University of Würzburg
Institute of Computer Science
Research Report Series

Simulation und Leistungsbewertung von Cluster Tools in der Halbleiterfertigung

Mathias A. Dümmler

Report No. 231 5. Juli 1999

University of Würzburg
Department of Computer Science
Am Hubland, D-97074 Würzburg, Germany
duemmler@informatik.uni-wuerzburg.de

Presented at: ASIM '99, 13. Symposium Simulationstechnik, Weimar, Germany, September 21-24

Simulation und Leistungsbewertung von Cluster Tools in der Halbleiterfertigung

Mathias A. Dümmler
University of Würzburg
Department of Computer Science
Am Hubland, D-97074 Würzburg, Germany
duemmler@informatik.uni-wuerzburg.de

Zusammenfassung

Cluster Tools kommen in den letzten Jahren verstärkt bei der Fertigung von Halbleiterchips zum Einsatz. In diesem Beitrag wird auf die wesentlichen Eigenschaften von Cluster Tools und auf die Probleme bei deren Einsatz eingegangen. Ein Programm zur Modellierung und Simulation von Cluster Tools wird vorgestellt. Anhand einer Studie wird die Verwendung des Simulators in Kombination mit Genetischen Algorithmen zur Verbesserung der Durchlaufzeiten in Cluster Tools demonstriert.

1 Einleitung

Seit dem Beginn der neunziger Jahre hält ein neuer Typ von Maschinen, die sogenannten Cluster Tools, Einzug in die Halbleiterfertigung [1]. In einem Cluster Tool werden mehrere Prozeßschritte in einer Maschine integriert, die bisher in separaten Geräten ausgeführt wurden. In Abbildung 1 ist ein Cluster Tool schematisch dargestellt.

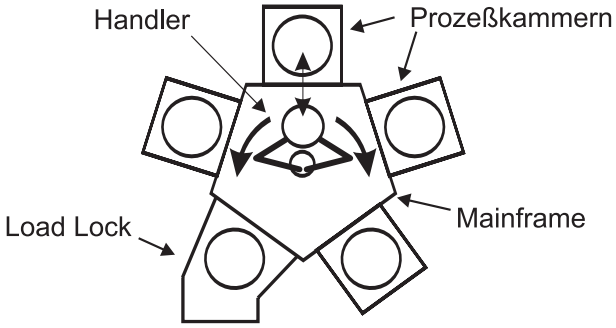

Abbildung 1: Cluster Tool

Die Bearbeitungsschritte erfolgen in Prozeßkammern, die an eine zentrale Baueinheit, den sogenannten Mainframe, angekoppelt sind. Die standardisierten Schnittstellen zwischen Mainframe und Prozeßkammern ermöglichen dabei die Verwendung von Komponenten verschiedener Hersteller.

Die Wafer, also die Siliziumscheiben, auf denen die Chipstrukturen hergestellt werden, gelangen in Losen von meist 25 Wafern über eine Schleuse (Load Lock) in das

Cluster Tool, in dem Vakuum herrscht. Für jeden Wafertyp existiert eine Prozeßvorschrift, die angibt, welche Kammern eines Cluster Tools ein Wafer besuchen muß und welcher Prozeß in der jeweiligen Kammer ablaufen soll. Die Wafer werden im Cluster Tool durch einen oder mehrere Roboter, sogenannte Handler, transportiert.

Es existieren im wesentlichen zwei Verwendungsarten für Cluster Tools. Im rein sequentiellen Betrieb läuft in jeder Kammer eines Cluster Tools ein anderer Prozeß ab. Wafer mit gleicher Prozeßvorschrift folgen daher immer dem gleichen Weg durch ein Cluster Tool. Im parallelen Betrieb dagegen stehen für einen oder mehrere Prozeßschritte alternative Kammern zur Verfügung. Dadurch lassen sich Engpässe, die durch hohe Prozeßzeiten in einer Kammer entstehen können, umgehen.

