

리눅스를 포팅한 라우터 설계 및 구현

*양진철, *김성현, *김종수, *문종욱, *정기현, **최경희
*아주대 전자공학부, **아주대 정보및컴퓨터공학부
전화 : (031) 212 - 7627 / 팩스 : (031) 212 - 7620

Design and Implementation of a Router System ported Linux

*Jin-Chul Yang, *Sung-Hyun Kim, Jong-Wook Moon, Jong-Su Kim
*Kihyun Chung, *Kyunghee Choi
*Dept. of Electronics Engineering, Ajou University
* Dept. of Information and Computer Engineering, Ajou University
E-mail : yjinchul@hanmail.net

Abstract

This paper addresses design and implementation of a router based on Linux. Around MPC860T CPU which includes a common communication controller as the main processor, the hardware has two WAN ports and one Ethernet port. Linux was optimized and ported into the hardware, and used as the operating system.

I. 서론

컴퓨터 통신의 발달과 더불어 통신망이 고속화됨에 따라 인터넷이 급속도로 발전하게 되었다. 이런 네트워크와 컴퓨팅 환경의 발전으로 인하여 인터넷을 사용하는 통신서비스의 이용이 급격히 증가됨에 따라 LAN과 WAN상에서의 많은 트래픽 현상을 일으키게 되었다. 이로 인해 사용자의 요구를 수용하기 위해 네트워크 상에서 메시지를 최종 목적지까지 전달하기 위한 전송 경로에 대해 매우 안정적이고 효율적인 관리가 점점 필요로 되어지고 있다. 이를 해결하고 네트워크를 구축해 주는 장비가 라우터이다. 라우터는 서로 상이한 구조를 갖는 망들을 연결할 수 있는 기능을 제공하며 부분 망 내에서 주고받는 데이

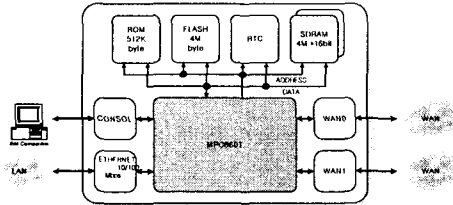
터는 부분 망 내에서만 움직이도록 제한을 가하여 네트워크에 발생 될 수 있는 불필요 작업량을 제거하여 준다. 라우터는 OSI 계층 구조의 물리층, 링크 계층 그리고 네트워크 계층상에서 동작한다. 네트워크 계층의 주요기능 중의 하나가 바로 패킷의 전송 경로를 설정하고 해제하는 라우팅 기능이다. 라우터는 패킷 내의 목적지 주소를 읽을 수 있으므로 라우터 내의 Routing Table 을 이용하여 데이터 패킷이 지나가야 할 경로를 지정한다. 라우터는 OSI 하위 3계층을 포함하여 동작하므로 하위의 3계층이 서로 다른 망들을 연결하는데 사용할 수 있다. 그러나 이러한 경우에도 종단 DTE간의 트랜스포트 이상 상위 계층은 동일한 프로토콜이 동작하여야 한다. 또한 라우터는 망 내의 혼잡 상태를 제어할 수 있는 기능을 갖고 있으며, 따라서 어떤 경로에 많은 트래픽이 있는 경우에는 패킷을 다른 경로로 전송하는 등의 기능도 갖는다.

본 연구의 목적은 많은 트래픽 현상에도 안정적이고 효율적인 Linux 기반 Router의 관련된 기술들을 연구 및 개발하고 이를 독자적인 Standalone의 하드웨어에 이식하여 Slim형 Router를 개발하는데 있다. (이하 개발된 라우터를 슬림 라우터라 부른다.)

본 논문의 구성은 다음과 같다. 2장에서는 현재 개발한 슬림 라우터의 하드웨어가 어떻게 구성되고 설계되어 있는지를 살펴보고, 3장에서는 슬림 라우터의 리눅스 포팅 대해 소개하고, 마지막으로 4장에서는 결론을 맺는다.

II. 슬립 라우터의 설계 및 구현

1 슬립형 라우터의 하드웨어의 구성



[그림 1] 슬립형 라우터의 구성도

위 그림에서 보는 것과 같이 라우터는 CPU 부분과 메모리부분 그리고 통신 부분으로 나뉜다.

