

300mm 웨이퍼용 Spinner MMI 를 위한 동적 레이아웃 설계

윤영호, 한광록, 손석원
호서대학교 컴퓨터공학과

e-mail : saw05@netsgo.com , krhan@office.hoseo.ac.kr

Dynamic Layout Design of Spinner MMI for 300mm Wafer

Young-Ho Yoon, Kwang-Rok Han, Seok-Won Sohn
Dept. of Computer Engineering, Hoseo University

요약

반도체 제조 공정의 효율적인 제어와 감시를 위한 모니터링 시스템을 구성하기 위해서는 장비가 가진 특성과 시스템 환경에서 요구되는 조건들과 동작 상태 등을 사용자 인터페이스를 통하여 한 눈에 감시할 수 있는 화면 레이아웃이 요구된다. 본 논문에서는 차세대 반도체 장비인 300mm 웨이퍼 가공용 Spinner 의 MMI 개발을 목표로 하여 장비의 구성 요소들의 선택적 접속에 따라서 사용자 인터페이스용 레이아웃을 능동적으로 설계하는 방법에 대하여 기술하였다. 장비를 구성하는 기본 요소들간의 관계를 정의하고, 장비의 사양과 웨이퍼의 가공 목적에 따라 구성 요소들을 자동으로 유연성있게 배치 하도록 하였다.

1. 서론

정보화 산업의 급속한 발전에 의해서 반도체 제조 공정의 대부분이 자동화가 이루어지고 있고, 아울러 공정을 효율적으로 관리하고 유지하기 위한 방법에 대하여 연구가 활발하게 진행되고 있다. 또한, 반도체 제조 장비의 빠르고 정확한 제어를 가능하게 하는 모니터링 시스템과 데이터 처리 기술의 고도화로 반도체 생산비용의 절감, 완제품의 신뢰성, 공정 수행의 안정성 등을 만족할 수 있게 되었다[1]. 반도체 제조 장비에서 제어 장치는 공정을 수행하는 장치에 대한 제어는 가능하지만 전체적인 상태나 공정의 흐름을 파악하는 것에는 한계가 있기 때문에 MMI(Man Machine Interface)를 사용한다. 일반적으로 장비 모니터링 시스템의 목적은 장비에 사용되는 제어 장치의 동작 상태를 감시하고 공정을 제어하여 생산성 향상과 치명적인 오류를 간접 제어함으로써 손실을 최소화하는데 있다[1]. 이를 위해 시스템의 안정성이 뒷받침 되도록 효율적인 데이터 관리, 공정 제어, 오류 발생을 신속히 MMI에서 알 수

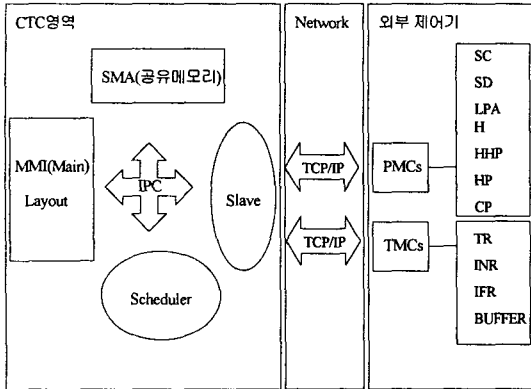
있어야 한다. 기존의 200mm용 Spinner에서 사용되었던 MMI는 화면 구성이 정적으로 이루어져 있기때문에 전문적인 오퍼레이터가 아니면 장비의 상태를 효율적으로 감시하기가 어려웠다.

따라서 본 논문은 기존의 200mm Spinner 의 단점을 보완하고 새로 추가되는 제어 모델을 수용할 수 있으면 다양한 장비 선택 사양에 능동적으로 대처할 수 있는 동적인 레이아웃 모델을 개발함으로써 사용자가 보다 편리하고 능동적으로 감시할 수 있는 MMI 를 설계하도록 한다[5]. 본 연구는 과학재단의 RRC 사업 중 반도체 장비국산화 연구 센터 과제[3][6]로 300 mm 웨이퍼용 Spinner 를 K-12 Spinner 라고 칭한다. 따라서 본 논문에서도 300mm Spinner 를 K-12 Spinner 라는 용어로 사용하기로 한다

본 논문의 구성은 2 장에서 전체 시스템의 구조[7]에 대한 소개와 3 장에서는 사용자 인터페이스에서 사용되는 유닛의 구조와 기능을 설명한다 4 장에서는 동적 레이아웃의 설계 및 구현에 대하여 설명

하고 5 장에서는 결론을 내린다.

