

---

# V850E2/MN4

R01AN0924JJ0100

## CSIH 制御編

Rev.1.00

2012.02.10

---

### 要旨

本アプリケーションノートは、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たものです。サンプルコードでは、CSIH0 と CSIH3 の間でデータの送信/受信を行います。CSIH0 はマスタ・モードで送信し、CSIH3 はスレーブ・モードで受信します。このサンプルコードはダイレクト・アクセスとデュアル・バッファ・モードの二種類のメモリ・モードを使用しています。

### 動作確認デバイス

V850E2/MN4 マイクロコントローラ

### 目次

1. 概要 .....	2
2. 利用環境 .....	8
3. ソフトウェアについて .....	9
4. アプリケーション例 .....	10

## 1. 概要

このアプリケーションノートでは、CSIH の以下4つの動作モードを使用例として説明します。

- 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

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 信号を介して供給します。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、通信クロックが供給されます。

サンプルコードの詳細については 4.1 フローチャートで説明します。

マスタ、デュアル・バッファ、送信オンリー・バッファ・モードの主な内容は次の通りです。

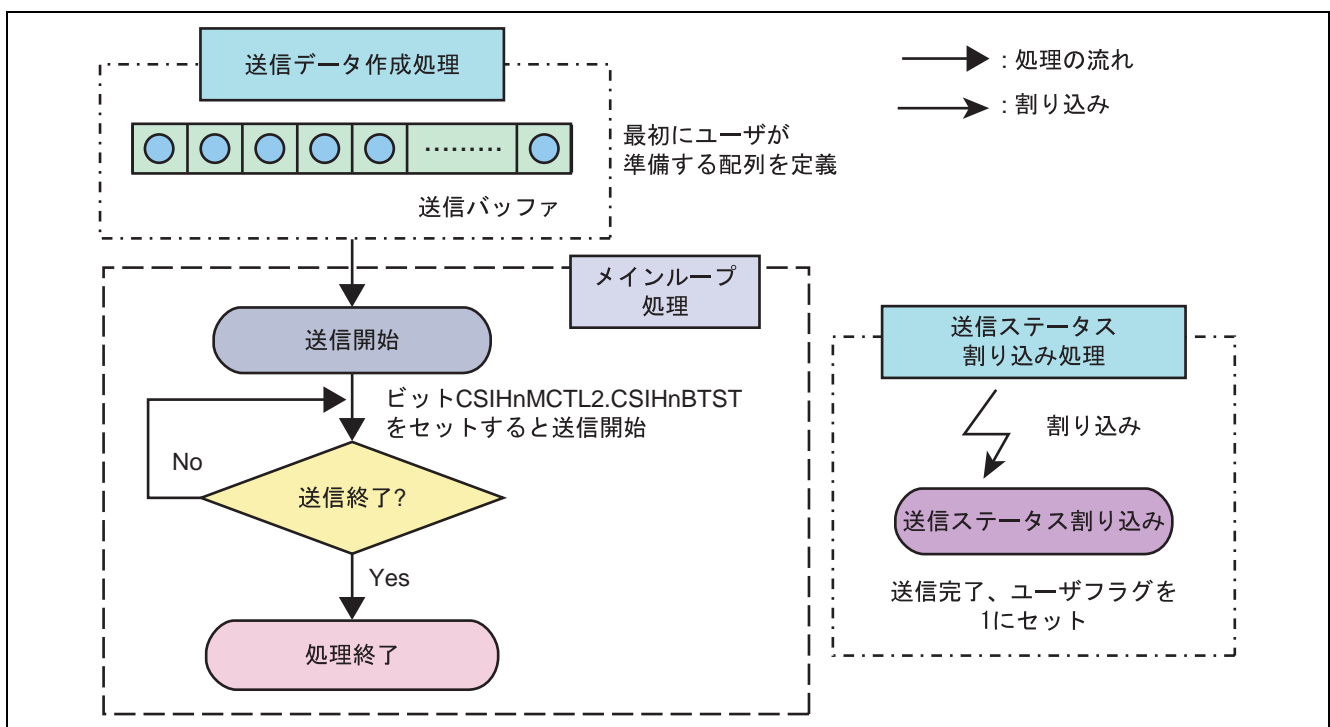


図 1.1 マスタ、デュアル・バッファ、送信オンリー・バッファ・モード

スレーブ、デュアル・バッファ、受信モードの主な内容は次の通りです。

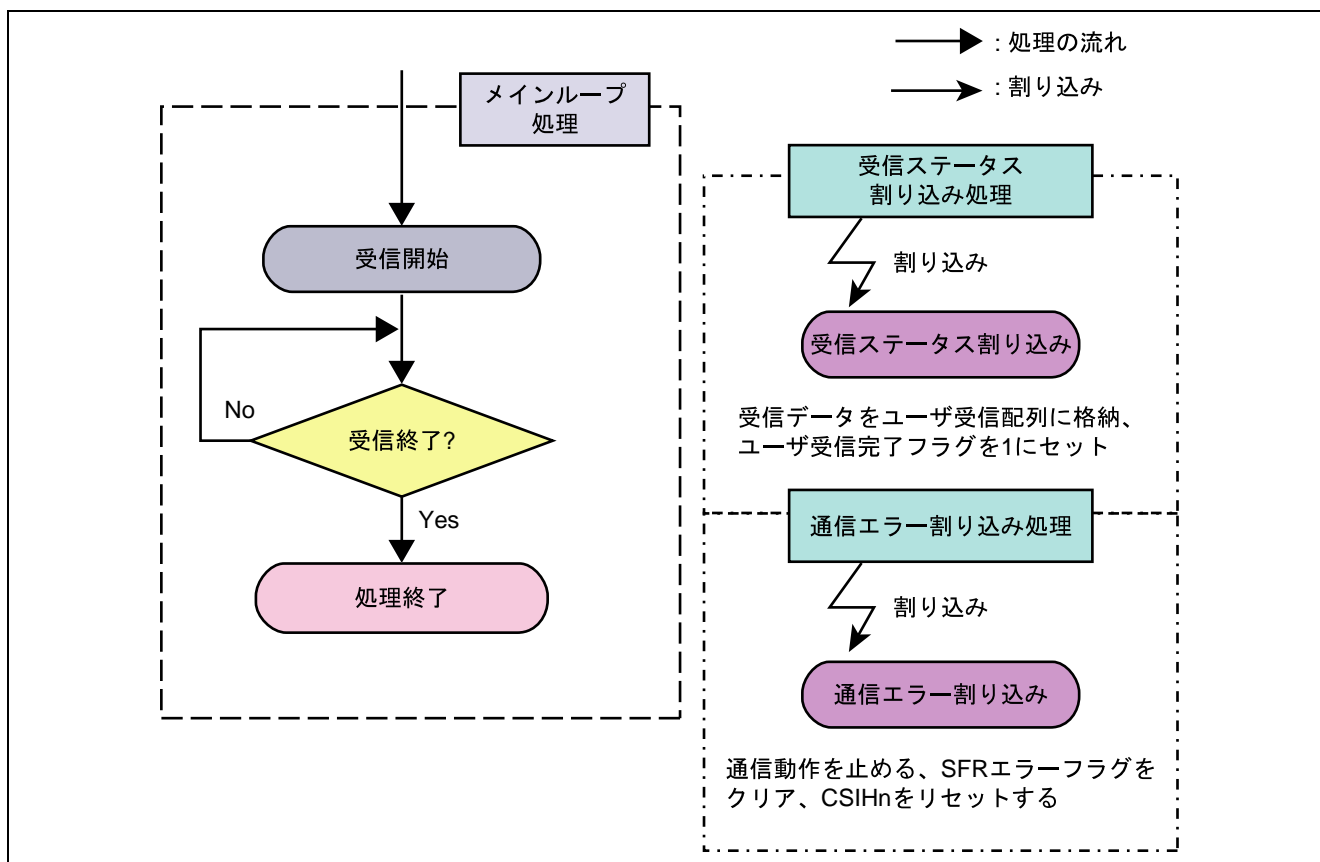


図 1.2 スレーブ、デュアル・バッファ、受信モード

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（ジョブ）の主な内容は次の通りです。

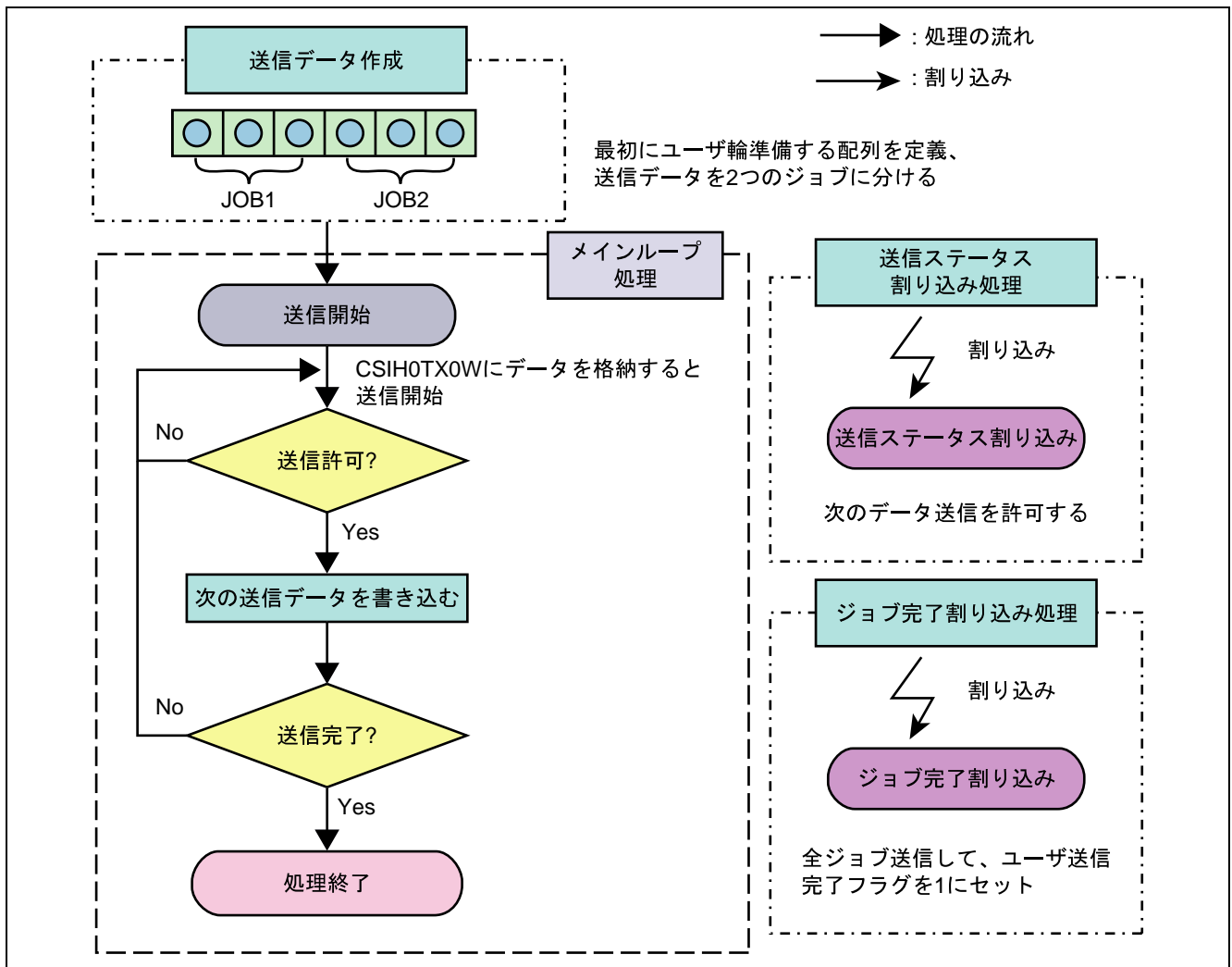


図 1.3 マスタ、ダイレクト・アクセス、送信オンリー・バッファ・モード（ジョブ）

スレーブ、ダイレクト・アクセス、受信モードの主な内容は次の通りです。

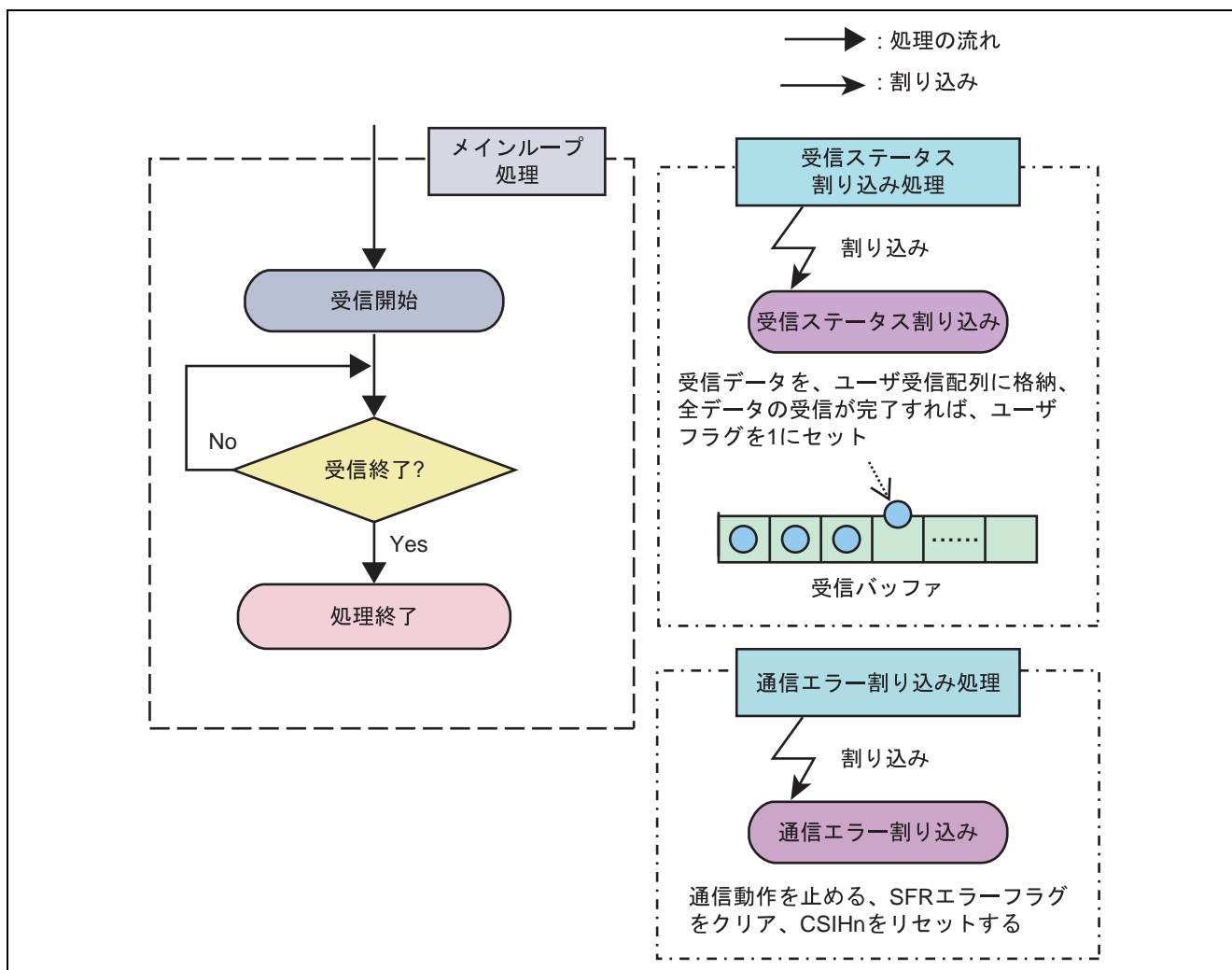


図 1.4 スレーブ、ダイレクト・アクセス、受信モード

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	CSIH3	
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・ビットなし	
データ長	8 ビット	
ポー・レート	64kbps	
FIFO 容量	なし	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)	2JOB	なし
EDL (拡張データ長)	なし	
LBM (ループ・バック・モード)	なし	
SS (スレーブ選択)	なし	

## 1.1 初期化

汎用レジスタや機能端子などの初期化処理を行います。

<ポートの設定>

- ポート n・ファンクション・コントロール・エクспанション・レジスタ (PFCE<sub>n</sub>)
- ポート n・ファンクション・コントロール・レジスタ (PFC<sub>n</sub>)
- ポート n・モード・コントロール・レジスタ (PMC<sub>n</sub>)
- ポート n・モード・レジスタ (PM<sub>n</sub>)

## 1.2 CSIH の設定

CSIH の動作を制御するために、以下の設定を行います。詳細は、4.2 レジスタの設定 を参照ください。

<CSIH 制御の設定>

- CSIH<sub>n</sub> 制御レジスタ 0 (CSIH<sub>n</sub>CTL0)
- CSIH<sub>n</sub> 制御レジスタ 1 (CSIH<sub>n</sub>CTL1)
- CSIH<sub>n</sub> 制御レジスタ 2 (CSIH<sub>n</sub>CTL2)
- CSIH<sub>n</sub> メモリ制御レジスタ 0 (CSIH<sub>n</sub>MCTL0)
- CSIH<sub>n</sub> メモリ制御レジスタ 1 (CSIH<sub>n</sub>MCTL1)
- CSIH<sub>n</sub> メモリ制御レジスタ 2 (CSIH<sub>n</sub>MCTL2)
- CSIH<sub>n</sub> コンフィギュレーション・レジスタ x (CSIH<sub>n</sub>CFG<sub>x</sub>)

## 1.3 割り込み許可

EI 命令で割り込みを許可します。

## 1.4 メインループ処理

- CSIH のデータ転送の動作は、メモリ・モードによって異なります。  
デュアル・バッファ・モードの場合は、送信側 CSIH0 で 9 バイト・データを送信し、受信側 CSIH3 で CSIH0 から送信された 9 バイトデータを受信します。内蔵 RAM 領域には、送信データと受信データの格納用ユーザ送信/受信配列として、それぞれ 9 バイト (=1 バイト (1 データ) ×9) の領域を確保しています。  
