Journal of Institute of Control, Robotics and Systems (2014) 20(6):689-694
http://dx.doi.org/10.5302/J.ICROS.2014.13.0007 ISSN:1976-5622 eISSN:2233-4335

다중화 구조 고신뢰성 제어기기를 위한 보팅 시스템버스 프로토콜

Voting System Bus Protocol for a Highly-Reliable PLC with Redundant Modules

정 우 혁, 박 재 현*
(Woohyuk Jeong[1] and Jaehyun Park[2,*])
[1]Department of Electronic Engineering, Inha University
[2]Department of Information and Communication Engineering, Inha University

Abstract: An SPLC (Safety Programmable Logic Controller) must be designed to meet the highest safety standards, IEEE 1E, and should guarantee a level of fault-tolerance and high-reliability that ensures complete error-free operation. In order to satisfy these criteria, I/O modules, communication modules, processor modules and bus modules of the SPLC have been configured in triple or dual modular redundancy. The redundant modules receive the same data to determine the final data by the voting logic. Currently, the processor of each rx module performs the voting by deciding on the final data. It is the intent of this paper to prove the improvement on the current system, and develop a voting system for multiple data on a system bus level. The new system bus protocol is implemented based on a TCN-MVB that is a deterministic network consisting of a master-slave structure. The test result shows that the suggested system is better than the present system in view of its high utilization and improved performance of data exchange and voting.

Keywords: safety PLC, TMR, fault-tolerant, TCN, MVB, IEC 61375, voting

I. 서론

원자력 발전소에서 사용되는 제어 시스템은 고도의 신뢰성이 요구되는 시스템으로, 제어 대상에 따라 발전소 제어계통 전반을 제어하는 비안전계통과, 원자로 계통을 제어하는 안전계통으로 나뉜다. 이들 중 안전계통은 원자력 발전소의 안전과 직결되는 시스템으로, 원자로의 이상이 발생했을 때 필요한 안전 조치를 취하는 역할을 담당한다. 안전계통에 속하는 제어시스템으로는 원자로보호계통(RPS: Reactor Protection System), 원자로노심보호계통(RCOPS: Reactor Core Protection System), 공학적안전설비-기기제어계통(ESF-CCS: Engineered Safety Features-Component Control System), 주요변수지시 및 감시계통(QAIS-P: Qualified Indication and Alarm System-Post accident monitoring instrumentation) 등이 포함된다. 기존 원자력 발전소의 안전계통의 경우 고도의 신뢰성을 확보하기 위하여 대부분 아날로그 제어 회로 및 기계식 릴레이로 설계되었으나, 아날로그 부품의 생산 감소 및 중단에 따라 최근의 원자력 발전소에서는 디지털 제어 기기를 적용한 안전계통의 설계가 보편화되고 있다[1,2].

안전계통에 사용되는 디지털 제어기는 고도의 신뢰성을 확보하기 위하여 최고 안전 등급 기준인 1E등급에 맞춰 설계되어야 하며, 오류 발생 시 시스템 정지를 최소화 하기 위하여 다중화 구조의 내고장성 설계기법이 필수 요건으로 요구되고 있다[3-7]. 이러한 요구조건을 충족하기 위해서 현재 사용중인 1E등급의 PLC (Programmable Logic Controller)는 이중화 혹은 삼중화 구조로 설계되어 있다. 또한 현재 설치된 PLC의 안정성과 가용성을 개선한 차세대 고신뢰성 제어기인 SPLC (Safety PLC)의 개발과 관련 기술의 연구도 활발하게 진행 중에 있다[1].

SPLC에서는 백플레인 버스, 통신망과 같은 수동형 모듈(passive module)은 이중화 구조로 설계하고 입출력 모듈, 프로세서 모듈들은 삼중화 구조를 갖는다. 다만 삼중화된 모듈에서 고장이 발생하면 자동적으로 이중화 혹은 단일 모드로 동작하도록 설계되어, 시스템의 가용성을 높일 수 있는 구조로 설계되어 있다[8]. 이와 같은 구조는 고도의 신뢰성이 요구되는 다양한 제어분야에 적용될 수 있는 유연한 구조이다.

다중화된 모듈간의 효율적인 데이터 공유를 위해서는 고속 시스템버스가 필요한데 SPLC에서는 직렬통신(serial link)을 통하여 프로세서모듈, 입출력모듈, 통신모듈 상호간에 데이터를 공유한다. 다중화된 모듈로부터 전송된 데이터는 데이터의 수신측에서 선별(보팅) 알고리즘을 통하여 유효한 데이터를 판단한다[1]. 예를 들어 SPLC에서 다중화 된 입력 모듈들은 다중화된 프로세싱 모듈로 각각의 입력 데이터를 전송하고 개별 프로세싱 모듈에서 입력 데이터에 대한 데이터 선별(보팅) 알고리즘을 통해 유효한 데이터를 독립적으로 판단하게 되는데 이때 소프트웨어적 시간 지연이 발생한다.

