KOMPSAT-2 MSC DCSU Recording Mechanism

J. T. Lee, S.G. Lee, S.T. Lee
KARI
P.O Box 113, YuSung-gu, DaeJoen 305-600, Korea
jtlee@kari.re.kr, sglee@kari.re.kr, stlee@kari.re.kr

Abstract: The DCSU performs satellite payload data acquisition and storage functions before sending the data to the ground station. While imaging, the DCSU makes a file per each input channel and store into memory stack. For the successful imaging mission, proper DCSU mission parameters should be uploaded before the mission such as file name, file size, output channel for the download transmission and so on. This paper will describe the DCSU recording mechanism and some notices that might be helpful for the ground operators.

Keywords: DCSU, KOMPSAT-2.

1. Introduction

The KOMSAT-2 DCSU(Data Compression & Storage Unit) is a storage device to hold the image data temporally before sending the data to the ground station. Not only in playback mode but also in direct mode, the DCSU uses memory stacks as a circular buffer. In addition to storage function, the DCSU also performs compression function to save memory space and to reduce transmission time for data download.

KOMPSAT-1 had a memory system call SSR (Solid State Recorder) in the FMU(Formatter & Multiplexer Unit). But it had no compression function and no file system. After the configuration of partition, one memory partition for an input channel, record operation was done by opening input channel gate. Even the handling of the memory write pointer was not necessary for nominal operation. The handling of memory read pointer was the most significant job to decide which image data to download. It could be incoming image data or previously stored image data.

As the KOMPSAT-2 DCSU has more functions, more configuration and operational parameters are needed to perform a mission. Improper configuration or flawed operation command will cause a loss of a pass.

2. DCSU System Architecture

1) Video Input Format

Below figures show the DCSU video input data format at compression and non-compression mode. Gap I is an extra time after sending video data header during the time form line sync mark to real pixel data first coming out. The size of Gap II is dependent on the line rate used in the mission. The format difference between compression mode and non-compression mode is using SOI and EOI marks or not(refer to table 1 for special characters for HotLink serial Communication). It's because the DCSU limits the maximum number of any special character per block of 239 byte data to less than 15 in non-compression mode.

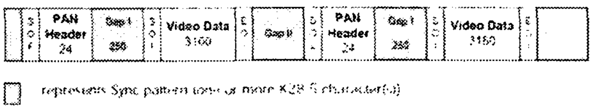

Fig. 1. PAN Input Data Format at Compression Mode.

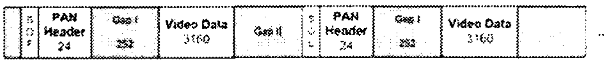

Fig. 2. PAN Input Data Format at Non-Compression Mode.

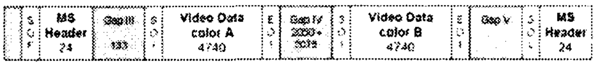

Fig. 3. MS Input Data Format at Compression Mode.

Fig. 4. MS Input Data Format at Non-Compression Mode.

Table 2. The DCSU Input Markers

Marker Name	Hotlink Control Char (n)	Hotlink Control Char (n+1)	Function
SYNC	K28.5		In between data in order to keep hotlink synchronized (at least one each 2048 bytes)
SOF	K28.1	K28.1	Start of first line of block of 32 PAN lines or 8 MS lines
SOL	K28.2	K28.2	Start of line (except if the line is the first of one block)
SOI	K28.3	K28.3	Start of image data (after instrument header)
EOI	K28.4	K28.4	End of image data (might be before end of line period)

2) Elementary Operation List

The DCSU performs a mission by executing elementary operation such as Write, Read, Direct, OBCAUX Write operation. Write and Direct elementary operation have 2 operation lists. One for compression mode and other is for non-compression mode. The selection of which list to be executed is decided by the DCSU mode on time receiving an execution command. So there are 6 elementary operation lists in GCU work data memory space.

To perform a mission each operation must be uploaded into elementary operation list with proper parameters first. Mission starts when the DCSU receive TC_EXEC STC from the PMU indicating which kind of operation should be executed.

