

VRAM 転送マネージャ

Vブランク期間における VRAM へのデータ転送

2008-05-30

任天堂株式会社発行

このドキュメントの内容は、機密情報であるため、**厳重な取り扱い、管理を行ってください。**

目次

1	はじめに	4
2	VRAM転送マネージャ.....	4
2.1	概要	4
2.2	VRAM転送マネージャの初期化.....	5
2.3	VRAM転送タスクの登録	5
2.4	VRAM転送タスクの実行	7
2.5	VRAM転送タスクの総転送量を取得	7

コード

コード 2-1	VRAMマネージャの初期化	5
---------	---------------------	---

表

表 2-1	データ転送先の種類	6
-------	-----------------	---

図

図 2-1	VRAMマネージャの概念図	4
-------	---------------------	---

改訂履歴

改訂日	改訂内容
2008-05-30	NITRO-System の名称変更による修正 (NITRO-System から TWL-System に変更)。
2008-04-08	<ul style="list-style-type: none"> ・改訂履歴の書式を変更。 ・ドキュメントの副題を記載。 ・改ページ位置を調整。
2005-01-05	NITRO という表記をニンテンドーDS に統一。
2004-07-16	初版。

1 はじめに

TWL およびニンテンドーDS では、画面表示期間中に VRAM の内容を書き換えることが出来ません。通常、アプリケーションで画面を表示しながら VRAM の内容を書き換える場合には、Vブランク期間中に VRAM にデータを書き込むようにプログラミングします。TWL-System ライブラリの VRAM 転送マネージャでは、アプリケーションからの VRAM 書き換え要求をキューに溜めておき、Vブランク期間に要求されているデータを VRAM に書き込むことが可能となる機能を提供します。

2 VRAM転送マネージャ

2.1 概要

VRAM 転送マネージャは、ユーザからの VRAM 書き換え要求を VRAM 転送タスクとしてキューに登録しておき、後のあるタイミングで登録されている VRAM 転送タスクに従って VRAM へのデータ書き込みを行うという機能を提供するものです。画面表示期間中に登録された VRAM 転送タスクに従って、Vブランク期間中にデータを VRAM に書き込むという用途に使用します。

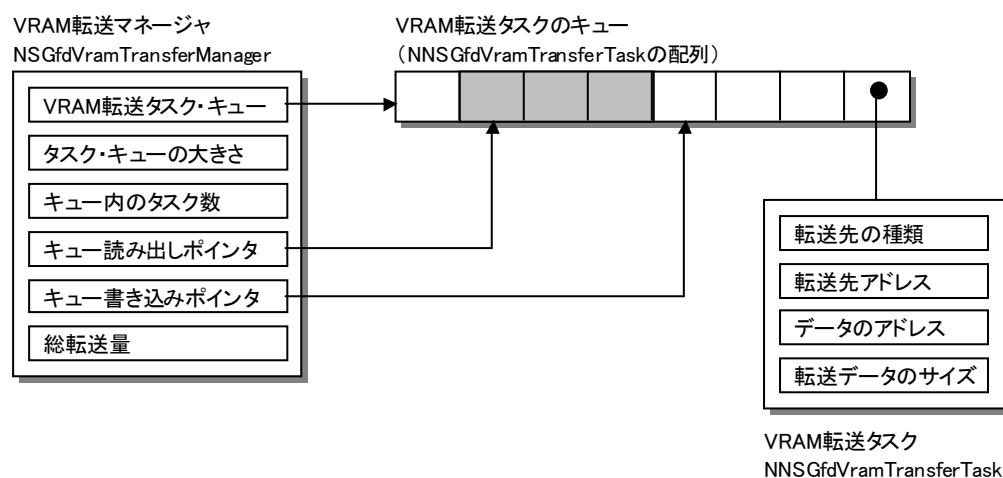


図 2-1 VRAM マネージャの概念図

2.2 VRAM転送マネージャの初期化

VRAM 転送マネージャは、これを使用する前に初期化を行う必要があります。VRAM 転送マネージャを初期化するためには、下記の関数を呼び出します。

```
void NNS_GfdInitVramTransferManager(  
    NNSGfdVramTransferTask* pTaskArray, u32 lengthOfArray);
```

