SH7785 グループ

SH7785　DU 設定例

R01AN0244JJ0101
Rev.1.01
2010.12.01

要旨

この資料は、SH7785 のディスプレイユニット(DU)の使用方法と，液晶ディスプレイに表示する設定例を掲載しています。

動作確認デバイス

SH7785

目次

1. はじめに

1.1 仕様

- 本アプリケーションノートでは，2 章で DU のハードウェアマニュアルを補足する説明を行い，3 章では，液晶ディスプレイに 4 種類の画像を表示し、各画像の表示の ON/OFF させる応用例を掲載しています。

1.2 使用機能

- シリアルコミュニケーションインタフェース(SCIF1)
- 割込みコントローラ(INTC)
- ディスプレイユニット(DU)
- DMA コントローラ(DMAC)
- 内蔵メモリ(U メモリ)

1.3 適用条件

評価ボード:	アルファプロジェクト製 SH-4A ボード型番 AP-SH4A-3A	
	外付けメモリ	(エリア 0): NOR 型フラッシュメモリ 16M バイト Spansion 製 S29GL128P90TFIR20
		(エリア 3): DDR2-SDRAM64M バイト Micron 製 MT47H16M16BG-3
	表示出力	DVI-I コネクタ
		RGB->DVI 変換 IC: TexasInstruments 製 TFP410
マイコン:	SH7785(R8A7785)	
動作周波数:	CPU クロック:	600MHz
	SuperHyway クロック:	300MHz
	DDR2 クロック:	300MHz
	バスクロック:	100MHz
	周辺クロック:	50MHz
エリア 0 バス幅:	16bit 固定(MD5 端子＝Low レベル，MD6 端子＝High レベル)	
クロック動作モード:	モード 16(MD0 端子＝Low レベル，MD1 端子＝Low レベル，MD2 端子＝Low レベル，MD3 端子＝Low レベル，MD4 端子＝High レベル)	
エンディアン:	ビッグエンディアン(MD8 端子＝Low レベル)	
アドレスモード:	29bit アドレスモード(MD13 端子=Low レベル)	
ツールチェイン:	SuperH RISC engine Standerd Toolchain Ver.9.3.2.0	
コンパイラオプション:	High-performance Embedded Workshop で include 指定以外はデフォルト設定 -cpu=sh4a -include="$(PROJDIR)¥inc","$(PROJDIR)¥inc¥drv" -object="$(CONFIGDIR)¥$(FILELEAF).obj" -debug -gbr=auto -chgincpath -errorpath -global_volatile=0 -opt_range=all -infinite_loop=0 -del_vacant_loop=0 -struct_alloc=1 –nologo	

アセンブラオプション: -cpu=sh4a -endian=big -round=zero -denormalize=off -include="$(PROJDIR)¥inc"
-debug -object="$(CONFIGDIR)¥$(FILELEAF).obj"
-literal=pool,branch,jump,return -nolist -nologo -chgincpath -errorpath

1.4 本アプリケーションで用いる用語の説明

- フレーム:
  ディスプレイに表示するイメージの通りに画素情報をメモリに配置したもので，フレーム情報が配置されたバッファをフレームバッファと呼びます。DU はフレームバッファから画素情報を読み出して画像を表示します。
- リフレッシュレート[Hz]:
  DU に接続された液晶ディスプレイが 1 秒間に何回画面を書き換えるかを表し，DU が 1 秒間に出力する垂直同期信号 VSYNC の周波数に等しくなります。
- フレームレート[fps]:
  画像表示システムが 1 秒間に更新する画像の枚数です。DU が更新できる画像枚数の上限は，リフレッシュレート[Hz]と等しくなります。

  フレームレートがリフレッシュレートより低い場合，例えば VSYNC が 60Hz の DU に対してフレームレートが 30fps のとき，DU は VSYNC60 回につき 30 回，更新された画像を表示しています。

1.5 本アプリケーションノートの適用範囲

本アプリケーションノートは，OS 非搭載でフレームバッファの画像を RGB インタフェース方式の液晶ディスプレイに表示する。DU の基本的な使用方法について説明します。以下の機能については，本アプリケーションノートの説明対象外となります。

— 表示データフォーマット(8bit/pixel，ARGB(1555)，YC)
— スクロール
— ラップアラウンド
— ブリンキング
— TV 同期モード(外部同期モード)
— 同期方式切り替えモード
— インタレースインクモード
— インタレースシンク&ビデオモード
— YC→RGB 色空間変換機能
— カラーパレット
— コンポジット出力

1.6 関連アプリケーションノート

本資料の参考プログラムは，「SH7785 グループアプリケーションノート SH7785 初期設定例 (R01AN0242JJ0101)」に変更を加えた設定条件で動作確認しています。併せて参照ください。

2. DUの動作説明

2.1 DUの概要

DU は外部メモリ上に配置されたフレームバッファから画像データを読出し，液晶ディスプレイに出力して画像を表示させることが可能な画像系のジュールです。本アプリケーションノートでは RGB 形式の BMP 画像(16bit:RGB565)データを読出し，RGB インタフェース方式の液晶ディスプレイに画像を表示させる設定例を紹介します。

図 1 に DU の動作イメージを示します。

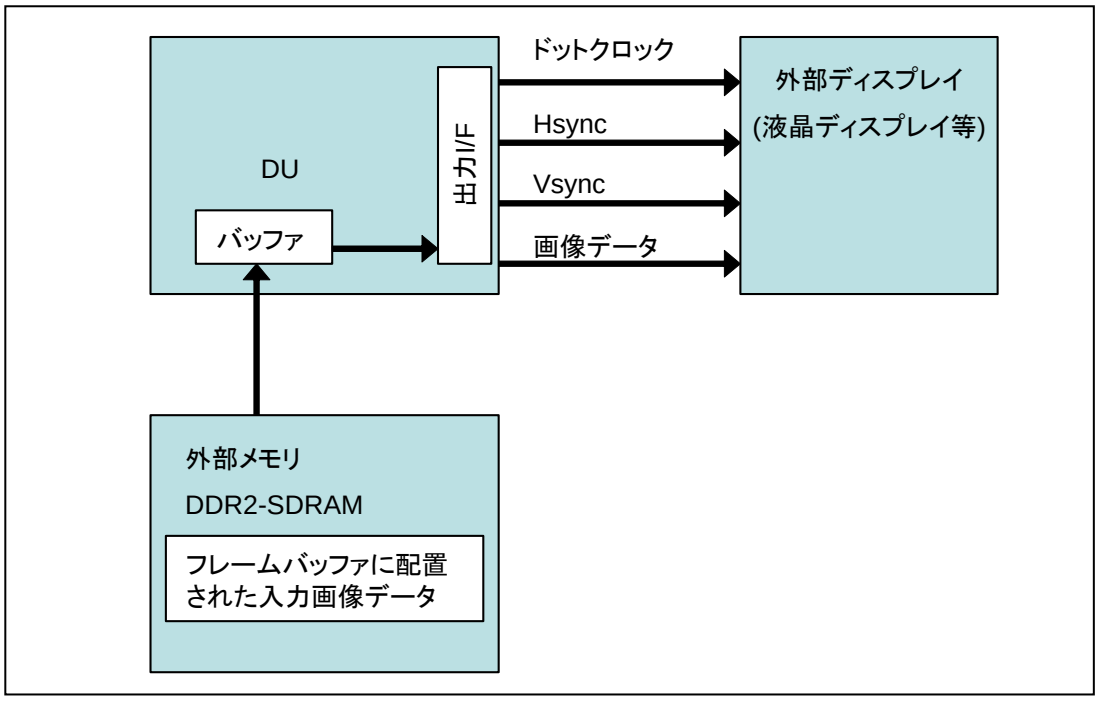

図 1 DU の動作イメージ

2.1.1 出力画面構造

DU は，表示面をプレーンと呼びます。プレーンは最大で 6 画面あり，画面サイズによって使用できるプレーン数が限られます(画面サイズ 480x234 で 6 画面，画面サイズ WVGA(854x480)で 4 画面，画面サイズ SVGA(800x600)で 3 画面)。プレーンは重ね合わせることが可能で，重ね合わせ順序は任意に設定が可能です。

また，各プレーンは表示オン/オフや表示データフォーマット，ブレンディング機能などの独自設定が可能で，ダブルバッファ構成となっています。本アプリケーションノートでは画面サイズ WVGA で 4 画面を使用し，プレーンの重ね合わせで透過色の設定例を紹介します。

図 2 にプレーン構成及び重ね合わせ概略を示します。

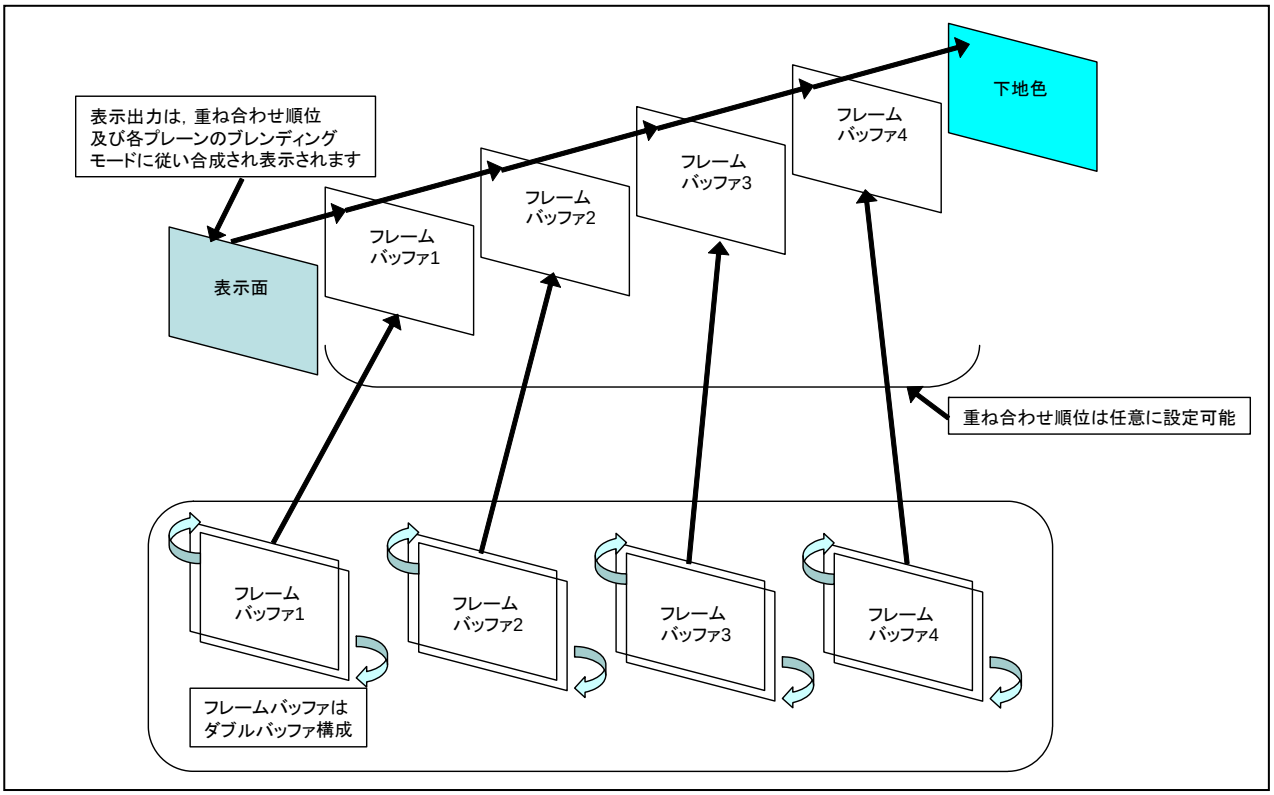

図 2 プレーン構成及び重ね合わせ概略

2.1.2 画像データの読み出し

DU は，フレームバッファを画面左上の原点から右方向に読み出します。(プレーン開始位置 X レジスタ(PnSPXR)，プレーン n 開始位置 Y レジスタ(PnSPYR)，にてフレームバッファの左上を原点に読出し開始位置を設定することが可能)

図 3 に画像データの入力を示します。

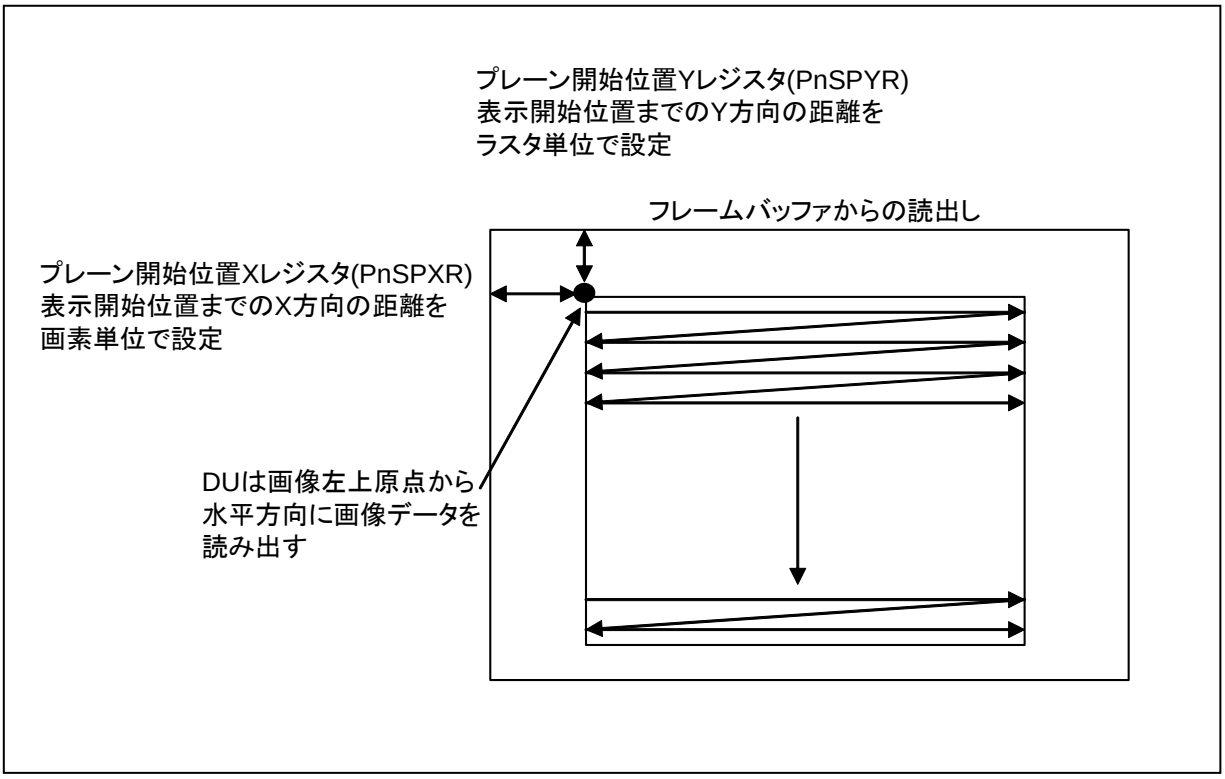

図 3 画像データの入力

2.1.3 画像データの出力

DU は，下記の 3 つの同期信号に同期して画像データを出力します。

— ドットクロック(DCLKOUT):
1 画素の情報はドットクロックに同期して出力します。

— 水平同期信号(Hsync):
画像の横幅 1 ライン分の情報は水平同期信号に同期して出力します。同期信号の前後の画素情報を出力しない期間をそれぞれ水平フロントポーチ、水平バックポーチと呼びます。

— 垂直同期信号(Vsync):
画像 1 フレーム分の情報は垂直同期信号に同期して出力します。同期信号の前後の画素情報を出力しない期間をそれぞれ垂直フロントポーチ、垂直バックポーチと呼びます。

図 4 に画像データと同期信号の出力を示します。

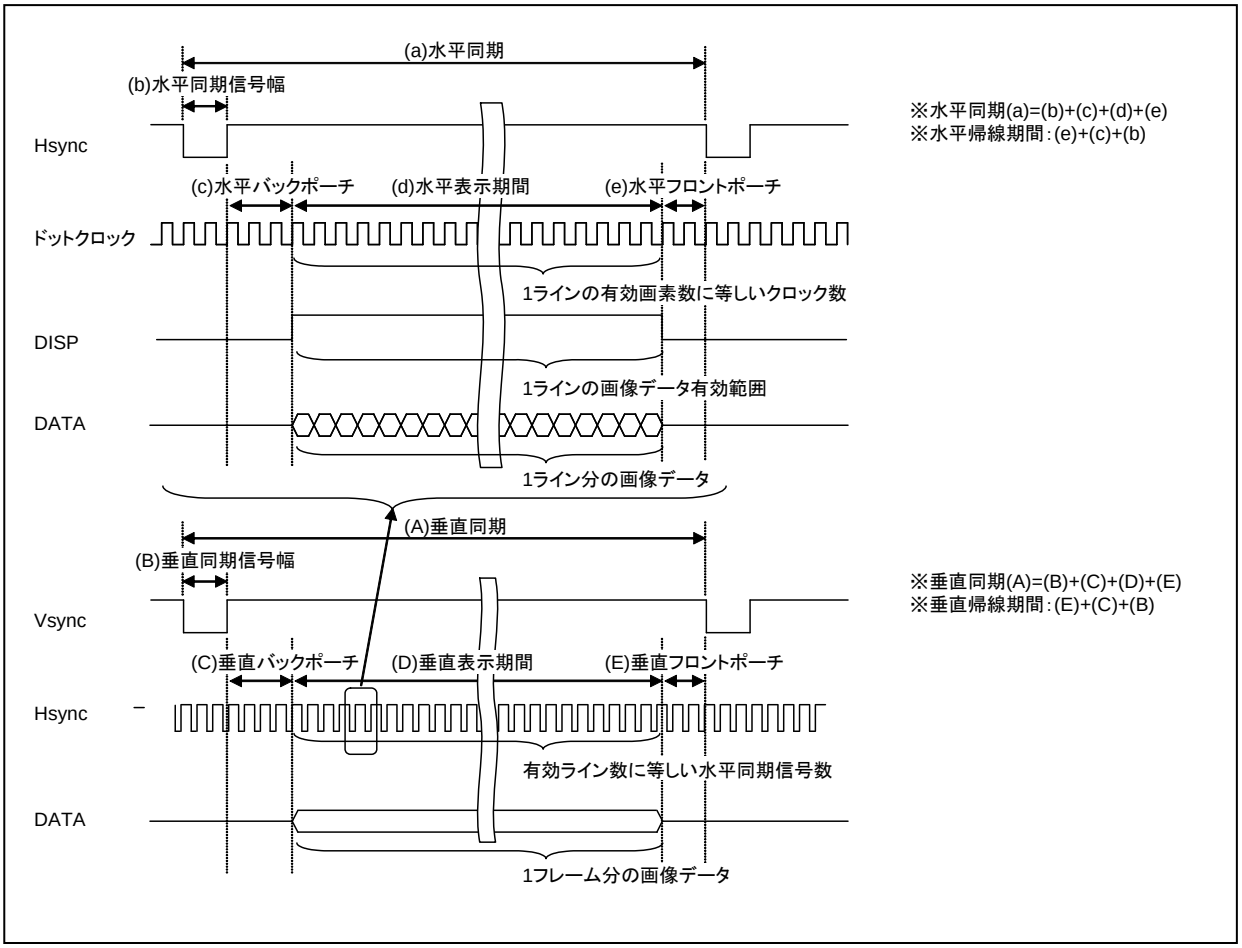

図 4 画像データと同期信号の出力

2.1.4 エンディアン変換

DU は、表示システム制御レジスタ（DSYSR）の DSEC ビットの設定によりビッグエンディアン／リトルエンディアンの変換が可能です。

DU 内部はリトルエンディアン固定となっており、表示システム制御レジスタ（DSYSR）の DSEC ビットを 1 に設定することで、メモリ上にビッグエンディアンで配置された表示データをリトルエンディアンに変換して読み出しが行えます。

表 1 にエンディアン変換単位を示します。

表 1 エンディアン変換単位

PnMR/PnDDF	データファーマット	エンディアン変換の単位
00	8bit/pixel	バイト単位
01	16bit/pixel	ワード単位
10	ARGB	ワード単位
11	YC	バイト単位

図 5 に各単位におけるエンディアン変換図を示します。

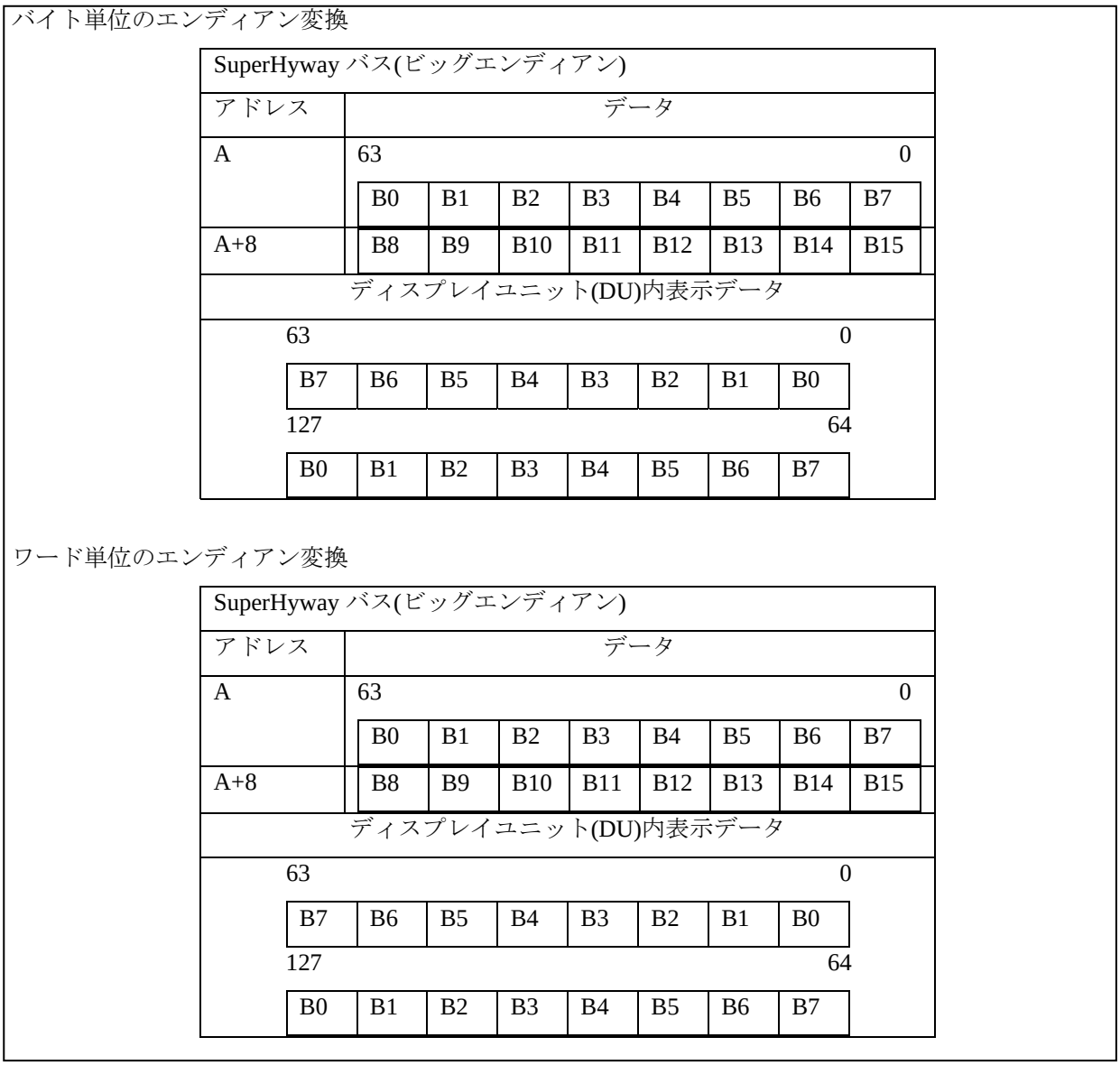

図 5 エンディアン変換図

2.1.5 スクロール

DU は，各プレーンごとに表示領域と表示サイズおよび開始位置をプレーンごと独立に設定することにより，スムースばスクロール処理が可能です。

スクロール表示を行うためには，各プレーンの表示領域開始アドレス 0～1 レジスタ(PnDSA0～1R)で指定したメモリの先頭を原点として，プレーン n 表示開始位置 X，Y(プレーン n 開始位置 X レジスタ(PnSPXR)およびプレーン n 開始位置 Y レジスタ(PnSPYR)で指定された座標)をサイクリックに設定することにより叶です。

スクロール表示概要を図 6 に示します。表示開始位置を A から B に設定するこによりスクロール表示を行います。

*注:各プレーンの表示サイズなどの領域設定は,メモリ構成領域外を表示しないように設定してください。

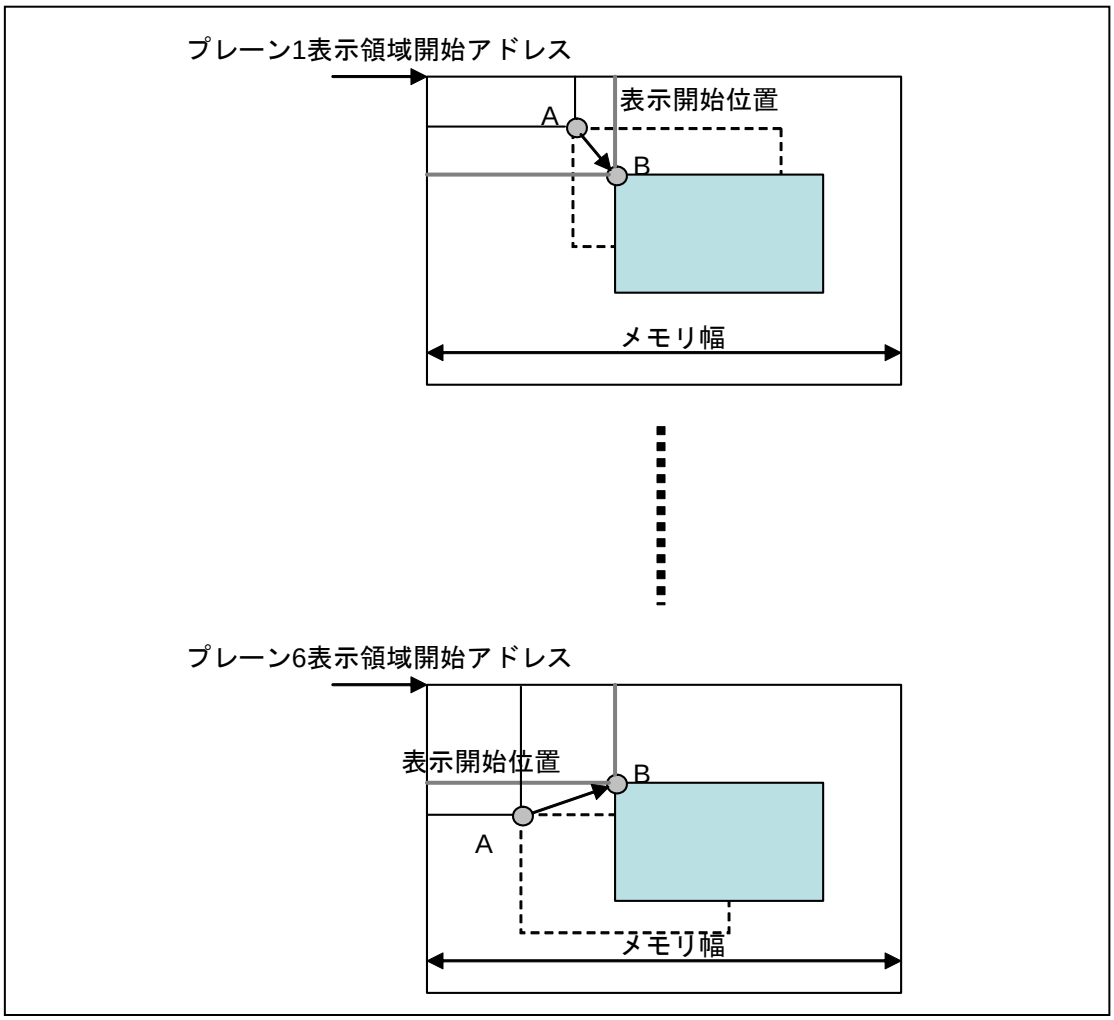

図 6 スクロール機能概略図

2.2 DUの仕様

DU の主な仕様を表 2 に示します。

表 2 DU の仕様一覧

項目	仕様
表示可能な最大サイズ (使用可能な最大プレーン数)	画面サイズ:480x234，最大プレーン数:6 画面 画面サイズ:854x480，最大プレーン数:4 画面 画面サイズ:800x600，最大プレーン数:3 画面
入力データ形式	8bit/pixel 16bit/pixel:RGB 16bit/pixel:ARGB YC:UYVY 形式，YUYV 形式
出力データ形式	8bit/pixel 16bit/pixel:RGB 16bit/pixel:ARGB YC→RGB
DU 出力インタフェース	水平同期，垂直同期信号によるインタフェース 信号の極性が設定可能 画像信号の出力幅と出力位置が設定可能 コンポジット同期信号出力可能 ※1 ※1：本アプリケーションノートでは取り扱いません
ドットクロック	ソースクロックを 2 種類から選択し，分周して出力可能 外部入力クロック(DCLKIN) DU クロック(DUck) 最大 50MHz まで出力可能。 分周率は 1/1 分周～1/32 分周の設定が可能。
プレーン	最大 6 画面のプレーン 各プレーン表示 ON/OFF 設定可能 重ね合わせる優先順位設定可能 表示サイズ設定可能

2.3 DUの設定

DU の機能及びレジスタへの設定方法を示します。

2.3.1 表示出力の設定

DU は Hsync，Vsync，デジタル RGB 出力の他に，コンポジット同期信号出力が可能です。

本アプリケーションノートではコンポジット同期信号出力は取り扱いません。

表 3 に表示出力の設定に使用する DU のレジスタを示します。

表 3 表示出力設定

機能	レジスタ名
表示出力モード設定	表示システム制御レジスタ(DSYSR)
表示出力信号設定	表示モードレジスタ(DSMR)

