お客様各位

カタログ等資料中の旧社名の扱いについて

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

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

2010年4月1日
ルネサスエレクトロニクス株式会社

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

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

ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したものですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。

   標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット

   高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）

   特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

注 1. 本資料において使用されている「当社」とは、ルネサスエレクトロニクス株式会社およびルネサスエレクトロニクス株式会社がその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

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

サンプル・プログラム関数使用法説明書【シリアル通信（CSIB）編】（V850E/IA3, V850E/IA4, V850ES/IK1, V850ES/IE2）

この説明書では，V850E/IA4 用のクロック同期式シリアル・インタフェース B（CSIB）のサンプル・プログラム関数について説明します。

なお，この説明書では V850E/IA4 を使用した場合で記述しています。V850E/IA3, V850ES/IK1, V850ES/IE2 を使用した場合は，この説明書を参考にしてください。

ご注意

このサンプル・プログラムはあくまで参考用のものであり，当社がこの動作を保証するものではありません。このサンプル・プログラムを使用する場合，お客様のセット上で十分な評価をしたうえでご使用いただきますようお願いいたします。

資料番号　U18234JJ1V0AN00（第 1 版）
発行年月　July 2006 NS　CP(K)

© NEC Electronics Corporation 2006

CMOSデバイスの一般的注意事項

① 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デバイス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。入力レベルが固定な場合はもちろん，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等が入らないようご使用ください。

② 未使用入力の処理

CMOSデバイスの未使用端子の入力レベルは固定してください。

未使用端子入力については，CMOSデバイスの入力に何も接続しない状態で動作させるのではなく，プルアップかプルダウンによって入力レベルを固定してください。また，未使用の入出力端子が出力となる可能性（タイミングは規定しません）を考慮すると，個別に抵抗を介してV_{DD}またはGNDに接続することが有効です。

資料中に「未使用端子の処理」について記載のある製品については，その内容を守ってください。

③ 静電気対策

MOSデバイス取り扱いの際は静電気防止を心がけてください。

MOSデバイス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，または導電性の緩衝材，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，MOSデバイス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。

④ 初期化以前の状態

電源投入時，MOSデバイスの初期状態は不定です。

電源投入時の端子の出力状態や入出力設定，レジスタ内容などは保証しておりません。ただし，リセット動作やモード設定で定義している項目については，これらの動作ののちに保証の対象となります。

リセット機能を持つデバイスの電源投入後は，まずリセット動作を実行してください。

⑤ 電源投入切断順序

内部動作および外部インタフェースで異なる電源を使用するデバイスの場合，原則として内部電源を投入した後に外部電源を投入してください。切断の際には，原則として外部電源を切断した後に内部電源を切断してください。逆の電源投入切断順により，内部素子に過電圧が印加され，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「電源投入切断シーケンス」についての記載のある製品については，その内容を守ってください。

⑥ 電源OFF時における入力信号

当該デバイスの電源がOFF状態の時に，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OFF時における入力信号」についての記載のある製品については，その内容を守ってください。

本製品のうち，外国為替及び外国貿易法の規定により規制貨物等（または役務）に該当するものについては，日本国外に輸出する際に，同法に基づき日本国政府の輸出許可が必要です。

非該当品　　：μPD70F3184, 70F3186, 70F3329, 70F3713, 70F3714

ユーザ判定品：μPD703183, 703185, 703186, 703327, 703329

- **本資料に記載されている内容は2006年7月現在のもので，今後，予告なく変更することがあります。量産設計の際には最新の個別データ・シート等をご参照ください。**
- **文書による当社の事前の承諾なしに本資料の転載複製を禁じます。当社は，本資料の誤りに関し，一切その責を負いません。**
- **当社は，本資料に記載された当社製品の使用に関連し発生した第三者の特許権，著作権その他の知的財産権の侵害等に関し，一切その責を負いません。当社は，本資料に基づき当社または第三者の特許権，著作権その他の知的財産権を何ら許諾するものではありません。**
- **本資料に記載された回路，ソフトウエアおよびこれらに関する情報は，半導体製品の動作例，応用例を説明するものです。お客様の機器の設計において，回路，ソフトウエアおよびこれらに関する情報を使用する場合には，お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し，当社は，一切その責を負いません。**
- **当社は，当社製品の品質，信頼性の向上に努めておりますが，当社製品の不具合が完全に発生しないことを保証するものではありません。当社製品の不具合により生じた生命，身体および財産に対する損害の危険を最小限度にするために，冗長設計，延焼対策設計，誤動作防止設計等安全設計を行ってください。**
- **当社は，当社製品の品質水準を「標準水準」，「特別水準」およびお客様に品質保証プログラムを指定していただく「特定水準」に分類しております。また，各品質水準は，以下に示す用途に製品が使われることを意図しておりますので，当社製品の品質水準をご確認ください。**

  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

  特別水準：輸送機器（自動車，電車，船舶等），交通用信号機器，防災・防犯装置，各種安全装置，生命維持を目的として設計されていない医療機器

  特定水準：航空機器，航空宇宙機器，海底中継機器，原子力制御システム，生命維持のための医療機器，生命維持のための装置またはシステム等

当社製品のデータ・シート，データ・ブック等の資料で特に品質水準の表示がない場合は，標準水準製品であることを表します。意図されていない用途で当社製品の使用をお客様が希望する場合には，事前に当社販売窓口までお問い合わせください。

（注）

（1）本事項において使用されている「当社」とは，NECエレクトロニクス株式会社およびNECエレクトロニクス株式会社がその総株主の議決権の過半数を直接または間接に保有する会社をいう。

（2）本事項において使用されている「当社製品」とは，（1）において定義された当社の開発，製造製品をいう。

M8E　02.11

はじめに

注　意 1.　この説明書で使用するプログラムは，NEC エレクトロニクスのホームページ（http://www.necel.co.jp/）よりダウンロードしてください。

2.　このサンプル・プログラムを使用する場合は，次のスタートアップ・ファイル，リンク・ディレクティブ・ファイルを参照し，必要に応じて修正してください。

・スタートアップ・ファイル　　：IA4_start.s

・リンク・ディレクティブ・ファイル　　：IA4_link.dir

凡　例　関数一覧表は次のように構成されています。

テーマ（ハードウエア略号）

【機　　能】　テーマの説明
【関　数　名】　サンプル関数の名前
【引　き　数】　引き数の型と概要
【処　理　内　容】　サンプル関数の処理内容
【使　用 S　F　R】　レジスタ名と設定内容
【call　　関数】　呼び出し関数の名前と機能
【変　　数】　サンプル関数での使用変数の型，名前，概要
【割　り　込　み】　関数名
【割り込み 要 因】　名称
【フ ァ イ ル 名】　対応するサンプル・プログラム・ファイル名
【注　意　事　項】　関数使用上の注意。使い方

割り込み関数

【関　数　名】　割り込み関数の名前
【処　理　内　容】　割り込み関数の処理内容
【使　用 S　F　R】　レジスタ名と設定内容
【call　　関数】　なし
【変　　数】　変数名，機能
【フ ァ イ ル 名】　対応するサンプル・プログラム・ファイル名
【注　意　事　項】　なし

製品間の違い　V850E/IA4 と V850E/IA3, V850ES/IK1, V850ES/IE2 のクロック同期式シリアル・インタフェース B（CSIB）についての製品間の違いを次に示します。

項　　目	V850E/IA4	V850E/IA3	V850ES/IK1	V850ES/IE2
チャネル	2ch （1chはUARTAと兼用）	2ch （1chはUARTAと兼用）	1ch	
通信クロック （マスタ・モード時）	f_{XX}/4, f_{XX}/8, f_{XX}/16, f_{XX}/32, f_{XX}/64, f_{XX}/128, f_{XX}/256	f_{XX}/4, f_{XX}/8, f_{XX}/16, f_{XX}/32, f_{XX}/64, f_{XX}/128, f_{XX}/256	f_{XX}/2, f_{XX}/4, f_{XX}/8, f_{XX}/16, f_{XX}/32, f_{XX}/64, f_{XX}/128	

備考　f_{XX}：周辺クロック周波数

関連資料　関連資料は暫定版の場合がありますが，この資料では「暫定」の表示をしておりません。あらかじめご了承ください。

V850E/IA3, V850E/IA4, V850ES/IK1, V850ES/IE2 **に関する資料**

資　料　名	資料番号
V850E1　**ユーザーズ・マニュアル　アーキテクチャ編**	U14559J
V850E/IA3, V850E/IA4　**ユーザーズ・マニュアル　ハードウエア編**	U16543J
V850ES　**ユーザーズ・マニュアル　アーキテクチャ編**	U15943J
V850ES/IK1　**ユーザーズ・マニュアル　ハードウエア編**	U16910J
V850ES/IE2　**ユーザーズ・マニュアル　ハードウエア編**	U17716J
V850シリーズによるインバータ制御　アプリケーション・ノート　ホール・センサによるベクトル制御編	U17338J
V850シリーズによるインバータ制御　アプリケーション・ノート　エンコーダによるベクトル制御編	U17324J
V850シリーズによるインバータ制御　アプリケーション・ノート　ゼロクロス検出による120度通電方式制御編	U17209J
サンプル・プログラム関数使用法説明書　アプリケーション・ノート　シリアル通信（UARTA）編（V850E/IA3, V850E/IA4, V850ES/IK1, V850ES/IE2）	U18233J
サンプル・プログラム関数使用法説明書　アプリケーション・ノート　シリアル通信（CSIB）編（V850E/IA3, V850E/IA4, V850ES/IK1, V850ES/IE2）	**このマニュアル**
サンプル・プログラム関数使用法説明書　アプリケーション・ノート　DMA**機能編**（V850E/IA3, V850E/IA4）	U18235J
サンプル・プログラム関数使用法説明書　アプリケーション・ノート　タイマM**編**（V850E/IA3, V850E/IA4, V850ES/IK1, V850ES/IE2）	U18236J
サンプル・プログラム関数使用法説明書　アプリケーション・ノート　ウォッチドッグ・タイマ編（V850E/IA3, V850E/IA4, V850ES/IK1, V850ES/IE2）	U18237J
サンプル・プログラム関数使用法説明書　アプリケーション・ノート　タイマP**編**（V850E/IA3, V850E/IA4, V850ES/IK1, V850ES/IE2）	U18238J
サンプル・プログラム関数使用法説明書　アプリケーション・ノート　タイマQ**編**（V850E/IA3, V850E/IA4, V850ES/IK1, V850ES/IE2）	U18239J
サンプル・プログラム関数使用法説明書　アプリケーション・ノート　タイマENC**編**（V850E/IA3, V850E/IA4）	U18240J
サンプル・プログラム関数使用法説明書　アプリケーション・ノート　ポート機能編（V850E/IA3, V850E/IA4, V850ES/IK1, V850ES/IE2）	U18241J
サンプル・プログラム関数使用法説明書　アプリケーション・ノート　クロック・ジェネレータ編（V850E/IA3, V850E/IA4, V850ES/IK1, V850ES/IE2）	U18242J
サンプル・プログラム関数使用法説明書　アプリケーション・ノート　スタンバイ機能編（V850E/IA3, V850E/IA4, V850ES/IK1, V850ES/IE2）	U18243J
サンプル・プログラム関数使用法説明書　アプリケーション・ノート　割り込み機能編（V850E/IA3, V850E/IA4, V850ES/IK1, V850ES/IE2）	U18244J
サンプル・プログラム関数使用法説明書　アプリケーション・ノート　A/D**コンバータ**0, 1**編**（V850E/IA3, V850E/IA4, V850ES/IK1, V850ES/IE2）	U18245J
サンプル・プログラム関数使用法説明書　アプリケーション・ノート　A/D**コンバータ**2**編**（V850E/IA3, V850E/IA4）	U18246J

目　　次

クロック同期式シリアル・インタフェース B（CSIBn）（n = 0, 1）
　連続転送モード（マスタ・モード，送信モード） … 8
クロック同期式シリアル・インタフェース B（CSIBn）（n = 0, 1）
　連続転送モード（マスタ・モード，受信モード） … 13
クロック同期式シリアル・インタフェース B（CSIBn）（n = 0, 1）
　連続転送モード（マスタ・モード，送受信モード） … 18
クロック同期式シリアル・インタフェース B（CSIBn）（n = 0, 1）
　連続転送モード（スレーブ・モード，送信モード） … 24
クロック同期式シリアル・インタフェース B（CSIBn）（n = 0, 1）
　連続転送モード（スレーブ・モード，受信モード） … 29
クロック同期式シリアル・インタフェース B（CSIBn）（n = 0, 1）
　連続転送モード（スレーブ・モード，送受信モード） … 35
クロック同期式シリアル・インタフェース B（CSIBn）（n = 0, 1）
　シングル転送モード（マスタ・モード，送信モード） … 42
クロック同期式シリアル・インタフェース B（CSIBn）（n = 0, 1）
　シングル転送モード（マスタ・モード，受信モード） … 47
クロック同期式シリアル・インタフェース B（CSIBn）（n = 0, 1）
　シングル転送モード（マスタ・モード，送受信モード） … 53
クロック同期式シリアル・インタフェース B（CSIBn）（n = 0, 1）
　シングル転送モード（スレーブ・モード，送信モード） … 59
クロック同期式シリアル・インタフェース B（CSIBn）（n = 0, 1）
　シングル転送モード（スレーブ・モード，受信モード） … 64
クロック同期式シリアル・インタフェース B（CSIBn）（n = 0, 1）
　シングル転送モード（スレーブ・モード，送受信モード） … 70

クロック同期式シリアル・インタフェース B（CSIBn）（n = 0, 1）
連続転送モード（マスタ・モード，送信モード）

【機　　能】	連続転送モードで，通信モードをマスタ・モード，転送方向モードを MSB ファーストに設定し，10 回のデータ送信を行います。 なお，通信起動トリガを有効にし，通信クロック = fxx/256，転送データ長を 8 ビットに設定します。
【関　数　名】	csib1_main
【引　き　数】	なし
【処 理 内 容】	送信回数カウント（count_tx）を初期値 0 に設定します。各設定関数を呼び出したあと，送信を開始します。
【使 用 S F R】	CB0TIC　：0x07（CSIB0 送信許可割り込み要求信号（INTCB0T）クリア，マスク解除，優先レベル 7 に設定） CB0STR.CB0TSF　通信状態フラグ
【call 関数】	csib_port, csib_set, csib_start, csib_end
【変　　数】	unsigned char buf_tx[ ]　：送信データ格納バッファ volatile unsigned char count_tx　：送信カウント変数 unsigned char count　：送信データ生成変数
【割 り 込 み】	csib_int_send
【割り込み 要 因】	INTCB0T
【ファイル名】	csib1.c
【注 意 事 項】	なし

【関　数　名】	csib_port
【処 理 内 容】	ポート 4 を CSIB0 入出力端子に設定します。
【使 用 S F R】	PMC4　：0x06（$\overline{\text{SCKB0}}$ 入出力，SOB0 出力設定）
【call 関数】	なし
【変　　数】	なし
【ファイル名】	csib1.c
【注 意 事 項】	なし

【関　数　名】 csib_set

【処 理 内 容】 CSIB0 制御レジスタの設定を行います。

【使 用S F R】 CB0CTL1 ：0x06（通信タイプ 1，通信クロック = fxx/256（0.25 MHz）に設定）
CB0CTL2 ：0x00（転送データ長を 8 ビットに設定）
CB0CTL0 ：0x43（CSIB0 送信動作許可，MSB ファースト，連続転送モード，通信起動トリガ有効に設定）

【call 関数】 なし

【変 数】 なし

【ファイル名】 csib1.c

【注 意 事 項】 CB0CTL0 レジスタは CB0PWR ビット = 0 時のみ CB0TXE, CB0RXE ビットの書き換え可能です。ただし，同時に CB0PWR ビット = 1 とするのは可能です。

【関　数　名】 csib_start

【処 理 内 容】 CSIB0 動作許可し，送信データ・レジスタに値をライトします。

【使 用S F R】 CB0CTL0.CB0PWR ：1（CSIB0 動作許可）
CB0TX 送信データ・レジスタ

【call 関数】 なし

【変 数】 unsigned char buf_tx[ ] ：送信データ格納バッファ
count_tx ：送信カウント変数

【ファイル名】 csib1.c

【注 意 事 項】 なし

【関　数　名】 csib_end

【処 理 内 容】 CSIB0 動作，送信動作を禁止します。

【使 用S F R】 CB0CTL0.CB0PWR ：0（CSIB0 動作禁止）
CB0CTL0.CB0TXE ：0（CSIB0 送信動作禁止）

【call 関数】 なし

【変 数】 なし

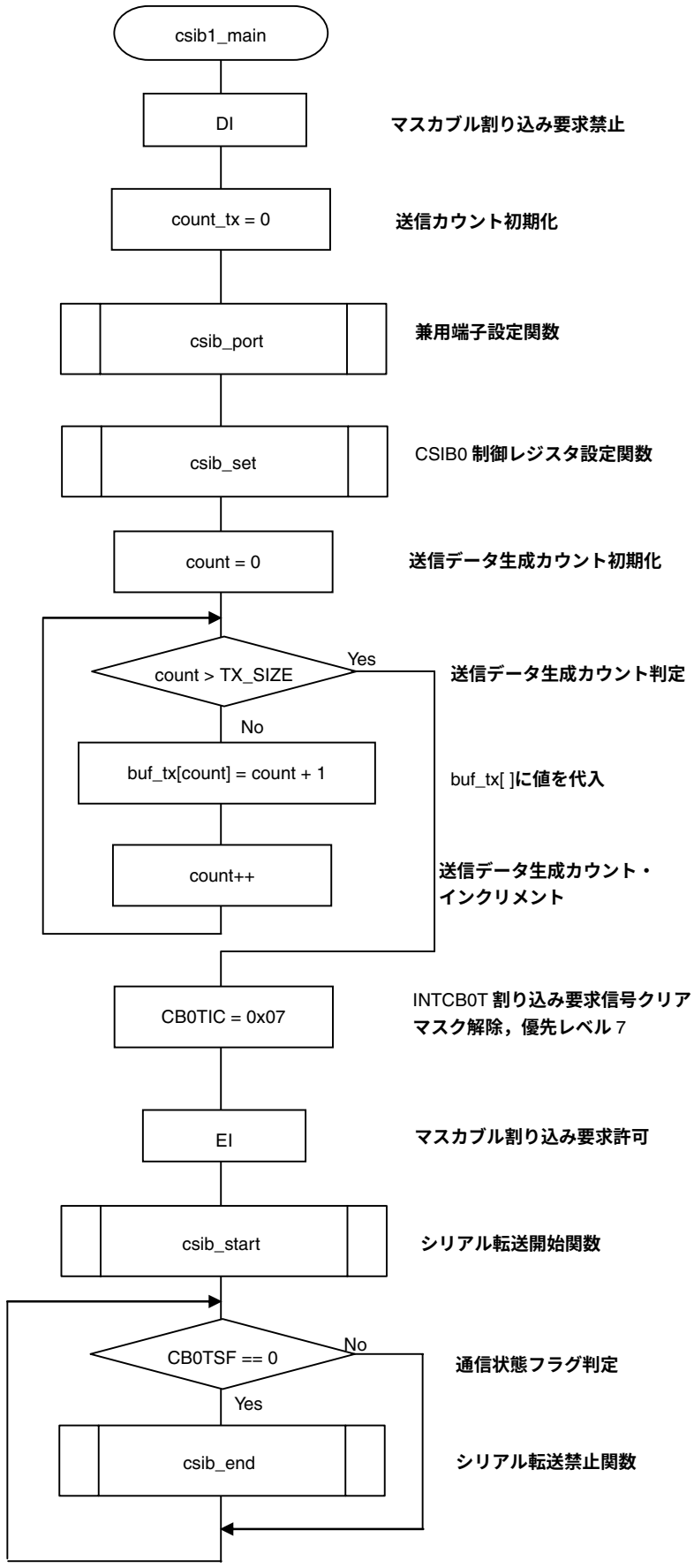
【ファイル名】 csib1.c

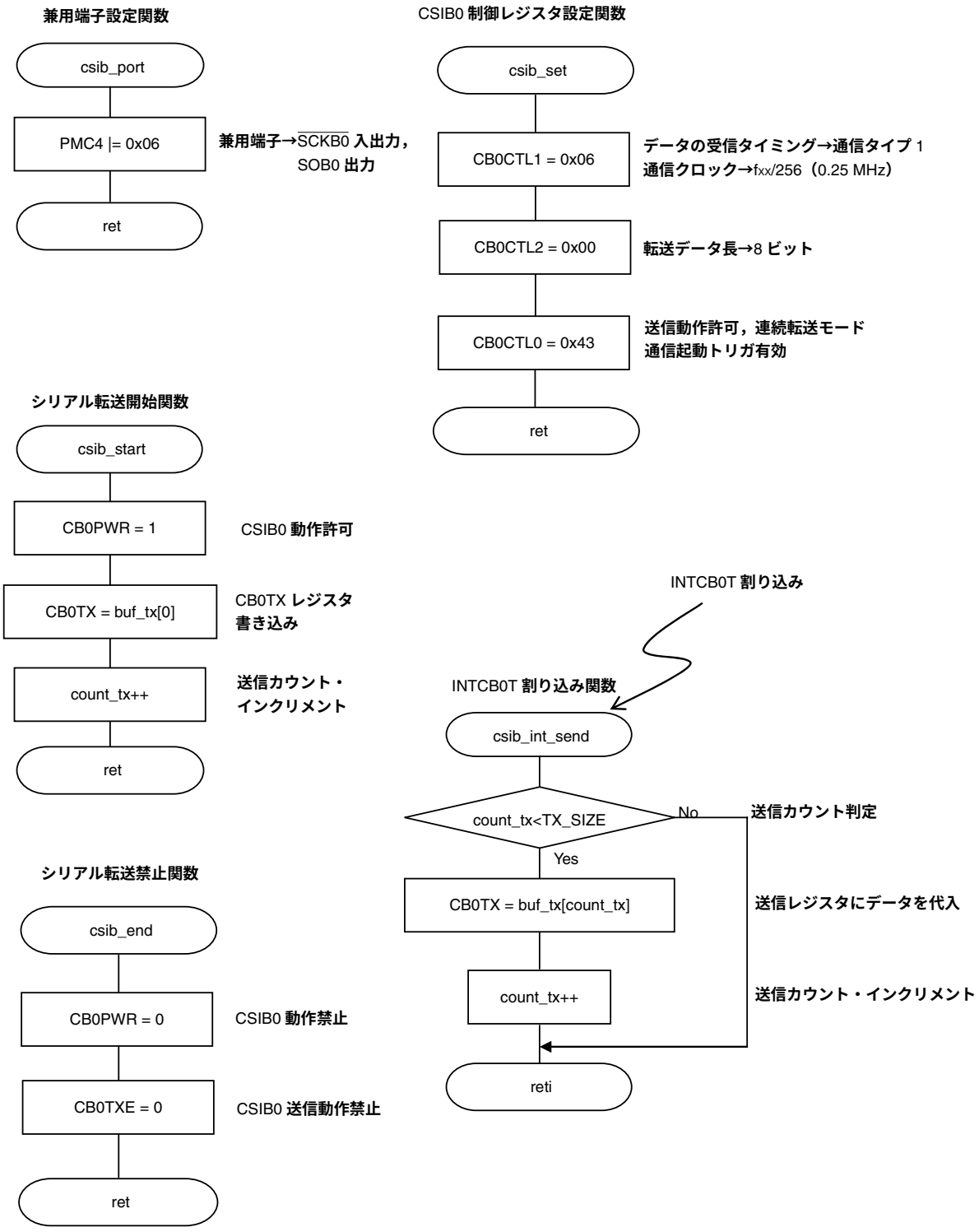
【注 意 事 項】 なし

割り込み関数

【関　数　名】	csib_int_send
【処　理　内　容】	次のデータを送信するための新たなデータをセットします。
【使　用S　F　R】	CB0TX　　送信データ・レジスタ
【call　関数】	なし
【変　数】	unsigned char buf_tx[ ]　：送信データ格納バッファ volatile unsigned char count_tx　：送信カウント変数
【フ ァ イ ル 名】	csib1.c
【注　意　事　項】	なし

