

GUI 를 이용한 슬라이싱 머신의 자동화 알고리즘 개발

김형태*, 양해정**, 송창섭***

A Development of the Algorithms for Automation of Slicing Machine with GUI Interface

Hyoung-Tae Kim*, Hae-Jeong Yang**, and Chang-Seop Song***

ABSTRACT

In this study, PC-based slicing machine and driving software were constructed for the purpose of automation of semi-conductor cutting process. The biggest feature of software is variation of parameter and include data base, signal monitoring, error report, corresponding action or automatic motion planing. Parameters were drawn and algorithms were developed to make software by GUI interface. The cutting experiment was done for sampled wafer to see the effectiveness of the soft automation. From the experimented and implemented results, it is shown that parameters for automation of slicing process could be drawn, then its algorithms constructed. It could be considered what is the merit of this slicing machine by comparing the PC-based and the NC-based.

Key Words : Slicing Machine (슬라이싱 머신), Semiconductor (반도체), Automation (자동화), GUI (그래픽 작업 환경), PC-Based CNC (PC 기반 CNC), Wafer Cutting (웨이퍼 절단), Object-Oriented (객체 지향)

1. 서론

슬라이싱 머신은 재료를 단방향으로 절단하는 작업에 사용되는 기기로서 반도체, 농업, 축산, 식품가공, 자동차 부품생산등에 사용되고 있다. 반도체 산업에서 슬라이싱 머신은 웨이퍼를 y 축으로 절단하지 않고 x 축으로만 절단하는 작업에 투입되고 있으며, 주로 전화기의 필터나 캠코더의 OLPE, 비디오 FDD 헤드, 실리콘 웨이퍼 등을 생산하는데 사용된다.

반도체 산업은 특성상 기술 집약적이고, 라이프 사이클이 짧기 때문에 제조하기 위한 반도체 장비 적기 투자와 양산체제의 조기확보가 사업의

성패를 가늠하게 된다.⁽¹⁾ 그러므로, 반도체 장비 기술은 생산성과 수율 증대를 향상시키기 위하여 장비의 자동화가 활발히 이루어지고 있으며,⁽²⁾ 반도체 공정의 자동화 사례도 발표되고 있다.⁽³⁾⁽⁴⁾ 슬라이싱 머신도 이러한 반도체 산업의 요구에 부응하기 위해 자동화의 필요성이 제기되고 있다.

또한 장비의 선호도와 개발 추세는 현재 CNC 나 NC 형태의 장비는 PC 기반의 GUI 와 Multi-tasking 환경을 가진 PC-Based CNC 타입으로 변화하고 있으며 이에 따라 프로그래밍 기법도 사용자가 NC 코드를 직접 작성하는 패턴에서 탈피하여 프로그래머가 객체 지향론적 방법으로 작성된 소프트웨어를 사용자가 쉽게 이용하는 패턴으로 변

* 한양대학교 대학원

** 한국산업기술대학교 기계설계학과

*** 한양대학교 기계공학부

화되었다. 시스템 개발에 있어서 객체 지향론적 방법은 현재 컴퓨터 시스템 개발 분야에서 비용 및 시간 감소, 시스템 안정성 증가, 시스템 유지·보수의 용이성 등의 장점을 가지고 있기 때문에 우수한 것으로 평가 받고 있으며, 많은 각광을 받고 있다.⁽⁵⁾⁽⁶⁾

그러나, 현재 사용되고 있는 대부분의 슬라이싱 머신들은 대부분 NC 코드(code)에 의해 사용자가 직접 프로그램을 작성하여 운용되고, 화면이 소형 액정 화면으로 텍스트(text)에 의해 표시된다. 이러한 이유로 사용자가 작동하기가 다소 어려워 어느 정도 훈련을 해야 한다. 또한 공정을 운용하기 위한 정보를 입력하는 것이 비교적 불편하며, 사용자에게 장비의 상태에 대한 정보를 보여주는 데도 한계가 발생하고 있다. 따라서 슬라이싱 머신을 새로 구입하여 설치하거나 공정이 변경되는 경우, 공정을 운용하는데 까지 걸리는 시간과 비용이 추가로 발생하게 되고, 다품종 소량 생산에 적합하지 않는 문제점을 발생하게 된다. 그러므로 이러한 슬라이싱 머신의 단점을 보완하기 위해서는 hard automation 보다는 유연성을 확보할 수 있는 soft automation 을 실시해야 한다.⁽⁷⁾

따라서, 생산성 향상 및 유연성(flexibility) 확보를 위한 장비의 soft automation 을 위하여 PC-Based CNC 슬라이싱 머신과 GUI 환경에서 동작하는 소프트웨어를 제작하였다. 소프트웨어는 몇가지 제안된 알고리즘을 토대로 객체 지향 언어인 C++로 작성되었고, 사용자가 몇가지 파라미터만 입력하면 슬라이싱 공정이 자동으로 행해지도록 제작되었다. 또한 제작된 슬라이싱 머신의 성능과 유용성을 테스트하기 위해 실제 생산되는 시편을 절단하여 결과를 고찰해 보았다.