Die Integration mehrerer Prozeßschritte in ein Cluster Tool führt zu einer Reihe von Vorteilen gegenüber herkömmlichen Maschinentypen [1]. Beispielsweise müssen die Wafer zwischen den Prozeßschritten die Maschine nicht verlassen und daher nicht einer eventuellen Verunreinigung durch Partikel in der Umgebungsluft ausgesetzt werden. Solche Verunreinigungen führen aufgrund der sehr geringen Strukturgrößen auf den Halbleiterchips meist dazu, daß einzelne Chips oder ein kompletter Wafer unbrauchbar werden. Weiterhin ist kein Operator nötig, um die Wafer von einem Prozeßschritt zum nächsten zu transportieren. Durch den Wegfall von Wartezeiten auf einen Operator und durch die kürzeren Transportzeiten verringert sich die Durchlaufzeit der Wafer.

Die Leistungsbewertung von Cluster Tools ist aufgrund der Komplexität des Zusammenspiels zwischen Handlern, Wafern und Prozeßkammern nicht trivial. Analytische Verfahren, wie etwa die in [2] vorgestellte Methode zur Berechnung der Bearbeitungszeit eines Loses, sind nur für sehr einfache Modelle von Cluster Tools geeignet und unterliegen starken Beschränkungen, etwa der Annahme identischer Bearbeitungszeiten in den einzelnen Prozeßkammern.

Eine Reihe von Autoren haben sich daher bereits mit der Simulation von Cluster Tools befaßt [3][4]. Im kommerziellen Bereich sind Simulatoren für Cluster Tools sowohl von einigen Hardware-Herstellern als auch von Anbietern von Simulationssoftware erhältlich.

Bei der Entwicklung von Simulationsmodellen von Cluster Tools stößt man auf eine Reihe von Schwierigkeiten. So ist etwa das genaue Verfahren zur Steuerung der Handler meist nur dem Hersteller eines Cluster Tools bekannt. Diese Steuerung hat jedoch starken Einfluß auf die Leistung des Gesamtsystems, muß also möglichst genau modelliert werden. Ferner können in Cluster Tools auch Deadlocks auftreten, etwa wenn Prozeßkammern von einem Wafer wiederholt besucht werden [5]. Solche Deadlocks gilt es in der Simulation zumindest zu erkennen, besser aber noch rechtzeitig zu vermeiden.

2 Beschreibung des Simulators

Um detaillierte Untersuchungen im Zusammenhang mit Cluster Tools durchführen zu können, wurde ein Simulationsprogramm zur Modellierung von Cluster Tools in C++ entwickelt. Der Erwerb eines kommerziellen Programms zur Simulation von Cluster Tools kam für diese Studien nicht in Frage, da sich diese Programme als nicht flexibel genug und nicht erweiterbar erwiesen. Die Verwendung eines Simulationspaketes, wie

etwa Arena von Systems Modeling Corporation, wurde in der Planungsphase auch ausgeschlossen. Den Hauptteil des Simulationsprogramms machen nämlich die Algorithmen zur Steuerung der Handler aus, die sich in solchen Simulationspaketen üblicherweise nur sehr aufwendig realisieren lassen.

Der hier vorgestellte Simulator ist so ausgelegt, daß sich Cluster Tool-Typen verschiedener Hersteller modellieren lassen. Das Simulationsmodell besteht aus einer beliebigen Anzahl von Cluster Tools, von denen sich jedes aus einer beliebigen Anzahl von Prozeßkammern, Load Locks und Handlern zusammensetzt. Es können mehrere Prozeßsequenzen spezifiziert werden, die angeben, wie lange die Wafer in einer Kammer bearbeitet werden müssen. Für jeden Prozeßschritt einer Sequenz werden eine oder mehrere Prozeßkammern angegeben, in denen der Schritt abgearbeitet werden kann. Für die einzelnen Komponenten eines Cluster Tools lassen sich auch Ausfälle modellieren.

