

要旨

本アプリケーションノートは、M16C/63,64A,64C,65,65C,6C,5LD,56D,5L,56,5M,57 グループの製品の過去の開発時に発生したトラブル事例と解決方法を掲載しています。開発時にトラブルが発生した場合、該当するトラブル事例を本アプリケーションノートから検索して問題の解決を図ることを目的としています。

対象デバイス

M16C/63,64A,64C,65,65C,6C,5LD,56D,5L,56,5M,57

この資料で使用している「i」(UARTi、UiMR レジスタなど)は、特殊モード1を使用できるシリアルインタフェースのチャンネルを示しています。特殊モード1を使用できる UARTi はマイコンにより異なりますので、ハードウェアマニュアルで確認してください。

UARTi 特殊モード1による I²C バスインタフェースは、UARTi のクロック同期回路に付加された I²C バス通信のための補助機能を、ソフトウェアで制御することにより、簡易的に I²C バス通信を実現します。通信制御をソフトウェアで行いますので、ハードウェアで実現する I²C バスインタフェースモジュールに比べてソフトウェアの処理時間やタイミングに制約があります。お客様のソフトウェアにおける、I²C バス通信以外のプログラムとの相互影響を含め、お客様システムでの十分な検証と評価を行ってください。

下表に本アプリケーションノートに掲載しているトラブル事例一覧を示します。

本アプリケーションノートに掲載しているトラブル事例一覧

トラブル事例	掲載している章
I ² C モード マスタ/スレーブ動作時の共通のトラブル	1.章
I ² C モード マスタ動作時のトラブル	2.章
I ² C モード スレーブ動作時のトラブル	3.章

本アプリケーションノートを参照するにあたって

- 各章の先頭にトラブル事例と確認内容の一覧を記載しています。また、トラブル事例と確認内容にリンクを貼っています。詳細はリンク先に記載していますので、該当するトラブル事例または確認内容を選択してリンク先へ移動してください。
- リンク先から前の画面に戻りたいときは、ALT+←キーで戻ることができます。
- 参考アプリケーションノートはM16C/65Cグループの製品を代表に紹介しています。
ルネサス エレクトロニクスホームページに、ご使用の製品のアプリケーションノートがあるかご確認ください。

本アプリケーションノートでは、一部の回路名、モード名、信号名に略称を使用しています。
下表に本アプリケーションノートで使用している略称一覧を示します。

本アプリケーションノートで使用している略称一覧

名称	略称
ハイインピーダンス	Hi-z
High-performance Embedded Workshop	HEW

目次

1.	I2Cモード マスタ／スレーブ動作時の共通のトラブル	4
1.1	意図しない割り込みが発生する	5
1.2	Fast-modeで通信できない	7
1.3	各コンディションが検出できない	9
1.4	ビットずれが発生する	12
1.5	正しいデータが受信できない	13
1.6	オーバランエラーが発生する	15
1.7	SDAi端子が“L”に固定されたまま通信ができなくなる	16
1.8	期待どおりの転送速度にならない	17
2.	I2Cモード マスタ動作時のトラブル	18
2.1	各コンディションが正しく生成できない	19
2.2	SDAi端子が“L”になる／“L”パルスが出力される	25
2.3	SCLi端子が“L”になる／“L”パルスが出力される	28
2.4	期待どおりデータが出力されない	32
2.5	Fast-Modeで通信できない	36
2.6	常にACKが返ってくる	38
2.7	意図しないアービトレーションロストが発生する	39
3.	I2Cモード スレーブ動作時のトラブル	41
3.1	期待どおりのデータが出力されない	42
3.2	ビットずれが発生する	45
3.3	各コンディションが検出されない	46
4.	解析手法	47
4.1	レジスタが期待どおり設定できているか確認する	47
4.2	動作中の周辺機能クロックf1の周波数を確認する	51
4.3	割り込み発生箇所を確認する	53
4.4	通信中に端子が“L”ホールドした時、どのデバイスが“L”にしているか確認する	54
4.5	簡易I2Cモード通信時の信号に問題がないか確認する	56
4.6	Nチャネルオープンドレイン出力端子がプルアップできているか確認する	66
4.7	割り込みが抜けたり発生しなくなる原因を確認する	67
5.	該当する現象がない／該当する現象がわからない	68
5.1	解決しない場合は	68
6.	参考ドキュメント	69

1. I²Cモード マスタ/スレーブ動作時の共通のトラブル

表 1.1 にトラブル事例と確認内容一覧を示します。確認内容の詳細と解決方法は、表の「参照」欄に示した項を参照してください。

表 1.1 トラブル事例と確認内容一覧

節	トラブル事例	確認内容	参照
1.1	意図しない割り込みが発生する	モードなどの変更後にIRビットを“0”にしましたか	1.1.1
		割り込み要因を切り替えていませんか	1.1.2
1.2	Fast-modeで通信できない(注1)	UiBRGカウンタソースは、10MHz以上になっていますか	1.2.1
		SDAiデジタル遅延値の設定は適切ですか	1.2.2
1.3	各コンディションが検出できない	入力レベルは推奨動作条件を満たしていますか	1.3.1
		UiBRGカウンタソースは、10MHz以上になっていますか	1.3.2
		SDAiデジタル遅延値の設定は適切ですか	1.3.3
		UiBRGレジスタの値が“03h”未満になっていませんか	1.3.4
1.4	ビットずれが発生する	UiBRGレジスタの値が“03h”未満になっていませんか	1.4.1
1.5	正しいデータが受信できない	受信割り込みでUiRBレジスタを読み出していませんか	1.5.1
		送信割り込み要因は、送信完了にしていますか	1.5.2
1.6	オーバランエラーが発生する	次のデータを受信する前にUiRBレジスタを読み出していますか	1.6.1
1.7	SDAi端子が“L”に固定されたまま通信ができなくなる	ビットずれが発生していませんか	1.7.1
1.8	期待どおりの転送速度にならない	クロック同期化機能によるサンプリング遅延を考慮しましたか	1.8.1
-	該当する現象がない/該当する現象がわからない	-	5.

注1. Fast-mode : I²C-bus規格で最高400kbpsの速度で通信することを可能としたモード

1.1 意図しない割り込みが発生する

現象の例 : UiSMR3レジスタのCKPHビットを“1”にして、クロック位相設定をクロック遅れありに変更すると受信割り込みが発生した。
UiC1レジスタのUiIRSビットを“1”にして、送信割り込み要因を“送信完了”にすると送信割り込みが発生した。など

1.1.1 モードなどの変更後にIRビットを“0”にしましたか

次のいずれかのビットを変更した後、UARTの各割り込み制御レジスタのIRビットを“0”(割り込み要求なし)にしましたか。

- UiMRレジスタのSMD2～SMD0ビット
- UiSMRレジスタのIICMビット
- UiSMR2レジスタのIICM2ビット
- UiSMR3レジスタのCKPHビット

モードやクロックの位相などを変更すると、UARTの各割り込み制御レジスタのIRビットが“1”(割り込み要求あり)になることがあります。

◆解決方法

次のビットを変更した後、UARTの各割り込み制御レジスタのIRビットを“0”(割り込み要求なし)にしてください。

- UiMRレジスタのSMD2～SMD0ビット
- UiSMRレジスタのIICMビット
- UiSMR2レジスタのIICM2ビット
- UiSMR3レジスタのCKPHビット

◆アプリケーションノート

UARTi特殊モード1を使用したI²C-busインタフェース(マスタ送信/受信)(RJJ05B1596)

UARTi特殊モード1を使用したI²C-busインタフェース(スレーブ送信/受信)(RJJ05B1604)

(この要因が該当するかわからない場合)

E8aエミュレータなどのデバッガを使用またはプログラムで、UiMRレジスタ、UiSMRレジスタ、UiSMR2レジスタ、UiSMR3レジスタを設定する前まで動作させてください。その後、1命令ずつ実行し、割り込みが発生するか確認してください。

UiMRレジスタ、UiSMRレジスタ、UiSMR2レジスタ、UiSMR3レジスタを設定する箇所での割り込みが発生した場合、本要因が該当しています。

また、「4. 解析手法」の「4.3 割り込み発生箇所を確認する」の節を参照してください。

1.1.2 割り込み要因を切り替えていませんか

割り込み要因を切り替えていませんか。切り替える場合は、割り込み禁止にしてから割り込み要因を切り替えていますか。

割り込み要因を切り替える際は、ユーザーズマニュアルハードウェア編の「特殊モード1 (I2Cモード)」の節の表「I2Cモード時の各機能」の注に記載の手順に従って切り替える必要があります。手順に従っていない場合、意図しない割り込みが発生する可能性があります。

◆解決方法

割り込み要因を切り替える場合、以下の手順で行ってください。

- (1) 要因を切り替える割り込みを禁止する
- (2) 要因を切り替える
- (3) その割り込みの割り込み制御レジスタのIRビットを“0” (割り込みなし)にする
- (4) その割り込みの割り込み制御レジスタのILVL2~ILVL0を設定する

◆アプリケーションノート

UARTi特殊モード1を使用したI2C-busインタフェース(マスタ送信/受信)(RJJ05B1596)

UARTi特殊モード1を使用したI2C-busインタフェース(スレーブ送信/受信)(RJJ05B1604)

(この要因が該当するかわからない場合)

E8aエミュレータなどのデバッガを使用またはプログラムで、割り込み要因を切り替える処理の前まで動作させてください。その後、1命令ずつ実行し、割り込みが発生するか確認してください。

割り込み要因を切り替える箇所で割り込みが発生した場合、本要因が該当しています。また、「4. 解析手法」の「4.3 割り込み発生箇所を確認する」の節を参照してください。

1.2 Fast-mode で通信できない

現象の例 :400kbps でデータを受信すると、正しいデータが受信できない。など

1.2.1 UiBRG カウントソースは、10MHz 以上になっていますか

I²C-bus 規格の Fast-mode で使用するとき、スタートコンディション/ストップコンディションのセットアップ時間、ホールド時間を確保できるよう UiBRG カウントソースは 10MHz 以上になっていますか。

I²C-bus 規格の Fast-mode において、スタートコンディション/ストップコンディションのセットアップ時間、ホールド時間はどちらも最小 600ns です。それに対して、マイコンがスタートコンディション、ストップコンディションを検出するには、セットアップ時間、ホールド時間ともに UiBRG カウントソースの 6 サイクル以上必要となります。確保できるセットアップ時間、ホールド時間は、UiBRG カウントソースが 10MHz のときに 600ns となり、これより遅い場合 (10MHz 未満) は、I²C-bus 規格を満たすことができなくなります。

このため、I²C-bus 規格の Fast-Mode の仕様を満たすためには、UiBRG カウントソースは 10MHz 以上である必要があります。

◆解決方法

Fast-Mode で使用する場合、UiBRG カウントソースは 10MHz 以上にしてください。

◆アプリケーションノート

UARTi 特殊モード 1 を使用した I²C バスインタフェース (RJJ05B1545)

UARTi 特殊モード 1 を使用した I²C-bus インタフェース (マスタ送信/受信) (RJJ05B1596)

UARTi 特殊モード 1 を使用した I²C-bus インタフェース (スレーブ送信/受信) (RJJ05B1604)

(この要因が該当するかわからない場合)

E8a エミュレータなどのデバッガを使用またはプログラムで、システムクロック選択ビット (CM0 レジスタの CM07 ビット、CM1 レジスタの CM11 ビット、CM2 レジスタの CM21 ビット) や UiBRG カウントソース選択ビット (Uic0 レジスタの CLK1 ~ CLK0 ビット) などを確認し、CPU クロックおよびカウントソースの設定を確認してください。

UiBRG カウントソースの設定が 10MHz 未満となっている場合は、本要因が該当しています。

また、「4. 解析手法」の「4.1 レジスタが期待どおり設定できているか確認する」、「4.2 動作中の周辺機能クロック f1 の周波数を確認する」の節を参照してください。

1.2.2 SDAiデジタル遅延値の設定は適切ですか

SDAiデジタル遅延値設定ビット(UiSMR3レジスタのDL2～DL0ビット)の設定は、スタートコンディション/ストップコンディションのセットアップ時間、ホールド時間を考慮して設定していますか。

I²C-bus規格のFast-modeにおいて、スタートコンディション/ストップコンディションのセットアップ時間、ホールド時間は、どちらも最小600nsです。SDAiデジタル遅延値設定ビットは、I²C-bus規格を満たすよう設定する必要があります。

◆解決方法

SDAiデジタル遅延値設定ビット(UiSMR3レジスタのDL2～DL0ビット)は、スタートコンディション/ストップコンディションのセットアップ時間、ホールド時間の規格を満たすよう設定してください。

◆アプリケーションノート

UARTi特殊モード1を使用したI²Cバスインタフェース(RJJ05B1545)

UARTi特殊モード1を使用したI²C-busインタフェース(マスタ送信/受信)(RJJ05B1596)

UARTi特殊モード1を使用したI²C-busインタフェース(スレーブ送信/受信)(RJJ05B1604)

(この要因が該当するかわからない場合)

プログラムを変更し、SDAiデジタル遅延値設定ビット(UiSMR3レジスタのDL2～DL0ビット)によるSDAi出力の遅延量を小さくして、Fast-modeで通信できるか確認してください。または、通信時のSCLi端子、SDAi端子をオシロスコープで確認してください。

SDAi出力の遅延量を小さくすることでFast-modeで通信できるようになった場合は、本要因が該当している可能性があります。スタートコンディション出力4.1レジスタが期待どおり設定できているか確認する時、SCLi端子が“H”から“L”に変化した後にSDAi端子が“H”から“L”に変化していたり、データ出力時、SCLi端子“H”中にSDAi端子が変化している場合は、本要因が該当しています。

また、「4. 解析手法」の「4.1 レジスタが期待どおり設定できているか確認する」、「4.5 簡易I2Cモード通信時の信号に問題がないか確認する」の節を参照してください。

1.3 各コンディションが検出できない

現象の例 : スタートコンディションが検出できない。
リスタートコンディションが検出できない。
ストップコンディションが検出できない。など

1.3.1 入力レベルは推奨動作条件を満たしていますか

入力している“L”や“H”のレベルは推奨動作条件を満たしていますか。
“L”入力電圧、“H”入力電圧、および、“L”出力電圧の規格は、I²C-bus規格と異なります。I²C-bus規格を満たしていても、ユーザーズマニュアルハードウェア編の電気的特性の推奨動作条件を満たしていない場合、各コンディションを正しく認識しない可能性があります。

◆解決方法

SCLi、SDAi端子と共用する入出力ポートの推奨動作条件を参照し、規格を満たすようにしてください。

◆アプリケーションノート

UARTi特殊モード1を使用したI²Cバスインタフェース(RJJ05B1545)
UARTi特殊モード1を使用したI²C-busインタフェース(マスタ送信/受信)(RJJ05B1596)
UARTi特殊モード1を使用したI²C-busインタフェース(スレーブ送信/受信)(RJJ05B1604)

(この要因が該当するかわからない場合)

通信時のSCLi端子、SDAi端子をオシロスコープで確認してください。
入力電圧がユーザーズマニュアルハードウェア編の電気的特性の推奨動作条件を満たしていない場合、本要因が該当している可能性があります。
また、「4. 解析手法」の「4.5 簡易I2Cモード通信時の信号に問題がないか確認する」の節を参照してください。

1.3.2 UiBRG カウントソースは、10MHz 以上になっていますか

I²C-bus 規格の Fast-mode で使用するとき、スタートコンディション/ストップコンディションのセットアップ時間、ホールド時間を確保できるよう UiBRG カウントソースは 10MHz 以上になっていますか。

I²C-bus 規格の Fast-mode において、スタートコンディション/ストップコンディションのセットアップ時間、ホールド時間はどちらも最小 600ns です。それに対して、マイコンがスタートコンディション、ストップコンディションを検出するには、セットアップ時間、ホールド時間ともに UiBRG カウントソースの 6 サイクル以上必要となります。確保できるセットアップ時間、ホールド時間は、UiBRG カウントソースが 10MHz のときに 600ns となり、これより遅い場合 (10MHz 未満) は、I²C-bus 規格を満たすことができなくなります。

このため、I²C-bus 規格の Fast-Mode の仕様を満たすためには、UiBRG カウントソースは 10MHz 以上である必要があります。

◆解決方法

Fast-Mode で使用する場合、UiBRG カウントソースは 10MHz 以上にしてください。

◆アプリケーションノート

UARTi 特殊モード 1 を使用した I²C バスインタフェース (RJJ05B1545)

UARTi 特殊モード 1 を使用した I²C-bus インタフェース (マスタ送信/受信) (RJJ05B1596)

UARTi 特殊モード 1 を使用した I²C-bus インタフェース (スレーブ送信/受信) (RJJ05B1604)

(この要因が該当するかわからない場合)

E8a エミュレータなどのデバッガを使用またはプログラムで、システムクロック選択ビット (CM0 レジスタの CM07 ビット、CM1 レジスタの CM11 ビット、CM2 レジスタの CM21 ビット) や UiBRG カウントソース選択ビット (Uic0 レジスタの CLK1 ~ CLK0 ビット) などを確認し、CPU クロックおよびカウントソースの設定を確認してください。

UiBRG カウントソースの設定が 10MHz 未満となっている場合は、本要因が該当しています。

また、「4. 解析手法」の「4.1 レジスタが期待どおり設定できているか確認する」、「4.2 動作中の周辺機能クロック f1 の周波数を確認する」の節を参照してください。

1.3.3 SDAi デジタル遅延値の設定は適切ですか

SDAi デジタル遅延値設定ビット (UiSMR3 レジスタの DL2~DL0 ビット) の設定は、スタートコンディション/ストップコンディションのセットアップ時間、ホールド時間を考慮して設定していますか。

I²C-bus 規格の Fast-mode において、スタートコンディション/ストップコンディションのセットアップ時間、ホールド時間は、どちらも最小 600ns です。SDAi デジタル遅延値設定ビットは、I²C-bus 規格を満たすよう設定する必要があります。

◆ 解決方法

SDAi デジタル遅延値設定ビット (UiSMR3 レジスタの DL2~DL0 ビット) は、スタートコンディション/ストップコンディションのセットアップ時間、ホールド時間の規格を満たすよう設定してください。

◆ アプリケーションノート

UARTi 特殊モード 1 を使用した I²C バスインタフェース (RJJ05B1545)

UARTi 特殊モード 1 を使用した I²C-bus インタフェース (マスタ送信/受信) (RJJ05B1596)

UARTi 特殊モード 1 を使用した I²C-bus インタフェース (スレーブ送信/受信) (RJJ05B1604)

(この要因が該当するかわからない場合)

プログラムを変更し、SDAi デジタル遅延値設定ビット (UiSMR3 レジスタの DL2~DL0 ビット) による SDAi 出力の遅延量を小さくして、Fast-mode で通信できるか確認してください。または、通信時の SCLi 端子、SDAi 端子をオシロスコープで確認してください。

SDAi 出力の遅延量を小さくすることで Fast-mode で通信できるようになった場合は、本要因が該当しています。スタートコンディションの出力で、SCLi 端子が“H”から“L”に変化した後に、SDAi 端子が“H”から“L”に変化している場合は、本要因が該当しています。

また、「4. 解析手法」の「4.1 レジスタが期待どおり設定できているか確認する」、「4.5 簡易 I2C モード通信時の信号に問題がないか確認する」の節を参照してください。