連続転送モード（マスタ・モード，送信モード）（1/2）




兼用端子設定関数
csib_port
PMC4 |= 0x06
兼用端子→SCKB0 入出力，
SOB0 出力
ret
CSIB0 制御レジスタ設定関数
csib_set
CB0CTL1 = 0x06
データの受信タイミング→通信タイプ 1
通信クロック→fxx/256（0.25 MHz）
CB0CTL2 = 0x00
転送データ長→8 ビット
CB0CTL0 = 0x43
送信動作許可，連続転送モード
通信起動トリガ有効
ret
シリアル転送開始関数
csib_start
CB0PWR = 1
CSIB0 動作許可
CB0TX = buf_tx[0]
CB0TX レジスタ
書き込み
count_tx++
送信カウント・
インクリメント
ret
INTCB0T 割り込み
INTCB0T 割り込み関数
csib_int_send
count_tx<TX_SIZE
No
送信カウント判定
Yes
CB0TX = buf_tx[count_tx]
送信レジスタにデータを代入
count_tx++
送信カウント・インクリメント
reti
シリアル転送禁止関数
csib_end
CB0PWR = 0
CSIB0 動作禁止
CB0TXE = 0
CSIB0 送信動作禁止
ret

クロック同期式シリアル・インタフェース B（CSIBn）（n = 0, 1）
連続転送モード（マスタ・モード，受信モード）

【機　　能】	連続転送モードで，通信モードをマスタ・モード，転送方向モードを MSB ファーストに設定し，10 回のデータ受信を行います。 通信起動トリガを有効にし，通信クロック = fxx/256，転送データ長を 8 ビットに設定します。
【関 数 名】	csib2_main
【引 き 数】	なし
【処 理 内 容】	受信回数カウント（count_rx）を初期値 0 に設定します。各設定関数を呼び出したあと，受信を開始します。
【使 用 S F R】	CB0RIC ：0x07（CSIB0 受信終了割り込み要求信号（INTCB0R）クリア，マスク解除，優先レベル 7 に設定） CB0REIC ：0x07（CSIB0 受信エラー割り込み要求信号（INTCB0RE）クリア，マスク解除，優先レベル 7 に設定） CB0STR.CB0TSF 通信状態フラグ
【call 関数】	csib_port, csib_set, csib_start, csib_end
【変　　数】	unsigned char buf_rx[ ] ：受信データ格納バッファ volatile unsigned char count_rx ：受信カウント変数
【割 り 込 み】	csib_int_receive, csib_error
【割り込み 要 因】	INTCB0R, INTCB0RE
【ファイル名】	csib2.c
【注 意 事 項】	なし

【関 数 名】	csib_port
【処 理 内 容】	ポート 4 を CSIB0 入出力端子に設定します。
【使 用 S F R】	PMC4 ：0x05（$\overline{\text{SCKB0}}$ 入出力，SIB0 入力設定）
【call 関数】	なし
【変　　数】	なし
【ファイル名】	csib2.c
【注 意 事 項】	なし

【関　数　名】 csib_set

【処 理 内 容】 CSIB0 制御レジスタの設定を行います。

【使 用S F R】 CB0CTL1 ：0x06（通信タイプ 1，通信クロック = fxx/256（0.25 MHz）に設定）
CB0CTL2 ：0x00（転送データ長を 8 ビットに設定）
CB0CTL0 ：0x23（CSIB0 受信動作許可，MSB ファースト，連続転送モード，通信起動トリガ有効に設定）

【call 関数】 なし

【変 数】 なし

【ファイル名】 csib2.c

【注 意 事 項】 CB0CTL0 レジスタは CB0PWR ビット = 0 時のみ CB0TXE, CB0RXE ビットの書き換え可能です。ただし，同時に CB0PWR ビット = 1 とするのは可能です。

【関　数　名】 csib_start

【処 理 内 容】 CSIB0 動作許可し，受信データ・レジスタをダミー・リードします。

【使 用S F R】 CB0CTL0.CB0PWR ：1（CSIB0 動作許可）
CB0RX 受信データ・レジスタ

【call 関数】 なし

【変 数】 unsigned char buf_rx[ ] ：受信データ格納バッファ

【ファイル名】 csib2.c

【注 意 事 項】 なし

【関　数　名】 csib_end

【処 理 内 容】 CSIB0 動作，受信動作を禁止します。

【使 用S F R】 CB0CTL0.CB0PWR ：0（CSIB0 動作禁止）
CB0CTL0.CB0RXE ：0（CSIB0 受信動作禁止）

【call 関数】 なし

【変 数】 なし

【ファイル名】 csib2.c

【注 意 事 項】 なし

割り込み関数

【関　数　名】 csib_int_receive

【処　理　内　容】 受信したデータをバッファに格納します。

【使　用S　F　R】 CB0RX　　受信データ・レジスタ
CB0CTL0.CB0SCE　：0（通信起動トリガ無効）

【call　　関数】 なし

【変　　数】 unsigned char buf_rx[ ]　　：受信データ格納バッファ
volatile unsigned char count_rx　：受信カウント変数

【フ ァ イ ル 名】 csib2.c

【注　意　事　項】 なし

【関　数　名】 csib_error

【処　理　内　容】 受信エラー・フラグをクリアします。

【使　用S　F　R】 CB0RX　　受信データ・レジスタ
CB0STR.CB0OVE　：0（オーバラン・エラー・フラグ・クリア）

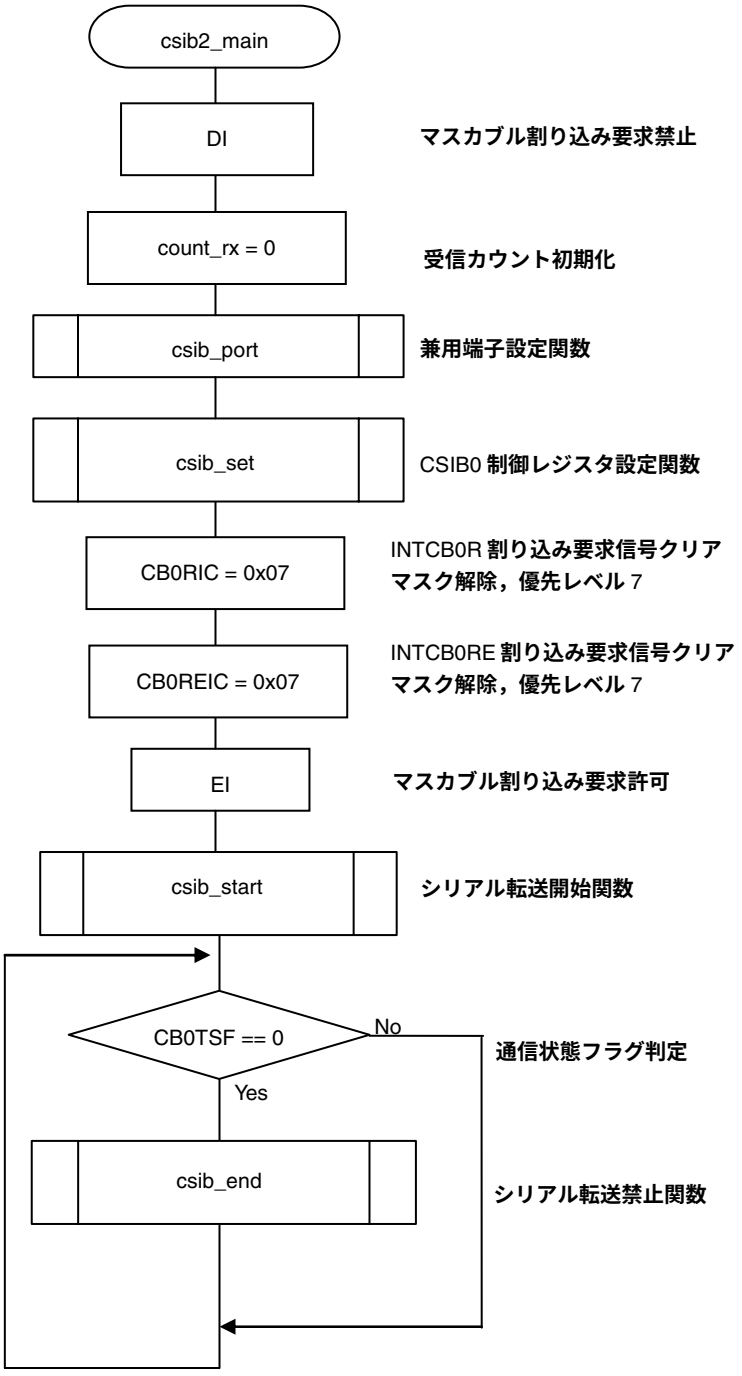
【call　　関数】 なし

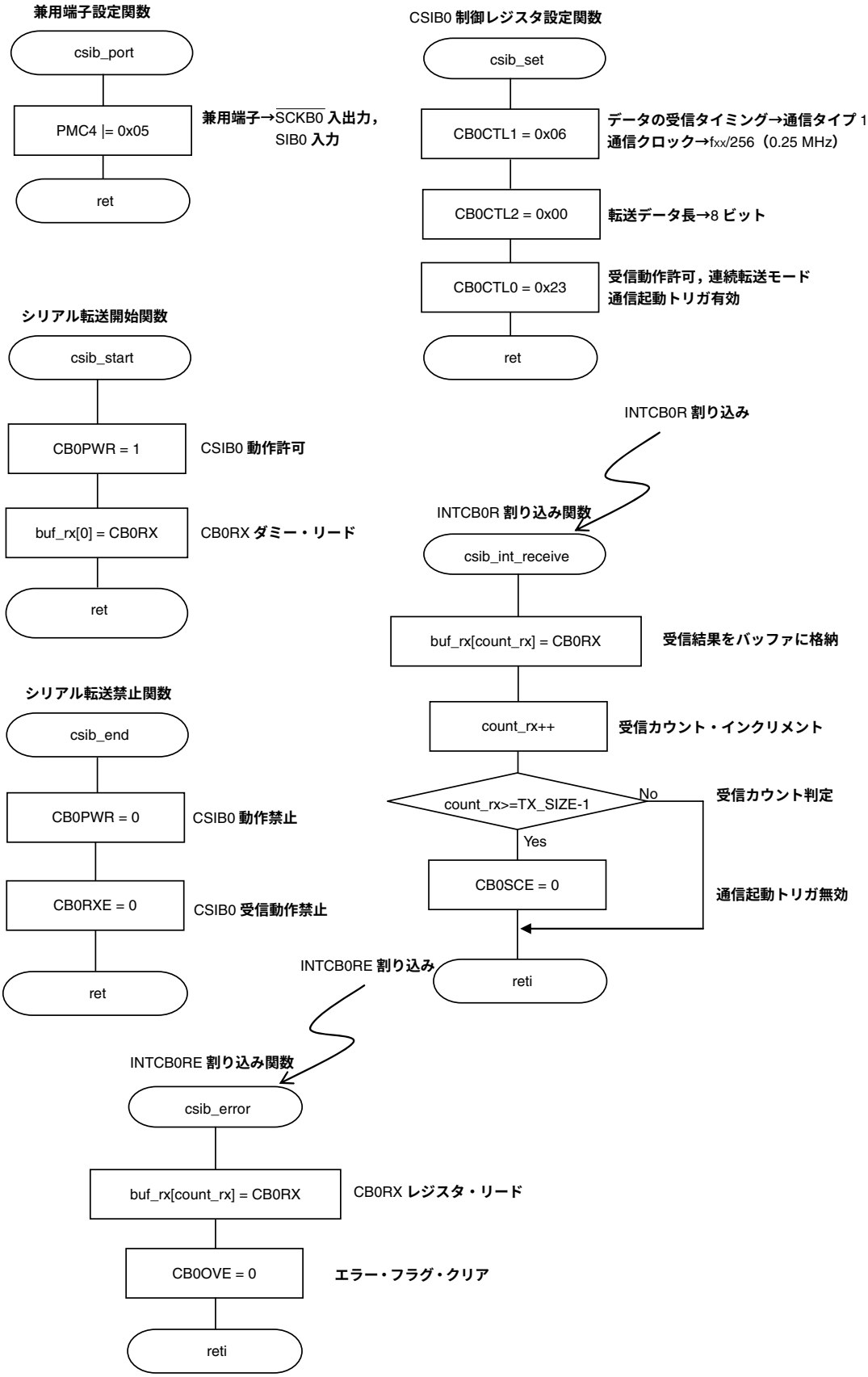
【変　　数】 unsigned char buf_rx[ ]　　：受信データ格納バッファ
volatile unsigned char count_rx　：受信カウント変数

【フ ァ イ ル 名】 csib2.c

【注　意　事　項】 なし

連続転送モード（マスタ・モード，受信モード）（1/2）




兼用端子設定関数
csib_port
PMC4 |= 0x05
兼用端子→SCKB0 入出力，
SIB0 入力
ret
CSIB0 制御レジスタ設定関数
csib_set
CB0CTL1 = 0x06
データの受信タイミング→通信タイプ 1
通信クロック→fxx/256（0.25 MHz）
CB0CTL2 = 0x00
転送データ長→8 ビット
CB0CTL0 = 0x23
受信動作許可，連続転送モード
通信起動トリガ有効
ret
シリアル転送開始関数
csib_start
CB0PWR = 1
CSIB0 動作許可
buf_rx[0] = CB0RX
CB0RX ダミー・リード
ret
INTCB0R 割り込み
INTCB0R 割り込み関数
csib_int_receive
buf_rx[count_rx] = CB0RX
受信結果をバッファに格納
count_rx++
受信カウント・インクリメント
count_rx>=TX_SIZE-1
No
受信カウント判定
Yes
CB0SCE = 0
通信起動トリガ無効
reti
シリアル転送禁止関数
csib_end
CB0PWR = 0
CSIB0 動作禁止
CB0RXE = 0
CSIB0 受信動作禁止
ret
INTCB0RE 割り込み
INTCB0RE 割り込み関数
csib_error
buf_rx[count_rx] = CB0RX
CB0RX レジスタ・リード
CB0OVE = 0
エラー・フラグ・クリア
reti

クロック同期式シリアル・インタフェース B（CSIBn）（n = 0, 1）
連続転送モード（マスタ・モード，送受信モード）

【機　　能】	連続転送モードで，通信モードをマスタ・モード，転送方向モードを MSB ファーストに設定し，送受信を各 10 回行います。 通信起動トリガを有効にし，通信クロック = fxx/256，転送データ長を 8 ビットに設定します。
【関 数 名】	csib3_main
【引 き 数】	なし
【処 理 内 容】	送信回数カウント（count_tx）を初期値 0 に設定します。受信回数カウント（count_rx）を初期値 0 に設定し，各設定関数を呼び出したあと，送受信を開始します。
【使 用 S F R】	CB0REIC ：0x07（CSIB0 受信エラー割り込み（INTCB0RE）処理許可） CB0RIC ：0x07（CSIB0 受信終了割り込み（INTCB0R）処理許可） CB0TIC ：0x07（CSIB0 送信許可割り込み（INTCB0T）処理許可） CB0STR.CB0TSF 通信状態フラグ
【call 関数】	csib_port, csib_set, csib_start, csib_end
【変 数】	unsigned char buf_tx[ ] ：送信データ格納バッファ unsigned char buf_rx[ ] ：受信データ格納バッファ volatile unsigned char count_tx ：送信回数カウント変数 volatile unsigned char count_rx ：受信回数カウント変数 unsigned char count ：送信データ生成変数
【割 り 込 み】	csib_error, csib_int_send, csib_int_receive
【割り込み 要 因】	INTCB0RE, INTCB0R, INTCB0T
【ファイル名】	csib3.c
【注 意 事 項】	なし

【関 数 名】	csib_port
【処 理 内 容】	ポート 4 を CSIB0 入出力端子に設定します。
【使 用 S F R】	PMC4 ：0x07（$\overline{\text{SCKB0}}$ 入出力，SOB0 出力，SIB0 入力設定）
【call 関数】	なし
【変 数】	なし
【ファイル名】	csib3.c
【注 意 事 項】	なし

【関　数　名】 csib_set

【処 理 内 容】 CSIB0 制御レジスタの設定を行います。

【使 用S F R】 CB0CTL1 ：0x06（通信タイプ 1，通信クロック = fxx/256（0.25 MHz）に設定）
CB0CTL2 ：0x00（転送データ長を 8 ビットに設定）
CB0CTL0 ：0x63（CSIB0 送信動作許可，受信動作許可，MSB ファースト，
連続転送モード，通信起動トリガ有効に設定）

【call 関数】 なし

【変 数】 なし

【ファイル名】 csib3.c

【注 意 事 項】 CB0CTL0 レジスタは CB0PWR ビット = 0 時のみ CB0TXE, CB0RXE ビットの書き換え可能です。ただし，同時に CB0PWR ビット = 1 とするのは可能です。

【関　数　名】 csib_start

【処 理 内 容】 CSIB0 動作許可し，送信データ・レジスタに値をライトします。

【使 用S F R】 CB0CTL0.CB0PWR ：1（CSIB0 動作許可）
CB0TX 送信データ・レジスタ

【call 関数】 なし

【変 数】 unsigned char buf_tx[ ] ：送信データ格納バッファ
volatile char count_tx ：送信カウント変数

【ファイル名】 csib3.c

【注 意 事 項】 なし

【関　数　名】 csib_end

【処 理 内 容】 CSIB0 動作，送受信動作を禁止します。

【使 用S F R】 CB0CTL0.CB0PWR ：0（CSIB0 動作禁止）
CB0CTL0.CB0RXE ：0（CSIB0 受信動作禁止）
CB0CTL0.CB0TXE ：0（CSIB0 送信動作禁止）

【call 関数】 なし

【変 数】 なし

【ファイル名】 csib3.c

【注 意 事 項】 なし

【関　数　名】 csib_error

【処 理 内 容】 受信エラー・フラグをクリアします。

【使 用S F R】 CB0RX　　受信データ・レジスタ
CB0STR.CB0OVE　：0（オーバラン・エラー・フラグ・クリア）

【call　　関数】 なし

【変　　数】 unsigned char buf_rx[ ]　　：受信データ格納バッファ
volatile unsigned char count_rx ：受信カウント変数

【ファイル名】 csib3.c

【注 意 事 項】 なし

【関　数　名】 csib_int_send

【処 理 内 容】 次のデータを送信するための新たなデータをセットします。

【使 用S F R】 CB0TX　　送信データ・レジスタ

【call　　関数】 なし

【変　　数】 unsigned char buf_tx[ ]　　：送信データ格納バッファ
volatile unsigned char count_tx ：送信カウント変数