Das Simulationsmodell wird mittels eines Textfiles spezifiziert, in das die einzelnen Komponenten des Modells und ihre Parameter eingetragen werden. Dieses File wird dann vom Simulationsprogramm ausgewertet und das Simulationsmodell erstellt. Zu den in jedem Simulationslauf ermittelten Leistungsgrößen gehören die Durchlaufzeit von Wafern und die Auslastung der einzelnen Komponenten des Cluster Tools. Diese Leistungsgrößen ermöglichen es, eventuelle Schwachpunkte oder Engpässe in einem Cluster Tool zu identifizieren. Sollen Maßnahmen ergriffen werden, um diese Schwachstellen zu beseitigen, etwa durch die Anschaffung einer weiteren Prozeßkammer, kann mit einem neuen Simulationsmodell die Leistungssteigerung, die durch diese Maßnahme erzielt wird, abgeschätzt werden.

Der modulare Aufbau des Simulators ermöglicht es, einzelne Komponenten des Programms auszutauschen. Beispielsweise kann das Modul zu Steuerung der Handler derzeit aus vier Modulen mit jeweils unterschiedlicher Steuerungsstrategie ausgewählt werden.

Um die Bedienung des Simulators zu vereinfachen, wurde mit dem Tabellenkalkulationsprogramm Excel der Firma Microsoft unter Visual Basic ein Front End mit vordefinierten Arbeitsblättern entwickelt, in welche die Modellparameter eingetragen werden. Dieses Programm erlaubt auch die automatische Erstellung von Grafiken, um die Simulationsergebnisse zu veranschaulichen. Es besteht weiterhin die Möglichkeit, die Abläufe im Cluster Tool am Bildschirm darzustellen.

3 Simulationsmodell eines Cluster Tools

Im Rahmen einer Studie, in der das vorgestellte Simulationsprogramm verwendet wurde, wurde das in Abbildung 2 dargestellte Cluster Tool näher untersucht. Es besteht aus zwei Hauptmodulen, an die elf Prozeßkammern und zwei Load Locks angekoppelt sind und in denen sich jeweils ein Roboter zum Transport der Wafer befindet. Die beiden Load Locks des Cluster Tools können im Parallelbetrieb benutzt werden, d.h. zwei Lose mit Wafern unterschiedlichen Typs können gleichzeitig bearbeitet werden.

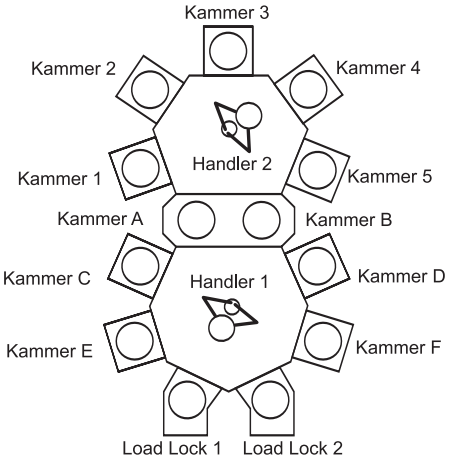

Abbildung 2: Untersuchtes Cluster Tool

Die Prozeßsequenzen, die in der Studie verwendet wurden, sind in Tabelle 1 wiedergegeben. Für jeden Prozeßschritt ist die Bearbeitungszeit eines Wafers in der entsprechenden Kammer angeben. Ein leeres Feld bedeutet, daß die Kammer für diese Sequenz nicht benutzt wird. Der erste Bearbeitungsschritt von Sequenz 1, 2 und 3 kann in Kammer E oder Kammer F ausgeführt werden. Die Bearbeitungszeit 0 in Kammer A läßt erkennen, daß diese Kammer nur als Transferkammer benutzt wird, um Wafer in den oberen Teil des Cluster Tools zu befördern.