2.3.2 プレーンパラメータの設定

DU がメモリから画像データを読み出す際に必要とするフレームバッファ情報をプレーンの設定を合せて図 7 に示します。

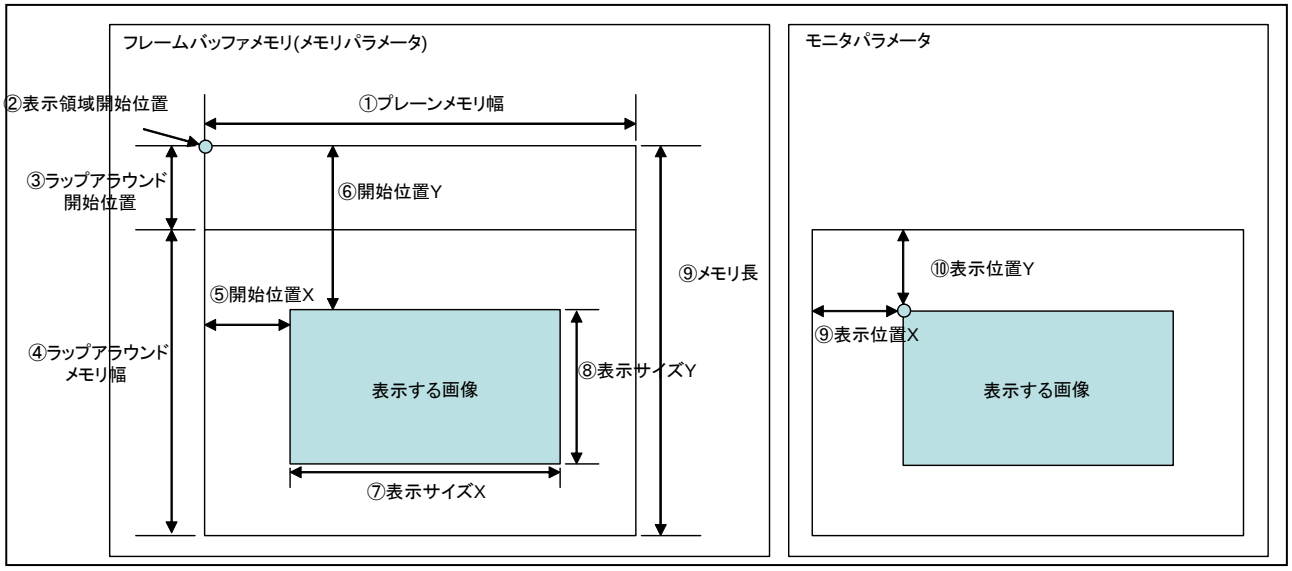

図 7 プレーン構成

表 4 に DU の各プレーンの設定に使用するレジスタを示します。

表 4　各プレーンの設定レジスタ

NO	機能	レジスタ名	説明
①	プレーンメモリ幅(MWX)	PnMWXR	プレーンの X 方向のメモリ幅を 16～4096 画素まで 16 画素単位で設定します。
②	表示領域開始位置(DSA)	PnDSA0,1	2.3.3 メモリ割付参照
③	ラップアラウンド開始位置(WASPY)	PnWASPR	DSA で設定したアドレスを基準に，各プレーンのラップアラウンドエリアの Y 方向開始位置をライン単位で設定します。
④	ラップアラウンドメモリ幅(WAMWY)	PnWAMWR	ラップアラウンドの Y 方向のメモリ幅を 240～4095 ラインの範囲で任意に設定します。
⑤	開始位置 X(SPX)	PnSPXR	DSA 設定したアドレスを原点として表示開始位置までの X 方向の距離を画素単位で設定します。
⑥	開始位置 Y(SPY)	PnSPYR	DSA 設定したアドレスを原点として表示開始位置までの Y 方向の距離をラスタ単位で設定します。
⑦	表示サイズ X(DSX)	PnDSXR	各プレーンの X 方向の表示サイズを画素単位で設定します。
⑧	表示サイズ Y(DSY)	PnDSYR	各プレーンの Y 方向の表示サイズをラスタ単位で設定します。
⑨	表示位置 X(DPX)	PnDPXR	モニタの左上を原点として表示位置までの X 方向の距離を画素単位で設定します。
⑩	表示位置 Y(DPY)	PnDPYR	モニタの左上を原点として表示位置までの Y 方向の距離をラスタライン単位で設定します。

※n=1～6

2.3.3 メモリ割付

DU の各プレーンは表示面の表示領域開始アドレスを個別に設定することが出来ます。表示領域開始アドレスレジスタには，使用するメモリ領域の先頭アドレスを 29 ビットまたは 32 ビットで各々に設定します。

32 ビットで設定する場合，表示拡張機能許可レジスタ(DFER)の DSAE を’1’にセットしてください。

また，プレーン毎の表示領域開始アドレス 0 及び 1 を使用してダブルバッファ制御を行い各プレーンを表示します。

表 5 に DU のメモリ割付設定レジスタを示します。

表 5 メモリ割付設定レジスタ

表示面	設定レジスタ名称	
プレーン 1	プレーン 1 表示領域開始アドレスレジスタ 0	P1DSA0
	表示領域開始アドレスレジスタ 1	P1DSA1
プレーン 2	プレーン 2 表示領域開始アドレスレジスタ 0	P2DSA0
	表示領域開始アドレスレジスタ 1	P2DSA1
プレーン 3	プレーン 3 表示領域開始アドレスレジスタ 0	P3DSA0
	表示領域開始アドレスレジスタ 1	P3DSA1
プレーン 4	プレーン 4 表示領域開始アドレスレジスタ 0	P4DSA0
	表示領域開始アドレスレジスタ 1	P4DSA1
プレーン 5	プレーン 5 表示領域開始アドレスレジスタ 0	P5DSA0
	表示領域開始アドレスレジスタ 1	P5DSA1
プレーン 6	プレーン 6 表示領域開始アドレスレジスタ 0	P6DSA0
	表示領域開始アドレスレジスタ 1	P6DSA1

表 6 に DU の設定アドレス 32 ビット拡張許可レジスタを示します。

表 6 設定アドレス 32 ビット拡張

機能	レジスタ名	ビット名
設定アドレス 32 ビット拡張許可	表示拡張機能許可レジスタ(DFER)	DSAE

2.3.4 プレーン優先順位の設定

DU の各プレーンを重ね合わせる優先順位を設定することが出来ます。

また、各優先順位の表示の有効/無効の設定をすることも出来ます。

表 7 に DU のプレーン優先順位及び有効/無効設定で使用するレジスタを示します。

表 7 優先順位，表示の有効/無効設定

機能	レジスタ名
優先順位設定	表示プレーン優先順位レジスタ (DPPR)
表示の有効/無効	

2.3.5 下地色の設定

DU は表示サイズまたは透過色などにより表示するプレーンのない場合に表示する色を設定することが出来ます。

表 8 に DU の下地色設定で使用するレジスタを、図 8 に下地色の表示を示します。

表 8 下地色設定

機能	レジスタ名
下地色設定	下地色レジスタ(BPOR)

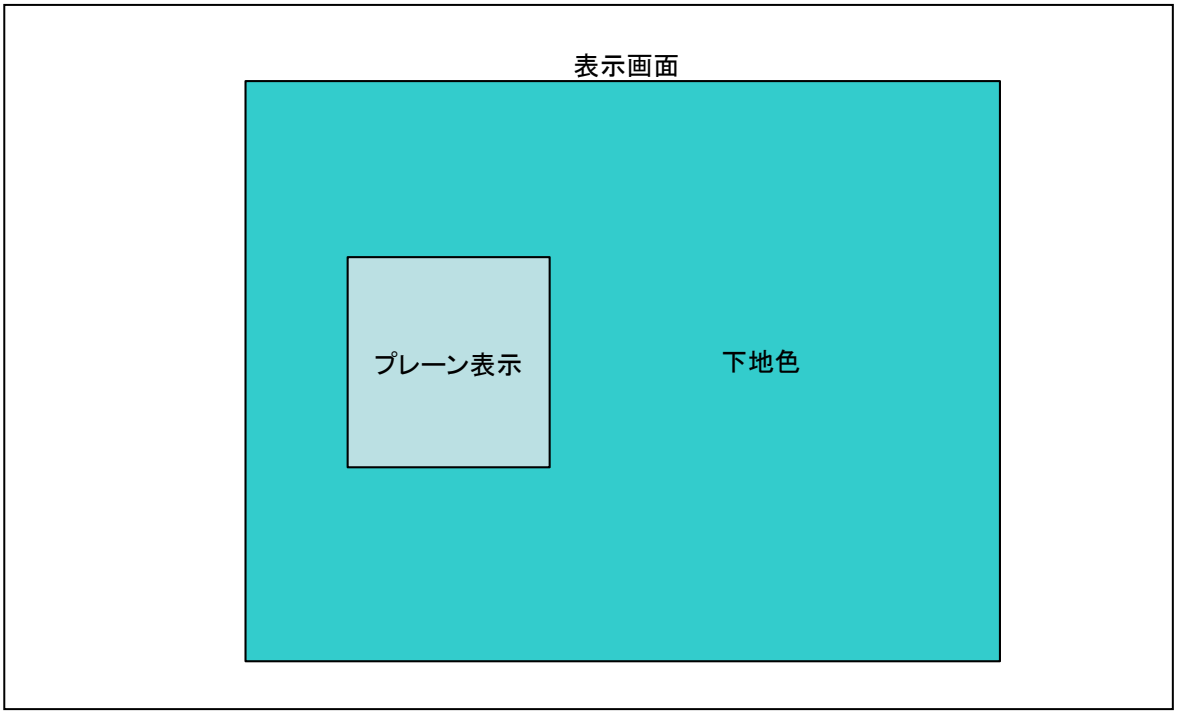

図 8 下地色表示

2.3.6 同期信号の設定

表示させる対象(液晶ディスプレイ等)の図 4 の(A)~(E)，(a)～(e)の仕様に合せて，図 9 の同期信号パラメータを設定します。

また，表 9 に DU の表示タイミング設定に使用するレジスタを示します。

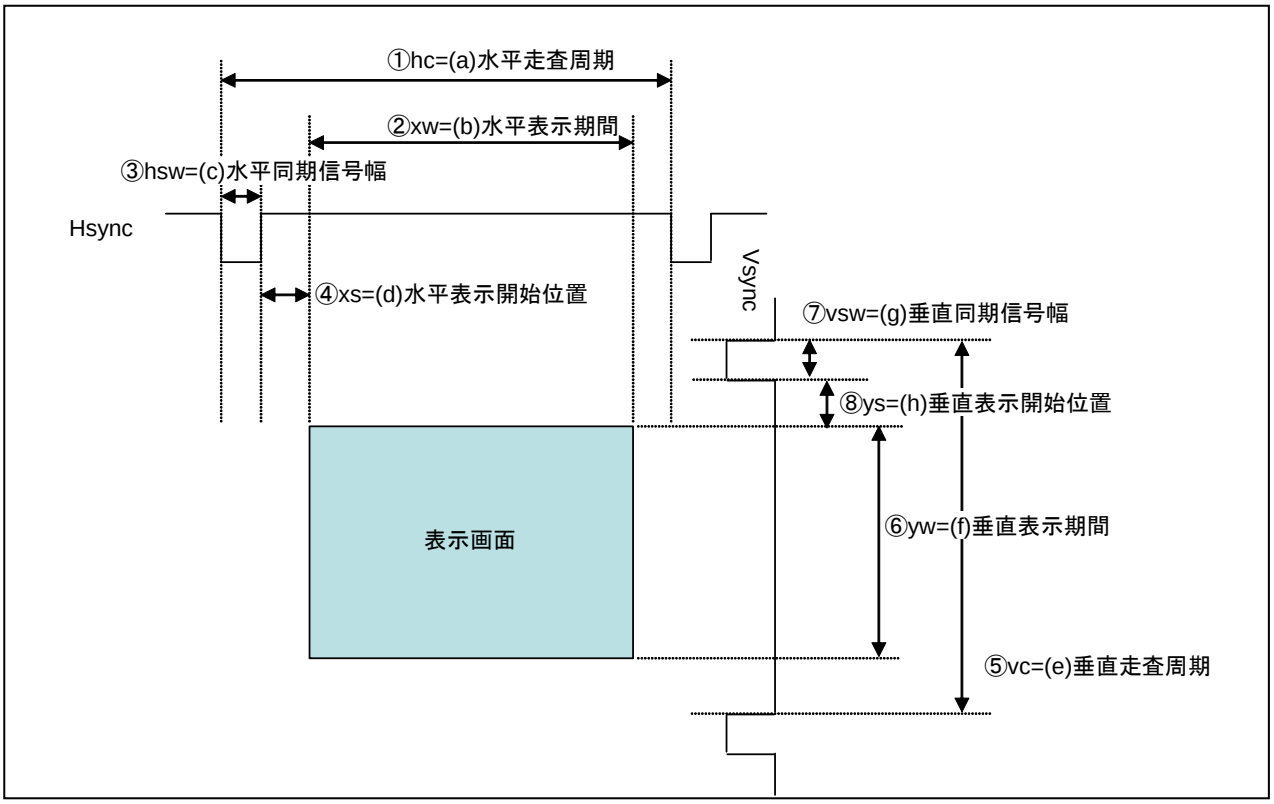

図 9 同期信号の構成

表 9 表示タイミング設定レジスタ

レジスタ名称	ビット名称	同期方式(マスターモード)
水平表示開始位置レジスタ(HDSR)	HDS	hsw+xs-19
水平表示終了位置レジスタ(HDER)	HDE	hsw+xs-19+xw
垂直表示開始位置レジスタ(VDSR)	VDS	ys-2
垂直表示終了位置レジスタ(VDER)	VDE	ys-2+yw
水平同期パルス幅レジスタ(HSWR)	HSW	hsw-1
水平走査周期レジスタ(HCR)	HC	hc-1
垂直同期位置レジスタ(VSPR)	VSP	vc-vsw-1
垂直走査周期レジスタ(VCR)	VC	vc-1

設定についての注意事項

hsw+xs+xw<hc+18(10 進)を満足するように設定してください。
vsw+ys+yw<vc を満足するように設定してください。
VDS は 1 以上にしてください。
HDE<HC を満足するように設定してください。

2.3.7 ドットクロックの設定

表示させる対象(液晶ディスプレイ等)の AC 特性に従い，ドットクロックを設定します。ドットクロックはソースクロックを分周して生成します。

- ソースクロック: DU クロック(DUck), DCLKIN 端子から入力する外部入力クロックの 2 種類から選択します。DU クロックを選択する場合は，表示拡張機能許可レジスタ(DEFR)の DCKE ビットを'1'にセットしてください。分周率が整数比(1/1，1/2，1/3，…)でのみ設定可能なので，所望のドットクロックの整数倍の周期のソースクロックを選択してください。また，以下の制限事項があります。
  — DU クロック選択時，ドットクロック生成回路で生成される分周後ドットクロックの周波数が 50MHz 以下となるようにしてください。
  — 外部入力クロックは 50MHz 以下の周波数としてください。
- 分周率は 1/1 分周で使用するか，1/1 分周以外(1/2，1/3，…1/32)を選択します。1/16 分周以降を使用する場合は，表示拡張機能許可レジスタ(DEFR)の DCKE ビットを'1'にセットしてください。

表 10 に DU のクロック設定で使用するレジスタを示します。

表 10 クロック設定レジスタ

機能	レジスタ名	ビット名
DU クロック設定	周波数制御レジスタ(FRQCR1)	S3FC3-0
DUck 選択許可 1/17～1/32 分周選択許可	表示拡張機能許可レジスタ(DEFR)	DCKE
ソースクロック選択 ドットクロック出力許可 ドットクロック分周比設定	外部同期制御レジスタ(ESCR)	DCLKSEL DCLKDIS FRQSEL

2.3.8 表示インタフェースの設定

表示させる対象(液晶ディスプレイ等)の仕様により，出力モード，信号選択，信号の極性を設定します。

- 出力モード: マスターモードと TV 同期式モードを選択します。本アプリケーションノートではマスターモードを選択した際の設定内容を示します。
- 信号選択: VSYNC，ODDF，DISP，CSYNC の信号を選択します。
  - VSYNC: VSYNC 端子に VSYNC，CSYNC の出力を選択します。本アプリケーションノートでは VSYNC を選択しています。
  - ODDF: ODDF 端子に ODDF，CLAMP の出力を選択します。本アプリケーションノートでは使用しない為，初期値のままです。
  - DISP: DISP 端子に DISP，CSYNC，DE の出力を選択します。本アプリケーションノートでは DISP を選択しています。
  - CSYNC: HSYNC 端子に CSYNC，HSYNC の出力を選択します。本アプリケーションノートでは HSYNC を選択しています。
- 信号極性: DISP，HSYNC，VSYNC の極性を選択します。
  - DISP: 表示期間の極性を選択します。本アプリケーションノートでは High を選択しています。
  - HSYNC: 水平同期信号の極性を選択します。本アプリケーションノートでは Low を選択しています。
  - VSYNC: 垂直同期信号の極性を選択します。本アプリケーションノートでは Low を選択しています。

各設定の詳細は，「SH7785 ハードウェアマニュアル(RJJ09B0285)19 章ディスプレイユニット(DU)」の「19.3.1 表示システム制御レジスタ(DSYSR)」または「19.3.2 表示モードレジスタ(DSMR)」をご参照ください。

注：表示するメモリまたはモニタから画像がはみ出した場合は画像が表示されません。

また、表示位置をずらして表示メモリ外を表示した場合は画像データは保障されません(ゴミデータが表示されます)

表 11 に表示インタフェースの設定で使用する DU のレジスタを示します。

表 11 表示インタフェースの設定

機能	レジスタ名
出力モード	表示システム制御レジスタ(DSYSR)
信号選択 信号極性	表示モードレジスタ(DSMR)

3. 応用例の説明

DU を使用して液晶ディスプレイに画像を表示する為の参考例として, 端子接続例と設定例を説明します。

3.1 液晶ディスプレイの仕様

本アプリケーションノートで使用する液晶ディスプレイの仕様を示します。使用する液晶ディスプレイは, VESA 規格の液晶ディスプレイで，DVI ケーブルにて接続します。DVI での出力は RGB->DVI 変換 IC(TI 製 TFP410)を使用しています。

3.1.1 DVI変換仕様

表 12 に本アプリケーションノートで使用する TFP410 の仕様を示します。

表 12 TFP410 の仕様(データシートから抜粋)

項目	仕様
解像度	VGA～UXGA(本応用例では WVGA で使用)
周波数	25MHz～125MHz(本応用例では 33MHz で使用)
入力信号	CMOS，R・G・B 各 8 ビットデジタル

表 13 に本アプリケーションノートで使用する TFP410 の端子機能を示します。

表 13 TFP410 の端子機能

項目	仕様
IDCK+	ドットクロック入力
HSYNC	水平同期信号入力
VSYNC	垂直同期信号入力
DE	表示開始信号入力
D23-16(R7-0)	赤データ信号入力(8 ビット，MSB:R7，LSB:R0)
D15-8(G7-0)	緑データ信号入力(8 ビット，MSB:G7，LSB:G0)
D7-0(B7-0)	青データ信号入力(8 ビット，MSB:B7，LSB:B0)
TXC+，TXC-	差動クロックペア出力
TX0+，TX0-	差動データ 0 ペア出力
TX1+，TX1-	差動データ 1 ペア出力
TX2+，TX2-	差動データ 2 ペア出力

3.1.2 DVIコネクタ端子機能

表 14 に本アプリケーションで使用する DVI コネクタの端子機能を示します。

表 14 DVI コネクタ端子機能

項目	仕様
TXC+，TXC-	差動クロックペア
TX0+，TX0-	差動データ 0 ペア
TX1+，TX1-	差動データ 1 ペア
TX2+，TX2-	差動データ 2 ペア

3.1.3 インタフェースタイミング

表 15 に本アプリケーションノートで設定している WVGA のタイミングを示します。

また，図 9 同期信号の構成との対応も合せて示します。

表 15 WVGA タイミング

項目		値	単位	図 9 との対応
CLK	周波数	37.5	MHz	
HSYNC	TOTAL	1258	CLK	水平同期周期(hc)
	バックポーチ	110		水平表示開始位置(xs)
	フロントポーチ	220		hc-(hsw+xs+xw)
	有効表示期間	800		水平表示期間(xw)
	同期信号幅	128		水平同期信号幅(hsw)
VSYNC	TOTAL	525	HSYNC	垂直同期周期(yc)
	バックポーチ	35		垂直表示開始位置(ys)
	フロントポーチ	5		vc-(vsw+ys+yw)
	有効表示期間	480		垂直表示期間(yw)
	同期信号幅	5		垂直同期信号幅(ysw)

表 16 にレジスタに設定する値を算出した例を示します。

表 16 WVGA タイミングレジスタ設定

機能	レジスタ名	ビット名	値	設定値例
水平タイミング設定	水平表示開始位置レジスタ (HDSR)	HDS	hsw+xs-19 =128+110-19 =219	0xDB
	水平表示終了位置レジスタ (HDER)	HDE	hsw+xs-19+xw =128+110-19+800 =1019	0x3FB
	水平同期パルス幅レジスタ (HSWR)	HSW	hsw-1 =128-1 =127	0x7F
	水平走査周期レジスタ(HCR)	HC	hc-1 =1258-1 =1257	0x4E9
垂直タイミング設定	垂直表示開始位置レジスタ (VDSR)	VDS	ys-2 =5-2 =3	0x3
	垂直表示終了位置レジスタ (VDER)	VDE	ys-2+yw =5-2+480 =483	0x1E3
	垂直同期位置レジスタ(VSPR)	VSP	vc-vsw-1 =525-5-1 =519	0x207
	垂直走査周期レジスタ(VCR)	VC	vc-1 =525-1 =524	0x20c

表 17 に本アプリケーションノートで設定しているクロックの設定例を示します。

表 17　クロック設定

機能	レジスタ名	ビット名	値	設定値例
ドット クロック	周波数制御レジスタ 1(FRQCR1)	S3FC3-0	入力 CLK33.33MHZ 33.33*9/2 =150MHz	0x4
	表示拡張機能許可レジスタ(DEFR)	DCKE	DUck 許可	0x1
	外部同期制御レジスタ(ESCR)	DCLKSEL	ソースクロック DUck	0x1
		DCLKDIS	DCLKOUT 許可	0x1
		FRQSEL	150MHz/4 =37.5MHz	0x3

液晶ディスプレイのドットクロックの仕様が約 34MHz, 水平同期周期が 31.5KHz で実際に使用するドットクロックは 37.5MHz と少し周波数が高い為，画像が左右どちらかにずれてしまいます。画像すれを修正するには，水平同期周期を増やして水平同期周期を 31.5KHz に近い値に調整し、左右のずれはフロントポーチ，バックポーチを 1 ドット増減しながら調整します。本アプリケーションノートで使用した評価ボードでは，DVI 変換 IC に DISP 信号(表示開始信号)入力がある為、DISP 信号を使用して調整を行うことなく画面表示位置ずれを回避できます。

DISP 信号の有無は LCD モジュール，信号変換 IC(アナログ RGB、DVI 等)の仕様によります。

3.2 DVI接続回路例

3.2.1 端子接続例

図 10 に本アプリケーションにおける DVI 接続回路例を示します。

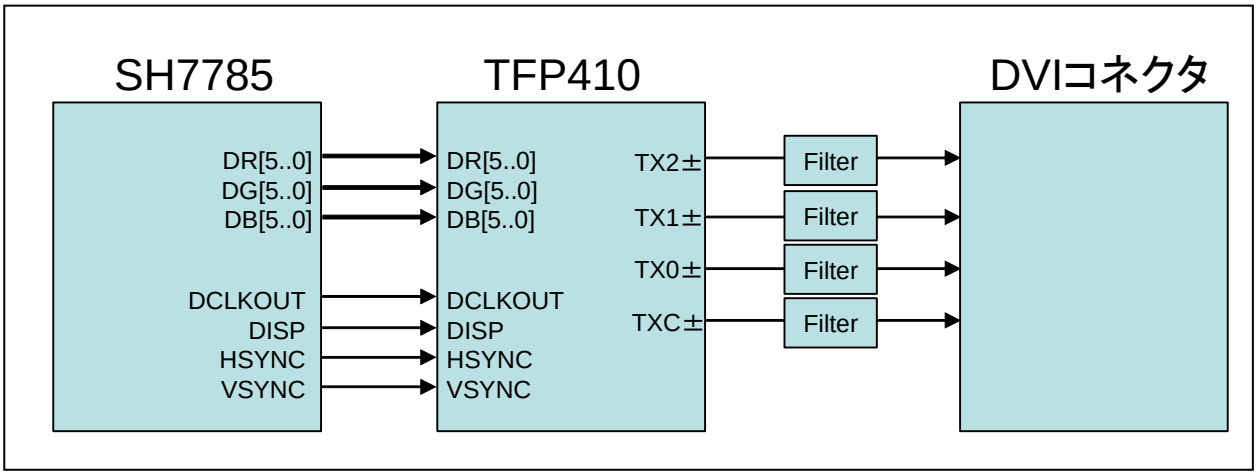

図 10 接続回路例

3.3 参考プログラムの仕様

ここでは参考プログラムの仕様と各処理のフローチャートを説明します。

3.3.1 仕様

(1)WVGA サイズ表示で 4 プレーンを使用して，5 種類の BMP ファイルを表示します。

— プレーン 1: 80x80 サイズの画像を描画。
— プレーン 2: 80x80 サイズの画像を 2 枚使用し、描画。
— プレーン 3: 80x80 サイズの画像を描画。
— プレーン 4: 100x80 サイズの画像を描画。

(2)DU を起動し、1 フレーム割込みごとに以下の処理を行います。

本アプリケーションでは画像の表示位置変更については，スクロール表示機能を使用しておらず，プレーン表示位置 X，Y レジスタを使用してモニタ上での表示位置を変更することによって，マウスカーソルと同等の機能をしたサンプルプログラムになります。

— プレーン 1: 初期表示位置はモニタ右上端。①～④を繰り返す。
- ①: モニタ右上端に画像が到達すると，モニタ表示上で左上端に移動させる為，プレーン 1 表示 X レジスタ(P1DPXR)の値に’-2’減少させる。
- ②:モニタ左上端に画像が到達すると，モニタ表示上で左下端に移動させる為，プレーン 1 表示 Y レジスタ(P1DPYR)の値に’+2’増加さる。
- ③:モニタ左下端に画像が到達すると，モニタ表示上で右下端に移動させる為，プレーン 1 表示 X レジスタ(P1DPXR)の値に’+2’増加させる。
- ④:モニタ右下端に画像が到達すると，モニタ表示上で右上端に移動させる為，プレーン 1 表示 Y レジスタ(P1DPYR)の値に’-2’減少させる。

— プレーン 2: 初期表示位置はモニタ左上端。①～④を繰り返す。20 フレーム毎にプレーン 2 モードレジスタ(P2MR)の P2DC を’1’設定してフレームバッファを切換えて画像を更新。
- ①: モニタ左上端に画像が到達すると，モニタ表示上で右上端に移動させる為，プレーン 2 表示 X レジスタ(P2DPXR)の値に’+1’増加させる。
- ②:モニタ右上端に画像が到達すると，モニタ表示上で右下端に移動させる為，プレーン 2 表示 Y レジスタ(P2DPYR)の値に’+1’増加さる。
- ③:モニタ右下端に画像が到達すると，モニタ表示上で左下端に移動させる為，プレーン 2 表示 X レジスタ(P2DPXR)の値に’-1’減少させる。
- ④:モニタ左下端に画像が到達すると，モニタ表示上で左上端に移動させる為，プレーン 2 表示 Y レジスタ(P2DPYR)の値に’+1’増加させる。

— プレーン 3: 初期表示位置はモニタ左下端。①～⑥を繰り返す。
- ①:モニタ左上端に画像が到達すると，右下端に向かって斜めに移動させる為，プレーン 2 表示 X レジスタ(P2DPXR)とプレーン 2 表示 Y レジスタ(P2DPYR)の値に’+2’ずつ増加させる。
- ②:モニタ右下端に画像が到達すると，左上端に向かって斜めに移動させる為，プレーン 2 表示 X レジスタ(P2DPXR)とプレーン 2 表示 Y レジスタ(P2DPYR)の値に’-2’ずつ減少させる。
- ③:モニタ左端に画像が到達すると，右端に向かって斜めに移動させる為，プレーン 2 表示 X レジスタ(P2DPXR)の値に’+2’増加させ，プレーン 2 表示 Y レジスタ(P2DPYR)の増減値は変化無し。
- ④:モニタ上端に画像が到達すると，下端に向かって斜めに移動させる為，プレーン 2 表示 Y レジスタ(P2DPYR)の値に’+2’増加させ，プレーン 2 表示 X レジスタ(P2DPXR)の増減値は変化無し。
- ⑤:モニタ右端に画像が到達すると，左端に向かって斜めに移動させる為，プレーン 2 表示 X レジスタ(P2DPXR)の値に’-2’減少させ，プレーン 2 表示 Y レジスタ(P2DPYR)の増減値は変化無し。
- ⑥:モニタ下端に画像が到達すると，上端に向かって斜めに移動させる為，プレーン 2 表示 Y レジスタ(P2DPYR)の値に’-2’減少させ，プレーン 2 表示 X レジスタ(P2DPXR)の増減値は変化無し。

— プレーン 4:　初期表示位置はモニタ左上端。①～⑥を繰り返す。表示される画像はモニタ上に横 8×縦 6=48 となる。

- ①:画像メモリに画像を追加。
- ②:プレーン 4 表示領域開始アドレス 0 または 1(P4DSA0，1R)に設定されているフレームバッファに対して画像メモリから DMA で転送。
- ③: 60 フレーム毎にプレーン 4 モードレジスタ(P4MR)の P4DC を'1'設定してフレームバッファを切換えて画面を更新。
- ④:モニタ上に 48 枚の画像が表示されたら，画像メモリの先頭から画像を 1 枚ずつ削除。
- ⑤:②と同じ。
- ⑥:③と同じ。

