

OpenRISC 1200 IP Core Specification

Author: Damjan Lampret
lampret@opencores.org

日本語訳 大川 崇 (Takashi Okawa)
takashi.okawa@ieee.org

Rev. 0.7J
August 2nd, 2010

Preliminary Draft

Revision History

Rev.	Date	Author	Description
0.1	28/3/01	Damjan Lampret	First Draft
0.2	16/4/01	Damjan Lampret	First time published
0.3	29/4/01	Damjan Lampret	All chapters almost finished. Some bugs hidden waiting for an update. Awaiting feedback.
0.4	16/5/01	Damjan Lampret	Synchronization with OR1K Arch Manual
0.5	24/5/01	Damjan Lampret	Fixed bugs
0.6	28/5/01	Damjan Lampret	Changed some SPR addresses.
0.7	06/9/01	Damjan Lampret	Simplified debug unit.
0.7J	02/8/10	Takashi Okawa	Japanese version of Rev. 0.7

Table Of Contents

Introduction	7
OpenRISC Family	7
OpenRISC 1200	8
Features	8
Architecture	9
CPU/DSP	10
Data Cache	13
Instruction Cache	15
Data MMU	17
Instruction MMU	19
Programmable Interrupt Controller	21
Tick Timer	21
Power Management Support	22
Debug unit	22
Clocks & Reset	23
WISHBONE Interfaces	23
Operation	25
Reset	25
CPU/DSP	26
Data Cache Operation	32
Instruction Cache Operation	35
Data MMU	37
Instruction MMU	41
Programmable Interrupt Controller	44
Tick Timer	45
Power Management	45
Debug Unit	46
Development Interface	47
Registers	51
Registers list	51
Register VR description	52
Register UPR description	53
Register CPUCFGR description	54
Register DMMUCFGR description	54
Register IMMUCFGR description	55
Register DCCFGR description	55
Register ICCFGR description	56
Register DCFGR description	57
IO ports	58

Instruction WISHBONE Master Interface 58
 Data WISHBONE Master Interface 59
 System Interface 60
 Development Interface 60
 Power Management Interface 60
 Interrupt Interface 61
 Core HW Configuration 62

Table Of Figures

Figure 1. Core's Architecture.....	9
Figure 2. CPU/DSP Block Diagram.....	10
Figure 3. Block Diagram of Debug Unit.....	23
Figure 4. Power-Up and Reset Sequence.....	25
Figure 5. Power-Up and Reset Sequence w/ Gated Clock.....	25
Figure 6. WISHBONE Write Cycle.....	32
Figure 7. WISHBONE Block Read Cycle.....	33
Figure 8. WISHBONE Block Read/Write Cycle.....	34
Figure 9. WISHBONE Block Read Cycle.....	36
Figure 10. 32-bit Address Translation Mechanism using Two-Level Page Table.....	38
Figure 11. 32-bit Address Translation Mechanism using Two-Level Page Table.....	42
Figure 12. Development Interface Cycles.....	48
Figure 13. Assertion of External Watchpoint Trigger.....	50
Figure 14. Core's Interfaces.....	58

Table Of Tables

Table 1. Possible Data Cache Configurations of OR1200	13
Table 2. Possible Instruction Cache Configurations of OR1200	15
Table 3. Possible Data TLB Configurations of OR1200.....	17
Table 4. Possible Instruction TLB Configurations of OR1200.....	19
Table 5. Block Diagram of the Interrupt Controller.....	21
Table 6. Power Consumption	22
Table 7. List of 32-bit Implemented Instructions	29
Table 8. Execution Time of Integer Instructions.....	30
Table 9. List of Implemented Exceptions	32
Table 10. Protection Attributes for Load/Store Accesses	39
Table 11. Cached and uncached regions	40
Table 11. Protection Attributes for Instruction Fetch Accesses.....	43
Table 13. Cached and uncached regions	44
Table 12. Development Interface Operation Commands	48
Table 13. Status of the Load/Store Unit	49
Table 14. Status of the Instruction Unit	49
Table 15. List of All Registers	52
Table 16. VR Register	53
Table 17. UPR Register.....	53
Table 18. CPUCFGR Register	54
Table 19. DMMUCFGR Register	55
Table 20. IMMUCFGR Register.....	55
Table 21. DCCFGR Register	56
Table 22. ICCFGR Register	57
Table 23. DCFGR Register	57
Table 24. Instruction WISHBONE Master Interface' Signals	59
Table 25. Data WISHBONE Master Interface' Signals	59
Table 26. System Interface Signals	60
Table 27. Development Interface	60
Table 28. Power Management Interface.....	61
Table 29. Interrupt Interface.....	61

1

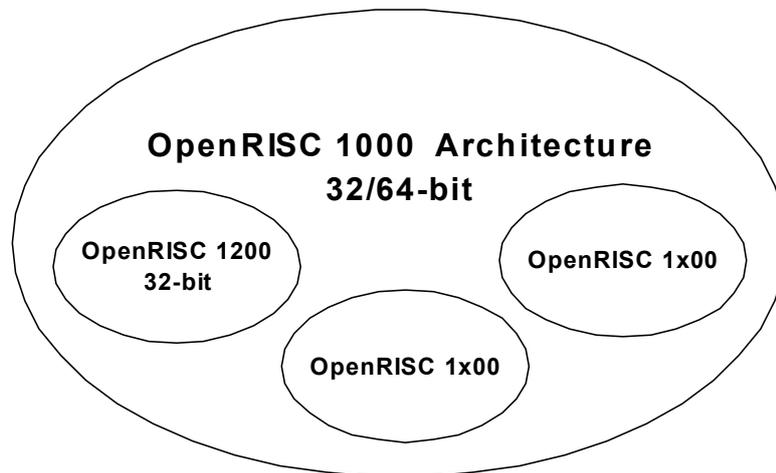
Introduction

この資料は、OpenRISC 1200 の実装仕様書です。ここでは、OpenRISC アーキテクチャーのうち、実装依存となっている全ての仕様が定義されています。これには、データ及び命令キャッシュのタイプとサイズ、データ及び命令MMUのタイプとサイズ、実行パイプラインの詳細、実行ユニットや割り込み制御及びその他のユニットの実装が含まれます。

また、この資料は、命令セットやメモリ・アドレッシング・モード等、OpenRISC アーキテクチャーの基本的な内容は含まれていません。これらについては、OpenRISC 1000 System Architecture Manual を参照してください。

OpenRISC Family

OpenRISC 1000 は、フリーのオープン・ソース OpenRISC プロセッサのアーキテクチャー・ファミリーです。OpenRISC 1000 は、アプリケーションごとの価格と性能に関する最適解を実現する、様々なチップ及びシステム・アーキテクチャーを可能とします。これは、32/64 ビット・ロード/ストア RISC アーキテクチャーで、性能・シンプルな構成・低消費電力への要求・拡張性及び多機能性に重点をおいて設計されています。OpenRISC 1000 アーキテクチャーのターゲットは、中～高性能ネットワーク、組込み機器、自動車、モバイルコンピュータ市場です。



最初の数字が「1」となる「1xxx」の識別番号を持つ OpenRISC の実装は、OpenRISC 1000 ファミリーになります。2 番目の数字は、OpenRISC 1000 アーキテクチャーの実装された機能及びその実装方法について示します。最後の 2 つの数字は、実際のアプリケーションに対応した実装に関するコンフィグレーションを示します。

OpenRISC 1200

OpenRISC 1200 (OR1200) は、ハーバード・アーキテクチャーの 32 ビット・スカラー型 RISC アーキテクチャーで、5 段の整数型パイプライン、MMU、基本 DSP 処理機能を持ちます。データ及び命令キャッシュは、デフォルトの設定では、1-way direct-mapped 8KB 16 byte line サイズとなります。これらのキャッシュはハードウェアによるタグ付けの機能を持ちます。

また、デフォルトの設定では、データ及び命令 MMU が実装され、それらは 64-entry hash based 1 way direct-mapped TLB を持ちます。

その他の機能として、リアルタイム・デバックのためのデバック・ユニット、高分解能タイマー、プログラマブル割り込みコントローラー、パワーマネジメント機能を持ちます。

0.18u 6LM プロセスに実装された OpenRISC 1200 は、300MHz で 300 Dhrystone 2.1 MIPS、300 DSP MAC 32x32 operation 以上の性能を示しました。これは、同等のクラスのプロセッサに対して、20% 以上の高い性能を示しています。また、デフォルトのコンフィギュレーションで実装した場合の規模は、約 1 M トランジスター数となります。

OR1200 は、組込み機器やモバイル機器及びネットワーク機器をターゲットとしております。これは、同クラスの 32-bit RISC プロセッサに対して優位性を持つと共に、最近の Operating System (OS) を効率よく動作させることができます。

OR1200 の競合は、ARM10, ARC Tensilica RISC プロセッサになります。

Features

OR1200 IP core の主な特徴は次の通りです。

- プロセッサ・コアの全ての主要機能に関する実装及び実装方法について、ユーザーが選択/設定することが可能
- 高性能プロセッサ・コア 300 Dhrystone 2.1 MIPS (300 MHz using 0.18u process)
- 高性能キャッシュ及びMMU
- WISHBONE Rev. B 準拠 (SoC バス・インターフェース)

2

Architecture

Figure 1 に、OpenRISC 1200 (OR1200) IP core の概要を示します。OR1200 は、次のブロックから構成されます。

- CPU/DSP central block
- Direct-mapped data cache
- Direct-mapped instruction cache
- Data MMU based on hash based DTLB
- Instruction MMU based on hash based ITLB
- Power management unit and power management interface
- Tick timer
- Debug unit and development interface
- Interrupt controller and interrupt interface
- Instruction and Data WISHBONE host interfaces

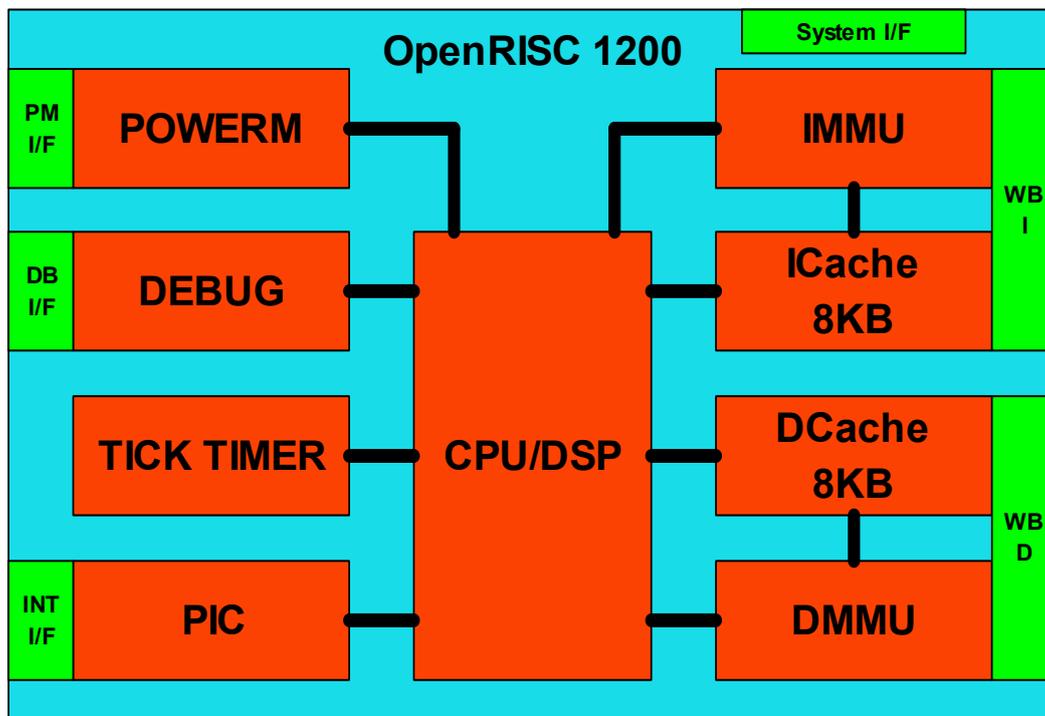


Figure 1. Core's Architecture

CPU/DSP

CPU/DSP は、OR1200 RISC プロセッサの心臓部です。Figure 2 に、CPU/DSP 基本ブロックを示します。

OR1200 CPU/DSP は、OpenRISC 1000 アーキテクチャーの 32-bit の部分のみ実装していません。OpenRISC 1000 アーキテクチャーのうち、64-bit 部分及び浮動小数点演算ユニット及びベクター・ユニットは、OR1200に実装されていません。

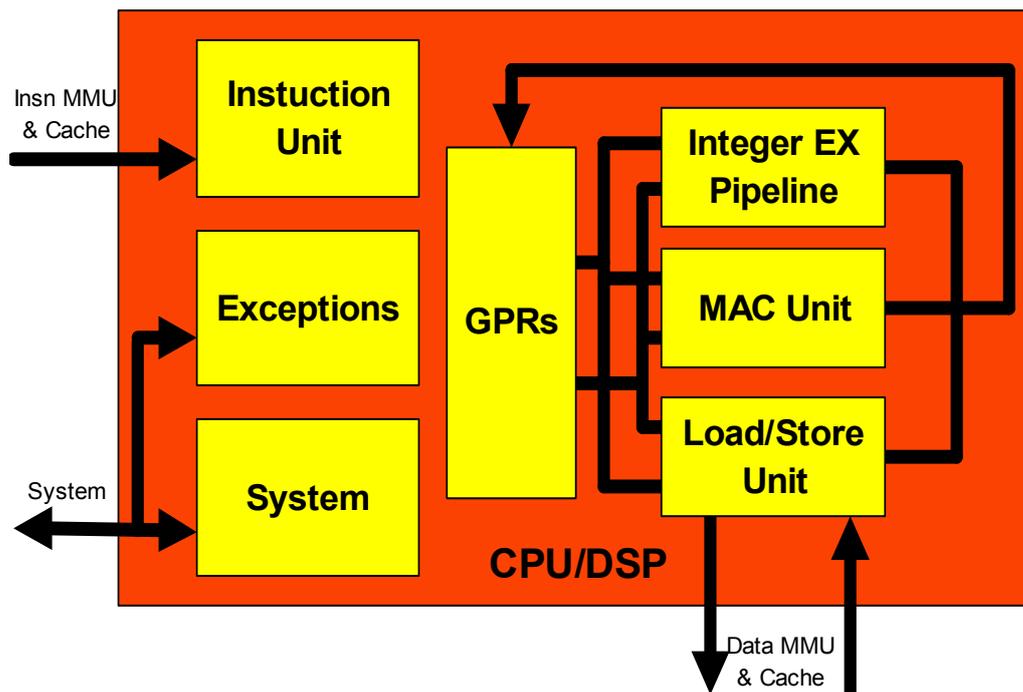


Figure 2. CPU/DSP Block Diagram

Instruction unit

命令ユニットは、基本命令パイプライン、メモリ・サブシステムからの命令フェッチ、実行可能状態となった実行ユニットへの命令ディスパッチ、正確な実行モデルとIn-Orderでの実行を保証するために状態の履歴を維持する機構が実装されています。また、命令ユニットは、条件ブランチ及び無条件ジャンプ命令を実行します。

