OpenRISC 1200 IP Core Specification

Author: Damjan Lampret
lampret@opencores.org

Rev. 0.2
April 16, 2001

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

Table Of Contents

Table Of Figures

Table Of Tables

1

Introduction

Purpose of this document is to define specifications of the OpenRISC 1200 implementation. This specification defines all implementation specific variables that are not part of the general architecture specification. This includes type and size of data and instruction caches, type and size of data and instruction MMUs, details of all execution pipelines, implementation of exception unit, interrupt controller and other supplemental units.
This document does not cover general architecture topics like instruction set, memory addressing modes and other architectural definitions. See *OpenRISC 1000 System Architecture Manual* for more information about architecture.

The OR1200 is a 32-bit scalar RISC with Harvard microarchitecture, 5 stage integer pipeline, virtual memory support (MMU) and basic DSP capabilities.
Default caches are 2-way set-associative 8KB data cache and 2-way set-associative 8KB instruction cache, each with 16-byte line size. Both caches are physically tagged.
By default MMUs are implemented and they are constructed of 64-entry hash based 2-way set-associative data TLB and 64-entry hash based 2-way set-associative instruction TLB.
Supplemental facilities include debug port for real-time debugging, high resolution tick timer, programmable interrupt controller and power management support.
When implemented in a typical 0.18u 6LM process it should provide over 400 dhrystone 2.1 MIPS at 400MHz and 400 DSP MAC 32x32 operations, at least 30% more than any other competitor in this class. OR1200 in default configuration has about 1M transistors.

OR1200 is intended for embedded, portable and networking applications. It can successfully compete with latest scalar 32-bit RISC processors in his class and can efficiently run any modern operating system.
Competitors include ARM10, ARC and Tensilica RISC processors.

Features

The following lists the main features of OR1200 IP core:

- All major characteristics of the core can be set by the user
- High performance of 400 Dhrystone 2.1 MIPS at 400 MHz using 0.18u process
- High performance cache and MMU subsystems
- WISHBONE SoC Interconnection Rev. B compliant interface

2

Architecture

Figure 0-1 below shows general architecture of OR1200 IP core. It consists of several building blocks:

- CPU/DSP central block
- N-way set-associative data cache
- N-way set-associative 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 debug interface
- Interrupt controller and interrupt interface
- Instruction and Data WISHBONE host interfaces

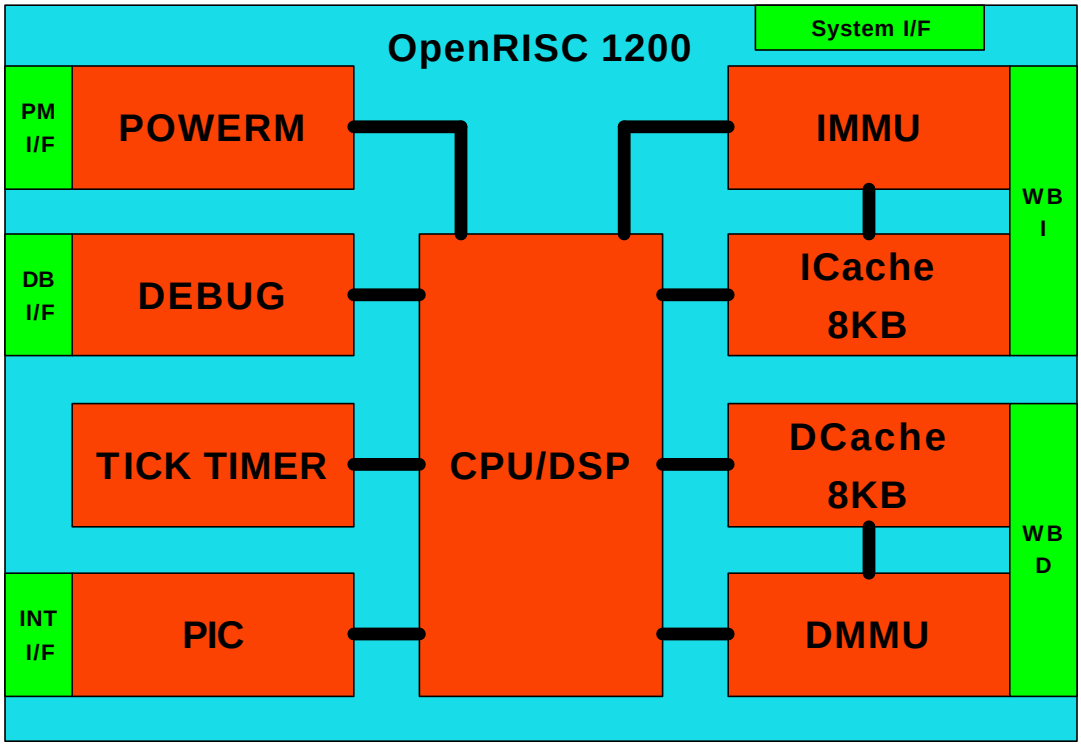


Figure 0-1. Core's Architecture

CPU/DSP

CPU/DSP is a central part of the OR1200 RISC processor. Figure 0-2 shows basic block diagram of the CPU/DSP.
OR1200 CPU/DSP implements only 32-bit part of the OpenRISC 1000 architecture. 64-bit part of the architecture as well as floating-point and vector operations are not implemented in OR1200.

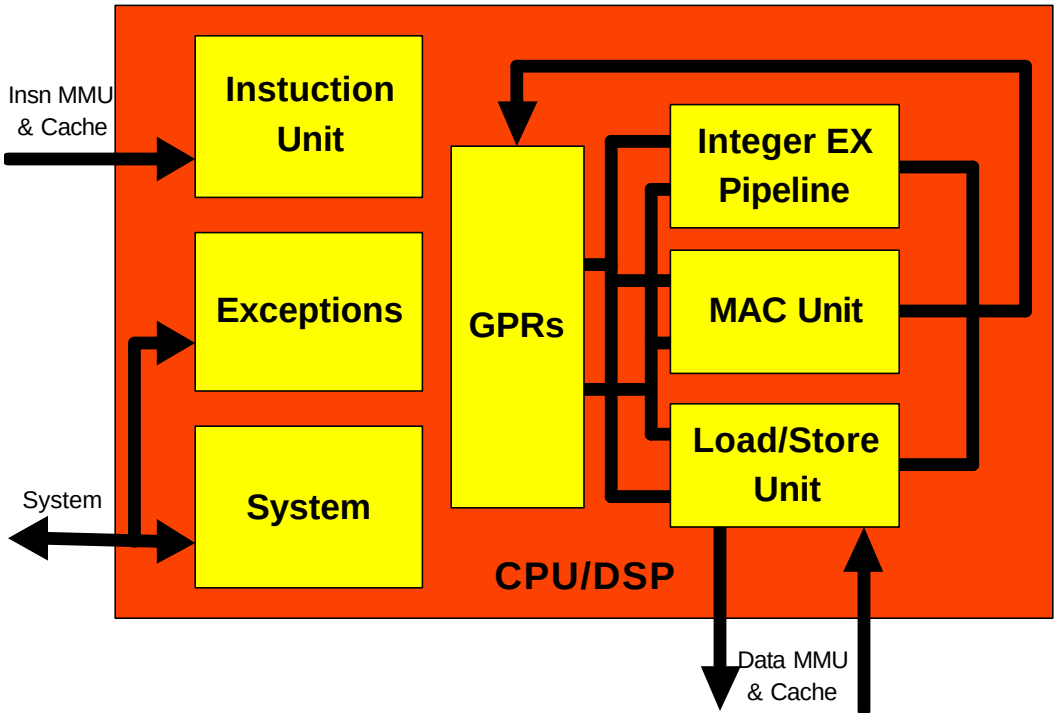


Figure 0-2. CPU/DSP Block Diagram

Instruction unit

The instruction unit implements the basic instruction pipeline, fetches instructions from the memory subsystem, dispatches them to available execution units, and maintains a state history to ensure a precise exception model and that operations finish in order. It also executes conditional branch and unconditional jump instructions.

