

Programmable Controller의 명령어 수행 SYSTEM

최호현, 안재봉, 지동근, 유지훈, 이영준
 금성산전주식회사 연구소

Instruction Execution SYSTEM in Programmable Controller

H.H. CHOE, J.B. AHN, D.K. CHI, J.H. YOO, Y.J. LEE
 Goldstar Industrial Systems Co., Ltd. R&D Lab.

Abstract

The high-speed tendency of Programmable Controller requires a fast processing of instruction. This project investigates the instruction execution system for a fast processing in Programmable Controller.

1. 서론

Programmable(Logic) Controller(이하 PLC로 약칭함)의 고기능화 및 고속화의 요구에 따라 PLC의 명령어를 Software로만 처리하면 Scanning time이 길어지므로 대용량의 user program area를 제공해야 하는 중형급 이상의 PLC에 있어서는 빠른 scanning time의 실현을 위해 명령어의 수행을 Hardware로 처리할 필요성이 생긴다. 그러나 PLC에서 사용되는 전체 명령을 Hardware로 수행하려면 Hardware cost가 무척 높아져서 PLC 상품으로서의 경쟁력을 상실하므로 PLC가 사용되는 용도 및 응용분야를 고려하여 Hardware로 처리할 명령과 Software로 처리할 명령을 분류하여 SYSTEM을 구성하여야 한다.

여기서는 금성계전(주) 연구소에서 개발한 PLC 'STARCON-A56'의 RUN SYSTEM을 중심으로 PLC의 명령어 수행 SYSTEM을 고찰한다.

2. 명령어 구성

일반적으로 PLC의 명령어는 STEP으로 표시되며 명령어의 종류에 따라 1 step 명령어와 복수 step 명령어로 구성된다. 1 step은 2 bytes를 차지하는 것이 보통이다.

'STARCON-A56'에서는 LOAD, AND, OR, OUT 등의 Sequence 기본 명령을 Hardware로 처리하고 그외의 Sequence 응용명령(SC, SR등)과 Data 연산명령(MOV, ADD등)은 Software로 명령어 수행을

하도록 구성하였다. Hardware로 처리하는 명령어는 모두 1 step 명령어이며 수행시간은 5.6us로 일정하다.

3. 명령어 수행 SYSTEM

PLC에서의 명령어 수행은 그림 1에서의 Memory에 들어있는 user program에 따라 차례로 처리해 나간다

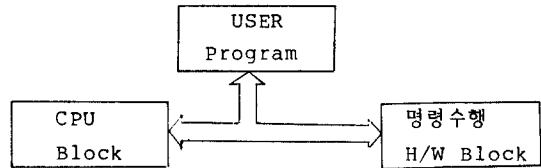
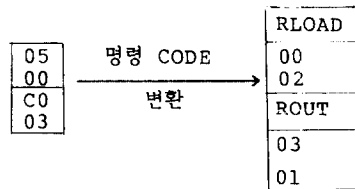


그림 1. 명령 수행에 필요한 기본 구성

일반적으로 고속의 scanning을 필요로 하지 않는 소형급 PLC에서는 그림 1에서의 명령 수행을 위한 Hardware block이 없이 모든 명령을 CPU block에서 다 처리를 한다. 이 경우 LOAD, AND, OR, OUT 등의 Sequence 기본 명령 수행 시간은 일반적으로 명령어 1개당 평균 약 40-80us 정도 소요되며 그림 2와 같이 특수한 Software 기법을 사용한다 하더라도 명령어 1개 수행시간은 약 15us 정도가 소요된다.



a) LOAD 0.1, OUT 3.0 명령 code

RLOAD : POP H ; H←bit 위치, L←Card no.
 MOV C,B ; (MRG)←(ARG)
 AND ; (IRG) check
 NZ 10\$
 MOV L,H
 MVI H, DATA-HIGH

```

ANA M
10$ : MOV B, A
      XRA A
      RET ; 이 명령에 의해 ROUT
          routine으로 Jump 한다.

ROUT : POP H ; H bit 위치, L Card no.
      ORA B
      JZ 10$
      MOV A, H
      MVI H, DATA-HIGH
      ORA M
      MOV M, A
      XRA A
      RET ; 이 명령에 의해 다음 step의
          명령수행 routine으로 Jump
          한다.