ダイレクト・アクセス・モードの場合は、送信側 CSIH0 で 6 バイト・データを用意し、3 バイトずつを 1 つのジョブに分けて送信します。受信側 CSIH3 で 6 バイト・データを受信して、内蔵 RAM 領域に格納します。送信用/受信用のユーザ配列はそれぞれ 6 バイト (=1 バイト (1 データ) ×6) の領域を確保しています。
- CSIH のデータ転送を開始する条件は、メモリ・モードによって異なります。  
＜1＞CSIH マスタ・モードで送信するとき：  
ダイレクト・アクセス・モードの場合は、CSIH の設定完了後、CSIHnTX0H レジスタにデータを書き込むことで、データ送信が開始します。  
デュアル・バッファ・モードの場合は、CSIHnMCTL2.CSIHnBTST ビットを 1 にセットすることで送信が開始します。  
＜2＞CSIH スレーブ・モードで受信するとき：  
ダイレクト・アクセス・モードの場合は、外部クロック CSIHnTSCCK を検出したことで、受信が開始します。  
デュアル・バッファ・モードの場合は、CSIHnMCTL2.CSIHnBTST ビットを 1 にセットすることで受信が開始します。
- 割り込みの発生タイミングについては、以下の通りです。  
ダイレクト・アクセス送信モードの場合、各データを送信した後、送信ステータス割り込み (CSIHnTIC) が発生します。ジョブ・モードの場合、CSIH0CTL0.JOBE ビットを 1 にセットし、そのジョブの転送動作の完了後、ジョブ完了割り込み (CSIHnTIJC) が発生します。  
ダイレクト・アクセス受信モードでは、各データを受信した後、受信ステータス割り込み (CSIHnTIR) が発生します。  
デュアル・バッファ・モードで通信する場合は、設定した転送データ数のデータを転送した後、送信ステータス割り込み (CSIHnTIC) と受信ステータス割り込み (CSIHnTIR) が発生します。  
通信エラーが発生した後、通信ステータス割り込み (CSIHnTIRE) が発生します。

## 2. 利用環境

この章では、このサンプルコードで使用する場合の回路図および利用環境を説明します。

### 2.1 回路図

回路図の詳細は V850E2/MN4 ターゲット・ボード QB-V850E2MN4DUAL-TB ユーザーズマニュアル (R20UT0683XJ) を参照してください。

このプログラムでは、CSIH0 と CSIH3 の間で CSI 通信を行います。CSIH0 はマスタ・モードで送信し、CSIH3 はスレーブ・モードで受信します。CSIH0 の端子 SO0F は P4\_12、端子 SI0F は P4\_13、端子 SCK0F は P4\_11 を使用しています。CSIH3 の端子 SO3F は P4\_3、端子 SI3F は P4\_6、端子 SCK3F は P4\_7 を使用しています。端子間の接続については、SO0F が SI3F と、SI0F が SO3F と、SCK0F が SCK3F と接続しています。

LED1 と LED2 はポート 13 と接続しています。LED1 は P13\_7、LED2 は P13\_6 を使用しています。

### 2.2 開発環境

サンプルコードを動作させるために、次の開発ツールをインストールする必要があります。

- CubeSuite+

ルネサスエレクトロニクス社製統合開発環境 CubeSuite+は、コーディング、アセンブル/コンパイル、エミュレータやシミュレータを使ったデバッグ、フラッシュプログラミング等のお客様のアプリケーション開発に必要なあらゆるソフトウェア開発ツールをシームレスかつ簡単に操作することができます。

- MINICUBE

ルネサスエレクトロニクス社製 JTAG 接続方式の汎用型インサーキットエミュレータです。実プロセッサをオンボードの状態でのデバッグでき、透過性の高い安定したエミュレーション機能を提供しています。

MINICUBE と TB ボードを接続するには、変換アダプタが必要です。

- Multi

Green Hills software, Inc.製統合開発ツール

- IAR Embedded Workbench

IAR システムズ社製統合開発ツール



### 3. ソフトウェアについて

この章では、ダウンロードする圧縮ファイルのファイル構成について説明します。

#### 3.1 ファイル構成

ダウンロードする圧縮ファイルのファイル構成は、次のようになっています。

ファイル名 (ツール構造)	説明	共通ソース ファイル	CubeSuite+用 ファイル	Multi 用 ファイル
crtE.s	ハードウェア初期化処理		●	
startup.s				●
V850E2MN4.dir	リンク・ディレクティブ・ファイル		●	
V850E2_MN4 CSIH.ld				●
vector.s	ベクタ・テーブル			●
csih.h	変数、関数宣言	●		
main.c	メイン処理	●		
initial.c	ソフトウェア初期化処理	●		
csih_transmit.c	送信処理	●		
csih_receive.c	受信処理	●		
interrupt.c	割り込み処理	●		

## 4. アプリケーション例

この章では、CSIH 機能の設定について説明します。

### 4.1 フローチャート

このサンプルコードのフローチャートを、次に示します。動作モード設定により、各サブルーチンに入ります。

#### 4.1.1 メイン処理フロー

メイン処理フローでは、まずマスク割込みを禁止します。各設定を行ったあと、マスク割込みと送受信ステータス割込みの許可を行います。メインループ処理では、通信制御処理と通信完了待ちを繰り返し実行します。

