お客様各位

カタログ等資料中の旧社名の扱いについて

2010年4月1日を以ってNECエレクトロニクス株式会社及び株式会社ルネサステクノロジが合併し、両社の全ての事業が当社に承継されております。従いまして、本資料中には旧社名での表記が残っておりますが、当社の資料として有効ですので、ご理解の程宜しくお願い申し上げます。

ルネサスエレクトロニクス ホームページ（http://www.renesas.com）

2010年4月1日
ルネサスエレクトロニクス株式会社

【発行】ルネサスエレクトロニクス株式会社（http://www.renesas.com）

【問い合わせ先】http://japan.renesas.com/inquiry

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。

SH7618 グループ
HIF ブートモード

要旨

SH7618 は，RISC (Reduced Instruction Set Computer) タイプの SH-2 CPU をコアにして，イーサネットコントローラおよび，他のマイコンとの接続を容易にする HIF (Host Interface) モジュールを内蔵しています。

HIF は，SH7618 と HIF に接続したデバイス間で，容易にデータ転送を行うためのモジュールです。HIF の機能の一つとして HIF ブートモードがあります。HIF ブートモードでは，SH7618 が HIFRAM (HIF 専用内蔵 RAM) に格納したプログラム (ブートプログラム) で起動します。ブートプログラムをデータ転送プログラムとすることで，外部デバイスが HIFRAM を介して，アプリケーションプログラムを SH7618 に転送できます。プログラムの転送先は，SH7618 の内蔵 RAM または外部バスに接続した RAM になります。外部デバイスが SH7618 にプログラムを転送することで，SH7618 専用のプログラム格納 ROM を削減できます。

本アプリケーションノートは，SH7618 の HIF ブートモードでの起動方法について述べており，ユーザソフトウェア設計の際のご参考として役立てていただくようまとめたものです。

動作確認デバイス

SH7618

目次

1. HIF の概要

1.1 HIF の特長

1.1.1 HIF および HIFRAM の特長

以下に HIF および HIFRAM の特長を示します。

- HIF イネーブル (HIFEBL) 端子へ L レベルを入力することで, HIF 端子をハイインピーダンス状態に設定可能です。HIF 端子をハイインピーダンスとすることで，HIF のモジュールスタンバイ状態，または外部デバイスのスタンバイ状態での HIF 端子のドライブによるデバイス破壊を防止します。
- SH7618 の CPU および HIF に接続した外部デバイスからアクセス可能な HIF 専用 RAM (HIFRAM) を 1KB ×2 バンク内蔵しています。また，異なるバンクに SH7618 の CPU と外部デバイスが同時にアクセス可能なため，効率的なデータ転送が可能です。
- HIF のレジスタで，SH7618 の CPU と外部デバイスがアクセスする HIFRAM のバンクを別々に設定可能です。
- SH バスとは独立した HIF 専用のバスを有しているため，SH7618 の動作とは非同期 (SH7618 がバス権を開放せず) に，HIF に接続した外部デバイスから HIFRAM へのアクセスが可能です。
- HIF に接続した外部デバイスから HIF を介して 32 ビット単位でアクセス可能です。
- SH7618 の CPU からは 8/16/32 ビット単位でアクセス可能です。
- HIFRAM 上のプログラムから起動する HIF ブートモード機能を有しています。

HIFRAM は 2 バンク構成で，それぞれ共通のアドレスにマッピングされています。また，HIF に接続した外部デバイスは，HIFADR レジスタで HIFRAM のアドレスを指定しアクセスします。表 1 に HIFRAM のアドレスを示します。

表 1　HIFRAM のアドレス

分類	開始アドレス	終了アドレス	サイズ
SH7618 の CPU からみたマップ	H'F84E0000	H'F84E03FF	1KB
外部デバイスからみたマップ*	H'0000	H'03FF	1KB

【注】 * 外部デバイスが HIFADR レジスタにセットする値です。

1.1.2 外部デバイスの接続

表 2 に，HIF の端子構成を示します。

表 2　HIF の端子構成

名称	記号	入出力	機能
HIF データ端子	HIFD15～HIFD0	入出力	HIF へのアドレス/データ/コマンド入出力
HIF チップセレクト	$\overline{\text{HIFCS}}$	入力	HIF へのチップセレクト入力
HIF レジスタセレクト	HIFRS	入力	HIF へのアクセスレジスタの切り替え 0: HIF インデックスレジスタで指定したレジスタへのアクセス 1: HIF インデックスレジスタへのライト/HIF 汎用ステータスレジスタのリード
HIF ライト	$\overline{\text{HIFWR}}$	入力	ライトストローブ信号 外部デバイスが HIF へデータを書き込む場合，ローレベルを入力します。
HIF リード	$\overline{\text{HIFRD}}$	入力	リードストローブ信号 外部デバイスが HIF からデータを読み出す場合，ローレベルを入力します。
HIF 割り込み	$\overline{\text{HIFINT}}$	出力	HIF から外部デバイスへの割り込み要求
HIF モード	HIFMD	入力	HIF ブートモードで起動するかどうかを指定 ハイレベルを入力した状態でパワーオンリセットを解除することで，SH7618 は HIF ブートモードで起動します。
HIFDMAC 転送要求	HIFDREQ	出力	外部デバイスに対して HIFRAM への DMA 転送要求
HIF ブートレディ	HIFRDY	出力	SH7618 内部で，HIF モジュールのリセットが解除され，外部デバイスから HIF モジュールへのアクセスが受付可能になったことを示します。 SH7618 のリセット入力端子のネゲートが検出されてから，周辺クロック換算で最大 10 クロック後にアサート出力されます。
HIF 端子イネーブル	HIFEBL	入力	HIF の端子の活性/非活性を選択 0: HIF 端子が非活性状態 (ハイインピーダンス状態) 1: HIF 端子が活性状態 (HIF 端子を使用可能)

図 1 に HIF と外部デバイスの接続例を示します。

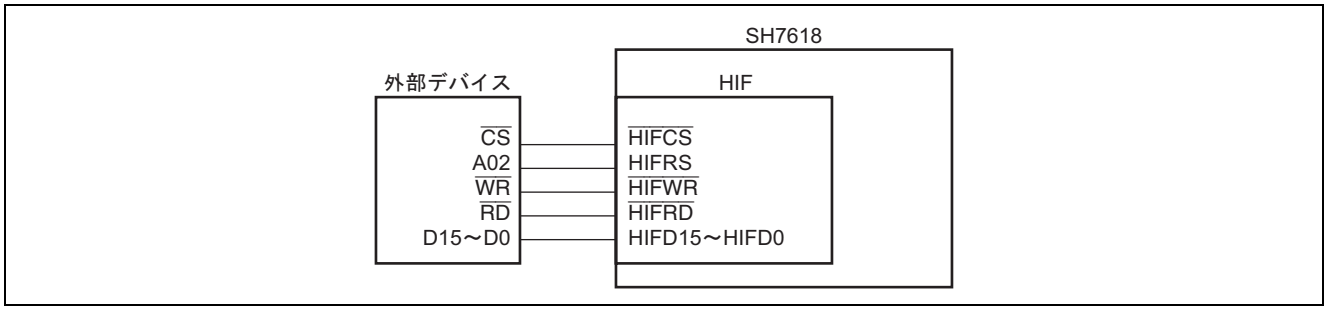


図 1　外部デバイスと HIF の接続例

1.2 外部デバイスからのアクセス

1.2.1 HIF の各レジスタへのアクセス

HIF のレジスタを以下に示します。

- HIF インデックスレジスタ (HIFIDX)
- HIF 汎用ステータスレジスタ (HIFGSR)
- HIF ステータス/コントロールレジスタ (HIFSCR)
- HIF メモリ制御レジスタ (HIFMCR)
- HIF 内部割り込み制御レジスタ (HIFIICR)
- HIF 外部割り込み制御レジスタ (HIFEICR)
- HIF アドレスレジスタ (HIFADR)
- HIF データレジスタ (HIFDATA)
- HIF ブート制御レジスタ (HIFBCR)
- HIFDREQ トリガレジスタ (HIFDTR)
- HIF バンク割り込み制御レジスタ (HIFBICR)

外部デバイスから HIF のレジスタへは，HIFIDX レジスタでアクセスするレジスタおよびそのときのワード位置 (上位 2 バイト or 下位 2 バイト) を指定してアクセスします。HIFIDX レジスタをアクセスするか，HIFIDX レジスタで指定したレジスタをアクセスするかは，HIFRS 端子で選択します。

表 3 に $\overline{\text{HIFCS}}$, HIFRS, $\overline{\text{HIFWR}}$, $\overline{\text{HIFRD}}$ 端子の組み合わせとそのときの HIF の動作を示します。

表 3　端子設定と HIF 動作の対応

$\overline{\text{HIFCS}}$	HIFRS	$\overline{\text{HIFWR}}$	$\overline{\text{HIFRD}}$	外部デバイスからのアクセス
1	*	*	*	**ノーオペレーション**
0	0	1	0	HIFIDX **レジスタで指定したレジスタからのリード**
0	0	0	1	HIFIDX **レジスタで指定したレジスタへのライト**
0	1	1	0	HIFGSR **レジスタからのリード**
0	1	0	1	HIFIDX **レジスタへのライト**
0	*	1	1	**ノーオペレーション**
0	*	0	0	**設定禁止**

【注】 *: Don’t Care

1.2.2 HIFRAM への書き込み

以下に外部デバイスから HIFRAM への書き込み手順を，図 2 に HIFRAM へのライトシーケンスを示します。

1. HIFADR レジスタで書き込む HIFRAM のアドレスを指定
   HIFIDX レジスタで HIFADR レジスタの下位 16 ビットを指定し，書き込む HIFRAM のアドレスを指定します。
2. HIFDATA レジスタにデータを書き込む
   HIFIDX レジスタで HIFDATA レジスタの上位 16 ビットを指定しデータを書き込みます。続けて HIFDATA レジスタへアクセスし，HIFDATA レジスタの下位 16 ビットへデータを書き込みます。
3. HIFMCR レジスタの WT ビットを 1 にセット
   HIFMCR レジスタの WT ビットを 1 にセットすると，HIFADR レジスタで指定した HIFRAM のアドレスに，HIFDATA レジスタの 32 ビットのデータが書き込まれます。その後，HIFMCR レジスタの WT ビットは自動的に 0 クリアされます。

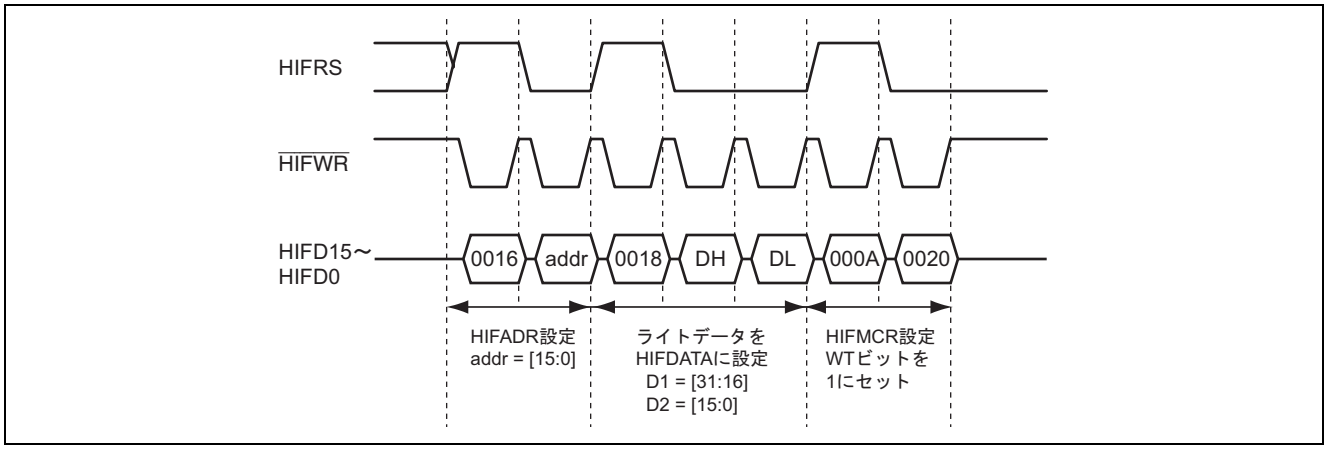


図 2　HIFRAM への 1 回のライトシーケンス

また，一度 HIFRAM のアドレスを指定して，それ以降のアドレスに対して，連続して書き込むことが可能です。以下に連続書き込みの手順を，図 3 に連続ライトシーケンスを示します。

1. HIFADR レジスタで HIFRAM の書き込み開始アドレスを指定
   HIFIDX レジスタで HIFADR レジスタの下位 16 ビットを指定し，書き込みを開始する HIFRAM のアドレスを指定します。
2. HIFDATA レジスタに最初のデータを書き込む
   HIFIDX レジスタで HIFDATA レジスタの上位 16 ビットを指定しデータを書き込みます。続けて HIFDATA レジスタへアクセスし，HIFDATA レジスタの下位 16 ビットへデータを書き込みます。
3. HIFRAM への連続書き込みに設定
   HIFMCR レジスタの WT ビットと LOCK ビットを同時に 1 にセットすることで，連続書き込みが可能になります。このとき，HIFMCR レジスタの AI/AD ビットの設定により，HIFRAM のアドレスをインクリメント (+4) するかデクリメント (-4) するかを選択可能です。
4. HIFIDX レジスタで HIFDATA レジスタの上位 16 ビットを指定
5. HIFDATA レジスタへデータを書き込む
   連続して HIFDATA レジスタにアクセスすると，HIFMCR レジスタの AI/AD ビットにより HIFADR レジスタの値が変化し，HIFRAM の連続したアドレスへデータが書き込まれます。

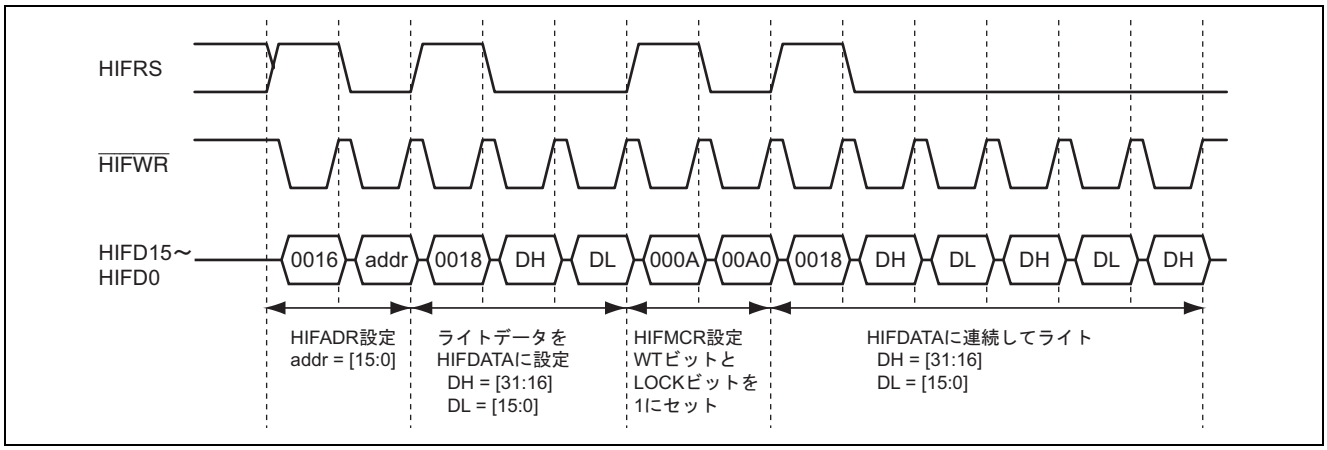


図 3　HIFRAM への連続ライトシーケンス

1.2.3 HIFRAM からの読み出し

以下に HIFRAM からの読み出し手順を，図 4 にリードシーケンスを示します。

1. HIFADR レジスタで読み出す HIFRAM のアドレスを指定
   HIFIDX レジスタで HIFADR レジスタの下位 16 ビットを指定し，読み出す HIFRAM のアドレスを指定します。
2. HIFMCR レジスタの RD ビットを 1 にセット
   HIFMCR レジスタの RD ビットを 1 にセットすると，HIFADR レジスタで指定したアドレスに対応した HIFRAM のデータが HIFDATA レジスタに読み出されます。HIFDATA レジスタに HIFRAM のデータが読み出されると，RD ビットは自動的に 0 クリアされます。
3. HIFDATA レジスタからデータを読み出す
   HIFIDX レジスタで HIFDATA レジスタの上位 16 ビットを指定し，HIFDATA レジスタを読み出します。続けて HIFDATA レジスタを読み出すことで，HIFDATA レジスタの下位 16 ビットを読み出します。

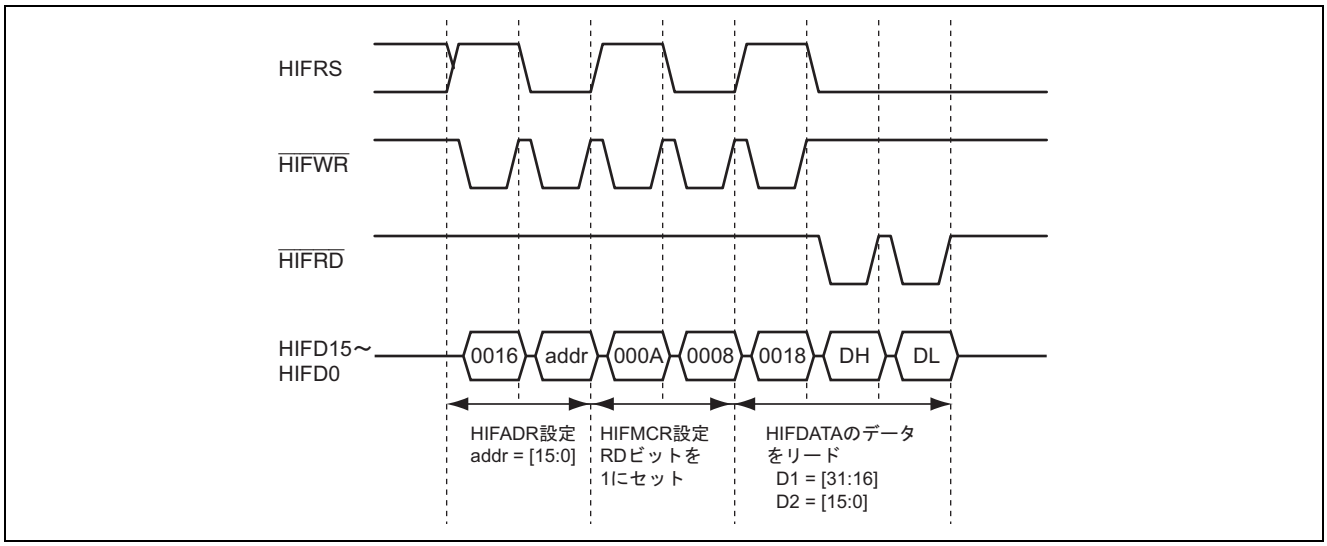


図 4　HIFRAM からの 1 回のリードシーケンス

また，一度 HIFRAM のアドレスを指定して，それ以降のアドレスに対して，連続して読み出すことが可能です。以下に連続読み出しの手順を，図 5 に連続リードシーケンスを示します。

1. HIFADR レジスタで HIFRAM の読み出し開始アドレスを指定
   HIFIDX レジスタで HIFADR レジスタの下位 16 ビットを指定し，読み出しを開始する HIFRAM のアドレスを指定します。
2. HIFRAM からの連続読み出しに設定
   HIFMCR レジスタの RD ビットと LOCK ビットを同時に 1 にセットすることで，連続読み出しが可能になります。このとき，HIFMCR レジスタの AI/AD ビットの設定により，HIFRAM のアドレスをインクリメント (+4) するかデクリメント (-4) するかを選択可能です。
3. HIFIDX レジスタで HIFDATA レジスタの上位 16 ビットを指定
4. HIFDATA レジスタからデータを読み出す
   連続して HIFDATA レジスタを読み出すと，HIFMCR レジスタの AI/AD ビットにより HIFADR レジスタの値が変化し，HIFRAM のデータが読み出されます。

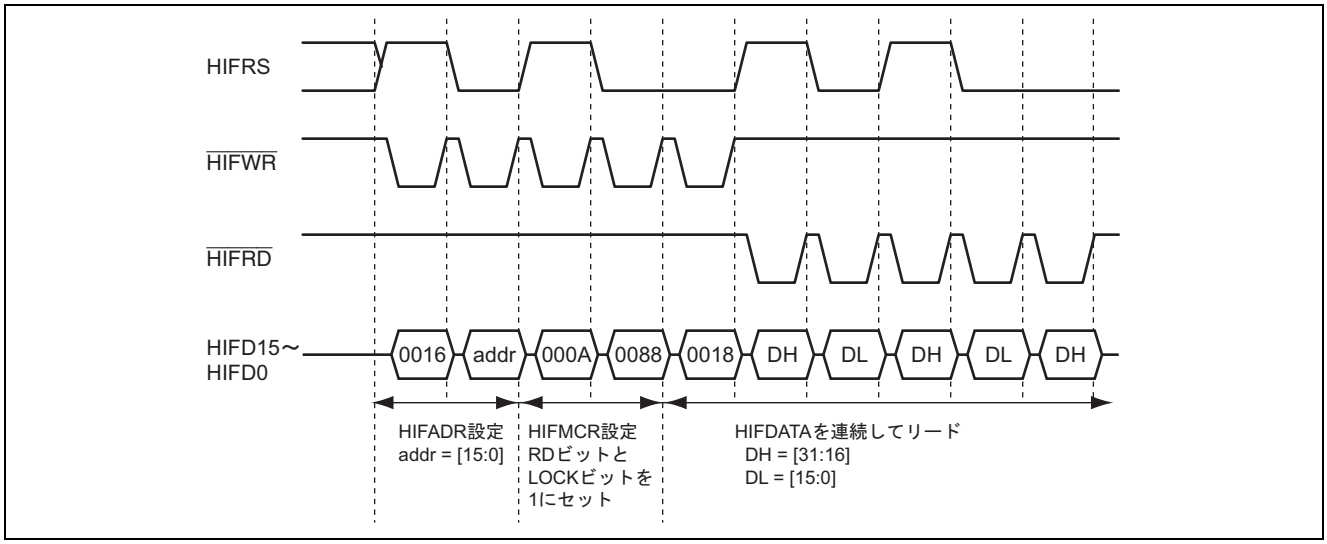


図 5　HIFRAM からの連続リードシーケンス

1.3 HIF ブートモード

SH7618 には，HIFRAM 上のプログラムから起動する，HIF ブートモード機能があります。

HIF ブートモードでは, HIF に接続した外部デバイスが SH7618 ブートプログラムを HIFRAM に書き込み, SH7618 を HIFRAM 上のプログラムから起動します。ブートプログラムを外部デバイスのプログラムと同じ ROM に格納することで，SH7618 の起動用 ROM を削減できます。

