Camera Controller in MSC(Multi-Spectral Camera)

Young Sun Kim*, Sang-Soon Yong, Jong-Pil Kong, Haeng-Pal Heo, Jong-Euk Park and Heong-sik Youn
Korea Aerospace Research Institute, Eun-Dong 45, Yusung-Gu, Taejon, Korea
*yskim1203@kari.re.kr

Abstract: The CC's main objective is to manage and control the various operation of the MSC camera. The CC has capability to control the various camera operation modes such as INIT mode, WAIT mode, STANDBY mode, READY IMAGING, DEFAULT READY IMAGING, IBIT and IMAGING mode as well as to manage the interface of the PMU. This paper also shows not only the design concepts in the both of the hardware and the operational software, but also the implementation results for the various CC functions.
Keywords: CC, MSC, operation.

1. Design Concept

The MSC has two Electro-optical channels, which are the PAN and the MS channel. Both channels can perform imaging at synchronous rate. The MSC shall cover a swath width of 15km from an altitude of 685km. The PAN(Panchromatic) channel has 15,000 active pixels of 1m-ground resolution. The MS(Multi-Spectral) channel has 3750 active pixels of 4m-ground resolution in each of the 4 spectral bands.

The camera module in the MSC is a slave to the PMU(Payload Management Unit), which is also a subsystem of the MSC, and fully controlled by the PMU. In addition, the camera module receives the required power supplies from the PMU. Fig.1 shows the camera module block diagram.

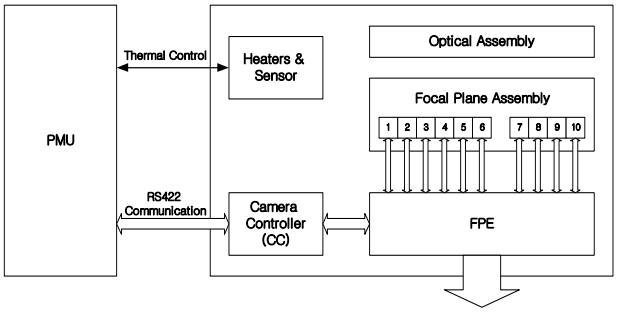

Fig. 1. Camera module block diagram

The CC, as a part of the camera module, is responsible for controlling the PAN and the MS camera to take pictures of ground target. In the next, this paper shows the design concept in the hardware and the operational software point in turn.

1) CC Hardware Design

The CC has capability to control the video-processor, read the code from the flash, read and write to the SRAM, read telemetry voltage, transmit and receive commands from the PMU through RS422 interface, communicate the FPE through the serial protocol defined by ourselves, generate the line sync for the PAN and the MS channel and power up the regulator of the FPE and so on. To do above, the CC comprises of a CPU, a FPGA, a MUX & A/D, a UART(Universal Asynchronous Receiver Transmitter), memory devices, logic devices. Fig.2 depicts the block diagram of CC hardware.

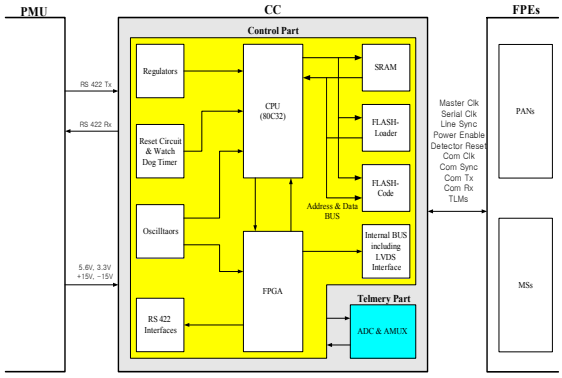

Fig. 2. CC Block Diagram

In this diagram, especially, the FPGA block is important because the many functions are implemented using FPGA technology for the reliability and the specified function. The FPGA block is divided into three blocks, which are the DECODE_LATCH, the WATCH_DOG and the COMM_UART block.