```

10\$: - - - -

b) LOAD, OUT 명령 수행 Program

그림 2. 'STARCON-A48' 명령어 수행 예

그림 2와 같은 명령 수행은 PLC의 RUN mode로된 초기에 Initial 작업으로 그림 2의 a)와 같은 form으로 명령 code를 변환하여 이 변환된 memory 영역을 STACK으로 이용하는 Software 기법을 사용한 것이다. 이러한 고도의 software 기법을 사용해도 PLC의 고속화 요구에 부합되지 않는 경우에는 그림 1의 명령 수행 Hardware block을 추가하여 명령 수행 SYSTEM을 구성하여야 한다. 명령 수행을 위한 Hardware의 design에 있어서 속도와 cost 곡선이 그림 3과 같으므로 여러 종류의 유형이 있겠으나 여기서는 'STARCON-A56'의 명령 수행을 위한 Hardware block을 고찰한다.

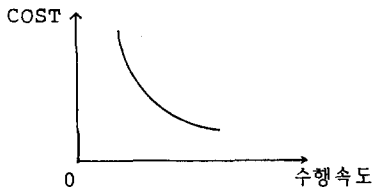


그림 3. H/W 설계에서의 cost-speed 곡선

4. 명령어 수행 Hardware block

'STARCON-A56'에서의 명령 수행 Hardware는 그림 4에서와 같이 4 종류의 function block으로 구성되어 있다.

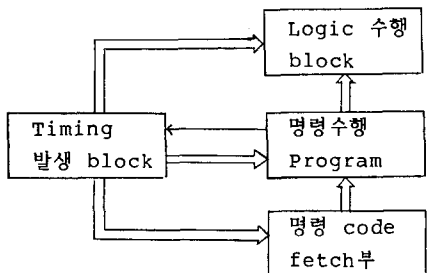


그림 4. 'STARCON-A56'의 H/W RUN SYSTEM

(1) Timing 발생 block

명령의 hardware 수행을 위한 7개의 STATUS 신호 (S1-S7)를 발생한다. 이 Timing 발생 block을 그림 1의 CPU block에서 Enable 시키면 S1 status 신호부터 S7 신호까지 차례로 발생되어 명령을 수행해 나가며 H/W로 수행 불가능한 명령을 만나면 Timing 발생 block이 reset되어 명령 수행을 CPU block으로 넘긴다. 각 STATUS는 800ns 동안 차례로 Active하게 되므로 명령 1개 수행시간은 $800ns \times 7 = 5.6\mu s$ 이 된다.

(2) 명령 CODE Fetch부

Hardware로 수행할 수 있는 명령은 code의 2nd byte가 I/O의 low address 정보를 갖도록 명령 CODE를 구성하였다.

그림 5에서와 같이 이 block은 S1과 S2 STATUS 동안 Enable 되어 S1 status에서는 명령 code의 1st byte를 명령 수행 program block으로 넘겨준 후 Address를 1 증가시켜 S2 status에서 명령 code의 2nd byte를 명령 수행 program block으로 넘겨준다.

