

## CERTIFICATE OF TRANSLATION ACCURACY

January 6, 2026

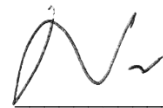
I, the undersigned, Steve Ni, hereby certify:

I am a professional translator. I work as an independent contractor. I was contracted by Divergent Language Solutions, LLC to provide document translation from Japanese into English for this matter.

I am fluent in both Japanese and English. My qualifications include 20 years of full-time professional translation experience from Japanese into English.

I am familiar and competent with both the Japanese and English languages. I have received and translated from Japanese into English the Japanese Patent No. JPB 003179054, and hereby attest that the English translation is a complete and accurate rendition of the Japanese original to the best of my knowledge and belief.

I declare under penalty of perjury under the laws of the United States of America that the foregoing is true and correct. Executed on January 6, 2026.



Steve Ni



Divergent Language Solutions, LLC

600 California Street, 11<sup>th</sup> Floor | San Francisco, CA 94108 | p 415.400.4538 | f 415.525.4313

[divergent@divergentls.com](mailto:divergent@divergentls.com) | [www.divergentls.com](http://www.divergentls.com) | IPR2025-01309; Samsung

Ex. 1028 - Page 1

(19) Japanese Patent Office (JP)

(12) Published Patent Gazette (B2)

(11) Japanese Patent No.  
Patent No. 3179054  
(P3179054)

(45) Date of publication: June 25, 2001 (6.25.2001)

(24) Date of registration: April 13, 2001 (4.13.2001)

(51) Int. Cl.<sup>7</sup>  
G06F 1/08

Identification Code

FI  
G06F 1/04 320 A

Number of Claims: 7 (Total 13 pages)

(21) Application No.: Patent Application No. H9 (1987)-  
235012

(22) Date of Application: August 29, 1997 (8.29.1997)

(65) Publication Number: Japanese Unexamined Patent  
Application Publication Official Gazette No. H11 (1999)-  
73237

(43) Date of Publication: March 16, 1999 (3.16.1999)  
Date of Request for Examination: May 21, 1999 (5.21.1999)

(73) Patent Holder: 000005223  
Fujitsu Limited  
1-1 Kamikodanaka 4-chome, Nakahara-ku, Kawasaki-shi,  
Kanagawa Prefecture, Japan  
(73) Patent Holder: 000237639  
Fujitsu Kiden Limited  
1776 Yano-kuchi, Inagi-shi, Tokyo, Japan  
(72) Inventor: Tatsunari Maniwa  
Inside Fujitsu Kiden Limited, 1776 Yano-kuchi, Inagi-shi,  
Tokyo, Japan  
(72) Inventor: Hideki Yokoyama  
Inside Fujitsu Kiden Limited, 1776 Yano-kuchi, Inagi-shi,  
Tokyo, Japan  
(74) Agent: 100074099  
Patent Attorney Yoshiyuki Osuga (and 1 other)  
  
Examiner: Hideo Mizuno

Continued on the last page

(54) [Title of Invention] Data Processing Device and Clock Switching Method

1

(57) [Scope of Patent Claims]

[Claim 1] A data processing device comprising:  
a communication speed storage means for storing communication  
speed data indicating the communication speed;  
a transfer data storage means for storing transfer data;  
a detection means for detecting access to the aforementioned  
communication speed storage means and the aforementioned  
transfer data storage means; and  
a clock switching means that switches a clock signal serving as a  
reference for CPU operation based on the communication speed  
data stored in the aforementioned communication speed storage  
means when access to the aforementioned communication speed  
storage means is detected by the aforementioned detection means,  
and switches to a clock signal of either the frequency during  
normal operation or a lower frequency during standby mode when  
it is detected by the aforementioned detection means that there  
has been no access to the aforementioned transfer data storage  
means for a certain amount of time.

2

[Claim 2] A data processing device comprising:  
a communication speed storage means for storing communication  
speed data indicating the communication speed;  
a transfer data storage means for storing transfer data;  
a detection means for detecting access to the aforementioned  
communication speed storage means and the aforementioned  
transfer data storage means; and  
a clock switching means that switches the frequency of the clock  
signal serving as a reference for CPU operation based on clock  
information notified from an application program, and when  
access to the aforementioned communication speed storage  
means is detected by the aforementioned detection means,  
switches the clock signal serving as a reference for CPU operation  
based on the communication speed data stored in the  
aforementioned communication speed storage means, and  
switches to a clock signal of either the frequency during normal  
operation or a lower frequency during standby mode when it is  
detected by the aforementioned detection

10

means that there has been no access to the aforementioned transfer data storage means for a certain amount of time.

[Claim 3] A data processing device according to Claim 1 or 2, which is a portable terminal device.

[Claim 4] A data processing device according to Claim 1, 2, or 3, wherein the aforementioned clock switching means has a counter that starts counting up when access to the aforementioned communication speed storage means is detected by the aforementioned detection means, and resets the count value when access to the aforementioned transfer data storage means is detected by the aforementioned detection means.

[Claim 5] A data processing device according to Claim 1 or 2, wherein the aforementioned clock switching means is configured as a BIOS program, wherein the aforementioned BIOS reads communication speed data from the aforementioned communication speed storage means based on a signal output when access to the aforementioned communication speed storage means is detected by the aforementioned detection means, switches the frequency of the clock signal, and switches the clock signal to the frequency during normal operation or a lower frequency during standby mode based on the signal output when it is detected by the aforementioned detection means that there has been no access to the aforementioned transfer data storage means for a certain amount of time.

[Claim 6] A clock switching method, wherein communication speed data indicating the communication speed is stored in a communication speed storage unit; access to the aforementioned communication speed storage unit and a transfer data storage unit that stores transfer data is detected; when access to the aforementioned communication speed storage unit is detected, a clock signal serving as a reference for CPU operation is switched based on the communication speed data stored in the aforementioned communication speed storage unit; and when it is detected that there has been no access to the aforementioned transfer data storage unit for a certain amount of time, the clock signal is switched to either the frequency during normal operation or a lower frequency during standby mode.

[Claim 7] A clock switching method according to Claim 6, wherein the user can set the frequency of the clock signal supplied to the CPU during the execution of an application program that the user wants to run at high speed among multiple executable application programs.

[Detailed Explanation of the Invention]

[0001]

[Technical Field to Which the Invention Belongs] This invention relates to a data processing device and a clock switching method that has a function for switching the frequency of a clock signal.

[0002]

[Conventional Technology] As the processing capability of microprocessors increases, their operating clocks have also been accelerated. In battery-powered portable personal computers, power-saving measures are being implemented to extend the usable time of devices. Since the power consumption of a

microprocessor increases in proportion to the frequency of its operating clock, when the microprocessor is not executing any processes, it operates at a low-speed clock to reduce power consumption. For example, when the power is on and no key input is detected for a certain amount of time, the clock frequency is switched to a lower speed to minimize power consumption.

[0003] On the other hand, portable terminal devices such as handheld terminals that have an input unit and a display unit, which store data read by an optical reading unit like a barcode reader or data entered from the input unit into internal memory, have been widely used. Similar to portable personal computers, these portable terminal devices are powered by batteries, so a power-saving mode is established to switch the clock to a lower speed when no input is detected to extend the available usage time. [0004] In addition, there have been considerations for allowing users to switch the frequency of the microprocessor's operating clock, enabling them to set a lower clock frequency to extend the device's available usage time.

[0005]

[Problems to be Solved by the Invention] However, setting the microprocessor's clock frequency lower to extend usable time can lead to problems, such as a decrease in achievable communication speed during data transfer even when the processing speed is sufficient for normal operations.

[0006] The challenge of this invention is to reduce power consumption during normal operation while achieving high-speed processing.

[0007]

[Means for Solving the Problems] The data processing device described in Claim 1 comprises a clock switching means for switching the frequency of a clock signal that serves as a reference for CPU operation based on clock switching information notified from an application program, and an execution means for executing the application program based on the clock signal switched by the clock switching means.

[0008] According to the invention described in Claim 1, since the frequency of the CPU (including microprocessors, etc.) clock signal can be instructed to switch from the application program side, it is possible to achieve high-speed processing by setting the clock signal frequency higher when executing communication processing than during normal operation, and to operate the CPU at a lower frequency during normal operation, thereby reducing power consumption and extending the available usage time of the device.

[0009] The data processing device described in Claim 5 comprises a detection means for detecting the startup of a process that needs to be executed at a speed faster than normal, and a clock switching means for switching the frequency of the clock signal that serves as a reference for CPU operation to a

frequency higher than during normal operation when the startup of that process is detected by the detection means.

[0010] According to the invention described in Claim 5, since the CPU's clock signal can be set higher when executing processes that require high-speed processing, it is possible to reduce power consumption during normal operation while achieving high-speed processing.

[0011] The data processing device described in Claim 7 comprises a communication speed storage means for storing communication speed data that indicates the speed of data communication, a detection means for detecting whether there has been access to the communication speed storage means, a reading means for reading the communication speed data stored in the communication speed storage means when access to it is detected by the detection means, and a clock switching means for switching the frequency of the clock signal that serves as a reference for CPU operation according to the read communication speed data.

[0012] According to the invention described in Claim 7, during data communication, the CPU can be operated with a clock signal frequency corresponding to the communication speed, allowing the CPU to operate at a processing speed that satisfies the required communication speed.

[0013] The data processing device described in Claim 9 comprises a setup means for setting at least the name of a process and the frequency of the CPU clock signal when executing that process during setup, and a storage means for storing the process name set by the setup means and the clock signal frequency when executing that process.

[0014] According to the invention described in Claim 9, during setup, it is possible to set the name of the process to be processed at high speed and its execution frequency, which allows the CPU to operate at a relatively low frequency during normal operation to reduce the device's power consumption, and to operate at a high frequency for specific processes to shorten processing time.

[0015]

[Embodiment of the Invention] The following describes this embodiment of the invention with reference to the drawings. Here, we will explain the case when this invention is applied to a handheld terminal. A handheld terminal is a portable input terminal device that comprises an input unit for entering data, a display unit, and a communication unit. It has the functionality to store the entered data or the processing results of that data internally and transfer the stored data to a host computer or similar.

[0016] In Fig. 1, the chipset 11 comprises a main CPU 12, a clock generator 13 including a digital PLL (DPLL), a real-time clock (RTC) 15, a programmable interval timer (PIT) 16, and so on.

[0017] The clock generator 13 is a circuit that multiplies and divides the 32KHz reference clock signal generated by the crystal oscillator 14 to produce clock signals of desired

frequencies. As shown in Fig. 2, the clock generator 13 comprises a digital PLL circuit 21 that multiplies the 32KHz signal generated by the crystal oscillator 14 (the phase locked loop; composed of a phase comparator, error amplifier, voltage-controlled oscillator, etc.), a division circuit 22 that further divides the signal multiplied by the digital PLL circuit 21 into 1/2, 1/3, and 1/4 to output signals such as 66MHz, 33MHz, and 16MHz, a division circuit 23 that divides the signals output from both the digital PLL circuit 21 and the division circuit 22, a selector that selects one of the output from division circuit 23 and the 14.318MHz signal output from the digital PLL circuit 21, and an output circuit 25 that determines whether to output the output from selector 24 externally or not, along with a division circuit 26 that further divides the output signal from output circuit 25 by 1/2 and outputs it.

[0018] The CPU clock switching register 27 in the clock generator 13 is written by the main CPU 12 with the frequency specified by the setup menu or application program described later. The frequency written to this CPU clock switching register 27 determines the multiplication factor of the digital PLL circuit 21 and the division ratio of the division circuit 22, thereby generating the clock signal of the specified frequency.

[0019] The CPU clock division register 28 stores data for further dividing the signal multiplied by the digital PLL circuit 21 or the signal divided by the division circuit 22. For example, data is written to output a low-frequency clock signal from the division circuit 23 during sleep mode.

[0020] The 14MHz setting enable register 29 is a register where data is written to specify whether to select the 14.318MHz clock signal using selector 24. Normally, data is written to disable the setting of the 14.318MHz clock signal, so that selector 24 selects the output signal from division circuit 23. Then, when switching the frequency of the clock signal, data that enables the setting of the 14.318MHz signal is written to the 14MHz setting enable register 29, causing selector 24 to select the 14.318MHz clock signal. As a result, when the frequency of the clock signal is switched, the clock signal's frequency is temporarily switched to 14.318MHz before it changes to the specified frequency.

[0021] The CPU clock enable register 30 is a register that stores data specifying whether to output the clock signal from output circuit 25 or not. Typically, data is written that permits the output of the clock signal. For example, if the CPU operation needs to be halted during sleep mode, data that disallows the output of the clock signal is written.

[0022] Returning to Fig. 1, the real-time clock (RTC) 15

comprises a timing unit that measures the date and time, and memory that stores operational conditions set by the setup menu, and it is backed up by a battery so that it can continue measuring the date and time even when powered off.

