



Or CAD의 특징과 인스톨 방법

李 根 喆

第一電算訓練院 院長

製圖는 電氣, 電子, 通信分野는 물론 電算分野에 이르기까지 한 번 정도는 경험하게 되는 작업이다.

수년전만 해도 回路를 제도하려면 우선 각종 製圖器具를 갖추어야 하고 템플릿과 같은 高價의 옵션 장치를 갖추어야만 回路를 그릴 수 있었다.

그러나 最近에는 각종 유용한 CAD 프로그램들이 많이 나와 회로를 쉽게 그릴 수 있게 되었는데 이 중에서 가장 많이 사용하는 프로그램이 Or CAD일 것이다.

보통 Or CAD라고 하는 프로그램은 크게 세 가지 즉, 電氣·電子回路를 제도할 수 있는 Or CAD/SDTIII와 회로를 시뮬레이션하는 Or CAD/VST, 그리고 基板設計를 수행하는 Or CAD/PCB로 분류할 수 있는데 보통 Or CAD라고 말하면 Or CAD/SDT 시리즈를 말하는 것이 대부분이다.

Or CAD/SDTIII는 모든 종류의 모니터와 프린터 및 플로터를 지원하므로 사용자가 표준 入出力裝置 이외에 특별한 專用 하드웨어를 사용할 필요가 없으며 IBM PC 및 그밖의 互換機種에 사용할 수 있도록 개발된 간편한 프로그램

이다.

1. Or CAD/SDTIII의 특징

Or CAD/SDTIII 프로그램은 回路圖를 제작할 수 있는 프로그램인 DRAFT, 그래픽 라이브러리(Library) 편집기라고 하는 LIBEDIT, 연결망 목록(Netlist), 設計點檢, 部品目錄 등을 도출해 내는 여러 가지 유틸리티(Utility)와 라이브러리들로 구성되어 있다.

한편 이들이 갖고 있는 모든 특징을 열거할 수 없으므로 요약하면 다음과 같다.

첫째, DRAFT는 회로도 작도, 편집 및 저장할 수 있는 주 프로그램으로서 특징을 보면 사용자가 본형(Template)의 치수(1mm의 해상력)를 지정하고 도선(Wire), 버스(Bus), 커넥터(Connector), 라벨(Label) 및 접합점(Junction)의 위치를 선정한다.

또한 사용자가 텍스트의 크기를 지정할 수 있고 부품의 회전 및 대칭, 작업회로의 자동적인 패닝(Panning), 5단계의 줌(Zoom), 무제한적인 계층회로의 단계, 문자탐색, 세로 텍스트 표시, DOS 명령을 수행하기 위한 일시적인 작업 보류 및 작업도면의 크기를 지정(A-E)한다.

둘째, 부품 라이브러리는 Or CAD/SDTIII에 가장 널리 이용되는 部品과 素子の 라이브러리를 포함하고 있으며 라이브러리에는 TTL, CMOS, MEMORY, ECL(Emitter Coupled Logic), 아날로그 및 마이크로 프로세서 등 거의 모든 소자를 포함하고 있다.

셋째는, 사용자 라이브러리의 작성으로서 이것은 사용자 자신이 기존과 고유의 라이브러리를 작성, 수정할 수 있다. 먼저 텍스트 편집기를 이용하여 라이브러리 원시(Source) 파일을 만드는 방법이 있는데 원시 파일이란 Or CAD 기호 기술언어(Symbolic Description Language)의 명령어를 포함하는 ASCII 텍스트 파일이다.

라이브러리 원시 파일에 Composer라고 하는 유틸리티 파일을 실행하여 Or CAD/SDTIII가 읽을 수 있는 라이브러리 데이터 파일을 만들 수 있다.

한편 그래픽 목적 편집기인 LIBEDIT는 사용자가 원할 경우 새로운 IEEE/ANSI 표준기호를 쉽게 만들 수 있으며 그래픽 화면기호를 자동적으로 Or CAD/SDTIII가 읽을 수 있는 데이터 형식으로 변환시켜준다.

