
SH7262/SH7264 グループ

R01AN0548JJ0101

Rev. 1.01

低消費電力モード ディープスタンバイ処理例

2011.02.23

要旨

本アプリケーションノートは、SH7262/SH7264 の低消費電力モードを使用したディープスタンバイ処理例について説明します。

動作確認デバイス

SH7262/SH7264

以下、総称して「SH7264」として説明します。

目次

1. はじめに.....	2
2. 応用例の説明.....	3
3. 参考プログラムリスト.....	21
4. 参考ドキュメント.....	35

1. はじめに

1.1 仕様

本アプリケーションノートでは、ディープスタンバイモードを使用した応用例を説明します。ディープスタンバイモード中もリアルタイムクロックを動作させ、保持が必要なデータを保持用内蔵 RAM に保存します。また、ディープスタンバイモード時に外部デバイスの電源を切断する場合の例も説明します。

1.2 使用機能

- 低消費電力モード
- リアルタイムクロック (RTC)
- 割り込みコントローラ (INTC)
- 汎用入出力ポート
- FIFO 内蔵シリアルコミュニケーションインタフェース (SCIF)

1.3 適用条件

マイコン	SH7262/SH7264
動作周波数	内部クロック : 144 MHz バスクロック : 72 MHz 周辺クロック : 36 MHz
統合開発環境	ルネサスエレクトロニクス製 High-performance Embedded Workshop Ver.4.07.00
C コンパイラ	ルネサスエレクトロニクス製 SuperH RISC engine ファミリ C/C++コンパイラパッケージ Ver.9.03 Release00
コンパイルオプション	High-performance Embedded Workshop でのデフォルト設定 (-cpu=sh2afpu -fpu=single -object="\$(CONFIGDIR)¥\$(FILELEAF).obj" -debug -gbr=auto -chgincpath -errorpath -global_volatile=0 -opt_range=all -infinite_loop=0 -del_vacant_loop=0 -struct_alloc=1 -nologo)

1.4 関連アプリケーションノート

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

- SH7262/SH7264 グループ 初期設定例

1.5 "L"アクティブ端子 (信号) の表記について

端子名(信号名)末尾の # は "L" アクティブ端子(信号)であることを示します。

2. 応用例の説明

この章では、外部デバイスとの端子接続例を含むディープスタンバイモードの概要と、本応用例におけるディープスタンバイモード設定手順の参考プログラムフローを説明します。参考プログラムは、ディープスタンバイモード時に外部デバイスの電源を切断しない前提で設定を行っていますが、参考例として、外部デバイスの電源を切断する例も紹介します。

2.1 ディープスタンバイモードの動作概要

2.1.1 低消費電力モードの概要

SH7264 の低消費電力モードは、スリープモード、ソフトウェアスタンバイモード、ディープスタンバイモード、モジュールスタンバイモードの 4 種類があります。表 1 に各モードでの状態および解除方法を示します。

本応用例で使用するディープスタンバイモードは、LSI 内部の各モジュールに対して、電源供給を停止するため、他の低消費電力モードよりも消費電力を大幅に低減させることができます。データは保持用内蔵 RAM に保存することができます。

表1 低消費電力モードの状態

低消費電力モード	状態									解除方法
	CPG	CPU	CPUレジスタ	高速内蔵RAMキャッシュメモリ	大容量内蔵RAM (保持用内蔵RAM含む)	内蔵周辺モジュール	リアルタイムクロック	電源	外部メモリ	
スリープモード	動作	停止	保持	動作	動作	動作	動作 ^{※1}	動作	オートリフレッシュしてください	<ul style="list-style-type: none"> ・割り込み ・マニュアルリセット ・パワーオンリセット ・DMA アドレスエラー
ソフトウェアスタンバイモード	停止	停止	保持	停止 (内容は保持) ^{※2※3}	停止 (内容は保持) ^{※2※4}	停止	動作 ^{※1}	動作	セルフリフレッシュしてください	<ul style="list-style-type: none"> ・NMI 割り込み ・IRQ 割り込み ・パワーオンリセット
ディープスタンバイモード	停止	停止	停止	停止 (内容は非保持)	停止 (保持用内蔵RAMの内容は保持) ^{※5}	停止	動作 ^{※1}	停止	セルフリフレッシュしてください	<ul style="list-style-type: none"> ・NMI 割り込み^{※6} ・パワーオンリセット^{※6} ・リアルタイムクロックアラーム割り込み^{※6} ・解除用端子の変化^{※6}
モジュールスタンバイモード	動作	動作	保持	動作	動作	指定モジュールが停止	停止	動作	オートリフレッシュしてください	<ul style="list-style-type: none"> ・MSTP ビットを0にクリア ・パワーオンリセット (ただしユーザデバッグインタフェース、ダイレクトメモリアクセスコントロールのみ)

【注】^{※1} リアルタイムクロックはRCR2 レジスタのSTART ビットが1のとき動作します。なお、パワーオンリセットによりディープスタンバイモードを解除した場合、動作状態を保持することができません。再度、リアルタイムクロックの初期設定を行ってください。

^{※2} パワーオンリセットによりソフトウェアスタンバイモードを解除した場合、保持している内容は初期化されます。

^{※3} 高速内蔵RAMは、SYSCR1 レジスタのRAME ビット、またはSYSCR2 レジスタのRAMWE ビットをディスエーブルにすることにより、パワーオンリセットでソフトウェアスタンバイモードを解除した場合でも保持し続けることができます。

^{※4} 大容量内蔵RAM (保持用内蔵RAM含む)は、SYSCR3 レジスタのVRAME ビット、またはSYSCR4 レジスタのVRAMWE ビットをディスエーブルにすることにより、パワーオンリセットでソフトウェアスタンバイモードを解除した場合でも、データを保持し続けることができます。

^{※5} RRAMKP レジスタのRRAMKP3~RRAMKP0 ビットを1に設定すると、ディープスタンバイモード遷移時に保持用内蔵RAMの対象エリアの内容を保持することができます。ただし、パワーオンリセットによりディープスタンバイモードを解除した場合、保持している内容は初期化されます。RRAMKP3、RRAMKP2 ビットは、640KB版でのみ使用できます。

^{※6} ディープスタンバイモードは、割り込み (NMI、リアルタイムクロックアラーム割り込み)、リセット (パワーオンリセット) および解除用端子 (PC8~PC5、PG11、PG10、PJ3、PJ1) の変化により解除されます。ただし、リセット以外によってディープスタンバイモードを解除する場合も、割り込み例外処理ではなくパワーオンリセット例外処理が実行されます。PG11、PG10は、640KB版でのみ解除用端子として動作します。

2.1.2 ディープスタンバイモードへの遷移手順

図 1にディープスタンバイモードへの遷移手順を示します。保持用内蔵RAMやリアルタイムクロックの設定、ディープスタンバイモードの解除要因や起動方法を設定します。

低消費電力モード関連のレジスタを設定する場合において、設定値をレジスタに反映させた上で後続命令を実行する場合には、レジスタ書き込み命令と後続命令の間に同じレジスタに対するダミーリードが必要です。また、SLEEP 命令実行と同時に NMI、IRQ 割り込みおよびマニュアルリセットが発生すると、割り込みを受け付けてディープスタンバイが解除される場合があるため、注意してください。

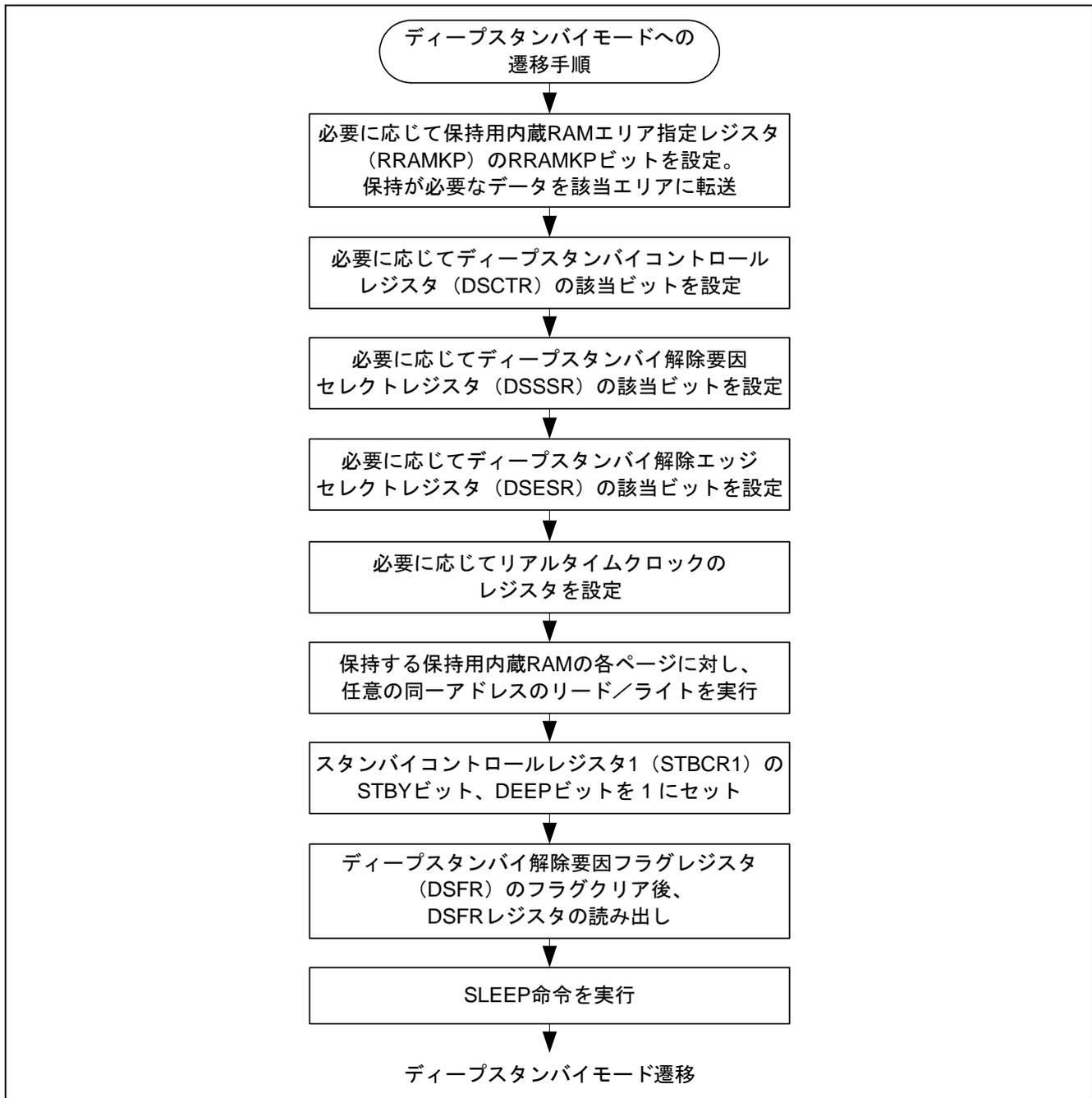


図1 ディープスタンバイモードへの遷移手順

2.1.3 ディープスタンバイモードの端子状態

ディープスタンバイモード時の出力端子の状態として、端子状態の保持または Hi-Z のどちらかを選択できます。ただし、個々の端子毎に設定することはできません。表 2 に端子状態一覧（抜粋）を、表 3 にディープスタンバイモード時の端子状態を設定するレジスタ一覧を示します。端子状態の詳細については「SH7262 グループ、SH7264 グループ ハードウェアマニュアル 付録 A.端子状態」を参照してください。

なお、ディープスタンバイモード解除後は、DSFR レジスタの IOKEEP ビットを 0 にクリアするまで一部の端子を除いて端子状態が保持されます。ディープスタンバイ解除後の端子状態については、2.1.6 保持用内蔵 RAM からの起動と端子保持の解除を参照してください。