[0023] The programmable interval timer (PIT) 16 consists of three (3) independent timers/counters that can be used programmatically. The programmable interrupt controller (PIC) 17 accepts interrupt requests from external sources and interrupts the main CPU 12.

[0024] The communication IC 18 is a circuit for transmitting and receiving data. Transmitted data is output from the RS232C port 42 via the driver/receiver 41, while received data is input to the communication IC 18 through the RS232C port 42 and the driver/receiver 41.

[0025] A keyboard (KB) 43 for data input and a liquid crystal display (LCD) 44 with a touch panel (TP) 45 are connected to the keyboard controller 46. Operation signals from the touch panel 44 or keyboard 43 are detected by the keyboard controller 46 and output to the chipset 11. The keyboard controller 46 has an internal CPU and operates based on a clock generated by oscillator 47.

[0026] The liquid crystal display (LCD) 44 displays data input from the keyboard 43 and shows input keys or graphics indicating the content of the input keys on the switch parts of the transparent touch panel 45. The video graphic array (VGA) is a circuit that controls the display of the liquid crystal display 44. The PC card controller 49 is a controller that manages PC cards inserted into PC card slots 50 and 51, having the capability to directly execute program codes from the ROM or RAM within the PC card. The aforementioned keyboard controller 46, video graphic array 48, and PC card controller 49 are connected to the chipset 11 via the ISA bus 53.

[0027] The communication controller 52 is a controller that manages data transfer via infrared using the light-emitting and light-receiving element 54. Transmission signals output from the communication controller 52 are sent through the driver 55 to the light-emitting and light-receiving element 54, while received signals detected by the light-emitting and light-receiving element 54 are output back to the communication controller 52 through the driver 55.

[0028] The flash ROM 56 is a 16M-bit ROM that stores control programs for overall circuit management and application programs like data transfer processing. The RAM 57 consists of two (2) 2M-bit DRAMs that store data input from the keyboard 43. The RAM 58 is a 2M or 4M-bit DRAM that holds parity

data for the data stored in RAM 57. These ROM 56, RAM 57, and RAM 58 are connected to the chipset 11 via a local bus 59, which enables high-speed data transfer.

[0029] The printer controller 60 is a circuit that controls the printing of the printer 61 and has its own CPU for control purposes. Signals output from this printer controller 60 are amplified by the driver 62 and supplied to the print head of the printer 61 for printing.

[0030] The power control unit 63 is a circuit that rectifies the alternating voltage supplied from the charging terminal 64 to create a power voltage for driving various circuit components. The rectified voltage is supplied to the lithium secondary battery 65 to charge the lithium secondary battery 65. In addition, the power control unit 63 supplies the voltage from the lithium battery 66 to the memory of the real-time clock 15, ensuring that the data in memory is protected even when powered off or if the lithium secondary battery 65 discharges.

[0031] Next, Fig. 3 is a block diagram showing the configuration of the communication IC 18. The select/control logic 72 is a circuit that controls the selection of registers within the communication IC 18 and the writing and reading of data to/from the selected register based on address data A0 to A2 output from the main CPU 12 and control signals such as the read signal RD and write signal WR. The data D0 to D7 sent from the main CPU 12 is first stored in the data buffer 71 and then transferred to the appropriate register via the internal data bus under the control of the select/control logic 72.

[0032] The modem control register 73 is a register where data is written to control the modem control logic 74. The modem control logic 74 sends and receives modem status signals with the modem of the communication partner according to the data written in the modem control register 73, establishing the state of both the sending and receiving sides. The modem status register 75 is a register that stores data indicating the state of the modem control logic 74.

[0033] When transmitting data, the communication speed data indicating the transmission rate is first written to the data buffer 71. This communication speed data in the data buffer 71 is then written to the communication rate setting register 76 under the control of the select/control logic 72. The baud rate generator 77 generates a baud rate clock corresponding to the frequency of the communication speed data and outputs it to the transmission timing control 78 and externally.

[0034] Next, when the transmission data is written to the data buffer 71, the data is transferred to the transmission register 79, and further converted into serial data by the transmission shift register 80 for output to the external system. The shifting operation of the transmission shift register 80 is controlled by the transmission timing control 78, and the shifting operation of

the communication shift register 80 occurs at a timing synchronized with the baud rate clock from the baud rate generator 77.

[0035] When receiving data, serial received data is converted into parallel data by the reception shift register 81 and written to the reception buffer register 82. The parallel data is then written to the data buffer 71 and transferred to the main CPU 12. The shifting operation of the reception shift register 81 occurs at a timing synchronized with the receive clock received by the reception timing control 83.

[0036] Next, the operation of switching the frequency of the clock signal for a specific process in a handheld terminal with the configuration described above will be explained. First, the 1st embodiment of this invention, which involves setting the frequency of the clock signal in the setup menu, will be described with reference to the flowchart in Fig. 4.

[0037] The clock information indicating the frequency of the clock signal, etc., is stored in the memory of the real-time clock 15 of the chipset 11, and this memory is backed up by a battery so that data is retained even when the power is off.

[0038] When the power is turned on, the BIOS (a part of the operating system that controls input devices) initialization process is activated (Fig. 4, S1). Once the initialization process is activated, it is then determined whether a specific key was operated during startup (S2). If the specific key has not been operated, the process proceeds to step S3, where the clock information for normal operation stored in the memory of the real-time clock 15 is read out. Based on the stored clock information, the frequency of the clock signal is written to the CPU clock switch register 27 of the clock generator 13.

[0039] On the other hand, if it is determined in step S2 that a specific key was operated, the setup process is activated, and the setup screen is displayed (S5). In this setup process, three (3) types of frequencies are displayed as the clock signal frequency for normal operation: 16M, 33M, and 66MHz. In addition, three different frequencies are shown as the clock signal frequency for processing names that the user may want to execute at high speed (e.g., data transfer processing). The user can select the normal operational frequency and the processing they wish to perform at high speed from the displayed options along with the clock signal frequency for that process.

[0040] It is then determined whether the selection of the clock signal frequency switch has been completed (S6). Once the selection of the processing name and frequency is finished, the selected processing name, frequency, and other clock information are written to the previously mentioned memory of the real-time clock 15 (S7).

[0041] If no processing name is selected in step S6 and only the normal operational frequency is set, the clock information

previously stored in memory will be overwritten with the newly selected frequency. In addition, if both the processing name and the clock signal frequency are selected, the clock information consisting of the processing name and the frequency for executing that process will be written to another memory area of the real-time clock 15.

[0042] After the clock information in the memory is rewritten and the system is reset, the BIOS reads the frequency stored in the memory of the real-time clock 15 and writes that read frequency to the CPU clock switch register 27 of the aforementioned clock generator 13. As a result, a clock signal of the specified frequency is output from the clock generator 13, and the main CPU 12 operates at that frequency.

[0043] In addition, when the processing selected in the setup program is activated, the frequency corresponding to the processing name stored in the memory of the real-time clock 15 is read out and written to the CPU clock switch register 27. This allows the frequency of the clock signal output from the clock generator 13 to be switched when the specified processing is executed.

[0044] By allowing the user to set the frequency of the clock signal for normal operation and the frequency for executing specific processes through the setup menu, it is possible, for instance, to lower the frequency of the clock signal during normal operation and increase the frequency during execution of high-speed processes such as data transfer. This results in reduced power consumption during normal operation, extending the usable time of the handheld terminal while simultaneously enabling fast data transfer.

[0045] Next, the 2nd embodiment of this invention will explain how to instruct the switching of the clock signal frequency from the application program side. Fig. 5 is an explanatory diagram of a command that instructs the switching of the clock signal frequency. The command for indicating the frequency switch is "SYSCLK," with the value of the parameters following the command specifying the frequency. For example, if "0" is specified as a parameter, the clock signal frequency will switch to the default value, which is the frequency at startup. If "1" is specified as a parameter, the clock signal frequency will switch to 16MHz; if "2" is specified, it will switch to 33MHz; and if "3" is specified, it will switch to 66MHz.

[0046] Fig. 6 shows an example of a program for switching the clock signal frequency from the application program side. The figure illustrates a program created in C language, where system("command") is an instruction that executes the string

indicated by "command" as a DOS command.

[0047] Assuming the default frequency is 16MHz, processing A is executed with a clock signal of 16MHz. When processing A is completed, and the next processing B executes system("sysclk 3"), the program specified by that command will be activated.

[0048] Fig. 7 is a flowchart of the frequency switch process activated by the command of the application program. First, the specified parameters are checked (Fig. 7, S11), and it is determined whether the parameter value is out of the range 0 to 3 (S12). If the parameter value is out of range, the process proceeds to step S24, sets a parameter indicating abnormal termination, and ends the process.

[0049] If the specified parameter is determined to be within range at step S12, the process moves to step S13, where the main CPU 12 is set to interrupt-disabled status to prevent control from being passed to other processes during the clock signal switching. Then, it is determined whether the parameter value is "0" or not, which indicates whether to switch to the default frequency (S14). If the parameter value is determined to be "0," the process proceeds to step S15, where the default clock information (clock signal frequency) stored in the memory of real-time clock 15 is read out. That read frequency is then written to the CPU clock switch register 27 (S16). As a result, the frequency of the clock signal generated by the clock generator 13 is switched to the default value (for example, 16MHz).

[0050] It should be noted that when rewriting the contents of the CPU clock switch register 27, the 14MHz setting enable register 29 is first enabled, allowing the selector 24 to select a 14.318MHz signal to temporarily switch the clock signal frequency to 14.318MHz. After that, the frequency of the CPU clock switch register 27 is rewritten, and once the frequency switch is complete, the 14MHz setting enable register 29 is disabled, allowing the selector 24 to select the clock signal of the changed frequency.

[0051] When it is determined in step S14 that the parameter value is not "0," the process proceeds to step S17, where it checks whether the parameter value is "1," indicating a command to switch to 16MHz. If the parameter value is "1" and indicates a switch to 16MHz, the process advances to step S18, where the value of 16MHz is written to the CPU clock switch register 27. This results in the frequency of the clock signal generated by the clock generator 13 being switched to 16MHz.

[0052] If, at step S17, the parameter value is determined not to be "1," the process moves to step S19 to check whether the parameter value is "2," indicating a command to switch to 33MHz. If the parameter value is "2" and indicates a switch to

33MHz, the process advances to step S20, where the value of 33MHz is written to the CPU clock switch register 27. Consequently, the frequency of the clock signal generated by the clock generator 13 is switched to 33MHz.

[0053] If it is determined at step S19 that the parameter value is not "2," then it means the parameter value is "3," which indicates a command to switch to 66MHz. The process then moves to step S21, where the value of 66MHz is written to the CPU clock switch register 27. As a result, the frequency of the clock signal generated by the clock generator 13 is switched to 66MHz.

[0054] Once the frequency switching of the clock signal is completed as described above, the process advances to step S22, where interrupts to the main CPU 12 are enabled. This indicates that the frequency switching process has been completed, so a normal termination parameter is set (S23), and the frequency switching process is concluded, returning control to DOS.

[0055] With the frequency of the clock signal switched as described above, processing B shown in Fig. 6 is executed with a 66MHz clock signal. At the end of processing B, a command can be executed to switch the clock signal frequency back to the default (in this case, 16MHz), thereby restoring the clock signal frequency to its original 16MHz.

[0056] According to this 2nd embodiment, it is possible to switch the operating frequency of the main CPU 12 from the application program side. This allows the main CPU 12 to operate at a lower frequency during normal use, reducing current consumption and extending the usable time of the device. When high-speed processing is required, the main CPU 12 can operate at a higher frequency, thereby shortening processing time.

[0057] Next, the 3rd embodiment of this invention is explained with reference to Fig. 8 and Fig. 9, which involves detecting access to specific registers in order to switch the clock signal frequency.

[0058] Fig. 8 is a block diagram of an access detection circuit 90 that detects accesses to the communication IC 18 during data transfer. When sending data from a handheld terminal to a host computer, the communication speed is set by writing the communication speed to the communication rate setting register 76 of the communication IC 18 from the main CPU 12. The decode circuit 91 decodes the address data output from the main CPU 12 and, upon detecting that the address data matches the address of the communication rate setting register 76, outputs an address match signal to the access monitoring circuit 92. When the access monitoring circuit 92 receives the address match signal from the decode circuit 91, it instructs the interrupt generation circuit 93 to output an interrupt signal and activates its internal counter.

[0059] After the communication speed is set in the

communication rate setting register 76 of the communication IC 18 and data transfer begins, data is written to the transmission register 79 at regular intervals. The decode circuit 91 compares the address data output from the main CPU 12 with the address of the transmission register 79, and if the addresses match, it outputs an address match signal to the access monitoring circuit 92. Every time the access monitoring circuit 92 receives the address match signal for the transmission register 79, it resets its counter.

[0060] Once the data transmission is complete and access to the transmission register 79 ends, the address match signal for the transmission register 79 will no longer be output from the decode circuit 91, causing the counter in the access monitoring circuit 92 to count up. When the counter counts up, the access monitoring circuit 92 instructs the interrupt generation circuit 93 to output an interrupt signal.

