PCI 기반 LEON2-FT 프로세서를 위한 시스템 소프트웨어 설계 및 시뮬레이션

최종욱*, 정회원, 남병규**

System Software Design and Simulation for LEON2-FT Processor based on PCI

Jong-Wook Choi*, Byeong-Gyu Nam**

요 약

인공위성의 임무가 다양해지고 요구사항이 높아짐에 따라 탑재컴퓨터의 성능 향상이 필수적으로 대두되었으며, 인공위성 탑재컴퓨터의 활용도를 높이기 위해 표준화 설계 및 시스템 재구성이 가능한 모듈화 기반으로 개발 되고 있다. 현재 한국항공우주연구원에서 개발 중인 차세대 인공위성의 탑재컴퓨터 경우 높은 성능을 제공하기 위해 SPARC v8 기반의 LEON2-FT/ AT697F 프로세서를 채택하였으며 SpaceWire, MIL-STD-1553B, CAN 등의 다양한 통신 디바이스들을 표준화 된 통신칩으로 구성하여 프로세서에서 PCI 통신을 통해 각종 디바이스들을 제어 및 통신 할 수 있도록 개발 하고 있다. 본 논문에서 차세대 탑재컴퓨터의 LEON2-FT 프로세서와 PCI 기반에서의 시스템 소프트웨어 개발 방안에 대해서 기술하며, PCI 소프트웨어 컴포넌트 설계 및 실시간 운영체제인 VxWorks 6.5 포팅 그리고 개발 된 시스템 소프트웨어를 검증하기 위한 시뮬레이션 방안을 제시한다.

Key Words : LEON2, AT697F, Processor, PCI, System Software, Simulation

ABSTRACT

The need for high performance of on-board computer (OBC) is essential due to the growing requirements and diversified missions, and so OBC has been developed on the basis of the standard design and reconfigurable modularization in order to improve the utilization of OBC for different missions. The processor in OBC of next generation satellite which is currently developed by KARI is adopted the LEON2-FT/AT697F processor based SPARC v8 as main processor and controls various devices such as SpaceWire, MIL-STD-1553B and CAN through PCI on the standardized communication chips. This paper presents the architecture and design of system software for LEON2-FT processor based on PCI, and development of PCI software component. Also it describes the porting of VxWorks 6.5 for LEON2-FT and the test under the simulation environment for LEON2-FT and PCI with communication chips.

I. 서 론

인공위성의 임무가 다양해지고 요구사항이 높아짐에 따라 탑재컴퓨터의 성능 향상이 필수적으로 대두되었으며, 인공위성의 다양한 활용을 위한 표준화 설계 그리고 위성 기능 및 임무에 따라 시스템 재구성이 가능한 모듈화 설계의 필요성이 계속 재기되었다. 한국항공우주연구원 (항우연)에서 개발 된 기존 저궤도 위성에서는 SPARC v7 기반의 MCMERC32 프로세서를 사용하였으며, MCMERC32 프로세서에서 제공되는 MIL-STD-1553B, UART 등의 인터페이스를 통해 외부 유닛간의 통신을 수행하였으며, 탑재컴퓨터 내부 통신을 위해 추가적으로 SpaceWire (SpW)를 활용하였다. 차세대 인공위성의 탑재컴퓨터 (NGSCB, Next Generation Standard CPU Board)의 경우 높은 성능을 만족하기 위해 SPARC v8 기반의 LEON2-FT/AT697F[1] 프로세서를 선택하였으며, 다양한 임무와 모듈화를 위해 SpW, 1553B 등의 다양한 통신 인터페이스를 별도의 통신칩으로 구성하여 프로세서와 PCI (Peripheral Component Interconnect) 버스를 통해 제어 할 수 있도록 개발 하고 있다. NASA 및 ESA 경우 각 컴포넌트간의 확장을 위하여 표준 버스로 PCI를 사용해온 반면 기존 항우연에서 개발 된 탑재컴퓨터의 경우 내부 버스나 SpW를 사용했기 때문에 탑재

*한국항공우주연구원 위성비행소프트웨어팀 jwchoi@kari.re.kr, **충남대학교 컴퓨터공학과 bgnam@cnu.ac.kr
접수일자 : 2013년 2월 18일, 수정완료일자 : 2013년 2월 25일, 최종게재확정일자 : 2013년 3월 4일

소프트웨어 개발을 위해서는 반드시 새로운 PCI 인터페이스와 관련된 디바이스 드라이브가 개발 되어야 하며, 또한 프로세서가 LEON2-FT로 변경되었기 때문에 이와 관련된 프로세서 연구 및 VxWorks BSP 개발이 병행되어야 한다.

