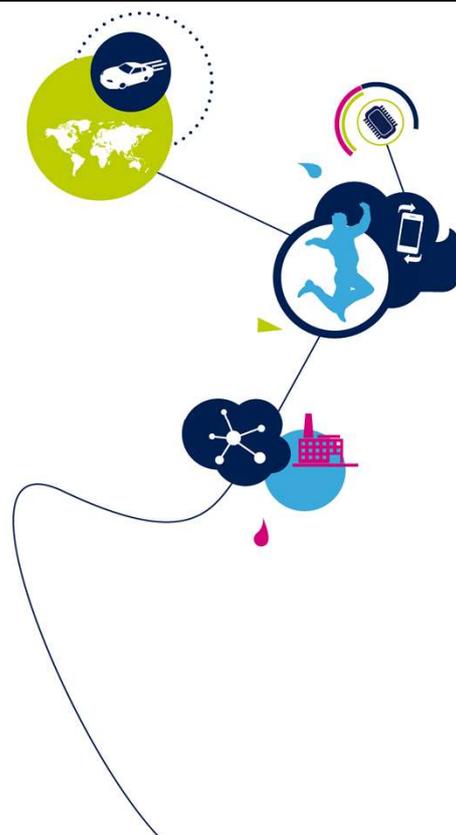
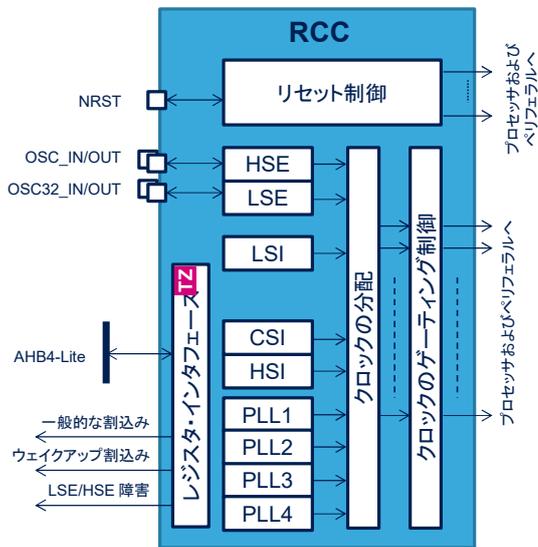


STM32MP1 - RCC

リセットおよびクロック制御
1.0 版



こんにちは、STM32MP1 リセットおよびクロック制御(RCC)のプレゼンテーションへようこそ。
RCC は、電源コントローラブロックと密接に連携する複雑なブロックです。
よって、このスライドをご覧になる前に、電源制御モジュールのプレゼンテーションをご覧ください。



- STM32MP1 リセットおよびクロック制御(RCC)は下記を管理：
 - すべてのクロックの生成
 - PLL、RC発振回路、水晶発振器 ...
 - すべてのクロックのゲーティング
 - すべてのシステムとペリフェラルのリセットの制御

アプリケーション側の利点

- 消費電力と精度の要件を満たすためのクロックソース選択の高い柔軟性
- 安全で柔軟なリセット管理



RCC は次のいくつかのクロックソースを提供します。

- 3つの内部RC発振回路
- 外部の水晶または発振子を使用する2つの発振回路
- 4つのPLL(位相ロックループ)

クロック分配ブロックは、これらすべてのクロック・ソースを使用して、ペリフェラル、バス相互接続、およびプロセッサに柔軟なクロック・セットを提供します。

クロック・ゲーティング制御ブロックは、電力消費を最適化するためにクロックのゲーティングの役割を担っています。

多くのペリフェラルには、バス・インタフェース・クロックから独立した独自のクロックがあり、ペリフェラル・インタフェース速度に影響を与えずにバス・インタフェース・クロックを動的に変更できます。

STM32MP1のRCCでは、クロック・ソースの選択に高い柔軟性があり、アプリケーションは消費電力と精度の両方の要件を満たすことができます。

さらに、RCCにはリセット制御ブロックも組み込まれており、システム全体のリセットを生成します。

最後に、TrustZoneに対応したRCCのレジスタ・インタフェースによって、アプリケーションはRCCの各機能を適切に制御できます。

RCC は、レジスタを介して完全に制御することが可能

RCC は TrustZone 対応

レジスタの属性:

- 常にセキュア
- セキュリティ保護可能 (TZEN および MCKPROT による)
- 非セキュア

さらに、一部のレジスタは MPU のみが変更可能

主なアクセス規則:

- すべてのレジスタは制約なしに読出し可能
- セキュアなレジスタは、セキュア書込みアクセスによってのみ変更可能
- セキュア・レジスタへの非セキュア書込みは、ハードフォールトを生成
- 非セキュア・レジスタは、セキュアまたは非セキュア書込みアクセスで変更可能
- MCU による「MPU-ONLY」レジスタへの書込みは不可
- MCU による「MPU-ONLY」レジスタの読出しは可能



RCC はレジスタを介して完全に制御することができます。

RCC には TrustZone 機能があり、次の 3 種類のセキュア属性を扱えます。

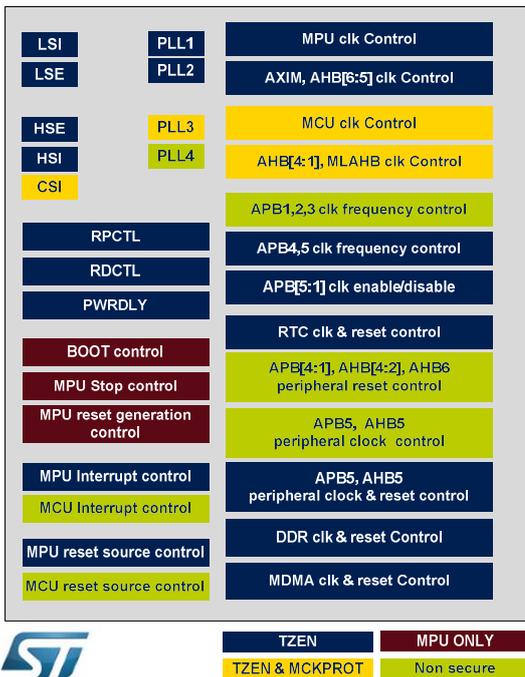
- 常にセキュアなレジスタ: これらのレジスタは、書込みセキュア・アクセスを実行することによってのみ変更できます。
- セキュリティ保護可能レジスタ: これらのレジスタは、常にセキュアなレジスタにある TZEN および MCKPROT の 2 つのビットを介して、セキュア・モードまたは非セキュア・モードに切り替えることができます。
TZEN および MCKPROT ビットを使用すると、アプリケーションでセキュリティ境界を制御できます。これについては、次のスライドで詳しく説明します。
- 非セキュア・レジスタ: これらのレジスタは、セキュアおよび非セキュア書込みアクセスで変更できます。

TrustZone に加えて、RCC の一部のレジスタは MPU のみ書き込むことができます。これは、MPU 専用のサービスを提供するレジスタの場合です。

これらのレジスタも TrustZone の影響を受けるため、MPU によって実行されるセキュア書込みアクセスのみを許可するレジスタを設けることができます。

RCC には以下のアクセス規則が適用されます。

- すべてのレジスタは制約なしに読み出すことができます。MCU も、MPU 専用のレジスタを読み出すことができます。
- セキュアなレジスタは、セキュア書込みによってのみ変更できます。
- セキュア・レジスタへ非セキュア書込みをしようとすると、ハード・フォールトが発生します。
- 非セキュア・レジスタは、セキュアまたは非セキュア書込みアクセスで変更できます。
- MCU は「MPU-ONLY」レジスタへ書き込むことはできません。



- TrustZone によって、非セキュア・アプリケーションがリセットを実行することや、セキュア・アプリケーションで使用されるペリフェラルを無効にすることが防げます。
- TZEN = 1 の場合、紺色の項目はセキュア・アクセスを介してのみ制御できます。
- TZEN = 1 かつ MCKPROT = 1 の場合、安全な境界は黄色の項目まで拡張されます。

アプリケーション側の利点

TZEN = 1 の場合の目的:

- APB5 と AHB5 にあるすべてのペリフェラルのクロックとリセット機能の保護
- DDR、MDMA、および要求されたすべての相互接続バスのクロックとリセット機能の保護

RCC レベルでは、TrustZone 機能により、非セキュア・アプリケーションがリセットを実行したり、セキュア・アプリケーションで使用されているペリフェラルを無効にしたりすることが防げます。

この図は、TrustZone の境界の簡略図を示しています。

RCC は、次の 3 つの異なる保護モードにすることができます。

最初のモードは非セキュアモード (TZEN = 0) です。この場合、すべてのレジスタは非セキュアな方法でアクセスできます。一部のレジスタは、MPU 以外のマスタのアクセスから保護されています。

2 番目のモードは、TZEN = 1 かつ MCKPROT = 0 のセキュア・モードです。この場合、APB5 および AHB5 ドメインにあるペリフェラルは、セキュアモードでのみ使用できます。DDR、および MDMA、さらに必要なすべての相互接続バス、およびクロック・ジェネレータも保護されます。この設定では、セキュア・ペリフェラルはセキュア・エレメントによって提供されるクロックを使用する必要があります。

3 番目のモードは、TZEN = 1 かつ MCKPROT = 1 のセキュア・モードです。このモードは前のモードと似ていますが、MCU サブシステムも保護するために境界が拡大されています。

外部コンポーネント不要の安全で柔軟なリセット管理

- RCC は以下の数種類のリセットを生成します。
 - アプリケーション・リセット (app_rst)
 - システム・リセット (nreset)
 - バックアップ・ドメイン・リセット (nreset_vsw)
 - ローカル・リセット (MPU および MCU リセット)

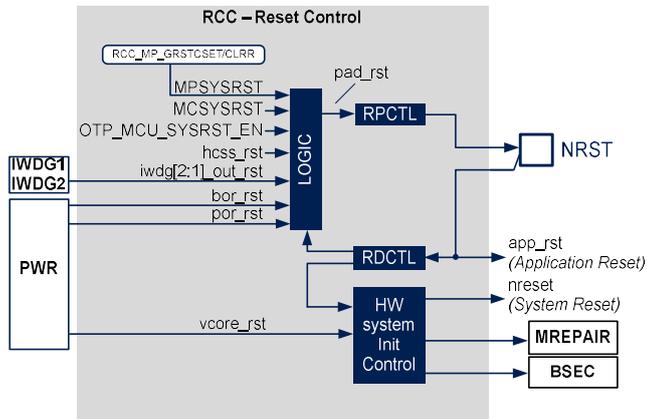


次に、RCC のリセット部分に注目しましょう。

RCC では、アプリケーション・コストを削減するために外部コンポーネントを必要としない、安全で柔軟なリセット管理の豊富な機能を提供しています。

RCC では、パワーオン・リセット、システム・リセット、ローカル・リセット、バックアップ・ドメイン・リセットなど、いくつかの種類のリセットを管理しています。

内部フィルタと電力監視により、外部コンポーネントは不要 システム・リセット・ソースは外部コンポーネントをリセット可能



- アプリケーション・リセット・ソース：
 - VDD の無効な電圧 (por_rst、bor_rst)
 - NRST ピンでのローレベル (外部リセット)
 - IWDG[2:1] タイムアウト・イベント
 - MCU によって開始されたソフトウェア・リセット (許可されている場合)
 - MPU によって開始されたソフトウェア・リセット
 - HSE 発振回路の障害
- システム・リセット・ソース：
 - アプリケーション・リセットのアサート
 - VDDCORE の無効な電圧



PWR ブロックに含まれた電圧監視、NRST PAD に組み込まれたフィルタ、および RCC リセット制御によって、外部コンポーネントの量が大幅に削減されています。

