

차세대 저궤도 위성의 Boot ROM 소프트웨어 설계 및 검증에 대한 연구

최종욱*, 이재승**, 양승은***, 신현규****, 채동석*****,
이종인*****

A Study of the Boot ROM S/W Design and Verification for the Next Generation LEO Satellite

Jong-Wook Choi*, Jae-Seung Lee**, Seung-Eun Yang***, Hyun-kyu Shin****,
Dong-Seok Chae*****,
Jong-In Lee*****

Abstract

The next generation LEO satellite has 64KB PROM which contains the boot loader and the monitor software, and two 4MB NVMEMs which are used for flight software storage. The boot loader has two operation modes which are the flight software mode and the monitor mode. In the flight software mode, it checks CRC checksum of selected NVMEM and copies flight software image from NVMEM to RAM. And then it starts VxWorks RTOS in RAM, creates flight software tasks, and starts execution of flight software. In the monitor mode, it activates monitor software which performs NVMEM reprogramming and board-level testing on the ground. This paper is to present the design of Boot ROM software and verification method using simulator.

초 록

차세대 저궤도 위성의 탑재소프트웨어 저장메모리는 부트 코드와 모니터 소프트웨어를 내장한 부트 루프과 비행소프트웨어 이미지를 저장한 듀얼 NVMEM 형태로 개발되고 있다. 부트 루프에 내재된 부트 로더는 프로세서 초기화 이후 GPIO의 입력에 따라 비행소프트웨어 모드 혹은 모니터 모드로 전이하게 된다. 비행소프트웨어 모드에서는 듀얼 NVMEM중 선택된 NVMEM에 대한 CRC를 체크하여 정상적일 경우 NVMEM에 저장되어 있는 탑재소프트웨어를 코드가 수행 될 RAM으로 적재 한 후 VxWorks RTOS를 구동하여 테스크를 생성하여 탑재소프트를 수행하게 된다. 모니터 모드에서는 지상에서 NVMEM reprogramming을 가장 메인으로 수행하며 기본적인 보드 레벨 테스트를 수행할 수 있다. 본 논문에서는 차세대 저궤도 위성의 부트 루프 소프트웨어 설계 및 시뮬레이터 기반의 검증 방법에 대하여 기술한다.

키워드 : 부트 로더(boot loader), 시뮬레이터(simulator),
탑재컴퓨터(MCMERC32SC, PM32)

접수일(2007년 12월 14일), 수정일(1차 : 2008년 6월 5일, 2차 : 2008년 6월 17일, 게재확정일 : 2008년 7월 1일)

* 위성SW팀/jwchoi@kari.re.kr
*** 위성SW팀/seyang@kari.re.kr
***** 위성SW팀/dschae@kari.re.kr

** 위성SW팀/jslee@kari.re.kr
**** 위성SW팀/hkshin@kari.re.kr
***** 위성SW팀/jilee@kari.re.kr

1. 서 론

기존 저궤도 위성은 부트 루트(Boot ROM)으로 1개의 EEPROM만을 가진 단일 구조 형태로 구성된다. 반면 차세대 저궤도 위성에서는 부트 코드와 모니터 소프트웨어를 내장한 부트 룰과 비행소프트웨어 이미지를 저장한 듀얼 NVMEM (Non-Volatile MEMory) 형태로 구성된다. 차세대 위성의 프로세서 모듈인 IBMU(Integrated Bus Management Unit) PM32(Processor Module 32bit)는 3개의 boot-up 메모리로 구성된다. 탑재 CPU인 MCMERC32SC의 메모리 영역[표 1 참조] 0x00000000~0x0000FFFF에 맵핑되어 있는 64KB 부트 룰은 부트 로더(boot loader)와 모니터 소프트웨어(Monitor Software)를 포함하고 있으며 실제 메모리 타입은 PROM을 사용한다. MCMERC32SC의 메모리 영역 0x00800000~0x00FFFFFF에 맵핑되어 있는 2개의 NVMEM은 4MB FLASH 메모리로 비행소프트웨어 이미지를 저장한다. PM32에 파워가 인가되거나 PM32 리셋되었을 때 부트 룰에 저장되어 있는 부트 로더가 구동된다. 부트 로더는 MCMERC32SC 프로세서를 초기화하고 MCMERC32SC GPIO(General Purpose Input/Output) bit<5>에 할당되어 있는 NVMEM Reprogramming Enable 입력에 따라 비행소프트웨어(Flight Software) ROM Entry 또는 모니터 소프트웨어로 천이한다. 비행소프트웨어 ROM Entry에서는 선택된 NVMEM에서 비행소프트웨어 이미지에 대한 CRC 체크 한 뒤 RAM으로 복사하고 부트 로그(boot log)를 RAM의 특정영역에 저장한 뒤 RAM에 있는 VxWorks를 구동하여 비행소프트웨어 테스크(task)가 생성되어 비행소프트웨어가 시작된다. 모니터 소프트웨어에서는 가장 메인인 되는 NVMEM Reprogramming을 수행하며 기본적인 테스트들을 수행할 수 있다.