表2 端子状態一覧（抜粋）

端子機能		端子状態						
分類	端子名		通常状態	パワーオンリセット	ディープスタンバイ解除から IOKEEP ビットのクリアまで		ディープスタンバイモード	
					EBUSKEEPE ビット			リセット状態 ^{※1}
					0	1		
クロック	EXTAL	クロック動作モード 0,2	I	I	I		I/Z ^{※2}	
		クロック動作モード 1,3	Z	Z	Z		Z	
	XTAL		O	O	O		O/L ^{※2}	
	CKIO		O	O	O	O/Z ^{※3}	O/Z ^{※3}	
アドレスバス	A25~A21, A0		O	— ^{※4}	O/Z ^{※5}		O/Z ^{※5}	
	A20 ~ A1	ブートモード 0	O	Z	O	O/Z ^{※5}	O/Z ^{※5}	
		ブートモード 1~3	O	— ^{※4}	O/Z ^{※5}		O/Z ^{※5}	
データバス	D15 ~ D0	ブートモード 0	I/Z	Z	I/Z	Z	Z	
			O/Z	Z	O/Z	Z	Z	
		ブートモード 1~3	I/Z	— ^{※4}	—		Z	
			O/Z	— ^{※4}	Z		Z	
バス制御	CS0#	ブートモード 0	O	Z	O	H/Z ^{※5}	H/Z ^{※5}	
		ブートモード 1~3	O	— ^{※4}	H/Z ^{※5}		H/Z ^{※5}	
	CS6# ~ CS1#, CE1A#, CE1B#, CE2A#, CE2B#		O	— ^{※4}	H/Z ^{※5}		H/Z ^{※5}	
	RD#	ブートモード 0	O	Z	O	H/Z ^{※5}	H/Z ^{※5}	
		ブートモード 1~3	O	— ^{※4}	H/Z ^{※5}		H/Z ^{※5}	
	RD/WR#, BS#, ICIOWR#/AH#, ICIORD#, WE1#/DQMLU/WE#, WE0#/DQMLL		O	— ^{※4}	H/Z ^{※5}		H/Z ^{※5}	
	RAS#, CAS#, CKE		O	— ^{※4}	O/Z ^{※6}		O/Z ^{※6}	
	WAIT#, IOIS16#		I	— ^{※4}	—		Z	
リアルタイムクロック	RTC_X1		I/Z	I	I/Z ^{※7}		I/Z ^{※7}	
	RTC_X2		O/H	O	O/H ^{※7}		O/H ^{※7}	
汎用入出力ポート	PB21 (A21)		I	Z	Z		Z	
			O/Z	Z	O/Z ^{※8}		O/Z ^{※8}	

【記号説明】 I: 入力、O: 出力、H: ハイレベル出力、L: ローレベル出力、Z: ハイインピーダンス

【注】 ^{※1} ディープスタンバイモードから復帰後、一定期間パワーオンリセット状態になります。

^{※2} コントロールレジスタ 5 (RCR5) の RCKSEL ビットの設定に従います。

^{※3} 周波数制御レジスタ (FRQCR) の CKOEN ビットの設定に従います。

^{※4} 端子状態は、初期化された端子機能になります（汎用入出力ポート）。

^{※5} 共通コントロールレジスタ (CMNCR) の HIZMEM ビットの設定に従います。

^{※6} 共通コントロールレジスタ (CMNCR) の HIZCNT ビットの設定に従います。

^{※7} コントロールレジスタ 2 (RCR2) の RTCEN ビットの設定に従います。

^{※8} スタンバイコントロールレジスタ 3 (STBCR3) の HIZ ビットの設定に従います。

表3 ディープスタンバイモード時の端子状態を設定するレジスタ一覧

レジスタ名	アドレス	ビット名	設定値	内容
周波数制御レジスタ (FRQCR)	H'FFFE 0010	CKOEN[1:0]	B'00 (または B'11)	ディープスタンバイモード時に CKIO 端子の出力を L レベル出力または H レベル出力 (または Hi-Z) にします。
スタンバイコントロールレジスタ (STBCR3)	H'FFFE 0408	HIZ	B'0 (または B'1)	ディープスタンバイモード時に、出力端子を端子状態保持 (または Hi-Z) にします。*
共通コントロールレジスタ (CMNCR)	H'FFFC 0000	HIZMEM	B'1 (または B'0)	ディープスタンバイモード時に A25~A0、BS#、CSn#、CE2x#、RD#、WE#、DQMx/AH#、RD# の端子状態をドライブ状態 (または Hi-Z) にします。
		HIZCNT	B'1 (または B'0)	ディープスタンバイモード時に CKE、RAS#、CAS# の端子状態をドライブ状態 (または Hi-Z) にします。
コントロールレジスタ 5 (RCR5)	H'FFFE 6026	RCKSEL	B'0 (または B'1)	RTC の動作クロックを RTC_X1 (または EXTAL) に設定します。
コントロールレジスタ 2 (RCR2)	H'FFFE 601E	RTCEN	B'1 (または B'0)	RTC_X1 端子の内蔵水晶発振器を動作/外部クロック入力許可 (または内蔵水晶発振器を停止/外部クロック入力禁止) に設定します。

【注】* この設定はウォッチドッグタイマの WTSCR レジスタの TME ビットが 0 の状態で行ってください。

2.1.4 外部デバイスの電源を切断しない場合の端子接続例

図2にディープスタンバイモード時に外部デバイスの電源を切断しない場合の端子接続例を示します。NORフラッシュメモリを接続し、ブートモード0で起動する場合を想定しています。

外部デバイスの電源を切断しない場合は、端子状態を保持する設定を行います。プルアップ/プルダウン抵抗に不要な電流が流れないようにしてください。

図2の接続例では、PB21/A21 端子にプルダウン抵抗が接続されています。そのため、参考プログラムでは、ディープスタンバイモードに遷移する前にPB21/A21 端子の機能を汎用入力ポートに設定しています。CS0#、RD#、WE0#端子およびD15~D0 端子は、プルアップ抵抗が接続されていますが、これらの端子はHレベルに端子状態を保持するため、特別な処理は不要です。端子状態の詳細については表2を参照してください。

また、消費電力を軽減するために、プルアップ/プルダウン抵抗は 100kΩ を推奨します。

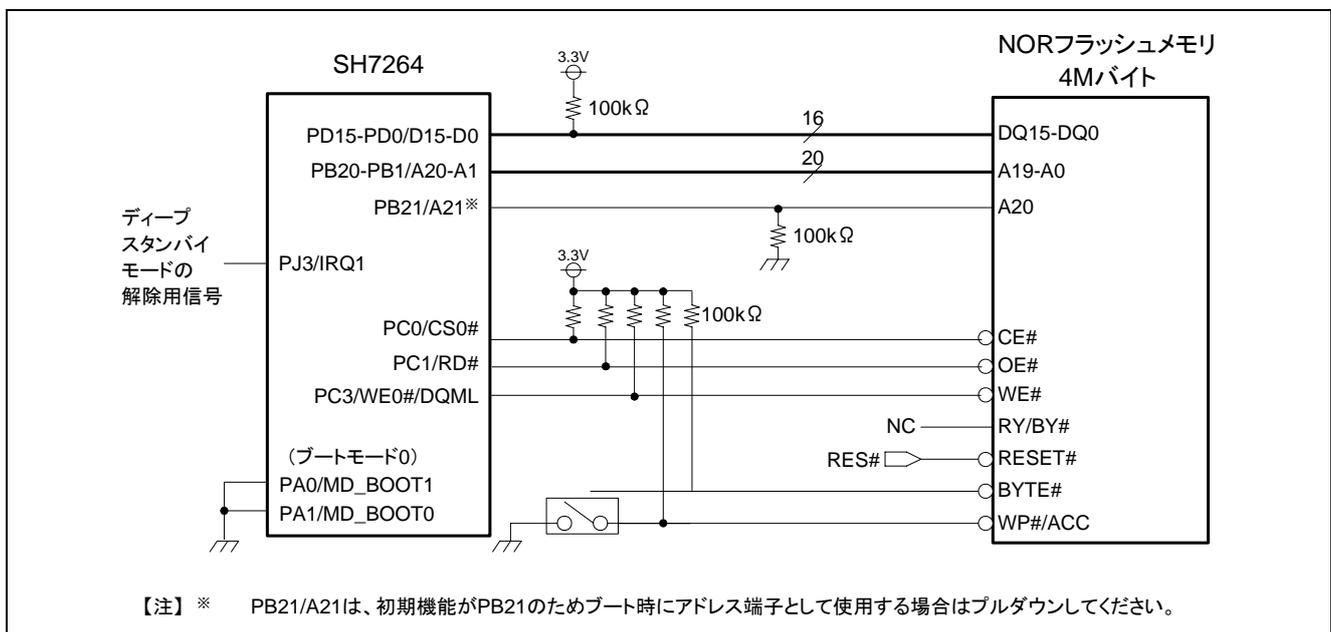


図2 端子接続例（外部デバイスの電源を切断しない場合）

2.1.5 ディープスタンバイモードの解除

表 4にディープスタンバイモードの解除要因を示します。

NMI 端子および解除要因に選択した全ての解除用端子は、ディープスタンバイモード遷移時の端子レベルを以下の状態にしてください。

- 解除を立ち上がりエッジに設定した場合：L レベル
- 解除を立ち下がりエッジに設定した場合：H レベル

また、解除要因となる端子は、発振安定時間経過時の端子レベルを以下の状態にしてください。なお、リセットによる解除の場合は、クロックの発振が安定するまで RES#端子を L レベルに保持してください。

- 解除を立ち上がりエッジに設定した場合：H レベル
- 解除を立ち下がりエッジに設定した場合：L レベル

表4 ディープスタンバイモードの解除要因

解除要因	トリガ	関連レジスタ	備考
RES#	L レベル	—	
NMI 割り込み	立ち上がりエッジ、 または立ち下がり エッジ	ディープスタンバイ解除エッジ セレクトレジスタ (DSESR)	
リアルタイムクロックの アラーム割り込み	指定時間経過	ディープスタンバイ解除要因 セレクトレジスタ (DSSSR)	割り込み優先レベルに 関係なく受け付けられ ます。
解除用端子	立ち上がりエッジ、 または立ち下がり エッジ	ディープスタンバイ解除エッジ セレクトレジスタ (DSESR) ディープスタンバイ解除要因 セレクトレジスタ (DSSSR)	

図3に ディープスタンバイモードの解除フローを示します。

リセットによる解除は、通常のパワーオンリセットと同様、リセット時のブートモードに従ったパワーオンリセット例外処理が実行されます。リセット以外による解除は、LSI内部でカウントした発振安定時間経過後、パワーオンリセット例外処理が実行されます。ただし、DSCTRレジスタのRAMBOOTビットの値によってプログラムカウンタ (PC) とスタックポインタ (SP) を読み出す領域が異なります。起動方法の詳細については 2.1.6 保持用内蔵RAMからの起動と端子保持の解除を参照してください。

