Reduction of Average Cycle Time at a Wafer Fabrication Facility

Subhash C. Sarin and Sameer T. Shikalgar

Grado Department of Industrial and Systems Engineering
Virginia Tech
Blacksburg, VA 24061
540-231-7140
e-mail: sarins@vt.edu

Abstract

This paper is concerned with the development of effective solutions for the reduction of average cycle time at a wafer fabrication facility. The wafer fabrication environment is quite different from the usual flow shop or job shop environments, with a distinguishing feature being the reentrant flow of the lots through the system. Lots at different stages of their manufacturing cycle may revisit the machines. This gives rise to the need of effective policies to sequence lots through the system.

Two methodologies have been developed to effect a reduction in the cycle time. The first methodology is a heuristic procedure based on the idea of reducing idle time on the bottleneck machine. The second methodology is based on mathematical programming.

The proposed methodologies are implemented using the data obtained from the M/A-COM's wafer fabrication facility in Roanoke, Virginia. The facility consists of ninety-two machines and its products can be classified into six different types. The performance of one of the proposed methodologies is compared with that of the policies currently followed at the M/A-COM facility in Roanoke, Virginia and the results are presented.

INTRODUCTION AND PROBLEM DEFINITION

The importance of semiconductor wafer fabrication scheduling has been increasing steadily over the past decade. Wafer fabrication is the most technologically complex and capital intensive phase in semiconductor manufacturing. It involves the processing of wafers of silicon and gallium arsenide in order to build up the layers and patterns of metal and wafer material. Many operations have to be performed in a clean room environment to prevent particulate contamination of wafers. Also, since the machines on which the wafers are processed are expensive, service contention is an important concern. All these factors underline the importance of seeking policies to optimize the performance with respect to some objective. M/A-COM is one such group of companies which has various wafer fabrication facilities in the U.S.

M/A-COM Inc., at its Virginia based Roanoke facility, provides Gallium Arsenide (GaAs) integrated circuit solutions (IC) to the commercial and military wireless and radar markets. M/A-COM designs and manufactures a broad line of GaAs microwave and millimeter wave products using wafer fabrication techniques.

M/A-COM's clean room manufacturing environment consists of 5 different areas. These are Photolithography, Process I, Process II, Materials, and Implant, which are shown in Figure 1. Typically, wafers are prepared for manufacturing and introduced to manufacturing in the Materials area. Wafers are then transported to Photolithography Area to apply the desired circuit patterns with a substance called photoresist. The patterns are cleared from photoresist in the Process I Area. These patterns are then implanted with the desired ions to give the necessary properties to the pattern at the Implant Area. In the Process II area, metal is deposited to obtain the "Gate" metal. This is a typical sequence to form certain properties on a circuit.

Depending on the type of circuit design, a wafer goes through 100 to 200 process steps over a period of a few weeks. Another unique property of wafer fabrication is that the fabrication is not linear through the facility but wafers may revisit machines during different stages of their manufacturing. Wafer fabrication planning and control is complicated by random disturbances, namely, the reentrant flow of operations, random yields due to the special nature of the operations, random machine breakdowns and random repair times.

Presently it is observed that the lots spend a substantial portion of their cycle time waiting in queue for a machine. This leads to an increase in the overall Work-In-Process (WIP) and a reduction in throughput. The main areas for the analysis of this problem can be identified as follows:

- **Operations Bottlenecks:** Bottleneck machines govern the cycle time of the entire operation. Identification of bottleneck machines is essential so as to sequence the jobs in an optimal way at those machines. Optimal sequencing at the bottleneck machines reduces the cycle time considerably.

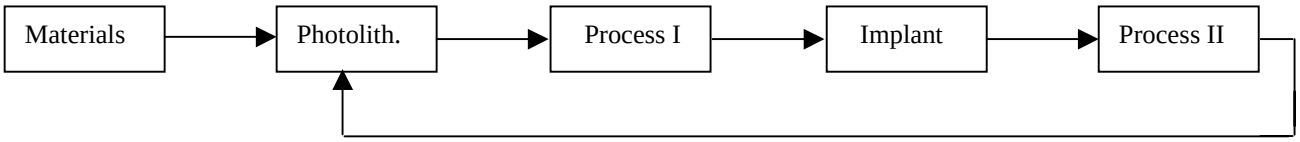

Figure 1. Manufacturing Areas of M/A-COM

- **Work-In-Process (WIP):** Excessive WIP represents low efficiency of the operation. It is necessary to keep WIP at a minimum to reduce locked up capital and to ensure the smooth flow of jobs through the system.

