Proceedings of the 2008 Winter Simulation Conference
S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, J. W. Fowler eds.

SCHEDULING A MULTI-CHIP PACKAGE ASSEMBLY LINE WITH REENTRANT PROCESSES AND UNRELATED PARALLEL MACHINES

Sang-Jin Lee
Tae-Eog Lee

Dept. of Industrial and Systems Engineering
335 Gwahang-no KAIST
Yusung-gu, Deajeon, Korea

ABSTRACT

A multi-chip package(MCP) consists of several chip modules in a single package. We consider a scheduling problem for assembling MCPs. In order to assemble an MCP, a lot should repeat assembly process stages such as die attach and wire bonding as many as the number of chips to be assembled. The two key process stages have many parallel machines of various types. A machine processes different types of MCP lots with significant setup times. We therefore should limit the number of setups significantly while not sacrificing the on-time delivery performance much. We propose scheduling strategies of appropriately allocating the machine capacity to products and lots depending on the production progress of products and lots. We report experimental performances of the proposed methods.

1 INTRODUCTION

Semiconductor manufacturers have increased capacity of memory chips continually. In addition to shrinking circuit widths, they recently assemble multiple chips into a single packaging module. Such multi-chip packaging is primarily intended to increase the capacity of flash memories such as NOR or NAND flash memories or RAM such as SRAM/PSRAM, DRAM or UtRAM packaging modules. However, heterogeneous chips, which are used together for specific applications, are often combined in a single packaging module. For example, a 2.5Gb MCP is made of two NAND flash memory chips and two mobile DRAM chips. An MCP is used for RF power amplifiers and mobile devices such as cellular phones, PDA, digital still cameras, and video cameras. Recent leading edge MCPs combine up to 20 chips.

Once a wafer is fabricated and probed in a FAB, the chips are detached from the wafer, assembled into a plastic mold package, and tested. Chips undergo a series of assembly *process stages*: BL(Backlap) and TM(Tape Mounting), sawing, DA(Die Attach), cure, plasma, WB(Wire Bonding), and molding. An MCP repeats the assembly process stages from DA to WB as many as the number of chips to be assembled shown in Figure 1. A lot for a product with three chips undergoes the *process steps*, DA1, WB1, DA2, WB2, DA3, and WB3, where each number after the process stage name indicates the number of visits to the process stage. The key process stages, DA and WB processes, have many parallel machines, which may have different processing capacity and characteristics. For instance, a typical MCP assembly line has 80 DA machines of five classes and 240 WB machines of six classes, produces several different MCPs simultaneously as many as 20,000∼30,000 MCPs each month, and keeps WIP(Work-In-Progress) of around 3,000 lots. Typical packaging cycle times of MCPs are around one or two weeks. Setup times of a DA or WB machine for switching from one MCP type to another MCP type are 10∼60 minutes. Since the production planning system determines the daily or shift production requirements for an MCP assembly line to meet the order due dates with the minimum work-in-progress(WIP), we should schedule the MCP assembly operations so as to maximize the on-time delivery(OTD) rate with an affordable number of setups. Excessive setups may cause long machine waiting for setup since the number of staffs for setup operations is rather limited. Excessive setups also reduce the effective capacity of the line and the throughput. One the other hand, too large batching at a machine to reduce setups leads to job waiting and hence degrades the OTD performance. We therefore should pursue an appropriate trade-off between the OTD performance and the number of setups.

The MCP scheduling problem can be viewed as a hybrid flow shop, which processes multiple lots in a sequence of processes, each with unrelated parallel machines. An MCP line has many heterogenous parallel machines and the machines require significant setup times and setup technical staffs. The machines at each process stage are shared by operations for different products and lots of different assembly stages. Therefore, it is essential to appropriately allocate the machine capacity to products and lots depending on the pro-

978-1-4244-2708-6/08/$25.00 ©2008 IEEE

duction progress to meet the production requirements while reducing setups to an affordable level. Although there have been numerous works on scheduling versions of reentrant flow shops Pan and Chen (2005), Pan and Chen (2003), Demirkol and Uzsoy (2000), Kang, Kim, and Shin (2007), Choi, Kim, and Lee (2005), Choi and Kim (2008), Chen, Chen, Wu, and Chen (2007), Bertel and Billaut (2004), Liu and Wu (2004), Chen, Pan, and Wu (2007), Chen, Pan, and Wu (2008), Chen, Pan, and Lin (2008), most of them assume single machine or identical parallel machines for each process stage and no setup time.

