Product Overview

PowerPC 405 CPU Core

.

© Copyright International Business Machines Corporation 2006

All Rights Reserved

Printed in the United States of America September 2006

The following are trademarks of International Business Machines Corporation in the United States, or other countries, or both.

IBM
IBM Logo
PowerPC
PowerPC Architecture
PowerPC 405
PPC405
CoreConnect
RISCWatch
RISCTrace

Virtex 2 Pro and Virtex 4 are registered trademarks of Xilinx, Inc.

Star IP is a registered trademark of Synopsys, Inc.

Other company, product and service names may be trademarks or service marks of others.

All information contained in this document is subject to change without notice. The products described in this document are NOT intended for use in applications such as implantation, life support, or other hazardous uses where malfunction could result in death, bodily injury, or catastrophic property damage. The information contained in this document does not affect or change IBM product specifications or warranties. Nothing in this document shall operate as an express or implied license or indemnity under the intellectual property rights of IBM or third parties. All information contained in this document was obtained in specific environments, and is presented as an illustration. The results obtained in other operating environments may vary.

THE INFORMATION CONTAINED IN THIS DOCUMENT IS PROVIDED ON AN "AS IS" BASIS. In no event will IBM be liable for damages arising directly or indirectly from any use of the information contained in this document.

IBM Systems and Technology Group
2070 Route 52, Bldg 330
Hopewell Junction, NY 12533-6351

The IBM home page can be found at
http://www.ibm.com

The IBM Systems and Technology Group’s Microelectronics Division home page can be found at
http://www.ibm.com/chips

Overview

The PowerPC 405 CPU core is a 32-bit RISC processor and a key element of IBM's Power Architecture™ licensing portfolio. The 405 CPU core is now in its fourth technology generation and part of IBM's 90nm ASIC IP library. The 405 CPU core is 100% PowerPC compatible and possesses all of the qualities necessary to make powerful and cost effective system-on-a-chip designs a reality. Optimized interfaces to the CoreConnect bus structure provide for a high bandwidth, low latency system interface solution. It combines the performance and features of standalone microprocessors with the modularity, low power and small die area of embedded CPU cores. Developers using the PPC405 core benefit from a broad set of enablement tools and software, all built around this Open architecture. The Power Architecture's power, flexibility and scalability has made it a leading architecture within embedded applications. From handheld devices to super computers, the scalability of the Power Architecture has been demonstrated. The PPC405 core clearly demonstrates the scalability of the PowerPC Architecture in its optimal fit for applications such as:

- *Consumer video applications including digital cameras, video games and set-top boxes*
- *Portable products such as PDAs and handheld GPS receivers*
- *Office automation products such as printers, X-terminals, and fax machines*
- *Networking and storage products such as disk drive controllers, routers, LAN switches, ATM switches, high performance modems, and network interface cards*
- *Industrial machine control and robotics*

In addition to its multiple IBM technology implementations representing over 60 IBM ASIC chip designs, the 405 CPU has been implemented in two generations of Xilinx FPGAs: Virtex2 Pro and Virtex4. The 405 CPU is available in a synthesizable form through the Synopsys Star IP program via license from IBM.

The IBM PowerPC Embedded Environment architecture specification provides for optimized CPU implementations that maintain 100% code compatibility with the classical PowerPC User Instruction Set Architecture (Book 1) instructions and registers. The optimized CPU implementation targeted to embedded environments are reflected in the Embedded Environment extensions to the PowerPC Virtual Environment Architecture (Book 2) and PowerPC Operating Environment Architecture (Book 3). The most significant optimizations include:

- *Memory management incorporating a wide dynamic range of memory page sizes*
- *Software-based memory management*
- *Efficient support of both big and little endian memory views*
- *Two-level interrupt hierarchy*
- *User-extendible instruction space for auxiliary processors*
- *Architected hardware debug facilities available to the programming model*

Example System-on-Chip Implementation

A typical system-on-a-chip (SOC) implementation based on the PowerPC 405 CPU and CoreConnect, uses a three level bus structure for system level communication, configuration and control functions. High bandwidth memory and system interfaces are tied to the 405 Core via the Processor Local Bus (PLB). Less demanding peripherals share the On-chip Peripheral Bus (OPB) and communicate to the PLB through the OPB bridge. On-chip configuration and control registers implemented in various IP cores and at chip-top in an SOC are connected to the CPU by the Device Control Register bus (DCR). This three level bus architecture provides common interfaces for the IP cores and enables quick turn-around custom solutions for high volume applications. Figure 1 illustrates a representative SOC configuration.

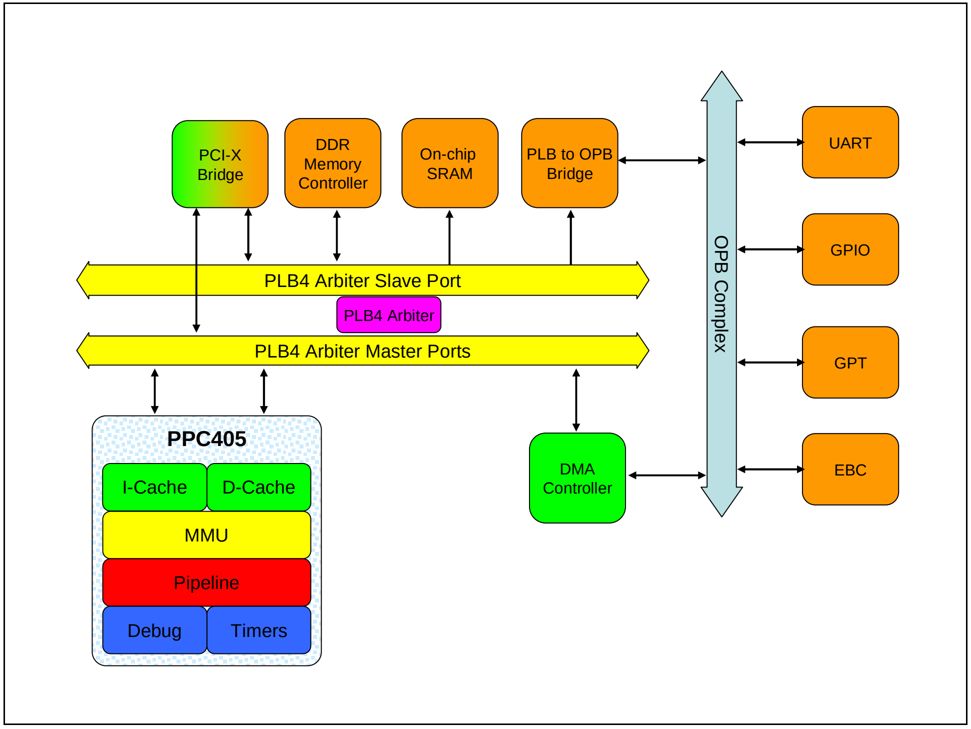


Figure 1: PPC405 CPU and CoreConnect based SOC Example

PPC405F6 CPU – 90nm Hard Core

Implemented in IBM’s CMOS-9SF 90nm process, the PPC405F6 CPU core demonstrates a truly outstanding level of balance and innovation in a hard core design point. Key facets of this optimized design include:

