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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
7 September 2001 (07.09.2001)

PCT

(10) International Publication Number
WO 01/64395 A2

(51) International Patent Classification⁷: **B24B 37/00**

(21) International Application Number: PCT/US01/06658

(22) International Filing Date: 27 February 2001 (27.02.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/516,432 1 March 2000 (01.03.2000) US

(71) Applicant: **SPEEDFAM-IPEC CORPORATION** [US/US]; Attention: James L. Farmer, (480) 705-2668, 305 North 54th Street, Chandler, AZ 85226 (US).

(72) Inventors: **NIMTZ, Jack**; 1741 Juniper Lane, S.W., Rochester, MN 55902 (US). **RICKORDS, Gary**; 2524 S. El Paradiso Drive, #11, Mesa, AZ 85202 (US). **FRESEMAN, Mark**; 16101 E. Powderhorn, Fountain Hills, AZ 85268 (US). **SMITH, Randy**; 13568 S.E. 151st Terrace, Clackamas, OR 97014 (US).

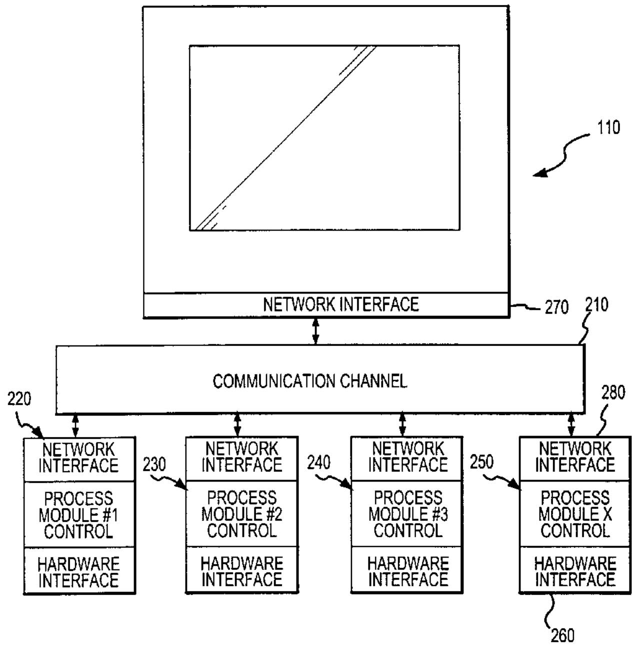
(74) Agent: **FARMER, James, L.**; Speedfam-ipec Corporation, 305 North 54th Street, Chandler, AZ 85226 (US).

(81) Designated States *(national)*: DE, GB, JP, KR, SG.

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.

(54) Title: A MODULAR CONTROL SYSTEM AND METHOD FOR A CMP TOOL



(57) Abstract: A system and method for an automated chemical mechanical planarization (CMP) machine. The system comprises a plurality of interchangeable CMP process modules. Each module is a compilation of grouped software codes called "objects". Process modules may be added or deleted from the system as needed, whether during the machine's assembly or during actual use. The system is configured to allow users to input a process "recipe" for each wafer of a wafer cassette which will guide the system to process the wafer in a desired way.

WO 01/64395 A2

A MODULAR CONTROL SYSTEM AND METHOD FOR A CMP TOOL

Field of the Invention

The present invention relates generally to a system and method for chemical mechanical polishing (CMP) of workpieces and, more particularly, to a modular control system and method for an automated CMP tool.

Background of the Invention

Tools for chemical mechanical planarization (CMP) of workpieces are generally well known. Workpieces, which include but are not limited to, wafers, come in the form of a flat, substantially planar disk. Typical work pieces may include semiconductor wafers, magnetic disks, and optical disks. For many applications, particularly in the area of integrated circuits, the wafer serves as a high-tech building block. In order to produce quality microelectronic devices, it is critical that the wafer surface be manufactured uniform, planar, and devoid of any imperfections.

The manufacture of a semiconductor wafer generally includes a repetitious process of polishing and cleaning the wafer surface. The wafer may go through several material layering steps where one or more dielectrics or metals are "coated" on the wafer surface. After each layer is formed, it is often desirable to thoroughly clean, rinse, and dry the wafer to remove any debris from the surface. Any excess material on the wafer surface which may have accumulated during the layering step is polished off. Polishing the surface also planarizes the wafer. It is often a common practice to polish the wafer in two steps, first, a main polish followed by a buff polish. Of course, even the two step polish may be performed several times for a given wafer application.

Chemical mechanical planarization ("CMP") machines or tools are widely known in the industry. The increasing demand for faster, more consistent wafer planarization has provoked a need for fully automated CMP machines. The automated CMP machine has the capability of performing all of the CMP processes within one enclosed structure under a set of commands. For example, the CMP machine may house the required software and hardware to route, polish, clean, and dry the wafer from start to finish.

The introduction of an automated CMP tool has significantly increased the speed and efficiency of processing wafers in the CMP industry, however, there are several drawbacks to the present CMP automation system. First, due to the nature of being semi to fully automatic, the CMP tool is an interactive combination of software and hardware. At substantially the same time the CMP tool hardware is assembled, complex software is written for the specific processes of that tool. For example, if a tool is being designed with a robotic process, a polishing process, and a cleaning process, unique software is programmed for only those processes. If a need develops to have the machine perform an additional process (e.g., a buff process) after production of the CMP tool, the tool, including the software and the hardware, would likely have to be completely redesigned. For example, the original software code would not have a buff process program and would be rendered obsolete. The addition of new components is often difficult and can effect the performance of existing software. In addition, modifying the existing machine to add additional hardware components would be cumbersome and useless without the corresponding software.

Second, each time a CMP machine is manufactured, considerable time is spent testing the software and hardware components specific to that machine. As is commonly known, writing and testing software code can take enormous amounts of valuable engineering time. Nevertheless, unless the code is tested and retested there is a high probability the code will be unreliable and result in future programming problems.

Next, should a problem or modification arise in the software code of a CMP machine, all of the processes within the machine are halted until the "bug" can be removed. Essentially, a minor problem for one process results in a major problem for the entire system. So, regardless of which process application the specific problem occurs in, all the programmed processes may be affected.

Fourth, due to interdependencies between the software and hardware components, substantially all of the CMP machine manufacturing must be completed at the same developer site. For each new machine or update of an existing machine, software code must be rewritten to coincide with the new processes of the machine. The code must be redeveloped for all aspects of the process from the top level application down to the device drivers. Because the code may be entirely unique to a single CMP machine or process within a machine, building and testing of both the hardware and software necessarily occurs at the same site. Additionally, because the software is generally a single code specific to an individual CMP machine, testing an application within the code requires testing the code on the actual hardware.

Accordingly, there exists a need for an improved automated CMP system and method that lends itself to diversity, modification, and repair, while increasing efficiency and maintaining a high level of reliability.

Summary of the Invention

The present invention overcomes the problems outlined above and provides for an improved modular system and method for chemical mechanical planarization (CMP). More particularly, the present invention provides a distributive system of interchangeable process modules for an automated CMP machine or tool. In a preferred embodiment, the CMP machine may contain a plurality of individual modules, wherein each module contains singular codes that control a specific CMP process or function.

Individual modules may be added or deleted as necessitated by the design of the CMP tool. Each module bears its own code that is easily interfaced with a main control unit or main computer via a standard Ethernet connection. Thereby, system modifications are easily realized during the tool's assembly or during actual use by simply removing or adding modules.

More specifically, in accordance with a preferred embodiment, the distributive CMP system may comprise a plurality of programming objects within the main control unit and the individual modules. Classes may contain process-specific code for a specific mechanism, where the resulting objects are then used together to form the process module. The main control unit may comprise classes to make up the controller code. Eventually, an entire program code for the CMP system may be realized by a combination of all the classes. The classes may be duplicated for application in many different tools which use the same physical component. For example, once a class for a pad conditioner has passed all reliability tests, the class is suitable for implementation as part of a group within any module which uses that type of pad conditioner. This means the same tested class (code) may also be used in a different group for a different application and even a different tool. In most instances, whether a tool is being constructed for the first time or modified with updated processes, code may be reused, thus minimizing the time exhausting task of reprogramming and retesting.

