RL78/G13

IIC マルチマスタ通信（管理マスタ）CC-RL

R01AN4709JJ0100
Rev. 1.00
2019.12.20

要旨

本アプリケーションノートは、“アクセス権”を用いてスレーブへのアクセスを排他制御する IIC マルチマスタ通信について説明します。 IIC マルチマスタ通信システムは、アクセス権を管理しているマスタ（管理マスタ）、アクセス権を要求するマスタ（クライアント）、スレーブで構成されます。

本アプリケーションノートでは、アクセス権を管理しているマスタのプログラムについて説明します。

対象デバイス

RL78/G13

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

本アプリケーションノートでは、スレーブへのアクセスを排他制御する IIC マルチマスタ通信について説明します。 本 IIC マルチマスタ通信システムは、1 台のアクセス権を管理しているマスタ（管理マスタ）、m 台のアクセス権を要求するマスタ（クライアント）、n 台のスレーブで構成されます（図 1.1参照）。

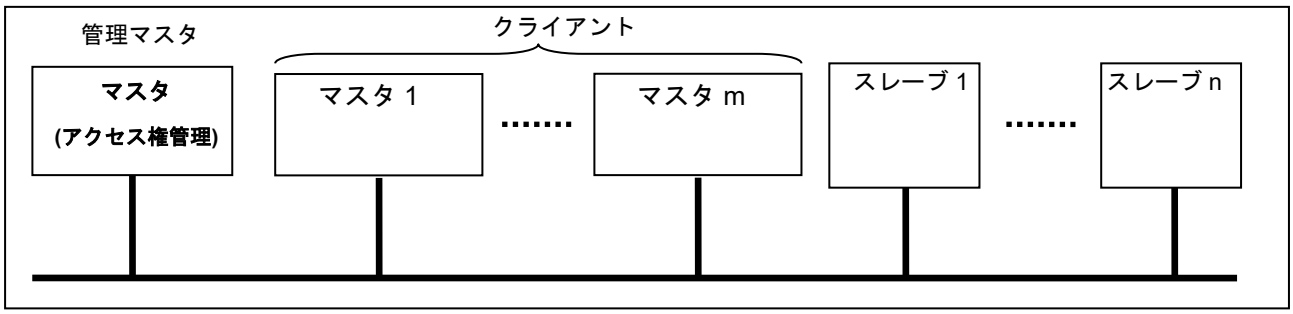


図 1.1 IIC マルチマスタ通信システムの構成例

1.1 排他制御

1.1.1 アクセス権

複数のマスタが存在する IIC 通信では、複数のマスタが同時に IIC バスにアクセスすることがあり得ます。しかし、IIC バス通信では、スレーブにアクセスできるマスタは 1 つに限られます。そのため、複数のマスタからの同時アクセスによる競合を防ぐために、あるマスタがスレーブにアクセスしている期間は、他のマスタがスレーブにアクセスしないようにする排他制御が必要になります。

本アプリケーションでは、スレーブへのアクセス権を 1 台のマスタが管理してアクセス権を取得したマスタ（クライアント）のみがスレーブにアクセスすることで、複数のマスタからの同時アクセスによる競合を防ぎます。

1.1.2 アクセス権を利用した排他制御

通常の IIC マルチマスタ通信では、複数のマスタが同時に複数のスレーブにアクセスした場合、スレーブ・アドレスや通信データによって IIC バス通信のアービトレーションが行われ、IIC バスの使用権が決定されます。つまり、複数のマスタが同時にスレーブにアクセスした場合、特定のスレーブや特定の通信データが優先されることになり、大きな数値のアドレスを持つスレーブへのアクセスは待たされることになります。

本アプリケーションでは、スレーブまたはスレーブ機能をもったマスタ（クライアント）にアクセスするために、マスタ（クライアント）は必ず、管理マスタからアクセス権を取得しなければなりません。

アクセス権を得ていないマスタ（クライアント）は、IIC バス通信のアービトレーションによって IIC バスの使用権を得ていても、スレーブへのアクセスは禁止とします。IIC バスの使用権を誤って取得した場合は、速やかに IIC バスを解放しなければなりません。さらに、管理マスタからアクセス権を得ているマスタ（クライアント）の処理が完了するまで、ウエイトしなければなりません。

また、すべてのマスタ（管理マスタ、クライアント）は IIC バス通信の転送クロックを同じにしなければなりません。これは、転送クロック起因のアービトレーションを避けるためです。

1.1.3 アクセス権の取得制御

クライアントは、スレーブへのアクセス権を取得するために、スタート・コンディション生成後、管理マスタのアドレスに続いて、要求元アドレスとして自分のアドレスとその反転データを管理マスタに送信します。反転データを送信した時の管理マスタからの ACK 応答でアクセス権を取得したと判断します。NACK 応答の場合は、アクセス権が取得できなかったと判断します。

なお、アクセス権の取得要求は、要求元アドレス送信に合わせて行います。要求元アドレス送信の 1～7 クロックの期間は、アドレス A6 から A0 まで送信（MSB ファースト）し、8 クロック目のデータとして “0”（アクセス権の取得要求）を送信します。

表 1.1 アクセス権取得の成功／失敗

クライアントの送信データ	管理マスタの応答	
管理マスタのアドレス	ACK応答	ACK／NACK応答
クライアントのアドレス（要求元アドレス）	ACK応答	ACK／NACK応答
クライアントのアドレスの反転データ	ACK応答	NACK応答
アクセス権の取得	**成功**	**失敗**

図 1.2にアクセス権の取得に成功した例を示します。

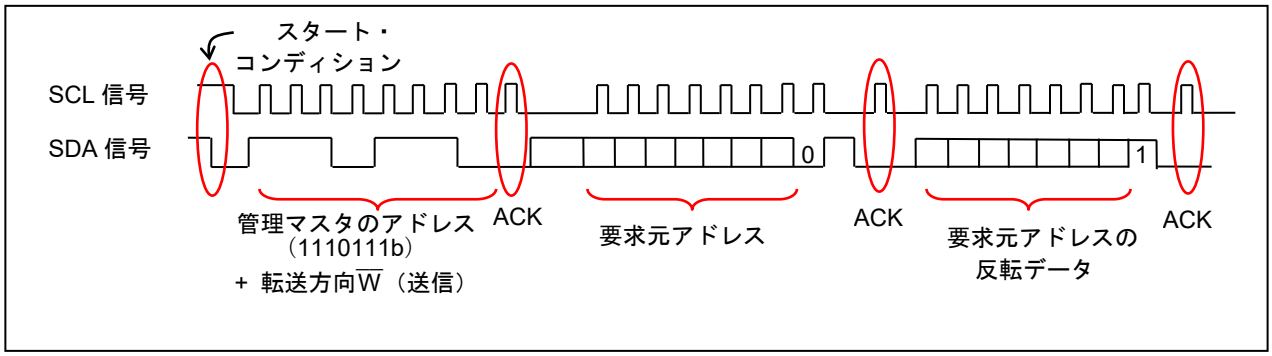


図 1.2 アクセス権取得に成功した例

複数のマスタ（クライアント）が同時にアクセス権を取得しようとした場合では、下記のタイミングで IIC バスのアービトレーションが発生します。

(1) スタート・コンディション生成時

スタート・コンディション生成の遅いマスタがアービトレーション負けになります。しかし、タイミングの差が小さい場合は、複数のマスタがアービトレーション負けを検出できないことが考えられます。

(2) 管理マスタのアドレス送信時

管理マスタのアドレスは、クライアント、スレーブのアドレスよりも大きな数値（1110111b）にしています。そのため、管理マスタとスレーブへのアクセスが競合した場合は、管理マスタにアクセスしていたクライアントがアービトレーション負けになります。これにより、アクセス権を取得しているクライアントによるスレーブへのアクセスが優先されます。

なお、同時にアクセス権を要求した場合は、アービトレーション負けは検出されません。

(3) 要求元アドレスの送信時

各クライアントのアドレスでアービトレーションが実行されます。スタート・コンディション生成から要求元アドレスの転送までに、アービトレーション負けになかったクライアントは、IIC バスの使用権を確保します。しかし、管理マスタからアクセス権を取得するまでは、スレーブへのアクセスは禁止です。アクセス権の取得は、要求アドレスの反転データ送信時の管理マスタからの ACK 応答／NACK 応答で決まります。

まず、クライアントは、アクセス権を取得するために、スタート・コンディション生成後、管理マスタのアドレスを送信します。

管理マスタからの ACK 応答を確認後、要求元アドレスとして自分のアドレスを管理マスタに送信します。要求元アドレス送信において、要求元アドレス送信の 1～7 クロックの期間は、アドレス A6 から A0 まで送信（MSB ファースト）し、8 クロック目のデータとして“0”（アクセス権の取得要求）を送信します。

管理マスタからの ACK 応答を確認後、クライアントは、要求元アドレスの反転データを送信します。これは、要求元アドレスの確からしさを確認する処理です。送信された要求元アドレスとその反転データに矛盾がなく、どのクライアントもアクセス権を取得していなければ、管理マスタは ACK 応答をします（アクセス権取得成功）。前述以外の状態であれば、管理マスタは NACK 応答をします（アクセス権取得失敗）。

アクセス権取得に成功したクライアントは、スタート・コンディションを生成して、アクセスしたいスレーブへのアクセスを開始します。

図 1.3に IIC バスのアービトレーション負けにならなかったが、アクセス権の取得に失敗した例を示します。

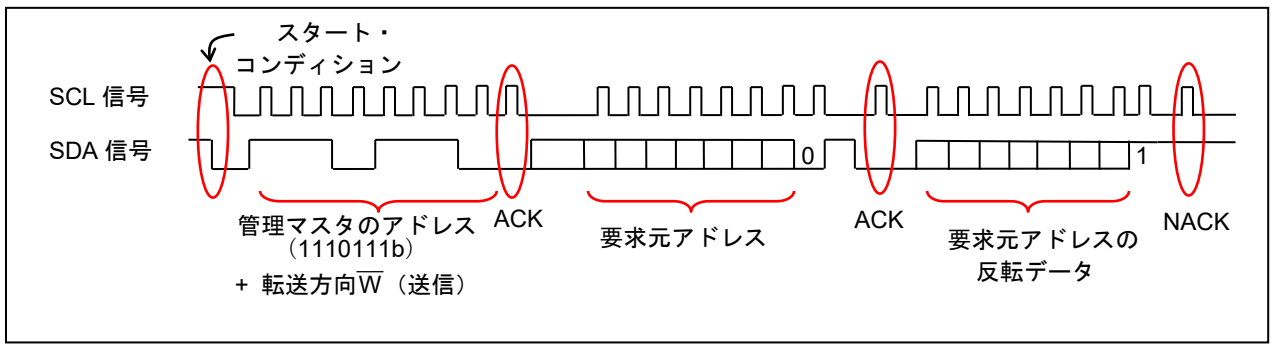


図 1.3 アクセス権取得に失敗した例

前述の(1)～(3)で IIC バスのアービトレーションに負けたクライアントおよびアクセス権の取得に失敗したクライアントは、一定時間は IIC バスにアクセスしないようにします。これにより、IIC バスのアクセス競合が発生する頻度を下げて、転送効率の低下を防ぎます。

1.1.4 アクセス権の解放制御

アクセス権の解放要求は、アクセス権を取得したクライアントのみが実行できます。

クライアントは、アクセス権を解放するために、スタート・コンディション生成後、管理マスタのアドレスに続いて、要求元アドレスとして自分のアドレスとその反転データを管理マスタに送信します。反転データを送信した時の管理マスタからの ACK 応答でアクセス権を解放できたと判断します。NACK 応答の場合は、アクセス権を解放できなかったと判断します。

要求元アドレス送信において、要求元アドレス送信の 1～7 クロックの期間は、アドレス A6 から A0 まで送信（MSB ファースト）し、8 クロック目のデータとして“1”（アクセス権の解放要求）を送信します。

表 1.2 アクセス権解放の成功／失敗

クライアントの送信データ	管理マスタの応答	
管理マスタのアドレス	ACK応答	ACK／NACK応答
クライアントのアドレス（要求元アドレス）	ACK応答	ACK／NACK応答
クライアントのアドレスの反転データ	ACK応答	NACK応答
アクセス権の解放	**成功**	**失敗**

図 1.4にアクセス権の解放に成功した例を示します。

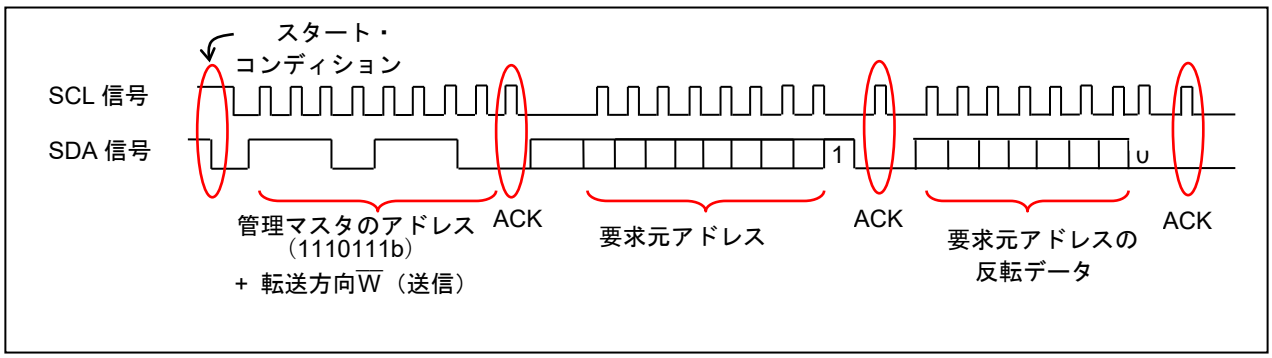


図 1.4 アクセス権解放に成功した例

1.1.5 アクセス権の取得／解放動作

(1) 複数のアクセス権の取得処理の競合動作

複数のクライアントが同時にアクセス権の取得処理を行った場合は、管理マスタとクライアントの処理は図 1.5のようになります。なお、クライアント１のアドレスは、クライアント２のアドレスより小さいとします。

①と②でクライアント１とクライアント２が同時にアクセス権の取得処理を行うと、アドレスが小さいクライアント１がアービトレーションに勝ちます。アービトレーションに勝ったクライアント１はスレーブにアクセスした後、③でアクセス権の解放処理を行います。クライアント２は、アービトレーションに負けた後、一定時間ウエイトします。その後、④で再度アクセス権の取得処理を行います。