2. 하드웨어 구성

2.1 슬라이싱 머신

슬라이싱 머신은 크게 정반, X, Y 및 Z 각 축의 테이블, 스텝 모터, 스피들, 마그네틱 척(magnetic cherk), 다이아몬드 블레이드(diamond blade), 터치 스크린, 절삭수 노즐, 배선부로 이루어져 있다.⁽⁸⁾ 각 축 테이블에는 모터와 보울 스크류가 커플링되어 있으며, 정반위에 설치된 Y 테이블에는 Z 테이블이 올려져 있다. Z 테이블에는 스피들이 고정되어 있어 레일을 따라 상하로 움직인

다. 즉, Y 와 Z 축은 스피들을 이송하는 역할을 하고 있다. 스피들 끝에는 웨이퍼등을 절단하는 다이아몬드 블레이드와 절삭수를 분사하는 절삭수 노즐이 설치되어 있다. X 테이블은 마그네틱 척을 좌우로 이동시키는 역할을 한다. 슬라이싱 머신에서 절단될 자재는 지그(jig)에 고정된 후, 지그를 다시 마그네틱 척에 정렬 및 고정시킨 후에 절단 동작에 들어간다. 사용자와 interactive 하게 장비에 대한 정보 표시와 데이터 입력이 가능하도록 터치 스크린을 장착하였고, PC 와 연결되어 있다. 배선부에서는 PC 에서 나오는 신호를 증폭하여 PLC 나 모터로 전달하여 동작이 수행되도록 하였고, PC 쪽에서 필요한 신호를 발생시켜 전송하도록 제작하였다. 또한 스피들 회전수를 조정할 수 있는 인버터(inverter)도 설치하였다. Fig. 1 은 슬라이싱 머신의 내부이고, Fig. 2 는 슬라이싱 머신의 외관이다.



Fig. 1 Photo of slicing machine for inside view

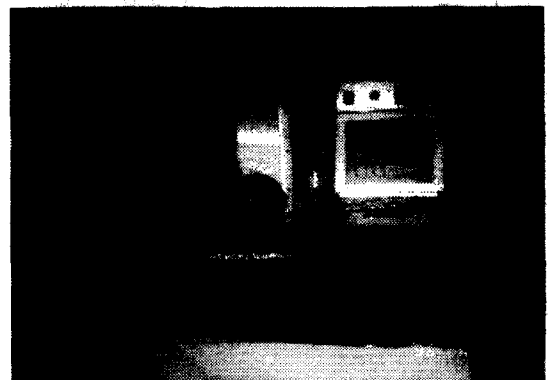


Fig. 2 Photo of slicing machine for outside view

2.2 Controller

PC는 슬라이싱 머신을 동작시키고, 관리하며, 현재 상태를 디스플레이하는 역할을 맡고 있으며 OS는 Windows 95이다. 내부에는 사용자가 NC 코드로 지령을 내릴 수 있는 모션 컨트롤러가 장착되어 있다. 모션 컨트롤러가 수행하는 기능은 위치제어, 각종 파라미터 변경, 디지털 입·출력 송수신, 리미트 신호 체크등을 실시한다. 모션 컨트롤러에서 나가는 모든 신호는 배선부와 연결되는 인덱서 보드(indexer board)와 연결되어 있으며, 출력 신호의 경우 배선부의 릴레이 박스(relay box)를 통하여 증폭된 후, 슬라이싱 머신의 각 연결부로 전달된다. 신호의 입출력 관계는 Fig. 3 과 같다.

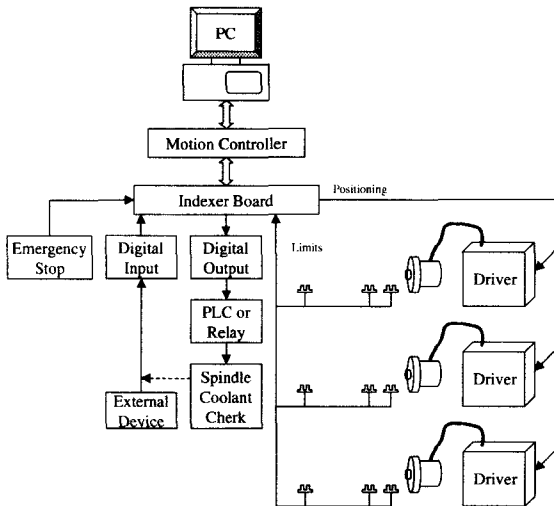


Fig. 3 Signal interaction between PC and slicing machine

3. 소프트웨어

3.1 슬라이싱 머신의 프로그램 구조

프로그램 제작시 각 부분을 클래스나 함수로 모듈화시켜 필요한 부분에서 모듈을 호출하면 기능이 자동으로 실행되도록 구조화를 시켰다. 모듈화된 클래스는 메인 프레임(main frame), 뷰(view), 도큐먼트(document)등 기본 클래스에서 호출되는 클래스 들이며, 주로 하드웨어 설정이나 절단시 화면에 나타나는 윈도우와 대응된다. 모듈화된 함수는 이송동작이나 I/O 체크등 프로그램 전체에서 빈번히 호출되는 서브루틴(sub-routine)들이 주종을

이룬다. 기본 클래스는 실행 파일을 구성하고, 나머지 부분은 라이브러리 파일인 DLL(Dynamic Linking Library)로 작성하여 실행파일에서 서브루틴을 호출하도록 구성되어 있다.