以下に，SH7618 を HIF ブートモードで起動するまでの手順を，図 6 に HIF ブートモードで起動するまでのシーケンスを示します。

1. SH7618 を HIF ブートモードにセット
   HIFMD 端子が H レベルの状態でパワーオンリセットを解除することで，SH7618 を HIF ブートモードに設定します (パワーオンリセット解除後も HIFMD 端子には H レベルを入力してください)。
2. HIF に接続した外部デバイスが HIFRAM に SH7618 のブートプログラムを書き込む
   SH7618 は HIF ブートモードで起動すると待機状態となるため，その間に外部デバイスが HIFRAM に SH7618 のブートプログラムを書き込みます。また，SH7618 は HIF ブートモードで起動すると，HIFRAM の内容がエリア 0 の前半 32MB の先頭 1KB にマッピングされるため, HIFRAM の先頭アドレスにはリセットベクタを書き込みます。
3. SH7618 を起動
   HIFRAM へのブートプログラムの書き込み終了後，外部デバイスが HIFBCR レジスタの AC ビットを 0 クリアすると，SH7618 が HIFRAM の先頭アドレスをリセットベクタとして読み込み，プログラムを実行します。

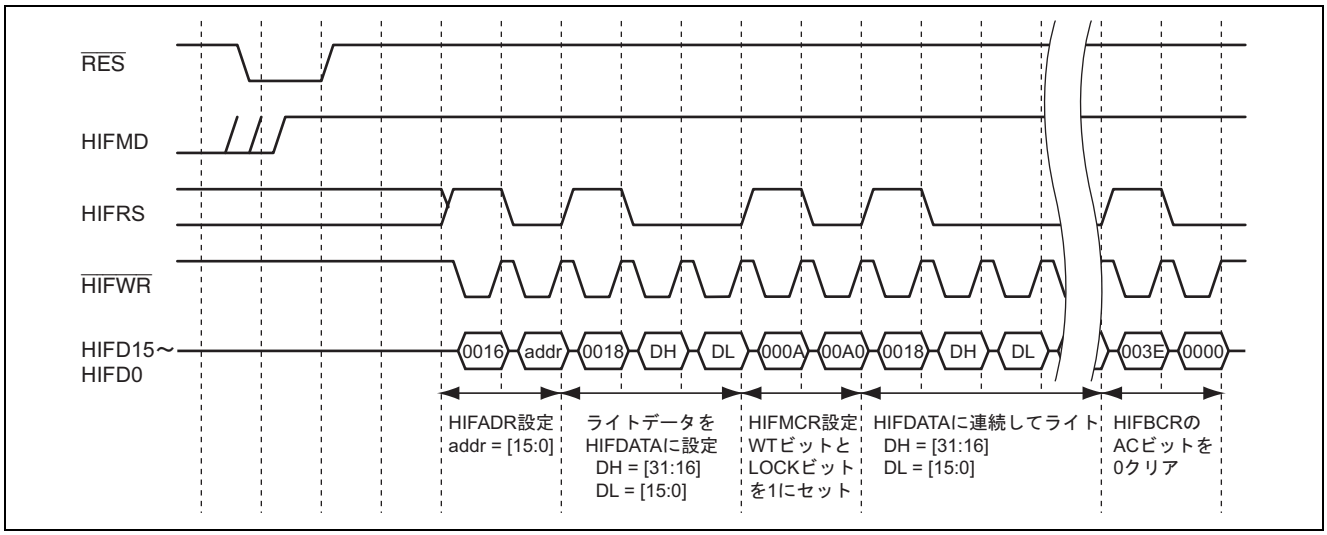


図 6　HIF ブートシーケンス

図 7 に HIF ブート時の HIFRAM のメモリマップを示します。

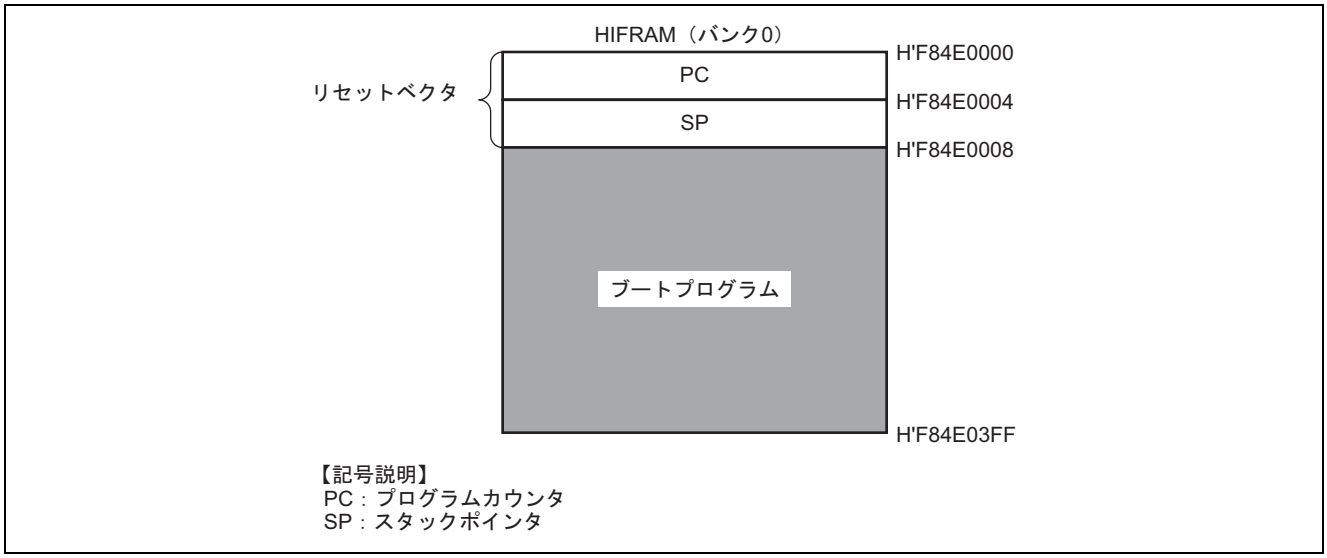


図 7　HIF ブートモード時の HIFRAM のメモリマップ

HIF ブートモード時，SH7618 の CPU および外部デバイスからは HIFRAM のバンク 0 をアクセスしますので，ブートプログラムは HIFRAM のバンク 0 に格納されます。HIFRAM のバンク 1 にはブートプログラムが書き込まれないため，HIFRAM 上でプログラムを実行中に SH7618 の CPU からアクセスするバンクを変更しないでください。

2. HIF ブート応用例

2.1 HIF ブートモードでの起動

2.1.1 概要

本タスク例では，SH7618 を HIF ブートモードで起動します。

HIF に接続した外部デバイスが，HIFRAM へブートプログラムを書き込み，SH7618 を起動します。ブートプログラムを外部デバイスのプログラムと同じ ROM に格納しておくことで，SH7618 の起動用 ROM を削減できます。

2.1.2 仕様

1. HIF に接続する外部デバイスとして SH7641 を使用し, SH7641 の CS5A 空間 (H'B4000000～H'B5FFFFFF*) に HIF を接続します。
2. HIFMD 端子が H レベルの状態でパワーオンリセットを解除し，SH7618 を HIF ブートモードにセットします。
3. SH7641 がブートプログラムを HIFRAM に書き込みます。
4. ブートプログラムの書き込み終了後，SH7641 が HIFBCR レジスタの AC ビットをクリアします。
5. SH7618 は，HIFRAM の先頭アドレスに書き込まれたデータをリセットベクタとして読み込み，起動します。
6. 本タスク例において，SH7641 の動作周波数は，CPU クロック 100MHz，外部バスクロック 50MHz，周辺クロック 25MHz とします。また，SH7618 は HIF ブートモードで HIFBCR レジスタの AC ビットを 0 クリアされるまで HIF のレジスタ以外の設定が不可能なため，動作周波数は内部クロック 50MHz，外部バスクロック 50MHz，周辺クロック 12.5MHz (モード 5，入力クロック 25MHz) となります。

【注】 * **HIF をキャッシュエリアに接続した場合，HIFGSR の値を正常にリードできない可能性があります。そのため，HIF の接続はノンキャッシュエリアとしています。**

図 8 に SH7618 と SH7641 の接続例を示します。

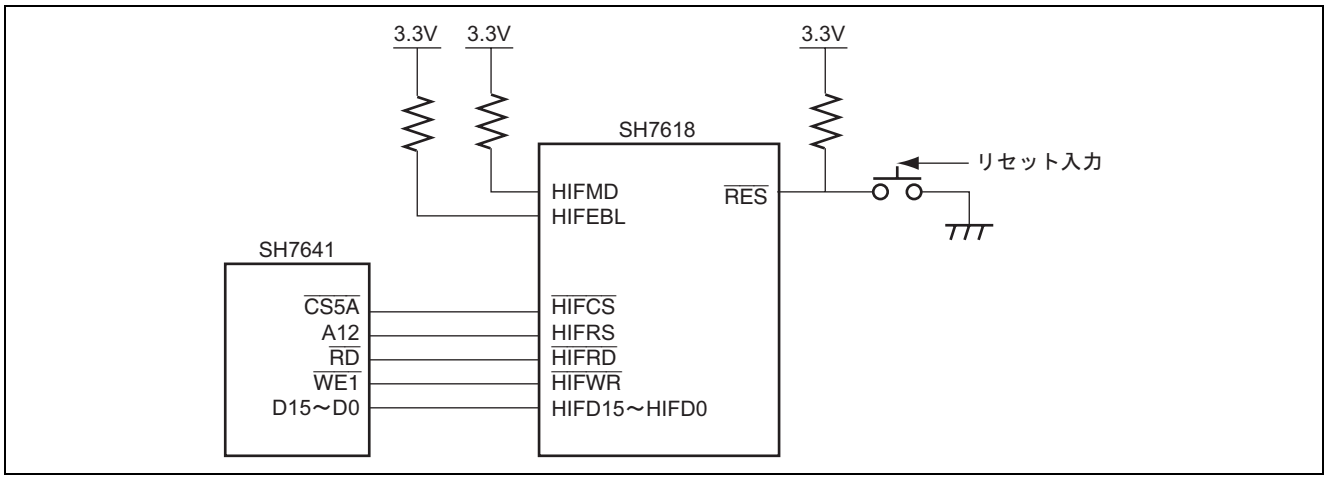


図 8 接続例

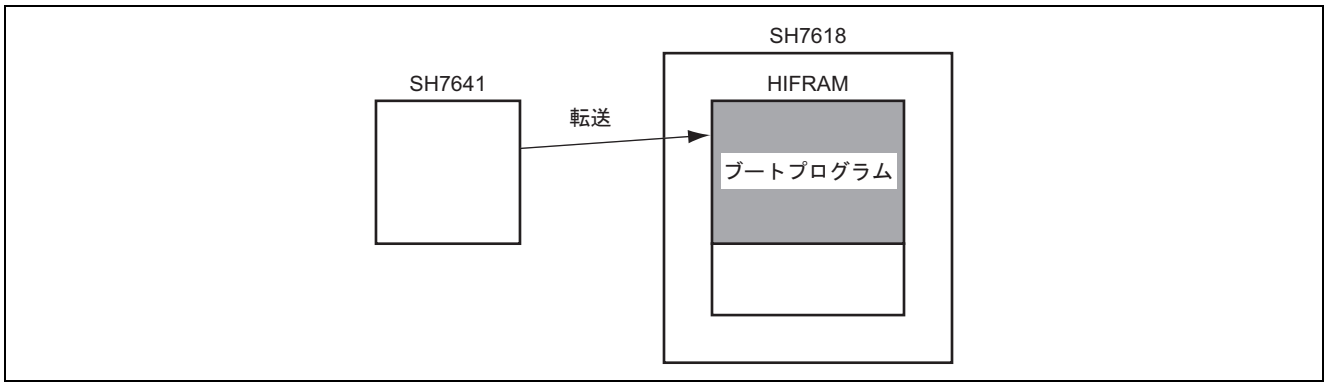


図 9　HIF ブートモードでの起動

図 10 に HIF ブート時の HIFRAM のメモリマップを示します。

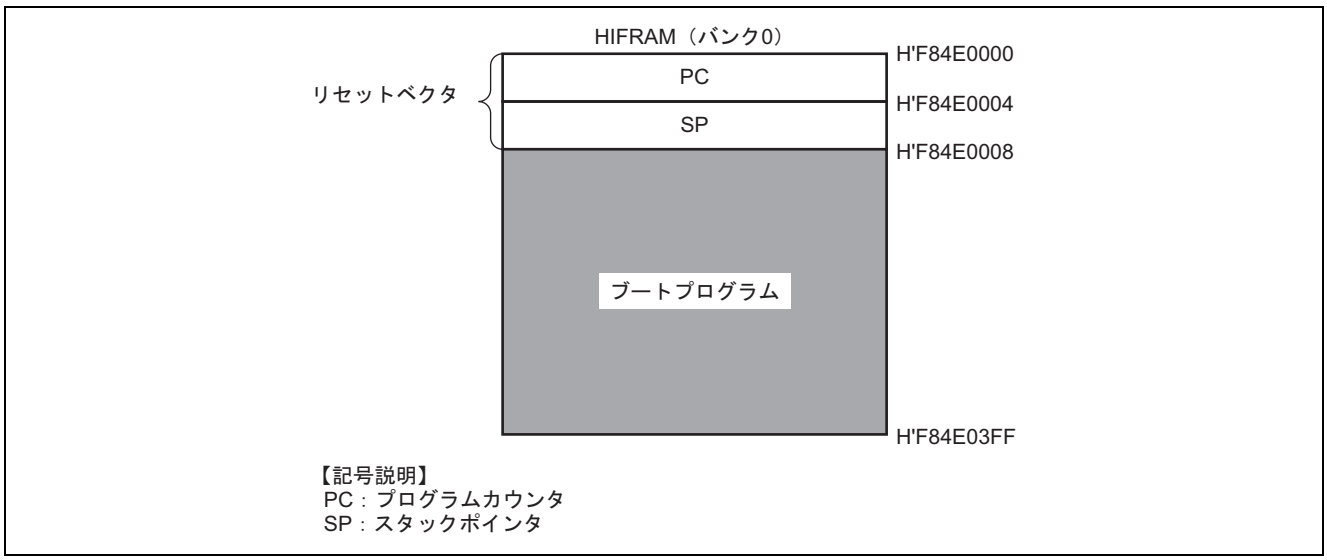


図 10　HIFRAM のメモリマップ

本タスク例では，SH7641 の CS5A エリアに SH7618 の HIF を接続しています。SH7641 が HIF のレジスタをアクセスするときのアドレスを表 4 に示します。

表 4　SH7641 による HIF アクセス時のアドレス

アドレス	アクセスレジスタ
H'B4000000	HIFIDX レジスタで指定した HIF のレジスタ
H'B4001000	ライト時: HIFIDX レジスタ リード時: HIFGSR レジスタ

2.1.3 ソフトウェア説明

1. モジュール説明

表 5 に本タスク例で使用する SH7641 のモジュールを示します。

表 5　SH7641 のモジュール説明

モジュール名	ラベル名	機能
SH7641 メインルーチン	main_7641	SH7641 の外部バスおよび端子の初期設定
HIFRAM 連続書き込みルーチン	write_HIFRAM	HIFRAM へのブートプログラム連続書き込み
SH7618 起動ルーチン	hif_boot	SH7618 を起動

2. 使用内蔵レジスタ説明

表 6 に本タスク例で使用する SH7618 の内蔵レジスタを示します。

表 6　SH7618 の使用内蔵レジスタ説明

レジスタ名			設定値	機能
	ビット	ビット名		
HIFADR				HIF アドレスレジスタ
	31 ～ 10	—	すべて 0	リザーブビット
	9 ～ 2	A9 ～ A2	—	HIFRAM アドレス指定 外部デバイスが HIFRAM をアクセスする際のアドレスを 32 ビット境界で指定します。
	1 0	—	0	リザーブビット
HIFDATA				HIF データレジスタ
	31 ～ 0	D31 ～ D0	—	32 ビットデータ 外部デバイスから HIFRAM へのアクセスに使用します。
HIFBCR				HIF ブート制御レジスタ
	31 ～ 8	—	すべて 0	リザーブビット
	7 ～ 1	—	すべて 0	AC ビット書き込み用補助 AC ビットに 1 をセットするためのビットパターン (H'A5) 書き込みに使用します。
	0	AC	—	HIFRAM アクセス排他制御 SH7618 を HIF ブートモードで起動すると 1 にセットされます。 HIFRAM へのプログラム書き込み終了後 SH7641 が 0 クリアし SH7618 が起動します。

レジスタ名	ビット	ビット名	設定値	機能
HIFIDX				HIF インデックスレジスタ
	31 ～ 8	—	すべて 0	リザーブビット
	7 ～ 2	REG5 ～ REG0	—	HIF 内蔵レジスタ選択 外部デバイスがアクセスする HIF のレジスタを選択します。
	1 0	BYTE1 BYTE0	—	HIF 内蔵レジスタ内バイト選択 外部デバイスが HIF の内蔵レジスタをアクセスするときのワード位置を指定するビットです。
HIFMCR				HIF メモリ制御レジスタ
	31 ～ 8	—	すべて 0	リザーブビット
	7	LOCK	1	ロックビット 外部デバイスが HIFRAM を連続アクセスするときに使用します。 LOCK ビットと WT ビットを同時に 1 をセットすると HIFRAM への連続書き込みが可能になります。
	6	—	0	リザーブビット
	5	WT	1	ライトビット 本ビットを 1 にセットすると HIFDATA の値が HIFADR に対応する HIFRAM の位置へ書き込まれます。
	4	—	0	リザーブビット
	3	RD	0	リードビット 本ビットを 1 にセットすると HIFADR に対応する HIFRAM のデータが HIFDATA に読み出されます。
	2 1	—	0	リザーブビット
	0	AI/AD	0	アドレスオートインクリメント/デクリメント LOCK = 1 かつ AI/AD = 0 のとき SH7641 が HIFDATA をアクセスするたびに HIFADR がインクリメント (+4) され，連続した HIFRAM のアドレスに対しての書き込みおよび読み出しが可能です。

表 7 に本タスク例で使用する SH7641 の内蔵レジスタを示します。

表 7　SH7641 の使用内蔵レジスタ説明

レジスタ名	ビット	ビット名	設定値	機能
CS5ABCR				CS5A 空間バスコントロールレジスタ
	10 9	BSZ1 BSZ0	1 0	データバス幅指定 CS5A 空間をアクセスするときのデータバス幅を指定します。 BSZ [1,0] = B'10 のときデータバス幅を 16 ビットに設定
CS5AWCR				CS5A 空間ウェイトコントロールレジスタ
	31 ～ 19	—	すべて 0	リザーブビット
	18 17 16	WW2 WW1 WW0	すべて 0	ライトアクセスウェイトサイクル数 ライトアクセス時に挿入するウェイト数を指定します。 WW [2-0] = B'000 のとき WR ビットで指定するリードアクセスウェイトと同じ数のウェイトを挿入
	15 ～ 13	—	すべて 0	リザーブビット
	12 11	SW1 SW0	0 1	アドレス，$\overline{CS}$ アサート→$\overline{RD}$，$\overline{WEn}$ アサート遅延ウェイト数 SW [1, 0] = B'01 のときアドレス，$\overline{CS}$ アサートから $\overline{RD}$，$\overline{WEn}$ アサートまで 1.5 サイクルのウェイトを挿入
	10 9 8 7	WR3 WR2 WR1 WR0	1 0 0 0	リードアクセスウェイトサイクル数 リードアクセス時に挿入するウェイトサイクル数を指定します。本タスク例では，リード/ライトアクセス共に本ビットで指定したウェイトを挿入します。 WR [3-0] = B'1000 のときリード/ライトアクセス時 10 サイクルのウェイトを挿入
	6	WM	0	外部ウェイトマスク指定 外部ウェイト入力の有効/無効を指定します。 WM=0 のとき外部ウェイト入力有効
	5 ～ 2	—	すべて 0	リザーブビット
	1 0	HW1 HW0	0 1	$\overline{RD}$，$\overline{WEn}$ ネゲート→アドレス，$\overline{CS}$ ネゲート遅延サイクル数 HW [1, 0] =B'01 のとき $\overline{RD}$，$\overline{WEn}$ ネゲート→アドレス，$\overline{CS}$ ネゲートまで 1.5 サイクルのウェイトを挿入
PCCR				ポート C コントロールレジスタ
	3 2	PC1MD2 PC1MD1	1	PC1 モード 2，1 PC1MD [2, 1] = B'11 のとき PTC1 端子を CS5A 機能に設定

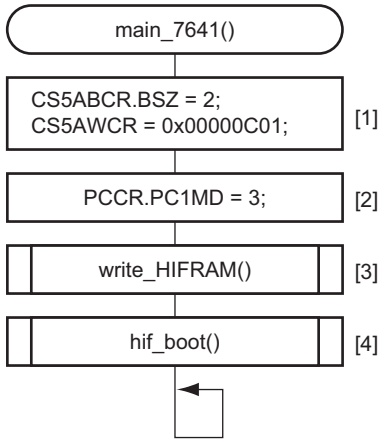
3. 変数説明
表 8 に本タスク例で使用する変数を示します。

表 8　変数説明

変数	内容	データ長	初期値	使用モジュール名
*trans_src_addr	転送元 (ブートプログラム格納) アドレスを示すポインタ	2 バイト	—	HIFRAM 連続書き込みルーチン
hif_addr	書き込みを開始する HIF のアドレス	2 バイト	H'0000	HIFRAM 連続書き込みルーチン
t_size	転送するプログラムサイズ	2 バイト	H'300	HIFRAM 連続書き込みルーチン

2.1.4　フローチャート

1. SH7641 メインルーチン



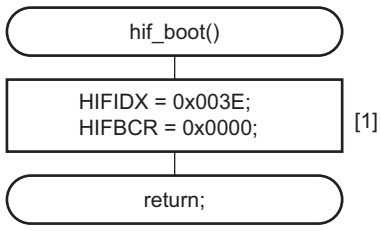
[1] <CS5A空間のバス設定>
・データバス幅16ビット
・アドレス、$\overline{CS5A}$アサートから$\overline{RD}$、$\overline{WEn}$アサートまで1.5サイクルのウェイトを挿入
・$\overline{RD}$、$\overline{WEn}$ネゲートからアドレス、$\overline{CS5A}$ネゲートまで1.5サイクルのウェイトを挿入
・リード／ライトアクセス時10サイクルのウェイトを挿入

[2] 端子機能を$\overline{CS5A}$に設定

[3] HIFRAMへの連続書き込みルーチンの呼び出し

[4] SH7618起動ルーチンの呼び出し

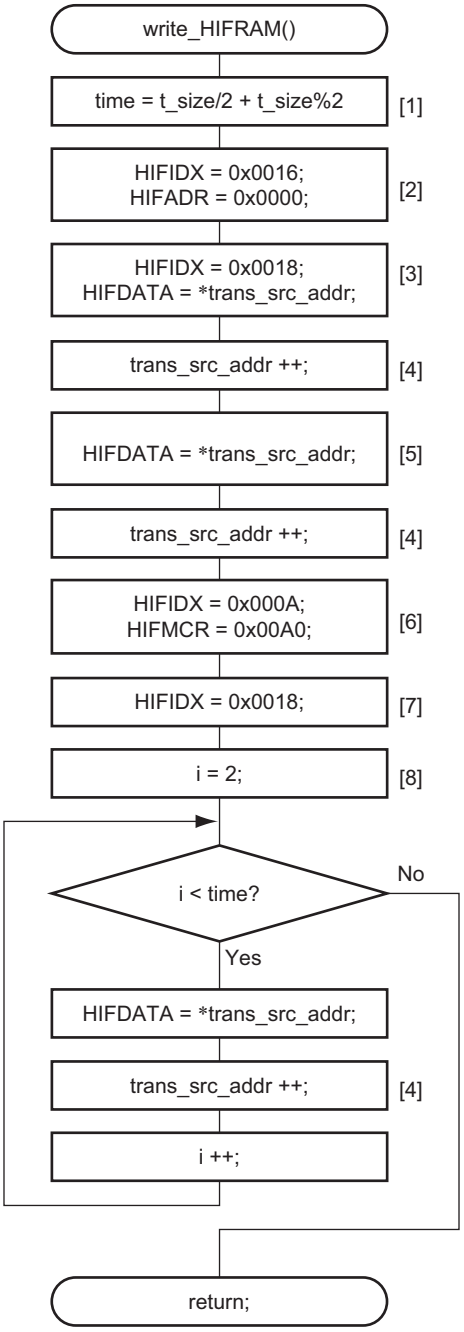
2. SH7618 起動ルーチン



[1] HIFIDXレジスタでHIFBCRの下位16ビットを指定し、HIFBCRレジスタのACビットを0クリアする
（SH7618はHIFRAMの先頭アドレスにライトされたデータをリセットベクタとして読み込み起動する）