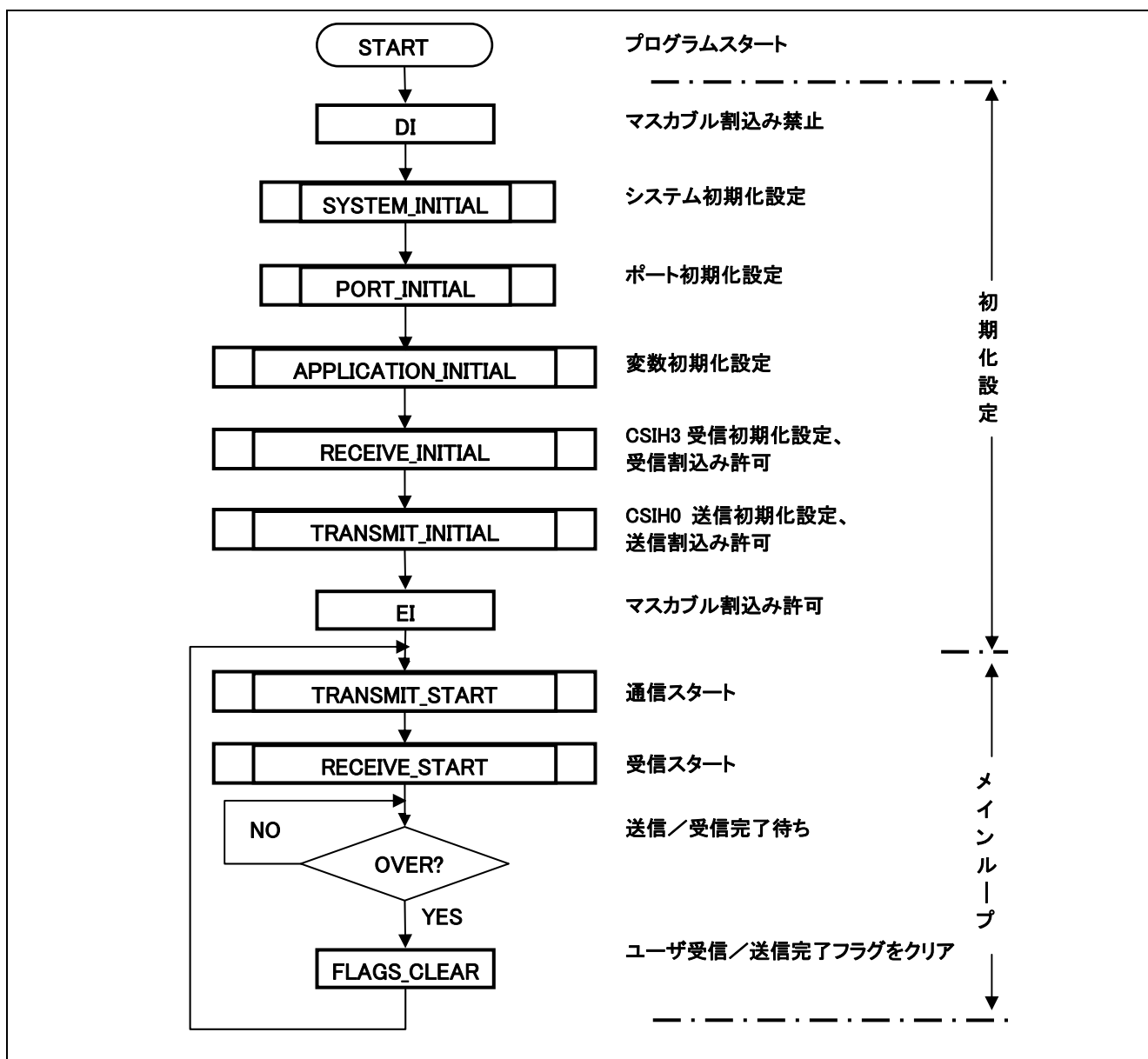


図 4.1 メイン処理フロー

## 4.1.2 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。

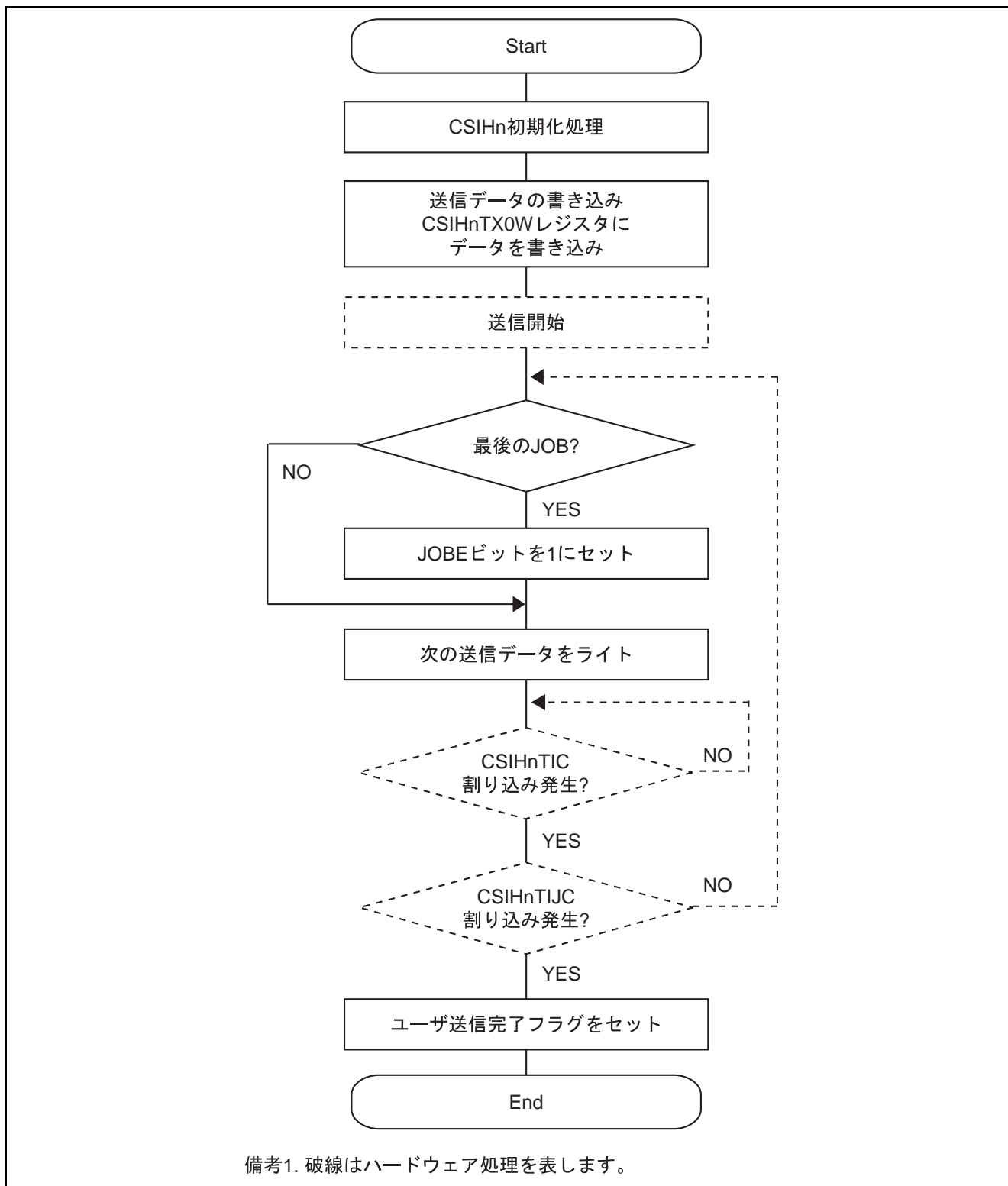


図 4.2 マスタ、ダイレクト・アクセス、送信オンリー・バッファ・モードのフローチャート

## 4.1.3 スレーブ、ダイレクト・アクセス、受信モード

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

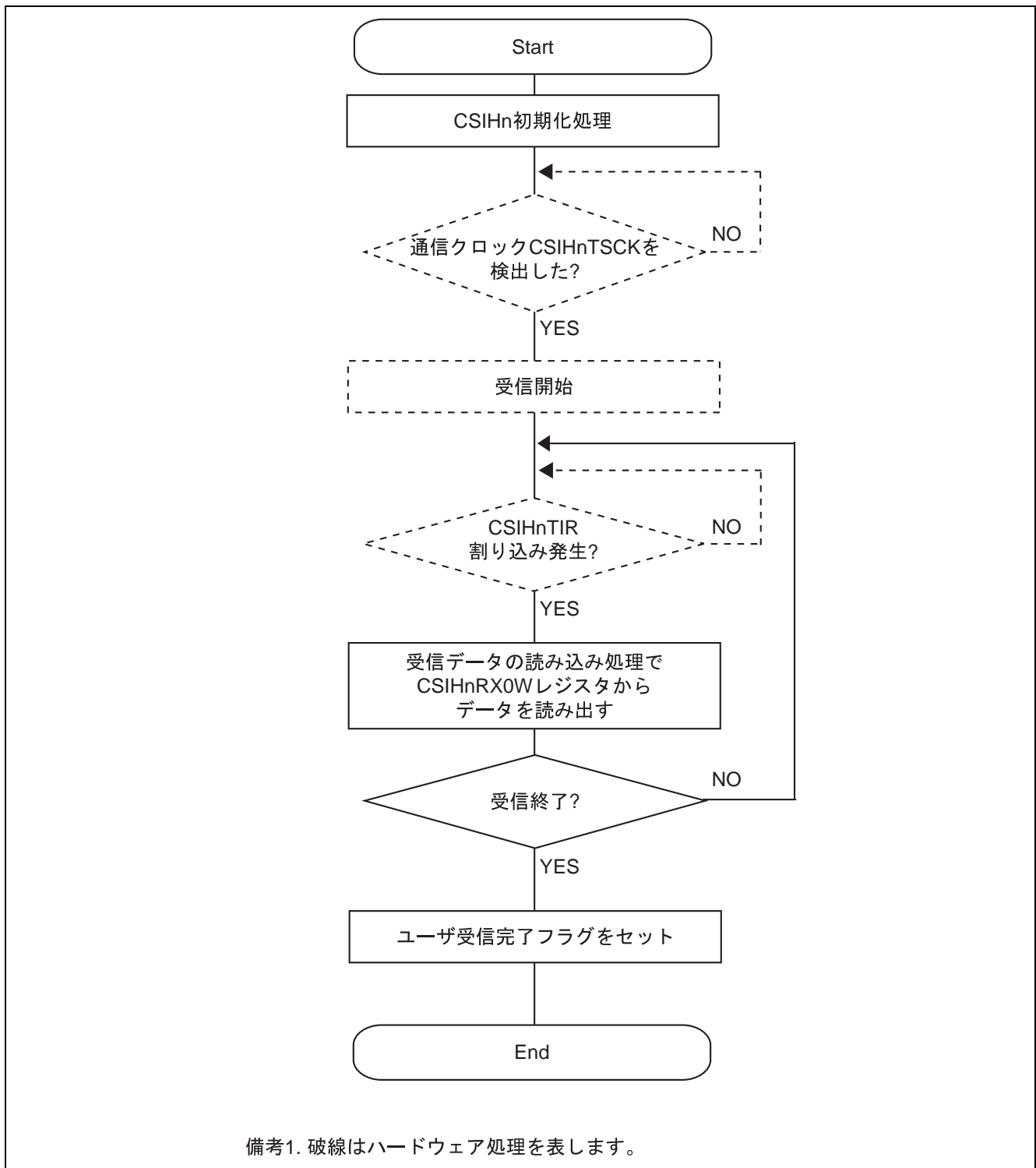


図 4.3 スレーブ、ダイレクト・アクセス、受信モードのフローチャート

## 4.1.4 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、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。

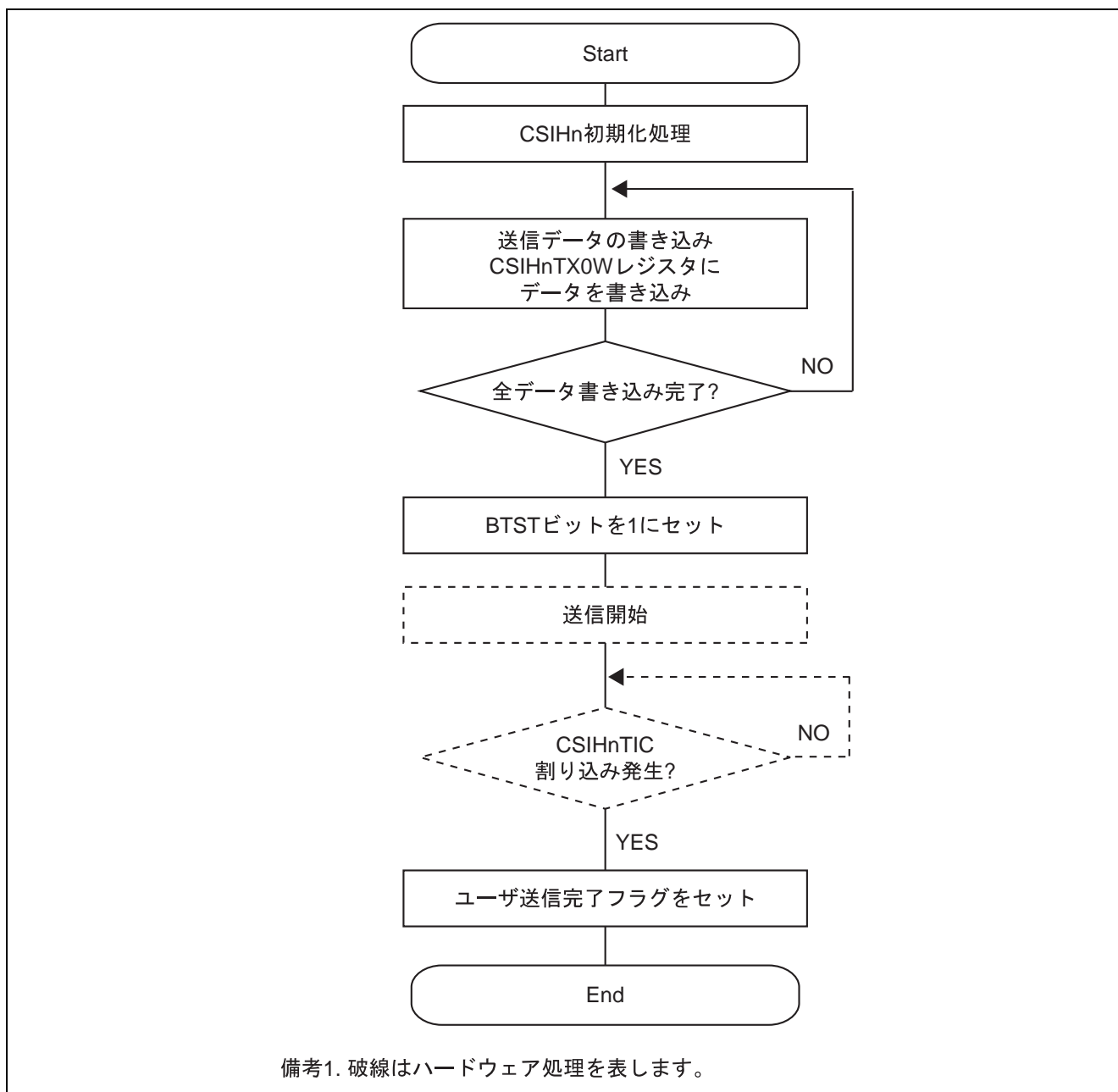


図 4.4 マスタ、デュアル・バッファ、送信オンリー・バッファ・モードのフローチャート

## 4.1.5 スレーブ、デュアル・バッファ、受信モード

スレーブ、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。

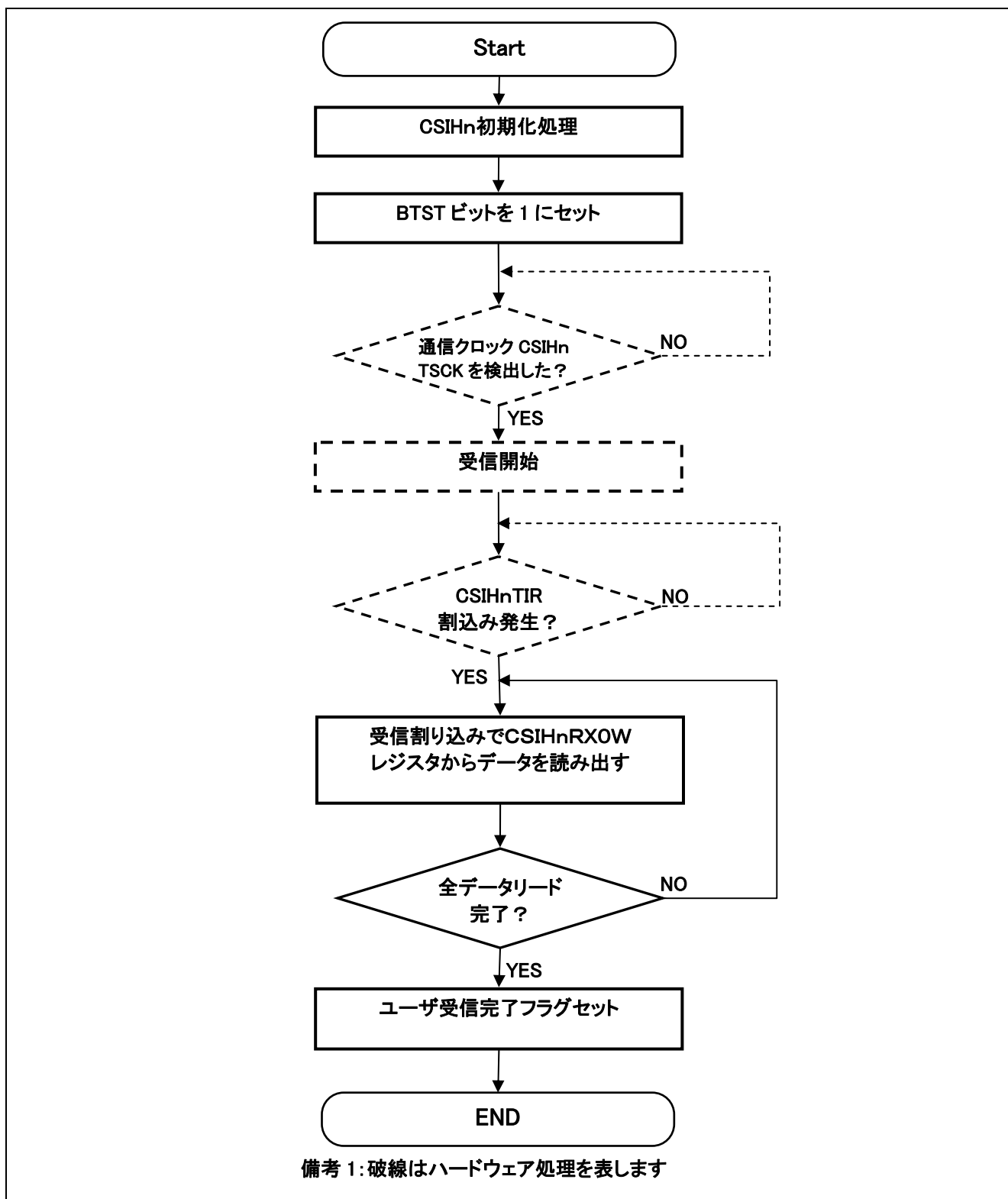


図 4.5 スレーブ、デュアル・バッファ、受信モードのフローチャート

#### 4.1.6 通信エラー割り込み処理

通信中エラーが発生した場合、通信エラー割り込みに入り、割り込み処理を行います。通信動作を止めて、SFR エラー・フラグをクリアします。同時に、CSIHn をリセットします。

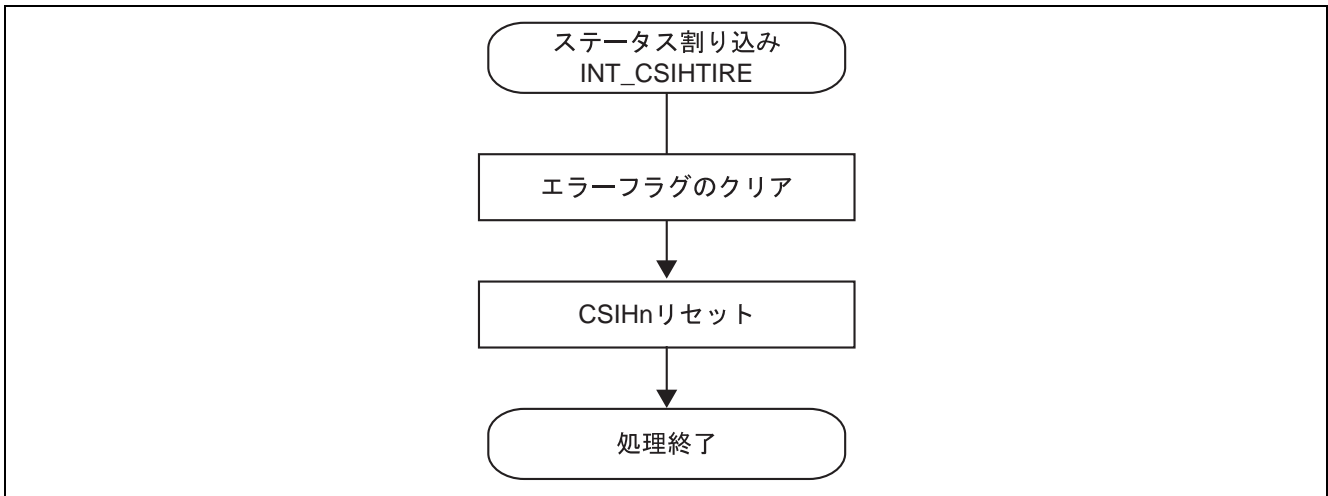


図 4.6 通信エラー割り込み処理

## 4.2 レジスタの設定

この節では、4.1 フローチャートに基づき、各レジスタの設定について説明します。CSIHの動作を制御するため、以下のレジスタを設定します。

### 4.2.1 ポートの設定

このアプリケーションノートでは、CSIH0とCSIH3の2つのマクロを使って、シリアル送受信動作を行っています。CSIH0とCSIH3の端子が有効になるように、ポートを必ず適切に設定してください。

LEDはポート13と接続しています。LED1はP13\_7、LED2はP13\_6を使用しています。

マクロ	端子	PMC	PFCE	PFC	PM	対応機能
CSIH0	SO0F	1	1	1	0	兼用モード4、出力
	SI0F	1	1	1	1	兼用モード4、入力
	SCK0F	1	1	1	0	兼用モード4、出力
CSIH3	SO3F	1	1	1	0	兼用モード4、出力
	SI3F	1	1	1	1	兼用モード4、入力
	SCK3F	1	1	1	1	兼用モード4、入力
PORT	P13_6	0	0	0	0	ポート・モード、出力
	P13_7	0	0	0	0	ポート・モード、出力

#### 設定例

```

/* alternative mode 4 in that csih
CSIH0:Master Mode,transmission
  P4_12: CSIH0A0SO
  P4_13: CSIH0A0SI
  P4_11: CSIH0A0SCO
CSIH3:Slave Mode,reception
  P4_3: CSIH3A3SO
  P4_6: CSIH3A3SI
  P4_7: CSIH3A3SCI */
PFCE4 = 0x38c8;
PFC4  = 0x38c8;
PMC4  = 0x38c8;
PM4   = 0x20c0;

/* P13_6,7: LEDs,IO,OUTPUT */
PFCE13 = 0x0000;
PFC13  = 0x0000;
PMC13  = 0x0000;
PM13   = 0x0000;

```



## 4.2.2 CSIH 制御レジスタ 2 (CSIHnCTL2)

CSIHnCTL2 レジスタは、通信クロックを選択するレジスタです。

マスタ・モードでは、CSIHnCTL2 レジスタの CSIHnPRS[2:0] と CSIHnBRS[11:0] ビットで送信ポー・レートを選択できます。使用できる最大ポー・レートは、マスタ・モードでは Pclk/4、スレーブ・モードでは Pclk/6 です。また使用できる最小ポー・レートは、いずれのモードでも Pclk/524160 です。

サンプルコードでは、通信クロックを 64kbps に設定し、CSIHnPRS[2:0] を 1、CSIHnBRS[11:0] を 260 に設定しています。

アクセス 16 ビット単位でリード/ライト可能です。

アドレス <CSIHn\_base\_OS> + 0014<sub>H</sub>

初期値 E000<sub>H</sub> どのリセット要因でも初期化されます。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSIHnPRS[2:0]			0	CSIHnBRS[11:0]											