The sequencer can dispatch a sequential instruction on each clock if the appropriate execution unit is available. The execution unit must discern whether source data is available and to ensure that no other instruction is targeting the same destination register.

Instruction unit handles only ORBIS32 instruction class. ORFPX32/64 and ORVDX64 instruction classes are not supported by current OR1200.

General-Purpose Registers

OpenRISC 1200 implements 32 general-purpose 32-bit registers. OpenRISC 1000 architecture also support shadow copies of register file to implement fast switching between working contexts, however this feature is not implemented in current OR1200 implementation.

Load/Store Unit

The load/store unit (LSU) transfers all data between the GPRs and the CPU's internal bus. It is implemented as an independent execution unit so that stalls in memory subsystem only affect master pipeline if there is a data dependency.

The following are LSU's main features:

- all load/store instruction implemented in hardware
- address entry buffer
- pipelined operation
- aligned accesses for fast memory access

When load and store instructions are issued, the LSU determines if all operands are available. These operands include the following:

- address register operand
- source data register operand (for store instructions)
- destination data register operand (for load instructions)

Integer Execution Pipeline

The core implements the following types of 32-bit integer instructions:

- Arithmetic instructions
- Compare instructions
- Logical instructions
- Rotate and shift instructions

Most integer instructions can execute in one cycle. For details about timing see table TBD.

MAC Unit

The MAC unit executes DSP MAC operations. MAC operations are 32x32 with 48-bit accumulator. MAC unit is fully pipelined and can accept new MAC operation in each new clock cycle.

System Unit

The system unit connects all other signals of the CPU/DSP that are not connected through instruction and data interfaces. It also implements all system special-purpose registers (e.g. supervisor register).

Exceptions

Core exceptions can be generated when an exception condition occurs. Exception sources in OR1200 include the following:

- 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

Exception handling is transparent to user software and uses the same mechanism to handle all types of exceptions. When an exception is taken, control is transferred to an exception handler at an offset defined by for the type of exception encountered. Exceptions are handled in supervisor mode.

The core implements a precise exception model. This means that when an exception is taken, the following conditions are met:

- Subsequent instructions in program flowe are discarded
- Previous instructions finish and write back their results
- The address of faulting instruction is saved in EPCR registers and the machine state is saved to ESR registers

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 0-1. List of Implemented Exceptions

The OR1200 exception support does not include support for fast context switching.

Data Cache

The default configuration of OR1200 data cache is 8-Kbyte, two-way set associative data cache, which allows rapid core access to data. However data cache can be configured according to the Table 0-2.

	Direct mapped	2-way set associative	4-way set associative	8-way set associative
1KB per set	1KB	2KB	4KB	8KB

2KB per set	2KB	4KB	8KB	16KB
4KB per set	4KB	**8KB (default)**	16KB	32KB
8KB per set	8KB	16KB	32KB	64KB

Table 0-2. Possible Data Cache Configurations of OR1200

Features:

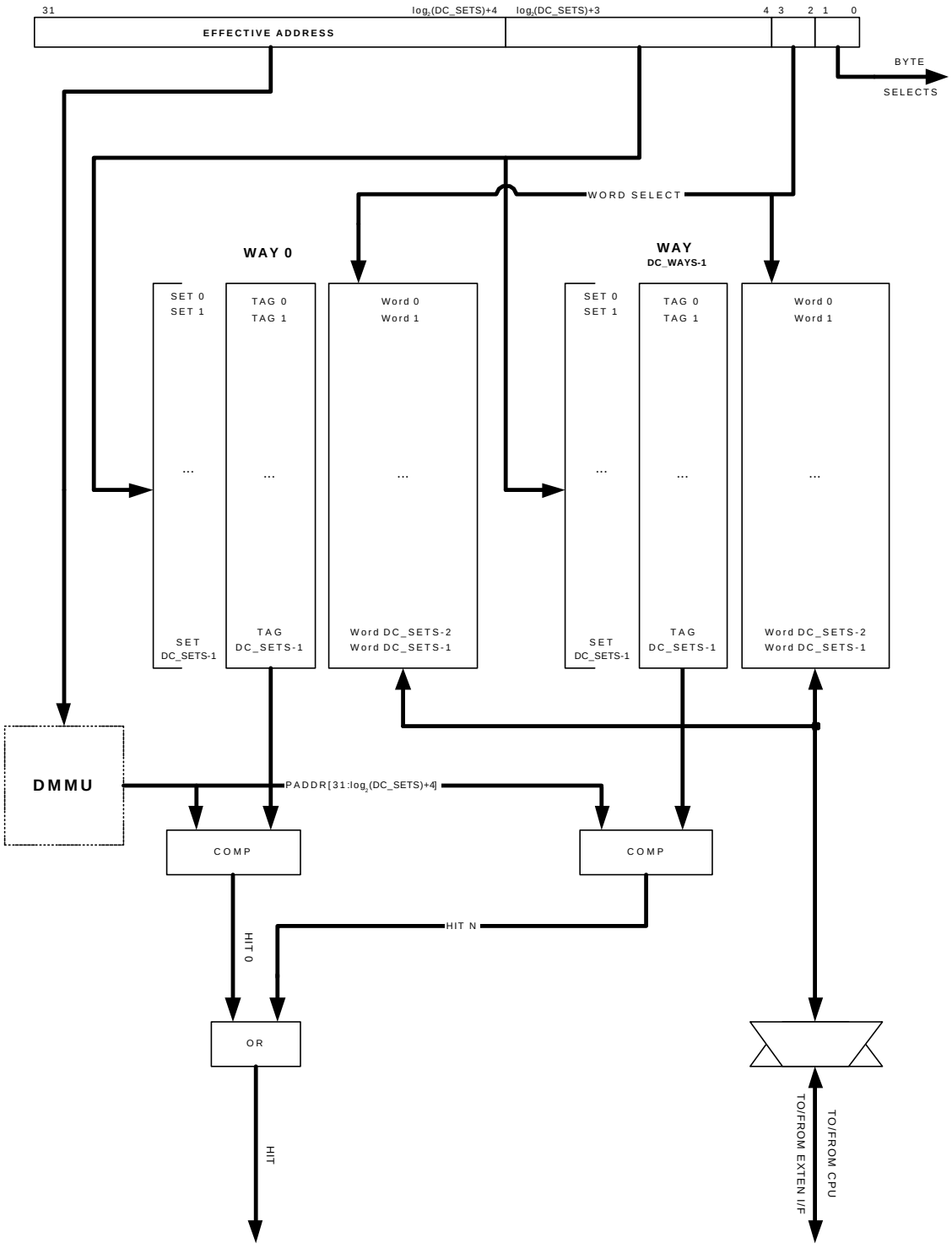
- data cache is separate from instruction cache (Harvard architecture)
- 1-4 way set associative with default of 2
- data cache implements a least-recently used (LRU) replacement algorithm within each set
- the cache directory is physically addressed. The physical address tag is stored in the cache directory
- write-through operation
- it can be disabled, invalidated or locked by writing to cache special purpose registers
- individual cache blocks can be locked so that frequently accessed data are guaranteed to be resident in the cache