3. HIFRAM 連続書き込みルーチン

<引数>
*trans_src_addr ：転送元アドレスのポインタ
hif_addr ：HIFRAMへのライト開始アドレス
t_size ：転送プログラムサイズ



[1] HIFRAMへのライト回数を設定
time：転送回数

[2] HIFIDXレジスタでHIFADRレジスタの下位16ビットを指定し、HIFADRレジスタにHIFRAMの先頭アドレス（0x0000）を設定

[3] HIFIDXレジスタでHIFDATAレジスタの上位16ビットを指定し、HIFDATAレジスタ[31:16]にライトデータを設定

[4] 転送元アドレスをインクリメント（+2）

[5] HIFDATAレジスタ[15:0]にライトデータを設定

[6] HIFIDXレジスタでHIFMCRレジスタの下位16ビットを指定し、HIFRAMへの連続ライトモードに設定（HIFADRレジスタはHIFDATAレジスタをアクセスするたびにオートインクリメント）

[7] HIFIDXレジスタでHIFDATAレジスタ[31:16]を指定

[8] HIFRAMへブートプログラムのライトが終了するまでHIFDATAレジスタへ連続ライト（先頭の4バイトはライト済みのためi = 2からループ開始）

2.1.5 プログラムリスト

```
/*****************************************************************************************************************/
// SH7618 HIF boot mode application note
//  A boot program is written to HIFRAM and SH7618 is booted
//      CPU:      SH7641,SH-3 DSP,Big Endian
//      Clock:    External input = 12.5MHz
//                CPU clock = 100MHz
//                External BUS clock = 50MHz
//                Peripheral clock = 25MHz
//      Written:  '04/4 Rev.2.0
/*****************************************************************************************************************/

#include "7641.h"

/*****************************************************************************************************************/
/*  Protocol declaration of the function                                                                         */
/*****************************************************************************************************************/
/*------------ Symbol Definition ---------------------------------------------------------------------------------*/
#define BOOT_STRAGE_ADDR  0xA5600000                          // storing address of a boot program
#define HIFRAM_START      0x0000                              // write address of HIFRAM
#define BOOT_P_SIZE       0x300                               // boot program size(Byte)

    //----------The value when specifying the register of HIF
#define SEL_HIFMCR_LO     0x000A                              // HIFMCR[15:0]
#define SEL_HIFBCR_LO     0x003E                              // HIFBCR[15:0]
#define SEL_HIFADR_LO     0x0016                              // HIFADR[15:0]
#define SEL_HIFDATA_UP    0x0018                              // HIFDATA[31:16]

/*------------ Function Definition -------------------------------------------------------------------------------*/
void main(void);

void write_HIFRAM(unsigned short*, unsigned short , unsigned short);
void hif_boot(void);

/*-----------------------------------------------------------------------------------------------------------------*/
    //----------The address when accessing the register of HIF
// select of the register of HIF
#define HIF_REG_SEL       (*(volatile unsigned short*)0xB4001000)
// data write to the register of HIF
#define HIF_DATA_WR       (*(volatile unsigned short*)0xB4000000)

/*****************************************************************************************************************/
/*       Main routine                                                                                            */
/*****************************************************************************************************************/
void main(void)
{
    //----------set of bus interface
    BSC.CS5ABCR.BIT.BSZ = 2;
    BSC.CS5AWCR = 0x00000C01;

    //----------set as a  CS5A function
    PFC.PCCR.BIT.PC1MD = 3;                                   /* bit[2-3]-PC1MD=b'11 : PC1=>CS5A          */

    //----------boot program is written to HIFRAM
    write_HIFRAM((unsigned short*)BOOT_STRAGE_ADDR , HIFRAM_START , BOOT_P_SIZE);

    //----------SH7618 is booted
    hif_boot();                                               /* HIF boot                                 */

    //----------Loop
    while(1);                                                 /* Loop                                     */
}
```

```
/*****************************************************************************************************************/
//  function:    write_HIFRAM
//  operation:   boot program writing to HIFRAM
//  argument:    trans_src_addr ; storing address of a boot program
//               hif_addr     ; head address of HIFRAM which transmits a boot program
//               t_size       ; transmit program size
//  return:      non-return
/*****************************************************************************************************************/
void write_HIFRAM(unsigned short *trans_src_addr , unsigned short hif_addr , unsigned short t_size)
{
    volatile unsigned short time , i ;

    time = t_size/2 + t_size%2;                                 /* calculate times of transmission          */

    HIF_REG_SEL = SEL_HIFADR_LO;                                /* select HIFADR register                   */
    HIF_DATA_WR = hif_addr;                                     /* set of HIFRAM address                    */

    HIF_REG_SEL = SEL_HIFDATA_UP;                               /* select HIFDATA register[31:16]           */
    HIF_DATA_WR = (*trans_src_addr);                            /* set to a HIFDATA register[31:16]         */

    trans_src_addr ++;                                          /* storing address is increment(+2)         */

    HIF_DATA_WR = (*trans_src_addr);                            /* set to a HIFDATA register[15:0]          */

    trans_src_addr ++;                                          /* storing address is increment(+2)         */

    HIF_REG_SEL = SEL_HIFMCR_LO;                                /* select HIFMCR register[15:0]             */
    HIF_DATA_WR = 0x00A0;                                       /* set as continuation write mode           */

    HIF_REG_SEL = SEL_HIFDATA_UP;                               /* select HIFDATA register[31:16]           */

    for( i=2 ; i<time ; i++){
        HIF_DATA_WR = (*trans_src_addr);                        /* write to HIFDATA register(HIFRAM)        */

        trans_src_addr ++;                                      /* storing address is increment(+2)         */
    }
}

/*****************************************************************************************************************/
//  function:    hif_boot
//  operation:   SH7618 is booted
//  argument:    non-argument
//  return:      non-return
/*****************************************************************************************************************/
void hif_boot(void)
{
    HIF_REG_SEL = SEL_HIFBCR_LO;                                /* select HIFBCR register[15:0]             */
    HIF_DATA_WR = 0x0000;                                       /* clear AC bit of HIFBCR register          */
}
```

2.2 HIF ブートモードによる内蔵 RAM へのプログラム転送

2.2.1 概要

本タスク例では，SH7618 の HIF ブートモードによる起動後の，HIFRAM を介した内蔵 RAM へのプログラム転送と実行について述べています。

SH7618 のブートプログラムを外部デバイスのプログラムと同じ ROM に格納しておくことで，SH7618 の起動用 ROM が削減できます。

2.2.2 仕様

1. HIFRAM 上でプログラムを実行しつつ，HIFRAM の空き領域を使用してプログラムを転送します。
2. SH7641 を HIF に接続して，SH7618 を HIF ブートモードで起動します。
3. HIFRAM の未使用領域をバッファとして使用し, SH7641 が HIFRAM バッファ領域に転送したプログラムを SH7618 が内蔵 RAM へ転送します。
4. プログラムは複数回に分けて SH7618 の内蔵 RAM へ転送します。
5. プログラムの転送の際に, SH7641 から HIFRAM への転送と HIFRAM から内蔵 RAM への転送が競合しないように, HIF 汎用ステータスレジスタ (HIFGSR) を使用して SH7641 と SH7618 の間で同期をとります。
6. SH7641 は，転送量を示すヘッダを付けて HIFRAM バッファにプログラムを書き込みます。
7. SH7618 は，ヘッダで指定された量を HIFRAM バッファから内蔵 RAM へ転送します。
8. プログラムの転送終了後，SH7618 は内蔵 RAM に転送したプログラムを実行します。
9. 本タスク例において，SH7641 の動作周波数は，CPU クロック 100MHz，外部バスクロック 50MHz，周辺クロック 25MHz とします。また，SH7618 は起動するまで各レジスタが初期値であるため，起動後に SH7618 の動作周波数を設定しています。起動後の SH7618 の動作周波数は，内部クロック 100MHz，外部バスクロック 50MHz，周辺クロック 50MHz としています。

図 11 に SH7641 と SH7618 の接続例を示します。

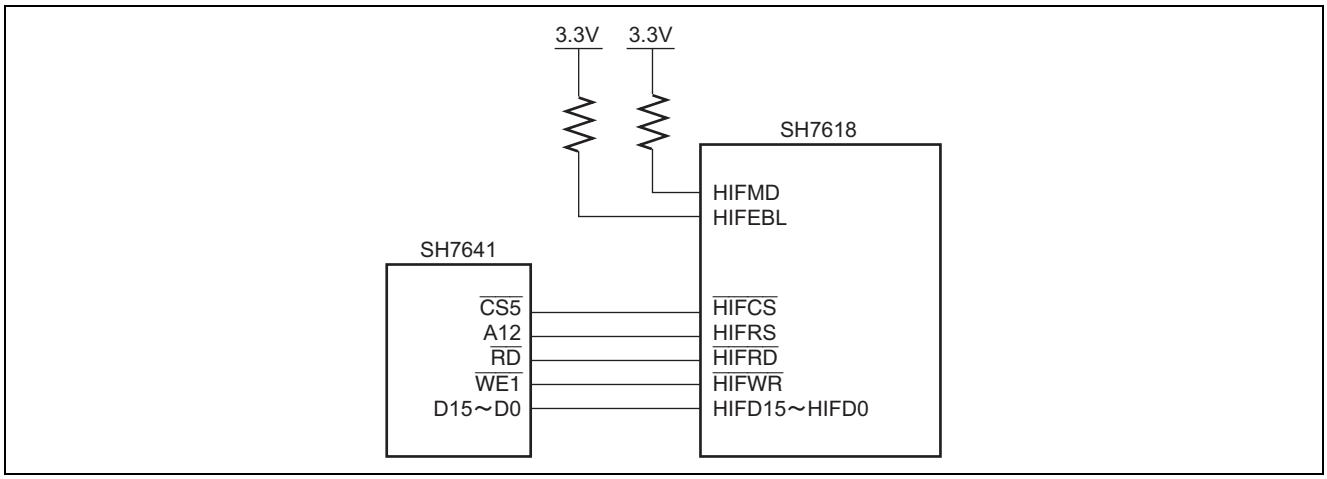


図 11 接続例

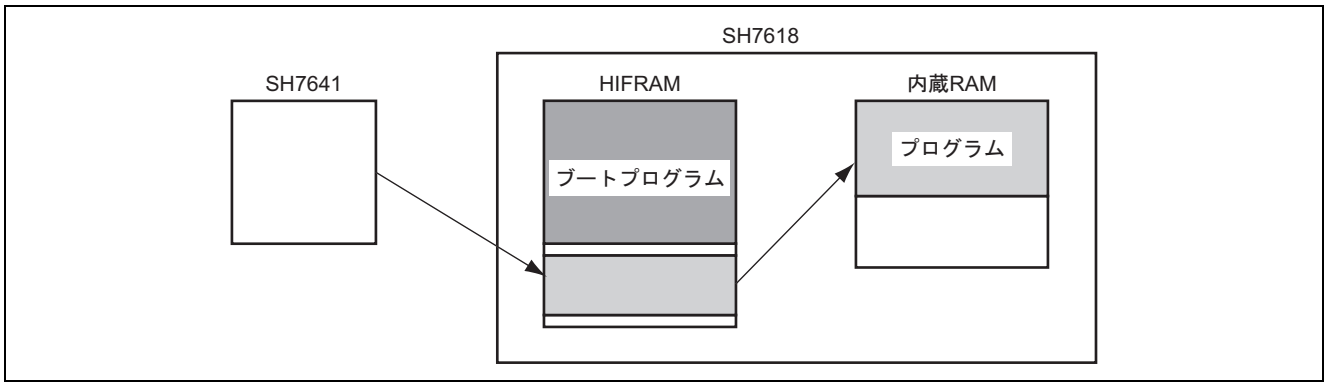


図 12　内蔵 RAM へのプログラム転送

内蔵 RAM へのプログラム転送時の HIFRAM のメモリマップを図 13 に示します。

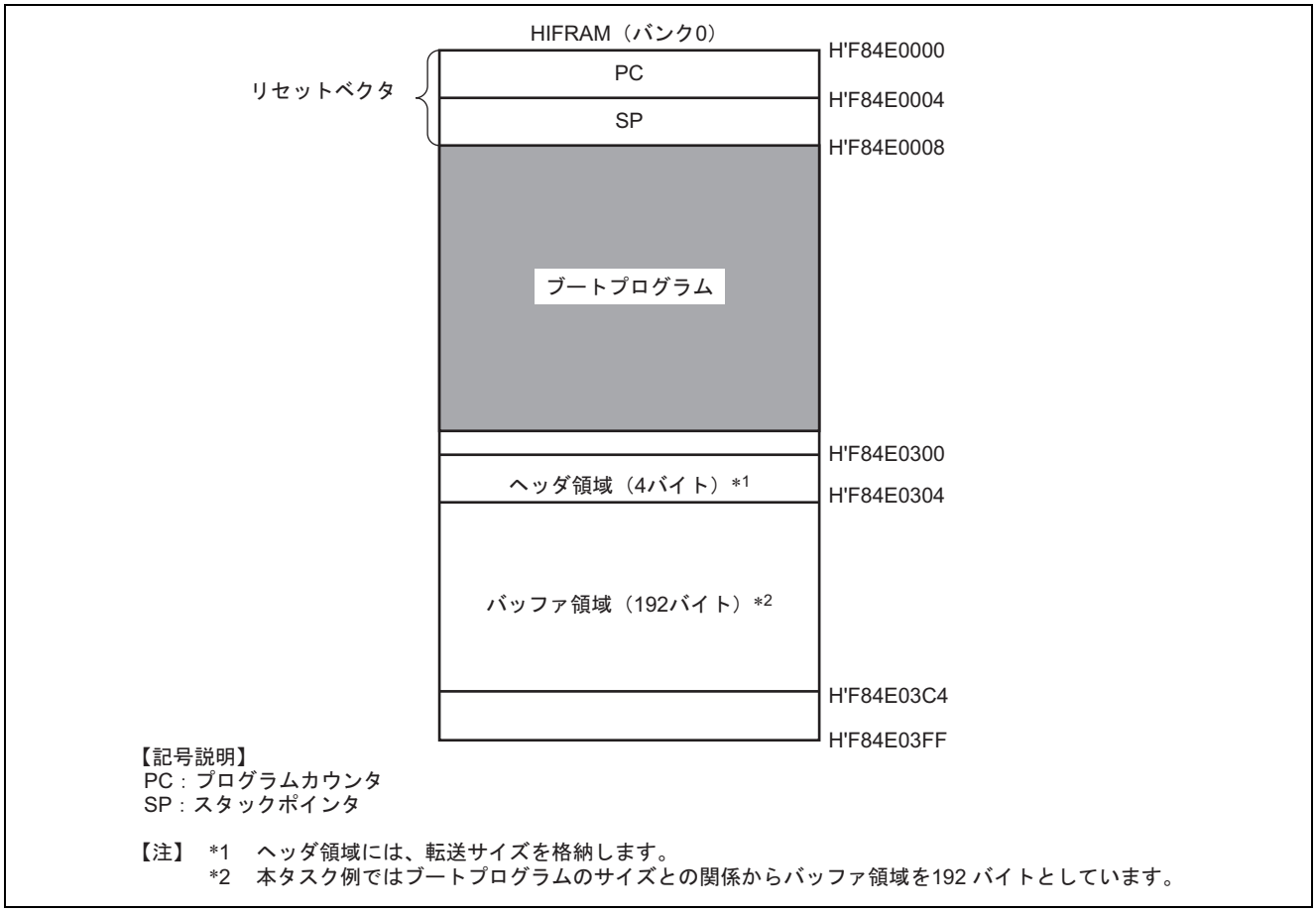


図 13　HIFRAM のメモリマップ

HIFGSR レジスタは，ユーザが自由に定義して使用可能で，本タスク例では SH7641 と SH7618 の間で同期をとるために，ビット 1 を TEND に，ビット 0 を HIFS に定義して使用しています。表 9 に本タスク例での HIFGSR レジスタの内容を示します。

表 9　HIFGSR レジスタの内容

ビット	ビット名	初期値	説明
31〜16	—	すべて 0	リザーブビット
15 〜 2	STATUS15 〜 STATUS2	すべて 0	汎用ステータスビット 本タスク例では使用していません。
1	TEND	—	トランスミットエンド プログラムの転送が終了したかどうかを示します。 SH7618 は本ビットを参照してプログラム転送が終了したことを確認し，内蔵 RAM へ転送したプログラムを実行します。 SH7641 は全プログラムの転送が終了すると，本ビットを 1 にセットします。 1: 全プログラムの転送が終了 SH7618 は内蔵 RAM に転送したプログラムを実行 0: プログラム転送中 内蔵 RAM へ転送するプログラムが SH7641 側に残存している
0	HIFS	—	HIFRAM ステータス HIFRAM の状態を示します。 SH7618 と SH7641 は本ビットを参照して転送を行います。 SH7618 は，SH7641 から HIFRAM への転送が可能になったときに，1 のセットのみ行います。 SH7641 は，HIFRAM バッファへのプログラム転送が終了したときに，0 クリアのみ行います。 1: SH7641 から HIFRAM への書き込みが可能な状態 SH7618 は待機状態 0: SH7641 から HIFRAM への書き込みが不可の状態 SH7618 は HIFRAM バッファ上のプログラムを内蔵 RAM へ転送

2.2.3 動作説明

図 14 に HIFGSR の値と，そのときの動作状態を示し，表 10 に SH7618 および SH7641 の動作内容を説明します。また，図 15 に HIFGSR レジスタの HIFS ビットの値とそのときの動作状態を示します。

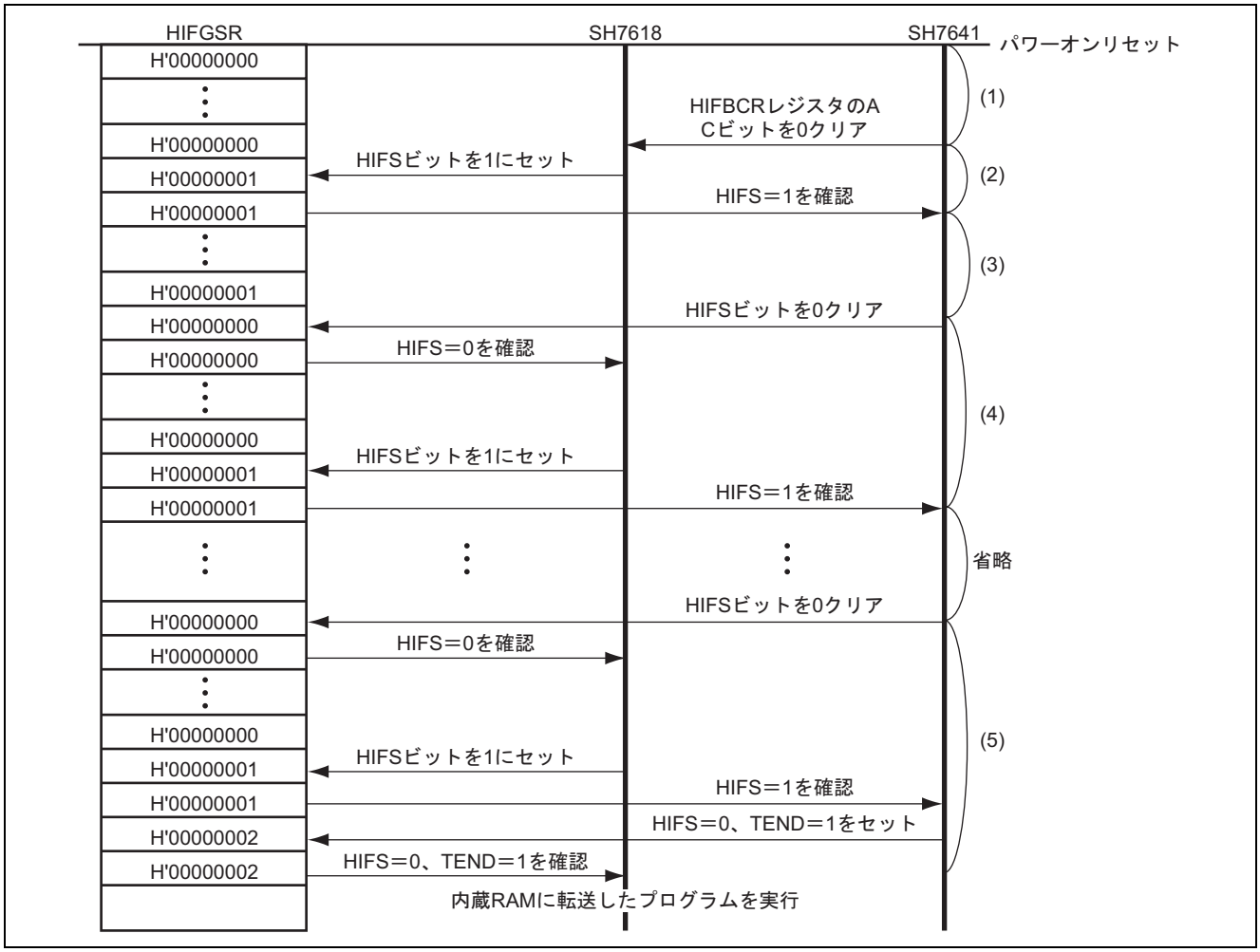
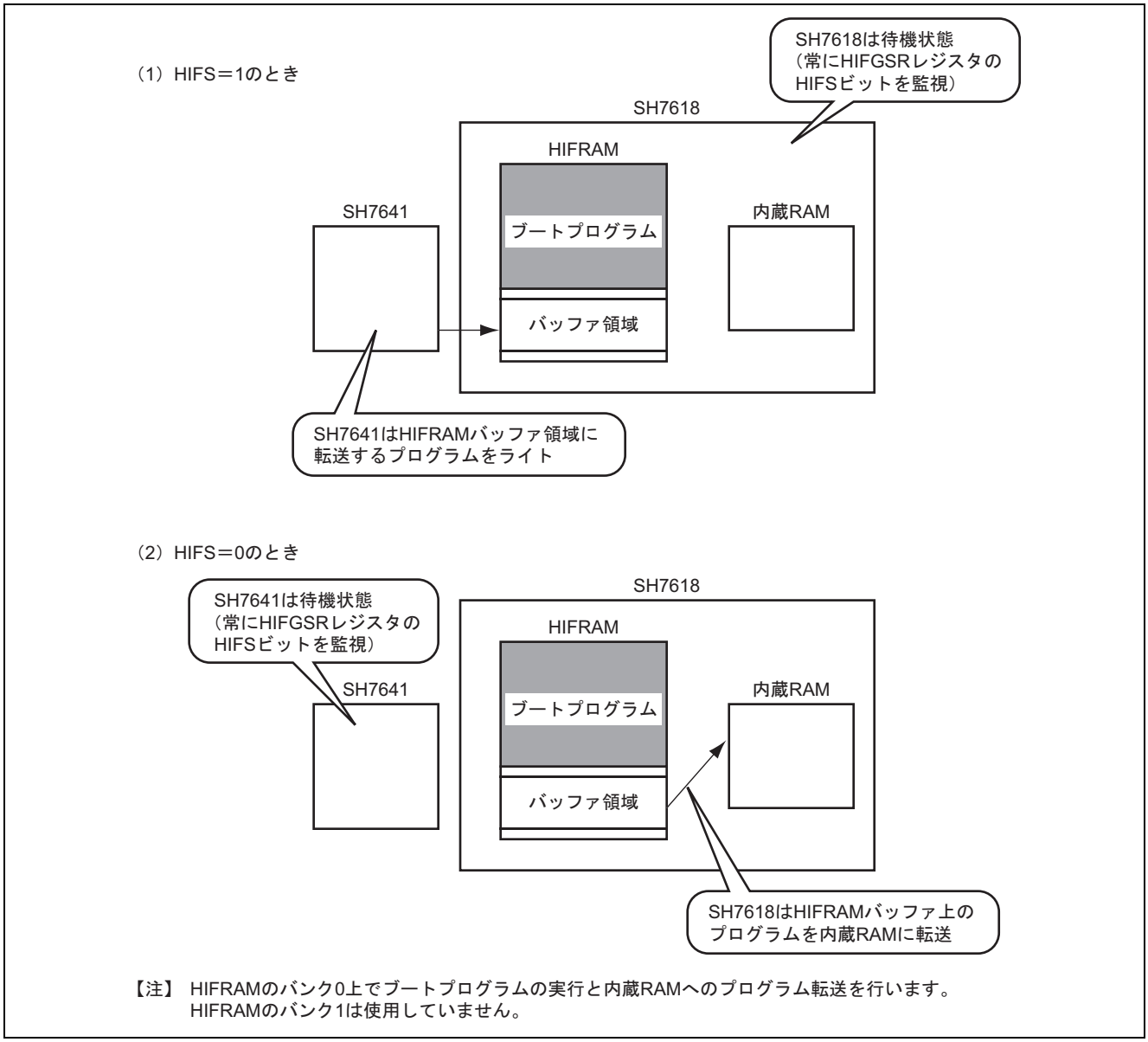