1.1 CPU

CPU는 모토롤라사의 MPC860T를 사용하는 데 이것은 범용 Communication Processor로서, Embedded PowerPC core와 SIU(system interface unit), CPM(Communication Processor Module), Fast Ethernet Controller로 구성되어 있는데, Embedded PowerPC Core는 32 bit PowerPC 603 프로세서이고, SIU는 기본적으로 Memory controller를 내장하고 있고 내부 외부의 인터페이스를 담당하므로 라우터에 필요한 Flash Memory, ROM, SDRAM의 인터페이스를 제공한다. CPM은 독립적인 통신 장치를 가지고 데이터를 송수신하는 기능을 하는데, 독립적인 통신 장치로는 WAN port를 위해 SCC(Serial Communication Channel)을 사용하고 콘솔용 UART를 위해서 SMC(Serial Management Channel)을 사용한다. CPM 내부에는 다양한 통신 프로토콜 스택을 지원하기 위해 별도의 RISC가 있다. 통신 처리에 있어서 PowerPC가 상위 layer의 기능을 최대화하기 위해 사용한다면 RISC는 수신된 데이터를 메모리로 옮긴다든지 실제 통신을 제어하는 하위 layer 기능을 처리할 수 있다. 마지막으로 Fast Ethernet Controller는 10/100 Mbps Ethernet용 통신모듈로 방대한 양의 데이터를 처리하기 위해서 DMA controller가 포함되어 있다.

1.2 메모리

슬립 라우터의 메모리는 ROM, SDRAM, Flash Memory가 쓰인다. 이 디바이스들은 MPC860T의 SIU unit의 memory controller에 의해서 인터페이스가 지원된다.

Address Range	Memory Type	Size	Extend	Internal
00000000 ~ 00FFFFFF	SDRAM	16M byte	64M byte	CS1
01000000 ~ 03FFFFFF	EMPTY			
04000000 ~ 043FFFFFF	FLASH	4M byte	16M byte	CS2
04400000 ~ 04FFFFFF	EMPTY			
05000000 ~ E5FFFFFF	EMPTY			
E6000000 ~ E6007FFF	RTC	2K byte		CS3
E6008000 ~ EFFFFFFF	EMPTY			
F0000000 ~ F007FFFF	ROM	4M byte		CS0
F0080000 ~ FEFFFFFF	EMPTY			
FF000000 ~ FF07FFFF	DPRAM	16K byte		
FF080000 ~ FFFFFFFF	EMPTY			

[표 1] 슬립형 라우터의 메모리 맵

ROM은 512Kbyte의 메모리 용량을 가지고 CPU와의 인터페이스를 위해서 CS0를 사용한다. 여기에는 하드웨어 초기화 프로그램, 모니터 프로그램, IL(Image Loader)가 들어 있다.

SDRAM은 16Mbyte의 메모리 용량을 가지고 CPU와의 인터페이스를 위해서 CS1을 사용한다. 이것은 시스템 메모리로 실제 리눅스가 움직이는 공간으로 부팅 시 압축된 커널 이미지가 SDRAM의 특정한 부분에 압축이 풀리고 다시 풀린 커널을 재배치 시키고 부팅이 시작되는 장소이다. 그리고 압축된 루트 파일 시스템 이미지가 풀리는 곳이다.

Flash Memory는 4Mbyte의 메모리 용량을 가지며 CPU와의 인터페이스를 위해 CS2를 사용한다. 리눅스가 부팅이 되려면 LKI(Linux Kernel Image)와 RFSI(Root File System Image)가 필요하다. 그런데 슬립 라우터는 임베디드 시스템으로 일반 PC처럼 플로피 디스크나 하드디스크 같은 저장 공간이 없다. 그래서 이러한 저장공간을 대신하여 Flash Memory가 쓰여진다.

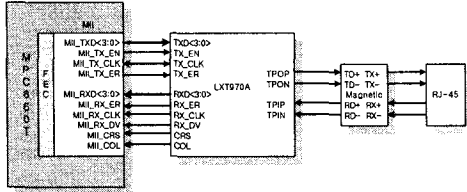
1.3 통신

슬립 라우터의 통신 부분은 다시 Console, LAN, WAN 세 부분으로 나뉘어진다.

