

VDPの機能

目次

1 VDPに関する基礎知識 1	4 VDPコマンド..... 10
1.1 VDPの種類 1	4.1 V9938・V9958のコマンド 10
1.2 VDPのレジスタ 1	4.2 VDPコマンドにおける座標系 10
1.3 VRAM 2	4.3 ロジカルオペレーション 10
1.4 I/Oポート 2	4.4 各コマンドの説明 10
1.5 TMS9918A互換モード 2	4.5 コマンド終了時のレジスタの状態 18
1.6 画面構成 2	4.6 コマンド処理の高速化 18
1.7 パレットテーブル 2	
2 基本入出力..... 3	5 スプライト 18
2.1 コントロールレジスタのアクセス 3	5.1 スプライトに関する基礎知識 18
2.2 ステータスレジスタのアクセス 3	5.2 スプライトモード1 19
2.3 パレットレジスタのアクセス 3	5.3 スプライトモード2 21
2.4 VRAM (VIDEO RAM) のアクセス..... 4	5.4 スプライトに対するハードウェアスクロールの影響 23
3 VDPレジスタの機能 4	6 その他 23
3.1 コントロールレジスタ R#0 ~ R#23、R#25 ~ R#27、 R#32 ~ R#46 (Write only) 4	6.1 V9958で削除された機能 23
3.2 ステータスレジスタ S#0 ~ S#9 (Read only) 9	6.2 コントロールレジスタR#9のIL・EO 24
	6.3 水平帰線割り込み 24

1 VDPに関する基礎知識

この章では、VDPを操作するに当たっての基本的な知識を示します。

1.1 VDPの種類

VDPにはMSXのバージョンに応じて、いくつかの種類が存在します。MSXのバージョンとVDPの種類対応は表1-1の通りです。

表1-1 MSXのバージョンとVDP

機種	VDP
MSX1	TMS9918A
MSX2	V9938
MSX2+	V9958
MSXturboR	V9958

1.2 VDPのレジスタ

VDPは、その機能を外部から利用するための“VDPレジスタ”を内部に持っています。VDPレジスタ(以下、レジスタ)は、その機能によって、以下の3つに大別されます。

1. コントロールレジスタ
2. ステータスレジスタ
3. パレットレジスタ

1.2.1 コントロールレジスタ (R#0 ~ R#23、R#25 ~ R#27、R#32 ~ R#46)

VDPの動作を制御する書き込み専用の8bitレジスタ群です。一般にR#nという記号で表します。R#0 ~ R#23・R#25 ~ R#27は主に画面

モードの設定に使用し、R#32 ~ R#46はVDPコマンドの実行に使用します。

VDPによっては存在しないレジスタもあるので注意が必要です。VDPと存在するレジスタ番号の関係を表1-2に示します。

表1-2 VDPとコントロールレジスタの対応

コントロールレジスタ番号	VDPの種類	
R#0 ~ R#7	TMS9918A	V9938 V9958
R#8 ~ R#23		V9938 V9958
R#25 ~ R#27		V9958
R#32 ~ R#46		V9938 V9958

1.2.2 ステータスレジスタ (S#0 ~ S#9)

VDPから得られる各種情報を読み取るために存在する、読み出し専用の8bitレジスタ群です。一般にS#nという記号で表します。

TMS9918AにはS#0のみ存在します。

1.2.3 パレットレジスタ (P#0 ~ P#15)

カラーパレットを設定するために存在する、書き込み専用のレジスタ群です。一般にP#nという記号で表します。nはパレット番号を意味し、各パレットを512色中の1色に設定します。1つのパレットレジスタは9bitの長さを持ち、RGB各色につき、それぞれ3bitずつを割り当てています。

TMS9918Aは、固定パレットであり、パレットレジスタは存在しません。また、カラーコード0は透明色として扱われます。

カラーパレットは、システムの初期化時に表1-3のように設定されます。この設定値は、TMS9918Aの固定パレットと同等です。「0」は輝度が最も低く、「7」は輝度が最も高いことを表します。

GRAPHIC7(SCREEN 8)では、カラーパレットを使用しません。
また、GRAPHIC5(SCREEN 6)では、P#0 ~ 3のみ使用します。

表1-3 システム初期化時のカラーパレット設定値

カラーコード	G	R	B
0	0	0	0
1	0	0	0
2	6	1	1
3	7	3	3
4	1	1	7
5	3	2	7
6	1	5	1
7	6	2	7
8	1	7	1
9	3	7	3
10	6	6	1
11	6	6	4
12	4	1	1
13	2	6	5
14	5	5	5
15	7	7	7

1.3 VRAM

TMS9918Aには16KbytesのVRAMを、V9938・V9958には128KbytesのVRAMと64Kbytesの拡張RAMを接続できます。このメモリ空間にアクセスするために、TMS9918Aは14bit、V9938・V9958は17bitのアドレスカウンタを持っています。なお、このメモリはVDPが管理するものであり、CPUが直接アクセスすることはできません。

V9938・V9958の拡張RAMは、その内容を画面に表示することはできませんが、VDPコマンド実行時にはワークエリアとして、VRAMと同等に扱うことができます。ただし、MSX規格にこの拡張RAMは入っていません。

1.4 I/Oポート

VDPはCPUとデータをやり取りするために、2つ、もしくは4つのI/Oポートを持っています。このポートはCPUのI/O空間を通して接続されています。各ポートの機能を表1-4に示します。

なお、表中m、nで表されたアドレスは、それぞれMAIN ROMの0006H、0007H番地に記録されています。アプリケーションプログラムは、必ずこの番地を参照してVDPへのアクセスを行ってください。ただし、MSXturboRでは規格上、VDPのI/Oポートが98H～9BHに固定されたため、MAIN ROMを参照せずにm = n = 98Hとしてプログラムを作成しても問題ありません。

表1-4 VDPのポート

ポート	アドレス	機能
ポート#0 (READ)	m	VRAMからデータを読み出す
ポート#0 (WRITE)	n	VRAMにデータを書き込む
ポート#1 (READ)	m + 1	ステータスレジスタを読み出す
ポート#1 (WRITE)	n + 1	コントロールレジスタにデータを書き込む
ポート#2 (READ)	n + 2	パレットレジスタにデータを書き込む
ポート#3 (WRITE)	n + 3	間接指定されたコントロールレジスタにデータを書き込む

注意：mの値はMAIN ROMの0006Hを参照して得る（MSXturboRでは98H）
nの値はMAIN ROMの0007Hを参照して得る（MSXturboRでは98H）

1.5 TMS9918A 互換モード

V9938・V9958はTMS9918Aとの上位互換性を持っています。互換性は、TMS9918A互換モードと呼ばれる画面モードを使用することで確保されます。TMS9918A互換モードに含まれる画面モードを表1-5に示します。

表1-5 TMS9918A互換モード

VDPモード	BASICのSCREENモード
TEXT1	SCREEN 0 (40字モード)
GRAPHIC1	SCREEN 1
GRAPHIC2	SCREEN 2
MULTI COLOR	SCREEN 3

1.6 画面構成

TMS9918A・V9938・V9958の画面は、いくつかの独立した面が重なって構成されています。その構成を次に示します。

画面手前

スプライト面0

スプライト面1

・

・

・

スプライト面31

パターン面

バックドロップ面（単色）

スーパーインポーズ面

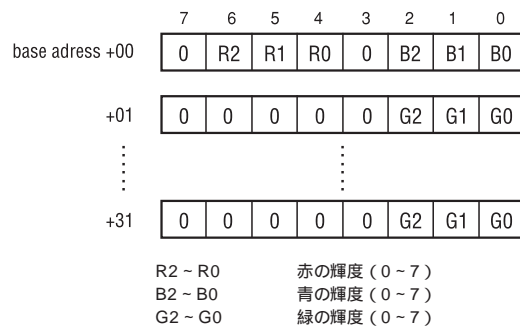
最後部面（常に黒）

画面奥

1.7 パレットテーブル

パレットレジスタ（P#0～P#15）は書き込み専用のため、システムによってその値を保持する領域（パレットテーブル）がVRAM上に確保されます。パレットテーブルは、1色2bytesの長さを持ち、全体で32bytesあります。

型式は次のようになっています。パレットテーブルのアドレスに関しては、「VRAMマップ」を参照してください。



2 基本入出力

VDPの操作に必要なコントロールレジスタ、ステータスレジスタ、パレットレジスタ、VRAMにアクセスする方法について説明します。

2.1 コントロールレジスタのアクセス

VDPのコントロールレジスタ (R#n) にデータをセットする方法には、TMS9918Aでは“直接指定”と呼ばれる方法が、V9938・V9958では直接指定に加え“間接指定”と呼ばれる方法が存在します。

2.1.1 直接指定

ポート#1 (WRITE) に、データ、レジスタ番号の順に出力する方法です。この順番は必ず守らなければならないので注意が必要です。例えば、割り込みルーチンがVDPをアクセスしている場合には、割り込みを禁止してから出力を行います。

この方法はTMS9918A・V9938・V9958で共通に使えます。



2.1.2 間接指定

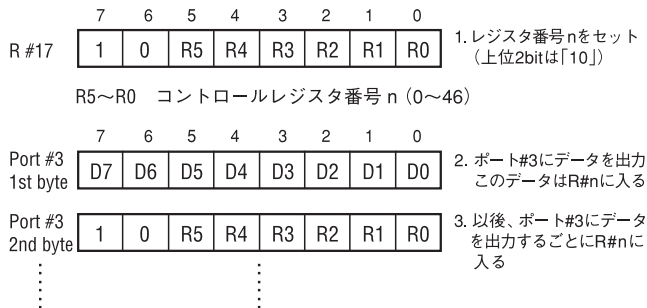
コントロールレジスタR#17 (Control register pointer) にレジスタ番号を指定する方法で、“非オートインクリメントモード”と“オートインクリメントモード”の2つのモードがあります。

この方法はTMS9918Aでは使用できません。

a) 非オートインクリメントモード

直接指定でR#17に「レジスタ番号 OR 80H」をセットし、ポート#3にデータを出力します。R#17の値は、直接指定でのみ変更可能です。

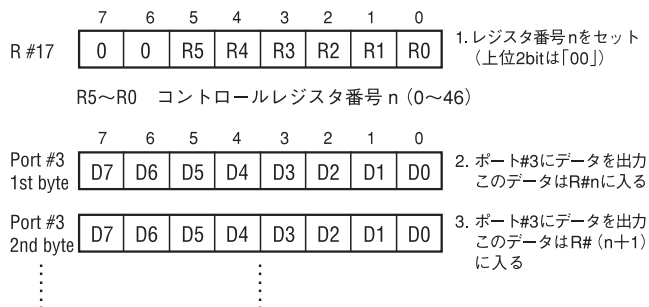
非オートインクリメントモードではR#17の値は変化しないので、データを出力した後に再び同一レジスタにデータをセットしたい場合、R#17の再セットを行う必要はありません。このモードはVDPコマンドの実行時等、同一レジスタに連続してデータを送る場合に使用します。



b) オートインクリメントモード

直接指定でR#17に「レジスタ番号」をセットし、ポート#3にデータを出力します。R#17の値は、直接指定でのみ変更可能です。

オートインクリメントモードでは、ポート#3にデータを出力するたびに、R#17の値がオートインクリメントされます。このモードはVDPコマンドのセット、スクリーンモードの変更等、連続した多数のレジスタを書き換える場合に使用します。

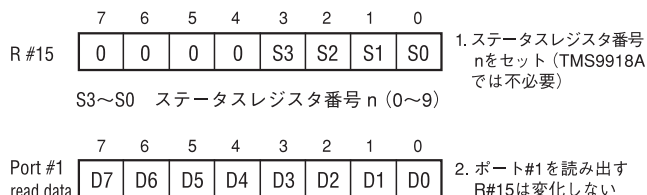


2.2 ステータスレジスタのアクセス

VDPのステータスレジスタ (S#n) を読み出すには、コントロールレジスタR#15 (Status register pointer) にステータスレジスタ番号をセットし、ポート#1 (READ) を読み出します。

MSXの割り込みルーチンは、R#15に「0」がセットされていることを前提としています。したがって、R#15を書き換える場合には、必ず割り込みを禁止し、目的の処理が終了した後、R#15に「0」をセットしてから割り込みを許可してください。

なお、TMS9918Aには、S#1 ~ S#9およびR#15は存在せず、ポート#1 (READ) を読み出す操作のみを行うことで、必ずS#0の値が返ります。

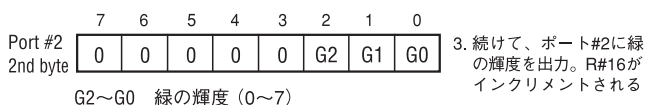


2.3 パレットレジスタのアクセス

V9938・V9958のパレットレジスタ (P#n) にデータをセットするためには、コントロールレジスタR#16 (Color palette address pointer) を使用します。まず、R#16にパレット番号をセットし、次にポート#2にデータを2bytes連続で出力します。R#16はポート#2にデータを2bytes書き込むとオートインクリメントされます。

なお、TMS9918Aにはパレット機能がないため、パレットレジスタは存在しません。





2.4 VRAM(VIDEO RAM)のアクセス

TMS9918Aには16KbytesのVRAMを、V9938・V9958には128KbytesのVRAMと64Kbytesの拡張RAMを接続できます。V9938・V9958の拡張RAMは、その内容を画面に表示することはできませんが、VDPコマンド実行時にはワークエリアとして、VRAMと同等に扱うことができます。