1.3.4 UiBRG レジスタの値が“03h”未満になっていませんか

UiBRG レジスタの値が“03h”未満になっていませんか。

内部回路が SCL クロックのレベルを認識するまで、最大で UiBRG カウントソースの 3 サイクルを要します。UiBRG レジスタに“03h”未満の値が設定されていると、自身が生成したコンディション波形が正しく認識できない場合があります。

◆ 解決方法

I²C モードでは、UiBRG レジスタに設定する値を“03h”以上にしてください。

◆ アプリケーションノート

UARTi 特殊モード 1 を使用した I²C バスインタフェース (RJJ05B1545)

UARTi 特殊モード 1 を使用した I²C-bus インタフェース (マスタ送信/受信) (RJJ05B1596)

UARTi 特殊モード 1 を使用した I²C-bus インタフェース (スレーブ送信/受信) (RJJ05B1604)

(この要因が該当するかわからない場合)

E8a エミュレータなどのデバッガを使用またはプログラムで、通信時の UiBRG レジスタの設定値を確認してください。

UiBRG レジスタの設定値が“03h”未満の場合、本要因が該当します。

また、「4. 解析手法」の「4.1 レジスタが期待どおり設定できているか確認する」の節を参照してください。

1.4 ビットずれが発生する

現象の例 : データを受信しても受信割り込みが発生しない。など

1.4.1 UiBRG レジスタの値が“03h”未満になっていませんか

UiBRG レジスタの値が“03h”未満になっていませんか。

内部回路が SCL クロックのレベルを認識するまで、最大で UiBRG カウントソースの 3 サイクルを要します。このため、UiBRG レジスタの値は、“03h”以上を設定する必要があります。したがって、接続可能な I²C-bus のビットレートは、UiBRG カウントソースの速度の 1/3 以下です。UiBRG レジスタに“00h”~“02h”を設定した場合は、ビットずれを起こす可能性があります。

◆解決方法

I²C モードでは、UiBRG レジスタに設定する値を“03h”以上にしてください。

◆アプリケーションノート

UARTi 特殊モード 1 を使用した I²C バスインタフェース (RJJ05B1545)

UARTi 特殊モード 1 を使用した I²C-bus インタフェース (マスタ送信/受信) (RJJ05B1596)

UARTi 特殊モード 1 を使用した I²C-bus インタフェース (スレーブ送信/受信) (RJJ05B1604)

(この要因が該当するかわからない場合)

E8a エミュレータなどのデバッガを使用またはプログラムで、通信時の UiBRG レジスタの設定値を確認してください。

UiBRG レジスタの設定値が“03h”未満の場合、本要因が該当します。

また、「4. 解析手法」の「4.1 レジスタが期待どおり設定できているか確認する」の節を参照してください。

1.5 正しいデータが受信できない

現象の例 : 受信割り込みで受信データを読み出すと期待するデータが受信できていない。など

1.5.1 受信割り込みでUiRBレジスタを読み出していませんか

UiSMR2レジスタのIICM2ビットを“1”(UART送信/受信割り込み)にして、受信割り込みでUiRBレジスタを読み出していませんか。

I²Cモードでは、割り込み要因にUART送信/UART受信割り込みを選択している場合、データを受信すると受信割り込みと送信割り込みの2つが発生します。それぞれ割り込みの発生タイミング、および、UiRBレジスタに格納されるデータの並びが異なります。

受信割り込みは、8ビットのデータ受信後、ACK/NACKビット送信前に発生します。

- 受信割り込み時のデータの並び

b15	...	b9	b8	b7							b0
			D0	—	D7	D6	D5	D4	D3	D2	D1

送信割り込みは、8ビットのデータとACK/NACKビットを送信中に発生します。

- 送信割り込み時のデータの並び

b15	...	b9	b8	b7							b0
			D8	D7	D6	D5	D4	D3	D2	D1	D0

◆解決方法

UiSMR2レジスタのIICM2ビットを“1”(UART送信/UART受信割り込み)にして、UART受信割り込み処理でUiRBレジスタを読み出した場合は、必要に応じてデータを加工してください。または、送信割り込み時に読み出してください。

◆アプリケーションノート

UARTi特殊モード1を使用したI²Cバスインタフェース(RJJ05B1545)

UARTi特殊モード1を使用したI²C-busインタフェース(マスタ送信/受信)(RJJ05B1596)

UARTi特殊モード1を使用したI²C-busインタフェース(スレーブ送信/受信)(RJJ05B1604)

(この要因が該当するかわからない場合)

受信データを読み出すタイミングを確認してください。

受信割り込みで受信データを読み出している場合、本要因が該当している可能性があります。

また、「4. 解析手法」の「4.5 簡易I2Cモード通信時の信号に問題がないか確認する」の節を参照してください。

1.5.2 送信割り込み要因は、送信完了にしていますか

UiC1レジスタのUiIRSビットを“1”（送信完了）にしていますか。
I²Cモードでは、必ずUiIRSビットを“1”にする必要があります。

◆解決方法

UiC1レジスタのUiIRSビットを“1”（送信完了）にしてください。

◆アプリケーションノート

UARTi特殊モード1を使用したI²Cバスインタフェース(RJJ05B1545)

UARTi特殊モード1を使用したI²C-busインタフェース(マスタ送信/受信)(RJJ05B1596)

UARTi特殊モード1を使用したI²C-busインタフェース(スレーブ送信/受信)(RJJ05B1604)

(この要因が該当するかわからない場合)

E8aエミュレータなどのデバッガを使用またはプログラムで、UiC1レジスタのUiIRSビットが“1”（送信完了）になっているか確認してください。

UiC1レジスタのUiIRSビットが“1”（送信完了）になっていない場合、本要因が該当していません。

また、「4. 解析手法」の「4.1 レジスタが期待どおり設定できているか確認する」、「4.5 簡易I²Cモード通信時の信号に問題がないか確認する」の節を参照してください。

1.6 オーバランエラーが発生する

現象の例 :連続でデータを受信すると、オーバランエラーが発生する。など

1.6.1 次のデータを受信する前にUiRBレジスタを読み出せていますか

CPUクロックの動作が遅かったり、他の機能の割り込みが発生するなどして、前のデータをUiRBレジスタから読み出す処理が実行される前に、次のデータを受信していませんか。

UiRBレジスタを読む前に次のデータ受信を開始し、次のデータの8ビット目を受信したことが考えられます。

◆解決方法

データを受信した場合、UiRBレジスタを読み出してから次の受信を開始(UiTBレジスタをライト)するようシステムを設計してください。

または、SCLウェイト自動挿入機能を有効(UiSMR2レジスタのSWCビットを“1”またはUiSMR4レジスタのSWC9ビットを“1”)にして、データ受信後にSCLi端子を“L”に固定してください。UiSMR2レジスタのSWCビットを“1”にした場合は、8ビット受信後にSCLi端子を“L”に、UiSMR4レジスタのSWC9ビットを“1”にした場合は、9ビット受信後にSCLi端子を“L”に固定できます。

◆アプリケーションノート

UARTi特殊モード1を使用したI²Cバスインタフェース(RJJ05B1545)

UARTi特殊モード1を使用したI²C-busインタフェース(マスタ送信/受信)(RJJ05B1596)

UARTi特殊モード1を使用したI²C-busインタフェース(スレーブ送信/受信)(RJJ05B1604)

(この要因が該当するかわからない場合)

受信データを読み出す直前にテストポートを反転させるプログラムを追加してください。オシロスコープでテストポートとRXDi端子を確認してください。

テストポートの反転周期よりRXDi端子に受信データが入力される周期の方が速い場合、本要因が該当している可能性があります。

1.7 SDAi端子が“L”に固定されたまま通信ができなくなる

現象の例 : 送信の途中でSDAi端子が“L”で停止した。など

1.7.1 ビットずれが発生していませんか

SCLi端子にノイズが発生、または、規格を満たしていない信号が入力されていませんか。

SCLi端子にノイズが発生、または、規格を満たしていない信号が入力された場合、マスタとスレーブ間でクロックの不一致(ビットずれ)が発生する可能性があります。

これは双方のデバイスの VIH/VIL 特性の違いやノイズ除去能力の差によって、SCLi端子に発生したノイズなどをクロックとして認識する/しないの差により発生します。

ビットずれが発生すると、SDAi端子が“L”固定になる可能性があります。

◆解決方法

通信先デバイスによって、SDAi端子が“L”に固定されている場合、一度シリアルインタフェース無効に設定し、SCLi/SDAi端子をプログラマブル入出力ポートに切り替えます。その後、SCLi端子に対応するポートから擬似的にクロックを出力(Hi-zと“L”出力で擬似クロック出力)し、SDAi端子が開放されるか確認してください。

一回の擬似クロック出力でSDAi端子を開放しない場合、SDAi端子を解放するまで繰り返し擬似クロック出力を行ってください。

SDAi端子の解放を確認後、再度I2Cモードに設定し、スタートコンディションとストップコンディションを発行して、一度通信を終了してください。

ほとんどのスレーブデバイスは、スタートコンディションとストップコンディションにより、I2C-busが初期化(リセット)されるため、再度通信をやり直すなどの処置を行ってください。

なお、スレーブデバイスのマニュアルに初期化方法(リセット方法)が明記されている場合には、マニュアルに従ってスレーブデバイスを初期化(リセット)するようにしてください。

◆アプリケーションノート

UARTi特殊モード1を使用したI2Cバスインタフェース(RJJ05B1545)

UARTi特殊モード1を使用したI2C-busインタフェース(マスタ送信/受信)(RJJ05B1596)

UARTi特殊モード1を使用したI2C-busインタフェース(スレーブ送信/受信)(RJJ05B1604)

(この要因が該当するかわからない場合)

ビットずれが発生すると、マスタ側またはスレーブ側のいずれかのデバイスからのデータが最後まで出力されず、途中(“0”を出力)で止まっていることが考えられます。

自デバイスをリセットして、SDAi端子の“L”固定が解放されるか確認してください。

SDAi端子の“L”固定が解放された場合は、自デバイスが原因でSDAi端子の“L”固定が発生している可能性があります。“L”固定が解放されない場合は、通信先のデバイスが原因でSDAi端子の“L”固定が発生している可能性があります。

自デバイスが原因でSDAi端子の“L”固定が発生していた場合は、送信条件を満たす直前(送信データの設定前など)で、E8aエミュレータやプログラムを変更してプログラムを停止させ、そのときのCLKi端子の状態をオシロスコープで確認してください。

UiC0レジスタのCKPOLビットが“0”のときに外部クロックが“L”の場合、またはUiC0レジスタのCKPOLビットが“1”のときに外部クロックが“H”の場合、本要因が該当している可能性があります。

また、「4. 解析手法」の「4.4 通信中に端子が“L”ホールドした時、どのデバイスが“L”になっているか確認する」、「4.5 簡易I2Cモード通信時の信号に問題がないか確認する」の節を参照してください。

1.8 期待どおりの転送速度にならない

現象の例 : カウントソースを考慮して、ビットレートを設定したが期待するビットレートよりも遅い。
など

1.8.1 クロック同期化機能によるサンプリング遅延を考慮しましたか

UiSMR2レジスタのCSCビットを“1”(クロック同期化機能を有効)にしたときに、クロック同期化の際に発生するサンプリング遅延を考慮しましたか。

CSCビットを“1”にした場合、ノイズフィルタ幅+UiBRGカウントソースの1~1.5サイクルのサンプリング遅延が発生し、SCLクロックの“H”認識が遅れるため、SCLクロックの“H”幅が伸びます。そのため、SCLクロックのビットレートの設定に対して、実際のSCLクロックは遅くなります。

◆解決方法

約100ns(注1)のノイズフィルタ幅+UiBRGカウントソースの1~1.5サイクルのサンプリング遅延を考慮して、ビットレートを設定してください。
転送速度の算出例は次のとおりです。

$$\text{転送速度} = 1 / ((2 \times (n+1)) / f_j) + 100\text{ns} + (1.5 \text{cycle} \times 1 / f_j) + t_F + t_R$$

f_j : UiBRGカウントソース

n : UiBRGレジスタ設定値

t_F : SCLクロック立ち下がり時間

t_R : SCLクロック立ち上がり時間

注1. 最大200ns

◆アプリケーションノート

UARTi特殊モード1を使用したI²Cバスインタフェース(RJJ05B1545)

UARTi特殊モード1を使用したI²C-busインタフェース(マスタ送信/受信)(RJJ05B1596)

UARTi特殊モード1を使用したI²C-busインタフェース(スレーブ送信/受信)(RJJ05B1604)

(この要因が該当するかわからない場合)

E8aエミュレータなどのデバッガを使用またはプログラムでUiSMR2レジスタのCSCビットの設定値を確認してください。

UiSMR2レジスタのCSCビット“1”(クロック同期化機能を有効)である場合、かつ、UiBRGカウントソースの1~1.5サイクルのサンプリング遅延を考慮して、ビットレートを設定していない場合、本要因が該当している可能性があります。

また、「4. 解析手法」の「4.1 レジスタが期待どおり設定できているか確認する」、「4.2 動作中の周辺機能クロックf_lの周波数を確認する」の節を参照してください。

2. I²Cモード マスタ動作時のトラブル

表 2.1 にトラブル事例と確認内容一覧を示します。確認内容の詳細と解決方法は、表の「参照」欄に示した項を参照してください。

表 2.1 トラブル事例と確認内容一覧

節	トラブル事例	確認内容	参照
2.1	各コンディションが正しく生成できない	各コンディション生成ビットの設定は、I2Cモードのときにしましたか	2.1.1
		コンディション生成回路の選択は、各コンディション生成ビットの設定後にしましたか	2.1.2
		複数のコンディションを同時に生成しようとしていませんか	2.1.3
		各コンディション生成ビットの設定は、STSPSELビットを“0”にした後1/2サイクル以上待ちましたか	2.1.4
		SDAi デジタル遅延機能による遅延量は適切ですか	2.1.5
		SCLi、SDAi 端子は、プルアップしていますか	2.1.6
		プルアップ抵抗値は適切ですか	2.1.7
2.2	SDAi 端子が“L”になる／“L”パルスが出力される	SDAi 出力の初期値は“H”に設定しましたか	2.2.1
		SCLi／SDAi 端子に対応するポートを“H”に設定しましたか	2.2.2
		ACK 出力していませんか	2.2.3
		シリアル入出力回路選択時、アイドル中はNACK出力設定にしていますか	2.2.4
		コンディション生成後は、シリアル入出力回路に戻しましたか	2.2.5
2.3	SCLi 端子が“L”になる／“L”パルスが出力される	SCLi／SDAi 端子に対応するポートを“H”に設定しましたか	2.3.1
		SCL ウェイト機能を使用時、ウェイトを解除しましたか	2.3.2
		UiSMR2 レジスタの SWC2 ビットが“1”になっていませんか	2.3.3
		スタートコンディション生成前にクロック位相を変更していませんか	2.3.4
		コンディション生成後は、シリアル入出力回路に戻しましたか	2.3.5
2.4	期待どおりデータが出力されない	クロック同期化機能を有効にしていますか	2.4.1
		ACK 出力していませんか	2.4.2
		コンディション生成後は、シリアル入出力回路に戻しましたか	2.4.3
		送信バッファフルの状態でのデータを設定していませんか	2.4.4
		送信割り込み要因は、送信完了にしていますか	2.4.5
		アービトレーションロストを検出していませんか	2.4.6
2.5	Fast-Mode で通信できない	UiBRG カウントソースは10MHz以上になっていますか	2.5.1
		通信ビットレートを400kbpsに設定していませんか	2.5.2
2.6	常にACKが返ってくる	送信データの9ビット目は、“1”に設定していますか	2.6.1
2.7	意図しないアービトレーションロストが発生する	アクノリッジビット受信時に発生していませんか	2.7.1
		データ受信時に発生していませんか	2.7.2
-	該当する現象がない／該当する現象がわからない	-	5.

2.1 各コンディションが正しく生成できない

現象の例 : スタートコンディション生成処理を実施しても、スタートコンディションが出力されない。
など

2.1.1 各コンディション生成ビットの設定は、I²Cモードのときにしましたか

I²Cモード(UiSMRレジスタのIICMビットが“1”)のときに、各コンディション生成ビットを設定しましたか。

各コンディション生成ビット (UiSMR4レジスタのSTAREQビット、RSTAREQビット、STPREQビット) は、I²Cモード(UiSMRレジスタのIICMビットが“1”)のとき“1”にできます。I²Cモード以外の場合は、“1”を書かないでください。

◆解決方法

各コンディションを生成する場合は、I²Cモード(UiSMRレジスタのIICMビットが“1”)に設定した状態で、各コンディション生成ビット (UiSMR4レジスタのSTAREQビット、RSTAREQビット、STPREQビット) を“1”にしてください。

◆アプリケーションノート

UARTi特殊モード1を使用したI²Cバスインタフェース(RJJ05B1545)

UARTi特殊モード1を使用したI²C-busインタフェース(マスタ送信/受信)(RJJ05B1596)

(この要因が該当するかわからない場合)

各コンディション生成ビットを設定する前に、I²Cモード(UiSMRレジスタのIICMビットが“1”)を選択しているかプログラムを確認してください。

各コンディション生成ビットを設定する前に、I²Cモードを選択していない場合、本要因が該当している可能性があります。

また、「4. 解析手法」の「4.1 レジスタが期待どおり設定できているか確認する」の節を参照してください。

2.1.2 コンディション生成回路の選択は、各コンディション生成ビットの設定後にしましたか

コンディション生成ビット (UiSMR4 レジスタの STAREQ ビット、RSTAREQ ビット、STPREQ ビット) を“1”にした後、コンディション生成回路を選択 (UiSMR4 レジスタの STSPSEL ビットを“1”に) しましたか。

ハードウェアによる自動生成をする場合、コンディション生成ビットの内、いずれか1つを“1”にした後に、UiSMR4 レジスタの STSPSEL ビットを“1”にする必要があります。

ハードウェアによる自動生成をしない場合は、STSPSEL ビットを“0”のまま、ポートを使ったプログラムで実現する必要があります。

◆解決方法

STSPSEL ビットを“0” で使用する場合は、スタートコンディション、ストップコンディションは、ポートを使ったプログラムで実現してください。

◆アプリケーションノート

UARTi 特殊モード1を使用した I²C バスインタフェース (RJJ05B1545)

UARTi 特殊モード1を使用した I²C-bus インタフェース (マスタ送信/受信) (RJJ05B1596)

(この要因が該当するかわからない場合)

コンディション生成回路を選択する前に、各コンディション生成ビットを設定しているかプログラムを確認してください。

コンディション生成回路を選択する前に、各コンディション生成ビットを設定していない場合、本要因が該当している可能性があります。

また、「4. 解析手法」の「4.1 レジスタが期待どおり設定できているか確認する」の節を参照してください。

2.1.3 複数のコンディションを同時に生成しようとしていませんか

コンディション生成時、複数のコンディション生成ビット (UiSMR4 レジスタの STAREQ、RSTAREQ、STPREQ ビット) を“1”にしていませんか。

複数のコンディション生成ビットを同時に“1”にしないでください。

◆解決方法

複数のコンディション生成を同時に選択しないでください。複数のコンディション生成が必要な場合は、コンディションごとに処理をしてください。

◆アプリケーションノート

UARTi 特殊モード1を使用した I²C バスインタフェース (RJJ05B1545)

