

마이크로프로세서를 사용한 고리원자력발전소용 SSILS의 설계

*하달규, *김창구, *박기현, **전성준, **문성득

*경기도 안양시 삼창기업주식회사 부설연구소

** 부산시 남구 용당동 부경대학교

요 약

본논문에서는 고리원자력발전소용 SSILS(Solid State Interposing Logic System)을 소프트웨어의 건전성을 확보하면서 마이크로프로세서로 설계해 가는 과정을 기술하였다. 설계 제작된 시제품은 IEEE Std 344-1975에 따라 내진시험을 하였고 IEEE Std. 323-1983에 따라 환경시험을 필하였다. 그리고 소프트웨어의 건전성은 자체적으로 확인하였다.

제 1 장 서 론

최근들어 디지털회로의 발달로 산업의 전분야에 컴퓨터를 중심으로 하는 디지털 방식의 적용이 날로 증가해 가고 있다. 이러한 추세에 따라 원자력 계통에도 디지털화하려는 노력이 증가해 가고 있으나 소프트웨어의 무절제한 사용은 사전에 미처 발견되지 아니한 오류에 의해 원자력 안전 계통에 크나큰 어려움을 가져다줄 수 있다. 안전성의 확보는 사용전에 완벽한 시험을 거쳐야 확보할 수 있다. 그러나 많은 경우 소프트웨어의 완벽한 시험이란 많은 시간과 비용을 소모하면서도 가능하지 않다. 이는 모든 경우의 집합이란 거의 불가능하기 때문이다. 그래서 소프트웨어 공학에서는 개발후의 시험(acceptance

test) 비용을 줄이며 시험의 효과를 높이기 위하여 개발 초기부터 오류를 발견하고 처리하고자 하는 연구를 많이 하였다.

삼창기업과 부경대가 공동으로 고리 원자력발전소에서 사용되는 SSILS(Solid State Interposing Logic System)중 표1에 표시한 논리회로로만 구성된 8개의 제어카드를 microprocessor로 설계하였다. 본 연구에서는 개발단계를 그림1과 같이 하여 그 건전성을 확보하고자 하였다. 사양 검토 단계에서는 사양을 명확히 하고 그 결과를 logic diagram과 동작설명으로 문서화한다. 예비설계단계에서는 사양으로 주어진 logic diagram을 Boole 함수와 flow-chart로 표현한다. 하드웨어 개발단계에서는 각제어카드에 주어진 사양을 모두 수용할 수 있도록 하드웨어를 설계한다. 설계 및 단위 시험단계에서는 SSILS프로그램의 가장 작은 구성단위를 macro로 정의하고 이 macro와 하드웨어를 시험한다. 프로그래밍과 통합단계에서는 macro를 이용해 각 제어카드를 3개 정도로 분할하여 프로그램하고 시험한후 다시 통합하는 과정을 거친다. 검증(수락)시험은 사용자 또는 의뢰자의 요구를 만족하는지 확인하는 단계이다.

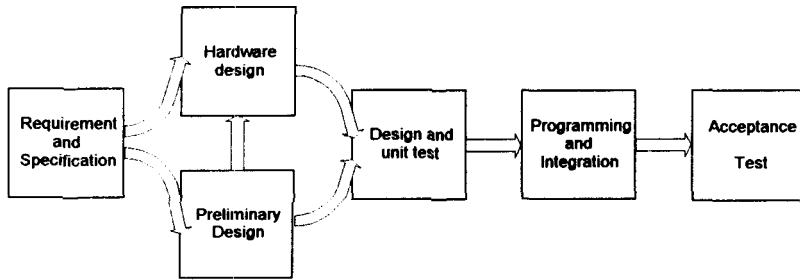


그림1 SSILS의 논리모듈 개발 과정

표1 개발한 8종의 논리제어카드

| 약어 | 정식명 | 약어 | 정식명 |
|-----|-------------------------|-----------|-----------------------------|
| ACB | Air Circuit Breaker | BOP/ESFAS | Balance of Plant ESFAS |
| SOV | Solenoid Operated Valve | MSIV/FWIV | Main Steam Isolation Valve |
| MOV | Motor Operated Valve | RLM | Reflash Logic Module |
| ALM | Auxiliary Logic Module | NRS | Non-Reversing Motor Starter |