RW	RW	RW	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

注意 本レジスタの内容は、CSIHnCTL0.CSIHnPWR = 0 のときのみ変更することができます。

表 22-20 CSIHnCTL2 レジスタの内容 (1/2)

ビット位置	ビット名	機能			
15 ~ 13	CSIHnPRS [2:0]	動作モードと基本クロックの値を選択します。			
		CSIHnPRS2	CSIHnPRS1	CSIHnPRS0	基本クロック (PRISOUT) の選択
		0	0	0	PCLK (マスタ・モード)
		0	0	1	PCLK / 2 (マスタ・モード)
		0	1	0	PCLK / 4 (マスタ・モード)
		0	1	1	PCLK / 8 (マスタ・モード)
		1	0	0	PCLK / 16 (マスタ・モード)
		1	0	1	PCLK / 32 (マスタ・モード)
		1	1	0	PCLK / 64 (マスタ・モード)
1	1	1	CSIHnTSCK(in) 経由の外部クロック (スレーブ・モード)		

図 4.7 CSIHnCTL2 レジスタのフォーマット (1/2)

表 22-20 CSIHnCTL2 レジスタの内容 (2/2)

ビット位置	ビット名	機能	
11 ~ 0	CSIHnBRS [11:0]	ポー・レートを選択します。	
		CSIHnBRS[11:0]	CSIHnTBLK クロックの選択
		0	BRG の停止
		1	$PCLK / (2^m \times 1 \times 2)$
		2	$PCLK / (2^m \times 2 \times 2)$
		3	$PCLK / (2^m \times 3 \times 2)$
		4	$PCLK / (2^m \times 4 \times 2)$
		...	...
		4095	$PCLK / (2^m \times 4095 \times 2)$

備考 m = 0-6 : CSIHnPRS[2:0] で設定した値

図 4.8 CSIHnCTL2 レジスタのフォーマット (2/2)

CSIHnCTL2 = 0x2104;  
CSIHnCTL2 = 0xE000;

/\* master mode;Pclk/2<sup>1</sup>\*260\*2 \*/  
/\* slave mode \*/

## 4.2.3 CSIH 制御レジスタ 0 (CSIHnCTL0)

CSIHnCTL0 レジスタは、動作クロックの制御、送受信の許可/禁止、CSIH のメモリ使用を設定します。また、現在のジョブの終了時に通信を強制的に停止します。

アクセス 8ビット単位または1ビット単位でリード/ライト可能です。

アドレス <CSIHn\_base\_USER> + 0000<sub>H</sub>

初期値 00<sub>H</sub> どのリセット要因でも初期化されます。

7	6	5	4	3	2	1	0
CSIHn PWR	CSIHn TXE	CSIHn RXE	0	0	0	CSIHn JOBE	CSIHn MBS
R/W	R/W	R/W	R	R	R	R/W	R/W

表 22-16 CSIHnCTL0 レジスタの内容

ビット位置	ビット名	機能
7	CSIHnPWR	動作クロックを制御します。 0: 動作クロックを停止させます。 1: 動作クロックを供給します。 CSIHnPWR をクリア (0) すると、内部回路がリセットされ、動作が停止し、CSIH がスタンバイ状態に設定されます。内部回路へのクロックの供給が停止します。  通信中に CSIHnPWR がクリア (0) されると、実行中の通信はただちに中止されます。その場合は、通信を最初からやり直す必要があります。
6	CSIHnTXE	送信を許可または禁止します。 0: 送信を禁止します。 1: 送信を許可します。
5	CSIHnRXE	受信を許可または禁止します。 0: 受信を禁止します。 1: 受信を許可します。
1	CSIHnJOBE	現在のジョブの終了時に通信を停止します (CSIHnTXoW.CSIHnEOJ = 1 (ジョブ終了) 時に送信バッファにデータをライトされると通信が終了になります)。 0: 通信停止を要求しません。 1: 通信を停止します。 このビットを使って実行中のジョブを中止することができます。このビットは自動的にクリア (0) されます。このビットをセット (1) しても、リード値は常に 0 です。 FIFO モードでは、CSIHnSTCR0.CSIHnPCT = 1 に設定することによってポイントをクリアしたあと、次の通信を開始する必要があります。 注意 CSIHnJOBE は CSIHnCTL1.CSIHnJE = 1 のときにのみ有効です。 このビットはスレープ・モードでは設定禁止です。このビットは常に 0 がリードされます。
0	CSIHnMBS	送信データまたは受信データあるいはその両方のメモリをバイパスします。 0: メモリ・モード 送信データまたは受信データあるいはその両方に CSIH のメモリを使用します。 1: ダイレクト・アクセス・モード CSIH のメモリをバイパスします。 注意 スレープ・モード時は、CSIHnCTL0.CSIHnPWR = 0 → 1 と同じタイミングで再度ライトしてください。

