

## A SYNCHRONOUS ALGORITHM TO REDUCE COMPLEXITY IN WAFER FLOW

**Hilario L. Oh**  
[ohl@svg.com](mailto:ohl@svg.com)  
Silicon Valley Group, Inc.  
San Jose, CA

**Tae-Sik Lee**  
[tslee@mit.edu](mailto:tslee@mit.edu)  
Massachusetts Institute of Technology  
Cambridge, MA

### ABSTRACT

Single-wafer processing yields better on-wafer result than a batch process. This is because a single-wafer processing provides superior process parameter control. A single-wafer processing has to rely on parallel processing at high speed with extensive use of redundant process modules and transporters to compete with the throughput of batch processing. A cluster of the many single-wafer modules and transporters may result in a complex wafer movement, which requires complex coordination of wafer processing and wafer transport.

This paper shows that the complex wafer movement is a consequence of the coupling between the wafer-processing functional requirement and the wafer-transport functional requirement. By adding “planned delays”, i.e., queues, to the process time of the non-critical process steps, it is shown that the two functional requirements can be successfully de-coupled. The consequences of the de-coupling are (1) a synchronization of the two functional requirements and (2) a drastic reduction in the number of wafer flow paths. Item (1) ensures that processed wafers are transported always in a timely manner and, thus, the consistency in on-wafer result and throughput is improved. Item (2) minimizes the need of orchestration of wafer flow, and allows a consistent wafer process history. The end result is a reduced wafer-to-wafer variation in on-wafer result.

A real life example of queuing to de-couple the wafer processing and the wafer transporting in a photo-resist processing system is presented to illustrate the concept and the methodology.

**Keywords:** axiomatic design, decoupling, complexity, scheduling

### 1 INTRODUCTION

Single-wafer processing yields better on-wafer result than processing hundreds of wafers in batch. This is because one can control the process parameters better in a single-wafer module. Another advantage is the flexibility in capacity planning that the single-wafer processing provides. This flexibility is limited in batch-wafer processing. However, single-wafer processing has to rely on parallel processing at high speed with extensive use of redundant process modules and transporters to match the throughput of batch-wafer processing system. Typically, parallel

processing at high speed is implemented through a single-wafer cluster tool.

A single-wafer cluster tool refers to a group of single-wafer process modules organized around a group of wafer transporters to perform sequentially a series of process steps on the wafer [Perkinson, et al.]. Figure 1 shows a cluster tool with five process modules organized around one transporter. Wafers enter and exit the cluster tool through a buffer called the load port. The load port serves as the interface between the fab and the cluster tool. Once the transporter takes a wafer from the load port, the wafer is transported sequentially through the series of modules for processing. Figure 2 is a schematic representation of the wafer movement in the cluster tool. The lengths of the bar indicate the process and transport times. The process time is the time from when a wafer enters a module for processing to when the wafer is ready to exit. The transport time is the time required for a transporter to move a wafer between two modules. Process time of some steps is so critical that it can not tolerate transport delay. A series of process steps, together with the process and transport time associated with each step, constitutes a recipe.

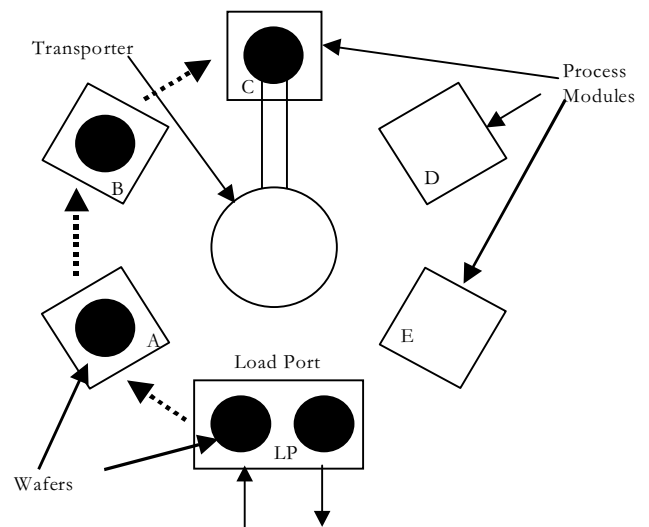
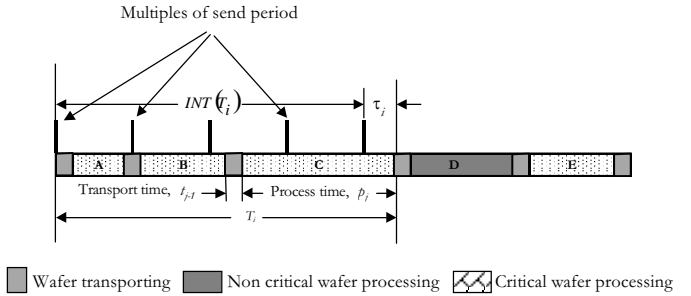


