



# インテル® Xeon Phi™ コプロセッサ (開発コード名: Knights Corner) の パフォーマンス・モニタリング・ユニット

---

改訂: 1.01

最終更新日: 2012 年 7 月 10 日

資料番号: 327357-001JA

本資料に掲載されている情報は、インテル製品の概要説明を目的としたものです。本資料は、明示されているか否かにかかわらず、また禁反言によらずにかかわらず、いかなる知的財産権のライセンスも許諾するものではありません。製品に付属の売買契約書『Intel's Terms and Conditions of Sale』に規定されている場合を除き、インテルはいかなる責任を負うものではなく、またインテル製品の販売や使用に関する明示または黙示の保証 (特定目的への適合性、商品適格性、あらゆる特許権、著作権、その他知的財産権の非侵害性への保証を含む) に関してもいかなる責任も負いません。

「ミッション・クリティカルなアプリケーション」とは、インテル製品がその欠陥や故障によって、直接的または間接的に人身傷害や死亡事故が発生するようなアプリケーションを指します。そのようなミッション・クリティカルなアプリケーションのためにインテル製品を購入または使用する場合は、直接的か間接的にかかわらず、あるいはインテル製品やそのいかなる部分の設計、製造、警告にインテルまたは委託業者の過失があったかどうかにかかわらず、製造物責任、人身傷害や死亡の請求を起因とするすべての賠償請求費用、損害、費用、合理的な弁護士費用をすべて補償し、インテルおよびその子会社、委託業者および関連会社、およびそれらの役員、経営幹部、従業員に何らの損害も与えないことに同意するものとします。

インテル製品は、予告なく仕様や説明が変更されることがあります。機能または命令の一覧で「留保」または「未定義」と記されているものがありますが、その「機能が存在しない」あるいは「性質が留保付である」という状態を設計の前提にしないでください。これらの項目は、インテルが将来のために留保しているものです。インテルが将来これらの項目を定義したことにより、衝突が生じたり互換性が失われたりしても、インテルは一切責任を負いません。この情報は予告なく変更されることがあります。この情報だけに基づいて設計を最終的なものとししないでください。

本資料で説明されている製品には、エラッタと呼ばれる設計上の不具合が含まれている可能性があり、公表されている仕様とは異なる動作をする場合があります。現在確認済みのエラッタについては、インテルまでお問い合わせください。

最新の仕様をご希望の場合や製品をご注文の場合は、お近くのインテルの営業所または販売代理店にお問い合わせください。

本資料で紹介されている資料番号付きのドキュメントや、インテルのその他の資料を入手するには、1-800-548-4725 (アメリカ合衆国) までご連絡いただくか、[インテルの Web サイト](#)を参照してください。

Intel、インテル、Intel ロゴ、Xeon、Xeon Phi は、アメリカ合衆国および / またはその他の国における Intel Corporation の商標です。

\* その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

© 2013 Intel Corporation. 無断での引用、転載を禁じます。

## 目次

---

1	インテル® Xeon Phi™ コプロセッサのパフォーマンス・モニタリング・ユニット	4
1.1	はじめに	4
1.2	コア・パフォーマンス・モニタリング・ユニット (PMU) とイベントの概要	4
1.3	パフォーマンス・モニタリング・プログラミング手法	5
1.3.1	想定されるサンプリング手法	5
1.3.2	コア PMU におけるリング 0 とリング 3 によるプログラミングの可能性	5
1.3.3	プロダクション・ドライバーとインストールされたドライバー	6
1.4	コア PMU のプログラミング	6
1.4.1	基本的なプログラミング	7
1.4.2	コア PMU 命令	7
1.4.3	コア PMU 制御レジスター	8
1.4.4	ウォームリセット/INIT 動作	21

# 1 インテル® Xeon Phi™ コプロセッサのパフォーマンス・モニタリング・ユニット

## 1.1 はじめに

複雑なアプリケーションのパフォーマンス・チューニングにおける課題の多くは、特定のアーキテクチャーではコードがどのように実行されるかという、アーキテクチャー固有の洞察を開発者がほとんど (あるいは全く) 得られないことに起因します。特に、ドライバーとアプリケーションのパフォーマンス・チューニングの質が製品の成功に直結するハイパフォーマンス・コンピューティング・アプリケーションでは、これが大きな課題となります。現代のソフトウェアのパフォーマンス・チューニングは、ハードウェアで利用可能な各種パフォーマンス・イベントとカウンターを利用する必要があります。優れたパフォーマンス解析ツール (インテル® VTune™ Amplifier XE など) は、これらのハードウェア・カウンターを利用して、アプリケーションの正確かつ高度なパフォーマンス解析を提供します。

## 1.2 コア・パフォーマンス・モニタリング・ユニット (PMU) とイベントの概要

インテル® Xeon Phi™ コプロセッサのコア PMU は、インテル® Pentium® プロセッサ (別名 P54C) の PMU を基にしています。インテル® Pentium® プロセッサの 42 個のパフォーマンス・イベントのほとんどは、インテル® Xeon Phi™ コプロセッサでも利用できます。コア PMU は、インテル® Pentium® Pro プロセッサ形式 (「P6 形式」) のプログラミング・インターフェイスにアップグレードされています。以下は、アップグレードされたコア PMU の概要です。

