"APPARATUS FOR ACCELERATING STORE OPERATIONS IN A RISC COMPUTER"

BACKGROUND OF THE INVENTION

Field of the Invention:

This invention relates to computers and, more particularly, to methods and apparatus for accelerating store operations in reduced instruction set computers.

History of the Prior Art:

The development of digital computers progressed through a series of stages beginning with processors which were able to process only a few basic instructions and in which the programming needed to be done at a machine language level and continuing to processors capable of handling very complicated instructions written in high level languages. At least one of reasons for this development is that high level languages are easier for programmers and more programs are developed more rapidly. Another reason is that up to some point in the development the more advanced machines executed operations more rapidly.

There came a point, however, where the constant increase in the ability of the computers to run more complicated instructions actually began to slow the operation of the computer over what investigators felt was possible with machines operating with only a small number of basic instructions. These investigators began to design advanced machines for running a limited number of instructions, a so-called reduced instruction set, and were able to demonstrate that these machines did, in fact, operate more rapidly for some types of operations. Thus began the reduced instruction set computer which has become known by its acronym, RISC.

The central processing unit of the typical RISC computer is very simple. It fetches an instruction every clock cycle. In its simplest embodiment, all instructions except for load and store act upon internal registers within the central processing unit. A load instruction is used to fetch data from external memory and place it in an internal register, and a store instruction is used to take the contents of an internal register and place it in external memory. RISC processors obtain higher speeds of operation by pipelining the instructions. Processors utilized to provide pipelined operations normally cycle through fetch, decode, execute, and write back steps of operation in executing each instruction. In a typical pipelined system, the individual instructions are overlapped so that an instruction executes once each clock cycle of the system.

One design of RISC computer is based on the Scalable Process Architecture (SPARC) designed by Sun Microsystems, Inc., Mountain View, California, and implemented in the line of SPARC computers manufactured by that company. One salient feature of the SPARC computers is the thirty-two bit instruction word used. As with most computers, instructions have addresses which direct you to particular positions in the computer and commands which tell the computer what to do. Essentially, an instruction directs the processor to do this operation to the contents of the registers addressed by this instruction. In a SPARC computer, an instruction includes register addresses and commands. The normal instruction (called a register + register instruction) specifies two addresses of registers to be read, an operation to be performed on the data at those register addresses, and the address of a register in which the results of the operation are to be placed. Such instructions require that the

'SPARC' is a registered Trade Mark.

register file associated with the computer have two read ports and one write port. As with other RISC computers, in none of these normal instructions does the processor of the SPARC computer go off-chip; it simply operates with information in the register file.

However, a load or store instruction is used to get data from memory or store it to memory. Such load and store instructions thus require that the processor access memory off-chip. The SPARC instructions to accomplish this designate two register addresses at which data are found which are to be added to form a memory address (as contrasted to a register address). For a load instruction, the third address designates the register to which the data at the memory address is to be written. For a store address, however, the third address designates the register which holds the information which is to be written to the memory address. Thus, while most instructions in a SPARC computer are set up to read two registers and write to a third register, a store instruction reads three registers, two to determine a memory address, and the third to retrieve the data to be written to the memory address. Since the third register must be read rather than written to derive the data to be stored, this operation requires three read ports.

Adding a read port is expensive. For this reason, SPARC computers typically do a store operation in two cycles to obviate the need for an extra read port. During the first cycle, they read the first two registers and compute the memory address. During the second cycle, they read out the data from the register designated in the third address field using one of the two read ports. This, of course, slows the operation of the computer.

SUMMARY OF THE INVENTION

It is, therefore, an object of the present invention to increase the speed of a SPARC based RISC computer in handling store operations.

It is another more specific object of the present invention to eliminate the need for an average of two clock cycles of processor time to accomplish a store operation in a SPARC based RISC computer.

