



インテル® Itanium® 2 プロセッサ

ハードウェア・デベロッパーズ・マニュアル

2002 年 6 月

© 1999-2003 Intel Corporation
無断での引用、転載を禁じます。
資料番号 : 251109J-001

Web: www.intel.co.jp/jp/developer/ (日本語)
<http://developer.intel.com> (英語)

【輸出規制に関する告知と注意事項】

本資料に掲載されている製品のうち、外国為替および外国為替管理法に定める戦略物資等または役務に該当するものについては、輸出または再輸出する場合、同法に基づく日本政府の輸出許可が必要です。また、米国産品である当社製品は日本からの輸出または再輸出に際し、原則として米国政府の事前許可が必要です。

【資料内容に関する注意事項】

- ・本ドキュメントの内容を予告なしに変更することがあります。
- ・インテルでは、この資料に掲載された内容について、市販製品に使用した場合の保証あるいは特別な目的に合うことの保証等は、いかなる場合についてもいたしかねます。また、このドキュメント内の誤りについても責任を負いかねる場合があります。
- ・インテルでは、インテル製品の内部回路以外の使用は責任を負いません。また、外部回路の特許についても開知いたしません。
- ・本書の情報はインテル製品を使用できるようにする目的でのみ記載されています。

インテルは、製品について「取引条件」で提示されている場合を除き、インテル製品の販売や使用に関して、いかなる特許または著作権の侵害をも含み、あらゆる責任を負わないものとします。

- ・いかなる形および方法によっても、インテルの文書による許可なく、この資料の一部またはすべてを複製することは禁じられています。

インテル、Itanium は、アメリカ合衆国およびその他の国における Intel Corporation またはその子会社の商標または登録商標です。

* 一般にブランド名または商品名は、各社の商標または登録商標です。

© 2002-2003 Intel Corporation. 無断での引用、転載を禁じます。

I2C は、Phillips 社が開発した 2 線式通信バス / プロトコルです。SMBus は、インテルが開発した I2C バス / プロトコルのサブセットです。I2C バス / プロトコルまたは SMBus バス / プロトコルを使用する際、Phillips Electronics(米国ネバダ州) および North American Phillips Corporation を含む各法人のライセンスが必要な場合があります。

目次

第 1 章	はじめに	1-1
1.1	Itanium® 2 プロセッサのシステムバス	1-1
1.2	PAL (Processor Abstraction Layer)	1-1
1.3	用語	1-2
1.4	参考資料	1-2
1.4.1	改訂履歴	1-3
第 2 章	Itanium® 2 プロセッサのマイクロアーキテクチャ	2-1
2.1	概要	2-1
2.1.1	6 命令を同時に処理できる EPIC コア	2-1
2.1.2	プロセッサ・パイプライン	2-2
2.1.3	プロセッサのブロック・ダイアグラム	2-3
2.2	命令処理	2-4
2.2.1	命令プリフェッチと命令フェッチ	2-4
2.2.2	分岐予測	2-5
2.2.3	ディスパーサル・ロジック	2-5
2.3	実行	2-5
2.3.1	浮動小数点ユニット (FPU)	2-5
2.3.2	整数ロジック	2-6
2.3.3	レジスタ・ファイル	2-6
2.3.4	レジスタ・スタック・エンジン (RSE)	2-7
2.4	制御	2-8
2.5	メモリ・サブシステム	2-8
2.5.1	L1 命令キャッシュ	2-9
2.5.2	L1 データ・キャッシュ	2-9
2.5.3	ユニファイド L2 キャッシュ	2-9
2.5.4	ユニファイド L3 キャッシュ	2-9
2.5.5	ALAT (Advanced Load Address Table)	2-9
2.5.6	トランスレーション・ルックアサイド・バッファ (TLB)	2-10
2.5.7	キャッシュ・コヒーレンシ	2-10
2.5.8	ライト・コアレシング	2-10
2.5.9	メモリの順序付け	2-11
2.6	IA-32 実行	2-11
第 3 章	システムバスの概要	3-1
3.1	Itanium® 2 プロセッサのシステムバスの信号伝送	3-1
3.1.1	コモン・クロック・シグナリング	3-1
3.1.2	ソース・シンクロナス・シグナリング	3-2
3.2	信号の概要	3-3
3.2.1	制御信号	3-4
3.2.2	アービトレーション信号	3-4
3.2.3	要求信号	3-5
3.2.4	スヌープ信号	3-5
3.2.5	応答信号	3-6
3.2.6	データ信号	3-7
3.2.7	据え置き信号	3-8
3.2.8	エラー信号	3-8
3.2.9	実行制御信号	3-9
3.2.10	IA-32 互換性信号	3-9

	3.2.11	プラットフォーム信号	3-10
	3.2.12	診断信号	3-10
第 4 章		データの健全性	4-1
	4.1	エラーの分類	4-1
	4.2	Itanium® 2 プロセッサ・システム・バスのエラー検出	4-1
	4.2.1	直接に保護されるバス信号	4-2
	4.2.2	間接的に保護されるバス信号	4-2
	4.2.3	保護されないバス信号	4-3
	4.2.4	Itanium® 2 プロセッサ・システム・バスのエラー・コード・アルゴリズム	4-3
第 5 章		コンフィグレーションと初期化	5-1
	5.1	設定の概要	5-1
	5.2	設定機能	5-1
	5.2.1	データ・バス・エラー・チェック	5-2
	5.2.2	応答 /ID 信号パリティ・エラー・チェック	5-2
	5.2.3	アドレス / 要求信号パリティ・エラー・チェック	5-3
	5.2.4	イニシエータ・バス・エラーでの BERR# のアサート	5-3
	5.2.5	ターゲット・バス・エラーでの BERR# のアサート	5-3
	5.2.6	BERR# サンプリング	5-3
	5.2.7	BINIT# エラーのアサート	5-3
	5.2.8	BINIT# エラーのサンプリング	5-3
	5.2.9	インオーダー・キューのバイプライン化	5-3
	5.2.10	要求バス・パーキング有効	5-3
	5.2.11	対称エージェントのアービトレーション ID	5-3
	5.2.12	クロック周波数比	5-5
	5.3	初期化の概要	5-6
	5.3.1	RESET# による初期化	5-6
	5.3.2	INIT による初期化	5-6
第 6 章		テスト・アクセス・ポート (TAP)	6-1
	6.1	インターフェイス	6-2
	6.2	TAP ロジックへのアクセス	6-2
	6.3	TAP レジスタ	6-4
	6.4	TAP 命令	6-4
	6.5	リセット動作	6-5
第 7 章		統合ツール	7-1
	7.1	インターゲット・プローブ (ITP)	7-1
	7.2	ロジック・アナライザ・インターフェイス (LAI)	7-1
付録 A		信号リファレンス	A-1
	A.1	アルファベット順の信号リファレンス	A-1
	A.1.1	A[49:3]# (I/O)	A-1
	A.1.2	A20M# (I)	A-1
	A.1.3	ADS# (I/O)	A-1
	A.1.4	AP[1:0]# (I/O)	A-1
	A.1.5	ASZ[1:0]# (I/O)	A-1
	A.1.6	ATTR[3:0]# (I/O)	A-2
	A.1.7	BCLKp/BCLKn (I)	A-2
	A.1.8	BE[7:0]# (I/O)	A-3
	A.1.9	BERR# (I/O)	A-3
	A.1.10	BINIT# (I/O)	A-4
	A.1.11	BNR# (I/O)	A-4

A.1.12	BPM[5:0]# (I/O)	A-4
A.1.13	BPRI# (I)	A-4
A.1.14	BR[0]# (I/O) および BR[3:1]# (I)	A-4
A.1.15	BREQ[3:0]# (I/O)	A-5
A.1.16	CCL# (I/O)	A-6
A.1.17	CPUPRES# (O)	A-6
A.1.18	D[127:0]# (I/O)	A-6
A.1.19	D/C# (I/O)	A-6
A.1.20	DBSY# (I/O)	A-6
A.1.21	DBSY_C1# (O)	A-6
A.1.22	DBSY_C2# (O)	A-6
A.1.23	DEFER# (I)	A-7
A.1.24	DEN# (I/O)	A-7
A.1.25	DEP[15:0]# (I/O)	A-7
A.1.26	DHIT# (I)	A-7
A.1.27	DPS# (I/O)	A-8
A.1.28	DRDY# (I/O)	A-8
A.1.29	DRDY_C1# (O)	A-8
A.1.30	DRDY_C2# (O)	A-8
A.1.31	DSZ[1:0]# (I/O)	A-8
A.1.32	EXF[4:0]# (I/O)	A-8
A.1.33	FCL# (I/O)	A-9
A.1.34	FERR# (O)	A-9
A.1.35	GSEQ# (I)	A-9
A.1.36	HIT# (I/O) および HITM# (I/O)	A-9
A.1.37	ID[9:0]# (I)	A-9
A.1.38	IDS# (I)	A-9
A.1.39	IGNNE# (I)	A-9
A.1.40	INIT# (I)	A-9
A.1.41	INT (I)	A-10
A.1.42	IP[1:0]# (I)	A-10
A.1.43	LEN[2:0]# (I/O)	A-10
A.1.44	LINT[1:0] (I)	A-10
A.1.45	LOCK# (I/O)	A-10
A.1.46	NMI (I)	A-11
A.1.47	OWN# (I/O)	A-11
A.1.48	PMI# (I)	A-11
A.1.49	PWRGOOD (I)	A-11
A.1.50	REQ[5:0]# (I/O)	A-11
A.1.51	RESET# (I)	A-12
A.1.52	RP# (I/O)	A-12
A.1.53	RS[2:0]# (I)	A-13
A.1.54	RSP# (I)	A-13
A.1.55	SBSY# (I/O)	A-13
A.1.56	SBSY_C1# (O)	A-13
A.1.57	SBSY_C2# (O)	A-13
A.1.58	SPLCK# (I/O)	A-13
A.1.59	STBn[7:0]# および STBp[7:0]# (I/O)	A-14
A.1.60	TCK (I)	A-14
A.1.61	TDI (I)	A-14
A.1.62	TDO (O)	A-14

A.1.63	THRMTRIP# (O)	A-14
A.1.64	THRMALERT# (O)	A-15
A.1.65	TMS (I)	A-15
A.1.66	TND# (I/O)	A-15
A.1.67	TRDY# (I)	A-15
A.1.68	TRST# (I)	A-15
A.1.69	WSNP# (I/O)	A-15
A.2	信号のまとめ	A-15

図目次

2-1	サポートしている並列処理の2つの例	2-2
2-2	Itanium® 2 プロセッサ・コア・パイプライン	2-3
2-3	Itanium® 2 プロセッサのブロック・ダイアグラム	2-4
2-4	Itanium® 2 プロセッサのFMAC ユニット	2-6
2-5	Itanium® 2 プロセッサのキャッシュ階層	2-8
3-1	コモン・クロック・ラッチ・プロトコル	3-2
3-2	ソース・シンクロナス・ラッチ・プロトコル	3-3
5-1	BR[3:0]# の物理的な相互接続 (4つの対称エージェント)	5-4
5-2	BR[3:0]# の物理的な相互接続 (2つの対称エージェント)	5-5
6-1	テスト・アクセス・ポートのブロック・ダイアグラム	6-1
6-2	TAP コントローラのステート・ダイアグラム	6-2

表目次

3-1	制御信号	3-4
3-2	アービトレーション信号	3-4
3-3	要求信号	3-5
3-4	スヌープ信号	3-5
3-5	応答信号	3-6
3-6	データ信号	3-7
3-7	STBp[7:0]# と STBn[7:0]# に対応する信号	3-7
3-8	据え置き信号	3-8
3-9	エラー信号	3-8
3-10	実行制御信号	3-9
3-11	プラットフォーム信号	3-10
3-12	診断信号	3-10
4-1	直接的なバス信号の保護	4-2
5-1	電源投入時の設定機能	5-2
5-2	Itanium® 2 プロセッサの BREQ[3:0]# バス信号の相互接続 (4ウェイ・プロセッサ)	5-4
5-3	Itanium® 2 プロセッサの BREQ[3:0]# バス信号の相互接続 (2ウェイ・プロセッサ)	5-4
5-4	アービトレーション ID の設定	5-5
5-5	Itanium® 2 プロセッサのシステムバス周波数とプロセッサ・コア 周波数の比の設定	5-5
5-6	Itanium® 2 プロセッサのリセット・ステート (PAL ファームウェアの実行後)	5-6

5-7	Itanium® プロセッサの INIT ステート	5-6
6-1	Itanium® 2 プロセッサの TAP コントローラ用の命令	6-4
A-1	アドレス空間のサイズ	A-2
A-2	有効なメモリ・タイプの信号エンコード	A-2
A-3	バイト・イネーブル上の特殊なトランザクションのエンコード	A-3
A-4	BR0# (I/O)、BR1#、BR2#、BR3# 信号のローテート方式の相互接続 (4P の場合)	A-5
A-5	BR0# (I/O)、BR1#、BR2#、BR3# 信号のローテート方式の相互接続 (2P の場合)	A-5
A-6	BR[3:0]# 信号とエージェント ID	A-5
A-7	DID[9:0]# のエンコード	A-7
A-8	拡張機能信号	A-8
A-9	データ転送のサイズ	A-10
A-10	REQa#/REQb# 信号によって定義されるトランザクションのタイプ	A-12
A-11	STBp[7:0]# および STBn[7:0]# の対応関係	A-14
A-12	出力信号	A-15
A-13	入力信号	A-16
A-14	入力 / 出力信号 (シングル・ドライバ)	A-17
A-15	入力 / 出力信号 (マルチ・ドライバ)	A-18

インテル® Itanium® 2 プロセッサは、Itanium アーキテクチャに基づくプロセッサ製品ファミリの第 2 世代であり、高性能サーバ/ワークステーションのニーズに応えるように設計されている。

Itanium アーキテクチャは、EPIC (Explicitly Parallel Instruction Computing: 明示的並列命令コンピュティング) の採用により、RISC および CISC 技術を超えるアーキテクチャである。EPIC は、プロセッサに対して明示的な並列実行を可能にするインテリジェントなコンパイラにより、広範囲にわたる処理リソースをペアにして利用する技術である。Itanium 2 プロセッサの大きな内部リソースと分岐予測およびスペキュレーションを組み合わせ、Microsoft Windows*、HP-UX*、Linux* の各バージョンなどのさまざまなオペレーティング・システム上で実行される高性能アプリケーション向けの最適化が可能になる。Itanium 2 プロセッサは、数千個のプロセッサを使用するシステムなど、非常に大規模なシステムもサポートしており、きわめて負荷の大きいエンタープライズ・アプリケーションやテクニカル・コンピューティング・アプリケーションに対応する、余裕のある処理能力とパフォーマンスを提供する。Itanium 2 プロセッサは、SMBus の幅広い互換性と、総合的な信頼性、可用性、保守性 (RAS) 機能により、最大限の稼働時間を必要とするアプリケーションに理想的である。Itanium 2 プロセッサは、高性能サーバ/ワークステーション向けに、現在のアプリケーションに対応するすぐれたパフォーマンスや信頼性と、将来の成長する e-Business のニーズに対応するスケーラビリティを提供する。

1.1 Itanium® 2 プロセッサのシステムバス

Itanium 2 プロセッサのほとんどの信号は、Itanium プロセッサの AGTL+ (Assisted Gunning Transceiver Logic) シグナリング・テクノロジーを使用する。終端電圧 (V_{CTERM}) がベースボード上で発生し、これがシステムバスのハイの基準電圧になる。Itanium 2 プロセッサのシステムバス信号のほとんどをドライブするバッファは、ローからハイへの移行時に、 V_{CTERM} にアクティブにドライブされる。これにより、立ち上がり時間が短縮され、ノイズが軽減される。ただし、これらの信号は、オープンドレインと見なす必要があり、終端部分が V_{CTERM} になる必要がある (これがハイ・レベルを提供する)。オンダイ・ターミネーションが有効になると、Itanium 2 ベースのシステムバスの両端にあるバス・エージェント内のアクティブ・ターミネーションにより、システムバスの終端部分は V_{CTERM} になる。また、オフダイ・ターミネーションもサポートされている。この場合は、 V_{CTERM} に接続された外部抵抗によってターミネーションが提供される。

AGTL+ 入力は、基準信号 (V_{REF}) を必要とする差分レシーバを使用する。このレシーバは、 V_{REF} を使用して、信号が論理 0 か論理 1 かを判定する。Itanium 2 プロセッサは、オンダイで V_{REF} を発生するため、オフチップの基準電圧ソースは不要である。

1.2 PAL (Processor Abstraction Layer)

Itanium 2 プロセッサは、プロセッサ・モデル固有の PAL (Processor Abstraction Layer) ファームウェアを必要とする。PAL ファームウェアは、プロセッサの初期化、エラー回復などの機能をサポートする。PAL ファームウェアは、各種のプロセッサ・ハードウェア上で、システム・ファームウェアとオペレーティング・システムに対する整合性のあるインターフェイスを提供する。PAL については、『インテル® Itanium® アーキテクチャ・ソフトウェア・デベロッパーズ・マニュアル、第 2 巻: システム・アーキテクチャ』を参照のこと。プラットフォームは、Itanium 2 プロセッサを初期化できるように、リセット時にファームウェアのアドレス空間と PAL へのアクセス機能を提供しなければならない。

SAL (System Abstraction Layer) ファームウェアには、プラットフォームの初期化、オペレーティング・システムへのブート、ランタイム機能を提供する、プラットフォーム固有のファームウェアが含まれている。SAL の詳細は、『Itanium Processor Family System Abstraction Layer Specification』を参照のこと。

1.3 用語

本書では、信号名の後の「#」記号は、アクティブ・ロー信号を示す。アクティブ・ロー信号とは、信号がロー・レベルにドライブされたとき、(信号名に応じた)アクティブな状態になるという意味である。例えば、RESET# がローのときは、プロセッサのリセットが要求されている。NMI がハイのときは、マスク不可割り込みが発生している。信号名がアクティブな状態を意味するのではなく、バイナリ・シーケンスの一部(アドレスやデータなど)を記述している場合は、「#」記号は、信号が反転されていることを示す。例えば、D[3:0] = 'HLHL' は 16 進数の 'A' を示し、D [3:0] # = 'LHLH' も 16 進数の 'A' を示す (H = ハイの論理レベル、L = ローの論理レベル)。

多くの場合、信号は信号名と同じ名前の物理ピンに 1 対 1 で対応しているが、異なる信号が 1 つのピンに対応付けられることもある。例えば、アドレス・ピン A[49:3]# の場合がこれに当たる。第 1 クロックでは、これらのアドレス・ピンがアサートされ、有効なアドレスを示す。第 1 クロックは、小文字の a またはピン名それ自体によって示される (Aa[49:3]# または A[49:3]#)。第 2 クロックでは、これらのアドレス・ピン上で他の情報がアサートされる。これらの信号は、機能名 (DID[9:0]# など) で参照されるか、ピン名に小文字の b を付けて参照される (Ab[25:16]# など)。また、いくつかのピンは、RESET# のアサートからデアサートへのエッジで設定機能を持つ。

「システムバス」の用語は、プロセッサ、システム・コア・ロジック、他のバス・エージェントの間のインターフェイスを示す。システムバスは、プロセッサ、メモリ、I/O に対するマルチプロセッシング・インターフェイスである。

信号名はすべて大文字で示す。例: VCTERM

電圧レベル、電流レベル、または時間値を示す記号は、普通の下付文字 (例えば、V_{CC,core}) または大文字の省略形の下付文字 (例えば、T_{CO}) で示す。

1.4 参考資料

本仕様書の読者は、以下の資料に記載されている内容と概念をよく理解している必要がある。

タイトル	資料番号
Intel® Itanium® 2 Processor at 1 GHz and 900 MHz Datasheet	250945
Intel® Itanium® 2 Processor Specification Update	251141
インテル® Itanium® アーキテクチャ・ソフトウェア・デベロッパーズ・マニュアル <ul style="list-style-type: none"> • 第 1 巻: アプリケーション・アーキテクチャ • 第 2 巻: システム・アーキテクチャ • 第 3 巻: 命令セット・リファレンス 	245317J 245318J 245319J
Intel® Itanium® 2 Processor BSDL Model	
インテル® Itanium® 2 プロセッサ・リファレンス・マニュアル: ソフトウェアの開発と最適化	251110J
Intel® Itanium® Processor Family System Abstraction Layer Specification	245359
Intel® Itanium® Processor Family Error Handling Guide	249278
ITP700 Debug Port Design Guide	249679
システム・マネージメント・バスの仕様	http://www.smbus.org/specs

参考資料の最新の改訂版は、インテルの御社担当に問い合わせるか、<http://developer.intel.com> を参照のこと。

1.4.1 改訂履歴

バージョン 番号	説明	改訂時期
001	初版	2002年7月



Itanium® 2 プロセッサの マイクロアーキテクチャ

2

本章では、Itanium 2 プロセッサのマイクロアーキテクチャの概要について説明する。Itanium アーキテクチャの詳細は、『インテル® Itanium® アーキテクチャ・ソフトウェア・デベロッパーズ・マニュアル』を参照のこと。

2.1 概要

Itanium 2 プロセッサは、Itanium 命令セット・アーキテクチャ (ISA) の 2 番目の製品である。Itanium 2 プロセッサは、ハードウェアとソフトウェアを緊密に結合する EPIC 設計構想を採用している。この設計手法では、ソフトウェアがすべての使用可能なコンパイル時の情報を利用でき、この情報をハードウェアに効率的に伝達できるように、ハードウェアとソフトウェアの間のインターフェイスが設計される。EPIC 設計は、メモリ・レイテンシ、メモリ・アドレスの明確化、制御フローの依存関係など、現代のコンピュータの基本的なパフォーマンス・ボトルネックに対処するものである。EPIC 構造は、強力なアーキテクチャ・セマンティクスを提供し、ソフトウェアが大きなスケジューリング範囲にわたってグローバルな最適化を行い、ハードウェアが命令レベルの並列性 (ILP) を利用できるようにする。ハードウェアは、この拡張された ILP を利用できるように、大量の実行リソースを備えている。また、EPIC では、動的なランタイム最適化が重視され、コンパイル済みコードを高いスループットで実行するようにスケジューリングできる。この手法により、ハードウェアとソフトウェアの相乗効果が高まり、パフォーマンスが全体として向上する。

Itanium 2 プロセッサは、1GHz または 900MHz で動作する 6 つの命令を同時に処理できる 8 段のパイプラインを持っている。これにより、ILP を活かせる大量のリソースと高いクロック周波数を組み合わせ、各命令のレイテンシが最小限に抑えられる。実行リソースは、6 個の整数ユニット、6 個のマルチメディア・ユニット、2 個のロード・ユニットと 2 個のストア・ユニット、3 個の分岐ユニット、2 個の拡張精度浮動小数点ユニット、2 個の追加の単精度浮動小数点ユニットで構成される。ハードウェアは、動的プリフェッチ機能、分岐予測機構、レジスタ・スコアボード、ノンブロッキング・キャッシュを使用する。3 レベルのオンダイ・キャッシュにより、メモリ・レイテンシが全体として最小限に抑えられる。キャッシュには、コア・スピードでアクセスでき、32GB/ サイクル以上のデータ帯域幅を実現する。3MB または 1.5MB の L3 キャッシュが含まれる。システムバスは、グルーレス MP をサポートし、システムバスにつき最大 4 個のプロセッサに対応する。したがって、非常に大規模なシステムの効果的なビルディング・ブロックとして使用できる。バランスのとれたコアとメモリ・サブシステムにより、電子商取引ワークロードから高性能テクニカル・コンピューティングまで、広範囲にわたるアプリケーションに対応する高性能が得られる。

2.1.1 6 命令を同時に処理できる EPIC コア

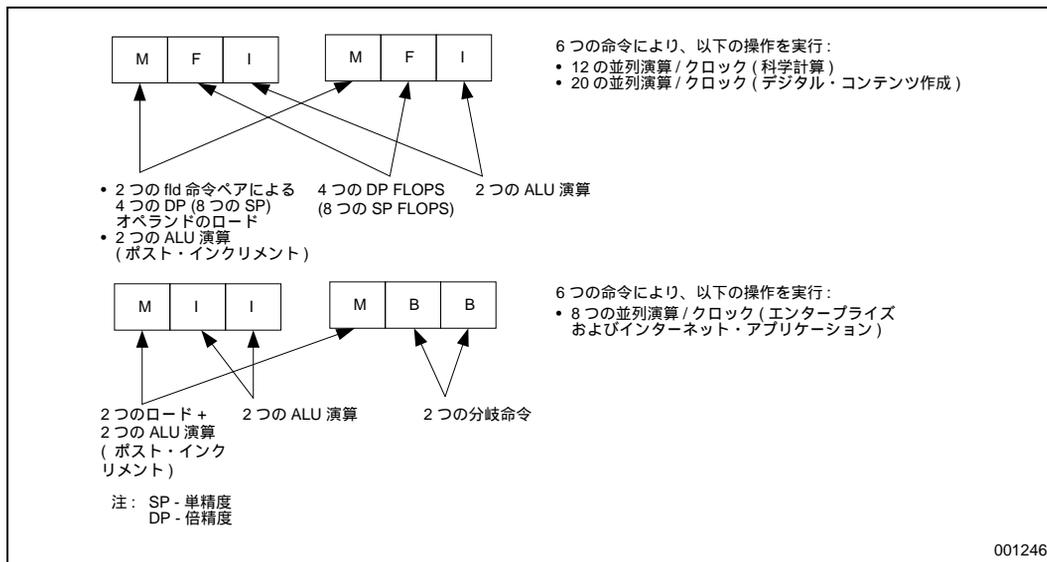
Itanium 2 プロセッサは、EPIC 設計に基づいて、6 つの命令を同時に処理できる 8 段のパイプラインを持つ。このパイプラインは、6 個の整数 ALU、6 個のマルチメディア ALU、2 個の拡張精度浮動小数点ユニット、2 個の追加の単精度浮動小数点ユニット、2 個のロード・ユニットと 2 個のストア・ユニット、および 3 個の分岐ユニットで構成される実行ユニットを使用する。マシンは、1 クロックにつき 6 つの命令 (すなわち、2 つの命令バンドル) のフェッチ、発行、実行、リタイアが行える。

