
V850E2/ML4

R01AN1222JJ0100

Rev.1.00

2012.06.22

CSIH 制御編

要旨

本アプリケーションノートでは、V850E2/ML4のクロック同期式3線・シリアル・インターフェースCSIHの設定方法、およびサンプルコードの動作概要や使用方法を説明します。

本アプリケーションノートでは次の4つのモードでのCSIHの使用例を示します。

- マスタ、デュアル・バッファ、送信オンリー・バッファ・モード
- スレーブ、デュアル・バッファ、受信モード
- マスタ、ダイレクト・アクセス、送信オンリー・バッファ・モード
- スレーブ、ダイレクト・アクセス、受信モード

対象デバイス

V850E2/ML4

開発環境

CubeSuite+, GHS MULTI V5.1.7D、IAR for V850 Kickstart V3.80

本アプリケーションノートを他のマイコンへ適用する場合、そのマイコンの仕様にあわせて変更し、十分評価してください。

目次

1. 仕様.....	3
2. 動作確認条件.....	5
3. ハードウェア説明.....	6
3.1 ハードウェア構成例	6
3.2 使用端子一覧.....	7
4. ソフトウェア説明.....	8
4.1 動作概要.....	8
4.2 必要メモリサイズ.....	9
4.3 ファイル構成.....	10
4.4 オプション設定メモリ	11
4.5 変数一覧.....	12
4.6 関数一覧.....	13
4.7 関数仕様.....	14
4.8 フローチャート	19
4.8.1 メイン処理	19
4.8.2 マスタ、ダイレクト・アクセス、送信オンリー・バッファ・モード	20
4.8.3 スレーブ、ダイレクト・アクセス、受信モード	21
4.8.4 マスタ、デュアル・バッファ、送信オンリー・バッファ・モード.....	22
4.8.5 スレーブ、デュアル・バッファ、受信モード	23
4.8.6 割り込み処理.....	24
5. サンプルコード	28
6. 参考ドキュメント.....	28

1. 仕様

このサンプルコードでは、CSIH の次の 4 つの動作モードの使用があります。

- マスタ、デュアル・バッファ、送信オンリー・バッファ・モード
- スレーブ、デュアル・バッファ、受信モード
- マスタ、ダイレクト・アクセス、送信オンリー・バッファ・モード
- スレーブ、ダイレクト・アクセス、受信モード

マスタ・モードでは、内蔵のポー・レート・ジェネレータ (BRG) によってシリアル通信クロックが生成され、CSIHnTSCCK 信号を介して通信クロックを出力します。スレーブ・モードでは、ほかのデバイスが通信マスタになり、通信クロックを入力します。

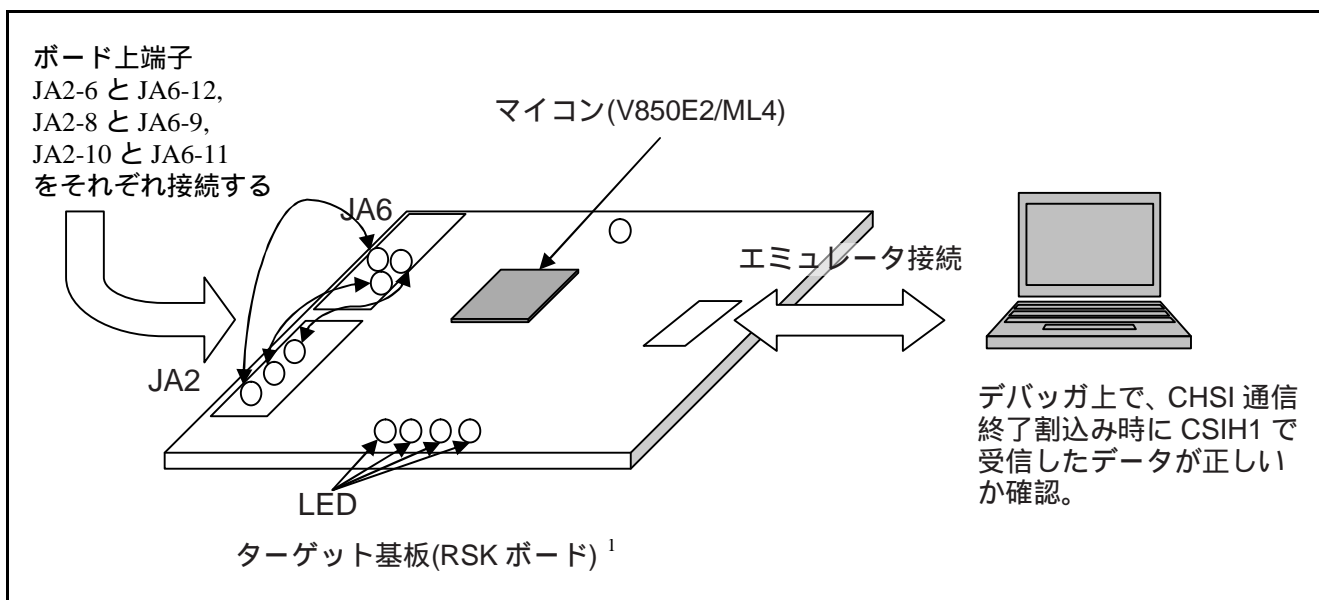
通信基本仕様は以下のとおりです。

メモリ・モード	ダイレクト・アクセス	デュアル・バッファ
受信用 I/F	CSIH1	
送信用 I/F	CSIH0	
転送方向	MSB ファースト	
送受信時のパリティ	パリティ無し	
データ長	8 ビット	
ポー・レート	64kbps	
FIFO 容量	FIFO 無し	64 バイト (送信 / 受信毎)
通信データ数	6 バイト	9 バイト
JOB(ジョブ数)	2JOB	JOB 無し
EDL (拡張データ長)	無し	
LBM (ループ・バック・モード)	無し	
SS (スレーブ選択)	無し	

表 1.1 に使用する周辺機能と用途を、図 1.1 に使用例を示します。

表1.1 使用する周辺機能と用途

周辺機能	用途
ポート(P1_4, P1_5, P4_3, P4_4)	LED に接続し、LED の点灯を制御
CSIH0 - SO0F	CSIH0 出力端子 (JA6-9)
CSIH0 - SI0F	CSIH0 入力端子 (JA6-12)
CSIH0 - SCK0F	CSIH0 クロック(入力) 端子 (JA6-11)
CSIH1 - SO1F	CSIH1 出力端子 (JA2-6)
CSIH1 - SI1F	CSIH1 入力端子 (JA2-8)
CSIH1 - SCK1F	CSIH1 クロック(出力)端子 (JA2-10)

図1.1 使用例¹

¹ RSK ボードは 2012 年 8 月量産予定

2. 動作確認条件

本アプリケーションノートのサンプルコードは、下記の条件で動作を確認しています。

表2.1 動作確認条件

項目	内容
使用マイコン	V850E2/ML4
動作周波数	200MHz(発振 10MHz × PLL 20 逡倍)
動作電圧	3.3V
統合開発環境	CubeSuite+ V1.00
	GHS MULTI V5.1.7D
	IAR for V850 Kickstart V3.80.1
C コンパイラ	CX V1.20(CubeSuite+)、最適化：デフォルト
	C-V850E 5.1.7 RELEASE(GHS MULTI)、最適化：デフォルト
	IAR C/C++ Compiler for V850 3.80.1 [Kickstart] (3.80.1.30078) 、最適化：デフォルト
動作モード	通常動作モード
サンプルコードのバージョン	V1.00
使用ボード	RSK ボード
使用デバイス	E1 エミュレータもしくは MINICUBE
使用ツール	なし