According to the present invention there is provided a processor for a computer which is capable of handling pipelined instructions each contain register addresses and which includes an instruction register and a register file having a plurality of registers and a first number of read ports less than the possible number of addresses which may be included in a single instruction, the improvement comprising means for storing one of the register addresses of an instruction held in the instruction register during a first clock cycle of a two cycle instruction, means for gating the register address stored by the means for storing one of the register addresses to a read port normally used to receive a register address directly from a instruction placed in the instruction register during the second cycle of the two cycle instruction, and means for reissuing the instruction in the instruction register during there second cycle of the two cycle instruction in the read port normally used to receive a register address directly from an instruction placed in the instruction register is necessary to the instruction following the two cycle instruction.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a block diagram illustrating a typical central processing unit of a prior art SPARC based RISC computer.

Figure 2 is a diagram illustrating the positions of bits in a register + immediate format instruction held in a register in a typical SPARC based RISC computer.

Figure 3 is a block diagram illustrating a central processing unit of a SPARC based RISC computer constructed in accordance with the present invention.

NOTATION AND NOMENCLATURE

Some portions of the detailed descriptions which follow are presented in terms of algorithms and symbolic representations of operations on data bits within a computer memory. These algorithmic descriptions and 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, 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 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 associate 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 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 a block diagram of the central processing unit 10 of a typical SPARC based RISC computer of the prior art. The central processing unit 10 comprises an arithmetic and logic unit 12, a general register file 14 which may include at least thirty-two individual registers arranged in groups of eight global registers, eight local registers, eight IN registers, and eight OUT registers. The details of these registers are not pertinent to the explanation of this invention and are not, therefore, described hereafter. Also included are a number of control and status registers utilized for controlling the operation of the computer with which the central processing unit 10 is associated. Shown in Figure 1 is a single register 16 which may be termed an instruction register and which holds the instruction being executed.

The register 16 includes thirty-two bit positions which may be designated as positions [31...0]. The register 16 illustrated holds a standard form of instruction in the SPARC format ; such a format is termed a register + register (RR) format. Such an instruction includes in bit positions [4...0] the five bit address of a register which is to be read, in bit positions [18...14] the five bit address of another register which is to be read, in bit positions [24...19] the operation or command to be executed, and in bit positions [29...25] the five bit address of the register to be written to in a normal SPARC instruction execution. It should be noted that the five bits of each address are sufficient to designate thirty-two individual registers of the register file 14. Bit position [13] normally stores a zero to indicate that the format of the instruction is the RR format as described above.

In the normal execution of a SPARC RR format instruction, during a first clock cycle, the address held in bits [4..0] is transferred via five lines 18 and multiplexor 27 to the register file 14 causing the read out of data on the thirty-two lines 20 from the addressed register. During the same clock cycle, the address held in bits [18...14] is transferred via the five lines 22 to the register file 14 causing the read out of data on the thirty-two lines 24 from the addressed register. A multiplexor 29 operates in response to a zero in bit position 13 of the instruction to transfer the data on lines 24 to the arithmetic and logic unit 12. The data on the lines 20 and 24 is manipulated by the arithmetic and logic unit 12 in accordance with the instruction and the result placed in the register of the register file 14 designated by the bits [29...25] of the instruction on the five lines 26.

In the execution of a SPARC store instruction, on the other hand, during a first clock cycle, the address held in bits [4..0] is transferred via five lines 18 to the register file 14 via the multiplexor 27 causing the read out of data on the thirty-two lines 20 from the addressed register. During the same clock cycle, the address held in bits [18...14] is transferred via the five lines 22 to the register file 14 causing the read out of data on the thirty-two lines 24 from the addressed register. The data on the lines 20 and 24 is added by an address calculator arithmetic and logic unit 28 and placed in an address register 36 to provide the memory address at which information is to be stored in memory 30.