Or CAD/SDTIII의 융통성은 회로설계과정이 끝난 후 다양한 유틸리티 프로그램을 쉽게 실행하는데 있으며 그 내용을 열거하면 다음과 같다.

ANNOTATE, BACKANNO, CLEANUP, COMPOSER, CROSSREF, DECOMP, ERC, LIBARCH, LIBEDIT, NETLIST, PARTLIST, PLDTALL, PRINTALL, TREELIST.

2. Or CAD 설치환경

실제로 Or CAD를 시스템에 설치하기 전에 소프트웨어와 하드웨어적인 구성을 보면 7개의 Or CAD/SDTIII 소프트웨어 디스켓이 필요하

다. 즉

- Master Software Disk 1
- Master Software Disk 2
- Library Disk 1
- Library Disk 2
- Driver Disk 1
- Driver Disk 2
- Driver Disk 3

이외에 필요한 하드웨어 사항으로는

첫째, 두 개의 360KB FDD 또는 HDD를 갖춘 IBM PC 및 호환기종 또는 두 개의 720KB FDD 또는 HDD를 갖춘 IBM PS/2 및 호환기종,

둘째, DOS 버전 2.0 이상이며 최소한 512KB 이상의 메모리,

셋째, Or CAD/SDTIII의 백업 복사본을 만들기 위한 포매팅된 7장의 공 디스켓,

넷째, 두 개의 FDD의 경우 작업 디스켓을 만들기 위한 5장의 공 디스켓 등을 들 수 있다.

다음에 Or CAD/SDTIII를 설치하는 경우 두 개의 FDD 또는 HDD 시스템에 설치할 수 있다.

2·1 두 개의 FDD 시스템인 경우

프로그램의 보호를 위하여 백업 복사본을 이용하여 설치해야 하며 프로그램을 실행할 경우에도 백업 복사본을 이용하여야 한다.

5개의 디스켓에 Or CAD/SDTIII의 작업 디스켓을 만드는데 이것은 드라이버와 라이브러리 및 작업도면 파일을 로드하는데 필요한 것이다. 5개의 디스켓 포시는 Draft 1, Draft 2, Sheet, Driver/Library 및 Library Utility이다. 먼저 Master Software Disk 1과 2로부터 Draft.EXE와 Or CAD SDT.OVL을 작업 디스켓 Draft 1에 복사한다. 실행 파일은 파일명 뒤의 擴張子が EXE이고 오버레이(Overlay) 파일은 확장자가 OVL이다.

PDRAFT.OVL, PSPICE.BAT 와 필요한 매크로 파일(확장자 MAC)을 Master Software Disk 1과 2에서 작업 디스켓 Drfat 1에 복사한다.

다음에 Master Software Disk 1과 2로부터 ANNOTATE.EXE, CLEANUP.EXE, ERC.EXE, PARTLIST.EXE 그리고 TREELIST.EXE 등 유틸리티 프로그램을 복사한다. 필요에 따라서 유용한 유틸리티 프로그램을 복사할 수도 있다.

그러나 모든 유틸리티 프로그램을 DRAFT 1에 저장시킬 수 없으므로 작업 디스켓 DRAFT 2에 BACKANNO.EXE, CROSSREF.EXE, NETLIST.EXE, PRINTALL.EXE, PLOTALL.EXE, Or CAD SDT.OVL 등을 복사한다.

여기서 프로그램의 실행중에는 똑같은 오버레이 파일을 가지고 있어야 한다.

작업 디스켓 Driver Library에 PRINTER.DRV를 Driver Disk로부터 복사하고 프린터와 플로터 그리고 모니터를 위한 드라이버 파일을 필요에 따라 복사한다. 또한 Library Disk에서 목적 라이브러리(확장자 LIB)를 Driver Library 디스켓에 복사한다.

끝으로 Master Software Disk에서 LIBARCH.EXE, LIBEDIT.EXE, COMPOSER.EXE, DECOMP.EXE 그리고 Or CAD SDT.OVL을 작업 디스켓 Library Utility에 복사한다.

