



(51) International Patent Classification:

G05B 19/418 (2006.01) G06N 20/00 (2019.01)
H01L 21/67 (2006.01)

(21) International Application Number:

PCT/US2021/033747

(22) International Filing Date:

21 May 2021 (21.05.2021)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

16/881,370 22 May 2020 (22.05.2020) US

(71) Applicant: **APPLIED MATERIALS, INC.** [US/US];
3050 Bowers Avenue, Santa Clara, California 95054 (US).

(72) Inventor: **RHEE, Adrian**; 3050 Bowers Avenue, Santa Clara, California 95054 (US).

(74) Agent: **PORTNOVA, Marina et al.**; LOWENSTEIN SANDLER LLP, One Lowenstein Drive, Roseland, New Jersey 07068 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ,

CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, IT, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

- as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))
- as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))

(54) Title: PREDICTIVE WAFER SCHEDULING FOR MULTI-CHAMBER SEMICONDUCTOR EQUIPMENT

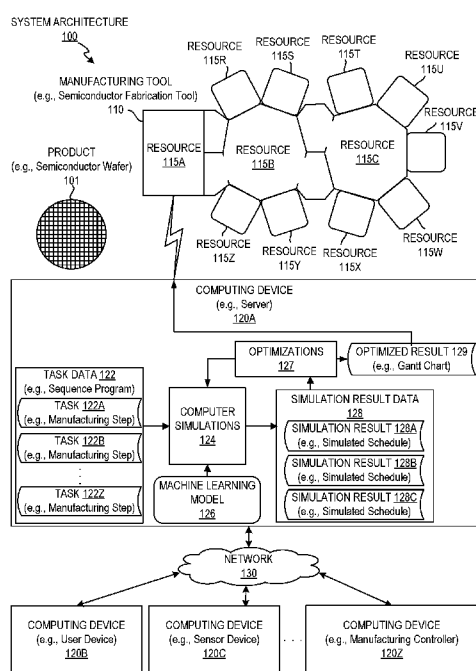


FIG. 1

(57) Abstract: Disclosed herein is technology for performing a simulation based optimization to identify a schedule for a manufacturing tool. An example method may include determining, by a processing device, resources of a manufacturing tool, wherein the resources comprise a first chamber and a second chamber; accessing task data indicating a first manufacturing task and a second manufacturing task; determining a duration to perform the first manufacturing task using the first chamber and a duration to perform the second manufacturing task using the second chamber; updating a machine learning model based on the duration to perform the first manufacturing task and the duration to perform the second manufacturing task; performing a set of computer simulations that uses the machine learning model and the task data to produce a set of simulation results; storing, by the processing device, a simulation result of the set of simulation results in a data store.

Published:

- *with international search report (Art. 21(3))*
- *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

PREDICTIVE WAFER SCHEDULING FOR MULTI-CHAMBER SEMICONDUCTOR EQUIPMENT

TECHNICAL FIELD

[001] The present disclosure relates to predictive modeling, and, more particularly, to a simulation based optimization technique for scheduling manufacturing resources.

BACKGROUND

[002] Manufacturing involves the creation of products using steps that may include human labor, machines, or a combination thereof. The manufacturing steps may be associated with settings that determine when, where, or how the steps are performed and effect the products being manufactured. Process engineers typically schedule manufacturing resources that are used to perform the manufacturing steps and the schedule may be based on their industry experience and past experiments.

SUMMARY

[003] The following is a simplified summary of the disclosure in order to provide a basic understanding of some aspects of the disclosure. This summary is not an extensive overview of the disclosure. It is intended to neither identify key or critical elements of the disclosure, nor delineate any scope of the particular implementations of the disclosure or any scope of the claims. Its sole purpose is to present some concepts of the disclosure in a simplified form as a prelude to the more detailed description that is presented later.

[004] In an aspect of the disclosure, a method includes determining, by a processing device, resources of a manufacturing tool, wherein the resources comprise a first chamber and a second chamber; accessing task data indicating a first manufacturing task and a second manufacturing task; determining a duration to perform the first manufacturing task using the first chamber and a duration to perform the second manufacturing task using the second chamber; updating a machine learning model based on the duration to perform the first manufacturing task and the duration to perform the second manufacturing task; performing, by the processing device, a set of computer simulations that uses the machine learning model and the task data to produce a set of simulation results; storing, by the processing device, a simulation result of the set of simulation results in a data store.

[005] In another aspect of the disclosure, a system includes a memory and a processing device coupled to the memory. The processing device is to determine resources of a manufacturing tool, wherein the resources comprise a first chamber and a second chamber;

access task data indicating a first manufacturing task and a second manufacturing task; determine a duration to perform the first manufacturing task using the first chamber and a duration to perform the second manufacturing task using the second chamber; update a machine learning model based on the duration to perform the first manufacturing task and the duration to perform the second manufacturing task; perform a set of computer simulations that uses the machine learning model and the task data to produce a set of simulation results; store a simulation result of the set of simulation results in a data store.

[006] In an aspect of the disclosure, a non-transitory machine-readable storage medium storing instructions which, when executed, cause a processing device to perform operations involving: accessing the machine learning model to forecast the processing of a set of wafers; performing a first computer simulation using a set of start times for the set of wafers; detecting at least one wafer of the set is idle for a duration that exceeds a predefined threshold value; modifying the set of start times to include a different start time for the at least one wafer; and performing a second computer simulation using the modified set of start times.

BRIEF DESCRIPTION OF THE DRAWINGS

[007] The present disclosure is illustrated by way of example, and not by way of limitation in the figures of the accompanying drawings.

[008] FIG. 1 is a block diagram illustrating an exemplary system architecture, according to certain embodiments.

[009] FIG. 2 is an example computing device, according to certain embodiments.

[0010] FIG. 3 is a flow diagram illustrating an example method for performing simulation-based optimization of a manufacturing process, according to certain embodiments.

[0011] FIG. 4 is a block diagram illustrating a computer system, according to certain embodiments.

DETAILED DESCRIPTION

[0012] Manufacturing processes continue to increase in complexity and often involve many tasks that are performed to create a manufactured product. The tasks may be performed by a single tool or one or more tools. Each task may require one or more resources of the tool to complete the task and a resource conflict may occur when a set of products are attempting to use the same resource. To address the conflict, a tool may schedule one or more of the products to occupy the resource and may make the remaining products wait in a queue until the resource becomes available. Queuing the products may

reduce the throughput of the tool and in some situations may reduce the quality of the products. For example, manufactured products may be the result of many tasks and variations in the times the product are idle between tasks may introduce variations to the finished product. This may arise because the variations between the tasks may alter the properties of the product. Changes to thermal, moisture, texture, or other property may adversely affect current or subsequent tasks and may reduce the quality of the finished product.

[0013] Aspects of the present disclosure address the above and other deficiencies by providing technology that uses machine learning and computer simulations to optimize resource scheduling for a manufacturing process. In one example, the technology may be embedded within a manufacturing tool that can perform multiple tasks sequentially or parallel on a series of products. The products may be semiconductor wafers and the manufacturing tool may have multiple chambers that can each perform a task on one of the semiconductor wafers. In one example, the technology may involve a computing device determining the resources of the manufacturing tool, which may include a layout and capabilities of a set of chambers that can each hold one or more products (e.g., wafers). The technology may access task data that indicates a sequence of manufacturing tasks that the manufacturing tool can perform on the one or more products. Each task may be associated with a recipe for performing the task and may use a particular chamber of the manufacturing tool.

[0014] The technology may track how long the manufacturing tool takes to perform different manufacturing tasks using the different resources. For example, different resources may have different product capacities, load times, modification times, clean times, or a combination thereof. The technology may train a machine learning model to model the temporal aspects of the manufacturing tool based on the tracked times. The technology may then perform a set of computer simulations that uses the machine learning model and the task data to produce a set of simulation results. Each of the simulation results may be a schedule for the manufacturing tool that indicates the order in which the resources perform the tasks on a set of products. The technology may then calculate optimization values based on the simulation results and select one of the simulation results based on the optimization values. The optimization values may be specific to a set of one or more tasks and may be based on product idle time (e.g., wafer idle time between tasks), resource utilization (e.g., chamber idle time), tool throughput, other performance metric, or a combination thereof.

[0015] Systems and methods described herein include technology that enhances one or more manufacturing tools used in a manufacturing process. In particular, aspects of the

present disclosure may enhance the quality of manufactured products by reducing variations in product idle time during the manufacturing process. This may increase the consistency of manufactured products. The technology may also increase the utilization and/or throughput of the manufacturing tool or of particular resources of the manufacturing tool.

[0016] Various aspects of the above referenced technology are described in details herein below by way of examples, rather than by way of limitation. The examples provided below discuss a computing device that is communicatively coupled with a manufacturing tool. In other examples, the computing device may not be communicatively coupled to the manufacturing tool and may access data associated with the manufacturing tool from a data store and may generate data that can be subsequently used by a user or device to configure the manufacturing tool.