In a particularly preferred embodiment, user commands are received at a user interface application and formed into a set of commands. Essentially, the commands form a "recipe" for processing a wafer within the CMP distributive system. A top level application within the main control unit divides the recipe into sub-recipes which are then transmitted to each corresponding process module. Each process module contains a group of objects wherein each object contains functionality specific to one operation within the module. With respect to a further embodiment of the invention, the sub-recipe is again divided and sent to corresponding objects within the module where the commands will be implemented.

In another embodiment, the distributive system and method may accept a highly individualized recipe for one workpiece or a general recipe for a set of workpieces. The user can identify each workpiece with a distinguishing recipe for processing the workpiece. The system is suitably configured to route the individual recipe along with its corresponding workpiece. This allows for diversity within the CMP system. Specifically, the CMP tool can be configured to process a plurality of workpieces simultaneously with each workpiece being processed differently.

Brief Description of the Drawings

These and other features, aspects, and advantages of the present invention will become better understood with regard to the following description, appended claims, and accompanying drawings where:

Figure 1 illustrates, in perspective view, an automated CMP tool;

Figure 2 illustrates an exemplary modular architecture for a CMP tool;

Figure 3 illustrates, in block format, a process module shown in Figure 2;

Figure 4 illustrates an exemplary object schematic of a computer control unit;

Figure 5 illustrates an exemplary object schematic of a process module;

Figure 6 illustrates an exemplary object oriented software architecture for a CMP tool in accordance with a preferred embodiment of the invention; and

Figure 7 is a flow chart illustrating object oriented programming in a CMP system.

Detailed Description of the Preferred Embodiment

The present invention may be described herein in terms of function block components and various processing steps. It should be appreciated that such functional blocks may be realized by any number of hardware components configured to perform the specified functions. For example, the present invention may employ various integrated circuit components, e.g., memory elements, digital signal processing elements, logic elements, look-up tables, and the like, which may carry out a variety of functions under the control of one or more microprocessors or other control devices. In addition, those skilled in the art will appreciate that the present invention may be practiced in conjunction with any number of data transmission protocols and that the system described herein is merely one exemplary application for the invention.

It should be appreciated that the particular implementations shown and described herein are illustrative of the invention and its best mode and are not intended to otherwise limit the scope of the present invention in any way. Indeed, for the sake of brevity, conventional techniques for signal processing, data transmission, signaling and network control, and other functional aspects of the systems (and components of the individual operating components of the systems) may not be described in detail herein. Furthermore, the connecting lines shown in the various figures contained herein are intended to represent exemplary functional relationships and/or physical couplings between the various elements. It should be noted that many alternative or additional functional relationships or physical connections may be present in a practical communication system.

The present invention relates to an improved modular system and method for automated chemical mechanical planarization (CMP). Although the modular system may be suitable for virtually any industry or machine that uses multiple function software and hardware, the present invention is conveniently described with reference to the CMP industry and more particularly to the fully automated CMP machine 100 exemplified in Figure 1. The automated CMP machine or tool may be applicable to a variety of applications and may be used to process a variety of workpieces. For ease of discussion and for ease of understanding, the tool and the use of the tool will be described in the context of processing semiconductor wafers. Although described and illustrated in the context of semiconductor wafers, it is not intended that the invention be limited to such an illustrative embodiment.

Automated CMP machine or tool 100 comprises a control unit terminal 110 capable of receiving commands from a user. Preferably, the user may select options by "“touching” the terminal 110. This is common to the industry as a “touch screen monitor.” Other selection methods are available and may include a computer mouse (not shown) to “point and click” on desired selections and/or a keyboard. All commands are received, processed, and stored by the circuitry of control unit terminal 110. The control unit supervises all the processes of the CMP machine by running a control program which essentially “automates” the machine.

Advanced CMP machines contain all the required processes to completely clean, polish, dry and route the wafer from start to finish. The automated tool 100, as directed by the control unit 110, will move the wafer or cassette of wafers through a series of processes as prescribed by the user. The methods of operation and apparatus for performing each CMP process are well known in the art and therefore will not be discussed here in great detail, but will instead be briefly recounted in describing the present invention.

With continued reference to Figure 1, a wafer load/unload station 120 is configured to accommodate a plurality of wafers housed in a wafer cassette. A robotic arm 130 removes each wafer from the cassette to an index transfer device 140. Index transfer device 140 contains a plurality of load and unload cups 180. After a wafer is deposited on one of the load cups 180, index transfer device 140 rotates so that robotic arm 130 can continue filling all of the load cups. Each wafer is then carried to a polishing table 150. The wafer is retained against a polishing pad 170, positioned on table 150, by a carrier element (not shown). During polishing, table 150 and the polishing pad 170 disposed thereon rotate about their vertical axes. Simultaneously, the carrier element (not shown) spins the wafer about their respective vertical axes and oscillates the wafer back and forth across the polishing pad 170. In this manner, the surface of the wafer is polished and/or planarized.

After a predetermined period of polishing time, the wafer is lifted from polishing table 150 and transported back to index transfer device 140 and placed into an unload cup. A flipper arm (not shown) removes the wafer from the unload cup and transfers it to a cleaning station 160. Cleaning station 160 preferably comprises several components (not shown) that scrub, rinse, spin and dry the wafer. Each of these components will be discussed further below. Once the wafer has passed through the last cleaning component, the wafer is returned to its original slot in the wafer cassette. Those skilled in the art will appreciate that various other processes and combination of processes may be incorporated within CMP machine 100.

Processing wafers using the CMP machine is largely a series of process steps in which wafers are routed to individual stations. Each station may include a single process step or a combination of steps to perform the desired function. Depending on the wafer application, a particular wafer may be processed differently (i.e; polished longer or even several times) than similar wafers within the same wafer cassette. Diversity among the wafers of the same cassette is largely possible because the machine is divided into individual process modules. Referring now to Figure 2, each process module 220, 230, 240, 250 comprise the necessary software and hardware for performing a single individual CMP process. Any number of process modules may be housed in a particular CMP machine, as depicted by process module number "x" 250. Structural limitations and computer storage capabilities may indeed be the only physical barriers to overcome in designing a CMP with multiple process modules.

Each process module comprises a network interface 280 linking the individual process module 250 to the control unit network interface 270 via a communication channel 210. Those skilled in the art will recognize that any standard Ethernet or the like could be implemented as communication channel 210. Process module 250 receives commands and direction from control unit 110 and also reports back to control unit 110, thereby establishing a two-way communication channel 210.

Process module 250 further comprises a hardware interface 260. Referring to Figures 2 and 3, hardware interface 260 may be illustrated as having a fieldbus board 320 and a motion control board 330. Process module 250 receives commands from control unit 110 via communications channel 210, and in particular through network interface 280. Module 250 also receives input at fieldbus board 320 from a number of report stations strategically positioned about the module. Report stations may include, but are not limited to, a plurality of sensors 360, closed loop controls 370, and open loop controls 380. Sensors 360 may be used to detect the presence or absence of a wafer in a preset location within the module. Other forms of sensors 360 may include interlock and leak sensors. Closed loop controls 370 use the feedback from sensors 360 to correct outputs to various module operations, for example the water valve controls. Unlike the closed loop controls 370, open loop controls 380 do not control through feedback, but instead receive a control voltage or current. Typically, data received at fieldbus board 320 will be relayed to the control unit 110 for processing and storage.

Module 250 directs a plurality of servo motor controls 340 by coupling motion control board 330 to servo controls 340 through standard coupling techniques well known in the art. Only the motors 350 necessary to carry out the desired process of module 250 will be linked to servo controls 340. In other words, process module 250 has a very specific function within the CMP machine. For discussion purposes only, assume module 250 is designed to perform a polishing function. Motion control board 330 would then connect module 250 to only the servo motor controls 340 which control the polishing motors.