- **Equipment Utilization:** Equipment utilization needs to be monitored to ensure effective utilization of all the machines. Under-utilization of machines can result in increased delays and cycle time. Equipment utilization needs to be maintained at some predetermined level to obtain increase in efficiency.

The problem that is addressed can concisely be defined as follows: given the number of lots in queue at a particular workstation, which lot needs should be processed next so that the average cycle time of the entire system is reduced. This problem is entirely a sequencing problem since the sequence for the processing of the lots at a particular machine is considered.

A few unique issues arise while considering the scheduling of lots in a semiconductor manufacturing environment. First, lots at the same machine could either be at the start of their manufacturing route or towards the end or anywhere in between. Hence, there is a need to determine if lots, in any one of the above-mentioned classes, dominate over the others in terms of reducing the objective. If lots in one class consistently outperform the others, then it would be beneficial to schedule those first so as to reduce mean cycle time. However, if no one class is dominant, other factors need to be considered to attain the minimum average cycle time.

The second issue relates to the lot size of the parts flowing through the system. Each lot consists of a number of identical wafers. These wafers flow through the system as a single unit and are never split. The number of wafers in a lot is determined by the cassette size (in which the wafers are kept) and the number of wafers that can be processed at the same time on each machine. If the time taken by each lot to go from one machine to the other is negligible, it would be beneficial to have a lot size equal to the smallest number of lots a machine can take. If, however, the transit times are large, it would be advantageous to have larger lot sizes.

The third issue relates to the capacity of each machine. Some machines may be capable of processing only a single lot at a time. However, there are a few machines, e.g., ovens, which are able to process multiple lots at a time. This depends on the type of lots and the type of operation that needs to be done. These machines have a significant impact on the cycle time of the products and, hence, this factor needs to be taken into account while considering the sequencing of the lots.

In this paper, we briefly describe two methodologies in order to reduce average cycle time. Some preliminary results are presented and application to a real life case is also described.

BRIEF DESCRIPTION OF METHODOLOGIES

Sarin and Kalir [1] have proposed the idea of minimization of idle time at the bottleneck machine. They prove that “When sufficiently small lot sizes are utilized, minimizing the idle time on the bottleneck machine must necessarily result in an optimal schedule.”

Using this idea, a Bottleneck Minimal Idleness (BMI) heuristic has been presented which endeavors to minimize the idle time at the bottleneck. This is done by sequencing the lots in decreasing order of closeness of their secondary bottleneck machine to the primary bottleneck machine. ‘Secondary bottleneck machine’ means the upstream machine with the next largest unit processing time after the bottleneck machine (which is referred to as the primary bottleneck machine). By utilizing this approach, some of the idle time that might have been created on the machines closer to the bottleneck machine is absorbed, because it overlaps with the processing of the previous lots.

All the analysis done for this heuristic is applied to the flow shop environment. For further details of the algorithm and results refer to the paper [6].

The idea of minimizing the idle time at the bottleneck machine can be extended to the reentrant nature of a wafer fabrication environment.

Consider an example of four lots, two each, of part type A and B. The start times for the two lots of part type A are {0, 6} and the start times for the two lots of part type B are {0, 6}. The route and the processing times of part types A and B are given below:

Table 1. Processing Time Data for Example 1

Route for Part A	Processing Times for Part A	Route for Part B	Processing Times for Part B
Machine 1	3	Machine 1	3
Machine 2	12	Machine 3	4
Machine 1	4		
Machine 3	3		

For this example, it can be seen that machine 2 is the bottleneck machine. Hence, the proposed algorithm will try to minimize the idleness at machine 2. Hence, the schedule is as follows:

