアクティブとしますが、HBLANK 時になると強制的にインアクティブにします。また、DMA コントローラの使用法として DMA 要求信号は DMA 要求源に対し DMA アクノリッジ信号 ( $\overline{\text{DACK}}$ ) がもどされるまでアクティブ状態を保持しておかなければなりません ( $\mu\text{PD8257-5}$ ,  $\mu\text{PD8237A-5}$  カタログ参照)。

従って、GDC と DMA コントローラ ( $\mu\text{PD8257-5}$ ,  $\mu\text{PD8237A-5}$  等) を接続する場合には図 6-19 に示すような外部回路が必要となります。

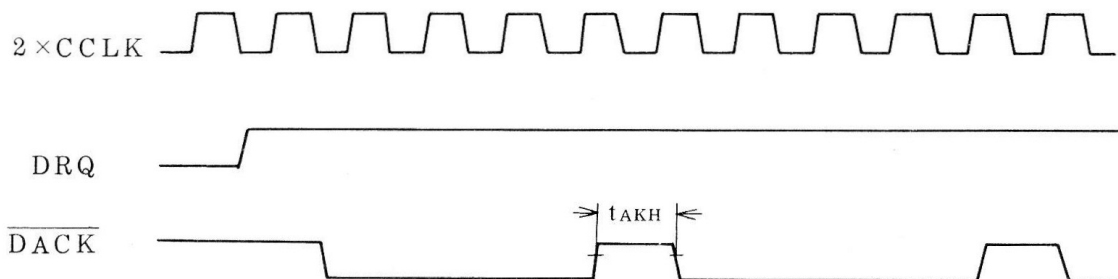
図6-19 DMAコントローラ・インタフェース



### 6.12.3 $\overline{\text{DACK}}$ 信号について

GDCは $\overline{\text{DACK}}$ 信号を利用して内蔵カウンタをデクリメントし転送ワード数(バイト数)をカウントするため、 $\overline{\text{DACK}}$ 信号は1バイト転送が終了するたびに1度"H"レベルにする必要があります(図6-20参照)。

図6-20  $\overline{\text{DACK}}$ 信号タイミング

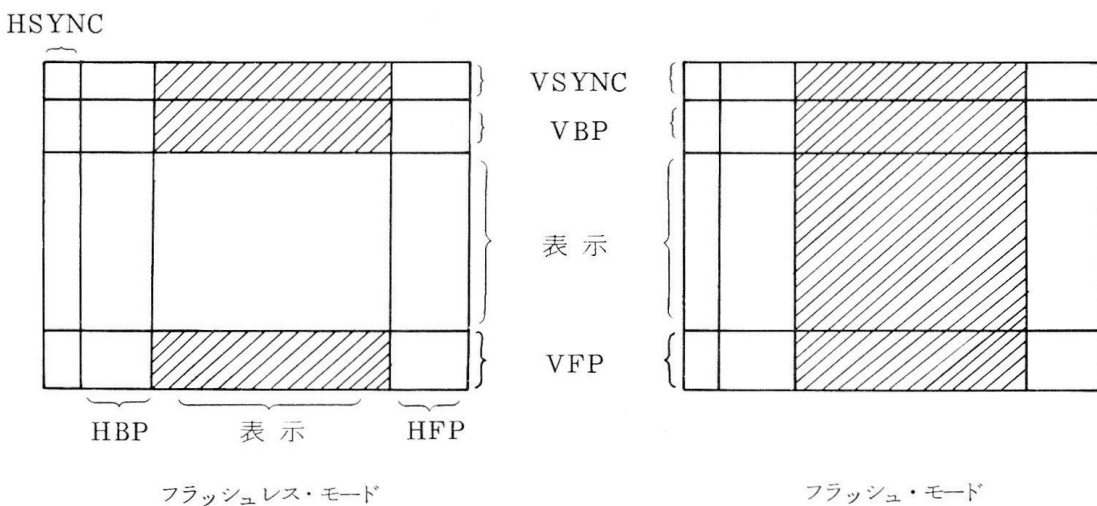


### 6.12.4 2次元矩形領域に対するDMAについて

バイト指定のDMA転送を2次元矩形領域に対して行う場合(VECTWコマンド中のパラメータDC≧0に設定)には、1回の転送に5クロックを必要とします。その他の場合(バイト指定のDMA転送で1次元領域を設定、あるいはワード指定のDMA転送)には4クロックを必要とします。