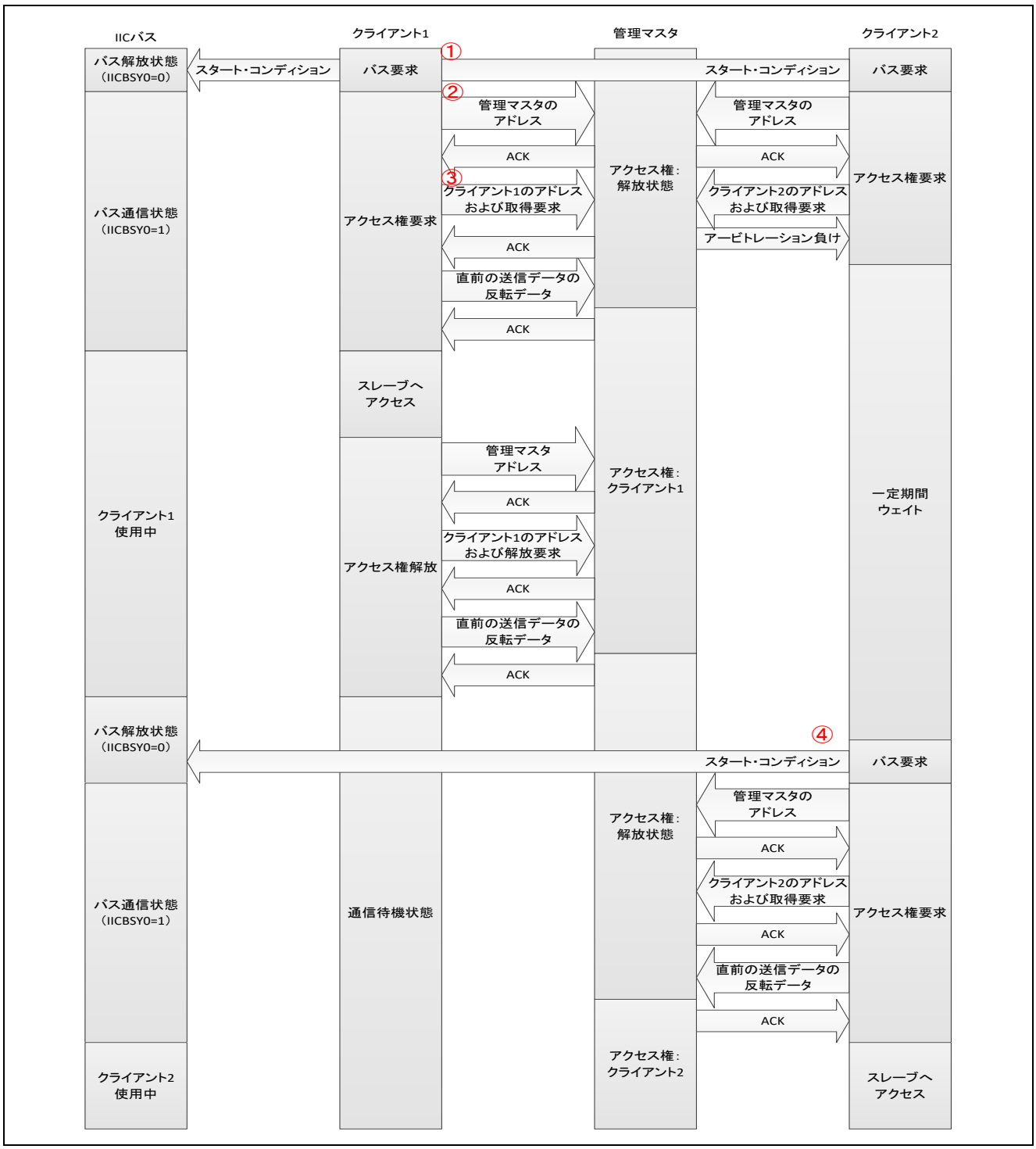


図 1.5 アクセス権取得時の競合動作

(2) アクセス権の取得処理と解放処理の競合動作

アクセス権の取得処理と解放処理が競合した場合の動作は図 1.6のようになります。

クライアント 2 がスレーブとのアクセスが終わり、①でストップ・コンディションを生成すると、IIC バスは解放状態になります。このとき、②と③でクライアント 1 のアクセス権の取得処理とクライアント 2 のアクセス権の解放処理が同時に発生した場合は､IIC バスのアドレスが小さいクライアント 1 がアービトレーションに勝ちます。しかし、クライアント 2 がアクセス権を保持しているため、クライアント 1 が管理マスタに次のデータを送信しても NACK 応答となります。

NACK 応答となったため、④のようにクライアント 1 はストップ・コンディションを生成して IIC バスを解放しなければなりません。IIC バスが解放されると、⑤のようにクライアント 2 は再度アクセス権解放処理を行います。クライアント 1 は一定期間経過後、⑥のように再びアクセス権の取得動作を行います。

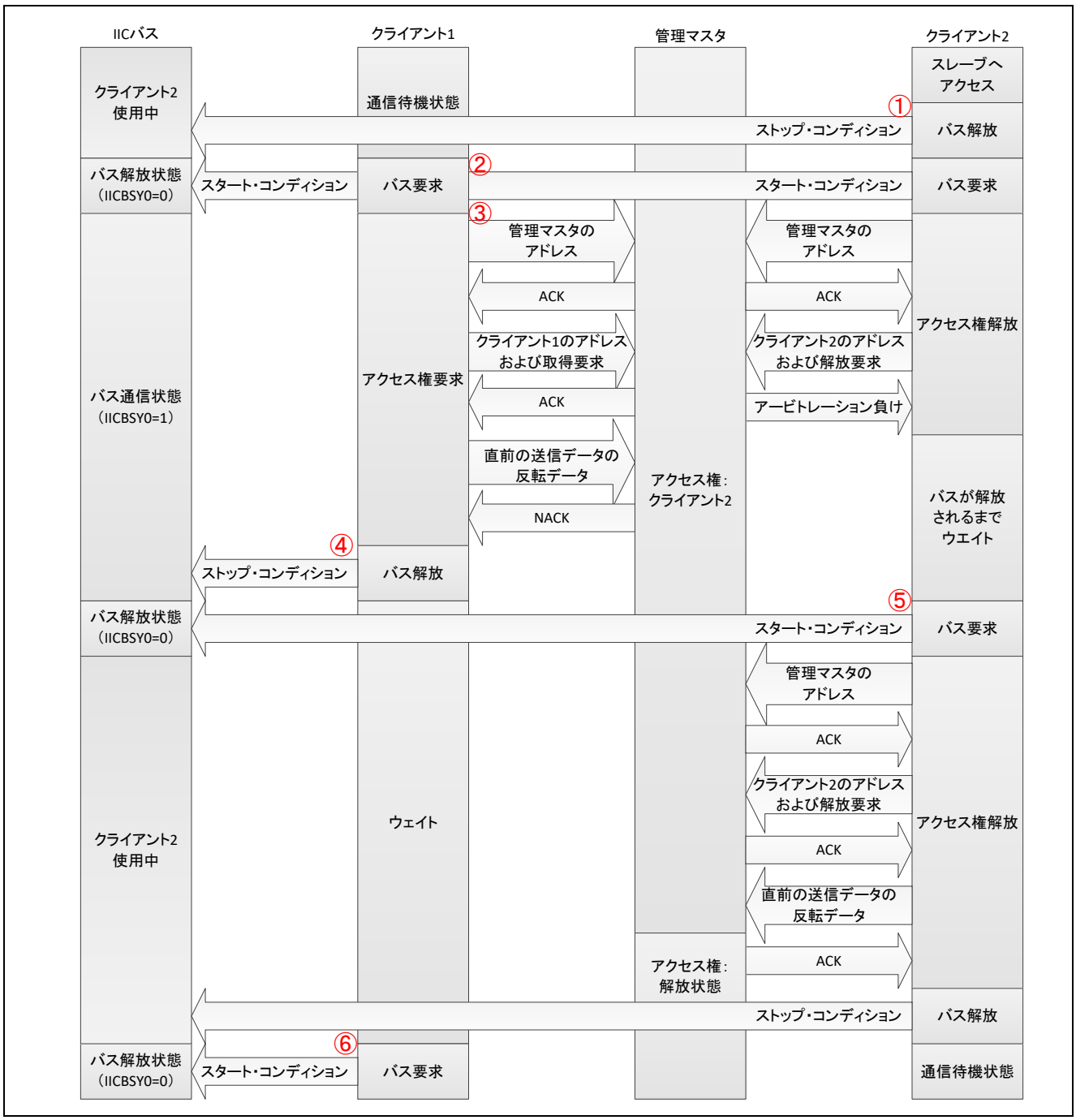


図 1.6 アクセス権解放時の競合動作

1.2 管理マスタの動作

1.2.1 アクセス権の制御フロー

図 1.7と図 1.8に管理マスタのアクセス権を制御する簡略フローを示します。シリアル・インタフェース IICA の割り込み処理でアクセス権の制御を行います。

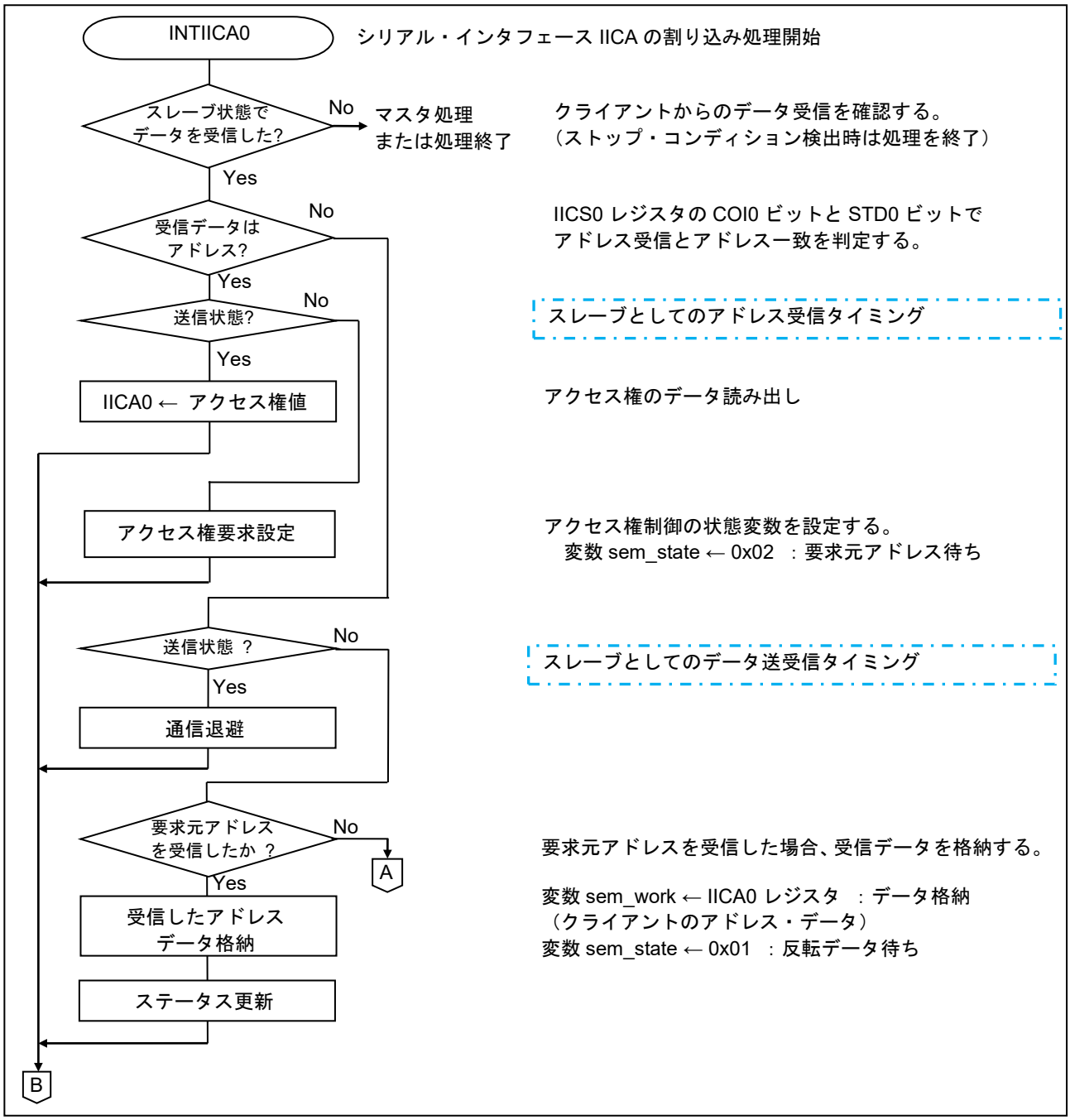


図 1.7 管理マスタのアクセス権制御フロー（1/2）

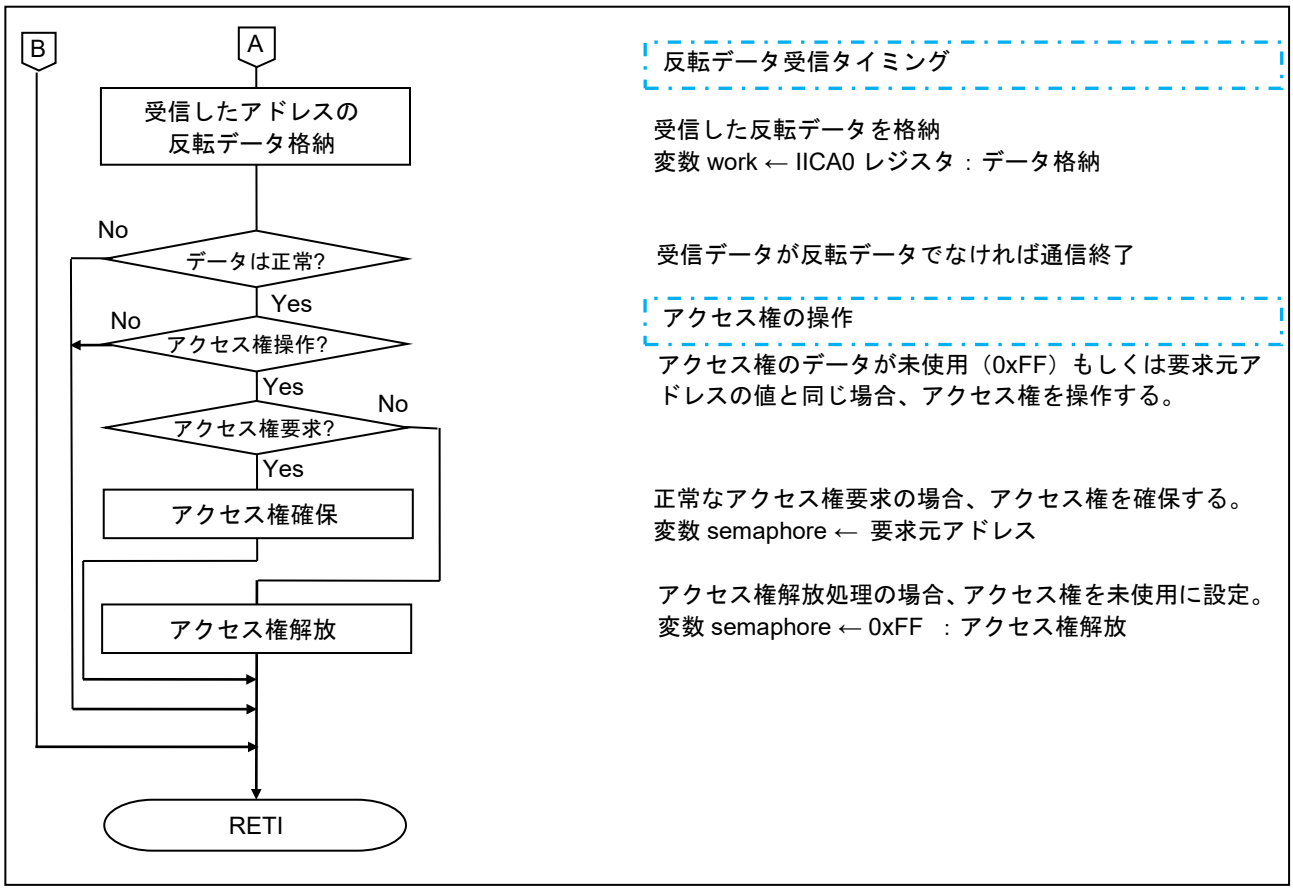


図 1.8 管理マスタのアクセス権制御フロー（2/2）

1.2.2 管理マスタのマスタとしての排他制御

管理マスタは、自分自身がマスタとしてスレーブをアクセスするときにもアクセス権の制御が必要です。アクセス権の優先順位は、アクセス権の管理者である管理マスタが最上位とします。