命令バンドルには、コンパイラによって割り当てられる、3 つの命令とテンプレート・インジケータが入る。命令バンドル内の各命令は、命令のタイプに従って、いずれかの実行パイプラインにディスパースされる。命令のタイプは、ALU 整数 (A)、非 ALU 整数 (I)、メモリ (M)、浮動小数点 (F)、分岐 (B)、または拡張 (L) である。Itanium 2 プロセッサは、Itanium プロセッサと比べて実行ユニットの数が増えたため、コンパイラのディスパース・オプションの数は 3 倍以上に増え

ている。命令とバンドルの詳細は、『インテル® Itanium® アーキテクチャ・ソフトウェア・デベロッパーズ・マニュアル』を参照のこと。Itanium 2 プロセッサの命令ディスパースルの詳細については、『インテル® Itanium® 2 プロセッサ・リファレンス・マニュアル：ソフトウェアの開発と最適化』を参照のこと。

図 2-1 は、各種のワークロードでサポートしている並列操作レベルの 2 つの例を示している。エンタープライズ・コードや電子商取引コードでは、バンドル・ペア内の MII/MBB テンプレートの組み合わせにより、1 クロックにつき 6 つの命令、すなわち 8 つの並列操作 (2 つのロード/ストア、2 つの汎用 ALU 演算、2 つのポストインクリメント ALU 演算、2 つの分岐命令) を実行できる。あるいは、MIB/MIB のペアによって、2 つの分岐操作の代わりに 1 つの分岐ヒントと 1 つの分岐操作を使用して、同じ演算の組み合わせを実行できる。科学計算用コードでは、各バンドル内で MFI テンプレートを使用すると、1 クロックにつき 12 の並列操作 (レジスタへの 4 つの倍精度オペランドのロード、4 つの倍精度浮動小数点演算、2 つの整数 ALU 演算、2 つのポストインクリメント ALU 演算) を実行できる。単精度浮動小数点データを使用するデジタル・コンテンツ作成コードでは、マシンの SIMD 機能により、実質的に 1 クロックにつき最大 20 の並列操作 (8 つの単精度オペランドのロード、8 つの単精度浮動小数点演算、2 つの整数 ALU 演算、2 つのポストインクリメント ALU 演算) を実行できる。

図 2-1. サポートしている並列処理の 2 つの例

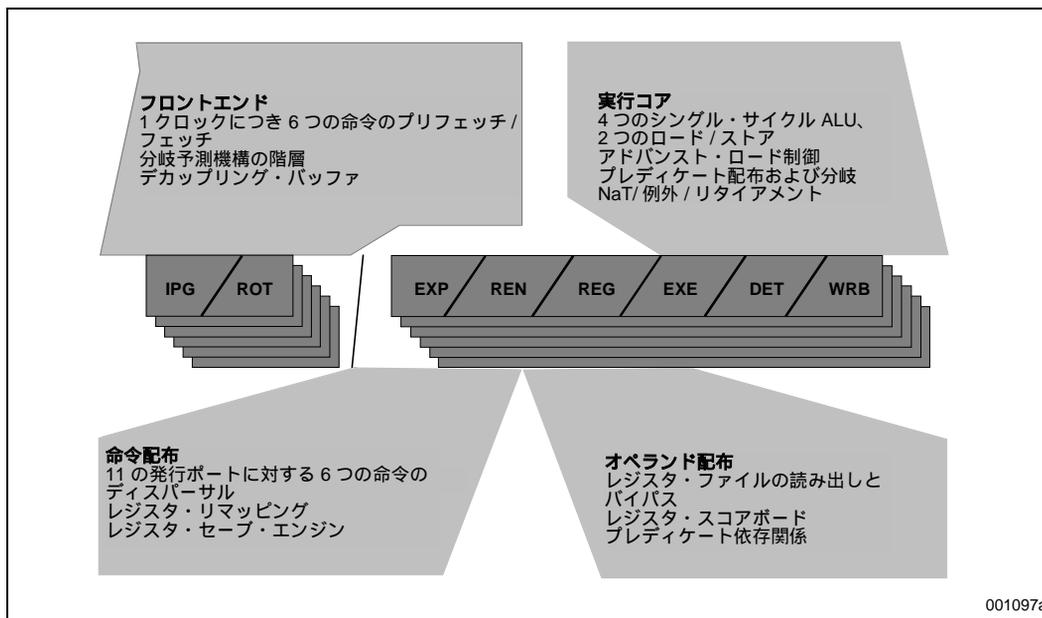


2.1.2 プロセッサ・パイプライン

Itanium 2 プロセッサのハードウェアは、図 2-2 に示すように、1 クロックにつき最大 6 つの命令を並行して実行できる 8 段のコア・パイプラインで構成されている。パイプラインの最初の 2 つのステージは、命令フェッチを実行し、フェッチした命令を命令ローテーション (ROT) ステージのデカップリング・バッファに入れる。デカップリング・バッファにより、マシンのフロントエンドはバックエンドから独立して動作できる。図中のコア・パイプラインを切り離している太い線が、デカップリング・ポイントを示している。次の 2 つのステージ (拡張 (EXP) とレジスタ・リネーム (REN)) は、ディスパースルとレジスタ・リネームを実行する。レジスタ読み出し (REG) ステージは、オペランドの配布を実行する。REG ステージは、レジスタ・ファイルにアクセスし、プレディケート制御の処理後にバイパス・ネットワークを介してデータを配布する。最後の 3 つのステージは、ワイドな並列実行を行い、例外の管理とリタイアメントを実行する。特に、例外検出 (DET) ステージは、分岐の解決と、メモリ例外の管理やスペキュレーションをサポートする。

Itanium 2 プロセッサ・パイプラインの詳細は、『インテル® Itanium® 2 プロセッサ・リファレンス・マニュアル：ソフトウェアの開発と最適化』を参照のこと。

図 2-2. Itanium® 2 プロセッサ・コア・パイプライン

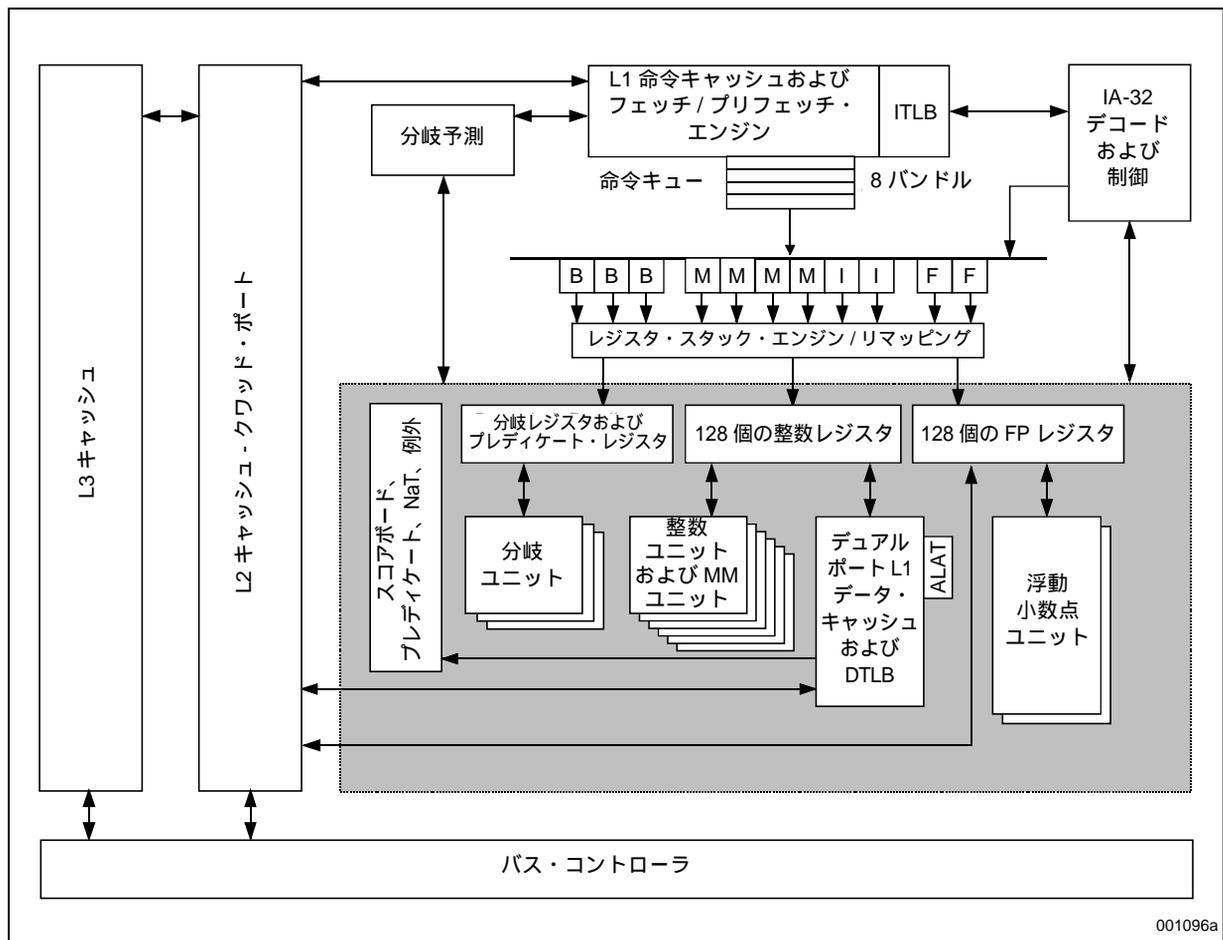


2.1.3 プロセッサのブロック・ダイアグラム

図 2-3 は、Itanium 2 プロセッサのブロック・ダイアグラムを示している。Itanium 2 プロセッサの機能は、以下の5つのグループに分けられる。次の各項では、各グループの動作について詳しく説明する。

1. 命令処理
命令処理ブロックは、命令プリフェッチ、命令フェッチ、L1 命令キャッシュ、分岐予測、命令アドレス生成、命令バッファ、命令発行、ディスペーサル、リネーム用のロジックで構成される。
2. 実行
実行ブロックは、マルチメディア・ロジック、整数 ALU 実行ロジック、浮動小数点 (FP) 実行ロジック、整数レジスタ・ファイル、L1 データ・キャッシュ、FP レジスタ・ファイルで構成される。
3. 制御
制御ブロックは、例外ハンドラ、パイプライン制御部、レジスタ・スタック・エンジン (RSE) で構成される。
4. メモリ・サブシステム
メモリ・サブシステムは、ユニファイド L2 キャッシュ、オンチップ L3 キャッシュ、PIC (Programmable Interrupt Controller)、命令およびデータ・トランスレーション・ルックアサイド・バッファ (TLB)、ALAT (Advanced Load Address Table) および外部システム・バス・インターフェイス・ロジックで構成される。
5. IA-32 互換実行エンジン
IA-32 互換実行エンジンは、IA-32 アプリケーション用命令の実行のためのフェッチ、デコード、スケジューリングを行う。

図 2-3. Itanium® 2 プロセッサのブロック・ダイアグラム



2.2 命令処理

2.2.1 命令プリフェッチと命令フェッチ

Itanium 2 プロセッサは、命令をパイプライン・キャッシュからデカップリング・バッファに投機的にプリフェッチする。Itanium 2 プロセッサは、高度な分岐予測手法とコンパイラ・ヒントを使用して、スペキュレーティブ・プリフェッチを行う。Itanium 2 プロセッサの命令シーケンス部が、命令のフェッチと実行ユニットに対するディスパースルを行う。命令アドレス生成ユニットは、次の命令ポインタ (IP) を選択する。命令ポインタは、次のシーケンシャル・アドレス、静的および動的な分岐予測アドレス、互換性ロジックが伝達する命令アドレス、予測ミスになった分岐を修正する検証済みのターゲット・アドレス、または例外ハンドラのアドレスの中から選択される。

Itanium 2 プロセッサは、L1 命令キャッシュ (L1I) から 2 つの命令バンドル (1 バンドルにつき 3 つの命令) を読み出し、命令バッファに入れる。命令バッファは、実行ユニットによって処理されるのを待っている命令バンドルを格納する。命令キャッシュ・ミスによって発生する分岐予測バブルの影響を軽減するために、命令バッファから読み出されたバンドルは、実行リソースが使用可能かどうかに基づいて、命令発行/リネーム・ロジックに送られる。

2.2.2 分岐予測

分岐予測ロジックは、高度な予測方式を使用して、分岐の有無と命令キャッシュからの各分岐による読み出しターゲットを予測する。Itanium 2 プロセッサは、0 バブル分岐予測アルゴリズムとバックアップ分岐予測テーブルを搭載している。分岐が発生すると、分岐ターゲットは、命令インタ生成ロジックにリストアされる。

命令プリフェッチ・ロジックは、L1I キャッシュと L2 キャッシュの間のインターフェイスとして機能する。このロジックは、L1I ミスを防ぐために、L2 内の命令が必要になる前に L2 から命令をプリフェッチする。プリフェッチ処理は、コンパイラの制御の下で実行される。それでも L1 命令キャッシュ・ミスが発生した場合は、命令プリフェッチ・ロジックは命令アドレス生成ロジックをストールさせ、L2 キャッシュから情報を検索する。必要な命令が L2 キャッシュ内にない場合は、さらに L3 キャッシュをチェックする。

2.2.3 ディスパーサル・ロジック

Itanium 命令には 12 種類のテンプレートがある。テンプレートには、それ以降の命令の並列発行を中止するようにハードウェアに指示する、明示的ストップ・ビットが含まれている。1 バンドルにつき 3 つの命令があり、ハードウェアは 1 クロックにつき 2 バンドル (すなわち、6 つの命令) を処理できる。ディスパーサル・ロジックは、発行ポートを介して、フルにパイプライン化された機能ユニットのうち 1 つに各命令を送り込む。

命令バッファは、最大 8 つの命令バンドルを保持する。命令バッファは、各サイクルで 2 つのバンドルをディスパーサル・ロジックに渡せる。一般的に、命令は、サポートしている実行ポートのうち最初に使用可能なポートに送られる。

2.3 実行

Itanium 2 プロセッサの実行ロジックは、6 個のマルチメディア・ユニット、6 個の整数ユニット、2 個の浮動小数点ユニット、3 個の分岐ユニット、4 個のロード / ストア・ユニットで構成される。Itanium 2 プロセッサは、汎用レジスタと FP レジスタを使用して、進行中の実行を管理する。整数ロードは L1 データ・キャッシュによって処理されるが、整数ストアは L2 キャッシュによって処理される。FP ロードと FP ストアも、L2 キャッシュによって処理される。L1 内でルックアップが行われるたびに、スペキュレーティブな要求が L2 キャッシュに送られる。

マルチメディア・エンジンは、64 ビット・データを、 2×32 ビット、 4×16 ビット、または 8×8 ビットのバックド・データ・タイプとして扱う。バックド・データ・タイプ、すなわち SIMD (Single Instruction Multiple Data) データ・タイプには、算術演算、シフト演算、データ整列演算の 3 つのクラスの算術演算を実行できる。一方、整数エンジンは、最大 6 つの非バックド整数算術演算および論理演算をサポートしている。各サイクルで、最大 6 つの整数演算またはマルチメディア演算を実行できる。

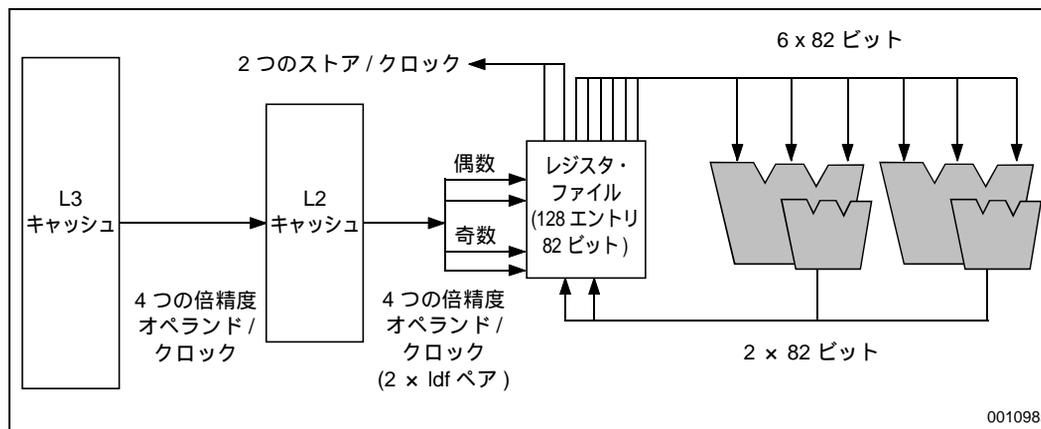
2.3.1 浮動小数点ユニット (FPU)

Itanium 2 プロセッサは、大きな浮動小数点実行帯域幅を持つ。Itanium 2 プロセッサの FPU は、4 段のパイプラインを持つ。バイパス・ロジックが追加されたので、各種の FP ステージから FP ライトバック・ステージへの迅速なデータ転送が行える。FP ロジックには、FMAC (FP Multiply Accumulate) ハードウェア・ユニット、高速丸めロジック、SIMD 形式のサポートも含まれている。Itanium 2 プロセッサは、各クロックサイクルで、最大 2 つの FP 命令 (または 2 つの整数乗算命令) と、2 つの FP ロードや 2 つの FP ストア (または 4 つの FP ロード) 命令を発行できる。

命令を FP パイプラインに入れる前に、数値例外がないかどうか、数値オペランドがチェックされる。チェックの結果は、パイプラインの終わりで書き戻される。

FPU は、82 ビット値を処理する 2 つの FMAC をサポートしている。FMAC は、単精度、倍精度、拡張倍精度の浮動小数点演算を実行できる。FPU は、8 つの読み出しポートと少なくとも 6 つの書き込みポートを持つ 128 エントリの FP レジスタ・ファイルを備えている。FP レジスタは、各クロックで、メモリからの 4 つの倍精度オペランドのロード、FMAC からの 2 つの 82 ビット値のライトバック、(2 つの並列拡張精度 FMAC のための) 2 つのストア操作をサポートしている。FMAC ユニットの構造は、[図 2-4](#) を参照のこと。

図 2-4. Itanium® 2 プロセッサの FMAC ユニット



2.3.2 整数ロジック

6 個の整数実行ユニットは、64 ビットの算術命令、論理命令、シフト命令、ビットフィールド操作命令を実行する。さらに、整数実行ユニットは、32 ビット・ポインタの処理を高速化する命令を実行できる。その他の操作には、IA-32 互換エンジンのためのプレディケート計算、リニア・アドレス計算、フラグ生成が含まれる。

整数ロジックは、6 個の汎用 ALU と 2 個のロード・ポートおよび 2 個のストア・ポートを持つ。これらの ALU は、フル・バイパス機能を持つ。

2.3.3 レジスタ・ファイル

Itanium 2 プロセッサは、Itanium アーキテクチャが提供する大量のレジスタ・リソースを実装している。レジスタの数が多いため、メモリとの間で読み出しや書き込みを行うことなく、多くの操作を実行できる。主な実行レジスタには、128 個の汎用レジスタ、128 個の浮動小数点レジスタ、64 個のプレディケート・レジスタ、8 個の分岐レジスタが含まれる。

2.3.3.1 汎用レジスタ

128 個の (64 ビット) 汎用レジスタは、すべての整数計算および整数マルチメディア計算の中心となるリソースである。汎用レジスタには、GR0 ~ GR127 の番号が付いており、すべてのプログラムがすべての特権レベルで使用できる。

汎用レジスタは、2 つのサブセットに分けられる。汎用レジスタ 0 ~ 31 は、静的汎用レジスタと呼ばれる。その中で、GR0 は特殊なレジスタであり、ソース・オペランドとしては常に 0 として読み出され、GR0 に書き込もうとすると無効操作フォルトが発生する。汎用レジスタ 32 ~ 127 は、スタック汎用レジスタと呼ばれる。スタック・レジスタは、設定可能な数のローカル・レジスタおよび出力レジスタから構成されるレジスタ・スタック・フレームの割り当てにより、プログラムで使用可能になる。

2.3.3.2 浮動小数点レジスタ

128 個の (82 ビット) 浮動小数点レジスタは、すべての浮動小数点計算に使用される。これらのレジスタには、FR0 ~ FR127 の番号が付いており、すべてのプログラムがすべての特権レベルで使用できる。浮動小数点レジスタは、2 つのサブセットに分けられる。浮動小数点レジスタ 0 ~ 31 は、静的浮動小数点レジスタと呼ばれる。その中で、FR0 と FR1 は特殊なレジスタである。ソース・オペランドとして使用すると、FR0 は常に +0.0 として読み出され、FR1 は常に +1.0 として読み出される。FR0 または FR1 をデスティネーションとして使用すると、フォルトが発生する。

浮動小数点レジスタ 32 ~ 127 は、ローテート浮動小数点レジスタと呼ばれる。これらのレジスタをプログラムによってリネームすれば、ループを高速化できる。

2.3.3.3 プレディケート・レジスタ

64 個の (1 ビット) プレディケート・レジスタは、比較命令の結果を保持する。プレディケート・レジスタには、PR0 ~ PR63 の番号が付いており、すべてのプログラムがすべての特権レベルで使用できる。これらのレジスタは、命令の条件付き実行に使用される。

プレディケート・レジスタは、2 つのサブセットに分けられる。プレディケート・レジスタ 0 ~ 15 は、静的プレディケート・レジスタと呼ばれる。その中で、PR0 はソース・オペランドとしては常に '1' として読み出され、デスティネーションとして使用すると、結果は廃棄される。静的プレディケート・レジスタは、条件分岐にも使用される。

プレディケート・レジスタ 16 ~ 63 は、ローテート・プレディケート・レジスタと呼ばれる。これらのローテート・レジスタを使用して、ソフトウェア・パイプライン・ループを効率的に処理できる。

2.3.3.4 分岐レジスタ

8 個の (64 ビット) 分岐レジスタは、分岐情報を保持する。分岐レジスタには、BR0 ~ BR7 の番号が付いており、すべてのプログラムがすべての特権レベルで使用できる。分岐レジスタを使用して、間接分岐の分岐ターゲット・アドレスを指定できる。

2.3.4 レジスタ・スタック・エンジン (RSE)

Itanium 命令セット・アーキテクチャ (ISA) は、大きなレジスタ・ファイルと、間接参照ベースのレジスタ・アクセス機構により、プロシージャ・インターフェイスでのスピルとフィルを回避する。間接参照機構により、レジスタ・フレームをスタックでき、レジスタ・ファイルを使用してプロシージャ間変数を共有できる。

プロシージャが呼び出されると、呼び出し元プロシージャのレジスタを明示的に保存しなくても、呼び出し先プロシージャは、レジスタの新しいフレームを使用できる。古いレジスタは、物理的容量が十分にある限り、大きなオンチップ物理レジスタ・ファイル内にとどまる。必要なレジスタの数が使用可能な物理的容量を超えると、レジスタ・スタック・エンジン (RSE) と呼ばれるステート・マシンがレジスタをメモリに保存し、次の呼び出しに必要なレジスタを解放する。RSE により、あたかもレジスタの数が無制限にあるかのように見える。

呼び出しのリターン時に、ベース・レジスタは、呼び出しの前に呼び出し元プロシージャがレジスタへのアクセスに使用していた値に復元される。多くの場合、これらのレジスタの保存操作が必要になる前にリターンが検出されるため、レジスタの復元は不要である。RSE が呼び出し先プロシージャのレジスタの一部を保存していた場合は、RSE が必要な数の呼び出し先のレジスタを復元できるまで、リターン時にプロセッサはストールする。Itanium 2 プロセッサは、RSE の強制レイジー・モードをサポートしている。これについては、『インテル® Itanium® 2 プロセッサ・リファレンス・マニュアル: ソフトウェアの開発と最適化』を参照のこと。

『インテル® Itanium® アーキテクチャ・ソフトウェア・デベロッパーズ・マニュアル』は、RSE について詳しく説明している。

2.4 制御

Itanium 2 プロセッサの制御部は、例外ハンドラとパイプライン制御部で構成される。例外ハンドラは、例外を優先度に応じて処理する。パイプライン制御部は、レジスタ・ソースの依存関係を検出するスコアボードと、データ・スペキュレーションをサポートするキャッシュを持つ。マシンは、ソース・オペランドがまだ使用できない場合にのみストールする。パイプライン制御部は、プレディケート・レジスタを介してプレディケーションをサポートする。

パイプライン制御部には、パフォーマンス監視ユニットも含まれている。パフォーマンス監視ユニットが収集したデータをダンプして、Itanium 2 プロセッサのパフォーマンスを分析できる。