ただし、MSX規格にこの拡張RAMは入っていないので、R#45のbit6には常に「0」を設定してください。

メモリマップは次のようになっています。

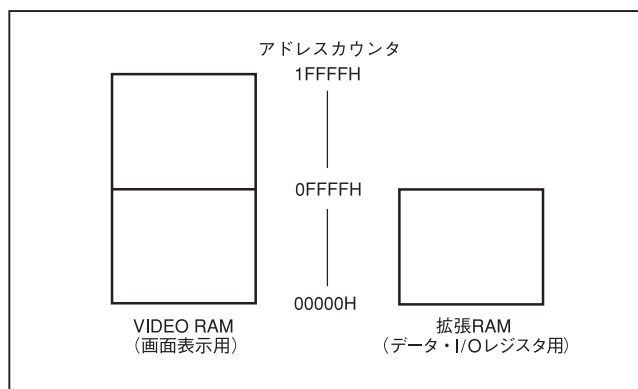


図2-1 VDPのメモリマップ

VRAMアクセスは次の手順で行います。

バンク切り替え (VRAM ~ 拡張RAM)

アドレスカウンタセット (A16 ~ A14)

アドレスカウンタセット (A7 ~ A0)

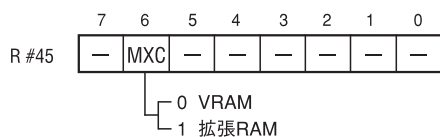
アドレスカウンタセット (A13 ~ A8) 読み出し・書き込み指定

データの読み出し・書き込み

なお、TMS9918AでVRAMをアクセスする場合は、手順 ~ のみを行います。

バンク切り替え (VRAM ~ 拡張RAM)

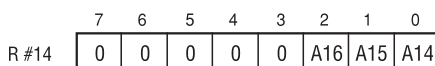
R#45のbit6にVRAMを使う場合は「0」、拡張RAMを使う場合は「1」をセットします。MSXでは常に「0」をセットしてください。



アドレスカウンタセット (A16 ~ A14)

コントロールレジスタR#14 (VRAM access base address register) を使って、アドレスカウンタの上位3bit (A16 ~ A14) をセットしま

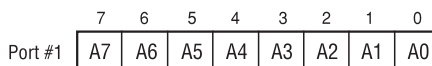
す。このレジスタは、TMS9918A互換モードのときには、オートインクリメントされません。



A16~A14 アドレスカウンタの上位3bit

アドレスカウンタセット (A7 ~ A0)

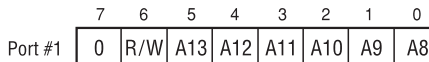
ポート#1に、アドレスカウンタの下部8bit (A7 ~ A0) にセットするデータを出力します。



A7~A0 アドレスカウンタの下部8bit

アドレスカウンタセット (A13 ~ A8) 読み出し・書き込み指定

ポート#1に、アドレスカウンタの中部6bit (A13 ~ A8) にセットするデータ、および読み出し・書き込みの指定を出力します。bit6が読み出し・書き込みのスイッチになっており、「0」を指定すると読み出し、「1」を指定すると書き込みになります。

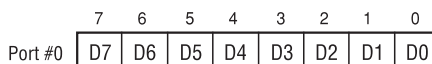


R/W 読み出し・書き込みのスイッチ 0=読み出し、1=書き込み
A13~A8 アドレスカウンタの中部6bit

データの読み出し・書き込み

ポート#0にアクセスすることで、VRAMの読み出し・書き込みができます。ポート#0にアクセスするたびにアドレスカウンタがオートインクリメントされるので、連続して読み出し・書き込みを行えます。

ただし、TMS9918A互換モードのときには、A16 ~ A14がオートインクリメントされません。



V9938・V9958でVRAMのアクセスを行うには、VDPコマンドを使用する方法もあります (詳細は「4. VDPコマンド」を参照)。

3 VDPレジスタの機能

VDPを制御するコントロールレジスタ、VDPの状態を保持するステータスレジスタについて説明します。

3.1 コントロールレジスタ R#0 ~ R#23、R#25 ~ R#27、R#32 ~ R#46 (Write only)

コントロールレジスタはVDPを制御する、書き込み専用のレジスタ群です。

3.1.1 モードレジスタ

VDPの各種動作モードを設定するレジスタ群です。

Mode register 0

	7	6	5	4	3	2	1	0
R#0	0	DG	IE2	IE1	M5	M4	M3	0

- DG カラーバスの設定
1 = カラーバスを入力モードにしてデータをVRAMに取り込む (デジタル機能を持ったMSX2・MSX2+・MSXturboRで使用可能) 0 = カラーバスを使用しない
- IE2 ライトペンによる割り込みを 1 = 使用する、0 = 使用しない
MSXでは、この機能は使用していないので常に「0」を指定する
- IE1 水平帰線による割り込みを 1 = 使用する、0 = 使用しない
- M5 ~ M3 画面モードの設定に使用

TMS9918Aでは、DG・IE2・IE1・M5・M4に「0」を設定してください。

Mode register 1

	7	6	5	4	3	2	1	0
R#1	4/16K	BL	IE0	M1	M2	0	SI	MAG

- 4/16K TMS9918Aで、VRAMの種類を選択する
1 = 8Kbitまたは16KbitのDRAMを使用、0 = 4KbitDRAMを使用
V9938・V9958では常に「0」をセットする
- BL 1 = 画面表示、0 = 画面非表示
- IE0 垂直帰線による割り込みを 1 = 使用する、0 = 使用しない
- M1 ~ M2 画面モードの設定に使用
- SI スプライトのサイズ 1 = 16 x 16ドット、0 = 8 x 8ドット
- MAG スプライトを 1 = 拡大する、0 = 拡大しない

M5 ~ M1の設定値と画面モードの対応は表3-1の通りです。

表3-1 画面モードとモードレジスタ

M5 ~ M1	VDPの画面モード	BASICのスクリーンモード
00001	TEXT1	SCREEN 0 (40字モード)
01001	TEXT2	SCREEN 0 (80字モード)
00000	GRAPHIC1	SCREEN 1
00100	GRAPHIC2	SCREEN 2
00010	MULTI COLOR	SCREEN 3
01000	GRAPHIC3	SCREEN 4
01100	GRAPHIC4	SCREEN 5
10000	GRAPHIC5	SCREEN 6
10100	GRAPHIC6	SCREEN 7
11100	GRAPHIC7	SCREEN 8
11100	YJK/RGB混在	SCREEN 10 (R#25と併せて設定)
11100	YJK/RGB混在	SCREEN 11 (R#25と併せて設定)
11100	YJK	SCREEN 12 (R#25と併せて設定)

Mode register 2

	7	6	5	4	3	2	1	0
R#8	MS	LP	TP	CB	VR	0	SPD	BW

- MS マウスを 1 = 使用する (カラーバスは入力モード) 0 = 使用しない (カラーバスは出力モード)
MSXでは、この機能は使用していないので常に「0」を指定する
- LP ライトペンを 1 = 使用する、0 = 使用しない
MSXでは、この機能は使用していないので常に「0」を指定する
- TP カラーコード0の色を 1 = カラーパレットの色にする、0 = 透明色として扱う
SCREEN 0・10・11では常にカラーパレットの色である
- CB カラーバスを 1 = 入力モードにする、0 = 出力モードにする
- VR VRAMの種類を選択する
1 = 64K x 1bitまたは64K x 4bit、0 = 16K x 1bitまたは16K x 4bit
システムは常に「1」をセットする

- SPD スプライトを 1 = 表示しない、0 = 表示する
- BW 1 = 白黒32階長、0 = カラー (Composit encoderにのみ有効)
MSXではこの機能は使用していない

Mode register 3

	7	6	5	4	3	2	1	0
R#9	LN	0	S1	S0	IL	EO	NT	DC

- LN 1 = 縦212ドット表示、0 = 縦192ドット表示
- S1 ~ S0 同期モード選択 (表3-2参照)
- IL 1 = Interlace (完全NTSCタイミング)、0 = Non Interlace (不完全NTSCタイミング)
- EO Even/Odd fieldで 1 = 2枚の絵を1/60秒毎に交互に表示 (PALの場合は1/50秒毎) 0 = 同じ絵を表示
- NT 1 = PAL (313line)、0 = NTSC (262line) RGB出力のみ有効
- DC DLCLK端子を 1 = 入力モードにする、0 = 出力モードにする
普通のアプリケーションプログラムが書き換えてはならない
システムは「0」をセットする

V9938・V9958では、同期モードをS1・S0の2bitで指定します。

表3-2 同期モード

S1	S0	同期モード	Ys	用途
0	0	パソコン同期	常にV9938・V9958を選択(0)	V9938・V9958の画面を表示
0	1	標準同期	表示画面の透明部分で発生	スーパーインポーズ・デジタル等
1	0	標準同期	常に外部信号を選択	外部信号の表示
1	1	予約	予約	予約

	7	6	5	4	3	2	1	0
R#25	0	CMD	VDS	YAE	YJK	WTE	MSK	SP2

- CMD コマンド機能を
0 = GRAPHIC4 ~ 7モードでのみ使用可能にする
1 = 全画面モードで使用可能にする (GRAPHIC4 ~ 7ではV9938と同等に機能する。それ以外の画面モードではGRAPHIC7として動作し、座標系はGRAPHIC7モードの(X, Y)座標系を用いる)
- VDS VDPの端子8を制御する (普通のアプリケーションプログラムが書き換えてはならない)
システムは「0」をセットする
- YAE YJK方式のデータフォーマットを選択
0 = YJK、1 = YJK/RGB混在
- YJK VRAM上のデータを
0 = RGB方式として扱う
1 = YJK方式と見なし、これをRGB信号 (各5bit) に変換してアナログ出力する (スプライトの表示色はパレットが有効)
- WTE ウェイト機能を
0 = 無効にする (TMS9918A・V9938と同等)
1 = 有効にする (CPUがVRAMをアクセスした際、V9958のVRAMアクセスが完了するまで、すべてのV9958ポートへのアクセスに対してウェイト信号を発生)
レジスタとパレットへのアクセス未完了およびコマンドのデータレディによるウェイト機能は存在しない
- MSK 画面左端8ドットを、0 = マスクしない、1 = マスクしバックドロップの色を表示する
GRAPHIC5・6では16ドットのマスク
- SP2 ハードウェア横スクロールを使用する時に用いる
ハードウェア横スクロール時、水平方向画面サイズを
0 = 1ページとする (スクロールは1ページ内で行われ、隠れた部分が反対側に表示される)
1 = 2ページとする (スクロールは2ページで行われ、スクロールによって裏のページが表示される)

ウェイト機能はMSXでは使用していません。WTEを「1」にするときは、その前にダミーのVRAMアクセスを行ってください。

YAE、YJKの各bitの詳細については、MSX-Datapack volume1

Color burst register 3

	7	6	5	4	3	2	1	0
R#22	0	0	0	0	0	1	0	1

これらのレジスタには、上記の値が起動時にセットされます。この値をすべて「0」にすると、コンポジットビデオ出力の色成分の信号を消すことができます。しかし、MSXのコンポジットビデオ出力は多くの場合、VDPのRGB出力から外部回路により合成されるため、このレジスタの機能は使用されません。

3.1.4 ディスプレイレジスタ

ディスプレイ上の表示位置を制御するレジスタ群です。

Interrupt line register

	7	6	5	4	3	2	1	0
R#19	IL7	IL6	IL5	IL4	IL3	IL2	IL1	IL0

IL7 ~ IL0 割り込みを発生させる走査線番号

V9938・V9958では、VDPが特定の走査線の出力を終えたときに、割り込みを発生させることができます。

割り込みを発生させるには、このレジスタに割り込みを発生させる走査線番号を設定し、R#0のbit4に「1」をセットします。割り込みが発生すると、S#1のbit0がセットされ、割り込みベクタ(0038H)が割り込みを禁止されてコールされます。通常は、0038Hからシステムの割り込みルーチンへ進みます(詳細は「6.3 水平帰線割り込み」を参照)。

Display adjust register

	7	6	5	4	3	2	1	0
R#18	V3	V2	V1	V0	H3	H2	H1	H0

V3 ~ V0 縦方向の表示位置 (-7 ~ +8)
H3 ~ H0 横方向の表示位置 (-7 ~ +8)

ディスプレイ上の表示位置を補正するためのレジスタです。設定値と表示位置の関係を図3-1に示します。

Display offset register

	7	6	5	4	3	2	1	0
R#23	D07	D06	D05	D04	D03	D02	D01	D00

D07 ~ D00 表示開始ライン

表示を開始するラインをセットするためのレジスタです。指定したラインのデータから順に表示されていきます。このレジスタの値を連続的に変えることによって、画面のハードウェア縦スクロールを行え

