



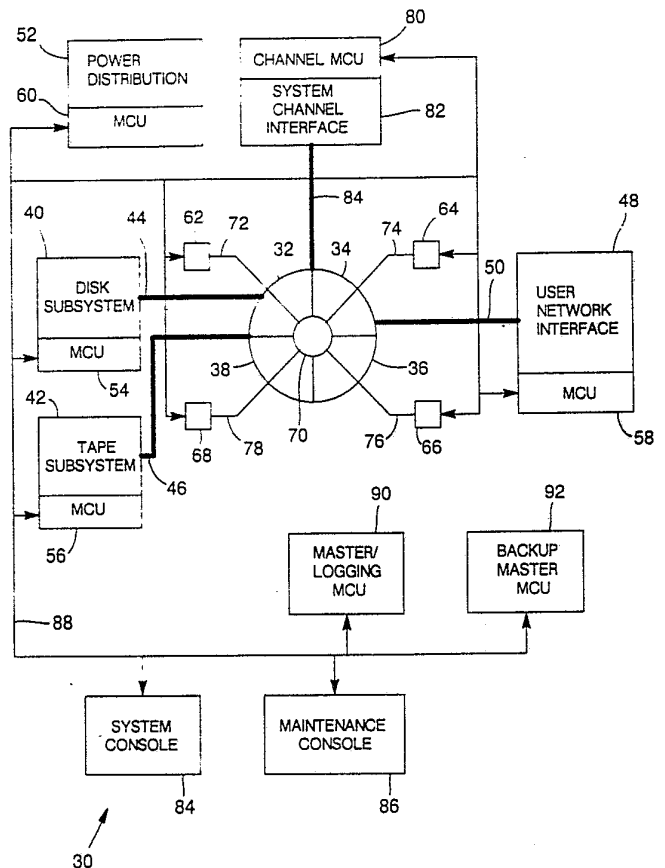
INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

<p>(51) International Patent Classification ⁵ : G06F 11/32</p>	<p>A1</p>	<p>(11) International Publication Number: WO 91/20035 (43) International Publication Date: 26 December 1991 (26.12.91)</p>
<p>(21) International Application Number: PCT/US91/04075 (22) International Filing Date: 10 June 1991 (10.06.91) (30) Priority data: 535,901 11 June 1990 (11.06.90) US (71) Applicant: SUPERCOMPUTER SYSTEMS LIMITED PARTNERSHIP [US/US]; 1414 W. Hamilton Avenue, Eau Claire, WI 54701 (US). (72) Inventors: SPIX, George, A. ; 3309 Westover Lane, Eau Claire, WI 54701 (US). COLLIER, Glen, L. ; 2215 Folsom Street, Apt. 329, Eau Claire, WI 54703 (US). THROOP, G., Joseph ; 3355 Sharon Drive, Eau Claire, WI 54701 (US). CLOUNCH, David, L. ; 18931 Wisconsin Drive, Chippewa Falls, WI 54729 (US). BEARD, Douglas, R. ; S10505 Lowes Creek Road, Eleva, WI 54738 (US). RHEA, Cris, J. ; W2610 Rim Rock Road, Eau Claire, WI 54701 (US).</p>		<p>(74) Agents: PEDERSEN, Brad, D.; Dorsey & Whitney, 2200 First Bank Place East, Minneapolis, MN 55402 (US) et al. (81) Designated States: AT (European patent), BE (European patent), CH (European patent), DE (European patent), DK (European patent), ES (European patent), FR (European patent), GB (European patent), GR (European patent), IT (European patent), JP, KR, LU (European patent), NL (European patent), SE (European patent). Published <i>With international search report.</i> <i>Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.</i></p>

(54) Title: CONTROL AND MAINTENANCE ARCHITECTURE FOR A HIGHLY PARALLEL MULTIPROCESSOR SYSTEM

(57) Abstract

The present invention includes methods and apparatus for a maintenance and control systems (54, 56, 58, 60, 62, 64, 66, 68) for sensing and controlling the numerous sections of a highly parallel multiprocessor system. The control and maintenance system communicates with all processors (32, 34, 36, 38), all peripheral systems (40, 42), all user interfaces (48) to the multiprocessor system, a system console (84), and the power and environmental control subsystems (52).



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	ES	Spain	MG	Madagascar
AU	Australia	FI	Finland	ML	Mali
BB	Barbados	FR	France	MN	Mongolia
BE	Belgium	GA	Gabon	MR	Mauritania
BF	Burkina Faso	GB	United Kingdom	MW	Malawi
BG	Bulgaria	GN	Guinea	NL	Netherlands
BJ	Benin	GR	Greece	NO	Norway
BR	Brazil	HU	Hungary	PL	Poland
CA	Canada	IT	Italy	RO	Romania
CF	Central African Republic	JP	Japan	SD	Sudan
CG	Congo	KP	Democratic People's Republic of Korea	SE	Sweden
CH	Switzerland	KR	Republic of Korea	SN	Senegal
CI	Côte d'Ivoire	LI	Liechtenstein	SU	Soviet Union
CM	Cameroon	LK	Sri Lanka	TD	Chad
CS	Czechoslovakia	LU	Luxembourg	TC	Togo
DE	Germany	MC	Monaco	US	United States of America
DK	Denmark				

5

10 **CONTROL AND MAINTENANCE ARCHITECTURE FOR A HIGHLY
 PARALLEL MULTIPROCESSOR SYSTEM**

TECHNICAL FIELD

 This invention relates generally to the field of maintenance and
15 control of computer systems. More particularly, it relates to an integrated
 system for controlling and maintaining a high-speed supercomputer and
 its peripheral devices using a number of maintenance control units
 connected to the various sections of the computer system.

20 **BACKGROUND ART**

 Various high-speed computer processing systems, sometimes
 referred to as supercomputers, have been developed to solve a variety of
 computationally intensive applications, such as weather modeling,
 structural analysis, fluid dynamics, computational physics, nuclear
25 engineering, real-time simulation, signal processing, etc. The overall
 design or architectures for such present supercomputers can be generally
 classified into one of two broad categories: minimally parallel processing
 systems and massively parallel processing systems.

 The minimally parallel class of supercomputers includes both
30 uniprocessors and shared memory multiprocessors. A uniprocessor is a
 very high-speed processor that utilizes multiple functional elements,
 vector processing, pipeline and look-ahead techniques to increase the
 computational speed of the single processor. Shared-memory
 multiprocessors are comprised of a small number of high-speed processors
35 (typically two, four or eight) that are tightly-coupled to each other and to a
 common shared-memory using either a bus-connected or direct-connected
 architecture.

At the opposite end of the spectrum, the massively parallel class of supercomputers includes both array processors and distributed-memory multicomputers. Array processors generally consist of a very large array of single-bit or small processors that operate in a single-instruction-multiple-
5 data (SIMD) mode, as used for example in signal or image processing. Distributed-memory multicomputers also have a very large number of computers (typically 1024 or more) that are loosely-coupled together using a variety of connection topologies such as hypercube, ring, butterfly switch and hypertrees to pass messages and data between the computers in a
10 multiple-instruction-multiple-data (MIMD) mode.