図6-21にGDCからDMA要求発生タイミングを示します。

図6-21 DMA要求発生タイミング



□ 部分はDMA要求禁止時間帯

## 6.12.5 HBLANK時のDMA転送について

DMA転送時には図6-22, 図6-23に示す条件を満たす必要があります。

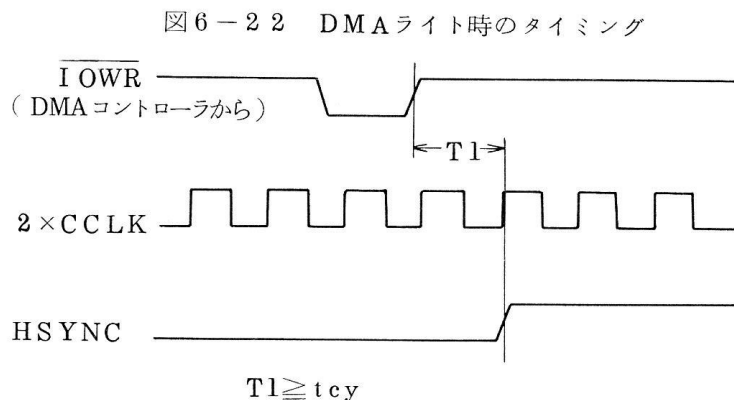
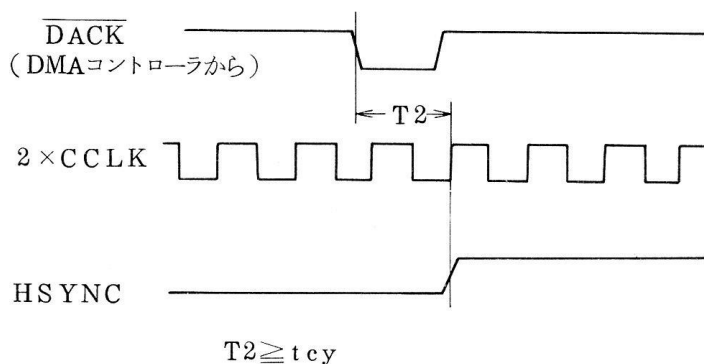


図6-23 DMAリード時のタイミング



通常のDMAコントローラでは $\overline{\text{DACK}}$ 信号は $\overline{\text{IOWR}}$ 信号より前にアクティブとなるため、DMAライト時の条件を満たしていればDMAリード時の条件は満足していることとなります。そこで、DMA動作時において前記条件を満たすことを考えると、GDCは水平帰線時間にはDMA要求を行わないので水平帰線時間に入った直後に $\overline{\text{IOWR}}$ 信号がアクティブとなった場合が最悪条件となります(図6-24参照)。

