

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2011-187874  
(P2011-187874A)

(43) 公開日 平成23年9月22日 (2011.9.22)

(51) Int.Cl.	F I	テーマコード (参考)
HO 1 L 21/027 (2006.01)	HO 1 L 21/30 5 1 6 Z	5 B 0 4 2
GO 6 F 11/30 (2006.01)	GO 6 F 11/30 E	5 F 0 4 6
HO 1 L 21/02 (2006.01)	HO 1 L 21/30 5 0 2 G	5 F 1 4 6
	HO 1 L 21/30 5 1 4 E	
	HO 1 L 21/02 Z	

審査請求 未請求 請求項の数 4 O L (全 16 頁)

(21) 出願番号 特願2010-54300 (P2010-54300)  
(22) 出願日 平成22年3月11日 (2010.3.11)

(71) 出願人 000001007  
キヤノン株式会社  
東京都大田区下丸子3丁目30番2号  
(74) 代理人 100126240  
弁理士 阿部 琢磨  
(74) 代理人 100124442  
弁理士 黒岩 創吾  
(72) 発明者 西巻 明美  
東京都大田区下丸子三丁目30番2号 キ  
ヤノン株式会社内  
Fターム(参考) 5B042 JJ03  
5F046 AA17 CD01 DD03 DD06  
5F146 AA17 CD01 DD03 DD06

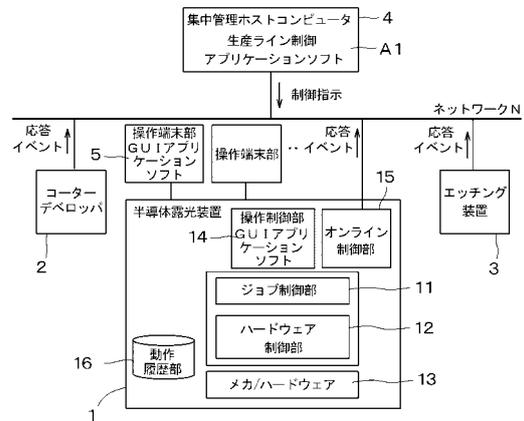
(54) 【発明の名称】 半導体露光システム

(57) 【要約】

【課題】半導体露光装置において、アプリケーションソフトのプロセスが異常状態になった場合に、装置全体を再起動せずに装置のダウンタイムを最低限にし生産量への影響を軽減する。

【解決手段】アプリケーションソフトのプロセスが異常状態になった場合に、生産処理状況を考慮して局所的なプロセスの再起動を行う。更に、処理の実行単位であるジョブにプロセスを関連させて対応情報を持つことによって、プロセス構成が可変のシステムにも対応する。

【選択図】 図 1



**【特許請求の範囲】****【請求項 1】**

複数のアプリケーションソフトから構成するシステムであって、プロセス間の関連とプロセス異常を監視し、前記プロセスの異常状態検出時に装置状態によってプロセスの状態変更タイミングを決定するプロセス管理手段と、ジョブ実行に関わる前記プロセスとジョブを対応して管理するジョブ管理手段とから構成することを特徴とする半導体露光システム。

**【請求項 2】**

前記プロセス管理手段は実行中のジョブ特性により、前記プロセスの状態変更タイミングを決定することを特徴とする請求項 1 に記載の半導体露光システム。

10

**【請求項 3】**

複数のアプリケーションソフトから構成するシステムであって、プロセス間の関連とプロセスの異常を監視するプロセス管理手段と、ジョブ実行に関わるプロセスとジョブを対応して管理するジョブ管理手段と、前記プロセスの異常状態検出時のジョブの継続処理に必要な前記プロセスを特定してプロセス状態を変更しジョブを継続処理するジョブ実行手段とから構成し、ロット内のウェハ状態を均一にすることを特徴とする半導体露光システム。

**【請求項 4】**

前記ジョブ実行手段は、継続ジョブの実行時にジョブパラメータを更新し、継続ジョブの処理時間を短縮することを特徴とする請求項 3 に記載の半導体露光システム。

20

**【発明の詳細な説明】****【技術分野】****【0001】**

本発明は、アプリケーションソフトのプロセスが異常状態に陥った場合等の処理を行い得る半導体露光システムに関するものである。

**【背景技術】****【0002】**

図 19 は従来からの半導体製造ラインにおける半導体生産工程の説明図である。(a) は露光開始の状態である。前工程の基盤洗浄、蒸着、レジスト塗布、露光、現像、現像後検査、エッチング、エッチング後検査、レジスト剥離、洗浄、イオン注入、ウェハプローブ検査の各工程から成り、これらの工程を経ることによって半導体を生産する。

30

**【0003】**

従来、露光工程を担う半導体露光装置は、アプリケーションソフト障害によって露光処理が中断又は停止すると、システム全体を再起動している。しかし、システムの再起動はハードウェアや機械装置の初期化動作を伴うため、再起動から定常状態になるまでの時間が長い。また、前後の連携システムと不整合を生じ、調整にも時間を要するため、装置の再起動は頻繁に実施できない。

**【0004】**

これを解決するために、障害の発生したプロセスを特定して、そのプロセスを正常状態に戻し、装置全体の再起動をせずに半導体露光装置を定常状態に移行することが考えられる。

40

**【0005】**

ところで、露光処理の単位はロットであり、生産ライン全体では、ロットを処理単位として生産計画や生産管理が行われ、1ロットは複数のウェハから成る。半導体露光装置はウェハを1枚ずつ認識できるが、製造ラインを構成する他の装置群はウェハ単位の認識ができるとは限らない。そのため、装置を含む生産ライン全体の共通単位はロットとなり、ロットを構成するウェハは物理的にはキャリアと呼ばれる入れ物に入る。

**【0006】**

エッチング前の検査工程でエラーが検出されると、エラーが検出されたロットはリワークにかけられる。リワークはこれまでウェハに施された処理層を削除して、処理が施され

50

る前の状態にする。リワーク後のウェハは改めてレジスト、露光等の処理が施される。リワークは上述のような生産ラインでは、管理単位であるロットで処理されることが常である。また、エッチング後の検査工程でエラーが検出されると、それらはスクラップされる。

#### 【0007】

(b)は露光工程での停止処理の説明図である。キャリアからウェハが1枚ずつ装置内に搬入されて、ウェハの位置合わせ、傾き調整や計測などの処理を行い、露光ステージまでウェハが搬送される。ステージに至る前のウェハは未露光状態である。ステージにウェハが到達すると、更に計測や位置合わせが行われ、その後レチクルパターンを露光する。露光はレイアウトパターンに従ってショット単位で行われる。

10

#### 【0008】

露光処理の途中で処理が止まってしまった場合は、全パターンの露光がなされていない半露光ウェハができる。全ショットの露光が完了したウェハは露光済み状態となり、半導体露光装置から搬出されてキャリアに入る。ここで、半導体露光装置に障害が発生すると、ウェハは様々な位置で停止するため、未露光、半露光、露光済みの3つの異なる状態のウェハが製造されてしまう。

#### 【0009】

(c)は障害が発生した場合の露光処理の停止状態である。半導体露光装置内のウェハ群は、障害発生直前の処理位置に留まる。障害が発生すると、装置によっては自動でウェハを搬出する機能も作用するが、全て確実に自動搬出されない場合は手作業で装置から取り出すか、半導体露光装置を再起動後にウェハを搬出する。このように、通常とは異なる方法で装置外に搬出してキャリアに戻す。このような状況下では、キャリア内は複数状態のウェハが混在する。

