

PC와의 인터페이스



오 준 호
한국과학기술원 생산공학과 교수

●1954년생
●디지털 제어, 디지털/아날로그 신호처리, 마이크로프로세서응용, 적응제어, 로봇틱스, 기타 제어이론 및 응용에 관심을 가지고 있다.

1. 머리말

인터페이스(interfacer)라함은 성격이 다른 두가지 이상의 장치를 연결시켜 정보의 교환을 가능하게 하는 공유영역을 의미한다. 일례로 사람(human)과 컴퓨터간의 정보교환을 가능하게 하는 인터페이스는 자판(keyboard)과 모니터라 할 수 있다. 즉 사람은 자판을 통하여 컴퓨터에 원하는 명령을 입력시키고 그 결과를 모니터 스크린 혹은 프린터 출력등을 통하여 확인하는 것이다.

이와같이 성격이 전혀 다른 장치간에도 정보의 교환이 가능하며, 이를 구현하는 기술이 인터페이스 기술이다. PC와 외부 장치와의 인터페이스 방법에는 I/O(입·출력)포트 방식, 메모리 맵 방식, 인터럽트 방식, 직접 메모리 접근(DMA)방식등이 있으며 이들은 각기 고유의 목적에 따라 사용되고 있다. 또한 각 방식마다 사용하기 적합한 전용 칩(chip)들이 상용화 되고있다.

메모리 맵 방식은 메모리의 한 영역을 외부 장치로 간주하여 I/O 를 수행하는 방법이며 포트 I/O 방식은 메모리 공간과 별도의 영역을 설정, 이를 통하여 I/O를 수행하는 방법이다. 두가지 방식에는 장단점이 있으나 기존 메모리 공간과 간섭을 일으키지 않고 별도의 공간을 확보할 수 있는 포트 I/O방식이 널리 쓰

이고있다. 이미 언급한 두 가지 방식은 I/O를 위한 타이밍 및 제어를 CPU(central processing unit)가 관장하고 있는 반면, DMA는 DMA 제어가 메모리 I/O 주기를 관장하여 외부로 부터 직접 메모리 접근을 가능하게한 방법이다. 이는 주로 다량의 정보를 단시간내에 I/O 할 경우 매우 유용하며 컴퓨터 내부에서 하드디스크와 메모리간의 데이터 입·출력 등의 통신에서도 사용되고 있다.

본 글에서는 이해하기 쉬우며 응용 범위도 넓은 I/O포트 방식의 인터페이스를 중심으로 작동원리, 회로설계에, 제작및 실험에 관하여 간략하게 살펴보기로 하겠다. 본 글중 PC라함은 IBM PC/XT 및 호환 기종을 일컬으며, 여기에 언급되는 기술은 XT급뿐 아니라 AT급의 PC에도 무리없이 적용될 수 있다.

2. PC의 시스템 버스 및 타이밍

PC의 시스템 버스는 PC 내부의 확장 슬롯에 노출되어 있어 확장 보드를 사용함으로써 접근이 가능하다. PC의 버스는 총 62개로서 이들은 다음에서 보이는 바와 같이 그 기능을 구분할 수 있다. 확장 슬롯의 버스 단자 배열표는 그림 1과 같다.

