

## 반도체 테스트 핸들러의 온도 제어 시스템 개발 I - 시스템 구성

# Development of Temperature Control System for Semiconductor Test Handler I - System Design

°조수영\*, 이호준\*, 강태삼\*\*, 이성은\*, 김재용\*\*, 김영록\*\*\*

\*호서대학교 정보통신공학부(Tel:+82-418-40-5434; Fax:+82-418-549-4715; E-mail:hojoon@dogsuri.hoseo.ac.kr)

\*\*호서대학교 전기전자제어공학부(Tel:+82-418-40-5485; Fax:+82-418-40-5480; E-mail:tskang@dogsuri.hoseo.ac.kr)

\*\*\*(주)미래산업 부설연구소(Tel:+82-417-554-5070; Fax:+82-417-554-5090; E-mail:yorkim@miare.co.kr)

**Abstract** The temperature control system for semiconductor test handler is designed. We controlled the temperature of chamber using 3-wire RTD sensor and MVME Embedded controller. VxWorks that is a real-time operating system is used and heater is controlled by PWM. Temperature fluctuation of chamber is decreased within 0.3°C, which is about one-half of that of commercial controller.

**Keywords** VxWorks, VME, RTD, Test Handler

### 1. 서론

반도체 생산과정중에 후공정 장비인 테스트 핸들러는 완제품으로 만들어진 반도체들을 저온에서 고온까지의 온도 범위내에서 동작하는 항온조 속에서 반도체를 테스트가 감사하도록 소켓에 접속시켜주고, 검사된 반도체를 등급에 따라 분류하는 장비이다. 일정한 온도를 유지한 상태에서 테스트하는 장비이다. 이 테스트 핸들러의 항온조내의 온도를 일정하게 유지하기 위해서는 적절한 하드웨어, 개발 tool 그리고 RTD 센서의 선정과 교정이 매우 중요하다.

본 논문에서는 하드웨어의 구축과 RTD 센서의 선정, 컨트롤러 보드와 Industry Pack과 I/O 시스템, VxWorks에 연결되어 제어 되는 전체 시스템을 구축하고, 기존의 제어기와 본 실험에 사용된 제어기의 성능을 비교 분석하였다.

### 2. 온도 제어 시스템

#### 2.1 센서 선정 및 온도대 저항의 관계

센서는 제어 시스템의 정밀도 및 신뢰도 부분에서 중요한 역할을 하며, 제어 시스템의 성능은 센서의 성능에 절대적으로 의존한다. 따라서 항온조 시스템에 적합한 센서를 선정하는 것은 매우 중요하다. 본 실험에서는 정밀도가 0.1°C ~ 0.01°C이고, sensitivity가 +0.5%(25°C)인 백금 저항 온도계를 사용하였다. 백금 저항 온도계는 -250°C ~ +900°C까지의 측정이 가능하므로 동작 범위가 넓고, 다른 RTD 센서에 비해서 선형성이 우수하다. 또한, 반응속도가 몇초 이내이다.

이상적인 백금저항 온도계는 -200°C에서 0Ω, 0°C에서 100Ω,

200°C에서 200Ω의 저항값을 가져야하지만, 일반적으로 센서의 비선형성에 의하여 0°C에서 100Ω이 워려질 때 -200°C에서 약 1

8.5Ω, 200°C에서 약 175.8Ω, 260°C ~ 270°C 정도에서 200Ω의 저항값이 나오기 때문에 온도 7차방정식을 이용하여 값을 교정하여야한다. 식 (2.1)은 온도 7차 방정식을 나타낸다.[4]

$$R(t) = R(0)(1 + at + bt^2 + ct^3 + dt^4 + \dots) \quad (2.1)$$

R(t) : t°C에서의 저항값  
R(0) : 0°C에서의 저항값  
a, b, c, ... : 저항의 온도계수

본 실험을 위하여 30여개의 Pt 센서를 검증해 본 결과 저항의 온도 계수는 평균적으로  $a=0.3975312 \times 10^{-2}$ ,  $b=-0.5733870 \times 10^{-6}$ ,  $c=-0.2683396 \times 10^{-10}$  정도의 값을 보였다. 본 실험에서는 c값은 상대적으로 매우 작으므로 2차 방정식까지의 값을 사용하여 온도 보정을 실시하였다. 식 (2.1)을 온도 t에 대한 식으로 정리하면 식 (2.2)와 같다.

$$t = \frac{-a + [a^2 - 4b(1 - \frac{R(t)}{R(0)})]^{1/2}}{2b} \quad (2.2)$$

저항온도계의 온도대 저항 특성은 0°C에서 100°C 온도 사이의 평균 기울기로 구별하고 있다. 이 기울기를 α 값이라 부르며 식 (2.3)과 같이 정의되어진다.

$$\alpha = \frac{R(100) - R(0)}{100R(0)} \quad (2.3)$$

R(100) : 100°C에서의 저항값

백금 저항온도계의 경우에는 α 값이 0.003925보다 크거나 같을

때 표준형 백금 저항온도계로 사용한다. 즉,  $\alpha$  값이 클수록 고순도의 백금 저항온도계이다.