20

#### 【0010】

(d)は先の複数状態のウェハが混在したキャリアをリワークのために状態毎に手作業で仕分けした状態である。リワーク処理には削り取り強度のパラメータを設定でき、設定パラメータに依存して、ウェハから層を取り去る際の強度が変化する。同一状態のウェハが収まっているキャリアに対しては、適切に設定したパラメータによって全てのウェハがダメージなくリワークできる。

#### 【0011】

ところで、未露光、露光済みの複数状態のウェハが格納されたキャリアに対して、一度にリワークを実施する場合を考える。露光済み状態のウェハに合わせてリワークパラメータを設定すると、露光済み状態のウェハは最適な状態になるが、未露光状態のウェハはリワーク処理が強過ぎるためダメージを受けてしまう。また、未露光ウェハにパラメータを合わせた場合には、未露光状態のウェハは最適な状態になるが、露光済み状態のウェハはリワーク処理が弱いため、加工層が完全に取り除かれず再利用できない。

30

#### 【0012】

ダメージを受けたウェハや加工層が残ったウェハは、再度製造工程を経た場合に検査工程でエラーになる可能性が高い。従って、単一状態でウェハをまとめてロットを造り、そのロットに合致するリワークパラメータを設定してリワークを実施する必要がある。

40

#### 【0013】

上述のように、従来はロットの露光途中で露光処理が停止してしまった場合には、ロット内のウェハは加工済みウェハ、半加工ウェハ、未加工ウェハの3つの状態になる。リワーク後のウェハの歩留まりを上げるためには、各状態でウェハを仕分けして、状態毎にリワークを行う必要がある。ウェハの状態は見た目では区別がつかないため、仕分け作業は半導体露光装置のログやマテリアルの位置情報を考慮して加工度合いを判断する。しかし、この判断作業は解析時間がかかり、かつ精度が低い。

#### 【0014】

例えば特許文献1、2には、複数のアプリケーションソフトのプロセスから構成されるシステムが、障害を検知して、プロセスの再起動制御を行う技術が提案されている。これ

50

らの特許文献 1、2 は、分散サーバシステムやアプリケーションソフトのプロセス管理システムに関するものである。プロセスの障害発生時に全プロセスを再起動するのではなく、必要最低限のプロセスを再開することによって、全サービスが提供できなくなる状態を避けることが必要である。これを実現するために、関連のあるプロセス間でグループを予め定義した情報に従って、プロセスの再起動制御を行っている。

【先行技術文献】

【特許文献】

【0015】

【特許文献 1】特開 2004 - 265129 号公報

【特許文献 2】特開 2004 - 102492 号公報

10

【発明の概要】

【発明が解決しようとする課題】

【0016】

従来のプロセス再起動の方法は、固定的にプロセスをグループ化し、グループ単位でプロセスの再起動処理を行っている。一方、半導体露光装置のような顧客カスタマイズが必要な装置は、顧客要望の対応を迅速に行うために、装置を構成するアプリケーションソフトが随時に更新/追加されるようになる。そのため、プロセス構成を固定的に決められず、プロセス構成が随時変更されるようなシステムでは、予めプロセスグループを定義することができず、従来技術では適応できない。

【0017】

20

更に、障害発生時のロットは、最終的にはリワーク処置されることが分かっているため、半導体としての製品品質や精度は問題とせず、ロット内のウェハ特性を一致させ、リワークがより簡易にできるようにすることが望ましい。

【0018】

本発明の目的は、上述の問題点を解消し、異常状態となった場合に容易に再起動できる半導体露光システムを提供することにある。

【課題を解決するための手段】

【0019】

上記目的を達成するための本発明に係る半導体露光システムは、複数のアプリケーションソフトから構成するシステムであって、プロセス間の関連とプロセス異常を監視し、前記プロセスの異常状態検出時に装置状態によってプロセスの状態変更タイミングを決定するプロセス管理手段と、ジョブ実行に関わる前記プロセスとジョブを対応して管理するジョブ管理手段とから構成することを特徴とする。

30

【0020】

本発明に係る半導体露光システムは、複数のアプリケーションソフトから構成するシステムであって、プロセス間の関連とプロセスの異常を監視するプロセス管理手段と、ジョブ実行に関わるプロセスとジョブを対応して管理するジョブ管理手段と、前記プロセスの異常状態検出時のジョブの継続処理に必要な前記プロセスを特定してプロセス状態を変更しジョブを継続処理するジョブ実行手段とから構成し、ロット内のウェハ状態を均一にすることを特徴とする。

40

【発明の効果】

【0021】

本発明に係る半導体露光システムによれば、アプリケーションソフトのプロセスが異常状態になった場合に、装置のダウンタイムを最低限にし生産量への影響を軽減している。

【0022】

また、処理の実行単位であるジョブにプロセスを関連させて対応情報を持ち、露光処理を実現するジョブに関連しているプロセスを特定してプロセスの状態変更を行い、ロット内のウェハ特定を均一にしてリワーク時の作業負荷や管理負荷を軽減が可能となる。

【0023】

エラーリカバリ方法では処理の実行単位であるジョブにプロセスを関連させて対応情報

50

を持ち、プロセス構成が可変のシステムにも対応させ、装置のダウンタイムを最低限にして生産量への影響を低減できる。

【図面の簡単な説明】

【0024】

【図1】半導体製造ラインの構成図である。

【図2】ジョブ制御部の構成図である。

【図3】ジョブ-プロセス対応表である。

【図4】ジョブの状態遷移図である。

【図5】ジョブ種のプロパティ表である。

【図6】ジョブスケジューラの動作フローチャート図である。

10

【図7】ジョブパラメータ表である。

【図8】ジョブ実行のシーケンス図である。

【図9】プロセスの状態遷移表である。

【図10】プロセス情報表である。

【図11】異常検出の動作フローチャート図である。

【図12】プロセス間関連表である。

【図13】プロセス異常検出後の動作フローチャート図である。

【図14】プロセス状態変更処理の動作フローチャート図である。

【図15】実施例2の工程図である。

【図16】プロセス異常検出のシーケンス図である。

20

【図17】継続ジョブ投入処理の動作フローチャート図である。

【図18】テンプレートパラメータ特性定義表である。

【図19】従来の製造工程の工程図である。

【発明を実施するための形態】

【0025】

本発明を図1～図18に図示の実施例に基づいて詳細に説明する。

【実施例1】

【0026】

図1は半導体露光システムの構成図を示している。半導体露光装置を含む半導体製造ラインにおいては、半導体露光装置1はコーターデベロッパ2、エッチング装置3、生産ラインのアプリケーションソフトA1を搭載する集中管理ホストコンピュータ4などの各種装置とネットワークNにより接続されている。また、半導体露光装置1には複数の作業者が装置を操作するために、アプリケーションソフトを搭載した複数の操作端末部5が接続されている。

30

【0027】

半導体露光装置1はジョブ制御部11、ハードウェア制御部12、メカ/ハードウェア13、操作端末部5と同等のアプリケーションソフトを搭載した操作制御部14、オンライン制御部15、各種動作の履歴を保持する動作履歴部16から構成されている。メカ/ハードウェア13は実行に伴う各種処理を実行し、ジョブ制御部11はこの実行を制御し、操作制御部14は装置状態の監視、露光処理等を指示する。また、オンライン制御部15は生産ライン全体を統括するアプリケーションソフトA1から指示を受けて動作する。