The preferred embodiment of the present invention is realized through object oriented programming. Briefly, program codes controlling a specific function are written in small "classes." Classes are then formed into groups (modules) which are essentially stand-alone programs. Each of the modules are then combined to produce a system program or control program. At program start-up, objects are created by the program to carry out the necessary functions. The objects are generally configured to remain in effect until the entire system is shut down. When shut down occurs, objects are deleted and new objects are created at the next start-up. Those skilled in the art will recognize that the classes form the actual written code which in turn creates the objects. However, for ease of understanding, the term "object" as used here forth will denote a created object as well as the class for which the code resulted. Object orientated programming is well known in the programming field and the precise details of the object programming is beyond the scope of this invention.

Figure 4 illustrates an exemplary object schematic of the control program 410 of control unit 110. Control program 410 comprises a plurality of objects including, but not limited to, a data log 415, an alarm log 420, an event log 425, an Automated Reliability Availability and Maintainability (ARAMS) 430, a scheduler 435, a human/machine interface (HMI) 440, a generic equipment model (GEM) 445, and a network interface 270. Those skilled in the art will appreciate that Figure 4 is merely an exemplary illustration of preferable objects and is not intended to limit the present invention in any way. A brief description of the objects depicted in Figure 4 will follow.

The user inputs a set of commands at control terminal 110. The commands are received at the HMI object 440. The set of commands are formed into a "recipe" for processing a wafer or cassette of wafers in the CMP machine. In one embodiment, each wafer of a wafer cassette may have an individualized recipe which will be sent from the control program 410 to each process module as prescribed by its recipe. For example, the user may be prompted to enter the recipe for wafer #1 of cassette #1, then the recipe for wafer #2 of cassette #1, and so on until all the wafers are given a corresponding recipe. The recipe data may comprise a modular routing order which the user chooses. Only that recipe data which pertains to a module will be sent to that module, the remaining recipe parts will be held back until needed.

Referring back to Figure 4, a GEM object 445 is a semiconductor standard interface to a host computer. The host may include a central computer system for the plant or developer site. An ARAMS object 430 may track the equipment state within the CMP machine. For example, the equipment state or processing state may include whether the equipment (module) is in active mode or engineering mode, or whether the module is out of commission for maintenance and not in use at all. A scheduler object 435 maintains the order in which the wafer will be routed to process modules according to the user (i.e.; the recipe.) An event log object 425 receives data from the modules that indicates a process has been completed, or that the desired event has taken place. An alarm log object 420 keeps record of an alarm which may have occurred during the running of the CMP machine. An alarm may include a signal sent from a transporter module (not shown) to the control program 410 to notify the control program 410 that there are no more wafers to transport between process modules. In more serious circumstances, an alarm may notify the control program 410 which in turn notifies the user that robotic arm 130 has broken a wafer. In the latter case, and others, the alarm log object 420 may halt the CMP machine completely. A data log object 415 essentially collects data particular to a wafer. Data sent to the log 415 may comprise the wafer location, the route the recipe followed, and any errors or warnings that may have occurred during the processing of the wafer. The data may then be stored in the control program 410, or preferably, all the runtime data from each of the objects is sent from the datalog 415 to a database (not shown) with greater storage capacity, apart from the CMP machine. Data may still be retrieved from the control program 410 even if it is sent to the database.

Although not depicted in the same manner as the other objects, control network interface 270 and module network interface 280 (Figure 3) are both software objects. The versatility of the software object lends itself well for use as a network interface. Modules may be added to the system via communication channel 210 by simply programming a class to create another object for the newly added module. In a similar manner, modules may be deleted. Thus, modular architecture with object orientated programming allows for simplified modifications to a CMP tool.

The configuration barrel 450 may be coined the "collective brains" of the system. The user can configure the tool by entering a recipe which is correlated in the configuration barrel 450. Individual objects make up the entire system which is then overseen by the configuration 450.

Each of the process modules are comprised of function specific objects. Figure 5 illustrates an exemplary object schematic of process module 250. Figure 6 illustrates an exemplary object oriented software architecture for a CMP tool. Again for sake of clarity, assume process module 250 is a polishing module. Referring to Figures 5 and 6 collectively, module application 510 is a high level object configured to receive the polishing recipe from the control program 410 and more specifically, from the top level application object 620. The recipe is sent to module application 510 at substantially the same time the wafer is placed on the polishing table 150 (Figure 1). The recipe may be sub-divided prior to or after being sent to module application 510. The recipe is divided into a sub-recipe or subset of commands and sent to the mid level objects 520 and 530. For example, sub-application 520 may be a carrier application 642 and sub-application 530 may be a pad conditioner application 644. The sub-recipe will only contain the parameters and/or data necessary for sub-application 520 to perform its coded function. The sub-recipe for carrier application 642 may include, but is not limited to, speed parameters with which to rotate or oscillate the wafer while on polishing table 150, the amount of down-force to apply to the wafer while on polishing table 150, and/or a period of time to polish the wafer. The pad conditioner application 644 may receive a sub-recipe for controlling the pad movement speed and/or the amount of slurry (chemical additive) added during the polishing process. Once the sub-recipes are received in the mid level objects 520 and 530, the sub-recipe may be further divided into a smaller group of sub-commands to be sent to the lower level objects 540 and 550.

Generally, the lower level objects of Figures 5 and 6 are driver applications. In a particular embodiment, driver 1 object 540 may be a motion control driver P 645 as shown on Figure 6. A group of sub-commands sent to motion control driver P 645 may include instructions for movement of servo motors within the module. Driver N object 550 may be an input/output (I/O) device driver P 655 as shown on Figure 6. I/O driver P 655 may receive a group of sub-commands for operation of switches, valves and sensors within the module. As noted by Figure 6, the lower level objects 645, 675, 655 and 680 are enstantiations of a class. At runtime, as many objects as needed to fulfill the recipe will be created. In one exemplary embodiment, two motion control driver objects 645, 675 are created, one for polish application 640 and one for cleaner application 650. The same class (program code) was used (repeated) to create both motion drivers 645, 675. This is an important aspect of object oriented programming. Code which has been tested and found reliable may be used and reused as many times as needed, not only within the same CMP machine, but with other CMP machines. Further, if a "bug" is found in a particular class, that class can be "debugged" and replaced without affecting the rest of the program. As a result, software maintenance downtime will be minimized, and many different operations may use the same class and all those operations benefit from the debug.

In yet another application, if an error exists in one module, the user can reroute the wafer and avoid the faulty process module altogether. In any event, the modular architecture with object orientated programming avoids having to completely shut down the machine during repairs or updates.

Referring back to Figure 3, the lower level objects may be compared to process module hardware interface 260. As previously described, fieldbus board 320 receives inputs from sensors 360, closed loop controls 370, and open loop controls 380. I/O device driver 655 is configured to be the equivalent software required to carry out fieldbus board 320 tasks. I/O device driver 655 receives sensor and valve data specific for each process module in a fashion similar to the way fieldbus board 320 receives the physical counterparts (i.e., the electric currents). Motion control board 330 drives the servo motor controls 340 upon receiving a group of sub-commands in motion control driver 645. Again it is helpful to realize that typically, motion control driver 645 and I/O device driver 655 will only receive a group of sub-commands from one process module.

Figure 6 is an exemplary object oriented software architecture for a CMP tool in accordance with a preferred embodiment of the present invention, wherein each of the clouds represents an object. Arrows at the end of signal lines depict the flow of signals between the objects and those skilled in the art will recognize that any suitable communication means may be used. HMI application 440, as described above in detail, receives the user commands and routes them to a top level application 620. The commands are formed into a recipe for a given wafer or cassette of wafers. The top level application 620 may be compared to the configuration barrel 450 of Figure 4. The recipe is routed back and forth from the top level application 620 to each of the high level objects shown as polish, cleaner, automation and Xth applications 640,650,660, 670. Each of the high level objects may be contained within a process module 220,230,240,250 as shown in Figure 2. The recipe received at each of the high level applications is a process-specific recipe for only the process application to which it is sent.

