

1 비트 마이크로프로세서를 이용한 프로그램형 제어기의 설계

The Study on the Design of PC using the Single-bit microprocessor

우 광 방
김 현 기

연세대학교 전기공학과

1. 서론

1968년 미국의 General Motors 사가 산업용 각종 제어 장치가 구비해야할 조건을 구매회사에 명기하므로써 연구되기 시작한 프로그램머블 컨트롤러(Programmable Controller:PC)는 70년대로 마이크로프로세서의 출현과 더불어 발달하여 기능별로그 종류가 다양해져 가고 있다. Hard wired logic이었던 제어반을 PC를 사용하여 Soft wired logic으로 대체하므로써 신뢰성, 기밀 유지, 적은 전력 소모, 시퀀스 변경에 대한 신속한 대응등의 이점음 가져올 수 있어서 초기에는 중화학공업의 대규모 공장 위주의 I/O 점수 512점 이상 되는 PC가 많았으나, 최근엔 공장기계나 중소규모 업계의 생산라인에 까지 적용되어 I/O 점수 128점 이하의 소용량 PC가 대중화 되고 있다.

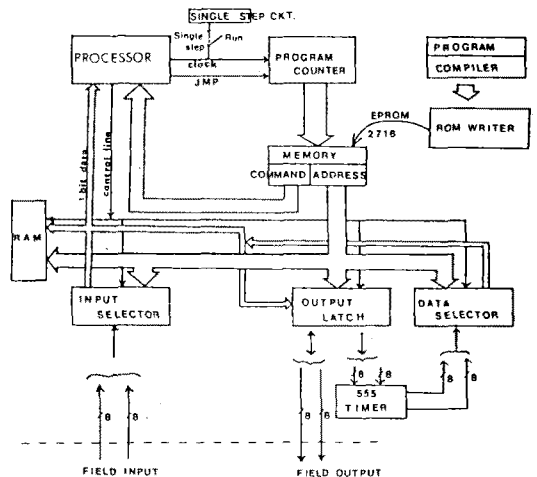
PC에서는 스캔타임의 해결 방법으로서 듀얼 프로세서(Dual Processor) 방식, 멀티 프로세서 방식, 고속 처리를 요하는 부분의 하드웨어 사용, 비트 슬라이스 마이크로 프로세서 방식(Bit slice microprocessor) 등을 사용한다. 그러나 현재의 PC는 멀티 비트 마이크로 프로세서를 사용하므로써 어떤 경우에는 필요없는 기능을 갖고 하드웨어의 구성이 복잡하여 가격과 스캔 타임의 문제가 있다. 4.5.6

본 논문에서는 컨트롤 전용의 프로세서로서 효과적인 MC 14500B 1 비트 마이크로 프로세서를 이용한 PC를 설계 제작하고 이에 따른 간단한 언어를 정의하여 컴파일러를 구성하고자 한다.

2. 시스템 구성

4개의 보드 즉 프로세서 보드, 입력 보드, 출력 보드, 타이머 보드로 구성되며 각 보드는 본 하드웨어의 구성에 적당하게 정의한 버스로 연결하였다.

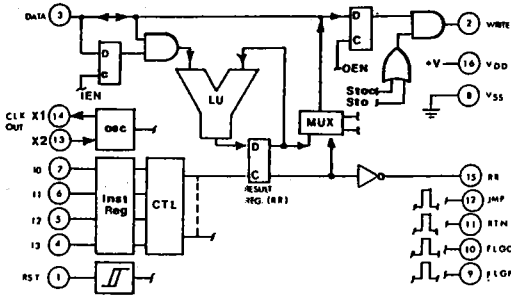
다음 그림에 전체 시스템 구성을 보인다.