40

【0028】

操作者が露光処理や各種の計測処理、装置状態の取得のために、操作制御部14のGUI(グラフィックユーザインタフェース)を操作すると、操作内容はコマンドに変換され、ジョブ制御部11に送信される。ジョブ制御部11はこのコマンドを解析し、コマンドをジョブとして実行する。ジョブは状態を持っており、ジョブ実行状況に応じて、実行中にエラー解除待ち、終了処理中のように状態が推移する。ジョブ制御部11のジョブ実行によって、ハードウェア制御部12がハードウェアレジスタに値を設定し、割り込み処理やデータ転送処理などを行う。

【0029】

50

動作履歴部 16 は障害発生時の原因解析データや、消耗品の交換時期予測のための予測データとして利用される。動作履歴部 16 には、システム状態や、操作制御部 14 及び生産ラインアプリケーションソフト A 1 からの機能実行指示内容と、それに伴う応答やイベント内容がタイムスタンプと共に保持 / 管理されている。

【 0030 】

図 2 は半導体露光装置 1 のジョブ制御部 11 の構成図を示している。ジョブ制御部 11 は実際のジョブ実行を担うジョブ実行部 21、ジョブの実行スケジュールやジョブの実行状態の監視を行うジョブ管理部 22、ジョブ制御部 11 を構成するプロセス群を管理するプロセス管理部 23 から構成されている。

【 0031 】

ジョブ実行部 21 は露光処理に必要な各処理要素から構成され、レチクル搬送 / ウェハ搬送 / 各種調整 / 各種計測 / 露光処理がある。ジョブの入力パラメータに従って、処理すべき要素を選択し処理が施される。実際の処理は、メカ / ハードウェア 13 に対してシグナル送信やアドレスへの命令書き込み制御を行ってメカやハードウェアを動作させる。データベースやファイルなどの管理データを作成 / 更新して処理を進めることもある。各処理要素はそれを担うプロセスから構成されており、処理依頼の際に各プロセスにジョブ識別子と処理パラメータを渡す。

【 0032 】

ジョブ管理部 22 はジョブキュー 22 a、ジョブスケジューラ 22 b、ジョブプロパティ定義部 22 c、パラメータ生成部 22 d、ジョブ履歴部 22 e から構成され、アプリケーションソフトから処理が依頼されると依頼の単位でジョブインスタンス J I を生成する。ジョブインスタンス J I が生成されると、ジョブが実行可能な状態になるまでジョブキュー 22 a に積まれる。ジョブスケジューラ 22 b はジョブプロパティ定義部 22 c で予め定義されるジョブの優先順位や、ジョブが利用するハードウェアやアプリケーションソフト資源を考慮して、図 3 に示すジョブ - プロセス対応表 L 1 から実行可能なジョブを選択する。

【 0033 】

パラメータ生成部 22 d はアプリケーションソフトから依頼された処理内容をジョブの入力パラメータとして定義する。これらのパラメータはジョブ実行に必要な前提条件やジョブ実行方法の指示としてジョブ実行部 21 で扱われる。ジョブ履歴部 22 e はジョブの実行状況や計測 / 露光処理結果を記録するものであり、アプリケーションソフトからのジョブ実行状況の問い合わせ応答やジョブの履歴参照に利用される。

【 0034 】

プロセス管理部 23 はライフサイクル管理部 23 a、異常検出部 23 b、プロセス間関連表 L 7 から構成されている。ライフサイクル管理部 23 a はプロセスのライフサイクルである生成や消滅、状態変更タイミングを制御し、異常検出部 23 b はプロセス状態の異常を検知し、プロセス間の関連を監視し、後述するプロセス間関連表は関連情報を保持する。

【 0035 】

ジョブ管理部 22 のジョブインスタンス J I は状態を持っており、図 4 に示すジョブの状態遷移関連表 L 2 のように遷移する。ジョブは生成時にジョブを識別するためのジョブ識別子が付与される。ジョブキュー 22 a にジョブ識別子をエントリすると、ジョブは実行待ち状態になる。スケジューラによって実行が許可されるとジョブは実行中状態になり、ジョブ実行部 21 によって露光処理や各種計測処理が行われる。ジョブ実行部 21 によって要求の処理が完了すると、ジョブは実行完了状態に遷移後に消滅する。また、アプリケーションソフトからの指示によって一時停止や復帰動作を行う。このときに、実行状態から一時停止状態に、一時停止状態から実行状態に遷移する。更に、ハードウェアの調整などによって解除可能な障害が発生した場合には、障害解除待ち状態に遷移し、障害が取り除かれた後に実行状態に遷移する。

【 0036 】

10

20

30

40

50

ジョブスケジューラ 2 2 b はジョブキュー 2 2 a にエントリしているジョブの実行可否を判断する。図 5 はジョブプロパティ定義部 2 2 c のジョブ種のプロパティ表 L 3 を示し、ジョブ実行の事前条件やジョブ特性を定義している。例えば、この図 5 によればジョブ種が露光ジョブの場合は「半導体生産」というジョブ分類に属し、2 番目に高い優先度で処理される。更に、「計測センサ A、計測センサ B、計測センサ C、ステージ、搬送パス W」の資源が利用されていない場合に実行開始でき、かつ露光ジョブがこの資源を占有することを示す。そして露光ジョブは複数同時に実行できないという制約を持っている。予め定義されるジョブ種のプロパティ表 L 3 に従って、ジョブスケジューラがジョブの実行開始を判断する。

#### 【 0 0 3 7 】

図 6 はジョブ実行判断の動作フローチャート図であり、先ず、ジョブキュー 2 2 a にジョブが存在するか確認し ( S 1 0 1 )、ジョブが存在しない場合は判断処理を終了する。ジョブが存在した場合はジョブの利用資源が他のジョブによって占有されていないか、ハードウェア故障を起していないかを確認する ( S 1 0 2 )。利用可能な場合に、同一ジョブ種のジョブが実行中か否かを確認し ( S 1 0 3 )、実行していなければ実行候補のジョブリストに登録する ( S 1 0 4 )。仮に、同一ジョブ種のジョブが実行中であっても、並行動作が許可されていれば ( S 1 0 5 )、同様に実行候補のジョブリストへ登録する ( S 1 0 4 )。

#### 【 0 0 3 8 】

ジョブキュー 2 2 a 内に存在する全てのジョブに対して、S 1 0 1 ~ S 1 0 5 のステップを実施し ( S 1 0 6 )、実行候補のジョブリストを作成する。実行候補のジョブリストから優先順位の最も高いジョブを取得し ( S 1 0 7 )、その中から最先にジョブキュー 2 2 a にエントリしたジョブを実行対象として確定し、ジョブ実行部 2 1 にジョブ実行を依頼する ( S 1 0 8 )。

#### 【 0 0 3 9 】

このように、ジョブスケジューラ 2 2 b はジョブキュー 2 2 a 内から実行開始可能なジョブを選択する。この判断処理はタイマなどによって定期的実施される。

#### 【 0 0 4 0 】

図 7 はジョブ実行前にパラメータ生成部 2 2 d が生成するジョブパラメータ表 L 4 の例を示す。パラメータ生成部 2 2 d は露光時のショットパターンの基になるレチクル情報やレイアウト情報、位置計測や傾き計測のためのマーク情報、露光時の処理内容を決める露光情報、実際に露光を施すウェハの情報のような複数のパラメータを定義する。