On a miss, the cache is filled in with 16-byte bursts. The burst fill is performed as a critical-word-first operation; the critical word is simultaneously written to the cache and forwarded to the requesting unit, thus minimizing stalls due to cache fill latency. Data cache provides storage for cache tags and performs cache block replacement function.

Data cache is tightly coupled to external interface to allow efficient access to the system memory controller.

The data cache supplies data to the GPRs by means of a 32-bit interface to the load/store unit. The LSU provides all logic required to calculate effective addresses, handles data alignment to and from the data cache, and provides sequencing for load and store operations. Write operations to the data cache can be performed on a byte, half-word or word basis.

The data cache is organized as 256 sets of two blocks. Each block consists of 16 bytes, state bits and an address tag.



Each block contains four contiguous words from memory that are loaded from a four-word aligned boundary. As a result, cache blocks are aligned with page boundaries.

Instruction Cache

The default configuration of OR1200 instruction cache is 8-Kbyte, two-way set associative instruction cache, which allows rapid core access to instructions. However instruction cache can be configured according to the Table 0-3.

	Direct mapped	2-way set associative	4-way set associative	8-way set associative
1KB per set	1KB	2KB	4KB	8KB
2KB per set	2KB	4KB	8KB	16KB
4KB per set	4KB	**8KB (default)**	16KB	32KB
8KB per set	8KB	16KB	32KB	64KB

Table 0-3. Possible Instruction Cache Configurations of OR1200

Features:

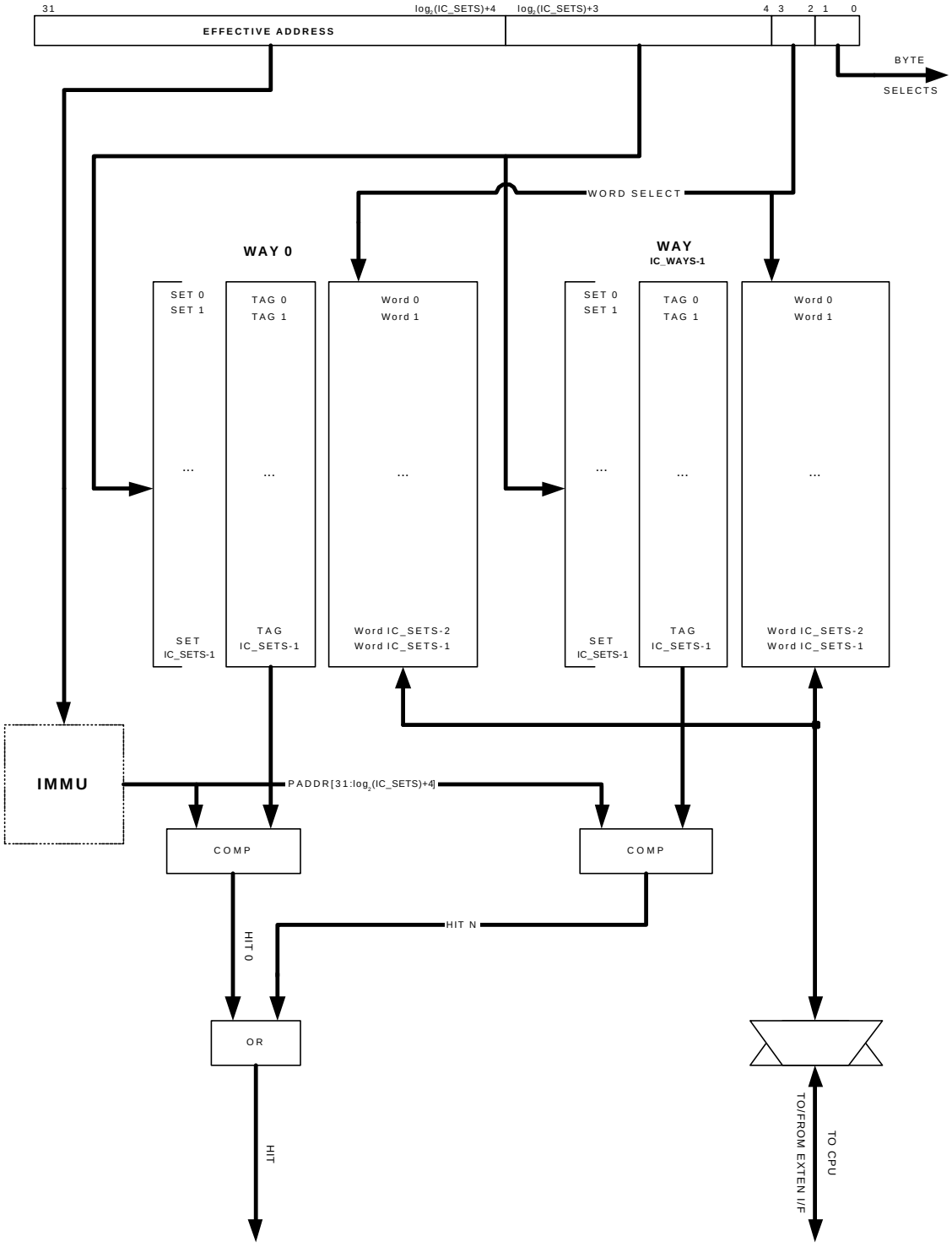
- instruction cache is separate from data cache (Harvard architecture)
- 1-4 way set associative with default of 2
- instruction cache implements a least-recently used (LRU) replacement algorithm within each set
- the cache directory is physically addressed. The physical address tag is stored in the cache directory
- it can be disabled, invalidated or locked by writing to cache special purpose registers
- individual cache blocks can be locked so that frequently accessed instructions are guranteed to be resident in the cache

On a miss, the cache is filled in with 16-byte bursts. The burst fill is performed as a critical-word-first operation; the critical word is simultaneously written to the cache and forwarded to the requesting unit, thus minimizing stalls due to cache fill latency. Instruction cache provides storage for cache tags and performs cache block replacement function.

Instruction cache is tightly coupled to external interface to allow efficient access to the system memory controller.

The instruction cache supplies instructions to the instruction sequencer by means of a 32-bit interface to the instruction fetch subunit. The instruction fetch subunit provides all logic required to calculate effective addresses.