본 논문에서는 이러한 다중화 데이터 처리 과정, 즉 보팅 알고리즘을 시스템 버스 수준에서 수행할 수 있는 새로운 시스템 버스를 제안함으로써 보팅에 필요한 시간을 줄이는데 목적을 둔다. 또한 제안된 시스템버스 프로토콜을 FPGA로 구현한 결과를 보인다.

* Corresponding Author
Manuscript received December 9, 2013 / revised March 14, 2014 / accepted April 7, 2014
정우혁: 인하대학교 전자공학과(whjeong@emcl.org)
박재현: 인하대학교 정보통신공학부(jhyun@inha.ac.kr)
※ 본 논문은 인하대학교의 지원으로 연구되었음.

II. 다중화 제어기 구조

1. SPLC 구조

SPLC의 제어기 기본 단위는 입출력 모듈, 프로세서 모듈, 통신 모듈, 전원 모듈, 버스 모듈로 구성되는 제어노드(단일 랙)로서, 그림 1과 같이 각 모듈의 성격에 따라 표 1과 같이 단일 랙 안에서 이중화 혹은 삼중화 모듈들로써 구성된다. 기본적으로 데이터 처리 및 판단 기능이 없는 버스 모듈과 통신 모듈은 이중화로 구성하고 데이터 처리가 필요한 입출력 모듈과 프로세서 모듈은 삼중화로 구성되어 있다. 각 모듈은 직렬버스로 구성된 시스템버스로 연결되어 있다.

2. 데이터 처리

이중화 구조를 갖는 모듈은 상시대기방식(hot-standby)으로 동작한다. 즉, 수신 모듈에서 박동 신호, CRC (Cyclic Redundancy Check) 등의 진단 기능을 통해 송신 모듈의 이상 유무를 판단하여 주모듈에 이상이 없을 경우 주모듈(primary)로부터 수신한 데이터를 이용하고, 주모듈의 이상을 감지할 경우 부모듈(secondary)로부터 수신한 데이터를 이용한다. 두 모듈이 모두 고장 날 경우 안전조치(safe action)를 위한 출력을 생성한다.

삼중화 구조에서의 데이터 처리는 건전한 모듈의 개 수에 따라 다른 결정 방식를 따른다. 그림 2는 삼중화 모듈에서의 데이터 처리 방법을 도식적으로 나타낸 것으로 기본적으로 디지털 데이터는 2/3 보팅(voting)을 실시하고 아날로그 데이터는 중간 값을 취한다. 하나 이상의 모듈에 대해 고장이 감지되면 디지털 데이터의 경우 그림 2(a)에서 보이는 것과 같이 건전한 모듈이 생성한 두 값이 동일한 경우는 그 값을, 그렇지 않은 경우 안전조치 값을 유효한 데이터로 판단하게 된다. 아날로그 데이터의 경우 그림 2(b)와 같은 논리로 건전한 모듈이 생성한 데이터의 평균값을 유효한 데이터로 사용한다[1].

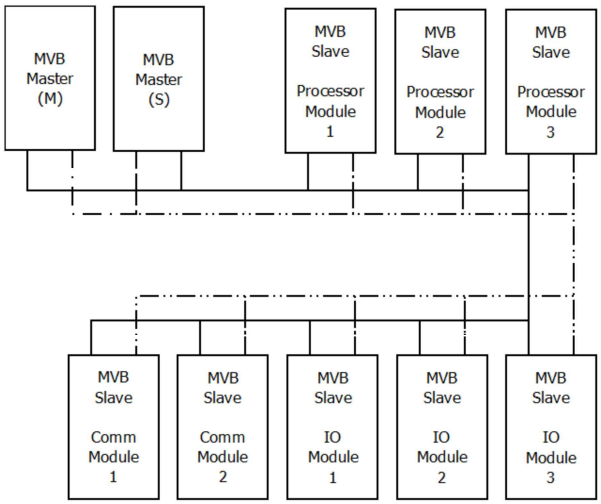


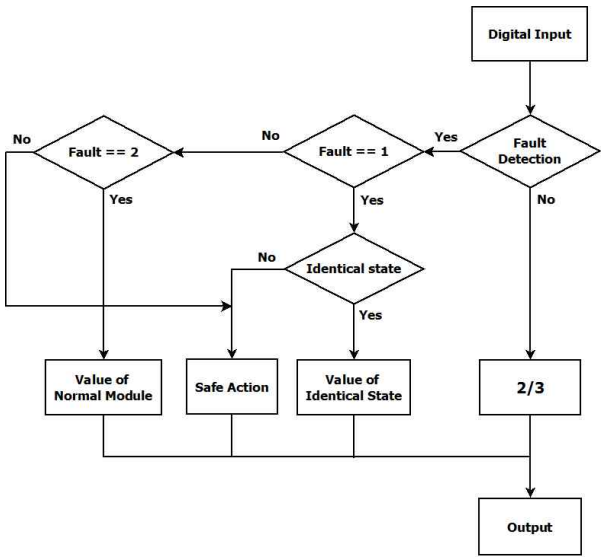
그림 1. SPLC 시스템버스 구성.
Fig. 1. System bus configuration.