なお、図6-24ではGDCのクロックの逆相がDMAコントローラのクロックとして使用されている場合を示しています。



図 6-24 HBLANK時の DMA タイミング

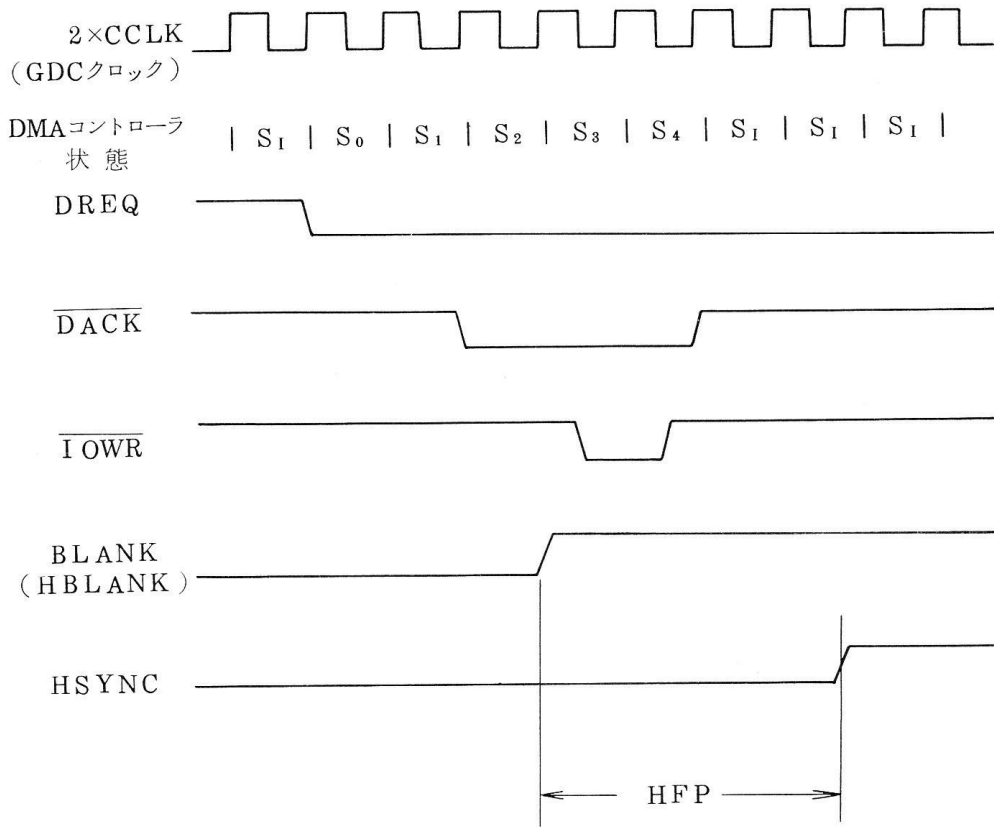


図 6-24 より明らかなように HFP の時間幅を広げることにより前記の条件を満たすことができます。また、GDC のクロックと DMA コントローラのクロックが同じであり、GDC 以外に、DMA 要求源がない場合には、

