Title: SPLIT LEVEL CACHE

Abstract

A memory system for a data processor which includes a single-chip integer unit (10), an array processor such as a floating point unit (20), a main memory (27) and a split level cache. The split level cache includes an on-chip, fast data cache (12) with low latency for use by the integer unit (10) for loads and stores of integer and address data and an off-chip, pipelined global cache (14) for storing arrays of data such as floating point data for use by the array processor and integer and address data for refilling the data cache. Coherence between the data cache (12) and global cache (14) is maintained by writing through to the global cache during integer stores. Data cache words are invalidated when data is written to the global cache during an array processor store.
FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>AT</td>
<td>Austria</td>
<td>GB</td>
<td>United Kingdom</td>
<td>MR</td>
<td>Marshall Islands</td>
</tr>
<tr>
<td>AU</td>
<td>Australia</td>
<td>GE</td>
<td>Georgia</td>
<td>MW</td>
<td>Malawi</td>
</tr>
<tr>
<td>BS</td>
<td>Barbados</td>
<td>GN</td>
<td>Guinea</td>
<td>NE</td>
<td>Niger</td>
</tr>
<tr>
<td>BE</td>
<td>Belgium</td>
<td>GR</td>
<td>Greece</td>
<td>NL</td>
<td>Netherlands</td>
</tr>
<tr>
<td>BF</td>
<td>Burkina Faso</td>
<td>HU</td>
<td>Hungary</td>
<td>NO</td>
<td>Norway</td>
</tr>
<tr>
<td>BG</td>
<td>Bulgaria</td>
<td>IE</td>
<td>Ireland</td>
<td>NZ</td>
<td>New Zealand</td>
</tr>
<tr>
<td>BJ</td>
<td>Benin</td>
<td>IT</td>
<td>Italy</td>
<td>PL</td>
<td>Poland</td>
</tr>
<tr>
<td>BR</td>
<td>Brazil</td>
<td>JP</td>
<td>Japan</td>
<td>PT</td>
<td>Portugal</td>
</tr>
<tr>
<td>BY</td>
<td>Belarus</td>
<td>KE</td>
<td>Kenya</td>
<td>RO</td>
<td>Romania</td>
</tr>
<tr>
<td>CA</td>
<td>Canada</td>
<td>KG</td>
<td>Kyrgyzstan</td>
<td>RU</td>
<td>Russian Federation</td>
</tr>
<tr>
<td>CF</td>
<td>Central African Republic</td>
<td>KP</td>
<td>Democratic People's Republic of Korea</td>
<td>SD</td>
<td>Sudan</td>
</tr>
<tr>
<td>CG</td>
<td>Congo</td>
<td>CR</td>
<td>Croatia</td>
<td>SE</td>
<td>Sweden</td>
</tr>
<tr>
<td>CH</td>
<td>Switzerland</td>
<td>CY</td>
<td>Cyprus</td>
<td>SI</td>
<td>Slovenia</td>
</tr>
<tr>
<td>CI</td>
<td>Côte d'Ivoire</td>
<td>CZ</td>
<td>Czech Republic</td>
<td>SK</td>
<td>Slovakia</td>
</tr>
<tr>
<td>CM</td>
<td>Cameroun</td>
<td>DE</td>
<td>Germany</td>
<td>SN</td>
<td>Senegal</td>
</tr>
<tr>
<td>CN</td>
<td>China</td>
<td>DK</td>
<td>Denmark</td>
<td>TD</td>
<td>Chad</td>
</tr>
<tr>
<td>CS</td>
<td>Czechoslovakia</td>
<td>ES</td>
<td>Spain</td>
<td>TG</td>
<td>Togo</td>
</tr>
<tr>
<td>CZ</td>
<td>Czech Republic</td>
<td>FI</td>
<td>Finland</td>
<td>TJ</td>
<td>Tajikistan</td>
</tr>
<tr>
<td>DE</td>
<td>Germany</td>
<td>FR</td>
<td>France</td>
<td>TT</td>
<td>Trinidad and Tobago</td>
</tr>
<tr>
<td>ES</td>
<td>Spain</td>
<td>GA</td>
<td>Gabon</td>
<td>UA</td>
<td>Ukraine</td>
</tr>
<tr>
<td>FI</td>
<td>Finland</td>
<td>GB</td>
<td>United Kingdom</td>
<td>US</td>
<td>United States of America</td>
</tr>
<tr>
<td>FR</td>
<td>France</td>
<td>GR</td>
<td>Greece</td>
<td>UZ</td>
<td>Uzbekistan</td>
</tr>
<tr>
<td>GA</td>
<td>Gabon</td>
<td>HU</td>
<td>Hungary</td>
<td>VN</td>
<td>Viet Nam</td>
</tr>
</tbody>
</table>


SPLIT LEVEL CACHE

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates generally to memory systems utilized by high-speed data processors and more particularly relates to a memory hierarchy including high-speed data caches.

