

저궤도 위성용 탑재소프트웨어 개발을 위한 MCM-ERC32의 VASI UART 기능 소개

이재승*, 최종욱*, 원영진*, 이종인*
*한국항공우주연구원 위성전자그룹
e-mail : jslee@kari.re.kr

An Introduction to VASI UART of MCM-ERC32 to Develop Flight Software for LEO Satellites

Jae-Seung Lee*, Jong-Wook Choi*, Young-Jin Won*, Jong-In Lee*
*Satellite Electronics Department, Korea Aerospace Research Institute

요 약

고신뢰도가 요구되는 위성용 탑재소프트웨어를 개발하기 위해서는 소프트웨어 처리기반으로 고성능의 탑재컴퓨터가 요구된다. 향후 개발될 위성을 위한 고성능 탑재컴퓨터로는 유럽에서 개발되어 사용되고 있는 MCM-ERC32를 채용할 예정이다.

ESA(European Space Agency)의 지원 하에 개발된 MCM-ERC32는 32-비트의 ERC32SC 프로세서, 부가적인 기능을 제공하는 ASIC인 VASI(Very Advanced Sparc Interface), 그리고 메모리(SRAM, DRAM, EEPROM, etc.)로 구성되어 있다. MCM-ERC32에는 ERC32 프로세서에서 제공되는 2개의 UART(A/B)와 VASI에서 제공하는 4개의 UART(0/1/2/3), 총 6개의 시리얼 인터페이스가 있다. ERC32에서 제공하는 시리얼 인터페이스는 8-비트 모드만 지원되며 전송속도에도 제한이 있기 때문에 탑재소프트웨어의 업로드 및 디버깅용으로 활용될 예정이며, 탑재체 간의 인터페이스로는 VASI에서 제공하는 시리얼 인터페이스를 사용할 예정이다.

VASI에서 제공하는 UART는 MCM-ERC32에 적합하도록 개발되어 일반적인 임베디드 시스템의 시리얼 인터페이스와는 구별되는 송수신 방법 및 기능을 제공한다. 본 논문에서는 이러한 VASI UART의 구성 및 특징과 기능들에 대하여 설명하도록 한다.

1. Introduction

향후 개발되는 저궤도 위성에는 더 많은 기능들과 신기술의 적용이 요구된다. 이러한 다양한 기능들과 신기술을 효율적으로 적용하기 위해서는 고성능의 위성용 탑재컴퓨터가 요구되며, 이를 위해 유럽에서는 ESA의 지원 하에 자체적으로 제작한 MCM-ERC32[1] 프로세서를 사용하고 있다. 국내에서 개발되는 저궤도 위성의 임무 수행을 위한 소프트웨어 처리기반으로 MCM-ERC32가 사용될 예정이며, 이를 기반으로 한 탑재소프트웨어의 개발이 진행되고 있다.

MCM-ERC32는 ERC32SC[2,3] 프로세서, VASI, 메모리로 구성되어 있으며, 1355, 1553B, UART 등 다양한 외부 인터페이스 기능을 지원한다. MCM-ERC32에서는 총 6개의 시리얼 인터페이스를 제공하는데,

ERC32SC 프로세서에 제공하는 시리얼 인터페이스는 8-비트 전송모드만이 가능하며 전송속도도 낮게 제한되어 있어 위성의 탑재체 인터페이스에 적용하기는 어렵다. 따라서 탑재컴퓨터와 외부 탑재체 간의 시리얼 인터페이스로는 VASI에서 제공하는 4개의 UART를 활용할 예정이다.

VASI UART는 8-, 16-, 32-비트의 다양한 전송모드를 제공하며 데이터 송수신을 위해 DRAM 영역을 할당하여 사용한다. MCM-ERC32에서는 VASI의 레지스터를 이용하여 VASI UART 인터페이스의 초기화 및 전송 기능 등을 제어할 수 있도록 되어 있어 기존의 시리얼 인터페이스 기능과는 다른 독특한 구조를 가지고 있다.

본 논문에서는 위성의 탑재컴퓨터와 외부 탑재체 사이의 시리얼 인터페이스로 활용하기 위한 VASI

UART의 구성 및 특징에 대하여 소개하도록 한다.