$$HFP \geq 2 \text{ 文字 (4 クロック)}$$

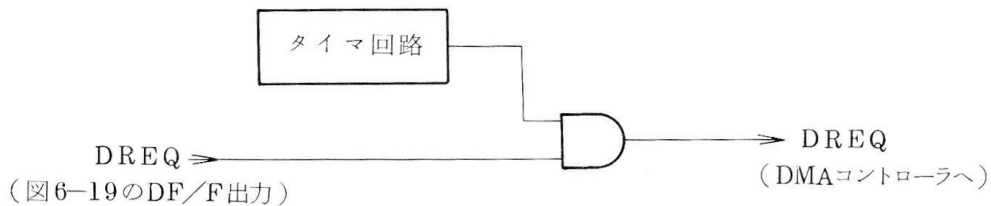
とすることで前記の条件を満たすことができます。

HFP の時間幅を広げることができない場合には次のような方法があります。

(i) DREQ 信号のインアクティブ化を早める

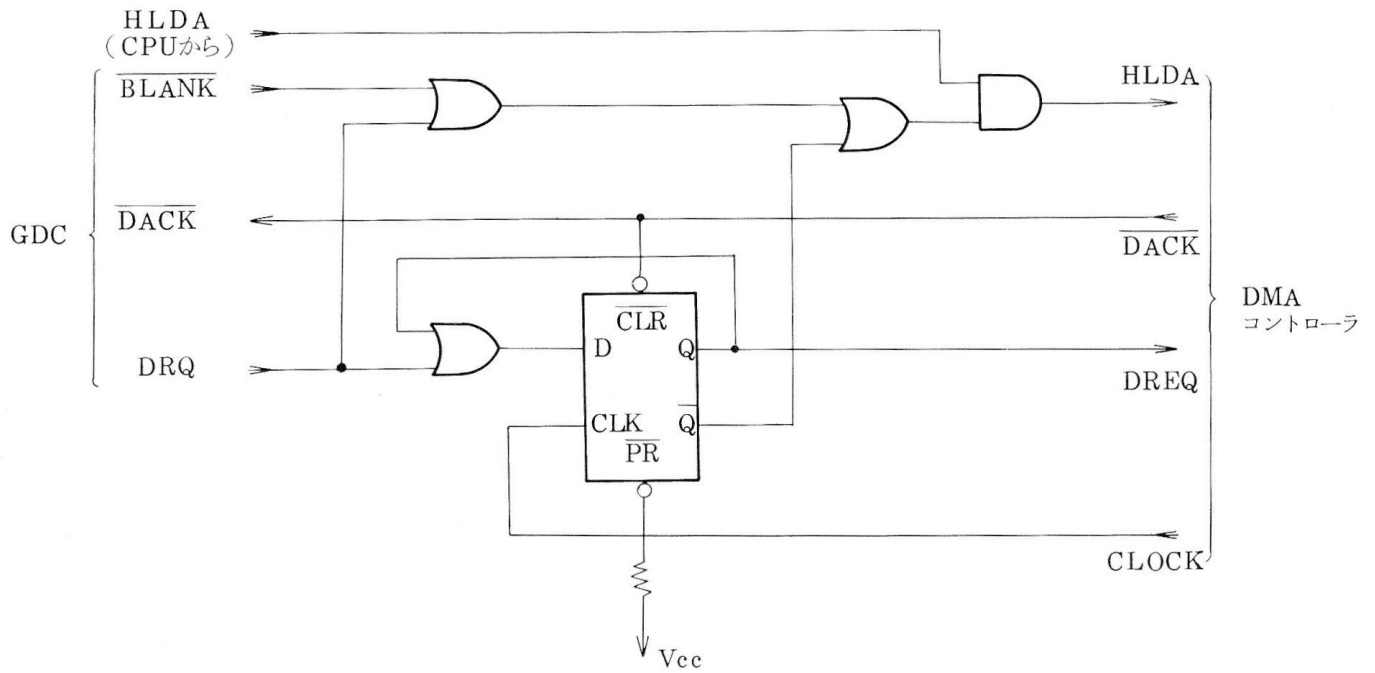
外部にタイマ回路を設けて、水平帰線時間になる前に DREQ 信号をインアクティブとします。

図 6-25 DMA コントローラ・インタフェース



- (ii) GDCからDMA転送要求があった場合には他からのDMA要求を禁止する  
 (CPUからのHLDA信号をインアクティブにする)。

図6-26 DMAコントローラ・インタフェース



なお、Dフリップ・フロップ部分は図6-19で説明した回路です。

### 6.13 マスタ・スレーブ動作を行った場合の2×CCLKの発生回路

同一システム内でGDCを2個マスタ・スレーブ動作で並列に動かす場合の2×CCLK, および各種同期信号の発生回路例を図6-27に示します. この例では, マスタGDCを文字モード(1文字期間=9ドット分), スレーブGDCを文字/グラフィック混在モードのグラフィック制御(1文字期間=8ドット分)のみとして使用しています. また, そのタイミング図を図6-28に示します.