UARTi 特殊モード1を使用した I²C-bus インタフェース (マスタ送信/受信) (RJJ05B1596)

(この要因が該当するかわからない場合)

コンディション生成時、複数のコンディション生成ビット (UiSMR4 レジスタの STAREQ、RSTAREQ、STPREQ ビット) を“1”にした状態で、コンディション生成回路を選択していないかプログラムを確認してください。

複数のコンディション生成ビット (UiSMR4 レジスタの STAREQ、RSTAREQ、STPREQ ビット) を“1”にした状態で、コンディション生成回路を選択していた場合、本要因が該当している可能性があります。

また、「4. 解析手法」の「4.1 レジスタが期待どおり設定できているか確認する」の節を参照してください。

2.1.4 各コンディション生成ビットの設定は、STSPSELビットを“0”にした後1/2サイクル以上待ちましたか

STSPSELビットを“0”にした後に、コンディション生成ビットを設定する場合は、STSPSELビットを“0”にしてから1/2サイクル以上待ちましたか。

例えば、ストップコンディションを生成後、次のスタートコンディションを生成する場合、UiSMR4レジスタのSTSPSELビットを一度“0”にした後、スタートコンディション生成ビットを設定する前に、SCLクロックの1/2サイクル以上待つ必要があります。

UiSMR4レジスタのSTSPSELビットを“0”にした後、SCLクロックの1/2サイクル以内に各コンディション生成ビット(STAREQ、RSTAREQ、STPREQ)を“0”から“1”にすると、各コンディションが正しく生成できない場合があります。

◆解決方法

UiSMR4レジスタのSTSPSELビットを“0”にした後、送受信クロックの1/2サイクル以上待つから、各コンディション生成ビット(STAREQ、RSTAREQ、STPREQ)を“1”にしてください。

◆アプリケーションノート

UARTi特殊モード1を使用したI²Cバスインタフェース(RJJ05B1545)

UARTi特殊モード1を使用したI²C-busインタフェース(マスタ送信/受信)(RJJ05B1596)

◆テクニカルアップデート

TN-16C-A130B/J

(この要因が該当するかわからない場合)

STSPSELビットを“0”にした後に、コンディション生成ビットを生成する場合は、STSPSELビットを“0”にしてから1/2サイクル以上待っているかプログラムを確認してください。

STSPSELビットを“0”にしてから1/2サイクル以上待っていない場合は、本要因が該当していません。

2.1.5 SDAi デジタル遅延機能による遅延量は適切ですか

SDAi デジタル遅延機能による遅延量(UiSMR3 レジスタのDL2～DL0 ビットの設定値)は適切ですか。送受信速度が速い場合やカウントソースを分周している場合、SDAi デジタル遅延量によっては、SDAi よりも先に SCLi が変化している可能性があります。(スタートコンディション生成時、SCLi 端子が“L”になってから SDAi 端子が“L”に変化する。など)

◆ 解決方法

次の3つの条件を満たすよう SDAi デジタル遅延量を設定してください。

- コンディション生成時は、SCLi 端子レベルが“H”の間に、SDAi 端子が変化すること
- データ出力時は、SCLi 端子レベルが“L”の間に、SDAi 端子が変化すること
- セットアップ時間、ホールド時間などの電気的特性および規格を満たすこと

実際の出力波形を確認し、十分に余裕を持った設定をしてください。

◆ アプリケーションノート

UARTi 特殊モード1を使用したI²Cバスインタフェース(RJJ05B1545)

UARTi 特殊モード1を使用したI²C-busインタフェース(マスタ送信/受信)(RJJ05B1596)

(この要因が該当するかわからない場合)

プログラムを変更し、SDAi デジタル遅延値設定ビット(UiSMR3 レジスタのDL2～DL0 ビット)による SDAi 出力の遅延量を変更して、正常にコンディション生成できるか確認してください。または、通信時の SCLi 端子、SDAi 端子をオシロスコープで確認してください。

SDAi 出力の遅延量を変更することでコンディションが生成できるようになった場合は、本要因が該当している可能性があります。スタートコンディションの出力で、SCLi 端子が“H”から“L”に変化した後に、SDAi 端子が“H”から“L”に変化している場合は、本要因が該当しています。

また、「4. 解析手法」の「4.1 レジスタが期待どおり設定できているか確認する」、「4.5 簡易I2Cモード通信時の信号に問題がないか確認する」の節を参照してください。

2.1.6 SCLi、SDAi端子は、プルアップしていますか

SCLi、SDAi端子は、Nチャンネルオープンドレイン出力(UiC0レジスタのNCHビットが“1”)に設定し、その端子をプルアップしていますか。

I²C-bus通信では、他のデバイスとワイヤードANDで接続するため、SCLi、SDAi端子をNチャンネルオープンドレイン出力端子にする必要があります。このため、SCLi、SDAi端子はプルアップする必要があります。

◆解決方法

I²C-busで通信する場合は、Nチャンネルオープンドレイン出力(UiC0レジスタのNCHビットを“1”)に設定し、SCLi、SDAi端子をプルアップしてください。

◆アプリケーションノート

UARTi特殊モード1を使用したI²Cバスインタフェース(RJJ05B1545)

UARTi特殊モード1を使用したI²C-busインタフェース(マスタ送信/受信)(RJJ05B1596)

(この要因が該当するかわからない場合)

E8aエミュレータなどのデバッガを使用またはプログラムでUiC0レジスタのNCHビットの設定値を確認してください。

UiC0レジスタのNCHビットが“0”(SDAi、SCLi端子はCMOS出力)である場合、本要因が該当しています。

また、「4. 解析手法」の「4.1 レジスタが期待どおり設定できているか確認する」、「4.5 簡易I²Cモード通信時の信号に問題がないか確認する」、「4.6 Nチャンネルオープンドレイン出力端子がプルアップできているか確認する」の節を参照してください。

2.1.7 プルアップ抵抗値は適切ですか

SCLi、SDAi端子のプルアップに使用している抵抗値は適切ですか。

Nチャンネルオープンドレイン出力時の“H”レベルは、プルアップによる“H”で実現しますので、“H”になるまでの時間は、抵抗値に依存します(抵抗値が大きくなるほど“H”になるまでの時間が遅くなる)。このため、抵抗値が大きすぎる場合、“H”幅が短くなったり、“H”になる前に次の“L”を出力してしまい、通信先のデバイスに“H”を送信できないなどの問題が発生する可能性があります。

◆解決方法

次の2つの条件を満たすよう、抵抗値を設定してください。

- 通信先デバイスで正しく“H”を認識するレベルまで入力電圧が上昇すること
- “H”レベルを保持する期間が電气的特性および規格を満たすこと

実際の出力波形を確認し、十分に余裕を持った設定をしてください。

◆アプリケーションノート

UARTi特殊モード1を使用したI²Cバスインタフェース(RJJ05B1545)

UARTi特殊モード1を使用したI²C-busインタフェース(マスタ送信/受信)(RJJ05B1596)

(この要因が該当するかわからない場合)

オシロスコープでSCLi端子とSDAi端子を確認してください。

SCLi端子とSDAi端子からの出力波形が大きくなり、通信先デバイスの規格が満たせていない場合は、本要因が該当しています。

また、「4. 解析手法」の「4.5 簡易I2Cモード通信時の信号に問題がないか確認する」の節を参照してください。

2.2 SDAi端子が“L”になる／“L”パルスが出力される

現象の例 : スタートコンディション生成前やストップコンディション生成後に短い“L”パルスが出力される。など

2.2.1 SDAi出力の初期値は“H”に設定しましたか

SDAi出力の初期値の設定として、対応するポートレジスタを“1”(“H”レベル)に設定してからI²Cモード(UiMRレジスタのSMD2～SMD0ビット)を選択しましたか。

SDAi出力の初期値は、ポートレジスタで設定することができます。

I²Cモード(UiMRレジスタのSMD2～SMD0ビット)を選択すると、UiSMR4レジスタのSTSPSELビットが“0”、かつ、ACKCビットが“0”のとき、シリアル入出力回路のレベル(SDAi端子の出力の初期値)が出力されます。このため、ポートレジスタを“0”にした場合、または、ポートレジスタを未設定とした場合(リセット後の値が“0”のため)、I²Cモードに設定するとSDA端子が“L”になります。

◆解決方法

シリアルインタフェース無効(UiMRレジスタのSMD2～SMD0ビットが“000b”)の状態で、SDAi端子に対応するポートレジスタを“1”に設定してください。その後、I²Cモード(UiMRレジスタのSMD2～SMD0ビットが“010b”)を選択してください。

◆アプリケーションノート

UARTi特殊モード1を使用したI²Cバスインタフェース(RJJ05B1545)

UARTi特殊モード1を使用したI²C-busインタフェース(マスタ送信/受信)(RJJ05B1596)

(この要因が該当するかわからない場合)

E8aエミュレータなどのデバッガを使用し、問題の現象が出るタイミングでプログラムを停止させてください。例えば、ストップコンディション生成時に“L”パルスが出る場合は、ストップコンディション生成前で、プログラムを停止させてください。プログラムを停止させた後、そのときのSDAi端子に対応するポートレジスタの値を確認してください。

ポートレジスタの値が“0”の場合、本要因が該当している可能性があります。

また、「4. 解析手法」の「4.5 簡易I²Cモード通信時の信号に問題がないか確認する」の節を参照してください。

2.2.2 SCLi/SDAi端子に対応するポートを“H”に設定しましたか

SCLi/SDAi端子に対応するポートを“H”(ポートレジスタを“1”や端子を共用している周辺機能の出力レベルを“H”にする。など)に設定してからI²Cモード(UiMRレジスタのSMD2~SMD0ビット)を選択しましたか。

一般的に、端子の機能や周辺機能の動作モードを切り替えると、内部スイッチング回路の切り替え信号の遅延の差で、機能や動作モードが切り替わるまでの数ns程度短いパルスが出ることがあります。

◆解決方法

シリアルインタフェース無効(UiMRレジスタのSMD2~SMD0ビットが“000b”)の状態、次の処理のいずれか、もしくは、両方を行ってください。その後、I²Cモード(UiMRレジスタのSMD2~SMD0ビットが“010b”)を選択してください。

- SCLi/SDAi端子に対応するポートレジスタを“1”にする。
- 共用している周辺機能の出力レベルを“H”にする。

◆アプリケーションノート

UARTi特殊モード1を使用したI²Cバスインタフェース(RJJ05B1545)

UARTi特殊モード1を使用したI²C-busインタフェース(マスタ送信/受信)(RJJ05B1596)

(この要因が該当するかわからない場合)

プログラムを変更し、SCLi/SDAi端子に対応するポートレジスタを“1”、共用している周辺機能の出力レベルを“H”に設定してください。オシロスコープでSCLi端子とSDAi端子を確認してください。

SCLi端子とSDAi端子からの“L”パルスが出なくなった場合、本要因が該当している可能性があります。

また、「4. 解析手法」の「4.5 簡易I²Cモード通信時の信号に問題がないか確認する」の節を参照してください。

2.2.3 ACK出力していませんか

UiSMR4レジスタのACKCビットを“1”、ACKDビットを“0”にして、ACK出力していませんか。

UiSMR4レジスタのACKCビットを“1”、ACKDビットを“0”にすると、ACK出力するため、SDAi端子の出力が“L”になります。

◆解決方法

ACK出力しない場合は、UiSMR4レジスタのACKDビットを“1”にして、“L”を出力しないようにしてください。

◆アプリケーションノート

UARTi特殊モード1を使用したI²Cバスインタフェース(RJJ05B1545)

UARTi特殊モード1を使用したI²C-busインタフェース(マスタ送信/受信)(RJJ05B1596)

(この要因が該当するかわからない場合)

E8aエミュレータなどのデバッガを使用またはプログラムで、UiSMR4レジスタのACKCビット、ACKDビットの値を確認してください。

UiSMR4レジスタのACKCビットを“1”、ACKDビットを“0”にしている場合、本要因が該当します。

また、「4. 解析手法」の「4.1 レジスタが期待どおり設定できているか確認する」、「4.5 簡易I²Cモード通信時の信号に問題がないか確認する」の節を参照してください。

2.2.4 シリアル入出力回路選択時、アイドル中はNACK出力設定にしていますか

スタートコンディション生成前など、UiSMR4 レジスタの STSPSEL ビットが“0”(シリアル入出力回路)の間、ACKC ビットおよび ACKD ビットを“1”にして、NACK 出力設定にしていますか。

UiSMR4 レジスタの STSPSEL ビットは“0”にすると、シリアル入出力回路に接続されます。シリアル入出力回路の状態(前回送信したデータの最終ビットが“L”の場合など)によって、“L”を出力する可能性があります。

UiSMR4 レジスタの ACKC ビット、ACKD ビットを“1”に設定し、NACK 出力状態にすることで、意図せず“L”出力しないように設定できます。

◆解決方法

スタートコンディション生成前など、UiSMR4 レジスタの STSPSEL ビットが“0”の間、ACKC ビット、ACKD ビットを“1”に設定してください。

◆アプリケーションノート

UARTi 特殊モード1を使用した I²C バスインタフェース (RJJ05B1545)

UARTi 特殊モード1を使用した I²C-bus インタフェース(マスタ送信/受信)(RJJ05B1596)

(この要因が該当するかわからない場合)

E8a エミュレータなどのデバッガを使用またはプログラムで、非コンディション生成時の ACKC ビットおよび ACKD ビットを確認してください。

UiSMR4 レジスタの ACKC ビット、ACKD ビットを“1”に設定していない場合、本要因が該当している可能性があります。

また、「4. 解析手法」の「4.1 レジスタが期待どおり設定できているか確認する」、「4.5 簡易 I2C モード通信時の信号に問題がないか確認する」の節を参照してください。

2.2.5 コンディション生成後は、シリアル入出力回路に戻しましたか

各コンディション生成後は、UiSMR4 レジスタの STSPSEL ビットを“0”(シリアル入出力回路)に戻していますか。

UiSMR4 レジスタの STSPSEL ビットを“1”にして、スタートコンディションを生成した後は、データ送信するときに STSPSEL ビットを“0”に戻す必要があります。

UiSMR4 レジスタの STSPSEL ビットを“1”にすると、SCLi、SDAi 端子は、コンディション生成する回路に接続されているため、スタートコンディション生成後は、“L”出力を継続します。

◆解決方法

データ送受信時は、UiSMR4 レジスタの STSPSEL ビットを“0”(シリアル入出力回路)に設定してください。

◆アプリケーションノート

UARTi 特殊モード1を使用した I²C バスインタフェース (RJJ05B1545)

UARTi 特殊モード1を使用した I²C-bus インタフェース(マスタ送信/受信)(RJJ05B1596)

(この要因が該当するかわからない場合)

E8a エミュレータなどのデバッガを使用またはプログラムで、データ送受信時の UiSMR4 レジスタの STSPSEL ビットを確認してください。

UiSMR4 レジスタの STSPSEL ビットを“0”に設定していない場合、本要因が該当している可能性があります。

また、「4. 解析手法」の「4.1 レジスタが期待どおり設定できているか確認する」、「4.5 簡易 I2C モード通信時の信号に問題がないか確認する」の節を参照してください。

2.3 SCLi端子が“L”になる／“L”パルスが出力される

現象の例 :8ビットのデータを受信後、SCLi端子が“L”になり、通信が停止した。など

2.3.1 SCLi／SDAi端子に対応するポートを“H”に設定しましたか

SCLi／SDAi端子に対応するポートを“H”(ポートレジスタを“1”や端子を共用している周辺機能の出力レベルを“H”にする。など)に設定してからI²Cモード(UiMRレジスタのSMD2～SMD0ビット)を選択しましたか。

一般的に、端子の機能や周辺機能の動作モードを切り替えると、内部スイッチング回路の切り替え信号の遅延の差で、機能や動作モードが切り替わるまでの数ns程度短いパルスが出ることがあります。

◆解決方法

シリアルインタフェース無効(UiMRレジスタのSMD2～SMD0ビットが“000b”)の状態、次の処理のいずれか、もしくは、両方を行ってください。その後、I²Cモード(UiMRレジスタのSMD2～SMD0ビットが“010b”)を選択してください。

- SCLi/SDAi端子に対応するポートレジスタを“1”にする。
- 共用している周辺機能の出力レベルを“H”にする。

◆アプリケーションノート

UARTi特殊モード1を使用したI²Cバスインタフェース(RJJ05B1545)

UARTi特殊モード1を使用したI²C-busインタフェース(マスタ送信/受信)(RJJ05B1596)

(この要因が該当するかわからない場合)

プログラムを変更し、SCLi／SDAi端子に対応するポートレジスタを“1”、共用している周辺機能の出力レベルを“H”に設定してください。オシロスコープでSCLi端子とSDAi端子を確認してください。

SCLi端子とSDAi端子からの“L”パルスが出なくなった場合、本要因が該当している可能性があります。

また、「4. 解析手法」の「4.5 簡易I²Cモード通信時の信号に問題がないか確認する」の節を参照してください。

2.3.2 SCL ウェイト機能を使用時、ウェイトを解除しましたか

UiSMR2 レジスタの SWC ビットを“1”にして、SCL ウェイト機能を使用時、受信データの読み出し、アクノリッジ生成後に SCL のウェイトを解除しましたか。または、UiSMR4 レジスタの SWC9 ビットを“1”にして、SCL ウェイト機能を使用時、アクノリッジ判定後に SCL のウェイトを解除しましたか。

UiSMR2 レジスタの SWC ビットが“1”の場合、8 ビット受信後、SCLi 端子を“L”に固定します。UiSMR4 レジスタの SWC9 ビットが“1”の場合、9 ビット受信後、SCLi 端子を“L”に固定します。

UiSMR2 レジスタの SWC ビット、UiSMR4 レジスタの SWC9 ビットいずれも“0”にすることで、“L”固定を解除できます。

◆解決方法

SCL ウェイト機能を使用時は、受信データの読み出しやアクノリッジ生成/判定など必要な処理を行った後に、UiSMR2 レジスタの SWC ビット、および、UiSMR4 レジスタの SWC9 ビットを“0”にして、“L”固定を解除してください。

◆アプリケーションノート

UARTi 特殊モード1を使用した I²C バスインタフェース (RJJ05B1545)

UARTi 特殊モード1を使用した I²C-bus インタフェース(マスタ送信/受信)(RJJ05B1596)

(この要因が該当するかわからない場合)

E8a エミュレータなどのデバッガを使用またはプログラムで、SCLi 端子が“L”固定されたときの UiSMR2 レジスタの SWC ビット、UiSMR4 レジスタの SWC9 ビットを確認してください。

UiSMR2 レジスタの SWC ビット、UiSMR4 レジスタの SWC9 ビットが“1”(SCLi 端子を“L”に固定)の場合、本要因が該当しています。

また、「4. 解析手法」の「4.1 レジスタが期待どおり設定できているか確認する」、「4.5 簡易 I2C モード通信時の信号に問題がないか確認する」の節を参照してください。

2.3.3 UiSMR2 レジスタの SWC2 ビットが“1”になっていませんか

UiSMR2 レジスタの SWC2 ビットを“1”にして、SCLi 端子を“L”に固定設定にしているませんか。

UiSMR2 レジスタの SWC2 ビットは、通常、使用する必要はありません。UiSMR2 レジスタの SWC2 ビットを“1”(“L”出力)にすると、送受信中でも SCLi 端子を“L”に固定できます。

