

반도체 제조 트랙장비의 온라인 스케줄링 방법

윤 현 중* · 이 두 용†

(2000년 6월 15일 접수, 2001년 1월 19일 심사완료)

On-Line Scheduling Method for Track Systems in Semiconductor Fabrication

Hyun Joong Yoon and Doo Yong Lee

Key Words : Scheduling(스케줄링), Track System(트랙장비), Deadlock Avoidance(교착회피), Semiconductor Fabrication(반도체 제조)

Abstract

This paper addresses an on-line scheduling method for track systems in semiconductor fabrication. A track system is a clustered equipment performing photolithography process in semiconductor fabrication. Trends toward high automation and flexibility in the track systems accelerate the necessity of the intelligent controller that can guarantee reliability and optimize productivity of the track systems. This paper proposes an efficient on-line scheduling method that can avoid deadlock inherent to track systems and optimize the productivity. We employ two procedures for the on-line scheduling. First, we define potential deadlock set to apply deadlock avoidance policy efficiently. After introducing the potential deadlock set, we propose a deadlock avoidance policy using an on-line Gantt chart, which can generate optimal or near-optimal schedule without deadlock. The proposed on-line scheduling method is shown to be efficient in handling deadlock inherent to the track systems through simulation.

1. 서 론

트랙장비(track system)는 노광기(exposure)와 함께 반도체 제조시스템에서 사진공정을 담당하는 장비이다. 트랙장비는 반도체 감광재료인 포토레지스트(photoresist)를 웨이퍼에 도포한 후 이를 현상하는 코터 앤드 디벨로퍼(coater & developer)장비로서, 코팅(coting), 베이킹(baking), 쿨링(cooling), 디벨로핑(developing) 등의 프로세스 모듈로 이루어져 있다. 초기의 트랙장비는 프로세스 모듈에서 다른 프로세스 모듈로 웨이퍼를 이송하기 위해 오링(O-ring) 벨트, 에어베어링(air bearing), 워킹빔(walking beam) 시스템 등을 이용하였으나, 점차 트랙장비의 유연성(flexibility)이 높아지면서 이송 로봇을

이용하는 추세로 변하고 있다. 트랙장비에서 로봇 시스템의 도입으로 필요 없는 공정이나 고장이 난 프로세스 모듈을 거치지 않고 원하는 모듈에서 모듈로 웨이퍼를 직접 이송 시키는 것이 가능해져 트랙장비를 보다 효율적으로 운용할 수 있게 되었다.⁽¹⁾ 그리고, 로봇 시스템이 도입된 트랙장비는 각 프로세스 모듈에서 소요되는 가공시간의 균형을 맞추고 특정 모듈에서 생길 수 있는 병목현상을 해소하기 위해 일반적으로 동일한 기능을 하는 모듈을 두 개 이상 가지고 있는데, 이와 같은 병렬 모듈 시스템의 도입은 하나의 모듈이 고장이 나더라도 트랙장비 전체를 멈추지 않고 계속 작업하는 것이 가능하도록 하였다. 이러한 추세로 인하여 보다 복잡해진 트랙장비를 위한 능동적인 제어기의 개발이 중요한 문제로 부각되었으며, 높은 수준의 자동화 장비인 트랙장비의 운용 효율 높이고 생산성을 극대화 시켜줄 수 있는 교착상태를 회피하는 스케줄링(deadlock-free scheduling) 방법의 개발이 필요하게 되었다.

교착상태(deadlock)는 트랙장비를 포함한 자동화

* 한국과학기술원 기계공학과

† 책임저자, 회원, 한국과학기술원 기계공학과

Email : leedy@kaist.ac.kr

TEL : (042)869-3229 FAX : (042)869-3210

생산시스템을 운용하는데 반드시 해결해야 할 중대한 문제이다. Viswanadham et al.⁽²⁾는 페트리넷(Petri net) 모델을 이용하여 유연생산시스템(flexible manufacturing system)에서 교착방지정책(deadlock prevention policy)과 교착회피정책(deadlock avoidance policy)을 제안하였으며, Banaszak과 Krogh⁽³⁾는 페트리넷 모델과 ‘제한정책(restriction policy)’을 이용한 유연생산시스템의 교착회피정책을 제시하였다. Ezpeleta et al.^(4,5)는 페트리넷의 사이폰(siphon) 구조를 이용하여 교착방지정책을 제안하였고, Jeng et al.⁽⁶⁾ Xiong과 Zhou,⁽⁷⁾ Damasceno와 Xie⁽⁸⁾는 유연생산시스템에서 교착상태를 피하면서 동시에 생산성을 높일 수 있는 교착상태를 회피하는 스케줄링 방법을 페트리넷 모델을 사용하여 제시하였다. 교착상태 회피 문제에서 페트리넷 모델을 이용하는 접근 방법은 교착상태가 페트리넷 모델에서 표현이 가능하고 부품의 유연한 경로를 모델에서 쉽게 표현될 수 있다는 장점이 있다. 그러나, 페트리넷 모델은 기계나 로봇과 같은 자원의 수가 증가함에 따라서 모델이 갖는 상태의 수가 급격하게 증가하므로 실시간 제어 시스템에 적용하기가 어려운 단점이 있다. 페트리넷 외에 교착상태를 표현하기 위한 방법으로는 유향그래프(directional graph)를 이용하는 방법이 있다.^(9,10) Fanti et al.⁽⁹⁾는 유향그래프를 이용하여 현재 교착상태가 아니지만 미래에 반드시 교착상태가 되는 이차 레벨 교착상태(second level deadlock)를 정의한 후 다섯 가지 교착회피정책을 제시하였다. Lawley⁽¹¹⁾와 Reveliotis⁽¹²⁾는 노트(knot) 개념을 도입하여 부품들의 유연한 경로를 고려한 유연생산시스템에서 교착상태를 찾아내는 알고리즘과 교착회피정책을 제안하였다.