#### 【 0 0 4 1 】

図 8 はジョブ管理部 2 2 からジョブ実行部 2 1 にジョブ実行を依頼する際のシーケンス図を示している。アプリケーションソフト A 1 からジョブ管理部 2 2 に処理要求が入力されると ( S 2 0 1 )、ジョブ管理部 2 2 はジョブパラメータとジョブインスタンス J I を作成する ( S 2 0 2 )。作成したジョブインスタンス J I はジョブキュー 2 2 a に登録される。ジョブスケジューラ 2 2 b は定期的にジョブキュー 2 2 a を監視しており、実行可能なジョブを検出すると、ジョブ実行部 2 1 にジョブの実行依頼をする ( S 2 0 3 )。

#### 【 0 0 4 2 】

ジョブ実行部 2 1 は受け取ったジョブのパラメータを解析して、必要な処理の要素と処理順番を決定する ( S 2 0 4 )。決定した処理順番に従って、処理要素へ実行を依頼する ( S 2 0 5 )。処理の要素、例えば「レチクル搬送」は、処理依頼時に受け取ったジョブ識別子と自己のプロセス識別子を図 3 のジョブ - プロセス対応表 L 1 に登録する ( S 2 0 6 )。そして、処理パラメータに従ってレチクルの搬送処理を行い ( S 2 0 7 )、処理が完了するとジョブ - プロセス対応表 L 1 から先に登録した自己に関する内容を削除する ( S 2 0 8 )。処理が完了すると処理完了を通知し、「レチクル搬送」プロセスは処理依頼待ち状態になる ( S 2 0 9 )。

#### 【 0 0 4 3 】

ジョブ実行部 2 1 は全ての処理要素を実施し、全ての処理が終了するとジョブ管理部 2

10

20

30

40

50

2 にジョブ完了を通知する ( S 2 1 0 ) 。ジョブ管理部 2 2 はジョブ - プロセス対応表 L 1 から当該ジョブを削除する ( S 2 1 1 ) 。ジョブ管理部 2 2 はアプリケーションソフトに対して要求動作の終了を通知する ( S 2 1 2 ) 。

【 0 0 4 4 】

図 9 はプロセスの状態遷移表 L 5 を示している。プロセスは実行可能な R e a d y 状態。実際に CPU を割り当てられている r u n n i n g 状態。ディスク I O 処理の完了待ちやシグナル受信待ちの w a i t 状態。サスペンドシグナルを送られた S t o p p e d 状態。プロセスを終了したがメモリ上の管理情報が残っている Z O M B I E 状態から成っている。

【 0 0 4 5 】

ライフサイクル管理部 2 3 a がシステム起動直後に生成する常駐プロセスやジョブ実行に伴って動的に生成が必要な非常駐プロセスを生成する。異常検出部 2 3 b はプロセス状態の異常を検出し、必要な常駐プロセスが存在しないことや、プロセスが Z O M B I E 状態にあることを検出する。

【 0 0 4 6 】

図 1 0 は L i n u x ( 登録商標 ) システムから取得可能なプロセス情報表 L 6 の例である。プロセスのユーザ情報 ( U S E R ) やプロセス ID ( P I D ) 、プロセス状態 ( S T A T ) 、スタート時間 ( S T A R T T I M E ) 、コマンド名 ( C O M M A N D ) といった各種情報が OS システムから取得できる。このプロセス情報表 L 6 を用いて、異常状態検出手段はプロセス数の変化やプロセス状態の変化を知る。特に、プロセス状態を示す S T A T は、「 R 」や「 S 」のアルファベット記号で r u n n i n g 状態や R e a d y 状態を示している。

【 0 0 4 7 】

図 1 1 は異常検出部 2 3 b の異常検出動作フローチャート図である。まず、プロセス管理部 2 3 a が保持する常駐プロセスのリストを取得する ( S 3 0 1 ) 。そして、OS システムのプロセス監視機構を用いて現在のプロセスリストを取得する ( S 3 0 2 ) 。現在のプロセスリストに常駐プロセスが存在するか否かを確認し ( S 3 0 3 ) 、もし常駐プロセスが存在しなければ、異常プロセスリストに該当する常駐プロセス名を登録する ( S 3 0 4 ) 。 S 3 0 3 の常駐プロセスの存在チェックを繰り返し実行し、全常駐プロセスに対して存在確認を行う ( S 3 0 5 ) 。

【 0 0 4 8 】

次に、現在のプロセスリストに対して、Z O M B I プロセスの有無を確認する。プロセス監視機構から取得したプロセス情報のステータス情報に、Z O M B I E 状態を示す「 Z 」が記載されていた場合は ( S 3 0 6 ) 、そのプロセスは異常状態になっていると見做して異常プロセスリストに登録する ( S 3 0 7 ) 。 S 3 0 6 の Z O M B I プロセスチェックを繰り返し実行し、OS システムが管理している全プロセスの状態の確認を行う ( S 3 0 8 ) 。常駐プロセスの存在と、現在動作しているプロセス状態の確認を完了すると、異常プロセスをプロセス管理部 2 3 に通知する ( S 3 0 9 ) 。この異常検出処理の S 3 0 1 ~ S 3 0 9 のステップをタイマにより定期的に行う。

【 0 0 4 9 】

図 1 2 はプロセス管理部 2 3 が有するプロセス間の関連を示すプロセス間関連表 L 7 の例を示す。複数のプロセスが通信によってメッセージを授受する本システムでは、通信ポートを設けてサービスを提供しているプロセスをサーバ、そのポートに接続をしてサービスの提供を受けているプロセスをクライアントとして管理する。プロセスが通信ポートを生成するとき、プロセス間関連表 L 7 にサーバ情報としてプロセス識別子とポート番号 ( サービス名 ) を登録する。また、プロセスがポートに接続したときは、プロセス間関連表 L 7 上の該当するポート情報に対応させて、クライアント情報としてプロセス識別子を登録する。

【 0 0 5 0 】

図 1 3 はプロセス異常検出後の処理動作フローチャート図であり、プロセスの異常を検

10

20

30

40

50

出した場合には、ジョブ管理部 2 2、ジョブ実行部 2 1、プロセス管理部 2 3 が連携してジョブの継続処理を行う。

【 0 0 5 1 】

先ず、プロセス管理部 2 3 がプロセス異常を検知すると、ジョブ管理部 2 2 から異常プロセスの関連ジョブのジョブ分類を取得する ( S 4 0 1 )。プロセス管理部 2 3 がジョブ管理部 2 2 に異常プロセスの関連ジョブの停止を依頼すると、ジョブ管理部 2 2 は異常プロセスの識別子を基にジョブ - プロセス対応表 L 1 から該当ジョブを検索して ( S 4 0 2 )、ジョブ実行部 2 1 がジョブを停止する ( S 4 0 3 )。S 4 0 1 によって取得したジョブ分類が、「半導体生産」であった場合は ( S 4 0 4 )、直ちに異常状態のプロセスの再起動や他の関連するプロセスを矛盾なく動作可能な状態に遷移させる ( S 4 0 9 )。

10

【 0 0 5 2 】

S 4 0 4 でジョブ分類が「半導体生産」以外の場合は、ジョブ管理部 2 2 から実行中ジョブの分類を取得する ( S 4 0 5 )。プロセス管理部 2 3 は実行中のジョブが「半導体生産」であるかどうかを判断し ( S 4 0 6 )、「半導体生産」に分類されるジョブが実行中であれば、ジョブ完了を待ってから ( S 4 0 7 )、S 4 0 9 のプロセス状態を変更する。全ての実行中のジョブに対してジョブ分類を判断し ( S 4 0 8 )、分類が「半導体生産」でない場合はプロセス状態を変更する ( S 4 0 9 )。