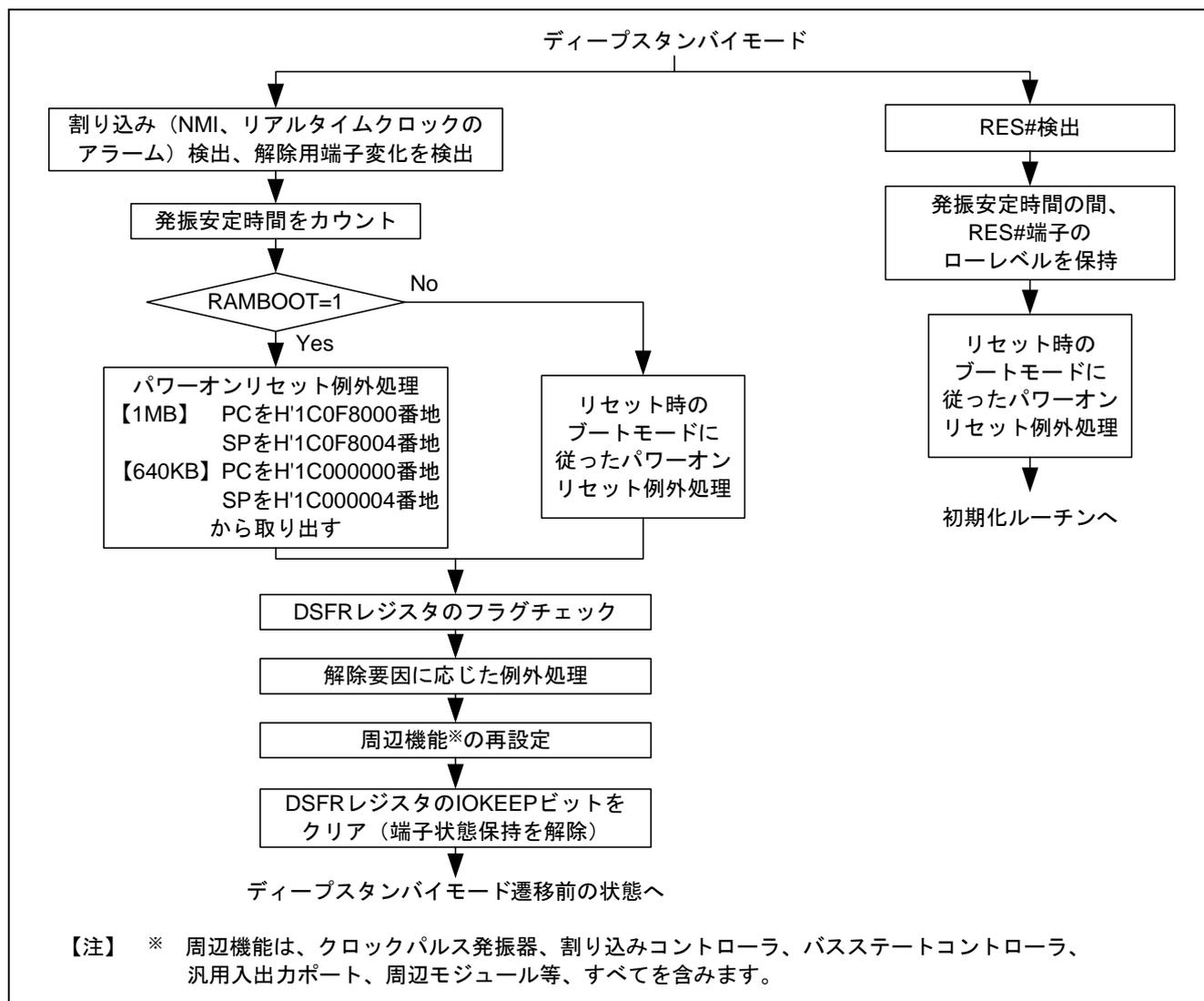


図3 ディープスタンバイモードの解除フロー

2.1.6 保持用内蔵 RAM からの起動と端子保持の解除

ディープスタンバイモード解除後の動作は、通常のブートモードに従って外部メモリから起動するか、保持用内蔵RAMから起動するかのどちらかをDSCTRレジスタのRAMBOOTビットによって設定します。ディープスタンバイモード解除直後に外部メモリをアクセスできない場合や、ディープスタンバイモード解除後のブートプログラムとパワーオンリセットによるブートプログラムを区別したい場合には、保持用内蔵RAMからの起動を選択します。表5にディープスタンバイモード解除後の起動方法と端子保持状態を示します。

表5 ディープスタンバイモード解除後の起動方法と端子保持状態

DSCTR レジスタの EBUSKEEPE ビット	DSCTR レジスタの RAMBOOT ビット	起動方法	ディープスタンバイモード 解除後の端子状態
0	0	外部メモリ	外部メモリ制御端子は保持しない。 その他の端子は IOKEEP ビットクリア時に端子保持解除。
0	1	保持用内蔵 RAM	外部メモリ制御端子は保持しない。 ディープスタンバイ解除後外部メモリ制御端子は端子保持解除。 その他の端子は IOKEEP ビットクリア時に端子保持解除。
1	0	—	設定禁止
1	1	保持用内蔵 RAM	外部メモリ制御端子は保持する。 外部メモリ制御端子もその他の端子も、 IOKEEP ビットクリア時に端子保持解除。

ディープスタンバイモード解除後の端子保持状態は、DSCTR レジスタの EBUSKEEP ビットと、DSFR レジスタの IOKEEP ビットによって解除されます。

IOKEEP ビットを 0 にクリアすることで全ての端子の端子保持状態が解除されますが、外部メモリ制御端子は、EBUSKEEP ビットが 0 の場合、ディープスタンバイモード解除直後に端子保持状態が解除されます。外部メモリから起動する場合は EBUSKEEP ビットを 0 に設定してください。表6に EBUSKEEPE ビットの設定が有効となる外部メモリ制御端子の一覧を示します。また、図4に EBUSKEEPE ビットと端子保持タイミングを示します。

表6 EBUSKEEPE ビットの設定が有効となる外部メモリ制御端子の一覧

ブートモード 0 (CS0 空間)	ブートモード 2 (NAND フラッシュメモリ)	ブートモード 1、3 (シリアルフラッシュメモリ)
A[20:1] D[15:0] CS0#, RD#, CKIO	NAF[7:0] FRE#, FCLE, FALE, FWE#, FCE#, FRB	RSPCK0, SSL00, MOSI0, MISO0

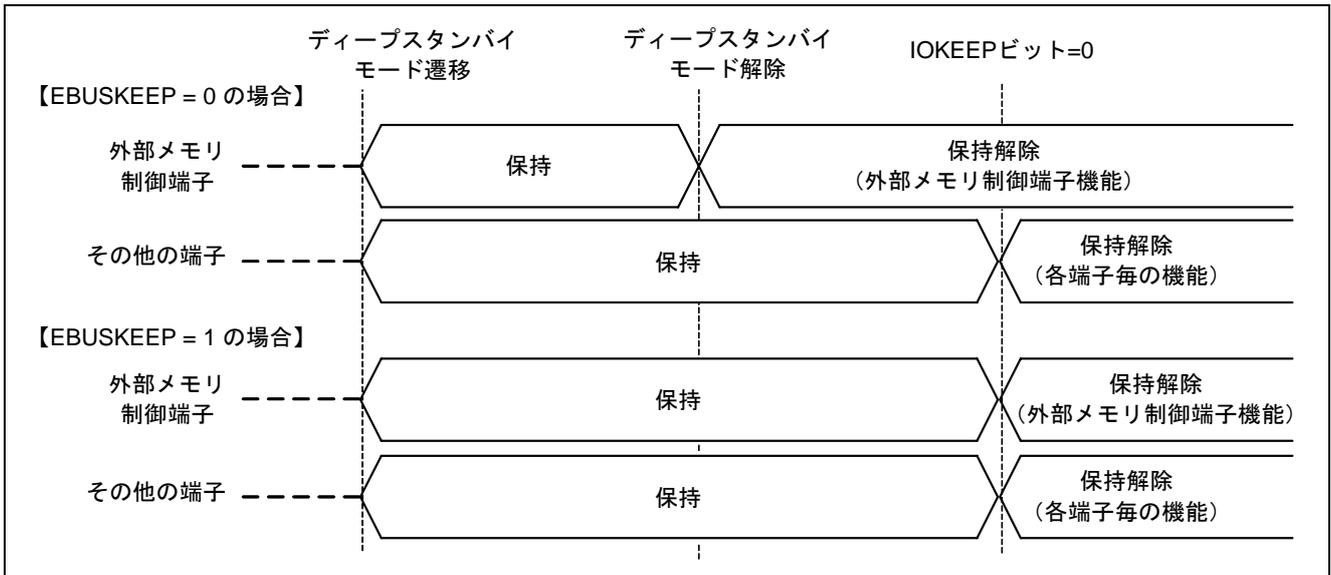
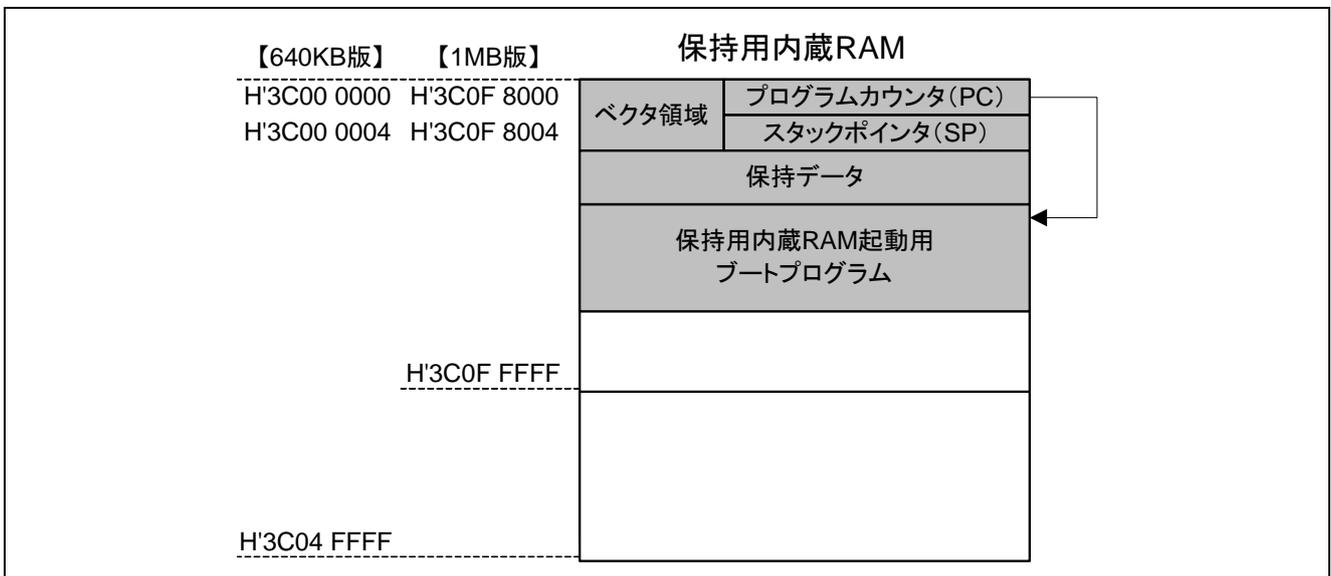


図 5に 保持用内蔵RAMから起動する場合の動作イメージを示します。ベクタ領域からプログラムカウンタとスタックポインタの値を取り出し、保持用内蔵RAMからの起動用に用意したブートプログラム（以下、RAMブートプログラムとします。）に分岐します。ベクタ領域の値とRAMブートプログラムは、ディープスタンバイモードに遷移する前に設定してください。

なお、保持用内蔵 RAM の容量とベクタ領域のアドレスが 1MB 版と 640KB 版で異なるため、注意してください。



2.1.7 応用例のタイミングチャート

図6に本応用例のタイミングチャートを示します。NMI割り込みによりディープスタンバイモードへの遷移設定と遷移を行い、PJ3端子の立ち下がりエッジでディープスタンバイモードを解除します。

本応用例では保持用内蔵RAMから起動させるためRAMBOOTビットを1に設定します。また、CKOEN[1:0]ビットをB'11に設定し、ディープスタンバイ中はCKIO端子を出力オフとします。CKIOはディープスタンバイ解除後、IOKEEPビットを0クリアしてから出力されます。

なお、パワーオンリセット例外処理ではレジスタの初期化が行われますが、解除要因がリセットである場合を除き、ディープスタンバイモード解除時に実行されるパワーオンリセット例外処理ではDSFRレジスタの値は保持されます。解除要因がリセットの場合はDSFRレジスタも初期化されます。そのため、DSFRレジスタを参照することにより、ディープスタンバイモードの解除要因を判断することができます。