アクセス権（ソフトウェアでは変数：semaphore）の値が FFH で、IIC バスがビジーではない（IICBSY0 ビット＝0）場合、IIC バスは使用されていません。このとき、管理マスタは、アクセス権（semaphore）を自身のアドレスを示す“11101110b”に変更して IIC バスのスレーブに対するアクセスを開始します。

IIC バスのマスタ（MASTS0＝1）となった後は、通常のマスタ動作と同じです。他のマスタの IIC バスへのアクセスを禁止するには、できるだけストップ・コンディションを生成しないようにします。

アクセス権（semaphore）の更新時間を不用意に増大さないために、IICBSY0 ビットの確認の直前からアクセス権の書き換えの直後まで割り込み禁止にします。

クライアントがアクセス権を確保している場合や IIC バスがビジーの場合は、一定期間経過した後に再びアクセス権取得処理をやり直します。

アクセス権を取得していても、IIC バスへのアクセスで管理マスタがアービトレーション負けとなる場合があります。このとき、管理マスタは必ずスレーブになります。しかし、アクセス権を取得していないクライアントは直ぐに IIC バスを解放します。IIC バスが解放されたら、管理マスタはスタート・コンディション生成から再び処理を行います。なお、管理マスタがアービトレーション負けとなるのは、スタート・コンディション生成時のみです。管理マスタのアドレスは、クライアント、スレーブのアドレスよりも大きな数値（1110111b）のため、管理マスタとスレーブへのアクセスが競合した場合は、管理マスタにアクセスしていたクライアントがアービトレーション負けになります。

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	RL78/G13（R5F100LEA）
動作周波数	● 高速オンチップ・オシレータ（HOCO）クロック：32MHz ● CPU／周辺ハードウエア・クロック： 32MHz
動作電圧	5.0V（2.7V～5.5V で動作可能） LVD 動作（V_{LVD}）：リセット・モード 立ち上がり時 TYP. 2.81V (2.76V～2.87V) 立ち下がり時 TYP. 2.75V (2.70V～2.81V)
統合開発環境（CS+）	ルネサス エレクトロニクス製 CS+ V8.01.00
C コンパイラ（CS+）	ルネサス エレクトロニクス製 CC-RL V1.08.00
統合開発環境（e2studio）	ルネサス エレクトロニクス製 e^2 studio V7.3.0
C コンパイラ（e2studio）	ルネサス エレクトロニクス製 CC-RL V1.08.00

3. 関連アプリケーションノート

本アプリケーションノートに関連するアプリケーションノートを以下に示します。併せて参照してください。

RL78/G13 初期設定（R01AN2757J）アプリケーションノート
RL78/G13 シリアル・インタフェース IICA（マスタ送受信）（R01AN2759J）アプリケーションノート
RL78/G13 IIC マルチマスタ通信（クライアント）（R01AN4708J）アプリケーションノート

4. ハードウェア説明

4.1 ハードウェア構成例

図 4.1に本アプリケーションノートで使用するハードウェア構成例を示します。

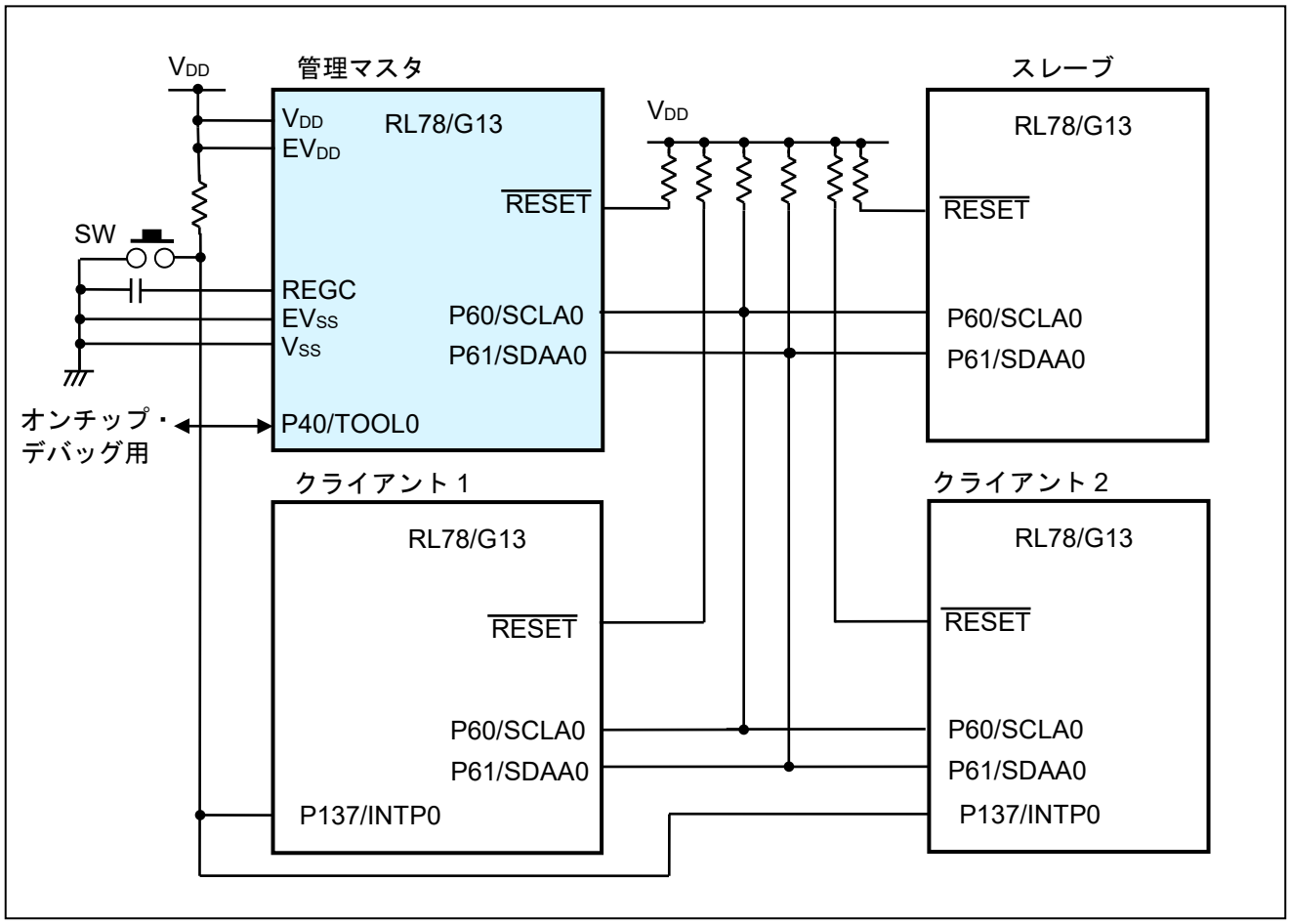


図 4.1 ハードウェア構成

注意 1 この回路イメージは接続の概要を示す為に簡略化しています。実際に回路を作成される場合は、端子処理などを適切に行い、電気的特性を満たすように設計してください（入力専用ポートは個別に抵抗を介して VDD 又は VSS に接続して下さい）。

2 EVSS で始まる名前の端子がある場合には VSS に、EVDD で始まる名前の端子がある場合には VDD にそれぞれ接続してください。

3 VDD は LVD にて設定したリセット解除電圧（V_{LVD}）異常にしてください。

4.2 使用端子一覧

表 4.1に使用端子と機能を示します。

表 4.1 使用端子と機能

端子名	入出力	内容
P60/SCLA0	入出力	IICA0 のシリアル・クロック入出力端子
P61/SDAA0	入出力	IICA0 のシリアル・データ送受信用端子

5. ソフトウェア説明

5.1 動作概要

本アプリケーションノートでは、シリアル・インタフェース IICA を利用して、スレーブへ 16 バイトのデータ送信、スレーブからの 16 バイトのデータ受信を順次行います。その後、クライアントからアクセス権要求に応じたアクセス権の管理を行います。

(1) シリアル・インタフェース IICA の初期設定を行います。

＜設定条件＞

- 動作モードをファースト・モードに設定します。
- 転送クロックを 400kHz に設定します。
- 自局アドレスとして SVA0 レジスタに 0xEE を設定します。
- デジタル・フィルタの動作をオンに設定します。
- 9 クロック目で割り込みが発生するように設定します。
- ストップ・コンディション割り込み禁止に設定します。
- P60/SCLA0 端子をシリアル・クロック（入出力設定)に、P61/SDAA0 端子をデータ送受信（入出力設定）に設定します。

(2) スレーブに送信する 16 バイトのデータを 0x20～0x2F に初期化します。

(3) スレーブにアクセスするため、アクセス権を取得します。

(4) アクセス権取得後、スレーブへ 16 バイトのデータを送信します。

(5) スレーブからの 16 バイトのデータを受信します。

(6) アクセス権を解放します。

(7) クライアントからのアクセスを待ち、クライアントからのアクセス権の取得要求または解放要求に応じた処理を実行します。

注意 本サンプルコードは、RL78/G13 IIC マルチマスタ通信（クライアント）（R01AN4708J）アプリケーションノートに対応しています。

5.2 オプション・バイトの設定一覧

表 5.1にオプション・バイト設定を示します。

表 5.1 オプション・バイト設定

アドレス	設定値	内容
000C0H	11101110B	ウォッチドッグ・タイマ 動作停止 （リセット解除後、カウント停止）
000C1H	01111111B	LVD リセット・モード 2.81V (2.76V～2.87V)
000C2H	11101000B	HS モード、HOCO : 32MHz
000C3H	10000101B	オンチップ・デバッグ許可

5.3 定数一覧

表 5.2にサンプルコードで使用する定数を示します。

表 5.2 サンプルコードで使用する定数

定数名	設定値	内容
BUFSIZE	16	スレーブとの送受信データ数
MD_OK	0x00	ステータス（正常）
MD_ERROR1	0x82	ステータス（IIC バス異常）
MD_ERROR3	0x83	ステータス（アービトレーション負け）
WAITTIME	100	スタート・コンディション生成のタイムアウト時間

5.4 変数一覧

表 5.3にグローバル変数を示します。

表 5.3 グローバル変数

型	変数名	内容	使用関数
volatile uint8_t	semaphore	アクセス権（アクセス権を取得したマスタのスレーブ・アドレス）	R_IICA0_get_sem、R_IICA0_rel_sem、r_iica0_interrupt
volatile uint8_t	sem_work	アクセス権を要求しているクライアントのスレーブ・アドレス	r_iica0_interrupt
volatile uint8_t	sem_state	アクセス権制御の状態変数	r_iica0_interrupt
volatile uint8_t	g_iica0_status	IICA0 の通信状態変数	R_IICA0_wait_comend、R_IICA0_bus_check、r_iica0_interrupt
volatile uint8_t *	gp_iica0_rx_address	受信データ格納ポインタ	R_IICA0_Master_Receive、 r_iica0_interrupt
volatile uint16_t	g_iica0_rx_len	受信データ数	R_IICA0_Master_Receive、r_iica0_interrupt
volatile uint16_t	g_iica0_rx_cnt	受信データ数カウンタ	R_IICA0_Master_Receive、r_iica0_interrupt
volatile uint8_t *	gp_iica0_tx_address	送信データ読み出しポインタ	R_IICA0_Master_Send、r_iica0_interrupt
volatile uint16_t	g_iica0_tx_cnt	送信データ数カウンタ	R_IICA0_Master_Send、r_iica0_interrupt
volatile uint8_t	g_iica0_txbuf	送信データ用バッファ	main、 R_MAIN_UserInit
volatile uint8_t	g_iica0_rxbuf	受信データ用バッファ	main

5.5 関数一覧

表 5.4に関数を示します。

表 5.4 関数

関数名	概要
R_IICA0_get_sem	アクセス権の取得処理
R_IICA0_rel_sem	アクセス権の解放処理
R_IICA0_wait_comend	通信完了待ち処理
R_IICA0_Stop_cond	ストップ・コンディション生成処理
R_IICA0_Master_Send	マスタ送信起動処理
R_IICA0_Master_Receive	マスタ受信起動処理
R_IICA0_bus_check	IIC バス状態確認とスタート・コンディション生成処理
r_iica0_interrupt	IICA0 割り込み処理

5.6 関数仕様

サンプルコードの関数仕様を示します。

[関数名] R_IICA0_get_sem

概要	アクセス権取得処理	
ヘッダ	r_cg_macrodriver.h、r_cg_serial.h、r_cg_userdefine.h	
宣言	MD_STATUS R_IICA0_get_sem(void);	
説明	アクセス権の状態を確認し、アクセス権が解放されていればアクセス権を取得します。	
引数	なし	
リターン値	[MD_OK]の場合	アクセス権の取得成功
	[MD_ERROR1]の場合	アクセス権の取得失敗
備考	なし	

