

ARRANGEMENT FOR IDENTIFYING UNCONTROLLED EVENTS AT THE PROCESS MODULE LEVEL AND METHODS THEREOF

BACKGROUND OF THE INVENTION

[Para 1] Advances in plasma processing have provided for growth in the semiconductor industry. To be competitive, a manufacturing company needs to be able to process the substrates into quality semiconductor devices. Tight control of the process parameters is generally needed to achieve satisfactory results during substrate processing. When the processing parameters (e.g., RF power, pressure, bias voltage, ion flux, plasma density, and the likes) fall outside of a pre-defined window, undesirable processing results (e.g., poor etch profile, low selectivity, damage to the substrate, damage to the processing chamber, and the likes) may result. Accordingly, the ability to identify conditions when the processing parameters are outside the pre-defined windows is important in the manufacture of semiconductor devices.

[Para 2] During substrate processing, certain uncontrolled events may happen that may damage the substrate and/or cause damage to the processing chamber components. To identify the uncontrolled events, data may be collected during substrate processing. Monitoring devices, such as sensors, may be employed to collect data about the various process parameters (such as bias voltage, reflected power, pressure, and the likes) during substrate processing. As discussed herein, sensor refers to a device that may be employed to detect conditions and/or signals of a plasma processing component. For ease of discussion, the term "component" will be used to refer to an atomic or a multi-part assembly in a processing chamber.

[Para 3] The type and amount of data that are being collected by the sensors have increased in recent years. By analyzing the data collected by the sensors in relation to the process module data and the process context data (chamber event data), parameters that are outside of the pre-defined window may be identified. Accordingly, corrective actions (such as recipe adjustment) may be provided to stop the uncontrolled event(s), thereby preventing further damage from occurring to the substrate and/or the processing chamber components.

BRIEF SUMMARY OF THE INVENTION

[Para 4] The invention relates, in an embodiment, to a process-level troubleshooting architecture (PLTA) configured to facilitate substrate processing in a plasma processing system. The architecture includes a process module controller. The architecture also includes a plurality of sensors, wherein each sensor of the plurality of sensors communicates

with the process module controller to collect sensed data about one or more process parameters. The architecture further includes a process-module-level analysis server, wherein the process-module-level analysis server communicates directly with the plurality of sensors and the process module controller. The process-module-level analysis server is configured for receiving data, wherein the data include at least one of the sensed data from the plurality of sensors and process module and chamber data from the process module controller. The process-module-level analysis server is also configured for analyzing the data and sending interdiction data directly to the process module controller when a problem is identified during the substrate processing.

[Para 5] The above summary relates to only one of the many embodiments of the invention disclosed herein and is not intended to limit the scope of the invention, which is set forth in the claims herein. These and other features of the present invention will be described in more detail below in the detailed description of the invention and in conjunction with the following figures.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWINGS

[Para 6] The present invention is illustrated by way of example, and not by way of limitation, in the figures of the accompanying drawings and in which like reference numerals refer to similar elements and in which:

[Para 7] Fig. 1 shows a prior art overall logic view of an interconnecting tool environment with a host-level analysis server.

[Para 8] Fig. 2 shows a simple block diagram of an interconnecting tool environment with a cluster tool level solution for correlating data between the sensors and the process model controllers.

[Para 9] Fig. 3 shows, in an embodiment of the invention, a simple logic overview of a process-level troubleshooting architecture.

[Para 10] Fig. 4 shows, in an embodiment of the invention, a simple functional diagram of a process module level analysis server.

DETAILED DESCRIPTION OF EMBODIMENTS

[Para 11] The present invention will now be described in detail with reference to a few embodiments thereof as illustrated in the accompanying drawings. In the following description, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be apparent, however, to one skilled in the art, that the present invention may be practiced without some or all of these specific details. In

other instances, well known process steps and/or structures have not been described in detail in order to not unnecessarily obscure the present invention.

[Para 12] Various embodiments are described hereinbelow, including methods and techniques. It should be kept in mind that the invention might also cover articles of manufacture that includes a computer readable medium on which computer-readable instructions for carrying out embodiments of the inventive technique are stored. The computer readable medium may include, for example, semiconductor, magnetic, opto-magnetic, optical, or other forms of computer readable medium for storing computer readable code. Further, the invention may also cover apparatuses for practicing embodiments of the invention. Such apparatus may include circuits, dedicated and/or programmable, to carry out tasks pertaining to embodiments of the invention. Examples of such apparatus include a general-purpose computer and/or a dedicated computing device when appropriately programmed and may include a combination of a computer/computing device and dedicated/programmable circuits adapted for the various tasks pertaining to embodiments of the invention.

[Para 13] As aforementioned, to gain a competitive edge, manufacturers have to be able to effectively and efficiently troubleshoot problems that may arise during substrate processing. Troubleshooting generally involves analyzing the plethora of data collected during processing. To facilitate discussion, Fig. 1 shows a prior art overall logic view of an interconnecting tool environment with a host-level analysis server.

[Para 14] Consider the situation wherein, for example, a manufacturing company may have one or more cluster tools (such as etch tools, cleaning tools, strip tools, and the likes). Each cluster tool may have a plurality of processing modules, wherein each processing module is configured for one or more specific processes. Each cluster tool may be controlled by a cluster tool controller (CTC), such as CTC 104, CTC 106, and CTC 108. Each cluster tool controller may interact with one or more process module controller (PMC), such as PMCs 110, 112, 114, and 116. For ease of discussion, examples will be provided in relation to PMC 110.