【ファイル名】 csib3.c

【注 意 事 項】 なし

【関　数　名】 csib_int_receive

【処 理 内 容】 受信したデータをバッファに格納します。

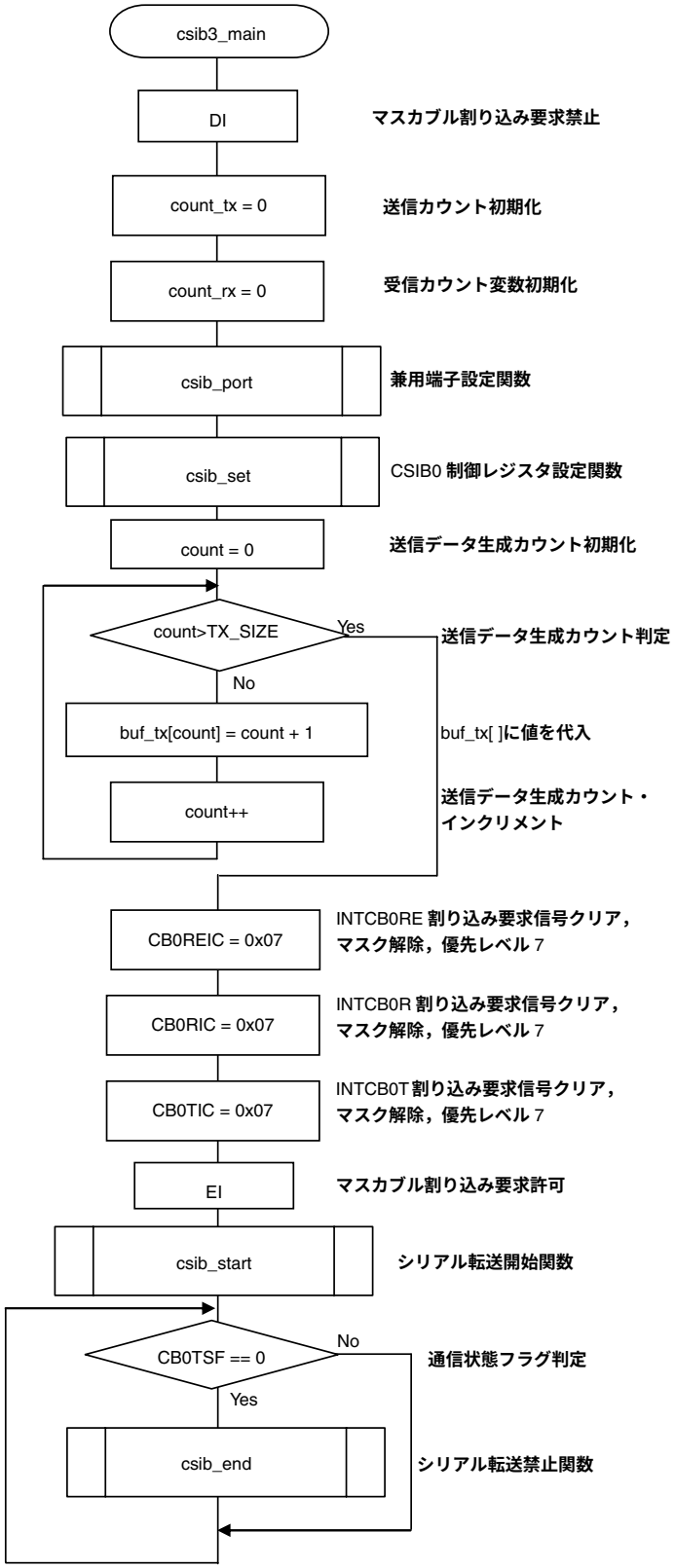
【使 用S F R】 CB0RX　　受信データ・レジスタ

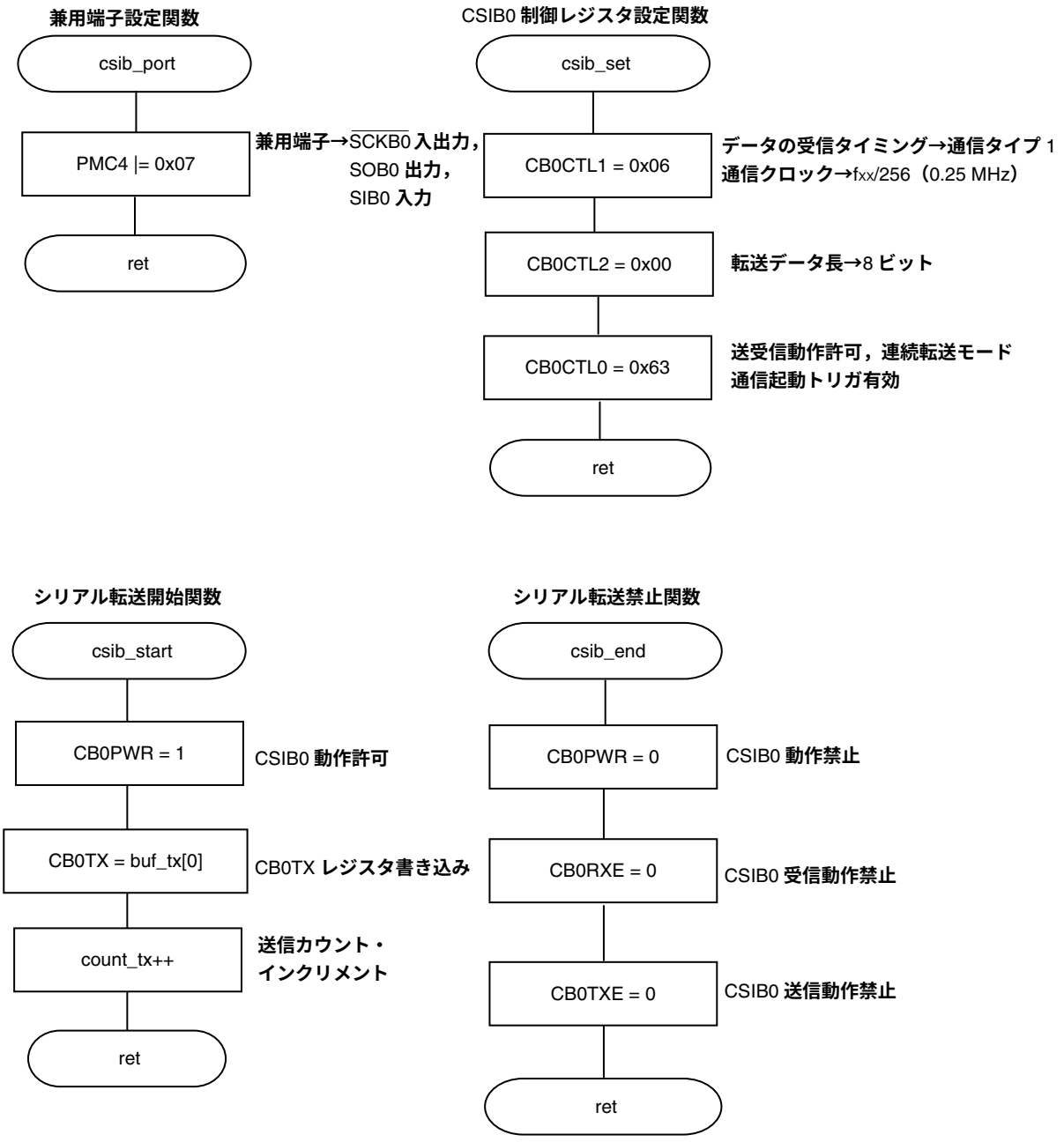
【call　　関数】 なし

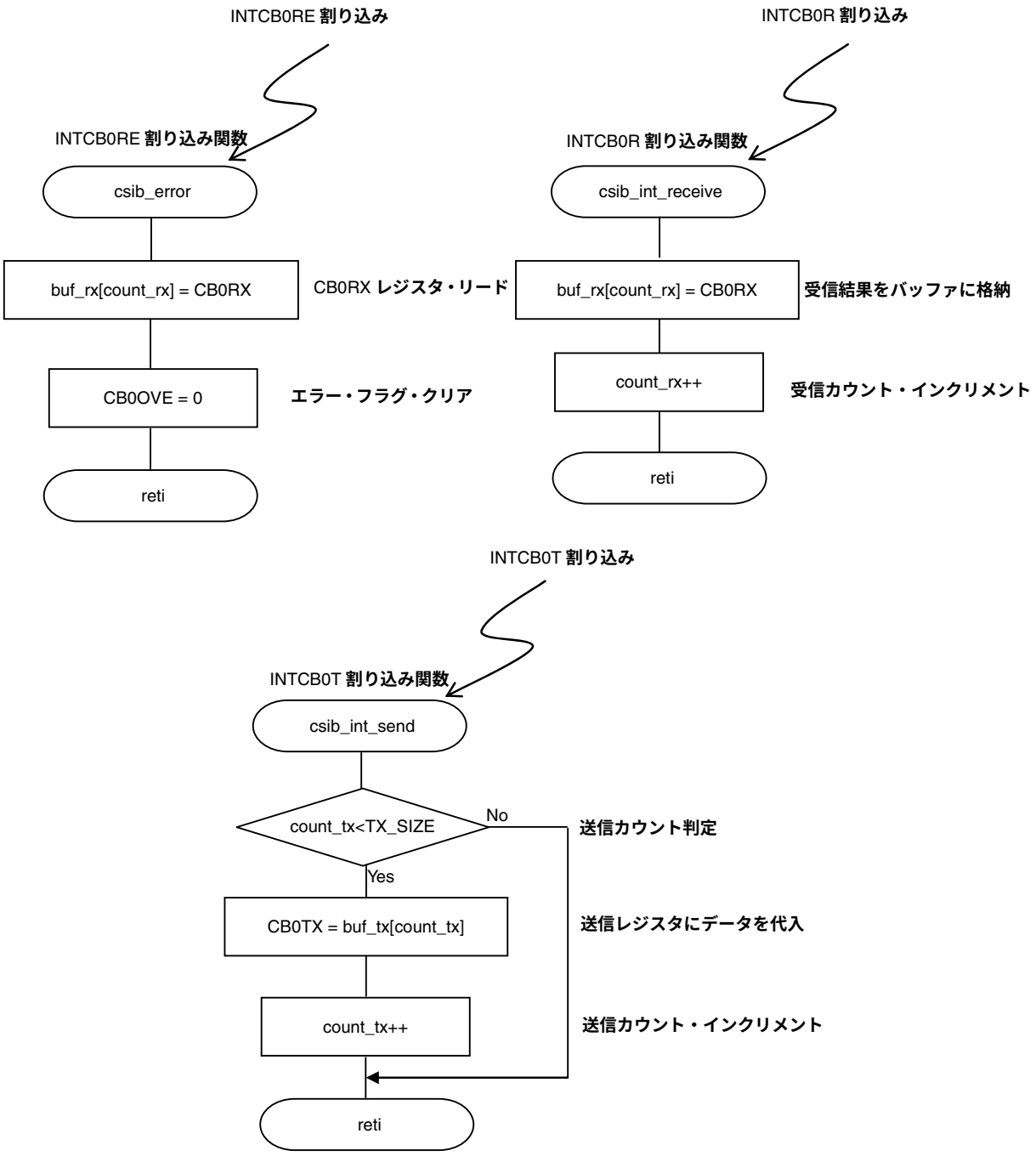
【変　　数】 unsigned char buf_rx[ ]　　：受信データ格納バッファ
volatile unsigned char count_rx ：受信カウント変数

【ファイル名】 csib3.c

【注 意 事 項】 なし


csib3_main
DI
マスカブル割り込み要求禁止
count_tx = 0
送信カウント初期化
count_rx = 0
受信カウント変数初期化
csib_port
兼用端子設定関数
csib_set
CSIB0 制御レジスタ設定関数
count = 0
送信データ生成カウント初期化
count>TX_SIZE
Yes
送信データ生成カウント判定
No
buf_tx[count] = count + 1
buf_tx[ ]に値を代入
count++
送信データ生成カウント・
インクリメント
CB0REIC = 0x07
INTCB0RE 割り込み要求信号クリア，
マスク解除，優先レベル 7
CB0RIC = 0x07
INTCB0R 割り込み要求信号クリア，
マスク解除，優先レベル 7
CB0TIC = 0x07
INTCB0T割り込み要求信号クリア，
マスク解除，優先レベル 7
EI
マスカブル割り込み要求許可
csib_start
シリアル転送開始関数
CB0TSF == 0
No
通信状態フラグ判定
Yes
csib_end
シリアル転送禁止関数


兼用端子設定関数
csib_port
PMC4 |= 0x07
兼用端子→SCKB0 入出力，
SOB0 出力，
SIB0 入力
ret
CSIB0 制御レジスタ設定関数
csib_set
CB0CTL1 = 0x06
データの受信タイミング→通信タイプ 1
通信クロック→fxx/256（0.25 MHz）
CB0CTL2 = 0x00
転送データ長→8 ビット
CB0CTL0 = 0x63
送受信動作許可，連続転送モード
通信起動トリガ有効
ret
シリアル転送開始関数
csib_start
CB0PWR = 1
CSIB0 動作許可
CB0TX = buf_tx[0]
CB0TX レジスタ書き込み
count_tx++
送信カウント・
インクリメント
ret
シリアル転送禁止関数
csib_end
CB0PWR = 0
CSIB0 動作禁止
CB0RXE = 0
CSIB0 受信動作禁止
CB0TXE = 0
CSIB0 送信動作禁止
ret


INTCB0RE 割り込み
INTCB0RE 割り込み関数
csib_error
buf_rx[count_rx] = CB0RX
CB0RX レジスタ・リード
CB0OVE = 0
エラー・フラグ・クリア
reti
INTCB0R 割り込み
INTCB0R 割り込み関数
csib_int_receive
buf_rx[count_rx] = CB0RX
受信結果をバッファに格納
count_rx++
受信カウント・インクリメント
reti
INTCB0T 割り込み
INTCB0T 割り込み関数
csib_int_send
count_tx<TX_SIZE
No
送信カウント判定
Yes
CB0TX = buf_tx[count_tx]
送信レジスタにデータを代入
count_tx++
送信カウント・インクリメント
reti

クロック同期式シリアル・インタフェース B（CSIBn）（n = 0, 1）
連続転送モード（スレーブ・モード，送信モード）

【機　　能】	連続転送モードで，通信モードをスレーブ・モード，転送方向モードを MSB ファーストに設定し，10 回のデータ送信を行います。 通信起動トリガを有効にし，通信クロックは外部クロック，転送データ長を 8 ビットに設定します。
【関 数 名】	csib4_main
【引 き 数】	なし
【処 理 内 容】	送信回数カウント（count_tx）を初期値 0 に設定します。各設定関数を呼び出したあと，送信を開始します。
【使 用 S F R】	CB0TIC　：0x07（CSIB0 送信許可割り込み要求信号（INTCB0T）クリア，マスク解除，優先レベル 7 に設定） CB0STR.CB0TSF　通信状態フラグ確認
【call 関数】	csib_port, csib_set, csib_start, csib_end
【変 数】	unsigned char buf_tx[ ]　：送信データ格納バッファ volatile unsigned char count_tx　：送信カウント変数 unsigned char count　：送信データ生成変数
【割 り 込 み】	csib_int_send
【割り込み 要 因】	INTCB0T
【フ ァ イ ル 名】	csib4.c
【注 意 事 項】	なし

【関 数 名】	csib_port
【処 理 内 容】	ポート 4 を CSIB0 入出力端子に設定します。
【使 用 S F R】	PMC4　：0x06（$\overline{\text{SCKB0}}$ 入出力，SOB0 出力設定）
【call 関数】	なし
【変 数】	なし
【フ ァ イ ル 名】	csib4.c
【注 意 事 項】	なし

【関　数　名】 csib_set

【処　理　内　容】 CSIB0 制御レジスタの設定を行います。

【使　用 S　F　R】 CB0CTL1　：0x07（通信タイプ 1，外部クロックに設定）
CB0CTL2　：0x00（転送データ長を 8 ビットに設定）
CB0CTL0　：0x43（CSIB0 送信動作許可，MSB ファースト，連続転送モード，通信起動トリガ有効に設定）

【call　関数】 なし

【変　数】 なし

【ファイル名】 csib4.c

【注　意　事　項】 CB0CTL0 レジスタは CB0PWR ビット = 0 時のみ CB0TXE, CB0RXE ビットの書き換え可能です。ただし，同時に CB0PWR ビット = 1 とするのは可能です。

【関　数　名】 csib_start

【処　理　内　容】 CSIB0 動作許可し，送信データ・レジスタに値をライトします。

【使　用 S　F　R】 CB0CTL0.CB0PWR ：1（CSIB0 動作許可）
CB0TX　送信データ・レジスタ

【call　関数】 なし

【変　数】 unsigned char buf_tx[ ]　：送信データ格納バッファ
volatile unsigned char count_tx　：送信カウント変数

【ファイル名】 csib4.c

【注　意　事　項】 なし

【関　数　名】 csib_end

【処　理　内　容】 CSIB0 動作，送信動作を禁止する。

【使　用 S　F　R】 CB0CTL0.CB0PWR ：0（CSIB0 動作禁止）
CB0CTL0.CB0TXE　：0（CSIB0 送信動作禁止）

【call　関数】 なし

【変　数】 なし

【ファイル名】 csib4.c

【注　意　事　項】 なし

【関　数　名】 csib_int_send

【処　理　内　容】 次のデータを送信するための新たなデータをセットします。

【使　用S　F　R】 CB0TX　　送信データ・レジスタ

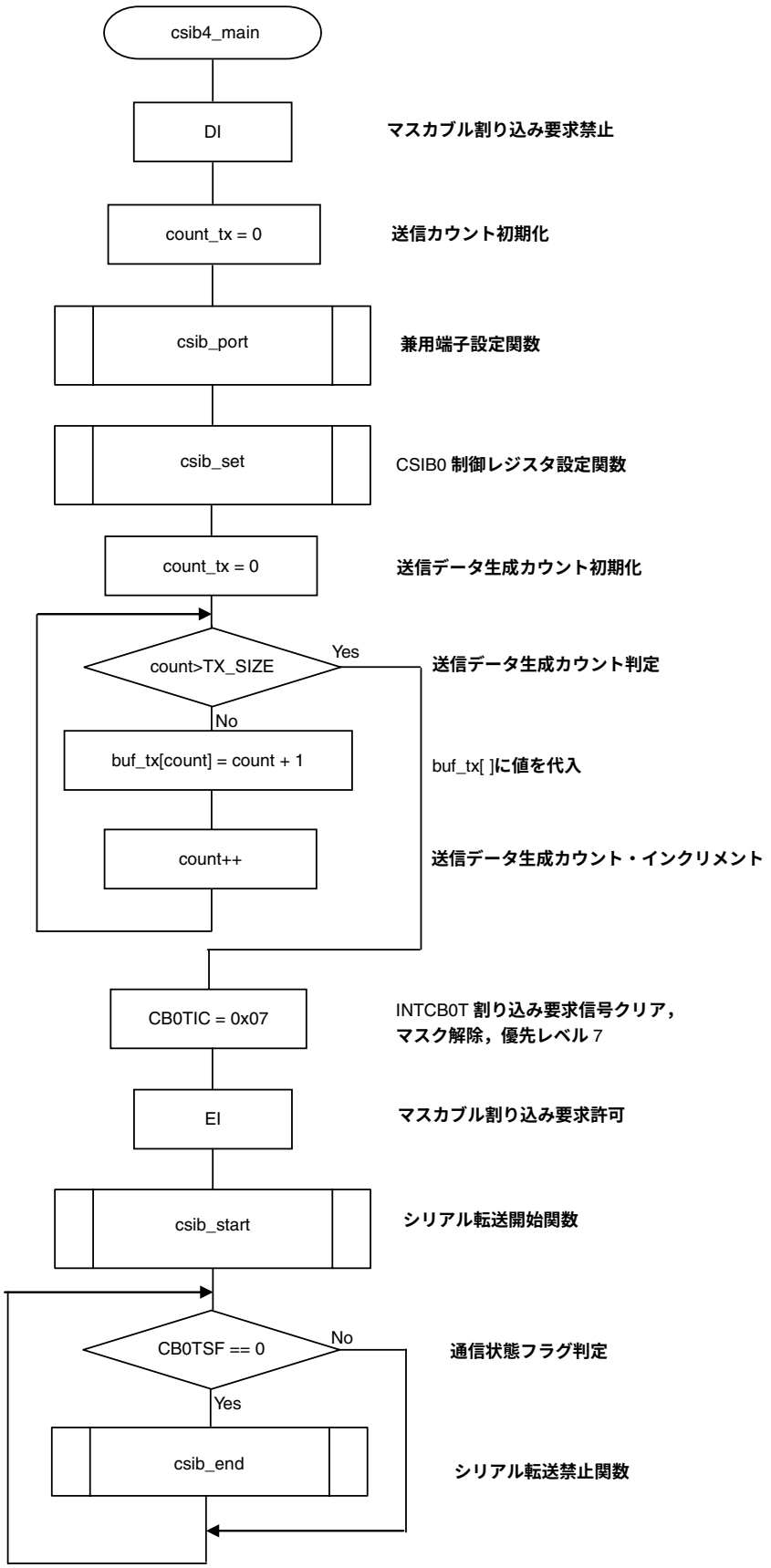
【call　　関数】 なし

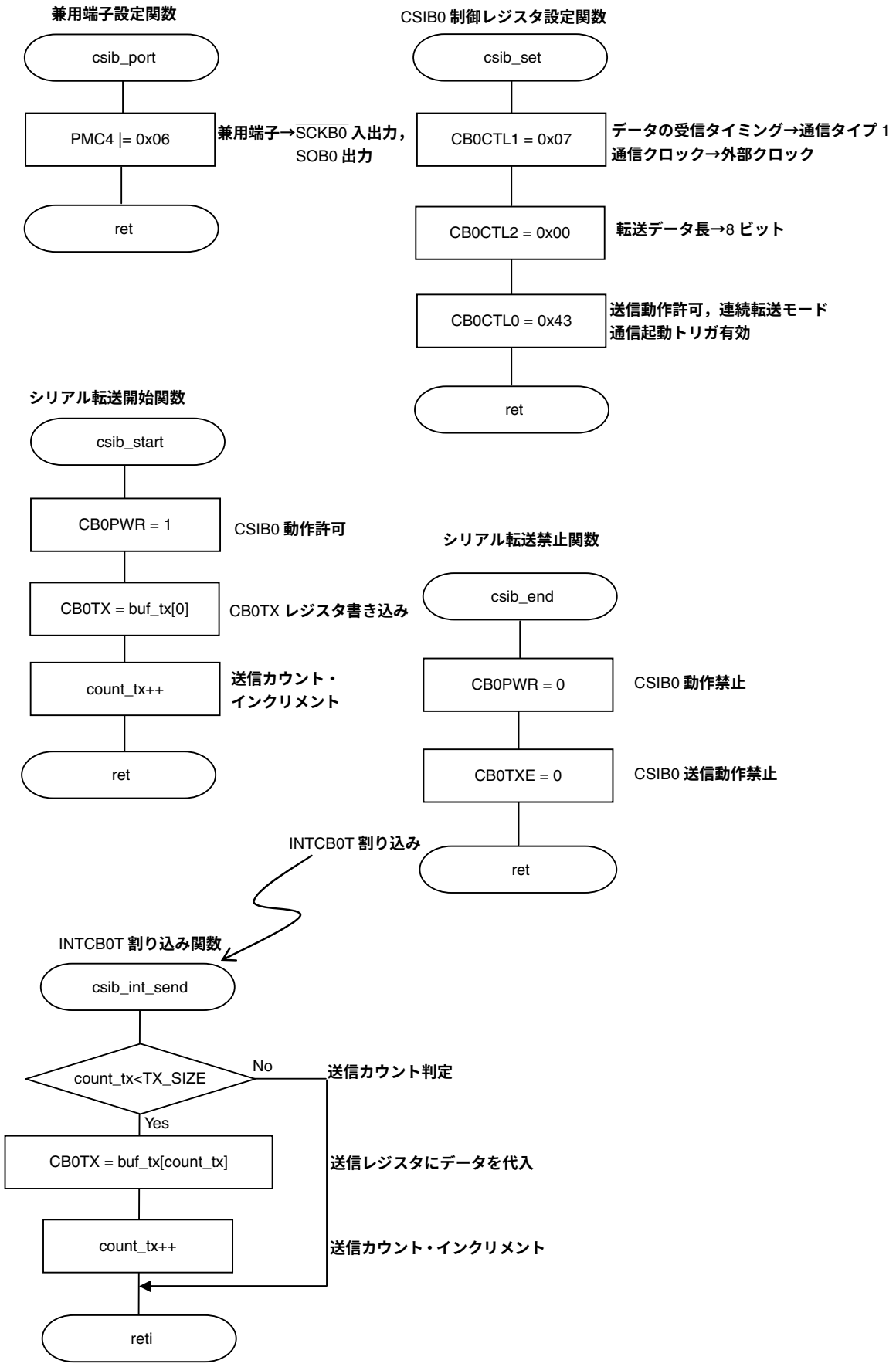
【変　　数】 unsigned char buf_tx[ ]　　：送信データ格納バッファ
volatile unsigned char count_tx　：送信カウント変数

【フ ァ イ ル 名】 csib4.c

【注　意　事　項】 なし

連続転送モード（スレーブ・モード，送信モード）（1/2）




兼用端子設定関数
csib_port
PMC4 |= 0x06
兼用端子→SCKB0 入出力，
SOB0 出力
ret
CSIB0 制御レジスタ設定関数
csib_set
CB0CTL1 = 0x07
データの受信タイミング→通信タイプ 1
通信クロック→外部クロック
CB0CTL2 = 0x00
転送データ長→8 ビット
CB0CTL0 = 0x43
送信動作許可，連続転送モード
通信起動トリガ有効
ret
シリアル転送開始関数
csib_start
CB0PWR = 1
CSIB0 動作許可
CB0TX = buf_tx[0]
CB0TX レジスタ書き込み
count_tx++
送信カウント・
インクリメント
ret
シリアル転送禁止関数
csib_end
CB0PWR = 0
CSIB0 動作禁止
CB0TXE = 0
CSIB0 送信動作禁止
ret
INTCB0T 割り込み
INTCB0T 割り込み関数
csib_int_send
count_tx<TX_SIZE
No
送信カウント判定
Yes
CB0TX = buf_tx[count_tx]
送信レジスタにデータを代入
count_tx++
送信カウント・インクリメント
reti

