

아리랑위성 프로세서간 통신 설계

천이진, 이종인, 정창호, 강수연
한국항공우주연구원

The Design of Inter-processor Communication of KOMPSAT

Yee-Jin CHEON, Jong-In LEE, Chang-Ho JUNG, Soo-Yeon KANG
Space Division, Korea Aerospace Research Institute

요 약

아리랑 위성은 자세, 전력, 열 제어 및 지상 명령 수신, 측정 데이터 수집 그리고 탑재체 지원을 위해서 3개의 80C186 프로세서를 사용하고 있다. 단일 프로세서가 아닌 여러 프로세서가 존재하게 되면 상호 간의 메시지 전달을 위해 통신 채널이 요구된다. 프로세서간의 상호 통신을 위해서 직접 연결을 사용하기도 하지만 아리랑 위성은 모듈화 개념 및 향후 확장을 위해서 MIL-STD-1553B 표준 버스 방식을 채택하고 있다. 메시지는 지상 명령 전송 및 측정 데이터 수집을 포함하므로 원활한 통신이 이루어지지 않을 경우, 위성 시스템에 심각한 문제를 발생시킨다. 일반적으로 위성 설계는 안정성과 신뢰성을 추구하므로 통신 설계는 다중 프로세서가 존재하는 위성의 경우 매우 중요한 의미를 지닌다. 본 논문에서는 아리랑 위성 MIL-STD-1553B 데이터 버스의 버퍼링(Buffering) 설계와 메시지의 적절한 배치를 통한 Timed-Scheduling 설계 개념을 설명한다.

1. 서 론

아리랑 위성은 탑재 프로세서로서 80C186을 채택하고 있으며 각 프로세서는 기능에 따라 전력계 제어장치, 원격 구동 장치 그리고 탑재 컴퓨터로 구분되어 있다. 전력계 제어장치(ECU: Electrical Power Subsystem Control Unit)은 전력 발생 및 분배 그리고 위성체에 대한 열제어를 관리하고, 원격 구동 장치(RDU: Remote Drive Unit)은 자세 센서로부터 정보를 받아들이고 위성 자세 및 궤도에 대한 제어를 지원한다. 탑재 컴퓨터(OBC: On-Board Computer)는 지상으로부터의 명령 수신 및 명령의 라우팅(Routing), 원격 측정 데이터의 수집 및 지상으로의 송신을 담당한다. 3 프로세서 이외에 궤도 정보 및 동기화 신호를 제공하기 위해 GPS 수신기가 사용되고 임무를 위한 탑재체가 존재한다.

이렇게 다중 프로세서가 사용되면 상호간의 통신 채널이 존재하게 되고, 통신 채널의 구성은 크게 나누어 중앙 집중형 구조와 분산형 구조로 구분된다^[1]. 분산형 구조는 다시 연결 구조형(Ring Architecture)와 버스형 구조(Bus Architecture)로 구분된다^[1]. 버스형 구조는 모든 프로세서가 데이터 버스를 공유하고 통신 채널에서 표준화된 통신 프로토콜을 사용한다. 아리랑 위성의 경우는 사용된 탑재 프로세서만을 고려한다면 버스형 구조에 속한다고 볼 수 있으나 자세 센서들을 고려한다면 하이브리드 구조에 속한다. 본 논문에서 탑재 프로세서만을 고려한다면 버스형 구조로 보아도 무방할 것이다.

이러한 버스형 구조에서는 앞서 밝힌 바와 같이 표준화된 통신 프

로토콜을 사용하고 아리랑 위성의 경우는 명령-응답 방식의 MIL-STD-1553B 데이터 버스를 사용하고 있다. 버스형 구조는 프로세서간의 배선에서 생기는 무게의 감소뿐만 아니라 모듈화 및 향후 확장성이 좋은 이점을 지니기 때문에 아리랑 위성의 경우 MIL-STD-1553B 표준 버스 방식을 채택하고 있다. 본 논문에서는 아리랑 위성에 있어서 데이터 버스의 메시지 처리를 어떻게 이루어지며, 신뢰성과 안정성을 지니는 Timed-Scheduling 개념에 대해서 설명한다.