【 0 0 5 3 】

図 1 4 はプロセス管理部 2 3 のプロセス状態の変更処理 ( 図 1 3 の S 4 0 9 ) の動作フローチャート図である。異常が検出されたプロセスが Z O M B I E 状態の場合には ( S 5 0 1 )、当該プロセスを終了させる ( S 5 0 2 )。予め定義したプロセスのライフサイクル情報によって、当該プロセスとライフサイクルを共にするプロセスが存在するかを確認する ( S 5 0 3 )。ライフサイクルを共にするプロセスが存在する場合には、当該プロセスとライフサイクルを共にするプロセスを終了させ ( S 5 0 4 )、当該プロセスに加えてライフサイクルを共にするプロセスを起動する ( S 5 0 5 )。ライフサイクルを共にするプロセスが存在しない場合には、当該プロセスのみを起動する ( S 5 0 6 )。

20

【 0 0 5 4 】

当該プロセスが起動の際にサーバポートを生成した場合とすると ( S 5 0 7 )、プロセス間関連表 L 7 からポートに対するクライアントを特定する。 ( S 5 0 8 ) そして、クライアントプロセスに対して、サーバプロセスが再起動してポートが開設されたことを通知し ( S 5 0 9 )、プロセス間関連表 L 7 の内容を新たに起動したプロセス情報で更新する。クライアントはサーバプロセスのポートが開設されたことを知ると、サーバポートに再接続を行い、これらの一連の処理によって再度プロセス間のメッセージ通信が可能になる。

30

【 0 0 5 5 】

上述の実施例によれば、半導体露光装置 1 のエラーリカバリ方法は、アプリケーションソフトのプロセスが異常状態になった場合でも、装置全体の再起動をせずに局所的なプロセスの再起動によりダウンタイムを最低限にできる。更に、異常プロセスの再起動タイミングを制御することによって、半導体生産量への影響を最小に抑える。本実施例 1 では、露光処理を実現するジョブに関連しているプロセスを特定して、プロセスの状態変更を行うため、プロセス構成が可変のシステムにも対応できる。

40

【 実施例 2 】

【 0 0 5 6 】

図 1 5 は実施例 2 の半導体露光装置の生産工程の説明図である。ロット内のウェハ状態を均一にする半導体露光装置は、アプリケーションソフトのプロセスの異常状態が発生しても露光処理を継続して実行し、ロット内のウェハ状態を均一にする。そして、効率的なリワーク工程を実現する。

【 0 0 5 7 】

( a ) はアプリケーションソフトのプロセスの異常状態が発生したときの露光処理の停止状態と、その後の露光処理の継続状態を示している。半導体露光装置内のウェハ群は、

50

障害発生直前の処理位置に留まっている。様々な処理がなされている途中で異常が発生すると、ウェハは未加工状態、半加工状態、加工状態と複数の状態になる。

【0058】

本実施例2では、異常状態に陥ったアプリケーションソフトのプロセスを特定し、そのプロセスを正常状態に遷移させて再度露光処理を実施可能にする。

【0059】

(b)は継続して露光処理を実施した様子を示し、装置内に留まっているウェハやキャリア内ウェハに対して継続処理を行い、ロットを構成する全てのウェハに対して露光処理を施した状態にする。

【0060】

(c)はリワーク処理を示し、リワークはウェハ状態に応じて仕分け作業をすることなくリワーク作業が可能である。このように、ロット内の全ウェハを露光済み状態にすることによって、リワークのための仕分け作業やロットの再構成作業が不要となる。効率的なリワーク作業を実現する。

【0061】

実施例2のロット内のウェハ状態を均一にする半導体露光装置1のジョブ制御部11は実施例1の図2の構成と同様である。ジョブ実行部21はアプリケーションソフトのプロセスの異常状態検出時に、ジョブとプロセスの関係から状態変更が必要なプロセスを特定する。異常状態に陥ったプロセスが関連していたジョブを特定し、実行中のジョブを終了させる。異常状態のプロセスを正常状態に移行させ、先に終了させたジョブについてジョブパラメータを再生成して、継続処理を行うためのジョブを実行する。

【0062】

継続ジョブは未加工状態にあるウェハを露光処理し、異常状態発生前に処理されたウェハと同様の加工状態にする。プロセスの異常状態を解決して継続ジョブを実行する仕組みによって、ロット内のウェハの状態が均一になる。その結果、効率的なリワーク作業が可能となる。

【0063】

また、ジョブとプロセスの関連情報から状態変更対象のプロセスを特定するため、仮に装置内のプロセス構成が変更になったとしても影響を受けない。更に、アプリケーションソフトのプロセスの異常により停止してしまった露光処理のロットは、リワーク対象になるため、継続の露光処理は半導体製品製造のような精度が必要なく、速やかに処理できることが望ましい。そのため、継続用の露光処理ジョブのパラメータは製品精度を上げるために必要な計測などの処理を省き、高速でジョブを実行できるパラメータを選択する。このようなパラメータは、ジョブ実行前にジョブ管理部22のパラメータ生成部22dによって生成され、ジョブはこのパラメータに従った動作を行う。

【0064】

異常発生以降の継続処理は、ウェハ状態を同一にするための必要最低限の動作に留まるため、継続ジョブの実行時には通常処理よりも処理時間の短縮が可能となる。

【0065】

図16はプロセス異常検出後の処理動作のシーケンス図であり、プロセスの異常を検出した場合は、図2のジョブ管理部22、ジョブ実行部21、プロセス管理部23が連携してジョブの継続処理を行う。まず、プロセス管理部23がプロセス異常を検知すると、ジョブ管理部22に対して異常プロセスが関連するジョブの停止を依頼する(S601)。ジョブ管理部22はプロセス管理部23から受け取った異常プロセスの識別子を基に、ジョブ-プロセス対応表L1から該当するジョブを検索し(S602)、該当ジョブの入力パラメータとジョブの処理経過を示す履歴情報を保持する。そして、ジョブの停止処理をジョブ実行部21に依頼する(S603)。

【0066】

ジョブ実行部21はジョブ管理部22から受け取ったジョブ識別子と対応する処理を停止し(S604)、処理要素全てを停止する(S605)。ジョブ管理部22はジョブ停

10

20

30

40

50

止が完了したことをプロセス管理部 2 3 に通知する ( S 6 0 6 )。プロセス管理部 2 3 は異常状態にあるプロセスを再起動して定常状態にし、他の関連するプロセスについても矛盾なく動作可能な状態に遷移させる ( S 6 0 7 )。ジョブ管理部 2 2 がプロセス準備完了の通知を受信すると ( S 6 0 8 )、先に停止したジョブのパラメータを変更して継続ジョブをジョブキュー 2 2 a にエントリする ( S 6 0 9 )。このとき、図 5 のジョブ種のプロパティ表 L 3 に従って「継続露光ジョブ」は最優先でスケジューリングされ、ジョブ実行シーケンスに基づいてジョブが実行される。

#### 【 0 0 6 7 】

図 1 7 はジョブ管理部 2 2 の継続ジョブ投入の動作フローチャート図である。まず、パラメータ生成部 2 2 d がジョブの停止処理の際に、保持した対象ジョブの入力パラメータを取得する ( S 7 0 1 )。予め定義した露光処理の継続ジョブパラメータのテンプレートを取得し ( S 7 0 2 )、テンプレートパラメータの特性定義によって該当ジョブの入力パラメータを変更する ( S 7 0 3 ~ S 7 0 5 )。全ての露光処理パラメータに対してパラメータの更新を行い ( S 7 0 6 )、ジョブキュー 2 2 a に新たなジョブとしてエントリする ( S 7 0 7 )。