[0017] FIG. 1 is a block diagram illustrating an exemplary system architecture 100, according to certain embodiments. The system architecture 100 may be used to optimize a manufacturing process that may provide or produce one or more products 101. Product 101 may be for use or sale and may be a tangible or intangible object, good, service, other product, or a combination thereof. A tangible product may be touched by a human and may include physical products, goods, elements, or other object (e.g., etched wafer, semiconductor device, microchip, or other electronic device). Intangible products may be perceived directly or indirectly by a human without being touched and may include circuit designs, device layouts, manufacturing recipes, tool configuration, computer programs, services, other intangible element, or a combination thereof. In the example shown in FIG. 1, system architecture 100 may include a manufacturing tool 110 and one or more computing devices 120A-Z.

[0018] Manufacturing tool 110 may be any tool that contributes to a manufacturing process that produces product 101. The tool may be a machine, a computing device, a computer program, other device, or a combination thereof. Manufacturing tool 110 may include computer software (e.g., program code) and computer hardware (e.g., integrated circuits) and may be used for semiconductor device fabrication. Semiconductor device fabrication may involve the manufacture of semiconductor devices and integrated circuits (ICs). Manufacturing tool 110 may perform a multiple-step sequence that may include photolithographic and chemical processing steps (such as surface passivation, thermal oxidation, planar diffusion and junction isolation) during which electronic circuits are gradually created on a product made of a semiconducting material (e.g., silicon wafer). Manufacturing tool 110 may process product 101 using one or more resources 115A-Z.

[0019] Resources 115A-Z may be different portions of manufacturing tool 110 and may each be the same or similar to a partition, area, section, region, part, or other portion of manufacturing tool 110. In one example, each of the resource 115A-Z may be a chamber for performing one or more tasks on product 101. A chamber may hold one or more products and may be the same or similar to an enclosure, a compartment, a container, an opening, a vessel, a receptacle, a storage unit, a housing, or a combination thereof. Resources 115A-Z may include one or more different types of resources that can be used to process a product. Processing a product may involve performing an action that involves the product and may include one or more operations to load a product, transfer a product, modify a product, other operation, or a combination thereof.

[0020] In the example shown in FIG. 1, resources 115A-C may be configured to move products into manufacturing tool 110 or within manufacturing tool 110. Resource 115A may be an external transfer resource that includes a port and a robotic component to load one or more products 101 into manufacturing tool 110 from a transport device that is external to manufacturing tool 110. The transport device may transport a set of one or more products to a manufacturing tool, from a manufacturing tool, or between manufacturing tools. The transport device may be a pod that has an enclosure for securely storing a set of one or more products and may be the same or similar to a Front Opening Unified Pod (FOUP). Resources 115B-C may be internal transfer resources that are capable of storing products and transferring the products within manufacturing tool 110. The transfer resources may transfer a product into a resource, out of a resource, between resources, or a combination thereof. The internal and external transfer resources may be chambers with electro-mechanical actuators (e.g., robots) that can move wafers into and out of different chambers and may be referred to as robotic chambers. Resources 115R-Z may be portions of the manufacturing tool 110 that are capable of modifying product 101 in view of tasks 122A-Z. Each resource may be configured to perform a particular task (e.g., task 122A) or a set of multiple tasks (e.g., tasks 122B-C).

[0021] Tasks 122A-Z may correspond to manufacturing steps that are performed by manufacturing tool 110 to produce product 101. In one example, tasks 122A-Z may be semiconductor fabrication tasks and may include wafer cleaning (e.g., wet cleaning), photolithography, ion implantation, etching (e.g., wet etching, dry etching, atomic layer etching), plasma ashing, thermal treatment (e.g., annealing, thermal oxidization), vapor deposition (e.g., chemical vapor deposition (CVD), atomic layer deposition (ALD), physical vapor deposition (PVD)), molecular-beam epitaxy (MBE), polishing (e.g., chemical-

mechanical polishing), measuring (e.g., wafer or die measurement), testing (e.g., wafer or IC testing), back grinding, dicing, bonding, encapsulation, other task, or a combination thereof. Tasks 122A-Z may be associated with task data 122.

[0022] Task data 122 may identify the tasks and a sequence that the tasks can be performed in to produce product 101. Task data 122 may be referred to as sequence data, a sequence program, a recipe, or other term. Task data 122 may be provided by an entity designing the product to an entity manufacturing the product and the entities may be part of the same or different business entities. The entity designing the product may indicate a beginning task, one or more intermediate tasks, and an ending task. One or more of the tasks may be performed serially or in parallel on a particular product or on different products in a set (e.g., different wafers in a pod).

[0023] Task data 122 may include input data, output data, or a combination thereof for one or more of tasks 122A-Z. The input data for a task may include any data that defines one or more inputs for a manufacturing step. For example, input data may indicate one or more attributes of input products, configurations, settings, or other data. The input data may be the same or similar to parameter data, settings data, configuration data, other data, or a combination thereof. The input data may include one or more values that indicate how the manufacturing process will be performed (e.g., parameter values, setting values, configuration values). In one example, the input data may include one or more values corresponding to time (e.g., deposition time, etch time, oxidation time, implantation time, cleaning time), energy (e.g., temperature, current, voltage, electromagnetic frequency), input rate (e.g., gas flow rate, wafer spin speed), distance (e.g. space between substrate and tool, feature width, height, depth), pressure (e.g., Pascal, bar), input substance (e.g., precursor, reactant, dilutant), other attribute or property, or a combination thereof.

[0024] The output data may be any data that describes expected outputs of a manufacturing step. Output data may describe one or more attributes of product 101, by-products, other output, or a combination thereof. The output data may include values that indicate an actual attribute of product before, during, or after the step is performed. The attributes may correspond to one or more measurements of product 101. The measurements may be related to dimensions (e.g., length, width, height, depth, thickness, radius, diameter, area, volume, size), material property (e.g., reflectivity, emissivity, absorptivity, conductivity, density, texture), uniformity (e.g., film thickness uniformity), location (e.g., relative or absolute position), other attribute, or a combination thereof. In one example, the output data may indicate the attributes of product 101 in the form of an N-point metrology, wherein N

indicates the number of reference measurements for the product (e.g., 49 point metrology). The N-point metrology may provide critical dimensions for one or more electronic components of a semiconductor product (e.g., dimensions of transistor or via).

[0025] Computing devices 120-Z may include one or more computing devices that are associated with manufacturing tool 110. Computing devices 120A-Z may include an embedded system, a server, a workstation, a personal computer, a laptop computer, a tablet computer, a mobile phone, a palm-sized computing device, a personal digital assistant (PDA), etc. Computing devices 120A-Z may include computing devices implemented with x86 hardware (e.g., Intel® or AMD®), PowerPC®, SPARC®, or other hardware.

[0026] One or more of the computing devices 120A-Z may function as manufacturing control devices, sensor devices, user devices, other devices, or a combination thereof. A manufacturing control device (e.g., controller) may control a portion of the manufacturing tool 110 and may access, generate, or transmit input data, output data, or a combination thereof. A sensor device (e.g., sensor) may be capable of perceiving an aspect of manufacturing tool 110 or an aspect of product 101 and may involve a measurement component that is capable of measuring an attribute of the manufacturing tool 110 or product 101. In one example, the sensor device may include an image capturing module or an acoustic capturing module. User devices may be the same or similar to a client device and may provide a user interface for a user (e.g., process engineer). The user interface may present (e.g., display and/or announce) information to a user and may include one or more control elements for gathering user input. One or more of computing devices 120A-Z may run computer simulations 124 using a machine learning model 126.

[0027] Machine learning model 126 may mathematically model the durations of time that manufacturing tool 110 consumes to perform tasks. In one example, machine learning model 126 may be a single machine learning model that models manufacturing tool 110. In another example, machine learning model 126 may include a set of machine learning models and each model of the set may model one or more resources. In either example, each machine learning model may be the same or similar to a mathematical model, statistical model, neural network, other mathematical representation, or a combination thereof. Machine learning model 126 may include mathematical data that represents relationships between model input (e.g., tasks and/or resources) and model output (e.g., time durations). The mathematical data may include one or more matrices, indexes, value pairs, vectors, functions, equations, expressions, operators, operands, coefficients, variables, weights, biases, links, other data, or a combination thereof.

[0028] In one example, machine learning model 126 may be a data structure that includes one or more matrices that represent mathematical relationships. Each matrix may include one or more dimensions and may correlate a particular manufacturing task and/or resource to one or more durations of time. The matrix may include a single dimension (e.g., list), two dimensions (e.g., table), or three or more dimensions (e.g., 3D table). Each element of the matrix may correspond to a time duration and the location of the element in the matrix may indicate the corresponding task and/or resource. For example, a first axis (e.g., rows) may correspond to the resources and a second axis (e.g., columns) may correspond to the tasks and an intersection of a point on each axis may identify an element (e.g., time duration). The element may include one or more values that are based on the time duration the manufacturing tool took to perform the task using the resource. The one or more values may be based on historical values, maximum values, minimum values, average values, aggregate values, other numeric or statistical values, or a combination thereof.

[0029] In another example, machine learning model 126 may be a data structure that includes one or more mathematical functions that represent the mathematical relationships. The mathematical function may include a variable that corresponds to a resource and a variable that corresponds to the task and the output of the mathematical function may correspond to a time duration. In either example, machine learning model 126 may be an artifact of a machine learning process that analyzes training data and creates a model that represents relationship patterns and inferences derived from the training data.