어드레스 버스 : A0~A19

데이터 버스 : D0~D7

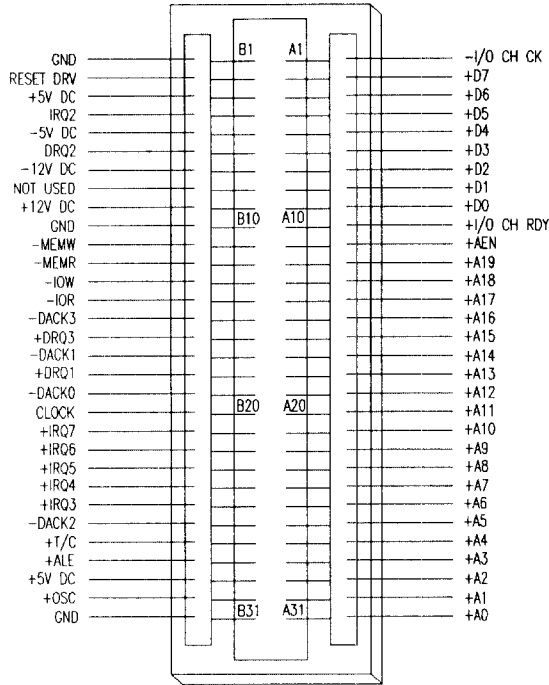


그림 1 PC의 시스템 버스 확장 슬롯의 단자배열

- 콘트롤 버스 : $\overline{\text{IOR}}$, $\overline{\text{IOW}}$, $\overline{\text{MEMR}}$, $\overline{\text{MEMW}}$, ALE, AEN, 등
- DMA 버스 : DACK, DRQ, T/C
- 인터럽트 버스 : IRQ2~IRQ7
- 기타 버스 : OSC, CLK, RESET, 등
- 버스 파워 : GND, $\pm 5V$, $\pm 12V$

이들중 I/O 포트 방식에 사용되는 버스에 대하여 중점적으로 살펴 보기로 하겠다.

2.1 어드레스 버스

어드레스 버스는 A0 부터 A19까지의 20비트로 구성되어 있어 약 10^6 (1M)의 독립적인 위치를 지시할 수 있다. 이 어드레스 버스는 포트 입·출력시, 메모리 주기 수행시 모두 공용으로 사용되고 있다. 메모리 주기 수행시에는 $\overline{\text{MEMR}}$, $\overline{\text{MEMW}}$ 신호와, 그리고 포트 입·출력시는 $\overline{\text{IOW}}$, $\overline{\text{IOR}}$ 신호와 병행하여 사용되

어야 한다. 포트 입·출력 방식에서는 20비트의 버스중 처음 16비트(A0~A15)까지 사용이 가능하다, PC의 포트 I/O 설계에서는 주로 처음 10비트(A0~A9)까지가 많이 이용된다.

2.2 $\overline{\text{IOR}}$ (I/O Read)버스

$\overline{\text{IOR}}$ 버스는 입력 명령에 의하여 어드레스 버스가 포트 I/O 혹은 DMA시 입력 어드레스를 지칭할때 "L"* 으로 작동함으로써 데이터의 입력 시점(timing)을 결정하여 준다. 그림 2에서 보이는 바와 같이 "valid port address"기간 중 포트 어드레스가 안정화되는 중간 시간동안 "L"으로 작용하며 다시 "H"로 상승하기 직전(약 30 ns)에 데이터 버스에 지시된 값을 읽어 들이게 된다.

*"L"과"H"는 논리 표시로서 0와 1. 트랜지스터 논리(TTL)신호로는 0V와 5V를 각각 의미한다.

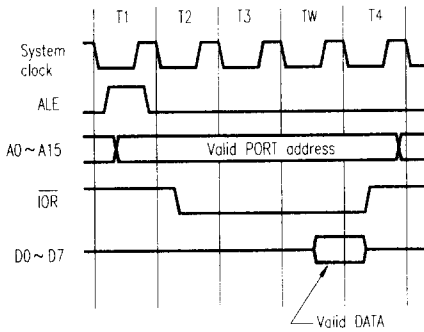


그림 2 포트 I/O 입력 주기시 타이밍선도

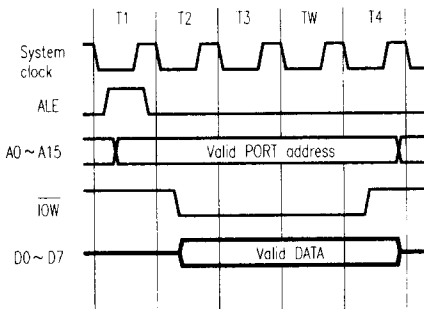


그림 3 포트 I/O 출력주기시 타이밍선도

2.3 IOW(I/O Write)버스

IOW버스는 출력 명령에 대하여 어드레스 버스가 포트 I/O혹은 DMA시 입력 어드레스를 지칭할 때 “L”으로 작동함으로써 데이터의 출력 시점을 결정하여 준다. 그림 3에서 보이는 바와같이 출력될 데이터(“valid data”)는 IOW상승 에지(rising edge)부근에서 “valid”하여진다. 그러므로 외부장치는 IOW의 상승 에지에서 유용한 출력 데이터를 얻을 수 있다.