However, the information to be stored is held in a register of the register file 14 designated by the bits [29...25] of the instruction on the five lines 26. In the usual instruction, these bits designate a write port of the register file 14. In contrast to the usual instruction, the information in the register addressed by

these bits must be read rather than written. Therefore, the usual operation is to delay the execution by one cycle, and on the next cycle place the address in bits [29...25] at a read port of the register file 14 using the multiplexor 27 so that the information in the addressed register may be read on that second cycle from the read ports onto lines 24 for transfer to the memory 30. It should be noted that control of the multiplexor 27 to transfer the bits from positions [29...25] is exercised during a second store cycle.

Although it takes two clock cycles to accomplish the typical store operation, it is not necessary to increase the circuitry in order to provide another read port for the register file.

The present invention eliminates the system delay caused by a two cycle store operation in approximately three-fourths of the store operations without increasing the number of read ports. First, it is necessary to understand that the SPARC instruction format provides another form of instruction termed a register + immediate (RI) format instruction. This format is illustrated in Figure 2. The format is selected when the bit in position 13 of the instruction register 18 is a one. In this instruction format, the data contained in bits [12...0] is treated as a constant and manipulated with the data found in the register of the register file 14 addressed by bits [18...14]. This instruction is used for many if not most operations including loops in which a constant is added to or otherwise combined with a result to give a next result. Consequently, this form of instruction is much used in the operation of a SPARC processor. In fact, a RI format instruction has been found to occur as the next instruction following a store instruction in approximately three-fourths of all cases.

Since the RI format instruction only designates one register to be read from a read port of the register file 14 to form an address, one read port is left open during the execution of the instruction. This read port may be utilized by the store instruction in order to speed the operation of the central processing unit 10 in a SPARC based computer. This may be easily accomplished in a SPARC based computer because of its pipelining of instructions. That is, although two cycles are still necessary to accomplish the execution of the store instruction, the second cycle of the store instruction (during which the data contained in the address designated by the bits in positions [29...25] is read to access a position in the register file 14 containing data to be transferred to memory) is overlapped with the next instruction in all cases in which the next cycle is of the RI format. Consequently, the total time required to execute the store and the following RI format instruction is two cycles in the execution phase of the operation, the same number as for executing two non-store instructions.

Figure 3 illustrates in block diagram form an improved central processing unit 10 constructed in accordance with the invention. The central processing unit 10 includes all of the elements illustrated in Figure 1 and includes, in addition, a register 32 placed in the path from the lines 34 between the bit positions [29...25] and the multiplexor 27. During a first cycle of a store operation, the information in bit positions [18...14] and [4...0] of the instruction register 18 is transferred to the register file 14 on the lines 22 and 18, (the data on the lines 18 via the multiplexor 27) to control the access of the registers of the register file 14 for the address information that is to be read from the read ports onto lines 20 and 24. During the the same cycle, register 32 receives input from the bit positions [29...25] on lines 34 and stores that data. This

information is held in the register 32 as the next (non-store) instruction is placed into the register 18 on the next cycle clearing the register 18 of the store instruction.

On the second cycle of the store operation, the multiplexor 27 receives enabling input and the register address in bit positions [29...25] is transferred from the register 32 to the register file 14 via the multiplexor 27 and the read port normally connected to the lines 18. This information accesses the register of the register file 14 containing the information to be stored. If the instruction in the register 18 at this time is of the RI format, no address information is included in the bit positions [4...0] to be transferred to the read port. Consequently, the data in the register file 14 addressed by bits [29...25] is transferred on the lines 24 to memory 30 during the same cycle that the data in the register file 14 addressed by the bits in positions [18...14] is manipulated by the arithmetic and logic unit 12 with the constant provided in bit positions [12...0]. Since it has been determined that instructions of the RI format occur in the next cycle following a store instruction in a typical SPARC based computer three-quarters of the time, the arrangement of the invention will, in general, produce what averages in a pipelined system to a one cycle store instruction three-quarters of the time.