- Wide range of voltage and temperature operating points for tunable power / performance
  - Very low power / low voltage operation
  - Performance and power optimized at multiple operational voltages using patent pending hybrid Vt circuitd
  - High operational clock rates in SOC implementations
- Highly optimized layout for minimum silicon area impact and technology cost
- Cache and MMU TLB arrays include parity error detection
- Portability to other 90nm vendor technologies
- Design for manufacturability
- Very high fault coverage integrated manufacturing test architecture

Operation at supply voltages as low as 0.7V is enabled through dual mode, hybrid V(t) SRAM modules and dual mode register arrays for the general purpose register bank (GPRs). Operation at higher voltages vs. lower voltages is accomplished by voltage mode switches which can be strapped at chip-top or can be controlled by a programming model accessible configuration register. In addition to using lower operating voltages to achieve very low power operation, extensive use of various circuit and logic level techniques to achieve low power at all voltages are heavily employed. This includes extensive two-phase clock gating, a robust hybrid Vt cell library both in function available and drive strength, custom macros and use of patent pending tuning algorithms. A key benefit of the two-phase clocking architecture is that early and late clock gating can be used to reduce downstream logic transitions for logic that is not doing "useful work". For gating control signal paths that are determined in the first half of the clock cycle, early gating can "freeze" (not clock) both the master and slave portions of the latch reducing the power used in these latches to leakage power. Where gating control signal paths are not determined until the second half of the cycle, the master latch will still be clocked and use some power, but the slave latch will be "frozen" and the down-stream logic will not encounter possible logic transitions as a result.

At higher voltage operating levels, very high CPU clock rates for a five-stage pipeline design point can be achieved. This is provided for by the dual-mode SRAM and register arrays as well as clock tree feedback that enables common path pessimism removal. Integrated decoupling capacitors in the core layout helps minimize generated noise effects as well as shielding the SRAM and UTLB structures from externally generated noise sources. For extended lifetime applications, this core is rated for use at up to 200K power-on hours. Table 1 outlines the key specifications for the PPC405F6 core.

Complete with integrated instruction and data caches of 16KB each, the PPC405F6 core occupies only 2.17mm of silicon. Even more impressive is that this superior level of physical optimization was accomplished with standard density SRAMs and industry standard twin-well ground rules using only four levels of metal interconnect. As demonstrated by an existing implementation in UMC 90nm technology, this design point can be easily ported to a variety of major technology vendors.

Design for manufacturability is enhanced with the use of SRAM arrays which incorporate redundancy, significant use of redundant via structures and critical area minimization. The SRAM redundancy feature can be disabled if desired for foundry applications not having access to the IBM ASIC fuse controller. Extensive use of BIST structures for the SRAMS and UTLB plus a full access LSSD scan path set provide for essentially complete fault coverage in manufacturing test.

PPC405F6 Specifications	
Performance (Dhrystone 2.1)	1.52 DMIPS/MHz
Performance vs. power utilization (nominal process, 55C, 40 KPOH)	9.2 DMIPS/mW @ 1.2V, 658 MHz
	15.0 DMIPS/mW @ 0.9V, 388 MHz
	20.6 DMIPS/mW @ 0.7V, 190 MHz
Implementation technology	90nm – IBM CMOS-9SF – IBM ASIC Cu-08 – 4 level metal
CPU clock frequency – nominal process 55C, 40 KPOH, includes 50ps clock jitter	658 MHz @ 1.2V
CPU clock frequency – worst case process, 105C, 40 KPOH, includes 50ps clock jitter	433 MHz @ 1.1V
	288 MHz @ 0.9V
	140 MHz @ 0.7V
Power dissipation – nominal, active plus static	0.145 mW/MHz @ 1.2V, 55C; 109 mW @ 658 MHz
	0.08 mW/MHz @ 0.9V, 55C; 39 mW @ 388 MHz
	0.049 mW/MHz @ 0.7V, 55C; 14 mW @ 190 MHz
Operating voltage range	0.7V → 1.3V (1.2V nominal)
Operating temperature range	-55C → 125C
Core size	2.17 mm^2 (configured with 16KB/16KB L1 cache)
Extended lifetime	200 KPOH

Table 1: PPC405F6 CPU core specifications

Functional Units, Programming Model and Interfaces

The PowerPC 405 CPU core consists of a five stage single issue execution pipeline, a 32 element set of 32-bit general purpose registers (GPRs), 64-entry TLB memory management unit, L1 instruction and data caches of 16KB each, architecturally provided set of timers, a program model accessible hardware debug module and several system interfaces. Figure 2 contains a high-level block diagram of the 405 CPU core.

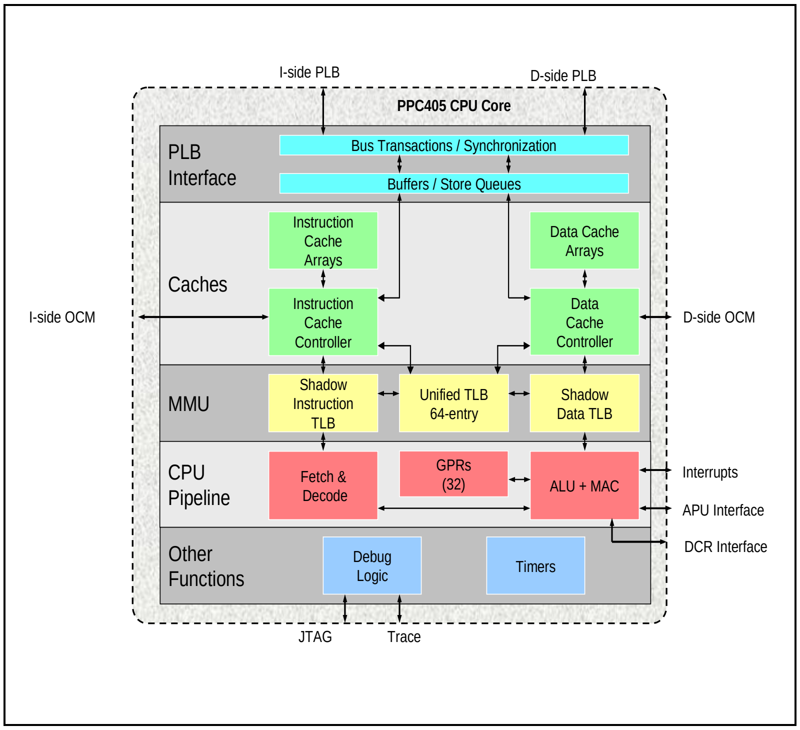


Figure 2: PPC405 CPU core block diagram.

