차세대 위성용 탑재소프트웨어 개발을 위한 고성능 탑재컴퓨터 ERC32 프로세서 소개

이재승[O] 최종욱 채동석 이종인 김학정
한국항공우주연구원
{jslee[O], jwchoi, dschae, jilee, hjkim}@kari.re.kr

Introduction to High-Performance ERC32 Processor for the Development of Flight Software

Jae-Seung Lee[O] Jong-Wook Choi, Dong-Seok Chae, Jong-In Lee, Hak-Jung Kim
Korea Aerospace Research Institute

요 약

국내에서 개발된 저궤도 관측위성에는 인텔계열의 프로세서가 사용되어 졌다. 인텔계열의 프로세서를 이용한 임베디드 시스템은 다양하고 상세한 기술문서들이 많이 제공될 수 있고 국내에도 관련된 기술 습득이 이루어져 있다는 장점이 있다. 그러나 프로세서에 대한 새로운 기술의 개발 및 적용이 단기간에 이루어지기 때문에 이를 이용한 시스템 개발에 계속적으로 개발비용 부담이 가중되고, 선진국의 첨단기술 유출 방지를 위한 여러 규제들로 인해 인텔 계열의 프로세서를 사용한 위성용 임베디드 시스템 개발에 걸림돌이 되고 있다.

이러한 문제점들을 해결하기 위해 차세대 위성에 사용 가능한 탑재컴퓨터에 대한 조사 및 기술분석을 수행하였으며, 유럽에서 자체적으로 개발하여 위성용 탑재컴퓨터로 사용하고 있는 MCM-ERC32가 차세대 위성 개발에 적합한 것으로 판단되었다. MCM-ERC32는 SPARC v.7을 기반으로 한 ERC32SC 프로세서를 이용한 탑재컴퓨터 보드로 향후 위성에 요구되는 다양한 기능들을 수행하기에 충분한 고성능 탑재컴퓨터이다.

국내에서는 MCM-ERC32를 이용한 개발 경험이 전무한 상황이며, 관련된 기술자료도 충분하지 않은 실정이다. 이에 따라 MCM-ERC32에 적합한 저궤도 위성용 탑재소프트웨어 개발을 위한 연구가 진행되고 있으며, 본 논문에서는 MCM-ERC32를 이용한 탑재소프트웨어 개발에 필요한 특징적인 ERC32 프로세서의 개념과 기능에 대해 소개하고자 한다.

1. 서론

유럽에서는 ESA(European Space Agency)의 지원 하에 자체적으로 개발한 MCM-ERC32를 위성용 고성능 탑재컴퓨터로 사용하고 있다. MCM-ERC32는 ERC32 Single-Chip, ASIC VASI(Very Advanced SPARC Interface), SRAM, DRAM 등으로 구성되어져 있으며, 이러한 모든 장치들을 하나의 보드에 집적하여 제작하므로 공간 및 비용절감이 가능하다.

향후 개발될 저궤도 관측위성에 MCM-ERC32를 사용하기 위한 개발기술에 대한 연구가 진행 중이며, 그 일환으로 MCM-ERC32에서 제공되는 기능들을 사용할 수 있도록 지원하는 BSP(Board Support Package) 및 탑재소프트웨어의 개발이 수행되고 있다.

ERC32(Embedded Real-time Computer 32-bit)는 MCM(Multi-Chip Module)의 핵심적인 부분을 구성하는 프로세서로 기존의 저궤도 관측위성에 사용되어왔던 인텔계열의 프로세서와는 전혀 다른 새로운 아키텍쳐를 가지고 있으며 RISC(Reduced Instruction Set Computer) 프로세서로서의 다양한 기능을 제공한다. 그러나 ERC32 프로세서는 유럽에서 독자적으로 개발되었으며, 국내에서는 사용 경험 및 관련 기술이 전무한 상황이다.

본 논문에서는 향후 위성용 고성능 탑재컴퓨터로 사용하게 될 MCM-ERC32에 활용 가능한 BSP 및 low-level 소프트웨어 개발 시 유의해야 하는 ERC32 프로세서의 특징적인 기능 및 개념에 대하여 소개한다.

먼저 2절에서는 ERC32의 전반적인 구성에 대하여 설명하고, 3절과 4절에서는 ERC32를 구성하는 가장 기본적인 정수연산 유닛인 TSC691E와 다른 프로세서와 구별되는 윈도우 레지스터의 기능에 대해 설명하도록 한다.