본 논문에서는 현재 항우연에서 개발 중인 차세대 탑재컴퓨터에서 탑재소프트웨어 개발을 위한 PCI 디바이스 드라이버 개발 방안 및 PCI 소프트웨어 컴포넌트 설계에 대해서 기술하며, 실시간 운영체제인 VxWorks 6.5를 LEON2-FT에 포팅하는 과정에 대해서 설명한다. 그리고 TSIM-LEON2를 이용하여 통신칩과 PCI 인터페이스를 시뮬레이션 및 테스트 방안을 제시한다.

Ⅱ. 차세대 탑재컴퓨터 및 LEON2-FT/AT697F 프로세서

인공위성 탑재컴퓨터에 사용되는 프로세서는 우주환경에서도 정상적으로 동작할 수 있도록 특수하게 설계된다. NASA의 경우 1990년도 초중반에는 인텔계열의 8085/80386 프로세서가 사용되었으나, 2000년대부터는 PowerPC 계열의 RAD750 및 RAD6000 프로세서가 사용되었다. ESA에서는 초기 MIL-STD-1750 프로세서가 사용되었으나, 1990년대 이후 ESA 자체 프로세서인 ERC32가 개발되었으며 2000년대 초중반에는 대부분의 위성에서 ERC32 프로세서를 사용하였고, 이후 차세대 프로세서인 LEON2/3 프로세서를 위성의 탑재프로세서로 사용 중에 있다. 또한 ESA에서는 NGMP (Next Generation Microprocessor)[2] 프로젝터를 통해 멀티코어 기술과 L1/L2 cache를 적용한 LEON4 프로세서를 개발하고 있다.

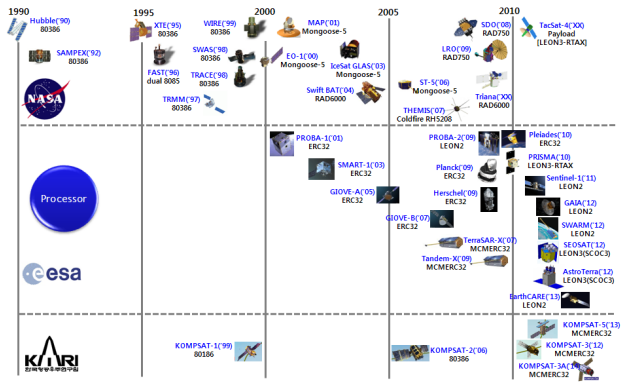

그림 1. Processor Usage in Space

항우연에서는 아리랑위성 1호/2호에서는 인텔계열의 80186/80386 프로세서를 사용하였으며, 아리랑위성 3호/5호/3A에서는 ERC32를 확장한 MCMERC32 프로세서를 채택했으며, 현재 개발 중인 정지궤도복합위성의 경우 ERC32 프로세서를 사용한다. 그리고 차세대 탑재컴퓨터의 경우 아리랑 위성 6호 및 차세대 중형위성 (CAS500, Compact Advanced Satellite 500)에 적용 될 예정이다.

1. ESA LEON Processors

ERC32 프로세서의 경우 20MHz의 시스템 클록의 한계를 가지고 있으며, 인터페이스가 복잡하여 SoC (System On Chip) 형태로 개발이 어려우며 무엇보다 ERC32 프로세서 자체가 미국 라이센서를 기반으로 하고 있어 ERC32 프로세서를 변경/향상[3] 시킬 수 없었다. ESA에서는 이러한 문제를 해결하기 위하여 LEON 프로젝터를 2000년도 초반부터 시작하였으며, LEON 프로세서의 개발 개념으로 유럽 자체 디자인, radiation harden/SEU free, 표준 인터페이스, VHDL (VHSIC Hardware Description Language) 설계, 100MIPS 이상의 고성능 등을 목표[4]로 개발 되었다.

최초 Gaisler Research를 통해 개발 된 LEON1 프로세서는 SPARC v8 (5-stage pipeline), MEIKO FPU, 프로세서의 성능 향상을 위하여 instruction/data cache를 채택하였으며, 외부 확장을 위해 PCI 인터페이스를 사용하였다.

이후 LEON2 프로젝터를 통해 ARM 표준 버스인 AMBA (Advanced Microcontroller Bus Architecture) 버스를 내부 버스로 사용하였으며 소프트웨어 개발과 디버깅을 위해 DSU (Debugging Support Unit)을 채택하였다. Gaisler Research에서 LEON2-1.0.32-xst 이후 개발을 종료하였고 Atmel을 통해 AT697E/AT697F ASIC (Application-Specific Integrated Circuit) 형태로 프로세서[1]가 판매되고 있다. AT697F의 경우 최대 100MHz로 동작하여 90DMIPS/23MFLOPS[5] 이상의 고성능을 보여주며 PROBA-2/3, Exomars Rover, Gaia, Bepi Colombo, SWARM, EarthCare 등 인공위성 메인 프로세서 및 탑재체 컨트롤러로 사용되고 있다.

