CCSDS IN THE LORAL 550

Jerry Sutton and Larry Taylor
Loral Test & Information Systems
15378 Avenue of Science
San Diego, CA 92128

ABSTRACT

The Consultative Committee on Space Data Systems (CCSDS) did not create a specification like the IRIG 106, but rather a recommendation [1-4]. That means that each country, community, and application is free to select subsets, adapt techniques, and even alter the structure to suit particular needs. This variability places new demands on a decommutation system. The implementation of the CCSDS Recommendation in the Loral 550 accommodates this "variability within a structure" by using a modular and adaptable collection of structured components. The result covers the two most popular versions of CCSDS: Conventional/Telecommand and the Advanced Orbital Systems (AOS) in both operational and test modes, and couples the CCSDS inputs and outputs to a host of other data format and processing options.

KEYWORDS

CCSDS, Telemetry, Packets, AOS, Telecommanding, Satellites

CONVENTIONAL/TELECOMMAND AND AOS VERSIONS

Customers are currently requesting two major versions of CCSDS. Figure 1 shows the various levels and services we have seen. One version, a subset of the original recommendation, is called "Conventional CCSDS" for the Down or Return Link and "Telecommand" for the Up or Forward Link. The two directions are implemented very differently. The other version is the AOS. Both directions in the AOS are the same.

TEST MODES

In addition to the four modes described above (two versions times two directions), each mode has a corresponding test mode. In test mode for the AOS Return Link, for example, the processor should generate a CCSDS data stream for the Return Link to

AOS Return Link

Physical layer support
- *Sync strategy*
- *Polarity detect/correct*
- *Bit slip detect/correct*

CADU processing
- *Derandomization*
- *CRC*
- *Bit order reversal*

Reed–Solomon Decoding
VCDU processing
Fill detect and discard
Insert Service
Bitstream Service (B_PDU processing)
Multiplexing Service (M_PDU processing)
Path Service (Packet processing)
E_PDU processing
ISO 8473 Internet Packet Service
Depacketizing measurands
DES decryption
Decommutation/decoding/display
NASCOM processing
Data storage and replay
Data rate buffering
Status and Statistics gathering, reporting

AOS Forward Link

ISO 8473 Internet Packet Encapsulation
CCSDS Packet Encapsulation
Path Service (packet processing)
Multiplexing Service (M_PDU processing)
Bitstream Service (B_PDU processing)
DES encryption
VCDU processing
Reed–Solomon Encoding
CADU generation
Physical layer support
- *Accurate clock generation*

CADU processing
- *Sync word insertion*
- *Randomization*
- *CRC generation*
- *Fill insertion*

NASCOM processing
Status and Statistics gathering, reporting

Conventional Return Link

Physical layer support
- *Sync strategy*
- *Polarity detect/correct*
- *Bit slip detect/correct*

Transfer frame processing
- *Derandomization*
- *CRC*
- *Bit order reversal*

Reed–Solomon Decoding
Fill detect and discard
CLCW extraction
Packet processing
Segment recombining
Depacketizing measurands
DES decryption
Decommutation/decoding/display
NASCOM processing
Data storage and replay
Data rate buffering
Status and Statistics gathering, reporting

Telecommanding Forward Link

Segmentation
BCH Encoding
CLTU assembly
Transfer frame generation
- *Sync word insertion*
- *Randomization*
- *Acquisition sequence insertion*
- *Idle/fill insertion*

Physical layer support
- *Accurate clock generation*

NASCOM processing
Status and Statistics gathering, reporting

Figure 1. CCSDS Services Available from the Loral 550

decode. This could be anything from a simple replay of a static test file to a fully dynamic Return Link Simulator, depending on the sophistication of the capabilities that need to be tested[*].

USAGE

We have seen three basic applications for CCSDS processors: 1) operational units whose purpose is to extract data from operational space vehicles, 2) test units whose purpose is to evaluate the performance of other devices that are under test (e.g., satellites, operational units), and 3) network devices whose purpose is to perform limited levels of CCSDS processing on the data stream(s) and measure their quality before forwarding them. The first two require all levels of service while the third requires only a few.