UiSMR2 レジスタの SWC2 ビットを“0”にすることで、“L”固定を解除できます。

◆解決方法

UiSMR2 レジスタの SWC2 ビットを“1”にして、SCLi 端子を“L”固定にした場合は、必要な処理を行った後に、UiSMR2 レジスタの SWC2 ビットを“0”にして、“L”固定を解除してください。

◆アプリケーションノート

UARTi 特殊モード1を使用した I²C バスインタフェース (RJJ05B1545)

UARTi 特殊モード1を使用した I²C-bus インタフェース(マスタ送信/受信)(RJJ05B1596)

(この要因が該当するかわからない場合)

E8a エミュレータなどのデバッガを使用またはプログラムで、SCLi 端子が“L”固定されたときの UiSMR2 レジスタの SWC2 ビットを確認してください。

UiSMR2 レジスタの SWC2 ビットが“1”(SCLi 端子を“L”に固定)の場合、本要因が該当しています。

また、「4. 解析手法」の「4.1 レジスタが期待どおり設定できているか確認する」、「4.5 簡易 I2C モード通信時の信号に問題がないか確認する」の節を参照してください。

2.3.4 スタートコンディション生成前にクロック位相を変更していませんか

スタートコンディション生成前に、クロック位相設定ビット (UiSMR3 レジスタの CKPH ビット) を “1” (クロック遅れあり) にして、クロックの位相を変更していませんか。

UiSMR3 レジスタの CKPH ビットを “1” にすると、クロックの位相が変わり、SCLi 端子の初期値が “L” になります。このため、スタートコンディション生成前 (UiSMR4 レジスタの STSPSEL ビットが “0” でシリアル入出力回路が選択されている間) に、UiSMR3 レジスタの CKPH ビットを “1” にすると、SCLi 端子が “L” になり、正常にスタートコンディションが生成できません。

◆解決方法

スタートコンディション生成前に、UiSMR3 レジスタの CKPH ビットは、通常次のように変更してください。

- ・スタートコンディション生成前 (UiSMR4 レジスタの STSPSEL ビットが “0” の間)、 “0” にする。
- ・スタートコンディション生成後 (UiSMR4 レジスタの STSPSEL ビットを “1” にした後、 “0” にする前に)、 “1” にする。
- ・ストップコンディション生成後 (UiSMR4 レジスタの STSPSEL ビットを “1” にした後、 “0” にする前に)、 “0” にする。

◆アプリケーションノート

UARTi 特殊モード1を使用したI²Cバスインタフェース (RJJ05B1545)

UARTi 特殊モード1を使用したI²C-busインタフェース (マスタ送信/受信) (RJJ05B1596)

(この要因が該当するかわからない場合)

E8a エミュレータなどのデバッガを使用またはプログラムで、UiSMR3 レジスタの CKPH ビットの設定手順を確認してください。

UiSMR3 レジスタの CKPH ビットをスタートコンディション生成前に “1” にしたり、ストップコンディション生成前に “0” している場合、本要因が該当している可能性があります。

また、「4. 解析手法」の「4.1 レジスタが期待どおり設定できているか確認する」、「4.5 簡易I2Cモード通信時の信号に問題がないか確認する」の節を参照してください。

2.3.5 コンディション生成後は、シリアル入出力回路に戻しましたか

各コンディション生成後は、UiSMR4レジスタのSTSPSELビットを“0”(シリアル入出力回路)に戻していますか。

UiSMR4レジスタのSTSPSELビットを“1”にして、スタートコンディションを生成した後は、データ送信するときにSTSPSELビットを“0”に戻す必要があります。

UiSMR4レジスタのSTSPSELビットが“1”にすると、SCLi、SDAi端子は、コンディション生成する回路に接続されているため、スタートコンディション生成後は、“L”出力を継続します。

◆解決方法

データ送受信時は、UiSMR4レジスタのSTSPSELビットを“0”(シリアル入出力回路)に設定してください。

◆アプリケーションノート

UARTi特殊モード1を使用したI²Cバスインタフェース(RJJ05B1545)

UARTi特殊モード1を使用したI²C-busインタフェース(マスタ送信/受信)(RJJ05B1596)

(この要因が該当するかわからない場合)

E8aエミュレータなどのデバッガを使用またはプログラムで、データ送受信時のUiSMR4レジスタのSTSPSELビットを確認してください。

UiSMR4レジスタのSTSPSELビットを“0”に設定していない場合、本要因が該当している可能性があります。

また、「4. 解析手法」の「4.1 レジスタが期待どおり設定できているか確認する」、「4.5 簡易I2Cモード通信時の信号に問題がないか確認する」の節を参照してください。

2.4 期待どおりデータが出力されない

現象の例 : 連続でデータを送信すると、たまに1バイト送信データが抜けることがある。など

2.4.1 クロック同期化機能を有効にしていますか

クロック同期化機能を有効(UiSMR2レジスタのCSCビットを“1”)にしていますか。

クロック同期化機能は、他のデバイスがウェイトを挿入するなど、自身が出力したクロックと SCLi 端子に入力されたクロックが異なったとき、内部で生成するクロックを SCLi 端子から入力されるクロックと同期させるための機能です。

クロック同期化機能が無効(UiSMR2レジスタのCSCビットを“0”)にすると、通信先デバイスがウェイトを挿入した際に同期化が働かないため、正しくデータが出力されません。

例えば、クロック同期化機能が無効(UiSMR2レジスタのCSCビットを“0”)のとき、通信先デバイスがウェイトを挿入するためにクロックを“L”ホールドした場合でも、データを送信し続けます。このため、クロックが“L”ホールド中に出力したデータは、通信先デバイスで受信できません。また、送信途中で“L”ホールドが解除された場合、残りのクロックおよびデータを出力するため、ビットずれが発生する可能性があります。クロック同期化機能を有効(UiSMR2レジスタのCSCビットを“1”)にすることで、通信先デバイスがクロックを“L”ホールドした場合に通信が一時的に停止し、“L”ホールドを解除後に通信を再開できます。

◆解決方法

マスタとして使用する場合は、クロック同期化機能を有効(UiSMR2レジスタのCSCビットを“1”)にして使用してください。

◆アプリケーションノート

UARTi特殊モード1を使用したI²Cバスインタフェース(RJJ05B1545)

UARTi特殊モード1を使用したI²C-busインタフェース(マスタ送信/受信)(RJJ05B1596)

(この要因が該当するかわからない場合)

E8aエミュレータなどのデバッガを使用またはプログラムで、UiSMR2レジスタのCSCビットの設定値を確認してください。

UiSMR2レジスタのCSCビットが“0”の場合、本要因が該当している可能性があります。

オシロスコープで、SCLi端子とSDAi端子を確認し、通信先デバイスによって、“L”ホールドされていないか確認してください。

通信先デバイスによって、“L”ホールドされている場合、本要因が該当しています。

また、「4. 解析手法」の「4.1 レジスタが期待どおり設定できているか確認する」、「4.5 簡易I2Cモード通信時の信号に問題がないか確認する」の節を参照してください。

2.4.2 ACK出力していませんか

UiSMR4レジスタのACKCビットを“1”、ACKDビットを“0”にして、ACK出力していませんか。
UiSMR4レジスタのACKCビットを“1”、ACKDビットを“0”にすると、ACK出力するため、SDAi端子の出力が“L”になります。

◆解決方法

ACK出力しない場合は、UiSMR4レジスタのACKDビットを“1”にして、“L”を出力しないようにしてください。

◆アプリケーションノート

UARTi特殊モード1を使用したI²Cバスインタフェース(RJJ05B1545)

UARTi特殊モード1を使用したI²C-busインタフェース(マスタ送信/受信)(RJJ05B1596)

(この要因が該当するかわからない場合)

E8aエミュレータなどのデバッガを使用またはプログラムで、UiSMR4レジスタのACKCビット、ACKDビットの値を確認してください。

UiSMR4レジスタのACKCビットを“1”、ACKDビットを“0”にしている場合、本要因が該当します。

また、「4. 解析手法」の「4.1 レジスタが期待どおり設定できているか確認する」、「4.5 簡易I2Cモード通信時の信号に問題がないか確認する」の節を参照してください。

2.4.3 コンディション生成後は、シリアル入出力回路に戻しましたか

各コンディション生成後は、UiSMR4レジスタのSTSPSELビットを“0”(シリアル入出力回路)に戻していますか。

UiSMR4レジスタのSTSPSELビットを“1”にして、スタートコンディションを生成した後は、データ送信するときにSTSPSELビットを“0”に戻す必要があります。

UiSMR4レジスタのSTSPSELビットを“1”にすると、SCLi、SDAi端子は、コンディション生成する回路に接続されているため、スタートコンディション生成後は、“L”出力を継続します。

◆解決方法

データ送受信時は、UiSMR4レジスタのSTSPSELビットを“0”(シリアル入出力回路)に設定してください。

◆アプリケーションノート

UARTi特殊モード1を使用したI²Cバスインタフェース(RJJ05B1545)

UARTi特殊モード1を使用したI²C-busインタフェース(マスタ送信/受信)(RJJ05B1596)

(この要因が該当するかわからない場合)

E8aエミュレータなどのデバッガを使用またはプログラムで、データ送受信時のUiSMR4レジスタのSTSPSELビットを確認してください。

UiSMR4レジスタのSTSPSELビットを“0”に設定していない場合、本要因が該当している可能性があります。

また、「4. 解析手法」の「4.1 レジスタが期待どおり設定できているか確認する」、「4.5 簡易I2Cモード通信時の信号に問題がないか確認する」の節を参照してください。

2.4.4 送信バッファフルの状態でのデータを設定していませんか

送信バッファフルの状態でのデータを設定していませんか。

送信バッファレジスタにデータが残っているとき(UiC1レジスタのTIビットが“0”)に送信バッファレジスタ(UiTBレジスタ)に値を設定すると、送信バッファレジスタの値が上書きされ、送信データに抜けが発生したような現象が発生する可能性があります。

◆解決方法

送信バッファレジスタが空になったことを確認して送信データを設定する場合は、送信割り込みの割り込み要求ビットを確認して次の送信データを設定してください。

送信割り込みの割り込み要因は、UARTi送信割り込み要因選択ビットで選択できます。

◆アプリケーションノート

UARTi特殊モード1を使用したI2Cバスインタフェース(RJJ05B1545)

UARTi特殊モード1を使用したI2C-busインタフェース(マスタ送信/受信)(RJJ05B1596)

(この要因が該当するかわからない場合)

送信バッファレジスタに値を設定した後、送信割り込み要求ビットを確認してから次の送信データが設定されているかプログラムを確認してください。

送信割り込み要求ビットで確認していない場合、本要因が該当している可能性があります。また、「4. 解析手法」の「4.5 簡易I2Cモード通信時の信号に問題がないか確認する」の節を参照してください。

2.4.5 送信割り込み要因は、送信完了にしていますか

UiC1レジスタのUiIRSビットを“1”(送信完了)にしていますか。

I2Cモードでは、必ずUiIRSビットを“1”にする必要があります。

◆解決方法

UiC1レジスタのUiIRSビットを“1”(送信完了)にしてください。

◆アプリケーションノート

UARTi特殊モード1を使用したI2Cバスインタフェース(RJJ05B1545)

UARTi特殊モード1を使用したI2C-busインタフェース(マスタ送信/受信)(RJJ05B1596)

UARTi特殊モード1を使用したI2C-busインタフェース(スレーブ送信/受信)(RJJ05B1604)

(この要因が該当するかわからない場合)

E8aエミュレータなどのデバッガを使用またはプログラムで、UiC1レジスタのUiIRSビットが“1”(送信完了)になっているか確認してください。

UiC1レジスタのUiIRSビットが“1”(送信完了)になっていない場合、本要因が該当しています。

また、「4. 解析手法」の「4.1 レジスタが期待どおり設定できているか確認する」、「4.5 簡易I2Cモード通信時の信号に問題がないか確認する」の節を参照してください。

2.4.6 アービトレーションロストを検出していませんか

UiSMR2レジスタのALSビットが“1”(SDA出力自動停止機能が許可)のときに、UiRBレジスタのABTビットが“1”(アービトレーションロスト検出)の状態ではデータを送信していませんか。

UiSMR2レジスタのALSビットを“1”(SDA出力自動停止機能が許可)にしている場合、アービトレーションロストを検出するとSDAi端子からの出力が停止します。UiRBレジスタのABTビットが“1”(アービトレーションロスト検出)のまま、送信するとSDAi端子からデータが出力されません。

◆解決方法

UiSMR2レジスタのALSビットが“1”(SDA出力自動停止機能が許可)で使用する場合、UiRBレジスタのABTビットが“1”(アービトレーションロスト検出)になった後は、通信を再開する前に“0”(アービトレーションロスト未検出)にしてください。

◆アプリケーションノート

UARTi特殊モード1を使用したI²Cバスインタフェース(RJJ05B1545)

UARTi特殊モード1を使用したI²C-busインタフェース(マスタ送信/受信)(RJJ05B1596)

(この要因が該当するかわからない場合)

E8aエミュレータなどのデバッガを使用またはプログラムで、データが出力されなくなったときのUiRBレジスタのABTビットを確認してください。

UiRBレジスタのABTビットが“1”の場合、本要因が該当している可能性があります。

2.5 Fast-Modeで通信できない

現象の例 :400kbpsに設定し、送信すると通信先のデバイスで受信ができていない。など

2.5.1 UiBRG カウントソースは10MHz以上になっていますか

UiBRG カウントソースを10MHz以上に設定し、スタートコンディション/ストップコンディションのセットアップ時間、ホールド時間を600ns以上確保できていますか。

I²C-bus規格のFast-Modeにおいて、スタートコンディション/ストップコンディションのセットアップ時間、ホールド時間はどちらも最小600nsです。それに対して、マイコンがスタートコンディション、ストップコンディションを検出するには、セットアップ時間、ホールド時間ともにUiBRG カウントソースの6サイクル以上必要となります。確保できるセットアップ時間、ホールド時間は、UiBRG カウントソースが10MHzのときに600nsとなり、これより遅い場合(10MHz未満)は、I²C-bus規格を満たすことができなくなります。

このため、I²C-bus規格のFast-Modeの仕様を満たすためには、UiBRG カウントソースは10MHz以上である必要があります。

◆解決方法

Fast-Modeで使用する場合は、UiBRG カウントソースは10MHz以上にしてください。

◆アプリケーションノート

UARTi特殊モード1を使用したI²Cバスインタフェース(RJJ05B1545)

UARTi特殊モード1を使用したI²C-busインタフェース(マスタ送信/受信)(RJJ05B1596)

(この要因が該当するかわからない場合)

E8aエミュレータなどのデバッガを使用またはプログラムで、システムクロック選択ビット(CM0レジスタのCM07ビット、CM1レジスタのCM11ビット、CM2レジスタのCM21ビット)やUiBRG カウントソース選択ビット(UiC0レジスタのCLK1~CLK0ビット)などを確認し、CPUクロックおよびカウントソースの設定を確認してください。

UiBRG カウントソースの設定が10MHz未満となっている場合は、本要因が該当しています。

また、「4. 解析手法」の「4.1 レジスタが期待どおり設定できているか確認する」、「4.2 動作中の周辺機能クロックf1の周波数を確認する」の節を参照してください。

2.5.2 通信ビットレートを400kbpsに設定していませんか

通信ビットレートを400kbpsに設定し、SCLクロックの“L”幅が1.3 μ s未満になっていませんか。
I²Cモードで生成するSCLクロックのDutyは50%です。I²C-busのFast-Modeの最大SCLクロック(400kbps)を設定すると、SCLクロックの“L”幅は1.25 μ sとなります。この値は、Fast-ModeのI²C-bus規格(fLOW = Min.1.3 μ s)を満たしません。

◆解決方法

SCLクロックの設定を384.6kbps以下とし、SCLクロックの“L”幅が1.3 μ s以上になるようにしてください。ユーザーズマニュアルハードウェア編の「特殊モード1 (I²Cモード)」の節の「SCLクロックの周波数の考え方」の内容を参照してください。

◆アプリケーションノート

UARTi特殊モード1を使用したI²Cバスインタフェース(RJJ05B1545)

UARTi特殊モード1を使用したI²C-busインタフェース(マスタ送信/受信)(RJJ05B1596)

(この要因が該当するかわからない場合)

E8aエミュレータなどのデバッガを使用またはプログラムで、システムクロック選択ビット(CM0レジスタのCM07ビット、CM1レジスタのCM11ビット、CM2レジスタのCM21ビット)やUiBRGカウントソース選択ビット(UiC0レジスタのCLK1~CLK0ビット)などを確認し、通信ビットレートの設定が384.6kbpsになっているか確認してください。

384.6kbps以下になっていない場合、Fast-ModeのI²C-bus規格であるfLOW = Min.1.3 μ sがみだせないため、本要因が該当します。

また、「4. 解析手法」の「4.1 レジスタが期待どおり設定できているか確認する」、「4.2 動作中の周辺機能クロックf1の周波数を確認する」、「4.5 簡易I2Cモード通信時の信号に問題がないか確認する」の節を参照してください。

2.6 常にACKが返ってくる

現象の例 : 存在しないデバイスを指定した場合でもACKが返ってくる。など

2.6.1 送信データの9ビット目は、“1”に設定していますか

送信バッファレジスタへ送信データ書き込み時、9ビット目のデータを“0”に設定していませんか。または、送信データをバイトサイズでの書き込みで行っていませんか。

I²Cモードでは、送信データの9ビット目をACK/NACKビットとして送信します。このため、送信データの9ビット目を“0”に設定した場合、ACK/NACKビットのタイミングで“L”を出力するため、通信先デバイスからACKが返ってきたように見える可能性があります。また、送信データを1バイト単位で書き込んだ場合、送信バッファレジスタの9ビット目の状態によって、同様にACK/NACKビットを“L”出力する可能性があります。

◆解決方法

データ送信時は、9ビット目のデータを“1”にして、ワード単位で送信バッファレジスタへ書き込みを行ってください。

◆アプリケーションノート

UARTi特殊モード1を使用したI²Cバスインタフェース(RJJ05B1545)

UARTi特殊モード1を使用したI²C-busインタフェース(マスタ送信/受信)(RJJ05B1596)

(この要因が該当するかわからない場合)

E8aエミュレータなどのデバッガを使用またはプログラムで、送信バッファレジスタへの設定内容を確認してください。

送信バッファレジスタへの書き込みで9ビット目を“0”にしている場合、または、1バイト単位で書き込みしている場合、本要因が該当している可能性があります。

2.7 意図しないアービトレーションロストが発生する

現象の例 : マルチマスタ通信をしていないときでもアービトレーションロストが発生する。など

2.7.1 アクノリッジビット受信時に発生していませんか

マスタ送信動作のデータ送信中のアクノリッジビット受信時、または、マスタ送信およびマスタ受信動作のスレーブアドレス送信中のアクノリッジビット受信時にアービトレーションロストを検出していませんか。

アクノリッジビットの受信は、UiTBレジスタの9ビット目を“1”にすることで、データ送信時の9ビット目にSDAi端子からの出力を解放し、通信先デバイスからのアクノリッジを受信します。このとき、ACKを受信すると、SDAi端子を解放するために設定したデータと受信データ(ACK)が一致しないことでアービトレーションロストが発生します。