LEON2 프로세서의 경우 프로세서만을 위해 설계되었으나 ESA 및 협력업체에서 개발 된 다양한 IP (Intellectual Property)를 함께 설계할 수 있는 디자인이 필요했으며 SoC 및 Multi-Processor를 위해 ESA 주관 하에 Aeroflex Gaisler에서 GRLIB IP library를 개발하였다. GRLIB는 LEON3 프로세서, AMBA AHB (Advanced High-performance Bus)/APB (Advanced Peripheral Bus) 버스, 메모리 컨트롤러, Ethernet, SpW, CAN 등 수많은 라이브러리로 구성되며, VHDL으로 코딩되어 있어 쉽게 다양한 FPGA에 합성 할 수 있으며 이를 바탕으로 위성급 ASIC 으로도 개발 되었다. PRISMA, Spot 6/7, Seosat, KRS, Sentinel 5P, CSO 등의 위성에서 LEON3 프로세서를 메인 프로세서로 사용하고 있다.

2. LEON2-FT/AT697F Processor

Atmel에서 개발 된 AT697F는 그림 2와 같이 SPARC v8

기반의 Integer Unit (IU), MEIKO Floating-Point Unit (FPU), 32KB 4-way associative Instruction cache, 16KB 2-way associative Data cache, 하드웨어 기반의 multiplier/divider를 가지고 있으며, APB bus에 인터럽트 컨트롤러, 2개의 32bit 타이머, WDT (Watchdog Timer), 2개의 UART, GPIO를 가지고 있다. PCI 인터페이스를 위해 PCI/AMBA bridge 기반의 PCI controller를 가지고 있으며, 소프트웨어 로딩 및 디버깅을 위하여 DSU와 JTAG을 가지고 있다.

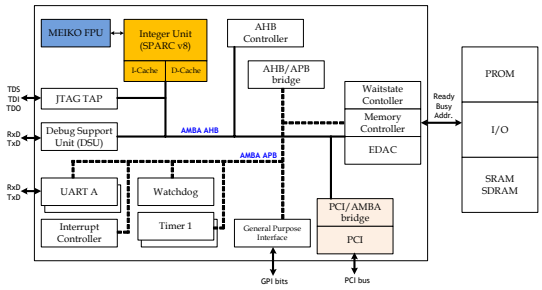

그림 2. AT697F Block Diagram

AT697F에서는 6개의 영역으로 메모리가 할당 되며, 해당 메모리에 대한 디바이스 타입은 아래 표1과 같으며, AT697F에서는 확장성을 위해 PCI를 제공하며 PCI 영역의 경우 non-cacheable 영역으로 0xA0000000 ~ 0xFFFFFFFF 맵핑되어 있다.

표 1. AT697F Memory Mapping

Address Range	Area	Cacheability
0x0000.0000~0x1FFF.FFFF	PROM	Cacheable
0x2000.0000~0x3FFF.FFFF	I/O	Non-cacheable
0x4000.0000~0x7FFF.FFFF	RAM	Cacheable
0x8000.0000~0x8FFF.FFFF	REG	Non-cacheable
0x9000.0000~0x9FFF.FFFF	DSU	Non-cacheable
0xA000.0000~0xFFFF.FFFF	PCI	Non-cacheable

AT697F는 PCI spec 2.2와 호환되며 33MHz로 동작하며, round-robin 알고리즘을 통해 최대 4개의 PCI agent를 관리한다. PCI 인터페이스는 initiator (master) 및 target으로 동작할 수 있으며, data 전송은 8개 word로 구성된 4개의 synchronizing data FIFO를 통해서 수행된다. PCI 관련 레지스터들은 0x80000100 ~ 0x80000280 영역에 할당되어 있으며 Host-Bridge 모드로 동작한다.

3. 차세대 탑재컴퓨터 프로세서 보드

차세대 탑재컴퓨터의 구조는 그림 3과 같이 메인 프로세서로 AT697F를 사용하며, FPGA에 구현된 2개의 통신칩과 PCI 통신을 수행 한다. 2개의 통신칩에는 다양한 미션을 수행 할 수 있도록 SpW, 1553B, DMA UART, UART, CAN, GPIO등의 통신 모듈과 메모리가 AMBA 버스로 연결되며, AT697F와 PCI 통신을 수행 할 수 있도록 PCI Initiator/Target을 가지고 있다. 통신칩에 집적되는 모든 통신 모듈은 VHDL로 개발 된 IP들이며 AMBA 표준 버스를 사용하기 때문에 새로운 모듈의 추가/삭제가 용이하다.