Figure 1. Schematic of a five-module cluster tool



**Figure 2. Timing diagram of processing and transporting a wafer**

To satisfy the throughput requirement, given in wafer per hour (WPH), a series of wafers are sent through the cluster tool successively at a constant send period; see Figure 3. This send period, SP, is given by

$$SP = \frac{3600}{WPH}(\text{sec})$$

By the sixth wafer (for a recipe with five process steps), the cluster tool will be fully populated with wafers. For every wafer exiting the cluster tool, there is another wafer entering to replenish it. All process and transport tasks performed on the wafers occur in a periodic fashion, the periodicity being the send period. When the system reaches this state, it is said to be in steady periodic state. The movement of the wafers, i.e., the wafer flow, in a cluster tool under steady periodic state is predictable. Its management, i.e., the orchestration of the wafer processing and the wafer transporting, determines the throughput and the on-wafer result delivered by the cluster tool.

## 2 AXIOMATIC DESIGN PERSPECTIVE OF WAFER FLOW

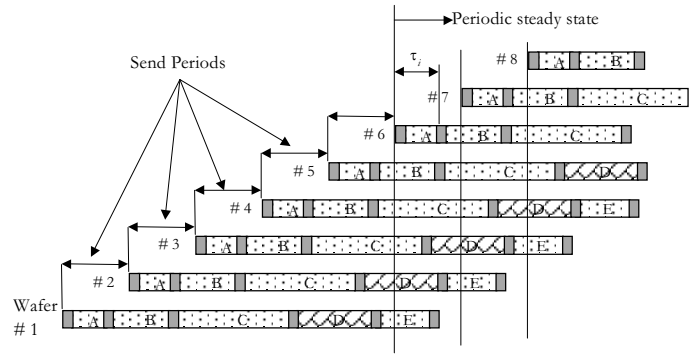
### 2.1 SOURCE OF "COUPLING"

The management of wafer flow can be viewed from an axiomatic design perspective; see Table 1.

**Table 1. Decomposition of wafer flow**

	DP1: number of process modules	DP2: number of transporters
FR1: to process wafer per recipe and throughput required	A11 = X	A12 = ?
FR2: to transport processed wafer in a timely manner	A21 = X	A22 = X

The number of process modules DP1, necessary to satisfy FR1 is determined by the required throughput and the number of process steps with associated process times called for by the recipe. For example in Figure 3, at least five process modules are needed to process wafers in five process steps. Additionally, one redundant module must each be added to the system for the



**Figure 3. Cascade of wafers**

process steps C and D to effect parallel processing of consecutive wafers. Without the redundant modules, consecutive wafers can not be processed sequentially at these process steps. This is because a wafer, e.g., wafer #3, that has just been processed at the process module B (or C) can not leave for the process module C (or D) since it is still occupied by the wafer, wafer #2, ahead of it. This situation occurs whenever the process time of a process step plus the transport times to and away from that step is longer than the send period required by the throughput requirement.

The choice of the number of process modules DP1, to satisfy FR1 will affect FR2 (A21). For example, by choosing DP1 to satisfy FR1, the particular recipe and the throughput requirement would generate the timing of the transport tasks indicated by dark gray bars shown in Figure 3. Note in the figure that wafers at process steps A, B, and D, complete their process and demand their transport all at the same time. Thus sufficient number of transporters, DP2, should be available to transport the processed wafers away from A, B, and D to satisfy FR2 (A22). If the number of transporters DP2 is not sufficient to handle all of these demands, then some of the wafer transports have to be delayed until a transporter is available. This situation, wherein more than one wafers call for transport within a time interval shorter than the transport time of a transporter, is hereafter called "transport conflict." When a transport conflict occurs, a wafer at the most critical process module is transported first while the others at the less critical process modules are kept waiting, i.e. delayed.

