

새로운 IEEE 1394 송수신 디코딩 알고리즘과 Arbitration 회로

정회원 이 제 훈*, 박 광 로**, 서 은 미*, 조 경 록*

A New Decoding Algorithm and Arbitration Logic in IEEE 1394 Communications

Je-hoon Lee*, Kwang-roh Park**, Eun-mi Seo*, Kyoung-rok Cho* *Regular Members*

요 약

IEEE 1394 버스는 데이터 패킷 전송시 반이중(half duplex)으로, 0과 1의 두 상태를 이용하여 전송한다. 그러나 버스 자동 구성 및 중재 기간에는 양방향으로 버스 중재 선 상태(arbitration line state) 신호를 주고받으며, 이는 Z, 0, 1의 세 논리 상태를 가지고 있다. IEEE 1394 버스를 채택한 노드는 시스템에 연결시 자동으로 네트워크를 트리 구조로 구성하고, 6 비트 물리 ID를 할당하며, 이는 버스 리셋, 트리 식별, 자기 식별의 세 과정을 통해 구성된다. 또 전송할 데이터가 있는 경우 노드는 버스의 사용권을 얻기 위한 버스 중재(arbitration) 후 전송을 시작한다. 이러한 시스템 자동 구성을 위한 과정들과 버스 중재과정에서 양방향으로 아날로그 0, 1, Z의 중재 선 상태 신호를 주고받게 된다. 본 논문에서는 기존 IEEE 1394를 채택한 노드들과 호환되며, 중재 선 상태를 0과 1의 논리 상태만을 사용하여 버스 자동 구성 및 버스 중재를 디지털 회로로 구성할 수 있는 중재 선 상태 디코딩 알고리즘을 제안하였고, VHDL을 이용하여 전체 시스템의 동작을 시뮬레이션하였다.

ABSTRACT

Data communication in IEEE 1394 serial bus is based on half duplex mode using two signal states, 0 and 1. During auto configuration and normal arbitration period, IEEE 1394 serial bus employs bidirectional arbitration line state signal and three logical states, 0, 1 and Z. The node's configuration of IEEE 1394 serial bus is configured tree topology and assigned 6 bit physical Id to each node as new devices are attached and/or removed from the bus. When any node on the bus wishes to transmit data packet, it must arbitrate for use of the bus. In system auto configuration phase and bus arbitration phase, node signals are transmitted with bidirectional communications which has analog level, 0, 1, Z. In this paper, we proposed a new auto configuration and arbitration logic by define a line state decoding and interpreting rule that uses line state signal which is logical state 0 and 1. We designed a layout using VHDL and simulated a the behavior of the entire system.

I. 서 론

IEEE 1394 버스는 컴퓨터와 멀티미디어 A/V기 기 그리고 CD, DVD와 같은 대용량의 저장매체들 간 고속으로 데이터를 전송하기 위해 제안된 고성능 직렬 버스 인터페이스 표준이다. 실시간으로 멀티

미디어 데이터를 압축 없이 전송할 수 있는 100 ~ 400Mbps의 대역폭으로 규정되며, 노드간 전송거리는 4.5M를 지원한다. P1394b의 경우 3.2Gbps 전송대역과 100M의 노드간 전송거리를 규정하여 현재 홈 네트워크와 관련된 DAVIC 표준에 규정된 1Gbps 전송대역을 만족한다. 따라서 고속 데이터

* 충북대학교 정보통신공학과(leejh@hbt.chungbuk.ac.kr), ** ETRI 교환전송연구소
논문번호: 00430-1103, 접수일자: 2000년 11월 3일

전송에 적합할 뿐만 아니라 직렬 버스이기에 비교적 원거리통신이 가능하며, 자동으로 네트워크를 구성하고 ID를 할당하므로 홈 네트워크의 백본 네트워크 혹은 PC와 DTV 및 A/V기기간 클러스터링 네트워크를 구성할 미디어로 주목받고 있다^[1].

본 논문에서는 버스 자동 구성 및 버스 중재동안 사용되는 0, 1, Z의 세 논리 상태의 신호를 제어하는 아날로그 회로 블록 중 Z의 상태를 제외하고 데이터 패킷 송수신처럼 0과 1만을 사용하고, 대신 노드가 현재 전송하는 상태를 이용하여 수신된 arbitration line state를 해석하는 알고리즘을 제안하고 디지털로 버스 중재 회로를 설계하고 이를 검증하였다.

본 논문의 2장은 IEEE 1394 버스와 기존의 버스들간의 차이점 및 IEEE 1394 PHY 블록의 동작에 대해 기술하였다. 3장은 제안된 IEEE 1394 버스 자동 구성 과정의 요구 사항과 동작 알고리즘과 검증에 대해 기술하였다. 4장은 개선된 IEEE 1394 버스의 버스 자동 구성 알고리즘을 검증하기 위해 현 IEEE 1394 표준을 만족하는 버스 중재 회로를 설계하고, 이를 통해 시뮬레이션함으로써 개선된 알고리즘의 구현을 확인하는 성능 실험을 수행하였으며, 실험을 통한 분석 결과를 제시하였다. 마지막으로 5장은 향후 전망 및 결론을 기술한다.