10

#### 【 0 0 6 8 】

図 1 8 はテンプレートパラメータ特性定義表 L 8 を示し、テンプレートパラメータは高速にかつ精度を追求せず、露光処理を実施するためのパラメータの特性を定義している。露光処理に必要な入力パラメータの 1 つ 1 つに対して、パラメータ特性として [ 無効 ]、[ 固定 ]、[ 継承 ] の何れかが定義される。無効は対象ジョブの入力パラメータの値をそのまま利用することを示し、固定はテンプレートパラメータの値を使用することを示し、入力パラメータの値は採用しない。継承はプロセス異常発生前に動作していたジョブの経過情報を考慮して新たにパラメータ値を設定することを意味する。経過情報はジョブ履歴部 2 2 e から取得する。

20

#### 【 0 0 6 9 】

例えば、ウェハ情報のウェハスタートインデックスは露光処理の開始ウェハのインデックス番号を示す。通常はキャリアの 1 枚目のウェハから処理を開始するため、本パラメータには「 1 」が設定される。テンプレートパラメータの定義によると、ウェハ情報のウェハスタートインデックスのパラメータ特性は継承である。露光処理をウェハの 1 枚目 ~ 5 枚目まで完了した段階でプロセス異常が生じた場合は、継続ジョブの露光開始ウェハは処理済ウェハを考慮して、6 枚目から開始となる。

30

#### 【 0 0 7 0 】

従って、ウェハのスタートインデックスパラメータには「 6 」が設定される。このように [ 継続 ] の特性を持つパラメータは、処理履歴を基に処理済み量を考慮して算出される。ジョブの入力パラメータが露光処理が 5 枚目完了時点で異常発生した場合は、図 7 の継続ジョブパラメータ表 L 4 のパラメータ特性を考慮して再設定される。

#### 【 0 0 7 1 】

本実施例 2 によれば、半導体露光装置はアプリケーションソフトのプロセスが異常状態になった場合でも、ロット内のウェハ特定を均一にして、リワーク時の作業負荷や管理負荷を軽減する。本実施例 2 では、露光処理を実現するジョブに関連しているプロセスを特定してプロセスの状態変更を行うため、プロセス構成が可変のシステムにも対応できる。

40

#### 【 0 0 7 2 】

更に、障害発生時のロットは最終的には、リワーク処置されることが分かっているため、半導体としての製品品質や精度は問題とせず、障害発生以降の処理はより高速に露光することによって、生産効率の向上が期待できる。

#### 【 符号の説明 】

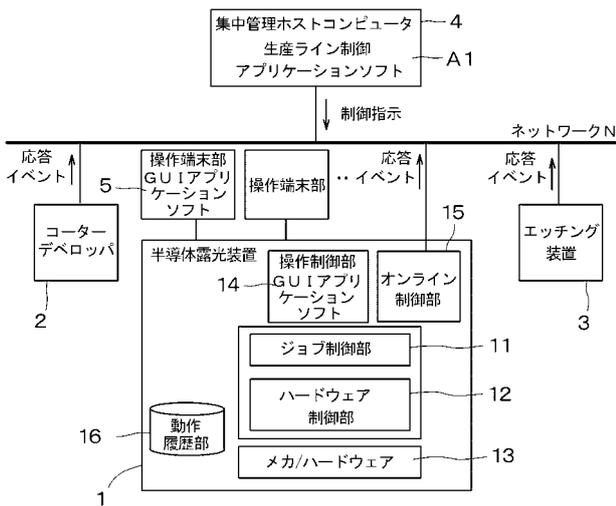
#### 【 0 0 7 3 】

- 1 半導体露光装置
- 2 コーターデベロッパ
- 3 エッチング装置

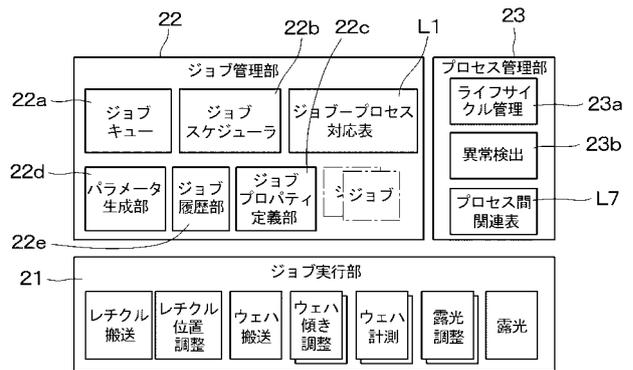
50

- 4 集中管理ホストコンピュータ
- 5 操作端末部
- 1 2 ハードウェア制御部
- 1 3 メカ/ハードウェア
- 1 4 操作制御部
- 1 5 オンライン制御部
- 1 6 動作履歴部
- 2 1 ジョブ実行部
- 2 2 ジョブ管理部
- 2 3 プロセス管理部

【 図 1 】



【 図 2 】



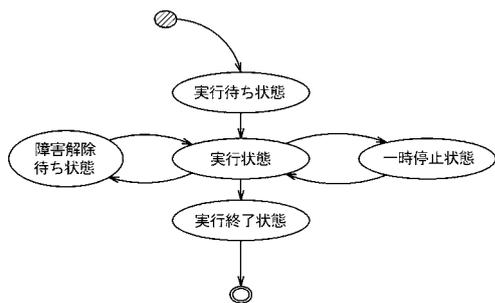
【 図 3 】

ジョブ-プロセス対応表 L 1

ジョブ識別子	プロセス識別子
1000	60001
1000	65000
1000	78787
1000	78789
1000	78790
1000	78791
1000	78792
1000	80011
1000	80012
1000	80013
1000	90000
1001	90000

【 図 4 】

ジョブの状態遷移関連表 L 2

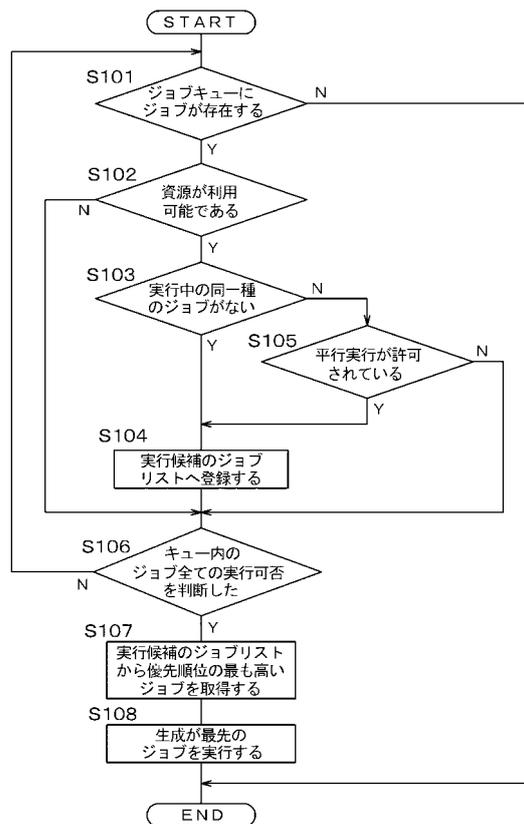


【 図 5 】

ジョブ種のプロパティ表 L 3

ジョブの種類	分類	優先順位	利用資源	同種ジョブの平行実行
継続露光ジョブ	半導体生産	1	計測センサA、計測センサB、計測センサC、ステージ、搬送ハスW	x
露光ジョブ	半導体生産	2	計測センサA、計測センサB、計測センサC、ステージ、搬送ハスW	x
ウェハ搬送ジョブ	マテリアル制御	3	搬送ハスW	○
レチクル搬送ジョブ	マテリアル制御	3	搬送ハスR	○
履歴参照ジョブ	装置管理	5		○
ウェハ位置計測ジョブ	マテリアル制御	3	計測センサA	x
ウェハ傾き計測ジョブ	マテリアル制御	3	計測センサB、計測センサC	x
レシビ更新ジョブ	生産定義	4	レシビDB	x
レシビコピージョブ	生産定義	5	レシビDB	○
レシビ削除ジョブ	生産定義	4	レシビDB	x
レーザー量調整	ハード/メカ調整	3	レーザーボリュームA	x
レンズ位置調整	ハード/メカ調整	3	レンズアームC	x

