

버퍼공유기법을 사용한 멀티채널 네트워크 컨트롤러 구현

論 文

56-4-22

Implementation of the Multi-Channel Network Controller using Buffer Sharing Mechanism

李 泰 樹* · 朴 宰 賢†
(Tae-Su Lee · Jaehyun Park)

Abstract - This paper presents an implementation of a new type of architecture to improve an overflow problem on the network buffer. Each receiver channel of network system stores the message in its own buffer. If some receiver channel receives many messages, buffer overflow problem may occur for the channel. This paper proposes a network controller that implements a receiver channel with shared-memory to save all of the received messages from the every incoming channels. The proposed architecture is applied to ARINC-429, a real-time control network for commercial avionics system. For verifying performance of the architecture, ARINC-429 controller is designed using a SOPC platform, designed by Verilog and targeted to Xilinx Virtex-4 with a built-in PPC405 core.

Key Words : 통신버퍼 공유, FPGA, SOPC, ARINC-429

1. 서 론

고속으로 동작되는 네트워크의 특징으로 인하여 데이터의 버퍼링은 모든 통신 네트워크에서 사용되는 기본적인 기법이다[2]. 통신의 성능을 향상시키기 위하여 데이터 버퍼를 효율적으로 관리하는 기법에 관한 다양한 연구가 진행되고 있다[3][4][5]. 일반적으로 통신시스템은 수신되는 데이터를 관리하기 위하여 FIFO를 사용하는 구조로 설계되었다. 하나의 네트워크 장치 내에 여러 수신채널이 있는 구조에서 각 채널은 수신된 데이터를 관리하기 위하여 각각의 FIFO를 사용한다. 이러한 구조의 네트워크 장치에서 특정 채널로 데이터의 수신이 집중되는 경우 FIFO의 한계보다 많은 데이터가 수신되어 네트워크의 성능을 저하시키는 문제가 발생할 수 있다. 이런 문제가 발생하는 상황에서 다른 채널의 FIFO를 사용하여 데이터를 처리할 수 있게 되면 전체 통신 시스템의 안정성을 향상시킬 수 있다.

SoC설계 기법 중 NOP(Network-on-Chip)구조를 위하여 FIFO를 사용하는 경우에 발생하는 HOL(Head of Line)문제를 개선하기 위한 방안으로 공유메모리를 사용하는 방법이 제안되었다[7]. 공유메모리를 사용하는 기본 개념을 멀티채널 통신시스템의 데이터 버퍼로 확장하여 각 채널의 FIFO를 하나의 대용량 공유메모리로 대체하는 구조의 네트워크 장치를 제안하여 통신의 성능을 향상시키고자 한다.

SoC를 설계하고 시험하는데 사용되는 디바이스로ASIC(Application Specific Integrated Circuit)를 이용할 수 있는데, 집적도와 성능의 향상으로 칩 내부에 대용량의 메모리를

내장할 수 있게됨에 따라, 새로운 어플리케이션에 응용되는 사례가 많이 있다. 또한 최근에는 PLD (Programmable Logic Device)의 가격 하락과 집적되는 게이트 수의 비약적인 증가로 PLD기반의 SoC도 많이 사용되고 있다..

PLD(Programmable Logic Device)기반의 SoC인 SOPC(System On a Programmable Chip)를 기반으로 시스템을 설계할 경우 성능이 검증된 IP(Intelligent Property)를 사용함으로써 설계시간을 단축할 수 있고 시스템의 수정, 보완이 용이한 장점이 있다. 표준이나 인터페이스 방식이 확정되지 않은 시스템, 또는 미리 알려지지 않은 요구조건을 충족시키기 위한 시스템을 설계하게 될 경우 PLD기반의 플랫폼에서 개발을 한다면 변경된 인터페이스나 요구조건에 맞춰 수정과 보완이 용이하다. 이러한 SOPC의 장점을 통신시스템 설계에 적용할 경우 기존에 검증되었던 IP를 활용하여 시스템의 설계 시간을 단축할 수 있고, 시스템의 신뢰도를 높일 수 있다. 이에 따른 개발 비용 절감효과 또한 얻을 수 있다. 표준이나 인터페이스가 변경되는 경우에도 시스템 전체를 재설계 할 필요 없이 최소의 시간과 비용만 투자하여 PLD의 구조를 변경함으로써 새로운 환경에 적용 가능한 시스템 설계가 가능하다. 이러한 SOPC의 장점을 적용하여 FPGA기반에서 Xilinx사의 소프트 코어 프로세서인 MicroBlaze를 이용한 네트워크 SoC가 제안되었다[1].