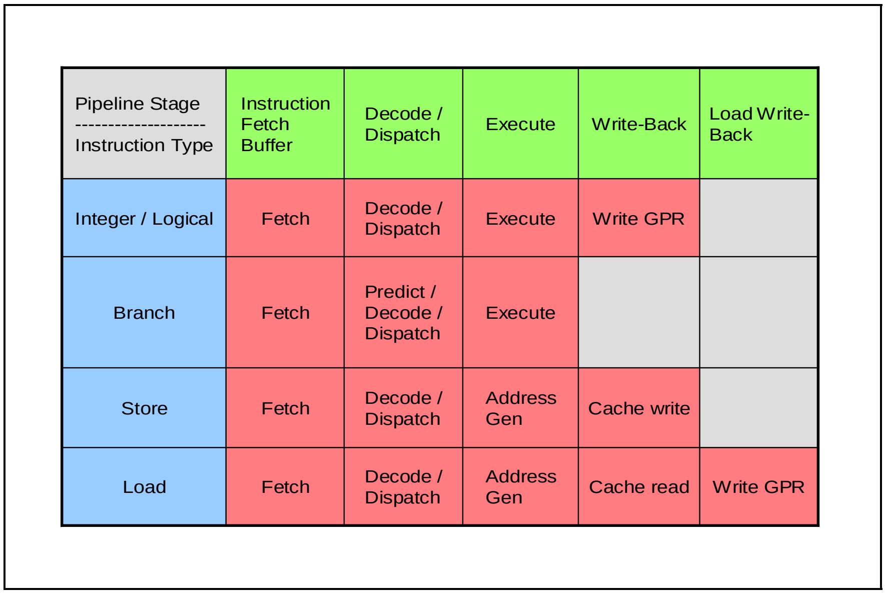
CPU pipeline

The 405 CPU operates on instructions in a five stage pipeline consisting of a fetch, decode, execute, write-back, and load write-back stage. The various instruction types use the pipeline stages in different ways as illustrated in figure 3.

The fetch and decode logic maintains a steady flow of instructions to the execute unit by placing up to two instructions in the fetch queue. The fetch queue consists of two pre-fetch buffers, PFB1 and PFB0, and the decode stage (DCD). When the queue is empty, instructions flow directly into the decode stage.

Static branch prediction as implemented on the 405 CPU core takes advantage of standard statistical properties of code. Branches with negative address displacement are by default assumed taken. Branches that do not test the condition or count registers are also predicted as taken. The 405 CPU core bases branch prediction upon these default conditions when a branch is not resolved and speculatively fetches along the predicted path. The default prediction can be overridden by software. Branches are examined in both the decode and pre-fetch buffer 0 stages. Two branch instructions can be handled simultaneously. If the branch in decode is not taken, the fetch logic fetches along the predicted path of the branch instruction in pre-fetch buffer 0. Table 2 shows the instruction timing for various branch instructions cases based on which pipeline stage the branch is resolved.

The 405 CPU core has a single issue execute unit which contains the GPR register file, arithmetic logic unit (ALU) and the multiply-accumulate (MAC) unit. The execution unit performs all 32-bit PowerPC integer instructions in hardware and is compliant with the IBM PowerPC Embedded Environment specification. Table 3 lists instruction latencies for the majority of the instruction types.



Pipeline Stage ------------------- Instruction Type	Instruction Fetch Buffer	Decode / Dispatch	Execute	Write-Back	Load Write-Back
Integer / Logical	Fetch	Decode / Dispatch	Execute	Write GPR	
Branch	Fetch	Predict / Decode / Dispatch	Execute		
Store	Fetch	Decode / Dispatch	Address Gen	Cache write	
Load	Fetch	Decode / Dispatch	Address Gen	Cache read	Write GPR

Figure 3: PPC405 pipeline utilization by instruction category

The register-file consists of thirty-two 32 bit general purpose registers (GPR) which are accessed with three read ports and two write ports. During the decode stage, data is read out of the GPR's and feed to the execute unit. Likewise, during the write-back stage, results are written to the GPR. The use of the five-port structure for the register file enables either a load or a store operation to execute in parallel with an ALU operation. Extensive power management features and extended low voltage capability have been incorporated in the GPR.

The MAC unit adds nine operations to the 405 CPU core instruction set. MAC instructions operate on either signed or unsigned 16 bit operands and accumulate the results in a 32 bit GPR. All MAC unit instructions have a single cycle throughput assuming no data dependencies. Table 4 lists the multiply-accumulate and half-word multiply operations supported by the MAC unit. Most of the instructions have four flavors: unsigned, signed, modulo or saturate. Modulo and saturate determine the behavior of the instruction when the accumulated value overflows.

The 405 CPU services exceptions generated by error conditions, the internal timer facility, debug events, and the external interrupt controller interface. All together, there are nineteen possible exceptions including the two provided from an external interrupt controller.

Exceptions are divided into two classes, critical and non-critical. Each class of exceptions has its own pair of save/restore registers for holding the machine status. Separate save/restore registers allows the 405 CPU core to quickly handle critical interrupts.

When an interrupt is taken, the 405 CPU automatically writes the machine status to save/restore register SRR0 and SRR1 for non-critical interrupts or SRR2 and SRR3 for critical interrupts. SRR0 and SRR2 are written with the address of the instruction triggering the exception or the next sequential instruction. SRR1 and SRR3 are written with a copy of the machine state register. The machine status is automatically restored at the end of an exception handler when the return from interrupt (rfi) or return from critical interrupt (rfci) instruction is executed.

The EIC interface extends interrupt support to logic external to the 405 CPU core through the external and critical interrupt signals. These inputs are level sensitive. The critical interrupt and external interrupt signals are conceptually logic OR's of all implementation-specific critical and non-critical interrupts outside the core.

The auxiliary processor unit interface (APU) enables a custom design implementation to use an auxiliary processor to execute instructions that are not part of the PowerPC Architecture or IBM PowerPC Embedded Environment, or to execute PowerPC Floating Point instructions in hardware.

Branch Instructions	Execution Latency (#1)	Execution Latency (#2)
Branch known taken (unconditional branch) #1 – resolved in PFB0 stage #2 – resolved in DCD stage	1	2
Branch known not taken (fall-through)	1	--
Taken branch predicted taken (conditional branch) #1 – resolved in PFB0 stage #2 – resolved in DCD stage	1	2
Taken branch predicted not taken (conditional branch) #1 – resolved in DCD stage #2 – resolved in EXE stage	2	3
Not taken branch predicted taken (conditional branch) #1 – resolved in DCD stage #2 – resolved in EXE stage	2	3
Not taken branch predicted not-taken (conditional branch)	1	--
Note: conditional branches have a dependency on the condition register (CR)		

Table 2: Execution timing for branch instructions

Common Instructions	***Execution Latency***	
Arithmetic	1	
Traps	2	
Logical	1	
Shifts/rotates	1	
Divide	35	
Loads[b] (cache hit)	1	
Load multiples/strings (cache hit)	1/transfer	
Stores (cache hit)	1	
Store multiples/strings (caches hit)	1/transfer	
Move to/from Device Control Register (DCR)	3	
Move to/from Special Purpose Register	1	
Note: Load instructions have a 1 cycle use penalty. Data acquired during the execute pipeline stage is not available until after the completion of the write back pipeline stage. Write back takes one clock cycle. Note: DCR access instructions depend on chip-top bus clocking implementation; three cycles is the shortest latency.		