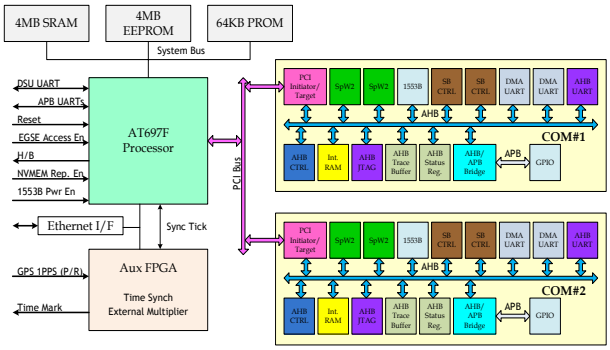

그림 3. NGSCB/PM697 Block Diagram

통신칩의 내부 구조를 확인하면, LEON3/GRLIB 구조와 매우 유사하나 프로세서 모듈 대신 PCI Initiator/Target으로 변경된 것을 알 수 있다. 기존 GRLIB에서 제공되는 모든 라이브러리를 사용할 수 있으며 AMBA 버스를 통해 서로 연결되며, 인터럽트의 경우 PCI Initiator로 라우팅 되어 AT697F 프로세서에 전용 인터럽트로 할당 된다.

Ⅲ. 시스템 소프트웨어 설계/구현 및 테스트

PCI 기반의 LEON2-FT 프로세서를 위한 시스템 소프트웨어 개발을 위해서는 우선 PCI 디바이스 드라이버가 개발되어야 하며, 이를 바탕으로 실시간 운영체제인 VxWorks 6.5를 위한 BSP (Board Support Package)를 개발해야 한다. 아직 차세대 탑재컴퓨터의 개발이 완료되지 않았기 때문에 시스템 소프트웨어를 테스트 및 개발 할 수 있는 시뮬레이터도 함께 개발이 되어야 한다. LEON2-FT와 통신칩을 시뮬레이션 하기 위하여 Aeroflex Gaisler에서 개발 된 TSIM-LEON2[6]를 이용하여 개발/테스트 환경을 구축하였다.

1. AT697F PCI Initialization

AT697F의 PCI를 운영하기 반드시 HOST 모드로 설정되어야 하며, PCI를 통해서 AT697F의 메모리를 액세스 할 수 있도록 SRAM을 PCI Address Space로 맵핑해야 한다. AT697F PCI의 초기화 과정은 아래 표2와 같이 9단계로 수행되어야 하며, 이 과정을 통해 AT697F PCI의 마스터로 동작하며 SRAM과 PCI 영역이 맵핑된다.

2. PCI Initialization for Communication Chips

AT697F PCI 초기화 이후 PCI Configuration Space (PCI/CS)를 통해 통신칩의 PCI를 초기화 해주어야 한다. PCI/CS를 액세스 할 경우 AT697F PCI DMA 기능을 통해 Configuration 명령을 전송하게 되고, 명령을 받은 통신칩은 해당 PCI/CS 레지스터 값을 설정 또는 전송하게 된다. 그림 4는 통신칩#0의 Device ID를 읽어오는 과정으로 AT697F에서 PCI 레지스터를 설정한 뒤 PCI DMA를 통해 Configuration Read 명령을 전송하면, PCI/CS 주소 0x00 (Device ID)을 읽어 오게 된다. 수신된 데이터는 PCI DMA를 통해서 AT697F SRAM 특정 영역에 해당 레지스터의 값이 기록된다.

표 2. AT697F PCI Initialization Step

Step	PCI Operation on AT697F
#1	Check AT697F PCI Device ID
#2	Reset AT697F PCI Core
#3	Mask all AT697F PCI Interrupts
#4	Map AT697F SRAM at PCI Address Space
#5	Set AT697F PCI Target Page Address
#6	Enable AT697F PCI Master and Target Memory Command Response
#7	Set AT697F Latency Timer
#8	Set AT697F PCI Initiator Configuration
#9	Enable all AT697F PCI Interrupts