이 디스켓은 사용자 부품 라이브러리를 만들거나 기존의 라이브러리를 수정할 때 사용된다.

2·2 HDD 시스템인 경우

Or CAD/SDTIII를 유용하게 사용하는 하나의 방법은 HDD에 일련의 서브디렉터리(Sub-directory)를 만들어 사용하는 것이다. 이것은 사용자가 드라이버, 라이브러리 그리고 작업도

면을 보다 쉽게 사용할 수 있다.

이것은 HDD에 Driver, Library, Sheet의 3개 서브디렉터리를 가진 새로운 Or CAD 디렉터리를 만드는 것이다.

그리고 Or CAD Master Software Disk의 모든 파일을 Or CAD 디렉터리에 복사하는데 이것을 루트(Root) 디렉터리라고 한다.

다음에 Driver Disk의 모든 파일을 Drvier 서브디렉터리에 복사하고 Library Disk의 모든 파일을 Library 서브디렉터리에 복사한다. 마지막으로 Sheet 서브디렉터리는 회로작업 도면 파일을 저장하기 위하여 비워둔다.

마우스의 설치에 있어서 Or CAD/SDTIII는 특별한 마우스를 지원하지 않으며 일반적인 마우스의 지원을 받는데 MOUSE.COM이나 MSMOUSE.COM 등의 파일을 실행해야 마우스를 사용할 수 있다.

특별히 마우스를 사용할 때는 직렬 포트(COM 1 또는 COM 2)를 /1 또는 /2로 지정하여 사용한다. 여기서는 HDD나 FDD의 구성정보는 생략하기로 한다.

한편 Or CAD/SDTIII의 구성 파일 수행은 DRAFT.EXE와 Or CAD/SDT.OVL 및 유틸리티 프로그램이 있는 Or CAD 디렉터리에서 실행하여야 하며 만일 FDD를 사용할 때는 드라이브 A에 DRAFT 1 디스켓을, 드라이브 B에 Driver/Library 디스켓을 넣고 실행한다.

각각의 경우 DOS 프롬프트에서 다음과 같이 입력한다.

```
DRAFT/C <ENTER>
```

/C는 구성 메뉴를 화면상에 표시하는 스위치인데 Or CAD/SDTIII의 구성을 변경할 때만 /C를 사용한다.

본고에서는 평면 파일, 계층 파일, 단일도면 파일의 구조 및 Or CAD/SDTIII에서 작업도면의 파일을 로드, 저장, 프린트하는 것과 키보드와 마우스에 의한 Draft의 호출, 명령어의 실

행은 지면 관계상 생략하기로 한다.

3. 시뮬레이션

Or CAD/VST에서 VST는 Verification & Simulation Tools의 의미로서 PC 수준에서는 회로도 작성이나 PCB 설계와는 달리 아직 시뮬레이션이 제대로 수행되지 않고 있다.

PC에서는 게이트 수준으로만 설계된 로직회로를 시뮬레이션을 통하여 검증하려고 한다면 별다른 문제 없이 해낼 수 있겠지만 마이크로 프로세서나 메모리 素子が 들어 있는 회로의 로직 시뮬레이션을 한다면 상당히 어려운 문제에 부딪히게 된다. 특히 메모리칩의 시뮬레이션이라면 더욱 그러하다.

예를 들어 640KB 크기의 메모리 회로를 시뮬레이션한다면 실제로 PC에서도 이만큼의 메모리가 필요하게 된다.

마이크로 프로세서에서는 모든 핀에 관한 입력과 출력의 정보를 모두 입력시켜 주어야 하는데 PC로서는 이를 제대로 감당해 내기가 힘들다.

하지만 이같은 문제에도 불구하고 비교적 작은 회로부의 검증을 하기 위해서라면 Or CAD/VST와 같은 TOOL도 아주 유용하게 쓰일 수 있다. 가령 마이크로 프로세서와 입출력 포트와의 인터페이스를 TTL 소자를 사용하여 구성하고자 한다면 이 인터페이스 회로부만은 손쉽고 정확히 시뮬레이션해낼 수 있는 것이다. 어쨌든 이를 적절히 활용하는 것이 필요하다.

