Chapter 1 — Introduction

Semiconductor wafer fabrication [Old77a, Gis86a] is perhaps the most complex manufacturing process found today. This complexity is, in part, the result of constant devise miniaturization, process intricacy, product diversity, uncertainty, and changing technologies. Aside from the obvious technical challenges that circuit designers face, semiconductor wafer fabrication offers a wide range of complex issues related to production planning and scheduling. The introduction of computer-integrated manufacturing (CIM) systems in the fabrication process has added a new dimension of problems.

Penfield *et al.*, in an unpublished document [Pen84a], include scheduling of lots in a wish-list for computer-aided fabrication of semiconductor wafers. They say —

> *Scheduling of Lots.* The system must provide movement recommendations for manually operated fabrication lines and it must control movements for lines with automated material handling systems. It must coordinate movements over a large, complex facility with the objective of meeting production requirements at each stage of the line. The system must have a fast-turnaround a capability, *i.e.*, it must be able to expedite high-priority lots such as prototypes without excessively retarding the production of other material. The scheduler must respond to random (*i.e.*, unplanned) and planned events that disrupt the fabrication process, including machine failures, material shortages, and operator absences. It should also schedule maintenance. Scheduling computations must be fast; the line must never be held up while movements are calculated. The scheduling algorithm should be based on historical experience on machine reliability, yield, and operation times, and it must be able to update estimates of such quantities as mean time between failures (MTBF). It should be able to provide such data to management as when a given lot will be completed. Due-date information must be available as soon as a fast-turnaround lot is loaded.

In this dissertation we are concerned with the problem of shop floor control in the first stage of VLSI manufacturing, in which many devices are simultaneously built up on a wafer of silicon. It takes place in a clean room environment known as a semiconductor fab. Partly because of contamination problems, some of these facilities are among the few existing examples of *paperless factories*, in which the status of

machines and inventories is available in real-time from the factory CIM system. Hence, real-time decisions about dispatching (which job to do next when a machine becomes available) and lot release (of raw wafers into the factory) could be based on the global factory state. The possibility of making decisions based on an expanded information set raises two research questions:

- How should global information be summarized and used for decisions?
- How much improvement can one expect as a result, compared with decisions based on local information?

We do not consider high-level production planning in this study. For all purposes, we assume an exogenous entity (*e.g.*, a high-level scheduling module [Lea86a]) provides the short-interval scheduling module with a product mix and lot start rates.

In this introductory Chapter we briefly outline what is to follow in the next six Chapters of this dissertation.

A modern integrated circuit fabrication facility commonly produces devices using one or more fabrication processes. Presently, the three most important processes are the Bi-Polar, NMOS (n-channel MOSFET), and CMOS (complementary MOSFET) fabrication processes. Each process has associated with it a sequence of process steps that wafers of that type, grouped in lots, must follow. Within a process, a product is defined by the set of masks used in photolithography. After fabrication products are further broken down into sub-products (bins) according to performance characteristics, *e.g.* clock speed, power dissipation, etc. The number of products a typical fab may produce can range in the hundreds. Lots are moved around in the fab, competing for scarce resources, *e.g.* unreliable equipment (processing and materials handling), labor, shelf space (buffer), and raw materials.

Market conditions generate varying demands for IC products. High-level production planning models analyze these demands and generate out schedules, product by product, that guide the shop-floor scheduler in making its decisions.

There exists no consensus as to what should be the objective of a scheduling policy. One natural objective is the minimization of average cycle time, given a target average throughput rate. Cycle time is defined to be the time a job spends in the fab, i.e. processing time plus waiting time, while throughput rate is the average number of jobs that leave the fab per unit time. Minimization of cycle time, and consequently waiting time, among many effects, decreases the time a wafer is exposed to particles in the clean room, thus increasing yield, improves the response time to changes in the demand pattern, reduces in-process inventory, and reduces the engineering response time. It also reduces many hidden costs, e.g. production control personnel costs, and production losses due to hot-lot dispatching. Possibly the most significant cost of high inventory levels is its contribution to low yield. Some machine failures result in process malfunctions that are not detected until final test, so everything that has passed the failed machine in the interim may have to be scrapped.

