



[12] 发明专利申请公开说明书

[21] 申请号 01822650.7

[43] 公开日 2004年9月15日

[11] 公开号 CN 1529835A

[22] 申请日 2001.10.22 [21] 申请号 01822650.7

[30] 优先权

[32] 2001. 2. 21 [33] US [31] 09/789,871

[86] 国际申请 PCT/US2001/050178 2001.10.22

[87] 国际公布 WO2002/069063 英 2002.9.6

[85] 进入国家阶段日期 2003.8.14

[71] 申请人 先进微装置公司

地址 美国加利福尼亚州

[72] 发明人 C·A·巴帝 A·J·帕斯汀

A·J·塔派克 J·S·O·休特

A·O·彼得森 T·J·松德尔曼

M·L·米勒

[74] 专利代理机构 北京纪凯知识产权代理有限公司

司

代理人 戈泊程伟

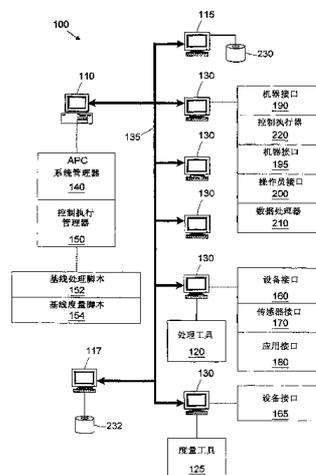
权利要求书2页 说明书14页 附图6页

[54] 发明名称 使用基线控制脚本控制工具的方法和装置

[57] 摘要

本发明提供一种控制制造系统(100)的方法,其包括:在多个工具中处理工件,对于该多个工具(120,125)中的选定工具(120,125)启动基线控制脚本(152,154);提供上下文信息给该基线控制脚本(152,154);根据该上下文信息确定工具类型;根据该工具类型选择该选定工具(120,125)的控制例程;及执行该控制例程以产生该选定工具(120,125)的控制动作。制造系统(100)包含适用于处理工件的多个工具(120,125)、控制执行管理器(150)及控制执行器(220)。控制执行管理器(150)适合用于启动多个工具(120,125)中的选定工具(120,125)的基线控制脚本(152,154)并且提供上下文信息给该基线控制脚本(152,154)。控制执行器(220)适合用于执行基线控制脚本(152,154)、根据上下文信息确定工具类型、根据该工具

类型而选择所选定工具(120,125)的控制例程,以及执行该控制例程以产生该选定工具(120,125)的控制动作。



1. 一种控制制造系统(100)的方法, 包括:
在多个工具(120, 125)中处理工件;
对于该多个工具(120, 125)中的选定工具(120, 125)启动一基线控制
- 5 脚本(152, 154);
提供该基线控制脚本(152, 154)的上下文信息;
根据该上下文信息确定工具类型;
根据该工具类型选择用于该选定工具(120, 125)的控制例程; 及
执行该控制例程以产生该选定工具(120, 125)的控制动作。
- 10 2. 根据权利要求1的方法, 其中所述选择控制例程进一步包括链接至一控制例程的库(240)。
3. 根据权利要求1的方法, 其中所述上下文信息包括与所述选定工具相关的一实体识别码, 且确定所述工具类型进一步包括基于该实体识别码而确定该工具类型。
- 15 4. 根据权利要求1或3的方法, 其中所述上下文信息包括一操作识别码, 且选择所述控制例程进一步包括根据所述工具类型及该操作识别码而选择该控制例程。
5. 根据权利要求1、3或4的方法, 其中所述上下文信息包括一产品识别码, 且选择所述控制例程进一步包括根据所述工具类型及该产品识别码而选择该控制例程。
- 20 6. 一种制造系统(100), 包括:
多个工具(120, 125), 其适合用于处理工件;
一控制执行管理器(150), 其适合用于启动用于所述多个工具(120, 125)中的选定工具(120, 125)的基线控制脚本(152, 154), 且提供上下文
- 25 信息给该基线脚本(152, 154); 和
一控制执行器(220), 其适宜用于执行所述基线脚本(152, 154)、根据所述上下文信息确定工具类型、根据该工具类型而选择一用于所述

选定工具的控制例程、且执行该控制例程以产生该选定工具(120, 125)的控制动作。

7. 根据权利要求6的系统(100), 其中所述控制执行器(220)适合用于链接至一控制例程的库(240)以选择该控制例程。

5 8. 根据权利要求6的系统(100), 其中所述上下文信息包括与所述选定工具相关的一实体识别码, 且所述控制执行器(220)适合用于根据该实体识别码而确定所述工具类型。

10 9. 根据权利要求6或8的系统(100), 其中所述上下文信息包括一操作识别码, 并且所述控制执行器(220)适合用于根据所述工具类型及该操作识别码而选择所述控制例程。

10 10. 根据权利要求6、8或9的系统(100), 其中所述上下文信息包括一产品识别码, 并且所述控制执行器(220)适合用于根据所述工具类型及该产品识别码而选择所述控制例程。

使用基线控制脚本控制工具的方法和装置

技术背景

- 5 本发明一般涉及半导体器件工业的领域，尤其涉及一种利用基线控制脚本来控制工具的方法和装置。

背景技术

- 10 在半导体工业中有持续的推动力以提高集成电路器件的质量、可靠度和生产率，该集成电路器件例如为微处理器、存储器器件和相似的器件。这种推动力由于用户对于工作更可靠的更高质量的计算机和电子器件的需求而加强。在如晶体管的半导体器件的制造以及与此类晶体管结合的集成电路器件的制造中，这些需求已经导致连续改进。此外，减少在典型晶体管的器件的制造中的缺陷也进一步降低每个晶
- 15 体管的整体成本和结合此类晶体管的集成电路器件的成本。

- 一般而言，利用多种处理工具而在一批晶片上执行一组处理步骤，该多种处理工具包含光刻步进控制器、蚀刻工具、沉积工具、抛光工具、快速热处理工具、注入工具等。基于半导体处理工具的技术在最近这几年来已经吸引更多的注目，导致显著的改进。然而，尽管在此
- 20 领域中有进展，目前可在市场上购到的处理工具仍有某些缺陷。尤其此类工具通常缺乏先进的处理数据监视能力，如以用户友好的格式提供历史参数数据的能力，和事件记录，以及整个批次的处理参数及当前处理参数的实时图形显示和遥控(remote)即局部及全球的监视。这些缺陷可以引起关键处理参数如生产率、精确度、稳定性和可靠性、处
- 25 理温度、机械工具参数等等的非优化控制。这种变异性显示为批量中的变异、批量间的变异、以及工具间的变异，这些变异可能增大成为产品质量与功能上的差异，而对此类工具的理想监视和诊断系统将提供一种监视此变异的手段和提供一种关键参数的最佳控制手段。