クロック同期式シリアル・インタフェース B（CSIBn）（n = 0, 1）
連続転送モード（スレーブ・モード，受信モード）

【機　　能】	連続転送モードで，通信モードをスレーブ・モード，転送方向モードを MSB ファーストに設定し，10 回のデータ受信を行います。 通信起動トリガを有効にし，通信クロックは外部クロック，転送データ長を 8 ビットに設定します。
【関 数 名】	csib5_main
【引 き 数】	なし
【処 理 内 容】	受信回数カウント（count_rx）を初期値 0 に設定します。各設定関数を呼び出したあと，受信を開始します。
【使 用 S F R】	CB0RIC ：0x07（CSIB0 受信終了割り込み要求信号（INTCB0R）クリア，マスク解除，優先レベル 7 に設定） CB0REIC ：0x07（CSIB0 受信エラー割り込み要求信号（INTCB0RE）クリア，マスク解除，優先レベル 7 に設定） CB0STR.CB0TSF 通信状態フラグ確認
【call 関数】	csib_port, csib_set, csib_start, csib_end
【変　　数】	unsigned char buf_rx[ ] ：受信データ格納バッファ volatile unsigned char count_rx ：受信カウント変数
【割 り 込 み】	csib_int_receive, csib_error
【割り込み 要 因】	INTCB0R, INTCB0RE
【フ ァ イ ル 名】	csib5.c
【注 意 事 項】	なし

【関 数 名】	csib_port
【処 理 内 容】	ポート 4 を CSIB0 入出力端子に設定します。
【使 用 S F R】	PMC4 ：0x05（$\overline{\text{SCKB0}}$ 入出力，SIB0 入力設定）
【call 関数】	なし
【変　　数】	なし
【フ ァ イ ル 名】	csib5.c
【注 意 事 項】	なし

【関　数　名】 csib_set

【処 理 内 容】 CSIB0 制御レジスタの設定を行います。

【使 用S F R】 CB0CTL1 ：0x07（通信タイプ 1，通信クロックを外部クロックに設定）
CB0CTL2 ：0x00（転送データ長を 8 ビットに設定）
CB0CTL0 ：0x23（CSIB0 受信動作許可，MSB ファースト，連続転送モード，通信起動トリガ有効に設定）

【call 関数】 なし

【変 数】 なし

【ファイル名】 csib5.c

【注 意 事 項】 CB0CTL0 レジスタは CB0PWR ビット = 0 時のみ CB0TXE, CB0RXE ビットの書き換え可能です。ただし，同時に CB0PWR ビット = 1 とするのは可能です。

【関　数　名】 csib_start

【処 理 内 容】 CSIB0 動作許可し，受信データ・レジスタをダミー・リードします。

【使 用S F R】 CB0CTL0.CB0PWR ：1（CSIB0 動作許可）
CB0RX 受信データ・レジスタ

【call 関数】 なし

【変 数】 unsigned char buf_rx[ ] ：受信データ格納バッファ

【ファイル名】 csib5.c

【注 意 事 項】 なし

【関　数　名】 csib_end

【処 理 内 容】 CSIB0 動作，受信動作を禁止します。

【使 用S F R】 CB0CTL0.CB0PWR ：0（CSIB0 動作禁止）
CB0CTL0.CB0RXE ：0（CSIB0 受信動作禁止）

【call 関数】 なし

【変 数】 なし

【ファイル名】 csib5.c

【注 意 事 項】 なし

割り込み関数

【関　数　名】 csib_int_receive

【処　理　内　容】 受信したデータをバッファに格納します。

【使　用S　F　R】 CB0RX　　受信データ・レジスタ
CB0CTL0.CB0SCE　：0（通信起動トリガ無効）

【call　　関数】 なし

【変　　数】 unsigned char buf_rx[ ]　　：受信データ格納バッファ
volatile unsigned char count_rx　：受信カウント変数

【フ ァ イ ル 名】 csib5.c

【注　意　事　項】 なし

【関　数　名】 csib_error

【処　理　内　容】 受信エラー・フラグをクリアします。

【使　用S　F　R】 CB0RX　　受信データ・レジスタ
CB0STR.CB0OVE　：0（オーバラン・エラー・フラグ・クリア）

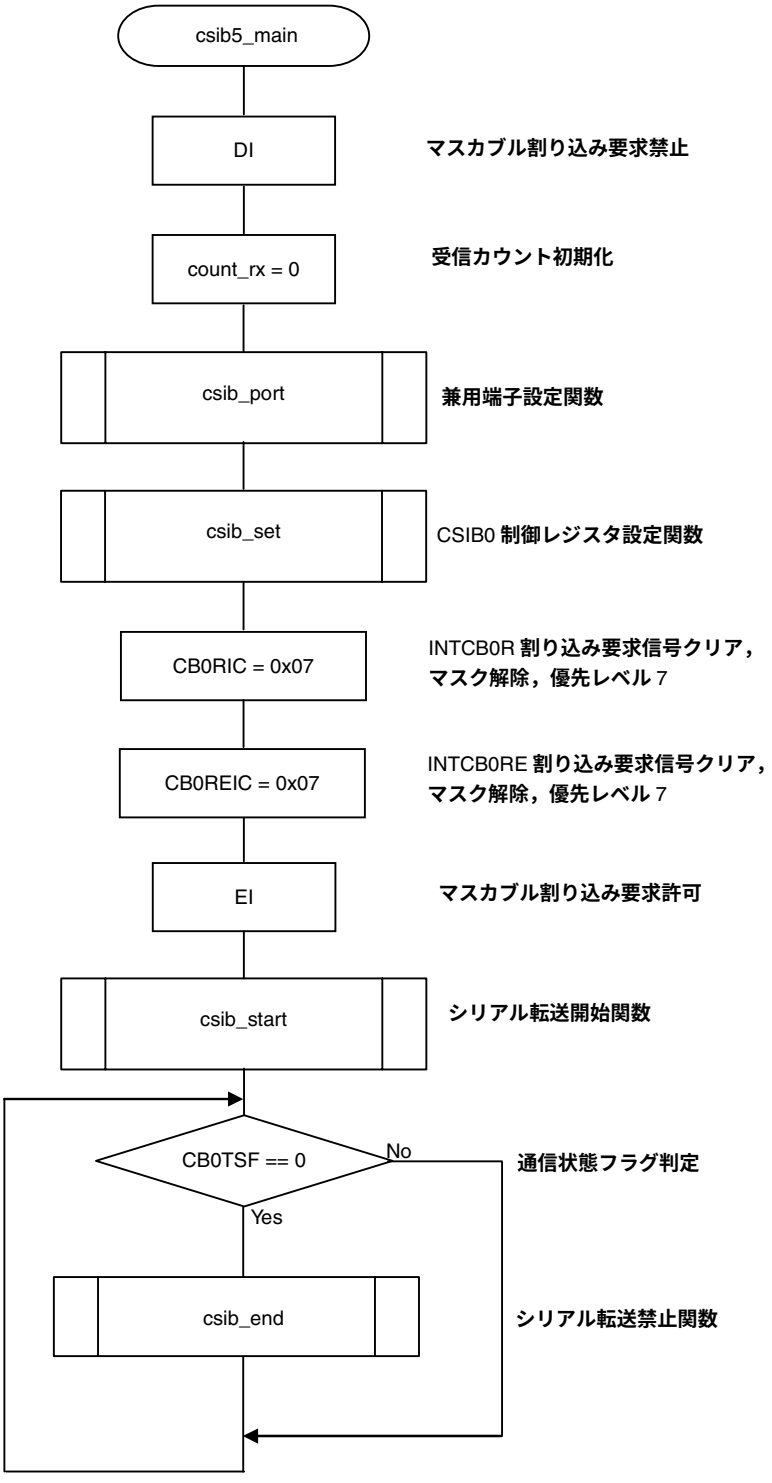
【call　　関数】 なし

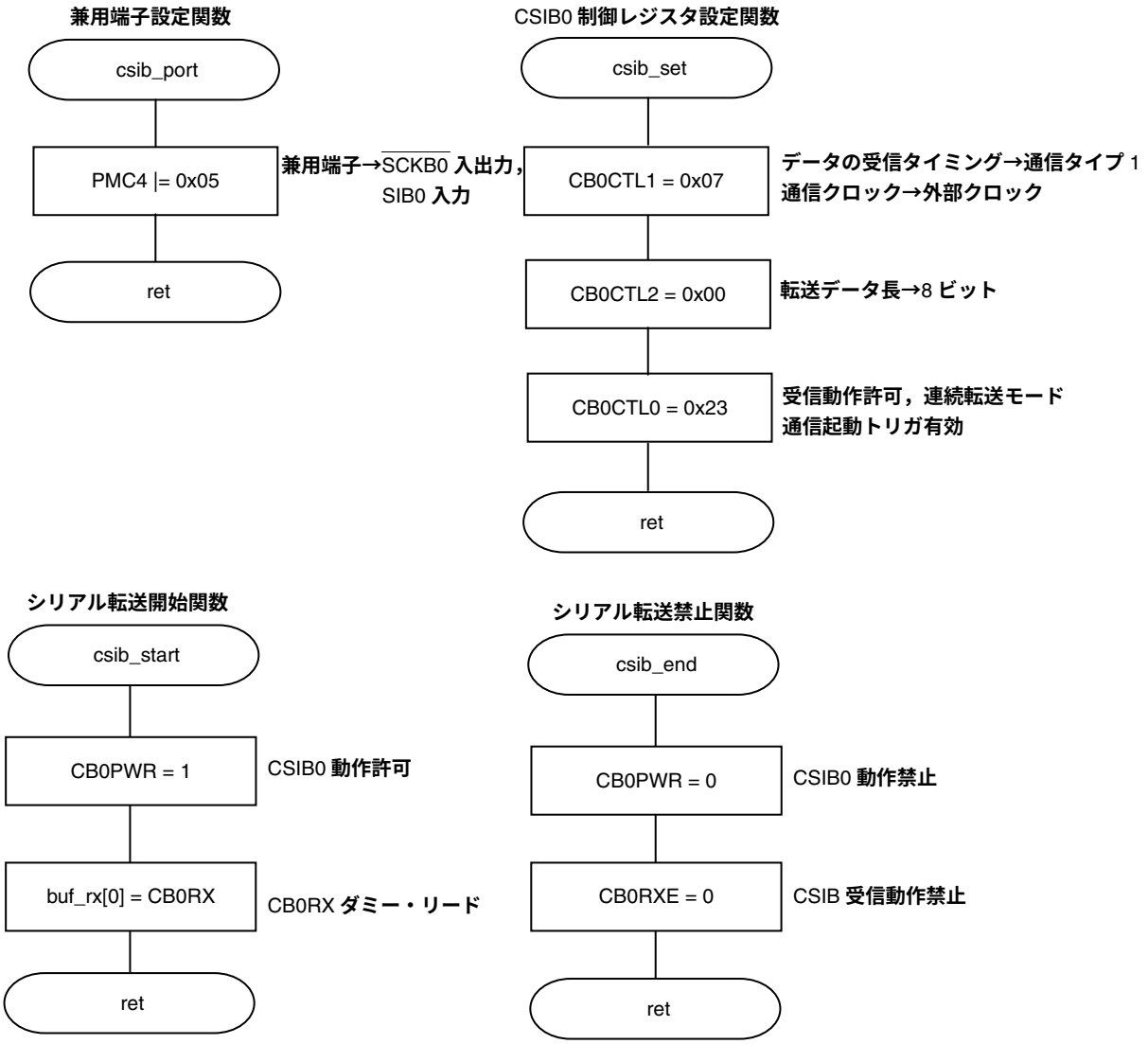
【変　　数】 unsigned char buf_rx[ ]　　：受信データ格納バッファ
volatile unsigned char count_rx　：受信カウント変数

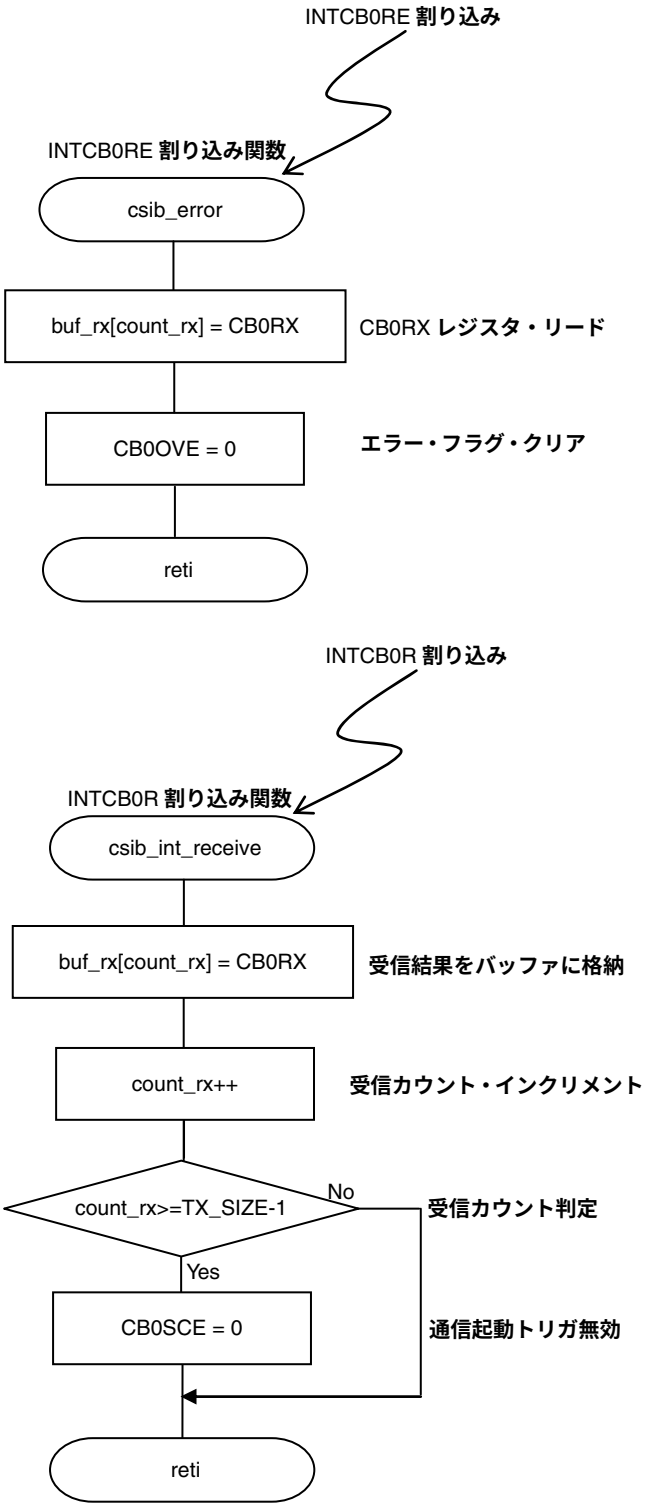
【フ ァ イ ル 名】 csib5.c

【注　意　事　項】 なし

連続転送モード（スレーブ・モード，受信モード）（1/3）




兼用端子設定関数
csib_port
PMC4 |= 0x05
兼用端子→SCKB0 入出力，
SIB0 入力
ret
CSIB0 制御レジスタ設定関数
csib_set
CB0CTL1 = 0x07
データの受信タイミング→通信タイプ 1
通信クロック→外部クロック
CB0CTL2 = 0x00
転送データ長→8 ビット
CB0CTL0 = 0x23
受信動作許可，連続転送モード
通信起動トリガ有効
ret
シリアル転送開始関数
csib_start
CB0PWR = 1
CSIB0 動作許可
buf_rx[0] = CB0RX
CB0RX ダミー・リード
ret
シリアル転送禁止関数
csib_end
CB0PWR = 0
CSIB0 動作禁止
CB0RXE = 0
CSIB 受信動作禁止
ret


INTCB0RE 割り込み
INTCB0RE 割り込み関数
csib_error
buf_rx[count_rx] = CB0RX
CB0RX レジスタ・リード
CB0OVE = 0
エラー・フラグ・クリア
reti
INTCB0R 割り込み
INTCB0R 割り込み関数
csib_int_receive
buf_rx[count_rx] = CB0RX
受信結果をバッファに格納
count_rx++
受信カウント・インクリメント
count_rx>=TX_SIZE-1
No
受信カウント判定
Yes
CB0SCE = 0
通信起動トリガ無効
reti

クロック同期式シリアル・インタフェース B（CSIBn）（n = 0, 1）
連続転送モード（スレーブ・モード，送受信モード）

【機　　能】	連続転送モードで，通信モードをスレーブ・モード，転送方向モードを MSB ファーストに設定し，送受信を各 10 回ずつ行います。 通信起動トリガを有効にし，通信クロックを外部クロックに，転送データ長を 8 ビットに設定します。
【関　数　名】	csib6_main
【引　き　数】	なし
【処 理 内 容】	送信回数カウント（count_tx）を初期値 0 に設定します。受信回数カウント（count_rx）を初期値 0 に設定し，各設定関数を呼び出したあと，送受信を開始します。
【使 用 S F R】	CB0REIC ：0x07（CSIB0 受信エラー割り込み要求信号（INTCB0RE）クリア，マスク解除，優先レベル 7 に設定） CB0RIC ：0x07（CSIB0 受信終了割り込み要求信号（INTCB0R）クリア，マスク解除，優先レベル 7 に設定） CB0TIC ：0x07（CSIB0 送信許可割り込み要求信号（INTCB0T）クリア，マスク解除，優先レベル 7 に設定） CB0STR.CB0TSF 通信状態フラグ
【call 関数】	csib_port, csib_set, csib_start, csib_end
【変　　数】	unsigned char buf_tx[ ] ：送信データ格納バッファ unsigned char buf_rx[：受信データ格納バッファ volatile unsigned char count_tx ：送信カウント変数 volatile unsigned char count_rx ：受信カウント変数 unsigned char count ：送信データ生成変数
【割 り 込 み】	csib_error, csib_int_send, csib_int_receive
【割り込み 要 因】	INTCB0RE, INTCB0T, INTCB0R
【ファイル名】	csib6.c
【注 意 事 項】	なし

【関　数　名】 csib_port

【処 理 内 容】 ポート 4 を CSIB0 入出力端子に設定します。

【使 用 S F R】 PMC4 ：0x07（$\overline{\text{SCKB0}}$ 入出力，SOB0 出力，SIB0 入力設定）

【call 関数】 なし

【変 数】 なし

【ファイル名】 csib6.c

【注 意 事 項】 なし

【関　数　名】 csib_set

【処 理 内 容】 CSIB0 制御レジスタの設定を行います。

【使 用 S F R】 CB0CTL1 ：0x07（通信タイプ 1，通信クロックを外部クロックに設定）
CB0CTL2 ：0x00（転送データ長を 8 ビットに設定）
CB0CTL0 ：0x63（CSIB0 送信動作許可，受信動作許可，MSB ファースト，連続転送モード，通信起動トリガ有効に設定）

【call 関数】 なし

【変 数】 なし

【ファイル名】 csib6.c

【注 意 事 項】 CB0CTL0 レジスタは CB0PWR ビット = 0 時のみ CB0TXE, CB0RXE ビットの書き換え可能です。ただし，同時に CB0PWR ビット = 1 とするのは可能です。

【関　数　名】 csib_start

【処 理 内 容】 CSIB0 動作許可し，送信データ・レジスタに値をライトします。

【使 用 S F R】 CB0CTL0.CB0PWR ：1（CSIB0 動作許可）
CB0TX 送信データ・レジスタ

【call 関数】 なし

【変 数】 unsigned char buf_tx[ ] ：送信データ格納バッファ
volatile unsigned char count_tx ：送信カウント変数

【ファイル名】 csib6.c

【注 意 事 項】 なし

【関　数　名】 csib_end

【処 理 内 容】 CSIB0 動作，送受信動作を禁止します。

【使 用S F R】 CB0CTL0.CB0PWR ：0（CSIB0 動作禁止）
CB0CTL0.CS0RXE ：0（CSIB0 受信動作禁止）
CB0CTL0.CB0TXE ：0（CSIB0 送信動作禁止）

【call 関数】 なし

【変 数】 なし

【ファイル名】 csib6.c

【注 意 事 項】 なし

割り込み関数

【関　数　名】 csib_error

【処 理 内 容】 受信エラー・フラグをクリアします。

【使 用S F R】 CB0RX 受信データ・レジスタ
CB0STR.CB0OVE ：0（オーバラン・エラー・フラグ・クリア）

【call 関数】 なし

【変 数】 unsigned char buf_rx[ ] ：受信データ格納バッファ
volatile unsigned char count_rx ：受信カウント変数

【ファイル名】 csib6.c

【注 意 事 項】 なし

【関　数　名】 csib_int_send

【処 理 内 容】 次のデータを送信するための新たなデータをセットします。

【使 用S F R】 CB0TX 送信データ・レジスタ

【call 関数】 なし

【変 数】 unsigned char buf_tx[ ] ：送信データ格納バッファ
volatile unsigned char count_tx ：送信カウント変数