図 4.9 CSIHnCTL0 レジスタのフォーマット (1/2)

- 注意
1. CSIHnPWR = 0 のときは、CSIHnTXE, CSIHnRXE, CSIHnJOBE, CSIHnMBS ビットのいずれも変更しないでください。ただし CSIHnPWR ビットを 0 から 1 に変更すると同時に、CSIHnTXE, CSIHnRXE, CSIHnMBS ビットを変更することはできません。
  2. データ送信が保留中または進行中、つまり CSIHnSTR0.CSIHnTSF = 1 のときに、CSIHnTXE または CSIHnRXE または CSIHnMBS を変更しないでください。

図 4.10 CSIHnCTL0 レジスタのフォーマット (2/2)

```

CSIHnCTL0 = 0x00;          /* stop CSIH0 */
CSIHnPWR = 1;             /* permit CSIHn */
CSIHnTXE = 1;            /* permit transmission */
CSIHnRXE = 1;            /* permit reception */
CSIHnMBS = 0;            /* memory mode */
CSIHnMBS = 1;            /* direct access mode */
CSIHnJOBE = 1;           /* stop communication after this JOB */

```

## 4.2.4 CSHn 制御レジスタ 1 (CSHnCTL1)

CSHnCTL1 レジスタは、通信動作の制御を行いレジスタです。割り込みのタイミングと割り込み遅延モードの指定、各チップ選択信号のアクティブ出力レベル、最後のデータを転送したあとのチップ選択信号の動作の設定などを行うことができます。

アクセス 32ビット単位でリード/ライト可能です。

アドレス <CSHn\_base\_OS> + 0010<sub>H</sub>

初期値 0000 0000<sub>H</sub> どのリセット要因でも初期化されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	CSHn CKR	CSHn SLIT
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSHnCSL[7:0]								CSHn EDLE	CSHn JE	CSHn DCS	CSHn CSRI	CSHn LBM	CSHn SIT	CSHn HSE	CSHn SSE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

注意 本レジスタの内容は、CSHnCTL0.CSHnPWR = 0 のときにのみ変更することができます。

表 22-17 CSHnCTL1 レジスタの内容 (1/2)

ビット位置	ビット名	機能
17	CSHnCKR	CSHnTSCK のクロック位相を選択します。 0: CSHnTSCK のデフォルト・レベルはハイ・レベル 1: CSHnTSCK のデフォルト・レベルはロウ・レベル 注意 チップ・セレクト機能を使用せずこのビットを使用する場合は、CSHnCFGx.CSHnCKPx = 0 に設定してください。
16	CSHnSLIT	割り込み CSHnTIC のタイミングを選択します。 0: 通常の割り込みのタイミング (転送後に割り込みが発生します)。 1: CSHnTXoW, CSHnTXoH レジスタの内容がシフト・レジスタに転送されると、ただちに割り込みが発生します (ダイレクト・アクセス・モードでのみ機能します)。 詳細については、1329 ページの (1) 「CSHnTIC (通信割り込み)」を参照してください。
15~8	CSHnCSL[7:0]	チップ・セレクト信号 x (CSHnTCSSx) のアクティブ出力レベルを選択します (n = 0-7)。 0: チップ・セレクト信号をアクティブ・ロウにします。 1: チップ・セレクト信号をアクティブ・ハイにします。 詳細については、1314 ページの 22.3.3 「チップ・セレクト (CS) 機能」を参照してください。
7	CSHnEDLE	拡張データ長 (EDL) モードを有効または無効にします。 0: 拡張データ長モードを無効にします。 1: 拡張データ長モードを有効にします。 詳細については、1325 ページの (2) 「16 ビットを上回るデータ長」を参照してください。

図 4.11 CSHnCTL1 レジスタのフォーマット (1/3)

表 22-17 CSIHnCTL1 レジスタの内容 (2/2)

ビット位置	ビット名	機能
6	CSIHnJE	ジョブ・モードを有効または無効にします。 0: ジョブ・モードを無効にします。 1: ジョブ・モードを有効にします。 詳細については、1319 ページの 22.3.5 「ジョブ概念」を参照してください。 CSIHnCTL0.CSIHnJOBE, CSIHnTX0W.CSIHnEOJ, CSIHnTX0W.CSIHnCIRE ビットは CSIHnCTL1.CSIHnJE = 1 のときのみ有効です。 このビットはスレープ・モードでは設定禁止です。
5	CSIHnDCS	データー真性チェックを有効または無効にします。 0: データー真性チェックを無効にします。 1: データー真性チェックを有効にします。 詳細については、1342 ページの (1) 「データー真性チェック」を参照してください。
4	CSIHnCSRI	最後のデータが転送されたあとのチップ・セレクト信号の動作を定義します。 0: チップ・セレクト信号がアクティブ・レベルを保持します。 1: チップ・セレクト信号がインアクティブ・レベルに戻ります。 最後のデータの判定はダイレクト・アクセス・モード / FIFO モード時の割り込みタイミングで行います。CSIHnCTL1.CSIHnSLIT = 1 のときはダイレクト・アクセス・モードです。
3	CSIHnLBM	ループ・バック・モード (LBM) を制御します。 0: ループ・バック・モードをインアクティブにします。 1: ループ・バック・モードをアクティブにします。 詳細については、1352 ページの 22.3.15 「ループ・バック・モード」を参照してください。 このビットはスレープ・モードでは設定禁止です。
2	CSIHnSIT	割り込み遅延モードを選択します。 0: 遅延を生成しません。 1: すべての割り込みについて半クロック周期の遅延を生成します。 このビットはマスタ・モードでのみ有効になります。スレープ・モードでは遅延は生成されません。 詳細については、1337 ページの (5) 「全割り込み遅延」を参照してください。
1	CSIHnHSE	ハンドシェイク・モードを有効または無効にします。 0: ハンドシェイク機能を無効にします。 1: ハンドシェイク機能を有効にします。 詳細については、1338 ページの 22.3.13 「ハンドシェイク機能」を参照してください。
0	CSIHnSSE	スレープ選択機能を有効または無効にします。 0: 入力信号 CSIHnTSSI を無視します。 1: 入力信号 CSIHnTSSI を認識します。 スレープ選択機能を使用しない場合は、このビットを 0 に設定する必要があります (1312 ページの 22.3.2 「マスタ/スレープの接続」も参照してください)。

CSIHnCTL1.CSIHnSSE の詳細を次に示します。

表 22-18 受信におけるスレープ選択機能の動作

CSIHnCTL0. CSIHnRXE	CSIHnCTL1. CSIHnSSE	CSIHn TSSI	受信動作
0	-	-	受信を禁止します。
1	0	-	可能
1	1	0	可能
1	1	1	無効

図 4.12 CSIHnCTL1 レジスタのフォーマット (2/3)

表 22-19 送信におけるスレーブ選択機能の動作

CSIHnCTL0. CSIHnTXE	CSIHnCTL1. CSIHnSSE	CSIHn TSSI	送信動作
0	-	-	送信を禁止します。
1	0	-	可能
1	1	0	可能
1	1	1	無効

図 4.13 CSIHnCTL1 レジスタのフォーマット (3/3)

```

CSIHnCTL1 = 0x00010140;      /* TIC at start;CS0 inactive;JOB mode enable */
                               /* Output initial CSIHnTSCO value at high level */
                               /* Transmit status interrupt request is generated at beginning of
                               transmission */
                               /* Set chip select signal CS0 to active low*/
                               /* Disable extended data length mode */
                               /* Enable job mode */
                               /* Disable data consistency check */
                               /* Chip select signals retain active level */
                               /* Set loopback mode inactive */
                               /* No interrupt delay mode */
                               /* Disable handshake function */
                               /* Disable slave selection (SS) */
CSIHnCTL1 = 0x00000000;      /* Normal interrupt timing */
                               /* CS0 inactive */
                               /* Disable job mode */
CSIHnCTL1 = 0x00000000;      /* Set chip select signals to active low */
                               /* hand shaking disable */

```

## 4.2.5 CSIH コンフィギュレーション・レジスタ x (CSIHnCFGx)

CSIHnCFGx レジスタは、各チップ選択信号 CSIHnCSSx のプリスケアラ、パリティ、データ長、ブロードキャスト用のリセッソプの設定、シリアル・データ方向、クロック位相とデータ位相、強制アイドル状態の設定、アイドル時間、ホールド時間、データ間遅延時間、セットアップ時間を設定するレジスタです。

マスタ・モードでは、1つ以上のチップ選択信号を使用できます。複数のスレーブがマスタに接続されている場合は、チップ選択信号を利用して1つ以上のスレーブを通信相手として選択できます。選択されたスレーブのみが通信可能となります。

各チップ選択信号のビットはボー・レートに応じて適切な値を設定する必要があります。サンプルコードではデフォルト値を設定しています。

アクセス 32ビット単位でリード/ライト可能です。

アドレス CSIHnCFG0 : <CSIHn\_base\_OS> + 1044<sub>H</sub>

CSIHnCFG1 : <CSIHn\_base\_OS> + 1048<sub>H</sub>

CSIHnCFG2 : <CSIHn\_base\_OS> + 104C<sub>H</sub>

CSIHnCFG3 : <CSIHn\_base\_OS> + 1050<sub>H</sub>

CSIHnCFG4 : <CSIHn\_base\_OS> + 1054<sub>H</sub>

CSIHnCFG5 : <CSIHn\_base\_OS> + 1058<sub>H</sub>

CSIHnCFG6 : <CSIHn\_base\_OS> + 105C<sub>H</sub>

CSIHnCFG7 : <CSIHn\_base\_OS> + 1060<sub>H</sub>

初期値 0000 0000<sub>H</sub> どのリセット要因でも初期化されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CSIHn PSCLx[1:0]	CSIHn PSx[1:0]	CSIHnDLSx[3:0]						0	0	0	0	CSIHn RCBx	CSIHn DIRx	CSIHn CKPx	CSIHn DAPx
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSIHn IDLx	CSIHnIDx[2:0]			CSIHnHDx[3:0]				CSIHnNx[3:0]				CSIHnSPx[3:0]			
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

注意 CSIHnCTL0.CSIHnPWR = 0 のときのみライト・アクセスが可能です (同じ値をライトするなら CSIHnCTL0.CSIHnPWR = 1 でも可能です)。

図 4.14 CSIHnCFGx レジスタのフォーマット (1/5)



表 22-28 CSIHnCFGx レジスタの内容 (1/4)