ます。ただし、スクロールは256ライン目で上下がつながる形で行われるため、スプライトテーブル等は別のページに置く必要があります。

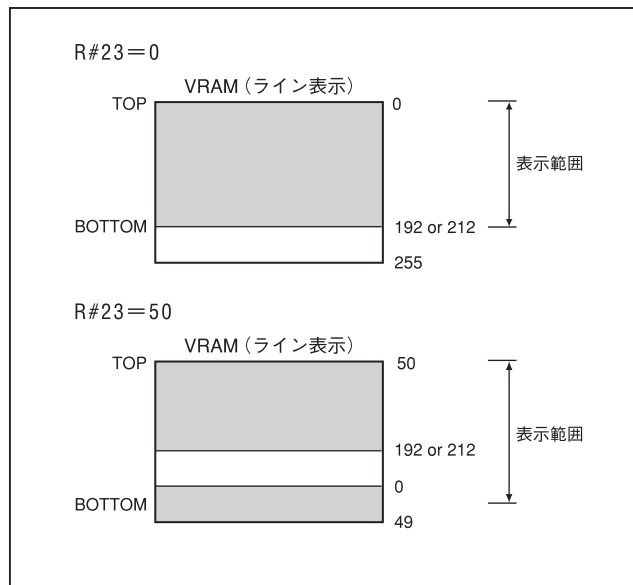


図3-2 オフセットレジスタの設定例

Horizontal scroll high

	7	6	5	4	3	2	1	0
R#26	0	0	H08	H07	H06	H05	H04	H03

Horizontal scroll low

	7	6	5	4	3	2	1	0
R#27	0	0	0	0	0	H02	H01	H00

H08 ~ H00 横スクロール量をドット単位で指定(GRAPHIC5・6は2ドット単位)
H08 ~ H03の値に関して、左方向へ8ドット単位で設定値だけシフト(H08は1ページスクロールの時は無効)
H02 ~ H00の値に関して、右方向へ1ドット単位で設定値だけシフト
ただし、TEXT1,TEXT2ではH02 ~ H00のみ有効

V9958によるハードウェア横スクロール時のVRAMアクセスは8ドット単位(GRAPHIC5・6は16ドット単位)で行われるため、R#27が「0」のとき以外は左端部の表示用データが不定になります。この不定のデータを表示させないために、MSK(R#25のbit1)によるマスクが必要になります。

SP2(R#25のbit0)に水平方向の画面サイズを設定できます。

縦方向の補正(V)	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8
								中央								下
SET ADJUST 命令の値	-7	-6	-5	-4	-3	-2	-1	0	+1	+2	+3	+4	+5	+6	+7	+8
横方向の補正(H)	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8
								中央								右
SET ADJUST 命令の値	-7	-6	-5	-4	-3	-2	-1	0	+1	+2	+3	+4	+5	+6	+7	+8

図3-1 表示位置の補正值

SP2=0で1画面の両端がつながってスクロールします。SP2=1では、パターンネームテーブルを奇数ページにセットすることで、偶数ページ、奇数ページがつながってスクロールします。

3.1.5 アクセスレジスタ

VDPのレジスタやVRAMをアクセスするとき使用するレジスタ群です（詳細は「2. 基本入出力」を参照）。

VRAM Access base address register

	7	6	5	4	3	2	1	0
R#14	0	0	0	0	0	A16	A15	A14

A16 ~ A14 アドレスカウンタの上位3bit

V9938・V9958のVRAMをアクセスするとき、アドレスの上位3bitをこのレジスタにセットします。

また、このレジスタの値は、VRAMをアクセスするとA13（アドレスカウンタのbit13）からの繰り上がりを受けて自動的にインクリメントされます。ただし、TMS9918A互換モードでは、このレジスタのインクリメントは行われません。

Status register pointer

	7	6	5	4	3	2	1	0
R#15	0	0	0	0	S3	S2	S1	S0

S3 ~ S0 ステータスレジスタ番号（0 ~ 9）

V9938・V9958のステータスレジスタ（S#0 ~ S#9）を読み出す際、このレジスタにステータスレジスタ番号（0 ~ 9）をセットします。

Color palette address register

	7	6	5	4	3	2	1	0
R#16	0	0	0	0	P3	P2	P1	P0

P3 ~ P0 パレット番号（0 ~ 15）

V9938・V9958のパレットレジスタ（P#0 ~ P#15）にアクセスする際、このレジスタにパレット番号（0 ~ 15）をセットします。

Control register pointer

	7	6	5	4	3	2	1	0
R#17	All	0	R5	R4	R3	R2	R1	R0

└─ 0 オートインクリメント
└─ 1 オートインクリメント禁止

R5 ~ R0 コントロールレジスタ番号（0 ~ 46）

V9938・V9958では、このレジスタをポインタとして他のレジスタをアクセスすることができます。また、All（bit7）の指定によって、内容を自動的にインクリメントさせることができます。

3.1.6 コマンドレジスタ

V9938・V9958のコマンドを実行するとき使用するレジスタ群です（詳細は「4. VDPコマンド」を参照）。

Source X low register

	7	6	5	4	3	2	1	0
R#32	SX7	SX6	SX5	SX4	SX3	SX2	SX1	SX0

Source X high register

	7	6	5	4	3	2	1	0
R#33	0	0	0	0	0	0	0	SX8

Source Y low register

	7	6	5	4	3	2	1	0
R#34	SY7	SY6	SY5	SY4	SY3	SY2	SY1	SY0

Source Y high register

	7	6	5	4	3	2	1	0
R#35	0	0	0	0	0	0	SY9	SY8

Destination X low register

	7	6	5	4	3	2	1	0
R#36	DX7	DX6	DX5	DX4	DX3	DX2	DX1	DX0

Destination X high register

	7	6	5	4	3	2	1	0
R#37	0	0	0	0	0	0	0	DX8

Destination Y low register

	7	6	5	4	3	2	1	0
R#38	DY7	DY6	DY5	DY4	DY3	DY2	DY1	DY0

Destination Y high register

	7	6	5	4	3	2	1	0
R#39	0	0	0	0	0	0	DY9	DY8

Number of dot X low register

	7	6	5	4	3	2	1	0
R#40	NX7	NX6	NX5	NX4	NX3	NX2	NX1	NX0

Number of dot X high register

	7	6	5	4	3	2	1	0
R#41	0	0	0	0	0	0	0	NX8

Number of dot Y low register

	7	6	5	4	3	2	1	0
R#42	NY7	NY6	NY5	NY4	NY3	NY2	NY1	NY0

Number of dot Y high register

	7	6	5	4	3	2	1	0
R#43	0	0	0	0	0	0	NY9	NY8

Color register

	7	6	5	4	3	2	1	0
R#44	CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0

Argument register

	7	6	5	4	3	2	1	0
R#45	0	MXC	MXD	MXS	DIY	DIX	EQ	MAJ

Command register

	7	6	5	4	3	2	1	0
R#46	CM3	CM2	CM1	CM0	LO3	LO2	LO1	LO0

3.2 ステータスレジスタ S#0 ~ S#9 (Read only)

VDPの状態を得るための読み出し専用のレジスタ群です。

Status register 0

	7	6	5	4	3	2	1	0
S#0	F	5S	C	5th sprite#				

- F 垂直帰線割り込みフラグ
垂直帰線割り込み(R#0のbit5がセットされているとき)が発生すると「1」になる。S#0を読み出すと「0」になる
- 5S 第5スプライトフラグ
Fが「0」で、かつ1水平線上にスプライトが5個(スプライトモード2では9個)並ぶと「1」になり、同時に5th sprite#がセットされる
- C 衝突フラグ
スプライトが衝突すると「1」になる。S#0を読み出すと「0」になる。
- 5th sprite# 1水平線上にスプライトが5個(スプライトモード2では9個)並んだとき、第5(第9)スプライトの番号がセットされる

Status register 1

	7	6	5	4	3	2	1	0
S#1	FL	LPS	ID#				FH	

- FL ライトペンスイッチ(R#8のbit6が「1」のとき)
ライトペンが光を検出すると「1」になる。このとき、IE2(R#0のbit5)が「1」であれば割り込みが発生する。S#1を読み出すと「0」になる
- LPS ライトペンスイッチ(R#8のbit6が「1」のとき)
ライトペンのスイッチが押されると「1」になる。S#1を読み出しても「0」にならない
- ID# VDPのID番号
V9938では、00000
V9958では、00010
- FH 水平帰線割り込みフラグ
水平帰線(R#19で指定)による割り込み(R#0のbit4が「1」のとき)が発生すると「1」になる。S#1を読み出すと「0」になる

FL(bit7)とLPS(bit6)はマウス・ライトペン用のレジスタです。MSXでは、V9938のマウス・ライトペンインターフェイスは使用していないので、FL・LPSは意味を持ちません。また、V9958ではこれらの機能は削除されています。

Status register 2

	7	6	5	4	3	2	1	0
S#2	TR	VR	HR	BD	1	1	EO	CE

- TR 転送レディフラグ
CPU to VRAM、VRAM to CPU等のコマンドを実行するときは、CPUはこのフラグを見ながらデータの転送を行う。このbitが「1」のとき、転送可能
- VR 垂直帰線期間(画面を書き終わって次に書き始めるまでの期間)フラグ
垂直帰線期間中は「1」になる
- HR 水平帰線期間(走査線を書き終わって次に書き始めるまでの期間)フラグ
水平帰線期間中は「1」になる
- BD 境界色発見フラグ
サーチコマンドの実行で、境界色または非境界色を発見したとき「1」になる
- EO 表示フィールドフラグ(インターレースモード・2画面交互表示のときに意味を持つ)
現在の表示フィールドが 0 = 第1フィールド、1 = 第2フィールド
- CE コマンド実行フラグ
コマンド実行中は「1」になる

Column register low

	7	6	5	4	3	2	1	0
S#3	X7	X6	X5	X4	X3	X2	X1	X0

Column register high

	7	6	5	4	3	2	1	0
S#4	1	1	1	1	1	1	1	X8

Row register low

	7	6	5	4	3	2	1	0
S#5	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0

Row register high

	7	6	5	4	3	2	1	0
S#6	1	1	1	1	1	1	EO	Y8

これらのレジスタには、スプライトの衝突座標がセットされます(詳細は「5.3.6 スプライトの衝突」を参照)。

Color register

	7	6	5	4	3	2	1	0
S#7	C7	C6	C5	C4	C3	C2	C1	C0

POINT、VRAM to CPU等のコマンドを実行すると、読み込まれたVRAMのデータがこのレジスタにセットされます。

Border X register low

	7	6	5	4	3	2	1	0
S#8	BX7	BX6	BX5	BX4	BX3	BX2	BX1	BX0

Border X register high

	7	6	5	4	3	2	1	0
S#9	1	1	1	1	1	1	1	BX8

サーチコマンドで発見した境界色または非境界色のX座標がこのレジスタにセットされます。

4 VDPコマンド

V9938・V9958のVDPコマンドについて説明します。

4.1 V9938・V9958のコマンド

VRAMにアクセスする方法の1つに、V9938・V9958に備わっているコマンドを用いるというものがあります。このコマンドを“VDPコマンド”と呼び、描画(LINE、PSET等)やVRAMの部分転送が手軽に行えます。

VDPコマンドは、V9938ではGRAPHIC4～7・YJK/RGB混在・YJK (SCREEN 5～8・10～12)でのみ使用可能です。V9958では、R#25のbit6をセットすると、すべての画面モードでVDPコマンドの使用が可能になります。このときのパラメータは、GRAPHIC7 (SCREEN 8)の座標系で指定します。

表4-1にVDPコマンドの一覧を示します。

表4-1 V9938・V9958のコマンド一覧

コマンド名	コマンドの内容	転送単位	論理演算	R#46のbit7～4
HMMC	CPU VRAM高速転送	byte	無効	1111
YMMM	Y方向のVRAM間高速転送	byte	無効	1110
HMMM	VRAM間高速転送	byte	無効	1101
HMMV	矩形領域の高速塗り潰し	byte	無効	1100
LMMC	CPU VRAM論理転送	dot	有効	1011
LMCM	VRAM CPU論理転送	dot	無効	1010
LMMM	VRAM間論理転送	dot	有効	1001
LMMV	矩形領域の論理塗り潰し	dot	有効	1000
LINE	直線の描画	dot	有効	0111
SRCH	カラーコードのサーチ	dot	無効	0110
PSET	点の描画	dot	有効	0101
POINT	カラーコードの読み出し	dot	無効	0100
STOP	VDPコマンドの中断	-	-	0000

4.2 VDPコマンドにおける座標系

VDPコマンドでの位置パラメータは、(X, Y)座標で指定します。このときの座標系は、VRAMの全領域をカバーし、ページに関わらず連続的にアクセスできます。

表4-2にVDPコマンドの座標系と各画面モードとの関係を示します。

表4-2 VDPコマンドの座標系と画面モード

画面モード	X座標の範囲	Y座標の範囲
GRAPHIC4	0～255	0～1023
GRAPHIC5	0～511	0～1023
GRAPHIC6	0～511	0～511
GRAPHIC7	0～255	0～511
YJK/RGB混在	0～255	0～511
YJK	0～255	0～511

4.3 ロジカルオペレーション

一部のコマンドでは、転送元のデータと転送先のデータとで論理演算を行うことができます。論理演算(ロジカルオペレーション)は、コマンドを実行するときにR#46のbit3～0に書き込みます。

表4-3に論理演算の一覧を示します。

表4-3 ロジカルオペレーションコード一覧

演算名	BASICでの演算名	演算動作(書き込まれる値)	R#46のbit3～0
IMP	PSET	SC	0000
AND	AND	SC AND DC	0001
OR	OR	SC OR DC	0010
EOR	XOR	SC XOR DC	0011
NOT	PRESET	NOT SC	0100
TIMP	TPSET	if SC=0 then DC else SC	1000
TAND	TAND	if SC=0 then DC else SC AND DC	1001
TOR	TOR	if SC=0 then DC else SC OR DC	1010
TEOR	TXOR	if SC=0 then DC else SC XOR DC	1011
TNOT	TPRESET	if SC=0 then DC else NOT SC	1100

SC=Source Color code : 転送元のカラーコード

DC=Destinaton Color code : 転送先のカラーコード

4.4 各コマンドの説明

各コマンドの内容と実行方法を説明します。

4.4.1 HMMC (CPU VRAM高速転送)

CPUからVRAM上の矩形領域へデータを転送するコマンドです。データの転送は1byte単位で行われるため、X座標として指定できる値は、画面モードによって制限を受けます。