표 1. Memory Layout

IBMU PM32 Memory Map	Mode	R/W	IBMU PM32 Memory Map	Mode	R/W
0x0000 0000 Boot ROM (64KB)	8Bit	R	0x0400 0000 Extended RAM : N/A	32Bit	R/W
0x0000 FFFF			0x00FF FFFF		
0x0080 0000 NVMEM A (4MB)	8Bit(W) 32Bit(R)	R/W	0x1000 0000 SMC S32 #1 Register (352Bytes)	32Bit	R/W
0x008F FFFF			0x1000 018C		
0x00C0 0000 NVMEM B (4MB)	8Bit(W) 32Bit(R)	R/W	0x1100 0000 SMC S32 #2 Register (352Bytes)	32Bit	R/W
0x0100 0000			0x1100 019C		
0x01EF FFFF Extended PROM : N/A	8Bit	R	0x1200 0000 DRAM #1 (32KB)	32Bit	R/W
0x01FF FFFF			0x1200 FFFF		
0x01FB 0000 Exchange Memory : N/A	32Bit	R/W	0x1200 0000 DRAM #2 (32KB)	32Bit	R/W
0x01FB 0008 ERC32 MEC Registers (12-Bytes)	32Bit	R/W	0x1300 0000		
0x01FB 000E			0x1300 0000		
0x0200 0000			0x1300 00FF		
0x023F FFFF SRAM (6MB)	32Bit	R/W	0x1400 0000		
0x0240 0000			0x1400 0920		
0x02FF FFFF DRAM (Max 32 MB) : 32MB Used	32Bit	R/W	0x8000 0000		
0x02FF FFFF			0x8FFF FFFF		
			Extended General Area : N/A	32Bit	R/W

2. Boot ROM Software

2.1 Boot ROM Software Modes

부트 룰 소프트웨어는 크게 부트 로더, 비행소프트웨어 ROM Entry, 모니터 소프트웨어 ROM Entry, 모니터 소프트웨어 RAM 이미지 4가지로 구성된다.

① 부트 로더 : MCMERC32SC의 IU, MEC, FPU를 초기화하고 메모리, waitstate, I/O configuration 등을 설정한다. 마지막으로 ERC32 GPIO<5>의 입력에 따라 비행소프트웨어 ROM Entry나 모니터 소프트웨어 ROM Entry로 천이 한다.

② 모니터 소프트웨어 ROM Entry : ERC32 GPIO<5>이 0(Enable)일 때 모니터 소프트웨어 ROM Entry가 구동되며, ERC32 UART 통신 채널을 형성하며, 부트 룰에 저장되어 있는 모니터 소프트웨어 RAM 이미지를 RAM 0x025F0000으로 복사한 후 RAM에서 모니터 소프트웨어를 구동한다.

③ 모니터 소프트웨어 RAM 이미지 : 모니터 소프트웨어는 대부분 C로 코딩되었고 스택(stack) 영역을 사용하기 때문에 RAM에서 구동된다. 부트 룰 소프트웨어 빌드(build)시 모니터 소프트웨어는 RAM에서 구동될 수 있게 빌드되고 부트 룰에 이미지 형태로 저장되어, 모니터 소프트웨어 ROM Entry에서 이 이미지를 RAM으로 옮겨서 구동한다.

④ 비행소프트웨어 ROM Entry : ERC32 GPIO<5>이 1(Disable)일 때 비행소프트웨어 ROM Entry가 구동되며, 초기 운영에 필요한 신호와 정보를 ERC32 GPIO, I/O3 Area FPGA Register를 통해서 주고받으며, 선택된 NVMEM에서 비행소프트웨어 이미지의 CRC를 체크하여 비행소프트웨어 이미지를 RAM에 복사하고, 부트 로그를 RAM에 저장한 뒤 RAM에 있는 VxWorks를 구동한다.