2.5 メモリ・サブシステム

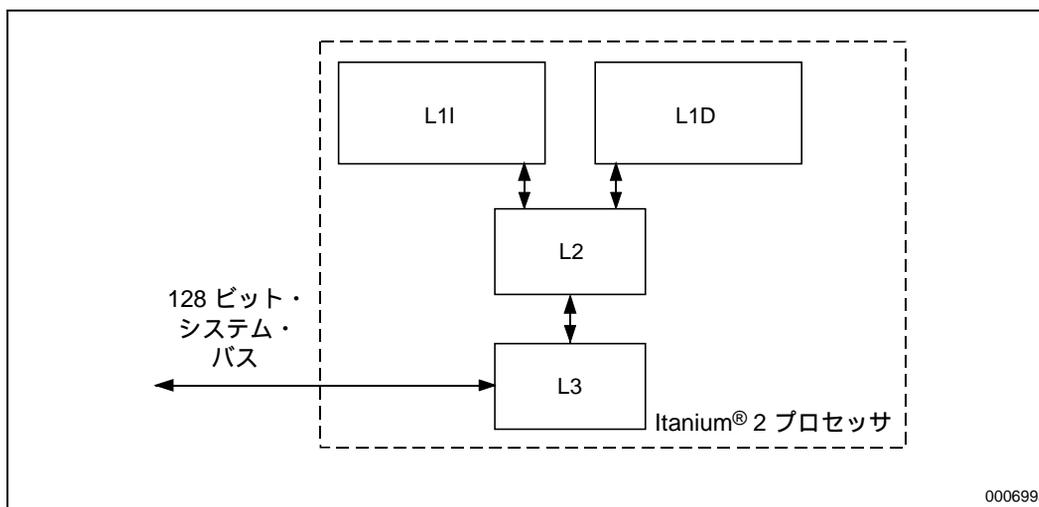
メイン・システム・メモリは、128 ビット・システム・バス (図 2-5 を参照) を介してアクセスされる。システムバスは、Itanium プロセッサのシステムバスと同じように、トランザクション型のパイプライン・バスである。Itanium 2 プロセッサのメモリ・サブシステムは、システム・バス・インターフェイス・ロジック、L1D キャッシュ、L2 キャッシュ、L3 キャッシュ、割り込みコントローラ・ユニット、ALAT、および TLB で構成される。

Itanium 2 プロセッサは、アライメントの合っていないすべての IA-32 メモリ・アクセスをサポートしている。Itanium アーキテクチャ内で 8 バイト境界を超えるメモリ参照が行われると、unaligned フォルトが発生する。アライメントの合っていないアクセスによるパフォーマンスの低下と、データ・メモリ unaligned フォルト・ハンドラのオーバーヘッドを回避するには、できるだけアライメントの合ったメモリ・オペランドを使用する。

L1、L2、L3 は、ノンブロッキング・キャッシュである。データ用の L1 キャッシュと命令用の L1 キャッシュは別個に存在する。L1 データ・キャッシュはクワッド・ポートである。L2 キャッシュはユニファイド・キャッシュであり、命令とデータの両方を格納する。L2 キャッシュはクワッド・ポートであり、Itanium 2 プロセッサのフル・クロック・スピードでアクセスできる。L2 キャッシュ内の命令へのアクセスにはすべてのポートが使用されるが、データ要求の場合は、4 ポートのうち 1 つ、2 つ、3 つ、またはすべてのポートを使用できる。L2 キャッシュに対する要求がミスになると、その要求は迅速に L3 キャッシュに転送される。

統合型外部割り込みコントローラは、外部バス・ロジックによってシステムバスとのインターフェイスをとり、メモリ・マップされた位置を介して、システムバスからの外部割り込みと内部割り込みを受信する。

図 2-5. Itanium® 2 プロセッサのキャッシュ階層



2.5.1 L1 命令キャッシュ

Itanium 2 プロセッサの L1 命令 (L1I) キャッシュのサイズは 16KB である。L1I は、シングルサイクル、ノンブロッキング、デュアル・ポートの 4 ウェイ・セット・アソシアティブ・キャッシュ・メモリ (ライン・サイズ 64 バイト) である (ウェイ予測機能はない)。タグ配列はデュアル・ポートである。1 つのポートは命令フェッチ用で、もう 1 つのポートは、プリフェッチ、スヌープ、フィル、カラム無効化の間で共有される。データ配列もデュアル・ポートであり、読み出し (フェッチ) とフィルが同時に行える。L1I はフルにパイプライン化され、各サイクルで 2 つの命令バンドル (6 つの命令) を配布できる。

L1I キャッシュは、物理的にインデックスとタグが付けられる。

2.5.2 L1 データ・キャッシュ

L1 データ・キャッシュは、4 ポート (2 つのロード・ポートと 2 つのストア・ポート)、サイズ 16KB のノンブロッキング・キャッシュである。L1D は、ウェイ予測機能なしの 4 ウェイ・セット・アソシアティブ・キャッシュ (ライン・サイズ 64 バイト) として構成される。このキャッシュは、2 つのロードと 2 つのストアが同時に行える。L1 データ・キャッシュは、整数データのみを格納する (浮動小数点ロード・データやセマフォ・ロード・データは格納しない)。L1D キャッシュは、ライト・アロケートなしのライトスルー・ポリシーを使用する。L1D キャッシュは、ロードとストアのために物理的にインデックスとタグが付けられる。

2.5.3 ユニファイド L2 キャッシュ

ユニファイド L2 キャッシュ・メモリは 4 ポートを持ち、バンキングにより最大 4 つの同時アクセスをサポートする。L2 キャッシュは、256 KB、8 ウェイ・セット・アソシアティブ (ライン・サイズ 128 バイト) で 16 バイト・バンクを持つ、ノンブロッキング、アウトオブオーダー・キャッシュである。L2 キャッシュの読み出し帯域幅は、1 秒につき 64GB である。L2 キャッシュは、ライト・アロケートありのライトバック・ポリシーを使用する。L2 キャッシュは、物理的にインデックスとタグが付けられる。

L2 は、すべての L1I キャッシュ・ミスと L1D キャッシュ・ミスの処理以外に、すべての浮動小数点メモリ・アクセス (1 クロックにつき最大 4 つの同時浮動小数点ロード) を処理する。Itanium 2 プロセッサのセマフォ命令も、L2 がすべて処理する。

2.5.4 ユニファイド L3 キャッシュ

Itanium 2 プロセッサのオンチップ L3 キャッシュのサイズは、1.5MB または 3MB である。L3 キャッシュは、物理的にインデックスとタグが付けられる。L3 キャッシュは、シングル・ポート、12 ウェイ・セット・アソシアティブ (ライン・サイズ 128 バイト) のフルにパイプライン化されたノンブロッキング・キャッシュである。このキャッシュは、8 つの実行待ちの要求 (そのうち 7 つはロード/ストア、1 つはフィル) を蓄えられる。L3 からコア/L1I/L1D または L2 への最大転送速度は、32GB/サイクルである。L3 は、シングル・ビット補正およびダブル・ビット検出 ECC により、タグとデータの両方を保護する。

2.5.5 ALAT (Advanced Load Address Table)

ALAT (Advanced Load Address Table) と呼ばれるキャッシュ構造により、Itanium 2 プロセッサのデータ・スペキュレーションが有効になる。ALAT は、マシンが発行するスペキュレーティブなデータ・ロードの情報とこれらのロードで別名参照されるストアの情報を保持する。このキャッシュ構造は、32 エントリのフル・アソシアティブ配列であり、1 サイクルにつき 2 つのロードと 2 つのストアを処理できる。ALAT は、アドバンスド・ロード「チェック」操作のための別名参照情報を提供する。

2.5.6 トランスレーション・ルックアサイド・バッファ (TLB)

Itanium 2 プロセッサは、DTLB (Data Translation Lookaside Buffer) と ITLB (Instruction Translation Lookaside Buffer) の 2 種類の TLB を持つ。Itanium 2 プロセッサの DTLB は、L1 DTLB と L2 DTLB の 2 つのレベルに分けられている。L1 DTLB ヒットと L2 DTLB ヒットに依存するのは、L1D キャッシュ・ロードだけである。ストアと L2/L3 キャッシュ・ヒットは、L2 DTLB ヒットにのみ依存する。

DTLB または ITLB 内の TLB ミスは、ハードウェア・ページ・テーブル・ウォークによって処理される。ハードウェア・ページ・テーブル・ウォークは、Itanium 命令セット・アーキテクチャで定義された 8B および 32B VHPT (Virtual Hash Page Table) フォーマットをサポートしている。VHPT データは、L2 キャッシュおよび L3 キャッシュにのみ格納され、L1D には格納されない。

2.5.6.1 データ TLB (DTLB)

1 次 DTLB (DTLB1) は、L1 キャッシュにヒットしたロード・トランザクションの仮想アドレスから物理アドレスへの変換を実行する。1 次 DTLB は、2 つの読み出しポートと 1 つの書き込みポートを持つ。1 次 DTLB は 32 エントリのフル・アソシアティブ・バッファである。1 次 DTLB は 4KB ページをサポートし、より大きなキャッシュも 4KB サブセクション単位でサポートする。

2 次 DTLB (DTLB2) は、ストア操作のデータ・メモリ参照の仮想アドレスから物理アドレスへの変換と、ロード操作の保護チェックを処理する。2 次 DTLB は、128 エントリのフル・アソシアティブ・バッファで、4KB ~ 4GB のページ・サイズをサポートする。DTLB2 は 4 ポートを持つ。128 エントリのうち 64 エントリは、トランスレーション・レジスタ (TR) として構成できる。

2.5.6.2 命令 TLB (ITLB)

1 次 ITLB (ITLB1) は、命令トランザクションが L1I キャッシュにヒットできるように、仮想アドレスから物理アドレスへの変換を実行する。1 次 ITLB は、デュアル・ポート、32 エントリのフル・アソシアティブ・バッファである。1 次 ITLB は 4KB ページのみをサポートする。

2 次 ITLB (ITLB2) は、ITLB1 ミスになった命令メモリ参照の仮想アドレスから物理アドレスへの変換を実行する。2 次 ITLB は、128 エントリのフル・アソシアティブ・バッファで、4KB ~ 4GB のページ・サイズをサポートする。128 エントリのうち 64 エントリは、TR として構成できる。

2.5.7 キャッシュ・コヒーレンシ

3 レベルのキャッシュ・システムでは、異なるキャッシュ内のデータの整合性を保つ必要がある。特定のメモリ・アドレスへの読み出しアクセスは、常にそのアドレスの最新のデータを提供しなければならない。L1 はライトスルー・キャッシュであるため、有効ビットを維持している。有効ビットは、キャッシュ・ラインが有効であるかどうかを示す。L2 キャッシュと L3 キャッシュは、MESI プロトコルを使用してキャッシュ・コヒーレンシを維持する。

2.5.8 ライト・コアレシング

ライト・コアレシング (WC) メモリ・タイプは、フレーム・バッファに対するキャッシュ不可参照のパフォーマンスを向上させるために、複数のデータ書き込みストリームを 1 つの大きな書き込みトランザクションにまとめる。Itanium 2 プロセッサでは、WC ロードは、コアレシング・バッファからではなく、メモリから直接に実行される。

Itanium 2 プロセッサでは、WC アクセス専用の 2 エントリの 128 バイト・バッファ (WCB) を使用する。キャッシュ・ライン内の各バイトが有効ビットを持つ。すべての有効ビットが真の場合、そのラインはフルであり、プロセッサによって排出 (フラッシュ) される。ラインの排出は、不完全なフル・ラインであっても、「先に書き込まれたデータが先にフラッシュされる」順序で開始される。

Itanium 2 プロセッサでは、フレーム・バッファまたはグラフィックス・コントローラに対するキャッシュ可能参照のパフォーマンスを向上させるために、グラフィックス・コントローラなどの外部エージェントが、キャッシュ・ラインのステートを変更しないで、プロセッサのキャッシュからラインを読み出せる。

2.5.9 メモリの順序付け

Itanium 2 プロセッサでは、メモリ・アクセスの緩い順序付けモデルを使用して、メモリ・システムのパフォーマンスの向上が図られている。メモリ・トランザクションは、参照可能性に基づいて順序づけられる。トランザクションが参照可能になるとは、その時点以降、後続するトランザクションがそのトランザクションの動作に影響を与えられなくなる意味である。

Itanium 2 プロセッサでは、トランザクションは、L1D (L1D がその命令を処理できる場合)、L2、または L3 にヒットするか、システムバス上の参照可能ポイントに到達した時点で、参照可能と見なされる。

2.6 IA-32 実行

Itanium 2 プロセッサは、IA-32 アプリケーション・バイナリをサポートしている。つまり、Itanium 2 プロセッサは、単一プロセッサ構成でもマルチプロセッサ構成でも、Itanium ベースのオペレーティング・システム (OS) 上で、IA-32 アプリケーションと Itanium ベースのアプリケーションを混在させて実行できる。Itanium 2 プロセッサの IA-32 エンジンは、EPIC マシンのレジスタ、キャッシュ、実行リソースを利用できるように設計されている。レガシー・バイナリ上で高性能が得られるように、IA-32 エンジンは、命令の動的なスケジューリングを実行する。

本章では、Itanium 2 プロセッサのシステムバス、バス・トランザクション、バス信号の概要を説明する。Itanium 2 プロセッサは、本章で解説していない信号もサポートしている。すべての信号のリストは、『Intel® Itanium® 2 Processor at 1 GHz and 900 MHz Datasheet』と付録 A「信号リファレンス」を参照のこと。

3.1 Itanium® 2 プロセッサのシステムバスの信号伝送

Itanium 2 プロセッサのシステムバスは、コモン・クロック・シグナリングとソース・シンクロナス・データ・シグナリングの 2 種類の信号伝送方式をサポートしている。3.1.1 項と 3.1.2 項は、それぞれの信号伝送の特性について詳しく説明している。各方式に対応するタイミング・チャートでは、四角、三角、丸の記号を使用して、信号のドライブ、受信、サンプリングの時点を示す。四角は、そのクロックで信号がドライブ(アサートまたはデアサート)されることを示す。三角は、その時点またはそれ以前に信号が受信されることを示す。丸は、そのクロックで信号がサンプリング(観察、ラッチ、キャプチャ)されることを示す。黒い線は、0 クロックまたは 1 クロック以上が許容されることを示す。

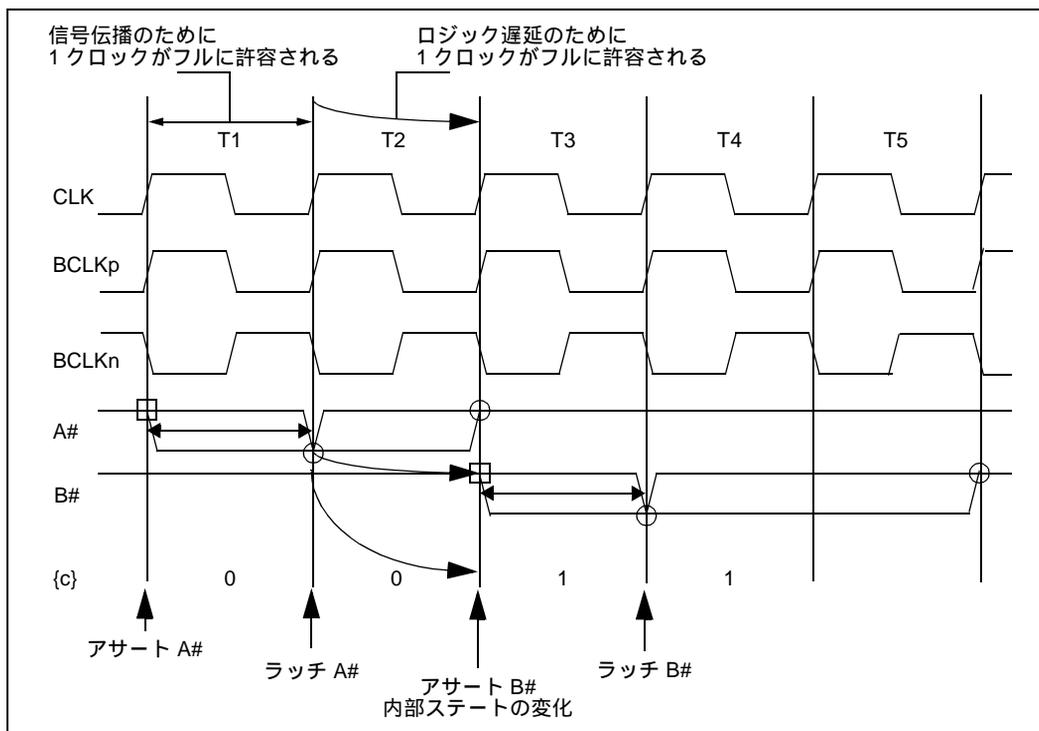
本書のすべてのタイミング・ダイアグラムは、アサートまたはデアサートされたときの信号を示している。システム・バス・エージェントによって観察される信号値には、1 クロックの遅延がある。{rcnt} などのカッコで囲まれた小文字の信号名は内部信号であり、バスに対してドライブされない。バス信号がドライブされてから 1 クロック後にその信号がサンプリングされ、さらにその 1 クロック後に内部状態が変化する。要求フェーズ信号 [REQUEST] など、カッコで囲まれた大文字の信号名は、信号のグループを示す。内部状態や、内部信号が外部信号に与える影響を示すために、タイミング・ダイアグラムに内部信号が含まれる場合もある。内部状態は、バス信号のサンプリングの 1 クロック後に変化する。バス信号は、その信号がドライブされてから 1 クロック後にサンプリングされる。

3.1.1 コモン・クロック・シグナリング

システムバス上のすべての信号は、データバス信号を除いて、シンクロナス・コモン・クロックラッチ・プロトコル(1 倍の転送速度)を使用する。バスクロックの立ち上がりエッジで、システムバス上のすべてのエージェントは、アクティブな出力をドライブし、必要な入力をサンプリングするように要求される。バッファとラッチステージの間の出力バスと入力バスには、追加のロジックは配置されていない。したがって、ラッチ・プロトコルに従って、すべてのバス信号のセットアップ時間とホールド時間は一定に保たれる。システムバスは、(1) 立ち上がりクロックエッジの有効なサンプリング時間帯の間にすべての入力がサンプリングされることと、(2) その影響が次の立ち上がりクロックエッジまで続くことを要求する。この方法で、信号のドライブ、フライト時間、セットアップのために 1 クロックがフルに許容される。また、受信側エージェントの応答の算出に、少なくとも 1 クロックがフルに許容される。

図 3-1 は、バス上に現れるラッチ・バス・プロトコルを示している。後の説明では、このプロトコルは、「A# のアサートが観察されてから 1 クロック後に B# がアサートされる」または「A# がアサートされてから 2 クロック後に B# がアサートされる」として記述される。A# は T1 でアサートされているが、A# のアサートは T2 まで観察されない。A# のアサートが観察されるまでに、(矢印付きの直線で示すように)信号の伝播のためにフルに 1 クロックが経過している。受信側エージェントは、T2 の間に応答を決定し、T3 で B# をアサートする。つまり、受信側エージェントが A# のアサートを観察した時点(T2 の立ち上がりエッジ)から、受信側エージェントが(1 つの矢印を持つ曲線で示すように)応答を算出し、この応答を B# でドライブする時点(T3 の立ち上がりエッジ)までに、フルに 1 クロックサイクルが経過している。同様に、エージェントは、T2 の立ち上がりエッジで A# のアサートを観察し、(最下部の T2 の矢印付き曲線で示すように)T2 クロックをフルに使うって応答を算出している。この応答は、T3 の立ち上がりエッジで {c} 信号でドライブされるはずである(図 3-1 には示されていない)。B# は T3 の立ち上がりエッジでドライブされるが、伝播のために T3 クロックをフルに使用する。B# のアサートは、T4 で観察される。

図 3-1. コモン・クロック・ラッチ・プロトコル



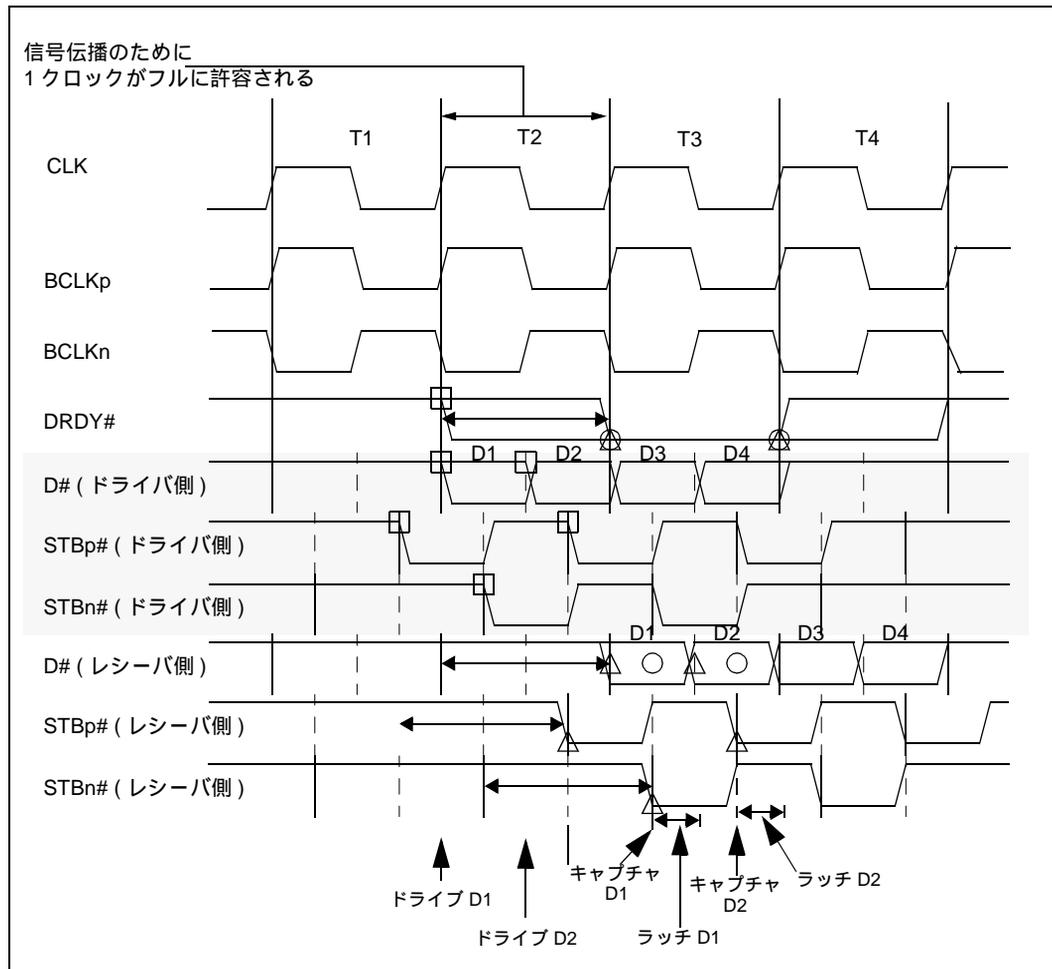
複数のシステム・バス・エージェントによって同じクロックでドライブされる信号は、電気的ローから電気的ハイへの移行時に「ワイヤード OR グリッチ」を示す。この状況を解決するために、このような信号状態の遷移は、デアサートされてから安全に観察されるまでに、2クロックの整合時間をとるように定義されている (図の B# を参照)。この基準を満たす必要があるバス信号は、BINIT#、HIT#、HITM#、BNR#、TND#、BERR# である。

3.1.2 ソース・シンクロナス・シグナリング

データバスは、ソース・シンクロナス・ラッチ・プロトコル (2 倍の転送速度) を使用して動作する。ソース・シンクロナス・ラッチ・プロトコル (図 3-2 を参照) は、ストローブ信号を使ってデータの送信とラッチを行い、妥当な信号フライト時間で非常に高い転送速度が得られる。データバス以外のシステムバスは、常にコモン・クロック・ラッチ・プロトコルを使用する。

ソース・シンクロナス・ラッチ・プロトコルは、コモン・クロック・プロトコルの 2 倍の「周波数」でデータバスを動作させる。通常なら 1 チャンクのデータがドライブされる時間で、2 チャンクのデータをバス上にドライブできる。ただし、ワーストケースのフライト時間は、コモン・クロック・ラッチ・プロトコルと同じである。したがって、最初のデータ転送がラッチされる前に、2 番目のデータ転送がドライブされる場合がある。バスクロックの立ち上がりエッジと 50% 点で、ドライブ側エージェントは新しいデータを送信する。バスクロックの 25% 点と 75% 点で、ドライブ側エージェントはセンタリングされた差分ストローブ信号を送信する。受信側エージェントは、ストローブ信号を使って、確実にデータをキャプチャする。

図 3-2. ソース・シンクロナス・ラッチ・プロトコル



ドライバ側エージェントは、データをドライブする前に、STBp# を事前にドライブする。このエージェントは、データに対してセンタリングされた STBp# と STBn# で、立ち上がりエッジと立ち下がりエッジを送信する。ドライバ側エージェントは、最後のデータの送信後、すべてのストロブ信号をデアサートしなければならない。受信側エージェントは、両方のストロブ信号の差分を使用して、コモンクロックに対して非同期に、有効なデータをキャプチャする。データは、キャプチャされてから 1 コアサイクル以内に、コア内にラッチされる。コモンクロックに同期する信号 (DRDY#) は、有効なデータが送信されたことを受信側エージェントに通知する。

3.2 信号の概要

この節では、Itanium 2 プロセッサの各種の信号の機能について説明する。この節では、機能に基づいて信号を分類している。すべての信号のリストは、『Intel® Itanium® 2 Processor at 1 GHz and 900 MHz Datasheet』を参照のこと。

3.2.1 制御信号

制御信号 (表 3-1) は、プロセッサの基本動作を制御する。

表 3-1. 制御信号

信号の機能	信号名
正相バスクロック	BCLKp
逆相バスクロック	BCLKn
プロセッサとシステム・バス・エージェントのリセット	RESET#
電源正常	PWRGOOD