1. コア/PMU の配列: インテル® Xeon Phi™ コプロセッサには、物理コアごとに個別にプログラム可能なコア PMU があります。つまり、物理的なインテル® Xeon Phi™ コプロセッサの x86 コアの配列と 1 対 1 の “コア PMU の配列” があると考えられます。
2. 4 ウェイスレッド: インテル® Xeon Phi™ コプロセッサの各コアは、同時に 4 つのハードウェア・スレッドを処理することができます。
3. 各コアはスレッドごとに 2 カウンター: コアあたり 2 つのカウンターではなく、コア PMU ごとにスレッドあたり 2 つのカウンターがあります。複数のスレッドを使用するイベントまたはスレッドを全く使用しないイベントもカウンターをインクリメントします。
4. P6 形式の PMU 選択と制御: コア PMU のプログラミング・インターフェイスは、インテル® Pentium® Pro プロセッサ (P6) のインターフェイス形式 (「P6 形式」) にアップグレードされています。
5. 新しいコア PMU 命令: インテル® Xeon Phi™ コプロセッサの命令セットには、アプリケー

ション・レベル・コード (リング 3 コード) 向けにパフォーマンス・モニタリング機能を拡張するいくつかの命令が追加されています。

6. 新しいコア PMU イベント: インテル® Xeon Phi™ コプロセッサのベクトル・プロセッシング・ユニット (VPU) などの新しい機能ユニットに対する情報を提供するため、新しいパフォーマンス・モニタリング・イベントが追加されています。

本資料では、それぞれの変更について詳しく述べます。

## 1.3 パフォーマンス・モニタリング・プログラミング手法

### 1.3.1 想定されるサンプリング手法

#### 1.3.1.1 手動インストルメンテーション

パフォーマンス・カウンター設定ツール (リング 0) を使って、コードのクリティカル・セクションにリング 3 のパフォーマンス・モニタリング命令 (RDTSC、RDPMC、SPFLT など) を手動で記述することで、詳細なパフォーマンス情報を得られます。インテル® Xeon Phi™ コプロセッサの PMU は、リング 3 からはアクセスできません。コア PMU をアクセスする設定ツール (リング 0) を利用して、監視するハードウェア・イベントを設定する必要があります。

#### 1.3.1.2 タイムベース・サンプリング

インテル® VTune™ Amplifier XE のようなパフォーマンス・モニタリング・ツールは、インテル® Xeon Phi™ コプロセッサのマイクロオペレーティング・システムへリング 0 のアクセスが可能なソフトウェア・スタック上に構築されます。このようなツールは、一定の間隔でプロセッサに割り込みをかけ、イベント数を収集し、必要に応じて PMU カウンターをリセットします。

#### 1.3.1.3 イベントベース・サンプリング

インテル® VTune™ Amplifier XE のようなパフォーマンス・モニタリング・ツールは、イベントベースのメカニズムを用いてパフォーマンス・モニタリング情報を収集することができます。通常、これらのツールは、オーバーフロー割り込みメカニズムを利用します。具体的には、PMU カウンターを「予測可能なオーバーフローの少し前」の値にリセットし、任意のクロックレートでオーバーフロー割り込みを生成します。イベントが発生するとオーバーフロー割り込みが生成されるため、マシンのすべての状態がサンプリングされ、マシンの状態を統計的に取得することができます。

### 1.3.2 コア PMU におけるリング 0 とリング 3 によるプログラミングの可能性

オリジナルのインテル® Pentium® プロセッサでは、パフォーマンス・カウンターへのアクセスはリング 0 ドライバーのみに制限されていました。つまり、エンドユーザーは、インテル® VTune™ Amplifier XE などのフロントエンド・ツールが必要でした。しかし、インテル® Pentium® Pro プロセッサ (P6) 以降のアーキテクチャーでは、RDPMC (パフォーマンス・モニター・カウンターの読み込み)

などの特別な命令を利用して、リング 3 のアプリケーション・レベルのコードでもカウンター値を読み込めるようになりました。PMU の設定は引き続きリング 0 に制限されますが、最適化が重要な場合は、RDPMC 命令を使ってコードセグメントをインストルメントし、ドライバーを介さずに直接パフォーマンス情報を取得することができます。

インテル® Xeon Phi™ コプロセッサのコア PMU は P6 と類似しており、リング 3 のアプリケーション・レベルのコードは RDPMC 命令によってコア・パフォーマンス・カウンターを照会できます。カウンターの設定にはリング 0 ベースのインターフェイスが必要です。また、アプリケーション・レベルのコードは、新しい SPFLT 命令によってスレッドベースのパフォーマンス・フィルター機能を制御できます。

### 1.3.3 プロダクション・ドライバーとインストルメントされたドライバー

一般に、コア PMU へのアクセスは、プロダクション・マイクロオペレーティング・システムを含む、プロダクション・ドライバー・スタックにおいて提供されます。

## 1.4 コア PMU のプログラミング

インテル® Xeon Phi™ コプロセッサのコア PMU は、スレッドあたり 2 つの汎用イベントカウンターを提供します。これらのカウンターはすべて 40 ビット精度であり、イベント数が 40 ビットで表現できない場合は、パフォーマンス・モニタリング・インターフェイス (PMI) を呼び出すようにカウンターを設定することができます。また、対応する P6 形式のパフォーマンス・イベント・セレクション・レジスターでイベントを選択し、各汎用カウンターの動作を設定できます。

コア PMU 操作のプログラミングに使用するモデル固有レジスター (MSR) は、リング 0 から RDMSR および WRMSR 命令によって読み書きできます。これらの機能と MSR については、この節で説明します。さらに、新しい命令により、タイムスタンプ・カウンター、パフォーマンス・カウンター、パフォーマンス・フィルター・マスク・レジスターへのリング 3 からのアクセスが提供されます。ここでは、これらについても述べます。

