

MCM-ERC32 에서의 위성탑재소프트웨어 개발을 위한 RTC(Real-Time Clock) 설계

이재승*, 박성우*, 김대영*, 이종인*, 김학정*

*한국항공우주연구원 위성전자그룹

e-mail : jslee@kari.re.kr

A Design of RTC(Real-Time Clock) on MCM-ERC32 for the Development of Flight Software

Jae-Seung Lee*, Seong Woo Park*, Day-Young Kim*, Jong-In Lee*, Hak-Jung Kim*

*Satellite Electronics Department, Korea Aerospace Research Institute

요 약

향후 국내에서 개발되는 저궤도 관측위성의 고성능 탑재컴퓨터로 유럽에서 자체적으로 개발하여 위성용으로 활용하고 있는 MCM-ERC32 를 사용할 예정이다. MCM-ERC32 는 크게 32-비트 ERC32SC 프로세서와 프로세서의 기능을 보완하고 추가적인 기능들을 제공하기 위해 제작된 ASIC 인 VASI(Very Advanced Sparc Interface), 그리고 메모리(SRAM, DRAM, EEPROM, etc.)로 구성되어 있다.

위성의 탑재소프트웨어를 설계 및 개발하는데 있어서 가장 기본적으로 요구되는 기능이 타이머이다. 탑재소프트웨어는 타이머를 통하여 태스크들의 관리와 스케줄링 등을 수행하게 된다. 위성과 같이 높은 정확도가 요구되는 실시간 임베디드 시스템에서는 타이머의 구현이 매우 중요하다. ERC32SC 프로세서 자체에서도 RTC, GPT(General Purpose Timer), WDT(Watchdog Timer)와 같은 기본적인 타이머 기능을 제공하지만 VASI 에서도 클락과 사이클이라는 개념을 이용한 RTC 를 제공한다. 어느 타이머를 사용하는가는 전적으로 개발자의 선택이다. ERC32SC 프로세서에서 제공하는 타이머는 상용의 임베디드 시스템에서 제공하는 기능과 동일하다.

본 논문에서는 위성탑재소프트웨어 개발에 필요한 RTC 를 설계하기 위한 MCM-ERC32 에서 제공하는 VASI RTC 의 구조와 기능에 대하여 소개하고자 한다.

1. Introduction

저궤도 관측위성용 탑재컴퓨터로 기존의 인텔 계열 프로세서를 대신하여 유럽에서 개발된 MCM-ERC32 를 사용하기 위한 연구가 계속적으로 수행되고 있다. 이와 연계하여 MCM-ERC32[1]를 사용한 임베디드 시스템 구현을 위한 탑재소프트웨어[2]의 설계와 개발도 함께 진행되고 있다.

타겟 프로세서에서 임베디드 소프트웨어를 구현하기 위해 요구되는 기능 중 기본적인 것이 타이머이다. 탑재소프트웨어는 타이머를 이용하여 태스크의 생성 및 스케줄링 등 소프트웨어 실행을 관리하기 위해 사용한다.

ERC32SC 프로세서[3,4]에서도 RTC, GPT, WDT 등의

일반적인 타이머 기능들을 제공하고 있지만, VASI 에서는 향상된 기능들을 추가한 VASI RTC(Cycle, Slot, Dummy Slot)를 제공한다. VASI RTC 는 관련된 레지스터를 이용하여 편리하게 제어가 가능하며 dummy slot 을 이용하여 보다 효율적으로 RTC 기능을 구현할 수 있는 방법을 제공한다.

본 논문에서는 기존의 위성탑재컴퓨터에서 제공되었던 RTC 기능과 달라진 MCM-ERC32 에서만 제공되는 VASI RTC 의 특별한 기능에 대하여 설명하고자 한다.

먼저 2 절에서는 VASI RTC 의 개략적인 소개와 특징에 대해 설명을 하고, 3 절에서는 VASI RTC 기능을 지원하는 VASI 레지스터에 대하여 설명하도록 한다. 그리고 4 절에서는 VASI RTC 의 기능을 이용하여 위

성탐재소프트웨어를 구현하기 위한 방안과 MCM-ERC32 를 이용한 테스트 결과를 알아보도록 한다.

2. RTC Unit of VASI

VASI RTC 는 다음과 같은 3 가지 타이머 컨트롤러를 제공한다.