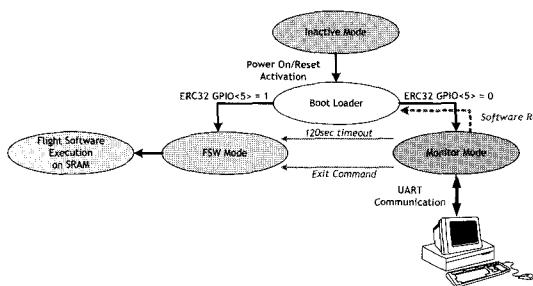


그림 1. Boot ROM Software Mode Diagram

2.2 Boot Loader

최초 MCMERC32SC 프로세서에 리셋/파워가 인가되면 ERC32와 VASI는 초기화 되고 레지스터들은 초기 값으로 설정되거나 마지막으로 수행한 상태를 유지한다. 리셋이 되었을 때 리셋 벡터는 0x00000000로 설정되고 SPARC의 PC (Program Counter)와 nPC(next PC)는 각각 0x00000000과 0x00000004로 설정되어 부트 루 영역 0x00000000부터 시작하게 된다. 부트 루 영역의 0x00000000~0x00000FFF(4KB)는 부트 루 트랩 테이블(trap table)이 위치한다. 트랩 테이블의 위치를 결정하는 중요한 레지스터로서 TBR(Trap Base Register)이 사용되며 리셋이 되면 TBR 초기화 되지 않고 기존 값을 그대로 가지고 있기 때문에 부트 로더에서 반드시 TBR을 0으로 초기화 해주어야 한다. tba(trap base address)는 20bit로 구성되며 트랩 테이블 위치에 맞게 설정해주어야 한다. tt는 트랩 타입으로서 트랩이 발생했을 때 해당 트랩 타입이 저장되며 해당 트랩 테이블로 점프하게 되고 트랩 테이블에서는 트랩 handler를 호출하여 트랩을 처리하게 된다.

SPARC은 256개의 트랩을 가지고 있으며 트랩 0x00은 리셋 트랩으로서 부트 로더 코드로 점프하도록 구성된 반면 나머지 255개의 트랩은 모두 UNIMP 코드로 설정되었다. 부팅 과정에서 기본적으로 트랩은 비활성화로 설정되나 만약 트랩이 발생할 경우 IU error 모드로 천이하게 되고 모든 동작을 멈추게 된다. 결국 3분이 지난 뒤 RU(Reconfiguration Unit)에 의한 Heartbeat failure event에 의한 System Reconfiguration이

수행된다.

부트 로더는 그림 2와 같이 MCMERC32SC 프로세서를 초기화 하는 과정을 수행한다. 초기화 과정에서는 메모리, waitstate를 설정하고, 모든 인터럽트를 비활성화하고 전체 6MB의 RAM과 레지스터들을 초기화 한다. 마지막 단계에서 GPIO<5> NVMEM Reprogramming Enable 입력에 따라 비행소프트웨어 ROM Entry나 모니터 모드 ROM Entry로 천이 한다.

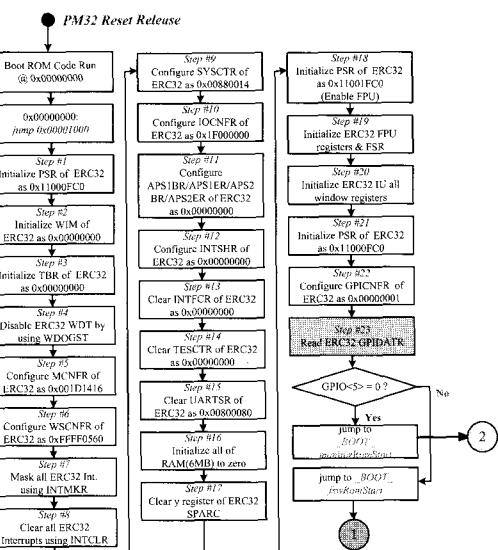


그림 2. Behavior of boot loader function

2.3 Flight Software ROM Entry

비행소프트웨어 ROM Entry는 PROM에서의 비행소프트웨어의 시작 코드로서, EGSE Reprogramming 비활성화 되었을 경우 수행된다. 전체적인 수행방식은 그림 3과 같다. 비행소프트웨어 ROM Entry에서는 제일 먼저 GPIO<0>을 통하여 Launch Abort Signal을 생성하고, NVMEM Power On/Write Enable 명령을 수행한다. NVMEM 상태 레지스터를 통해 NVMEM A/B중 선택된 NVMEM를 확인하고, 선택된 NVMEM에 대한 CRC32계산을 수행한다. 계산된 CRC32 결과와 pre-stored된 CRC32값이 동일 할 경우 부트 로그를 RAM에 저장하고, 그림 4와