4. 네트리스트

Or CAD/SDTIII는 기능에 있어서는 비교적 간단한 정도에 지나지 않지만, 전자회로를 작성하는 소프트웨어로서의 어지간한 기능은 다 갖추고 있으며 무엇보다도 사용법이 간단하다는 특징을 가지고 있기 때문에 아마추어 뿐만 아니라 전자회로 개발을 직업으로 삼는 전문가들도

상당히 많이 사용하고 있다. 전자회로 설계시에 는 거의 대부분의 경우에 Or CAD를 사용하고 있다.

Or CAD/SDTIII은 사용법도 쉬울 뿐만 아니라 일단 그린 회로도들 가지고 PCB 설계를 할 때에도 P-CAD나 CADSTAR 등과 같은 고급 PCB 설계 카드 소프트웨어와 쉽게 연결이 될 수 있도록 배려가 되어 있다. 물론 Or CAD/PCB 프로그램을 사용하여 PCB 설계를 할 수도 있지만 어느 정도 수준 이상의 설계가 되려면 적어도 P-CAD나 CADSTAR 이상의 소프트웨어를 사용해야 가능하다. 따라서 여기에서는 Or CAD로 그린 회로도를 PCB 설계용 소프트웨어로 옮기는 방법에 대해서 잠시 설명하기로 하겠다.

Or CAD로 회로도를 그리고 나서 데이터를 PCB 카드 소프트웨어로 넘겨주기 전에 거쳐야 할 과정이 있다. 우선 Clean-Up 프로그램을 돌려서 회로도상에 부품이나 연결과정에서의 중복된 점이 있는가를 확인하고 바로 잡아야 한다. 그런 다음에 ERC(Electrical Rules Check) 프로그램을 실행시켜서 그 회로상에 기본적인 전기적 규칙을 위배한 부분이 있는지를 검증해야 한다. 이때 흔히 에러 메시지가 뜨게 되는 것은 연결이 안된 핀이 있을 경우, 입력과 입력 혹은 출력과 출력 신호끼리 연결한 경우, 그리고 버스 신호중의 라벨이 적혀지지 않은 경우 등인데 이들 메시지들을 일일이 확인해 보고 문제가 없을 때에만 그 다음 과정으로 넘어가야 한다.

이런 검증을 통해 완전한 회로도가 완성된 뒤에야 비로소 이 회로도로부터 네트리스트(Netlist)라는 것이 문자 그대로 그물처럼 이리저리 연결된 회로도의 리스트를 만드는 일이라는 것을 알게 된다. 이때 회로도에는 어떤 부품이 있으며 그 부품의 몇 번 핀은 각각 연결이 되어 있는지의 정보가 그 네트리스트에 들어간

다. 만약 회로도도 부정확한 것이라면 당연히 네트리스트도 정확한 것이 될 수 없다. 네트리스트는 일반적인 아스키 화일로 만들어지므로 쉽게 보고 확인하거나 변경할 수 있다. 그리고 그 형식만 알고 있다면 회로도도 그리지 않고도 직접 네트리스트를 에디터로 작성하여 캐드 프로그램에 넘길 수도 있다.

Or CAD/SDTIII에서 네트리스트를 만들기 위한 명령은 다음과 같은 형식으로 내린다. 사용하려는 PCB 설계 소프트웨어는 CADSTAR 이고 완성된 회로도의 화일명이 XXX이며 여기서 뽑아진 네트리스트의 화일명이 YYY가 되게 하려면 다음과 같은 명령을 내린다.

NETLIST XXX YYY/SRACALREDAC
또는

NETLIST XXX YYY RACALRECAC/S
여기서 /S는 네트리스트 프로그램을 제어하기 위한 스위치라고 부르며, 사용자가 정의하고자 하는 특정 네트리스트 형식을 지정하기 위하여 쓰인다. RECALREDAC라는 것은 CAD-STAR를 개발한 회사인 Racal-Redac을 가리킨다. 만약 P-CAD용의 네트리스트를 만들려고 한다면,

NETLIST XXX YYY PCAD/S
라고 하면 될 것이다. Or CAD/SDTIII에서는 P-CAD와 CADSTAR 방식을 포함해서 20가지 이상의 네트리스트 형식을 지원하고 있는데, 이들 중에는 로직 시뮬레이션을 하기 위한 형식도 있다.