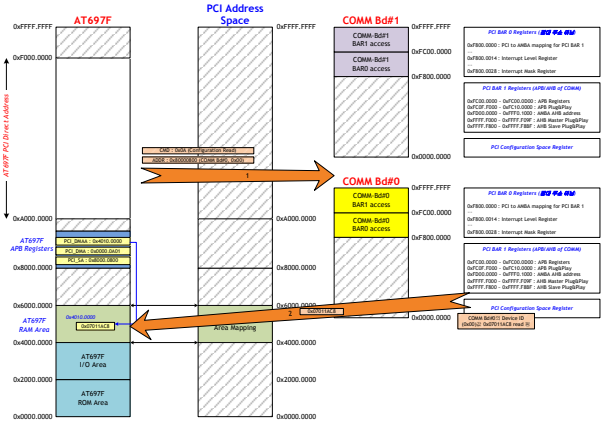

그림 4. PCI Configuration Space Read Operation

통신칩의 PCI를 초기화 하는 과정은 PCI/CS를 통해서 이루어지며, 아래 표3과 같이 14단계를 통해 이루어진다. 이 과정을 통해 통신칩의 PCI Address Space가 BAR0/BAR1 레지스터를 통해 설정되며, Master 모드로 동작하게 된다.

모든 PCI 초기화 및 설정 이후 그림 5와 같이 AT697F의 0xC8000000 ~ 0xD0000000 영역에 통신칩#0이 맵핑되며, 0xD0000000 ~ 0xD8000000 영역에 통신칩#1이 맵핑된다. 결과적으로 AT697F에서는 해당 영역에 통신칩 관련 레지스터와 메모리가 할당 되어 Direct Addressing 기법 즉 Memory Mapped I/O 방식으로 통신칩을 액세스 할 수 있으며, 대용량의 데이터를 전송 할 경우에는 DMA 기능을 통해서 송수신 할 수 있다.

표 3. Comm. Chips PCI Initialization Step

Step	Addr	Operation for Communication Chip
#1	0x00	Read Device ID/Vendor ID
#2	0x04	Read Status/Command
#3	0x08	Read Class Code & Revision ID
#4	0x0C	Read BIST, Header Type, Latency Timer
#5	0x04	Clear Status/Command
#6	0x10	Write PCI address to BAR0
#7	0x10	Read BAR0 and check
#8	0x14	Write PCI address to BAR1
#9	0x14	Read BAR1 and check
#10	0x04	Enable Memory Space & Bus Mater to Status/Command
#11	0x04	Read Status/Command and check
#12	0x0C	Set Latency Timer
#13	0x3C	Set Interrupt Line
#14	-	PCI to AMBA mapping for PCI BAR1

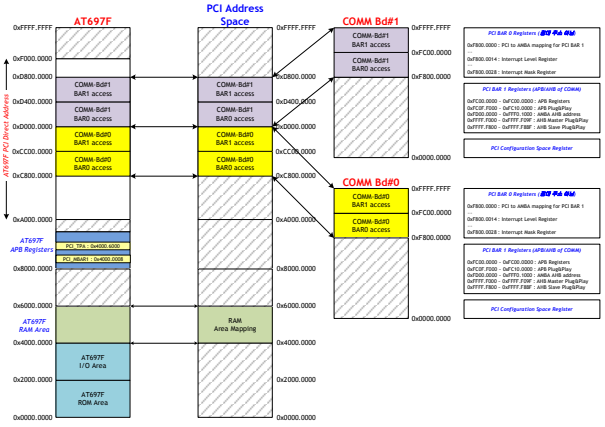

그림 5. PCI Address Space after Initialization

3. AT697F Access to Communication Chips

AT697F의 PCI initiator는 앞에서 설명한 것처럼 PCI Direct Addressing Controller를 이용한 Memory-Mapped 액세스를 지원하며, 초기화 과정에서 설정된 PCI 영역을 디바이스 드라이브에서 메모리 접근 방식으로 액세스 할 경우 자동적으로 PCI 주소로 변경되어 통신칩의 레지스터와 메모리를 접근 할 수 있다. 이 방식을 사용할 경우 마치 AT697F 보드에 해당 디바이스가 있는 것처럼 사용할 수 있기 때문에 디바이스 드라이브 개발 시 용이하며 현재 통신칩의 모든 디바이스들은 이 영역 안에 맵핑되어 있다. 아래 그림 6은 디바이스 드라이브에서 통신칩#0의 특정 레지스터를 변경하기 위해 PCI 주소로 접근 할 경우 자동적으로 PCI 주소로 변환되어 해당 레지스터 값을 변경하는 과정을 보여준다.