2.1.3.1 LAN 인터페이스

MPC860T는 FEC(Fast Ethernet Controller)를 지원하는데, 이것은 MII(Media Independent Interface), MAC(Media Access Control), FIFO, DMA로 구성되어 있다. MAC은 MII를 완전히 제공하고, DMA는 외부 메모리로 혹은 외부 메모리로부터 데이터 전송 시 burst 전송을 하는데 사용된다. 100Mbps 통신을 하다 보면 다양한 버스트

latency가 생기는데 이를 보상하기 위해 FIFO를 사용한다.

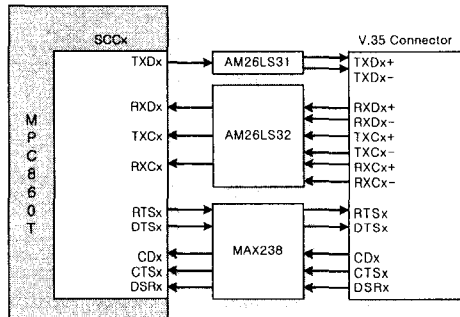


[그림 2] MII data 인터페이스

위 그림은 MPC860T와 Fast Ethernet Transceiver의 인터페이스를 보인 것으로, Fast Ethernet Transceiver는 PSC(Physical Coding Sublayer), PMA(Physical Medium Attachment), PMD(Physical Medium Dependent) 세 개의 부 계층으로 나뉘어진다. PSC는 MII interface와 접속되는 부분으로 Encoding과 Decoding을 담당하고, PMA는 링크와 캐리어를 감지하는 일을 담당하고, PMD는 시그널의 드라이빙을 담당한다.

1.3.2 WAN

WAN은 V.35 인터페이스를 사용한다. 슬립 라우터 시스템에서는 MPC680T의 SCC1과 SCC3에 연결하여 WAN port 2개를 이용하고 있으며, 이는 외부적으로 CSU/DSU 모뎀에 연결된다.



[그림 3] MPC860과 V.35 Conn과의 인터페이스

Data와 Clock은 Balanced(differential) 신호로서 5V의 신호를 +12V와 -12V로 differential하게 전환해 주기 위해 전송할 때 AM26LS31과 수신할 때 AM26LS32 드라이버가 사용된다. Control 신호는 Unbalanced 신호로 5V 신호를 12V로 전환하기 MAX238이 사용된다. 이 밖에 신호의 과전압 방지를 위해서 IC03-6의 data interface가 이용된다.

1.3.3 Console

Console은 MPC860T의 SMC0를 사용하는데

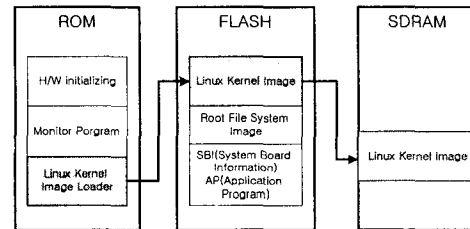
이것은 드라이버로 MAX 232C에 접속되어 PC와 연결되어진다. SMC0는 UART 프로토콜을 이용하여 라우터를 설정하고, 라우터에 대한 정보를 얻을 수 있는 통로가 된다.

III. 슬립 라우터의 리눅스 포팅

1. 라우터의 하드웨어 초기화 과정

시스템 시작(Power-On or Reset) 후 하드웨어 초기화가 이루어지는데 기본적인 초기화 프로그램은 ROM에 위치하게 된다. 초기화 프로그램은 처음에 PowerPC core Register들을 설정하고, 명령 및 데이터 캐쉬를 초기화하고, Memory Controller를 초기화함으로써 각 드라이브들(ROM, Flash Memory, SDRAM, RTC)의 메모리 영역 및 데이터 버스 크기등을 설정하고, SDRAM이 구동되기 위해 MPC860T의 UPM을 사용하는데 이 곳의 RAM Array에 SDRAM 패턴 및 Command를 저장하여 구동 시킨다. 그 다음 Baud Rate Generator를 설정하고 SMC1를 UART용으로 설정하여 모니터링을 위한 console용 포트를 열게 된다.