命令ユニット内部のシーケンサーは、実行可能状態となっている実行ユニットに対して、各クロック・サイクルで連続的に命令をディスパッチします。実行ユニットは、ソース・データが利用可能かどうか識別し、また、他の命令が同じデスティネーション・レジスターに出力しようとしていないことを保障します。

命令ユニットは、ORBIS32命令クラスのみ処理します。現在のOR1200は、ORFPX32/64 及びORVDX64 命令クラスを、サポートしていません。

ORBIS32

Basic Instruction Set with 32 bits wide instructions aligned on 32-bit boundaries in memory and operating on 32 bits

ORFPX32/64

Floating-Point eXtension with 32 bits wide instructions aligned on 32-bit boundaries in memory and operating on 32 bits and 64 bits data

ORVDX64

Vector/DSP eXtension with 32 bits wide instructions aligned on 32-bit boundaries in memory and operating on 8, 16, 32 and 64 bits data

General-Purpose Registers

OR1200 には、32個の 32-bit General Purpose Register (GPR) が実装されています。OpenRISC 1000 アーキテクチャーでは、高速なコンテキスト切り替えを目的とする、レジスタ・ファイルのシャドウ・コピーの機能が定義されていますが、現在の OR1200 では、その機能は実装されていません。

OR1200のGeneral Purpose Register file は、1 ワードあたり 32 bitsで32ワードの容量を持つ2つの同期型 2ポート・メモリで実装されています。

Load/Store Unit

ロード・ストア ユニット (LSU) は、GPRとCPU内部バス間のデータ転送を行います。LSUは実行ユニットに対して独立に実装されていることから、メモリ・サブシステムのストールは、データ依存性を持つパイプライン上の処理に対してのみ影響を与えます。

LSUの主要な機能は次のとおりです。

- アトミック命令を含む全てのロード/ストア命令は、ハードウェアで実装されています
- アドレス入力バッファ
- ロード/ストア命令のパイプライン処理
- 高速アクセスのためのAligned Accesses

ロード及びストア命令が発行されると、LSUは全てのオペランドが利用可能か確認します。それらのオペランドは次のものが含まれます。

- アドレス・レジスタ
- ストア命令のための、ソース・データ・レジスタ
- ロード命令のための、デスティネーション・データ・レジスタ

Integer Execution Pipeline

OR1200には、次の32 bit 整数型命令が実装されています。

- 算術演算命令
- 比較命令
- 論理演算命令
- ローテーション、シフト命令

多くの整数型命令は、1 サイクルで実行されます。（実行サイクルの詳細については、テーブル (TBD) を参照してください）

MAC Unit

MACユニットは、DSPのMACオペレーションを実行します。MACオペレーションは、48-bit の32x32アキュムレーターを利用します。MACユニットは完全にパイプライン化されており、後段のMACオペレーションは次のクロック・サイクルに受け入れることができます。

System Unit

システム・ユニットには、CPU/DSPのうち命令/データ・インターフェースに接続されていないその他の信号が接続されています。このユニットには、Supervisor Register 等の、全ての Special Purpose Registers (SPR)が実装されています。

Exceptions

例外条件が成立すると例外処理が実行されます。OR1200の例外が発生する条件には、次のものが含まれます。

- External interrupt request
- Certain memory access condition
- Internal errors, such as an attempt to execute unimplemented opcode
- System call
- Internal exception, such as breakpoint exceptions

例外処理は、ユーザー・ソフトウェアに対して透過的であり、全てのタイプの例外は同じ仕組みを利用します。例外が発生した場合、例外ごとに定義されたオフセット（アドレス）の例外ハンドラー（Exception Handler）に制御が移ります。例外処理はSupervisor Modeで実行されず。

Data Cache

OR1200 データ・キャッシュのデフォルト・コンフィギュレーションは、8-Kbyte、1-way direct-mapped です。これは高速なデータ・アクセスとなります。しかしながら、データ・キャッシュは、Table 1 のコンフィギュレーションも可能です。

	Direct mapped
1KB per set	1KB
2KB per set	2KB
4KB per set	4KB
8KB per set	8KB (default)

Table 1. Possible Data Cache Configurations of OR1200

主な特徴：

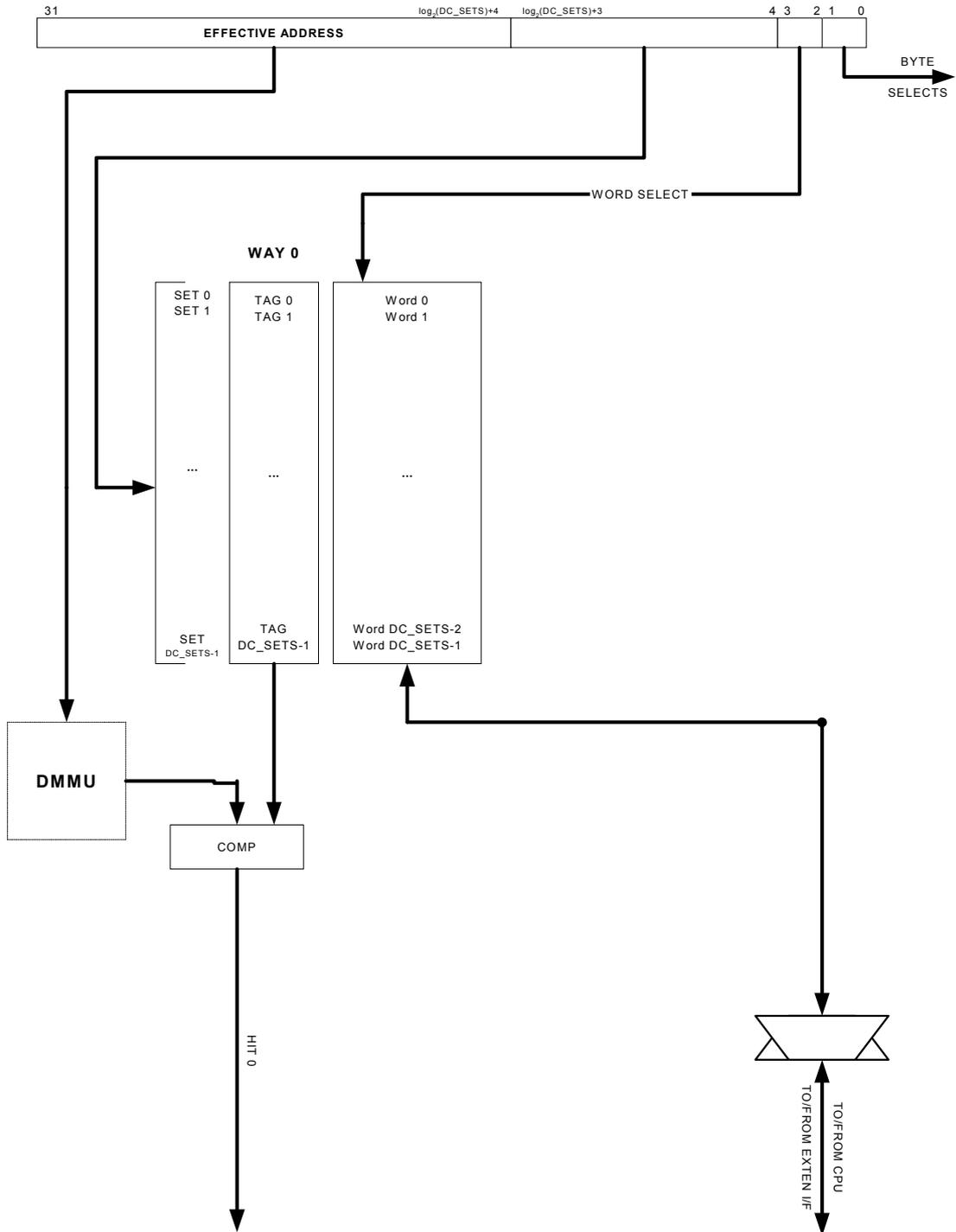
- データ・キャッシュは、命令キャッシュと分離されています（ハーバード・アーキテクチャー）。
- データ・キャッシュは、各Setに対して、Least-Recently Used replacement (LRU) アルゴリズムを適用します。
- キャッシュ・ディレクトリ処理は、ハードウェアで実行されます。物理アドレス・タグは、キャッシュ・ディレクトリに保存されます。
- Write-Through オペレーションをサポートします。
- Special Purpose Register の設定により、キャッシュを無効化することができます。

キャッシュ・ミスが発生した場合、キャッシュは、連続する16-byteデータで補充されます。連続した16-byteのデータは、critical-word-first オペレーションにより、キャッシュに書き込まれます。Critical word は、キャッシュに書き込まれると同時に、データを要求したユニットに送られます。この仕組みにより、キャッシュの書き込み時間を削減し、ストール時間を最小化します。データ・キャッシュは、また、キャッシュ・タグの保存とキャッシュ・ライン交換処理機能を持ちます。

データ・キャッシュは、システム・メモリ・コントローラへの効率的なアクセスを可能とするための外部インターフェースを持ちます。

データ・キャッシュは、ロード・ストア・ユニット (LSU) の32-bit インターフェースにより、データをGPRに供給します。LSUは、論理アドレスを算出する機能を持ち、データ・キャッシュに対して Data Alignment 処理を行い、LSUに入力された順序でロード・ストア処理を実行します。データ・キャッシュへの書き込み処理は、byte, half-word, word 単位で行われます。

データ・キャッシュは、16 bytes データ、state bit、address tagで構成される 1 line を、512 sets 持つ構造となっています。



各ラインは、連続する4つのwordを含みます。これらは、メモリ上の4wordのaligned boundaryからロードされたデータです。結果的に、キャッシュ・ラインはページ境界に整列されます。

Instruction Cache

OR1200 命令キャッシュのデフォルト・コンフィギュレーションは、8-Kbyte、1-way direct-mapped です。これは高速なデータ・アクセスとなります。しかしながら、データ・キャッシュは、Table 2 のコンフィギュレーションも可能です。

	Direct mapped
1KB per set	1KB
2KB per set	2KB
4KB per set	4KB
8KB per set	8KB (default)

Table 2. Possible Instruction Cache Configurations of OR1200

主な特徴：

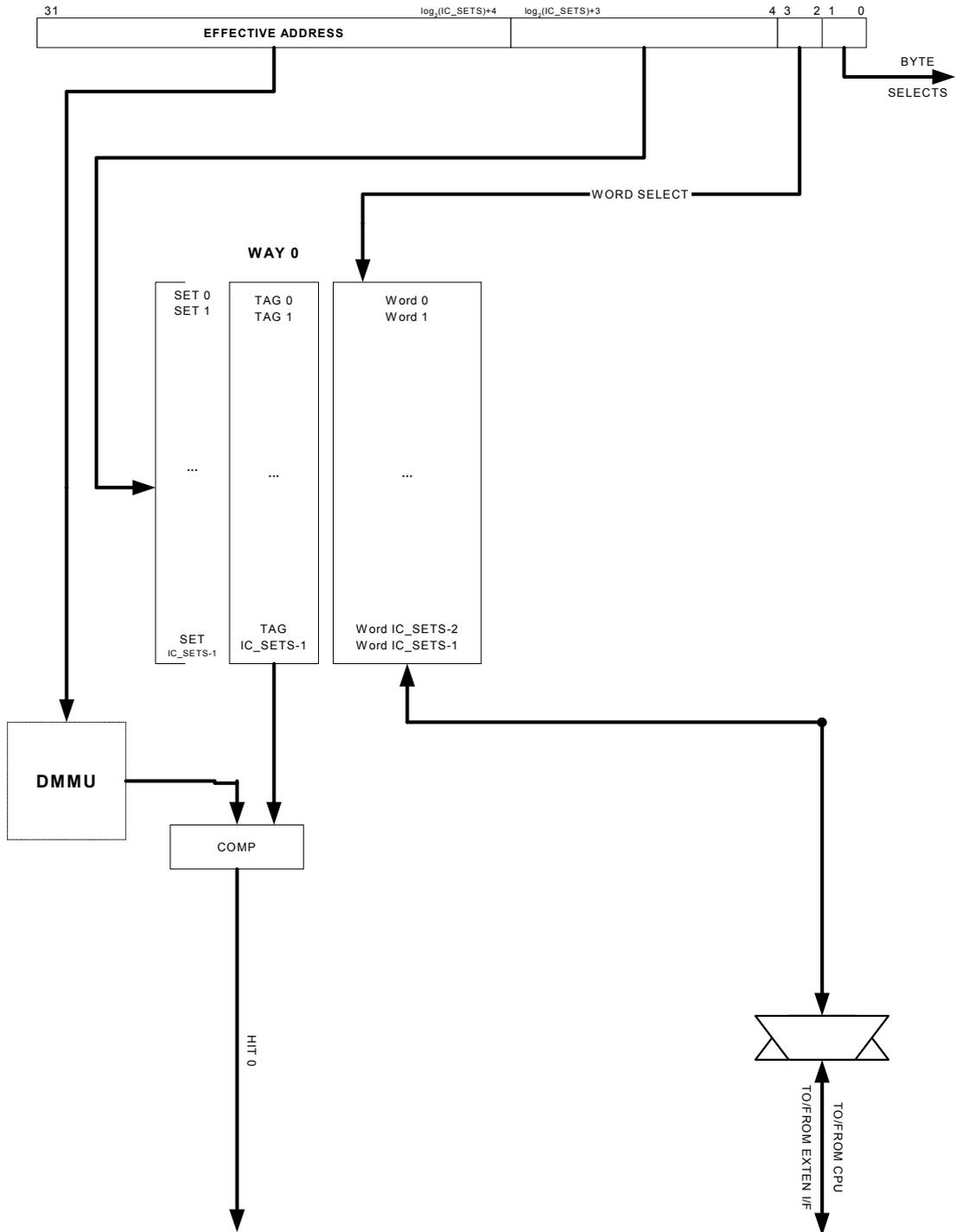
- 命令キャッシュは、データ・キャッシュと分離されています（ハーバード・アーキテクチャ）。
- 命令キャッシュは、各Setに対して、Least-Recently Used replacement (LRU) アルゴリズムを適用します。
- キャッシュ・ディレクトリ処理は、ハードウェアで実行されます。物理アドレス・タグは、キャッシュ・ディレクトリに保存されます。
- Special Purpose Register の設定により、キャッシュを無効化することができます。

キャッシュ・ミスが発生した場合、キャッシュは、連続する16-byteデータで補充されます。連続した16-byteのデータは、critical-word-first オペレーションにより、キャッシュに書き込まれます。Critical word は、キャッシュに書き込まれると同時に、データを要求したユニットに送られます。この仕組みにより、キャッシュの書き込み時間を削減し、ストール時間を最小化します。データ・キャッシュは、また、キャッシュ・タグの保存とキャッシュ・ライン交換処理機能を持ちます。

命令キャッシュは、システム・メモリ・コントローラへの効率的なアクセスを可能とするための外部インターフェースを持ちます。

命令キャッシュは、命令フェッチ・サブユニットの32-bitインターフェースにより、命令シーケンサーに対して各命令を供給します。命令フェッチ・サブユニットは、論理アドレスを算出するための機能を有します。

命令キャッシュは、16 bytes データ、state bit、address tagで構成される 1 line を、512 sets 持つ構造となっています。



各ラインは、連続する4つのwordを含みます。これらは、メモリ上の4wordのaligned boundaryからロードされたデータです。結果的に、キャッシュ・ラインはページ境界に整列されます。