Automation application 660 comprises programming for transporting or routing the wafer within the CMP machine. The automation application 660 receives commands from the top level application 620 indicating that a wafer is ready for transport. The transport may be to or from the wafer cassette or to or from a process module. Automation application 660 divides the command into sub-commands and sends them down to serial device driver 662 for implementation. Serial device driver 662 communicates to CMP operations that contain serial interfaces, generally this will comprise such functions as communication to the robotic arm 130.

A cleaner application object 650 is within a cleaner module in the preferred embodiment of the present invention. A recipe is suitably received in cleaner application 650 which may comprise a speed with which to scrub a given wafer, the amount of surfactant to use, the amount of de-ionized water to spray on the wafer, a period of time to clean the wafer, and an air speed for drying the wafer. Those skilled in the art will appreciate that further commands may be input to form the cleaner recipe. High level application 650 may divide the accepted recipe into a sub-recipe for mid level objects such as roller box application 652 and spin/rinse/dry (SRD) application 654. As previously discussed for the polishing application 640, the mid level objects may further divide the sub-recipe into a group of commands to be sent to drivers 675, 680.

As further illustrated in Figure 6, the high level object Xth application 670 represents that CMP tools may be assembled with additional process modules and objects to accommodate additional wafer applications. In one exemplary embodiment, a measurement module and application is included in the CMP system. In many cases, it is desirable to "spot check" a given wafer to ensure the recipe delivered is being accurately applied or achieving the desired result. As an example, the user may wish to measure the wafer thickness between the main polish and a buff polish. Measuring every wafer would be inefficient and time consuming; therefore periodic sampling is often done. The user may choose to transport, for example, every fifth wafer from the main polish module to a measure module. The parameters to be measured will be determined by the user and the recipe.

In yet another exemplary embodiment of the present invention, an alignment module may be added to the modular system. Typically, an alignment module and its necessary object programming will be the last module in a CMP machine. In order to perform an alignment operation, wafers comprising a flat circular disk may have a notch cut from the outer edge of the circular disk. The transporter module will receive a command from the top level application to transport the wafer to an alignment module. The wafer will be positioned at the alignment module so that the robotic hardware may grab the wafer and place the wafer back in its original cassette such that its notch is in alignment with the notches of similar wafers. Those skilled in the art will recognize that an alignment module may perform various other methods of aligning that are considered to be with the scope of the invention.

Finally, Figures 4 and 6 illustrate that each of the high level objects may report data to the datalog application object 415. As previously discussed, each of the process modules carries out specific CMP functions that are directed through the object oriented software. The runtime data from each of the modules may be continually sent to the data log 415 within the control program 410 for storage.

Figure 7 is a flow chart illustrating an exemplary object oriented method in a CMP machine for processing a wafer. Figure 7 is merely an example illustrating one embodiment of the present invention and is not intended to limit the scope of the present invention. First, a user accesses 710 a control unit terminal which may be similar to control unit terminal 110 of Figure 1. The user is given a prompt 715 to enter wafer process data. The data may include, but is not limited to, the number of process modules within the CMP tool to route the wafer through, the length of time in each module (i.e.; how long to polish, clean), and any characteristics which may be specific to a process module. Next, the data is formed 720 into a recipe which will guide the system while processing the wafer. The recipe is sent 725 to the first process application as indicated by the recipe. At each process application, the recipe may be divided 730 into a sub-recipe. Each sub-recipe may contain data for controlling a single function within the process module. The sub-recipe is sent 735 to a process sub-application. A typical sub-application may include a wafer carrier application 642 of Figure 6 or spin/rinse/dry (SRD) application 654 of Figure 6. The sub-application is generally coded to perform a precise function of the process such as, for example, moving the wafer while being polished. The sub-application may then further divide 740 the sub-recipe into driver commands. The driver objects may receive the commands to operate a driver. Drivers may include motion control drivers 645, 675 of Figure 6 and I/O drivers 655, 680 of Figure 6. The driver will then complete the requested task 750. Query 755 determines if there are more driver tasks to complete. If there are additional driver tasks, remaining driver commands are sent to the chosen driver and this process is repeated until there are no more driver tasks remaining. If there are no additional driver tasks, query 760 determines if there are additional sub-application tasks to complete. If there are additional sub-application tasks to process, the remaining sub-recipe is sent to the corresponding sub-application until there are no more sub-applications remaining. If there are no additional sub-application tasks, query 765 determines if there are additional process applications to complete. If there are additional processes, the recipe is sent to the appropriate process application until there are no more processes remaining. If there are no more processes to complete, the user is notified 770 that the wafer process is complete and the process ends 775.

The present invention has been described above with reference to preferred embodiments. However, those skilled in the art having read this disclosure will recognize that changes and modifications may be made to the preferred embodiments without departing from the scope of the present invention. These and other changes or modifications are intended to be included within the scope of the present invention, as expressed in the following claims.

We Claim

1. A system for chemical mechanical planarization (CMP) of wafers comprising:
a communication channel;
a control unit comprising:
a user interface configured to receive a plurality of commands from a user, said commands forming a recipe for at least one of said wafers,
a plurality of objects, and
a network interface object linking said control unit to said communication channel,
a plurality of modules, wherein each of said modules is interchangeable and commands an individual CMP process, each of said modules comprising:
a network interface object linking each of said modules to said control unit via said communication channel,
a hardware interface object linking each of said modules to a plurality of module hardware, and
a process program for performing said individual CMP process, said process program being activated by said control unit.

2. The system of claim 1, wherein:
said user interface comprises a touch screen; and
said plurality of objects within said control unit comprises:
an alarm object for recording system alarms,
an event object for recording process occurrences of said modules,
a schedule object to maintain an order in which said control unit will activate said modules, and
a data log object for recording said individual CMP process data from each of said modules.

3. The system of claim 1, wherein said plurality of modules comprises:
a transporting module configured to route said wafers within said system;
at least one polishing module configured to polish and planarize said wafers; and
a cleaning module configured to clean, spin, rinse and dry said wafers.

4. The system of claim 3 further comprising:
a buffing module configured to provide a second polish to said wafers;

and

an alignment module configured to align said wafers in a given order.

5. The system of claim 1, wherein said recipe is transmitted via said communication channel from said control unit to each of said modules commanding said individual CMP processes.

6. The system of claim 5, wherein said recipe comprises:

a routing element for maintaining an order in which said recipe is transmitted to each of said modules;

a subset of commands specific for each of said modules, said subset comprising:

a first sub-recipe for a transporting module which indicates that said at least one wafer is ready for transport within said system;

a second sub-recipe for a polishing module, said second sub-recipe comprising;

an amount of down force to apply to said at least one wafer during polishing,

an amount of slurry to introduce when polishing said at least one wafer,

a rotation and speed of rotation for polishing said at least one of said wafer, and

a length of time for polishing said at least one wafer;

a third sub-recipe for a cleaning module, said third sub-recipe comprising;

a scrubbing speed for cleaning said at least one wafer,

an amount of surfactant to introduce when cleaning said at least one wafer,

an amount of de-ionized water to spray on said at least one wafer,

a length of time for said cleaning module to operate, and

an air speed for drying said at least one wafer.

7. The system of claim 3 further comprising a measuring module which commands a measuring process for measuring at least one parameter of said at least one wafer.

8. The system of claim 7, wherein said one of said plurality of objects comprises a data log object, and said measuring module transmits, via said communication channel, measurement data from said at least one wafer to said datalog object.

9. A method for chemical mechanical planarization (CMP) of a wafer in a distributive system wherein said system comprises a plurality of interchangeable process modules, said method comprising the steps of:

a) receiving said wafer at a load/unload station within a CMP tool;

b) forming a recipe for said wafer from a set of input commands;

c) receiving said recipe at each of said plurality of interchangeable process modules;

d) routing said wafer through each of said process modules at substantially the same time as said recipe is received at each of said modules;

e) repeating steps c) and d) until said wafer has been routed to each of said process modules as specified by said recipe; and

f) delivering said wafer to said load/unload station.

10. The method of claim 9, wherein said step of receiving said recipe occurs first at a transporting module, wherein said recipe contains data for routing said wafer within said system.

11. The method of claim 10, wherein said step of receiving said recipe occurs second at a polishing module, wherein said transporting module routes said wafer to said polish module.