2. Description of the Relevant Art

Super-computers operating at billions of floating point operations (giga-flops) based on microprocessor architectures are being brought to the marketplace.

Although supercomputers are usually characterized by their impressive floating-point compute speeds, memory size and bandwidth also differentiates them from other computers. Each floating point operation (flop) requires between one and three memory accesses to executed. Accordingly, a several hundred megaflop microprocessor requires a memory hierarchy that can deliver gigabytes per second of bandwidth.

Pipelined memories are available that provide sufficiently high bandwidth, however, it is well-known that the latency associated with these memories directly impacts the execution speed of integer programs such as operating system and compilers as well as the scaler portion of floating point programs. These programs prefer to access integer and address data directly from a short latency memory device such as a small on-chip cache.

Accordingly, the conflicting demands of integer and floating point operations provide a major challenge to designing microprocessor based super-computers.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram of the invention;
Fig. 2 is a more detailed diagram of the system disclosed in Fig. 1;
Fig. 3 is a high-level diagram of the pipelined architecture of the off-chip global cache; Fig. 4 is a more detailed diagram of the pipeline depicted in Fig. 3; Fig. 5 is a diagram depicting the format of a physical address; Fig. 6 is a detailed diagram of a custom TAG RAM; Fig. 7 is a logical diagram of the on-chip data cache; and Fig. 8 is a block diagram of the store data paths.

SUMMARY OF THE INVENTION

The present invention is a split level cache including an a small, short latency data cache for storing integer and address data and a large, higher latency global cache for storing floating point data or data required by an array processor. The data cache is a proper subset of the global cache and is the primary cache for the integer unit. The global cache stores arrays of data for use by a floating point unit or an array processor and is the primary cache for the array processor and secondary cache for the integer unit.

According to one aspect of the invention, stores by the floating processor are written only to the global cache thereby causing lack of coherency with the data cache. This incoherency is tracked by attaching a valid bit to each word in a cache line stored in the data cache and resetting the valid bit corresponding to given word that is modified in the global cache during a floating point store operation.

According to another aspect of the invention, stores by the integer unit to the data cache are written through to the global cache to avoid incoherency. The valid bit corresponding to a word in the data cache modified by an integer store operation is set to indicate coherency.

According to another aspect of the invention, the global cache is set associative and includes odd and even banks of interleaved data stores to permit access of both banks simultaneously.
According to another aspect of the invention, the global cache is a pipelined cache and includes a set select signal stage that encodes a set select signal based on stored tags during a single machine cycle.

Other features and advantages will be apparent in view of the appended drawings and following detailed description.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

Fig. 1 is a block diagram of a preferred embodiment of the invention that utilizes a small high-speed, non-pipelined on-chip cache having low latency for providing integer and address data to in integer unit and a large off-chip, pipelined cache for providing floating point data to a floating point unit.

In Fig. 1, an integer unit (IU) 10 includes an on-chip first level data cache (D$) 12. An address output of the IU 10 is coupled to an off-chip second level cache (G$) 14 by an address bus 16 and a data input of the on-chip first level cache 12 is coupled to a data output port of the off-chip second level cache by a FILL bus 18.

An instruction port of the IU 10 is coupled to an instruction port of a floating point unit (FPU) 20 by a TBUS 22. Additionally the data in port of the FPU 20 is coupled to the data out port of the off-chip second level cache 14 by LOAD bus 24 and the data out port of the FPU 20 is coupled to the data input port of the off-chip second level cache 14 by a STORE bus 26. Additionally, the data input port of off-chip second level cache 14 is coupled the data input/output port of a main memory 27.