[0061] In other words, interrupt signals are output from the interrupt generation circuit 93 when setting the communication speed during data transmission and when the data transmission is completed. Thus, when the BIOS detects this interrupt signal, it can switch the clock signal frequency, enabling high-speed processing of the data transfer operation.

[0062] Next, Fig. 9 is a flowchart showing the processing steps for switching the frequency based on an interrupt signal from the interrupt generation circuit 93. Upon detecting the interrupt signal, the register of the programmable interrupt controller 17 is referenced to check the cause of the interrupt (Fig. 9, S31). If the interrupt cause relates to data transfer, it first determines whether there has been access to the communication rate setting register 76 (S32). If it is determined in step S32 that there has been access to the communication rate setting register 76, the process moves to step S33, where the communication speed stored in the communication rate setting register 76 is read out. The next step (S34) checks whether the read communication speed is below a predetermined value (for example, 38.4 KBPS) or exceeds this value.

If the read communication speed is determined to be below the predetermined value, the process advances to step S35, where the clock frequency of 33MHz is written to the CPU clock switch register 27 of the clock generator 13. This results in the clock generator 13 outputting a 33MHz clock signal as the operating clock for the main CPU 12.

[0063] On the other hand, if it is determined in step S34 that the communication speed exceeds the predetermined value, the process moves to step S36, where the clock frequency of 66MHz is written to the CPU clock switch register 27 of the clock generator 13. Consequently, the clock generator 13 outputs a 66MHz clock signal as the operating clock for the main CPU 12. [0064] If it is determined in step S32 that there has not been access to the communication rate register 76, the process

advances to step S37 to determine whether the counter in the access monitoring circuit 92 has counted up. If the counter has counted up, it is judged that the data transfer processing has completed, and the process moves to step S38 to read the default clock information stored in the memory of the real-time clock 15. Then, the read frequency is written to the CPU clock switch register 27 of the clock generator 13 (S39). As a result, the clock generator 13 outputs the clock signal at the normal operating frequency to the main CPU 12.

[0065] According to this 3rd embodiment, access to registers or memory caused by processing that requires high-speed operations, such as data transfer processing, is detected by hardware. During the execution of such processes, the clock signal frequency of the CPU is increased or switched to a frequency corresponding to the communication speed. Once the processing is complete, it switches back to the normal operating frequency. This allows the main CPU 12 to operate at a lower frequency clock signal during normal operation, reducing power consumption and extending the device's available time. When executing processes that require high-speed processing, the main CPU 12 can operate at a higher frequency clock signal, achieving fast processing.

[0066] In the aforementioned embodiment, data transfer processing was cited as an example, but the invention is not limited to this; it can be applied to other processes that require high-speed processing. In addition, in the 3rd embodiment, it is also possible to store addresses accessed when a specific process is triggered and detect the activation of that specific process by detecting access to those addresses, thereby switching the clock signal frequency at that time.

[0067] Furthermore, while the above embodiment has explained the application of this invention to handheld terminals, it is not limited to this; it can also be applied to portable personal computers, portable small devices, and other data processing apparatuses.

[0068]

[Effects of the Invention] According to this invention, when executing programs that require high-speed processing, the frequency of the CPU's operating clock is set higher than during normal operation, thereby reducing power consumption during normal operation and extending the device's usable time while simultaneously achieving high-speed processing. Moreover, since the switching of the clock signal frequency can be performed from the application program side, the CPU can be operated at a processing speed suitable for that program, reducing power consumption and shortening processing time. Furthermore, when performing data communication, by operating the CPU at a frequency corresponding to the communication speed, power consumption can be reduced while realizing high-speed data communication.

[Brief Description of Drawings]

[Fig. 1] It is a block diagram of the handheld terminal of this embodiment of the invention.

[Fig. 2] It is a block diagram of the clock generator 13.

[Fig. 3] It is a block diagram of the communication IC 18.

[Fig. 4] It is a flowchart of the frequency switching process in the 1st embodiment.

[Fig. 5] It is an explanatory diagram of a command that instructs frequency switching of the clock signal in the 2nd embodiment.

[Fig. 6] It is a diagram showing an example of the application side program in the 2nd embodiment.

[Fig. 7] It is a flowchart of the frequency switching process in the 2nd embodiment.

[Fig. 8] It is a circuit block diagram of the access monitoring circuit that detects access to the communication IC 18.

10

[Fig. 9] It is a flowchart of the frequency switching process in the 3rd embodiment.

[Explanation of Reference Numerals]

11 Chipset

12 Main CPU

13 Clock Generator

18 Communication IC

27 CPU Clock Switching Register

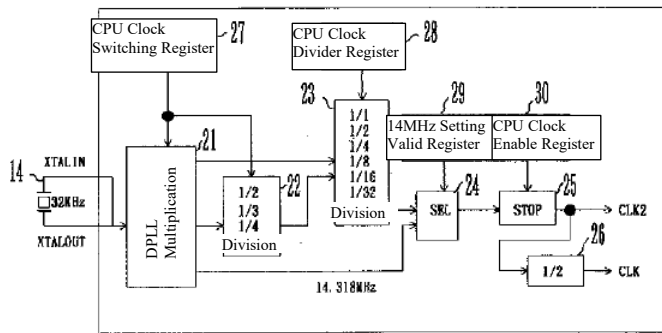
76 Communication Rate Setting Register

91 Decode Circuit

92 Access Monitoring Circuit

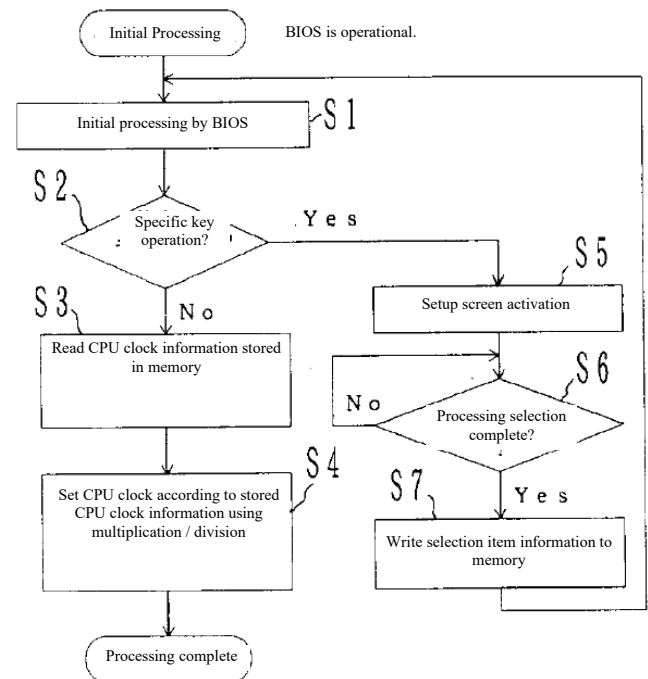
[Fig. 2]

Block diagram of the clock generator 13



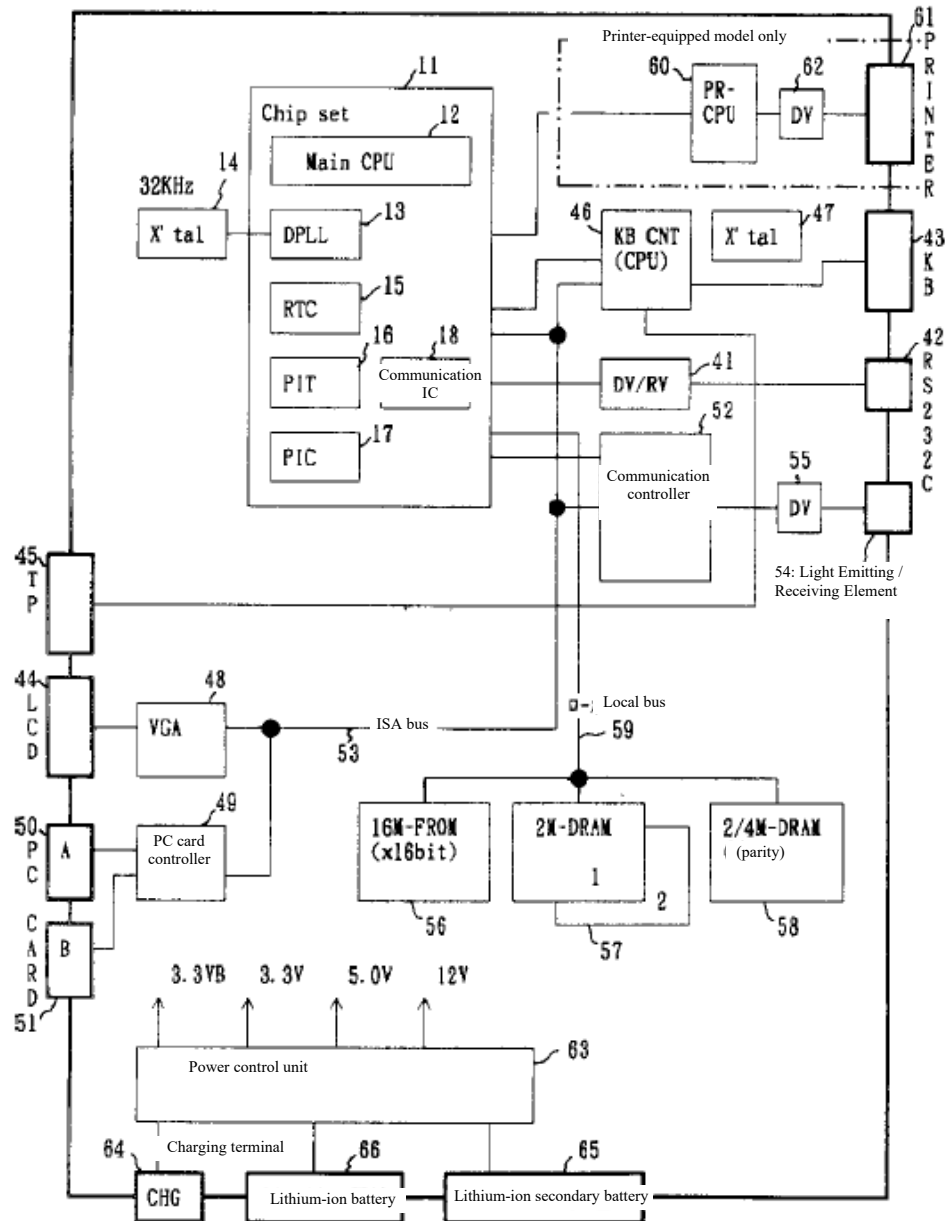
[Fig. 4]

Flowchart of the frequency switching process in the 1st embodiment



[Fig. 1]

Block diagram of the handheld terminal



[Fig. 5]

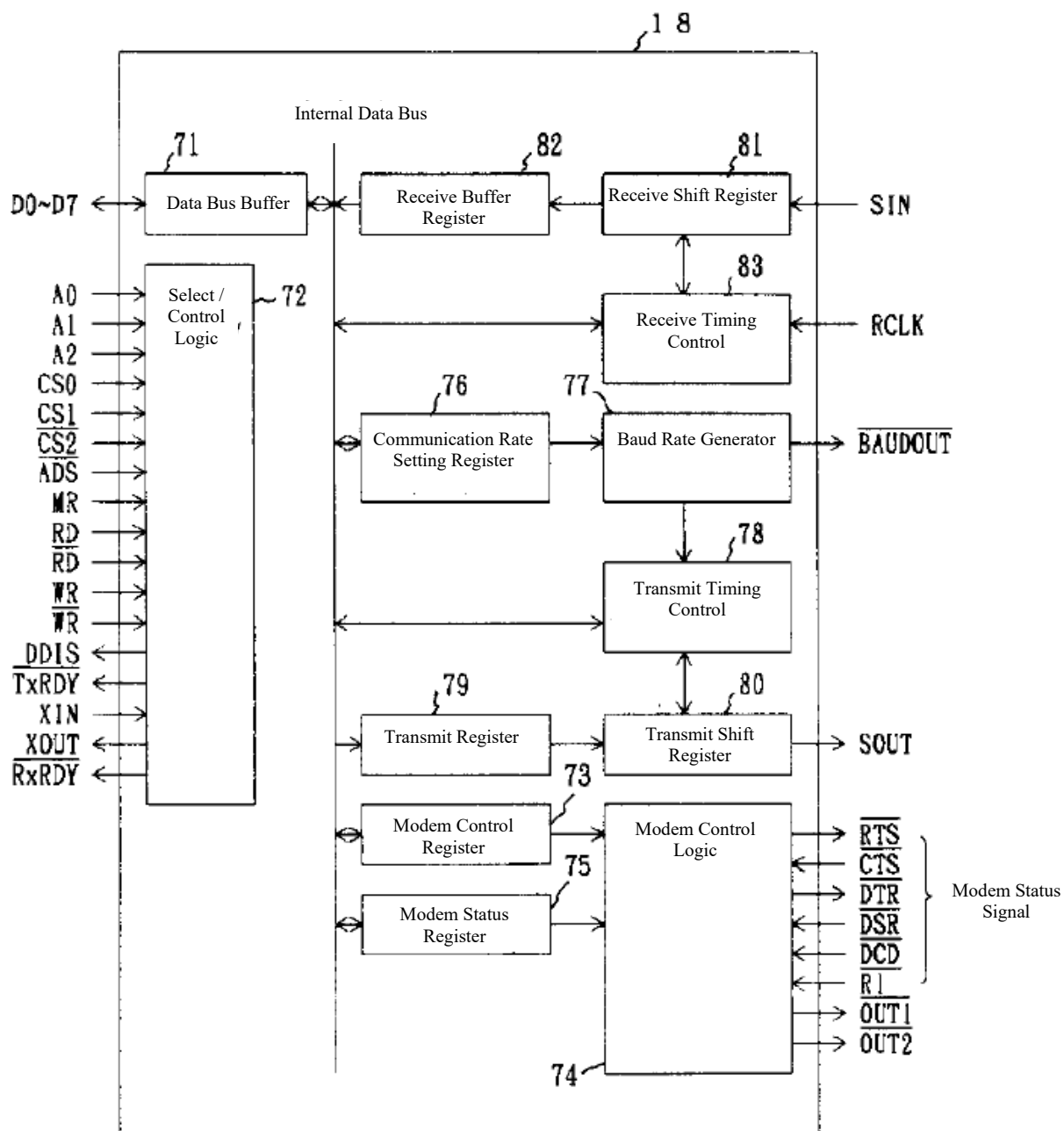
An explanatory diagram of a command that instructs frequency switching of the clock signal in the 2nd embodiment

SYSCLK [Parameter]

[Specified Parameter] 0: Set CPU clock to default (CPU clock at rising edge)  
 1: Switch CPU clock to 16MHz.  
 2: Switch CPU clock to 33MHz.  
 3: Switch CPU clock to 66MHz.

