

**CACHE MEMORY ARRANGEMENT WITH WRITE BUFFER PIPELINE
PROVIDING FOR CONCURRENT CACHE DETERMINATIONS**

BACKGROUND OF THE INVENTION

1. Field of the Invention

This invention relates to computer systems and, more particularly, to
5 methods and apparatus for reducing the access time for implementing store
operations in a computer system.

2. History of the Prior Art

In computer systems, the access of main memory to retrieve information
10 often takes a substantial portion of the operational time of the system. For this
reason, the use of a cache memory to increase system speed has become
prevalent in more advanced systems. A cache memory makes use of a
relatively small amount of fast random access memory in which recently used
instructions and data are stored as they are used by a processor. Such
15 instructions and data are then available in the cache to be accessed by the
associated processor more rapidly than they might be in main memory. The
basic theory of caching is that, in general, information which has been recently
used is more likely to be used sooner than is other information. The cache
memory is often both physically faster than the random access memory used for
20 main memory and is arranged so that it may be addressed more rapidly than
main memory. For example, in virtual memory systems, a cache does not
need to go through the look-up operation of a memory management unit to
determine the physical addresses necessary for accessing main memory.
Such caching arrangements have operated to greatly increase the speed of
25 operation of computer systems for certain types of information.

There are many forms of cache memory which may be provided for use with a particular processor. In most of these forms of cache memory, information from a number of different physical locations in main memory may map to the same location in cache memory. Consequently, after some finite
5 period of time, information stored in a position in cache memory may have to be replaced by other information needed by the processor. Some cache memory systems (called write-back systems) do not transfer information written to the cache memory to main memory as soon as it is placed in the cache memory but retain that information until a later more convenient time. Such systems tend to
10 make less use of the system bus and are thus faster.

In a write-back caching arrangement, valid information residing at a position in the cache in which new information is to be stored may not yet have been transferred to main memory so that the storage of the new information
15 would cause the loss of the old information. Thus, data cannot be stored in the cache on a cache miss. For this reason, a store instruction typically requires a check for a cache miss; if there is not a miss, then the accessed information is in the cache, and a store may be accomplished. This check requires an extra clock cycle of operation and causes substantial delay in a pipelined computer
20 system in which it is expected that a command will be executed on each clock cycle.

SUMMARY OF THE INVENTION

It is therefore, an object of the present invention to accelerate the operation of a computer system.

It is another, more specific, object of the present invention to reduce the delay attendant upon the execution of a store instruction caused by the need to address the cache to determine whether a cache miss will occur before storing information to the cache.

According to the present invention there is provided a computer system, comprising:

a cache memory;

means for buffering data and an associative address to be stored in the cache memory;

means for checking the cache memory to determine whether the cache memory holds the associative address, wherein the means for checking the cache memory generates a cache miss if the cache memory does not hold the associative address; and

means responsive to the cache miss for terminating a store operation of the cache memory before the data and the associative address leave the means for buffering.

These and other objects and features of the invention will be better understood by reference to the detailed description which follows taken together with the drawings in which like elements are referred to by like designations throughout the several views.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a block diagram of a cache memory arrangement constructed in accordance with the invention.

5

Figure 2 is a timing diagram illustrating the sequence of events occurring in a typical operation using the invention.

10

NOTATION AND NOMENCLATURE

Some portions of the detailed descriptions which follow may be presented in terms of algorithms and symbolic representations of operations on data bits within a computer memory. These algorithmic descriptions and
15 representations are the means used by those skilled in the data processing arts to most effectively convey the substance of their work to others skilled in the art. An algorithm is here, and generally, conceived to be a self-consistent sequence of steps leading to a desired result. The steps are those requiring physical manipulations of physical quantities. Usually, though not necessarily,
20 these quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated. It has proven convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers, or the like. It should be borne in mind, however, that all of these and
25 similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities.

Further, the manipulations performed are often referred to in terms, such as adding or comparing, which are commonly associated with mental operations performed by a human operator. No such capability of a human operator is necessary or desirable in most cases in any of the operations described herein which form part of the present invention; the operations are machine operations. Useful machines for performing the operations of the present invention include general purpose digital computers or other similar devices. In all cases the distinction between the method of operations in operating a computer and the method of computation itself should be borne in mind. The present invention relates to apparatus and to method steps for operating a computer in processing electrical or other (e.g. mechanical, chemical) physical signals to generate other desired physical signals.