2. 프로세서간 상호 통신

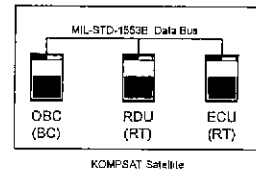


그림 1 아리랑 위성 데이터 버스 구조

앞서 설명한 바와 같이 아리랑 위성은 데이터 버스로서 MIL-STD-1553B를 채택하고 있다. 그림 1은 아리랑 위성의 데이터 버스 구조를 나타낸 것이다. 실제 과학 탑재체와 GPS 수신기와 같은 탑재체도 데이터 버스에 연결되어 있거나 본 논문에서는 3개의 프로세서

간 통신만을 고려하므로 제외시켰다. OBC는 지상과의 링크(Link)를 제공하므로 지상으로부터의 명령 수신, 원격측정 데이터(Telemetry)의 다운링크(Downlink)를 수행하고 수신된 명령이 OBC 이외의 명령인 경우에는 해당 프로세서로 전송한다. 원격 측정 데이터의 경우 각 프로세서는 관련 데이터를 수집하고 OBC로 전송을 하면 OBC는 원격 측정 데이터를 취합하여 지상으로 다운링크한다. 앞서 설명한 바와 같이 MIL-STD-1553B 데이터 버스는 명령-응답의 형태를 취하기 때문에 데이터 버스의 통신을 제어하기 위한 제어가 존재한다. 이와 같은 제어를 BC(Bus Controller)라고 하며 OBC가 이 역할을 수행한다. BC를 제외한 터미널을 RT(Remote Terminal)라고 하고 RDU, ECU 및 GPS 등과 같은 탑재체가 이에 해당된다.

MIL-STD-1553B 데이터 버스는 모든 RT에 동일한 메시지를 전달할 수 있는 Broadcasting 기능을 제공한다. 아리랑 위성의 경우는 각 프로세서의 시간을 이 기능을 통해서 각 RT로 전달한다. 이외에 BC-RT 통신만이 아니라 RT-RT 통신 기능, 즉 RDU-ECU의 통신을 제공한다. 이러한 RT-RT 통신의 제어도 BC가 개입하여 이루어진다.

3. RT Ping-Pong Buffering

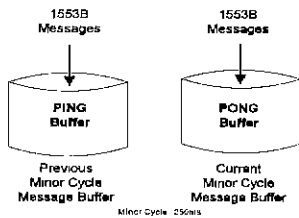


그림 2 RT Ping-Pong Buffering

데이터 버스를 통해 전달된 메시지는 버퍼에 저장되고 RT 소프트웨어에 의해 처리된다. 하나의 버퍼만을 이용하여 메시지의 전달과 수신을 하는 경우 BC와 RT의 통신 동기가 정확히 이루어지지 않으면 심각한 문제를 발생시킨다. 예를 들어 현재 RT에서 수신에 대한 처리를 하고 있는 도중에 BC로부터 송신 명령이 전달되면 수신된 버퍼의 내용이 BC로 전달되는 상황이 발생한다. 이와 같은 문제점을 해결하기 위해 어떤 제조사는 Double-Buffering이라고 하는 개념을 도입하기도 한다²⁾.

아리랑 위성의 경우는 이전의 마이너 사이클에 전달된 메시지를 현재 마이너 사이클에서 후처리(Post-processing)하는 방식으로 처리하고, 다른 버퍼를 이용하여 다음 마이너 사이클의 송신 메시지를 준비하는 핑-퐁 버퍼링 개념을 도입하였다. 그림 2는 핑-퐁 버퍼링을 나타낸다. 예를 들어 이전의 마이너 사이클의 수신에 사용된 버퍼가 핑 버퍼였다면 현 마이너 사이클에서는 버퍼를 전환(Toggle)하여 송신을 위해서 퐁 버퍼를 이용한다. 따라서 송신 및 수신은 현재 마이너 사이클에서 이루어지지 않고 하나의 마이너 사이클 지연이 생기게 된다. 이와 같은 이유는 대부분의 위성에 사용되는 소프트웨어는 확정적(Deterministic)인 성격을 지니므로 태스크들의 운용이 Timed-Scheduling 방식이기 때문이다.

마이너 사이클 단위로 핑 버퍼와 퐁 버퍼가 전환되지만 정확한 의

미에서 통신과 각 태스크의 기준점을 제공하는 마이너 사이클에서 버퍼가 전환되지는 않는다. 그림 3은 마이너 사이클 기준점에 대해서 핑-퐁 버퍼의 전환, 그리고 데이터 버스의 통신을 시간적인 순서에 대해 나타낸 것이다.