3. ハードウェア説明

3.1 ハードウェア構成例

図 3.1に RSK ボード上での接続例を示します。

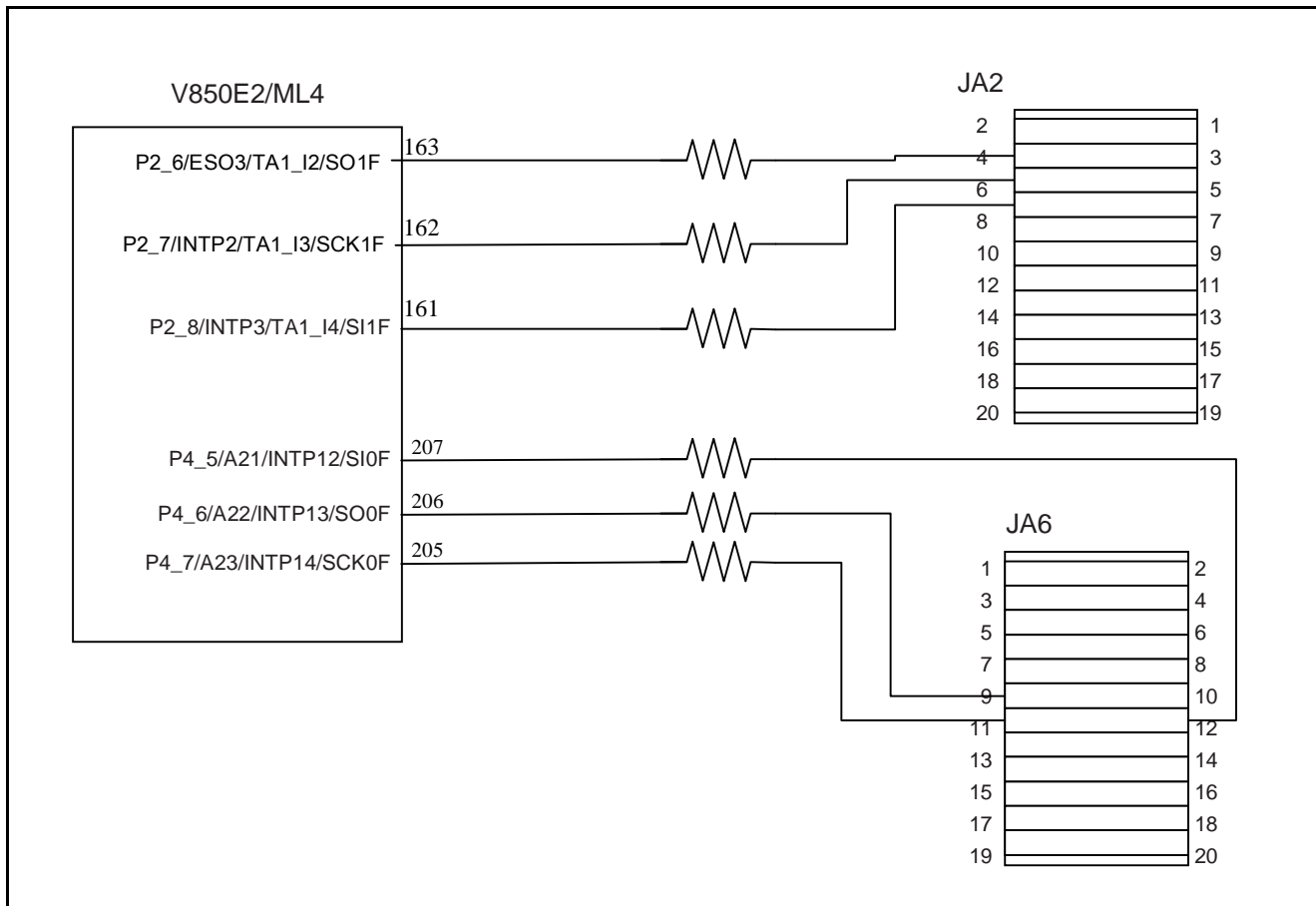


図3.1 RSK ボード上での接続例

3.2 使用端子一覧

表 3.1に使用端子と機能を示します。

表3.1 使用端子と機能

端子名	入出力	内容
PORT P1_4	出力	ポート・モード、出力、LED0
PORT P1_5	出力	ポート・モード、出力、LED1
PORT P4_3	出力	ポート・モード、出力、LED2
PORT P4_4	出力	ポート・モード、出力、LED3
SO0F	出力	CSH0 のシリアル送信データ出力
SI0F	入力	CSH0 のシリアル送信データ入力
SCK0F	出力	CSH0 のシリアル・クロック入出力(マスタのため出力)
SO1F	出力	CSH1 のシリアル送信データ出力
SI1F	入力	CSH1 のシリアル送信データ入力
SCK1F	入力	CSH1 のシリアル・クロック入出力(スレーブのため入力)

4. ソフトウェア説明

4.1 動作概要

ソフトウェアの動作の概要を次の図に示します。main()で各種初期化関数を呼び、割り込み待ち状態に入ります。受信割り込みが発生すると、割り込み処理関数で、受信データ結果の格納、CSIHの再開を実行します。

図 4.1にシーケンス(主要部分)を示します。

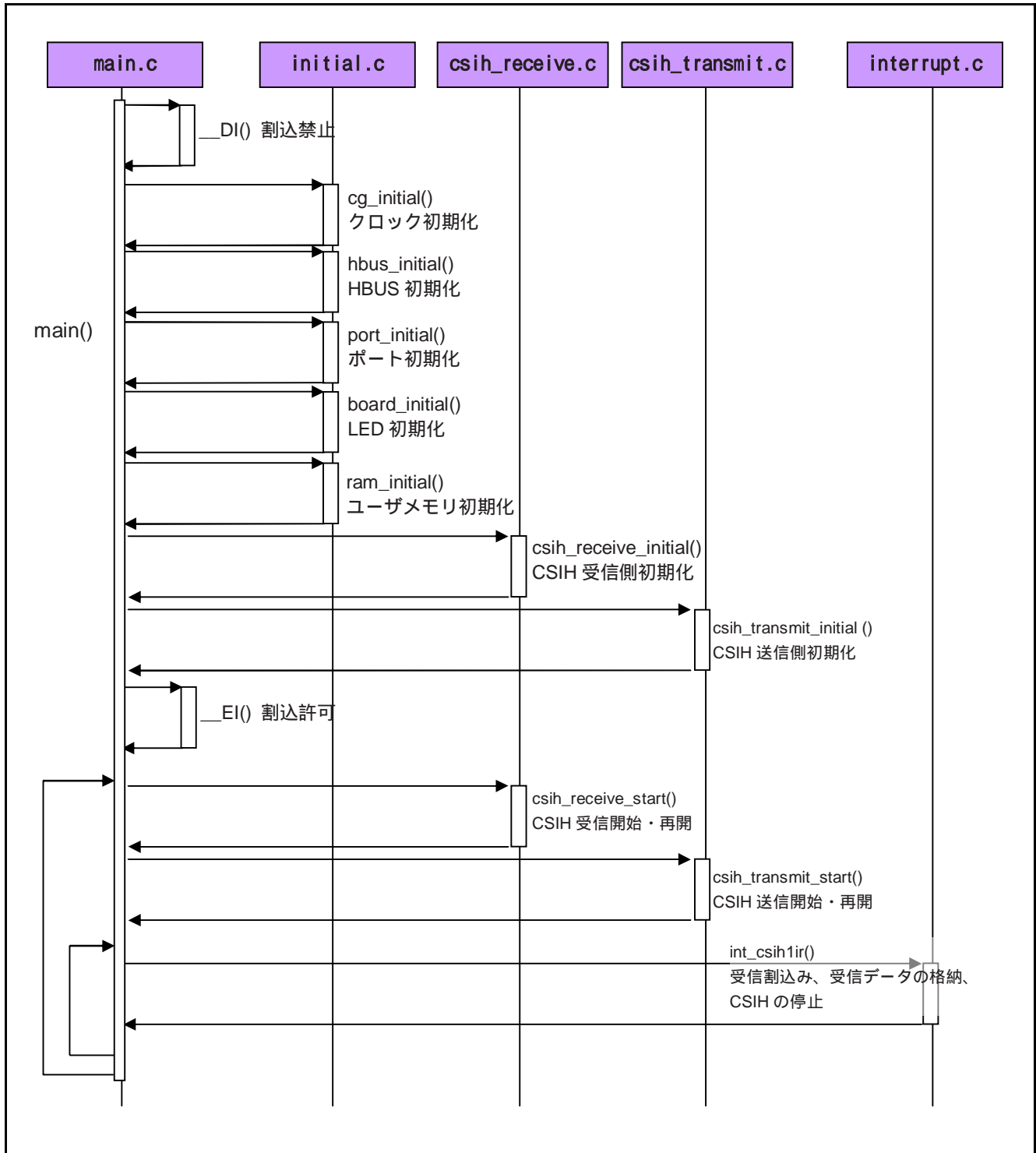


図4.1 シーケンス

4.2 必要メモリサイズ

表 4.1に必要メモリサイズを示します。(CubeSuite+、最適化オプション = デフォルトで測定)

表4.1 必要メモリサイズ

使用メモリ	サイズ	備考
ROM	6696	CubeSuite+の生成する map ファイルに出力された ROM 領域で使用するサイズ
RAM	4204	CubeSuite+の生成する map ファイルに出力された RAM 領域で使用するサイズ
最大使用ユーザスタック	12	CubeSuite+のスタック見積もりツールで算出
最大使用割り込みスタック	204	同上

【注】 必要メモリサイズはCコンパイラのバージョンやコンパイルオプションにより異なります。

4.3 ファイル構成

表 4.2にサンプルコードで使用するファイルを示します。なお、統合開発環境で自動生成されるファイルは除きます。

表4.2 サンプルコードで使用するファイル

ファイル名	概要	備考
crtE.s	ハードウェア初期化处理	CubeSuite+でのみ使用
startup.s		GHS MULTI でのみ使用
V850E2ML4.dir	リンク・ディレクティブ・ファイル	CubeSuite+でのみ使用
V850E2 ML4 CSIH.ld		GHS MULTI でのみ使用
vector.s	ベクタ・テーブル	GHS MULTI でのみ使用
csih.h	変数、関数宣言	
df4022_800.h	V850E2/ML4 用レジスタマクロ宣言	GHS MULTI でのみ使用
V850E2ML4PortRegister.h	portconfig.c 用ヘッダファイル	
main.c	メイン処理	
initial.c	ソフトウェア初期化处理	
csih_transmit.c	送信処理	
csih_receive.c	受信処理	
interrupt.c	割り込み処理	
portconfig.c	ポート初期化	