같이 NVMEM에 저장되어 있는 비행소프트웨어 /VxWorks를 RAM으로 복사한 뒤 VxWorks를 구동하여 비행소프트웨어 테스크를 생성하여 정상적인 운영을 수행한다. 만약 선택된 NVMEM의 CRC32계산이 다를 경우 다른 NVMEM를 선택하여 동일한 과정을 수행한다. 모든 NVMEM의 CRC32계산이 다를 경우 RU에 의한 System Reconfiguration이 수행되도록 heartbeat failure event를 발생시킨다.

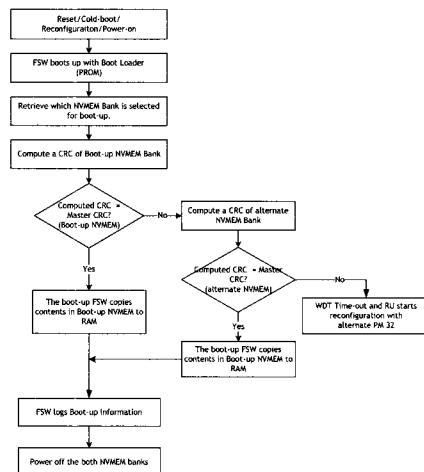


그림 3. FSW ROM Entry Conceptual Operation

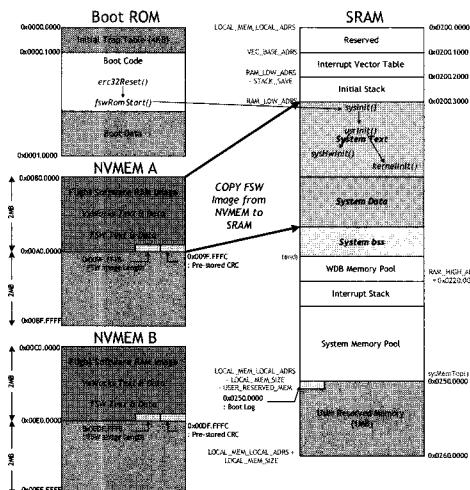


그림 4. NVMEM Copy and VxWorks Start

2.4 Monitor Software

모니터 소프트웨어 ROM Entry는 PROM에 저장되어 있는 모니터 소프트웨어 RAM 이미지를 RAM 마지막 영역(0x025F0000)으로 복사한 뒤 모니터 소프트웨어 RAM 코드를 수행한다. 모니터 소프트웨어는 ERC32 UART B를 통해 PC에서 구동되는 RomLoader 소프트웨어와 연결된다. 모니터 소프트웨어는 지상에서 비행소프트웨어 변경을 위한 NVMEM를 Reprogramming하는 것을 주목적으로 하며, NVMEM read/write/erase/control 등의 기능을 제공하며, 비행소프트웨어 이미지를 RAM에 업로드 하는 기능과 CRC32 계산, RAM 코드 수행 등의 기능을 제공한다.