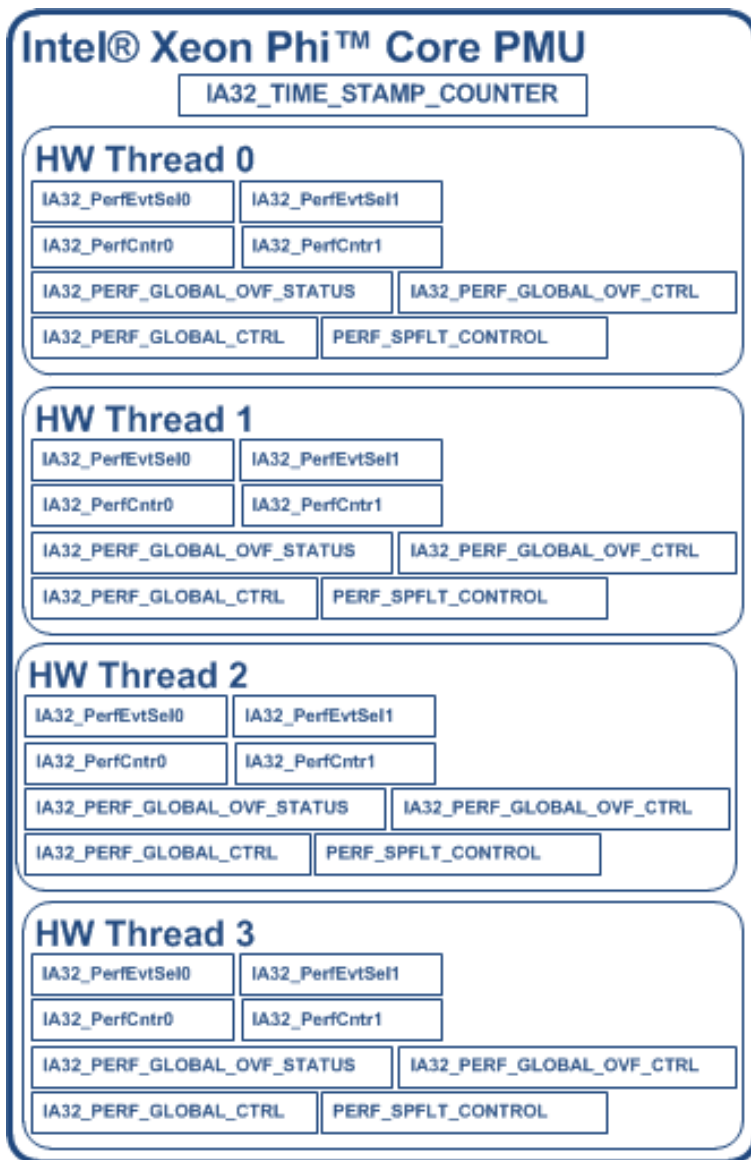


図 1-1: コア PMU アーキテクチャー

### 1.4.1 基本的なプログラミング

コア PMU を使ってイベントを簡単に監視できます。それには次のレジスターを設定します。選択されたイベントに対応するビットとマスター PMU レジスターの AND (論理積) が 1 になると、イベント・モニタリング・ハードウェアはカウンターへのイベントのフィルタリングを開始します。

- IA32\_PerfEvtSelX と対応するカウンターの初期化
- IA32\_PERF\_GLOBAL\_CTRL

### 1.4.2 コア PMU 命令

表 1-1 は、コア PMU の制御と照会を行うため、リング 0 とリング 3 コードで使用され、実行スレッ

ドに適用される命令の一覧です。

表 1-1: コア PMU 命令

命令	説明	権限 (CPL)	スレッド固有	入力	出力
RDMSR	モデル固有レジスタの読み込み。リング 0 コードでコア PMU レジスタの読み込みに使用されます。	リング 0	はい	ECX: MSR のアドレス	EDX:EAX = 64 ビットの MSR 値
WRMSR	モデル固有レジスタへの書き込み。リング 0 コードでコア PMU レジスタへの書き込みに使用されます。	リング 0	はい	EDX:EAX = 64 ビットの MSR 値 ECX: MSR のアドレス	なし
RDTSC	タイムスタンプ・カウンターの読み込み。現在のタイムスタンプ・カウンター値を読み込みます。	リング 0 ~ 3	いいえ	なし	EDX:EAX = 64 ビットのタイムスタンプ値
RDPMC	パフォーマンス・モニタリング・カウンターの読み込み。フィルタリングされた PMU カウンターを含む、任意の PMU パフォーマンス・モニタリング・カウンターのカウント数を読み込みます。	リング 0 ~ 3	はい	ECX: カウンター番号 0x0: IA32_PerfCnt0 0x1: IA32_PerfCnt1	EDX:EAX = ゼロ拡張された 40 ビットのカウンター値
SPFLT	カウンターの有効/無効を示すユーザー設定フラグを設定します。	リング 0 ~ 3	はい	任意の GPR[0]: 0x0: クリア (無効) 0x1: 設定 (有効)	PERF_SPFLT_CONTROL の USER_PREF ビットを設定/クリア

RDMSR、WRMSR、RDTSC、および RDPMC レジスタについては、『Intel® 64 and IA-32 Architectures Software Developer Manual』(英語)で詳しく説明されています。Intel® Xeon Phi™ コプロセッサ固有の特徴として、RDTSC 命令は 4 ~ 5 クロックサイクルで実行できるように拡張されています。

SPFLT は、ソフトウェア・スレッドによりパフォーマンス・カウンターの有効/無効を細かく制御できるユニークな命令です。任意のコード領域でコードをインストルメントしてカウンターを有効/無効にすることができます。ソフトウェアは、カウンターを有効/無効にすることを指定できるだけで、特定のカウンターを制御することはできません(この動作は仮想化をサポートします)。

