A microprocessor architecture including a unified cache debug unit. A debug unit on the processor chip receives data/command signals from a unit of the execute stage of the multi-stage instruction pipeline of the processor and returns information to the execute stage unit. The cache debug unit is operatively connected to both instruction and data cache units of the microprocessor. The memory subsystem of the processor may be accessed by the cache debug unit through either of the instruction or data cache units. By unifying the cache debug in a separate structure, the need for redundant debug structure in both cache units is obviated. Also, the unified cache debug unit can be powered down when not accessed by the instruction pipeline, thereby saving power.
MICROPROCESSOR ARCHITECTURE INCLUDING UNIFIED CACHE DEBUG UNIT

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] This application claims priority to provisional application No. 60/572,228 filed May 19, 2004, entitled “Microprocessor Architecture,” hereby incorporated by reference in its entirety.

FIELD OF THE INVENTION

[0002] This invention relates generally to microprocessor architecture and more specifically to an improved cache debug unit for a microprocessor.

BACKGROUND OF THE INVENTION

[0003] A major focus of microprocessor design has been to increase effective clock speed through hardware simplifications. Exploiting the property of locality of memory references, cache memories have been successful in achieving high performance in many computer systems. In the past, cache memories of microprocessor-based systems were provided off-chip using high performance memory components. This was primarily because the amount of silicon area necessary to provide an on-chip cache memory of reasonable performance would have been impractical. Increasing the size of an integrated circuit to accommodate a cache memory adversely impacts the yield of the integrated circuit in a given manufacturing process. However, with the density achieved recently in integrated circuit technology, it is now possible to provide on-chip cache memory economically.

[0004] In a computer system with a cache memory, when a memory word is needed, the central processing unit (CPU) looks into the cache memory for a copy of the memory word. If the memory word is found in the cache memory, a cache “hit” is said to have occurred, and the main memory is not accessed. Thus, a figure of merit which can be used to measure the effectiveness of the cache memory is the “hit” ratio. The hit ratio is the percentage of total memory references in which the desired datum is found in the cache memory without accessing the main memory. When the desired datum is not found in the cache memory, a “cache miss” is said to have occurred and the main memory is then accessed for the desired datum. In addition, in many computer systems there are portions of the address space which are not mapped to the cache memory. This portion of the address space is said to be “uncached” or “uncacheable”. For example, the addresses assigned to input/output (I/O) devices are almost always uncached. Both a cache miss and an uncacheable memory reference result in an access to the main memory.

[0005] In the course of developing or debugging a computer system, it is often necessary to monitor program execution by the CPU or to interrupt one instruction stream to direct the CPU to execute certain alternate instructions. A known method used to debug a processor utilizes means for observing the program flow during operation of the processor. With systems having off-chip cache, program observability is relatively straightforward by using probes. However, observing the program flow of processors having cache integrated on-chip is much more difficult because most of the processing operations are performed internally within the chip.

[0006] As integrated circuit manufacturing techniques have improved, on-chip cache has become standard in most microprocessors designs. Due to difficulties in interfacing with the on-chip cache, debugging systems have also had to move onto the chip. Modern on-chip cache memories may now employ cache debug units directly in the cache memory themselves.

[0007] There is therefore a need for a cached processor having relatively simple design, reduced silicon footprint and reduced power consumption that allows the real time capture of data in the cached processor for debug purposes and which can be used at high frequencies.

[0008] It should be appreciated that the description herein of various advantages and disadvantages associated with known apparatus, methods, and materials is not intended to limit the scope of the invention to their exclusion. Indeed, various embodiments of the invention may include one or more of the known apparatus, methods, and materials without suffering from their disadvantages.


SUMMARY OF THE INVENTION

[0010] Various embodiments of the invention are disclosed that overcome one or more of the shortcomings of conventional microprocessors through a microprocessor architecture having a unified cache debug unit. In these embodiments, a separate cache debug unit is provided which serves as an interface to both the instruction cache and the data cache. In various exemplary embodiments, the cache debug unit has shared hardware logic accessible to both the instruction cache and the data cache. In various exemplary embodiments, a cache debug unit may be selectively switched off or on at a separate clock than the instruction pipeline. In various exemplary embodiments, an auxiliary unit of the execute stage of the microprocessor core is used to pass instructions to the cache debug unit and to receive responses back from the cache debug unit. Through the instruction cache and data cache respectively, the cache debug unit may also access the memory subsystem to perform cache flushes, cache updates and various other debugging functions.