메인 프로그램이 하는 일은 툴바등 메뉴를 띄우는 역할을 하고, 뷰가 하는 일은 각 서브 메뉴에 해당하는 클래스 호출, 메시지 처리를 한다. 위치/속도 표시 윈도우에서는 모션 컨트롤러로부터 위치 속도등을 표시해 주는 데이터를 읽어와서 사용자가 보기 쉽도록 mm 단위로 환산하여 표시하는 역할을 한다. I/O 디스플레이 윈도우도 모션 컨트롤러로부터 I/O 신호 및 리미트 신호를 읽어서 화면에 On-Off를 표시해 주면서 오류를 체크한다. 하드웨어 셋업시 나타나는 윈도우들도 사용자가 입력한 데이터를 환산 또는 변형시켜 모션 컨트롤러에 전달하거나 내부변수에 설정한다. Fig. 4는 프로그램의 클래스 구조를 나타낸 것이다.

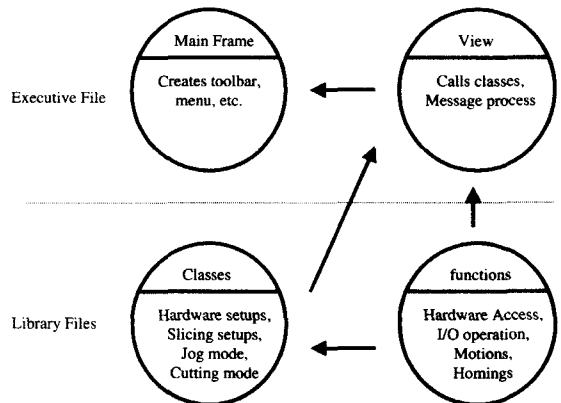


Fig. 4 Diagram of program structure

3.2 메인 화면의 제어 알고리즘

메인 화면은 풀-다운(pull-down) 메뉴와, 툴바(toolbar), 로그(log) 기록, 위치·속력 윈도우, I/O 윈도우, 슬라이싱 파라미터 윈도우로 구성되어 있다. 화면 최상단의 타이틀 바(title bar)는 현재 사용되고 있는 데이터를 표시한다. Fig. 5는 메인화면을 나타낸 것이다. 풀-다운 메뉴는 일반적인 윈도우 어플리케이션과 같이 tree 식 계층구조로 가지고 있다. 슬라이싱 머신을 운용하는데 필요한 기능을 파일, 편집, 보기, 하드웨어 셋업, 하드웨어 작동, 윈도우 디스플레이, 슬라이싱, 도움말등 8가지 항목으로 나누고 하위 항목으로 세분화된 기능

을 추가하였다. Fig. 6 은 메뉴를 tree 구조로 나타낸 것이다. 툴바는 데이터 입력력과 작업 시작, 호밍(homing)등 풀-다운 메뉴에서 자주 사용되는 기능을 사용하기 편하도록 모아놓은 것이다. 로그 기록은 슬라이싱 머신 사용중에 발생한 사고나 중요 사건을 표시 및 저장한 기록이므로 차후에 발생한 사건을 검토할 경우 유용하게 사용할 수 있다. 위치·속력 윈도우는 현재의 위치와 속력을 표시하며, I/O 윈도우는 디지털 I/O 상태와 리미트 상태를 체크 및 표시한다. 슬라이싱 파라미터 윈도우는 설정된 슬라이싱 파라미터 값을 표시한다.