[Fig. 3]

Block diagram of the communication IC 18



[Fig. 6]

An example of the application-side program in the 2nd embodiment

[Format] ret = system("command"); \* The argument command is executed as a DOS command.

<Overview>

```

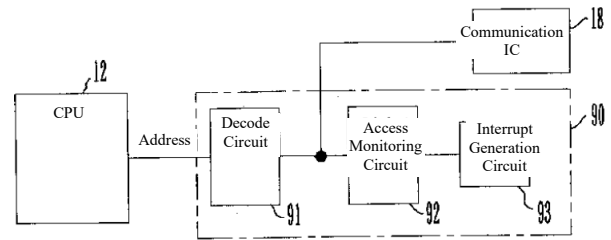
A
.
system('sysclk 3');
.
B
.
system('sysclk 0');
.
C
.

```

↓ A processing is handled at CPU clock 16MHz.  
 ↓ \* Executes the command to switch the CPU clock from 16MHz to 66MHz.  
 ↓ B processing is handled at CPU clock 66MHz.  
 ↓ \* Executes the command to switch the CPU clock to default (in this case, 16MHz).  
 ↑ C processing is handled at CPU clock 16MHz.

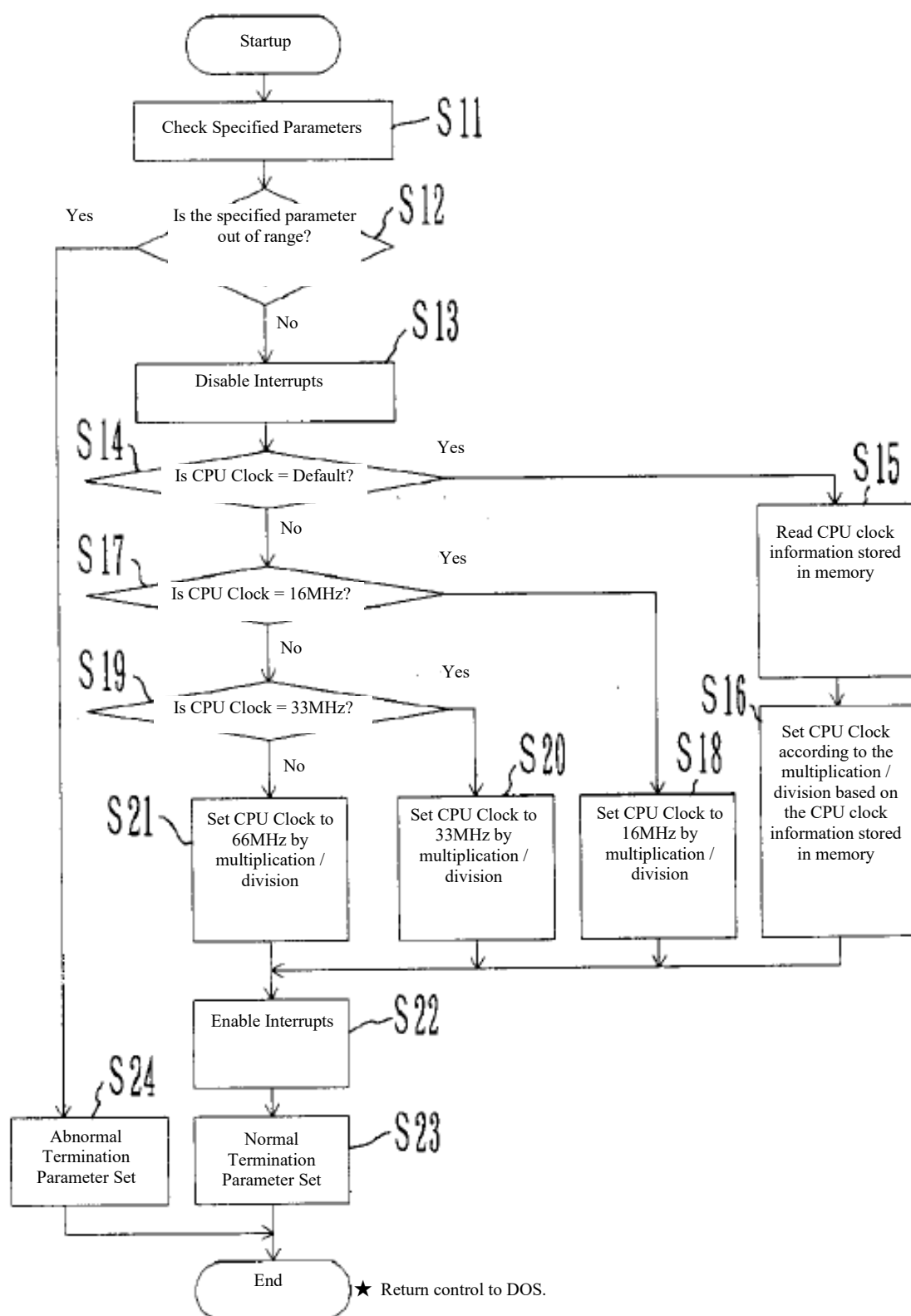
[Fig. 8]

Circuit block diagram of the access detection circuit 90 that detects access to the communication IC 18



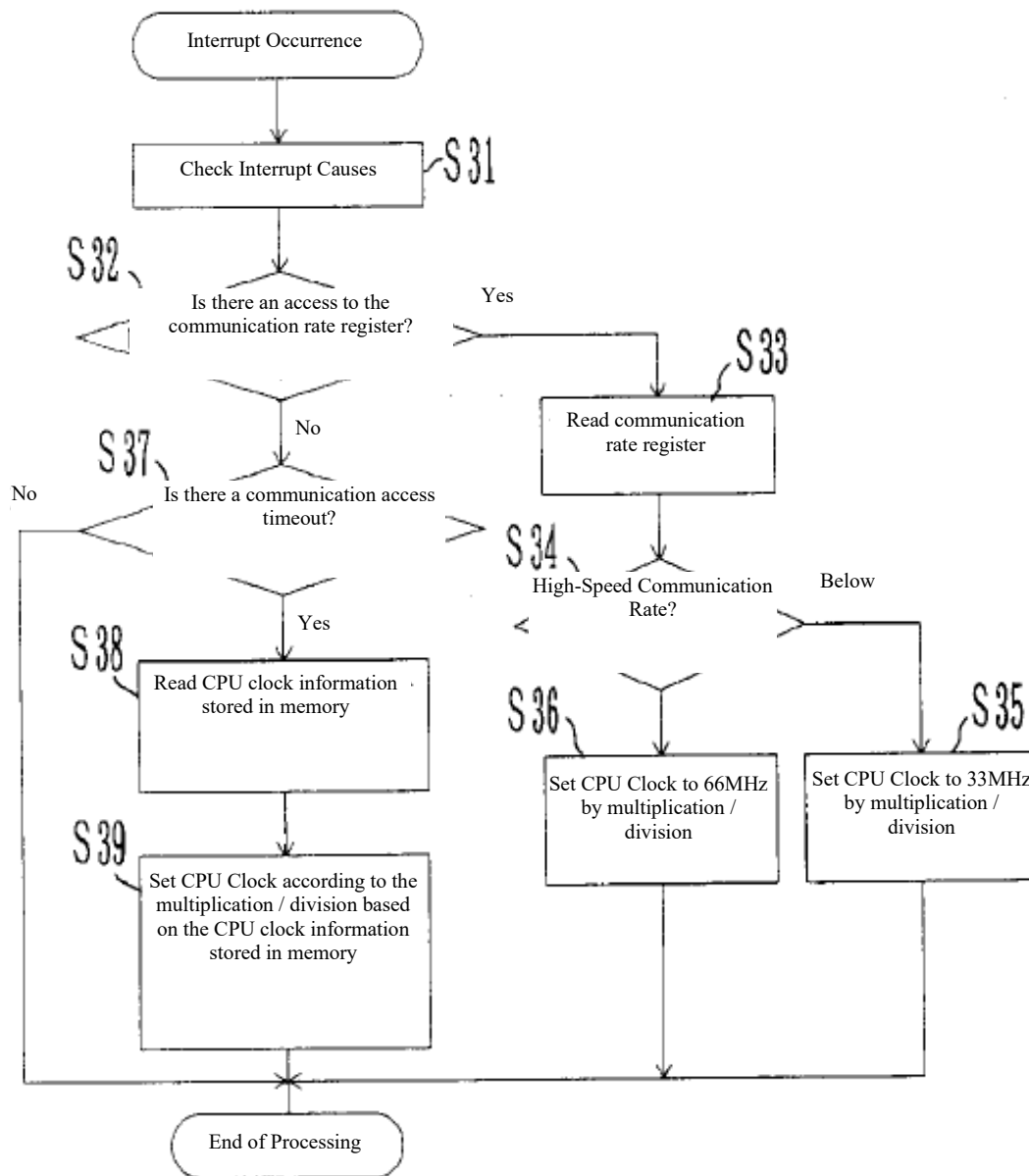
[Fig. 7]

Flowchart of the frequency switching process in the 2nd embodiment



[Fig. 9]

Flowchart of the frequency switching process in the 3rd embodiment



Continuation of the front page

## (56) References

Japanese Unexamined Patent Application Publication  
Official Gazette No. Heisei 2(1990)-3809 (JP, A)  
Japanese Unexamined Patent Application Publication  
Official Gazette No. Heisei 8(1996)-36445 (JP, A)  
Japanese Unexamined Patent Application Publication  
Official Gazette No. Heisei 9(1997)-297688 (JP, A)

(58) Fields Investigated (Int.Cl.<sup>7</sup>, DB Name)  
G06F 1/08



(19)日本国特許庁 (J P)

(12) 特 許 公 報 (B 2)

(11)特許番号

特許第3179054号  
(P3179054)

(45)発行日 平成13年 6 月25日 (2001. 6. 25)

(24)登録日 平成13年 4 月13日 (2001. 4. 13)

(51)Int.Cl.<sup>7</sup>

識別記号

F I

G 0 6 F 1/08

G 0 6 F 1/04

3 2 0 A

請求項の数 7 (全 13 頁)

(21)出願番号 特願平9-235012

(22)出願日 平成 9 年 8 月29日 (1997. 8. 29)

(65)公開番号 特開平11-73237

(43)公開日 平成11年 3 月16日 (1999. 3. 16)

審査請求日 平成11年 5 月21日 (1999. 5. 21)

(73)特許権者 000005223

富士通株式会社

神奈川県川崎市中原区上小田中 4 丁目 1  
番 1 号

(73)特許権者 000237639

富士通機電株式会社

東京都稲城市矢野口1776番地

(72)発明者 真庭 達成

東京都稲城市矢野口1776番地 富士通機  
電株式会社内

(72)発明者 横山 秀樹

東京都稲城市矢野口1776番地 富士通機  
電株式会社内

(74)代理人 100074099

弁理士 大菅 義之 (外 1 名)