- 一种改善半导体处理线的操作技术包含利用全厂范围的控制系
- 30 统以自动控制各种处理工具的操作。此制造工具与制造框架或处理模块

的网络通信。每个制造工具通常连接一设备接口。该设备接口连接一利于制造工具和制造框架之间通信的机器接口。该机器接口通常可以是先进处理控制(advanced process control)(APC)系统的一部份。APC 系统根据制造模型而启动一控制脚本，该制造模型可以是一个自动检索
5 需要执行制造处理的数据的软件程序。通常半导体器件通过多个制造工具而分阶段进行多个处理，产生有关所处理半导体器件质量的数据。

在制造处理期间，可能发生多种情况而影响所制造的半导体器件(简称为器件)的性能。也就是说，制造的过程步骤中的变化造成器件性能的变化。诸如结构关键尺寸、掺杂物含量、触点电阻、微粒污染等
10 因素都可能潜在地影响器件的目标效率。控制在处理线中的多种工具与效率模块一致以减少处理变异(variation)。一般地控制工具包含光刻步进控制器、研磨工具、蚀刻工具和沉积工具。预处理及/或后处理度量数据被提供给该工具的处理控制器。诸如处理时间的操作方法参数通过根据效率模块和度量信息
15 的处理控制器计算以试图达成后处理产生尽可能接近目标值。在此方法中减少的变化导致生产率增加、成本减少、器件性能更高等，全部同等于利润率增加。

在分布式计算机环境中，如全厂范围的先进处理控制(APC)系统中，配置控制和效率问题是普遍的。通常有许多软件开发者编写控制代码以构造处理控制器。一个特定开发者可能大范围工作以开发某特定类型的控制器。一般每个开发者有特有的编程风格，且依靠其自己
20 创造的例程。例如，一个开发者可能有一组例程，用以接口库或其它在 APC 框架中的实体以实现多种数学功能和基本的实用功能。

一个与此类安排相关的问题就是在处理控制脚本之间很少一致。大量的用户脚本也显示出配置控制问题和效率问题。或许为了另一个
25 开发者已创造的不同类型的处理控制器，开发者们会花费相当多的时间复制已经开发的程序代码。调试非标准化的程序代码也是耗时的并进一步减低效率。

本发明目的在于克服或至少减少上述的一个或多个问题的影响。

30 **发明概要**

本发明在一方面可视为一种控制制造系统的方法。该方法包括在

多个工具中处理工件；为多个工具中的选定工具启动基线控制脚本；提供上下文信息给该基线控制脚本；根据该上下文信息确定工具类型；根据该工具类型选择控制例程给该选定工具；及执行该控制例程以产生一控制动作给该选定工具。

- 5 本发明在另一方面可视为一制造系统，该制造系统包括适宜处理工件的多个工具，一控制执行管理器，以及一控制执行器。该控制执行管理器适宜启动基线控制脚本给多个工具中的选定工具并提供上下文信息给该基线控制脚本。控制执行器适合于执行基线控制脚本、根据上下文信息而确定工具类型、根据该工具类型而选择控制例程给该
- 10 选定工具，以及执行该控制例程以产生控制动作给该选定工具。

附图简单说明

本发明可以通过以下配合附图的说明而了解，其中相同的参考编号表示相同的组件，其中：

- 15 图 1 为根据本发明的一示范实施例的先进处理控制(APC)系统的简化方框图；

图 2 为示意图，表明图 1 的系统中基线控制脚本和多种共享的基线库系统之间的链接；

图 3 为简化方框图，说明基线处理脚本的组织；

- 20 图 4 为简化方框图，说明基线度量脚本的组织；

图 5 为简化方框图，说明多个控制器基线处理脚本的组织；而

图 6 为根据本发明的另一个说明性实施例而集成与多个控制器的方法的简化流程图。

- 25 尽管本发明可有多种修改和替代形式，以下就特定的实施例通过图示方式显示并且在本文中详细描述。但应该了解本文所述的特定实施例并非意图局限本发明于所公开的特定形式，而是相反，本发明意图涵盖所有变量形式、对等物及任意在本发明的精神和领域中的形式，如所附权利要求范围的规定。

具体实施方式

30 以下详述本发明的示范实施例。为清楚起见，未将实际实施方案

的所有特征描述于本文中。当然应认识到，在任何此类实际的实施例的开发中为了达到开发者的特定目标，必须作许多实施上特定的决定，如与系统相关和商业相关的约束相一致，该约束随着一个实施方案到另一个实施方案而变化。再者，也应认识到此种研发努力可能复杂而耗时，但对得益于本发明所做公开的本领域技术人员而言，仍然是例行工作。

现参考附图，并且优先参考图 1，该图为先进处理控制(APC)系统 100 的简化方框图。APC 系统 100 是可换的标准化的软件组件的一种分布式软件系统，其允许处理流程到处理流程控制及故障检测/分类。

该软件组件实施依据半导体设备及材料国际组织 (SEMI) 计算机集成制造(CIM) 框架服从系统技术规格和先进处理控制(APC) 框架而执行结构体系标准。CIM(SEMI E81-0699- CIM 框架领域体系结构的临时规格) 及 APC(SEMI E93-0999- CIM 框架先进处理控制组件临时规格) 规格可以公开地从 SEMI 取得。该特定的体系结构主要依赖利用面向对象的程序设计的软件，并利用对象管理组(OMG) 的共同对象请求代理体系结构 (CORBA) 以及分布对象系统的共同对象请求代理体系结构服务 (CORBA_Services) 规格。OMG CORBA 的体系结构的信息和规格也易于公开取得。适合实现如本文所述的 APC 系统 100 功能的一种示范性软件系统为 KLA-Tencor 公司所提供的 Catalyst 系统。

所述器件利用 CORBA 接口规定语言(IDL) 互相通信并且依靠一共同组的服务以支持其交互作用。标准的一组分布式对象服务由对象管理组(OMG) 规定。其中这些服务有：

CORBA——用于所有直接的组件对组件交互作用的基于标准的通信协议。可根据面向对象的远程启用通信模型规定标准接口。这些接口和所有 APC 通信均利用接口规定语言(IDL) 规定。各组件通过在彼此接口上调用操作而通信。数据作为操作参数和返回值而在各组件之间传递。