一つ目のリセットは、アプリケーション・リセットです。これは、バックアップ・ドメインにあるレジスタを除くほとんどのレジスタをリセットします。

以下の多くのソースでアプリケーション・リセットを生成することができます。

VDD 電源の無効な電圧 (詳細は PWR ブロックを参照)、
 ブラウンアウト機能による VDD の無効な電圧。ブラウンアウト機能により、VDD 電源の独自の閾値レベルを選択できます (詳細は PWR ブロックを参照)、
 NRST パッドでのローレベル、
 独立型ウォッチドッグの 1 つからのタイムアウト (IWDG2、および 1)、
 Cortex® M4 (MCU) によって開始されたソフトウェア・リセット・リクエスト (オプションバイトで許可されている場合)、
 Cortex A7 (MPU) によって開始されたソフトウェア・リセット・リクエスト (RCC レジスタ経由)、
 HSE 発振回路の障害。

アプリケーション・リセットは PAD NRST をアサートし、外部コンポーネントのリセットを可能にすることに注意してください。

システム・リセットは以下の場合に生成されます。

アプリケーション・リセットがアサートされる。

無効な VDDCORE が検出された場合 (vcore_rst)、この状況は、電源投入シーケンスの後、またはチップが STANDBY モードを終了したときに現れます。

オプションバイトの設定とメモリ修復シーケンスは、無効な VDDCORE 電圧が検出されるたびに、またはアプリケーション・リセットが発生したときにトリガされます。

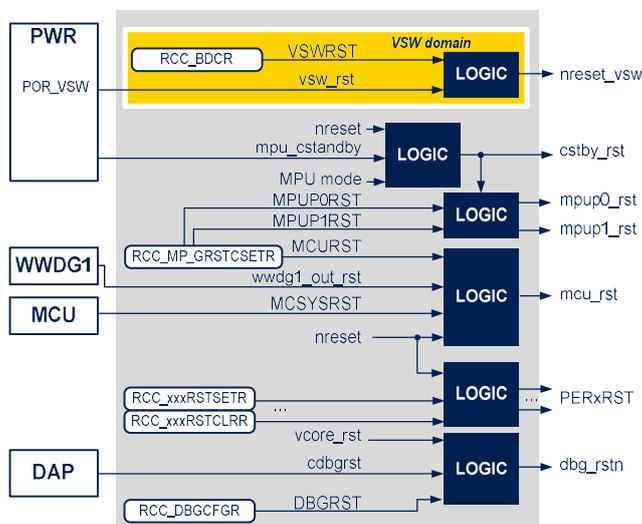
RPCTL (リセットパルス制御) という名前のブロックによって、アプリケーションで NRST パッドの最小アクティベーション時間を定義できます。

RDCTL (リセット遅延制御) という名前のブロックによって、アプリケーションは、DDR がセルフ・リフレッシュ・モードになるまでシステム・リセットのアクティベーションを遅らせることができます。

これら 2 つのブロックは後で説明します。

リセット・ソース (2/3)

7



- 下記によって生成されるバックアップ・ドメイン・リセット (nreset_vsw)
 - RCC_BDCR レジスタの VSWRST ビット
 - VDD または VBAT パワーオン。ただし、両方の電源供給がともにオフ状態であった場合
- 下記によって生成される CStandby リセット (cstby_rst)
 - nreset のアサート(システム・リセット)
 - CStandby の終了
- 下記によって生成される MPU リセット (mpup[1:0]_rst)
 - レジスタの MPUP[1:0]RST ビット
 - nreset のアサート(システム・リセット)
 - CStandby モードの終了。ただし、PWR で許可されている場合



バックアップ・ドメインのリセットは以下の場合に発生します。
VSWRST ビットがアプリケーションによって“1”にセットされた場合。図に示したように、このビットは RCC_BDCR レジスタに配置されています。

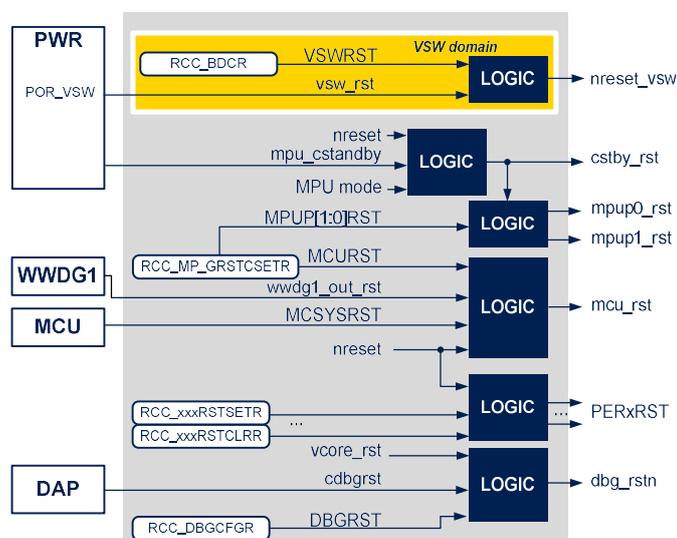
VDD および VBAT の両方がパワーオンになった場合(ただし両方の電源がともにオフ状態であった場合)。

MPU リセットは以下の場合に実施されます。

MPU が MPUPxRST ビットを介してそのプロセッサの 1 つをリセットした場合。1 つのプロセッサは他方のプロセッサに影響を与えることなくリセットできます。スヌープ制御ユニットはリセットされません。

システム・リセットが発生した場合。

システムが CStandby モードを終了した場合(ただし、PWR 管理によって許可されている場合)。



- 下記によって生成される MCU リセット (mcu_rst)
 - レジスタの MCURST ビット
 - nreset のアサート (システム・リセット)
 - WWDG1 リセット (ウィンドウ外またはタイムアウト)
 - MCU システム・リセット (MCSYSRST)
- 下記によって生成されるペリフェラル・リセット
 - RCC レジスタの PERxRST ビット
 - nreset のアサート (システム・リセット)
- 下記によって生成される CoreSight デバッグ・リセット
 - RCC レジスタの DBGRST ビット
 - DAP からのリクエスト
 - VDDCORE が無効 (vcore_rst)

MCU リセットは以下の場合に実施されます。
 MPU が MCURST ビットをアサートした場合、
 システム・リセットが発生した場合、
 ウィンドウ型ウォッチドッグが適切なタイミングで再ロードされなかった場合、
 MCU がシステム・リセットをリクエストした場合。システムのリセットはオプション・バイトでブロックできますが、MCU のリセットは行われます。

ペリフェラルは、RCC レジスタのそれぞれのリセット・ビットまたはシステム・リセットのいずれかによってリセットされます。

デバッグ部分は以下の場合にリセットされます。
 DBGRST ビットがアサートされた場合、
 DAP のリクエスト時、
 VDDCORE が無効な場合。

リセット部分	por_rst	vcore_rst	app_rst	nreset	cstby_rst	mpup1_rst	mpup0_rst	mcu_rst	dbg_rst	PERxRST	nreset_vsw
VDD ドメイン	✓	-	-	-	-	-	-	-	-	-	-
DDR インタフェース	✓	✓	✓	✓	✓	-	-	-	-	-	-
MPU プロセッサ 0	✓	✓	✓	✓	✓	-	✓	-	-	-	-
MPU プロセッサ 1	✓	✓	✓	✓	✓	✓	-	-	-	-	-
MCU および WWWDG1	✓	✓	✓	✓	-	-	-	✓	-	-	-
AXI/AHB ブリッジ	✓	✓	✓	✓	-	-	-	-	-	-	-
OTP 再ロード	✓	✓	✓	✓	-	-	-	-	-	-	-
デバッグ・コンポーネント	✓	✓	✓	-	-	-	-	-	✓	-	-
IWDG[2:1]	✓	-	✓	-	-	-	-	-	-	-	-
RCC および PWR の一部	✓	✓	✓	✓	✓	-	-	-	-	-	✓
RTC、BKPSRAM、および RETRAM	-	-	-	-	-	-	-	-	-	-	✓
ペリフェラル	✓	✓	✓	✓	-	-	-	-	-	✓	-



この表は、最も重要なリセット・ソースによってリセットされる部分の簡略図です。

パワーオン・リセット (por_rst) は、対象範囲が最大のリセットです。これは、VDD ドメインにあるすべてのロジック、および VDDCORE のロジックのほとんどをリセットします。VBAT によって駆動されるバックアップ・ドメインにあるロジック、および VSW ドメインにあるロジックはリセットされません。パワーオン・リセットはシステム・リセットと app_rst もトリガし、パワーオン・リセット中に NRST PAD がアサートされることに注意してください。

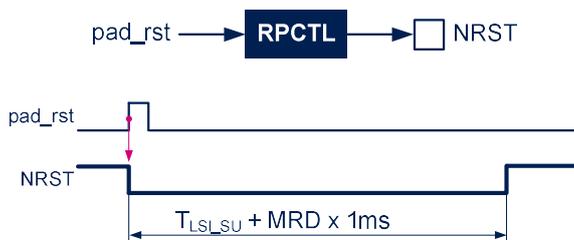
アプリケーション・リセット (app_rst) は、RCC、PWR にある一部のリソースを除いて、VDDCORE ドメインにあるほとんどのロジックをリセットします。バックアップおよび VSW ドメインは、このリセットの影響を受けません。

システム・リセット (nreset) は、RCC、PWR にある一部のリソースを除いて、VDDCORE ドメインにあるほとんどのロジックをリセットします。デバッグ・コンポーネント、IWDG[2:1]、バックアップおよび VSW ドメインはこのリセットの影響を受けません。

vcore リセット (vcore_rst) は、RCC、PWR にある一部のリソースを除いて、VDDCORE ドメインにあるほとんどのロジックをリセットします。IWDG [2:1] とバックアップおよび VSW ドメインは、このリセットの影響を受けません。

バックアップ・ドメイン・リセット (nreset_vsw) は、RTC と外部低速発振回路、バックアップ RAM、および保持 RAM を含むバックアップおよび VSW ドメインにあるすべてのコンポーネントをリセットします。

リセット時間幅は、外部コンポーネントの制約に適合可能



調整可能な最小パルス・リセット時間幅:

- 1~31ms
- RPCTL ブロックはバイパス可能
- LSI発振回路による精度



リセット・パルス制御ブロックにより、アプリケーションは NRST パットの最小アクティブ化時間を制御できます。

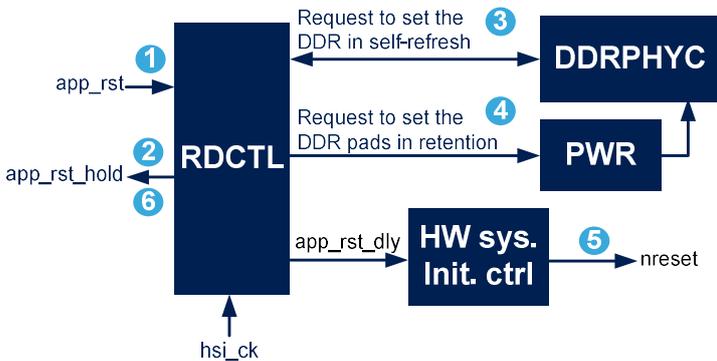
一部の外部デバイスでは NRST のアクティブ化のために最小リセット時間幅が必要になる場合があるため、特にこの機能は役立ちます。

リセット・パルス時間幅は、LSI 発振回路の精度で 1~31ms に調整できます。

リセットが発生したときに LSI のスイッチがオフになっていると、RPCTL 制御レジスタが必要とする限り、LSI のスイッチがオンになります。

リセットは、DDR がセルフ・リフレッシュになるまで遅延可能

- アプリケーション・リセットが発生したときに DDR をセルフ・リフレッシュ・モードに設定可能



- RDCTL はシステム・リセットの伝播を遅らせます。
- RDCTL は DDR をセルフ・リフレッシュ・モードに切り替えるようにリクエストします。
- これが完了すると、RDCTL はシステム・リセットを伝播させます。
- タイムアウトの場合、システム・リセットは伝播されません。



life.augmented

リセット遅延制御ブロックによって、アプリケーションは、回路へシステム・リセット (nreset) を生成する前に、外部 DDR デバイスをセルフ・リフレッシュ・モードに設定することができます。