본 논문에서는 네트워크를 구성하는 한 장치내의 수신채널이 통신 데이터를 관리하기 위하여 공유메모리를 사용하는 구조를 제안하고 이를 민간 항공기용 데이터버스인 ARINC-429통신에 적용하여 제안한 구조의 성능을 검증하는 내용을 다룬다. 버퍼공유기법의 ARINC-429통신 컨트롤러를 HDL을 이용하여 설계하고 SOPC기반 플랫폼에 적용하여 그 성능과 기능을 검증하였다.

서론에 이어서 2장에서는 버퍼공유기법 네트워크 컨트롤러의 구조와 제안하는 알고리즘에 대하여 설명하고 3장에서는 제안하는 구조를 적용할 ARINC-429통신과 ARINC-429

* 學生會員 : 仁荷大學校 情報通信工學科 碩士

† 교신저자, 終身會員 : 仁荷大學校 情報通信工學科 教授 · 工博

E-mail : jhyun@inha.ac.kr

接受日字 : 2006年 12月 15日

最終完了 : 2007年 12月 25日

통신 컨트롤러의 구조에 대하여 설명한다. 4장에서는 공유 메모리 제어기가 적용된 ARINC-429통신 컨트롤러의 기능을 검증하고, 5장을 통해 결론을 제시하여 논문을 끝맺는다.

2. 버퍼공유기법 네트워크 컨트롤러

2.1 버퍼공유기법 네트워크 컨트롤러 구조

일반적인 네트워크 장치의 경우 각 수신채널은 수신된 데이터를 관리하기 위하여 개별적인 버퍼를 사용한다. 이러한 구조의 수신채널에서 발생할 수 있는 메모리 오버플로우 문제를 개선하여 통신의 안정성을 높이기 위한 방법으로 한 장치내의 수신채널들이 하나의 대용량 수신메모리를 공유하도록 하여 메시지 수신에 집중되는 채널에서 발생할 수 있는 메모리 오버플로우 문제를 보완하고 통신의 안정성을 높이는 구조를 제안한다.

버퍼공유기법의 수신 장치를 내장한 시스템은 그림 1과 같은 구조를 갖는다. 장치내의 여러 수신채널로부터 수신되는 모든 데이터들은 하나의 공유메모리에 저장되어 관리된다. 수신채널은 공유메모리 제어기(SMC: Shared-Memory Controller)에게 수신된 데이터를 전달하고 저장을 요청한다.