Elementary operation list is a kind of FIFO buffer that stores maximum 64 operations per list. When execution command is received, the top operation of the list is executed first then read pointer moves to the next operation. There is no way to execute the second one first. Previous executed operation is not removed from the list automatically and there is no indication how many operation lists are stored or how many free space are remained. The ground station has to send TC_RAZ STC from time to time to clear the elementary operation list.

The validation of operational parameters is not performed during uploading elementary operation list. Validation is checked after receiving execute command. Faulty command detection during mission might cause a lost of a pass.

3) File Catalog

The DCSU stores the data as a file format. After a recording mission one file per input channel is made. A file ID consists of file folder ID requested by record operation and source channel ID appended by the DCSU (refer to table 2). One file catalog entry is made of 8 fields(refer to table 3). Every field is 32 bits.

The DCSU file consists of linked list of sectors. First address means the first sector address from which the file starts. The next sector address is represented on the sector allocation table at the first sector address. The file size is the exact number of sectors that hold file data. In write operation the file size equal to the request sector size and in direct mode the file size equal to the circular buffer size. The sector allocation is done at the beginning of the mission but in direct mode with the record the surplus sector than incoming data will be freed at the end of direct operation. File CRC is a checksum. At read operation the CRC value will be compared with calculated one while reading. When reading a partial file the CRC error might occur. It can be ignored.

Table 2. The Composition of File ID

File ID (B15:0)		
Folder ID (B15:4)	Source Channel ID (B3:0)	
	OBCAUX	0b0000
	PAN1	0b0001
	PAN2	0b0010
	PAN3	0b0011
	PAN4	0b0100
	PAN5	0b0101
	PAN6	0b0110
	MS1	0b0111
	MS2	0b1000

Table 3. The Composition of File Catalog

Field 1	File ID
Field 2	First Address
Field 3	File Size
Field 4	File CRC
Field 5	Reserved
Field 6	Reserved
Field 7	Reserved
Field 8	Reserved

The DCSU manages maximum 1024 File ID. There is no telemetry that indicates how many files are stored and which file IDs are used. Trying record mission with already exist file ID will make an error. It is better the ground station keep DCSU file log book than dump whole file catalog table from the DCSU GCU memory to know file status. The number of free sectors is on the DCSU global state telemetry.

File erase command will not erase actual data sectors but remove the file ID from the file catalog.

3. DCSU Recording Sequence

1) Configuration Set-up

One of things to do before recording mission is proper configuration of the DCSU. After the DCSU power-on it automatically configures itself with nominal value. Primary interface board will be used for a mission and from 1 to 8 compression chains will be used at operational mode. Both memory boards and all 18 clusters will be powered on at retention mode. Among the 18 clusters, nominal 16 clusters will be used for read/write operation. The rest redundant clusters will be just powered on and have an idle state.

Operator has to check current configurations, anomalies, if any, and then reconfigure the DCSU, if it needed. Reconfiguration can be achieved by the DCSU STC; LD_CONF_MEM, LD_CONF_ROUT and LD_CONF_ COMP. GCU(General Control Unit) redundancy is only selectable by PMU(Payload Management Unit) power command. But CCU(Compression Control Unit) redundancy can be controlled by the LD_CONF_ COMP STC. Default configuration is in affect until another valid configuration command is arrived.

Sector level memory configuration which is originated by faulty memory device will be lost while rebooting. Unlike cluster reconfiguration, sector level memory reconfiguration can not be initiated by user command. Only memory self-test test performs sector level memory reconfiguration automatically with test results. If there is any invalid sectors in nominal clusters, memory cluster reconfiguration or memory self-test is needed after every DCSU reboot. Memory configuration and memory self-test operation are memory data destructive operations.

2) Uploading Tables

The DCSU has a JPEG-like compression function. Same compression but slightly differ in header and rate regulation function. Rate regulation function is controller by compression ratio. The DCSU always try to generate request number of compressed data by temporally more strong compression or by adding some zero bytes.