Scheduling policies can be compared based on the mean delay/mean throughput tradeoff curve (Fig. 1.1). This curve, $D(t)$, describes the average delay or waiting time as a function of fab throughput, t. It is well known that as throughput approaches the capacity of the fab the average queueing time goes to infinity.

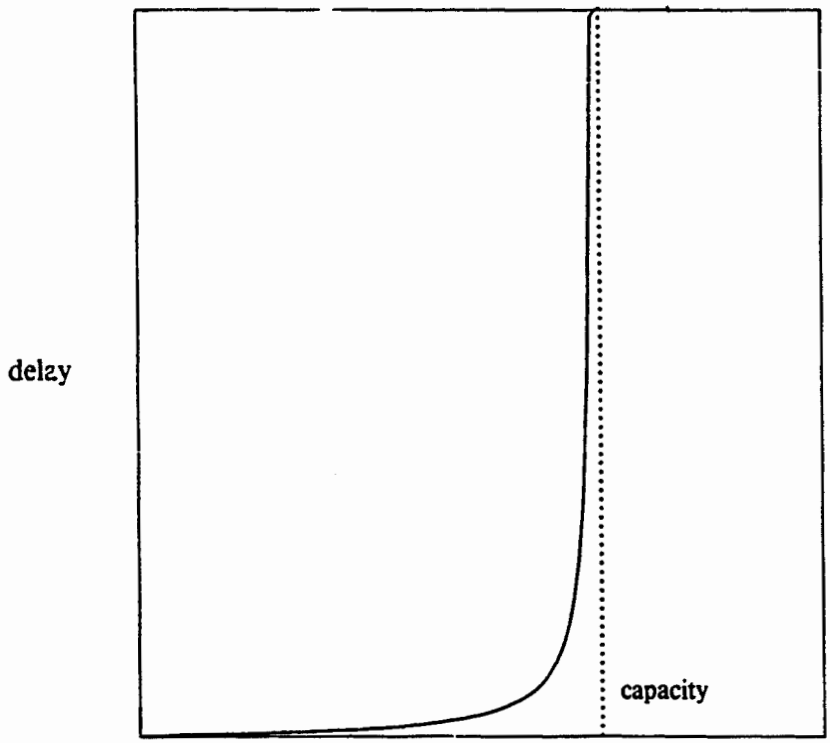

Fig. 1.1 - Mean Delay-Mean Throughput Trade-Off Curve

We say that a scheduling policy S_A is superior to policy S_B for a given throughput interval T if $D_A(t) < D_B(t)$ for $t \in$ T, where $D_A(t)$ and $D_B(t)$ are the tradeoff curves for policies S_A and S_B, respectively (see Fig. 1.2).