OMG 事件服务——支持组件之间的异步通信。许多先进处理控制 (APC) 对象随其改变状态而发出事件。这些事件由相关的事件用户接收。先进处理控制(APC) 系统中的事件使用的例子包括但不限于通信组件状态(包含错误状态)、由缺陷检视和分类软件检测到的缺陷警告通

知，以及机器状态和收集数据的报告。

OMG 交易服务——能使一组件发现另一个与其交互作用的组件。当安装一组件时，其服务的描述(服务提供者)即被输出至交易服务站。另一组件其后可以要求符合某些标准的服务提供者列表。该交易服务站提供其它能提供所需服务的组件的列表。组件启动即使用该能力，从而使一个组件得以发现必须与之交互作用的其它组件。当计划执行组件需要发现能力提供者以提供在此计划中指定的所需权能时，一旦计划启动也使用该能力。

这些服务在本技术中是为公知的。OMG 的 CORBA/IIOP 规格文件和 CORBA 服务规格文件在本领域的规格文件中广为分布并且提供更详细的项目。

在一示范实施例中，先进处理控制(APC)系统 100 适用于控制半导体制造环境。各组件利用 CORBA 接口规定语言(IDL)而互相通信。该协同操作的软件组件管理处理控制计划/策略；从处理设备、度量工具和附加传感器收集数据；以此种信息调用多种处理控制应用程序/算法；以及更新处理模型和更改/下载适当的操作方法参数。APC 系统 100 为一种控制半导体生产处理的全工厂范围的软件系统，但这并非为实施本发明所必需。本发明所教授的策略可以应用在不同的任何规模的计算机系统。

在一示范实施例中，所述 APC 系统 100 包含 APC 主计算机 110、数据库服务器 115、117、处理工具 120、度量工具 125 及一个或多个工作站 130。APC 系统的组件通过总线 135 互连。总线 135 可能实际包含多个层并利用多种协议。APC 系统的全部操作通过驻在于 APC 主计算机 110 中的 APC 系统管理器 140 控制。APC 系统管理器 140 提供：对于为所述 APC 框架开发的所有服务器的管理、配置、事件和状态服务；在 APC 系统 100 中各组件的规定、分组、安装和管理；为了诊断和监控目的而捕获活动及追踪信息的集中化服务；组件配置信息，包含设定值、系统环境设置的集中化储存；及相关对象和事件通道的列表。然而，在其它实施例中，这些功能可能分成一种或多种软件组件，例如基本管理器、系统管理器、注册器及纪录。

APC 系统 100 包含一处理模块网络。这些处理模块有时称作“集

成组件(integration component)”。集成组件作为对现有工厂系统的接口而工作，并且提供运行 APC 计划的能力。“APC 计划”为一应用程序，其被调用以完成某些特定工作，下文将作更完整的讨论。该集成组件显示它们可以被 APC 系统 100 中的多种处理资源所托管(hosted)。这些特定的托管位置提供用于示范目的。处理资源是互连的，并且多种软件组件可以依据系统的复杂性而分布在多种计算机中或者集中化。在本特定实施例中的每个集成组件是软件实现的。它们利用在本技术中公知的面向对象的程序设计技术以 C++编程。APC 系统 100 的优点为其模块结构，其提供了软件组件的可移植性。所述集成组件包含但不限于 APC 管理器 140；控制执行管理器 150；与工具 120、125 相关的设备接口 160、165；与处理工具 120 相关的传感器接口 170；应用接口 180；机器接口 190、195；操作员接口 200；及数据处理器 210。

控制执行管理器 150 是主要用于“设计”APC 系统 100 操作的组件。控制执行管理器 150 解释 APC 计划，执行主要脚本及次脚本，并且当事件要求时调用事件脚本。各种计划、脚本和次脚本可以用于多种实施方案。可以特定地实施多种计划、脚本和次脚本的特定数目和功能。例如，本实施例包含但不限于以下计划：

数据收集计划——由传感器和机器接口使用的数据结构，其规定要求从特定处理设备收集何种数据，及该数据应如何报告返回；

持续期间计划——规定触发条件和触发延迟的计划，该触发条件和触发延迟导致传感器器动作，例如开始数据收集、终止数据收集；

报告计划——规定如何处理所收集的数据及何时发出该数据可利用性的信号的计划；及

取样计划——规定通过外部传感器收集数据的频率的计划；

控制计划——一批控制脚本，设计用以执行 APC 活动；及

控制脚本——APC 系统在特殊规定情况下执行的一系列动作/活动。

控制执行管理器 150 协调在所有集成组件中，对于给定工具如处理工具 120 的用户规定的处理控制计划的实行。当指令下达，控制执行管理器 150 就检索计划及其相关脚本。该管理器预处理次脚本以提供例程给主要脚本和事件脚本。它也获得为执行该计划所需的能力列

表，如该计划所规定的，并且连接至适当集成组件以提供所需能力。

控制执行管理器 150 然后委派给责任给控制执行器 220 以运行该计划。在所示实施例中，控制执行管理器 150 利用基线控制脚本以确定将执行的控制行动。基线处理脚本 152 被指定用于如处理工具 120 5 的处理工具，而基线度量脚本 154 则被指定用于如度量工具 125 的度量工具。基线脚本 152、154 的更详细讨论参考图 2 至图 6 提供于下。

控制执行管理器 150 依据适当的基线处理脚本 152 或基线度量脚本 154 产生控制执行器 220，以顺序执行计划，并将该计划的完成或该计划执行中的错误报告返回至控制执行管理器 150。因此，控制执行管
10 理器 150 负责所执行的全部计划的整体管理，而每个控制执行器 220 则只负责运行一个计划。控制执行器 220 由控制执行管理器 150 产生，在计划的使用期限中存在，而在报告计划完成或中断之后由控制执行管理器 150 毁掉。控制执行管理器 150 可以经由多个控制执行器 220 同时启动多个计划。

15 机器接口 190、195 填补 APC 框架如 APC 管理器 140 与设备接口 160、165 之间的间隙。机器接口 190、195 使处理或度量工具 120、125 与 APC 框架接口，并且支持机械设置、激活、监视和数据收集。在此特定实施例中，机器接口 190、195 主要在设备接口 160、165 的特定通信和 APC 框架的 CORBA 通信之间进行翻译。更具体地说，机器接
20 口 190、195 从设备接口 160、165 接收指令、状态事件和收集数据并当需要时发送给其它 APC 组件和事件通道。依次地，来自其它 APC 组件的响应由机器接口 190、195 接收并发送给设备接口 160、165。当需要时，机器接口 190、195 还重新格式化和重建消息及数据。机器接口 190、195 支持启动/关闭在 APC 系统管理器 140 中的程序。它们也
25 作为 APC 数据收集器工作，缓冲由设备接口 160、165 所收集的数据并且发送适当的数据收集事件。

