(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
16 August 2007 (16.08.2007)

PCT

(10) International Publication Number
WO 2007/090734 A1

(51) International Patent Classification:
G06F 1/08 (2006.01) *G06F 1/32* (2006.01)

(21) International Application Number:
PCT/EP2007/050634

(22) International Filing Date: 23 January 2007 (23.01.2007)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
11/348,551 7 February 2006 (07.02.2006) US

(71) Applicant *(for all designated States except US)*: **OMX TECHNOLOGY AB** [SE/SE]; Tullvaktsvägen 15, S-105 78 Stockholm (SE).

(72) Inventor: **WINBOM, Håkan**; Stråkvägen 54, S-191 42 Sollentuna (SE).

(74) Agent: **NORDLUNG, Tore**; OMX Technology AB, Patent Department, Tullvaktsvägen 15, 105 78 Stockholm (SE).

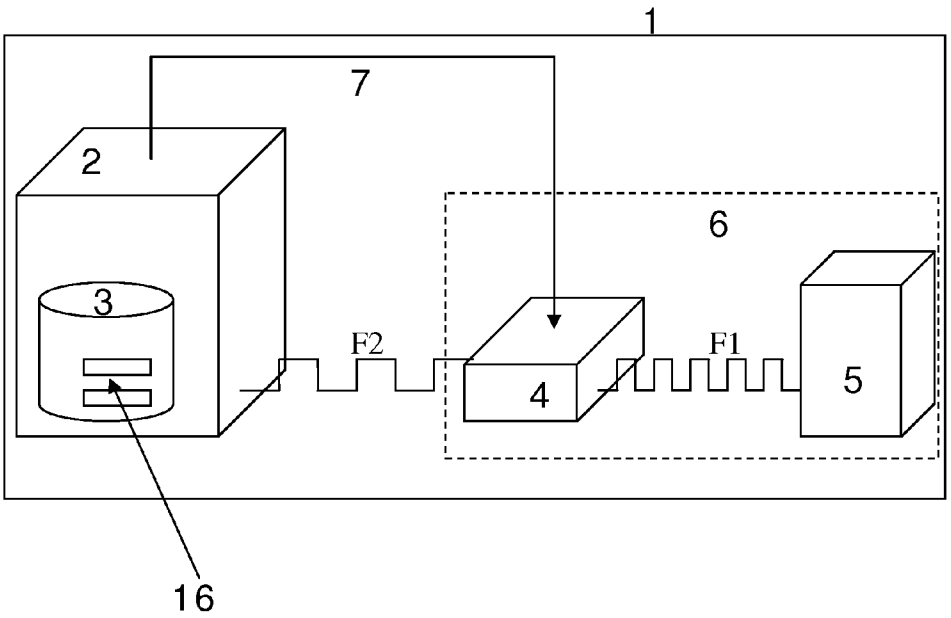
(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: APPLICATION SOFTWARE INITIATED SPEEDUP



(57) Abstract: The invention discloses an instruction, processor, system and method which allow application level software to explicitly request a temporary performance boost, from computing hardware. More specifically it relates to advanced management of working frequency of a processor in order to achieve the performance boost. Preferably a processor according to the invention may be implemented in electronic exchanges or similar applications where peak periods may occur.

WO 2007/090734 A1

APPLICATION SOFTWARE INITIATED SPEEDUP

FIELD OF THE INVENTION

The present invention relates generally to management of processor working frequency in computer systems and electronic devices.

BACKGROUND OF THE INVENTION

Today there exist many different kinds of computer systems. The systems range from individual computers to distributed computer systems interconnected by one or more networks.

One of the more important problems for the above described systems and computers is that they at one point or another will experience peak periods, which they may have a hard time to handle. Usually the systems are designed to handle peak periods up to a certain level. This level is usually based on a prediction of a future traffic scenario in the system weighted against the hardware cost. Thus, when trying to estimate the capacity for a system there is a risk for underestimating the future traffic in the system which may result in a shut down of the system. An example of an incident where the capacity of the system was not enough happened at the Tokyo Stock Exchange (TSE) on the 18th of January 2006. According to the article "Temporary Suspension of All Transactions" (TSE news), the stock exchange had to close in order to prevent a system crash.

Thus, there may be situations when the capacity of the system is not enough. Hence the systems are not able to, or have a hard time to handle for example all the requests sent by the users using the systems, or a hard time to send out a large amount of information. As a result of this, users may experience very long waiting times for their request to be processed and may become very frustrated because of this, or even worse the system may have to be shut down in order to avoid a crash being out of control.

This situation may especially occur in financial trading systems (electronic exchanges). These systems receive many quotes/orders from traders and/or brokers, trading in financial instruments, energy contracts and so forth. Examples of when peak periods can occur are during the auction period before opening of an exchange, or when a task is sent that may result in a processor demanding change in an orderbook, or information dissemination, or because something else happened such as a political statement or rumor in the news or any happening that can trigger an increase of quotes/orders sent from traders.