[関数名] R_IICA0_rel_sem

概要	アクセス権解放処理
ヘッダ	r_cg_macrodriver.h、r_cg_serial.h、r_cg_userdefine.h
宣言	void R_IICA0_rel_sem(void);
説明	アクセス権を解放状態（変数 semaphore = 0xFF）に設定します。
引数	なし
リターン値	なし
備考	なし

[関数名] R_IICA0_wait_comend

概要	通信完了待ち処理	
ヘッダ	r_cg_macrodriver.h、r_cg_serial.h、r_cg_userdefine.h	
宣言	MD_STATUS R_IICA0_wait_comend(void);	
説明	IIC バスでの通信完了を待ち、通信結果を戻します。	
引数	なし	
リターン値	[MD_OK]の場合	正常終了
	[MD_ERROR3]の場合	異常終了（ACK 応答なし）
備考	なし	

[関数名] R_IICA0_Stop_cond

概要	ストップ・コンディション生成処理
ヘッダ	r_cg_macrodriver.h、r_cg_serial.h、r_cg_userdefine.h
宣言	void R_IICA0_Stop_cond(void);
説明	IIC バスにストップ・コンディションを生成します。
引数	なし
リターン値	なし
備考	なし

[関数名] R_IICA0_Master_Send

概要	マスタ送信起動処理	
ヘッダ	r_cg_macrodriver.h、r_cg_serial.h、r_cg_userdefine.h	
宣言	MD_STATUS R_IICA0_Master_Send(uint8_t adr、 uint8_t * const tx_buf、 uint16_t tx_num);	
説明	アクセス権取得処理を行います。アクセス権を取得した後、IIC バスが使用可能であれば、スタート・コンディションを生成します。その後、スレーブのアドレス、送信するデータ数、送信データが格納されているアドレスを設定し、スレーブ・アドレスを送信して通信を開始します。	
引数	adr	スレーブのアドレス
	tx_buf	送信データの格納されているアドレス
	tx_num	送信するデータの数
リターン値	[MD_OK]の場合	正常に処理開始
	[MD_ERROR1]の場合	IIC バスが使用中
備考	なし	

[関数名] R_IICA0_Master_Receive

概要	マスタ受信起動処理	
ヘッダ	r_cg_macrodriver.h、r_cg_serial.h、r_cg_userdefine.h	
宣言	MD_STATUS R_IICA0_Master_Receive(uint8_t adr、 uint8_t * const rx_buf、uint16_t rx_num);	
説明	アクセス権取得処理を行います。アクセス権を取得した後、IIC バスが使用可能であれば、スタート・コンディションを生成します。その後、通信相手のスレーブ・アドレス、受信データ数、受信データを格納するアドレスを設定し、スレーブ・アドレスを送信して通信を開始します。	
引数	adr	スレーブ・アドレス
	rx_buf	受信データを格納するアドレス
	rx_num	受信データ数
リターン値	[MD_OK]の場合	正常に処理開始
	[MD_ERROR1]の場合	IIC バスが使用中
備考	なし	

[関数名] R_IICA0_bus_check

概要	IIC バス状態確認とスタート・コンディション生成処理	
ヘッダ	r_cg_macrodriver.h、r_cg_serial.h、r_cg_userdefine.h	
宣言	MD_STATUS R_IICA0_bus_check(void);	
説明	IIC バスの使用状態を確認して他のマスタが IIC バスを使用中でなければ、スタート・コンディションを生成します。	
引数	なし	
リターン値	[MD_OK]の場合	正常にスタート・コンディション生成
	[MD_ERROR1]の場合	スタート・コンディション生成できず
備考	なし	

[関数名] r_iica0_interrupt

概要	IICA0 割り込み処理
ヘッダ	r_cg_macrodriver.h、r_cg_serial.h、r_cg_userdefine.h
宣言	#pragma interrupt r_iica0_interrupt(vect=INTIICA0,bank=RB2)
説明	マスタとして選択されたときは、スレーブとのデータ送受信処理を行います。スレーブとして選択されたときには、アクセス権の取得処理または解放処理を行います。
引数	なし
リターン値	なし
備考	スレーブとしての処理は全て割り込み処理で行います。

