

# 웨이퍼 오류 패턴 인식 시뮬레이션

김상진\* · 한영신\* · 이철기\*

## Wafer Fail Pattern Classification Simulation

Sang Jin Kim\* · Young Shin Han\*\* · Chill Gee Lee\*\*\*

### Abstract

Semiconductor Manufacturing has emerged as one of the most important world industries. Even with the highly automated and precisely monitored facilities used to process the complex manufacturing steps in a near particle free environment, processing variations in wafer fabrication still exist. The causes of these variations may arise from equipment malfunctions, delicate and difficult processing steps, or human mistakes. In this paper, we could specify the cause stage and the cause equipment and take countermeasures at a speed by the conventional method, without depending on the experience and skills of the engineer

## 1. 서론

수율은 한마디로 “불량률의 반대”라고 할 수 있다. 반도체 생산과정에서의 수율은 간단히 말해서 처음 input된 웨이퍼의 양 대비 output된 웨이퍼의 양이라고 할 수 있다. 여기서 output 이라고 할 수 있는 것은 사용 가능한 웨이퍼를 말한다. 사용 가능한 웨이퍼란 Die의 수가 일정한 웨이퍼에서 bad die의 수가 일정수준을 넘지 않은 것을 말한다. 반도체 생산과정에서의 수율은 상당히 중요한데 그것이 왜냐하면 웨이퍼의 특성상 잘못된 부분은 고치거나 부분적인 교체가 어렵기 때문이다.

그리고 거대한 반도체 산업의 특성상 웨이퍼 한 장당의 1%의 수율의 향상이라도 엄청난 경제적인 수익을 올릴 수 있다. 여기서 EDS검사의 데이터를 토대로 웨이퍼의 수율을 높일 수 있는 검사가 진행

되는데 웨이퍼의 죽은 유형을 판독하여 그 유형에 맞는 생산 공정에서의 불량 원인을 찾아내어 그 원인을 제거함으로써 수율을 높일 수 있는 것이다.

현재는 이 오류패턴인식이 전문적인 불량 분석 팀에 의해서 이루어지고 있는데 이 작업을 오류패턴 인식기를 통해 좀더 빠르고 정확한 오류패턴 인식으로 얼마나 많은 수율증가의 효과를 낼 수 있는가를 시뮬레이션을 통해 알아낼 수 있다.

## 2. 시스템 설계

### 2.1 Assumption

본 논문에서는 다음과 같은 가정을 기준으로 하여 실험 및 결과를 도출한다.

- 1) Interarrival time은 각각의 상황에 다르게 부여한다.
- 2) 사람과 Machine은 절대 Down되지 않는다.
- 3) lot 단위 공정의 구현은 생략하고, Wafer 단위 공정으로 구현한다.
- 4) feedback에 의해 前단계 공정의 문제점을 해

본 연구는 한국과학재단 목적기초연구 (R01-2000-00250)지원으로 수행되었음.

\* 성균관대학교 정보통신공학부 컴퓨터공학과

- 결하여 수율의 증대를 가져온다. 한 가지 Pattern에 의한 feedback으로 인하여 0.1/276의 수율증대가 있다고 가정한다.
- 5) 새로운 패턴의 오류가 발견되었을 경우 0.5/276의 수율 감소가 있다고 가정한다.
  - 6) 현재 200개 이상의 오류패턴이 존재하지만, 10개의 패턴에 대한 검출로 가정한다.
  - 7) Interarrival Time은 다음과 같이 정의한다.
    - 사람 : 30초, 60초, 90초, 120초 4가지 경우
    - Machine : 30초, 60초, 90초, 120초 4가지 경우
    - 단, 사람과 Machine의 interarrival time은 서로 다르게 하여 비교 할 수도 있다.
  - 8) 오류 패턴 검출 시간은 다음과 같이 정의한다.
    - 사람 : 60초
    - Machine : 0.1초
  - 9) 같은 패턴이 25이상 모이면 그 패턴에 대한 정보의Feedback이 이루어진다.
  - 10) Feedback에 의해 前단계 공정이 최적화 되면, 0.1/276의 수율 증대를 가져 온다
  - 11) 사람의 경우 최대 94%까지의 수율 증대를 기대할 수 있다.
  - 12) Machine의 경우 최대 96%까지의 수율 증대를 기대할 수 있다.
  - 13) 서로 다른 패턴의 정보에 대한 Feedback에 의한 것이라도 수율상승은 똑같다.