2. ERC32

ERC32 프로세서는 Rad Hard 32-bit SPARC Embedded Processor(TSC695F[1])로서 SPARC 아키텍쳐 V7[2,3]을 기본으로 하는 고집적, 고효율성을 가진 32-비트 RISC 내장형 프로세서이다. ERC32는 크게 정수 연산을 담당하는 IU(Integer Unit, TSC691E[4]), 실수 연산을 담당하는 FPU(Floating-Point Unit, TSC692E), 메모리 및 기능들의 제어를 담당하는 레지스터가 포함된

MEC(Memory Controller, TSC693E)의 크게 3개 유닛으로 구성되어 있다. 이 외에도 외부 인터페이스를 위한 DMA Arbiter를 포함하고 있다. 아래의 그림 1은 ERC32의 전체적은 구성을 나타낸다.

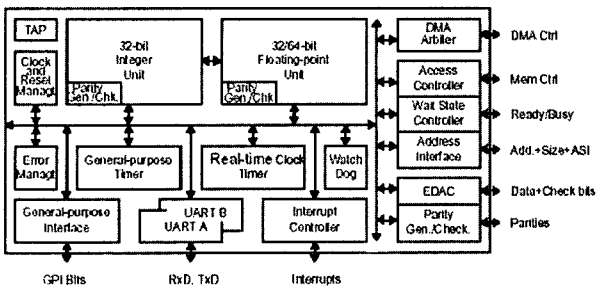

그림 1. ERC32 Block Diagram

실시간 운영체제를 이용한 임베디드 시스템을 위하여 ERC32는 워치독 타이머와 2개의 범용 타이머, 인터럽트 컨트롤러, 병렬/직렬 인터페이스 등을 제공하며, 오류 정정을 위해 내/외부 버스 인터페이스에 패러티 점검 기능과 외부 데이터 버스에 대한 EDAC(Error Detection and Correction) 기능을 지원한다. 또한 OCD(On-Chip Debugger) 및 JTAG과 같은 테스트 인터페이스도 자체적으로 제공한다.

이미 언급한 바와 같이 ERC32는 RISC 프로세서이므로 CISC 프로세서와 비교하여 다음과 같은 특징을 가지고 있다.

- Simple Instruction Set
- Same Length Instruction
- Reduced Memory Access
- Small Number of Addressing Mode
- 1 Machine-cycle Instruction
- Pipelining

ERC32에서는 상태점검 및 기능제어를 위하여 많은 레지스터들을 제공하고 있으며, 아래의 그림 2에 ERC32에서 제공하는 레지스터 모델을 정리하였다.

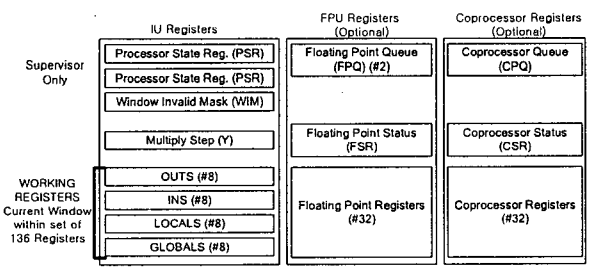

그림 2. ERC32 Register Model

실제 사용되는 ERC32 프로세서는 그림 2에서 보여주는 레지스터들 중 코프로세서용 레지스터는 지원하지 않는다.

3. ERC32 Integer Unit (TSC691E)

IU는 프로세서의 전반적인 동작을 제어하며 대표적인 기능은 다음과 같다.

- 정수연산 명령 수행
- load/store를 위한 메모리 연산 수행
- PC(Program Counter) 관리
- FPU 수행을 위한 명령 관리

2절에서 설명한 바와 같이 ERC32는 RISC 프로세서로서 pipeline 구조를 사용하며, 하나의 명령 수행이 fetch, decode, execute, write의 4단계로 병렬 수행되어 동시에 4개의 다른 명령이 수행되는 방식을 가지게 된다.

ERC32의 IU는 32-비트 크기의 제어 및 상태 레지스터와 136개의 범용 레지스터를 가지고 있으며, 각 레지스터의 기능 및 특징에 대해서는 참고문헌을 통하여 자세하게 확인할 수 있다.

4. Window Register

본 절에서는 ERC32에서 제공하는 레지스터 제어방식인 윈도우 레지스터의 기능에 대해서 설명한다.