【ファイル名】 csib6.c

【注 意 事 項】 なし

【関　数　名】 csib_int_receive

【処 理 内 容】 **受信したデータをバッファに格納します。**

【使 用S F R】 CB0RX **受信データ・レジスタ**

【call 関数】 **なし**

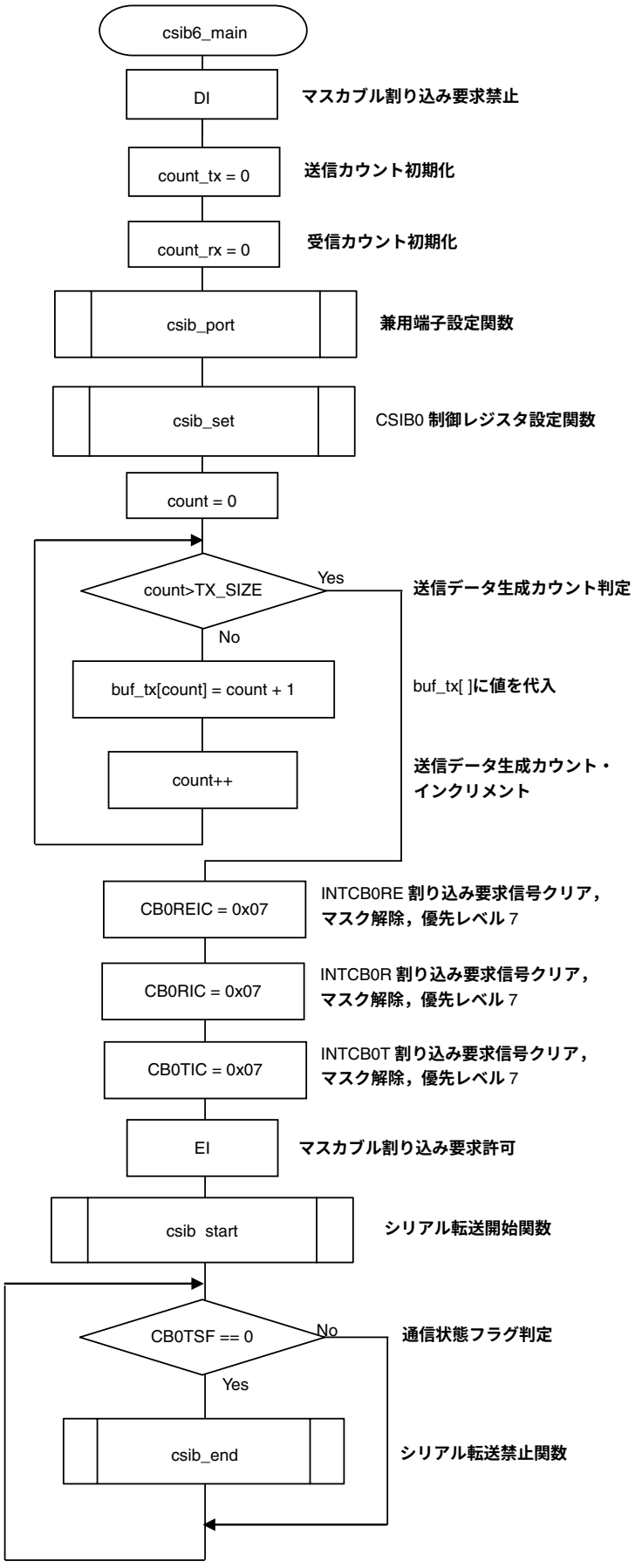
【変 数】 unsigned char buf_rx[ ] **：受信データ格納バッファ**

volatile unsigned char count_rx **：受信カウント変数**

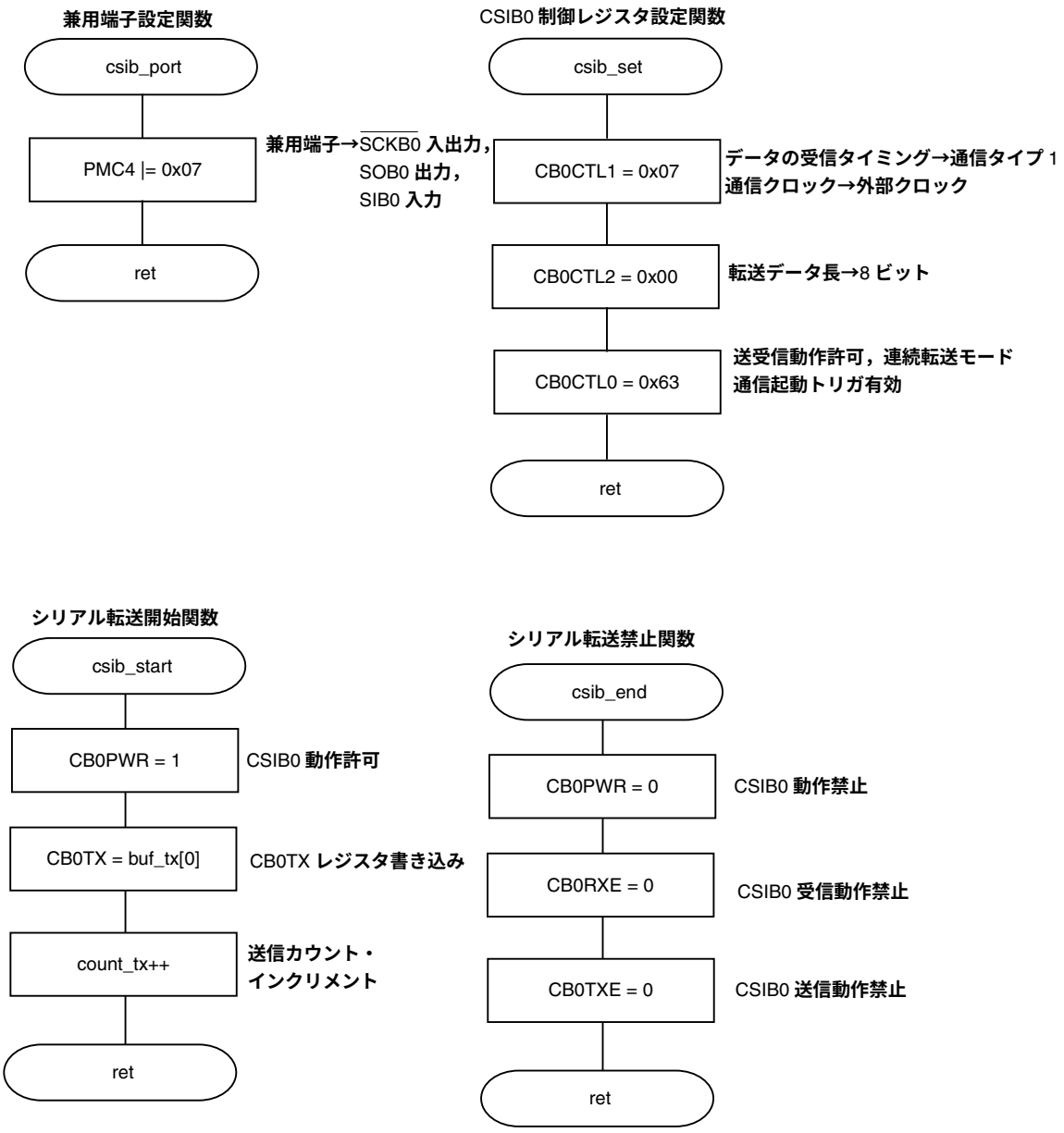
【フ ァ イ ル 名】 csib6.c

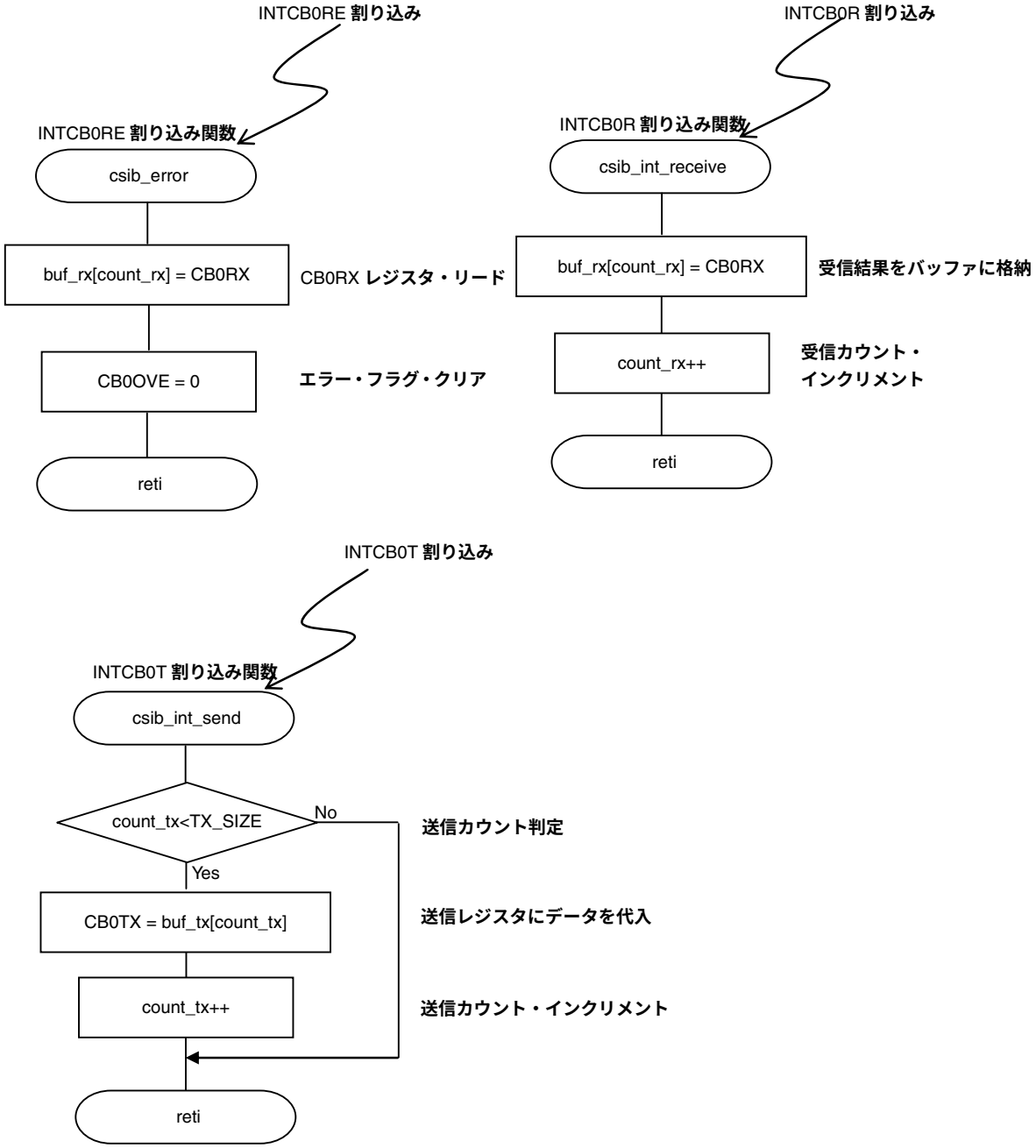
【注 意 事 項】 **なし**

連続転送モード（スレーブ・モード，送受信モード）（1/3）



連続転送モード（スレーブ・モード，送受信モード）（2/3）




INTCB0RE 割り込み
INTCB0RE 割り込み関数
csib_error
buf_rx[count_rx] = CB0RX
CB0RX レジスタ・リード
CB0OVE = 0
エラー・フラグ・クリア
reti
INTCB0R 割り込み
INTCB0R 割り込み関数
csib_int_receive
buf_rx[count_rx] = CB0RX
受信結果をバッファに格納
count_rx++
受信カウント・
インクリメント
reti
INTCB0T 割り込み
INTCB0T 割り込み関数
csib_int_send
count_tx<TX_SIZE
No
送信カウント判定
Yes
CB0TX = buf_tx[count_tx]
送信レジスタにデータを代入
count_tx++
送信カウント・インクリメント
reti

クロック同期式シリアル・インタフェース B（CSIBn）（n = 0, 1）
シングル転送モード（マスタ・モード，送信モード）

【機　　能】	シングル転送モードで，通信モードをマスタ・モード，転送方向モードを MSB ファーストに設定し，10 回のデータ送信を行います。 通信起動トリガを有効にし，通信クロック = fxx/256，転送データ長を 8 ビットに設定します。
【関 数 名】	csib7_main
【引 き 数】	なし
【処 理 内 容】	送信回数カウント（count_tx）を初期値 0 に設定します。各設定関数を呼び出したあと，送信を開始します。
【使 用 S F R】	CB0RIC　：0x07（CSIB0 受信終了割り込み要求信号（INTCB0R）クリア，マスク解除，優先レベル 7 に設定）
【call 関数】	csib_port, csib_set, csib_start, csib_end
【変 数】	unsigned char buf_tx[ ]　：送信データ格納バッファ volatile unsigned char count_tx　：送信カウント変数 unsigned char count　：送信データ生成変数
【割 り 込 み】	csib_int_send
【割り込み 要 因】	INTCB0R
【ファイル名】	csib7.c
【注 意 事 項】	なし

【関 数 名】	csib_port
【処 理 内 容】	ポート 4 を CSIB0 入出力端子に設定します。
【使 用 S F R】	PMC4　：0x06（$\overline{\text{SCKB0}}$ 入出力，SOB0 出力設定）
【call 関数】	なし
【変 数】	なし
【ファイル名】	csib7.c
【注 意 事 項】	なし

【関　数　名】 csib_set

【処 理 内 容】 CSIB0 制御レジスタの設定を行います。

【使 用S F R】 CB0CTL1 ：0x06（通信タイプ 1，通信クロック = fxx/256（0.25 MHz）に設定）
CB0CTL2 ：0x00（転送データ長を 8 ビットに設定）
CB0CTL0 ：0x41（CSIB0 送信動作許可，MSB ファースト，シングル転送モード，通信起動トリガ有効に設定）

【call 関数】 なし

【変 数】 なし

【ファイル名】 csib7.c

【注 意 事 項】 CB0CTL0 レジスタは CB0PWR ビット = 0 時のみ CB0TXE, CB0RXE ビットの書き換え可能です。ただし，同時に CB0PWR ビット = 1 とするのは可能です。

【関　数　名】 csib_start

【処 理 内 容】 CSIB0 動作許可し，送信データ・レジスタに値をライトします。

【使 用S F R】 CB0CTL0.CB0PWR ：1（CSIB0 動作許可）
CB0TX 送信データ・レジスタ

【call 関数】 なし

【変 数】 unsigned char buf_tx[ ] ：送信データ格納バッファ
volatile unsigned char count_tx ：送信カウント変数

【ファイル名】 csib7.c

【注 意 事 項】 なし

【関　数　名】 csib_end

【処 理 内 容】 CSIB0 動作，送信動作を禁止します。

【使 用S F R】 CB0CTL0.CB0PWR ：0（CSIB0 動作禁止）
CB0CTL0.CB0TXE ：0（CSIB0 送信動作禁止）

【call 関数】 なし

【変 数】 なし

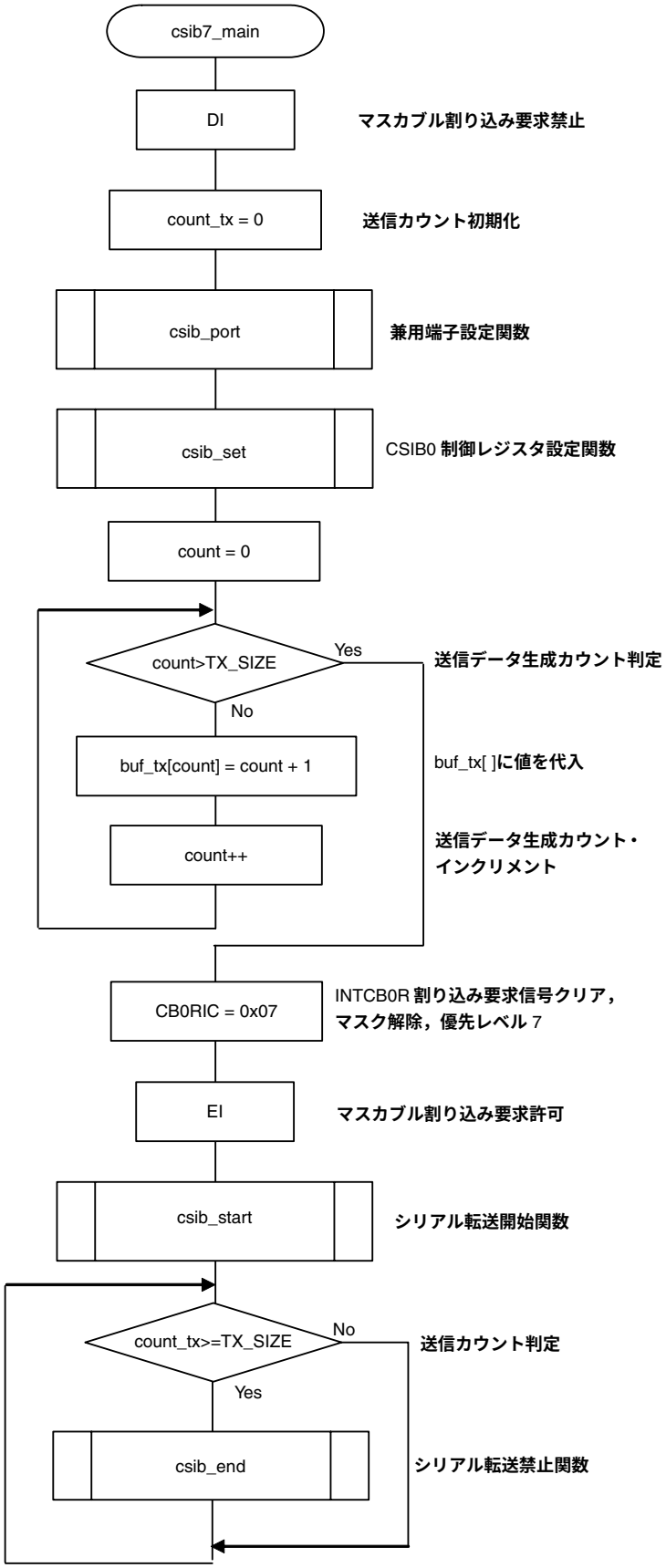
【ファイル名】 csib7.c

【注 意 事 項】 なし

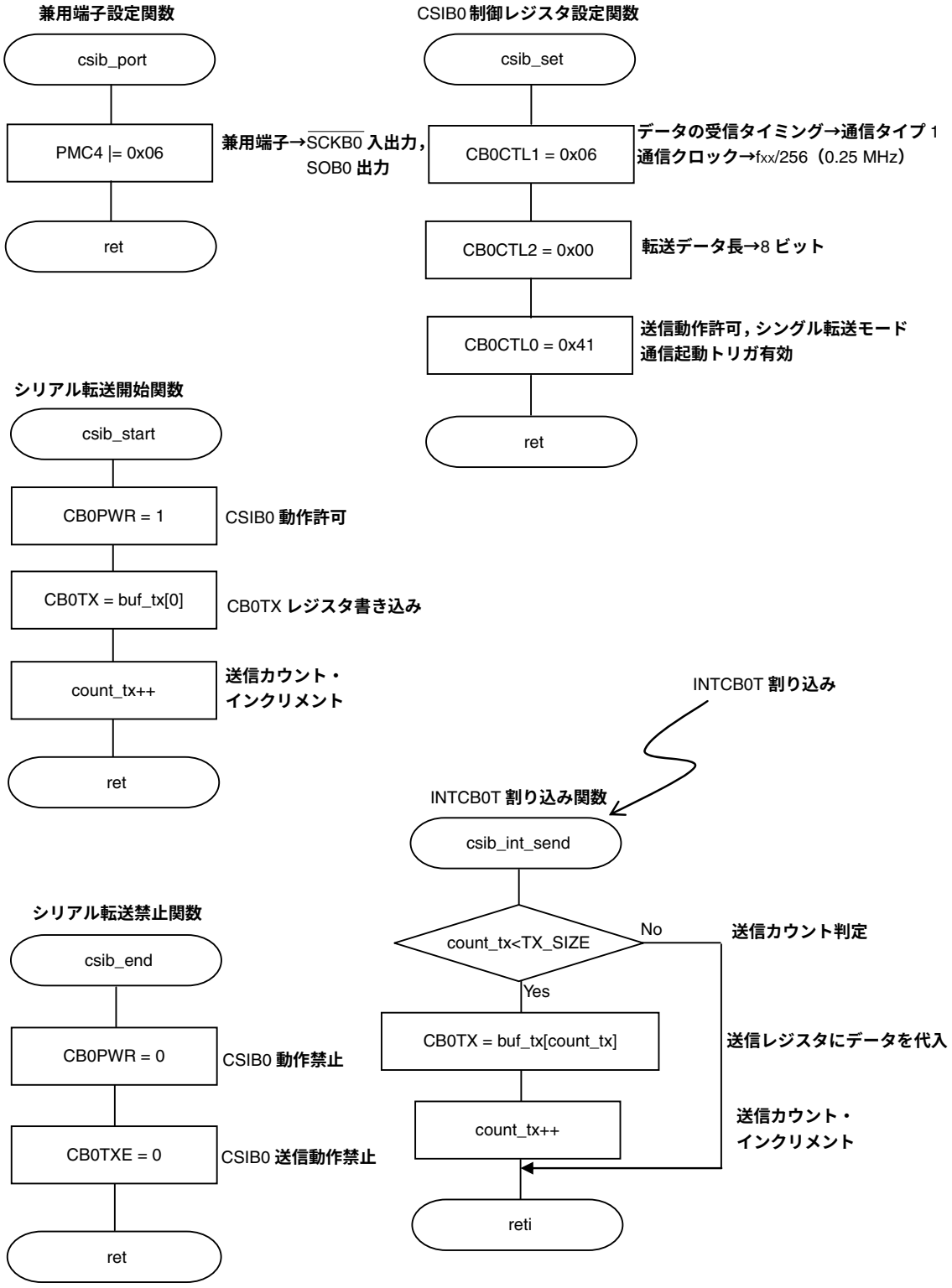
割り込み関数

【関　数　名】	csib_int_send
【処　理　内　容】	次のデータを送信するための新たなデータをセットします。
【使　用S　F　R】	CB0TX　　送信データ・レジスタ
【call　関数】	なし
【変　数】	unsigned char buf_tx[ ]　：送信データ格納バッファ volatile unsigned char count_tx　：送信カウント変数
【フ ァ イ ル 名】	csib7.c
【注　意　事　項】	なし

シングル転送モード（マスタ・モード，送信モード）（1/2）



シングル転送モード（マスタ・モード，送信モード）（2/2）



クロック同期式シリアル・インタフェース B（CSIBn）（n = 0, 1）
シングル転送モード（マスタ・モード，受信モード）

【機　　能】	シングル転送モードで，通信モードをマスタ・モード，転送方向モードを MSB ファーストに設定し，10 回のデータ受信を行います。 通信起動トリガを有効にし，通信クロック = fxx/256，転送データ長を 8 ビットに設定します。
【関　数　名】	csib8_main
【引　き　数】	なし
【処　理　内　容】	受信回数カウント（count_rx）を初期値 0 に設定します。各設定関数を呼び出したあと，受信を開始します。
【使　用 S　F　R】	CB0RIC　：0x07（CSIB0 受信終了割り込み要求信号（INTCB0R）クリア，マスク解除，優先レベル 7 に設定） CB0REIC　：0x07（CSIB0 受信エラー割り込み要求信号（INTCB0RE）クリア，マスク解除，優先レベル 7 に設定）
【call　　関数】	csib_port, csib_set, csib_start, csib_end
【変　　数】	unsigned char buf_rx[ ]　：受信データ格納バッファ volatile unsigned char count_rx　：受信カウント変数
【割　り　込　み】	csib_int_receive, csib_error
【割り込み 要 因】	INTCB0R, INTCB0RE
【フ ァ イ ル 名】	csib8.c
【注　意　事　項】	なし

【関　数　名】	csib_port
【処　理　内　容】	ポート 4 を CSIB0 入出力端子に設定します。
【使　用 S　F　R】	PMC4　：0x05（$\overline{\text{SCKB0}}$ 入出力，SIB0 入力設定）
【call　　関数】	なし
【変　　数】	なし
【フ ァ イ ル 名】	csib8.c
【注　意　事　項】	なし

【関　数　名】 csib_set

【処 理 内 容】 CSIB0 制御レジスタの設定を行います。

【使 用S F R】 CB0CTL1 ：0x06（通信タイプ 1，通信クロック = fxx/256（0.25 MHz）に設定）
CB0CTL2 ：0x00（転送データ長を 8 ビットに設定）
CB0CTL0 ：0x21（CSIB0 受信動作許可，MSB ファースト，シングル転送モード，通信起動トリガ有効に設定）

【call 関数】 なし

【変 数】 なし

【ファイル名】 csib8.c

【注 意 事 項】 CB0CTL0 レジスタは CB0PWR ビット = 0 時のみ CB0TXE, CB0RXE ビットの書き換え可能です。ただし，同時に CB0PWR ビット = 1 とするのは可能です。

