

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
28 November 2002 (28.11.2002)

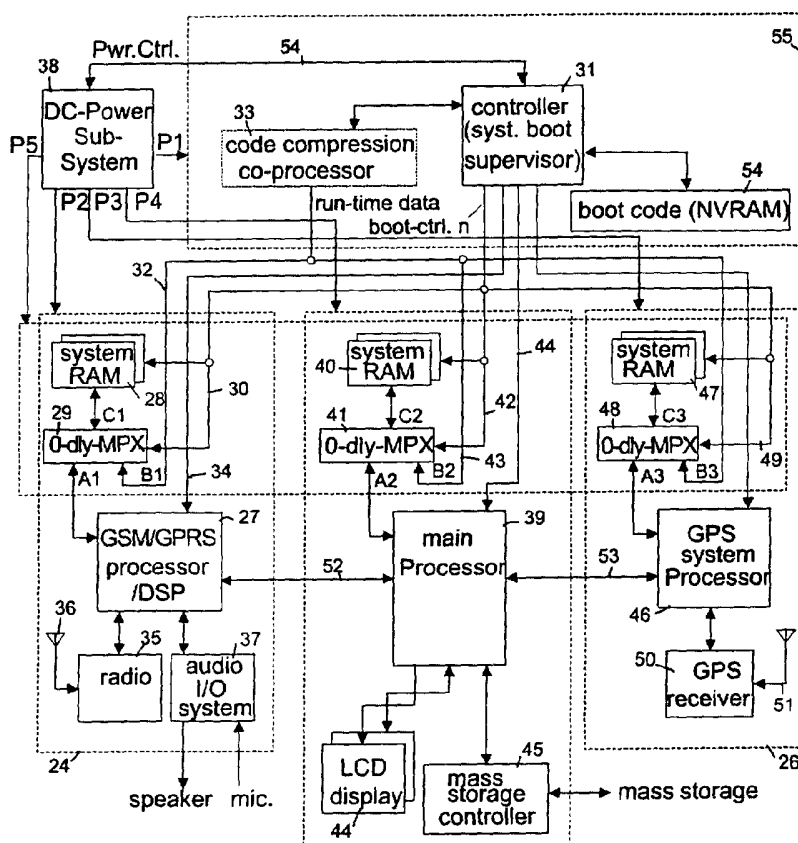
PCT

(10) International Publication Number
WO 02/095581 A2

- (51) International Patent Classification⁷: **G06F 9/445** (74) Agent: **KLEIN, Hans-Jörg**; IBM Deutschland GmbH, Intellectual Property, Pascalstr. 100, 70548 Stuttgart (DE).
- (21) International Application Number: PCT/EP02/04480
- (22) International Filing Date: 24 April 2002 (24.04.2002) (81) Designated States (*national*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZM, ZW.
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
01112316.3 19 May 2001 (19.05.2001) EP
- (71) Applicant: **INTERNATIONAL BUSINESS MACHINES CORPORATION** [US/US]; New Orchard Road, Armonk, NY 10504 (US).
- (71) Applicant (*for LU only*): **IBM DEUTSCHLAND GMBH** [DE/DE]; Pascalstrasse 100, 70569 Stuttgart (DE).
- (72) Inventor: **STAIGER, Dieter E.**; Falkenstrasse 28, 71093 Weil im Schönbuch (DE).
- (84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: ELECTRONIC SYSTEM AND METHOD FOR BOOTING OF AN ELECTRONIC SYSTEM



(57) Abstract: An electronic system is provided having at least first and second sub-systems (24, 25, 26) and a memory (54) for storage of boot-images of the respective sub-systems. A controller 31 loads the boot-images stored in the memory (54) into the respective sub-systems (24, 25, 26) in sequential order via multiplexers (29, 41 and 48). This allows to utilize an inexpensive memory device for the memory (54) and to reduce the overall code size.



Published:

— without international search report and to be republished
upon receipt of that 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.

D E S C R I P T I O N

**ELECTRONIC SYSTEM AND METHOD FOR BOOTING OF AN
ELECTRONIC SYSTEM**Field of the invention

The present invention relates to the field of electronic systems and operation of electronic systems and more particularly to booting of electronic systems and bringing electronic systems into a sleep-mode.

Background and related art

From US Pat.No. 5,991,544 a process and apparatus for managing a software load image is known. The software load management process is used in a computer node operating in a distributed computing environment and is used to permit the computer to initialize its software load image when the computer is first booted. After the computer is turned on, the operating system is initialized. Following this, an application initialization module that is part of the loader, proceeds in registering the software entities, such as any software sub-systems as mail domains, etc. in the software load and starting their execution. The registration process involves searching a symbol table for a registration trigger that is in the form of a certain registration procedure name. Each software sub-system includes a registration procedure, and the name of that procedure is placed in the symbol table. The name of the registration procedure includes a generic component to enable the application initialization module to identify the entry from other entries in the symbol table.

As opposed to distributed computing environments the sub-systems of embedded electronic systems, such as so called electronic control units (ECU), are booted autonomously

- 2 -

without a loader system. Manufactures of embedded systems are experiencing a continuously increasing demand on performance and for a variety of functionalities for embedded devices, such as devices used in industrial manufacturing tools, embedded handheld computers, internet access appliances and in ECUs of modern automobiles.

Directly related and in synchronism with the growth in requirements, the size as well as the performance demand for the booting characteristics of the sub-systems of such devices is increasing dramatically. As opposed to most typically used boot storage representations realized by on-processor-chip ROM or FLASH memory, the most modern embedded systems are utilizing standard nonvolatile memory modules - for two reasons:

(i) Processor modules featuring the extended size of nonvolatile storage embedded on the chip are not available as standard devices and the cost per bit is significantly higher compared to state of the art nonvolatile memory modules

(ii) The significant increase in volume of code stored on the boot devices is driving the system developers to migrate from ROM memories to using re-programmable FLASH memories. FLASH memories allow simplified software version upgrade on one hand, and on the other hand allow to download new software, applications and resulting services on demand - a fact becoming gaining importance in the e-business driven world. For these reasons FLASH memory is becoming a major cost factor for embedded systems.

Fig. 1 shows a block diagram of a prior art smart phone 1. The smart phone 1 has a cell phone sub-system 2, a personal digital assistant (PDA) sub-system 3 and a global positioning system (GPS) sub-system 4.

- 3 -

The cell phone sub-system 2 has a cell phone engine 5 which is based on a digital signal processing unit. The cell phone engine 5 can work in multiple modes of operations such as in a GSM, GPRS and / or UMTS - mode. The cell phone engine 5 is coupled to FLASH memory 6 and to RAM 7.