II. IEEE 1394 개요

IEEE 1394는 기존의 다른 PC 인터페이스와는 구분되는 특징을 가지고 있다. 표 1처럼, 첫째 직렬 버스이기에 저장매체에 사용되는 IDE 혹은 SCSI의 병렬 인터페이스에 비해 장거리 전송이 가능하며,

표 1. 버스 인터페이스 비교

종류	SCSI II	USB 1.0	IEEE1394
최대노드수	15	127	63
전송속도	80MBps	1.5~12Mbps	100~400Mbps
다이아스간 거리	3M	5M	4.5M
허브필요	불필요	필요	불필요
호스트필요	필요	필요	불필요
전송모드	Asy.	Asy. & Iso.	Asy. & Iso.

■ Asy. (Asynchronous mode) : 전송중 에러발생 여부 및 발생시 재전송 요구하는 모드
 ■ Iso. (Isochronous mode) : 실시간 전송 중시
 ■ 필요 전송 속도 : MPEG1 - 1.5Mbps, MPEG2 - 3 ~ 24Mbps

400Mbps의 고속 데이터 전송을 규격화하였기에 PC 주변기기뿐만 아니라 가전 및 AV 기기들간의 네트워크에 적합한 버스 표준이다. 둘째, 기존 버스 표준들이 비동기 전송만을 허용하는데 반해 등시 전송도 지원하여 전송의 리얼타임성을 보증하기 때문에 동영상 데이터 전송에 적합한 특징을 갖는다. 마지막으로 USB의 경우도 현재 100Mbps대역에 등시 전송도 지원하지만 IEEE 1394 버스는 USB가 PC 베이스의 네트워크 환경만을 지원하는데 반해, 모든 노드들은 네트워크의 루트로 동작 가능하고 63개의 노드까지 지원하는 등 네트워크 환경에 적합한 특징을 갖는다. IEEE 1394 버스 표준은 PC 인터페이스뿐만 아니라 홈 네트워킹과 같은 소규모 네트워크 구성에 유력한 미디어로 표 2에 홈 네트워크의 백본 혹은 서브넷으로 사용될 여러 미디어들을 비교하였다.

홈 네트워크에 사용되는 미디어는 몇 가지 제한을 가진다. 첫째, 설치의 용이성이다. 이는 미디어를 맥내에 가설하는 점에서의 문제이며 따라서 USB나 IEEE 1394, gigabit ethernet의 경우 미디어를 새로 가설을 해야 한다. 이들의 경우 백본 보다 대용량의 데이터 전송이 요구되는 구역에서 클러스터링 네트워크로 사용될 수 있고, 이 경우 IEEE 1394는 호스트를 따로 필요로 하지 않기에 자유로운 네트워크 구성이 가능하고, 버스 자동 구성 지원 등 구성의 편리함과 등시 전송을 지원하므로 멀티미디어 데이터 전송에 적합하다. 따라 IEEE 1394 버스는 홈 네트워크에서 백본 네트워크와 연동되는 DTV, DVCR 그리고 A/V 기기와 PC를 연결하는 클러스터링 네트워크 미디어로서 적합하다.

그림 1은 PHY chip의 블록도이며, 크게 I/O 포트, data encoder/decoder, 중재 회로, link/PHY 인터페이스 블록으로 구성된다.

2.1 IEEE 1394 PHY Chip 블록별 기능

1) 데이터 인코더

표 2. 홈 네트워킹 미디어 비교

규격	미디어	전송속도(bps)	동작모드	전송거리
USB	USB Cable	12M/100M	Asy, Iso	5M
IEEE1394	1394 Cable	400M/3.2G	Asy, Iso	4.5M
Ethernet	LAN	10M/1G	Asy	수십M
HomePNA	전화선	1M/10M	Asy	150M
SWAP	RF	2M	Asy, Iso	40M
Bluetooth	RF	1M	Asy, Iso	10M

데이터를 전송하는 노드는 그림 2의 DS-link 인코딩 기법을 사용하여 전송할 data의 클럭 정보를 strobe에 실어, data 신호와 strobe 신호로 전송한다.

strobe 신호는 data 신호가 같은 값이 연속시 상태를 천이시킴으로써 생성된다.

2) 데이터 디코더

데이터를 수신하는 노드는 그림 2와 같이 수신된 data 신호와 strobe 신호를 exor 연산하여 데이터의 클럭 정보를 복원한다.

3) 포트 인터페이스

포트 인터페이스는 두 인터페이스 신호 TPA/TPA*, TPB/TPB*과 전력 분포 신호 VP/VG로 구성된다.