[0030] The training data may include time series data that is based on historical or experimental runs of manufacturing tool 110. In one example, machine learning model 126 may include one or more regression models for analyzing task data 122 and deriving one or more durations of time. In another example, machine learning model 126 may include one or more neural networks that are trained using a deep learning technique and may include multiple layers. A neural network may be referred to as an artificial neural network (ANN), a network, a model, or other term. The neural network may be the same or similar to a Recurrent Neural Network (RNN), a Feed Forward Neural Networks (FFNN), a Multi-Layer Perceptron Neural Network (MLP), a Fully Connected Neural Network, a Radial Basis Function Neural Network (RBF), a Modular Neural Network, a Kohonen Self Organizing Neural Network, a Modular Neural Network, Convolutional Neural Network (CNN), other network, or a combination thereof. In either example, machine learning model 126 may be used perform performing computer simulations 124.

[0031] Computer simulations 124 may represent a computer simulation engine that executes a set of computer simulations to identify an optimized schedule. Each of computer simulations 124 may involve using machine learning model 126 to predict future values for manufacturing tool 110 based on previously observed values of one or more manufacturing tools. The prediction of future values may be referred to as time series forecasting and may involve time series analysis. In one example, each computer simulation may involve simulating the manufacturing tool 110 to identify a manufacturing schedule.

[0032] The manufacturing schedule may be based on a set of start times for the set of products and the computer simulation may iterate through the tasks for each product to populate the manufacturing schedule. A first task iteration may determine when a first product will be processed by each of the respective resources. The second task iteration may determine when a second product will be processed by each of the respective resources. The second iteration may account for resource conflicts and may delay the second product from being processed until after a resource finishes processes the first product. Manufacturing tool 110 may include multiple resources that can perform the same task and the second iteration may indicate the second product is processed by schedule by one of the other resources. The iterations may occur for each product until all the products are scheduled for processing. Earlier computer simulations may have the products start at the same time and that may result in products that have different wait times within the manufacturing tool (e.g., wafer idle time between tasks). Subsequent computer simulation may use different start times that cause subsequent products to wait to be loaded by the manufacturing tool but have little to no wait time once the manufacturing tool begins to process them. The subsequent computer simulations may identify a manufacturing schedule that reduces resource conflicts and product idle times. The computer simulations 124 may output the manufacturing schedules in the form of simulation result data 128.

[0033] Simulation result data 128 may include one or more simulation results 128A-C, which may each be the output of a computer simulation and may represent one or more manufacturing schedules. A manufacturing schedule may be represented as a set of one or more timelines that are organized by resource (e.g., resource schedule), by product (e.g., product schedule), by task (e.g., task schedule), other element, or a combination thereof. In one example, the simulation result may be organized by resource and each timeline may correspond to a respective resource (e.g., respective chamber) and may include time blocks that indicate when the resource processes each of the products (e.g., all the wafers modified by a particular chamber). In another example, the simulation result may be organized by

product and each timeline may correspond to a respective product (e.g., respective wafers) and may include time blocks that indicate when and where the product was processed (e.g., all the chambers that processed a particular wafer). In either example, the manufacturing schedules may be represented as a series of time values and may be analyzed via optimizations 127 to identify optimized results 129, as discussed in more detail below in regards to FIG. 2.

[0034] FIG. 2 depicts a block diagram illustrating an exemplary computing device 120 that includes technology to perform simulation based optimizations for a manufacturing tool, in accordance with one or more aspects of the present disclosure. Computing device 120 may be the same or similar to one or more of computing devices 120A-Z of FIG. 1. The components and modules discussed herein may be performed on a single computing device or on multiple different computing devices. In one implementation, one or more of the components may reside on different computing devices (e.g., model training on a first computing device and computer simulations on a different computing device). More or less components or modules may be included without loss of generality. For example, two or more of the components may be combined into a single component, or features of a component may be divided into two or more components. In the example shown in FIG. 2, computing device 120 may include a tool analysis component 210, a model creation component 220, and a computer simulation component 230.

[0035] Tool analysis component 210 may enable computing device 120 to analyze the manufacturing tool to determine the resources and capabilities of manufacturing tool. The analysis may occur before the manufacturing tool is shipped to a customer or may occur after the manufacturing tool is shipped to the customer and in either example may occur before, during, or after, an initialization of the manufacturing tool. The initialization may be the same or similar to a hardware initialization (e.g., system startup, BIOS), a software initialization (e.g., Operation System (OS) boot, program startup), other initialization, or a combination thereof. In one example, tool analysis component 210 may include a resource determination module 212, a task data module 214, and a time tracking module 216.

[0036] Resource determination module 212 may determine the resources and resource capabilities of the manufacturing tool by accessing one or more data storage objects, by communicating with one or more resources, or a combination thereof. As discussed above, the resources may include one or more areas, sections, or chambers of the tool that can be used to process a manufactured product (e.g., move or modify a wafer). The data storage objects may store data that indicates one or more of the resources of one or more

manufacturing tools and the data may be referred to as resource data or tool data. The data storage objects may include one or more file objects (e.g., files, directories), database objects (e.g., records, tuples, value pairs), Binary Large Objects (BLOBs), other data structures, or a combination thereof. Resource determination module 212 may also or alternatively determine the resources by communicating with one or more of the resources. The communication may be the same or similar to scanning or querying each of the resources and may involve transmitting requests and receiving responses with data indicating the resources of the manufacturing tool. Resource determination module 212 may store the data in data store 240 as illustrated by resource data 242.

[0037] Resource data 242 may include identification data, layout data, capability data, other data, or a combination thereof. The identification data may enable computing device 120 to uniquely identify a resource and may include one or more identifiers of the resources (e.g., resource identifiers, universally unique identifiers (UUID), a globally unique identifiers (GUID)). Layout data may indicate the location of a resource with respect to the tool or with respect to another resource. The layout data may enable resource determination module 212 to determine the layout of the manufacturing tool. The layout may indicate the location (e.g., relative location) of a resource with respect to one or more other resources or with respect to the tool. The location of the resource may be used to determine the time it takes to move a product between resources. The capability data may indicate the capabilities of the resource, which may include the type of resource (e.g., modification chamber, movement chamber) and one or more tasks that the resource can perform.

[0038] Task data module 214 may access task data that identifies the tasks for manufacturing the product. The task data may identify the steps and an order of the steps that manufacturing tools perform on the product. Task data module 214 may enable computing device 120 to receive the task data via user input, a data storage object, or combination thereof. In one example, the task data may be a sequence program that is stored in a centralized data store that is accessible over a network (e.g., recipe repository, sequence repository). Task data module 214 may receive (e.g., download) the task data and store it as task data 122 in data store 240. In one example, task data 122 may include a sequence of recipes to process a wafer and each recipe may correspond to a respective chamber of the manufacturing tool.

[0039] Time tracking module 216 may enable computing device 120 to determine how long the manufacturing tool takes to perform different tasks. The tasks may be any processing task and may involve tasks that modify a resource (e.g., pre-clean or post-clean a

resource), modify a product (e.g., photolithography, etch), move a product (e.g., load into tool or transfer in or out of a resource), or a combination thereof. In one example, time tracking module 216 may be used to monitor the manufacturing tool during the processing of one or more products. In another example, the manufacturing tool may record (e.g., log) time durations during the processing of the one or more products and time tracking module 216 may access the recorded times. In either example, time tracking module 216 may modify, aggregate, filter, or supplement the time durations before, during, or after being stored as time data 244 in data store 240.

[0040] Time data 244 may indicate the durations of time that the manufacturing tool consumes to perform tasks. Time data 244 may include an entry (e.g., log entry) that indicates one or more times (e.g., start time, end time, and/or duration), resources (e.g., loading chamber A, modification chamber Z), products (e.g., wafer identifier, pod identifier), tasks (e.g., sequence identifier, recipe identifier), tools (e.g., manufacturing tool 110), other data, or a combination thereof. In one example, time data 244 may be generated while a plurality of wafers are processed by a particular manufacturing tool and may indicate a plurality of wafer load times, processing times of the first chamber, processing times of the second chamber, and transfer times between chambers.

[0041] Model creation component 220 may use time data 244 to train a machine learning model 126 to model the manufacturing tool. In one example, model creation component 220 may include a data access module 222 and a training module 224.

[0042] Data access module 222 may access data that is associated with the manufacturing tool and can be used to train a machine learning model. The data may be referred to as training data and may include or be derived from any data in data store 240, such as, time data 244, resource data 242, task data 122, other data, or a combination thereof. Data access module 222 may access the training data directly or indirectly from the manufacturing tool. Accessing the training data directly from the manufacturing tool may involve accessing the training data from a communication channel or data store of the manufacturing tool. Accessing the training data indirectly from the manufacturing tool may involve accessing the training data from a data store that is separate from the manufacturing tool (e.g., from a log server). Data access module 222 may perform one or more processing operations on the accessed data before using it as training data. The processing may be referred to as pre-processing or post-processing and may involve one or more operations to correlate, normalize, add, remove, aggregate, filter, sterilize, anonymize, or other operation.

[0043] The training data may include or be based on historical data, simulated data, augmented data, other data, or a combination thereof. The historical data may be derived from one or more physical executions of the manufacturing process and the physical executions may be for commercial use, test use, experimental use, research and development use, other use, or a combination thereof. Augmented data may be based on historical data that is manipulated to include a variation that may or may not be present in the data being manipulated. The variation may alter the data using one or more mathematical transformations (e.g., reduce or increase durations based on recipe modifications).