審査官 水野 恵雄

最終頁に続く

(54)【発明の名称】 データ処理装置及びクロック切換方法

1

(57)【特許請求の範囲】

【請求項 1】 データの通信速度を示す通信速度データを記憶する通信速度記憶手段と、  
転送データを記憶する転送データ記憶手段と、  
前記通信速度記憶手段及び前記転送データ記憶手段に対するアクセスを検出する検出手段と、  
前記検出手段により前記通信速度記憶手段に対するアクセスが検出されたときには、前記通信速度記憶手段に記憶されている通信速度データに基づいて C P U の動作の基準となるクロック信号を切り換え、前記検出手段により前記転送データ記憶手段に対するアクセスが一定時間以上行われていないことが検出されたときには、通常動作時の周波数または通常動作時より低い待機時の周波数のクロック信号へ切り換えるクロック切り換え手段とを備えることを特徴とするデータ処理装置。

2

【請求項 2】 データの通信速度を示す通信速度データを記憶する通信速度記憶手段と、  
転送データを記憶する転送データ記憶手段と、  
前記通信速度記憶手段及び前記転送データ記憶手段に対するアクセスを検出する検出手段と、  
アプリケーションプログラムから通知されるクロック情報に基づいて C P U の動作の基準となるクロック信号の周波数を切り換え、前記検出手段により前記通信速度記憶手段に対するアクセスが検出されたときには、前記通信速度記憶手段に記憶されている通信速度データに基づいて C P U の動作の基準となるクロック信号を切り換え、前記検出手段により前記転送データ記憶手段に対するアクセスが一定時間以上行われていないことが検出されたときには、通常動作時の周波数または通常動作時より低い待機時の周波数のクロック信号に切り換えるクロ

ック切り換え手段とを備えることを特徴とするデータ処理装置。

【請求項 3】携帯型の端末装置であることを特徴とする請求項 1 または 2 記載のデータ処理装置。

【請求項 4】前記クロック切り換え手段は、前記検出手段により前記通信速度記憶手段に対するアクセスが検出されたときカウントアップを開始し、前記検出手段により前記転送データ記憶手段に対するアクセスが検出されたとき、カウント値をリセットするカウンタを有することを特徴とする請求項 1、2 または 3 記載のデータ処理装置。

【請求項 5】前記クロック切り換え手段は、BIOS のプログラムとして構成され、前記検出手段により前記通信速度記憶手段に対するアクセスが検出されたとき出力される信号に基づいて前記 BIOS が前記通信速度記憶手段から通信速度データを読み出し、クロック信号の周波数を切り換え、前記検出手段により前記転送データ記憶手段に対するアクセスが一定時間以上行われていないことが検出されたとき出力される信号に基づいてクロック信号を通常動作時の周波数または通常動作時より低い待機時の周波数に切り換えることを特徴とする請求項 1 または 2 記載のデータ処理装置。

【請求項 6】データの通信速度を示す通信速度データを通信速度記憶部に記憶させ、前記通信速度記憶部及び転送データを記憶する転送データ記憶部に対するアクセスを検出し、前記通信速度記憶部に対するアクセスが検出されたときには、前記通信速度記憶部に記憶されている通信速度データに基づいて CPU の動作の基準となるクロック信号を切り換え、前記転送データ記憶部に対するアクセスが一定時間以上行われていないことが検出されたときには、通常動作時の周波数または通常動作時より低い待機時の周波数のクロック信号に切り換えることを特徴とするクロック切り換え方法。

【請求項 7】実行可能な複数のアプリケーションプログラムの中で高速で実行したいアプリケーションプログラムと前記アプリケーションプログラムの実行時に CPU に供給するクロック信号の周波数をユーザが設定できるようにしたことを特徴とする請求 6 記載のクロック切り換え方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、クロック信号の周波数の切り換え機能を有するデータ処理装置及びクロック切り換え方法に関する。

【0002】

【従来の技術】マイクロプロセッサの処理能力が高まると共に、その動作クロックも高速化してきている。電池で駆動される携帯型のパーソナルコンピュータなどでは、機器の使用可能時間を長くするために省電力の工夫

が行われている。マイクロプロセッサの消費電力は動作クロックの周波数に比例して増加するので、マイクロプロセッサが処理を実行していないときには、低速のクロックで動作させて消費電力を減らすことが行われている。例えば、電源がオン状態でキー入力がある一定時間以上行われないときには、クロック周波数を低速に切り換えて消費電力を少なくしている。

【0003】他方、入力部と表示部を備え、バーコードリーダ等の光学読み取り部で読み取ったデータ、あるいは入力部から入力されたデータを内部のメモリに記憶しておいて、記憶したデータをホストコンピュータ等に転送できるハンディターミナル等の携帯端末装置が広く利用されてきている。携帯端末装置も携帯型パーソナルコンピュータと同様に電池により駆動されるので、使用可能な時間を長くするために、入力が行われないときクロックを低速に切り換える省電力モードが設けられている。

【0004】また、マイクロプロセッサの動作クロックの周波数をユーザが切り換えられるようにして、ユーザがクロックの周波数を低く設定することで、機器の使用可能時間を長くすることができるようにしたものも考えられている。

【0005】

【発明が解決しようとする課題】しかしながら、使用可能な時間を長くするためにマイクロプロセッサのクロック周波数を低く設定すると、通常の処理では処理速度が充分であっても、例えばデータ転送時の実行可能な通信速度が遅くなるという問題が生じる。

【0006】本発明の課題は、通常動作時の消費電力を少なくし、かつ高速処理を実現することである。

【0007】

【課題を解決するための手段】請求項 1 記載の発明のデータ処理装置は、アプリケーションプログラムから通知されるクロック切り換え情報に基づいて CPU の動作の基準となるクロック信号の周波数を切り換えるクロック切り換え手段と、クロック切り換え手段で切り換えられたクロック信号に基づいてアプリケーションプログラムを実行する実行手段とを備える。

【0008】請求項 1 記載の発明によれば、アプリケーションプログラム側から CPU（マイクロプロセッサ等を含む）のクロック信号の周波数の切り換えを指示できるので、例えば、通信処理を実行するときのクロック信号の周波数を通常動作時より高くして高速の処理を実現し、通常動作時はそれより低い周波数で CPU を動作させることで消費電力を少なくし、装置の使用可能時間を長くすることができる。

【0009】請求項 5 記載の発明のデータ処理装置は、通常より速い速度で実行したい処理の起動を検出する検出手段と、検出手段でその処理の起動が検出されたとき、CPU の動作の基準となるクロック信号の周波数を

通常動作時より高い周波数に切り換えるクロック切り換え手段とを備える。

【0010】請求項5記載の発明によれば、高速の処理を必要とする処理を実行するときのCPUのクロック信号を通常動作時より高くできるので、通常動作時の消費電力を少なくし、かつ高速処理を実現できる。

【0011】請求項7記載の発明のデータ処理装置は、データの通信速度を示す通信速度データを記憶する通信速度記憶手段と、通信速度記憶手段に対するアクセスが行われたか否かを検出する検出手段と、検出手段で通信速度記憶手段に対するアクセスが行われたことが検出された場合に、通信速度記憶手段に記憶されている通信速度データを読み出す読み出し手段と、読み出された通信速度データに応じてCPUの動作の基準となるクロック信号の周波数を切り換えるクロック切り換え手段とを備える。

【0012】請求項7記載の発明によれば、データ通信のとき、通信速度に応じた周波数のクロック信号でCPUを動作させることができるので、要求される通信速度を満足する処理速度でCPUを動作させることができる。

【0013】請求項9記載の発明のデータ処理装置は、セットアップ時に少なくとも処理名と該処理を実行するときのCPUのクロック信号の周波数を設定する設定手段と、設定手段で設定された処理名とその処理を実行するときのクロック信号の周波数とを記憶する記憶手段とを備える。

【0014】請求項9記載の発明によれば、セットアップ時に、高速で処理したい処理名とその実行周波数とを設定することができるので、通常動作時は比較的低い周波数でCPUを動作させて装置の消費電力を少なくし、特定の処理のときだけ高い周波数でCPUを動作させて処理時間を短くすることができる。

【0015】

【発明の実施の形態】以下、本発明の実施の形態を図面を参照しながら説明する。以下、本発明をハンドヘルドターミナルに適用した場合について説明する。ハンドヘルドターミナルは、携帯型の入力端末装置であり、データを入力する入力部と、表示部と、通信部とを有し、入力したデータ、あるいはそのデータの処理結果を内部に記憶しておいて、記憶したデータをホストコンピュータ等に転送する機能を有している。

【0016】図1において、チップセット11は、メインCPU12、デジタルPLL(DPLL)含むクロックジェネレータ13、リアルタイムクロック(RTC)15、プログラマブルインターバルタイマ(PIT)16等からなる。

【0017】クロックジェネレータ13は、水晶発振器14で生成される32KHzの基準クロック信号を通過、分周して、所望の周波数のクロック信号を生成する

回路である。クロックジェネレータ13は、図2に示すように水晶発振器14で生成される32KHzの信号を通過するデジタルPLL(phase locked loop; 位相比較器、誤差増幅器、電圧制御発振器等からなる)回路21と、そのデジタルPLL回路21で通過された信号を、1/2、1/3、1/4に分周して66MHz、33MHz、16MHz等の信号を出力する分周回路22と、デジタルPLL回路21から出力される信号と分周回路22から出力される信号とを分周する分周回路23と、分周回路23の出力とデジタルPLL回路21から出力される14.318MHzの信号との一方を選択して出力するセレクタと、セレクタ24の出力を外部に出力するか、それとも出力しないかを定める出力回路25と、その出力回路25の出力信号を1/2に分周して出力する分周回路26等で構成されている。

【0018】クロックジェネレータ13のCPUクロック切替レジスタ27には、後述するセットアップメニュー、あるいはアプリケーションプログラムにより指定される周波数がメインCPU12により書き込まれる。このCPUクロック切替レジスタ27に書き込まれる周波数により、デジタルPLL回路21の通過率、分周回路22の分周比が決められ、指定された周波数のクロック信号が生成される。

【0019】CPUクロック分周レジスタ28には、デジタルPLL回路21で通過された信号または分周回路22で分周された信号を、さらに分周する為のデータが書き込まれる。例えば、スリープモードのときに低周波数のクロック信号を分周回路23から出力させるためのデータが書き込まれる。

【0020】14MHz設定有効レジスタ29は、セレクタ24で14.318MHzのクロック信号を選択するか否か指定するデータが書き込まれるレジスタであり、通常はセレクタ24が分周回路23の出力信号を選択するように、14.318MHzのクロック信号の設定を無効とするデータが書き込まれている。そして、クロック信号の周波数の切り換え時に、14.318MHzの信号の設定を有効にするデータが14MHz設定有効レジスタ29に書き込まれ、セレクタ24は14.318MHzのクロック信号を選択する。これにより、クロック信号の周波数の切り換え時には、クロック信号の周波数が一旦14.318MHzに切り換えられた後、指定された周波数に切り換えられる。

【0021】CPUクロック許可レジスタ30は、出力回路25からクロック信号を出力するか、それとも出力しないかを指定するデータが記憶されるレジスタであり、通常はクロック信号の出力を許可するデータが書き込まれており、例えば、スリープモードにおいてCPUの動作を停止させる場合に、クロック信号の出力を不許可とするデータが書き込まれる。

【0022】図1に戻り、リアルタイムクロック15

は、日付及び時刻を計時する計時部と、セットアップメニューで設定される動作条件等を記憶するメモリとを有し、電源オフ時にも日付及び時刻の計時を行えるように電池によりバックアップされている。

【0023】プログラマブルインターバルタイマ16は、独立した3個のタイマ／カウンタからなり、これらのタイマ／カウンタはプログラムにより使用される。プログラマブルインターラプトコントローラ(PIC)17は、外部からの割り込み要求を受け付け、メインCPU12に対して割り込みをかける。

【0024】通信IC18は、データを送受信するための回路であり、送信データはドライバ／レシーバ41を介してRS232Cポート42から出力され、受信データはRS232Cポート42、ドライバ／レシーバ41を介して通信IC18に入力される。

【0025】データ入力を行うキーボード(KB)43と液晶表示部(LCD)44のタッチパネル(TP)45は、キーボードコントローラ46に接続されており、タッチパネル44またはキーボード43の操作信号がキーボードコントローラ46により検出されてチップセット11に出力される。キーボードコントローラ46は内部にCPUを有し、発振器47で生成されるクロックに基づいて動作する。

【0026】液晶表示部44は、キーボード43から入力されたデータを表示すると共に、透明タッチパネル45のスイッチの部分に入力キー、あるいは入力キーの内容を示す図形等を表示する。ビデオグラフィックアレイ(VGA)は、液晶表示部44の表示を制御する回路である。PCカードコントローラ49は、PCカードスロット50、51に挿入されるPCカードの制御を行うコントローラであり、PCカード内のROMまたはRAMのプログラムコードを直接実行できる機能を有している。上述したキーボードコントローラ46、ビデオグラフィックアレイ48及びPCカードコントローラ49はISAバス53を介してチップセット11に接続されている。