반도체 제조 장비의 성능 분석 및 스케줄링 분야에서 대부분의 선행 연구들은 클러스터툴(cluster tool)에 집중되어 있다.⁽¹³⁻¹⁸⁾ Perkinson et al.^(13,14)는 이론적인 모델을 이용하여 클러스터툴의 생산율을 분석하는 방법을 제시하였고, Venkatesh et al.⁽¹⁵⁾는 두 팔 로봇(dual-blade robot)이 있는 클러스터툴의 생산율을 분석하는 방법을 제시하였다. Srinivasan⁽¹⁶⁾는 페트리넷을 이용하여 클러스터툴을 모델링하고 성능 분석하는 방법을 제시하였다. 그러나, 클러스터툴 안에서 웨이퍼 흐름이 점차 복잡해짐에 따라 클러스터툴을 운용할 때 교착상태가 발생하지 않도록 웨이퍼 흐름을 스케줄링하는 방법의 개발이 요구 되었으며, Bodner⁽¹⁷⁾와 Lawley et al.⁽¹⁸⁾는 클러스터툴에 적용이 가능한 교착회피정책을

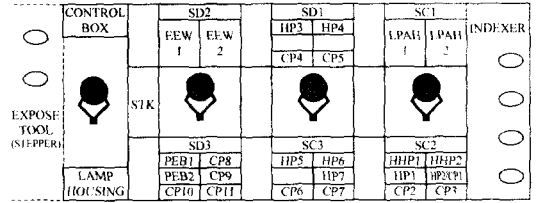


Fig. 1 Layout of a track system

제시하였다.

그러나, 일반적으로 트랙장비의 프로세스 모듈의 수와 로봇의 수는 클러스터툴보다 약 세 배에서 다섯 배 정도 많기 때문에 클러스터툴에 적용되는 교착회피정책을 실시간으로 트랙장비에 적용하는 것은 어렵다. 또, 트랙장비에서는 일부 모듈들 사이에서만 교착상태가 발생하고 그 외의 모듈들은 교착상태가 발생하지 않기 때문에, 클러스터툴 대상으로 개발된 교착회피정책을 트랙장비의 모든 모듈들을 대상으로 적용하는 것은 교착회피정책 적용하는데 필요한 계산의 복잡도 면에서 매우 비효율적이다. 따라서 본 논문에서는 트랙장비 내의 프로세스 모듈들 중 교착상태와 관련이 있는 모듈들의 집합인 교착가능집합(potential deadlock set, PDS)을 정의하여 이 집합에 포함되는 모듈들에 대해서만 교착회피정책을 적용하고, 교착상태가 생기지 않는 모듈에 대해서는 할당규칙(dispatching rule)을 적용하는 방법을 제시하고 있다.

본 논문에서 제시하고 있는 트랙장비를 위한 교착상태 없는 온라인 스케줄링 방법은 크게 두 부분으로 이루어져 있다. 첫번째 부분은 트랙장비의 프로세스 모듈들을 교착가능집합과 교착없는집합(deadlock-safe set, PSS)으로 구분하는 것이다. PDS는 미래에 교착상태를 유발할 가능성이 있는 모듈들의 집합이며, PSS는 PDS의 여집합으로서 교착상태가 발생하지 않는 모듈들의 집합이다. 두번째 부분은 교착상태를 회피하는 스케줄링 방법을 적용하는 단계로서, PDS에 포함된 모듈들은 교착회피정책이 적용되며 PSS에 포함된 모듈들은 할당규칙이 적용이 된다.

2. 트랙장비의 개요

Fig. 1은 31개의 프로세스 모듈과 4개의 이송로봇으로 이루어져 있는 전형적인 트랙장비의 구조를 보여준다. 프로세스 모듈에는 쿨 플레이트(cool plate), 핫 플레이트(hot plate), 노광 후 핫 플레이트(post exposure bake hot plate), 엣지 노광(edge

exposure), 스핀 코터(spin coater), 스핀 디벨로퍼(spin developer), 저압력 증착(low pressure adhesion) 등이 있다. 트랙장비의 한쪽 끝은 카세트(cassette)들의 입출력을 위한 인덱서(indexer)가 있으며 다른 한쪽 끝은 노광기와 연결되어 있다. 세 개의 로봇은 각 셀 안에서 모듈과 모듈 사이에 웨이퍼를 이송하거나 셀과 셀 사이에 웨이퍼를 이송하는데 사용되며 하나의 로봇은 트랙장비와 노광기 사이에 웨이퍼를 이송하는 역할을 한다. 셀과 셀 사이에는 세 개의 핀으로 이루어진 스테이지(stage)가 있으며 각각의 스테이지는 한번에 하나의 웨이퍼만 저장할 수 있다. 그리고, 트랙장비와 노광기 사이에는 16 행 2 열의 스테이지로 이루어진 스택커(stacker)가 있다.

사진공정에서 Deep-UV 를 이용하는 공정은 TARC(Top Anti-Reflective Coating)와 BARC(Bottom Anti-Reflective Coating)의 두 가지 전형적인 공정으로 분류된다. Table 1 은 세 종류의 웨이퍼들의 프로세스 흐름과 가공시간을 보여준다. 사진공정에서는 공정의 특성상 한 번 거친 프로세스 모듈을 다시 거치지 않기 때문에, 트랙장비에서 단일 종류의 웨이퍼를 가공할 경우에는 교착상태가 발생하지 않지만 여러 종류의 웨이퍼를 가공할 경우에는 교착상태가 발생할 수 있다. Fig. 2 는 다섯 개의 프로세스 모듈에서 가공 중인 다섯 개의 웨이퍼가 현재 교착상태에 빠진 상태를 보여준다. 즉, 다섯 개의 웨이퍼 중 가공을 마친 어떤 한 웨이퍼의 다음 작업에 해당하는 모든 프로세스 모듈이 다른 웨이퍼에 의해 선점되어 있고, 이러한 상태가 서로 순환적으로 맞물려 있는 순환대기상태(circular wait state)가 되어 현재 상태에서 절대로 변할 수 없는 교착상태가 된다. 교착회피정책의 목적은 이러한 순환대기상태가 생기지 않도록 웨이퍼들의 프로세스 흐름을 동적으로 제어하는 것이다.