[0044] Training data may include labeled data, unlabeled data, or a combination thereof. Labeled data may include primary data (e.g., a time duration) that is supplemented with auxiliary data (e.g., resource or product label). The auxiliary data be a tag, annotation, link, label, or other data. Unlabeled data may be data that is absent labels (e.g., missing labels, free of labels, without labels). The unlabeled data may be stored without labels and may or may not remain unlabeled prior to being accessed by training module 224.

[0045] Training module 224 may enable computing device 120 to analyze training data and create one or more machine learning models 126. Training module 224 may create new machine learning models or update existing machine learning models. Training a machine learning model may involve using one or more training techniques that are proprietary, standardized, open source, or a combination thereof. In one example, training module 224 may analyze the training data and identify one or more time durations for each combination of resource and task. For example, training module 224 may identify a duration of time (e.g., average duration, maximum duration, most recent duration) that a resource consumes to perform each applicable task. In another example, training module 224 may provide the training data to a training framework that creates the machine learning model 126. The training framework may be the same or similar to Tensorflow™, Keras™, PyTorch™, Open Neural network Exchange (Onnx™), Cognitive Toolkit (CNTK), Azure Machine Learning Service, Create ML, other machine learning framework, or a combination thereof.

[0046] Training module 224 may store the results of the training as machine learning model 126 in data store 240. The machine learning model 126 may be stored as one or more file objects (e.g., files, directories, links), database objects (e.g., records, tuples), other storage objects, or a combination thereof. Data store 240 may be a memory (e.g., random access memory), a drive (e.g., a hard drive, a flash drive), a database system, or another type of component or data storage device capable of storing data. Data store 240 may include multiple storage components (e.g., multiple drives or multiple databases) that may span

multiple computing devices (e.g., multiple server computers). Machine learning model 126 may be transmitted to one or more other data stores or computing devices and made accessible to computer simulation component 230.

[0047] Computer simulation component 230 may enable computing device 120 to perform a set of computer simulations that use the machine learning model 126 and task data 122 to produce a set of simulation results. Each simulation result may include a schedule for the resources that are used to process the products and may indicate the times that each resource processes the one or more products. The schedule may be used for a future execution of the manufacturing tool or for a current execution of the manufacturing tool. For example, input from a current execution (e.g., delay of resource X) may be used to update a model and simulate the remaining tasks to identify a revised schedule. The revised schedule may then be used by the manufacturing tool to alter the execution of subsequent tasks. These revisions to the schedule may be done in real time with little to no delay of a manufacturing process (e.g., no need to stop or restart the manufacturing tool). In the example of FIG. 2, computer simulation component 230 may include an initiation module 232, an execution module 234, an optimization module 236, and a result providing module 238.

[0048] Initiation module 232 may initiate one or more of the computer simulations using simulation inputs. The simulation inputs may include machine learning model 126 and model input. The model input may be based on task data 122, resource data 242, product quantity data, user input data, other data, or a combination thereof. The model input may be in the form of one or more input vectors that identify the tasks and/or resources that correspond to a product. Each product of a set may be associated with the same product vector or different product vectors. The model input may also include a set of start times that indicate when the manufacturing tool will begin processing the one or more of the products. The machine learning model 126 may be accessed from a local storage device, a remote storage device, or a combination thereof. In one example, initiation module 232 may receive machine learning model 126 from a local storage location (e.g., data store 240), which may occur when computing device 120 created the machine learning model or when the machine learning model is created by another computing device and is installed, packaged, downloaded, uploaded, or transmitted to computing device 120. In another example, initiation module 232 may access machine learning model 126 over a computer network 130 from a server on the same network or on a different network (e.g., internet), which may enable a customer to receive machine learning model 126 or an update to the model from a third party (e.g., manufacturing tool creator or distributor).

[0049] Execution module 234 may enable computing device 120 to execute a set of computer simulations sequentially, in parallel, or a combination thereof. During an execution of a computer simulation, execution module 234 may detect one or more resource conflicts. A resource conflict may arise when a product (e.g., second product) is waiting for a resource to finish processing another product (e.g., first product). Execution module 234 may address the conflict by adjusting the start times of one or more of the products, which may involve adjusting the start time of the first product to be earlier (e.g., expediting) or adjusting the start time of the second product to be later (e.g., delaying), or a combination thereof. The adjusted times may be incorporated into the simulation results.

[0050] Each simulation result may represent a set of one or more timelines that may be organized or arranged by resource or product. In one example, the time lines may be organized by resource and each of the timelines may correspond to a respective resource and may indicate when the resource processes each product (e.g., all the wafers modified by the respective chamber). In another example, the time lines may be organized by product and each of the timeline may correspond to a respective product and may indicate when and where the product is processed (e.g., all the chambers that processed the respective wafer). In either example, a simulation result may be represented as a set of output vectors.

[0051] An output vector may be a data structure that stores time series data and may represent a timeline for a particular resource (e.g., chamber), a particular product (e.g., wafer), a particular tool, other item, or a combination thereof. Each output vector may include time data, resource data, task data, or other data. The set of output vectors may include one or more resource vectors (e.g., chamber vectors) that correspond to specific resources or product vectors (e.g., wafer vectors) that correspond to specific manufactured products. In one example, the output vectors may be a set of resource vectors that each correspond to a particular resource of the manufacturing tool (e.g., a chamber). The resource vector may include time series data that includes a plurality of time values and product identifiers. The time series data may indicate multiple products that are processed by the resource and the time the products are processed (e.g., start time, end time, duration). In another example, the output vectors may be a set of product vectors that each correspond to a particular product manufactured by the manufacturing tool (e.g., a wafer). The product vector may include time series data that includes a plurality of time values and resource identifiers that indicate the multiple resources that process the product and the time the resource process the product.

[0052] Optimization module 236 may analyze the simulation results and calculate optimization values for the simulation results. The optimization value may be based on one or more products, resources, tools, or a combination thereof. For example, the optimization values may be based on product idle times (e.g., maximum wafer idle time), resource utilizations (e.g., chamber utilization rate), tool throughput (e.g., wafers per hour), other value, or a combination thereof. The optimization value may also or alternatively be based on one or more particular tasks. For example, the optimization value (e.g., product idle time) may be based on a time before, during, or after the particular task or set of tasks. This may be particularly advantageous when some of the tasks performed by the manufacturing tool on the product may be time dependent (e.g., time sensitive photolithography task). The time dependent variations across different products of a set may cause the manufactured products to vary in a manner that exceeds a predefined threshold (e.g., threshold value of thermal loss).

[0053] In one example, optimization module 236 may use the set of output vectors to determine the optimization value, such as an idle time of a product. The idle time may be based on one or more durations of time that the product waits for a resource conflict to be resolved (e.g., a resource to become available) and may include a duration before, during, or after a task is performed. The or more durations may include a duration before the product is loaded into the manufacturing tool from a transportation device, a duration in a storage resource (e.g., wafer buffer or queue), other duration, or a combination thereof. As discussed above, the output vectors may include multiple time values and the time values can be compared to determine the idle time of a wafer. When the output vectors are product vectors the idle time for a particular product can be determined by analyzing a single product vector. When the output vectors are resource vectors the idle time for a particular product may be determined by analyzing resource vectors for every resource that processes the particular product. In either example, determining the optimization value (e.g., idle time) may be performed by comparing time values of one or more of the output vectors.

[0054] Optimization module 236 may compare the optimization values of different computer simulations. The optimization value of a simulation result may be compared to one or more optimization values of the set of simulation results. The comparison may be a mathematical comparison that determines one or more differences between optimization values. The differences may be weighted and/or aggregated before or after the comparison. In one example, optimization module 236 may select the simulation result from the set of simulation results based on the comparison of optimization values (e.g., select the simulation result with the highest value or lowest value). In another example, optimization module 236

may determine whether to execute one or more additional computer simulations based on a comparison. This may involve determining whether the additional simulations are identifying more optimal solutions (e.g. schedules with less idle time). When subsequent simulation results are more optimized or substantially more optimized (e.g., optimization value difference above a threshold value) optimization module 236 may communicate with initiation module 232 to perform one or more computer simulations. When subsequent simulations results are less optimized or are nominally optimized (e.g., below a threshold value) optimization module 236 may not perform additional computer simulations and may select an optimized result 129 from the existing set of simulation results.

[0055] In one example, the output of a computer simulation may be used to determine the input of a subsequent computer simulation. For example, a first computer simulation may produce a resource schedule in which the last manufactured product is idle during an intermediate step. The duration of the idle time may be used to adjust the start time of the product in a subsequent computer simulation. For example, a simulation result may indicate product *x* was idle for duration *y* after task *n*. (e.g., fifth wafer delayed for 90 seconds) and the input for a subsequent computer simulation may delay the start time of product *x* for a duration of *n*. In another example, the initiation module may produce multiple different sets of product start times and may use each as input to computer simulations that are run and produce different simulation results.