【 図 6 】

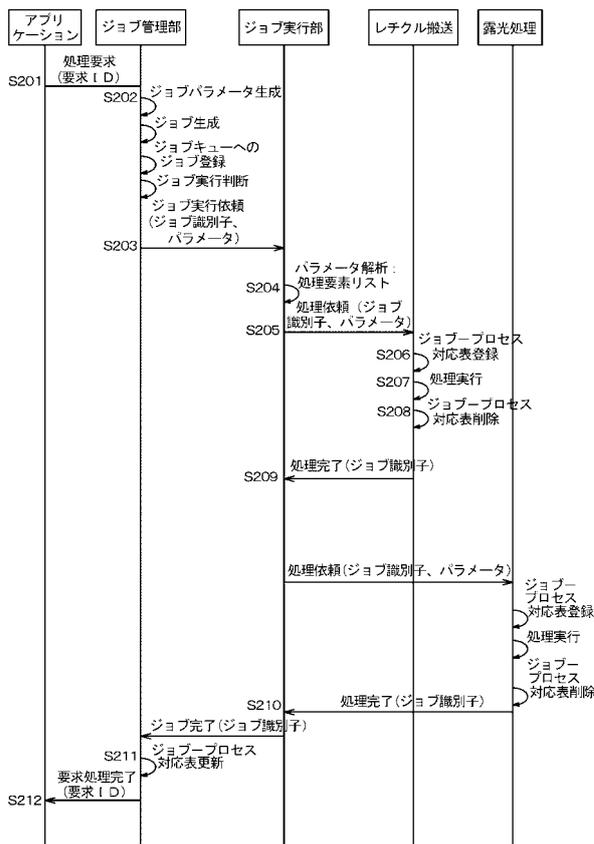


【 図 7 】

ジョブパラメータ表 L 4

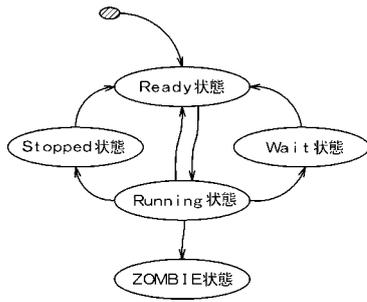
<b>レチクル情報</b>	
・レチクル名	first
・レチクルID	100
・デザイン	pattern100
<b>レイアウト情報</b>	
・サイズXY	100,100
・オリジンXY	5.5
・アングル	30
・ショット情報	256
<b>マーク情報</b>	
・マーク形状	pattern A
・マーク位置XY	12.12
<b>露光情報</b>	
・露光量	567
・露光前傾き計測	ON
・露光前ゆがみ計測	ON
・露光前位置計測	ON
・露光モード	Auto
・移動スピード	mid
・位置誤差補正	ON
・ゆがみ誤差補正	ON
<b>ウェハ情報</b>	
・ロット名	abc-1212
・ロットID	0123
・キャリアID	23,24
・ウェハ枚数	25
・ウェハサイズXYZ	400,400.3
・ウェハ種	global
・ウェハスタート Index	1

【 図 8 】



【 図 9 】

プロセスの状態遷移表 L5

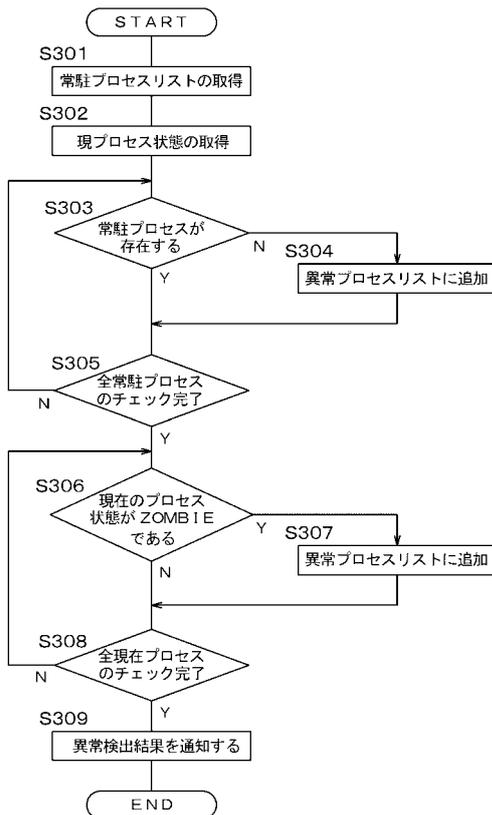


【 図 10 】

プロセス情報表 L6

USER	PID	STAT	START	TIME	COMMAND
jobpro	3961	S	Jun27	0:00	vncconfig-iconic
jobpro	3962	S	Jun27	0:00	xterm-geometry
jobpro	3963	S	Jun27	0:00	twm
root	3966	S	Jun27	0:00	/usr/sbin/atd
jobpro	4374	S	Jun27	0:00	/jobpro/bin/abc
jobpro	4376	S	Jun27	0:00	!tosh
jobpro	4704	S	Jun27	0:01	/jobpro/bin/def
root	26982	S	09:15	0:09	/jobpro/appli/bin/logsystem
jobpro	27003	S	09:15	0:00	/jobpro/appli/bin/logmon
jobpro	27238	S	09:16	0:05	/jobpro/com/bin/Communicator
jobpro	27250	S	09:16	0:08	/jobpro/com/bin/MessageWatcher
jobpro	27251	S	09:16	0:02	/jobpro/com/bin/MessageMonitor
jobpro	27252	S	09:16	0:19	/jobpro/com/bin/PortManager
root	27311	S	09:16	1:06	/jobpro/bin/jobController
jobpro	27312	S	09:16	0:00	/jobpro/bin/jobFactory
jobpro	27321	S	09:16	0:01	/jobpro/bin/Ameasure
jobpro	27336	S	09:16	0:00	/jobpro/bin/Wmeasure
jobpro	27342	S	09:16	0:03	/jobpro/bin/Rmeasure
jobpro	27351	S	09:16	0:02	/jobpro/bin/ReticleController
jobpro	27357	S	09:16	0:01	/jobpro/bin/WaferController
jobpro	27360	S	09:16	0:02	/jobpro/bin/Exposure
jobpro	27363	S	09:16	0:11	/jobpro/bin/RecipeController
jobpro	27366	S	09:16	0:01	/jobpro/bin/DBmanager
jobpro	27369	S	09:16	0:01	/jobpro/bin/ProcessController
jobpro	27372	S	09:16	0:00	/jobpro/bin/ProcessChecker