In this paper, we examine a scheduling problem for a real MCP assembly line. We propose scheduling strategies of appropriately allocating the machine capacity to products and lots depending on the production progress so that the number of setups is effectively reduced while the on-time delivery performance is not significantly sacrificed. We report experimental performances of the proposed methods.

2 SOLUTION APPROACHES

Since there are many machines and products with complex process flow, we may consider traditional scheduling or dispatching rules that mostly assign individual lots to individual machines. However, we expect that the machine capacity should be better allocated to jobs or lots with different attributes. The reasons follow. The production plan and the lot assembly progress are quite different for each product. Lots in progress are characterized by the final product and the number of chips attached so far. The numbers of chips in lots may be different even if the lots are for the same product and in the same assembly step. There are many machines with different speeds, of which available times depend on the production progress or schedule. Furthermore, they are shared by many heterogeneous jobs with these different attributes. Production of some products may be behind the daily production plan. Lot flows of a product through its process steps may not be well streamlined, and a process step of a product may be a bottleneck when the machines are too less allocated to the process step. The number of total setups highly depends on how the machines are allocated to the jobs or lots. Therefore, it is essential to appropriately allocate the machine capacity to different products and lots so that the production plan is met as possible and the number of setups is reduced. We consider alternative strategies of allocating the machine capacity to the jobs as illustrated in Figure 2. We will explain them in detail.

2.1 Allocating Machines to Products for Expediting Delayed Products

When the number of completed products of a specific type is behind the production plan, the lots in progress for the product type should be expedited. To do this, we allocate the machines in a process stage to each product type in advance so that more machines are allocated to a product type behind the schedule. The machines allocated for a product type are shared by lots for the same product type. Once the machines are allocated to each product type, individual lots are scheduled by a dispatching rule. For a time bucket of length b, the number of chips for product p to be assembled at process stage s is $d_p \times r_{ps}$, where d_p is the number of chips required for product p for the time bucket and r_{ps} is the number of reentrances of lots of product p at process stage s. d_p is a weighted sum of the required amount of chips and work-in-progress of product p in order to prevent machine idle. We let c_{ps} indicate the average number of chips for product p processed per unit time at a machine for process stage s. Since the machines are not identical, it is averaged over the machines in the process stage allocated for the product. Therefore, the required number of machines ξ_{ps} for processing lots of product p in process stage s is the same as $\frac{d_p}{b} \times r_{ps}/c_{ps}$. This is written as

$$\xi_{ps} = \frac{d_p r_{ps}}{b c_{ps}}. \tag{1}$$

Then, the lots in a process stage are dispatched to the pre-allocated machines as follows.

PROD: Allocating Machines to Products

1. Compute the required number of machines for each product in process stage s as ξ_{ps}.
2. Sort the products in the decreasing order of ξ_{ps}'s and sort the machines according to their available epoches.
3. Allocate the machines to product p as many as ξ_{ps} in the order of the products. Let Λ_{ps} be the set of allocated machines for product p. If there is no machine to be allocated, stop.
4. Allocate the lots in a process stage to the machines in Λ_{ps} that are available during the time bucket.

2.2 Allocating Machines to a Virtual Lot

To reduce the number of setups, we should increase the run size at a machine. However, it also increases the WIP and degrades the OTD performance. To prevent this side effect, we introduce a notion of virtual lot. In a queue for a process step, a *virtual lot* is a set of lots for a product that require the same assembly operation. By assigning all lots of a virtual lot to a machine or a limited number of machines, we can reduce the number of setups. If a virtual lot has too many lots, there can be significant difference between the start time of the first lot and the last one. Therefore, the