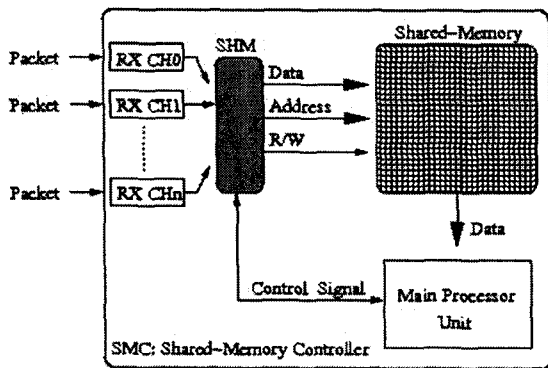


그림 1 버퍼공유기법의 수신장치
Fig. 1 Receiver using sharing buffer

2.2 공유메모리 제어기

공유메모리 제어기는 수신채널의 데이터 저장 요청과 메인 프로세서의 데이터 요청을 처리하며 공유메모리를 제어하는 역할을 한다. 공유메모리 제어기는 각 채널로 수신된 데이터들의 연결을 관리하기 위하여 FAP (First Address Pointer)와 LAP (Last Address Pointer), NAP (Next Address Pointer) 레지스터를 사용한다. 각 수신채널에게 할당된 FAP는 해당 채널로 수신되어 현재 공유메모리에 저장되어있는 데이터 중에서 가장 먼저 저장되었던 데이터의 주소를 나타낸다. LAP는 해당 채널과 관련되어 가장 마지막에 저장되었던 데이터의 주소를 나타낸다. 공유메모리 제어기는 FAP와 LAP를 이용하여 채널 데이터 연결의 처음과 끝을 생성한다. 공유메모리에 저장되어있는 데이터들 사이의 연결을 유지하기 위하여 공유메모리의 주소들과 대응되는 하나의 NAP가 존재한다. NAP는 공유메모리 저장된 데이터의 다음 데이터를 나타내는 주소로 채널 데이터들 간의

연결을 유지시키는 역할을 한다. 공유메모리 제어기는 수신된 데이터를 저장하기 위하여 공유메모리 영역 중에 현재 데이터가 저장되어 있지 않은 영역을 알 수 있어야 한다. 공유메모리 제어기는 MST (Memory Status Table)정보를 사용하여 공유메모리의 상태정보를 관리 한다.

2.3 공유메모리 제어 알고리즘

본 논문에서 제안하는 공유메모리 제어기는 3가지의 동작 상태를 갖는다. 공유메모리에 데이터를 저장하는 동작 상태와 공유메모리에 있는 데이터를 로드하는 동작 상태 그리고 공유메모리에 데이터를 저장하기 위하여 사용가능한 메모리 영역을 검색하는 동작 상태로 나뉜다. 3가지 동작 상태를 설명하는 그림 2, 그림 3, 그림 4는 3장에서 설명할 4개의 수신채널로부터 수신되는 데이터를 관리하기 위한 공유메모리 제어기의 제어 알고리즘을 나타낸 것이다.

그림 2는 공유메모리의 영역 중에서 데이터가 저장되어 있지 않은 영역을 찾는 과정을 보여준다. 공유메모리 제어기는 메모리에 데이터를 저장하거나 메모리로부터 데이터를 로드하지 않는 동안 MST를 이용하여 메모리의 상태를 확인한다. CAP(Current Address Pointer)는 MST와 대응되는 공유메모리의 주소를 나타낸다. 공유메모리 제어기는 CAP가 현재 사용되지 않는 메모리 영역을 가리킬 때 까지 MST를 검색하는 동작을 반복하면서 사용가능한 메모리 영역을 찾게 된다.

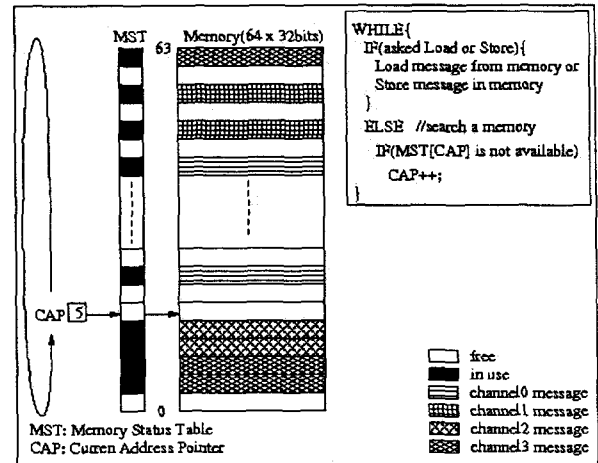


그림 2 공유메모리 탐색
Fig. 2 Searching shared memory

그림 3은 수신된 데이터를 공유메모리에 저장하는 과정을 보여준다. 공유메모리 제어기는 데이터의 저장을 요청한 채널의 LAP와 LAP가 가리키는 주소의 NAP를 수신된 메시지를 저장할 주소인 CAP로 설정하여 마지막으로 저장되었던 데이터와 현재 저장하려는 데이터의 연결을 생성한다. CAP가 가리키는 주소에 수신된 데이터를 저장하고 MST를 재설정 한다. 공유메모리에 해당 채널과 관련된 데이터가 없는 경우 FAP를 CAP로 설정한다.