図 14　HIFGSR の値と動作状態

表 10　SH7618 および SH7641 の動作説明

	SH7618 の動作内容	SH7641 の動作内容
(1)	• HIFMD 端子に H レベルを入力した状態でパワーオンリセットを解除し HIF ブートモードにセット* • HIFBCR レジスタの AC ビットを 0 にクリアされるまでウェイト*	• リセット解除後 HIFRAM へブートプログラムを書き込む • ブートプログラムの書き込み終了後 HIFBCR レジスタの AC ビットを 0 クリア
(2)	• HIFBCR レジスタの AC ビットが 0 にクリアされ HIFRAM に書き込まれたプログラムを実行* • SH7618 の初期設定および HIFRAM の初期化 • HIFGSR レジスタの HIFS ビットを 1 にセット	• HIFGSR レジスタの HIFS ビットが 0 の間ウェイト
(3)	• HIFGSR レジスタの HIFS ビットが 1 の間ウェイト	• HIFGSR レジスタの HIFS ビットが 1 にセットされたことを確認 • HIFRAM バッファに転送するプログラムを書き込む • プログラム書き込み終了後 HIFGSR レジスタの HIFS ビットを 0 クリア
(4)	• HIFGSR レジスタの HIFS ビットが 0 クリアされたことを確認 • SH7641 が書き込んだ HIFRAM バッファ上のプログラムを内蔵 RAM に転送 • プログラムの転送終了後 HIFGSR レジスタの HIFS ビットを 1 にセット	• HIFGSR レジスタの HIFS ビットが 0 の間ウェイト
(5)	• HIFGSR レジスタの HIFS ビットが 0 クリアされたことを確認 • SH7641 が書き込んだ HIFRAM バッファ上のプログラムを内蔵 RAM に転送 • プログラムの転送終了後 HIFGSR レジスタの HIFS ビットを 1 にセット • HIFGSR レジスタの HIFS = 0, TEND = 1 を確認し内蔵 RAM へ転送したプログラムへ処理を移行	• 転送する全プログラムの書き込みが終了し HIFGSR レジスタの HIFS ビットを 0 にクリア • HIFGSR レジスタの HIFS ビットが 0 の間ウェイト • HIFGSR レジスタの HIFS ビットが 1 にセットされたことを確認 • HIFGSR レジスタに HIFS = 0, TEND = 1 を書き込む

【注】　*　ハードウェアによる処理で，それ以外はソフトウェアによる処理になります。



【注】 HIFRAMのバンク0上でブートプログラムの実行と内蔵RAMへのプログラム転送を行います。
HIFRAMのバンク1は使用していません。

図 15　HIFS ビットの値と動作状態

2.2.4 ソフトウェア説明

1. モジュール説明

表 11 に本タスク例で使用する SH7618 のモジュール, 表 12 に本タスク例で使用する SH7641 のモジュールを示します。

表 11 SH7618 のモジュール説明

モジュール名	ラベル名	機能
HIFRAM メインルーチン	hifram_main	SH7618 の初期設定と HIFRAM バッファに書き込まれたプログラムを内蔵 RAM に転送
転送プログラム実行ルーチン	jump_uram	内蔵 RAM へプログラムカウンタを移動し転送したプログラムを実行 【注】本モジュールはアセンブリ言語で記述しています

表 12 SH7641 のモジュール説明

モジュール名	ラベル名	機能
SH7641 メインルーチン	main_7641	外部バスおよび端子の設定と各モジュールの呼び出し
HIFRAM 連続書き込みルーチン	write_HIFRAM	HIFRAM へプログラムを書き込む
SH7618 起動ルーチン	hif_boot	SH7618 を HIF ブートモードで起動
HIFRAM バッファ書き込みルーチン	sync_7618	SH7618 と同期をとり HIFRAM バッファにプログラムを書き込む

2. 使用レジスタ説明

表 13 に本タスク例で使用する SH7618 の内蔵レジスタを示します。

表 13　SH7618 の使用内蔵レジスタ説明

レジスタ名			設定値	機能
	ビット	ビット名		
HIFADR				HIF アドレスレジスタ
	31 ～ 10	—	すべて 0	リザーブビット
	9 ～ 2	A9 ～ A2	—	HIFRAM アドレス指定 外部デバイスが HIFRAM をアクセスする際のアドレスを 32 ビット境界で指定します。
	1 0	—	0	リザーブビット
HIFDATA				HIF データレジスタ
	31 ～ 0	D31 ～ D0	—	32 ビットデータ 外部デバイスから HIFRAM へのアクセスに使用します。
HIFBCR				HIF ブート制御レジスタ
	31 ～ 8	—	すべて 0	リザーブビット
	7 ～ 1	—	すべて 0	AC ビット書き込み用補助 AC ビットに 1 をセットするためのビットパターン (H'A5) 書き込みに使用します。
	0	AC	—	HIFRAM アクセス排他制御 SH7618 を HIF ブートモードで起動すると 1 にセットされます。 HIFRAM へのプログラム書き込み終了後 SH7641 が 0 クリアし SH7618 が起動します。

レジスタ名		設定値	機能
ビット	ビット名		
HIFIDX			HIF インデックスレジスタ
31 〜 8	—	すべて 0	リザーブビット
7 〜 2	REG5 〜 REG0	—	HIF レジスタ選択 外部デバイスがアクセスする HIF のレジスタを選択します。
1 0	BYTE1 BYTE0	—	HIF レジスタ内バイト選択 外部デバイスが HIF のレジスタをアクセスするときのワード位置を指定するビットです。
HIFMCR			HIF メモリ制御レジスタ
31 〜 8	—	すべて 0	リザーブビット
7	LOCK	—	ロックビット 外部デバイスが HIFRAM を連続アクセスするときに使用します。
6	—	0	リザーブビット
5	WT	—	ライトビット 本ビットを 1 にセットすると HIFDATA の値が HIFADR に対応する HIFRAM の位置へ書き込まれます。
4	—	0	リザーブビット
3	RD	—	リードビット 本ビットを 1 にセットすると HIFADR に対応する HIFRAM のデータが HIFDATA に読み出されます。
2 1	—	0	リザーブビット
0	AI/AD	0	アドレスオートインクリメント/デクリメント LOCK = 1 かつ AI/AD = 0 のとき SH7641 が HIFDATA をアクセスするたびに HIFADR がインクリメント (+4) され，連続した HIFRAM のアドレスに対しての書き込みおよび読み出しが可能です。
HIFGSR			HIF 汎用ステータスレジスタ
31 〜 16	—	すべて 0	リザーブビット
15 〜 2	STATUS15 〜 STATUS2	すべて 0	汎用ステータスビット 本タスク例では使用していません。
1	TEND	—	トランスミットエンド プログラム転送の状況を示します。 TEND = 1 のとき SH7618 は内蔵 RAM へ転送したプログラムを実行します。 TEND = 0 のとき SH7618 は内蔵 RAM へプログラムを転送します。
0	HIFS	—	HIFRAM ステータス HIFRAM の状態を示します。 HIFS = 0 のとき SH7641 から HIFRAM への転送が可能です。 HIFS = 1 のとき SH7641 から HIFRAM への転送は禁止です。

表 14 に本タスク例で使用する SH7641 の内蔵レジスタを示します。

表 14　SH7641 の使用内蔵レジスタ説明

レジスタ名	ビット	ビット名	設定値	機能
CS5ABCR				CS5A 空間バスコントロールレジスタ
	10 9	BSZ1 BSZ0	1 0	データバス幅指定 CS5A 空間をアクセスするときのデータバス幅を指定します。 BSZ [1, 0] = B'10 のときデータバス幅を 16 ビットに設定
CS5AWCR				CS5A 空間ウェイトコントロールレジスタ
	31 ～ 19	—	すべて 0	リザーブビット
	18 17 16	WW2 WW1 WW0	0	ライトアクセスウェイトサイクル数 ライトアクセス時に挿入するウェイト数を指定します。 WW [2-0] = B'000 のとき WR ビットで指定するリードアクセスウェイトと同じ数のウェイトを挿入
	15 ～ 13	—	すべて 0	リザーブビット
	12 11	SW1 SW0	0 1	アドレス，$\overline{CS}$ アサート→$\overline{RD}$，$\overline{WEn}$ アサート遅延ウェイト数 SW [1, 0] = B'01 のときアドレス，$\overline{CS}$ アサートから $\overline{RD}$，$\overline{WEn}$ アサートまで 1.5 サイクルのウェイトを挿入
	10 9 8 7	WR3 WR2 WR1 WR0	—	リードアクセスウェイトサイクル数 リードアクセス時に挿入するウェイト数を指定します。 本タスク例では，SH7618 の HIF ブートモードでの起動前と起動後で挿入するウェイト数を変更しています。 起動前: WR [3-0] = B'1000 (10 サイクルのウェイトを挿入) 起動後: WR [3-0] = B'0010 (2 サイクルのウェイトを挿入)
	6	WM	0	外部ウェイトマスク指定 外部ウェイト入力の有効/無効を指定します。 WM = 0 のとき外部ウェイト入力有効
	5 ～ 2	—	すべて 0	リザーブビット
	1 0	HW1 HW0	0 1	$\overline{RD}$，$\overline{WEn}$ ネゲート→アドレス，$\overline{CS}$ ネゲート遅延サイクル数 HW [1, 0] = B'01 のとき $\overline{RD}$，$\overline{WEn}$ ネゲート→アドレス，$\overline{CS}$ ネゲートまで 1.5 サイクルのウェイトを挿入
PCCR				ポート C コントロールレジスタ
	3 2	PC1MD2 PC1MD1	1	PC1 モード 2，1 PC1MD [2, 1] = B'11 のとき PTC1 端子を CS5A 機能に設定

3. 使用変数説明

表 15 に本タスク例において SH7618 のプログラム内で使用する変数を, 表 16 に本タスク例において SH7641 のプログラム内で使用する変数を示します。

表 15　SH7618 プログラム内の変数説明

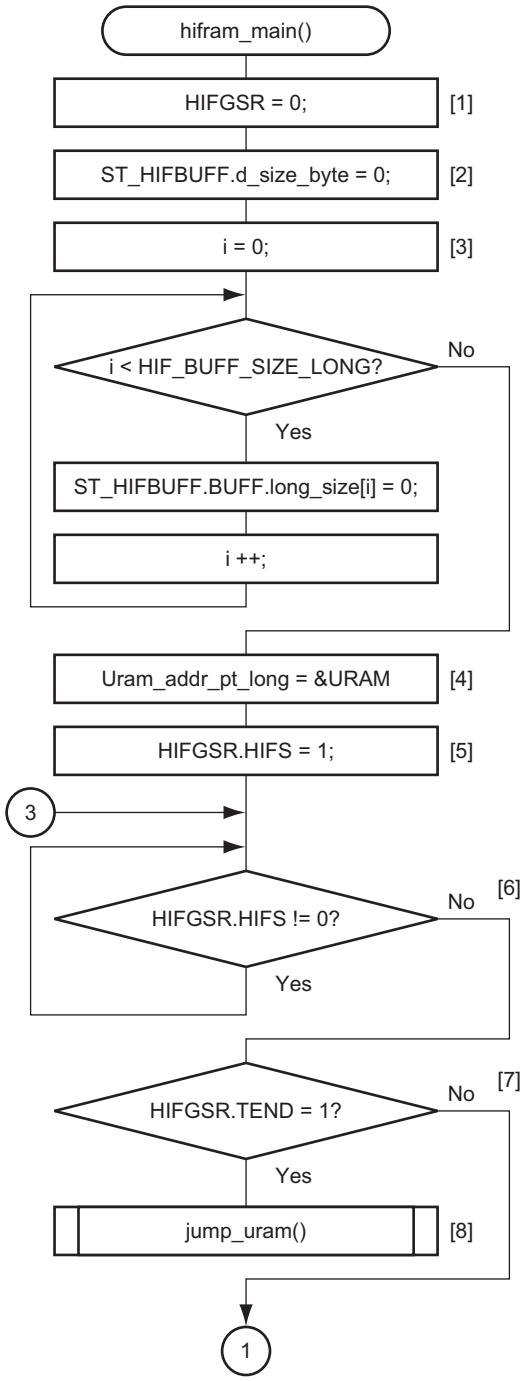
変数	内容	データ長	初期値	使用モジュール名
t_count	ロングワードサイズでの転送における転送回数	4 バイト	—	HIFRAM メインルーチン
*Uram_address_pt_byte	プログラムの転送先 (内蔵 RAM) アドレスを示すポインタ	1 バイト	—	HIFRAM メインルーチン
*Uram_address_pt_long	プログラムの転送先 (内蔵 RAM) アドレスを示すポインタ	4 バイト	—	HIFRAM メインルーチン

表 16　SH7641 プログラム内の変数説明

変数	内容	データ長	初期値	使用モジュール名
*trans_src_addr	転送元 (ブートプログラム格納) アドレスを示すポインタ	2 バイト	—	HIFRAM 連続書き込みルーチン
hif_addr	書き込みを開始する HIF のアドレス	2 バイト	H'0000	HIFRAM 連続書き込みルーチン
t_size	転送するプログラムサイズ	2 バイト	H'300	HIFRAM 連続書き込みルーチン
*s_addr	転送元 (ブートプログラム格納) アドレスを示すポインタ	2 バイト	—	HIFRAM バッファ書き込みルーチン
h_addr	HIFRAM ヘッダアドレス	2 バイト	H'0300	HIFRAM バッファ書き込みルーチン
b_addr	HIFRAM バッファアドレス	2 バイト	H'0304	HIFRAM バッファ書き込みルーチン
t_size	全転送プログラムサイズ	4 バイト	H'0300	HIFRAM バッファ書き込みルーチン
b_size	HIFRAM バッファサイズ (1 回の転送サイズ)	2 バイト	H'C0	HIFRAM バッファ書き込みルーチン

2.2.5 フローチャート

1. HIFRAM メインルーチン



[1] HIFGSRレジスタを初期化

[2] HIFRAMヘッダを初期化

[3] HIFRAMバッファを初期化

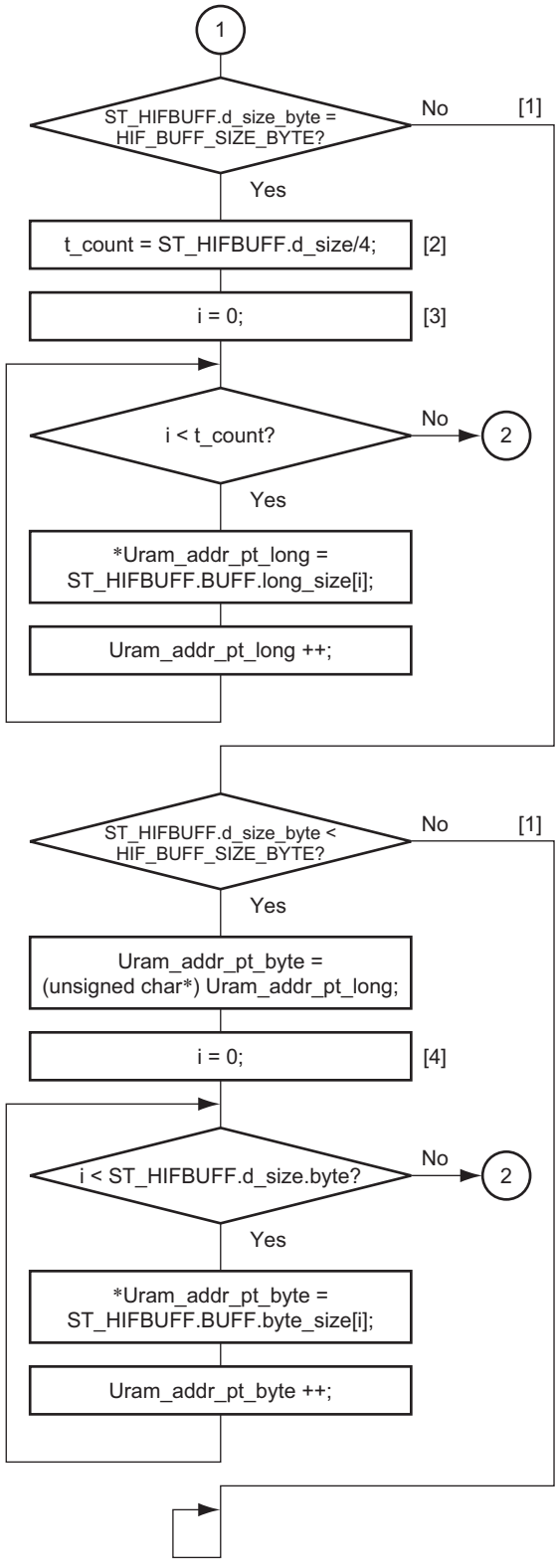
[4] プログラムの転送先アドレスをセット

[5] HIFGSRレジスタのHIFSビットを1にセットし
外部デバイスからHIFRAMへの書き込みを許可

[6] 外部デバイスからHIFRAMへの書き込みが
終了するまでウェイト

[7] 全プログラムの転送が終了したかを判定
TEND＝1のとき転送終了

[8] 転送プログラム実行ルーチンの呼び出し

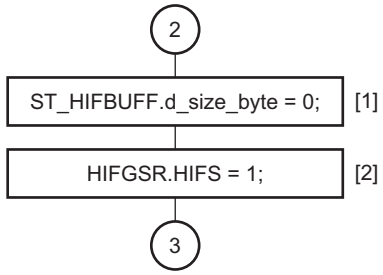


[1] ロングワードサイズで転送するか
バイトサイズで転送するかを判定
・転送サイズ＝バッファサイズのときロングワード
・転送サイズ＜バッファサイズのときバイト
・上記以外のときエラー（無限ループ）

[2] ロングワードサイズでの転送回数をセット

[3] HIFRAMバッファから内蔵RAMへ
ロングワードサイズで転送

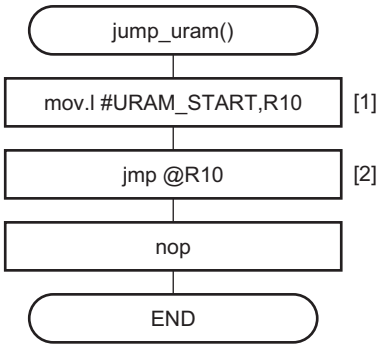
[4] HIFRAMバッファから内蔵RAMへ
バイトサイズで転送



[1] HIFRAMヘッダを初期化

[2] HIFGSRレジスタのHIFSビットを1にセットし
外部デバイスからHIFRAMへの書き込みを許可

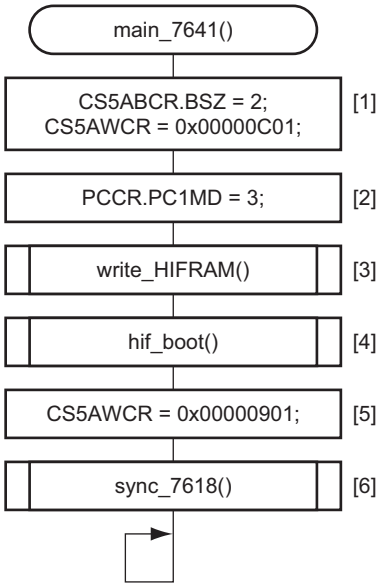
2. 転送プログラム実行ルーチン



[1] 転送したプログラムの実行開始アドレスをセット
URAM_START：内蔵RAMの先頭アドレス

[2] セットしたアドレスに分岐

3. SH7641 メインルーチン



[1] <CS5A空間のバス設定>
・データバス幅16ビット
・アドレス、$\overline{\text{CS5A}}$アサートから$\overline{\text{RD}}$、$\overline{\text{WEn}}$アサートまで
1.5サイクルのウェイトを挿入
・$\overline{\text{RD}}$、$\overline{\text{WEn}}$ネゲートからアドレス、$\overline{\text{CS5A}}$ネゲートまで
1.5サイクルのウェイトを挿入
・リード／ライトアクセス時10サイクルのウェイトを挿入