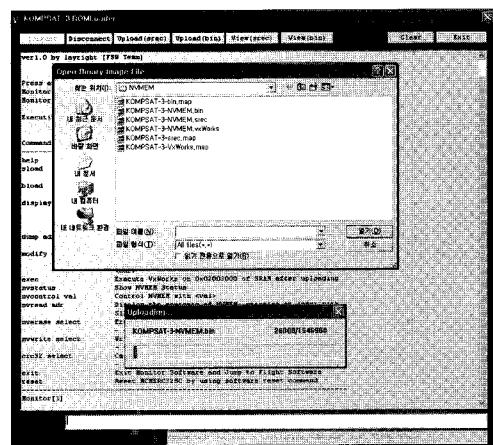


그림 5. Flight Software Image Upload

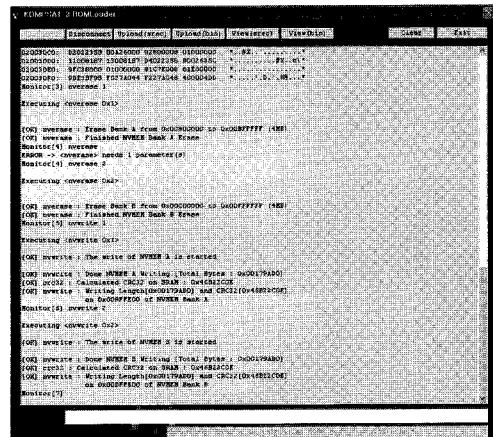


그림 6. NVMEM Reprogramming

모니터 소프트웨어에는 트랩/인터럽트가 활성화 된다. 모니터 RAM 소프트웨어는 C로 코딩되었으며 C 코드에서 기본적으로 스택을 사용하고 함수를 호출/리턴 할 때마다 SPARC window를 save/restore하기 때문에 window under/over flow 트랩이 발생할 수 있다. 또한 모니터 모드에서 120초 동안 아무런 입력이 없으면 비행소프트웨어 ROM Entry로 천이되기 위해서는 ERC32의 RTC(real time clock) 인터럽트를 사용해야 하기 때문에 인터럽트를 활성화한다.

2.5 NVMEM(Flash) Memory Operation

비행소프트웨어가 저장되어 있는 NVMEM는 2개의 4MB Flash로 구성된다. Flash는 512개의 block으로 구성되며 block은 다시 16개 page로 구성된다. NVMEM는 operation은 크게 read, write, erase, write protection, power control, NVMEM control 등으로 나뉜다.

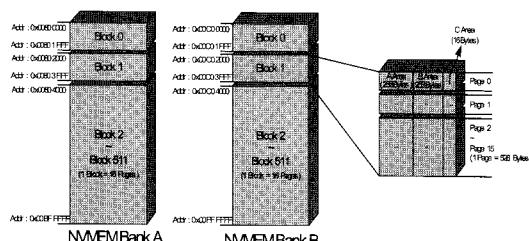


그림 7. NVMEM Configuration

NVMEM read는 page(512bytes) 단위로 수행되며 NVMEM에 저장되어 있는 비행소프트웨어를 RAM으로 복사하기 위해서는 아래 그림 8과 같은 과정을 반복 수행하게 된다.

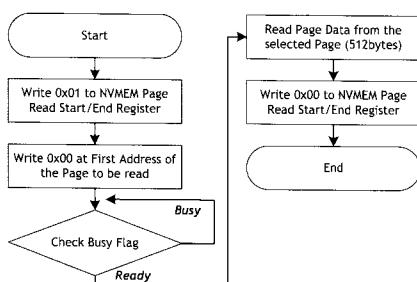


그림 8. NVMEM Page Read Operation

NVMEM write operation도 page 단위로 수행되며 erase operation은 block(8192bytes) 단위로 수행된다. NVMEM는 기본적으로 부팅과정에서만 Power On되고 부트과정이 끝나면 Power Off된다. 추후 On-Orbit NVMEM Reprogramming을 수행할 경우 NVMEM Power On하여 NVMEM Reprogramming을 수행하게 된다.

2.6 Boot Software Build

부트 루트 소프트웨어는 아래 그림 9와 같은 hierarchy를 가지고 있다. 부트 로더, 비행소프트웨어 ROM Entry, 모니터 소프트웨어 ROM Entry는 PROM에서 수행되며, 모두 어셈블리로 구현되었다. 모니터 소프트웨어 RAM 코드는 트랩/인터럽트와 관련된 모듈은 어셈블리로 구현되었으며 다른 코드들은 C로 구현이 되었다.

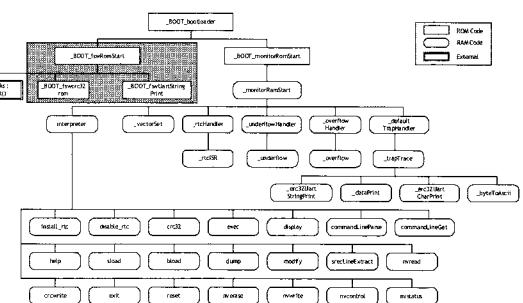


그림 9. Boot ROM S/W Hierarchy

부트 루트 S/W를 빌드하기 위하여 Tornado 2.0 for SPARC에서 제공하는 toolchain을 사용한다. C/ASM 컴파일을 위하여 `ccsparc`과 linker인 `ldsparc`을 사용한다. 빌드 과정에서 모니터 소프트웨어 RAM 모듈을 부트 루트에 저장하기 위해서 `binToAsm` 명령을 사용하여 이미지 파일을 생성하고 부트 루트 빌드 시 이미지 형태로 저장된다.

