

半導体生産工程のシミュレーション

米田 清, 藤原 睦

1. はじめに

半導体(集積回路)生産システムの規模と複雑さは、生産工程の中でも最大級である。その挙動については直観が効かないので、モデルで予測することになる。

目的が狭く限定できるなら、それを達成する最低限の機能を備えた、抽象的なモデルの方が扱いやすい。たとえば指標として平均月産とか平均在庫量が出れば良いなら、待ち行列網モデルを数値的に解けばことたりる。しかしそれでは、個々のロットや加工装置の動きを追跡したいという要求に答えられない。たとえば「今日、この品種のロットを5個投入したら、いつ製品を倉入れできるか。答えは納期回答に使う。」「今度あの機械が1時間以上空くのはいつか、今週中なら、空きに保守を入れる。」「そこの試作ロットは、特急にすれば、いつ上がるのか、犠牲になって遅れるのは何か。」等々。このような要求に対応するには、定常状態の平均指標しか出ないのでは困る。また、たとえば定常状態の平均指標にしか興味がない場合でも、数式モデルは記述の詳細さを自由に選択できないため、説明に困難がつきまとう。つまり、「この仮定を外してほしい」といわれても困ることがある。

こうした場面で説得力をもつのはシミュレーションしかない。このような事情で、'92年の Winter Simulation Conference でも wafer fabrication という独立したセッションが設けられていた。

コンピュータの世界では、スケーラビリティ(scalability)がよく議論される。スケーラブルとは、同じ方法が小さい問題から大きい問題まで一貫して通用することである。小さな例題しか解けない手法とか、逆にある程度の規模がないと成り立たない近似は、スケーラブルでない。半導体生産工程のシミュレーションでは、1つのシミュレーション方式で複数の工場の、個別なモデルを扱う。常識的な大きさの試作工場から、通常の離散シ

ミュレーションでは扱わない大きさの量産工場までが対象に含まれる。そのため、いかにスケーラビリティを確保するかが問題になる。

スケーラビリティは、少なくとも以下の側面について問題になる。モデル化に関して：誤りのないモデルをどう労力をかけずに作りデータを集めるか。実装に関して：ソフトウェアの信頼性をどう保証するか。実行に関して：シミュレーションをどう実用的な時間内に実行し結果を解釈するか。いずれも、小さい工場を少しだけ扱うなら、従来からある方法でも何とかなる。しかし工場が大きく数も多くなると、質的といえるほどの違いが出る。

以下、まず2節で半導体生産工程について説明する。ついで3節でそのシミュレーション技術について現状を述べる。4節でスケーラビリティの問題に対する1つの解答を示し、5節に要点をまとめる。

2. 半導体生産工程

半導体生産工程には前工程と後工程がある。前工程ではウエハという円板に、チップと呼ばれる集積回路をたくさん、いわば印刷する。後工程ではそのウエハをチップに切り分けてから、電気製品に組み込まれている形にするまでの加工を行なう。

2.1 前工程

前工程の生産方式は各国各社に独特の流儀がある。典型的にはジョブショップ型に構成し、複数の品種を並行して生産する。生産管理の単位はロットである。ロットは同一品種のウエハ1枚から30枚くらいまでで構成する。ウエハの直径は20cmくらいで、ロットは1人で運べる大きさである。生産の実行組織(たとえば製造課)は原材料の投入から製品の完成まで一貫して担当する。前工程の場合、シミュレーションの対象はこの組織である。以下、便宜上これを工場と呼ぶ。

工場内には約200台の製造装置があり、約40の装置群に分かれている。1つの装置群にある製造装置はほぼ同種で、装置群がジョブショップに相当する。作業はすべて、環境の管理されたクリーンルーム内で行なわれる。装置