표 1. SPLC 설계 사양 [1].
Table 1. Specification of SPLC [1].

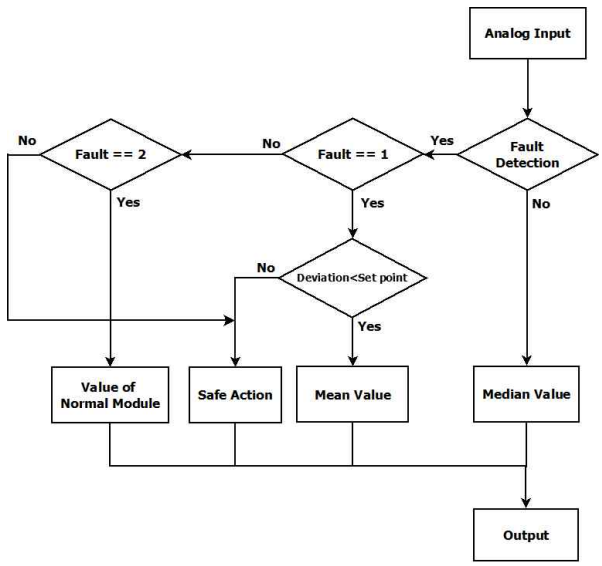
항목		SPLC
다중화구조 (단일랙)	버스	Serial, 이중화
	통신	이중화
	입출력	선택적 다중화(최대 삼중화)
	프로세서	선택적 다중화(최대 삼중화)
	전원	이중화
프로세서 모듈		266 MHz CPU
통신모듈		20Mbps, 64노드 지원
운영체제	스케줄링	고정우선순위
	진단 기능	제어기 및 사용자 프로그램 진단

3. 직렬 버스

SPLC의 제어노드를 구성하는 입출력 기기와, 프로세싱모듈, 통신모듈간에 주기적인 데이터를 공유를 위해서 신뢰성이 보장되는 시스템 버스가 필요한데 SPLC의 시스템 버스로 EtherCAT, CAN 등의 다양한 통신망을 제안하고 있다[1]. 본 논문에서는 실시간 데이터 통신이 보장되고 PLC의 입출력 신호와 같은 상태데이터(state data)의 전달에 적합한 TCN

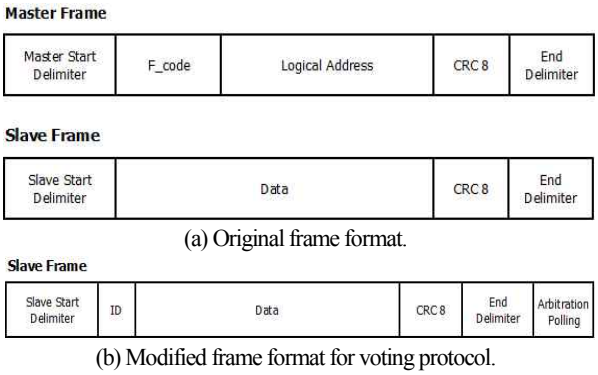


(a) Processing of digital data.



(b) Processing of analog data.

그림 2. 삼중화 모듈에서의 데이터의 처리.
Fig. 2. Processing of triple redundant data.



(a) Original frame format.

(b) Modified frame format for voting protocol.

그림 3. MVB 프레임의 포맷.

Fig. 3. MVB frame format.

(Train Communication Network)에 정의된 MVB (Multi-function Vehicle Bus) 통신 프로토콜을 기본 프로토콜로 이용하기로 한다[9]. 다만 MVB 표준은 전송속도로 1.5Mbps를 규정하고 있는데, 이는 SPLC에 활용하기에는 부족하므로, 전송속도를 100Mbps로 높여서 사용하는 것을 가정한다.

MVB 상의 데이터 교환은 기본주기(basic period)에 맞춰 버스 관리자에서 전송되는 마스터 프레임(MF: Master Frame)에 의해 이루어진다. 각 슬레이브 장치는 슬레이브 프레임(SF: Slave Frame)을 통하여 데이터를 전송하거나(source), 수신(sink)할 수 있다. MVB의 프레임 구조는 그림 3(a)와 같다[9]. 마스터 프레임은 16비트의 길이를 가지며, 프로세스 데이터는 펑션 코드(F_Code)에 따라 16, 32, 64, 128, 256 비트의 데이터 길이를 갖는다. 본 논문에서는 현재 표준에서 사용하지 않는 F_Code에 디지털 삼중화 데이터를 위한 F_Code를 정의하고 이들 데이터에 대한 프로토콜을 구현하므로써 기존 프로토콜과 호환성을 유지하도록 하였다.