3. 교착가능집합

2 장에서 언급했듯이 여러 종류의 웨이퍼를 가공하는 트랙장비에서는 교착상태가 발생할 수 있다. 그러나, 트랙장비에서는 일반적으로 일부 프로세스 모듈들에서만 교착상태가 발생하므로 모든 모듈을 대상으로 교착회피정책을 적용하는 것은 교착회피정책을 적용하는데 요구되는 복잡도나 트랙장비의 생산율을 높이는 데 비효율적이다. 따라서, 트랙장비 내에 있는 프로세스 모듈들 중에서 교착상태를 유발할 수 있는 모듈들을 모두 찾아내어 선별적으로 교착회피정책을 적용하는 접근방법

Table 1 Process flows of the three wafer types

Op	TARC #1		TARC #2		BARC	
	Module	PT*	Module	PT	Module	PT
1	INDEXER		INDEXER		INDEXER	
2	HHP1/HHP2	80	HHP1/HHP2	80	HHP1/HHP2	80
3	LPAH1/LPAH2	90	LPAH1/LPAH2	90	LPAH1/LPAH2	90
4	CP2/CP3	60	CP2/CP3	60	CP2/CP3	60
5	SC1/SC2	65	SC1/SC2	65	SC3	50
6	SC3	50	HP1/HP2	90	HP1/HP5	90
7	HP5/HP6	90	CP6/CP7	60	CP1/CP6	60
8	CP10/CP11	60	SC3	50	SC1/SC2	65
9	STEPPER	100	STEPPER	100	HP6/HP7	90
10	PEB1/PEB2	90	PEB1/PEB2	90	CP10/CP11	60
11	CP8/CP9	60	CP8/CP9	60	STEPPER	100
12	EEW1/EEW2	90	EEW1/EEW2	90	PEB1/PEB2	90
13	SD1/SD2/SD3	130	SD1/SD2/SD3	130	CP8/CP9	60
14	HP3/HP4	90	HP3/HP4	90	EEW1/EEW2	90
15	CP4/CP5	60	CP4/CP5	60	SD1/SD2/SD3	130
16	INDEXER		INDEXER		HP3/HP4	90
17					CP4/CP5	60
18					INDEXER	

*PT: Processing time in seconds

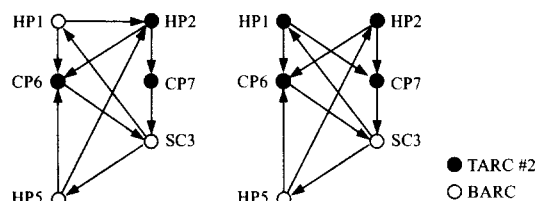


Fig. 2 Examples of deadlocks in the track system

이 필요하다. 이때, 교착상태를 유발할 수 있는 모듈들의 집합은 교착가능집합(potential deadlock set, PDS)이라고 정의하며, 교착상태를 유발하지 않는 모듈들의 집합은 교착없는집합(deadlock-safe set, PSS)이라고 정의한다.

PDS 는 자원요구행렬(resource request matrix)을 이용하여 구할 수 있다. 자원들의 집합을 $R = \{r_1, r_2, \dots, r_{|R|}\}$ 이라 하고 집합 R 에 포함된 원소들의 개수를 $|R|$ 로 나타내면, 자원순서 $h = \langle r_1, r_2, \dots, r_{|R|} \rangle$ 에 대해서 부품 종류 τ 의 행렬 S^τ 을 다음과 같이 정의한다.

$$S^\tau = \begin{matrix} & r_1 & r_2 & \cdots & \cdots & r_{|R|} \\ \begin{matrix} r_1 \\ r_2 \\ \vdots \\ \vdots \\ r_{|R|} \end{matrix} & \begin{bmatrix} s'_{1,1} & s'_{1,2} & \cdots & \cdots & s'_{1,|R|} \\ \vdots & \vdots & & & \vdots \\ s'_{|R|,1} & s'_{|R|,2} & \cdots & \cdots & s'_{|R|,|R|} \end{bmatrix} \end{matrix} \quad (1)$$

$$s'_{i,j} = \begin{cases} 1 & \text{if the part being processed in } r_i \in R \text{ requests } r_j \in R \\ & \text{for the next processing,} \\ 0 & \text{otherwise.} \end{cases}$$

정의 1: 총 T 개의 부품에 대하여 자원요구행렬 (resource request matrix, RRM) S 는 $S = \sum_{r=1}^T S^r$ 으로 정의한다.

자원요구행렬에서 각 원소의 값은 자원순서 h 에 따라 다르게 결정되는데, 이때 자원순서 h 의 변경은 실제 시스템 내 자원순서 변경을 의미하는 것은 아니다. 교착 가능성이 있는 경우에는 자원요구행렬의 좌측 하단 삼각형(lower-triangle)에 있는 원소들 중 적어도 하나는 0 이 아닌 값을 가지고 있음을 Fig. 3 으로부터 알 수 있다. 이와 같이 좌측 하단 삼각형에서 0 이 아닌 값을 갖는 원소는 교착상태의 가능성을 암시하므로 교착가능지표 (potential deadlock indicator, PDI)를 다음과 같이 정의한다.

$$L = \{ \langle i, j \rangle \mid s_{i,j} > 0 \text{ and } i > j, \text{ for } i, j = 1, 2, \dots, |R| \} \quad (2)$$

교착가능지표의 원소 $\langle i, j \rangle \in L$ 를 고려할 때 집합 $D(i, j) = \{ r_k \mid r_k \in R, \text{ for } j \leq k \leq i \}$ 에 속하는 모든 자원 $r \in D(i, j)$ 는 $\langle i, j \rangle$ 와 관련되어 교착상태를 발생할 가능성이 있다. 이와 같이 교착상태를 발생할 수 있는 자원들의 집합을 교착가능집합이라 하며 다음과 같이 정의한다.

정의 2: 교착가능집합 (potential deadlock set, PDS) 은 모든 $(i, j) \in \{ (i, j) \mid \langle i, j \rangle \in L \}$ 에 대하여 $D = \bigcup D(i, j)$ 으로 정의한다.

예제 1: Fig. 4 는 6 대의 기계와 1 대의 로봇으로 이루어져 있는 생산 셀이다. 두 종류의 부품이 생산이 되며 프로세스의 흐름은 각각 JT1: $\{M1, M4\} \rightarrow \{M2, M3\} \rightarrow \{M5\}$ 와 JT2: $\{M5\} \rightarrow \{M6\} \rightarrow \{M2\}$ 이다. 자원순서를 $h1 = \langle r_1, r_2, r_3, r_4, r_5, r_6 \rangle = \langle M1, M2, M3, M4, M5, M6 \rangle$ 라 하면 행렬 S^{JT1} 와 S^{JT2} 는

$$S^{JT1} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad S^{JT2} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