Other applications having heavy transactions and peak periods are bank systems, warehouse systems and Internet broker services.

One known way to achieve increased performance of a computer is to hard-wire a processor so that it statically runs on a clock-speed higher than normal. (this is commonly referred to as "over-clocking"). This typically leads to heat problems since the computer system is operated continuously at this higher speed and is therefore not suitable for commercial systems that need to have a certain level of security. Hobbyists are often statically "over-clocking" their systems in order to increase the performance for gaming and so forth. Another problem with this approach is that it is not possible to lower the frequency in an easy way since it is done manually by hard-wiring.

Some computer systems are using "variable" (clock) speed in order to preserve energy, reduce heat and related noise. One such technology is Intel's "speed step". However this technology has been applied to battery powered "lap top" computers and office space "desktop" computers for saving battery power and for noise reduction. In these examples it is the operating system and/or the "BIOS" which detects some level of inactivity by the user, and "slows" down the clock by a supervisor state instruction.

Supervisor state instructions are instructions only executable in "supervisor state" or some similar mode which is reserved for the operating system.

Hence, over-clocking is known and most computer chips can operate at a higher rate than specified by the manufacturer. However the above described techniques increases the risk for failures, mainly because of heat related problems, but also because it is necessary to make physical changes to the mother-board of a computer in order to statically over-clock a processor.

Thus, it is known to statically increase the speed of a working frequency for processors of today, or for example when a processor has slowed down the clock due to inactivity and when it experience activity, the OS speeds up the working frequency. However it only speeds up the frequency to a maximum rate specified by the manufacturer and only at times specified by manufacturers, and only after it experienced an increase in workload. This is a very inefficient way of managing the working frequency of a processor since it takes a while for the processor to realize that it is under a heavy workload. The time for the processor to realize this may be rather long from a processor point of view.

Thus processors are managed very inefficient which results in low capacity usage compared to what processors actually could deliver.

BRIEF DESCRIPTION OF THE INVENTION

Thus it is an object of the present invention to provide at least a partial solution to some of the problems mentioned above.

It is further an object of the present invention to provide a solution that increases the capacity usage of a processor.

It is further an object of the present invention to provide a solution that brings about a performance increase of underlying hardware.

It is further an object of the present invention to provide a solution so that application level software can request a performance increase from the underlying hardware.

It is further an object of the present invention to provide a solution that changes the working frequency of a processor.

It is further an object of the present invention to provide a solution that saves time for a user.

It is further an object of the present invention to provide a solution that cuts the time for processor demanding executions.

It is further an object of the present invention to provide a solution increasing the working frequency of a processor just before a processor demanding request arrives.

It is further an object of the present invention to provide a solution that decrease the response time.

According to a first aspect of the invention, the above and other objects are fulfilled by a processor comprising:

- a frequency generating unit for generation of a variable working frequency,
- a memory comprising an instruction for changing the working frequency,
- an interface for receiving a user instruction,

wherein the instruction for changing the working frequency is activatable via the interface by the user instruction sent from a user level application.

Thus the processor being programmed to:

- receive a user instruction via the interface from a user level application for changing the working frequency, and

- based on the user instruction, change the working frequency by using the instruction stored in the memory.

The above processor has the advantage that a user level application may directly request a performance increase, such as increasing the work frequency of the processor.
Usually an operating system when initiating a hardware operation has to progress through: user mode to OS mode transition, the OS executes the hardware instruction and then transition back to user mode (which may involve a scheduling decision, and possible dispatch of an other process). These steps may be referred to as overhead cost such as cost in time and cost in processor power and so forth. The above described transition process slows down a computer or a computer system. Therefore a processor according to the invention may be able to cut overhead and in this way make use of processor capacity currently not utilized and thereby deliver results faster compared to other processors.

For most computer systems a user level application may be the first line of early warning systems for e.g. a change in workload, since it is in the user application where users requests different operations/functionalities to be executed. A user level application may detect the need for speed-up in several different ways, for example when known processor demanding operations/functions are requested.

In the case the user level application is a server application the need for speed-up may be detected by for example an increase of the input queue.

Thus if a processor demanding operation/functionality is requested by a user, the user level application can send a speed-up user instruction to the processor and in this way prepare the processor for an upcoming increase in workload, by increasing the working frequency.

Preferably the user instruction is triggered and sent in the user application based on a user action. A user action may be any kind of action that a user may perform, either directly via an interface or indirectly based on the user action. For example a user action can be an action such as: trade action, delete action, amend action, screen update action, direct request of speed up action, this is not an exhaustive list, other actions not mentioned here may also be added in the list.

In an embodiment the user instruction may be triggered and sent by a user action belonging to a list, e.g. a “top-ten” list comprising the ten most processor demanding actions. Thus when a user executes an action, the system checks whether this action belongs to the list and if it does the system sends a speed-up user instruction. This list may be continuously updated and may be based on statistics from the system. Thus in a

further embodiment, the invention may comprise a self learning function that updates the "top-ten" list. The "top-ten" list may comprise an arbitrary number of processor demanding actions and is thus not limited to ten. The length of the list may thus be decided by the context wherein the present invention will be used. Preferably this function would be application software controlled.