12. The method of claim 9, wherein said method is capable of operating on a plurality of wafers simultaneously, wherein each of said plurality of wafers has a corresponding recipe which is received at each of said modules as each of said wafers is being individually routed through each of said modules.

13. The method of claim 11, wherein said step of receiving said recipe occurs third at a cleaning module, wherein said transporting module routes said wafer to said cleaning module.

14. A method of object oriented programming for use with a plurality of chemical mechanical planarization (CMP) process module applications within a CMP system, said method comprising the steps of:

a) receiving a set of commands from a user at a user interface application;

b) routing said commands to a top level application wherein said commands form a recipe for processing a wafer within said system;

c) routing to each of said process module applications said recipe from said top level application;

d) dividing said recipe at each of said process module applications into a subset of commands, said subset of commands controlling a specific CMP process within each of said process module applications; and

e) routing to a sub application said subset of commands, said sub application including a process program for said specific CMP process.

15. The method of claim 14, further comprising the steps of:

a) dividing said subset of commands into a group of sub-commands; and

b) routing said sub-commands from said sub application to a driver, said driver controlling CMP process hardware necessary to carry out said specific CMP process within each of said process module applications.

16. The method of claim 15, wherein said driver comprises a motion control driver controlling at least one motor within one of said process module applications.

17. The method of claim 15, wherein said driver comprises an I/O device driver.

18. A computer program product for use with a chemical mechanical planarization (CMP) computer having a display device, comprising:

a computer readable medium with a first computer program recorded thereon, the first program including:

a first code section for enabling the computer to display a plurality of user options on a display device;

a second code section for enabling the computer to establish a wafer recipe comprising a plurality of sub-recipes, in response to directions designated by an input device, in such a manner that said wafer recipe specifies a guide;

a third code section for enabling the computer to establish a process order from said recipe;

a fourth code section for enabling the computer to track a process event;

a fifth code section for enabling the computer to record an alarm in response to a signal sent from a process module;

a sixth code section for enabling the computer to store data in response to a data message from said process module; and

a seventh code section for enabling the computer to establish an interface between said first computer program and a communication channel;

a computer readable medium with a second computer program recorded thereon, the second program including:

a first code section for enabling the computer to establish an interface between said second computer program and said communication channel;

a second code section for enabling the computer to select, in turn, successive sub-recipes from said plurality of sub-recipes; and

a third code section for enabling the computer to direct a plurality of process drivers.

19. A modular apparatus useful in a chemical mechanical planarization (CMP) tool, said apparatus configured to receive a substantially planar disk, said apparatus comprising:

a fieldbus board coupled to at least one report station;

a motion control board coupled to at least one servo motor control, said servo motor control coupled to a motor;

a network interface configured to connect said modular apparatus to a control unit through a communication channel, said control unit sending said modular apparatus a command set; and

a process program for directing said apparatus to perform a process function.