만약 RACALREDAC/S라는 것을 쓰지 않고 나머지만 가지고 명령을 내리면 YYY 화일의 네트리스트 형식은 EDIF 타입이 될 것이다. EDIF라는 것은 Electronic Design Intermediate Format의 약자로서 전자공업 전반에 걸친 표준을 가리키며, 컴퓨터를 사용한 설계에 있어서 어떤 중립적인 형식을 취하는 것이다. 이 형식을 사용하면 서로 다른 종류의 시스템 사이에

서도 전자설계 데이터를 교환할 수 있게 된다. 하지만 아직 전반적인 표준으로서 자리잡지는 못했다.

일단 이렇게 네트리스트를 만들어도 이것을 캐드 소프트웨어에서 당장 사용하지 못하는 경우가 많다. 그것은 각 소프트웨어의 버전이 차이가 날 수 있는 데다가 Or CAD에서 완벽한 지원을 해주지 못하는 경우도 있기 때문이다. 이런 문제가 발생한 경우에는 어쩔 수 없이 에디터를 사용하여 일일이 수정을 해주어야 하는 불편을 감수해야 한다.

5. PCB

회로 설계에 있어서의 각 용어도 그렇지만 PCB에 관한 것은 더욱 일반적인 것은 아니다. 그래서 캐드 소프트웨어의 사용 매뉴얼을 보거나 프로그램을 실행시킬 때에도 제대로 모르고 하게 되는 경우가 종종 있음을 볼 수 있다. 여기서는 PCB 설계과정에서 사용되는 기본적인 용어에 대해서 몇 가지 알아보기로 한다.

PCB라는 말은 Printed Circuit Board의 첫자만 따서 만들어진 단어이다.

가장 싸고 간단한 PCB가 단층 기판(Single-Sided PCB)이다. 단층 기판의 위쪽은 단지 부품을 고정하기 위해서 사용되고 아래쪽은 회로의 배선을 위해서 사용된다. 부품의 연결은 보드를 관통해 뚫려있는 구멍을 이용해서 이루어진다. 이 구멍을 홀(Hole)이라고 부르며 이 구멍의 아래쪽은 납땜을 하여 부품을 고정시키고 리드선을 전기적으로 연결하는 등의 그런 영역이 그 구멍을 둘러싸고 있다. 이것을 패드(Pad)라고 부른다.

단층 기판의 재질은 보통 베이크라이트라고 부르는 재질인데 반하여 양면 이상의 기판의 재질은 에폭시이다. 양면 기판은 영어로는 Double-Sided PCB, 또는 Both-Layered PCB라고 부른다. 그 이상의 층을 가진 기판은 4,

6, 8, 10층 등으로 짝수 개의 층을 가지며, 이들을 다층 기판(Multi Layer PCB)라고 부른다. 이렇게 층수가 많아지면서 32층의 기판까지 제작되기도 하는데 그렇다고 해서 기판의 두께가 더 두꺼워지는 것은 아니다. 흔히 보는 단층 기판의 두께나 고성능 전자기기에서 사용되는 10층 정도의 기판의 두께는 다같이 1.6mm이다.

다층 기판의 각 층들은 나름대로 회로를 연결하여 쓸 수 있지만 실제로는 한 층씩 걸러가면서 VCC 및 GND만 연결되는 층이 있다. 이것은 회로요소간의 간섭을 최소화하고 노이즈의 영향을 최소화시키기 위한 것이다. 가령 4층 기판에서라면 가장 위층과 가장 아래층은 일반 회로 연결을 위해 사용되고, 안쪽에 들어 있는 두 개의 층은 각각 VCC 및 GND 신호로만 잘리게 된다. 이들을 각각 전원 플레인(Power Plane), 그라운드 플레인(Ground Plane)이라고 부른다.