[2] 端子機能を$\overline{\text{CS5A}}$に設定

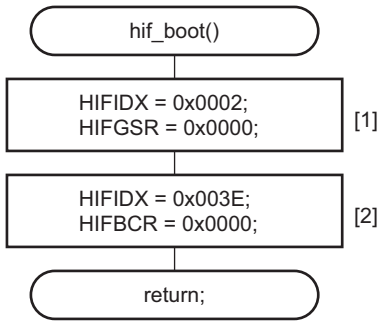
[3] HIFRAMへの連続書き込みルーチンの呼び出し

[4] SH7618起動ルーチンの呼び出し

[5] CS5A空間リード／ライトアクセス時の挿入ウェイトを
2に変更（SH7618が起動し動作周波数をあげているため
より高速なアクセスが可能となるため）

[6] HIFRAMバッファ書き込みルーチンの呼び出し

4. SH7618 起動ルーチン



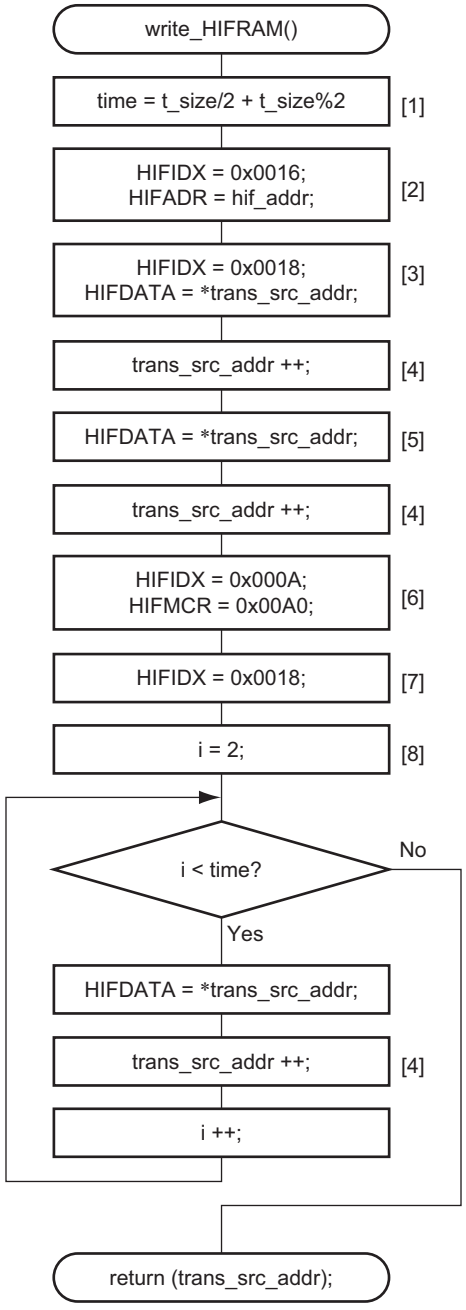
[1] HIFIDXレジスタでHIFGSRの下位16ビットを指定し、
HIFGSRレジスタを初期化（H'0000をライト）

[2] HIFIDXレジスタでHIFBCRの下位16ビットを指定し、
HIFBCRレジスタのACビットを0クリアする
（SH7618はHIFRAMの先頭アドレスにライトされたデータを
リセットベクタとして読み込み起動する）

5. HIFRAM への連続書き込みルーチン

<引数>
*trans_src_addr ：転送元アドレスのポインタ
hif_addr ：HIFRAMへのライト開始アドレス
t_size ：転送プログラムサイズ

<戻り値>
trans_src_addr ：転送元アドレス



[1] HIFRAMへのライト回数を設定
time：転送回数

[2] HIFIDXレジスタでHIFADRレジスタの下位16ビットを指定し、HIFADRレジスタにHIFRAMの書き込み開始アドレスをセット

[3] HIFIDXレジスタでHIFDATAレジスタの上位16ビットを指定し、HIFDATAレジスタ[31:16]にライトデータを設定

[4] 転送元アドレスをインクリメント（+2）

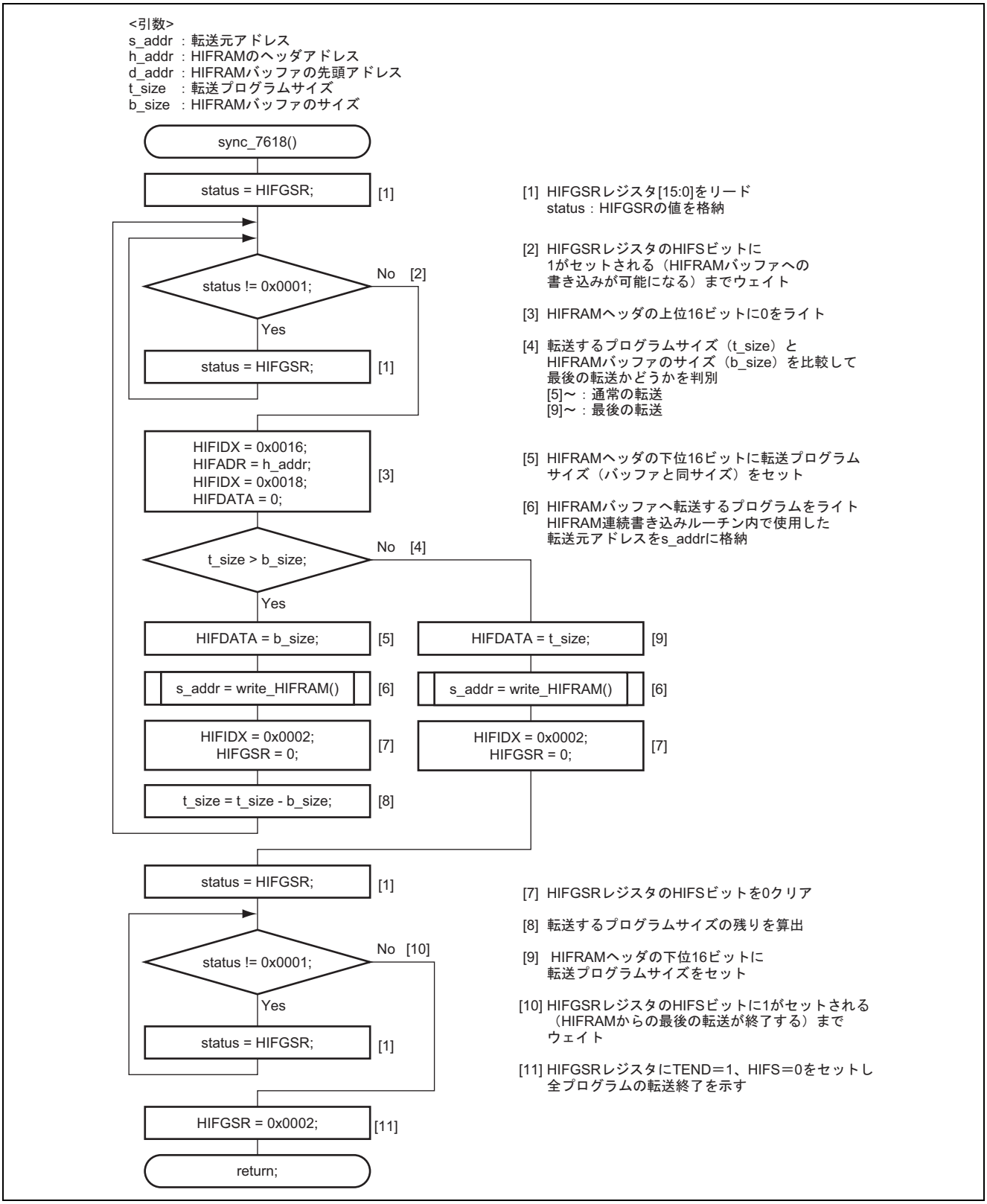
[5] HIFDATAレジスタ[15:0]にライトデータを設定

[6] HIFIDXレジスタでHIFMCRレジスタの下位16ビットを指定し、HIFRAMへの連続ライトモードに設定
（HIFADRレジスタはHIFDATAレジスタをアクセスするたびにオートインクリメント）

[7] HIFIDXレジスタでHIFDATAレジスタ[31:16]を指定

[8] HIFRAMへブートプログラムのライトが終了するまでHIFDATAレジスタへ連続ライト（先頭の4バイトはライト済みのためi = 2からループ開始）

6. HIFRAM バッファ書き込みルーチン



2.2.6 プログラムリスト

1. SH7618 プログラム 1

```
/*****************************************************************************************************************/
// SH7618 HIF boot mode application note
//  A program is transmitted to U memory from HIFRAM buffer
//      CPU:      SH7618,SH-2,Big Endian
//      Clock:    External input = 25MHz
//                CPU clock = 100MHz
//                External BUS clock = 50MHz
//                Peripheral clock = 50MHz
//      Written:  '04/4 Rev.2.0
/*****************************************************************************************************************/
//------------ Symbol Definition ------------------------------------------------------------------------------
#define HIF_BUFF_SIZE_BYTE    0xC0                              /* HIFRAM buffer size(Byte)                  */
#define HIF_BUFF_SIZE_LONG    (HIF_BUFF_SIZE_BYTE/4)            /* HIFRAM buffer size(long word access)      */

struct st_hifbuff{                                              /* definition of HIFRAM buffer               */
    unsigned long d_size_byte;                                  /* HIFRAM header(transmission data size)     */
    union{
        unsigned long long_size[HIF_BUFF_SIZE_LONG];            /* HIFRAM buffer(long word access)           */
        unsigned char byte_size[HIF_BUFF_SIZE_BYTE];            /* HIFRAM buffer(byte access)                */
    } BUFF;
};

//------------ Definition of HIF Register ---------------------------------------------------------------------
union st_hifgsr{                                                /* definition of HIFGSR                      */
    unsigned long LONG;                                         // Long Access
    struct {                                                    // Bit Access
        unsigned long :24;                                      // reserve
        unsigned long :6;                                       // no use
        unsigned long TEND:1;                                   // TEND
        unsigned long HIFS:1;                                   // HIFS
    } BIT;
};

//------------ Function Definition ----------------------------------------------------------------------------
void hifram_main(void);

extern void jump_uram(void);

//-----------------------------------------------------------------------------------------------------------------
#pragma section _HIF_BUFF
volatile struct st_hifbuff ST_HIFBUFF;
#pragma section

#define ST_HIFGSR (*(volatile union st_hifgsr*)0xF84D0004)      /* SH7618 HIFGSR register address            */

#define URAM (*(volatile unsigned long*)0xE55FF000)             /* U memory top address,Non Cache area       */

unsigned char* Uram_addr_pt_byte;                               /* pointer of U memory address(byte access)  */
unsigned long* Uram_addr_pt_long;                               /* pointer of U memory address(long word access)*/
```

```
/*****************************************************************************************************************/
/*      Main routine                                                                                             */
/*****************************************************************************************************************/
void hifram_main( void )
{
    unsigned long i;
    unsigned long t_count;                              /* times of transmission when long word access     */

    //----------initialize HIFGSR
    ST_HIFGSR.LONG = 0;

    //----------clear in HIFRAM buffer
    ST_HIFBUFF.d_size_byte = 0;
    for(  i=0 ;  i<HIF_BUFF_SIZE_LONG ; i++ )
    {
        ST_HIFBUFF.BUFF.long_size[i] = 0;
    }

    Uram_addr_pt_long = &URAM;                          /* transmission destination address                */
    ST_HIFGSR.BIT.HIFS = 1;                             /* set HIFS=1(The writing to HIFRAM is possible)    */

    while(1)
    {
        while(ST_HIFGSR.BIT.HIFS != 0);                 /* wait until write end to HIFRAM from external device*/

        if( ST_HIFGSR.BIT.TEND == 1 )                   /* transmission end(TEND=1)                         */
        {
            break;                                      /* escape from loop                                 */
        }
        else
        {
            //----------transmit to U memory from HIFRAM buffer
            //----------transmit by long-word size
            if( ST_HIFBUFF.d_size_byte == HIF_BUFF_SIZE_BYTE )
            {
                t_count = ST_HIFBUFF.d_size_byte/4;     /* times of transmission                            */

                for( i=0; i<t_count; i++ )
                {
                    *Uram_addr_pt_long = ST_HIFBUFF.BUFF.long_size[i];
                    Uram_addr_pt_long++;
                }
            }
            //----------transmit by byte size
            else if( ST_HIFBUFF.d_size_byte < HIF_BUFF_SIZE_BYTE )
            {
                Uram_addr_pt_byte = (unsigned char*)Uram_addr_pt_long;

                for( i=0; i<ST_HIFBUFF.d_size_byte; i++ )
                {
                    (*Uram_addr_pt_byte) = ST_HIFBUFF.BUFF.byte_size[i];
                    Uram_addr_pt_byte++;
                }
            }
            else
            {
                while(1);                               /* error                                            */
            }

            ST_HIFBUFF.d_size_byte = 0;                 /* clear the HIFRAM header                          */
            ST_HIFGSR.BIT.HIFS = 1;                     /* set HIFS=1(The writing to HIFRAM is possible)    */
        }
    }
    //----------Execution of the transmitted program
    jump_uram();
}
```

2. SH7618 プログラム 2

```
;*************************************************************************************************************/
;   function         : jump_uram
;   operation    : Execution of the transmitted program
;   CUP      : SH7618
;   date     : 2004.4
;*************************************************************************************************************/
        .EXPORT  _jump_uram
;
; MS7618 U memory address,user program start address
URAM_START:  .equ     H'E55FF000
        .SECTION     HIF_P, CODE, ALIGN=4
;*************************************************************************************************************/
;
_jump_uram:
;
    mov.l    #URAM_START,R10             ;Program Start
    jmp @R10
    nop
;
    .END
;
```

3. SH7641 プログラム

```
/*****************************************************************************************************************/
// SH7618 HIF boot mode application note
//  SH7618 is booted in HIF boot mode, and a program is written to a HIFRAM buffer
//       CPU:     SH7641,SH-3 DSP,Big Endian
//       Clock:   External input = 12.5MHz
//                CPU clock = 100MHz
//                External BUS clock = 50MHz
//                Peripheral clock = 25MHz
//       Written: '04/4 Rev.2.0
/*****************************************************************************************************************/

#include "7641.h"

/*****************************************************************************************************************/
/*  Protocol declaration of the function                                                                         */
/*****************************************************************************************************************/
/*------------ Symbol Definition ---------------------------------------------------------------------------------*/
#define BOOT_STRAGE_ADDR  0xA5600000                          // storing address of a boot program
#define HIFRAM_START      0x0000                              // write address of HIFRAM
#define BOOT_P_SIZE       0x300                               // boot program size(Byte)

#define URAM_STRAGE_ADDR  0xA5601000                          // storing address of a uram program
#define URAM_P_SIZE       0x300                               // uram program size(Byte)

#define HIF_BUFF_SIZE     0xC0                                // HIFRAM buffer size(192Byte)
#define HIF_HEAD_ADDR     0x0300                              // HIFRAM header address
#define HIF_BUFF_ADDR     0x0304                              // HIFRAM buffer address

    //----------The value when specifying the register of HIF
#define SEL_HIFMCR_LO     0x000A                              // HIFMCR[15:0]
#define SEL_HIFBCR_LO     0x003E                              // HIFBCR[15:0]
#define SEL_HIFADR_LO     0x0016                              // HIFADR[15:0]
#define SEL_HIFDATA_UP    0x0018                              // HIFDATA[31:16]
#define SEL_HIFGSR_LO     0x0002                              // HIFGSR[15:0]

/*------------ Function Definition -------------------------------------------------------------------------------*/
void main_7641(void);

unsigned short* write_HIFRAM(unsigned short* , unsigned short , unsigned short);
void hif_boot(void);

void sync_7618(unsigned short* , unsigned short , unsigned short , unsigned long , unsigned short );

/*-----------------------------------------------------------------------------------------------------------------*
/
    //----------The address when accessing the register of HIF
// select of the register of HIF
#define HIF_REG_SEL       (*(volatile unsigned short*)0xB4001000)
// data write to the register of HIF
#define HIF_DATA_WR       (*(volatile unsigned short*)0xB4000000)
// data read from the HIFGSR register
#define HIF_GSR_RD        (*(volatile unsigned short*)0xB4001004)
```

```
/*****************************************************************************************************************/
/*      Main routine                                                                                             */
/*****************************************************************************************************************/
void main_7641(void)
{
    //----------set of bus interface
    BSC.CS5ABCR.BIT.BSZ = 2;
    BSC.CS5AWCR = 0x00000C01;

    //----------set as a  CS5A function
    PFC.PCCR.BIT.PC1MD = 3;                                     /* bit[2-3]-PC1MD=b'11 : PC1=>CS5A          */

    //----------boot program is written to HIFRAM
    write_HIFRAM((unsigned short*)BOOT_STRAGE_ADDR,HIFRAM_START,BOOT_P_SIZE);

    //----------SH7618 is booted
    hif_boot();                                                 /* HIF boot                                 */

    //----------change of bus wait cycle
    BSC.CS5AWCR = 0x00000901;

    //----------synchronization is taken SH7618, and a program is written to HIFRAM
    sync_7618((unsigned short*)URAM_STRAGE_ADDR , HIF_HEAD_ADDR ,
                        HIF_BUFF_ADDR , URAM_P_SIZE , HIF_BUFF_SIZE);

    while(1);                                                   /* Loop                                     */
}

/*****************************************************************************************************************/
//  function:   write_HIFRAM
//  operation:  boot program writing to HIFRAM
//  argument:   trans_src_addr ; storing address of a boot program
//              hif_addr      ; head address of HIFRAM which transmits a boot program
//              t_size        ; transmit program size
//  return:     trans_src_addr ; storing address of a boot program
/*****************************************************************************************************************/
unsigned short* write_HIFRAM(unsigned short *trans_src_addr , unsigned short hif_addr , unsigned short t_size)
{
    volatile unsigned short time , i ;

    time = t_size/2 + t_size%2;                                 /* calculate times of transmission          */

    HIF_REG_SEL = SEL_HIFADR_LO;                                /* select HIFADR register                   */
    HIF_DATA_WR = hif_addr;                                     /* set of HIFRAM address                    */

    HIF_REG_SEL = SEL_HIFDATA_UP;                               /* select HIFDATA register[31:16]           */
    HIF_DATA_WR = (*trans_src_addr);                            /* set to HIFDATA register[31:16]           */

    trans_src_addr ++;                                          /* storing address is increment(+2)         */

    HIF_DATA_WR = (*trans_src_addr);                            /* set to HIFDATA register[15:0]            */

    trans_src_addr ++;                                          /* storing address is increment(+2)         */

    HIF_REG_SEL = SEL_HIFMCR_LO;                                /* select HIFMCR register[15:0]             */
    HIF_DATA_WR = 0x00A0;                                       /* set as continuation write mode           */

    HIF_REG_SEL = SEL_HIFDATA_UP;                               /* select HIFDATA register[31:16]           */

    for( i=2 ; i<time ; i++){
        HIF_DATA_WR = (*trans_src_addr);                        /* write to HIFDATA register(HIFRAM)        */

        trans_src_addr ++;                                      /* storing address is increment(+2)         */
    }

    return(trans_src_addr);
}
```

```
/*****************************************************************************************************************/
//  function:    hif_boot
//  operation:   SH7618 is booted
//  argument:    non-argument
//  return:      non-return
/*****************************************************************************************************************/
void hif_boot(void)
{
    //----------initialize HIFGSR
    HIF_REG_SEL = SEL_HIFGSR_LO;                                /* select HIFGSR register[15:0]             */
    HIF_DATA_WR = 0x0000;                                       /* H'0000 is written to HIFGSR register     */

    //----------boot SH7618
    HIF_REG_SEL = SEL_HIFBCR_LO;                                /* select HIFBCR                            */
    HIF_DATA_WR = 0x0000;                                       /* clear AC bit of HIFBCR register          */
}

/*****************************************************************************************************************/
//  function:    sync_7618
//  operation:   synchronization is taken SH7618, and a program is written to HIFRAM
//  argument:    *s_addr ; pointer of source address
//               h_addr ; HIFRAM header address
//               b_addr ; HIFRAM buffer address
//               t_size ; transmission data size
//               b_size ; HIFRAM buffer size
//  argument:    non-argument
/*****************************************************************************************************************/

void sync_7618(unsigned short* s_addr , unsigned short h_addr , unsigned short b_addr , unsigned long t_size , unsigned
short b_size)
{
    volatile unsigned short status;

    while(1){
        status = HIF_GSR_RD;                                    /* read from HIFGSR register[15:0]          */

        while(status != 0x0001){                                /* wait until HIFGSR.HIFS=1                 */
            status = HIF_GSR_RD;                                /* read from HIFGSR register[15:0]          */
        }

    //----------preparation write to header
        HIF_REG_SEL = SEL_HIFADR_LO;                            /* select HIFADR register[15:0]             */
        HIF_DATA_WR = h_addr;                                   /* set of HIFRAM header address             */
        HIF_REG_SEL = SEL_HIFDATA_UP;                           /* select HIFDATA register[31:16]           */
        HIF_DATA_WR = 0;                                        /* set to HIFDATA register[31:16]           */

    //----------usual transmission
        if(t_size > b_size)
        {
    //----------transmission data size is written to header
            HIF_DATA_WR = b_size;                               /* write to HIFDATA register[15:0]          */

    //----------write to HIFRAM buffer
            s_addr = write_HIFRAM(s_addr,b_addr,b_size);

    //----------transmission end process
            HIF_REG_SEL = SEL_HIFGSR_LO;                        /* select HIFGSR register[15:0]             */
            HIF_DATA_WR = 0x0000;                               /* set to HIFGSR register[15:0]             */
                                                                // TEND=0 , HIFS=0
        }

    //----------last transmission
        else
        {
    //----------transmission data size is written to header
            HIF_DATA_WR = (unsigned short)t_size;               /* write to HIFDATA register[15:0]          */

    //----------write to HIFRAM buffer
            s_addr = write_HIFRAM(s_addr,b_addr,b_size);
```

```
    //----------transmission end process
            HIF_REG_SEL = SEL_HIFGSR_LO;                    /* select HIFGSR register[15:0]          */
            HIF_DATA_WR = 0x0000;                           /* set to HIFGSR register[15:0]          */
                                                            // TEND=0 , HIFS=0

            break;                                          /* escape from loop                      */
         }
    //----------calculate transmission data size
        t_size = t_size - b_size;
    }

    status = HIF_GSR_RD;                                    /* read from HIFGSR register[15:0]       */

    while( status==0x0000 )                                 /* wait until HIFGSR.HIFS=1              */
    {
        status = HIF_GSR_RD;                                /* read from HIFGSR register[15:0]       */
    }

    HIF_DATA_WR = 0x0002;                                   /* set to HIFGSR register[15:0]          */
                                                            // TEND=1 , HIFS=0
}
```

2.3 HIFRAM を 2 バンク使用したプログラム転送

2.3.1 概要

本タスク例ではより大容量のプログラムを高速に転送し実行するため，HIFRAM を 2 バンク使用したプログラム転送方法について述べています。

HIFRAM を 2 バンク使用することで，HIFRAM へのプログラムの書き込みと，HIFRAM から SDRAM へのプログラム転送を同時に実行でき，より高速なプログラム転送が可能になります。

2.3.2 仕様

1. SH7641 を HIF に接続して，SH7618 を HIF ブートモードで起動します。
2. SH7618 起動後，HIFRAM を介して転送プログラムを内蔵 RAM に転送し，実行します。
3. 転送プログラムは, HIFRAM を 2 バンク使用して SDRAM へアプリケーションプログラムを転送します。
4. アプリケーションプログラムの転送では，SH7618 と SH7641 は HIFRAM の異なるバンクをアクセスします。
5. SH7618 と SH7641 がアクセスするバンクを切り替えるため，HIF 汎用ステータスレジスタ (HIFGSR) を使用して SH7641 と SH7618 の間で同期をとります。
6. SH7641 は転送量を示すヘッダを付けて HIFRAM へアプリケーションプログラムを書き込みます。
7. SH7618 はヘッダで指定された量を HIFRAM から SDRAM へ転送します。
8. SDRAM へアプリケーションプログラムの転送が終了すると，SH7618 は SDRAM 上のアプリケーションプログラムを実行します。
9. 本タスク例において，SH7641 の動作周波数は，CPU クロック 100MHz，外部バスクロック 50MHz，周辺クロック 25MHz とします。また，SH7618 の動作周波数は，内部クロック 100MHz，外部バスクロック 50MHz，周辺クロック 50MHz とします。