### 2.2 Fail Pattern 알고리즘

Fail Pattern을 검출하는 과정은 다음과 같다.

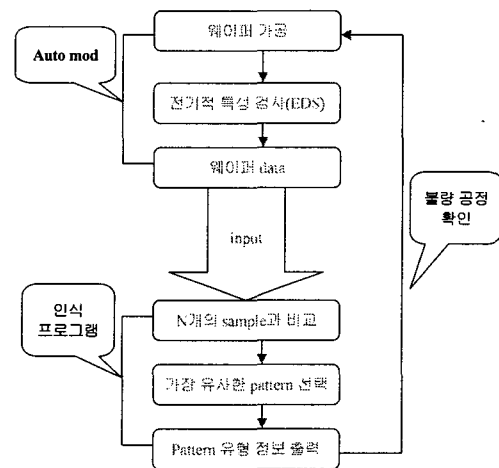
- 1) N개의 Sample Pattern을 생성하고 저장한다.
- 2) 임의의 사용자가 정의하는 오류패턴을 인식한다.
- 3) 각각의 Pattern을 읽어오는 Subroutine을 활용한다.
- 4) Correlation을 적용하여 Pattern을 분석/분류 한다.
- 5) 정확도/수행시간을 비교하여 알고리즘을 수정한다.

### 2.3 System Design

#### 1) 전체 흐름도

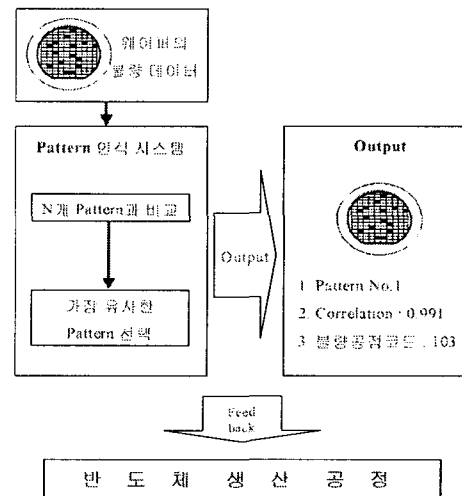
시스템은 크게 오류 패턴의 시뮬레이션을 담

당하는 AutoMod 부분과 패턴 인식 알고리즘을 이용하여, 오류를 찾아내는 인식 프로그램으로 나뉜다. 단 인식 프로그램은 Visual C++을 이용한다.전체적인 시뮬레이션의 visual 한 표현과 웨이퍼오류패턴인식의 input 패턴생성을 automod로 구현한 다음 이로서 생성된 웨이퍼의 오류 데이터를 기반으로하여 인식프로그램에 연동시킴으로서 웨이퍼 오류데이터의 random한 생성과 이의 패턴인식, 그리고 자동화를 했을 때의 수율 상승효과를 알아볼 수 있도록 하였다.



<그림 2.1. System Design>

### 2) 오류 인식 시스템의 기본 구조



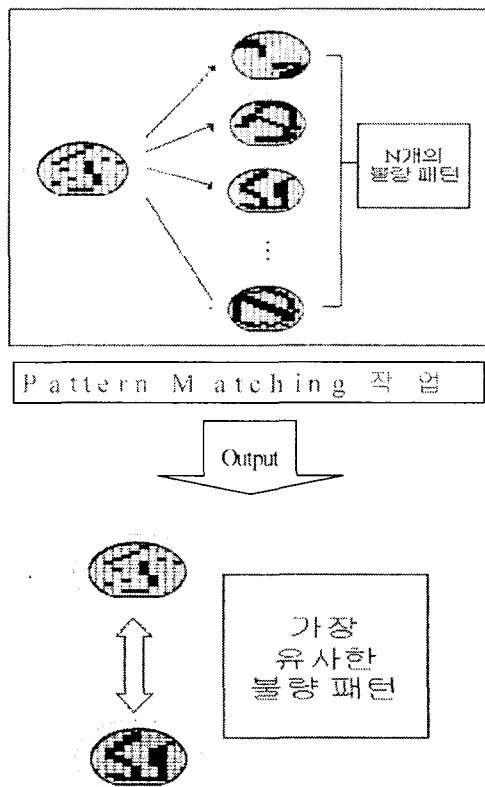
<그림 2.2. Fail Pattern 인식>

웨이퍼의 모양을 비주얼하게 표현하여 각각의 Die마다의 죽은 형태의 정보를 5가지의 색으로 표현하였다. 죽은 Die의 경우 그 5가지의 색으로 표현되고 정상적인 Die의 경우 흰색으로 나타난다.