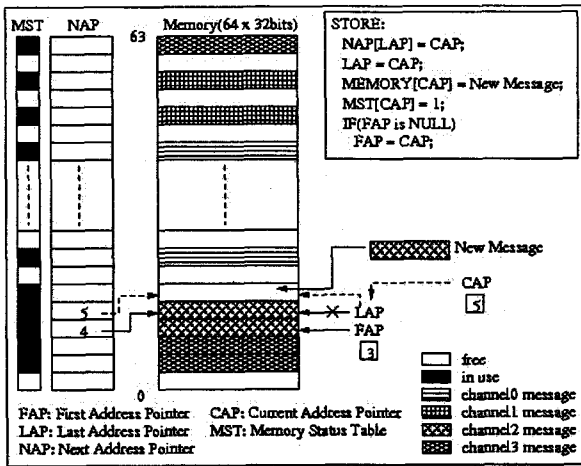


그림 3 데이터 저장 순서
 Fig 3 Data storing sequence

그림 4는 공유메모리로부터 데이터를 로드하는 과정을 보여준다. 저장된 데이터의 로드 요청이 발생하면 공유메모리 제어기는 해당 채널의 FAP가 가리키는 공유메모리 주소의 데이터를 로드하고 MST를 재설정 한다. FAP와 LAP가 같은 주소를 가리키게 되면 현재 로드한 데이터가 해당 채널의 마지막 데이터임을 의미하고 FAP와 LAP가 서로 다른 주소를 가리키는 경우 데이터들의 연결을 유지하기 위하여 FAP를 FAP의 NAP로 재설정한다.

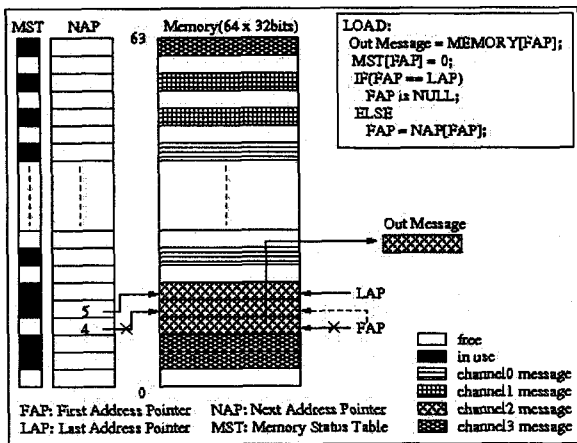


그림 4 데이터 로드 순서
 Fig. 4 Data loading sequence

3. ARINC-429통신 컨트롤러

본 논문에서 제안한 공유메모리를 사용하는 통신 컨트롤러의 성능을 검증하기 위하여 항공기내 제어 네트워크 프로토콜인 ARINC-429통신을 위한 컨트롤러에 제안한 구조를 적용시켜 HDL로 ARINC-429통신 컨트롤러를 설계한다.

3.1 ARINC-429통신

민간 항공기의 데이터 버스로 널리 알려진 ARINC-429통신은 항공기내 장치와 시스템을 제어하기 위한 목적으로 사용되는 네트워크 프로토콜이다. ARINC-429통신 버스는 공식적으로 Mark 33 DITS (Digital Information Transfer System)규정으로 알려져 있다[8]. 트위스트 페어 케이블 (Twisted Shielded Pair)을 통해 서로 연결되어 있는 항공기내 장치와 시스템들은 ARINC-429통신 규정에 정의된 특성과 프로토콜에 따라 메시지를 전송한다. ARINC-429통신 데이터 버스는 1980년대에 설계된 민간 항공기인 보잉 757, 767과 에어버스 A300, A310 및 A320 등에 사용되고 있다. ARINC-429통신은 단방향 데이터 전송 방식으로 규정되어 있다. 장치간의 개별 통신선이 연결되어 있으므로 버스의 데이터 흐름을 제어하는 버스 제어기가 필요하지 않다. 한 버스 내에 최소 하나의 수신 장치가 있어야 하며 최대 20개의 수신 장치가 연결되어 네트워크를 구성할 수 있다. 표 1은 ARINC-429통신의 특징을 보여준다[9].