【 図 11 】

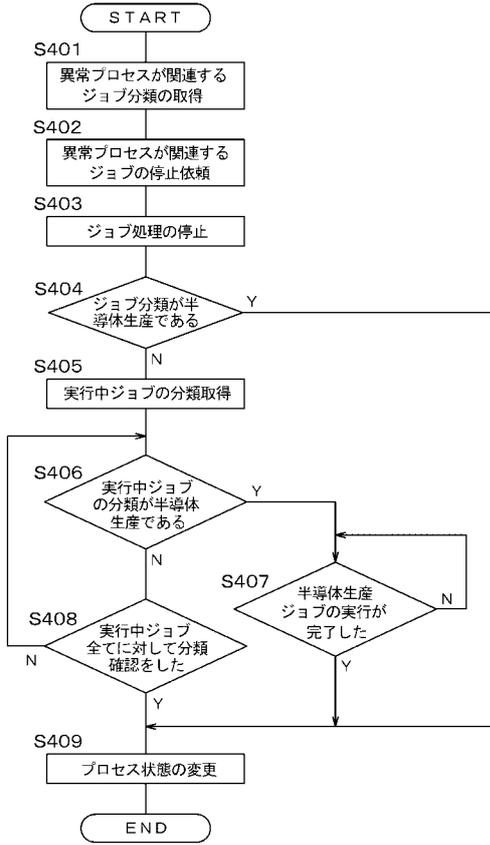


【 図 12 】

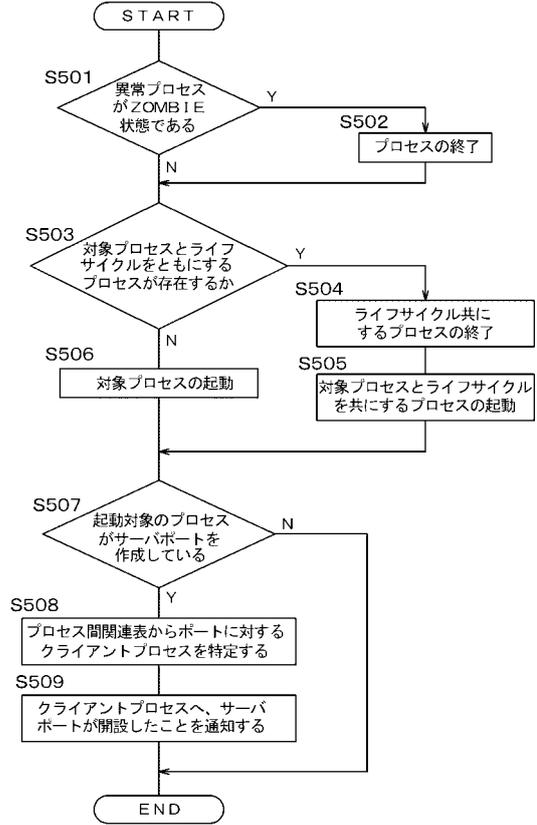
プロセス間関連表 L7

Server 情報		Client 情報
プロセス識別子	ポート(Service)	プロセス識別子
10000	121212 処理指示	60001
		65000
		78787
		78789
		78790
		78791
		78792
		80011
		80012
		80013
		90000
10000	121213 状態通知	60001
		65000
		78787
		78789
		78790
		78791
		78792
		80011
		80012
		80013
		90000
60001	151515 状態通知	65000
78787	232323 処理指示	78789
		78790
		78791
		78792
80011	343434 状態通知	80012
		80013

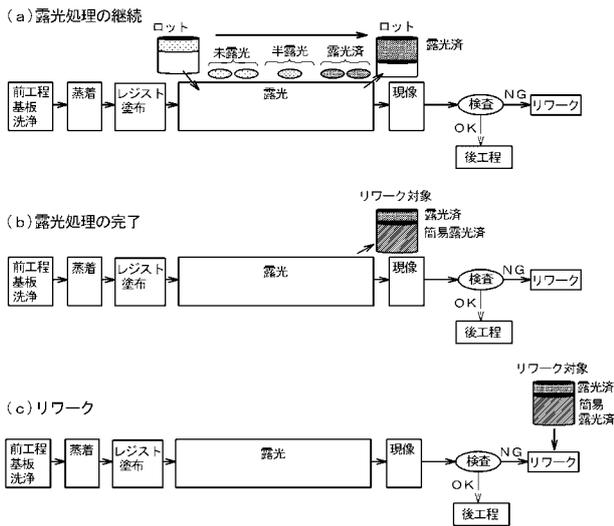
【 図 1 3 】



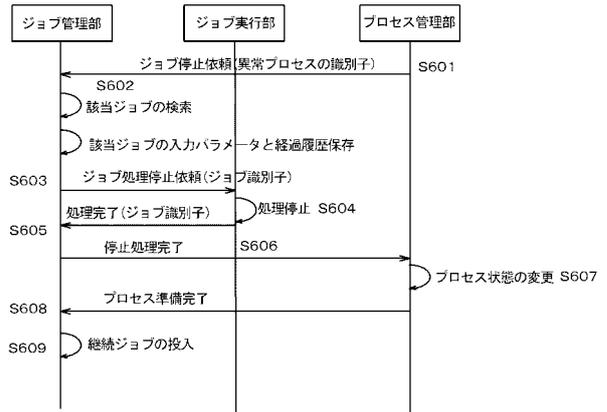
【 図 1 4 】



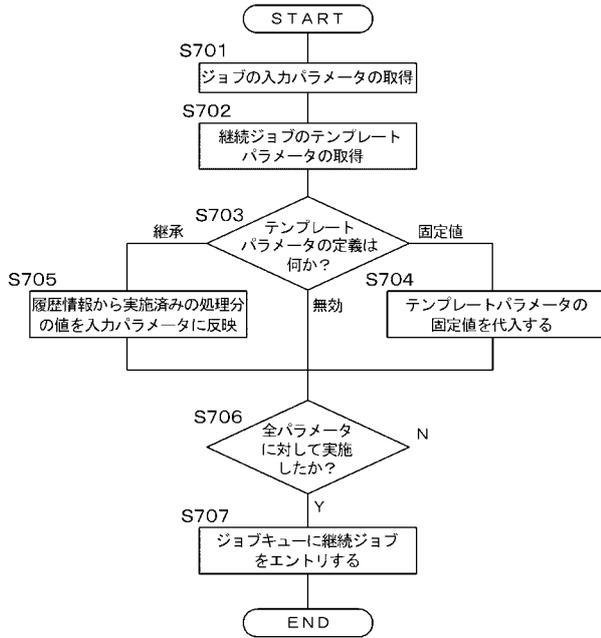
【 図 1 5 】



【 図 1 6 】



【 図 1 7 】



【 図 1 8 】

テンプレートパラメータ特性定義表 L 8

パラメータ分類	パラメータ名称	パラメータ特性	値
レチクル情報	レチクル名	無効	
	レチクルID	無効	
	デザイン	無効	
レイアウト情報	サイズX.Y	無効	
	オリジンX.Y	固定	5.5
	アングル	固定	0
	ショット情報	無効	
マーク情報	マーク形状	固定	なし
	マーク位置	固定	なし
露光情報	露光量	無効	
	露光前傾き計測	固定	OFF
	露光前ゆがみ計測	固定	OFF
	露光前位置計測	固定	OFF
	露光モード	固定	Auto fast
	移動スピード	固定	
	位置誤差補正 ゆがみ誤差補正	固定	OFF
ウェハ情報	ロット名	無効	
	ロットID	無効	
	キャリアID	無効	
	ウェハ枚数	無効	
	ウェハサイズX.Y.Z	無効	
	ウェハ種	無効	
	ウェハスタートIndex	継承	

【 図 1 9 】