부트 루트인 PROM은 64KB이기 때문에 전체 부트 루트 소프트웨어는 64KB에 맞게 빌드되어야 한다. 전체적으로 빌드된 부트 루트 소프트웨어의 메모리 사용은 그림 8과 같이 구성되며 74.5%를 사용한다. 부트 루트 트랩 테이블은 부트 루트의 가장 최하위에 위치하며, 부트 로더는 4KB 이후에

위치하게 된다. 비행소프트웨어 ROM Entry는 특정영역에 위치하게 되고, 모니터 소프트웨어는 ROM의 경우 부트 로더 이후에 배치되며 그 뒤에 RAM으로 복사될 모니터 소프트웨어 RAM 이미지는 _binArrayStart/_binArrayEnd의 주소를 가지며 이미지 형태로 PROM에 저장되게 된다.

3. Boot ROM Software 검증

3.1 차세대 위성 IBMU 시뮬레이터

비행소프트웨어 검증을 위해 위성소프트웨어 팀에서는 IBMU 시뮬레이터를 개발하였다. IBMU 시뮬레이터는 Gaisler Research에서 개발된 TSIM ERC32를 기반으로 IBMU PM32에서 사용하고 있는 MCMERC32SC 프로세서를 구현/

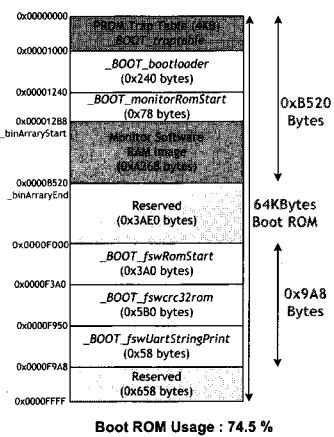


그림 10. Boot ROM Memory Usage

시뮬레이션 하였다. 또한 IBMU PM32가 가지고 있는 FLASH A/B, SpaceWire 통신을 담당하는

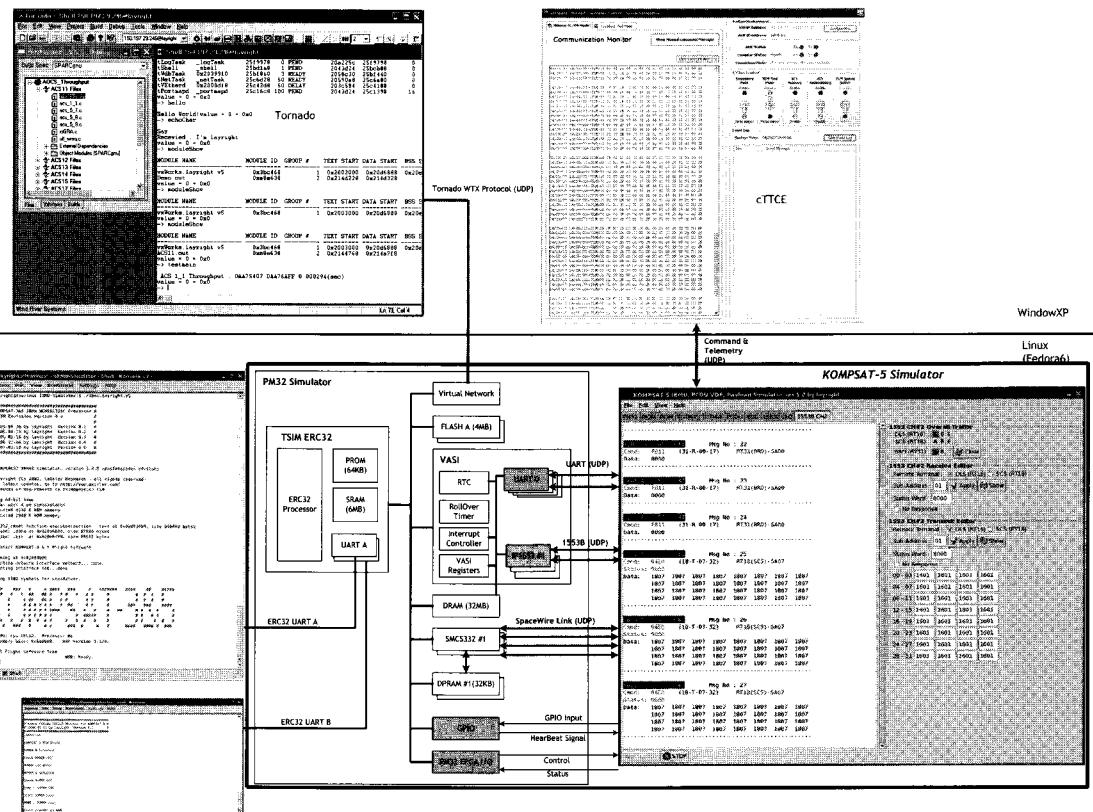


그림 11. 차세대 위성 IBMU 시뮬레이터 구성

SMCS332 #1, DPRAM #1, GPS와 동기를 수행하는 External RTC, 외부 인터페이스인 Payload, GPS등과 통신을 수행하는 1553B 모듈이 구현되었으며, PCDU/VDE와 통신을 수행하는 UART 모듈이 개발되어 PM32 전체를 시뮬레이션 한다. IBMU TCTM, PIOM, BIOM 시뮬레이터 또한 개발되어 PM32와 SpaceWire 통신을 수행하게 되고, PM32의 1553B와 연결된 GPS, Payload도 1553B Protocol을 지원하는 시뮬레이터가 개발되었다.

비행소프트웨어 개발 시 Linux에서 구동되는 IBMU 시뮬레이터와 Window에서 구동되는 비행소프트웨어의 개발 툴인 Tornado를 virtual ethernet을 통하여 Tornado Target Server로 연결되어 실시간으로 비행소프트웨어를 다운로딩 및 수행, 디버깅을 수행할 수 있다.

3.2 시뮬레이터 기반의 Boot ROM Software 검증

실제 부트 루트 소프트웨어를 검증하기 위해서는 하드웨어 레벨에서 ICE(In-Circuit Emulator)나 JTAG과 같은 디버깅 장비를 지원해야 하나 현재 MCMERC32SC를 지원하는 ICE 및 JTAG이 존재하지 않기 때문에 부트 루트 소프트웨어는 개발된 IBMU/PM32 시뮬레이터에서 코드 레벨 검증을 수행하였다. 비행소프트웨어의 검증은 단위시험(unit test) 및 code coverage에 대한 테스트가 우선적으로 수행되었으며 SRS(Software Requirement Spec.)에 따라 verification test가 수행되었다.