(3)(2)の処理を無限ループします。

(4)コンソールでプレーン 1～4 の表示の ON/OFF を制御します。

3.3.2 参考プログラムメインフロー

図 11 に参考プログラムのメインフローを示します。

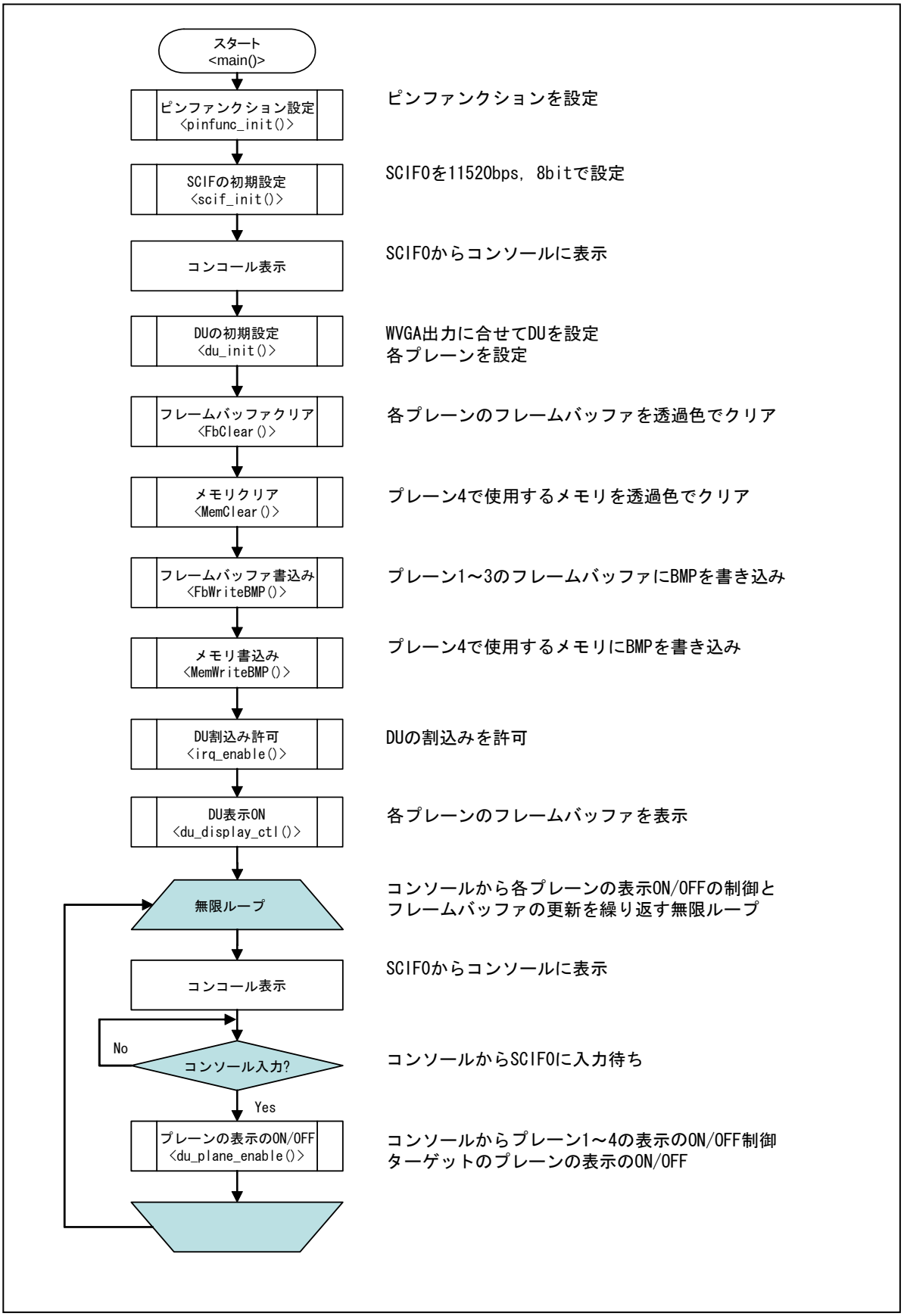

図 11 参考プログラムメインフロー

3.3.3 ピンファンクション設定

図 12 にピンファンクションの設定フローを示します。

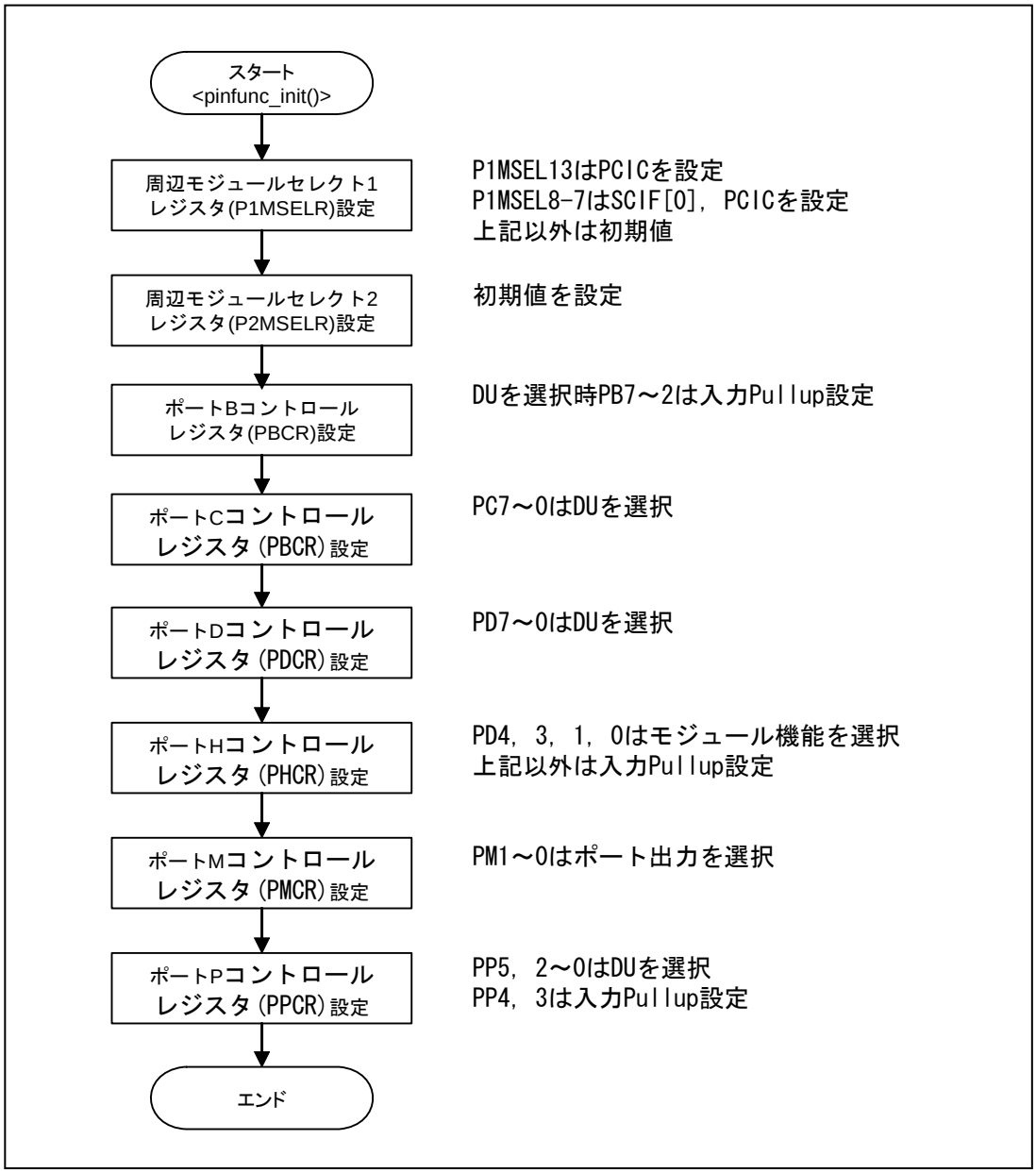

図 12 ピンファンクション設定フロー

3.3.4 SCIFの初期設定

図 13 に SCIF の初期設定フローを示します。

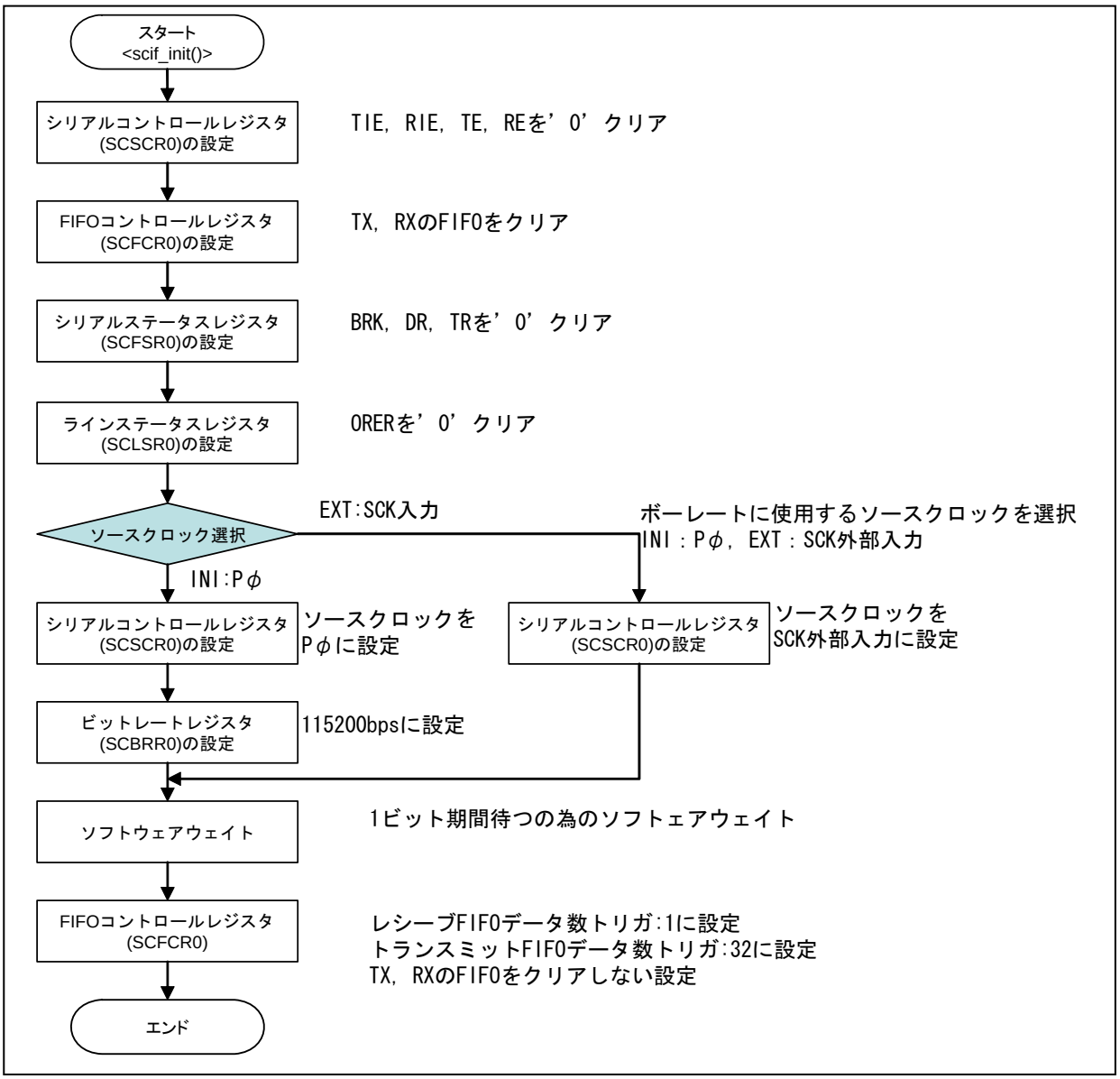

図 13 SCIF 初期設定フロー

3.3.5 DUの初期設定

図 14～16 に DU の初期設定フローを示します。

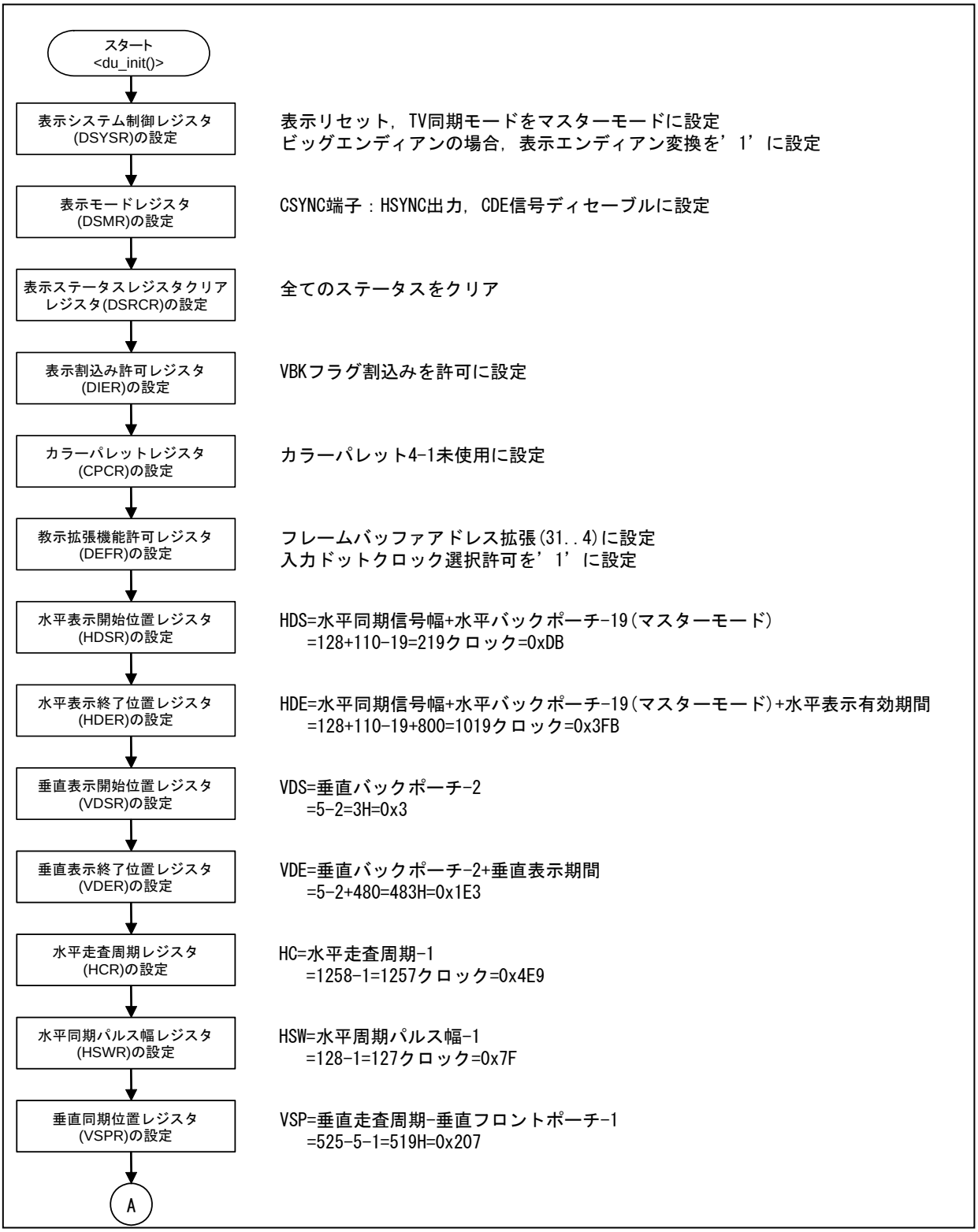

図 14　DU 初期設定例フロー1

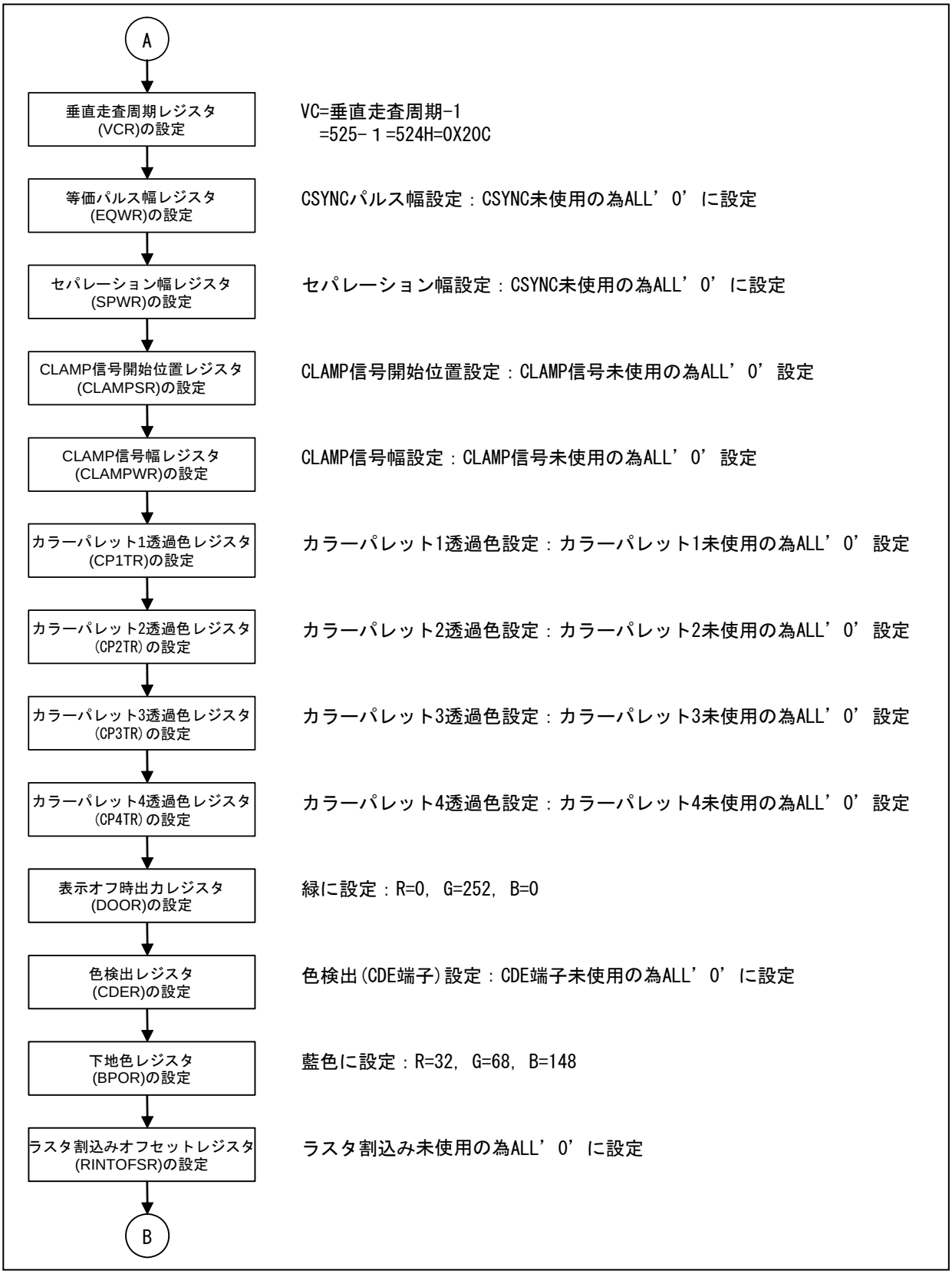

図 15　DU 初期設定例フロー2

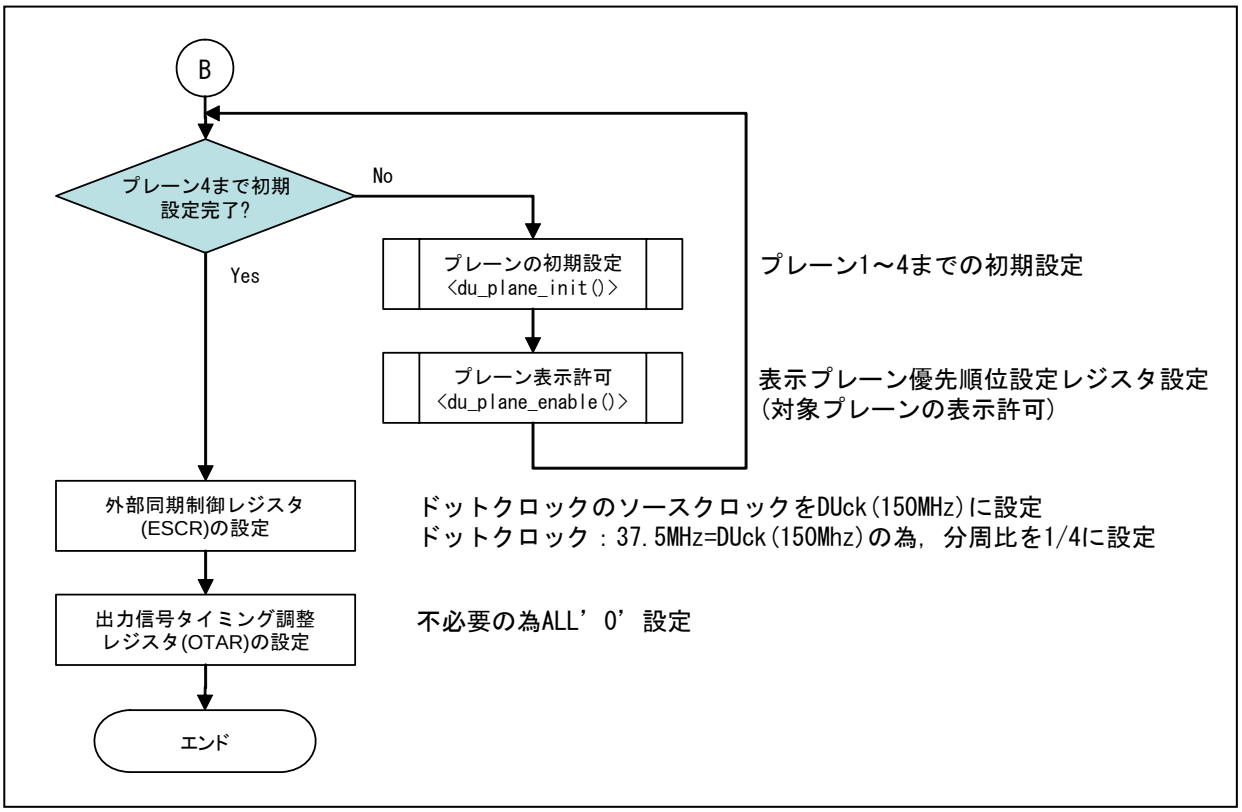

図 16　DU 初期設定例フロー3

3.3.6 プレーンの初期設定

図 17，18 に各プレーンの初期設定のフローを示します。

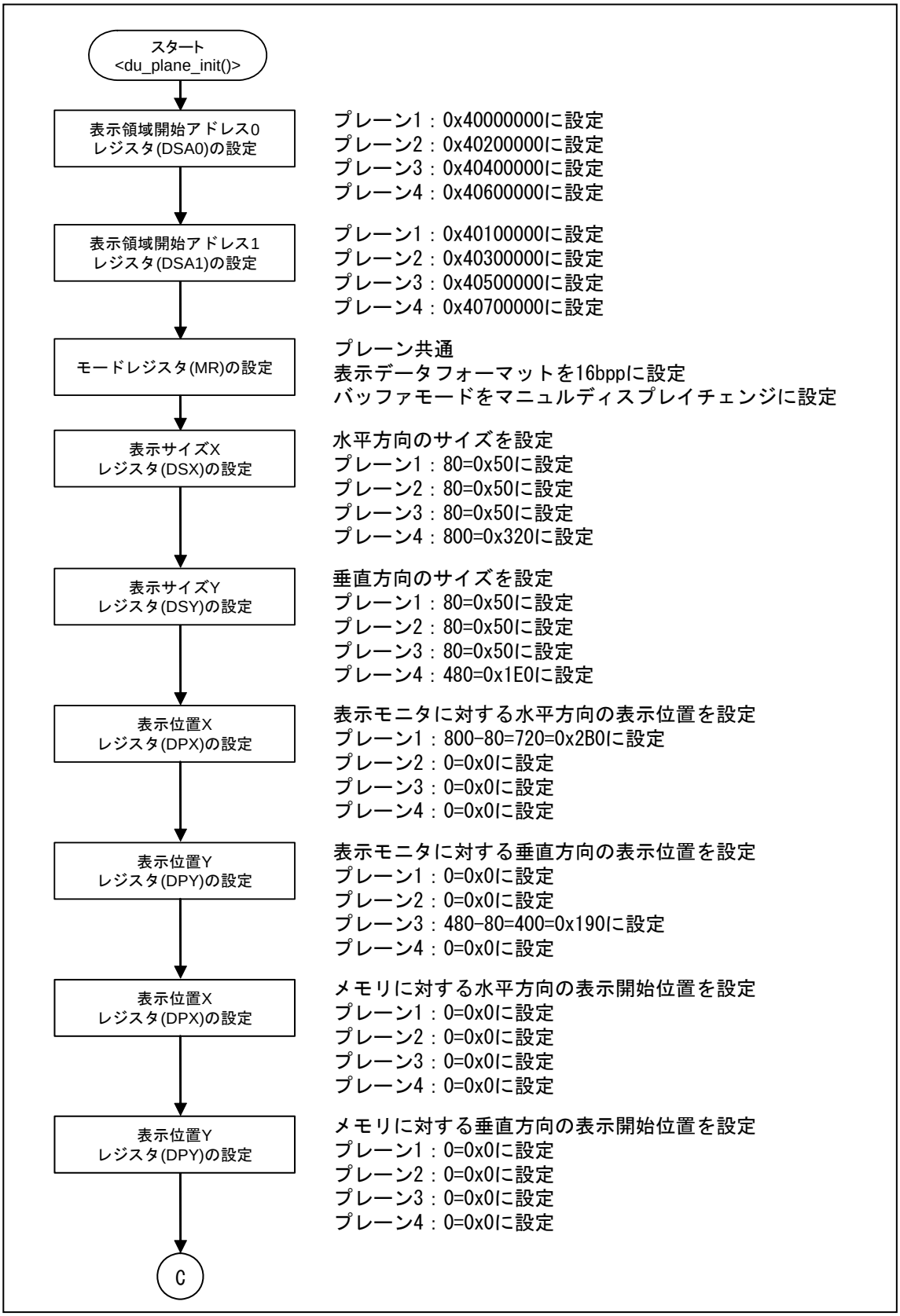

図 17　各プレーン初期設定フロー1

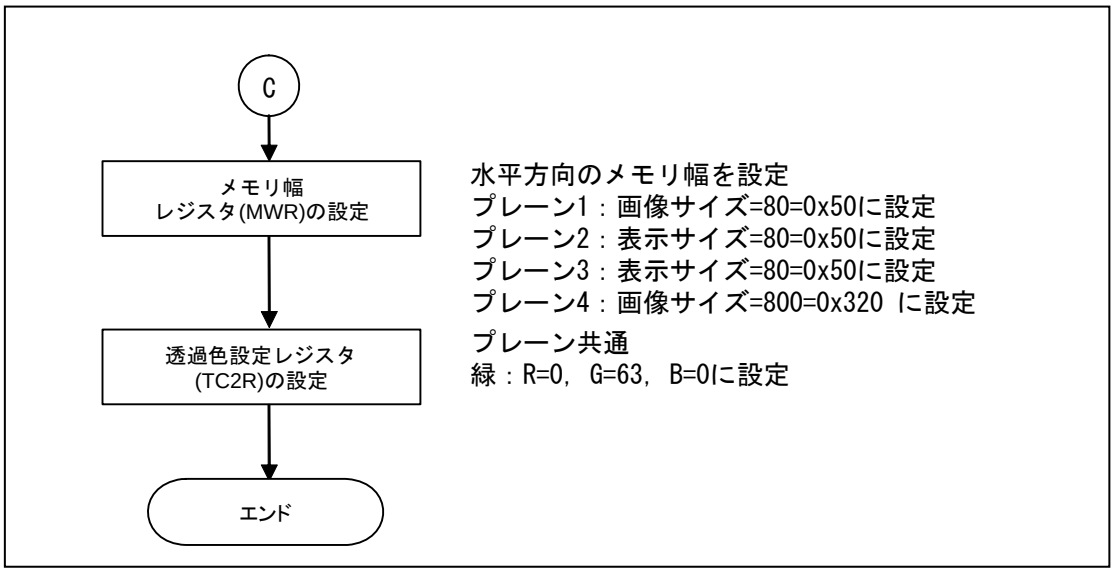

図 18　各プレーンの初期設定フロー2

3.3.7 プレーン表示ON

図 19 に各プレーンの表示 ON のフローを示します。

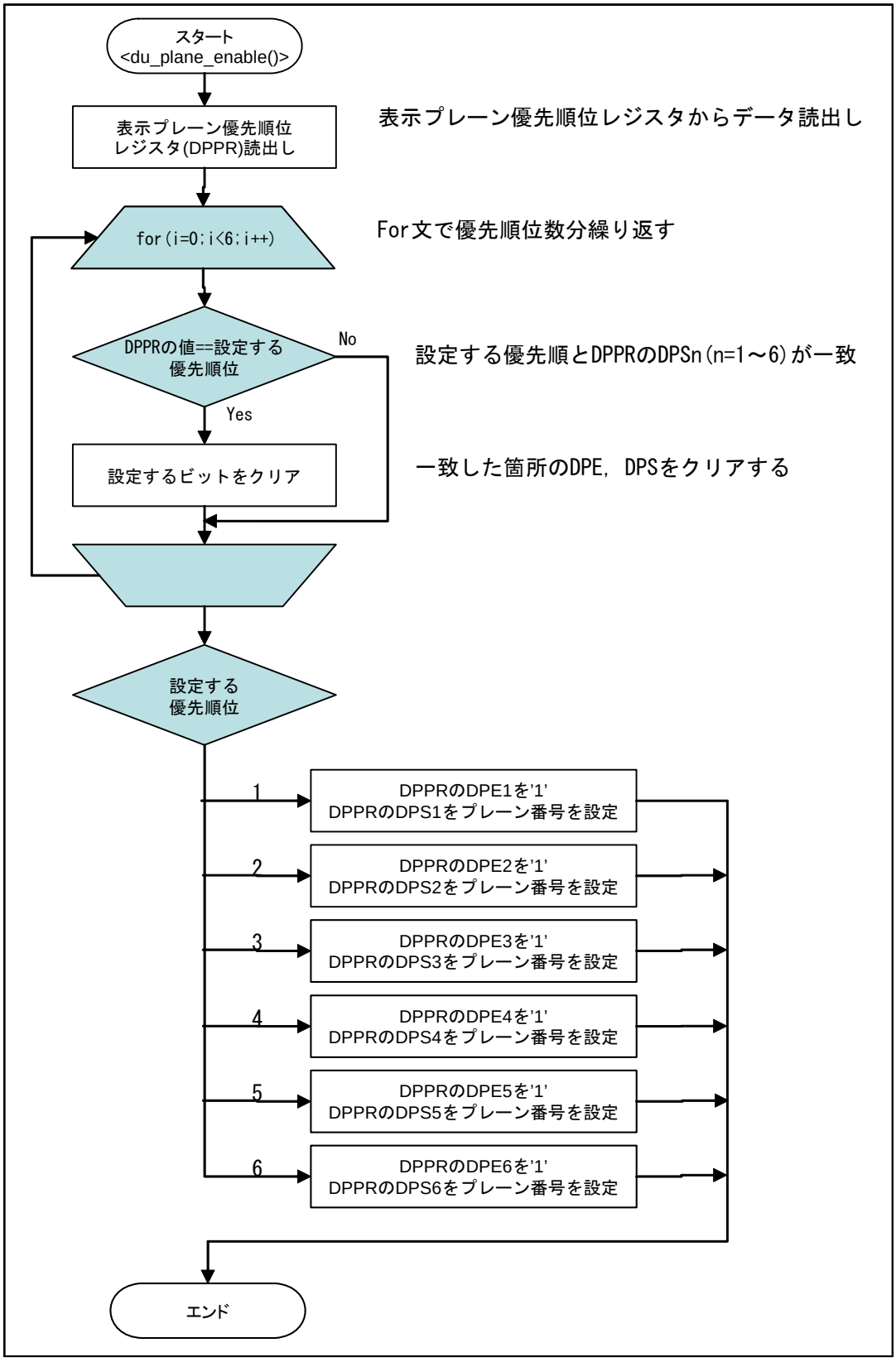

図 19 プレーン表示 ON フロー

3.3.8 プレーン表示OFF

図 20 に各プレーンの表示 OFF のフローを示します。

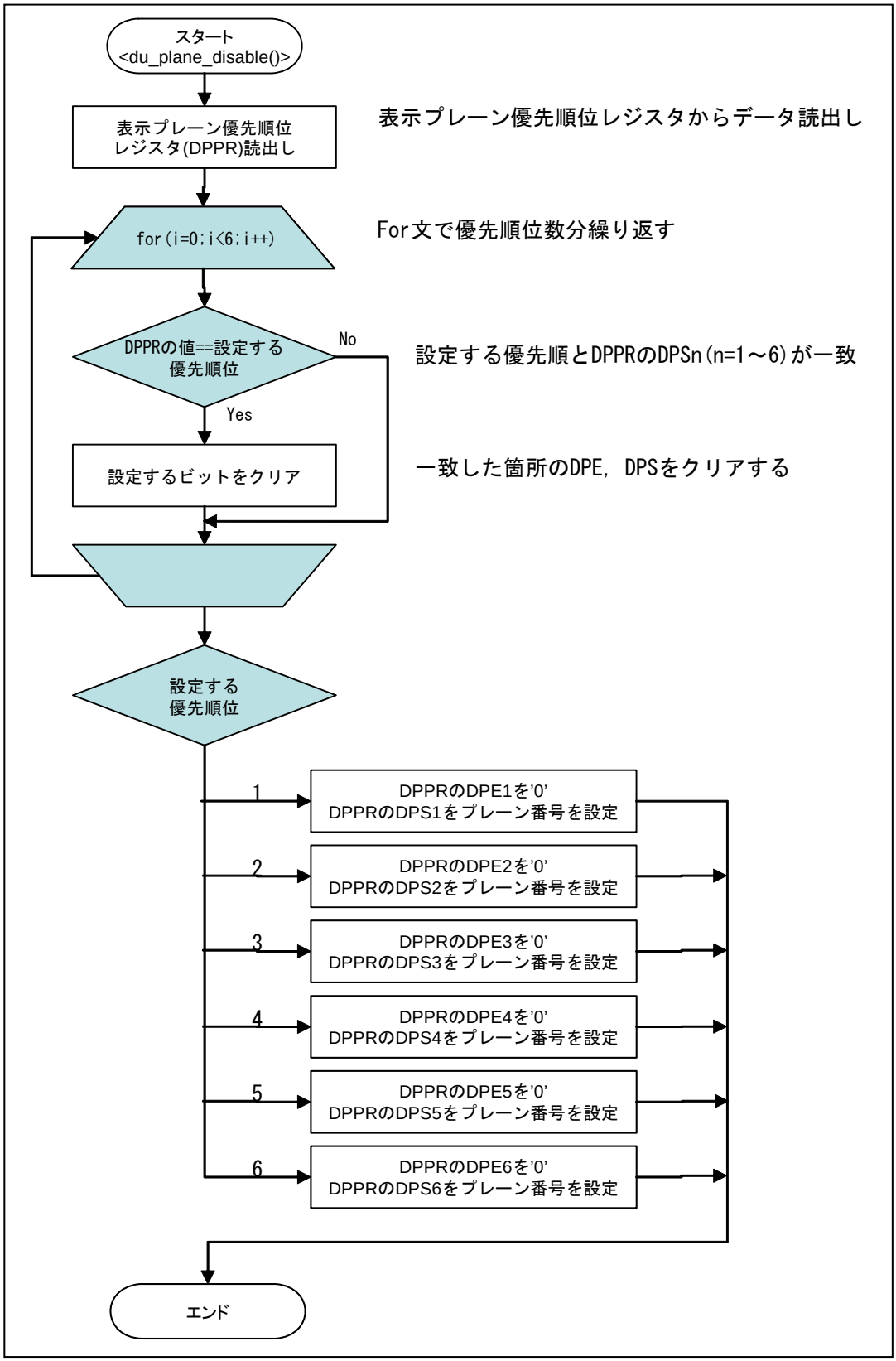

図 20 プレーン表示 OFF フロー

3.3.9 フレームバッファのクリア

図 21 にプレーン 1～3 のフレームバッファのクリアのフローを示します。

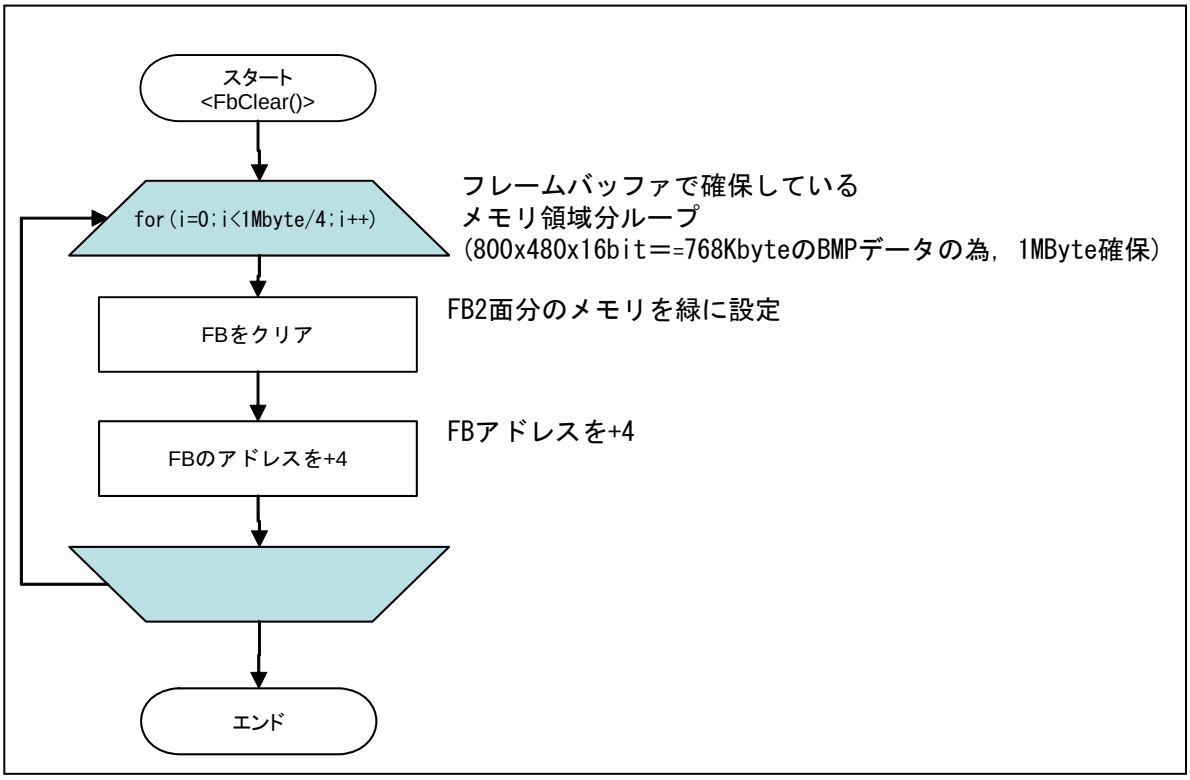

図 21 フレームバッファクリアフロー

3.3.10 メモリクリア

図 22 にプレーン 4 で使用する画像メモリのクリアフローを示します。

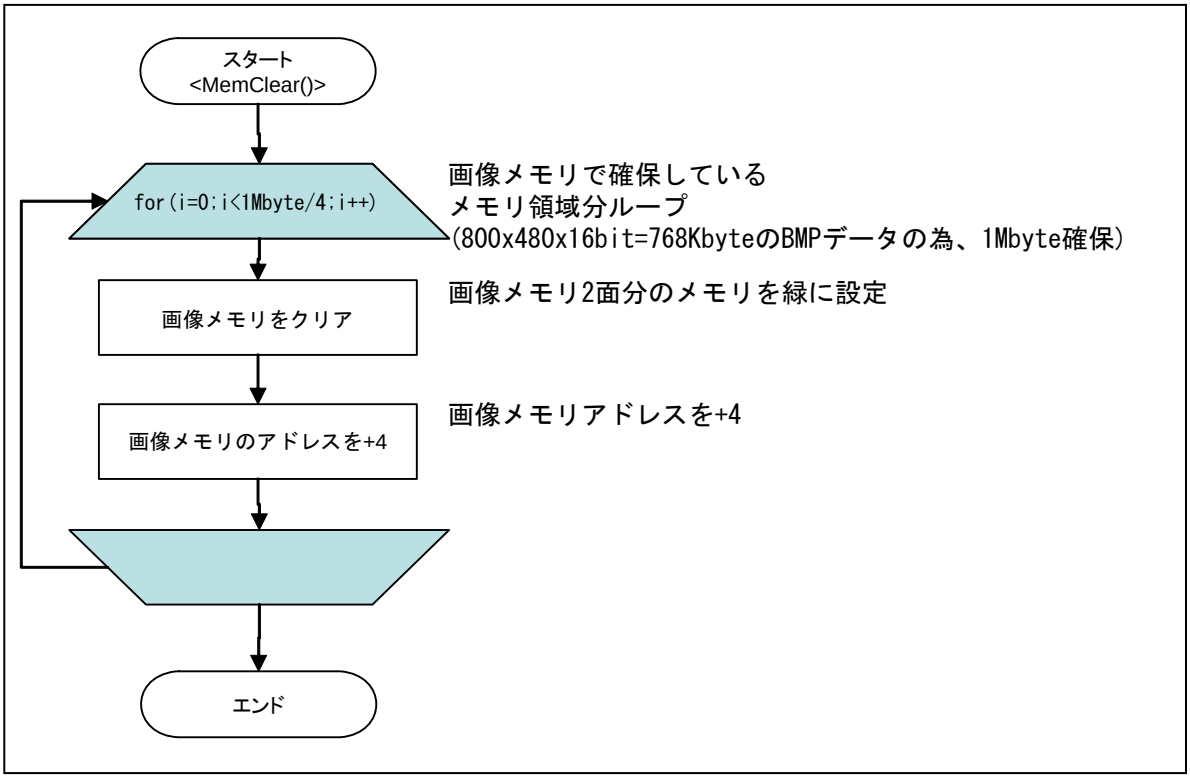