[0056] Result providing module 238 may enable computing device 120 to provide simulation results to a user or device associated with the manufacturing tool. Providing the simulation results may involve presenting the simulation results, transmitting the simulation results, storing the results, or a combination thereof. Result providing module 238 may present the simulation results using a user interface (e.g., graphical user interface (GUI)) that displays one or more simulation results and details about the computer simulations (e.g., optimization values, number of simulations, value differences, best schedule). In one example, the user interface may include one or more display graphs that visually represent the simulation result as a schedule using one or more timelines (e.g., resource schedule, product schedule, task schedule). The display graphs (e.g., bar charts, Gantt charts) may include a graph organized by resource (e.g., resource graph) that includes a set of horizontal bars for a first resource and a set of horizontal bars for a second resource (e.g., first and second chambers). Each set of horizontal bars may represent time blocks that correspond to products that are processed by the respective resource. The display graphs may also or alternatively include a graph organized by product (e.g. product graph) that includes a set of

horizontal bars for a first product and a set of horizontal bars for a second product (e.g., first and second wafers). Each set of horizontal bars may represent time blocks that correspond to resources processing the respective product. The user interface may also include one or more control elements that are configured to receive user input of a user (e.g., process engineer). The control elements may be used to select one or more of the simulation results to be used by the manufacturing tool (e.g., selection of an optimized schedule).

[0057] Result providing module 238 may also provide optimized data associated with one of the simulated results to the manufacturing tool. The optimized data may be related to optimized results 129 and may include one or more starting times, timelines, schedules, or other data. In one example, providing the optimized data to the manufacturing tool may involve establishing a communication channel with the manufacturing tool and using the communication channel to transmit the optimized data from computing device 120 to the manufacturing tool. In another example, providing the optimized data to the manufacturing tool may involve storing the optimized data in a data storage object and enabling the manufacturing tool to access the data storage object (e.g., storing on a network storage device).

[0058] FIG. 3 depicts a flow diagram of one illustrative example of a method 300 for performing a simulation based optimization to identify a schedule for a manufacturing tool, in accordance with one or more aspects of the present disclosure. Method 300 and each of its individual functions, routines, subroutines, or operations may be performed by one or more processors of the computer device executing the method. In certain implementations, method 300 may be performed by a single computing device. Alternatively, method 300 may be performed by two or more computing devices, each computing device executing one or more individual functions, routines, subroutines, or operations of the method.

[0059] For simplicity of explanation, the methods of this disclosure are depicted and described as a series of acts. However, acts in accordance with this disclosure can occur in various orders and/or concurrently, and with other acts not presented and described herein. Furthermore, not all illustrated acts may be required to implement the methods in accordance with the disclosed subject matter. In addition, those skilled in the art will understand and appreciate that the methods could alternatively be represented as a series of interrelated states via a state diagram or events. Additionally, it should be appreciated that the methods disclosed in this specification are capable of being stored on an article of manufacture to facilitate transporting and transferring such methods to computing devices. The term “article of manufacture,” as used herein, is intended to encompass a computer program accessible

from any computer-readable device or storage media. In one implementation, method 300 may be performed by model creation component 220 and computer simulation component 230 as shown in FIG. 2.

[0060] Method 300 may be performed by processing devices of a server device or a client device and may begin at block 302. At block 302, a processing device may determine resources of a manufacturing tool. The resources may include a first chamber, a second chamber, and one or more additional chambers. The processing device may determine the resources by analyzing data of the manufacturing tool to identify a layout of the resources and to determine a type of each of the resources. In one example, the first chamber may be a modification chamber that modifies a product (e.g., an etching chamber) and the second chamber may be a movement chamber that moves the product (e.g., a robot chamber).

[0061] At block 304, the processing device may access task data indicating a first manufacturing task and a second manufacturing task. The task data may include a sequence of recipes to process a wafer product and the sequence of recipes may include a recipe for the first chamber and a recipe for the second chamber. The sequence of tasks may include a beginning task, a plurality of intermediate tasks, and an end task. One or more of the resources may be capable of performing the same task and each resource may be capable of performing one or more tasks.

[0062] At block 306, the processing device may determine a first duration to perform the first manufacturing task using the first chamber and a second duration to perform the second manufacturing task using the second chamber. The processing device may determine the durations by analyzing time data that was previously generated by the manufacturing tool while using the same task data or different task data to process a plurality of wafers. The time data may have been generated during a past manufacturing run (e.g., a historical run that has finished) or a current run (e.g., ongoing run that has not finished). In one example, the time data may include a plurality of times for the same manufacturing tool and the plurality of times may include one or more wafer load times, wafer processing times of the first chamber, wafer processing times of the second chamber, wafer transfer times between the first chamber and the second chamber, other times, or a combination thereof.

[0063] At block 308, the processing device may update a machine learning model based on the duration to perform the first manufacturing task and the duration to perform the second manufacturing task. The update may involve accessing time data for the manufacturing tool from a data store associated with the manufacturing tool and using the time data to train the machine learning model to temporally model the manufacturing tool. The trained machine

learning model may then be stored in the manufacturing tool or on a device that is connected to the manufacturing tool over a computer network.

[0064] At block 310, the processing device may perform a set of computer simulations that uses the machine learning model and the task data to produce a set of simulation results. Performing the set of computer simulations may involve accessing the machine learning model to forecast the processing of a set of wafers. The processing device may perform the first computer simulation using a set of start times for the set of wafers and may detect that at least one wafer of the set is idle for a duration that exceeds a predefined threshold value (e.g., greater than n seconds). The processing device may modify the set of start times to include a different start time for the at least one wafer and perform a second computer simulation using the modified set of start times.

[0065] The processing device may also calculate an optimization value for the simulation results. The optimization value may be based on one or more of a wafer idle time, a chamber utilization, a tool throughput, or a combination thereof. The processing device may compare the optimization value to one or more optimization values of the set of simulation results and select the simulation result from the set of simulation results based on the comparing. In one example, the optimization value may be calculated in view of a set of intermediate tasks (e.g., wafer idle time during tasks C and F).

[0066] At block 312, the processing device may store a simulation result of the set of simulation results in a data store. The simulation result may be a set of resource vectors that includes a resource vector for each chamber of the manufacturing tool that is used to process the set of products. Each resource vector may include a plurality of time values and may represent a time line for a particular chamber that processes a plurality of products. In one example, the processing device may provide for display a Gantt chart representing the stored simulation result for the manufacturing tool. The Gantt chart may include a set of horizontal bars for the first chamber that represent time blocks for one or more wafers of a set and a set of horizontal bars for the second chamber that represent time blocks for one or more wafers of the set. Responsive to completing the operations described herein above with references to block 312, the method may terminate.

[0067] FIG. 4 is a block diagram illustrating a computer system 400, according to certain embodiments. In some embodiments, computer system 400 may be connected (e.g., via a network, such as a Local Area Network (LAN), an intranet, an extranet, or the Internet) to other computer systems. Computer system 400 may operate in the capacity of a server or a client computer in a client-server environment, or as a peer computer in a peer-to-peer or

distributed network environment. Computer system 400 may be provided by a personal computer (PC), a tablet PC, a set-top box (STB), a Personal Digital Assistant (PDA), a cellular telephone, a web appliance, a server, a network router, switch or bridge, or any device capable of executing a set of instructions (sequential or otherwise) that specify actions to be taken by that device. Further, the term "computer" shall include any collection of computers that individually or jointly execute a set (or multiple sets) of instructions to perform any one or more of the methods described herein.

[0068] In a further aspect, the computer system 400 may include a processing device 402, a volatile memory 404 (e.g., random access memory (RAM)), a non-volatile memory 406 (e.g., read-only memory (ROM) or electrically-erasable programmable ROM (EEPROM)), and a data storage device 416, which may communicate with each other via a bus 408.

[0069] Processing device 402 may be provided by one or more processors such as a general purpose processor (such as, for example, a complex instruction set computing (CISC) microprocessor, a reduced instruction set computing (RISC) microprocessor, a very long instruction word (VLIW) microprocessor, a microprocessor implementing other types of instruction sets, or a microprocessor implementing a combination of types of instruction sets) or a specialized processor (such as, for example, an application specific integrated circuit (ASIC), a field programmable gate array (FPGA), a digital signal processor (DSP), or a network processor).

[0070] Computer system 400 may further include a network interface device 422. Computer system 400 also may include a video display unit 410 (e.g., an LCD), an alphanumeric input device 412 (e.g., a keyboard), a cursor control device 414 (e.g., a mouse), and a signal generation device 420.

[0071] In some implementations, data storage device 416 may include a non-transitory computer-readable storage medium 424 on which may store instructions 426 encoding any one or more of the methods or functions described herein, including instructions encoding the tool analysis component 210, model creation component 220, or computer simulation component 230 of FIG. 2 and for implementing methods described herein.

[0072] Instructions 426 may also reside, completely or partially, within volatile memory 404 and/or within processing device 402 during execution thereof by computer system 400, hence, volatile memory 404 and processing device 402 may also constitute machine-readable storage media.

[0073] While computer-readable storage medium 424 is shown in the illustrative examples as a single medium, the term "computer-readable storage medium" shall include a

single medium or multiple media (e.g., a centralized or distributed database, and/or associated caches and servers) that store the one or more sets of executable instructions. The term "computer-readable storage medium" shall also include any tangible medium that is capable of storing or encoding a set of instructions for execution by a computer that cause the computer to perform any one or more of the methods described herein. The term "computer-readable storage medium" shall include, but not be limited to, solid-state memories, optical media, and magnetic media.