[Para 15] In order to identify conditions that may require intervention, sensors may be employed to collect data (sensed data) about processing parameters during substrate processing. In an example, during substrate processing a plurality of sensors (such as sensors 118, 120, 122, 124, 126, 128, 130, 132, 134, 136, 138, and 140) may interact with the process module controllers to collect data about one or more processing parameters. The type of sensors that may be available may depend upon the type of data that may be collected. For

example, sensor 118 may be configured to collect voltage data. In another example, sensor 120 may be configured to collect pressure data. Generally, the sensors that may be employed to collect data from a process module may be of different brands, makes, and/or models. As a result, a sensor may have little or no interaction with another sensor.

[Para 16] Usually, a sensor is configured to collect measurement data about one or more specific parameters. Since most sensors are not configured to perform processing, each sensor may be coupled to a computing module (such as a computer, user interface, and the likes). The computing module is usually configured to process the analog data and to convert the raw analog data into a digital format.

[Para 17] In an example, sensor 118 collects voltage data from PMC 110 via sensor cable 144. The analog voltage data received by sensor 118 is processed by a computing module 118b. The data collected by the sensors are sent to a host-level analysis server (such as data box 142). Before sending the data onward to data box 142 over the network connection, the data is first converted from an analog format into a digital format by the computing module. In an example, computing module 118b converts the analog data collected by sensor 118 into a digital format before sending the data over a network path 146 to data box 142.

[Para 18] Data box 142 may be a centralized analysis server that is configured to collect, process, and analyze data from a plurality of sources, including the sensors and the process modules. Usually, one data box may be available to process the data collected during substrate processing by all of the cluster tools of a single manufacturing company.

[Para 19] The actual amount of data that may be transmitted to data box 142 may be significantly less than the amount collected by the sensors. Usually, a sensor may collect a massive amount of data. In an example, a sensor may collect data at rates of up to 1 megabyte per second. However, only a fraction of the data collected by the sensors is sent to data box 142.

[Para 20] One reason for not transmitting the entire data streams collected by the sensors to data box 142 is due to the network bandwidth limitation when using cost-effective, commercially available communication protocols. The network pipeline to data box 142 may not be able to handle large volume of data from a plurality of sources (such as sensors 118, 120, 122, 124, 126, 128, 130, 132, 134, 136, 138, and 140) being sent to a single receiver (such as data box 142). In other words, the network path between the sensor arrangements (sensor and computing module) and data box 142 may experience major traffic congestion as data box 142 tries to receive the massive amount of data coming from all of the sensor

arrangements. As can be appreciated from the foregoing, if data box 142 is unable to handle the incoming traffic, the data packets being sent may be dropped and may have to be resent, thereby putting an additional burden onto the already heavily congested network pipeline.

[Para 21] In addition, data box 142 may not be able to handle a high volume of incoming data from multiple sources while at the same time performing other important functions, such as processing and analyzing data. As aforementioned, data box 142 is not only configured to receive the incoming data packets but data box 142 is also configured to process and analyze all of the incoming data streams, for example. Since data box 142 is the analysis server for the different data streams being collected, data box 142 needs sufficient processing capability to perform analysis on the plethora of data streams.

[Para 22] Since data box 142 has limited processing resource, only a fraction of the data collected from each sensor is sent to data box 142. In an example, of the thousands of data items that may be collected by a single sensor, only 10-15 data items at 1-5 hertz may be forwarded to data box 142. In one example, only a summary of the data collected by sensor 118 may be sent to data box 142.

[Para 23] In addition to receiving data from the plurality of sensors, data box 142 may also be receiving data from the process module controllers. In an example, process module data and process context data (chamber-event data) may be collected by each process module controller and forwarded to data box 142. For ease of discussion, process module data and process context data may also be referred to as process module and chamber data. For example, process module data and process context data may be collected by PMC 110 and be sent to CTC 104 via a path 148. CTC 104 is not only managing the data from PMC 110 but may also be handling the data from the other processing module controllers within the cluster tool (such as PMC 112, PMC 114, and PMC 116).

[Para 24] The data collected by the cluster tool controller is then transmitted to a fab host 102 via a semiconductor equipment communication standard/generic equipment module (SECS/GEM) interface. In an example, CTC 104 transmits data collected from PMCs 110, 112, 114, and/or 116 to fab host 102 through SECS/GEM 156 via a path 150. Fab host 102 may not only be receiving data from CTC 104, but also may be receiving data from other cluster tool controllers, such as CTCs 106 and 108, for example. The data collected by fab host 102 is then forwarded to data box 142 via a path 158. Due to the sheer volume of data being collected, not all data being sent to fab host 102 is forwarded to data box 142. In many instances, only a summary of the data may be transmitted to data box 142.

[Para 25] Data box 142 may process, analyze and/or correlate the data collected by the sensors and the process module controllers. If an anomaly is identified, data box 120 may then determine the source of the problem, such as a parameter that is not in conformance with a recipe step being performed in PMC 110, for example. Once the source of the problem has been identified, data box 142 may send an interdiction in the format of an Ethernet message to fab host 102. Upon receiving the message, fab host 102 may forward the message through SECS/GEM 156 to CTC 104. The cluster tool controller may then relay the message to the intended process module controller, which is PMC 110 in this example.