2.4 AEN 버스(Address Enable)

AEN 버스는 DMA 제어기 주기동안 “H”로서 작동한다. 전항에서 언급한 바와 같이, IOW, IOR신호는 DMA와 포트 I/O주기동안 겸용으로 작동하므로 포트 I/O시에는 AEN 버스가 “L”인 것은 확인하여야 DMA 주기와의 혼선을 피할 수 있다.

2.5 RESET 버스

이 버스는 8088에 처음 전력이 가해 졌을 때 (power-on) CPU 내부의 모든 전압이 안정화 될 때까지 “H”로 작동한다. 즉 RESET 버스는 “power-on”시 최초로 발생하는 매우 짧은 펄스이다. 이 신호는 정상 작동중이라도 전력 레벨이 적정 범위를 벗어나면 다시 “H”로 된다. 이 신호는 주로 “power-on”시 주변 칩 혹은 장치등을 초기화(reset)시키는 신호로 유용하게 사용된다.

3. 포트 I/O방식에 의한 평행 I/O (Parallel I/O)설계

포트 I/O 방식에 의한 평행 I/O 설계 예를 그림 4에서 보이고 있다. 이의 작동을 살펴보면 다음과 같다.

3.1 어드레스 해독기(Address Decoder)

칩 SN74LS688은 일종은 논리 비교기(logic comparator)로서 P0~P7 단자의 입력과 Q0~Q7 단자의 입력을 비트단위로 비교한 후 그 내용이 같으면 $\overline{P=Q}$ 단자가 “L”이 된다. 그림 4의 경우의 해독기는 A3~A9와 AEN 신호를 해독하도록 설계되어 있다. PC의 경우 포트 I/O공간은 1024개(10비트)로서 구성되어 있으며 이는 다시 전반부(0~511)와 후반부(512~1023)로 논리적으로 구분되어 있다.

이는 MSB(most significant bit)인 A9에 의하여 결정된다. 즉 A9 이 “L”이면 0~511 까지, “H”이면 512~1023 까지를 지칭하게 된다. IBM PC 및 호환 기종에서는 컴퓨터 내부에서 사용되는 표준 I/O 장치에 대하여 이미 0~511 까지를 미리 확보해 놓고 있으므로 이들과의 간섭을 피하기 위하여는 A9을 “H”로 하여(Q0를 +5V에 연결함)후반부 영역을 사용하는 것이 안전하다. 후반부 영역 역시 표준 외부 장치에 해당되는 영역을 피하는 것이 유리하다. 표준 외부 장치에 지정된 포트 어드레스는 표 1에 나타내었다. 또한 AEN신호는 포트 I/O시 항상 “L”이어야 하므로 이에 대응되