Because of the inherent limitations of the present architectures for minimally parallel and massively parallel supercomputers, such computer processing systems are unable to achieve significantly increased processing speeds and problem solving spaces over current systems. The
15 previously filed and related application entitled CLUSTER ARCHITECTURE FOR A HIGHLY PARALLEL SCALAR/VECTOR MULTIPROCESSOR SYSTEM, PCT Serial No. PCT/US90/07655, sets forth a new cluster architecture for interconnecting parallel processors and associated resources that allows the speed and coordination of current
20 minimally parallel multiprocessor systems to be extended to larger numbers of processors, while also resolving some of the synchronization problems associated with massively parallel multicomputer systems. This range between minimally parallel and massively parallel systems will be referred to as highly parallel computer processing systems and can include
25 multiprocessor systems having sixteen to 1024 processors. The cluster architecture described in the related application provides for one or more clusters of tightly-coupled, high-speed processors capable of both vector and scalar parallel processing that can symmetrically access shared resources associated with the cluster, as well as shared resources associated
30 with other clusters.

Just as the traditional system architectures were ill-suited for solving the problems associated with highly parallel multiprocessor systems, so too are the traditional control and maintenance architectures. As used within the present specification, the terms control and
35 maintenance refer to any operation by which a system operator can control the operation of the system such as starting, stopping, or n-stepping the master clock, setting or sensing internal machine states, executing

diagnostic routines, and capturing errors at run-time for later display and analysis.

Prior art control and maintenance architectures include the use of scan paths for setting and/or sensing critical internal machine parameters.

5 Control of the scan paths is typically via an external maintenance or diagnostic system. As computer execution speeds increase and systems become more densely packaged, physical access to critical internal machine parameters becomes more difficult, accentuating the need for remote electronic access to these parameters.

10 In highly parallel multiprocessor systems, the packaging density of the design requires that all internal machine registers be accessible to a control and maintenance subsystem. High performance systems use high clock speeds, requiring an increased packaging density, which in turn renders physical access to the system for sensing with traditional test
15 equipment such as oscilloscopes and logic analyzers very difficult, if not impossible. In addition, these traditional diagnostic tools may well be incapable of operating at a high enough speed to be useful.

Furthermore, the complexity of a highly parallel multiprocessor system makes analysis of failing machine sequences extremely difficult
20 unless all internal registers can be sensed by the maintenance subsystem. The amount of information that must be retrieved from a highly parallel multiprocessor system undergoing diagnostic testing is massive, and easily exceeds the capability of traditional scan path architectures. Access to all internal machine registers is also necessary to provide the system with the
25 ability to restart from a specific machine state, such as after stopping the machine in an error situation.

The ability to stop and restart the machine necessarily requires that the maintenance subsystem have the ability to control all processor clocks. In addition, a highly parallel multiprocessor architecture requires that
30 control over processor machine states and clocks be independent. This is necessary for removing a defective processor from operation without halting operation of the entire system. By the same reasoning, it is advantageous for the maintenance system to have control over the power up sequence of each processor, so that a defective processor may be
35 removed from operation, repaired, and restored to operation with minimal impact on the rest of the system.

In the same way that it is undesirable for maintenance work on one processor to halt operation of the entire system, so is it undesirable for maintenance work on one peripheral device to halt operation of the entire system. Thus it is desirable for a control and maintenance subsystem to have independent control over peripheral devices, including their on-line status and power up sequence.

It is clear that there is a need for a control and maintenance architecture specifically designed for the needs of a highly parallel multiprocessor system. Specifically, there is a need for a maintenance subsystem allowing setting and sensing capability for all internal machine registers, the ability to set and sense machine states by management of massive amounts of information, independent control of processor power up sequences, processor clocks, processor machine states, and peripheral devices.

15

SUMMARY OF THE INVENTION

The present invention is directed toward a control and maintenance architecture providing an integrated hardware and software solution to the problem of access to and control over the internal machine registers of a highly parallel multiprocessor system. This is accomplished by providing multiple Maintenance Control Units (MCUs), each of which is a computer workstation or other processing device, and all of which are connected together network via Ethernet or some other networking method. The MCUs are distributed to various portions of the system, such as the processor clusters, the disk subsystem, the tape subsystem, the power distribution subsystem, the user network interface, the system channel interface, and any other appropriate subsystems. In addition, there is a master/logging MCU for coordinating the MCUs' activities, and for gathering error logs from the various MCUs and a maintenance console for providing an interface with a maintenance operator.

An operator working on the maintenance console can log onto any of the distributed MCUs via the network interconnection, and can execute diagnostic programs or perform other control functions on the desired subsystem, or can log onto the MCUs associated with the system processors so as to set or sense internal states of the processors. In addition, the operator can log onto one or more of the system processors themselves, such as for the purpose of executing self diagnostic routines.

35

In a preferred embodiment of the present invention, the master/logging MCU can operate as a file server and can store any or all of the programs to be run on any of the MCUs, and can store start up code for the processors themselves. Scan paths allow processor MCUs access to
5 each of the processors, and through these scan paths the processors can be initialized upon power up with code stored in the master/logging MCU, and can be set and sensed for diagnostic purposes. A backup master MCU is available in the event that the master/logging MCU fails.

In addition, each of the other MCUs may have disk storage of their
10 own, such as for storage of logging information in the event that it cannot be sent to the master/logging MCU due to MCU, network, or other failure.

The software portion of the maintenance system according to a preferred embodiment of the present invention includes various testing routines for testing various portions of the multiprocessor system,
15 including the peripheral and other subsystems. The architecture of the software subsystem is such that a given routine can be executed against multiple targets, including actual hardware (such as a multiprocessor cluster), and a computer simulation of the hardware. Thus a common operator interface is provided to engineering development tools such as
20 simulators, manufacturing tools such as subsystem testers, and actual system hardware.

Accordingly, it is an objective of the present invention is to provide a control and maintenance architecture specifically designed for the needs of a highly parallel multiprocessor system, and which provides setting and
25 sensing capability for all internal machine registers.

Another objective of the present invention is to provide the ability to set and sense machine states by providing a maintenance architecture capable of managing the massive amount of information associated with a highly parallel multiprocessor system.

30 A further object of the present invention is to provide independent control of processor power up sequences, processor clocks, processor machine states, and peripheral devices through a distributed maintenance architecture.

A still further object of the present invention is to provide an
35 architecture for a control and maintenance subsystem, where control and diagnostic routines can be executed on any of the multiple MCUs

distributed throughout the multiprocessor system, and where these programs can be controlled from a single maintenance console.

A still further object of the present invention is to provide a common operator interface between actual system hardware and engineering and development tools such as computer simulators and subsystem testers.

These and other objectives of the present invention will become apparent with reference to the drawings, the detailed description of the preferred embodiment and the appended claims.

DESCRIPTION OF THE DRAWINGS

Figure 1 shows a computer and peripheral architecture typical of those found in the prior art.

Figure 2 shows a block diagram of a highly parallel multiprocessor system architecture according to a preferred embodiment of the present invention.

Figure 3 is a partial block diagram showing the redundant interface between multiple processor MCUs and multiple clusters.

Figure 4 is a block diagram showing the processor MCUs, the Logic Support Station (LSS) control cabinet, and the scan paths.

Figure 5 shows a partial logic diagram of clock control circuitry and error notification circuitry, both of which interact with all MCUs.

Figure 6 is a flow chart of a power up sequence which can be controlled by the MCUs according to a preferred embodiment of the present invention.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

Referring now to Figure 1, a block diagram of a typical prior art computer system 10 is shown. A Central Processing Unit (CPU) 12 is connected to various peripheral devices including a disk subsystem 14, a tape subsystem 16, an interface to the user network 18, and the system console 20. To the extent that maintenance operations are to be performed on the various components of the system, a console is provided for the component, such as a disk subsystem console 22 and tape subsystem console 24. These consoles 22, 24 may be permanently installed, or may be a terminal temporarily attached for the duration of the maintenance operations to be performed.