The FLASH memory 6 is a nonvolatile memory device and contains the software code including the operating system for the cell phone sub-system 2. When the smart phone 1 is switched on the cell phone sub-system 2 is booted out of the FLASH memory 6. Further the major programs and applications of the cell phone sub-system 2 are executed directly out of the FLASH memory 6. The RAM 7 mainly serves to store variables and data generated and to be altered throughout the system runtime. Therefore the RAM 7 typically is a volatile storage device, such as a SDRAM.

Likewise the PDA sub-system 3 has a PDA system processor 8 which is coupled to FLASH memory 9 and RAM 10. Further the PDA system processor 8 is coupled to human-machine interface 11, which comprises a touch screen and to mass storage 12 which can be a smart media, such as a FLASH card, or an IBM micro drive unit.

Further the GPS sub-system 4 has a GPS receiver 13 which is coupled to FLASH memory 14 and RAM 15.

As explained with respect to the cell phone sub-system 2 the FLASH memories 9 and 14 serve for storing the software including the operation system of the PDA sub-system 3 and the GPS sub-system 4, respectively. As a consequence the operation of the cell phone sub-system 2, the PDA sub-system 3 and the GPS sub-system 4 is autonomous and independent of each other. A data request from one sub-system to another sub-system is handled by the receiving sub-system as an external request. Such requests and corresponding data can be interchanged via busses 16 and 17.

- 4 -

A disadvantage of the prior art smart phone 1 is that the FLASH memories 6, 9 and 14 need to be of the NOR-type for storage of the executable code. Such NOR-FLASH memories are a major cost factor for such an electronic system. The usage of inexpensive FLASH memories of the NAND-type is not possible in prior art electronic devices of the type shown in Fig. 1 as such devices have a slow access time in the random access mode and further more, the exceptionally low cost FLASH memories used e.g. in today's high volume 'still camera' market cannot be applied since this type of FLASH is known to have an high memory error making it in-acceptable to be used as an code execution memory or device boot memory.

It is therefore an object of the invention to provide an improved electronic system and an improved method for booting.

The solution of this and other objects of the invention are provided basically by applying the features of the respective independent claims. Preferred embodiments of the invention are given in the dependent claims.

The invention allows to store the boot-images of the individual sub-systems of an electronic system in an inexpensive memory device. For starting system operation the individual boot-images are loaded to the sub-system, such that the normal system operation can immediately start without having to start separate boot-programs. This allows to reduce the overall code size and therefore the physical size of the electronic system, which is of particular importance for portable and handheld devices.

The invention is particularly beneficial for applications in so called loosely coupled multi processing architectures of the kind as depicted in Fig. 1. In this case the boot-images are loaded into individual storage sub-system before the start of the normal system operation. Usage of the invention for

- 5 -

loosely coupled multi processing architecture results in a significant cost advantage and increase of efficiency.

Further the invention allows to save a substantial amount of power, especially in the so called sleep-mode. In particular the invention allows to partly or completely disable or stop the operation of the processors of the individual sub-systems. In a preferred embodiment even the refreshing of the SDRAMs of the individual sub-systems can be discontinued.

A preferred embodiment of the invention is explained in greater detail in the following by making reference to the drawings in which:

- Fig. 1 is a block diagram of a prior art smart phone,
- Fig. 2 is a block diagram of a loosely coupled multi processor system in accordance with the invention,
- Fig. 3 is a block diagram of an alternative embodiment,
- Fig. 4 is a block diagram of an implementation of a smart phone in accordance with the invention,
- Fig. 5 is illustrative of a zero-delay multiplexer which can be used in the embodiment of Fig. 4,
- Fig. 6 is illustrative of an embodiment of a zero-delay multiplexer which can be used in the embodiment of Fig. 4,
- Fig. 7 is a flow chart being illustrative of a method for generating boot-images for the sub-systems and
- Fig. 8 is illustrative of an embodiment of the method of the invention.

- 6 -

Fig. 2 shows an electronic system having sub-systems S1, S2, S3 ...Sn. Each of the sub-systems has a RAM 18 and a microprocessor 19 which is coupled to its respective RAM 18.

Each of the sub-systems S1 to Sn is coupled to a controller 20 via bus 21. The controller 20 is coupled to nonvolatile memory 22 which comprises a memory portion 23 containing codes S1, S2, S3... Sn for the respective sub-systems S1 to Sn. For example the code S1 is the boot-image for the sub-system S1, the code S2 is the boot-code for the sub-system S2 and so on.

The codes S1 to Sn are stored in the memory portion 23 in sequential order. This allows to utilize an inexpensive memory device for the memory 22 such as an NAND-FLASH memory. Storing the codes S1 to Sn in sequential order avoids random memory access operations. As a consequence the access time provided by an inexpensive NAND-FLASH memory device is largely sufficient for sequentially reading the respective codes.

When the electronic system of Fig. 2 is switched on the controller 20 performs a read operation of the memory portion 23 in order to sequentially read out the codes S1 to Sn. The controller 20 provides the respective code portions to the sub-systems S1 to Sn via bus 21. After the boot-images have been loaded into all sub-system S1 to Sn the normal system operation can immediately start.

Fig. 3 is an alternative embodiment of the electronic system of Fig. 2. The same reference numbers are used for like elements.

In contrast to the embodiment of Fig. 2 the electronic system of Fig. 3 has a memory 22 for storing portions of boot-image code for different sub-systems in the same memory line. In the case of n sub-systems the first line of the memory 22 contains the first code portions S11, S21, S31, ..., Sn1 of all the

- 7 -

sub-systems S1 to Sn. The following lines of memory contain the further code portions.

This is to support the parallel boot-loading for all sub-systems S1 to Sn to be booted. In this case the Flash memory system 22 can be realized to provide a data width representing the summed data width of all respective sub-systems.

Sub-system sequential boot vs Sub-systems parallel boot:
For example, assuming three sub-systems to be booted, each of which having an oriented 32-bit processing width, and a 64 Mbyte total Flash memory 22, 32-bit wide access and 50 ns access time.

In this example a sub-system S1 to S3 sequential boot takes:

$$64 \cdot \frac{10^6}{4} \cdot 50 \cdot 10^{-9} = 0.8 \quad \text{seconds.}$$

The DRAM loading time for all three sub-systems would be 0,8 seconds, thus resulting in an overall system boot-time of < 1 seconds in this example.

