Chapter 7 — Concluding Remarks

In this dissertation we investigate the problem of real-time scheduling of wafer batches on unreliable machines in a semiconductor manufacturing facility (fab). We assume global factory information is readily available through a computer integrated manufacturing system. 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, the expected inventory of work awaiting processing (WAP) equals *average delay* × *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 research we are concerned only with the first tradeoff.

Some features of wafer fabrication are not found in most other job shops. Because each layer of the wafer requires exposure of a photoresistive material through a mask (the process is known as photolithography) each batch of wafers makes many visits to the photolithography work station. Since this equipment is very expensive, this work station is often the bottleneck. The special feature of work flow in wafer fabrication is that it is re-entrant at a bottleneck work station. A scheduling policy should focus on the queue of work at the bottleneck because it will on average be the biggest queue in the factory and furthermore, whenever the bottleneck work station is forced to become idle due to lack of work, that lost time represents an unrecoverable loss of final output.

One conclusion we have reached from our work and that of others [Bur86a, Wei86a] is that dispatching decisions are not as important as release decisions. Effective flow control requires a sufficiently large inventory of raw material so that new work can be introduced whenever desired. While such policies shift

inventory from WAP to raw material, it is much less expensive to hold inventory in that form. The raw wafers can be made into any product, so the risk of obsolescence due to demand shifts or engineering changes is lower. They are not as subject to contamination and yield loss.

The most common release control used in VLSI fabrication is the open loop strategy of *uniform starts*, i.e. release new work into the shop at a constant rate equal to the desired output rate, and independent of current inventory levels or machine status. In fact, wafers are usually released in lots of 25 or 50. A second release strategy is the *Fixed-WIP* rule which starts a new lot of wafers whenever a lot of wafers is completed. Wein [Wei86a] has proposed a variation of the Fixed-WIP strategy in which inventory is measured, not by counting wafers, but by summing the remaining work to be performed at the bottleneck work station. This strategy releases wafers containing the equivalent of one hour of work at the bottleneck whenever bottleneck machine completes an hour of work. He calls this strategy *Workload Regulating* release. Our proposed strategy is similar to Wein's but is derived from a somewhat different approach. We start with a simple idea: To reduce inventory, do not start new work. The difficulty with that idea is that eventually the bottleneck work station starves and no finished work leaves. Hence, we are lead to the *starvation avoidance* rule: Start new work just in time to avoid idling the bottleneck work station due to lack of work.

Starvation Avoidance was tested extensively on several networks and compared to other scheduling policies. The following ranking of release strategies in order of increasing effectiveness: Uniform, Fixed-WIP, Workload Regulating and Starvation Avoidance seems to hold for all the experiments we performed. We would expect this ranking to hold for a very large class of shops for the following reasons: Any reasonable closed loop control should be better than open loop control, so the Uniform rule is the worst. All the closed loop rules adjust the arrival rate to the shop so it is nega-

tively correlated with the queue length at the bottleneck, (thus showing that increasing the variability of the arrival process to a queueing network does not necessarily increase delays). All the closed loop rules are equivalent if the bottleneck is the last operation before completed work leaves the shop (and each lot visits there only once). Otherwise, a simple thought experiment suggests why Fixed-WIP is the worst of the closed loop controls. Imagine a breakdown of the last work station. Then no work leaves, so under Fixed-WIP no new work starts, and, if the breakdown lasts long enough, the bottleneck starves. But both SA and Workload Regulating ignore all lots that have passed the bottleneck for the last time and continue to feed work into the bottleneck. Similarly, if the bottleneck station breaks down, Fixed-WIP will continue to pile up the inventory of new work in front of the bottleneck (until everything after the bottleneck has left the shop), but both SA and Workload Regulating will stop releases. In straight line flow shops, SA and Workload Regulating are equivalent. In re-entrant flows, Workload Regulating counts all the work remaining at the bottleneck on each lot, not just the next bottleneck operation. Let L = 0.9 hour and consider two cases of an (almost) empty shop. In case 1, two jobs are in queue at the bottleneck, each with 0.5 hours of work at the next bottleneck operation. SA would not start a new lot (assuming α = 1). Now suppose only one job is in queue with two bottleneck operations to be performed each taking 0.5 hours, and one hour of processing on some other machine between them. Workload Regulating would not distinguish between these cases, but SA would start a new lot in the second case.

We can make the following conclusions about SA.

- SA is an effective scheduling policy in that it produces near-capacity throughput while maintaining the average job delay considerably lower than when traditional job release policies are used.

- Near capacity, SA outperformed all other policies in all test cases.
- As is the case with inventory control, SA is sensitive to the randomness of the lead time. If the lead time from new wafer starts to the bottleneck work station increases in length or variability, not only will the tradeoff curve shift upwards (longer delays for any output rate) but the relative reduction in delay achieved by SA compared with UNIF will be not as dramatic. This is clearly seen in comparing the two versions of the Silicon Systems model. This is an example of a common phenomenon in control systems: Introducing time lags or noise in the feedback loop of a stochastic system will degrade performance. Randomness of lead time can be reduced by having the first bottleneck visit occur early in the process recipe.
- SA is easy to implement, provided a CIM system with up-to-date shop floor information is available. In practice, the throughput control parameter α used in SA must be set. This parameter controls the factory throughput. One way to set α is with simulation. This, however, requires a validated simulation model which may not be available. Another approach is as follows: Run the fab for a period of time releasing work with the UNIF control policy and measure the virtual inventory W. By using $\alpha = W / L$ as the control parameter setting (where L is the lead time for replenishment) then with high probability the throughput will be larger than the average throughput measured under the UNIF release policy. Since near capacity the slope of the delay/throughput curve for SA is steep, by reducing α one can still maintain a high throughput rate, while reducing considerably the mean delay. In practice, α can be decreased slowly, with careful monitoring of the corresponding delay and throughput rates, until a desired point on the delay/throughput curve is reached.

Future research should determine if the results for the single process, single bottleneck case described in this thesis hold for the several multi-process, multi-

bottleneck cases. In this dissertation we describe only one interpretation of the SA principle. With other definitions of virtual inventory (e.g. making use of improved lead time and equipment downtime estimates) other interpretations of SA can be investigated.