Such an architecture does not readily lend itself to the needs of a high performance multiprocessor system for several reasons. With separate consoles 20, 22, 24 and segregated control of CPU 12 and peripherals 14, 16, coordination of peripheral maintenance with CPU 12 operations is difficult and results in a substantial negative impact upon system operation. For example, if an error develops on a disk drive, the operator must first halt the CPU 12, then move to the disk subsystem console 22 to perform the maintenance tasks, and then restart the CPU 12 from the system console 20. This results in substantial down time for the system 10 and a corresponding impact on execution of user programs.

With high performance supercomputers such as highly parallel multiprocessor systems, execution time is expensive, and routine maintenance operations are ideally performed with minimal impact of system execution.

Referring now to Figure 2, a block diagram of a computer system maintenance architecture 30 according to a preferred embodiment of the present invention is shown. Multiple processors are connected in a highly parallel multiprocessor configuration, and they are grouped in clusters 32, 34, 36, 38. Associated with each cluster are one or more Input/Output Concentrators (IOC) and a Secondary Memory Systems (SMS) (not shown). Peripheral subsystems include a disk subsystem 40 and a tape subsystem 42, each of which may contain many disk or tape drive units, and both of which are connected to the clusters 32-38 through high speed interfaces 44, 46, such as the High Performance Peripheral Interface (HPPI). A user network interface 48 is also connected to the clusters 32-38 through a high speed interface 50. Power distribution and environmental control (cooling, etc.) is managed by the power distribution subsystem 52, which can independently sequence power to any one or more of the processors within one of the clusters 32-38, and to any of the subsystems 40, 42.

Maintenance Control Units (MCUs) are connected to every subsystem and processor cluster in the system. A disk MCU 54 is connected to the disk subsystem 40, and controls disk maintenance operations by communicating with disk controllers in the disk subsystem (not shown), each of which may support a string of drives. A tape MCU 56 operates in a similar fashion to the disk MCU 54, and controls maintenance operations on the tape subsystem 42. A user network MCU 58 controls the maintenance operations of the user network interface 48,

and power distribution MCU 60 controls the power distribution subsystem 52.

Four processor MCUs, 62, 64, 66, 68 are associated with the four processor clusters 32-38, and connect to the clusters through a clock tower 5 70 via interfaces 72, 74, 76, 78 which in the preferred embodiment of the present invention are TTL busses, but which could be any interface capable of providing the necessary data throughput. Interfaces 72-78 also connect processor MCUs 62-68 to the IOCs and SMSs associated with each cluster 32-38, and references to the connection between processor MCUs 62-68 and 10 clusters 32-38 hereinafter also refer to the connection to the associated IOCs and SMSs.

High speed supercomputers are typically implemented using Emitter Coupled Logic (ECL) devices, which have limitations on interconnect lengths. Since the processor MCUs 62-68 interact directly 15 with processor hardware in ECL, they are preferably connected to the clusters 32-38 through some centrally located point, so as to minimize interconnect lengths. In a preferred embodiment of the present invention, the processor MCUs 62-68 connect to the clusters 32-38 through the clock tower 70 which is centrally located. In an alternate embodiment 20 of the present invention, the MCUs 62-68 connect to the clusters 32-38 through a separate cabinet.

There is also a channel MCU 80 which connects through a system channel interface 82 to the processor clusters 32-38 through a high speed interface 84 such as HPPI. This interface allows channel MCU 80 to 25 communicate with the operating system that is executing on the processor clusters 32-38. Alternatively, the functions performed by the channel MCU 80 could be performed by user network MCU 58 through high speed interface 50, or could be performed by the master/logging MCU 90.

A system console 84 provides an interface to the system operator, 30 and a maintenance console 86 provides an interface to a maintenance operator. The consoles 84, 86 are preferably bit-image terminals so as to allow graphical interface displays. All the MCUs and both consoles are connected together via a network interface 88 such as Ethernet. This arrangement allows the maintenance operator to communicate with any 35 of the MCUs from the maintenance console 88, and execute maintenance or diagnostic routines. It also allows a maintenance operator to communicate with system hardware through processor MCUs 62-68, and

with the system operating system through channel MCU 80. A system operator can also communicate with the system operating system through channel MCU 80.

In addition, information can be gathered from the processor clusters 5 32-38 and the various peripheral subsystems 40, 42, 48, 52 independently by each of the various MCUs, with the gathered information being collected by the master/logging MCU 90 for storage and analysis. In the event that the master/logging MCU 90 fails, a backup master MCU 92 can be provided. Additionally, each MCU may be provided with its own local 10 disk storage on which the information can be stored in the event that neither the master/logging MCU 90 nor the backup master MCU 92 is available.

Each MCU (54, 56, 58, 60, 62, 64, 66, 68, 80, 90, 92) is a workstation or other processor device, and preferably has its own disk storage in the event 15 that both the master/logging MCU 90 and the backup master MCU 92 are unavailable. The MCUs also preferably run the same operating system platform as the processor clusters 32-38 and system development tools (not shown). For example, in the preferred embodiment, the processor clusters 32-38 run a version of the Unix operating system (available from AT&T). 20 In order to obtain the maximum benefit from any diagnostic or maintenance software written, it is desirable to have the design development tools and the MCUs all running the same operating system to increase the portability of the design development tools. Thus, it is desirable to have the MCUs also run a version of Unix. In this way, a 25 diagnostic test written (for example, as a Unix shell script) for a development simulator can still be used when actual hardware is available. It is also desirable to have the maintenance and diagnostic programs being executable against multiple targets. The various subsystems and the processor clusters each can operate as a target against 30 which diagnostic and maintenance routines can be run. These routines can be stored by the master/logging MCU 90 (or any other MCU), and can be controlled through the maintenance console 86.

Those skilled in the art will recognize that there are many variations on the shown example within the scope of the present 35 invention. Specifically, the system or maintenance consoles 84, 86 may appear on the MCU network 88 directly as shown, or may appear attached to an MCU such as the master/logging MCU 90. In addition, other

processor and subsystem configurations will require more or fewer MCUs, as determined by the specific system configuration.

One advantage to the maintenance architecture 30 of the present invention is that it allows maintenance operations to be performed on various components of the system with minimal impact of system performance. For example, if an error develops on a disk drive, the disk subsystem MCU 54 reports the error to the master/logging MCU 90 over the MCU network 88. In order to gracefully take the faulty drive off-line, the maintenance operator (or a monitor program executing on an MCU) issues requests to the operating system executing on processor clusters 32-38 through channel MCU 80 to remove the defective drive from its current configuration. The operating system then can proceed to move data off of the defective device onto others, or to terminate processes whose data cannot be moved. When the defective drive is no longer in use, the operating system informs the operator (or the monitor program) that the defective drive is no longer in use and can be taken off line, which can be accomplished through the disk MCU 54. Drive repair or diagnostic routine execution can then proceed, without further interference in processor execution. A reverse procedure brings the drive back on line after repair operations are complete.

A similar procedure allows for the removal of one or more processors from the system, without totally stopping the system. A request to the operating system to remove a specific processor from its multiprocessor configuration results in current and future processes being diverted so as to run without the specific processor. When removed from operation, the operating system informs the operator (or monitor program) that the processor is no longer being used, and may be shut down for repair or for running diagnostic routines.