図 22 メモリクリアフロー

3.3.11 フレームバッファ書込み

図 23 にプレーン 1～3 のフレームバッファに BMP を書き込むフローを示します。

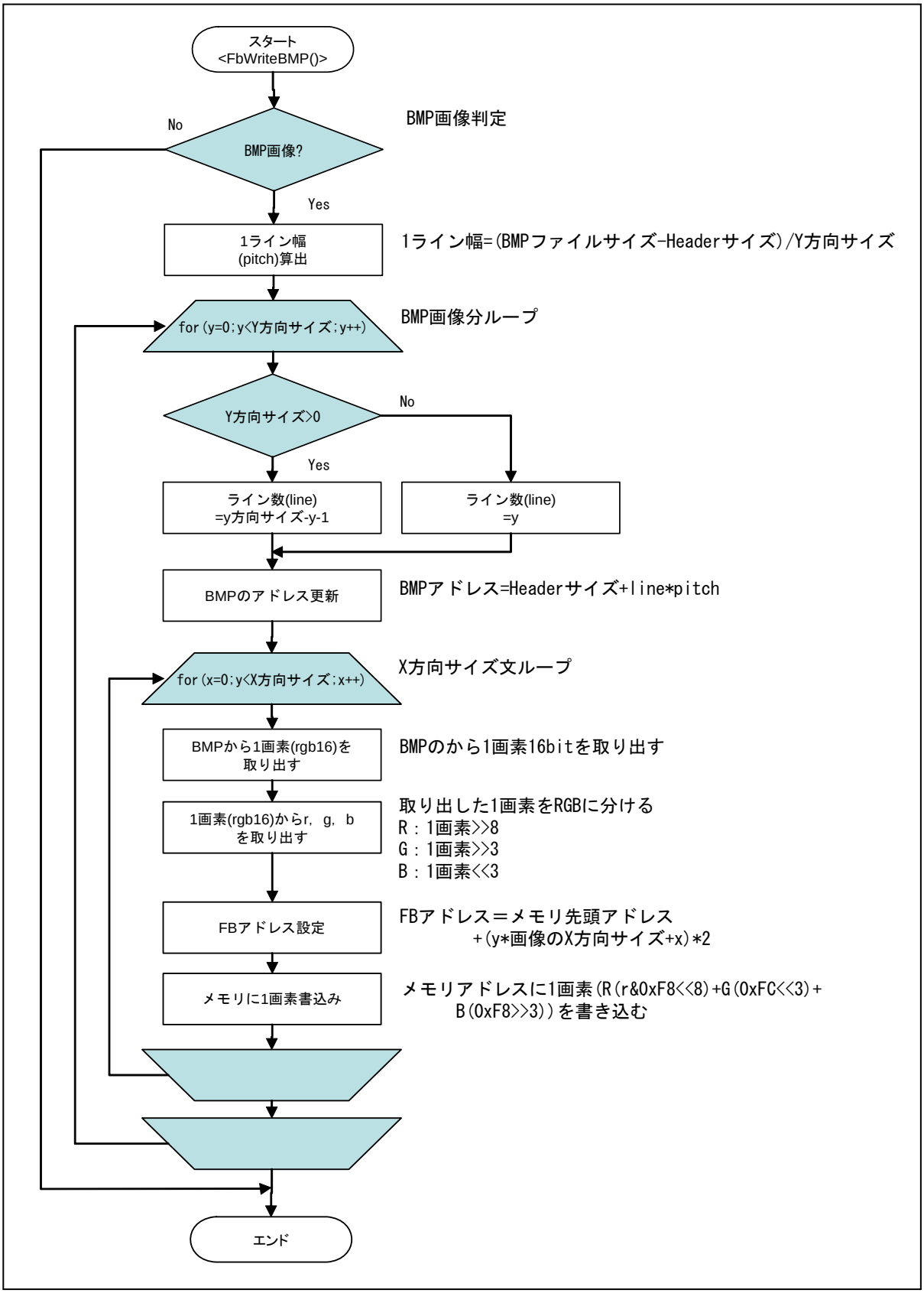

図 23 フレームバッファ BMP 書込みフロー

3.3.12 メモリ書込み

図 24 にプレーン 4 で使用するメモリに BMP を書き込むフローを示します。

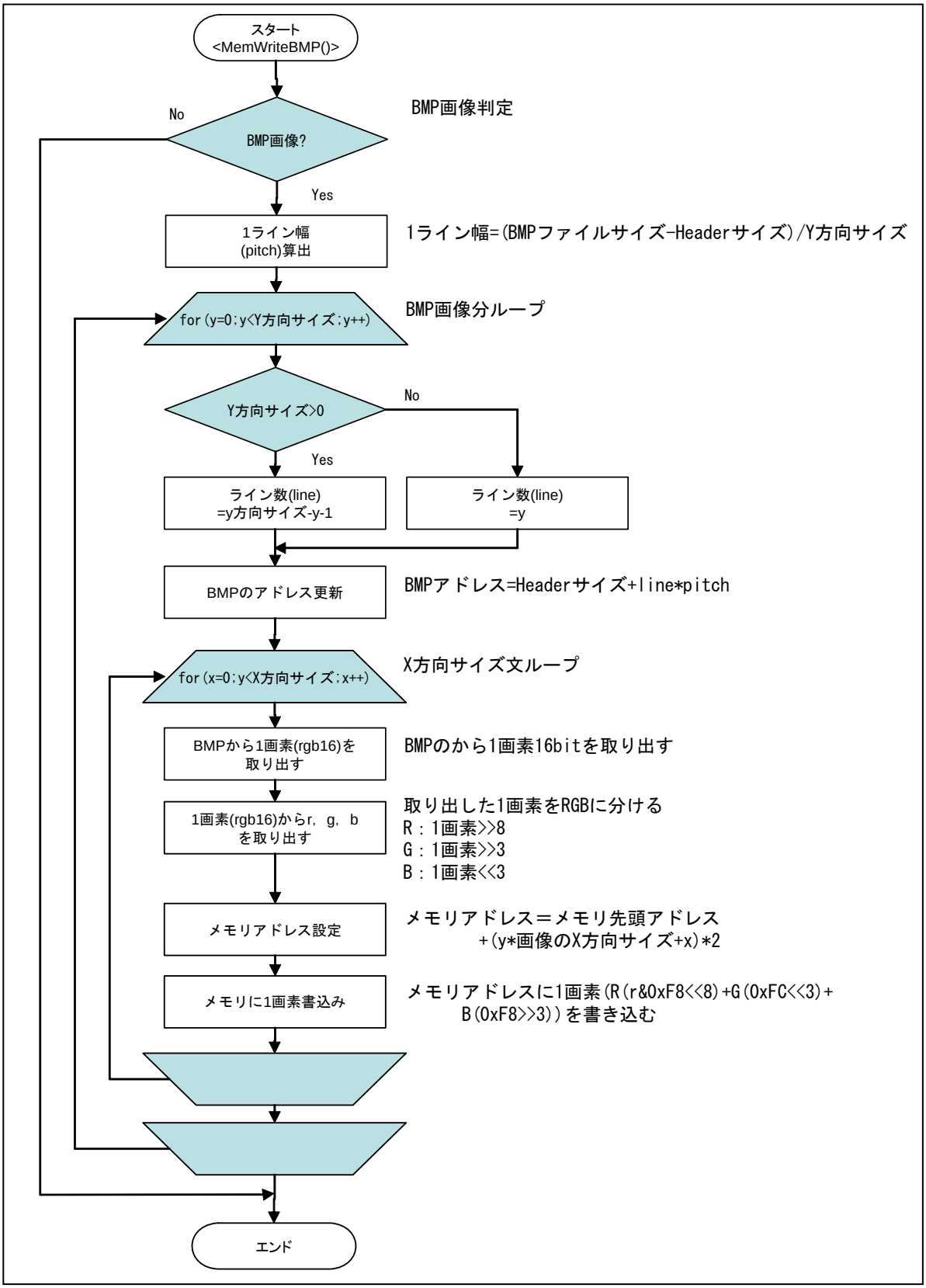

図 24 メモリ BMP 書込みフロー

3.3.13 DU表示ON/OFF

図 25 に DU の表示 ON/OFF のフローを示します。

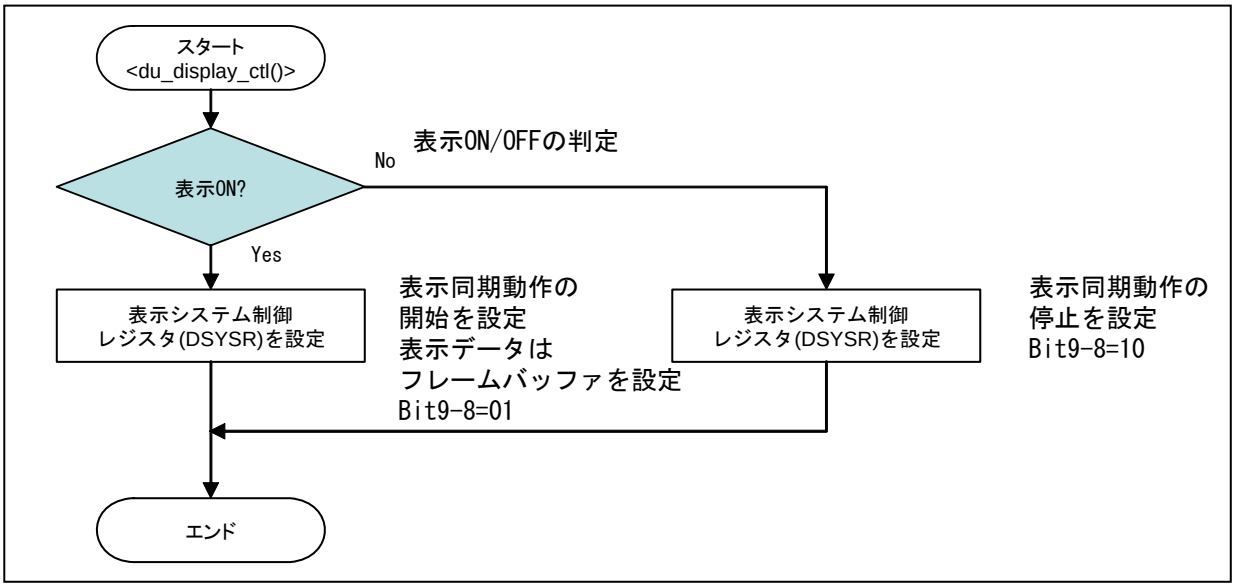

図 25 DU の表示 ON/OFF フロー

3.3.14 割込み許可

図 26 に周辺モジュール割込みの許可フローを示します。

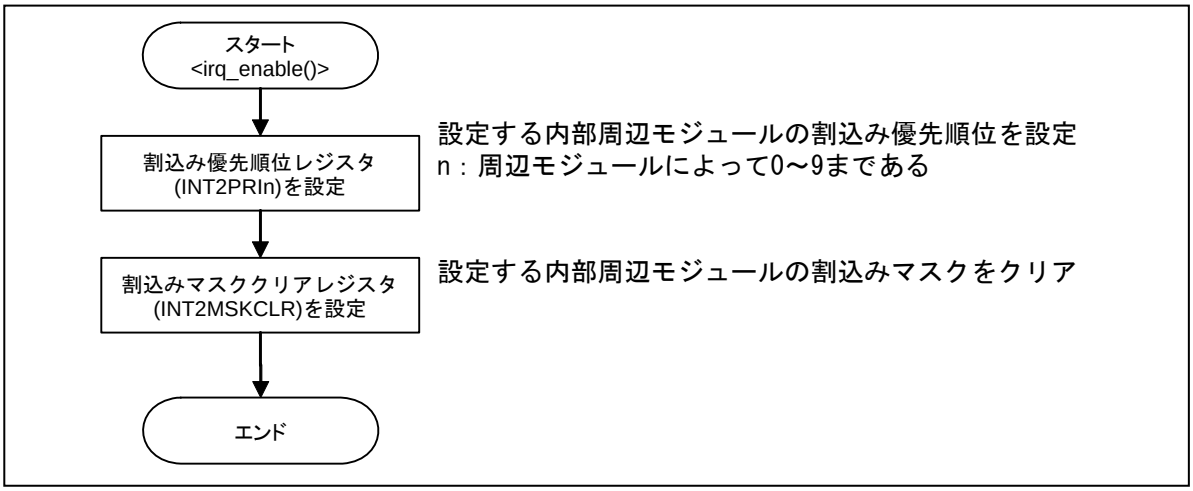

図 26 周辺モジュール割込み許可フロー

3.3.15 割込み禁止

図 27 に周辺モジュール割込みの禁止フローを示します。

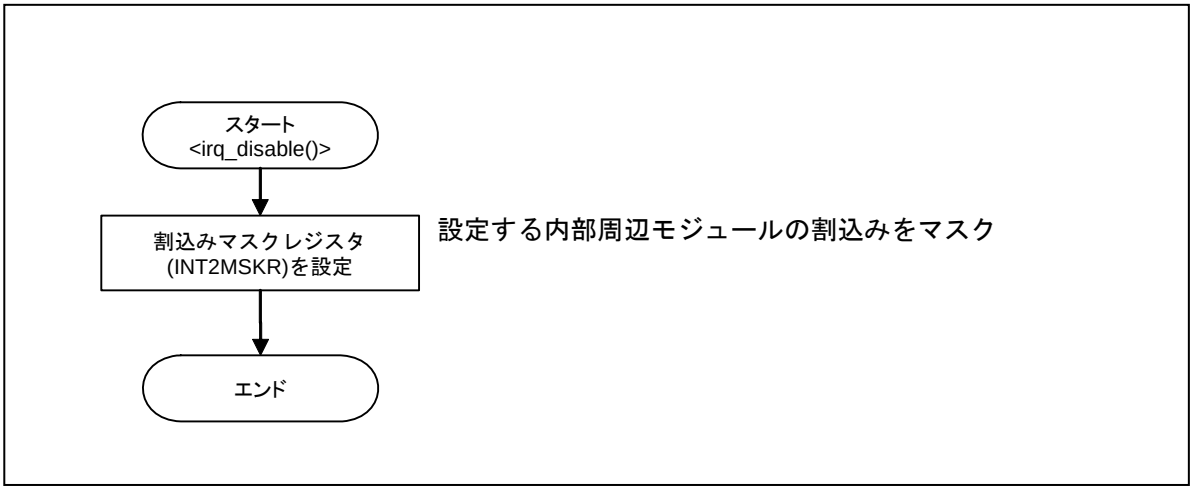

図 27 周辺モジュール割込み禁止フロー

3.3.16 DUの割込み

図 28～33 に DU の割込みのフローを示します。

本サンプルプログラムでは VBK 割込みのみ使用しており、VBK 割込み以外の割込み関数は用意していますが、使用していません。

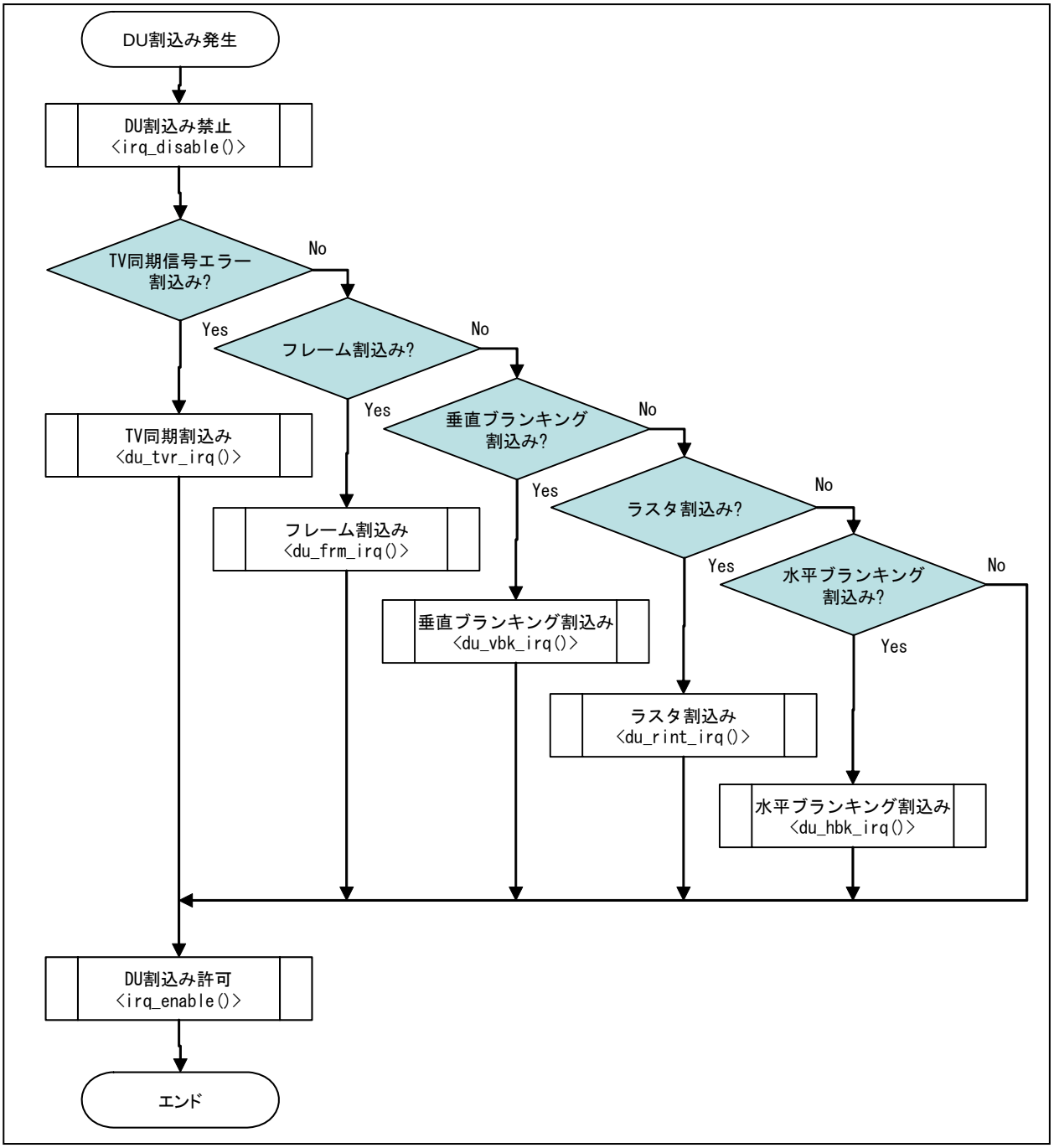

図 28 DU 割込みフロー

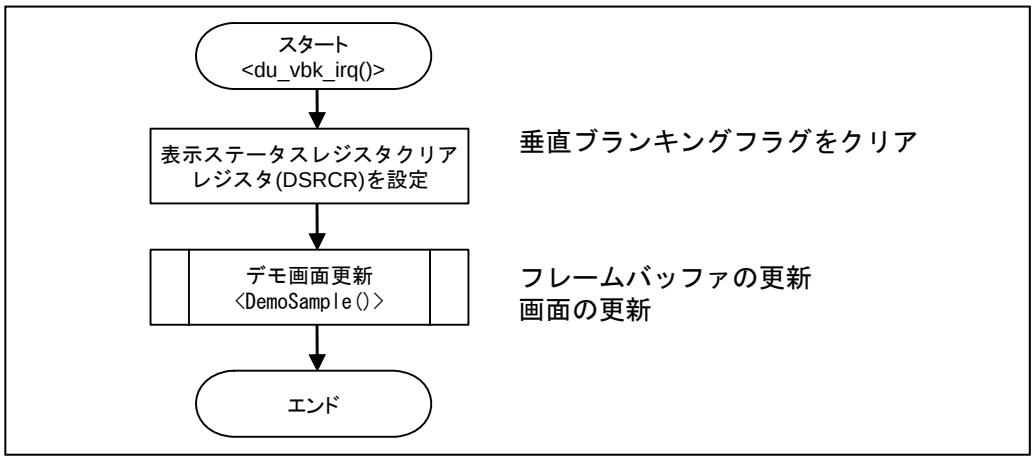

図 29　垂直ブランキングフラグ割込みフロー

図 30　TV 同期信号エラーフラグ割込みフロー

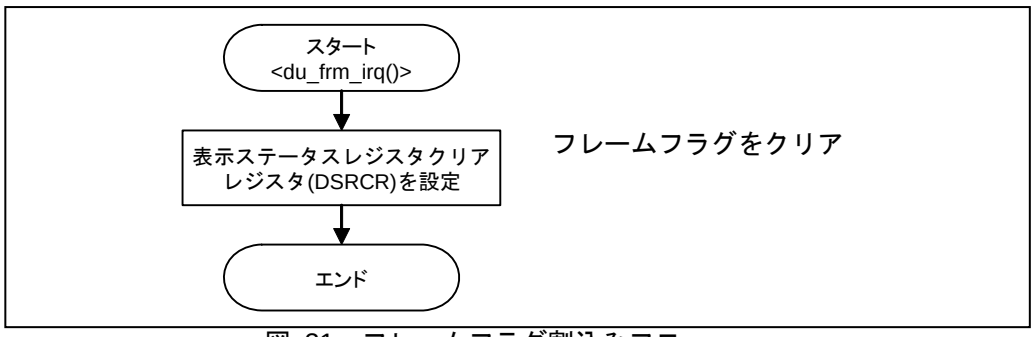

図 31　フレームフラグ割込みフロー

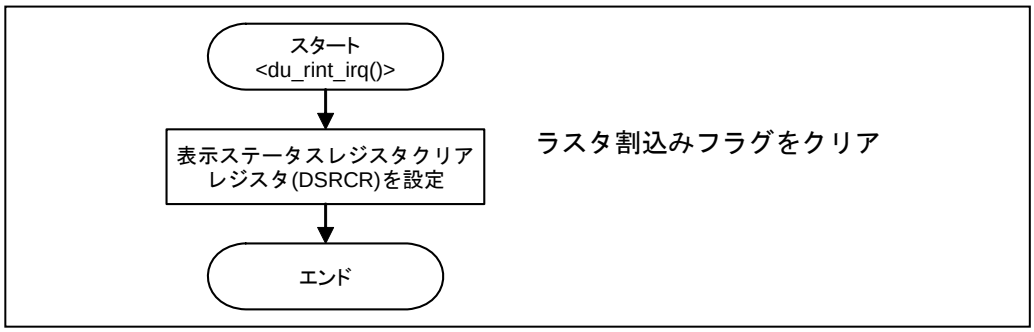

図 32　ラスタフラグ割込みフロー

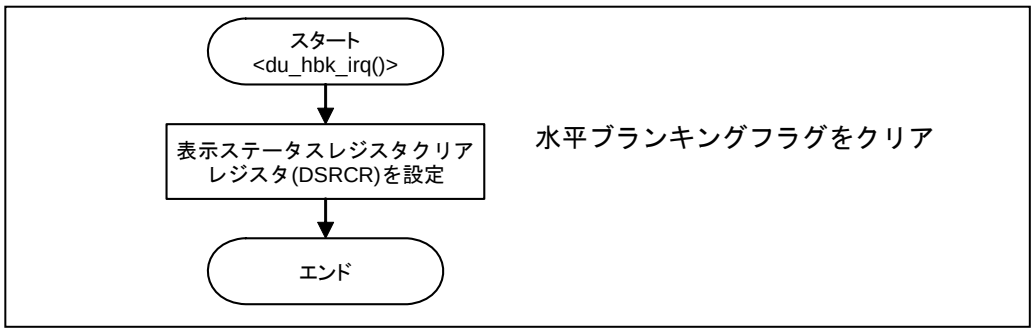

図 33　水平ブランキングフラグ割込みフロー

3.3.17 DemoSample

図 34～37 に DemoSample のフローを示します。

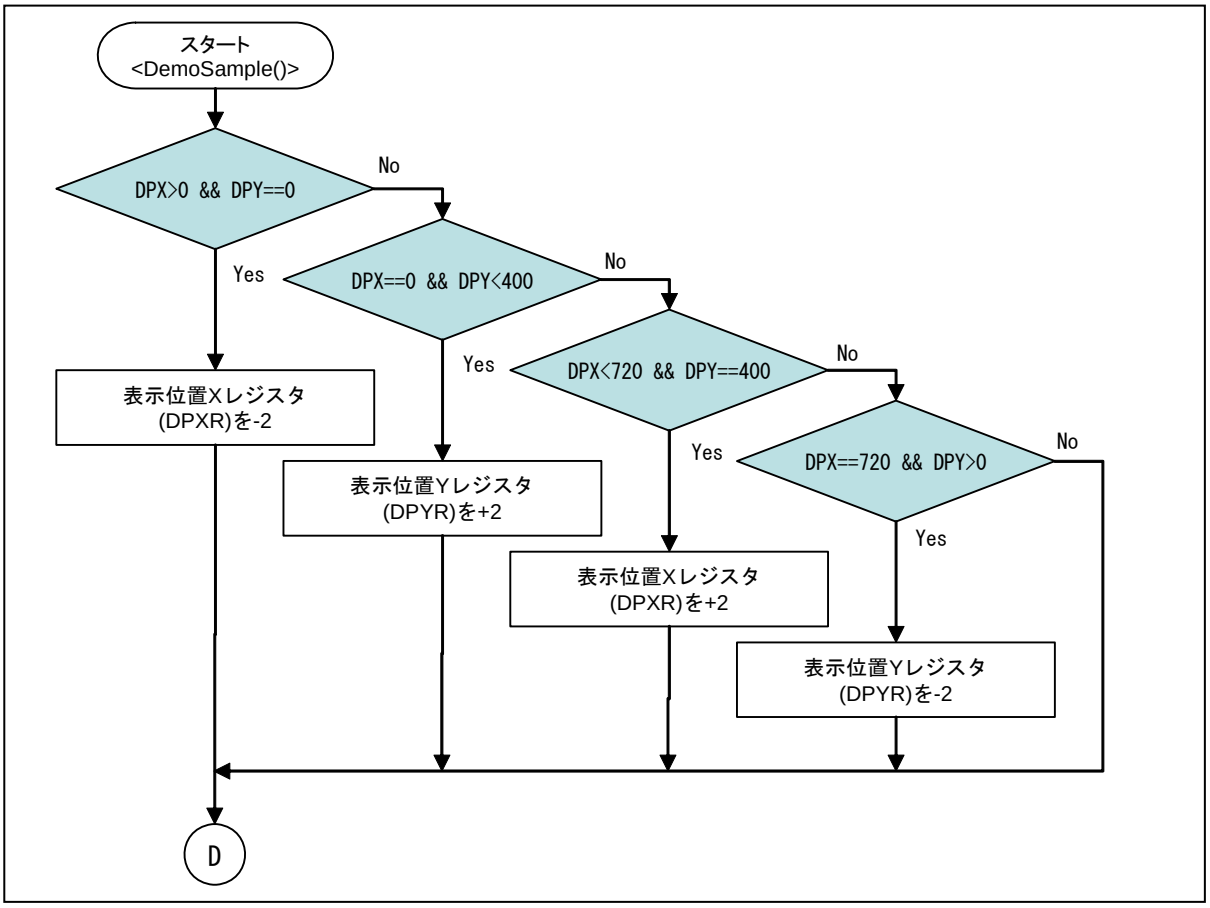

図 34 DemoDample プレーン 1 フロー