LEVELS OF SERVICE (A CCSDS RETURN LINK TUTORIAL)

The CCSDS Recommendation defines many levels of service, all the way from receiving and forwarding raw Channel Access Data Units (CADUs) to extracting individual measurands or commands.

To illustrate the levels in one of these links, consider the expanded description of the AOS Return Link processing steps shown in Figure 2. CADU processing (steps 1-3) consists of frame sync, optional derandomization (removing a polynomial that ensures sufficient bit transitions in the RF link), and optional CRC calculation (another polynomial that ensures no errors are in the frame).

Step 4) Reed-Solomon processing can detect and correct up to 16 errors per frame or detect that there are more than 16 errors. Uncorrectable errors in any of the above tests are grounds for discarding the frame or redirecting it for further analysis. The result of passing the tests is a valid Virtual Channel Data Unit (VCDU).

Step 5) A first-order sort is performed at this point to send all members of each series of VCDU to its own processing channel so that lower level data units embedded in the VCDUs can be reassembled across VCDU boundaries. Examining counters and flags in the headers of the VCDUs determines whether all data units are present or not. VCDUs are made up of smaller data units such as Bitstream Protocol Data Units (B_PDUs) and Multiplexed Protocol Data Units (M_PDUs).

---

[*] The test modes need to include scoring software to keep track of the errors found versus the errors induced, along with a display of the score. These tools are invaluable for checking system integrity as well as for evaluating another unit under test.