5.7 フローチャート

図 5.1に本アプリケーションノートの全体フローを示します。

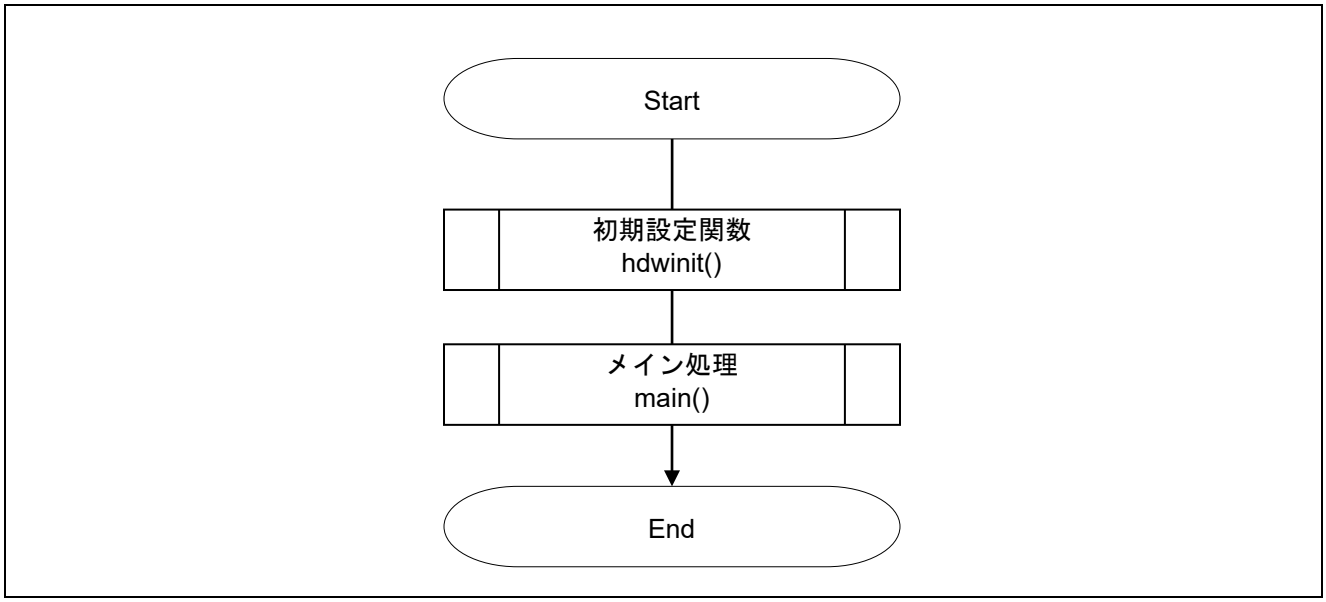


図 5.1 全体フロー

5.7.1 初期設定関数

図 5.2に初期設定関数のフローチャートを示します。

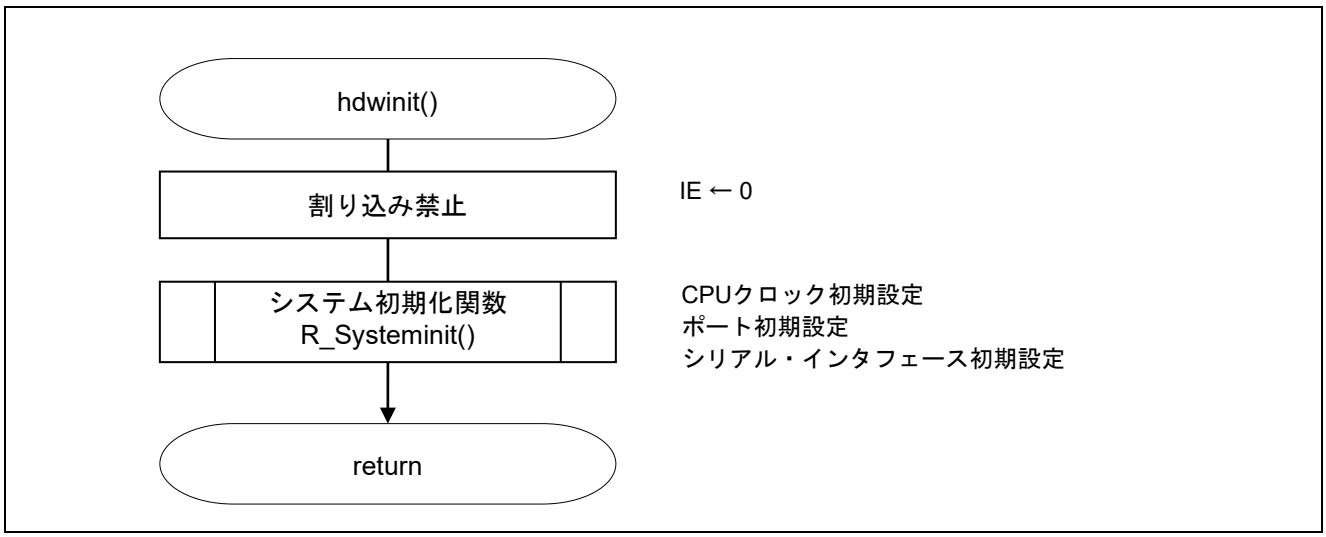


図 5.2 初期設定関数

5.7.2 システム初期化関数

図 5.3にシステム初期化関数のフローチャートを示します。

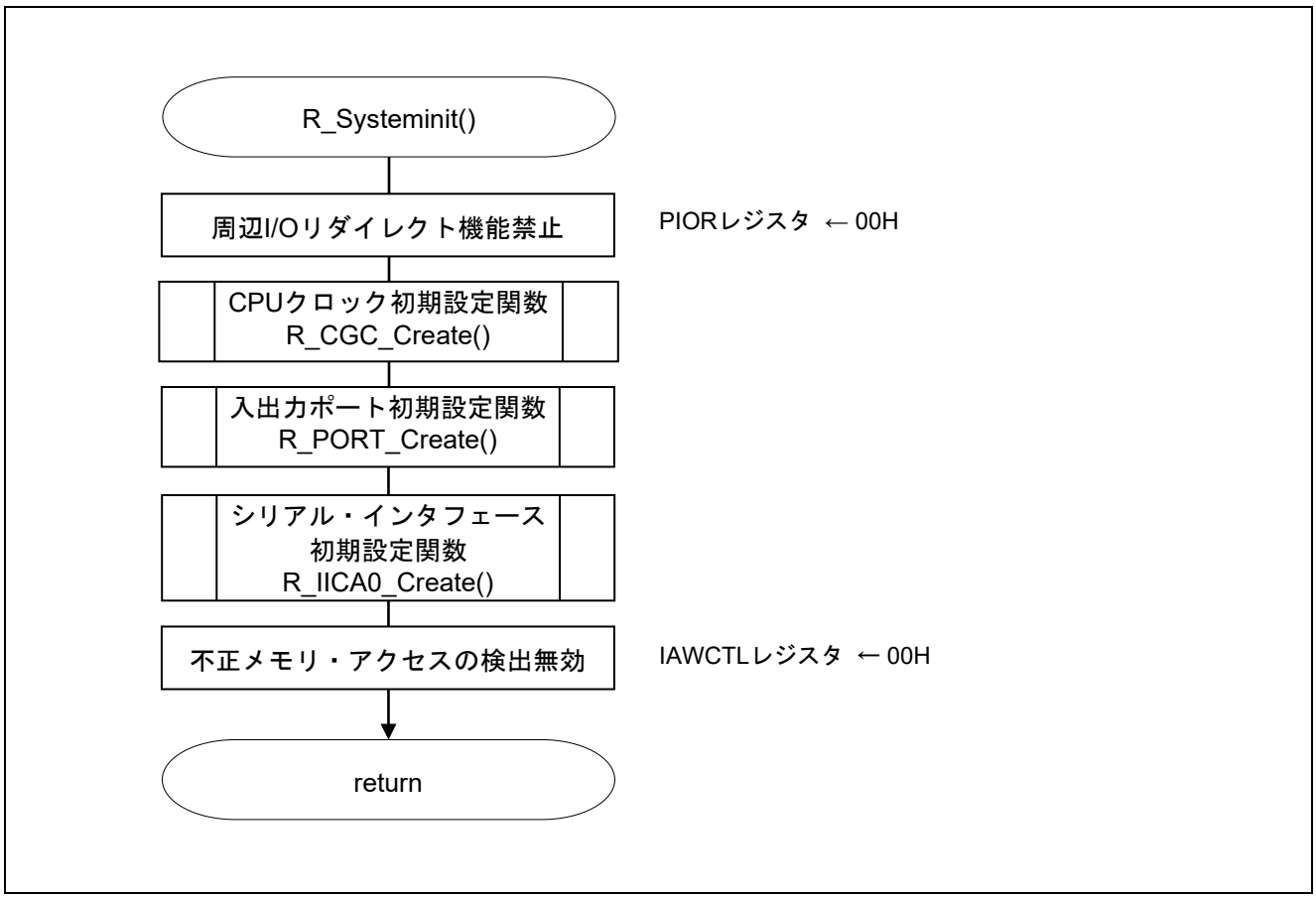


図 5.3 システム初期化関数

5.7.3 CPU クロック初期設定関数

図 5.4に CPU クロック初期設定関数のフローチャートを示します。

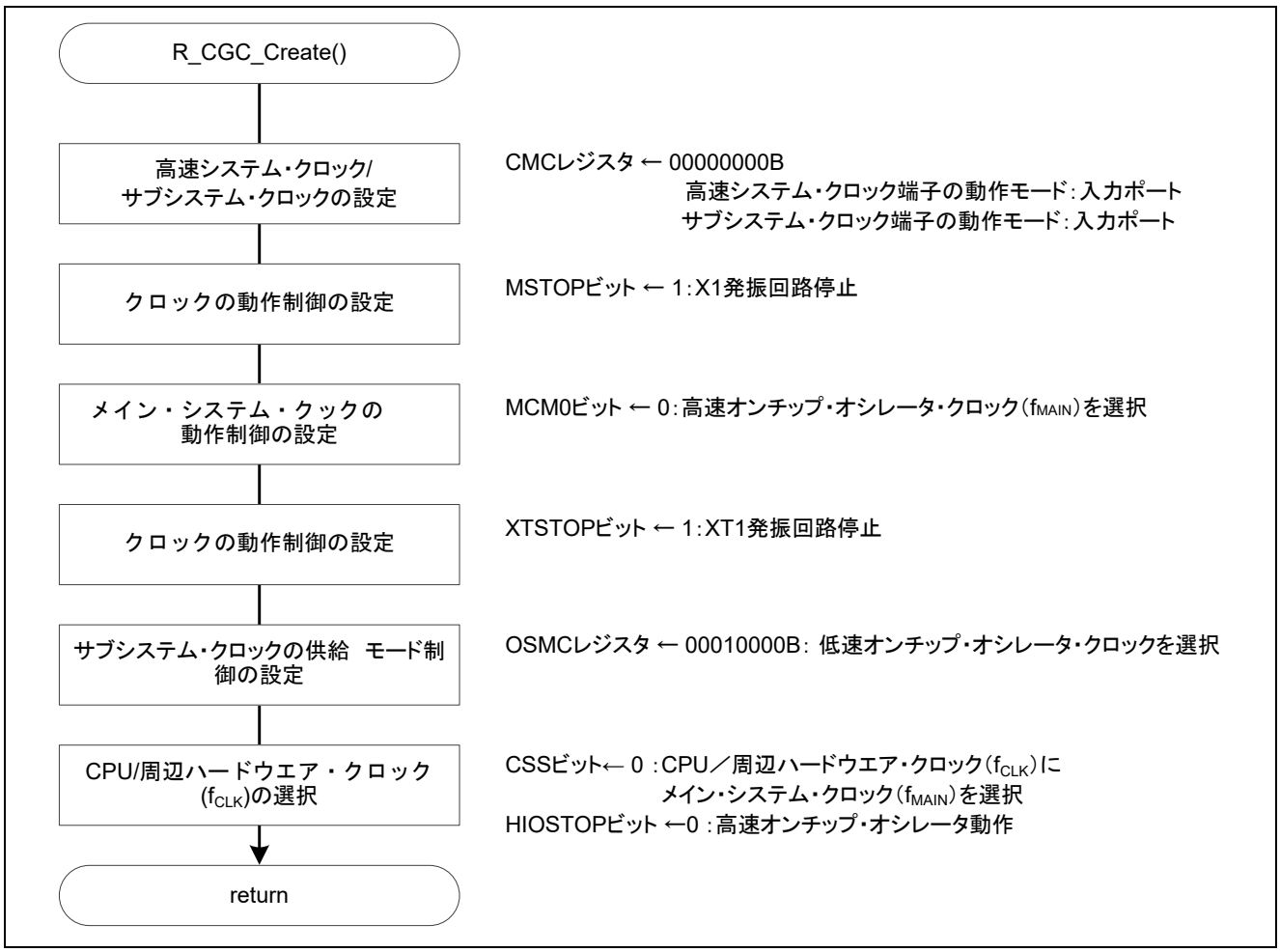


図 5.4 CPU クロック初期設定関数

5.7.4 入出力ポート初期設定関数

図 5.5に入出力ポート初期設定関数のフローチャートを示します。

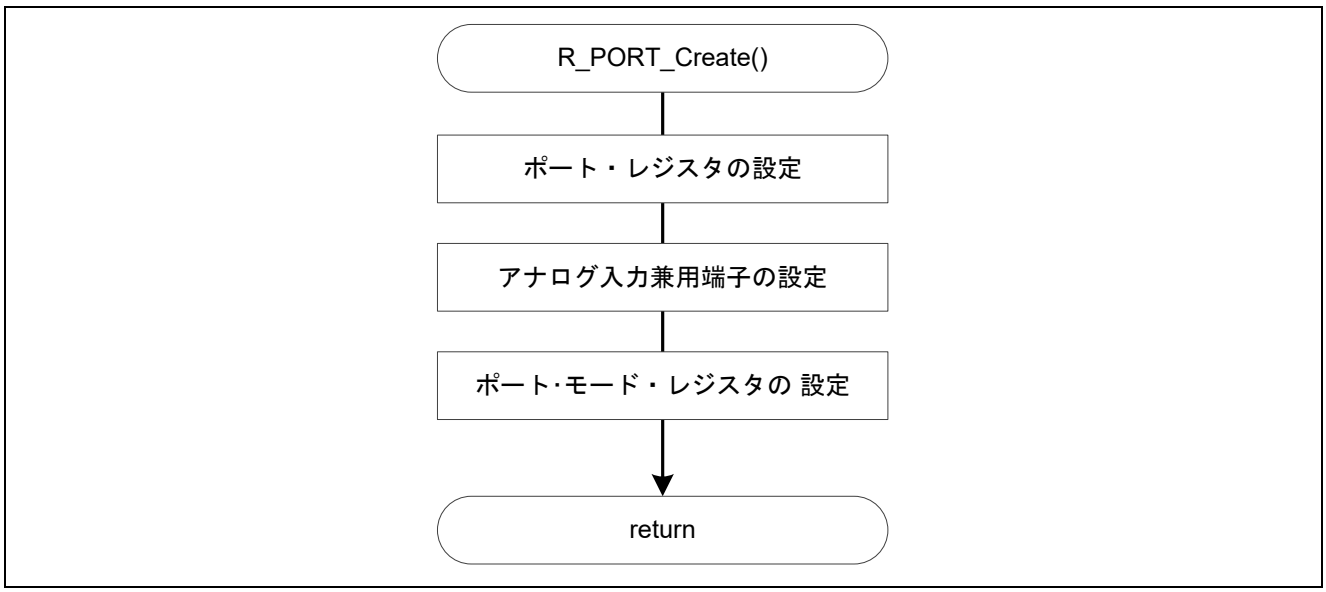


図 5.5 入出力ポート初期設定関数

注 未使用ポートの設定については、RL78/G13 初期設定（R01AN2575J）アプリケーションノート“フローチャート”を参照して下さい。

注意 未使用のポートは、端子処理などを適切に行い、電気的特性を満たすように設計してください。また、未使用の入力専用ポートは個別に抵抗を介して VDD 又は VSS に接続してください。