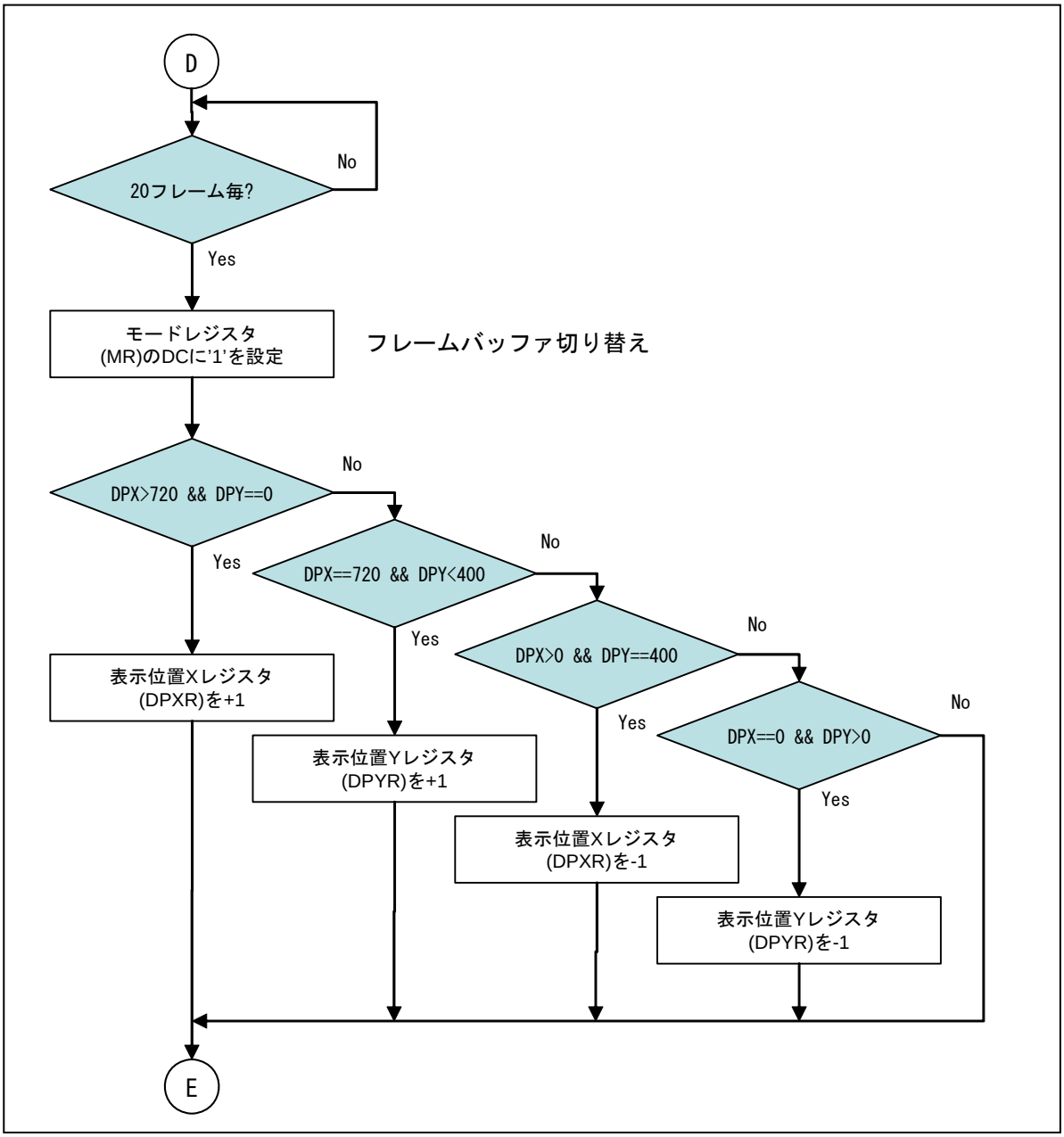

図 35　DemoSample プレーン 2 フロー

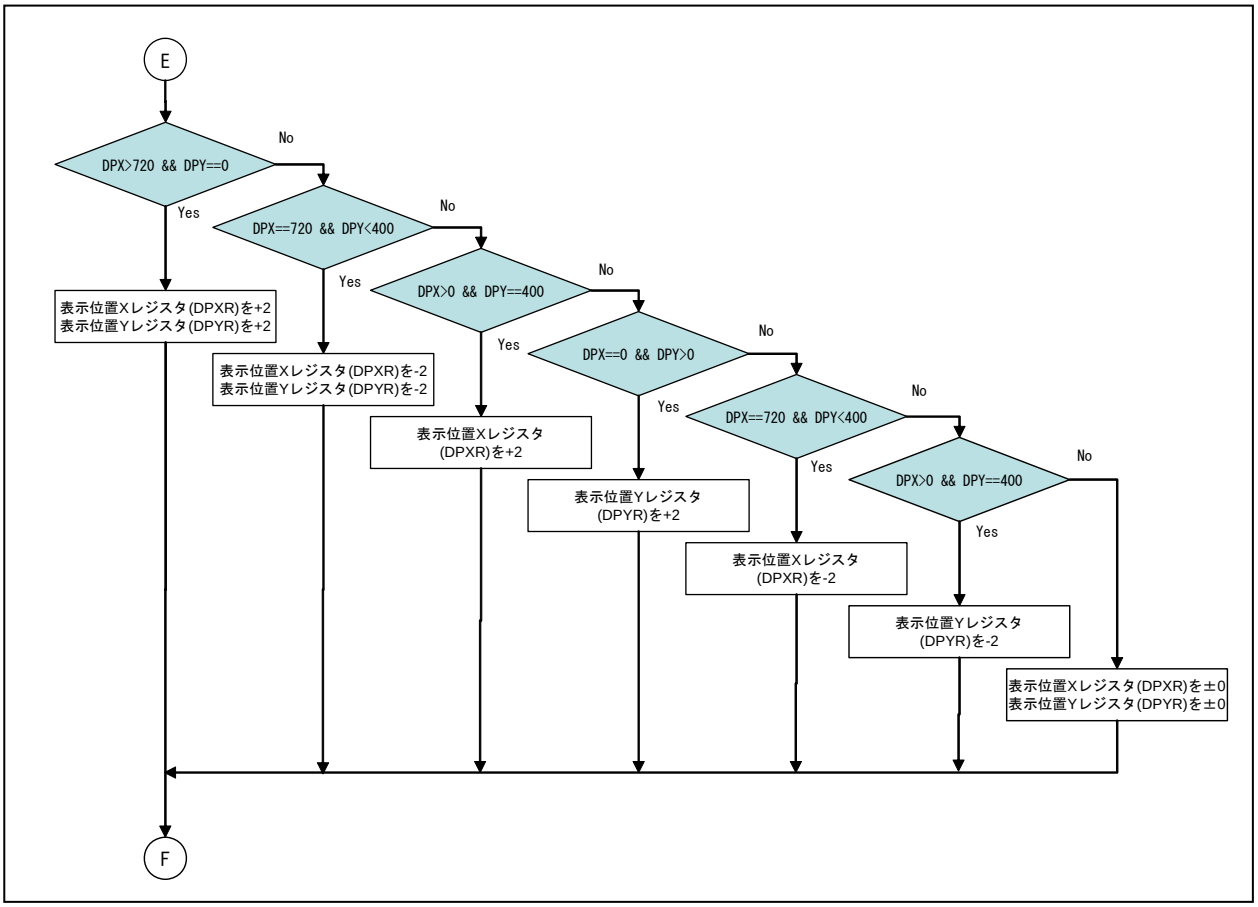

図 36　DemoSample プレーン 3 フロー

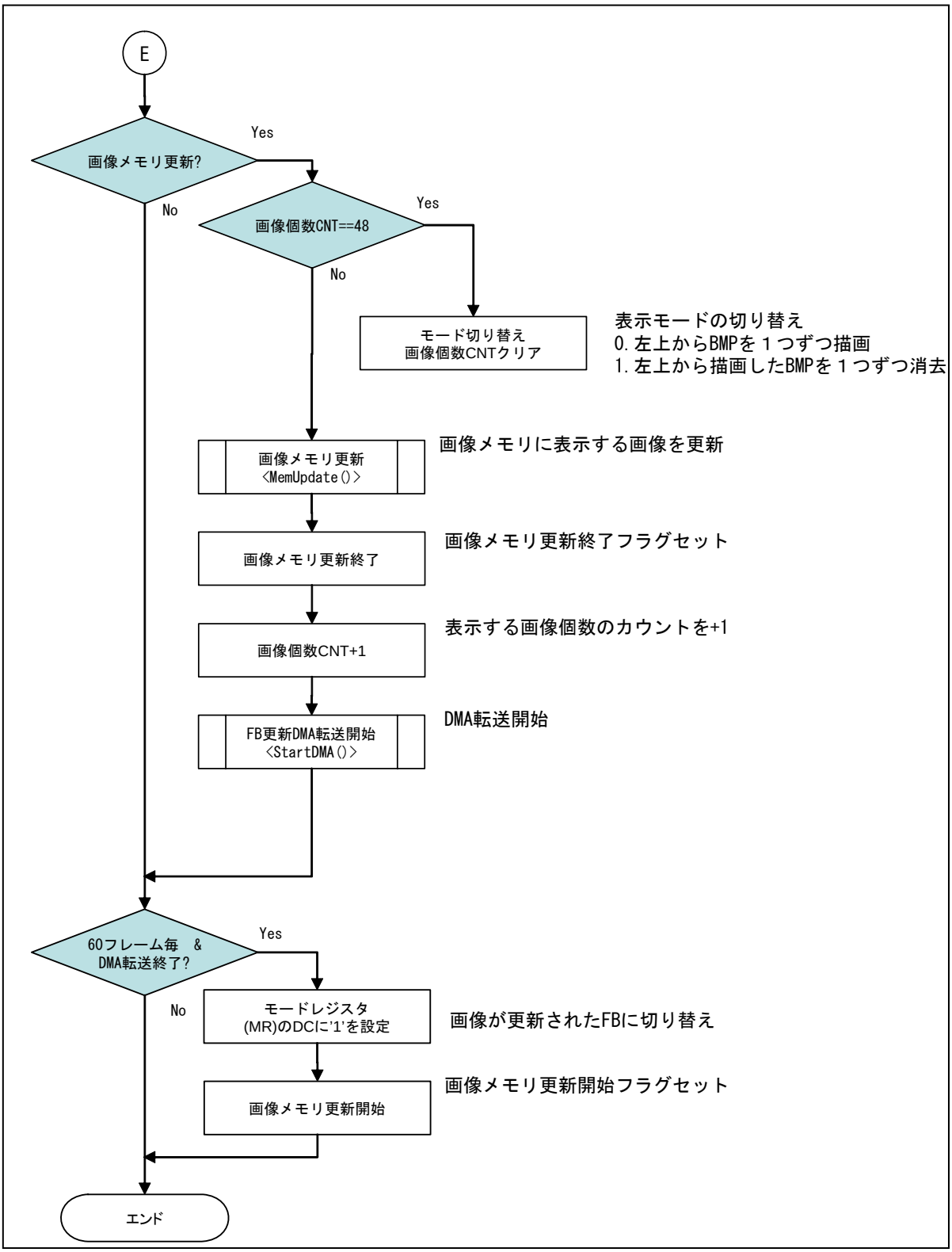

図 37 DemoSample プレーン 4 フロー

3.3.18 画像メモリ更新

図 38 にプレーン 4 で使用する画像メモリの更新フローを示します。

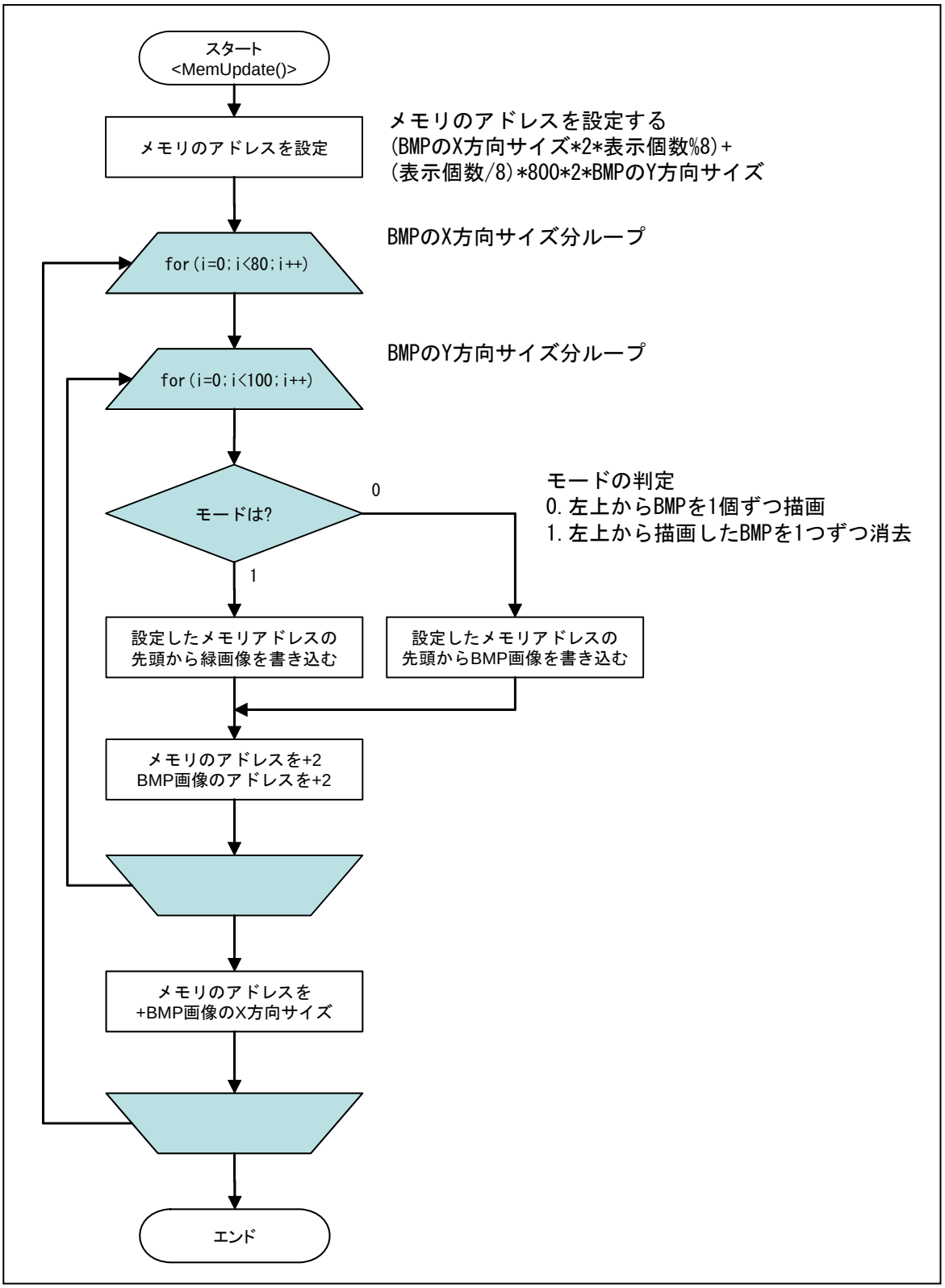

図 38 プレーン 4 画像メモリ更新フロー

3.3.19 FB更新DMA転送開始

図 39 にプレーン 4 のフレームバッファ更新の為の DMA 転送開始フローを示します。

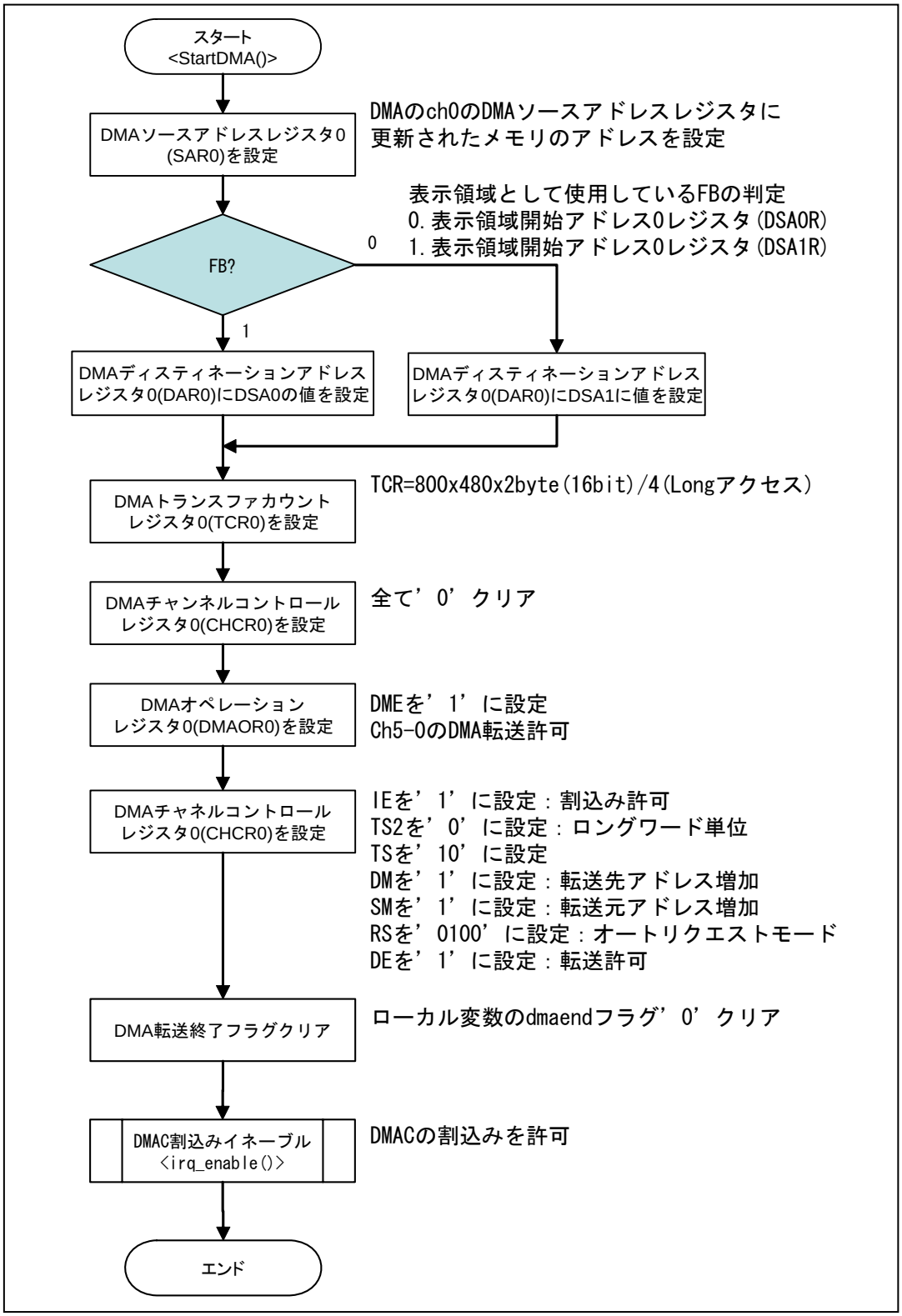

図 39 プレーン 4 フレームバッファ更新 DMA 転送開始フロー

3.3.20 DMA割込み

図 40 に DMA 転送終了割込みフローを示します。

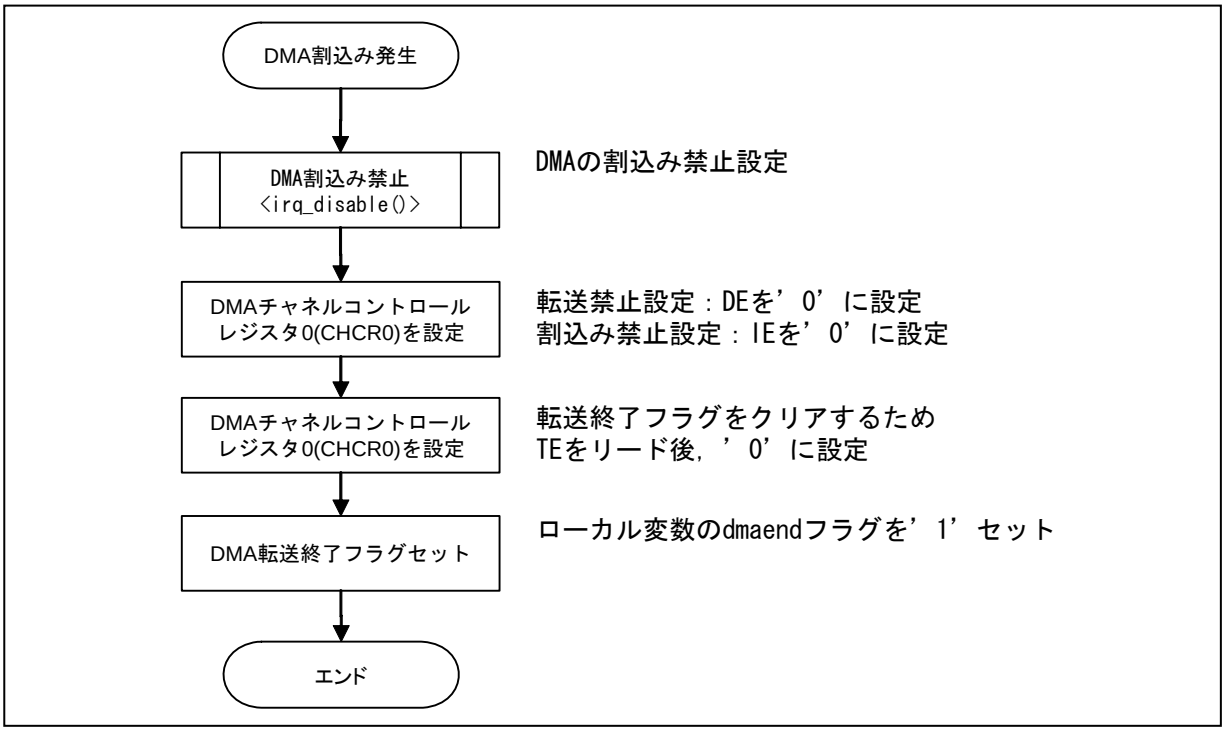

図 40 DMA 割込み終了フロー

3.3.21 セクション配置

表 18 に本応用例での各セクション配置を示します。

表 18 セクション配置

セクション名	セクション用途	領域	配置アドレス(仮想アドレス)	
P	プログラム領域	ROM	0x00002000	P0 領域 (キャッシング可能，MMU アドレス変換可能)
C	定数領域	ROM		
C$BSEC	未初期化データ領域用アドレス構造	ROM		
C$DSEC	初期化データ領域用アドレス構造	ROM		
D	初期化データ	ROM		
BMP	BMP ファイル	ROM	0x00010000	
B	未初期化データ領域	RAM	0x0C000000	
R	初期化データ領域	RAM		
S	スタック領域	RAM	0x0DFF8000	
INTHandler	例外/割込みハンドラ	ROM	0x80001000	P1 領域 (キャッシング可能，MMU アドレス変換不可)
VECTTBL	リセットベクタテーブル 割込みベクタテーブル	ROM		
INTTBL	割込みマスクテーブル	ROM		
PIntPRG	割込み関数	ROM		
FRAMEBUF	フレームバッファ	RAM	0x8C100000	
RSTHandler	リセットハンドラ	ROM	0xA0000000	P2 領域 (キャッシング不可，MMU アドレス変換不可)
PResetPRG	リセットプログラム	ROM		
PnonCACHE	プログラム領域 (キャッシュ無効アクセス)	ROM		

