

다중 입출력 검사 방식의 세분화된 스케줄링을 통한 개선

박원찬*, 류환규*, 김정호**, 안기홍**

*(주) 로템기술

**한밭대학교 컴퓨터공학과

e-mail : parkwc@rodemeng.com, hgryu@rodemeng.com, jhkim@hanbat.ac.kr, khahn@hanbat.ac.kr

Improvement in Multiple Input-Output Inspecting Method through Classified Scheduling

Wonchan Park*, Hwanky Ryu*, Junggho Kim**, Kihong Ahn**

*Rodem Engineering, Inc.

**Dept. of Computer Engineering, Hanbat University

요 약

복잡한 구조를 가진 케이블의 빠르고 정확한 검사를 위해 다중 입출력 검사 장치가 사용된다. 기존의 검사 장치의 제어 방법은 한번에 여러 가지 회선 특성을 가진 케이블의 검사가 불가능하며, 장치 안정성 및 검사 성능에 있어서 개선의 여지가 있으므로 이를 해결하기 위해서 본 논문에서는 입출력 장치의 검사 시퀀스 및 하드웨어 개량을 통한 개선 방법을 제안한다. 제안된 방법은 검사 명령 및 검사 결과를 저장하는 데이터 패킷에서 검사 명령을 좀더 세분화하여 패킷을 최적화하고 중복 명령을 제거하였으며, 입출력 신호 충돌 가능성을 차단하였다. 또한, I2C 버스 양방향 회로를 사용하여 전압 레벨 차이를 극복하고 활용성을 높일 수 있었다.

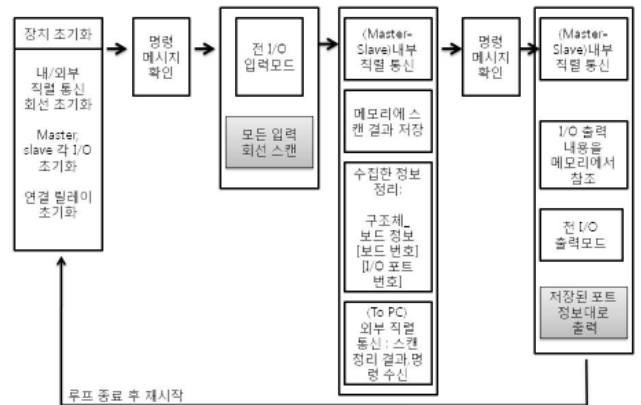
1. 서론

복잡한 netlist 구조를 가진 케이블을 빠르고 정확하게 검사하기 위해 일반적으로 다중 입출력 검사 장치가 사용되고 있다. 이는 시각검사와 같은 인간의 오감을 이용한 검사작업을 수행할 경우 단순반복적인 특성으로 인해 피로가 누적되어 생산성 향상이 매우 어렵기 때문이다. [1] 다중 입출력 검사장치를 통해 여러 개의 단자를 가진 케이블의 말단에 신호를 주고 받으며 무한 루프 검사를 통해 입출력 신호의 변화를 바로 감지하는 방식으로 검사가 이루어진다.

[그림 1]은 기존의 다중 입출력 검사 장치의 검사 방식을 나타낸 것이다. 이러한 방식은 단일 방식의 검사만 가능하며, 수백여개에 이르는 전체 회선을 모두 순차적으로 빠짐없이 무한 루프 방식으로 처리하므로 전체 시퀀스가 필요 이상으로 길어지게 된다. 또한, 검사기 말단의 입출력 신호가 충돌하게 되는 경우에 로직 게이트 회로가 파손되는 경우가 발생한다.

이러한 기존의 검사장치를 개선하기 위해 본 논문은 다음과 같이 검사 방식을 세분화하고 하드웨어를 개선하는 방식을 제안한다. 우선, 입력 신호 검사의 중복을 제거하고, 검사 제어용의 메시지 종류를 세분

화시켜서 제어 명령의 종류를 늘이고 전체 패킷(최대 4096bit) 중 검사 결과 데이터의 비중을 줄이도록 하였다.



(그림 1) 기존 검사 시퀀스

또한, 하드웨어 변경을 통해 CPU 로직 레벨보다 높은 신호와의 다중 양방향 통신이 가능하도록 하고

신호 손실을 줄이도록 하였다.