The data cache is organized as 256 sets of two blocks. Each block consists of 16 bytes, state bits and an address tag.



Each block contains four contiguous words from memory that are loaded from a four-word aligned boundary. As a result, cache blocks are aligned with page boundaries.

Data MMU

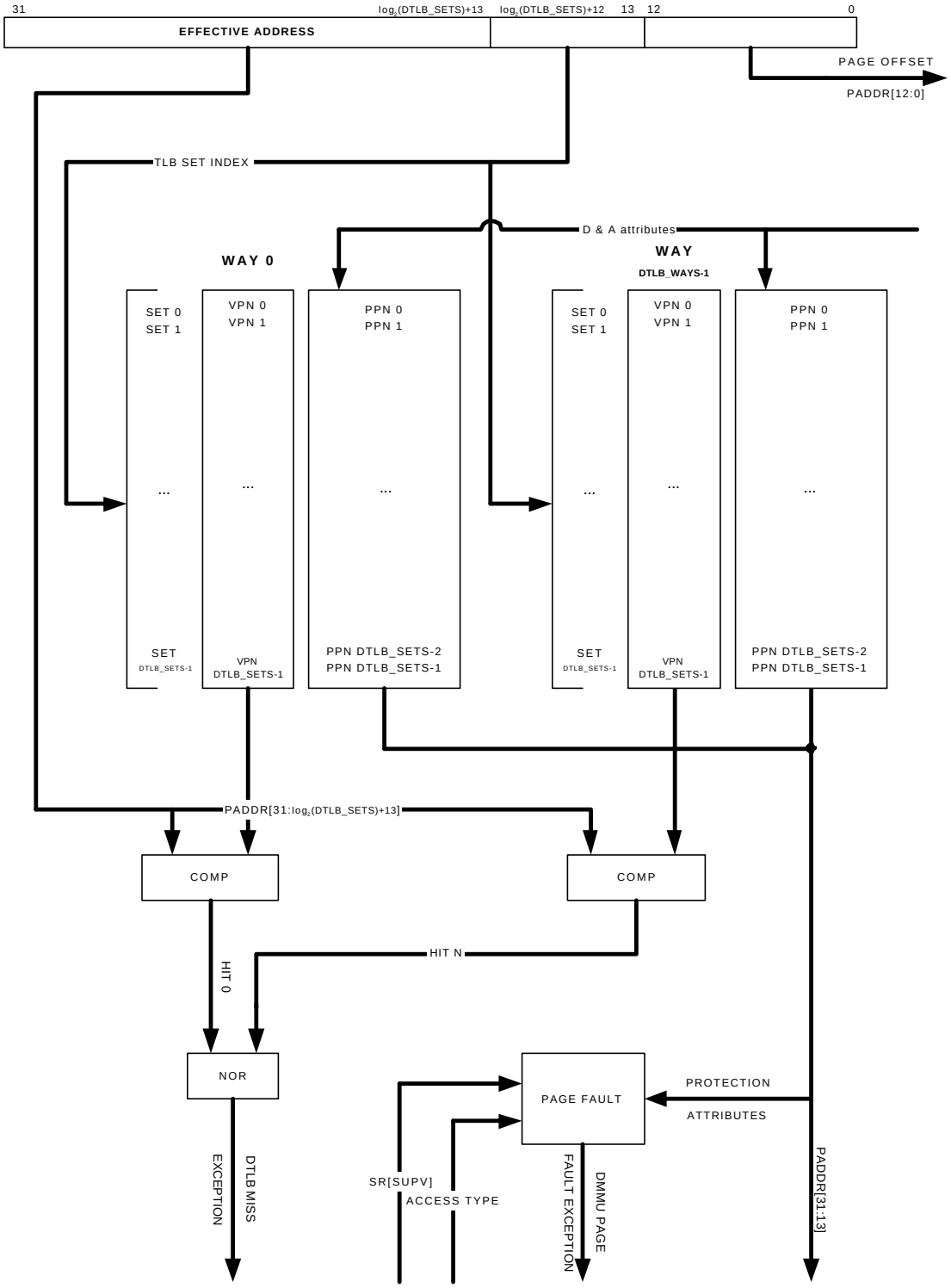
The OR1200 implements a virtual memory management scheme that provides cache control, memory access protection and effective-to-physical address translation. Protection granularity is as defined by OpenRISC 1000 architecture - 8-Kbyte and 16-Mbyte pages.

	Direct mapped	2-way set associative
16 entries per way	16 DTLB entries	32 DTLB entries
32 entries per way	32 DTLB entries	**64 DTLB entries (default)**
64 entries per way	64 DTLB entries	128 DTLB entries
128 entries per way	128 DTLB entries	256 DTLB entries

Table 0-4. Possible Data TLB Configurations of OR1200

Features:

- data MMU is separate from instruction MMU
- two pages sizes - 8-Kbyte and 16-Mbyte
  - dirty bit
  - accessed bit
  - caching-inhibited attribute
- comprehensive page protection scheme
- 1-2 way set associative hash based translation lookaside buffer (DTLB) with the default of 2 ways and the following features:
  - miss and fault exceptions
  - software tablewalk
  - locking of one DTLB sets to ensure fast translation of critical data
  - high performance because of hashed based design
  - variable number DTLB entries with default of 32 per each way



The MMU hardware supports two-level software tablewalk.