20. The apparatus of claim 19 wherein said report station comprises a sensor.

21. The apparatus of claim 19 wherein said report station comprises a loop control.

22. The apparatus of claim 19 wherein said network interface comprises an object oriented programming code.

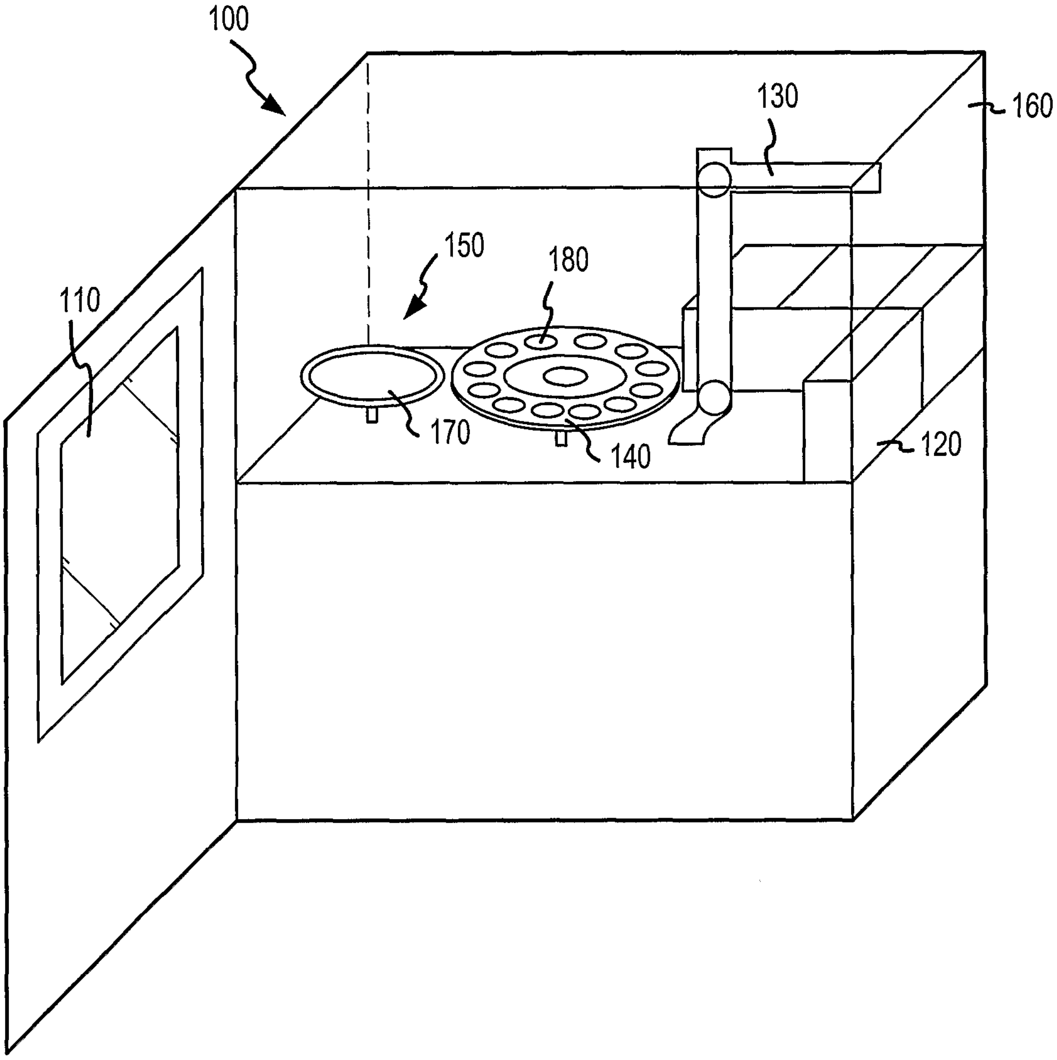


FIG.1

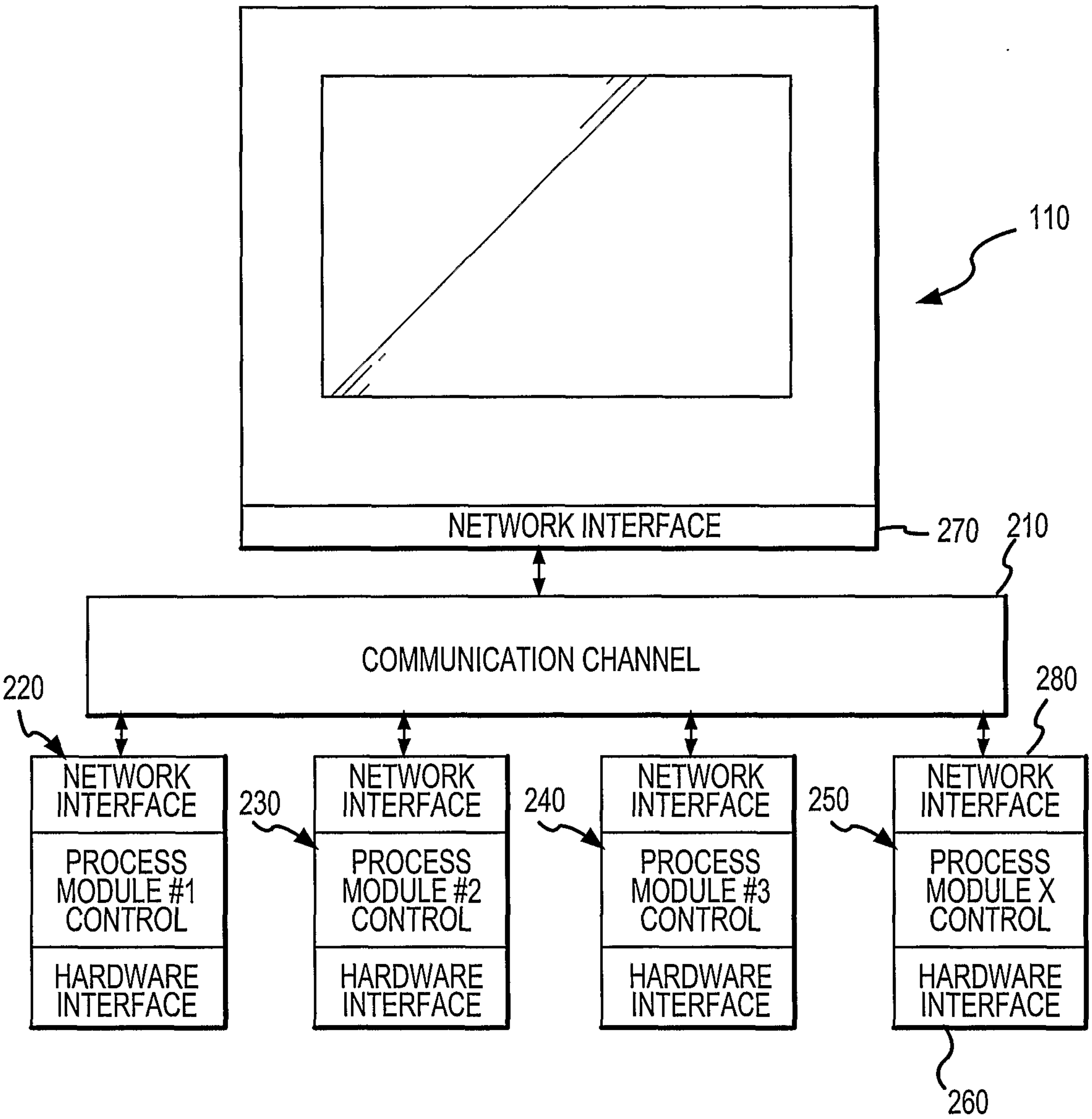

110
NETWORK INTERFACE
270
210
COMMUNICATION CHANNEL
220
280
NETWORK INTERFACE
PROCESS MODULE #1 CONTROL
HARDWARE INTERFACE
230
NETWORK INTERFACE
PROCESS MODULE #2 CONTROL
HARDWARE INTERFACE
240
NETWORK INTERFACE
PROCESS MODULE #3 CONTROL
HARDWARE INTERFACE
250
NETWORK INTERFACE
PROCESS MODULE X CONTROL
HARDWARE INTERFACE
260

FIG. 2

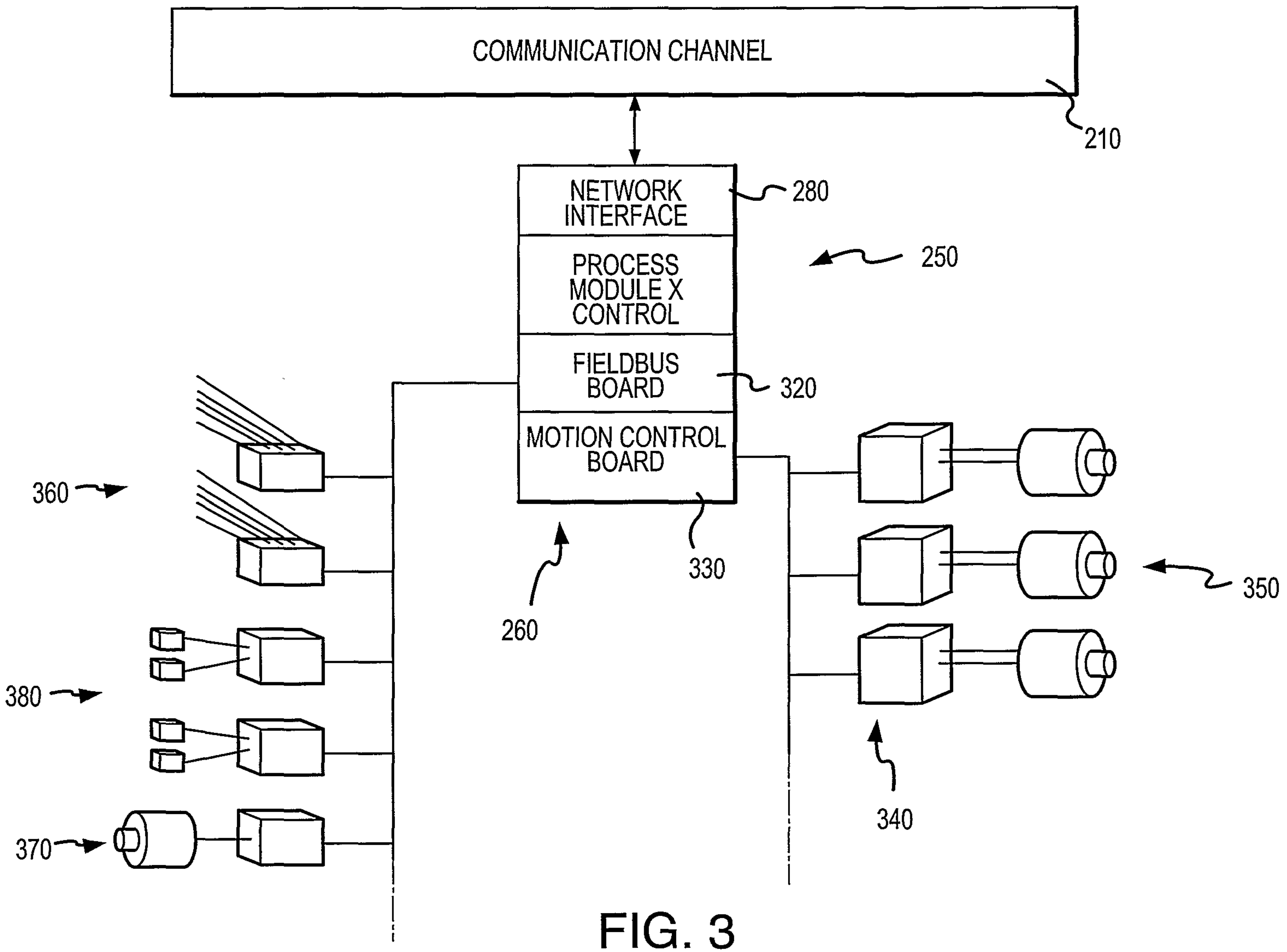

COMMUNICATION CHANNEL
210
NETWORK INTERFACE
280
PROCESS MODULE X CONTROL
250
FIELDBUS BOARD
320
MOTION CONTROL BOARD
330
260
360
380
370
340
350