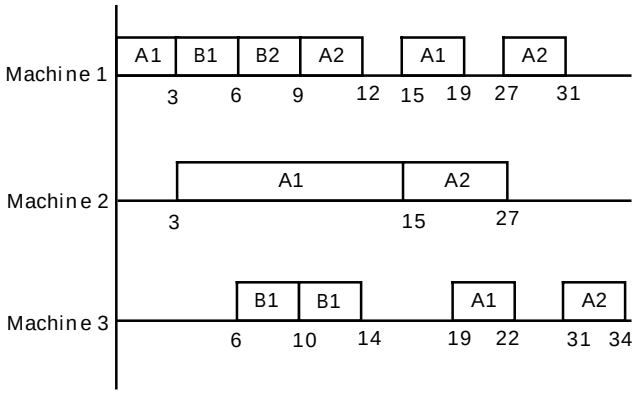

Figure 2. Gantt Chart for Experiment 1

The completion times of the lots are as follows:

Table 2. Completion Time Table for Experiment 1

Lot Type	Completion Time	Ready Time	Cycle Time
A1	22	0	22
B1	10	0	10
A2	34	6	28
B2	14	6	8

Mean Cycle Time = (22+10+28+8)/4 = 17 units

Hence, the mean cycle time for the lots is 17 units. Also, notice that at time 6, when lots A2 and B2 enter the system, lot B2 is chosen over lot A2 since it does not cause any idleness at the bottleneck machine.

If at time 0, at machine 1, lot B1 were to be chosen before lot A1, the schedule would have been:

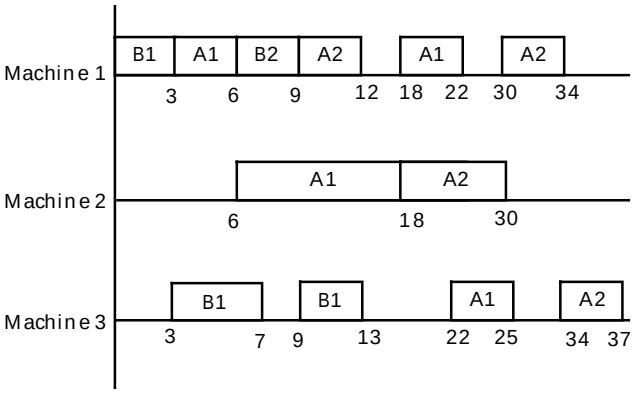

Figure 3. Gantt Chart for Experiment 2

with the completion times:

Table 3. Processing Time Data for Example 1

Lot Type	Completion Time	Ready Time	Cycle Time
A1	25	0	25
B1	7	0	7
A2	37	6	31
B2	13	6	7

Mean Cycle Time = (25+7+31+7)/4 = 17.5 units
The mean cycle time for the lots is 17.5 units.

Hence, a delay in the start of the bottleneck machine results in an increase in the average cycle time of the lots.

Also, in schedule 1, if at time 6, lot A2 were to be chosen over lot B2, the corresponding schedule would be:

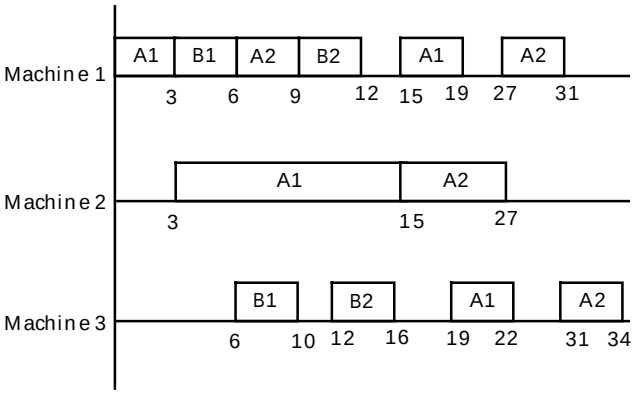

Figure 4. Gantt Chart for Experiment 3

with the completion times:

Table 4. Completion Time for Experiment 3

Lot Type	Completion Time	Ready Time	Cycle Time
A1	22	0	22
B1	10	0	10
A2	34	6	34
B2	16	6	10

Mean Cycle Time = (22+10+34+10)/4 = 17.5 units

Here, lot B2 was delayed unnecessarily in favor of lot A2. In spite of processing lot A2 first, there is no improvement in its cycle time since it is waiting for the bottleneck machine to complete its processing. Hence, when looking at the idleness of the bottleneck machine it is also necessary to check if any lots, which have lesser processing time remaining, can be processed without causing any delay in the system.