III. 다중화 보팅 시스템 버스

SPLC의 다중화 데이터 처리 방식은 앞서 설명한대로 프로세싱 모듈 혹은 출력 모듈과 같이 데이터를 처리하는 모듈에서 다중화된 데이터 생성모듈(입력 모듈 등)로부터 데이터를 수집한 후 그림 2에 따른 보팅 알고리즘을 수행한다. 이와 같은 보팅 알고리즘은 디지털 데이터의 경우 각 비트 별로 수행되어야 하므로 전송된 데이터의 길이가 긴 경우 많은 시간이 소요된다. 예를들어 시스템 버스를 통하여 전송된 데이터가 32비트의 디지털 데이터의 경우, 32번의 보팅 연산을 수행하여야 한다. 따라서 본 논문에서는 디지털 데이터에 대하여 보팅 알고리즘을 효율적으로 지원하기 위한 시스템 버스 프로토콜을 제안한다. 제안된 보팅 프로토콜은 MSC (MVB Slave Controller)에서 데이터 프레임 처리와 SPLC 삼중화 보팅이 동시에 이루어지는 프로토콜로서, 데이터에 대한 선별작업이 시스템버스를 통한 통신과정에서 하드웨어적으로 이루어짐으로써 소프트웨어의 부담을 덜 수 있으며, 고속으로 수행되는 장점이 있다. 다만, 아날로그 데이터에 대한 보팅은 기존의 방법대로 각 프로세싱 혹은 출력모듈에서 소프트웨어적으로 실시하도록 한다.

1. 버스 프로토콜

제안하는 프로토콜은 MVB 프로토콜을 기반으로 SPLC

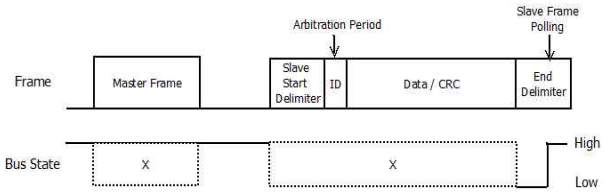


그림 4. 전송 주기에 따른 시스템 버스 상태.

Fig. 4. State of system bus by transmission period.

삼중화 데이터의 보팅을 시스템 버스 수준에서 구현하는 것이다. MVB는 버스 관리자가 MF을 통해 Slave를 Polling하여 데이터 교환이 이루어진다. 하지만 하나의 MF은 하나의 SF만을 Polling 하므로 삼중화 데이터를 처리하기 위해서 버스 관리자는 총 세 번을 Polling해야 한다. 이와 같이 동일한 데이터를 삼중화된 모듈로부터 반복적으로 전송하는 경우, 한 번의 Polling으로 세 모듈이 모두 데이터를 전송하도록 개선하여 데이터 전송시간을 줄일 수 있다. 그러나 동일한 MF에 대해 세 모듈이 모두 데이터를 전송하기 위해서는 상호간에 충돌이 발생하지 않도록 전송 주기 동안 버스를 중재(arbitration)해야 한다. 이를 위하여 제안된 프로토콜은 그림 3(b)와 같이 표준 SF에 모듈의 ID를 추가한 형태의 변형된 SF를 사용한다. 그림 4는 버스 중재를 위해 수정된 전송 주기에 따른 버스 상태를 나타낸 것이다. ID 필드에서 각 모듈의 전송 여부가 결정되며, End Delimiter의 상승 Edge에서 동일한 MF에 대한 다른 모듈로부터의 전송 주기가 시작되거나 해당 MF에 대한 전송 주기가 종료된다.

MVB 버스에 연결된 삼중화 모듈은 보팅 시스템의 중재에 사용되는 2 bit의 ID를 할당 받는다. 데이터 전송을 위해서는 버스 관리자가 MF을 전송한 후에 Source인 Slave들이 SF을 전송하는데, SF의 Slave Start Delimiter을 전송한 후에 자신의 ID를 MSB로부터 LSB 순서로 전송한다. ID는 버스상에서 Wired-OR 로직에 따라 0과 1이 중첩되면 1로 읽히게 된다. 예를들어 ID가 0, 1, 2인 세 개의 모듈이 동시에 ID를 전송하는 경우 MSB의 전송구간에서는 ID가 2인 노드의 MSB 값이 1이므로 버스의 신호값은 1이 된다. 0번과 1번 모듈은 자신의 ID의 MSB와 버스의 신호값이 일치하지 않으므로 더 이상 ID를 전송하지 않고 대기하게 되며 2번 모듈은 자신의 ID의 MSB와 버스의 신호값이 일치하므로 LSB를 전송한다. 각 모듈은 중재구간동안 버스의 신호와 자신의 ID가 일치하면 중재구간을 마치고 나머지 SF을 전송한다. SF 전송이 끝나면 2번 모듈은 다음 MF이 전송될 때까지 대기한다. ID가 2인 노드가 SF를 전송하면, 바로 두 번째 중재구간이 시작되는데, 이미 SF를 전송한 2번 모듈은 참가하지 않고 0번과 1번 모듈만 참가한다. 이 경우, 앞서 설명한 순서에 따라 1번 모듈이 전송 주기를 차지하여 SF을 전송하게 되고, 다시 전송 주기가 시작되면 남은 0번 모듈이 SF을 전송한다. 세 번의 전송이 끝나면 전체 전송 주기가 마무리된다.