よねだ きよし, ふじはら むつみ

㈱東芝 研究開発センター

〒210 川崎市幸区柳町70

は高価で、1台で数億円するものも少なくない。技術革新が激しいので、開発直後で動作の安定していない装置もしばしば使われる。

スループットは月産700から1,000ロットで、中間在庫は1,000から1,500ロットある。DRAM (dynamic random access memory) 工場では扱う品種数は10くらいある。ASIC (application specific integrated circuit) 工場の場合、細かく分類すれば品種数は中間在庫量よりも多いくらいあり、ほとんどの品種は1ロットしか工場内にない。いずれの場合も品種ごとに異なる加工経路で装置群をまわる。各経路は100から300程度の工程からなり、1つのロットは1つの装置群をくりかえし訪問する。移動は磁気浮上搬送車などによる。ロットは加工経路の途中で複数のロットに枝別れすることがある。これは途中までは同一の品種として母体を作るため、ASICに多い。

傾向としては、ウエハは大きく、工程数は多く、工期は長くなりつつある。また、管理の対象をロットごとからウエハごとに引き下げようという枚葉管理も、過去に採用した会社がある。当時は情報の通信と処理に関する技術が未成熟で失敗した。現在のCIM (computer integrated manufacturing) 環境なら、成功する可能性がある。もしも将来このきめ細かい管理法をとるなら、シミュレーションの対象もロットでなくウエハになるため、モデルの規模も一挙に数十倍になる。これに対応しようとするならシミュレーション方式のスケラビリティが問題になる。

2.2 後工程

後工程では、まずウエハをチップに切り分け、ついで組立て、配線、パッケージ化、試験などが行なわれる。生産管理の単位は、やはりロットで、後工程ロットは前工程のロットを細分化したものになっている。後工程は加工拠点が地理的に分散していたり、複数の別会社にまたがることもある。その場合、輸送が介在する。シミュレーションは輸送を含む後工程全体を工場として扱う。加工装置の総数は1,000台以上になる。ロットの経路は前工程と異なり、同じジョブをくりかえし訪問することはない。品種は加工の途中で枝分かれし、最終的には投入時の数倍に増える。また、1つのロットが複数の納期に対応するように分割されることがある。

後工程は加工技術自体としては前工程ほど複雑ではない。とはいえ、経路が複雑で、ロットの分割が頻繁に起こり、品種が多く、ジグの使い回しがあるなどの理由で、

モデルは前工程と同程度以上の複雑さになる。

3. 技術状況

半導体生産工程のシミュレーションは短期と長期とで技術的な性格が異なる。

短期シミュレーションは工場運営のために行なわれ、生産指示や納期回答が具体的な目的である(たとえば[1])。通常は数カ月程度を扱う。いろいろ条件を変えてみてシミュレーションをする必要上、実時間的な高速性が要求される。ワークステーションで数分から15分くらいまでで実行したい。確率的なシミュレーションは複数のランを要するので、行なっている余裕がない。つまり乱数はふつう、使わない。短期シミュレーションに必要な機能は、工場の現状を初期状態として取り込めることである。当然、各ロットがどこにあってどうしているかを実時間で把握する、管理インフラの整備が前提条件になる。

長期シミュレーションは研究的な用途が主で、設備計画とか、ロット投入やディスパッチングのルール比較などに使われる(たとえば[2])。定常的な状態での性能指標を問題にするので、確率シミュレーションを行なう。この場合、工場の現状ではなく、工場にロットがない状態を初期状態にとることもできる。過渡状態にある、始めの方のデータを捨てればよいからである。そこで、短期シミュレーションほどのデータ収集インフラは、なくてもすむ。反面、どれだけ初期データを捨てればいいのか、どう実験を計画した結果を解釈するかなど、統計的な問題をかかえることになる。

SEMATECHはアメリカが安全保障という観点から日本の半導体産業に対抗する目的で作った半官半民の組織で、規模は700人である。'93年3月、Cornell大学のSchruben教授を団長に、日本の半導体業界での工程モデル化技術の調査に数名を派遣している。そのoperational modeling部門が、半導体生産工程シミュレーションの研究拠点である。10人強の研究者がおり、テーマはfactory performance modeling systemとsemiconductor workbench for integrated modelingである。ワークベンチでは、Arena[3]というソフトウェアが1つのデータベースから、モデルのテンプレートに従っていろいろな詳細度のシミュレーション用モデルを生成し、SIMAN[4]というシミュレーション言語に落とす。この言語は'80年代から業界で使われており、市販されている。結果はCinemaというモジュールを通して、アニメーションになる。市販の半導体工程