To perform compression function, it needs compression parameters such as compression ratio, Huffman table and quantization table. For better compressed images the DCSU have 64 compression parameter tables which indicate compression ratio, beta and K value, 16 Huffman tables and 8 quantization tables. 32 compression parameter tables, 3 Huffman and 2 quantization tables are preloaded on a PROM so that these tables are always available after the DCSU power on. Other 32 compression parameter tables, 8 Huffman tables and 4 quantization tables are free to be uploaded and used by user. These user defined tables is located at GCU data memory. It's recommended to use Huffman table index

0 for compression ratios 3 and 5, Huffman table index 1 for compression ratio near 1.26 and Huffman table index 2 for compression ratios greater than 8.

3) Uploading Elementary Operation List

To perform a mission elementary operation list must be uploaded at retention mode before start a mission.

4) Changing Mode to Operational Mode

Missions are only executable at the DCSU operational mode(compression or non-compression mode). By entering operational mode, interface configurations and compression configurations will take effect.

5) Sending TC_EXEC command

Upon reception of TC_EXEC STC, the DCSU takes these sequences.

1. The DCSU verify the elementary operation is executable.
   If write, direct or OBCAUX operation is currently under execution when the write operation is requested, write operation will be ignored and an anomaly that indicates the elementary operation is not executable will be risen. If read operation is currently under execution, the write operation is executable but direct operation is not executable.
   The DCSU also check the elementary operation is executable with current hardware configurations.

2. Check the number of elementary operations stored in the list is equal or greater than the TC_EXEC parameter. Multiple read operations can be done by one command.

3. Check the input and the output of the DCSU concerned by the operation are available.

4. Initiate file allocation with requested file ID.
   Whatever the DCSU operation(direct or write) the memory allocation depends on memory fragmentation(distribution of used and unused sectors). In the best case of an empty memory, the allocation duration for a folder of 8 files is:
   allocation time = 150 + 8 x max(4, 0.015 x nb_add) ms
   in a worst case of fragmentation the previous rule becomes:
   allocation time = 150 + 8 x max(4, 0.125 x nb_add) ms
   In example, the allocation of a folder containing 8 files of 100 sectors each will take;
   Allocation at best case
   = 150 + 8 x max(4, 0.015 x 100) ms
   = 150 + 8 x max(4, 1.5) ms
   = 182 ms
   Allocation at worst case
   = 150 + 8 x max(4, 0.125 x 100) ms
   = 150 + 8 x max(4, 12.5) ms
   = 250 ms
   It means the sector allocation time get worse by using unless the DCSU is rebooted. Possible allocation time is in the table. 4.

Table 4. The DCSU Sector Allocation Time

Nb_add	Best Case (sec)	Worst Case (sec)
100	0.18	0.25
500	0.21	0.65
1000	0.27	1.15
2000	0.39	2.15
5000	0.75	5.15
8192	1.13	8.34

5. Initiate compression boards and CCU board in the case of compression mode. If compression ratio index is not one of the default compression ratio indexes, load the compression ratio from GCU memory to CCU memory. Also if Huffman table index and quantization table index is not one of the default tables indexes, load the tables from GCU memory to CCU memory. These are because the CCU board is powered off at retention mode.

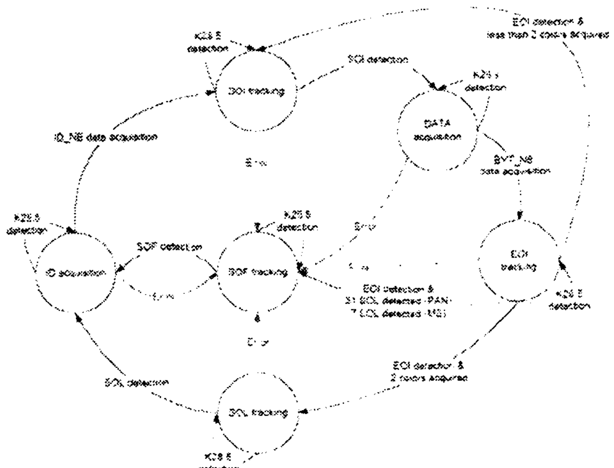

Fig. 5. Synchronization and error detection mechanism in compression mode.