본 실험에서는 3선식 Pt 센서를 이용하므로, 선저항 성분을 포함한 센서의 저항값을 읽어들이고, 다시 선저항 성분만의 저항값을 읽고, 선저항 성분을 포함한 센서의 저항값에서 선저항 성분의 저항값을 제거하여 순수한 센서만의 저항값을 읽어 들여서 식 (2.2)를 이용하여 온도를 측정하였다.

### 2.2 RTD 센서의 원리와 결선

RTD(Resistance Temperature Detector) 센서는 저항 센서에 여기전류를 흘려주어서 센서 양단에 걸리는 전압 강하를 측정하는 센서이다. RTD 센서는 결선 방식에 따라 2선식·3선식·4선식으로 분류된다.[5]

2선식은 회로도로는 간단하지만 센서에서 읽혀지는 저항값과 리드선의 선저항이 같이 읽혀지므로, 주변온도와 길이에 따라 저항값이 많이 변화하게 된다. 그래서, 보통 리드선은 저항이 작은 구리선을 사용하지만 정밀 온도 측정에는 적당하지 않다.

3선식 저항 측정기는 산업체에서 일반적으로 가장 많이 사용하는 방식으로써, 리드선에 의한 저항 측정기의 저항오차를 상쇄시켜서 저항 측정기만의 저항값을 읽을 수 있다. 특히, 산업체에서는 저항 측정기의 리드선이 긴 경우가 많으므로, 상대적으로 두 리드선간의 저항차가 작아지게 되므로 정확도가 높아지게 된다.

4선식의 저항 측정기는 리드선에 의한 저항에 상관없이 센서의 저항값만을 측정할 수 있다. 이 측정법은 가장 높은 정확도를 갖는 측정 방법이다. 하지만, 이 측정법은 2번을 측정하여 평균을 구해야하는 사용상의 복잡함이 있다. 특히, 안정한 전류원만 있다면 더 정확한 저항 측정이 가능하다.

본 논문에서는 3선식 저항 측정기를 이용하여 실험을 하였다. 그림 1은 3선식 저항 측정기의 원리를 보여주는 것이다.

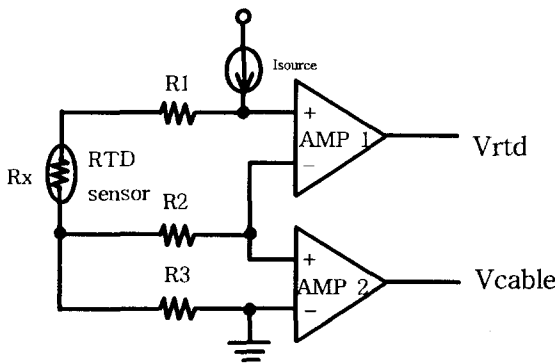


그림 1. 3선식 저항 측정기 회로  
Fig. 1. Circuit of 3-wire RTD

그림 1회로를 분석해보면,

$$\begin{aligned} V_{rtd} &= (R_1 + R_x) \cdot I_{source} \\ V_{cable} &= R_3 \cdot I_{source} \\ V &= V_{rtd} - V_{cable} \end{aligned} \quad (2.4)$$

저항 성분인  $R_1 = R_2 = R_3$  이면,

$$V = R_x \cdot I_{source} \quad (2.5)$$

식 (2.5)에서 보는 바와 같이 센서만의 전압값을 이용하여 온도를 측정할 수 있다.

### 2.3 온도 제어 시스템의 구성

본 실험을 위해서 RTD board는 IP(Industry Pack)-RTD를 사용하였고, 출력신호의 제어를 위해서는 IP-Digital 24를 사용하였다. IP-RTD는 12개까지의 RTD 입력(본 실험에서는 2개의 입력을 사용)중 하나를 선택하기 위하여 controller가 채널 선택신호(CS)를 IP-RTD에 주며 선택된 채널의 온도가 아날로그 출력(0 ~ +5V)을 통하여 나오며, 이 RTD는 0.25mA의 낮은 여기전류로 동작한다. 또한, 두 개의 인스트루먼트 증폭기를 사용하여 낮은 드래프트와 선형적인 특성을 제공한다. 이 인스트루먼트 증폭기의 전압이득은 100으로 고정하여 사용하며, 200Ω의 센서 저항값에 대하여 +5[V]의 전압이 되도록 하였다.