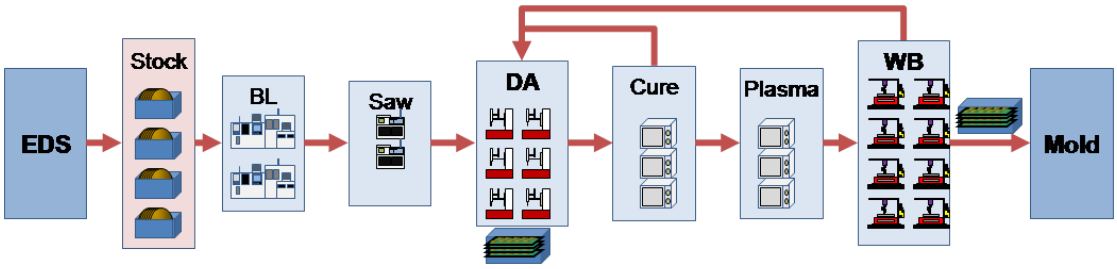

Figure 1: MCP assembly processes .

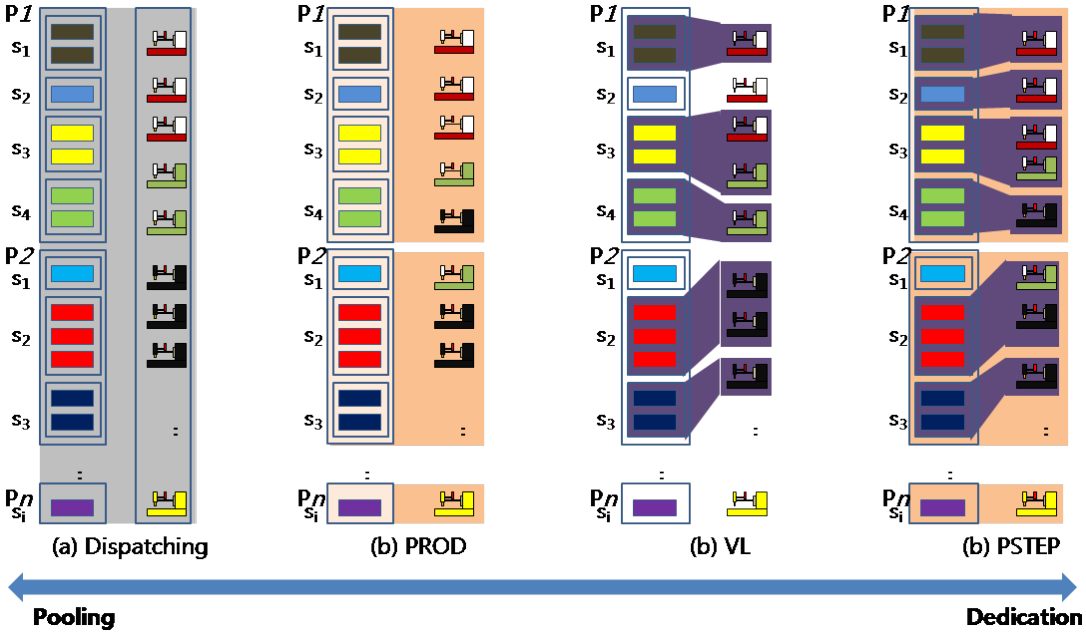

Figure 2: Alternative machine capacity allocation strategies to the jobs.

virtual lots should be reconfigured periodically from the lots in the queue. As the lots in a virtual lot are identical, the first available lot is assigned to the first available machines among available machines.

When the virtual lot is too large, the run size at a machine increases. This tends to increase the WIP or make other machines idle. We therefore determine an appropriate number of machines for processing a virtual lot. Let q_{pk} be the number of chips of a virtual lot of product p in process step k. Then, similarly as in equation (1), $\frac{q_{pk}}{b}$ is the time for a machine to process a virtual lot. Therefore, we can calculate the required number of machines by dividing $\frac{q_{pk}}{b}$ by the average processing time for process step k, c_{pk}. This is written as

$$\eta_{pk} = \frac{q_{pk}}{bc_{pk}}. \tag{2}$$

The scheduling procedure based on the method of allocating the machines to a virtual lot follows.

VL: Allocating Machines to Virtual Lots

1. Form a virtual lot by grouping lots of each product with the same number of chips attached.
2. Determine the required number of machines for each virtual lot of each product in process step k, η_{pk}.
3. Allocate the machines to a virtual lot for product p and process step k as many as η_{pk}. Let Ω_{pk} be the set of allocated machines for the product p and process step k. If there is no machine to be allocated, stop.