[Para 26] Unfortunately, the interdiction is usually not provided in real-time. Instead, the interdiction is usually received by the intended process module after the affected substrate has been processed or even after the entire substrate lot has exited the process module. Accordingly, not only have the substrate/substrate lot been damaged, but one or more processing chamber components may have also been negatively impacted, thereby increasing waste and increasing ownership cost.

[Para 27] One reason for the delay is due to the sheer volume of data being received from a plethora of sources. Even if data box 142 may be configured with a fast processor and have sufficient memory to handle the large volume of data streams, data box 142 may still need time to process, correlate and/or analyze all of the data being collected.

[Para 28] Another reason for the delay in receiving the interdiction by the process module is due to the incomplete data streams that are being received by data box 142. Since data box 142 is receiving data from a plethora of sources, the actual data that is being sent to data box 142 is significantly less than the data being collected. In an example, instead of sending the 1 gigahertz data stream that is being collected by sensor 118, only a fraction (about 1-5 hertz) of the data is actually being sent. As a result, even though data box 142 is receiving a high volume of data from all of its sources, the data that is being received is usually incomplete. Thus, determining an uncontrolled event may take time given that data box 142 may not have access to the complete data set from all sources.

[Para 29] In addition, the paths by which the data are being sent to data box 142 may vary. In an example, data are sent directly from a sensor arrangement (that is sensor and its computing module) after the analog data has been converted into digital data. In contrast, the data collected by the process module is transmitted over a longer network path (through at least cluster tool controller and fab host). Accordingly, data box 142 is unable to complete its analysis until all related data streams have been received.

[Para 30] Not only is the network path between a process module and data box 142 longer but the data streams sent through this path are usually faced with at least two bottlenecks. The first bottleneck is at the cluster tool controller. Since the data collected by the process modules within a cluster tool is being sent to a single cluster tool controller, the first bottleneck occurs since the data streams from the various process modules have to be processed through a single cluster tool controller. Given the sheer volume of data that can be transmitted from each process module, the network path to the cluster tool controller usually experiences heavy traffic congestion.

[Para 31] Once the data has been received by the cluster tool controller, the data is transmitted to fab host 102. The second bottleneck may occur at fab host 102. Given that fab host 102 may be receiving data from various cluster tool controllers, traffic into fab host 102 may also be experiencing congestion due to the high volume of data being received.

[Para 32] Since data box 142 needs the data from the different sources in order to determine an uncontrolled event, the traffic condition between a process module and data box 142 prevents timely delivery of the data streams to data box 142. As a result, precious time is lost before data box 142 has gathered all the necessary data to perform analysis. Furthermore, once an interdiction is prepared, the interdiction has to travel through the same lengthy path back to the affected process module before the interdiction can be applied to perform corrective action.

[Para 33] Another factor contributing to the delay is the challenge of correlating data from the various data sources. Since the data streams being received by data box 142 is usually a summary of the data collected from each sensor and/or process modules, correlating the data may be a challenging task since the data streams available may be of different time intervals. In an example, the selected data streams transmitted to data box 142 from sensor 118 may be at a one second interval while the data streams from PMC 110 may be at a two second interval. As a result, correlating data streams may require time before an uncontrolled event may be definitively determined.

[Para 34] An additional challenge for correlating the data is due to the different paths by which the data are being sent to data box 142. As the data is being transmitted through different computers, servers, and the likes, the data may be exposed to computer drift, network latency, network loading and the likes. As a result, data box 142 may have difficulty correlating the data from the various sources. Given that a tight correlation is required to quickly identify uncontrolled events, more analysis may be required to be performed before an uncontrolled event may be accurately identified.

[Para 35] Another disadvantage of the solution provided in Fig. 1 is the cost of ownership. In addition to the cost of maintaining a cluster tool system, additional cost is associated with the sensor arrangements. Since each sensor may be of different brands/makes/models, each sensor arrangement usually includes a sensor and a computing module. Physical space is usually required to house each of the sensor arrangements. Accordingly, the cost of housing the sensor arrangement can become pricey, especially in areas in which real estate prices can be high.

[Para 36] To reduce the actual time delay between the actual occurrence of the uncontrolled event within the process module and the receipt of the interdiction by the process module, a cluster-level analysis server is provided. Fig. 2 shows a simple block diagram of an interconnecting tool environment with a cluster-tool-level solution for correlating data between the sensors and the process model controllers.

[Para 37] Similar to Fig. 1, a cluster tool may include a plurality of process modules (such as PMCs 210, 212, 214, and 216). To collect data for analysis, each process module may be coupled to a plurality of sensors (such as sensors 218, 220, 222, 224, 226, 228, 230, 232, 234, 236, 238, and 240). Each sensor may interact with its corresponding process module controller via a sensor cable (such as sensor cable 244) to collect processing parameter data. The data collected by the sensor may be in analog format. A computing module (such as computing module 218b) may process and convert the data into a digital format before forwarding the data via a path 246 to a cluster-level analysis server (such as remote controller 242).