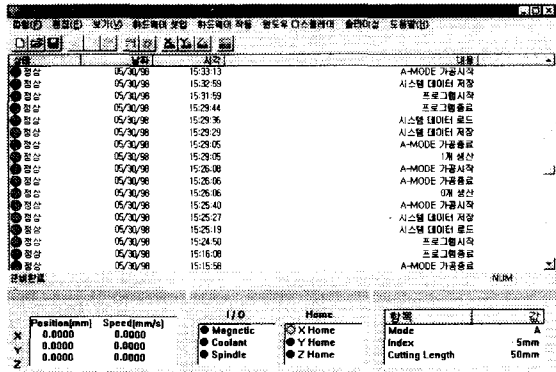


Fig. 5 View of main menu

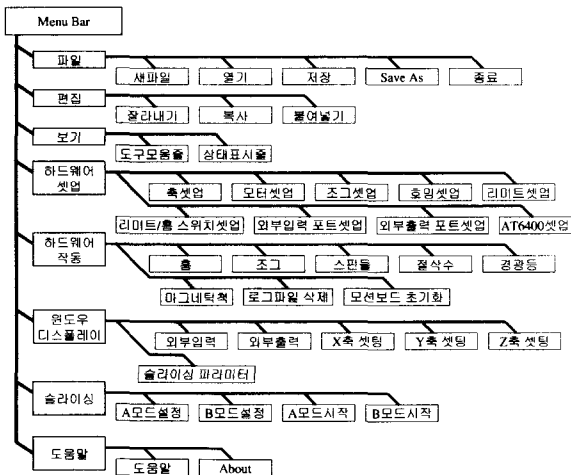


Fig. 6 Hierarchical structure in pull-down menu

3.3 데이터 입력 및 처리 알고리즘

슬라이싱 머신을 운용하는데 필요한 데이터는 하드 디스크로부터 파일 형태로 저장 및 호출할

수 있으며 다른 이름으로도 저장이 가능하도록 프로그램 하였다. 또한 데이터를 처음서부터 새로 작성할 수도 있다. 사용자가 파라미터를 입력하면 버퍼(buffer)에 임시 저장하였다가 데이터를 저장하는 명령이 발생하면 프로그램 자체 포맷으로 정리하여 데이터를 저장한다. 데이터를 호출하는 경우는 버퍼에 임시 저장한 다음, 소프트웨어 상에서 데이터를 참조하는 변수를 모두 업데이트(update)시키고, 모션 컨트롤러도 재설정 시킨다.

3.3.1 로그 기록

로그 기록을 작성하는 데이터는 비상정지나 리미트 센서 작동, 운전 시작 등의 사건이 감지되었을 때 발생된다. 사건이 발생되면 사건을 코드화하여 지정된 메시지 통로를 통해 메시지 큐(queue)로 코드를 전송한다. 메시지 큐에서는 코드를 해독하여 코드에 따른 대응을 실시한 다음, 메인 화면의 정상/오류 여부, 날짜, 시간, 내용 등의 리스트에 기록을 남기고 저장을 실시한다. 로그 기록 데이터는 사용자에게 의해 임의로 저장 및 호출이 불가능하게 되어 있으며, 다만 로그 기록을 지울 경우 텍스트 파일로 내용을 남기는 것은 가능하다. Table. 1 에 사건 코드와 내용을 정리하였다.

Table 1 Event codes and their contents

Code	내용	비고
0	소프트웨어 비상정지	오류
1	모션 컨트롤러 초기화	정상
2	모션 컨트롤러 초기화 실패	오류
3	+X Limit 작동	오류
4	-X Limit 작동	오류
5	+Y Limit 작동	오류
6	-Y Limit 작동	오류
7	+Z Limit 작동	오류
8	-Z Limit 작동	오류
9	시스템 데이터 호출	정상
10	시스템 데이터 저장	정상
11	새 시스템 데이터 작성	정상
12	스핀들 작동 오류	오류
13	하드웨어 비상정지	오류

로그 기록이 화면에 나타나기까지 거치는 단

세로는 사건의 감지, 메시지 전달, 사건의 판단, 조치동작, 사건의 통지 순서를 거치게 된다. 따라서 로그 기록은 사건 처리의 자동화 개념을 포함한다. 사건의 감지는 하드웨어적인 것인 경우, 모션 컨트롤러로 들어오는 신호를 항시 체크하여 오류가 있는지 판별하고, 소프트웨어적인 것은 임의의 동작이 발생하면, 그 결과를 체크하게 된다. 일단 사건이 감지되면 사건을 코드화하여 메시지 큐로 코드를 전달한다. 메시지 큐에서는 데이터 베이스화된 사건들과 코드를 비교하여 일치하는 것을 찾아 그에 알맞은 조치 동작을 행한다. 조치동작이 완료된 후에 사용자에게 내용을 통지하고, 사용자가 확인할 때까지 대기 상태로 된다.

3.3.2 파라미터 설정

사용자로부터 입력을 받아들이기 위해 슬라이딩 머신을 운용하는데 필요한 인자들을 분석하여 도출하였다. 도출된 인자들은 데이터의 대부분을 차지하고 있다. 파라미터는 크게 슬라이딩 머신 자체에 관한 것과 절단 동작 자체에 관한 것으로 분류할 수 있으며 전자를 하드 파라미터, 후자를 소프트웨어 파라미터로 하였다. 각각은 하드웨어 설정과 가공 설정에서 쉽게 변경할 수 있다. Table. 2는 소프트웨어 상에서 변경 가능한 파라미터들을 데이터 저장 순서대로 정리한 것이다.

슬라이딩 머신 자체에 관한 하드 파라미터는 가속도, 감가속도, 호밍 방법등 주로 모션 컨트롤러 설정과 연관이 있는 것들이 대부분이다. 이러한 파라미터들은 장비를 처음 셋업할 때 보정을 시켜 놓으면 이후로 잘 변경되지 않으며, 공정에 따라서 필요한 경우 변화시킬 수 있다.

