(12) UK Patent Application (19) GB (11) 2 243 701 (13) A

(43) Date of A publication 06.11.1991

(21) Application No 9105482.5

(22) Date of filing 15.03.1991

(30) Priority data
(31) 9005916 (32) 16.03.1990 (33) GB

(51) INT CL[5]
G06F 11/00

(52) UK CL (Edition K)
G4A AAP AFMF

(56) Documents cited
EP 0217668 A2 EP 0067875 A1 US 4688169 A
US 4462076 A
Commodore 64 Programmer's Reference Guide, 1983, CBM Inc & Howard W Sams & Co, pp 262, 263, 269.

(58) Field of search
UK CL (Edition K) G4A AAP AFMF
INT CL[5] G06F 11/00

(71) Applicant
Amstrad Public Limited Company

(Incorporated in the United Kingdom)

Brentwood House, 169 Kings Road, Brentwood, Essex, CM14 4EF, United Kingdom

(72) Inventor
Stephen David Gane

(74) Agent and/or Address for Service
Reddie & Grose
16 Theobalds Road, London, WC1X 8PL,
United Kingdom

(54) Controlling access to computer system features

(57) A computer system is designed for use with a removable and insertable program source, such as a ROM cartridge 9 or floppy disk 12, and includes a CPU 1 for executing stored program instructions either directly from the ROM cartridge or after being read into a RAM memory 3. A video/sound generating chip 2 provides basic features open to all programs and enhanced features open only to selected programs. Access to the enhanced features is controlled by comparing a sequence of bytes outputted by the CPU in response to the program instructions with a reference sequence provided by a pseudo-random sequence generator (Figure 2), and enabling the enhanced features in dependence upon the result of the comparison.