Tabelle 1: Bearbeitungszeiten in den Kammern

Kammer	E|F	C	D	A	1	2	4	B
Seq. 1	80	60	40	0	70	40		30
Seq. 2	60			0	70			
Seq. 3	80	60	40	0		90		30
Seq. 4				0			80	30

Durch die Möglichkeit der Parallelverarbeitung zweier verschiedener Prozeßsequenzen ergeben sich 16 Kombinationen von Sequenzen, die zu jeweils unterschiedlichem Durchsatz führen. Ein Ziel bei der Einschleusung von Losen muß es daher sein, möglichst günstige Kombinationen von Sequenzen einzustarten. Ungünstig für den Durchsatz ist etwa die Parallelbearbeitung von Losen der Sequenz 1 und der Sequenz 3, da sich diese Sequenzen sechs Kammern teilen müssen, sich also gegenseitig stark behindern. Günstig für den Durchsatz ist die Komination von Sequenz 2 mit Sequenz 3, da diese beiden Sequenzen nur die Kammern A, E und F gemeinsam verwenden.

4 Reihenfolgebildung von Losen mit Genetischen Algorithmen

Während für eine kleine Anzahl von Sequenzen die Ermittlung von Bearbeitungsreihenfolgen von Losen, die zu kurzen Durchlaufzeiten führen, noch von Hand ausgeführt

werden kann, ist diese Aufgabe für eine große Anzahl von Losen nur noch mit hohem Aufwand durchführbar. In diesem Abschnitt wird deshalb ein Verfahren vorgestellt, welches diese Aufgabe löst. Eine Anzahl von Losen mit unterschiedlichen Prozeßvorschriften soll so auf eine Menge von Cluster Tools aufgeteilt werden, daß die Bearbeitungszeit der Lose minimiert wird. Dazu soll auch die Bearbeitungsreihenfolge der Lose an den einzelnen Cluster Tools optimiert werden.

Zur Lösung dieses Problems wurde ein heuristischer Algorithmus zur Berechnung der Lossequenzen implementiert, der auf Genetischen Algorithmen basiert [6][7]. Zur Implementierung wurde die Programmbibliothek GAlib [8] verwendet. Da die GAlib in C++ geschrieben ist, konnte sie leicht in das bestehende Simulationsprogramm integriert werden.

Die Grundidee hinter den Genetischen Algorithmen ist es, einen evolutionären Prozeß, wie er in der Natur abläuft, zu imitieren: Nur die stärksten Individuen einer Generation überleben und pflanzen sich fort, um ihr Erbgut an die nächste Generation weiterzugeben. In dem hier behandelten Fall sind die Individuen die Lossequenzen. Sequenzen, die zu einer kurzen Bearbeitungszeit führen, sind dabei stärker oder haben eine höhere *Fitness* als Sequenzen mit langer Bearbeitungszeit.

Der Genetische Algorithmus erwartet als Eingabe eine Menge von Losen, die zur Bearbeitung auf einer Gruppe von Cluster Tools zur Verfügung stehen. Das Simulationsmodell dieser Cluster Tools wird als Textfile vom Simulator eingelesen. Um die Fitness einer Sequenz zu ermitteln, wird das Simulationsprogramm mit dieser Sequenz als Eingabe gestartet und die Zeit, die zur Bearbeitung dieser Sequenz auf den Cluster Tools benötigt wird, ausgegeben.

In Tabelle 2 sind die Resultate für einen Testlauf des Genetischen Algorithmus angegeben. In diesem Beispiel sollten drei Lose von jeder Prozeßsequenz aus Tabelle 1, also insgesamt zwölf Lose, so zur Bearbeitung auf zwei Cluster Tools aufgeteilt werden, daß die Gesamtbearbeitungszeit dieser zwölf Lose minimiert wird.