ビット位置	ビット名	機能																				
31 ~ 30	CSIHn PSCLx[1:0]	チップ・セレクト信号 x のプリスケアラを選択します。 <table border="1"> <thead> <tr> <th>CSIHn PSCLx1</th> <th>CSIHn PSCLx0</th> <th>プリスケアラの出力</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>CSIHnBCLK</td> </tr> <tr> <td>0</td> <td>1</td> <td>CSIHnBCLK / 2</td> </tr> <tr> <td>1</td> <td>0</td> <td>CSIHnBCLK / 4</td> </tr> <tr> <td>1</td> <td>1</td> <td>CSIHnBCLK / 8</td> </tr> </tbody> </table> <p>これらのビットはマスタ・モードでのみ利用可能です。 CSIHnBCLK については 22.3.6 「シリアル・クロックの選択」を参照してください。</p>	CSIHn PSCLx1	CSIHn PSCLx0	プリスケアラの出力	0	0	CSIHnBCLK	0	1	CSIHnBCLK / 2	1	0	CSIHnBCLK / 4	1	1	CSIHnBCLK / 8					
CSIHn PSCLx1	CSIHn PSCLx0	プリスケアラの出力																				
0	0	CSIHnBCLK																				
0	1	CSIHnBCLK / 2																				
1	0	CSIHnBCLK / 4																				
1	1	CSIHnBCLK / 8																				
29 ~ 28	CSIHn PSx[1:0]	チップ・セレクト信号 x の送信用と受信用のパリティを選択します。 <table border="1"> <thead> <tr> <th>CSIHn PSx1</th> <th>CSIHn PSx0</th> <th>送信</th> <th>受信</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>パリティを送信しません。</td> <td>パリティの受信を期待しません。</td> </tr> <tr> <td>0</td> <td>1</td> <td>0 に固定されたパリティ・ビットを追加します。</td> <td>パリティ・ビットの受信を期待しますが、パリティの判定は行いません。</td> </tr> <tr> <td>1</td> <td>0</td> <td>奇数パリティを追加します。</td> <td>奇数パリティ・ビットの受信を期待します。</td> </tr> <tr> <td>1</td> <td>1</td> <td>偶数パリティを追加します。</td> <td>偶数パリティ・ビットの受信を期待します。</td> </tr> </tbody> </table>	CSIHn PSx1	CSIHn PSx0	送信	受信	0	0	パリティを送信しません。	パリティの受信を期待しません。	0	1	0 に固定されたパリティ・ビットを追加します。	パリティ・ビットの受信を期待しますが、パリティの判定は行いません。	1	0	奇数パリティを追加します。	奇数パリティ・ビットの受信を期待します。	1	1	偶数パリティを追加します。	偶数パリティ・ビットの受信を期待します。
CSIHn PSx1	CSIHn PSx0	送信	受信																			
0	0	パリティを送信しません。	パリティの受信を期待しません。																			
0	1	0 に固定されたパリティ・ビットを追加します。	パリティ・ビットの受信を期待しますが、パリティの判定は行いません。																			
1	0	奇数パリティを追加します。	奇数パリティ・ビットの受信を期待します。																			
1	1	偶数パリティを追加します。	偶数パリティ・ビットの受信を期待します。																			
27 ~ 24	CSIHn DLSx[3:0]	チップ・セレクト信号 x のデータ長を選択します。 <table border="1"> <thead> <tr> <th>CSIHn DLSx[3:0]</th> <th>データ長</th> </tr> </thead> <tbody> <tr> <td>0000<sub>B</sub></td> <td>16 ビット</td> </tr> <tr> <td>0001<sub>B</sub></td> <td>1 ビット</td> </tr> <tr> <td>0010<sub>B</sub></td> <td>2 ビット</td> </tr> <tr> <td>---</td> <td>---</td> </tr> <tr> <td>1111<sub>B</sub></td> <td>15 ビット</td> </tr> </tbody> </table> <p>備考： CSIHnDLSx[3:0] ビットの設定についての詳細は 22.3.9 「データ長の選択」を参照してください。 また、CSIHnDLSx[3:0] ビットを 0001<sub>B</sub> (1 ビット) ~ 0110<sub>B</sub> (6 ビット) に設定するのは、ビット長が 16 ビット以上のときのみ可能です。</p>	CSIHn DLSx[3:0]	データ長	0000 <sub>B</sub>	16 ビット	0001 <sub>B</sub>	1 ビット	0010 <sub>B</sub>	2 ビット	---	---	1111 <sub>B</sub>	15 ビット								
CSIHn DLSx[3:0]	データ長																					
0000 <sub>B</sub>	16 ビット																					
0001 <sub>B</sub>	1 ビット																					
0010 <sub>B</sub>	2 ビット																					
---	---																					
1111 <sub>B</sub>	15 ビット																					
19	CSIHn RCBx	チップ・セレクト信号 x のブロードキャストの優先度の設定を選択します。 0: ドミナント (高優先度) 1: リセッシブ (低優先度) 詳細については、1314 ページの (1) 「コンフィギュレーション・レジスタ」を参照してください。																				

図 4.15 CSIHnCFGx レジスタのフォーマット (2/5)

表 22-28 CSIHnCFGx レジスタの内容 (2/4)

ビット位置	ビット名	機能																											
18	CSIHn DIRx	チップ・セレクト信号 x のシリアル・データ方向を選択します。 0: MSB ファーストでデータを送受信します。 1: LSB ファーストでデータを送受信します。 詳細については、1327 ページの 22.3.10 「シリアル・データ方向の選択」を参照してください。																											
17, 16	CSIHn CKPx  CSIHn DAPx	CKP: クロック位相選択ビット DKP: データ位相選択ビット  CSIHnCTL1.CSIHnCKR = 0 <table border="1" style="margin-top: 10px;"> <thead> <tr> <th>CSIHn CKPx</th> <th>CSIHn DAPx</th> <th>クロック位相とデータ位相の選択</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td> </td> </tr> <tr> <td>0</td> <td>1</td> <td> </td> </tr> <tr> <td>1</td> <td>0</td> <td> </td> </tr> <tr> <td>1</td> <td>1</td> <td> </td> </tr> </tbody> </table> CSIHnCTL1.CSIHnCKR = 1 <table border="1" style="margin-top: 10px;"> <thead> <tr> <th>CSIHn CKPx</th> <th>CSIHn DAPx</th> <th>クロック位相とデータ位相の選択</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td> </td> </tr> <tr> <td>0</td> <td>1</td> <td> </td> </tr> <tr> <td>1</td> <td>x</td> <td>設定禁止</td> </tr> </tbody> </table>	CSIHn CKPx	CSIHn DAPx	クロック位相とデータ位相の選択	0	0		0	1		1	0		1	1		CSIHn CKPx	CSIHn DAPx	クロック位相とデータ位相の選択	0	0		0	1		1	x	設定禁止
CSIHn CKPx	CSIHn DAPx	クロック位相とデータ位相の選択																											
0	0																												
0	1																												
1	0																												
1	1																												
CSIHn CKPx	CSIHn DAPx	クロック位相とデータ位相の選択																											
0	0																												
0	1																												
1	x	設定禁止																											
<p>注意 チップ・セレクトを行わない場合、CSIHnCKPx ビットは 0 固定とし、CSIHnCTL.CSIHnCKR ビットにて CSIHnCKPx ビットの設定値を代用してください。</p>																													

図 4.16 CSIHnCFGx レジスタのフォーマット (3/5)

表 22-28 CSIHnCFGx レジスタの内容 (3/4)

ビット位置	ビット名	機能												
15	CSIHn IDLx	<p>チップ・セレクト信号 x の強制アイドル状態の設定を選択します。</p> <p>0: チップ・セレクト値が変化しなかった場合、チップ・セレクト信号がアクティブな状態にとどまります。別のチップ・セレクト値が定義されると、チップ・セレクト信号 x はアイドル状態になります。</p> <p>1: チップ・セレクト信号 x に対応するスレーブヘータが 1 回転送されるたびにアイドル状態が挿入されます。</p> <p>このビットはマスタ・モードでのみ利用できます。 CSIHnCTL1.CSIHnJE=1 かつ CSIHnTX0W.CSIHnEOJ=1 のとき、たとえ CSIHnCFG0-7.CSIHnIDLn=0 に設定しても、アイドル状態は確実に挿入されます。アイドル状態については 1315 ページの「チップ・セレクトのタイミング」を参照してください</p>												
14 ~ 12	CSIHn IDx[2:0]	<p>チップ・セレクト信号 x のアイドル時間を選択します。</p> <table border="1"> <thead> <tr> <th>CSIHn IDx[2:0]</th> <th>アイドル時間</th> </tr> </thead> <tbody> <tr> <td>000<sub>B</sub></td> <td>0.5 シリアル・クロック周期</td> </tr> <tr> <td>001<sub>B</sub></td> <td>1.0 シリアル・クロック周期</td> </tr> <tr> <td>010<sub>B</sub></td> <td>1.5 シリアル・クロック周期</td> </tr> <tr> <td>...</td> <td>... (2.5, 3.5, 4.5, 6.5)</td> </tr> <tr> <td>111<sub>B</sub></td> <td>8.5 シリアル・クロック周期</td> </tr> </tbody> </table> <p>これらのビットはマスタ・モードでのみ利用可能です。</p>	CSIHn IDx[2:0]	アイドル時間	000 <sub>B</sub>	0.5 シリアル・クロック周期	001 <sub>B</sub>	1.0 シリアル・クロック周期	010 <sub>B</sub>	1.5 シリアル・クロック周期	...	... (2.5, 3.5, 4.5, 6.5)	111 <sub>B</sub>	8.5 シリアル・クロック周期
CSIHn IDx[2:0]	アイドル時間													
000 <sub>B</sub>	0.5 シリアル・クロック周期													
001 <sub>B</sub>	1.0 シリアル・クロック周期													
010 <sub>B</sub>	1.5 シリアル・クロック周期													
...	... (2.5, 3.5, 4.5, 6.5)													
111 <sub>B</sub>	8.5 シリアル・クロック周期													

図 4.17 CSIHnCFGx レジスタのフォーマット (4/5)

表 22-28 CSIHnCFGx レジスタの内容 (4/4)