4.4 オプション設定メモリ

本サンプルでは、オプション・バイトの設定は行っていません。必要に応じて設定してください。

4.5 変数一覧

表 4.3にグローバル変数を示します。

表4.3 グローバル変数

型	変数名	内容	使用関数
unsigned char	flag_mode;	転送モードのフラグ	csih_receive_initial() csih_receive_start() csih_transmit_initial() csih_transmit_start() int_csih0ic() int_csih0ijc() int_csih1ir() main()
unsigned int	flag_error;	CSIHのエラーフラグ	main() ram_initial() display() int_csih0ire() int_csih1ire()
unsigned char	count;	受信割り込みのカウンタ	ram_initial() int_csih1ire() int_csih1ir()
unsigned char	flag_transmit_over;	転送終了フラグ	main() int_csih0ic() int_csih0ijc() ram_initial() display()
unsigned char	flag_receive_over;	受信終了フラグ	ram_initial() int_csih1ir() csih_receive_start() display()
unsigned char	flag_job_transmit;	送信ジョブ終了フラグ	csih_transmit_1_start() ram_initial() int_csih1ir()
unsigned int	buf_receive[NUM];	受信バッファ	ram_initial() clear_receive_buffer() int_csih0ire()
unsigned int *	point_receive;	受信用ポインタ	ram_initial() clear_receive_buffer() int_csih0ire() int_csih1ire() int_csih1ir()
unsigned int *	point_transmit;	送信用ポインタ	ram_initial() int_csih0ire() int_csih1ire()
unsigned int	buf_transmit[NUM];	送信用バッファ	csih_transmit_1_start() csih_transmit_2_start()

4.6 関数一覧

表 4.4に関数を示します。

表4.4 関数

関数名	概要
void port_initial(void)	ポート・モードの設定を行う
void PortConfigulation1(void)	ポートグループ 1 の設定
void PortConfigulation2(void)	ポートグループ 2 の設定
void PortConfigulation4(void)	ポートグループ 4 の設定
void cg_initial(void)	特殊クロック周波数制御レジスタの初期化設定を行う
void hbus_initial(void)	AHB バスの初期化設定を行う
void board_initial(void)	LED 初期状態の設定を行う
void ram_initial(void)	ユーザ RAM の初期状態の設定を行う
void display(void)	LED に受信、送信、エラーの状況を表示する
void wait(int number)	指定時間処理を待つ
void clear_receive_buffer(void)	受信バッファのクリア
void main(void)	各初期化処理関数を呼び出したあと、永久ループに入る
void csih_receive_initial(void)	CSIH 受信側の初期化
void csih_transmit_initial(void)	CSIH 送信側の初期化
void csih_transmit_start(void)	CSIH 送信開始
void csih_receive_start(void)	CSIH 受信開始
interrupt void int_csih0ire(void)	CSIH0 エラー受信割込み
interrupt void int_csih0ic(void)	CSIH0 送信完了割込み
interrupt void int_csih0ijc(void)	CSIH0 送信ジョブ完了割込み
interrupt void int_csih1ire(void)	CSIH1 エラー受信割込み
interrupt void int_csih1ir(void)	CSIH1 受信割込み
void csih_receive_initial(void)	CSIH 受信側初期化
void csih_receive_1_initial(void)	ダイレクトアクセスモードによる受信側初期化
void csih_receive_2_initial(void)	デュアルバッファモードによる受信側初期化
void csih_transmit_initial(void)	CSIH 送信側初期化
void csih_transmit_1_initial(void)	ダイレクトアクセスモードによる送信側初期化
void csih_transmit_2_initial(void)	デュアルバッファモードによる送信側初期化
void csih_transmit_1_start(void)	ダイレクトアクセスモードによる送信開始
void csih_transmit_2_start(void)	デュアルバッファモードによる送信開始

4.7 関数仕様

サンプルコードの関数仕様を示します。

main()	
概要	メイン関数、最初に呼び出される関数
ヘッダ	-
宣言	void main(void)
説明	各初期化処理関数を呼び出したあと、永久ループに入り、CSIH 割り込みを待つ
引数	-
リターン値	-

port_initial()	
概要	ポート・モードの設定を行う
ヘッダ	csih.h
宣言	void port_initial (void)
説明	LED、CSIH で使用するポート機能の初期化を行う。
引数	-
リターン値	-

PortConfiguration0() ~ PortConfiguration8()	
概要	ポート・グループごとのポート・モードの設定を行う
ヘッダ	V850E2ML4PortRegister.h
宣言	void PortConfiguration0(void)...void PortConfiguration8(void)
説明	port_initial()より呼ばれ、ポートグループごとの設定を行う。実際に呼ばれるのは、PortConfiguration1(),PortConfiguration2(), PortConfiguration4()のみ。
引数	-
リターン値	-

cg_initial ()	
概要	クロック初期化
ヘッダ	csih.h
宣言	void cg_initial(void)
説明	特殊クロック周波数制御レジスタの初期化設定を行う
引数	-
リターン値	-

hbus_initial()	
概要	Hバス初期化
ヘッダ	csih.h
宣言	void hbus_initial(void)
説明	AHB バスの初期化設定を行う
引数	-
リターン値	-

board_initial()	
概要	ボード初期化
ヘッダ	csih.h
宣言	void board_initial(void)
説明	LED 初期状態の設定を行う
引数	-
リターン値	-

ram_initial()	
概要	ユーザメモリ初期化
ヘッダ	csih.h
宣言	void ram_initial(void)
説明	ユーザ RAM の初期状態の設定を行う
引数	-
リターン値	-

display()	
概要	LED に状況表示
ヘッダ	csih.h
宣言	void display(void)
説明	LED に受信、送信、エラーの状況を表示する
引数	-
リターン値	-

wait ()	
概要	指定時間処理を待つ
ヘッダ	csih.h
宣言	void wait(int number)
説明	CSIH の送信時処理待ちを行う。
引数	int number 待ち時間
リターン値	-

clear_receive_buffer()	
概要	受信バッファのクリア
ヘッダ	csih.h
宣言	void clear_receive_buffer(void)
説明	受信前にバッファを 0 クリアする。
引数	-
リターン値	-

csih_receive_initial()	
概要	CSIH 受信側の初期化
ヘッダ	csih.h

宣言	void csih_receive_initial(void)
説明	受信側初期化、モード flag_mode に応じてサブルーチンを呼ぶ。
引数	-
リターン値	-

csih_receive_1_initial()

概要	CSIH 送信側の初期化 1
ヘッダ	csih.h
宣言	csih_receive_1_initial(void)
説明	ダイレクトアクセスモードによる受信側初期化
引数	-
リターン値	-

csih_receive_2_initial()

概要	CSIH 送信側の初期化 2
ヘッダ	csih.h
宣言	csih_receive_2_initial(void)
説明	デュアルバッファモードによる受信側初期化
引数	-
リターン値	-

csih_transmit_initial()

概要	CSIH 送信側の初期化
ヘッダ	csih.h
宣言	void csih_transmit_initial(void)
説明	送信側初期化、モード flag_mode に応じてサブルーチンを呼ぶ。
引数	-
リターン値	-

csih_transmit_1_initial()

概要	CSIH 送信側の初期化 1
ヘッダ	csih.h
宣言	csih_transmit_1_initial(void)
説明	ダイレクトアクセスモードによる送信側初期化
引数	-
リターン値	-

csih_transmit_2_initial()

概要	CSIH 送信側の初期化 2
ヘッダ	csih.h
宣言	csih_transmit_2_initial(void)
説明	デュアルバッファモードによる送信側初期化
引数	-
リターン値	-

csih_transmit_start()	
概要	CSIH 送信開始
ヘッダ	csih.h
宣言	void csih_transmit_start(void)
説明	送信開始、モード flag_mode に応じてサブルーチンを呼ぶ。
引数	-
リターン値	-
csih_receive_start()	
概要	CSIH 受信開始
ヘッダ	csih.h
宣言	void csih_receive_start(void)
説明	受信開始、モード flag_mode に応じてサブルーチンを呼ぶ。
引数	-
リターン値	-
csih_transmit_1_start()	
概要	CSIH 送信開始 1
ヘッダ	csih.h
宣言	csih_transmit_1_start (void)
説明	ダイレクトアクセスモードによる送信開始
引数	-
リターン値	-
csih_transmit_2_start ()	
概要	CSIH 送信開始 2
ヘッダ	csih.h
宣言	csih_transmit_2_start(void)
説明	デュアルバッファモードによる送信開始
引数	-
リターン値	-
int_csih0ire()	
概要	CSIH0 エラー受信割込み
ヘッダ	-
宣言	__interrupt void int_csih0ire(void)
説明	エラーフラグを flag_error に格納した後、クリア。CSIH0 を停止し、バッファポイントを初期化する。
引数	-
リターン値	-
int_csih0ic()	
概要	CSIH0 転送完了割込み