以下に実行手順を説明します。

レジスタのセット

以下のレジスタをセットします。

Destination X register

	7	6	5	4	3	2	1	0
R#36	DX7	DX6	DX5	DX4	DX3	DX2	DX1	DX0

R#37	0	0	0	0	0	0	0	DX8
------	---	---	---	---	---	---	---	-----

DX8～DX0 転送先基準点のX座標(0～511)
GRAPHIC4・6ではDX0が、GRAPHIC5ではDX1～DX0が無視される

Destination Y register

	7	6	5	4	3	2	1	0
R#38	DY7	DY6	DY5	DY4	DY3	DY2	DY1	DY0

R#39	0	0	0	0	0	0	DY9	DY8
------	---	---	---	---	---	---	-----	-----

DY9 ~ DY0 転送先基準点のY座標 (0 ~ 1023)

Number of dot X register

	7	6	5	4	3	2	1	0
R#40	NX7	NX6	NX5	NX4	NX3	NX2	NX1	NX0

R#41	0	0	0	0	0	0	0	NX8
------	---	---	---	---	---	---	---	-----

NX8 ~ NX0 X方向転送ドット数 (1 ~ 512)
512を指定するときは、NX8 ~ NX0に「0」をセットする
GRAPHIC4・6ではNX0が、GRAPHIC5ではNX1 ~ NX0
が無視される

Number of dot Y register

	7	6	5	4	3	2	1	0
R#42	NY7	NY6	NY5	NY4	NY3	NY2	NY1	NY0

R#43	0	0	0	0	0	0	NY9	NY8
------	---	---	---	---	---	---	-----	-----

NY9 ~ NY0 Y方向転送ドット数 (1 ~ 1024)
1024を指定するときは、NY9 ~ NY0に「0」をセットする

Color register

	7	6	5	4	3	2	1	0
R#44	CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0

CR7 ~ CR0 転送データの1byte目

Argument register

	7	6	5	4	3	2	1	0
R#45	0	0	MXD	0	DIY	DIX	0	0

MXD 転送先のメモリ選択 0 = VRAM、1 = 拡張RAM (MSXでは常に「0」を指定)
DIX 転送先基準点からの転送方向 (X軸方向) 0 = 右、1 = 左
DIY 転送先基準点からの転送方向 (Y軸方向) 0 = 下、1 = 上

コマンドの実行

R#46に、11110000Bを書き込むとコマンドが実行されます。

Command register

	7	6	5	4	3	2	1	0
R#46	1	1	1	1	0	0	0	0

コマンド終了の確認

S#2を読んで、CE (S#2のbit0) が「0」であれば、コマンドの終了です。

2bytes目以降の転送

S#2を読んで、TR (S#2のbit7) が「1」であれば、データ1byteをR#44に書き込みます。TRが「0」のときは、「1」になるまで待ちます。

以下、手順 ~ を繰り返します。

4.4.2 YMMM (Y方向のVRAM間高速転送)

VRAM上の転送元基準点・Y方向転送ドット数・転送方向と画面の右端(左端)で指定される矩形領域をY方向に転送するコマンドです。データの転送は1byte単位で行われるため、X座標として指定できる値は、画面モードによって制限を受けます。

以下に、実行手順を説明します。

レジスタのセット

以下のレジスタをセットします。

Source Y register

	7	6	5	4	3	2	1	0
R#34	SY7	SY6	SY5	SY4	SY3	SY2	SY1	SY0

R#35	1	1	1	1	0	0	SY8	SY9
------	---	---	---	---	---	---	-----	-----

SY9 ~ SY0 転送元基準点のY座標 (0 ~ 1023)

Destination X register

	7	6	5	4	3	2	1	0
R#36	DX7	DX6	DX5	DX4	DX3	DX2	DX1	DX0

R#37	0	0	0	0	0	0	0	DX8
------	---	---	---	---	---	---	---	-----

DX8 ~ DX0 転送元および転送先基準点のX座標 (0 ~ 511)
GRAPHIC4・6ではDX0が、GRAPHIC5ではDX1 ~ DX0
が無視される

Destination Y register

	7	6	5	4	3	2	1	0
R#38	DY7	DY6	DY5	DY4	DY3	DY2	DY1	DY0

R#39	0	0	0	0	0	0	DY9	DY8
------	---	---	---	---	---	---	-----	-----

DY9 ~ DY0 転送先基準点のY座標 (0 ~ 1023)

Number of dot Y register

	7	6	5	4	3	2	1	0
R#42	NY7	NY6	NY5	NY4	NY3	NY2	NY1	NY0

R#43	0	0	0	0	0	0	NY9	NY8
------	---	---	---	---	---	---	-----	-----

NY9 ~ NY0 Y方向転送ドット数 (1 ~ 1024)
1024を指定するときは、NY9 ~ NY0に「0」をセットする

Argument register

	7	6	5	4	3	2	1	0
R#45	0	0	MXD	0	DIY	DIX	0	0

MXD 転送先のメモリ選択 0 = VRAM、1 = 拡張RAM (MSXでは常に「0」を指定)
DIX 転送元基準点からどちらの画面端まで転送するのを指定
0 = 右端、1 = 左端
DIY 転送元基準点からの転送方向 (Y軸方向) 0 = 下、1 = 上

コマンドの実行

R#46に、11100000Bを書き込むとコマンドが実行されます。

Command register

	7	6	5	4	3	2	1	0
R#46	1	1	1	0	0	0	0	0

コマンド終了の確認

コマンド実行中はCE (S#2のbit0) が「1」になります。CEが「0」になれば、コマンドの終了です。

4.4.3 HMMM (VRAM間高速転送)

VRAMからVRAMへ矩形領域のデータを転送するコマンドです。データの転送は1byte単位で行われるため、X座標として指定できる値は、画面モードによって制限を受けます。

以下に、実行手順を説明します。

レジスタのセット

以下のレジスタをセットします。

Source X register

	7	6	5	4	3	2	1	0
R#32	SX7	SX6	SX5	SX4	SX3	SX2	SX1	SX0

R#33	0	0	0	0	0	0	0	SX8
------	---	---	---	---	---	---	---	-----

SX8 ~ SX0 転送元基準点のX座標 (0 ~ 511)
GRAPHIC4・6ではSX0が、GRAPHIC5ではSX1 ~ SX0が無視される

Source Y register

	7	6	5	4	3	2	1	0
R#34	SY7	SY6	SY5	SY4	SY3	SY2	SY1	SY0

R#35	0	0	0	0	0	0	SY9	SY8
------	---	---	---	---	---	---	-----	-----

SY9 ~ SY0 転送元基準点のY座標 (0 ~ 1023)

Destination X register

	7	6	5	4	3	2	1	0
R#36	DX7	DX6	DX5	DX4	DX3	DX2	DX1	DX0

R#37	0	0	0	0	0	0	0	DX8
------	---	---	---	---	---	---	---	-----

DX8 ~ DX0 転送先基準点のX座標 (0 ~ 511)
GRAPHIC4・6ではDX0が、GRAPHIC5ではDX1 ~ DX0が無視される

Destination Y register

	7	6	5	4	3	2	1	0
R#38	DY7	DY6	DY5	DY4	DY3	DY2	DY1	DY0

R#39	0	0	0	0	0	0	DY9	DY8
------	---	---	---	---	---	---	-----	-----

DY9 ~ DY0 転送先基準点のY座標 (0 ~ 1023)

Number of dot X register

	7	6	5	4	3	2	1	0
R#40	NX7	NX6	NX5	NX4	NX3	NX2	NX1	NX0

R#41	0	0	0	0	0	0	0	NX8
------	---	---	---	---	---	---	---	-----

NX8 ~ NX0 X方向転送ドット数 (1 ~ 512)
512を指定するときは、NX8 ~ NX0に「0」をセットする
GRAPHIC4・6ではNX0が、GRAPHIC5ではNX1 ~ NX0が無視される

Number of dot Y register

	7	6	5	4	3	2	1	0
R#42	NY7	NY6	NY5	NY4	NY3	NY2	NY1	NY0

R#43	0	0	0	0	0	0	NY9	NY8
------	---	---	---	---	---	---	-----	-----

NY9 ~ NY0 Y方向転送ドット数 (1 ~ 1024)
1024を指定するときは、NY9 ~ NY0に「0」をセットする

Argument register

	7	6	5	4	3	2	1	0
R#45	0	0	MXD	MXS	DIY	DIX	0	0

MXD 転送先のメモリ選択 0 = VRAM、1 = 拡張RAM (MSXでは常に「0」を指定)

MXS 転送元のメモリ選択 0 = VRAM、1 = 拡張RAM (MSXでは常に「0」を指定)

DIX 転送基準点からの転送方向 (X軸方向) 0 = 右、1 = 左

DIY 転送基準点からの転送方向 (Y軸方向) 0 = 下、1 = 上

コマンドの実行

R#46に、11010000Bを書き込むとコマンドが実行されます。

Command register

	7	6	5	4	3	2	1	0
R#46	1	1	0	1	0	0	0	0

コマンド終了の確認

コマンド実行中はCE (S#2のbit0) が「1」になります。CEが「0」になれば、コマンドの終了です。

4.4.4 HMMV (矩形領域の高速塗りつぶし)

VRAM上の矩形領域を塗りつぶすコマンドです。データの転送は1byte単位で行われるため、X座標として指定できる値は、画面モードによって制限を受けます。以下に、実行手順を説明します。

レジスタのセット

以下のレジスタをセットします。

Destination X register

	7	6	5	4	3	2	1	0
R#36	DX7	DX6	DX5	DX4	DX3	DX2	DX1	DX0

R#37	0	0	0	0	0	0	0	DX8
------	---	---	---	---	---	---	---	-----

DX8 ~ DX0 転送先基準点のX座標 (0 ~ 511)
GRAPHIC4・6ではDX0が、GRAPHIC5ではDX1 ~ DX0が無視される

Destination Y register

	7	6	5	4	3	2	1	0
R#38	DY7	DY6	DY5	DY4	DY3	DY2	DY1	DY0

R#39	0	0	0	0	0	0	DY9	DY8
------	---	---	---	---	---	---	-----	-----

DY9 ~ DY0 転送先基準点のY座標 (0 ~ 1023)

Number of dot X register

	7	6	5	4	3	2	1	0
R#40	NX7	NX6	NX5	NX4	NX3	NX2	NX1	NX0
R#41	0	0	0	0	0	0	0	NX8

NX8 ~ NX0 X方向転送ドット数 (1 ~ 512)
512を指定するときは、NX8 ~ NX0に「0」をセットする
GRAPHIC4・6ではNX0が、GRAPHIC5ではNX1 ~ NX0
が無視される

Number of dot Y register

	7	6	5	4	3	2	1	0
R#42	NY7	NY6	NY5	NY4	NY3	NY2	NY1	NY0
R#43	0	0	0	0	0	0	NY9	NY8

NY9 ~ NY0 Y方向転送ドット数 (1 ~ 1024)
1024を指定するときは、NY9 ~ NY0に「0」をセットする

Color register

	7	6	5	4	3	2	1	0
R#44	CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0

CR7 ~ CR0 塗りつぶしデータ (1byte)

Argument register

	7	6	5	4	3	2	1	0
R#45	0	0	MXD	0	DIY	DIX	0	0

MXD 転送先のメモリ選択 0 = VRAM、1 = 拡張RAM (MSXでは常に「0」を指定)
DIX 転送先基準点からの転送方向 (X軸方向) 0 = 右、1 = 左
DIY 転送先基準点からの転送方向 (Y軸方向) 0 = 下、1 = 上

コマンドの実行

R#46に、11000000Bを書き込むとコマンドが実行されます。

Command register

	7	6	5	4	3	2	1	0
R#46	1	1	0	0	0	0	0	0

コマンド終了の確認

コマンド実行中はCE (S#2のbit0) が「1」になります。CEが「0」になれば、コマンドの終了です。

4.4.5 LMMC (CPU VRAM論理転送)

CPUからVRAM上の矩形領域へデータを転送するコマンドです。データの転送はドット単位で行われ、転送先のデータと論理演算できます。以下に、実行手順を説明します。

レジスタのセット

以下のレジスタをセットします。

Destination X register

	7	6	5	4	3	2	1	0
R#36	DX7	DX6	DX5	DX4	DX3	DX2	DX1	DX0
R#37	0	0	0	0	0	0	0	DX8

DX8 ~ DX0 転送先基準点のX座標 (0 ~ 511)

Destination Y register

	7	6	5	4	3	2	1	0
R#38	DY7	DY6	DY5	DY4	DY3	DY2	DY1	DY0
R#39	0	0	0	0	0	0	DY9	DY8

DY9 ~ DY0 転送先基準点のY座標 (0 ~ 1023)

Number of dot X register

	7	6	5	4	3	2	1	0
R#40	NX7	NX6	NX5	NX4	NX3	NX2	NX1	NX0
R#41	0	0	0	0	0	0	0	NX8

NX8 ~ NX0 X方向転送ドット数 (1 ~ 512)
512を指定するときは、NX8 ~ NX0に「0」をセットする

Number of dot Y register

	7	6	5	4	3	2	1	0
R#42	NY7	NY6	NY5	NY4	NY3	NY2	NY1	NY0
R#43	0	0	0	0	0	0	NY9	NY8

NY9 ~ NY0 Y方向転送ドット数 (1 ~ 1024)
1024を指定するときは、NY9 ~ NY0に「0」をセットする

Color register

	7	6	5	4	3	2	1	0
R#44	0	0	0	0	CR3	CR2	CR1	CR0
	0	0	0	0	0	0	CR1	CR0
	CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0