Data MMU

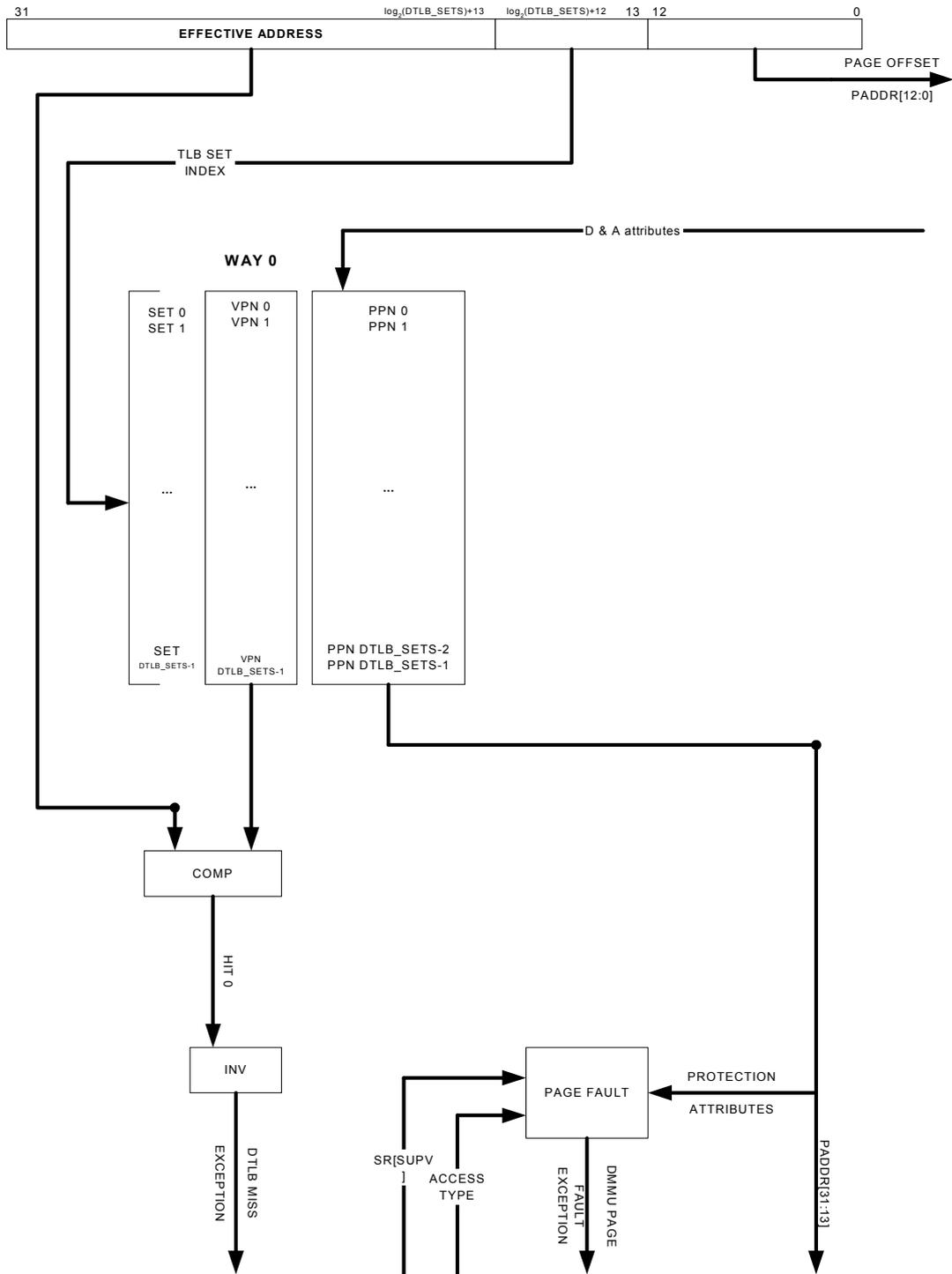
OR1200 は、メモリ・アクセス保護及び論理アドレスを物理アドレスに変換する機能を有するバーチャル・メモリ・マネージメント機構を実装しています。メモリ・アクセス保護機能は、OpenRISC 1000アーキテクチャーに定義されている通り、8-Kbyte と 16-Mbyteのページ単位で有効となります。

	Direct mapped
16 entries per way	16 DTLB entries
32 entries per way	32 DTLB entries
64 entries per way	64 DTLB entries (default)
128 entries per way	128 DTLB entries

Table 3. Possible Data TLB Configurations of OR1200

主な特徴：

- データMMUは、命令MMUと分離されています。
- 包括的なページ保護機構
- デフォルトで 1 way となる、direct mapped hash based translation lookaside buffer (DTLB) を持ち、次の特徴を持ちます。
 - TLB miss and fault exceptions
 - software tablewalk
 - high performance because of hashed based design
 - variable number DTLB entries with default of 64 per each way



MMUハードウェアは、two-level software tablewalk をサポートします。

Instruction MMU

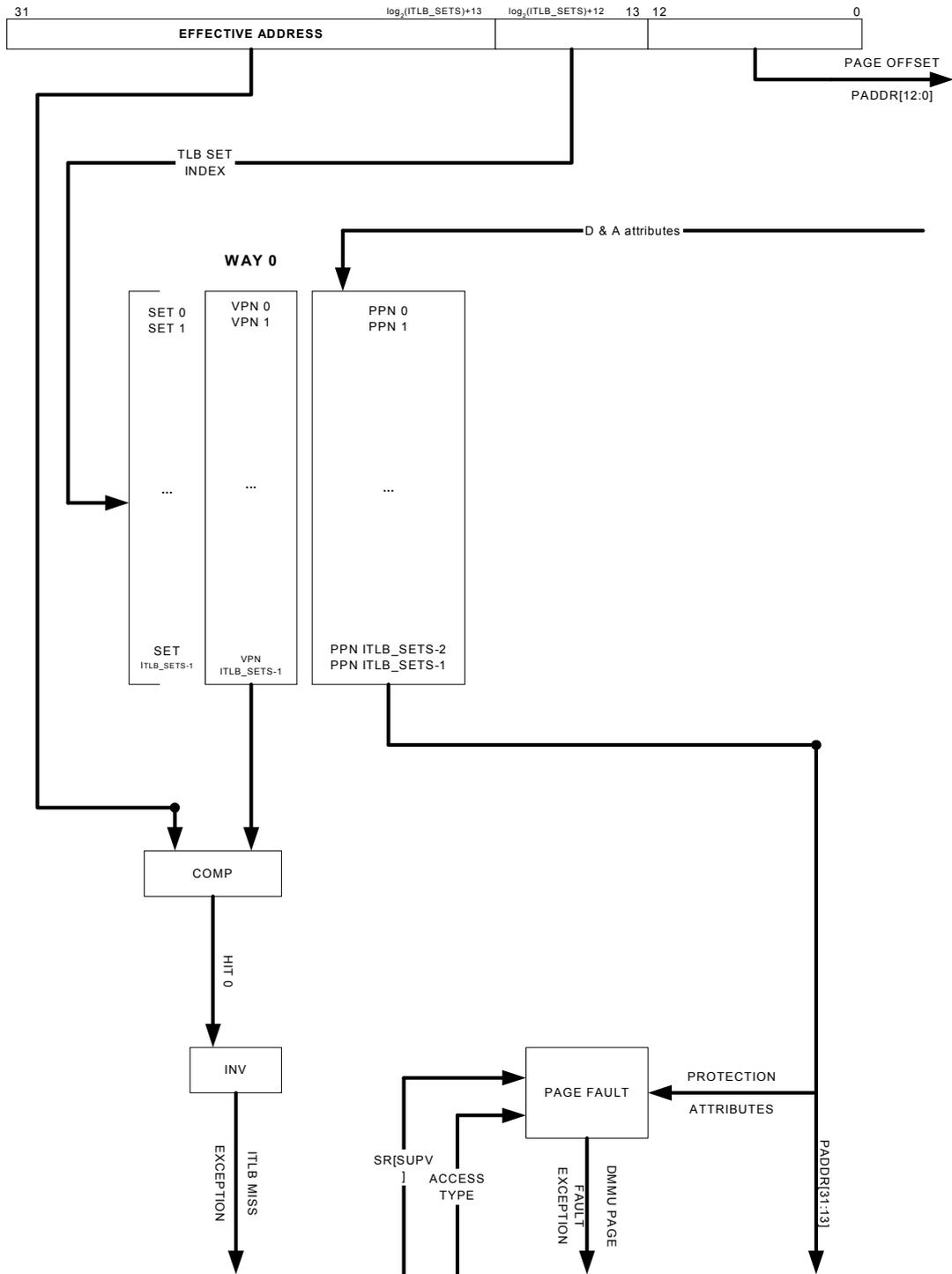
OR1200 は、メモリ・アクセス保護及び論理アドレスを物理アドレスに変換する機能を有するバーチャル・メモリ・マネージメント機構を実装しています。メモリ・アクセス保護機能は、OpenRISC 1000アーキテクチャーに定義されている通り、8-Kbyte と 16-Mbyteのページ単位で有効となります。

	Direct mapped
16 entries per way	16 DTLB entries
32 entries per way	32 DTLB entries
64 entries per way	64 DTLB entries (default)
128 entries per way	128 DTLB entries

Table 4. Possible Instruction TLB Configurations of OR1200

主な特徴：

- 命令MMUは、データMMUと分離されています。
- 包括的なページ保護機構
- デフォルトで 1 way となる、direct mapped hash based translation lookaside buffer (ITLB) を持ち、次の特徴を持ちます。
 - TLB miss and fault exceptions
 - software tablewalk
 - high performance because of hashed based design
 - variable number ITLB entries with default of 64 per each way



MMUハードウェアは、two-level software tablewalk をサポートします。

Programmable Interrupt Controller

割り込みコントローラーは、外部からの割り込み信号を受け、それらを低又は高優先割り込み例外としてCPUコアに通知します。

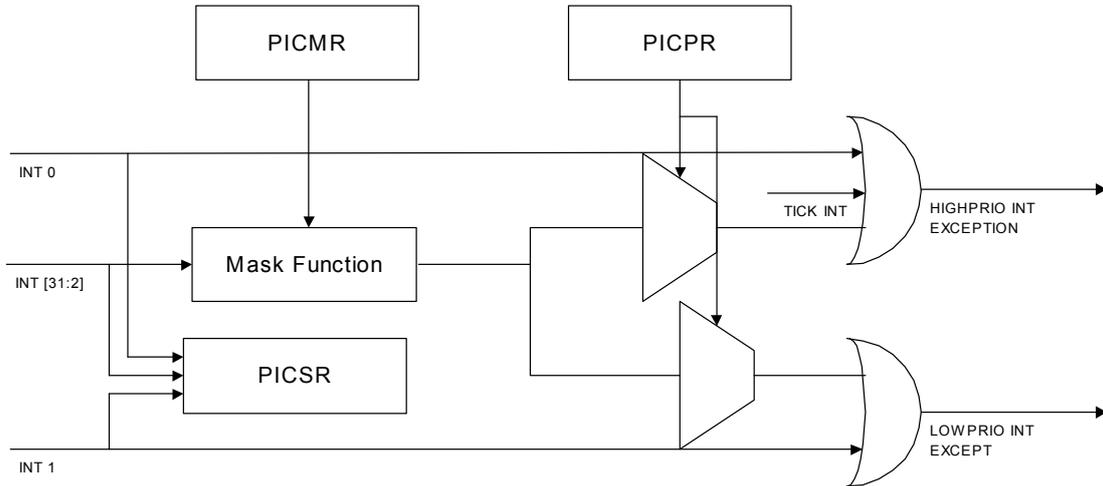


Table 5. Block Diagram of the Interrupt Controller

プログラム可能な割り込みコントローラーは、3つの Special Purpose Registersと 32本の割り込み信号を持ちます。割り込み入力 0 と 1 は常に有効で、それぞれ、高優先割り込み入力、低優先割り込み入力に接続されています。

その他の30の割り込み入力は、Special Purpose Registerにより、無効化及び低・高優先の設定が可能です。

Tick Timer

OR1200 は Tick Timer を実装しています。基本的には、Tick TimerのクロックはRISCクロックが入力され、オペレーティング・システムにおいて正確な時間の測定やシステム・タスクのスケジュールに利用されます。

OR1200 は、次の Tick Timer 機能を正確に実現しています。

- Maximum timer count of 2^{32} clock cycles
- Maximum time period of 2^{28} clock cycles between interrupts
- Maskable tick timer interrupt
- Single run, restartable or continues timer

独立したクロックにより動作する Tick Timer 機能により、”doze power management” モードを実装することができます。

Power Management Support

電力消費の最適化のために、OR1200 は内部のモジュールに対して動的に有効／無効化を行う低電力モードを提供します。

OR1200 は、電力消費を最小化する3つの重要な特徴を持ちます。

- Slow and Idle Modes (ソフトウェア制御によるクロック周波数の低減)
- Doze and Sleep Modes (interrupt wake-up)

Power Minimization Feature	Approx Power Consumption Reduction
Slow and Idle mode	2x – 10x
Doze mode	100x
Sleep mode	200x
Dynamic clock gating	N/A

Table 6. Power Consumption

Slow down モードは、全機能を有効化するために、外部クロック生成回路中のlow-power dividerを利用します。しかし、低い周波数となるため電力消費は減少します。

PRM[SDF] 4 bits が pm_clkds 信号によって各モジュールに伝達され、pm_clkds の値に従い、RISCプロセッサの外部クロック生成を調整します。

ソフトウェアにより Doze モードが開始されると、OR1200で動作中のソフトウェアは、一時中止されます。また、Tick Timer 以外のRISC内部のモジュールに分配されるクロックは無効化されます。一方、チップ上のOR1200以外のブロックは、通常動作を続けることができます。

OR1200 は、未解決の割り込み (pending interrupt) が発生すると、Dozeモードから通常モードに入ります。

Sleepモードでは、全てのOR1200 内部ユニットは無効化され、クロック分配も停止します。必要に応じて、実装がOR1200 core の動作電力を下げることもできます。

OR1200 は、未解決の割り込み (pending interrupt) が発生すると、Sleep モードから通常モードに入ります。

Dynamic Clock Gating は、OR1200ではサポートされていません。

Debug unit

デバック・ユニットは、ソフトウェア開発者によるシステムのソフトウェア・デバックをサポートします。このユニットでは、基本的なデバック機能のみサポートし、WatchPoint、BreakPoint、proglam-flow control registerなどの OpenRISC 1000 アーキテクチャーで定義されているアドバンス・デバック機能はサポートしていません。

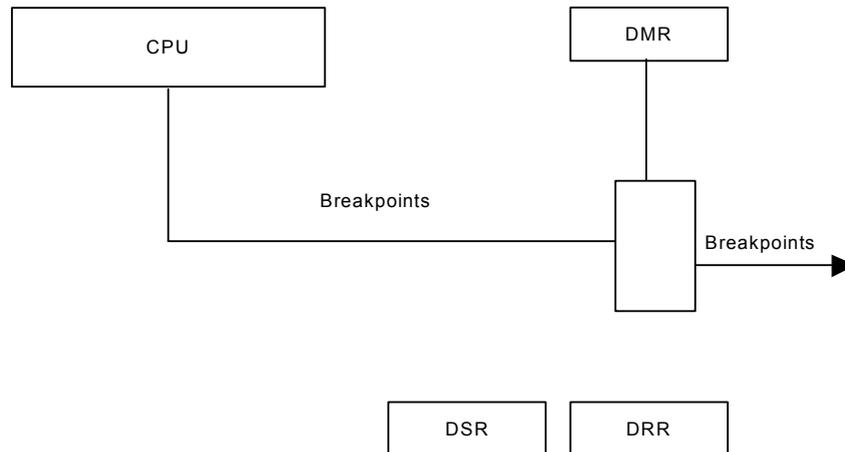


Figure 3. Block Diagram of Debug Unit

Watchpoints と Breakpoints は、デバック・レジスタで設定された条件に動作が適合することでイベントが発生します。Breakpointsは、Watchpointsとは異なり、現在のprogram-flowの実行を一時停止し、Breakpoint 例外により処理を再開させます。