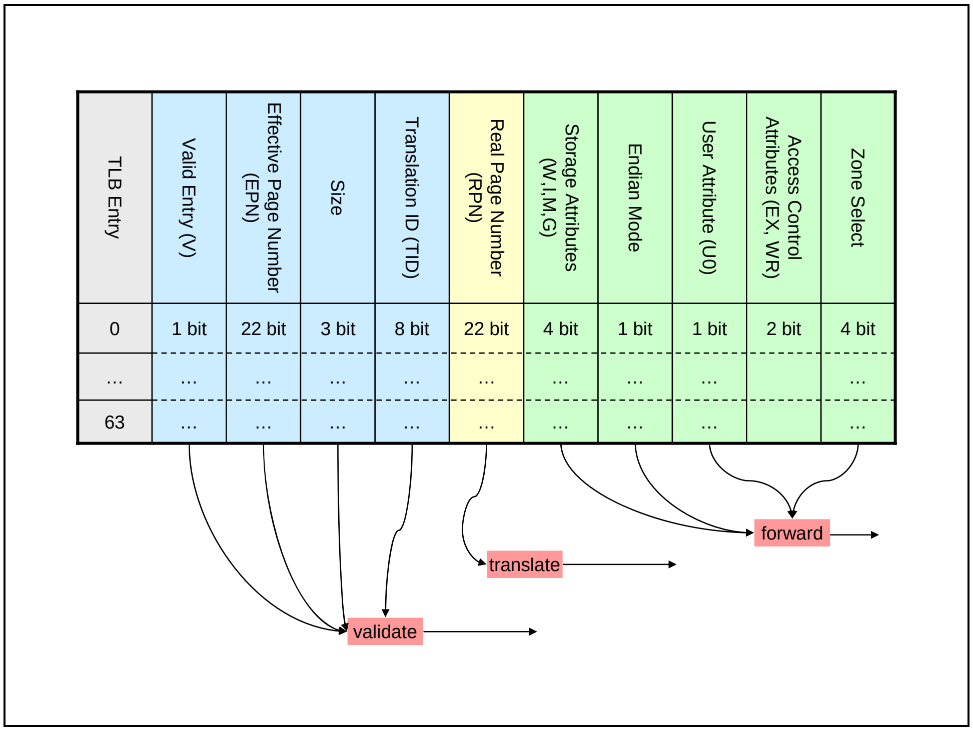
Table 3: Execution timing for most common instruction types

Multiply and Multiply Accumulate Instructions	Execution Latency	Execution Throughput
Multiply half word (16 x 16 → 32)	2	1
Multiply one sign extended 16 bit half word in a 32 bit register with a 32 bit word (16 x 32 → least significant 32 bits of a 48 bit product)	3	2
Multiply (32 x 32 → least significant 32 bits of a 64 bit product)	4	3
Multiply (32 x 32 → least significant 32 bits of a 64 bit product) with overflow detection	5	4
Multiply (32 x 32 → most significant 32 bits of 64 bit product) with or without overflow detection enabled	5	4
MAC (16 x 16 + 32 → 32)	2	1
Note: Throughput applies when no dependencies exist between the result of the multiplication and the source of the subsequent multiplication. However, the target accumulator from one MAC instruction may be used as the source accumulator for the next MAC instruction.		

Table 4: Instruction timing for MAC and multiply instructions

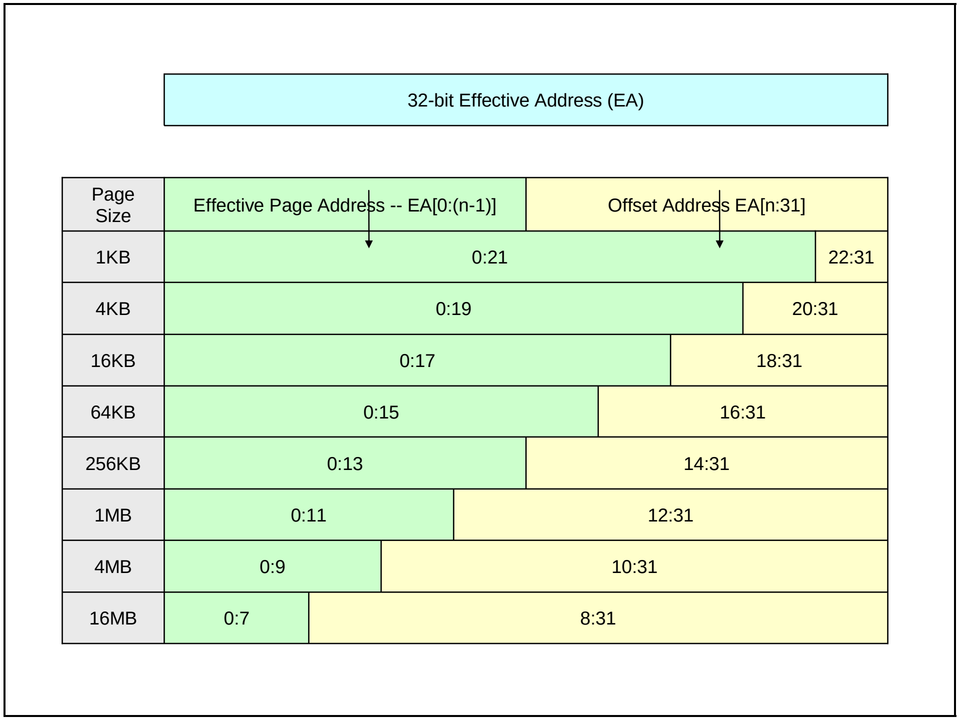
Memory management

The memory management unit (MMU) supports multiple page sizes as well as a variety of storage protection attributes and access control options. Multiple page sizes improve memory efficiency and minimize the number of TLB misses. The 405 CPU core gives programmers the flexibility to have any combination of the following eight possible pages sizes in the translation look-aside buffer (TLB) simultaneously: 1KB, 4KB, 16KB, 64KB, 256KB, 1MB, 4MB and 16MB. Figure 4 shows the various fields of each MMU entry and what function they perform in memory mapping and access.