◆解決方法

アクノリッジビット受信時に発生したアービトレーションロストは、無視するようにしてください。

また、UiSMR2レジスタのALSビットを“1”(SDA出力自動停止機能が許可)にしている場合は、アービトレーションロストを検出するとSDAi端子からの出力が停止してしまいますので、UiRBレジスタのABTビットを“0”(アービトレーションロスト未検出)にした後、次の通信をするようにしてください。

◆アプリケーションノート

UARTi特殊モード1を使用したI²Cバスインタフェース(RJJ05B1545)

UARTi特殊モード1を使用したI²C-busインタフェース(マスタ送信/受信)(RJJ05B1596)

(この要因が該当するかわからない場合)

E8aエミュレータなどのデバッガを使用またはプログラムで、アービトレーションロストが発生したタイミングを確認してください。

アクノリッジビット受信時に、アービトレーションロストが発生している場合は、本要因が該当しています。

2.7.2 データ受信時に発生していませんか

マスタ受信動作のデータ受信時にアービトレーションロストを検出していませんか。

データの受信は、UiTBレジスタに9ビットデータの“00FFh”または“01FFh”(b8はACKを生成する場合は“0”、NACKを生成する場合は“1”を設定)を設定することでSDAi端子を解放し、通信先デバイスからのデータを受信します。このとき、SDAi端子を解放するために設定したデータと受信データが一致しないことでアービトレーションロストが発生します。

◆解決方法

データの受信時に発生したアービトレーションロストは、無視するようにしてください。

また、UiSMR2レジスタのALSビットを“1”(SDA出力自動停止機能が許可)にしている場合は、アービトレーションロストを検出するとSDAi端子からの出力が停止してしまいますので、UiRBレジスタのABTビットを“0”(アービトレーションロスト未検出)にした後、次の通信をするようにしてください。

◆アプリケーションノート

UARTi特殊モード1を使用したI²Cバスインタフェース(RJJ05B1545)

UARTi特殊モード1を使用したI²C-busインタフェース(マスタ送信/受信)(RJJ05B1596)

(この要因が該当するかわからない場合)

E8aエミュレータなどのデバッガを使用またはプログラムで、アービトレーションロストが発生したタイミングを確認してください。

データの受信時に、アービトレーションロストが発生している場合は、本要因が該当しています。

3. I²Cモードスレーブ動作時のトラブル

表 3.1 にトラブル事例と確認内容一覧を示します。確認内容の詳細と解決方法は、表の「参照」欄に示した項を参照してください。

表 3.1 トラブル事例と確認内容一覧

節	トラブル事例	確認内容	参照
3.1	期待どおりのデータが出力されない	ACKビットおよび送信データの設定は間に合っていますか	3.1.1
		送信割り込み要因は、送信完了にしていますか	3.1.2
		リスタートコンディションを検出していませんか	3.1.3
		ビットずれが発生していませんか	3.1.4
3.2	ビットずれが発生する	ストップコンディション検出後、UARTモジュールを初期化しましたか	3.2.1
3.3	各コンディションが検出されない	SDAiのセットアップ時間、ホールド時間は確保できていますか	3.3.1
		リスタートコンディションを検出していませんか	3.3.2
-	該当する現象がない／該当する現象がわからない	-	5.

3.1 期待どおりのデータが出力されない

現象の例：リスタートコンディション受信後、データが受信できなくなる。など

3.1.1 ACKビットおよび送信データの設定は間に合っていますか

ACKビットや送信データを設定する前に通信先デバイスからクロックが入力されていませんか。
ACKビットの設定や送信データを設定する前に通信先デバイスからクロックが入力されると、意図しないデータが出力される可能性があります。

◆解決方法

SWCビットが“1”の場合、8ビット受信後、SCLi端子を“L”に固定します。SCLi端子を“L”に固定することで、他デバイスを待たせることができます。

ACKビットおよび送信データの設定が間に合わない場合は、SWCビットを“1”にしてください。

ACKビットおよび送信データを設定した後、SWCビットを“0”にして、“L”固定を解除してください。

◆アプリケーションノート

UARTi特殊モード1を使用したI²Cバスインタフェース(RJJ05B1545)

UARTi特殊モード1を使用したI²C-busインタフェース(スレーブ送信/受信)(RJJ05B1604)

(この要因が該当するかわからない場合)

送信データを設定する直前にテストポートを反転させるプログラムを追加してください。
オシロスコープでテストポートとSDAi端子を確認してください。

テストポートの反転周期よりSDAi端子から送信データが出力される周期の方が速い場合、本要因が該当している可能性があります。

また、「4. 解析手法」の「4.5 簡易I2Cモード通信時の信号に問題がないか確認する」の節を参照してください。

3.1.2 送信割り込み要因は、送信完了にしていますか

UiC1レジスタのUiIRSビットを“1”（送信完了）にしていますか。

I²Cモードでは、必ずUiIRSビットを“1”にする必要があります。

◆解決方法

UiC1レジスタのUiIRSビットを“1”（送信完了）にしてください。

◆アプリケーションノート

UARTi特殊モード1を使用したI²Cバスインタフェース(RJJ05B1545)

UARTi特殊モード1を使用したI²C-busインタフェース(マスタ送信/受信)(RJJ05B1596)

UARTi特殊モード1を使用したI²C-busインタフェース(スレーブ送信/受信)(RJJ05B1604)

(この要因が該当するかわからない場合)

E8aエミュレータなどのデバッガを使用またはプログラムで、UiC1レジスタのUiIRSビットが“1”（送信完了）になっているか確認してください。

UiC1レジスタのUiIRSビットが“1”（送信完了）になっていない場合、本要因が該当しています。

また、「4. 解析手法」の「4.1 レジスタが期待どおり設定できているか確認する」、「4.5 簡易I2Cモード通信時の信号に問題がないか確認する」の節を参照してください。

3.1.3 リスタートコンディションを検出していませんか

スレーブモード時に、リスタートコンディションを受信していませんか。
スレーブモード時、リスタートコンディションを検出すると、その後の処理を正しく実行しない場合があります。

◆解決方法

スレーブモード時は、リスタートコンディションを使用しないでください。

◆アプリケーションノート

UARTi特殊モード1を使用したI²Cバスインタフェース(RJJ05B1545)

UARTi特殊モード1を使用したI²C-busインタフェース(スレーブ送信/受信)(RJJ05B1604)

(この要因が該当するかわからない場合)

オシロスコープでSCL_i端子とSDA_i端子を確認してください。

スレーブ動作時に、リスタートコンディションが発生している場合、本要因が該当していません。

また、「4. 解析手法」の「4.5 簡易I2Cモード通信時の信号に問題がないか確認する」の節を参照してください。

3.1.4 ビットずれが発生していませんか

SCLi端子にノイズが発生、または、規格を満たしていない信号が入力されていませんか。

SCLi端子にノイズが発生、または、規格を満たしていない信号が入力された場合、マスタとスレーブ間でクロックの不一致(ビットずれ)が発生する可能性があります。

これは双方のデバイスのVih/Vil特性の違いやノイズ除去能力の差によって、SCLi端子に発生したノイズなどをクロックとして認識する/しないの差により発生します。

ビットずれが発生すると、SDAi端子が“L”固定になる可能性があります。

◆解決方法

通信先デバイスによって、SDAi端子が“L”に固定されている場合、一度シリアルインタフェース無効に設定し、SCLi/SDAi端子をプログラマブル入出力ポートに切り替えます。その後、SCLi端子に対応するポートから擬似的にクロックを出力(Hi-zと“L”出力で擬似クロック出力)し、SDAi端子が開放されるか確認してください。

一回の擬似クロック出力でSDAi端子を開放しない場合、SDAi端子を解放するまで繰り返し擬似クロック出力を行ってください。

SDAi端子の解放を確認後、再度I2Cモードに設定し、スタートコンディションとストップコンディションを発行して、一度通信を終了してください。

ほとんどのスレーブデバイスは、スタートコンディションとストップコンディションにより、I2C-busが初期化(リセット)されるため、再度通信をやり直すなどの処置を行ってください。

なお、スレーブデバイスのマニュアルに初期化方法(リセット方法)が明記されている場合には、マニュアルに従ってスレーブデバイスを初期化(リセット)するようにしてください。

◆アプリケーションノート

UARTi特殊モード1を使用したI2Cバスインタフェース(RJJ05B1545)

UARTi特殊モード1を使用したI2C-busインタフェース(スレーブ送信/受信)(RJJ05B1604)

(この要因が該当するかわからない場合)

ビットずれが発生すると、マスタ側またはスレーブ側のいずれかのデバイスからのデータが最後まで出力されず、途中(“0”を出力)で止まっていることが考えられます。

自デバイスをリセットして、SDAi端子の“L”固定が解放されるか確認してください。

SDAi端子の“L”固定が解放された場合は、自デバイスが原因でSDAi端子の“L”固定が発生している可能性があります。“L”固定が解放されない場合は、通信先のデバイスが原因でSDAi端子の“L”固定が発生している可能性があります。

自デバイスが原因でSDAi端子の“L”固定が発生していた場合は、送信条件を満たす直前(送信データの設定前など)で、E8aエミュレータやプログラムを変更してプログラムを停止させ、そのときのCLKi端子の状態をオシロスコープで確認してください。

UiC0レジスタのCKPOLビットが“0”のときに外部クロックが“L”の場合、またはUiC0レジスタのCKPOLビットが“1”のときに外部クロックが“H”の場合、本要因が該当している可能性があります。

また、「4. 解析手法」の「4.5 簡易I2Cモード通信時の信号に問題がないか確認する」の節を参照してください。

3.2 ビットずれが発生する

現象の例 : 連続データを受信しているとき、2フレーム目のデータが期待どおりに受信できない。など

3.2.1 ストップコンディション検出後、UARTモジュールを初期化しましたか

ストップコンディション検出後、UARTモジュールを初期化しましたか。

ストップコンディションのSCLの立ち上がりをクロックとして受信するため、ビットずれが発生します。

◆解決方法

スレーブ動作時は、ストップコンディション検出のたびにUARTモジュールを初期化してください。

◆アプリケーションノート

UARTi特殊モード1を使用したI²Cバスインタフェース(RJJ05B1545)

UARTi特殊モード1を使用したI²C-busインタフェース(スレーブ送信/受信)(RJJ05B1604)

(この要因が該当するかわからない場合)

ストップコンディション検出時の処理(割り込み処理など)で、UARTモジュールを初期化しているかプログラムを確認してください。

ストップコンディション検出時の処理(割り込み処理など)で、UARTモジュールを初期化していない場合、本要因が該当している可能性があります。

3.3 各コンディションが検出されない

現象の例 : スタートコンディションが検出されない。
リスタートコンディションが検出されない。
ストップコンディションが検出されない。など

3.3.1 SDAiのセットアップ時間、ホールド時間は確保できていますか

スタートコンディション生成時、SDAi立ち下がり後、SCLiが立ち下がるまでに6サイクル以上のホールド時間が確保できていますか。ストップコンディション生成時、SDAiが立ち上がる前に、SCLi端子が立ち上がってから6サイクル以上のセットアップ時間が確保できていますか。

セットアップ時間、ホールド時間が確保できていない場合、各コンディションは検出できない場合があります。

◆解決方法

セットアップ時間、ホールド時間は、UiBRGのカウントソースの6サイクル以上を確保してください。

◆アプリケーションノート

UARTi特殊モード1を使用したI²Cバスインタフェース(RJJ05B1545)

UARTi特殊モード1を使用したI²C-busインタフェース(スレーブ送信/受信)(RJJ05B1604)

(この要因が該当するかわからない場合)

オシロスコープでSCLi端子とSDAi端子を確認してください。

SDAiのセットアップ時間、ホールド時間を確認し、UiBRGのカウントソースの6サイクル以上が確保できているか確認してください。セットアップ時間、ホールド時間ともにUiBRGのカウントソースの6サイクル以上が確保できていない場合、本要因が該当しています。

また、「4. 解析手法」の「4.5 簡易I2Cモード通信時の信号に問題がないか確認する」の節を参照してください。

3.3.2 リスタートコンディションを検出していませんか

スレーブモード時に、リスタートコンディションを受信していませんか。

スレーブモード時、リスタートコンディションを検出すると、その後の処理を正しく実行しない場合があります。

◆解決方法

スレーブモード時は、リスタートコンディションを使用しないでください。

◆アプリケーションノート

UARTi特殊モード1を使用したI²Cバスインタフェース(RJJ05B1545)

UARTi特殊モード1を使用したI²C-busインタフェース(スレーブ送信/受信)(RJJ05B1604)

(この要因が該当するかわからない場合)

オシロスコープでSCLi端子とSDAi端子を確認してください。

スレーブ動作時に、リスタートコンディションが発生している場合、本要因が該当しています。

また、「4. 解析手法」の「4.5 簡易I2Cモード通信時の信号に問題がないか確認する」の節を参照してください。

4. 解析手法

本章では、以下の開発環境を使用した解析手法を記載しています。

- 統合開発環境：HEW
- デバッガ：E8aエミュレータデバッガ、E100エミュレータ

4.1 レジスタが期待どおり設定できているか確認する

レジスタや変数などが期待どおりに設定できているか確認する方法を紹介します。

4.1.1 デバッガで確認する方法

デバッガを使用することで、任意のタイミングでレジスタや変数の値を確認することができます。レジスタの設定前の値と設定後の値を確認して、期待どおりに設定できているか確認します。

◆ 確認手順

- (1) 確認したいレジスタを設定するコードにブレークポイントを設定する。
ただし、1つ前のコードがPRCRレジスタのPRC2ビットを“1”にするコードの場合は、そのコードとレジスタを設定するコードの次のコードにブレークポイントを設定する。
- (2) プログラムを実行し、ブレークポイントで停止させる。
- (3) レジスタを設定する前の値をメモリウィンドウやI/Oウィンドウなどで確認する。
- (4) ステップ実行し、レジスタ設定のコードを実行する。
ただし、PRCRレジスタのPRC2ビットでプロテクトされているレジスタの確認は、ステップ実行では確認できません。PRCRレジスタのPRC2ビットとその次のコードの間でプログラムを停止させないでください。
- (5) レジスタを設定した後の値をメモリウィンドウやI/Oウィンドウなどで確認し、期待どおり設定できているか確認する。

図 4.1 にデバッガを使用したレジスタの確認手順を、図 4.2 に PRC2 ビットでプロテクトされているレジスタの確認手順を示します。

◆ 判断方法

確認手順(5)のときにメモリウィンドウやI/Oウィンドウでレジスタに期待どおり値が設定できているか確認してください。期待どおりの値になっていない場合、レジスタにプロテクトが設定されていたり、設定順序が定められていることがありますので、ユーザーズマニュアルハードウェア編を参照し、レジスタの設定条件および設定手順を確認してください。

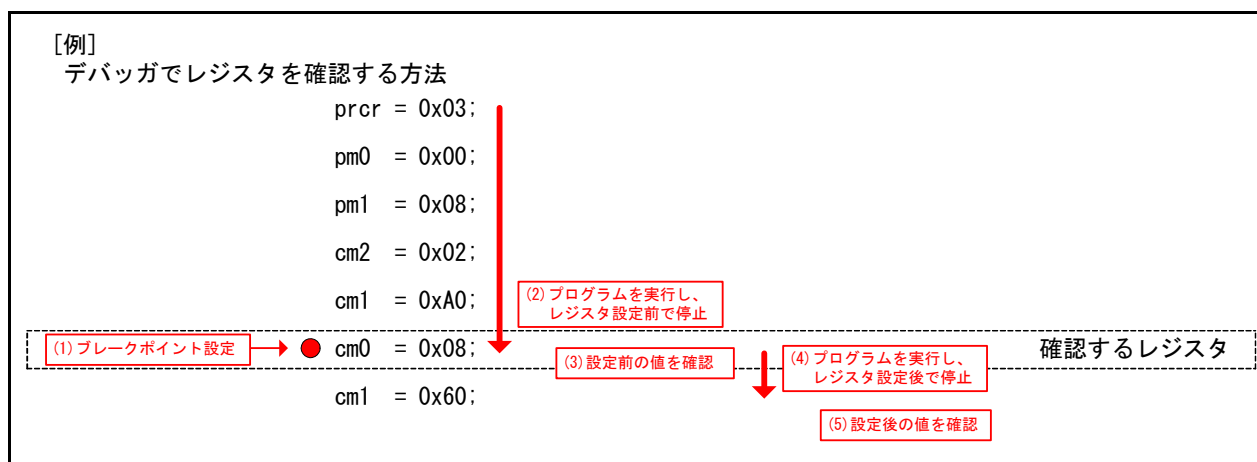


図 4.1 デバッガを使用したレジスタの確認手順

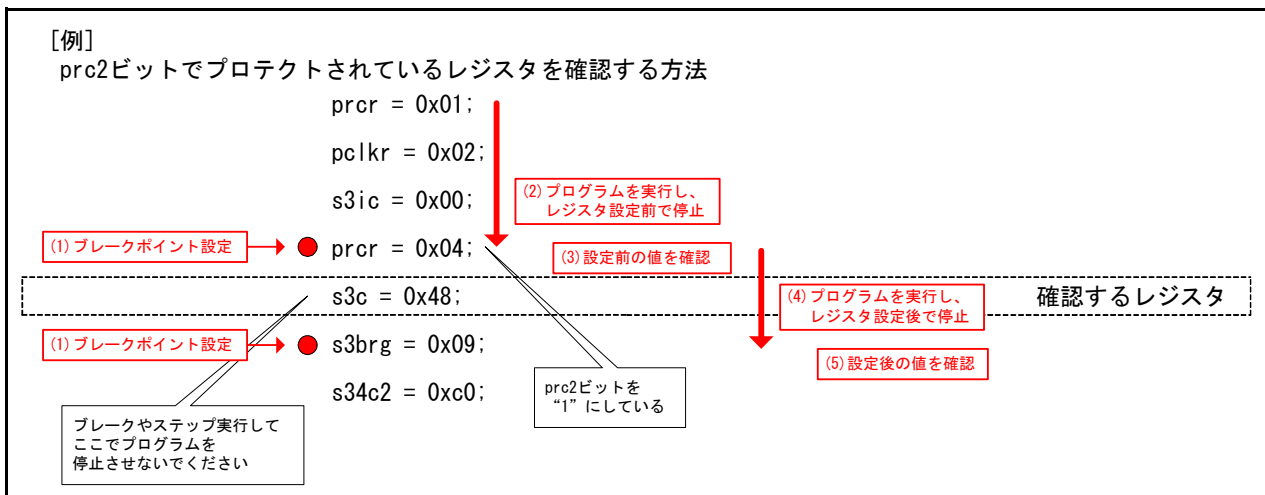


図 4.2 PRC2ビットでプロテクトされているレジスタの確認手順

4.1.2 オシロスコープで確認する方法

レジスタ設定後にそのレジスタを読み出して確認し、期待どおりの値が設定できていればテストポートを“H”出力するテスト用コードを追加します。テストポートの状態をオシロスコープで確認して期待どおりの値が設定できているか判断します。

システム上、出力にしても問題ないポートをテストポートとして使用してください。

◆確認手順

- (1) プログラムにテスト用テストポートを出力するコードを追加する。
図 4.3 にテスト用コードの追加例を示します。
- (2) プログラムを実行し、テストポートの状態をオシロスコープで確認する。