2. 전체 시스템 구조



(그림 1) K-12의 전체 구성도

K-12 Spinner의 구성도는 (그림1)과 같다. (그림1)에 나타난 바와 같이 전체 구성은 크게 CTC, PMC, TMC로 나눌 수 있으며 각 부분의 기능은 다음과 같다.

- CTC : 메인 컴퓨터로 K-12의 전체를 관리하며, 하위 제어기들을 관리하여 공정을 수행하도록 구성한다. 내부적으로 사용자인터페이스(UI), 공유 메모리(SMA), 슬레이브(Slave), 스케줄러(Scheduler)로 이루어진다.

- PMC : 모든 프로세스 유닛들을 직접 제어하고 필요할 때 정보를 CTC에 전송하여 전송된 상태를 기준으로 동작되고 제어된다.

- TMC : 로봇제어기로 주로 웨이퍼 이송을 담당하고 Real Time OS의 환경에서 동작하고 내부적으로 구현된 로봇언어를 이용하여 로봇의 이동을 제어한다.

(그림 1)에서 CTC, 외부제어기 간의 통신은 TCP/IP 를 이용하여 이루어진다[4]. 따라서 모든 외부제어기에서 일어나는 상황을 사용자 인터페이스의 레이아웃에 상세히 표시하고 공유메모리나 스케줄러에서 외부제어기로 어떤 신호가 전송되었는지를 감시해야 한다.

(그림1)의 구성도에서 PMC나 TMC에 웨이퍼 가공 목적에 따라 외부 제어 장치들이 선택적으로

결합되는데 이때 결합되는 각 유닛(Unit)의 내용은 다음과 같다.

Module : TR 또는 로봇의 반경을 기준으로 접근 가능한 모든 위치 단위

Unit : 웨이퍼가 1매 이상 보관이 가능한 단위.

Transfer Unit

TR : Transfer Robot으로 웨이퍼를 움직일 수 있는 유닛

INDEXER : TR에서 받은 웨이퍼를 FOUP나 BUFFER로 움직일 수 있는 유닛

IFC : Interface Robot으로 버퍼나 Stepper로 움직일 수 있는 유닛

BUFFER : Up/Down Buffer으로 웨이퍼를 1층 2층으로 움직여 주는 유닛

Process Unit

SC(W) : Spin Coater로 웨이퍼에 막을 생성

SD(W) : Spin Developer로 웨이퍼에 현상

EEW : Edge Exposure Width로 현상 후 가장 자리에 있을 수 있는 이 물질을 제거

LPAH : Low Pressure Adhesion Hot Plate Baker로 코팅하고 나온 막을 강화하기 위하여 HMDS가스를 주입하는 유닛

HHP : Higher Hot Plate Baker로 고열의 온도로 웨이퍼를 굽는 유닛 (HP보다 높은 열)

HP : Hot Plate Baker로 높은 온도로 웨이퍼를 굽는 유닛

CP : Cool Plate Baker 냉각기로써 HP에서 나온 뜨거운Unit 식혀주는 유닛

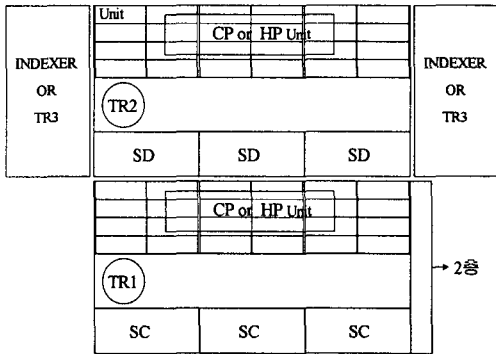
PEB : HP 와 CP 가 하나로 뭉쳐있는 유닛

3. 사용자 인터페이스와 웨이퍼의 흐름

3.1 K-12 Spinner의 기본 레이아웃

K-12 Spinner는 작업 공정이나 웨이퍼 가공 목적에 따라 외부 제어 장치들을 선택적으로 결합하여 운영을 할 수 있다. 따라서 이와 같은 선택적인 장비운영에 능동적으로 대처할 수 있는 동적이니 화면 레이아웃이 요구된다. 전체적인 Spinner의 레이아웃은 Coater와 Developer의 구성

형식에 따라 달라지게 된다. (그림2)는 K-12 Spinner 3.2 웨이퍼의 흐름도 중에서 일반적으로 많이 사용되도록 설계된 3C-3D(3 Coater - 3 Developer)의 레이아웃을 나타낸다.