DETAILED DESCRIPTION OF THE INVENTION

Referring now to Figure 1 there is illustrated in block diagram form a memory arrangement 10 constructed in accordance with the present invention. The arrangement 10 includes an array of random access memory 12 which in the preferred embodiment is constructed of very fast memory devices. The random access memory 12 is utilized as cache memory for at least some portion of a computer system. Associated with the random access memory 12 is a control circuit 14. The control circuit 14 of the preferred embodiment may be a state machine adapted to produce certain output signals in response to selected input signals in a manner well known to the prior art. The control circuit 14, for example, receives input signals representing addresses being

accessed, write control signals, and write cancel signals. The write signals are provided from a source of write signals 16 and are furnished directly to the control circuit 14. The address signals are furnished from a source such as a processor, not shown, to an address register 17. From the register 17 the address signals are used by a series of logic elements to provide control signals to the control circuit 14. The write cancel signals are furnished from circuitry 31 which determines whether or not a cache miss occurs on a store access of the random access memory 12.

10 The address register 17 is connected by a plurality of lines to a first address buffer register 20. The number of lines utilized is sufficient in the particular computer system to transfer all of the bits of an address in parallel to the first address buffer register 20. In like manner, the first address buffer register 20 is connected by a like number of lines to a second address buffer register 21. Each of the first address buffer register 20 and the second address buffer register 21 are constructed of circuitry adapted to store an address being accessed in the memory arrangement 10. It should be noted that any number of buffers might be used in the pipeline formed to produce the results produced by the invention with only minor variation in the circuitry.

20

 Data to be stored in the random access memory 12 is furnished to a data register 18 from a source such as a processor, not shown. In a manner similar to the addressing arrangement, the data register 18 is connected by a plurality of lines to a first data buffer register 23. The number of lines utilized is sufficient
25 In the particular computer system to transfer all of the bits of data to be stored in the random access memory 12 in a single store operation in parallel to the first

data buffer register 23. The first data buffer register 23 is connected by a like number of lines to a second data buffer register 24. Each of the first data buffer register 23 and the second data buffer register 24 are constructed of circuitry adapted to store the data being stored in the memory arrangement 10. Again, a larger or smaller number of data buffer registers might be utilized; but the invention is more easily described with only two stages.

A multiplexor 26 receives input directly from the address register 17 and from the second address buffer register 21. Similarly, a multiplexor 27 receives input directly from the data register 18 and from both the first data buffer register 23 and from the second data buffer register 24. Each of the multiplexors 26 and 27 furnishes input to the random access memory 12 so that information selected to be stored at a particular address in random access memory 12 may be stored therein in a manner well known to the prior art.

15

A pair of comparators 28 and 29 furnish the results of a comparison of address inputs to the control circuit 14. These comparators each receive input directly from the address register 17. The comparator 28 also receives the address stored in the first address buffer register 20 while the comparator 29 also receives the address stored in the second address buffer register 21.

20

The random access memory 12 furnishes a data out signal on a plurality of lines to a multiplexor 33. The multiplexor 33 also receives data directly from the multiplexor 27. The data transferred by the multiplexor 33 is provided to a data out register 34 which operates to provide sufficient time for utilization of the output of the random access memory 12.

25

As explained above, in prior art systems, a store instruction typically requires a check for a cache miss; if there is not a miss, then the accessed information is in the cache, and a store may be accomplished. This requires a
5 delay for a number of clock cycles of operation and causes substantial delay in a pipelined computer system in which it is expected that a command will be executed on each clock cycle. For example, it is expected that the invention will be utilized in a reduced instruction set computer (RISC) such as the Scalable Processor Architecture (SPARC) computers designed and
10 manufactured by Sun Microsystems, Inc., Mountain View, California.

In order to obviate the delay caused by a check for a cache miss in such systems, the present memory arrangement 10 utilizes the first address buffer register 20, the second address buffer register 21, the first data buffer register
15 23, and the second data buffer register 24 to provide a buffered input to the random access memory 12 so that a sufficient time is provided to allow the tag portion of the cache to determine whether a miss has occurred at the address accessed for a store operation. This buffered input allows the pipeline to continue without the halt normally required in prior art systems in order to check
20 for a cache miss and thereby substantially increases the speed of operation of the memory arrangement 10.