5.7.5 シリアル・インタフェース初期設定関数

図 5.6にシリアル・インタフェース初期設定関数のフローチャートを示します。

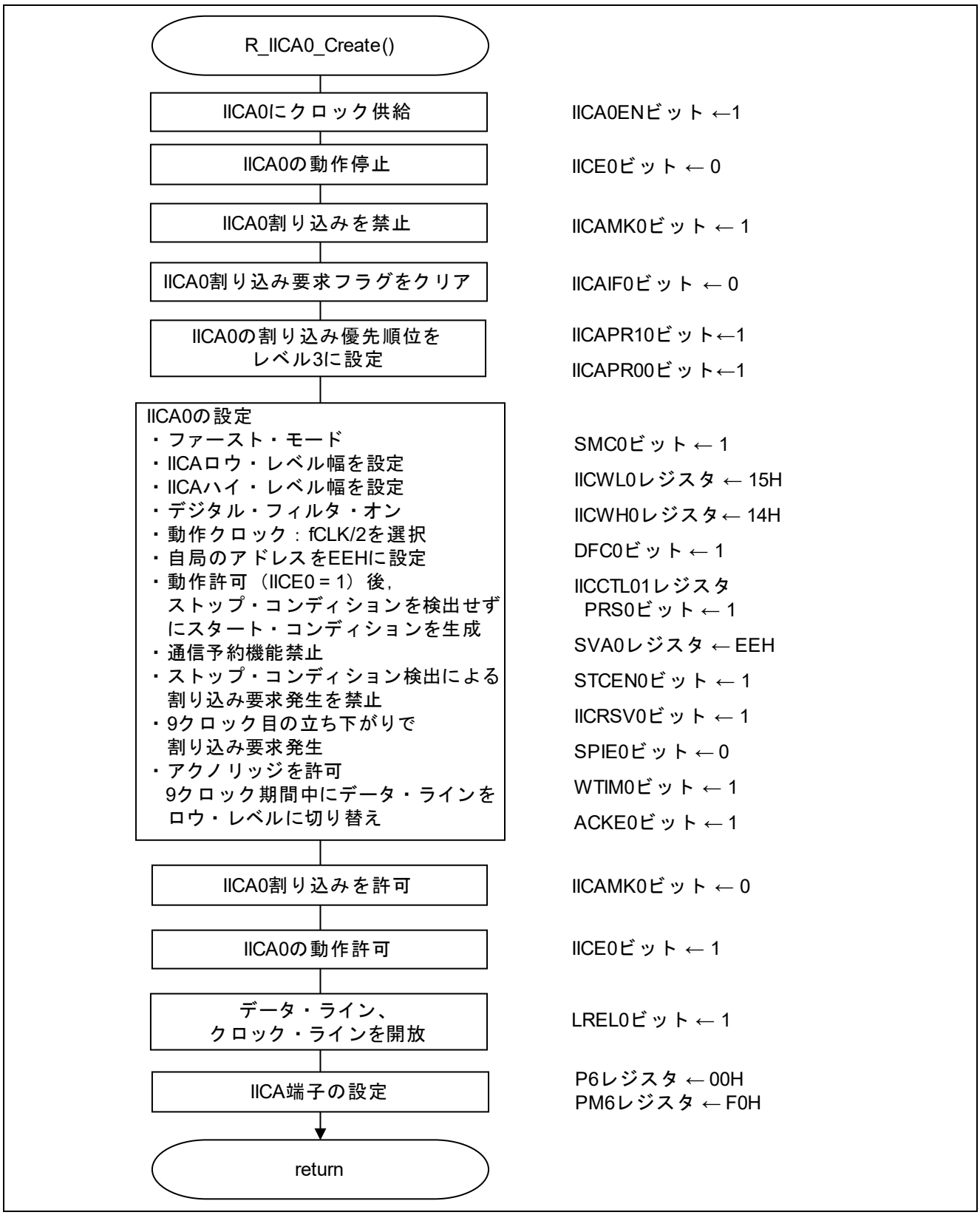


図 5.6 シリアル・インタフェース初期設定関数

5.7.6 メイン処理

図 5.7にメイン関数のフローチャートを示します。メイン関数はマスタとして IIC バスをアクセスするための関数の使い方の例を示します。

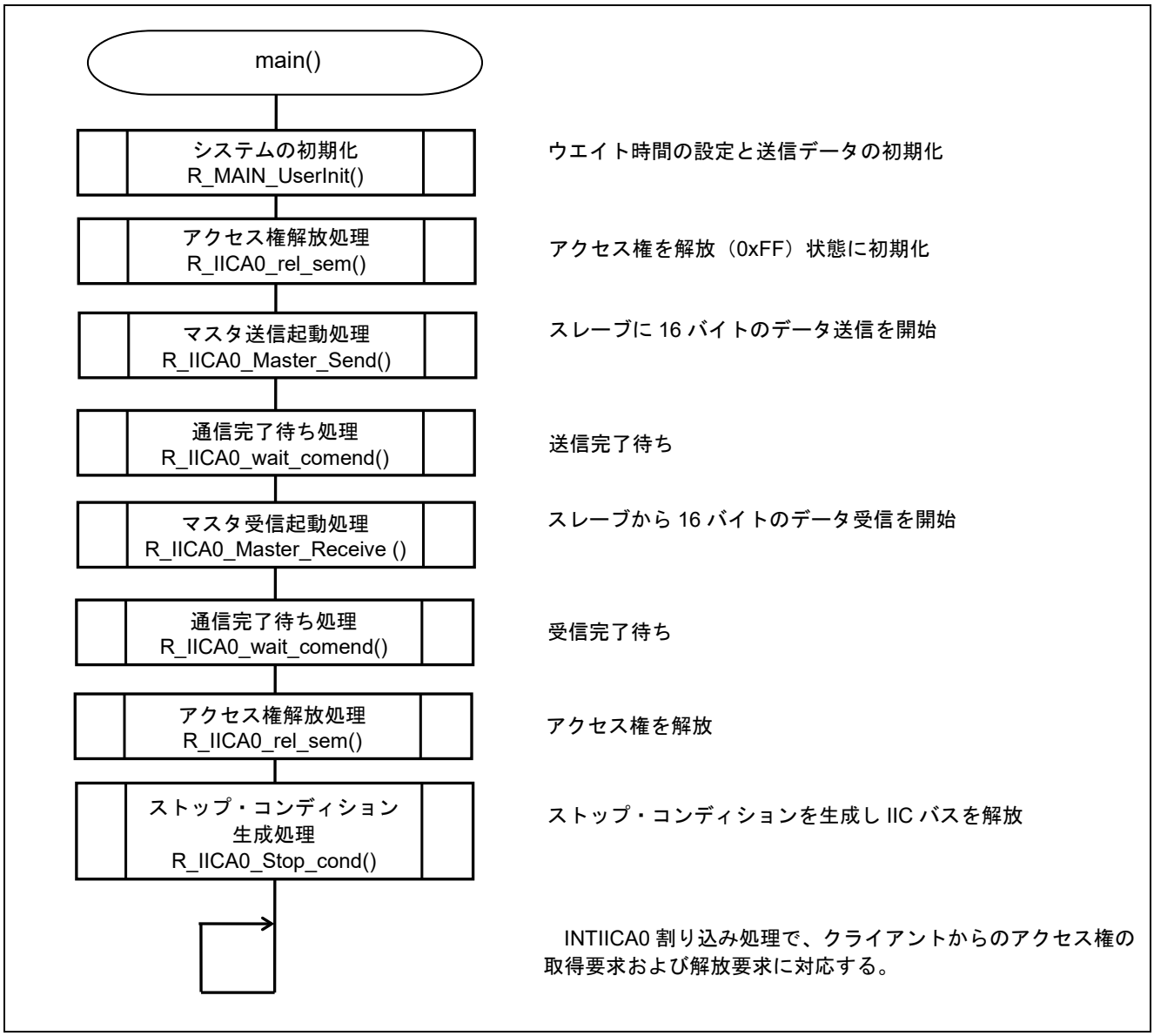


図 5.7 メイン関数

5.7.7 システムの初期化処理

図 5.8にシステムの初期化処理関数のフローチャートを示します。

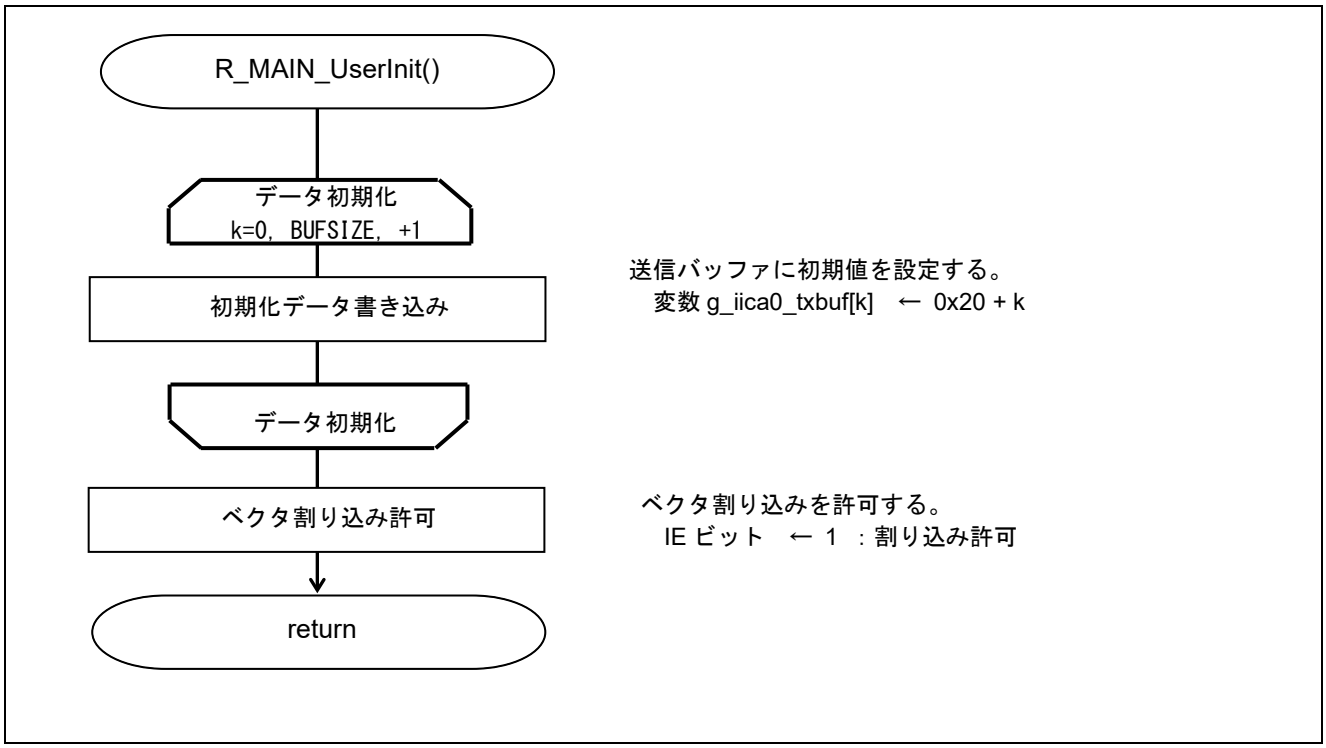


図 5.8 システムの初期化処理関数

5.7.8 アクセス権解放処理

図 5.9にアクセス権解放処理のフローチャートを示します。

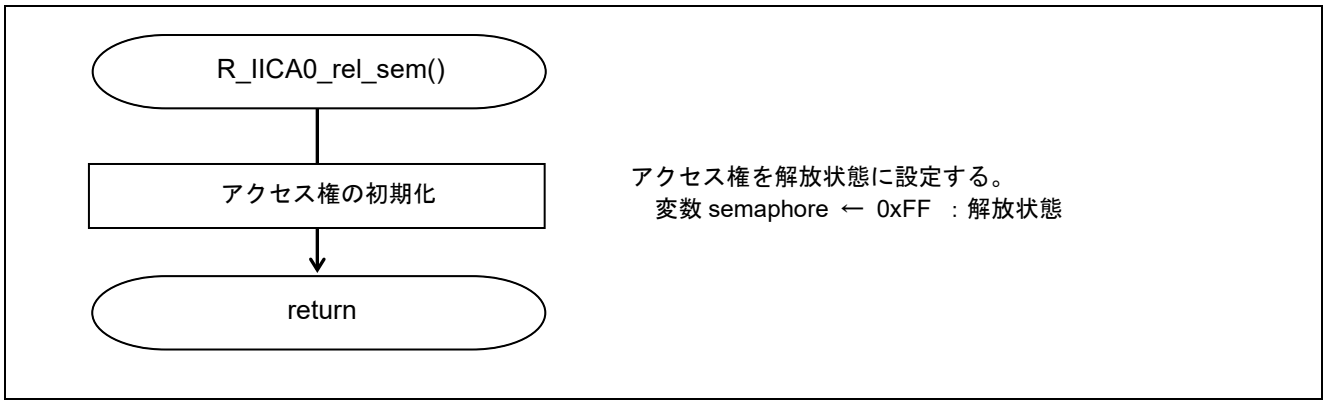


図 5.9 アクセス権解放処理関数

5.7.9 マスタ送信起動処理

図 5.10にマスタ送信起動処理のフローチャートを示します。

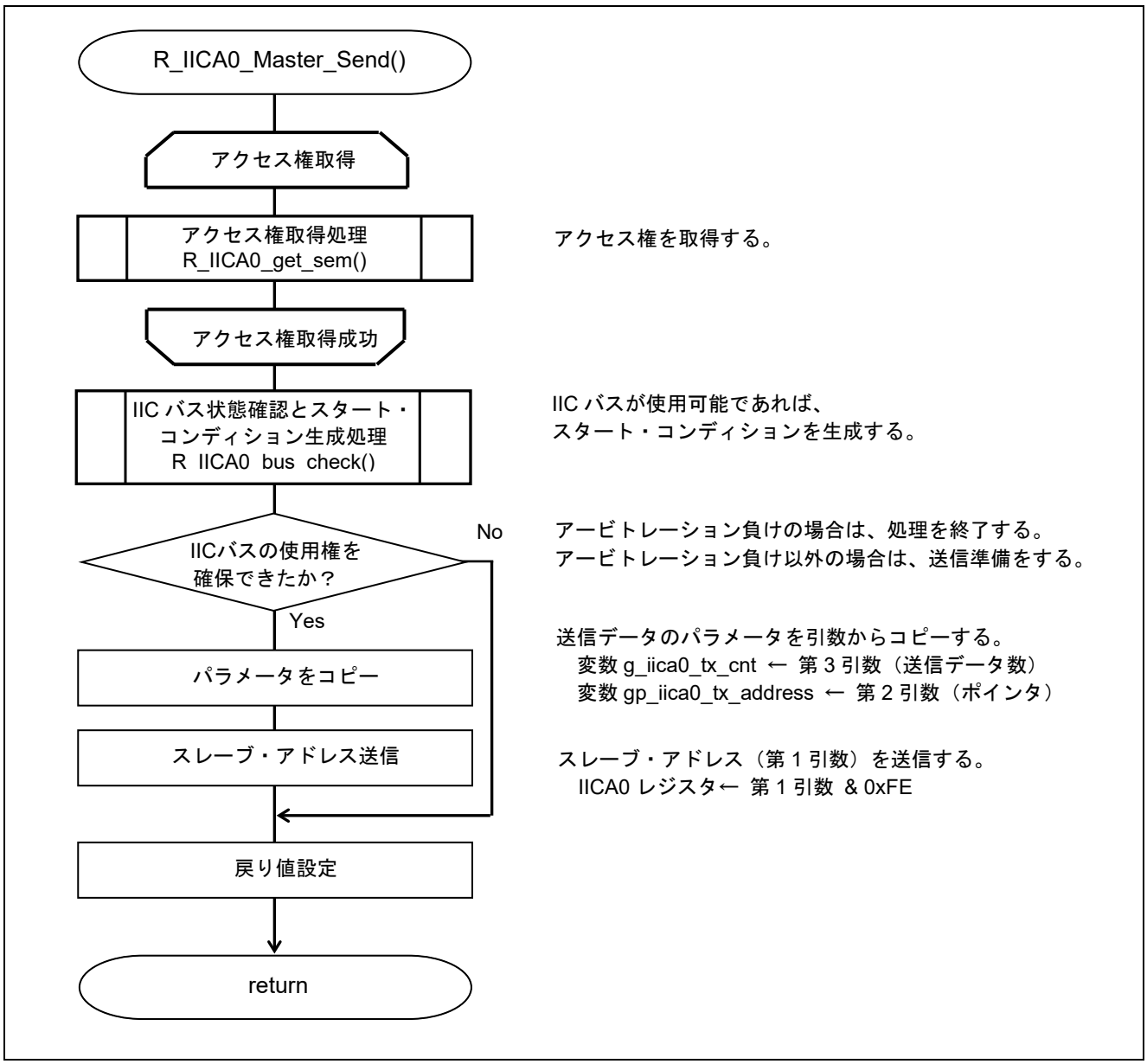


図 5.10 マスタ送信起動処理

5.7.10 アクセス権の取得処理

図 5.11にアクセス権の取得処理のフローチャートを示します。

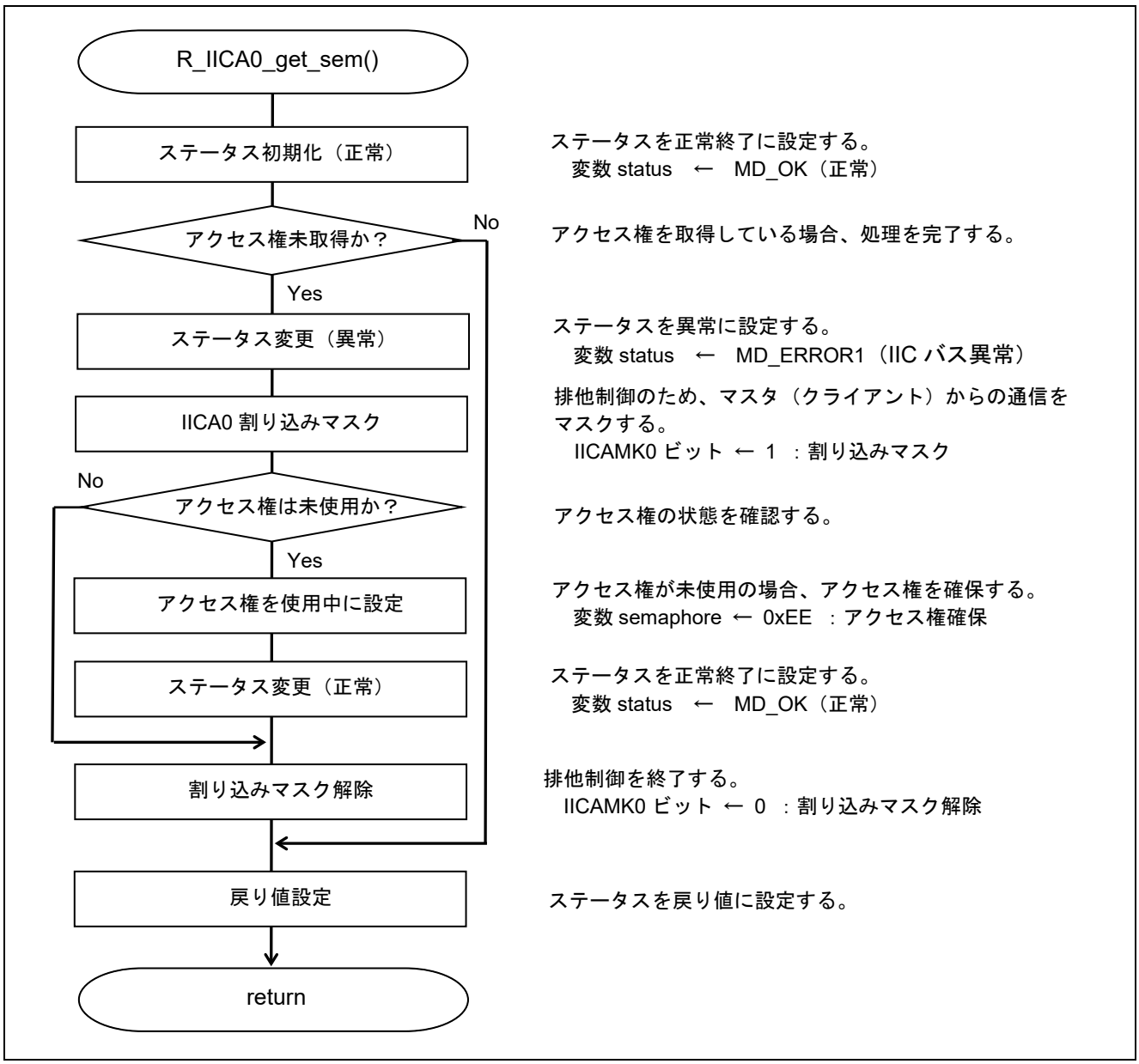


図 5.11 アクセス権取得処理

5.7.11 IIC バス状態確認とスタート・コンディション生成処理

図 5.12に IIC バス状態確認とスタート・コンディション生成処理のフローチャートを示します。

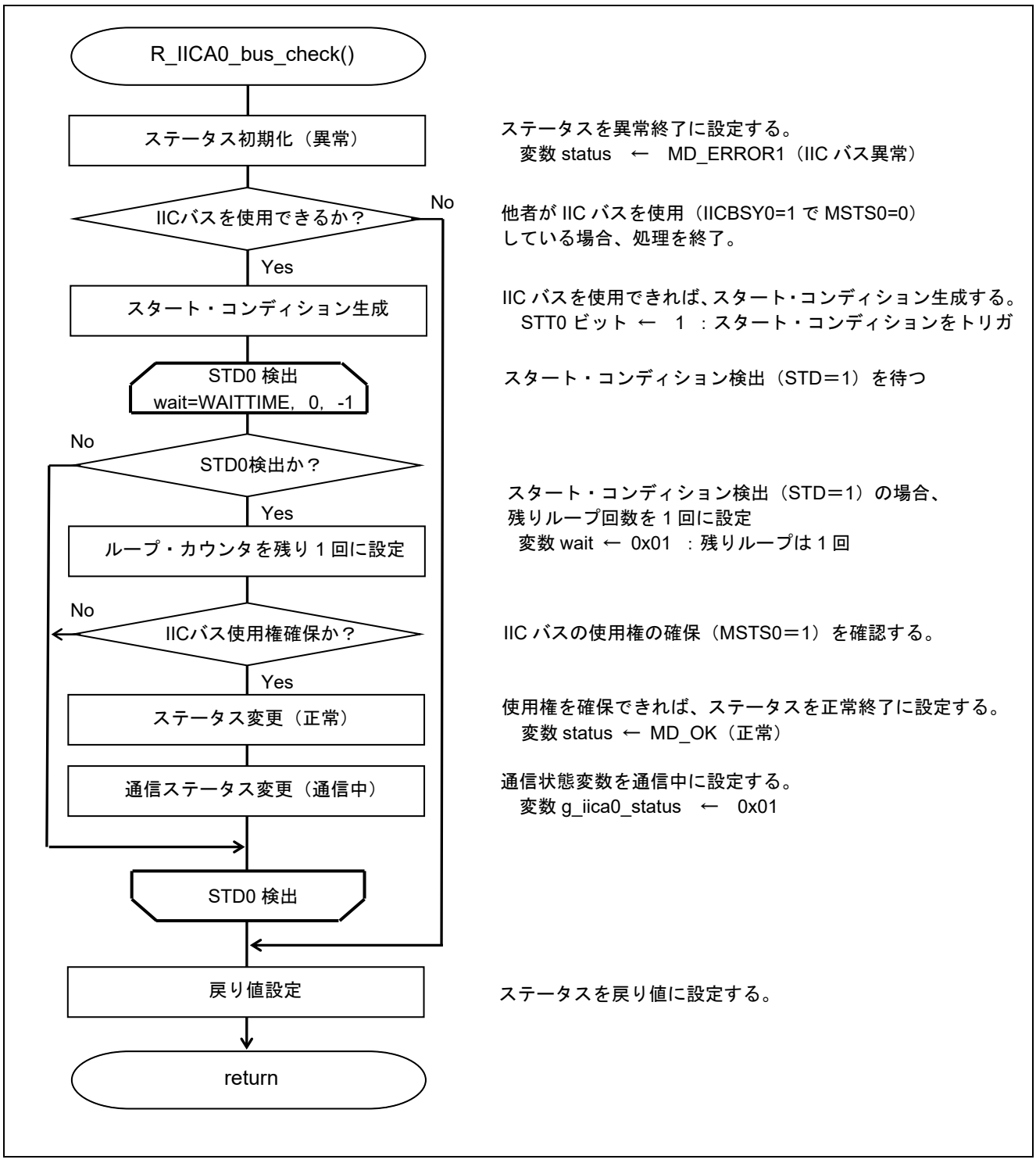


図 5.12 IIC バス状態確認とスタート・コンディション生成処理

5.7.12 マスタ受信起動処理

図 5.13にマスタ受信起動処理のフローチャートを示します。

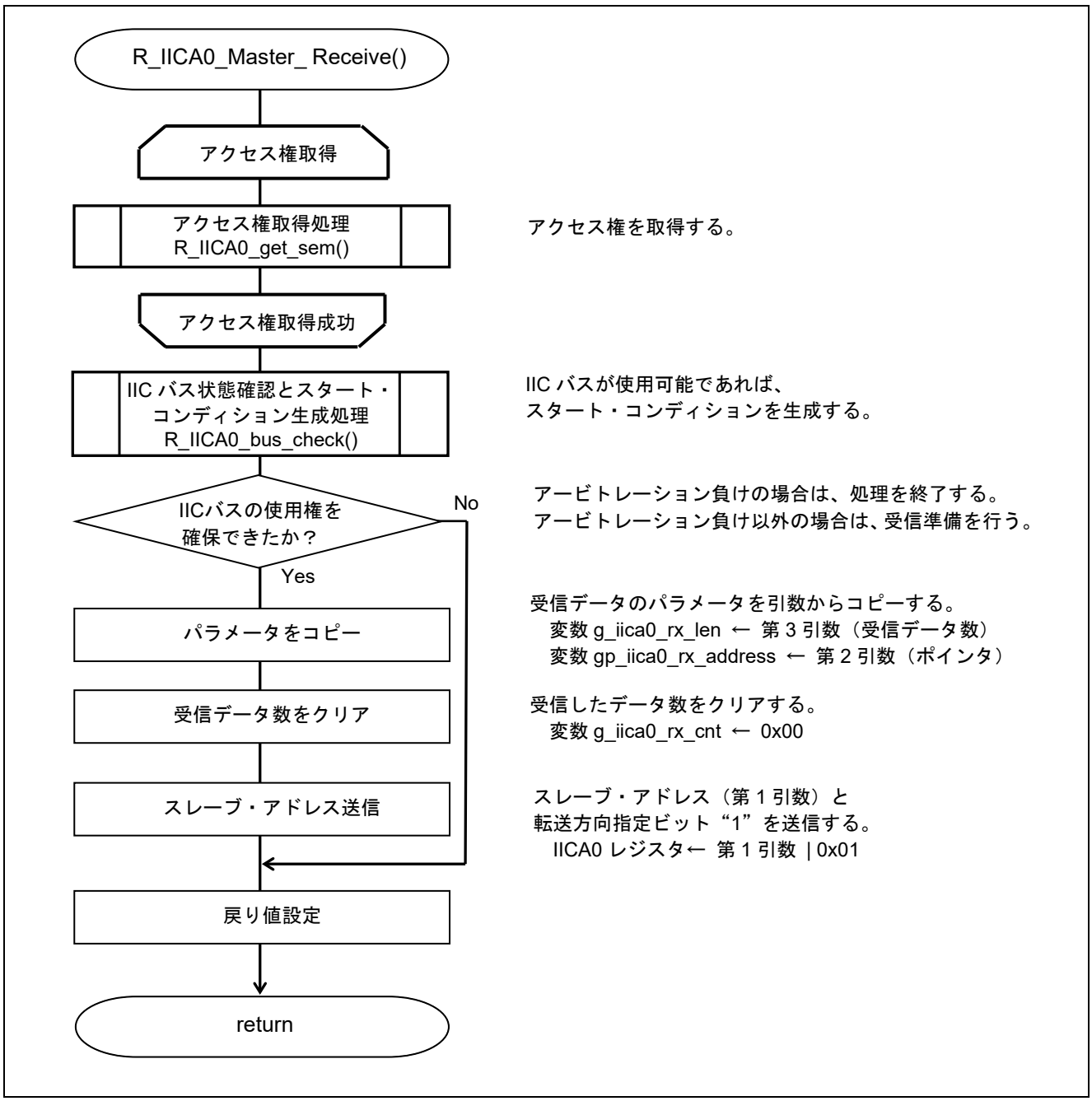


図 5.13 マスタ受信起動処理

5.7.13 通信完了待ち処理

図 5.14に通信完了待ち処理のフローチャートを示します。