2. 세분화된 스케줄링 방법

다수의 입출력 회선을 검사하기 위해 하나의 Back Plane 블럭(이하 BP)과 내부 직렬 통신을 수행하는 10 개의 Slave 블럭이 연결되고, 각각의 Slave 블럭은 64 개의 입출력 포트를 제어하고, BP 는 다시 외부의 PC 의 USB 통신을 통해 검사 제어 명령을 받고 검사 결과를 전송한다. BP 보드에서는 내부 직렬 통신 (BP~Slave)을 수행하고, PC 와 외부 직렬 통신을 수행한다.

전체 검사 과정에서 I/O 성능은 주 프로세서 속도보다 항상 떨어지므로 전체 시스템에 병목현상을 야기할 수 있다.[2] 때문에 기존의 입출력 일괄 처리 방식 대신에, 신속성을 높이기 위하여 주 프로세서에서 I/O 제어를 위한 명령을 좀더 최적화하여 필요한 I/O 에 우선 접근하는 방식을 사용하였다.

본 시스템의 스케줄링 방법은 총 640 개의 입출력 포트를 개별적으로 제어할 수 있도록 한다. 이를 위해 제어를 위한 함수들을 다음 [표 1]과 같이 일괄 제어용 함수와 개별용으로 나누어 구성하였다.

[표 1] 스케줄링을 위한 함수 목록

함수명	기능
OnInitBoardReq	체결된 slave 블럭 인식, 필요없는 블럭 스캔 요청 방지 CMD_INIT_BOARD_RESP 메시지 생성
OnCardStatusReq	Slave 블럭의 종류에 따른 검사 방식 확인 CMD_CARD_STATUS_RES 메시지 생성
OnSpecNodeOutReq	특정 Slave 블럭 번호와 I/O 번호의 출력 CMD_SPEC_NODE_OUT_RESP 메시지 생성
OnSpecNodeReadReq	특정 Slave 블럭 번호와 I/O 번호의 입력 스캔 CMD_SPEC_NODE_READ_RESP 메시지 생성
OnSingleInitBoardReq	CMD_SINGLE_INIT_BOARD_RESP 메시지 생성
SingleCabletestStart	단일 slave 블럭 검사 시작
MultiCabletestStart	여러 개의 slave 블럭 검사 시작
OnSpecNodeALLReadReq	모든 입력 신호 스캔

OnSpecNodeALLStatusReadReq	모든 입력 신호/slave 블럭/상태 스캔
Board_init_Relay_Req	Slave 블럭 중 Relay 종류를 모두 초기화 CMD_Led_All_Init_RESP 메시지 생성
Board_init_LED_Req	Slave 블럭 중 LED 종류를 모두 초기화
Board_init_IO_Req	Slave 블럭 중 IO 종류를 모두 초기화 CMD_IO_All_Init_RESP 메시지 생성

또한, BP 블럭과 Slave 블럭들 간의 통신에는 다음과 같은 데이터 형식을 사용하였다.

- (1) data[40] : 각 명령에 필요한 주소 정보(블럭 번호, I/O 번호, 실행 결과)를 포함
- (2) portData[8] : 각 I/O 포트 입력 스캔 결과를 8 개 단위로 저장
- (3) result_data[] : slave 블럭의 검사 경과를 저장. 검사 된 수가 100 개를 초과하면 무한루프를 방지하기 위해 검사를 정지
- (4) gBoardInfo.portTable[board_max][port_max] : 이전 루프에서 저장된 블럭 및 port 별 I/O.
- (5) ScanBoardInfo.portTable[board_max][port_max] : 스캔된 블럭 및 port 별 I/O 정보를 저장. gBoardInfo.portTable 와 비교하여 입력신호 중복 스캔을 방지함.
- (6) addr : 각 블럭 별 출력 설정 핀 결정

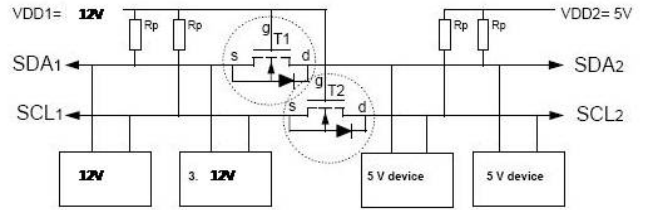
BP 블럭과 PC 간의 외부 직렬 통신에는 최대 4096bit 의 크기를 갖는 데이터 형식을 통해 통신 기능을 수행하도록 하였다. PC 로 전송하는 데이터 형식 과 그 역할은 다음과 같다.