Referring now to Figure 3, a partial block diagram of the MCU to processor cluster connection is shown. Four processor MCUs 62, 64, 66, 68 are connected to the MCU network 88, and to the LSS control cabinet 96, which in a preferred embodiment is the clock tower 70, or alternately may be a separate cabinet. These four processor MCUs provide access to the hardware associated with four processor clusters 32, 34, 36, 38. Within the LSS control cabinet 96, each MCU 62-68 connects via a TTL interface 72, 74, 76, 78 to a signal converter 102, 104, 106, 108 which converts between the TTL signals of the MCUs 62-68 and the ECL signals of the clusters 32-38.

Each signal converter 102-108 has two ports on the ECL side, 112a and 112b, 114a and 114b, 116a and 116b, 118a and 118b, each of which is independently selectable by the MCUs 62-68.

5 Scan path logic 122, 124, 126, 128 perform the various logic functions associated with scan paths such as address decoding and serializing/deserializing, in a manner known in the art. Pairs of port connections from the signal converters 102-108 connect to scan path logic 122-128 in a cross-coupled fashion so as to provide redundant access to the clusters by the MCUs 62-68. In this way, each MCU 62-68 can access two
10 clusters 32-38, and each cluster 32-38 can be accessed by two MCUs 62-68. Thus if an MCU fails, the cluster it is associated with remains accessible, since another MCU has access to the same cluster. For example, if MCU 62 fails, the scan paths of cluster 32 are accessible by MCU 64 through signal converter 104, port 114b, and scan path logic 122. This redundancy is
15 important, since without it, the failure of a relatively inexpensive MCU would cause an entire cluster of processors to be inaccessible to the maintenance system.

Under the circumstance of a failing MCU, the total bandwidth between MCUs and clusters is reduced since one MCU must now handle
20 two clusters, although all clusters remain available to the maintenance system. The bandwidth is important in error logging. If an error is detected in the clusters, machine state information is saved by the scan paths, and is gathered by the MCUs 62-68. It is important that the information is read from the scan paths by the MCUs 62-68 as quickly as
25 possible, since a subsequent error in the cluster may overwrite the previous error before it can be gathered. For example, if a single bit memory error is detected, the scan paths can latch the entire machine state for subsequent analysis. However, if a subsequent single bit memory error occurs prior to the latched information being gathered by the MCUs, the
30 second error will overwrite the first error. With a failing MCU, gathering scan path information takes longer due to the reduced bandwidth, resulting in a greater likelihood of an overwrite of scan information by a subsequent cluster error. If more bandwidth is required, more MCUs can be added.

35 Due to the possibility of losing error information due to overwriting by a subsequent error, it is desirable to treat some errors as being more serious than others, and preventing the more serious error

from being overwritten at all. For example, single bit memory errors are not fatal to system operation if the Error Correction Code (ECC) of the memory system can correct for single bit errors. However, double bit errors, if they are not ECC correctable, are fatal to system operation. If a
5 double bit error occurs, it is desirable that the scan paths latch the current machine state, and halt the system clocks to prevent overwrite of the error information.

Those skilled in the art will recognize that many variations of the MCU to cluster interconnections are possible within the scope of the
10 present invention. Specifically, if redundant access is not desired, then each signal converter 102-108 would need only one port on the ECL side and there would be no cross coupling. If redundancy was desired but without the loss of bandwidth associated with a single MCU failure, then redundant MCUs could be provided as well. In general, the number of
15 MCU per cluster affects both the overall bandwidth and the redundancy of access to cluster hardware, and can be varied as required by the specific application.

Referring now to Figure 4, a block diagram shows in more detail the interconnections between the processor MCUs, the LSS control cabinet,
20 and the scan paths. MCUs 62 and 64 connect to the scan path logic 122 and 124 which includes signal converters 102 and 104 as previously shown in Figure 3. Each scan path logic block 122, 124 also includes LSS address decode logic 142, 144. Connected to the LSS address decode logic 142, 144 are multiple groups of two parallel data channels 146, 148, 150, 152, 154, 156
25 which connect to LSSs 158, 160, 170, 176, 178, 180 which are distributed throughout the various parts of the system. The LSSs connect to the various parts of the system through scan paths (not shown), serializing and deserializing data to and from the scan paths in a manner known in the art. Information being gathered or set via the scan paths is bit shifted
30 serially through the path, to be collected (or initialized) by the LSSs.

Those skilled in the art will recognize that each LSS can support any number of independent scan paths desired, without departing from the scope of the present invention. In addition, the LSS Select decode logic 142, 144 may support any number of data channels and LSSs. The number
35 of scan paths appropriate for a given MCU depends upon the amount of information contained in each scan path, the data bandwidth available for data capture, and how quickly all information must be gathered.

Besides initializing and capturing machine state information, the processor MCUs and the LSSs must also control the system clock, such as being able to start and stop the clock, or to single step or multi-step the clock.

5 Referring now to Figure 5, a partial circuit diagram is shown, depicting clock control logic and error notification logic. In a preferred embodiment of the present invention, the system is divided into multiple Maintenance Partitions (MPs), each of which is provided with clock control independent of the other MPs. The partitioning is somewhat
10 arbitrary, but each partition should contain related logic so as to make testing of the MP meaningful. In addition, the size of the MP should be large enough to be a testable portion of the system, but small enough to be easily managed.

Figure 5 shows the clock enable circuit 240 of one MP, and a counter
15 circuit 242 which is common to all MPs. This arrangement allows the clock to be controlled on a per-MP basis. An OR gate 246 enables the clock to the MP if the output of either AND gate 248 or AND gate 250 is at a logical high. AND gate 248 outputs a logical high if the MP is selected for counted clock operation, and AND gate 250 outputs a logical high if MP is
20 selected for free running clock operation. In addition to MP specific control, AND gates 252 and 254 allow system wide control over counted clock mode and free running clock mode, respectively.

To allow free running clocks, the system enable for free running
clocks 256 must be set, as well as the MP enable for free running clocks 258.
25 These bits are set by programming the appropriate LSS register.

To allow counted clocks, the system enable for counted clocks 260
must be set, as well as the MP enable for counted clocks 262. A counter 264
counts down on system clock edges, and when the count reaches zero, the
counted clock enable is turned off by AND gate 266. The enable bits and
30 the counter 264 are also set by programming the appropriate LSS register.

In addition, there is a mechanism for shutting off the clock enables
in the event that a fatal error is detected by a MP. If "stop on error" is
desired, the stop on error bit 268 is set. When a fatal error occurs 270,
NAND gate 272 outputs a logical low, which breaks AND gates 252 and
35 254, disabling both counted clocks and free running clocks. A separate stop
on error bit is provided for each MP. The bit is set by programming the
appropriate LSS register.

Referring now to Figure 6, a flow chart shows an example of a typical operation under the control of the maintenance system MCUs. Specifically, Figure 6 shows a flow chart of a power sequencing control program to be executed by the power distribution subsystem 52. Communication between this subsystem and other subsystems or the maintenance console 86 occurs through power distribution MCU 60 and the MCU network 88.