In those cases in which the instruction following the store instruction is not of the RI format, the store instruction executes in the same manner. However, the second cycle of the store instruction utilizes the read port controlled by the multiplexor 27 so that the information in the bit positions [4...0] of the new instruction cannot be transferred to the register file 14. Consequently, the instruction following the store instruction must be reissued in

such a case. This is accomplished by reading the bit 13 of the instruction in the instruction register 18 during the second cycle of the store operation; if the bit is a zero, this indicates that the instruction is an RR format instruction. When the bit 13 of the instruction following the store is a zero, the entire instruction is simply ignored, and the same instruction is placed in the register on the next cycle of operation. It should be noted that in such a case the store instruction executes in the same two cycle time which would be required by a more typical system so that the arrangement will not in any case be slower in executing a single store instruction than the prior art circuitry.

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 in terms of the claims which follow.



CLAIMS

1. A processor for a computer which is capable of handling pipelined instructions each containing register addresses and which includes an instruction register and a register file having a plurality of registers and a first number of read ports less than the possible number of addresses which may be included in a single instruction, the improvement comprising means for storing one of the registered addresses of an instruction held in the instruction register during a first clock cycle of a two cycle instruction, means for gating the register address stored by the means for storing one of the register addresses to a read port normally used to receive a register address directly from an instruction placed in the instruction register during the second cycle of the two cycle instruction, and means for reissuing the instruction in the instruction register during the second cycle of the two cycle instruction if the read port normally used to receive a register address directly from an instruction placed in the instruction register is necessary to the instruction following the two cycle instruction.

2. A processor as claimed in Claim 1 in which the means for gating at least two register addresses to a single read port comprises a multiplexer arranged to receive addresses from the instruction register and from the means for storing one of the register addresses of an instruction held in the instruction register during a first clock cycle of a two cycle instruction.

3. A processor as claimed in Claim 1 in which the means for reissuing the instruction register during the

* C L M S *

second cycle of the two cycle instruction if the read port normally used to receive a register address directly from an instruction placed in the instruction register is necessary to the instruction following the two cycle instruction comprises means for storing an indication with an instruction in the instruction register indicating that the read port is not used for that instruction, and means for testing whether an instruction contains such an indication.

4. A central processing unit (CPU) capable of executing a plurality of computer instructions in a pipeline fashion, said instructions being stored in a instruction register coupled to said CPU and represented therein by bit patterns including a first bit pattern and a second bit pattern, said first bit pattern requiring one CPU clock cycle to execute and said second bit pattern requiring two CPU clock cycles to execute, said bit patterns indicative of each of said instruction's operation code including register addresses and a control signal, said CPU comprising:

first register means, including a plurality of addressable registers having information signals stored therein, said first register means coupled to said instruction register and to a plurality of read ports and at least one write port, the number of said read ports coupled to said first register means is at least greater than or equal to the number of said addressable registers addressed in one clock cycle by one of said instructions stored in said instruction register, said first register means receiving said register addresses in said instruction register and transferring said information signals stored in each of said addressable registers addressed by said addresses to one of said read ports;

second register means coupled to said instruction register for storing one of said register addresses stored in said instruction register, said register address being stored in said second register means when said instruction register contains one of said instructions having said second bit pattern; and

control means having a first input coupled to said second register means, a second input coupled to said instruction register, and an output coupled to said first register means for controlling the input to said first register means;

said control means, in the event said CPU executes the first clock cycle of said instruction having said second bit pattern and said instruction having said second bit pattern is immediately followed thereafter by said instruction having said first bit pattern, transfer said address signals stored in said second register means to said first register means enabling said CPU to execute the second clock cycle of said instruction having said second bit pattern simultaneously with the first clock cycle of said instruction having said first bit pattern.

5. The central processing unit of Claim 4 wherein said register means includes a register file having one less read port than the maximum number of register addresses stored in said instruction register.