◆判断方法

確認手順(2)でテストポートの変化をオシロスコープで確認してください。

「図 4.3 テスト用コードの追加例」を実行した場合の、結果の判断方法は以下のとおりです。

<期待どおりに設定できている場合>

「図 4.4 期待どおり設定できている場合の波形」のように全てのテストポートが“H”出力されます。レジスタに値は設定できているため、問題の原因は他にあると考えられます。

<期待どおりに設定できていない場合>

「図 4.5 期待どおりに設定できていない場合の波形」のように(D)のテストポートが“H”出力されません。レジスタにプロテクトが設定されている場合や設定順序が定められている場合がありますので、ユーザズマニュアルハードウェア編を参照し、レジスタの設定条件および設定手順を確認してください。

<レジスタ設定後にプログラムが停止または暴走した場合>

「図 4.6 設定後プログラムが停止または暴走した場合の波形」のように(B)および(D)のテストポートが“H”出力されない可能性があります。ユーザズマニュアルハードウェア編を参照し、レジスタの設定手順、電気的特性の推奨動作条件および注意事項が守れているか確認してください。

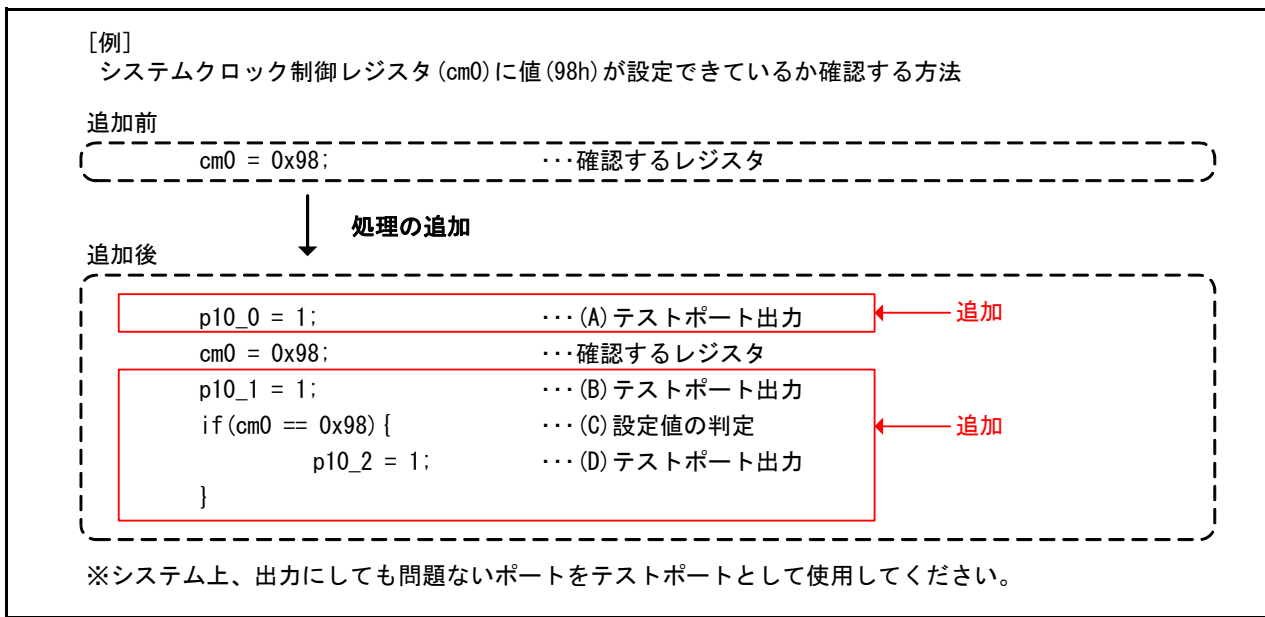


図 4.3 テスト用コードの追加例

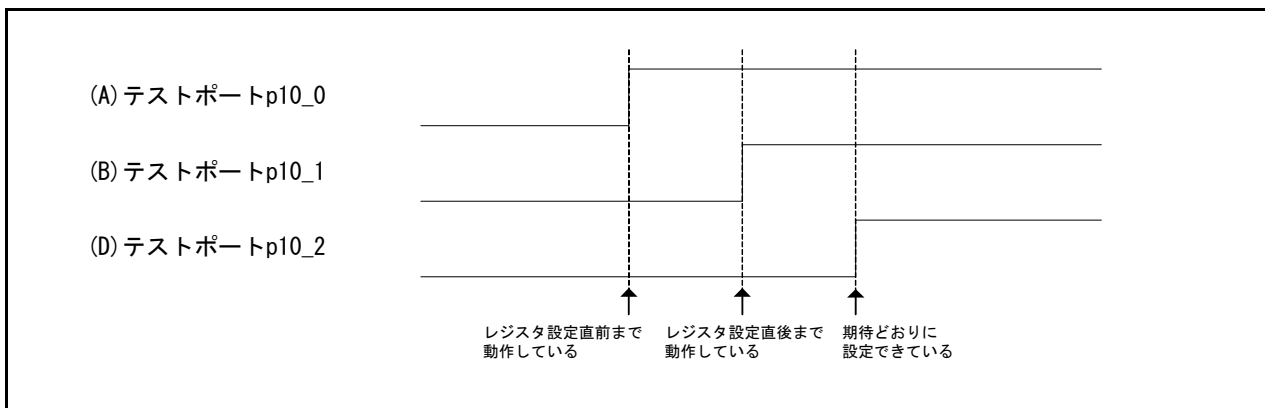


図 4.4 期待どおり設定できている場合の波形

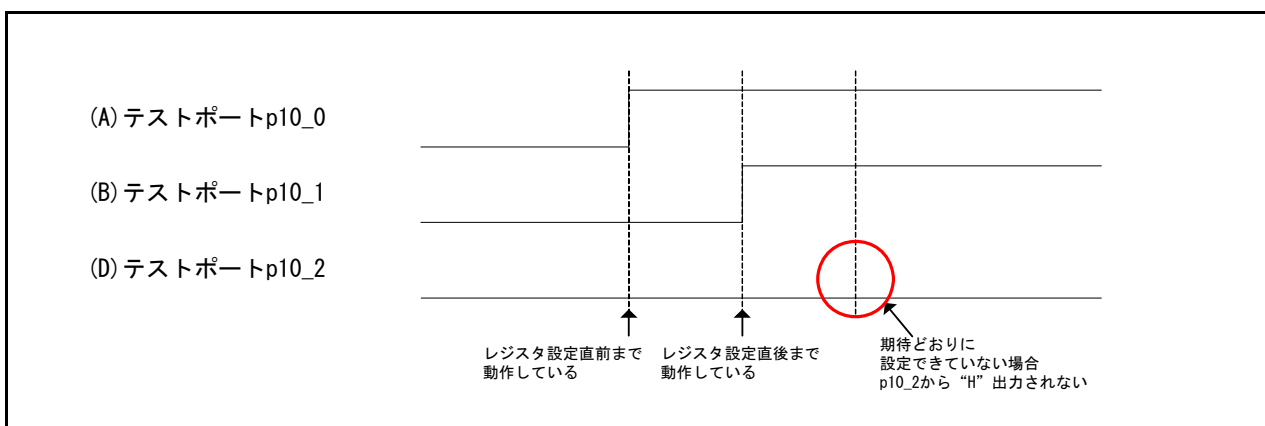


図 4.5 期待どおりに設定できていない場合の波形

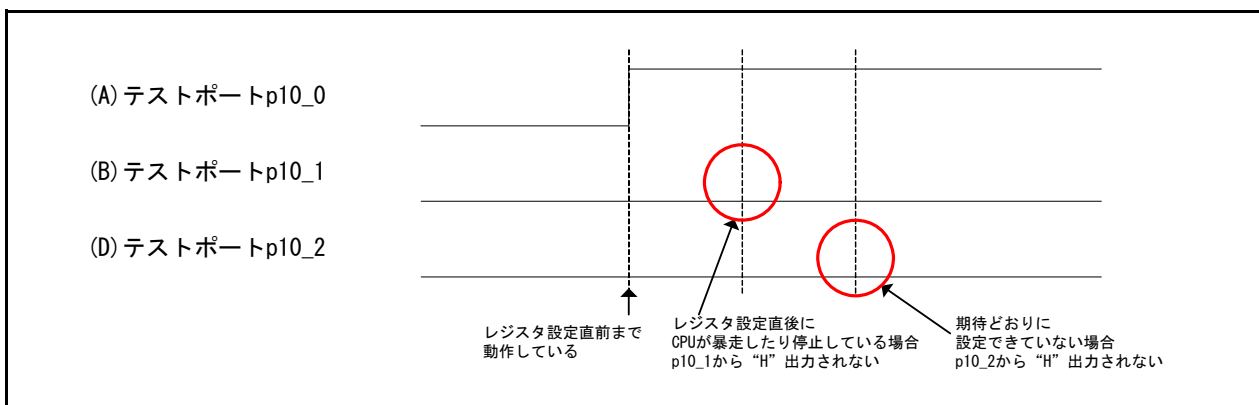


図 4.6 設定後プログラムが停止または暴走した場合の波形

4.2 動作中の周辺機能クロックf1の周波数を確認する

周辺機能クロックf1が期待どおりに設定できているか確認する方法を紹介します。

4.2.1 クロック出力機能でCLKOUTからf1を出力しオシロスコープで確認する方法

シングルチップモード時、CLKOUT端子からf1、f8、f32またはfCを出力できるクロック出力機能があります。f1は、CPUクロックの分周器を通過する前のクロックと同一になります。

クロック出力機能でCLKOUT端子からf1を出力するよう設定し、CLKOUT端子から出力される周波数をオシロスコープで確認します。

CLKOUT端子から出力するクロックは、25MHz以下にしてください。25MHzを超える場合は、f8を使用してください。

CLKOUT端子からの出力が、システム上、問題がないことを確認した上で実施してください。

◆確認手順

- (1) プログラムにCLKOUT端子からf1を出力(PCLKRレジスタのPCLK5ビットを“1”)にするコードを追加する。f1が25MHzを超える場合は、f1の代わりにf8を出力(CM0レジスタのCM01～CM00ビットを“10b”)にするコードを追加する。
- (2) プログラムを実行し、CLKOUT端子をオシロスコープで確認する。

◆判断方法

確認手順(2)でCLKOUT端子の周波数をオシロスコープで確認してください。

確認した周波数に、設定している分周を考慮してください。

その周波数が期待する周波数でない場合は、システムクロック関連のレジスタに期待どおりの値が設定できているか確認してください。方法は「4.1 レジスタが期待どおり設定できているか確認する」の節を参照してください。

4.2.2 タイマAのパルス出力をオシロスコープで確認する方法

タイマAのカウンタソースをf1、カウンタ値を“0”にしてパルス出力すると、TAiOUT端子からf1の1/2の周波数を出力できます。

f1は、CPUクロックの分周器を通過する前のクロックと同一になります。

タイマAを使用して、f1の1/2の周波数を出力するよう設定し、TAiOUT端子から出力される周波数をオシロスコープで確認します。

TAiOUT端子からの出力が、システム上、問題がないことを確認した上で実施してください。

◆確認手順

- (1) タイマAでf1の1/2の周波数を出力するコードを追加する。

[タイマA設定内容]

動作モード : タイマモード
パルス出力機能 : パルス出力あり
カウンタソース : f1
タイマレジスタ : 0000h
カウンタ動作 : 開始

- (2) プログラムを実行し、TAiOUT端子をオシロスコープで確認する。(iはタイマ出力の設定をしたチャンネル)

◆判断方法

確認手順(2)でTAiOUT端子の周波数をオシロスコープで確認してください。

確認した周波数を2倍し、設定している分周を考慮してください。

その周波数が期待する周波数でない場合は、システムクロック関連のレジスタに期待どおりの値が設定できているか確認してください。方法は「4.1 レジスタが期待どおり設定できているか確認する」の節を参照してください。

4.3 割り込み発生箇所を確認する

意図しないタイミングで割り込みが発生していないか確認する方法を紹介します。

4.3.1 ICE(E100)のトレース機能で確認する方法

ICEでは、プログラムを実行したときのトレースを確認することができます。

トレースから割り込みの発生箇所を確認します。

◆確認手順

- (1) 割り込み発生箇所を確認したい割り込み処理の先頭のコードにブレークを設定する。
- (2) プログラムを実行し、ブレークポイントで停止させる。
- (3) トレースウィンドウを開いてトレースを確認し、期待どおりのタイミングで割り込みが発生しているか確認する。

◆判断方法

確認手順(3)のときに意図しない箇所で割り込みが発生していないかをトレースウィンドウにて確認してください。意図しない箇所で割り込みが発生している場合は、次の2点を確認してください。

- 割り込みが発生する直前の数命令に問題がある可能性があります。ユーザーズマニュアルハードウェア編を参照し、レジスタの設定条件および設定手順を確認してください。
- 外部信号を受けたときに発生する割り込みを使用している場合は、意図しないタイミングで信号が入力されている可能性があります。オシロスコープで受信端子に不正な信号が入力されていないか確認してください。

4.3.2 デバッガで確認する方法

割り込み処理からの復帰(REIT)をステップ実行すると、割り込みが発生したコードの、次のコードで停止させることができます。これにより、割り込みが発生した箇所を判断します。

◆確認手順

- (1) 割り込み処理の復帰命令(REIT)にブレークポイントを設定する。
- (2) プログラムを実行し、ブレークポイントで停止させる。
- (3) ステップ実行し、REIT命令で復帰した番地の直前のコードを確認する。

◆判断方法

確認手順(3)のときに復帰した番地の直前のコードを確認してください。そのコードの実行直後に割り込みが発生しています。意図しない箇所で割り込みが発生している場合は、次の2点を確認してください。

- 割り込みが発生する直前の数命令に問題がある可能性があります。ユーザーズマニュアルハードウェア編を参照し、レジスタの設定条件および設定手順を確認してください。
- 外部信号を受けたときに発生する割り込みを使用している場合は、意図しないタイミングで信号が入力されている可能性があります。オシロスコープで受信端子に不正な信号が入力されていないか確認してください。

4.4 通信中に端子が“L”ホールドした時、どのデバイスが“L”にしているか確認する

通信が“L”ホールドによって中断された時に、どのデバイスが“L”を出力しているか確認する方法を紹介します。

4.4.1 M16Cマイコンをリセットする方法

M16Cマイコンは、リセットするとSFRの状態は初期値に戻り、端子の状態も入力ポートになります。リセット後のSDAi端子およびSCLi端子を確認し、M16Cマイコンが“L”ホールドしていたかを判断します。

◆確認手順

(1) リセットを入力し、SDAi端子、SCLi端子の状態をオシロスコープなどで確認する。

◆判断方法

確認手順(1)のときにSDAi端子およびSCLi端子が“H”になるか確認してください。結果の判断方法は、以下のとおりです。

<“H”になった場合>

M16Cマイコンによって、“L”ホールドしています。通信手順に誤りがないか確認してください。また、ノイズなどが発生し、ビットずれが発生していないか確認してください。

<“L”のままの場合>

外部デバイスによって、“L”ホールドされています。ノイズなどが発生し、ビットずれが発生していないか確認してください。

4.4.2 通信ラインに抵抗を接続する方法

M16Cマイコンと外部デバイス(EEPROMなど)間に、抵抗を直列に接続することで、オシロスコープの観測点から抵抗を介したデバイスからの“L”出力は、浮き上がることが確認できます。

通信中のSDAi端子およびSCLi端子を確認して、どのデバイスから“L”を出力しているかを判断します。

◆確認手順

(1) M16Cマイコンと外部デバイス間に抵抗を直列に接続する。

図4.7に通信ラインに抵抗を接続する場合の回路例と確認した波形の判断方法を示します。

(2) プログラムを実行し、SDAi端子、SCLi端子の状態をオシロスコープで確認する。

◆判断方法

確認手順(2)のときにSDAi端子、SCLi端子の変化を確認してください。

“L”レベルが浮き上がる場合は、観測点と抵抗を介したデバイスが“L”を出力しています。“L”レベルがGNDレベルの場合は、観測点とSDAi端子、SCLi端子との間に抵抗がないデバイスが“L”ホールドしています。

結果の判断方法は、「図4.7 通信ラインに抵抗を接続する場合の回路例と確認した波形の判断方法」に示します。

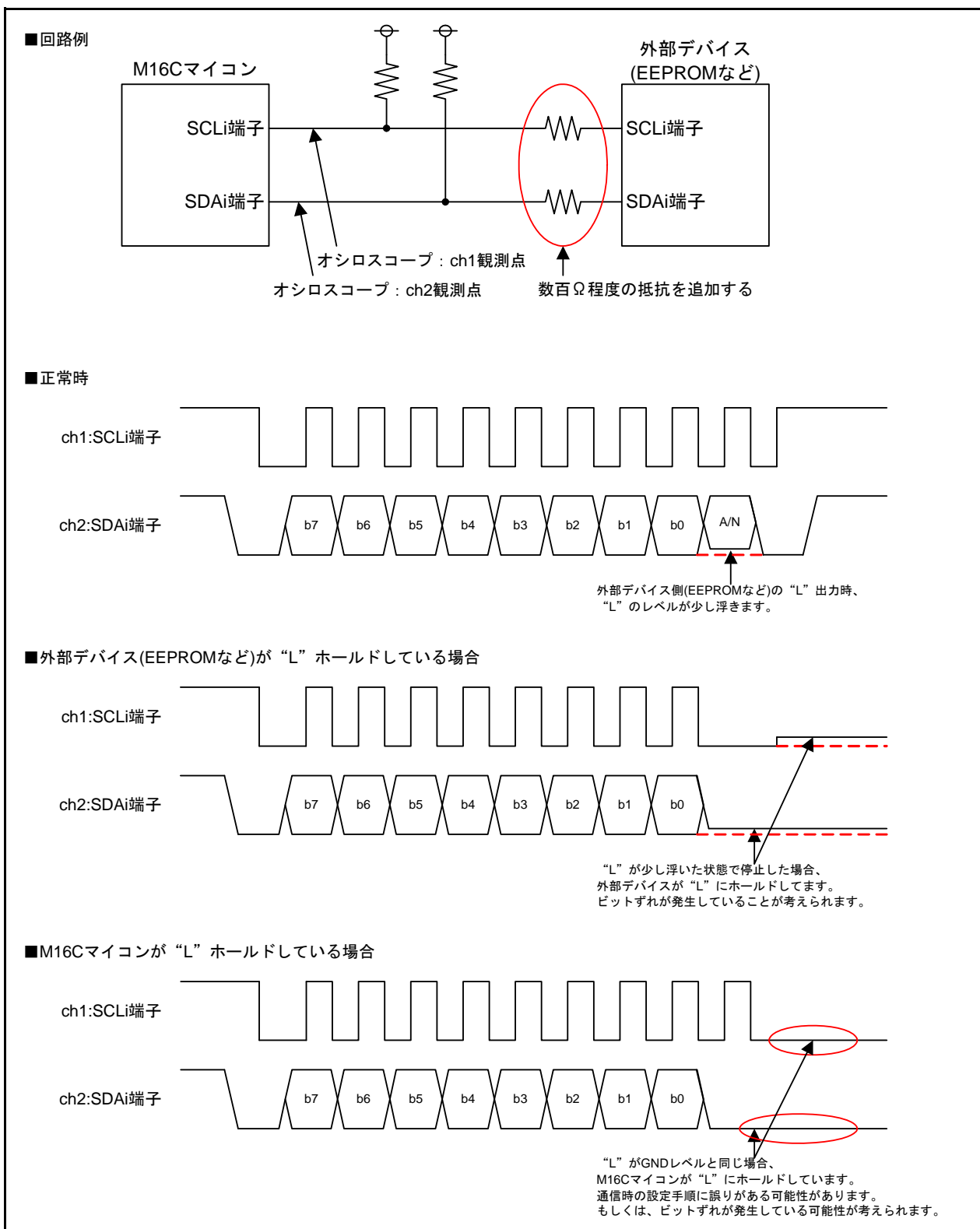


図 4.7 通信ラインに抵抗を接続する場合の回路例と確認した波形の判断方法

4.5 簡易 I2C モード通信時の信号に問題がないか確認する

出力しているクロックやデータの信号に問題がないか確認する方法を紹介します。
確認方法は、簡易 I2C モードの通信動作ごとに、次の順序で解説します。

- 簡易 I2C モードマスタ動作
- 簡易 I2C モードマスタ動作コンディション生成
- 簡易 I2C モードスレーブ動作

4.5.1 簡易 I2C モードマスタ動作時の信号をオシロスコープで確認する方法

マスタ動作時の波形をオシロスコープで捉えるトリガとするために、マスタ送信またはマスタ受信のデータを設定する直前に、テストポートを反転させるテスト用コードを追加します。