正相バスクロック (BCLKp) 入力信号は、システム・バス・クロックの差分ペアの正相である。この信号は、この概要の一部の波形では CLK と呼ばれる場合もある。この信号は、信号伝送方式で使用され、バス周波数とクロック期間を指定する。各プロセッサは、コンフィグレーション時に指定された乗数をバス周波数に掛けて、CLK から内部クロックを計算する。詳細は、[第 5 章「コンフィグレーションと初期化」](#)を参照のこと。

逆相バスクロック (BCLKn) 入力信号は、システム・バス・クロックの差分ペアの逆相である。

RESET# 信号は、すべてのシステム・バス・エージェントを確認済みの状態にリセットする。

注： RESET# 信号それ自体は、Itanium 2 プロセッサの内部キャッシュを無効にしない。その次の PAL 呼び出しを使用して、Itanium 2 プロセッサのすべての内部キャッシュを無効にする。修正済みの (ダーティな) キャッシュ・ラインは書き戻されない。RESET# がデアサートされた後、各プロセッサは、コンフィグレーション時に定義された電源投入リセット・ベクタで実行を開始する。

電源正常 (PWRGOOD) 入力信号は、電源投入中にデアサートされ、システムによって RESET# が先にアサートされた後にアサートされなければならない。

3.2.2 アービトレーション信号

アービトレーション信号 (表 3-2) は、バスの所有権 (バス・トランザクションの開始要求) に関するアービトレーションに使用される。

表 3-2. アービトレーション信号

信号の機能	信号名
対称エージェントのバス要求	BREQ[3:0]#, BR[3:0]#
プライオリティ・エージェントのバス要求	BPRI#
次の要求をブロック	BNR#
ロック	LOCK#

BR[3:0]# は、プロセッサの物理ピンである。すべてのプロセッサは、BR0# だけをアサートする。BREQ[3:0]# は、4 個のプロセッサの間のシステム・バス・アービトレーション信号である。4 個のプロセッサそれぞれの BR0# が、個別の BREQ[3:0]# 信号に接続される。

最大 5 つまでのエージェントが、要求バスに関するアービトレーションに同時に参加できる。エージェントの内訳は、1 ~ 4 つの対称エージェント (BREQ[3:0]# に対応) と 1 つのプライオリティ・エージェント (BPRI# に対応) である。各プロセッサは、対称エージェントとしてアービトレーションされる。プライオリティ・エージェントは、通常は I/O エージェントおよびメモリ・エージェントの代わりにアービトレーションされる。要求バスの所有権を取得するのは、トランザクションを開始するために必要な前提条件である。

対称エージェントのバス・アービトレーションは、ラウンドロビン循環優先度方式に基づいて行われる。このアービトレーションは、各エージェントに対して公平で対称的である。対称エージェントは、BREQ_n# 信号をアサートするとバスを要求する。BREQ[3:0]# でサンプリングされた値と最後の対称バス・オーナーに基づいて、すべてのエージェントが同時に、次の対称バス・オーナーを決定する。

プライオリティ・エージェントは、BPRI# をアサートするとバスを要求する。BPRI# がアサートされると、対称アービトレーション方式は一時的に無効にされるが、それ以外は変更されない。BPRI# のアサートがサンプリングされると、BPRI# のデアサートがサンプリングされるまで、対称エージェントは他のアンロックされたトランザクションを発行できない。プライオリティ・エージェントは、常に次のバス・オーナーになる。

任意のバス・エージェントは、BNR# をアサートして、要求バスに対する次のトランザクションの発行をブロックできる。この信号は、通常は、アドレス・バッファまたはデータ・バッファなどのシステム・リソースが一時的にビジーまたはフィル状態になり、次のトランザクションを受け入れられないときにアサートされる。また、バスの初期化の後に、BNR# をアサートして、すべてのバス・エージェントが初期化されるまで、最初のトランザクションを遅らすことができる。

Itanium 2 プロセッサのシステム環境では、LOCK# のアサートやサンプリングは行われない。

3.2.3 要求信号

要求信号 (表 3-3) を使用して、トランザクションを開始できる。

表 3-3. 要求信号

信号の機能	信号名
アドレス・ストローブ	ADS#
要求	REQ[5:0]#
アドレス	A[49:3]#
アドレス・パリティ	AP[1:0]#
要求パリティ	RP#

ADS# のアサートは、トランザクションの開始を定義する。REQ[5:0]#、A[49:3]#、AP[1:0]#、および RP# は、ADS# がアサートされたクロック内で有効である。

ADS# がアサートされたクロック内で、A[49:3]# 信号は、要求の一部としてアクティブロー・アドレスを指定する。アドレスの下位 3 ビットは、0 ~ 8 バイト転送用のバイト・イネーブル信号にマッピングされる。AP[1]# は、アドレス信号 A[49:27]# を保護する。AP[0]# は、アドレス信号 A[26:3]# を保護する。保護される信号とパリティ信号で構成される信号セットの中に偶数の電気的ロー信号がある場合は、システムバス上のパリティ信号は正常である。保護される信号がアクティブ・ハイかアクティブ・ローに関係なく、パリティは電圧レベルを使用して計算される。

要求パリティ (RP#) 信号は、要求ピン REQ[5:0]# とアドレス・ストローブ信号 ADS# を保護する。

3.2.4 スヌープ信号

スヌープ信号 (表 3-4) を使用して、システム・バス・エージェントがスヌープの結果を取得し、トランザクションを制御できる。

表 3-4. スヌープ信号

信号の機能	信号名
グローバル・トランスレーション・キャッシュのページを行わない	TND#
未修正のキャッシュ・ラインの保持	HIT#
修正済みキャッシュ・ラインへのヒット	HITM#

表 3-4. スヌープ信号 (続き)

信号の機能	信号名
トランザクションの完了を据え置く	DEFER#
シーケンシャルなトランザクションの保証	GSEQ#

バス・エージェントは、TND# 信号をアサートして、グローバル・トランスレーション・キャッシュ・パージ (PTC.g) 命令の完了を遅らすことができる。これは、システムバス上で PTC.g トランザクションが完了した後でも可能である。ソフトウェアは、システム内で PTC.g 命令が 1 つだけ実行されているのを保証する。

HIT# 信号と HITM# 信号を使用して、スヌーピング・エージェント内でキャッシュ・ラインが有効か無効か、キャッシング・エージェント内でラインが修正済み (ダーティ) ステートになっているかどうか、トランザクションを延長する必要があるかどうかを指定できる。HIT# 信号と HITM# 信号を使用して、システム・レベルでキャッシュ・コヒーレンスを維持できる。

メモリ・エージェントは、HITM# アクティブを検出した場合、データ・リターンを放棄し、キャッシュ・ラインの暗黙的ライトバックのターゲットになる。メモリ・エージェントは、書き戻されるキャッシュ・ラインと書き込みデータをマージし、メモリを更新しなければならない。また、メモリ・エージェントは、トランザクションの暗黙的ライトバック応答を生成しなければならない。

HIT# と HITM# のアサートが同時にサンプリングされた場合は、キャッシング・エージェントがスヌープ・ステータスを示す準備ができていないため、トランザクションを延長する必要があるという意味である。

DEFER# 信号のデアサートは、トランザクションがインオーダーで完了するのを保証できることを示す。この信号をアサートするエージェントは、適切な応答を生成すると、インオーダー・キューからトランザクションが正しく削除されるように保証する。

GSEQ# 信号がアサートされると、要求側エージェントは、トランザクションがまだ参照可能になっていなくても、次のシーケンシャルなキャッシュ不可書き込みを発行できる。また、GSEQ# 信号のアサートにより、プラットフォームは、トランザクションを再試行しないのを保証し、同じエージェントからの他のキャッシュ不可書き込みに対するトランザクションのシーケンシャルな性質を保証する責任を負う。

3.2.5 応答信号

応答信号 (表 3-5) は、要求側エージェントに応答情報を与える。

表 3-5. 応答信号

信号の機能	信号名
応答ステータス	RS[2:0]#
応答パリティ	RSP#
ターゲット・レディ (書き込みの場合)	TRDY#

要求フェーズで開始された要求は、各エージェントによって保守されるインオーダー・キューに入れられる。応答側エージェントは、インオーダー・キューのトップにあるトランザクションを完了する役割を受け持つ。応答側エージェントは、トランザクションによってアドレス指定されたエージェントである。

書き込みトランザクションでは、応答側エージェントが TRDY# をアサートして、書き込みデータまたはライトバック・データを受け入れる準備が完了したことを示す。暗黙的ライトバックを伴う書き込みトランザクションでは、TRDY# は 2 回アサートされる。1 回目は書き込みデータの転送に対応し、2 回目は暗黙的ライトバック・データの転送に対応する。

RSP# 信号は、RS[2:0]# のパリティ保護を提供する。保護される信号とパリティ信号で構成される信号セットの中に偶数のロー信号がある場合は、システムバス上のパリティ信号は正常である。保護される信号がアクティブ・ハイかアクティブ・ローかに関係なく、パリティは電圧レベルを使用して計算される。

3.2.6 データ信号

データ応答信号 (表 3-6) は、バス上のデータの転送を制御し、データ・バスを提供する。すべてのデータ転送は、2 倍の転送速度で行われる。

表 3-6. データ信号

信号の機能	信号名
データ・レディ	DRDY#, DRDY_C1#, DRDY_C2#
データ・バス・ビジー	DBSY#, DRDY_C1#, DRDY_C2#
ストローブ・バス・ビジー	SBSY#, SBSY_C1#, SBSY_C2#
データ	D[127:0]#
データ ECC 保護	DEP[15:0]#
正相データ・ストローブ	STBp[7:0]#
逆相データ・ストローブ	STBn[7:0]#

DRDY# は、有効なデータがバス上にあり、ラッチを必要としていることを示す。データバスのオナは、有効なデータが転送される各クロックで DRDY# をアサートする。DRDY# をデアサートすると、データ・フェーズに待機状態を挿入できる。

DBSY# は、マルチクロック・データ転送で、最初の DRDY# の前と各 DRDY# のアサートの間、データバスを保留する。シングル・クロック・データ転送では、DBSY# をアサートする必要はない。

SBSY# は、マルチクロック・データ転送で、最初の DRDY# の前と各 DRDY# のアサートの間、ストローブ・バスを保留する。SBSY# は、バス上のすべてのデータ転送でアサートしなければならない。

データバス制御信号 DBSY#、DRDY#、SBSY# のそれぞれが、Itanium 2 プロセッサ上でコピーされ、システム・エージェント内でのデータ・バス・チップのパーティショニングを可能にする。DBSY#、DRDY#、SBSY# 信号の 2 つのコピーは出力専用であり、第 3 のコピーは入力と出力の両方に使用される。

D[127:0]# 信号は、各エージェントの間に 128 ビット・データ・バスを提供する。不完全な転送の場合は、BE[7:0]# と A[4:3]# が、データバスのどのバイトに有効なデータが含まれているかを指定する。

DEP[15:0]# 信号は、D[127:0]# に対してオプションの ECC (エラー修正コード) 保護を提供する。DEP[15:0]# は、どのバイトがイネーブルであるかに関係なく、各クロックでデータバス全体に対して有効な ECC 保護を提供する。

STBp[7:0]# と STBn[7:0]# (および DRDY#) を使用して、ソース・シンクロナス・ラッチ・プロトコルにより、2 倍の転送速度でデータを転送できる。データ転送をドライブするエージェントは、対応するデータ信号と ECC 信号とともにストローブ信号をドライブする。データ転送を受信するエージェントは、このストローブ信号を使用して、有効なデータをキャプチャする。各ストローブ信号ペアは、表 3-7 に示すように、16 のデータ信号と 2 つの ECC 信号に対応付けられる。

表 3-7. STBp[7:0]# と STBn[7:0]# に対応する信号

ストローブ信号	データ信号	ECC 信号
STBp[7]#, STBn[7]#	D[127:112]#	DEP[15:14]#
STBp[6]#, STBn[6]#	D[111:96]#	DEP[13:12]#
STBp[5]#, STBn[5]#	D[95:80]#	DEP[11:10]#
STBp[4]#, STBn[4]#	D[79:64]#	DEP[9:8]#
STBp[3]#, STBn[3]#	D[63:48]#	DEP[7:6]#
STBp[2]#, STBn[2]#	D[47:32]#	DEP[5:4]#
STBp[1]#, STBn[1]#	D[31:16]#	DEP[3:2]#
STBp[0]#, STBn[0]#	D[15:0]#	DEP[1:0]#

3.2.7 据え置き信号

据え置く側のエージェントは、据え置き信号 (表 3-8) を使用して、以前の据え置きトランザクションを完了できる。要求側エージェントが据え置き応答をサポートしている (DPS# がアサートされている) 場合は、すべての据え置き可能トランザクション (DEN# がアサートされている) は、据え置き応答信号を使用できる。

表 3-8. 据え置き信号

信号の機能	信号名
ID ストロープ	IDS#
トランザクション ID	ID[9:0]#

IDS# がアサートされると、据え置き応答が開始される。ID[9:0]# は、DID[9:0]# で送信された、据え置きトランザクションの ID を返す。詳細は、付録 A 「信号リファレンス」を参照のこと。

3.2.8 エラー信号

表 3-9 は、システムバス上のエラー信号を示している。

表 3-9. エラー信号

信号の機能	信号名
バス初期化	BINIT#
バス・エラー	BERR#
温度トリップ	THRMTRIP#
温度アラート	THRMALERT#

BINIT# は、バスの将来の正常な動作の妨げとなるバス状態を通知する。BINIT# のアサートは、電源投入時にコンフィグレーション・レジスタで有効または無効にできる (第 5 章 「コンフィグレーションと初期化」を参照)。BINIT# のアサートが無効になっている場合は、BINIT# はアサートされず、エラーを検出したプロセッサだけがエラー回復処置を実行する。

BINIT# のサンプリングは、電源投入リセット時に有効または無効にできる。BINIT# のサンプリングが無効になっている場合は、BINIT# は無視され、BINIT# のアサートがサンプリングされても、プロセッサは何も処置を実行しない。BINIT# のサンプリングが有効になっている場合は、BINIT# のアサートがサンプリングされると、すべてのプロセッサ・バス・ステート・マシンはリセットされる。すべてのエージェントは、バス・アービトレーション用の自分のローテート ID をリセットし、内部ステート情報は失われる。キャッシュの内容は影響を受けない。適切なプロセッサ・エラー回復処置が行われるためには、BINIT# のサンプリングとアサートが有効になっている必要がある。

BINIT# がアサートされるたびに、マシン・チェック・アボートが発生する。これは電源投入時に設定可能である。

BERR# は、バス・トランザクションによって発生したエラー状態のうち、バス・プロトコルの正常な動作に影響を与えないもの (例えば、メモリ・データ・エラーまたは未修正ラインのスヌープ・エラー) を通知する。BERR# のアサートの原因となるバス・エラーは、プロセッサまたは他のバス・エージェントによって検出される。BERR# のアサートは、電源投入リセット時に有効または無効にできる。BERR# のアサートが無効になっている場合は、BERR# はアサートされない。BERR# のアサートが有効になっている場合は、プロセッサは 2 つの動作モードをサポートする。これは電源投入時に設定可能である (詳細は、5.2.6 項および 5.2.7 項を参照のこと)。BERR# のサンプリングが無効になっている場合は、BERR# のアサートは無視され、プロセッサは何も処置を実行しない。BERR# のサンプリングが有効になっている場合は、BERR# のアサートがサンプリングされ、プロセッサ・コアにマシン・チェック例外が通知される。

BERR# がアサートされるたびに、マシン・チェック例外が発生する。これは電源投入時に設定可能である。

THRMTRIP# は、温度トリップ信号である。Itanium 2 プロセッサは、内部温度センサを使用して、致命的な加熱状態から保護される。このセンサは、偽りのトリップが発生しないように、通常の動作温度より十分に高い温度に設定される。プロセッサが温度トリップ状態に移行すると、データは失われる。この状態は、THRMTRIP# ピンのアサートによってシステムに通知される。この信号は、一度アサートされると、プラットフォームによって RESET# がアサートされるまで、アサートされたままになる。温度センサ自体には、ヒステリシスは組み込まれていない。つまり、ケース温度が指定された最大値以下に下がっている間に、RESET# パルスがプロセッサをリセットする。

温度アラートのオープンドレイン信号は、THRMALERT# ピンによってシステムに伝達される。この信号は、プロセッサの温度ダイオードの測定温度が、センサの高温または低温レジスタ内で設定された温度しきい値データと等しくなるか、それを超えたときにアサートされる。プラットフォームは、この信号を使用して、各種の温度調整機能を実行できる。例えば、外部割り込みを生成し、プロセッサ・コアが加熱していることをオペレーティング・システムに知らせることができる。

3.2.9 実行制御信号

実行制御信号 (表 3-10) は、プロセッサの実行フローを変更する信号で構成される。

表 3-10. 実行制御信号

信号の機能	信号名
プロセッサの初期化	INIT#
プラットフォーム管理割り込み	PMI#
プログラマブル・ローカル割り込み	LINT[1:0]

INIT# は、プロセッサに対するマスク不可割り込みをトリガする。プラットフォームの互換性を保証するために必要なセマンティクスは、PAL ファームウェアの割り込みサービス・ルーチン内で提供される。INIT# は、通常は、ハング状態またはアイドル状態のプロセッサに対して使用される。

PMI# は、プラットフォーム管理割り込みピンである。この信号は、プロセッサに対する最高の優先度を持つ割り込みをトリガする。PMI# は、通常はシステムによって使用され、プラットフォーム固有のファームウェアによって処理されるシステム・イベントをトリガする。

LINT[1:0] は、割り込みインターフェイスによって定義される、プログラマブル・ローカル割り込みピンである。これらのピンは、RESET# の後は無効にされる。LINT[0] は、通常はソフトウェアによって INT (8259 互換のマスク可能な割り込み要求信号) として設定される。LINT[1] は、通常はソフトウェアによって NMI (マスク不可割り込み) として設定される。

3.2.10 IA-32 互換性信号

IA-32 システム環境との互換性を保証するために、FERR#、IGNNE#、A20M# の各信号が用意されている。Itanium 2 プロセッサでは、FERR# 信号は、IA-32 アプリケーションの実行中にアサートされ、マスクされていない浮動小数点エラーを示す。IGNNE# 信号と A20M# 信号は無視される。

3.2.11 プラットフォーム信号

プラットフォーム信号 (表 3-11) は、プラットフォームをサポートする信号である。

表 3-11. プラットフォーム信号

信号の機能	信号名
プロセッサの有無	CPUPRES#

CPUPRES# を使用して、ソケット内の Itanium 2 プロセッサの有無を検出できる。グラウンド (GND) レベルは、Itanium 2 プロセッサがインストールされていることを示す。オープンは、プロセッサがインストールされていないことを示す。

3.2.12 診断信号

診断信号 (表 3-12) は、プロセッサのプロープ、プロセッサのパフォーマンスの監視、バウンダリ・スキャンに関する IEEE 1149.1 仕様のサポートのための信号である。

表 3-12. 診断信号

信号の機能	信号名
ブレイクポイント / パフォーマンスの監視	BPM[5:0]#
バウンダリ・スキャン / テスト・アクセス	TCK、TDI、TDO、TMS、TRST#

BPM[5:0]# は、ブレイクポイント / パフォーマンス監視信号である。これらの信号は、プロセッサ・イベントを監視するためのブレイクポイントとプログラマブル・カウンタのステータスを示す、プロセッサからの出力として設定できる。これらの信号は、プログラムの実行を中断する入力として設定できる。

テスト・クロック (TCK) 信号を使用して、5 つの信号に基づくテスト・アクセス・ポート (TAP) 上の動作を測定できる。テスト・データ・イン (TDI) 信号は、プロセッサ内にシリアル・テスト・データを転送する。テスト・データ・アウト (TDO) 信号は、プロセッサからシリアル・テスト・データを転送する。テスト・モード選択 (TMS) 信号を使用して、TAP コントローラのステータスの変化のシーケンスを制御できる。テスト・リセット (TRST#) 信号を使用して、TAP コントローラを非同期で初期化できる。

Itanium 2 プロセッサは、エラーの検出、封じ込め、修正、回復を実行する、高度なマシン・チェック・アーキテクチャをサポートしている。システムバスは、アドレス信号、要求信号、および応答信号のパリティ保護、ほとんどの制御信号のパリティ保護またはプロトコル保護、データ信号の ECC 保護機能を持っている。

マシン・チェック・アーキテクチャの詳細は、『Itanium® Processor Family Error Handling Guide』を参照のこと。

4.1 エラーの分類

Itanium 2 プロセッサでは、エラーは以下のカテゴリに分類される（順番に重要度が高くなる）。重要度の高いエラーだけを報告するようにシステムを設計することで、システムのロジックを簡単にできる。

1. ハードウェアによって修正されるエラー
このエラーは、プロセッサまたはシステム・ハードウェアによって修正可能である。現在実行中のプロセスは、中断されずに続行される。
2. ファームウェアによって修正されるエラー
このエラーは、ファームウェアによって修正可能である。現在実行中のプロセスは、マシン・チェック・アポルト (MCA) が処理された後に再開される。
3. ローカル MCA を伴う回復可能なエラー
このエラーは、ハードウェアによってもファームウェアによっても修正できない。エラーの影響を受けるエージェントは 1 つだけである。エラーの処理は OS に任せられるが、常に回復が可能であるとは限らない。
4. グローバル MCA を伴う回復可能なエラー
このエラーは、ハードウェアによってもファームウェアによっても修正できない。バス上の複数のエージェントが影響を受ける。エラーの処理は OS に任せられるが、常に回復が可能であるとは限らない。
5. グローバル MCA を伴う回復不能なエラー
このエラーは、ハードウェア、ファームウェア、または OS のいずれによっても修正できない。バス上の複数のエージェントが影響を受ける。システムを再起動する必要がある。

4.2 Itanium® 2 プロセッサ・システム・バスのエラー検出

Itanium 2 プロセッサ・システム・バスの主要なアドレス・バスとデータ・バスは、パリティ保護または ECC 保護を提供する 18 のチェック・ビットによって保護される。16 の ECC ビットは、データバスを保護する。シングル・ビット・データ・エラーは、自動的に修正される。2 ビットのパリティ・コードは、アドレス・バスを保護する。

3 つの制御信号グループは、個別のパリティ・ビット RP#, RSP#, IP[1:0]# によって直接に保護される。バス・プロトコルの仕様は、プロトコル違反エラーを検出できるように厳密に定義されているため、その他のほとんどのバス信号のエラーは、間接的に検出される。ただし、一部のバス信号のエラーは検出できない。

各データ保全性機能は、電源投入時のコンフィグレーションで個別に有効にされるため、エージェントがすべてのデータ保全性機能を有効にする必要はない。第 5 章「コンフィグレーションと初期化」を参照のこと。

4.2.1 直接に保護されるバス信号

ほとんどのシステムバス信号は、パリティまたは ECC によって保護される。表 4-1 は、パリティ信号および ECC 信号と、各パリティ信号および ECC 信号によって保護される信号を示している。

表 4-1. 直接的なバス信号の保護

信号	保護される信号
RP#	ADS#、REQ[5:0]#
AP[0]#	A[26:3]#
AP[1]#	A[49:27]#
RSP#	RS[2:0]#
IP[0]#	IDS#、IDa[9:0]#
IP[1]#	IDS# (デアサート)、IDb[9:2,0]#
DEP[7:0]#	D[63:0]#
DEP[15:8]#	D[127:64]#

- アドレス / 要求バス信号
AP[1:0]# または RP# 上で検出されたパリティ・エラーは、電源投入時のコンフィグレーションで定義されたオプションに従って報告される。
 - アドレス / 要求パリティが無効になっている場合
パリティ・エラーを検出したエージェントは、そのエラーを無視し、通常の動作を続行する。このオプションは、通常は、電源投入時のシステム初期化とシステム診断に使用される。
- 応答信号
応答パリティが有効になっている場合は、RSP# 上で検出されたパリティ・エラーは、そのエラーを検出したエージェントによって、グローバル MCA を伴う回復不能なエラーとして報告される。
- 据え置き信号
IP[1:0]# 上で検出されたパリティ・エラーは、そのエラーを検出したエージェントによって、グローバル MCA を伴う回復不能なエラーとして報告される。
- データ転送信号
Itanium 2 プロセッサのデータバスは、データ・バス・エラー・チェックなしオプションまたは ECC オプションに設定できる。ECC を選択した場合は、シングル・ビット・エラーを自動的に修正でき、ダブル・ビット・エラーとポイズン・データを検出できる。修正されたシングル・ビット ECC エラーは、続行可能なエラーである。ダブル・ビット・エラーとポイズン・データは、ローカル MCA を伴う回復不能なエラーの原因になる。

4.2.2 間接的に保護されるバス信号

一部のバス信号は、パリティや ECC によって直接に保護されない。しかし、信号伝送は厳密なプロトコルに従う必要があるため、これらのバス信号は間接的に保護される。一部のプロセッサまたは他のバス・エージェントは、プロトコル違反がないかどうかチェックすると、バスのエラー検出または修正機能を強化できる。特に断らない限り、P6 ファミリー・プロセッサのシステム・バス・プロトコル・エラーは、致命的エラーとして扱われる。

4.2.3 保護されないバス信号

Itanium 2 プロセッサの以下のシステムバス信号は、ECC やパリティによって保護されない。

- BCLK、RESET#、PWRGOOD#、LINT[1:0]#、CPUPRES#、INIT# は保護されない。
- エラー信号 THRMTRIP#、THRMALERT# は保護されない。