### 1.4.3 コア PMU 制御レジスタ

表 1-2 は、コア PMU 操作のプログラミングに使用されるモデル固有レジスタの一覧です。



表 1-2: コア PMU 制御レジスター

レジスターアドレス		レジスター名	説明	スレッド化?	幅
16 進表記	10 進表記				
0x10	16	IA32_TIME_STAMP_COUNTER	タイムスタンプ・カウンター	いいえ	64
0x20	32	IA32_PerfCnt0	イベントのカウンタ数。 コア PMU カウンター 0。	はい	40
0x21	33	IA32_PerfCnt1	イベントのカウンタ数。 コア PMU カウンター 1。	はい	40
0x28	40	IA32_PerfEvtSel0	IA32_PerfCnt0 用のパフォーマンス・イベントの選択と設定レジスター	はい	32
0x29	41	IA32_PerfEvtSel1	IA32_PerfCnt1 用のパフォーマンス・イベントの選択と設定レジスター	はい	32
0x2C	44	PERF_SPFLT_CONTROL	SPFLT 制御レジスター。この MSR は、ソフトウェアが IA32_PerfCntN を有効/無効にすることを許可するかどうか、SPFLT 命令を制御します。	はい	64
0x2D	45	IA32_PERF_GLOBAL_STATUS	カウンター・オーバーフロー・ステータス。この MSR は、読み込み専用で、すべてのカウンターのオーバーフロー・ステータスを表示します。各ビットはスティッキー・ビットとして実装され、カウンター・オーバーフローによって設定されます。	はい	32
0x2E	46	IA32_PERF_GLOBAL_OVF_CTRL	カウンター・オーバーフロー制御。この MSR は、書き込み専用で、カウンター・オーバーフロー・ステータス・レジスターのオーバーフロー・ステータスをクリアします。設定されている各ビットに対応するオーバーフロー・ステータスがクリアされます。	はい	32
0x2F	47	IA32_PERF_GLOBAL_CTRL	マスター PMU による PMU のグローバルな有効化/無効化。ビットが設定されている場合、コア PMU は各パフォーマンス・イベント・セレクション・レジスター(それぞれ個別に有効/無効にされる)で設定されているイベントをカウントします。ビットが設定されていない場合、パフォーマンス・モニタリングは無効になります。このレジスターは、タイムスタンプ・カウンターの操作には影響しません。	はい	32

### 1.4.3.1 IA32\_TIME\_STAMP\_COUNTER

説明: タイムスタンプ・カウンター

アドレス: 0x10

デフォルト値: 0x0000000000000000

通常のアクセス: RW

サイズ: 64 ビット

スレッド化対応: いいえ

注: タイムスタンプ・カウンター (TSC) は、コア・クロックレートでインクリメントされるため、ウォールクロックとしての使用には適しません。TSC はコアあたり 1 つのみで、すべてのスレッドによって共有されます。

表 1-3: IA32\_TIME\_STAMP\_COUNTER レジスタのレイアウト

ビット	説明
63:0	タイムスタンプ・カウンター: リセットからのコア・クロックサイクル数

### 1.4.3.2 IA32\_PerfCnt0/1

説明: パフォーマンス・イベントのカウンタ数

アドレス: 0x20 (0)、0x21 (1)

デフォルト値: 0x000000000000

通常のアクセス: RW

サイズ: 40 ビット

スレッド化対応: はい

注: 各 40 ビットカウンタは、カウンタが有効で、指定されたハードウェア・イベントのカウンタを許可されている場合、そのハードウェア・イベントが発生するとインクリメントされます。このレジスタへの読み書きは、40 ビットすべてを設定します。この動作は、下位 32 ビットのみ書き込み、上位 8 ビットは符号拡張する一部の IA32 プロセッサとは異なります。

表 1-4: IA32\_PerfCnt0/1 レジスタのレイアウト

ビット	説明
63:40	予約済み
39:0	イベントのカウンタ数: 関連付けられたパフォーマンス・イベント・セレクション・レジスタで指定されたイベントのカウンタ数

### 1.4.3.3 IA32\_PerfEvtSel0/1

説明: パフォーマンス・イベント・セレクション・レジスタ

アドレス: 0x28 (0)、0x29 (1)

デフォルト値: 0x00000000

通常のアクセス: RW

サイズ: 32 ビット

スレッド化対応: はい

注: なし。

表 1-5: IA32\_PerfEvtSel0/1 レジスタのレイアウト

ビット	説明
31:24	カウンターマスク (CMASK): 非ゼロの場合、プロセッサはこのマスクと 1 サイクルのイベントのカウント数を比較します。イベントのカウント数がこのマスク以上の場合、カウンタは 1 だけインクリメントされます。そうでない場合、カウンタはインクリメントされません。このマスクを使用して、イベントがクロックあたり複数回発生した場合のみカウントすることもできます (例えば、クロックあたり 2 つの命令がリタイアされた場合)。カウンターマスクが 0 の場合、カウンタは各サイクルで発生したイベントの回数だけインクリメントされます。
23	反転 (INV): 設定されている場合、カウンターマスクの比較結果を反転し、より大きいかあるいは小さいかの比較も行えるようにします。
22	有効化カウンタ (EN): 設定されている場合、パフォーマンス・イベントのカウントは IA32_PERF_GLOBAL_CTRL の対応するビットが設定されている場合のみ有効になります。
21	スレッド・カウント・モード: 設定されていない場合、このスレッドのコンテキストに一致するイベントをカウントします。設定されている場合、すべてのスレッドのコンテキストに一致するイベントをカウントします。スレッド固有のコンテキストがないイベントは、スレッド・カウント・モードに関係なく、常にカウントされます。
20	APIC 割り込みの有効化 (INT): 設定されている場合、プロセッサはカウンタ・オーバーフローが発生すると、ローカル APIC を介して例外を生成します。
19	予約済み
18	境界検出 (E): 設定されている場合、イベントの境界検出を有効にします。プロセッサは、ほかのフィールドによって表現可能なすべての条件のアサート停止からアサートへの遷移数をカウントします。連続するアサートは識別されません。これにより、ソフトウェアは、特定の状態 (例えば、割り込みが処理されるまでの待機時間) で費やされた時間を計測できるだけでなく、その状態で費やされた平均時間も計測することができます。
17	オペレーティング・システム・モード (OS): プロセッサがリング 0 で動作しているときにイベントのカウントを有効にします。このフラグと USR フラグを一緒に使用することで、すべての権限レベルでイベントのカウントを許可できます。
16	ユーザーモード (USR): プロセッサがリング 1、2、または 3 で動作しているときにイベントのカウントを有効にします。このフラグと OS フラグを一緒に使用することで、すべての権限レベルでイベントのカウントを許可できます。
15:8	ユニットマスク (UMASK): イベントの選択フィールドで選択されたイベントの論理ユニットをより細かく制限し、特定のアーキテクチャーの条件を検出するようにします。例えば、キャッシュイベントで、このマスクを使って MESI プロトコルのキャッシュ状態を検出できます。
7:0	イベントの選択: この 8 ビットのフィールドを使用して、ハードウェア・イベントをエンコードできます。不正なイベントのエンコーディングは、ハードウェアの例外やエラーを引き起こしません。また、カウンタはインクリメントされません。