- (1) serRecvBuf[] : PC 로부터 제어 명령 수신. 앞의 [표 2]의 명령에서 발생한 메시지를 포함하며, 이에 따라 BP 블럭을 제어함.
- (2) serTransBuf[] : 데이터 적체 허용 범위 내의 크기로 검사 결과 및 검사장치 상태를 전달.

[그림 2]은 이에 따른 전체 검사 절차와 블럭 간의 통신 구조를 나타내며, 다음과 같은 검사 과정을 제어 신호에 따라 반복한다.

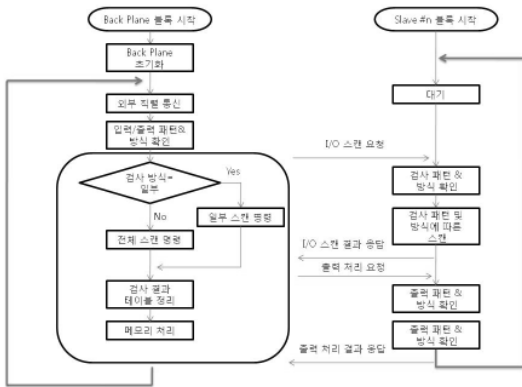
- (1) BP 블럭의 전체 기능을 초기화
- (2) PC 와의 USB 직렬통신 과정으로 검사 패턴 등의 제어 정보 수신
- (3) 검사할 입출력 패턴과 검사 방식을 확인
- (4) 일부를 우선 스캔할 경우와 전체 일괄 스캔할 경우를 구분하여 처리

- (5) 대기중이던 Slave 블럭에서 통신을 통해 스캔 요청을 수신
- (6) Slave 블럭에서 스캔 패턴 및 방식에 따라 입력 신호를 검사하고 검사 결과와 메시지를 BP 블럭으로 전송
- (7) BP 블럭에서 Slave 블럭으로 출력 패턴 및 방식 전송
- (8) Slave 블럭에서 제어 신호에 따라 출력 완료 및 확인 메시지 전송
- (9) BP 블럭에서 검사 결과 테이블을 메모리로 처리하고 USB 직렬 통신으로 보냄과 동시에 제어 정보를 수신



(그림 3) N-ch MOSFET 전압 레벨 변환 방식

아래 [그림 4]는 이를 기반으로 한 전체 하드웨어 구조를 나타낸다. BP 블럭이 3 가지 종류의 I/O 블록을 제어하는 구조로 구성하였으며, BP 블럭은 다시 PC 와 USB 통신으로 연결되어 제어신호와 검사 결과를 주고 받는다. 장치 전면에는 각 Slave 블럭의 체결 상태와 정상 작동 여부를 표시하는 인터페이스 블럭을 배치하여 이상 발생에 빠르게 대처하도록 하였다.



(그림 2) 검사 전체 절차



(그림 4) 전체 하드웨어 구조

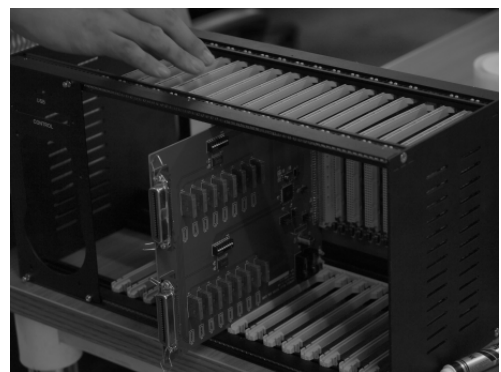
3. 검사장치 하드웨어 개선 방법

다중 입출력 검사장치의 개선을 위해 회로 개선 역시 수행하였다. 계획된 기능은 양방향 입출력 검사, 다중 출력 점점 검사, 다중 출력 LED 표시 기능의 3 가지로, 이들 기능은 각각 다른 종류의 Slave 블럭 안에 내장되었다.