[0011] At least one exemplary embodiment of the invention provides a microprocessor comprising a multistage pipeline, a cache debug unit, a data pathway between the cache debug unit and an instruction cache unit, a data pathway between the cache debug unit and a data cache unit, and a data pathway between a unit of the multistage pipeline and the cache debug unit.

[0012] At least one additional exemplary embodiment provides a microprocessor comprising a multistage pipeline, a data cache unit, an instruction cache unit, and a unified...
cache debug unit operatively connected to the data cache unit, the instruction cache unit, and the multistage pipeline.

Yet another exemplary embodiment of this invention provides a RISC-type microprocessor comprising a multistage pipeline, and a cache debug unit, wherein the cache debug unit comprises an interface to an instruction cache unit of the microprocessor, and an interface to a data cache unit of the microprocessor.

Other aspects and advantages of the invention will become apparent from the following detailed description, taken in conjunction with the accompanying drawings, illustrating by way of example the principles of the invention.

**BRIEF DESCRIPTION OF THE DRAWINGS**

**FIG. 1** is a block diagram illustrating a processor core in accordance with at least one exemplary embodiment of this invention; and

**FIG. 2** is a block diagram illustrating an architecture for a unified cache debug unit for a microprocessor in accordance with at least one embodiment of this invention.

**DETAILED DESCRIPTION OF THE DISCLOSURE**

The following description is intended to convey a thorough understanding of the invention by providing specific embodiments and details involving various aspects of a new and useful microprocessor architecture. It is understood, however, that the invention is not limited to these specific embodiments and details, which are exemplary only.

It further is understood that one possessing ordinary skill in the art, in light of known systems and methods, would appreciate the use of the invention for its intended purposes and benefits in any number of alternative embodiments, depending upon specific design and other needs.

Discussion of the invention will now be made by way of example in reference to the various drawing figures. **FIG. 1** illustrates in block diagram form, an architecture for a microprocessor core and peripheral hardware structure in accordance with at least one exemplary embodiment of this invention. Several novel features will be apparent from **FIG. 1** which distinguish the illustrated microprocessor architecture from that of a conventional microprocessor architecture. Firstly, the microprocessor architecture of **FIG. 1** features a processor core having a seven stage instruction pipeline. A fetch stage (PET) includes an instruction cache, a branch prediction unit (BPU) and connection to instruction ram and a cache debug unit (CDU). An align stage (ALN) is shown in **FIG. 1** following the fetch stage 110.

Because the microprocessor core shown in **FIG. 1** is operable to work with a variable bit-length instruction set, namely, 16-bits, 32-bits, 48-bits or 64-bits, the align stage formats the words coming from the fetch stage into the appropriate instructions. In various exemplary embodiments, instructions are fetched from memory in 32-bit words. Thus, when the fetch stage retrieves or fetches a 32-bit word at a specified fetch address, the entry at that fetch address may contain an aligned 16-bit or 32-bit instruction, an unaligned 16-bit instruction preceded by a portion of a previous instruction, or an unaligned portion of a larger instruction preceded by a portion of a previous instruction based on the actual instruction address. For example, a fetched word may have an instruction fetch address of 0x4, but an actual instruction address of 0x6. In various exemplary embodiments, the 32-bit word fetched from memory is passed to the align stage where it is aligned into a complete instruction. In various exemplary embodiments, this alignment may include discarding superfluous 16-bit instructions or assembling unaligned 32-bit or larger instructions into a single instruction. After completely assembling the instruction, the N-bit instruction is forwarded to the decoder (DEC). 130.

Still referring to **FIG. 1**, an instruction extension interface is also shown which permits interface of customized processor instructions that are used to complement the standard instruction set architecture of the microprocessor. Interfacing of these customized instructions occurs through a timing registered interface to the various stages of the microprocessor pipeline in order to minimize the effect of critical path loading when attaching customized logic to a pre-existing processor pipeline. Specifically, a custom opcode slot is defined in the extensions instruction interface for the specific custom instruction in order for the microprocessor to correctly acknowledge the presence of a custom instruction as well as the extraction of the source operand addresses that are used to index the register file. The custom instruction flag interface is used to allow the addition of custom instruction flags are used by the microprocessor for conditional evaluation, even using the standard condition code evaluators. This custom extension condition code evaluators in order to determine whether the instruction is executed or not based upon the condition evaluation result (EXEC) 150. A custom ALU interface permits user defined arithmetic and logical extension instructions the result of which are selected in the result select stage.