- 그림 1 - 전체 시스템 구성 및 흐름도

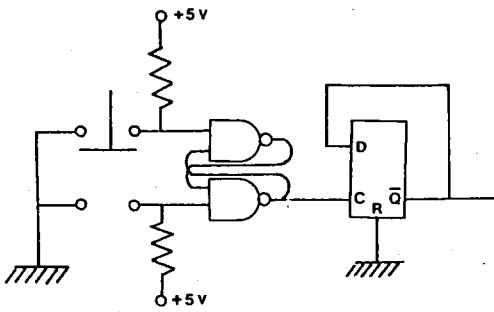
적용 프로세스의 시퀀스가 결정되면 사용자는 릴레이 래더 다이어그램(Ladder Diagram)이나 심볼릭 로직 다이어그램(Symbolic logic diagram)을 작성하여 프로그램한 후 ROM Writer로 EPROM에 넣어서 본 PC에 장착한다. 전원을 넣고 시스템을 reset시키면 프로그램 카운타가 동작하여 이에 의해 메모리에 있는 16 비트 인스트럭션이 1 스텝씩 수행된다. 프로그램의 마지막

엔 항상 JMP (jump) 명령이 있게되는데, 이에대한 JMP flag 로 프로그램 카운터를 reset 시켜서 스토어된 프로그램을 반복 수행한다. 다음에 본 제어기에서 사용한 7 비트 마이크로 프로세서 MC 14500B 의 블록 다이어그램을 보인다.



- 그림 2 - MC 14500B 블록 다이어그램

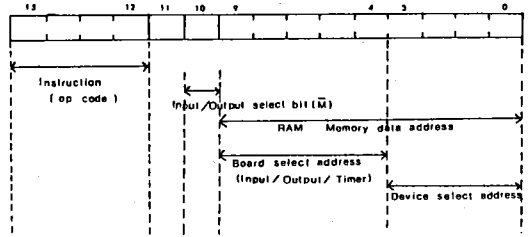
명령어 라인 IO-13 로 들어온 명령어들은 클럭펄스의 하강구간에서 인스트럭션 레지스터 (IR) 에 래치되고, 컨트롤 로직 (CTL) 에서 디코우드되어 로직유닛 (LU) 에 로직명령을 보내므로서 외부로 부터의 1비트 데이터와 Result Register (RR) 의 데이터에대한 논리 연산을 하게 된다. 또한 CTL 에서는 JMP, RTN, FLOG, FLG 등 디코우드하여 핀 9-12로 flag 를 보낸다. 이러한 flag 들은 1클럭 사이클 동안 액티브되어 외부 제어신호로 사용된다. 내부 발진기 (Oscillator) 에서 나온 클럭은 시스템 전체를 동기시키는데 사용되며 X1 과 X2 사이에 가변저항을 담아 조정한다. 본 논문에서는 Dual type D Flip-Flop 과 Quad 2 input NAND gate 를 이용한 수동 펄스 발생회로를 설계하여 X2 에 연결하므로서 싱글 스텝 (Single step) 이 가능하게 하였다.



- 그림 3.- 수동 펄스 발생 회로

그림 2에서 데이터핀과 Write 핀에 접속되어 있는 두개의 내부 래치는 데이터가 프로세서로 들어오거나 나가는 것을 제어하는데 이것은 IEN, OEN 명령에 의해 enable 또는 disable 된다.

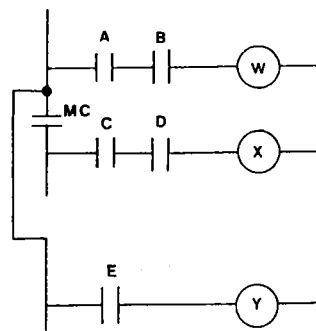
다음은 본 제어기의 16비트 명령어 형식이다.



- 그림 4 - 본 제어기의 16비트 명령어 형식
그림 4에서 상위 bit 15-bit 12 는 4 bit 의 OP 코드로서 프로세서의 16개 명령어로 사용했으며 bit 10은 RAM, I/O의 선택 비트로서 bit 10이 "0"이면 bit 9-bit 0 는 temporary storage 로 사용한 RAM의 어드레스가 되고, "1"이면 bit 9-bit 4 는 I/O 보드나 Timer 보드를 선택하는 보드선택 어드레스, bit 3-bit 0 는 각 보드가 포함하고 있는 I/O 포트의 번지가 된다.

3. 소프트웨어

그림 2의 RR 핀을 입력보드의 한 입력어드레스로 할당하므로서 다음과 같은 IF THEN 이나 IF THEN ELSE 블록의 프로그램이 가능하다.