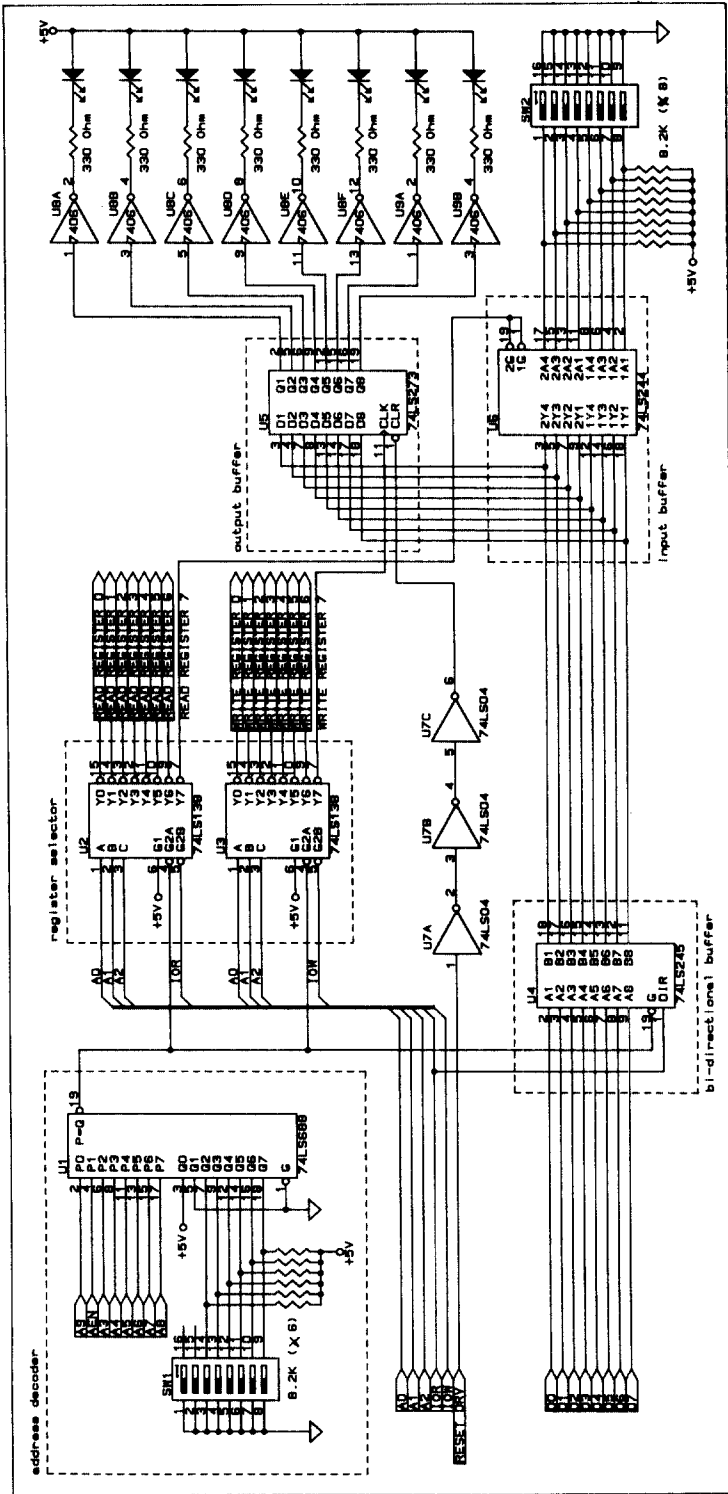


그림 4 포트 I/O 방식에 의한 평행 I/O(parallel I/O)실계 예

표 1 외부 확장용 I/O 포트의 맵(map)

I/O번지		용 도
16진수	10진수	
000-1FF	0-511	Base system board에서 사용
200	511	사용 안됨
201	513	Game control
202-277	514-631	사용 안됨
278-27F	632-639	Second print adapter
280-2F7	640-759	사용 안됨
2F8-2FF	760-767	COM2
300-377	768-887	사용안됨
378-37F	888-895	Printer port adapter card
380-3AF	896-943	사용 안됨
3B0-3BF	944-959	Monochrome and printer adapter
3C0-3CF	960-975	사용 안됨
3D0-3DF	976-991	Color/graphics adapter
3E0-3F7	992-1007	사용 안됨
3F0-3F7	1008-1015	5 1/4 inch diskette drive adapter card
3F8-3FF	1016-1023	COM1

는 비교 단자 Q1을 0V(ground) 시켜주어야 한다.