4.2.4 Itanium® 2 プロセッサ・システム・バスのエラー・コード・アルゴリズム

4.2.4.1 パリティ・アルゴリズム

すべてのバス・パリティ信号は、同じアルゴリズムを使用して正常なパリティを計算する。すべての保護される信号がハイであるか、偶数の保護される信号がローである場合は、正常なパリティ信号はハイである。奇数の保護される信号がローである場合は、正常なパリティ信号はローである。保護される信号がアクティブ・ハイかアクティブ・ローかに関係なく、パリティは電圧レベルを使用して計算される。パリティ信号は、保護される信号の数に基づいて「偶数」パリティまたは「奇数」パリティを提供すると見なせるが、本書ではこれらの用語を使用しない。

4.2.4.2 Itanium® 2 プロセッサ・システム・バスの ECC アルゴリズム

Itanium 2 プロセッサのシステムバスは、ECC コードを使用して、シングル・ビット・エラーの修正、ダブル・ビット・エラーの検出、ポイズン・データの送信、1 ニブルに限られたすべてのエラーの検出を行う。システム設計者は、これらのエラーをすべて検出するか、これらのエラーの一部を検出するかを選択できる。また、追加のシステム・レベル・キャッシュ、メイン・メモリ配列、または I/O サブシステム・バッファ内でも、同じ ECC コードを使用できる。

本章では、Itanium 2 プロセッサの設定オプションと初期化について詳しく説明する。

Itanium 2 プロセッサは、シングルプロセッサ構成にもマルチプロセッサ構成にも対応している。1つのシステムバスで1～4個のプロセッサを使用できる。マルチ・システム・バス構成もサポートしている。

5.1 設定の概要

Itanium 2 プロセッサには、ハードウェアによって指定される設定オプションと、PAL によって指定される設定オプションがある。

Itanium 2 プロセッサは、RESET# 信号のアサートからデアサートへの移行時に、プロセッサのハードウェア設定をサンプリングする。サンプリングされた情報によって、それ以降の動作のために、プロセッサや他のバス・エージェントが設定される。これらの設定オプションは、次のリセット以外には変更できない。すべてのリセット操作で、バス・エージェントは再設定される。詳細は、『Intel® Itanium® 2 Processor at 1 GHz and 900 MHz Datasheet』を参照のこと。

Itanium 2 プロセッサは、追加の PAL 設定オプションでも設定できる。PAL 設定オプションは、PAL に対するプロシージャ・コールで変更できる。これらのオプションを変更するときは、複数の Itanium 2 プロセッサ・システム・バス・エージェント間の同期を考慮に入れる必要がある。

5.2 設定機能

表 5-1 は、Itanium 2 プロセッサのシステムバスに関連する設定機能を示している。これらの設定機能は、プロセッサのタイプに固有のコンフィグレーション・レジスタ内のフィールドを介して利用できる。これらの機能の一部は、リセット時に (RESET# 信号のアサートからデアサートへの移行時に) バス信号によって設定される。一部の機能は、PAL による設定が可能である。

表の「信号名」の項目は、リセット時に設定フィールドに影響を与えるバス信号を示す。特定の設定機能で、この項目が「なし」になっている場合は、この設定フィールドはリセット時にバス信号によって設定されない。「値」の項目は、その設定機能の推奨バス信号値を示す。特定の設定機能で、この項目が「0」になっている場合は、リセット時にバス信号がデアサートされる。「1」になっている場合は、リセット時にバス信号がアサートされる。「なし」になっている場合は、この設定機能はリセット時にバス信号によって設定されない。

「PAL 呼び出し」の項目は、設定機能を制御できる PAL 呼び出し (存在する場合) を示す。特定の設定機能で、この項目が「なし」になっている場合は、この設定機能は PAL 呼び出しでは設定できない。また、この設定フィールドを読み出す PAL 呼び出しも定義されていない。

「制御」の項目は、設定フィールドに適用される PAL 読み出し / 書き込み制御を示す。特定の設定機能で、この項目が「読み出し」になっている場合は、PAL による読み出しだけが可能である。「読み出し / 書き込み」になっている場合は、PAL による読み出しと修正が可能である。

「デフォルト」の項目は、リセット後の設定フィールドのデフォルト値を示す。バス信号によって設定される設定機能の場合、この項目は、「バス信号値」の項目に示すバス信号値によって設定されるデフォルト値を示す。

要求バス・パーキング機能を有効にすると、要求トラフィックのパターンとシステム・エージェントによっては、パフォーマンスに多少影響を与える。システムは、リセット時に、A15# 信号を使用して、必要に応じてこの機能を設定できる。

表 5-1. 電源投入時の設定機能

機能	バス信号		PAL 呼び出し	制御	デフォルト値
	信号名	値			
データ・エラー・チェック有効	なし	なし	PAL_BUS_SET_FEATURES (書き込み制御の場合)と PAL_BUS_GET_FEATURES (読み出し制御の場合)	読み出し/ 書き込み	無効
応答 /ID エラー・チェック有効	なし	なし			
アドレス / 要求エラー・チェック有効	なし	なし			
BERR# アサート有効	なし	なし			
BERR# サンプリング有効	なし	なし			
BINIT# アサート有効	なし	なし			
E ステートのキャッシュ・ライン 置換時のキャッシュ・ライン置換 トランザクション有効	なし	なし			
S ステートのキャッシュ・ライン 置換時のキャッシュ・ライン置換 トランザクション有効	なし	なし			
BINIT# サンプリング有効	A10#	0			
要求バス・パーキング有効	A15#	0			
インオーダー・キュー段数 1	A7#	0	PAL_BUS_GET_FEATURES	読み出し	無効、デフォルトの IOQ 段数は 8
出力トライステート有効	A[31:28]#	0000	なし	読み出し	無効
対称アービトレーション ID	BR0#、 BR1#、 BR2#、 BR3#	BREQ0# のアサート が必要	PAL_FIXED_ADDR	読み出し	BREQ0# と BR[3:0]# の間 のバス・マッ ピングに基づ いて決定され る。
クロック比	A[21:17]#	00000	PAL_FREQ_RATIOS	読み出し	2/8

5.2.1 データ・バス・エラー・チェック

Itanium 2 プロセッサのデータ・バス・エラー・チェック機能を有効または無効にできる。RESET# のアサート後は、データ・バス・エラー・チェック機能は常に無効になる。PAL からシステムに制御が渡される前に、データ・パリティ・エラー・チェック機能は有効になる。データ・バス・エラー・チェック機能は、PAL の呼び出しによって有効にできる。この機能の詳細は、『Intel® Itanium® アーキテクチャ・ソフトウェア・デベロッパーズ・マニュアル』を参照のこと。

5.2.2 応答 /ID 信号パリティ・エラー・チェック

Itanium 2 プロセッサのシステムバスは、応答信号 RS[2:0]# およびトランザクション ID 信号 ID[9:0]# のパリティ保護をサポートしている。RESET# のアサート後は、応答信号パリティ・チェック機能は無効になる。PAL からシステムに制御が渡される前に、応答パリティ信号チェック機能は有効になる。応答パリティ信号チェック機能は、PAL の呼び出しによって有効または無効にできる。

5.2.3 アドレス / 要求信号パリティ・エラー・チェック

Itanium 2 プロセッサのアドレス・バスは、要求信号 A[49:3]#、ADS#、REQ[4:0]# のパリティ保護をサポートしている。RESET# のアサート後は、要求信号パリティ・チェック機能は無効になる。PAL からシステムに制御が渡される前に、アドレス / 要求パリティ・エラー・チェック機能は有効になる。このチェック機能は、PAL の呼び出しによって有効または無効にできる。

5.2.4 イニシエータ・バス・エラーでの BERR# のアサート

Itanium 2 プロセッサのシステム・バス・エージェントがバス・エラーを検出したときの BERR# 信号のアサートを有効にできる。RESET# のアサート後は、検出されたエラーに対する BERR# 信号アサート機能は無効になる。この機能は、PAL の呼び出しによって有効にできる。

5.2.5 ターゲット・バス・エラーでの BERR# のアサート

Itanium 2 プロセッサの受信側 (ターゲット) バス・エージェントがエラーを検出したときの BERR# 信号のアサートを有効にできる。RESET# のアサート後は、ターゲット・バス・エラーでの BERR# 信号アサート機能は無効になる。この機能は、PAL の呼び出しによって有効にできる。

5.2.6 BERR# サンプリング

BERR# サンプリング・ポリシーが有効になっている場合は、BERR# 入力の受信側エージェントは、グローバル・マシン・チェック・アポート (MCA) を発生させる。この機能は、PAL の呼び出しによって有効にできる。

5.2.7 BINIT# エラーのアサート

BINIT# エラーのアサートが有効になっている場合は、Itanium 2 プロセッサのシステム・バス・エージェントは、バス・プロトコル違反に対して BINIT# 信号をアサートする。RESET# のアサート後は、BINIT# 信号アサート機能は無効になる。この機能は、PAL の呼び出しによって有効にできる。

5.2.8 BINIT# エラーのサンプリング

RESET# のアサートからデアサートへの移行時に A[10]# のアサートがサンプリングされた場合は、BINIT# 入力の受信側エージェントによるバス初期化の制御が有効になる。

5.2.9 インオーダー・キューのパイプライン化

RESET# のアサートからデアサートへの移行時に A[7]# のアサートがサンプリングされた場合は、Itanium 2 プロセッサのシステム・バス・エージェントのインオーダー・キューの段数は 1 に設定される。RESET# のアサートからデアサートへの移行時に A[7]# のデアサートがサンプリングされた場合は、インオーダー・キューの段数はデフォルトにより 8 に設定される。この機能は、PAL の呼び出しでは変更できない。

5.2.10 要求バス・パーキング有効

Itanium 2 プロセッサのシステム・バス・エージェントがアイドル時の要求バス上にパークするように設定できる。RESET# のアサートからデアサートへの移行時に A[15]# のアサートがサンプリングされた場合は、最後に要求バスを所有していたプロセッサが、アイドル状態の要求バス上にパークする。RESET# のアサートからデアサートへの移行時に A[15]# のデアサートがサンプリングされた場合は、どのプロセッサも要求バス上にパークしない。

5.2.11 対称エージェントのアービトレーション ID

Itanium 2 プロセッサのシステムバスは、1 ~ 4 つのバス・エージェント間の対称分散アービトレーションをサポートしている。各プロセッサは、コンフィグレーション時に割り当てられるエー

エージェント ID に基づいて、アービトレーション・プライオリティ・キュー内の自分の最初の位置を特定する。エージェント ID は、0、2、4、または 6 である。特定の Itanium プロセッサ・システム・バス上の各論理プロセッサは、異なるエージェント ID を持っていなければならない。

BREQ[3:0]# バス信号は、表 5-2 と図 5-1 に示す循環方式で、4 つの対称エージェントに接続される。BREQ[3:0]# バス信号は、表 5-3 と図 5-2 に示す循環方式で、2 つの対称エージェントに接続される。各対称エージェントは、1 つの I/O ピン (BR0#) と 3 つの入力専用ピン (BR1#、BR2#、BR3#) を持つ。

表 5-2. Itanium® 2 プロセッサの BREQ[3:0]# バス信号の相互接続 (4 ウェイ・プロセッサ)

バス信号	エージェント 0 のピン	エージェント 1 のピン	エージェント 2 のピン	エージェント 3 のピン
BREQ[0]#	BR[0]#	BR[3]#	BR[2]#	BR[1]#
BREQ[1]#	BR[1]#	BR[0]#	BR[3]#	BR[2]#
BREQ[2]#	BR[2]#	BR[1]#	BR[0]#	BR[3]#
BREQ[3]#	BR[3]#	BR[2]#	BR[1]#	BR[0]#

表 5-3. Itanium® 2 プロセッサの BREQ[3:0]# バス信号の相互接続 (2 ウェイ・プロセッサ)

バス信号	エージェント 0 のピン	エージェント 1 のピン
BREQ[0]#	BR[0]#	BR[1]#
BREQ[1]#	BR[1]#	BR[0]#
BREQ[2]#	使用しない	使用しない
BREQ[3]#	使用しない	使用しない

図 5-1. BR[3:0]# の物理的な相互接続 (4 つの対称エージェント)

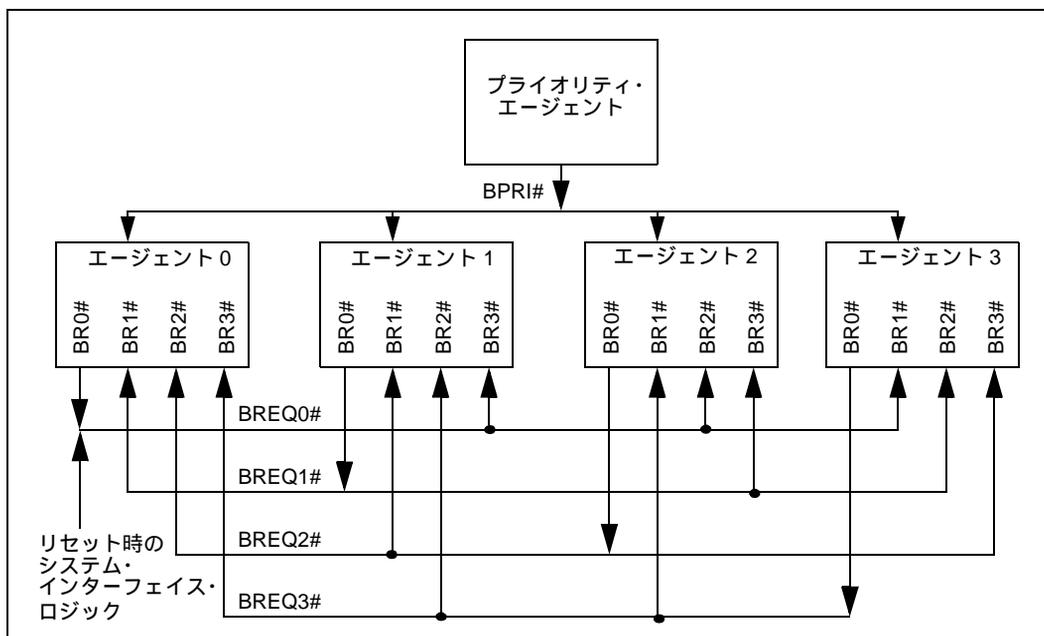
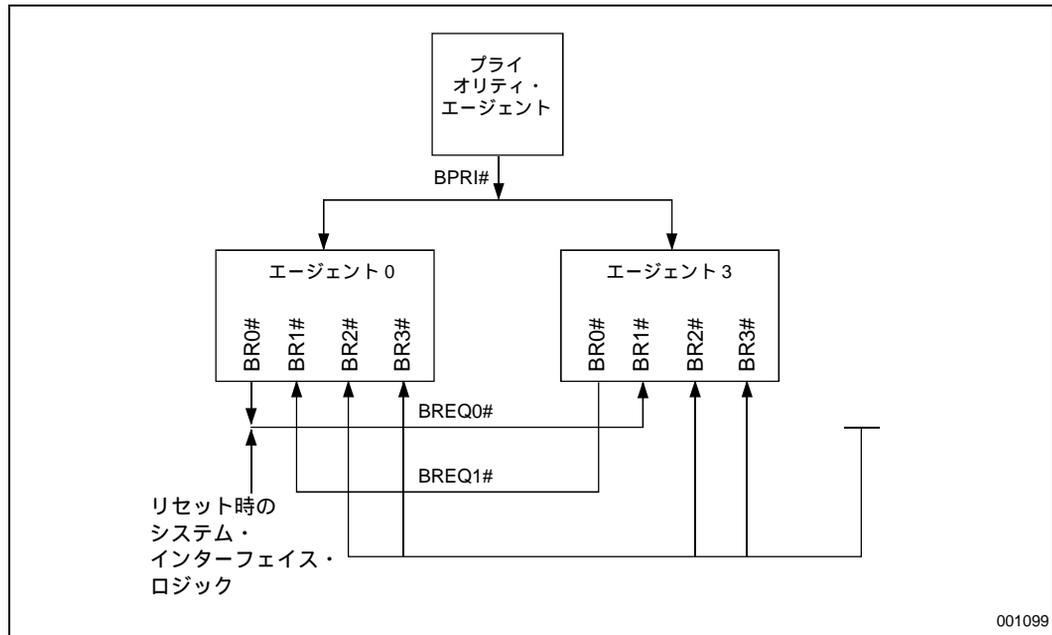


図 5-2. BR[3:0]# の物理的な相互接続 (2 つの対称エージェント)



システム・インターフェイスは、RESET# のアサートからデアサートへの移行時に、BREQ0# バス信号をアサートする役割を受け持つ。BREQ[3:1]# バス信号は、デアサートされたままになる。すべてのプロセッサは、RESET# のアサートからデアサートへの移行時に、自分の BR[3:1]# ピンをサンプリングし、サンプリングした値から自分のアービトレーション ID を確定する。

各物理プロセッサは、異なるアービトレーション ID とエージェント ID を持つ論理プロセッサである (表 5-4 を参照)。

表 5-4. アービトレーション ID の設定¹

BR0#	BR1#	BR2#	BR3#	アービトレーション ID	報告されるエージェント ID
L	H	H	H	0	0
H	H	H	L	1	2
H	H	L	H	2	4
H	L	H	H	3	6

1. L と H は電氣的レベルを示す。

5.2.12 クロック周波数比

表 5-5 は、Itanium 2 プロセッサのシステムバス比の設定を示している。

表 5-5. Itanium® 2 プロセッサのシステムバス周波数とプロセッサ・コア周波数の比の設定

バス周波数をプロセッサ周波数で割った値	A[21]#	A[20]#	A[19]#	A[18]#	A[17]#
2/9	1	0	1	1	0
2/10	1	0	1	0	1

5.3 初期化の概要

プロセッサと PAL ファームウェアは、リセット時にプロセッサの初期化とテストを実行する。

5.3.1 RESET# による初期化

Itanium 2 プロセッサは、RESET# 信号アクティブを検出すると、初期化を開始する。RESET# 信号のアサートはマスク不可であり、IA-32 命令と Itanium 命令を含むすべての命令境界を無視する。

表 5-6 は、リセット時にプロセッサ・ハードウェアと PAL ファームウェアによって初期化されるアーキテクチャ・ステートを示している。その他のすべてのアーキテクチャ・ステートは、ハードウェア・リセット時には未定義である。レジスタの詳細は、『インテル® Itanium® アーキテクチャ・ソフトウェア・デベロッパーズ・マニュアル』を参照のこと。

表 5-6. Itanium® 2 プロセッサのリセット・ステート (PAL ファームウェアの実行後)

プロセッサのリソース	記号	値	説明
命令ポインタ	IP	詳細は、『インテル® Itanium® アーキテクチャ・ソフトウェア・デベロッパーズ・マニュアル』を参照	Itanium® 2 プロセッサでは、SALE_RESET の入口点
レジスタ・スタック・コンフィグレーション・レジスタ	RSC	mode=0	強制レイジー・モード
現行フレーム・マーカ	CFM	sof=96、sol=0、sor=0、rrbs=0	すべての物理汎用レジスタは使用可能、レジスタ・ステートは未定義、汎用レジスタ・フレーム内のローカル変数なし、汎用レジスタ・フレーム内のローテーションなし、FR、GR、および PR レジスタのリネーム・ベースは 0 に設定
トランスレーション・レジスタ	TR	無効	すべての TLB はクリアされる
トランスレーション・キャッシュ	TC	無効	すべての TLB はクリアされる
キャッシュ	—	無効	すべてのキャッシュは無効になる

5.3.2 INIT による初期化

Itanium 2 プロセッサは、INIT 割り込みをサポートしている。INIT は、INIT# 信号のアサートまたは INIT 割り込みメッセージによって開始される。マシン・チェック (MC) の実行中以外は、INIT はマスクできない。マシン・チェックの実行中は、INIT 割り込みは保留される。INIT は、命令の境界で認識される。INIT 割り込みは、プロセッサのアーキテクチャ・ステート、キャッシュのステート、モデル固有レジスタ、整数ステート、浮動小数点ステートに影響を与えない。

表 5-7 は、INIT によって変更されるプロセッサ・ステートを示している。レジスタの詳細は、『インテル® Itanium® アーキテクチャ・ソフトウェア・デベロッパーズ・マニュアル』を参照のこと。

表 5-7. Itanium® プロセッサの INIT ステート

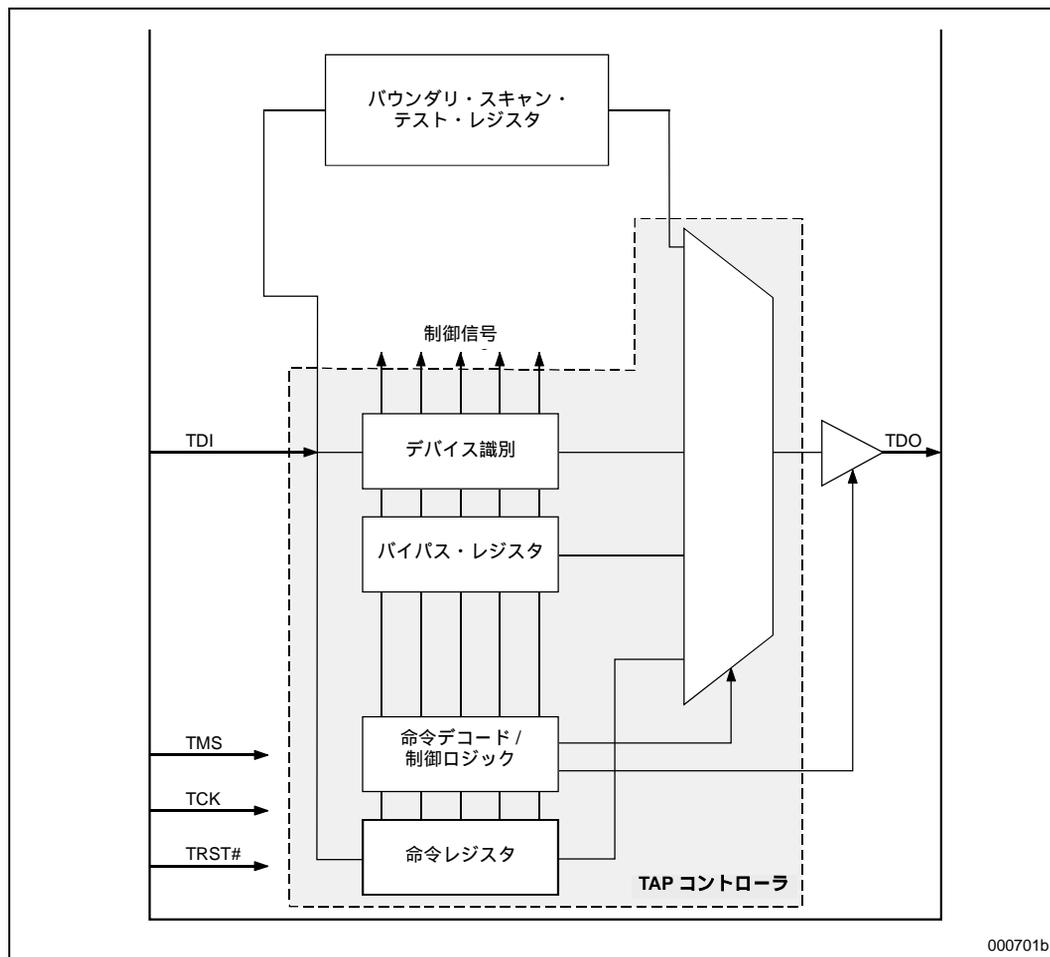
プロセッサのリソース	記号	値	説明
命令ポインタ	IP	詳細は、『インテル® Itanium® アーキテクチャ・ソフトウェア・デベロッパーズ・マニュアル』を参照	Itanium® 2 プロセッサでは、PALE_INIT の入口点
割り込み命令バンドル・ポインタ	IIP	IP の元の値	INIT の時点での IP の値
割り込みプロセッサ・ステータス・レジスタ	IPSR	PSR の元の値	INIT の時点での PSR の値
割り込み機能ステート	IFS	v=0	IFS を無効にする

本章では、Itanium 2 プロセッサのテスト・アクセス・ポート (TAP) ロジックの機能について説明する。TAP は、IEEE 1149.1 (JTAG) 仕様に適合している。本章では、1149.1 仕様に適合するテスト・ロジックの基本的な機能について説明する。IEEE 1149.1 仕様の詳細は、公刊された規格¹と、この仕様に関して業界で広く使われている資料を参照のこと。

TAP の簡単なブロック・ダイアグラムを図 6-1 に示す。Itanium 2 プロセッサは、統合型 TAP コントローラ、バウンダリ・スキャン・レジスタ、4 つの入力ピン (TDI、TCK、TMS、TRST#)、1 つの出力ピン (TDO) を搭載している。統合型 TAP コントローラは、命令レジスタ、デバイス ID レジスタ、バイパス・レジスタ、制御ロジックで構成される。

特定のバウンダリ・スキャン・チェーンの詳細は、『Intel® Itanium® 2 Processor Boundary Scan Description Language (BSDL) Model』を参照のこと。

図 6-1. テスト・アクセス・ポートのブロック・ダイアグラム



1. ANSI/IEEE 規格 1149.1-1990 (IEEE 規格 1149.1a-1993 を含む)、"IEEE Standard Test Access Port and Boundary Scan Architecture," IEEE Press 刊、Piscataway NJ、1993 年。