Subsystems S1 to S3 parallel boot:

$$64 \cdot \frac{10^6}{4 \cdot 3} \cdot 50 \cdot 10^{-9} = 0.267 \quad \text{seconds}$$

The DRAM loading time for all 3 sub-systems would be 0,267 sec, thus resulting in an overall system boot-time of < 0,3 ns.

Fig. 4 is another embodiment of an electronic system of the invention. The electronic system of Fig. 4 is a smart phone having a cell phone sub-system 24, a PDA sub-system 25 and a GPS sub-system 26.

- 8 -

The cell phone sub-system 24 has a cell phone engine 27 and a system RAM 28. The cell phone engine 27 and the system RAM 28 are coupled by a multiplexer 29. The multiplexer 29 and the system RAM 28 each have a control input for receiving boot control information via line 30 from controller 31.

The multiplexer 29 is coupled to the controller 31 via line 32 for receiving of boot-image data for the cell phone sub-system 24. The connection path between the multiplexer 29 to the controller 31 can include an optional code compression co-processor 33.

Further the cell phone engine 27 is coupled to the controller 31 via line 34 for the purposes of control and transmission of boot-image data to the cell phone engine 27.

Further the cell phone engine 27 is coupled to high frequency radio component 35 which is coupled to antenna 36 in order to realize an air interface.

Further the cell phone engine 27 is coupled to audio I/O system 37 which is coupled to a speaker and a microphone.

The cell phone sub-system 24 is powered by DC-power sub-system 38. In particular the system RAM 28 and the multiplexer 29 receive a power signal from output P5 of the DC-power sub-system 38 whereas the remaining components of the cell phone sub-system 24, including the cell phone engine 27, receive power via the line P2 of the DC-power sub-system 38.

The PDA sub-system 25 has a main processor 39 for the performance of typical PDA functions. Analogously to the cell phone sub-system 24 the main processor 39 is coupled to system RAM 40 via multiplexer 41. Again the system RAM 40 and the multiplexer 41 have control inputs for receiving boot control information from the controller 31 via line 42 and the multiplexer 41 is connected to controller 31 via line 43 to

- 9 -

receive boot-image data via optional code compression co-processor 33. Again main processor 39 is also connected to controller 31 by line 44.

Further the main processor 39 is coupled to output unit 44 including an LCD display and to a mass storage device, such as smart media, via mass storage controller 45.

The system RAM 40 and the multiplexer 41 receive power via line P5 of the DC-power sub-system 38 and the remaining components of the PDA sub-system 25, including main processor 39, receive power via line P4 of the DC-power sub-system 38.

The GPS sub-system 26 has a GPS system processor 46 which is coupled to its system RAM 47 via multiplexer 48. The system RAM 47 and the multiplexer 48 have control inputs which are coupled to the boot-control output of the controller 31 via line 49.

Further the GPS system processor 46 is coupled to GPS receiver 50 having an antenna 51.

The system RAM 47 and the multiplexer 48 receive power from the output P5 of the DC-power sub-system 38 whereas the remaining components of the GPS sub-system 26, including GPS system processor 46, receive power from the output P3 of the DC-power sub-system 38.

The above described cell phone sub-system 24, PDA sub-system 25 and GPS sub-system 26 are inter-coupled by busses 52 and 53 for loosely coupling the sub-systems.

The DC-power sub-system 38 is connected to the controller 31 via line 54 for the purposes of power control, in particular with respect to bringing the electronic system into a sleep-mode.

- 10 -

Further the controller 31 is coupled to nonvolatile memory 54. Nonvolatile memory 54 corresponds to memory 22 of Fig. 2 and serves to store the boot-image data of the cell phone sub-system 24, the PDA sub-system 25 and the GPS sub-system 26 in consecutive order. The optional code compression co-processor 33, the controller 31 and the memory 54 are comprised in a boot control system 55. The boot control system 55 receives power from the DC-power sub-system 38 from its output P1.

When the electronic circuit of Fig. 4 is switched on power is supplied to the boot control system 55 from the output P1 of the DC-power sub-system 38. The controller 31 directs the DC-power sub-system 38 to apply power to the components of the cell phone sub-system 24, the PDA sub-system 25 and the GPS sub-system 26.

The controller 31 performs a read request to the memory 54 in order to read out the boot-image data contained in memory 54 sequentially. A single read request is sufficient in order to start the screening of the complete boot-image data out of the memory 54 to the controller 31.

In the following it is assumed that the boot-image data of the cell phone sub-system 24, the PDA sub-system 25 and the GPS sub-system 26 are stored in the memory 54 in the same order. Thus the controller 31 first receives the boot-image data for the cell phone sub-system 24 and transmits the boot-image data to the cell phone sub-system 24.

This can be done via lines 32 and / or 34. Typically the boot-image data to be stored in system RAM 28 is provided to the multiplexer 29 which receives a control input via line 30 in order to pass the boot-image data to the system RAM 28 for storage. The cell phone engine 27 can receive register values via line 24 from the controller 31. The register values can be embedded in the boot-image data.

- 11 -

To identify the register values control data in the boot-image data can be utilized for detection by the controller 31 such that the controller 31 can direct the required portions of the boot-image data to the appropriate components of the cell phone sub-system 24.

When the complete boot-image data for the cell phone sub-system 24 is loaded the same procedures are subsequently carried out with respect to the PDA sub-system 25 and the GPS sub-system 26. When this is completed the normal system operation can start.

When the system is not used for some time it can be brought into a sleep-mode. There is a variety of options for such a sleep-mode:

1. The controller 31 can switch off the clocks of the microprocessors i.e. cell phone engine 27, main processor 39 and GPS system processor 46, while directing the DC-power sub-system 38 to continue to supply power via lines P2, P3, P4, respectively. This way the operation of the respective processors is stopped while preserving the contents of the processors register values. The refreshing of the system RAMs 28, 40 and 47 is done by the controller 31 in order to preserve the data in those memories. This way a substantial amount of power can be saved. In order to bring the system back to life only the clocks for the processors have to be switched on again and normal system operation can be resumed immediately.
2. This option is different from the first one in that both the clocks and the power of the microprocessors are switched off. This has the advantage that even more power can be saved. However for bringing the system back to life the registers of the microprocessors have to be reloaded from the memory 54 which takes some amount of time.

- 12 -

3. This option is different from the first two options in that the power and clocks for the sub-system 24, 25 and 26 are completely switched off. This has the advantage that even more power can be saved; on the other hand the complete boot-images have to be reloaded.