【注】 本タスク例では，HIFRAM を 2 バンク使用したプログラム転送についてのみ述べています。HIF ブートモードでの起動から内蔵 RAM へ転送したプログラムの実行までは，本アプリケーションノートの「2.2 HIF ブートによる内蔵 RAM へのプログラム転送」を参照してください。

図 16 に SH7618，SH7641 および SDRAM の接続例を示します。

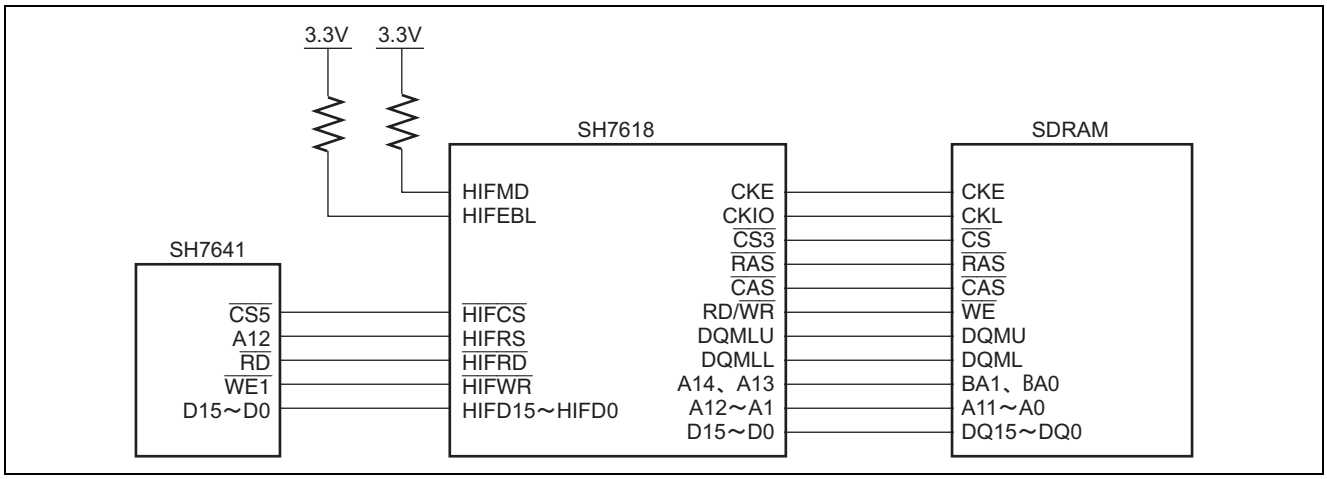
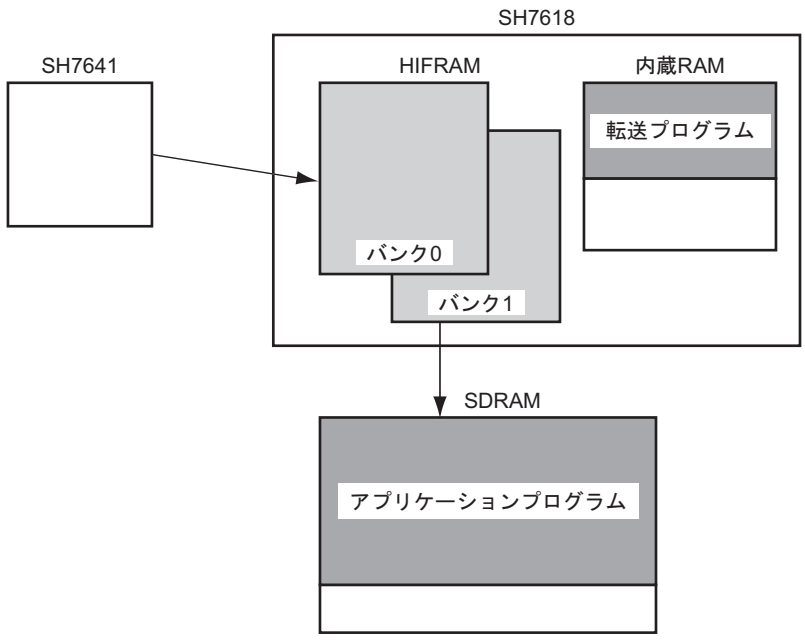


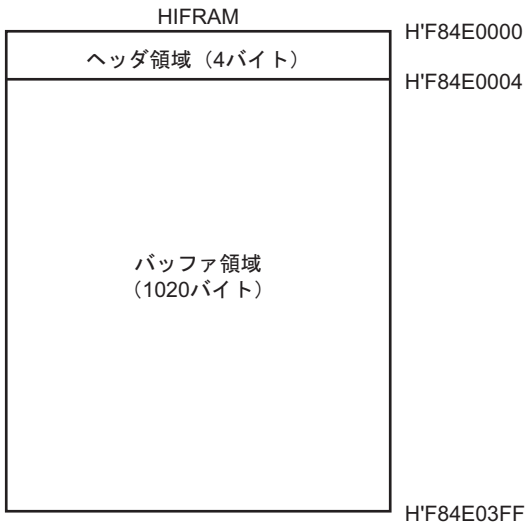
図 16 接続例



【注】内蔵RAM 上のプログラムを実行し、SDRAM へプログラムを転送します。

図 17　HIFRAM を 2 バンク使用したプログラム転送

図 18 に HIFRAM のメモリマップを示します。



【注】 ヘッダ領域には、1 回の転送サイズを格納します。
また、本メモリマップはバンク0およびバンク1 に共通です。

図 18　HIFRAM のメモリマップ

図 19 に HIFRAM から SDRAM への転送時の転送するプログラムサイズとバッファサイズの関係を示します。

本タスク例では，SH7641 がヘッダ領域に転送するプログラムサイズを書き込みます。SH7618 は，ヘッダ領域のデータを参照して HIFRAM バッファから SDRAM へプログラムを転送します。

SH7618 は，HIFRAM から SDRAM へのアプリケーションプログラムの転送をより高速に行うため，ロングワード単位で転送しています。転送時のデータサイズは, 転送するプログラムサイズに応じて決定します。

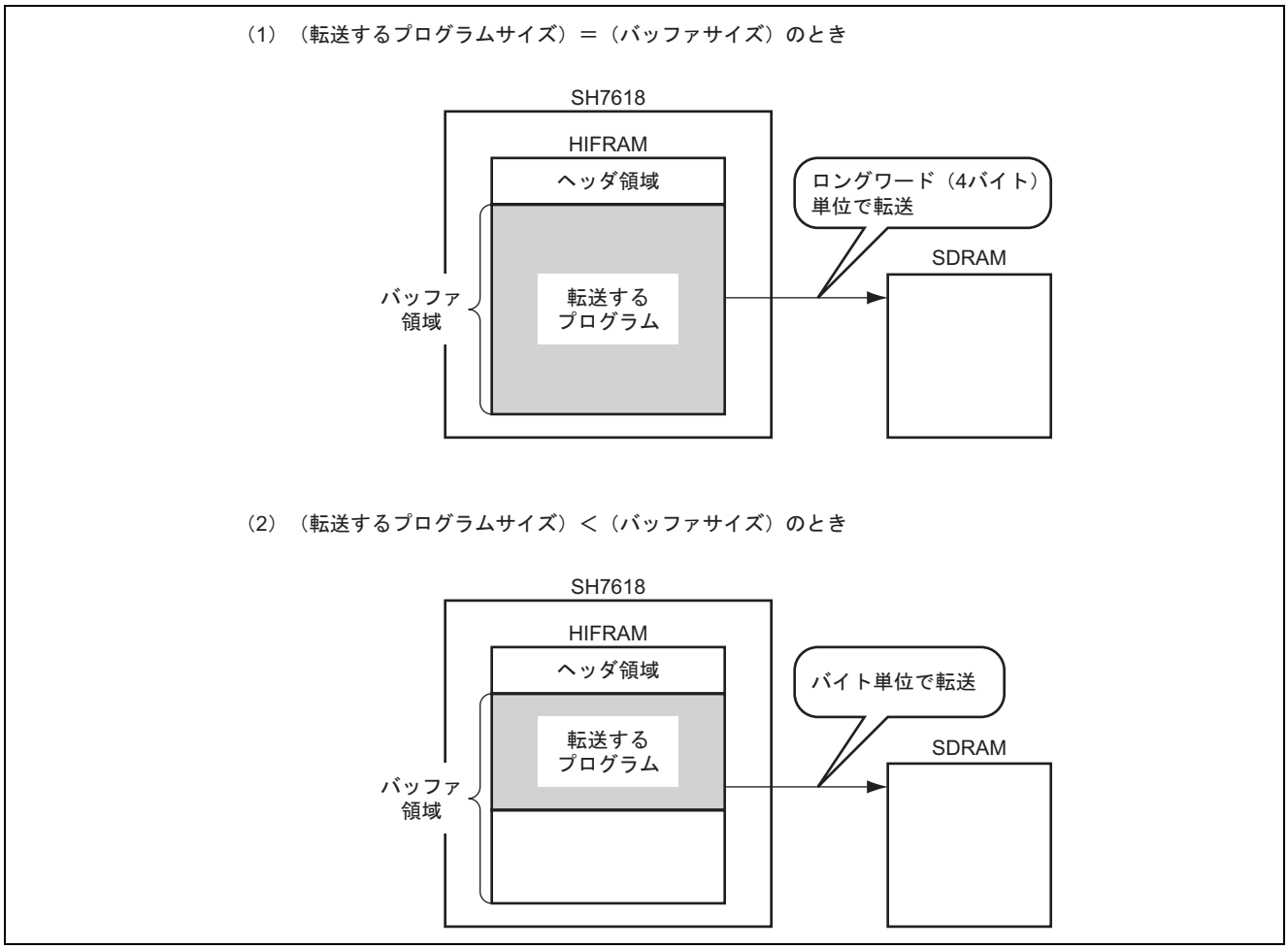


図 19　HIFRAM から SDRAM への転送

HIFGSR レジスタは，ユーザが自由に定義して使用可能で，本タスク例では SH7641 と SH7618 の間で同期をとるために，ビット 1 を TEND に，ビット 0 を HIFS に定義して使用しています。表 17 に本タスク例での HIFGSR レジスタの内容を示します。

表 17　HIFGSR レジスタの内容

ビット	ビット名	初期値	説明
31〜16	—	すべて 0	リザーブビット
15 〜 2	STATUS15 〜 STATUS2	すべて 0	汎用ステータスビット 本タスク例では使用していません。
1	TEND	0	トランスミットエンド プログラムの転送が終了したかどうかを示します。 SH7618 は本ビットを参照してプログラム転送が終了したことを確認し，内蔵 RAM へ転送したプログラムを実行します。 SH7641 は全プログラムの転送が終了すると，本ビットを 1 にセットします。 1: 全プログラムの転送が終了 SH7618 は内蔵 RAM に転送したプログラムを実行 0: プログラム転送中 内蔵 RAM へ転送するプログラムが SH7641 側に残存している
0	HIFS	0	HIFRAM ステータス HIFRAM の状態を示します。 SH7618 と SH7641 は本ビットを参照して転送を行います。 SH7618 は，SH7641 から HIFRAM への転送が可能になったときに，1 のセットのみ行います。 SH7641 は，HIFRAM バッファへのプログラム転送が終了したときに，0 クリアのみ行います。 1: SH7641 から HIFRAM への書き込みが可能な状態 SH7618 は待機状態 0: SH7641 から HIFRAM への書き込みが不可の状態 SH7618 は HIFRAM 上のプログラムを SDRAM へ転送

2.3.3 動作説明

図 20 に HIFGSR の値と，そのときの動作状態を示し，表 18 に SH7618 および SH7641 の動作内容を説明します。また，図 21 に HIFSCR レジスタの BMD ビットおよび BSEL ビットの値とそのときの動作状態を示します。

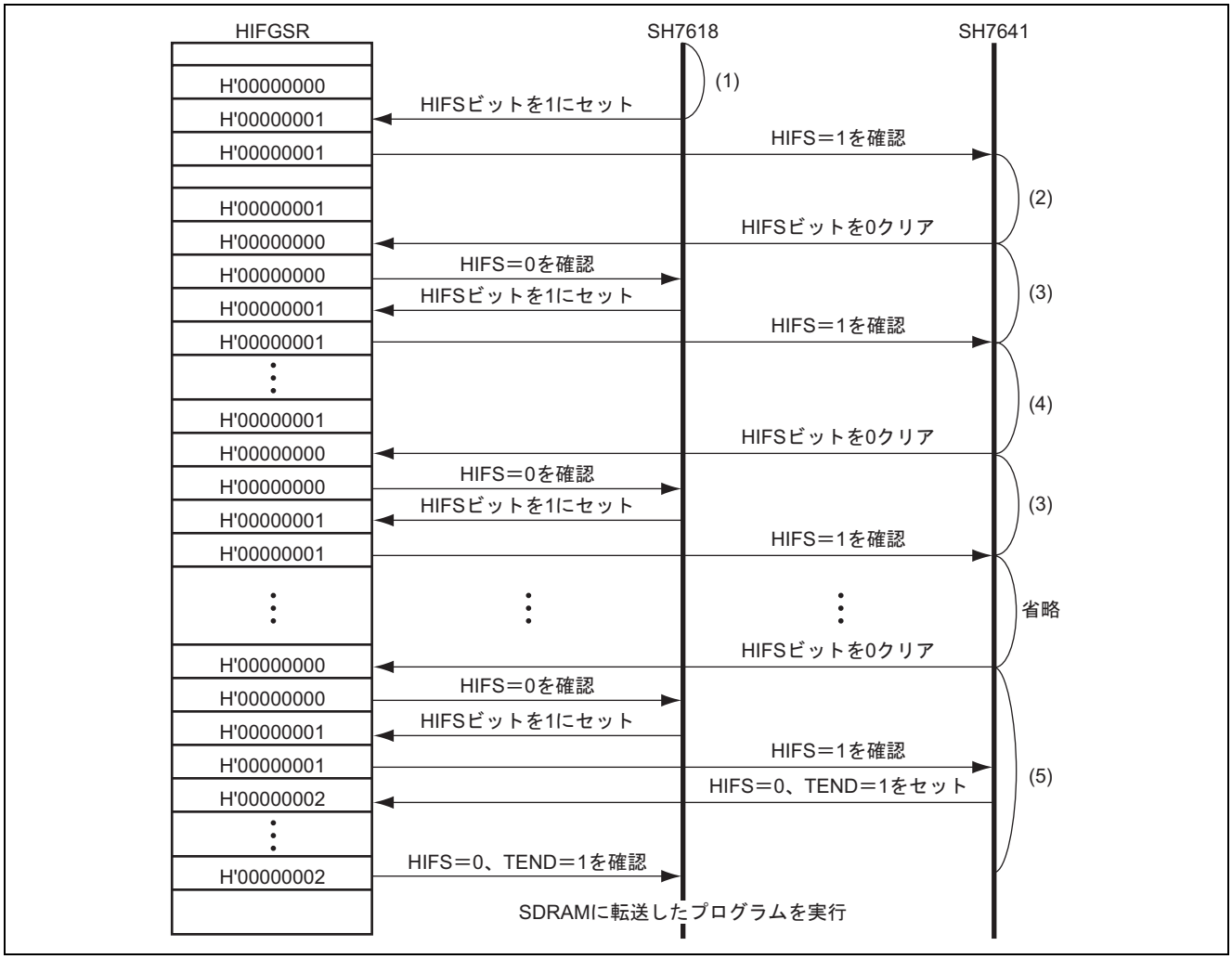
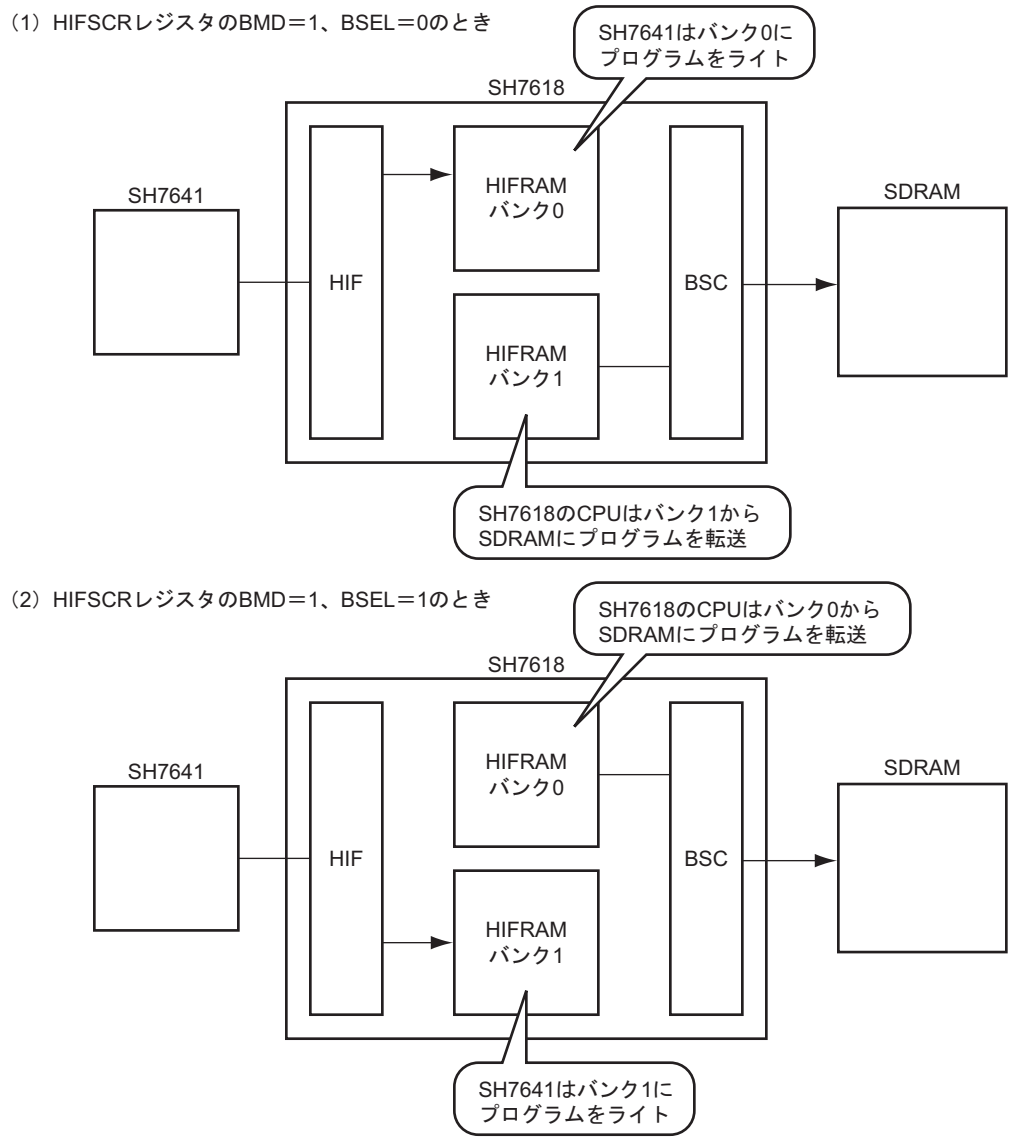


図 20　HIFGSR の値と動作状態

表 18　SH7618 および SH7641 の動作説明

	SH7618 の動作内容	SH7641 の動作内容
(1)	• HIFRAM をバンク 0，バンク 1 共に初期化 • HIFGSR レジスタの HIFS ビットを 1 にセット	• HIFGSR レジスタの HIFS ビットが 0 の間ウェイト
(2)	• HIFGSR レジスタの HIFS ビットが 1 の間ウェイト	• HIFGSR レジスタの HIFS ビットが 1 にセットされたことを確認 • HIFRAM バッファにプログラムを書き込む • プログラム書き込み終了後 HIFGSR レジスタの HIFS ビットを 0 クリア
(3)	• HIFGSR レジスタの HIFS ビットが 0 クリアされたことを確認 • HIFSCR レジスタの BSEL ビットを反転 (SH7618 および SH7641 のアクセスバンクの切り替え) • HIFGSR レジスタの HIFS ビットを 1 にセット • SH7641 が書き込んだ HIFRAM バッファ上のプログラムを SDRAM に転送開始	• HIFGSR レジスタの HIFS ビットが 0 の間ウェイト
(4)	• SDRAM へのプログラム転送終了後 HIFRAM へッダをクリア • HIFGSR レジスタの HIFS ビットが 1 の間ウェイト	• HIFGSR レジスタの HIFS ビットが 1 にセットされたことを確認 • HIFRAM バッファにプログラムを書き込む • プログラム書き込み終了後 HIFGSR レジスタの HIFS ビットを 0 クリア
(5)	• HIFGSR レジスタの HIFS ビットが 0 クリアされたことを確認 • HIFSCR レジスタの BSEL ビットを反転 (SH7618 および SH7641 のアクセスバンクの切り替え) • HIFGSR レジスタの HIFS ビットを 1 にセット • SH7641 が書き込んだ HIFRAM バッファ上のプログラムを SDRAM に転送 • HIFGSR レジスタの HIFS = 0, TEND = 1 を確認し SDRAM へ転送したプログラムへ処理を移行	• 転送する全プログラムの書き込みが終了し HIFGSR レジスタの HIFS ビットを 0 にクリア • HIFGSR レジスタの HIFS ビットが 0 の間ウェイト • HIFGSR レジスタの HIFS ビットが 1 にセットされたことを確認 • HIFGSR レジスタに HIFS = 0, TEND = 1 を書き込む



【記号説明】
BSC：バスステートコントローラ

【注】 SH7618のCPUとSH7641は、異なるバンクへ同時にアクセス可能なため、SH7641のHIFRAMへの書き込みとSH7618のCPU によるSDRAMへの転送は同時に実行します。

図 21　HIFSCR レジスタの値と動作状態

2.3.4 ソフトウェア説明

1. モジュール説明

表 19 に本タスク例で使用する SH7618 のモジュール, 表 20 に本タスク例で使用する SH7641 のモジュールを示します。

表 19　SH7618 のモジュール説明

モジュール名	ラベル名	機能
U メモリメインルーチン	uram_main	SH7618 の初期設定と SH7641 が HIFRAM バッファに書き込んだプログラムを内蔵 RAM に転送 【注】SH7618 の内蔵 RAM (U メモリ) 上で実行します
転送プログラム実行ルーチン	jump_uram	SDRAM へプログラムカウンタを移動し転送したプログラムを実行 【注】本モジュールはアセンブリ言語で記述しています

表 20　SH7641 のモジュール説明

モジュール名	ラベル名	機能
SH7641 メインルーチン	main_7641	外部バスおよび端子の設定と各モジュールの呼び出し
HIFRAM 連続書き込みルーチン	write_HIFRAM	HIFRAM へプログラムを書き込む
SH7618 起動ルーチン	hif_boot	SH7618 を HIF ブートモードで起動
HIFRAM バッファ書き込みルーチン	sync_7618	SH7618 と同期をとり HIFRAM バッファにプログラムを書き込む