절단 동작에 관한 소프트웨어 파라미터는 절단 동작 자체를 분석하여 필요한 정보를 추출해 내었다. 이러한 파라미터들은 특정 위치나, 거리, 반복횟수, 속도등으로 구성되어 있으며, 이 파라미터를 토대로 소프트웨어가 NC 코드를 작성하여 절단 동작을 수행하게 된다.

파라미터의 입력은 풀-다운 메뉴에서 하드웨어 설정이나 가공정보 설정을 누르면 해당되는 파라미터의 다이얼로그 박스가 화면에 나타나 편집할 수 있도록 하였다. 사용자가 숫자로 된 파라미터를 입력할 때는 키보드 또는 마우스 조작 없이 터치 스크린을 사용하여 직접 입력할 수 있도록 파라미터 입력 박스 옆쪽에 키패드(keypad)를 나타나

게 하였다. 스펀들이나 마그네틱 척 같이 외부와 입·출력 포트를 설정하는 경우는 슬라이더 바 (slider bar)로 포트를 변경할 수 있게 하였다. Fig. 7은 모터 설정시 나타나는 화면의 예이고, Fig. 8은 출력 포트 변경시의 화면이다.

Table 2 Parameters in slicing machine controller

항 목	파라미터
일방향 절단 Mode	초기 이송거리, 절단거리, 인덱스, 절단깊이, 이송속도, 절단속도, 복귀점, 다단 절단, 다단 인덱스, 대각선
왕복 절단 Mode	단방향 절단 Mode 와 동일
모션 보드 설정	Address, IRQ, DMA 등
리미트 스위치	각축의 (+), (-) 및 홈 limit
리미트 감가속도	각 축에 대하여 설정
모터	펄스, 펄스 폭, CCW 등
디지털 입력	조그, 비상 정지, 스펀들 가동
디지털 출력	조그, 경광등, 비상정지, 홈 최종 속도
X 축 설정	피치, 가속도 감가속도
X 축 홈	축의 (+)(-) 방향, Home
X 축 조그(jog)	조그 가속도, 감가속도, 이송속도,
Y 축 설정	X 축 설정과 동일
Y 축 홈	X 축 홈과 동일
Y 축 조그	X 축 조그와 동일
Z 축 설정	X 축 설정과 동일
Z 축 홈	X 축 홈과 동일
Z 축 조그	X 축 조그와 동일

3.4 동작 알고리즘

슬라이딩 머신에서 동작은 크게 생산을 위한 절단 동작과 장비의 운영을 도와주는 부분 동작으로 나눌 수 있다. 절단 동작은 각 부분동작으로 구성되며, 부분 동작이 모여 상위 레벨의 부분 동작을 만들기도 한다. 즉, 상위 레벨의 부분 동작은 하위 레벨의 부분 동작을 구성하는 서브루틴들을 호출하는 구조로 되어 있다.

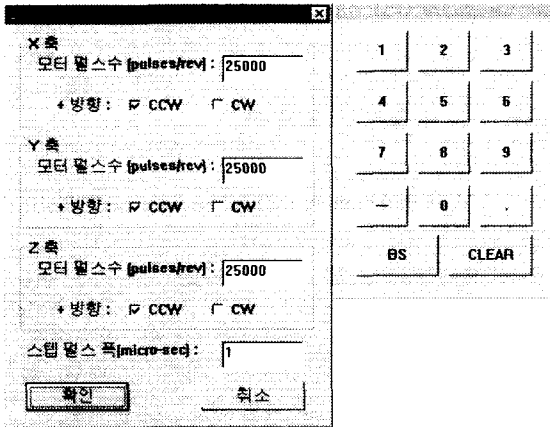


Fig. 7 View of keypad and dialog boxes for motor setup

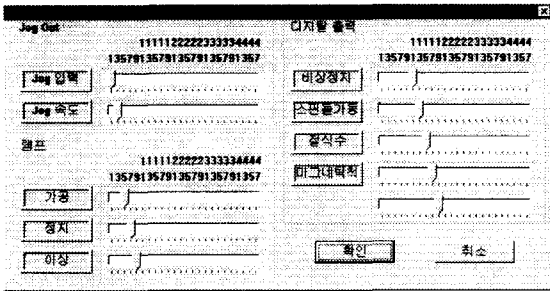


Fig. 8 View of digital output terminal setup

3.4.1 절단 모드

절단 모드는 셋업된 데이터에 따라 자동으로 절단을 수행하는 모드이다. 지원하는 모드는 A·B 모드를 지원하며, 슬라이딩 파라미터 조작으로 C·D 모드도 가능하다. 절단 모드를 실행시키면 Fig. 9와 같은 윈도우가 나타난다. 화면상에는 현재 수행중인 동작과 다음 동작, 생산량이 문자로 표시되고, 전체 진행 상황과 현재 동작의 진행 상황이 프로그레스 바(progress bar)로 표시된다. 시작 버튼을 누르면 절단 동작이 시작되고, 다시 누르면 중단되도록 하였다. 작업 수행 중에 일시 정지를 할 필요가 있는 경우에 일시 정지 버튼을 누르면 Z축이 원점으로 이송되어 대기 상태로 있다가 다시 누르면 Z축이 이전위치로 복귀하여 나머지 공정을 수행한다. 절단 모드에서 브레이드 파손이나 정렬 불량으로 인한 작업 중지가 발생할 때를 대비하여 바로 조치를 취할 수 있도록 호밍과 스피들·마그네틱 칩등의 On-Off 기능을 추가하였다.