[0074] The methods, components, and features described herein may be implemented by discrete hardware components or may be integrated in the functionality of other hardware components such as ASICs, FPGAs, DSPs or similar devices. In addition, the methods, components, and features may be implemented by firmware modules or functional circuitry within hardware devices. Further, the methods, components, and features may be implemented in any combination of hardware devices and computer program components, or in computer programs.

[0075] Unless specifically stated otherwise, terms such as "receiving," "determining," "selecting," "performing," "processing," "measuring," "reporting," "updating," "analyzing," "training," "creating," "adding," "calculating," "causing," "performing," "generating," "using," "modifying," "comparing," "accessing," "shifting," "storing," "detecting," or the like, refer to actions and processes performed or implemented by computer systems that manipulates and transforms data represented as physical (electronic) quantities within the computer system registers and memories into other data similarly represented as physical quantities within the computer system memories or registers or other such information storage, transmission or display devices. Also, the terms "first," "second," "third," "fourth," etc. as used herein are meant as labels to distinguish among different elements and may not have an ordinal meaning according to their numerical designation.

[0076] Examples described herein also relate to an apparatus for performing the methods described herein. This apparatus may be specially constructed for performing the methods described herein, or it may include a general purpose computer system selectively programmed by a computer program stored in the computer system. Such a computer program may be stored in a computer-readable tangible storage medium.

[0077] The methods and illustrative examples described herein are not inherently related to any particular computer or other apparatus. Various general purpose systems may be used in accordance with the teachings described herein, or it may prove convenient to construct more specialized apparatus to perform methods described herein and/or each of their

individual functions, routines, subroutines, or operations. Examples of the structure for a variety of these systems are set forth in the description above.

[0078] The above description is intended to be illustrative, and not restrictive. Although the present disclosure has been described with references to specific illustrative examples and implementations, it will be recognized that the present disclosure is not limited to the examples and implementations described. The scope of the disclosure should be determined with reference to the following claims, along with the full scope of equivalents to which the claims are entitled.

CLAIMS

What is claimed is:

1. A method comprising:
 - determining, by a processing device, resources of a manufacturing tool, wherein the resources comprise a first chamber and a second chamber;
 - accessing task data indicating a first manufacturing task and a second manufacturing task;
 - determining a duration to perform the first manufacturing task using the first chamber and a duration to perform the second manufacturing task using the second chamber;
 - updating a machine learning model based on the duration to perform the first manufacturing task and the duration to perform the second manufacturing task;
 - performing, by the processing device, a set of computer simulations that uses the machine learning model and the task data to produce a set of simulation results;
 - selecting a simulation result from the set of simulation results based on one or more optimization values; and
 - storing, by the processing device, the simulation result in a data store.
2. The method of claim 1, further comprising analyzing time data generated by the manufacturing tool, wherein the time data is generated while a plurality of wafers are processed by the manufacturing tool in view of the task data.
3. The method of claim 2, wherein the time data comprises a plurality of times for the manufacturing tool, wherein the plurality of times comprises a plurality of wafer load times, wafer processing times of the first chamber, wafer processing times of the second chamber, and wafer transfer times between the first chamber and the second chamber.
4. The method of claim 1, further comprising:
 - accessing time data for the manufacturing tool from a data store associated with the manufacturing tool;
 - training the machine learning model to model the manufacturing tool using the time data; and
 - storing the machine learning model in the manufacturing tool.

5. The method of claim 1, wherein performing the set of computer simulations comprises:
- accessing the machine learning model to forecast the processing of a set of wafers;
 - performing a first computer simulation using a set of start times for the set of wafers;
 - detecting at least one wafer of the set is idle for a duration that exceeds a predefined threshold value;
 - modifying the set of start times to include a different start time for the at least one wafer; and
 - performing a second computer simulation using the modified set of start times.
6. The method of claim 1, wherein the simulation result comprises a set of resource vectors that comprises a resource vector for each chamber of the manufacturing tool, wherein the resource vector comprises a plurality of time values and represents a time line for a chamber that processes a plurality of products.
7. The method of claim 1, further comprising:
- calculating the one or more optimization values, wherein the one or more optimization values comprises an optimization value for the simulation result that is based on one or more of a wafer idle time, a chamber utilization, or a tool throughput;
 - comparing the optimization value for the simulation result to at least one optimization value of another simulation result of the set of simulation results; and
 - selecting the simulation result from the set of simulation results based on the comparing.
8. The method of claim 7, wherein the task data indicates a sequence of tasks comprising a beginning task, a plurality of intermediate tasks, and an end task, and wherein the optimization value corresponds to the wafer idle time during a particular set of the intermediate tasks.
9. The method of claim 1, further comprising providing for display a Gantt chart representing the simulation result for the manufacturing tool, wherein the Gantt chart comprises a set of horizontal bars for the first chamber that represent time blocks for one or more wafers of a set and a set of horizontal bars for the second chamber that represent time blocks for one or more wafers of the set.

10. The method of claim 1, wherein determining the resources of the manufacturing tool comprise:

analyzing the manufacturing tool to identify a layout of the resources; and

determining a type of each of the resources, wherein the first chamber comprises a modification chamber that modifies a product and wherein the second chamber comprises a movement chamber that moves the product.

11. The method of claim 1, wherein the task data comprises a sequence of recipes to process a wafer, wherein the sequence of recipes comprises a recipe for the first chamber and a recipe for the second chamber.

12. A system comprising:

a memory; and

a processing device communicably coupled to the memory, the processing device to:

determine resources of a manufacturing tool, wherein the resources comprise a first chamber and a second chamber;

access task data indicating a first manufacturing task and a second manufacturing task;

determine a duration to perform the first manufacturing task using the first chamber and a duration to perform the second manufacturing task using the second chamber;

update a machine learning model based on the duration to perform the first manufacturing task and the duration to perform the second manufacturing task;

perform a set of computer simulations that uses the machine learning model and the task data to produce a set of simulation results;

select a simulation result from the set of simulation results based on one or more optimization values; and

store the simulation result in a data store.

13. The system of claim 12, wherein the time data comprises a plurality of times for the manufacturing tool, wherein the plurality of times comprises a plurality of wafer load times, wafer processing times of the first chamber, wafer processing times of the second chamber, and wafer transfer times between the first chamber and the second chamber.

14. The system of claim 12, wherein the simulation result comprises a set of resource vectors that comprises a resource vector for each chamber of the manufacturing tool, wherein the resource vector comprises a plurality of time values and represents a time line for a chamber that processes a plurality of products.

15. The system of claim 12, wherein the task data indicates a sequence of tasks comprising a beginning task, a plurality of intermediate tasks, and an end task.

16. A non-transitory machine-readable storage medium storing instructions which, when executed, cause a processing device to perform operations comprising:

accessing time data from a data store associated with a manufacturing tool comprising a first chamber and a second chamber, wherein the time data comprises a duration to perform a manufacturing task using the first chamber and a duration to perform a manufacturing task using the second chamber;

training, by the manufacturing tool, a machine learning model to model the manufacturing tool using the time data;

performing, by the manufacturing tool, a set of computer simulations that uses the machine learning model to produce a set of simulation results;

selecting a simulation result from the set of simulation results based on one or more optimization values; and

storing, by the processing device, the simulation result in a data store, wherein the simulation result comprises a schedule for the manufacturing tool.

17. The non-transitory machine-readable storage medium of claim 16, wherein the time data comprises a plurality of times for the manufacturing tool, wherein the plurality of times comprises a plurality of wafer load times, wafer processing times of the first chamber, wafer processing times of the second chamber, and wafer transfer times between the first chamber and the second chamber.

18. The non-transitory machine-readable storage medium of claim 16, wherein the operations further comprise:

accessing the machine learning model to forecast the processing of a set of wafers;

performing a first computer simulation using a set of start times for the set of wafers;

detecting at least one wafer of the set is idle for a duration that exceeds a predefined threshold value;