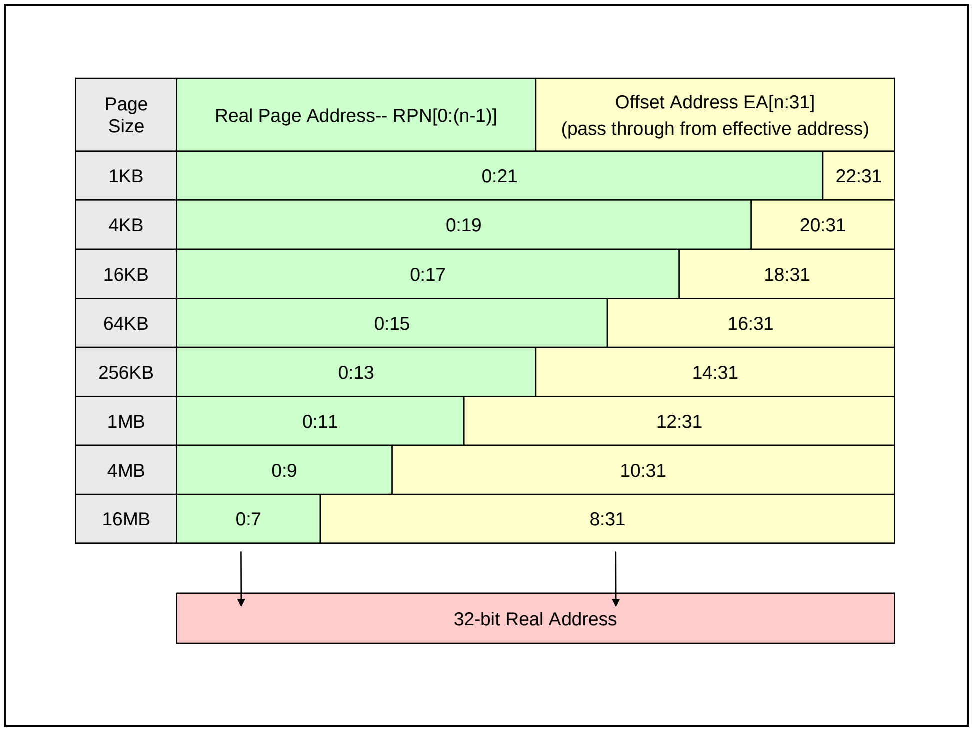
TLB Entry	Valid Entry (V)	Effective Page Number (EPN)	Size	Translation ID (TID)	Real Page Number (RPN)	Storage Attributes (W,I,M,G)	Endian Mode	User Attribute (U0)	Access Control Attributes (EX, WR)	Zone Select
0	1 bit	22 bit	3 bit	8 bit	22 bit	4 bit	1 bit	1 bit	2 bit	4 bit
...	...	...	...	...	...	...	...	...		...
63	...	...	...	...	...	...	...	...		...

Figure 4: MMU TLB entry definition



Page Size	Effective Page Address -- EA[0:(n-1)]	Offset Address EA[n:31]
1KB	0:21	22:31
4KB	0:19	20:31
16KB	0:17	18:31
64KB	0:15	16:31
256KB	0:13	14:31
1MB	0:11	12:31
4MB	0:9	10:31
16MB	0:7	8:31

Figure 5: MMU effective address and size validation



Page Size	Real Page Address-- RPN[0:(n-1)]	Offset Address EA[n:31] (pass through from effective address)
1KB	0:21	22:31
4KB	0:19	20:31
16KB	0:17	18:31
64KB	0:15	16:31
256KB	0:13	14:31
1MB	0:11	12:31
4MB	0:9	10:31
16MB	0:7	8:31

Figure 6: MMU real address generation

Each page of memory is accompanied by a set of storage attributes. These attributes include cacheability, write through/write back mode, allocate on write, big/little endian, guarded and user-defined. The user-defined attribute can be used to mark a memory page with an application specific meaning. The guarded attribute controls speculative accesses. The big/little endian attribute marks a memory page as having a particular byte ordering. Write through/write back specifies whether memory is updated in addition to the cache during store operations.

The MMU uses a 64-entry fully-associative unified TLB to reduce the overhead of address translation. Contention for the main TLB between data address and instruction address translation is minimized through the use of a four-entry instruction shadow TLB (ITLB) and an eight-entry data shadow TLB (DTLB). The ITLB and DTLB shadow the most recently used entries in the unified TLB. The MMU manages the replacement strategy of the ITLB and DTLB leaving the unified TLB to software control. Real-time operating systems are free to implement their own replacement algorithm for the unified TLB. Figures 5 and 6 show how the MMU effective and real address fields as well as the page size attributes are used in the address translation process. Figure 7 provides the flow of the translation process through the shadow TLB entries and subsequently through the UTLB.

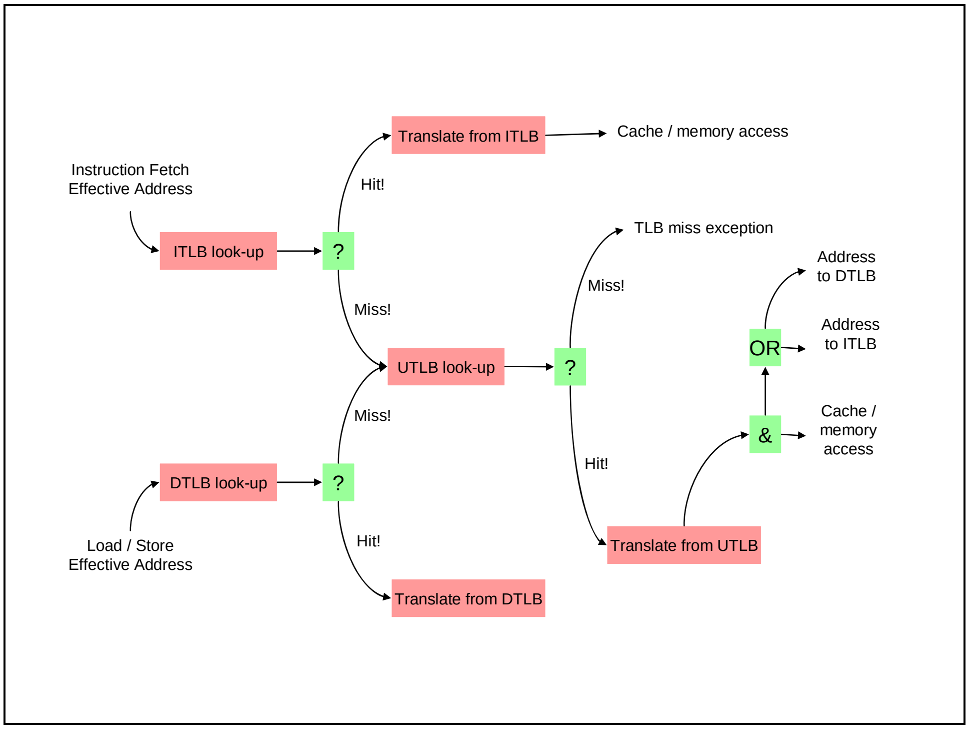


Figure 7: MMU TLB translation flow

The 405 CPU core offers real mode as an alternative to address translation and MMU storage attributes. Real mode supports the same storage attributes as the MMU. Disabling address translation for instruction addresses and/or data addresses enables real mode for all instruction and/or data addresses. Real mode offers separate control of the storage attributes for all instruction and data addresses through a set of registers that divide the 4GB address space into thirty-two, 128 MB fixed memory regions.