表 1-6: イベント一覧

FUB	UMASK	イベントコード	イベント名のニーモニック	説明
P54C	0x00	0x00	DATA_READ	Kユニット (L1) によってコミットされたメモリーデータ読み込みの数。プリフェッチ命令に関連したキャッシュアクセスはA0 ステッピングに含まれます。
P54C	0x00	0x01	DATA_WRITE	Kユニット (L1) によってコミットされたメモリーデータ書き込みの数。ストリーミング・ストア (L1 のヒット/ミス)、キャッシュ可能な書き込みの一部、UC プロモーションを含みます。
P54C	0x00	0x02	DATA_PAGE_WALK	データ・ページ・ウォークの数。
P54C	0x00	0x03	DATA_READ_MISS	キャッシュ可能か、キャッシュ可能でないかに関係なく、内部データキャッシュのミスを引き起こしたメモリー読み込みアクセスの数。プリフェッチ命令に関連したキャッシュアクセスは含まれません。
P54C	0x00	0x04	DATA_WRITE_MISS	キャッシュ可能かどうかに関係なく、内部データキャッシュのミスを引き起こしたメモリー書き込みアクセスの数。キャッシュ可能でないミスは含まれません。
P54C	0x00	0x06	DATA_CACHE_LINES_WRITTEN_BACK	原因にかかわらず、ライトバックされたダーティーラインの数。
P54C	0x00	0x09	MEMORY_ACCESSES_IN_BOTH_PIPES	パイプラインの両方のパイプでペアにされたデータメモリー読み込み/書き込みの数。
P54C	0x00	0x0A	BANK_CONFLICTS	実際のバンク競合の数。
	0x00	0x0C	CODE_READ	キャッシュ可能か、キャッシュ可能でないかに関係なく、命令読み込みの数。
P54C	0x00	0x0D	CODE_PAGE_WALK	コード・ページ・ウォークの数。

P54C	0x00	0x0E	CODE_CACHE_MISS	キャッシュ可能か、キャッシュ可能でないかに関係なく、内部コードキャッシュのミスを引き起こした命令読み込みの数。
P54C	0x00	0x11	L1_DATA_PF1	L1 から見たデータの vprefetch0 要求の数。
P54C	0x00	0x12	BRANCHES	分岐の数 (実際に分岐したかどうかに関係ない)。条件付き分岐、ジャンプ、呼び出し、リターン、ソフトウェアによる割り込みリターンを含みます。
P54C	0x00	0x15	PIPELINE_FLUSHES	分岐実行、分岐予測ミス、例外、割り込み、セグメント記述子のロードで BTB ミスによって発生したパイプライン・フラッシュの数。
P54C	0x00	0x16	INSTRUCTIONS_EXECUTED	実行された命令の数 (最大でクロックあたり 2 命令)。
P54C	0x00	0x17	INSTRUCTIONS_EXECUTED_V_PIPE	V パイプで実行された命令の数。 このイベントは、ペアにされた命令の数を示します。
P54C	0x00	0x1C	L1_DATA_PF1_MISS	L1 から見たデータの vprefetch0 要求のうち L1 ミスを引き起こした数。 vprefetch1 要求は含まれません。L1_DATA_PF1_DROP でカウントされます。
P54C	0x00	0x1E	L1_DATA_PF1_DROP	L1 から見たデータの vprefetch0 要求のうち何らかの理由で取り消された数。要求されたアドレスが別の要求と一致する場合や UC メモリー型の場合、vprefetch0 は取り消されることがあります。
P54C	0x00	0x1F	PIPELINE_AGI_STALLS	アドレス生成インターロック (AGI) ストールの数。同じクロックで U パイプラインと V パイプラインの両方で AGI が発生すると、このイベントは 2 回通知されます。