テストポートが反転するタイミングで、SDAi 端子から出力されるデータ、および SCLi 端子から出力されるクロックをオシロスコープで確認します。波形を確認することで発生している様々な問題を推測することができます。

テストポートは、システム上、出力にしても問題ないポートを使用してください。

◆ 確認手順

- (1) UiTB レジスタヘデータを設定する直前に、テストポートを反転させるコードを追加する。
- (2) プログラムを実行し、テストポートと SDAi 端子、SCLi 端子の状態をオシロスコープで確認する。
(2-1) 波形を取得するためのトリガは、テストポートの反転に設定し確認する。

◆ 判断方法

確認手順(2)でテストポートと SDAi 端子、SCLi 端子の変化をオシロスコープで確認してください。

以下に、簡易 I2C モードマスタ動作する場合の、正常時と異常時の信号パターンを紹介します。
異常時の信号パターンから、推測できる問題点を解説しています。

本項で紹介している信号パターンは、次の条件の場合の例です。

- UiSMR3 レジスタの CKPH ビットが“1”(クロック遅れあり)

• 正常時の信号パターン [A-1]

テストポートが反転した直後に、SDAi 端子から送信データ、SCLi 端子からクロックが出力されている。

この場合、SDAi 端子からの送信データが、UiTB レジスタに設定したとおりのデータであり、SCLi 端子からのクロックが、期待どおりの周期で出力されていれば、出力信号に問題はないため、原因は他にあると考えられます。

ただし、信号が相互の動作条件(規格など)を満たしているかは、確認してください。

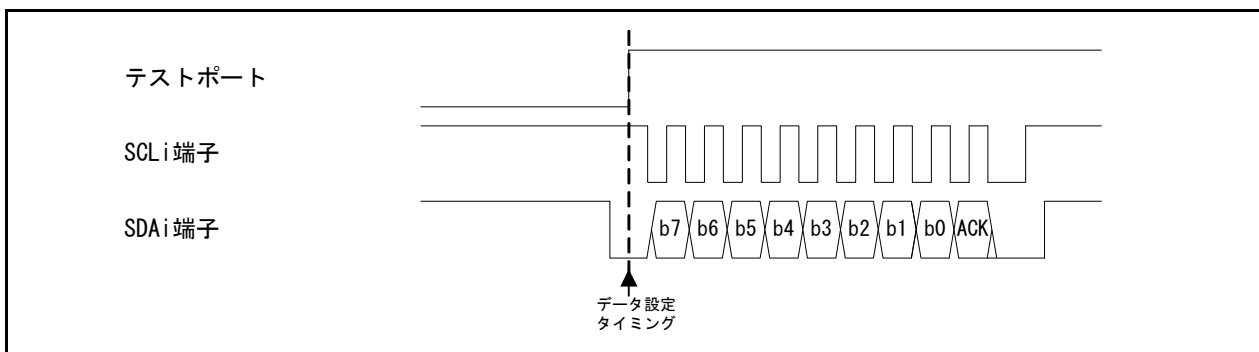


図 4.8 正常時の信号パターン [A-1]

• 異常時の信号パターン [A-2]

テストポートがスタートコンディション出力後に反転し、その後、SDAi 端子、SCLi 端子は“L”のまま変化しない。

この場合、送信条件が満たせていないことが考えられます。

送信許可(UiC1 レジスタの TE ビットを“1”)になっているか確認してください。

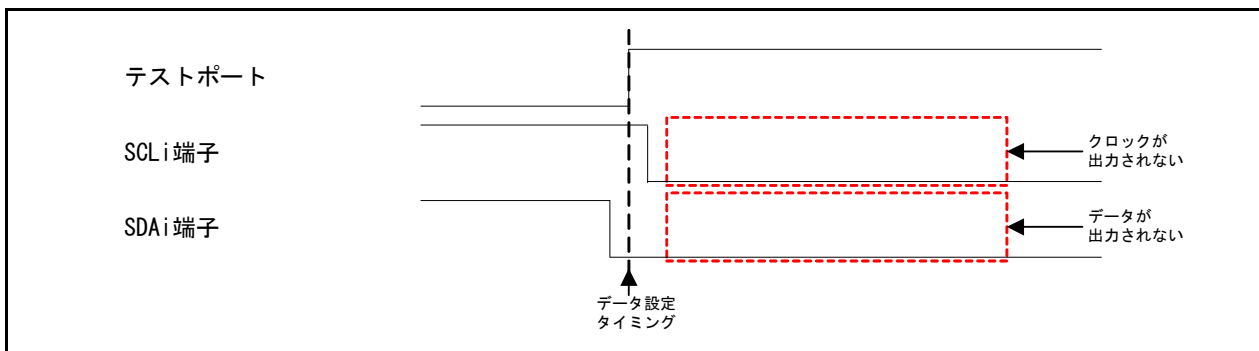


図 4.9 異常時の信号パターン [A-2]

• 異常時の信号パターン [A-3]

テストポートが反転した後も、SDAi端子、SCLi端子は“L”のまま変化しない。
この場合、SDAi端子およびSCLi端子がプルアップしていないことが考えられます。
外部回路で端子をプルアップしてください。

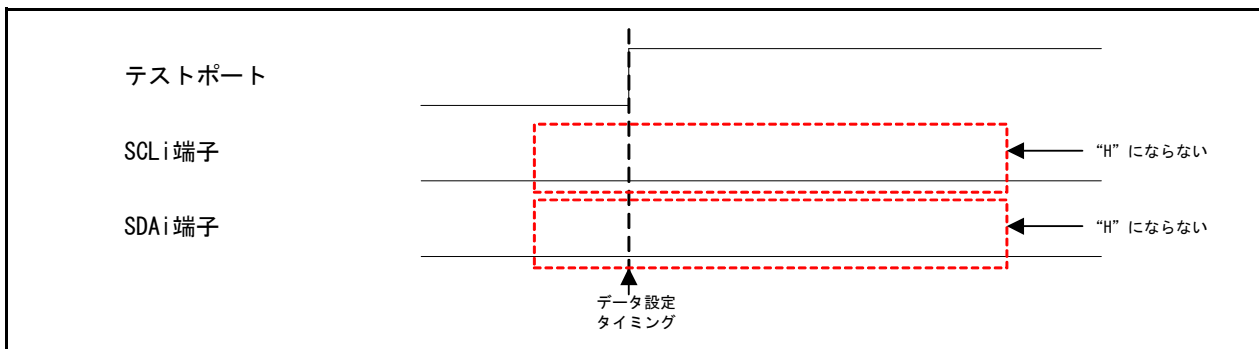


図 4.10 異常時の信号パターン [A-3]

• 異常時の信号パターン [A-4]

SDAi端子およびSCLi端子に、短い“L”パルスが発生している。
この場合、周辺回路や外部の影響でノイズが発生した可能性があります。SCLi端子にパルスが発生すると、通信先デバイスでビットずれが発生する可能性があります。
ノイズが発生している場合は、SCLi端子、SDAi端子に直列で数百Ω程度の直列抵抗を挿入するか、信号線をノイズ発生源から遠ざけるなどして、対策してください。
直列抵抗を使用する場合、I²Cバス上の定数が許容可能な範囲になるように設計してください。

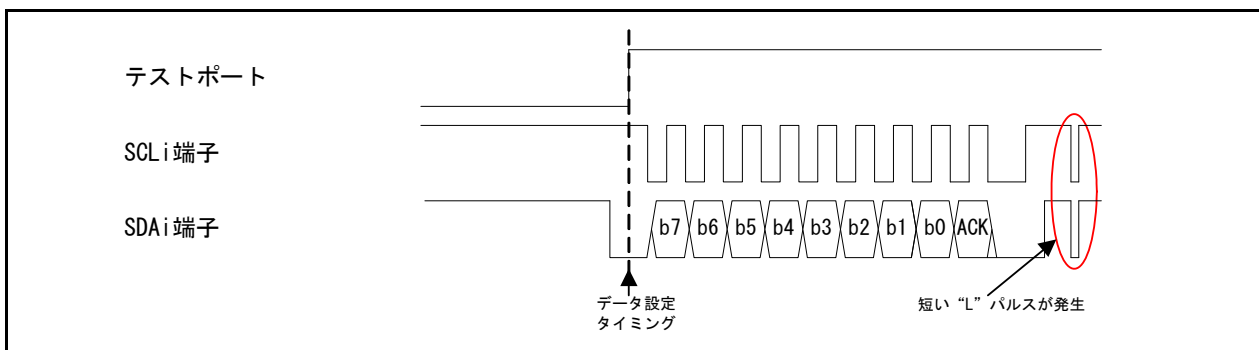


図 4.11 異常時の信号パターン [A-4]

• 異常時の信号パターン [A-5]

SDAi 端子、SCLi 端子から出力される信号の“L”のレベルが“0V”まで下がらない。もしくは、“H”のレベルが“VCC”まで上がらない。

この場合、SDAi 端子、SCLi 端子と接続している他の端子や通信先デバイスから信号が出力され、信号の衝突が発生している可能性があります。通信先デバイスの VIH、VIL を満たせず、“H”および“L”を正しく認識できない可能性があります。また、100Ω 程度の抵抗でプルアップ/プルダウンし、レベルが変化しない場合は、CMOS 出力されていると考えられます。

SDAi 端子、SCLi 端子と接続している端子や、通信先デバイスから信号が出力されていないか、回路や設定などを確認してください。

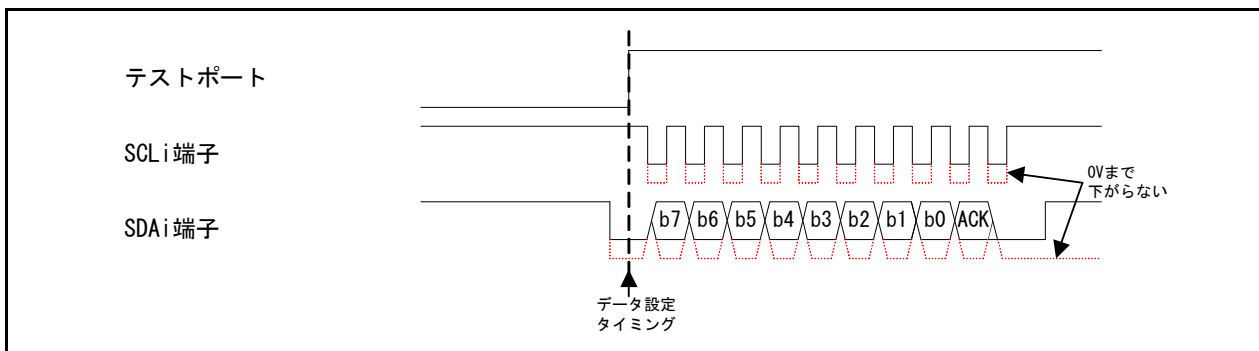


図 4.12 異常時の信号パターン [A-5]

• 異常時の信号パターン [A-6]

SDAi 端子、SCLi 端子から出力される信号の“H”のレベルが“VCC”まで上がらない。

この場合、SDAi 端子、SCLi 端子に接続しているプルアップ抵抗値や配線容量が大きすぎることによって、レベルが通信先デバイスの“VIH”まで上がらなかったり、十分な“H”期間が確保できず規格を満たせていない可能性があります。また、ビットずれが発生している可能性があります。

SDAi 端子、SCLi 端子に接続しているプルアップ抵抗値や配線容量が大きくなりすぎないように十分に余裕をもった回路設計を行ってください。

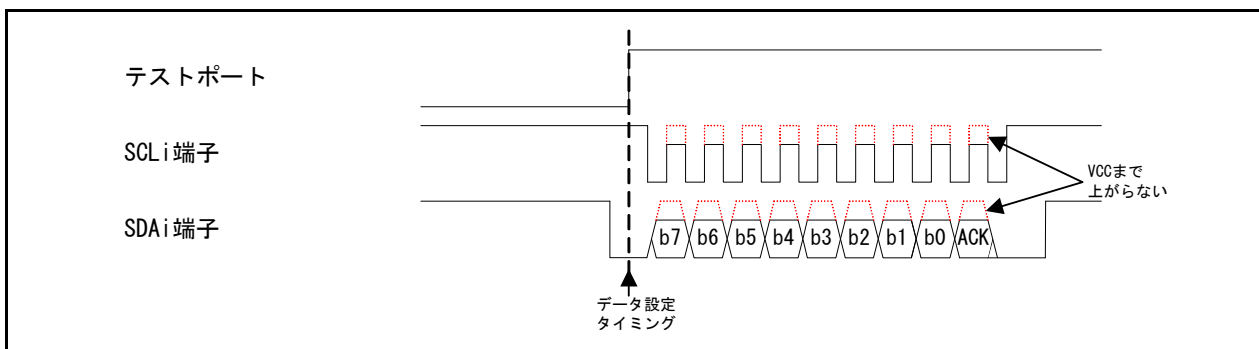


図 4.13 異常時の信号パターン [A-6]

• 異常時の信号パターン [A-7]

1バイトの送信データ出力中に、複数回テストポートが反転する。

この場合、送信バッファフル(UiC1レジスタのTIビットが“0”)の状態、UiTBレジスタに次のデータを設定しているため、データが上書きされ、送信データに抜けが発生することが考えられます。次の送信データは、送信完了割り込み内で設定するようにしてください。

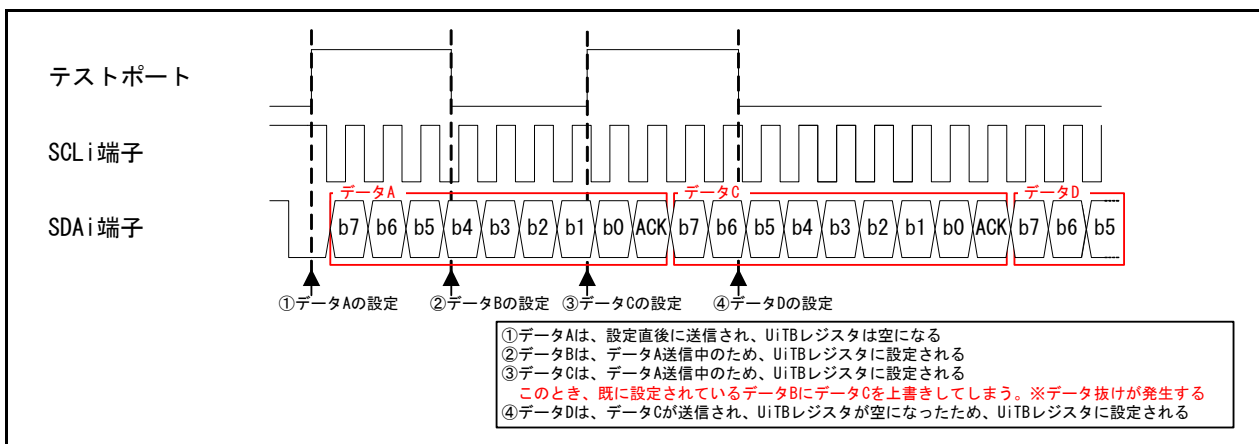


図 4.14 異常時の信号パターン [A-7]

• 異常時の信号パターン [A-8]

スレーブアドレス送信時にACK/NACKビットでNACKを受信する。

この場合、送信したスレーブアドレスが、送信先デバイスのアドレスと一致しなかった、または、通信先デバイスが書き込み処理中であり、NACK応答している可能性が考えられます。

通信先デバイスの仕様書などを確認し、正しいアドレスを送信するようにしてください。または、通信先デバイスの書き込み時間待った後、再度スレーブアドレスを送信してください。

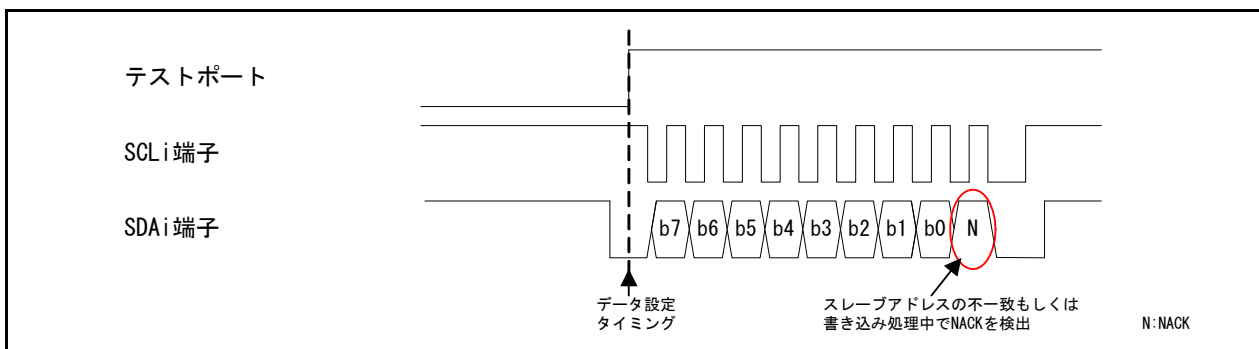


図 4.15 異常時の信号パターン [A-8]

4.5.2 マスタ動作コンディション生成時の信号に問題がないか確認する方法

簡易I²Cモードマスタ動作コンディション生成時の波形をオシロスコープで捉えられるように、スタートコンディション発行直前に、テストポートを反転させるテスト用コードを追加します。

テストポートが反転するタイミングで、SDAi端子から出力されるデータ、および、SCLi端子から出力されるクロックをオシロスコープで確認します。波形を確認することで発生している様々な問題を推測することができます。

テストポートは、システム上、出力にしても問題ないポートを使用してください。

◆確認手順

- (1) UiSMR4レジスタのSTSPSELビットへ“1”を設定する直前に、テストポートを反転させるコードを追加する。
- (2) プログラムを実行し、テストポートとSDAi端子、SCLi端子の状態をオシロスコープで確認する。
(2-1)波形を取得するためのトリガは、テストポートの反転に設定し確認する。