6. Activate INOU function.
   The DCSU open input channels and waiting video data to record. The SOF marker starts new video lines each 32 PAN lines, the 31 next PAN lines start with SOL marker. Because the DCSU try to find SOF marker first, any data before the first SOF marker will be ignored by the DCSU. If the fist SOF pattern changed to a different character, the following block of line is lost(32 PAN lines, 8 MS lines). The ASIC in charge of data recording ignores the incoming data(video data and other markers) until next SOF marker detection. If another SOF pattern, different from the first, changed to a different character it ignored. No indication is provided to indicate the loss of

a SOF marker.
During the various tracking phases, a marker is considered as detected when two successive control characters different from K28.5 are received and those two characters match the expected pattern. In other case an error flag will be asserted and the current marker is considered as lost. Following each error detection, the state machine gets back to the SOF marker tracking state. This ensures to restart properly the synchronization and compression process. But the data from the error to the next SOF mark will be ignored. In the case of marker lost: the current PAN strip(2600 pix) will contain one or several erroneous group of 32 lines and the current MS color will contain one or several erroneous group of 8 lines.
If a DC(Data Character) changed to a SC(Special Character), the current data(SC) is ignored. A byte is missing in the output data flow. If a SC changed to a DC, the erroneous SC is not ignored and a wrong data is inserted into the data flow. No indication is provided to indicate a DC changed to a SC. The Special characters are filtered within the limit of 14 in a block of 239 data. Synchronization mechanism in compression mode is in the Fig. 5.

7. Stop recording operation.
The DCSU provides / outputs the data on the I & Q outputs using sectors with identical and fixed size(244,736 bytes). This maintains even if the data is less than a sector, and a full sector is always transmitted to the output.
A recording operations based on a memory allocation of n sectors. The DCSU will stop the recording operation when the requested sectors are filled. But also the recording operation can be stopped by M_RET STC. Upon M_RET STC reception the DCSU shall get back to the retention mode. The stopping sequence is the following.
- delete the current sector(@3 on the Fig. 6)
- read data from beginning of the file to the last sector entirely written
- free the rest of sectors including the current sector(@3 to @n)

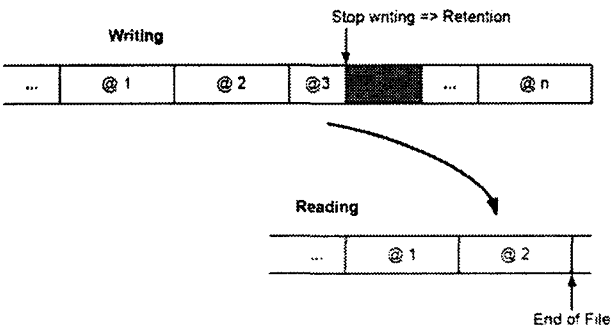

Fig. 6. Stopping Sequence.

This stopping sequence leads to the lost of the writing CRC due to an incomplete written sector is erased. However the previously written sectors are correctly RS encoded in memory. Thus, in case of reading of an incomplete file, the DCSU will return following report:
- the number of sector recorded up to the M_RET STC reception
- number of frame in error equal to 0(filed of RP_xx_EXE message)
- a "E_FILE_CRC_ERROR" (source = 0xB002: CRC_NOT_MATCHING) anomaly will be signaled in anomaly table for each incomplete file

8. Generate the elementary operation report

4. Conclusions

The KOMPSAT-2 DCSU performs the storage and compression function. All video data is record in the DCSU with file format to transmit to the ground station. This paper presents the mechanism of making a file from the receiving the image data and could help the ground operators preparing a mission and understanding the DCSU outcomes.

Acknowledgement

This Paper is the result of MSC research and development supported by the Ministry of Science and Technology.

References

[1] L. Girard, 13/06/2004. DCSU Operational Handbook.
[2] G. Salanie, 06/04/2004. DCSU Interface Data Sheet.
[3] DCSU Design Team, 16/05/2003. DCSU Architectural & Technical Description.
[4] L. Girad. 14/11/2003. GCU Software Requirement Specification.