ビット位置	ビット名	機能																					
11 ~ 8	CSIHn HDx[3:0]	チップ・セレクト信号 x のホールド時間をシリアル・クロック周期単位で指定します。																					
		<table border="1"> <thead> <tr> <th>CSIHn HDx[3:0]</th> <th>CSIHnCTL1.CSIHnSIT = 0 のときのホールド時間</th> <th>CSIHnCTL1.CSIHnSIT = 1 のときのホールド時間</th> </tr> </thead> <tbody> <tr> <td>0000<sub>B</sub></td> <td>0.5 シリアル・クロック周期</td> <td>1.0 シリアル・クロック周期</td> </tr> <tr> <td>0001<sub>B</sub></td> <td>1 シリアル・クロック周期</td> <td>1.5 シリアル・クロック周期</td> </tr> <tr> <td>0010<sub>B</sub></td> <td>1.5 シリアル・クロック周期</td> <td>2.0 シリアル・クロック周期</td> </tr> <tr> <td>...</td> <td>... (2.5, 3.5, 4.5, 6.5, 8.5, 9.5, 10.5, 11.5, 12.5, 14.5, 16.5, 18.5)</td> <td>... (3.0, 4.0, 5.0, 7.0, 9.0, 10.0, 11.0, 12.0, 13.0, 15.0, 17.0, 19.0)</td> </tr> <tr> <td>1111<sub>B</sub></td> <td>20.5 シリアル・クロック周期</td> <td>21.0 シリアル・クロック周期</td> </tr> </tbody> </table>	CSIHn HDx[3:0]	CSIHnCTL1.CSIHnSIT = 0 のときのホールド時間	CSIHnCTL1.CSIHnSIT = 1 のときのホールド時間	0000 <sub>B</sub>	0.5 シリアル・クロック周期	1.0 シリアル・クロック周期	0001 <sub>B</sub>	1 シリアル・クロック周期	1.5 シリアル・クロック周期	0010 <sub>B</sub>	1.5 シリアル・クロック周期	2.0 シリアル・クロック周期	...	... (2.5, 3.5, 4.5, 6.5, 8.5, 9.5, 10.5, 11.5, 12.5, 14.5, 16.5, 18.5)	... (3.0, 4.0, 5.0, 7.0, 9.0, 10.0, 11.0, 12.0, 13.0, 15.0, 17.0, 19.0)	1111 <sub>B</sub>	20.5 シリアル・クロック周期	21.0 シリアル・クロック周期			
		CSIHn HDx[3:0]	CSIHnCTL1.CSIHnSIT = 0 のときのホールド時間	CSIHnCTL1.CSIHnSIT = 1 のときのホールド時間																			
		0000 <sub>B</sub>	0.5 シリアル・クロック周期	1.0 シリアル・クロック周期																			
		0001 <sub>B</sub>	1 シリアル・クロック周期	1.5 シリアル・クロック周期																			
		0010 <sub>B</sub>	1.5 シリアル・クロック周期	2.0 シリアル・クロック周期																			
		...	... (2.5, 3.5, 4.5, 6.5, 8.5, 9.5, 10.5, 11.5, 12.5, 14.5, 16.5, 18.5)	... (3.0, 4.0, 5.0, 7.0, 9.0, 10.0, 11.0, 12.0, 13.0, 15.0, 17.0, 19.0)																			
1111 <sub>B</sub>	20.5 シリアル・クロック周期	21.0 シリアル・クロック周期																					
これらのビットはマスタ・モードでのみ利用可能です。																							
7 ~ 4	CSIHn INx[3:0]	チップ・セレクト信号 x のデータ間時間をシリアル・クロック周期単位で指定します。																					
		<table border="1"> <thead> <tr> <th>CSIHn INx[3:0]</th> <th>CSIHnCTL1.CSIHnSIT = 0 のときのデータ間時間</th> <th>CSIHnCTL1.CSIHnSIT = 1 のときのデータ間時間</th> </tr> </thead> <tbody> <tr> <td>0000<sub>B</sub></td> <td>0.0 シリアル・クロック周期</td> <td>0.5 シリアル・クロック周期</td> </tr> <tr> <td>0001<sub>B</sub></td> <td>0.5 シリアル・クロック周期</td> <td>1.0 シリアル・クロック周期</td> </tr> <tr> <td>0010<sub>B</sub></td> <td>1.0 シリアル・クロック周期</td> <td>1.5 シリアル・クロック周期</td> </tr> <tr> <td>0011<sub>B</sub></td> <td>2.0 シリアル・クロック周期</td> <td>2.5 シリアル・クロック周期</td> </tr> <tr> <td>...</td> <td>... (3.0, 4.0, 6.0, 8.0, 9.0, 10.0, 11.0, 12.0, 14.0, 16.0, 18.0)</td> <td>... (3.5, 4.5, 6.5, 8.5, 9.5, 10.5, 11.5, 12.5, 14.5, 16.5, 18.5)</td> </tr> <tr> <td>1111<sub>B</sub></td> <td>20.0 シリアル・クロック周期</td> <td>20.5 シリアル・クロック周期</td> </tr> </tbody> </table>	CSIHn INx[3:0]	CSIHnCTL1.CSIHnSIT = 0 のときのデータ間時間	CSIHnCTL1.CSIHnSIT = 1 のときのデータ間時間	0000 <sub>B</sub>	0.0 シリアル・クロック周期	0.5 シリアル・クロック周期	0001 <sub>B</sub>	0.5 シリアル・クロック周期	1.0 シリアル・クロック周期	0010 <sub>B</sub>	1.0 シリアル・クロック周期	1.5 シリアル・クロック周期	0011 <sub>B</sub>	2.0 シリアル・クロック周期	2.5 シリアル・クロック周期	...	... (3.0, 4.0, 6.0, 8.0, 9.0, 10.0, 11.0, 12.0, 14.0, 16.0, 18.0)	... (3.5, 4.5, 6.5, 8.5, 9.5, 10.5, 11.5, 12.5, 14.5, 16.5, 18.5)	1111 <sub>B</sub>	20.0 シリアル・クロック周期	20.5 シリアル・クロック周期
		CSIHn INx[3:0]	CSIHnCTL1.CSIHnSIT = 0 のときのデータ間時間	CSIHnCTL1.CSIHnSIT = 1 のときのデータ間時間																			
		0000 <sub>B</sub>	0.0 シリアル・クロック周期	0.5 シリアル・クロック周期																			
		0001 <sub>B</sub>	0.5 シリアル・クロック周期	1.0 シリアル・クロック周期																			
		0010 <sub>B</sub>	1.0 シリアル・クロック周期	1.5 シリアル・クロック周期																			
		0011 <sub>B</sub>	2.0 シリアル・クロック周期	2.5 シリアル・クロック周期																			
...	... (3.0, 4.0, 6.0, 8.0, 9.0, 10.0, 11.0, 12.0, 14.0, 16.0, 18.0)	... (3.5, 4.5, 6.5, 8.5, 9.5, 10.5, 11.5, 12.5, 14.5, 16.5, 18.5)																					
1111 <sub>B</sub>	20.0 シリアル・クロック周期	20.5 シリアル・クロック周期																					
これらのビットはマスタ・モードでのみ利用可能です。																							
3 ~ 0	CSIHn SPx[3:0]	チップ・セレクト信号 x のセットアップ時間をシリアル・クロック周期単位で指定します。																					
		<table border="1"> <thead> <tr> <th>CSIHn SPx[3:0]</th> <th>セットアップ時間</th> </tr> </thead> <tbody> <tr> <td>0000<sub>B</sub></td> <td>0.5 シリアル・クロック周期</td> </tr> <tr> <td>0001<sub>B</sub></td> <td>1.0 シリアル・クロック周期</td> </tr> <tr> <td>0010<sub>B</sub></td> <td>1.5 シリアル・クロック周期</td> </tr> <tr> <td>...</td> <td>... (2.5, 3.5, 4.5, 6.5, 8.5, 9.5, 10.5, 11.5, 12.5, 14.5, 16.5, 18.5)</td> </tr> <tr> <td>1111<sub>B</sub></td> <td>20.5 シリアル・クロック周期</td> </tr> </tbody> </table>	CSIHn SPx[3:0]	セットアップ時間	0000 <sub>B</sub>	0.5 シリアル・クロック周期	0001 <sub>B</sub>	1.0 シリアル・クロック周期	0010 <sub>B</sub>	1.5 シリアル・クロック周期	...	... (2.5, 3.5, 4.5, 6.5, 8.5, 9.5, 10.5, 11.5, 12.5, 14.5, 16.5, 18.5)	1111 <sub>B</sub>	20.5 シリアル・クロック周期									
		CSIHn SPx[3:0]	セットアップ時間																				
		0000 <sub>B</sub>	0.5 シリアル・クロック周期																				
		0001 <sub>B</sub>	1.0 シリアル・クロック周期																				
		0010 <sub>B</sub>	1.5 シリアル・クロック周期																				
...	... (2.5, 3.5, 4.5, 6.5, 8.5, 9.5, 10.5, 11.5, 12.5, 14.5, 16.5, 18.5)																						
1111 <sub>B</sub>	20.5 シリアル・クロック周期																						
これらのビットはマスタ・モードでのみ利用可能です。																							

図 4.18 CSIHnCFGx レジスタのフォーマット (5/5)

CSIHnCFG0 = 0x08000000;

/\* No parity \*/

/\* data length 8 bits \*/

/\* Recessive configuration: Dominant (higher priority) \*/

/\* MSB first \*/

## 4.2.6 CSIH メモリ制御レジスタ 0 (CSIHnMCTL0)

CSIHnMCTL0 レジスタではメモリ・モードとタイムアウトの設定を行うレジスタです。

CSIH はメモリ・モードとして、FIFO モード、デュアル・バッファ・モード、送信オンリー・バッファ・モード、ダイレクト・アクセス・モードの設定が可能です。

サンプルコードではデュアル・バッファ・モードのみを使用しています。またタイムアウト時間は検出しません。

アクセス 16ビット単位でリード/ライト可能です。

アドレス <CSIHn\_base\_OS> + 1040<sub>H</sub>

初期値 001F<sub>H</sub> どのリセット要因でも初期化されます。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	CSIHn MMS[1:0]		0	0	0	CSIHnTO[4:0]				
R	R	R	R	R	R	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

表 22-24 CSIHnMCTL0 レジスタの内容

ビット位置	ビット名	機能															
9~8	CSIHn MMS[1:0]	メモリ・モードを選択します。 <table border="1"> <thead> <tr> <th>CSIHn MMS1</th><th>CSIHn MMS0</th><th>説明</th></tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>FIFO モード</td></tr> <tr> <td>0</td><td>1</td><td>デュアル・バッファ・モード</td></tr> <tr> <td>1</td><td>0</td><td>送信オンリー・バッファ・モード</td></tr> <tr> <td>1</td><td>1</td><td>禁止</td></tr> </tbody> </table> <p>メモリ・モードを変更後、CSIHnSTCR0.CSIHnPCT ビットをセット (1) し個々のバッファ・ポインタなどをクリアしてください。            注意：            メモリ・モードは、CSIHnCTL0.CSIHnPWR = 0 かつ CSIHnCTL0.CSIHnMBS = 0 のときにのみ変更できます。</p>	CSIHn MMS1	CSIHn MMS0	説明	0	0	FIFO モード	0	1	デュアル・バッファ・モード	1	0	送信オンリー・バッファ・モード	1	1	禁止
CSIHn MMS1	CSIHn MMS0	説明															
0	0	FIFO モード															
0	1	デュアル・バッファ・モード															
1	0	送信オンリー・バッファ・モード															
1	1	禁止															
4~0	CSIHn TO[4:0]	FIFO モードのタイムアウトの設定を選択します。 <table border="1"> <thead> <tr> <th>CSIHnTO[4:0]</th><th>説明</th></tr> </thead> <tbody> <tr> <td>00000<sub>B</sub></td><td>タイムアウトを検出しません。</td></tr> <tr> <td>00001<sub>B</sub></td><td>タイムアウトを (1 × 8 × BRG 出力クロック) にします。</td></tr> <tr> <td>00010<sub>B</sub></td><td>タイムアウトを (2 × 8 × BRG 出力クロック) にします。</td></tr> <tr> <td>...</td><td></td></tr> <tr> <td>11111<sub>B</sub></td><td>タイムアウトを (31 × 8 × BRG 出力クロック) にします。</td></tr> </tbody> </table> <p>注意：            タイムアウトの設定は、CSIHnCTL0.CSIHnPWR = 0 のときにのみ変更することができます。            マスタ・モードのとき、またはメモリ・モードが FIFO モード以外 (ダイレクト・アクセス・モード、デュアル・バッファ・モード、送信モード) のときは、このビットに 00000<sub>B</sub> を設定してください。            タイムアウトの検出の詳細については、1345 ページの (3) 「タイムアウト・エラー」も参照してください。</p>	CSIHnTO[4:0]	説明	00000 <sub>B</sub>	タイムアウトを検出しません。	00001 <sub>B</sub>	タイムアウトを (1 × 8 × BRG 出力クロック) にします。	00010 <sub>B</sub>	タイムアウトを (2 × 8 × BRG 出力クロック) にします。	...		11111 <sub>B</sub>	タイムアウトを (31 × 8 × BRG 出力クロック) にします。			
CSIHnTO[4:0]	説明																
00000 <sub>B</sub>	タイムアウトを検出しません。																
00001 <sub>B</sub>	タイムアウトを (1 × 8 × BRG 出力クロック) にします。																
00010 <sub>B</sub>	タイムアウトを (2 × 8 × BRG 出力クロック) にします。																
...																	
11111 <sub>B</sub>	タイムアウトを (31 × 8 × BRG 出力クロック) にします。																

図 4.19 CSIHnMCTL0 レジスタのフォーマット

CSIH0MCTL0 = 0x0100;

/\* dual buffer mode; no timeout detection \*/

## 4.2.7 CSIH ステータス・クリア・レジスタ 0 (CSIHnSTCR0)

CSIH は、データ整合性エラー、パリティ・エラー、オーバーラン・エラー、タイムアウト・エラー、オーバフロー・エラーの 5 種類のエラーを検出することができます。

パリティ・エラー、データ整合性エラー、タイムアウト・エラーは CSIHnSTCR0 レジスタで個別に有効または無効にすることができます。いずれかのエラーが検出されると、受信エラー割り込み CSIHnTIRE が発生します。

サンプルコードでは、受信エラー割り込み CSIHnTIRE が検出された場合、ステータス・クリア・レジスタの各ビットを 1 にセットし、エラー・フラグをクリアします。

アクセス 16 ビット単位でライト可能です。

リードすると、常に値 0000<sub>H</sub> が返されます。

アドレス <CSIHn\_base\_USER> + 0008<sub>H</sub>

初期値 0000<sub>H</sub> どのリセット要因でも初期化されます。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSIHnTMOEC	CSIHnOFEC	0	0	0	0	0	CSIHnPCT	0	0	0	0	CSIHnDCEC	0	CSIHnPEC	CSIHnOVEC
W	W	R	R	R	R	R	W	R	R	R	R	W	R	W	W

表 22-23 CSIHnSTCR0 レジスタの内容

ビット位置	ビット名	機能
15	CSIHnTMOEC	タイムアウト・エラー・フラグ・クリア・コマンドを制御します。 0: 何も操作を行いません。リード値は常に 0 になります。 1: タイムアウト・エラー・フラグ (CSIHnSTR0.CSIHnTMOE) をクリアします。
14	CSIHnOFEC	オーバフロー・エラー・フラグ・クリア・コマンドを制御します。 0: 何も操作を行いません。リード値は常に 0 になります。 1: オーバフロー・エラー・フラグ (CSIHnSTR0.CSIHnOFE) をクリアします。
8	CSIHnPCT	FIFO ポインタを制御します。 0: 何も操作を行いません。リード値は常に 0 になります。 1: デュアル・バッファ・モード、送信オンリー・バッファ・モード、FIFO モードの場合、以下のすべての FIFO バッファ・ポインタをクリアします。 - CSIHnMRWP0.CSIHnTRWA[6:0] - CSIHnMRWP0.CSIHnRRA[6:0] - CSIHnMCTL2.CSIHnSOP[6:0] FIFO モードの場合のみ、さらに以下のすべてのステータス・ビットをクリアします。 - CSIHnSTR0.CSIHnSPF[7:0] - CSIHnSTR0.CSIHnSRP[7:0] - CSIHnSTR0.CSIHnFLF - CSIHnSTR0.CSIHnTSF また、CSIHnSTR0.CSIHnEMF が 1 にセットされます (FIFO エンプティ)。  注意: 通信中にこのビットがセットされると、実行中の通信は停止されます。
3	CSIHnDCEC	データー貫性エラー・フラグ・クリア・コマンドを制御します。 0: 何も操作を行いません。リード値は常に 0 になります。 1: データー貫性エラー・フラグ (CSIHnSTR0.CSIHnDCE) をクリアします。
1	CSIHnPEC	パリティ・エラー・フラグ・クリア・コマンドを制御します。 0: 何も操作を行いません。リード値は常に 0 になります。 1: パリティ・エラー・フラグ (CSIHnSTR0.CSIHnPE) をクリアします。
0	CSIHnOVEC	オーバーラン・エラー・フラグ・クリア・コマンドを制御します。 0: 何も操作を行いません。リード値は常に 0 になります。 1: オーバーラン・エラー・フラグ (CSIHnSTR0.CSIHnOVE) をクリアします。