앞선 예에서, 만일 2번 모듈이 고장일 경우 첫 번째 중재구간에서 1번 모듈이 전송 주기를 차지해서 SF을 전송한다. 다음 번 중재구간에서는 0번 모듈이 SF을 전송한다. 모든 모듈이 정상일 경우 SF 전송은 3회가 일어나야 한다. 하지만 세 번째 중재구간에서 일정 시간 동안 SF가 전송되지 않을

표 2. Register와 Fault Counter에 따른 출력 값.
Table 2. Output according to register value and fault counter.

D_1	D_2	D_3	Fault Counter		Voted Data	Safe Action	Output
			F_1	F_0			
0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0
0	1	0	0	0	0	0	0
0	1	1	0	0	1	0	1
1	0	0	0	0	0	0	0
1	0	1	0	0	1	0	1
1	1	0	0	0	1	0	1
1	1	1	0	0	1	0	1
0	0	X	0	1	0	0	0
0	1	X	0	1	X	1	Reg_S
1	0	X	0	1	X	1	Reg_S
1	1	X	0	1	1	0	1
0	X	X	1	0	0	0	0
1	X	X	1	0	1	0	1
X	X	X	1	1	X	1	Reg_S

경우 1개의 모듈이 고장 난 것으로 판단할 수 있다. 마찬가지로 두 번째 중재구간에서 일정 시간 동안 SF가 전송되지 않으면 2개의 모듈이 고장 난 것으로 판단할 수 있다. 중재구간동안 아무 모듈도 ID를 전송하지 않으면, 해당 전송주기를 종료하고 다음 데이터의 전송을 위한 MF가 전송되어 새로운 전송주기가 시작된다.

2. 디지털 데이터 보팅 알고리즘 구현

데이터를 수신하는 모듈은 그림 2에 설명된 SPLC의 보팅 논리에 따라 보팅을 실시한다. 디지털 데이터에 대한 동작을 정리하면 다음과 같다.

- 3개의 SF을 수신했을 경우 2/3 보팅을 실시하고, 최종 결과를 저장한다.
- 2개의 SF을 수신했을 경우 동일성 여부를 판단하여, 동일할 경우 수신 데이터를 저장하고 아닐 경우 안전조치(safe action)를 취한다.
- 1개의 SF을 수신했을 경우, 정상 데이터로 간주하여 수신 데이터를 저장한다.
- SF가 수신되지 않은 경우, 안전조치(safe action)를 취한다.

이를 정리한 표 2를 기반으로 보팅 로직을 논리식으로 표현하면 (1), (2)와 같다. 식에서 F_1, F_0, D_1, D_2, D_3은 각각 고장모듈 갯 수(fault counter)의 Bit 1, 0과 세 개의 모듈로부터 수신된 데이터 비트를 의미한다. 안전조치 출력을 의미하는 Safe Action 신호가 0일 때 Voted Data가 출력되고, Safe Action이 1일 때 사전에 정의된 안전조치에 해당하는 값(Reg_S)가 출력되므로, 전체 출력 값(output)의 논리식은 식 (3)과 같다.

$$Voted\ \ Data\ = D_1 \cdot D_3 + F_1 \cdot D_1 + D_1 \cdot D_2 + \overline{F_1} \cdot D_2 \cdot D_3 \quad (1)$$

$$Safe\ \ Action\ = F_1 \cdot F_0 + F_0 \cdot D_1 \cdot \overline{D_2} + F_0 \cdot \overline{D_1} \cdot D_2 \quad (2)$$

$$Output\ = Voted\ \ Data\ \cdot \overline{Safe\ \ Action\ } + Reg_S \cdot Safe\ \ Action \quad (3)$$

IV. 성능 분석

1. 데이터 전송 시간

표준 MVB 시스템을 사용하여 삼중화 데이터를 전송하기 위해서는 3번의 Polling 사이클을 거쳐야 한다. 따라서 삼중화 된 데이터를 전송하는데 걸리는 시간 $t_{m\,vb}$은 식 (4)와 같다. t_p, t_m, t_s 는 각각 Polling 주기, Master Frame 전송 시간, Slave Frame의 전송 시간을 의미한다.

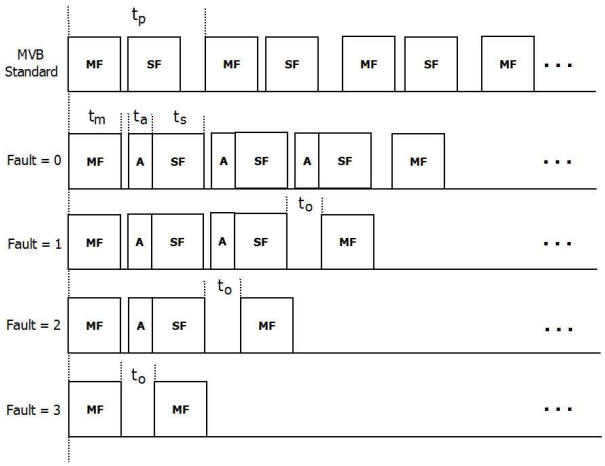


그림 5. 고장 모듈 수에 따른 버스 상태.
Fig. 5. State of system bus by the number of fault.