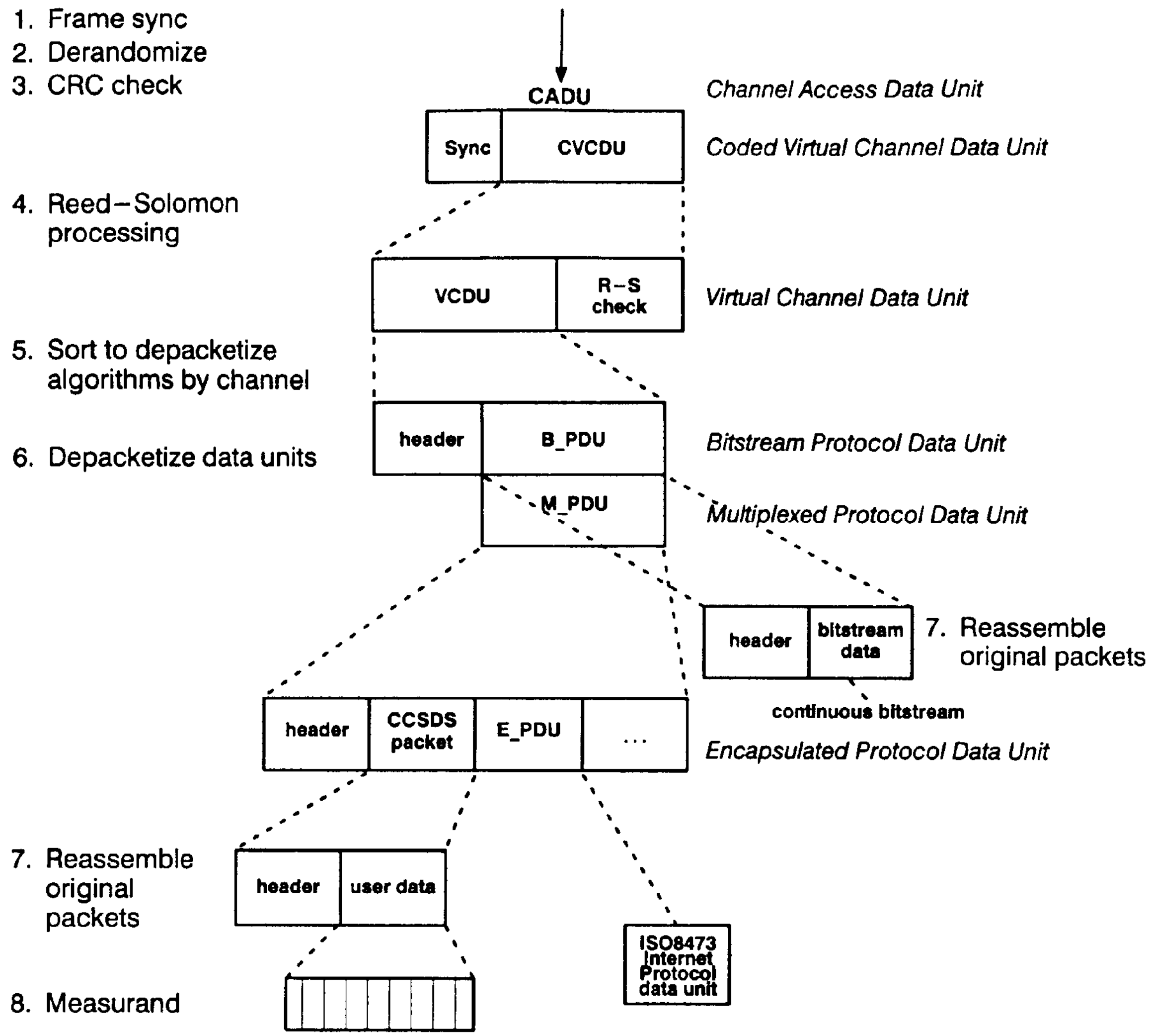

Figure 2. CCSDS Downlink Processing

Step 6) The M_PDUs are broken up into the next lower level of packets (e.g., CCSDS packets — to the unwary, the name of the packet is a cruel source of confusion with the overall technique) and sorted for transfer to the next lower level of processing for decoding.

However, this is not a simple case of continuing to break data units into smaller and smaller packages. At Step 7) the data units can become very long when lower level data packets are reassembled into their original form, since long packets have to be broken into pieces for shipment over the airwaves. At Step 8) they are finally decoded into measurands (or more packets in some cases, that go on to yet another layer of depacketizing).

A similar result occurs in the B_PDUs with the recovery of a continuous embedded bitstream of data. Such a bitstream might be a PCM stream, another CCSDS stream, or something else entirely. It is represented as a continuous series of 8-bit octets. It is up to some other process (e.g., a PCM decom, another CCSDS processing string, or an output interface) to handle the bitstream.

Figures 1 and 2, then, represent a basic set of requirements that a CCSDS processing system must provide. The CCSDS Recommendation contains a variety of other services that could be added when needed.

PROCESSING OPTIONS

Besides the processing steps of sorting and checking inherent in CCSDS processing, three other forms of processing were covered in the discussion of step 4) above — accept, dump, or divert. Acceptance results in passing the data unit on to the next processing step. Dump means the data unit goes no further. Divert means sending the data packet to a disk for storage or sending it on to another processor or a human being for further study. These could also happen at any other level. Other kinds of processing should include:

- Outputting intermediate levels of data units to storage or display — this could be handy during debugging operations
- Outputting measurands to storage, displays, or other processors
- Outputting data in another format (e.g., PCM, 1553, NASCOM packets)
- Reporting status of CCSDS processing steps (e.g., VCDUs processed, number of errors found)
- Encrypting data units before transmission for security
- Decrypting data units after reception
- Encoding with Reed-Solomon error protection
- Generating CRC
- Recovering from errors
- Reversing data order (for data played backwards from magnetic tape)

LORAL 550 IMPLEMENTATION OF CCSDS

The general strategy for implementing CCSDS processing in the Loral 550 is illustrated in Figure 3. The Loral 550 uses a combination of hardware, software, and the data bus to process the CCSDS data stream.

Frame synchronization, polynomial randomization/derandomization, and CRC generation/check are best performed in hardware because of the intensity of bit-shifting involved. The Loral 550 has both 10 Mbps and 50 Mbps Telemetry Input Output (TIO) modules. The 10 Mbps version has two input channels and two output channels. The 50 Mbps version has one of each. The output clock source is provided by a frequency synthesizer on the module. The clock can also be driven externally or derived from an external frequency reference.