传感器接口 170 收集由监视处理工具 120 操作的传感器所产生的数据。传感器接口 170 提供适当的接口环境以与外部传感器通信，该外部传感器例如为 LabVIEW 或其它传感器、基于总线的数据采集软
30 件。应用接口 180 提供适当的接口环境以执行诸如 LabVIEW、Mathematica、ModelWare、MatLab、Simca 4000 及 Excel 的控制插件

应用程序。该传感器可以通过初始设备制造商(OEM)装有处理工具 120, 或者它们可以是从 OEM 获得之后安装的“外接式附件”传感器。传感器接口 170 收集由传感器产生的数据。传感器可以产生例如关于操作情况的压力和温度的数据。应用接口 180 从控制执行器 220 取得数据并对该数据执行计算或分析。其结果随后被返回给控制执行器 220。机器接口 190 和传感器接口 170 利用共同一组功能以收集要利用的数据。设备接口 160 汇集由传感器收集的关于处理工具 120 的相应数据并将所汇集的数据传送至机器接口 190。

操作员接口 200 经由图形用户接口(GUI)(图中未示)而有利于晶片制造技术人员和 APC 系统 100 之间的通信。该 GUI 可以是基于 Windows®或 UNIX 的操作系统。但这并非为本发明的实施所必需。事实上, 有些替代实施例可能甚至不会利用 GUI 而可经由基于操作系统的磁盘操作系统(DOS)通信。操作员接口 200 显示对话框以提供信息、请求引导和收集额外的数据。通过 CORBA 接口, 操作员接口 200 组件允许技术人员在任意数目的显示器组上同时显示多种弹出的对话。操作员接口 200 也维持一组其中可以显现弹出对话的显示器。操作员接口 200 也可能提供通知操作, 即显示带有消息和“OK”按钮的简单弹出对话框的单向消息。

数据处理器 210 接收由其它 APC 系统 100 组件产生的数据, 并将数据储存于数据库服务器 115、117 上的数据储存库 230、232 中(例如相关数据库)。数据处理器 210 可以适合于接收结构化查询语言(SQL)指令, 或可选地, 该数据处理器 210 可以转换不同类型的存取协议以产生 SQL 指令或某种其它协议指令。集中化数据储存功能可增加多种组件的可移植性。

现参照图 2 所示简化方框图说明基线处理脚本 152、154 的一般操作。图 2 表示基线处理脚本 152、154 和多种共享的基线库之间的链接。一般而言, 基线处理脚本 152、154 提供开发 APC 系统 100 中的控制脚本的框架。基线处理脚本利用储存在库中的共享基线组件。在所示实施例中, 共享基线组件包含: 控制基线库 240, 其用于规定控制算法; 数学基线库 250, 其用于规定通常使用的数学功能(例如求和、平均、中位数等); 交互作用基线库 260, 其用于规定脚本完成的通信方面(例

如与下列组件的交互作用：数据储存库 230 和 232、机器接口 195、经由机器接口 195 的设备接口 160、操作员接口 200 和其它此类外部组件)；实用基线库 270，其用于规定共享的共同功能；设备库 280，其用于规定功能，或是对设备特定的其它库 240、250、260、270 中例程的例外；以及层库 290，其用于根据包含在对基线控制脚本的调用中的操作 ID，而规定层(例如多门极层)。基线库 240、250、260、270、280、290 可在基线控制脚本 152、154 操作期间由控制处理器 220 链接。

一般而言，基线控制脚本 152、154 基于包含在对脚本的调用中的信息以及在设备和层基线库 280、290 中的信息，而确定控制动作的性质。该基线控制脚本 152、154 链接至控制基线库 240 以存取必要的控制功能。基线控制脚本 152、154 链接至交互作用基线库 260 以存取汇集数据的功能，用以执行控制动作并与设备接口 160 通信从而更新工具 120、125 的操作方法。当需要时，在数学基线库 250 中的功能可以由基线控制脚本 152、154 或其它库中的功能调用。

现参照图 3，其为说明基线处理脚本 152 组织的简化方框图。基线处理脚本 152 包含应用程序配置方框 300、基线应用设置方框 310、控制器常量及上下文特定设置方框 320、前馈数据分析方框 330、控制线程方框 340、危险方框 350、控制动作和业务规则方框 360 及结果方框 370。

在应用程序配置方框 300 中，由控制器基于包含于来自设备接口 160 的调用中的信息，规定用户全局配置变量。这包含来自方案管理系统(RMS)(即方案设置的全局数据库)的变量值及所需的上下文变量。上下文变量值规定了控制线程，且通常包括诸如工具识别码、批号(lot number)、操作号等等变量的数值。此外，任何需要的基线变量也是给定数值。例子包括错误通知的电子邮件列表、暂停的数值、在认为是“子(child)”批的一个批量中所允许的晶片的最大数目、控制器所使用的先前层的信息(前馈信息)等等。

基线应用设置方框 310 利用如在应用程序配置方框 300 中设定的批号和晶片数量的值，并且返回晶片的批号、系列名称、父批名称、设备、数目和状态(即是否该批为父批或子批)。基线应用设置方框 310 也设定控制器会对之发送弹出窗口的终端的默认列表，以及所有弹出

窗口标题的第一部份。

5 控制器常量及上下文特定设置方框 320 利用之前规定的内容和 RMS 信息来设定控制器用以计算控制移动的数值。例如，控制器常量及上下文特定设置方框 320 可利用上下文信息(或“线程”标识)，依据在 RMS 中规定的数值而设定控制模型参数。特定的例子可以是根据如 RMS 中规定的特定蚀刻室及该蚀刻室的蚀刻速率的数值的上下文，设定在控制模型中所使用的蚀刻速率的数值。此外，控制器常量及上下文特定设置方框 320 通过如应用程序配置方框 300 中所设定的批号和层名，而利用查询来从数据库检索前馈信息。

10 前馈数据分析方框 330 检查与已给定批相关的数组数据中的元素且对遗失数值填充默认值。例如，先预处理的目标可用以设定所需的遗失测量值，作为控制器所使用的前馈信息的一部分。设定遗失的前馈信息数值的其它方法，其替代使用默认值，也可在前馈数据分析方框 330 中执行。