In comparison to the prior art system of Fig. 1 a significant overall system cost saving can be accomplished by means of the system of Fig. 4 by (i) a reduced size and amount of overall system FLASH memory (ii) application of highly cost effective large scale integrated FLASH memories as the entire boot / program / code of all sub-systems and applications can be concentrated in one common memory and (iii) utilizing highly cost effective large scale SDRAM memories in the core-processing-sub-systems of the overall system.

For further reduction of the size of the FLASH memory it is advantageous to use code compression and decompression by means of code compression co-processor 33. In addition to code compression an error correcting code can be used such that a very in-expensive memory device 54 having a high error rate can be utilized. Further the programs and applications exercising directly out of the high speed SDRAMs will increase the individual sub-system operation speed - as well as the overall system performance.

It is a particular advantage of the system of Fig. 4 that the SDRAMs can be disconnected from the high speed processor address and data path. Typically the summing-delay of the closed loop adding up the processor decision time plus the address generation time plus the storage access time delay results in the system cycle-time - the overall measure defining the system performance. For this very fact no additional delaying parts can be tolerated which would add a substantial delay time to this critical loop.

- 13 -

In order to disconnect the SDRAM from the processor loop and to tie SDRAM to an entirely different control system setting up and preparing the SDRAM during the boot time usage of a specific multiplexer type consisting of MOS transistors is advantageous. The objective is to provide a very low resistance switch which leads to a negligible time delay for the critical system speed determining loop in combination with the specific load capacity of the SDRAM.

Fig. 5 illustrates a preferable solution. The basic element of the switch is a fast, low on resistance, low capacitance, high current capacity MOS-FET switch. The combination of low on resistance and low capacitance is provided by a CMOS process allowing a short CMOS channel length.

The switch shown in Fig. 5 consists of a N-channel MOS transistor driven by a CMOS gate. When the switch is enabled the gate of the N-channel transistor is at V_{cc} (+5V) and the device is on, with a typical on resistance of $< 5\Omega$. When this is disabled, the switch is off. The N-channel is preferably structured causing no direct leakage and to represent very low capacitance across the transistor in the off state. The off state leakage of this type of circuits is typically $< 1\text{ nA}$ at room temperature.

Several companies are offering and producing this type of circuits - among them - Quality Semiconductor, Inc. of Santa Clara, USA.

Fig. 6 is illustrative of a block diagram of an integrated implementation of the multiplexers 29, 41 and 48 of Fig. 4 using devices of the type as shown in Fig. 4. As the on-resistance of the switch circuits is below 5Ω and the typical load capacity is 25 pF this leads to a delay time of below $0,2\text{ ns}$ which is of no significance for the processor critical loop delay.

- 14 -

Fig. 7 is illustrative of a method to obtain the boot-images of the sub-systems of the system of Fig. 4. In step 60 the sub-systems S1 to Sn are implemented in a test system. The test system is booted in the usual way. In step 61 the RAMs and the microprocessor registers of the sub-systems S1 to Sn are scanned in order to obtain corresponding boot-images I1 to In. In step 62 the boot-images I1 to In are concatenated by writing the boot-images sequentially into a nonvolatile RAM corresponding to memory 54 of the system in Fig. 4.

Fig. 8 is illustrative of the options provided by the system of Fig. 4 for bringing the system into a sleep-mode. In step 70 the boot-images I1 to In for the respective sub-systems S1 to Sn are loaded such that the normal system operation can start. When the system has not being used for some times the clocks of the microprocessors can be switched off in step 71 while power remains on. In step 72 the DRAMs are continued to be refreshed by the boot controller (cf. controller 31 of Fig. 4). In step 73 the clocks of the multi processors are switched on again such that normal system operation can be resumed in step 74.

Alternatively both the clocks and the power of the microprocessors can be switched off in step 75 while the DRAMs are continued to be refreshed by the boot controller in step 76. In step 77 the microprocessor registers are reloaded and the clocks are switched on in order to resume normal system operation in step 74.

As a further alternative both the clocks and the power for all components of the sub-systems can be switched off in step 78. In this case the complete boot-images I1 to In need to be reloaded in step 79 to resume normal system operation in step 74.

- 15 -

L I S T O F R E F E R E N C E N U M E R A L S

smart phone	01
cell phone sub-system	02
PDA sub-system	03
GPS sub-system	04
cell phone engine	05
FLASH memory	06
RAM	07
PDA system processor	08
FLASH memory	09
RAM	10
human machine interface	11
mass storage	12
GPS receiver	13
FLASH memory	14
RAM	15
Bus	16
Bus	17
RAM	18
microprocessor	19
controller	20
bus	21
memory	22
memory portion	23
cell phone sub-system	24
PDA sub-system	25
GPS sub-system	26
cell phone engine	27
system RAM	28
multiplexer	29
line	30
controller	31
line	32
code compression co-processor	33

- 16 -

line	34
radio component	35
antenna	36
audio I/O system	37
DC-power sub-system	38
main processor	39
system RAM	40
multiplexer	41
line	42
line	43
output device	44
mass storage controller	45
GPS system processor	46
system RAM	47
multiplexer	48
line	49
GPS receiver	50
antenna	51
bus	52
bus	53
memory	54
boot control system	55

- 17 -

CLAIMS

1. An electronic system having
 - a first sub-system (24) and a second sub-system (25, 26),
 - memory means (54) for storing of first boot-image data for the first sub-system and for storage of second boot-image data for the second sub-system,
 - controller means (31) for loading the first boot-image data to the first sub-system and for loading the second boot-image data to the second sub-system.
2. The electronic system of claim 1 at least one of the first and second sub-system having microprocessor means (27, 39, 46) and random access memory means (28, 40, 47).
3. The electronic system of claim 2 further comprising multiplexer means (29, 41, 48) being coupled between the processor means and the random access memory means and having a control input coupled to the controller means.
4. The electronic system of claim 3 wherein the multiplexer means is of the zero-delay type.
5. The electronic system of claims 2, 3 or 4 the random access memory means being of a DRAM type.
6. The electronic system of anyone of the proceeding claims the memory means being a single nonvolatile random access memory means.
7. The electronic system of claim 6 the nonvolatile random access memory means being of a low cost mass product non-volatile storage such as e.g. NAND-FLASH memory type.

- 18 -

8. The electronic system of anyone of the proceeding claims further comprising means for code decompression (33) of the first and second boot-image data.

9. The electronic system of anyone of the proceeding claims further comprising means for decoding and / or error correction of the coded first and second boot-image data.