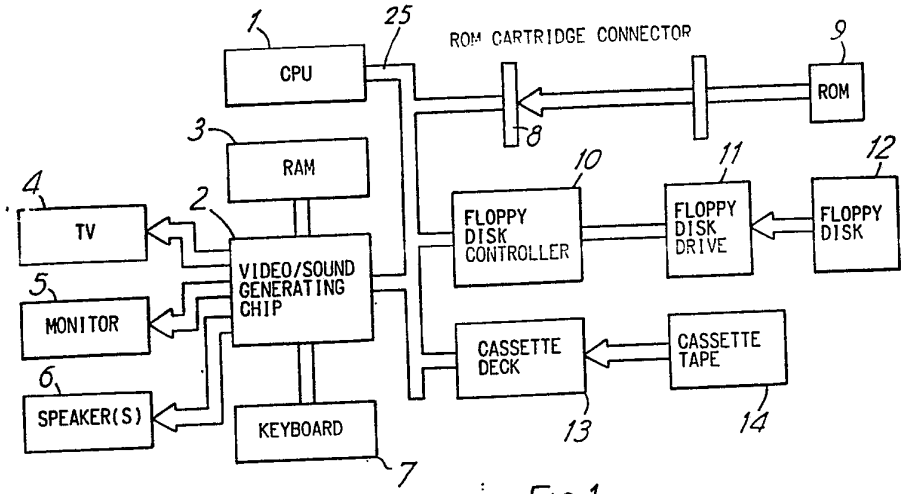

FIG.1

GB 2 243 701 A

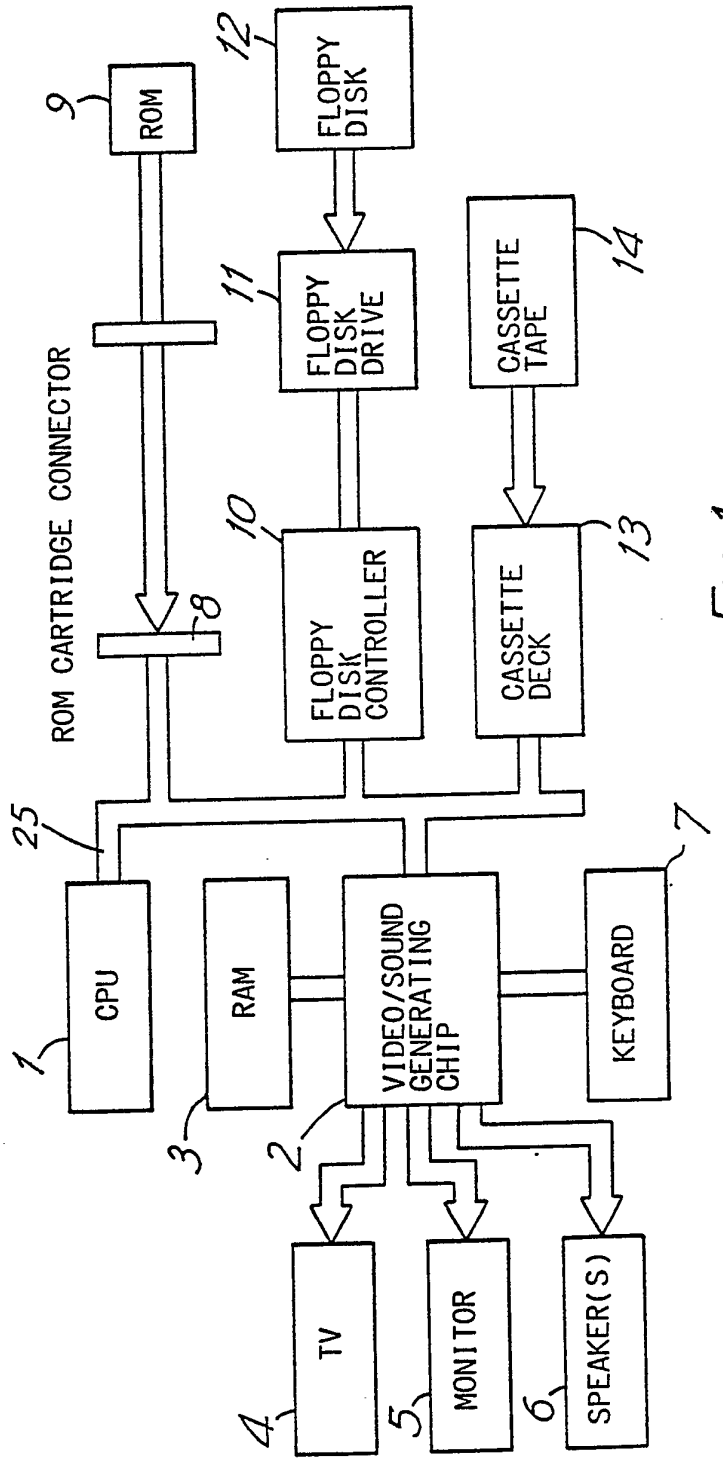
ROM
9
FLOPPY DISK
12
FLOPPY DISK DRIVE
11
CASSETTE TAPE
14
ROM CARTRIDGE CONNECTOR
8
FLOPPY DISK CONTROLLER
10
CASSETTE DECK
13
25
CPU
1
RAM
3
VIDEO/SOUND GENERATING CHIP
2
KEYBOARD
7
TV
4
MONITOR
5
SPEAKER(S)
6