[Para 38] Similar to Fig. 1, each process module controller may also transmit data (such as process module data and process context data) to a cluster tool controller (such as CTC 204 and 206). In an example, data collected by PMC 210 may be transmitted to CTC 204 via a path 248. Besides receiving data from PMC 210, CTC 204 may also be receiving data from the other processing module controllers (such as PMC 212, 214, and 216). The data received by the cluster tool controller is then forwarded via a path 250 to a fab host 202.

[Para 39] Between fab host 202 and CTC 204, a serial tap may be connected to network path 250 to duplicate the data being forwarded to fab host 202. In an example, a serial tap 208 may intercept the data being forwarded by CTC 204 to fab host 202. The data is duplicated and a copy of the data stream is sent to remote controller 242 via a path 254. If the fab host is connected to more than one cluster tool controller, then for each cluster tool controller, a dedicated remote controller is associated with the cluster tool controller. In an example, the data being sent from CTC 206 to fab host 202 via a path 252 is intercepted by

another serial tap (256). The data is duplicated and sent via a path 258 to a remote controller (260) that is different than the remote controller (242) associated with CTC 204.

[Para 40] Hence, instead of a single data box to handle all the data from the various cluster tools, multiple remote controllers may be available to handle the data from the various cluster tools. In other words, each cluster tool is associated with its own remote controller. Since each remote controller is handling data from a fewer number of data sources (such as the process module controllers and the sensors associated with a single cluster tool), each remote controller is able to handle a higher volume of data from each source. In an example, instead of 30-100 data items being sent, about 40kB – 100kB data items at 10 hertz may now be received by each remote controller.

[Para 41] Data received from the sensors and the process module controllers are analyzed by the remote controller. If a problem is identified, the remote controller may send an interdiction to the cluster tool controller. In an example, remote controller 242 identifies a problem within PMC 210. An interdiction is sent via paths 254 and 250 through serial tap 208 to CTC 204. Upon receiving the interdiction, CTC 204 forwards the interdiction to the intended process module controller, which is PMC 210 in this example.

[Para 42] Since the remote controller is only responsible for handling data from one cluster tool instead of a plurality of clusters tools (as being done by data box 142), more data may be analyzed and better correlation may exist between the different data sets. As a result, the remote controller may perform better and faster analysis, thereby providing more timely intervention to correct an uncontrolled event within a processing module. In an example, instead of receiving an interdiction to prevent an identified uncontrolled event from happening in the next substrate lot (such as the interdiction provided by data box 142), the interdiction sent by remote controller 242, for example, may enable the process engineers to salvage at least part of the substrate lot that is scheduled to be processed.

[Para 43] Although the remote controller solution is a better solution than the data box solution, the remote controller solution still depends upon summary data to perform its analysis. As a result, problems that may be occurring during substrate processing may remain unidentified. Further, the path between the process module and the remote controller is still not a direct path. As a result, computer drift, network latency, and/or network loading may cause time discrepancy that may make it difficult for the remote controller to correlate the data from the sensors with the data from the process modules.

[Para 44] Thus, even though the remote controller solution has increased the timeliness of the interdiction, the remote controller solution is still inadequate. At best, the interdiction

may be able to prevent a problem experienced by the affected substrate from occurring during the processing of the next substrate. In a fiercely competitive market where cost needs to be minimized, waste due to damaged substrate and/or downtime due to damaged processing chamber components may translate into market loss. Accordingly, a real-time solution for identifying uncontrolled event is desired.

[Para 45] In accordance with embodiments of the present invention, a process-level troubleshooting architecture (PLTA) is provided in which troubleshooting is performed at the process module level. Embodiment of the invention includes a process-level troubleshooting architecture that provides for real-time analysis with real-time interdiction. Embodiment of the invention further includes arrangements for load balancing and fault tolerance between sensors.

[Para 46] In an embodiment of the invention, the process-level troubleshooting architecture is a network system in which an analysis server is communicating with a single processing module and its corresponding sensors. In an embodiment, the information being exchanged in the network is bidirectional. In an example, the analysis server may be continually receiving process data from the processing module and sensors. Conversely, the sensors may be receiving data from the processing module and the processing module may be receiving instructions from the analysis server.

[Para 47] Consider the situation wherein, for example, a substrate is being processed. During substrate process, a plurality of data may be collected. In an example, data about pressure is collected every 100 milliseconds. If the processing takes one hour, 36,000 data items have been collected for the pressure parameter. However, a plurality of other process data (e.g., voltage bias, temperature, etc.), besides pressure data, may also be collected. Thus, a considerable amount of data is being collected by the time the substrate process has completed.

[Para 48] In the prior art, the data are transmitted to an analysis server that may be configured to service data collected from a plurality of processing modules (such as remote controller 242 of Fig. 2) if not from a plurality of cluster tools (such as data box 142). Since the data streams are coming from a plurality of sources, time is required to analyze and/or correlate the data. Further, since the analysis server of the prior art may not be able to process and analyze all of the data collected, only a fraction of data collected from each source is transmitted to the analysis server. As a result, the complex task of coordinating, processing, correlating, and/or analyzing the data streams requires time that may not always be readily available.

[Para 49] In one aspect of the invention, the inventors herein realized that a more accurate and quicker analysis may be performed if more granular data is available for analysis. In order to analyze more data from a single source, the analysis server has to be analyzing data from fewer sources. In an embodiment, an arrangement is provided for processing and/or analyzing data at a process module level. In other words, a process-module-level analysis server is provided for performing analysis for each process module and its corresponding sensors.