P54C	0x00	0x20	L1_DATA_HIT_INFLIGHT_PF1	vprefetch0 にヒットしたデータ要求の数。 vprefetch0 は、必ずしもデータ要求と同じスレッドによって発行されたとは限りません。
P54C	0x00	0x21	PIPELINE_SG_AGI_STALLS	vscatter* および vgather* 命令によるアドレス生成インターロック (AGI) ストールの数。
P54C	0x00	0x28	DATA_READ_OR_WRITE	メモリーデータの読み込み/書き込みの数 (内部データキャッシュのヒットとミスを含めた数)。プリフェッチ命令に関連した読み込みキャッシュアクセスは A0 ステッピングに含まれます。
P54C	0x00	0x29	DATA_READ_MISS_OR_WRITE_MISS	キャッシュ可能か、キャッシュ可能でないかに関係なく、内部データキャッシュのミスを引き起こしたメモリー読み込み/書き込みアクセスの数。
P54C	0x00	0x2A	CPU_CLK_UNHALTED	プロセッサが停止しなかったサイクル数。
P54C	0x00	0x2B	BRANCHES_MISPREDICTED	BTB ヒットで発生した分岐予測ミスの数。BTB ミスは予測を伴わないため、分岐予測ミスとは見なされません。
P54C	0x00	0x2C	MICROCODE_CYCLES	マイクロコードの実行に費やしたサイクル数。マイクロコードの実行中、ほかのすべてのスレッドはストールします。
P54C	0x00	0x2D	FE_STALLED	フロントエンドが先に進めなかったサイクル数。読み込み-変更-書き込み命令など、パイプラインの進行を遅らせ、フロントエンドにバックプレッシャーを与えるすべてのマルチサイクル命令が含まれます。フロントエンドで発行する命令がなかったサイクルも含まれます。

P54C	0x00	0x2E	EXEC_STAGE_CYCLES	完了した E ステージのサイクル数。マルチサイクルの E ステージ命令に費やされたサイクルも含まれます。FPU または VPU パイプラインに送られる命令の場合、このイベントは E ステージで整数に費やされたサイクルをカウントします。
P54C	0x00	0x37	L1_DATA_PF2	L1 から見たデータの vprefetch1 要求の数。これには、コアによって取り消された要求も含まれるため、必ずしも L2 から見た数と同じになるとは限りません。要求されたアドレスが別の要求と一致する場合や UC メモリー型の場合、vprefetch1 はコアによって取り消されることがあります。
P54C	0x00	0x38	L2_DATA_PF1_MISS	L2 から見たデータの vprefetch0 要求のうち L2 ミスを引き起こした数。
P54C	0x00	0x3A	LONG_DATA_PAGE_WALK	“長い” データ・ページ・ウォーク (つまり、L2 uTLB ミスも引き起こしたページウォーク) の数。 DATA_PAGE_WALK イベントのサブセットです。
P54C	0x00	0x3B	LONG_CODE_PAGE_WALK	“長い” コード・ページ・ウォーク (つまり、L2 uTLB ミスも引き起こしたページウォーク) の数。 DATA_CODE_WALK イベントのサブセットです。
CRI	0x10	0xC8	L2_READ_HIT_E	E 状態の L2 読み込みヒット。A0 ステッピングでのプリフェッチが含まれる可能性があります。
CRI	0x10	0xC9	L2_READ_HIT_M	M 状態の L2 読み込みヒット
CRI	0x10	0xCA	L2_READ_HIT_S	S 状態の L2 読み込みヒット
CRI	0x10	0xCB	L2_READ_MISS	L2 読み込みミス。同じアドレスへのプリフェッチ要求とデマンド要求により、実際よりも多く (2 重に) カウントされます。

CRI	0x10	0xCC	L2_WRITE_HIT	L2 読み込みヒット。A0 ステッピングで実際よりも少なくカウントされることがあります。
CRI	0x10	0xD7	L2_VICTIM_REQ_WITH_DATA	L2 がビクティム要求を受け取り、データを送った数。
CRI	0x10	0xE3	SNP_HITM_BUNIT	L2 キャッシュミスになり、別のL2 キャッシュでヒットしたコード読み込みアクセスの数。DATA アクセスとして開始され、読み込みミスにプロモートされたものも含まれます。
CRI	0x10	0xE6	SNP_HIT_L2	BUNIT におけるスヌープ・ヒット・モディファイド (HITM) の数。
CRI	0x10	0xE7	SNP_HITM_L2	L2 におけるスヌープヒットの数。
CRI	0x10	0xF0	L2_CODE_READ_MISS_CACHE_FILL	L2 におけるスヌープ・ヒット・モディファイド (HITM) の数。
CRI	0x10	0xF1	L2_DATA_READ_MISS_CACHE_FILL	L2 キャッシュミスになり、別のL2 キャッシュでヒットしたデータ読み込みアクセスの数。CODE アクセスとして開始され、読み込みミスにプロモートされたものも含まれます。
CRI	0x10	0xF2	L2_DATA_WRITE_MISS_CACHE_FILL	L2 キャッシュミスになり、別のL2 キャッシュでヒットしたデータ書き込み (RFO) アクセスの数。
CRI	0x10	0xF5	L2_CODE_READ_MISS_MEM_FILL	L2 キャッシュミスになり、メインメモリーでヒットしたコード読み込みアクセスの数。DATA アクセスとして開始され、読み込みミスにプロモートされたものも含まれます。
CRI	0x10	0xF6	L2_DATA_READ_MISS_MEM_FILL	L2 キャッシュミスになり、メインメモリーでヒットしたデータ読み込みアクセスの数。CODE アクセスとして開始され、読み込みミスにプロモートされたものも含まれます。