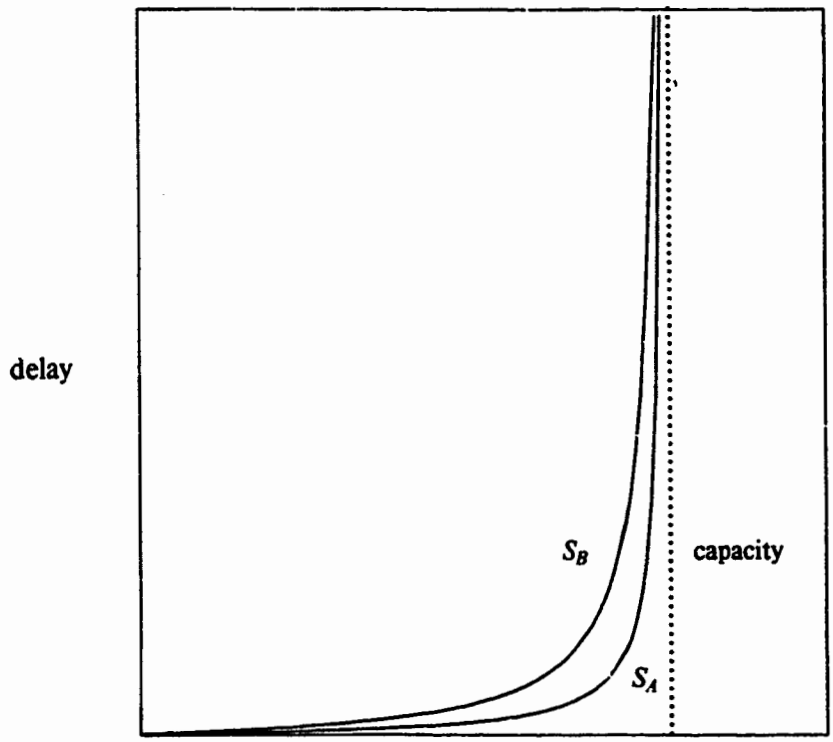

Fig. 1.2 - Scheduling Policy Comparison

Our objective is to find decision policies that will be efficient on the frontier of delay (or inventory level) and throughput. By Little's Law [Lit61a], the expected inventory of work awaiting processing (WAP) equals *average delay* x *flow rate.* Consequently, for a fixed output rate, reducing WAP reduces time spent in the shop. However, reducing WAP will tend to reduce throughput by: (1) reducing the buffering between work stations so that machine downtime at any station will have a high probability of forcing idle time elsewhere due to lack of work; and (2) reducing batch sizes so a greater fraction of machine time is spent on set ups. In this paper we are concerned only with the first tradeoff.

We limit ourselves to one particular aspect of wafer fab scheduling, *i.e.* scheduling wafers on a fab with unreliable processing equipment, having at most two bottleneck stations (such as photolithography and ion implant). There are several other scheduling problems that require further investigation, *e.g.* operator scheduling, scheduling of operations that have a long cycle time, scheduling of batch operations, and scheduling with set-up or changeover times.

The plan for the remainder of this dissertation is as follows. In Chapter 2 we discuss at a superficial level the processes involved in transforming a slice of raw silicon into a wafer of integrated circuits. We begin by presenting an overview of the overall manufacturing process (fab, sort, assembly, test) and then describe in more detail the process of fabrication. We describe deposition and epitaxy, oxidation, introduction of impurities, photolithography, and metallization.

In Chapter 3 we review the literature of job shop scheduling (in particular, job shop dispatching) and scheduling and simulation of semiconductor wafer manufacturing. The assumptions most found in the literature are listed, as are the most cited system performance measures. The notion of equivalent performance measures is reviewed. We enunciate several important theoretical results for the single-machine scheduling problem. For the multi-machine scheduling problem we review attempts at solving the problem exactly and then concentrate on heuristics for approximate solutions. In particular we review the literature of job shop dispatching heuristics and job shop simulation research. We list the relevant work related to the most studied simple dispatching rules as well as to more complex heuristics and weighted dispatching rules. Optimization techniques for weighted dispatching rules are surveyed. We conclude the Chapter reviewing the more recent literature of shop-floor level production planning in the semiconductor industry.

In Chapter 4 we introduce a queueing model of the dynamics of a semiconductor wafer fab and present a C programming language implementation of the model.

The model is a generalization of the classical job shop, allowing for unreliable parallel machines, multiple products, deterministic and stochastic (rework) job routes, deterministic or stochastic inter-station travel times, regular- and high-priority (hot) jobs, and two levels of scheduling control – job dispatching and job release. The C implementation of the model is a discrete-event simulation program called *FabSim*. Every aspect of the model is implemented in FabSim. Moreover, the program allows numerous dispatching and release control combinations, including optimal multi-rule dispatching with Hooke and Jeeves pattern search for optimization. FabSim collects statistics and displays them for analysis. Several improvements to both the model and the simulation program are suggested.

In Chapter 5 we introduce a class of new job release mechanisms called Starvation Avoidance (SA). We define the concept of *virtual inventory* in job shop scheduling and adapt ideas from inventory management to control work release into the shop. In inventory management when the inventory falls below a given safety stock, a new order is made. In our analogy, when the *virtual inventory*, *i.e.* the work content (on hand) at the bottleneck plus the work content (on order) moving towards the bottleneck (and expected to arrive within a limited time frame) falls below a given safety stock level, a new job is released into the shop. We discuss a dispatching scheme designed to aid SA achieve its objective, *i.e.* minimize idle time on critical shop equipment.

In Chapter 6 we compare several dispatching rules and release strategies on hypothetical and real data. We first compare 17 scheduling policies on a small but representative fab queueing network. Then we compare Starvation Avoidance with Uniform release on several large fab networks. FabSim is used for the simulation experiments, which are carried out on the Cray X/MP-14 supercomputer at U.C. – Berkeley.

Concluding remarks are made in Chapter 7.