Therefore, while it is necessary not to cause idleness at the bottleneck machine, if the lot with lesser percent of processing time remaining, can be processed without causing any idleness on the bottleneck machine, it should be done first.

PROPOSED ALGORITHM

Determination of the Bottleneck. The bottleneck machine is defined as the machine at which has the cumulative time of all the lots in the system, at that instant is the largest. Let:

M – number of machines in the fabrication area m=1,…,M
L_i – identification of lot i
Q – set of all the lots in the system Q={L_1,L_2,…,L_N}
p_{im} – the processing time of lot i on machine m
BN – the bottleneck machine;

$$BN = \arg \max_{1 \le m \le M} \left\{ \sum_{L_i \in Q} L_i \cdot p_{im} \right\}$$

Dynamic Nature of the System. The system that is modeled is a dynamic system, i.e., lots enter and leave the system at regular intervals of time. This could cause the bottleneck machine to change from one machine to the other, since the bottleneck machine is a function of the lots in the system at that instant. This necessitates calculating the bottleneck regularly. Hence, to take into account the dynamic nature of the system, the bottleneck machine is calculated each time a decision needs to be made at the machine. This takes into account the various changes that the system may have undergone in between two decisions at a particular machine.

Batching Machines. Some machines are capable of processing more than one lot at a time. These machines are referred to as batching machines. The batching machines have a specified minimum batch size, as well as, a specified maximum batch size, which they can process at one time. Each time a batching machine selects lots from its queue to be batched, it checks if the minimum batch size has been attained. If the minimum batch size has not been attained, then it will wait for more lots to enter its queue.

Machine Downtimes. The machines are subject to failures and calibrations. Both these events are defined as downtime for a machine.

Least Percent Processing Time Remaining. Each lot in the station's queue is initially sorted in the order of least percent processing time remaining first. This is calculated as follows:

$$\text{percent processing remaining} = \frac{\text{remaining processing time}}{\text{actual processing time + remaining processing time}}$$

The actual processing time is the processing time the lot has incurred up to the current step. As seen in the previous examples in this chapter, by unnecessarily delaying lots having least percent processing time remaining, the average cycle time of the system increases. Hence this rule is used to initially give preference to lots which can leave the system quickly.

On the basis of the minimization of idle time on the bottleneck and checking the least percent processing time remaining a new algorithm is proposed which endeavors to minimize the idle time at the bottleneck and hence reduces the average cycle time of the lots in the system. The algorithm is as follows:

Step 1. The lots in the present station's queue are sorted on the basis of least percent processing time remaining. If $L(\theta_1)$ is the percent processing time remaining of lot θ_1, then the lots are arranged in the order of θ_1, θ_2,θ_3,θ_4, …… where $L(\theta_1) <= L(\theta_2) <= L(\theta_3) <= L(\theta_4)$ ……

Step 2. The bottleneck of the entire system is calculated based on the lots, which are in the system.

Step 3. After calculating the bottleneck of the system, the lots in the p resent station's queue are divided into three groups. Group 1: The lots visiting the bottleneck during the remainder of their route. While looking at the stations on the remainder of a lot's route, the status of the

station is also determined. If the station is down or will be down when the lot reaches it, the lot is **not** place din group 1. Also the lot with the least time to the bottleneck is identified. If B is the set of lots visiting the bottleneck during the remainder of their route and the time to the bottleneck for lot θ_1 is $t_{\theta 1}$ then lot θ_B is the lot with the least time to the bottleneck $q_B = \min_{q_l \hat{I} B} \{tq_l\}$. Group 2: The lots not visiting the bottleneck during the remainder of their route and the stations along the remainder of the route are not down or will not be down when the lot reaches it. Group 3: All the remaining lots.

Step 4. After dividing the lots into groups, the status of the bottleneck machine is determined. If the bottleneck machine is idle and if group 1 is not empty, then lot θ_B, is scheduled.

Step 5. If group 1 is not empty and if the bottleneck machine is not idle, the lots in group 2 are checked to see if by scheduling any one of them now will it subsequently cause any idleness on the bottleneck machine. This is done by adding the processing time of the lot on the present machine and the time taken by the lot θ_B. Ties are broken using the least percent processing time remaining.