図6-27 2×CCLK生成回路および同期信号生成回路例

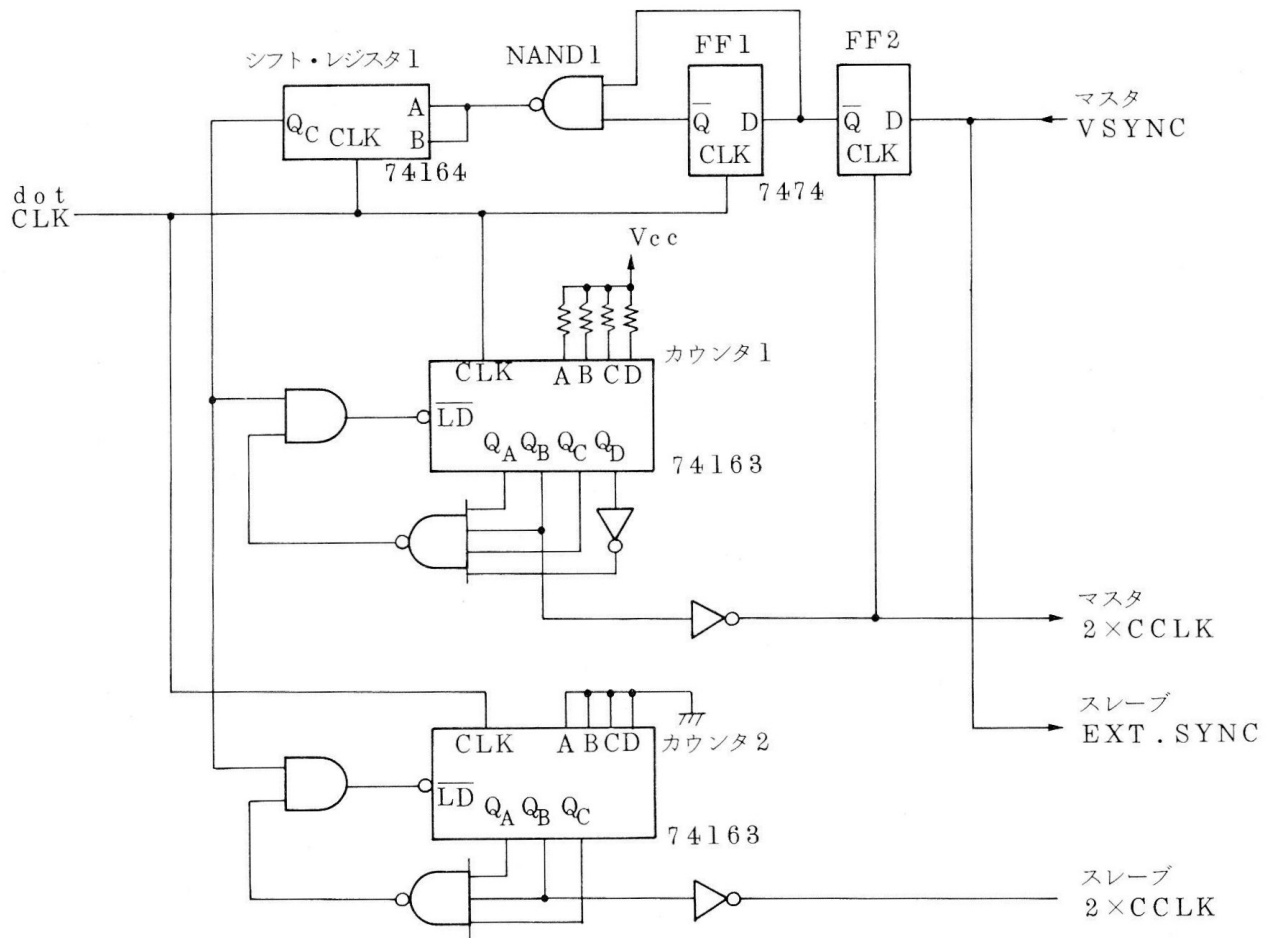