アプリケーション・リセットが発生すると(①参照)、RDCTL はこのリセットをアクティブに維持し、システム・リセットの伝播を防止します(②参照)。

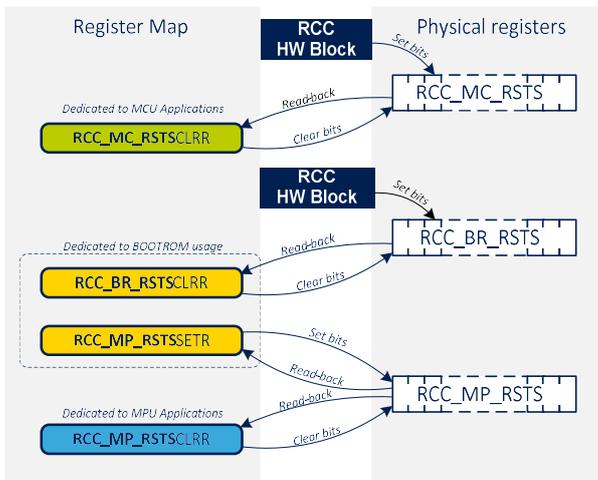
次に、RDCTL は DDRPHYC に DDR をセルフ・リフレッシュに切り替えるようにリクエストします(③参照)。

これが完了すると、RDCTL は PWR に DDR パッドを保持に切り替えるように要求します(④参照)。

最後に、システム・リセットが伝播され(⑤参照)、アプリケーション・リセットの維持はなくなります(⑥参照)。

DDRPHYC がセルフ・リフレッシュのリクエストに応答しない場合、システム・リセットは、ユーザがプログラム可能なタイムアウトの遅延後に回路に伝播されます。

必要に応じて、RDCTL ブロックをバイパスすることもできます。



- RCC は、リセットの根本原因に従って 2 つの物理レジスタを更新します。
 - MCU 用に 1 つ (RCC_MC_RSTS)
 - BOOTROM 用に 1 つ (RCC_BR_RSTS)
- MPU 用の物理レジスタ RCC_MP_RSTS は BOOTROM によって更新されます。
- MPU は、RCC_MP_RSTCLR レジスタを介してリセットの根本原因をチェックおよび認識できます。
- MCU は、RCC_MC_RSTCLR レジスタを介してリセットの根本原因をチェックおよび認識できます。



RCC はリセットの根本原因を `RCC_MC_RSTS` と `RCC_BR_RSTS` の 2 つのレジスタに格納します。BOOTROM は、`RCC_BR_RSTCLR` レジスタを介してリセットの根本原因をチェックおよび認識できます。さらに、BOOTROM は `RCC_MP_RSTSET` レジスタを介して物理レジスタ `RCC_MP_RSTS` を更新します。

MPU は、`RCC_MP_RSTCLR` レジスタを介してリセットの根本原因をチェックおよび認識できます。MCU は、`RCC_MC_RSTCLR` レジスタを介してリセットの根本原因をチェックおよび認識できます。

- RCC により、MCU がリセットされるたびに MPU が MCU を HOLD_BOOT に設定できるようになります。
- システム・リセットがアクティブになると (nreset)、MCU は自動的に HOLD_BOOT になります。
- MCU がリセットされるたびに、RCC は BOOT_MCU ビットをチェックします。
 - BOOT_MCU = "0" の場合、MCU は HOLD_BOOT を維持します。
 - BOOT_MCU = "1" の場合、MCU は再起動されます。
- HOLD_BOOT では、MCU が無効になり、システムからは CStop のように見えます。



HOLD_BOOT 機能により、MCU がリセットされるたびに MPU が MCU を 停止状態に維持できるようになります。

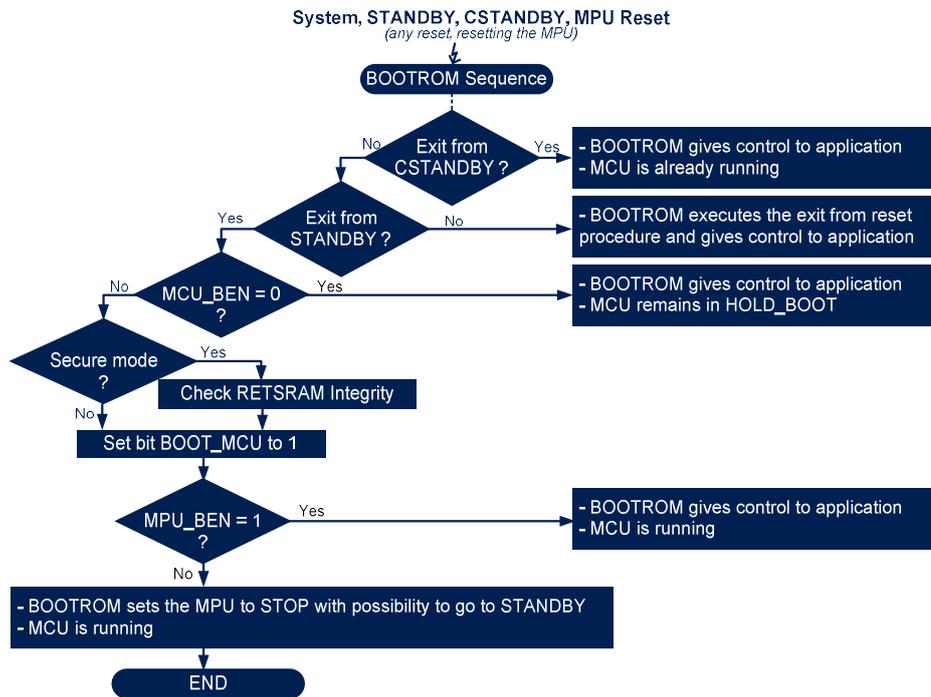
システムリセット後は、HOLD_BOOT 機能が有効になり MCU は停止します。

次に、MPU はアプリケーションコードを MCU の SRAM にロードし、BOOT_MCU ビットを 1 に設定して MCU が動作できるようにします。

回路が動作中の場合、MPU は MCU のリセット後の MCU の動作を次のように決定できます。

BOOT_MCU = 1 の場合、MCU は そのリセット後に単純に再起動できます。

BOOT_MCU = 0 の場合、MCU は そのリセット後に停止状態にとどまることができます。



MPU は、STANDBY モード後にアプリケーションを再起動できるプロセッサを決定できます。
この機能は、MPU_BEN と MCU_BEN の 2 つのビットで制御されます。

たとえば、システムが STANDBY モードの場合、イベントによってシステムが復帰されると、BOOTROM は MCU をアクティブにする必要があるかどうか (MCU_BEN ビット) をチェックします。MCU をアクティブにする必要がある場合、MCU に制御を渡す前に、BOOTROM コードによって RETSRAM の内容がチェックされるか。されないかのいずれかになります。

MPU アプリケーションは MPU_BEN ビットに従ってブートされるか否かが決まります。
MCU が MPU をウェイクアップする必要がある場合は、SEV イベントを生成することができます。

低電力、精度、および性能のためのクロック・ソースの選択

- 3つの内部クロック・ソース
 - 高速内部 64MHz RC発振回路(HSI)
 - 低電力内部 4MHz RC発振回路(CSI)
 - 低速内部 32kHz RC発振回路(LSI)
- 2つの外部発振回路
 - クロック・セキュリティ・システムを備えた高速外部 8~48MHz 発振回路(HSE)
 - クロック・セキュリティ・システムを備えた低速外部 32.768kHz 発振回路(LSE)
- それぞれ 3つの独立した出力を備えた 4つの PLL



クロック供給部分に注目すると、RCC では、低電力、精度、およびパフォーマンスの要件に応じて選択できる幅広いクロックソースを提供しています。

STM32MP1 マイクロ・プロセッサには、以下の 3つの内部RC発振回路が組み込まれています。

- 64、32、16、または 8MHz で動作できる高速内部RC発振回路(HSI)
- 4MHz で動作する低電力内部RC発振回路(CSI)
- 低速内部 32kHz RC発振回路(LSI)

また、外部の水晶または発振子で動作する以下の 2つの発振回路も組み込まれています。

- クロック・セキュリティ・システムを備えた高速外部 8~48MHz 発振回路(HSE)
- 同じくクロック・セキュリティ・システムを備えた低速外部 32.768kHz 発振回路(LSE)

4つの PLL も利用可能で、それぞれに 3つの独立した出力があり、さまざまな周波数でさまざまなペリフェラルにクロックを供給します。USB と DSI には専用の PLL があることに注意してください。

1%の精度、高速、速い復帰時間

パラメータ	値
精度(標準)	温度範囲で: +/- 1%
起動時間	最大 3 μ s
消費電力(標準)	標準 320 μ A

- HSI クロックは工場調整済みで、ユーザによる調整も可能
- HSI クロックは、STOP モードからの復帰用クロックとして選択可能
- システムが STOP モードを終了すると、MCU は HSI クロックで再開します
- 起動ペナルティを回避するために、STOP モード中に HSI クロックをアクティブにしておくことが可能
- 一部のペリフェラルは、STOP モード中にカーネル・クロックが必要な場合に HSI クロックを有効にすることが可能



life.augmented

高速内部発振回路は、1%の精度、速い復帰時間を備えた 64MHz の RC発振回路です。

HSI クロックは生産試験中に調整されていますが、ユーザによる調整もできます。

専用の分周器によって、64、32、16、または 8MHz のクロックを生成できます。

システムが STOP モードから復帰するとき、HSI クロックは MCU のシステム・クロックとして選択されます。

復帰時間を短縮するためにシステムが STOP モードに移行しても、HSI クロックは電力供給されたままです。

I2C や UART などの一部のペリフェラルは、独自の処理のためにシステム STOP モードで HSI クロックのアクティブ化を要求できます。

HSI が無効になっている場合、RCC は自動的に HSI を有効にし、それを要求したペリフェラルにこのクロックを提供します。

低電力内部(CSI)クロック

17

低電力、速い復帰時間

パラメータ	値
精度(標準)	温度および供給電圧範囲で: +/- 5%
起動時間(最大)	2.4μs
消費電力(標準)	25μA

- CSI クロックは工場では調整済みで、ユーザによる調整も可能
- 起動ペナルティを回避するために、STOP モード中に CSI クロックをアクティブにしておくことが可能
- 一部のペリフェラルは、STOP モード中にカーネルが必要な場合に CSI クロックを有効にすることが可能



life.augmented

低電力内部発振回路は、5% の精度、速い復帰時間を備えた 4MHz の RC 発振回路です。

CSI クロックは生産試験中に調整されていますが、ユーザによる調整もできます。

復帰時間を短縮するためにシステムが STOP モードに移行しても、CSI クロックは有効のままにしておくことができます。

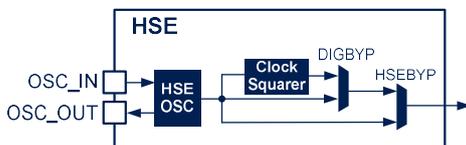
I2C や UART などの一部のペリフェラルは、独自の処理のためにシステム STOP モードで CSI のアクティブ化を要求できます。CSI クロックが無効になっている場合、RCC は自動的にそれを有効にし、このクロックを要求したペリフェラルに提供します。

安全な水晶システム・クロック

パラメータ	値
最大限界ゲイン gm	1.5mA/V
起動時間 (標準)	2ms
消費電力 (標準)	0.5mA ⁽¹⁾

⁽¹⁾ 24MHz 水晶の場合

- HSE 8~48MHz は、下記からクロックを生成できます。
 - 外部デジタル・ソース
 - 外部アナログ・ソース
 - 最小 200mVpp の振幅
 - 外部水晶 / セラミック発振子 (4~48MHz)
- クロック・セキュリティ・システム (CSS)
 - HSE クロックの障害自動検出
 - 障害イベントの検出時:
 - アプリケーション・リセットが生成されます
 - バックアップ・レジスタと BKPSRAM を保護するために、障害イベントが生成されます