A user action can also be an action that is automatically executed by a server application. Thus when a server receives a task that has been sent from a user level application by a user, the server application on the server may request a change of the working frequency of the processor such as a speed-up request.

Thus a user level application can also be a server application. However in this case the server based user level application preferably has a software function that receives data from client computers and thereafter analyses the data in order to see if it comprises processor demanding tasks or instructions, and if it does, the server based user level application may request a speed-up of the working frequency of the processor that will handle the workload.

The processor preferably comprises a temperature sensor for measuring a temperature of the processor in order to optimize the working frequency based on the temperature and based on the user instruction. In this way it is possible to control the working frequency so that the processor performs at an optimal speed in relation to the temperature.

The processor may further comprise a counter for counting the number of instructions that has been executed in speed-up mode, and when the number of instructions executed in speed-up mode, reaches a predetermined threshold, the speed-up mode is "canceled" or lowered. In this way the temperature problem can be avoided since the threshold would preferably be set to a number that would not overheat the processor. Thus, this may be an alternative and/or complementary to heat sensing.

Hence, it would be possible to optimize the working frequency in such a way that if the user instruction would normally trigger a speed-up of the working frequency to e.g. 95% of its maximum potential, due to the temperature the increase would only be up to 90% of its maximum potential. This may be based on a function, for example a linear function or exponential function and so forth. Thus, instead of certain thresholds, a continuous control of the speed-up would be possible to achieve in this way.

The frequency generating unit may comprise two or more frequency generating devices producing different frequencies in order to generate the variable working frequency. In this

embodiment the frequency generating unit preferably comprises a switch for selecting one of the different frequencies in order to generate the variable working frequency.

The frequency generating device may be a crystal that vibrates at a regular frequency when an electrical current is applied to it and in this way produces a clock frequency. However any other device able to produce a clock frequency that can be used in a processor may be used.

Furthermore the frequency generating unit may comprise a frequency generating device generating a frequency and a divisor for changing the frequency to the variable working frequency. Preferably the instruction according to the invention changes the working frequency by changing a divisor factor.

The divisor factor may be an integer or a floating point number by which the divisor can be controlled in order to obtain a variable working frequency.

The frequency generating unit may comprise a frequency generating device generating a frequency and control means for direct control of the frequency generating device in order to output a variable working frequency.

For example the control means could be a device generating a variable pressure to the crystal, or a device generating a temperature change of the crystal, and in this way the frequency of the crystal could be changed.

The processor may be implemented in an electronic device such as computers, servers or hand held and portable devices such as mobile phones, palm pilots, cameras or in any other electronic device where it may be necessary to obtain a performance increase during peak periods or for processor demanding instructions.

Such an electronic device has the advantage that it may request a performance increase of the underlying hardware in order to execute a processor demanding task. Thus it may deliver a result faster to a user since it is able to at a very early stage identify a processor demanding task that the user has requested.

In a preferred implementation the processor is used in an electronic exchange systems, examples of such systems are OMX systems CLICK™, SAXESS™ or in clearing systems such as SECUR™.

Thus an electronic exchange using a processor according to the invention has the advantage that it may request a performance increase of the underlying hardware if/when

it receives a processor demanding request. An example of such a processor demanding request may be a delete all order request for a trading firm or any other processor demanding request that may occur in an electronic exchange environment.

In a second aspect of the invention the above objects are achieved by providing an instruction controlling working frequency in a processor, the instruction is stored in a memory of the processor, and is characterized in that it is activatable by a user level application, for changing the working frequency.

Thus the above instruction has the advantage that it provides a solution that directly dramatically cuts overhead such as superfluous steps which causes additional cost such as cost in time and so forth. Thus the instruction makes it possible to, at an early stage, request a hardware performance change. In this way capacity currently not being used can be accessed.

Since the instruction doesn't have to be activated by the operating system when controlling the clock frequency it is possible to achieve for example an earlier speed-up of the processor.

Preferably the change of the working frequency is a temporary increase of the working frequency in order to handle a future increase in work load. However it may also be a decrease of the working frequency.

The frequency may be controlled in a way so as to optimize the usage of a processor. For example the speed of the processor may thus be decided based on a request sent from a user level application. However an additional functionality may optimize the usage of the processor even further based on heat generated from the processor. Hence, means for measuring the temperature may be necessary to use. Thus in order to avoid overheating of the processor the frequency can be decreased for a while and in this way keep the heat at an allowable level but at the same time running the processor at a maximum speed.

In a third aspect of the invention, the above and other objects are fulfilled by a user level application for trading, the user level application comprising program instructions for:

- based on a user action, requesting a change of a working frequency of a processor in order to change performance of the processor.

The above user level application is preferably a trading application such as CLICK Trade™ for trading of commodities and/or financial instruments. Thus a software module installed in the front-end of a computer system, such as in a client computer.