GRAPHIC4・6

GRAPHIC5

GRAPHIC7・YJK/
RGB混在・YJK

CR7 ~ CR0 転送データの1byte目

Argument register

	7	6	5	4	3	2	1	0
R#45	0	0	MXD	0	DIY	DIX	0	0

MXD 転送先のメモリ選択 0 = VRAM、1 = 拡張RAM (MSXでは常に「0」を指定)

DIX 転送先基準点からの転送方向 (X軸方向) 0 = 右、1 = 左
DIY 転送先基準点からの転送方向 (Y軸方向) 0 = 下、1 = 上

コマンドの実行

R#46の上位4bitに1011B、下位4bitにロジカルオペレーションコード (表4-3参照) を書き込むとコマンドが実行されます。

Command register

	7	6	5	4	3	2	1	0
R#46	1	0	1	1	LO3	LO2	LO1	LO0

LO3 ~ LO0 ロジカルオペレーションコード

コマンド終了の確認

S#2を読んで、CE (S#2のbit0) が「0」であれば、コマンドの終了です。

2 bytes目以降の転送

S#2を読んで、TR (S#2のbit7) が「1」であれば、データ1byteをR#44に書き込みます。TRが「0」のときは、「1」になるまで待ちます。

以下、手順 ~ を繰り返します。

4.4.6 LMCM (VRAM CPU論理転送)

VRAM上の矩形領域のデータをCPUへ転送するコマンドです。データの転送はドット単位で行われます。

以下に、実行手順を説明します。

レジスタのセット

以下のレジスタをセットし、さらにS#7を読んでTR (S#2のbit7)を「0」にします。

Source X register

	7	6	5	4	3	2	1	0
R#32	SX7	SX6	SX5	SX4	SX3	SX2	SX1	SX0

R#33	0	0	0	0	0	0	0	SX8
------	---	---	---	---	---	---	---	-----

SX8 ~ SX0 転送元基準点のX座標 (0 ~ 511)

Source Y register

	7	6	5	4	3	2	1	0
R#34	SY7	SY6	SY5	SY4	SY3	SY2	SY1	SY0

R#35	0	0	0	0	0	0	SY9	SY8
------	---	---	---	---	---	---	-----	-----

SY9 ~ SY0 転送元基準点のY座標 (0 ~ 1023)

Number of dot X register

	7	6	5	4	3	2	1	0
R#40	NX7	NX6	NX5	NX4	NX3	NX2	NX1	NX0

R#41	0	0	0	0	0	0	0	NX8
------	---	---	---	---	---	---	---	-----

NX8 ~ NX0 X方向転送ドット数 (1 ~ 512)
512を指定するときは、NX8 ~ NX0に「0」をセットする

Number of dot Y register

	7	6	5	4	3	2	1	0
R#42	NY7	NY6	NY5	NY4	NY3	NY2	NY1	NY0

R#43	0	0	0	0	0	0	NY9	NY8
------	---	---	---	---	---	---	-----	-----

NY9 ~ NY0 Y方向転送ドット数 (1 ~ 1024)
1024を指定するときは、NY9 ~ NY0に「0」をセットする

Argument register

	7	6	5	4	3	2	1	0
R#45	0	0	0	MXS	DIY	DIX	0	0

MXS 転送元のメモリ選択 0 = VRAM、1 = 拡張RAM (MSXでは常に「0」を指定)

DIX 転送元基準点からの転送方向 (X軸方向) 0 = 右、1 = 左

DIY 転送元基準点からの転送方向 (Y軸方向) 0 = 下、1 = 上

コマンドの実行

R#46に、10100000Bを書き込むとコマンドが実行されます。

Command register

	7	6	5	4	3	2	1	0
R#46	1	0	1	0	0	0	0	0

データの読み込み

S#2を読んで、TR (S#2のbit7)が「1」であれば、S#7を読み込みます。TRが「0」のときは、「1」になるまで待ちます。

Color register

	7	6	5	4	3	2	1	0
S#7	0	0	0	0	CR3	CR2	CR1	CR0

GRAPHIC4・6

	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	CR1	CR0

GRAPHIC5

	7	6	5	4	3	2	1	0
	CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0

GRAPHIC7・YJK/
RGB混在・YJK

CR7 ~ CR0 データ1ドット分がセットされる

コマンド終了の確認

S#2を読んで、CE (S#2のbit0)が「0」であれば、コマンドの終了です。以下、手順 ~ を繰り返します。

4.4.7 LMMM (VRAM間論理転送)

VRAMからVRAMへ矩形領域のデータを転送するコマンドです。データの転送はドット単位で行われ、転送先のデータと論理演算できます。以下に、実行手順を説明します。

レジスタのセット

以下のレジスタをセットします。

Source X register

	7	6	5	4	3	2	1	0
R#32	SX7	SX6	SX5	SX4	SX3	SX2	SX1	SX0

R#33	0	0	0	0	0	0	0	SX8
------	---	---	---	---	---	---	---	-----

SX8 ~ SX0 転送元基準点のX座標 (0 ~ 511)

Source Y register

	7	6	5	4	3	2	1	0
R#34	SY7	SY6	SY5	SY4	SY3	SY2	SY1	SY0

R#35	0	0	0	0	0	0	SY9	SY8
------	---	---	---	---	---	---	-----	-----

SY9 ~ SY0 転送元基準点のY座標 (0 ~ 1023)

Destination X register

	7	6	5	4	3	2	1	0
R#36	DX7	DX6	DX5	DX4	DX3	DX2	DX1	DX0

R#37	0	0	0	0	0	0	0	DX8
------	---	---	---	---	---	---	---	-----

DX8 ~ DX0 転送先基準点のX座標 (0 ~ 511)

Destination Y register

	7	6	5	4	3	2	1	0
R#38	DY7	DY6	DY5	DY4	DY3	DY2	DY1	DY0

R#39	0	0	0	0	0	0	DY9	DY8
------	---	---	---	---	---	---	-----	-----

DY9 ~ DY0 転送先基準点のY座標 (0 ~ 1023)

Number of dot X register

	7	6	5	4	3	2	1	0
R#40	NX7	NX6	NX5	NX4	NX3	NX2	NX1	NX0
R#41	0	0	0	0	0	0	0	NX8

NX8 ~ NX0 X方向転送ドット数 (1 ~ 512)
512を指定するときは、NX8 ~ NX0に「0」をセットする

Number of dot Y register

	7	6	5	4	3	2	1	0
R#42	NY7	NY6	NY5	NY4	NY3	NY2	NY1	NY0
R#43	0	0	0	0	0	0	NY9	NY8

NY9 ~ NY0 Y方向転送ドット数 (1 ~ 1024)
1024を指定するときは、NY9 ~ NY0に「0」をセットする

Argument register

	7	6	5	4	3	2	1	0
R#45	0	0	MXD	MXS	DIY	DIX	0	0

MXD 転送先のメモリ選択 0 = VRAM、1 = 拡張RAM (MSXでは常に「0」を指定)

MXS 転送元のメモリ選択 0 = VRAM、1 = 拡張RAM (MSXでは常に「0」を指定)

DIX 転送基準点からの転送方向 (X軸方向) 0 = 右、1 = 左
DIY 転送基準点からの転送方向 (Y軸方向) 0 = 下、1 = 上

コマンドの実行

R#46の上位4bitに1001B、下位4bitにロジカルオペレーションコード (表4-3参照) を書き込むとコマンドが実行されます。

Command register

	7	6	5	4	3	2	1	0
R#46	1	0	0	1	LO3	LO2	LO1	LO0

LO3 ~ LO0 ロジカルオペレーションコード

コマンド終了の確認

コマンド実行中はCE (S#2のbit0) が「1」になります。CEが「0」になれば、コマンドの終了です。

4.4.8 LMMV (矩形領域の論理塗りつぶし)

VRAM上の矩形領域を塗りつぶすコマンドです。データの転送はドット単位で行われ、転送先のデータと論理演算できます。

以下に、実行手順を説明します。

レジスタのセット

以下のレジスタをセットします。

Destination X register

	7	6	5	4	3	2	1	0
R#36	DX7	DX6	DX5	DX4	DX3	DX2	DX1	DX0
R#37	0	0	0	0	0	0	0	DX8

DX8 ~ DX0 転送先基準点のX座標 (0 ~ 511)

Destination Y register

	7	6	5	4	3	2	1	0
R#38	DY7	DY6	DY5	DY4	DY3	DY2	DY1	DY0
R#39	0	0	0	0	0	0	DY9	DY8

DY9 ~ DY0 転送先基準点のY座標 (0 ~ 1023)

Number of dot X register

	7	6	5	4	3	2	1	0
R#40	NX7	NX6	NX5	NX4	NX3	NX2	NX1	NX0
R#41	0	0	0	0	0	0	0	NX8

NX8 ~ NX0 X方向転送ドット数 (1 ~ 512)
512を指定するときは、NX8 ~ NX0に「0」をセットする

Number of dot Y register

	7	6	5	4	3	2	1	0
R#42	NY7	NY6	NY5	NY4	NY3	NY2	NY1	NY0
R#43	0	0	0	0	0	0	NY9	NY8

NY9 ~ NY0 Y方向転送ドット数 (1 ~ 1024)
1024を指定するときは、NY9 ~ NY0に「0」をセットする

Color register

	7	6	5	4	3	2	1	0
R#44	0	0	0	0	CR3	CR2	CR1	CR0
	0	0	0	0	0	0	CR1	CR0
	CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0

CR7 ~ CR0 塗りつぶしデータ (1ドット)

Argument register

	7	6	5	4	3	2	1	0
R#45	0	0	MXD	0	DIY	DIX	0	0

MXD 転送先のメモリ選択 0 = VRAM、1 = 拡張RAM (MSXでは常に「0」を指定)

DIX 転送先基準点からの転送方向 (X軸方向) 0 = 右、1 = 左
DIY 転送先基準点からの転送方向 (Y軸方向) 0 = 下、1 = 上

コマンドの実行

R#46の上位4bitに1000B、下位4bitにロジカルオペレーションコード (表4-3参照) を書き込むとコマンドが実行されます。

Command register

	7	6	5	4	3	2	1	0
R#46	1	0	0	0	LO3	LO2	LO1	LO0

LO3 ~ LO0 ロジカルオペレーションコード

コマンド終了の確認

コマンド実行中はCE (S#2のbit0) が「1」になります。CEが「0」になれば、コマンドの終了です。

4.4.9 LINE (直線の描画)

基準点と長辺・短辺で指定される長方形の対角線を描画するコマンドです。データの転送はドット単位で行われ、転送先のデータと論理演算できます。

以下に、実行手順を説明します。

レジスタのセット

以下のレジスタをセットします。

Destination X register

	7	6	5	4	3	2	1	0
R#36	DX7	DX6	DX5	DX4	DX3	DX2	DX1	DX0
R#37	0	0	0	0	0	0	0	DX8

DX8 ~ DX0 転送先基準点のX座標 (0 ~ 511)

Destination Y register

	7	6	5	4	3	2	1	0
R#38	DY7	DY6	DY5	DY4	DY3	DY2	DY1	DY0
R#39	0	0	0	0	0	0	DY9	DY8

DY9 ~ DY0 転送先基準点のY座標 (0 ~ 1023)

Number of dot X register

	7	6	5	4	3	2	1	0
R#40	MJ7	MJ6	MJ5	MJ4	MJ3	MJ2	MJ1	MJ0
R#41	0	0	0	0	0	0	MJ9	MJ8

MJ9 ~ MJ0 長辺ドット数 (0 ~ 1023)

Number of dot Y register

	7	6	5	4	3	2	1	0
R#42	MI7	MI6	MI5	MI4	MI3	MI2	MI1	MI0
R#43	0	0	0	0	0	0	0	MI8

MI8 ~ MI0 短辺ドット数 (0 ~ 511)

Color register

	7	6	5	4	3	2	1	0
R#44	0	0	0	0	CR3	CR2	CR1	CR0
	0	0	0	0	0	0	CR1	CR0
	CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0

CR7 ~ CR0 描画データ (1ドット)

Argument register

	7	6	5	4	3	2	1	0
R#45	0	0	MXD	0	DIY	DIX	0	MAJ

MXD 転送先のメモリ選択 0 = VRAM、1 = 拡張RAM (MSXでは常に「0」を指定)

DIX 転送先基準点からの転送方向 (X軸方向) 0 = 右、1 = 左

DIY 転送先基準点からの転送方向 (Y軸方向) 0 = 下、1 = 上

MAJ 長辺は 0 = X軸と平行、1 = Y軸と平行 / または長辺 = 短辺

コマンドの実行

R#46の上位4bitに0111B、下位4bitにロジカルオペレーションコード (表4-3参照) を書き込むとコマンドが実行されます。

Command register

	7	6	5	4	3	2	1	0
R#46	0	1	1	1	L03	L02	L01	L00

L03 ~ L00 ロジカルオペレーションコード

コマンド終了の確認

コマンド実行中はCE (S#2のbit0) が「1」になります。CEが「0」になれば、コマンドの終了です。

4.4.10 SRCH (カラーコードのサーチ)

基準点からX軸方向に境界色 (または非境界色) をサーチするコマンドです。

以下に、実行手順を説明します。

レジスタのセット

以下のレジスタをセットします。

Source X register

	7	6	5	4	3	2	1	0
R#32	SX7	SX6	SX5	SX4	SX3	SX2	SX1	SX0
R#33	0	0	0	0	0	0	0	SX8

SX8 ~ SX0 サーチ基準点のX座標 (0 ~ 511)