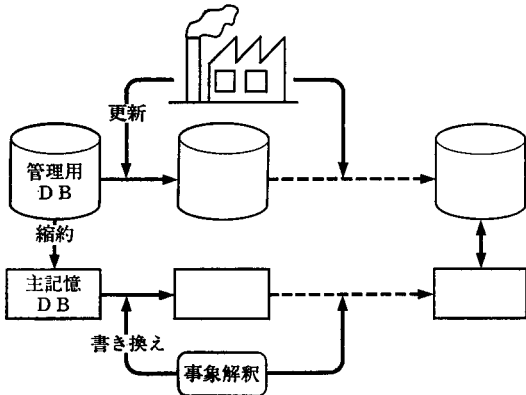


図1 データベースの更新過程とシミュレーション

シミュレータには他にも Achilles[5]や MANSIM がある。工場のデータベースとシミュレーションの連係については、たとえば[6]の事例がある。

日本では各社が個別に技術を開発しており[7, 8, 9], ソフトウェアをパッケージ化して流通する土壌はない。工程シミュレータを全くもたない企業は、業界にないであろう。技術的には、今のところ日本とアメリカで大差はないように思われる。短期シミュレーションでは日本がやや進んでいる。インフラへの投資が先行しており、技術者が現場に近いからである。長期シミュレーションについては、アメリカの方が水準が高い。取り組みが国家的で、産学が共同しているからである。

4. 事例

以下、半導体生産工程シミュレータ DEUS [10, 11] で採用した、スケーラビリティを確保する方法を紹介する。対象は前後工程のどちらでもよく、短期シミュレーションである。

4.1 モデル化の方法

工場管理用のデータベースは、現実的に達成しうる最も詳細な、工場の状態記述になっている。当然、データベースが整備されていることが前提である。整備されているなら、工場を運営するうえで必要なデータはみなデータベースに入っている。たとえば、各ロットの進捗状況、各装置の動作状況、各製品の加工経路、各製品の加工法、操業予定、投入計画など。工場の現在の状態を知るには、データベースを見ればよい。

データベースをそのまま工場の状態記述モデルに採用する。それなら、シミュレーション用の状態記述モデルを別途、作成する必要がない。しかしデータベースには

シミュレーションでは必要のないデータも含まれている。たとえば、品質管理のために採取しているデータがすべて必要なシミュレーションはない。それらを落として、図1のようにシミュレーション用コンピュータにダウンロードする。管理用データベースの内容よりも量が減っているため、主記憶内に格納できる。

このように管理用データベースの縮約版を主記憶内データベースとして構成し、シミュレーションのための状態記述とする。

これによって、モデル作成のスケーラビリティを確保する。データベースがあれば、これ以上は詳細にならないという一種の **universal property** をもったモデルができてしまっているのだから、対象がいくら大きくても不安はない。データ収集についても、データベースを利用する以上、すでに集まっている。結果の解釈も、管理用データが理解できる人には、原則的にシミュレータの状態記述が理解できることになるから、問題は少ない。

実際に作業すればもちろん、データの不備、散在、矛盾などに出合う。それらの問題が顕在化することは、シミュレーションするという立場からはやっかいでも、データベースを整備する推進力になり、CIM の基盤を固めるものになる。統計手法を使おうとすると、統計資料を整備せざるをえないようなものである。

4.2 シミュレータの構成

ソフトウェアの誤りを防ぐ最良の方法は、手続き、すなわちプログラムを書かないことである。プログラムがなければバグはない。プログラムは、小さければ正しい可能性が高く、大きいならば誤っている可能性が高い。スケーラビリティを確保するには、シミュレーションの対象がいくら大きくなってもプログラムが小さくてすむようにしたい。そのためのアーキテクチャを説明する。

工場で起こるさまざまな事象のうち、重要なものはみな、図1の上半分に示すように、データベースの書き換えを引き起こす。たとえばあるロットの加工がある装置で開始されたとか、それが終了したとかが引き金になってデータベースが更新される。それと同じように主記憶内データベースを図1の下半分のように書き換えてゆく機構を作れば、シミュレータが完成する。