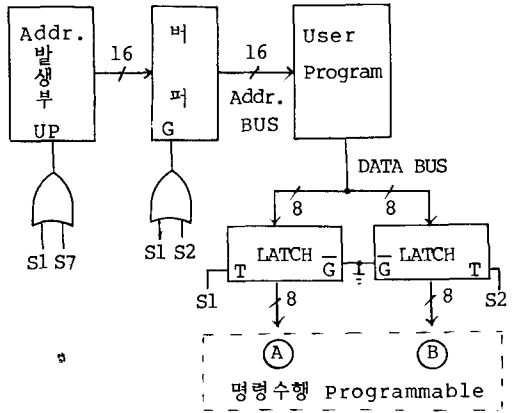


그림 5. 명령 CODE Fetch block

(3) 명령 수행 program block

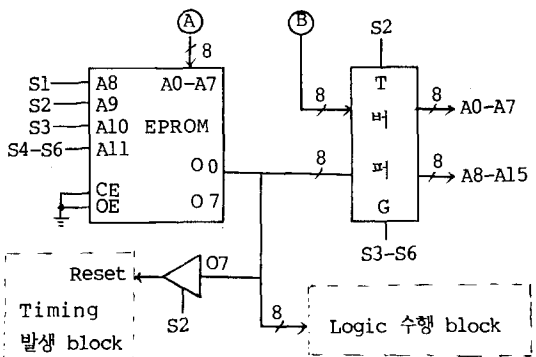


그림 6. 명령 수행 Program Block

명령 code Fetch부에서 S1 status에서 받은 명령 code의 1st byte 내용이 명령 수행 program 이 있는 EPROM의 Address line에 Latch되어 각 명령에 대해 Program된 data가 EPROM의 data line으로 흘러나온다. S2 status에서 현 수행할 명령이 Hardware로 수행 가능할 경우 7F 이하의 I/O 상위 Address가, 수행 불가능할 경우는 FF가 각각 EPROM의 data line에서 흘러나와 Timing 발생 block의 reset 조건을 형성한다. 이 경우 Timing 발생 block이 reset 되면 S3 이후의 status가 나타나지 않으므로 그림 4의 모든 function block은 dead 상태가 된다. 그림 1의 CPU block에서는 명령을 Hardware로 수행할 동안 자기 진단기능 또는 주변기기와의 통신기능등을 수행하면서 계속 Timing 발생 block의 상태를 감시하다가 Timing 발생 block이 reset 되면 현 수행할 명령을 read하여 Software로 명령처리를 한 후 Timing 발생 block을 다시 enable 시키므로 PLC의 전 명령을 각각의 성격에 따라 Hardware 또는 Software로 처리할 수 있게 된다. 현 명령이 Hardware로 수행 가능할 경우는 S2 status에서 Timing 발생 block을 reset 시키지 않으므로 S3 이하의 STATUS가 차례로 Active되어 각 STATUS에서의 명령 수행 Program의 내용에따라 Logic 수행 block을 control 하면서 명령을 처리해 나간다.

(4) Logic 수행 block

S3 이후의 STATUS에서 Enable되어 명령 수행 Program 대로 LOAD, AND, OR, OUT등의 function을 수행한다. LOAD X005 명령 수행을 예로 들어 Logic 수행 block을 고찰해 본다. LOAD X 명령의 1st byte code가 10H 이므로 명령수행 Program은 EPROM의 110번지에 00, 210H번지에 68H, 410H번지에 08, 810번지에 1AH의 data가 Program되어 있다. 따라서 S1 status에서 fetch한 10H란 data가 명령수행 program block 내의 EPROM Address에 Latch 되므로 S2 status에서는 68H란 X 영역의 상위 Address가 EPROM에서 흘러나온다. 이 경우 07이 1이 아니므로 Timing 발생 block은 reset 되지 않고 S3 이후의 STATUS가 계속된다. S3 부터 S6 STATUS 동안에는 그림 6에서 보는 바와 같이 SYSTEM Address bus상에 X005의 Actual Address인 6805가 실리게 된다. S3 status 에서는 08 이란 data를 이용하여 그림 2와 같은 개념으로 ARG의 내용을 MRG로 이동(그림 7의 Shift Register의 좌 Shift) 시킨후 S5 STATUS에서 SYSTEM DATA BUS상에 있는 X005의 on, off 정보를 ARG로 LOAD 시키므로 LOAD 명령의 1 cycle을 종료한다.

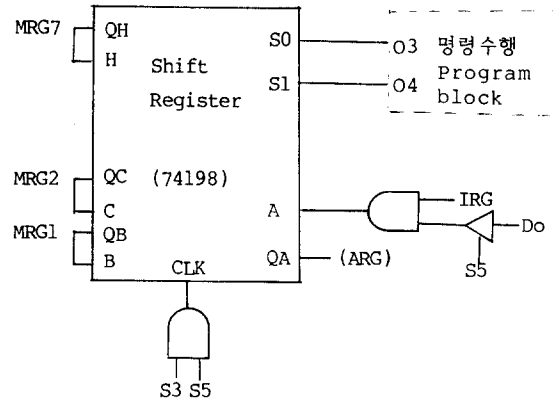


그림 7. Logic 수행 block(LOAD 관련 part)

5. 결론

HARDWARE 명령 수행은 S3 이하의 1-2개의 STATUS만으로도 처리 가능하나 입력 및 출력 unit와 PLC에 속한 각종 특수 unit등의 Access Time을 고려하여 'STARCON-A56'에서는 5개의 STATUS로 처리하였다. 또한 각 STATUS가 Active되는 시간도 사용되는 소자(RAM, EPROM등)를 고속 소자의 채택으로 800ns 이하로도 단축이 가능하다. 따라서 PLC의 명령어 수행 SYSTEM의 design에 있어서는 PLC의 등급 및 Total System 사양을 충분히 고려하여 각 경우에 대한 최적의 SYSTEM을 설계하여야 한다.

References

1. Louis Nashelsky "Introduction to Digital Computer Technology" 탑 출판사 (1978)
2. Morris E. Levine "Digital Theory and Experimentation Using Integrated Circuits" 탑 출판사 (1977)
3. M. Morris Mano "Computer System Architecture" Prentice-Hall, Inc. (1982)
4. "The MCS-80/85 Family User's Manual" Intel (1983)