이고, 정의 1 에 의하여 자원요구행렬은

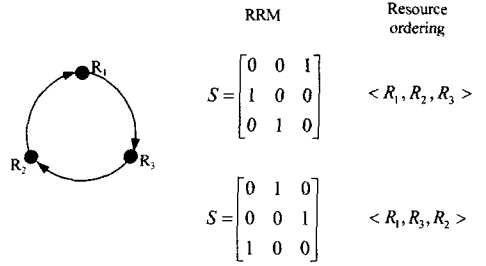


Fig. 3 The RRM of the system with three resources

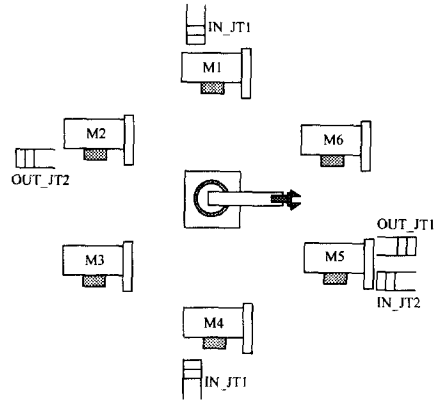


Fig. 4 An example of production cell

$$S = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

가 된다. 이때 PDI 는 $L = \{ \langle 4, 2 \rangle, \langle 4, 3 \rangle, \langle 6, 2 \rangle \}$ 가 되고, $D(4, 2) = \{ r_2, r_3, r_4 \} = \{ M2, M3, M4 \}$, $D(4, 3) = \{ r_3, r_4 \} = \{ M3, M4 \}$, $D(6, 2) = \{ r_2, r_3, r_4, r_5, r_6 \} = \{ M2, M3, M4, M5, M6 \}$ 이므로 PDS 는 $D = \{ M2, M3, M4, M5, M6 \}$ 가 된다.

반면, 자원 순서를 $h2 = \langle r_1, r_2, r_3, r_4, r_5, r_6 \rangle = \langle M1, M4, M3, M2, M5, M6 \rangle$ 로 가정하면 자원요구행렬은

$$S = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

가 되며, PDI 는 $L = \{ \langle 6, 4 \rangle \}$ 이고 $D(6, 4) = \{ r_4, r_5, r_6 \}$

$= \{M2, M5, M6\}$ 이므로 PDS는 $D = \{M2, M5, M6\}$ 가 된다.

위의 예제에서 볼 수 있듯이 자원의 순서는 PDI와 PDS 원소의 개수에 영향을 주므로, PDI와 PDS에 포함된 원소의 개수를 최소화 하도록 자원의 순서를 재정렬하는 것이 필요하다. 따라서, 본 논문에서는 PDI와 PDS 원소의 개수를 목적함수로 하여 최적의 자원 순서를 구한다.

$$\text{첫번째 목적함수: } f_L = |L| \quad (4)$$

$$\text{두번째 목적함수: } f_D = |D| \quad (5)$$

위의 두 개의 목적함수는 유전자 알고리즘을 이용하여 순차적으로 최소화 한다. 먼저 첫번째 목적함수인 PDI 원소 개수의 최소값 f_L^* 을 구한 후, f_L^* 의 값을 갖는 여러 해 중에서 두 번째 목적함수인 PDS 원소 개수의 최소값 f_D^* 을 구한다. 이와 같은 방법으로 구해진 해는 자원들의 순서인 $\langle r_1, r_2, \dots, r_{|R|} \rangle$ 의 형태로 표현되며 이로부터 궁극적으로 관심이 있는 PDS를 얻을 수 있다. 유전자 알고리즘을 적용할 때 유전 연산자로는 OX(order crossover)와 역치(inversion)를 사용한다.⁽²⁰⁾ 예를 들면 예제 1의 경우 해를 표현하기 위한 리스트는 자원순서 $h = \langle M1, M2, M3, M4, M5, M6 \rangle$ 또는 $h = \langle M1, M4, M3, M2, M5, M6 \rangle$ 등의 형태를 갖는다. 자원순서 $h = \langle M1, M2, M3, M4, M5, M6 \rangle$ 인 경우에는 $L = \{ \langle 4, 2 \rangle, \langle 4, 3 \rangle, \langle 6, 2 \rangle \}$ 이고 $D = \{ M2, M3, M4, M5, M6 \}$ 이므로 첫번째와 두 번째 목적함수는 각각 $f_L = |L| = 3$ 과 $f_D = |D| = 5$ 가 된다. 반면, $h = \langle M1, M4, M3, M2, M5, M6 \rangle$ 인 경우에는 $L = \{ \langle 6, 4 \rangle \}$ 이고 $D = \{ M2, M5, M6 \}$ 이므로 첫번째와 두 번째 목적함수는 각각 $f_L = |L| = 1$ 과 $f_D = |D| = 3$ 이 된다. 따라서, 유전자 알고리즘을 적용할 때는 먼저 첫번째 목적함수를 대상으로 최적화 시키므로 첫번째 목적함수의 최소값인 $f_L^* = |L| = 1$ 값을 갖는 자원순서들을 구하고 이들 중에서 두 번째 목적함수인 f_D 의 최소값을 갖는 자원순서 h 를 최종적으로 선택한다.

PSS는 PDS의 여집합이므로 앞에서 찾은 PDS으로부터 쉽게 구할 수 있다.

4. 교착상태를 회피하는 스케줄링

자동화된 트랙장비의 운용에 있어서 신뢰성은 중요한 문제 중 하나이다. 따라서, 트랙장비에서 사용되는 스케줄러는 교착상태가 생기지 않는 것을 보장해야 하며 프로세스 모듈의 고장 같은 예상하지 못한 이벤트에 강건하게 대처할 수 있어야

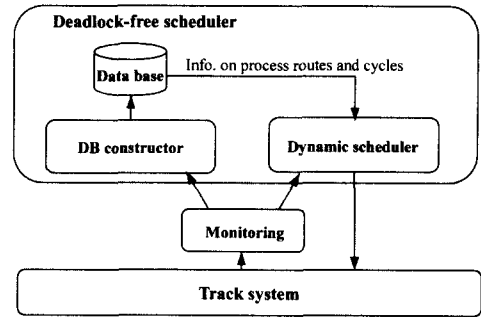


Fig. 5 Architecture of the deadlock-free scheduler

한다. 즉, 트랙장비에서 어떤 프로세스 모듈이 고장이 났을 때 다른 선택 가능한 모듈이 있으면 전체 시스템을 정지 시키지 않고 가공을 계속 진행할 수 있어야 한다. 신뢰성 외에 트랙장비의 생산성을 높이는 것도 트랙장비의 운용에 있어서 중요한 문제이다. 따라서, 트랙장비의 생산성을 높일 수 있는 할당규칙을 선택하여 적용하는 것이 필요하다. 본 논문에서는 부품들의 총생산시간(make-span)을 목적함수로 하여 이를 최소화 하는 할당규칙을 선택할 수 있도록 하였다.

Fig. 5에서 볼 수 있듯이 본 논문에서 제시하는 교착상태 없는 스케줄러는 크게 두 개의 부분으로 나뉜다. 하나의 부분은 DB 구축기(data base constructor)이고 다른 부분은 동적 스케줄러(dynamic scheduler)이다.