Instruction MMU

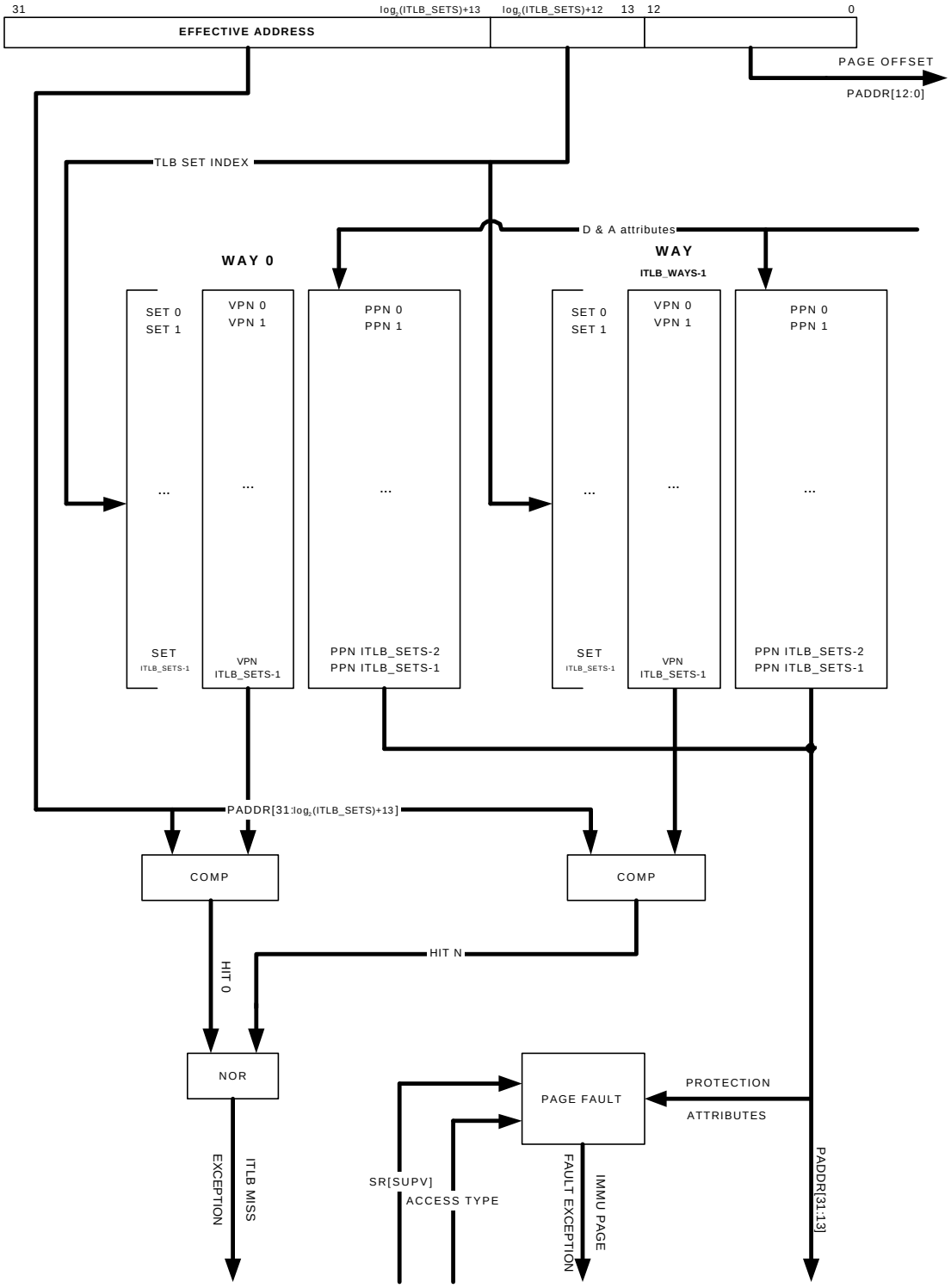
The OR1200 implements a virtual memory management scheme that provides cache control, memory access protection and effective-to-physical address translation. Protection granularity is as defined by OpenRISC 1000 architecture - 8-Kbyte and 16-Mbyte pages.

	Direct mapped	2-way set associative
16 entries per way	16 ITLB entries	32 ITLB entries
32 entries per way	32 ITLB entries	**64 ITLB entries (default)**
64 entries per way	64 ITLB entries	128 ITLB entries
128 entries per way	128 ITLB entries	256 ITLB entries

Table 0-5. Possible Instruction TLB Configurations of OR1200

Features:

- instruction MMU is separate from data MMU
- two pages sizes - 8-Kbyte and 16-Mbyte
  - accessed bit
- comprehensive page protection scheme
- 1-2 way set associative hash based translation lookaside buffer (ITLB) with the default of 2 ways and the following features:
  - miss and fault exceptions
  - software tablewalk
  - locking of one ITLB sets to ensure fast translation of critical data
  - high performance because of hashed based design
  - Variable number of ITLB entries with default of 32 entries per way



The MMU hardware supports two-level software tablewalk.

Programmable Interrupt Controller

The interrupt controller receives interrupts from external sources.

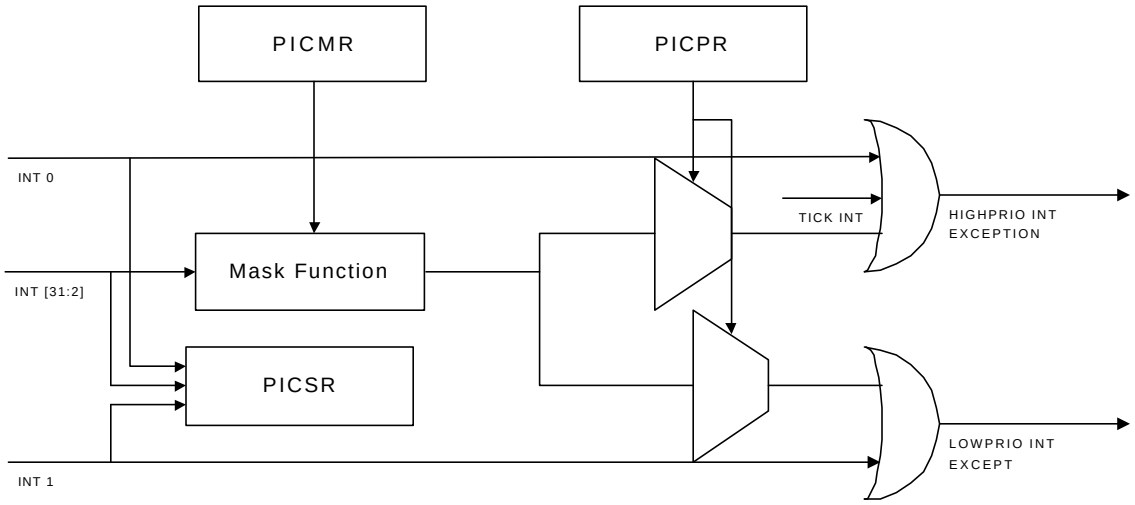


Table 0-6. Block Diagram of the Interrupt Controller

OR1200 can have up to 32 interrupt inputs that can be grouped into two priority groups or disabled on individual interrupt basis.

Tick Timer

OR1200 implements tick timer facility. Basically this is a timer that is clocked by RISC clock and is used by the operating system to precisely measure time and schedule system tasks.

Power Management Support

To optimize power consumption, the OR1200 provides low-power modes that can be used to dynamically activate and deactivate certain internal modules.

OR1200 has three major features to minimize power consumption:

- Slow and Idle Modes (SW controlled clock freq reduction)
- Doze and Sleep Modes (interrupt wake-up)
- Dynamic Clock gating (on clock by clock basis unit clock gating)

Modes summarized in a table with approx power consumption equations

Slow and idle modes take advantage of the low-power dividers in external clock generation circuitry to enable full functionality, but at a lower frequency so that a power consumption is reduced.

When software initiates the doze mode, software processing on the core suspends. The clocks to the RISC internal modules are disabled except to the RTC and internal timer. However any external modules continue to function as normal.
The OR1200 will leave doze mode and enter normal mode when a pending interrupt from an external peripheral occurs.

In sleep mode, the only internal modules that are activated are the real-time clock and periodic timer.
The OR1200 will leave sleep mode and enter normal mode when a pending interrupt from a RTC or periodic timer occurs.

If enabled, the clock gating feature automatically disables clock subtrees to major RISC internal blocks on a clock cycle basis. These blocks are CPU, FPU/VU, IC, DC, IMMU and DMMU. This mode can be used in a combination with other low-power modes.
Cache or MMU blocks that are already disabled when software enables this mode, have completely disabled clock subtrees until clock gating is disabled or until the blocks are again enabled.

Debug unit

Debug unit assists software developers to debug their systems. It provides support for watchpoints, breakpoints and program-flow control registers.