CRI	0x10	0xF7	L2_DATA_WRITE_MISS_MEM_FILL	L2 キャッシュミスになり、メインメモリーでヒットしたデータ書き込み (RFO) アクセスの数。
CRI	0x10	0xFC	L2_DATA_PF2	L2 から見たデータの vprefetch1 要求の数。A0 ステッピングでの vprefetch1 ヒットのみカウントされます。
CRI	0x10	0xFD	L2_DATA_PF2_DROP	L2 から見たデータの vprefetch1 要求のうち何らかの理由で取り消された数。
CRI	0x10	0xFE	L2_DATA_PF2_MISS	L2 から見たデータの vprefetch1 要求のうち L2 ミスを引き起こした数。vprefetch2 要求は含まれません。L2_DATA_PF2_DROP でカウントされます。
CRI	0x10	0xFF	L2_DATA_HIT_INFLIGHT_PF2	vprefetch1 にヒットしたデータ要求の数。vprefetch1 は、必ずしもデータ要求と同じスレッドによって発行されたとは限りません。
VPU	0x20	0x00	VPU_DATA_READ	発行された読み込みトランザクションの数。一般に、各読み込みトランザクションは、1つの 64B キャッシュラインを読み込みます。アライメントに問題がある場合、複数のキャッシュラインに対する読み込みは、それぞれ別々にカウントされます。
VPU	0x20	0x01	VPU_DATA_WRITE	発行された書き込みトランザクションの数。一般に、各書き込みトランザクションは、1つの 64B キャッシュラインに書き込みます。アライメントに問題がある場合、複数のキャッシュラインに対する書き込みは、それぞれ別々にカウントされます。

VPU	0x20	0x03	VPU_DATA_READ_MISS	VPU L1 データキャッシュの読み込みミスの発生回数をカウントします。
VPU	0x20	0x04	VPU_DATA_WRITE_MISS	VPU L1 データキャッシュの書き込みミスの発生回数をカウントします。
VPU	0x20	0x05	VPU_STALL_REG	レジスター依存による VPU ストールの発生回数をカウントします。RAW、WAW、WAR 依存が含まれます。
VPU	0x20	0x16	VPU_INSTRUCTIONS_EXECUTED	U パイプと V パイプで実行された VPU 命令の数。
VPU	0x20	0x17	VPU_INSTRUCTIONS_EXECUTED_V_PIPE	V パイプでペアにされ、実行された VPU 命令の数。
VPU	0x20	0x18	VPU_ELEMENTS_ACTIVE	発行された VPU 命令に対して (マスクにより) アクティブな要素の累積数。

#### 1.4.3.4 PERF\_SPFLT\_CONTROL

説明: SPFLT 制御レジスター  
 アドレス: 0x2C  
 デフォルト値: 0x00000000\_00000000  
 通常のアクセス: RW  
 サイズ: 64 ビット  
 スレッド化対応: はい  
 注: なし。

表 1-7: PERF\_SPFLT レジスターのレイアウト

ビット	説明
63	ユーザー設定: カウンターを有効/無効にする、ソフトウェアにより制御される設定ビット。0 の場合、ソフトウェアによってカウンターは無効に設定されます。1 の場合、ソフトウェアによってカウンターは有効に設定されます。このビットは、特定の SPFLT 制御カウンターの N ビットが設定されている (つまり、ユーザー設定と SPFLT 制御カウンターの N ビットの AND (論理積) が 1 の) カウンターにのみ影響します。
62:2	予約済み
1	SPFLT 制御カウンター 1: 設定されている場合、SPFLT は IA32_PerfCnt1 のイベントをカウントします。ユーザー設定ビットと組み合わせて使用され、オペレーティング・システムによって制御されます。
0	SPFLT 制御カウンター 0: 設定されている場合、SPFLT は IA32_PerfCnt0 のイベントをカウントします。ユーザー設定ビットと組み合わせて使用され、オペレーティング・システムによって制御されます。

### 1.4.3.5 IA32\_PERF\_GLOBAL\_STATUS

名前: パフォーマンス・カウンター・オーバーフロー・ステータス  
アドレス: 0x2D  
デフォルト値: 0x00000000  
通常のアクセス: RO  
サイズ: 32 ビット  
スレッド化対応: はい

注: 各ビットはスティッキー・ビットとして実装され、カウンター・オーバーフローによって設定されます。このレジスターは、APIC 割り込みが有効でない場合も更新されます。このレジスターの目的は、割り込みサービスルーチンが、どのカウンターがオーバーフローしたかを特定できるようにすることです。

表 1-8: IA32\_PERF\_GLOBAL\_STATUS レジスターのレイアウト

ビット	説明
31:2	予約済み
1	PerfCnt1: カウンター 1 でオーバーフローが発生しました。
0	PerfCnt0: カウンター 0 でオーバーフローが発生しました。

### 1.4.3.6 IA32\_PERF\_GLOBAL\_OVF\_CONTROL

名前: パフォーマンス・カウンター・オーバーフロー・ステータス  
アドレス: 0x2E  
デフォルト値: 0x00000000  
通常のアクセス: WO  
サイズ: 32 ビット  
スレッド化対応: はい  
注: なし。

表 1-9: IA32\_PERF\_GLOBAL\_OVF\_CONTROL レジスターのレイアウト

ビット	説明
63:2	予約済み
1	PerfCnt1: カウンター 1 で発生したオーバーフローをクリアします。
0	PerfCnt0: カウンター 0 で発生したオーバーフローをクリアします。

### 1.4.3.7 IA32\_PERF\_GLOBAL\_CTRL

名前: グローバル・パフォーマンス・カウンター制御  
アドレス: 0x2F  
デフォルト値: 0x00000000  
通常のアクセス: RW  
サイズ: 32 ビット  
スレッド化対応: はい

注: これらのグローバルビットは、IA32\_PerfEvtSelx レジスターのそれぞれのビットと組み合わせて使用されます。

表 1-10: IA32\_PERF\_GLOBAL\_CTRL レジスターのレイアウト

ビット	説明
31:2	予約済み
1	IA32_PerfEvtSel1 によるパフォーマンス・モニタリングの制御: 0 の場合、カウンター 1 は無効になります。1 の場合、カウンター 1 は有効になります。
0	IA32_PerfEvtSel0 によるパフォーマンス・モニタリングの制御: 0 の場合、カウンター 0 は無効になります。1 の場合、カウンター 0 は有効になります。