[Para 50] In an embodiment, the process-module-level analysis server includes a shared memory backbone that may include one or more processors. Each processor may be configured to interact with one or more sensors. In an example, data collected by sensor 1 may be processed by processor 1 while data collected by sensor 2 is processed by processor 2.

[Para 51] Unlike the prior art, the processors may share its processing power with one another to perform load balancing and fault tolerance. In the prior art, a computing module is configured to handle the data collected by a sensor. Since each computing module is an individual unit and usually does not interact with one another, load balancing is usually not performed. Unlike the prior art, the set of processors within the process-module-level analysis server may perform load balancing. In an example, if processor 1 is experiencing data overload while processor 2 is receiving little or no data, processor 2 may be recruited to assist processor 1 in processing the data from sensor 1.

[Para 52] Furthermore, in the prior art, if a computing module is malfunctioning, other computing modules is unable to take over the processing performed by the malfunctioning computing module since the computing modules tend to be of different brands/makes/models. Unlike the prior art, workload may be redistributed between the processors as needed. For example, if processor 2 is unable to perform its function, the workload may be redistributed to other processor until processor 2 is fixed. As can be appreciated from the foregoing, the processors eliminate the need for individual computing modules, thereby also reducing the physical space required to house the computing modules.

[Para 53] In an embodiment of the invention, the processors may be divided into two types of processors: primary processor and secondary processor. Both primary and secondary processors are configured to handle data from sensors. In an example, if secondary processor 1 is associated with sensor 1 then secondary processor 1 usually only process data coming from sensor 1. Likewise, if secondary processor 2 is associated with

sensors 2 and 3, then secondary processor 2 usually only process data coming from those two sensors (2 and 3).

[Para 54] In an embodiment, the shared memory backbone may include one or more primary processors. The set of primary processors may be configured not only to handle data from the sensors but may also be configured to handle data coming from the processing module. In addition, the set of primary processors is configured to correlate the data between the various sources (such as the sensors and processing module) and perform analysis. If an interdiction is needed, the set of primary processors is configured to send the interdiction to the process module controller.

[Para 55] The features and advantages of the present invention may be better understood with reference to the figures and discussions that follow.

[Para 56] Fig. 3 shows, in an embodiment of the invention, a simple logic overview of a process-level troubleshooting architecture. Although a manufacturing company may have more than one cluster tool, a single cluster tool is used as an illustration of one embodiment of the invention. Although a cluster tool may have a varying number of processing modules, the example illustrated in Fig. 3 include a single cluster tool with four processing modules.

[Para 57] The data collected by each processing module is collected by its corresponding processing module controllers (PMC 306, PMC 308, PMC 310, and PMC 312) and transmitted to a fab host 302 via a cluster tool controller (CTC) 304. The data that may be transmitted by the PMCs may be the same type of data (process module data and process context data) that has been previously sent in the prior art. Unlike the prior art, the data being transmitted to fab host 302 is not relied upon by the processing modules to perform troubleshooting. Instead, the data may be archived and be made available for future analysis.

[Para 58] In an embodiment, a process-module-level analysis server (APECS 314) is provided to perform the analysis needed for troubleshooting. Consider the situation wherein a substrate is being etched in PMC 308. During substrate processing, sensors 316, 318, and 320 are collecting data from PMC 308. In an example, sensor 316 is configured to collect voltage bias data from PMC 308. Analog data collected from PMC 308 is sent via sensor cable 328 to sensor 316. Likewise, sensors 318 and 320 may be collected data via sensor cables 330 and 332, respectively. The data collected by the sensors may then be transmitted via one of the paths 322, 324, and 326 to APECS 314 for processing and/or analysis.

[Para 59] Unlike the prior art, data collected by the sensors do not have to be preprocessed (such as summarized, for example) before being transmitted to the analysis server (APECS 314). In an embodiment, instead of having a computing module to process

the data, each sensor may include a simple data converter that may be employed to convert the analog data into digital data before forwarding the data to APECS 314. Alternatively, a data converter, such as a field-programmable gate array (FPGA) may be built into APECS 314, in an embodiment. In an example, each processor may include a data converter algorithm for converting the data into a digital format as part of its processing. As can be appreciated from the foregoing, by eliminating the need for a computing module, less physical space is required to house the cluster tool and its hardware. As a result, the cost of ownership may be reduced.

[Para 60] Since APECS 314 is dedicated to processing data only from one processing module and its corresponding sensors, APECS 314 is able to handle a higher volume of data coming from a single source. In other words, instead of having to pare down the volume of data transmitted from each sensor, APECS 314 is configured to handle most, if not all, of the data collected by each sensor. In an example, instead of just 10-15 data items being sent for analysis, now two thousands plus data items from each sensor may be available for analysis by APECS 314. As a result, the data stream that is available for APECS 314 to process and analyze is a more complete data set.

[Para 61] In an embodiment, APECS 314 is also configured to handle the data coming from the processing module. Unlike the prior art in which the data stream is sent through a lengthy data path through various servers (e.g., cluster tool controller, fab host, etc.) before being received by the analysis server (such as the data box or the remote controller), the data collected by the process module is sent directly to APECS 314 without having to go through other servers. In an example, process module data may be sent from PMC 308 to APECS 314 via a path 334. If an uncontrolled event is identified, an interdiction may be sent directly to PMC 308 via a path 336 without having to go through other servers first.