DB 구축기와 DB 시스템은 트랙장비를 운용할 때 발생 가능한 교착상태를 온라인 상태에서 찾아내어 피하기 위한 것이다. DB 시스템은 오프라인(off-line)에서 구축되며 각 부품의 종류별로 모든 프로세스의 흐름에 대하여 발생할 수 있는 사이클(cycle)을 찾아내어 사이클에 대한 정보를 가지게 된다. 이때, 각 프로세스 흐름에 대하여 발생할 수 있는 모든 사이클을 효율적으로 찾아내기 위해서 PDI와 백트래킹 탐색(backtracking search)⁽²¹⁾을 사용한다. 이와 같은 사이클을 찾는 과정은 오프라인에서 한 번 수행되며, 일단 DB 시스템에 저장된 사이클에 대한 정보는 온라인에서 동적 스케줄러로 교착상태 없는 스케줄을 생성할 때 교착상태를 피하기 위해 이용된다. 사이클은 다음과 같은 알고리즘으로 구할 수 있다.

Algorithm FC:

begin

SL := [START]; NSL := [START]; CS := START;

while NSL ≠ []

do begin

if CS has no precedent resource or (CS = START

Table 2 Finding cycles for the case of $\langle 6, 4 \rangle \in L$

Iteration	CS	SL	NSL
0	M4	[M4]	[M4]
1	M6	[M6 M4]	[M6 M4]
2	M5	[M5 M6 M4]	[M5 M6 M4]
3	M1	[M1 M5 M6 M4]	[M1 M2 M3 M4 M5 M6 M4]
4	M2	[M2 M5 M6 M4]	[M2 M3 M4 M5 M6 M4]
5	M4	[M4 M2 M5 M6 M4]	[M4 M2 M3 M4 M5 M6 M4]
6	M3	[M3 M5 M6 M4]	[M3 M4 M5 M6 M4]
7	M1	[M1 M3 M5 M6 M4]	[M1 M2 M3 M4 M5 M6 M4]
8	M2	[M2 M3 M5 M6 M4]	[M2 M3 M4 M5 M6 M4]
9	M4	[M4 M2 M3 M5 M6 M4]	[M4 M2 M3 M4 M5 M6 M4]
10	M4	[M4 M5 M6 M4]	[M4 M5 M6 M4]

Table 3 Finding cycles for the case of $\langle 4, 2 \rangle \in L$

Iteration	CS	SL	NSL
0	M2	[M2]	[M2]
1	M4	[M4 M2]	[M4 M2]
2	M1	[M1 M4 M2]	[M1 M2 M6 M4 M2]
3	M2	[M2 M4 M2]	[M2 M6 M4 M2]
4	M6	[M6 M4 M2]	[M6 M4 M2]
5	M1	[M1 M6 M4 M2]	[M1 M2 M3 M4 M6 M4 M2]
6	M2	[M2 M6 M4 M2]	[M2 M3 M4 M6 M4 M2]
7	M3	[M3 M6 M4 M2]	[M3 M4 M6 M4 M2]
8	M1	[M1 M3 M6 M4 M2]	[M1 M2 M3 M4 M6 M4 M2]
9	M2	[M2 M3 M6 M4 M2]	[M2 M3 M4 M6 M4 M2]
10	M4	[M4 M6 M4 M2]	[M4 M6 M4 M2]

```

and the length of SL is not 1)
then begin
  while SL is not empty and CS = the first element
    of SL
  do begin
    remove first element from SL;
    remove first element from NSL;
    CS := first element of NSL;
  end
  add CS to SL;
end
else begin
  place precedent resources of CS on NSL;
  CS := first element of NSL;
  add CS to SL;
end
if CS = START then return (SL);    % SL returns
a cycle
end;
return FAIL;
end
    
```

SL(state list)는 현재 탐색중인 시퀀스의 상태를 나타내며, NSL(new state list)는 추후에 평가될 자원을 포함하고, CS(current state)는 현재 상태를 나타낸다.

예제 2: Fig. 4의 생산 셀에서 세 종류의 부품이 JT1: {M1, M2} → {M5}, JT2: {M1, M2} → {M3, M4} → {M5}, JT3: {M5} → {M6} → {M4} → {M2}의 프로세스 흐름을 갖는다고 가정하고 자원 순서를 $h = \langle M1, M2, M3, M4, M5, M6 \rangle$ 라 하면, PDI와 PDS는 각각 $L = \{ \langle 6, 4 \rangle, \langle 4, 2 \rangle \}$ 와 $D = \{ M2, M3, M4, M5,$

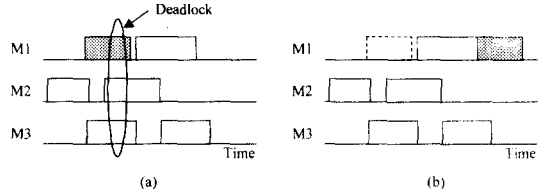


Fig. 6 Generation of a deadlock-free on-line Gantt chart through shifting (a) Current state with deadlock (b) The deadlock state is avoided by shifting

$M6\}$ 이 된다. Table 2, 3은 $\langle 6, 4 \rangle, \langle 4, 2 \rangle \in L$ 의 각각 경우에 대하여 알고리즘 FC의 적용 과정을 보여준다. 표에서 굵은 글씨는 사이클을 의미하며 결과적으로 총 6개의 사이클이 존재함을 알 수 있다.

동적 스케줄러는 온라인에서 트랙장비를 운용하는데 필요한 스케줄을 실시간으로 생성하는 부분으로서, PDS에 포함된 프로세스 모듈에 대해서는 온라인 간트차트(on-line Gantt chart)를 이용하여 실시간으로 교착상태 없는 스케줄을 생성하고 PSS에 포함된 프로세스 모듈에 대해서는 할당규칙만을 적용한다.

PDS에 포함된 프로세스 모듈을 대하여 교착상태 없는 온라인 간트차트는 공정블록의 이동(shifting operation block)을 이용하여 구할 수 있다. 즉, 온라인 간트차트에서 교착상태를 유발하는 공정블록을 교착을 회피하도록 오른쪽으로 이동시키는 작업으로서, 이 때 현재 상태의 교착상태 여부는 DB 구축기에 저장되어 있는 사이클 정보를 이용하여 알 수 있다. 공정블록의 이동은 생산율을 저하시킬 수 있는데, 이러한 단점을 보완하기 위하여 모든 가능한 작업 시퀀스를 고려하여 교착상태 없는 온라인 간트차트를 구성한 후 가장 작은 생산시간을 갖는 간트차트를 선택한다. Fig. 6은 교착상태를 유발하는 공정블록의 이동을 통하여 교착상태 없는 온라인 간트차트를 구성하는 과정을 보여준다.