8~48MHz の範囲で動作する水晶またはセラミックの発振子を HSE クロックに接続できます。

HSE はデジタル・バイパス・モードに設定することもでき、OSC_IN 入力でデジタルク・ロックを受信します。

アナログ・バイパス・モードでは、振幅が小さく、ピーク間で最小 200mV までのクロックを使用できます。

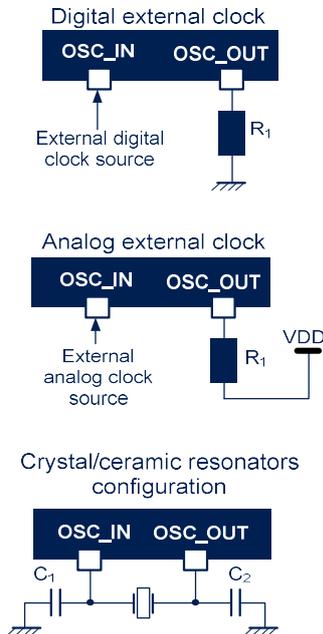
クロック・セキュリティ・システムによって、HSE の障害を自動的に検出できます。

この場合、アプリケーション・リセットが生成され、障害イベントが TAMP ブロックに提供され、バックアップ・レジスタとバックアップ RAM の保護が可能になります。

一部のペリフェラルは、独自の処理のためにシステム STOP モードで HSE クロックのアクティブ化を要求できます。

HSE クロックが無効になっている場合、RCC は自動的にそれを有効にし、このクロックを要求したペリフェラルに提供します。

HSE モードのスマート検出



- BOOTROM は以下の HSE 設定を検出します。
 - プルダウンが OSC_OUT に接続されている場合、BOOTROM は HSE クロックをデジタル・バイパス・モードに設定します。
 - プルアップが OSC_OUT に接続されている場合、BOOTROM は HSE クロックをアナログ・バイパス・モードに設定します。
 - それ以外の場合、BOOTROM は HSE をオシレータ・モードに設定します。

USB ポートを介してブートが要求された場合、BOOTROM は HSE 発振回路を使用する必要があります。そのため、BOOTROM は HSE クロックが製品でどのように使用されているかを知る必要があり、それを適切に設定します。

外部デジタル・クロックが HSE クロックとして提供されることを BOOTROM に通知するには、OSC_OUT を抵抗を介してグラウンドに接続する必要があります。外部アナログ・クロックが HSE クロックとして提供されることを BOOTROM に通知するには、OSC_OUT を抵抗を介して VDD に接続する必要があります。プルアップまたはプルダウンが検出されない場合、BOOTROM は HSE をオシレータ・モードで設定します。

低速内部 (LSI) クロック

20

超低電力内部 32kHz 発振回路

パラメータ	値
全温度および VDD 範囲での精度 (標準)	32kHz +/- 5%
起動時間 (最大)	170μs
消費電力 (標準)	130nA

- LSI クロックは一部の内部機能に使用され、RTC、LPUART、LPTIM、IWDG などのさまざまなペリフェラルに使用できます。
- STOP モードと STANDBY モードの間、LSI クロックはアクティブのままです。



life.augmented

STM32MP1 マイクロプロセッサには、超低電力 32kHz RC発振回路が組み込まれており、VBAT モードを除くすべてのモードで使用できます。

LSI クロックは、RTC、LPUART、一部の RCC ブロック、低電力タイマ、および独立型ウォッチドッグのクロックに使用できます。

低速外部 (LSE) クロック

21

低電力または高駆動に設定可能な 32.768kHz

パラメータ	値
最大限界ゲイン gm	0.5 μ A/V ⁽¹⁾ 、0.75 μ A/V ⁽²⁾ 、 1.7 μ A/V ⁽³⁾ 、2.7 μ A/V ⁽⁴⁾
起動時間(最大)	2 秒
消費電力(標準)	290nA ⁽¹⁾ 、390nA ⁽²⁾ 、 550nA ⁽³⁾ 、900nA ⁽⁴⁾

(1) 超-低駆動モード

(2) 中-低駆動モード

(3) 中-高駆動モード

(4) 高駆動モード

- LSE は下記からクロックを生成できます。
 - 外部デジタル・ソース
 - 外部アナログ・ソース
 - 外部水晶／セラミック発振子
- 外部水晶の特性に応じて選択可能な駆動能力
- すべての電力モードおよび VBAT モードで使用可能
- LSE のクロック・セキュリティ・システムは、VBAT を除くすべてのモードで使用できます。
- LSE クロックは、RTC、U(S)ART、LPUART、LPTIM に使用できます。



32.768kHz の低速外部発振回路は、外部水晶または発振子、またはバイパス・モードで外部クロック・ソースを使用することができます。

発振回路の駆動能力は、発振回路の特性を外部水晶デバイスに適合させるために選択することができます。

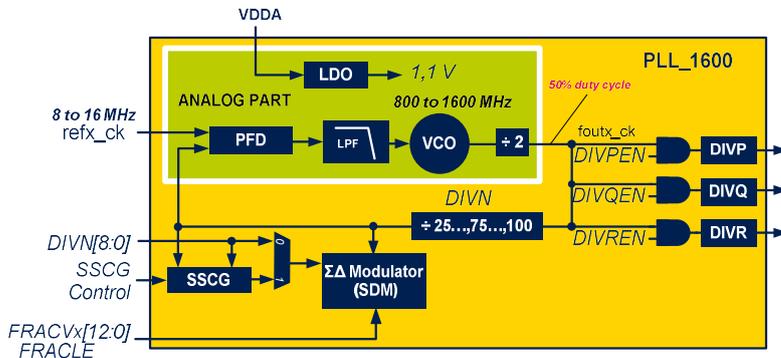
クロック・セキュリティ・システムによって、LSE 発振回路の障害を監視できます。

障害が発生した場合、アプリケーションは RTC クロックを LSI クロックに切り替えることができます。

クロック・セキュリティ・システムは、VBAT モードを除くすべてのモードで機能します。

LSE クロックは、RTC、USART または低電力 UART のペリフェラル、および低電力タイマのクロックに使用できます。

高周波、分数、スペクトル拡散可能



• 2つの PLL_1600:

- 入力周波数範囲: 8 から 16MHz
- VCO 周波数範囲: 800 から 1600MHz
- 13bit の分数逡倍数
 - 動作中にプログラム可能
- 内蔵 LDO
- EMI を低減するためのスペクトル拡散
- PLL あたり 3 つの出力
 - 1~128 の範囲の後段分周器付き



デバイスに組み込まれた PLL では、システムまたはペリフェラルのクロックに必要な周波数を生成する柔軟な方法を提供しています。

STM32MP1 マイクロプロセッサには、2 つの PLL1600 が内蔵されています。

入力周波数は、8~16MHz でなければなりません。

VCO 周波数は、800~1600MHz の周波数範囲でなければなりません。

また、PLL からは 3 つの異なる出力が提供されます。これらの出力はすべて、後段分周器 (DIVP、DIVQ、DIVR) を介して、VCO 周波数を 2 で割ったものから導出されます。

さらに、PLL を無効にすることなく、後段分周器の値を変更できます。アプリケーションは、対応する後段分周器を無効にし、分周比を変更して、後段分周器を再度有効にする必要があります。

50% に近いデューティ・サイクルを実現するために、アプリケーションは後段分周器を偶数の値にプログラムする必要があります。

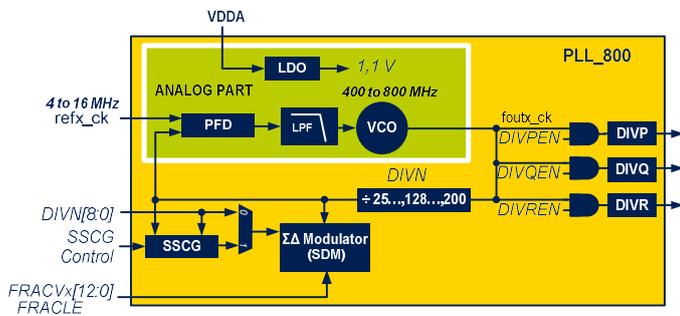
PLL は分数モードに切り替えることができ、VCO 周波数の高精度化が可能になります。13bit の分数分周器は、PLL を無効にすることなく変更できます。この機能は、正確なクロック・ドリフト補正を行うのに役立ちます。

最後に、PLL には、特に DDR インタフェース向けに、EMI の量を減らすためのスペクトル拡散メカニズムが組み込まれています。

中間周波数、分数、スペクトル拡散可能

• 2つの PLL_800:

- 入力周波数範囲: 4 から 16MHz
- VCO 周波数範囲: 400 から 800MHz
- 13bit の分数逡倍数
 - 動作中にプログラム可能
- 内蔵 LDO
- EMI を低減するためのスペクトル拡散
- PLL あたり 3 つの出力
 - 1~128 の範囲の後段分周器付き



STM32MP1 マイクロプロセッサには、2 つの PLL800 も用意されています。

アーキテクチャは前のものと非常に似ており、これらの PLL は中速クロックの生成により特化されています。

入力周波数は、4~16MHz でなければなりません。

VCO 周波数は、400~800MHz の周波数範囲でなければなりません。

その他の機能は PLL1600 に類似しています。

- 整数モードでは、3つの出力の1つの周波数は下記のとおり:

$$F_{pll_y_ck} = \frac{F_{ref_ck} \cdot (DIVN + 1)}{DIVy + 1}$$

$$F_{VCO_PLL_800} = F_{ref_ck} \cdot (DIVN + 1)$$

$$F_{VCO_PLL_1600} = 2 \cdot F_{ref_ck} \cdot (DIVN + 1)$$

- 分数モードでは、3つの出力の1つの周波数は下記のとおり:

$$F_{pll_y_ck} = \frac{F_{ref_ck} \cdot \left((DIVN + 1) + \frac{FRACV}{2^{13}} \right)}{DIVy + 1}$$

y は、P、Q、または R です

F_{ref_ck} は、PLL に提供される周波数です

$F_{pll_y_ck}$ は、PLL で生成される3つの出力クロックの1つです



このスライドは、整数モードと分数モードの場合の、PLL の入力周波数と出力周波数の関係を示しています。
アプリケーションでは、VCO の周波数範囲と、PLL に提供される信号のクロック周波数を守るように注意する必要があります。

スペクトル拡散ブロック(SSCG)

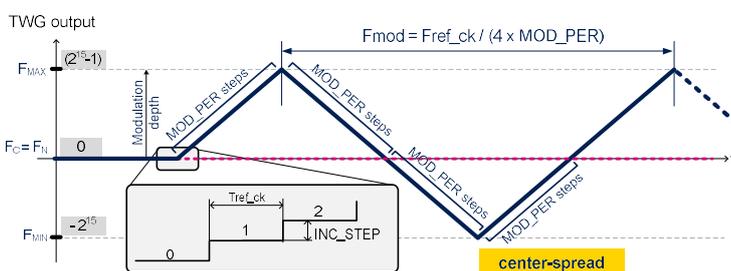
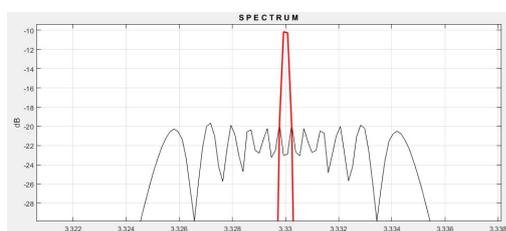
- スペクトル拡散ブロックでは、VCO 周波数の微小三角変調を実施します。
- 以下のパラメータが調整可能です。