Der Genetische Algorithmus wurde fünfmal angewendet, um dieses Problem zu lösen. In Tabelle 2 ist die Gesamtbearbeitungszeit für die beste Sequenz angegeben, die der Algorithmus erzielte. Um die Qualtität der gefundenen Lösungen abschätzen zu können, wurde die mittlere Bearbeitungszeit für 20 zufällig erzeugte Sequenzen ermittelt. Die Verbesserung gegenüber diesem Mittelwert ist in der dritten Spalte dargestellt. Ferner ist die Anzahl von Sequenzen angegeben, die der Genetische Algorithmus insgesamt in einem Lauf erzeugt hat und die Laufzeit des Algorithmus auf einem Pentium II 266.

Allgemein läßt sich sagen, daß der Genetische Algorithmus in kurzer Zeit Sequenzen mit optimaler oder beinahe optimaler Gesamtbearbeitungszeit erzeugt. Das Programm erfordert nur sehr wenige Eingaben vom Benutzer und läßt sich leicht in ein bestehendes Planungssystem für eine Halbleiterfertigungsanlage integrieren.

5 Zusammenfassung

In diesem Beitrag wurde ein Programm zur Modellierung und Simulation von Cluster Tools vorgestellt. Mit dem Simulator können Cluster Tools verschiedener Hersteller und unterschiedlicher Bauart modelliert werden.

Tabelle 2: Ergebnisse

Lauf	Bearbeitungszeit (s)	Verbesserung (%)	getestete Sequenzen	Laufzeit (s)
1	10373	14.6	115	164
2	10216	15.9	116	164
3	10142	16.5	113	159
4	10386	14.5	112	162
5	10631	12.5	116	163

Ferner wurde ein Verfahren vorgestellt, das den Simulator in Verbindung mit einem Genetischen Algorithmus verwendet, um die Reihenfolgebildung von Losen, die auf einer Menge von Cluster Tools bearbeitet werden sollen, zu optimieren. Es ist geplant, das vorgestellte Verfahren im Fertigungsbetrieb unter reellen Bedingungen zu testen, um das Verbesserungspotential, das durch die Optimierung der Bearbeitungsreihenfolge erzielt werden kann, zu ermitteln.

Es sind einige Erweiterungen des Genetischen Algorithmus denkbar. Bei der Bewertung einer Lossequenz sollte außer der Bearbeitungszeit einer Sequenz auch noch die Einhaltung von Fälligkeitsterminen der Lose und eventuelle Prioritäten berücksichtigt werden. Weiterhin sollte dem Benutzer die Möglichkeit gegeben werden, dem Genetischen Algorithmus selbst Sequenzen vorzuschlagen, die dann weiter optimiert werden.

Literatur

[1] P. Singer, "The driving forces in cluster tool development," *Semiconductor International*, S. 113–118, Juli 1995.

[2] T. L. Perkinson, P. K. McLarty, R. S. Gyurcsik, "Single-wafer cluster tool performance: An analysis of throughput," *IEEE Transactions on Semiconductor Manufacturing*, Vol. 7, S. 369–373, Aug. 1994.

[3] R. W. Atherton, F. T. Turner, L. F. Atherton, M. A. Pool, "Performance analysis of multi-process semiconductor manufacturing equipment," in *Proceedings of IEEE/SEMI Advanced Semiconductor Manufacturing Conference*, S. 131–136, 1990.

[4] J. Mauer, R. Schelasin, "Using simulation to analyze integrated tool performance in semiconductor manufacturing," *Microelectronic Engineering*, Vol. 25, Nr. 2/4, S. 139–146, 1994.

[5] L. W. Schruben, "Deadlock detection and avoidance in cluster tools," in *Proceedings of the 1999 International Conference on Semiconductor Manufacturing Operational Modeling and Simulation*, S. 31–35, 1999.

[6] "Encore (The EvolutioNary COmputation REpository network)." ftp://ftp.krl.caltech.edu/pub/EC/Welcome.html, 1997.

[7] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley, 1989.

[8] M. Wall, "GAlib. A C++ library of genetic algorithm components." http://lancet.mit.edu/ga/, 1995.