표 1 ARINC-429 특징
 Table 1 Characteristics of ARINC-429

ARINC-429 특징	
전송 매체	트위스트 페어 케이블
라인 임피던스	75+5[Ohms]
신호 방식	BRZ
전송 방향	단방향
전송 방식	비동기 브로드캐스트
전송 순서	LSB
전송 속도	12-14.5Khz, 100Khz
오류 검출방식	오드패리티
워드 크기	32비트
데이터 크기	19비트
전송메시지 간격	최소 4비트 시간

ARINC-429통신은 단방향 방식을 통신버스가기 때문에 장치간의 양방향 통신을 위해서는 전송과 수신에 각각 다른 버스 선을 필요로 한다. 전송 채널은 전송선을 통해 데이터를 전송할 수 있고, 수신채널은 수신 선을 통해 데이터를 수신할 수 있다.

그림 5는 양방향 데이터 통신을 위한 ARINC-429통신 네트워크의 구조를 보여준다. ARINC-429통신의 메시지는 32비트 구성되어있다. 최하위비트인 LSB의 비트 번호를 1, 최상위비트인 MSB의 비트번호를 32로 규정한다. 32비트의 데이터를 역할에 따라 Label, SDI, DATA, SSM, P 총 5개의 필드로 구분한다[9]. 그림 6은 ARINC-429의 메시지 형식을 나타낸다.

Label필드는 8비트로 구성되어있고 32비트의 ARINC-429 데이터 중 비트 1-8에 할당 되어 3자리의 8진수로 표기된다. Label 필드는 수신된 메시지의 데이터 유형을 나타내고 있기 때문에 메시지를 해석하는데 사용되어 수신장치에서 메시지의 수신여부를 판단하는 목적으로 사용된다. Label필드는 비트번호 1이 MSB이고 비트번호 8이 LSB이다.

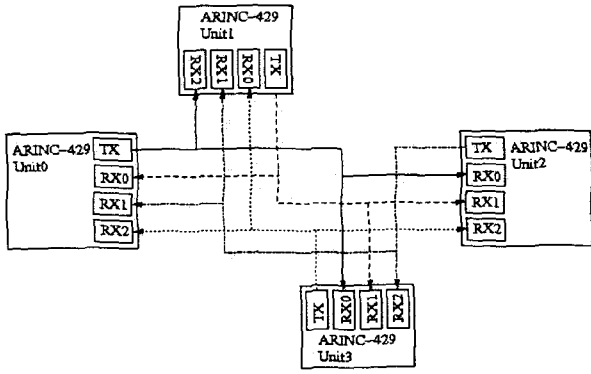


그림 5 ARINC-429통신 네트워크 구조
Fig. 5 Structure of ARINC-429 network

P	SSM	MSB	DATA-19bits														LSB	SDI	8bit Octal Label																																																																												
32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1																																																																
MSB																LSB		LSB		MSB																																																																											
MSB																																32-bit ARINC-429 Word																																LSB																															

그림 6 ARINC-429 메시지 형식
Fig. 6 Message format of ARINC-429

SDI(Source/Destination Identifier)필드는 2비트로 구성되어 있고 선택적으로 사용 가능하다. 32비트의 데이터 중에 9번과 10번 비트에 해당되며 사용될 경우 멀티시스템에서 전송장치와 수신장치를 나타내기 위해 사용될 수 있다. 경우에 따라서는 데이터 비트로 사용되기도 한다.

SSM(Sign/Status Matrix)필드는 30, 31번 비트에 할당되어 상태정보로써 기능을 할 경우에는 하드웨어 장치의 상태 또는 동작 모드, 데이터의 유용성 등을 나타내고 부호로서 기능을 할 경우에는 방향, 부호 등을 나타낸다.

P(Parity)비트는 ARINC-429메시지 중 32번째 비트에 할당되어 메시지의 에러 유무를 판단하는 목적으로 사용된다. 일반적으로 Odd parity로 설정된다.

Data필드는 32비트 중 11번에서 29번 비트에 할당된 값으로 총 19비트가 데이터를 나타내는 목적으로 사용된다. SDI필드가 본래의 기능으로 사용되지 않을 경우 데이터 비트는 31번 비트까지 확장되어 사용될 수 있다.