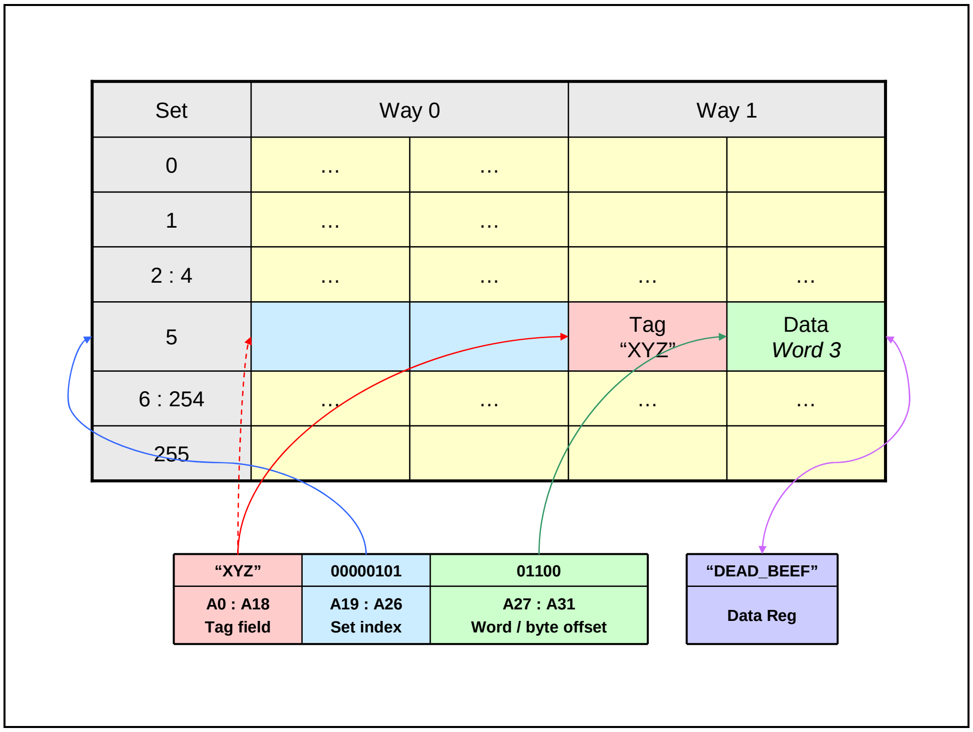
Instruction and data caches

The 405 CPU core accesses memory through the instruction and data cache unit controllers. These cache controller units coordinate access to instruction and data memory spaces existing in system memory, in cache units and in processor-local on-chip memory (OCM). All cacheable accesses are presented first to the cache units and subsequently to the system memory. Cache hits are handled as single cycle memory accesses to the instruction and data caches. Cache misses result in a system memory access via the processor local bus interface.

The 405 core has separate 16KB two-way set-associative instruction and data caches, each with 32 byte cache lines. The cache set-associativity is architected using a patent pending technique for optimized power saving without sacrificing performance.

In order to maximize CPU performance, the cache arrays are non-blocking. Non-blocking caches allow the 405 CPU core to overlap execution of instructions while cache miss line fill operations. The caches, therefore, continue supplying data and instructions without interruption to the pipeline.

The cache controllers replace cache lines according to the least recently used (LRU) replacement policy. The LRU bit determines which line of a set (congruence class) to replace. The most recently accessed line in the set is retained and the other line is marked as LRU, using the LRU bit. The cache controller replaces the LRU line during a line fill.



Set	Way 0		Way 1	
0	…	…		
1	…	…		
2 : 4	…	…	…	…
5			Tag “XYZ”	Data *Word 3*
6 : 254	…	…	…	…
255				

Figure 8: Instruction and data cache architecture

The instruction cache controller unit (ICU) utilizes a 32 byte (eight instruction word) fill buffer to enhance cache performance. For cacheable memory, a requested cache line first passes through the fill buffer prior to being written into the cache arrays. When the fill buffer contains all eight words of a cache line, the ICU writes the contents of the fill buffer to the cache. Using the fill buffer reduces cache blocking on line fills from eight to two cycles. While the cache line is being captured in the fill buffer, the ICU forwards the target word (the word requested by the fetch logic) to the fetch queue, bypassing the cache arrays. If the fetch logic makes further requests of words in the cache line, the ICU places them on the bypass path to expedite the request.

Non-cacheable memory accesses benefit from the bypass path and the fill buffer because non-cacheable instruction fetches are also handled as line reads on the system bus. In response to a PLB fetch request, the ICU places the target word on the bypass path while the fill buffer captures the remaining instructions. The fill buffer will hold the instructions until overwritten by another request. A performance boost occurs when the fetch logic requests instructions held in the fill buffer since the fill buffer has the same access latency as a cache hit.

The ICU supports big endian or little endian byte ordering for instructions stored in external memory. Since the PowerPC architecture is big endian internally, the ICU rearranges the instructions stored as little endian into the big endian format. Therefore, the instruction cache always contains instructions in big endian format so that the byte ordering is correct for the execute unit. This feature allows the 405 core to be used in systems designed to function in a little endian environment.

In the data cache controller unit (DCU), data passes through the fill buffer and the write buffer prior to being written into the data cache on cacheable line fills. After the fill buffer receives the eighth word of the cache line, the data cache controller writes the contents of the fill buffer into the write buffer. The cache controller then takes two clock cycles to write the contents of the write buffer into the cache arrays. Using the write buffer reduces cache blocking from 8 to 3 cycles and frees the fill buffer for a subsequent line reads.

Cacheable data requested by the execute unit is sent on the bypass path directly into the operand registers of the execute unit. The data cache controller places the target word on the bypass path as the fill buffer captures data words off the PLB. Additional requests of the cache line held in the fill buffer are also forwarded directly to the operand registers in the execute unit.

For non-cacheable line accesses, the DCU utilizes the fill buffer even though the data will not be written into the data cache. As with cacheable line accesses, the targeted data word is sent on the by pass path directly to the operand registers of the execute unit. The data captured in the fill buffer is accessible to the execute unit until it is overwritten by a subsequent read (load) request. Data accessed in the fill buffer achieves cache hit performance which gives non-cacheable accesses a noticeable performance boost.

Single queued flushes are non-blocking since the DCU contains a 2-line flush queue. A 2-line flush queue enables the DCU to access the data cache arrays for load and store cache hits while handling a single line flush. Flushed data is first collected in the line buffer built into the output of the data cache arrays. The DCU then passes the flushed data cache line to the second queue position, the flush buffer. From the flush buffer, the data is sent to memory by way of the PLB.

The DCU supports write-back or write-through mode and allocate-on-write. In write-back mode, store hits are written to the cache and not to main memory. Main memory is later modified if and when the line is flushed from the cache. In write-through mode, the data cache controller writes main memory for store misses as well as store hits; every store operation generates a PLB write request. Both modes can be used with allocate-on-write. When allocate-on-write is enabled, a store miss to cacheable memory forces the data cache controller to allocate a line in the data cache, generate a line fill and flush the LRU line if marked dirty. The DCU may also be configured to write-through without allocating a cache line on a cache miss. If a cache line is not allocated, the DCU handles the requested data as if it is non-cacheable. Controlling allocation of cacheable memory allows programmers to hold critical data in the data cache.