검사되는 케이블의 총 길이가 수 m~수십 m 이상 되는 경우가 흔하기 때문에, 검사를 위한 신호가 중간에 약해지는 것을 방지하기 위해 검사 장치의 말단에서는 CPU 의 전압 레벨보다 높은 전압 레벨을 사용할 필요가 있다. 본 검사장치에서는 양방향 Slave 블럭 내의 CPU 전압 레벨(5V)을 높여서 12V 레벨로 양방향으로 입출력 신호를 주고받는 방식으로 설계하였다. 여기서는 N-채널 MOSFET 을 사용하여 양쪽의 전압 레벨이 차이가 있어도 정상적으로 양방향 통신이 가능하도록 하였다. [3] 또한, 신호 말단에 pull-up 저항이 연결된 open-drain 회로를 구성하여 Low-active 출력 혹은 Hi-Z 상태 중 하나가 되도록 하여 입출력 신호 충돌로 인한 회로 파손을 방지하도록 하였다. 이와 같은 구조를 [그림 3]에 표시하였다.

4. 구현된 검사장치 및 방법

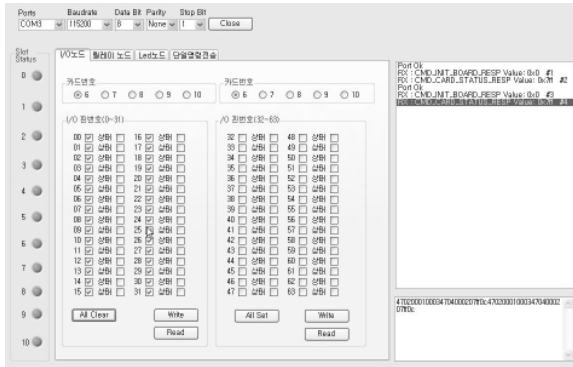
[그림 5]는 본 연구를 통해 구현한 검사장치의 구성 사진이다. 각 Slave 블럭은 체결 순서와 블럭 종류 정보를 BP 블럭에 보내어 PC 인터페이스 화면을 통해 각 Slave 블럭 상태를 확인할 수 있도록 하였다.



(그림 5) 구현된 검사장치

앞서 정의한 함수와 데이터 관리 절차를 통해 [그림 6]과 같은 PC 용 검사장치 제어 인터페이스 화면으로 검사장치의 전체 기능을 제어하게 된다. 인터페이스 화면은 다음 제어 기능들을 수행하도록 구현되었다.

- (1) USB 통신 설정 : 통신 포트 및 속도 설정
- (2) Slave 블럭 관리 : 체결된 Slave 블럭의 확인 및 정상 상태 여부 확인
- (3) Slave 종류별 제어 인터페이스 : Slave 종류별로 세부 제어 혹은 일괄 제어 처리
- (4) 프로토콜 확인 : 전송 프로토콜 및 메시지 확인



(그림 6) PC 용 제어 인터페이스 화면

5. 결론

다중입출력 제어 검사 장치는 복잡한 신호를 신속하고 정확하게 처리하는 임베디드 장치로 산업 진보에 걸쳐 응용된다. 본 연구에서는 기존의 다중 입출력 검사 방식 알고리즘과 장치 하드웨어 및 사용자 인터페이스 개선을 통해 검사 성능과 편의성 향상을 도모하였다. 입출력 처리 부하 스케줄링의 개선을 통해 검사 처리 방식을 최적화하였다. 또한 하드웨어 개선을 통해 기존 장치의 문제점을 보완하고 PC 인터페이스를 통해 장치 상태와 검사 결과를 더욱 편리하게 확인하도록 하였다.

향후에는 더욱 최적화된 알고리즘으로 입출력 데이터와 함께 아날로그 데이터의 최적화된 처리도 함께 수행하며 대규모의 신호 처리가 가능한 입출력 제어 시스템으로의 확장을 위한 연구가 요구된다.

참고문헌

- [1] 이문규, 윤찬균, “집속케이블의 품질검사를 위한 자동시각검사 시스템”, 산업공학 제 9 권 제 1 호, 1996.
- [2] Temmy Noergaard, “임베디드 시스템 아키텍처”, 도서출판 ITC, 2007
- [3] PHILIPS, “Bi-directional level shifter for I²C-bus and other systems.”, AN97055, 1997