Another novel feature of the microprocessor architecture illustrated in **FIG. 1** is the fast results forwarding block in the execute stage of the instruction pipeline. The fast result forwarding block selects the relevant results from a group of simple execution units (comprised of the Normalizing Unit, Barrel Shifter, Logical Unit and Fast Adder) of the execute stage to be written directly to the register file on the same output clock pulse, reducing the number of required clock cycles for non-computationally intensive operations. More complex arithmetic instructions that require an entire cycle to compute their results forward the results in the write back stage (WB) through the select stage (SEL) that contains a results selector that is used to select the correct output from multiple arithmetic units.

With continued reference to **FIG. 1**, yet another novel feature of the microprocessor architecture shown in this figure is the inclusion of a cache debug unit (CDU) shown in the example of **FIG. 1** as connected to the fetch stage of the instruction pipeline. Throughout this specification and claims the cache debug unit will be referred to as a unified cache debug unit. In various embodiments, the unified cache debug unit architecture serves as a debug unit for both an instruction cache and a data cache of the microprocessor.

Referring now to **FIG. 2**, an exemplary architecture of a cache debug unit (CDU) such as that depicted in
FIG. 1 is illustrated. In general, the cache debug provides a facility to check if certain things are stored in cache and to selectively change the contents of cache memory. Under certain circumstances it may be necessary to flush cache, pre-load cache, or to look at or change certain locations in a cache based on instructions or current processor pipeline conditions.

[0024] As noted herein, in a conventional microprocessor architecture employing cache debug, a portion of each of the instruction cache and data cache will be allocated for debug logic. Usually, however, these debug functions are performed off line, rather than at run time, and/or are expected to be slow. Furthermore, there are strong similarities to the debug functions in both the instruction cache and the data cache causing redundant logic to be employed in the processor design, thereby increasing costs and complexity of the design. Although the debug units are seldom used during runtime, they consume power even when not being specifically invoked because of their inclusion in the instruction and data cache components themselves.

[0025] In various exemplary embodiments, this design drawback of conventional cache debug units is overcome by a unified cache debug unit 200, such as that shown in FIG. 2. The unified cache debug unit 200 ameliorates at least some of these problems by providing a single unit that is located separately from the instruction cache 210 and data cache 220 units. In various exemplary embodiments, the unified cache debug unit 200 may interface with the instruction pipeline through the auxiliary unit 240. In various embodiments, auxiliary unit 240 interface allows the requests to be sent to the CDU 200 and responses to such requests to be received from the CDU 200. These are labeled as Aux request and Aux response in FIG. 2. In the example shown in FIG. 2, a state control device 250 may dictate to the CDU 200 the current state, such as in the event of pipeline flushes or other system changes which may preempt a previous command from the auxiliary unit 240.

[0026] As shown in the exemplary embodiment illustrated in FIG. 2, the instruction cache 210 is comprised of an instruction cache RAM 212, a branch prediction unit (BPU) 214 and a multi-way instruction cache (MWIC) 216. In various embodiments, the CDU 200 communicates with the instruction cache RAM 212 through the BPU 214 via the instruction cache RAM access line 201 labeled IS RAM Access. In various embodiments, this line only permits contact between the CDU 200 and the instruction cache RAM 212. Calls to the external memory subsystem 230, are made through the multi-way instruction cache (MWIC) 216, over request fill line 202. For example, if the CDU 200 needs to pull a piece of information from the memory subsystem 230 to the instruction cache RAM 212 the path through the request fill line 202 is used.

[0027] With continued reference to FIG. 2, in various exemplary embodiments, the structure of the data cache 220, in some respects mirrors that of the instruction cache 210. In the example illustrated in FIG. 2, the data cache 220 is comprised of a data cache RAM 222, a data cache RAM control 224 and a data burst unit 226. In various exemplary embodiments, the CDU 200 communicates with the data cache RAM 222 through the data cache RAM control 224 via the data cache RAM access line 203. In various embodiments, this line may permit communication between the CDU 200 and the data cache RAM 222 only. Thus, in various embodiments, calls to the external memory subsystem 230 through the data cache 220, are made through the data burst unit (DBU) 226, over fill/flush request line 204. Because, in various embodiments, the data cache 220 may contain data not stored in the memory subsystem 230, the CDU 200 may need to take data from the data cache 220 and write it to the memory subsystem 230.

[0028] In various exemplary embodiments, because the CDU 200 is located outside of both the instruction cache 210 and the data cache 220, the architecture of each of these structures is simplified. Moreover, because in various exemplary embodiments, the CDU 200 may be selectively turned off when it is not being used, less power will be consumed than with conventional cache-based debug units which receive power even when not in use. In various embodiments, the cache debug unit 200 remains powered off until a call is received from the auxiliary unit 240 or until the pipeline determines that an instruction from the auxiliary unit 240 to the cache debug unit 200 is in the pipeline. In various embodiments, the cache debug unit will remain powered on until an instruction is received to power off. However, in various other embodiments, the cache debug unit 200 will power off after all requested information has been sent back to the auxiliary unit 240. Moreover, because conventional instruction and data cache debug units have similar structure, reduction in total amount of silicon may be achieved due to shared logic hardware in the CDU 200.