Referring again to Figure 1, in the case of a read operation, the address to be read is placed in the address register 17 during a first clock cycle. Since
25 the data is to be obtained from the cache or other memory, no data is placed in the data register 18. On the next clock cycle, the address in the address

register 17 is furnished by the multiplexor 26 to the random access memory 12. This is the normal path for a read operation.

In a store operation, on the other hand, the address is furnished to the
5 address register 17 and the data to be stored is placed in the data register 18 during a first clock cycle. It will be understood that in various schemes of operation, the data might lag the address by one clock cycle; however, to make understanding the operation easier, the data and the address for that data are considered to be furnished on the same clock cycle. On the next clock
10 cycle, the address and data are placed, respectively, in the first address buffer register 20 and the first data buffer register 23. It should be noted that only a store causes information to be placed in the first address and data buffer registers 20 and 23. Upon the appearance of the next store operation, the address in the first address buffer register 20 and the data in the first data buffer
15 register 23 are clocked into the second address buffer register 21 and the second data buffer register 24, respectively. This provides two clock cycles at the least in which the system may test to determine whether the address information is stored in the random access memory 12 before the new information is required to be placed at that address in the random access
20 memory 12. Then, upon the appearance of the next store operation, the address in the second address buffer register 21 and the data in the second data buffer register 24 are clocked into the random access memory 12 at the addressed position by a write enable signal from the control circuit 14 directed to the random access memory 12. If the address accessed for the store
25 operation is not present in the random access memory 12, however, the write enable signal is not furnished and the write operation is cancelled so that it will

not write over information at the addressed position of the random access memory 12.

The two comparators 28 and 29 are utilized to allow a read operation to occur in a situation in which a store to an address has taken place but the information is not yet in the random access memory 12 because it has not yet propagated through in the pipeline provided by the first data buffer register 23 and the second data buffer register 24. The address to be read in the address register 17 is compared by the comparators 28 and 29 with the addresses held in the register stages of the address pipeline. If, on a read operation, the address in the address register 17 compares with the address in either the first address buffer register 20 or the second address buffer register 21, then the control circuit 14 causes the address in the related first data buffer register 23 or second data buffer register 24 to be selected by the multiplexor 27 and furnished directly through the multiplexor 33 to the output register 34 without waiting for the information to be stored in the random access memory 12.

The operation of the memory arrangement 10 may be better understood by referring to the timing diagram of Figure 2. A number of system clock cycles from 1 through 9 are illustrated along the top of Figure 2. At the left edge of the figure are listed the various signals. The figure illustrates the condition of these signals during each of the clock cycles shown.

In Figure 2, it may be seen that a new address is furnished on the input address lines to the address register 17 on each clock cycle. The output of the address register 17 lags the input by one clock cycle. The write input signal is furnished in the example of Figure 2 to the control circuit 14 during clock cycles 2-4 because addresses 2, 3 and 4 are the addresses to be written. These signals indicate to the control circuit 14 that the data signals furnished at that time on the data lines as input to the data register 18 are to be stored in the random access memory 12. These signals also indicate to the control circuit 14 that the address associated with the data is to be clocked into the first address buffer register 20. Consequently, the addresses 2, 3, and 4 are sequentially clocked into the first address buffer register 20 from the address register 17 and the data associated with those addresses are clocked into the first data buffer register 23 from the data register 18 on clock cycles 4, 5, and 6, respectively.

It should be noted that the information in the first address and data buffer registers 20 and 23 during the clock cycles 1-3 is indeterminate and not important to the operation of the memory array 10 as is indicated by the question marks shown during those clock cycles for the particular buffer registers. This is true because the buffer registers act as a first-in first-out circuit and are only filled when a store cycle occurs. It is for this reason that the address 1 is not placed in the buffer pipeline. Thus, on clock cycle 5, the signals in the first address and data buffer registers 20 and 23 (address 2) are placed in the second address and data buffer registers 21 and 24 when the address 3 information is placed in the first address and data buffer registers 20 and 23. In the same manner, on clock cycle 6, the signals in the first address

and data buffer registers 20 and 23 (address 3) are placed in the second address and data buffer registers 21 and 24 when the address 4 information is placed in the first address and data buffer registers 20 and 23. Once the information is in the second address and data buffer registers 21 and 24, it is available to be written into the random access memory 12. It should be noted that the address 4 and data 4 signals are held in the first address and data buffer registers 20 and 23 after clock cycle 6 because there are no further store operations performed to cause those buffer registers to be necessary to later operations. In a like manner, the address 3 and data 3 signals are held in the second address and data buffer registers 21 and 24 after clock cycle 6 because there are no further store operations performed.