- 変調周期(MOD_PER)
- 変調度(MOD_INC)
- 変調モード(SSCG_MODE)

$$M_D (\%) = \frac{\text{MOD_PER} \times \text{INC_STEP} \times 100 \times 5}{(2^{15} - 1) \times (\text{DIVN} + 1)}$$

$$F_{\text{mod}} = \frac{F_{\text{ck_ref}}}{4 \times \text{MOD_PER}}$$

- ここでは、変調度(M_D)と変調周波数(F_{mod})の式を示します。
- 例として EMI 低減の例(黒い曲線)を示します。



スペクトル拡散クロック・ジェネレータ(SSCG)では、16bit の分数分周器を使用して、VCO 周波数の三角周波数変調を実施します。

アプリケーションでは以下を調整できます。

- 変調周期(MOD_PER フィールドを介して)
- 変調度(MOD_INC フィールドを介して)
- 変調モード(SSCG_MODE フィールドを介して)このオプションによって、SSCG で VCO 周波数を中心とした変調にするか、VCO 周波数よりも低い側の変調にするかのどちらかにできます。

SSCG は 主にEMI 干渉に影響を与える DDR インタフェースのために使用されます。

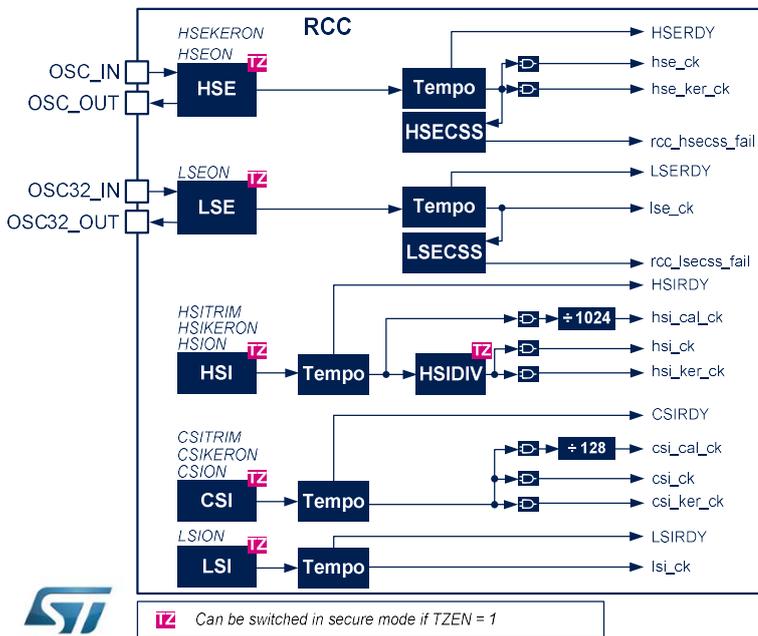
これは EMI 干渉は、急峻なエッジを持つ高周波で反転する I/O の数によるものだからです。

変調周波数は DDR インタフェースの速度に比べて非常に低いため、SSCG はジッタに大きな影響を与えません。

変調周波数は通常 20~50kHz の範囲で、変調度は約 1% です。

小さなグラフは、クロック拡散の効果例を示しています。周波数を制限された周波数帯域に拡散すると、信号の出力がこの帯域に拡散し、出力のピークが減少します。赤色の信号は SSCG なしの VCO 出力で、黒色の信号は SSCG ありの VCO 出力です。

発振回路のクロック分配



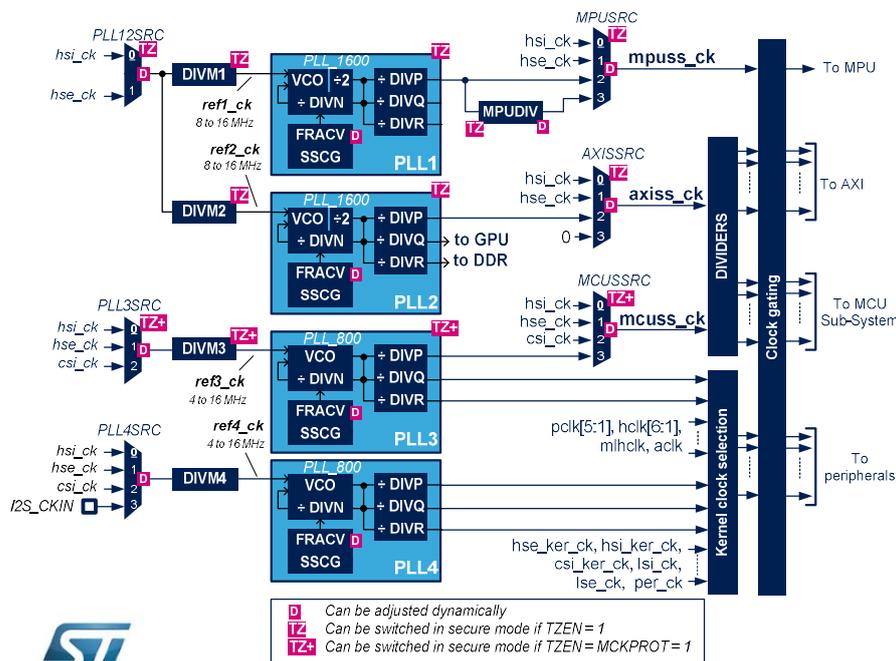
- 各発振回路には準備ロジックがあります。
- 発振回路の制御はセキュア・モードに切り替えることができます。
- HSE クロック、HSI クロック、および CSI クロックは、カーネル・クロックとシステム・クロックを提供します。

発振回路は、準備ができていなければクロックを提供しません。レディ・フラグは過渡期間に基づいています。LSE がバイパス・モードの場合、この過渡期間は LSE クロックではバイパスされます。

TZEN ビットが“1”にセットされているとき、発振回路の制御はセキュア・モードに切り替えることができます。

HSE 発振回路、HSI 発振回路、および CSI 発振回路は、ペリフェラル専用のカーネル・クロックと、システム・クロックを提供します。これらのクロックは、ゲーティング・ロジックを除いて同一です。

簡略化した最上位のクロック分配



- PLL1 は MPU 専用
- PLL2 は AXI サブシステム専用
 - PLL1 と PLL2 は同じクロック・ソースを共有
- PLL3 は MCU および各種ペリフェラル専用
- PLL4 はペリフェラル 専用
 - PLL4 はそのリファレンス・クロックをパッド (I2S_CKIN) から受信可能
- すべてのマルチプレクサは動的です



PLL1 および PLL2 は HSE または HSI のいずれかをリファレンス・クロックとして使用できます。

PLL3 は HSE、HSI、または CSI のいずれかをリファレンス・クロックとして使用でき、PLL4 は HSE、HSI、CSI、または I2S_CKIN のいずれかをリファレンス・クロックとして使用できます。

PLL1 は MPU 専用です。DIVP は、50% のデューティ・サイクルを守るために、バイパスされるか、偶数の分周比を使用する必要があります。

PLL2 は、AXI サブシステム、GPU、および DDR 専用です。

PLL3 は MCU サブシステムとペリフェラル・カーネル・クロック専用で、最後に PLL4 はペリフェラル・カーネル・クロック専用です。

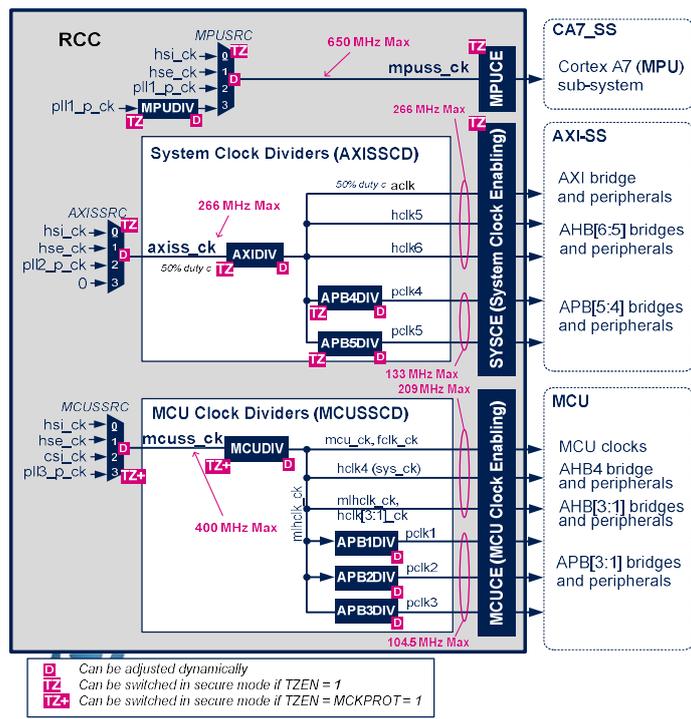
各 PLL には、各 PLL のリファレンス・クロックを調整するための専用の前段分周器があります。

PLL の 1 つが有効になっている場合は、PLL のクロック・ソースを変更しないでください。

さらに、MPU、AXI、GPU、DDR のクロックの生成をセキュア・モードに切り替えることができます。

必要に応じて、MCU クロックの生成もセキュア・モードに切り替えることができます。

簡略化したサブシステムのクロック・ツリー



- 動的周波数分周器による容易な周波数調整
- AXI サブシステムは動的に変更可能
- APB クロックは動的に変更可能

STM32MP1 マイクロプロセッサには主に以下の 3 つのサブシステムがあります。

- MPU サブシステム
- AXI サブシステム
- MCU サブシステム

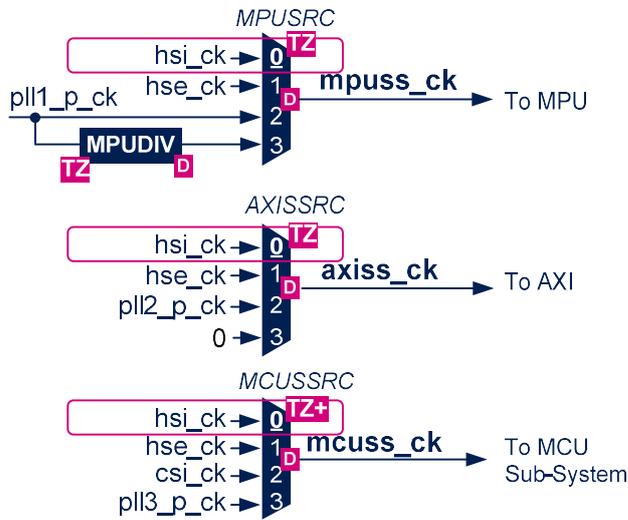
MPU に提供されるクロックは、MPUSRCの切り替えによって、または MPUDIV を変更することによって、動的に変更できます。

AXI サブシステムに提供されるクロックも、AXISSRC の切り替えによって、または AXIDIV を変更することによって、動的に変更できます。さらに、APB4 および APB5 の周波数は動的に変更できます。

MCU サブシステムに提供されるクロックも、MCUSSRCの切り替えによって、または MCUDIV を変更することによって、動的に変更できます。さらに、APB1、2、および 3 の周波数は動的に変更できます。

MPU には最大 650MHz までのクロックを供給できます。MCU およびマルチ・レイヤ AHB には、最大 209MHz までのクロックを供給できます。AXI クロックは 266MHz を超えてはなりません。その他の制限は図に示されています。

リセット後に選択されるクロック



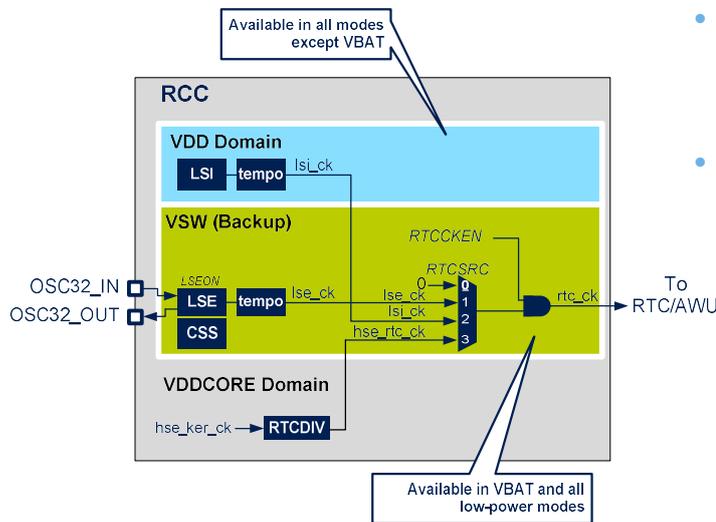
下記の後:

- パワー・オン・リセット
 - システム・リセット
 - アプリケーション・リセット
 - STANDBY からの終了
-
- MPU は HSI クロックで始動
 - AXI は HSI クロックで始動
 - MCU は HSI クロックで始動



D Can be adjusted dynamically
TZ Can be switched in secure mode if TZEN = 1
TZ+ Can be switched in secure mode if TZEN = MCKPROT = 1

パワー・オン・リセット、システム・リセット、アプリケーション・リセット、または STANDBY モードからの終了後、MPU、AXI サブシステム、および MCU サブシステムは HSI クロックでクロック供給されます。



- RTC/AWU のソースは以下のいずれかです。
 - LSE、LSI、または分周された HSE クロック
- RTCSRC は、バックアップ・ドメインをリセットした後、または LSE に障害が発生した後に一度だけ変更できます。
 - LSE クロックが選択されている場合、RTC は VBAT を含むすべてのモードで動作します。
 - LSI クロックが選択されている場合、RTC は VBAT を除くすべてのモードで動作します。
 - HSE クロックが選択されている場合、RTC は CRun、CSleep、STOP で動作します (HSEKERON = "1" の場合)。



RTC/AWU のクロックは、LSE、LSI、または HSE クロックの分周から選択できます。

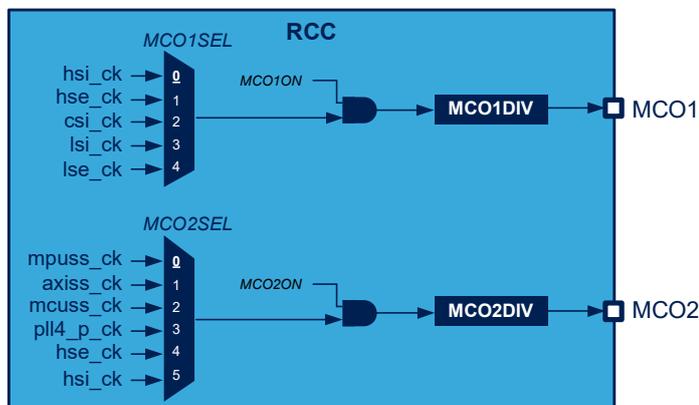
分周された HSE クロックが RTC/AWU のカーネル・クロックとして使用される場合、周波数は 4MHz を超えてはなりません。

RTCSRC スイッチは、バックアップ・ドメインのリセット後、または LSE の障害が発生した後に一度だけ変更できます。

また、LSI クロックが RTC/AWU のカーネル・クロックとして使用されている場合、RTC/AWU は VBAT モードではクロック供給されなくなります。

分周された HSE クロックが RTC/AWU のカーネル・クロックとして使用されている場合、RTC/AWU は STANDBY または VBAT モードではクロック供給されなくなり、HSEKERON が "1" に設定されている場合は STOP モードでクロック供給できます。

LSE クロックは、VBAT モードを含むすべての低電力モードで有効のままです。



- 2つのクロック出力:クロック・ソースの多くの選択を提供する MCO1 と MCO2

アプリケーション側の利点

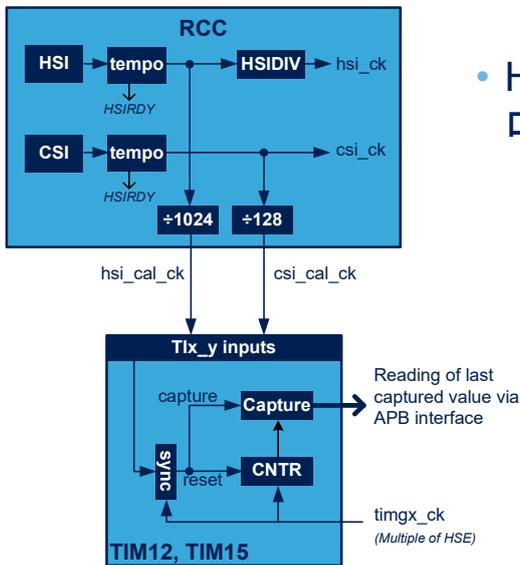
- MCO1 または MCO2 は外部デバイスに分配できます。



RCC は次の 2 つのクロック出力信号も提供します。MCO1 および MCO2。

プリスケラによって、アプリケーションは周波数を PAD の能力に適合させることができます。

この機能によって、回路の内部クロックを外部デバイスに分配できます。

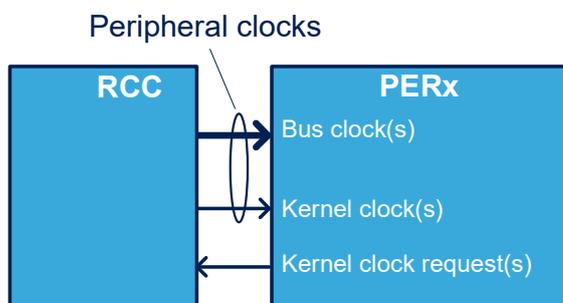


- HSE クロックに対して CSI クロックおよび HSI クロックの周波数を評価することが可能
 - 精度は 0.3% 以上
 - タイマは HSE クロックから導出したクロックでカウント
 - タイマは、較正クロックのうちの 1 つの 1 周期の間カウント
 - タイマのキャプチャレジスタは、最後に完了した較正值を保持
 - TIM 割込みサービスを使用可能



RCC には、HSE クロックに対して CSI および HSI クロックの周波数を 0.3% より高い精度で評価できる機能があります。タイマ12 と 15 は、そのための較正クロック信号を受け取ります。タイマは、較正クロック信号の1 周期内でタイマクロック周期の数をカウントします。タイマのキャプチャレジスタは、最後に完了した較正值を保持します。割込みサービスも使用可能です。

ペリフェラルに影響を与えずに変更可能なバス速度



- ペリフェラルは通常、次のクロックを受け取ります：
 - 1つまたは複数のバス・クロック
 - 1つまたは複数のカーネル・クロック
- バス・クロックにより、レジスタへのアクセスとメモリへのデータ転送ができます。
- カーネル・クロックは、通常、ペリフェラルが次のような外部インターフェースを処理するために使用されます：UART、I2S、SPI、USB、イーサネット ...



STM32MP1 マイクロプロセッサの多くのペリフェラルは、データと制御フロー用（プロセッサ・バス・インターフェース経由）とペリフェラル固有のインターフェース用に異なるクロックを持っています。

通常、プロセッサ・バス・インターフェースを介したデータおよび制御フロー用のクロックは「バス・クロック」と呼ばれ、ペリフェラル固有のインターフェース用のクロックは「カーネル・クロック」と呼ばれます。

ペリフェラルクロックは、ペリフェラルが受け取るクロック、すなわち、「バス・クロック」と「カーネル・クロック」を表します。

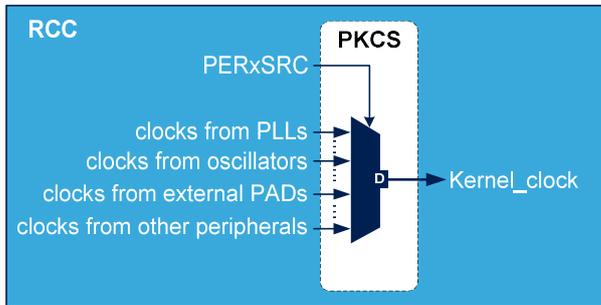
バス・クロックとカーネル・クロックを分離することによって、アプリケーションは、ペリフェラルに影響を与えることなく、相互接続とプロセッサの動作周波数を変更できます。ペリフェラルによっては、ペリフェラルがシステムにデータを転送する必要がない限り、バス・クロックを無効にすることもできます。

したがって、バス・プロセッサとメモリ用の周波数選択、およびペリフェラル・インターフェースの実際のニーズに応じた周波数選択との間で、優れた柔軟性を提供します。

例：UART には、とりわけ、シリアル・インターフェース通信用のボーレート・ジェネレータによって使用されるカーネル・クロックと、レジスタ・インターフェース用の APB クロックが備わっています。

さらに、一部のペリフェラルは、特定のイベントを検出したときにカーネルクロックを要求できます。

動的でグリッチのないカーネル・クロック・スイッチ



- カーネル・クロックを使用するほとんどのペリフェラルには、最適なカーネル・クロックを選択するためのクロック・スイッチがあります。
 - クロック・スイッチは動的でグリッチがありません。
 - ペリフェラルのニーズに応じて、いくつかのクロック・ソースが提案されています。
- カーネル・クロック・スイッチは、RCC レジスタを介して設定できます。

カーネル・クロックの分配は、このプレゼンテーションでは詳しく説明しません。具体的な詳細については、リファレンスマニュアルを参照してください。

カーネル・クロックを使用するほとんどのペリフェラルには、最適なクロックソースを選択するための動的なクロックスイッチがあります。

提案されているクロックソースは、一般的には以下があげられます。

PLL 出力のうちの 1 つ

内部または外部の発振回路:これは、システムが STOP モードのときにカーネル・クロックを必要とするペリフェラルに不可欠です。

パッド:たとえば、外部 PHY を使用するペリフェラルや、外部デバイスからのクロック基準を使用するオーディオで使用されます。

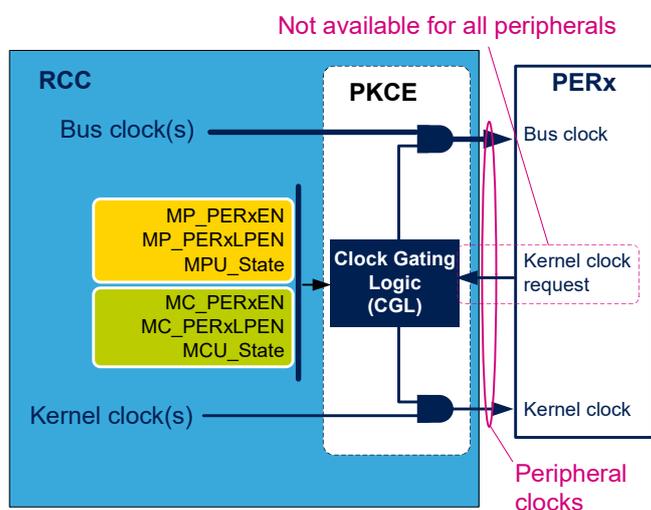
その他の内部ペリフェラル:ブロック間の同期のために使用されます。

動的切り替えにより、あるソースから別のソースへの移行が容易になります。

RCC レジスタによって、すべてのペリフェラルのカーネル・クロックを設定できます。

ペリフェラル・クロックのゲーティング

35



- ペリフェラル・クロックのゲーティングは、一般的にいくつかの下記パラメータによって制御されます。
 - イネーブル・ビット(MP_PERxEN、MC_PERxEN)
 - 低電力イネーブル・ビット(MP_PERxLPEN、MC_PERxLPEN)
 - プロセッサの状態
- MP_PERxEN ビットを“1”に設定すると、MPU に対してペリフェラル PERx が有効になります。
- MC_PERxEN ビットを“1”に設定すると、MCU に対してペリフェラル PERx が有効になります。
- この操作は、ペリフェラルの「割当て」と呼ばれます。



life.augmented

ペリフェラルは通常、次のクロックを受け取ります：

- 1 つまたは複数のバス・クロック
- 1 つまたは複数のカーネル・クロック

各プロセッサは、RCC に配置された専用レジスタを介して、ペリフェラル・クロックのクロック・ゲーティングを制御できます。

ペリフェラル・クロックのゲーティングは、いくつかの下記パラメータに依存します。

クロック・イネーブル・ビット：各プロセッサには、MP_PERxEN および MC_PERxEN という名前の専用のイネーブル・ビットがあります。

低電力クロック・イネーブル・ビット

プロセッサの状態 (CRun、CSleep、または CStop)