PCI DMA Controller를 이용할 경우 PCI Address Space에 할당 되지 않은 영역도 모두 액세스 할 수 있으며, 대용량의 데이터를 송수신 하는 경우 DMA transaction을 사용할

경우 CPU 점유율을 낮출 수 있는 장점을 가지고 있다. 통신칩의 1553B, SpW의 경우 자주 다량의 데이터를 송수신하기 때문에 DMA 방식이 용이하며, 통신칩에서도 DMA 방식을 통해 자동적으로 AT697F SRAM에 데이터를 송신할 수 있다. DMA 완료 후 자동적으로 PCI 인터럽트가 발생하여 상위 응용프로그램에게 데이터 송/수신의 완료를 알리게 된다. 그림 7은 디바이스 드라이브를 통해 통신칩#0의 특정 영역을 읽어오는 과정으로, AT697F의 PCI DMA 관련 레지스터를 설정한 후 DMA operation을 활성화 하면 자동적으로 SRAM 특정 영역에 데이터가 수신되고 PCI 인터럽트가 발생한다.

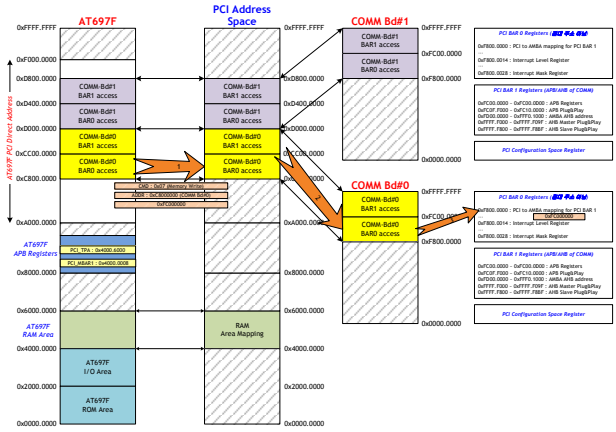

그림 6. AT697F PCI Direct Addressing for Write Operation

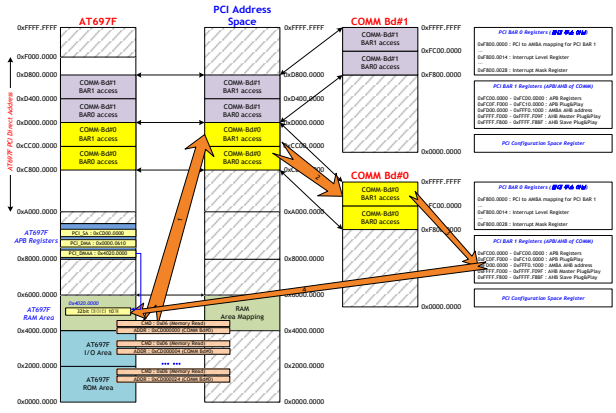

그림 7. AT697F PCI DMA Transfer for Read Operation

4. AT697F & Communication Chips Simulation

Aeroflex Gaisler에서 판매하고 있는 TSIM-LEON2 에뮬레이터만이 LEON2 프로세서를 지원하며, TSIM-LEON2에서는 Atmel AT697 PCI를 에뮬레이션 할 수 있도록 외부 모듈 로딩 기능을 제공 하고 있다. PCI로 제어되는 통신칩을 시뮬레이션을 하기 위하여 Communication Chips Emulation Module을 개발 하였으며 TSIM-LEON2 구동시 함께 로딩되어 수행된다.

통신칩에 대한 에뮬레이션 기능은 PCI로 액세스 될 때 PCI Access 모듈이 수행되고, 메모리 접근시 PCI 주소에 따라 BAR0/BAR1로 구분한 뒤, BAR0 액세스인 경우 PCI Address Space를 시뮬레이션 해주며, BAR1 액세스인 경우 각 통신 모듈의 코어에 대한 에뮬레이션을 할 수 있도록 개발하였다. 시뮬레이션 모델은 모두 C 언어로 개발되었으며, 동적 다운로드 기능을 가지고 있다. 현재 기본적인 통신 모듈에 대한 모사 기능만이 구현되었으며 개발 진행에 따라 점차적으로 모든 기능을 구현 할 예정이다.

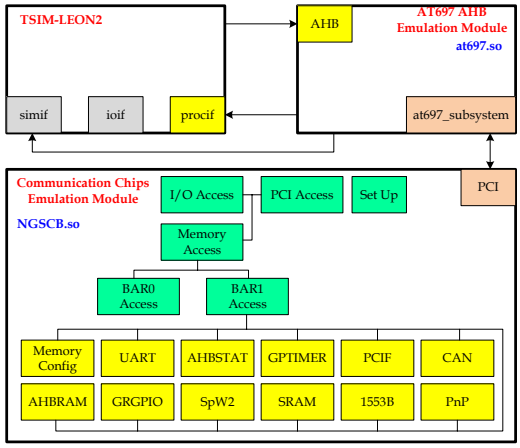

그림 8. Communication Chips Emulation Model

5. 시스템 소프트웨어 테스트

개발된 디바이스 드라이브와 함께 시스템 소프트웨어를 시뮬레이터에 로딩 및 수행한 결과 정상적으로 PCI I/F를 모두 초기화하였으며, 통신칩의 Plug & Play (PnP) 정보를 정확히 액세스 할 수 있었다.