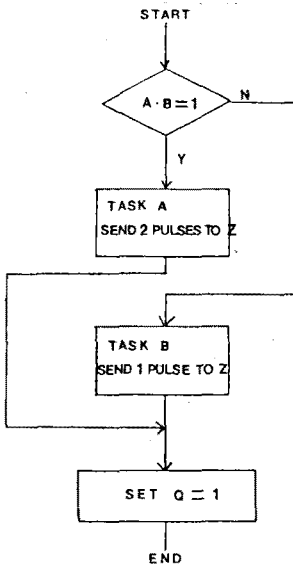


1. LD A
2. AND B
3. STO W
4. IEN MC
5. LD C
6. AND D
7. STO X
8. ORC RR
9. IEN RR
10. LD E
11. STO Y

- 그림 5 - 래더 다이어그램의 예
위 프로그램에서 MC (Master Contactor) 가 "1" 이어야만 두번째 사다리 가로대 (Ladder rung) 가 enable되며 프로그램 스텝 8은 $RR + \overline{RR}$ 이므로 스텝 9에서 IEN RR 하므로서 이후

의 입력은 영향을 받지 않는다.

다음은 IF THEN ELSE 블록의 예이다.



1. LD A
2. AND B
3. OEN RR
4. STO Z
5. STOC Z
6. STO Z
7. STOC Z
8. LDC RR
9. OEN RR
10. STO Z
11. STOC Z
12. ORC RR
13. OEN RR
14. STO Q

- 그림 6 - IF THEN ELSE의 예

그림 6의 프로그램에서 스택 3에서 RR의 값에 따라 4,5,6,7의 수행여부가 결정되며 8,9에서는 RR 값이 리플리먼트되어 10,11을 수행하고 12, 13에서 RR=1이 되어 Output Enable 시키게 된다.

4. 실험결과

샘플 프로그램을 ROM 에 넣고 입력은 각번지 별 딥(dip) 스위치로 조합을 하여 타이머 보드, 출력 보드로 LED 를 구동 시켜서 전체 시스템의 정상적인 동작을 얻었다. 컴파일러는 Cromemco 의 SBASIC 으로 구성되어 샘플 로직 시퀀스에 적용시켜 올 바른 컴파일 리스트를 얻었다.

5. 결론

프로그램형 제어기의 주기능은 ON/OFF 스위칭이다. PC 에서 대부분의 사용자들이 필요로 하는 기능은 비트연산과 타이머이다. 이러한 일에 컨트롤 전용의 1비트 마이트로 프로세서를 채택한 제어기가 매우 효율적이라고 결론짓는다.

즉, 1. 하드웨어가 복잡하지 않고 제작비용과 운용비용이 절감된다.

2. $V_{DD} = 15$ 에서 5 MHz 이상으로써, 수행은 1 instruction/clock period 이므로 속도(scan

time)가 빠르다.

3. 모듈화가 용이하여 확장이 편리하다.
4. 16개의 명령어를 사용하므로 소프트웨어의 구사가 쉽다. 특히 심볼릭 로직에 대한 프로그램에 적합하다.

참고 문헌

1. Gordon H, Smith: "Converting Relay Logic to Software" Machine Design, Vol. 50, No. 19, Aug. 24, 1978, pp93-99.
2. K. Rajarman: "A Microprocessor-Based Time Sequence Controller for Processor Control Applications". IEEE Trans, Ind. Electronics & Control Inst, Vol. IECI-128, No. 3, pp 214-219 Aug. 1981.
3. V. Gregory, B. Dellande, et al.: "MC14500B Industrial Control Unit Hand Book" Motorola Semiconductor Products, Inc., 1977.
4. V.J. Maggioli: "How to Apply Programmable Controllers", Hydrocarbon Processing, Vol. 57, No. 12, pp137-142, Dec. 1978.
5. "Control and Automation Systems" Machine Design May 31, 1984. pp85-96.
6. R.L. Tabachnick et al.: "Sequence Controllers with Standard Hardware and Custom Firmware" IEEE MICRO, pp9-25, May 1981.
7. Wayne P. Kraemer: "Testing and Start-Up of Programmable Controller Systems" IEEE Trans. Industry Applications, Vol. 1A 16, No. 5, September/October, 1980.
8. V.J. Maggioli: "Programmable Controllers in Process Control Application" IEEE Trans. Industry Applications, Vol. 1A-15, No. 6, pp695-700, November/December 1979.
9. Paul J.A. Zsombor-Murray, et al.: "Binary Decision based Programmable Controller." IEEE MICRO, Aug., Oct., 1983.