IP-Digital 24는 24개의 신호선을 입력과 출력으로 사용이 가능하다. 본 실험에서는 14개의 신호선을 A/D converter로부터 입력받고, 2개의 신호선은 제어신호 입력으로 사용하고, 2개의 신호선(Output Enable, Start of Conversion)은 출력으로 사용하였다. IP-RTD로부터의 출력신호는 AD711을 사용하여 증폭하고, 14bit A/D converter를 이용하여 디지털 신호로 변환하였다. IP-RTD로부터 입력 받은 2개의 센서값을 이용하여, 컨트롤러에서 처리한 후 IP-Digital 24를 이용하여 chamber에 2개의 제어신호를 주게 된다. 그림 2는 본 실험에서 사용한 컨트롤 시스템의 블록도를 보여준다.

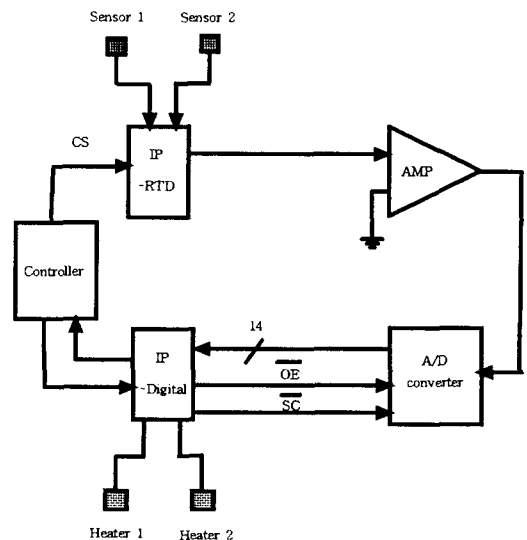


그림 2. 온도 제어를 위한 시스템 블록도  
Fig. 2. System block diagram for temperature control

### 2.4 시스템 개발 환경 구성

본 논문에서 구성한 전체 시스템은 VME 486 PC를 사용하여 Windows 운영체제를 구성하고, RS-232C cable을 통하여 MV-ME162 Embedded Controller Board를 제어하며, Ethernet cable을 통하여 데이터 전송을 한다.[1]-[3]

본 실험에서 MVME162 controller는 제어 시스템의 개발에 제어용과 모니터링을 위한 이상적인 플랫폼을 제공하게 구성되어 있다. MVME162는 Industry Pack과 연결되어, Local MC68

040 bus에 Industry Pack module의 I/O제어, 인터페이스, 아날로그와 디지털 함수들을 구성하여 시스템 환경을 이루었으며, 이 시스템으로 실시간 제어를 가능하게 구성하였다.

임계 실시간 제어를 위하여 VxWorks라는 Unix 체계의 OS를 사용하였다. VxWorks는 네트워크를 이용하여 MC68040의 CPU를 동작 가능하게 하고, testing, timing, debugging을 가능하게 하는 기능이 있으므로 본 실험에 사용하였다. 한번 개발이 완성되면 VxWorks 시스템은 독립되어 수행하거나 또는 VxWorks나 Unix를 수행하는 다른 시스템과 networking이 된다. 상호 개발 Host 시스템이므로 Unix를 편집, 컴파일, 링크, 그리고 실시간 코드를 저장하는데 사용하였고, 그것을 VxWorks상에서 디버깅하고 실행하였다. 개발 후에는 VxWorks의 host 시스템 또는 network 환경에서 쉽게 실행할수 있도록 ROM에 저장하여 필요시마다 단독으로 수행될 수 있게하였다.

IP-Digital 24로부터 출력되는 PWM 신호의 출력 펄스는 오실로스코프 상으로 볼 수 있으며 이 제어 신호는 chamber의 히터로 연결되어 있는 SSR(Solid State Relay)을 제어하고, 제어된 chamber내의 온도 분포는 chamber내에 고정시켜둔 온도 측정용 RTD 센서로부터 Chart Recorder 상으로 출력이 된다. 또한, 온도 제어용 RTD sensor로부터는 제어되는 온도값이 화면에 출력된다. 전체 시스템 구성은 그림 3과 같다.