The G$ 14 includes a G$ tag store 28 and a G$ data store 30. The address bus 16 includes an OFFSET bus 160 for carrying low order address bits, an INDEX bus 161 for carrying higher order bits and a TAG bus 16T for the TAG field bits. The INDEX bus 161 and TAG bus 16T are coupled to the address inputs of G$ tag store 28 and the INDEX and OFFSET buses 16I and 160 are coupled to the address of the G$ data store 30. The set-select output of the G$ tag store 28 is coupled to a
set address input of the G$ data store 30 by a SELECT bus 34.

A control unit 35 generates timing and control signals for controlling data transfers. This control unit 35 is implemented utilizing standard technology and is not part of the invention.

Fig. 2 is a more detailed diagram of the embodiment depicted in Fig. 1. From Fig. 2, the IU 10 and FPU 20 are implemented as single chips with D$ 12 included on the IU. In the preferred embodiment the IU 10 is a RISC type processor that executes all data processing instructions on data stored in a register file 36. The processed data or data to be processed is transferred between the register file and memory by executing load and store operations.

An address generator (AGEN) 38 in the IU generates virtual addresses which are utilized to address the D$ 12 and are fed to a TLB 40 which generates the physical addresses in external main memory 27 for data loads and stores.

In the memory hierarch of the preferred embodiment, a virtual memory storing all data that can be accessed by the data processor is addressed by virtual addresses generated by the AGEN 38. The data held in the external main memory 27 is a subset of the data held in the virtual memory and is addressed by physical addresses generated by the TLB 40. The data held in the G$ 14 is a subset of the data held in main memory 27 and is addressed by the INDEX and OFFSET fields of the physical address output by the TLB 40 and a SELECT field output by the G$ tag store 28. The data held in the D$ 12 is a subset of the data held in the G$ 14. The D$ 12 is accessed by the INDEX and OFFSET fields of the virtual address generated by the AGEN 38 and is physically tagged.

As described more fully below, the G$ tag store 28 is implemented utilizing tag RAMS and the G$ data store 30 is implemented utilizing commodity synchronous static RAMS (SSRAMS) organized as a 4-way set associative cache with the select field output by the G$ tag store 28 selecting one of the four associative sets. Additionally, the G$ data store 30 is organized as an interleaved memory having odd and even
banks that can be simultaneously accessed to increase bandwidth.

The SSRAMS integrate input and output registers onto the chip, thus pipelining a RAM access into three cycles, i.e., address setup, RAM access, and data output cycles. Additional pipeline stages result from the address translation mechanism. A 4-way set associative tag RAM encodes set-information into 2 bits. Each data SSRAM chip is partitioned into four segments by high-order address bits which are fed from the tag RAM.

Thus, the D$ pipeline has five stages which are depicted at a high level in Fig. 3 and in greater detail in Fig. 4. The processing at each stage is completed in one machine cycle. In the diagrams the ordered G$ pipeline stages are labelled G, H, I, J, and K.

Referring now to Figs. 3 and 4, addresses are sent from the IU chip to the tag RAM chips in the G stage, the tags are looked up and hit/miss information is encoded into set-select information in the H stage, the encoded information from the tag RAMs 42 is fed to data RAMs in the I stage, the SSRAM 44 is accessed internally within the chip in the J stage, and the accessed data is sent back to the IU chip and FPU chip from the SSRAM chips 44 in the K stage.

A full cycle is allocated to chip crossings in the I and K stages because TTL drivers with significant loading at the target frequency of 75 MHz require almost an entire cycle.

The tag RAMs 42 utilized in the J stage of G$ pipeline will now be described with reference to Figs. 5 and 6. The general format of a physical address is depicted in Fig. 5. Each address includes a TAG field, an INDEX field, and an OFFSET field. As is well known in the art, the INDEX and OFFSET fields form the physical addresses of the cache with the INDEX field accessing a physical block in the cache and the OFFSET field accessing a line in that block. The entire physical main memory 27 is much larger than the cache and its addresses include the physical TAG field.

As stated above, the preferred embodiment is 4-way set associative meaning that a block in the external main
memory having a given INDEX field may be mapped into one of a set of 4 blocks in the cache accessed by the given INDEX. In the preferred embodiment the two higher order bits of the cache address are utilized as set-select data and the remaining address bit function as the index field. For each indexed location in the G$ store 30 four TAG fields are stored in the G$ TAG store 28 at a location accessed by an INDEX field.