Upon entering the control program, the system configuration is read 200 to determine which devices are currently part of the configuration. The configuration file may be stored on the master/logging MCU 90, and updates the operating system through the channel MCU 80 as devices are removed or added to the system configuration. If the system is being powered up cold 202, then the cooling system is sequenced on first 204. System power is then applied 206, and a check of environmental parameters is made 208. If all the environmental sensors are OK, then the power distribution MCU 60 is notified of the status 210. The power distribution MCU is then ready to accept requests from other MCUs 212 to adjust the power up status 214 of various devices. If environmental sensors indicate a failure, the control program attempts to recover 216 by adjusting appropriate device parameters 218, or it powers down the device or subsystem 220 if the error is of the type that cannot be corrected.

The example control program depicted in Figure 6 is illustrative only, and many other operations may similarly be controlled by interaction of the various MCUs over the MCU network. These operations may be automated as in the above example, or may be under operator control through a menu prompted interactive program.

Those skilled in the art will recognize that while the preferred embodiment contemplates maintenance and control of a highly parallel multiprocessor system, the disclosed architecture could easily be applied to other system types without departing from the scope of the invention. Specifically, the present invention could be applied to systems such as minimally or massively parallel supercomputers, as well as smaller mainframe computers, while still realizing the benefits afforded by the present invention over the prior art.

Although the description of the preferred embodiment has been presented, it is contemplated that various changes could be made without deviating from the spirit of the present invention. Accordingly, it is

intended that the scope of the present invention be dictated by the appended claims rather than by the description of the preferred embodiment.

We claim:

CLAIMS

1. A control and maintenance architecture for a computer system, comprising:

5 a plurality of maintenance and control units for performing maintenance and control operation on a computer, wherein said plurality of maintenance and control units are distributed to various parts of the computer system;

a maintenance and control network, connected to said plurality of maintenance and control units; and

10 a maintenance and control console connected to said network, and allowing communication of maintenance and control information between an operator working on said console and said plurality of maintenance and control units.

2. A control and maintenance architecture for a highly parallel 15 computer processing system, comprising:

C multiprocessor clusters operably connected to one another, where C is an integer between 2 and 256, inclusive, each multiprocessor cluster comprising:

20 shared resource means for storing and retrieving data and control information;

P high-speed processors capable of both vector and scalar parallel processing, where P is an integer between 2 and 256, inclusive;

25 Q distributed external interface means for transferring data and control information between said shared resource means and one or more external data sources, where Q is an integer between 2 and 256, inclusive;

30 Z arbitration node means operably connected to said processors, said distributed external interface means, and said shared resource means for symmetrically interconnecting said processors and said distributed external interface means with said shared resource means, where Z is an integer between 1 and 128, inclusive, and the ratio of P to Z is greater than or equal to 2;

35 remote cluster adapter means operably connected to remote cluster adapter means in all other of said multiprocessor clusters for allowing said arbitration node

means to access said shared resource means of all other of said multiprocessor clusters and for allowing all other of said multiprocessor clusters to access said shared resource means of this multiprocessor cluster;

5 a plurality of maintenance and control units for performing maintenance and control operation on the high-speed processors;

a maintenance and control network, connected to said plurality of maintenance and control units; and

10 a maintenance and control console connected to said network, and allowing communication of maintenance and control information between an operator working on said console and said plurality of maintenance and control units.

3. A control and maintenance subsystem for use with a computer system, comprising:

15 a computer processor means including a first maintenance and control unit;

a peripheral device connected to said computer processor, and including a second maintenance and control unit; and

20 a console means for interfacing with an operator, wherein said first and second maintenance control units and said console means are connected together so as to form an MCU network and to allow communication of maintenance and control information between said first and second maintenance control units and said system console means.

25 4. A control and maintenance subsystem according to claim 3, further comprising a power distribution subsystem connected to said computer processor means, and including a third maintenance control unit connected to said MCU network.

30 5. A control and maintenance subsystem for use in a highly parallel supercomputer, comprising:

a plurality of computer processor means connected in a highly parallel configuration, and including at least a first maintenance and control unit;

35 one or more peripheral device subsystem connected to said computer processor means, and including at least a second maintenance and control unit; and

a system console means for interfacing with an operator, wherein said first and second maintenance control units and said system console means are connected together so as to form an MCU network and to allow communication of maintenance and control information between said first and second maintenance control units and said system console means.

5
6. A control and maintenance subsystem for use in a highly parallel supercomputer, comprising:

10 a plurality of computer processor means grouped in clusters and connected in a highly parallel configuration;

a plurality of processor maintenance and control units, wherein each of said processor maintenance and control units is connected to and associated with one of said clusters of computer processor means;

15 a peripheral device subsystem connected to said plurality of computer processors;

a peripheral maintenance and control unit connected to said peripheral device subsystem;

20 a power distribution means connected to said plurality of said processor means and said peripheral device subsystem for controlling power sequencing of said processor means and said peripheral device subsystem;

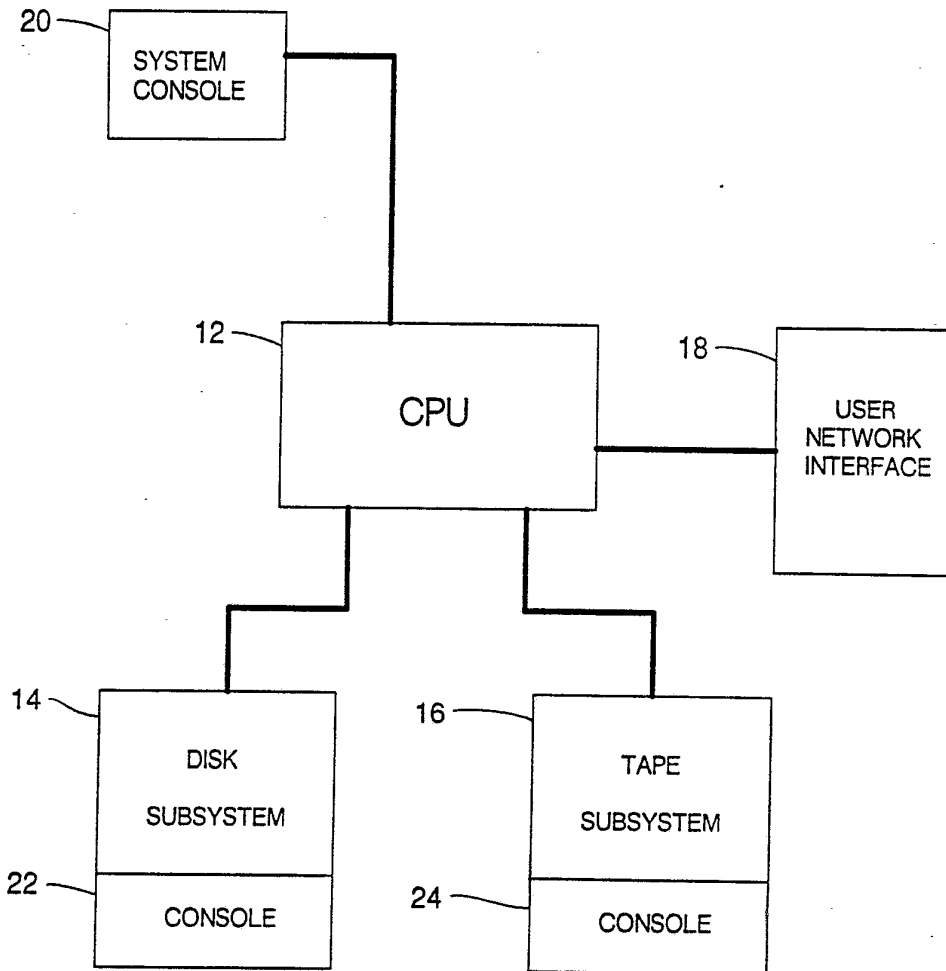
a power maintenance control unit connected to said power distribution means; and

25 a system console means for interfacing with an operator, wherein said processor, peripheral, and power maintenance control units and said system console means are connected together so as to form an MCU network and so as to allow communication of maintenance and control information between said processor, peripheral, and power maintenance control units and said system console means.