FIG.1

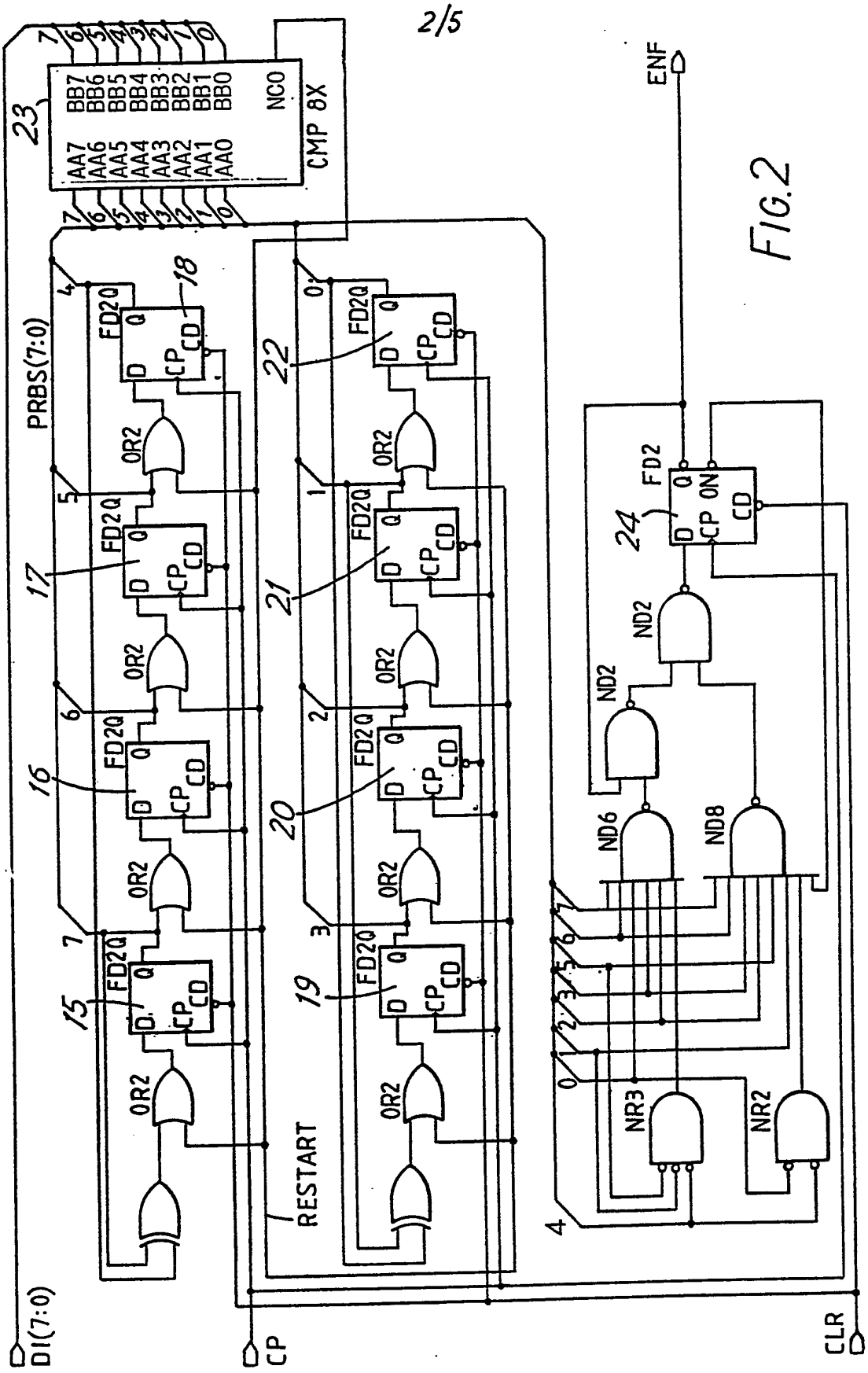
23
CMP 8X
BB7
BB6
BB5
BB4
BB3
BB2
BB1
BB0
NCO
AA7
AA6
AA5
AA4
AA3
AA2
AA1
AA0
PRBS(7:0)
DI(7:0)
CP
CLR
ENF
RESTART
FD2Q
FD2
OR2
ND2
ND6
ND8
NR3
NR2
15
16
17
18
19
20
21
22
24