However the user level application may also be any software module in an electronic exchange, such as matching module software or any other server based application. Thus a software module installed in the back-end of a computer system.

Some of the user actions that occur when a user trades by using a user level application, can result in a very processor demanding task being sent to the electronic exchange. When such a task is selected by a user, the user level application for trading may select a function comprising a speed-up request and send it together with the task, and in this way ask for a performance increase in order to execute the task faster.

In a fourth aspect of the invention, the above and other objects are fulfilled by a method for increasing performance of a processor based on a prediction of a future change in workload, the processor comprising an instruction stored in a memory, the method comprising the steps of:

- from a user level application, sending a user instruction for changing a working frequency of the processor, and
- based on the user instruction, changing the working frequency by using the instruction stored in the memory.

The above method has the advantage of requesting a performance increase of underlying hardware in order to achieve a faster execution of a processor demanding request.

Thus allowing the application to execute this particular instruction directly dramatically cuts the overhead. Usually user mode applications are prevented from direct hardware access for integrity reasons, however in this particular case the instruction would not change the result of a calculation, it would just deliver the result faster by changing the working frequency of a processor, thus it is safe to allow the application program to invoke this instruction directly without Operating System intervention.

The user level application may be a computer program product loaded on to a computer-readable data carrier or a computer program product downloadable via a network such as the Internet.

The present invention can be used to design system which allows higher absolute peak loads, beyond the current maximum achieved. The invention may also improve cost efficiency by not having to design computer systems quite as large, because the system will be able to handle short term peak loads which are significantly higher than the sustained load capacity.

It also allows the computing system to maintain reasonable response times even for costly transactions.

These and other aspects of the invention will be apparent from and elucidated with reference to the embodiments described hereinafter.

BRIEF DESCRIPTION OF FIGURES

Fig. 1 illustrates an overview of the system comprising a CPU, a divisor and a frequency generating unit, wherein the CPU has a memory comprising at least one instruction.
Fig. 2 illustrates a central computer system comprising the electronic exchange and client computers comprising e.g. trading applications and or clearing applications.
Fig. 3 illustrates a client computer comprising a user level application and its user interface whereby a user may execute an instruction.
Fig. 4 illustrates a first embodiment of a frequency generating unit connected to a processor.
Fig. 5 illustrates a second embodiment of a frequency generating unit.
Fig. 6 illustrates an embodiment of the system comprising a temperature sensor.

Figures are preferably schematically drafted in order to facilitate the understanding of the invention. Therefore other designs that could be drafted in the same schematic way are implicitly also disclosed in this document.

DESCRIPTION OF PREFERRED EMBODIMENTS

The present invention proposes two new hardware instructions preferably accessible directly by application level software (SW), as opposed to access by BIOS/ Operating system (OS) and device drivers.

A user mode instruction is an instruction which an application process may execute directly without assistance of the operating system. Usually hardware related instructions are only executable in "supervisor state" or some similar mode which is reserved for the operating system. Thus for an application to initiate a hardware operation one usually has to progress through:

- User mode to OS mode transition,
- OS executes the hardware instruction, and
- transition back to user mode (which may involve a scheduling decision, and possible dispatch of an other process)

The first instruction according to the invention may be Speed_Up(x%), this instruction would preferably request the hardware to speed up as x% of the maximum possible speed-up for as long time as possible.

The maximum time would typically be determined by increased temperature of the CPU or surrounding chips, e.g. once a preset maximum temperature is reached, the speed-up is "canceled" or lowered. The maximum time could also be determined by a counter that counts the number of instructions that has been executed in speed-up mode, and when the number of instructions executed in speed-up mode, reaches a predetermined threshold, the speed-up mode is "canceled" or lowered.

The second instruction according to the invention would preferably be a reset instruction, for example Speed_Up_release(), as means to allow the application to voluntarily reset to normal mode of execution.

The above two instructions are preferably user mode instructions.

In a multitasking multi CPU (Symmetric Multi-Processing - SMP) environment it is still envisioned that any Application level process or thread should be allowed to request Speed_up(), normally it should not relinquish control of the CPU by issuing OS and/or I/O call before it calls Speed_Up_Release(). However it could possibly be preempted by the operating system before it is finished, in these cases the speed up state would apply to the OS and/or other thread dispatched on this CPU until a heat limit, time limit or other limit is reached.

The objective with the % argument to Speed_Up() is to allow the application to voluntarily request e.g. just 50% of the possible speed up, with the assumption that it may be attained approximately twice as long, and/or that the next request will be able to get a speed up.

EXAMPLE

The following set of example comes from the domain of "electronic" securities trading, the invention is by no means limited to this domain, it is just an illustrative set of example of a good utilization of the invention.

In order to facilitate the understanding of the example, some terms will first be defined.

The following is a list of terms, and how they are to be perceived in this description:

- Order: an instruction to make a transaction, i.e. to buy or sell a certain amount of a specific instrument at a given price. Orders are usually received from "investors", i.e. private investors or stock brokers.