30

1/6

FIG. 1
PRIOR ART



10 ↗

2/6
FIG. 2

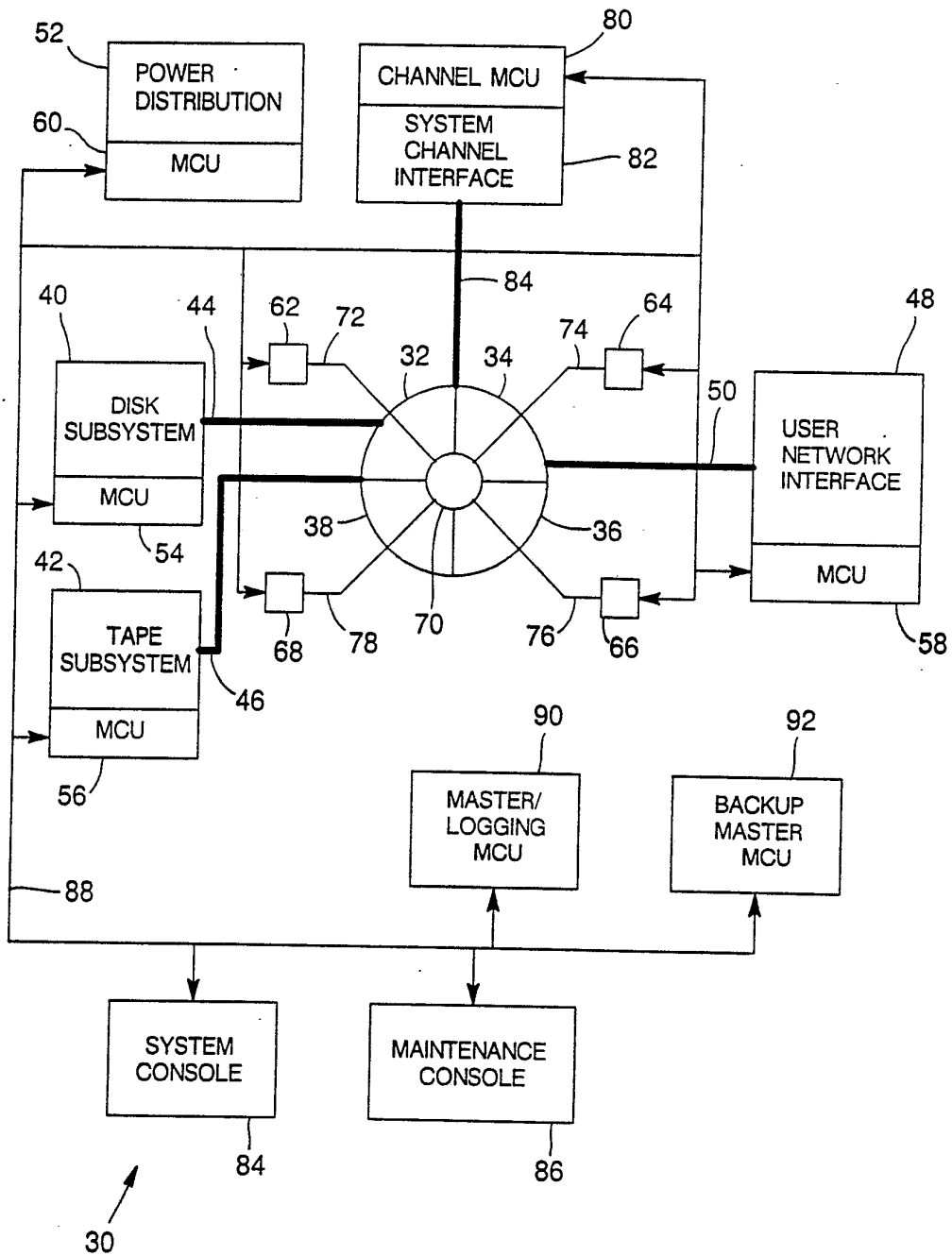
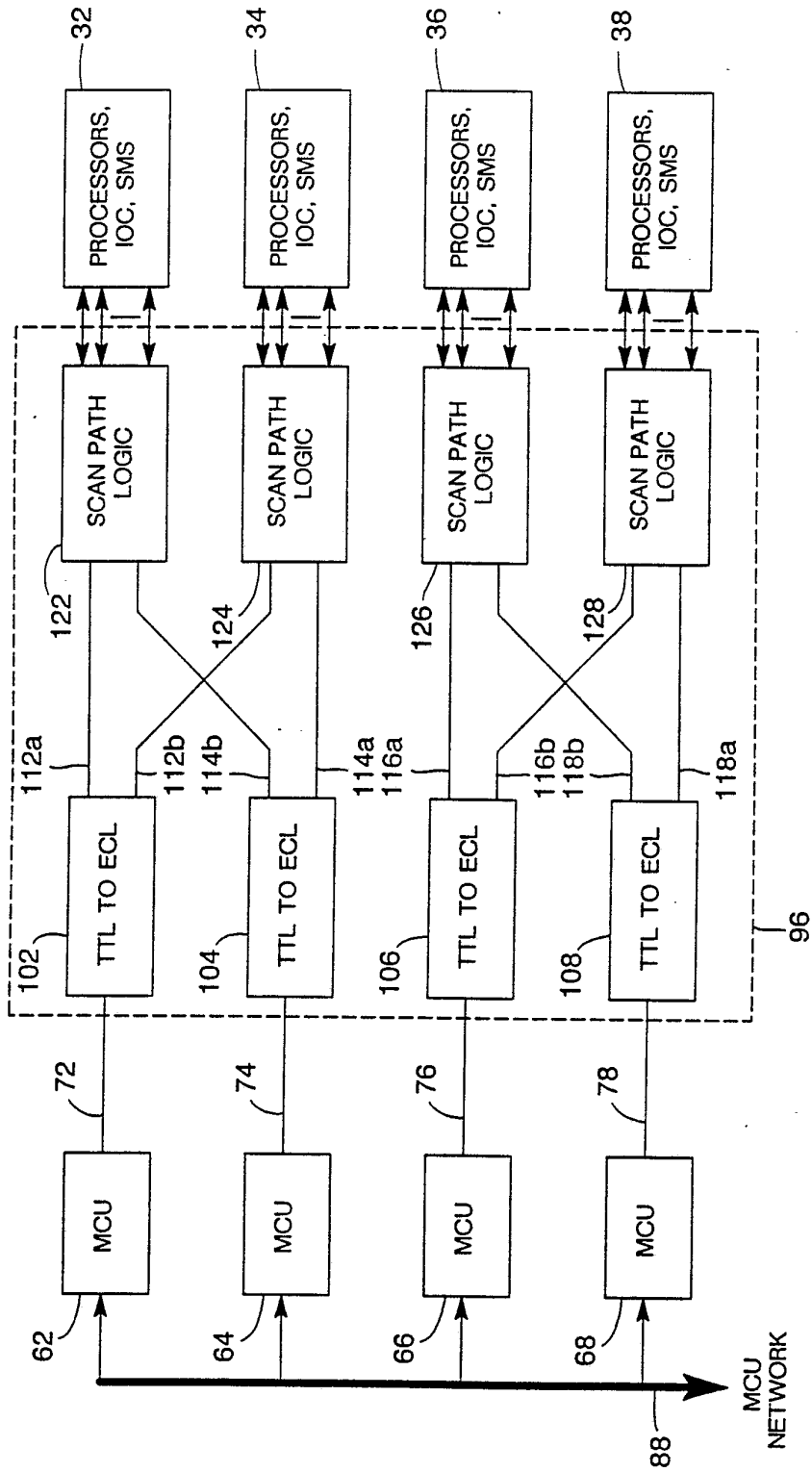