제 2 장 본 론

1. 사양분석단계

사양분석단계에서는 사양을 검토하여 명확히 한다. 많은 경우 소프트웨어 project의 실패나 지연이 명확하지 않은 사양에서 출발한 것이 원인으로 알려져 있다. SSILS의 논리제어카드는 논리회로만으로 구성되어 있으므로 검토 결과를 그림2와 같이 logic diagram으로 문서화하고 동작설명을 첨부한다. 입력과 출력신호를 정확히 정의하여 하드웨어 설계의 기초자료를 제공한다. 그림2는 논리제어카드중 SOV(Solenoid Operated Valve)의 제어로직부분이다.

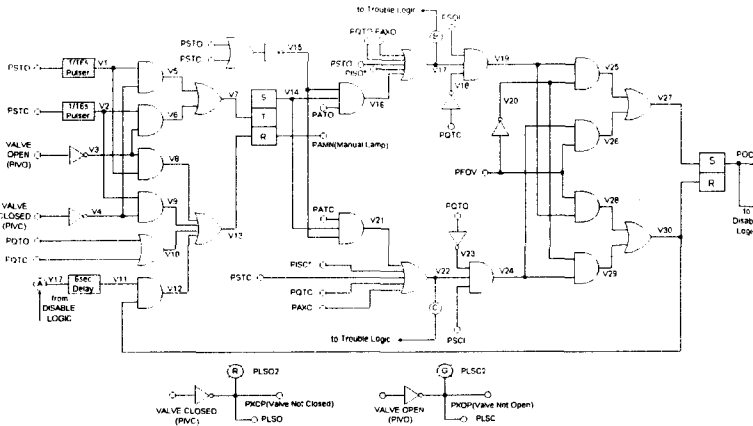


그림2 SOV 제어로직

2. 예비설계단계

예비설계단계에서는 logic diagram으로 주어진 사양을 Boole 함수로 표현하고 flow-chart를 작성한다. 프로그램1은 SOV의 제어로직부분은 Boole 함수로 표현한 예이고 그림3은 이의 flow-chart이다. 프로그램1과 그림3에서 보듯이 SSILS의 논리제어카드의 branch가 아주 적고 직선적인 프로그램이 얻어져 소프트웨어의 건전성을 확보하기가 쉽다.

3. 하드웨어의 설계

사양분석단계에서 정리된 입출력 신호를 근거로하여 하드웨어를 설계하였는데 논리제어카드중 7개는 동일한 PCB를 사용할 수 있도록 하였다. 설계에 사용된 프로세서는 Intel의 MCS-51 계통과 호환성이 있는 AT89C52이다. MCS-51은 Boolean processor라고도 불리는데 SSILS의 논리제어카드와 같이 논리회로로 구성된 경우에 아주 적합한 프로세서다.

프로그램1 SOV 제어로직 Boole 함수

```

PLSO = /P1VC
PXCP = /P1VC
PXOP = /P1VO
PLSC = /P1VO
V1 = F_pul1/16( PSTO )
V2 = F_pul1/16( PSTC )
V3 = /P1VO
V4 = /P1VC
V5 = V4 ^ V1
V6 = V2 ^ V3
V7 = V5 + V6
V8 = V1 ^ V3
V9 = V2 ^ V4
V10 = PQTC + PQTO
V11 = F_del6 ( Y17 )
V12 = V11 ^ V30
V13 = V8 + V9 + V10 + V12
V14 = F_ff ( V7, V13 )
PAMN = /V14
V15 = /( PSTO + PSTC )
V16 = V14 ^ V15 ^ PATO
V17 = PAXO + PISO* + PQTO + PSTO + V16
V18 = /PQTC
V19 = PSCI ^ V18 ^ V17
V20 = /PFOV
V21 = PATC ^ V14 ^ V15
V22 = V21 + PSTC + PQTC + PISC* + PAXC
V23 = /PQTO
V24 = PSCI ^ V23 ^ V22
V25 = V19 ^ V20
V26 = PFOV ^ V24
V27 = V25 + V26
V28 = PFOV ^ V19
V29 = V24 ^ V20
V30 = V28 + V29
POCM = F_rsf ( V30, V27 )
    
```