- Market Maker: An entity usually contracted by the operator of the trading system, for example a stock exchange, to maintain both buy and sell prices (and volumes) in the trading system for certain instruments. There may be some restrictions to this, e.g. that buy and sell prices should not differ more than a certain percentage.

- Quote: an instruction to a trading system or an exchange containing both buy and sell "bids" for one or more instrument. Usually issued by market makers, and due to their nature (bids), transient in time.

- Deal: a match between orders and/or quotes, comprising at least two trades (one buy, one sell) which make up the legs of the deal.

- Order book: a table or a list in a trading system or an exchange, comprising all buy and sell orders as well as all quotes. There is essentially one order book for each tradable object.

- Auction: orders are allowed to be stored in the order book during some time interval, but are not matched until some preset time or event occurs.

- External events: e.g. if the central bank adjusts the interest rate, this is an external event which will cause almost all active traders to re-evaluate their positions in the market. Market makers will most likely delete all their quotes and soon submit new, while "normal" traders may cancel their orders and submit new orders with different prices.

- Wild card delete: this is a request by participating firm to delete all their orders in all order books. This is a very costly transaction because it directly implies searching all order books for order by this user/firm and removes them. It also implies that the market information system will have to become active and "broadcast" the new state of the market.

- User mode instruction: A user mode instruction is an instruction which an application process may execute directly without assistance of the operating system. Usually hardware related instructions are only executable in "supervisor state" or some similar mode which is reserved for the operating system. Thus, a user mode instruction is an

instruction which a user level application may execute directly without assistance of the operating system.

Usually most hardware related instructions are only executable in "supervisor state" or some similar mode which is reserved for the operating system. Thus for an application to initiate a hardware operation one usually has to progress through

- User mode to OS mode transition
- OS executes the hardware instruction
- Transition back to user mode (which may involve a scheduling decision, and possible dispatch of an other process)

(The above steps may be referred to as overhead in this application.)

Thus allowing the application to execute this particular instruction directly dramatically cuts the overhead.

Usually user mode applications are prevented from direct hardware access for integrity reasons, however in this particular case the instruction would not change the result of the calculation, it would just deliver the result faster, thus it is safe to allow the application program to invoke this instruction directly without operating system intervention.

Example 1.

In this scenario envision that an external event causes e.g. a market maker to requests a delete of all his quotes in all orderbooks. The software maintaining the order books, just calls the Speed_Up(100) function, iterates through the order books and once it is done it would call Speed_Up_Release(), this operation may possible finish in 20% less absolute time, which would be a very significant improvement.

If the Speed_Up() state was canceled before the wild card delete was finished, e.g. due to thermal reasons, it would still be a beneficial improvement.

Market maker firms using automated quote engines are very sensitive to increased response times; in fact they may treat increased response time as an external event which causes them to issue “wildcard delete” transactions. Indeed if Market Maker firm 1 issues wild card delete, and this causes response times to deteriorate, this may cause other market maker firms to also issue “wildcard delete” transaction, aggravating the issue.

Example 2.

Online auction for an instrument, in this case it is a highly CPU bound job to scan the order book for matching orders and quotes, thus it is ideal for use of Speed_Up(100) followed by Speed_Up_release() once done.

Example 3.

High trade rate.

In some exchanges the order books may be filled with large amounts of "retail" orders (each with low volume), suddenly a large participant e.g. a bank decides to buy/sell a huge volume of a particular instrument, thus one deal can generate thousands of trades, this causes a burst of CPU bound work to finish this processing, this is also ideally suited to Speed_Up()/Speed_Up_Reset().

Figure 1, illustrates an overview of an embodiment of the system according to the invention. The system comprises a CPU 2, a memory 3 comprising instructions 16, a frequency generating unit 6 comprising a divisor 4 and a frequency generator 5. Furthermore the figure illustrates a first frequency F1 and a second frequency F2 and a control connection 7 for sending control commands from the CPU to the Divisor so as to change the working frequency F2.

An instruction 16 preferably comprises a command and a value. The command may be of the type Speed_Up or Speed_Up_release as described above and the value may be a percentage of the maximum possible speed, also described earlier in this document.

Figure 2, illustrates a distributed computer system comprising client computers 8 and a central computer 9. The client computers 8 and the central computer 9 may be interconnected by any kind of network such as the Internet, WAN, LAN, or direct connections such as P2P. Furthermore also wireless communication may be used, for example when the client computer 8 is a hand held electronic device such as a Palmpilot, mobile phone or a portable computer. The figure illustrates a star network but any other type of network may be used, for example ring network or bus network and so forth. Furthermore the communication may be routed over nodes not shown in figure 2. Secure channels may be necessary to establish between the client computers 8 and the central computer 9.

The central computer 9 may be a computer comprising different types of server applications. As an example in the present invention the central computer 9 comprises an electronic exchange comprising a common database that further comprises market, product, user and transaction information necessary to operate an electronic exchange. Furthermore the central computer 9 may comprise other subsystems that are necessary

for an electronic exchange. Some example of subsystems are an information dissemination system that process and broadcast information about trades and deals, a market place subsystem that manage the exchange's central order book, and so forth. An example of an electronic exchange is the CLICK™ system manufactured by OMX Technology.