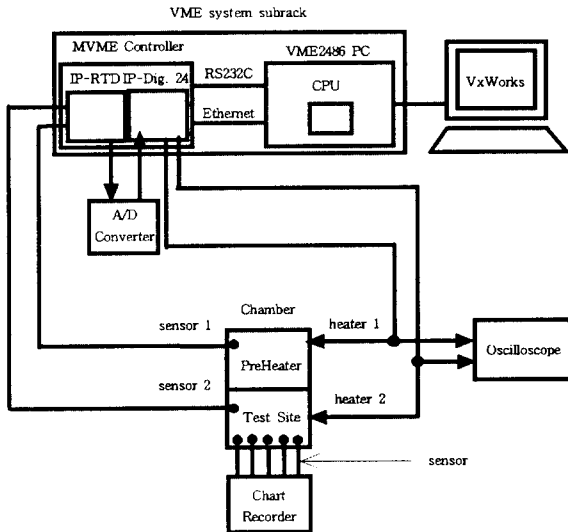


그림 3. 전체 시스템 구성도  
Fig. 3. System Block Diagram

### 3. 실험 및 고찰

본 실험에서 사용한 미래산업의 Test Handler MR-5100(수직식), MR-5300(수평식)은 후공정 장비로써 완제품의 반도체를 검사하는 장비이다. 테스트 과정은 PreHeter에서 예열된 반도체를 챔버내의 Test Site에서 32개의 반도체를 동시에 검사한다. 챔버내의 온도를 일정하게 제어하기 위해서는 32개의 각 지점에 Pt 센서를 모두 연결하여 온도를 읽어서 Chart Recorder상에서 그래프로 그려보는 실험을 하였다. 그러나, 기구학적 문제로 인하여 그림 4에 나타난 것과 같이 32개 지점을 모두 같게 하는 것은 차후의 문제로 여기고, 본 논문에서는 각각의 지점에서의 온

도 오차를 일정하게 하는 연구를 수행하였다.

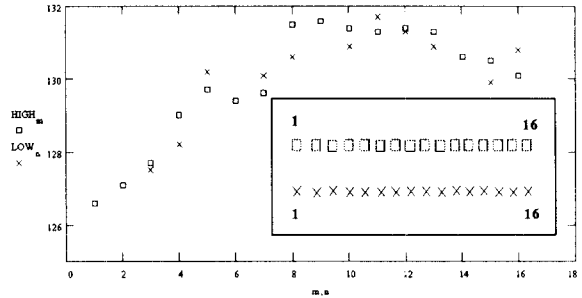


그림 4. 125°C에서의 챔버내의 온도 분포  
Fig. 4. Temperature distribution of chamber at 125°C

그림 4의 온도 분포는 챔버내의 온도를 125°C까지 올렸을때의 결과로써 챔버의 상단부와 하단부의 온도 오차가 1°C정도 발생하였고 챔버 외벽에서 발생하는 온도 손실 때문이다. 하지만, 본 실험에서는 챔버의 Test Site 부분을 Test 장비가 아닌 실험용 덮개를 사용하는데 기인한 오차도 큰 영향을 받은 결과인데, 그림 4에서 보면 가운데와 가장자리 부분이 6°C정도 차이가 나지만 실제로 Test Handler를 사용할 때는 챔버의 Test Site에 검사 장비를 장착했을때는 오차가 1°C ~ 2°C 정도로 아주 작아진다. 또한, 구조의 조정으로 이 오차는 더 작아질 수 있다. 하지만, 본 실험에서는 검사용 장비를 사용하지 않고 실험용 덮개를 사용하였다.

본 연구에서의 실험은 50°C, 80°C, 110°C까지 가열하여 일정한 온도를 유지하도록 하였다. 기존의 결과와 비교하기 위하여, 기존의 제어기를 사용하여서도 똑같은 실험을 반복하였으며, 그 결과는 그림 5에서부터 그림 8에 나타내었다. 실험 결과를 보면 본 연구에서 제시된 제어기가 오버슈트는 좀 큰 편이나(본 연구에서는 고려하지 않음) 테스트 핸들러에서 중요시하는 정상 상태에서 온도가 더욱 안정되게 유지되는 것을 볼 수 있다. 각 온도에서의 정상 상태에서의 오차를 비교해 보면 표 1과 같다.