The 405 CPU core has hardware support for big endian and little endian accesses. Since byte ordering and alignment depends on the data size (byte, half word or word) accessed, the data cache exactly mirrors external memory. Alignment and byte ordering are handled by byte steering logic between the DCU and the GPR during load operations. Byte steering logic in the DCU returns data to its original byte ordering before writing it back to the data cache, on-chip memory or memory accessed over the PLB. Having byte steering logic integrated into 405 CPU core eliminates the difficulty of integrating peripherals originally designed for a little endian system.

Timers

The PPC405 contains a 64-bit time base register and three timer modules: an auto-reloading decrementing counter (DEC), the fixed interval timer (FIT), and the watch-dog timer (WDT). The time base register counter increments synchronously with the CPU clock or an externally supplied clock source. The three timers are synchronous with the time base. The DEC is a 32-bit register that decrements at the time base increment rate. The user loads the DEC register with a value to create the desired delay. When the register reaches zero, the timer stops decrementing and generates an interrupt. Optionally, the DEC can be programmed to reload the value last written to the DEC auto-reload register, after which the DEC continues to decrement. The FIT generates periodic interrupts based on one of four selectable bits in the time base. When the selected bit changes from 0 to 1, the PPC405 generates a FIT exception. The watchdog timer provides a periodic critical-class interrupt based on a selected bit in the time base. This interrupt can be used for system error recovery in the event of software or system lockups. Users may select one of four time periods for the interval and the type of reset generated if the watchdog timer expires twice without an intervening clear from software. If enabled, the watchdog timer generates a reset unless an exception handler updates the watchdog timer status bit before the timer has completed two of the selected timer intervals.

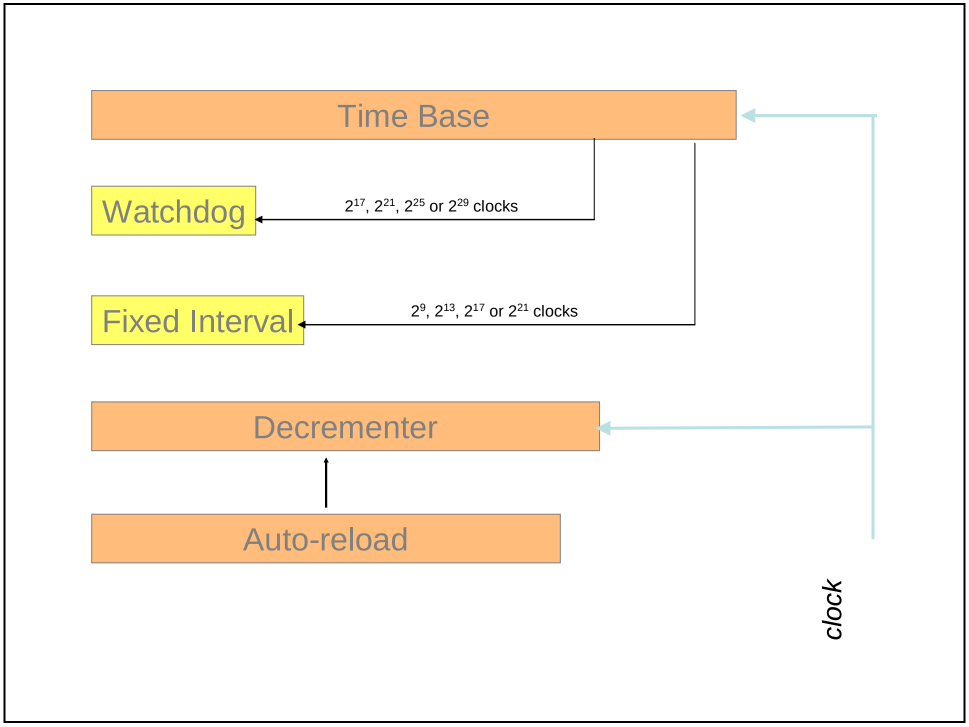

Time Base
Watchdog
2^{17}, 2^{21}, 2^{25} or 2^{29} clocks
Fixed Interval
2^{9}, 2^{13}, 2^{17} or 2^{21} clocks
Decrementer
Auto-reload
clock

Figure 9: PPC405 timer facilities

Debug facilities

All architected resources on the PPC405 core can be accessed through the debug logic. Upon a debug event, the CPU provides debug information to a debug tool. Three different types of debug tools are supported and depend on the current debug mode setting in the CPU: software debug tools, JTAG probe based debuggers and instruction trace tools.

Internal debug mode provides operating systems and BIOS level software with access to the 405's debug resources. Exception handling software residing at a dedicated interrupt vector and triggered via trap instructions or the hardware debug resource registers (e.g. instruction address compare registers) provides the mechanism and communication path for a debug tool. Exception-handling software has read-write access to all registers and can set hardware or software breakpoints.

In external debug mode, the PPC405 enters stop state (i.e., stops instruction execution) when a debug event occurs. This mode offers a debug tool non-invasive read-write access to all registers in the CPU via the JTAG interface. Once the core is in stop state, the debug tool can restart the PPC405, step an instruction, freeze the timers or set hardware or software break points. In addition to pipeline control, the debug logic is capable of writing instructions into the instruction cache, eliminating the need for external memory during initial board bring up. With the debug wait option enabled, the PPC405 will respond to interrupts and temporarily leave stop state to service them before returning to debug wait mode. In external debug mode with debug wait disabled, interrupts are disabled while in stop state. Debug wait option is particularly useful when debugging real-time control systems.

In real-time trace debug mode, instruction trace information is continuously broadcast to the trace port. When a debug event occurs, an external debug tool saves instruction trace information before and after the event. The number of traced instructions depends only on the memory buffer depth of the trace tool.

A comprehensive set of debug event types are supported including: instruction completion, branch taken, exception occurred, trap instruction encountered, instruction address compare triggered, data address compare triggered, data value compare triggered, debug tool induced unconditional event.

System Interfaces

The two processor local bus (PLB) interfaces are the primary system interfaces for the 405 CPU core to access system memory and I/O devices. Both of these interfaces are implemented as 64-bit PLB master devices. Both of these interfaces primarily use PLB line transaction types as most processor traffic is of the form of cache line fills and flushes. The data side does implement single beat (non-block type) transactions for supporting non-cacheable reads and writes from the data side interface.

The Device Control Register (DCR) bus interface is a configuration bus for components external to the 405 CPU core. Using the DCR bus to manage status and configuration registers reduces PLB traffic and improves system integrity. System resources on the DCR Bus can be more effectively isolated from wayward code since the DCR bus is not part of the system memory map and accesses to it are privileged mode only.

Instruction side and data side OCM interfaces provide access to on-chip processor-local memory if that memory is implemented. Non-cacheable accesses are presented to the (OCM) interfaces in the same time frame as cacheable accesses are presented to the cache units. If memory exists at the requested address, the access is satisfied via that path. If there is no response on the OCM interface, then the access is presented to system memory as in the cache miss scenario. The OCM interfaces can have the same access time as a cache hit (single cycle) for lower CPU clock speeds. At higher CPU speeds, OCM accesses are normally two cycles. Instruction side local-memory is often used to hold critical code such as an interrupt handler that requires guaranteed low-latency access. Data side local-memory offers the same fixed low-latency access and may be used to hold critical data such as filter coefficients for a DSP application. Figure 10 shows the attachment of memory through the OCM interfaces.