NNS_GfdInitVramTransferManager()関数は、VRAM 転送マネージャを初期化し、VRAM 転送マネージャを使用可能な状態にします。

VRAM 転送マネージャは、VRAM 転送タスクを NNSGfdVramTransferTask 構造体の配列に保存します。VRAM 転送マネージャを初期化する場合には、ユーザーが NNSGfdVramTransferTask 構造体の配列を用意する必要があります。ユーザーは NNSGfdVramTransferTask 構造体の配列を準備し、NNS_GfdInitVramTransferManager() 関数の1番目の引数の pTaskArray に配列へのポインタを、2番目の引数の lengthOfArray に配列の大きさをそれぞれ指定する必要があります。

1つの VRAM 転送タスクを登録する為には、16 バイトの容量が必要となります。NNSGfdVramTransferTask 構造体の配列の大きさは、VRAM 転送マネージャに一度に登録したい VRAM 転送タスクの最大数に合わせて準備して下さい。

コード 2-1 VRAM マネージャの初期化

```
#define NUM_TASKS    8  
  
NNSGfdVramTransferTask    taskArray[NUM_TASKS];  
  
NNS_GfdInitVramTransferManager(taskArray, NUM_TASKS);  
....
```

2.3 VRAM転送タスクの登録

VRAM 転送マネージャに VRAM 転送タスクを登録する為には、下記の関数を呼び出します。

```
BOOL NNS_GfdRegisterNewVramTransferTask(  
    NNS_GFD_DST_TYPE type, u32 dstAddr, void* pSrc, u32 szByte);
```

NNS_GfdRegisterNewVramTransferTask()関数は、指定された VRAM 書き換え用データを VRAM 転送タスクという形で VRAM 転送マネージャのキューに順番に登録していきます。VRAM 転送マネージャに一度に登録できる VRAM 転送タスクの数は、VRAM 転送マネージャの初期化時に指定した NNSGfdVramTransferTask 構造体の配列の要素数となります。

データ転送先の種類の type 引数には、VRAM 転送マネージャによって、どのような種類のデータを転送するかを指定します。type には、以下の種類を指定する事ができます。