6. The central processing unit of claim 5 wherein said control signal stored in said instruction register has a first state and a second state, said first state indicating that said instruction register contains an instruction having said first bit pattern, said second state indicating that said instruction register contains an instruction having said second bit pattern.

* C L M S *

7. The central processing unit of claim 6 wherein said control means further consecutively tests said control signal stored in said instruction register for an instruction sequence requiring said CPU to execute said instruction having said second bit pattern and said instruction having said second bit pattern is immediatly followed thereafter by another said instruction having said second bit pattern, such that, if said instruction sequence occurs, said control means reissues the first instruction having said second bit pattern in said sequence into said instruction register after the CPU executes the first clock cycle of said first instruction having said second bit pattern, thereby delaying by one clock cycle the execution of the first clock cycle of the second instruction having said second bit pattern in said instruction sequence.

8. The central processing unit of Claim 7 wherein said control means includes a multiplexor.

9. A central processing unit (CPU) capable of executing a plurality of computer instructions in a pipeline fashion, said instructions being stored in an instruction register coupled to said CPU and represented therein by bit pattern including a first bit pattern and a second bit pattern, said first bit pattern requiring one CPU clock cycle to execute and said second bit pattern requiring two CPU clock cycles to execute, said bit patterns indicative of each of said instruction's operation code including register addresses and a control signal, said CPU comprising:

a file register, having a plurality of addressable registers, said file register being coupled to said instruction register and to a first read port, a

second read port and a first write port, said file register used for storing information signals in said plurality of addressable registers;

an address storage register coupled to said instruction register for storing one of said register addresses stored in said instruction register, said register address being stored in said address storage register when said instruction register contains one of said instruction having said second bit pattern;

a control circuit having a first and second input, said first input coupled to said address storage register, and said second input coupled to said instruction register, said control circuit further including output coupled to said file register for selecting which of said first and second inputs are coupled to said file register; and,

said control circuit, in the event said CPU executes the first clock cycle of said instruction having said second bit pattern and said instruction having said second bit pattern is immediately followed thereafter by said instruction having said first bit pattern, transfers said address signals stored in said address storage register to said file register thereby enabling said CPU to execute the second clock cycle of said instruction having said bit pattern simultaneously with the first clock cycle of said instruction having said first bit pattern.

10. The central processing unit of Claim 9 wherein said register addresses stored in said instruction register address, a second register address, and a third register address, said first register address being directly coupled to said file register, said second register address being directly coupled to said control circuit, and said third register address being directly

coupled to both said file register and said address storage register such that the information signals addressed by said first address are read out of said file register through said first read port, and the information signals addressed by said second and third address are read out of said file register through said second read port.

11. The central processing unit of Claim 10 wherein said control signal has two states, a first state and a second state, said first state indicative of an instruction having said first bit pattern and said second state indicative of an instruction having said second bit pattern.

12. The central processing unit of Claim 11 wherein said control circuit further tests the state of said control signal stored in said instruction register such that if said control signal is in said first state said control circuit transfers said address stored in said address storage register to said register file, and if said control signal is in said second state said control circuit transfers said second address to said register file.

13. The central processing unit of Claim 12 wherein said control circuit further consecutively tests said control signal stored in said instruction register for an instruction sequence requiring said CPU to execute said instruction having said second bit pattern and said instruction having said second bit pattern is immediately followed thereafter by another said instruction having said second bit pattern, such that, if said instruction sequence occurs, said control circuit reissues the first instruction having said

second bit pattern in said sequence into said instruction register after the CPU executes the first clock cycle of said first instruction having said second bit pattern, thereby delaying by one clock cycle the execution of the first clock cycle of the second instruction having said second bit pattern in said instruction sequence.

14. The central processing unit of Claim 16 wherein said control circuit includes a multiplexor.

15. A processor substantially as hereinbefore described with reference to Fig. 3 of the accompanying drawings.