【関　数　名】 csib_start

【処 理 内 容】 CSIB0 動作許可し，受信データ・レジスタをダミー・リードします。

【使 用S F R】 CB0CTL0.CB0PWR ：1（CSIB0 動作許可）
CB0RX 受信データ・レジスタ

【call 関数】 なし

【変 数】 unsigned char buf_rx[ ] ：受信データ格納バッファ

【ファイル名】 csib8.c

【注 意 事 項】 なし

【関　数　名】 csib_end

【処 理 内 容】 CSIB0 動作，受信動作を禁止します。

【使 用S F R】 CB0CTL0.CB0PWR ：0（CSIB0 動作禁止）
CB0CTL0.CB0RXE ：0（CSIB0 受信動作禁止）

【call 関数】 なし

【変 数】 なし

【ファイル名】 csib8.c

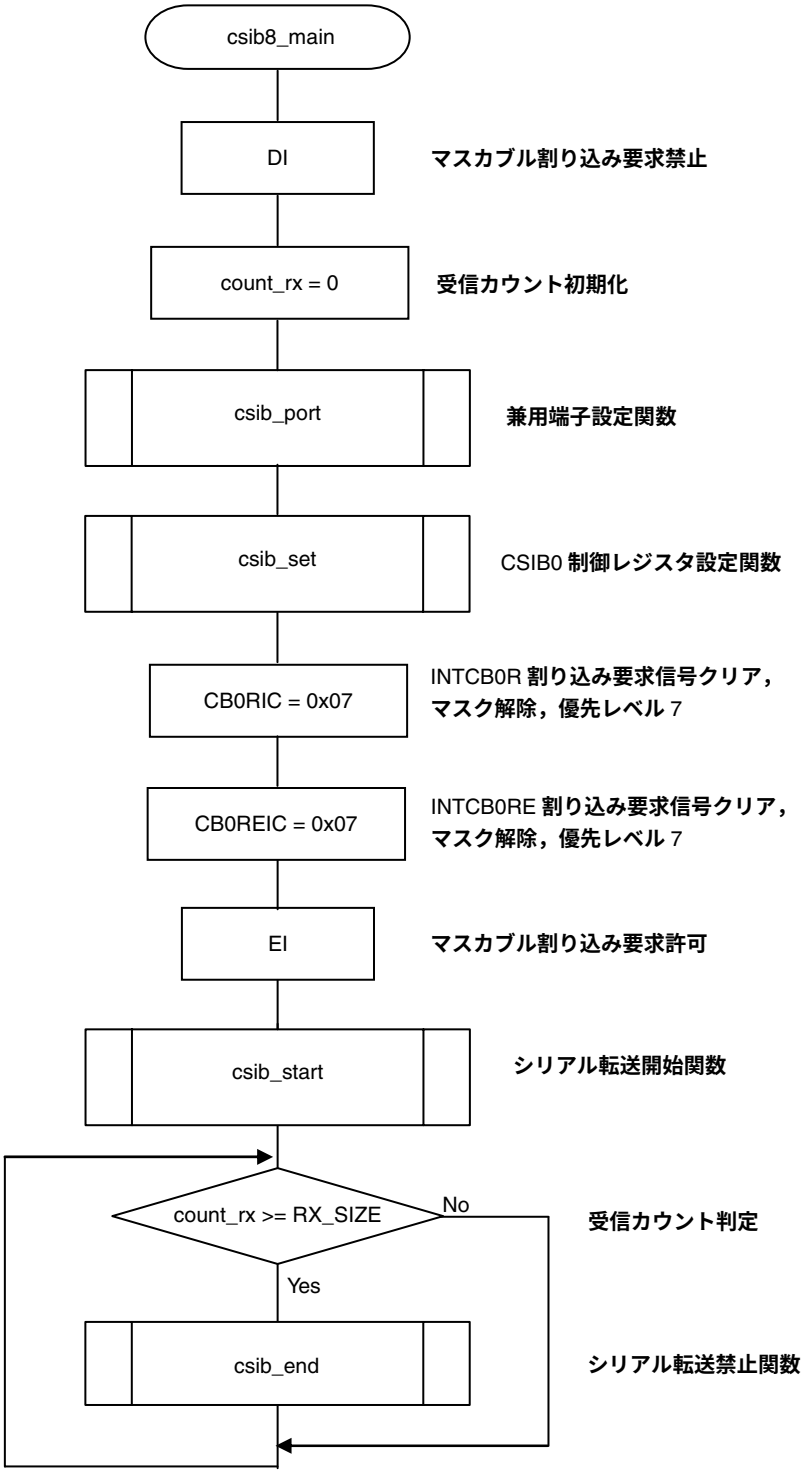
【注 意 事 項】 なし

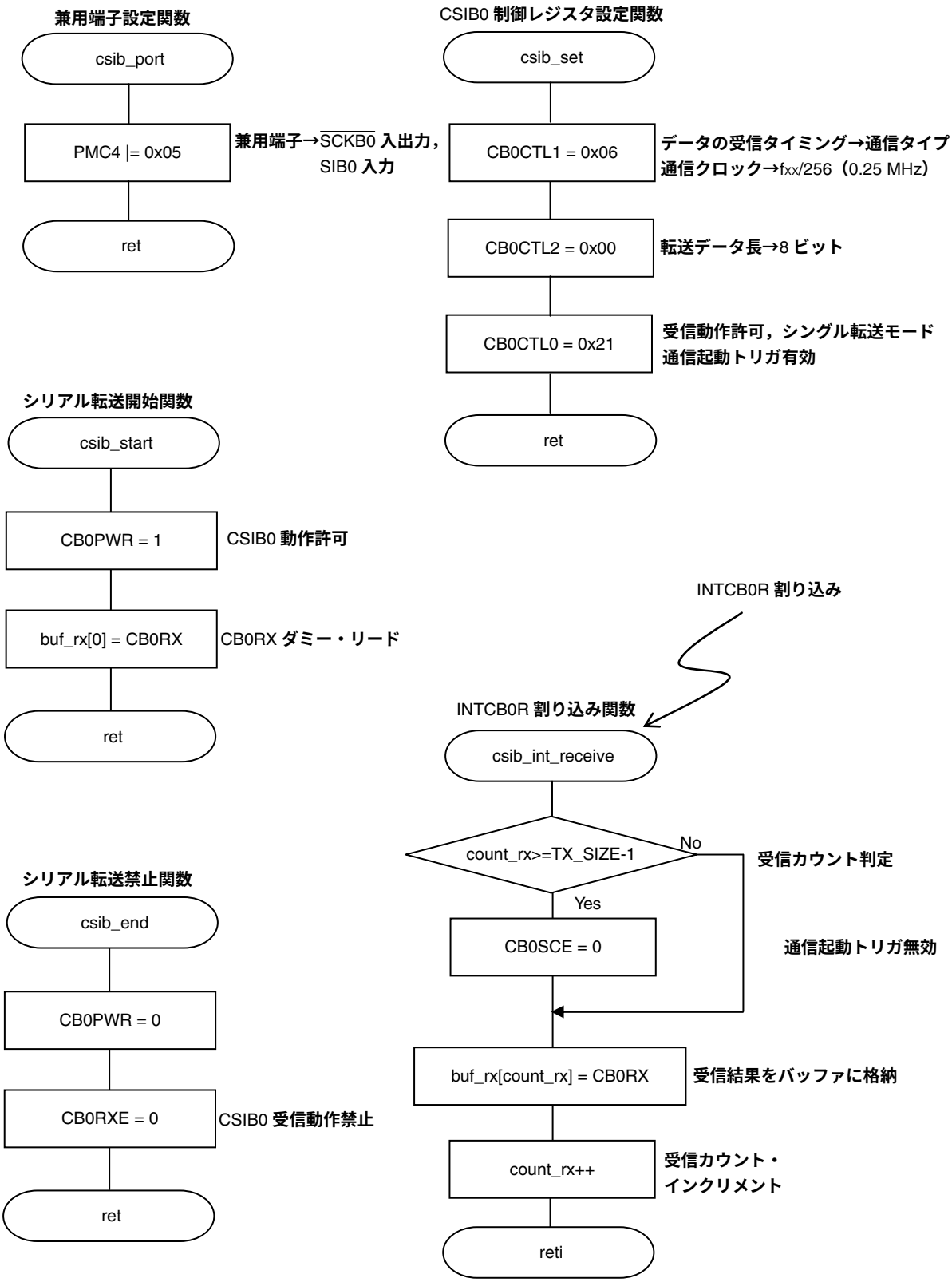
割り込み関数

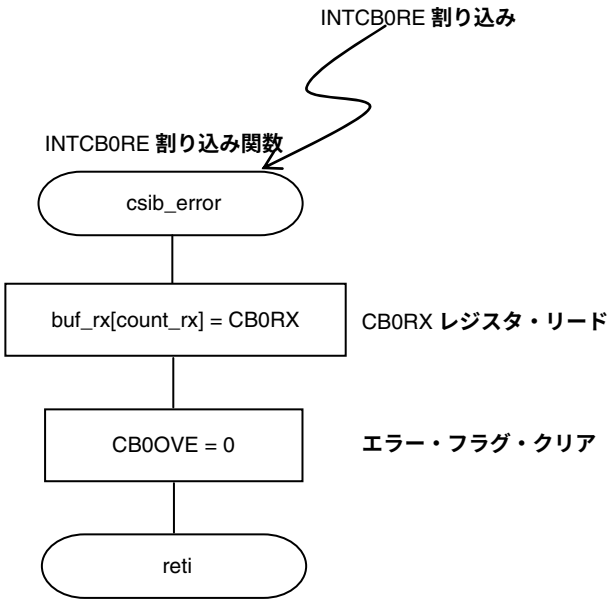
【関　数　名】	csib_int_receive
【処　理　内　容】	受信したデータをバッファに格納します。
【使　用S　F　R】	CB0RX　　受信データ・レジスタ CB0CTL0.CB0SCE　：0（通信起動トリガ無効）
【call　関数】	なし
【変　数】	unsigned char buf_rx[ ]　：受信データ格納バッファ volatile unsigned char count_rx　：受信カウント変数
【ファイル名】	csib8.c
【注　意　事　項】	なし

【関　数　名】	csib_error
【処　理　内　容】	受信エラー・フラグをクリアします。
【使　用S　F　R】	CB0RX　　受信データ・レジスタ CB0STR.CB0OVE　：0（オーバラン・エラー・フラグ・クリア）
【call　関数】	なし
【変　数】	unsigned char buf_rx[ ]　：受信データ格納バッファ volatile unsigned char count_rx　：受信カウント変数
【ファイル名】	csib8.c
【注　意　事　項】	なし

シングル転送モード（マスタ・モード，受信モード）（1/3）




兼用端子設定関数
csib_port
PMC4 |= 0x05
兼用端子→SCKB0 入出力，
SIB0 入力
ret
CSIB0 制御レジスタ設定関数
csib_set
CB0CTL1 = 0x06
データの受信タイミング→通信タイプ 1
通信クロック→fXX/256（0.25 MHz）
CB0CTL2 = 0x00
転送データ長→8 ビット
CB0CTL0 = 0x21
受信動作許可，シングル転送モード
通信起動トリガ有効
ret
シリアル転送開始関数
csib_start
CB0PWR = 1
CSIB0 動作許可
buf_rx[0] = CB0RX
CB0RX ダミー・リード
ret
INTCB0R 割り込み
INTCB0R 割り込み関数
csib_int_receive
count_rx>=TX_SIZE-1
No
受信カウント判定
Yes
CB0SCE = 0
通信起動トリガ無効
buf_rx[count_rx] = CB0RX
受信結果をバッファに格納
count_rx++
受信カウント・
インクリメント
reti
シリアル転送禁止関数
csib_end
CB0PWR = 0
CB0RXE = 0
CSIB0 受信動作禁止
ret


INTCB0RE 割り込み
INTCB0RE 割り込み関数
csib_error
buf_rx[count_rx] = CB0RX
CB0RX レジスタ・リード
CB0OVE = 0
エラー・フラグ・クリア
reti

クロック同期式シリアル・インタフェース B（CSIBn）（n = 0, 1）
シングル転送モード（マスタ・モード，送受信モード）

【機　　能】	シングル転送モードで，通信モードをマスタ・モード，転送方向モードを MSB ファーストに設定し，送受信を各 10 回行います。 通信起動トリガを有効にし，通信クロック = fxx/256，転送データ長を 8 ビットに設定します。
【関　数　名】	csib9_main
【引　き　数】	なし
【処 理 内 容】	送信回数カウント（count_tx）を初期値 0 に設定します。受信回数カウント（count_rx）を初期値 0 に設定し，各設定関数を呼び出したあと，送受信を開始します。
【使 用 S F R】	CB0REIC ：0x07（CSIB0 受信エラー割り込み要求信号（INTCB0RE）クリア，マスク解除，優先レベル 7 に設定） CB0RIC ：0x07（CSIB0 受信終了割り込み要求信号（INTCB0R）クリア，マスク解除，優先レベル 7 に設定） CB0TIC ：0x07（CSIB0 送信許可割り込み要求信号（INTCB0T）クリア，マスク解除，優先レベル 7 に設定）
【call　　関数】	csib_port, csib_set, csib_start, csib_end
【変　　数】	unsigned char buf_tx[ ] ：送信データ格納バッファ unsigned char buf_rx[ ] ：受信データ格納バッファ volatile unsigned char count_tx ：送信カウント変数 volatile unsigned char count_rx ：受信カウント変数 unsigned char count ：送信データ生成変数
【割 り 込 み】	csib_error, csib_int_send, csib_int_receive
【割り込み 要 因】	INTCB0RE, INTCB0T, INTCB0R
【ファイル名】	csib9.c
【注 意 事 項】	なし

【関　数　名】	csib_port
【処 理 内 容】	ポート 4 を CSIB0 入出力端子に設定します。
【使 用 S F R】	PMC4 ：0x07（$\overline{\text{SCKB0}}$ 入出力，SOB0 出力，SIB0 入力設定）
【call　　関数】	なし
【変　　数】	なし
【ファイル名】	csib9.c
【注 意 事 項】	なし

【関　数　名】 csib_set

【処 理 内 容】 CSIB0 制御レジスタの設定を行います。

【使 用S F R】 CB0CTL1 ：0x06（通信タイプ 1，通信クロック = fxx/256（0.25 MHz）に設定）
CB0CTL2 ：0x00（転送データ長を 8 ビットに設定）
CB0CTL0 ：0x61（CSIB0 送信動作許可，受信動作許可，MSB ファースト，
シングル転送モード，通信起動トリガ有効に設定）

【call 関数】 なし

【変 数】 なし

【ファイル名】 csib9.c

【注 意 事 項】 CB0CTL0 レジスタは CB0PWR ビット = 0 時のみ CB0TXE, CB0RXE ビットの書き換え可能です。ただし，同時に CB0PWR ビット = 1 とするのは可能です。

【関　数　名】 csib_start

【処 理 内 容】 CSIB0 動作許可し，送信データ・レジスタに値をライトします。

【使 用S F R】 CB0CTL0.CB0PWR ：1（CSIB0 動作許可）
CB0TX 送信データ・レジスタ

【call 関数】 なし

【変 数】 unsigned char buf_tx[ ] ：送信データ格納バッファ
volatile unsigned char count_tx ：送信カウント変数

【ファイル名】 csib9.c

【注 意 事 項】 なし

【関　数　名】 csib_end

【処 理 内 容】 CSIB0 動作，送受信動作を禁止します。

【使 用S F R】 CB0CTL0.CB0PWR ：0（CSIB0 動作禁止）
CB0CTL0.CB0RXE ：0（CSIB0 受信動作禁止）
CB0CTL0.CB0TXE ：0（CSIB0 送信動作禁止）

【call 関数】 なし

【変 数】 なし

【ファイル名】 csib9.c

【注 意 事 項】 なし

割り込み関数

【関　数　名】 csib_error

【処 理 内 容】 受信エラー・フラグをクリアします。

【使 用S F R】 CB0RX　　送信データ・レジスタ
CB0STR.CB0OVE　：0（オーバラン・エラー・フラグ・クリア）

【call　　関数】 なし

【変　　数】 unsigned char buf_rx[ ]　　：受信データ格納バッファ
volatile unsigned char count_rx　：受信カウント変数

【フ ァ イ ル 名】 csib9.c

【注 意 事 項】 なし

【関　数　名】 csib_int_send

【処 理 内 容】 次のデータを送信するための新たなデータをセットします。

【使 用S F R】 CB0TX　　送信データ・レジスタ

【call　　関数】 なし

【変　　数】 unsigned char buf_tx[ ]　　：送信データ格納バッファ
volatile unsigned char count_tx　：送信カウント変数

【フ ァ イ ル 名】 csib9.c

【注 意 事 項】 なし

【関　数　名】 csib_int_receive

【処 理 内 容】 受信したデータをバッファに格納します。

【使 用S F R】 CB0RX　　受信データ・レジスタ

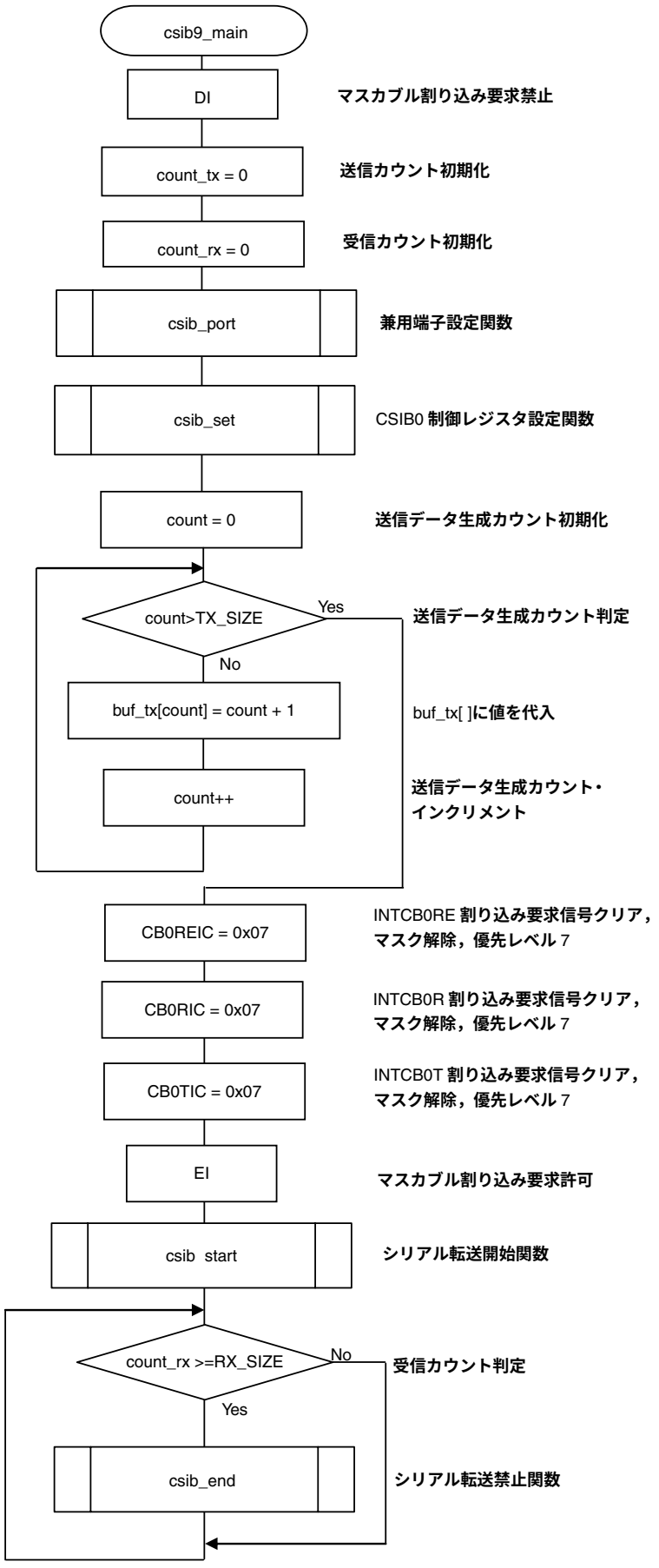
【call　　関数】 なし

【変　　数】 unsigned char buf_rx[ ]　　：受信データ格納バッファ
volatile unsigned char count_rx　：受信カウント変数