ヘッダ	-
宣言	__interrupt void int_csih0ic(void)
説明	ダイレクトアクセスモードの場合、処理待ち。 デュアルバッファモードの場合、転送完了フラグを立て、ステータスをクリアし、CSIH を停止する。
引数	-
リターン値	-

int_csih0ijc()

概要	CSIH0 転送ジョブ完了割込み
ヘッダ	-
宣言	__interrupt void int_csih0ijc(void)
説明	ダイレクトアクセスモードの場合、転送完了フラグを立て、ステータスをクリアし、CSIH を停止する。 デュアルバッファモードの場合、何もしない。
引数	-
リターン値	-

int_csih1ire()

概要	CSIH1 エラー受信割込み
ヘッダ	-
宣言	__interrupt void int_csih1ire(void)
説明	エラーフラグを flag_error に格納した後、クリア。CSIH1 を停止し、バッファポインタを初期化する。
引数	-
リターン値	-

int_csih1ire()

概要	CSIH1 受信完了割込み
ヘッダ	-
宣言	__interrupt void int_csih1ire(void)
説明	ダイレクトアクセスモードの場合：CSIH1RX0H レジスタより受信データを 1 つ取得し、ポインタを 1 つ進める。割り込みが複数回発生し、受信データ数が一定数(6)になったら、受信完了とし、フラグをクリアし、CSIH1 を停止する。 デュアルバッファモードの場合：CSIH1RX0H レジスタより受信データを一定数(6)取得する。ポインタもそのつど進める。受信完了とし、フラグをクリアし、CSIH1 を停止する。
引数	-
リターン値	-

4.8 フローチャート

4.8.1 メイン処理

図 4.2にメイン処理のフローチャートを示します。

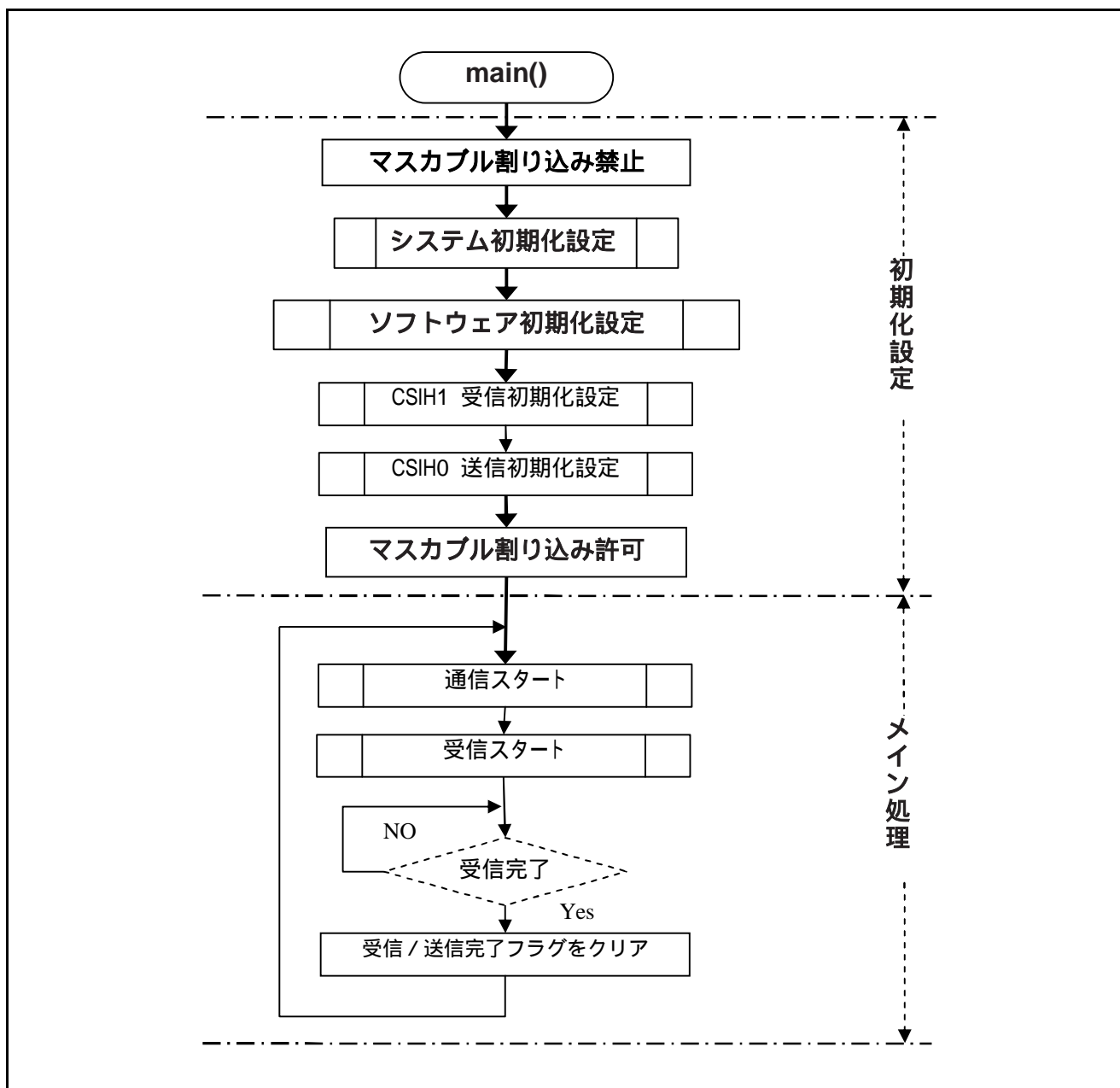


図4.2 メイン処理

4.8.2 マスタ、ダイレクト・アクセス、送信オンリー・バッファ・モード

マスタ、ダイレクト・アクセス、送信オンリー・バッファ・モードは、CSIHnTX0W または CSIHnTX0H レジスタに送信データを書き込むことで、送信を開始します。48 ビット（6 バイト）の送信データを 2 つのジョブに分けて、送信動作を行います、通信データ長は 8 ビットです。