Figure 3, illustrates a client computer 8 comprising a user interface preferably having two parts, one monitor 10 and one input device 11. However the user interface may be a touch screen and in this case the two parts are integrated in one embodiment.

The monitor 10 may be any type of device that is able to present information to a user, such as an LCD display, plasma screen and so forth. The input device 11 may be any type of input device such as a keyboard, mouse, joystick or a keyboard specially designed for e.g. trading. As described above the client computer may be any kind of electronic device such as a mobile phone, Palmpilot and so forth that is able to communicate with other electronic devices over a network or by wireless communication.

Figure 4, illustrates an embodiment of a frequency generating unit 6 connected to a processor 12. Preferably the frequency generating unit 6 comprises a frequency generator 5 and a divisor 4 for changing the frequency generated by the generator 5. The generator 5 generates a first frequency F1 which is changed in to a working frequency F2 that is sent to the processor.

Figure 5, illustrates a second embodiment of a frequency generating unit 6 connected to a processor 12. In this embodiment the generating unit 6 preferably comprises two frequency generators 5 and a Switch device 13. Each frequency generator preferably produce a frequency having a different frequency from the other frequency generator, a first and a second frequency F1, F2. The switch 13 may switch between the most suitable frequency based on an instruction from a user level application and in this way a third frequency F3 (working frequency) may be sent to the processor 12. The switch 13 is preferably controlled by instructions sent from the processor 12.

The number of frequency generators is not limited to two; an arbitrary number of generators may be used. In a further embodiment a divisor may also be connected to at least one of the lines which would provide the possibility of changing either the F1 or F2 or F3 or all of them depending on where the devisor(s) is placed in the system.

Figure 6, illustrates a third embodiment of the system according to the invention wherein a temperature sensor 14 is connected to the processor 12 or in the vicinity of the processor such as to the CPU or motherboard. The sensor 14 may be used in order to watch the temperature of the processor 12 so that the temperature stays within allowable limits.

Preferably the temperature may be controlled by increasing or decreasing the working frequency of the processor 12. However the temperature may also be controlled by increasing an air flow if the processor is chilled by air, or by increasing a flow of a liquid if the processor is chilled by a liquid.

In a further embodiment the system may comprise a counter 15 for counting the number of instructions that is executed by the processor 12 as described earlier in the text.

In the above description the term "comprising" does not exclude other elements or steps and "a" or "an" does not exclude a plurality.

Furthermore the terms "include" and "contain" does not exclude other elements or steps.

CLAIMS

1. A processor comprising:
- a frequency generating unit for generation of a variable working frequency,
- a memory comprising an instruction for changing the working frequency,
- an interface for receiving a user instruction,

wherein the instruction for changing the working frequency is activatable via the interface by the user instruction sent from a user level application.

2. A processor according to claim 1, further comprising a temperature sensor for measuring a temperature of the processor in order to optimize the working frequency.

3. A processor according to claim 1, wherein the frequency generating unit comprises two or more frequency generating devices producing different frequencies in order to generate the variable working frequency.

4. A processor according to claim 3, wherein the frequency generating unit further comprises a switch for selecting one of the different frequencies in order to generate the variable working frequency.

5. A processor according to claim 1, wherein the frequency generating unit comprises a frequency generating device generating a frequency and a divisor for changing the frequency in to the variable working frequency.

6. A processor according to claim 1, wherein the frequency generating unit comprises a frequency generating device generating a frequency and control means for direct control of the frequency generating device in order to output a variable working frequency.

7. An electronic device comprising a processor according to claim 1.

8. An electronic exchange comprising a processor according to claim 1.

9. An instruction controlling working frequency in a processor, the instruction being stored in a memory of the processor, characterized in that the instruction is activatable by a user level application, so as to change the working frequency according to the instruction.

10. An instruction according to claim 9, wherein the change of the working frequency is a temporary increase of the working frequency in order to handle a future increase in work load.

11. An instruction according to claim 9, wherein the instruction comprises a command part and a value part comprising data for changing the working frequency in a processor.

12. A user level application for trading, the user level application comprising program instructions for:

- based on a user action, requesting a change of a working frequency of a processor in order to change performance of the processor.

13. A user level application according to claim 12, wherein the user level application is chosen from a group of applications comprising:

- a server application, and
- a user application.

14. A user level application for trading according to claim 12, wherein the user level application further being programmed to:

- based on a prediction of a future increase of workload on the processor, request an increase of the working frequency of the processor.

15. A method for increasing performance of a processor based on a prediction of a future change in workload, the processor comprising an instruction stored in a memory, the method comprising the steps of:

- from a user level application, sending a user instruction for changing a working frequency of the processor, and
- based on the user instruction, changing the working frequency by using the instruction stored in the memory.