Clocks & Reset

OR1200 core は、いくつかのクロック入力があります。"clk_cpu" 信号はCPU/DSPブロックと、その他の独自のクロックを持たないRISC内部のブロックに、クロック信号として接続されます。データ・キャッシュのクロックは"clk_dc"信号で、命令キャッシュのクロックは "clk_ic" 信号、データMMUのクロックは "clk_dmmu"信号、命令MMUのクロックは "clk_immu"信号、Tick Timerのクロックは "clk_tt"信号になります。全てのクロックは、低クロック・スキューで同位相である必要があります。

OR1200 は、非同期リセット信号を持ちます。リセット信号は "rst" で、"High"にアサートすることで OR1200内部の全てのFlip-Flopを直接リセットします。

WISHBONE Interfaces

2つの WISHBONE インターフェースは、OR1200 coreと外部のペリフェラル及び外部メモリ・サブシステムを接続します。それらは、WISHBONE SoC Interconnection specification Rev. B に準拠します。これらは、32-bit バス幅で実装されており、その他のバス幅はサポートされていません。



3

Operation

このセクションでは、OR1200 のオペレーションについて説明します。OpenRISC 1000 アーキテクチャーに関するオペレーションについては、OpenRISC 1000 System Architecture Manual を参照してください。

Reset

OR1200 は非同期リセット信号を持ち、システム・レベルでのソフト・リセット、ハード・リセットとして利用されます。

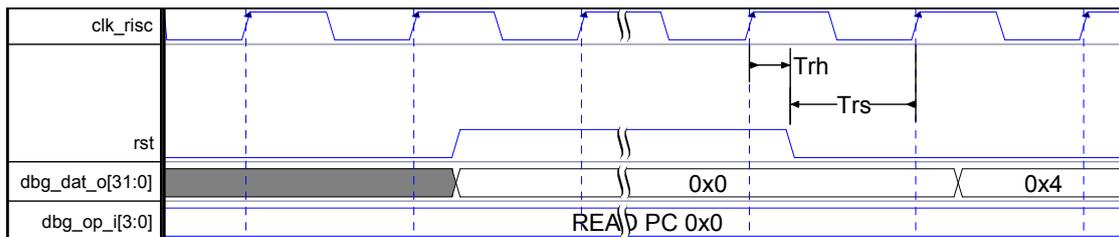


Figure 4. Power-Up and Reset Sequence

Figure 4 に、OR1200 core に電源が投入された後に、非同期リセットの適用方法について示します。リセット信号は、RISC core 内部のほとんどのフリップ・フロップの非同期リセット端子に接続されています。このため、リセット信号は、フリップ・フロップに入力される RISC clock に対して、Hold Time と Setup Time を保証する必要があります。

システムが Gated Clock を実装している場合は、リセット期間中のクロックをゲートすることで、リセットのタイミングを保証することができます。

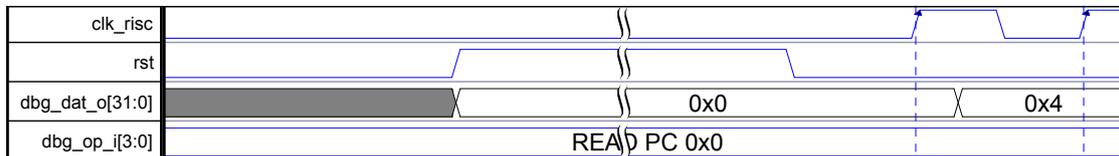


Figure 5. Power-Up and Reset Sequence w/ Gated Clock

CPU/DSP

CPU/DSP は、OpenRISC 1000 アーキテクチャーの 32-bit 部分を実装しており、全ての特徴のサブセットのみ実装しています。

Instructions

Table 7 に、OR1200 に実装されている全ての命令を示します。

Insn	3 1	3 1	3 1	3 1	3 1	2 6	2 5	2 5	2 5	2 10	2 0	2 0	2 0	1 6	1 5	1 5	1 5	1 5	1 1	10	9	8	7	7	7	4	3	3	3	0		
l.add	opcode 0x38					D					A					B					reserved	opcode 0x0		reserved					opcode 0x0			
	3 1	3 0	3 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	10	9	8	7	6	5	4	3	2	1	0
l.addc	opcode 0x38					D					A					B					reserved	opcode 0x0		reserved					opcode 0x1			
	3 1	3 0	3 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	10	9	8	7	6	5	4	3	2	1	0
l.addi	opcode 0x27					D					A					I																
	3 1	3 0	3 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	10	9	8	7	6	5	4	3	2	1	0
l.and	opcode 0x38					D					A					B					reserved	opcode 0x0		reserved					opcode 0x3			
	3 1	3 0	3 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	10	9	8	7	6	5	4	3	2	1	0
l.andi	opcode 0x29					D					A					K																
	3 1	3 0	3 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	10	9	8	7	6	5	4	3	2	1	0
l.bf	opcode 0x4					N																										
	3 1	3 0	3 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	10	9	8	7	6	5	4	3	2	1	0
l.bnf	opcode 0x3					N																										
	3 1	3 0	3 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	10	9	8	7	6	5	4	3	2	1	0
l.trap	opcode 0x2100															K																
	3 1	3 0	3 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	10	9	8	7	6	5	4	3	2	1	0
l.j	opcode 0x0					N																										
	3 1	3 0	3 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	10	9	8	7	6	5	4	3	2	1	0
l.jal	opcode 0x1					N																										
	3 1	3 0	3 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	10	9	8	7	6	5	4	3	2	1	0
l.jalr	opcode 0x12					Reserved										B					reserved											
	3 1	3 0	3 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	10	9	8	7	6	5	4	3	2	1	0

l.jr	opcode 0x11	Reserved								B				reserved																			
	3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	10	9	8	7	6	5	4	3	2	1	0
l.lbs	opcode 0x24	D								A				I																			
	3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	10	9	8	7	6	5	4	3	2	1	0
l.lbz	opcode 0x23	D								A				I																			
	3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	10	9	8	7	6	5	4	3	2	1	0
l.lhs	opcode 0x26	D								A				I																			
	3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	10	9	8	7	6	5	4	3	2	1	0
l.lhz	opcode 0x25	D								A				I																			
	3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	10	9	8	7	6	5	4	3	2	1	0
l.lws	opcode 0x22	D								A				I																			
	3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	10	9	8	7	6	5	4	3	2	1	0
l.lwz	opcode 0x21	D								A				I																			
	3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	10	9	8	7	6	5	4	3	2	1	0
l.mfspr	opcode 0x7	D								A				K																			
	3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	10	9	8	7	6	5	4	3	2	1	0
l.movhi	opcode 0x6	D								Reserved				K																			
	3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	10	9	8	7	6	5	4	3	2	1	0
l.mtspr	opcode 0x10	K								A				B				K															
	3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	10	9	8	7	6	5	4	3	2	1	0
l.mul	opcode 0x38	D								A				B				reserved	opcode 0x3	reserved	opcode 0x6												
	3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	10	9	8	7	6	5	4	3	2	1	0
l.muli	opcode 0x2c	D								A				I																			
	3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	10	9	8	7	6	5	4	3	2	1	0
l.mulu	opcode 0x38	D								A				B				reserved	opcode 0x3	reserved	opcode 0xb												
	3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	10	9	8	7	6	5	4	3	2	1	0
l.nop	opcode 0x15	reserved																															
	3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	10	9	8	7	6	5	4	3	2	1	0
l.or	opcode 0x38	D								A				B				reserved	opcode 0x0	reserved	opcode 0x4												
	3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	10	9	8	7	6	5	4	3	2	1	0

	3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	1 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	9	8	7	6	5	4	3	2	1	0	
l.slli	opcode 0x2e					D					A					reserved					opcode 0x0		L										
	3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	1 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	9	8	7	6	5	4	3	2	1	0	
l.sra	opcode 0x38					D					A					B					reserved	opcode 0x2					reserved	opcode 0x8					
	3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	1 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	9	8	7	6	5	4	3	2	1	0	
l.srai	opcode 0x2e					D					A					reserved					opcode 0x2		L										
	3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	1 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	9	8	7	6	5	4	3	2	1	0	
l.srl	opcode 0x38					D					A					B					reserved	opcode 0x1					reserved	opcode 0x8					
	3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	1 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	9	8	7	6	5	4	3	2	1	0	
l.srli	opcode 0x2e					D					A					reserved					opcode 0x1		L										
	3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	1 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	9	8	7	6	5	4	3	2	1	0	
l.sub	opcode 0x38					D					A					B					reserved	opcode 0x0		reserved					opcode 0x2				
	3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	1 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	9	8	7	6	5	4	3	2	1	0	
l.sw	opcode 0x35					I					A					B					I												
	3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	1 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	9	8	7	6	5	4	3	2	1	0	
l.sys	opcode 0x2000															K																	
	3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	1 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	9	8	7	6	5	4	3	2	1	0	
l.xor	opcode 0x38					D					A					B					reserved	opcode 0x0		reserved					opcode 0x5				
	3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	1 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	9	8	7	6	5	4	3	2	1	0	
l.xori	opcode 0x2b					D					A					I																	

Table 7. List of 32-bit Implemented Instructions

各命令のオペレーション等の詳しい説明は、OpenRISC 1000 System Architecture Manual を参照してください。

Instruction Unit

命令ユニットは、フェッチする命令の論理アドレスの生成と、命令キャッシュから命令をフェッチします。各クロックサイクルにおいて1命令のフェッチが可能です。命令フェッチ論理アドレスは、IMMUによって物理アドレスに変換されます。

General-Purpose Registers

汎用レジスタ (General-Purpose Register, GPR) ファイルは、各クロックサイクルで2つのロード・オペランドを供給し、デスティネーション・レジスタへのデータを保存します。

GPRsは、また、Development Interface を経由して、リード及びライト動作を実行します。

Load/Store Unit

LSU は、データ・キャッシュに該当するデータが存在すると仮定して、1ロード命令を2クロックサイクルで実行します。ストア命令の実行は、データ・キャッシュに該当するデータが存在すると仮定して、1クロックサイクルで行われます。

LSUは、ロード/ストア論理アドレスの生成を実行します。論理アドレスはDMMUによって、物理アドレスに変換されます。

ロード/ストア論理アドレスと、ロード及びストア・データは、Development interface を介してアクセスされます。

Integer Execution Pipeline

このプロセッサ・コアは、次の32ビット整数命令を実装しています。

- 算術命令 Arithmetic instructions
- 比較命令 Compare instructions
- 論理演算命令 Logical instructions
- ローテート及びシフト命令 Rotate and shift instructions

Instruction Group	Clock Cycles to Execute
Arithmetic except Multiply/Divide	1
Multiply	3
Divide	Not implemented
Compare	1
Logical	1
Rotate and Shift	1
Others	1

Table 8. Execution Time of Integer Instructions

Table 9 に、整数型パイプラインにより実行される各命令の実行サイクルを示します。ほとんどの命令は1クロック・サイクルで実行されます。

MAC Unit

MAC ユニットは、`l.mac` 命令を実行します。これは、32x32 の完全なパイプライン化された乗算器と48ビットのアキュムレータを実装しています。また、MACユニットは、各クロックサイクル毎に1つの新しい `l.mac` 命令を受け付けることができます。

System Unit

システム・ユニットは、システム・コントロール及び `status special-purpose registers` を実装しており、また、全ての `l.mtspr / l.mfspr` 命令を実行します。

Exceptions

このプロセッサ・コアは、Precise 例外モデルを実装しています。これは、例外が認識された時に、次の条件が成立します。

- 例外を発生させた命令の後続命令を破棄します。
- 例外を発生させた命令よりも前に発行された命令は、通常に完了させ結果を保存しません。
- 例外を発生させた命令のアドレスを、EPCRレジスタに格納し、状態をESRレジスタに保存します。

EXCEPTION TYPE	VECTOR OFFSET	CAUSING CONDITIONS
Reset	0x100	Caused by reset.
Bus Error	0x200	Caused by an attempt to access invalid physical address.
Data Page Fault	0x300	Generated artificially by DTLB miss exception handler when no matching PTE found in page tables or page protection violation for load/store operations.
Instruction Page Fault	0x400	Generated artificially by ITLB miss exception handler when no matching PTE found in page tables or page protection violation for instruction fetch.
Low Priority External Interrupt	0x500	Low priority external interrupt asserted.
Alignment	0x600	Load/store access to naturally not aligned location.
Illegal Instruction	0x700	Illegal instruction in the instruction stream.
High Priority External Interrupt	0x800	High priority external interrupt asserted.

D-TLB Miss	0x900	No matching entry in DTLB (DTLB miss).
I-TLB Miss	0xA00	No matching entry in ITLB (ITLB miss).
System Call	0xC00	System call initiated by software.
Breakpoint	0xD00	Initiated by the debug unit.