[Para 62] Further details about the process module level analysis server are provided in Fig. 4. Fig. 4 shows, in an embodiment of the invention, a simple functional diagram of a process-module-level analysis server. A process-module-level analysis server (such as APECS 400) may be assigned to each process module. APECS 400 is a bi-directional server and is configured for processing incoming data and for sending interdictions when uncontrolled events are identified.

[Para 63] Data sources may flow from two main sources, data collected by sensors and data collected by a process module. In an embodiment, APECS 400 is configured to receive incoming data from a plurality of sensors (sensors 410, 412, 414, 416, 420, 422, 424, and 426). Given that some cluster tool owners may have already invested a considerable amount

of money into the traditional sensor arrangement (sensor with a computing module), APECS 400 is configured to accept data from both the traditional sensor arrangements and the modified sensors (sensor that does not require a computing module).

[Para 64] In an embodiment, APECS 400 may include an interface, such as Ethernet switch 418, for interacting with traditional sensor arrangements (such as sensors 410, 412, 414, and 416). In an example, data collected by sensor 410 is first converted from an analog format into a digital format by computing module 410b before the digital data is transmitted to APECS 400 (via paths 430, 432, 434, or 436). Ethernet switch 418 is configured to interact with the traditional sensor arrangements to accept the data streams. The data streams are then passed (via paths 446, 448, 450, or 452) to one of the processors (402, 404, 406, and 408) within APECS 400 for processing.

[Para 65] Instead of utilizing a traditional sensor arrangement for measuring process parameters, a modified sensor (one without a computing module) may be employed. Since the data collected does not have to be summarized, a computing module is no longer required for processing. Instead, a modified sensor may include a data converter (not shown), such as an inexpensive FPGA, for converting data from an analog format to a digital format, in an embodiment. Alternatively, instead of installing a data converter within the sensors, a data converter (not shown) may be installed within APECS 400. Regardless if the data converter is installed externally or internally to APECS 400, the elimination of the computing module provides a cost saving in the ownership of the cluster tool. In an example, the cost to purchase, house, and maintain the computing module is substantially eliminated.

[Para 66] In an embodiment of the invention, APECS 400 include a set of processors (402, 404, 406, and 408) for handling the incoming data. The set of processors may be physical processing units, virtual processors, or a combination thereof. Each processor is responsible for handling the data streams from the sources associated with the processor. In an example, data streams flowing in from sensor 422 via a path 440 are handled by processor 404. In another example, data streams collected by sensor 424 are transmitted to processor 406 via a path 442 for processing.

[Para 67] The number of processors and its relationship with the sensors may depend upon a user's configuration. In an example, even though Fig. 4 only shows a one-to-one relationship between the processors and the sensors, other relationships may exist. In an example, a processor may be configured to handle data from more than one source. In another example, more than one processor may be configured to handle data streams from one sensor.

[Para 68] Each of the processors shares a shared memory backbone 428, in an embodiment. As a result, load balancing may be performed when one or more processors are overloaded. In an example, if the data streams flowing in from sensor 426 via a path 444 is overwhelming processor 408 processing capability, other processors may be recruited to help reduce the load on processor 408.

[Para 69] Besides load balancing, a shared memory backbone also provides an environment for fault tolerance. In other words, if one of the processor is not working properly, the processing previously supported by the malfunctioning processor is redistributed to the other processors. In an example, if processor 406 is not functioning properly and is unable to process the data streams coming from sensor 424, processor 404 may be directed to handle the data streams from sensor 424. Accordingly, the ability to redistribute the workload enables the improperly functioning processor to be replaced without incurring downtime for the entire server.

[Para 70] In an embodiment, two types of processors may exist within APECS 400. The first type of processors is a secondary processor (such as processor 404, 406, or 408). Each secondary processor is configured to process the data streams received from its corresponding sensor(s). Additionally, each processor is configured to analyze the data and to identify any potential problem that may exist with the corresponding sensor(s), in an embodiment.

[Para 71] The second type of processor is known as a primary processor (402). Although Fig. 4 only shows one primary processor, the number of primary processors may depend upon user's configuration. In an embodiment, a primary processor may be configured to handle data streams from one or more sensors. In an example, data streams collected by sensor 420 are sent via a path 438 to primary processor 402 for processing.

[Para 72] Another source of data for a primary processor is a process module. In other words, the process module data and the process context data collected by a process module is processed by the primary processor. In an example, data collected by a process module is sent through a process control bus via a path 454 to APECS 400. The data first traverses through Ethernet switch 418 before flowing via path 446 to primary processor 402.

[Para 73] In addition to processing data, the primary processor is also configured to analyze data from multiple sources. In an example, data correlation between data streams from sensors 422 and 424 is performed by primary processor 402. In another example, data correlation between data streams from one or more sensors with data streams from a process module is also performed by primary processor 402.