FIG. 3



4/6
FIG. 4

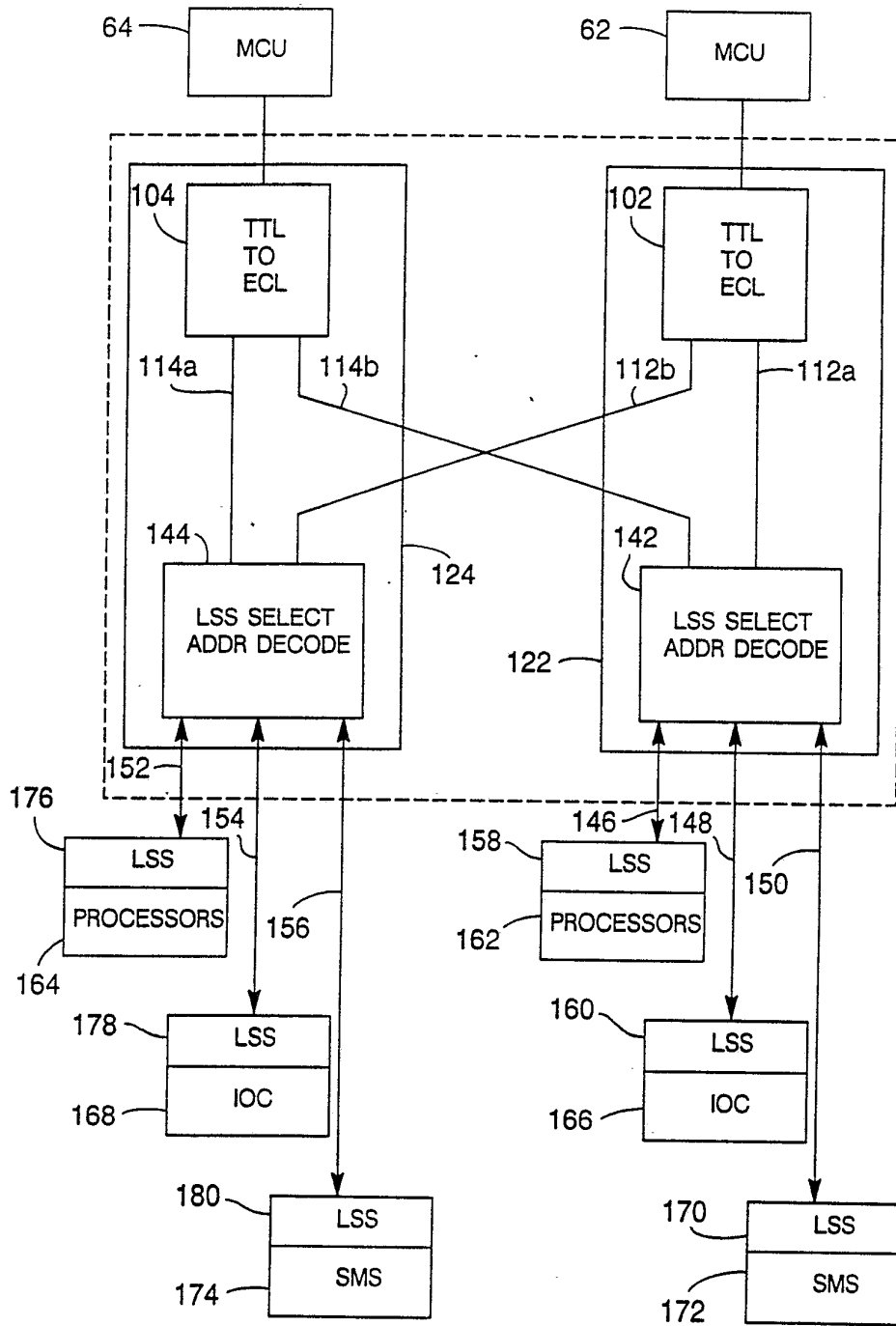


FIG. 5

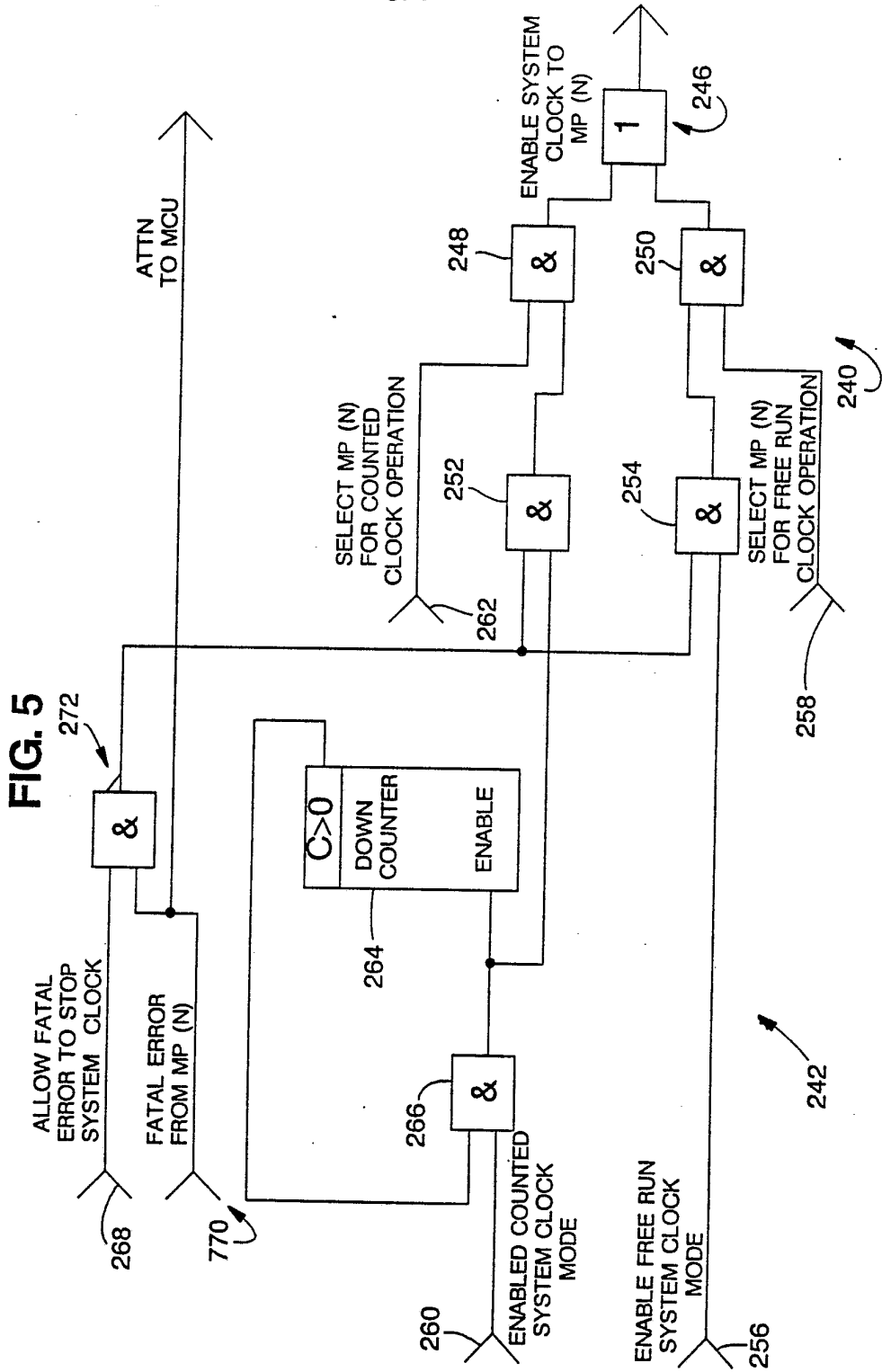
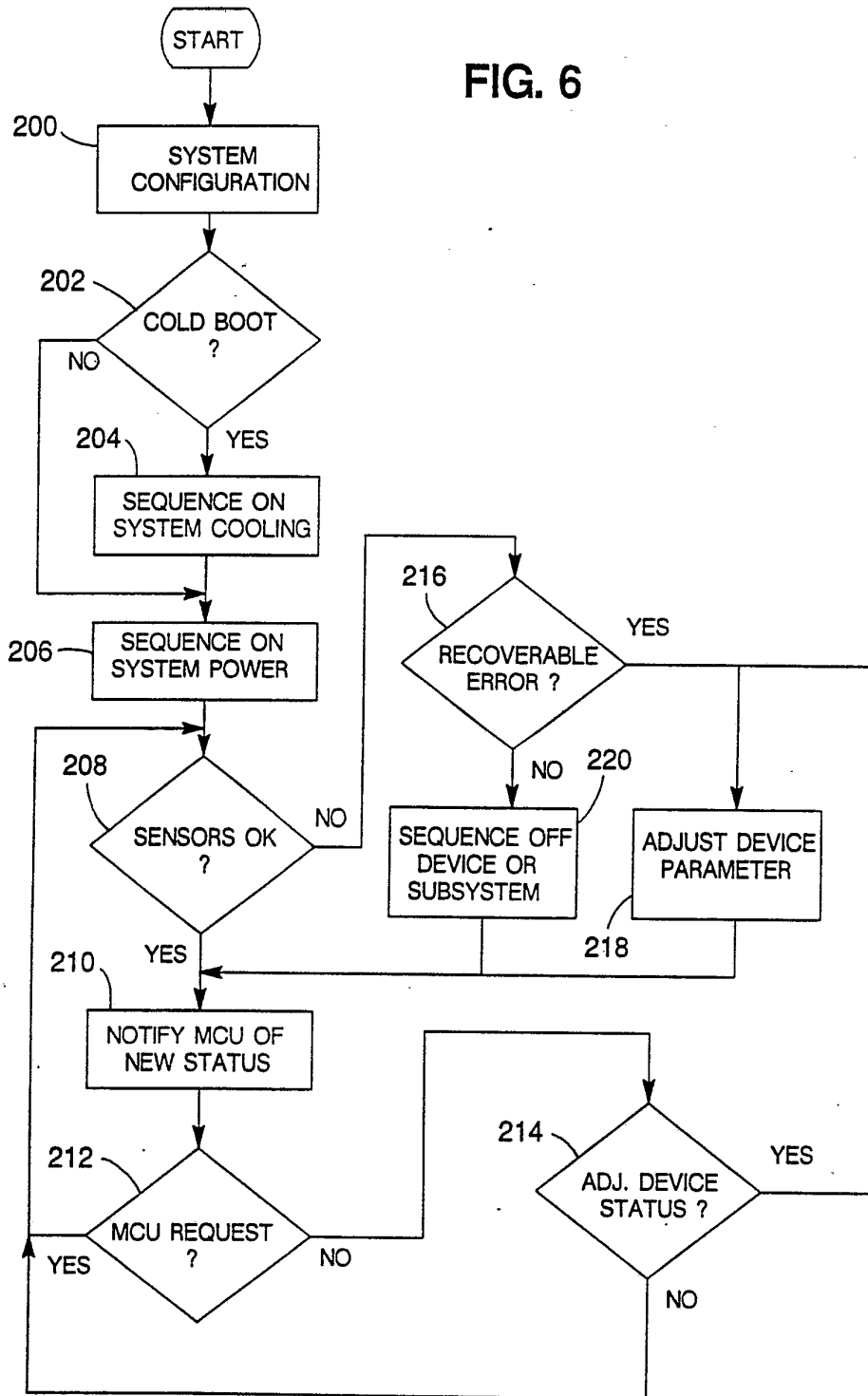
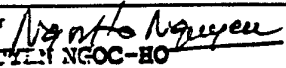


FIG. 6



INTERNATIONAL SEARCH REPORT

International Application No. PCT/US91/04075

I. CLASSIFICATION OF SUBJECT MATTER (if several classifications apply, indicate all) ⁶		
According to International Patent Classification (IPC) or to both National Classification and IPC		
IPC (5): G06F 11/32		
U.S.Cl.: 395/575		
II. FIELDS SEARCHED		
Minimum Documentation Searched ⁷		
Classification System	Classification Symbols	
U.S.Cl.	364/200,900 Ms File 395/575	
Documentation Searched other than Minimum Documentation to the Extent that such Documents are Included in the Fields Searched ⁸		
III. DOCUMENTS CONSIDERED TO BE RELEVANT ⁹		
Category [*]	Citation of Document, ¹¹ with indication, where appropriate, of the relevant passages ¹²	Relevant to Claim No. ¹³
Y	US, A, 4,106,092 (MILLERS) 08 August 1978 (e.g. See col. 11, line 59 - col. 12, line 55).	1-6
Y	US, A, 4,695,946 (ANDREASEN ET AL.) 22 September 1987 (See fig. 10, col. 27, lines 15-28, col. 4, line 2 - col. 6, line 8).	1-6
Y	US, A, 4,439,826 (LAWRENCE ET AL.) 27 March 1984 (e. g. See col. 5, lines 3-52).	1-6
Y	US, A, 3,812,469 (HAUCK ET AL.) 21 May 1974 (e. g. See col. 2, line 59 - col. 7, line 25).	1-6
Y	US, A, 4,891,751 (CALL ET AL.) 02 January 1990 (See fig. 5, and col. 4, lines 10-40).	2
<p>[*] Special categories of cited documents: ¹⁰</p> <p>"A" document defining the general state of the art which is not considered to be of particular relevance</p> <p>"E" earlier document but published on or after the international filing date</p> <p>"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)</p> <p>"O" document referring to an oral disclosure, use, exhibition or other means</p> <p>"P" document published prior to the international filing date but later than the priority date claimed</p> <p>"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention</p> <p>"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step</p> <p>"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.</p> <p>"&" document member of the same patent family</p>		
IV. CERTIFICATION		
Date of the Actual Completion of the International Search		Date of Mailing of this International Search Report
16 September 1991		24 OCT 1991
International Searching Authority		Signature of Authorized Officer
ISA/US		 Eric Coleman NGUYEN NGOC-HO INTERNATIONAL DIVISION