図 4.3 にマスタ、ダイレクト・アクセス、送信オンリー・バッファ・モードのフローチャートのフローチャートを示します。

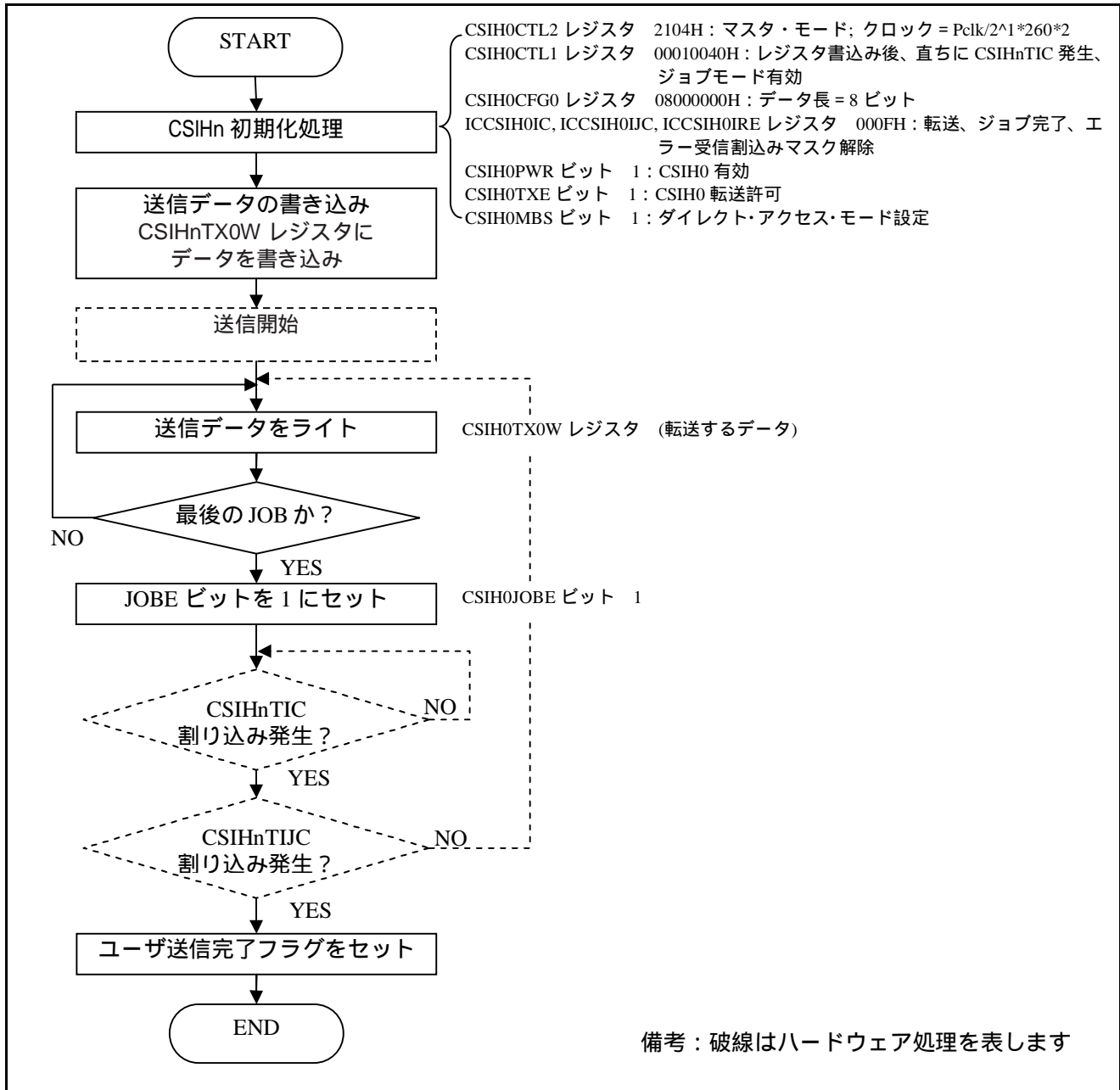


図4.3 マスタ、ダイレクト・アクセス、送信オンリー・バッファ・モードのフローチャート

4.8.3 スレープ、ダイレクト・アクセス、受信モード

スレープ、ダイレクト・アクセス、受信モードは、外部クロック CSIHTSCK を検出することで受信を開始します。

図 4.4 にスレープ、ダイレクト・アクセス、受信モードのフローチャートを示します。

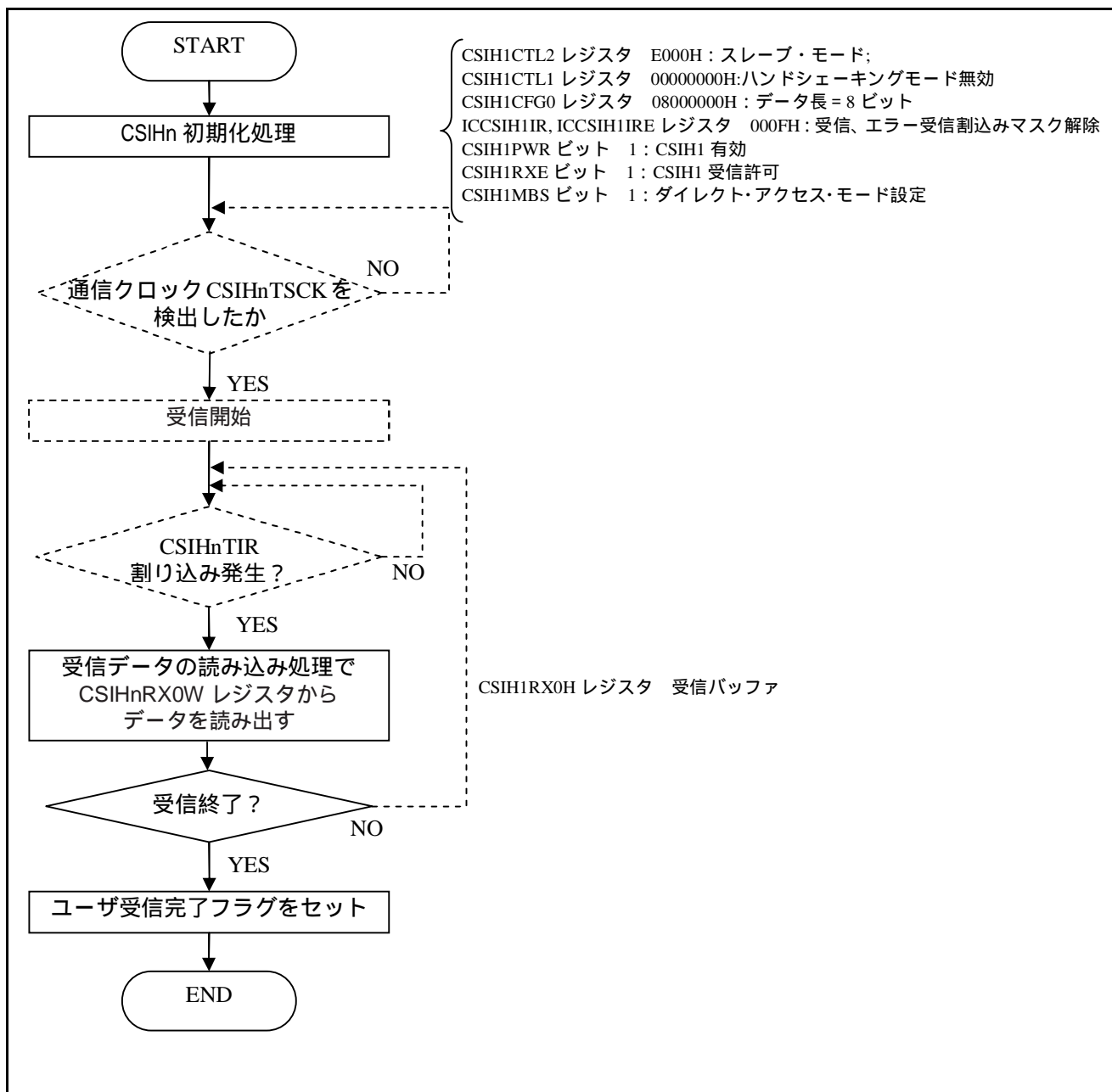


図4.4 スレープ、ダイレクト・アクセス、受信モード

4.8.4 マスタ、デュアル・バッファ、送信オンリー・バッファ・モード

マスタ、ダイレクト・アクセス、送信オンリー・バッファ・モードは、CSIHnTX0W または CSIHnTX0H レ

マスタ、デュアル・バッファ、送信オンリー・バッファ・モードは、CSIHnMCTL2.CSIHnBTST ビットを 1 にセットすることで、送信を開始します。

図 4.5 にマスタ、デュアル・バッファ、送信オンリー・バッファ・モードのフローチャートを示します。

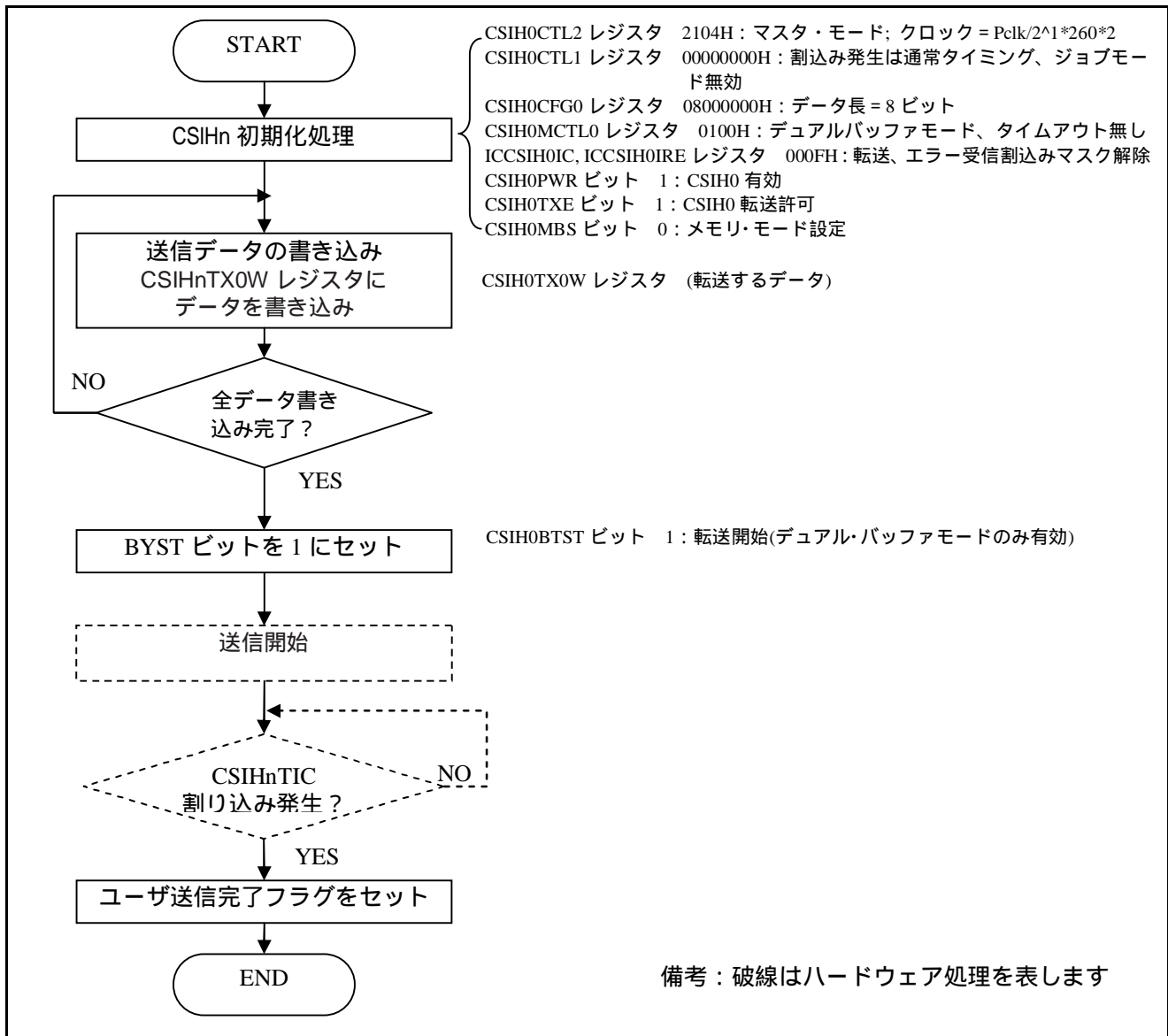


図4.5 マスタ、デュアル・バッファ、送信オンリー・バッファ・モードのフローチャート

4.8.5 スレープ、デュアル・バッファ、受信モード

スレープ、デュアル・バッファ、受信モードは、CSIHnMCTL2.CSIHnBTST ビットを 1 にセットした後、外部クロック CSIHnTSCK を検出することで受信を開始します。