レジスタの初期化については「SH7262 グループ、SH7264 グループ ハードウェアマニュアル 36.3 各動作モードにおけるレジスタの状態の一覧」を参照してください。

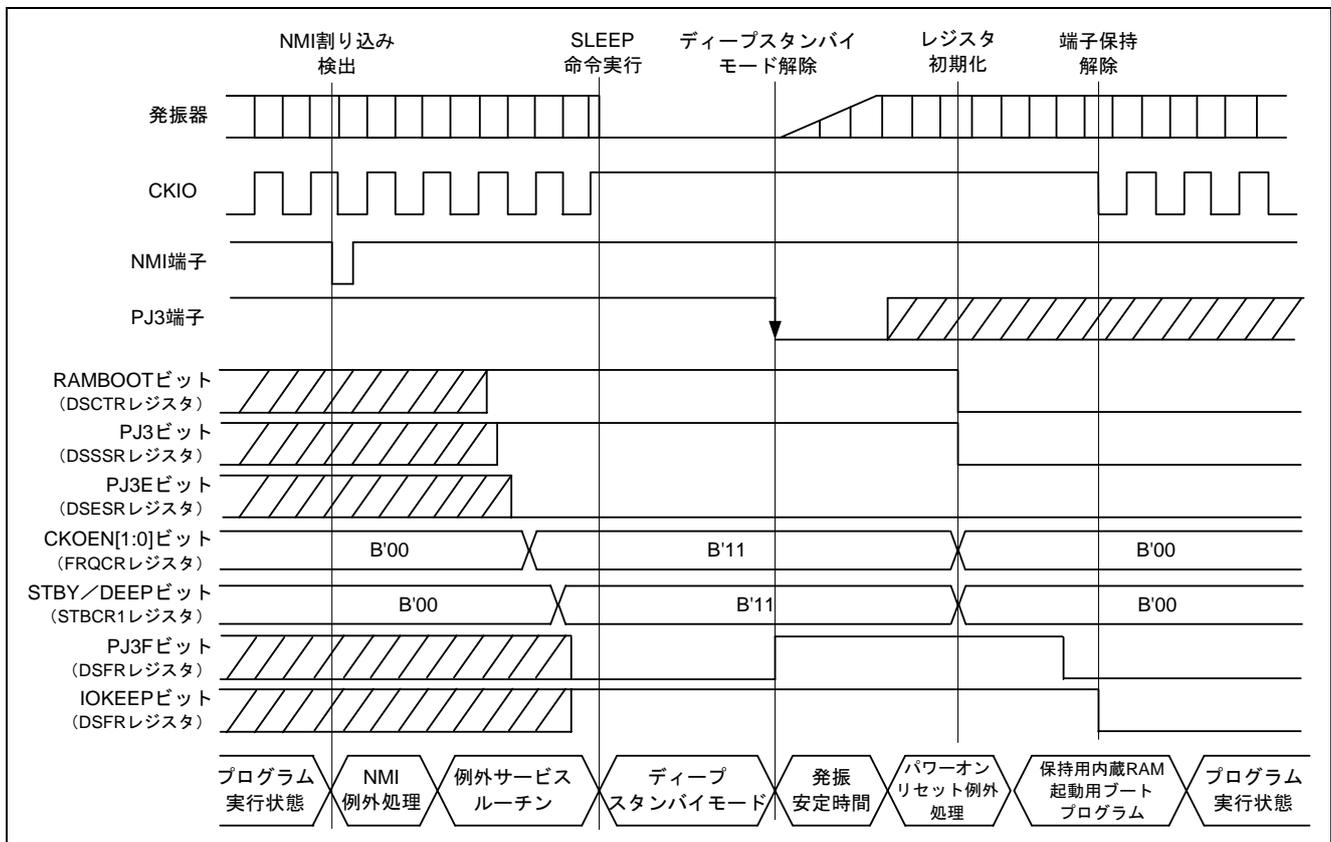


図6 本応用例のタイミングチャート

2.2 参考プログラムの動作

ここでは、参考プログラムの動作を説明します。

メイン関数はリアルタイムクロックの時刻を標準出力に出力し、NMI 割り込みルーチンでディープスタンバイモードへの遷移設定を行います。ディープスタンバイモード解除後は、RAM ブートプログラムが実行され、メイン関数へ分岐します。

なお、参考プログラムは、ディープスタンバイモード時に外部デバイスの電源を切断しない前提で設定を行っています。また、SDRAM のセルフリフレッシュ機能は設定していません。

2.2.1 メイン関数フロー

図7に本参考プログラムにおけるメイン関数フローを示します。リアルタイムクロックの時刻をポーリングし、時刻に変化があった場合は時刻データをSCIFに出力します。

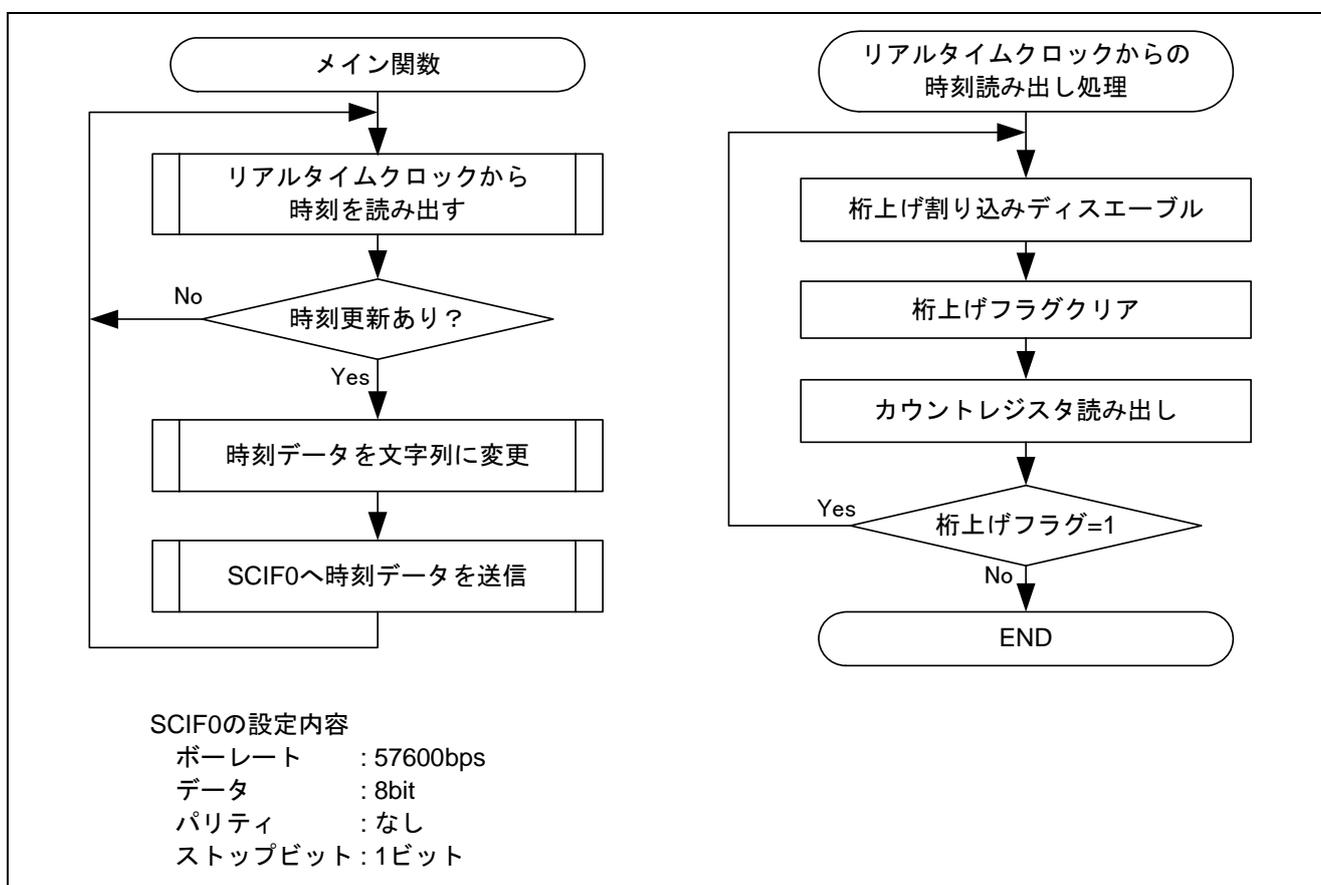


図7 参考プログラムのメイン関数フロー

2.2.2 NMI 割り込みのフロー

図8および図9に参考プログラムのNMI割り込みフローを示します。スタンバイ開始時刻を保持用内蔵RAMに保存し、不要な電流を流さないための端子処理をした後、ディープスタンバイモードへ遷移します。

640KB版の場合、割り込みコントローラレジスタ0 (ICR0) の NMIE ビットによって NMI 割り込みが禁止されます。そのため、パワーオンリセット後およびディープスタンバイモード解除後は、NMIE ビットを 0 にクリアする必要があります。

また、ディープスタンバイモードへ遷移するときに意図せず外部バスをアクセスした状態で端子保持とならないように、ディープスタンバイに遷移する処理では以下の点に注意してください。

- CPU で外部バスをアクセスしないようにプログラム、変数領域を内蔵 RAM に配置するなどセクション配置に注意する
- DMAC、VDC3 など CPU 以外のバスマスタが外部バスをアクセスしないように、ディープスタンバイモードへの遷移前にこれらの機能を停止する