【0027】通信用コントローラ52は、発光・受光素子54を介して行う赤外線によるデータ転送を制御するコントローラである。通信用コントローラ52から出力される送信信号はドライバ55を介して発光・受光素子54に出力され、発光・受光素子54で受光された受信信号はドライバ55を介して通信用コントローラ52に出力される。

【0028】フラッシュROM56は、回路全体の制御を行う制御プログラム及びデータ転送処理等のアプリケーションプログラムを記憶する16MビットのROMである。RAM57は、キーボード43から入力されたデータ等を記憶する2Mビットの2個のDRAMからなる。RAM58は、RAM57に記憶されるデータのパーティティデータを記憶する2Mまたは4MビットのDRA

Mである。これらのROM56、RAM57、58は、高速なデータ転送が可能なローカルバス59によりチップセット11に接続されている。

【0029】プリンタコントローラ60は、プリンタ61の印字を制御する回路であり、制御用のCPUを有している。このプリンタコントローラ60から出力される信号はドライバ62で増幅されてプリンタ61の印字ヘッドに供給されて印字が行われる。

【0030】電源制御部63は、充電端子64から供給される交流電圧を整流して、回路各部を駆動するための電源電圧を作成する回路であり、整流後の電圧をリチウム二次電池65に供給してリチウム二次電池65の充電を行う。また、電源制御部63は、リチウム電池66の電圧をリアルタイムクロック15のメモリ等に供給して、電源オフ時、あるいはリチウム二次電池65が放電した場合でもメモリのデータを保護するようにしている。

【0031】次に、図3は、通信IC18の構成を示すブロック図である。セレクト／コントロールロジック72は、メインCPU12から出力されるアドレスデータA0～A2と、リード信号RD、ライト信号WR等の制御信号に従って通信IC18内のレジスタの選択及びその選択したレジスタへのデータの書き込み及び読み出し等の制御を行う回路である。メインCPU12から送られてくるデータD0～D7は、一旦データバッファ71に格納された後、セレクト／コントロールロジック72の制御の元に内部データバスを介して該当するレジスタに転送される。

【0032】モデム制御レジスタ73は、モデム制御ロジック74を制御するデータが書き込まれるレジスタであり、モデム制御ロジック74はモデム制御レジスタ73に書き込まれたデータに従って、通信相手のモデムとの間でモデムステータス信号を送受信して送信側及び受信側の状態を定める。モデムステータスレジスタ75は、モデム制御ロジック74の状態を示すデータを記憶するレジスタである。

【0033】データを送信する場合には、先ず通信速度を示す通信速度データがデータバッファ71に書き込まれ、そのデータバッファ71の通信速度データがセレクト／コントロールロジック72の制御の元に通信レート設定レジスタ76に書き込まれる。ボーレートジェネレータ77は、通信速度データに対応する周波数のボーレートクロックを生成して送信タイミングコントロール78及び外部に出力する。

【0034】次に、送信データがデータバッファ71に書き込まれると、そのデータが送信レジスタ79に転送され、さらに送信シフトレジスタ80でシリアルデータに変換されて外部に出力される。送信シフトレジスタ80のシフト動作は、送信タイミングコントロール78により制御されており、通信シフトレジスタ80のシフト

動作はボーレートジェネレータ 77 のボーレートクロックに同期したタイミングで行われる。

【0035】データを受信する場合には、受信シフトレジスタ 81 でシリアルを受信データがパラレルデータに変換され受信バッファレジスタ 82 に書き込まれ、さらにそのパラレルデータがデータバッファ 71 に書き込まれてメイン CPU 12 に転送される。受信シフトレジスタ 81 のシフト動作は、受信タイミングコントロール 83 で受信される受信クロックに同期したタイミングで行われる。

【0036】次に、以上のような構成のハンドヘルドターミナルにおいて、特定の処理のクロック信号の周波数を切り換える場合の動作を説明する。まず、セットアップメニューでクロック信号の周波数を設定する本発明の第 1 の実施の形態を、図 4 のフローチャートを参照して説明する。

【0037】クロック信号の周波数等を示すクロック情報はチップセット 11 のリアルタイムクロック 15 のメモリに記憶されており、このメモリは電源オフ時にもデータが保持されるように電池でバックアップされている。

【0038】電源をオンすると、BIOS（オペレーションシステムの機能の一部で入力装置の制御をするシステム）のイニシャル処理が起動される（図 4、S1）。イニシャル処理が起動されると、次に、起動時に特定のキーが操作されたか否かが判別される（S2）。特定のキーが操作されていなければ、ステップ S3 に進みリアルタイムクロック 15 のメモリに記憶されている通常動作時のクロック情報を読み出す。そして、その記憶されているクロック情報に従って、クロックジェネレータ 13 の CPU クロック切替レジスタ 27 にクロック信号の周波数を書き込む。

【0039】他方、ステップ S2 で特定のキーが操作されたと判別された場合には、セットアップ処理が起動され、セットアップ画面が表示される（S5）。このセットアップ処理では、通常動作時のクロック信号の周波数として 16M、33M、66MHz の 3 種類が表示され、さらに処理名とその処理を実行するクロック信号の周波数として 3 種類の周波数が表示されるので、ユーザは通常動作時の周波数と、表示されている処理の内

で高速で実行したい処理（例えば、データ転送処理）と、その処理を実行するときのクロック信号の周波数とを選択する。

【0040】そして、クロック信号の周波数の切り換えの選択が終了したか否かが判別される（S6）。処理名及び周波数の選択が終了すると、選択された処理名、周波数等のクロック情報が前述したリアルタイムクロック 15 のメモリに書き込まれる（S7）。

【0041】ステップ S6 で処理名が選択されず、通常動作時の周波数のみが設定された場合には、それまでメ

モリに記憶されていたクロック情報が新たに選択された周波数に書き換えられる。また、処理名とクロック信号の周波数の両方が選択された場合には、リアルタイムクロック 15 のメモリの別のメモリエリアに、処理名とその処理を実行するときのクロック信号の周波数がクロック情報として書き込まれる。

【0042】メモリのクロック情報が書き換えられた後にシステムをリセットすると、BIOS がリアルタイムクロック 15 のメモリに記憶されている周波数を読み出し、その読み出した周波数を前述したクロックジェネレータ 13 の CPU クロック切替レジスタ 27 に書き込む。これにより、クロックジェネレータ 13 から、指定された周波数のクロック信号が出力され、メイン CPU 12 はその周波数のクロックで動作する。

【0043】また、セットアッププログラムで選択された処理が起動されるときに、リアルタイムクロック 15 のメモリに処理名に対応して記憶されている周波数を読み出され、その周波数が CPU クロック切替レジスタ 27 に書き込まれる。これにより、指定された処理が実行されるとき、クロックジェネレータ 13 から出力されるクロック信号の周波数が切り換えられる。

【0044】上述したようにセットアップメニューで通常動作時のクロック信号の周波数と、特定の処理を実行するときのクロック信号の周波数をユーザが設定できるようにすることで、例えば通常動作時のクロック信号の周波数を低くし、高速処理を要求されるデータ転送処理等の実行時のクロック信号の周波数を高くすることができる。これにより、通常動作時の消費電力を少なくしてハンディターミナルの使用可能時間を長くし、同時に高速なデータ転送を実現できる。

【0045】次に、アプリケーションプログラム側からクロック信号の周波数の切り換えを指示する本発明の第 2 の実施の形態を説明する。図 5 は、クロック信号の周波数の切り換えを指示するコマンドの説明図である。周波数の切り換えを指示するコマンドは「SYSCLOCK」であり、コマンドに続くパラメータの値で周波数を指定するようになっている。例えば、パラメータとして「0」が指定されると、クロック信号の周波数がデフォルトの値、つまり起動時の周波数へ切り換えられる。パラメータとして「1」が指定されると、クロック信号の周波数が 16MHz に切り換えられ、パラメータとして「2」が指定されると、クロック信号の周波数が 33MHz に切り換えられ、パラメータとして「3」が指定されると、クロック信号の周波数が 66MHz に切り換えられる。

【0046】図 6 は、アプリケーションプログラム側からクロック信号の周波数を切り換えるプログラムの一例を示す図である。同図は C 言語により作成されたプログラムを示しており、system("command") は、"command" が示す文字列を DOS のコマンドとして実行する命令で

ある。

【0047】デフォルトの周波数が16MHzであるとすると、A処理は16MHzのクロック信号により実行される。A処理が終了し、次のB処理でsystem("sysclk 3")が実行されると、そのコマンドで指定されるプログラムが起動される。

【0048】図7は、アプリケーションプログラムのコマンドにより起動される周波数の切り換え処理のフローチャートである。先ず、指定されたパラメータをチェックし(図7、S11)、パラメータの値が0~3の範囲外か否かを判別する(S12)。パラメータの値が範囲外のときには、ステップS24に進み異常終了を示すパラメータを設定して処理を終了する。

【0049】ステップS12で、指定されたパラメータが範囲内と判別されたときには、ステップS13に進みクロック信号の切り換え中は制御権を他の処理に渡さないようにメインCPU12を割り込み禁止の状態にする。そして、パラメータの値が「0」か否か、すなわちデフォルトの周波数への切り換えの指示か否かを判別する(S14)。この判別でパラメータの値が「0」と判別されたときには、ステップS15に進みリアルタイムクロック15のメモリに記憶されているデフォルトのクロック情報(クロック信号の周波数)を読み出す。そして、その読み出した周波数をCPUクロック切替レジスタ27に書き込む(S16)。これにより、クロックジェネレータ13で生成されるクロック信号の周波数が、デフォルトの値(例えば、16MHz)に切り換えられる。

【0050】なお、CPUクロック切替レジスタ27の内容を書き換える際には、予め14MHz設定有効レジスタ29を有効にして、セクタ24で14.318MHzの信号を選択させてクロック信号の周波数を一旦14.318MHzに切り換えた後、CPUクロック切り換えレジスタ27の周波数を書き換え、周波数の切り換えが完了してから、14MHz設定有効レジスタ29を無効にして、セクタ24で変更後の周波数のクロック信号を選択するようにしている。

【0051】ステップS14の判別でパラメータの値が「0」でないと判別されたときには、ステップS17に進みパラメータの値が「1」か否か、すなわち16MHzへの切り換えの指示か否かを判別する。パラメータの値が「1」で16MHzへの切り換えの指示の場合には、ステップS18に進みCPUクロック切替レジスタ27に16MHzの値を書き込む。これにより、クロックジェネレータ13で生成されるクロック信号の周波数が16MHzに切り換えられる。

【0052】ステップS17でパラメータの値が「1」でないと判別された場合には、次のステップS19に進みパラメータの値が「2」か否か。すなわち33MHzへの切り換えの指示か否かを判別する。パラメータの

値が「2」で、33MHzへの切り換えの指示であったときには、ステップS20に進みCPUクロック切替レジスタ27に33MHzの値を書き込む。これにより、クロックジェネレータ13で生成されるクロック信号の周波数が33MHzに切り換えられる。

【0053】ステップS19でパラメータの値が「2」ではないと判別された場合には、パラメータの値が「3」で66MHzへの切り換えが指示された場合であるので、ステップS21に進みCPUクロック切替レジスタ27に66MHzの値を書き込む。これによりクロックジェネレータ13で生成されるクロック信号の周波数が66MHzに切り換えられる。

【0054】以上のようにしてクロック信号の周波数の切り換えが終了したなら、ステップS22に進みメインCPU12に対する割り込みを許可する。これにより周波数の切り換え処理が終了したので、正常終了パラメータを設定して(S23)、周波数切り換え処理を終了し、制御権をDOSに戻す。

【0055】上述した処理によりクロック信号の周波数が切り換えられると、図6の処理Bが66MHzのクロック信号で実行され、B処理の最後でクロック信号の周波数をデフォルト(この場合は16MHz)に切り換えるコマンドを実行することにより、クロック信号の周波数を元の16MHzに戻すことができる。

【0056】この第2の実施の形態によれば、アプリケーションプログラム側からメインCPU12の動作周波数を切り換えることができるので、通常はメインCPU12を低い周波数で動作させて消費電流を少なくして、装置の使用可能時間を長くし、高速の処理速度を要求される処理を実行するときにはメインCPU12を高い周波数で動作させ、処理時間を短縮することができる。

【0057】次に、特定のレジスタに対するアクセスを検出して、クロック信号の周波数を切り換えるようにした本発明の第3の実施の形態を、図8及び図9を参照して説明する。

【0058】図8は、データ転送時の通信IC18に対するアクセスを検出するアクセス検出回路90の回路ブロック図である。ハンディターミナルからホストコンピュータにデータを送信する場合には、通信速度を設定するために、メインCPU12から通信IC18の通信レイト設定レジスタ76に通信速度が書き込まれる。デコード回路91は、メインCPU12から出力されるアドレスデータをデコードして、通信レイト設定レジスタ76のアドレスと一致するアドレスデータを検出したなら、アクセス監視回路92にアドレス一致信号を出力する。アクセス監視回路92は、デコード回路91からアドレス一致信号を受け取ると、割り込み発生回路93に指示して割り込み信号を出力させると共に、内部のカウンタを起動する。