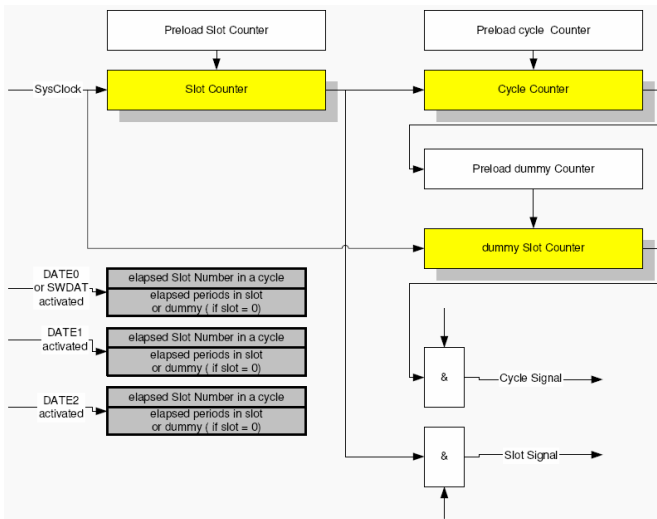
- Slot Counter : 시스템 클럭을 사용하여 카운팅
- Cycle Counter : slot counter 가 0 이 될 때마다 클럭킹
- Dummy Slot Counter : 시스템 클럭에 의해 카운팅되며 0 이 아닌 값을 가지고 있을 경우에만 동작

한 slot 의 최대 주기는 시스템 클럭이 25MHz 일 경우 171 초까지 될 수 있다. 그리고 각각의 카운터 값이 초기 설정값에서 감소하여 0 이 되면 해당 인터럽트가 발생하도록 설정할 수 있다.

또한 다음과 같은 datation 이벤트 기능을 이용하여 현재의 RTC 카운터 값을 저장할 수 있다.

- H/W datation : MCM-ERC32 는 현재의 cycle, slot, dummy slot 의 카운터 값을 저장하도록 하는 외부 입력 신호와 해당 레지스터를 각각 3 개씩 지원한다.
- S/W datation : H/W datation 에 사용되는 3 개 중 하나는 소프트웨어에서 레지스터를 이용하여 제어할 수 있다.

RTC 기능을 보여주는 다이어그램을 아래 그림 1 에 나타내었다.



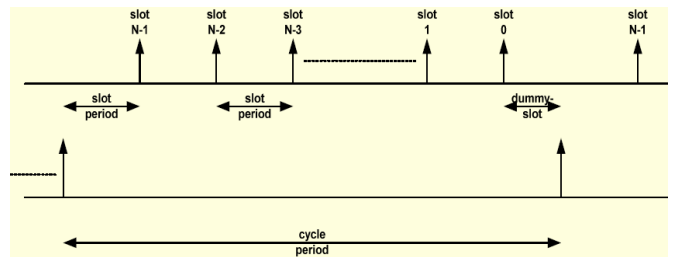
(그림 1) RTC Functional Diagram

한 slot 을 이루는 전체 시스템 클럭의 수에 해당하는 시간을 slot period 라고 하며 slot 카운터에 시스템 클럭의 주기를 곱하여 계산된다. Dummy slot period 도 같은 방법으로 계산되어 진다.

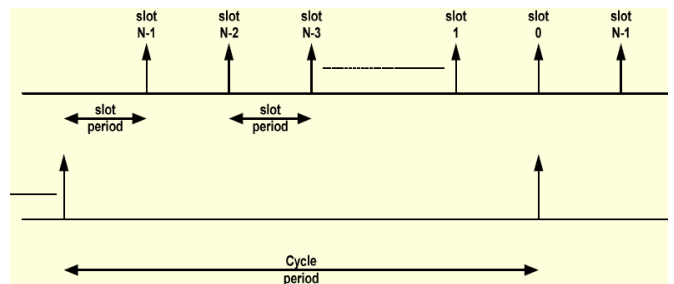
Cycle period 는 한 cycle 을 구성하는 전체 slot period

와 dummy slot period 의 합을 의미한다. 만약 dummy slot 카운터가 0 이라면, cycle period 는 전체 slot 의 수에 해당하며, cycle 이벤트가 발생할 때 slot 이벤트도 동시에 발생하게 된다. 반대로 dummy slot 카운터가 0 이 아닐 경우에는 slot 및 cycle 카운터가 0 이 되더라도 slot 이벤트만 발생하며, dummy slot 카운터가 0 이 되어야 cycle 이벤트가 발생하게 된다. 즉, cycle period 가 전체 slot 의 수와 동일하지 않으며 dummy slot 이 카운팅되는 동안에 slot 카운터는 정지된다.

아래의 그림 2 와 그림 3 은 각각 dummy slot 이 0 이 아닐 경우와 0 일 경우에 대해 cycle 및 slot 주기와 이벤트를 예를 들어 설명한 것이다.