図 4.6 にスレープ、デュアル・バッファ、受信モードのフローチャートを示します。

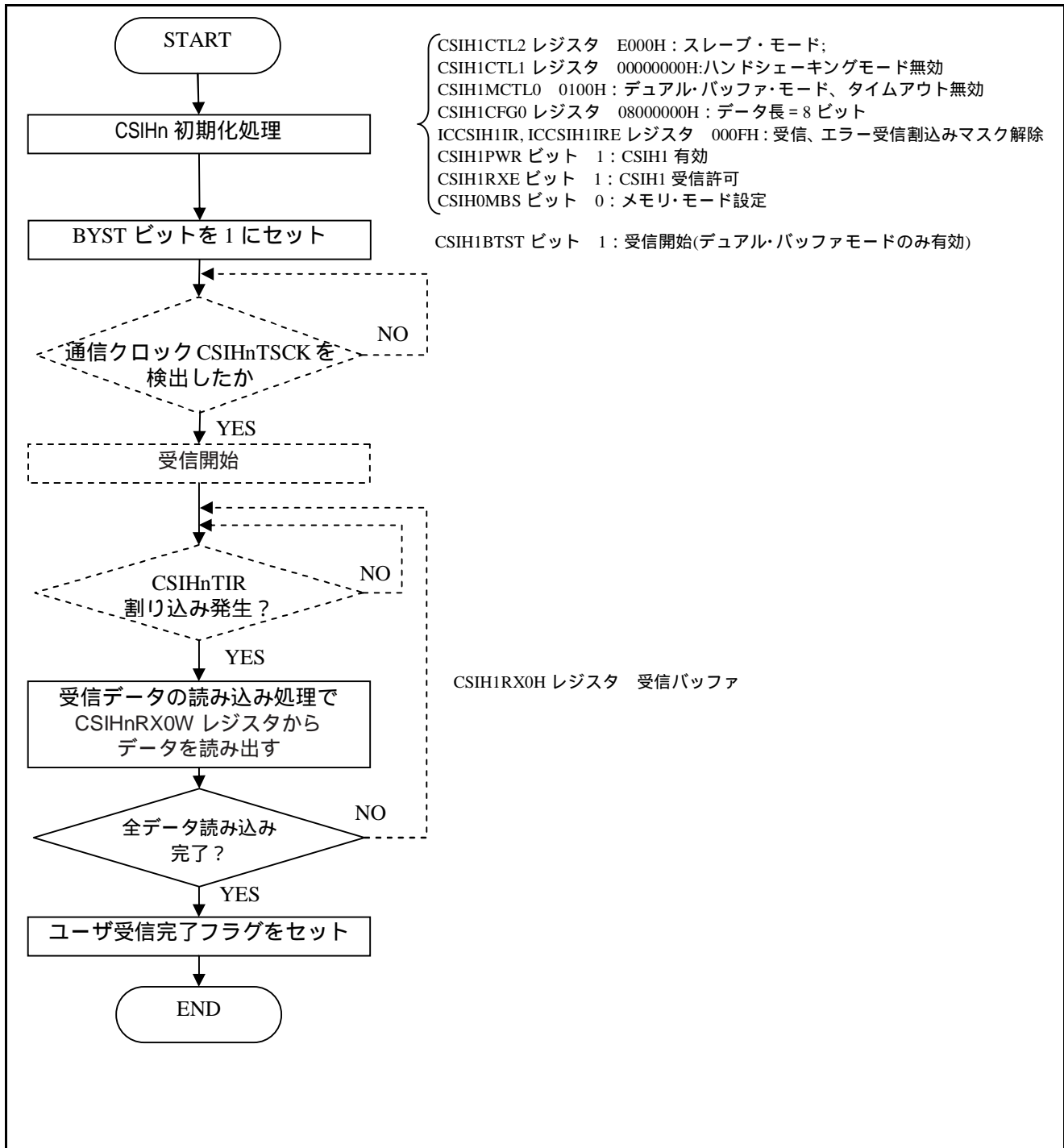


図4.6 スレープ、デュアル・バッファ、受信モードのフローチャート

4.8.6 割り込み処理

図 4.7に転送エラー割り込み処理のフローチャートを示します。

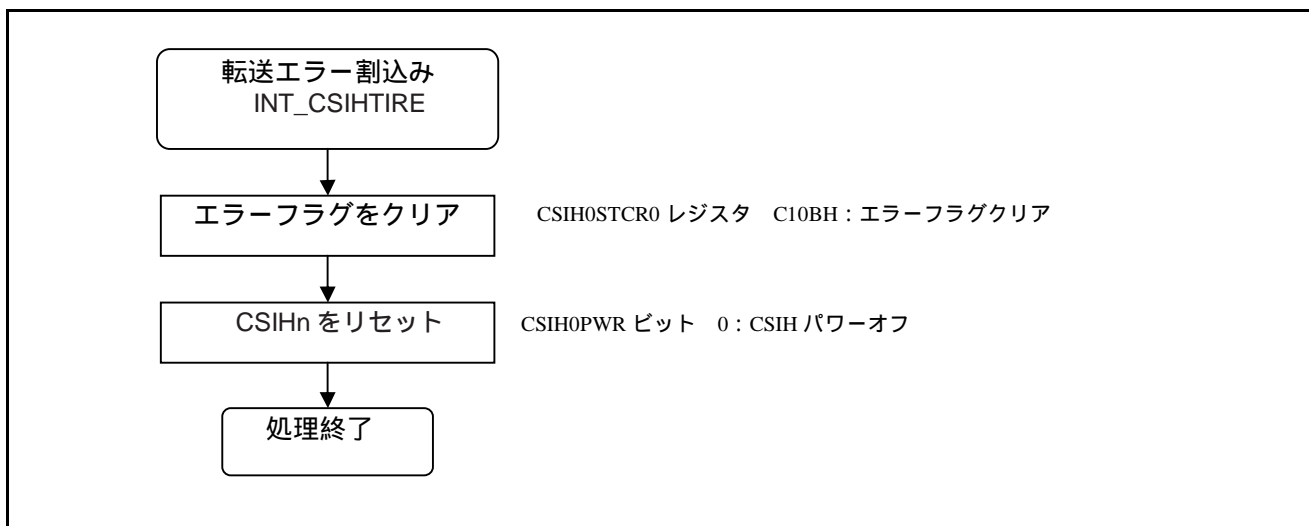


図4.7 転送エラー割り込み処理

図 4.8に転送終了割り込み処理のフローチャートを示します。

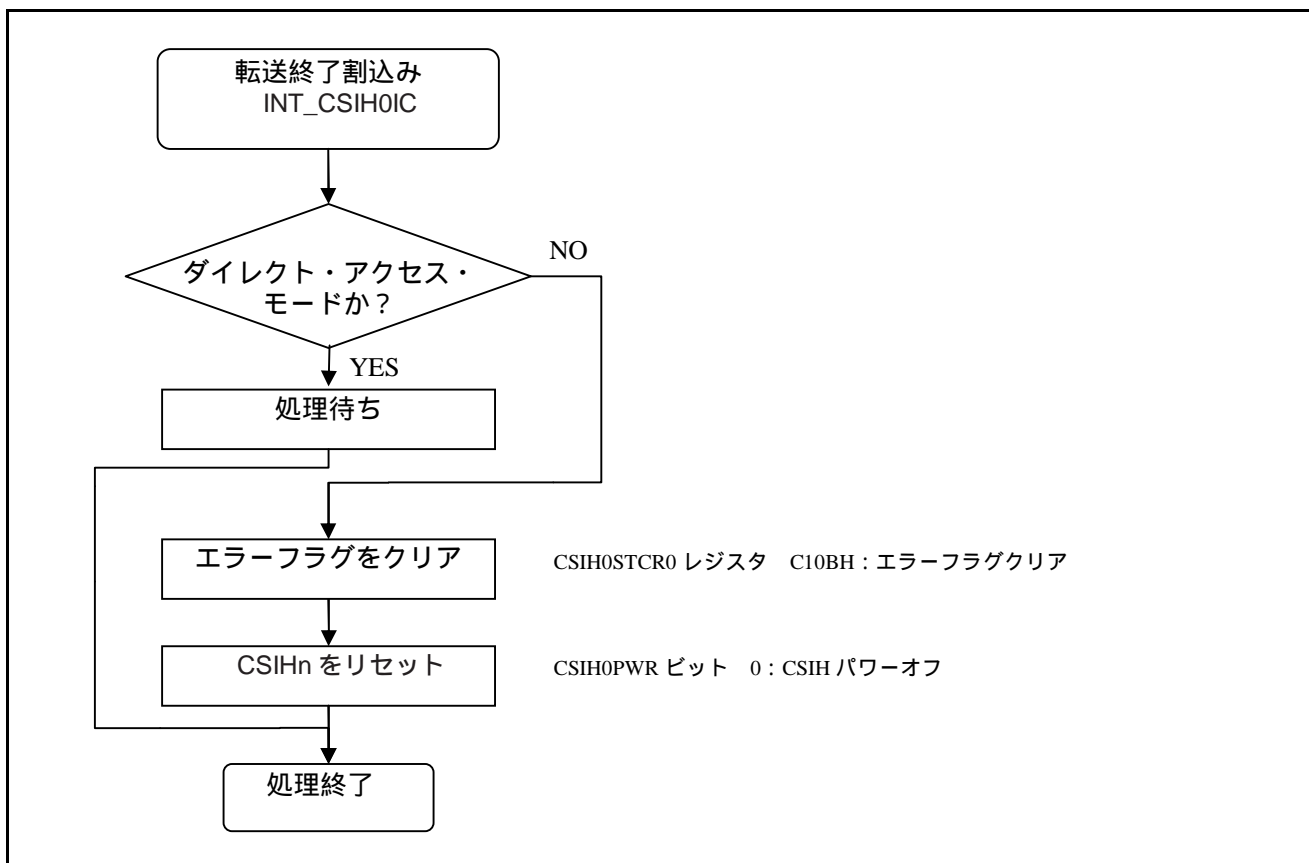


図4.8 転送終了割り込み処理

図 4.9にダイレクトアクセスモードでの、転送ジョブ終了割り込み処理のフローチャートを示します。

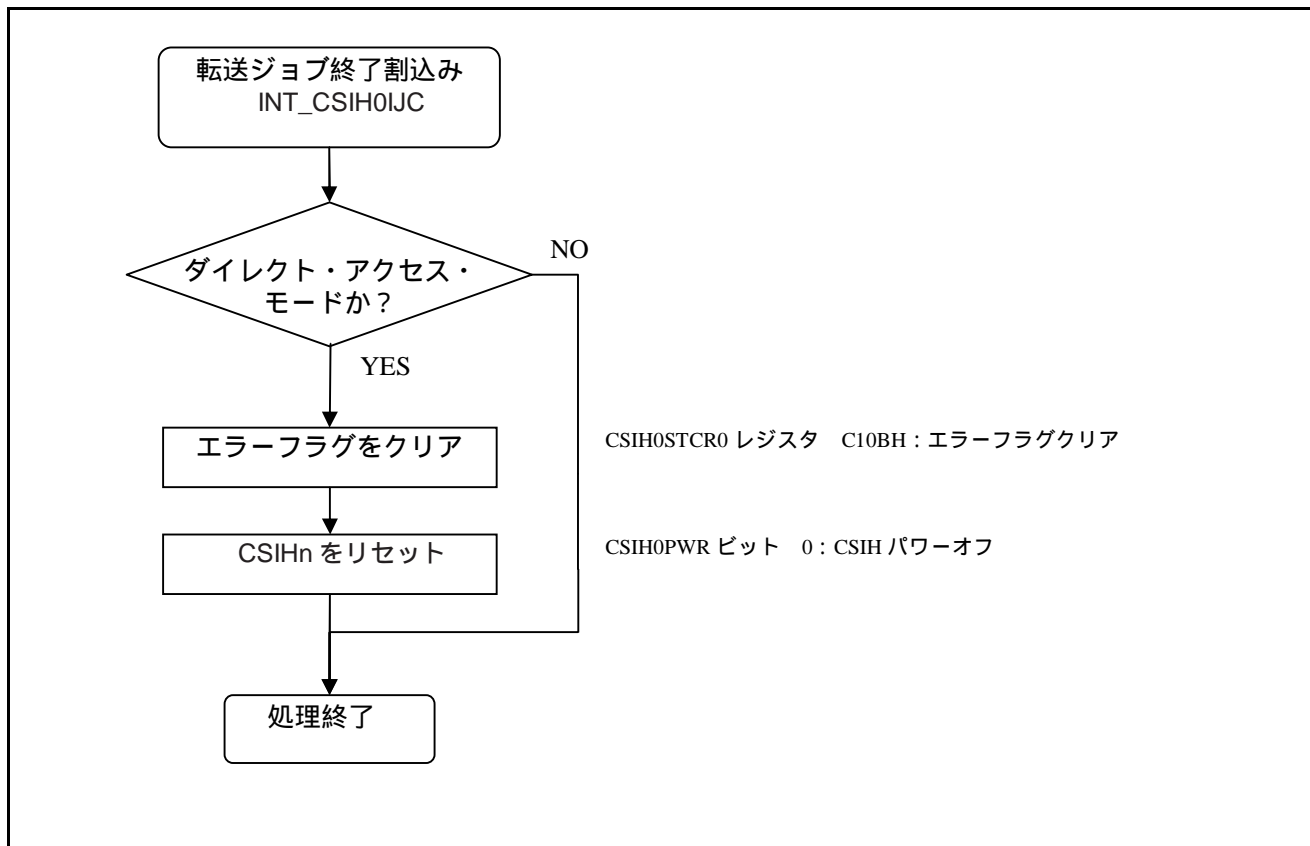


図4.9 転送ジョブ終了割り込み処理

図 4.10に、受信エラー割り込み処理のフローチャートを示します。

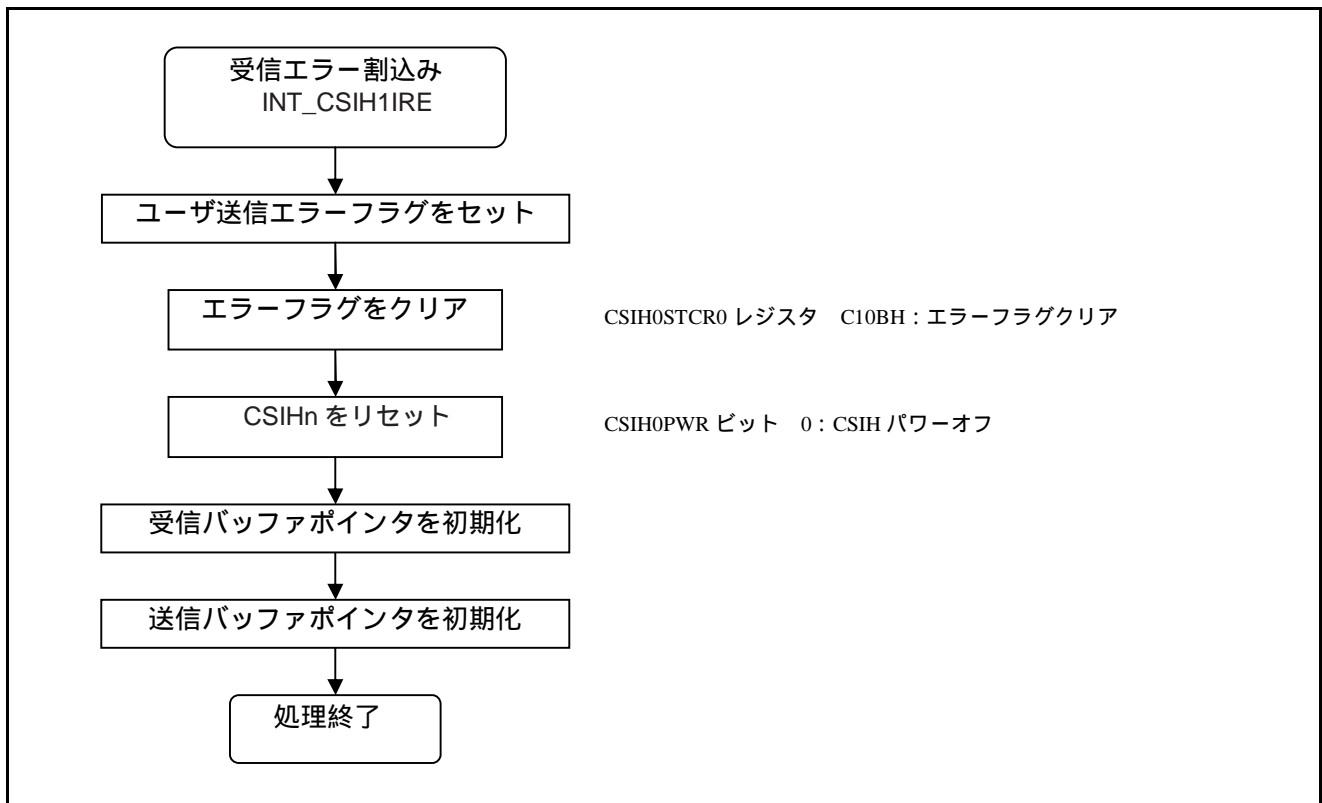


図4.10 受信エラー割り込み処理

図 4.11 に受信割り込み処理のフローチャートを示します。

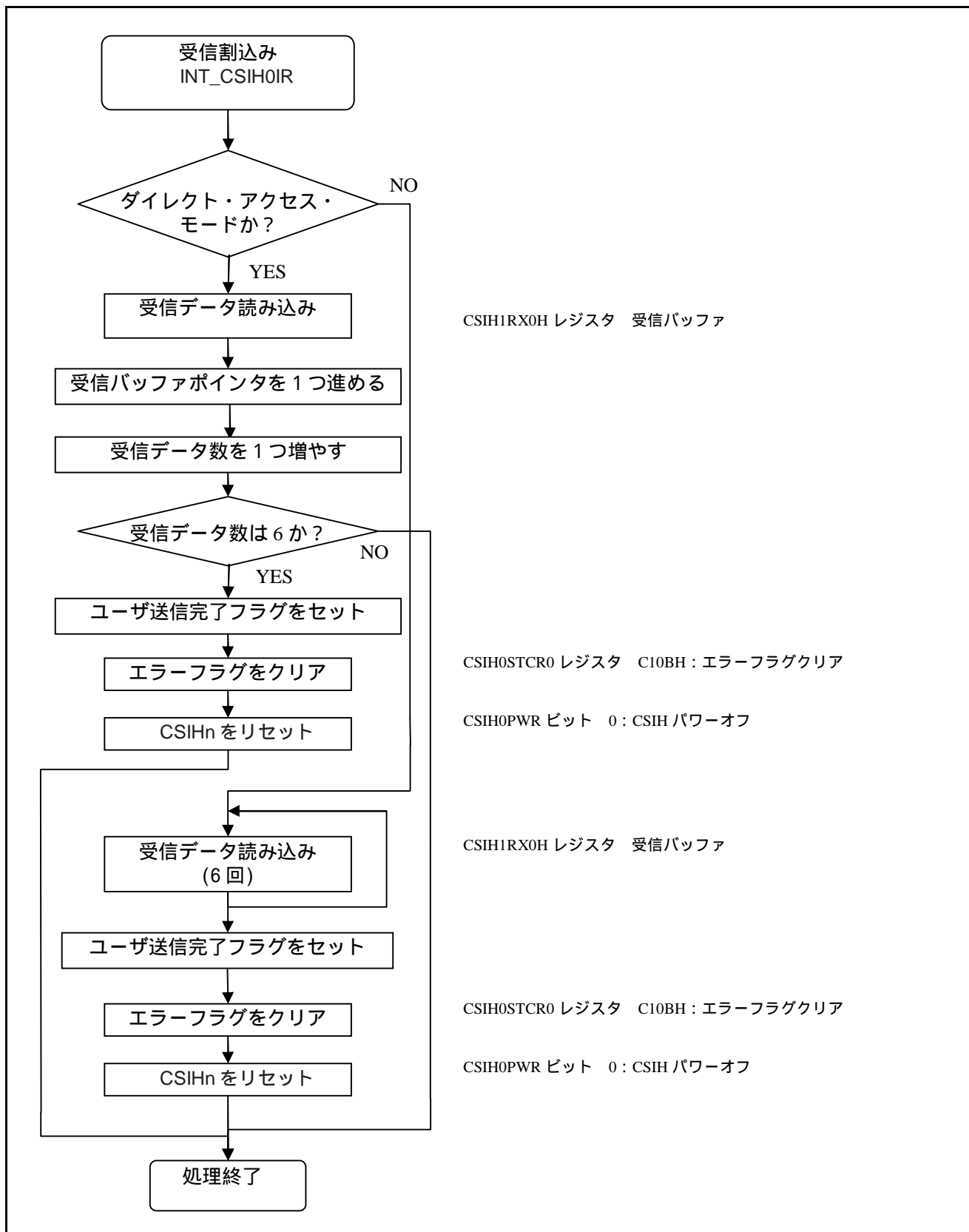


図4.11 受信割り込み処理

5. サンプルコード

サンプルコードは、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。

6. 参考ドキュメント

ユーザズマニュアル：ハードウェア

V850E2/ML4 ユーザズマニュアル ハードウェア編 (R01UH0262JJ)

(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。)

テクニカルアップデート/テクニカルニュース

(最新の情報をルネサス 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	V850E2/ML4 アプリケーションノート A/D コンバータ制御編
------	-------------------------------------

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2012.06.22	—	初版発行

すべての商標および登録商標は、それぞれの所有者に帰属します。