Table 9. List of Implemented Exceptions

OR1200 の例外処理には、高速コンテキスト・スイッチのサポートは含まれていません。

Data Cache Operation

Data Cache Load/Store Access

ロード・ストア ユニット(LSU)は、データ・キャッシュからデータを要求し、General-Purpose Register (GPR) file にデータを保存すると共に整数型実行ユニットにデータを転送します。従って、LSUユニットの動作は、データ・キャッシュと密接に関係しています。

データ・キャッシュ・ミス又はDTLBミスが発生しなければ、ロード動作は2クロック・サイクルで実行され、ストア動作は1クロック・サイクルで完了します。また、LSUはデータ・アラインメント処理を行います。

データは、ワード (4 byte, 32-bit) ・ハーフワード (2 byte, 16-bit) ・バイト(1 byte, 8-bit)単位でデータ・キャッシュに書き込まれます。データ・キャッシュはライト・スルー・モードでのみ動作するため、全ての書き込みは直ぐにメイン・メモリー又は次のレベルのキャッシュ (L2) に対しても実行されます。

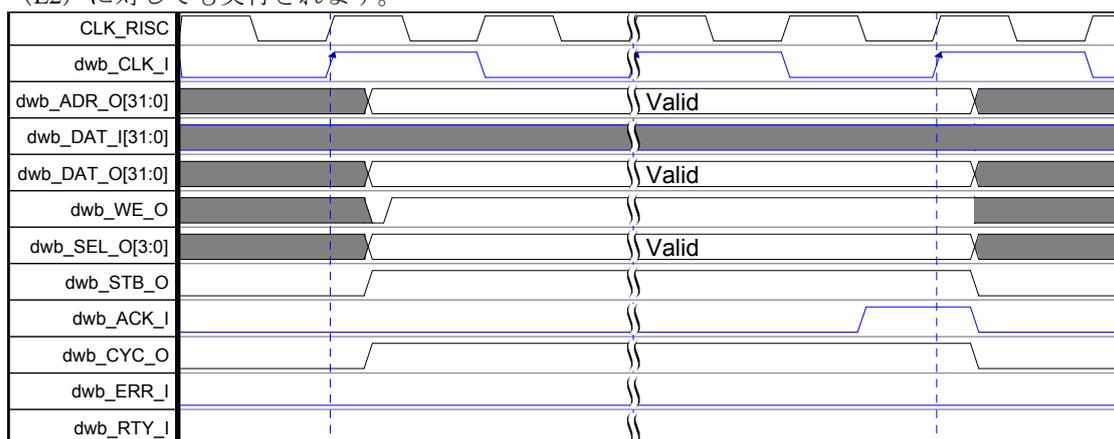


Figure 6. WISHBONE Write Cycle

Figure 6 に、ストア命令がデータ・キャッシュにヒットした場合の、WISHBONEインターフェース上のライト・スルー動作 (実行サイクル) について示します。

dwb_ERR_I 信号又は dwb_RTY_I 信号が、dwb_ACK_I 信号の代わりにアサートされた場合は、バス・エラー例外が発行されます。

Data Cache Line Fill Operation

実行中のロード命令に対してキャッシュ・ミスが発生した場合には、ロード命令が要求するワードが最初にロードされる (critical word first)、連続した4ビートのロードが実行されます。Critical Wordがデータ・キャッシュだけでなくロード・ストア・ユニットに転送されることで、データ・キャッシュ・ミスによるパフォーマンスの低下を最小限に抑えます。

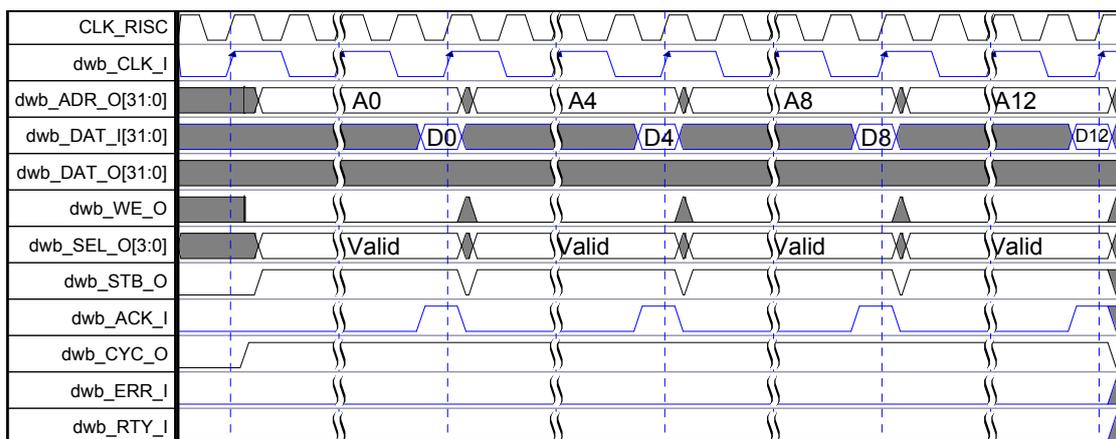


Figure 7. WISHBONE Block Read Cycle

Figure 7 に、4つのリード動作から成る、WISHBONEインターフェース上のキャッシュ・リードの動作を示します。

dwb_ERR_I 信号又はdwb_RTY_I 信号がdwb_ACK_I 信号の代わりにアサートされた場合には、バス・エラー例外が発行されます。

実行中のストア命令に対してキャッシュ・ミスが発生した場合には、ストア命令の対象ワードが最初にロードされる (critical word first)、連続した4ビートのロードが実行されます。連続するワード・データの最初のワードがリードされた後、ストア命令の書き込みデータに関するライト動作が実行されます。このライト動作は、メイン・メモリ又は次のレベルのキャッシュ (L2) に対しても実行されます。ストア命令のデータ幅に関係なく、常にワード幅でのデータ書き込みが実行されます。

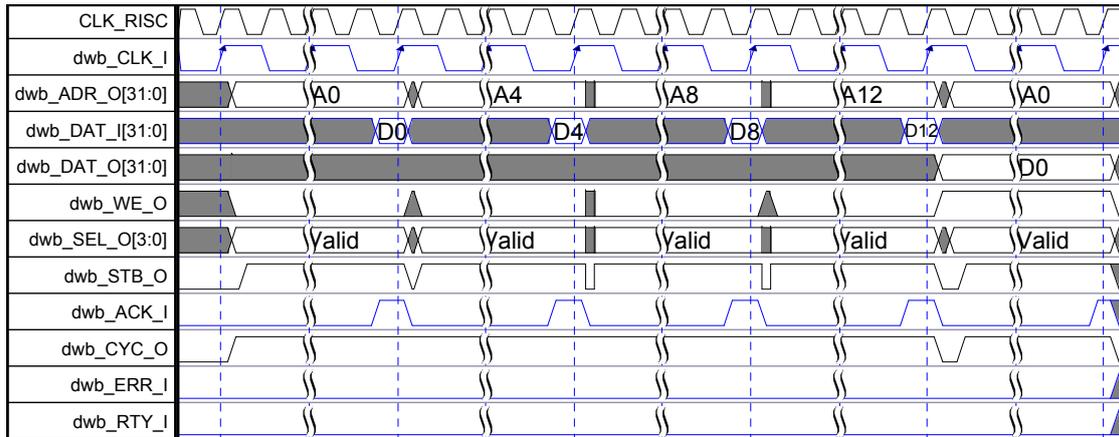


Figure 8. WISHBONE Block Read/Write Cycle

Figure 8 に、WISHBONEインターフェース上の、キャッシュ・リード動作と、それに続くライト動作を示します。

dwb_ERR_I 信号又はdwb_RTY_I 信号がdwb_ACK_I 信号の代わりにアサートされた場合には、バス・エラー例外が発行されます。

Cache/Memory Coherency

OR1200 のデータ・キャッシュは、ライト・スルー・モード (write-through mode) でのみ動作します。なお、OR1200 は、マルチ・プロセッサ環境での利用を想定していません。従って、ローカル・データ・キャッシュと、そのほかのプロセッサ及びメイン・メモリとのコヒーレンシ (coherency) 機能は実装されていません。

Data Cache Enabling/Disabling

データ・キャッシュは、起動時 (power up時) には無効化されています。SRレジスタのSR[DCE] ビットを“1”に設定することで、データ・キャッシュ全体が有効になります。データ・キャッシュが有効化されるまでは、無効化 (Invalidated) に設定されています。

Data Cache Invalidation

OR1200 のデータ・キャッシュは、データ・キャッシュ全体の無効化 (invalidation) はサポートしていません。データ・キャッシュ全体を無効化 (invalidation) するための通常の方法は、全てのデータ・キャッシュに対して個別にアクセスし、各キャッシュ・ラインを無効化 (invalidation) します。

Data Cache Locking

データ・キャッシュには、データ・キャッシュ・コントロール・レジスタ DCCR に、Way ロック・ビットが実装されています。LWxビットを“1”にセットすることで、個々の Way をロックすることができます。

Data Cache Line Prefetch

データ・キャッシュ・ラインのプリフェッチは、OpenRISC 1000 アーキテクチャーにおいてオプションとなる機能であり、OR1200には実装されていません。

Data Cache Line Flush

データ・キャッシュはライト・スルー・モード (write-through mode) 動作することから、データ・キャッシュ・ラインのフラッシュ操作は、各ラインの無効化 (invalidation) のみ実行されます。この操作は、論理アドレス (EA) をDCBFR レジスタに書き込むことで実行されます。

実質的には、データ・キャッシュ・ラインのフラッシュとデータ・キャッシュ・ラインの無効化に違いはありません。

Data Cache Line Invalidate

データ・キャッシュ・ラインの無効化処理は、1つのデータ・キャッシュ・ラインに対して実行されます。論理アドレス (EA) をDCBIRレジスタに書き込むことで、無効化処理が実行されます。

Data Cache Line Write-back

データ・キャッシュはライト・スルー・モード (write-through mode) のみで動作するため、データ・キャッシュ・ラインのライト・バック (write-back) は実行されません。

Data Cache Line Lock

個々のデータ・キャッシュ・ラインをロックする機能は、OR1200 には実装されていません。

Instruction Cache Operation

Instruction Cache Instruction Fetch Access

命令ユニットは、命令キャッシュに命令を要求し、命令ユニット内部の命令キューに転送します。したがって、命令ユニットの動作は命令キャッシュと密接に関係しています。

命令キャッシュ・ライン・ミス及びITLBミスが発生しない場合には、命令フェッチの動作は1クロック・サイクルで実行されます。

命令キャッシュにおいては、データ・キャッシュがストアー命令によって明示的に更新されるような動作は行われません。

Instruction Cache Line Fill Operation

キャッシュ・ミスが発生した場合には、要求されたワードが最初にリードされる（critical word first）、連続した4ビットのリードが実行されます。Critical Wordが命令キャッシュだけでなく命令ユニットに転送されることで、命令キャッシュ・ミスによるパフォーマンスの低下を最小限に抑えます。

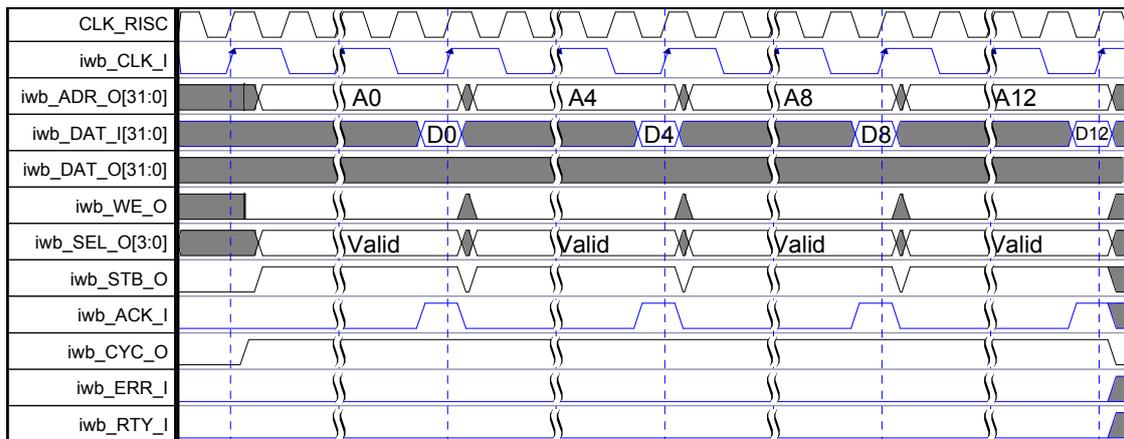


Figure 9. WISHBONE Block Read Cycle

Figure 9 に、WISHBONEインターフェース上の4つのリード転送から成るキャッシュ・ライン・データのリード動作（実行サイクル）について示します。

iwb_ERR_I 信号又は iwb_RTY_I 信号が、iwb_ACK_I 信号の代わりにアサートされた場合は、バス・エラー例外が発行されます。

Cache/Memory Coherency

OR1200 はマルチ・プロセッサ環境で利用されることを想定されていません。従って、ローカル命令キャッシュと、その他のプロセッサ又はメイン・メモリ間でのコヒーレンシー（coherency）機能は実装されていません。

Instruction Cache Enabling/Disabling

命令キャッシュは、起動時（power up）には無効化されています。命令キャッシュ全体は、SRレジスタのSR[ICE]ビットに“1”を書き込むことで有効化されます。命令キャッシュが有効になる前は、無効化（invalidation）されています。

Instruction Cache Invalidation

OR1200 の命令キャッシュは、命令キャッシュ全体の無効化 (invalidation) 機能は実装されていません。命令キャッシュ全体を無効化 (invalidation) するための通常の方法は、全ての命令キャッシュに対して個別にアクセスし、各キャッシュ・ラインを無効化 (invalidation) します。

Instruction Cache Locking

命令キャッシュには、命令キャッシュ・コントロール・レジスタ ICCR に、Way ロック・ビットが実装されています。LWxビットを“1”にセットすることで、個々の Way をロックすることができます。

Instruction Cache Line Prefetch

OpenRISC 1000 アーキテクチャーにおいて、命令キャッシュ・ラインのプリフェッチ機能はオプションであり、OR1200 には実装されていません。

Instruction Cache Line Invalidate

命令キャッシュ・ラインの無効化処理は、1つの命令キャッシュ・ラインに対して実行されます。これは、ICBIR レジスタに論理アドレス (EA) を書き込むことで実行されます。

Instruction Cache Line Lock

個々の命令キャッシュ・ラインをロックする機能は、OR1200 には実装されていません。

Data MMU

Translation Disabled

ロード・ストア・アドレス変換 (address translation) 機能は、SRレジスタのSR[DME]ビットをクリアすることで無効化されます。アドレス変換機能が無効化されると、ロード・ストアにおいて利用される論理アドレスと同様に、データ・キャッシュ・アクセスにおいて、物理アドレスが使用されます。また、オプションとして dwb_ADDR_O 信号に出力されるアドレスが、物理アドレスになります。

Translation Enabled

ロード・ストア・アドレス変換は、SRレジスタのSR[DME]ビットを設定することで有効化されます。アドレス変換が有効化されると、論理アドレス (EA) から物理アドレスへの変換と、メモリ・アクセス時のページ・プロテクション機能が利用できます。

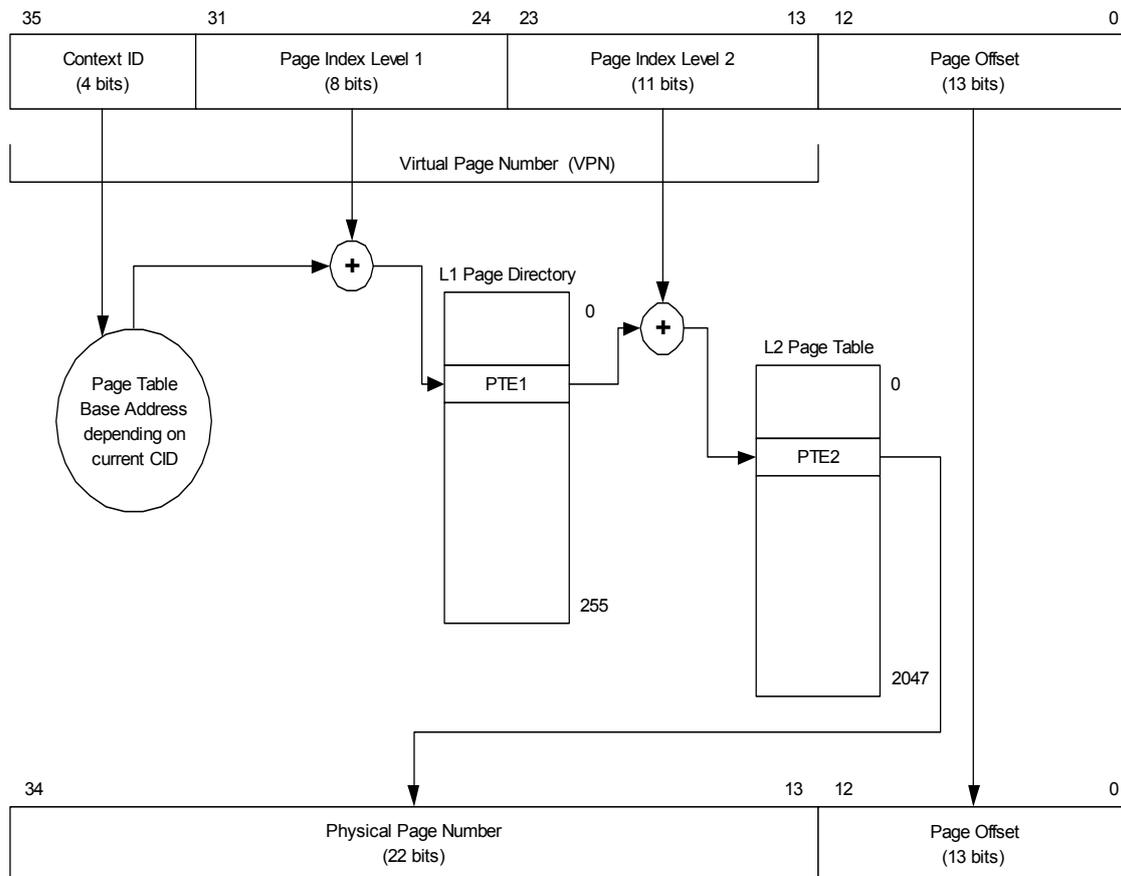


Figure 10. 32-bit Address Translation Mechanism using Two-Level Page Table

OR1200 におけるページ・テーブルは、オペレーティング・システムのバーチャル・メモリ・マネージメント・サブシステムによって管理されなければなりません。Figure 10 に、2段階のページ・テーブルによるアドレス変換処理を示します。アドレス変換の詳細及びページ・テーブル内容の詳細と、1段階でのページ・テーブルによるアドレス変換機能については、OpenRISC 1000 System Architecture Manual を参照してください。