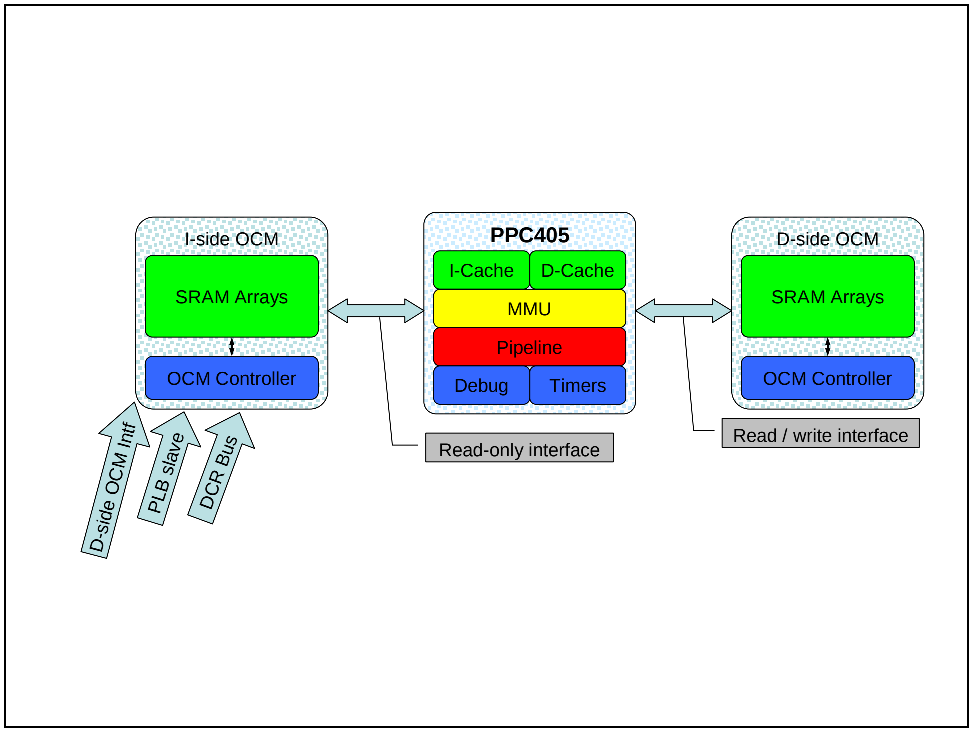


Figure 10: Instruction and data local memory attachment via OCM interfaces

Debugging interfaces on the 405 CPU core, consisting of the JTAG and trace ports, offers access to resources internal to the core and assist in software development. The JTAG port may be configured to provide basic JTAG chip testing functionality in addition to the ability for external debug tools to gain control of the processor for debug purposes. The trace port furnishes programmers with a mechanism for acquiring instruction traces.

The JTAG port complies with IEEE Std 1149.1, which defines a test access port (TAP) and boundary scan architecture. Extensions to the JTAG interface provide debuggers such as RISCWatch with processor control that includes stopping, starting and stepping the 405 CPU core. These extensions are compliant with the IEEE 1149.1 specifications for vendor-specific extensions.

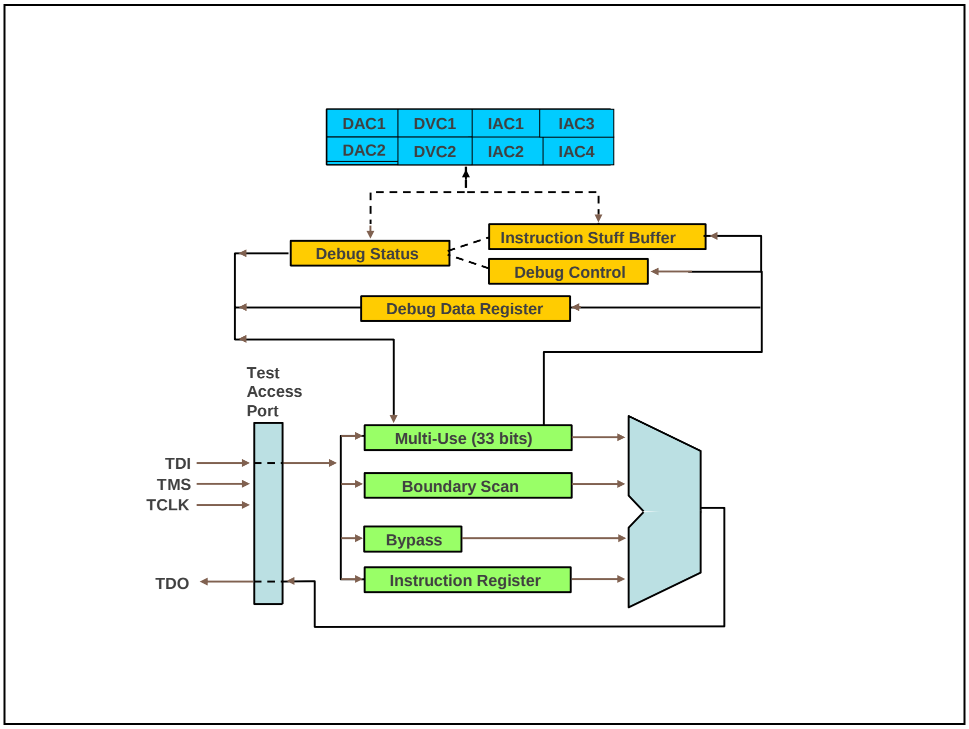


Figure 11: JTAG port access to debug logic

The trace port provides instruction trace information to an external trace tool, such as RISCTrace. The 405 CPU core is capable of back trace and forward trace. Back trace is the tracing of instructions prior to a debug event while forward trace is the tracing of instructions after a debug event.

Enablement and Infrastructure

The PPC405 CPU core, as a member of the PowerPC 400 Family, is supported by the IBM PowerPC Embedded Tools^{TM} program, in which over 80 third party vendors have combined with IBM to provide a complete tools solution. Development tools for the PPC405 include C/C++ compilers, debuggers, real-time operating systems (RTOS), hardware level and system level simulation models, evaluation and development platforms and technical training programs. As part of the tools program, IBM provides the RISCWatch^{TM} debugger and RISCTrace^{TM} instruction trace capability, full functional hardware simulation models, an instruction set simulator (ISS) and SystemC models. The PowerPC embedded application binary interface (EABI) is supported by the major industry tools and OS providers as well as the Linux open source environment.

Availability statement and references

The PPC405 CPU core is currently available in 90nm and 130nm hard core implementations through IBM ASIC products. The synthesizable implementation of the core is available through the Synopsys Star IP program via license from IBM. The 405 core is also available as an embedded core in the Xilinx Virtex 2 Pro and Virtex 4 FPGA families.

For further information regarding the PowerPC 405 CPU core, contact an IBM Microelectronics sales representative via the IBM web pages: http://www-03.ibm.com/chips/support/contact/