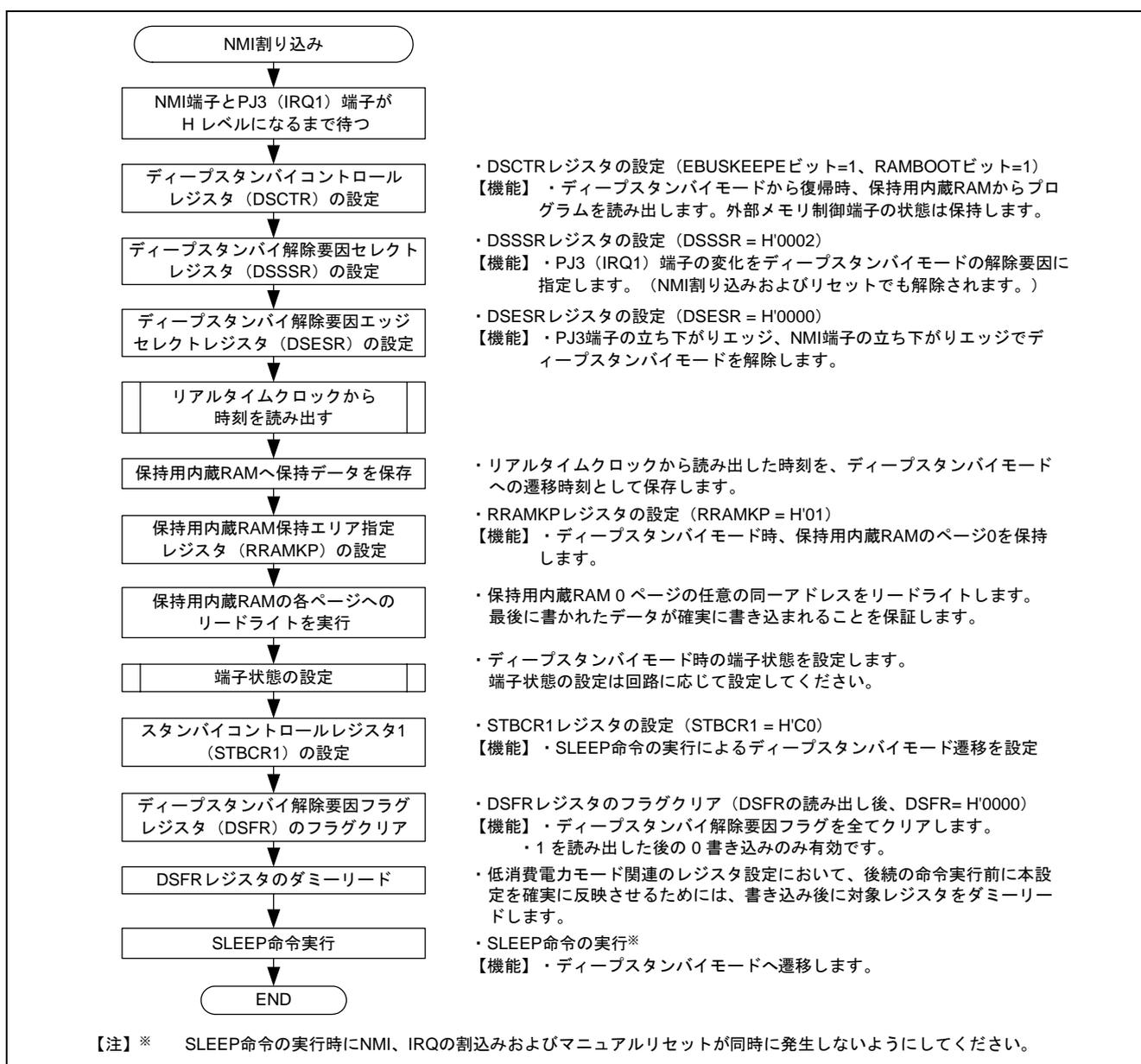


図8 参考プログラムのNMI割り込みフロー1

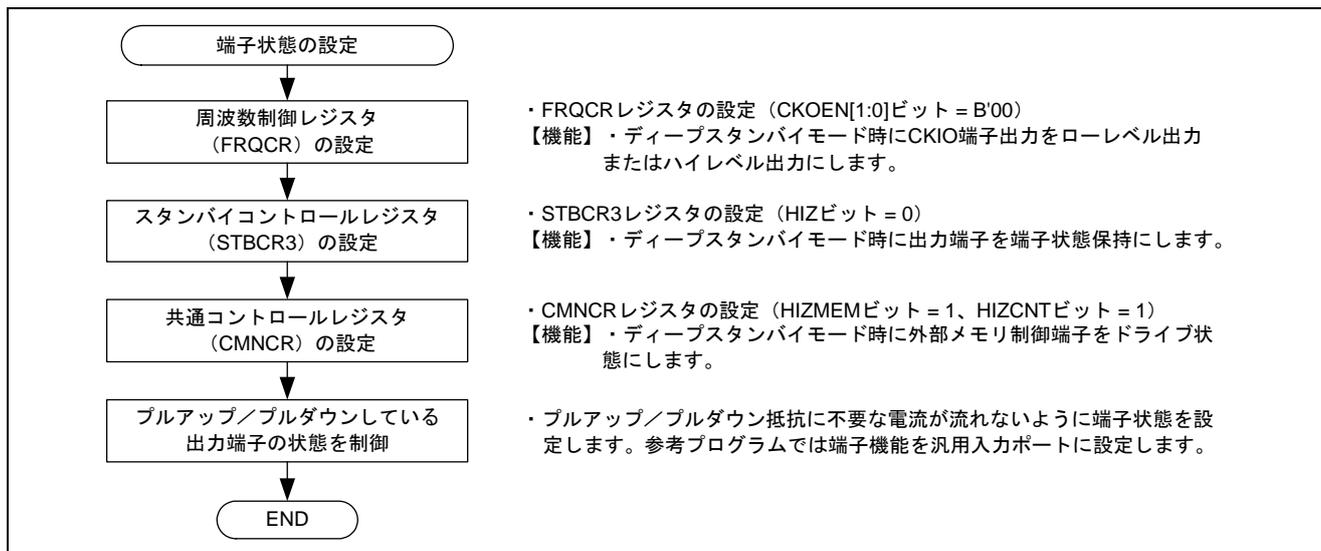


図9 参考プログラムのNMI 割込みフロー2

2.2.3 RAM ブートプログラムのフロー

図 10に参考プログラムのRAMブートプログラムフローを示します。端子保持を解除して、周辺機能およびデータの初期化を行います。その後、起動履歴情報のSCIF出力を行った後、メイン関数に分岐します。

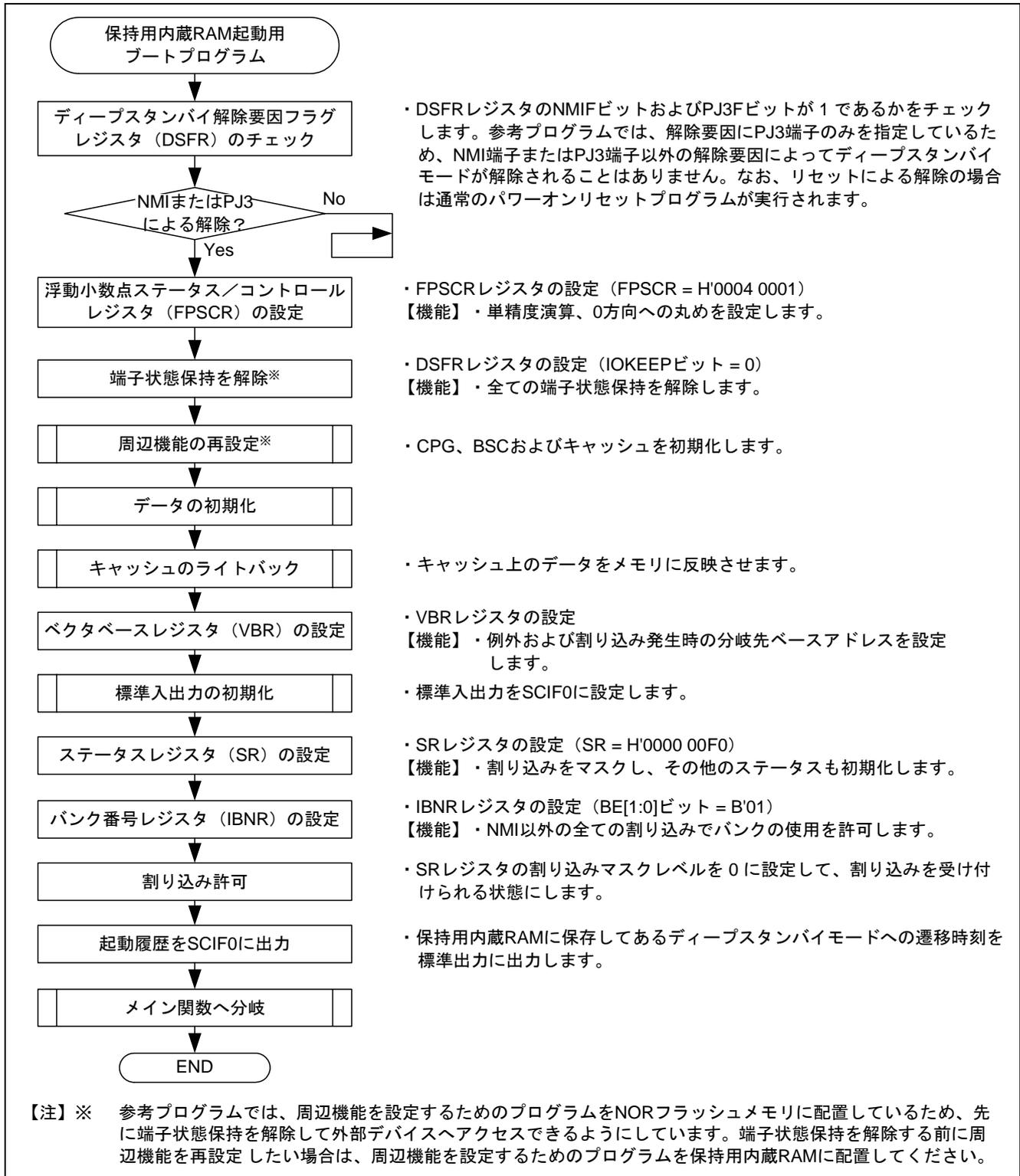


図10 参考プログラムのRAMブートプログラムフロー

2.2.4 リアルタイムクロックの初期化フロー

図 11に 参考プログラムのリアルタイムクロック初期化フローを示します。リアルタイムクロックの初期化はパワーオンリセットプログラムで行っています。

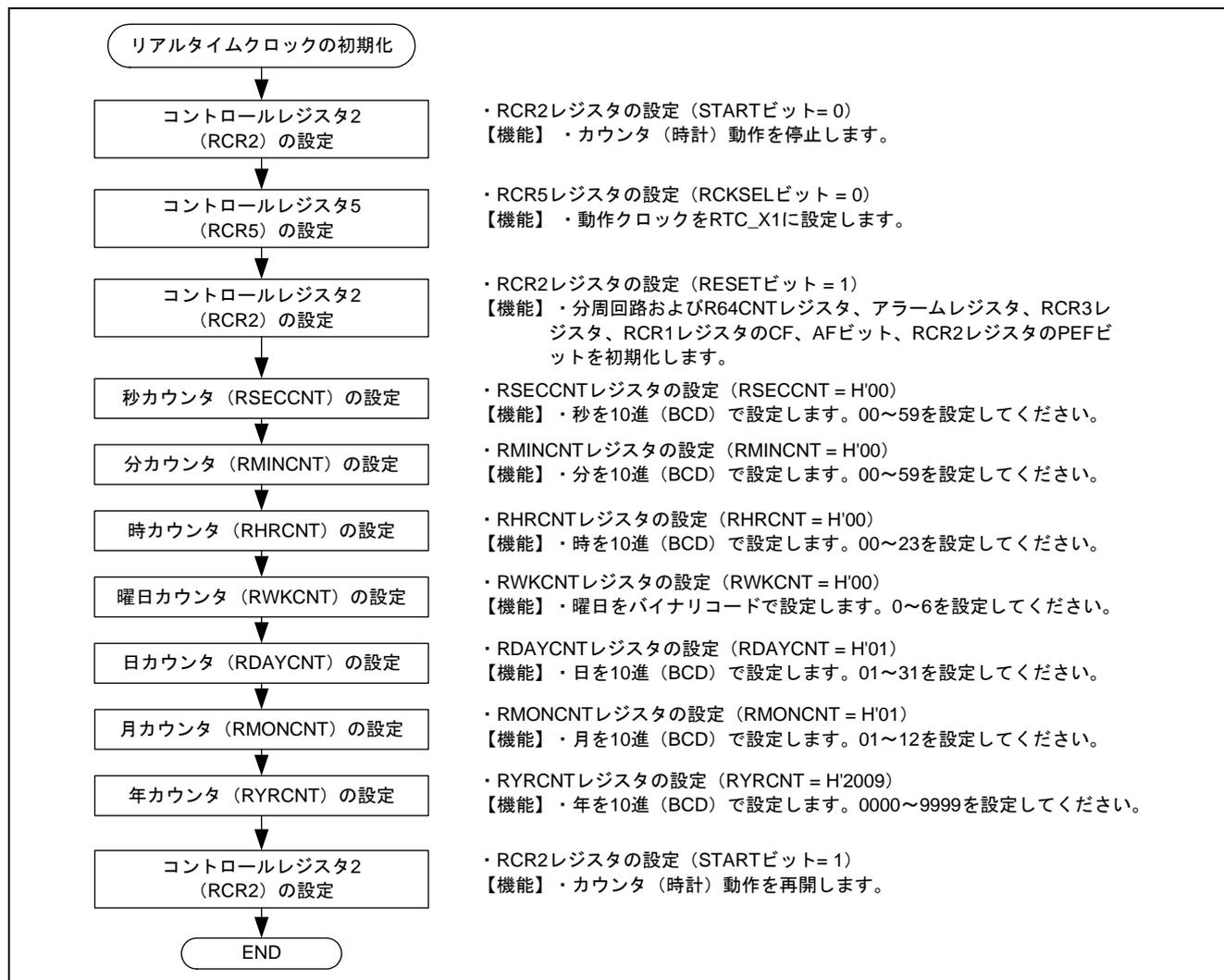


図11 参考プログラムのリアルタイムクロック初期化フロー

2.3 【補足】外部デバイスの電源を切断する場合

本応用例では、ディープスタンバイモード時も、外部デバイスの電源を投入していますが、ディープスタンバイモード時に外部デバイスの電源を切断することで更なる低消費電力化を行うことが可能です。ここでは、外部デバイスの電源を切断するための概要を説明します。