2. VASI Serial Link Interface

VASI에서는 2개의 양방향 동기식 시리얼 인터페이스와 2개의 양방향 비동기식 시리얼 인터페이스를 제공하며 모두 115,000 bauds 까지 지원된다. 2개의 동기식 시리얼 인터페이스는 비동기식 모드로도 사용이 가능하며 현재 탑재소프트웨어 운영개념에서 동기식 모드는 사용되지 않을 예정이다.

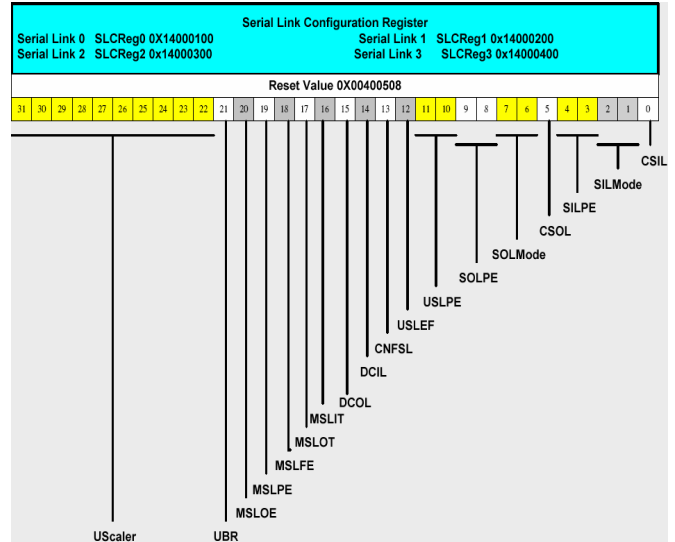
데이터 전송은 DMA(Direct Memory Access)에 의해 제어되며, 각각의 시리얼 링크에 대해 ERC32SC 프로세서가 DAM 레지스터를 프로그램한다. 데이터가 DRAM에 저장될 때마다 Serial Link Receive Current Buffer Address 레지스터가 증가되고 탑재소프트웨어에 의해 정의된 수만큼 데이터를 전송받으면 DMA는 Serial Link Receive DMA Current Number of Word 레지스터를 리셋한다. 데이터 전송이 완료되면 해당 DMA 채널의 워드 수는 '0'이 되며 Interrupt Pending 레지스터의 해당 인터럽트 플래그를 셋하게 된다. 해당 인터럽트 플래그는 데이터의 마지막 비트가 전송되면 셋되므로 만약 CRC가 있다면 CRC 전송 전에 셋된다.

시리얼 인터페이스의 전송속도, 전송모드, 그리고 송수신 기능과 관련한 VASI UART의 모든 기능들은 VASI 레지스터에 의해 제어된다. 다음의 그림 1은 VASI UART를 제어하기 위해 제공되는 VASI 레지스터를 나타내며 4개의 UART에 대한 각각의 레지스터가 별도로 존재한다.

Serial Link 0 Configuration Register (SLCReg0)	14 000100
Serial Link 0 Nominal Interrupt Register (SLNIReg0)	14 000104
Serial Link 0 Error Interrupt Register (SLEIReg0)	14 000108
Serial Link 0 Receive DMA reserved area Base Address Register (SLRBAReg0)	14 00010C
Serial Link 0 Receive DMA reserved area End Address Register (SLREAReg0)	14 000110
Serial Link 0 Receive DMA Buffer Size Register (SLRBSReg0)	14 000114
Serial Link 0 Receive DMA Current Buffer Address Register (SLRCBReg0)	14 000118
Serial Link 0 Receive DMA Current Number of Words Register (SLRCNReg0)	14 00011C
Serial Link 0 Transmit DMA Start Address Register (SLTSAReg0)	14 000120
Serial Link 0 Transmit DMA Number of Words Register (SLTNBReg0)	14 000124

(그림 1) VASI Register List for UART #0

그림 1의 UART 관련 레지스터 중 시리얼 인터페이스의 기본적인 설정을 관리하는 SLCReg 레지스터의 비트별 정보를 그림 2에 나타내었으며 각 비트별 설정방법 및 기능을 아래에 설명하였다. SLCReg 레지스터를 제외한 각 레지스터의 자세한 기능에 대해서는 참고문헌을 통해 확인할 수 있다.