6.1 インターフェイス

TAP スキャン・チェーンは、プロセッサ・パッケージ上の 5 つの専用ピンを使用してシリアルにアクセスされる。

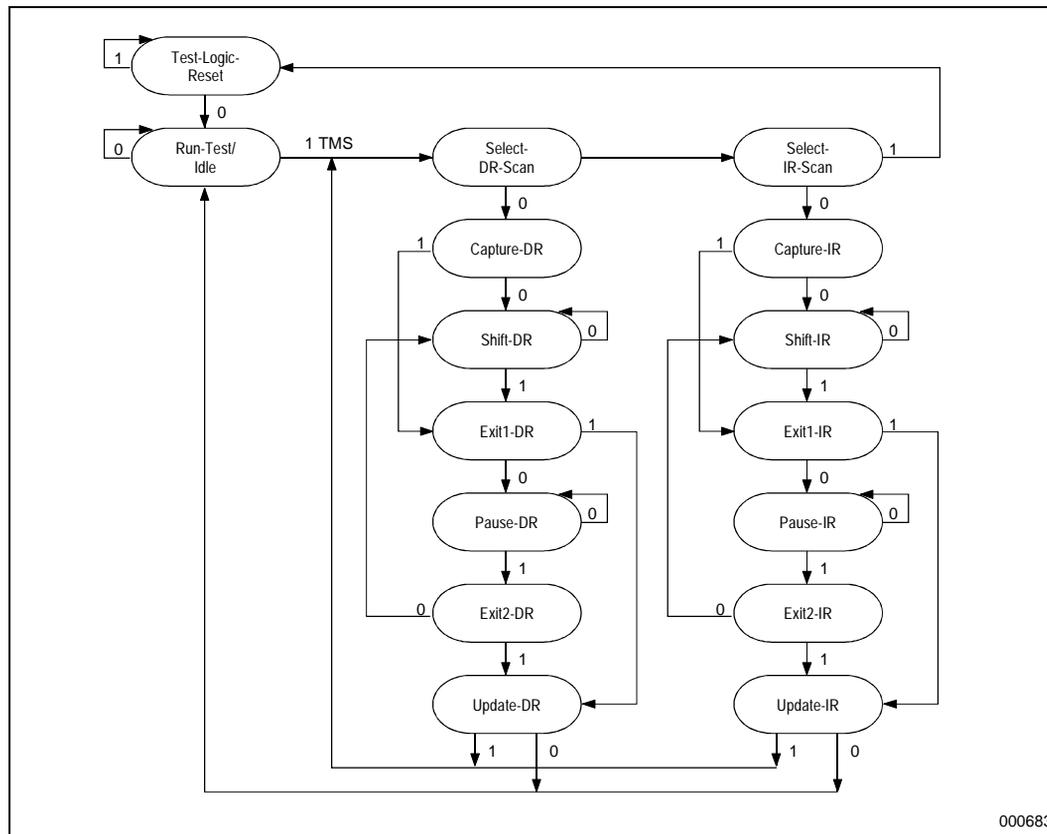
- TCK: TAP クロック信号
- TMS: 「テスト・モード選択」。TAP 有限ステート・マシンを制御する。
- TDI: 「テスト・データ入力」。テスト命令とテスト・データをシリアルに入力する。
- TRST#: 「テスト・リセット」。TAP ロジックをリセットする。
- TDO: 「テスト・データ出力」。このピンからテスト出力がシリアルに読み出される。

TMS、TDI、TDO は、TCK と同期して動作する (TCK は、他のすべてのプロセッサ・クロックから独立している)。TRST# は、非同期入力信号である。

6.2 TAP ロジックへのアクセス

TAP は、IEEE 1149.1 仕様に適合する TAP コントローラ有限ステート・マシンを介してアクセスされる。この有限ステート・マシン (図 6-2) には、リセット・ステート、テスト実行/アイドル・ステート、2 つの大きな分岐が含まれる。これらの分岐によって、TAP 命令レジスタまたはいずれかのデータ・レジスタにアクセスできる。TMS ピンは、この有限ステート・マシンを横断する制御入力として使用される。TAP 命令とテスト・データは、TDI ピンを使用して、(それぞれ Shift-IR ステートおよび Shift-DR ステートで) シリアルにロードされる。ステートの移行は、TCK の立ち上がりエッジで行われる。

図 6-2. TAP コントローラのステート・ダイアグラム



TAP コントローラ・ステート・マシンの各ステートの簡単な説明を以下に示す。各ステートとその動作の詳細は、IEEE 1149.1 規格を参照のこと。

- Test-Logic-Reset: このステートでは、テスト・ロジックは無効にされ、プロセッサは正常に動作する。このステートでは、命令レジスタ内の命令は強制的に IDCODE になる。TAP 有限ステート・マシン (TAPFSM) は、元のステートに関係なく、TMS 入力 が 5 クロック以上にわたってアサートされた場合、常に Test-Logic-Reset ステートに移行する。また、TRST# ピンがアサートされた場合や電源投入時にも、TAP コントローラはただちに Test-Logic-Reset ステートに移行する。TRST# ピンがアサートされている間は、TAPFSM はこのステートを終了できない。
- Run-Test/Idle: 各スキャン動作の間のコントローラ・ステート。コントローラは、一度このステートに移行すると、TMS がローになっている間はこのステートのままになる。このステートでは、選択されたテスト・ロジック内の動作は、特定の命令が発行されたときのみ行われる。このステートで機能を実行しない命令の場合は、現在の命令によって選択されたすべてのテスト・データ・レジスタは、以前のステートを保持する。
- Select-IR-Scan: これは一時的なコントローラ・ステートである。このステートでは、現在の命令によって選択されたすべてのテスト・データ・レジスタは、以前のステートを保持する。
- Capture-IR: このステートでは、命令レジスタに含まれるシフト・レジスタは、TCK の立ち上がりエッジで、(最下位 2 ビットが “01” の) 固定値をロードする。このステートでは、命令レジスタ (現在の命令) のラッチ・パラレル出力は変化しない。
- Shift-IR: 命令レジスタに含まれるシフト・レジスタは、TDI と TDO の間に接続され、TCK の各立ち上がりエッジで、シリアル出力の方向に 1 段シフトされる。出力は、TCK の立ち下がりエッジで、TDO に到着する。このステートでは、現在の命令は変化しない。
- Exit-IR: これは一時的なステートである。このステートでは、現在の命令は変化しない。
- Pause-IR: 命令レジスタのシフトを一時的に停止できる。このステートでは、現在の命令は変化しない。
- Exit2-IR: これは一時的なステートである。このステートでは、現在の命令は変化しない。
- Update-IR: 命令レジスタ内にシフトされた命令は、TCK の立ち下がりエッジで、命令レジスタのパラレル出力内にラッチされる。新しい命令が一度ラッチされると、その命令は、次の Update-IR まで (または TAPFSM がリセットされるまで)、現在の命令のままになる。
- Select-DR-Scan: これは一時的なコントローラ・ステートである。現在の命令によって選択されたすべてのテスト・データ・レジスタは、以前のステートを保持する。
- Capture-DR: このステートでは、TCK の立ち上がりエッジで、現在の命令によって選択されたテスト・データ・レジスタに、データがパラレルにロードされる。現在の命令によって選択されたテスト・データ・レジスタがパラレル入力を持っていない場合や、選択されたテストでキャプチャ機能が不要な場合は、テスト・データ・レジスタは以前のステートを保持する。
- Shift-DR: 現在の命令によって選択された結果、TDI と TDO の間に接続されたデータ・レジスタは、TCK の各立ち上がりエッジで、シリアル出力の方向に 1 段シフトされる。出力は、TCK の立ち下がりエッジで、TDO に到着する。データ・レジスタがラッチ・パラレル出力を持っている場合、新しいデータがシフトされている間は、ラッチ値は変化しない。
- Exit1-DR: これは一時的なステートである。現在の命令によって選択されたすべてのデータ・レジスタは、以前の値を保持する。
- Pause-DR: TCK を停止することなく、選択されたデータ・レジスタのシフトを一時的に停止できる。現在の命令によって選択されたすべてのレジスタは、以前の値を保持する。

- Exit2-DR: これは一時的なステートである。現在の命令によって選択されたすべてのデータ・レジスタは、以前の値を保持する。
- Update-DR: 一部のテスト・データ・レジスタは、ラッチ・パラレル出力を持っている。これは、特定の命令に回答して、データがシフト・レジスタ・パス内でシフトされている間に、パラレル出力が変化するのを防ぐためである。データは、TCK の立ち下がりエッジで、シフト・レジスタ・パスから、これらのテスト・データ・レジスタのパラレル出力内にラッチされる。

6.3 TAP レジスタ

TAP を介してアクセスできるすべてのテスト・レジスタは、以下のとおりである。

1. バウンダリ・スキャン・レジスタ

バウンダリ・スキャン・レジスタは、複数のシングルビット・シフト・レジスタで構成される。バウンダリ・スキャン・レジスタは、Itanium 2 プロセッサ上のすべての入力ピンから出力ピンへのシフト・レジスタ・パスを提供する。データは、バウンダリ・スキャン・レジスタを介して、TDI から TDO に転送される。

2. バイパス・レジスタ

バイパス・レジスタは、TDI と TDO の間の最短経路を提供する 1 ビット・シフト・レジスタである。バイパス・レジスタは、ボード上のコンポーネントがテスト動作を実行していないときに選択される。バイパス・レジスタは、スキャン・サイクルの始点でロジック 0 をロードする。

3. デバイス識別 (ID) レジスタ

デバイス ID レジスタには、メーカーの識別コード、バージョン番号、部品番号が格納される。デバイス ID レジスタは、IEEE 1149.1 仕様の規定に従って、32 ビットの固定長を持つ。

4. 命令レジスタ

命令レジスタには、BYPASS、EXTEST、SAMPLE/PRELOAD、IDCODE、HIGHZ、CLAMP の各命令のうち 1 つを指定する、4 ビットのコマンド・フィールドが 1 つ含まれる。命令レジスタの最上位ビットは TDI に接続され、最下位ビットは TDO に接続される。

6.4 TAP 命令

表 6-1 は、IEEE 1149.1 規格に定義された、TAP コントローラ用の命令を示している。IEEE 1149.1 の規定により、BYPASS (すべて 1) を除くすべての TAP 命令の命令コードは、0000 xxxx の形式になる。

表 6-1. Itanium® 2 プロセッサの TAP コントローラ用の命令

命令	エンコード (2 進値)	エンコード (16 進値)
IEEE 1149.1 規格		
BYPASS	1111 1111	FFh
EXTEST	0000 0000	00h
SAMPLE/PRELOAD	0000 0001	01h
追加の命令		
IDCODE	0000 0010	02h
HIGHZ	0000 1000	08h
CLAMP	0000 1011	0Bh

- BYPASS: バイパス・レジスタには、1 段のシフト・レジスタが含まれる。バイパス・レジスタは、TDI ピンと TDO ピンの間のシリアルな最短経路を提供する。このバイパスにより、システム・ボード上の他のコンポーネントとの間でテスト・データを迅速に転送できる。
- EXTEST: この命令は、TDI を通して、バウンダリ・スキャン・チェーン内にデータをシリアルにロードする。また、この命令は、出力バッファがバウンダリ・スキャン・レジスタ内に保持されているデータをドライブするように強制する。この命令と SAMPLE/PRELOAD を組み合わせると、コンポーネント間のボードレベルの相互接続をテストできる。
- SAMPLE/PRELOAD: この命令は、入力バッファからデータをサンプリングする。サンプリングされたデータは、バウンダリ・スキャン・レジスタ内にキャプチャされ、TDO ピンからシリアルにアンロードされる。また、この命令は、次のバウンダリ・スキャン命令を選択する前に、バウンダリ・スキャン・チェーン内にデータを事前にロードできる。この命令と EXTEST を組み合わせると、コンポーネント間のボードレベルの相互接続をテストできる。
- IDCODE: この命令は、デバイス識別値を TDO にシフトできるように、TDI と TDO の間にデバイス ID レジスタを置く。デバイス ID レジスタには、メーカーの識別コード、部品番号、バージョン番号が格納される。この命令は、TAP のリセット後のデフォルトの命令である。
- HIGHZ: この命令は、コンポーネントのすべての出力バッファを非アクティブ・ドライブ・ステートにする。このステートで、コンポーネントを損傷する危険を冒さずに、ボードレベルのテストを実行できる。HIGHZ 命令の実行中は、バイパス・レジスタは TDI と TDO の間に置かれる。
- CLAMP: この命令は、出力バッファがバウンダリ・スキャン・チェーン内のデータをドライブする間のバイパス・レジスタを選択する。この命令は、データがシフトされる間、バウンダリ・スキャン・チェーン内の値から受信側コンポーネントを保護する。

6.5 リセット動作

TAP とそれに関連するハードウェアをリセットするには、TAP コントローラ有限ステート・マシンを Test-Logic-Reset ステートに移行する。TAP は、リセット時には完全に無効になる（つまり、TAP をリセットすると、プロセッサはあたかも TAP が存在しないかのように動作する）。ただし、TAP には、プロセッサの通常のリセット信号にตอบสนองするロジックは存在しない。TAP は、次のいずれかの方法で Test-Logic-Reset ステートに移行できる。

- プロセッサの電源を投入する。これで、TAP コントローラは自動的に（非同期で）リセットされる。
- 任意の時点で TRST# ピンをアサートする。これで、TAP コントローラは非同期でリセットされる。
- TCK の連続 5 サイクルにわたって、TMS ピンをハイのままにする。これで、TAP コントローラは Test-Logic-Reset ステートに移行する。



Itanium 2 プロセッサは、インターゲット・プローブ (ITP) 機器と (ロジック・アナライザ・インターフェイス (LAI) を介して) ロジック・アナライザ機器をサポートしている。これらの機器により、プロセッサとシステムバスの動作を監視できる。各機器には、設計と使用に関して独自の留意点がある。

7.1 インターゲット・プローブ (ITP)

Itanium 2 プロセッサは、プログラムの実行の制御、レジスタ / メモリ / IO アクセス、ブレークポイント制御のための ITP をサポートしている。ITP ツールは、デバッガおよびエミュレータが持っている機能の一部を提供する。ITP を使用しても、プロセッサ信号の高速動作には影響を与えないため、システムバスは最大限の動作速度を維持できる。

ITP の詳細は、『ITP700 Debug Port Design Guide』を参照のこと。

7.2 ロジック・アナライザ・インターフェイス (LAI)

ロジック・アナライザ・インターフェイス (LAI) モジュールを使用して、基板上の信号にロジック・アナライザを接続できる。サードパーティのロジック・アナライザ・ベンダは、バス・モニタリング機能を持つ各種の製品を提供している。

Itanium 2 プロセッサのシステムバスは、ロジック・アナライザ機器を使ってモニタできる。Itanium 2 プロセッサのマルチプロセッサ・システムは非常に複雑なため、システムのデバッグと動作確認のためにシステムバス信号のプローブ / キャプチャ機能を用意する場合は、LAI がきわめて重要である。LAI を利用できる Itanium 2 プロセッサ・ベースのシステム設計にあたっては、機械的な条件と電気的な条件の 2 種類の留意点に注意する必要がある。詳細は、ロジック・アナライザ・ベンダまでお問い合わせのこと。

本章には、Itanium 2 プロセッサのすべてのシステムバス信号の説明がアルファベット順に記載されている。本章の最後の表は、信号を方向別（出力、入力、I/O）にまとめたものである。

プロセッサ固有のピンを含むすべてのピンアウトのリストは、『Intel® Itanium® 2 Processor at 1 GHz and 900 MHz Datasheet』を参照のこと。

A.1 アルファベット順の信号リファレンス

A.1.1 A[49:3]# (I/O)

アドレス (A[49:3]#) 信号 (バイト・イネーブル付き) は、 2^{50} バイトの物理メモリ空間を定義する。ADS# がアクティブの場合、これらのピンは、トランザクションのアドレスを送信する。また、これらのピンは、トランザクションの識別子や、ADS# のアサートの次のサイクル内の外部機能など、トランザクションに関連する他の情報の送信にも使用される。これらの信号は、Itanium 2 プロセッサのシステムバス上のすべてのエージェントの適切なピンを接続しなければならない。A[49:27]# 信号は、AP1# パリティ信号によってパリティ保護される。A[26:3]# 信号は、APO# パリティ信号によってパリティ保護される。

RESET# のアクティブから非アクティブへの移行時に、プロセッサは、A[49:3]# ピンをサンプリングして、電源投入時の設定を決定する。

A.1.2 A20M# (I)

Itanium 2 プロセッサのシステム環境では、A20M# は無視される。

A.1.3 ADS# (I/O)

アドレス・ストローブ (ADS#) 信号のアサートは、A[49:3]#、REQ[5:0]#、AP[1:0]#、RP# の各ピン上のトランザクション・アドレスの有効性を示す。すべてのバス・エージェントは、ADS# のアクティブ化を観察すると、新しいトランザクションに関連するパリティ・チェック、プロトコル・チェック、アドレスのデコード、内部スヌープ、または据え置き応答 ID のマッチング処理を開始する。

A.1.4 AP[1:0]# (I/O)

アドレス・パリティ (AP[1:0]#) 信号は、ADS# および A[49:3]# と一緒に、要求を開始するエージェントによってドライブされる。AP[1]# は A[49:27]# を保護し、AP[0]# は A[26:3]# を保護する。保護される信号のうち偶数個がローの場合は、正常なパリティ信号はハイである。保護される信号のうち奇数個がローの場合は、正常なパリティ信号はローである。この定義により、すべての保護される信号がハイである場合は、パリティはハイになる。

A.1.5 ASZ[1:0]# (I/O)

ASZ[1:0]# 信号は、メモリ・アドレス空間サイズ信号である。これらの信号は、REQa[4:3]# ピン上の要求フェーズの第 1 クロックで、要求を開始するエージェントによってドライブされる。ASZ[1:0]# 信号は、REQa[2:1]# 信号が 01B、10B、または 11B に等しい場合にのみ有効である（この値は、メモリ・アクセス・トランザクションを示す）。ASZ[1:0]# 信号のデコードは、表 A-1 に定義されている。

表 A-1. アドレス空間のサイズ

ASZ[1:0]#		メモリ・アドレス空間	メモリ・アクセス範囲
0	0	予約済み	予約済み
0	1	36 ビット	0 ~ (64G バイト - 1)
1	0	50 ビット	64G バイト ~ (1P バイト - 1)
1	1	予約済み	予約済み

64GB より小さいメモリ領域にアクセスするメモリ・トランザクション (すなわち、Aa[49:36]# がすべて 0) は、ASZ[1:0]# を 01 に設定しなければならない。64GB に等しいか、64GB より大きいメモリ領域にアクセスするメモリ・トランザクション (すなわち、Aa[49:36]# の中に 0 でない値がある) は、ASZ[1:0]# を 10 に設定しなければならない。64G バイト (36 ビット) アドレス空間をサポートするすべての観察側バス・エージェントは、ASZ[1:0]# が 01 に等しいときは、そのトランザクションに回答しなければならない。64G バイト (36 ビット) より大きなサイズのアドレス空間をサポートするすべての観察側バス・エージェントは、ASZ[1:0]# が 01 または 10 に等しいときは、そのトランザクションに回答しなければならない。

A.1.6 ATTR[3:0]# (I/O)

ATTR[3:0]# 信号は、属性信号である。これらの信号は、Ab[35:32]# ピン上の要求フェーズの第 2 クロックで、要求を開始するエージェントによってドライブされる。ATTR[3:0]# 信号は、すべてのトランザクションで有効である。ATTR[3]# 信号は予約済みである。ATTR[2:0]# は、メモリ・タイプに基づいてドライブされる。表 A-2 を参照のこと。

表 A-2. 有効なメモリ・タイプの信号エンコード

ATTR[2:0]#	説明
000	キャッシュ不可
100	ライト・コアレーシング
101	ライトスルー
110	ライトプロテクト
111	ライトバック

A.1.7 BCLKp/BCLKn (I)

BCLKp および BCLKn 差分クロック信号は、バス周波数を指定する。すべてのエージェントは、コモン・クロック・ラッチ・プロトコルを使用する信号については、BCLKp と BCLKn の差分交差で、出力をドライブし、入力をラッチする。

BCLKp と BCLKn は、Itanium 2 プロセッサの内部クロック周波数を間接的に決定する。それぞれの Itanium 2 プロセッサは、BCLKp と BCLKn の周波数に、電源投入時の設定で定義され、許容される比を掛けて、自分の内部クロック周波数を計算する。

A.1.8 BE[7:0]# (I/O)

BE[7:0]# 信号は、不完全なトランザクションのバイト・イネーブル信号である。これらの信号は、Ab[15:8]# ピン上の要求フェーズの第 2 クロックで、要求を開始するエージェントによってドライブされる。

メモリ・トランザクションまたは I/O トランザクションの場合、バイト・イネーブル信号は、128 ビット・データ・バスの対応するバイト上に有効なデータが要求されているか、転送されていることを示す。BE[0]# は最下位バイトが有効であることを示し、BE[7]# は最上位バイトが有効であることを示す。BE[7:0]# は、幅 16 バイトのバスのうち 8 バイトの有効性だけを指定する。したがって、A[3]# によって、BE[7:0]# がデータバスの上位半分と下位半分のどちらを有効にしているのかを判断する。

特殊なトランザクションの場合 ((REQa[5:0]# = 001000B) および (REQb[1:0]# = 01B))、BE[7:0]# 信号は、表 A-3 に定義されている特殊なサイクル・エンコードを伝達する。その他のすべてのエンコードは予約済みである。

表 A-3. バイト・イネーブル上の特殊なトランザクションのエンコード

特殊なトランザクション	バイト・イネーブル [7:0]#
NOP	0000 0000
シャットダウン	0000 0001
フラッシュ (INVD)	0000 0010
ホルト	0000 0011
同期 (WBINVD)	0000 0100
予約済み	0000 0101
ストップグラント確認	0000 0110
予約済み	0000 0111
xTPR アップデート	0000 1000

据え置き応答トランザクションの場合、BE[7:0]# 信号は予約済みである。据え置きフェーズの転送サイズは、BIL (Bus Invalidate Line) トランザクションを除いて、要求フェーズで指定されたサイズと常に同じになる。

BIL トランザクションは、1 つのキャッシュ・ライン (128 バイト) を返せる。

A.1.9 BERR# (I/O)

バス・エラー (BERR#) 信号をアサートして、グローバル MCA による回復可能なエラーを示すことができる。BERR# のアサート条件は、システム・レベルで設定可能である。設定オプションにより、次の方法で BERR# をドライブできる。

- バス・トランザクションを要求するエージェントが、内部エラーを観察した後にアサートする。
- 任意のバス・エージェントが、バス・トランザクション中にエラーを観察したときにドライブする。

BERR# のサンプリングが有効になっている場合、バス・エージェントが BERR# 信号のアサートをサンプリングすると、プロセッサはマシン・チェック・ハンドラに移行する。

BERR# はワイヤード OR 信号であるため、複数のバス・エージェントが同時にドライブできる。

A.1.10 BINIT# (I/O)

設定によって有効にされている場合、バス初期化 (BINIT#) 信号をアサートして、将来の正常な動作の妨げになるバス状態を報告できる。

電源投入時の設定で BINIT# の観察が有効になっている場合は、BINIT# のアサートがサンプリングされると、すべてのバス・ステート・マシンはリセットされる。すべてのエージェントは、バス・アービトレーション用のローテート ID をリセット後と同じ状態にリセットする。内部カウント情報は失われる。L2 キャッシュと L3 キャッシュは影響を受けない。

電源投入時の設定で BINIT# の観察が無効になっている場合は、プライオリティ・エージェントを除くすべてのエージェントは BINIT# を無視する。プライオリティ・エージェントは、システム・アーキテクチャに適合する方法でエラーを処理しなければならない。

BINIT# は、ワイヤード OR 信号である。

A.1.11 BNR# (I/O)

新しいバス・トランザクションを受け入れられないバス・エージェントは、次の要求のブロック (BNR#) 信号を使用してバス・ストールをアサートし、内部トランザクション・キューのオーバーフローを避けることができる。バスのストール中は、現在のバス・オーナは新しいトランザクションを発行できない。

複数のエージェントが同時にバス・ストールを要求する場合があるため、BNR# はワイヤード OR 信号である。複数のドライブ側エージェントが同時にエッジ遷移をドライブしたために発生するワイヤード OR グリッチを避けるために、BNR# は特定のクロックエッジでアサートされ、サンプリングされる。

A.1.12 BPM[5:0]# (I/O)

BPM[5:0]# 信号は、ブレイクポイントの挿入とパフォーマンスの監視に使用される、システム・サポート信号である。これらの信号は、パフォーマンスの監視に使用されるプログラマブル・カウンタを示すプロセッサからの出力として設定することも、ブレイクポイントのステータスを示すプロセッサへの入力としても設定できる。

A.1.13 BPRI# (I)

プライオリティ・エージェントは、バス・プライオリティ・エージェント要求 (BPRI#) 信号を使用して、システムバスの所有権を要求する。BPRI# のアサートが観察されると、プライオリティ・エージェント以外のすべてのエージェントは、(進行中のロックされた操作の一部になっている要求を除いて) 新しい要求の発行を停止する。プライオリティ・エージェントは、自分の要求がすべて完了するまで BPRI# をアサートし続け、その後 BPRI# をデアサートしてバスを解放する。

A.1.14 BR[0]# (I/O) および BR[3:1]# (I)

BR[3:0]# は、システム内で BREQ[3:0]# 信号をドライブする物理的バス要求ピンである。

BREQ[3:0]# 信号は、個々のプロセッサ・ピンに対してローテート方式で相互接続される。表 A-4 と表 A-5 は、4P システムバス構成と 2P システムバス構成について、プロセッサとバス信号の間のローテート方式の相互接続を示している。

表 A-4. BR0# (I/O)、BR1#、BR2#、BR3# 信号のローテート方式の相互接続 (4P の場合)