Referring to Fig. 6, an INDEX register 50 is coupled to the ADDR input of a TAG RAM 52 and DIRTY BIT RAM 54 by an INDEX bus 56. A TAG register 58 is coupled to the first input of each of four comparators 60A-D by a TAG bus 62. The second input of each comparator 60A-D is coupled, respectively, to a TAG-OUT output of the TAG store by a TAG-OUT bus 64. The outputs of each comparator 60A-D are routed to the inputs of an ENCODER 66 and a NOR gate 68. The output of the ENCODER 66 is routed to a SET-SELECT register 70 and the output of the NOR gate 68 is routed to a MATCH register 72. A cache read/write signal (RWSA) is stored in a RWSA register 74 and provided to DIRTY BIT WRITE LOGIC 76 by an RWSA bus 78. The outputs of the ENCODER and NOR gates 66 and 68 are also routed the DIRTY BIT WRITE LOGIC 76.

When a given physical address is provided by the IU 10 its TAG field is stored in the TAG register and its INDEX field is stored in an INDEX register. The storage location in the TAG store 52 accessed by the INDEX field stores the TAG field associated with cache data stored in each the 4 locations in the associative set that can be accessed by the INDEX field. The TAG RAM 52 is organized so that the TAG field of data stored in the first associative set is provided to the second input of first comparator 60A, of the second associative set to the second comparator 60B an so on.

If the TAG data supplied to the second output of any comparator 60A-B matches the TAG field stored in the TAG register 58 then the comparator output is logic "1" and the comparator output is logic "0" otherwise.

The NOR gate 68 receives the comparator outputs and generates a "0" only if one of its inputs is not zero. Thus,
a "0" output by the NOR gate indicates a hit, i.e., that the block of data specified by the entire physical address is stored in the G$ 14.

When the cache is hit, the ENCODER 66 outputs a two-bit signal specifying the set storing the data specified by the given entire physical address. This set select signal functions as the two high-order bits given to the SSRAM chips forming the G$ store 30. Thus, as described above, the generation of hit/miss data and set-specifying information is performed at the H stage during a single machine cycle.

As is well known in the art, if data specified by the entire physical address is not stored in the cache then the data from the external main memory 27 is written to the cache and replaces data previously stored in the cache. If the replaced data in the cache has not been modified since it was transferred from the external main memory 27 to the cache then the data in the cache and main memory 27 are the same and the data is coherent. Accordingly, the cache data can be replaced without problem because the same data is stored in the main memory 27 and can be accessed when needed. However, if the replaced data in the cache has been modified by a write operation then it is no longer coherent with the data stored in main memory 27. Thus, before replacing the data in the cache the cache data is stored in a writeback operation so that the correct data is stored in the main memory 27 and can be accessed by the processor. Accordingly, when data in the cache is modified by a write operation a "dirty bit" associated with the data is cleared to indicate that the data is "dirty", i.e., is no longer coherent with data stored in the main memory.

In Fig. 6, whenever the RWSA signal indicates that data is to be written to the cache and the MATCH signal indicates a cache hit, a dirty bit stored in a DIRTY BIT RAM storage location accessed by the INDEX field is cleared. Thus, the dirty bit is processed in the J pipeline stage during a single machine cycle.
The system described in Fig. 6 is utilized to indicate lack of coherence between data stored in the G$ 14 and the main memory 27. However, because of unique organization of the split level cache a system for indicating lack of coherence between the D$ 12 and G$ 14 is also required.

As described above, the IU 10 utilizes the D$ 12 as the primary cache and G$ 14 as a secondary cache and the FPU 20 utilizes the G$ 14 as the primary cache and does not utilize the D$ 12 at all. If the same data is stored in both caches 12 and 14, there are two sources of incoherency. The first source is a modification of only D$ data by an IU write to the D$ 12 and the second source is a modification of only G$ data by an FPU write to the G$ 14.

The first source of incoherency is solved by a write through operation of D$ data writes to the G$ 14. The write through operation is possible because of the very high write bandwidth of the G$ 14. No external buffer is required because the G$ 14 can absorb write through at full bandwidth.

The second source of incoherency is unique to the split level cache system and results in incoherence between integer and floating point data. The problem is that while IU loads and stores access both the D$ 12 and G$ 14, FPU loads and stores only access the G$ 14. Thus, for example, if a particular memory location is first written by an IU store and then rewritten by an FPU store then a subsequent IU load would get stale data from the D$ 14 unless something is done.