4. 参考プログラム例

サンプルプログラムリスト”DU_SampleProgram.c”

サンプルプログラムのメイン関数になります。

```
001 /***********************************************************************
002 * DISCLAIMER
003
004 * This software is supplied by Renesas Electronics Corporation. and is only
005 * intended for use with Renesas products. No other uses are authorized.
006
007 * This software is owned by Renesas Electronics Corporation. and is protected under
008 * all applicable laws, including copyright laws.
009
010 * THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
011 * REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
012 * INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
013 * PARTICULAR PURPOSE AND NON-INFRINGEMENT.  ALL SUCH WARRANTIES ARE EXPRESSLY
014 * DISCLAIMED.
015
016 * TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
017 * ELECTRONICS CORPORATION. NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
018 * FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
019 * FOR ANY REASON RELATED TO THE THIS SOFTWARE, EVEN IF RENESAS OR ITS
020 * AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
021
022 * Renesas reserves the right, without notice, to make changes to this
023 * software and to discontinue the availability of this software.
024 * By using this software, you agree to the additional terms and
025 * conditions found by accessing the following link:
026 * http://www.renesas.com/disclaimer
027 ***********************************************************************/
028 /* Copyright (C) 2010. Renesas Electronics Corporation., All Rights Reserved.       */
029 /*""FILE COMMENT""*********** Technical reference data ***************
030 * System Name  : SH7785 Sample Program
031 * File Name    : DU_DemoSample.c
032 * Abstract     : SH7785 DU Demo Sample Program
033 * Version      : Ver 1.00
034 * Device       : SH7785
035 * Tool-Chain   : High-performance Embedded Workshop (Version 4.07.00.007)
036 *              : C/C++ Compiler Package for SuperH Family (V.9.3.2.0)
037 * OS           : None
038 * H/W Platform : アルファプロジェクト製 SH-4A ボード 型番 AP-SH4A-3A
039 * Description  : SH7785DU の Demo のサンプルプログラムです。
040 *              :
041 * Operation    :
042 * Limitation   :
043 *              :
```

```
044 ***********************************************************************
045 * History      : 30.SEP.2010 Ver. 1.00 First Release
046 *""FILE COMMENT END""**************************************************/
047 /***********************************************************************/
048 /*                                                                     */
049 /*  FILE        :DU_DemoSample.c                                       */
050 /*  DATE        :Tue, Jul 20, 2010                                     */
051 /*  DESCRIPTION :Main Program                                          */
052 /*  CPU TYPE    :Other                                                 */
053 /*                                                                     */
054 /*  This file is generated by Renesas Project Generator (Ver.4.16).    */
055 /*                                                                     */
056 /***********************************************************************/
057
058
059
060 #ifdef __cplusplus
061 //#include <ios>                        // Remove the comment when you use ios
062 //_SINT ios_base::Init::init_cnt;       // Remove the comment when you use ios
063 #endif
064
065 #include "config.h"
066 #include "du.h"
067 #include "bitmap.h"
068 #include "intc.h"
069
070 void main(void);
071 #ifdef __cplusplus
072 extern "C" {
073 void abort(void);
074 }
075 #endif
076
077 /* ==== 変数宣言 ==== */
078 #define   BUFF_MAX        7
079 #define   yposcnt         XRES * 2 * 80
080 #define   xupdate         (XRES - 100) * 2
081 #define   pio_start       0
082 #define   pio_end         1
083 #define   membase         5
084 int ImageCnt = 0;
085 int xshift=0, yshift=0;
086 static int modify = 0;
087 static int update = 0;
088 static int dmaend = 0;
089 static int mode = 0;
090 unsigned long start_address = (unsigned long)__sectop("BMP");
```

```
091 #define offsadd           fb_base + FRAME_SIZE*(membase*2)
092 #define offdadd           fb_base + FRAME_SIZE*(membase*2+1)
093
094 /* ==== マクロ定義 ==== */
095 #define   xposcnt(a)      100 * 2 * (a)
096
097 /* ==== 関数宣言 ==== */
098 void pinfunc_init( void );
099 int FbWriteBMP( int planenum, int fb );
100 void FbUpdate( int planenum, int fb);
101 void FbClear( int planenum );
102 void FbCopy( int planenum );
103 void MemClear( void );
104 void MemUpdate( int cnt, int mode );
105 void BuffClear(char *pBuff, int size);
106 /*** DU ***/
107 extern void du_init(void);
108 extern void du_display_ctl(int on_off);
109 extern void du_vbk_irq(void);
110 extern struct plane_info du_plane_info;
111 extern void du_plane_enable( int planenum, int pri );
112 extern void du_plane_disable( int planenum, int pri );
113 /*** SCIF ***/
114 extern int scif_init(void);
115 extern char       scif_recive_data( char  *Data );
116 extern void scif_transmit_data( char *Data );
117 extern void scif_transmit_data_byte( char *Data );
118
119
120 /*""FUNC COMMENT""*****************************************************
121 * ID                :
122 * Outline           : サンプルプログラムメイン
123 *                   : (DU 表示)
124 * Include           :
125 * Declaration       : void main(void)
126 * Description       : SCIF の初期化後、コンソールに"SH7785 DU DEMO Sample"を表示します。
127 *                   : DU の初期化後、4 プレーン分のフレームバッファをクリアし、BMP ファイルを
128 *                   : 各プレーンのフレームバッファに書き込む。
129 *                   : 液晶ディスプレイに WVGA 画像を表示し、VSYNC 割込みを許可する。
130 *                   : コンソールから各プレーンの表示/非表示の制御を行える。
131 *                   :
132 * Limitation        :
133 *                   :
134 * Argument          : none
135 * Return Value      : none
136 * Calling Functions :
137 *""FUNC COMMENT END""*************************************************/
```

```
138 void main(void)
139 {
140   int ret;
141   int p1=1, p2=1, p3=1, p4=1;
142   char KeyBuff[BUFF_MAX];
143
144   ret = scif_init();
145   if( ret == 0 )
146           scif_transmit_data("¥n¥rSH7785 DU DEMO Sample¥n");
147
148   pinfunc_init();
149   du_init();
150   FbClear(PLANE1);
151   FbClear(PLANE2);
152   FbClear(PLANE3);
153   FbClear(PLANE4);
154   MemClear();
155   if(FbWriteBMP(PLANE1, 0) != 0)
156           scif_transmit_data("¥rBMP Write Error Plane1 FB1¥n");
157   if(FbWriteBMP(PLANE2, 0) != 0)
158           scif_transmit_data("¥rBMP Write Error Plane2 FB0¥n");
159   if(FbWriteBMP(PLANE2, 1) != 0)
160           scif_transmit_data("¥rBMP Write Error Plane2 FB1¥n");
161   if(FbWriteBMP(PLANE3, 0) != 0)
162           scif_transmit_data("¥rBMP Write Error Plane3 FB1¥n");
163   if(MemWriteBMP() != 0)
164           scif_transmit_data("¥rBMP Write Error Memory¥n");
165
166   du_display_ctl(DISP_ON);
167   irq_enable(_DU);
168   while(1) {
169           scif_transmit_data("¥rPLANE ON/OFF SETTING(TOGGLE)¥n");
170
171           if(p1) {
172                   scif_transmit_data("¥r### PLANE1 DISPLAY = ON  ###¥n");
173                   du_plane_enable( PLANE1, du_plane_info.plane[PLANE1].pri );
174           } else {
175                   scif_transmit_data("¥r### PLANE1 DISPLAY = OFF ###¥n");
176                   du_plane_disable( PLANE1, du_plane_info.plane[PLANE1].pri );
177           }
178
179           if(p2) {
180                   scif_transmit_data("¥r### PLANE2 DISPLAY = ON  ###¥n");
181                   du_plane_enable( PLANE2, du_plane_info.plane[PLANE2].pri );
182           } else {
183                   scif_transmit_data("¥r### PLANE2 DISPLAY = OFF ###¥n");
184                   du_plane_disable( PLANE2, du_plane_info.plane[PLANE2].pri );
```

```
185         }
186
187         if(p3) {
188                 scif_transmit_data("¥r### PLANE3 DISPLAY = ON  ###¥n");
189                 du_plane_enable( PLANE3, du_plane_info.plane[PLANE3].pri );
190         } else {
191                 scif_transmit_data("¥r### PLANE3 DISPLAY = OFF ###¥n");
192                 du_plane_disable( PLANE3, du_plane_info.plane[PLANE3].pri );
193         }
194
195         if(p4) {
196                 scif_transmit_data("¥r### PLANE4 DISPLAY = ON  ###¥n");
197                 du_plane_enable( PLANE4, du_plane_info.plane[PLANE4].pri );
198         } else {
199                 scif_transmit_data("¥r### PLANE4 DISPLAY = OFF ###¥n");
200                 du_plane_disable( PLANE4, du_plane_info.plane[PLANE4].pri );
201         }
202
203         scif_transmit_data("¥r-- Please Select Number --¥n");
204         scif_transmit_data("¥r-- 1 : PLANE1           --¥n");
205         scif_transmit_data("¥r-- 2 : PLANE2           --¥n");
206         scif_transmit_data("¥r-- 3 : PLANE3           --¥n");
207         scif_transmit_data("¥r-- 4 : PLANE4           --¥n");
208         scif_transmit_data("¥r-- >");
209
210         BuffClear( KeyBuff, BUFF_MAX );                         // Buff クリア
211         while( scif_recive_data( KeyBuff ) != 0);
212         switch( KeyBuff[0] ) {
213                 case '1' :
214                                         p1 = !p1;
215                                         break;
216                 case '2' :
217                                         p2 = !p2;
218                                         break;
219                 case '3' :
220                                         p3 = !p3;
221                                         break;
222                 case '4' :
223                                         p4 = !p4;
224                                         break;
225                 default :
226                                         break;
227         }
228         scif_transmit_data("¥n¥n");
229
230
231 }
```

```
232
233 }
234 /*""FUNC COMMENT""*****************************************************
235 * ID                :
236 * Outline           : サンプルプログラムメイン
237 *                   : (DU 表示)
238 * Include           :
239 * Declaration       : void pinfunc_init( void )
240 * Description       : ピンファンクションの設定
241 *                   :
242 *                   :
243 *                   :
244 *                   :
245 *                   :
246 * Limitation        :
247 *                   :
248 * Argument          : none
249 * Return Value      : none
250 * Calling Functions :
251 *""FUNC COMMENT END""*************************************************/
252
253 void pinfunc_init( void )
254 {
255   GPIO.P1MSELR.WORD = 0x2180;
256   GPIO.P2MSELR.WORD = 0x0000;
257   GPIO.PBCR.WORD = 0xFFF0;
258   GPIO.PCCR.WORD = 0x0000;
259   GPIO.PDCR.WORD = 0x0000;
260   GPIO.PHCR.WORD = 0xFC30;
261   GPIO.PMCR.WORD = 0xFFF5;
262   GPIO.PPCR.WORD = 0x03C0;
263 }
264
265 /* FB 関数 */
266 /*""FUNC COMMENT""*****************************************************
267 * ID                :
268 * Outline           : サンプルプログラムメイン
269 *                   : (DU 表示)
270 * Include           :
271 * Declaration       : void pset(int planenum, int fb, int x,int y,unsigned char r,unsigned char g,unsigned
char b)
272 * Description       : 1 画素(16bit)のデータをフレームバッファに書き込む
273 *                   :
274 *                   :
275 *                   :
276 *                   :
277 *                   :
278 * Limitation        :
```

```
279 *                   :
280 * Argument          : planmenum:プレーン番号, fb:FB 面, x:x 方向位置, y:y 方向位置,
281 *                   : r:画素赤, g:画素緑, b:画素青
282 * Return Value      : none
283 * Calling Functions :
284 *""FUNC COMMENT END""***************************************************/
285 void      pset(int planenum, int fb, int x,int y,unsigned char r,unsigned char g,unsigned char b)
286 {
287   int     offset, c, mask;
288   int memadd;
289
290   if ((x >= du_plane_info.xw) || (y >= du_plane_info.yw)) {
291           return;
292   }
293   offset = (y * du_plane_info.plane[planenum].dsx + x) * 2;
294
295   if (fb)
296           memadd = DUP(planenum).DSA1R.LONG;
297   else
298           memadd = DUP(planenum).DSA0R.LONG;
299
300   memadd += offset;
301   *(volatile unsigned short *)memadd = RGB16(r, g, b);
302 }
303
304 #if defined(_BIG)
305 /*""FUNC COMMENT""*****************************************************
306 * ID                :
307 * Outline           : サンプルプログラムメイン
308 *                   : (DU 表示)
309 * Include           :
310 * Declaration       : unsigned short swap_endian16(unsigned short value)
311 * Description       : 16bit データをバイトスワップする
312 *                   :
313 *                   :
314 *                   :
315 *                   :
316 *                   :
317 * Limitation        :
318 *                   :
319 * Argument          : 変換前値
320  Return Value       : 変換後値
321 * Calling Functions :
322 *""FUNC COMMENT END""***************************************************/
323
324 unsigned short swap_endian16(unsigned short value)
325 {
```

```
326     return ((value & 0xFF) << 8) | ((value >> 8) & 0xFF);
327 }
328
329 /*""FUNC COMMENT""*****************************************************
330 * ID                :
331 * Outline           : サンプルプログラムメイン
332 *                   : (DU 表示)
333 * Include           :
334 * Declaration       : unsigned short swap_endian32(unsigned long value)
335 * Description       : 32bit データをバイトスワップする
336 *                   :
337 *                   :
338 *                   :
339 *                   :
340 *                   :
341 * Limitation        :
342 *                   :
343 * Argument          : 変換前値
344  Return Value       : 変換後値
345 * Calling Functions :
346 *""FUNC COMMENT END""*************************************************/
347 unsigned long swap_endian32(unsigned long value)
348 {
349     return (value >> 24) | (value << 24) | ((value >> 8) & 0xFF00) | ((value << 8) & 0xFF0000);
350 }
351
352 /*""FUNC COMMENT""*****************************************************
353 * ID                :
354 * Outline           : サンプルプログラムメイン
355 *                   : (DU 表示)
356 * Include           :
357 * Declaration       : int FbWriteBMP( int planenum, int fb )
358 * Description       : BMP 画像をフレームバッファに書き込む
359 *                   : (ビッグエンディアン)
360 *                   :
361 *                   :
362 *                   :
363 *                   :
364 * Limitation        :
365 *                   :
366 * Argument          : planmenum:プレーン番号, fb:FB 面
367  Return Value       : -1 : BMP ファイルエラー
368 * Calling Functions :
369 *""FUNC COMMENT END""*************************************************/
370 int FbWriteBMP( int planenum, int fb )
371 {
372   bitmap_file_header_t *file_header = (bitmap_file_header_t *)start_address;
```

```
373   bitmap_header_t *bitmap_header = (bitmap_header_t *)(start_address + 14);
374   long x, y;
375   unsigned long bitmap_pitch, line;
376   unsigned char red, green, blue, index;
377   unsigned short rgb16;
378   unsigned long offsetadd;
379
380   if (swap_endian16(file_header->bitmap_file_type) != 0x4D42) {
381           scif_transmit_data("¥n¥rERROR_INVALID_BITMAP_FILE¥n");
382           return(-1);
383   }
384
385   // Calculate pitches.
386   bitmap_pitch = swap_endian32(file_header->bitmap_file_size);
387   bitmap_pitch -= swap_endian32(file_header->bitmap_file_bits_offset);
388   bitmap_pitch /= swap_endian32(bitmap_header->bitmap_height);
389
390   // Load all lines.
391   for (y = 0; y < swap_endian32(abs(bitmap_header->bitmap_height)); y++)
392   {
393           // Seek to line.
394           if (swap_endian32(bitmap_header->bitmap_height) > 0)
395           {
396                   line = swap_endian32(bitmap_header->bitmap_height) - y - 1;
397           }
398           else
399           {
400                   line = y;
401           }
402
403           /* アドレス更新 */
404           offsetadd = swap_endian32(file_header->bitmap_file_bits_offset) + line * bitmap_pitch;
405
406           // Load all pixels.
407           for (x = 0; x < swap_endian32(bitmap_header->bitmap_width); x++)
408           {
409                   rgb16 = swap_endian16(*(unsigned short *)(start_address + offsetadd + (x * 2)));
410                   red = ((rgb16 & 0xF800) >> 8);
411                   green = ((rgb16 & 0x07E0) >> 3);
412                   blue = ((rgb16 & 0x001F) << 3);
413                   pset(planenum, fb, x, y, red, green, blue);
414           }
415   }
416   start_address += swap_endian32(file_header->bitmap_file_size);
417   return 0;
418 }
419
```

```
420 #else
421 /*""FUNC COMMENT""************************************************
422 * ID                : 
423 * Outline           : サンプルプログラムメイン
424 *                   : (DU 表示)
425 * Include           : 
426 * Declaration       : int FbWriteBMP( int planenum, int fb )
427 * Description       : BMP 画像をフレームバッファに書き込む
428 *                   : (リトルエンディアン)
429 *                   : 
430 *                   : 
431 *                   : 
432 *                   : 
433 * Limitation        : 
434 *                   : 
435 * Argument          : planmenum:プレーン番号, fb:FB 面
436  Return Value       : -1 : BMP ファイルエラー
437 * Calling Functions : 
438 *""FUNC COMMENT END""*************************************************/
439 int FbWriteBMP( int planenum, int fb )
440 {
441   bitmap_file_header_t *file_header = (bitmap_file_header_t *)start_address;
442   bitmap_header_t *bitmap_header = (bitmap_header_t *)(start_address + 14);
443   long x, y;
444   unsigned long bitmap_pitch, line;
445   unsigned char red, green, blue, index;
446   unsigned short rgb16;
447   unsigned long offsetadd;
448 
449   if (file_header->bitmap_file_type != 0x4D42) {
450           scif_transmit_data("¥n¥rERROR_INVALID_BITMAP_FILE¥n");
451           return(-1);
452   }
453 
454   // Calculate pitches.
455   bitmap_pitch = file_header->bitmap_file_size;
456   bitmap_pitch -= file_header->bitmap_file_bits_offset;
457   bitmap_pitch /= bitmap_header->bitmap_height;
458 
459   // Load all lines.
460   for (y = 0; y < abs(bitmap_header->bitmap_height); y++)
461   {
462           // Seek to line.
463           if (bitmap_header->bitmap_height > 0)
464           {
465                   line = bitmap_header->bitmap_height - y - 1;
466           }
```

```
467           else
468           {
469                   line = y;
470           }
471
472           /* アドレス更新 */
473           offsetadd = file_header->bitmap_file_bits_offset + line * bitmap_pitch;
474
475           // Load all pixels.
476           for (x = 0; x < bitmap_header->bitmap_width; x++)
477           {
478                   rgb16 = *(unsigned short *)(start_address + offsetadd + (x * 2));
479                   red = ((rgb16 & 0xF800) >> 8);
480                   green = ((rgb16 & 0x07E0) >> 3);
481                   blue = ((rgb16 & 0x001F) << 3);
482                   pset(planenum, fb, x, y, red, green, blue);
483           }
484   }
485
486   start_address += file_header->bitmap_file_size;
487   return 0;
488 }
489 #endif
490 /*""FUNC COMMENT""*****************************************************
491 * ID                :
492 * Outline           : サンプルプログラムメイン
493 *                   : (DU 表示)
494 * Include           :
495 * Declaration       : void FbClear( int planenum )
496 * Description       : フレームバッファを透過色(緑)に初期化する
497 *                   :
498 *                   :
499 *                   :
500 *                   :
501 *                   :
502 * Limitation        :
503 *                   :
504 * Argument          : planmenum:プレーン番号
505  Return Value       : none
506 * Calling Functions :
507 *""FUNC COMMENT END""*************************************************/
508 void FbClear( int planenum )
509 {
510   unsigned long address0 = DUP(planenum).DSA0R.LONG;
511   unsigned long address1 = DUP(planenum).DSA1R.LONG;
512   int i;
513   unsigned long data = (unsigned long)(color(0, 63, 0)) << 16 | color(0, 63, 0);
```

```
514
515   for (i = 0; i < FRAME_SIZE/4; i++) {
516           *(unsigned long *)address0 = data;
517           *(unsigned long *)address1 = data;
518           address0 += 4;
519           address1 += 4;
520   }
521 }
522
523
524 #if defined(_BIG)
525 /*""FUNC COMMENT""*****************************************************
526 * ID                : 
527 * Outline           : サンプルプログラムメイン
528 *                   : (DU 表示)
529 * Include           : 
530 * Declaration       : int MemWriteBMP( void )
531 * Description       : (ビッグエンディアン)
532 *                   : 
533 *                   : 
534 *                   : 
535 *                   : 
536 *                   : 
537 * Limitation        : 
538 *                   : 
539 * Argument          : none
540  Return Value       : -1：BMP ファイルエラー
541 * Calling Functions : 
542 *""FUNC COMMENT END""*************************************************/
543 int MemWriteBMP( void )
544 {
545   bitmap_file_header_t *file_header = (bitmap_file_header_t *)start_address;
546   bitmap_header_t *bitmap_header = (bitmap_header_t *)(start_address + 14);
547   long x, y;
548   unsigned long bitmap_pitch, line;
549   unsigned char red, green, blue, index;
550   unsigned short rgb16;
551   unsigned long offsetadd;
552   int     offset;
553   int memadd;
554
555   if (swap_endian16(file_header->bitmap_file_type) != 0x4D42) {
556           scif_transmit_data("¥n¥rERROR_INVALID_BITMAP_FILE¥n");
557           return(-1);
558   }
559
560   // Calculate pitches.
```

```
561   bitmap_pitch = swap_endian32(file_header->bitmap_file_size);
562   bitmap_pitch -= swap_endian32(file_header->bitmap_file_bits_offset);
563   bitmap_pitch /= swap_endian32(bitmap_header->bitmap_height);
564
565
566   // Load all lines.
567   for (y = 0; y < swap_endian32(abs(bitmap_header->bitmap_height)); y++)
568   {
569           // Seek to line.
570           if (swap_endian32(bitmap_header->bitmap_height) > 0)
571           {
572                   line = swap_endian32(bitmap_header->bitmap_height) - y - 1;
573           }
574           else
575           {
576                   line = y;
577           }
578
579           /* アドレス更新 */
580           offsetadd = swap_endian32(file_header->bitmap_file_bits_offset) + line * bitmap_pitch;
581
582           // Load all pixels.
583           for (x = 0; x < swap_endian32(bitmap_header->bitmap_width); x++)
584           {
585                   rgb16 = swap_endian16(*(unsigned short *)(start_address + offsetadd + (x * 2)));
586                   red = ((rgb16 & 0xF800) >> 8);
587                   green = ((rgb16 & 0x07E0) >> 3);
588                   blue = ((rgb16 & 0x001F) << 3);
589                   offset = (y * swap_endian32(bitmap_header->bitmap_width) + x) * 2;
590                   memadd = offsadd;
591                   memadd += offset;
592                   *(volatile unsigned short *)memadd = RGB16(red, green, blue);
593           }
594   }
595
596   start_address += swap_endian32(file_header->bitmap_file_size);
597   return 0;
598
599 }
600 #else
601 /*""FUNC COMMENT""*****************************************************
602 * ID                : 
603 * Outline           : サンプルプログラムメイン
604 *                   : (DU 表示)
605 * Include           : 
606 * Declaration       : int MemWriteBMP( void )
607 * Description       : (リトルエンディアン)
```

```
608 *                   :
609 *                   :
610 *                   :
611 *                   :
612 *                   :
613 * Limitation        :
614 *                   :
615 * Argument          : none
616  Return Value       : -1：BMP ファイルエラー
617 * Calling Functions :
618 *""FUNC COMMENT END""***************************************************/
619 int MemWriteBMP( void )
620 {
621   bitmap_file_header_t *file_header = (bitmap_file_header_t *)start_address;
622   bitmap_header_t *bitmap_header = (bitmap_header_t *)(start_address + 14);
623   long x, y;
624   unsigned long bitmap_pitch, line;
625   unsigned char red, green, blue, index;
626   unsigned short rgb16;
627   unsigned long offsetadd;
628   int     offset;
629   int memadd;
630
631   if (file_header->bitmap_file_type != 0x4D42) {
632           scif_transmit_data("¥n¥rERROR_INVALID_BITMAP_FILE¥n");
633           return(-1);
634   }
635
636   // Calculate pitches.
637   bitmap_pitch = file_header->bitmap_file_size;
638   bitmap_pitch -= file_header->bitmap_file_bits_offset;
639   bitmap_pitch /= bitmap_header->bitmap_height;
640
641
642   // Load all lines.
643   for (y = 0; y < abs(bitmap_header->bitmap_height); y++)
644   {
645           // Seek to line.
646           if (bitmap_header->bitmap_height > 0)
647           {
648                   line = bitmap_header->bitmap_height - y - 1;
649           }
650           else
651           {
652                   line = y;
653           }
654
```

```
655           /* アドレス更新 */
656           offsetadd = file_header->bitmap_file_bits_offset + line * bitmap_pitch;
657
658           // Load all pixels.
659           for (x = 0; x < bitmap_header->bitmap_width; x++)
660           {
661                   rgb16 = *(unsigned short *)(start_address + offsetadd + (x * 2));
662                   red = ((rgb16 & 0xF800) >> 8);
663                   green = ((rgb16 & 0x07E0) >> 3);
664                   blue = ((rgb16 & 0x001F) << 3);
665                   offset = (y * bitmap_header->bitmap_width + x) * 2;
666                   memadd = offsadd;
667                   memadd += offset;
668                   *(volatile unsigned short *)memadd = RGB16(red, green, blue);
669           }
670   }
671
672   start_address += file_header->bitmap_file_size;
673   return 0;
674
675 }
676 #endif
677
678 /*""FUNC COMMENT""*****************************************************
679 * ID                : 
680 * Outline           : サンプルプログラムメイン
681 *                   : (DU 表示)
682 * Include           : 
683 * Declaration       : void MemClear( void )
684 * Description       : 画像メモリを透過色(緑)に初期化する
685 *                   : 
686 *                   : 
687 *                   : 
688 *                   : 
689 *                   : 
690 * Limitation        : 
691 *                   : 
692 * Argument          : none
693  Return Value       : none
694 * Calling Functions : 
695 *""FUNC COMMENT END""*************************************************/
696 void MemClear( void )
697 {
698   unsigned long address0 = offsadd;
699   unsigned long address1 = offdadd;
700   int i;
701   unsigned long data = (unsigned long)(color(0, 63, 0)) << 16 | color(0, 63, 0);
```

```
702
703   for (i = 0; i < FRAME_SIZE/4; i++) {
704           *(unsigned long *)address0 = data;
705           *(unsigned long *)address1 = data;
706           address0 += 4;
707           address1 += 4;
708   }
709 }
710
711 /*""FUNC COMMENT""*****************************************************
712 * ID                : 
713 * Outline           : サンプルプログラムメイン
714 *                   : (DU 表示)
715 * Include           : 
716 * Declaration       : void MemUpdate( int cnt, int mode )
717 * Description       : 画像メモリに BMP 画像 1 枚を追加する
718 *                   : 
719 *                   : 
720 *                   : 
721 *                   : 
722 *                   : 
723 * Limitation        : 
724 *                   : 
725 * Argument          : cnt:BMP 表示画像枚数, mode:0=増加/1=減少
726  Return Value       : none
727 * Calling Functions : 
728 *""FUNC COMMENT END""*************************************************/
729 void MemUpdate( int cnt, int mode )
730 {
731   unsigned long sadd = offsadd;
732   unsigned long dadd = offdadd;
733   int i, j;
734
735   /* 転送先スタートアドレス設定 */
736   dadd += (xposcnt(cnt%8)) + ((cnt/8) * yposcnt);
737
738   for(i = 0;i < 80; i++) {
739           for(j = 0;j < 100; j++) {
740                   if( mode )
741                           *(unsigned short *)dadd = color(0, 63, 0);
742                   else
743                           *(unsigned short *)dadd = *(unsigned short *)sadd;
744
745                   sadd += 2;
746                   dadd += 2;
747
748           }
```

```
749           dadd += xupdate;
750    }
751 }
752
753 /*""FUNC COMMENT""*****************************************************
754 * ID                :
755 * Outline           : サンプルプログラムメイン
756 *                   : (DU 表示)
757 * Include           :
758 * Declaration       : void StartDMA( int planenum, int fb )
759 * Description       : DMAC-ch0 の初期設定及び DMA 転送スタート
760 *                   : DMA の割込み許可
761 *                   :
762 *                   :
763 *                   :
764 *                   :
765 * Limitation        :
766 *                   :
767 * Argument          : planmenum:プレーン番号, fb:FB 面
768  Return Value       : none
769 * Calling Functions :
770 *""FUNC COMMENT END""*************************************************/
771 void StartDMA( int planenum, int fb )
772 {
773    *(unsigned long *)0xFC808020 = offdadd | 0xA0000000;
774    if ( fb )
775           *(unsigned long *)0xFC808024 = DUP(planenum).DSA0R.LONG | 0xA0000000;
776    else
777           *(unsigned long *)0xFC808024 = DUP(planenum).DSA1R.LONG | 0xA0000000;
778
779    DMAC0.TCR.BIT.CNT       = XRES * YRES * 2 / 4;
780    DMAC0.CHCR.LONG         = 0;
781    DMAC.DMAOR0.BIT.DME     = 1;    /* DMAC0-5 許可 */
782    DMAC0.CHCR.BIT.IE       = 1;    /* 割込み許可 */
783    DMAC0.CHCR.BIT.TS2      = 0;    /* long アクセス */
784    DMAC0.CHCR.BIT.TS       = 2;    /* long アクセス */
785    DMAC0.CHCR.BIT.DM       = 1;    /* 転送先増加 */
786    DMAC0.CHCR.BIT.SM       = 1;    /* 転送元増加 */
787    DMAC0.CHCR.BIT.RS       = 4;    /* オートリクエスト */
788    DMAC0.CHCR.BIT.DE       = 1;    /* 転送許可 */
789    dmaend = 0;
790    irq_enable( _DMAC0 );
791 }
792
793 /*""FUNC COMMENT""*****************************************************
794 * ID                :
795 * Outline           : サンプルプログラムメイン
```

```
796 *                   : (DU 表示)
797 * Include           :
798 * Declaration       : void dmac0_irq( void )
799 * Description       : DMAC の割込みの禁止, 転送禁止
800 *                   : DMA 転送終了フラグセット
801 *                   :
802 *                   :
803 *                   :
804 *                   :
805 * Limitation        :
806 *                   :
807 * Argument          : none
808  Return Value       : none
809 * Calling Functions :
810 *""FUNC COMMENT END""***************************************************/
811 void dmac0_irq( void )
812 {
813   int tmp;
814   DMAC0.CHCR.BIT.IE       = 0;    /* 割込み禁止 */
815   DMAC0.CHCR.BIT.DE       = 0;    /* 転送禁止 */
816   tmp = DMAC0.CHCR.BIT.TE;
817   DMAC0.CHCR.BIT.TE       = 0;    /* フラグクリア */
818   dmaend = 1;
819 }
820
821 /* デモサンプル */
822
823 /*""FUNC COMMENT""*****************************************************
824 * ID                :
825 * Outline           : サンプルプログラムメイン
826 *                   : (DU 表示)
827 * Include           :
828 * Declaration       : void DemoSample( void )
829 * Description       : PLANE1～4 のフレームバッファの更新
830 *                   :
831 *                   :
832 *                   :
833 *                   :
834 *                   :
835 * Limitation        :
836 *                   :
837 * Argument          : none
838  Return Value       : none
839 * Calling Functions :
840 *""FUNC COMMENT END""***************************************************/
841 void DemoSample( void )
842 {
```

```
843   int i = 0;
844   int xpos = XRES - 80;
845   int ypos = YRES - 80;
846
847   /* PLANE1 */
848   if ((DUP(i).DPXR.BIT.DPX > 0) && (DUP(i).DPYR.BIT.DPY == 0))     /* ←に移動 */
849           DUP(i).DPXR.BIT.DPX -= 2;
850   else if((DUP(i).DPXR.BIT.DPX == 0) && (DUP(i).DPYR.BIT.DPY < ypos)) /* ↓に移動 */
851           DUP(i).DPYR.BIT.DPY += 2;
852   else if((DUP(i).DPXR.BIT.DPX < xpos) && (DUP(i).DPYR.BIT.DPY == ypos)) /* →に移動 */
853           DUP(i).DPXR.BIT.DPX += 2;
854   else if((DUP(i).DPXR.BIT.DPX == xpos) && (DUP(i).DPYR.BIT.DPY > 0)) /* ↑に移動 */
855           DUP(i).DPYR.BIT.DPY -= 2;
856
857   i++;
858
859   /* PLANE2 */
860   if((modify%20) == 0)
861           DUP(i).MR.BIT.DC = 1;
862
863   if ((DUP(i).DPXR.BIT.DPX < xpos) && (DUP(i).DPYR.BIT.DPY == 0)) /* →に移動 */
864           DUP(i).DPXR.BIT.DPX += 1;
865   else if((DUP(i).DPXR.BIT.DPX == xpos) && (DUP(i).DPYR.BIT.DPY < ypos)) /* ↓に移動 */
866           DUP(i).DPYR.BIT.DPY += 1;
867   else if((DUP(i).DPXR.BIT.DPX > 0) && (DUP(i).DPYR.BIT.DPY == ypos)) /* ←に移動 */
868           DUP(i).DPXR.BIT.DPX -= 1;
869   else if((DUP(i).DPXR.BIT.DPX == 0) && (DUP(i).DPYR.BIT.DPY > 0)) /* ↑に移動 */
870           DUP(i).DPYR.BIT.DPY -= 1;
871
872   modify++;
873   i++;
874
875   /* PLANE3 */
876   if ((DUP(i).DPXR.BIT.DPX == 0) && (DUP(i).DPYR.BIT.DPY == 0)) {
877           xshift = 2;
878           yshift = 2;
879   } else if ((DUP(i).DPXR.BIT.DPX == xpos) && (DUP(i).DPYR.BIT.DPY == ypos)) {
880           xshift = -2;
881           yshift = -2;
882   } else if (DUP(i).DPXR.BIT.DPX == 0) {
883           xshift = 2;
884   } else if(DUP(i).DPYR.BIT.DPY == 0) {
885           yshift = 2;
886   } else if(DUP(i).DPXR.BIT.DPX == xpos) {
887           xshift = -2;
888   } else if(DUP(i).DPYR.BIT.DPY == ypos) {
889           yshift = -2;
```

```
890   } else {
891           xshift = xshift;
892           yshift = yshift;
893   }
894
895   DUP(i).DPXR.BIT.DPX += xshift;
896   DUP(i).DPYR.BIT.DPY += yshift;
897
898   i++;
899   /* PLANE4 */
900   /* PIO で画像メモリに元画像を転送 */
901   if( update == pio_start ) {
902           if( ImageCnt == 48 ) {
903                   mode = !mode;
904                   ImageCnt = 0;
905           }
906
907           MemUpdate( ImageCnt, mode );
908           update = pio_end;
909           ImageCnt++;
910           StartDMA( 3, DU.DSSR.BIT.DFB4 );
911   }
912
913   if(((modify%60) == 0) && dmaend) {
914           DUP(i).MR.BIT.DC = 1;
915           update = pio_start;
916           dmaend = 0;
917   }
918 }
919
920 /*""FUNC COMMENT""*****************************************************
921 * ID                : 
922 * Outline           : サンプルプログラムメイン
923 *                   : (DU 表示)
924 * Include           : 
925 * Declaration       : void BuffClear(char *pBuff, int size)
926 * Description       : シリアル受信データ用バッファの初期化
927 *                   : 
928 *                   : 
929 *                   : 
930 *                   : 
931 *                   : 
932 * Limitation        : 
933 *                   : 
934 * Argument          : *pBuff:バッファ, size:バッファサイズ
935  Return Value       : none
936 * Calling Functions : 
```

```
937 *""FUNC COMMENT END""***************************************************/
938 void BuffClear(char *pBuff, int size)
939 {
940   int i;
941   for( i = 0; i < size; i++ )     /* シリアルデータ受信ワークのクリア     */
942   {
943           *( pBuff + i ) = 0;
944   }
945 }
946
947
948
949 #ifdef __cplusplus
950 void abort(void)
951 {
952
953 }
954 #endif
```