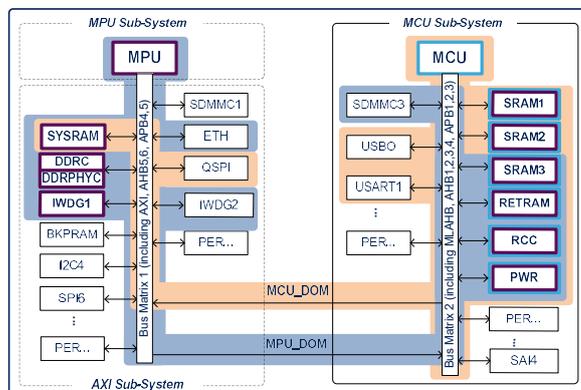
MP_PERxEN ビットを“1”に設定すると、MPU に対してペリフェラル PERx が有効になります。

MC_PERxEN ビットを“1”に設定すると、MCU に対してペリフェラル PERx が有効になります。

この操作は、ペリフェラルの「割当て」と呼ばれます。

次のスライドは、クロック・ゲーティング条件に関する追加情報です。

- 割当てにより、MPU または MCU がペリフェラルを有効にしたことが RCC に通知されます。この情報は低電力モードで使用されます。
- ペリフェラルを使用する前に、プロセッサはそれを独自の用途に割り当てる(つまり、有効にする)必要があります。
- SYSRAM、DDRC、DDRPHYC、および IWDG1 は、MPU に暗黙的に割り当てられています(つまり、有効にされています)。
- RCC、PWR、SRAM1、2、3、RETRAM は、両方のプロセッサに暗黙的に割り当てられています(つまり、有効にされています)。
- 各プロセッサ (MPU および MCU) には、ペリフェラルを割り当てるための専用のレジスタ・セットがあります。これらのレジスタは、多くの場合 RCC_MP_xxxxx または RCC_MC_xxxxx などの名前が付けられています。



RCC には 2 つのレジスタ・セットがあり、これによって各プロセッサがペリフェラルを独自に使用できるように有効化(割当て)できることに注意が必要です。

ペリフェラルの割当てにより、MPU または MCU がペリフェラルを有効にしたことが RCC に通知されます。この情報は RCC によって低電力モードでのクロック制御のために使用されます。したがって、ペリフェラルを使用する前に、プロセッサはそれを独自の用途に割り当てる必要があります。

同じペリフェラルを両方のプロセッサで割り当てることができます。リソースの競合を回避することはアプリケーションに委ねられます。

一部のペリフェラルは、プロセッサに暗黙的に割り当てられています。

SYSRAM、DDRC、DDRPHYC、および IWDG1 は暗黙的に MPU に割り当てられ、MCU も SYSRAM を割り当てることができますが、デフォルトでは MCU に割り当てられていません。SYSRAM が MCU に割り当てられている場合は、MPU が CStop モードに移行しても SYSRAM は引き続き使用できます。そうでない場合、SYSRAM はクロックゲーティングされています。

他のいくつかのペリフェラルは両方のプロセッサに割り当てられています。これは、RCC、PWR、SRAM1、2、3、RETRAM の場合です。

プロセッサにペリフェラルが割り当てられている場合、このペリフェラルのステータスは、低電力モードのプロセッサの状態によって異なります。

プロセッサと、このプロセッサによって割り当てられたペリフェラル、および関連する相互接続は、RCC によってプロセッサドメインと見なされます。

RCC がペリフェラル割当てを利用する簡単な例を示すと、あるドメインのペリフェラルの 1 つが CRun または CSleep モードにある他のドメインのプロセッサによって使用されている場合、RCC はこのドメインのクロック・ゲーティングを許可しないという規則を述べるすることができます。

システムの状態	MPU の状態	MCU の状態
RUN	CRun/CSleep/CStop/CStandby	CRun/CSleep
STOP	CStop/CStandby(PWR_MPUCR.PDDS = 1 の場合)	CStop(PWR_MCUCR.PDDS = 0 の場合)
	CStop/CStandby(PWR_MPUCR.PDDS = 0 の場合)	CStop(PWR_MCUCR.PDDS = 1 の場合)
STANDBY	CStop/CStandby(PWR_MPUCR.PDDS = 1 の場合)	CStandby(PWR_MCUCR.PDDS = 1 の場合)

- 2つのプロセッサのいずれかが少なくとも CRun または CSleep モードの場合、システムは RUN モードです。
 - システムが RUN モードのとき、発振回路と PLL はアクティブのままです。一部のブリッジは、このブリッジに接続されているペリフェラルの割当てに従ってクロック・ゲーティングできます
- 両方のプロセッサが CStop モードであり、そのうちの 1 つが STANDBY モードへの移行を許可していない場合は、システムは STOP モードです。
- 両方のプロセッサが CStop モードであり、両方が STANDBY モードへの移行を許可している場合、システムは STANDBY モードです。
- STANDBY モードでは、VDDCORE はスイッチオフされます。



次の表は、システムの状態とプロセッサの状態を簡単に示したものです。

2つのプロセッサのいずれかが少なくとも CRun または CSleep モードの場合、システムは RUN モードです。

両方のプロセッサが CStop モードであり、そのうちの 1 つが STANDBY モードへの移行を許可していない場合は、システムは STOP モードです。

両方のプロセッサが CStop モードであり、両方が STANDBY モードへの移行を許可している場合、システムは STANDBY モードです。

STANDBY モードでは、VDDCORE はスイッチオフされます。

システム状態の詳細については、PWR のトレーニングスライドを参照してください。

STOP モードへの移行

MPU が CStop モードまたは CStandby モードで、MCU が CStop モードの場合、システムは STOP モードまたは LP(LV)-Stop モードになります。

1. RCC は、DDRC が DDR をセルフ・リフレッシュに設定するのを待ちます。
2. RCC はすべてのクロックを停止します。
3. RCC はすべてのクロックがゲートされていることを PWR に通知します。

STOP モード時

以下を除くすべてのクロックが無効になります。

1. 有効にされている場合、LSE と LSI は有効のままになります。
2. HSE 発振回路、HSI 発振回路、および CSI 発振回路は、それらのxxxKERON ビットが“1”に設定されている場合、有効のままです。
3. HSE 発振回路、HSI 発振回路、および CSI 発振回路は、ペリフェラルの要求に応じて一時的にアクティブにできます。



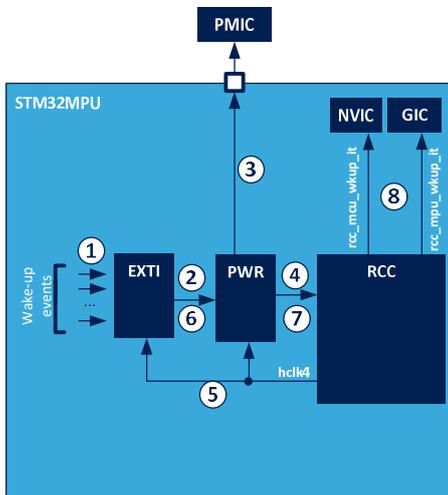
STOP モードへの移行が要求されると、RCC は次の操作を実行します。RCC は、DDRC がセルフ・リフレッシュ・モードに入るのを待ちます。次に、RCC はすべてのクロックを停止します。より正確には、PLL は無効にされ、発振回路は、アクティブのままであることが要求されている場合を除いて無効になります。最後に、RCC はすべてのクロックがゲートされていることを PWR ブロックに通知します。その後、PWR ブロックは、設定された方法に従って、VDDCORE 電圧を下げるか、または現在の VDDCORE 電圧を維持することができます。

システムが STOP モードの場合、以下を除くほとんどのクロックが無効になります。

LSE と LSI は、有効にされている場合、動作し続けることができます。HSE 発振回路、HSI 発振回路、または CSI 発振回路は、KERON ビットが“1”に設定されている場合、STOP モードでアクティブのままにできます。この機能により、STOP モードでのペリフェラルの非常に高速な反応と、STOP モードからのより迅速な復帰が可能になります。HSE 発振回路、HSI 発振回路、および CSI 発振回路も、ペリフェラルの要求に応じてアクティブにできます。

STOP モードの終了 (1/2)

39



- システムは、STOP モードで動作できるペリフェラルによって生成されたイベントによって、STOP モードを終了します。
- 終了シーケンスの主なステップ:
 1. ペリフェラルがウェイクアップ・イベントを EXTI に生成(①)
 2. EXTI は PWR に通知(②)
 3. PWR は、必要に応じて PMIC の状態を復元し、VDDCORE が有効になるまで待ちます(③)
 4. PWR は RCC に、STOP モードの終了を実行できることを通知(④)
 5. RCC は EXTI および PWR にクロック hclk4 を供給(⑤)
 6. その後、PWR と EXTI は、復帰が MCU または MPU のどちらに対して要求されているかを明確にできます(⑥、⑦)
 7. RCC は対応するプロセッサにクロックを復元し、ウェイクアップ割込みを生成(⑧)



システムは、STOP モードで動作できるペリフェラルによって生成されたイベントによって、STOP モードを終了します。

STOP モードに入る前に、アプリケーションは EXTI をプログラムして、どのイベントがシステムの STOP モードを終了させるのかを定義する必要があります。

ペリフェラルが予期されたウェイクアップ・イベントを生成すると、次の手順が発生します。

EXTI は、ウェイクアップ・イベントが STOP モードの終了を要求していることを PWR ブロックに通知します(図のステップ 2)。

PWR ブロックは、必要に応じて PMIC の状態を復元し、VDDCORE が有効になるまで待ちます(ステップ 3 参照)。

VDDCORE が公称電圧に達した後、PWR は RCC に、STOP モードの終了を実行できることを通知します(ステップ 4 参照)。

RCC は、HSI クロックがまだ利用可能でない場合は有効にし、EXTI および PWR にクロック hclk4 を供給します(ステップ 5 参照)。

その後、PWR と EXTI は、ウェイクアップ・イベントが MCU または MPU のどちらを復帰する必要があるかを定めることができます(ステップ 6 および 7 参照)。

最後に、RCC は対応するプロセッサにクロックを復元し、対応するウェイクアップ割込みを生成します(ステップ 8 参照)。

- MPU に対するシステム STOP モードの終了：
 - 選択された発振回路と PLL1 および PLL2 の自動アクティブ化を含め、mpuss_ck と axiss_ck のロック設定が復元されます。
 - PLL1 および PLL2 のアクティブ化は、プログラム可能な遅延 (PWRLP_TEMPO) によって遅延させることができます。
- MCU に対するシステム STOP モードの終了：
 - HSI クロックがシステム・クロックとして選択されます。
 - HSI の準備ができるとすぐに MCU が起動するか、または PWRLP_TEMPO 遅延が経過するのを待ちます。
 - PLL3 および PLL4 が無効にされます。



システムが STOP モードを終了するとき：

RCC は、STOP モードに移行する前と同じクロック設定を MPU および AXI サブシステムに提供します。つまり、PLL1 と PLL2 が STOP モードに移行する前に使用されていた場合は、それらは自動的に再度有効にされます。

RCC は、高速再復帰のために、HSI クロックを MCU に提供します。

ウェイクアップ・イベントが MPU 向けであった場合、RCC は MPU に対するウェイクアップ割込みを生成します。

ウェイクアップ・イベントが MCU 向けであった場合、RCC は MCU に対するウェイクアップ割込みを生成します。

PLL1 および PLL2 のアクティブ化を遅延させるために、PWRLP_TEMPO という名前のプログラム可能な遅延を設定できるように注意してください。

MCU の再復帰も、MCTMPSKP ビットに応じて PWRLP_TEMPO だけ遅らせることができます。

この機能は、外部電源デバイスの能力に応じて回路を調整するのに役立ちます。

カーネル・クロックのリクエスト 41

- UART や I2C などの一部のペリフェラルは、処理のためにカーネル・クロックをリクエストするイベントを非同期的に検出できます。
- RCC は、ペリフェラルを割り当てているプロセッサが CStop 状態である場合、またはシステムが STOP 状態である場合、オンデマンドでこれらのペリフェラルにカーネルクロックを供給できます。
- カーネル・クロック・リクエストは、選択したカーネル・クロックが次の発振回路である場合にのみ機能します: HSE、HSI、または CSI。
- 起動時間を短縮するために、STOP モード中に HSE、HSI、または CSI をアクティブのままにすることができます。
- STOP モードでは、LSI および LSE クロックがペリフェラル用に使用可能です。