TPA/TPA*은 데이터 송수신시 data 신호를 전송하고 strobe 신호를 받으며, 버스 중재간 Arb_A_Tx를 전송하고 Arb_B를 받는다. TPB/TPB*은 데이터 송수신시 strobe 신호를 전송하고 Data 신호를 받고, 버스 중재 과정 동안 Arb_B_Tx를 전송하고 Arb_A를 받는다. 데이터 송수신시 strobe, data 신호가 0, 1의 두 논리 상태를 갖는데 비해, arbitration line state Arb_A와 Arb_B는 버스 자동 구성 및 버스 중재 과정 동안 사용되며, 신호 상태는 드라이버가 undriven되었을 때의 Z, 1로 드라이브되었을 때의 1, 그리고 0으로 드라이브되었을 때의 0의 세 논리 상태를 갖는다.

4) 버스 중재 회로

버스 자동 구성 및 버스 중재간 노드를 제어하고 버스를 통해 데이터를 전송하려는 노드에 버스 사용권을 부여한다. 자신과 연결된 다른 노드를 제어하기 위해 strb/data 출력 드라이버를 통해 Arb_A/B_Tx 신호를 전송하고, 두 비교기에서 arbitration line state신호를 입력받는다. 이 때 0, 1 그리고 Z의 세 가지 논리 상태를 가지고 있다.

이 상태를 수신단의 비교기로부터 입력된 Arb_A/B_Rx의 두 신호를 받아, 표 3의 디코딩 규칙에 의해 신호를 해석한다. 디코딩된 특정 Arb_A/B 신호를 사용하여 버스 자동 구성과 버스 중재 기간에 다양한 이벤트와 조건을 정의한다^{[2],[3]}.

수신된 신호가 동일한 경우 수신 받는 포트가 parent 노드와 연결되었는지, child 노드와 연결되었는지에 따라 그 arbitration line state를 구별되기도 한다. 다른 노드를 제어하기 위해서 Arb_A/B_Tx 신호를 전송시 표 4와 같은 값을 드라이브한다. 표

5와 같이, 해석된 arbitration line state, Arb_A와 Arb_B가 특정 값을 가질 때 노드에 제어 명령을 내린다^{[4],[5]}.

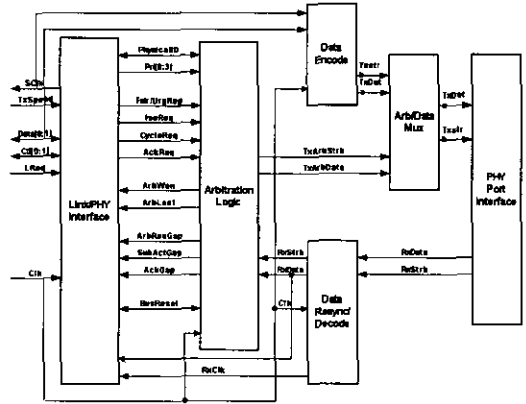


그림 1. PHY Chip 블록도

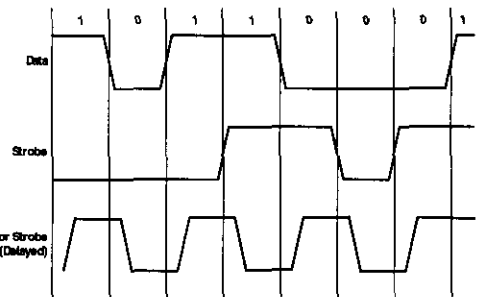


그림 2. 데이터-스트로브 인코드/디코드

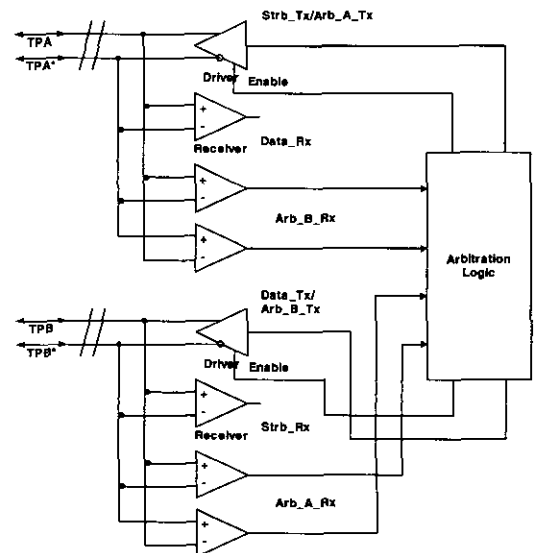


그림 3. 포트 인터페이스와 Arbitration Logic 블록도

2.2 새로운 중재 선 상태 신호 디코딩 제안

본 논문에서는 버스 자동 구성 및 버스 중재시 사용되는 선 상태를 0, 1, Z의 세 논리 상태에서 Z를 리턴던시로 처리하여, 데이터 송수신시와 같이 0, 1의 두 논리 상태를 갖는 중재 디코딩 알고리즘을 제안하였다. Z 상태를 사용하지 않는 대신 포트의 전송 신호와 노드의 상태에 따라 수신된 중재 선 상태를 표 6과 같이 해석한다.