(그림 2) Slot & Cycle Period with Dummy Slot > 0



(그림 3) Slot & Cycle Period with Dummy Slot = 0

Dummy slot 은 cycle period 가 정확하지 않을 경우 이를 보정하기 위한 용도로 사용될 수 있다.

3. RTC Registers Description

MCM-ERC32 의 VASI 에서는 cycle 과 slot period 를 설정하기 위해 다음과 같은 3 개의 레지스터를 제공한다.

- Cycle Counter Register
- Slot Counter Register
- Dummy Slot Counter Register

위의 레지스터 값이 설정되면 slot 및 dummy slot 카운터는 시스템 클럭에 따라 감소하고, cycle 카운터는 slot 이벤트에 의해 감소하게 된다. 각 카운터 값이 0 이 되면 해당하는 이벤트를 발생시키고 카운터 값은 설정한 값으로 리셋된다.

현재 카운팅되고 있는 값을 사용하려고 할 경우에는 위의 레지스터를 이용할 수 없다. 위의 각 카운터 레지스터 값을 읽으면 초기에 설정한 값이 읽혀질 뿐

현재 변하고 있는 카운터 값은 읽혀지지 않는다. VASI에서는 현재 카운팅되고 있는 값을 저장하여 사용할 수 있도록 3 개의 Software Datation Register (DATE0, 1, 2)를 제공한다.

다음의 그림 4 는 VASI 에서 제공하는 datation 레지스터를 나타낸다.

Signal activation	Current Slot register	Current slot/dummy slot register
DATE0 / SWDATReg Access	CSNReg 0 (0x14000050)	CSDTReg 0 (0x14000054)
DATE 1	CSNReg 1 (0x14000058)	CSDTReg 1 (0x1400005C)
DATE2	CSNReg 2 (0x14000060)	CSDTReg 2 (0x14000064)

(그림 4) Event Datation Registers

이 레지스터들은 외부에서 하드웨어 입력신호를 주면 현재의 카운터 값을 해당 레지스터에 저장한다. 저장되는 카운터 값은 slot number 와 slot / dummy slot 카운터 값이며 이를 저장하기 위한 다음의 레지스터가 각각 3 개씩 제공된다.

- Current Slot Number Register
- Current Slot/Dummy Slot Time Register

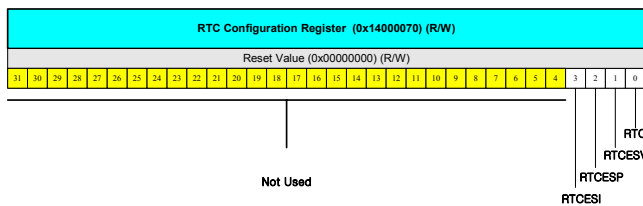
두 번째 레지스터는 현재 dummy slot 이 진행되고 있을 경우에는 dummy slot 카운터 값이 저장되고, 아닐 경우에는 slot 카운터 값이 저장된다. 따라서 dummy slot 카운터가 0 으로 설정되어 dummy slot 이 사용되지 않는 경우에는 두 번째 레지스터는 항상 slot 카운터 값을 저장하게 된다.

3 개의 datation 레지스터 중 하나는 외부의 하드웨어 입력신호 외에도 다음의 레지스터를 설정하여 소프트웨어적인 방법으로도 사용이 가능하다.

- Software Datation Register

위의 레지스터에 임의의 값을 쓰면(write access) 외부에서 하드웨어 신호가 입력된 것과 같은 작용을 하여 해당 레지스터에 카운터 값들이 저장된다. 이러한 기능은 소프트웨어 내부적으로 태스크의 시작시간을 확인할 필요가 있거나 타이밍 분석이 필요한 태스크에서 활용할 수 있다.

위와 같은 기능들을 제공하는 RTC 타이머 기능을 enable/disable 하고 리셋을 하거나 기타 해당기능들을 설정하기 위해 RTC Configuration Register 와 RTC Reset Register 가 존재하며 각각의 레지스터의 구성을 그림 5 와 그림 6 에 나타내었다.

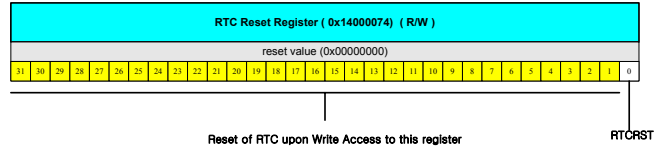


(그림 5) RTC Configuration Register

그림 5 에 표시된 레지스터의 구성에서 알 수 있듯이 해당 비트를 이용하여 다음과 같은 설정을 할 수

있다.