サンプルプログラムリスト”du.c”

DU の初期設定を行っています。

```
001 /***********************************************************************
002 * DISCLAIMER
003
004 * This software is supplied by Renesas Electronics  Corporation. and is only
005 * intended for use with Renesas products. No other uses are authorized.
006
007 * This software is owned by Renesas Electronics  Corporation. and is protected under
008 * all applicable laws, including copyright laws.
009
010 * THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
011 * REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
012 * INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
013 * PARTICULAR PURPOSE AND NON-INFRINGEMENT.  ALL SUCH WARRANTIES ARE EXPRESSLY
014 * DISCLAIMED.
015
016 * TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
017 * ELECTRONICS CORPORATION. NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
018 * FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
019 * FOR ANY REASON RELATED TO THE THIS SOFTWARE, EVEN IF RENESAS OR ITS
020 * AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
021
022 * Renesas reserves the right, without notice, to make changes to this
023 * software and to discontinue the availability of this software.
024 * By using this software, you agree to the additional terms and
025 * conditions found by accessing the following link:
026 * http://www.renesas.com/disclaimer
027 ***********************************************************************/
028 /* Copyright (C) 2010. Renesas Electronics Corporation., All Rights Reserved.*/
029 /*""FILE COMMENT""********** Technical reference data ***************
030 * System Name  : SH7785 Sample Program
031 * File Name    : du.c
032 * Abstract     : SH7785 DU設定例 Sample Program
033 * Version      : Ver 1.00
034 * Device       : SH7785
035 * Tool-Chain   : High-performance Embedded Workshop (Version 4.07.00.007)
036 *              : C/C++ Compiler Package for SuperH Family (V.9.3.2.0)
037 * OS           : None
038 * H/W Platform : アルファプロジェクト製 SH-4A ボード 型番 AP-SH4A-3A
039 * Description  : SH7785 DU設定例のサンプルプログラムです。
040 *              :
041 * Operation    :
042 * Limitation   :
043 *              :
044 ***********************************************************************
045 * History      : 30.Sep.2010 Ver. 1.00 First Release
```

```
046 *""FILE COMMENT END""**************************************************/
047
048 #include "du.h"
049
050 /* ==== 関数宣言 ==== */
051 extern void DemoSample( void );
052 extern void delay( int cnt );
053
054 struct plane_info du_plane_info = {
055   DU_CLK_INTERNAL,                              /* clock_source */
056   HTOTAL,                                       /* hc X方向TOTAL */
057   VTOTAL,                                       /* vc Y方向TOTAL */
058   XRES,                                         /* xw X方向の表示サイズ */
059   YRES,                                         /* yw Y方向の表示サイズ */
060   X_FRONT,                                      /* xe X方向フロントポーチ */
061   X_BACK,                                       /* xs X方向バックポーチ */
062   Y_FRONT,                                      /* ye X方向フロントポーチ */
063   Y_BACK,                                       /* ys X方向バックポーチ */
064   H_WIDTH,                                      /* hsw Hsync幅 */
065   V_WIDTH,                                      /* vsw Vsync幅 */
066   {{      /* plane1 */
067           DISP_ON,                              /* plane_enable */
068           0,                                            /*dsa0フレームバッファアドレス0*/
069           0,                                            /*dsa1フレームバッファアドレス1*/
070           80,                                           /* dsx X方向のプレーンの表示サイズ
*/
071           80,                                           /* dsy Y方向のプレーンの表示サイズ
*/
072           XRES-80,                              /* dpx dsa原点の表示位置までのX方向の距離
*/
073           0,                                            /* dpy dsa原点の表示位置までのY方
向の距離 */
074           0,                                            /* spx dsa原点の開始位置までのX方
向の距離 */
075           0,                                            /* spy dsa原点の開始位置までのY方
向の距離 */
076           0,                                            /* waspy ラップアラウンドエリアのY
方向開始位置 */
077           80,                                           /* wamwy ラップアラウンドのY方向の
メモリ幅240-4095 */
078           1,                                            /* pri 優先順位 */
079           80,                                           /* mwx プレーンのX方向のメモリ幅
16-4096*/
080           80,                                           /* mly Y方向のメモリ領域 */
081   },
082   {       /* plane2 */
083           DISP_ON,                              /* plane_enable */
084           0,                                            /*dsa0フレームバッファアドレス0*/
085           0,                                            /*dsa1フレームバッファアドレス1*/
086           80,                                           /* dsx X方向のプレーンの表示サイズ
*/
```

```
087             80,                                     /* dsy Y 方向のプレーンの表示サイズ
*/
088             0,                                      /* dpx dsa 原点の表示位置までの X 方
向の距離 */
089             0,                                      /* dpy dsa 原点の表示位置までの Y 方
向の距離 */
090             0,                                      /* spx dsa 原点の表示開始位置までの
X 方向の距離 */
091             0,                                      /* spy dsa 原点の表示開始位置までの
Y 方向の距離 */
092             0,                                      /* waspy ラップアラウンドエリアの Y
方向開始位置 */
093             80,                                     /* wamwy ラップアラウンドの Y 方向の
メモリ幅 240-4095 */
094             2,                                      /* pri 優先順位 */
095             80,                                     /* mwx プレーンの X 方向のメモリ幅
16-4096*/
096             80,                                     /* mly Y 方向のメモリ領域 */
097     },
098     {       /* plane3 */
099             DISP_ON,                        /* plane_enable */
100             0,                                      /*dsa0 フレームバッファアドレス 0*/
101             0,                                      /*dsa1 フレームバッファアドレス 1*/
102             80,                                     /* dsx X 方向のプレーンの表示サイズ
*/
103             80,                                     /* dsy Y 方向のプレーンの表示サイズ
*/
104             0,                                      /* dpx dsa 原点の表示位置までの X 方
向の距離 */
105             YRES-80,                        /* dpy dsa 原点の表示位置までの Y 方向の距離
*/
106             0,                                      /* spx dsa 原点の表示開始位置までの
X 方向の距離 */
107             0,                                      /* spy dsa 原点の表示開始位置までの
Y 方向の距離 */
108             0,                                      /* waspy ラップアラウンドエリアの Y
方向開始位置 */
109             80,                                     /* wamwy ラップアラウンドの Y 方向の
メモリ幅 240-4095 */
110             3,                                      /* pri 優先順位 */
111             80,                                     /* mwx プレーンの X 方向のメモリ幅
16-4096*/
112             80,                                     /* mly Y 方向のメモリ領域 */
113     },
114 #if !defined(CONFIG_SCREEN_SVGA)
115     {       /* plane4 */
116             DISP_ON,                        /* plane_enable */
117             0,                                      /*dsa0 フレームバッファアドレス 0*/
118             0,                                      /*dsa1 フレームバッファアドレス 1*/
119             XRES,                           /* dsx X 方向のプレーンの表示サイズ */
120             YRES,                           /* dsy Y 方向のプレーンの表示サイズ */
121             0,                                      /* dpx dsa 原点の表示位置までの X 方
向の距離 */
122             0,                                      /* dpy dsa 原点の表示位置までの Y 方
向の距離 */
```

```
  123           0,                                      /* spx dsa 原点の表示開始位置までの
X 方向の距離 */
  124           0,                                      /* spy dsa 原点の表示開始位置までの
Y 方向の距離 */
  125           0,                                      /* waspy ラップアラウンドエリアの Y
方向開始位置 */
  126           YRES,                                 /* wamwy ラップアラウンドの Y 方向のメモリ幅
240-4095 */
  127           4,                                      /* pri 優先順位 */
  128           XRES,                                 /* mwx プレーンの X 方向のメモリ幅 16-4096*/
  129           YRES,                                 /* mly Y 方向のメモリ領域 */
  130   },
  131 #if defined(CONFIG_SCREEN_VGA)
  132   {       /* plane5 */
  133           DISP_ON,                              /* plane_enable */
  134           0,                                      /* dsa0 フレームバッファアドレス 0 */
  135           0,                                      /* dsa1 フレームバッファアドレス 1 */
  136           40,                                     /* dsx X 方向のプレーンの表示サイズ
*/
  137           40,                                     /* dsy Y 方向のプレーンの表示サイズ
*/
  138           0,                                      /* dpx dsa 原点の表示位置までの X 方
向の距離 */
  139           0,                                      /* dpy dsa 原点の表示位置までの Y 方
向の距離 */
  140           0,                                      /* spx dsa 原点の表示開始位置までの
X 方向の距離 */
  141           0,                                      /* spy dsa 原点の表示開始位置までの
Y 方向の距離 */
  142           0,                                      /* waspy ラップアラウンドエリアの Y
方向開始位置 */
  143           YRES,                                 /* wamwy ラップアラウンドの Y 方向のメモリ幅
240-4095 */
  144           5,                                      /* pri 優先順位 */
  145           0,                                      /* mwx プレーンの X 方向のメモリ幅
16-4096*/
  146           0,                                      /* mly Y 方向のメモリ領域 */
  147   },
  148   {       /* plane6 */
  149           DISP_ON,                              /* plane_enable */
  150           0,                                      /* dsa0 フレームバッファアドレス 0 */
  151           0,                                      /* dsa1 フレームバッファアドレス 1 */
  152           40,                                     /* dsx X 方向のプレーンの表示サイズ
*/
  153           40,                                     /* dsy Y 方向のプレーンの表示サイズ
*/
  154           0,                                      /* dpx dsa 原点の表示位置までの X 方
向の距離 */
  155           0,                                      /* dpy dsa 原点の表示位置までの Y 方
向の距離 */
  156           0,                                      /* spx dsa 原点の表示開始位置までの
X 方向の距離 */
  157           0,                                      /* spy dsa 原点の表示開始位置までの
Y 方向の距離 */
```

```
  158           0,                                                      /* waspy ラップアラウンドエリアの Y
方向開始位置 */
  159           0,                                                      /* wamwy ラップアラウンドの Y 方向の
メモリ幅 240-4095 */
  160           6,                                                      /* pri 優先順位 */
  161           0,                                                      /* mwx プレーンの X 方向のメモリ幅
16-4096*/
  162           0,                                                      /* mly Y 方向のメモリ領域 */
  163   },
  164 #endif /* CONFIG_SCREEN_VGA | CONFIG_SCREEN_WVGA */
  165 #endif /* CONFIG_SCREEN_SVGA */
  166   },
  167 };
  168
  169 /*""FUNC COMMENT""*****************************************************
  170 * ID                :
  171 * Outline           : サンプルプログラムメイン
  172 *                   : (DU 表示)
  173 * Include           :
  174 * Declaration       : void du_plane_init( int planenum )
  175 * Description       : プレーンの初期設定
  176 *                   :
  177 *                   :
  178 *                   :
  179 *                   :
  180 *                   :
  181 * Limitation        :
  182 *                   :
  183 * Argument          : planenum：プレーン番号
  184 * Return Value      : none
  185 * Calling Functions :
  186 *""FUNC COMMENT END""*************************************************/
  187 void du_plane_init( int planenum )
  188 {
  189           /* 各プレーン DSA 設定 */
  190           DUP(planenum).DSA0R.LONG = fb_base + FRAME_SIZE*(planenum*2);
  191           DUP(planenum).DSA1R.LONG = fb_base + FRAME_SIZE*(planenum*2+1);
  192
  193           /* 各プレーンモード設定 */
  194           DUP(planenum).MR.LONG = FMT_BPP16 | BUF_MODE_MANU;
  195
  196           /* 各プレーン画面サイズ設定 */
  197           DUP(planenum).DSXR.BIT.DSX = du_plane_info.plane[planenum].dsx;
  198           DUP(planenum).DSYR.BIT.DSY = du_plane_info.plane[planenum].dsy;
  199
  200           /* 各プレーン表示位置設定 */
  201           DUP(planenum).DPXR.LONG = du_plane_info.plane[planenum].dpx;
  202           DUP(planenum).DPYR.LONG = du_plane_info.plane[planenum].dpy;
```

```
203
204             /* 各プレーン開始位置設定 */
205             DUP(planenum).SPXR.BIT.SPX = du_plane_info.plane[planenum].spx;
206             DUP(planenum).SPYR.BIT.SPY = du_plane_info.plane[planenum].spy;
207
208             /* 各プレーン X 方向メモリ幅設定 */
209             DUP(planenum).MWR.LONG = du_plane_info.plane[planenum].mwx;
210
211             /* 各プレーン透過色設定 */
212             DUP(planenum).TC2R.BIT.TC2 = color(0, 63, 0);
213
214 }
215
216 /*""FUNC COMMENT""*****************************************************
217 * ID                :
218 * Outline           : サンプルプログラムメイン
219 *                   : (DU 表示)
220 * Include           :
221 * Declaration       : void du_plane_enable( int planenum, int pri )
222 * Description       : プレーンの表示 ON
223 *                   :
224 *                   :
225 *                   :
226 *                   :
227 *                   :
228 * Limitation        :
229 *                   :
230 * Argument          : planenum：プレーン番号, pri：優先順位
231 * Return Value      : none
232 * Calling Functions :
233 *""FUNC COMMENT END""*************************************************/
234 void du_plane_enable( int planenum, int pri )
235 {
236   unsigned long tmp;
237   int i, dpe, dps;
238   tmp = DU.DPPR.LONG;
239
240
241   for(i = 0;i < 6; i++) {
242             if( ((tmp >> (i * 4)) & 0x7) == (pri - 1))
243                       tmp &= ~(0xf << (i * 4));
244   }
245
246   switch (pri) {
247             case 1:
248                                 dpe = 3;
249                           dps = 0;
```

```
250                         break;
251             case 2:
252                             dpe = 7;
253                         dps = 4;
254                         break;
255             case 3:
256                             dpe = 11;
257                         dps = 8;
258                         break;
259             case 4:
260                             dpe = 15;
261                         dps = 12;
262                         break;
263             case 5:
264                             dpe = 19;
265                         dps = 16;
266                         break;
267             case 6:
268                             dpe = 23;
269                         dps = 20;
270                         break;
271             default:
272                             break;
273     }
274     tmp &= ~(0xf << dps);
275     tmp |= (1 << dpe) + (planenum << dps);
276     DU.DPPR.LONG = tmp;
277 }
278
279 /*""FUNC COMMENT""*****************************************************
280 * ID                :
281 * Outline           : サンプルプログラムメイン
282 *                   : (DU 表示)
283 * Include           :
284 * Declaration       : void du_plane_init( int planenum )
285 * Description       : プレーンの表示 OFF
286 *                   :
287 *                   :
288 *                   :
289 *                   :
290 *                   :
291 * Limitation        :
292 *                   :
293 * Argument          : planenum：プレーン番号, pri：優先順位
294 * Return Value      : none
295 * Calling Functions :
296 *""FUNC COMMENT END""*************************************************/
```

```
297 void du_plane_disable( int planenum, int pri )
298 {
299   unsigned long tmp;
300   int i, dpe, dps;
301   tmp = DU.DPPR.LONG;
302
303
304   for(i = 0;i < 6; i++) {
305           if( ((tmp >> (i * 4)) & 0x7) == (pri - 1))
306                   tmp &= ~(0xf << (i * 4));
307   }
308
309   switch (pri) {
310           case 1:
311                           dpe = 3;
312                   dps = 0;
313                   break;
314           case 2:
315                           dpe = 7;
316                   dps = 4;
317                   break;
318           case 3:
319                           dpe = 11;
320                   dps = 8;
321                   break;
322           case 4:
323                           dpe = 15;
324                   dps = 12;
325                   break;
326           case 5:
327                           dpe = 19;
328                   dps = 16;
329                   break;
330           case 6:
331                           dpe = 23;
332                   dps = 20;
333                   break;
334           default:
335                           break;
336   }
337   tmp &= ~(0xf << dps);
338   tmp |= (0 << dpe) + (planenum << dps);
339   DU.DPPR.LONG = tmp;
340 }
341
342 /*""FUNC COMMENT""*****************************************************
343 * ID                :
```

```
344 * Outline          : サンプルプログラムメイン
345 *                  : (DU 表示)
346 * Include          :
347 * Declaration      : void du_init( void )
348 * Description      : DU の初期設定
349 *                  :
350 *                  :
351 *                  :
352 *                  :
353 *                  :
354 * Limitation       :
355 *                  :
356 * Argument         : none
357 * Return Value     : none
358 * Calling Functions :
359 *""FUNC COMMENT END""**************************************************/
360 void du_init( void )
361 {
362   int i;
363
364   /* 初期化 */
365   DU.DSYSR.BIT.DRES = 1;
366   DU.DSYSR.BIT.TVM = MASTER_MODE;
367 #if defined(_BIG)
368   DU.DSYSR.BIT.DSEC = 1;                  /* ビッグエンディアン */
369 #endif
370
371   DU.DSMR.BIT.CSPM = 1;                   /* HSYNC 信号出力 */
372   DU.DSMR.BIT.CDED = 1;                   /* CDE 信号ディセーブル */
373
374   DU.DSRCR.LONG = 0x0000CB00;             /* 表示ｽﾃｰﾀｽﾚｼﾞｽﾀのｸﾘｱ */
375
376   DU.DIER.BIT.VBE = 1;                    /* 垂直割込み許可 */
377
378   DU.CPCR.LONG = 0x00000000;              /* ｶﾗｰﾊﾟﾚｯﾄ 4-1 未使用 */
379
380 //        DU.DEFR.BIT.DSAE = 1;                   /* フレームアドレス拡張(31-4) */
381   DU.DEFR.BIT.DCKE = 1;
382
383   /* 水平表示開始位置             */
384   DU.HDSR.BIT.HDS = du_plane_info.hsw + du_plane_info.xs - REVISE;
385
386   /* 水平表示終了位置             */
387   DU.HDER.BIT.HDE = DU.HDSR.BIT.HDS + du_plane_info.xw;
388
389   /* 垂直表示開始位置             */
390   DU.VDSR.BIT.VDS = du_plane_info.ys - 2;
```

```
391
392   /* 垂直表示終了位置              */
393   DU.VDER.BIT.VDE = DU.VDSR.BIT.VDS + du_plane_info.yw;
394
395   /* 水平走査周期                  */
396   DU.HCR.BIT.HC = du_plane_info.hc - 1;
397
398   /* 水平同期ﾊﾟﾙｽ幅                */
399   DU.HSWR.BIT.HSW = du_plane_info.hsw - 1;
400
401   /* 垂直同期位置                  */
402   DU.VSPR.BIT.VSP = du_plane_info.vc - du_plane_info.vsw - 1;
403
404   /* 垂直走査同期位置              */
405   DU.VCR.BIT.VC = du_plane_info.vc - 1;
406
407   DU.EQWR.LONG = 0;
408   DU.SPWR.LONG = 0;
409   DU.CLAMPSR.LONG = 0;
410   DU.CLAMPWR.LONG = 1;
411
412   DU.CP1TR.LONG = 0;                      /* カラーパレット 1 未使用 */
413   DU.CP2TR.LONG = 0;                      /* カラーパレット 2 未使用 */
414   DU.CP3TR.LONG = 0;                      /* カラーパレット 3 未使用 */
415   DU.CP4TR.LONG = 0;                      /* カラーパレット 4 未使用 */
416   DU.DOOR.LONG = 0x0000FC00;      /* プレーン未使用時色設定：緑 */
417   DU.CDER.LONG = 0;                       /* 色検出未設定 */
418   DU.BPOR.LONG = 0x00204494;      /* 下地色設定：藍青 */
419   DU.RINTOFSR.LONG = 0;           /* ラスタ割込み未設定 */
420
421
422   /* 各プレーンの初期化 */
423   for (i=0;i<PLANE_NUM;i++) {
424           du_plane_init( i );
425           du_plane_enable( i, du_plane_info.plane[i].pri );
426   }
427
428   DU.ESCR.BIT.DCLKSEL = du_plane_info.clk_sorce;
429   DU.ESCR.BIT.FRQSEL = 3;         /* 入力クロック 4 分周 */
430
431   DU.OTAR.LONG = 0;
432 }
433
434 /*""FUNC COMMENT""*****************************************************
435 * ID                :
436 * Outline           : サンプルプログラムメイン
437 *                   : (DU 表示)
```

```
438 * Include           :
439 * Declaration       : void du_display_ctl( int on_off )
440 * Description       : DU の表示 ON/OFF 制御
441 *                   :
442 *                   :
443 *                   :
444 *                   :
445 *                   :
446 * Limitation        :
447 *                   :
448 * Argument          : on_off : Value=1 で ON、0 で OFF
449 * Return Value      : none
450 * Calling Functions :
451 *""FUNC COMMENT END""**************************************************/
452 void du_display_ctl( int on_off )
453 {
454   if (on_off) {
455           DU.DSYSR.BIT.DRES = 0;
456           DU.DSYSR.BIT.DEN = 1;
457   } else {
458           DU.DSYSR.BIT.DRES = 1;
459           DU.DSYSR.BIT.DEN = 0;
460   }
461 }
462
463 /*""FUNC COMMENT""*****************************************************
464 * ID                :
465 * Outline           : サンプルプログラムメイン
466 *                   : (DU 表示)
467 * Include           :
468 * Declaration       : void du_tvr_irq(void)
469 * Description       : TVR の割込み処理
470 *                   :
471 *                   :
472 *                   :
473 *                   :
474 *                   :
475 * Limitation        :
476 *                   :
477 * Argument          : none
478 * Return Value      : none
479 * Calling Functions :
480 *""FUNC COMMENT END""**************************************************/
481 void du_tvr_irq(void)
482 {
483   DU.DSRCR.BIT.TVCL = 1;
484 }
```

```
485
486 /*""FUNC COMMENT""************************************************
487 * ID                :
488 * Outline           : サンプルプログラムメイン
489 *                   : (DU 表示)
490 * Include           :
491 * Declaration       : void du_frm_irq(void)
492 * Description       : FRM の割込み処理
493 *                   :
494 *                   :
495 *                   :
496 *                   :
497 *                   :
498 * Limitation        :
499 *                   :
500 * Argument          : none
501 * Return Value      : none
502 * Calling Functions :
503 *""FUNC COMMENT END""************************************************/
504 void du_frm_irq(void)
505 {
506   DU.DSRCR.BIT.FRCL = 1;
507 }
508
509
510 /*""FUNC COMMENT""************************************************
511 * ID                :
512 * Outline           : サンプルプログラムメイン
513 *                   : (DU 表示)
514 * Include           :
515 * Declaration       : void du_vbk_irq(void)
516 * Description       : VBK の割込み処理
517 *                   : DemoSample1 の実行
518 *                   :
519 *                   :
520 *                   :
521 *                   :
522 * Limitation        :
523 *                   :
524 * Argument          : none
525 * Return Value      : none
526 * Calling Functions :
527 *""FUNC COMMENT END""************************************************/
528 void du_vbk_irq(void)
529 {
530   DU.DSRCR.BIT.VBCL = 1;
531
```

```
532   DemoSample();
533 }
534
535 /*""FUNC COMMENT""*****************************************************
536 * ID                :
537 * Outline           : サンプルプログラムメイン
538 *                   : (DU 表示)
539 * Include           :
540 * Declaration       : void du_rint_irq(void)
541 * Description       : RINT の割込み処理
542 *                   :
543 *                   :
544 *                   :
545 *                   :
546 *                   :
547 * Limitation        :
548 *                   :
549 * Argument          : none
550 * Return Value      : none
551 * Calling Functions :
552 *""FUNC COMMENT END""*************************************************/
553 void du_rint_irq(void)
554 {
555   DU.DSRCR.BIT.RICL = 1;
556
557 }
558
559 /*""FUNC COMMENT""*****************************************************
560 * ID                :
561 * Outline           : サンプルプログラムメイン
562 *                   : (DU 表示)
563 * Include           :
564 * Declaration       : void du_hbk_irq(void)
565 * Description       : HBK 割込みの処理
566 *                   :
567 *                   :
568 *                   :
569 *                   :
570 *                   :
571 * Limitation        :
572 *                   :
573 * Argument          : none
574 * Return Value      : none
575 * Calling Functions :
576 *""FUNC COMMENT END""*************************************************/
577 void du_hbk_irq(void)
578 {
```

```
579    DU.DSRCR.BIT.HBCL = 1;
580
581 }
582
```