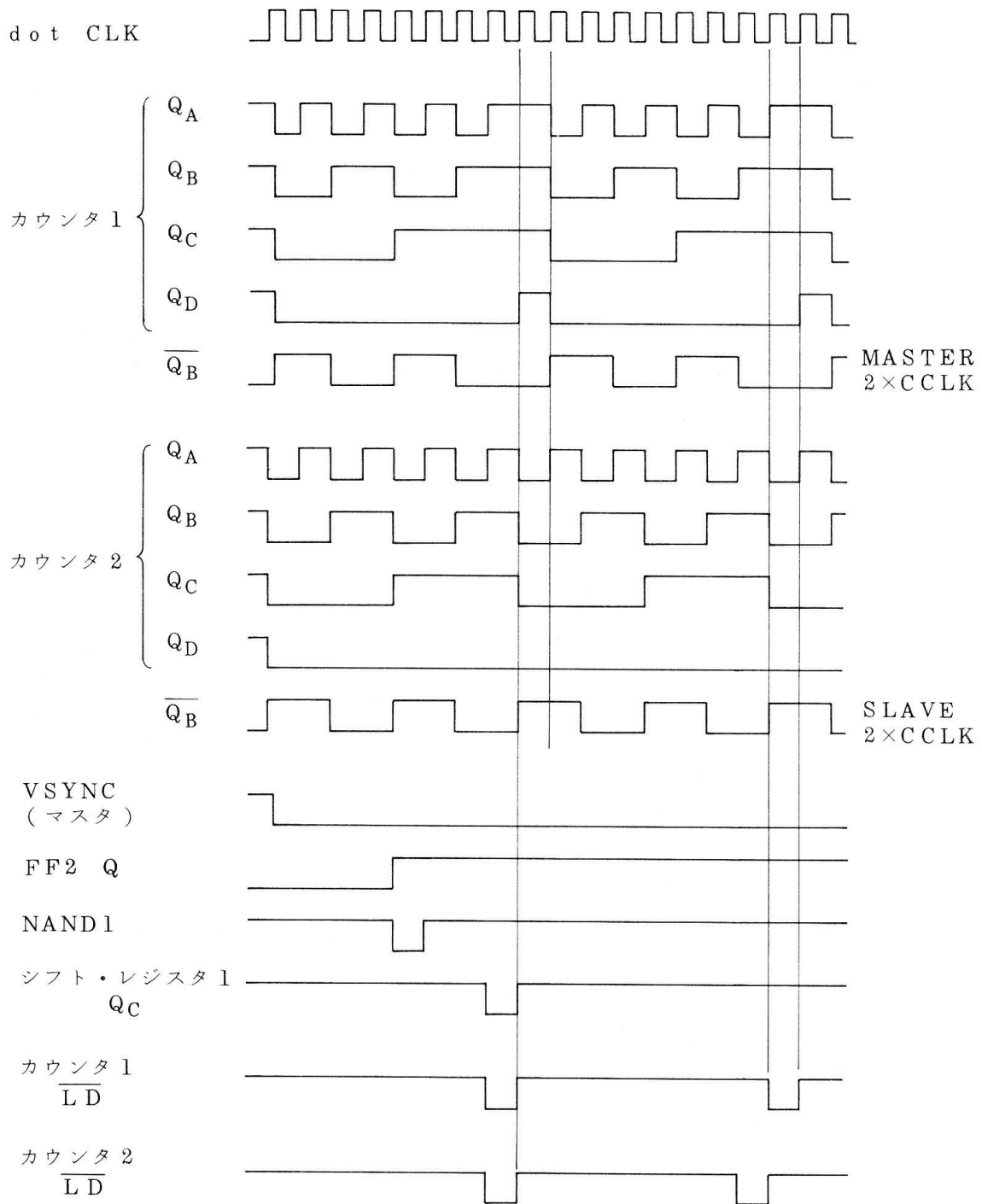