정확한 오류패턴의 형태를 정확히 알 수가 없어 임의로 잘 나타날 수 있는 200개의 sample 오류패턴을 생성해 놓았다.

### 3) Pattern Matching 구조

Input으로서 웨이퍼 오류데이터를 담은 파일과 사용자가 임의의 마우스 클릭으로서 만든 웨이퍼 오류데이터가 있으며 그렇게 받은 데이터와 sample 오류패턴 200개와의 correlation을 구하여 가장높은 값을 나타낸 sample로서 가장 근접한 오류패턴을 찾아낸다.



<그림 2.3. Pattern Matching>

## 4. 오류 패턴 인식 프로그램 구현

웨이퍼 오류 패턴 인식 프로그램의 구현에는 Visual한 부분을 담당하는 AutoMod와 인식 알고리즘을 이용하여, 오류패턴을 분석하는 Visual C++ 부분으로 크게 나눌 수 있다. 또한 AutoMod에서는 사람이 일할 경우와 Machine이 일하는 두가지의 Sub Menu로 나뉘어 진다.

### 4.1 시스템 환경

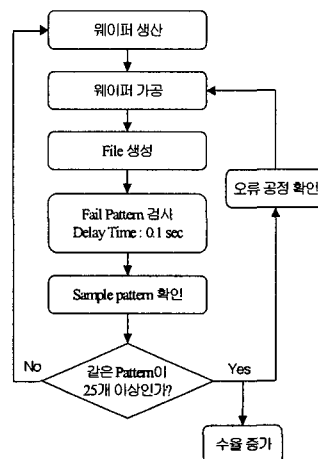
본 논문의 시스템 환경은 다음과 같다.

<표 4.1. 시스템 환경>

항 목	내 용
OS	MS Windows 2000 Server
CPU	Pantium4 2.0 GHz
RAM	512M
Graphic Card	GeForce4 계열
개발 Tool	MS Visual Studio 6.0 Ent. AutoMod 9.0
3D API	OpenGL

### 4.2 Automod machine part 흐름도

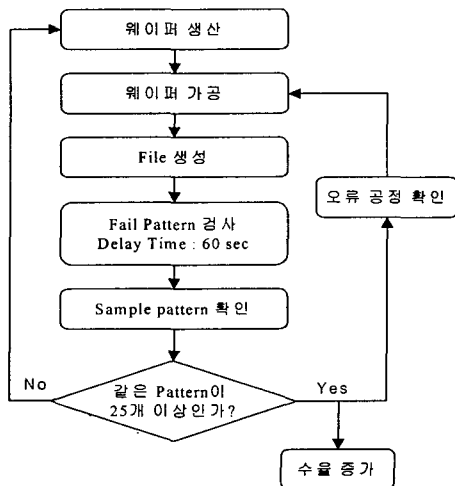
아래의 흐름도는 AutoMod를 이용하여 구현할 경우 Machine이 구동되는 상태를 나타낸다.



<그림 4.1. Machine Operation>

### 4.3 Automod 사람 part 흐름도

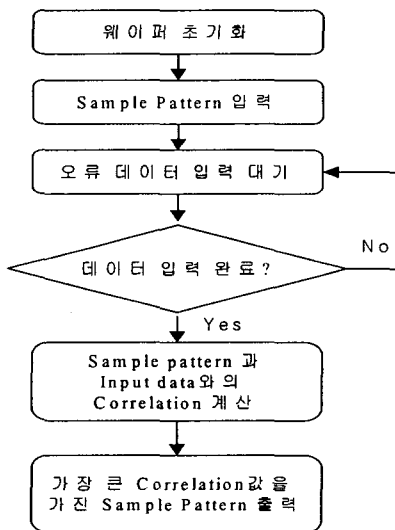
위에서 Machin을 이용한 경우와 비교하면, 모든 것이 동일하지만, Fail pattern을 검사하는 것이 사람에 의해 이루어 진다는 것과, 그 시간의 차이가 가장 크게 나타난다.