PM32 시뮬레이터에 부트 루트 소프트웨어를 로딩하고 그림 2와 같이 전체 Step 23단계까지 단계별로 테스트를 수행하게 된다. 그림 12는 부트로더가 ERC32 system register를 설정하는 부분을 보여주며 초기화 하는 부분까지 수행 후 step 기능을 이용하여 각 레지스터 변경되는지를 확인하였으며 GPIO<5> 입력에 따라 비행소프트웨어 ROM Entry로 천이하는지 모니터 모드로 천이하는지를 검증하였다.

```

layright@fswlinux:~/KOMPSAT-5-Simulator> tsim> cont
breakpoint 3
tsim> reg
0x1f86608 Error and status register 0x00000000
0x1f86609 Text control register 0x00000000
tsim> cont
breakpoint 3
tsim> reg
0x1f86608 MEC control register 0x00000014
0x1f86610 Memory control register 0x00101416
0x1f86614 I/O control register 0x10000000
0x1f86618 Waitstate control register 0xffff0050
0x1f86620 Access (1) base register 0x00000000
0x1f86622 Access (2) base register 0x00000000
0x1f86624 Access (2) end register 0x00000000
0x1f86644 Interrupt shape register 0x00000000
0x1f8664c Interrupt pending register 0x00000000
0x1f86654 Interrupt request register 0x00000000
0x1f86654 Interrupt force register 0x00000000
0x1f86656 *Watching ack. register 0xffffffff
0x1f86658 RTC counter register 0xffffffff
0x1f8665a RTC reload counter register 0xffffffff
0x1f8665c RTC reload counter enable register 0xffffffff
0x1f8665e RTC reload counter enable 0x000000ff
0x1f86684 GPT counter register 0x0000000f
0x1f86686 GPT reload counter register 0xffffffff
0x1f86688 GPT reload counter register 0xffffffff
0x1f8668a GPT reload counter enable register 0x000000ff
0x1f8668c GPT reload counter enable 0x000000ff
0x1f86690 System fault status register 0x00000078
0x1f86694 Falling address register 0x00000000
0x1f86698 Error and status register 0x00000000
0x1f8669c Text control register 0x00000000
tsim>