The address and data inputs to the random access memory 12 are illustrated as the A and D signals, respectively, in Figure 2. As explained above, when the operation is a read operation, the address input to the random access memory 12 is furnished by the multiplexor 26 directly from the address register 17. Thus, on clock cycles 2, 6, 7, and 8, the addresses in the address register 17 are furnished to the random access memory 12 by the multiplexor 26. Since no data is required on a read operation, the information shown for the D input is indicated by a question mark during these clock cycles.

On the other hand, information is stored in the random access memory 12 only upon the occurrence of a store operation. Since the store information is read by the multiplexors 26 and 27 from the second address and data buffer registers 21 and 24, respectively, and those buffer registers are not yet filled when the first store address appears in the address register at clock cycle 3, no

operation takes place even though a write enable signal is present at that clock cycle. The same thing happens at clock cycle 4. However, at clock cycle 5, address and data signals are present in the second address and data buffer registers 21 and 24, and this information is stored in the random access
5 memory 12 upon the occurrence of a write enable signal. The write cancel signal may occur two clock cycles after the store operation commences to give the system time to ascertain whether a miss has occurred at the address accessed for the store operation without causing any delay in the pipeline of the operation (the write cancel signal would be asserted in clock 4 to cancel
10 address 2). Consequently, the system operates at a substantially faster speed than do prior art systems.

The signal appearing on the address lines at clock cycle 8 should be noted. This is a read signal to address 3, an address which has not yet been
15 stored in the random access memory 12 but resides in the second address buffer register 21. This address in the address register is compared to the address in the second address buffer register 21 by the comparator 29 and causes the control circuit 14 to enable the multiplexor 27 and the multiplexor 33 to transfer the data in the second data buffer register 24 to the output register 34
20 in the manner described above. One special advantage of the invention is that the very small amount of hardware required to implement the invention allows its placement directly on the random access memory chips so that the complex data and address path interconnections are within that chip thereby decreasing the complexity and pin count for the rest of the system.

Although the present invention has been described in terms of a preferred embodiment, it will be appreciated that various modifications and alterations might be made by those skilled in the art without departing from the scope of the invention. The invention should therefore be measured
5 in terms of the claims which follow.

CLAIMS

1. A computer system, comprising:
a cache memory;
means for buffering data and an associative address to be stored in the cache memory;
means for checking the cache memory to determine whether the cache memory holds the associative address, wherein the means for checking the cache memory generates a cache miss if the cache memory does not hold the associative address; and
means responsive to the cache miss for terminating a store operation of the cache memory before the data and the associative address leave the means for buffering.
2. A computer system as claimed in claim 1 in which the means for buffering data to be stored in the cache memory comprises means for forming a pipeline of addresses to be accessed and data to be stored at such addresses.
3. A computer system as claimed in claim 2 in which the means for forming a pipeline of addresses to be accessed and data to be stored at such addresses comprises a first address buffer register for storing the address of information to be stored, and a first data buffer register for storing data to be stored at the address in such first address buffer register.
4. A computer system as claimed in claim 3 in which the means for forming a pipeline of addresses to be accessed and data to be stored at such addresses further comprises a second address buffer register for storing the address of information to be stored, and a second data buffer register for storing data to be stored at the address in such second address buffer register.



5. A computer system as claimed in claim 2 which further comprises an input address register, and means for selecting addresses from either the input address register or the pipeline of addresses to be accessed.

6. A computer system as claimed in claim 5 further comprising means for comparing address in the input address register to an address in the pipeline of addresses to be accessed to determine whether an address to be read exists in the pipeline of addresses to be accessed.

7. A computer system as claimed in claim 6 further comprising means for accessing data stored in the pipeline if an address to be read exists in the pipeline of addresses to be accessed.