$$t_{m\,vb} = 3 \cdot t_p \geq 3(t_m + t_s) \quad (4)$$

하지만 제안된 시스템을 사용한 경우 1번의 기본 주기로 삼중화 데이터를 전송하고 처리할 수 있다. 고장 모듈 수에 따른 버스 상태는 그림 5에 보이는 것과 같으며, 이 경우 버스를 통한 전송시간은 식 (5)와 같다. 식에서 t_a, t_o 는 각각 Arbitration 시간, 전송 대기 초과 시간을 의미하며 n은 고장 모듈 수를 의미한다.

$$t_{tm\,r} = \begin{cases} t_m + 3(t_a + t_s), & n = 0 \\ t_m + (3-n)(t_a + t_s) + t_o, & n \geq 1 \end{cases} \quad (5)$$

식 (5)에서 $n = 0$인 경우, 즉 모든 모듈이 정상인 경우, Arbitration에 필요한 시간 t_a, 가 충분히 작다면, 제안된 시스템의 데이터 전송 시간은 기존 시스템에 비하여 $2t_m$만큼 개선됨을 알 수 있다. 단, t_p의 길이에 따라 성능 향상은 달라질 수 있는데, 성능 향상은 $t_p - (t_m + t_s)$에 반비례한다.

2. 보팅 처리 시간

SPLC는 CPU의 소프트웨어를 통해 보팅을 실시하며, 제안된 시스템은 하드웨어를 통해 보팅을 실시한다. 삼중화 데이터에 대한 보팅은 비트 단위로 실시되며, 보팅 결과에 따라 비트별 출력으로 선별된 데이터 또는 Safe Action 데이터가 사용된다. CPU 클럭으로 SPLC의 CPU 클럭인 266MHz(1 클럭 = 3.8 ns)를 적용하는 경우 SPLC의 보팅 처리 시간은 고장 모듈의 수와 삼중화 데이터의 값에 따라 달라진다. 비교, 덧셈, 호출, 저장, 분기 연산에 각 1 클럭이 소요된다고 가정하고, 그림 6의 의사 코드에 따라 고장 모듈의 수에 따른 SPLC의 보팅 처리 시간 t_{vs_f0}, t_{vs_f1}, t_{vs_f2}, t_{vs_f3}은 식 (6)-(9)과 같이 계산된다. 고장 모듈이 없을 때와 하나일 때는 각 데이터의 값에 따라 처리 시간이 달라지므로 최대 시간(worst case)을 나타냈고 고장 모듈이 둘, 셋일 때는 데이터의 상태와

```
for(i=0; i<n, i++)  {
   a = i th bit of register for module 0;
   b = i th bit of register for module 1;
   c = i th bit of register for module 2;
   s = i th bit of register for safe action;
   if(fault == 0)  {
      if(a == b && b == c)          output(i) = a;
      else if(a == b && b != c)     output(i) = a;
      else if (a != b && b == c)    output(i) = b;
      else                          output(i) = a;
    }
   else if(fault == 1)  {
     if(a == b) output(i) = a;    // suppose 0 and 1 are normal module
     else       output(i) = s;    // a != b
   }
   else if(fault == 2) output(i) = a;   // suppose 0 is normal module
   else                output(i) = s;   // fault == 3
}
```

그림 6. 보팅 처리 의사 코드.
Fig. 6. Pseudo code of voting.

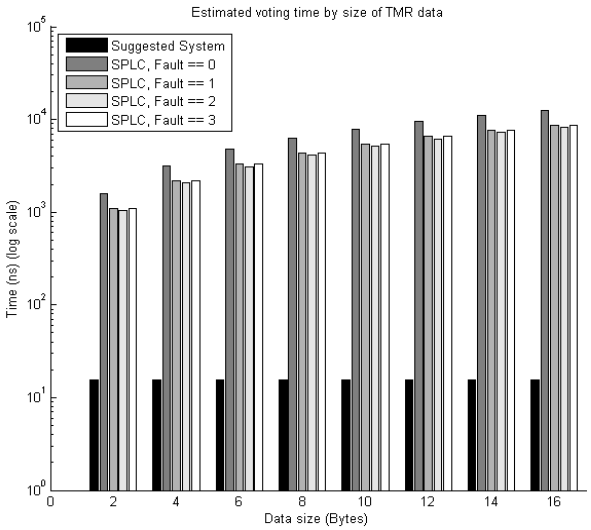


그림 7. 삼중화 데이터 크기에 따른 보팅 처리 시간.
Fig. 7. Estimated voting time by size of TMR data.