In principle, DP2 will not affect FR1 (A12= "O") if every process module is provided with a dedicated transporter. In reality, this is not possible because increased number of transporters complicates module layout, tool footprint and cost. Invariably, the number of transporters are insufficient, and thus it causes transport delays and affects FR1 (A12= "X"). From an Axiomatic Design perspective, wafer flow in cluster tool as described above is inherently coupled.

### 2.2 INCREASED COMPLEXITY AS A CONSEQUENCE OF "COUPLING"

The root cause of increased complexity in wafer flow comes from the increased parallel flow paths necessary to process wafers with long process time at high throughput. As described earlier, whenever the process time  $p_i$  of a process step plus the

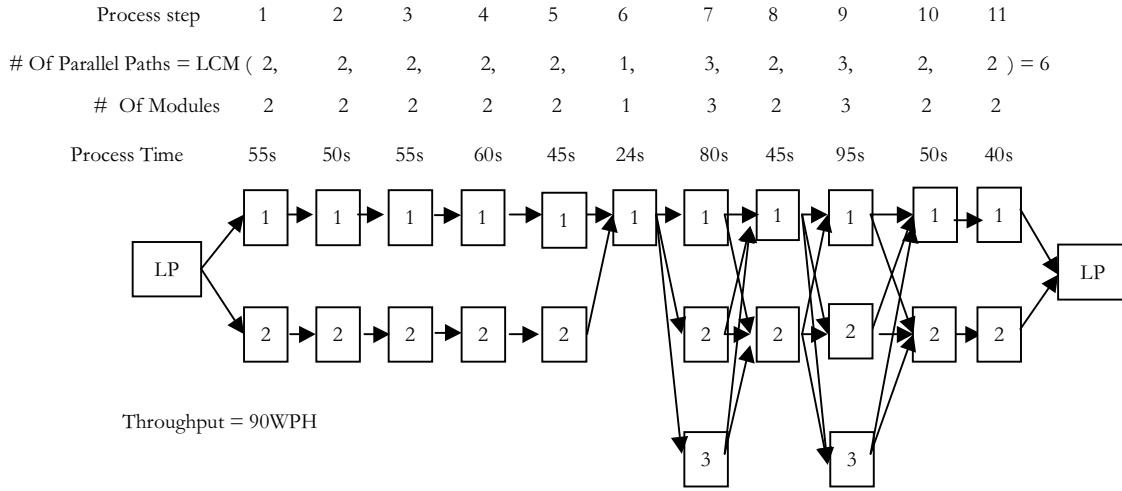


Figure 4. Parallel flow paths in a cluster tool

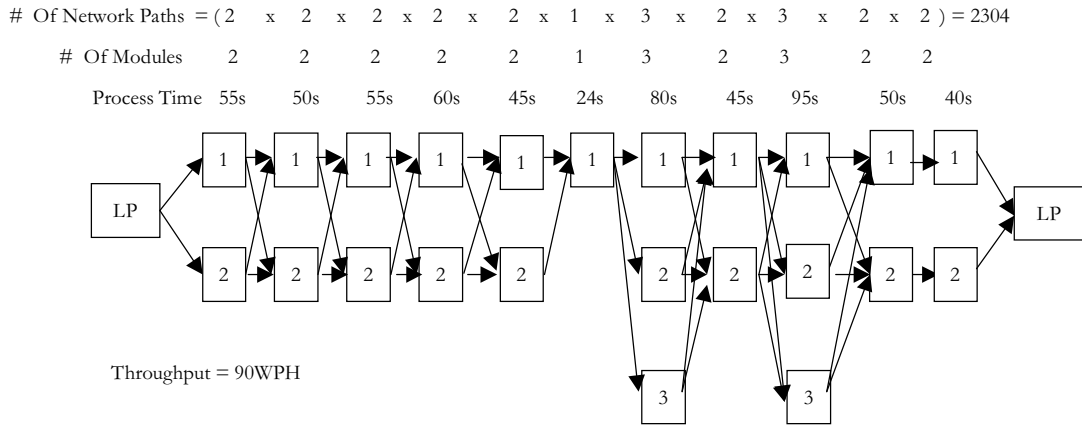


Figure 5. Degeneration of parallel flow into a network of flow

transport times to and away from that step,  $t_{i-1}$  and  $t_i$ , is longer than the send period, redundant module is needed to effect parallel processing at that step. The number of modules  $m_i$  needed at the  $i$ th process step to effect parallel processing is:

$$m_i = 1 + INT \left( \frac{p_i + t_{i-1} + t_i}{SP} \right); i = 1, 2, \dots, N \quad (1)$$

where  $N$  is the total number of process steps in the recipe, and the symbol  $INT(\bullet)$  denotes a function that rounds a real number down to the nearest integer. This integer,  $INT(\bullet)$  on the right hand side of Equation (1), is the number of redundant module needed.

The consequence of adding redundant modules is an increase in the number of parallel flow paths. Consider for example the wafer flow shown in Figure 4. For those process steps with one redundant module, successive wafers will be

processed in the sequence of (1, 2, 1, 2, 1, 2, 1, 2,...); while for those process steps with two redundant modules, in the sequence of (1, 2, 3, 1, 2, 3, 1, 2,...). If the flow is maintained in a periodic steady state, then the 7th wafer will repeat the pattern of the 1st wafer, the 8th wafer will repeat the pattern of the 2nd wafer, ...etc. In other words, every 6th wafer will take the same path, 6 being the least common multiple (LCM) of (1, 2, 3). Thus, in a steady periodic wafer flow, the number of parallel flow paths is the least common multiple of the numbers of modules for each process step:

$$No. \text{ of parallel paths} = LCM (m_1, m_2, \dots, m_N) \quad (2)$$

Parallel wafer flow in the presence of transport conflicts will degenerate into a complex network of flow. This is so because the transport demands that were delayed in transport conflicts will alter the pattern of inter-arrival time of subsequent transport demands. The alteration may create unpredictable future

conflicts that may lead to instability of the wafer flow. The delayed demands could also spill over a send period onto the neighboring send periods and begin to destroy the periodicity of the wafer flow. Once the periodicity is lost, the inter-arrival time of transport demands becomes randomized. Driven by the random occurrence of the transport demands, a processed wafer will be sent to whatever process modules that happen to be available at the time when the demand is made. Thus, wafers may not go through the parallel flow paths as shown in Figure 4. Instead, they will go through a network of unpredictable flow paths to complete the process steps as in Figure 5. The original, relatively few, deterministic parallel flow paths as shown in Figure 4 will, in the presence of transport conflicts, degenerate into a complex network of thousands possible flow paths as in Figure 5. The latent throughput of a parallel flow can not be achieved without complex coordination of wafer processing and wafer transporting. The on-wafer result will vary from wafer to wafer because the wafers experienced varying process history as they travel through the myriad of flow paths. In other words, complexity of wafer flow increases as a consequence of coupling brought about by insufficient DP2 (A12= "X") to handle transport conflicts. To reduce the complexity, one has to eliminate the coupling through the identification and resolution of the transport conflicts. This is discussed in the next section.

## 2.3 SYNCHRONOUS ALGORITHM TO DE-COUPLE WAFER FLOW

### 2.3.1 Identifying transport conflicts

Given a recipe, the time a transport demand takes place can be derived as follows. Measure time from when a wafer leaves the load port; see Figure 2. Let the time when the wafer in the  $i$ th module is ready for pick up be  $T_i$ ;  $i = 1, 2, \dots, N$ . Since  $T_i$  is the accumulation up to the  $i$ th module of the process time  $p_j$ ;  $j = 1, 2, \dots, i$ ; and the transport time  $t_k$ ;  $k = 1, 2, \dots, i-1$ ; then

$$T_i = \sum_{j=1}^i (p_j + q_j) + \sum_{k=0}^{i-1} t_k; \quad i = 1, 2, \dots, N \quad (3)$$

In the above equation,  $q_i$  is an intentional delay, called queue, whose value is yet to be determined. By subtracting multiples of send period SP from  $T_i$ , the remainder  $\tau_i$  is the timing of the transport demand by a wafer at the  $i$ th module measured from the beginning of and within a send period; see Figures 2 and 3.

$$\tau_i = T_i - (SP) \cdot INT\left(\frac{T_i}{SP}\right); \quad i = 1, 2, \dots, N \quad (4)$$