DMMUCR and Flush of Entire DTLB

DMMUCR レジスタは、OR1200 には実装されていません。従って、ページ・テーブル・ベース・ポインター (PTBP) は、ソフトウェアの変数として用意する必要があります。DTLB 全体をフラッシュするためには、全ての各DTLエントリをソフトウェアによりフラッシュする必要があります。ソフトウェア・フラッシュは、ソフトウェアにより各ビットをTLBエントリからPTEに書き戻すことで実行されます。

Page Protection

有効なPTEに定義されたページ内でバーチャル・アドレスが決定された後、そのアドレスは、メモリ保護機構 (memory protection mechanism) により検証されます。この保護機構により当該ページがアクセス禁止とされている場合には、データ・ページ・フォールト例外が生成されません。

メモリ保護機構は、スーパーバイザーとユーザー・モードの両方のリード・アクセス及びライト・アクセスに対して、選択的にアクセス許可を設定することができます。このページ保護機構は、設定可能な様々なページの粒度に対して全てのレベルで保護機能を提供します。

Protection attribute	Meaning
DTLBWyTR[SREx]	Enable load operations in supervisor mode to the page.
DTLBWyTR[SWEEx]	Enable store operations in supervisor mode to the page.
DTLBWyTR[UREx]	Enable load operations in user mode to the page.
DTLBWyTR[UWEEx]	Enable store operations in user mode to the page.

Table 10. Protection Attributes for Load/Store Accesses

Table 10 に、DTLBWyTRレジスタで設定されるページ・プロテクションの属性を示します。個々のページの適切な保護方法は、設定可能な7つのページ保護の方法から、PTEのPPIフィールドによりプログラムされます。OR1200は DMMUPRレジスタを実装していないため、PTE[PPI]の適切な保護ビットへの変換は、ソフトウェア及びDTLBWyTRレジスタへの書き込みにより実行されなければなりません。

DTLB Entry Reload

OR1200 は、ハードウェアによるDTLBエントリの再ロード機能を実装していません。そのため、正しいページ・テーブル・エントリ (PTE) をページ・テーブルから検索する処理とDTLBへコピーする処理は、ソフトウェア関数により実行されなければなりません。

DTLBにコピー (cached) されていない、物理アドレスに対応するロード・ストア論理アドレスがLSUにより算出された場合には、DTLBミス例外が生成されます。

DTLB を再ロードする関数は、正確なPTEをロードし、設定可能なDTLB way の一つを決定し、正確な値をそのEntryのDTLBWyMRとDTLBWyTRレジスタに設定します。

DTLB Entry Invalidation

Special-Purpose Register (SPR) である DTLBEIR に論理アドレスが書き込まれることで、それと一致するローカルDTLB中の該当するエントリが無効化されます。

Locking DTLB Entries

ソフトウェアにより全てのDTLBエントリーの再ロードが実行されると、DTLBエントリーに対するハードウェアによるロックは解除されます。その際、いくつかのエントリーの置き換えが行われないようにする処理は、ソフトウェアの再ロード処理が担う必要があります。

Page Attribute – Dirty (D)

OR1200 の DTLBには、Dirty属性 (D) は実装されていません。このため、ページ保護機構に Dirty属性 (ビット) を持たせることは、オペレーティング・システムが担う必要があります。

Page Attribute – Accessed (A)

OR1200 の DTLBには、Accessed (A) 属性は実装されていません。このため、ページ保護機構に、Accessed属性 (ビット) を持たせることは、オペレーティング・システムが担う必要があります。

Page Attribute – Weakly Ordered Memory (WOM)

OR1200 では、全てのメモリ・アクセスの実行順番は保たれることから、Weakly Ordered Memory (WOM) 属性は必要とされません。このため、この属性は実装されていません。

Page Attribute – Write-Back Cache (WBC)

OR1200 では、データ・キャッシュはライト・スルー・モード (write-through mode) でのみ動作することから、Write-Back Cache (WBC)属性は必要とされません。このため、この属性は実装されていません。

Page Attribute – Caching-Inhibited (CI)

OR1200 の DTLB には、Caching-Inhibited (CI)属性は実装されていません。キャッシュ領域及び非キャッシュ領域は、データ論理アドレス (EA) の30 ビット目によって分けられています。

Effective Address	Region
0x00000000 - 0x3FFFFFFF	Cached
0x40000000 - 0x7FFFFFFF	Uncached
0x80000000 - 0xBFFFFFFF	Cached
0xC0000000 - 0xFFFFFFFF	Uncached

Table 11. Cached and uncached regions

メモリ・マップ上に定義されたI/Oレジスタへのアクセスや、外部インターフェースへ直接リード・ライトのアクセスを行う際には、非キャッシュ領域へのアクセスとして実行する必要があります。

Page Attribute – Cache Coherency (CC)

OR1200 は、マルチ・プロセッサ環境での利用をサポートする機能が実装されていないことや、データ・キャッシュはライト・スルー・モード (Write-Through Mode) でのみ動作するこ

とから、キャッシュ・コヒーレンシー (Cache Coherency, CC) 属性は必要とされません。このため、この属性は実装されていません。

Instruction MMU

Translation Disabled

命令フェッチ・アドレス変換 (address translation) 機能は、SRレジスタのSR[IME]ビットをクリアすることで無効化されます。アドレス変換機能が無効化されると、命令フェッチにおいて利用される論理アドレスと同様に、命令キャッシュ・アクセスにおいて、物理アドレスが使用されます。また、オプションとして `iwb_ADDR_O` 信号に出力されるアドレスが、物理アドレスになります。

Translation Enabled

命令フェッチ・アドレス変換 (address translation) は、SRレジスタのSR[IME]ビットを設定することで有効化されます。アドレス変換が有効化されると、論理アドレス (EA) から物理アドレスへの変換と、命令フェッチ時のページ・プロテクション機能が利用できます。

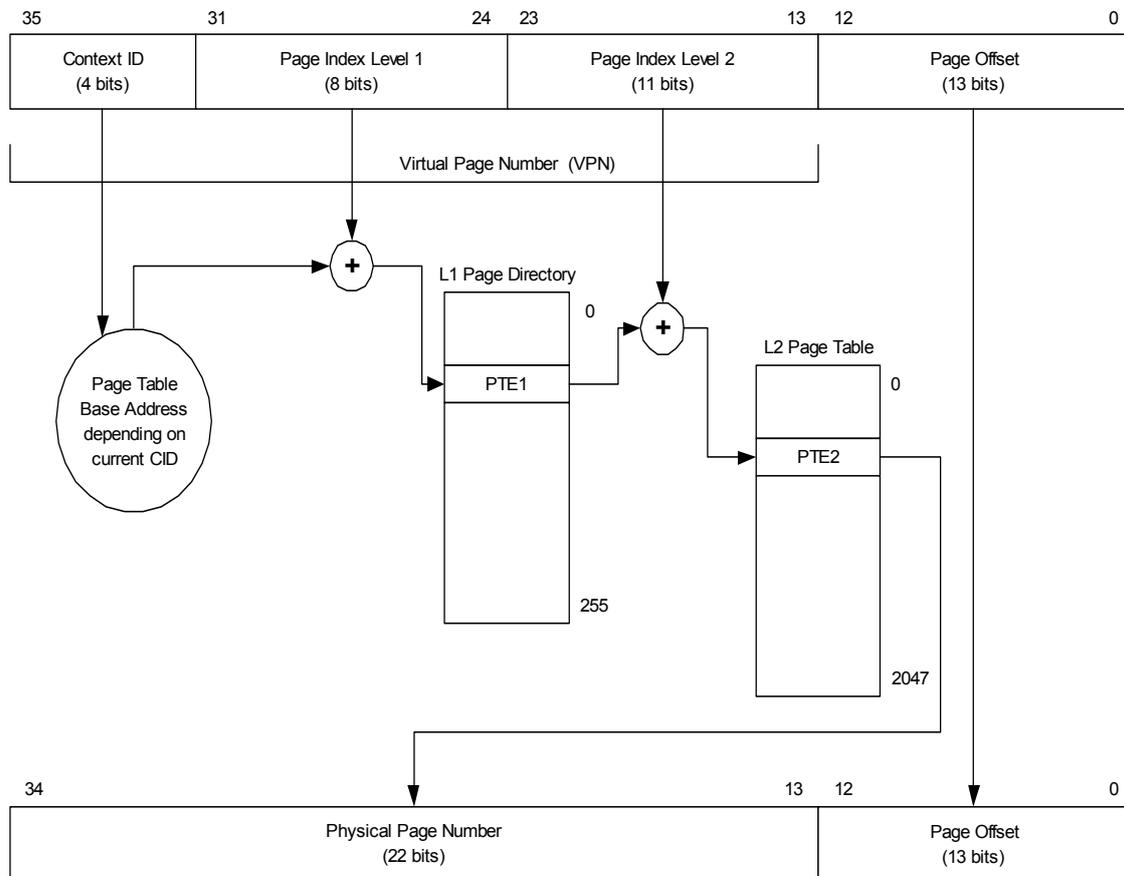


Figure 11. 32-bit Address Translation Mechanism using Two-Level Page Table

OR1200 におけるページ・テーブルは、オペレーティング・システムのバーチャル・メモリ・マネージメント・サブシステムによって管理されなければなりません。Figure 11 に、2段階のページ・テーブルによるアドレス変換処理を示します。アドレス変換の詳細及びページ・テーブル内容の詳細と、1段階でのページ・テーブルによるアドレス変換機能については、OpenRISC 1000 System Architecture Manual を参照してください。

IMMUCR and Flush of Entire ITLB

IMMUCR レジスタは、OR1200 には実装されていません。従って、ページ・テーブル・ベース・ポインター (PTBP) は、ソフトウェアの変数として用意する必要があります。ITLB全体をフラッシュするためには、全ての各ITLエントリをソフトウェアによりフラッシュする必要があります。ソフトウェア・フラッシュは、ソフトウェアにより各ビットをTLBエントリからPTEに書き戻すことで実行されます。

Page Protection

有効なPTEに定義されたページ内でバーチャル・アドレスが決定された後、そのアドレスは、メモリ保護機構（memory protection mechanism）により検証されます。この保護機構により当該ページがアクセス禁止とされている場合には、データ・ページ・フォールト例外が生成されません。

メモリ保護機構は、スーパーバイザーとユーザー・モードの両方のリード・アクセス及びライト・アクセスに対して、選択的にアクセス許可を設定することができます。このページ保護機構は、設定可能な様々なページの粒度に対して全てのレベルで保護機能を提供します。

Protection attribute	Meaning
ITLBWyTR[SXEx]	Enable execute operations in supervisor mode of the page.
ITLBWyTR[UXEx]	Enable execute operations in user mode of the page.

Table 12. Protection Attributes for Instruction Fetch Accesses

Table 13 に、ITLBWyTRレジスタで設定されるページ・プロテクションの属性を示します。個々のページの適切な保護方法は、設定可能な7つのページ保護の方法から、PTEのPPIフィールドによりプログラムされます。OR1200は IMMUPRレジスタを実装していないため、PTE[PPI]の適切な保護ビットへの変換は、ソフトウェア及びITLBWyTRレジスタへの書き込みにより実行されなければなりません。

ITLB Entry Reload

OR1200 は、ハードウェアによるITLBエントリの再ロード機能を実装していません。そのため、正しいページ・テーブル・エントリ（PTE）をページ・テーブルから検索する処理とITLBへコピーする処理は、ソフトウェア関数により実行されなければなりません。

ITLBにコピー（cached）されていない、物理アドレスに対応する命令フェッチ論理アドレスが算出された場合には、ITLBミス例外が生成されます。

ITLB を再ロードする関数は、正確なPTEをロードし、設定可能なITLB way の一つを決定し、正確な値をそのEntryのITLBWyMRとITLBWyTRレジスタに設定します。

ITLB Entry Invalidation

Special-Purpose Register (SPR) である ITLBEIR に論理アドレスが書き込まれることで、それと一致するローカルITLB中の該当するエントリが無効化されます。

Locking ITLB Entries

ソフトウェアにより全てのITLBエントリーの再ロードが実行されると、ITLBエントリーに対するハードウェアによるロックは解除されます。その際、いくつかのエントリーの置き換えが行われないようにする処理は、ソフトウェアの再ロード処理が担う必要があります。

Page Attribute – Dirty (D)

Dirty属性 (D) はPTEに属しますが、IMMUに利用されることはありません。

Page Attribute – Accessed (A)

OR1200 の DTLBには、Accessed (A) 属性は実装されていません。このため、ページ保護機構に、Accessed属性 (ビット) を持たせることは、オペレーティング・システムが担う必要があります。

Page Attribute – Weakly Ordered Memory (WOM)

OR1200 では、全ての命令フェッチ・アクセスの実行順番は保たれることから、Weakly Ordered Memory (WOM) 属性は必要とされません。このため、この属性は実装されていません。

Page Attribute – Write-Back Cache (WBC)

Write-Back Cache 属性 (WBC) はPTEに属しますが、IMMUに利用されることはありません。

Page Attribute – Caching-Inhibited (CI)

OR1200 の ITLB には、Caching-Inhibited (CI)属性は実装されていません。キャッシュ領域及び非キャッシュ領域は、命令論理アドレス (EA) の30 ビット目によって分けられています。

Effective Address	Region
0x00000000 - 0x3FFFFFFF	Cached
0x40000000 - 0x7FFFFFFF	Uncached
0x80000000 - 0xBFFFFFFF	Cached
0xC0000000 - 0xFFFFFFFF	Uncached

Table 13. Cached and uncached regions

Page Attribute – Cache Coherency (CC)