- RTCV : VASI RTC 의 실행 / 정지
- RTCESV : RTC 의 외부신호 동기화 여부
- RTCESP : 외부신호에 동기화할 때 rising/falling edge 선택
- RTCESI : 외부신호 동기에 의해 cycle 인터럽트의 발생 여부



(그림 6) RTC Reset Register

그림 6 의 레지스터는 0 이 아닌 다른 값을 쓰면 RTC 가 리셋되며 마지막 비트는 외부신호의 rising 및 falling edge 에서 토글되어 외부신호의 입력여부를 확인할 수 있는 기능을 제공한다.

VASI에서는 정확한 시간측정이 요구되는 시험이나 검증에 사용할 수 있도록 roll-over 타이머 기능을 제공하는 Current Clock Time Register 가 있다. MCM-ERC32 가 처음 동작을 시작하거나 리셋될 경우 이 레지스터는 0 으로 초기화되며 시스템 클락에 따라 카운팅되며, 0xFFFFFFFF 까지 증가한 후 다시 0 으로 초기화되어 다시 카운팅을 반복한다. 이 레지스터에 의해 발생하는 인터럽트는 없으며 단순히 카운팅만 계속적으로 수행된다. 이러한 기능은 소프트웨어 개발자에게 운영체제의 타이머를 사용하지 않고 두 이벤트 사이의 타이밍이나 주기를 측정하는데 용이한 해결방안을 제공한다.

RTC 의 cycle 및 slot 에 의해 발생한 이벤트는 ERC32SC 의 외부 인터럽트와 연결하여 해당 인터럽트를 소프트웨어에서 사용할 수 있으며, 또한 레지스터 설정을 통하여 마스크시킬 수도 있다.

그 외에도 MCM-ERC32 에서 제공하는 UART, 1553B Interface, 1355 Interface 등의 해당 레지스터를 설정하여 해당 태스크를 cycle 또는 slot 이벤트와 동기화할 수 있는 기능도 제공한다.

4. RTC Tests on MCM-ERC32

향후 저궤도 관측위성에 탑재될 MCM-ERC32 의 개발모델이 개발되었으며, 개발모델을 이용하여 VASI RTC 가 제공하는 기능들에 대한 테스트를 수행하였다.

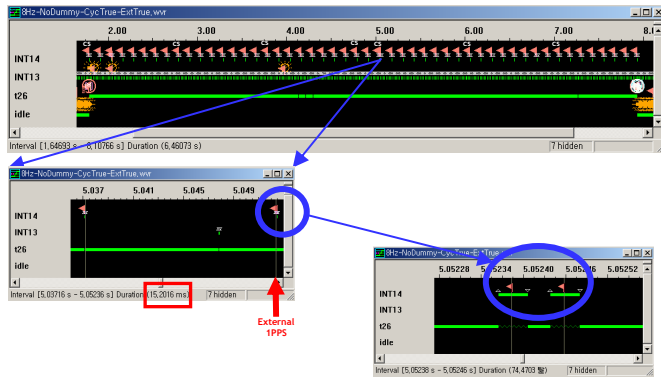
특히 cycle 과 slot 을 이용하여 탑재소프트웨어의 태스크 스케줄링을 구현하기 위한 테스트를 위주로 시험이 수행되었다. 또한 정확한 타이밍을 유지하기 위하여 GPS sync 신호를 이용한 외부신호 동기화 방안을 적용하기 위하여 다양한 설정에 따른 결과 분석도 함께 수행되었다.

본 절에서는 향후 저궤도 관측위성용 탑재소프트웨어 설계에 활용하기 위해 수행된 dummy slot 과 외부

신호 동기화에 따른 cycle 및 slot 인터럽트 발생 결과를 소개하도록 한다.

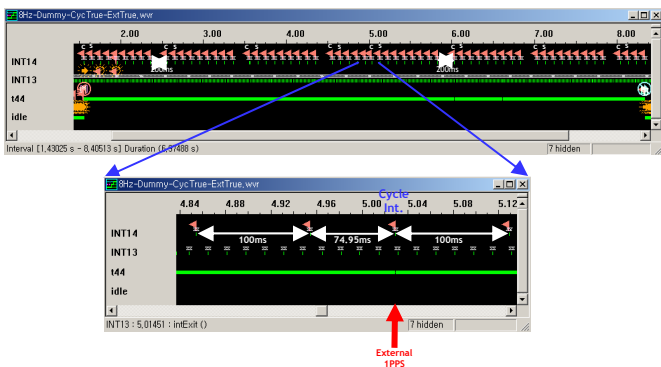
실험을 수행하기 위하여 NetROM 인터페이스를 이용하여 ROM 영역의 에뮬레이션과 MCM-ERC32 보드와의 이더넷 인터페이스를 구축하였다. 실시간 운영체제로는 VxWorks 를 사용하였으며 cycle/slot 인터럽트의 타이밍 결과를 분석하기 위한 개발 툴로는 Tornado 를 사용하였다. 아래의 그림 7 과 그림 8 은 Tornado 에서 제공하는 WindView 툴을 이용하여 cycle 및 slot 인터럽트를 시간축으로 표시하였다.