PDS에 교착상태 없는 스케줄을 생성하는 과정에서 고려해야 할 두 가지 경우가 있는데, 첫번째는 부품이 PDS에 포함된 모듈에 처음으로 할당될 때 교착없는 스케줄을 생성하는 것이고 두 번째는 PDS에 포함된 모듈이 고장이 났을 때 교착없는 스케줄을 재생성하는 것이다.

부품이 PDS에 포함된 프로세스 모듈에 처음으로 할당되는 경우, 동적 스케줄러는 그 부품의 가능한 작업 시퀀스에 대하여 온라인 간트차트를 생성하면서 생산시간을 비교하여 가장 좋은 성능을 보이는 간트차트를 선택한다. 이때, 교착상태가 없는

온라인 간트차트는 공정블록의 이동을 통하여 생성할 수 있다. 즉, 부품이 PDS 에 포함된 모듈에 처음 할당될 때 교착상태 없는 스케줄을 생성하는 과정은 다음과 같다.

단계 1: 투입되는 부품의 가능한 라우트 중 하나를 무작위로 선택한다.

단계 2: 선택된 라우트에 대상으로 공정블록의 이동을 이용하여 교착상태 없는 온라인 간트차트를 구성한다.

단계 3: 생성된 온라인 간트차트의 생산시간을 계산한다.

단계 4: 투입되는 부품의 다른 가능한 라우트가 있으면 단계 1 부터 다시 수행하고, 그렇지 않으면 단계 5를 수행한다.

단계 5: 생성된 간트차트들의 생산시간을 비교하여 최소의 것을 선택한다.

특정 프로세스 모듈이 고장이 난 경우 현재의 온라인 간트차트로부터 교착상태 없는 새로운 스케줄을 재생성하는 과정은 다음과 같다.

단계 1: 고장이 난 프로세스 모듈에 할당된 공정블록 중 가장 앞에 있는 것을 선택한다.

단계 2: 선택된 공정블록에 대하여 가공이 가능한 다른 모듈이 있으면 단계 3 을 수행하고, 그렇지 않으면 단계 4 를 수행한다.

단계 3: 교착상태 없는 온라인 간트차트를 재구성한다.

단계 3.1: 가공 가능한 모듈 중 하나를 무작위로 선택하고 해당 공정블록을 할당한다.

단계 3.2: 공정블록의 이동을 이용하여 교착상태 없는 온라인 간트차트를 구성한다.

단계 3.3: 온라인 간트차트의 생산시간을 계산한다.

단계 3.4: 다른 가공 가능한 모듈이 있으면 단계 3.1 을 수행하고, 그렇지 않으면 3.5 단계를 수행한다.

단계 3.5: 가장 작은 생산시간을 갖는 온라인 간트차트를 선택한다.

단계 4: 해당 공정블록은 모듈이 수리될 때 까지 기다린다.

단계 5: 고장이 난 모듈에 할당된 다른 공정블록이 있으면 단계 1 을 수행하고, 그렇지 않으면 마친다.

고장이 난 모듈에 대하여 온라인 간트차트를 재구성하는 과정은 두 개의 루프로 구성되어 있다. 바깥 루프는 고장이 난 모듈에 할당된 공정블록을 하나씩 선택하는 것으로서, 선택된 공정블록에 대해 가공이 가능한 다른 모듈이 있으면 교착상태 없는 온라인 간트차트를 재구성한다. 안의 루프는

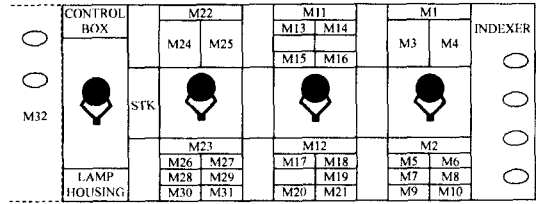


Fig. 7 The example of the track system

Table 4 Process flows of the three wafer types

Op	TARC #1		TARC #2		BARC	
	Module	PT	Module	PT	Module	PT
1	INDEXER		INDEXER		INDEXER	
2	M5/M6	80	M5/M6	80	M5/M6	80
3	M3/M4	90	M3/M4	90	M3/M4	90
4	M9/M10	60	M9/M10	60	M9/M10	60
5	M1/M2	65	M1/M2	65	M12	50
6	M12	50	M7/M8	90	M7/M17	90
7	M17/M18	90	M20/M21	60	M8/M20	60
8	M30/M31	60	M12	50	M1/M2	65
9	M32	100	M32	100	M18/M19	90
10	M26/M28	90	M26/M28	90	M30/M31	60
11	M27/M29	60	M27/M29	60	M32	100
12	M24/M25	90	M24/M25	90	M26/M28	90
13	M11/M22/M23	130	M11/M22/M23	130	M27/M29	60
14	M13/M14	90	M13/M14	90	M24/M25	90
15	M15/M16	60	M15/M16	60	M11/M22/M23	130
16	INDEXER		INDEXER		M13/M14	90
17					M15/M16	60
18					INDEXER	

바깥 루프에서 선택된 공정블록의 이동을 통해 교착상태 없는 온라인 간트차트를 재구성하고 가장 작은 생산시간을 갖는 온라인 간트차트를 선택하는 과정을 의미한다.

5. 시뮬레이션

본 논문에서 제안된 교착상태를 회피하는 스케줄링 방법은 2 장에서 소개한 트랙장비를 대상으로 시뮬레이션을 통하여 평가되었다. Fig. 7 은 Fig. 1 의 트랙장비와 동일한 장비로서 스케줄링 방법 적용시 편의를 위하여 모든 프로세스 모듈과 노광기에 M1 부터 M32 까지 번호를 붙였다. 시뮬레이션에서 사용된 가정들은 다음과 같다. 1) 웨이퍼의 종류별 프로세스 흐름은 Table 4 와 같이 주어진다. 2) 웨이퍼는 카세트 단위로 트랙장비에 투입되는데 하나의 카세트는 25 개의 웨이퍼를 포함하며 각 카세트에는 10 개의 TARC#1, 10 개의 TARC#2, 5 개의 BARC 가 포함되어 있다. 3) 프로세스 모듈의 평균 고장 간격 시간(mean time between failure, MTBF)과 평균 수리 시간(mean time to repair, MTTR)은 각각 500 시간과 2 시간으로 가정하며 지수분포를 따른다고 가정한다. 4) 이송 로봇이 웨이퍼를 모듈에서 모듈로 이송하는데 걸리는 시간

은 10 초로 가정하였으며, 시뮬레이션은 Intel Pentium III (500MHz) PC 에서 수행하였다.