[Para 74] Since the data paths for each of the data sources are now of about similar length, correlating the data is significantly less challenging than that experienced in the prior art. In an example, since the data flow from the process module to APECS 400 without having to go through other servers (such as a cluster tool controller and/or a fab host), the data streams from the process module does not experience changes due to computer and/or network conditions (such as computer drift, network latency, network loading and the likes) that may have occurred when the data streams have to be transmitted through other servers (such as a cluster tool controller, a fab host, and the likes) as described in Fig. 1 and Fig. 2. In addition, the wait time for receiving all of the relevant data streams require to perform correlation and analysis is now significantly reduced. Thus, correlating data from different sources is significantly simplified when external conditions (such as computer drift, network latency, network loading and the likes) have been substantially eliminated.

[Para 75] Besides the data path, quicker and more accurate analysis may be performed since a higher volume of data with more granularities from a single source provides more data points for performing correlation. In the prior art, correlation between data sources is usually difficult because the data that is available for analysis is usually incomplete since the prior art analysis server is unable to handle a high volume of data from a plethora of data sources. Unlike the prior art, the number of data sources is significantly reduced since each analysis server is now only responsible for analyzing data from a limited number of sources (the process module and the sensors associated with the process module). Since the number of data sources has been significantly reduced, the analysis server has the capacity to handle a higher volume of data from a single source. Given that more granular details are provided, better correlation may be achieved between the data streams of the various sources.

[Para 76] If a problem (such as an uncontrolled event) is identified, primary processor is configured to send an interdiction to the process module. In an embodiment, a direct digital output line 456 is employed to send an interdiction from APECS 400 to the process module. With a direct digital output line between the two devices, the interdiction does not have to be first converted into an Ethernet message before the interdiction can be transmitted. Accordingly, the time required to properly format the interdiction and then convert it back is substantially eliminated. Thus, APECS 400 is able to provide real-time interdictions or near-real time interdictions to the process module to handle the uncontrolled event.

[Para 77] In an embodiment, a primary processor may also be configured to interact with other devices via a path 458. In an example, if a cluster tool controller sends a request to APECS 400, the request may be sent via path 458 and be handled by primary processor.

402. In another example, notification to the fab host may be sent via path 458 and the cluster tool controller.

[Para 78] As can be appreciated from one or more embodiments of the present invention, a process-level troubleshooting architecture is provided. By localizing the analysis server at the process module level, data granularity is provided for analysis resulting in a quicker and more accurate analysis. With a similar data path for the various data sources, better correlation exists between the various data streams. With quicker and more accurate analysis, troubleshooting may be performed on more timely basis with the interdiction provided in a timely manner to provide corrective action that may be employed to not only prevent the next substrate from being damaged but also to provide corrective action to fix the uncontrolled event impacting the affected substrate, thereby saving the affected substrate from being damaged. Thus, fewer numbers of substrates are wasted and damages to the processing chamber components may be substantially reduced.

[Para 79] While this invention has been described in terms of several preferred embodiments, there are alterations, permutations, and equivalents, which fall within the scope of this invention. Although various examples are provided herein, it is intended that these examples be illustrative and not limiting with respect to the invention.

[Para 80] Also, the title and summary are provided herein for convenience and should not be used to construe the scope of the claims herein. Further, the abstract is written in a highly abbreviated form and is provided herein for convenience and thus should not be employed to construe or limit the overall invention, which is expressed in the claims. If the term "set" is employed herein, such term is intended to have its commonly understood mathematical meaning to cover zero, one, or more than one member. It should also be noted that there are many alternative ways of implementing the methods and apparatuses of the present invention. It is therefore intended that the following appended claims be interpreted as including all such alterations, permutations, and equivalents as fall within the true spirit and scope of the present invention.

CLAIMS

1. A process-level troubleshooting architecture configured to facilitate substrate processing in a plasma processing system, comprising:

a process module controller configured to control operation of the plasma processing system, the process module controller further configured to collect and provide first data, wherein the first data corresponds to chamber event data associated with the plasma processing system;

a plurality of sensors, wherein first sensors of said plurality of sensors collect second data from said process module controller, wherein the second data corresponds to sensed data about process parameters of said process module controller sensed by said first sensors, wherein said first sensors transmit said sensed data directly to said process module controller without preprocessing by a computing module; and

a process-module-level analysis server that is located remotely from the process module controller and the plurality of sensors, wherein said process-module-level analysis server communicates directly with said first sensors and said process module controller, wherein said first sensors transmit said second data collected from said process module controller directly to said process-module-level analysis server in real-time, and wherein said process-module-level analysis server is configured for

receiving data, wherein said data includes (i) said second data from said first sensors and (ii) said first data from said process module controller, and wherein said second data and said first data are separately received by said process-module-level analysis server from said first sensors and said process module controller, respectively,

analyzing said received data, and

sending interdiction data directly to said process module controller when a problem associated with operation of the process module controller is identified during said substrate processing, wherein said process-module-level analysis server is configured to identify said problem based on said first data received from said process module controller and said second data collected from said process module controller by said first sensors.