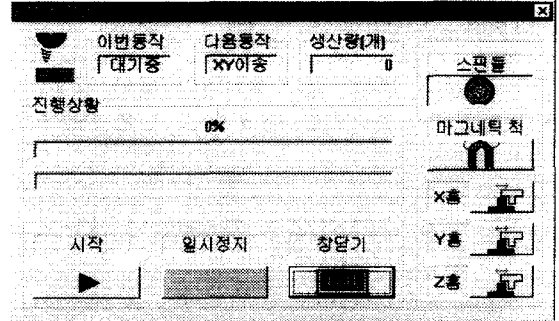


Fig. 9 View of cutting mode

절단 방법은 A·B·C·D 4가지가 있으나, 크게 일방향 절단 방법과 왕복 절단 방법이 있다. A·C 모드가 일방향 절단에 해당되며, 대부분의 경우에 사용되는 방법이고, X 축이 (+)방향으로 이송될 때만 절단된다. B·D 모드는 왕복 절단 방법에 속한다. Fig. 10은 일방향 절단시의 툴 패스(tool path)이고, Fig. 11은 왕복 절단시 툴 패스이다.

일방향 절단 방법의 궤적을 초기 이송위치, 절삭깊이, 절삭거리, 복귀점, 인덱스, 인덱스 횡수, 다단 절단 횡수등으로 표현이 가능하므로 설정된 데이터에 따라 자동으로 절단 동작을 수행하게 된다. 왕복 절단 방법의 궤적은 초기 이송위치, 절삭깊이, 절삭거리, 인덱스, 인덱스 횡수, 다단 절단 횡수등으로 구성되며, 일방향 절단과 마찬가지로 설정된 데이터에 의해 자동으로 절단 동작을 수행한다. 속력은 절단 방법에 상관없이 이송속도와 절삭속도로 설정되도록 하였다.

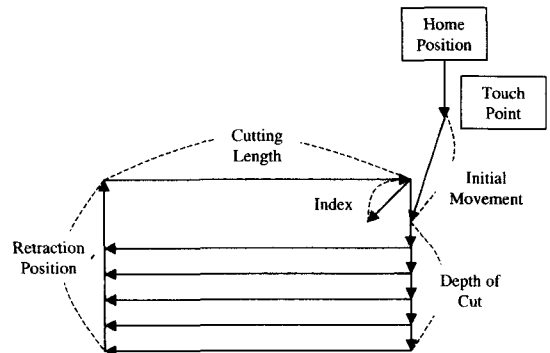


Fig. 10 Tool path in uni-directional slicing

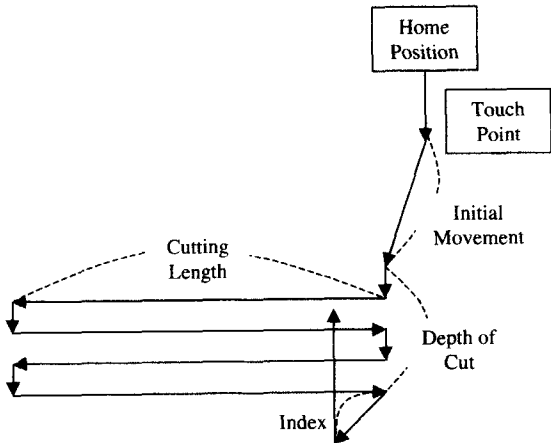


Fig. 11 Tool path in bi-directional slicing

작업중 일시 정지 기능은 멀티-태스킹(multi-tasking) 프로그래밍 기법중 하나인 스레드(thread)를 사용하여 구현하였다. 절단 동작을 메인 스레드로 하고, 절단 시작을 하면 메인 스레드를 호출하며, 일시 정지 기능 입력이 들어오는 경우에는 메인 스레드를 일시 정지시킨 다음, 대기 상태에 들어 간다. 작업 계속 입력이 들어오면 이전 위치로 돌아가서 일시 정지 이전의 동작을 수행하는 서브 스레드를 호출하고, 서브 스레드 종료시 메인 스레드를 계속하게 된다. 이전 동작으로 복귀하는데 서브 스레드를 사용한 이유는 사용자가 일시 정지를 한 동작에서 여러 번 걸 수도 있기 때문이다. Fig. 12는 절단 동작 중 일시 정지시의 알고리즘을 나타낸 것이다.