<그림 4.2. Human Operation>

### 4.4 Visual C++에서 인식 프로그램의 흐름도

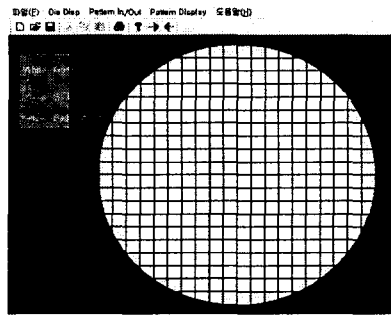
아래의 흐름도는 Machine이 오류패턴을 인식하는 과정을 나타낸다.



<그림 4.3. 오류 인식 알고리즘>

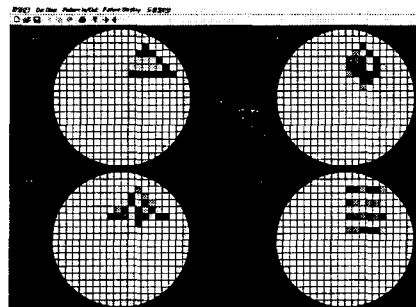
### 4.5 Pattern 인식의 구현

(1) 아래의 그림은 아무것도 실행하지 않은 초기 상태를 나타낸다.



<그림 4.4. 초기 실행 화면>

(2) 아래의 그림은 현재 입력되어있는 Sample pattern을 출력결과 나타낸다.



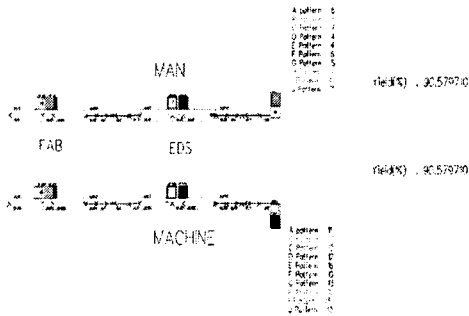
<그림 4.5. 패턴 인식의 예>

### 4.6 인식 프로그램의 적용

FAB(웨이퍼 가공) 부분부터 시작하여 target 시스템이 되는 EDS 공정 후 사람, 인식기를 통한 과정을 시뮬레이션 하는데 FAB 부분은 선처리된 것으로 가정되어 있으므로 간략하게 표현하였다.

그 다음으로 EDS 부분에서 나온 웨이퍼의 불량데이터를 Automod 밖으로 파일로 생성하게 되고 다시 automod 안에선 사람과 인식기가 각각 오류패턴검사를 한 결과로 우선적으로 10개의 sample오류패턴의 유형으로 검출하게 된다.

같은 패턴이 25개 이상 검출되었을 때 그 패턴의 공정에 이상이 생겼다고 판단, 이의 정보를 Feedback 시킴으로서 수율상승이 이루어진다. Feedback의 경우 정보가 매체이므로 이의 시각적인 구현은 생략하고 수율상승의 수치를 표현해 놓았다.



<그림 4.6. AutoMod를 이용한 시뮬레이션>

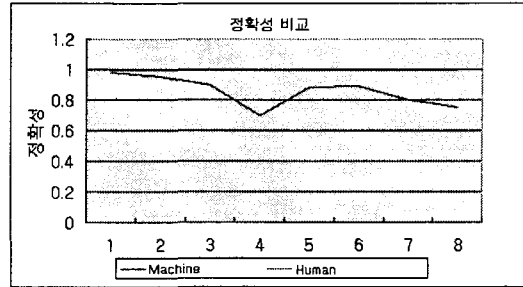
#### 4.7 실험 결과

아래의 표는 Wafer의 Interarrival Time에 따른 수율변화를 나타낸 것으로, 시간이 흐름에 따라 수율이 증가하는 것은 동일하나, Machine의 경우 Lot의 Interarrival Time에 영향을 더 많이 받는 것을 볼 수 있다.