IU의 136개 범용 레지스터는 8개의 전역 레지스터와 128개의 윈도우 레지스터로 나뉘어지며, 128개의 윈도우 레지스터는 다시 8개의 그룹으로 분류되고, 각 그룹은 ins, locals, outs 레지스터로 구성된다.

기본적인 SPARC 아키텍쳐에서는 최대 32개의 윈도우가 제공되지만 ERC32는 8개의 윈도우만을 제공한다. 운영체제가 선점하여 사용하는 1개를 고려하면 실제 사용가능한 윈도우는 7개뿐이다. 만약 서브 루틴의 호출 등이 많아서 7개 이상의 윈도우가 사용될 경우 window overflow/underflow가 발생하게 되므로 이를 고려하여 탑재소프트웨어 개발이 수행되어야 한다. SPARC 프로세서를 지원하는 상용의 실시간 운영체제로는 VxWorks 5.4가 있으며, VxWorks에서는 이러한 overflow/underflow를 스택을 이용하여 방지할 수 있는 기능이 제공되므로 유용하게 활용할 수 있다.

ERC32에서 레지스터 파일은 원형 스택 개념으로 구현되어 있으며, 다음의 그림 3에서 윈도우 #7에서 윈도우 #0까지 8개의 윈도우로 구성되어 있는 것을 알 수 있다.

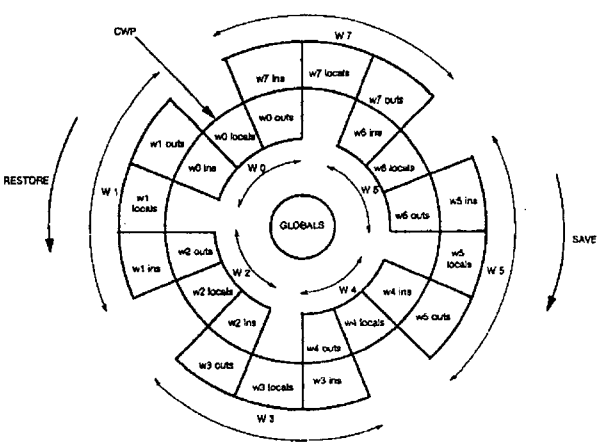

그림 3. Circular Stack of Overlapping Window

한 개의 윈도우는 24개의 윈도우 레지스터와 8개의 전

역 레지스터로 구성되며, 각 윈도우는 인접한 윈도우와 ins 및 outs 레지스터를 공유한다.

다음의 그림 4에서와 같이 CWP(Current Window Pointer)+1에 있는 outs 레지스터는 CWP의 ins 레지스터가 되고, CWP의 outs 레지스터는 CWP-1의 ins 레지스터가 되는 방식을 이용한다. 인접한 윈도우의 ins와 out, globals 레지스터가 공유되는 반면 locals 레지스터는 오직 해당 윈도우에서만 사용된다.

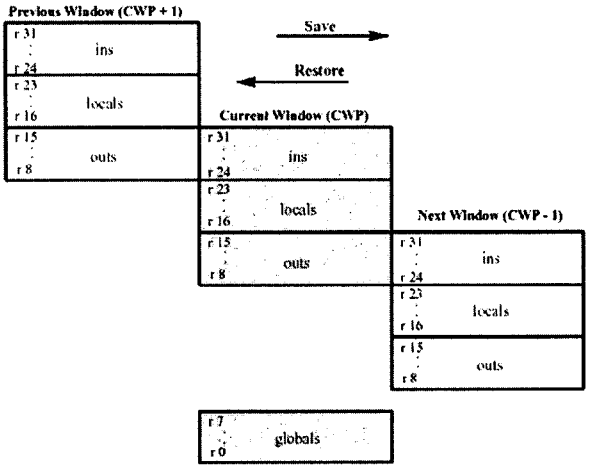

그림 4. Overlapping Windows

ERC32의 레지스터 윈도우 overlap은 프로시저의 호출과 리턴 시 인수들을 효율적으로 전송하는 기능을 제공한다. 만약 outs 및 globals 레지스터로 보낼 수 있는 인수보다 더 많은 인수를 호출되는 함수에 전송해야 한다면 메모리 스택을 이용하여 전송할 수 있다.