FIG.2

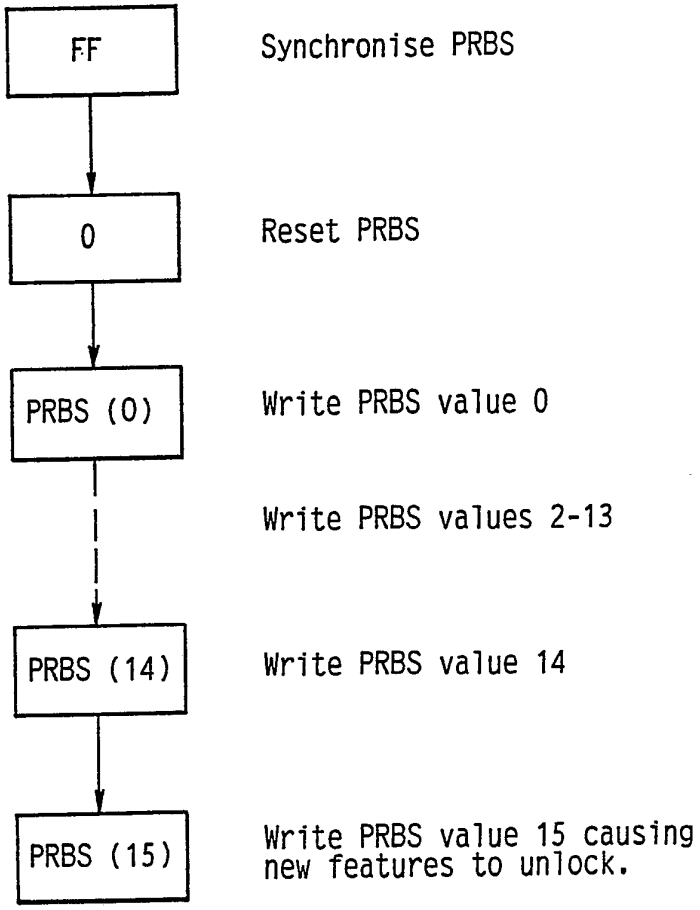
FF
Synchronise PRBS
0
Reset PRBS
PRBS (0)
Write PRBS value 0
Write PRBS values 2-13
PRBS (14)
Write PRBS value 14
PRBS (15)
Write PRBS value 15 causing new features to unlock.

FIG.3

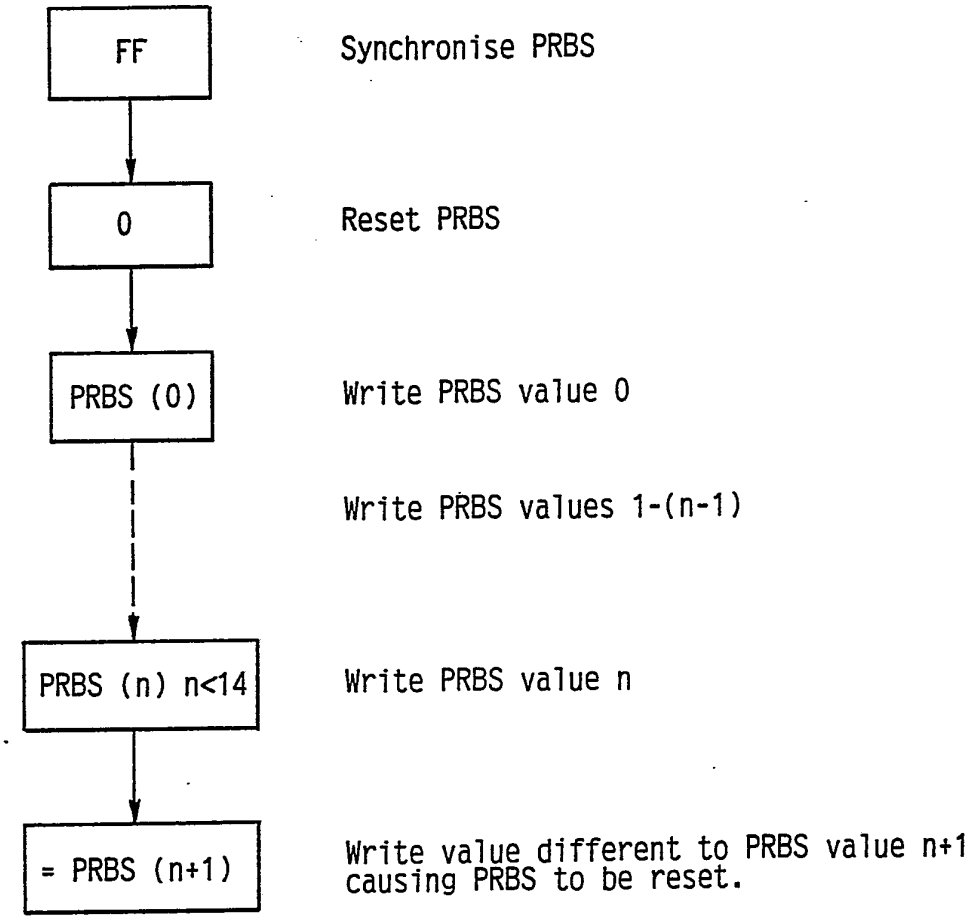
FF
Synchronise PRBS
0
Reset PRBS
PRBS (0)
Write PRBS value 0
Write PRBS values 1-(n-1)
PRBS (n) n<14
Write PRBS value n
= PRBS (n+1)
Write value different to PRBS value n+1 causing PRBS to be reset.