데이터의 전송은 Label필드의 최하위부터 패리티 비트까지 순차적으로 전송되며 다음과 같은 비트 순서를 갖는다 [9].

8,7,6,5,4,3,2,1,9,10,11,12,13, ... 32.

3.2 ARINC-429통신 컨트롤러 구현

멀티채널 네트워크 컨트롤러에서 수신메모리를 위한 버퍼 공유기법이 우수함을 증명하기 위하여 ARINC-429통신 컨트롤러에 버퍼공유기법을 적용하여 HDL로 설계하였다. 그림 7은 설계한 네트워크 컨트롤러의 수신부 블록다이어그램이다. 4개의 수신채널은 각각 수신되는 데이터를 하나의 메시지로 만들어 관리하기 위한 De-serializer블록과 공유메모리와 수신채널을 관리하기 위한 제어블록으로 구성된다.

설계한 ARINC-429 IP는 1Mhz의 동작주파수를 갖는다. 내부에 수신채널과 전송채널의 상태-제어 레지스터 액세스

하기 위해 9비트의 어드레스와 32비트의 데이터 버스를 갖는다. 2개의 전송채널과 4개의 수신채널을 포함하며 각 채널이 독립적인 전송완료 인터럽트와 수신완료 인터럽트의 소스가 된다. 공유메모리에 저장된 데이터가 공유메모리의 설정된 경계 영역을 넘게 되면 메모리 쓰레스홀드 인터럽트가 발생된다. 전송채널과 수신채널은 각각 12.5Kbps와 100Kbps의 통신 속도를 지원한다.

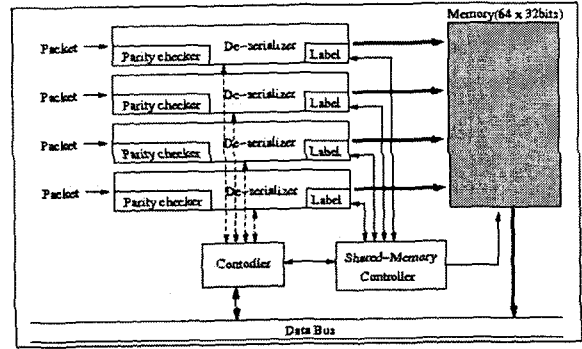


그림 7 ARINC-429 수신채널 구조
Fig. 7 Structure of ARINC-429 receiving channel

ARINC-429 IP의 설계와 synthesis를 위하여 Xilinx사의 ISE8.103i를 사용하였고 ModelSim SE PLUS 5.8d를 사용하여 시뮬레이션 하였다. Xilinx사의 하드코어 프로세서인 PPC405를 내장하고 있는 Virtex-4 FX12[10]에 적용시킨 ARINC-429통신 컨트롤러는 2,103(38%)개의 Slices와 1개의 Block RAM(2.8%)을 사용한다. 이를 등가 게이트로 환산하면 총 42,797개의 게이트가 ARINC-429통신 컨트롤러를 구현하는데 사용되었음을 확인하였다.

4. 성능 평가

본 논문을 통해 설계한 공유메모리 방식의 ARINC-429통신 컨트롤러의 성능을 검증하기 위해서 Xilinx의Virtex-4 FX12를 이용하여 제작된 Memec사의 Virtex-4 FX12 Mini-Module보드[11]를 사용하였다.[그림 8]