트랙장비에서 PDS 를 찾는 문제는 유전자 알고리즘을 이용하였으며 유전 연산자로는 OX 와 역치를 이용하였다. 유전자 알고리즘을 적용할 때 세 대(generation) 수는 20,000 에서 140,000 까지 20,000 씩 증가시키면서 매 경우에 대하여 5 번씩 시뮬레이션을 수행하여 결과를 비교하였다. Table 5 와 Fig. 8 은 각 세대 수에 대하여 시뮬레이션을 수행했을 때 얻어진 두 가지 목적함수의 최소값의 평균과 표준편차 및 시뮬레이션의 소요된 시간의 평균을 보여준다. 예를 들면 120,000 세대인 경우 f_L^* 는 7.6 이고 f_D^* 는 8.6 이며, 이는 PDI 원소의 개수는 약 7-8 이며 PDS 원소의 개수는 약 8-9 임을 의미한다.

세대 수 140,000 으로 시뮬레이션을 수행했을 때 두 목적함수를 최적화 시키는 자원 순서들 중 $h = \langle M5, M6, M3, M4, M9, M10, M2, M1, M12, M7, M17, M8, M21, M20, M19, M18, M30, M31, M32, M26, M28, M27, M29, M24, M25, M22, M23, M11, M13, M14, M16, M15 \rangle$ 인 경우를 고려해 보면, PDI 는 $L = \{ \langle 8, 1 \rangle, \langle 8, 2 \rangle, \langle 20, 1 \rangle, \langle 20, 2 \rangle, \langle 20, 12 \rangle, \langle 21, 12 \rangle \}$ 이고 PDS 는 $D = \{ M1, M2, M7, M8, M12, M17, M20, M21 \}$ 이다. 따라서, PDS 에 속하는 8 개의 프로세스 모듈에서는 교착상태가 발생할 수 있으므로 교착상태를 회피하는 스케줄링 방법을 적용할 때 온라인 간트차트를 이용한 교착회피정책을 적용하고 나머지 프로세스 모듈이 포함된 PSS 에 대해서는 할당규칙을 적용한다. 할당규칙으로는 EC, ERT, SPT 세 가지를 적용하여 총 생산시간을 비교하였다. 여기서, EC 는 이전 단계의 공정을 가장 먼저 끝낸 웨이퍼에 우선순위를 주는 것이고, ERT 는 트랙장비에 가장 먼저 투입된 웨이퍼에 우선순위를 주는 할당규칙이다. 그리고, SPT 는 가공시간이 가장 짧은 웨이퍼에 우선순위를 주는 할당규칙이다.

Table 6 은 트랙장비에서 250 개의 웨이퍼를 가공한다고 가정했을 때 교착상태를 회피하는 스케줄링 방법의 시뮬레이션 결과로서, 세 가지 할당규칙에 대하여 동일한 조건에서 각각 10 회 시뮬레이션을 수행하여 얻은 결과를 평균하였다. Table 6 은 각 할당규칙별로 총 생산시간과 단위 웨이퍼 생산시간(throughput)의 평균 및 95% 신뢰도 구간을 보여 주는데, 시뮬레이션 결과 할당규칙에 따른 총 생산시간이나 단위시간당 생산량의 편차는 크지 않았으나 EC 가 ERT 나 SPT 보다는 좋은 결과를 보임을 알 수 있다.

Table 5 Simulation result of finding PDS

Num. of generations	$f_L^* = L $	$f_D^* = D $	Computation time (sec)
20000	17.4 ($\sigma = 3.2$)	22.6 ($\sigma = 2.3$)	12.2 ($\sigma = 0.4$)
40000	11.6 ($\sigma = 2.4$)	19.0 ($\sigma = 2.7$)	23.8 ($\sigma = 0.4$)
60000	11.4 ($\sigma = 3.0$)	15.4 ($\sigma = 3.9$)	35.4 ($\sigma = 0.5$)
80000	8.8 ($\sigma = 1.1$)	11.0 ($\sigma = 4.5$)	46.6 ($\sigma = 0.5$)
100000	9.0 ($\sigma = 0.0$)	12.0 ($\sigma = 2.8$)	58.4 ($\sigma = 0.5$)
120000	7.6 ($\sigma = 2.2$)	8.6 ($\sigma = 1.3$)	68.8 ($\sigma = 1.1$)
140000	8.2 ($\sigma = 3.0$)	8.8 ($\sigma = 1.8$)	80.6 ($\sigma = 0.9$)

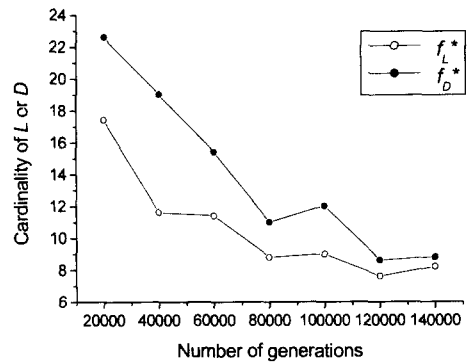


Fig. 8 Graph of the result of Table 5

Table 6 Simulation result of the deadlock-free scheduling

Dispatching rule	Makespan (sec)	Throughput (sec)
EC	33566.0 (335546.9, 335773.1)	134.3 (133.9, 134.7)
ERT	33623.5 (33510.0, 33737.0)	134.5 (134.1, 134.9)
SPT	33820.0 (33706.2, 33933.8)	135.3 (134.9, 135.7)

6. 결론

본 논문에서는 트랙장비를 대상 시스템으로 하여 온라인 상에서 적용이 가능한 교착상태를 회피하는 스케줄링 방법을 제시하였다. 먼저 트랙장비에 있는 자원들의 순서를 바꾸면서 교착상태를 유발할 수 있는 프로세스 모듈들의 집합인 PDS 에 포함된 모듈의 수를 최소화 하는 과정이 필요하며, 최적화 방법으로는 유전자 알고리즘을 이용하였다. 시뮬레이션 결과는 자원 요구 행렬을 이용한 접근 방법이 트랙장비 내의 프로세스 모듈들을 PDS 와 PSS 로 나누는데 효과적임을 보여주었다. 본 논문에서 제안된 교착상태를 회피하는 스케줄링 방법에서 PDS 에는 온라인 간트차트를 이용한 교착회

피정책을 적용하고 PSS에는 할당규칙을 적용하는데, 이와 같은 방법은 교착회피정책의 대상이 되는 프로세스 모듈의 수를 줄여 주어 교착회피정책을 적용하는데 요구되는 계산의 복잡도를 현저히 줄여주는 역할을 한다. 시뮬레이션 결과로부터, 제안된 교착상태를 회피하는 스케줄링 방법이 트랙장비에 효율적으로 적용되어, 결과적으로 트랙장비의 신뢰도를 높이고 생산성을 향상시키는데 효과적임을 보였다.