본 테스트에서 하나의 cycle 은 8 개의 slot 으로 구성되며 그림 7 에는 slot period 는 125msec 이고 dummy slot 이 없을 경우를, 그림 8 에는 slot period 가 100msec 이고 dummy slot 을 사용하였을 경우의 결과를 나타내었다. VASI 에서 제공하는 레지스터 설정을 통하여 24 개의 slot 인터럽트가 발생한 후에 외부 동기신호에 의해 cycle 인터럽트가 동기되도록 설정하였다. 외부 동기신호는 function generator 를 사용하여 GPS 1 Hz 를 모사하였다.



(그림 7) Synchronization Test with External 1 Hz (1)

그림 7 에서 INT14 가 cycle/slot 인터럽트를 의미하며 t26 이 탑재소프트웨어의 스케줄링을 담당하는 태스크를 의미한다. 그림의 결과를 통하여 125msec 마다 slot 인터럽트가 발생하며 8 개의 slot 마다 cycle 인터럽트가 발생하는 것을 확인할 수 있다. 또한 24 개의 slot 인터럽트 이후 외부 동기신호에 의한 동기화가 enable 되어 외부 동기신호가 들어오면 다시 cycle 과 slot 이 초기화되어 수행되는 것을 알 수 있다.



(그림 8) Synchronization Test with External 1 Hz (2)

위의 그림 8 은 dummy slot 이 존재하기 때문에 그림 7 과는 달리 cycle 의 시작 시에 cycle 인터럽트만 발생하는 것을 볼 수 있다.

그림 7 과 그림 8 의 경우 모두 cycle period 는 1 초를 기준으로 하며, 외부 동기신호에 의해 한번 동기화가 수행되면 이후 계속적으로 외부동기 신호와 일치하여 cycle/slot 인터럽트가 정상적으로 발생하는 것을 확인할 수 있다.

5. Conclusions

향후 개발될 저궤도 관측위성의 탑재컴퓨터로 유럽에서 개발된 MCM-ERC32 가 사용될 예정이며, 새로운 탑재컴퓨터에서 수행될 탑재소프트웨어의 설계가 진행되고 있다.

본 논문에서는 MCM-ERC32 가 제공하는 VASI RTC 를 탑재소프트웨어의 설계에 활용하기 위한 VASI RTC 의 기능과 특징, 그리고 시험결과를 소개하였다.

시험결과를 통하여 VASI RTC 를 이용한 스케줄링의 예를 제시하였으며, 외부동기화 방안을 적용할 경우 한번 동기화가 이루어지면 이후에는 동기가 정확하게 일치하여 더 이상 cycle 인터럽트가 흔들리는 현상이 발생하지 않는 것을 확인하였다. 또한 동기화 이후 약간의 오차가 발생하더라도 dummy slot 을 이용하여 오차에 의한 영향을 제거할 수 있을 것이다.

그러나 동기화 시점에서 그 이전의 cycle 이나 slot 주기를 버리고 새롭게 cycle/slot 이 시작되기 때문에 수행되고 있던 태스크들이 중간에 중단되거나 데이터가 손실되는 현상은 실시간 임베디드 시스템에서 고려되어야 할 것이다.

추후 시스템 레벨에서 이러한 사항들에 대한 대책이나 설계가 완료되면 그에 따른 탑재소프트웨어의 설계 및 개발이 계속적으로 진행될 예정이다.

참고문헌

- [1] “MCM ERC32SC User’s Manual”, ASTRIUM, 2004
- [2] “Description of MCM Software Low-Level Layers”, ASTRIUM, 2002
- [3] “TSC695F SPARC 32-bit Space Processor User Manual”, Atmel, 2003
- [4] “저궤도 위성용 탑재소프트웨어 개발을 위한 ERC32 프로세서 소개”, 한국정보처리학회 춘계학술발표논문집(하), pp 1553-1556, 2005