3.2 레지스터 셀렉터(Register Selector)

어드레스 해독기는 A0~A9까지의 어드레스 버스중 상위 7비트(A3~A9)까지만을 해독하고 있으므로 하위 3비트(A0~A2)는 멀티플렉서에 의하여 해독되도록 하였다. SN74LS138은 A, B, C 단자에 인가된 2진수를 8진수로 변환한 후 이에 해당하는 출력 단자(Y0~Y7)만을 "L"으로 만들어 주는 역할을 한다. 또한 G2A와 G2B 입력 단자에 의하여 출력 단자(Y0~Y7)이 일률적으로 제어되도록 되어 있다. 제어방식은 $G2A \oplus G2B$ (여기서 \oplus 는 논리합: OR)의 결과가 "L"때 출력 단자는 A, B, C입력에 의한 옳바른 결과를 나타내나, "H"일때는 모든 출력 단자를 "H"로 하여 준다. 이는 G2A와 G2B 단자가 \overline{IOR} , \overline{IOW} 및 SN74LS688 칩의 $\overline{P}=\overline{Q}$ 에 도시된 회로에 보이는 바와 같이 연결되어 입력 및 출력시 해당 어드레스의 출력 단자를 작동시킨다.

3.3 양방향 버퍼(Bi-Directional Buffer)

양방향 버퍼(SN74LS245)는 데이터의 전송 방향에 따라 입·출력단자의 임피던스를 제어하는 장치이다. 즉 \overline{G} 가 "H"이면 DIR에 관계없이 A, B양측의 임피던스가 무한대(isolation)가 되고 "L"일 때는 DIR에 따라 전송방향이 결정된다. DIR이 "L"이면 B단자(입력)에서 A단자(출력)로, "H"이면 A단자(입력)에서 B단자(출력)으로 데이터가 전송될 수 있다.

양방향 버퍼가 사용되는 이유는 데이터 버스를 타고 흐르는 데이터의 충돌을 피하기 위함이다. 여기에서 사용된 양방향 버퍼는 후술할 입력 버퍼와 기능이 중복되어 있어 논리상은 필요 없으나, 시스템 버스의 전류 분할에 의한 안정성을 높이기 위하여 이를 추가하는 것이 안전하다.

3.4 RESET

CPU에서 발생하는 RESET 신호는 매우 순간적 이므로 주변 칩들의 전력 상태가 정상이

로 되기 전에 다시 "L"로 돌아갈 가능성이 크다. 그러므로 약간의 지연을 통하여 주변 칩에 공급하는 것이 안전하다. 그러므로 NOT 게이트(SN74LS04)를 직렬로 중복 연결시킴으로써 지연을 유도한다.

3.5 입력 버퍼(Input Buffer)

양방향 버퍼와 매우 유사하게 작동하는 버퍼로서 데이터는 A단자로 부터 Y단자로만 진행하게 되어있다. 단, \overline{G} 단자가 "H"이면 Y단자의 임피던스가 무한대가 되어 데이터 버스와 전기적으로 분리되게 된다.

3.6 출력 데이터 래치(Output Data Latch)

출력 래치는 일종의 메모리로서 CLK의 상승에지 신호에 동기되어 D단자의 데이터를 수집한 후 즉시 Q단자에 수집된 데이터를 출력한다. Q단자는 CLK에 다음 상승 에지 신호가 검출될 때까지 그 값을 유지한다.

지금까지 예시된 회로를 참조하여 그 기능을 살펴 보았다. 이 회로는 8255등 전용 칩을 사용하지 않고 설계한 매우 간단한 것으로 실제에 널리 사용될 수 있으며 회로의 안정성이 뛰어나 까다로운 조건하에서도 매우 잘 작동한다. 위에서는 회로 및 논리에 대하여 간략히 살펴 보았으나 자세한 것은 참고 문헌등을 참조하여 연구하기를 바란다. 이 회로를 기본으로 약간의 회로 수정을 시도하여 좀더 다양한 기능의 회로 설계도 가능하리라 본다. 예를 들면 입력 및 출력회로를 더 확장하기 위하여는 입력 버퍼와 출력 데이터 래치부를 멀티 플렉서의 나머지 출력 단자에 평행으로 확장하면 된다. 예시된 회로에서는 입·출력 각각 최대 8개 까지의 확장이 가능하다.