バス信号	エージェント0のピン	エージェント1のピン	エージェント2のピン	エージェント3のピン
BREQ[0]#	BR[0]#	BR[3]#	BR[2]#	BR[1]#
BREQ[1]#	BR[1]#	BR[0]#	BR[3]#	BR[2]#
BREQ[2]#	BR[2]#	BR[1]#	BR[0]#	BR[3]#
BREQ[3]#	BR[3]#	BR[2]#	BR[1]#	BR[0]#

表 A-5. BR0# (I/O)、BR1#、BR2#、BR3# 信号のローテート方式の相互接続 (2P の場合)

バス信号	エージェント0のピン	エージェント3のピン
BREQ[0]#	BR[0]#	BR[1]#
BREQ[1]#	BR[1]#	BR[0]#
BREQ[2]#	未使用	未使用
BREQ[3]#	未使用	未使用

電源投入時の設定中に、プライオリティ・エージェントは、BR[0]# バス信号をアサートしなければならない。すべての対称エージェントは、RESET# のアサートからデアサートへの移行時に、自分の BR[3:0]# ピンをサンプリングする。エージェントがどのピン上でアサート・レベルをサンプリングしたかによって、そのエージェントのエージェント ID が決まる。次に、すべてのエージェントは、表 A-6 に示すように、適切なバス信号プロトコルに一致するようにピンを構成する。

表 A-6. BR[3:0]# 信号とエージェント ID

RESET# でアサートがサンプリングされるピン	アービトレーション ID	報告されるエージェント ID
BR[0]#	0	0
BR[3]#	1	2
BR[2]#	2	4
BR[1]#	3	6

A.1.15 BREQ[3:0]# (I/O)

BREQ[3:0]# 信号は、対称エージェント・アービトレーション・バス信号 (バス要求と呼ばれる) である。対称エージェント n は、BREQ n # 信号をアサートして、バスの獲得を要求する。エージェント n は、BREQ n # を出力としてドライブし、それ以外の BREQ[3:0]# 信号を入力として受信する。

対称エージェントは、ラウンドロビン機構に基づく分散型アービトレーションをサポートしている。すべての対称エージェントは、ローテート ID を内部ステートとして使用し、次のアービトレーション・イベントで最低の優先度になるエージェントを監視する。電源投入時に、ローテート ID は 3 に初期化され、エージェント 0 が最高の優先度を持つ対称エージェントになる。新しいアービトレーション・イベントの後、すべての対称エージェントのローテート ID は、シンメトリック・オーナー (バスのオーナーである対称エージェント) のエージェント ID に合わせて更新される。この更新により、新しいシンメトリック・オーナーは、次のアービトレーション・イベントで最低の優先度を与えられる。

対称エージェントが (すべての BREQ[3:0]# がデアサートされている) アイドル・バス上で BREQ n # をアサートしたとき、現在のシンメトリック・オーナーが BREQ n # をデアサートして、新しいバス・オーナー n にバスの所有権を解放したときに、新しいアービトレーション・イベントが発生する。新しいアービトレーション・イベントでは、すべての対称エージェントは、BREQ[3:0]# とローテート ID を使用して、新しいシンメトリック・オーナーを同時に決定する。シ

シンメトリック・オーナは、他の対称エージェントがバスの使用を要求していなければ、バス上にパーク（バスを保留）できる。シンメトリック・オーナは、BREQ_n# 信号をアサートし続けることで、パークを実行する。シンメトリック・オーナは、他の対称エージェントがアサートした BREQ_n# をサンプリングすると、できるだけ早く BREQ_n# をデアサートして、バスを解放する。シンメトリック・オーナは、BPRI# のアサートを観察すると、（既存のロックされた操作の一部になっている要求を除いて）新しい要求の発行を停止する。

対称エージェントは、シンメトリック・オーナになる前に、BREQ_n# をデアサートできる。対称エージェントは、BREQ_n# を 1 クロックの間デアサートした後、BREQ_n# を再びアサートできる。

A.1.16 CCL# (I/O)

CCL# は、キャッシュ洗浄信号である。この信号は、EXF[2]#/Ab[5]# ピン上の要求フェーズの第 2 クロックでドライブされる。CCL# は、メモリ書き込みトランザクションについてアサートされ、プロセッサ内の修正済みラインを、キャッシュ内で無効にされることなく、メモリに書き込めることを示す。

A.1.17 CPUPRES# (O)

CPUPRES# を使用して、ソケット内の Itanium 2 プロセッサの有無を検出できる。グラウンド・レベルは、Itanium 2 プロセッサがインストールされていることを示す。オープン・レベルは、Itanium 2 プロセッサがインストールされていないことを示す。

A.1.18 D[127:0]# (I/O)

データ (D[127:0]#) 信号は、各種のシステム・バス・エージェント間の 128 ビット・データ・バスを提供する。不完全な転送には、1 データ転送クロックが必要である（アサートされたバイト・イネーブル BE[7:0]# および A[3]# によって指定されるバイト上で、有効なデータが転送される）。特定の転送で有効でないデータ信号でも、適切な ECC を持っていなければならない（データ・バス・エラー・チェックが有効になっている場合）。データをドライブするエージェントが DRDY# をアサートして、有効なデータ転送を示す。

A.1.19 D/C# (I/O)

データ / コード (D/C#) 信号を使用して、REQa[1]# 上の情報がデータ (1) かコード (0) かを指定する。この信号は、メモリ読み出しトランザクション中にのみ使用される。

A.1.20 DBSY# (I/O)

データ・バス・ビジー (DBSY#) 信号は、システムバス上にデータをドライブする役割を受け持つエージェントによってアサートされ、データバスが使用中であることを示す。データバスは、DBSY# がデアサートされた後に解放される。

システム・エージェント内のデータ・バスのパーティショニングを可能にするために、DBSY# は 3 回複製される。データ・バス・ビジー信号のこのコピー (DBSY#) は、入力と出力の両方に使用される。

A.1.21 DBSY_C1# (O)

DBSY# は、データ・バス・ビジー信号のコピーである。データ・バス・ビジー信号のこのコピー (DBSY_C1#) は、出力専用である。

A.1.22 DBSY_C2# (O)

DBSY# は、データ・バス・ビジー信号のコピーである。データ・バス・ビジー信号のこのコピー (DBSY_C2#) は、出力専用である。

A.1.23 DEFER# (I)

エージェントは、DEFER# 信号をアサートして、トランザクションのインオーダーの完了を保証できないことを示す。DEFER# をアサートするのは、通常はプライオリティ・エージェントの役割である。

A.1.24 DEN# (I/O)

据え置きイネーブル (DEN#) 信号は、Ab[4]# ピン上の要求フェーズの第 2 クロックでバス上にドライブされる。DEN# は、応答側エージェントによってアサートされ、トランザクションの据え置きが可能なことを示す。

A.1.25 DEP[15:0]# (I/O)

データバス ECC 保護 (DEP[15:0]#) 信号は、データバス (D[127:0]#) に対するオプションの ECC 保護を提供する。これらの信号は、D[127:0]# をドライブする役割を受け持つエージェントによってドライブされる。電源投入時の設定で、バス・エージェントを、ECC チェック有効またはチェックなしに設定できる。

ECC エラー修正コードは、シングルビット・エラーの検出および修正と、ダブルビット・エラーまたはニブル・エラーの検出が可能である。ECC の詳細は、第 4 章「データの健全性」を参照のこと。

A.1.26 DHIT# (I)

据え置きヒット (DHIT#) 信号は、据え置きエージェントによって、据え置きフェーズでドライブされる。バス上の読み出しトランザクションの場合、DHIT# は、最終的なキャッシュ・ステータスを返す (据え置かれていないトランザクションであれば、このステータスは HIT# で表示されずである)。DID[9:0]# (I/O)

DID[9:0]# は、据え置き識別子信号である。要求側エージェントは、A[25:16]# を使用して、これらの信号を転送する。これらの信号は、すべてのトランザクションの要求フェーズの第 2 クロックで、Ab[25:16]# 上に転送される。ただし、Ab[20:16]# は、据え置き可能なトランザクション (DEN# がアサートされている) についてのみ定義されている。また、DID[9:0]# は、据え置き応答トランザクションの要求フェーズの第 1 クロックで、Aa[25:16]# 上に転送される。

据え置き識別子は、要求側エージェントが供給するトークンを定義する。DID[9]# と DID[8:5]# は、要求側エージェントのエージェント識別子を伝達する (常に有効)。DID[4:0]# は、要求に対応するトランザクション識別子を伝達する (DEN# がアサートされている場合にのみ有効)。この構成のバス仕様では、論理バス・エージェントの数は 32 個までに制限され、それぞれのバス・エージェントが最大 32 個の要求を生成できる。表 A-7 は、DID のエンコードを示している。

表 A-7. DID[9:0]# のエンコード

DID[9]#	DID[8:5]#	DID[4:0]#
エージェントのタイプ	エージェント ID[3:0]	トランザクション ID[4:0]

DID[9]# は、エージェントのタイプを示す。対称エージェントは 0 を使用する。プライオリティ・エージェントは 1 を使用する。DID[8:5]# は、エージェント ID を示す。対称エージェントは、自分のアービトレーション ID を使用する。DID[4:0]# は、エージェントのトランザクション ID を示す。トランザクション ID は、スヌープの結果をまだ報告していないエージェントによって発行された、すべての据え置き可能なトランザクションごとに固有でなければならない。

据え置き応答エージェントは、元のトランザクション中に受信した DID[9:0]# (Ab[25:16]#) 信号を、据え置き応答トランザクション中に Aa[25:16]# 信号上に送信する。このプロセスにより、元の要求側エージェントは、完了待ち状態の元の要求との間で識別子を一致させることができる。

A.1.27 DPS# (I/O)

据え置きフェーズ・イネーブル (DPS#) 信号は、Ab[3]# ピン上の要求フェーズの第2クロックで、バスに対してドライブされる。DPS# は、要求側エージェントが、据え置きフェーズを使用するトランザクションの完了をサポートしている場合にアサートされる。据え置きフェーズをサポートする要求側エージェントは、常に DPS# をアサートする。据え置きフェーズをサポートしない要求側エージェントは、常に DPS# をデアサートする。

A.1.28 DRDY# (I/O)

データ・レディ (DRDY#) 信号は、各データ転送でデータをドライブするエージェントによってアサートされ、データバス上の有効なデータを示す。マルチサイクル・データ転送では、DRDY# をデアサートして、アイドル・クロックを挿入できる。

DRDY# は、システム・エージェント内のデータ・バスのパーティショニングを可能にするために、3 回複製される。データ・レディ信号のこのコピー (DRDY#) は、入力と出力の両方に使用される。

A.1.29 DRDY_C1# (O)

DRDY# は、データ・レディ信号のコピーである。データ・フェーズのデータ・レディ信号のこのコピー (DRDY_C1#) は、出力専用である。

A.1.30 DRDY_C2# (O)

DRDY# は、データ・レディ信号のコピーである。データ・フェーズのデータ・レディ信号のこのコピー (DRDY_C2#) は、出力専用である。

A.1.31 DSZ[1:0]# (I/O)

データ・サイズ (DSZ[1:0]#) 信号は、要求フェーズの第2クロックで、要求側エージェントによって REQb[4:3]# 信号上に転送される。DSZ[1:0]# 信号は、要求側エージェントのデータ転送能力を定義する。Itanium 2 プロセッサでは、常に DSZ# = 01 である。

A.1.32 EXF[4:0]# (I/O)

拡張機能 (EXF[4:0]#) 信号は、要求フェーズの第2クロックで、要求側エージェントによって A[7:3]# ピン上に転送される。この信号は、要求側エージェントのモードまたは能力に基づいて、トランザクションに必要な特殊機能を指定する。この信号の定義を表 A-8 に示す。

表 A-8. 拡張機能信号

拡張機能信号	信号名エイリアス	機能
EXF[4]#	予約済み	予約済み
EXF[3]#	SPLCK#/FCL#	スプリット・ロック / フラッシュ・キャッシュ・ライン
EXF[2]#	OWN#/CCL#	メモリ更新不要 / キャッシュ洗浄
EXF[1]#	DEN#	据え置きイネーブル
EXF[0]#	DPS#	据え置きフェーズのサポート

A.1.33 FCL# (I/O)

フラッシュ・キャッシュ・ライン (FCL#) 信号は、A[6]# ピン上の要求フェーズの第 2 クロックで、バスに対してドライブされる。FCL# のアサートは、メモリ・トランザクションがグローバルなフラッシュ・キャッシュ (FC) 命令による開始を示す。

A.1.34 FERR# (O)

FERR# 信号のアサートは、IA-32 アプリケーションによって、マスクされていない浮動小数点エラーの発生を示す。

A.1.35 GSEQ# (I)

シーケンシャリティ保証 (GSEQ#) 信号のアサートは、プラットフォームが、シーケンシャルな性質を失わずに再試行なしのトランザクションの完了を保証していることを示す。

A.1.36 HIT# (I/O) および HITM# (I/O)

スヌープ・ヒット (HIT#) 信号とヒット・モディファイド (HITM#) 信号は、トランザクション・スヌープ操作の結果を伝達する。任意のバス・エージェントは、HIT# と HITM# を一緒にアサートして、スヌープ・ストールを要求できる。ストールを続けるには、HIT# と HITM# を一緒に再アサートする。

A.1.37 ID[9:0]# (I)

トランザクション ID (ID[9:0]#) 信号は、据え置きエージェントによってドライブされる。2 クロックのトランザクション ID 信号は、IDa[9:0]# および IDb[9:0]# として参照される。ID[9:0]# 信号は、2 クロックのうち第 1 クロックでは IP0# パリティ信号によって保護され、第 2 クロックでは IP[1]# パリティ信号によって保護される。

IDa[9:0]# は、Ab[25:16]# 上に送信された、据え置きトランザクションの ID (DID[9:0]#) を返す。

A.1.38 IDS# (I)

ID ストローブ (IDS#) 信号のアサートは、そのクロックの ID[9:0]# の有効性と、次のクロックの DHIT# および IP[1:0]# の有効性を示す。

A.1.39 IGNNE# (I)

Itanium 2 プロセッサのシステム環境では、IGNNE# は無視される。

A.1.40 INIT# (I)

初期化 (INIT#) 信号は、プロセッサに対するマスクされていない割り込みをトリガする。INIT# は、通常は、ハング状態またはアイドル状態のプロセッサへの割り込みに使用される。プラットフォームの互換性を保証するために必要なセマンティクスは、PAL ファームウェアの割り込みサービス・ルーチン内で提供される。

A.1.41 INT (I)

INT は、外部割り込みが発生したことを示す 8259 互換割り込み要求信号である。この割り込みはマスク可能である。プロセッサは、現在の命令の実行の完了後、処理を割り込みハンドラに渡す。プロセッサは、割り込み確認トランザクションを生成して、割り込みコントローラから割り込みベクタを取得する。

LINT[0] ピンは、INT 信号として使用するか、他のローカル割り込みとして使用するかを、ソフトウェアによって設定できる。

A.1.42 IP[1:0]# (I)

ID パリティ (IP[1:0]#) 信号は、据え置きフェーズの第 2 クロックで、据え置きエージェントによってドライブされる。IP0# は、第 1 クロックで IDa[9:0]# 信号と IDS# 信号を保護する。IP[1]# は、第 2 クロックで IDb[9:2, 0]# 信号と IDS# 信号を保護する。

A.1.43 LEN[2:0]# (I/O)

データ長 (LEN[2:0]#) 信号は、要求フェーズの第 2 クロックで、要求側エージェントによって REQb[2:0]# 信号を使用して転送される。LEN[2:0]# は、表 A-9 に示すように、要求側エージェントによって要求されるデータ転送のサイズを定義する。LEN[2:0]#、HITM#、RS[2:0]# 信号を組み合わせ、実際のデータ転送のサイズを定義する。

表 A-9. データ転送のサイズ

LEN[2:0]#	サイズ
000	0 ~ 8 バイト
001	16 バイト
010	32 バイト
011	64 バイト
100	128 バイト
101	予約済み
110	予約済み
111	予約済み

A.1.44 LINT[1:0] (I)

LINT[1:0] は、ローカル割り込み信号である。これらのピンは、RESET# の後は無効になる。LINT[0] は、通常は、ソフトウェアによって INT (8259 互換のマスク可能な割り込み要求信号) として設定される。LINT[1] は、通常は、ソフトウェアによって NMI (マスク不可割り込み) として設定される。いずれの信号も、非同期入力である。

A.1.45 LOCK# (I/O)

Itanium 2 プロセッサのシステム環境では、LOCK# はアサートもサンプリングもされない。

A.1.46 NMI (I)

NMI 信号は、マスク不可割り込み信号である。NMI をアサートすると、内部で与えられたベクタ値 2 を持つ割り込みが発生する。外部割り込み確認トランザクションは発生しない。NMI サービス・ルーチンの実行中に NMI がアサートされた場合は、その NMI は保留され、NMI サービス・ルーチンが EOI を実行した後で認識される。アサートされた NMI は、最大 1 つまで保留できる。

NMI は、立ち上がりエッジに依存する。NMI が同期してアサートされ、セットアップ時間とホールド時間の条件を満たしている場合は、特定のクロックでの NMI の認識は保証される。NMI が非同期でアサートされた場合は、NMI のアサートされるパルス幅とデアサートされるパルス幅は、少なくとも 2 クロック必要である。この信号は、NMI または他のローカル割り込み (LINT1 ピン) として使用されるように、ソフトウェアによって設定されていなければならない。

A.1.47 OWN# (I/O)

保証されたキャッシュ・ライン所有権 (OWN#) 信号は、Ab[5]# ピン上の要求フェーズの第 2 クロックで、バスに対してドライブされる。キャッシュ・ラインの所有権が保証されている場合は、OWN# がアサートされる。これにより、メモリ・コントローラは、暗黙的なライトバックによるメモリの更新を無視できる。

A.1.48 PMI# (I)

プラットフォーム管理割り込み (PMI#) 信号は、プロセッサに対して最高の優先順位を持つ割り込みをトリガする。PMI# は、通常はシステムによって使用され、プラットフォーム固有のファームウェアによって処理されるシステム・イベントをトリガする。

A.1.49 PWRGOOD (I)

電源正常 (PWRGOOD) 信号は、電源投入時にデアサート (L) され、システムによって RESET# が最初にアサートされた後にアサート (H) されなければならない。

A.1.50 REQ[5:0]# (I/O)

REQ[5:0]# は、要求コマンド信号である。これらの信号は、要求フェーズの第 1 クロックと第 2 クロックの両方で、現在のバス・オーナーによってアサートされる。第 1 クロックでは、REQa[5:0]# 信号が、スヌープ要求を開始するのに十分な詳細レベルに合わせてトランザクションのタイプを定義する。第 2 クロックでは、REQb[5:0]# 信号が、トランザクションのタイプを漏れなく定義するための追加情報を伝達する。REQb[4:3]# 信号は、DSZ[1:0]# か、データ転送に関連するトランザクションの要求側エージェントのデータ転送速度の情報を送信する。REQb[2:0]# 信号は、LEN[2:0]# (データ転送サイズの情報) を送信する。第 1 クロックと第 2 クロックで、REQ[5:0]# と ADS# は、パリティ RP# によって保護される。

すべての受信側エージェントは、REQ[5:0]# 信号を観察して、表 A-10 に示すようにトランザクションのタイプを判定し、必要に応じてトランザクションに参加する。

表 A-10. REQa#/REQb# 信号によって定義されるトランザクションのタイプ

トランザクション	REQa[5:0]#						REQb[5:0]#					
	5	4	3	2	1	0	5	4	3	2	1	0
据え置き応答	0	0	0	0	0	0	0	x	x	x	x	x
予約済み	0	0	0	0	0	1	0	x	x	x	x	x
割り込み確認	0	0	1	0	0	0	0	DSZ[1:0]#	0	0	0	0
特殊なトランザクション	0	0	1	0	0	0	0	DSZ[1:0]#	0	0	0	1
予約済み	0	0	1	0	0	0	0	DSZ[1:0]#	0	1	x	x
予約済み	0	0	1	0	0	1	0	DSZ[1:0]#	0	x	x	x
割り込み	0	0	1	0	0	1	0	DSZ[1:0]#	1	0	0	0
ページ TC	0	0	1	0	0	1	0	DSZ[1:0]#	1	0	1	1
予約済み	0	0	1	0	0	1	0	DSZ[1:0]#	1	1	x	x
I/O 読み出し	0	1	0	0	0	0	0	DSZ[1:0]#	x	x	x	x
I/O 書き込み	0	1	0	0	0	1	0	DSZ[1:0]#	x	x	x	x
予約済み	0	1	1	0	0	x	0	DSZ[1:0]#	x	x	x	x
メモリ読み出しおよび無効化	0	ASZ[1:0]#	0	1	0	0	0	DSZ[1:0]#	LEN[2:0]#			
予約済み	0	ASZ[1:0]#	0	1	1	0	0	DSZ[1:0]#	LEN[2:0]#			
メモリ読み出し	0	ASZ[1:0]#	1	D/C#	0	0	0	DSZ[1:0]#	LEN[2:0]#			
メモリ読み出し (現在)	1	ASZ[1:0]#	1	0	0	0	0	DSZ[1:0]#	LEN[2:0]#			
予約済み	1	ASZ[1:0]#	1	1	0	0	0	DSZ[1:0]#	LEN[2:0]#			
メモリ書き込み	0	ASZ[1:0]#	1	WSNP#	1	0	0	DSZ[1:0]#	LEN[2:0]#			
キャッシュ・ライン置換	1	ASZ[1:0]#	1	WSNP#	1	0	0	DSZ[1:0]#	0	0	0	0

A.1.51 RESET# (I)

RESET# 信号をアサートすると、すべてのプロセッサが確認済みの状態にリセットされ、修正済み (M ステート) ラインを書き戻さずに、すべてのキャッシュが無効にされる。「ウォーム」リセットの場合は、RESET# を 1 マイクロ秒の間アサートしなければならない。電源投入リセットの場合は、V_{CC} と BCLK_p が適切な仕様に達した後、RESET# を少なくとも 1ms の間アサートしなければならない。すべてのシステム・バス・エージェントは、RESET# のアサートを観察してから 2 クロック以内に、自分の出力をデアサートしなければならない。

電源投入時の設定では、RESET# のアサートからデアサートへの移行時に、多くのバス信号がサンプリングされる。

電源投入時の設定でプロセッサの出力がトライステートになっていない限り、RESET# のアサートからデアサートへの移行後、プロセッサはリセット・ベクタからプログラムの実行を開始する。

A.1.52 RP# (I/O)

要求パリティ (RP#) 信号は、要求側エージェントによってドライブされ、ADS# と REQ[5:0]# のパリティ保護を提供する。

保護される信号のうち偶数個がローの場合は、正常なパリティ信号はハイである。保護される信号のうち奇数個がローの場合は、正常なパリティ信号はローである。この定義により、すべての保護される信号がハイの場合は、パリティはハイになる。

A.1.53 RS[2:0]# (I)

応答ステータス (RS[2:0]#) 信号は、応答側エージェント (トランザクションを完了する役割を受け持つエージェント) によってドライブされる。

A.1.54 RSP# (I)

応答パリティ (RSP#) 信号は、RS[2:0]# (RSP# がパリティ保護を与える信号) のアサート中に、応答側エージェント (現在のトランザクションを完了する役割を受け持つエージェント) によってドライブされる。

保護される信号のうち偶数個がローの場合は、正常なパリティ信号はハイである。保護される信号のうち奇数個がローの場合は、正常なパリティ信号はローである。RS[2:0]# がアイドル状態 (RS[2:0]#=000) の間は、正常なパリティを保证するエージェントが RSP# をドライブしないため、RSP# もハイになる。

A.1.55 SBSY# (I/O)

ストローブ・バス・ビジー (SBSY#) 信号は、データを転送するエージェントがストローブ・バスを所有しているとき、データを転送するエージェントによってドライブされる。SBSY# は、最初の DRDY# の前と (マルチクロック・データ転送の場合は) DRDY# の各アサートの間、ストローブ・バスを保留する。SBSY# は、DBSY# より前にデアサートされる。これにより、次のデータ転送エージェントは、データバスが解放される前に、ストローブ信号を事前にドライブできる。

システム・エージェント内のデータ・バスのパーティショニングを可能にするために、SBSY# は 3 回複製される。ストローブ・バス・ビジー信号のこのコピー (SBSY#) は、入力と出力の両方に使用される。

A.1.56 SBSY_C1# (O)

SBSY# は、ストローブ・バス・ビジー信号のコピーである。ストローブ・バス・ビジー信号のこのコピー (SBSY_C1#) は、出力専用である。

A.1.57 SBSY_C2# (O)

SBSY# は、ストローブ・バス・ビジー信号のコピーである。ストローブ・バス・ビジー信号のこのコピー (SBSY_C2#) は、出力専用である。

A.1.58 SPLCK# (I/O)

スプリット・ロック (SPLCK#) 信号は、ロックされた操作の最初のトランザクションで、Ab[6]# ピン上の要求フェーズの第 2 クロックでドライブされる。この信号のドライブは、ロックされた操作が 4 つのロックされたトランザクションで構成されることを示す。

A.1.59 STBn[7:0]# および STBp[7:0]# (I/O)

STBp[7:0]# と STBn[7:0]# (および DRDY#) を BCLKp の代わりに使用して、2 倍の転送速度でデータを転送できる。これらの信号は、対応するバス信号に対して密な傾斜関係を持つデータ転送エージェントによってドライブされる。受信側エージェントは、これらの信号を使用して、有効なデータをラッチ内にキャプチャする。この機能は、STBp[7:0]# または STBn[7:0]# の立ち下がりエッジで構成される、独立した 2 倍周波数クロックに似ている。転送データは、DRDY# によって同期化される。表 A-11 に示すように、ストローブ信号の各ペアは、16 のデータバス信号と 2 つの ECC 信号に対応付けられる。