10. The electronic system of anyone of the claims 2 to 9 the controller means being adapted to discontinue the clocking of the microprocessor means while maintaining the power supply for the microprocessor means and being adapted to perform a refresh operation of the random access memory means.

11. The electronic system of anyone of the claims 2 to 10 the controller means being adapted to discontinue the clocking of the microprocessor means and to switch off the power supply for the microprocessor means and to reload the microprocessor means registers by means of boot-image data from the memory means for resuming normal system operation.

12. The electronic system of anyone of the claims 2 to 11 the controller means being adapted to discontinue the clocking of the microprocessor means, to discontinue a refresh operation of the random access memory means and to switch off a power supply of the microprocessor means and the random access memory means and being further adapted to reload boot-images to resume normal system operation.

13. The electronic system in accordance with anyone of the proceeding claims, the electronic system being a computer, a mobile computer, a personal digital assistant, a smart phone or any other pervasive computing device.

- 19 -

14. A method for initializing an electronic system having first and second sub-systems, the method comprising the steps of:

- storing of first boot-image data for the first sub-system in a memory means,
- storing of second boot-image data in the memory means at consecutive storage locations with respect to the first boot-image data.

15. The method of claim 14 the first boot-image data having a first sub-sequence of boot-image data for a random access memory of the first sub-system and having a second sub-sequence of boot-image data for microprocessor means of the first sub-system.

16. The method of claims 14 or 15 further comprising error correction coding of the boot-image data.

17. The method of claims 14, 15 or 16 further comprising code compression coding of the boot-image data.

18. The method according to claims 14 to 16 which further comprises a method for booting comprising the steps of:

- loading of first boot-image data into the first sub-system,
- loading of the second boot-image data into the second sub-system,
- starting normal operation of the electronic system.

19. The method of claim 18 further comprising error decoding of the boot-image data.

20. The methods of claims 18 or 19 further comprising code decompression of the boot-image data.

- 20 -

21. The method of claims 14 to 19 wherein said electronic system having first and second sub-systems, each of the sub-systems having separate microprocessor means and random access memory means.

22. The method of claim 21 wherein a method for operating said electronic system comprising the steps of:

- discontinuing the clocking of the microprocessor means while maintaining a power supply for the microprocessor means,
- refreshing the random access memory means by means of a controller,
- resuming the clocking of the microprocessor means by the controller to resume normal system operation.

23. The method of claim 21 wherein a method for operating said electronic system comprising the steps of:

- discontinuing the clocking of the microprocessor means and switching off a power supply for the microprocessor means,
- refreshing of the random access memory means by means of a controller,
- reloading the microprocessor means registers and continuing the clocking of the microprocessor means in order to resume normal system operation.

24. The method of claim 21 wherein a method for operating said electronic system comprising the steps of:

- discontinuing the clocking of the microprocessor means,
- switching off a power supply of the microprocessor means and the random access memory means,
- reloading the first and second boot-image data into the first and second sub-systems, respectively,
- resuming the clocking of the microprocessor means and resuming refreshing of the random access memory means and switching on the power supply for the microprocessor means and

- 21 -

the random access memory means in order to resume normal system operation.

25. The method of anyone of the claims 14 to 24, the method further comprising the steps of:

- storing of portions of boot-image data of the sub-systems in a line of the memory means,
- reading lines of boot-image data to load the boot-image data in parallel to the sub-systems.

26. A computer program product stored on a computer usable medium, comprising computer readable program means for causing an electronic system to perform a method according to anyone of the proceeding claims 14 to 25 when the program is run on the electronic system.