표 3. Arbitration 신호 디코딩 규칙

Arb_n_Rx	Arb_n_Tx	Arb_n	설명
Z	Z	Z	포트는 Undriven상태
0	Z	0	
1	Z	1	
Z	0	1	dominance rule
0	0	0	상대 포트는 0/Z 송신
Z	1	1	dominance rule
1	1	1	상대 포트는 1/Z 송신

표 4. 전송한 arbitration line state

Line State		Arbitration line state
Arb_A_Tx	Arb_B_Tx	
Z	Z	IDLE
Z	0	TX_REQUEST TX_GRANT
0	Z	TX_PARENT_NOTIFY
0	1	TX_DATA_PREFIX
1	Z	TX_CHILD_NOTIFY TX_IDENT_DONE
1	0	TX_DATA_END
1	1	BUS_RESET

표 5. 수신된 arbitration line state

Line State		Arbitration line state
Arb_A	Arb_B	
Z	Z	IDLE
Z	0	RX_PARENT_NOTIFY RX_REQUEST_CANCEL
0	Z	RX_SELF_ID_GRANT RX_REQUESTSET
0	0	RX_ROOT_CONTENTION RX_GRANT
0	1	RX_PARENT_HANDSHAKE RX_DATA_END
1	Z	RX_CHILD_HANDSHAKE
1	0	RX_DATA_PREFIX
1	1	BUS_RESET

표 6. 제안된 receiving line state 디코딩 규칙

Bus Reset & Tree ID					
Arb_Tx	Arb_A/B_Rx				
	10	00	01	11	
ZZ	Parent_Notify		×	Bus Reset	
0Z	×	Root_Cont.	Parent_Hand.		
1Z	Child_Hand.	×	×		
Self_ID/Arbitration Phase					
ZZ	S0	Data_Prefix	Self_ID Grant		Bus Reset
	Data_Rx		×	Data_End	
	S2'		×	Ident_Done	
	Arb		Request		Bus Reset
Z0	×	Grant	×	×	

제안된 알고리즘 사용시 기존 data/strobe 입력 드라이버와 Arb_A/B_Rx 입력 드라이버를 각각 사용하는 대신 간단한 data/strobe 입력 드라이버를 muxing하여 사용하며, 중재 선 상태를 검출하는 두 비교기는 사용하지 않음으로써 회로의 크기를 줄이고, 예정된 arbitration line state의 입력에 따라 동작시킴으로써 오류를 최대한 피할 수 있는 이점이 있다. 또한 수신측의 arbitration line state 해석만을 고쳐 기존 표준을 따르는 기기들과도 혼용하여 사용할 수 있다. 입력 드라이버를 통해 입력된 Arb_A/B_Rx는 표 3의 디코딩 규칙을 따르지 않고, Arb_A/B_Rx 신호와 포트가 현재 전송하는 arbitration line state와 노드의 상태에 따라 표 6에 정의된 arbitration line state 입력시, 정해진 동작을 수행한다. 표 4와 같이 Z 상태의 출력을 통해 버스를 idle 상태로 드라이브하는 명령도 사용하나, 수신 시에는 Z 상태를 사용하지 않고 값이 0이건 1이건 상관하지 않는 리턴던시로 판단한다.

중재 선 상태를 사용하는 동안, 노드의 상태는 크게 버스 리셋, 트리 식별, 자기 식별, 버스 중재 과정으로 구분되며, 이는 다시 그림 4와 같이 세부 상태로 나뉜다. 각 과정은 순차적으로 수행되며, 서로 일정부분 독립적이고, 각 과정별로 순차적인 동작을 갖기에 수신된 중재 선 상태를 노드의 현 상태에 따라 세분화할 수 있다. 이렇게 함으로써, Z 상태를 고려하지 않으면서 현 표준에 따르는 기기들과 호환된다. 버스 중재 과정동안 중재 선 상태를 0과 1의 논리 상태를 갖기에, 포트 인터페이스는 더욱 간단하게 구성되며, 최소의 아날로그 블록을 사

용한 회로로 구성할 수 있고, 중재 신호간 그리고 데이터 전송간 0, 1의 두 상태만을 사용하고, 현재 노드의 상태를 참조토록 함으로써 상태를 단순화하였고 arbitration line state 송수신시 해석 오류를 최대한 피할 수 있다.