表 A-11. STBp[7:0]# および STBn[7:0]# の対応関係

ストローブ信号のビット	データ信号のビット	ECC 信号のビット
STBp[7]#, STBn[7]#	D[127:112]#	DEP[15:14]#
STBp[6]#, STBn[6]#	D[111:96]#	DEP[13:12]#
STBp[5]#, STBn[5]#	D[95:80]#	DEP[11:10]#
STBp[4]#, STBn[4]#	D[79:64]#	DEP[9:8]#
STBp[3]#, STBn[3]#	D[63:48]#	DEP[7:6]#
STBp[2]#, STBn[2]#	D[47:32]#	DEP[5:4]#
STBp[1]#, STBn[1]#	D[31:16]#	DEP[3:2]#
STBp[0]#, STBn[0]#	D[15:0]#	DEP[1:0]#

A.1.60 TCK (I)

テスト・クロック (TCK) 信号は、IEEE 1149.1 に準拠するテスト・アクセス・ポート (TAP) 用のクロック入力を提供する。

A.1.61 TDI (I)

テスト・データ・イン (TDI) 信号は、Itanium 2 プロセッサ内にシリアル・テスト・データを転送する。TDI 信号は、IEEE 1149.1 に準拠するテスト・アクセス・ポート (TAP) に必要なシリアル入力を提供する。

A.1.62 TDO (O)

テスト・データ・アウト (TDO) 信号は、Itanium 2 プロセッサからシリアル・テスト・データを転送する。TDO 信号は、IEEE 1149.1 に準拠するテスト・アクセス・ポート (TAP) に必要なシリアル出力を提供する。

A.1.63 THRMTRIP# (O)

温度トリップ (THRMTRIP#) 信号は、内部温度センサの使用により、Itanium 2 プロセッサを致命的な加熱状態から保護する。このセンサは、偽りのトリップが発生しないように、通常の動作温度より十分に高い温度に設定される。プロセッサが温度トリップに移行すると、データは失われる (この状態は、THRMTRIP# 信号のアサートによってシステムに通知される)。THRMTRIP# がアサートされた場合は、プラットフォームは RESET# をアサートして、プロセッサの物理的状态を保護しなければならない。

A.1.64 THRMALERT# (O)

THRMALERT# は、プロセッサの温度ダイオードから得られる測定温度が、センサの高温 (THIGH) または低温 (TLOW) レジスタ内に設定された温度しきい値データに達するか、それを超えたときにアサートされる。プラットフォームは、この信号を使用して、温度調整機能を実行する。

A.1.65 TMS (I)

テスト・モード選択 (TMS) 信号は、IEEE 1149.1 に準拠するテスト・アクセス・ポート (TAP) 仕様のサポート信号であり、デバッグ・ツールによって使用される。

A.1.66 TND# (I/O)

TLB パージ非実行 (TND#) 信号をアサートすると、システムバス上で TLB パージ・トランザクションが完了した後であっても、TLB パージ命令の完了を遅らせることができる。

A.1.67 TRDY# (I)

ターゲット・レディ (TRDY#) 信号は、ターゲット・エージェントによってアサートされる。この信号は、ターゲット・エージェントが、書き込みまたは暗黙的ライトバックのデータ転送を受け入れられることを示す。

A.1.68 TRST# (I)

TAP リセット (TRST#) 信号は、IEEE 1149.1 に準拠するテスト・アクセス・ポート (TAP) のサポート信号であり、デバッグ・ツールによって使用される。

A.1.69 WSNP# (I/O)

書き込みスヌープ (WSNP#) 信号は、スヌープ・エージェントがメモリ書き込みトランザクションをスヌープすることを示す。

A.2 信号のまとめ

表 A-12 ~ 表 A-15 は、Itanium 2 プロセッサの出力信号、入力信号、I/O 信号の属性をまとめたものである。

表 A-12. 出力信号

信号名	アクティブ・レベル	クロック	信号グループ
CPUPRES#	ロー	—	プラットフォーム
DBSY_C1#	ロー	BCLKp	データ
DBSY_C2#	ロー	BCLKp	データ
DRDY_C1#	ロー	BCLKp	データ
DRDY_C2#	ロー	BCLKp	データ
FERR#	ロー	非同期	PC 互換性
SBSY_C1#	ロー	BCLKp	データ
SBSY_C2#	ロー	BCLKp	データ
TDO	ハイ	TCK	TAP
THRMTRIP#	ロー	非同期	エラー
THRMALERT#	ロー	非同期	エラー

表 A-13. 入力信号

信号名	アクティブ・レベル	クロック	信号グループ	条件
BPRI#	ロー	BCLKp	アービトレーション	常時
BR1#	ロー	BCLKp	アービトレーション	常時
BR2#	ロー	BCLKp	アービトレーション	常時
BR3#	ロー	BCLKp	アービトレーション	常時
BCLKp	ハイ	—	制御	常時
BCLKn	ハイ	—	制御	常時
D/C#	ロー	BCLKp	システムバス	要求フェーズ(メモリ読み出し)
DEFER#	ロー	BCLKp	スヌープ	スヌープ・フェーズ
DHIT#	ロー	BCLKp	システムバス	IDS#+1
GSEQ#	ロー	BCLKp	スヌープ	スヌープ・フェーズ
ID[9:0]#	ロー	BCLKp	据え置き	IDS#, IDS#+1
IDS#	ロー	BCLKp	据え置き	常時
INIT#	ロー	非同期	実行制御	常時 ¹
INT (LINT0)	ハイ	非同期	実行制御	
IP[1:0]#	ロー	BCLKp	システムバス	IDS#+1
NMI (LINT1)	ハイ	非同期	実行制御	
RESET#	ロー	BCLKp	制御	常時
RS[2:0]#	ロー	BCLKp	応答	常時
RSP#	ロー	BCLKp	応答	常時
PMI#	ロー	非同期	実行制御	
PWRGOOD	ハイ	非同期	制御	—
TCK	ハイ	—	診断	常時
TDI	ハイ	TCK	診断	常時
TMS	ハイ	TCK	診断	常時
TRST#	ロー	非同期	診断	常時
TRDY#	ロー	BCLKp	応答	応答フェーズ

1. RS[2:0]#のアサートとの同期アサートにより、同期化を保證する。

表 A-14. 入力/出力信号 (シングル・ドライバ)

信号名	アクティブ・レベル	クロック	信号グループ	条件
A[49:3]#	□-	BCLKp	要求	ADS#、ADS#+1
ADS#	□-	BCLKp	要求	常時
AP[1:0]#	□-	BCLKp	要求	ADS#、ADS#+1
ASZ[1:0]#	□-	BCLKp	システムバス	ADS#
ATTR[3:0]#	□-	BCLKp	システムバス	ADS#+1
BE[7:0]#	□-	BCLKp	システムバス	ADS#+1
BR0#	□-	BCLKp	システムバス	常時
BPM[5:0]#	□-	BCLKp	診断	常時
CCL#	□-	BCLKp	システムバス	ADS#+1
D[127:0]#	□-	BCLKp	データ	DRDY#
DBSY#	□-	BCLKp	データ	常時
D/C#	□-	BCLKp	システムバス	ADS#
DEN#	□-	BCLKp	システムバス	ADS#+1
DEP[15:0]#	□-	BCLKp	システムバス	DRDY#
DID[9:0]#	□-	BCLKp	システムバス	ADS#+1
DRDY#	□-	BCLKp	データ	常時
DPS#	□-	BCLKp	システムバス	ADS#+1
DSZ[1:0]#	□-	BCLKp	システムバス	ADS#+1
EXF[4:0]#	□-	BCLKp	システムバス	ADS#+1
FCL#	□-	BCLKp	システムバス	ADS#+1
LEN[2:0]#	□-	BCLKp	システムバス	ADS#+1
LOCK#	□-	BCLKp	アービトレーション	常時
OWN#	□-	BCLKp	システムバス	ADS#+1
REQ[5:0]#	□-	BCLKp	要求	ADS#、ADS#+1
RP#	□-	BCLKp	要求	ADS#、ADS#+1
SBSY#	□-	BCLKp	データ	常時
SPLCK#	□-	BCLKp	システムバス	ADS#+1
STBn[7:0]#	□-	—	データ	常時
STBp[7:0]#	□-	—	データ	常時
WSNP#	□-	BCLKp	システムバス	ADS#

表 A-15. 入力/出力信号 (マルチ・ドライバ)

信号名	アクティブ・レベル	クロック	信号グループ	条件
BNR#	ロー	BCLKp	システムバス	常時
BERR#	ロー	BCLKp	エラー	常時
BINIT#	ロー	BCLKp	エラー	常時
HIT#	ロー	BCLKp	スヌープ	スヌープ・フェーズ
HITM#	ロー	BCLKp	スヌープ	スヌープ・フェーズ
TND#	ロー	BCLKp	スヌープ	常時

A

A[43:3]#	3-5, A-1
A20M#	A-1
ADS#	3-5, A-1
ALAT (Advanced Load Address Table)	2-9
AP[1:0]#	4-2, A-1
ASZ[1:0]#	A-1
ATTR[7:0]#	A-2

B

BCLK	3-4
BCLKN	3-4, A-2
BCLKP	3-4, A-2
BE[7:0]#	A-3
BERR#	3-8, 5-3, A-3
BINIT#	3-8, 5-3, A-4
BNR#	3-4, A-4
BPM[5:0]#	3-10, A-4
BPRI#	3-4, A-4
BR[3:1]#	A-4
BR0#	A-4
BREQ[3:0]#	3-4, A-5
BREQ0#	5-5, A-5
BYPASS	6-5

C

CPUPRES#	A-6
----------	-----

D

D/C#	A-6
D[63:0]#	A-6
DBSY#	3-7, A-6
DBSY# 信号	A-6
DEFER#	3-6, A-6
DEN#	3-8, A-7
DEP[7:0]#	3-7, A-7
DHIT#	A-7
DID[7:0]#	A-7
DPS#	A-8
DRDY#	3-7, A-8
DSZ[1:0]#	A-8

E

EXTTEST	6-5
---------	-----

F

FCL#	A-9
FERR#	A-9

G

GSEQ#	A-9
-------	-----

H

HIT#	A-9
HITM#	A-9

I

IA-32 互換性信号	3-9
ID[7:0]#	A-9
IDCODE	6-5
IDS#	A-9
IGNNE#	A-9
INIT#	3-9, 5-6, A-9
INT	A-10
IP[1:0]#	A-10

L

L2 キャッシュ	2-9
L3 キャッシュ	2-9
LEN[1:0]#	A-10
LINT[1:0]	3-9, A-10
LOCK#	3-4, A-10

O

OWN#	A-11
------	------

P

PAL (Processor Abstraction Layer)	1-1
PMI#	A-11
PWRGOOD	A-11

R

REQ[4:0]#	3-5, A-11
RESET#	5-6, A-12
RESET# 入力信号	3-4
RP#	3-5, 4-1, A-12
RS[2:0]#	4-2, 5-2, A-13
RSP#	3-6, 4-1, A-13

S

SBSY#	3-7, A-13
SPLCK#	A-13
STBN[3:0]#	3-7, A-14
STBP[3:0]#	3-7, A-14

T

TCK	A-14
TDI	A-14
TDO	A-14
THERMTRIP#	3-9, A-14
THRMALERT#	A-15
TMS	A-15
TND#	A-15
TRDY#	3-6, A-15
TRST#	A-15

W

WSNP#	A-15
-------------	------

あ

アービトレーション ID	5-3
アービトレーション信号	3-4
アドレス信号	3-5, A-1

い

インオーダー・キューのパイプライン化	5-3
--------------------------	-----

え

エラー検出	4-1
エラー修正コード (ECC)	3-7
エラー信号	3-8
エラーの分類	4-1
エラー・コード・アルゴリズム	4-3

お

応答信号	3-6
------------	-----

か

回復可能なエラー	4-1, 6-4
----------------	----------

く

グローバル・エラー	4-1, 6-4
クロック周波数	5-5

クロック比	5-5
-------------	-----

こ

互換性信号	3-9
-------------	-----

し

システムバス	1-1
アービトレーション信号	3-4
エラー信号	3-8
応答信号	3-6
信号伝送	3-1
据え置き信号	3-8
スヌープ信号	3-5
制御信号	3-4
ソース・シンクロナス・シグナリング	3-2
データ信号	3-7
要求信号	3-5
実行制御信号	3-9
初期化	5-6
信号のまとめ	A-15
診断信号	3-10

す

据え置きイネーブル信号	A-7
スヌープ信号	3-5

そ

ソース・シンクロナス	3-2
属性信号	A-2

た

対称エージェント・アービトレーション・バス信号	A-5
-------------------------------	-----

て

ディスペーサル・ロジック	2-5
データ応答信号	3-6
データ転送信号	4-2
データ・サイズ (DSZ) 信号	A-8
データ・バス・エラー・チェック	5-2
データ・バス・ビジー信号	A-6
データ信号	3-7
テスト・アクセス・ポート (TAP)	6-1
TCK	6-2
TDI	6-2
TDO	6-2
TMS	6-2

TRST#	6-2	れ	
命令	6-4	レジスタ・スタック・エンジン (RSE)	2-7
レジスタ	6-4		
と		ろ	
統合ツール	7-1	ローカル・エラー	4-1, 6-4
トランスレーション・ルックアサイド・バッファ (TLB)	2-10	わ	
は		ワイヤード OR グリッチ	3-2
バイト・イネーブル信号	A-3	割り込み要求信号	A-10
バウンダリ・スキャン・チェーン	6-1		
バス信号の保護	4-2		
パリティ・アルゴリズム	4-3		
ふ			
封じ込め可能なエラー	4-1, 6-4		
浮動小数点ユニット (FPU)	2-5		
プラットフォーム信号	3-10		
分岐予測	2-5		
ま			
マスク不可割り込み (NMI) 信号	A-11		
め			
命令バッファ	2-5		
命令フェッチ	2-4		
命令プリフェッチ	2-4		
命令レジスタ	6-4		
メモリ			
アドレス空間サイズ信号	A-1		
メモリ・サブシステム	2-8		
よ			
要求信号	3-5		
要求パリティ (RP#) 信号	3-5		
ら			
ラッチ・バス・プロトコル	3-1		
り			
リセット動作	6-5		

第 1 章	はじめに	1-1
	1.1 Itanium® 2 プロセッサのシステムバス	1-1
	1.2 PAL (Processor Abstraction Layer)	1-1
	1.3 用語	1-2
	1.4 参考資料	1-2
	1.4.1 改訂履歴	1-3
第 2 章	Itanium® 2 プロセッサのマイクロアーキテクチャ	2-1
	2.1 概要	2-1
	2.1.1 6 命令を同時に処理できる EPIC コア	2-1
	2.1.2 プロセッサ・パイプライン	2-2
	2.1.3 プロセッサのブロック・ダイアグラム	2-3
	2.2 命令処理	2-4
	2.2.1 命令プリフェッチと命令フェッチ	2-4
	2.2.2 分岐予測	2-5
	2.2.3 ディスパーサル・ロジック	2-5
	2.3 実行	2-5
	2.3.1 浮動小数点ユニット (FPU)	2-5
	2.3.2 整数ロジック	2-6
	2.3.3 レジスタ・ファイル	2-6
	2.3.4 レジスタ・スタック・エンジン (RSE)	2-7
	2.4 制御	2-8
	2.5 メモリ・サブシステム	2-8
	2.5.1 L1 命令キャッシュ	2-9
	2.5.2 L1 データ・キャッシュ	2-9
	2.5.3 ユニファイド L2 キャッシュ	2-9
	2.5.4 ユニファイド L3 キャッシュ	2-9
	2.5.5 ALAT (Advanced Load Address Table)	2-9
	2.5.6 トランスレーション・ルックアサイド・バッファ (TLB)	2-10
	2.5.7 キャッシュ・コヒーレンシ	2-10
	2.5.8 ライト・コアレッシング	2-10
	2.5.9 メモリの順序付け	2-11
	2.6 IA-32 実行	2-11
第 3 章	システムバスの概要	3-1
	3.1 Itanium® 2 プロセッサのシステムバスの信号伝送	3-1
	3.1.1 コモン・クロック・シグナリング	3-1
	3.1.2 ソース・シンクロナス・シグナリング	3-2
	3.2 信号の概要	3-3
	3.2.1 制御信号	3-4
	3.2.2 アービトレーション信号	3-4
	3.2.3 要求信号	3-5
	3.2.4 スヌープ信号	3-5
	3.2.5 応答信号	3-6
	3.2.6 データ信号	3-7
	3.2.7 据え置き信号	3-8
	3.2.8 エラー信号	3-8
	3.2.9 実行制御信号	3-9
	3.2.10 IA-32 互換性信号	3-9
	3.2.11 プラットフォーム信号	3-10
	3.2.12 診断信号	3-10

第 4 章	データの保全性	4-1
4.1	エラーの分類	4-1
4.2	Itanium [®] 2 プロセッサ・システム・バスのエラー検出	4-1
4.2.1	直接に保護されるバス信号	4-2
4.2.2	間接的に保護されるバス信号	4-2
4.2.3	保護されないバス信号	4-3
4.2.4	Itanium [®] 2 プロセッサ・システム・バスのエラー・コード・アルゴリズム ...	4-3
第 5 章	コンフィグレーションと初期化	5-1
5.1	設定の概要	5-1
5.2	設定機能	5-1
5.2.1	データ・バス・エラー・チェック	5-2
5.2.2	応答 /ID 信号パリティ・エラー・チェック	5-2
5.2.3	アドレス / 要求信号パリティ・エラー・チェック	5-3
5.2.4	イニシエータ・バス・エラーでの BERR# のアサート	5-3
5.2.5	ターゲット・バス・エラーでの BERR# のアサート	5-3
5.2.6	BERR# サンプリング	5-3
5.2.7	BINIT# エラーのアサート	5-3
5.2.8	BINIT# エラーのサンプリング	5-3
5.2.9	インオーダー・キューのパイプライン化	5-3
5.2.10	要求バス・パーキング有効	5-3
5.2.11	対称エージェントのアービトレーション ID	5-3
5.2.12	クロック周波数比	5-5
5.3	初期化の概要	5-6
5.3.1	RESET# による初期化	5-6
5.3.2	INIT による初期化	5-6
第 6 章	テスト・アクセス・ポート (TAP)	6-1
6.1	インターフェイス	6-2
6.2	TAP ロジックへのアクセス	6-2
6.3	TAP レジスタ	6-4
6.4	TAP 命令	6-4
6.5	リセット動作	6-5
第 7 章	統合ツール	7-1
7.1	インターゲット・プローブ (ITP)	7-1
7.2	ロジック・アナライザ・インターフェイス (LAI)	7-1
付録 A	信号リファレンス	A-1
A.1	アルファベット順の信号リファレンス	A-1
A.1.1	A[49:3]# (I/O)	A-1
A.1.2	A20M# (I)	A-1
A.1.3	ADS# (I/O)	A-1
A.1.4	AP[1:0]# (I/O)	A-1
A.1.5	ASZ[1:0]# (I/O)	A-1
A.1.6	ATTR[3:0]# (I/O)	A-2
A.1.7	BCLKp/BCLKn (I)	A-2
A.1.8	BE[7:0]# (I/O)	A-3
A.1.9	BERR# (I/O)	A-3
A.1.10	BINIT# (I/O)	A-4
A.1.11	BNR# (I/O)	A-4
A.1.12	BPM[5:0]# (I/O)	A-4
A.1.13	BPRI# (I)	A-4

A.1.14	BR[0]# (I/O) および BR[3:1]# (I)	A-4
A.1.15	BREQ[3:0]# (I/O)	A-5
A.1.16	CCL# (I/O)	A-6
A.1.17	CPUPRES# (O)	A-6
A.1.18	D[127:0]# (I/O)	A-6
A.1.19	D/C# (I/O)	A-6
A.1.20	DBSY# (I/O)	A-6
A.1.21	DBSY_C1# (O)	A-6
A.1.22	DBSY_C2# (O)	A-6
A.1.23	DEFER# (I)	A-7
A.1.24	DEN# (I/O)	A-7
A.1.25	DEP[15:0]# (I/O)	A-7
A.1.26	DHIT# (I)	A-7
A.1.27	DPS# (I/O)	A-8
A.1.28	DRDY# (I/O)	A-8
A.1.29	DRDY_C1# (O)	A-8
A.1.30	DRDY_C2# (O)	A-8
A.1.31	DSZ[1:0]# (I/O)	A-8
A.1.32	EXF[4:0]# (I/O)	A-8
A.1.33	FCL# (I/O)	A-9
A.1.34	FERR# (O)	A-9
A.1.35	GSEQ# (I)	A-9
A.1.36	HIT# (I/O) および HITM# (I/O)	A-9
A.1.37	ID[9:0]# (I)	A-9
A.1.38	IDS# (I)	A-9
A.1.39	IGNNE# (I)	A-9
A.1.40	INIT# (I)	A-9
A.1.41	INT (I)	A-10
A.1.42	IP[1:0]# (I)	A-10
A.1.43	LEN[2:0]# (I/O)	A-10
A.1.44	LINT[1:0] (I)	A-10
A.1.45	LOCK# (I/O)	A-10
A.1.46	NMI (I)	A-11
A.1.47	OWN# (I/O)	A-11
A.1.48	PMI# (I)	A-11
A.1.49	PWRGOOD (I)	A-11
A.1.50	REQ[5:0]# (I/O)	A-11
A.1.51	RESET# (I)	A-12
A.1.52	RP# (I/O)	A-12
A.1.53	RS[2:0]# (I)	A-13
A.1.54	RSP# (I)	A-13
A.1.55	SBSY# (I/O)	A-13
A.1.56	SBSY_C1# (O)	A-13
A.1.57	SBSY_C2# (O)	A-13
A.1.58	SPLCK# (I/O)	A-13
A.1.59	STBn[7:0]# および STBp[7:0]# (I/O)	A-14
A.1.60	TCK (I)	A-14
A.1.61	TDI (I)	A-14
A.1.62	TDO (O)	A-14
A.1.63	THRMTRIP# (O)	A-14
A.1.64	THRMALERT# (O)	A-15

A.1.65	TMS (I)	A-15
A.1.66	TND# (I/O)	A-15
A.1.67	TRDY# (I)	A-15
A.1.68	TRST# (I)	A-15
A.1.69	WSNP# (I/O)	A-15
A.2	信号のまとめ	A-15

2-1	サポートしている並列処理の2つの例	2-2
2-2	Itanium [®] 2 プロセッサ・コア・パイプライン	2-3
2-3	Itanium [®] 2 プロセッサのブロック・ダイアグラム	2-4
2-4	Itanium [®] 2 プロセッサのFMACユニット	2-6
2-5	Itanium [®] 2 プロセッサのキャッシュ階層	2-8
3-1	コモン・クロック・ラッチ・プロトコル	3-2
3-2	ソース・シンクロナス・ラッチ・プロトコル	3-3
5-1	BR[3:0]# の物理的な相互接続 (4つの対称エージェント)	5-4
5-2	BR[3:0]# の物理的な相互接続 (2つの対称エージェント)	5-5
6-1	テスト・アクセス・ポートのブロック・ダイアグラム	6-1
6-2	TAP コントローラのステート・ダイアグラム	6-2

3-1	制御信号	3-4
3-2	アービトレーション信号	3-4
3-3	要求信号	3-5
3-4	スヌープ信号	3-5
3-5	応答信号	3-6
3-6	データ信号	3-7
3-7	STBp[7:0]# と STBn[7:0]# に対応する信号	3-7
3-8	据え置き信号	3-8
3-9	エラー信号	3-8
3-10	実行制御信号	3-9
3-11	プラットフォーム信号	3-10
3-12	診断信号	3-10
4-1	直接的なバス信号の保護	4-2
5-1	電源投入時の設定機能	5-2
5-2	Itanium [®] 2 プロセッサの BREQ[3:0]# バス信号の相互接続 (4 ウェイ・プロセッサ)	5-4
5-3	Itanium [®] 2 プロセッサの BREQ[3:0]# バス信号の相互接続 (2 ウェイ・プロセッサ)	5-4
5-4	アービトレーション ID の設定	5-5
5-5	Itanium [®] 2 プロセッサのシステムバス周波数とプロセッサ・コア 周波数の比の設定	5-5
5-6	Itanium [®] 2 プロセッサのリセット・ステート (PAL ファームウェアの実行後)	5-6
5-7	Itanium [®] プロセッサの INIT ステート	5-6
6-1	Itanium [®] 2 プロセッサの TAP コントローラ用の命令	6-4
A-1	アドレス空間のサイズ	A-2
A-2	有効なメモリ・タイプの信号エンコード	A-2
A-3	バイト・イネーブル上の特殊なトランザクションのエンコード	A-3
A-4	BR0# (I/O)、BR1#、BR2#、BR3# 信号のローテート方式の相互接続 (4P の場合)	A-5
A-5	BR0# (I/O)、BR1#、BR2#、BR3# 信号のローテート方式の相互接続 (2P の場合)	A-5
A-6	BR[3:0]# 信号とエージェント ID	A-5
A-7	DID[9:0]# のエンコード	A-7
A-8	拡張機能信号	A-8
A-9	データ転送のサイズ	A-10
A-10	REQa#/REQb# 信号によって定義されるトランザクションのタイプ	A-12
A-11	STBp[7:0]# および STBn[7:0]# の対応関係	A-14
A-12	出力信号	A-15
A-13	入力信号	A-16
A-14	入力 / 出力信号 (シングル・ドライバ)	A-17
A-15	入力 / 出力信号 (マルチ・ドライバ)	A-18