FIG.4

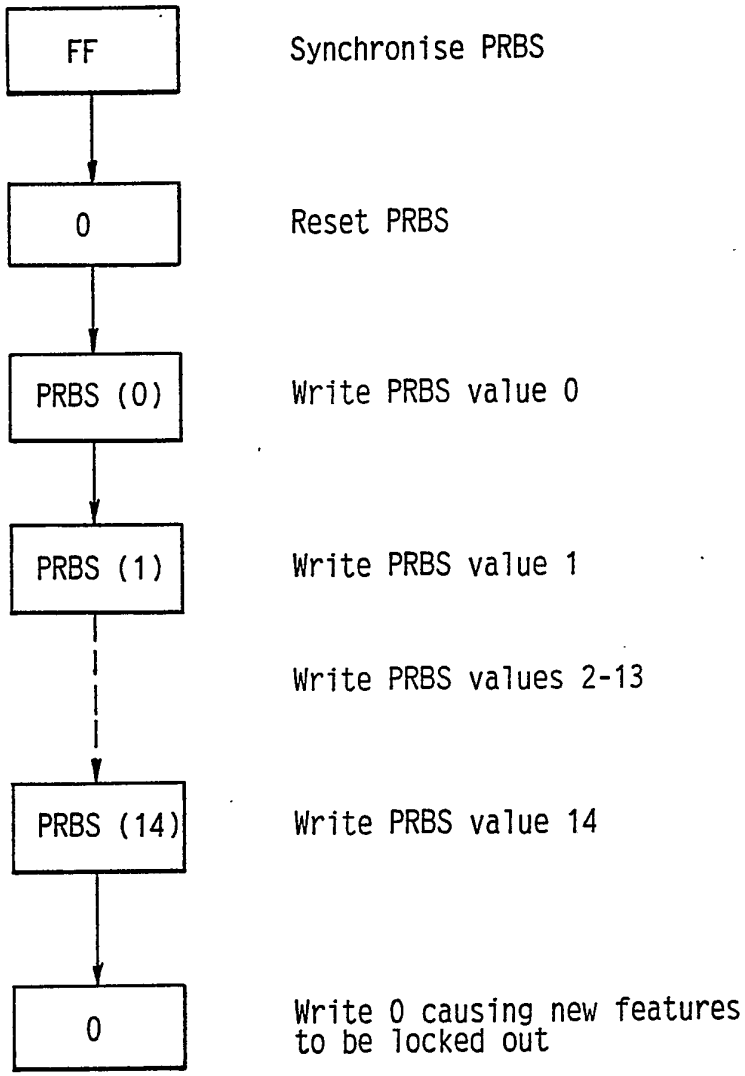
FF
Synchronise PRBS
0
Reset PRBS
PRBS (0)
Write PRBS value 0
PRBS (1)
Write PRBS value 1
Write PRBS values 2-13
PRBS (14)
Write PRBS value 14
0
Write 0 causing new features to be locked out

FIG.5

COMPUTER SYSTEM

This invention relates to computer systems and in particular to a computer which receives its software program from a removable program source such as a ROM cartridge, floppy disk, or cassette tape.

It is common practice for a computer to be designed and manufactured by a company specialising in hardware, and for software for this computer to be written and sold by many different specialist software companies. This software may cover many diverse applications, may be provided by a wide range of competing vendors, and may be available in different media formats, including cassette tape, floppy disk and ROM cartridge. In such circumstances the hardware vendor, manufacturing the computer, may be unable to regulate or control the quality or suitability of such software. He will however, make every effort to ensure that any design or manufacturing changes to the computer will not make it incompatible with software written for previous generations of this same computer.