다층 기판에서의 패드는 단지 그 층에서만 전기를 통하는 도체일 뿐이지만 다층 기판에서의 패드는 다른 모든 층과 연결이 될 수 있으며 부품의 리드선이 관통하게 되어 있는 홀을 통해 이루어진다. 이런 홀은 홀의 안쪽 벽이 구리로 싸여 있기 때문에 전기적인 연결이 되는데 이것을 THP(Trough Hole Plate)라고 부른다. 보통 간단히 스루 홀(Through Hole)이라고 말하기도 한다.

다층 기판에서는 THP가 항상 부품의 고정을 위해서만 있지는 않다. 어떠한 층에서 회로연결을 하려고 할 때 사방이 다른 연결선들로 막혀 있을 때에는 다른 층으로 옮겨서 그 연결을 완성하게 된다. 이런 목적을 위해 마련되는 THP를 비아(Via)라고 부른다. 일반적으로 비아의 구멍은 패드의 구멍보다는 구멍의 크기가 더 작게 된다. 어떤 한 층에서 패드나 비아 등이 서로 전기적인 연결이 되게 하려면 얇은 구리판으로 각 지점 사이를 이어주게 된다. 이런 것을

흔히들 패턴이라고 부르지만 정확히는 트랙(Track)이라는 용어가 적당할 것이다. 주위의 회로기판을 자세히 보면 표면이 대개는 초록색 계통으로 보일 것이다. 그러나 기판의 단면을 칼로 살짝 끊어보거나 모서리 등을 보면 그것이 원래는 베이지색 쪽임을 알 수 있다. 또 트랙도 그냥 보아서는 초록색으로 보이지만 칼로 살짝 끊으면 위의 철이 벗겨지면서 구리색이 드러난다. 이렇게 기판의 양쪽에 칠하는 잉크를 솔더 마스크(Solder Mask)라고 부른다. 솔더 마스크가 칠해진 곳에는 납이 묻지 않는다. 전체 기판 중에서 솔더 마스크가 칠해지지 않는 곳은 부품이 고정되는 패드 정도인데, 이 곳은 납땀이 잘 되어야 하므로 솔더 마스크를 칠하지 않는 것이다.

6. PCB 캐드 소프트웨어

가장 구하기 쉬우며 또 사용하기도 쉬운 것은 역시 Or CAD/PCB이다. 비교적 새로운 버전인 PCBII는 일반 AT 수준의 PC에서 사용할 때 270개 이상의 14핀짜리 IC, 6천개 이상의 패드, 그리고 1만6천개의 트랙을 다룰 수 있으며 최고 14핀 기판까지 설계할 수 있다. 이 정도라면 어지간한 PC용 확장 보드 수준 정도는 문제 없이 기판 설계를 해낼 수 있을 것이다. 최근에 판매되기 시작한 릴리즈IV는 이전의 버전보다 기능상에서 상당히 개량이 이루어졌다고 한다. 현재 사용하는 것은 CADSTAR로서 속도면에서나 효율성면에서 다른 소프트웨어보다 비교적 우수한 편이지만 한 가지 단점은 코프로세서가 있어야 한다는 것이다. 그리고 지원되는 그래픽 보드가 몇 가지로 한정되어 있는 점도 불편한 점이라고 할 수 있다.

PC 수준에서 운용될 수 있는 PCB 설계용 캐드 소프트웨어 가운데 비교적 널리 알려진 것으로서는 위의 Or CAD/PCB, CADSTAR 이외에도 P-CAD, Tango, EE-Designer, PADS-

PCB 등이 있다. 최근에는 마이크로소프트 윈도우즈상에서 구동되는 PCB 설계 소프트웨어가 등장하기도 했다.

그 반면에 매킨토시에서 운용되는 PCB 캐드는 PC처럼 많지는 않다. 하지만 더글라스 일렉트로닉스의 Douglas CAD/CAM Professional System과 뱀프(Vamp)사의 Mc CAD Electronic Design System 등은 둘 다 PC에서 돌아가는 캐드 소프트웨어 수준의 기능을 보유하고 있다.