Source Y register

	7	6	5	4	3	2	1	0
R#34	SY7	SY6	SY5	SY4	SY3	SY2	SY1	SY0
R#35	0	0	0	0	0	0	SY9	SY8

SY9 ~ SY0 サーチ基準点のY座標 (0 ~ 1023)

Color register

	7	6	5	4	3	2	1	0
R#44	0	0	0	0	CR3	CR2	CR1	CR0
	0	0	0	0	0	0	CR1	CR0
	CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0

CR7 ~ CR0 境界色

Argument register

	7	6	5	4	3	2	1	0
R#45	0	0	MXD	0	0	DIX	EQ	0

MXD 転送先のメモリ選択 0 = VRAM、1 = 拡張RAM (MSXでは常に「0」を指定)

DIX サーチ基準点からのサーチ方向 (X軸方向) 0 = 右、1 = 左

EQ 実行終了条件 0 = 境界色を発見したとき、1 = 境界色以外の色を発見したとき

コマンドの実行

R#46に、0110000Bを書き込むとコマンドが実行されます。

Command register

	7	6	5	4	3	2	1	0
R#46	0	1	1	0	0	0	0	0

コマンド終了の確認

コマンド実行中はCE (S#2のbit0) が「1」になります。CEが「0」になれば、コマンドの終了です。

実行結果の読み出し

コマンド終了時、BD (S#2のbit4) が「1」であればEQ (R#45のbit1) の終了条件を満たしたことを示し、S#8とS#9に発見したX座標がセットされます。BDが「0」の場合は、画面端まで検索して終了条件を満たさなかったことを示します。

Status register 2

	7	6	5	4	3	2	1	0
S#2	—	—	—	BD	1	1	—	CE

BD EQ (R#45のbit1) の終了条件を満たしたときに「1」となる
CE 「0」であればコマンド終了を示す

Border X register low

	7	6	5	4	3	2	1	0
S#8	BX7	BX6	BX5	BX4	BX3	BX2	BX1	BX0

S#9	1	1	1	1	1	1	1	BX8
-----	---	---	---	---	---	---	---	-----

BX8 ~ BX0 境界色を発見したX座標 (または、境界色以外の色を発見したX座標)

4.4.11 PSET (点の描画)

VRAM上に点を描画するコマンドです。データの転送はドット単位で行われ、転送先のデータと論理演算できます。

以下に、実行手順を説明します。

レジスタのセット

以下のレジスタをセットします。

Destination X register

	7	6	5	4	3	2	1	0
R#36	DX7	DX6	DX5	DX4	DX3	DX2	DX1	DX0

R#37	0	0	0	0	0	0	0	DX8
------	---	---	---	---	---	---	---	-----

DX8 ~ DX0 転送先基準点のX座標 (0 ~ 511)

Destination Y register

	7	6	5	4	3	2	1	0
R#38	DY7	DY6	DY5	DY4	DY3	DY2	DY1	DY0

R#39	0	0	0	0	0	0	DY9	DY8
------	---	---	---	---	---	---	-----	-----

DY9 ~ DY0 転送先基準点のY座標 (0 ~ 1023)

Color register

	7	6	5	4	3	2	1	0	
R#44	0	0	0	0	CR3	CR2	CR1	CR0	GRAPHIC4・6
	0	0	0	0	0	0	CR1	CR0	GRAPHIC5
	CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0	GRAPHIC7・YJK/ RGB混在・YJK

CR7 ~ CR0 描画色 (1ドット)

Argument register

	7	6	5	4	3	2	1	0
R#45	0	0	MXD	0	0	0	0	0

MXD 転送先のメモリ選択 0 = VRAM、1 = 拡張RAM (MSXでは常に「0」を指定)

コマンドの実行

R#46の上位4bitに0101B、下位4bitにロジカルオペレーションコード (表4-3参照) を書き込むとコマンドが実行されます。

Command register

	7	6	5	4	3	2	1	0
R#46	0	1	0	1	LO3	LO2	LO1	LO0

LO3 ~ LO0 ロジカルオペレーションコード

コマンド終了の確認

コマンド実行中はCE (S#2のbit0) が「1」になります。CEが「0」になれば、コマンドの終了です。

4.4.12 POINT (カラーコードの読み出し)

VRAM上の基準点のカラーコードを読み出すコマンドです。

以下に、実行手順を説明します。

レジスタのセット

以下のレジスタをセットします。

Source X register

	7	6	5	4	3	2	1	0
R#32	SX7	SX6	SX5	SX4	SX3	SX2	SX1	SX0

R#33	0	0	0	0	0	0	0	SX8
------	---	---	---	---	---	---	---	-----

SX8 ~ SX0 転送元基準点のX座標 (0 ~ 511)

Source Y register

	7	6	5	4	3	2	1	0
R#34	SY7	SY6	SY5	SY4	SY3	SY2	SY1	SY0

R#35	0	0	0	0	0	0	SY9	SY8
------	---	---	---	---	---	---	-----	-----

SY9 ~ SY0 転送元基準点のY座標 (0 ~ 1023)

Argument register

	7	6	5	4	3	2	1	0
R#45	0	0	0	MXS	0	0	0	0

MXS 転送元のメモリ選択 0 = VRAM、1 = 拡張RAM (MSXでは常に「0」を指定)

コマンドの実行

R#46に、0100000Bを書き込むとコマンドが実行されます。

Command register

	7	6	5	4	3	2	1	0
R#46	0	1	0	0	0	0	0	0

コマンド終了の確認

コマンド実行中はCE (S#2のbit0) が「1」になります。CEが「0」になれば、コマンドの終了です。

実行結果の読み出し

読み出されたカラーコードがS#7にセットされます。

Color register

	7	6	5	4	3	2	1	0	
S#7	0	0	0	0	CR3	CR2	CR1	CR0	GRAPHIC4・6
	0	0	0	0	0	0	CR1	CR0	GRAPHIC5
	CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0	GRAPHIC7・YJK/ RGB混在・YJK

CR7 ~ CR0 読み出されたカラーコード

4.4.13 STOP (VDPコマンドの中断)

VDPコマンドを中断します。R#46に、0000000Bを書き込むと、実行中のコマンドが中断されます。中断したコマンドは、R#46に同じ値を書き込むことで再開できます。

Command register

	7	6	5	4	3	2	1	0
R#46	0	0	0	0	0	0	0	0

表4-4 コマンド終了時のレジスタの状態

	R#32・33	R#34・35	R#36・37	R#38・39	R#40・41	R#42・43	R#44	R#45	R#46(H)	R#46(L)
HMMC	-	-	-	(2)	-	(3)	-	-	0	-
YMMM	-	(1)	-	(2)	-	(3)	-	-	0	-
HMMM	-	(1)	-	(2)	-	(3)	-	-	0	-
HMMV	-	-	-	(2)	-	(3)	-	-	0	-
LMMC	-	-	-	(2)	-	(3)	-	-	0	-
LMCM	-	(1)	-	-	-	(3)	(4)	-	0	-
LMMM	-	(1)	-	(2)	-	(3)	-	-	0	-
LMMV	-	-	-	(2)	-	(3)	-	-	0	-
LINE	-	-	-	(2)	-	-	-	-	0	-
SRCH	-	-	-	-	-	-	-	-	0	-
PSET	-	-	-	-	-	-	-	-	0	-
POINT	-	-	-	-	-	-	(4)	-	0	-

- 変化しない

(1)コマンド実行前の値をSY、終了時の値をSY*とし、Y方向実行ドット数をNとすれば

$$SY^* = SY + N \text{ (DIY=0)}$$

$$SY^* = SY - N \text{ (DIY=1)}$$

(2)コマンド実行前の値をDY、終了時の値をDY*とし、Y方向実行ドット数をNとすれば

$$DY^* = DY + N \text{ (DIY=0)}$$

$$DY^* = DY - N \text{ (DIY=1)}$$

ただし、LINEでMAJ (R#45のbit0) が「0」のときはN=N-1となる。

(3)コマンド実行前の値をNY、終了時の値をNYBとし、Y方向実行ドット数をNとすれば

$$NYB = NY - N$$

(4)コマンド終了時のカラーコード

4.5 コマンド終了時のレジスタの状態

VDPコマンド終了時のレジスタの状態は表4-4のようになります。

4.6 コマンド処理の高速化

次の操作を行うことで、VDPコマンドの処理速度が上がります。

4.6.1 スプライトの表示を禁止する

SPD (R#8のbit1) を「1」にすると、スプライトの処理に使われていた時間がコマンドの実行に使用されるようになり、コマンドの処理速度が上がります。

4.6.2 画面の表示を禁止する

BL (R#1のbit6) を「0」にすると、画面表示の処理に使われていた時間がコマンドの実行に使用されるようになり、コマンドの処理速度が上がります。

5 スプライト

TMS9918A・V9938・V9958のスプライトについて説明します。

5.1 スプライトに関する基礎知識

TMS9918A・V9938・V9958では1画面に32個のスプライトを表示することができます。スプライトのサイズは8×8ドットまたは16×16ドットです。横方向の1ドットの大きさは画面モードに関わらず一定で画面全体を256に分割した大きさです。縦方向に関しても1ドットの大きさは一定ですが、画面モードによって表示可能な範囲が変化します。このドットを単位として、画面上の任意の位置に表示することができます。

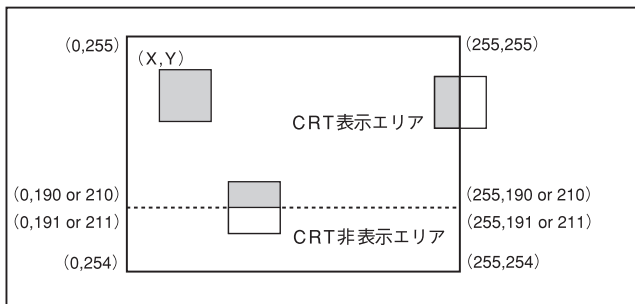


図5-1 スプライト表示範囲

スプライトは独立した表示面を持っているため、原則的に他の表示面に影響を及ぼしません。

V9938・V9958には2種類のスプライトモードが存在し、画面モードによって自動的に選択されます。

- ・スプライトモード1
GRAPHIC1 ~ 2・MULTI COLOR (SCREEN 1 ~ 3)
- ・スプライトモード2
GRAPHIC3 ~ 7・YJK/RGB混合・YJK (SCREEN 4 ~ 8・10 ~ 12)

5.2 スプライトモード1

TMS9918AおよびV9938・V9958のTMS9918A互換モードで使用されます。画面モードにGRAPHIC1 ~ 2・MULTI COLOR (SCREEN 1 ~ 3) を選択すると、自動的にスプライトモード1が使用されます。

5.2.1 特徴

1画面に表示できるスプライトは32個で、それぞれに#0 ~ #31までの番号が付けられています。番号の小さい方が優先度が高く、重なった場合は優先度の高いものが上に表示されます。また、1水平線上に5個以上のスプライトが並んだ場合は優先度の高い順に4個までが表示され、5個目以降は表示されません。

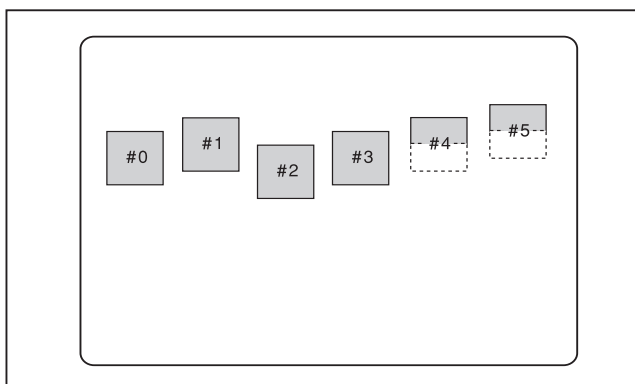


図5-2 スプライトモード1の水平方向のスプライト表示数

1水平線上に5個以上のスプライトが並んだときには、S#0のbit6が「1」になり、S#0のbit4 ~ 0に5個目のスプライト番号(0 ~ 31)がセットされます。また、2個のスプライトが衝突した(パターン「1」の部分が重なった)ときには、S#0のbit5が「1」になり、これによって衝突の発生を知ることができます。

5.2.2 表示

スプライトを表示するときには、次の設定を行ってください。

スプライトのサイズ・拡大・消去・カラーコード0の扱い

R#1で、スプライトのサイズ・拡大を、R#8でスプライトの表示状況・カラーコード0の扱いを選択します。

Mode register 1

	7	6	5	4	3	2	1	0
R#1	—	—	—	—	—	0	SI	MAG

SI スプライトのサイズ 1 = 16 × 16ドット、0 = 8 × 8ドット
MAG スプライトを 1 = 横2倍・縦2倍に拡大する、0 = 拡大しない

Mode register 2

	7	6	5	4	3	2	1	0
R#8	—	—	TP	—	—	0	SPD	—

TP カラーコード0の色を
1 = カラーパレットの色にする。スプライトのカラーコード0で指定された部分は表示され、他のスプライトと重なると衝突が発生する
0 = 透明色として扱う。スプライトのカラーコード0で指定された部分は表示されず、他のスプライトと重なっても衝突は発生しない
SPD スプライトを 1 = 表示しない、0 = 表示する

スプライトパターンジェネレータテーブル

VRAM上のスプライトパターンジェネレータテーブルに、スプライトのパターンをセットします(#0 ~ #255)。

スプライトアトリビュートテーブル

VRAM上のスプライトアトリビュートテーブルに、スプライトの座標・パターン番号・色等をセットします(#0 ~ #31)。

5.2.3 スプライトパターンジェネレータテーブル

スプライトパターンジェネレータテーブルは、スプライトのパターンを記憶する領域です。

パターン1個のフォントは8bytesで構成されており、スプライトパターンジェネレータテーブルは2Kbytesの大きさを持ちますが、必ずしもすべてのパターンを設定する必要はありません。