2. The process-level troubleshooting architecture of claim 1 wherein a second sensor of said plurality of sensors includes a computing module for performing preprocessing and packetizing of said sensed data before sending at least a portion of said second data to said process-module-level analysis server.
3. The process-level troubleshooting architecture of claim 2 wherein said process-module-level analysis server includes an Ethernet switch, wherein said Ethernet switch enables said process-module-level analysis server to communicate with said second sensor.
4. The process-level troubleshooting architecture of claim 1 wherein at least one of said first sensors converts said second data from an analog format to a digital format before sending said second data directly to said process-module-level analysis server.
5. The process-level troubleshooting architecture of claim 1 wherein said process-module-level analysis server includes a data converter for converting from an analog format into a digital format said second data sent by at least one of said first sensors.
6. A process-level troubleshooting architecture configured to facilitate substrate processing in a plasma processing system, comprising:

a process module controller configured to control operation of the plasma processing system, the process module controller further configured to collect and provide first data, wherein the first data corresponds to chamber event data associated with the plasma processing system;

a plurality of sensors, wherein first sensors of said plurality of sensors collect second data from said process module controller, wherein said second data corresponds to sensed data about process parameters of said process module controller sensed by said first sensors, wherein said first sensors transmit said sensed data directly to said process module controller without preprocessing by a computing module; and

a process-module-level analysis server that is located remotely from the process module controller and the plurality of sensors, wherein said process-module-level analysis server communicates directly with said first sensors and said process module controller, wherein said first sensors transmit said sensed data collected from said process module controller directly to said process-module-level analysis server in real-time, and wherein said process-module-level analysis server is configured for

receiving data, wherein said received data includes (i) said second data from said first sensors and (ii) said first data from said process module controller, and wherein said second data and said first data are separately received by said process-module-level analysis server from said first sensors and said process module controller, respectively,

analyzing said received data, and

sending interdiction data directly to said process module controller when a problem associated with operation of the process module controller is identified during said substrate processing, wherein said process-module-level analysis server is configured to identify said problem based on said first data received from said process module controller and said second data collected from said process module controller by said first sensors;

wherein said process-module-level analysis server includes a shared memory backbone, wherein said shared memory backbone is accessed by a plurality of processors, wherein each processor of said plurality of processors is configured at least for interacting with one sensor of said plurality of sensors through said shared memory backbone.

7. The process-level troubleshooting architecture of claim 6 wherein said shared memory backbone is configured to perform load balancing among said plurality of processors.

8. The process-level troubleshooting architecture of claim 7 wherein said shared memory backbone is configured to provide fault tolerance among said plurality of processors.

9. The process-level troubleshooting architecture of claim 6 wherein at least one processor of said plurality of processors is a secondary processor, wherein said secondary processor is configured at least for receiving said second data from at least one sensor of said plurality of sensors and for processing said second data.

10. The process-level troubleshooting architecture of claim 9 wherein at least one processor of said plurality of processors is a primary processor, wherein said primary processor is configured at least for performing one of

receiving said first data from said process module controller,

receiving said second data from at least one sensor of said plurality of sensors, correlating said first data with said second data, and

sending substantially real-time interdiction data to said process module controller via a direct digital output line.

11. A system configured to perform evaluation during substrate processing in a plasma processing system, comprising:

a plurality of processors for processing data, wherein said data includes (i) sensed data collected by and received from a plurality of sensors in real-time and (ii) chamber event data collected by a process module controller, wherein the sensed data corresponds to data about processing parameters of said process module controller sensed by said plurality of sensors, and wherein said sensed data and said chamber event data are separately received by said plurality of processors from said plurality of sensors and said process module controller, respectively,

wherein said system is located remotely from and is coupled directly to said process module controller and said plurality of sensors, and wherein first sensors of said plurality of sensors transmit said sensed data directly to said system without preprocessing by a computing module;

wherein said processing includes at least one of
receiving said sensed data,
analyzing said sensed data,
performing data correlation on said sensed data, and

sending interdiction data to said process module controller when a problem associated with operation of the process module controller is identified, wherein said system is configured to identify said problem based on said chamber event data received from said process module controller and said sensed data collected from said process module controller by said plurality of sensors;

a shared memory backbone, wherein said shared memory backbone is accessed by said plurality of processors;

a network interface, wherein said network interface is configured at least for receiving said sensed data; and

a direct digital output line, wherein said direct digital output line is a communication path between said plurality of processors and said process module controller.

12. The system of claim 11 wherein said network interface is configured for receiving sensed data from a second sensor of said plurality of sensors, wherein said second sensor includes a computing module for performing preprocessing before sending at least a portion of said sensed data to said plurality of processors via said network interface.

13. The system of claim 12 wherein said network interface is an Ethernet switch.

14. The system of claim 11 wherein at least one of said first sensors includes a data converter for converting said sensed data from an analog format into a digital format before sending at least a portion of said sensed data to said plurality of processors.

15. The system of claim 14 wherein said data converter is field-programmable gate array (FPGA).

16. The system of claim 11 wherein said shared memory backbone is configured to perform load balancing among said plurality of processors.

17. The system of claim 11 wherein said shared memory backbone is configured to provide fault tolerance among said plurality of processors.

18. The system of claim 11 wherein each processor of said plurality of processors is configured for communicating directly with at least one sensor of said plurality of sensors.

19. The system of claim 18 wherein at least one processor of said plurality of processors is a secondary processor, wherein said secondary processor is configured at least for receiving sensed data from at least one sensor of said plurality of sensors and to process said sensed data.

20. The system of claim 19 wherein at least one processor of said plurality of processors is a primary processor, wherein said primary processor is configured at least for performing one of

- receiving said process module and chamber data from said process module controller,

- receiving said sensed data from at least one sensor of said plurality of sensors,

- correlating said sensed data and said chamber event data, and

- sending said interdiction data to said process module controller via a direct digital output line.