4. 시험기판 제작 및 실험

예시된 회로를 제작하여 시험해 보기 위하여는 먼저 제작 방법을 결정하여야 한다. 가장

표 2 그림 4의 회로구성을 위한 부품 명세표

부품명	갯수
PC용 다목적 기판	1개
74LS688	1개
74LS245	1개
74LS244	1개
74LS273	1개
74LS138	2개
74LS04	1개
74LS06	2개
DIP 스위치 (8핀)	2개
발광 다이오우드	8개
래핑용 소켓 (20핀)	6개
래핑용 소켓 (16핀)	4개
래핑용 소켓 (14핀)	3개
래핑용 와이어	충분히
저항 (330Ω)	8개
저항 (8.2kΩ)	14개

손쉬운 방법은 와이어 래핑(wire wrapping) 기구를 이용하여 만드는 것이다. 래핑 기구 및 와이어의 사용법은 매우 간단하여 약간의 연습으로 훌륭한 결과를 얻을 수 있다. 표 2에 예시된 회로를 만들기 위한 부품 명세를 요약하였다. 부품은 청계천의 전자 상가등에서 쉬게 구할 수 있다. 일단 회로도에 따라 PC용 다목적 기판에 래핑용 IC소켓을 그림 5와 같이 배치한후 착오가 없도록 배선을 한다. 가능하면 경험자의 도움을 받으면 좋은 작품을 완성할 수 있다. 배선은 가능한 한 단거리를 이용하되 회로상 평행한 선들은 배선도 평행 혹은 다발로 하는 것이 추후 배선 검증에도 유리하며 잡신호의 배제에도 좋다. 각 칩에 공급되는 GND 및 V_{cc} 선을 제일 먼저 배선하여 전력선이 불필요하게 다른 배선들과 혼동되는 일이 없도록 주의하여야 한다. 각 칩에 공급되는 전력은 PC의 시스템 버스에 포함되어 있으므로 별도의 전력 공급은 필요없다. 배선을 너무 팽팽하게 하면 합선 및 단선의 우려가 있으므로 넉넉한 치수로 부드러운 곡선을 이루도록 하는 것

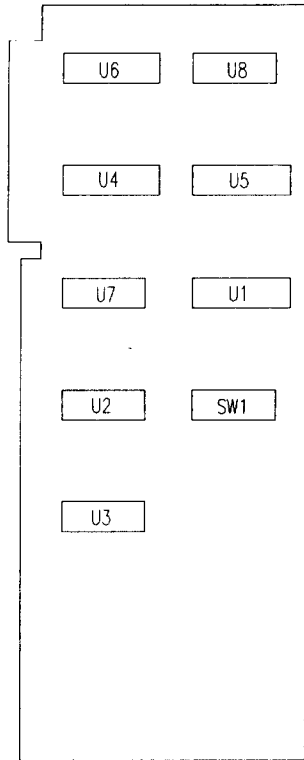


그림 5 부품배치도

이 여러모로 유리하다. 그림 1에 나타나 있는 PC 시스템 버스 확장 슬롯의 단자 번호를 잘 확인하여 좌우가 바뀌거나 하는 틀린 배선이 없도록 주의하여야 한다. 이를 잘못 배선할 경우 컴퓨터 본체를 손상하게 되는 경우도 있으니 특히 조심하여야 한다. LED들과 입력 스위치는 배선을 길게하여 추후 기판이 PC본체에 삽입된 후 밖으로 나올 수 있도록 하자.