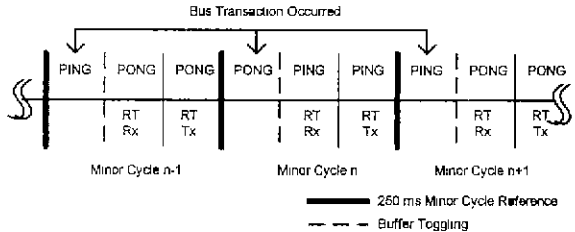


그림 3 Ping-Pong Buffer Toggling

그림 3에서 마이너 사이클 내의 버퍼간의 구분은 실제 처리되는 시간을 나타낸 것이 아니라 시간적인 순서만을 나타낸 것이다. BC로부터 명령 요청은 마이너 사이클 패러런스의 앞부분에서 일어남을 알 수 있다. 또한 실제 버스 통신이 이루어지고 난 다음, 즉 버퍼를 전환하고 난 다음에는 수신(Rx)과 송신(Tx)은 동일한 버퍼를 사용하는 것으로 되어있지만 수신된 데이터가 이전의 데이터를 후처리되고 다음 마이너 사이클에서의 실제 버스 통신을 위한 송신 데이터를 준비하는 것이다. 이런 면에서 보면 송신은 전처리(Pre-Processing)으로 볼 수 있다. 그림에서 RT Rx 및 RT Tx는 실제 버스 통신이 일어나는 부분이 아니며 메시지 프로세싱을 하는 부분이다.

마이너 사이클 n 번째의 경우를 예를 들면, 처음 부분은 BC에 의해 버스 통신이 이루어지는 부분이며 사용된 퐁 버퍼는 (n-1)에서 송신을 위해 준비된 메시지 버퍼이고 (n+1) 마이너 사이클에서 수신 처리될 메시지 버퍼이기도 하다. 따라서 (n-1)번째 메시지 송신은 n 번째 마이너 사이클에서 수행된다. BC에 의해 통신이 이루어지는 동안 송신만 이루어지는 것이 아니라 BC에 의해 BC나 다른 RT에서 수신도 이루어지며 (n+1) 마이너 사이클에서 버퍼가 퐁으로 다시 전환된 다음 수신 처리(Rx Processing)가 이루어진다. n 번째 마이너 사이클에서 버퍼가 퐁에서 전환된 핑 버퍼는 (n-1) 번째 마이너 사이클의 처음 부분에서 버스 통신이 이루어지는 동안 사용된 버퍼이며 RT Rx 프로세싱 동안에 메시지의 처리가 이루어진다. RT Rx 프로세싱이 끝나고 나면 핑 버퍼는 (n+1) 마이너 사이클의 처음 부분에서 실제 버스 통신을 위해 송신 메시지를 저장하는 데 사용된다.

4. BC Transaction

이 절에서는 MIL-STD-1553B 데이터 버스의 통신을 제어하는 BC는 어떻게 동작하는 지에 대해 설명한다. BC는 RT와 같이 응답만을 하는 형태가 아니라 명령을 RT로 전달하기 때문에 RT와 같이 핑-퐁 버퍼 구조를 가지지 않는다.

그림 4는 BC의 메시지 처리에 대한 흐름도를 나타낸 것이다. BC가 처음 부분에서는 이전 마이너 사이클 동안 수행된 통신 결과로 발생한 인터럽트 레지스터를 참조하여 데이터 버스에 심각한 문제가 발생한 지를 조사한다. 인터럽트 레지스터를 참조하여 문제가 발생하

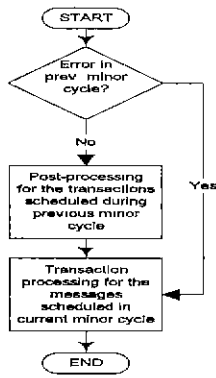


그림 4 BC Transaction Processing Flow Chart

지 않았다면 이전 마이너 사이클 동안에 처리된 통신 결과에 대해 메시지 처리를 수행한다. RT의 경우와 마찬가지로 후처리에 의해 메시지가 처리된다. 메시지가 처리되는 동안 인터럽트 로그 리스트를 검색하여 에러가 발생한 지 여부를 검사하고 문제가 생기면 에러를 표시하고, 그렇지 않은 경우는 메시지를 해당 데이터 어드레스에 저장한다. 이전 사이클에 대한 메시지 후처리가 끝나면 현 마이너 사이클에서 수행되어야 할 메시지의 통신을 위해서 명령 블록의 연결 구조를 만들고 1553B 하드웨어가 통신을 개시하도록 한다. 현재 마이너 사이클에서 수행되어야 할 통신을 검색하기 위해서 트랜잭션 테이블이 존재한다. 트랜잭션 테이블은 해당 트랜잭션 번호에 대해 활성/비활성을 표시하는 플래그, 주기를 나타내는 필드, 마이너 사이클로 표시되는 주기 내의 오프셋, 워드 카운터 주소를 나타내는 필드, 그리고 해당 메시지의 데이터 어드레스로 구성된다. 트랜잭션 테이블에서 현재 마이너 사이클의 카운터를 주기로 나눈 나머지가 오프셋이고 트랜잭션이 활성화되어 있으면 통신 레퍼런스 테이블에 첨가시킨다. 이 통신 레퍼런스 테이블은 트랜잭션 테이블의 검색이 완료되면 현 마이너 사이클에서 수행되는 통신을 위해 사용되고, 앞서 설명한 BC가 이전 마이너 사이클에서 발생한 메시지의 후처리를 할 때 기준으로 사용된다.