### 1.4.3.8 コア PMU での PMI 処理

40 ビットのカウンターでは、クロック周波数が 1.0GHz の場合、コア・クロックサイクルあたり 1 回発生するイベントがオーバーフローするには 18 分かかります。実際には、すべての設定可能なハードウェア・イベントはこれよりもかなり低い頻度で発生するため、カウンターのオーバーフローが大きな問題になることはないでしょう。しかし、意図的にカウンターがオーバーフローに近い状態に設定されているサンプリングの利用モデルでは、オーバーフロー割り込みがより頻繁に発生する可能性があります。オーバーフローが発生したときに、ローカル APIC への割り込みをかけるようにカウンターを設定することができます。すべてのカウンターのオーバーフロー・ステータスは、読み込み専用の IA32\_PERF\_GLOBAL\_STATUS ステータスレジスターで確認できます。各スレッドには個別のローカル APIC があり、コア PMU はオーバーフローを発生させたスレッドの APIC に割り込みをかけます。スレッド固有のコンテキストがないイベントの場合は、その時点で実行ステージに存在するスレッドにイベントを送ります。ローカル APIC は、コア PMU からのシグナルを受け取ると、ローカル・ベクトル・テーブル (LVT) でパフォーマンス・カウンター・オーバーフロー・イベント (offset 0x340) を参照します。そして、LVT エントリーから割り込み記述子テーブル (IDT) 内のエントリーを割り出します。IDT エントリーは、オーバーフローをクリアする命令を実行する EMON ドライバーによってインストールされるべきです。対応する割り込みサービスルーチンは、割り込みマスクを解除し、IA32\_PERF\_GLOBAL\_OVF\_CONTROL に書き込みを行って割り込みをクリアし、割り込みの終了 (EOI) を通知します。

### 1.4.3.9 複数または連続するカウンター・オーバーフロー

まれに、複数のカウンターで同時に、あるいは連続してオーバーフローが発生し、ローカル API に複数または連続したパフォーマンス・モニタリング割り込み (PMI) がかけられることがあります。コア PMU とスレッドの APIC の間には 1 ビットの割り込みシグナルしかないため、複数のオーバーフローを知らせることは不可能です。インテル® Pentium® 4 プロセッサ・ファミリーで導入された API 動作では、オーバーフロー割り込みを受け取ると、パフォーマンス・カウンター・レジスターの LVT エントリーを自動的にマスクするため、複数または連続した割り込みの処理がさらに複雑になります。以下は、『Intel® 64 and IA-32 Architectures Software Developer Manual』(英語) からの抜粋です。

「ローカル APIC は、パフォーマンス・モニタリング・カウンター割り込みを処理し、自動的に対応する LVT エントリーのマスクフラグを設定します。このフラグは、ソフトウェアによってクリアされるまで設定されたままになります。」

しかし、IA32\_PERF\_GLOBAL\_STATUS レジスターと割り込みサービスルーチン (ISR) のソフトウェア・メカニズムを組み合わせることで、1 ビットの割り込みシグナルでもこのような状況进行处理することができます。

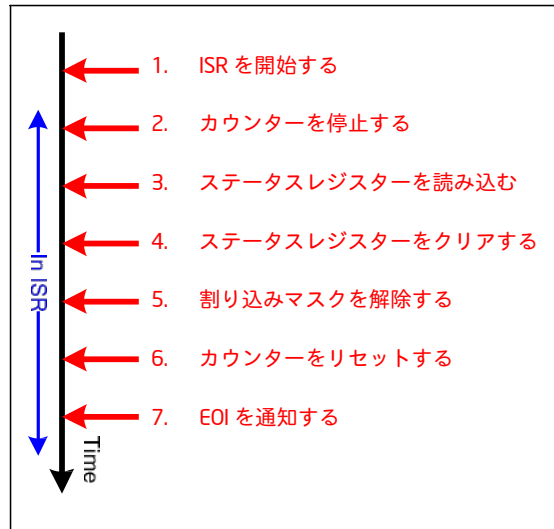


図 1-2: カウンター・オーバーフロー割り込み処理の実行タイムラインの例

図 1-2 は、ISR によって PMI がどのように処理されるかを示しています。これらのステップに従うことで、複数または連続した割り込みが発生した場合であっても、PMI の取り消し/遅延を回避することができます。

以下の点が重要です。

- APIC への 1 ビットの PMI インターフェイスは、カウンター処理が必要になると 0 から 1 に変わります。オーバーフローしたカウンターの情報は、オーバーフロー・ステータス・レジスターに格納されます。そのため、複数の割り込みが同時に発生したり、特定の割り込みが連続して発生しても問題ありません。最初の割り込みが APIC に通知され、残りの割り込みはオーバーフロー・ステータス・レジスターで追跡することができます。
- ISR の役割は、割り込みマスクを解除する前に、オーバーフロー・ステータス・レジスターを 0 にクリアすることだけです。そのため、上記のステップ 2 で、追加の PMI を回避するためカウンターを停止しています。もしステップ 2 でカウンターを停止しなければ、ステップ 3 以降で発生した PMI はクリアされず、APIC への割り込みは 1 のままになります。APIC へ別の割り込み要求を生成できるように、割り込みマスクを解除するときに 0 から 1 に変える必要があります。

#### 1.4.4 ウォームリセット/INIT 動作

コア PMU から見ると、ウォームリセットはコールドリセットと同じです。言い換えると、すべての制御レジスターとカウンターステートがゼロにリセットされます。INIT は、コア PMU 制御レジスターステートやカウンターに影響しません。