윈도우 레지스터의 outs 레지스터 #6은 스택 포인터(SP), 즉 사용하고 있는 메모리 스택의 주소가 저장되며, 스택 포인터는 호출된 프로시저에서 ins 레지스터 #6이 되어 프레임 포인터(FP)라는 이름으로 사용된다.

이상에서 설명한 윈도우 레지스터의 기능을 테스트하기 위하여 Gaisler Research Lab에서 개발된 LECCS 컴파일러와 ERC32 시뮬레이터인 TSIM Pro를 이용하였다. 다음의 그림 5~6에 태스크 스위칭에 따른 윈도우 레지스터의 기능에 대한 테스트 결과를 나타내었다.

그림 5는 test1 함수를 호출하기 전의 레지스터 상태를 보여준다. o0, o1, o2, o3 레지스터 값이 각각 1, 2, 3, 4이며 SP와 FP 사이의 영역을 덤프하여 1, 2, 3, 4 값이 존재하는 것을 확인할 수 있다. 또한 main 함수가 사용하는 내부 스택 영역은 SP(0x23ffd80)~FP(0x23ffe00)인 것을 확인할 수 있다.

그림 6은 main에서 test1을 호출하였을 때 레지스터의 상태를 보여준다. i0, i1, i2, i3의 값이 1, 2, 3, 4로 바뀐 것을 볼 수 있으며, 이것은 main 함수의 outs 레지스터가 test1의 ins 레지스터가 되었다는 것을 의미한다. 또한 CWP가 6에서 5로 변경되었으며, 변경된 SP와 FP 값을 통하여 test1이 사용하는 스택 영역을 알 수 있다.

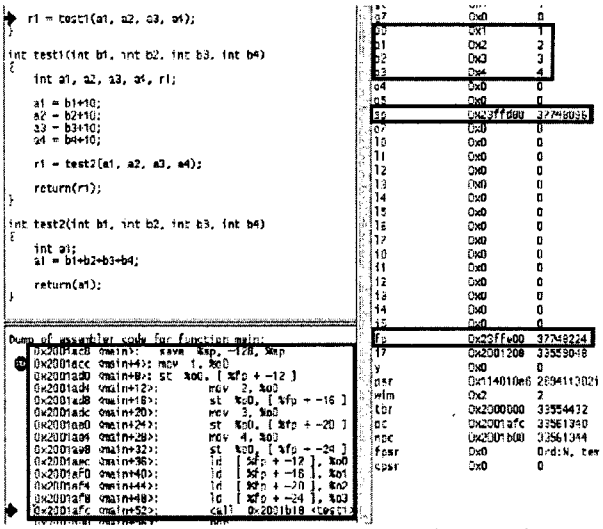

그림 5. Test for Window Register (Step #1)

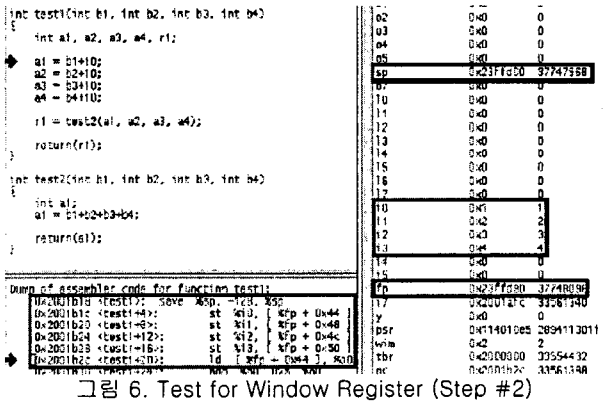

그림 6. Test for Window Register (Step #2)

5. 결론

본 논문에서는 향후 저궤도 관측위성 개발에 사용할 고성능 탑재컴퓨터인 ERC32의 기능에 대해 소개하였다.

추후 MCM-ERC32의 전체적인 기능분석 및 BSP 개발이 완료되면 실시간 운영체제와의 연동 개발 및 이를 이용한 위성탑재소프트웨어의 설계가 수행될 예정이다.

참고문헌

[1] " TSC695F SPARC 32-bit Space Processor User Manual" , Atmel, 2003

[2] " SPARC V7.0 Instruction Set for Embedded Real-time 32-bit Computer (ERC32) for SPACE Application" , Atmel, 1996

[3] " The SPARC Architecture Manual" , SPARC International Inc., 1992

[4] " TSC691E Integer Unit User' s Manual" , Atmel, 1996