Cache Coherency 属性 (CC) は、PTEに属しますが、IMMUに利用されることはありません。

Programmable Interrupt Controller

PICMR Special-Purpose Register (SPR) は、30 個までのプログラム可能な割り込みの発生の伝達を制御 (mask or unmask) するために利用されます。PICPR Special-Purpose Register は、最大 30個の割り込みに対して、低又は高優先の属性を割り当てるために利用されます。

PICSR Special-Purpose Register (SPR) は、各割り込み入力の状態を表すために利用されます。PICSR のビットは割り込み入力の状態を表し、実際の割り込み信号は、割り込みを発生させたデバイスによってクリアされなければなりません。

Tick Timer

Tick Timer 機能は、TTMR[M] の設定によって有効になります。TTCR の値は、クロック・サイクル毎に加算されます。そして、TTCR レジスタの下位 28 ビットの値が TTMR[TP] の値と一致し TTMR[IE] がセットされた場合にはいつでも、最も優先度の高い割り込みがアサートされます。

TTCR が一致し TTMR[M] の値が “0x1” に設定されている場合には、TTCR は “0” から カウントを再開します。TTMR[M] の値が “0x2” の場合には、TTCR が一致した後 TTCR は停止します。その後、カウントを再開する際には、TTCR の値を変更する必要があります。TTMR[M] が “0x3” に設定されている場合には、TTCR の値が一致してもカウントを継続します。

Power Management

Clock Gating and Frequency Changing Versus CPU Stalling

システムがクロック・ゲーティング (clock gating) をサポートしない場合で、クロック周波数をスロー・ダウン・モード (slow down mode) に変更することができない場合には、CPU は一定のクロック・サイクルの間、ストールすることができます。この方法では電力消費を低減しますが、電力消費の効果は非常に低い方法です。

Slow Down Mode

スロー・ダウン・モード (Slow down mode) は、ソフトウェアにより PMR[SDF] の 4-bit を設定することで制御されます。低い値を設定することは、プロセッサ・コアのより高い性能を得ることになります。通常は、PMR[SDF] の値は、プロセッサ・コアの使用状況をモニターするオペレーティング・システムのアイドル・ルーチン (IDLE Routine) により動的に設定されます。

PMR[SDF] は pm_clkspd 信号により、チップ内に通知されます。外部クロック生成回路は、pm_clkspd 信号の値に従い、クロック周波数を調整する必要があります。的確なスロー・ダウンのための要因について定義されていませんが、“0xF” に設定した場合には、クロック周波数が 32.768 KHz に低下します。

pm_clkspd 信号が “0xF” の値の場合には、pm_lvolt 信号がアサートされます。これは、外部電力供給源に対する低い電力への変更指示になります。

Doze Mode

ドーズ・モード (Doze Mode) への切り替えるためには、ソフトウェアが PMR[DME] を設定する必要があります。プログラマブル割り込みコントローラ (PIC) からの割り込みを受信後、pm_wakeup 信号がアサートされ、外部クロック生成回路が全てのクロックに対して有効になります。クロックが生成される (running) と、RISC は通常モード (normal mode) に戻り、PMR[DME] はクリアされます。

ドーズ・モード (Doze mode) が有効となった場合、pm_dc_gate, pm_ic_gate, pm_dmmu_gate, pm_immu_gate, pm_cpu_gate 信号がアサートされます。その結果、clk_tt 信号以外の全てのクロック信号は、外部クロック生成回路によってゲート (gated) されます。

Sleep Mode

スリープ・モードへ切り替えるためには、ソフトウェアが PMR[SME] を設定する必要があります。プログラマブル割り込みコントローラ (PIC) からの割り込みを受信後、pm_wakeup 信号がアサートされ、外部クロック生成回路が全てのクロックに対して有効になります。クロックが生成 (running) されると、RISC は通常モード (normal mode) に戻り、PMR[SME] はクリアされます。

スリープ・モードが有効となった場合、pm_dc_gate, pm_ic_gate, pm_dmmu_gate, pm_immu_gate, pm_cpu_gate, pm_tt_gate 信号がアサートされます。その結果、clk_tt 信号を含む全てのクロック信号は、外部クロック生成回路によってゲート (gated) されます。

スリープ・モードにおいて、pm_lvolt 信号がアサートされます。これは、外部電力供給源に対する低い電力への変更指示になります。

Clock Gating

クロック・ゲーティング (Clock gating) は、OR1200 の電力制御機能としては実装されていません。

Disabled Units Force Clock Gating

Special-Purpose Register (SPR) である SR の設定により無効化されるユニットには、クロックをゲートする信号があります。また、SR[DCE], SR[ICE], SR[DME], SR[IME] をクリアすることにより、次の信号が直接アサートされます。pm_dc_gate, pm_ic_gate, pm_dmmu_gate, pm_immu_gate

Debug Unit

デバッグ・ユニットは、開発インターフェース (development interface) を通して制御することができ、また、RISC に常駐しているデバック・ソフトウェアにより独立して動作させることができます。

Watchpoints

OR1200 デバックユニットは、OpenRISC 1000 アーキテクチャーで定義されている ウォッチポイント (Watchpoints) を実装していません。

Breakpoint Exception

ウォッチポイント (Watchpoints) を定義している Debug Mode Register 2 のDMR2[WGB] ビットの設定により、ブレイク・ポイント (breakpoint) 例外が起動します。ブレイク・ポイント (breakpoint) 例外の起動により、デバッガー (target resident debugger) が起動されます。

ブレイク・ポイント (Breakpoint) は、開発インターフェース (development interface) 上の dbg_bp_o 信号により、通知 (broadcasted) されます。

Development Interface

追加して利用する “*development and debug interface IP core*” は、IEEE.1149.1 (JTAG) protocol をサポートする標準のデバッガが、OpenRISC 1200 に接続するために利用されます。

Debugging Through Development Interface

DSR (Debug Stop Register) Special-Purpose Register (SPR) は、例外ハンドラーの実行を停止し、開発インターフェース (development intrface) に制御を移すための例外を定義します。それは、常駐するデバッグ・ソフトウェア (resident debug software) または開発インターフェースにより設定することができます。

DRR (Debug Reason Register) Special-Purpose Register (SPR) は、プログラム (program flow) の実行を停止し、その上、開発インターフェース (development intrface) に制御を移したイベントを表示します。それは、常駐するデバッグ・ソフトウェア (resident debug software) または開発インターフェースによりクリアします。

DIR Special-Purpose Register (SPR) は、実装されていません。

Reading PC, Load/Store EA, Load Data, Store Data, Instruction

プログラム・カウンタ (PC)、ロード・ストア論理アドレス (LSEA)、ロード・データ、ストア・データ、現在の実行パイプライン中の命令 のような重要な情報は、開発インターフェース (development interface) を通して、非同期的に読む (read) ことができます。

dbg_op_i[2:0]	Meaning
0x0	Reading Program Counter (PC)
0x1	Reading Load/Store Effective Address
0x2	Reading Load Data
0x3	Reading Store Data

0x4	Reading SPR
0x5	Writing SPR
0x6	Reading Instruction in Execution Pipeline
0x7	Reserved

Table 14. Development Interface Operation Commands

Table 14 に、開発インターフェース (developmentn interface) を介して 実施する、読み書き (read or write) 制御のためのコマンド一覧を示す。SPR への Read/Write 以外の全てのReadコマンドは非同期に実行される。

Reading and Writing SPRs Through Development Interface

SPR への Read , Write アクセスには、dbg_op_i 信号を、それぞれ 0x4 , 0x5 に設定する必要があります。

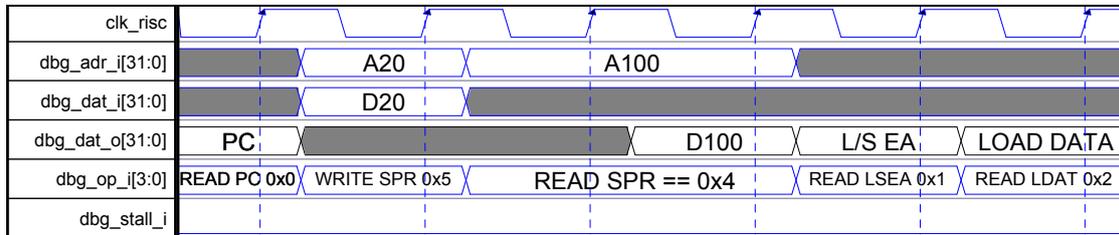


Figure 12. Development Interface Cycles

Figure 12 に、開発インターフェース (development interface) のサイクルについて示します。ライト (Write) は、メイン RISC クロックの立ち上がりエッジに同期し、1クロック・サイクルの間データを保持する必要があります。リード (Read) は、キャッシュ・ラインに同期したアクセスとなることから、2クロック・サイクルが必要となります。また、TLB エントリの参照のために1クロック・サイクル分の遅延が必要となります。

必要であれば、外部デバッガは、dbg_stall_i 信号をアサートすることで、CPU コアを停止させることができます。この方法により、参照したい全てのレジスタの値をRISCコアからリードするのに十分な時間を確保することができ、また、RISCコアによる同一レジスタへの書き込みを待つことなく、SPRへの書き込みを実行することができます。

Tracking Data Flow

外部デバッガは、デバッグもしくはプロファイル解析のためにRISCの内部のデータ・フローをモニターし記録することができます。これは、ロード・ストア・ユニット、ロード・ストア論理アドレスとロード・ストア・データを含む、開発インターフェース (development interface) 上の状態をモニターすることで実現されます。

dbg_lss_o[3:0]	Load/Store Instruction in Execution
0x0	No load/store instruction in execution
0x1	Reserved for load doubleword

0x2	Load byte and zero extend
0x3	Load byte and sign extend
0x4	Load halfword and zero extend
0x5	Load halfword and sign extend
0x6	Load singleword and zero extend
0x7	Load singleword and sign extend
0x8	<i>Reserved for store doubleword</i>
0x9	<i>Reserved</i>
0xA	Store byte
0xB	<i>Reserved</i>
0xC	Store halfword
0xD	<i>Reserved</i>
0xE	Store singleword
0xF	<i>Reserved</i>

Table 15. Status of the Load/Store Unit

外部トレース・バッファは、`dbg_iss_o` 信号上のロード・ストア・ユニットの動作及び状態を解析することで、目的とする全てのデータ・フロー・イベントを記録することができます。Table 15 は、ロード・ストア・ユニットの各状態と対応する `dbg_iss_o` 信号の値です。

Tracking Program Flow

外部デバッガは、デバッグもしくはプロファイル解析のためにRISCの内部のプログラム・フローをモニターし記録することができます。これは、命令ユニット、プログラム・カウンタ（PC）、フェッチ命令語を含む、開発インターフェース（development interface）上の状態をモニターすることで実現されます。

<code>dbg_is_o[1:0]</code>	Instruction Fetch Status
0x0	No instruction fetch in progress
0x1	Normal instruction fetch
0x2	Executing branch instruction
0x3	Fetching instruction in delay slot

Table 16. Status of the Instruction Unit

External trace buffer can capture all interesting program flow events by analyzing status of the instruction unit available on `dbg_is_o`. Table 16 lists different status encoding for the instruction unit.

外部トレース・バッファは、`dbg_is_o` 信号上の命令ユニットの動作及び状態を解析することで、目的とする全てのプログラム・フロー・イベントを記録することができます。Table 16 は、命令ユニットの各状態と対応する `dbg_is_o` 信号の値です。

Triggering External Watcpoint Event

Figure 13 に、開発インターフェース (development interface) がどのように `dbg_ewt_i` をアサートし、ウォッチポイント・イベント (Watchpoint event) を発生させるか示します。プログラムされた場合には、外部のウォッチポイント・イベントはブレイクポイント例外 (breakpoint exception) を発生させます。

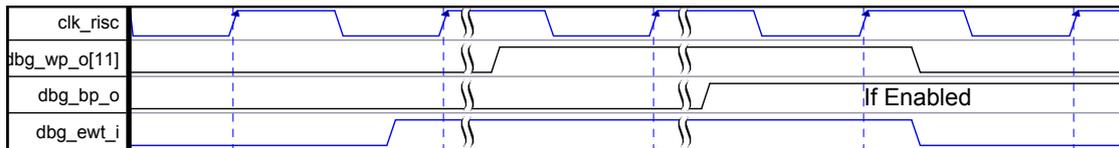


Figure 13. Assertion of External Watchpoint Trigger

4

Registers

この章は、OR1200 コア内部の全てのレジスタを説明します。下記の表の“GPR”番号を左に11ビットシフトし”REG”番号を加えることで、各Special-Purpose Register (SPR) のアドレスを算出します。USER MODEとSUPV MODEの欄は、ユーザー・モードとスーパーバイザー・モードにおける、各レジスタへの有効なアクセス・タイプを示しています。”R/W”は、リードとライト・アクセスを示し、”R”はリード・アクセスのみ可能であることを示しています。

Registers list

GRP #	REG #	REG NAME	USER MODE	SUPV MODE	DESCRIPTION
0	0	VR	–	R	Version Register
0	1	UPR	–	R	Unit Present Register
0	2	CPUCFGR	–	R	CPU Configuration Register
0	3	DMMUCFGR	–	R	Data MMU Configuration Register
0	4	IMMUCFGR	–	R	Instruction MMU Configuration Register
0	5	DCCFGR	–	R	Data Cache Configuration Register
0	6	ICCFGR	–	R	Instruction Cache Configuration Register
0	7	DCFGR	–	R	Debug Configuration Register
0	16	PC	–	R/W	PC mapped to SPR space
0	17	SR	–	R/W	Supervision Register
0	32	EPCR0	–	R/W	Exception PC Register
0	48	EEAR0	–	R/W	Exception EA Register
0	64	ESR0	–	R/W	Exception SR Register
0	1024-1055	GPR0-GPR31	–	R/W	GPRs mapped to SPR space
1	2	DTLBEIR	–	W	Data TLB Entry Invalidate Register
1	1024-1151	DTLBW0MR0-DTLBW0MR127	–	R/W	Data TLB Match Registers Way 0

1	1536-1663	DTLBW0TR0-DTLBW0TR127	–	R/W	Data TLB Translate Registers Way 0
2	2	ITLBEIR	–	W	Instruction TLB Entry Invalidate Register
2	1024-1151	ITLBW0MR0-ITLBW0MR127	–	R/W	Instruction TLB Match Registers Way 0
2	1536-1663	ITLBW0TR0-ITLBW0TR127	–	R/W	Instruction TLB Translate Registers Way 0
3	0	DCCR	–	R/W	DC Control Register
3	1	DCBIR	–	W	DC Block Invalidate Register
3	257	DCBFR	W	W	DC Block Flush Register
4	0	ICCR	–	R/W	IC Control Register
4	256	ICBIR	W	W	IC Block Invalidate Register
5	256	MACLO	R/W	R/W	MAC Low
5	257	MACHI	R/W	R/W	MAC High
6	16	DMR1	–	R/W	Debug Mode Register 1
6	17	DMR2	–	R/W	Debug Mode Register 2
6	20	DSR	–	R/W	Debug Stop Register
6	21	DRR	–	R/W	Debug Reason Register
8	0	PMR	–	R/W	Power Management Register
9	1	PICMR	–	R/W	PIC Mask Register
9	2	PICPR	–	R/W	PIC Priority Register
9	3	PICSR	–	R/W	PIC Status Register
10	0	TTMR	–	R/W	Tick Timer Mode Register
10	256	TTCR	R*	R/W	Tick Timer Count Register

Table 17. List of All Registers

Table 17 に、OpenRISC 1000 Special-Purpose Register のうち OR1200 に実装されているものを示します。レジスタ VR と UPR について、以下に説明します。その他のレジスタについては、*OpenRISC 1000 System Architecture Manual* を参照してください。

Register VR description

Special-Purpose Register VR は、OpenRISC 1000 プロセッサのバージョン及びリビジョンを示します。これは、また、プロセッサがどの実装をベースとしているかを示しています。

Bit #	Access	Reset	Description
5:0	R	Revision	REV Revision number
15:6	R	0x0	Reserved

23:16	R	0x00	CFG Configuration should be read from UPR and configuration registers
31:24	R	0x12	VER Version number for OR1200 is fixed at 0x1200.