15 控制线程方框 340 设定所需的各键的值及状态结构以询问数据储存库 230、232，从而检索与目前控制线程相关的控制状态。其中键被用来从数据储存库 230、232 检索线程状态数据。控制线程方框 340 搜索线程状态数据，该数据处于以此线程上下文所处理的当前批的有序数据的堆栈中。如果发现这样的数值，则它们被传送到一用户规定的包含控制模型的功能块，其计算并且返回线程状态的数值。若未在此堆栈中发现数值，该控制线程方框 340 搜寻该分层结构，并且从具有线程状态数值的第一层级检索数据。该堆栈及所有层级均视为包含相似数据，但是有不同的精确程度。

20 危险方框 350 执行数据库中的查询并检索在危险堆(即自最后一次度量操作以来，在给定线程上所处理的批的堆栈)中的批的数值。此数值与在这一危险类别中的批的数值的阈值比较，该阈值通常规定于 RMS 中。若未超出此阈值，控制器继续工作。若超出阈值，控制器中途停止且跳出显示，指示操作员根据危险堆中的批的列表而对批执行度量事件。

30 控制动作和业务规则方框 360 是所述控制器的核心。控制动作和业务规则方框 360 根据状态和目标信息计算控制器输入(处理方案更

新)。这些结果置于全局控制结果数组中。其次，控制动作和业务规则方框 360 执行业务规则，限制检验处理方案更新，和/或根据控制器的用户输入过载(overrides)而设定处理方案更新。

结果方框 370 从控制动作和业务规则方框 360 取得输出，其包含
5 处理方案更新、数据计算/格式化或事件，加将其与数据一起缓冲并格式
化。结果方框 370 发送该缓冲数据至设备接口 160 并通过机器接口
195 启动对设备接口 160 的设置/开始机器调用。其次，结果方框 370
储存数据，储存在数据储存库 230、232 中，对照当前上下文(线程)的
批号和层。该危险堆同样更新，其中当前批作为自最后一次度量事件
10 以来所处理的附加批。

现参照图 4，其提供说明基线度量脚本 154 组织的简化方框图。基
线度量脚本 154 包含度量工具设置方框 400、应用程序配置方框 410、
基线应用设置方框 420、收入工具数据方框 430、控制器常量及上下文
特定设置方框 440、控制线程方框 450、模型更新方框 460 及结果方框
15 470。

在度量工具设置方框 400 中，启动数据收集并将任何缓冲数据送
至控制处理器 220。机器接口 190 也被启动而发送设置/开始机器调用
给设备接口 165。

应用结构方框 410 和基线应用设置方框 420 执行类似于描述于上
20 文中相关于基线处理脚本 152 的相同名字的方框的功能。

收入工具数据方框 430 暂停基线度量脚本 154 以等待来自数据源
的数据设定，该数据源通常为一度量工具。解除脚本暂停的等待该事
件的时间段和事件名称在收入工具数据方框 430 中加以规定。

控制器常量及上下文特定设置方框 440 也执行类似上文关于基线
25 处理脚本 152 所描述的具有同样名称的方框的功能。

控制线程方框 450 设定所需的各键的值和状态结构，由当前线程
将计算出的控制状态储存至数据储存库 230、232。此外，控制线程方
框 450 计算所有为更新该线程状态所需的数值。此功能读取所规定的
全局变量数值并计算所需结果。该结果包含在更新控制器时使用的统
30 计数据或数值，诸如批量平均值、处理速率和距目标或预测的偏差。
此功能的结果置于全局控制结果数组中。

模型更新方框 460 用于执行业务规则、规格限度检查和控制器的过载。此功能读取规定的全局变量值且设定最后结果。模型更新方框 460 负责设定将用来更新控制器的数值和将要记录至控制历史的数值。此功能的结果置于全局控制结果数组中。

5 结果方框 470 从控制器常量及上下文特定设置方框 440 取得输出，将其缓冲并且格式化数据以兼容于设备接口 165。由基线度量脚本 154 输出的数据也写入控制历史文件。该控制历史的标题基于所提供的变量名产生。纪录文件将该标题编码在文件的第一行。如果计算出的标题不匹配该文件第一行，现存文件就改名且启用新文件。

10 现再参照图 5，其为多控制器基线处理脚本 500 的简化方框图，该多控制器基线处理脚本根据本发明的另一实施例能够在单一处理工具 120 上实施多个控制动作。例如，一光刻步进控制器可以既具有覆盖控制器也具有关键尺寸控制器。该控制器利用来自处理后晶片的反馈以调节各步进控制器参数，如曝光量、曝光时间、焦点等等。沉积工具，
15 如用以形成多晶硅层的工具，也可以具有多个控制器，以控制诸如多晶硅颗粒尺寸和多晶硅层厚度的参数。

当调用基线处理脚本 500 时，它根据包含在该调用中的信息而确定所需控制动作。待处理批量中的上下文确定哪一个控制器将会运行。该上下文通过操作 ID、实体 ID、产品 ID 和其它此类确定特殊运行要求的这类标识符来规定。首先用实体 ID 来确定工具类型的一般等级(例如，步进控制器、蚀刻机、加热炉等等)。例如，如果该实体 ID 识别
20 处理工具 120 为步进控制器，则调用步进控制器的控制码。

在步进控制器的控制码中，检查脚本中的上下文变量以确定将调用哪一个单独的控制器。操作 ID 则指示将进行的处理 (例如，多门极掩模和
25 第二互层(interlevel)介电层掩模(ILD))。每一个控制器提供一组上下文情况且只在所有那些上下文条件满足时运行。例如，CD 控制器可以对于多门极掩模运行，但不在第二互层介电层掩模(ILD)处理时运行。另一方面，重叠控制器可以在两种掩模事件中都运行。

基线处理脚本 500 根据工具组(例如步进控制器)而提供符合所需
30 工具代码的灵活性，且准备运行所有可用的控制器(例如，重叠、CD 等等)。相同的主要脚本运行，相同的次例程可被调用，但只有在当前

上下文中需要的控制器激活。

多控制器基线处理脚本 500 包含应用程序配置方框 510、基线应用设置方框 520、控制器常量及上下文特定设置方框 530、前馈数据分析方框 540、控制线程方框 550、危险方框 560、控制动作和业务规则方框 570 及结果方框 580。除了下述情形外，多控制器基线处理脚本 500 以类似于基线处理脚本 152 的方式工作。