1 / 5

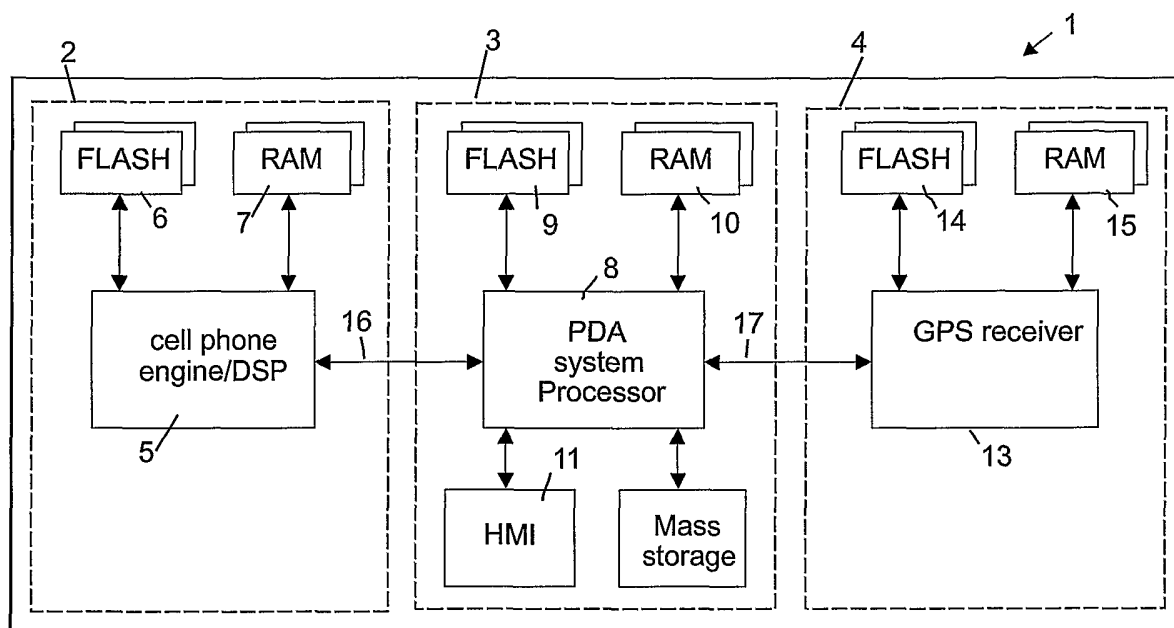


FIG. 1

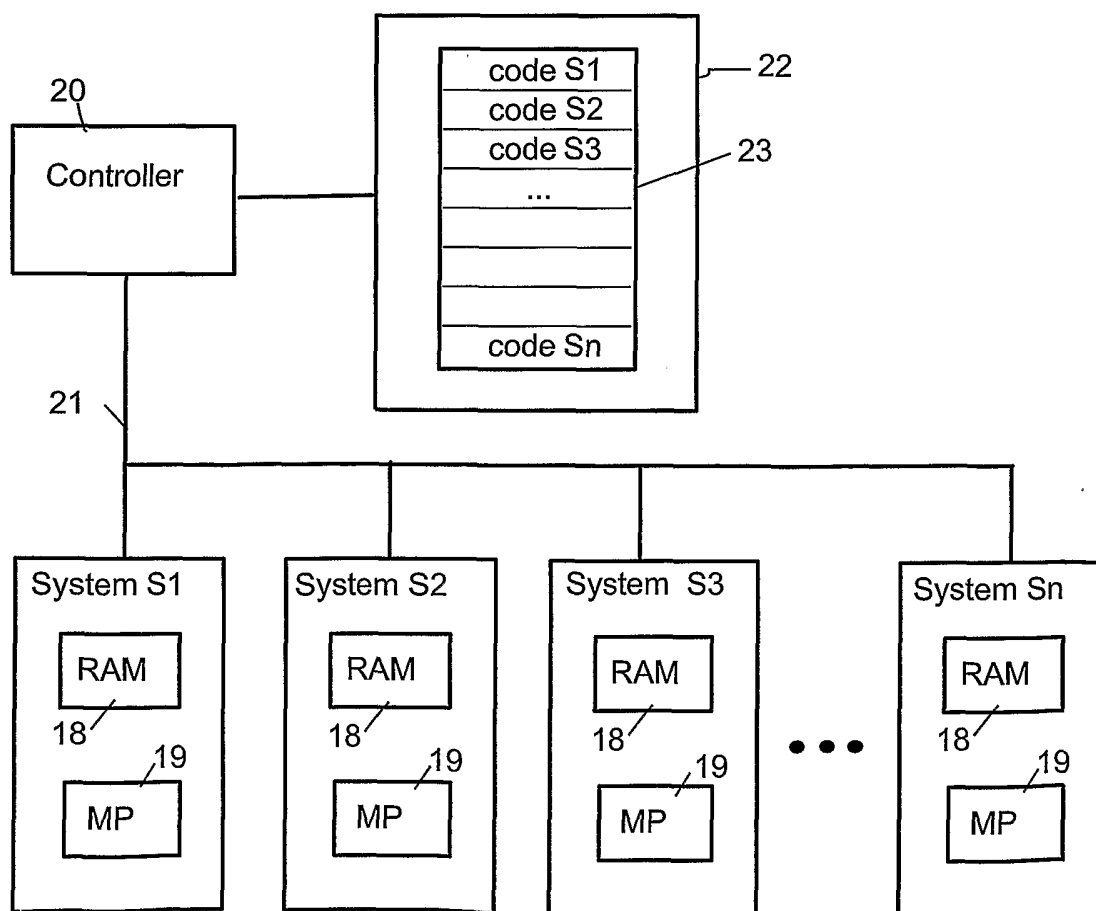


FIG. 2

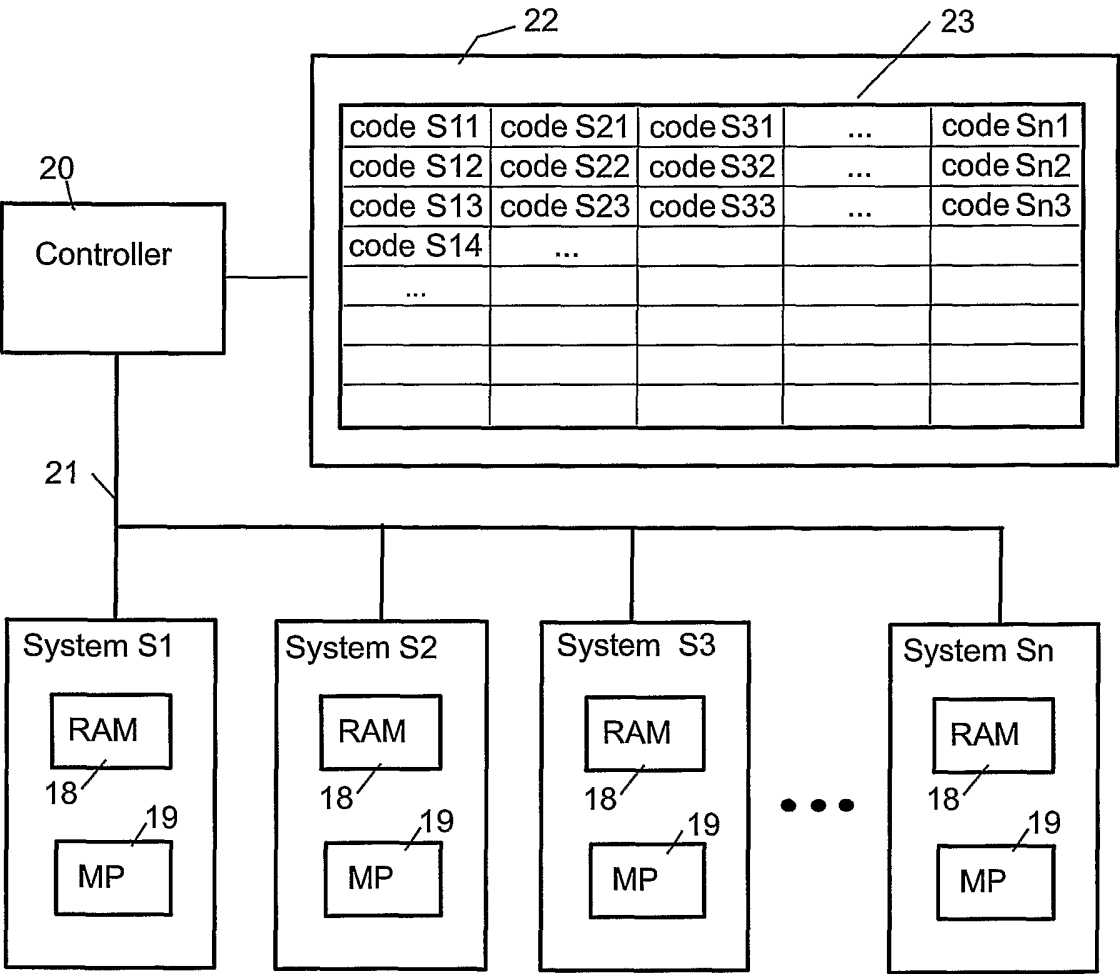