Reed-Solomon processing is done in software at lower speeds (less than about 800 Kbps) but is better done in hardware at higher data rates. In both cases, a programmable real-time processing module, the Field Programmable Processor or FPP3, is used to perform the function. In the lower speed case, the Reed-Solomon processing runs as an algorithm. In the higher speed case, the FPP3 manages the data input to and output from a dedicated hardware circuit.

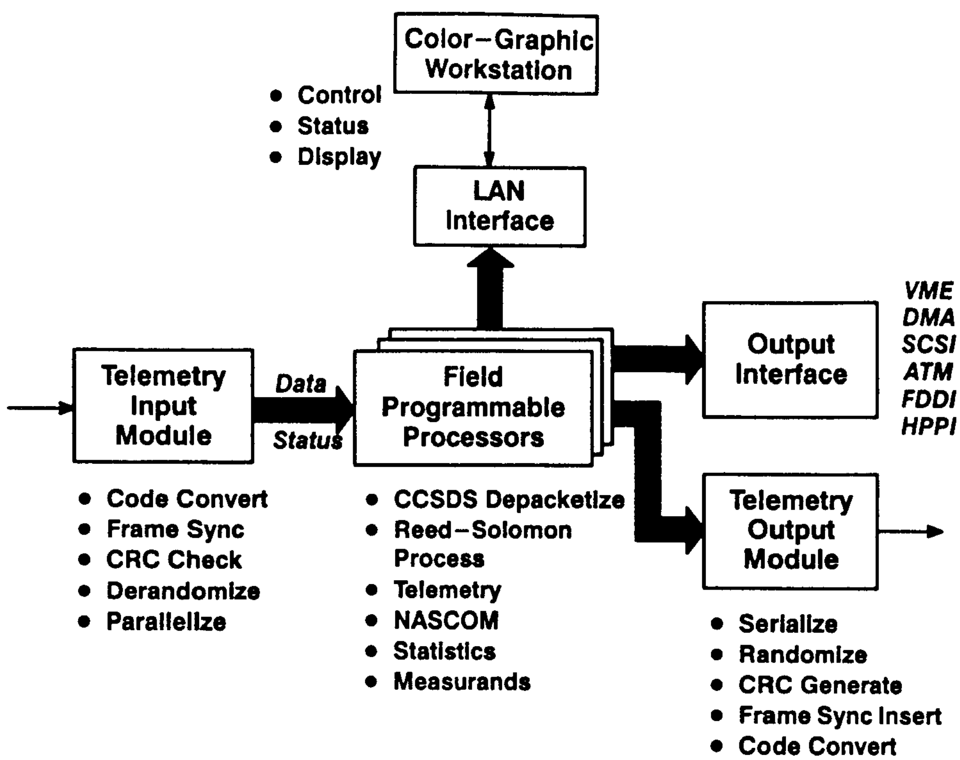

Figure 3. Packet Processing Strategy

Packetizing and depacketizing are best done in software. Each of the CCSDS processing levels is coded as an independent real-time data-driven algorithm in one or more FPP3s. As each algorithm completes its processing step, it places the resulting data packet or measurand back on the MUXbus for transmission to the next processing algorithm, even if that is in the same FPP3. This allows the sorting and reassembling process to occur automatically between levels.

A hardware assist is provided to the FPP3 at high data rates (greater than 10 Mbps). A mezzanine circuit card is added to the FPP3 to move data into and out of the module using hardware to allow the processor to spend its time processing rather than in I/O.

Thus, in the lower speed applications, the entire CCSDS processing task, from CADU to measurand, can be accomplished with two modules: the TIO and the FPP3. The number of FPP3s depends on both the data rate and the number of levels of processing to be performed.

In the higher speed applications, the same two modules are used with the addition of a small mezzanine module on the FPP3 that holds the hardware I/O and controls the Reed-Solomon decoder and encoder. Again the number of FPP3s is determined by the data rate and the number of processing levels.