図 6-28 図 6-27 回路のタイミング図



## 6.14 NTSC信号との同期について

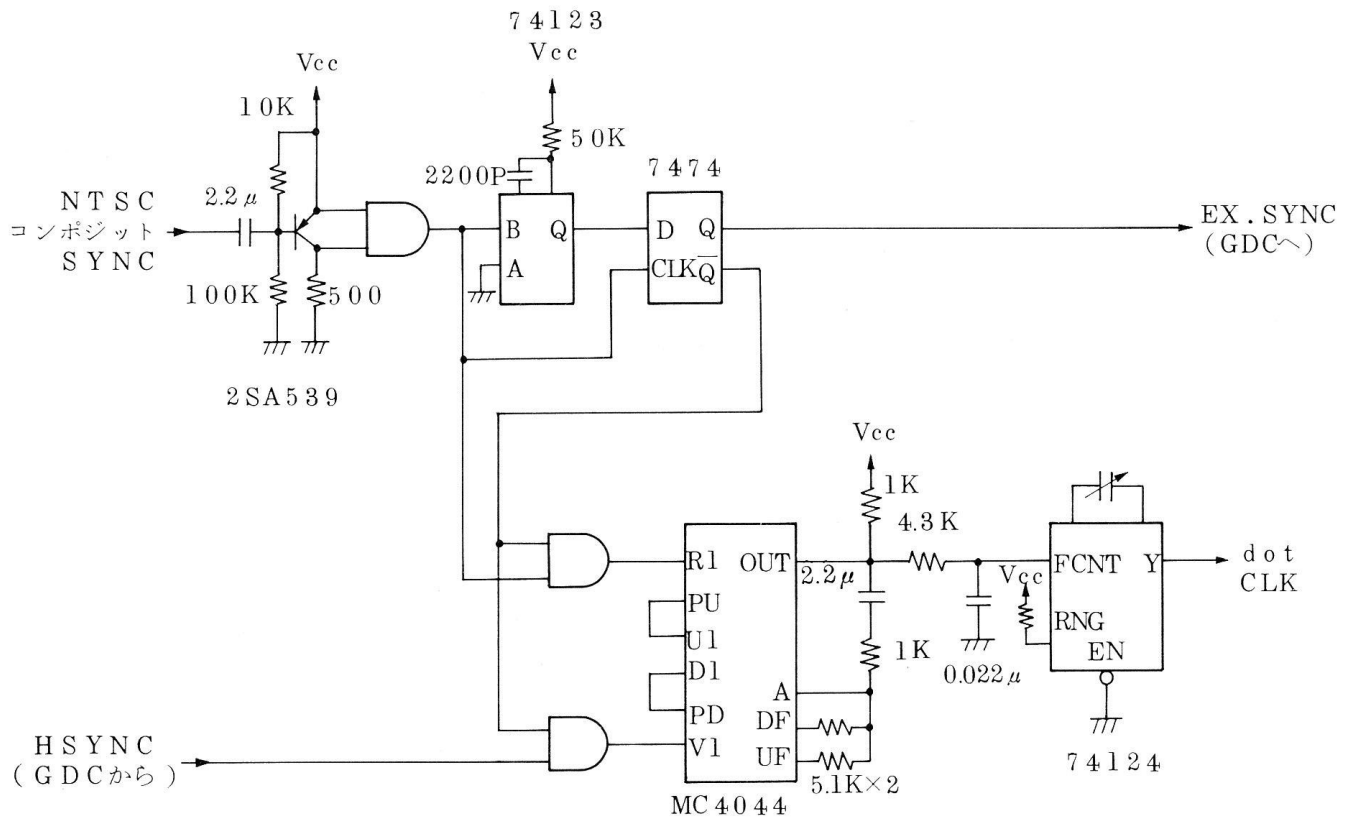
GDCとVTR，ビデオ・ディスク等の外部映像機器とを接続する場合の同期をとる回路例を図6-29に示します。

外部機器から供給されるNTSCのコンポジットSYNC信号を2SA593によりスライスし，ANDゲートにより整形することでTTLレベルのSYNC信号を取り出します。

MC4044はデジタルPLLで，位相検波，チャージ・ポンプ，アクティブ・ローパス・フィルタ用アンプを内蔵しています。この回路では2SA539により取り出されたHSYNC信号と，GDCから出力されるHSYNC信号の差を検出することで，チャージ・ポンプを駆動し，外付け抵抗，コンデンサから構成されるフィルタのコンデンサのチャージを利用して，74124(VCO：電圧制御形発振器)の発振周波数を制御しています。

74123および次段の7474はGDCに与えるVSYNC信号を生成しています。また，NTSCコンポジットSYNC信号には等価パルスが含まれていますので，この期間中にはPLLへの入力を禁止する必要があります。

図6-29 NTSC信号との同期回路例



## 6.15 インタレース時のVSYNC発生タイミングの修正について

第4章で述べたようにGDCの発生するVSYNCのタイミングは、どのようにしても第1フィールドと第2フィールドの時間長に差が出てきます。これはモニタによっては片寄った表示をする事になり、表示品質を著しく悪くする場合があります。

この時間差を修正するための回路例を図6-30に示します。GDCのVSYNC出力が第1フィールドでは表示サイクルの“D1”，第2フィールドでは“D2”に同期して出力される関係を用いて、第1フィールドと第2フィールドの識別をした後、第1フィールドでのVSYNC信号の立ち上がり（図ではロウ・アクティブ）を遅延させることにより、等間隔の走査を可能にしています。なお、FF2, FF3は第1フィールドと第2フィールドの判別を行う回路です。