Provided that the periodicity is maintained, the values of  $\tau_i$  will repeat themselves from one period to the next period. Transport conflict occurs within a send period whenever a pair of transport demands, the  $i$ th and the  $j$ th, are within a time interval shorter than the transport time of the transporter. Therefore the existence of conflicts can be identified by checking for the following inequalities among the  $N(N-1)/2$  pairs of  $(i, j)$  transport demands.

$$|\tau_i - \tau_j| < g; \quad j = 1, 2, \dots, (i-1); \quad i = 2, 3, \dots, N$$

where  $g \geq t_i$ ,  $i = 1, 2, \dots, N$  is the time allocated to the transporters for transport between process modules.

### 2.3.2 Queuing to resolve transport conflict

Since transport times  $t_k$  are fixed for a given cluster tool, it is clear from Equations (3) and (4) that the timing of transport demand  $\tau_i$  is solely dependent on the process times  $p_j$  as prescribed by the recipe. Thus if a recipe is such that it creates transport conflicts, the solution is neither to add more transporters nor to delay some of the conflicting transport demands. Instead, the conflicts should be resolved by modifying the recipe using queues  $q_i$ , i.e., intentional delays, to alter the process times. In other words, delays, instead of being the outcomes of transport conflicts, are deliberately inserted in the form of queues to ensure that transport conflicts do not occur in the first place. Adding queues to process times is easily realized since process modules can be conveniently used as temporary buffers for wafer. In the context of Axiomatic Design, the queues are the "de-couplers" that eliminate the transport conflicts, the sources of coupling in wafer flow.

However, queues can not be added indiscriminately to all process steps. Some process steps can not tolerate delay in picking up the processed wafer since it may adversely affect the on-wafer results. Such process steps are identified as critical process steps. The step whose process time is the longest among all process steps in the cluster tool, known as the gating step, is the bottleneck of the cluster tool. Process time at this step determines the throughput of the cluster tool. The gating step is also identified as critical process step since delays in this step will reduce the throughput of the cluster tool. Excessive delays that cause transport demands to spill over a send period onto the neighboring send periods and thus destroy the periodicity of the wafer flow must also be avoided. Therefore the basic strategy to add queues for transport conflict resolution is to implement the following steps.

- 1) Synchronize all activities to the send period, the heartbeat of the cluster tool. This means that all process times and transport times are normalized with the send period.
- 2) Provide sufficient transporters to complete all transport demands within one send period.
- 3) Insert queues at the non-critical steps in such a way that no delays occur at the critical steps and no transport conflicts occur.

Implementing Step (1) establishes periodicity in the system. This is central to synchronizing wafer-processing FR1 to wafer transporting FR2. Implementing Step (2) ensures that no transport task spills over the next send period. Thus, periodicity can be maintained. Implementing Step (3) ensures the periodicity that is established will not be destroyed by transport conflicts. Once the periodicity is established and maintained in the wafer flow, the processing and the transporting of wafer are in synchronization. The wafer flow will have a few deterministic parallel flow paths that require minimal coordination of wafer processing and wafer transporting. Since the flow is periodic, all

events planned or unplanned that occur during a period will terminate at the end of the period. The flow process is reset at the end of every send period.

Following Step (1), Equation (4) is divided into SP to obtain

$$\begin{aligned} \frac{\tau_i}{SP} &= \frac{T_i}{SP} - INT\left(\frac{T_i}{SP}\right) \\ &= \frac{1}{SP} \left[ \sum_{j=1}^i (p_j + q_j) + \sum_{k=0}^{i-1} t_k \right] - INT\left\{ \frac{1}{SP} \left[ \sum_{j=1}^i (p_j + q_j) + \sum_{k=0}^{i-1} t_k \right] \right\} \quad (5) \\ &; i=1,2,\dots,N \end{aligned}$$

To implementing Step (2), the number of transporters M necessary to accomplish all transport demands within one send period is

$$M \geq 1 + INT \left[ N / INT \left( \frac{SP}{g} \right) \right] \quad (6)$$

Finally, to follow Step (3), a set of queues  $q_j, j=1,2,\dots,N$ , to be added to the process time  $p_j, j=1,2,\dots,N$ , is searched for to ensure the following inequalities are satisfied for all  $N(N-1)/2$  pairs of (i, j) transport demands:

$$\left| \frac{\tau_j - \tau_i}{SP} \right| > \frac{g}{SP}; \quad j = 1,2,\dots,(i-1); i = 2,3,\dots,N \quad (7)$$

subject to the constraint that  $q_j^*$  for the critical steps are zeros.