일단 배선이 완성되면 틀린 곳이 없는지 검토를 하고 이상이 없으면 입·출력포트의 어드레스 세팅을 하여야 한다. 각자가 사용하고 있는 PC의 확장기판의 어드레스와 중복되지 않도록 표 1을 참조하여 값을 결정한다. 본 글에서는 편의상 308H (776D 또는 1100001000B)로 결정하였다. DIP(SW1)스위치를 그림 6과 같이 세팅한다. DIP스위치는 A3이상의 비트

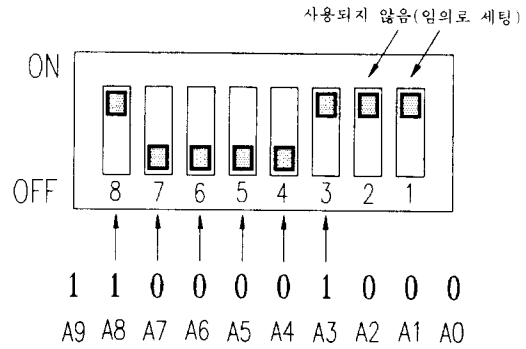


그림 6 DIP 스위치의 세팅

에 의한 어드레스 만을 결정하므로 DIP스위치에 의하여 결정되는 어드레스는 필히 8의 배수이어야 하며 실제 포트 어드레스는 이 값에 I/O 버퍼가 연결된 레지스터 번호를 더하여 주어야 한다. 예시된 회로의 경우 입·출력 포트 어드레스는 각각 $776+7=783$ 이다.

모든 준비가 끝나면 완성된 기판을 PC에 삽입하여야 한다. PC 전력을 끈 후 본체의 커버를 열고 확장 슬롯의 위치를 확인한 후 비어 있는 확장 슬롯에 조심스럽게 삽입한다. 입력 스위치, LED등을 슬롯 전면의 구멍을 통하여 밖으로 꺼내고 다시 PC의 커버를 닫는다. 본체의 커버를 열거나 확장 슬롯에 기판을 삽입할 때는 반드시 PC의 전력을 꺼야 하는 것을 잊지말아야 한다. 일단 설치가 끝나면 PC에 전력을 가하고 표 3에 있는 시험 프로그램을 수행하여 보자. 이 프로그램은 BASIC언어로 구성되어 있으므로 이해하는데에 어려움이 없으리라 본다. 이 프로그램의 기능은 다음과 같다.

프로그램을 수행시키면 LED가 교대로 점멸하기 시작한다. 이때 입력 DIP스위치(SW2)중 임의의 것을 "ON"으로하면 LED는 점멸을 멎는다. 프로그램상의 지연 루프의 계수를 증감시키면 점멸 속도가 변하는 것을 관찰할 수 있다. 지연 루프를 제거하면 이 프로그램은 1부터 8까지의 랜덤 수 발생기로도 사용될 수도 있다. 더욱 빠른 점멸을 필요로 할 때는 프로

표 3 시험 프로그램

```

100 DEFINT X,Y,DELAY
110 Y=1
120 DELAY=200
130 X=INP(783)
140 IF X <> 0 THEN 200
150 OUT 783,Y
160 FOR I=1 TO DELAY;NEXT I
170 Y=Y*2
180 IF Y=256 THEN Y=1
190 GOTO 130
200 X=INP(783)
210 IF X=0 GOTO 150
220 GOTO 200
    
```

그램을 C언어나 어셈블리로 재작성하여 보자. 이는 매우 간단한 기능의 프로그램이지만 예시된 회로 및 포트 I/O를 이해하는데에는 충분하리라 본다.

5. 맺음말

이제까지 간단한 포트 I/O 방식의 원리 및 실제에 대하여 살펴 보았다. 실제 제작 및 실

험에서 간단한 실험을 행하여 보았으나 그 응용 범위 및 활용도는 무궁무진하며 모든 PC 인터페이스는 이것으로 부터 시작된다고 해도 과언이 아니다. 관심있는 분들은 제작을 시도하여 보기를 바란다.

참 고 문 헌

- (1) Eggebrecht, L.C., 1983, "Interfacing to the IBM Persnal Computer", Howard W. Sams & Co., Inc.
- (2) TTL Data Book, Texas Instrument
- (3) Microprocessor and Peripheral Handbook, Intel.
- (4) IBM PC XT/AT Technical Reference.
- (5) James, W. Coffron, 1984, "The IBM PC Connection", SYBEX Inc.,
- (6) John, Uffenbeck., 1987, "The 8086/8088 Family-Design, Programming and Interfacing", Prentics-Hall, Inc.