The DECODE_LATCH block is the heart of the FPGA as it implements the vital functional requirements. This block generates enable signals according to the data and the command received from the microcontroller, operates two flash components and the RAM, generates the line synchronization signals to the PAN and the MS channels, controls A/D converter and internal distribution of A/D data and translates commands from the PMU to serial data and forwards it the FPEs. This block is comprised of a number of sub-blocks. The SERIAL DATA sub-block is in charge of receiving commands from the PMU and directing them to serially to the FPE. This paper defines the own communication protocol using a synchronous communication method for the purpose of a fast and accurate data transmission. This protocol is defined as three output signals, which are TTX, CCLK, SSYNC. The FPEs use them for timing and synchronizing the data received. The LINE SYNC GENERATION sub-block is to forward the line synchronization signals to the PAN and the MS imaging channel. The A2D INTERFACE sub-block serves as an interface

between microcontroller and the A/D converter. In general, it informs the A/D converter when to begin sampling the analog information and transmits converted data from the A/D converter to the microcontroller. The DISCRETE_LINE sub-block generates various enable signals that serve other sub-blocks within the FPGA as well as external units, according to communication received from the microcontroller. The discrete signals generated by this sub-module are the LLINE_SYNC_EN_PAN, the LLINE_SYNC_EM_MS, which are used by the LINE_SYNC_PAN_MS sub-block, the RRST_LOW, the EN_BUF_DA and the EN_CLKS, which enables the transmitting of all the fast clocks to the FPEs. Fig. 3 shows the block diagram of the DECODE_LATCH block.

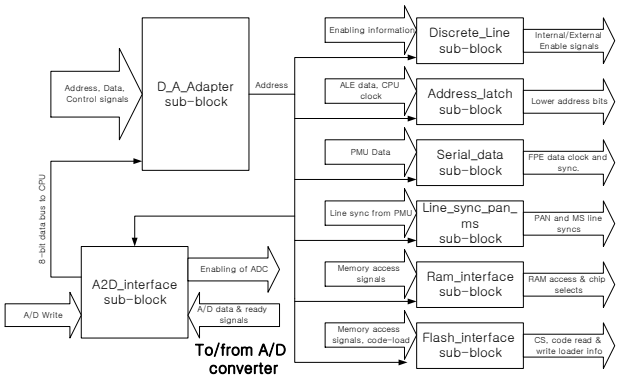

Fig. 3. FPGA's DECODE_LATCH block Diagram

The WATCH_DOG block supervises the microcontroller's operation by means of a polling function. If the watchdog input is not toggled by the microcontroller within a preset time period, exactly 1.6 sec, an output reset signal that acts as an interrupt is generated and transmitted to the microcontroller in order to reset it. An active low manual reset signal is also incorporated for direct control of the reset function, if required.

The COMM_UART block is to translate and forward the communication signals arriving from either the on-board UART. As for data arriving from either primary or redundant microcontroller, the block transmits the relevant data by implementing an AND function on both lines.

2) CC Operational Software Design

The CC software is designed to control the various operation modes as well as to manage the interface of the PMU. Since the CC is basically slave to the PMU, all the CC operational modes are initiated by the PMU.

In case of a communication malfunction between the CC and the PMU, the CC changes from the WAIT mode to the READY IMAGE mode with predefined imaging default parameters and start imaging. In this event there will not be the STANDBY mode. Once communication with the PMU is established, the CC switches autonomously to the STANDBY mode. It remains in this mode until a new command from the PMU arrives. Fig.2 shows the CC mode state transition diagram.

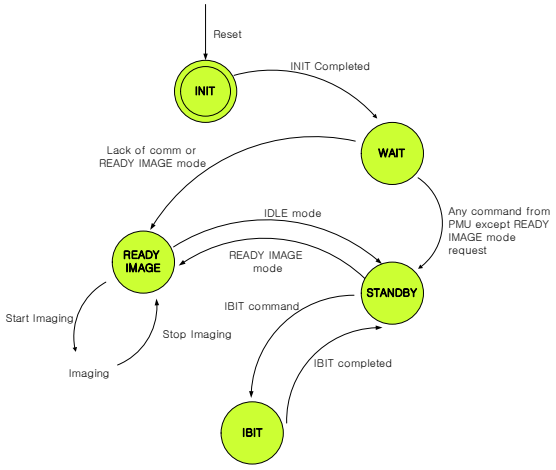

Fig. 4. Mode Transition Diagram

The INIT mode is the initial CC mode. The CC enters this mode once power is supplied to it. As this mode is transient mode, it ends upon completion of the initialization process. In this mode, the CC initializes software components and performs the PUBIT(Power Up BIT). Once the PUBIT is completed, the CC initializes RS-422 communication and switches current mode to the WAIT mode.