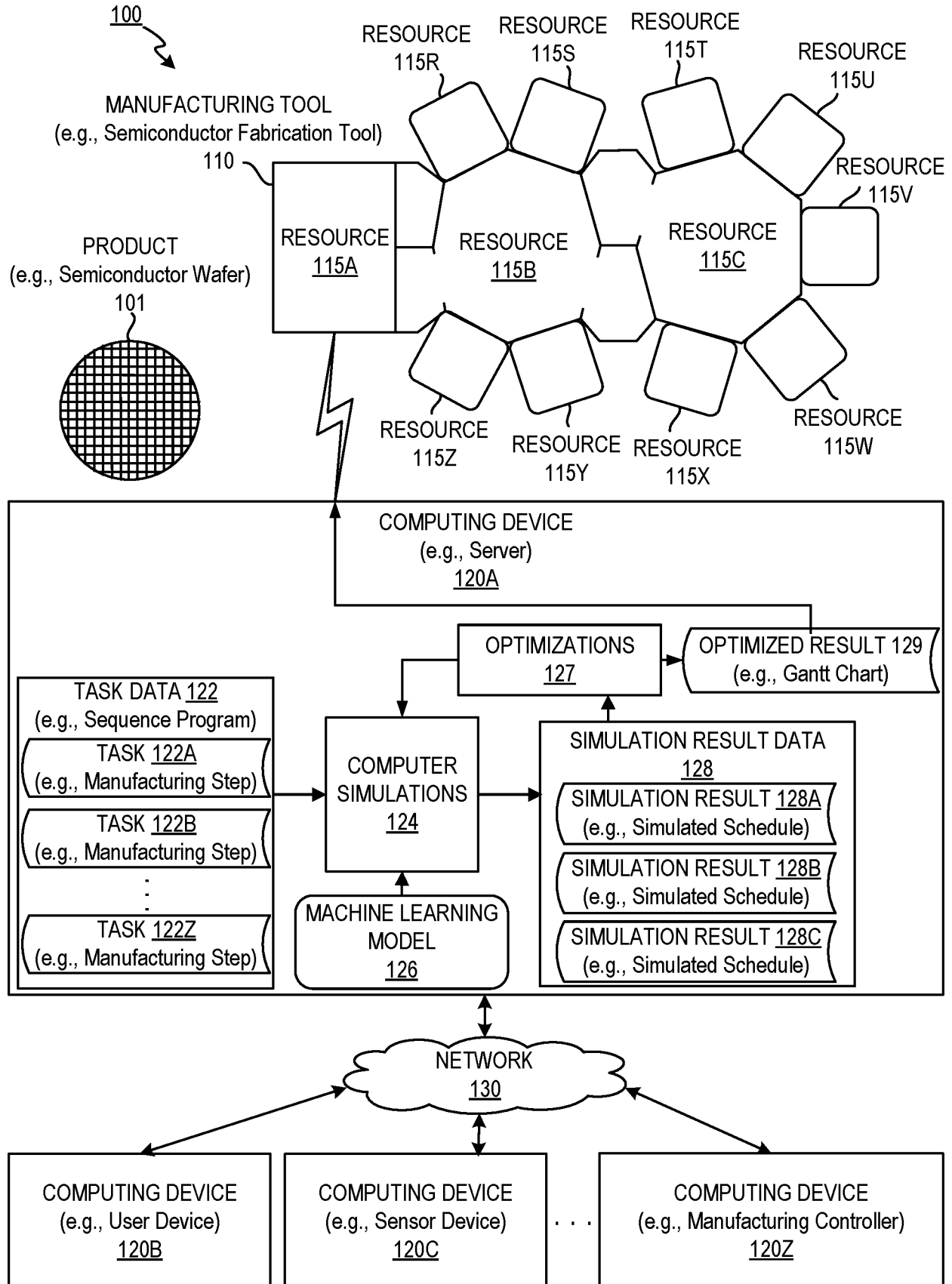
modifying the set of start times to include a different start time for the at least one wafer; and

performing a second computer simulation using the modified set of start times.

19. The non-transitory machine-readable storage medium of claim 16, wherein the simulation result comprises a set of resource vectors that comprises a resource vector for each chamber of the manufacturing tool, wherein the resource vector comprises a plurality of time values and represents a time line for a chamber that processes a plurality of products.

20. The non-transitory machine-readable storage medium of claim 16, wherein the operations further comprise, providing for display a Gantt chart representing the simulation result for the manufacturing tool, wherein the Gantt chart comprises a set of horizontal bars for the first chamber that represent time blocks for one or more wafers of a set and a set of horizontal bars for the second chamber that represent time blocks for one or more wafers of the set.