控制器常量和上下文特定设置方框 530 确定应用哪个控制器(例如，控制器 A、控制器 B 或两者)，且利用先前规定的上下文和 RMS 信息来设定每个控制器用以计算控制移动的数值。控制器常量和上下文特定设置方框 530 也通过如在应用结构方框 510 中设定的批量数目及层名，使用查询而从数据库检索每个所需控制器的前馈信息。前馈数据分析方框 540 检查与给定批相关的数据数组中的元素，并对于每个控制器的遗失数值填充默认值。

控制线程方框 550 设定所需的各键的值和状态结构以询问数据储存库 230、232，从而检索与当前控制线程相关的每个现用控制器的控制状态。其中各键被用来从数据储存库 230、232 检索线程状态数据。控制线程方框 550 搜索线程状态数据，该数据处于以此线程上下文所处理的最近批的有序数据的堆栈中。如果发现这样的数据，则它们被传送到一用户规定的包含控制器模型的功能块，其计算并返回线程状态的数值。如果未在该堆栈中发现数值，控制线程方框 340 搜索该分层结构，并且从具有线程状态数值的第一层级检索数据。该堆栈和所有层级均视为包括相似数据，但有不同的精确程度。

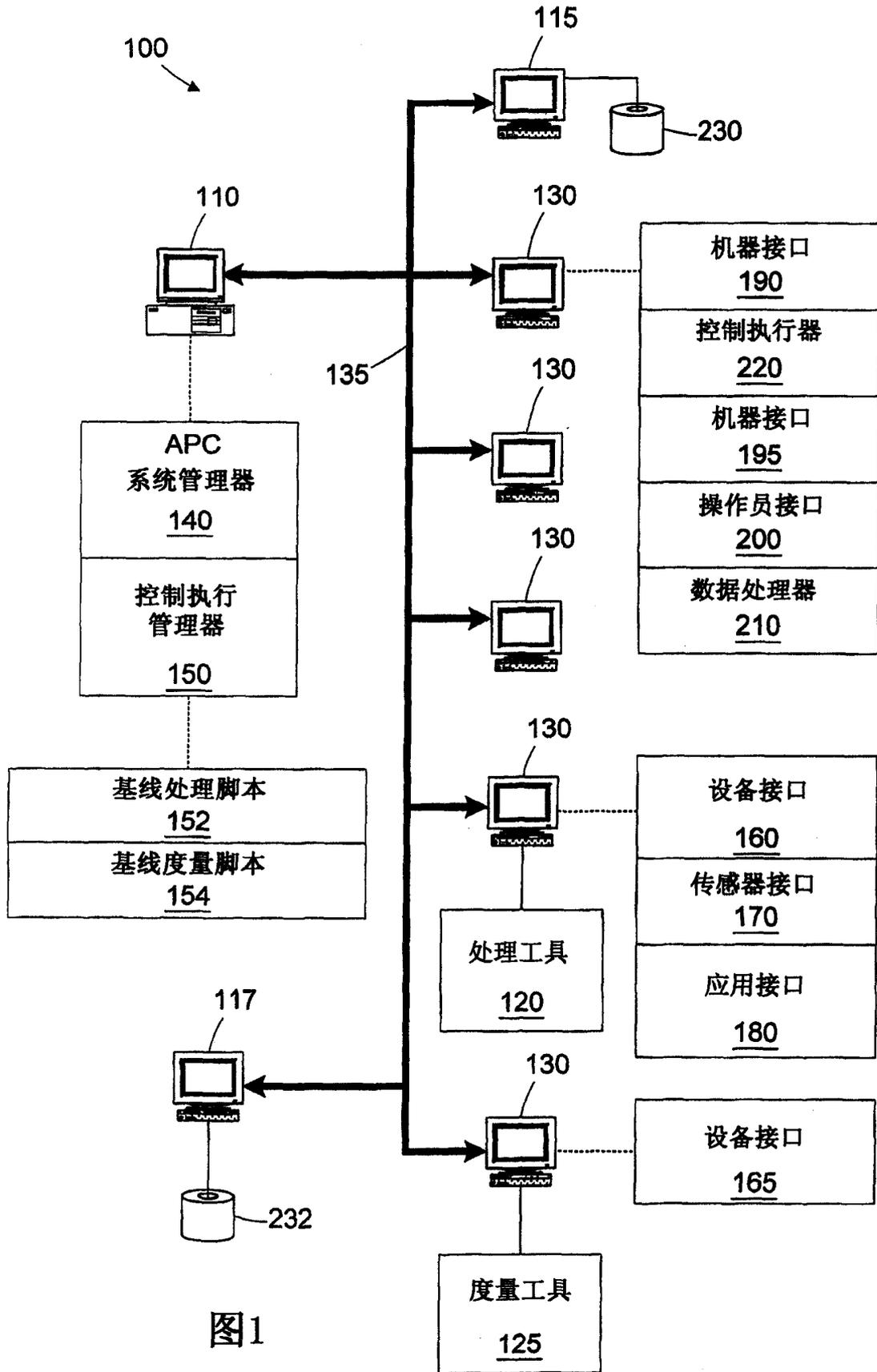
控制动作和业务规则方框 570 根据每个控制器的状态和目标信息计算控制器输入(处理方案更新)。因为利用多个控制器，一个控制器可能影响其它控制器所依赖用以确定其控制动作的状态信息。因此，控制器可被分配以相关的优先值，从而确定以之规定其控制动作的顺序。较高优先的控制器可基于其控制动作的规定，对应于次一个控制器而更新状态信息。通过这种方式的协作，各控制器将不会相关于操作方法变化而彼此冲突。

结果方框 580 汇集来自所有现用控制器的控制动作输出、缓冲数据和格式化数据。结果方框 580 将缓冲数据发送至设备接口 160，并通

过机器接口 195 启动对设备接口 160 的设置/开始机器调用。其次，结果方框 580 储存数据，储存在数据储存库 230、232 中，对照当前上下文(线程)的批号和层，并且更新危险堆。

5 现参照图 6，其提供根据本发明另一实施例而集成多个控制器的方法的简化流程图。在方框 600 中，在多个工具中处理工件。在方框 610，对于多个工具中的选定工具启动基线控制脚本(例如通过控制执行管理器 150)。在启动该基线控制脚本之后，控制控制执行器 220 执行其余任务。在方框 620 中识别一组为选定工具所需的控制例程。在方框 630 中，检索关于先前控制动作的控制状态信息，其与该组为选定工具所需的控制例程相关。在方框 640 中，执行出自该组所需控制例程的第一控制例程以产生第一控制动作。在方框 650 中，基于第一控制动作而改变控制状态信息，该控制状态信息与出自该组所需控制例程的第二控制例程相关。在方框 660 中，基于所改变的控制状态信息而执行第二控制例程以产生第二控制动作。