◆判断方法

確認手順(2)でテストポートとSDAi端子、SCLi端子の変化をオシロスコープで確認してください。

以下に、簡易I²Cモードマスタ動作する場合の、正常時と異常時の信号パターンを紹介します。異常時の信号パターンから、推測できる問題点を解説しています。本項で紹介している信号パターンは、次の条件の場合の例です。

- ・ UiSMR3レジスタのCKPHビットが“1”(クロック遅れあり)

• 正常時の信号パターン [B-1]

テストポートが反転した直後に、SDAi 端子と SCLi 端子からスタートコンディションが出力されている。また、テストポートが反転する直前まで、SDAi 端子と SCLi 端子が“H”レベルである。
この場合、出力信号に問題はないため、原因は他にあると考えられます。
ただし、信号が相互の動作条件(規格など)を満たしているかは、確認してください。

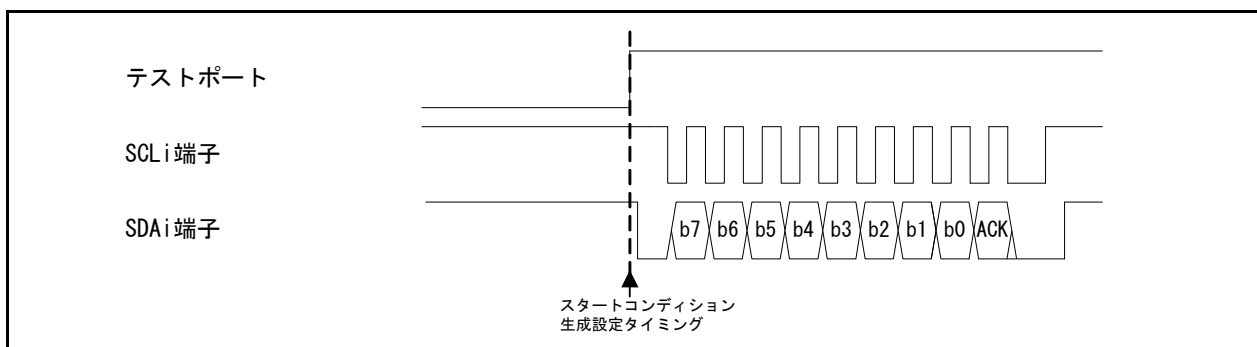


図 4.16 正常時の信号パターン [B-1]

• 異常時の信号パターン [B-2]

テストポートが反転する前に、SDAi 端子、SCLi 端子が“L”レベルになる区間がある。
この場合、スタートコンディションを生成するためのSFRの設定手順に誤りがあることが考えられます。

手順に関しては、以下のアプリケーションノートをご参照ください。

- UARTi 特殊モード1を使用した I²C バスインタフェース (RJJ05B1545)
- UARTi 特殊モード1を使用した I²C-bus インタフェース (マスタ送信/受信)(RJJ05B1596)
- UARTi 特殊モード1を使用した I²C-bus インタフェース (スレーブ送信/受信)(RJJ05B1604)

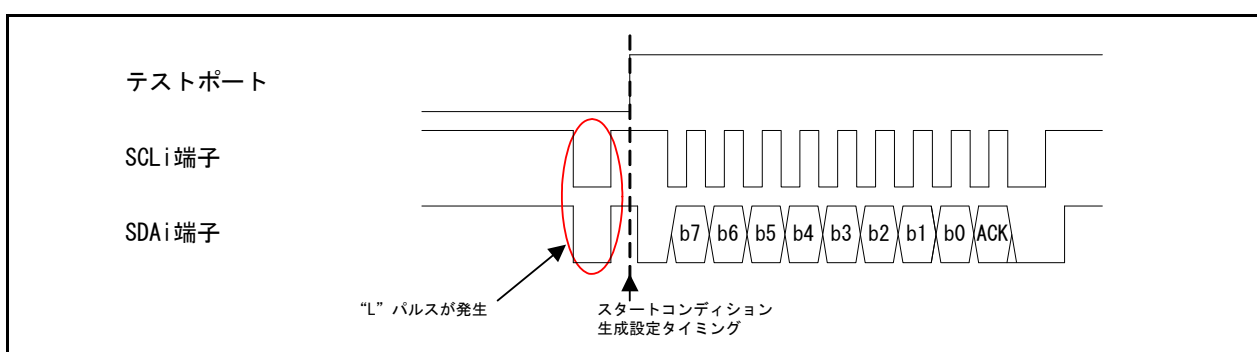


図 4.17 異常時の信号パターン [B-2]

• 異常時の信号パターン [B-3]

テストポートが反転した直後のスタートコンディションの出力で、SCLi 端子が SDAi 端子より先に“L”レベルになっている。

この場合、SDAi デジタル遅延機能による遅延量(UiSMR3 レジスタの DL2～DL0 ビットの設定値)が大きすぎると考えられます。

SDAi デジタル遅延機能による遅延量(UiSMR3 レジスタの DL2～DL0 ビットの設定値)が適切に設定されているかご確認ください。

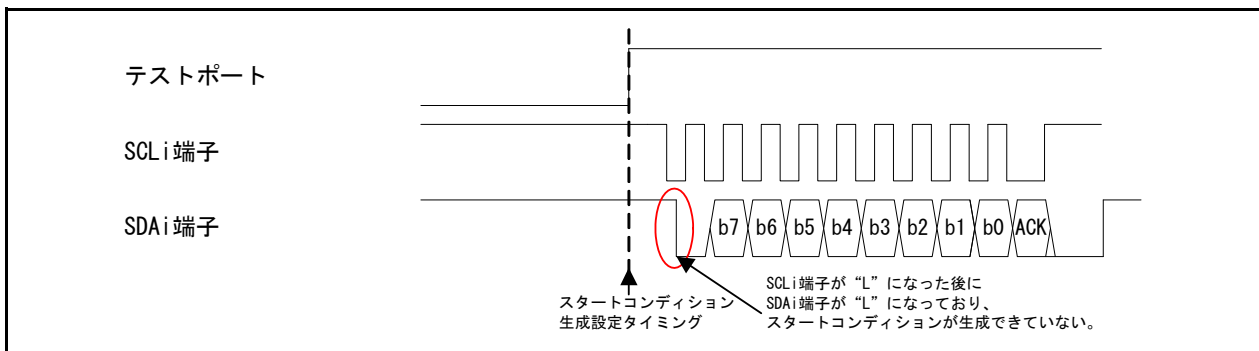


図 4.18 異常時の信号パターン [B-3]

4.5.3 スレーブ動作時の信号に問題がないか確認する方法

スレーブ動作時の波形をオシロスコープで捉えるトリガとするために、スレーブ受信またはスレーブ送信のデータを読み書きする直前に、テストポートを反転させるテスト用コードを追加します。

テストポートが反転するタイミングで、SDAi端子から出力されるデータおよびSCLi端子から出力されるクロックをオシロスコープで確認します。波形を確認することで発生している様々な問題を推測することができます。

テストポートは、システム上、出力にしても問題ないポートを使用してください。

◆確認手順

- (1) UiTBレジスタへデータを設定、または、UiRBレジスタからデータを読み出す直前に、テストポートを反転させるコードを追加する。
- (2) プログラムを実行し、テストポートとSDAi端子、SCLi端子の状態をオシロスコープで確認する。
(2-1)波形を取得するためのトリガは、テストポートの反転に設定し確認する。

◆判断方法

確認手順(2)でテストポートとSDAi端子、SCLi端子の変化をオシロスコープで確認してください。

以下に、簡易I²Cモードスレーブ動作する場合の、正常時と異常時の信号パターンを紹介いたします。異常時の信号パターンから、推測できる問題点を解説しています。

本項で紹介している信号パターンは、次の条件の場合の例です。

- ・ UiSMR3レジスタのCKPHビットが“1”(クロック遅れあり)

• 正常時の信号パターン [C-1]

テストポートがスレーブアドレス受信時の8ビット目のSCLiの立ち下がりで反転し、その後、データ送受信時の9ビット目のSCLiの立ち下がりで反転している。

この場合、SDAi端子への受信データがUiRBレジスタから読み出すことができ、SDAi端子からの送信データがUiTBレジスタに設定したとおりのデータであれば、入出力信号に問題はないため、原因は他にあると考えられます。

ただし、信号が相互の動作条件(規格など)を満たしているかは確認してください。

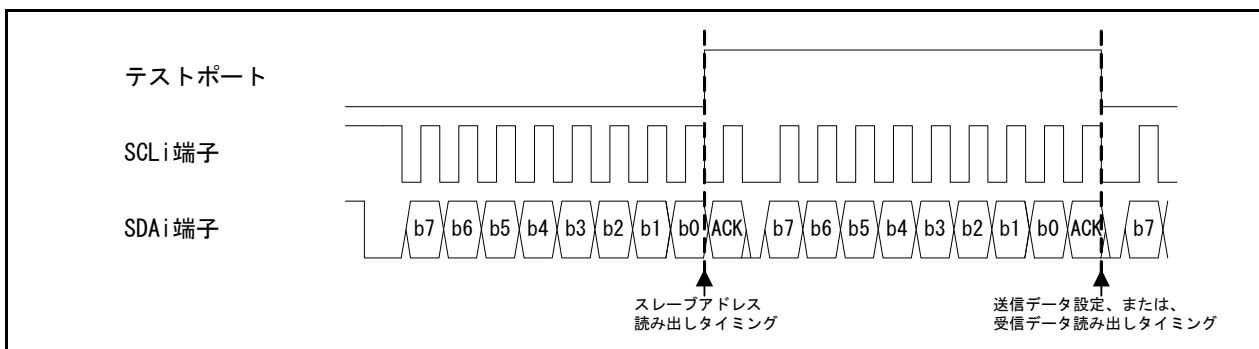


図 4.19 正常時の信号パターン [C-1]

• 異常時の信号パターン [C-2]

テストポートがスレーブアドレス受信時の8ビット目のSCLiの立ち下がりで反転し、その後、データ送受信時の8ビット目のSCLiの立ち下がりと9ビット目のSCLiの立ち下がりで反転している。

この場合、受信割り込みで受信データを読み出していると考えられます。受信割り込みで読み出した時と送信割り込みで読み出した時では、UiRBレジスタに格納されるデータの並びが異なります。

スレーブアドレス受信以降のデータの読み出しは、送信割り込みで行っているか確認してください。受信割り込みで読み出す場合は、データの並び替えを必要です。

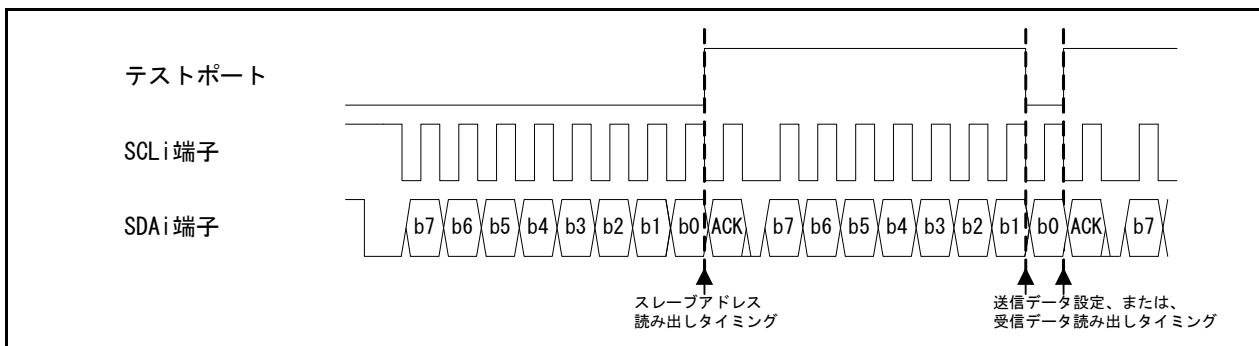


図 4.20 異常時の信号パターン [C-2]

4.6 Nチャネルオープンドレイン出力端子がプルアップできているか確認する

正しく端子をプルアップできているか確認する方法を紹介します。

4.6.1 オシロスコープで確認する方法

オシロスコープで、Nチャネルオープンドレイン出力端子が正しくプルアップできているか確認できます。

◆確認手順

- (1) プルアップできているか確認したい端子を、プルアップに使用している抵抗と同じ抵抗値の抵抗を介してVSSに接続(プルダウン)する。
- (2) プログラムを実行し、端子の状態をオシロスコープで確認する。

◆判断方法

確認手順(2)で端子の変化をオシロスコープで確認してください。
確認した結果の判断方法は以下のとおりです。

<Nチャネルオープンドレイン出力端子をプルアップできている場合>

出力していない時(リセット中)、端子のレベルは、 $1/2V_{CC}$ になります。

“H”出力時のレベルは、 $1/2V_{CC}$ 、“L”出力時のレベルは、VSSとなります。

この場合、Nチャネルオープンドレイン出力端子を正しくプルアップできていると判断できます。

外部のデバイスと相互に通信を行う場合、Nチャネルオープンドレイン出力端子から出力される波形の“H”のレベルは、 $1/2V_{CC}$ となっているため、通信は、正しくできない可能性があります。

外部のデバイスとの通信の確認を行う場合は、Nチャネルオープンドレイン出力端子に接続したプルダウン抵抗をはずしてください。

<Nチャネルオープンドレイン出力端子をプルアップできていない場合>

出力していない時(リセット中)、端子のレベルは、VSSになります。

“H”出力時のレベルは、VSS、“L”出力時のレベルは、VSSとなります。

プルアップしている端子が正しいか回路を見直してください。

また、外部のデバイスから“L”が入力されている可能性があります。

外部のデバイスから“L”が入力されている場合は、外部デバイスをリセットしておくなどして“L”出力していない状態で再度確認してください。

<Nチャネルオープンドレイン出力端子ではない場合>

出力していない時(リセット中)、端子のレベルは、VSSになります。

“H”出力時のレベルは、VCC、“L”出力時のレベルは、VSSとなります。

CMOS出力端子であるか、Nチャネルオープンドレイン出力端子への切り替えに失敗していることが考えられます。

確認している端子が正しいか確認してください。また、Nチャネルオープンドレイン出力の設定が正しく行えているかプログラムを確認してください。

4.7 割り込みが抜けたり発生しなくなる原因を確認する

プログラムに割り込みが抜けたり、発生しなくなる原因がないか確認する方法を紹介します。

4.7.1 “00000h”番地を読み出していないかをICE(E100)で確認する方法

割り込みが抜けたり、発生しなくなる原因には、次の2点が考えられます。

- “00000h”番地を読み出した場合
- 割り込み要求が発生した後、割り込み処理へ遷移する前に、割り込み要求をクリアしてしまった場合

本項では、“00000h”番地を読み出していないか確認する方法を紹介しています。

◆確認手順

- (1) 次の手順で、“00000h”番地を読み出したときにブレークが発生するようHEWを設定する。
 - (1-1)メニューバーから [表示] → [イベント] → [ハードウェアブレーク条件設定] を選択する。
 - (1-2)開いた「ハードウェアブレーク条件設定」ウィンドウで、「追加」ボタンを押す。
 - (1-3)開いた「イベント設定」ウィンドウで、次のとおりに設定し、「OK」ボタンを押す。
 - ・ イベントタイプ：データアクセス
 - ・ アクセスタイプ：MCUバス CPUとDMAC
 - ・ アドレス条件： 指定値 (=) 開始：0000h
 - ・ リード/ライト：リード/ライト
 - (1-4)「ハードウェアブレーク条件設定」ウィンドウの「適応」ボタンを押し、設定を反映させる。
- (2) プログラムを実行し、ブレークが発生するか確認する。

◆判断方法

- 確認手順(2)でブレークが発生するか確認してください。
ブレークが発生した場合は、“00000h”番地を読み出しています。
トレースウィンドウを開き、“00000h”番地を読み出したプログラムの場所を確認してください。

5. 該当する現象がない／該当する現象がわからない

5.1 解決しない場合は

解決しない場合は、サポート窓口までお問い合わせください。
サポート窓口では以下の情報を元に原因を推測します。
お問い合わせ時に以下の情報をご連絡ください。

- (1) マイコン型名
- (2) 実現したいこと
- (3) 発生している現象
- (4) 動作周波数(CPUクロック)
- (5) 接続している発振子の周波数
- (6) 電源電圧
- (7) 温度
- (8) 再現性
- (9) 依存性(電圧依存、周波数依存、基板依存)
- (10) 現象の発生回数(pcs/pcs)
- (11) 現象の発生頻度(times/hour)
- (12) 現象が発生している周辺機能名
 - ・通信モード
 - ・転送レート
- (13) 開発フェーズ(開発中、量産中)
- (14) 関連レジスタの設定値
- (15) シミュレータ、エミュレータの使用状況
- (16) コンパイラのバージョン

6. 参考ドキュメント

M16C/63グループ ユーザーズマニュアル ハードウェア編 Rev.2.20
M16C/64Aグループ ユーザーズマニュアル ハードウェア編 Rev.2.10
M16C/64Cグループ ユーザーズマニュアル ハードウェア編 Rev.1.10
M16C/65グループ ユーザーズマニュアル ハードウェア編 Rev.2.10
M16C/65Cグループ ユーザーズマニュアル ハードウェア編 Rev.1.10
M16C/6Cグループ ユーザーズマニュアル ハードウェア編 Rev.2.10
M16C/5LD、M16C/56Dグループ ユーザーズマニュアル ハードウェア編 Rev.1.20
M16C/5L、M16C/56グループ ユーザーズマニュアル ハードウェア編 Rev.1.10
M16C/5M、M16C/57グループ ユーザーズマニュアル ハードウェア編 Rev.1.10
(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	M16C/63,64A,64C,65,65C,6C,5LD,56D,5L,56,5M,57 開発時のトラブルの解決方法 シリアルインタフェースI2Cモード 編
------	---

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2014.04.01	-	初版発行
1.10	2014.10.01	-	解析手法の章を追加
		35	2.4.6項を追加
		35~39	2.7節、2.7.1項、2.7.2項を追加

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. 未使用端子の処理

【注意】未使用端子は、本文の「未使用端子の処理」に従って処理してください。

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

2. 電源投入時の処置

【注意】電源投入時は、製品の状態は不定です。

電源投入時には、LSIの内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。

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

3. リザーブアドレスのアクセス禁止

【注意】リザーブアドレスのアクセスを禁止します。

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

4. クロックについて

【注意】リセット時は、クロックが安定した後、リセットを解除してください。

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

5. 製品間の相違について

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

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

ご注意書き

1. 本資料に記載された回路、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、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。
2. 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。
3. 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に関し、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. 当社製品を改造、改変、複製等しないでください。かかる改造、改変、複製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）、交通用信号機器、
防災・防犯装置、各種安全装置等
当社製品は、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（原子力制御システム、軍事機器等）に使用されることを意図しておらず、使用することはできません。たとえ、意図し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損害が生じても、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ください。
6. 当社製品をご使用の際は、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他の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。
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い合わせください。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。
9. 本資料に記載されている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。また、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用用途に使用しないでください。当社製品または技術を輸出する場合は、「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。
10. お客様の転売等により、本ご注意書き記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は何らの責任も負わず、お客様にてご負担して頂きますのでご了承ください。
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。

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 エレクトロニクス株式会社 〒100-0004 千代田区大手町2-6-2（日本ビル）

■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/contact/>