2. 使用内蔵レジスタ説明

表 21 に本タスク例で使用する SH7618 の内蔵レジスタを示します。

表 21　SH7618 の使用内蔵レジスタ説明

レジスタ名			設定値	機能
	ビット	ビット名		
HIFADR				HIF アドレスレジスタ
	31 ～ 10	—	すべて 0	リザーブビット
	9 ～ 2	A9 ～ A2	—	HIFRAM アドレス指定 外部デバイスが HIFRAM をアクセスする際のアドレスを 32 ビット境界で指定します。
	1 0	—	0	リザーブビット
HIFDATA				HIF データレジスタ
	31 ～ 0	D31 ～ D0	—	32 ビットデータ 外部デバイスから HIFRAM へのアクセスに使用します。
HIFIDX				HIF インデックスレジスタ
	31 ～ 8	—	すべて 0	リザーブビット
	7 ～ 2	REG5 ～ REG0	—	HIF レジスタ選択 外部デバイスがアクセスする HIF のレジスタを選択します。
	1 0	BYTE1 BYTE0	—	HIF レジスタ内バイト選択 外部デバイスが HIF のレジスタをアクセスするときのワード位置を指定するビットです。

レジスタ名 ビット	レジスタ名 ビット名	設定値	機能
HIFMCR			HIF メモリ制御レジスタ
31 〜 8	—	すべて 0	リザーブビット
7	LOCK	—	ロックビット 外部デバイスが HIFRAM を連続アクセスするときに使用します。
6	—	0	リザーブビット
5	WT	—	ライトビット 本ビットを 1 にセットすると HIFDATA レジスタの値が HIFADR レジスタに対応する HIFRAM の位置へ書き込まれます。
4	—	0	リザーブビット
3	RD	—	リードビット 本ビットを 1 にセットすると HIFADR レジスタに対応する HIFRAM のデータが HIFDATA に読み出されます。
2 1	—	0	リザーブビット
0	AI/AD	0	アドレスオートインクリメント/デクリメント LOCK = 1 かつ AI/AD = 0 のとき SH7641 が HIFDATA レジスタをアクセスするたびに HIFADR がインクリメント (+4) され,連続した HIFRAM のアドレスに対しての書き込みおよび読み出しが可能です。
HIFGSR			HIF 汎用ステータスレジスタ
31 〜 16	—	すべて 0	リザーブビット
15 〜 2	STATUS15 〜 STATUS2	すべて 0	汎用ステータスビット 本タスク例では使用していません。
1	TEND	—	トランスミットエンド プログラム転送の状況を示します。 TEND = 1 のとき SH7618 は内蔵 RAM へ転送したプログラムを実行します。 TEND = 0 のとき SH7618 は内蔵 RAM へプログラムを転送します。
0	HIFS	—	HIFRAM ステータス HIFRAM の状態を示します。 HIFS = 0 のとき SH7641 から HIFRAM への転送が可能です。 HIFS = 1 のとき SH7641 から HIFRAM への転送は禁止です。

レジスタ名		設定値	機能
ビット	ビット名		
HIFSCR			HIF ステータスコントロールレジスタ
31 〜 12	—	すべて 0	リザーブビット
11	DMD	0	DREQ モード POL ビットと合わせて HIFDREQ 端子のアサートモードを制御します。本タスク例では使用していません。
10	DPOL	0	DREQ ポラリティ DMD ビットと合わせて HIFDREQ 端子のアサートモードを制御します。本タスク例では使用していません。
9	BMD	—	HIFRAM バンクモード HIFRAM バンクセレクトと合わせて SH7618 の CPU と HIF に接続した外部デバイスがアクセスする HIFRAM のバンクを選択します。SH7618 の CPU と外部デバイスから同一バンクへのアクセスが競合した場合，外部デバイスのアクセスが優先されます。 BMD = 0, BSEL = 0: SH7618 の CPU，外部デバイス共にバンク 0 をアクセス BMD = 0, BSEL = 1: SH7618 の CPU，外部デバイス共にバンク 1 をアクセス BMD = 1, BSEL = 0: SH7618 の CPU はバンク 1 を，外部デバイスはバンク 0 をアクセス BMD = 1, BSEL = 1: SH7618 の CPU はバンク 0 を，外部デバイスはバンク 1 をアクセス
8	BSEL	—	HIFRAM バンクセレクト BMD ビットと合わせて SH7618 の CPU と HIF に接続した外部デバイスがアクセスする HIFRAM のバンクを選択します。
7	—	0	リザーブビット
6	—	1	リザーブビット
5	MD1	1	HIF モード 1 HIF ブートモードで起動したかどうかを示します。 MD1=1 のとき SH7618 は HIF ブートモードで起動
4 〜 2	—	すべて 0	リザーブビット
1	EDN	0	HIFRAM アクセス時のエンディアン SH7618 の CPU が HIFRAM をアクセスするときのバイトオーダーを指定します。 EDN = 0 のとき SH7618 の CPU は HIFRAM をビックエンディアンでアクセス
0	BO	0	HIFDATA を含め HIF すべてのレジスタをアクセスするときのバイトオーダー HIF に接続した外部デバイスが HIF のレジスタをアクセスするときのバイトオーダーを指定します。 BO = 0 のとき外部デバイスは HIF のレジスタをビックエンディアンでアクセス

表 22 に本タスク例で使用する SH7641 の内蔵レジスタを示します。

表 22　SH7641 の使用内蔵レジスタ説明

レジスタ名		設定値	機能
ビット	ビット名		
CS5ABCR			CS5A 空間バスコントロールレジスタ
10 9	BSZ1 BSZ0	1 0	データバス幅指定 CS5A 空間をアクセスするときのデータバス幅を指定します。 BSZ [1, 0] = B'10 のときデータバス幅を 16 ビットに設定
CS5AWCR			CS5A 空間ウェイトコントロールレジスタ
31 ～ 19	—	すべて 0	リザーブビット
18 17 16	WW2 WW1 WW0	0	ライトアクセスウェイトサイクル数 ライトアクセス時に挿入するウェイト数を指定します。 WW [2-0] = B'000 のとき WR ビットで指定するリードアクセスウェイトと同じ数のウェイトを挿入
15 ～ 13	—	すべて 0	リザーブビット
12 11	SW1 SW0	0 1	アドレス，$\overline{CS}$ アサート→$\overline{RD}$，$\overline{WEn}$ アサート遅延ウェイト数 SW [1, 0] = B'01 のときアドレス，$\overline{CS}$ アサートから $\overline{RD}$，$\overline{WEn}$ アサートまで 1.5 サイクルのウェイトを挿入
10 9 8 7	WR3 WR2 WR1 WR0	—	リードアクセスウェイトサイクル数 リードアクセス時に挿入するウェイト数を指定します。 本タスク例では，SH7618 の HIF ブートモードでの起動前と起動後で挿入するウェイト数を変更しています。 起動前: WR [3-0] = B'1000 (10 サイクルのウェイトを挿入) 起動後: WR [3-0] = B'0010 (2 サイクルのウェイトを挿入)
6	WM	0	外部ウェイトマスク指定 外部ウェイと入力の有効/無効を指定します。 WM = 0 のとき外部ウェイト入力有効
5 ～ 2	—	すべて 0	リザーブビット
1 0	HW1 HW0	0 1	$\overline{RD}$，$\overline{WEn}$ ネゲート→アドレス，$\overline{CS}$ ネゲート遅延サイクル数 HW [1, 0] = B'01 のとき $\overline{RD}$，$\overline{WEn}$ ネゲート→アドレス，$\overline{CS}$ ネゲートまで 1.5 サイクルのウェイトを挿入
PCCR			ポート C コントロールレジスタ
3 2	PC1MD2 PC1MD1	1 1	PC1 モード 2，1 PC1MD[2,1]=B'11 のとき PTC1 端子を CS5A 機能に設定

3. 使用変数説明

表23に本タスク例においてSH7618のプログラム内で使用する変数を, 表24に本タスク例においてSH7641のプログラム内で使用する変数を示します。

表 23　SH7618 プログラム内の変数説明

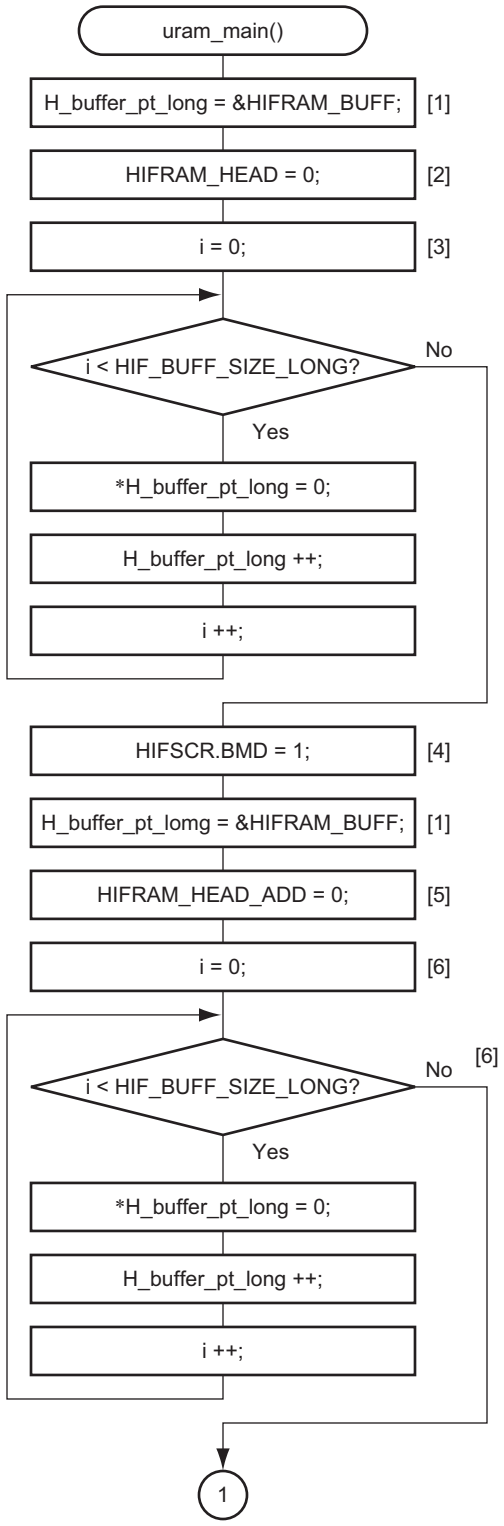
変数	内容	データ長	初期値	使用モジュール名
t_count	ロングワードサイズでの転送における転送回数	4 バイト	—	U メモリメインルーチン
*Sdram_address_pt_byte	バイト単位でのプログラム転送時の転送先 (内蔵 RAM) アドレスを示すポインタ	1 バイト	—	U メモリメインルーチン
*Sdram_address_pt_long	ロングワード単位でのプログラム転送時の転送先 (内蔵 RAM) アドレスを示すポインタ	4 バイト	—	U メモリメインルーチン
*h_buffer_pt_byte	バイト単位でのプログラム転送時の転送元 (HIFRAM バッファ) アドレスを示すポインタ	1 バイト	—	U メモリメインルーチン
*h_buffer_pt_long	ロングワード単位でのプログラム転送時の転送元 (HIFRAM バッファ) アドレスを示すポインタ	4 バイト	—	U メモリメインルーチン

表 24　SH7641 プログラム内の変数説明

変数	内容	データ長	初期値	使用モジュール名
*trans_src_addr	転送元 (ブートプログラム格納) アドレスを示すポインタ	2 バイト	—	HIFRAM 連続書き込みルーチン
hif_addr	書き込みを開始する HIF のアドレス	2 バイト	H'0000	HIFRAM 連続書き込みルーチン
t_size	転送するプログラムサイズ	2 バイト	H'300	HIFRAM 連続書き込みルーチン
*s_addr	転送元 (ブートプログラム格納) アドレスを示すポインタ	2 バイト	—	HIFRAM バッファ書き込みルーチン
h_addr	HIFRAM ヘッダアドレス	2 バイト	H'0000	HIFRAM バッファ書き込みルーチン
b_addr	HIFRAM バッファアドレス	2 バイト	H'0004	HIFRAM バッファ書き込みルーチン
t_size	全転送プログラムサイズ	4 バイト	H'200000	HIFRAM バッファ書き込みルーチン
b_size	HIFRAM バッファサイズ (1 回の転送サイズ)	2 バイト	H'3FC	HIFRAM バッファ書き込みルーチン

2.3.5 フローチャート

1. U メモリメインルーチン



[1] HIFRAMバッファのアドレスをセット

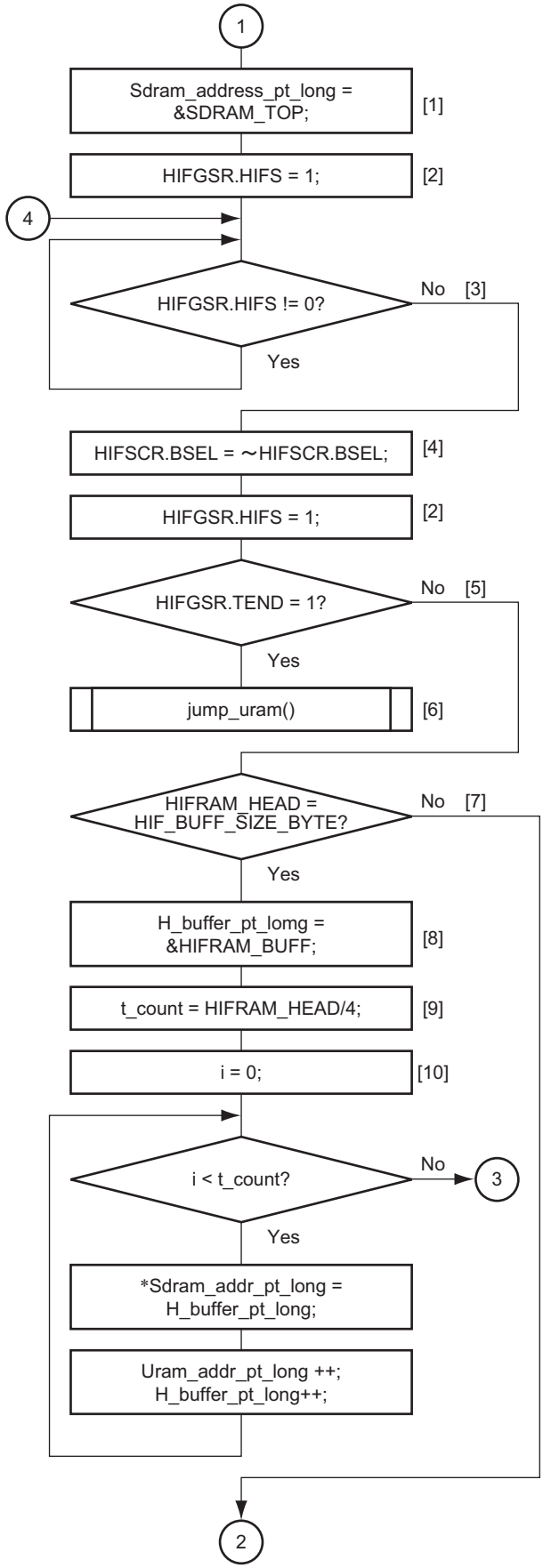
[2] HIFRAMヘッダを初期化（バンク0）

[3] HIFRAMバッファを初期化（バンク0）

[4] SH7618のCPUと外部デバイスからHIFRAMへのアクセスを異なるバンクに設定
SH7618のCPU ：バンク1をアクセス
外部デバイス ：バンク0をアクセス

[5] HIFRAMヘッダを初期化（バンク1）

[6] HIFRAMバッファを初期化（バンク1）



[1] SDRAMの転送先アドレスをセット

[2] HIFGSRレジスタのHIFSビットを1にセットし
外部デバイスからHIFRAMへの書き込みを許可

[3] 外部デバイスからHIFRAMへの書き込みが
終了したかどうかを確認

[4] HIFSRレジスタのBSELビットを反転
(SH7618のCPUおよび外部デバイスによる
アクセスバンクの切り替え)

[5] 全プログラムの転送が終了したかを判定
TEND＝1のとき転送終了

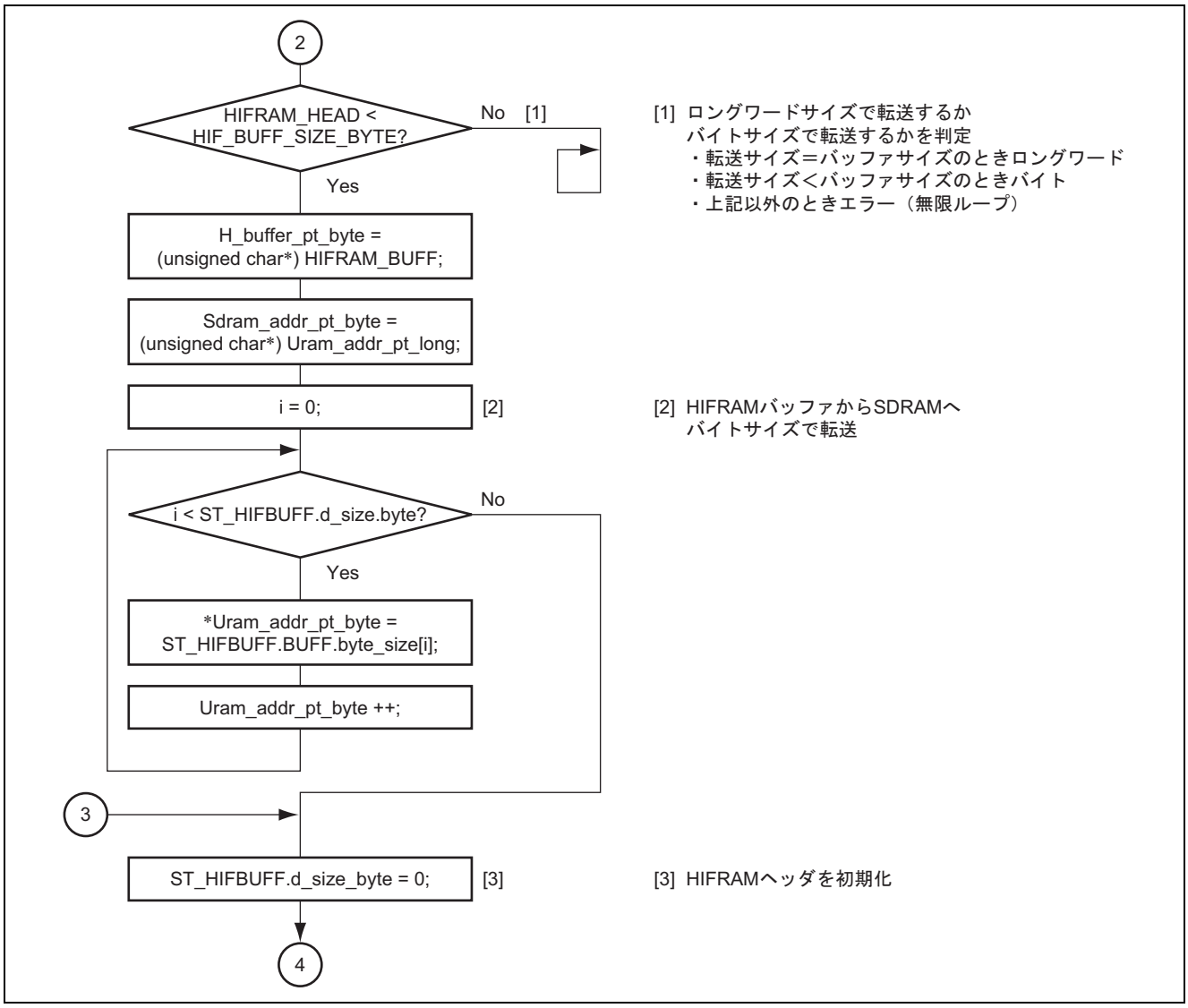
[6] 転送プログラム実行ルーチンの呼び出し

[7] ロングワードサイズで転送するか
バイトサイズで転送するかを判定
・転送サイズ＝バッファサイズのときロングワード
・転送サイズ＜バッファサイズのときバイト
・上記以外のときエラー（無限ループ）

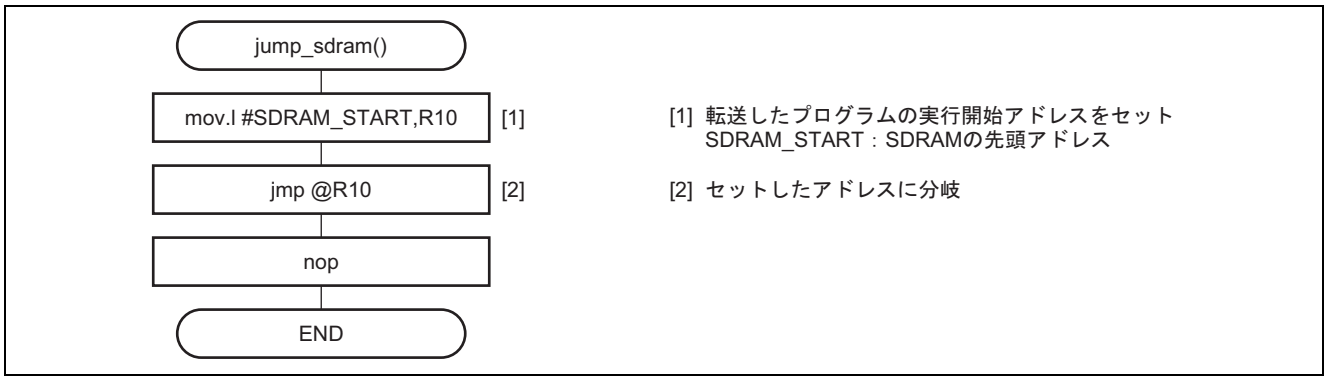
[8] HIFRAMバッファのアドレスをセット

[9] ロングワードサイズでの転送回数をセット

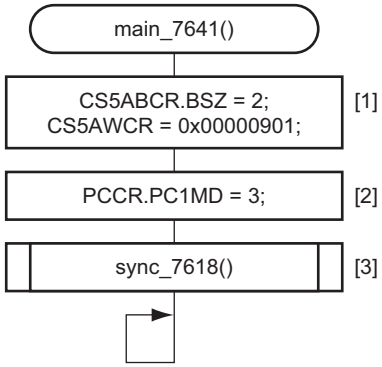
[10] HIFRAMバッファからSDRAMへ
ロングワードサイズで転送



2. 転送プログラム実行ルーチン



3. SH7641 メインルーチン

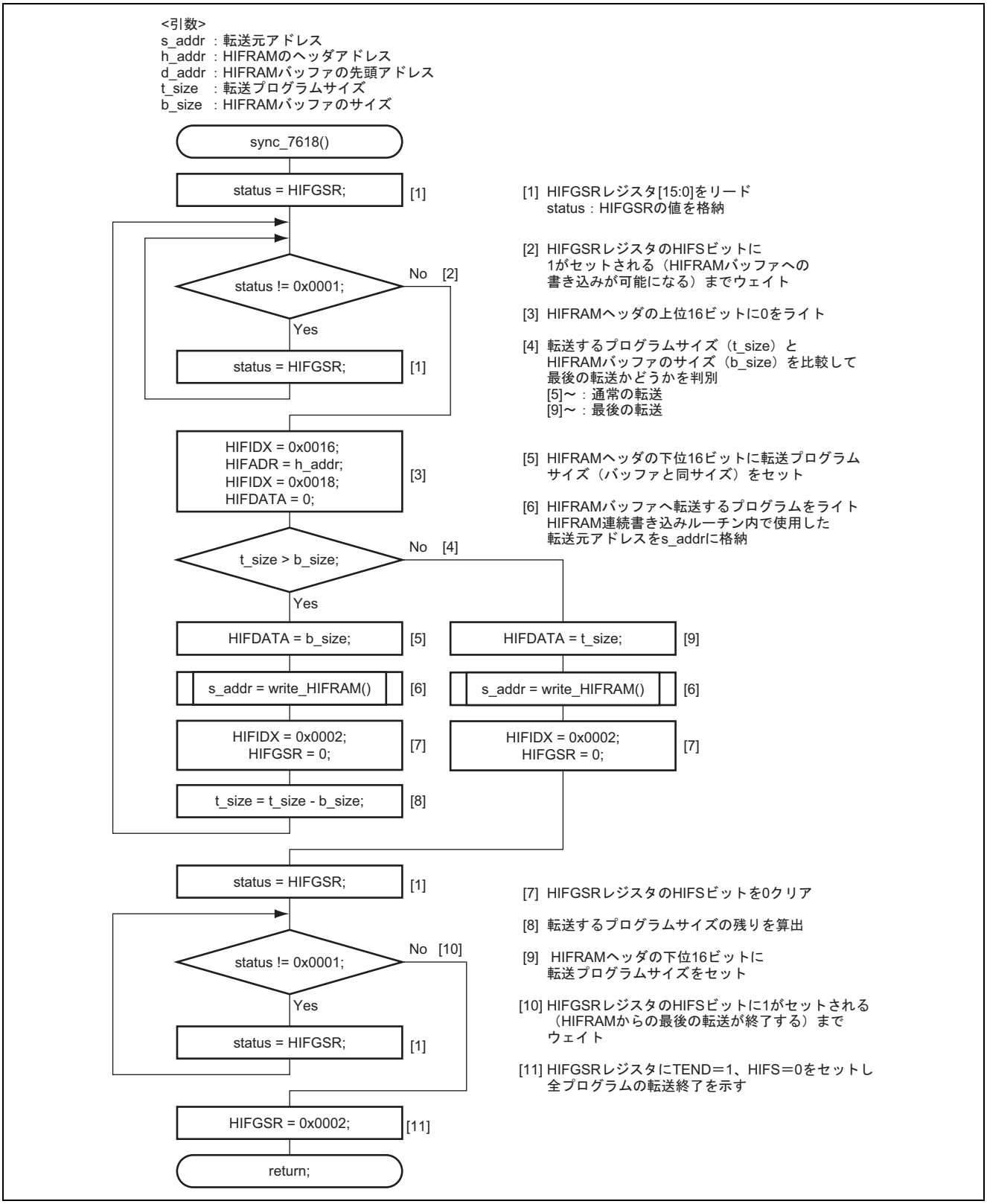


[1] <CS5A空間のバス設定>
- データバス幅16ビット
- アドレス、$\overline{CS5A}$アサートから$\overline{RD}$、$\overline{WEn}$アサートまで1.5サイクルのウェイトを挿入
- $\overline{RD}$、$\overline{WEn}$ネゲートからアドレス、$\overline{CS5A}$ネゲートまで1.5サイクルのウェイトを挿入
- リード/ライトアクセス時1サイクルのウェイトを挿入