サンプルプログラムリスト”du.h”

“du.c”で使用するヘッダーです。

```
001
002 #ifndef _DU_H_
003 #define _DU_H_
004
005 #include "config.h"
006 #include "iodefine.h"
007
008 #define   PLANE1  0
009 #define   PLANE2  1
010 #define   PLANE3  2
011 #define   PLANE4  3
012 #define   PLANE5  4
013 #define   PLANE6  5
014
015 #if defined(CONFIG_SCREEN_VGA)
016 #define XRES              640
017 #define   YRES            480
018 #define   X_FRONT         105
019 #define   X_BACK          16
020 #define   Y_FRONT         33
021 #define   Y_BACK          10
022 #define   H_WIDTH         39
023 #define   V_WIDTH         2
024 #define PLANE_NUM 4
025 #elif defined(CONFIG_SCREEN_WVGA)
026 #define XRES              800
027 #define   YRES            480
028 #define   X_FRONT         220
029 #define   X_BACK          110
030 #define   Y_FRONT         35
031 #define   Y_BACK          5
032 #define   H_WIDTH         128
033 #define   V_WIDTH         5
034 #define PLANE_NUM 4
035 #elif defined(CONFIG_SCREEN_SVGA)
036 #define XRES              800
037 #define   YRES            600
038 #define   X_FRONT         0
039 #define   X_BACK          0
040 #define   Y_FRONT         0
041 #define   Y_BACK          0
042 #define   H_WIDTH         0
043 #define   V_WIDTH         0
044 #define PLANE_NUM 3
045 #else
```

```
046 #define XRES              480
047 #define   YRES            234
048 #define   X_FRONT         0
049 #define   X_BACK          0
050 #define   Y_FRONT         0
051 #define   Y_BACK          0
052 #define   H_WIDTH         0
053 #define   V_WIDTH         0
054 #define PLANE_NUM 6
055 #endif
056
057 #define   HTOTAL  XRES + X_FRONT + X_BACK + H_WIDTH
058 #define   VTOTAL  YRES + Y_FRONT + Y_BACK + V_WIDTH
059
060 /* 透過色設定 RGB=5:6:5 R/B=0-31, G=0-63 */
061 #define   color(R, G, B)  (R << 11) | (G << 5) | B
062
063
064 #define   REVISE  19              /* マスター */
065 //#define REVISE  24      /* TV 同期 */
066
067 #define MASTER_MODE       0
068 #define TV_MODE           2
069
070 #define   DU_CLK_INTERNAL 1
071 #define   DU_CLK_EXTERNAL 0
072
073 #define   DISP_ON         1
074 #define   DISP_OFF        0
075
076 static int        fb_base = (int)__sectop("FRAMEBUF");
077 #define   FRAME_SIZE      1024 * 1024 * 1 /* 1MB */
078
079 #define DUP(ch)     (*(volatile struct st_dup     *)((unsigned long )0xFFF80000 + ((ch+1) << 8)))
080
081 /* PnMR */
082 #define   FMT_BPP8                0
083 #define   FMT_BPP16               1
084 #define   FMT_ARGB                2
085 #define   FMT_YC                  3
086 #define BUF_MODE_AUTO     (2 << 4)
087 #define BUF_MODE_MANU     (0 << 4)
088 #define DC_ON                     (1 << 7)
089 #define DC_OFF                    (0 << 7)
090 #define WAE_ON                    (1 << 16)
091 #define WAE_OFF                   (0 << 16)
092
```

```
093
094
095
096 struct plane_cfg {
097   int plane_enable;       /* プレーン表示  ON:1、OFF:0*/
098   int     dsa0;                   /* フレームバッファアドレス 0 */
099   int     dsa1;                   /* フレームバッファアドレス 1 */
100   int dsx;                        /* X 方向のプレーンの表示サイズ */
101   int dsy;                        /* Y 方向のプレーンの表示サイズ */
102   int     dpx;                    /* dsa 原点の表示位置までの X 方向の距離 */
103   int     dpy;                    /* dsa 原点の表示位置までの Y 方向の距離 */
104   int spx;                        /* dsa 原点の開始位置までの X 方向の距離 */
105   int spy;                        /* dsa 原点の開始位置までの Y 方向の距離 */
106   int waspy;                      /* ラップアラウンドエリアの Y 方向開始位置 */
107   int wamwy;                      /* ラップアラウンドの Y 方向のメモリ幅 240-4095 */
108   int pri;                        /* pri 優先順位 */
109   int mwx;                        /* プレーンの X 方向のメモリ幅 16-4096*/
110   int mly;                        /* Y 方向のメモリ領域 */
111 };
112
113 struct plane_info {
114   int clk_sorce;
115   int hc;                         /* X 方向 TOTAL */
116   int vc;                         /* Y 方向 TOTAL */
117   int xw;                 /* X 方向の表示サイズ */
118   int yw;                 /* Y 方向の表示サイズ */
119   int xe;                 /* X 方向バックポーチ */
120   int xs;                 /* X 方向フロントポーチ */
121   int ye;                 /* Y 方向バックポーチ */
122   int ys;                 /* Y 方向フロントポーチ */
123   int hsw;                /* Hsync 幅 */
124   int vsw;                /* Vsync 幅 */
125   struct plane_cfg plane[PLANE_NUM];
126 };
127
128
129
130
131 #endif /* _DU_H_ */
```

サンプルプログラムリスト”scif.c”

シリアルコミュニケーションチャンネル 1 の初期設定例です。

```
001 /***********************************************************************
002 * DISCLAIMER
003
004 * This software is supplied by Renesas Electronics  Corporation. and is only
005 * intended for use with Renesas products. No other uses are authorized.
006
007 * This software is owned by Renesas Electronics  Corporation. and is protected under
008 * all applicable laws, including copyright laws.
009
010 * THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
011 * REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
012 * INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
013 * PARTICULAR PURPOSE AND NON-INFRINGEMENT.  ALL SUCH WARRANTIES ARE EXPRESSLY
014 * DISCLAIMED.
015
016 * TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
017 * ELECTRONICS CORPORATION. NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
018 * FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
019 * FOR ANY REASON RELATED TO THE THIS SOFTWARE, EVEN IF RENESAS OR ITS
020 * AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
021
022 * Renesas reserves the right, without notice, to make changes to this
023 * software and to discontinue the availability of this software.
024 * By using this software, you agree to the additional terms and
025 * conditions found by accessing the following link:
026 * http://www.renesas.com/disclaimer
027 ***********************************************************************/
028 /* Copyright (C) 2010. Renesas Electronics Corporation., All Rights Reserved.*/
029 /*""FILE COMMENT""********** Technical reference data ***************
030 * System Name  : SH7785 Sample Program
031 * File Name    : scif.c
032 * Abstract     : SH7785 SCIF 設定例 Sample Program
033 * Version      : Ver 1.00
034 * Device       : SH7785
035 * Tool-Chain   : High-performance Embedded Workshop (Version 4.07.00.007)
036 *              : C/C++ Compiler Package for SuperH Family (V.9.3.2.0)
037 * OS           : None
038 * H/W Platform : アルファプロジェクト製 SH-4A ボード 型番 AP-SH4A-3A
039 * Description  : SH7785SCIF 設定例のサンプルプログラムです。
040 *              :
041 * Operation    :
042 * Limitation   :
043 *              :
044 ***********************************************************************
045 * History      : 01.Sep.2010 Ver. 1.00 First Release
```

```
046 *""FILE COMMENT END""**************************************************/
047
048
049 #include  "scif.h"
050
051 /*""FUNC COMMENT""*****************************************************
052 * ID               :
053 * Outline          : サンプルプログラムメイン
054 *                  : (DU 表示)
055 * Include          :
056 * Declaration      : int delay( int cnt )
057 * Description      : ソフトウェアウェイト
058 *                  : cnt 分 for 文を回す
059 *                  :
060 *                  :
061 *                  :
062 *                  :
063 * Limitation       :
064 *                  :
065 * Argument         : cnt
066 * Return Value     : none
067 * Calling Functions :
068 *""FUNC COMMENT END""*************************************************/
069 void delay( int cnt )
070 {
071   int i;
072   for(i=0;i<cnt;i++);
073 }
074
075 /*""FUNC COMMENT""*****************************************************
076 * ID               :
077 * Outline          : サンプルプログラムメイン
078 *                  : (DU 表示)
079 * Include          :
080 * Declaration      : int scif_init(void)
081 * Description      : SCIF の初期設定
082 *                  :
083 *                  :
084 *                  :
085 *                  :
086 *                  :
087 * Limitation       :
088 *                  :
089 * Argument         : none
090 * Return Value     : -1：ボーレートクロック計算エラー
091 * Calling Functions :
092 *""FUNC COMMENT END""*************************************************/
```

```
093 int scif_init(void)
094 {
095   unsigned short data;
096   int t = -1, cnt = 0;
097
098   SCIF.SCSCR.WORD = 0x0000;       /* TIE, RIE, TE, RE Clear */
099
100   SCIF.SCFCR.BIT.TFCL = 1;        /* Tx FIFO Clear */
101   SCIF.SCFCR.BIT.RFCL = 1;        /* Rx FIFO Clear */
102
103   SCIF.SCFSR.WORD = 0x0000;       /* BRK, DR, TR Clear */
104   SCIF.SCLSR.BIT.ORER = 0;        /* ORER Clear */
105
106 #if defined(CONFIG_SCIF_CLK_EXTERNAL)
107   SCIF.SCSCR.BIT.CKE = 2; /* クロックソース：SCK */
108 #elif  defined(CONFIG_SCIF_CLK_PCLK)
109   SCIF.SCSCR.BIT.CKE = 0; /* クロックソース：PCLK */
110   t = SCBRR_VALUE(CONFIG_BPS, CONFIG_SCIF_CLK_PCLK);
111 #endif /* CONFIG_SCIF_CLK */
112
113   if(t > 0) {
114           while(t >= 256) {
115                   cnt++;
116                   t >> 2;
117           }
118           if(cnt > 3)
119                   return -1;
120
121           SCIF.SCSMR.BIT.CKS = cnt;
122           SCIF.SCBRR = t;
123   }
124   delay(1000);
125
126   SCIF.SCFCR.BIT.RTRG = 0;
127   SCIF.SCFCR.BIT.TTRG = 0;
128   SCIF.SCFCR.BIT.TFCL = 1;        /* Tx FIFO Clear */
129   SCIF.SCFCR.BIT.RFCL = 1;        /* Rx FIFO Clear */
130
131   SCIF.SCFCR.BIT.TFCL = 0;        /* Tx FIFO Not Clear */
132   SCIF.SCFCR.BIT.RFCL = 0;        /* Rx FIFO Not Clear */
133   SCIF.SCSCR.BIT.TE  = 1;
134   SCIF.SCSCR.BIT.RE  = 1;
135   return 0;
136 }
137
138 /*""FUNC COMMENT""*****************************************************
139 * ID                :
```

```
140 * Outline          : サンプルプログラムメイン
141 *                  : (DU 表示)
142 * Include          :
143 * Declaration      : void scif_transmit_data( char *Data )
144 * Description      : SCIF の複数 Byte データ送信
145 *                  :
146 *                  :
147 *                  :
148 *                  :
149 *                  :
150 * Limitation       :
151 *                  :
152 * Argument         : *Data : 送信データ格納
153 * Return Value     : none
154 * Calling Functions :
155 *""FUNC COMMENT END""***********************************************/
156 void      scif_transmit_data( char        *Data )
157 {
158   while( *Data )
159   {
160           while(!(SCIF.SCFSR.BIT.TDFE));  /* 送信データ書き込み許可状態になるまでウエイト */
161           SCIF.SCFTDR = *Data;                                    /* 送信データ設定       */
162           Data++;
163           while(!(SCIF.SCFSR.BIT.TEND));  /* 送信終了待ち */
164           SCIF.SCFSR.BIT.TDFE = 0;
165           SCIF.SCFSR.BIT.TEND = 0;
166   }
167 }
168
169 /*""FUNC COMMENT""*****************************************************
170 * ID               :
171 * Outline          : サンプルプログラムメイン
172 *                  : (DU 表示)
173 * Include          :
174 * Declaration      : void scif_transmit_byte_data( char *Data )
175 * Description      : SCIF の 1Byte データ送信
176 *                  :
177 *                  :
178 *                  :
179 *                  :
180 *                  :
181 * Limitation       :
182 *                  :
183 * Argument         : *Data : 送信データ格納
184 * Return Value     : none
185 * Calling Functions :
186 *""FUNC COMMENT END""*************************************************/
```

```
187 void      scif_transmit_data_byte( char   *Data )
188 {
189          while(!(SCIF.SCFSR.BIT.TDFE));  /* 送信データ書き込み許可状態になるまでウエイト */
190          SCIF.SCFTDR = *Data;                                    /* 送信データ設定       */
191          while(!(SCIF.SCFSR.BIT.TEND));  /* 送信終了待ち */
192          SCIF.SCFSR.BIT.TDFE = 0;
193          SCIF.SCFSR.BIT.TEND = 0;
194 }
195
196 /*""FUNC COMMENT""*****************************************************
197 * ID                :
198 * Outline           : サンプルプログラムメイン
199 *                   : (DU 表示)
200 * Include           :
201 * Declaration       : char scif_recive_data( char *Data )
202 * Description       : SCIF のデータ受信
203 *                   :
204 *                   :
205 *                   :
206 *                   :
207 *                   :
208 * Limitation        :
209 *                   :
210 * Argument          : *Data：受信データ格納
211 * Return Value      : -1：受信エラー
212 * Calling Functions :
213 *""FUNC COMMENT END""*************************************************/
214 char      scif_recive_data( char  *Data )
215 {
216   unsigned char   ReadData, i = 0;
217   char    ret_cd = 0;
218
219   for(;;)
220   {
221          if(( SCIF.SCFSR.BIT.ER  ) ||
222             ( SCIF.SCFSR.BIT.BRK ) ||
223             ( SCIF.SCFSR.BIT.DR  ))              /* エラー発生？ */
224          {
225                 ReadData = SCIF.SCFRDR; /* データダミー読み込み */
226                 ret_cd = -1;    /* 受信エラー設定       */
227                 SCIF.SCFSR.WORD &= 0x0000;      /* エラークリア */
228                 SCIF.SCLSR.WORD &= 0x0000;
229          }
230          else if( SCIF.SCFSR.BIT.RDF )   /* データ受信？ */
231          {
232                 *Data = SCIF.SCFRDR;    /* データ取得   */
233                 SCIF.SCFSR.BIT.RDF = 0; /* 受信サインクリア     */
```

```
234                     SCIF.SCFSR.BIT.DR  = 0; /* 受信サインクリア      */
235                     scif_transmit_data_byte( Data );
236                     if( *Data == '¥n' )     /* 取得データが CR？     */
237                     {
238                             break;  /* 処理終了      */
239                     }
240                     if( *Data == 0x0d )     /* 取得データが CR？     */
241                     {
242                             break;  /* 処理終了      */
243                     }
244                     Data++; /* データ取得アドレス次設定      */
245                     if( ++i == 4 )
246                     {
247                             ret_cd = -1;
248                     }
249             }
250             if( ret_cd == -1 )
251             {
252                     break;
253             }
254     }
255     return( ret_cd );
256 }
257
258
```

サンプルプログラムリスト”scif.h”

“scif.c”で使用するヘッダーです。

```
01
02 #ifndef _SCIF_H
03 #define _SCIF_H
04
05 #include "iodefine.h"
06 #include "config.h"
07
08 #if defined(CONFIG_SCIF0)
09 #define SCIF   (*(volatile struct st_scif    *)0xFFEA0000) /* SCIF0   Address */
10 #elif defined(CONFIG_SCIF1)
11 #define SCIF   (*(volatile struct st_scif    *)0xFFEB0000) /* SCIF1   Address */
12 #elif defined(CONFIG_SCIF2)
13 #define SCIF   (*(volatile struct st_scif    *)0xFFEC0000) /* SCIF2   Address */
14 #elif defined(CONFIG_SCIF3)
15 #define SCIF   (*(volatile struct st_scif    *)0xFFED0000) /* SCIF3   Address */
16 #elif defined(CONFIG_SCIF4)
17 #define SCIF   (*(volatile struct st_scif    *)0xFFEE0000) /* SCIF4   Address */
18 #elif defined(CONFIG_SCIF5)
19 #define SCIF   (*(volatile struct st_scif    *)0xFFEF0000) /* SCIF5   Address */
20 #endif /* CONFIG_SCIFn */
21
22 //#define SCBRR_VALUE(bps, clk) ((clk+16*bps)/(16*bps)-1)
23 #define SCBRR_VALUE(bps, clk) ((clk)/(32*bps)-1)
24
25 /* SCFCR */
26 #define    RTRG1   0
27 #define    RTRG16  1
28 #define    RTRG32  2
29 #define    RTRG48  3
30 #define    TTRG32  0
31 #define    TTRG16  1
32 #define    TTRG2   2
33 #define    TTRG0   3
34
35
36
37 #endif /* _SCIF_H */
```

サンプルプログラムリスト”intprg.c”

DU 割込み処理関数と DMAC-ch0 割込み処理関数をハンドラに登録しています。

```
__途中省略…..
231 /* H'620 DMAC0 interrupt */
232 void INT_DMAC_DMINT0(void)
233 {
234   irq_disable( _DMAC0 );
235   dmac0_irq();
236 }
__途中省略…..
508 /* H'D80 DU interrupt */
509 void INT_DU_DUI(void)
510 {
511   irq_disable( _DU );
512
513   if( DU.DSSR.BIT.TVR & DU.DIER.BIT.TVE)
514           du_tvr_irq();
515   else if( DU.DSSR.BIT.FRM  & DU.DIER.BIT.FRE)
516           du_frm_irq();
517   else if( DU.DSSR.BIT.VBK  & DU.DIER.BIT.VBE)
518           du_vbk_irq();
519   else if( DU.DSSR.BIT.RINT  & DU.DIER.BIT.RIE)
520           du_rint_irq();
521   else if( DU.DSSR.BIT.HBK  & DU.DIER.BIT.HBE)
522           du_hbk_irq();
523
524   irq_enable( _DU );
525 }
__途中省略…..
```

サンプルプログラムリスト”intc.h”

周辺モジュールの割込みの許可/禁止，優先順位を設定しています。

```
001 /***********************************************************************
002 *
003 * Device      : SH-4A/SH7785
004 *
005 * File Name   : intc.h
006 *
007 * Abstract    : INTC .
008 *
009 * History     : 1.00  (2010-09-30)  [Hardware Manual Revision : 1.00]
010 *
011 * Copyright(c) 2010 Renesas Electronics Corp.
012 *               And Renesas Solutions Corp.,All Rights Reserved.
013 *
014 ***********************************************************************/
015
016 #ifndef _INTC_H_
017 #define _INTC_H_
018
019 static enum {
020   _TMU0,
021   _TMU1,
022   _TMU2,
023   _TMU2_IC,
024   _TMU3,
025   _TMU4,
026   _TMU5,
027   _SCIF0,
028   _SCIF1,
029   _SCIF2,
030   _SCIF3,
031   _SCIF4,
032   _SCIF5,
033   _WDT,
034   _H_UDI,
035   _DMAC0,
036   _DMAC1,
037   _HAC0,
038   _HAC1,
039   _PCIC0,
040   _PCIC1,
041   _PCIC2,
042   _PCIC3,
043   _PCIC4,
044   _PCIC5,
045   _SIOF,
```

```
046   _HSPI,
047   _MMCIF,
048   _FLCTL,
049   _GPIO,
050   _SSI0,
051   _SSI1,
052   _DU,
053   _GDTA
054 }int_num;
055
056 static enum {
057   PRI0, PRI1, PRI2, PRI3, PRI4, PRI5, PRI6, PRI7, PRI8, PRI9, PRI10,
058   PRI11, PRI12, PRI13, PRI14, PRI15, PRI16, PRI17, PRI18, PRI19, PRI20,
059   PRI21, PRI22, PRI23, PRI24, PRI25, PRI26, PRI27, PRI28, PRI29, PRI30,
060   PRI31
061 }priority;
062
063 struct intc2_table {
064   int pri;                        /* 優先順位 */
065   int pri_pos;            /* 優先順位ビット位置 */
066   char pri_add;           /* 優先順位アドレス OFFSET */
067   int st_pos;                     /* 割込み要因ビット位置 */
068 };
069
070 static struct intc2_table intc_table[] = {
071   /*      pri,    pri_pos,        pri_add,        st_pos */
072   {       PRI0,   24,                     0x00,           0       },              /* TMU0 */
073   {       PRI0,   16,                     0x00,           0       },              /* TMU1 */
074   {       PRI0,    8,                     0x00,           0       },              /* TMU2 */
075   {       PRI0,    0,                     0x00,           0       },              /* TMU2_IC */
076   {       PRI0,   24,                     0x04,           1       },              /* TMU3 */
077   {       PRI0,   16,                     0x04,           1       },              /* TMU4 */
078   {       PRI0,    8,                     0x04,           1       },              /* TMU5 */
079   {       PRI0,   24,                     0x08,           2       },              /* SCIF0 */
080   {       PRI0,   16,                     0x08,           3       },              /* SCIF1 */
081   {       PRI0,    8,                     0x08,           4       },              /* SCIF2 */
082   {       PRI0,    0,                     0x08,           5       },              /* SCIF3 */
083   {       PRI0,   24,                     0x0C,           6       },              /* SCIF4 */
084   {       PRI0,   16,                     0x0C,           7       },              /* SCIF5 */
085   {       PRI0,    8,                     0x0C,           8       },              /* WDT */
086   {       PRI0,   24,                     0x10,           9       },              /* H_UDI */
087   {       PRI15,  16,                     0x10,           10      },              /* DMAC0 */
088   {       PRI0,    8,                     0x10,           11      },              /* DMAC1 */
089   {       PRI0,   24,                     0x14,           12      },              /* HAC0 */
090   {       PRI0,   16,                     0x14,           13      },              /* HAC1 */
091   {       PRI0,    8,                     0x14,           14      },              /* PCI0 */
092   {       PRI0,    0,                     0x14,           15      },              /* PCI1 */
```

```
093   {       PRIO,    24,                    0x18,           16      },              /* PCI2 */
094   {       PRIO,    16,                    0x18,           17      },              /* PCI3 */
095   {       PRIO,     8,                    0x18,           18      },              /* PCI4 */
096   {       PRIO,     0,                    0x18,           19      },              /* PCI5 */
097   {       PRIO,    24,                    0x1C,           20      },              /* SIOF */
098   {       PRIO,    16,                    0x1C,           21      },              /* HSPI */
099   {       PRIO,     8,                    0x1C,           22      },              /* MMCIF */
100   {       PRIO,    24,                    0x20,           23      },              /* FLCTL */
101   {       PRIO,    16,                    0x20,           24      },              /* GPIO */
102   {       PRIO,     8,                    0x20,           25      },              /* SSIO */
103   {       PRIO,     0,                    0x20,           26      },              /* SSI1 */
104   {       PRI14,   24,                    0x24,           27      },              /* DU */
105   {       PRIO,    16,                    0x24,           28      },              /* GDTA */
106 };
107
108 #define   INTC2_OFFSET    0xFFD40000
109 /*""FUNC COMMENT""*****************************************************
110 * ID                :
111 * Outline           : サンプルプログラムメイン
112 *                   : (DU 表示)
113 * Include           :
114 * Declaration       : static void irq_enable( int module )
115 * Description       : INT2 の内部周辺モジュールの割込みの許可&優先順位を
116 *                   : 設定します。
117 *                   :
118 *                   :
119 *                   :
120 *                   :
121 * Limitation        :
122 *                   :
123 * Argument          : none
124 * Return Value      : none
125 * Calling Functions :
126 *""FUNC COMMENT END""*************************************************/
127 static void irq_enable( int module )
128 {
129   unsigned long tmp;
130   unsigned long address;
131   /* 優先順位設定 */
132   address = INTC2_OFFSET + intc_table[module].pri_add;
133   tmp = *(unsigned long *)address;
134   tmp |= (intc_table[module].pri << intc_table[module].pri_pos);
135   *(unsigned long *)address = tmp;
136
137   /* 割込みマスククリア */
138   INTC.INT2MSKCLR.LONG = (1 << intc_table[module].st_pos);
139 }
```

```
140
141 /*""FUNC COMMENT""************************************************
142 * ID                :
143 * Outline           : サンプルプログラムメイン
144 *                   : (DU 表示)
145 * Include           :
146 * Declaration       : static void irq_disable( int module )
147 * Description       : INT2 の内部周辺モジュールの割込みの禁止を
148 *                   : 設定します。
149 *                   :
150 *                   :
151 *                   :
152 *                   :
153 * Limitation        :
154 *                   :
155 * Argument          : none
156 * Return Value      : none
157 * Calling Functions :
158 *""FUNC COMMENT END""************************************************/
159 static void irq_disable( int module )
160 {
161   unsigned long address;
162
163   /* 割込みマスク設定 */
164   INTC.INT2MSKR.LONG = (1 << intc_table[module].st_pos);
165 }
166
167 #endif /* _INTC_H_ */
```

サンプルプログラムリスト”lowlevelinit.inc”

「SH7785 グループアプリケーションノート SH7785 初期設定例(R01AN0242JJ0101)から一部修正しています。

― DBSC2 の設定値を 1.3 章の適用条件となるように変更しています。

```
__途中省略.....
034 DBSC2_DBCONF_D:                 .equ    H'009A0002
035 DBSC2_DBTR0_D:                  .equ    H'050D1604
036 DBSC2_DBTR1_D:                  .equ    H'00040204
037 DBSC2_DBTR2_D:                  .equ    H'02120708
038 DBSC2_DBFREQ_D1:                .equ    H'00000000
039 DBSC2_DBFREQ_D2:                .equ    H'00000100
040 DBSC2_DBDICODTOCD_D:            .equ    H'00000E07
041
042 DBSC2_DBMRCNT_D_EMRS2:          .equ    H'00020000
043 DBSC2_DBMRCNT_D_EMRS3:          .equ    H'00030000
044 DBSC2_DBMRCNT_D_EMRS1_1:        .equ    H'00010004
045 DBSC2_DBMRCNT_D_EMRS1_2:        .equ    H'00010384
046 DBSC2_DBMRCNT_D_MRS_1:          .equ    H'00000952
047 DBSC2_DBMRCNT_D_MRS_2:          .equ    H'00000852
__途中省略.....
```

5. 実行結果

上記プログラムを実行すると、

- プレーン 4 面の表示
- プレーン 1～3 の表示位置の移動
- プレーン 4 の画像の増減
- シリアルコンソールからの各プレーンの表示の ON/OFF 制御

が繰り返されます。

6. 参考ドキュメント

- ソフトウェアマニュアル
  SH4-A ソフトウェアマニュアル(RJJ09B0090)
  (最新版をルネサスエレクトロニクスホームページから入手してください)

- ハードウェアマニュアル
  SH7785 グループハードウェアマニュアル(RJJ09B0285)
  (最新版をルネサスエレクトロニクスホームページから入手してください)

ホームページとサポート窓口

ルネサス エレクトロニクスホームページ
http://japan.renesas.com/

お問合せ先
http://japan.renesas.com/inquiry

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2010.12.01	—	初版発行
1.01	2010.12.01	全体	誤記修正

製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本文を参照してください。なお、本マニュアルの本文と異なる記載がある場合は、本文の記載が優先するものとします。

1. 未使用端子の処理

   【注意】未使用端子は、本文の「未使用端子の処理」に従って処理してください。

   CMOS製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI周辺のノイズが印加され、LSI内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。未使用端子は、本文「未使用端子の処理」で説明する指示に従い処理してください。

2. 電源投入時の処置

   【注意】電源投入時は，製品の状態は不定です。

   電源投入時には、LSIの内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。

   外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。

   同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

3. リザーブアドレスのアクセス禁止

   【注意】リザーブアドレスのアクセスを禁止します。

   アドレス領域には、将来の機能拡張用に割り付けられているリザーブアドレスがあります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

4. クロックについて

   【注意】リセット時は、クロックが安定した後、リセットを解除してください。

   プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。

   リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。

5. 製品間の相違について

   【注意】型名の異なる製品に変更する場合は、事前に問題ないことをご確認下さい。

   同じグループのマイコンでも型名が違うと、内部メモリ、レイアウトパターンの相違などにより、特性が異なる場合があります。型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。

ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したものですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。

   標準水準： コンピュータ、OA機器、通信機器、計測機器、AV機器、家電、工作機械、パーソナル機器、産業用ロボット

   高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）

   特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制するRoHS指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

注1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社がその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

注2. 本資料において使用されている「当社製品」とは、注1において定義された当社の開発、製造製品をいいます。

ルネサス エレクトロニクス株式会社

■営業お問合せ窓口 http://www.renesas.com

※営業お問合せ窓口の住所・電話番号は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス販売株式会社　〒100-0004　千代田区大手町2-6-2（日本ビル） (03)5201-5307

■技術的なお問合せおよび資料のご請求は下記へどうぞ。
総合お問合せ窓口：http://japan.renesas.com/inquiry

Colophon 1.0