図 12 にディープスタンバイモード時に外部デバイスの電源を切断する場合の端子接続例を示します。NORフラッシュメモリおよびSDRAMを接続し、ブートモード 0 で起動する場合を想定しています。

外部デバイスの電源を切断する場合は、外部デバイスの端子に電圧を印加できないため、ディープスタンバイモード時は出力端子を Hi-Z にします。また、プルアップ抵抗は SH7264 用の電源ではなく、外部デバイス用の電源 (EXVcc) に接続します。リセット信号もディープスタンバイモード中に L レベルとなるよう外部デバイス用のリセット信号 (EXRES#) を生成します。

図 13 に示したタイミングチャート例を満たすように電源を制御してください。NORフラッシュメモリをブートに使用するため、パワーオンリセット直後から EXVcc を投入して、EXRES# を必要期間、アサートする必要があります。また、ディープスタンバイモード時は EXVcc を切断しますので、EXRES# も L レベルにします。ディープスタンバイモード解除後は、端子状態の保持を解除する前に EXVcc を投入しておく必要があります。また SDRAM など初期化が必要なデバイスの場合は、外部デバイスの初期化処理の前に端子状態の保持を解除しなければならないため、注意してください。

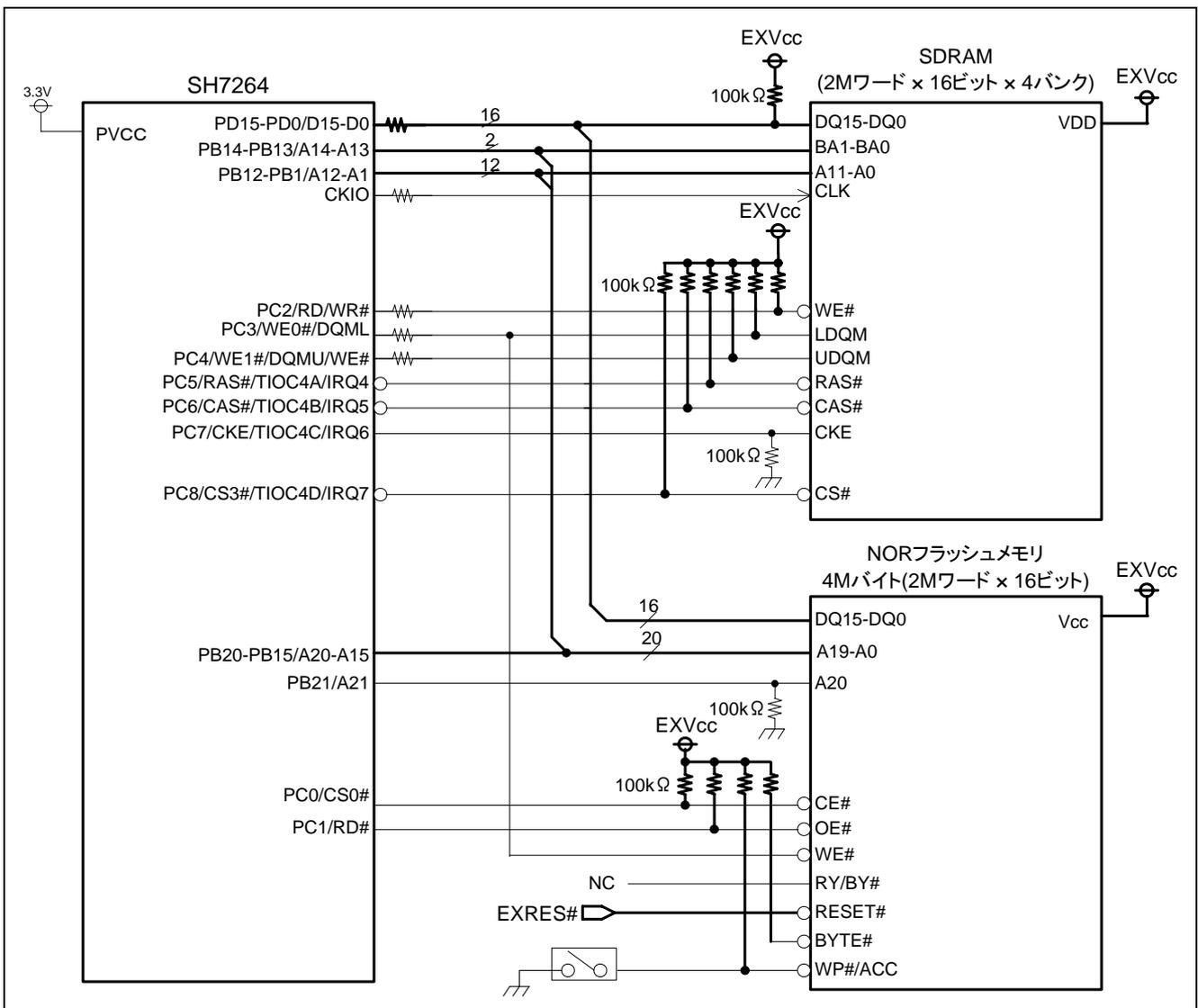


図 12 端子接続例 (外部デバイスの電源を切断する場合)

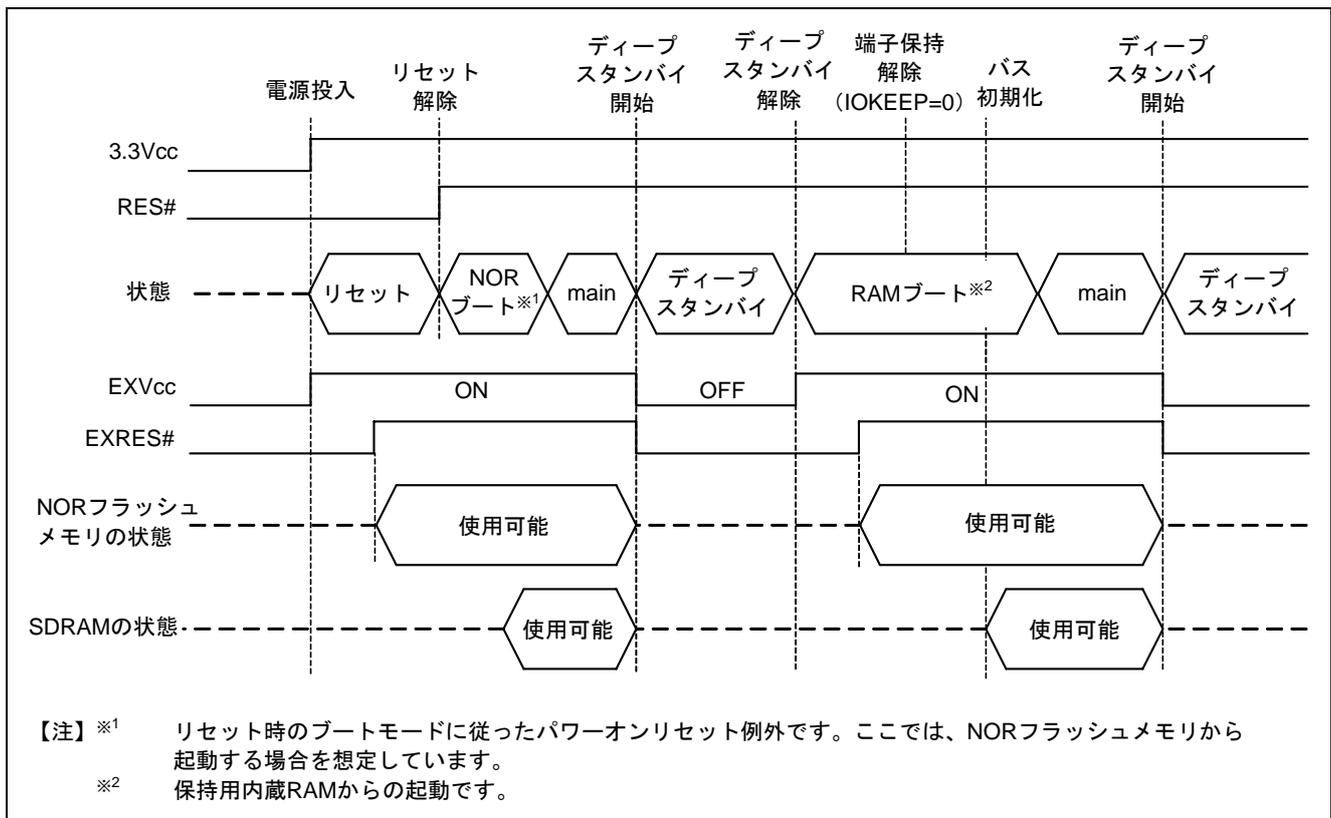


図13 電源制御のタイミングチャート例

3. 参考プログラムリスト

3.1 サンプルプログラムリスト"main.c" (1)

```
1  /*****
2  *   DISCLAIMER
3  *
4  *   This software is supplied by Renesas Electronics Corporation and is only
5  *   intended for use with Renesas products. No other uses are authorized.
6  *
7  *   This software is owned by Renesas Electronics Corporation and is protected under
8  *   all applicable laws, including copyright laws.
9  *
10 *   THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
11 *   REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
12 *   INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
13 *   PARTICULAR PURPOSE AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY
14 *   DISCLAIMED.
15 *
16 *   TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
17 *   ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
18 *   FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
19 *   FOR ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS
20 *   AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
21 *
22 *   Renesas reserves the right, without notice, to make changes to this
23 *   software and to discontinue the availability of this software.
24 *   By using this software, you agree to the additional terms and
25 *   conditions found by accessing the following link:
26 *   http://www.renesas.com/disclaimer
27 *****/
28 *   Copyright (C) 2009(2011) Renesas Electronics Corporation. All rights reserved.
29 *   "FILE COMMENT"***** Technical reference data *****
30 *   System Name : SH7264 Sample Program
31 *   File Name   : main.c
32 *   Abstract    : ディープスタンバイ処理例
33 *   Version     : 1.00.01
34 *   Device      : SH7262/SH7264
35 *   Tool-Chain  : High-performance Embedded Workshop (Ver.4.07.00).
36 *               : C/C++ compiler package for the SuperH RISC engine family
37 *               :                               (Ver.9.03 Release00).
38 *   OS          : None
39 *   H/W Platform: M3A-HS64G50(CPU board)
40 *   Description :
41 *****/
42 *   History     : Jul.28,2009 Ver.1.00.00
43 *               : Feb.23,2011 Ver.1.00.01
44 *   "FILE COMMENT END"*****/
```