<표 4.2. Interarrival Time에 따른 수율 변화>

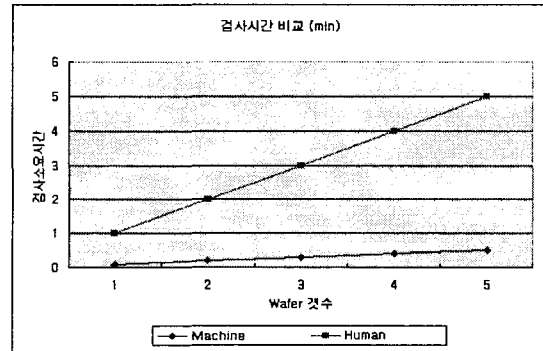
시뮬레이션 시간과 Wafer Interarrival Time에 따른 수율의 변화		단위 %									
Load Interarrival Time Human/Machine	구분	시뮬레이션 시간									
		초기	3시간	6시간	9시간	12시간	15시간	18시간	21시간	24시간	
30sec/30sec	Human	90.58	90.58	90.94	91.19	91.3	91.63	91.84	92.06	92.28	
	Machine	90.58	90.94	91.45	92.14	92.53	93	93.58	94.02	94.63	
60sec/30sec	Human	90.58	90.57	90.9	91.12	91.3	91.95	91.92	92.06	92.31	
	Machine	90.58	90.94	91.45	91.95	92.46	93	93.58	94.02	94.52	
30sec/60sec	Human	90.58	90.57	90.9	91.12	91.3	91.59	91.84	92.1	92.35	
	Machine	90.58	90.57	90.9	91.19	91.44	91.66	91.95	92.17	92.54	
60sec/60sec	Human	90.58	90.57	90.94	91.05	91.37	91.59	91.88	92.02	92.35	
	Machine	90.58	90.57	90.9	91.23	91.45	91.7	91.95	92.24	92.5	

Machine의 경우 시간의 흐름에 영향을 받지 않고, 거의 100%에 가까운 정확성을 나타내지만, Human의 경우 컨디션 및 교대시간 혹은 새로운 패턴의 오류에 대한 대응 등의 영향으로 인해, 고르지 않은 정확성을 나타낸다.



<그림 4.7. 정확성 비교>

아래의 결과는 Wafer의 수를 늘려가면서, 검사시간을 비교한 것이다. 그래프에서 보듯 사람의 경우 Wafer의 수가 늘어남에 따라 검사에 소요되는 시간의 증가율이 Machine보다 더 큰 것을 알 수 있다.



<그림 4.8. 검사 시간 비교>

### 5. 결론 및 향후 방향

Machine을 이용한 Pattern 분류는 속도와 정확성이 높으며, 사람에 의한 Pattern 분류에서 보다 더 높은 수율증대를 가져왔으며, 최상의 수율증대를 위해서는 사람이 일하는 것 보다 약 2배 빠른 Interarrival Time이 요구된다. 즉, 검사에 필요한 Wafer의 공급이 이전에 그것보다 더 빨라야 한다는 것이다.

참고문헌

- [1] Image Processing Analysis, and Machine Vision Second Edition, Milan Sonka, Vaclav Hlavac, Roger Boyle, PWS Publishing, ISBN 0-534-95393-X, 1998
- [2] Kevin. Zinke, "Yield enhancement techniques using neural network pattern detection", IEEE/SEMI Advanced Semiconductor Manufacturing Conference, 1997
- [3] Fukushima, T. "A method of feature extraction from an image for quality analysis", IEEE SMC 99 Conference Proceedings Vol.2, 1999
- [4] Carlos Ortega, "Human based knowledge for the probe failure pattern classification with the use of a back propagation neural network. Application of submicron linear technologies", IEEE/SEMI Advanced Semiconductor Manufacturing Conference, 1998
- [5] 김상운, 식별 알고리즘을 중심으로 한 패턴 인식 입문, 홍릉과학출판사, 1997
- [6] 이성황, 패턴인식의 원리 I, II 권, 홍릉과학출판사, 1994
- [7] 이성환, 문자인식 이론과 실제 I, II 권, 홍릉과학출판사, 1994
- [8] Yuji Sakai, "Wafer scale fail bit analysis system for VLSI Memory Yield improvement", International Conference on Microelectronics Test Structures Vol.3 1990
- [9] Robert V.Hogg, Probability & Statistical Inference
- [10] G. Nagy, S.Seth and M. Viswanathan, "A prototype document image analysis system for technical journals," IEEE Trans. on Computer, vol. 25, no. 7, pp. 10-22, 1992
- [11] B. S. Kim, S. M. kang and J. Y. Yang, "The bi-level image mapping using density information in character patterns," Journal of KITE, vol. 30, no. 8, pp. 645-651, 1993.
- [12] Algorithms for Graphics and Image Processing, T. Pavlidis, Computer Science Press. 1982