It is often possible for the manufacturer to achieve desirable reductions in manufacturing cost or increases in product reliability by incorporating the latest advances in technology whilst not affecting compatibility with previous generations of software. Such changes, however, cannot affect the features of the computer such as the number of display modes or the music and sound generation facilities.

However, the computer manufacturer may wish to upgrade his computer by adding enhanced features, such as extra display modes or new sound and music generating facilities, whilst still maintaining the previously available features and also maintaining compatibility with previous generations of software.

In such a case, the computer designer must recognise that his computer must operate correctly, firstly, with enhanced software that has been written with knowledge of the new enhanced features and designed specifically to exploit them and secondly, with old software that was written without knowledge of the new enhanced features. In particular, the computer designer must ensure that

old software does not accidentally activate the new features. The computer requires a method of ensuring that the new enhanced features are only available to new software which has been written with a full understanding of how to correctly use these new features.

Previously, computer manufacturers have sometimes published specifications for their products which indicate that certain areas of the I/O and memory map are reserved for future expansion or enhancement. However, there are cases where such reservations are not made.

Also, previously, some computer manufacturers wishing to enhance their products have sought to control the quality or suitability of third party software by, for example, publishing lists of suitable software. Such a method is only applicable when the number of software titles to be qualified is known and manageably small.

Also, previously, some computer manufacturers have insisted that software to be used with the new, enhanced features must be fitted with a hardware key, otherwise known as a "dongle". Such a method requires the expense of firstly, qualifying the software to check its suitability for use with the new computer, and secondly, the expense of fitting a hardware key or "dongle" to each item of software sold.

In United States Patent 4 462 076 is proposed a system which authenticates a removable cartridge, by requiring the CPU to read and check the contents of selected locations in the cartridge. A similar approach is used in the Commodore 64 computer system, see Commodore 64 Programmer's Reference Guide, 1982, Commodore Business Machines Inc., page 263. Such proposals require the computer to have special firmware (e.g. ROM software) to enable it to look for the authentication code.

European Patent Specification 67 875 and United States Patent 4 688 169 both describe systems in which a machine-specific code or "signature" is written on each copy of a program so that it will only run on one specific machine identified by that code or signature. Again it is necessary for the microprocessor to check that particular location of the program tape.

In another instance, European Patent Specifications 206 704

and 217 668 describe a ROM cartridge which contains its own microprocessor so that the output of that microprocessor can be compared with that of a corresponding device in the main computer to authenticate the cartridge. However, having a microprocessor in the ROM cartridge is both expensive and complex.

According to the invention there is provided a computer system comprising a removable and insertable program source, memory means for storing program instructions, a CPU for executing stored program instructions, feature means containing features accessed by instructions from the CPU, and access enabling means for enabling access to at least selected ones of the features, the access enabling means comprising means for comparing a sequence of bytes outputted by the CPU in response to the program instructions with a reference sequence of bytes, and for enabling the selected features in the feature means in dependence upon the result of the comparison.

In the present invention the program itself causes the CPU to output a sequence of bytes. Only when that happens, and when the thus-generated sequence agrees with the reference sequence, are the additional enhanced features made available to the program.

The invention also provides a program source, such as a ROM cartridge etc., for use in the computer system, and a method of selectively enabling functions in a computer system.

The present invention has the advantage of being simple to implement, requiring only that the processor, under software control, writes a predetermined sequence of values to a specified port location to make the enhanced features available. No special built-in firmware, such as might be necessary in computers which look for "security codes" at specific addresses within the program source, is required. No separate microprocessor is required in the program source.

The invention will now be described in more detail, by way of example, with reference to the accompanying drawings, in which:

Figure 1 is a block diagram of a computer system embodying the invention;

Figure 2 is a block circuit diagram illustrating lock circuitry within the video/sound generating chip in the system of Figure 1;

Figure 3 is a flow chart illustrating the steps in a successful attempt to unlock the new enhanced features in the computer;

Figure 4 is a flow chart illustrating the steps in an unsuccessful attempt to unlock the new features; and

Figure 5 is a flow chart illustrating the steps in a successful attempt to lock the new features.