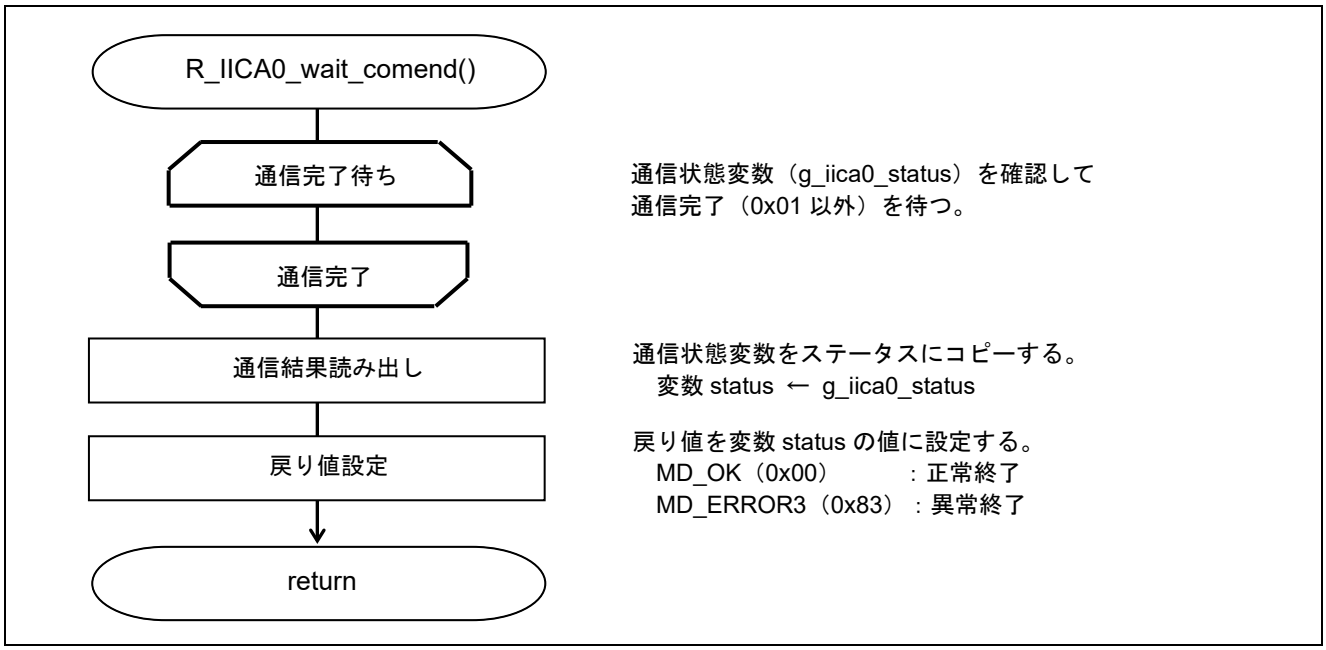


図 5.14 通信完了待ち処理関数

5.7.14 ストップ・コンディション生成処理

図 5.15に IIC バス解放処理のフローチャートを示します。

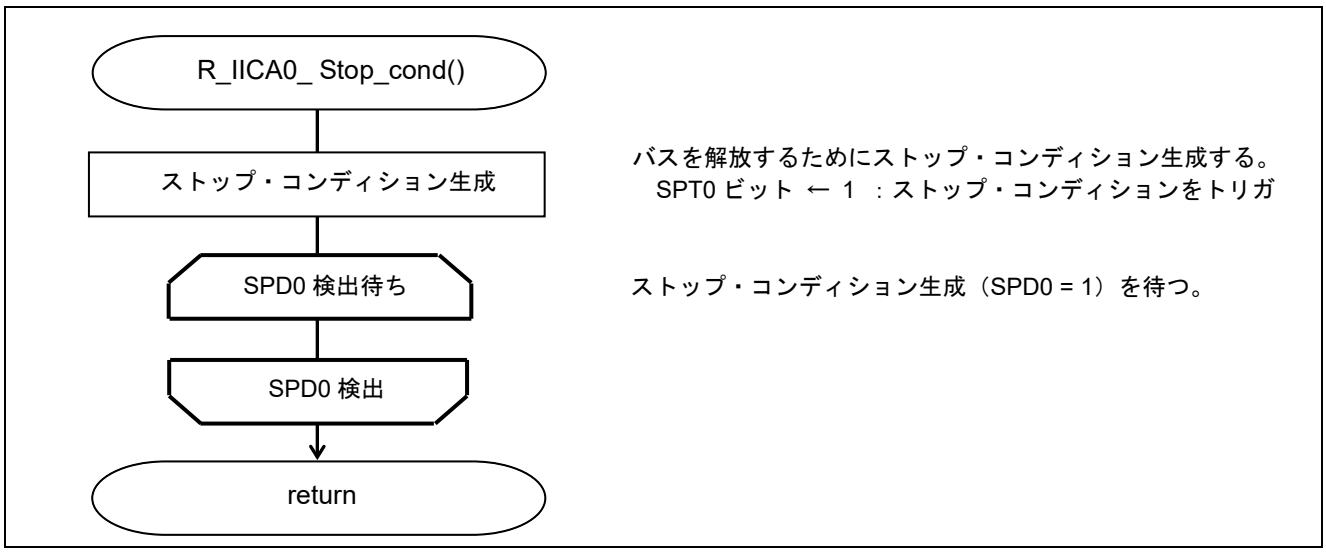
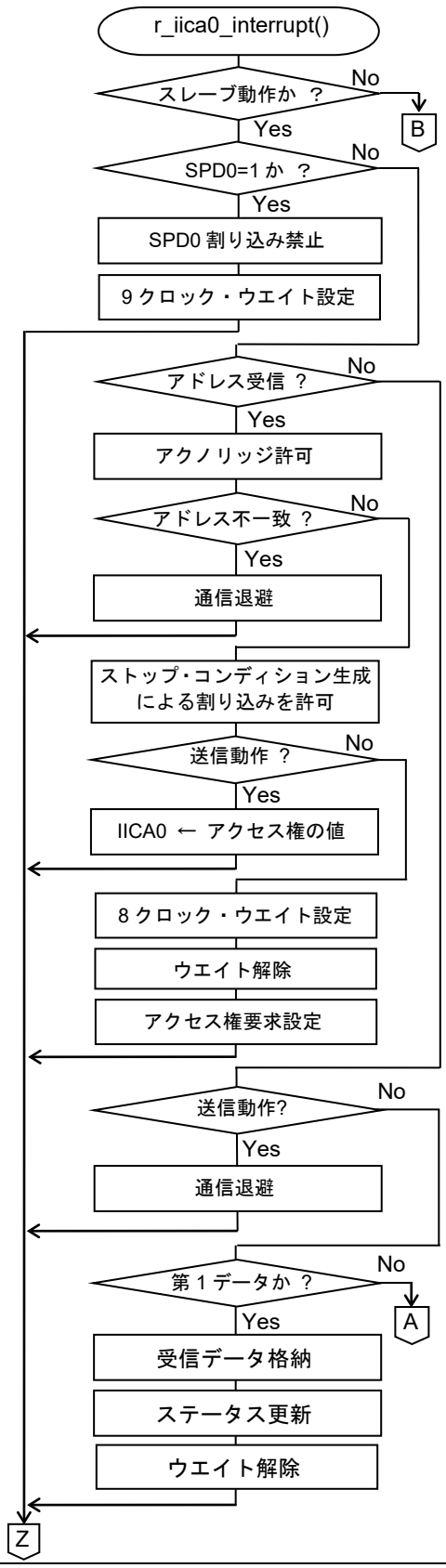


図 5.15 ストップ・コンディション生成処理

5.7.15 IICA0 割り込み処理

図 5.16～図 5.19に IICA0 割り込み処理関数のフローチャートを示します。



スレーブ（MSTS0 = 0）として選択されたかの確認

ストップ・コンディション（SP）検出による割り込みの場合、SP 検出を禁止し、9 クロック目の割り込みに設定して処理を終了する。
SPIE0 ビット ← 0 ：SPD0 割り込み禁止
WTIM0 ビット ← 1 ：9 クロック目でウエイト

STD0 ビット=0 のときに No に分岐

アドレス不一致（拡張コード）の場合、通信を終了する。

LREL0 ビット ← 1 ：通信退避

SPIE0 ビット=1

スレーブとしてのアドレス受信タイミング

アクセス権の値（semaphore）の読み出しの場合、
IICA0 に semaphore を設定する。（ACK 確認しない）

アクセス権の取得要求の場合、ACK 制御のために、8 クロック・ウエイトに変更する。
WTIM0 ビット ← 0 ：8 クロック・ウエイト
ウエイトを解除してデータを受信する。
WREL0 ビット ← 1 ：ウエイト解除
アクセス権・データの受信ステータスを設定する。
変数 sem_state ← 0x02 ：アクセス権・セット動作

スレーブとしてのデータ送受信タイミング

送信動作（TRC0 ビット=0）の場合、通信退避する。
（送信したデータに対する ACK 応答を確認しない）
LREL0 ビット ← 1 ：通信退避

変数 sem_state =2 のときに Yes へ分岐

要求元アドレス以外を受信する場合、反転データの受信処理に移動する。
要求元アドレスを受信する場合、受信データを格納する。
変数 sem_work ← IICA0 レジスタ ：データ格納
変数 sem_state ← 0x01 ： 反転データ待ち
WREL0 ビット ← 1 ：ウエイト解除