3.2 サンプルプログラムリスト"main.c" (2)

```
45  #include <machine.h>
46  #include <_h_c_lib.h>
47  #include <stdio.h>
48  #include "iodefine.h"
49  #include "rtc.h"
50
51  /* ==== マクロ定義 ==== */
52  #define RETENTION_RAM_PAGE_0  0x3C0F8000 /* 1MB 版の場合のページ 0 */
53  #define RETENTION_RAM_PAGE_1  0x3C0FC000 /* 1MB 版の場合のページ 1 */
54  // #define RETENTION_RAM_PAGE_0 0x3C000000 /* 640KB 版の場合のページ 0 */
55  // #define RETENTION_RAM_PAGE_1 0x3C004000 /* 640KB 版の場合のページ 1 */
56  // #define RETENTION_RAM_PAGE_2 0x3C008000 /* 640KB 版の場合のページ 2 */
57  // #define RETENTION_RAM_PAGE_3 0x3C028000 /* 640KB 版の場合のページ 3 */
58  #define INT_OFFSET 0x10          /* 例外処理ベクタテーブルの先頭から INT_Vectors までのオフセット */
59  #define FPSCR_Init 0x00040001    /* FPSCR レジスタの初期値 */
60  #define SR_Init    0x000000F0    /* SR レジスタの初期値 */
61
62  /* ==== 関数プロトタイプ宣言 ==== */
63  void main(void);
64  void io_int_nmi( void );
65  void DeepStandby_RamBoot( void );
66  /* ---- 外部参照 ---- */
67  extern void HardwareSetup(void);
68  extern void io_cache_writeback(void);
69  extern void _INIT_IOLIB(void);
70  /* ---- ローカル関数 ---- */
71  static void io_set_pin_status( void );
72  static void print_log( void );
73  static void time_to_str( char *str, const RTC_TIME *tm );
74
75  /* ==== 変数定義 ==== */
76  extern unsigned int INT_Vectors; /* ベクタテーブル(ベクタ番号 5 以降) */
77
78  /* ---- 保持用内蔵 RAM のデータ ---- */
79  #pragma section RETENTION_AREA /* 1MB 版の場合:0x3C0F8000,640KB 版の場合:0x3C000000 */
80  unsigned long vecttbl_dpstby[2] = { /* ディープスタンバイモード解除時に使用するベクタ領域 */
81      (unsigned long)DeepStandby_RamBoot,
82      (unsigned long)(__secend("S")),
83  };
84  static RTC_TIME stbyin_time; /* ディープスタンバイモード遷移時の時刻データ */
85  #pragma section
```

3.3 サンプルプログラムリスト"main.c" (3)

```
86  /*"FUNC COMMENT"*****
87  * ID      :
88  * Outline  : メイン関数
89  *-----
90  * Include   : stdio.h, rtc.h
91  *-----
92  * Declaration : void main(void);
93  *-----
94  * Description : リアルタイムクロックの時刻データをターミナルに表示します。
95  *-----
96  * Argument   : void
97  *-----
98  * Return Value : void
99  *-----
100 * Note       : None
101 *"FUNC COMMENT END"*****/
102 void main(void)
103 {
104     static RTC_TIME tm;
105     unsigned char old_sec = 0xFF;
106     char str[9];
107
108     while(1){
109         /* ---- リアルタイムクロックからの時刻読み出し ---- */
110         io_read_rtc( &tm );
111         /* ---- 時刻更新あり ---- */
112         if( tm.sec != old_sec ){
113             old_sec= tm.sec;
114             /* ---- 文字列へ変換 ---- */
115             time_to_str( str, &tm );
116             /* ---- 時刻をターミナルに表示 ---- */
117             printf( "RTC time is[%s]¥n", str );
118             fflush( stdout );
119         }
120     }
121 }
```

3.4 サンプルプログラムリスト"main.c" (4)

```

122  /*"FUNC COMMENT"*****
123  * ID      :
124  * Outline : NMI 割り込み
125  *-----
126  * Include : iodef.h
127  *-----
128  * Declaration : void io_int_nmi( void );
129  *-----
130  * Description : スタンバイ開始時刻を保持用内蔵 RAM に保存し、不要な電流を流さない
131  *              : ための端子処理をした後、ディープスタンバイモードへ遷移します。
132  *-----
133  * Argument  : void
134  *-----
135  * Return Value : void
136  *-----
137  * Note      : None
138  *"FUNC COMMENT END"*****/
139 void io_int_nmi( void )
140 {
141     static RTC_TIME tm;
142     volatile int dummy;
143     int *p;
144
145     /* ---- NMI 端子と PJ3 (IRQ1) 端子が H レベルになるまで待つ ---- */
146     while( INTC.ICR0.BIT.NMIL == 0 ){
147         /* wait */
148     }
149     while( PORT.PJPR0.BIT.PJ3PR == 0 ){
150         /* wait */
151     }
152     /* ---- ディープスタンバイコントロールレジスタ (DSCTR) の設定 ---- */
153     CPG.DSCTR.BYTE = 0xC0u;          /* EBUSKEEPE ビット=1 : 外部メモリ制御端子の状態を保持 */
154                                     /* RAMBOOT ビット =1 : 保持用内蔵 RAM から起動 */
155     /* ---- ディープスタンバイ解除要因セレクトレジスタ (DSSSR) の設定 ---- */
156     CPG.DSSSR.WORD = 0x0002u;      /* PJ3 (IRQ1) を解除要因に指定 */
157
158     /* ---- ディープスタンバイ解除要因エッジセレクトレジスタ (DSESR) の設定 ---- */
159     CPG.DSESR.WORD = 0x0000u;      /* PJ3 (IRQ1) または NMI の解除要因は、立ち下がりエッジに指定 */
160
161     /* ---- リアルタイムクロックから時刻を読み出し、保持用内蔵 RAM へ保存 ---- */
162     io_read_rtc( &tm );
163     stbyin_time = tm;
164
165     /* ---- 保持用内蔵 RAM エリア指定レジスタ (RRAMKP) の設定 ---- */
166     CPG.RRAMKP.BYTE = 0x01u;       /* ページ 0 のみ保持 */
167
168     /* ---- 保持用内蔵 RAM の書き込みを保證するためのダミーリードライト ---- */
169     p = (int *)RETENTION_RAM_PAGE_0;
170     dummy = *p;
171     *p = dummy;

```

3.5 サンプルプログラムリスト"main.c" (5)

```
172     /* ---- 端子状態の設定 ---- */
173     io_set_pin_status();
174
175     /* ---- スタンバイコントローラレジスタ 1 (STBCR1) の設定 ---- */
176     CPG.STBCR1.BYTE = 0xC0u;          /* SLEEP 命令で、ディープスタンバイモードへ遷移 */
177
178     /* ---- ディープスタンバイ解除要因フラグレジスタ (DSFR) のフラグクリア ---- */
179     CPG.DSFR.WORD &= 0x0000u;       /* 解除要因フラグを全てクリア (1 の読み出し後にクリア) */
180     dummy = CPG.DSFR.WORD;         /* 後続の命令実行前に設定を有効化するためのダミーリード */
181
182     /* ---- ディープスタンバイへ遷移 ---- */
183     sleep();
184
185     nop();
186     nop();
187     nop();
188     nop();
189 }
190 /* "FUNC COMMENT" "*****"
191  * ID          :
192  * Outline     : 保持用内蔵 RAM 起動用ブートプログラム
193  *-----
194  * Include     : stdio.h, machine.h, _h_c_lib.h
195  *-----
196  * Declaration : void DeepStandby_RamBoot( void );
197  *-----
198  * Description : ディープスタンバイモード解除時に本モジュールが実行されます。
199  *              : 端子状態保持の解除、ハードウェアの初期化を行った後、起動履歴を
200  *              : 標準出力に出力します。
201  *              : 本モジュールは保持用内蔵 RAM の保持対象ページに配置してください。
202  *-----
203  * Argument    : void
204  *-----
205  * Return Value : void
206  *-----
207  * Note        : None
208  * "FUNC COMMENT END" "*****"/>
```

3.6 サンプルプログラムリスト"main.c" (6)

```
209 #pragma section RETENTION_AREA
210 void DeepStandby_RamBoot( void )
211 {
212     unsigned short reg;
213
214     /* ---- ディープスタンバイ解除要因フラグレジスタ (DSFR) のチェック ---- */
215     reg = CPG.DSFR.WORD;
216     if( (reg & 0x0102u) == 0x0000u ){ /* PJ3 または NMI 割り込みによる解除でなければエラー*/
217         while(1){
218             /* error */
219         }
220     }
221     /* ---- 浮動小数点ステータス/コントロールレジスタ (FPSCR)の設定 ---- */
222     set_fpscr( FPSCR_Init ); /* 単精度演算、0 方向への丸め */
223
224     /* ---- 端子状態保持を解除 ---- */
225     CPG.DSFR.BIT.IOKEEP = 0; /* 1 リード後に 0 ライトすること */
226
227     /* ---- ハードウェアの初期化 ---- */
228     HardwareSetup();
229
230     /* ---- データの初期化 ---- */
231     _INITSCT();
232     io_cache_writeback();
233
234     /* ---- ベクタベースレジスタ (VBR) の設定 ====*/
235     set_vbr((void *)((char *)&INT_Vectors - INT_OFFSET));
236
237     /* ---- 標準入出力の初期化 ---- */
238     _INIT_IOLIB();
239
240     /* ---- ステータスレジスタ (SR) の設定 ---- */
241     set_cr(SR_Init);
242     nop();
243
244     /* ---- バンク番号レジスタ (IBNR) の設定 ---- */
245     INTC.IBNR.BIT.BE = 0x01; /* NMI 以外の全ての割り込みでバンクの使用を許可 */
246
247     /* ---- 割り込み許可 ---- */
248     set_imask(0);
249
250     /* ---- 起動履歴を標準出力に出力 ---- */
251     print_log();
252
253     /* ---- メイン関数へ分岐 ---- */
254     main();
255     sleep();
256 }
257 #pragma section
258
```

3.7 サンプルプログラムリスト"main.c" (7)

```
259  /*"FUNC COMMENT"*****
260  * ID      :
261  * Outline  : 端子状態の設定
262  *-----
263  * Include  : iodef.h
264  *-----
265  * Declaration : static void io_set_pin_status( void );
266  *-----
267  * Description : ディープスタンバイモード時の端子状態を設定します。
268  *-----
269  * Argument   : void
270  *-----
271  * Return Value : void
272  *-----
273  * Note       : None
274  *"FUNC COMMENT END"*****/
275  static void io_set_pin_status( void )
276  {
277      /* ---- 周波数制御レジスタ (FRQCR) の設定 ---- */
278      CPG.FRQCR.BIT.CKOEN = 0; /* ディープスタンバイモード時の CKIO 端子出力を L レベル
279                               * または H レベル出力に設定
280                               */
281      /* ---- スタンバイコントロールレジスタ 3 (STBCR3) の設定 ---- */
282      CPG.STBCR3.BIT.HIZ = 0; /* ディープスタンバイモード時の出力端子を Hi-Z ではなく
283                               * 端子状態保持に設定します。
284                               */
285      /* ---- 共通コントロールレジスタ (CMNCR) の設定 ---- */
286      BSC.CMNCR.BIT.HIZMEM = 1; /* 外部メモリ制御端子をドライブ状態に設定 */
287      BSC.CMNCR.BIT.HIZCNT = 1; /* SDRAM 用の外部メモリ制御端子をドライブ状態に設定 */
288
289      /* ---- プルアップ/プルダウン抵抗への不要電流対策 ---- */
290      /* A21 */
291      PORT.PBIOR1.BIT.PB21IOR = 0; /* PB21 の方向を入力に設定 */
292      PORT.PBCR5.BIT.PB21MD = 0; /* PB21/A21 の端子機能を汎用入出力ポートに設定 */
293  }
294
```

3.8 サンプルプログラムリスト"main.c" (8)

```
295  /*"FUNC COMMENT"*****
296  * ID      :
297  * Outline  : 起動履歴の出力
298  *-----
299  * Include  :  stdio.h, rtc.h
300  *-----
301  * Declaration :  static void print_log( void );
302  *-----
303  * Description :  ディープスタンバイモード遷移時と解除時の時刻データを標準出力に
304  *              :  出力します。
305  *-----
306  * Argument  :  void
307  *-----
308  * Return Value :  void
309  *-----
310  * Note      :  None
311  *"FUNC COMMENT END"*****/
312  static void print_log( void )
313  {
314      RTC_TIME tm;
315      char str1[9], str2[9];
316
317      /* ---- ディープスタンバイモード遷移時の時刻 ---- */
318      time_to_str( str1, &stbyin_time );
319
320      /* ---- ディープスタンバイモード解除時の時刻 ---- */
321      io_read_rtc( &tm );
322      time_to_str( str2, &tm );
323
324      /* ---- 時刻を標準出力に出力 ---- */
325      printf( "Deep standby time¥n" );
326      printf( "    => From %s¥n", str1 );
327      printf( "    => To   %s¥n", str2 );
328      fflush( stdout );
329  }
```