무관하게 처리 시간이 일정하다. t_{vs}는 ns 단위이며, 수식의 n은 비트 단위의 데이터 크기를 의미한다.

$$t_{vs\_f0} = 98.8 \cdot n \tag{6}$$

$$t_{vs\_f1} = 68.4 \cdot n \tag{7}$$

$$t_{vs\_f2} = 64.6 \cdot n \tag{8}$$

$$t_{vs\_f3} = 68.4 \cdot n \tag{9}$$

반면 제안된 시스템은 FPGA의 조합 논리(combinational logic) 회로를 통해 보팅을 실시하므로, 한 클럭 내에 보팅이 완료된다. 따라서 MSC 합성 결과를 고려하면 최대 시스템 클럭이 65MHz 이므로, 처리 시간 t_{vh}는 일정하게 $15.4ns$ 가 소요된다. 따라서 데이터가 32비트일 때 제안된 시스템과 SPLC의 보팅 처리 시간을 비교하면 최소 134배의 성능 향상이 있음을 확인할 수 있다. 삼중화 데이터의 크기가 증가할수록 SPLC의 처리 시간도 이에 비례하여 증가하지만, 제안된 시스템은 하드웨어 기반이므로 처리 시간이 데이터의 크기에 관계 없이 동일하다. 따라서 데이터 크기가 증가할 수록 제안된 시스템이 유리함을 알 수 있다.

V. 버스 컨트롤러 구현

본 논문에서 제안하는 보팅 시리얼 버스의 타당성을 검증하기 위하여 Xilinx Spartan-6 XC6SLX9 FPGA에 구현하여 실험을 진행하였다. 주기적 실시간 데이터 전송을 위한 F_code 0~4와 본 논문에서 제안하는 TMR (Triple Modular Redundancy) 데이터 보팅을 위한 F_code 5를 구현하였다. 그림 8은 구현된 MSC의 내부 구조를 보이고 있는데, 크게 전송부(TX), 이중화된 수신부(RX), 제어부(main controller), 타이머(timer), 데이터 및 상태 메모리, 메인 컨트롤러와 인터페이스를 위한 SPI 인터페이스, 그리고 보팅 시스템의 중재부(arbitration)과 선별부(voter)로 구성된다.

그림 9와 그림 10은 구현된 MSC의 동작을 검증하기 위한 실험 결과이다. 그림에서 첫 번째 신호는 버스의 전체 전송

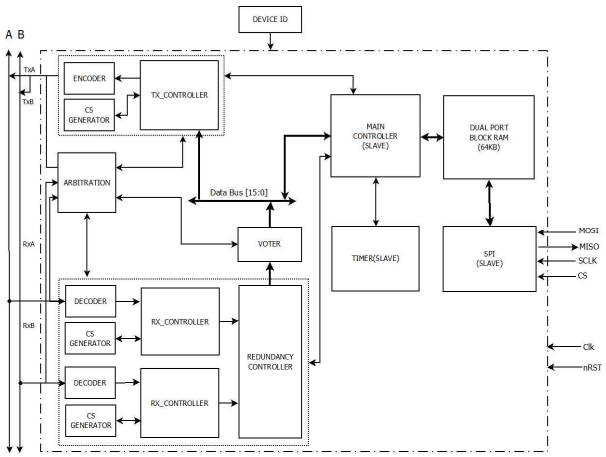


그림 8. MSC 내부 구조 블럭도.
Fig. 8. Block diagram of MSC.

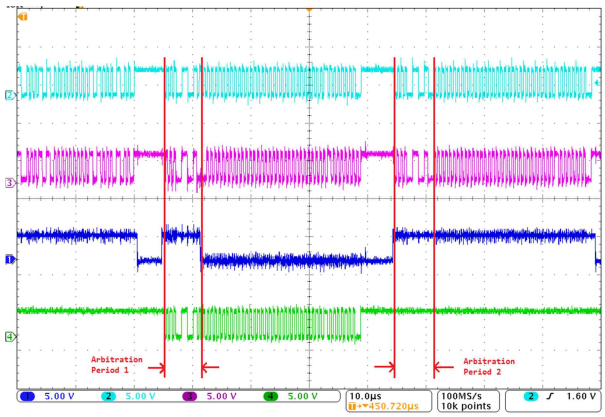


그림 9. 버스 마스터의 TMR 데이터 Polling에 대한 응답.
Fig. 9. Response for the polling TMR data from bus master.

신호를 측정한 것이며, 두 번째 신호는 0번 모듈의 전송 신호를 측정한 것이다. 그리고 세 번째 신호는 0번 모듈의 송수신 방향을 측정한 것이다. 즉 High이면 전송 Low 이면 수신을

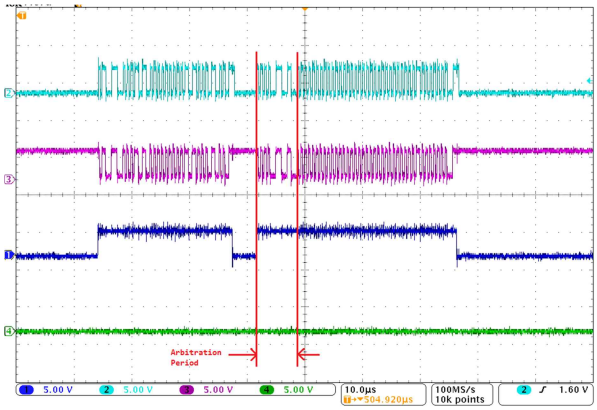


그림 10. 1번 모듈(ID = 1, 2)이 고장 난 경우의 응답.
Fig. 10. Response for the case of broken module 1(ID = 1, 2).