SYSTEM ARCHITECTURE

**FIG. 1**

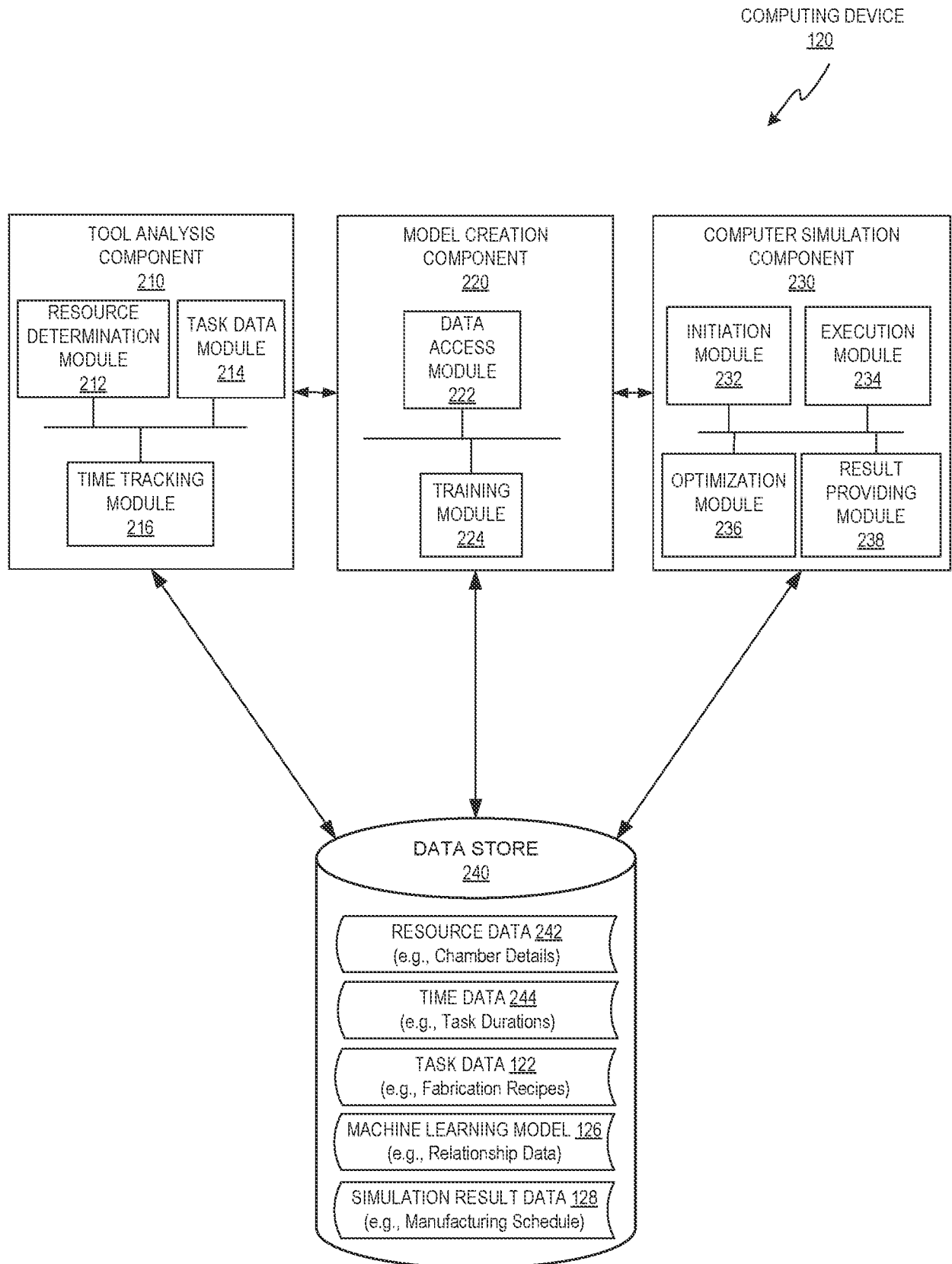


FIG. 2

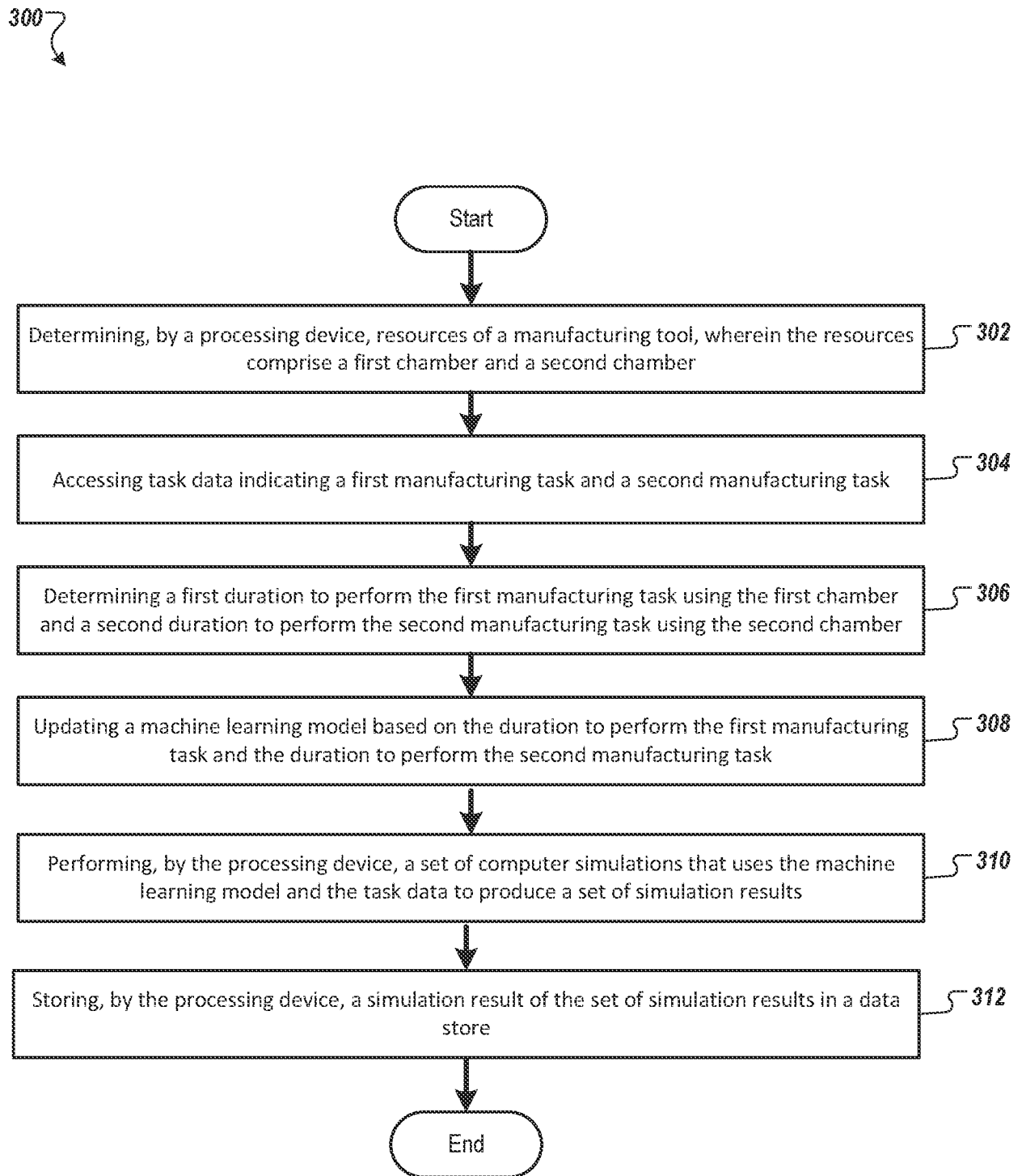
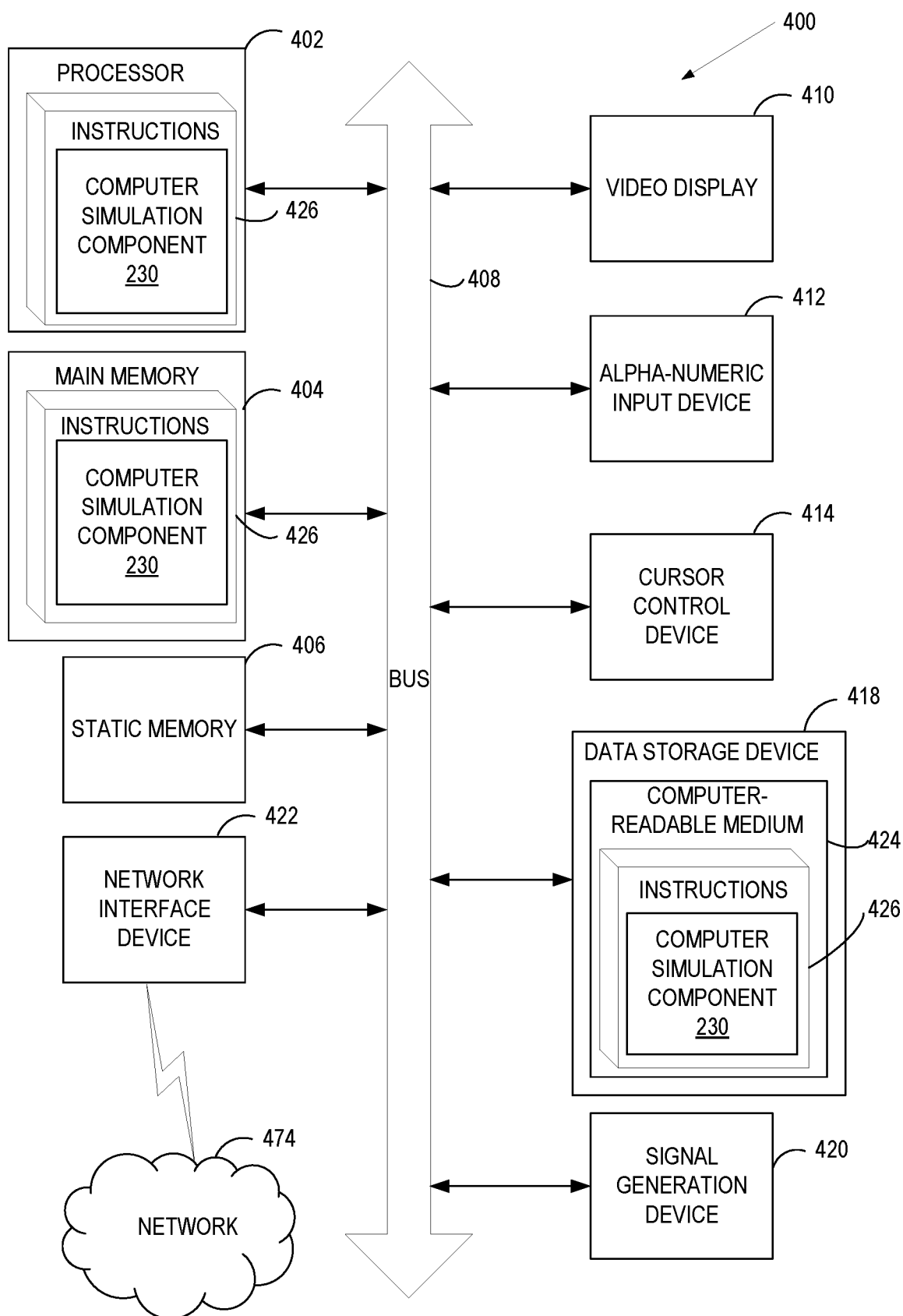


FIG. 3

**FIG. 4**

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US2021/033747

A. CLASSIFICATION OF SUBJECT MATTER**G05B 19/418**(2006.01)i; **H01L 21/67**(2006.01)i; **G06N 20/00**(2019.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G05B 19/418(2006.01); G05B 11/01(2006.01); G05B 15/00(2006.01); G06F 9/455(2006.01); G06Q 10/00(2006.01); G06Q 50/00(2006.01); G06Q 50/04(2012.01); H01L 21/67(2006.01)

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Korean utility models and applications for utility models
Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

eKOMPASS(KIPO internal) & Keywords: wafer, manufacture, tool, chamber, tool, task, duration, machine learning, simulation, time, vector, optimization, gantt chart

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	KR 10-2013-0141743 A (VMS SOLUTIONS CO., LTD.) 27 December 2013 (2013-12-27) paragraphs [0001]-[0044]; and figures 1-5	1-20
Y	US 2020-0110390 A1 (APPLIED MATERIALS, INC.) 09 April 2020 (2020-04-09) paragraphs [0006]-[0096]; claims 1-17; and figures 1A-13	1-20
Y	US 8082045 B1 (YOSHIKI ASHIZAWA et al.) 20 December 2011 (2011-12-20) column 2, line 44 - column 13, line 53; and figures 1-24	9,20
A	US 2010-0185480 A1 (KUNG-JENG WANG et al.) 22 July 2010 (2010-07-22) paragraphs [0021]-[0072]; and figures 1-6	1-20
A	KR 10-1999-0039062 A (SK HYNIX INC.) 05 June 1999 (1999-06-05) pages 2-3; and figures 1-3	1-20

☐ Further documents are listed in the continuation of Box C.☒ See patent family annex.

* Special categories of cited documents:

“A” document defining the general state of the art which is not considered to be of particular relevance

“D” document cited by the applicant in the international application

“E” earlier application or patent but published on or after the international filing date

“L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

“O” document referring to an oral disclosure, use, exhibition or other means

“P” document published prior to the international filing date but later than the priority date claimed

“T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

“X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

“Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

“&” document member of the same patent family

Date of the actual completion of the international search

10 September 2021

Date of mailing of the international search report

10 September 2021

Name and mailing address of the ISA/KR

Korean Intellectual Property Office
189 Cheongsa-ro, Seo-gu, Daejeon
35208, Republic of Korea

Facsimile No. +82-42-481-8578

Authorized officer

BYUN, Sung Cheal

Telephone No. +82-42-481-8262

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/US2021/033747

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
KR	10-2013-0141743	A	27 December 2013	KR	10-1371006	B1	25 March 2014
US	2020-0110390	A1	09 April 2020	CN	112805635	A	14 May 2021
				KR	10-2021-0055105	A	14 May 2021
				TW	202038030	A	16 October 2020
				US	10657214	B2	19 May 2020
				US	10705514	B2	07 July 2020
				US	10930531	B2	23 February 2021
				US	10955832	B2	23 March 2021
				US	2020-0110852	A1	09 April 2020
				US	2020-0111689	A1	09 April 2020
				US	2020-0279066	A1	03 September 2020
				US	2020-0333774	A1	22 October 2020
				WO	2020-076719	A1	16 April 2020
US	8082045	B1	20 December 2011	None			
US	2010-0185480	A1	22 July 2010	TW	201028878	A	01 August 2010
				TW	I391839	B	01 April 2013
				US	8271311	B2	18 September 2012
KR	10-1999-0039062	A	05 June 1999	None			