では、どのような書き換えをすればいいか。その情報は、ダウンロードした状態記述にほとんど全部含まれている。状態記述は、前節に述べたように、工場の現状とともに計画も含むからである。工場の将来の状態は、現在の状態と計画を、実行規則にしたがって進行させたもの

として決まる。データベースに明示的に書かれていないのは、その実行規則だけなのである。実行規則の大きな部分は、事象リストで表現される。そして事象リストも状態記述に含めてしまうことが可能である。

結局シミュレーションは、状態記述を、実行規則に照らして、解釈して書き換える操作の連続である。

状態は何らかの規則にもとづいて書かれているのだから、言語的な記述であるとみなせる。それを解釈して書き換えることは、言語の解釈的な実行、すなわちインタプリテーションにほかならない。

インタプリタからみれば、状態記述はデータであって、プログラムではない。この点が重要である。つまり、事象リストを含めて、なにもかも状態記述に放り込んでおけば、ほとんど何でもデータであってプログラムではない。それだけ手続きは抽象的ですし、プログラムの量が減る。よってインタプリタとしてのシミュレータはきわめて小さく作れる。どれだけ小さくなるかについては後述する。

4.3 主記憶内データベース

工場の管理用データベースを主記憶内データベースにダウンロードし、事象リストを加えて状態記述として解釈実行すればシミュレーションができることを述べた。残る問題は主記憶内データベースの構成法である。スケラビリティを確保するには、データが増えても、検索や登録にかかる時間や必要な記憶容量があまり増えはならない。たとえば指数的な増加では、問題外である。

現在のハードウェアでは、アルゴリズムを実行するうえでのネックはPU(processing unit)の遅さではなく、メモリ・アクセスの遅さである。PUの高速化はメモリの高高速化よりもベースが速い。また、PUの低価格化はメモリの低価格化よりも、はるかにベースが遅い。そのためPUは高価で速く、メモリは安価で遅いという状況がある。この傾向は今後も続くだろう。そこで主記憶内データベースは、メモリを多く使うことによって、アクセスを速くする。

具体的には、図2のようにデータに冗長性をもたせ、1つのレコードを複数の異なる形式で格納する。これは1つの取り引きを記録するのに複数の帳簿に記入するようなものである。検索する時は、もっている手がかり、たとえばロット名と、目的とする情報、たとえば次に行なわれるべき工程を結ぶように帳簿をたぐる。帳簿が少ないと検索に時間がかかる。帳簿が多過ぎると維持にメモリを食ううえ、記録の更新に時間がかかる。

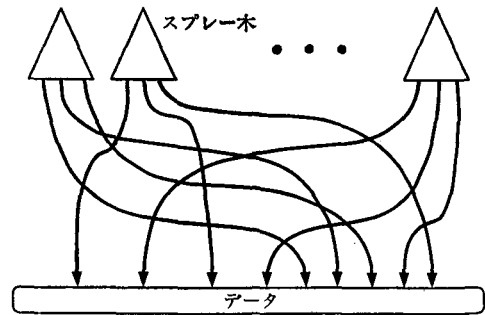


図2 主記憶内データベース

1つ1つの帳簿にあたるものはスプレー木 (splay tree) [12]というデータ構造で実現する。どの帳簿も同じ構造なので、何冊あろうとプログラムの量は変わらない。もちろんデータ自体は主記憶内に1つしかなく、帳簿の中身はポインタばかりである。

スプレー木は2進木の種類で、適応的に構造を変更する。データをアクセスするごとに、よくアクセスするデータを引きやすく、かつ、極端に引きにくいデータが平均として少なくなるように木を編成し直す。1回ごとのアクセスは再編成の手間だけ遅い。しかしデータに対するアクセス頻度の偏りを利用して、平均アクセス時間は短くなる。その定量的な評価は理論的にも実証的にもなされている。

その結果を適用すると、主記憶内データベースのアクセスはレコード数を N として $O(\log N)$ である。 N はほぼ、工場内のロットの数と装置の数の和に相当する。そしてシミュレーション期間内に何回データベースをアクセスするかは、その期間内に起こる事象の数に比例する。要するにシミュレーションにかかる時間は、工場の規模が大きくなってもあまり増えず、スケラビリティが確保されている。必要な記憶容量は N に比例するので、この増え方も対象の規模に見合う程度である。