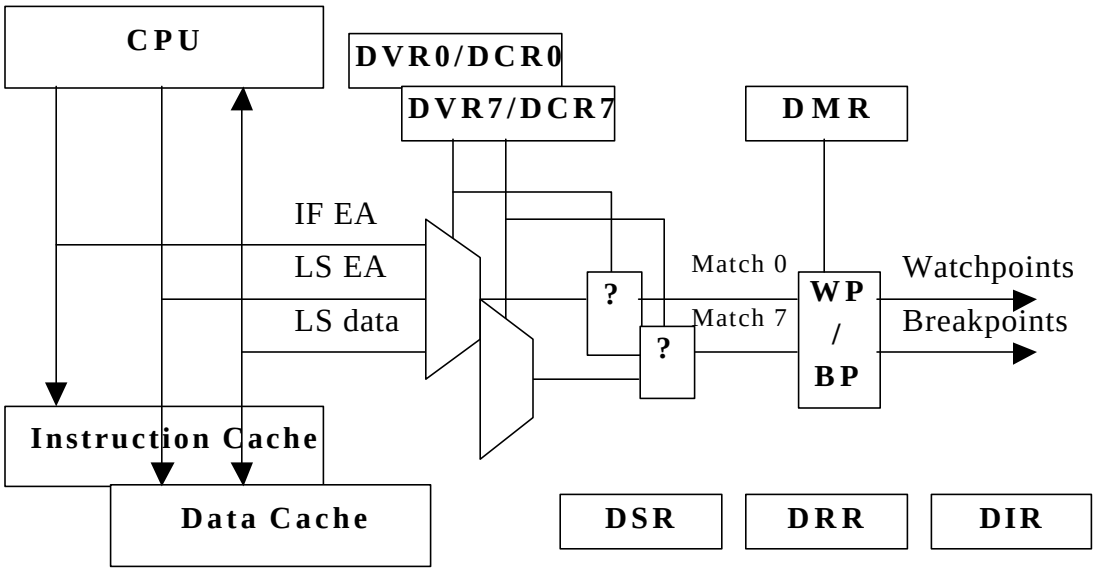


Figure 0-3. Block Diagram of Debug Unit

Watchpoints and breakpoints are events triggered by program- or data-flow matching the conditions programmed in the debug registers. Breakpoints unlike watchpoints also suspend execution of the current program-flow and start breakpoint exception.

Clocks & Reset

The OR1200 core has several clock inputs. Clock input clk_cpu clocks CPU/DSP block and all other parts of the RISC that do not have separate clocks. Data cache is clocked by clk_dc, instruction cache is clocked by clk_ic, data MMU is clocked by clk_dmmu and instruction MMU is clocked by clk_immu. All clocks must have the same phase and as low clock skew as possible.

OR1200 has asynchronous reset signal. Reset signal *arst*, when asserted high, immediately resets all flip-flops inside OR1200. When deasserted, OR1200 will start reset exception.

WISHBONE Interfaces

WISHBONE interfaces connect OR1200 core to external peripherals and external memory subsystem. They are WISHBONE SoC Interconnection specification Rev. B compliant. The implementation implements a 32-bit bus width and does not support other bus widths.


WISHBONE
COMPATIBLE

3

Operation

This section describes the operation of the OR1200 core.

Hardware Reset

CPU/DSP

CPU/DSP is implementation of the 32-bit part of the OpenRISC 1000 architecture.

Insn	31	31	31	31	31	26	25	25	25	25	21	20	20	20	20	16	15	15	15	15	11	10	10	8	7		6	5		4	3	3	3	0
l.add	opcode 0x38						D					A					B					reserved			opcode 0x0			reserved			opcode 0x0			
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7		6	5		4	3	2	1	0
l.addc	opcode 0x38						D					A					B					reserved			opcode 0x0			reserved			opcode 0x1			
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7		6	5		4	3	2	1	0
l.addi	opcode 0x25						D					A					I																	
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7		6	5		4	3	2	1	0
l.and	opcode 0x38						D					A					B					reserved			opcode 0x0			reserved			opcode 0x3			
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7		6	5		4	3	2	1	0
l.andi	opcode 0x28						D					A					K																	
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7		6	5		4	3	2	1	0
l.bf	opcode 0x4						N																											
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7		6	5		4	3	2	1	0
l.bnf	opcode 0x3						N																											
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7		6	5		4	3	2	1	0
l.j	opcode 0x0						N																											
	3	3	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	10	9	8	7		6	5		4	3	2	1	0

	1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1											
l.jal	opcode 0x1						N																									
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
l.jalr	opcode 0x12						reserved										B					reserved									
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
l.jr	opcode 0x11						reserved										B					reserved									
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
l.lbs	opcode 0x22						D					A					I														
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
l.lbz	opcode 0x21						D					A					I														
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
l.lhs	opcode 0x24						D					A					I														
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
l.lhz	opcode 0x23						D					A					I														
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
l.movhi	opcode 0x6						D					reserved					K														
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
l.mul	opcode 0x38						D					A					B					reserved			opcode 0x3			reserved			opcode 0x6	
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
l.muli	opcode 0x2b						D					A					I														
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
l.mulu	opcode 0x38						D					A					B					reserved			opcode 0x3			reserved			opcode 0xb	
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
l.nop	opcode 0x15						reserved																									
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
l.or	opcode 0x38						D					A					B					reserved			opcode 0x0			reserved			opcode 0x4	
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
l.ori	opcode 0x29						D					A					K														
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

l.rfe	opcode 0x9						Reserved																									
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
l.rori	opcode 0x2d						D					A					reserved								opcode 0x4			L				
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
l.sb	opcode 0x36						I					A					B					I										
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
l.sfeq	opcode 0x720											A					B					reserved										
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
l.sfges	opcode 0x72b											A					B					reserved										
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
l.sfgeu	opcode 0x723											A					B					reserved										
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
l.sfgts	opcode 0x72a											A					B					reserved										
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
l.sfgtu	opcode 0x722											A					B					reserved										
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
l.sfles	opcode 0x72d											A					B					reserved										
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
l.sfleu	opcode 0x725											A					B					reserved										
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
l.sflts	opcode 0x72c											A					B					reserved										
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
l.sfltu	opcode 0x724											A					B					reserved										
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
l.sfne	opcode 0x721											A					B					reserved										
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
l.sh	opcode 0x37						I					A					B					I										
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
l.sll	opcode 0x38						D					A					B					reserved			opcode 0x8							
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

l.slli	opcode 0x2d						D					A					reserved							opcode 0x0			L				
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
l.sra	opcode 0x38						D					A					B					reserved			opcode 0x28							
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
l.srai	opcode 0x2d						D					A					reserved							opcode 0x2			L				
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
l.srl	opcode 0x38						D					A					B					reserved			opcode 0x18							
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
l.srli	opcode 0x2d						D					A					reserved							opcode 0x1			L				
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
l.sub	opcode 0x38						D					A					B					reserved			opcode 0x0			reserved			opcode 0x2	
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
l.subi	opcode 0x27						D					A					I															
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
l.sw	opcode 0x35						I					A					B					I										
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
l.sys	opcode 0x20000																K															
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
l.xor	opcode 0x38						D					A					B					reserved			opcode 0x0			reserved			opcode 0x5	
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
l.xori	opcode 0x2a						D					A					I															

Figure 0-1. List of Implemented 32-bit Instructions

Instruction Unit

General-Purpose Registers

Load/Store Unit

Integer Execution Pipeline

MAC Unit

System Unit

Exceptions

Data Cache Operation

Data Cache Load/Store Access

Load/store unit requests data from the data cache and stores them into the general-purpose register file and forwards them to integer execution units. Therefore LSU is tightly coupled with the data cache.