7. 개발장비

디지털 회로를 설계하는 단계에서라면 단지 컴퓨터만 사용하면 되겠지만 실제로 개발에 들어가게 되면 그밖에도 계측장비가 필요하게 된다. 가장 기본적인 것으로는 테스터와 오실로스코프를 들 수 있다. 오실로스코프에도 여러 가지 종류가 있는데 요즘에는 측정된 신호를 마이크로 프로세서에 의해서 처리하여 보다 다양한 정보를 알려주는 디지털 오실로스코프가 많이 사용되고 있다. 이것을 사용하면 측정된 신호의 모습을 저장할 수도 있고 또 프린터로 하드카피해낼 수도 있는 한편 신호의 여러 가지 특징을 수치적으로 볼 수도 있다. 물론 가격은 일반적인 아날로그 오실로스코프보다는 비싼 편이다.

아마추어로서는 이같은 오실로스코프만해도 돈 생각이 나겠지만 실제로 하드웨어의 개발을 위해서는 이것만으로는 개발이 힘들다. 디지털 회로에서 특히 중요한 것이 회로에서의 각 신호들이 서로 타이밍이 잘 맞느냐에 대한 확인이라고 하겠는데 이것은 IC 칩 규격표의 타이밍 차트를 준수하는지의 여부를 검사함으로써 이루어진다. 이런 검사를 하는 계측기가 로직 분석기(Logic Analyzer)이다. 이것은 제품에 따라 달라지지만 한 번에 수십 개 이상의 신호를 분석해낼 수 있는 능력을 가진다. 로직 분석기를

사용하면 마치 규격표의 타이밍 다이어그램에서처럼 회로가 동작되는 모습을 눈으로 확인할 수 있는 것이다.

만약 컴퓨터 자체를 만들든지 아니면 마이크로 프로세서를 사용한 시스템을 만들려고 한다면 ICE(In-Circuit Emulator)라는 장비가 필요하다. 이것은 마이크로 프로세서를 대신하여 그 기능을 수행해 주는데, 기계어 코드를 마음대로 뜯어고칠 수도 있고 적당한 조건을 설정해서 그 조건이 만족하는 상태가 되면 프로그램이 일시적으로 정지할 수도 있는 등의 기능이 있으며, 이밖에도 마이크로 프로세서를 사용한 시스템 개발에 있어서 아주 강력한 기능들을 가지고 있다. 이를 사용하면 마치 마이크로 프로세서의 내부를 개발자의 눈으로 보면서 맘대로 뜯어고칠 수 있는 것과 같다고 할 수 있다.

이처럼 비싼 장비들을 구입할 수 없는 아마추어들에게 가장 바람직한 방법은 자신이 직접 위의 장비들을 대신 할 회로들을 만들어 쓰는 것이다. 우선 필요한 것은 롬 에뮬레이터(ROM Emulator)이다. 좋은 개발장비가 없는 상태에서 마이크로 프로세서를 사용한 전자회로를 만들면서 프로그램을 고치는 일은 상당히 귀찮고 또 시간을 낭비하는 일이다. EP-ROM에 한 번 프로그램을 써넣었다가, 조금 고치려고 하면 소켓에서 빼서 자외선 소거기에 넣어 몇 분 동안 지우고 다시 롬 라이터로 프로그램을 써넣는 등의 일을 반복하기 때문이다. 이때 EP-ROM을 사용하는 대신에 이 롬 에뮬레이터를 사용하면 일일이 롬을 지웠다 썼다 할 일이 없어진다. 단지 컴퓨터를 사용하여 필요할 때마다 프로그램 화일을 롬 에뮬레이터에 다운로드하면 간단히 프로그램을 교체해서 실험할 수 있다.

이외에도 자신이 만들어서 다른 회로를 개발할 때 사용할 수 있는 기기로는 롬 라이터(ROM Writer), 로직 테스터(Logic Tester) 등과 같은 것을 들 수 있다.