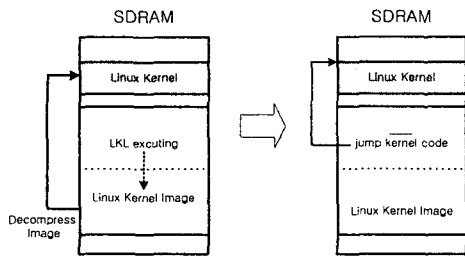
2. 리눅스의 부팅과정



[그림 4] IL(Image Loader)의 동작

ROM에서 부팅하는 방법으로 두 개의 분리된 로더를 지원하는 2단계로 쪼개는 것이다. 하나는 이미지 로더(IL)이고 다른 하나는 리눅스 터널 로더(LKL)로써 이미지 로더는 ROM에 위치한다. 시스템 시작(Power-On or Reset) 후 IL가 실행된 후 필수적인 하드웨어 초기화가 이루어지고 나서 Flash Memory에서 SDRAM의 적절한 위치로 리눅스 커널을 옮긴다. LKL은 IL이 끝나면 실행을 시작하여 LKL은 더 많은 하드웨어를 초기화한다. 리눅스 커널 이미지의 압축을 풀어서 리눅스 커널 부팅을 위한 환경을 셋업한다. 마지막으로 LKL은 리눅스 개시 시퀀스를 시작하기 위

해 커널 코드로 점프한다.



[그림 5] LKL(Linux Kernel Loader)의 동작

3. 디바이스 드라이버

이제 커널의 부팅이 시작되면서 디바이스 드라이버들이 등록되어 질 것이다.

리눅스에는 문자, 블록, 네트워크 세 종류의 디바이스 드라이버가 존재한다. ROM은 리눅스의 부팅에만 관여하는 디바이스로 부팅 후에는 쓰이지 않는다. Flash Memory는 임의의 접근이 가능한 블록 장치로 sf0, sf1, sf2로 나뉘어져 마운트되어 사용되어진다. SDRAM 또한 특정한 부분을 RAM disk로 사용하는 블록 장치로 이용된다. UART 프로토콜을 사용하는 콘솔용 SMC와 HDLC 프로토콜을 사용하는 WAN용 SCC는 항상 순차적으로만 처리될 수 있는 문자 디바이스로 사용되고, /dev/console 및 /dev/tty 장치파일을 접근함으로써 파일을 읽고 쓰듯이 하드웨어를 다룰 수 있다. FEC(Fast Ethernet Controller)는 네트워크 장치 드라이버로 사용된다. 이것은 리눅스 상에 FEC라는 이름이 붙여져 리눅스에 등록된다.

IV 결론

본 논문에서는 범용 통신 컨트롤러를 내장한 MPC860T를 사용하여 라우터 기능에 필요한 LAN port 및 WAN port를 가진 하드웨어 설계, 이 하드웨어에 리눅스 OS를 포팅하는 방법을 제시하여 리눅스를 탑재한 라우터를 설계 및 구현에 대해 언급하였다.

현재의 라우터 시스템을 바탕으로 네트워크 상의 컴퓨터, 통신기기, 단말기기, 통신회선 등의 물리적 오브젝트나 통신을 하고 있는 세션, 액세스, 메시지, 데이터 등의 논리적 오브젝트를 관리하여 사용자가 통합적으로 또한 효율적으로 워크를 이용할수있도록 SNMP(Simple Network

Mangement Program), RMON(Remote network MONitering) , MIB(Mangement Information Remote)같은 네트워크 관리를 위한 연구가 필요하며, 또한 IDS, IP firewall 같은 네트워크 상에서의 보안에 관한 연구가 필수적으로 이루어져야 할것이다.

참고 문헌

- [1] Motorola Inc, "MPC860 PowerQUICC User's Manual", 1998
- [2] Motorola Inc, "MPC860T Fast Ethernet Controller", 1998
- [3] Level One Inc, " LXT970 Dual-Speed Fast Ethernet Transceiver", 1999
- [4] Motorola Inc "MPC8xx SDRAM Interface", 1998
- [5] Michael Beck, Harald Bohme " Linux Kernel Internals" 2nd ED, Addison-Wesley,1998
- [6] Alessandro Rubini, " Linux Device Drivers", O'REILLY,1998
- [7] Remy Card, Eric Dumas, "the Linux Kernel Book", Wiley, 1998
- [8] Andrew Sun, "Using & Manasing PPP", O'REILLY, 1999