4. Allocate lots in a process step to the machines in Ω_{pk} that are available during the time bucket.

2.3 Allocating Machines to Process Steps of Products

Once the machines are allocated to each distinct process step of each product in advance, the logical job flows for each product can be viewed as a flow line as illustrated in Figure 3 even though the job flows are reentrant. For example, as seen in the figure, product 2 has three chips and hence is assembled by repeating DA and WB process stages three times. Therefore, the lots of product 2 go through six steps of DA and WB, each of which has dedicated machines. It is called a *logical flow line* (LFL) for the product. Once a LFL for a product has balanced workloads for the process steps, it ensures smooth flow of jobs and shorter and less variable cycle times. Therefore, the OTD performance can be improved significantly. Tang et al. (Tang, Zhou, and Qiu 2003) introduce a similar concept called virtual production line for different purposes, dynamically changing the logical line configuration to cope with disturbances such as machine breakdowns, adding new machines, or production plan changes.

We should appropriately allocate the machines to the products and assign the allocated machines of each product to the process steps of the product so that the production plan is met and setup changes are limited. Similarly as in *PROD*, the machines are allocated to each product so that the lot progress of delayed products are expedited. To reduce the number of setup changes, the method of allocating machines to virtual lots of each process step can be used. This also balances the workloads between the process steps of the LFL for each product. Consequently, we combine *PROD* and *VL* to allocate the machines to the process steps of each product. We now summarize the scheduling procedure as follows.

PSTEP: Allocating Machines to the Process Steps of Each Product

1. Allocating machines to products.
   (a) Calculate the required number of machines for the lots of each product p in process stage s as ξ_{ps}.
   (b) Sort the products in the decreasing order of ξ_{ps}'s and sort the machines according to their available times.
   (c) Allocate the machines to product p as many as ξ_{ps} in order of the products. Let Λ_{ps} be the set of allocated machines for product p.
2. Balancing among the process steps.
   (a) Compute the required number of chips for each process step k using η_{pk}.
   (b) Sort the required number of machines for each process step k in the descending order.
   (c) Allocate the machines in Ω_{pk} to each process step k as many as η_{pk}.
3. Scheduling lots for each process step with a simple dispatching rule like SPT, FIFO.

2.4 Known Dispatching Rules

We also consider conventional dispatching rules. The priority of each job is computed from the job attributes such as the process time, the due date, the job type, or the machine setup state. Each time a machine is freed, a lot with the highest priority among the waiting lots is assigned to the machine (Haupt 1989). We experiment well-known dispatching rules, FIFO(First In First Out), MFIFO(Modified FIFO selects the earliest available job considering the setup time), SPT(Short Processing Time), MSPT(Modified SPT selects the earliest finished job considering the setup time), MS(Minimum Slack), EDD(Earliest Due Date), ATCS(Apparent Tardiness Cost with Setups), CR(Critical Ratio), S/OPN(The Smallest Ratio of Slack Per Number of Operations Remaining), and S/WKR(The Smallest Ratio of Slack Per Work Remaining). ATCS, MFIFO, and MSPT consider setups.

3 EXPERIMENTAL RESULTS

We examine the performances of the proposed machine allocation strategies by simulation. A test case is defined by taking a combination from three levels on the number of products to be produced, (25, 30, 35), four levels on the required number of chips for each product, (3, 4, 5, 6), three choices in the production plan, (Type 1, Type 2, Type 3). Type 1 means that the total number of products of each type is similar and its daily distribution is rather uniform. Type 2 indicates that 30% of product types amounts to 70% of the total number of products and the product proportion is the same for each day. In Type 3, 30% of product types amounts to 70% of the total number of products, but the daily product proportion is not constant. We also randomly generate four test cases for each combination. For each proposed scheduling method, we test 216 cases.

We focus on only two key processes, DA and WB, because other processes have sufficient capacity. There are five classes of DA machines with the total number 80. There are six classes of WB machines with the total number 240. The production plan is given for three days. The arrival times of chips to the stock is known for the three days in advance. We examine the OTD rates and the number of machine setups for the time horizon.