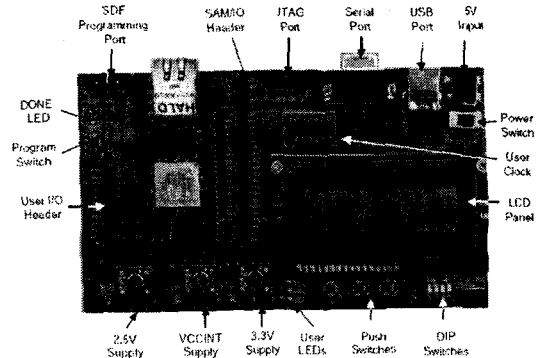


그림 8 Virtex-4 FX12 mini-module
Fig. 8 Virtex-4 FX12 mini-module

검증을 위해 구현한 시스템은 PPC405코어와 프로그램과 데이터를 위한 메모리 영역, 사용자 인터페이스를 위한 UART와 GPIO 그리고 인터럽트 제어하기 위한 INTC, 본 논문을 통해 설계한 ARINC-429 IP로 구성되어 있다. 전송/수신 완료 인터럽트와 메모리 쓰레스홀드 인터럽트는 INTC에 연결되어 상승에지 인터럽트로 작동 된다. PPC405와 사용자 IP는 OPB(Onchip Peripheral Bus)를 이용하여 통신할 수 있다[12]. Xilinx사의 EDK8.1.02i를 사용하여 PPC405기반의 ARINC-429통신 컨트롤러를 구성하였다. 그림 9는 설계된 ARINC-429통신 시스템의 블록 다이어그램을 나타낸다.

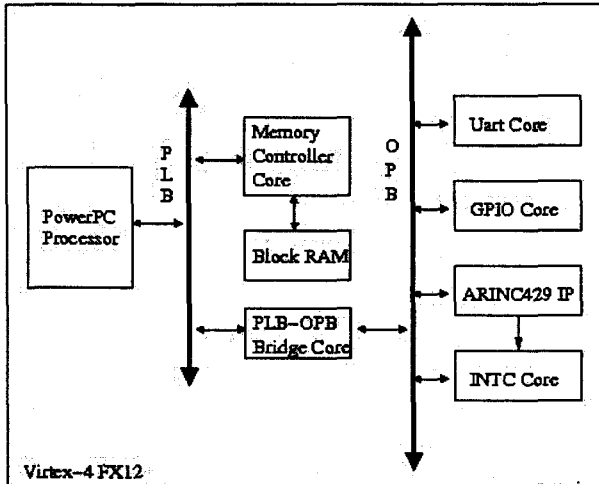


그림 9 ARINC-429 통신 시스템
Fig. 9 ARINC-429 network system

데이터 버퍼로 각 수신채널이 16개의 ARINC-429통신 메시지를 저장할 수 있는 FIFO를 사용하는 시스템의 경우 수신된 메시지가 처리 되지 않을 경우 최대 16개의 메시지가 FIFO에 저장할 수 있다. 이후에 수신되는 메시지에 대하여 다른 수신채널의 FIFO가 비워있더라도 그 영역을 사용할 수 없는 구조이기 때문에 FIFO에서 데이터 오버플로우에러가 발생하게 된다. 반면 4개의 수신채널이 사용하는 FIFO를 64개의 메시지를 저장할 수 있는 하나의 공유메모리로 대체한 ARINC-429통신 컨트롤러는 다른 채널로부터 수신되는 메시지가 없는 경우 하나의 수신채널로 최대 64개의 메시지를 저장할 수 있었다.

동작을 검증하기 위하여 4개의 수신채널의 입력을 전송채널0과 내부적으로 연결하는 테스트 모드로 설정하고 통신 속도를 100Kbps로 맞추어 테스트 하였다. 4개의 수신채널의 수신완료 인터럽트와 전송채널0의 전송완료 인터럽트를 활성화 시킨 후 총 60개의 메시지를 전송하였다. 각 수신채널이 수신완료 인터럽트를 통해서 각각 60개의 메시지를 처리하는 것을 확인하였다.

다음으로 수신채널 2와 3의 수신완료 인터럽트를 비활성화한 후 60개의 메시지를 전송하였다. 수신채널 0과 1을 통해서 수신되어 메모리에 저장된 메시지는 수신완료 인터럽트를 통해서 처리되었지만 수신채널 2와 3의 경우는 메모리에 저장되어 남아있는 상태가 된다. 이러한 이유로 64 x 32 비트의 수신메모리가 모두 사용 중인 상태가 되는 시점에서

수신된 메시지가 저장되지 못하여 수신완료 인터럽트가 더 이상 발생되지 않는 것을 확인하였다.