FIG. 3

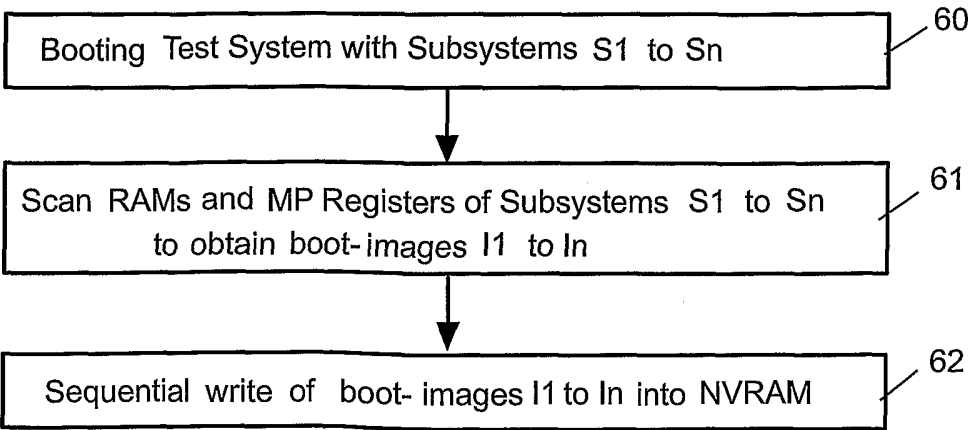


FIG. 7

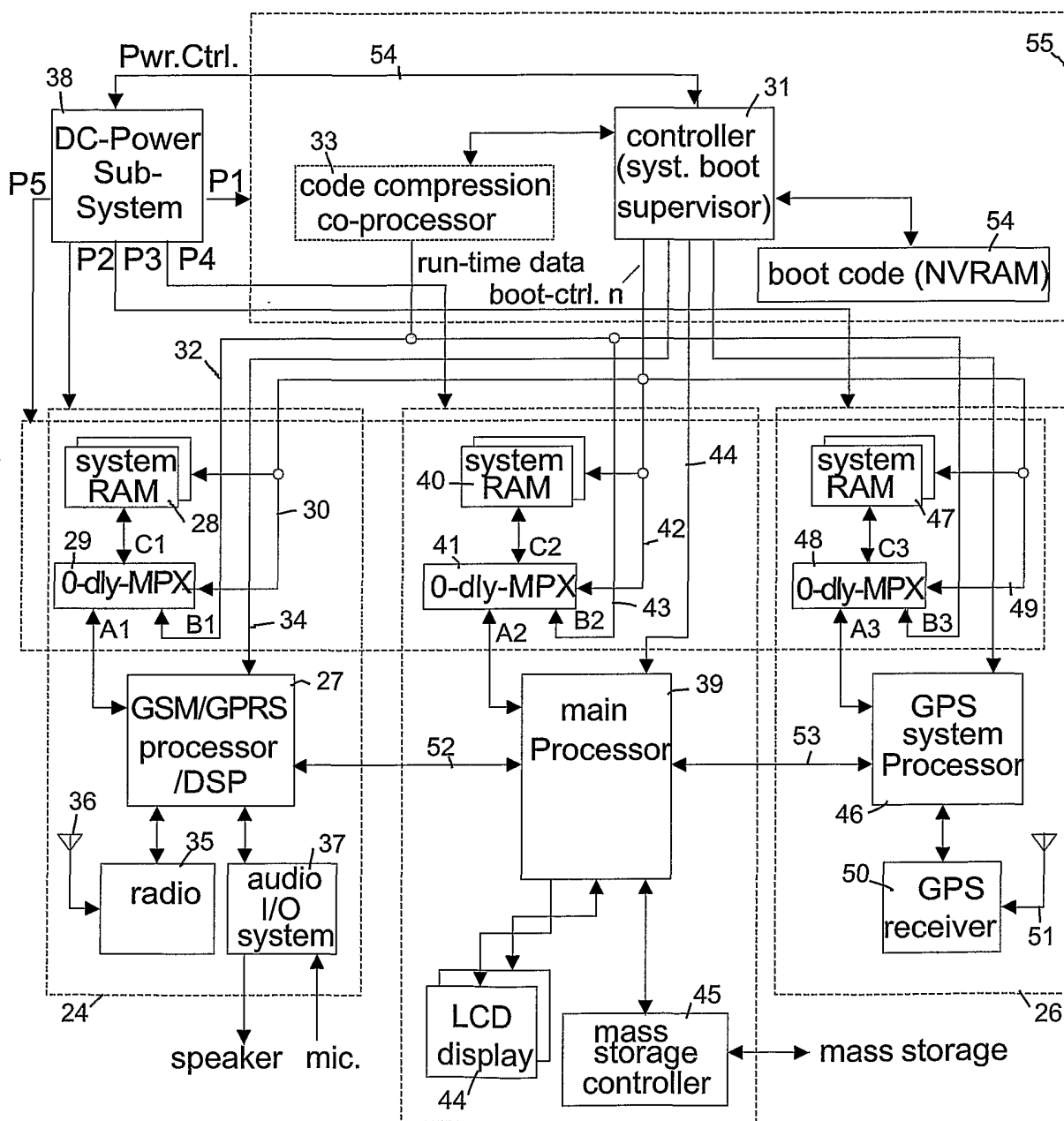


FIG. 4

4 / 5