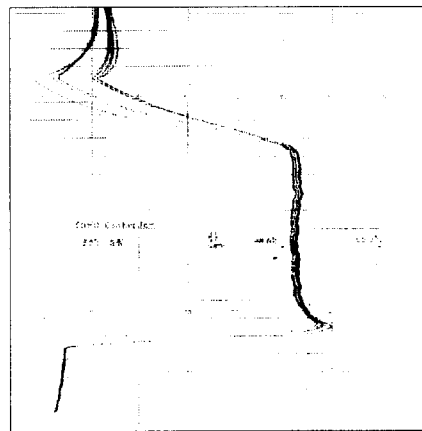


그림 5. 기존 제어기 50°C 설정  
Fig. 5. Traditional controller at 50°C

표 1. 기존의 제어기와 R.R.C. 제어기의 오차 비교  
 Table 1. Compare Traditional controller with R.R.C. controller at the point of fluctuation.

설정온도 \ 제어기종류	기존 제어기	R.R.C. 제어기
50℃	0.6℃이내	0.3℃이내
80℃	0.7℃이내	0.3℃이내
110℃	0.6℃이내	0.3℃이내

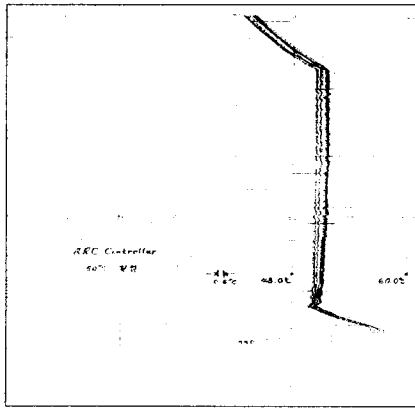


그림 6. R.R.C. 제어기 50℃ 설정  
 Fig. 6. R.R.C. controller at 50℃

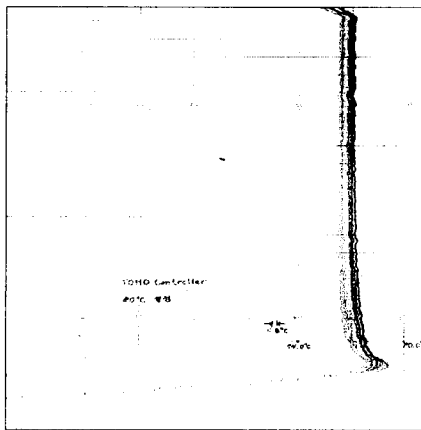


그림 7. 기존의 온도 제어기 80℃  
 Fig. 7 Traditional controller at 80℃

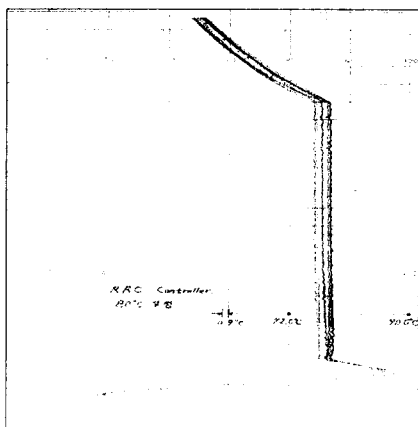


그림 8. R.R.C. 제어기 80℃ 설정  
 Fig. 8. R.R.C. controller at 80℃

#### 4. 결론

본 연구에서는 MC68040 CPU를 이용한 VME 시스템과 Industry Pack module을 이용하여 시스템을 구축하였고, A/D Converter 회로를 설계하였다. 그 결과로써 기존 온도 제어기의 최고치와 최저치의 온도 오차가 0.6℃이내에 비하여, R.R.C. 제어기는 0.3℃ 범위 내의 오차를 갖게하였다. 또한, 챔버내의 온도 분포를 파악함으로 인하여 챔버 성능을 개선하기 위한 방향을 제시하였고, 모든 프로그램을 모듈화하는 작업도 동시에 실시하여 다른 구조의 챔버에도 적용이 가능하게 하였다.

[후기]

본 연구는 한국과학재단 지정 호서대학교 반도체 제조장비 국산화 연구센터(R.R.C.)와 미래산업(주)의 지원하에 수행되었음을 밝힙니다.

#### 참고 문헌

- [1] VME-2486, User's Manual, Logical Design, Inc.
- [2] THE VMEbus SPECIFICATION, VMEbus international Trade Association
- [3] VxWorks Training Workshop, Wind River System, Inc.
- [4] 강기훈, 박종철, 박찬복, "온도", 한국표준연구소 정밀측정교재, 1985, 공업진흥청
- [5] 荒井由太郎 著, 김영혜 譯, "센서 인터페이스", No. 1, 1995, 기전연구소