```

그림 12. Boot Loader Test #1

```

layright@fswlinux:~/KOMPSAT-5-Simulator> tsim> break 0xf7000
breakpoint 5 at 0x00000f7000
tsim> break 0x1240
breakpoint 6 at 0x000001240
tsim> cont
breakpoint 5
tsim> dis
Flight S/W ROM Entry
00000f704 a012000 or %l0,%l0
00000f706 20000000000000000000000000000000 sethi %hi(0xf7000),%l1
00000f708 02140014 or %l1,%l1,%l1
00000f709 2242000 st %l1,%l1,%l1
00000f710 1100003c sethi %hi(0xf7000),%l0
00000f712 98122208 or %l0,%l2,%l0
00000f714 40000000000000000000000000000000 call %x00000f59
00000f716 92102017 nov 23,%l1
00000f724 a2192001 nov 1,%l1
00000f728 e2192000 st %l1,%l1,%l1,%l1
00000f730 a2192000 sethi %hi(0xf7000),%l1
00000f732 a2192000 sethi %hi(0xf7000),%l1
00000f734 25000000000000000000000000000000 sethi %hi(0xf7000),%l2
00000f736 a41a0004 or %l2,%l0,%l2
00000f738 e244000 st %l2,%l1,%l1
tsim>

```

그림 13. Boot Loader Test #2

비행소프트웨어 ROM Entry에서는 GPIO<0>을 통해 Launch Abort Signal이 생성되는지를 IBMU Simulator PM32 interface를 통해서 확인하였으며 NVMMEM 관련된 register가 각 단계별로 제어가 되어 NVMMEM에 저장되어 있는 비행소프트웨어 이미지에 대한 CRC 계산을 수행 이후 비행소프트웨어 이미지를 RAM에 복사 후 구동되는지를 검증하였다. 그림 14에서는 NVMMEM B가 선택되었으나 CRC에러가 발생하여 NVMMEM A를 다시 선택하여 CRC 체크 이후 정상적으로 비행소프트웨어가 구동되는 예를 보여준다. 각 단계별로 수행결과를 PM32 시뮬레이터의 ERC32

UART B를 통해 Pseudo Serial Monitor에 출력하게 된다.

3.3 STB 기반의 Boot ROM Software 검증

시뮬레이터 기반의 부트 루م 소프트웨어에 대한 코드 레벨 검증이 완료 된 이후 STB(Software Test Bed)에서 실제 IBMU STBM 모델에 부트 루م 소프트웨어를 로딩 하여 검증을 수행하였다. 실제 IBMU STBM의 경우 PROM을 에뮬레이션 해주는 NetROM을 사용하여 부트 루م 소프트웨어를 로딩 하였으며 STB 테스트에서는 코드별로 검증이 불가능하기 때문에 GPIO<5> 입력에 따라 비행소프트웨어 모드와 모니터 모드로 천이되는지를 검증을 하였다.

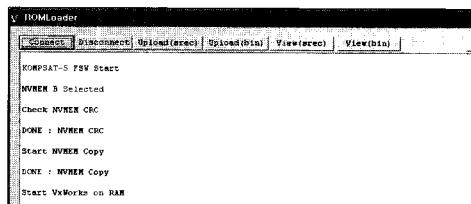


그림 15. Boot ROM Software 검증 결과

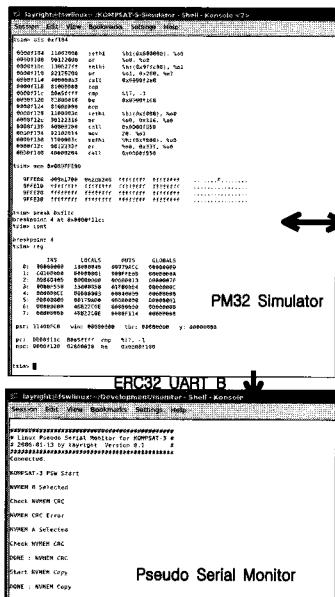


그림 14. Flight Software ROM Entry Test on Simulator

4. 결 론

본 논문에서는 차세대 위성의 부트 루م 소프트웨어의 상세 설계에 대해서 기술 하였으며 IBMU & PM32 시뮬레이터에서의 부트 루م 소프트웨어 검증방법에 대해서 기술하였다. IBMU STBM/EM/PQM에 실제 부트 루م 소프트웨어가 적용되어 사용되고 있으며, 지상에서 NVMEM reprogramming을 정상적으로 수행하고 있다. 추후 다른 위성 시스템 개발에서도 동일한 부트 루م 소프트웨어 architecture를 가질 수 있기 때문에 범용 적으로 재사용이 가능할 것으로 판단된다.

참 고 문 헌

- "TSC695F SPARC 32-bit Space Processor User Manual", Atmel Corp., 2003.
- "MCM ERC32 SC User's Manual", Astrium, 2004.
- "TSIM Simulator User's Manual Version 1.3.8", Gaisler Research, 2006.