그림 9. AT697F/Comm. Chips Initialization Test

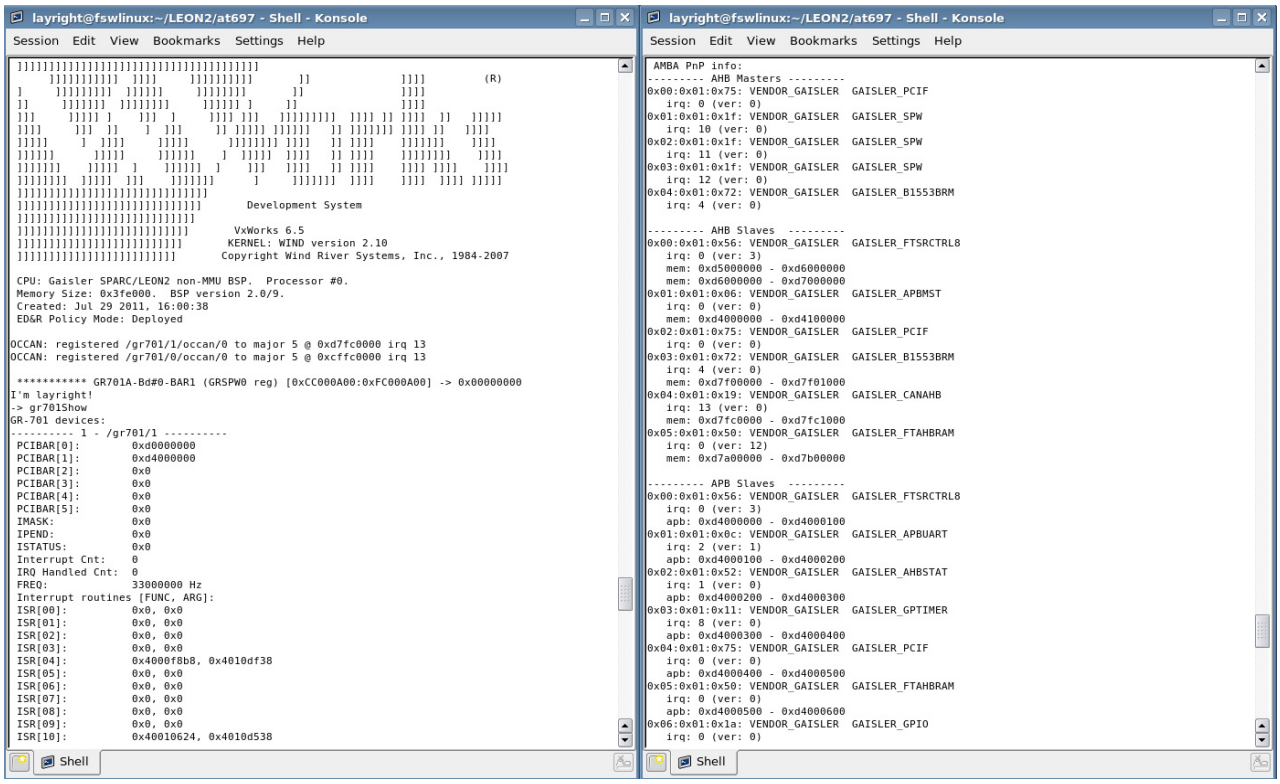

그림 10. VxWorks 6.5 Test with Simulation Model

이를 바탕으로 실제 개발 중에 있는 NGSCB/PM697 보드에서도 동일한 디바이스 드라이브를 이용하여 테스트 한 결과도 시뮬레이터의 결과와 동일하게 나왔으며, 추후 NGSCB/PM697 보드 개발 완료 이후 다양한 기능을 구현 할 예정이다.