제안된 교착상태를 회피하는 스케줄링 방법은 반도체 뿐만 아니라 평판 디스플레이(flat panel display)와 디스크 드라이브 헤드(disc-drive head) 등의 제조 시스템에서 사용되는 트랙장비에도 적용하는 것이 가능하다. PDS와 PSS를 구분하여 교착회피정책을 선별적으로 적용하는 방법은 상대적으로 PDS에 비해서 PSS에 포함된 모듈이나 자원의 수가 적은 생산 시스템에는 적용하기에 비효율적일 수 있으나, 일반적인 반도체 공정에서와 같이 일부의 모듈에서만 교착상태가 발생하는 제조 시스템에서는 제안된 방법이 매우 효율적으로 적용될 수 있다.

참고문헌

- (1) Biché, M., 1995, "Trends in Track System Architecture," *Solid State Technology*, pp. 83~86.
- (2) Viswanadham, N. Narahari, Y. and Johnson, T. L., 1990, "Deadlock Prevention and Deadlock Avoidance in Flexible Manufacturing Systems Using Petri Net Models," *IEEE Trans. Robotics and Automat.*, Vol. 6, No. 6, pp. 713~23.
- (3) Banaszak, Z. A. and Krogh, B. H., 1990, "Deadlock Avoidance in Flexible Manufacturing Systems with Concurrently Competing Process Flows," *IEEE Trans. Robotics and Automat.*, Vol. 6, No. 6, pp. 724~734.
- (4) Ezpeleta, J. Colom, J. M. and Martinez, J., 1995, "A Petri Net Based Deadlock Prevention Policy for Flexible Manufacturing Systems," *IEEE Trans. Robotics and Automat.*, Vol. 11, No. 2, pp. 173~184.
- (5) Ezpeleta, J. and Colom, J. M., 1997, "Automatic Synthesis of Colored Petri Nets for the Control of FMS," *IEEE Trans. Robotics and Automat.*, Vol. 13, No. 3, pp. 327~337.
- (6) Jeng, M. D. Chiou, W. D. and Wen, Y. L., 1998, "Deadlock-Free Scheduling of Flexible Manufacturing Systems Based on Heuristic Search and Petri Net Structures," in *Proc. IEEE Int. Conf. on Syst., Man, Cybern.*, San Diego, California, pp. 26~31.
- (7) Xiong, H. H. and Zhou, M. C., 1997, "Deadlock-Free Scheduling of an Automated Manufacturing System Based on Petri Nets," in *Proc. IEEE Int. Conf. on Robotics and Automat.*, Albuquerque, New Mexico, pp. 945~950.
- (8) Damasceno, B. C. and Xie, X., 1999, "Petri Nets and Deadlock-Free Scheduling of Multiple-Resource Operations," in *Proc. IEEE Int. Conf. on Syst., Man, Cybern.*, Tokyo, Japan, pp. 878~883.
- (9) Fanti, M. P. Maione, B. Mascolo, S. and Turchiano, B., 1997, "Event-Based Feedback Control for Deadlock Avoidance in Flexible Production Systems," *IEEE Trans. Robotics and Automat.*, Vol. 13, No. 3, pp. 347~363.
- (10) Wysk, R. A. Yang, N. S. and Joshi, S., 1991, "Detection of Deadlocks in Flexible Manufacturing Cell," *IEEE Trans. Robotics and Automat.*, Vol. 7, No. 6, pp. 853~859.
- (11) Lawley, M. A., 1999, "Deadlock Avoidance for Production System with Flexible Routing," *IEEE Trans. Robotics and Automat.*, Vol. 15, No. 3, pp. 497~509.
- (12) Reveliotis, S. A., 1999, "Accommodating FMS Operational Contingencies Through Routing Flexibility," *IEEE Trans. Robotics and Automat.*, Vol. 15, No. 1, pp. 3~19.
- (13) Perkinson, T. L. McLarty, P. K. Gyurcsik, R. S. and Cavin III, R. K., 1994, "Single-Wafer Cluster Tool Performance: An Analysis of Throughput," *IEEE Trans. Semiconduct. Manufact.*, Vol. 7, No. 3, pp. 369~373.
- (14) Perkinson, T. L. Gyurcsik, R. S. McLarty, P. K., 1996, "Single-Wafer Cluster Tool Performance: An Analysis of the Effects of Redundant Chambers and Revisitation Sequences on Throughput," *IEEE Trans. Semiconduct. Manufact.*, Vol. 9, No. 3, pp. 384~400.
- (15) Venkatesh, S. Davenport, R. Foxhoven, P. and Nulman, J., 1997, "A Steady-State Throughput Analysis of Cluster Tools: Dual-Blade Versus Single-Blade Robots," *IEEE Trans. Semiconduct. Manufact.*, Vol. 10, No. 4, pp. 418~424.
- (16) Srinivasan, R. S., 1998, "Modeling and Performance Analysis of Cluster Tools Using Petri Nets," *IEEE Trans. Semiconduct. Manufact.*, Vol. 11, No. 3, pp. 394~403.
- (17) Bodner, D. A., 1996, "Real-Time Control Approaches to Deadlock Management in Automated Manufacturing Systems," Ph.D. Dissertation, Georgia Institute of Technology, Atlanta, GA.
- (18) Lawley, M. A. Reveliotis, S. A. and Ferreira, P. M., 1998, "A Correct and Scalable Deadlock Avoidance Policy for Flexible Manufacturing Systems," *IEEE Trans. Robotics and Automat.*, Vol. 14, No. 5, pp. 796~809.
- (19) Zant, P. V., 1997, *Microchip Fabrication: A Practical Guide to Semiconductor Processing*. New York: McGraw-Hill.
- (20) Goldberg, D. E., 1989, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley.
- (21) Luger G. F. and Stubblefield, W. A., 1993, *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*. Redwood City, California: Benjamin/Cummings Publishing Company.