(그림 2) Serial Link Configuration Register

- CSOL/CSIL : DMA 송수신을 enable/disable (1/0)
- SOLMode /SILMode : 시리얼 송수신 모드 설정
00 : disable
01 : 8-비트 모드
10 : 16-비트 모드
11 : 32-비트 모드
- USLPE : even/odd parity 설정 (10/01)
- USLEF : 정지비트 설정
- CNFSL : 동기식/비동기식 설정 (1/0)
- MSLOT/MSLIT : '0'으로 설정하면 데이터 송수신이 끝났을 때 인터럽트를 발생
- MSLFE/MSLPE/MSLOE : '0'으로 설정하면 해당 에러 발생 시 인터럽트를 발생
- UBR/USclaer : 다음의 식을 이용하여 전송속도 설정

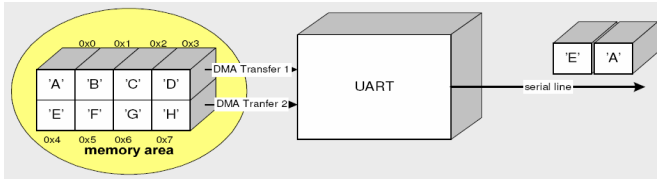
$$\text{Scaler} = \frac{\text{Clock Frequency}}{(16 * \text{Baud Rate} * (2 - \text{UBR}))} - 1$$

비동기식 시리얼 인터페이스의 전송모드는 8-, 16-, 32-비트의 3가지 모드가 있으며 각각의 전송모드에 따른 특징은 다음과 같다.

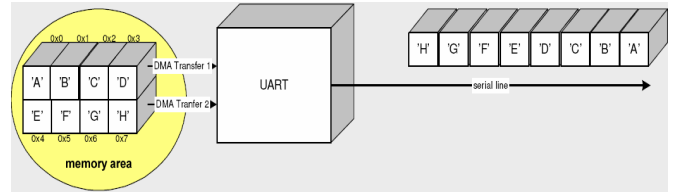
① 8-비트 모드

DMA는 메모리에서 32-비트 데이터를 읽고 UART는 최하위 주소의 1 바이트만 전송한다. 전송된 데이터는 Big-Endian의 경우 32-비트 워드의 최상위 바이트에 저장된다. 즉 1 바이트가 전송될 때마다 데이터는 메모리에 32-비트 워드로 저장되고, 저장된 32-비트 워드에서 최상위 바이트만이 실제 데이터로서 의미가 있다.

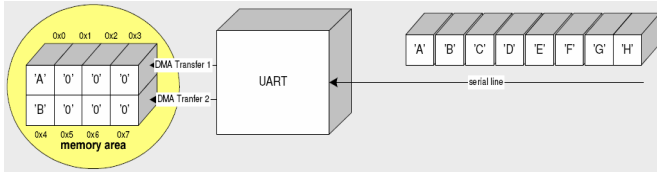
다음의 그림 3 과 그림 4는 VASI UART의 8-비트 모드에서 데이터 송수신을 보여준다.



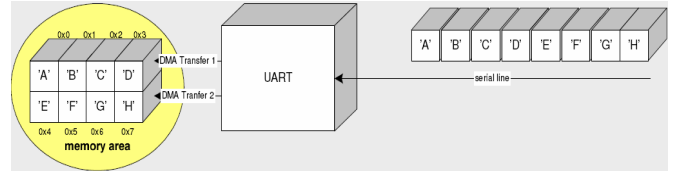
(그림 3) Data Transmission in 8-bit mode



(그림 7) Data Transmission in 32-bit mode



(그림 4) Data Reception in 8-bit mode

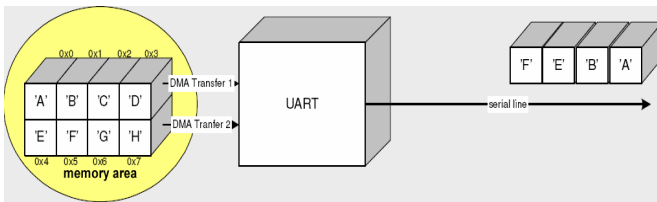


(그림 8) Data Reception in 32-bit mode

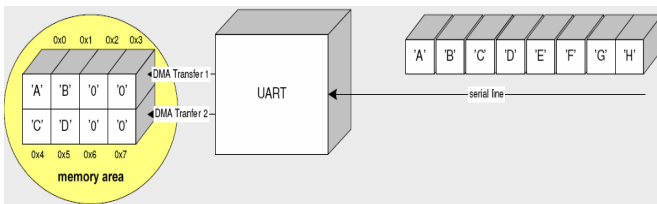
② 16-비트 모드

DMA 는 메모리에서 32-비트 워드를 읽어서 Big-Endian 의 경우 상위 16-비트 만을 전송한다. 전송된 데이터는 메모리의 32-비트 워드에 저장되며 상위 2 바이트만이 의미를 가지는 데이터이다.