図 4.20 CSIHnSTCR0 レジスタのフォーマット

```
/* Clear status flags to 0 */
CSIHnTMOEC = 1;      /* Clear timeout error flag */
CSIHnOFEC = 1;       /* Clear overflow error flag */
CSIHnPEC = 1;        /* Clear parity error flag */
CSIHnOVEC = 1;       /* Clear overrun error flag */
CSIHnPCT = 1;        /* Clear FIFO buffer pointer */
CSIHnDCEC = 1;       /* Clear data consistency error flag */
```

### 4.3 メモリ・モード

CSIH はメモリ・モードとして、FIFO モード、デュアル・バッファ・モード、送信オンリー・バッファ・モード、ダイレクト・アクセス・モードを備えています。メモリ・モードは CSIHnMCTL0.CSIHnMMS[1:0] を設定することで変更可能です。CSIH のデータ転送が開始する条件、割り込みタイミングなどは、メモリ・モード、動作モード、転送モードによって異なります。

表 4.1 データ転送の開始

メモリ・モード 動作モード		転送モード	
		送信専用、送受信	受信専用
FIFO、 ダイレクト・ アクセス	マスタ	CSIHnTXO レジスタへ書き込み	CSIHnTXO レジスタへ書き込み
	スレーブ	CSIHnTXO レジスタへ書き込みと マスタクロックの起動	マスタからのクロックの受信
送信オンリー・ バッファ	マスタ	CSIHnMCTL2.BTST=1	CSIHnMCTL2.BTST=1
	スレーブ	CSIHnMCTL2.BTST=1 かつ マスタクロックの起動	マスタからのクロックの受信
デュアル・ バッファ	マスタ	CSIHnMCTL2.BTST=1	CSIHnMCTL2.BTST=1
	スレーブ	CSIHnMCTL2.BTST=1 かつ マスタクロックの起動	マスタからのクロックの受信



## 4.4 関数仕様

今回のプログラムで使用する関数の仕様について説明します。

### 4.4.1 メイン (main.c)

【関数名】	main ()
【機能】	各初期化処理関数を呼び出したあと、永久ループに入る
【引数】	なし
【戻り値】	なし
【起動方法】	ハードウェア初期化完了したあと自動的にメイン関数に入る
【使用 SFR】	なし
【call 関数】	なし
【変数】	flag_mode, flag_transmit_over, flag_receive_over, flag_error
【ファイル名】	main.c
【注意事項】	なし

### 4.4.2 初期化処理 (initial.c)

【関数名】	port_initial ()
【機能】	ポートのモード設定を行う
【引数】	なし
【戻り値】	なし
【起動方法】	コールする
【使用 SFR】	PFCE4, PFC4, PMC4, PM4, PFCE13, PFC13, PMC13, PM13
【call 関数】	main ()
【変数】	なし
【ファイル名】	initial.c
【注意事項】	なし

【関数名】	cg_initial ()
【機能】	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】	なし
【戻り値】	なし
【起動方法】	コールする
【使用 SFR】	SFRCTL3
【call 関数】	main ()
【変数】	なし
【ファイル名】	initial.c
【注意事項】	なし

【関数名】 hbus\_initial ()  
【機能】 AHB バスの初期化設定を行う  
【引数】 なし  
【戻り値】 なし  
【起動方法】 コールする  
【使用 SFR】 ETARCFG0, ETARADRS0, ETARMASK0  
【call 関数】 main ()  
【変数】 なし  
【ファイル名】 initial.c  
【注意事項】 なし

【関数名】 board\_initial ()  
【機能】 LED 初期状態の設定を行う  
【引数】 なし  
【戻り値】 なし  
【起動方法】 コールする  
【使用 SFR】 P13  
【call 関数】 main ()  
【変数】 なし  
【ファイル名】 initial.c  
【注意事項】 なし

【関数名】 ram\_initial ()  
【機能】 受信バッファと各フラグ初期状態の設定を行う  
【引数】 なし  
【戻り値】 なし  
【起動方法】 コールする  
【使用 SFR】 なし  
【call 関数】 main ()  
【変数】 buf\_receive[], flag\_transmit\_over, flag\_receive\_over, flag\_job\_transmit, flag\_error, flag\_fifo\_error, count, LED, point\_receive, point\_transmit  
【ファイル名】 initial.c  
【注意事項】 なし

【関数名】 wait ()  
【機能】 一定ステップを待つ  
【引数】 int number  
【戻り値】 なし  
【起動方法】 引数設定でコールする  
【使用 SFR】 なし  
【call 関数】 main () , csih\_transmit\_1\_start ()  
【変数】 なし  
【ファイル名】 initial.c  
【注意事項】 なし

**【関数名】** display ()  
**【機能】** 各フラグの状態により、LED を制御する  
**【引数】** なし  
**【戻り値】** なし  
**【起動方法】** コールする  
**【使用 SFR】** P13  
**【call 関数】** main ()  
**【変数】** flag\_transmit\_over, flag\_receive\_over, flag\_error  
**【ファイル名】** initial.c  
**【注意事項】** なし

**【関数名】** clear\_receive\_buffer ()  
**【機能】** 受信バッファをクリアする  
**【引数】** なし  
**【戻り値】** なし  
**【起動方法】** コールする  
**【使用 SFR】** なし  
**【call 関数】** csih\_receive\_start()  
**【変数】** point\_receive, buf\_receive  
**【ファイル名】** initial.c  
**【注意事項】** なし

#### 4.4.3 受信 (csih\_receive.c)

**【関数名】** csih\_receive\_initial ()  
**【機能】** 通信モードフラグによる、サブルーチンを選ぶ  
**【引数】** なし  
**【戻り値】** なし  
**【起動方法】** コールする  
**【使用 SFR】** なし  
**【call 関数】** main ()  
**【変数】** flag\_mode  
**【ファイル名】** csih\_receive.c  
**【注意事項】** なし

**【関数名】** csih\_receive\_1\_initial ()  
**【機能】** CSIH3 マクロはダイレクト・アクセス受信モードで初期化設定を行う  
**【引数】** なし  
**【戻り値】** なし  
**【起動方法】** コールする  
**【使用 SFR】** CSIH3CTL0, CSIH3CTL1, CSIH3CTL2, CSIH3CFG0, ICCSIH3IR  
**【call 関数】** csih\_receive\_initial ()  
**【変数】** なし  
**【ファイル名】** csih\_receive.c  
**【注意事項】** なし

【関数名】	csih_receive_2_initial ()
【機能】	CSIH3 マクロはデュアル・バッファ受信モードで初期化設定を行う
【引数】	なし
【戻り値】	なし
【起動方法】	コールする
【使用 SFR】	CSIH3CTL0, CSIH3CTL1, CSIH3CTL2, CSIH3CFG0, ICCSIH3IR
【call 関数】	csih_receive_initial ()
【変数】	なし
【ファイル名】	csih_receive.c
【注意事項】	なし

【関数名】	csih_receive_start ()
【機能】	CSIH3 マクロはデュアル・バッファ受信モードで受信動作を行う。
【引数】	なし
【戻り値】	なし
【起動方法】	コールする
【使用 SFR】	CSIH3PWR, CSIH3MCTL2
【call 関数】	main ()
【変数】	なし
【ファイル名】	csih_receive.c
【注意事項】	なし

#### 4.4.4 送信 (csih\_transmit.c)

【関数名】	csih_transmit_initial ()
【機能】	通信モードフラグによる、サブルーチンを選ぶ
【引数】	なし
【戻り値】	なし
【起動方法】	コールする
【使用 SFR】	なし
【call 関数】	main ()
【変数】	flag_mode
【ファイル名】	csih_transmit.c
【注意事項】	なし

【関数名】	csih_transmit_1_initial ()
【機能】	CSIH0 マクロはダイレクト・アクセス送信モードで初期化設定を行う
【引数】	なし
【戻り値】	なし
【起動方法】	コールする
【使用 SFR】	CSIH0CTL0, CSIH0CTL1, CSIH0CTL2, CSIH0CFG0, ICCSIH0IC, ICCSIH0JC
【call 関数】	csih_transmit_initial ()
【変数】	なし
【ファイル名】	csih_transmit.c
【注意事項】	なし

**【関数名】** csih\_transmit\_2\_initial ()  
**【機能】** CSIH0 マクロはデュアル・バッファ送信モードで初期化設定を行う  
**【引数】** なし  
**【戻り値】** なし  
**【起動方法】** コールする  
**【使用 SFR】** CSIH0CTL0, CSIH0CTL1, CSIH0CTL2, CSIH0CFG0, ICCSIH0IC, CSIH0MCTL0, CSIH0MCTL2  
**【call 関数】** csih\_transmit\_initial ()  
**【変数】** なし  
**【ファイル名】** csih\_transmit.c  
**【注意事項】** なし

**【関数名】** csih\_transmit\_start ()  
**【機能】** 通信モードフラグによる、サブルーチンを選ぶ  
**【引数】** なし  
**【戻り値】** なし  
**【起動方法】** コールする  
**【使用 SFR】** なし  
**【call 関数】** main ()  
**【変数】** flag\_mode  
**【ファイル名】** csih\_transmit.c  
**【注意事項】** なし

**【関数名】** csih\_transmit\_1\_start ()  
**【機能】** CSIH0 マクロはダイレクト・アクセス送信モードで送信動作を行う  
**【引数】** なし  
**【戻り値】** なし  
**【起動方法】** コールする  
**【使用 SFR】** CSIH0CTL0, CSIH3CTL0, CSIH0TX0W  
**【call 関数】** csih\_transmit\_start ()  
**【変数】** flag\_job\_transmit  
**【ファイル名】** csih\_transmit.c  
**【注意事項】** なし

**【関数名】** csih\_transmit\_2\_start ()  
**【機能】** CSIH0 マクロはデュアル・バッファ送信モードで送信動作を行う  
**【引数】** なし  
**【戻り値】** なし  
**【起動方法】** コールする  
**【使用 SFR】** CSIH0CTL0, CSIH3CTL0, CSIH0TX0W, CSIH0TX0H, CSIH0MCTL2  
**【call 関数】** csih\_transmit\_start ()  
**【変数】** point\_transmit  
**【ファイル名】** csih\_transmit.c  
**【注意事項】** なし

## 4.4.5 割り込み (interrupt.c)

【関数名】	int_csih0ic ()
【機能】	CSIH0 マクロの送信ステータス割り込み処理を行う
【引数】	なし
【戻り値】	なし
【起動方法】	マスクをされていない状態で、CSIH0TIC 要求がある
【使用 SFR】	CSIH0CTL0
【call 関数】	なし
【変数】	flag_mode, flag_job_transmit, flag_transmit_over
【ファイル名】	interrupt.c
【注意事項】	なし

【関数名】	int_csih0ijc ()
【機能】	CSIH0 マクロのジョブ完了割り込み処理を行う
【引数】	なし
【戻り値】	なし
【起動方法】	マスクをされていない状態で、CSIH0TIJC 要求がある
【使用 SFR】	CSIH0CTL0
【call 関数】	なし
【変数】	flag_mode, flag_transmit_over
【ファイル名】	interrupt.c
【注意事項】	なし

【関数名】	int_csih0ire ()
【機能】	CSIH0 マクロの通信エラー割り込み処理を行う
【引数】	なし
【戻り値】	なし
【起動方法】	マスクをされていない状態で、CSIH0TIRE 要求がある
【使用 SFR】	CSIH0STCR0, CSIH0CTL0
【call 関数】	なし
【変数】	flag_error, point_receive, point_transmit
【ファイル名】	interrupt.c
【注意事項】	なし

【関数名】	int_csih3ir ()
【機能】	CSIH3 マクロの受信ステータス割り込み処理を行う
【引数】	なし
【戻り値】	なし
【起動方法】	マスクをされていない状態で、CSIH3TIR 要求がある
【使用 SFR】	CSIH3RX0H, CSIH3CTL0
【call 関数】	なし
【変数】	flag_mode, point_receive, count, flag_receive_over
【ファイル名】	interrupt.c
【注意事項】	なし

【関数名】	int_csih3ire ()
【機能】	CSIH3 マクロの通信エラー割り込み処理を行う
【引数】	なし
【戻り値】	なし
【起動方法】	マスクをされていない状態で、CSIH3TIRE 要求がある
【使用 SFR】	CSIH3STCR0, CSIH3CTL0
【call 関数】	なし
【変数】	flag_error, point_receive, point_transmit, count
【ファイル名】	interrupt.c
【注意事項】	なし

## 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/>

お問い合わせ先

<http://japan.renesas.com/inquiry>

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。



## 改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2012.02.10	—	初版発行

## 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っていると、内部メモリ、レイアウトパターンの相違などにより、特性が異なる場合があります。型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。

## ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連して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、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とその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

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



ルネサス エレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

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

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

(03)5201-5307

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