[2] 端子機能を$\overline{CS5A}$に設定

[3] HIFRAMバッファ書き込みルーチンの呼び出し
SH7618はSDRAMにプログラムを転送し実行

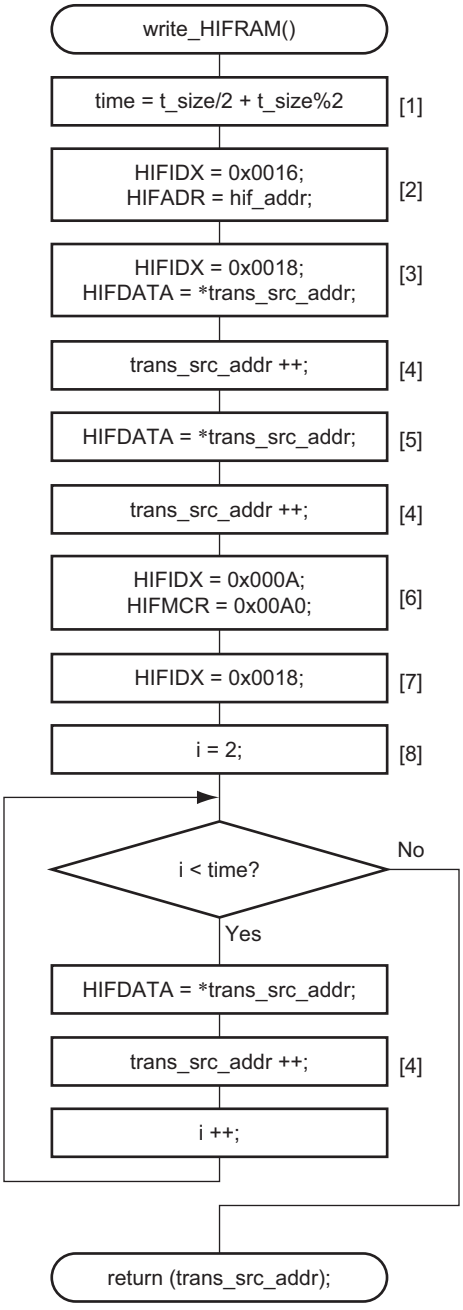
4. HIFRAM バッファ書き込みルーチン



5. HIFRAM への連続書き込みルーチン

<引数>
*trans_src_addr ：転送元アドレスのポインタ
hif_addr ：HIFRAMへのライト開始アドレス
t_size ：転送プログラムサイズ

<戻り値>
trans_src_addr ：転送元アドレス



[1] HIFRAMへのライト回数を設定
time：転送回数

[2] HIFIDXレジスタでHIFADRレジスタの下位16ビットを指定し、HIFADRレジスタにHIFRAMの書き込み開始アドレスをセット

[3] HIFIDXレジスタでHIFDATAレジスタの上位16ビットを指定し、HIFDATAレジスタ[31:16]にライトデータを設定

[4] 転送元アドレスをインクリメント（+2）

[5] HIFDATAレジスタ[15:0]にライトデータを設定

[6] HIFIDXレジスタでHIFMCRレジスタの下位16ビットを指定し、HIFRAMへの連続ライトモードに設定
（HIFADRレジスタはHIFDATAレジスタをアクセスするたびにオートインクリメント）

[7] HIFIDXレジスタでHIFDATAレジスタ[31:16]を指定

[8] HIFRAMへブートプログラムのライトが終了するまでHIFDATAレジスタへ連続ライト（先頭の4バイトはライト済みのためi = 2からループ開始）

2.3.6 プログラムリスト

1. SH7618 プログラム 1

```
/*****************************************************************************************************************/
// SH7618 HIF boot mode application note
//  Program transmission which used two bank of HIFRAM
//      CPU:      SH7618,SH-2,Big Endian
//      Clock:    External input = 25MHz
//                CPU clock = 100MHz
//                External BUS clock = 50MHz
//                Peripheral clock = 50MHz
//      Written:  '04/4 Rev.2.0
/*****************************************************************************************************************/

//------------ Symbol Definition ------------------------------------------------------------------------
#define HIF_BUFF_SIZE_BYTE    0x3FC                         /* HIFRAM buffer size(Byte)                    */
#define HIF_BUFF_SIZE_LONG    (HIF_BUFF_SIZE_BYTE/4)        /* HIFRAM buffer size(long word access)        */

//------------ Definition of HIF Register ---------------------------------------------------------------
union st_hifgsr{                                            /* definition of HIFGSR                        */
    unsigned long LONG;                                     // Long Access
    struct {                                                // Bit Access
        unsigned long :24;                                  // reserve
        unsigned long :5;                                   // no use
        unsigned long BS:1;                                 // BS
        unsigned long TEND:1;                               // FIN
        unsigned long HIFS:1;                               // HIFS
    } BIT;
};

union st_hifscr{                                            /* definition of HIFSCR                        */
    unsigned long LONG;                                     // Long Access
    struct {                                                // Bit Access
        unsigned long :20;                                  // reserve
        unsigned long DMD:1;                                // DREQ mode
        unsigned long DPOL:1;                               // DREQ polarity
        unsigned long BMD:1;                                // HIFRAM bunk mode
        unsigned long BSEL:1;                               // HIFRAM bunk select
        unsigned long :2;                                   // reserve
        unsigned long MD1:1;                                // HIF mode
        unsigned long :3;                                   // reserve
        unsigned long EDN:1;                                // HIFRAM endian
        unsigned long BO:1;                                 // HIF byte order
    } BIT;
};

//------------ Function Definition ----------------------------------------------------------------------
void uram_main(void);

extern void jump_sdram(void);

//-----------------------------------------------------------------------------------------------------------
#define ST_HIFGSR(*(volatile union st_hifgsr*)0xF84D0004)     /* SH7618 HIFGSR register address            */
#define ST_HIFSCR(*(volatile union st_hifscr*)0xF84D0008)     /* SH7618 HIFSCR register address            */

#define SDRAM_TOP (*(volatile unsigned long*)0xAC000000)      /* SDRAM top address,Non Cache area          */

#define HIFRAM_HEAD (*(volatile unsigned long*)0xF84E0000)    /* HIFRAM header area address                */
#define HIFRAM_BUFF_LONG (*(volatile unsigned long*)0xF84E0004)
                                                            /* HIFRAM buffer area when long-word access  */
#define HIFRAM_BUFF_BYTE (*(volatile unsigned char*)0xF84E0004)
                                                            /* HIFRAM buffer area when byte access       */

unsigned char* Sdram_address_pt_byte;                       /* pointer of SDRAM address(byte access)     */
unsigned long* Sdram_address_pt_long;                       /* pointer of SDRAM address(long word access)*/
```

```
/*****************************************************************************************************************/
/*      Main routine                                                                                             */
/*****************************************************************************************************************/
void uram_main( void )
{
    unsigned long i;
    unsigned long t_count;                          /* times of transmission when long word access            */
    unsigned char *h_buffer_pt_byte;                /* pointer of HIFRAM buffer address(byte access)          */
    unsigned long *h_buffer_pt_long;                /* pointer of HIFRAM buffer address(long word access)     */

    //----------clear in HIFRAM buffer 0 (bank0)
    h_buffer_pt_long = &HIFRAM_BUFF_LONG;

    HIFRAM_HEAD = 0;
    for(  i=0 ;  i<HIF_BUFF_SIZE_LONG ; i++ )
    {
        *h_buffer_pt_long = 0;
        h_buffer_pt_long ++;
    }

    ST_HIFSCR.BIT.BMD = 1;                          /* SH7618 and external device access the bank where HIFRAM differ*/

    //----------clear in HIFRAM buffer 1 (bank1)
    h_buffer_pt_long = &HIFRAM_BUFF_LONG;

    HIFRAM_HEAD = 0;
    for(  i=0 ;  i<HIF_BUFF_SIZE_LONG ; i++ )
    {
        *h_buffer_pt_long = 0;
        h_buffer_pt_long ++;
    }

    Sdram_address_pt_long = &SDRAM_TOP;             /* transmission destination address                       */

    ST_HIFGSR.BIT.HIFS = 1;                         /* set HIFS=1(The writing to HIFRAM is possible)          */

    while(1)
    {
        while(ST_HIFGSR.BIT.HIFS != 0);             /* wait until write end to HIFRAM from external device    */

        ST_HIFSCR.BIT.BSEL = ~ST_HIFSCR.BIT.BSEL;   /* access bank is change                                  */

        ST_HIFGSR.BIT.HIFS = 1;                     /* set HIFS=1(The writing to HIFRAM is possible)          */

        if( ST_HIFGSR.BIT.TEND == 1 )               /* transmission end(TEND=1)                               */
        {
            break;                                  /* escape from loop                                       */
        }
        else
        {
            //----------transmit to SDRAM from HIFRAM buffer
            //----------transmit by long-word size
            if( HIFRAM_HEAD == HIF_BUFF_SIZE_BYTE )
            {
                h_buffer_pt_long = &HIFRAM_BUFF_LONG;
                t_count = HIFRAM_HEAD/4;            /* times of transmission                                  */

                for( i=0 ; i<t_count ; i++ )
                {
                    *Sdram_address_pt_long = *h_buffer_pt_long;

                    Sdram_address_pt_long++;
                    h_buffer_pt_long++;
                }
            }
```

```
            //----------transmit by byte size
            else if( HIFRAM_HEAD < HIF_BUFF_SIZE_BYTE )
            {
                h_buffer_pt_byte = &HIFRAM_BUFF_BYTE;
                Sdram_address_pt_byte = (unsigned char*)Sdram_address_pt_long;

                for( i=0 ; i<HIFRAM_HEAD ; i++ )
                {
                    *Sdram_address_pt_byte = *h_buffer_pt_byte;

                    Sdram_address_pt_byte++;
                    h_buffer_pt_byte++;
                }
            }
            else
            {
                while(1);                       /* error                                    */
            }

            HIFRAM_HEAD = 0;                    /* clear the HIFRAM header                  */
        }
    }

    //----------Execution of the transmitted program
    jump_sdram();
}
```

2. SH7618 プログラム 2

```
;*************************************************************************************************************
;   function : jump_sdram
;   operation    : Execution of the transmitted program
;   CUP      : SH7618
;   date     : 2004.4
;*************************************************************************************************************
        .EXPORT  _jump_sdram
;
; MS7618 CS3 area SDRAM address,user program start address
SDRAM_START: .equ     H'AC000000
        .SECTION UM_P, CODE, ALIGN=4
;*************************************************************************************************************
;
_jump_sdram:
;
    mov.l    #SDRAM_START,R10            ;Program Start
    jmp @R10
    nop
;
    .END
```

3. SH7641 プログラム

```
/*****************************************************************************************************************/
// SH7618 HIF boot mode application note
//  Program transmission which used two bank of HIFRAM
//      CPU:      SH7641,SH-3 DSP,Big Endian
//      Clock:    External input = 12.5MHz
//                CPU clock = 100MHz
//                External BUS clock = 50MHz
//                Peripheral clock = 25MHz
//      Written:  '04/4 Rev.2.0
/*****************************************************************************************************************/

#include "7641.h"

/*****************************************************************************************************************/
/*  Protocol declaration of the function                                                                         */
/*****************************************************************************************************************/
/*------------ Symbol Definition ---------------------------------------------------------------------------------*/
#define TRANS_STRAGE_ADDR 0xA5602000                    // storing address of a sdram program
#define TRANS_P_SIZE      0x200000                      // sdram program size(Byte)

#define HIF_BUFF_SIZE     0x3FC                         // HIFRAM buffer size(192Byte)
#define HIF_HEAD_ADDR     0x0000                        // HIFRAM header address
#define HIF_BUFF_ADDR     0x0004                        // HIFRAM buffer address

    //----------The value when specifying the register of HIF
#define SEL_HIFMCR_LO     0x000A                        // HIFMCR[15:0]
#define SEL_HIFBCR_LO     0x003E                        // HIFBCR[15:0]
#define SEL_HIFADR_LO     0x0016                        // HIFADR[15:0]
#define SEL_HIFDATA_UP    0x0018                        // HIFDATA[31:16]
#define SEL_HIFGSR_LO     0x0002                        // HIFGSR[15:0]

/*------------ Function Definition -------------------------------------------------------------------------------*/
void main(void);

unsigned short* write_HIFRAM(unsigned short* , unsigned short , unsigned short );

void sync_7618(unsigned short* , unsigned short , unsigned short , unsigned long , unsigned short );

/*-----------------------------------------------------------------------------------------------------------------*/
    //----------The address when accessing the register of HIF
// select of the register of HIF
#define HIF_REG_SEL       (*(volatile unsigned short*)0xB4001000)
// data write to the register of HIF
#define HIF_DATA_WR       (*(volatile unsigned short*)0xB4000000)
// data read from the HIFGSR register
#define HIF_GSR_RD        (*(volatile unsigned short*)0xB4001004)

/*****************************************************************************************************************/
/*      Main routine                                                                                             */
/*****************************************************************************************************************/
void main(void)
{
    //----------set of bus interface
    BSC.CS5ABCR.BIT.BSZ = 2;
    BSC.CS5AWCR = 0x00000901;

    //----------set as a CS5A function
    PFC.PCCR.BIT.PC1MD = 3;                             /* bit[2-3]-PC1MD=b'11 : PC1=>CS5A                */

    //----------synchronization is taken SH7618, and a program is written to HIFRAM
    sync_7618((unsigned short*)TRANS_P_STRAGE_ADDR , HIF_HEAD_ADDR ,
                    HIF_BUFF_ADDR , TRANS_P_SIZE,HIF_BUFF_SIZE);

    //----------Loop
    while(1);                                           /* Loop                                           */
}
```

```
/*****************************************************************************************************************/
//  function:   write_HIFRAM
//  operation:   boot program writing to HIFRAM
//  argument:   trans_src_addr ; storing address of a boot program
//              hif_addr      ; head address of HIFRAM which transmits a boot program
//              t_size        ; transmit program size
//  return:     trans_src_addr ; storing address of a boot program
/*****************************************************************************************************************/
unsigned short* write_HIFRAM(unsigned short *trans_src_addr , unsigned short hif_addr , unsigned short t_size)
{
    time = t_size/2 + t_size%2;                         /* calculate times of transmission              */

    HIF_REG_SEL = SEL_HIFADR_LO;                        /* select HIFADR register                       */
    HIF_DATA_WR = hif_addr;                             /* set of HIFRAM address                        */

    HIF_REG_SEL = SEL_HIFDATA_UP;                       /* select HIFDATA register[31:16]               */
    HIF_DATA_WR = (*trans_src_addr);                    /* set to HIFDATA register[31:16]               */

    trans_src_addr ++;                                  /* storing address is increment(+2)             */

    HIF_DATA_WR = (*trans_src_addr);                    /* set to HIFDATA register[15:0]                */

    trans_src_addr ++;                                  /* storing address is increment(+2)             */

    HIF_REG_SEL = SEL_HIFMCR_LO;                        /* select HIFMCR register[15:0]                 */
    HIF_DATA_WR = 0x00A0;                               /* set as continuation write mode               */

    HIF_REG_SEL = SEL_HIFDATA_UP;                       /* select HIFDATA register[31:16]               */

    for( i=2 ; i<time ; i++){
        HIF_DATA_WR = (*trans_src_addr);                /* write to HIFDATA register(HIFRAM)            */

        trans_src_addr ++;                              /* storing address is increment(+2)             */
    }

    return(trans_src_addr);
}

/*****************************************************************************************************************/
//  function:   sync_7618
//  operation:  synchronization is taken SH7618, and a program is written to HIFRAM
//  argument:   *s_addr ; pointer of source address
//              h_addr ; HIFRAM header address
//              b_addr ; HIFRAM buffer address
//              t_size ; transmission data size
//              b_size ; HIFRAM buffer size
//  argument:   non-argument
/*****************************************************************************************************************/
void sync_7618(unsigned short* s_addr , unsigned short h_addr , unsigned short b_addr , unsigned long t_size , unsigned
short b_size)
{
    volatile unsigned short status;

    while(1){
        status = HIF_GSR_RD;                            /* read from HIFGSR register[15:0]              */

        while(status != 0x0001){                        /* wait until HIFGSR.HIFS=1                     */
            status = HIF_GSR_RD;                        /* read from HIFGSR register[15:0]              */
        }

    //----------preparation write to header
        HIF_REG_SEL = SEL_HIFADR_LO;                    /* select HIFADR register[15:0]                 */
        HIF_DATA_WR = h_addr;                           /* set of HIFRAM header address                 */
        HIF_REG_SEL = SEL_HIFDATA_UP;                   /* select HIFDATA register[31:16]               */
        HIF_DATA_WR = 0;                                /* set to HIFDATA register[31:16]               */
```

```
    //----------usual transmission
        if(t_size > b_size)
        {
    //----------transmission data size is written to header
            HIF_DATA_WR = b_size;                       /* write to HIFDATA register[15:0]              */

    //----------write to HIFRAM buffer
            s_addr = write_HIFRAM(s_addr,b_addr,b_size);

    //----------transmission end process
            HIF_REG_SEL = SEL_HIFGSR_LO;                /* select HIFGSR register[15:0]                 */
            HIF_DATA_WR = 0x0000;                       /* set to HIFGSR register[15:0]                 */
                                                        // TEND=0 , HIFS=0
        }

    //----------last transmission
        else
        {
    //----------transmission data size is written to header
            HIF_DATA_WR = (unsigned short)t_size;       /* write to HIFDATA register[15:0]              */

    //----------write to HIFRAM buffer
            s_addr = write_HIFRAM(s_addr,b_addr,b_size);

    //----------transmission end process
            HIF_REG_SEL = SEL_HIFGSR_LO;                /* select HIFGSR register[15:0]                 */
            HIF_DATA_WR = 0x0000;                       /* set to HIFGSR register[15:0]                 */
                                                        // TEND=0 , HIFS=0

            break;                                      /* escape from loop                             */
         }
    //----------calculate transmission data size
        t_size = t_size - b_size;
    }

    status = HIF_GSR_RD;                                /* read from HIFGSR register[15:0]              */

    while( status==0x0000 )                             /* wait until HIFGSR.HIFS=1                     */
    {
        status = HIF_GSR_RD;                            /* read from HIFGSR register[15:0]              */
    }

    HIF_DATA_WR = 0x0002;                               /* set to HIFGSR register[15:0]                 */
                                                        // TEND=1 , HIFS=0
}
```

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2005.09.14	—	初版発行

安全設計に関するお願い

1. 弊社は品質、信頼性の向上に努めておりますが、半導体製品は故障が発生したり、誤動作する場合があります。弊社の半導体製品の故障又は誤動作によって結果として、人身事故、火災事故、社会的損害などを生じさせないような安全性を考慮した冗長設計、延焼対策設計、誤動作防止設計などの安全設計に十分ご留意ください。

本資料ご利用に際しての留意事項

1. 本資料は、お客様が用途に応じた適切なルネサス テクノロジ製品をご購入いただくための参考資料であり、本資料中に記載の技術情報についてルネサス テクノロジが所有する知的財産権その他の権利の実施、使用を許諾するものではありません。
2. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例の使用に起因する損害、第三者所有の権利に対する侵害に関し、ルネサス テクノロジは責任を負いません。
3. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他全ての情報は本資料発行時点のものであり、ルネサス テクノロジは、予告なしに、本資料に記載した製品または仕様を変更することがあります。ルネサス テクノロジ半導体製品のご購入に当たりましては、事前にルネサス テクノロジ、ルネサス販売または特約店へ最新の情報をご確認頂きますとともに、ルネサス テクノロジホームページ(http://www.renesas.com)などを通じて公開される情報に常にご注意ください。
4. 本資料に記載した情報は、正確を期すため、慎重に制作したものですが万一本資料の記述誤りに起因する損害がお客様に生じた場合には、ルネサス テクノロジはその責任を負いません。
5. 本資料に記載の製品データ、図、表に示す技術的な内容、プログラム及びアルゴリズムを流用する場合は、技術内容、プログラム、アルゴリズム単位で評価するだけでなく、システム全体で十分に評価し、お客様の責任において適用可否を判断してください。ルネサス テクノロジは、適用可否に対する責任は負いません。
6. 本資料に記載された製品は、人命にかかわるような状況の下で使用される機器あるいはシステムに用いられることを目的として設計、製造されたものではありません。本資料に記載の製品を運輸、移動体用、医療用、航空宇宙用、原子力制御用、海底中継用機器あるいはシステムなど、特殊用途へのご利用をご検討の際には、ルネサス テクノロジ、ルネサス販売または特約店へご照会ください。
7. 本資料の転載、複製については、文書によるルネサス テクノロジの事前の承諾が必要です。
8. 本資料に関し詳細についてのお問い合わせ、その他お気付きの点がございましたらルネサス テクノロジ、ルネサス販売または特約店までご照会ください。