表 2-1 データ転送先の種類

カテゴリ	データ転送先の種類を示す定数名	意 味
3D	NNS_GFD_DST_3D_TEX_VRAM	3Dテクスチャイメージの書き換え
	NNS_GFD_DST_3D_TEX_PLTT	3Dテクスチャパレットの書き換え
	NNS_GFD_DST_3D_CLRIMG_COLOR	3Dクリアイメージカラーの書き換え
	NNS_GFD_DST_3D_CLRIMG_DEPTH	3Dクリアイメージデプスの書き換え
2Dメイン	NNS_GFD_DST_2D_BG0_CHAR_MAIN	2DメインのBG0キャラクタの書き換え
	NNS_GFD_DST_2D_BG1_CHAR_MAIN	2DメインのBG1キャラクタの書き換え
	NNS_GFD_DST_2D_BG2_CHAR_MAIN	2DメインのBG2キャラクタの書き換え
	NNS_GFD_DST_2D_BG3_CHAR_MAIN	2DメインのBG3キャラクタの書き換え
	NNS_GFD_DST_2D_BG0_SCR_MAIN	2DメインのBG0スクリーンの書き換え
	NNS_GFD_DST_2D_BG1_SCR_MAIN	2DメインのBG1スクリーンの書き換え
	NNS_GFD_DST_2D_BG2_SCR_MAIN	2DメインのBG2スクリーンの書き換え
	NNS_GFD_DST_2D_BG3_SCR_MAIN	2DメインのBG3スクリーンの書き換え
	NNS_GFD_DST_2D_BG2_BMP_MAIN	2DメインのBG2ビットマップの書き換え
	NNS_GFD_DST_2D_BG3_BMP_MAIN	2DメインのBG3ビットマップの書き換え
	NNS_GFD_DST_2D_OBJ_PLTT_MAIN	2DメインのOBJパレットの書き換え
	NNS_GFD_DST_2D_BG_PLTT_MAIN	2DメインのBGパレットの書き換え
	NNS_GFD_DST_2D_OBJ_EXTPLTT_MAIN	2DメインのOBJ拡張パレットの書き換え
	NNS_GFD_DST_2D_BG_EXTPLTT_MAIN	2DメインのBG拡張パレットの書き換え
	NNS_GFD_DST_2D_OBJ_OAM_MAIN	2DメインのOAMの書き換え
	NNS_GFD_DST_2D_OBJ_CHAR_MAIN	2DメインのOBJキャラクタの書き換え
2Dサブ	NNS_GFD_DST_2D_BG0_CHAR_SUB	2DサブのBG0キャラクタの書き換え
	NNS_GFD_DST_2D_BG1_CHAR_SUB	2DサブのBG1キャラクタの書き換え
	NNS_GFD_DST_2D_BG2_CHAR_SUB	2DサブのBG2キャラクタの書き換え
	NNS_GFD_DST_2D_BG3_CHAR_SUB	2DサブのBG3キャラクタの書き換え
	NNS_GFD_DST_2D_BG0_SCR_SUB	2DサブのBG0スクリーンの書き換え
	NNS_GFD_DST_2D_BG1_SCR_SUB	2DサブのBG1スクリーンの書き換え
	NNS_GFD_DST_2D_BG2_SCR_SUB	2DサブのBG2スクリーンの書き換え
	NNS_GFD_DST_2D_BG3_SCR_SUB	2DサブのBG3スクリーンの書き換え
	NNS_GFD_DST_2D_BG2_BMP_SUB	2DサブのBG2ビットマップの書き換え
	NNS_GFD_DST_2D_BG3_BMP_SUB	2DサブのBG3ビットマップの書き換え
	NNS_GFD_DST_2D_OBJ_PLTT_SUB	2DサブのOBJパレットの書き換え
	NNS_GFD_DST_2D_BG_PLTT_SUB	2DサブのBGパレットの書き換え
	NNS_GFD_DST_2D_OBJ_EXTPLTT_SUB	2DサブのOBJ拡張パレットの書き換え
	NNS_GFD_DST_2D_BG_EXTPLTT_SUB	2DサブのBG拡張パレットの書き換え
	NNS_GFD_DST_2D_OBJ_OAM_SUB	2DサブのOAMの書き換え
	NNS_GFD_DST_2D_OBJ_CHAR_SUB	2DサブのOBJキャラクタの書き換え

VRAM 転送タスクには、上記のデータ転送先の種類と、転送先のアドレス、データのアドレス、転送データのサイズの4つの情報が格納されます。実際に転送されるデータはコピーされませんので、VRAM 転送マネージャがデータをVRAM に書き込むまでは、データをメモリに保持しておく必要があります。なお、データはユーザがNNS_GfdDoVramTransfer()関数を呼び出した時点でVRAM に書き込まれます。

2.4 VRAM転送タスクの実行

VRAM 転送マネージャに登録されているデータを VRAM に書き込む為には、下記の関数を呼び出します。

```
void NNS_GfdDoVramTransfer(void);
```

通常、NNS_GfdDoVramTransfer()関数は、VRAM にデータを書き込むことが可能な期間であるVブランク期間中に呼び出す必要があります。NNS_GfdDoVramTransfer()関数は、VRAM 転送タスクを VRAM 転送マネージャのキューに登録された順番に全て実行します。この VRAM 転送マネージャでは、登録されている全ての VRAM 転送タスクがVブランク期間中に実行可能かどうかは管理していません。VRAM 書き換えデータの総量はユーザが管理し、Vブランク期間中に書き換えが可能な量を VRAM 転送マネージャに登録する必要があります。

2.5 VRAM転送タスクの総転送量を取得

VRAM 転送マネージャでは、VRAM 転送マネージャに登録されている VRAM 転送タスクの総転送量を取得する事が出来ます。VRAM 転送タスクの総転送量を取得する為には、以下の関数を呼び出します。

```
u32 NNS_GfdGetVramTransferTaskTotalSize(void);
```

この関数を用いて総転送量を取得して調べることにより、データの転送がVブランク中で完了しそうかどうかを判定することが可能となります。

© 2004-2008 Nintendo

任天堂株式会社の許諾を得ることなく、本書に記載されている内容の一部あるいは全部を無断で複製・複製・転写・頒布・貸与することを禁じます。