If there is no data cache block miss nor DTLB miss, load operations take two clock cycles to execute and store operations take one clock cycle to execute. LSU does all the data alignment work.

Data can be written to the data cache on a word, half-word or byte basis. Since data cache only operates in write-through mode, all writes are immediately written back to main memory or to the next level of caches.

Data Cache Block Fill Operation

When executing load instruction and a cache miss occurs, a 4 beat sequential read burst with critical word first is performed. Critical word is forwarded to the load/store unit to minimize performance loss because of the cache miss.

When executing store instruction and a cache miss occurs, a 4 beat sequential read burst with critical word first is performed. After read burst single word write is performed to write data of the store instruction back to main memory or next level of caches. Regardless of the wideness of store instruction, always a word write is performed.

Cache/Memory Coherency

Data cache in OR1200 operates only in write-through mode. Furthermore OR1200 is not intended for use in multiprocessor environments. Therefore no support for coherency between local data cache and caches of other processors or main memory is implemented.

Data Cache Enabling/Disabling

Data cache is disabled at power up. Entire data cache can be enabled by setting bit SR[DCE] to one. Before data cache is enabled, it must be invalidated.

Data Cache Invalidation

Data cache in OR1200 does not support invalidation of entire data cache. Normal procedure to invalidate entire data cache is to cycle through all data cache blocks and invalidate each block separately.

Data Cache Locking

Data cache implements way locking bits in data cache control register DCCR. Bits LWx lock individual ways when they are set to one.

Data Cache Block Prefetch

Data cache block prefetch is optional in the OpenRISC 1000 architecture and is not implemented in OR1200.

Data Cache Block Flush

Because data cache operates only in write-through mode, data cache block flush performs only block invalidation. Operation is performed by writing effective address to the DCBFR register.

Virtually the is no difference between data cache block flush and data cache block invalidate operation.

Data Cache Block Invalidate

Data cache block invalidate invalidates a block. Operation is performed by writing effective address to the DCBIR register.

Data Cache Block Write-back

Data cache block write-back operation does not do anything because data cache operates only in write-through mode.

Data Cache Block Lock

Locking of individual data cache blocks is not implemented in OR1200.

Instruction Cache Operation

Instruction Cache Instruction Fetch Access

Instruction unit requests instruction from the instruction cache and forwards them to the instruction queue inside instruction unit. Therefore instruction unit is tightly coupled with the instruction cache.

If there is no instruction cache block miss nor ITLB miss, instruction fetch operation takes one clock cycle to execute.

Instruction cache cannot be explicitly modified like data cache can be with store instructions.

Instruction Cache Block Fill Operation

On a cache miss, a 4 beat sequential read burst with critical word first is performed. Critical word is forwarded to the instruction unit to minimize performance loss because of the cache miss.

Cache/Memory Coherency

OR1200 is not intended for use in multiprocessor environments. Therefore no support for coherency between local instruction cache and caches of other processors or main memory is implemented.

Instruction Cache Enabling/Disabling

Instruction cache is disabled at power up. Entire instruction cache can be enabled by setting bit SR[ICE] to one. Before instruction cache is enabled, it must be invalidated.

Instruction Cache Invalidation

Instruction cache in OR1200 does not support invalidation of entire instruction cache. Normal procedure to invalidate entire instruction cache is to cycle through all instruction cache blocks and invalidate each block separately.

Instruction Cache Locking

Instruction cache implements way locking bits in instruction cache control register ICCR. Bits LWx lock individual ways when they are set to one.

Instruction Cache Block Prefetch

Instruction cache block prefetch is optional in the OpenRISC 1000 architecture and is not implemented in OR1200.

Instruction Cache Block Invalidate

Instruction cache block invalidate invalidates a block. Operation is performed by writing effective address to the ICBIR register.

Instruction Cache Block Lock

Locking of individual instruction cache blocks is not implemented in OR1200.

Data MMU
Instruction MMU
Programmable Interrupt Controller
Tick Timer
Power Management
Debug Unit

4

Registers

This section describes all registers inside the OR1200 core. Shifting *GRP* number 27 bits left and adding *REG* number computes the address of each special-purpose register. All registers are 32 bits wide from software perspective. *USER MODE* and *SUPV MODE* specify the valid access types for each register in user mode and supervisor mode of operation. R/W stands for read and write access and R stands for read only access.

Registers list

GRP #	REG #	REG NAME	USER MODE	SUPV MODE	DESCRIPTION
-	-	GPR0-GPR31	R/W	R/W	General-Purpose Registers
0	1	VR	-	Read Only	Version Register
0	2	UPR	-	Read Only	Unit Present Register
0	3	SR	-	R/W	Supervision Register
0	16	EPCR0	-	R/W	Exception PC Register
0	48	EEAR0	-	R/W	Exception EA Register
0	64	ESR0	-	R/W	Exception SR Register
0	80	CCR0	R/W	R/W	Condition Code Registers
1	0-255	DTLBMR0-DTLBMR255	-	Write Only	Data TLB Match Registers
1	256-511	DTLBTR0-DTLBTR255	-	Write Only	Data TLB Translate Registers
1	512	DMMUCR1	-	R/W	Data MMU Control Register 1
1	513	DMMUCR2	-	R/W	Data MMU Control Register 2
2	0-255	ITLBMR0-ITLBMR255	-	Write Only	Instruction TLB Match Registers
2	256-511	ITLBTR0-ITLBTR255	-	Write Only	Instruction TLB Translate Registers
2	512	IMMUCR1	-	R/W	Instruction MMU Control Register 1
2	513	IMMUCR2	-	R/W	Instruction MMU Control Register 2
3	0	DCCR	–	R/W	DC Control Register

3	1	DCBPR	W	W	DC Block Prefetch Register
3	2	DCBFR	W	W	DC Block Flush Register
3	3	DCBIR	–	W	DC Block Invalidate Register
3	4	DCBWR	W	W	DC Block Write-back Register
3	5	DCBLR	W	W	DC Block Lock Register
4	0	ICCR	–	R/W	IC Control Register
4	1	ICBPR	W	W	IC Block PreFetch Register
4	3	ICBIR	W	W	IC Block Invalidate Register
4	5	ICBLR	W	W	IC Block Lock Register
5	0	MACLO	R/W	R/W	MAC Low
5	1	MACHI	R/W	R/W	MAC High
6	0-3	DVR0-DVR3	-	R/W	Debug Value Registers
6	8-11	DCR0-DCR3	-	R/W	Debug Control Registers
6	16	DMR1	-	R/W	Debug Mode Register 1
6	17	DMR2	-	R/W	Debug Mode Register 2
6	18-19	DCWR0-DCWR1	-	R/W	Debug Watchpoint Counter Registers
6	20	DSR	-	R/W	Debug Stop Register
6	21	DRR	-	R/W	Debug Reason Register
6	22	DIR	-	R/W	Debug Instruction Register
8	0	PMR	-	R/W	Power Management Register
9	0	PICMR	-	R/W	PIC Mask Register
9	1	PICPR	-	R/W	PIC Priority Register
9	2	PICSR	-	R/W	PIC Status Register
10	0	TTCR	-	R/W	Tick Timer Control Register
10	1	TTIR	R/W	R/W	Tick Timer Incrementing Register