Step 6. IF no lots from group 2 are scheduled and group 1 is not empty, the lots of group 1 are checked to see if by scheduling any of them now, will there be any idleness caused on the bottleneck. If a lot does not cause any idleness on the bottleneck, it is scheduled next. Ties are broken using the least percent processing time remaining. If allots cause idleness on the bottleneck, then lot θ_B is scheduled next.

Step 7. If group 1 is empty and group 2 is not empty, the lot with the least percent processing time remaining is scheduled among lots in group 2.

Step 8. If group 1 and group 2 are both empty, the lots in group 3 are now divided into two sub groups. Sub Group a: The lots visiting the bottleneck along the remainder of their route. Sub Group b: The lots not visiting the bottleneck along the remainder of their route. The difference between these groups and groups 1 and 2 is that the status of the stations along their route are not considered.

Step 9. The same rules are applied as above by replacing group 1 with sub group a and group 2 with sub group b.

A multiple bottleneck version of the above algorithm has also been devised. It follows the same logic as above, except that it identifies two bottlenecks (primary and secondary) and tries to minimize the idleness on both the bottlenecks (giving preference to the primary bottleneck first). This is useful when there is no single dominant bottleneck in the system, but two machines, which are the bottlenecks.

METHODOLOGY 2

The second approach is a mathematical programming based approach. Linear and integer programming based models are widely used to analyze manufacturing systems. The advantage of these mathematical models is that they can be changed easily to take into account any changes in the manufacturing system. Also, since they almost always provide optimal solutions, their use can considerably increase the efficiency of the system. To this end, an integer programming model of the manufacturing system at M/A-COM is formulated and a procedure for efficiently solving the model is being developed.

The integer program has been formulated considering binary variables to be the starting time of the operation of a particular job on a particular machine. Using these binary variables the necessary constraints are formulated. The CPLEX software is used to solve this integer program. CPLEX is a non graphic based linear and integer programming software which can read text files containing constraints, objective function and variables and uses them as input for the integer program. Hence, a C++ program is written which can read the files containing route information, such as processing time, and machine information and convert them into the appropriate constraints. Since integer programs require a lot of computer time to solve and also use up a lot of computer memory, various modifications are proposed, which help in reducing the integer program into simpler programs. This would be useful in solving the problem in reasonable time.

RESULTS

The fabrication area of M/A-COM, Roanoke was simulated using the AutoSched AP software. The fabrication area consists of 96 stations, which are classified into station families. There are six different product types, which flow through the system in the form of lots consisting of 8 wafers each. The data for the study was provided by M/A-COM. This corresponded to lots entering the system from March 20, 2000 to August 28, 2000. These were 694 in all. A warm-up period of 35 days was assumed. Hence, the cycle times of all the lots completed before April 24, 2000 are not included. The simulation runs were terminated on the last day (August 28, 2000) a lot is fed into the system. Thus, the cycle times of only the last lot completed between April 24,

2000 and August 28, 2000 are included in the results. The results are presented in Table 5. From the results, it can be seen that the proposed multiple bottleneck rule outperforms the existing system.

Table 5. Comparison of Performance Measures for Methodology and Existing System

No.	Algorithm/Rules	Avg Cycle Time	Std Dev Cycle Time	% of Jobs Late	Avg Lateness	Var of Lateness
		Days	Days		Days	Days
1	Multiple Bottleneck	16.82	7.96	35.14	8.66	9.11
2	Bottleneck	23.02	9.6	64.09	10.16	9.24
3	Existing System	20.85	10.11	52.22	10.92	9.13

CONCLUDING REMARKS

In this paper, a new methodology is proposed to reduce cycle time for processing lots in a fab. The proposed methodology is described and implemented using the data provided by the M/A-COM facility in Roanoke, Virginia. The results show that the proposed methodology performs significantly better than the system that is followed currently to process lots.

REFERENCE

Kalir, Adar A. and Sarin, Subhash C., "A Near-Optimal Heuristic for the Lot Streaming Sequencing Problem in Multiple-Batch Flow-Shops," submitted to OMEGA, (2000).