Figure 1 shows a computer having a CPU 1 (central processing unit) which executes instructions from the applications software currently loaded. This software is loaded through a bus 25 into RAM (random access memory) 3 and may originate from a ROM (read only memory) cartridge 9 plugged onto the CPU bus via a ROM cartridge connector 8, or a floppy disk 12, the contents of which are read via a floppy disk drive 11 and floppy disk controller 10; or a cassette tape 14, the contents of which are read via a cassette deck 13. Although particularly suitable for use with a ROM cartridge, this invention is not limited to any particular method of loading software. When the program is on disk or tape it is downloaded to RAM. This mode of operation is also possible when the program is a ROM cartridge, although it is preferred that the CPU will execute instructions one by one directly from the cartridge. In the following description it will be assumed that the program is downloaded to RAM, this being applicable to all three types of program source.

The CPU executes instructions during the normal course of running software which read and write the contents of RAM memory 3; read the status of a keyboard 7, and read and write internal registers and memory in a video/sound generating chip 2 that control the video and sound features of the computer. These internal registers and memory include some concerned with controlling the old or basic features and also extra ones concerned with the new or enhanced features. The basic features are open to all programs but the enhanced features are open only to those programs written with these enhanced features specifically in mind. The chip 2 also provides the communication path for the keyboard 7 and RAM 3 within the bus 25.

Also included in the video/sound generating chip 2 is a lock

circuit that prevents old software from accidentally activating the new features. This lock has two states. When locked the new enhanced features are not accessible to the CPU, when unlocked, these features are available.

Figure 2 shows the details of the lock circuitry inside the video/sound generating chip. This comprises a pseudo-random-binary-sequence (PRBS) generator consisting of flip-flops 15 to 22 and their associated exclusive-OR gates. The flip-flops 15 to 22 in practice operate as two PRBS generators, each four bits wide, to give an overall eight-bit wide PRBS. The construction of a PRBS generator is well known and basically comprises a shift register with a number of stages, four as shown, the outputs of selected stages being combined and fed back as the input to the first stage. The PRBS generator can take sixteen different states including the all-zero states. Each state has a unique 8-bit value appearing on bus PRBS (7:0). The sixteen states are referred to as PRBS (0) to PRBS (15). As is well known with PRBS generators, where n _ is the number of stages in the register, the PRBS generator is capable of generating a pseudo-random sequence of 0's and 1's. The sequence repeats with a length which, provided the connections to the gates are correctly chosen, has a maximum of m _ = 2 n - 1.

The state of the lock is determined by the flip-flop 24 with output signal ENF. When ENF is zero the enhanced features are disabled (locked). When ENF is one, the enhanced features are enabled and available to the CPU (unlocked).

The PRBS generator advances its state whenever the CPU excecutes an output instruction to Port A, this being a port within the video/sound chip 2 by which the lock circuit interfaces to the bus 25, such that an output instruction to Port A is an output instruction to the lock. Such an output instruction results in a pulse on signal CP which clocks the eight flip-flops 15 to 22 comprising the PRBS generator.

During the rising (active) edge of this pulse the CPU data bus appears on bus DI (7:0) and is compared with the PRBS state PRBS (7:0) in an 8-bit comparator 23. Thus, for example, if PRBS (7:0) equals 3F (hex) and the CPU executes an instruction outputting data value 3F (hex) to port A, the PRBS generator will advance a state with the comparator 23 output indicating a true comparison.

On each positive edge of the clock pulse (CP), the PRBS generator will either:

a) advance through its sequence, if the data on DI (7:0) matches the PRBS state PRBS (7:0), orb) set to all 1's, i.e. restart the sequence from the beginning if the comparator indicates that DI (7:0) does not match PRBS (7:0).

On each positive clock edge (CP), the output flip-flop 24 will either:

a) go to a 0 state (locked) if the PRBS is at state PRBS 14, or

b) go to a 1 state (unlocked) if the PRBS is at state PRBS 15, or

c) hold its previous state.