4.4 実験結果

以上のように構成した半導体生産工程シミュレータをCで実装した。主記憶内データベースを扱うプログラムは千行に満たない。最も基礎にあるスプレー木を扱うプログラムや、事象リストの扱いもそこに含まれる。個々の事象のインタプリタが数百行で、ここにはたとえば、ロットの加工が終了するとどのように状態記述を変更するかといった解釈手続きが書いてある。以上、シミュレータ全体で千数百行にすぎない。したがって信頼性は高い。

プログラムが小さい代り、データは大きい。前工程のシミュレーションで、管理用データベースから取り込んで展開した初期状態が数メガバイトある。シミュレーションを行なうとデータが蓄積され、1カ月間のシミュレーションでは数十メガバイトになる。実行にはワークステーションで5分から10分かかる。

以上の具体的な数値は、スケーラビリティの観点からは性能の一端を示すに過ぎないことを強調しておく。これよりも対象が大きくなった時、モデル作成にかかわる技術者、計算にかかる時間、記憶容量など必要な資源がどれだけの勢いで増えるかが、より重要な問題なのである。

5. おわりに

半導体生産工程シミュレーションの技術課題の1つに、スケーラビリティがある。モデル化のスケーラビリティは、工場管理用のデータベースを主記憶内データベースに移して状態記述に用いることで実現できる。これを状態の言語的記述とみなして解釈実行することがシミュレーションである。この方法によると対象が大きくてもプログラムがきわめて小さく作れ、ソフトウェアの信頼性が確保される。実行時間は対数オーダー、必要記憶容量は線形で、いずれもスケーラビリティが確保されている。

参考文献

- [1] 冬木正彦, 井上一郎, 大規模生産システムにおける計画業務支援のためのシミュレーションシステム LSIP-モデルとシミュレータ, シミュレーション, Vol.11, No.4, pp.73-83, 1992.
- [2] D. J. Miller. Simulation of a semiconductor manufacturing line. *Communications of the ACM*, Vol.33, No.10, pp.98-108, 1990.
- [3] C. D. Pegden and D. A. Davis, Arena: A SIMAN/Cinema-based hierarchical modeling system. In *Proc. Winter Simulation Conference*, pp.390-399, 1992.
- [4] S. A. Conrad, D. T. Sturrock, and J. P. Poorte. Introduction to SIMAN/Cinema. In *Proc. Winter Simulation Conference*, pp.377-379, 1992.
- [5] R. W. Atherton. Detailed simulation for semiconductor manufacturing. In *Proc. Winter Simulation Conference*, pp.659-663, 1990.
- [6] S. S. Baum and P. G. Glassey. Supporting semiconductor manufacturing simulations tools using a structured data model. In *Proc. Winter Simulation Conference*, pp.879-884, 1992.
- [7] 林昌孝, 箱田義典. FAライン評価用シミュレータの開発と適用事例. IEレビュー, Vol.30, No.2, pp.34-41, 1989.
- [8] 野瀬純郎他. LSI生産ラインの生産能力評価技術. *NTT R&D*, Vol.42, No.6, pp.813-822, 1993.
- [9] S. Umeda. A manufacturing-oriented simulation package to support systems planning and its operation. In *Proc. Winter Simulation Conference*, pp.890-898, 1992.
- [10] M. Fujihara and K. Yoneda. Simulation through explicit state description and its application to semiconductor fab operation. In *Proc. 1992 Winter Simulation Conference*, pp.899-907, Washington, D.C., U.S.A., 1992.
- [11] S. Kamimura, M. Fujihara, and K. Yoneda. Realtime simulation for semiconductor fab operation. In *Proc. Pacific Conference on Manufacturing*, pp.11/3-6, Sakai, Japan, 1992.
- [12] D. D. Sleator and R. E. Tarjan. Self-adjusting binary search trees. *Journal of the ACM*, Vol.32, pp.652-686, 1985.