Although a direct solution would be to invalidate a cache line in the D$ 12 whenever an FPU store occurred, the problem is further complicated by the fact that each cache line holds several words and a particular cache line may contain a mix of floating point data and integer data. If a program first accesses integer character data causing the particular cache line to be loaded to the D$ 12 then a subsequent floating point store would invalidate the entire line. However, in that case the integer data in the particular cache line in D$ 12 would still be valid but, since the line has been invalidated, another reference to the
integer data in the given line would cause another cache miss 
requiring that the particular cache line again be loaded to 
the D$. Mixed integer and floating point structures are 
common, particularly in graphics applications, so this kind of 
cache thrashing cannot be tolerated. Thus, invalidating the 
entire D$ line would lead to poor performance. 

The solution utilized in the preferred embodiment is 
to attach a valid bit to each word in every cache line in the 
D$ 12 thereby providing finer granularity to the valid bits. 

During an FPU store operation, the address is provided to both 
the D$ 12 and G$ 14. If the addressed data is in the D$ the 
valid bit attached to the addressed word is cleared to 
indicate lack of coherence between the modified word in the D$ 
12 and G$ 14. The valid bit attached to word accessed during 
an IU store is set to indicate coherence between the caches 12 
and 14 due to the above-described write through operation. 

This solution allows the particular line in D$ to 
store contain only integer data because any floating point 
data which had been modified by an FPU write would have a 
cleared valid bit attached thereto. Accordingly, the above 
described thrashing problem would be obviated. 

The clearing of the valid bit in D$ 12 when storing 
floating point data in the G$ 14 will now be described with 
reference to Fig. 7 which is a detailed logical diagram of the 
D$ 12. 

The clearing of the valid bit in the D$ 12 during a 
floating point store operation will now be described with 
reference to Fig. 8. In the preferred embodiment, the D$ 12 
is dual-ported to support either two loads or one load and one 
store per cycle. The D$ 12 is 16KB direct-mapped cache with 
each line holding 32 bytes organized into 8 words. The D$ 12 
is refilled from the G$ 14 in 7 machine cycles. The D$ is 
virtually addressed and physically tagged and the D$ 12 is a 
proper subset of the G$ 14. 

In Fig. 8 the physical address output by the TLB 40 
of the IU 10 is held in a first holding register 90. The D$ 
12 is shown logically divided into a VRAM 12T and a data store 
12D. Each cache line 92 in the data store 12D includes 8 word
positions each holding four bytes of data. The INDEX field of the virtual address accesses a TAG field and a MASK field from a storage location in the VRAM 12V and a cache line in the data store 12D. The OFF field of the virtual address specifies a particular word position in the cache line 92. The TAG field and MASK field read from the VRAM 12V are held in a second holding register 94. The TAG field accessed from the VRAM 12V is the TAG field included in the physical address of the data held in the cache line 92 and the accessed MASK field holds eight valid bits, each attached to a corresponding word in the cache line 92. The TAG field held in the second holding register 94 is provided to the first input of a comparator 96 and the TAG field of the physical address is provided to the second input of the comparator 96. A MASK BIT CLEARING LOGIC 98 is coupled to the MASK storing part of the second holding logic, the OFF field of the virtual memory, and the output of comparator 96.

During the floating point store operation to a location specified by the physical address held in the first holding register 90, the TAG field and MASK field are accessed from the VRAM 12V during a first machine cycle. If the accessed TAG field matches the TAG field of the physical address the comparator output enables the MASK BIT CLEARING LOGIC to clear the VB held in the second holding register 94 which corresponds to the word in the cache line 92 specified by the OFF field of the virtual address. This match indicates the floating point data will be modified in the G$ 14 and that the word of data specified by the virtual address will no longer be coherent, thus, the clearing of the VB in the MASK invalidates this word in the D$ 12. During the next clock cycle the modified MASK field including the cleared VB is written back to the storage location in the VRAM 12V specified by the INDEX field virtual address.

The write through operation for integer data stored in the D$ 12 will now be described in detail with reference to Fig. 9. In the preferred embodiment, floating point instructions, addresses, and data are enqueued to compensate for the latency of the G$ 14 and the FPU 20. Additionally,
there is no direct store data path from the IU 10 to the G$ 14. Accordingly, for an IU store operation the D$ 12 transfers the data stored to an FP instruction queue 100 and the data is dispatched to the FPU 20 over the TBUS 22 and encoded as write-through data. Additionally, the physical address of the write-through data is transferred to a store address queue 102. The FPU 20 recognizes the write-through data and bypasses it to a store data queue 104. The timing of the queues is arranged so that the address of the write-through data and the write-through data are fed to the G$ 14 store during the same machine cycle.