Note that for a given cluster tool, the transport time  $t_j$  between modules are known. The process time  $p_j$  for all modules are also known for a prescribed recipe. Thus as indicated in Equation (5), the only unknown is the set of  $q_j^*$  that satisfy inequalities (7). While there are many solution sets, the preferred set is the one that gives the least sum of  $q_j^*$ . In other words, this is a constraint optimization problem involving the search for the  $q_j^*$  that satisfy inequality (7) subject to the constraint that  $q_j^*$  for the critical process steps are zero. The solution  $q_j^*$  once found would satisfy Step (3).

## 2.4 AN EXAMPLE

A photo-resist processing system is presented here to illustrate the concept and the methodology. Table 2 shows a system that performs eleven process steps. The process names and their process times are shown in the 2nd and the 3rd column of Table 2. The throughput requirement is 90 WPH. Therefore the send period is  $3600/90 = 40$  seconds. Since all process times except for "EXPOSURE" exceeds the send period, redundant modules are needed to meet the throughput requirement. The number of modules needed is calculated per Equation (1) in the 4th column. The least common multiple of these numbers is 6. Therefore, there will be 6 parallel flow paths calculated per Equation (2). These paths are shown in Figure 4.

Transport times between modules, equal to 7 seconds, are assumed to be the same for all transporters. With a send period of 40 seconds, a transport time of 7 seconds and 11 process steps, the number of transporters needed is at least 3 calculated per Equation (6). These are the CES, the MAIN and the STPR shown in the 5th column. The process modules are organized around the transporters as shown in Figure 6 and indicated in the 2nd and the 5th column: The cassette, the vapor prime, the chill, the hard bake and the chill module are organized around the CES; the chill, the resist coat, the soft bake, the post exposure bake, the chill, the developer and the hard bake modules are organized around the MAIN; the soft bake, the chill, the lithographic exposure and the post exposure bake modules are organized around the STPR.

Counting time from when a wafer leaves the cassette, column 6 shows the time when the ith process step demands a wafer transport. The timing of this transport demand is calculated per Equation (3) with queue  $q_j$  set to zero. In column 8, this timing is translated into the one referenced to the beginning of a send period, and is normalized with the send period using Equation (5). Since the normalized transport time is  $7/40 = 0.175$ , if the difference in timing of any pair of the transport demand in the 8th column is less than 0.175, there will be a transport conflict. The 9th column shows where and how

Table 2. Single-wafer photoresist processing

Throughput = **90 WPH** Sending Period, SP = **40 sec** Transport Time = **7 sec** Normalized Transport Time = **0.175**

Process				Robot Used	Pickup Time Per Recipe			Conflict	$q^{max}$	$q^*$	Pickup Time As Queued		
Step	Module	Time, $p_j$	# of Unit	In Picking	$T_j$	$T_j/SP$	$\tau_j/SP$	Pair	Allowed	Solved	$T_j$	$T_j/SP$	$\tau_j/SP$
0	CASSETTE	0		CES	0.00	0.000	0.000	1	24.000	0.000	0.00	0.000	0.000
1st	VAPOR PRIME	55	2	CES	62.00	1.550	0.550		24.000	9.860	71.86	1.796	0.796
2nd	CHILL	50	2	MAIN	119.00	2.975	0.975		24.000	12.100	140.96	3.524	0.524
3rd	RESIST COAT	55	2	MAIN	181.00	4.525	0.525	3	0.000	0.000	202.96	5.074	0.074
4th	SOFT BAKE	60	2	STPR	248.00	6.200	0.200	4	0.000	0.000	269.96	6.749	0.749
5th	CHILL	45	2	STPR	300.00	7.500	0.500		24.000	13.640	335.60	8.390	0.390
6th	EXPOSURE	24	1	STPR	331.00	8.275	0.275	4	0.000	0.000	366.60	9.165	0.165
7th	POST EXP BAKE	80	3	MAIN	418.00	10.450	0.450	2, 3	0.000	0.000	453.60	11.340	0.340
8th	CHILL	45	2	MAIN	470.00	11.750	0.750		24.000	10.120	515.72	12.893	0.893
9th	DEVELOP	95	3	MAIN	572.00	14.300	0.300	2	12.000	10.860	628.58	15.714	0.714
10th	HARD BAKE	50	2	CES	629.00	15.725	0.725		12.000	5.620	691.20	17.280	0.280
11th	CHILL	40	2	CES	676.00	16.900	0.900	1	24.000	0.280	738.48	18.462	0.462
0	CASSETTE	0											