3.4.2 부분 동작

슬라이싱 머신을 테스트 하거나, 셋업 및 위치 설정을 위하여 필요한 일부 동작을 실행하는 것을 말하며, 부분 동작에 해당하는 것으로는 조그, 호밍, 모션 컨트롤러 초기화, 스피들, 마그네틱 척, 절삭수등이 있다. 조그는 사용자가 버튼을 누르고 있는 동안 계속하여 이송이 되는 것이다. 조그 동작은 장비를 처음 셋업하거나 공정이 바뀌어서 특정한 위치를 찾아내야 하는 경우에 사용되므로 현 위치에서 상대위치로 이송되거나 절대위치로의 이송 및 원점을 찾는 동작도 종종 필요로 한다. 따라서 조그를 실행시키면 Fig. 13 과 같이 호밍 및 상대위치, 절대위치 이송도 가능하도록 윈도우를

제작하였다. 또한, 상대위치를 일정하게 설정해 놓고 계속하여 한 방향으로 이송되도록 응용하면 MPG(manual pulse generator) 기능도 가능하다.

호밍 기능은 홈 센서를 설치한 원점을 찾아 이송하는 동작이고, 모션 컨트롤러 초기화는 모션 컨트롤러의 설정 값을 PC 를 처음 실행시켰을 때와 같은 상태로 만들어 주는 기능이다. 스피들, 마그네틱 척, 절삭수 등은 각 해당 유니트를 버튼 토글(toggle)로 On-Off 시키는 기능이다.

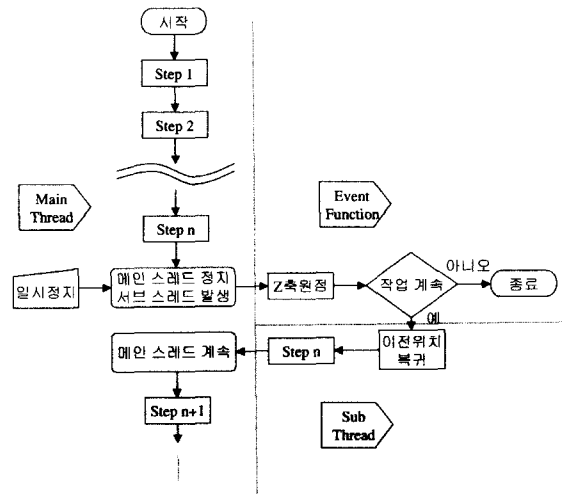


Fig. 12 Flow chart for slicing pause algorithm

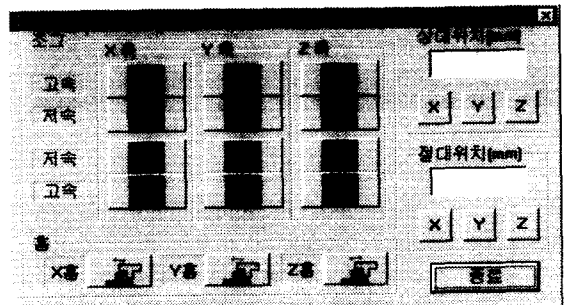


Fig. 13 View of jog mode

4. 실험 및 고찰

4.1 실험

슬라이싱 머신을 제작하여 설치한 후, 소프트웨어를 설치하여 장비에 맞도록 파라미터 입력 및 보정 작업을 수행하였다. 그리고 장비의 셋업이

시작하여 종료할 때까지 시간에 관한 요소를 측정하였다. 측정된 요소로는 셋업시간, 공정변경시간, 그리고 장비를 사용할 사용자 교육에 걸리는 시간 등을 고려하였다. 장비의 셋업이 종료한 이후에 70mm×12mm 크기의 그라파이트(graphite) 시편을 사용하여 시험 절단을 수행하였다. 시험절단시 그라파이트 시편을 지그에 본딩(bonding)한 후, 마그네틱 척에 고정시켰다. 그 다음 파라미터를 입력하고 정밀도를 측정하였다. Fig. 14는 그라파이트 시편을 절단하는 장면이다.

그라파이트 테스트 절단으로 프로그램과 알고리즘 및 하드웨어적인 장치에 이상이 없음을 확인한 후, L사에서 생산하고 있는 부품 2가지를 절단하여 정밀도를 측정하였다. 각 부품들은 그라파이트 시편과 같은 방법으로 각각에 해당되는 지그에 본딩한 후, 마그네틱 척에 고정시켰다. 그 후 파라미터를 변경시키고 절단을 수행하였다. Fig. 15는 테스트 절단에 사용된 콘텐서 시편들이다.