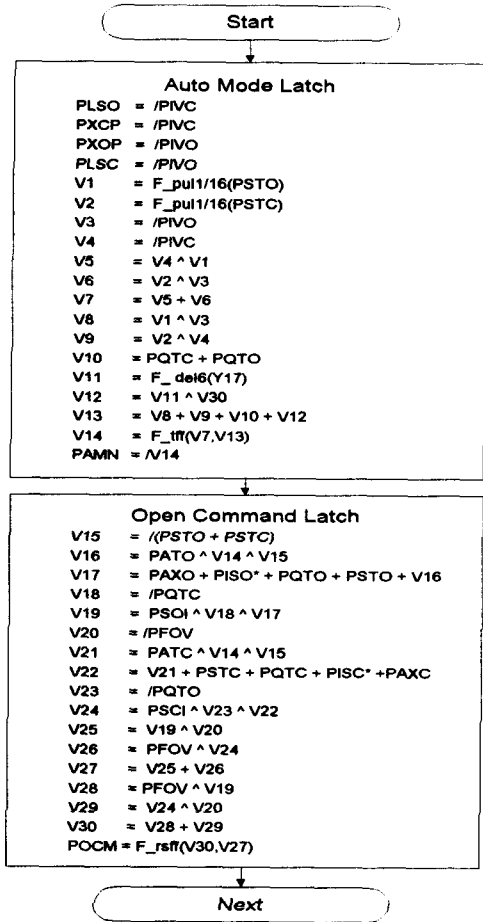


그림3 SOV의 Control Logic의 흐름도

4. macro 설계 및 단위 시험

가. macro 설계

프로그램의 수월성을 위하여 논리회로 구현에 필요한 AND, OR, Flip-Flop 등을 MCS-51 assembler가 제공하는 macro로 정의하고 이를 사용하여 프로그램을 하였다. 즉 macro가 프로그램의 최소 단위(unit)로 사용되었다. AND, OR 등 조합논리 기능의 macro는 매우 직접적이며 단순하게 정의가 되고 Timer, Pulser 등의 순차논리기능은 다소 복잡하며 프로그램기법이 필요하다. macro 설계의 예로 1 sec pulser의 설계는 다음과 같이 이루어진다. 1sec pulser는 다음의 사항을 만족시키면 되므로 이로부터 그림4의 flow-chart를 얻고 이 flow chart를 근거로 macro를 정의할 수 있다.

- 1) 입력이 "0"이면 출력이 "0"이다.

2) 입력이 "1"로 바뀌면 출력이 "1"이 되고 1[sec]를 설정해 둔다. 1[sec] 122[Hz] 클럭 122개에 해당한다.

3) 입력이 "1"이 되어 출력이 "1"이 된 후 1[sec]가 지나면 출력을 "0"로 만들고 flag(status)에 펄스 발생이 끝났음을 기록한다.

프로그램2 1sec pulser용 macro

```

:1sec Pulser
%*define(PULSR1(x,y)) local PUL_2 PUL_3 PUL_X
(
    JNB    %x, %PUL_2
    JNB    %y, %PUL_3
    MOV    A, RCAP2H
    CJNE  A, PUL1H, %PUL_X
    MOV    A, RCAP2L
    CJNE  A, PUL1L, %PUL_X
    CLR    %y
    SETB  PL1
    JMP    %PUL_X
%PUL_2:  CLR    PL1
        CLR    %y
        JMP    %PUL_X
%PUL_3:  JB     PL1, %PUL_X
        MOV    A, RCAP2L
        ADD    A, #(PTIME1 mod 256)
        MOV    PUL1L, A
        MOV    A, RCAP2H
        ADDC  A, #(PTIME1 / 256)
        MOV    PUL1H, A
        SETB  %Y
%PUL_X:)
    
```