【0059】通信IC18の通信レイト設定レジスタ7

6に通信速度が設定された後、データ転送が開始されると、送信レジスタ79に一定時間毎に送信データが書き込まれるので、デコード回路91でメインCPU12から出力されるアドレスデータを送信レジスタ79のアドレスと比較してアドレスが一致したならアドレス一致信号をアクセス監視回路92に出力する。アクセス監視回路92は、送信レジスタ79のアドレス一致信号を受信する毎にカウンタをリセットする。

【0060】データ送信が終了して送信レジスタ79に対するアクセスが終了すると、デコード回路91から送信レジスタ79のアドレス一致信号が出力されなくなるので、アクセス監視回路92のカウントアップする。アクセス監視回路92は、カウンタがカウントアップすると、割り込み発生回路93に対して割り込み信号の出力を指示する。

【0061】すなわち、割り込み発生回路93からは、データ送信を行う際の通信速度を設定するとき、データ送信が終了したときに割り込み信号が出力されるので、BIOS側でこの割り込み信号を検出したときクロック信号の周波数を切り換えることで、データ転送処理を高速で処理することができる。

【0062】次に、図9は割り込み発生回路93からの割り込み信号により周波数を切り換える場合の処理内容を示すフローチャートである。割り込み信号を検出したなら、プログラマブルインターラプトコントローラ17のレジスタを参照して割り込み要因をチェックする(図9、S31)。割り込み要因がデータ転送に関するものであれば、まず、通信レイト設定レジスタ76に対するアクセスか否かを判別する(S32)。ステップS32で通信レイト設定レジスタ76に対するアクセスと判別された場合には、ステップS33に進み通信レイト設定レジスタ76に記憶されている通信速度を読み出す。そして、その読み出した通信速度が所定値(例えば、38.4KBPS)以下か、それとも所定値を超えているかを判別する(S34)

読み出した通信速度が所定値以下であればステップS35に進み、クロック信号の周波数として33MHzをクロックジェネレータ13のCPUクロック切替レジスタ27に書き込む。これによりクロックジェネレータ13から33MHzのクロック信号がメインCPU12の動作クロックとして出力される。

【0063】他方、ステップS34で通信速度が所定値を超えていると判別されたときには、ステップS36に進み、クロック信号の周波数として66MHzをクロックジェネレータ13のCPUクロック切替レジスタ27に書き込む。これにより、クロックジェネレータ13から66MHzのクロック信号がメインCPU12の動作クロックとして出力される。

【0064】ステップS32で通信レイトレジスタ76に対するアクセスではないと判別された場合には、ステ

ップS37に進みアクセス監視回路92のカウンタがカウントアップしたか否かを判別する。カウンタがカウントアップしている場合には、データ転送処理が終了したものと判断して、ステップS38に進み、リアルタイムクロック15のメモリに記憶されているデフォルトのクロック情報を読み出す。そして、その読み出した周波数をクロックジェネレータ13のCPUクロック切替レジスタ27に書き込む(S39)。これにより、クロックジェネレータ13から通常動作時の周波数のクロック信号がメインCPU12に出力される。

【0065】この第3の実施の形態によれば、データ転送処理等の高速な処理が必要となる処理によるレジスタ等のメモリへのアクセスをハードウェアで検出し、その処理の実行中はCPUのクロック信号の周波数を高くし、あるいは通信速度に応じた周波数に切り換え、処理が終了したなら通常動作時の周波数に切り換えるようにしたので、通常動作時は低い周波数のクロック信号でメインCPU12を動作させて消費電力を少なくして、装置の使用可能な時間を長くし、高速な処理速度が要求される処理を実行するときには高い周波数のクロック信号でメインCPU12を動作させて高速の処理を実現できる。

【0066】上述した実施の形態では、データ転送処理を例にとり説明したが、これに限らず、高速な処理速度が必要となる他の処理にも本発明は適用できる。また、第3の実施の形態において、例えば特定の処理が起動されるときにアクセスされるアドレスを記憶しておいて、そのアドレスに対するアクセスを検出することでその特定の処理の起動を検出し、そのときクロック信号の周波数を切り換えるようにすることもできる。

【0067】さらに、上述した実施の形態は本発明をハンディターミナルに適用した場合について説明したが、これに限らず携帯型パーソナルコンピュータ、携帯型小型端末等のデータ処理装置にも適用できる。

【0068】

【発明の効果】本発明によれば、高速処理が必要なプログラムを実行するときには、CPUの動作クロックの周波数を通常動作時より高くすることで、通常動作時の消費電力を少なくして装置の使用可能時間を長くし、同時に高速な処理を実現できる。また、クロック信号の周波数の切り換えをアプリケーションプログラム側から行えるようにしたので、そのプログラムに適した処理速度でCPUを動作させることができ、消費電力を少なくし、かつ処理時間を短縮できる。さらに、データ通信を行うとき、通信速度に応じた周波数でCPUを動作させることで、消費電力を少なくし、かつ高速なデータ通信を実現できる。

【図面の簡単な説明】

【図1】本発明の実施の形態のハンディターミナルのブロック図である。

【図2】クロックジェネレータ13のブロック図である。

【図3】通信IC18のブロック図である。

【図4】第1の実施の形態の周波数切り換え処理のフローチャートである。

【図5】第2の実施の形態において、クロック信号の周波数の切り換えを指示するコマンドの説明図である。

【図6】第2の実施の形態のアプリケーション側のプログラムの一例を示す図である。

【図7】第2の実施の形態の周波数切り換え処理のフローチャートである。

【図8】通信IC18へのアクセスを検出するアクセス\*

\* 監視回路の回路ブロック図である。

【図9】第3の実施の形態の周波数切り換え処理のフローチャートである。

【符号の説明】

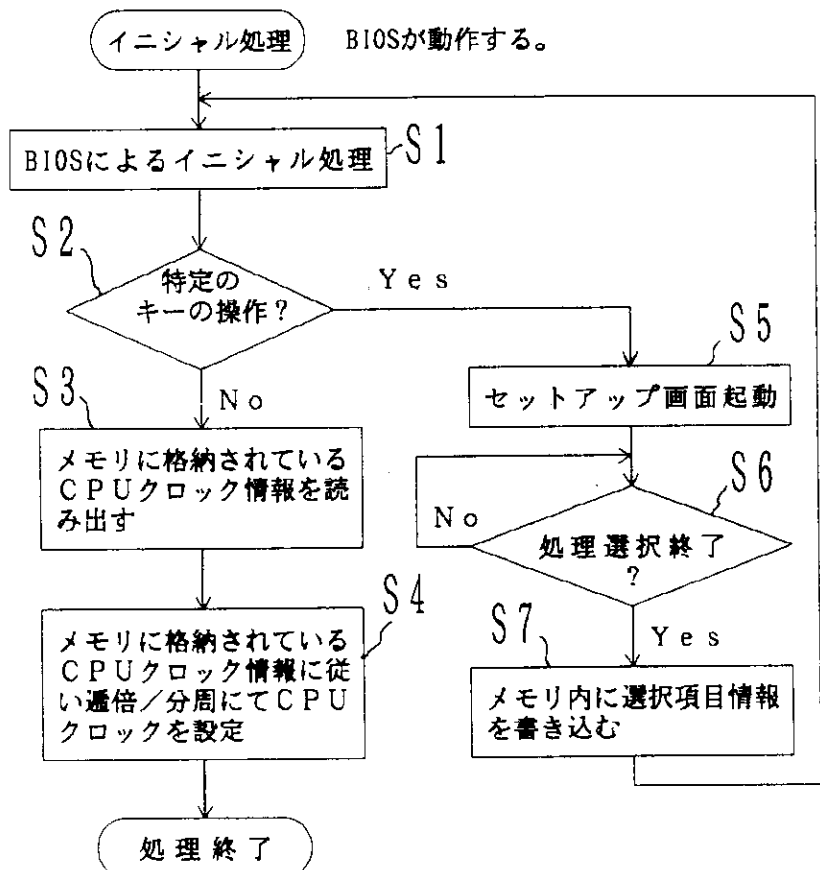
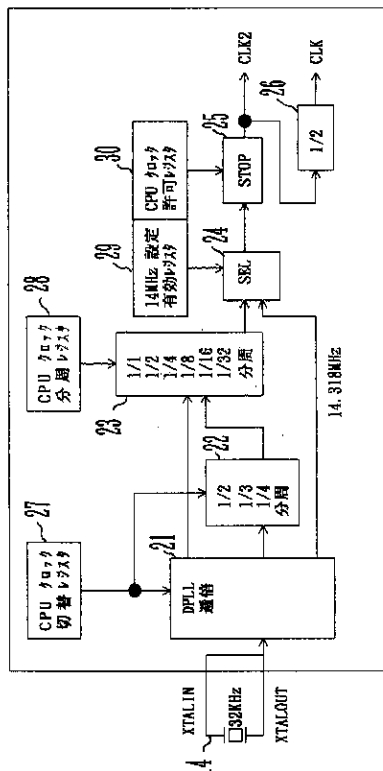
|    |               |
|----|---------------|
| 11 | チップセット        |
| 12 | メインCPU        |
| 13 | クロックジェネレータ    |
| 18 | 通信IC          |
| 27 | CPUクロック切替レジスタ |
| 76 | 通信レイト設定レジスタ   |
| 91 | デコード回路        |
| 92 | アクセス監視回路      |