의미한다. 마지막 네 번째 신호는 1번 모듈의 전송 신호를 측정한 것이다. 그림 9는 0번과 1번 모듈이 모두 정상일 때의 응답 SF을 측정한 것이다. 중재구간1(Arbitration Period 1)에서 상위 ID인 1번 모듈이 전송 주기를 차지해서 데이터를 전송하는 것을 확인할 수 있다. 그리고 중재구간 2에서 0번 모듈이 버스를 획득하여 전송 모드가 수신 모드로 전환한 것을 세 번째 신호를 통해 확인할 수 있다. 그림 10은 1번과 2번 모듈들이 고장 난 경우의 SF 응답을 측정한 것이다. 중재 주기(arbitration period)동안 고장난 모듈 1, 2는 ID를 전송하지 않으므로, ID인 0번 모듈이 전송 주기를 차지하여 데이터를 전송하는 것을 확인할 수 있다.

VI. 결론

본 논문은 원전 안전등급 제어기기인 SPLC (Safety Programmable Logic Controller)의 다중화 구조를 지원하는 시리얼 버스 구조를 제안하고 있다. SPLC는 신뢰성 확보를 위하여 삼중화 보팅 구조로 설계되어있는데, 삼중화 데이터는 입력 받는 모듈에서 각자 보팅을 실시하도록 설계되어 있다. 그러나 제안된 시리얼 버스는 삼중화된 데이터의 보팅을 버스 수준에서 하드웨어적으로 실시함으로써, 보팅에 따른 데이터 처리 시간을 줄이도록 설계되어 있다.

이를 위해 시스템 버스로 Master/Slave 구조의 결정론적 통신망인 IEC 61375 MVB를 택하고, MVB 버스의 전송 주기의 개선을 통해 삼중화 데이터 처리의 성능과 효율성이 증가된 보팅 프로토콜을 구현하였다.

제안 된 시스템 버스 보팅 프로토콜은 버스 관리자의 한 번의 Polling만으로도 삼중화 데이터를 모두 교환할 수 있고, 고장 모듈 수에 따라 버스의 데이터 교환 주기가 유동적이므로 버스의 효율성이 증가되며, 또한 전송 Slave Frame의 ID를 통해 고장 모듈을 판단할 수 있으므로 다중화 구조 SPLC의 시스템 버스로 적합함을 알 수 있다.

REFERENCES

[1] G. S Son, D. H. Kim, C. W. Son, J. K. Kim, and J. H. Park, "Design of SPLC architecture used in advanced nuclear safety system and reliability analysis using markov model," *Nuclear Technology*, vol. 184, pp. 297-309, 2013.
[2] K. C. Kwon and M. S. Lee, "Technical review on the localized digital instrumentation and control systems," *Nuclear Engineering and Technology (in Korean)*, vol. 41, no. 4, pp. 447-454, 2009.
[3] IEEE, "Standard for Qualifying Class 1E Equipment for Nuclear Power Generating Stations," *IEEE 323*, 2003.
[4] V. B. Prasad, "Fault tolerant digital systems," *Potentials, IEEE*, vol. 8, no. 1, pp. 17-21, 1989.
[5] C. A. L. Lisboa, E. Schuler, and L. Carro, "Going beyond TMR for protection against multple faults," *18th Symposium on Integrated Circuits and Systems Design*, pp. 80-85, 2005.
[6] V. P. Nelson, "Fault-tolerant computing: fundamental concepts," *Computer, IEEE*, vol. 23, no. 7, pp. 19-25, 1990.
[7] B. Frogner and H. S. Rao, "Control of nuclear power plants," *IEEE Transaction on Automatic Control*, vol. 23, no. 3, pp. 405-417, 1978.
[8] J. P. Noh, J. H. Park, K. S. Son, and D. H. Kim, "Reliability analysis of redundant architecture of dependable control system," *Journal of Institute of Control, Robotics and Systems (in Korean)*, vol. 19, no. 6, pp. 328-333, 2013.
[9] IEC, "International standard for electric railway equipment-train bus-Part 1: Train communication network," *IEC 61375-1*, 2007.
[10] J. Y. Sul, K. C. Kim, Y. S. Kim, and J. H. Park, "Implementation of high-reliable MVB network for safety system of nuclear power plant," *The Trans. of the Korean Institute of Electrical Engineers (in Korean)*, vol. 61, no. 6, pp. 859-864, 2012.

정 우 혁

2012년 인하대학교 정보통신공학부 학사. 2014년 인하대학교 대학원 전자공학과 석사. 관심분야는 무선센서 네트워크, 임베디드시스템, 실시간네트워크.

박 재 현

1986년 서울대학교 제어계측공학과 학사. 1998년 동 대학원 석사. 1994년 동 대학원 박사. 1995년~현재 인하대학교 정보통신공학부 교수. 관심분야는 임베디드시스템, 실시간네트워크.