5. 결 론

본 논문을 통해서 멀티채널 통신시스템에서 데이터를 관리하기 위한 메모리로 공유메모리를 사용하는 구조를 제안하고 이를 ARINC-429통신에 적용하여 버퍼공유기법을 적용한 ARINC-429통신 컨트롤러를 구현하였다.

기존의 각 채널이 FIFO를 이용하여 수신메시지를 저장하는 방식의 수신채널의 경우 수신된 메시지를 처리하기 이전에 각 채널로 수신된 메시지를 저장할 수 있는 공간보다 많은 메시지가 수신되는 경우 다른 채널의 FIFO영역이 사용 가능 하더라도 해당채널의 FIFO에서 오버플로우 문제가 발생하여 전체 통신의 성능이 저하되는 문제점이 발생할 수 있다. 하지만 본 논문에서 제안한 수신 메모리를 공유하는 방식은 모든 채널이 하나의 대용량 수신 메모리를 공유하며 사용함으로써 통신 성능의 안정성을 향상시켰다.

HDL로 구현된 ARINC-429통신 컨트롤러를 통하여 데이터 버퍼로 공유메모리를 사용하는 통신 컨트롤러가 각 통신 채널이 각각의 FIFO를 사용하는 구조의 통신 컨트롤러보다 메모리를 효율적으로 사용하는 것을 확인하였다. 공유메모리 방식의 통신 컨트롤러는 ARINC-429통신 뿐 아니라 다른 통신 컨트롤러의 데이터 버퍼 관리 기법으로 적용되어 통신의 성능을 향상시킬 수 있다.

참 고 문 헌

- [1] Peter Magnusson, "Evaluating Xilinx MicroBlaz for Network SoC Applications", Master's Thesis in Computer Engineering, pp. 30-31, 2004
- [2] S. Andrew, Tanenbaum, Computer Networks 4nd, Prentice Hall PTR, pp 506-510, 2003
- [3] M. Mobasseri, V.P.M Leung, "A new buffer management scheme for multimedia terminals in broadband satellite networks", Proceedings of the 35th Hawaii International Conference on System Sciences, Volume 5, pp, 2786-2790, 2002
- [4] M. Alisafae, S.M Fakhraie, M. Tehranipoor, "Architecture of an embedded queue management engin for high-speed network devices", 48th Midwest Symposium on Circuits and Systems, Vol.2, pp. 1907-1910, 2005
- [5] Y.-M Joo, N. McKeown, "Doubling memory bandwidth for network buffers", INFOCOM '98. Conference of the IEEE Computer and Communications Societies, Vol 2, pp.808-815, 1998
- [6] S.M Qasim, S.A Abbasi, "FPGA implementation of a single-channel HDLC Layer-2 protocol transmitter using VHDL" Proceedings of the 15th International Conference on Microelectronics, pp. 265-268, 2003
- [7] Po-Tsang Huang, Wei Hwang, "2-level FIFO

- architecture design for switch fabrics in network-on-chip", in ISCAS'06, pp. 4863-4866, 2006
- [9] ARINC Specification 429 PART 1-17, ARINC Inc., pp. 1-3, 2004
 - [8] Len Buckwalter, AVIONICS DATABASES 2nd, Avionics Communications Inc., pp. 10.4-5, 2003
 - [10] Virtex-4 User Guide, Xilinx Inc., 2006
 - [11] OPB IPIF(v3.01c), Xilinx Inc., 2005
 - [12] Virtex-4 FX12 Mini Module User Guide, Memec Inc., 2005

저 자 소 개



이 태 수 (李 泰 樹)

1982년 10월 14일생. 2005년 인하대 전자과 졸업. 2007년 대학원 정보통신공학과 석사

Tel : 032-863-0442

E-mail : leetaesu@emcl.inha.ac.kr



박 재 현 (朴 宰 賢)

1963년 10월 8일생. 1982년 서울대학교 제어계측과 졸업. 1994년 동 대학원 제어계측공학과 졸업(공학). 1995년~현재 인하대학교 정보통신공학부 교수. 연구분야: 임베디드시스템, 실시간시스템, 컴퓨터네트워크

Tel : 032-860-7713

E-mail : jhyun@inha.ac.kr