15 以上所公开的特定实施例仅是说明性的，得益于本文指导的本领域技术人员显然能够以不同但等效的方式修改且实施本发明。再者，本发明不应限定于在此所示的构造或设计细节，而仅限于所附权利要求书。因此揭露于上文的特定实施例显然可以变更或修改并且所有此类的变异都应认为在本发明的范围之内。因此，本文中所请求的保护
20 将在所附权利要求书中界定。



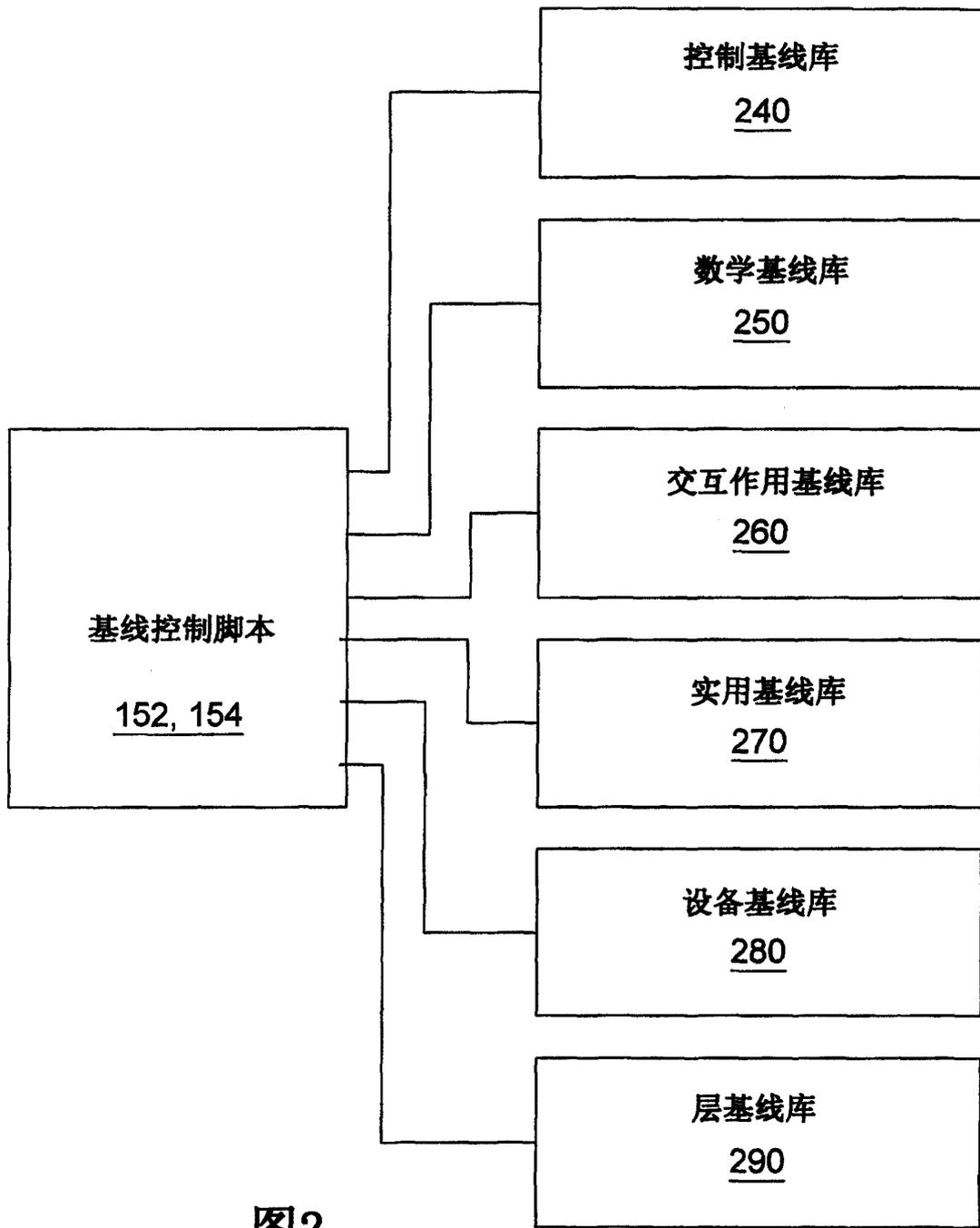


图2

152
↘

应用程序配置 <u>300</u>
基线应用设置 <u>310</u>
控制器常量及 上下文特定设置 <u>320</u>
前馈数据分析 <u>330</u>
控制线程 <u>340</u>
危险 <u>350</u>
控制动作和业务规则 <u>360</u>
结果 <u>370</u>

图3

154



度量工具设置 <u>400</u>
应用程序配置 <u>410</u>
基线应用设置 <u>420</u>
收入工具数据 <u>430</u>
控制器常量及 上下文特定设置 <u>440</u>
控制线程 <u>450</u>
模型更新 <u>460</u>
结果 <u>470</u>