FIG. 3

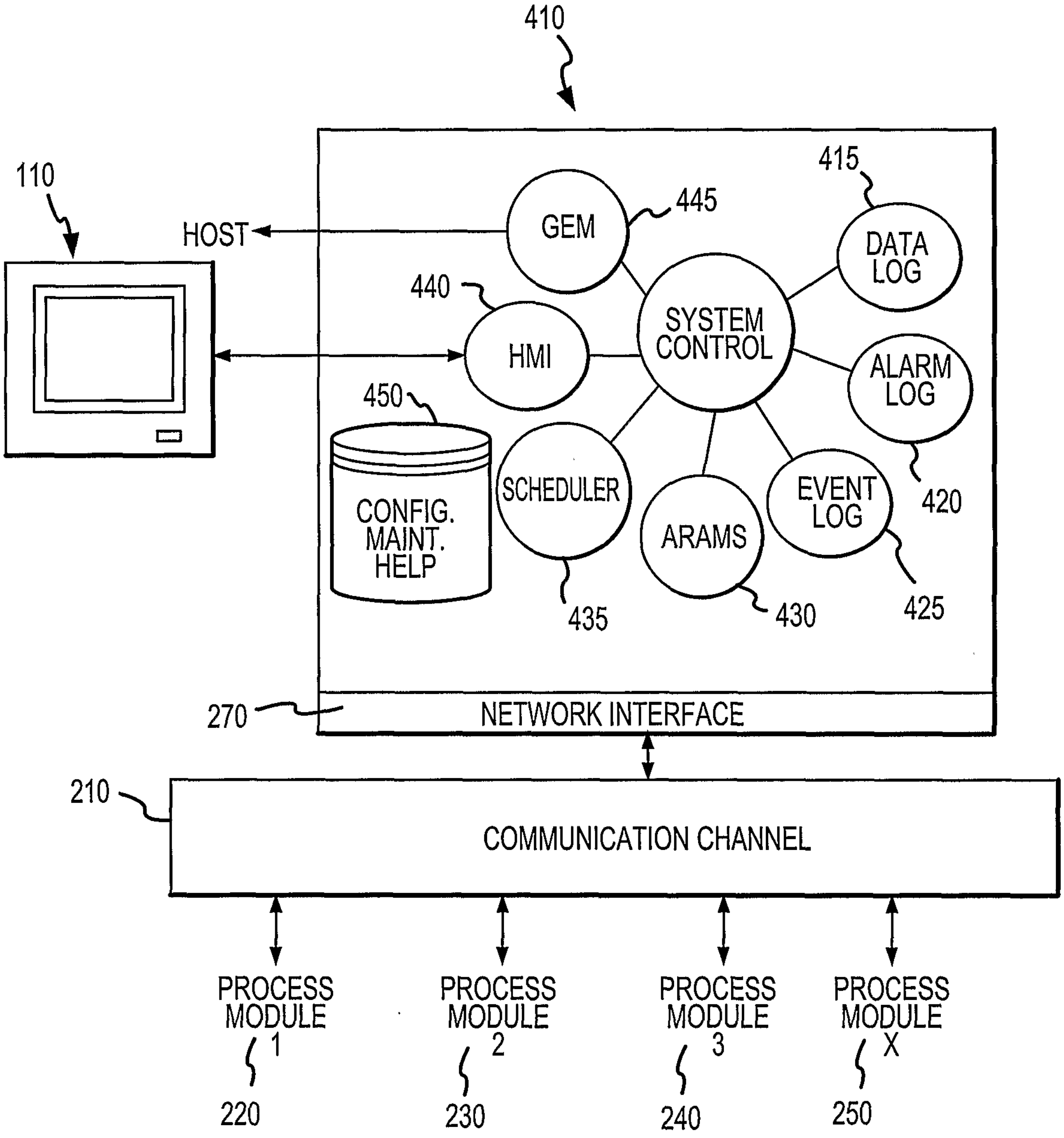

410
110
HOST
GEM
445
415
DATA LOG
440
HMI
SYSTEM CONTROL
ALARM LOG
450
CONFIG. MAINT. HELP
SCHEDULER
ARAMS
EVENT LOG
420
435
430
425
270
NETWORK INTERFACE
210
COMMUNICATION CHANNEL
PROCESS MODULE 1
PROCESS MODULE 2
PROCESS MODULE 3
PROCESS MODULE X
220
230
240
250

FIG. 4

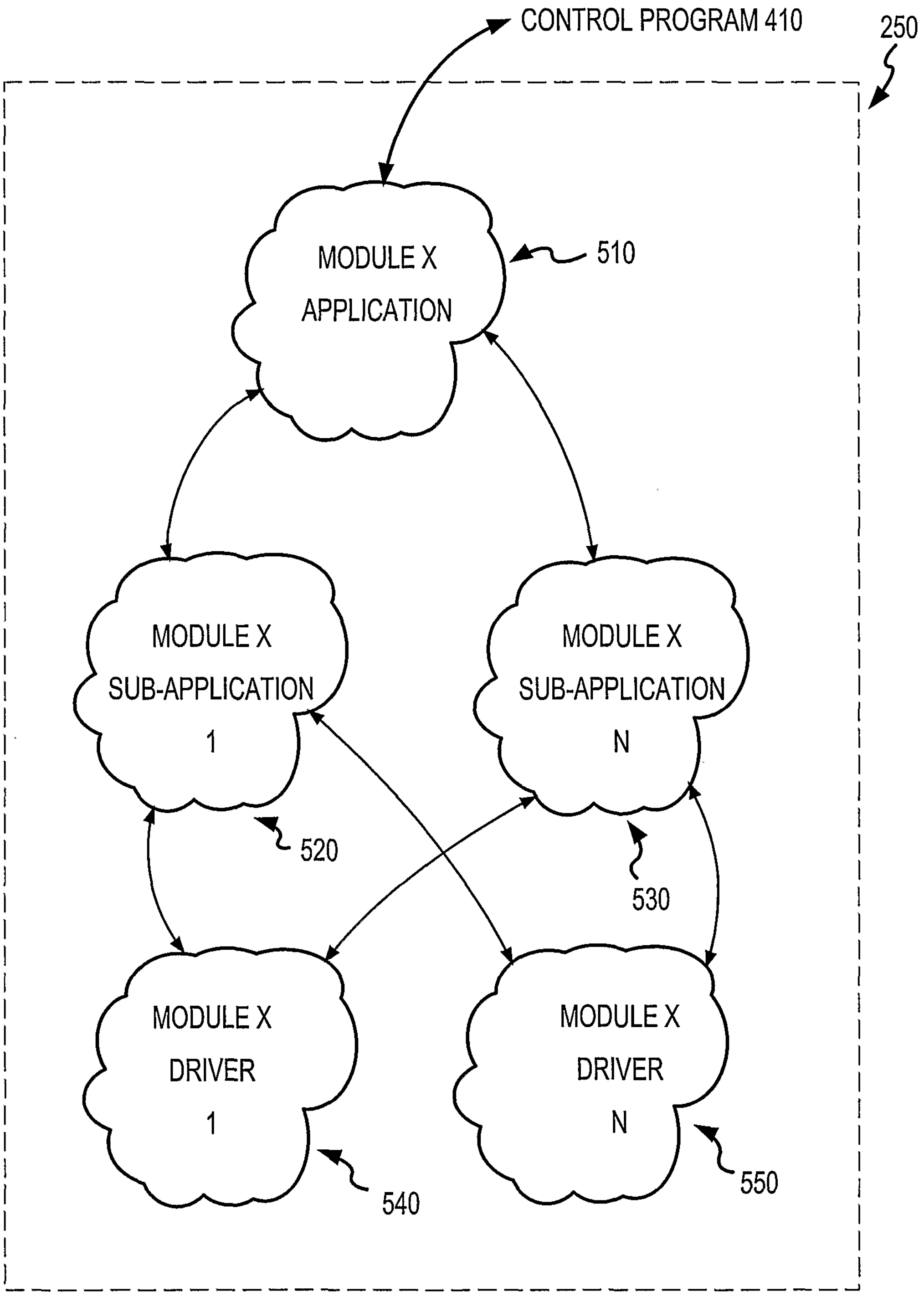

CONTROL PROGRAM 410
250
MODULE X APPLICATION
510
MODULE X SUB-APPLICATION 1
520
MODULE X SUB-APPLICATION N
530
MODULE X DRIVER 1
540
MODULE X DRIVER N
550