For each scheduling method, we compute a relative efficiency measure called *relative deviation index*. For a proposed scheduling method, the relative deviation index(RDI)

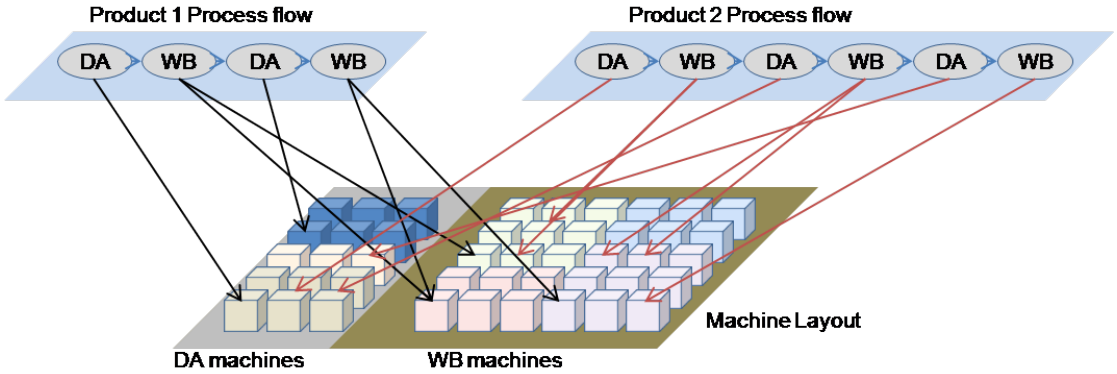

Figure 3: Logical flow line.

is computed for each test case as $|x-x_b|/|x_w-x_b|$, where x is the performance value of a test case of the proposed method, and x_b and x_w are the best value and the worst value, respectively, among the performance values of all scheduling methods (Choi, Kim, and Lee 2005). We compute the average RDI_{avg} and the standard deviation RDI_{std} for the performance values of all test cases for each proposed scheduling method. The smaller RDI_{avg} the proposed scheduling method has, the better the proposed method is. That is, $RDI_{avg} = 0$ means the best and $RDI_{avg} = 1$ indicates the worst.

The experimental results are summarized in Table 1. NBS indicates the number of test cases for which the proposed scheduling method generates the best performance. Since there are some test cases that have the same OTD value, the NBS sum of OTD performance is 236, higher than the total number of test cases. The conventional dispatching rules, especially MS, EDD, CR, SRMOP, and SRMWK, which all give higher priority to jobs with tighter due dates or the shorter remaining time to the due dates, have high OTD rates. However, they have excessive setups, beyond the upper limit, around $3,000$, that can be handled by the setup staffs. We note that MFIFO, MSPT, and even ATCS, which consider setups, all have excessive setups, around $7,000$. *VL* has the worst OTD performance, but the best performance in the number of setups, $1,450$, which is about 1/5 of the conventional dispatching rules. *PSTEP* makes a good trade-off between the OTD rates and the number of setups. It reduces setups significantly without much sacrificing the OTD performance. We observe that proper allocation of the machine capacity to products, virtual lots, or process stages or steps significantly reduces setups while sacrificing the OTD rates moderately.

4 CONCLUSION

We examine scheduling problems for an MCP assembly line that has reentrant job flows and significant setups for product switching, and should meet the production plan. We proposed strategies of allocating the machine capacity to products, virtual lots, or process stages or steps so that the number of setups is reduced while sacrificing the OTD performance reasonably. We found that we can reduce the number of setups significantly by allocating the machine capacity to the process step of each product appropriately while sacrificing the OTD rates not so much. We should further develop the OTD performance of *PSTEP*.

REFERENCES

Bertel, S., and J.-C. Billaut. 2004. A genetic algorithm for an industrial multiprocessor flow shop scheduling with recirculation. *European Journal of Operational Research* 159:651–662.

Chen, J., K. Chen, C. Wu, and W. Chen. 2007. A study of the flexible job shop scheduling problem with parallel machines and reentrant process. *International Journal of Advanced Manufacturing Technology* only online available.