III. 제안된 arbitration line state 해석을 이용한 네트워크 자동 구성 동작

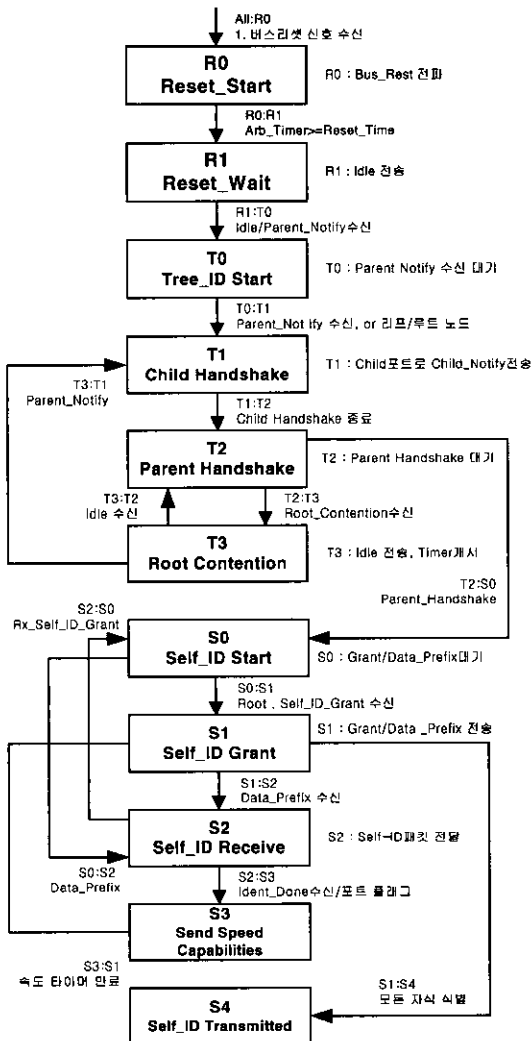


그림 4. 노드 자동 구성 상태도

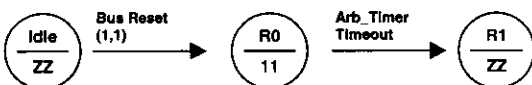


그림 5. 버스 리셋 과정 상태도

버스 자동 구성 과정은 그림 4와 같이, 버스 리셋, 트리 식별, 자기 식별 및 일반 버스 중재의 네 단계로 구성된다. 버스 리셋 과정은 구성된 시스템이 재구성의 필요가 생겼을 때 모든 노드들이 지닌 노드_ID를 비롯한 정보를 리셋한다. 트리 식별 과정은 노드들을 트리 구조로 토폴로지를 형성하며, 자기 식별 과정은 각 노드를 식별할 수 있는 물리_ID 6비트를 각각 부여한다. 이를 마치면 버스 중재 과정에 진입하며, 버스를 사용하려는 노드가 과정을 마쳐 버스 사용권을 획득하면 전송을 시작한다.

3.1 버스 리셋 과정

새로운 노드의 접속 혹은 기존 노드의 해제로, 시스템을 재구성해야 할 때, 가장 먼저 버스 리셋 과정을 통해 시스템의 버스 자동 구성 과정이 시작된다. 버스 리셋시 전이 과정은 그림 5와 같다.

버스 리셋 요구는 노드의 새로운 접속 및 해제에 의한 PHY의 전력 상태 변화, 인접 노드와의 버스 리셋 수신, S/W에 의해 초기화된 버스 리셋 요구 수신에 의해 생성된다. 버스 리셋이 발생한 노드나 bus reset을 수신한 노드는 R0:Reset Start 상태로 천이한다. R0 상태에서 접속된 다른 노드로 bus reset을 전파하고, 166.7μs의 reset_wait 시간이 지나면, R1: reset wait 상태로 천이한다. R1 상태에서는 출력을 드라이브하지 않고, 노드간 arbitration line state는 idle 상태로 유지된다. 버스 리셋 과정을 마치면, 각 노드는 자신과 접속된 포트의 수로서 접속 포트가 1개이면 리프, 그 이상이면 브랜치임을 알게 된다.

Arbitration line state 전송 예는 그림 6에 보여지며, 버스 리셋이 발생한 노드가 접속된 포트에 bus reset(1, 1) 신호를 전송하면, 수신 노드의 비교기 출력은 (1, 1)이 된다. 표 4에 의해 bus reset 수신으로 해석하고, R0 상태로 천이하며, 자기 정보를 리셋 후, 접속된 다른 노드들로 bus reset 신호를 전파한다.