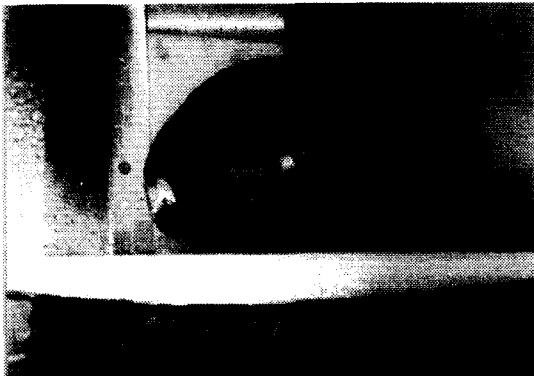


Fig. 14 Photo of test cutting for graphite

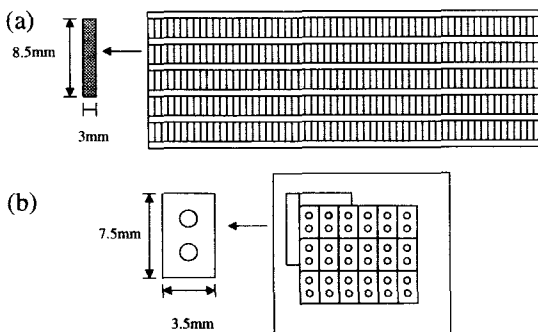


Fig. 15 View of test pieces

4.2 결과 및 고찰

장비를 셋업하여 보정할 때까지 걸리는 시간과 절단 제품이 바뀌는 경우 파라미터를 변경하는데 걸리는 평균 시간, 사용자를 숙련될 때까지의 교육 시간을 측정하였고, Table. 3에 O사의 기존 장비와 비교하였다. 그 결과 PC-Based 슬라이싱 머신이 셋업 시간을 제외하고 전체적으로 시간이 적게 걸려 생산라인을 새로 설치하거나 공정이 바뀌는 경우 유효함을 알 수 있다. 셋업 시간이 비슷한 것은 셋업 작업 내용이 PC Base 형태가 기존 것과 비슷하기 때문이라고 볼 수 있다.

PC-Base 형태는 그라파이트 절단시 3 μ m의 오차를 얻었고 2가지 시편을 절단한 결과 평균 2.5 μ m의 오차를 얻었다. 그리고, 기존 O사의 장비로는 평균 5 μ m의 오차가 발생하였다. 정밀도 자체는 각 기기마다 큰 편차를 보이지 않았는데, 이것은 정밀도가 소프트웨어적인 알고리즘 보다는 하드웨어적인 장치에 영향을 받기 때문이라고 볼 수 있다.

Table. 3 Comparison of time elements

항목	기존 형태	PC-Base 형태
셋업 시간	1 주	1 주
공정 변경 시간	1 시간~1 주	3 분~10 분
교육 시간	1 달	1 주

5. 결론

본 연구에서는 반도체 절단 공정에서 슬라이싱 머신을 자동화하기 위해 PC-Base 형태의 슬라이싱 머신을 제작하고, 구동 소프트웨어를 제작하였다. 또한, 실제로 시편을 절단하여 PC-Base 슬라이싱 머신의 유효성을 증명하였다. 이를 토대로 다음과 같은 결론을 얻을 수 있다.

1. 슬라이싱 작업을 PC-Base 슬라이싱 머신으로 자동화가 가능함을 보였다.
2. 슬라이싱 머신을 구동하는 소프트웨어는 GUI 및 멀티-태스킹 환경을 이용하여 작성할 수 있음을 보였다.
3. 슬라이싱 작업을 자동화하기 위한 파라미터를 추출하고 알고리즘을 구성하였다.
4. 자동화된 슬라이싱 머신을 실제로 절단작업을 수행시킨 후, 여러가지 요소를 기존 장비와 비

교하여 유효성이 있음을 보였다.

참고문헌

1. 조식승, "반도체 산업의 중장기 발전을 위한 인력개발 방안에 관한 연구," 한양대학교 대학원, pp. 35-45, 1997.
2. 생산기술연구원, "공업 기술 수요 조사 종합 보고서 [전기 전자 분야]," 상공부, pp. 1092-1109, 1990.
3. 박중헌, 김종원, 권옥현, "자동화된 VLSI 생산 시스템 운용을 위한 소프트웨어 구조 및 프로토콜 설계," 제어·자동화·시스템 논문지, Vol. 3, No. 1, pp. 94-100, 1997.
4. 이성진, "GUI를 지원하는 와이어 본딩 시스템의 구조," 서울대학교 대학원, pp. 1-42, 1998.
5. 이동엽, "객체지향 방법론을 적용한 철근 모듈 설계 시스템," 한양대학교 대학원, pp. 1-3, 1996.
6. 최승욱, "객체 지향기술을 이용한 VHDL 톨 개발 시스템의 구현에 관한 연구," 서울대학교 대학원, pp. 1-12, 1994.
7. Serope Kalpakjian, "Manufactureing Engineering and Technology 2nd Ed.," Addison Wesley, pp. 1121-1123, 1992.
8. 김형태, 양해정, 송창섭, "윈도우 기반의 반도체 절단 공정의 자동화 시스템," 한국정밀공학회 추계학술대회 논문집, pp. 750-753, 1998.