6. VxWorks 6.5 BSP 포팅 및 테스트

WindRiver에서는 현재 멀티코어까지 지원할 수 있는 VxWorks 6.9를 판매하고 있으나, 공식적으로 VxWorks 5.4 이후 SPARC 코어를 지원하지 않고 있다. 하지만 Aeroflex Gaisler에서는 자체적으로 VxWorks 6.5/6.7을 LEON2/3용으로 포팅 하였으며 NASA APL (Applied Physics Laboratory)에서는 cFE (Core Flight Executive) 개발[7]을 위하여 GR712RC (Dual LEON3-FT Processor)[8]에 VxWorks 6.7 사용하였다. 현재 항우연에서는 ERC32를 사용하는 탑재컴퓨터에서는 VxWorks 5.4를 사용하고 있으나 추후 LEON3/4를 사용할 경우 MMU (Memory Management Unit)와 멀티코어를 위해 VxWorks 6.7로 변경이 불가피하다. 그리고 현재 VxWorks 5.4에서는 PCI 라이브러리를 지원하지 않기 때문에 테스트를 위해서 VxWorks 6.5를 사용하였으며, 기존에 개발 된 LEON3-FT 프로세서용 VxWorks 6.5 BSP를 기반으로 LEON2-FT BSP가 개발 되었다. 그리고 통신칩에 필요한 소프트웨어 모듈은 Aeorflex Gaisler에서 제공하는 GR701A[9]를 기반으로 개발 되었으며, 추후 자체 모듈로 개발 할 예정이며, 또한 VxWorks 5.4에서도 구동 될 수 있게 모든 PCI 라이브러리를 개발 할 계획이다.

그림 10은 빌드된 VxWorks 6.5를 TSIM-LEON2와 함께 통신칩 시뮬레이션 모델을 로딩 한 뒤 테스트 한 결과이다. 정상적으로 VxWorks 실행 이후 PCI 인터페이스를 통해 초기화 및 디바이스가 등록 되었으며, 통신칩의 PnP 정보를 가져올 수 있었으며, 다양한 테스트를 통해 통신칩 모듈을 액세스 할 수 있었다.

Ⅳ. 결론

본 논문에서는 차세대 인공위성 탑재컴퓨터의 프로세서로 사용될 LEON2-FT/AT697F상에서 PCI 내부 버스를 통해 통신칩을 제어하기 위한 시스템 소프트웨어 개발 방안과 PCI 소프트웨어 컴포넌트 설계에 대해서 설명하였다. 그리고 테스트 환경을 구축하기 위하여 TSIM-LEON2와 개발된 시뮬레이션 모델을 통해 PCI 초기화 과정과 VxWorks 6.5를 로딩 후 모든 PCI 운영이 정상적으로 수행되었다.

시뮬레이터를 통해서 나온 결과와 실제 하드웨어에서 테스트 한 결과가 동일한 것을 확인하였으며, 추후 통신칩에 설계된 다양한 통신 모듈을 시뮬레이션 함으로서 소프트웨어 개발 플랫폼을 구성 할 수 있을 것으로 판단된다. 또한 개발된 PCI 디바이스 드라이브를 VxWorks 5.4에 추가적으로 포팅 할 경우 VxWorks 5.4에서도 PCI 인터페이스를 사용할 수 있을 것으로 판단된다.

참 고 문 헌

[1] Atmel, "Rad-Hard 32bit SPARC V8 Processor AT697F", 2008.
[2] The ESA Next Generation Microprocessor (NGMP), http://microelectronics,esa.int/ngmp/
[3] Jiri Gaisler, "LEON-1 Processor - First Evaluation Results," European Space Components Conference (ESCCON), Vol. 439, pp.183~187, 2000.
[4] Jiri Gaisler, "LEON SPARC Processor The past, present and future," RAMP (Research Accelerator for Multiple Processors), 2007.
[5] Atmel, "Atmel Space Rad-Hard Processors and Communications ICs," 2012.
[6] Aeroflex Gaisler, "TSIM2 Simulator User's Manual v2.0.24,", October, 2012.
[7] David Edell, "Multi-Core Processing in Flight Software," Workshops on Spacecraft Flight Software, Nov. 7-9, 2012.
[8] Aeroflex Gaisler, "Dual-Core LEON3-FT SPARC v8 Processor GR712RC User's Manual,", October, 2011.
[9] Aeroflex Gaisler, "PCI to SpaceWire and 1553 Bridge GR701A User's Manual v1.0.0,", June, 2007.

저자

최 종 욱(Jong Wook Choi) **정회원**

- 1999년 2월 : 경북대학교 전자공학 학사졸업
- 2001년 2월 : 경북대학교 전자공학 석사졸업
- 2000년 12월~현재 : 한국항공우주 연구원 위성비행소프트웨어 팀

<관심분야> : 시뮬레이터, 실시간운영체제

남 병 규(Byeong Gyu Nam)

- 1999년 2월 : 경북대학교 컴퓨터공학 학사졸업
- 2001년 2월 : 한국과학기술원 전기 및 전자공학 석사졸업
- 2007년 2월 : 한국과학기술원 전기 및 전자공학 박사졸업
- 2001년 1월~2002년 2월 : 한국전자통신연구원 책임연구원
- 2007년 4월~2010년 8월 : 삼성전자 선임 연구원
- 2010년 9월~현재 : 충남대학교 컴퓨터공학과 교수

<관심분야> : 모바일프로세서, GPU, SoC