3.2 트리 식별 과정

트리 식별 과정은 버스 리셋 과정 후 각 노드들을 트리 구조의 토폴로지를 구성하고, 루트를 결정토록 한다. 이 과정이 종료되면 각 노드는 접속된 포트들이 parent 혹은 child와 연결되었는지 레이블되고, 루트가 결정된다. 상태 천이는 그림 7과 같이 수행된다. 본 논문에서 제안한 디코딩 규칙은 Z 수신을 인정하지 않기에, R1 상태 진입시 버스는 idle

상태이나, 노드는 이를 수신하지 못한다. 리프 노드는 R1 상태에서 parent notify 전송 후 T0 상태로 천이하고, idle을 드라이브하는 포트에 parent notify 신호를 수신한 경우는 T1 상태로 천이하고, 이 신호의 Ack로 child notify를 전송한다. T0 상태의 leaf 노드는 parent로 예정된 노드로부터 parent handshake 수신시 포트가 Parent와 연결되었음을 레이블하고, 트리 식별 과정을 종료하고 자기 식별 과정으로 진행한다. T2 상태에서 child가 트리 식별 과정을 마치고 S0 상태로 천이한다면, parent 노드는 (1, 0)의 arbitration line state를 검출하고 rx_child_handshake 수신으로 해석한 후, S0 상태로 천이한다. 만일 parent notify 신호를 전송하고 있는 포트에 인접 노드에서 전송한 parent notify가 수신되어 중재 선 상태 신호가 (0, 0)이 되면 root contention 수신으로 해석되고, 두 노드는 idle 신호 전송후, parent-child handshaking 과정을 다시 수행하여 루트를 결정한다.

3.3 자기 식별 과정

자기 식별 과정은 트리 토폴로지로 구성들을 식별할 수 있도록 물리_ID 6비트를 부여하는 과정이다.

상태 천이는 그림 8과 같이 동작한다. 이전 과정에서 루트로 결정된 노드는 S0:self-Id start 상태에서 S1:self-Id grant 상태로 천이하고 식별되지 않은 child 노드가 있다면, 그 중 최하위 포트에 grant를, 다른 포트에 data prefix를 전송한다. grant를 수신한 노드는 S1 상태로 천이하고, 식별되지 않은 하위 포트가 없다면 parent로 data prefix를 전송하고 S4 상태로 천이한다. 이전 버스 리셋 단계의 R0 상태에서 0으로 초기화된 gap count 값을 노드 ID로 할당하고 이 정보를 담은 self ID 패킷을 다른 노드로 전송한다. 이와는 달리 부모로부터 data prefix를 수신하면, 다른 노드로부터의 self ID 패킷 수신을 대기한다. 한번 self ID 패킷 수신시 종료될 때마다, 자신의 gap count 값을 1 증가시키고, 다시 S0 상태로 천이하여 물리 ID 할당 개시를 대기한다.

3.4 일반 버스 중재 과정

버스 중재 과정은 버스를 통해 데이터를 전송하려는 노드가 버스의 사용권을 얻기 위한 과정이다. 그림 9와 같은 상태 천이로 동작한다. 이 과정을 종료하면, 버스의 사용권을 얻으려한 노드가 그에 대해 루트로부터 버스 사용권을 얻었는지를 판단하고, 얻은 경우 데이터를 전송하기 시작한다.

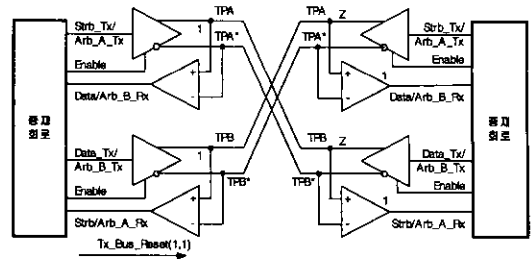


그림 6. 버스 리셋 신호 전파 예

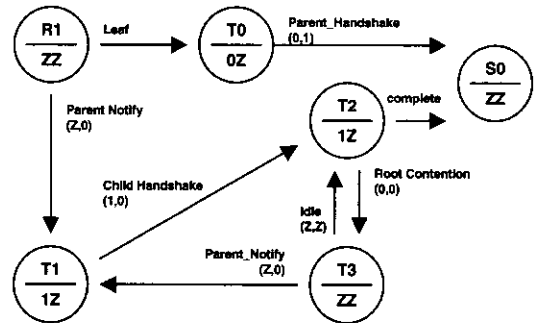


그림 7. 트리 식별 과정 상태도

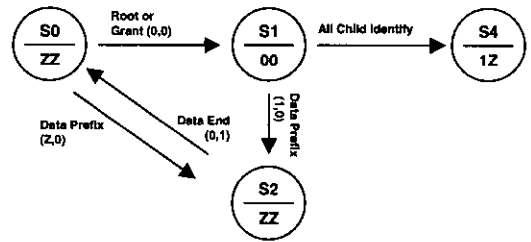


그림 8. 자기 식별 과정 상태도

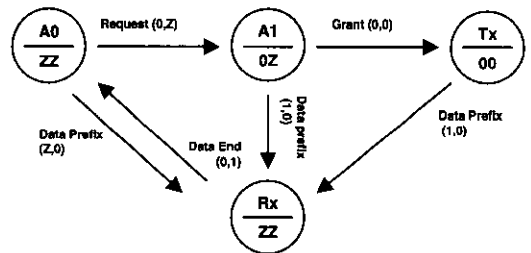


그림 9. 일반 버스 중재 과정 상태도

자기 식별 과정을 종료한 각 노드는 버스를 통해 전송할 데이터가 있을 경우 parent로 request 신호를 보내고 루트로부터 전송된 grant를 수신 받으면, 버스 사용 승인을 얻는다. grant를 수신한 노드는 Tx 상태로 천이하여 버스 사용권의 할당을 인지하