Table 18. VR Register

Register UPR description

Special-Purpose Register UPR は、プロセッサ内部に実装されたユニットを示します。実装された各ユニット及び機能を示す各ビットを持ちます。下位の 16 ビットは OpenRISC 1000 アーキテクチャーにて定義されているユニットのうち実装されているものについて示しています。上位の 16 ビットは、実装されているカスタム・ユニットについて示しています。

Bit #	Access	Reset	Description
0	R	1	UP UPR present
1	R	1	DCP Data cache present
2	R	1	ICP Instruction cache present
3	R	1	DMP Data MMU present
4	R	1	IMP Instruction MMU present
5	R	1	MP MAC present
6	R	1	DUP Debug unit present
7	R	0	PCUP Performance counters unit not present
8	R	1	PMP Power Management Present
9	R	1	PICP Programmable interrupt controller present
10	R	1	TTP Tick timer present
23:11	R	X	Reserved
31:24	R	0XXXXX	CUP The user of the OR1200 core adds custom units.

Table 19. UPR Register

Register CPUCFGR description

Special-Purpose Register CPUCFGR は、CPUの機能(capabilities)及び構成(configuration)を示しています。

Bit #	Access	Reset	Description
3:0	R	0x0	NSGF Zero number of shadow GPR files
4	R	0	HGF No half GPR files
5	R	1	OB32S ORBIS32 supported
6	R	0	OB64S ORBIS64 not supported
7	R	0	OF32S ORFPX32 not supported
8	R	0	OF64S ORFPX64 not supported
9	R	0	OV64S ORVDX64 not supported

Table 20. CPUCFGR Register

Register DMMUCFGR description

Special-Purpose Register DMMUCFGR は、DMMU の機能(capabilities)及び構成(configuration)を示しています。

Bit #	Access	Reset	Description
1:0	R	0x0	NTW One DTLB way
4:2	R	0x4 – 0x7	NTS 16, 32, 64 or 128 DTLB sets
7:5	R	0x0	NAE No ATB Entries
8	R	0	CRI No DMMU control register implemented
9	R	0	PRI No protection register implemented
10	R	1	TEIRI DTLB entry invalidate register implemented
11	R	0	HTR

		No hardware DTLB reload
--	--	-------------------------

Table 21. DMMUCFGR Register

Register IMMUCFGR description

Special-Purpose Register IMMUCFGR は、IMMU の機能(capabilities)及び構成(configuration)を示しています。

Bit #	Access	Reset	Description
1:0	R	0x0	NTW One ITLB way
4:2	R	0x4 – 0x7	NTS 16, 32, 64 or 128 ITLB sets
7:5	R	0x0	NAE No ATB Entries
8	R	0	CRI No IMMU control register implemented
9	R	0	PRI No protection register implemented
10	R	1	TEIRI ITLB entry invalidate register implemented
11	R	0	HTR No hardware ITLB reload

Table 22. IMMUCFGR Register

Register DCCFGR description

Special-Purpose Register DCCFGR は、データ・キャッシュの機能(capabilities)及び構成(configuration)を示しています。

Bit #	Access	Reset	Description
2:0	R	0x0	NCW One DC way
6:3	R	0x4 – 0x7	NCS 16, 32, 64 or 128 DC sets
7	R	0x0	CBS 16-byte cache block size
8	R	0	CWS

			Cache write-through strategy
9	R	1	CCRI DC control register implemented
10	R	1	CBIRI DC block invalidate register implemented
11	R	0	CBPRI DC block prefetch register not implemented
12	R	0	CBLRI DC block lock register not implemented
13	R	1	CBFRI DC block flush register implemented
14	R	0	CBWBRI DC block write-back register not implemented

Table 23. DCCFGR Register

Register ICCFGR description

Special-Purpose Register ICCFGR は、命令キャッシュの機能(capabilities)及び構成(configuration)を示しています。

Bit #	Access	Reset	Description
2:0	R	0x0	NCW One IC way
6:3	R	0x4 – 0x7	NCS 16, 32, 64 or 128 IC sets
7	R	0x0	CBS 16-byte cache block size
8	R	0	CWS Cache write-through strategy
9	R	1	CCRI IC control register implemented
10	R	1	CBIRI IC block invalidate register implemented
11	R	0	CBPRI IC block prefetch register not implemented
12	R	0	CBLRI IC block lock register not implemented
13	R	1	CBFRI IC block flush register implemented
14	R	0	CBWBRI IC block write-back register not implemented

Table 24. ICCFGR Register

Register DCFGR description

Special-Purpose Register DCFGR は、デバック・ユニットの機能(capabilities)及び構成(configuration)を示しています。

Bit #	Access	Reset	Description
2:0	R	0x0	NDP Zero DVR/DCR pairs
3	R	0	WPCI Watchpoint counters not implemented

Table 25. DCFGR Register

5

IO ports

OR1200 IP コアは様々なインターフェースを持ちます。Figure 14 に次に示す全てのインターフェースを示します。

- 命令及びデータ・インターフェース
(Instruction and data WISHBONE host interfaces)
- 電力制御インターフェース (Power management interface)
- 開発インターフェース (Development interface)
- 割り込みインターフェース (Interrupts interface)

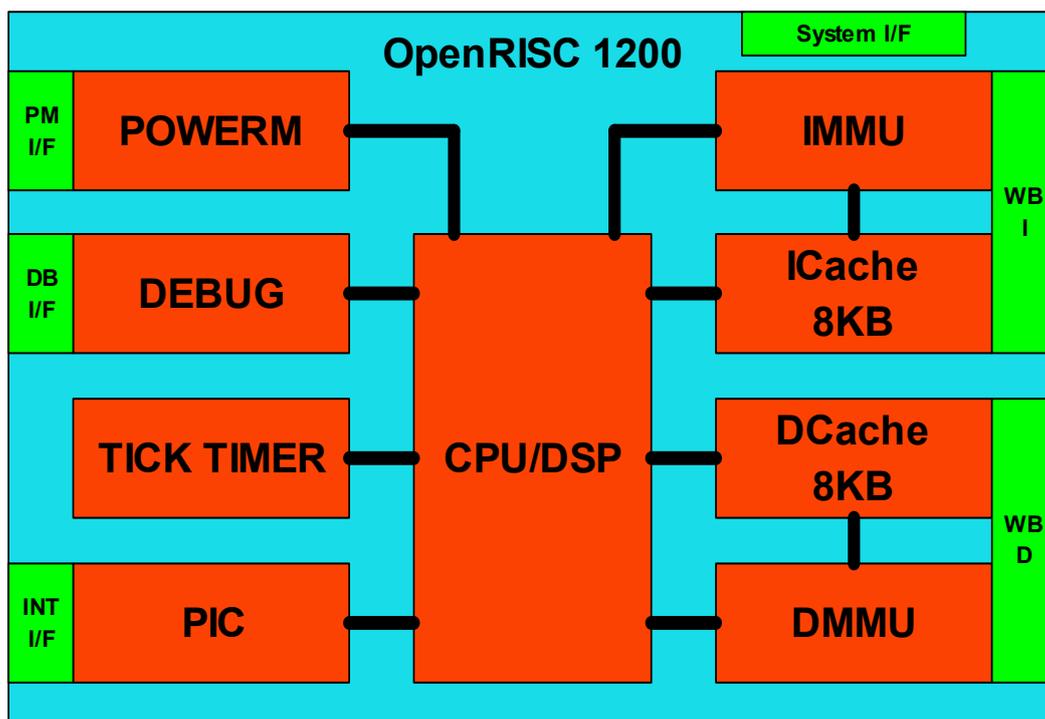


Figure 14. Core's Interfaces

Instruction WISHBONE Master Interface

OR1200 は、WISHBONE Rev B 準拠の2つのマスター・インターフェースを持ちます。命令インターフェースは、命令及び命令キャッシュ・ラインをフェッチする目的で、OR1200 コアとメモリ・サブシステムを接続するために使用されます。

Port	Width	Direction	Description
iwb_CLK_I	1	Input	Clock input
iwb_RST_I	1	Input	Reset input
iwb_CYC_O	1	Output	Indicates valid bus cycle (core select)
iwb_ADR_O	32	Outputs	Address outputs
iwb_DAT_I	32	Inputs	Data inputs
iwb_DAT_O	32	Outputs	Data outputs
iwb_SEL_O	4	Outputs	Indicates valid bytes on data bus (during valid cycle it must be 0xf)
iwb_ACK_I	1	Input	Acknowledgment input (indicates normal transaction termination)
iwb_ERR_I	1	Input	Error acknowledgment input (indicates an abnormal transaction termination)
iwb_RTY_I	1	Input	In OR1200 treated same way as iwb_ERR_I.
iwb_WE_O	1	Output	Write transaction when asserted high
iwb_STB_O	1	Outputs	Indicates valid data transfer cycle

Table 26. Instruction WISHBONE Master Interface' Signals

Data WISHBONE Master Interface

OR1200 は、WISHBONE Rev B 準拠の2つのマスター・インターフェースを持ちます。データ・インターフェースは、データ及びデータ・キャッシュ・ラインをリード・ライトする目的で、OR1200 コアと外部のペリフェラル (external peripherals) 及びメモリ・サブシステムを接続するために使用されます。

Port	Width	Direction	Description
dwb_CLK_I	1	Input	Clock input
dwb_RST_I	1	Input	Reset input
dwb_CYC_O	1	Output	Indicates valid bus cycle (core select)
dwb_ADR_O	32	Outputs	Address outputs
dwb_DAT_I	32	Inputs	Data inputs
dwb_DAT_O	32	Outputs	Data outputs
dwb_SEL_O	4	Outputs	Indicates valid bytes on data bus (during valid cycle it must be 0xf)
dwb_ACK_I	1	Input	Acknowledgment input (indicates normal transaction termination)
dwb_ERR_I	1	Input	Error acknowledgment input (indicates an abnormal transaction termination)
dwb_RTY_I	1	Input	In OR1200 treated same way as dwb_ERR_I.
dwb_WE_O	1	Output	Write transaction when asserted high
dwb_STB_O	1	Outputs	Indicates valid data transfer cycle

Table 27. Data WISHBONE Master Interface' Signals

System Interface

システム・インターフェースは、リセット・クロック及びその他のシステム信号と、OR1200 コアを接続します。

Port	Width	Direction	Description
Rst	1	Input	Asynchronous reset
clk_cpu	1	Input	Main clock input to the RISC
clk_dc	1	Input	Data cache clock
clk_ic	1	Input	Instruction cache clock
clk_dmmu	1	Input	Data MMU clock
clk_immu	1	Input	Instruction MMU clock
clk_tt	1	Input	Tick timer clock

Table 28. System Interface Signals

Development Interface

開発インターフェース (Development Interface) は、外部の開発ポートとRISCコア内部のデバッグ関連機能を接続します。デバッグ関連機能により、RISC内部のプログラムの実行制御、ブレークポイントとウォッチポイントの設定、命令及びデータ・フローのトレース (tracing) が可能となります。

Port	Width	Direction	Description
dbg_dat_o	32	Output	Transfer of data from RISC to external development interface
dbg_dat_i	32	Input	Transfer of data from external development interface to RISC
dbg_adr_i	32	Input	Address of special-purpose register to be read or written
dbg_op_I	3	Input	Operation select for development interface
dbg_lss_o	4	Output	Status of load/store unit
dbg_is_o	2	Output	Status of instruction fetch unit
dbg_wp_o	11	Output	Status of watchpoints
dbg_bp_o	1	Output	Status of the breakpoint
dbg_stall_i	1	Input	Stalls RISC CPU core
dbg_ewt_i	1	Input	External watchpoint trigger

Table 29. Development Interface

Power Management Interface

電力制御 (Power management) インターフェースは、外部の電力制御回路と RISC コアを接続する信号を提供します。外部の電力制御回路は、テクノロジー・スペシフィック (Technology Specific) で OR1200 コア内部に実装することができない機能を持つことが要求されます。

Port	Width	Direction	Generation	Description
pm_clk_sd	4	Output	Static (in SW)	Slow down outputs that control reduction of RISC clock frequency
pm_cpustall	1	Input	-	Synchronous stall of the RISC's CPU core
pm_dc_gate	1	Output	Dynamic (in HW)	Gating of data cache clock
pm_ic_gate	1	Output	Dynamic (in HW)	Gating of instruction cache clock
pm_dmmu_gate	1	Output	Dynamic (in HW)	Gating of data MMU clock
pm_immu_gate	1	Output	Dynamic (in HW)	Gating of instruction MMU clock
pm_tt_gate	1	Output	Dynamic (in HW)	Gating of tick timer clock
pm_cpu_gate	1	Output	Static (in SW)	Gating of main CPU clock
pm_wakeup	1	Output	Dynamic (in HW)	Activate all clocks
pm_lvolt	1	Output	Static (in SW)	Lower voltage

Table 30. Power Management Interface

Interrupt Interface

割り込みインターフェース (Interrupt Interface) は、外部のペリフェラル割り込み出力信号と RISC コアを接続する割り込み信号を持ちます。全ての割り込み入力は、メイン RISC クロックの立ち上がりエッジで評価されます。

Port	Width	Direction	Description
pic_ints	PIC_INTS	Input	External interrupts

Table 31. Interrupt Interface

A

Core HW Configuration

この章では、ユーザーによって設定可能な、コアのコンフィギュレーション・パラメータについて説明しています。パラメータは、実際にコアを使用したシミュレーションや論理合成を実施する前に、ユーザーによって設定される必要があります。

Variable Name	Range	Default	Description
EADDR_WIDTH	32	32	Effective address width
VADDR_WIDTH	32	32	Virtual address width
PADDR_WIDTH	24 – 36	32	Physical address width
DATA_WIDTH	32	32	Data width / Operation width
DC_IMPL	0 – 1	1	Data cache implementation
DC_SETS	512	512	Data cache number of sets
DC_WAYS	1	1	Data cache number of ways
DC_LINE	16	16	Data cache line size
IC_IMPL	0 – 1	1	Instruction cache implementation
IC_SETS	512	512	Instruction cache number of sets
IC_WAYS	1	1	Instruction cache number of ways
IC_LINE	16	16	Instruction cache line size in bytes
DMMU_IMPL	0 – 1	1	Data MMU implementation
DTLB_SETS	64	64	Data TLB number of sets
DTLB_WAYS	1	1	Data TLB number of ways
IMMU_IMPL	0 – 1	1	Instruction MMU implementation
ITLB_SETS	64	64	Instruction TLB number of sets
ITLB_WAYS	1	1	Instruction TLB number of ways
PIC_INTS	2 – 32	30	Number of interrupt inputs