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くと、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. 本資料に収録された回路、ソフトウェアおよびこれらに適用する情報は、半導体製品の動作例、応用例を提示するものです。お客様の機器・システムの設計において、回路、ソフトウェアおよびこれらに適用する情報を引用する場合には、お客様の責任において行ってください。これらの使用に際して、お客様または第三者に法的な損害が生じ、当社は、一切その責任を負いません。
2. 本資料に収録されている情報は、正確さを期すため慎重に作成したものでありますが、誤りがないことを保証するものではありません。万一、本資料に収録されている情報の誤りに起因する損害が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。
3. 本資料に収録された製品データ、表、図、プログラム、アルゴリズム、応用回路例等の情報の引用に際して発生した第三者の特許権、著作権その他の知的財産権に対する侵害に際し、当社は、何らの責任を負うものではありません。当社は、本資料に基づきお客様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。
4. 当社製品を電源、電圧、周波数などについて誤り、かかる電源、電圧、周波数による法外な使用に際し、当社は、一切その責任を負いません。
5. 当社は、当社製品の消費電流を「標準電流」および「高消費電流」に分類しております。
消費電流は、以下に示す用途に製品が使用されることを前提としてお示します。
標準電流： コンピュータ、OA機器、産業機器、計測機器、AV機器、
家電、工業機械、パーソナル機器、産業用ロボット等
高消費電流： 鉄道車両（制御系、電源、駆動等）、交通信号機制御、
防災・警報機器、各種安全装置等
当社製品は、実装条件、動作環境などによって異なる動作・システム（法外な使用例、入出力極め込み使用する場合）、もしくは多大な物理的ストレスを発生させるおそれのある動作・システム（誤り動作システム、軍事用途等）に使用されることを推奨していません。使用することはお控えください。なお、推奨しない用途に当社製品を使用したことによるお客様または第三者に損害が生じた場合、当社は一切その責任を負いません。なお、ご不明点がある場合は、当社営業にお問い合わせてください。
6. 当社製品をご使用する際は、当社が規定する最大定格、動作電圧範囲、動作特性、実装条件その他の保証範囲内でご利用ください。当社保証範囲を越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。