図6-30 インタレース時のVSYNC発生タイミング修正回路例

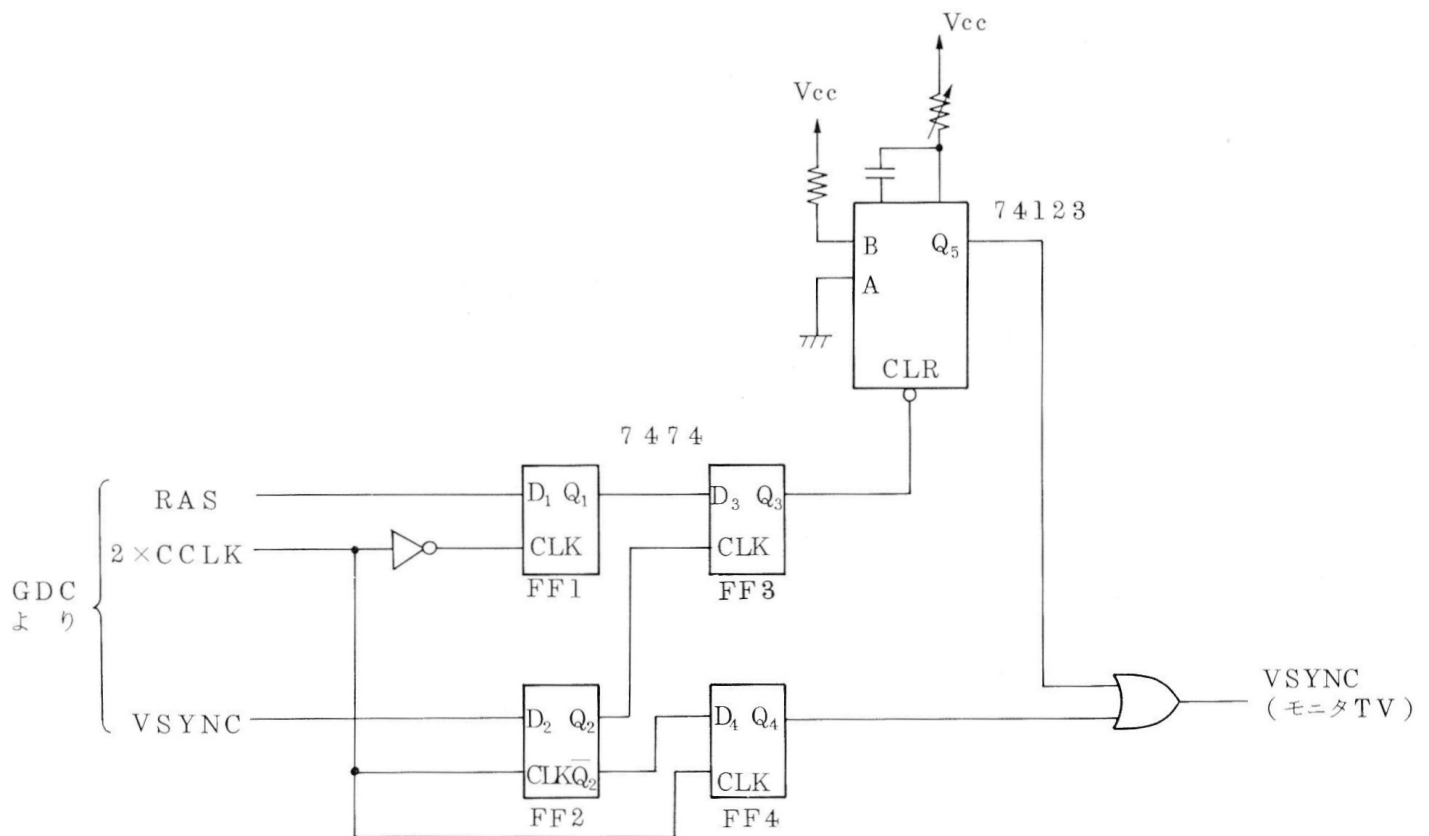


図 6-31 図 6-30 回路のタイミング図