UART や I2C などの一部のペリフェラルは、処理のためにカーネル・クロックをリクエストするイベントを非同期的に検出できます。

RCC は、ペリフェラルを割り当てている CPU が CStop モードである場合、またはシステムが STOP モードである場合、オンデマンドでこれらのペリフェラルにカーネル・クロックを供給できます。カーネル・クロック・リクエストは、選択したカーネル・クロック・ソースが次の発振回路である場合にのみ機能します: HSE、HSI、または CSI。

ペリフェラルがカーネル・クロックをリクエストするときの起動時間を短縮するために、HSE 発振回路、HSI 発振回路、または CSI 発振回路を STOP モード中にアクティブのままにすることができます。

STOP モードでは、有効になっていれば、LSI および LSE クロックはペリフェラル用に常に使用可能です。

ペリフェラル・クロックのゲーティング

42

- カーネル・クロックは、次の条件のいずれかが満たされる場合、ペリフェラル PERx に供給されます。
 1. ペリフェラルが割り当てられているプロセッサが CRun モードの場合
 2. ペリフェラルが割り当てられているプロセッサが CSleep モードで、かつ PERxLPEN =“1” の場合
 3. ペリフェラルが割り当てられているプロセッサが CStop モードで、かつ PERxLPEN =“1” であり、かつ ペリフェラルがカーネル・クロック・リクエストを生成しており、かつ カーネル・クロック・ソースが HSE、HSI、CSI、LSE、または LSI である場合
 4. ペリフェラルが割り当てられているプロセッサが CStop モードで、かつ PERxLPEN =“1” で、かつ、ペリフェラルのカーネル・クロック・ソースが LSE または LSI の場合
- バス・インタフェース・クロックは、条件 1 または 2 が満たされる場合にのみペリフェラルに供給されます。



life.augmented

カーネルクロックは、次の条件のいずれかが満たされる場合、ペリフェラルに供給されます。

- 1) ペリフェラルが割り当てられている CPU が CRun モードの場合
- 2) ペリフェラルが割り当てられている CPU が CSleep モードで、その LP-Enable ビットが“1”にセットされている場合
- 3) ペリフェラルが割り当てられている CPU が CStop モードで、かつ、その LP-Enable ビットが“1”にセットされており、かつ、ペリフェラルがカーネル・クロック・リクエストを生成しており、かつ、カーネル・クロック・ソースが HSE、HSI、CSI、LSE、または LSI である場合
- 4) ペリフェラルが割り当てられている CPU が CStop モードで、かつ、その LP-Enable ビットが“1”にセットされており、かつ、ペリフェラルのカーネル・クロック・ソースが LSE または LSI の場合

バス・インタフェース・クロックは、条件 1 または 2 が満たされる場合にのみペリフェラルに供給されます。

DDRC のセルフ・リフレッシュ・モード制御 (1/2)

- ソフトウェア・セルフ・リフレッシュ (SSR) :
 - 外部 DDR デバイスをすぐにセルフ・リフレッシュに切り替えるには、アプリケーションで DDRC を直接プログラムする必要があります。
- 自動セルフ・リフレッシュ (ASRx) :
 - DDRC は、**DDRC に定義された時間の間**、DDRC が非アクティブであると、自動的にセルフ・リフレッシュ・モードになります。
 - DDRC AXI ポートの 1 つでトランザクションが発生すると、DDR はセルフ・リフレッシュを終了します。
 - ASR1 では、RCC はプロセッサ・モード (CRun や CSleep など) とは無関係に DDRC のゲーティングを動的に制御します。
 - ASR2 では、RCC はプロセッサ・モード (CRun や CSleep など) とは無関係に DDRC と DDRPHYC ブロックのゲーティングを動的に制御します。



RCC には、DDR コントローラ (DDRC) および DDR PHY の低電力遷移を制御するために、特定のモードが組み込まれています。

ソフトウェア・セルフ・リフレッシュ・モード (SSR)

自動セルフ・リフレッシュ・モード (ASR)

ハードウェア・セルフ・リフレッシュ・モード (HSR)

ソフトウェア・セルフ・リフレッシュモードでは、外部 DDR デバイスをすぐにセルフ・リフレッシュ・モードに切り替えるには、アプリケーションで DDRC を直接プログラムする必要があります。

RCC では、他のペリフェラルに対して実行するのと同じように、DDRC および DDRPHYC クロックのゲーティングを実行します。

アプリケーションによって設定された時間内に DDRC が何のトランザクションも受信しなかった場合に DDR をセルフ・リフレッシュ・モードに切り替えるために、アプリケーションが DDRC をプログラムした場合は、自動セルフ・リフレッシュ・モードを使用する必要があります。このモードでは、定義された時間の間 DDR が使用されない場合に、DDR はセルフ・リフレッシュ・モードになります。DDRC が新しいトランザクションを受信すると、DDRC は DDR にセルフ・リフレッシュ・モードを終了するよう要求します。この場合、セルフ・リフレッシュ・モードの終了には、数マイクロ秒追加で時間がかかります。

このモードでは、RCC には以下の 2 つのレベルのクロック・ゲーティングが備わっています。

RCC は、DDR の状態に応じて、プロセッサの状態とは無関係に、DDRC クロックのゲーティングを動的に制御します。

RCC は、DDR の状態に応じて、プロセッサの状態とは無関係に、DDRC および DDRPHYC クロックのゲーティングを動的に制御します。このモードは、DDRPHYC の DLL がバイパスされている場合にのみ使用できます。たとえば、125MHz の範囲で動作する LPDDR2 に使用できます。

DDRC のセルフ・リフレッシュ・モード制御 (2/2)

- ハードウェア・セルフ・リフレッシュ (HSRx) :
 - 以下の場合、DDRC は自動的にセルフ・リフレッシュ・モードに移行:
 - インタフェースが DDRCxEN ビットで無効にされた場合
 - MPU が CRun モードから CSleep モードに遷移した場合
 - MPU が CRun モードから CStop モードに遷移した場合
 - 以下の場合、DDRC は自動的にセルフ・リフレッシュ・モードを終了:
 - インタフェースが DDRCxEN ビットで有効にされた場合
 - MPU が CSleep モードまたは CStop モードを終了した場合
 - DDRC がセルフ・リフレッシュ・モードの終了を要求した場合これは、別のマスタが AXI ポートの 1 つにアクセスしている場合に起き得ます。
- HSR1 では、RCC は AXI 低電力インタフェースを使用して、DDRC のゲーティングを動的に制御します。
- HSR2 では、RCC は AXI 低電力インタフェースを使用して、さらに DDRPHYC のゲーティングも動的に制御します。



ハードウェア・セルフ・リフレッシュ・モードでは、DDRC および DDRPHYC クロック・ゲーティングを制御するために、AXI 低電力インタフェースを使用しています。

ハードウェア・セルフ・リフレッシュ・モードでは、以下の場合、DDR は自動的にセルフ・リフレッシュ・モードになります。

インタフェースが DDRCxEN ビットで無効にされた場合
MPU が CRun モードから CSleep モードに遷移した場合
MPU が CRun モードから CStop モードに遷移した場合

以下の場合、DDR は自動的にセルフ・リフレッシュ・モードを終了します。

インタフェースが DDRCxEN ビットで有効にされた場合
MPU が CSleep モードまたは CStop モードを終了した場合
DDRC がセルフ・リフレッシュ・モードの終了を要求した場合。これは、別のマスタが AXI ポートの 1 つにアクセスしている場合に起き得ます。

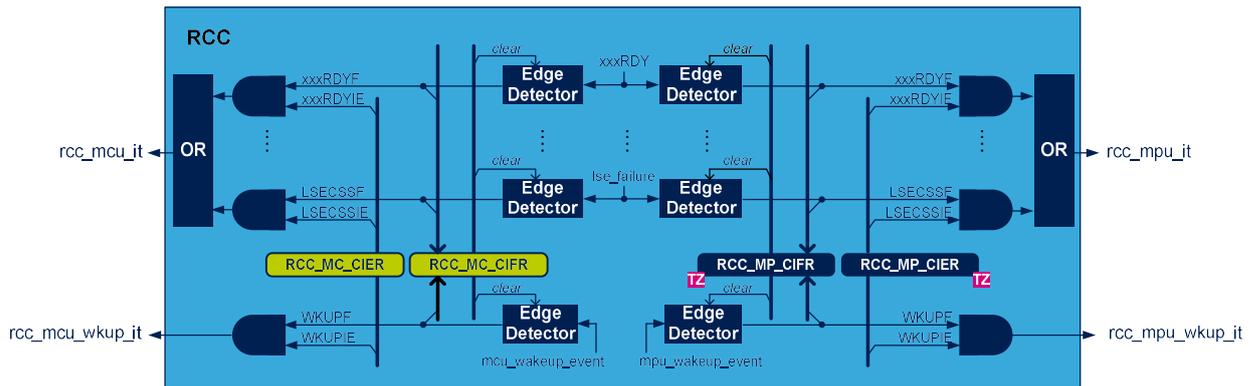
このモードは、MPU が低電力モードのときに DDR をセルフ・リフレッシュ・モードに設定する傾向がありますが、DMA などの他のマスタは、必要に応じて DDR のセルフ・リフレッシュ・モードを終了できます。

このモードでは RCC には以下の 2 つのレベルのクロック・ゲーティングも備わっています。

RCC は DDRC クロックのゲーティングを動的に制御します。

RCC は DDRC および DDRPHYC クロックのゲーティングを動的に制御します。このモードは、DDRPHYC の DLL がバイパスされている場合にのみ使用できます。たとえば、125MHz の範囲で動作する LPDDR2 に使用できます。

- 各割込みイベントの供給先:
 - MPU 向けの割込みインタフェース(セキュリティ保護可能)
 - MCU 向けの割込みインタフェース
- 各割込みインタフェースは 2 つの割込みラインを提供



RCC では、次の 2 つの割込みインタフェースが提供されています。

- MPU 向けに 1 つ
- MCU 向けに 1 つ

MPU 割込みインタフェースは、必要に応じて TZEN ビットを介してセキュア・モードに切り替えることができます。

各プロセッサに対して、RCC は汎用割込みとウェイクアップ割込みを提供します。

割込みイベント	説明
LSE クロック・セキュリティ・システム	LSE 発振回路で障害が検出されたときにセットされます
PLL1 レディ割込みフラグ	PLL1 ロックによるクロック・レディ
PLL2 レディ割込みフラグ	PLL2 ロックによるクロック・レディ
PLL3 レディ割込みフラグ	PLL3 ロックによるクロック・レディ
PLL4 レディ割込みフラグ	PLL4 ロックによるクロック・レディ
HSE レディ	HSE 発振回路によるクロック・レディ
HSI レディ	HSI 発振回路によるクロック・レディ
CSI レディ	CSI 発振回路によるクロック・レディ
LSE レディ	LSE 発振回路によるクロック・レディ
LSI レディ	LSI 発振回路によるクロック・レディ
MPU/MCU ウェイクアップ	MPU または MCU のアクティブ化が必要な場合にセットされます



イベントには主に下記の 3 種類があります。

- クロック障害イベント
- 発振回路および PLL のレディ・イベント
- ウェイクアップ・イベント

- 必要に応じて、このペリフェラルにリンクされている下記のトレーニングを参照してください。
 - STM32MP1 電源制御 (PWR)
 - STM32MP1 非同期割込みおよびイベント・コントローラ (AIEC)



このトレーニングに加えて、電源制御および割込みコントローラのトレーニングが役立つことがあります。