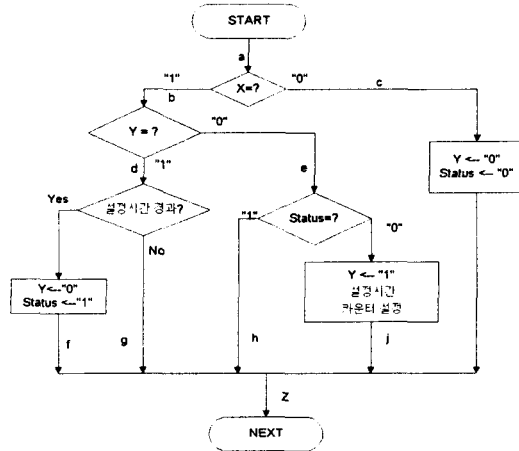


그림4 1sec pulser의 flow-chart

나. 단위 시험

프로그램의 최소 단위가 원하는 기능을 항상 발휘할 수 있을 지 확인하는 시험이 단위 시험이다. 예로서 위에서 설계한 1 sec pulser의 단위 시험은 다음과 같이 한다. 단위 시험은 그림 4와 같은 flow-chart에서 branch가 일어나는 모든 경우에 대하여 원하는 형태의 출력이 발생되는지를 확인하는 것이다. 입력에 따른 상태의 변화를 구하면 표2과 같다. case 1~5로 그림4의 flow-chart의 모든 decision과 segment를 cover한다. case6,7은 정상적으로는 나타날 수 없는 것으로 case2,3과 동일한 경로를 통하게 되고 입력이 바뀌거나 시간이 경과된 뒤 정상적인 상태로 돌아온다. 그림5는 상태천이도이다. 점선으로 나타낸 상태(1 1)은 정상적으로 나타날 수 없는 상태이다. 이 상태에서도 정상적인 상태로 돌아가게 하므로 이 프로그램은 모든 경우에 이상 없이 동작하게 된다. 프로그램은 그림6과 같은 신호를 인가하여 시험하여 이상유무를 확인할 수 있다. 그림6의 시험신호는 표2의 5 가지 case를 모두 포함한다. 그림 하단의 숫자가 표2의 case를 나타낸다.

표2 Pulsar의 입력에 따른 상태의 변화

| CASE | input | | 이전상태 | | 현재상태 | | 경로 |
|------|-------|----------------|------|---|------|---|-----------|
| | X | RCAP2 / MEM | STAT | Y | STAT | Y | |
| 1 | 0 | - | - | - | 0 | 0 | a→c→z |
| 2 | 1 | - | 0 | 0 | 0 | 1 | a→b→e→j→z |
| 3 | 1 | NOT Eq (n.T.E) | 0 | 1 | 0 | 1 | a→b→d→g→z |
| 4 | 1 | Eq (T.E) | 0 | 1 | 1 | 0 | a→b→d→f→z |
| 5 | 1 | - | 1 | 0 | 1 | 0 | a→b→e→h→z |
| 6 | 1 | NOT Eq (n.T.E) | 1 | 1 | 1 | 1 | a→b→d→g→z |
| 7 | 1 | Eq (T.E) | 1 | 1 | 1 | 0 | a→b→d→f→z |

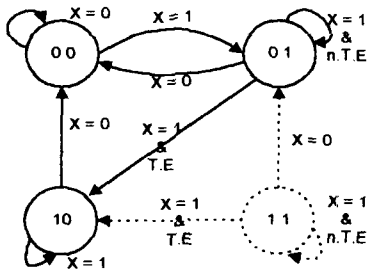


그림5 Pulsar의 state transition diagram

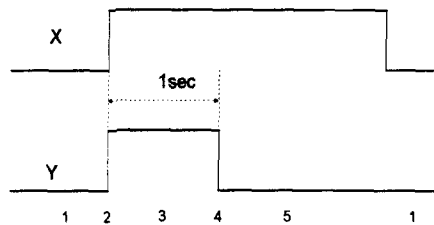


그림6 Test Set (Test Signal)