고, 데이터를 전송하기 시작한다. data prefix를 수신한 다른 노드는 Rx 상태로 천이하여 다른 노드로부터 전송되는 데이터를 수신할 준비를 하고, 데이터 전송이 끝나고 data end 신호가 수신되면, A0 상태로 천이하여 다시 버스 중재를 시작할 준비를 한다. request를 전송한 포트에서 arbitration line state가 (0,0)이 검출된 경우 grant의 수신으로, tx_request 수신시 부모로부터 grant 수신했을 경우, request cancel로 동작한다.

IV. 모의 실험을 통한 제안한 알고리즘 검증

그림 10은 제안한 디코딩 및 수신 신호 해석법을 하나의 노드에 적용하여, 버스 리셋이 발생하여, 시스템이 재구성되는 동안 노드의 상태 변화 시물레이션이다. 버스 리셋 신호 수신 후 버스 리셋 시작 상태로 천이를 시작으로 각 단계를 거쳐 데이터를 전송하기 위한 중재를 할 수 있는 A0 상태로 천이했음을 알 수 있고, 이로써, 제안된 arbitration line state 해석이 정상적으로 수행되었음을 알 수 있다.

그림 11과 같이 5개의 노드로 이루어진 네트워크 모델을 사용하여 시물레이션을 수행하였다. 노드 5가 기존 시스템에 접속하면서, 버스 리셋 신호를 전파한다. 버스 리셋과정을 거쳐, 모든 노드들은 자기 정보를 리셋한다. 자신과 접속한 포트의 수로 자신이 리프 혹은 브랜치인지 알게 된다. 그리고 트리 식별 과정에 진입하면 parent-child 관계를 핸드셰이킹하면서 트리 토폴로지를 구성한다.

만일 노드 3이 루트가 되면 그림 12와 같은 parent-child 관계를 갖는다. 그림 11에서 child와 연결된 포트는 노드 2의 0번 포트, 노드 3의 0번, 1번 포트 그리고 노드 4의 0번 포트이며, 그림 12의 트리 식별 과정에서 child와 연결된 포트는 1로 포트 레이블은 정상적으로 수행되었음을 알 수 있다. 다음 자기 식별 과정에서 각 노드는 자신을 식별할 수 있는 6비트의 물리 ID를 부여받는다. 노드 3이 루트로 결정되었기에 그림 11과 같은 물리 ID를 할당받게 된다.

그림 12는 그림 11과 같은 네트워크 모델에서의 버스 자동 구성 실험 결과이다. 버스 리셋이 발생한 후, 각각의 노드들은 그림과 같이 버스 리셋, 트리 식별, 자기 식별의 단계를 거친다. 노드 2의 0번 포트, 노드 3의 0번, 1번 포트 그리고 노드 4의 0번 포트가 트리 식별 과정 동안 child 노드와 연결된

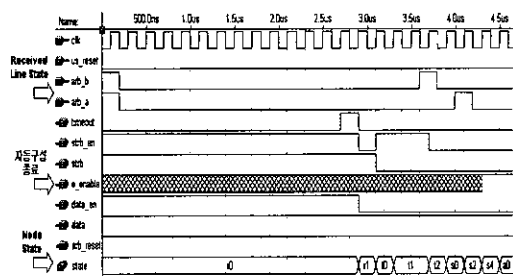


그림 10. 노드 자동 구성 모의 실험 결과

포트로, 그의 포트들은 parent 노드와 연결된 포트 로 레이블 되었다.

노드 3은 모든 포트가 자식 포트 로 레이블 되었기에 루트임을 알 수 있고, 자기 식별 과정에서 노드 #0인 물리 ID를, 노드 2는 #1인 물리 ID를, 노드 5는 #2인 물리 ID를, 노드 4는 #3인 물리 ID를, 마지막으로 루트인 노드 3은 #4인 물리 ID를 할당받았다. 이는 그림 11에서 노드 3이 루트가 되었다고 가정했을 때 계산된 값과 마찬가지로 결과를 얻었고 이는 0, 1의 두 논리 상태를 이용해도 정상적인 arbitration line state 해석을 수행했음을 알 수 있다.