FIG. 5

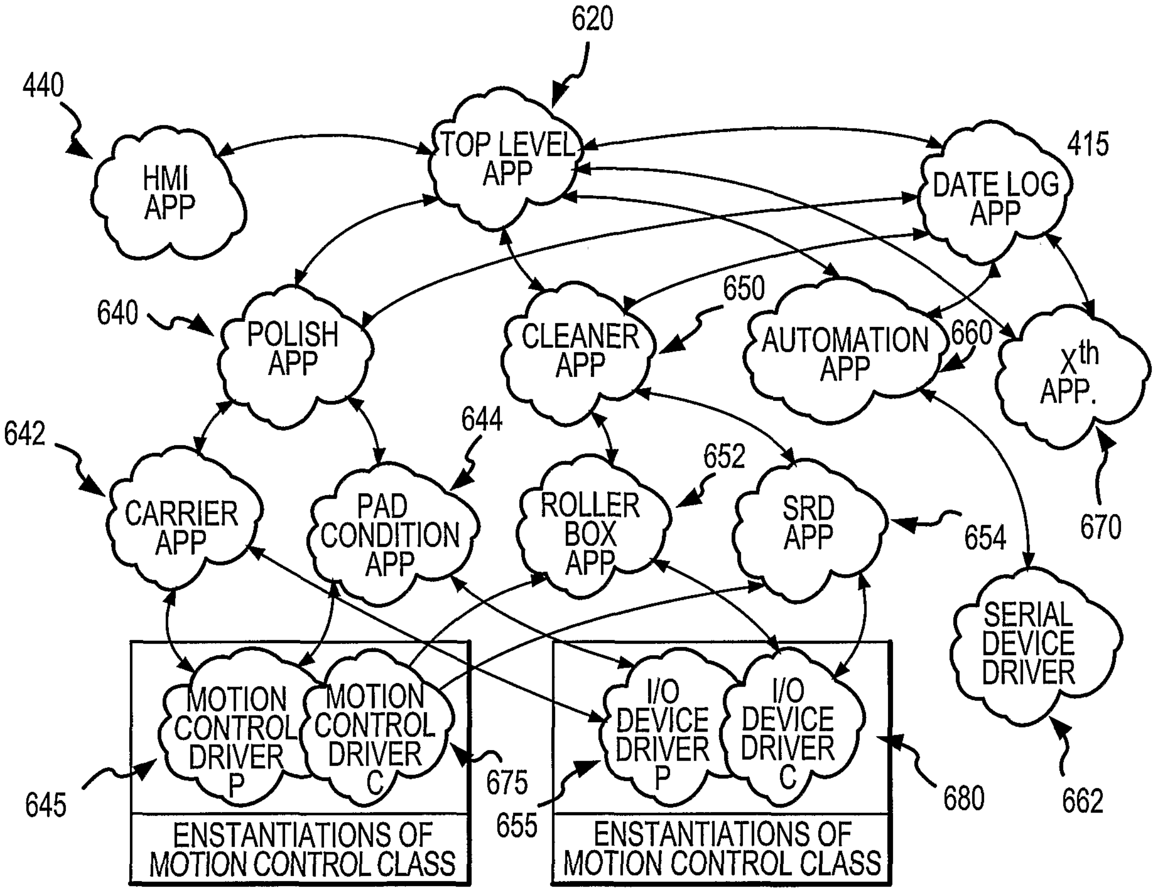
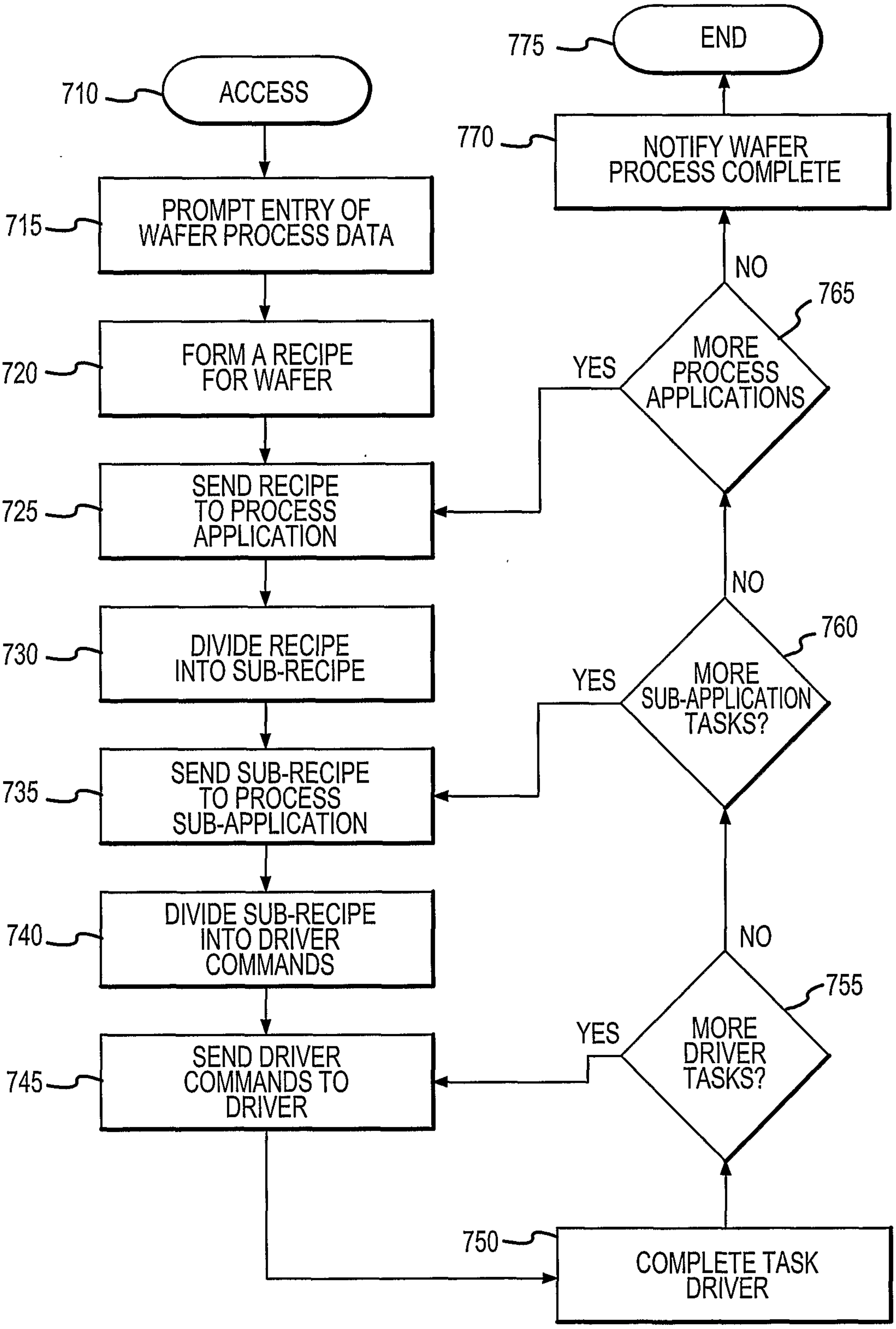

620
440
HMI APP
TOP LEVEL APP
415
DATE LOG APP
640
POLISH APP
CLEANER APP
650
AUTOMATION APP
660
X^th APP.
642
644
CARRIER APP
PAD CONDITION APP
ROLLER BOX APP
652
SRD APP
654
670
SERIAL DEVICE DRIVER
MOTION CONTROL DRIVER P
MOTION CONTROL DRIVER C
675
645
ENSTANTIATIONS OF MOTION CONTROL CLASS
I/O DEVICE DRIVER P
I/O DEVICE DRIVER C
655
680
662
ENSTANTIATIONS OF MOTION CONTROL CLASS

FIG. 6



710
ACCESS
715
PROMPT ENTRY OF WAFER PROCESS DATA
720
FORM A RECIPE FOR WAFER
725
SEND RECIPE TO PROCESS APPLICATION
730
DIVIDE RECIPE INTO SUB-RECIPE
735
SEND SUB-RECIPE TO PROCESS SUB-APPLICATION
740
DIVIDE SUB-RECIPE INTO DRIVER COMMANDS
745
SEND DRIVER COMMANDS TO DRIVER
750
COMPLETE TASK DRIVER
755
MORE DRIVER TASKS?
YES
NO
760
MORE SUB-APPLICATION TASKS?
YES
NO
765
MORE PROCESS APPLICATIONS
YES
NO
770
NOTIFY WAFER PROCESS COMPLETE
775
END

FIG. 7