5. Transaction Table 구성

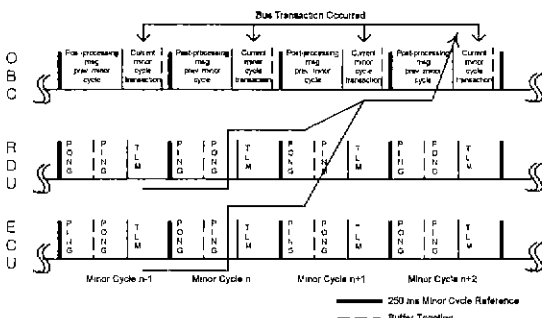


그림 5 Processor Coordination for Telemetry

3, 4절에서 설명한 RT와 BC의 통신을 원활히 수행하기 위해서는

BC에서 메시지를 요구하는 시간에 앞서 RT에서 요구되는 메시지가 준비되어야 한다. 이를 위해 BC와 RT에는 트랜잭션 테이블이 존재한다. 트랜잭션 테이블을 효과적으로 구성하기 위해서는 메시지 전달의 시간적 흐름을 이해해야 된다.

그림 5는 원격측정 데이터가 어떻게 RT에서 BC로 전달되며 각 마이너 사이클에서 RT 및 BC는 어떤 처리를 하는 지를 나타낸 것이다. 그림에서 각각의 처리는 실제 소요되는 시간을 나타내는 것이 아니라 순서만을 나타낸 것이고 BC인 OBC에 대해 표시된 버퍼 전환(Buffer Toggling)은 RT에 대한 시간적 기준을 제공하기 위한 것이다. 태스크 상으로 보면 1553B 처리가 원격측정 데이터 수집보다 선행된다. 예를 들어 RDU의 경우, (n-1) 마이너 사이클에서 수집한 원격측정 데이터는 n 마이너 사이클에서 쏘 버퍼를 이용하여 RT Tx 프로세싱이 수행된다(3절 참조). (n+1) 마이너 사이클의 앞부분에서 BC의 현재 마이너 사이클 트랜잭션이 수행되면 BC로 메시지가 전송된다. 4절에서 설명한 바와 같이 BC의 경우 메시지 처리는 후처리로 이루어지므로 (n+1) 마이너 사이클에서 BC로 송신된 메시지는 (n+2) 마이너 사이클에서 후처리에 의해서 처리된다. 따라서 RT에서 BC로의 메시지 송신은 메시지의 처리를 기준으로 할 때 3 마이너 사이클이 요구됨을 알 수 있다.

메시지 통신을 정상적으로 운영하기 위해서는 BC와 RT내에 존재하는 트랜잭션 테이블을 Timed-Scheduling 개념에 맞도록 구성해야 한다. 그림 5의 예를 들면 RT내의 트랜잭션 테이블에서 (n-1) 마이너 사이클에서 준비된 메시지 전송에 대해 1553B의 송신 처리는 n 마이너 사이클에서 이루어지도록 해야하고 BC의 경우 (n+1) 마이너 사이클에서 실제 데이터 통신이 이루어지도록 트랜잭션 테이블을 구성해야 한다. 이러한 Timed-Scheduling이 이루어지기 위해서는 트랜잭션 테이블 구성도 중요하지만 앞서 각 프로세서의 시간적인 동기가 우선되어야 한다^[1]. 프로세서간의 시간적 동기가 이루어지지 않았을 경우, 그림 5에서 알 수 있듯이 정상적인 메시지의 송수신이 이루어지지 않는다.

6. 결 론

본 논문에서는 아리랑 위성에서 사용된 MIL-STD-1553B 데이터 버스의 운용에 대해 설명하였다. RT의 펌-퐁 버퍼 및 BC의 통신 설계, 그리고 원활한 상호 통신을 보장하기 위한 트랜잭션 테이블 구성 방법에 대해 알아보았다. 안정성 있는 통신 설계는 위성 운용에 있어서 명령 전달 및 원격 측정 데이터 수집에 안정성을 제공해 준다.

7. 참 고 문 헌

- [1] W. J. Larson, J. R. Wertz, Space Mission Analysis and Design 2nd Ed., Microcosm Inc., pp 610-612, 1995
- [2] MIL-STD-1553 Designer's Guide 4th Ed., ILC DDC, 1993
- [3] Y. J. CHEON, et al., The Design of Inter-processor Synchronization of KOMPSAT, The 25th KISS Fall Conference, 1998(To be published)
- [4] 1553 Product Handbook, United Technologies Microelectronics Center, 1992