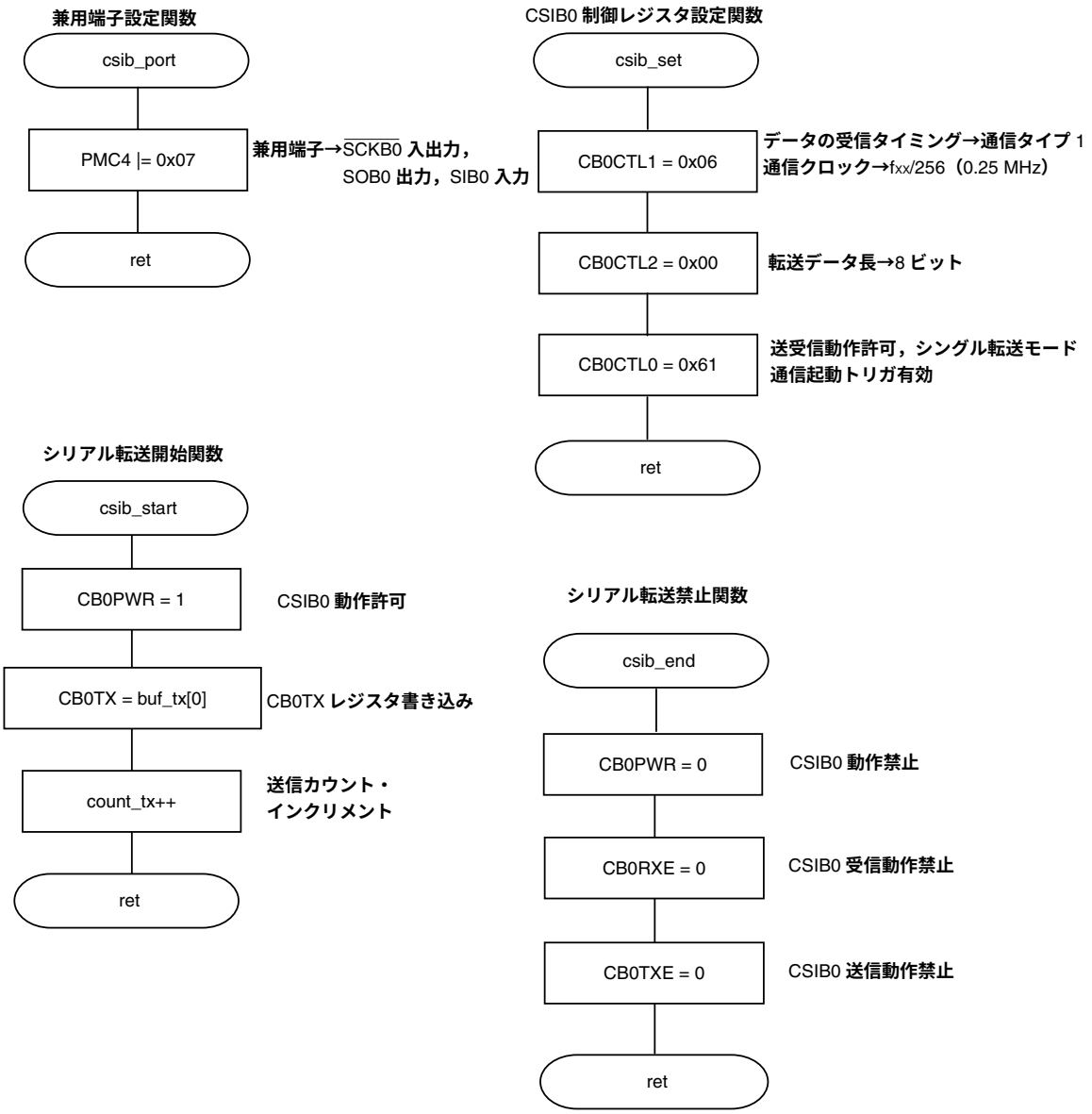
【フ ァ イ ル 名】 csib9.c

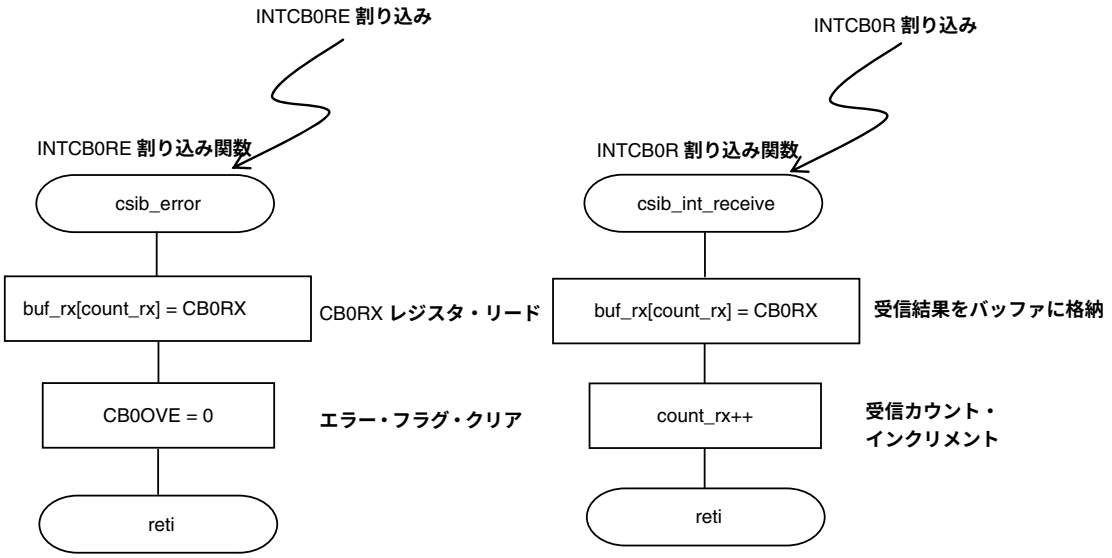
【注 意 事 項】 なし

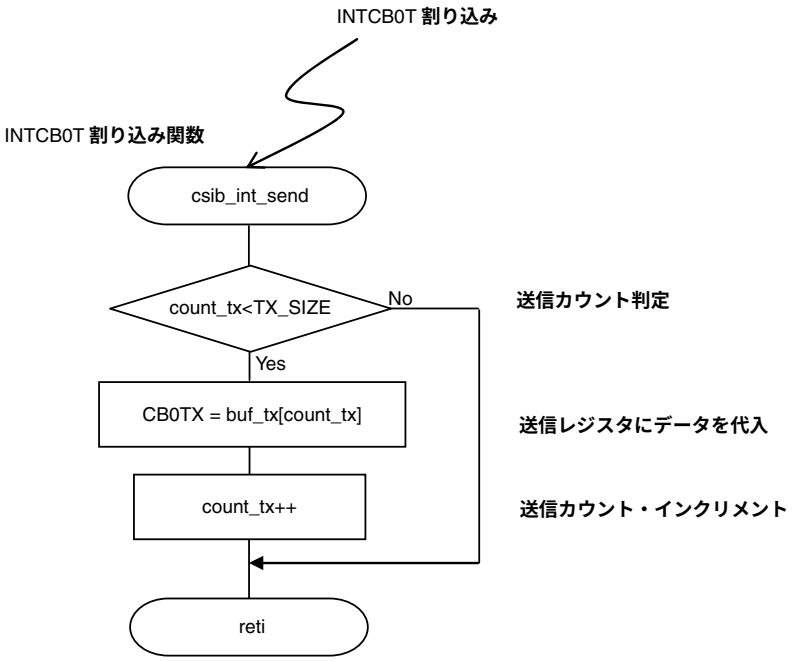
シングル転送モード（マスタ・モード，送受信モード）（1/3）



シングル転送モード（マスタ・モード，送受信モード）（2/3）




INTCB0RE 割り込み
INTCB0RE 割り込み関数
csib_error
buf_rx[count_rx] = CB0RX
CB0RX レジスタ・リード
CB0OVE = 0
エラー・フラグ・クリア
reti
INTCB0R 割り込み
INTCB0R 割り込み関数
csib_int_receive
buf_rx[count_rx] = CB0RX
受信結果をバッファに格納
count_rx++
受信カウント・
インクリメント
reti


INTCB0T 割り込み
INTCB0T 割り込み関数
csib_int_send
count_tx<TX_SIZE
No
送信カウント判定
Yes
CB0TX = buf_tx[count_tx]
送信レジスタにデータを代入
count_tx++
送信カウント・インクリメント
reti

クロック同期式シリアル・インタフェース B（CSIBn）（n = 0, 1）
シングル転送モード（スレーブ・モード，送信モード）

【機　　能】	シングル転送モードで，通信モードをスレーブ・モード，転送方向モードを MSB ファーストに設定し，10 回のデータ送信を行います。 通信起動トリガを有効にし，通信クロックは外部クロック，転送データ長を 8 ビットに設定します。
【関 数 名】	csib10_main
【引 き 数】	なし
【処 理 内 容】	送信回数カウント（count_tx）を初期値 0 に設定します。各設定関数を呼び出したあと，送信を開始します。
【使 用 S F R】	CB0RIC　：0x07（CSIB0 受信終了割り込み要求信号（INTCB0R）クリア，マスク解除，優先レベル 7 に設定）
【call 関数】	csib_port, csib_set, csib_start, csib_end
【変 数】	unsigned char buf_tx[ ]　：送信データ格納バッファ volatile unsigned char count_tx　：送信カウント変数 unsigned char count　：送信データ生成変数
【割 り 込 み】	csib_int_send
【割り込み 要 因】	INTCB0R
【フ ァ イ ル 名】	csib10.c
【注 意 事 項】	なし

【関 数 名】	csib_port
【処 理 内 容】	ポート 4 を CSIB0 入出力端子に設定します。
【使 用 S F R】	PMC4　：0x06（$\overline{\text{SCKB0}}$ 入出力，SOB0 出力設定）
【call 関数】	なし
【変 数】	なし
【フ ァ イ ル 名】	csib10.c
【注 意 事 項】	なし

【関　数　名】 csib_set

【処　理　内　容】 CSIB0 制御レジスタの設定を行います。

【使　用 S　F　R】 CB0CTL1　：0x07（通信タイプ 1，外部クロックに設定）
CB0CTL2　：0x00（転送データ長を 8 ビットに設定）
CB0CTL0　：0x41（CSIB0 送信動作許可，MSB ファースト，シングル転送モード，通信起動トリガ有効に設定）

【call　関数】 なし

【変　数】 なし

【ファイル名】 csib10.c

【注　意　事　項】 CB0CTL0 レジスタは CB0PWR ビット = 0 時のみ CB0TXE, CB0RXE ビットの書き換え可能です。ただし，同時に CB0PWR ビット = 1 とするのは可能です。

【関　数　名】 csib_start

【処　理　内　容】 CSIB0 動作許可し，送信データ・レジスタに値をライトします。

【使　用 S　F　R】 CB0CTL0.CB0PWR ：1（CSIB0 動作許可）
CB0TX　送信データ・レジスタ

【call　関数】 なし

【変　数】 unsigned char buf_tx[ ]　：送信データ格納バッファ
volatile unsigned char count_tx　：送信カウント変数

【ファイル名】 csib10.c

【注　意　事　項】 なし

【関　数　名】 csib_end

【処　理　内　容】 CSIB0 動作，送信動作を禁止します。

【使　用 S　F　R】 CB0CTL0.CB0PWR ：0（CSIB0 動作禁止）
CB0CTL0.CB0TXE　：0（CSIB0 送信動作禁止）

【call　関数】 なし

【変　数】 なし

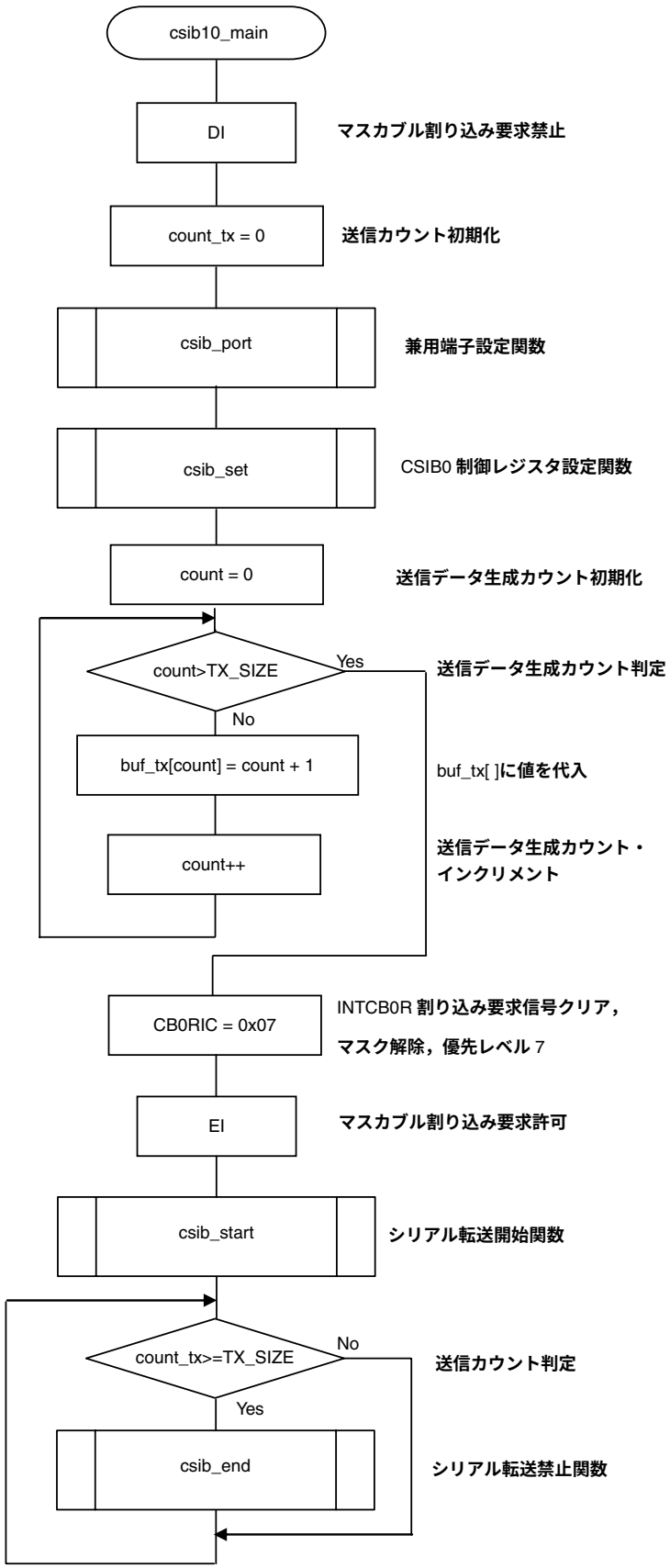
【ファイル名】 csib10.c

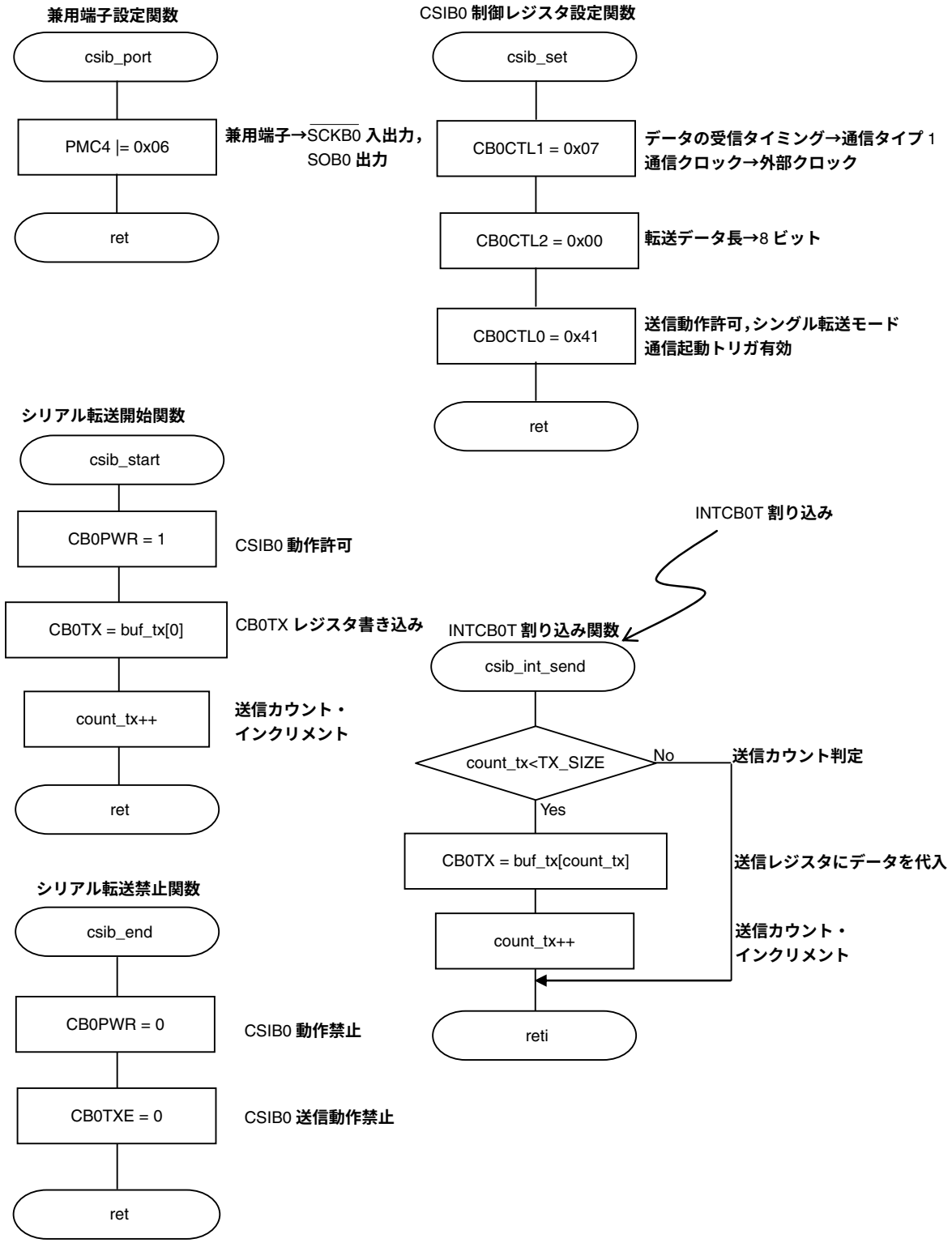
【注　意　事　項】 なし

割り込み関数

【関　数　名】	csib_int_send
【処　理　内　容】	**次のデータを送信するための新たなデータをセットします。**
【使　用S　F　R】	CB0TX　　**送信データ・レジスタ**
【call　関数】	**なし**
【変　数】	unsigned char buf_tx[ ]　**：送信データ格納バッファ** volatile unsigned char count_tx　**：送信カウント変数**
【フ ァ イ ル 名】	csib10.c
【注　意　事　項】	**なし**

シングル転送モード（スレーブ・モード，送信モード）（1/2）




兼用端子設定関数
csib_port
PMC4 |= 0x06
兼用端子→SCKB0 入出力，
SOB0 出力
ret
CSIB0 制御レジスタ設定関数
csib_set
CB0CTL1 = 0x07
データの受信タイミング→通信タイプ 1
通信クロック→外部クロック
CB0CTL2 = 0x00
転送データ長→8 ビット
CB0CTL0 = 0x41
送信動作許可，シングル転送モード
通信起動トリガ有効
ret
シリアル転送開始関数
csib_start
CB0PWR = 1
CSIB0 動作許可
CB0TX = buf_tx[0]
CB0TX レジスタ書き込み
count_tx++
送信カウント・
インクリメント
ret
INTCB0T 割り込み
INTCB0T 割り込み関数
csib_int_send
count_tx<TX_SIZE
No
送信カウント判定
Yes
CB0TX = buf_tx[count_tx]
送信レジスタにデータを代入
count_tx++
送信カウント・
インクリメント
reti
シリアル転送禁止関数
csib_end
CB0PWR = 0
CSIB0 動作禁止
CB0TXE = 0
CSIB0 送信動作禁止
ret

クロック同期式シリアル・インタフェース B（CSIBn）（n = 0, 1）
シングル転送モード（スレーブ・モード，受信モード）

【機　　能】	シングル転送モードで，通信モードをスレーブ・モード，転送方向モードを MSB ファーストに設定し，10 回のデータ受信を行います。 通信起動トリガを有効にし，通信クロックは外部クロック，転送データ長を 8 ビットに設定します。
【関 数 名】	csib11_main
【引 き 数】	なし
【処 理 内 容】	受信回数カウント（count_rx）を初期値 0 に設定します。各設定関数を呼び出したあと，受信を開始します。
【使 用 S F R】	CB0RIC ：0x07（CSIB0 受信終了割り込み要求信号（INTCB0R）クリア，マスク解除，優先レベル 7 に設定） CB0REIC ：0x07（CSIB0 受信エラー割り込み要求信号（INTCB0RE）クリア，マスク解除，優先レベル 7 に設定）
【call 関数】	csib_port, csib_set, csib_start, csib_end
【変　　数】	unsigned char buf_rx[ ] ：受信データ格納バッファ volatile unsigned char count_rx ：受信カウント変数
【割 り 込 み】	csib_int_receive, csib_error
【割り込み 要 因】	INTCB0R, INTCB0RE
【フ ァ イ ル 名】	csib11.c
【注 意 事 項】	なし

【関 数 名】	csib_port
【処 理 内 容】	ポート 4 を CSIB0 入出力端子に設定します。
【使 用 S F R】	PMC4 ：0x05（$\overline{\text{SCKB0}}$ 入出力，SIB0 入力設定）
【call 関数】	なし
【変　　数】	なし
【フ ァ イ ル 名】	csib11.c
【注 意 事 項】	なし

【関　数　名】 csib_set

【処 理 内 容】 CSIB0 制御レジスタの設定を行います。

【使 用S F R】 CB0CTL1 ：0x07（通信タイプ 1，外部クロックに設定）
CB0CTL2 ：0x00（転送データ長を 8 ビットに設定）
CB0CTL0 ：0x21（CSIB 受信動作許可，MSB ファースト，シングル転送モード，通信起動トリガ有効に設定）

【call 関数】 なし

【変 数】 なし

【ファイル名】 csib11.c

【注 意 事 項】 CB0CTL0 レジスタは CB0PWR ビット = 0 時のみ CB0TXE, CB0RXE ビットの書き換え可能です。ただし，同時に CB0PWR ビット = 1 とするのは可能です。

【関　数　名】 csib_start

【処 理 内 容】 CSIB0 動作許可し，受信データ・レジスタをダミー・リードします。

【使 用S F R】 CB0CTL0.CB0PWR ：1（CSIB0 動作許可）
CB0RX 受信データ・レジスタ

【call 関数】 なし

【変 数】 unsigned char buf_rx[ ] ：受信データ格納バッファ

【ファイル名】 csib11.c

【注 意 事 項】 なし

【関　数　名】 csib_end

【処 理 内 容】 CSIB0 動作，受信動作を禁止します。

【使 用S F R】 CB0CTL0.CB0PWR ：0（CSIB0 動作禁止）
CB0CTL0.CB0RXE ：0（CSIB0 受信動作禁止）

【call 関数】 なし

【変 数】 なし

【ファイル名】 csib11.c

【注 意 事 項】 なし

割り込み関数

【関　数　名】 csib_int_receive

【処　理　内　容】 受信したデータをバッファに格納します。

【使　用S　F　R】 CB0RX　　受信データ・レジスタ
CB0CTL0.CB0SCE　：0（通信起動トリガ無効）

【call　　関数】 なし

【変　　数】 unsigned char buf_rx[ ]　　：受信データ格納バッファ
volatile unsigned char count_rx　：受信カウント変数

【フ ァ イ ル 名】 csib11.c

【注　意　事　項】 なし

【関　数　名】 csib_error

【処　理　内　容】 受信エラー・フラグをクリアします。

【使　用S　F　R】 CB0RX　　受信データ・レジスタ
CB0STR.CB0OVE　：0（オーバラン・エラー・フラグ・クリア）

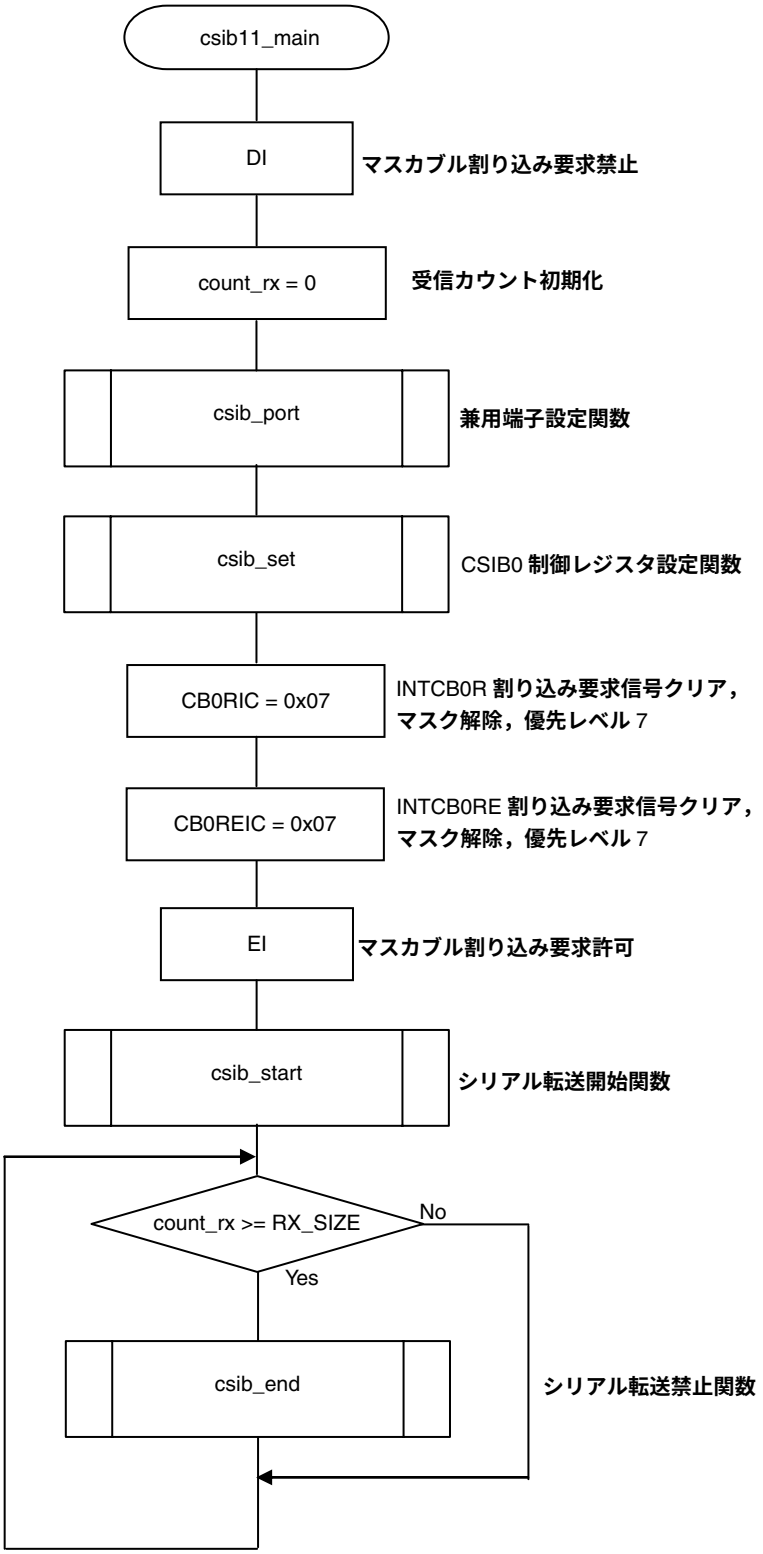
【call　　関数】 なし

【変　　数】 unsigned char buf_rx[ ]　　：受信データ格納バッファ
volatile unsigned char count_rx　：受信カウント変数