パターンには#0 ~ #255の番号が付けられており、スプライトのサイズが8 × 8ドットのときはスプライト1個にパターン1個が対応します。16 × 16ドットのときはスプライト1個に4個のパターンが対応し、パターン番号4N、4N + 1、4N + 2、4N + 3が左上、左下、右上、右下の順に並べられます。

スプライトパターンジェネレータテーブルの先頭アドレスは、R#6にセットします。指定できるのは先頭アドレスの上位6bit (A16 ~ A11、TMS9918AではA13 ~ A11)のみで、下位11bit (A10 ~ A0)は「0」と見なされます。よって、スプライトパターンジェネレータテーブルの先頭アドレスとして指定できるのは、0000Hから2Kbytes単位の位置になります。

Sprite pattern generator table base address register

	7	6	5	4	3	2	1	0
R#6	0	0	A16	A15	A14	A13	A12	A11

TMS9918Aでは、A16 ~ A14には「0」をセットしてください。

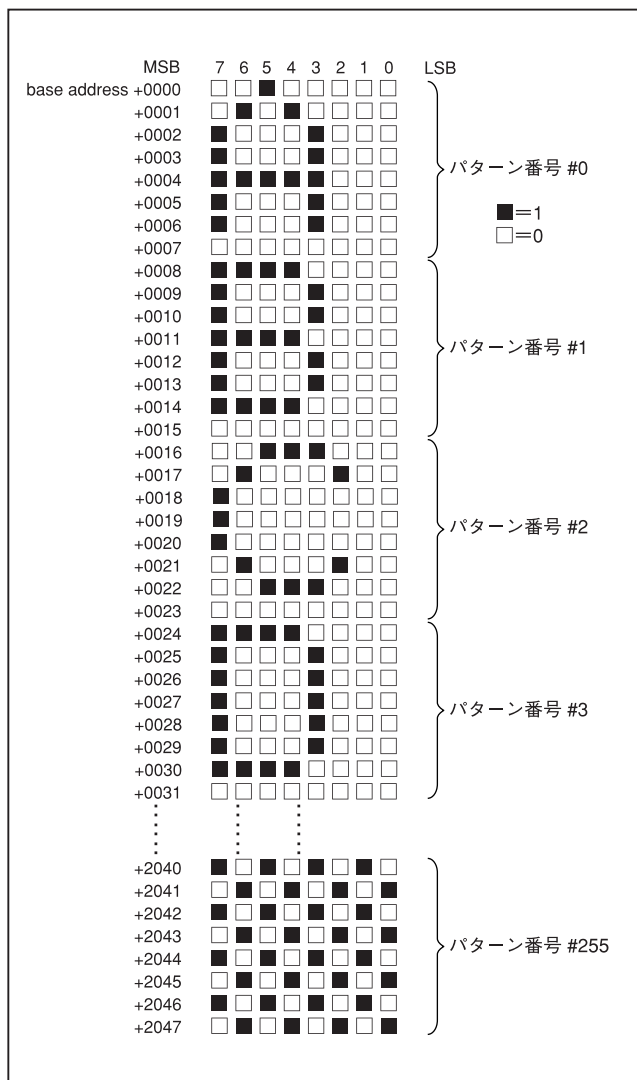


図5-3 スプライトパターンジェネレータテーブル設定例

16 × 16ドットの場合、図5-3のスプライトパターンジェネレータテーブルを例に取れば、次のようになります。

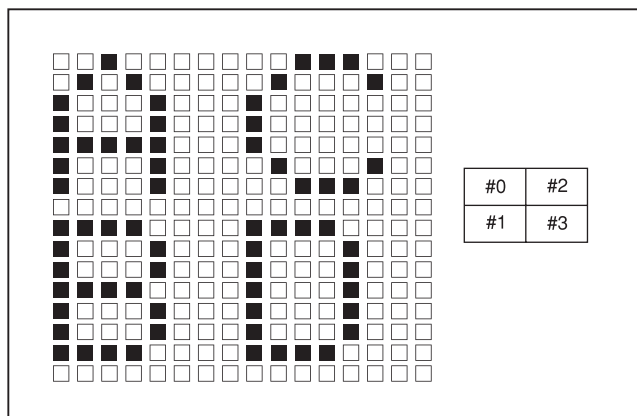


図5-4 スプライトパターンジェネレータテーブルデータ設定例

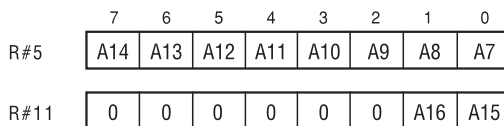
5.2.4 スプライトアトリビュートテーブル

スプライトアトリビュートテーブルは、32個のスプライトの各々の表示位置(X, Y)、「1」の部分の色(「0」の部分は透明)表示するパターン番号等を指定するためにVRAM上に確保される領域です。

1個のスプライトに4bytesのデータ領域が確保されており、テーブル全体で128bytesの大きさを持ちます。

スプライトアトリビュートテーブルの先頭アドレスは、R#5とR#11にセットします。指定できるのは先頭アドレスの上位10bit(A16 ~ A7、TMS9918AではA13 ~ A7)のみで、下位7bit(A6 ~ A0)は「0」と見なされます。よって、スプライトアトリビュートテーブルの先頭アドレスとして指定できるのは、0000Hから128bytes単位の位置になります。

Sprite attribute table base address register



TMS9918Aでは、A16 ~ A14には「0」をセットしてください。

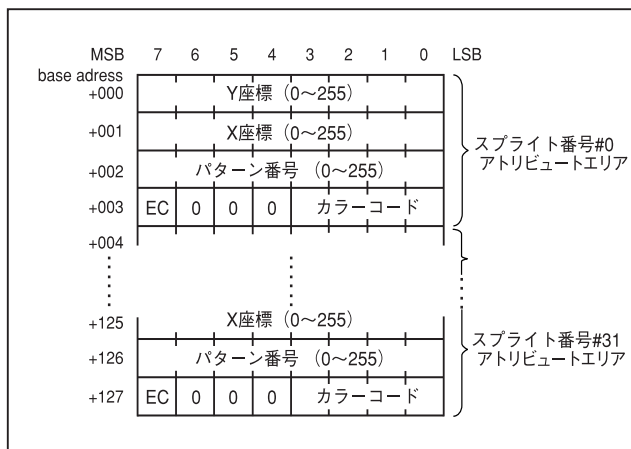


図5-5 スプライトモード1のスプライトアトリビュートテーブル

Y座標 (0 ~ 255)

スプライトのY座標を設定します。画面最上端のY座標が255であることに注意してください。

スプライトのY座標の値を208に設定すると、そのスプライトとそのスプライトより優先度の低い(パターン番号の大きい)スプライトは表示されません。例えば、スプライト番号#16のスプライトのY座標を208にすると、#16 ~ #31までのスプライトは表示されません。特定のスプライトのみを画面から消去するには、表示座標を画面外に設定します(図5-1参照)。

X座標 (0 ~ 255)

スプライトのX座標を設定します。

パターン番号 (0 ~ 255)

スプライトパターンジェネレータテーブル上のパターン番号を指定します。スプライトのサイズが16 × 16のときは、4個のパターンを1個のスプライトに使いますが、このこの場合、4個のパターン番号のうち、

どれを指定しても構いません。

スプライトのサイズを8×8にした場合は256種のパターンを、16×16の場合は64種のパターンを指定できます。

カラーコード(0～15)

スプライトパターンのbitが「1」の部分の色を指定します。

EC (Early Clock)

「1」のとき、スプライトの表示は32ドット左にシフトされます。これにより、スプライトを画面左端から1ドットずつ出現させることができます。

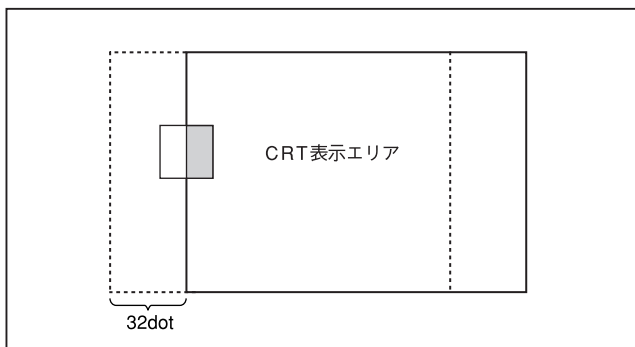


図5-6 ECを「1」にセットした例

5.3 スプライトモード2

V9938・V9958のGRAPHIC3以降 (SCREEN 4以降) で使用されます。画面モードにGRAPHIC3～7・YJK/RGB混在・YJK (SCREEN 4～8・10～12) を選択すると、自動的にスプライトモード2が使用されます。

5.3.1 特徴

1画面に表示できるスプライトは32個で、それぞれに#0～#31までの番号が付けられています。番号の小さい方が優先度が高く、重なった場合は優先度の高いものが上に表示されます。また、1水平線上に9個以上のスプライトが並んだ場合は優先度の高い順に8個までが表示され、9個目以降は表示されません。

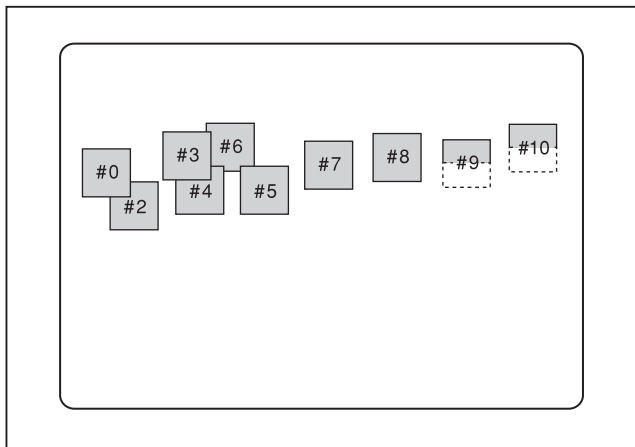


図5-7 スプライトモード2の水平方向のスプライト表示数

1水平線上に9個以上のスプライトが並んだときには、S#0のbit6が「1」になり、S#0のbit4～0に9個目のスプライト番号(0～31)がセットされます。

また、2個のスプライトが衝突した(パターンの「1」の部分が重なった)ときには、S#0のbit5が「1」になり、これによって衝突の発生を知ることができます。このとき、衝突座標がS#3～S#6にセットされます。

スプライトの色を1ラインごとに指定できます。また、スプライトを組み合わせて使うことにより、横方向の色をさらに多く指定できます。

5.3.2 表示

スプライトを表示するときには、次の設定を行ってください。

スプライトのサイズ・拡大・消去・カラーコード0の扱い

スプライトモード1と同様ですが、GRAPHIC7 (SCREEN 8) でR#8のbit5に「1」を指定した(カラーコード0を非透明色にした)場合、カラーコード0がR=0・B=0・G=0の固定色になることに注意してください。

スプライトパターンジェネレータテーブル

VRAM上のスプライトパターンジェネレータテーブルに、スプライトのパターンをセットします(#0～#255)。

スプライトアトリビュートテーブル

VRAM上のスプライトアトリビュートテーブルに、スプライトの座標・パターン番号をセットします(#0～#31)。

スプライトカラーテーブル

VRAM上のスプライトカラーテーブルに、スプライトの各ラインの色・衝突検出の有無・優先順位の有無等をセットします(#0～#31)。

5.3.3 スプライトパターンジェネレータテーブル

スプライトモード1と同様です。「5.2.3 スプライトパターンジェネレータテーブル」を参照してください。

5.3.4 スプライトアトリビュートテーブル

スプライトアトリビュートテーブルは、32個のスプライトの各々の表示位置(X, Y)表示するパターン番号を指定するためにVRAM上に確保される領域です。

1個のスプライトに4bytesのデータ領域が確保されており、テーブル全体で128bytesの大きさを持ちます。

スプライトアトリビュートテーブルの先頭アドレスは、R#5とR#11にセットします。指定できるのは先頭アドレスの上位8bit (A16～A9)のみで、下位9bit (A8～A0)は「0」と見なされます。ただし、A9には必ず「1」をセットします。よって、スプライトアトリビュートテーブルの先頭アドレスとして指定できるのは、0200Hから1Kbytes単位の位置になります。

Sprite attribute table base address register

	7	6	5	4	3	2	1	0
R#5	A14	A13	A12	A11	A10	A9	1	1

必ず「1」をセット

R#11	0	0	0	0	0	0	A16	A15
------	---	---	---	---	---	---	-----	-----

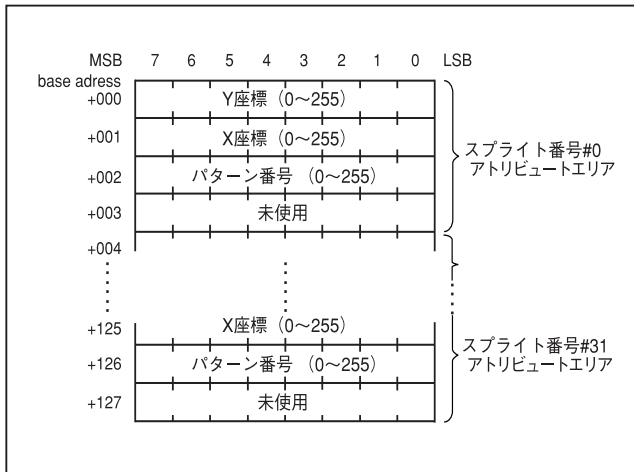


図5-8 スプライトモード2のスプライトアトリビュートテーブル

Y座標 (0 ~ 255)