The invention has now been described with reference to the preferred embodiments. However, modifications and substitutions will now be apparent to persons of skill in the art. For example, in the preferred embodiment the D$ is implemented on the same chip as the IU and G$ is implemented off-chip utilizing commodity SSRAMs. However, as technology changes it will be possible to implement a small short latency cache off chip and a large, higher latency cache on chip.

However, the principles of the invention would still be applicable to such alternative arrangements. Accordingly, it is not intended to limit the scope of the claims except as provided by the appended claims.
WHAT IS CLAIMED IS:

1. In a data processor including an integer unit and a floating point unit having a data input port and a data output port, with a virtual memory, accessed by a virtual address, holding all data for use by the data processor and an external main memory, accessed by a physical address, holding a subset of the data held in the virtual memory, and with the integer unit including an address generator for generating said virtual addresses and a translator for translating virtual addresses into physical addresses, an improved split level cache memory system comprising:

   a first memory cache, having a data output port coupled to said integer unit, a data input port, and an address input port coupled to receive part of said virtual memory, for storing integer and address data to be processed by said integer unit;

   a second memory cache, having a data input port, a data output port, and an address input port coupled to said integer unit to receive said physical address, for storing said integer and address data to be processed by said integer unit and for storing floating point data;

   first data transfer means, coupled to the data output port of said second memory cache and the data input port of said first memory cache and the data input of the floating point unit, for transferring data from said second memory cache to said first memory cache when integer and address data in said second memory cache is required for processing by said integer unit and between said second memory cache and said floating point unit when floating point data is required for processing by said floating point unit; and

   second data transfer means, coupled to data output port of floating point unit and the data input port of the second memory cache, for transferring data from said floating point unit to said second memory cache during a floating point store operation.
2. In a data processor including an integer unit fabricated on a single chip and a floating point unit having a data input port and a data output port, with a virtual memory, accessed by a virtual address, holding all data for use by the data processor and an external main memory, accessed by a physical address, holding a subset of the data held in the virtual memory, and with the integer unit including an address generator for generating said virtual addresses and a translator for translating virtual addresses into physical addresses, an improved split level cache memory system comprising:

a first memory cache, fabricated on said single chip, having a data output port coupled to said integer unit, a data input port, and an address input port coupled to receive part of said virtual memory, for storing integer and address data to be processed by said integer unit;

a second memory cache, separate from the single chip, and having a data input port, a data output port, and an address input port coupled to said integer unit to receive said physical address, for storing said integer and address data to be processed by said integer unit and for storing floating point data;

first data transfer means, coupled to the data output port of said second memory cache and the data input port of said first memory cache and the data input of the floating point unit, for transferring data from said second memory cache to said first memory cache when integer and address data in said second memory cache is required for processing by said integer unit and between said second memory cache and said floating point unit when floating point data is required for processing by said floating point unit; and

second data transfer means, coupled to data output port of floating point unit and the data input port of the second memory cache, for transferring data from said floating point unit to said second memory cache during a floating point store operation.
3. The split level cache of claim 2 wherein each virtual address and physical address includes INDEX and OFFSET fields and a first copy of a specific cache line is stored in the first cache memory and a second copy of the specific cache line is stored in the second cache memory, with the specific cache line including a plurality of words including integer data and floating point data, where said first memory cache further comprises:

   a first data store for storing said first copy of the specific cache line at a storage location accessed by said INDEX field of said virtual address;

   and wherein said second memory cache further comprises:

   a second data store for storing said second copy of the specific cache line at a storage location accessed by said INDEX field of said physical address;

   and further comprising:

   means for modifying a particular word of integer data included in the first copy of said specific cache line during an integer unit store operation; and

   means, coupled to said second data transfer means, for writing the modified particular word through to the particular word in the second copy of the specific cache line stored in said second data store so that the particular word in both copies of the given cache line are coherent.

4. The split level cache of claim 3 further comprising:

   means for attaching a valid bit to each word in said first copy of the specific cache line stored in said first data store;

   means for resetting said valid bit attached to a floating point word in said first copy of the specific cache line if the same floating point word in the second copy of the cache line is modified by a floating point store operation; and
means for setting said valid bit attached to an
integer word in said first copy of the specific cache line if
the integer word is modified by an integer store operation.