In addition to the central processing elements shown in Figure 3, a time base, such as an internal time clock, is supplied from hardware for forming data units in the Forward Link. This clock can be synchronized to an external IRIG or GPS time source. The IRIG Decoder/Generator module locks onto IRIG time when it is available and automatically flywheels whenever the IRIG signal disappears ensuring that the CCSDS Forward Link or test case data continues to flow in the system.

The Data Encryption Standard (DES) encryption/decryption step unique to the Space Station Forward Link is implemented in hardware as another attached processor on an FPP3 module.

At the heart of the Loral 550 is a high-speed data bus that routes data between processing modules by means of an ID tag. Data sorting and diverting are accomplished simply by assigning the proper tag to a piece of data when it is put out to the bus by any processing level. Packets are identified by a triplet of tags, one for the beginning word, a second for all the middle words, and a third for the end word of each packet.

Once data is broken down to any level identified by a tag, a variety of further processing steps can be performed on it. These encompass anything else the Loral 550 has been designed to do, including:

- PCM decommutation in either hardware or software
- 1553 processing
- NASCOM encoding/decoding
- Parallel I/O
- Analog input and output
- FDDI, ATM, HPPI, and other high-speed inputs/outputs
- EU conversion and other real-time processing algorithms in the FPP3
- A variety of color-graphic and data displays
- High-speed and/or high-capacity storage
- Printout of alphanumeric and graphic displays
- Archival and replay of previous data

In light of the above list, the CCSDS capability in the Loral 550 is just another capable I/O technique in a long list of capabilities provided in a sophisticated real-time signal processor.

ADAPTABILITY WITHIN STABILITY

The hardware and software algorithms of the Loral 550 make every attempt to implement the input and output services as defined in the Conventional/Telecommand and AOS CCSDS Recommendations. Since the basic recommendation is not adhered to as completely as the IRIG standards, Loral provides the source code to all the CCSDS algorithms and the utilities with which to easily make modifications to those algorithms for special needs.

In addition to the algorithm source code, the Loral 550 contains a variety of Application Program Interface (API) points so that users can easily add control layers, special processing routines, third-party software, and custom displays to the system to meet application needs. Even standard third-party VME modules can be added to the system to meet special requirements.

In the future, higher speed modules can be added to accept and produce even higher data rates than the current 50 Mbps limit. Distributing data via the MUXbus can spread the processing load across multiple FPP3 processors for more data throughput. Existing Gateway modules allow multiple chassis to be interconnected for more parallel processing so that even the proven 96 Mbps transfer rate of the MUXbus is not a throughput limitation.

SUMMARY

The versatile architecture of the Loral 550 combined with two new modules provides a sophisticated CCSDS capability in a cost-effective commercial-off-the-shelf product that is available now. This system meets the applications we have seen to date and is adaptable to meet the evolving needs of future users as well.

REFERENCES

[1] Consultative Committee for Space Data Systems Recommendation for Space Data Systems Standards, "Blue Book, Issue 3 (CCSDS 101.0-B-3) — Telemetry Channel Coding," CCSDS Meeting, Oberpfaffenhofen, Germany, May 1992.

[2] Consultative Committee for Space Data Systems Recommendation for Space Data Systems Standards, "Blue Book, Issue 2 (CCSDS 701.0-B-2) — Advanced Orbiting Systems, Networks and Data Links," CCSDS Meeting, Annapolis, MD, USA, November 1992.

[3] Consultative Committee for Space Data Systems Recommendation for Space Data Systems Standards, "Blue Book, Issue 3 (CCSDS 102.0-B-3) — Packet Telemetry," CCSDS Meeting, Annapolis, MD, USA, November 1992.

[4] Consultative Committee for Space Data Systems Recommendation for Space Data Systems Standards, "Blue Book, Issue 1 (CCSDS 201.0-B-1) — Telecommand," CCSDS Plenary Meeting, Frascati, Italy, January 1987.