다음의 그림 5 와 그림 6 은 16-비트 모드에서의 데이터 송수신을 보여준다.



(그림 5) Data Transmission in 16-bit mode



(그림 6) Data Reception in 16-bit mode

③ 32-비트 모드

MCM-ERC32 에서 DRAM 액세스는 DMA 에 의해 32-비트로만 이루어지기 때문에 VASI 에서는 ①, ②와 같은 모드도 지원되지만 데이터 전송에 특별한 주의가 요구된다. 이에 반해 32-비트 모드는 메모리에서 32-비트 워드를 읽어서 그대로 전송하며 전송된 데이터는 수신측 메모리에 같은 크기의 32-비트 워드에 저장된다.

다음의 그림 7 과 그림 8 은 32-비트 모드에서의 데이터 송수신을 보여준다.

3. VASI DMA Configuration

VASI 의 시리얼 링크는 프로세서와의 공유 메모리로 DRAM 을 사용하며, 워드 전송은 다음의 레지스터에 의해 정의되는 DMA 가 관리하며 이에 대한 DRAM 의 메모리 구조를 그림 9 에 나타내었다.

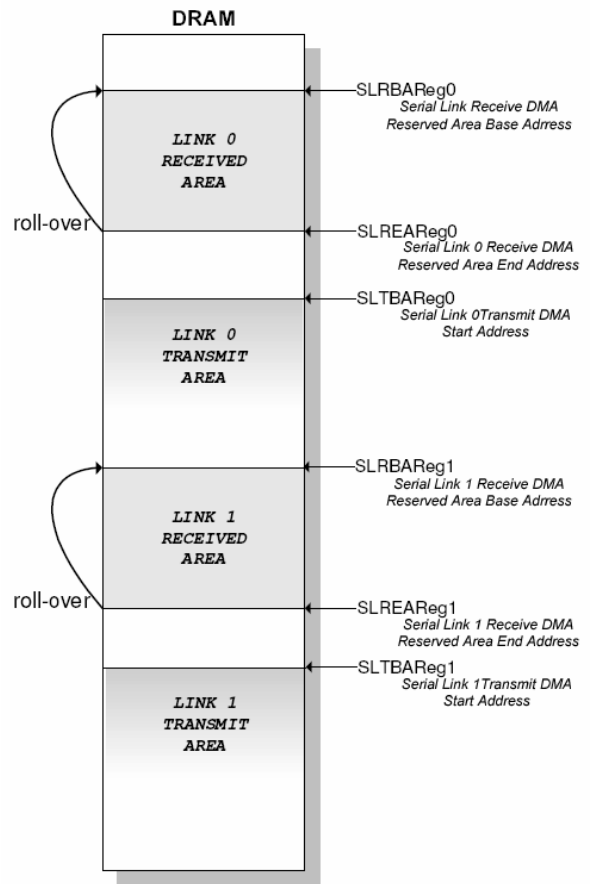


그림 9 Memory Design on DRAM

- SLRBAReg (Serial Link Receive DMA reserved area Base Address Register) : 전송받은 데이터가 저장될 영역의 시작 주소를 설정

- SLREAReg (Serial Link Receive DMA reserved area End Address Register) : 전송받은 데이터가 저장될 영역의 끝 주소를 설정
- SLTSAReg (Serial Link Transmit DMA Source Address Register) : 전송할 데이터가 저장된 영역의 주소를 설정. 하드웨어는 이 주소에서부터 전송할 워드 수만큼의 데이터를 전송.

위의 레지스터들은 VASI 에서 제공하는 4 개의 UART 채널에 대해 각각 별도로 존재한다.

DMA 의 DRAM 에 대한 읽기 및 쓰기 액세스는 워드 단위로 이루어진다. 선택된 모드에 관계없이 데이터 워드의 최상위 바이트는 DRAM 32-비트 주소의 최상위 바이트에 열라인되어야 한다. 데이터 워드의 크기는 앞의 그림 2 에서 설명한 SLCReg 레지스터의 비트 설정에 따라 결정된다.