The CPU output is of course dependent upon the program loaded in it. This program is derived from the program source, namely the ROM 9, floppy disk drive 12 or casette tape 14. Thus the computer can tell by the comparison described above whether the software on the ROM, floppy disk or tape has been written with the enhanced features or not. Only if it has been written with the enhanced features in mind are these features unlocked so as to be made available to the software. Thus software which was written before the advent of the enhanced features can not inadvertently invoke them or corrupt them.

Whenever it is necessary to operate the lock and the state of the PRBS generator is unknown it must be synchronised by first writing all 1's and then writing all 0's. This will cause the PRBS to reset.

Figures 3, 4 and 5 show sequences of CPU output data values to activate the lock. These figures are self-explanatory and are therefore not here described in detail. Figure 3 is a flow chart showing the steps involved in a successful attempt to unlock the enhanced features. Figure 4 is a flow chart showing the steps involved in an unsuccessful attempt to lock or unlock the enhanced features. Figure 5 is a flow chart showing the steps involved in a successful attempt to lock the new features. The boxes shown contain hexadecimal values to be written to Port A. Also PRBS (0-15) represent the PRBS states in order.

The system is thus seen to comprise two parts. The first

is a hardware lock mechanism, that compares data values written by the CPU to Port A with the state of a PRBS generator; and the second a software routine to send the correct sequence of data values, PRBS (0-15) to Port A. This sequence of values is deliberately chosen to be so obscure that the probability of old software accidentally activating the key is so small as to be negligible. This software may be implemented in many different forms; e.g. it may contain a series of output instructions each with an associated data operand corresponding to the PRBS state; it may, alternatively, comprise a loop with an associated look-up table.

The system described does not require detailed control by the computer manufacturer of the software to be run on it, and yet does not require the use of a dongle or the like. The invention can be used where no reservation of areas of the I/O and memory map has been made.

The invention is not limited to any particular software technique or program structure, but encompasses any software that can output the correct sequence of data values. Further, it is not restricted to any particular media for storing the program and in particular, encompasses ROM cartridge, floppy disk and cassette.

CLAIMS

1. A computer system comprising:

a removable and insertable program source;

memory means for storing program instructions;

a CPU for executing stored program instructions;

feature means containing features accessed by instructions from the CPU; and

access enabling means for enabling access to at least selected ones of the features;

the access enabling means comprising means for comparing a sequence of bytes outputted by the CPU in response to the program instructions with a reference sequence of bytes, and for enabling the selected features in the feature means in dependence upon the result of the comparison.

2. A computer system according to claim 1, in which the program source comprises a ROM cartridge.

3. A computer system according to claim 2, in which the memory means is at least in part comprised by the ROM cartridge and the CPU directly executes instructions held in the ROM cartridge.

4. A computer system according to claim 1, in which the program source comprises a floppy disk.

5. A computer system according to claim 1, in which the program source comprises a cassette tape.

6. A computer system according to claim 2, 4 or 5, in which the memory means comprises a memory accessible to the CPU and not within the removable program source, and further comprising loading means for loading a program in the progam source into the memory for execution by the CPU.

7. A computer system according to any preceding claim, in which the access enabling means comprises a pseudo-random binary sequence generator to generate the reference sequence of bytes.

8. A removable program source for use in a system in accordance with any preceding claim and comprising means for causing the CPU to execute program instructions such as to output a sequence of bytes for comparison in the access enabling means with the said reference sequence.

9. A method of selectively enabling functions in a computer system having a CPU, the method comprising the steps of connecting a removable and insertable program source to the computer system, executing instructions stored in the CPU such as to output a sequence of bytes, comparing the thus-outputted sequence of bytes with a reference sequence of bytes, and selectively enabling access by the CPU to selected operating features in dependence upon the result of the comparison.

10. A method according to claim 9, in which the reference sequence comprises a pseudo-random binary sequence.

Published 1991 at The Patent Office, Concept House, Cardiff Road, Newport, Gwent.NP9 1RH. Further copies may be obtained from Sales Branch, Unit 6, Nine Mile Point, Cwmfelinfach, Cross Keys, Newport, NP1 7HZ. Printed by Multiplex techniques ltd, St Mary Cray, Kent.