7. 当社は、当社製品の消費電力に制限を設けていますが、半導体製品は発熱を伴うため、使用条件によって発熱を伴う場合があります。また、当社製品は耐熱耐湿設計には行っておりません。当社製品の故障または損傷が発生した場合は、入会事務、火災事故、社会的損害等生じかねないよう、お客様の責任において、実装設計、回路設計、駆動条件設計等の実装設計およびシーリング処理等、お客様の設備・システムとしての故障防止を行ってください。特に、マイコンソフトウェアは、駆動時の動作は複雑なため、お客様の設備・システムとしての安全確保をお客様の責任で行ってください。
8. 当社製品の保護機能等の特長につきましては、製品説明に必ず当社営業窓口までお問い合わせください。ご説明に際しては、特定の物質の含有、使用を制限するRoHS指令等、適用される規制関連法令を十分御察のうえ、かかる指令に適合するようご説明もいたします。お客様がかかる指令を遵守しないことにより生じた損害に際して、当社は、一切その責任を負いません。
9. 本資料に収録されている当社製品および技術企業内外の第三者の権利による製品、装置、部品が記載されている設備・システムに適用することはお控えください。また、当社製品および技術企業関連の特許等の権利、軍事用途の権利その他の軍事用途に使用しないで行ってください。当社製品または技術を開示する場合は、「外注品等及IT外注品等」その権利加減関係等を守り、かかる法令の定めるところによる適宜な手続を行ってください。
10. お客様の機器等により、本資料記載の実装条件に照準して当社製品が故障され、その故障から損害が生じた場合、当社は何らの責任を負わず、お客様にご負担して頂きますのでご了承ください。
11. 本資料の資料または一部を当社の文書にある権利の承認を得ることなく転載または複製することをお控えください。

注1. 本資料において使用されている「注記」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傘下の関連会社の商標または登録商標であることを示しています。

注2. 本資料において使用されている「注記」とは、注1において記載された商標、関連製品をいいます。

RENESAS

ルネサスエレクトロニクス株式会社

販売お問い合わせ窓口

<http://www.renesas.com>

※営業時間：9:00～17:00（日・祭日を除く）※お問い合わせの際は、お客様のご住所を必ずお示しください。

ルネサス エレクトロニクス福岡営業所 〒100-8354 千代田区六千寿2-6-2（日本ビル）

092-6211-6307

無料の技術的なお問い合わせおよび資料のご請求は下記へどうぞ。
株式会社 営業課： help@open.renesas.com/contact/