图4

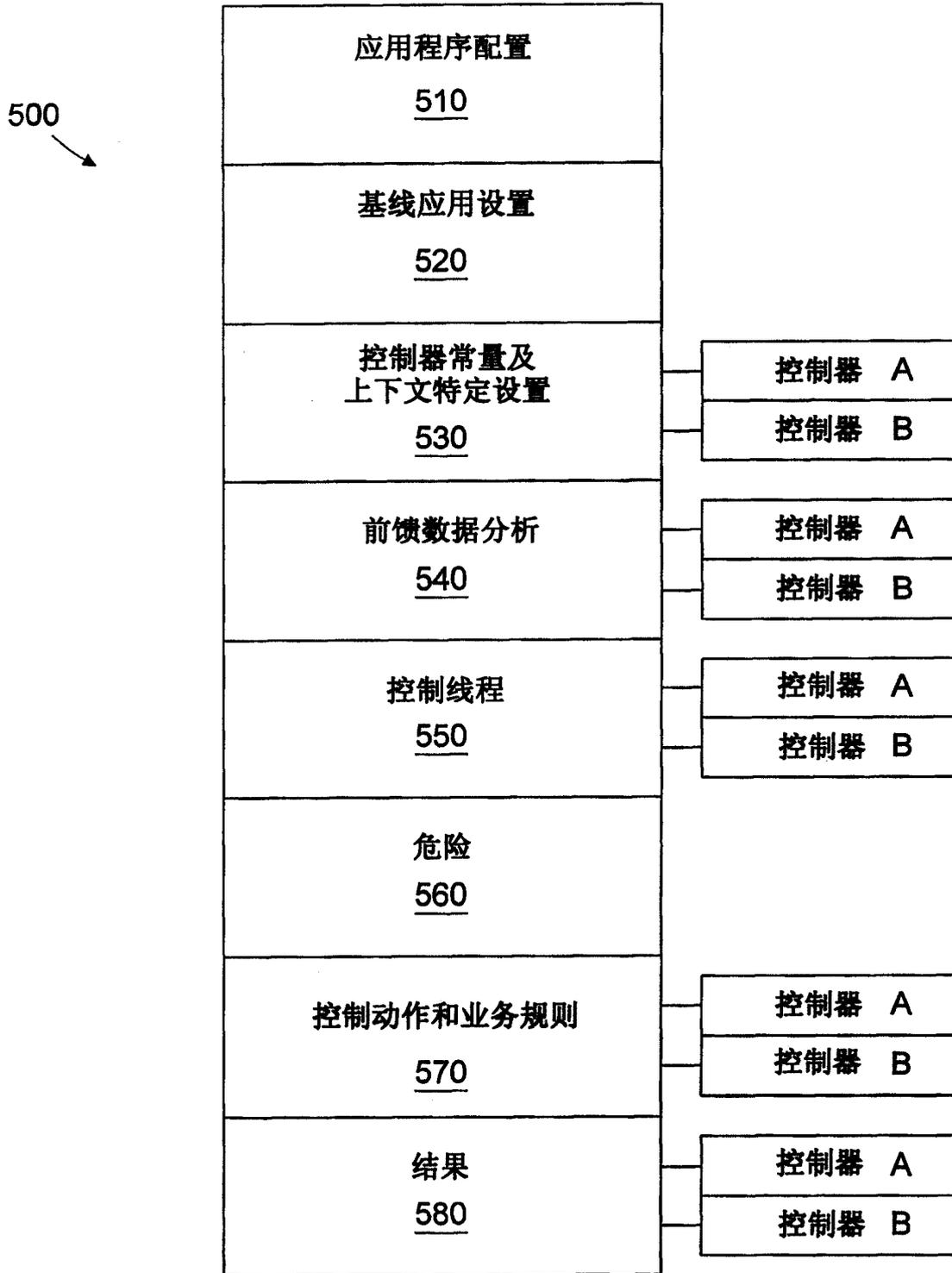


图5

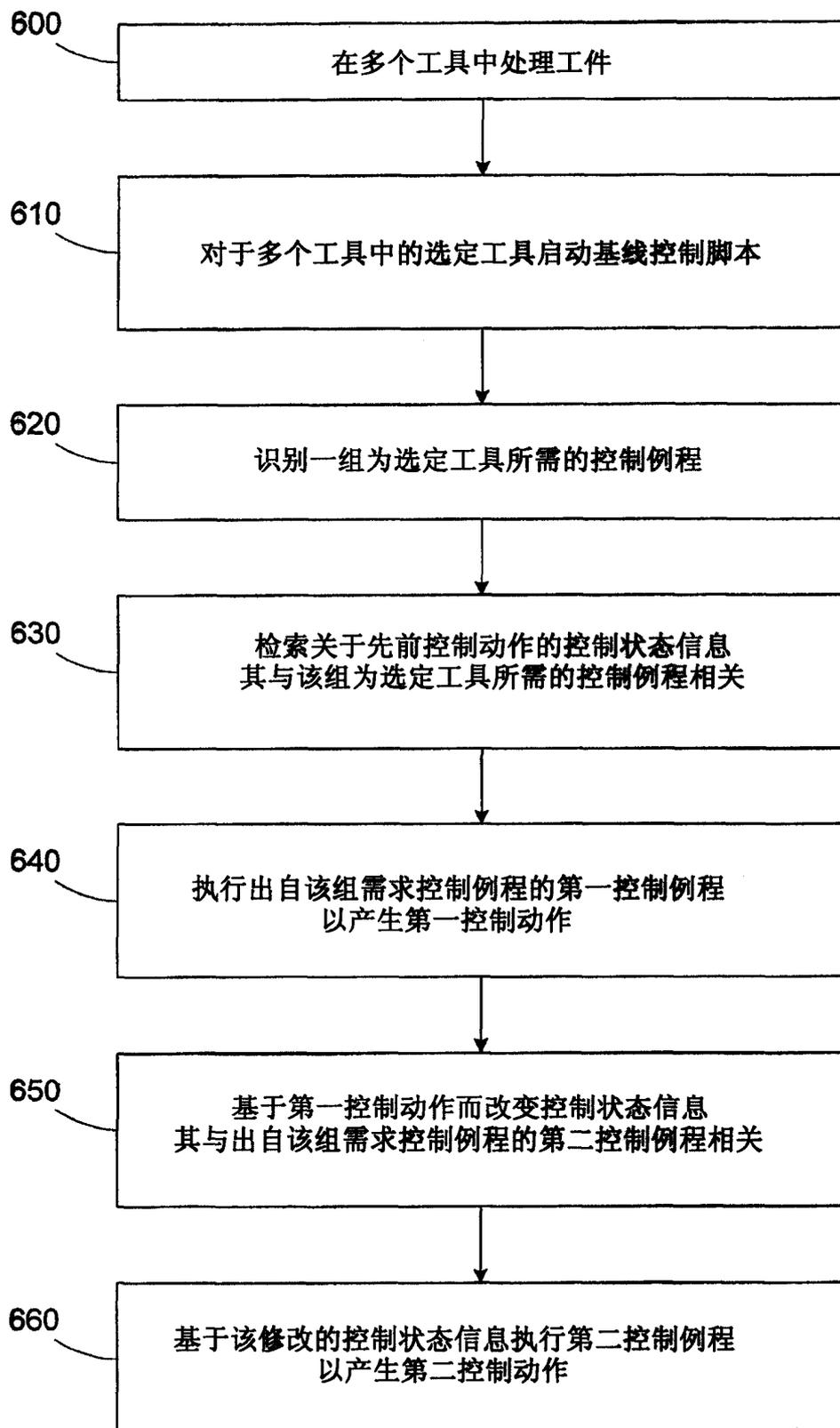


图6