5. 프로그래밍과 통합

2.에서 기술된 Boole 표현과 작성된 flow-chart로부터 단번에 프로그램을 완성하여 acceptance test를 할 수 있으나 bottom-up 방식으로 통합하였다. 전체 프로그램은 그림7에 서와 같이 크게 제어로직, 오동작검출로직과 보조로직의 세부분으로 나누어 부분시험을 한 후 통합한다. 부분시험시의 시험신호는 검증시험시의 시험신호와 유사한 것을 사용한다.

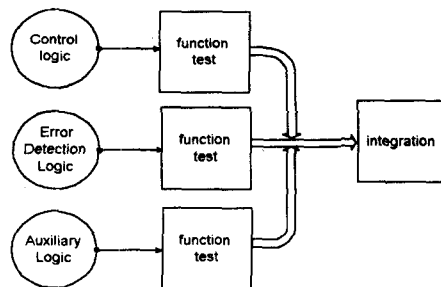


그림7 SOV의 통합 순서

6. 검증시험

소프트웨어의 검증시험(수락시험)으로 한국전력에서 사용중인 SSILS 시험절차를 따랐다. 표3은 SSILS 시험절차에 따른 시험서의 일부이다. 또 이와는 별도로 IEEE Std 344-1975 에 따라 내진 시험을 하였고 IEEE Std. 323-1983에 따라 환경시험을 필하였다. 내진시험과 환경시험은 미국의 검증회사인 NTS사에서 수행하였다.

표3 SOV 기능시험표

| step | ACTION | P O N H | P L T B | P A T B | P O M | P L C | P L S | P L D | P L D | P A M | P A O | P X O | P X T | P O C | P O O | P O B | P O A | P O A | P O O | P O N | |
|------|----------|------------------|------------------|------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|---|
| 49 | off | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 50 | BCM1 | | | | | | | | | | | | | | | | | | | | |
| 51 | PIVO | | | | | | | | | | | | | | | | | | | | |
| 52 | power on | 1 | | | | | 1 | | | 1 | | | 1 | | | | | | | | 1 |
| | 10sec | | | | | | | | | | | | | | | | | | | | |
| 53 | PITB | | f | 1 | | | | | | | | | | | | | | | | | 1 |
| | 1sec | | | | | | | | | | | | | | | | | | | | 0 |
| 54 | /PITB | | 0 | 0 | | | | | | | | | | | | | | | | | |
| 55 | /PIVO | | f | 1 | | 1 | | | | | | 1 | | | | | | | | | 1 |
| | 1sec | | | | | | | | | | | | | | | | | | | | 0 |
| 56 | PPCM | | 0 | 0 | | | | | | | | | | | | | | | | | |
| 57 | /PPCM | | f | 1 | | | | | | | | | | | | | | | | | 1 |
| | 1sec | | | | | | | | | | | | | | | | | | | | 0 |
| 58 | PPCM | | 0 | 0 | | | | | | | | | | | | | | | | | |

제 3 장 결 론

이상으로 고리원전의 SSILS의 논리제어카드를 마이크로프로세서로 설계하여 국산화하였고 설계한 회로가 필요한 기능을 발휘하며 이의 소프트웨어가 건전함을 확인하였다. 이 연구가 향후 전력산업의 국산화에 많은 기여를 하기 바란다.

본연구는 한국전력공사 지원자금으로 시행한 기술개발사업의 연구결과이다.

참고문헌

1. R. S. Pressman, Software Engineering, McGraw-Hill, 1992
2. B. T. Mynatt, Software Engineering with Student Project Guidance, Prentice Hall, 1990
3. Operation and Maintenance Instructions: Solid State Interposing Logic System, Consolidated Controls, 1982
4. Embedded Controller Handbook, Intel, 1987