(그림 2) 3C/3D의 레이아웃 구성도

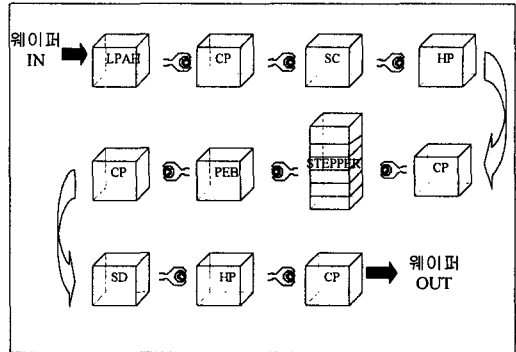
일반적으로 K-12 Spinner 장비는 1 층과 2 층으로 나누어 지며, (그림 2)에 표시 되어 있는 2 층 부분을 제거하면 단층 K-12 Spinner 가 된다.

(표 1) Spinner 에 따른 유닛 총 개수

타입	Unit Name			개수
	SD	SC	Unit (CP or HP)	
1C-1D	1	1	8 / 8 (2 층)	
1C-2D	2	1	16 / 32 (2 층)	
2C-2D	2	2	16 / 32 (2 층)	
2C-3D	3	2	24 / 48 (2 층)	
3C-3D	3	3	24 / 48 (2 층)	
3C-4D	4	3	36 / 72 (2 층)	
4C-4D	4	4	36 / 72 (2 층)	
4C-5D	5	4	40 / 80 (2 층)	
5C-5D	5	5	40 / 80 (2 층)	

(표 1)과 같이 유닛의 수가 타입에 따라서 결정 지워지고 INDEXER 나 TR3 는 타입에 상관없이 (그림 2)와 같이 한 개씩 존재 해야되고, 위치는 어느 곳이든 가능 해야된다

3.2 웨이퍼의 흐름도

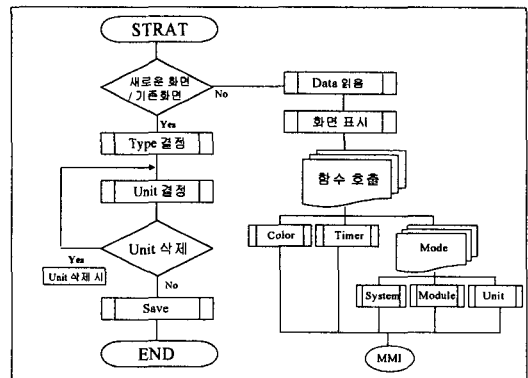


(그림 3) 웨이퍼의 흐름도

한 개의 웨이퍼가 들어오면 (그림 3)과 같은 공정을 거치게 된다. 그리고 중간에 유닛과 유닛 사이를 그 상태에 따라서 로봇이 움직이게 된다.

4. 동적 레이아웃 설계 및 구현

4.1 레이아웃 설계



(그림 4) 레이아웃 생성 과정

(그림 4)에서 보듯이 새로운 화면을 설계 시 타입이 결정 지어지며 (표 1)중에서 하나의 타입이 결정되고 유닛의 종류가 결정 지워진다. 만약 유닛 타입이 잘못 들어오면 삭제하여 다시 유닛 타입을 결정 지을 수가 있다. 모든 유닛이 설정되고 저장 하게 되면 모든 정보가 파일로 저장된다.

기존의 파일을 선택하면 기존의 파일을 읽은 다음 그 정보를 바탕으로 레이아웃 화면 설계가 이루어진다. 그리고 MMI에서 들어오는 정보를 화면에 실시간으로 표시해준다.