16. A computer-readable data carrier loaded with a computer program product according to claim 12.

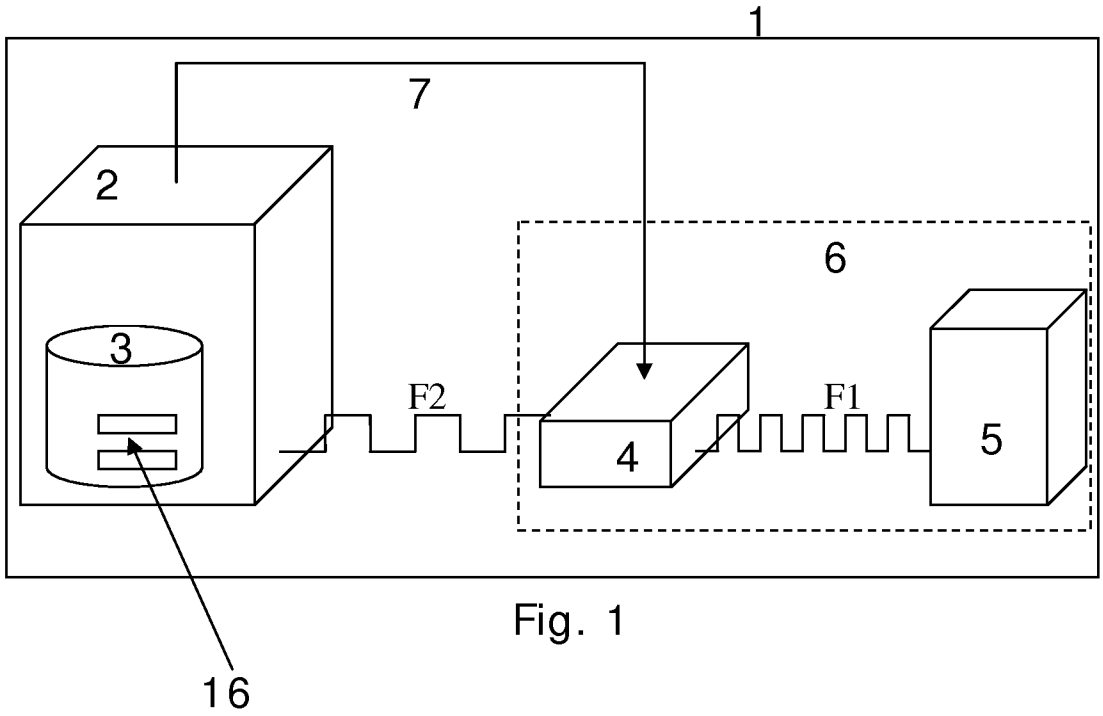


Fig. 1

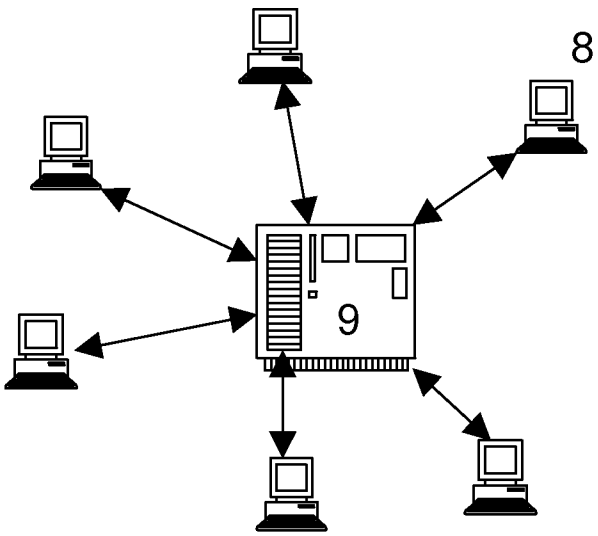


Fig. 2

Fig. 3

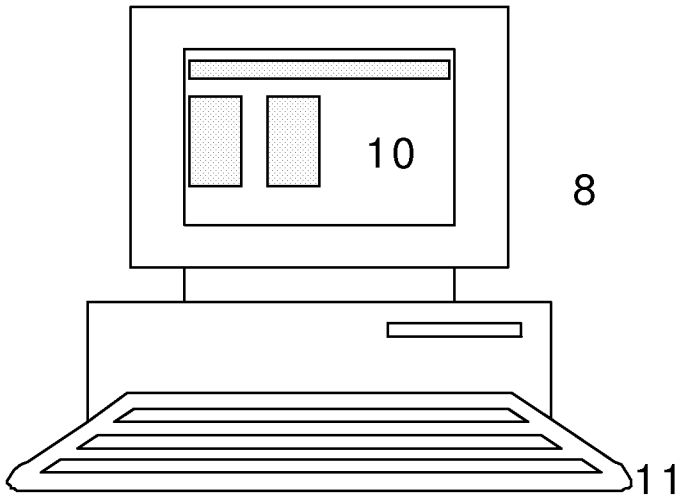


Fig. 4

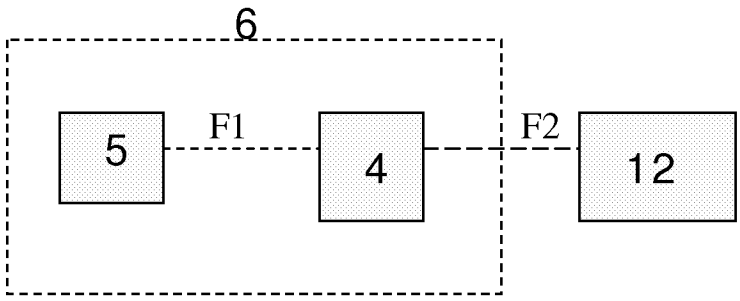


Fig. 5

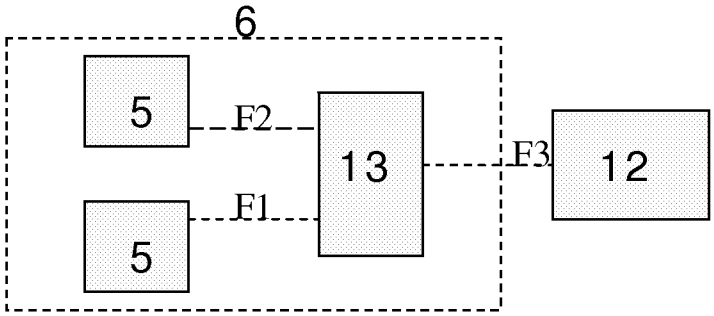
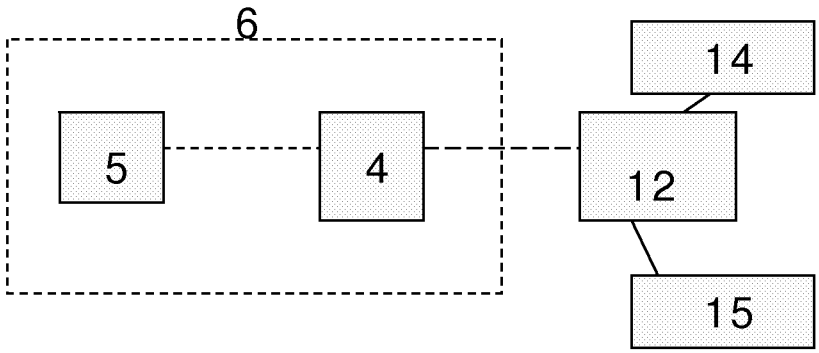


Fig. 6



INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2007/050634

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F1/08 G06F1/32

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)
G06F

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 practical, search terms used)
EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5 630 148 A (NORRIS DAVID [US]) 13 May 1997 (1997-05-13) the whole document	1,3-16
X	US 2002/124197 A1 (ATKINSON LEE W [US]) 5 September 2002 (2002-09-05) abstract paragraphs [0004], [0016] - [0021] paragraphs [0026] - [0029], [0036], [0042], [0043], [0052] - [0056]; figure 1 claims 37,40	1,3-16

-/--

[X] Further documents are listed in the continuation of Box C.

[X] See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
9 May 2007	23/05/2007

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31–70) 340–2040, Tx. 31 651 epo nl, Fax: (+31–70) 340–3016	CHARCOS LLORENS, V