図 5.16 IICA0 割り込み処理関数（1/4）

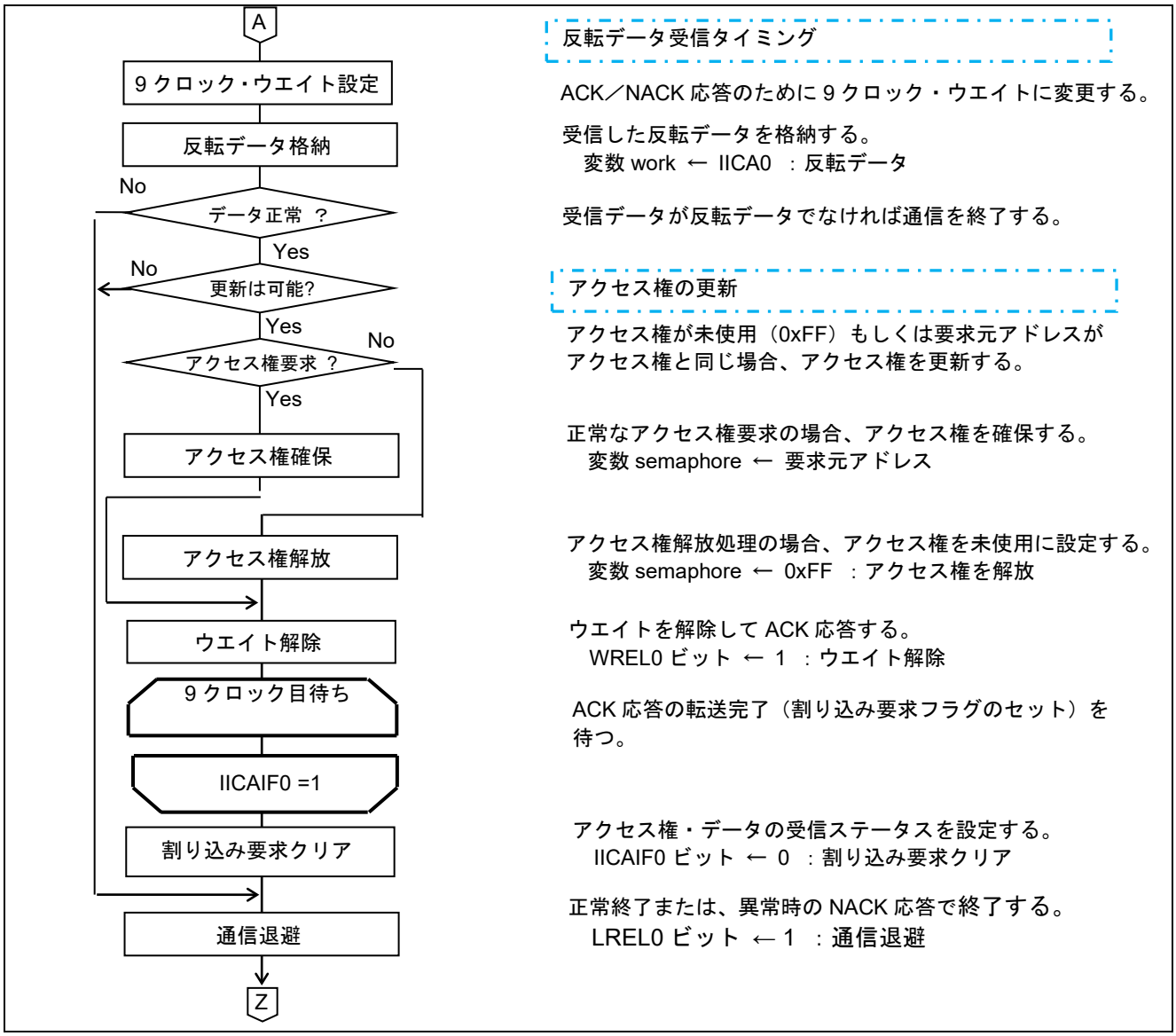
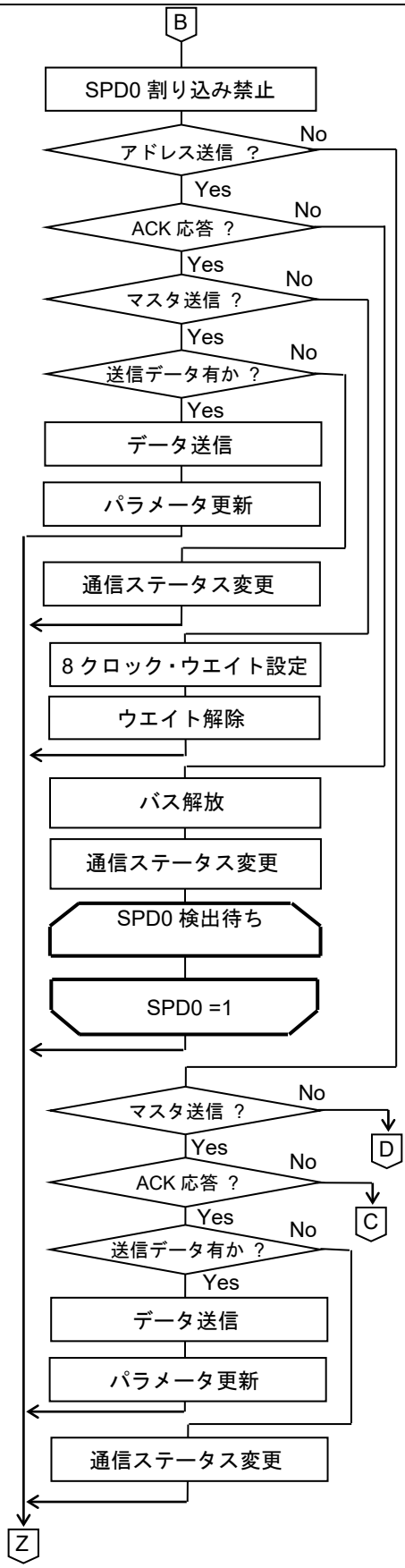


図 5.17 IICA0 割り込み処理関数（2/4）



マスタとしての動作

マスタ動作では、ストップ・コンディション検出による割り込みを禁止する。（SPIE0 = 0）

アドレス送信（STD0 = 1）かデータ転送（STD0 = 0）を確認する。

ACK 応答を確認する。

マスタ送信の場合、残りデータ数を確認する。

マスタ送信動作開始

残り送信データがあれば、次のデータを送信し、送信用パラメータを更新する。
- IICA0 レジスタ ← 変数 gp_iica0_tx_address
- 変数 gp_iica0_tx_address をインクリメント
- 変数 g_iica0_tx_cnt をデクリメント

送信完了の場合、通信ステータスを正常に設定する。
- 変数 g_iica0_status ← MD_OK（正常）

マスタ受信動作開始

マスタ受信の場合、8 クロック・ウエイトに設定し、IIC バスのウエイトを解除して受信動作を行う。

NACK 応答検出

NACK 応答の場合、ストップ・コンディションを生成して IIC バスを解放する。（SPT0 ビット ← 1）
通信ステータスをエラーに設定する。
- 変数 g_iica0_status ← MD_ERROR3

ストップ・コンディションが生成されるまで待つ。

データ転送完了

送信（TRC0 = 1）の場合、
スレーブからの ACK 応答を確認する。

スレーブからの ACK 応答があれば、送信を継続する。
次の送信データがあれば、次のデータを送信して送信用パラメータを更新する。
- IICA0 レジスタ ← 変数 gp_iica0_tx_address
- 変数 gp_iica0_tx_address をインクリメント
- 変数 g_iica0_tx_cnt をデクリメント

次の送信データがなければ、
通信ステータスを正常に設定する。
- 変数 g_iica0_status ← MD_OK（正常）

図 5.18 IICA0 割り込み処理関数（3/4）

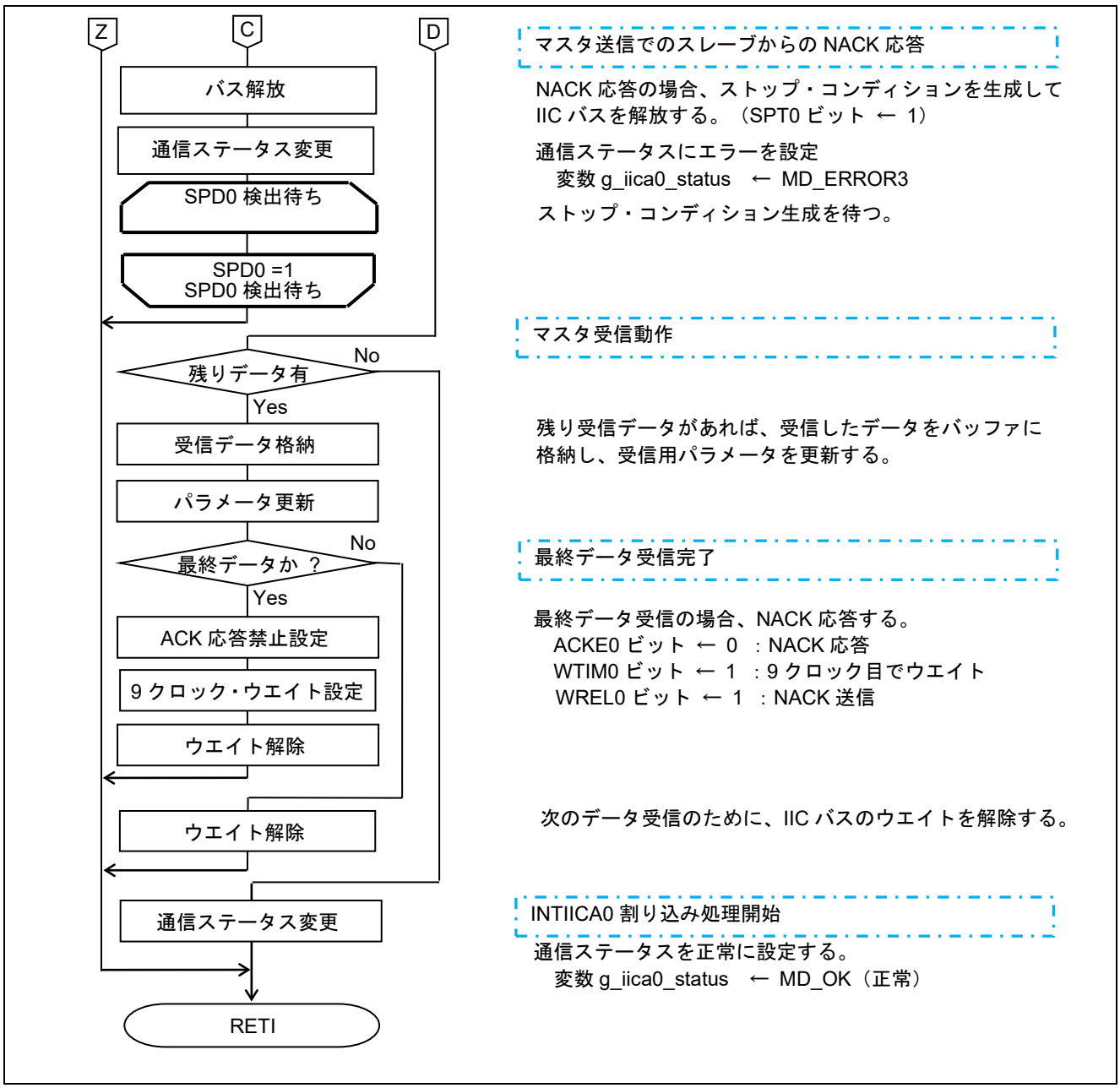


図 5.19 IICA0 割り込み処理関数（4/4）

6. 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。

7. 参考ドキュメント

RL78/G13 ユーザーズマニュアル ハードウェア編（R01UH0146J）

RL78 ファミリ ユーザーズマニュアル ソフトウェア編（R01US0015J）

（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

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2019.12.20	—	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、本ドキュメントおよびテクニカルアップデートを参照してください。

1. 静電気対策
   CMOS 製品の取り扱いの際は静電気防止を心がけてください。CMOS 製品は強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、当社が出荷梱包に使用している導電性のトレーやマガジンケース、導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。また、CMOS 製品を実装したボードについても同様の扱いをしてください。
2. 電源投入時の処置
   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. 電源オフ時における入力信号
   当該製品の電源がオフ状態のときに、入力信号や入出力プルアップ電源を入れないでください。入力信号や入出力プルアップ電源からの電流注入により、誤動作を引き起こしたり、異常電流が流れ内部素子を劣化させたりする場合があります。資料中に「電源オフ時における入力信号」についての記載のある製品は、その内容を守ってください。
4. 未使用端子の処理
   未使用端子は、「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。
5. クロックについて
   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。
6. 入力端子の印加波形
   入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。CMOS 製品の入力がノイズなどに起因して、V_{IL}（Max.）から V_{IH}（Min.）までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定の場合はもちろん、V_{IL}（Max.）から V_{IH}（Min.）までの領域を通過する遷移期間中にチャタリングノイズなどが入らないように使用してください。
7. リザーブアドレス（予約領域）のアクセス禁止
   リザーブアドレス（予約領域）のアクセスを禁止します。アドレス領域には、将来の拡張機能用に割り付けられている リザーブアドレス（予約領域）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。
8. 製品間の相違について
   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。同じグループのマイコンでも型名が違うと、フラッシュメモリ、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ幅射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品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・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て生じた損害（お客様または第三者いずれに生じた損害も含みます。以下同じです。）に関し、当社は、一切その責任を負いません。
2. 当社製品、本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因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またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
3. 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。
4. 当社製品を、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改変、複製、リバースエンジニアリング等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。
5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。
   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、産業用ロボット等
   高品質水準：輸送機器（自動車、電車、船舶等）、交通制御（信号）、大規模通信機器、金融端末基幹システム、各種安全制御装置等
   当社製品は、データシート等により高信頼性、Harsh environment 向け製品と定義しているものを除き、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負いません。
6. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合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、当社製品の品質および信頼性の向上に努めていますが、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、データシート等において高信頼性、Harsh environment 向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他社会的損害等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、お客様の機器・システムとしての出荷保証を行ってください。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、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
8. 当社製品の環境適合性等の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。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。
9. 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・システムに使用することはできません。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
10. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものといたします。
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禁じます。
12. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。

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直接的、間接的に支配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。

(Rev.4.0-1　2017.11)

本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレシア）

www.renesas.com

お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

www.renesas.com/contact/

商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。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。