3.9 サンプルプログラムリスト"main.c" (9)

```
330  /*"FUNC COMMENT"*****
331  * ID          :
332  * Outline     : 時刻データの文字列変換
333  *-----
334  * Include     : rtc.h
335  *-----
336  * Declaration : static void time_to_str( char *str, const RTC_TIME tm );
337  *-----
338  * Description : 引数で指定された時刻データを文字列("hh:mm:ss")に変換します。
339  *             : str は 9 バイト以上の空き領域を確保してください。
340  *-----
341  * Argument    : char *str          ; O : 文字列を格納するバッファアドレス
342  *             : const RTC_TIME tm ; I : 時刻データ
343  *-----
344  * Return Value : void
345  *-----
346  * Note        : None
347  *"FUNC COMMENT END"*****/
348  static void time_to_str( char *str, const RTC_TIME *tm )
349  {
350      *str++ = (( tm->hour >> 4 ) & 0xF) + '0';
351      *str++ = (( tm->hour      ) & 0xF) + '0';
352      *str++ = ':';
353      *str++ = (( tm->min >> 4 ) & 0xF) + '0';
354      *str++ = (( tm->min      ) & 0xF) + '0';
355      *str++ = ':';
356      *str++ = (( tm->sec >> 4 ) & 0xF) + '0';
357      *str++ = (( tm->sec      ) & 0xF) + '0';
358      *str  = '\0';
359  }
360  /* End of File */
361
```

3.10 サンプルプログラムリスト"rtc.c" (1)

```
1  /*****
2  *  DISCLAIMER
3  *
4  *  This software is supplied by Renesas Technology Corp. and is only
5  *  intended for use with Renesas products. No other uses are authorized.
6  *
7  *  This software is owned by Renesas Technology Corp. and is protected under
8  *  all applicable laws, including copyright laws.
9  *
10 *  THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
11 *  REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
12 *  INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
13 *  PARTICULAR PURPOSE AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY
14 *  DISCLAIMED.
15 *
16 *  TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
17 *  TECHNOLOGY CORP. NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
18 *  FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
19 *  FOR ANY REASON RELATED TO THE THIS SOFTWARE, EVEN IF RENESAS OR ITS
20 *  AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
21 *
22 *  Renesas reserves the right, without notice, to make changes to this
23 *  software and to discontinue the availability of this software.
24 *  By using this software, you agree to the additional terms and
25 *  conditions found by accessing the following link:
26 *  http://www.renesas.com/disclaimer
27 *****/
28 *  Copyright (C) 2009. Renesas Technology Corp., All Rights Reserved.
29 *  "FILE COMMENT"***** Technical reference data *****
30 *  System Name : SH7264 Sample Program
31 *  File Name   : rtc.c
32 *  Abstract    : リアルタイムクロックの制御
33 *  Version     : 1.00.00
34 *  Device      : SH7262/SH7264
35 *  Tool-Chain  : High-performance Embedded Workshop (Ver.4.04.01).
36 *              : C/C++ compiler package for the SuperH RISC engine family
37 *              :                               (Ver.9.02 Release00).
38 *  OS          : None
39 *  H/W Platform: M3A-HS64G50(CPU board)
40 *  Description :
41 *****/
42 *  History     : Jul.27,2009 Ver.1.00.00
43 *  "FILE COMMENT END"*****/
44 #include "iodefine.h"
45 #include "rtc.h"
```

3.11 サンプルプログラムリスト"rtc.c" (2)

```

46  /* ==== マクロ定義 ==== */
47  #define RYRCNT_INI      0x2009 /* 2009年 */
48  #define RMONCNT_INI    0x01 /* 1月 */
49  #define RDAYCNT_INI    0x01 /* 1日 */
50  #define RWKCNT_INI     0x1 /* 月曜日 */
51  #define RHRCNT_INI     0x00 /* 0時 */
52  #define RMINCNT_INI    0x00 /* 0分 */
53  #define RSECCNT_INI    0x00 /* 0秒 */
54
55  /* ==== 変数定義 ==== */
56  RTC_TIME tm_init = {
57      RYRCNT_INI, RMONCNT_INI, RDAYCNT_INI, RWKCNT_INI, RHRCNT_INI, RMINCNT_INI, RSECCNT_INI,
58  };
59  /* "FUNC COMMENT"*****
60  * ID          :
61  * Outline     : リアルタイムクロックの初期設定
62  *-----
63  * Include     : iodef.h, rtc.h
64  *-----
65  * Declaration : void io_init_rtc( void );
66  *-----
67  * Description : リアルタイムクロックを初期化します。
68  *-----
69  * Argument    : void
70  *-----
71  * Return Value : void
72  *-----
73  * Note        : None
74  * "FUNC COMMENT END"*****/
75  void io_init_rtc( void )
76  {
77      /* ---- カウンタ動作の停止 ---- */
78      RTC.RCR2.BIT.START = 0;
79
80      /* ---- コントロールレジスタ 5 (RCR5) の設定 ---- */
81      RTC.RCR5.BIT.RCKSEL = 0; /* 動作クロックを RTC_X1 に設定します。*/
82
83      /* ---- コントロールレジスタ 2 (RCR2) の設定 ---- */
84      RTC.RCR2.BIT.RESET = 1; /* 分周回路および R64CNT レジスタ、アラームレジスタ、
85                               * RCR3 レジスタ、RCR1 レジスタの CF、AF ビット、RCR2 レジスタ
86                               * の PEF ビットを初期化します。カウンタ動作を停止します。*/
87
88      /* ---- カウンタの設定 ---- */
89      RTC.RSECCNT = tm_init.sec;
90      RTC.RMINCNT = tm_init.min;
91      RTC.RHRCNT = tm_init.hour;
92      RTC.RWKCNT = tm_init.week;
93      RTC.RDAYCNT = tm_init.day;
94      RTC.RMONCNT = tm_init.mon;
95      RTC.RYRCNT = tm_init.year;
96
97      /* ---- カウンタ動作の再開 ---- */
98      RTC.RCR2.BIT.START = 1;
99  }

```

3.12 サンプルプログラムリスト"rtc.c" (3)

```
99  /*"FUNC COMMENT"*****
100  * ID      :
101  * Outline : リアルタイムクロックの時刻読み出し
102  *-----
103  * Include : iodef.h, rtc.h
104  *-----
105  * Declaration : void io_read_rtc( RTC_TIME *time );
106  *-----
107  * Description : リアルタイムクロックのカウンタレジスタの値を引数で指定した領域に
108  *              : 読み出します。
109  *-----
110  * Argument   : RTC_TIME *time : 0 : カウンタレジスタの値を格納する領域
111  *-----
112  * Return Value : void
113  *-----
114  * Note       : None
115  *"FUNC COMMENT END"*****/
116 void io_read_rtc( RTC_TIME *time )
117 {
118     /* ---- 桁上げ割り込みディスエーブル ---- */
119     RTC.RCR1.BIT.CIE = 0;
120
121     do{
122         /* ---- 桁上げフラグクリア ---- */
123         RTC.RCR1.BIT.CF = 0;
124
125         /* ---- カウンタレジスタ読み出し ---- */
126         time->sec = RTC.RSECNT;
127         time->min = RTC.RMINCNT;
128         time->hour = RTC.RHRCNT;
129         time->week = RTC.RWKCNT;
130         time->day = RTC.RDAYCNT;
131         time->mon = RTC.RMONCNT;
132         time->year = RTC.RYRCNT;
133
134     }while( RTC.RCR1.BIT.CF == 1 ); /* 桁上げフラグ = 1 の場合は再読み出し */
135 }
136 /* End of File */
137
```

3.13 サンプルプログラムリスト"rtc.h" (1)

```
1  /*****
2  *  DISCLAIMER
3  *
4  *  This software is supplied by Renesas Technology Corp. and is only
5  *  intended for use with Renesas products. No other uses are authorized.
6  *
7  *  This software is owned by Renesas Technology Corp. and is protected under
8  *  all applicable laws, including copyright laws.
9  *
10 *  THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
11 *  REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
12 *  INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
13 *  PARTICULAR PURPOSE AND NON-INFRINGEMENT. ALL SUCH WARRANTIES ARE EXPRESSLY
14 *  DISCLAIMED.
15 *
16 *  TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
17 *  TECHNOLOGY CORP. NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
18 *  FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
19 *  FOR ANY REASON RELATED TO THE THIS SOFTWARE, EVEN IF RENESAS OR ITS
20 *  AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
21 *
22 *  Renesas reserves the right, without notice, to make changes to this
23 *  software and to discontinue the availability of this software.
24 *  By using this software, you agree to the additional terms and
25 *  conditions found by accessing the following link:
26 *  http://www.renesas.com/disclaimer
27 *****/
28 *  Copyright (C) 2009. Renesas Technology Corp., All Rights Reserved.
29 *  "FILE COMMENT"***** Technical reference data *****
30 *  System Name : SH7264 Sample Program
31 *  File Name   : rtc.h
32 *  Abstract    : リアルタイムクロックの制御
33 *  Version     : 1.00.00
34 *  Device      : SH7262/SH7264
35 *  Tool-Chain  : High-performance Embedded Workshop (Ver.4.04.01).
36 *              : C/C++ compiler package for the SuperH RISC engine family
37 *              :                      (Ver.9.02 Release00).
38 *  OS          : None
39 *  H/W Platform: M3A-HS64G50(CPU board)
40 *  Description :
41 *****/
42 *  History     : Jul.27,2009 Ver.1.00.00
43 *  "FILE COMMENT END"*****/
```

3.14 サンプルプログラムリスト"rtc.h" (2)

```
44  #ifndef _RTC_H_
45  #define _RTC_H_
46
47  /* ==== マクロ定義 ==== */
48  typedef struct tag_trc{
49      unsigned short year;
50      unsigned char  mon;
51      unsigned char  day;
52      unsigned char  week;
53      unsigned char  hour;
54      unsigned char  min;
55      unsigned char  sec;
56  }RTC_TIME;
57
58  /* ==== 関数プロトタイプ宣言 ==== */
59  void io_init_rtc( void );
60  void io_read_rtc( RTC_TIME *time );
61
62  #endif /* _RTC_H_ */
63  /* End of File */
```

4. 参考ドキュメント

- ソフトウェアマニュアル
SH-2A、SH2A-FPU ソフトウェアマニュアル Rev.3.00
(最新版をルネサスエレクトロニクスのホームページから入手してください。)
- ハードウェアマニュアル
SH7262 グループ、SH7264 グループ ハードウェアマニュアル Rev.2.00
(最新版をルネサスエレクトロニクスの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/>

お問い合わせ先

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

すべての商標および登録商標は、それぞれの所有者に帰属します。

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2009.09.24	—	初版発行
1.01	2011.02.23	6、15、26	誤記修正

製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本文を参照してください。なお、本マニュアルの本文と異なる記載がある場合は、本文の記載が優先するものとします。

1. 未使用端子の処理

【注意】未使用端子は、本文の「未使用端子の処理」に従って処理してください。

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

2. 電源投入時の処置

【注意】電源投入時は、製品の状態は不定です。

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

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

3. リザーブアドレスのアクセス禁止

【注意】リザーブアドレスのアクセスを禁止します。

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

4. クロックについて

【注意】リセット時は、クロックが安定した後、リセットを解除してください。

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

5. 製品間の相違について

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

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

ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連して発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続きを行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。
標準水準： コンピュータ、OA機器、通信機器、計測機器、AV機器、家電、工作機械、パーソナル機器、産業用ロボット
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）
特定水準： 航空機器、航空宇宙機器、海中継機器、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、当社は、一切その責任を負いません。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

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

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



ルネサスエレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

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

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

(03)5201-5307

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