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2007/050634

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2003/061523 A1 (STANLEY RANDY P [US]) 27 March 2003 (2003-03-27) paragraphs [0016] - [0023], [0025], [0029] - [0032]; figures 1,2 paragraphs [0039] - [0056]; figures 3,4 paragraph [0059]	1-16
X	US 2003/097603 A1 (ATHAS WILLIAM C [US]) 22 May 2003 (2003-05-22) paragraphs [0010] - [0018], [0026] - [0034], [0037] - [0041]; figures 1,3	1-16
X	US 6 230 279 B1 (DEWA KOICHI [JP] ET AL) 8 May 2001 (2001-05-08) abstract column 2, line 45 - column 3, line 33 column 5, line 20 - column 8, line 67; figures 1-6 column 10, line 54 - column 11, line 23; figures 11-14	1-16

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/EP2007/050634

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5630148 A	13-05-1997	AU 2903295 A	15-01-1996
		BR 9508040 A	18-11-1997
		CN 1150845 A	28-05-1997
		DE 69524873 D1	07-02-2002
		DE 69524873 T2	05-09-2002
		EP 0765499 A1	02-04-1997
		JP 10501911 T	17-02-1998
		JP 2006302306 A	02-11-2006
		WO 9535540 A1	28-12-1995
US 2002124197 A1	05-09-2002	NONE	
US 2003061523 A1	27-03-2003	CN 1610872 A	27-04-2005
		EP 1499943 A2	26-01-2005
		TW 236585 B	21-07-2005
		WO 03027818 A2	03-04-2003
US 2003097603 A1	22-05-2003	US 2004210787 A1	21-10-2004
		US 2005060589 A1	17-03-2005
US 6230279 B1	08-05-2001	NONE	