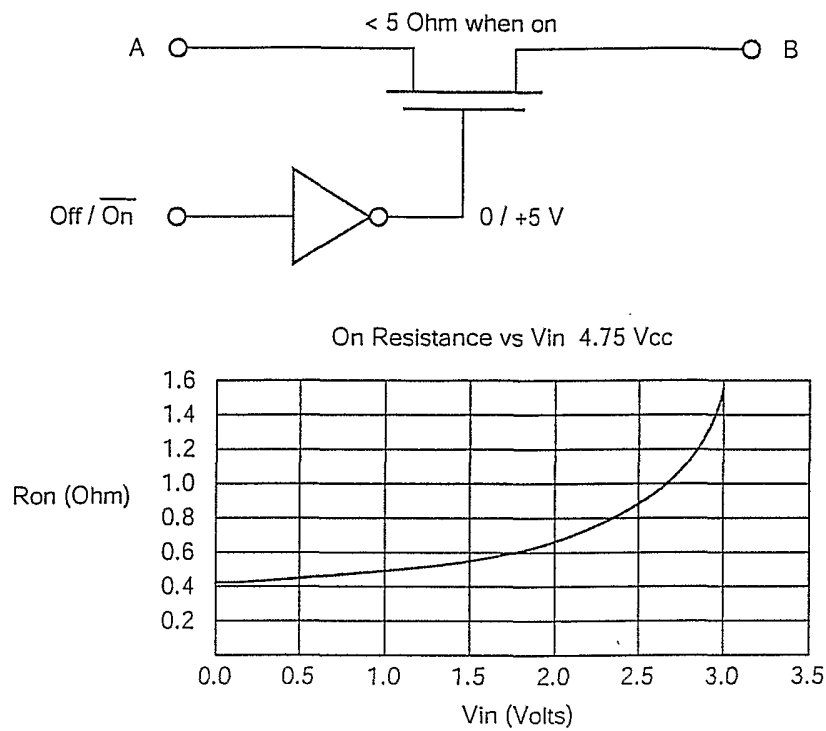


FIG. 5

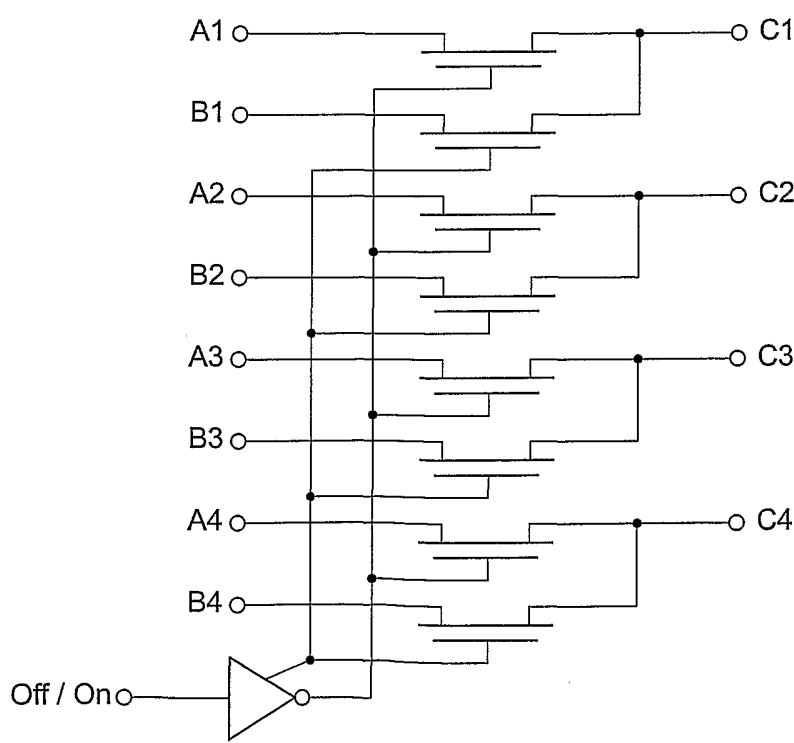


FIG. 6

5 / 5

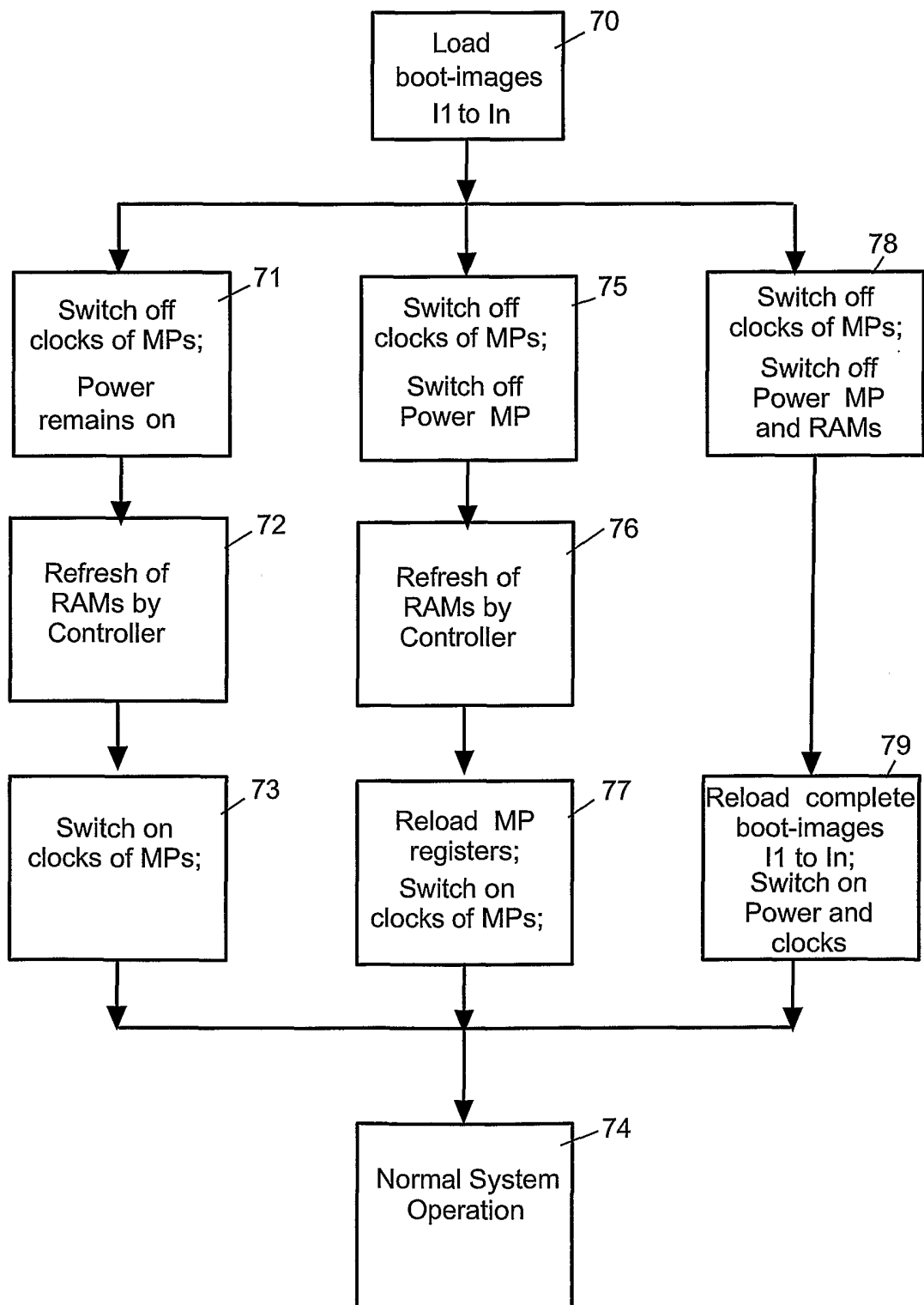


FIG. 8