4.2 레이아웃 구현

레이아웃 구현 시 가장 핵심이 되는 함수들은 다음과 같다.

```
SetLayoutMode( mMode)
  If mMode = 0 Then
  Elseif mMode = 1 Then
End Sub

Select Case LayoutType
  Case F(1-5)C(1-5)D
    Unit_(1-5)C(1-5)DLayoutType,
      Spin_Unit_Type, Index
End Select
```

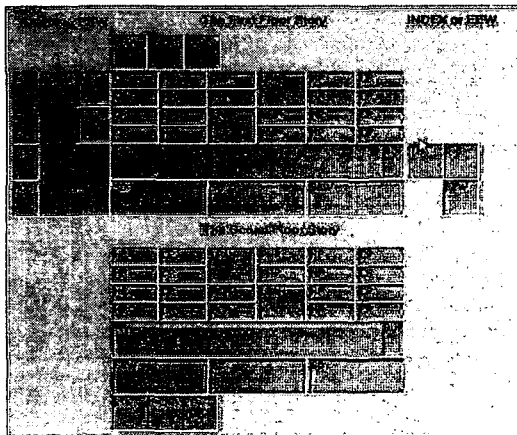
```
Set_Mode(1-5)C(1-5)D(LR_type, Lay_Type)
Set_Timer(Unit_No_Index, Time_No_Case)
Set_Color(Unit_No_Index, Color_No_Case)
```

SetLayoutMode 은 타입을 결정짓는 함수이다. 만약 mMode = 0 이면 새로운 레이아웃 작성이고, mMode = 1 이면 기존의 레이아웃을 불러드린다. Select Case LayoutType 은 유닛을 추가 시에 사용되는 함수인데, LayoutType 은 1CID ~ 5C5D 중 하나를 결정짓고, Spin_Unit_Type 은 유닛 타입, Index 은 위치를 나타낸다.

Set_Mode 은 기존의 레이아웃을 사용 시 레이아웃 타입과, 유닛 타입의 값을 파일로부터 읽어 드린 다음 화면에 표시를 한다.

Set_Timer 은 런타임시 MMI 에서 넘어온 함수 값을 가지고 특정 유닛에 Timer 기능을 작동시킨다.

Set_Color 은 런타임시 MMI 에서 넘어온 함수 값을 가지고 특정 유닛에 Timer 기능을 작동시킨다.



(그림 5) 실제 3C-3D 의 레이아웃 화면

따라서 위와 같은 과정을 통해서 레이아웃이 생성되고 MMI와 결합하여 그 기능을 발휘하게 된다. (그림 5)는 실제 레이아웃 상에서 적용되는 3C-3D 화면이다. (표 1)에 나타난 타입의 장비들을 동적으로 레이아웃을 생성하여 사용자가 시각적으로 보기 편리하고 사용하기가 용이하도록 구현하였다.

5. 결론

반도체 제조장비의 국산화 사업의 일환으로 제작된 본 프로그램은 내부적으로 슬레이브, 공유 메모리, 사용자인터페이스, 스케줄러에서 들어오는 신호를 화면에 나타내고 기존의 정적으로 구성된 레이아웃의 단점을 다양한 장비 선택 사양에 능동적으로 대처할 수 있는 동적인 레이아웃 모델을 설계함으로써 사용자가 보다 편리하고 능동적으로 감시할 수 있는 MMI를 구현하였다.

참 고 문 헌

- [1] 강호석, “반도체 제조 장비를 위한 모니터링 시스템의 기본 요소에 관한 연구”, 정보처리학회 논문집, 제7권 1호., 2000.
- [2] 임성락, 한광록, 최용엽, “Wet Station 장비를 제어하기 위한 모니터링 시스템의 설계”, 한국정보처리학회 논문지, 제6권 5호, pp.1385-1392, 1999.
- [3] 한광록, 황미자, “Wet Station 모니터링을 위한 제어 모듈의 설계 및 구현” 한국정보처리학회 논문지, 제5권 7호, pp.1814-1815, 1998.
- [4] Jerry Secrest, Jack Ghiselli, “SECS comm-unications Handbook”, Secrest Reserch Portola Valley and GW Associates, 1989.
- [5] Richard Simson, “Win32 Programming API Bible”, Waite Group Press, 1996.
- [6] Mitsubishi Electric Co., “Programmable Controller Melsec-A Manual”, 1196
- [7] Silberschatz, Galvin, “Operating System Concepts”, Addison Wesley, 1994.