[0029] While the foregoing description includes many details and specificities, it is to be understood that these have been included for purposes of explanation only. The embodiments of the present invention are not to be limited in scope by the specific embodiments described herein. For example, although many of the embodiments disclosed herein have been described with reference to cache debug unit in an RISC-type embedded microprocessor, the principles herein are equally applicable to cache debug units in microprocessors in general. Indeed, various modifications of the embodiments of the present inventions, in addition to those described herein, will be apparent to those of ordinary skill in the art from the foregoing description and accompanying drawings. Thus, such modifications are intended to fall within the scope of the following appended claims. Further, although the embodiments of the present inventions have been described herein in the context of a particular implementation in a particular environment for a particular purpose, those of ordinary skill in the art will recognize that its usefulness is not limited thereto and that the embodiments of the present inventions can be beneficially implemented in any number of environments for any number of purposes. Accordingly, the claims set forth below should be construed in view of the full breadth and spirit of the embodiments of the present inventions as disclosed herein.

1. In a microprocessor, a microprocessor core comprising:
   a multistage pipeline;
   a cache debug unit;
   a data pathway between the cache debug unit and an instruction cache unit;
   a data pathway between the cache debug unit and a data cache unit; and
a data pathway between a unit of the multistage pipeline and the cache debug unit.

2. The microprocessor according to claim 1, wherein the unit of the multistage pipeline is an auxiliary unit of an execute stage of the pipeline.

3. The microprocessor according to claim 1, further comprising a state control unit adapted to provide a current state of the pipeline to the cache debug unit.

4. The microprocessor according to claim 3, wherein a current state comprises at least one of a pipeline flush or other system change that preempts a previous command from the pipeline.

5. The microprocessor according to claim 1, further comprising a data pathway between the cache debug unit and a memory subsystem of the microprocessor through each of the instruction cache and data cache units.

6. The microprocessor according to claim 1, further comprising a power management control adapted to selectively power down the cache debug unit when not in demand by the microprocessor.

7. The microprocessor according to claim 1, wherein the microprocessor core is a RISC-type embedded microprocessor core.

8. A microprocessor comprising:
   a multistage pipeline;
   a data cache unit;
   an instruction cache unit; and
   a unified cache debug unit operatively connected to the data cache unit, the instruction cache unit, and the multistage pipeline.

9. The microprocessor according to claim 8, wherein the unified cache debug unit is operatively connected to the multistage pipeline through an auxiliary unit in an execute stage of the multistage pipeline.

10. The microprocessor according to claim 8, further comprising a state control unit adapted to provide a current state of the pipeline to the unified cache debug unit.

11. The microprocessor according to claim 10, wherein a current state comprises at least one of a pipeline flush or other system change that preempts a previous command from the multistage pipeline.

12. The microprocessor according to claim 8, further comprising a data pathway between the unified cache debug unit and a memory subsystem of the microprocessor through each of the instruction cache and data cache units.

13. The microprocessor according to claim 8, further comprising a power management control adapted to selectively power down the cache debug unit when not in demand by the microprocessor.

14. The microprocessor according to claim 8, wherein the architecture is a RISC-type embedded microprocessor architecture.

15. A RISC-type microprocessor comprising:
   a multistage pipeline; and
   a cache debug unit, wherein the cache debug unit comprises:
   an interface to an instruction cache unit of the microprocessor; and
   an interface to a data cache unit of the microprocessor.

16. The microprocessor according to claim 15, further comprising an interface between the cache debug unit and at least one stage of the multistage pipeline.

17. The microprocessor according to claim 16, wherein the at least one stage of the multistage pipeline comprises an auxiliary unit of an execute stage of the multistage pipeline.

18. The microprocessor according to claim 15, further comprising a state control unit adapted to provide a current state of the multistage pipeline to the cache debug unit.

19. The microprocessor according to claim 18, wherein a current state comprises at least one of a pipeline flush or other system change that preempts a previous command from the unit of the multistage pipeline.

20. The microprocessor according to claim 15, further comprising an interface between the cache debug unit and a memory subsystem through each of the instruction cache and data cache units.

21. The microprocessor according to claim 15, further comprising a power management control adapted to selectively power down the cache debug unit when not in demand by the multistage pipeline.