5. The split level cache of claim 1 wherein said
physical address includes TAG and INDEX fields and said second
cache memory includes a data store and a select signal
generator, wherein:

said data store is an \( N \)-way set associative memory,
where \( N \) is an integer equal to a power of 2, with a given set
selected by a set select signal and a given cache line in a
set selected by the INDEX field of the physical address;
and wherein said select signal generator includes:

\[ \text{a tag store for providing } N \text{ TAG fields accessed by } \]
\[ \text{the INDEX field, with each TAG field corresponding to data } \]
\[ \text{held in one of the sets of said data store;} \]

\[ N \text{ comparators, each comparator having a first input } \]
coupled to receive the TAG field of said physical address and
a second input coupled to receive a respective one of said TAG
fields provided by said TAG store, with each comparator having
an output signal which is set if the TAG fields at the
comparator inputs match and is reset if the TAG fields at the
comparator inputs do not match; and

\[ \text{encoder means, coupled to receive the output signals } \]
from said comparators, for generating a select signal encoding
the position of a comparator having an output signal
indicating that the TAG fields at its inputs match.
FIG. 1.

EXTERNAL CACHE PIPELINE

FIG. 3.

SUBSTITUTE SHEET (RULE 26)
FIG. 6.

SUBSTITUTE SHEET (RULE 26)
FIG. 7.
INTERNATIONAL SEARCH REPORT

A. CLASSIFICATION OF SUBJECT MATTER

IPC(6): G06F 12/08, 13/00
US CL: 395/425, 375

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S.: 395/375, 400, 425

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

Please See Extra Sheet.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>US, A, 5,051,885 (YATES, JR. ET AL) 24 September 1991, see column 3, lines 19-43; column 5, line 40 to column 6, line 18 and Figure 1.</td>
<td>1-5</td>
</tr>
<tr>
<td>Y</td>
<td>US, A, 5,214,765 (JENSEN) 25 MAY 1993, see column 1, line 17 to column 2, line 14; column 3, line 10 to column 7, line 25, and Figure 1.</td>
<td>1-5</td>
</tr>
<tr>
<td>Y</td>
<td>Hot Chips V Symposium, Stanford University, August 08-10, 1993, pp. 8.1.1-8.1.9.</td>
<td>1-2</td>
</tr>
<tr>
<td>Y</td>
<td>US, A, 4,928,225 (MCCARTHY ET AL) 22 MAY 1990, see column 2, lines 19-26 and 56-63; column 6, lines 40+, and Figures 2 and 3.</td>
<td>3-5</td>
</tr>
</tbody>
</table>

Further documents are listed in the continuation of Box C. See patent family annex.

Date of the actual completion of the international search: 24 MARCH 1995

Date of mailing of the international search report: 03 MAY 1995

Name and mailing address of the ISA/US Commissioner of Patents and Trademarks

Box PCT
Washington, D.C. 20231

Authorized officer: GLENN GOSSAGE

Facsimile No. (703) 305-3230

Telephone No. (703) 305-3820

Form PCT/ISA/210 (second sheet)(July 1992)
<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>US, A, 5,014,195 (FARRELL ET AL) 07 MAY 1991, see the abstract; column 2, lines 50-66; column 4, line 30 to column 5, line 17 and Figure 2.</td>
<td>5</td>
</tr>
<tr>
<td>A</td>
<td>US, A, 4,774,659 (SMITH ET AL) 27 September 1988, see entire document.</td>
<td>1-5</td>
</tr>
<tr>
<td>A</td>
<td>US, A, 5,155,816 (KOHN) 13 October 1992, see entire document.</td>
<td>1-5</td>
</tr>
<tr>
<td>A</td>
<td>EP, A, 0459232 (NATIONAL SEMICONDUCTOR CORP.) 04 December 1991, see entire document.</td>
<td>1-5</td>
</tr>
</tbody>
</table>
B. FIELDS SEARCHED
Electronic data bases consulted (Name of data base and where practicable terms used):

USPTO Automated Patent System (APS), files USPAT and JPOABS
ORBIT, files WPI, WPI and JAPIO

Search terms: cache, floating point or FPU or FP, integer, modify, write through, valid bit, tag, index, offset, set associative, comparator, encode