Table 0-1. List of All Registers

5

IO ports

OR1200 IP core has several interfaces. Figure 0-1 below shows all interfaces:

- Instruction and data WISHBONE host interfaces
- Power management interface
- Debug interface
- Interrupts interface

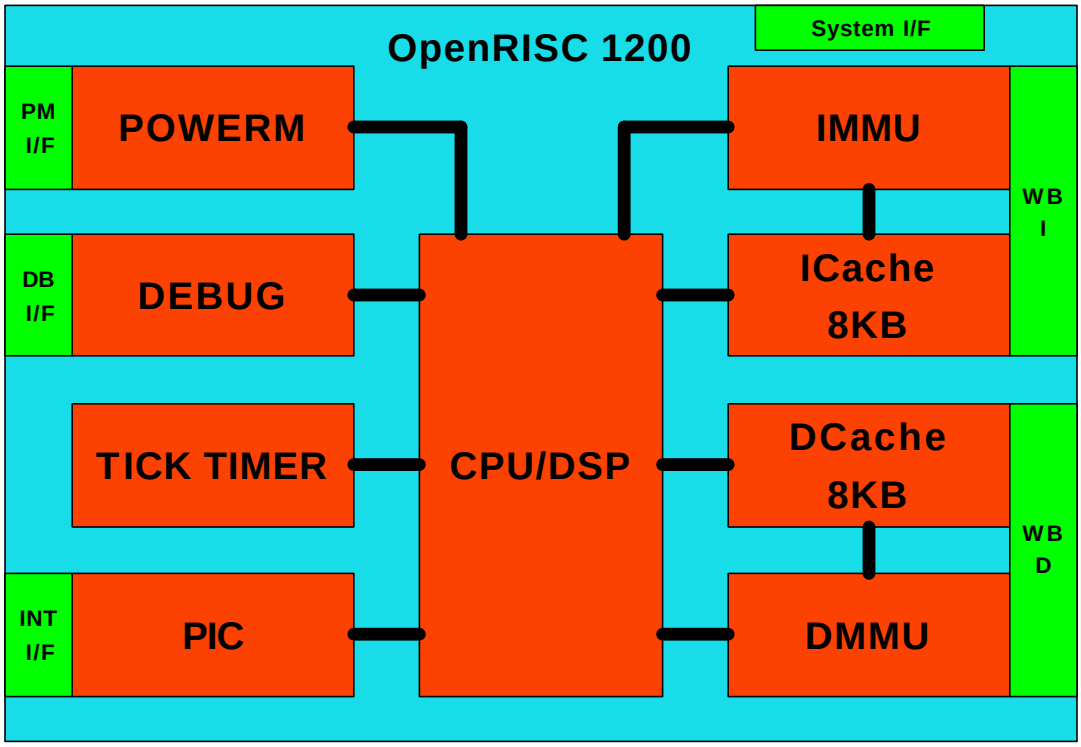


Figure 0-1. Core's Interfaces

Instruction WISHBONE Master Interface

OR1200 has two master WISHBONE Rev B compliant interfaces. Instruction interface is used to connect OR1200 core to memory subsystem for purpose of fetching instructions or instruction cache lines.

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	Not used
Iwb_WE_O	1	Output	Write transaction when asserted high
Iwb_STB_O	1	Outputs	Indicates valid data transfer cycle

Table 0-1. Instruction WISHBONE Master Interface' Signals

Data WISHBONE Master Interface

OR1200 has two master WISHBONE Rev B compliant interfaces. Data interface is used to connect OR1200 core to external peripherals and memory subsystem for purpose of reading and writing data or data cache lines.

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	Not used
dwb_WE_O	1	Output	Write transaction when asserted high
dwb_STB_O	1	Outputs	Indicates valid data transfer cycle

Table 0-2. Data WISHBONE Master Interface' Signals

System Interface

System interface connects reset, clock and other system signals to the OR1200 core.

Port	Width	Direction	Description
Arst	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
Buserr	1	Input	Invocation of bus error exception

Table 0-3. System Interface Signals

Debug Interface

Debug interface connects external development port to the RISC's internal debug facility. Debug facility allows control over program execution inside RISC, setting of breakpoints and watchpoints, and tracing of instruction and data flows.

Port	Width	Direction	Description
dbg_out	32	Output	Transfer of data from RISC to external development interface
dbg_in	32	Input	Transfer of data from external development interface inside RISC
dbg_addr	32	Input	Address of special-purpose register to be read or written
dbg_asel	4	Input	Asynchronous operation select for debug port
dbg_d_op	4	Output	Synchronous status of load/store unit
dbg_i_op	4	Output	Synchronous status of instruction fetch unit
dbg_wp	11	Output	Synchronous status of watchpoints
dbg_bp	1	Output	Synchronous status of the breakpoint
dbg_cpustall	1	Input	Synchronous stall of RISC's CPU core

Table 0-4. Debug interface

Power Management Interface

Power management interface provides signals for interfacing RISC core with external power management circuitry. External power management circuitry is required to implement functions that are technology specific and cannot be implemented inside OR1200 core.

Port	Width	Direction	Generation	Description
pm_clksd	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_cpugate	1	Output	Static (in SW)	Gating of main CPU clock
pm_cpuwakeup	1	Output	Dynamic (in HW)	Activate main CPU clock

Table 0-5. Power Management Interface

Interrupt Interface

Interrupt interface has interrupt inputs for interfacing external peripheral's interrupt outputs to the RISC core. All interrupt inputs are evaluated on positive edge of main RISC clock.

Port	Width	Direction	Description
pic_ints	PIC_INTS	Input	External interrupts

Table 0-6. Interrupt Interface

A

Core HW Configuration

This section describes parameters that are set by the user of the core and define configuration of the core. Parameters must be set by the user before actual use of the core in simulation or synthesis.

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	4 – 512	256	Data cache number of sets
DC_WAYS	1 – 4	2	Data cache number of ways
DC_LINE	16	16	Data cache line size
IC_IMPL	0 – 1	1	Instruction cache implementation
IC_SETS	4 – 512	256	Instruction cache number of sets
IC_WAYS	1 – 4	2	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	2 - 256	64	Data TLB number of sets
DTLB_WAYS	1 – 4	2	Data TLB number of ways
IMMU_IMPL	0 – 1	1	Instruction MMU implementation
ITLB_SETS	2 - 256	64	Instruction TLB number of sets
ITLB_WAYS	1 – 4	2	Instruction TLB number of ways
PIC_INTS	2 – 32	30	Number of interrupt inputs