Number of parallel paths = **6**

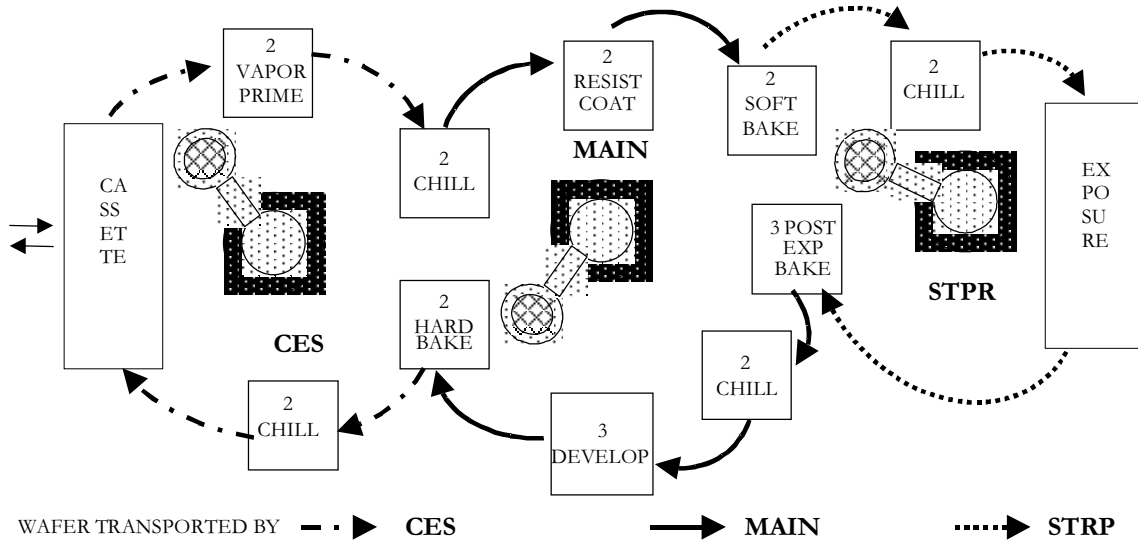


Figure 6. Assignment of modules to transporters

many these conflicts are. There is a total of 4 transport conflicts in this example. If these conflicts are not resolved, the parallel flow in Figure 4 will degenerate eventually to a network of flow as in Figure 5.

Genetic Algorithm, an optimization algorithm, is used to solve for the least sum of queue  $q_i^*$  that will resolve the transport conflicts per Equation (7). The maximum allowable queues for each process step are shown in the 10th column. Those steps with zero seconds are the critical process steps, those with 12 seconds are the somewhat critical steps; and those with 24 seconds are the non-critical steps. The Genetic Algorithm will search for the least sum of queue  $q_i^*$  within these maximum allowable values. The solution is shown in the 11th column. When this solution set is added to the respective process times in the 3rd column as prescribed by the recipe, the transport conflicts are resolved. None of the possible combinatorial pairs of  $\tau_i/SP$  shown in the 14th column have a difference in timing less than 0.175.

### 3 ACKNOWLEDGMENTS

Professor N. P. Suh, Massachusetts Institute of Technology, Cambridge, MA suggested the idea of a “de-coupler” to break the coupling of processing and transporting requirement.

### 4 REFERENCES

- [1] Perkinson T.L., Gyurcsik R.S., McLarty P.K., “Single-Wafer Cluster Tool Performance: An Analysis of the Effects of Redundant Chambers and Revisitation Sequences on Throughput”, *IEEE Transactions on Semiconductor Manufacturing Proceedings*, Vol 9, No. 3, pp.384-400, 1996.