スプライトのY座標を設定します。画面最上端のY座標が255であることに注意してください。

スプライトのY座標の値を216に設定すると、そのスプライトとそのスプライトより優先度の低い(パターン番号の大きい)スプライトは表示されません。例えば、スプライト番号#16のスプライトのY座標を216にすると、#16~#31までのスプライトは表示されません。特定のスプライトのみを画面から消去するには、表示座標を画面外に設定します(図5-1参照)。

X座標 (0 ~ 255)

スプライトのX座標を設定します。

パターン番号 (0 ~ 255)

スプライトパターンジェネレーターテーブル上のパターン番号を指定します。スプライトのサイズが16×16のときは、4個のパターンを1個のスプライトに使いますが、この場合、4個のパターン番号のうち、どれを指定しても構いません。

スプライトのサイズを8×8にした場合は256種のパターンを、16×16の場合は64種のパターンを指定できます。

5.3.5 スプライトカラーテーブル

スプライトカラーテーブルは、スプライトの各ラインの「1」の部分の表示色(パターンの「0」の部分は透明)優先順位の有無、衝突検出の有無、EC(Early Clock)の指定を行うテーブルです。各スプライトに16ライン分の領域が確保されていますが、スプライトサイズが8×8のときは、先頭の8ライン分を使用してください(図5-9参照)。EC(Early Clock)

「1」のとき、スプライトの指定されたラインの表示は32ドット左にシフトされます。これにより、スプライトを画面左端から1ドットずつ出現させることができます。

CC

「1」のとき、スプライトの指定されたラインの優先順位がなくなり、衝突検出が行われなくなります。スプライトのCCを「1」にしたラインは、そのスプライトより番号が小さく、かつCCが「0」のスプライトが表示されている水平線上のみに表示されます(図5-10参照)。

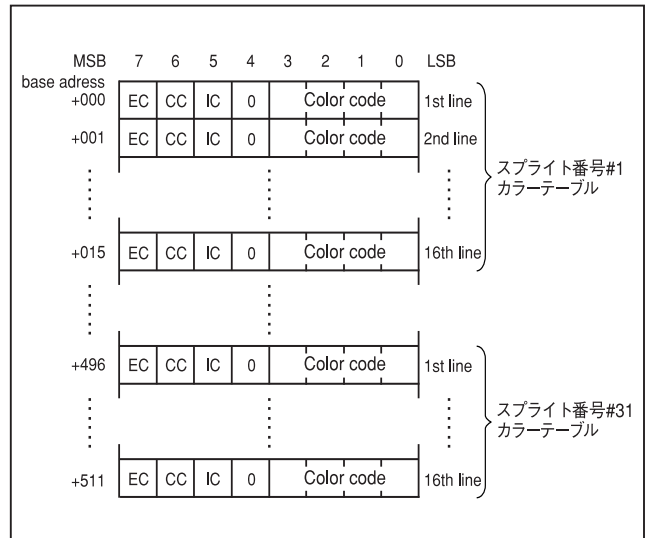


図5-9 スプライトカラーテーブル

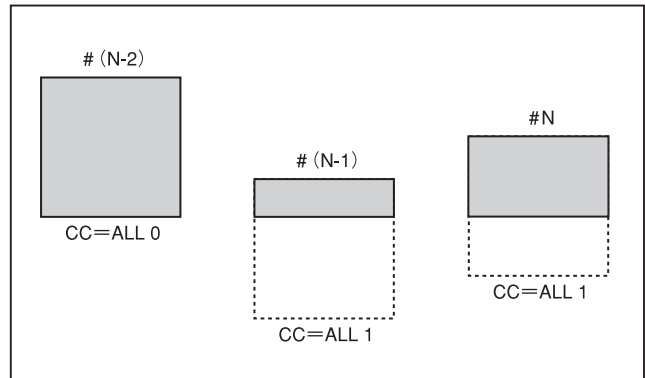


図5-10 スプライトカラーテーブルのCCの効果

CCを「1」にした部分が、そのスプライトより番号が小さく、かつ番号が最も近いスプライトのCCが「0」の部分と重なった場合、衝突検出は行われません。例えば、スプライト番号#2(CC=ALL 0)、#3(CC=ALL 0)、#4(CC=ALL 1)を表示した場合、#2と#3、#2と#4が重なった場合は衝突が発生しますが、#3と#4が重なっても衝突検出は行われません。また、CCが「1」の部分同士が重なっても、衝突検出は行われません。

衝突検出が行われない場合、重なった部分のカラーコードは各スプライトのカラーコードの論理和(OR)になります(上の例では、#3と#4が重なった場合)。この機能を利用して2枚以上のスプライトを重ねて使うことにより、横方向の色数を増やせます。

IC

「1」のとき、スプライトの指定されたラインの衝突検出が行われなくなります。

Color code

スプライトの各ラインのカラーコード(0~15)を指定します。GRAPHIC7(SCREEN 8)のときはパレットは使用されず、固定色が使われます(表5-1参照)。

表 5-1 GRAPHIC7 (SCREEN 8) モード時のスプライトの色

カラーコード	G	R	B
0	0	0	0
1	0	0	2
2	0	3	0
3	0	3	2
4	3	0	0
5	3	0	2
6	3	3	0
7	3	3	2
8	4	7	2
9	0	0	7
10	0	7	0
11	0	7	7
12	7	0	0
13	7	0	7
14	7	7	0
15	7	7	7

スプライトカラーテーブルは、スプライトアトリビュートテーブルの先頭アドレスから512bytesを引いた場所に作成されます。1個のスプライトに16bytesの領域が確保され、スプライトカラーテーブル全体は512bytesの大きさを持ちます。

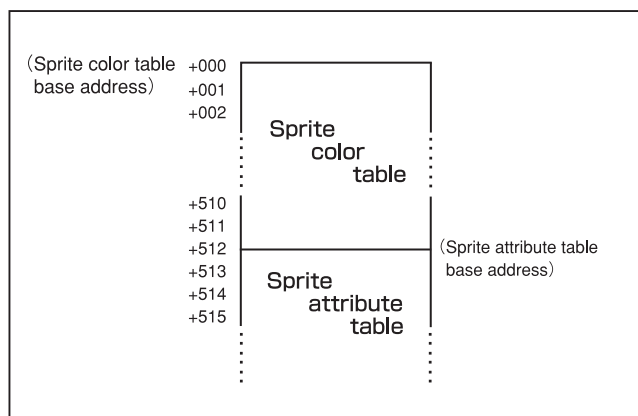


図 5-11 スプライトカラーテーブルとスプライトアトリビュートテーブルの位置

5.3.6 スプライトの衝突

スプライトの、CC・ICが「0」でかつ透明でない部分同士が重なると衝突が発生します（正確な条件は、「5.3.5 スプライトカラーテーブル」のCC・ICに関する説明を参照）。衝突が発生すると、S#0のbit5が「1」になります。このbitはS#0を読み出すことによってリセットされます。このとき、R#8のbit7,6がセットされていなければ（MSXではセットされることはない）、S#3～S#6にスプライトの衝突座標がセットされます。

Column register

	7	6	5	4	3	2	1	0
S#3	X7	X6	X5	X4	X3	X2	X1	X0
S#4	1	1	1	1	1	1	1	X8

X8～X0 衝突座標 X (12～267)

Row register

	7	6	5	4	3	2	1	0
S#5	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
S#6	1	1	1	1	1	1	EO	Y8

Y8～Y0 衝突座標 Y (9～264)
EO 表示フィールドフラグ（インターレースモード・2画面交互表示のときに意味を持つ）
現在の表示フィールドが 0 = 第1フィールド、1 = 第2フィールド

これらのレジスタは、S#5を読み出したときにリセットされます。また、このレジスタ群にセットされる値には、オフセットが付けられており、実際の衝突座標は次のようになります。

実際の衝突座標(XC,YC)

XC=X-12 (X=X8～X0)

YC=Y-9 (Y=Y8～Y0)

5.4 スプライトに対するハードウェアスクロールの影響

R#23 (Display offset register) によるハードウェア縦スクロールは、スプライトにも影響を与えます。

R#23を設定することにより、スプライト面はパターン面と同様に縦方向にスクロールします。このため、パターン面を縦スクロールさせながらスプライトを一定の位置に置く場合（シューティングゲーム等）、スプライトのY座標に補正を加える等の操作が必要となります。Y座標を補正する場合、Y座標を208（スプライトモード1の場合、モード2では216）に設定すると、そのスプライトより優先順位の低いスプライトが表示されなくなることに注意してください。

なお、R#26・R#27によるハードウェア横スクロールは、パターン面に対してのみ有効であり、スプライトがその影響を受けることはありません。

6 その他

6.1 V9958 で削除された機能

V9938に存在する機能で、V9958では削除されたものを説明します。

コンポジットビデオ出力

マウス・ライトペンインターフェイス

以上の機能が削除されています。いずれも、MSX2では使用されていなかった機能です。

これに伴い、レジスタの機能にも変更があります。削除されたbitには常に「0」を設定してください。

Mode register 0

	7	6	5	4	3	2	1	0
R#0	0	DG	IE2	IE1	M5	M4	M3	0

IE2 ライトペンによる割り込み
常に「0」を指定する

Mode register 2

	7	6	5	4	3	2	1	0
R#8	MS	LP	TP	CB	VR	0	SPD	BW

MS マウス
常「0」を指定する

LP ライトペン
常「0」を指定する

Status register 1

	7	6	5	4	3	2	1	0
S#1	FL	LPS	ID#				FH	

FL ライトペンスイッチ・マウススイッチ2
意味を持たない

LPS ライトペンスイッチ・マウススイッチ1
意味を持たない

6.2 コントロールレジスタR#9のIL・EO

コントロールレジスタR#9のIL・EO (bit3・bit2) の組合せによって2画面交互表示・インターレース表示を行えます。EOを「1」にしてこれらの機能を利用するときは、パターンネームテーブルを奇数ページに設定してください(画面モードがGRAPHIC4~7、YJK/RGB混合、YJKのときのみ有効)。

表6-1 IL・EOの機能

IL	EO	画面モード	動作
0	0	テキスト	ノンインターレース・同一画面
		グラフィック	ノンインターレース・同一画面
0	1	テキスト	
		グラフィック	ノンインターレース・1/60秒周期で2画面交互表示
1	0	テキスト	インターレース・第1フィールドと第2フィールドに同一画面
		グラフィック	インターレース・第1フィールドと第2フィールドに同一画面
1	1	テキスト	
		グラフィック	インターレース・第1フィールドに偶数ページ、第2フィールドに奇数ページ

テキスト : TEXT1・2、GRAPHIC1~3、MULTI COLOR
グラフィック : GRAPHIC4~7、YJK/RGB混合、YJK

TEXT1・2、GRAPHIC1~3、MULTI COLORで1/60秒周期の2画面交互表示や、第1フィールドに偶数ページ、第2フィールドに奇数ページを表示するインターレース表示を行うには、垂直帰線割り込み内でEO (S#2のbit1) を参照しながらページ切り替えを行ってください。

6.3 水平帰線割り込み

V9938・V9958は、IE1 (R#0のbit4) を「1」にセットすると水平帰線割り込みを発生します。このとき、R#19に割り込みを発生させる走査線番号を設定できます。

走査線番号とは、画面に表示される走査線に対して画面上部から順に付けられている番号で、画面最上部の走査線番号が「0」となり、以下1,2,3・・・と続いています。

Mode register 0

	7	6	5	4	3	2	1	0
R#0	0	—	—	IE1	—	—	—	0

IE1 水平帰線による割り込みを 1=使用する、0=使用しない

Interrupt line register

	7	6	5	4	3	2	1	0
R#19	IL7	IL6	IL5	IL4	IL3	IL2	IL1	IL0

IL7~IL0 割り込みを発生させる走査線番号

IE1に「1」がセットされている場合、V9938・V9958はR#19に設定された走査線を描き終わると、割り込みを発生し、FH (S#1のbit0) を「1」にします。

割り込みが発生すると、CPUが割り込みを禁止していなければ、割り込みベクタ0038Hが割り込みを禁止されてコールされます。通常は、0038Hからシステムの割り込みルーチンにジャンプしますが、0038Hを書き換えて直接アプリケーションの割り込みルーチンにジャンプすることも可能です。システムの割り込みルーチンを利用する場合は、FD9AH (H.KEYI:すべての割り込みが通る) からアプリケーションの割り込みルーチンにジャンプしてください。

割り込みルーチン内では、まずFH (S#1のbit0) の値を調べ、「0」であればそのままリターンします。FHが「1」であれば、水平帰線割り込みであると判断し割り込み処理を行います。

なお、FHを調べた後、システムの割り込みルーチンに実行を渡す場合はR#15に「0」を設定してからリターンしてください。

Status register 1

	7	6	5	4	3	2	1	0
S#1	FL	LPS	ID#				FH	

FH 水平帰線割り込みフラグ
水平帰線 (R#19で指定) による割り込み (R#0のbit4が「1」のとき) が発生すると「1」になる。S#1を読み出すと「0」になる

R#19を割り込みルーチンの中で再設定することで、1/60秒間に複数回の水平帰線割り込みをかけることが可能です。また、R#23によるハードウェア縦スクロールを行うと、割り込みを発生させる走査線番号もスクロールしてずれるので、割り込み位置を変化させない場合にはR#19にスクロール量を加算する等の補正を行ってください。