【図2】

【図4】

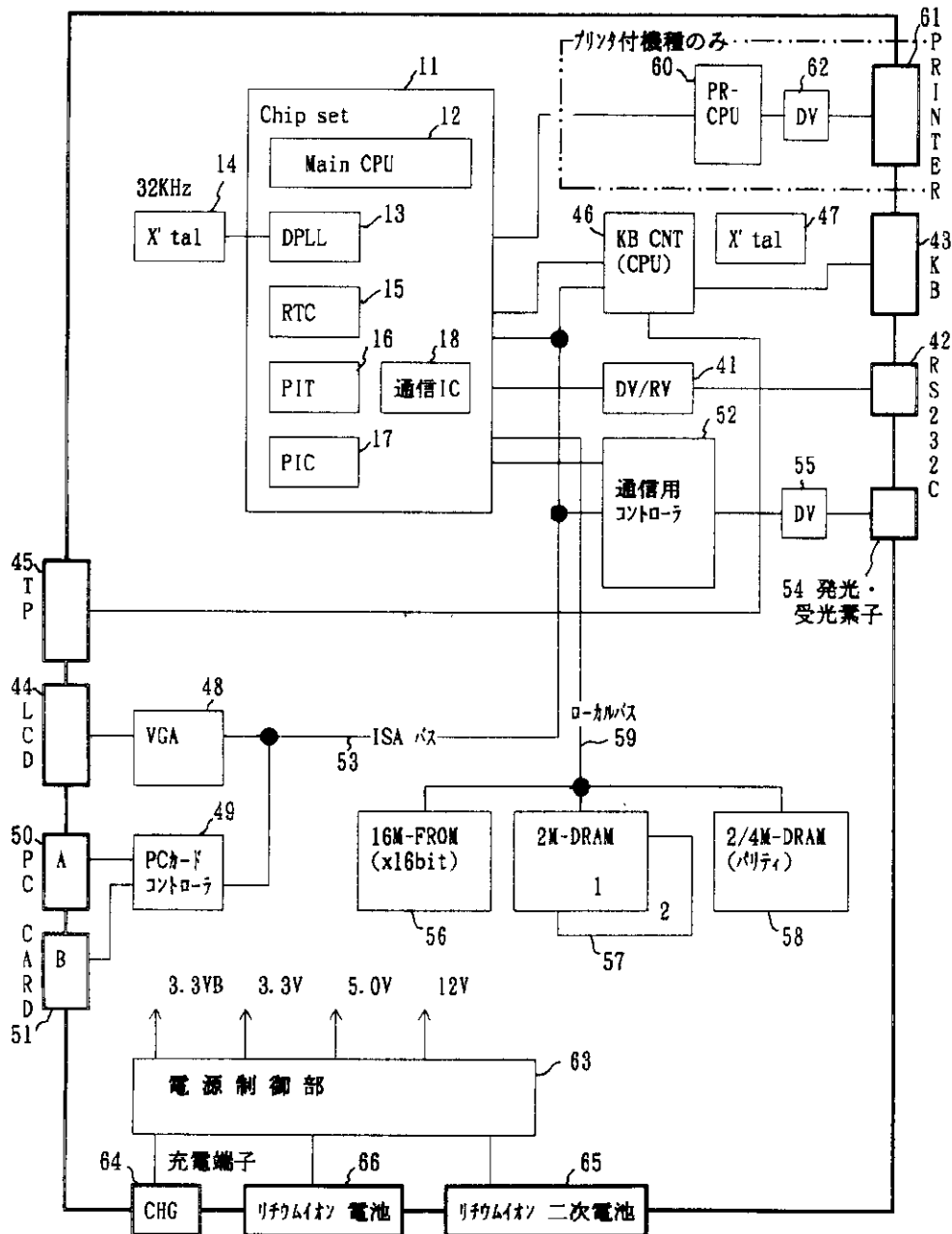
クロックジェネレータ13のブロック図

第1の実施の形態の周波数切り換え処理のフローチャート



【図1】

## ハンドヘルドターミナルのブロック図



【図5】

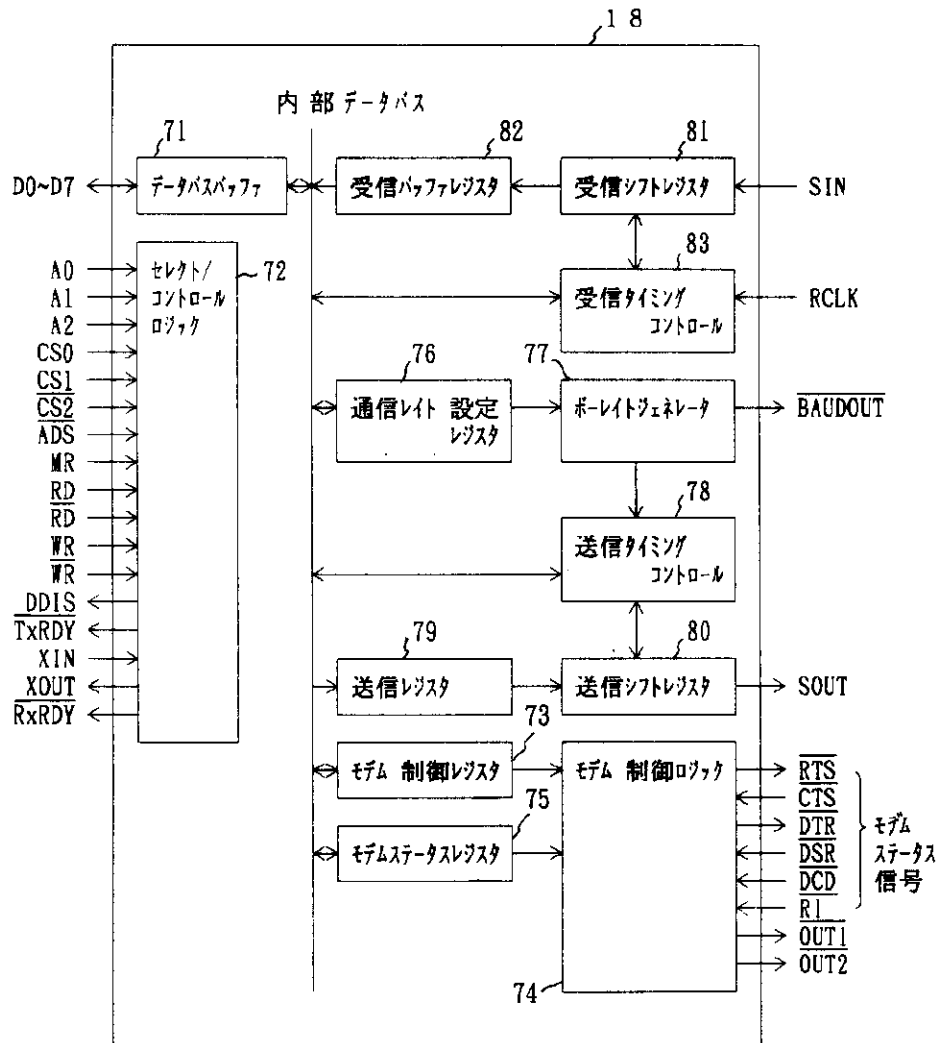
第2の実施の形態において、クロック信号の  
周波数の切り換えを指示するコマンドの説明図

SYSCLK [パラメータ]

- 【指定パラメータ】 0: CPU クロックをデフォルトにする (立ち上がり時のCPU クロック)  
1: CPU クロックを16MHz へ切り替える。  
2: CPU クロックを33MHz へ切り替える。  
3: CPU クロックを66MHz へ切り替える。

【図3】

## 通信 IC 18 のブロック図



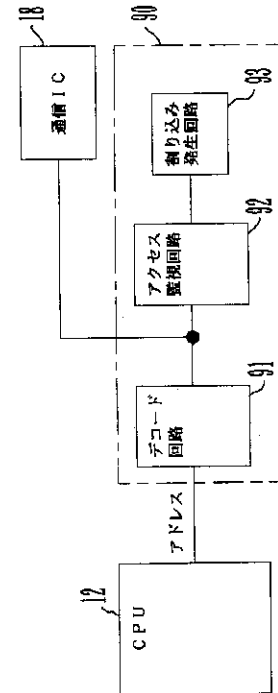
【図6】

第2の実施形態の  
アプリケーション側のプログラムの一例を示す図

【書式】 ret=system("command"); ※引数command が示す文字列をDOS のコマンドとして実行する。  
< 概要 >  
・ A 処理はCPU クロック16MHz で処理  
・ system('sysclk 3'); ※CPUクロックを16MHz より86MHz に切り替えるコマンド実行  
・ B 処理はCPU クロック86MHz で処理  
・ system('sysclk 0'); ※CPUクロックをデフォルト(この場合は16MHz)切り替えるコマンド実行  
・ C 処理はCPU クロック16MHz で処理

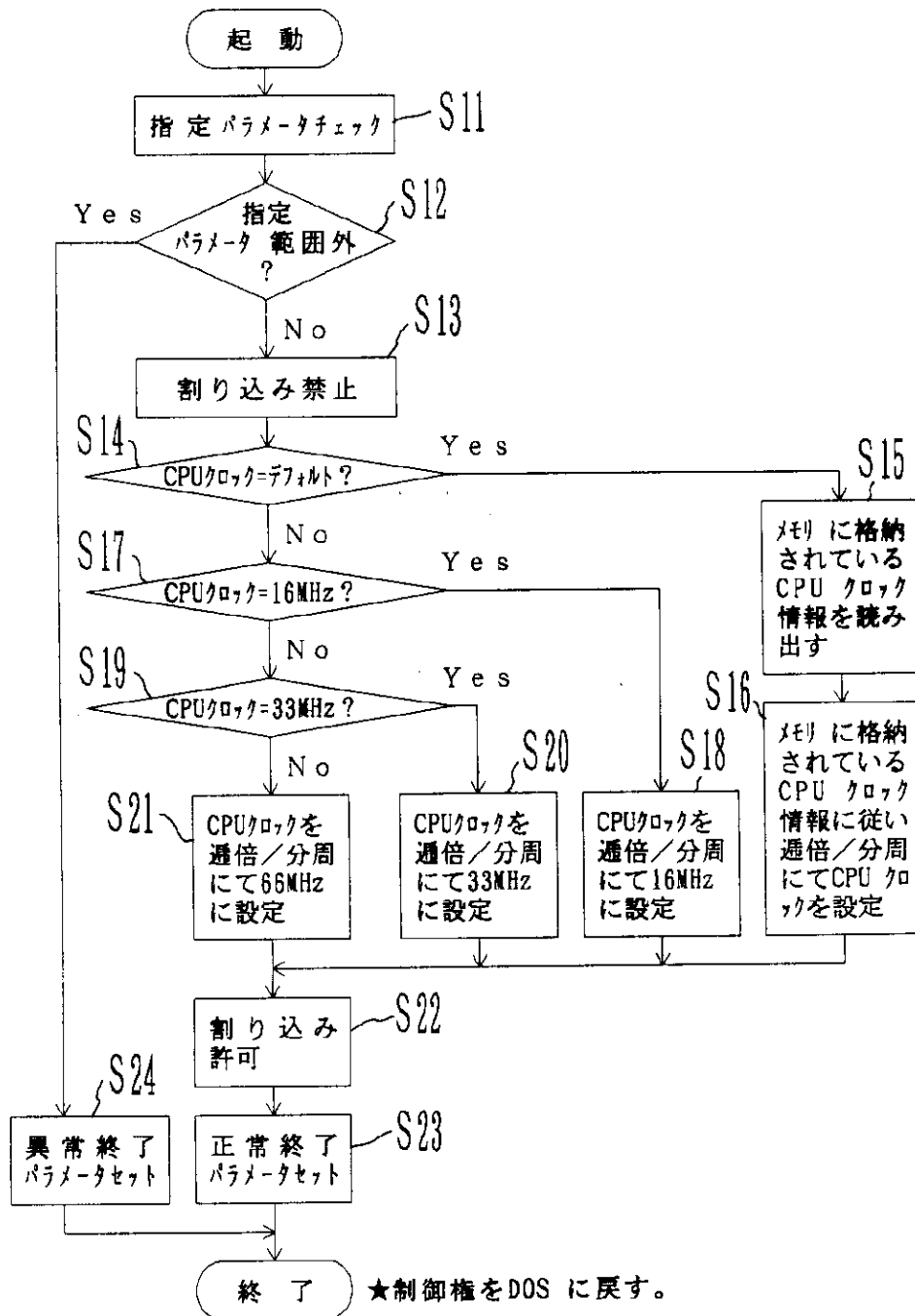
【図8】

通信IC 18へのアクセスを検出する  
アクセス検出回路 90 の回路ブロック図



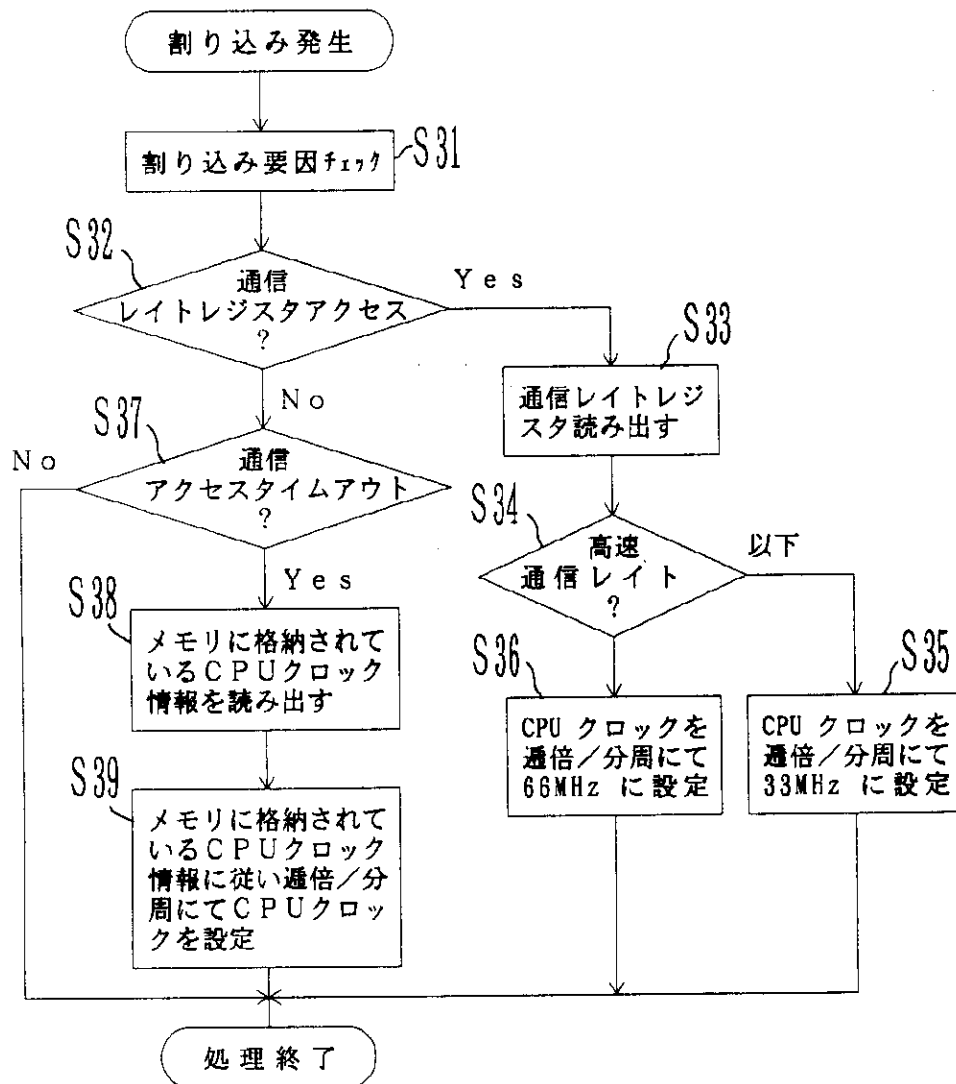
【図 7】

## 第 2 の実施の形態の周波数切り換処理のフローチャート



【図 9】

## 第 3 の実施の形態の周波数切り換え処理のフローチャート



フロントページの続き

(56)参考文献 特開 平 2-3809 (J P, A)  
 特開 平 8-36445 (J P, A)  
 特開 平 9-297688 (J P, A)

(58)調査した分野(Int.Cl.<sup>7</sup>, D B 名)  
 G06F 1/08