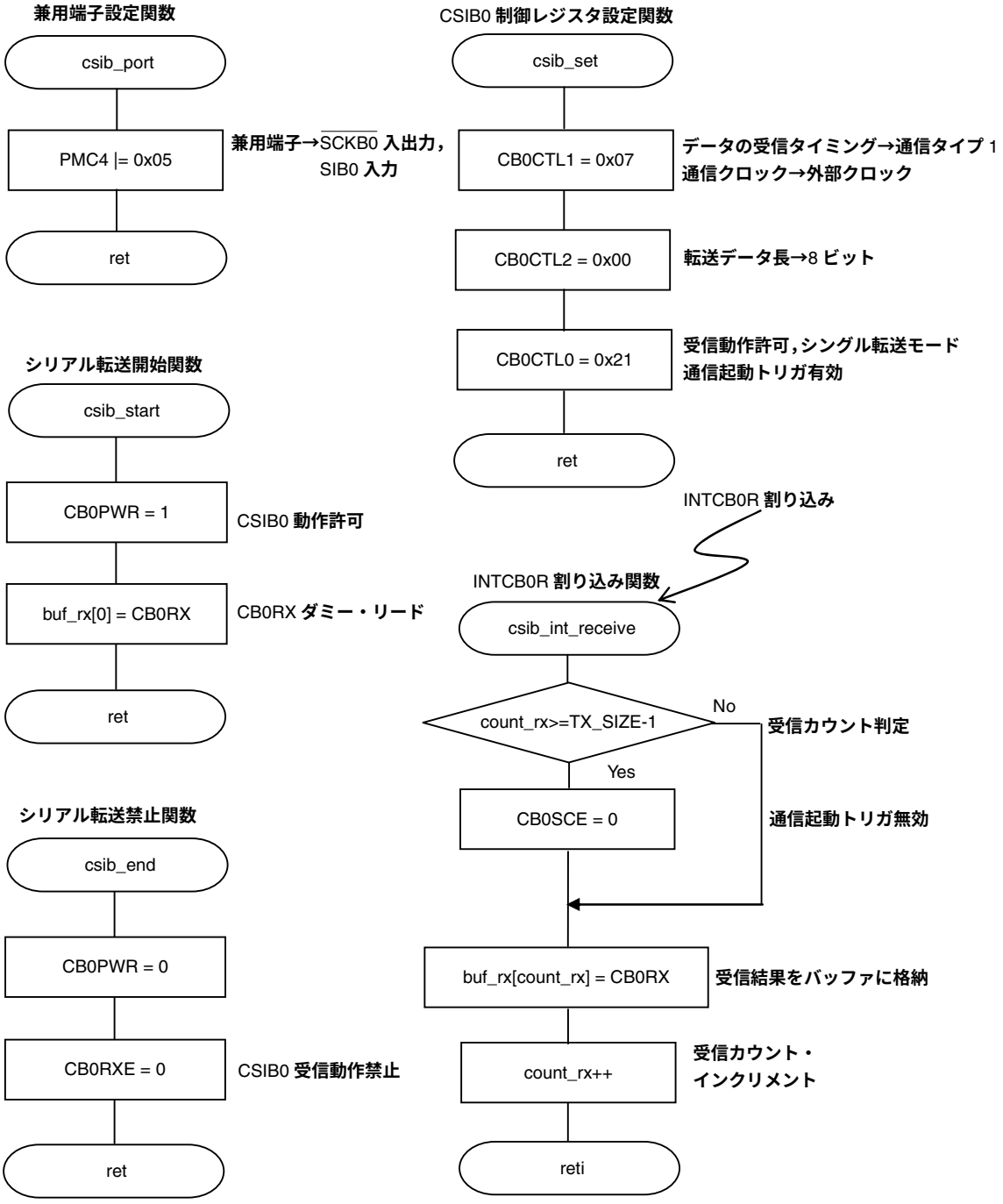
【フ ァ イ ル 名】 csib11.c

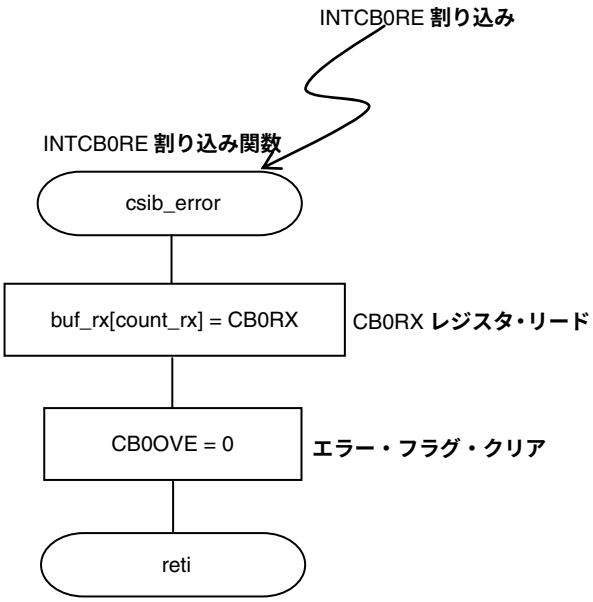
【注　意　事　項】 なし

シングル転送モード（スレーブ・モード，受信モード）（1/3）



シングル転送モード（スレーブ・モード，受信モード）（2/3）




INTCB0RE 割り込み
INTCB0RE 割り込み関数
csib_error
buf_rx[count_rx] = CB0RX
CB0RX レジスタ・リード
CB0OVE = 0
エラー・フラグ・クリア
reti

クロック同期式シリアル・インタフェース B（CSIBn）（n = 0, 1）
シングル転送モード（スレーブ・モード，送受信モード）

【機　　能】	シングル転送モードで，通信モードをスレーブ・モード，転送方向モードを MSB ファーストに設定し，送受信を各 10 回ずつ行います。 通信起動トリガを有効にし，通信クロックは外部クロック，転送データ長を 8 ビットに設定します。
【関　数　名】	csib12_main
【引　き　数】	なし
【処 理 内 容】	送信回数カウント（count_tx）を初期値 0 に設定します。受信回数カウント（count_rx）を初期値 0 に設定し，各設定関数を呼び出したあと，送受信を開始します。
【使 用 S F R】	CB0REIC ：0x07（CSIB0 受信エラー割り込み要求信号（INTCB0RE）クリア，マスク解除，優先レベル 7 に設定） CB0RIC ：0x07（CSIB0 受信終了割り込み要求信号（INTCB0R）クリア，マスク解除，優先レベル 7 に設定） CB0TIC ：0x07（CSIB0 送信許可割り込み要求信号（INTCB0T）クリア，マスク解除，優先レベル 7 に設定）
【call　関数】	csib_port, csib_set, csib_start, csib_end
【変　　数】	unsigned char buf_tx[ ] ：送信データ格納バッファ unsigned char buf_rx[ ] ：受信データ格納バッファ volatile unsigned char count_tx ：送信カウント変数 volatile unsigned char count_rx ：受信カウント変数 unsigned char count ：送信データ生成変数
【割 り 込 み】	csib_error, csib_int_send, csib_int_receive
【割り込み 要 因】	INTCB0RE, INTCB0T, INTCB0R
【ファイル名】	csib12.c
【注 意 事 項】	なし

【関　数　名】 csib_port

【処 理 内 容】 ポート 4 を CSIB0 入出力端子に設定します。

【使 用 S F R】 PMC4 ：0x07（$\overline{\text{SCKB0}}$ 入出力，SOB0 出力，SIB0 入力設定）

【call 関数】 なし

【変 数】 なし

【ファイル名】 csib12.c

【注 意 事 項】 なし

【関　数　名】 csib_set

【処 理 内 容】 CSIB0 制御レジスタの設定を行います。

【使 用 S F R】 CB0CTL1 ：0x07（通信タイプ 1，外部クロックに設定）
CB0CTL2 ：0x00（転送データ長を 8 ビットに設定）
CB0CTL0 ：0x61（CSIB0 送信動作許可，受信動作許可，MSB ファースト，シングル転送モード，通信起動トリガ有効に設定）

【call 関数】 なし

【変 数】 なし

【ファイル名】 csib12.c

【注 意 事 項】 CB0CTL0 レジスタは CB0PWR ビット = 0 時のみ CB0TXE, CB0RXE ビットの書き換え可能です。ただし，同時に CB0PWR ビット = 1 とするのは可能です。

【関　数　名】 csib_start

【処 理 内 容】 CSIB0 動作許可し，送信データ・レジスタに値をライトします。

【使 用 S F R】 CB0CTL0.CB0PWR ：1（CSIB0 動作許可）
CB0TX 送信データ・レジスタ

【call 関数】 なし

【変 数】 unsigned char buf_tx[ ] ：送信データ格納バッファ
volatile unsigned char count_tx ：送信カウント変数

【ファイル名】 csib12.c

【注 意 事 項】 なし

【関　数　名】 csib_end

【処 理 内 容】 CSIB0 動作，送受信動作を禁止します。

【使 用S F R】 CB0CTL0.CB0PWR ：0（CSIB0 動作禁止）
CB0CTL0.CB0RXE ：0（CSIB0 受信動作禁止）
CB0CTL0.CB0TXE ：0（CSIB0 送信動作禁止）

【call 関数】 なし

【変 数】 なし

【ファイル名】 csib12.c

【注 意 事 項】 なし

割り込み関数

【関　数　名】 csib_error

【処 理 内 容】 受信エラー・フラグをクリアします。

【使 用S F R】 CB0RX 受信データ・レジスタ
CB0STR.CB0OVE ：0（オーバラン・エラー・フラグ・クリア）

【call 関数】 なし

【変 数】 unsigned char buf_rx[ ] ：受信データ格納バッファ
volatile unsigned char count_rx ：受信カウント変数

【ファイル名】 csib12.c

【注 意 事 項】 なし

【関　数　名】 csib_int_send

【処 理 内 容】 次のデータを送信するための新たなデータをセットします。

【使 用S F R】 CB0TX 送信データ・レジスタ

【call 関数】 なし

【変 数】 unsigned char buf_tx[ ] ：送信データ格納バッファ
volatile unsigned char count_tx ：送信カウント変数

【ファイル名】 csib12.c

【注 意 事 項】 なし

【関　数　名】 csib_int_receive

【処　理　内　容】 受信したデータをバッファに格納します。

【使　用ＳＦＲ】 CB0RX　　受信データ・レジスタ

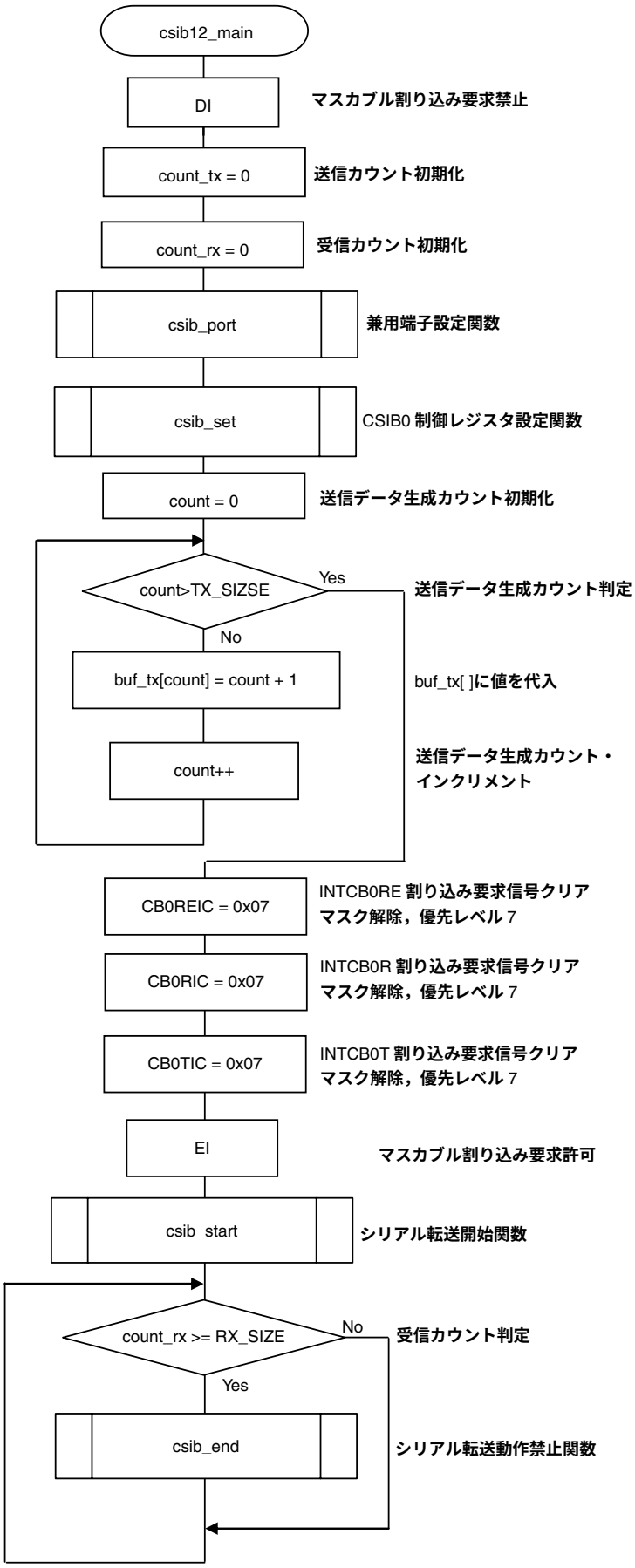
【call　　関数】 なし

【変　　数】 unsigned char buf_rx[ ]　　：受信データ格納バッファ
volatile unsigned char count_rx　　：受信カウント変数

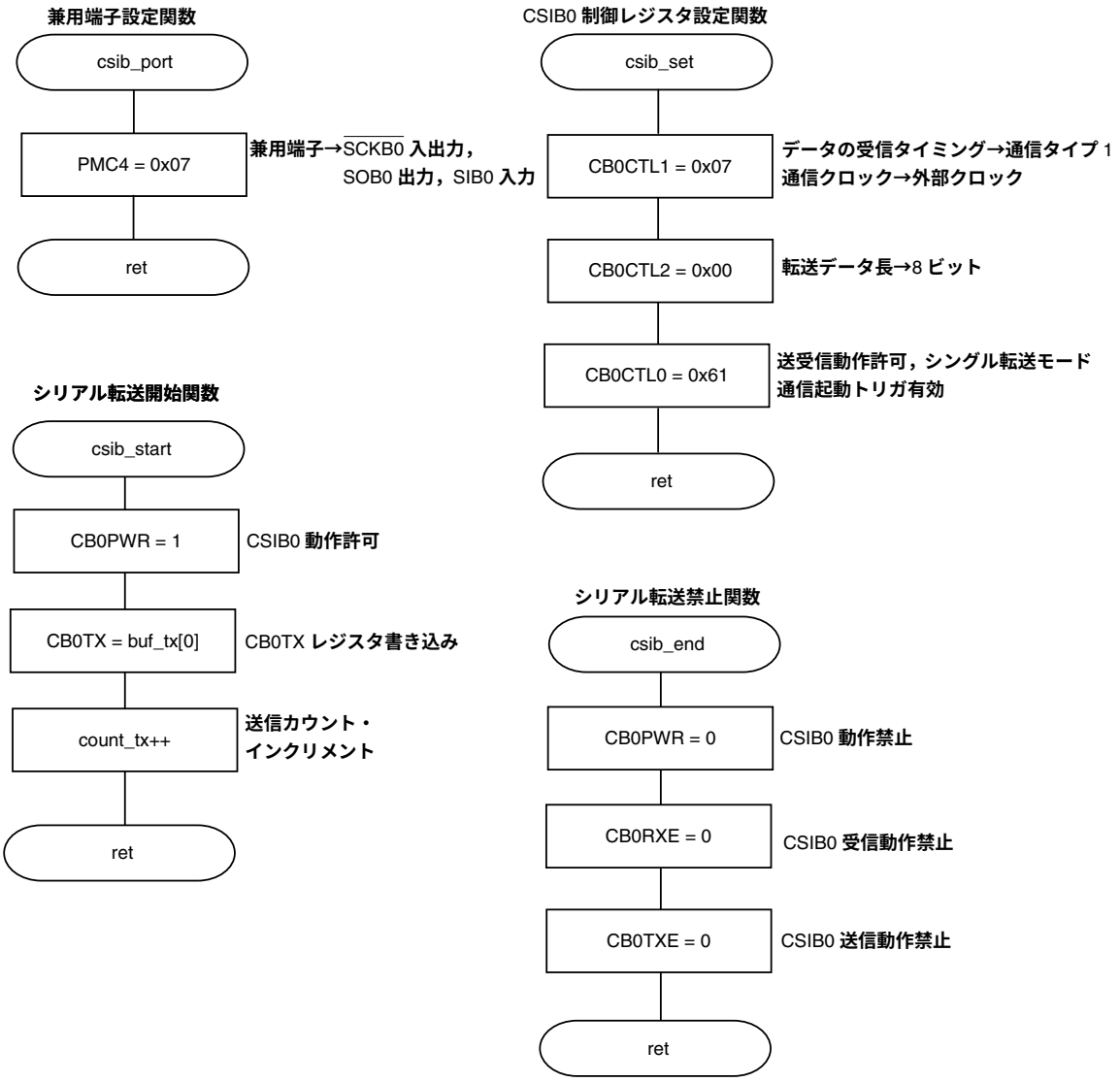
【ファイル名】 csib12.c

【注　意　事　項】 なし

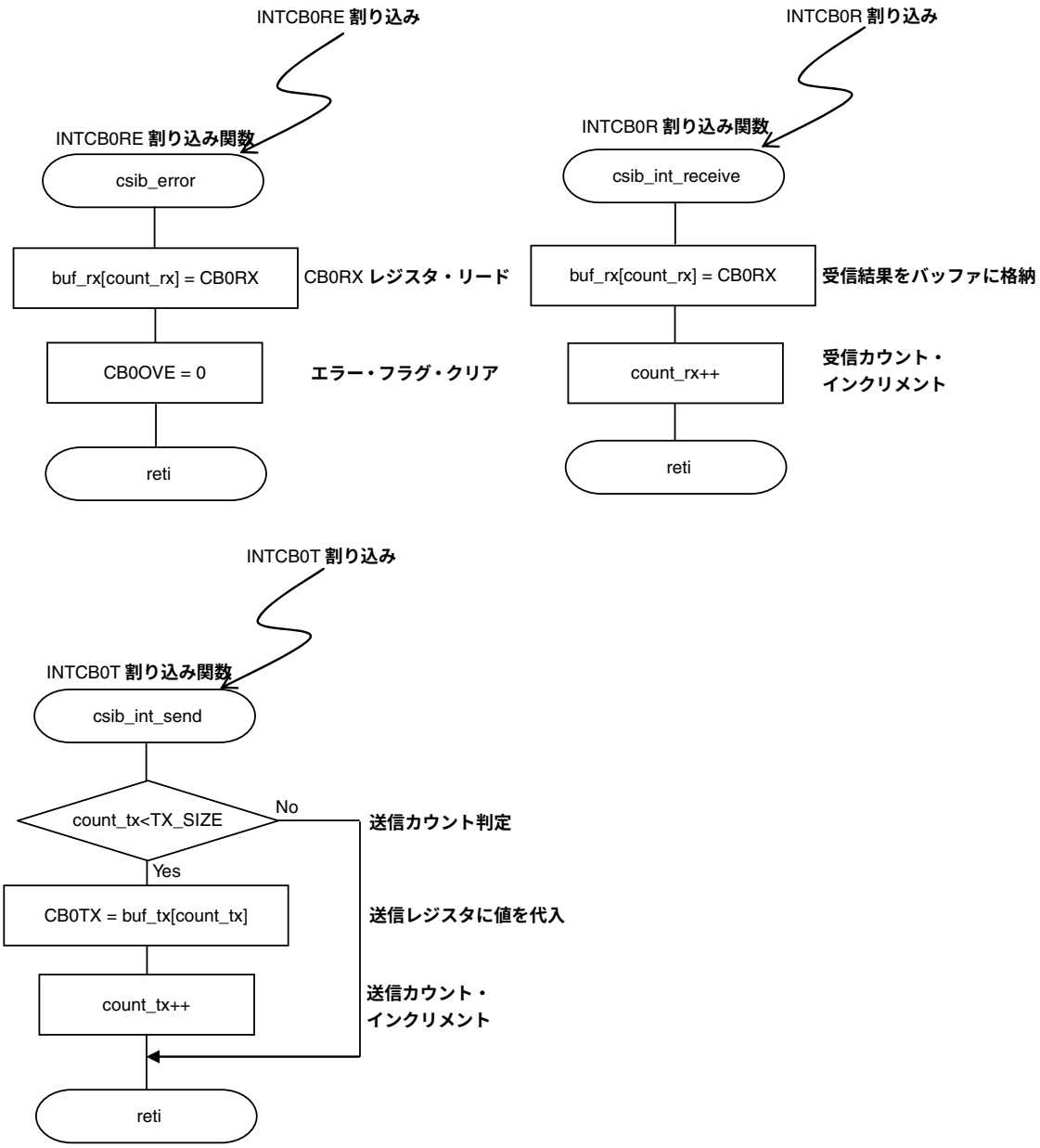
シングル転送モード（スレーブ・モード，送受信モード）（1/3）



シングル転送モード（スレーブ・モード，送受信モード）（2/3）



シングル転送モード（スレーブ・モード，送受信モード）（3/3）



【発　行】

NECエレクトロニクス株式会社

〒211-8668　神奈川県川崎市中原区下沼部1753
電話（代表）：044(435)5111

お問い合わせ先

【ホームページ】

NECエレクトロニクスの情報がインターネットでご覧になれます。
URL(アドレス)　http://www.necel.co.jp/

【営業関係，技術関係お問い合わせ先】

半導体ホットライン
（電話：午前 9:00～12:00，午後 1:00～5:00）

電　話　：044−435−9494
E-mail　：info@necel.com

【資料請求先】

NECエレクトロニクスのホームページよりダウンロードいただくか，NECエレクトロニクスの販売特約店へお申し付けください。

C04.2T