실제 탑재소프트웨어가 구동되는 메모리(SRAM)에 저장된 데이터 프레임의 구조와 각 전송 모드별 DRAM 에 저장되는 전송 데이터 영역의 구조를 아래의 그림 10 에서 그림 13 에 나타내었다.

MSB		LSB		Address
00	01	02	03	
'A'	'B'	'C'	'D'	0
'E'	'F'	'G'	'H'	+4

(그림 10) Original Data Frame Contents (SRAM)

실제 탑재소프트웨어가 구동되고 있는 SRAM 에 저장되어 있는 그림 10 과 같은 데이터를 VASI UART 를 이용하여 8-, 16-, 32-비트 모드로 전송하고자 할 경우 해당 데이터가 DRAM 의 공유 메모리에 다음의 그림과 같이 할당되어야 한다.

MSB		LSB		Address
00	01	02	03	
'A'	X	X	X	0
'B'	X	X	X	+4
'H'	X	X	X	+28

(그림 11) Transmit Area Contents of DRAM (8-bit)

MSB		LSB		Address
00	01	02	03	
'A'	'B'	X	X	0
'C'	'D'	X	X	+4
'E'	'F'	X	X	
'G'	'H'	X	X	+12

(그림 12) Transmit Area Contents of DRAM (16-bit)

MSB		LSB		Address
00	01	02	03	
'A'	'B'	'C'	'D'	0
'E'	'F'	'G'	'H'	+4

(그림 13) Transmit Area Contents of DRAM (32-bit)

4. Data Protection of VASI UART

그림 2 의 SLCReg 레지스터의 비트를 설정하여 다음과 같은 보호모드를 시리얼 링크에 적용할 수 있다.

- Parity : even / odd
- CRC : 다항식 $X^{16}+X^{12}+X^5+1$ 에 의해 생성된 16 비트 CRC

데이터 수신 시 CRC 는 DRAM 에 쓰여지지 않으며, CRC 에러 인터럽트는 레지스터에 정의된 데이터 버퍼 크기만큼 수신되었을 때 활성화 된다.

만약 에러가 발생하면 데이터는 DRAM 에 저장되며 해당 시리얼 에러/인터럽트 관련 레지스터가 업데이트 된다. 그리고 VASI 의 인터럽트 레지스터 설정에 따라 해당하는 VASI 에러가 발생하게 된다.

5. Conclusions

향후 저궤도 위성에 탑재할 고성능 탑재컴퓨터 MCM-ERC32 에서는 자체적으로 6 개의 시리얼 인터페이스를 제공하며 기존의 위성용 시리얼 인터페이스와는 다른 제어방식을 가지고 있다. 기존에 사용하던 UART 를 사용하게 되면 MCM-ERC32 의 UART 가 제공하는 탑재컴퓨터와 연동된 기능들을 활용할 수 없으며, 추가적으로 탑재컴퓨터 보드에 시리얼 인터페이스를 설치해야 하므로 탑재컴퓨터 보드의 크기 및 제작/시험 기간의 증가를 초래할 수 있다. 따라서 탑재컴퓨터 보드에서 제공되는 기능들을 사용하는 것이 효율적이다. 본 논문에서는 MCM-ERC32 에서 제공하는 VASI UART 의 기능 및 특징에 대하여 소개하였다.

기존의 시리얼 인터페이스와는 달리 초기화에서부터 기능설정과 에러처리에 이르는 UART 의 모든 제어가 VASI 의 레지스터 설정을 통하여 관리가 가능하기 때문에 그 절차가 단순하다. 또한 DMA 와 공유메모리인 DRAM 을 활용하여 1553B 인터페이스에서 사용하는 transaction-table 과 같이 전송할 데이터 들을 테이블로 작성하여 주기적인 UART 인터페이스를 수행하는 탑재소프트웨어에 적용할 수 있다.

본 논문에서 소개한 VASI UART 의 기능들은 실제 MCM-ERC32 보드에서 테스트가 수행되었으며, 현재 이를 이용한 저궤도 위성용 탑재소프트웨어의 설계가 수행되고 있다.

참고문헌

[1] "MCM ERC32SC User's Manual", ASTRIUM, 2004
 [2] "TSC695F SPARC 32-bit Space Processor User Manual", Atmel, 2003
 [3] "저궤도 위성용 탑재소프트웨어 개발을 위한 ERC32 프로세서 소개", 한국정보처리학회 춘계학술발표논문집(하), pp 1553-1556, 2005