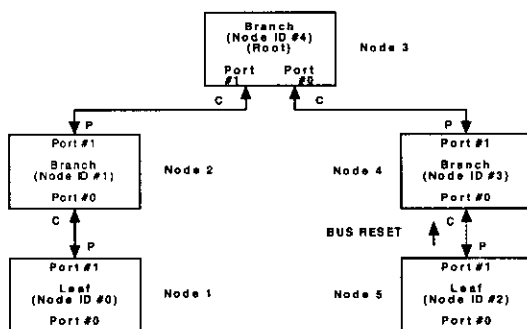


그림 11. 모의 실험의 토폴로지

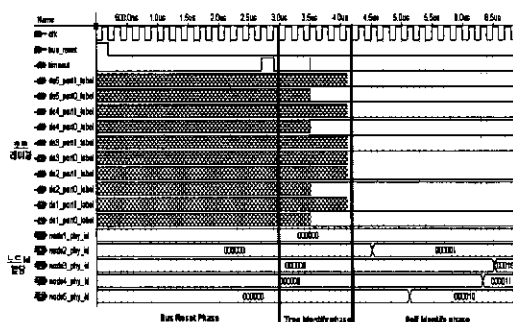


그림 12. 전체 시스템의 모의 실험 결과

V. 결론

IEEE 1394는 현재 제안되는 홈 네트워킹 미디어 중 높은 전송속도를 갖는 미디어로, 비동기 전송뿐만 아니라 멀티미디어 데이터 전송에 적합한 등시 전송도 지원한다. 시스템 재구성시 자동으로 네트워크를 구성하고 각 노드에 ID를 부여하므로 홈 네트워크 환경에 적합한 특징을 가지고 있다. IEEE 1394를 사용한 기기에서 네트워크 자동 구성 및 버스 중재 과정 동안 arbitration line state의 논리 상태는 0, 1, Z의 세 가지이고 이로 인해 아날로그 회로로 구성되었다.

본 논문에서는 0, 1, Z의 세 논리 상태의 arbitration line state를 데이터 패킷 송수신시와 같이 0과 1 상태만을 이용하는 디코딩 알고리즘을 제안하고 PHY 칩의 아날로그 블록을 디지털 회로로 구성하여, arbitration line state 해석시 노드의 현재 상태를 참조하여 해석토록 함으로써, arbitration line state의 부정확한 수신에서도, 해석 오류 발생률을 감소시킬 수 있다. 향후 과제로는 현재 네트워크 자동 구성시 노드 정보를 모두 리셋하도록 되어있으나, 이를 수정하여 리셋하지 않고, 기존 표준을 채택한 기기들과 호환되면서, 동작하도록 함으로써 버스 자동 구성 시간을 단축하고 시스템이 안정적인 동작을 할 수 있는 알고리즘을 개발한 예정이다.

참고 문헌

[1] Adam J.Kunzman, Alan T. Wetzel, "The Digital Interface for ATV" *IEEE Trans. Consumer Electronics*, vol. 14, pp 893-900, August 1995.

[2] 이강운, 정덕균, "P1394 시리얼 버스 IC의 설계", *대한전자공학회논문지*, 제 35권 (C)편 1호, pp. 34-41, 1998

[3] 황인철, 한상찬, 송병준, 김수원, "3.3V, 400MBPS IEEE 1394 물리층 트랜시버의 설계", *대한전자공학회지*, 제 21권 2호, pp. 783-786, 1998.

[4] IEEE Std. of a High Performance Serial Bus, IEEE Std. 1394-1995, December 1995

[5] Don Anderson, IEEE 1394a Mindshare, Inc., "FireWire System Architecture", Dec 1998.

이 제 훈(Je-hoon Lee)

정회원



1998년 2월 : 충북대학교
정보통신공학과 학사.
2001년 2월 : 충북대학교
정보통신공학과 석사.
2001년 3월 ~ : 충북대학교
정보통신공학과 박사과정.

<주관심 분야> 홈 네트워킹, 통신용 ASIC 설계.

박 광 로(Kwang-roh Park)

정회원



1982년 2월 : 경북대학교
전자공학과 학사.
1985년 2월 : 경북대학교
전자공학과 석사.
1984년 ~ : ETRI 네트워크기술
연구소 홈네트워킹팀
(책임연구원).

1999년 2월 : 충북대학교 정보통신공학과 박사과정
수료

<주관심 분야> 홈 네트워킹, 통신용 ASIC 설계.

서 은 미(Eun-mi Seo)

정회원



1997년 : 충주 산업대학교
정보통신공학과 학사.
1998년 3월 ~ : 충북대학교
정보통신공학과
석사과정.

<주관심 분야> ATM 망, 홈 네트워킹.

조 경 록(Kyoung-rok Cho)

정회원



1977년 : 경북대학교
전자공학과 공학사.
1989년 : 일본 동경대학교
전자공학과 공학석사.
1992년 : 일본 동경대학교
전자공학과 공학박사.

1979년~1986년 : (주)금성사 TV연구소 선임연구원.
1992년~현 : 충북대학교 정보통신공학과 부교수.

<주관심 분야> VLSI 시스템설계, 통신 시스템용 LSI 개발, 고속 마이크로프로세서 설계.