Chen, J.-S., C.-H. Pan, and C.-M. Lin. 2008. A hybrid genetic algorithm for the re-entrant flow-shop scheduling problem. *Expert Systems with Applications* 34:570–577.

Chen, J.-S., C.-H. Pan, and C.-K. Wu. 2007. Minimizing makespan in reentrant flow-shops using hybrid tabu search. *Interanational Journal of Advanced Manufacturing Technology* 34:353–361.

Chen, J.-S., C.-H. Pan, and C.-K. Wu. 2008. Hybrid tabu search for re-entrant permutation flow-shop scheduling problem. *Expert Systems with Applications* 34:1924–1930.

Table 1: Performances of the proposed scheduling methods.

		OTD				Setup number			
		avg.	RDI_{avg}	RDI_{std}	NBS	avg.	RDI_{avg}	RDI_{std}	NBS
Dispatching	FIFO	0.678	0.720	0.190	0	7456.0	0.995	0.011	0
	MFIFO	0.687	0.630	0.203	0	6782.7	0.884	0.027	0
	SPT	0.685	0.647	0.237	0	7277.4	0.965	0.022	0
	MSPT	0.686	0.638	0.204	1	6905.0	0.904	0.029	0
	MS	0.744	0.137	0.139	30	7218.1	0.956	0.021	0
	EDD	0.730	0.252	0.150	14	7336.3	0.975	0.012	0
	ATCS	0.669	0.788	0.214	0	7192.6	0.952	0.033	0
	CR	0.751	0.084	0.114	52	7208.4	0.954	0.018	0
	S/OPN	0.750	0.086	0.117	60	7207.9	0.954	0.018	0
	S/WKR	0.751	0.081	0.107	63	7203.8	0.953	0.019	0
PROD		0.697	0.544	0.265	8	4545.7	0.512	0.130	0
VL		0.658	0.890	0.147	0	1458.6	0.001	0.003	212
PSTEP		0.695	0.565	0.276	8	2002.6	0.091	0.040	4

Choi, S.-W., and Y.-D. Kim. 2008. Minimizing makespan on an m-machine re-entrant flowshop. *Computers and Operatons Research* 35:1684–1696.

Choi, S.-W., Y.-D. Kim, and G.-C. Lee. 2005. Minimizing total tardiness of orders with reentrant lots in a hybrid flow shops. *Interanational Journal of Production Research* 43:2149–2167.

Demirkol, E., and R. Uzsoy. 2000. Decomposition methods for reentrant flow shops with sequence-dependent setup times. *Journal of Scheduling* 3:155–177.

Haupt, R. 1989. A survey of priority rule-based scheduling. *OR Specktrum* 11:3–16.

Kang, Y.-H., S.-S. Kim, and H. Shin. 2007. A scheduling algorithm for the reentrant shop: an application in semiconductor manufacture. *International Journal of Advanced Manufacturing Technology* 35:566–574.

Liu, M., and C. Wu. 2004. Genetic algorithm using sequence rule chain for multi-objective optimization in re-entrant micro-electronic production line. *Robotics and Computer-Integrated Manufacturing* 20:225–236.

Pan, J.-H., and J.-S. Chen. 2003. Minimizing makespan in re-entrant permutation flow-shops. *Journal of the Operational Research Society* 54:642–653.

Pan, J.-H., and J.-S. Chen. 2005. Mixed binary integer programming formulations for the reentrant job scheduling problem. *Computers and Operational Research* 32:1197–1212.

Tang, Y., M. Zhou, and R. Qiu. 2003. Virtual production lines design for back-end semiconductor manufacturing systems. *IEEE Transactions on Semiconductor Manuacturing* 16:543–550.

AUTHOR BIOGRAPHIES

Tae-Eog Lee is a Professor of Department of Industrial and Systems Engineering at KAIST. His interests include semiconductor manufacturing scheduling and automation, and discrete event system modeling and scheduling. He is an associate editor of IEEE Transactions on Automation Science and Engineering. His e-mail address for this proceeding is <telee@kaist.ac.kr >.

Sang-Jin Lee is a Ph.D. candidate at the same department. His work includes scheduling semiconductor manufacturing systems. His e-mail address for this proceeding is < sangjin.lee@kaist.ac.kr >.