In the WAIT mode, the CC waits for any command from the PMU for 20 seconds. If no command arrives within this period, the CC will switch the current mode to the READY_IMAGE mode with pre-defined default parameters and set the START_IMAGING command. On any message received from the PMU the CC first switches to the STANDBY mode and then executes the requested message command.

Upon command from the PMU, or at the end of the IBIT mode, the CC enters the STANDBY mode. Main purpose of this mode is to economize power consumption and in the same time to be ready to enter the READY_IMAGE mode. In this mode, all bands are disabled, no telemetry of the FPEs is monitored and the periodic BIT is carried out.

The CC enters the READY IMAGE mode from the STANDBY mode upon the READY IMAGE mode command from the PMU, or from the WAIT mode with default parameters. It ends upon the STANDBY command from the PMU. In this mode, selected bands are power on, telemetry of activated bands is monitored, and the periodic BIT is carried out for the selected bands. And optionally the START IMAGING or the STOP IMAGING command will be requested according the camera's state. In the case of default imaging command the START IMAGING will be set automatically.

Upon the Initiated BIT(Built In Test) command from the PMU, the CC enters the IBIT mode from the STANDBY mode. Since all the FPEs are disabled, this

BIT activates selectively the FPE, performs telemetry readings and then deactivates the FPE. The BIT results are stored in the SOH table. The BIT results and the appropriate TLM data are transmitted.

2. Test Results

The stand alone CC is fully tested in the both point view of the hardware and the software. Fig. 5 and Fig. 6 show the test environment and the test software for the CC operation test.

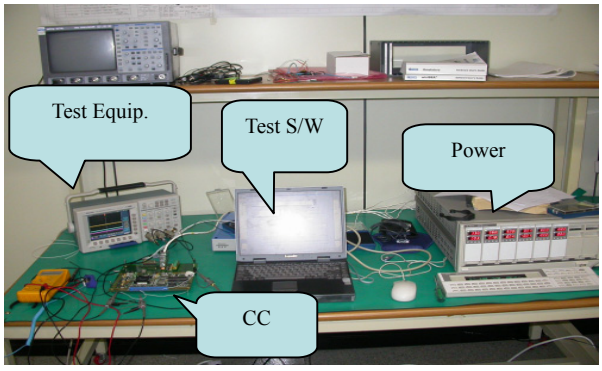

Fig. 5. CC Test Setup

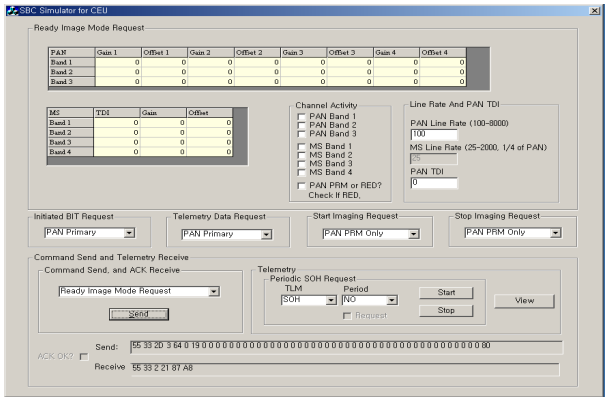

Fig. 6. Test Software for CC Test

This paper shows the serial communication prototype and the line sync implementation results as examples. The serial communication prototype between the CC and the FPEs should be required in order to set the TDI(Time Delay Integration), the gain and the offset and so on. Fig. 7 describes the serial communication prototype implementation result. In order to generate the line sync for the PAN channel and the MS channel, the CC receives the operation line rate from PMU. After receiving line rate, the CC converts the actual control value and transmits the line sync to FPEs by LVDS interface. The MS line sync should be generated once 4 PAN channel line syncs. Fig.8 shows the line sync implementation result. Except these tests, the all the functions of the CC are implemented and tested successfully.

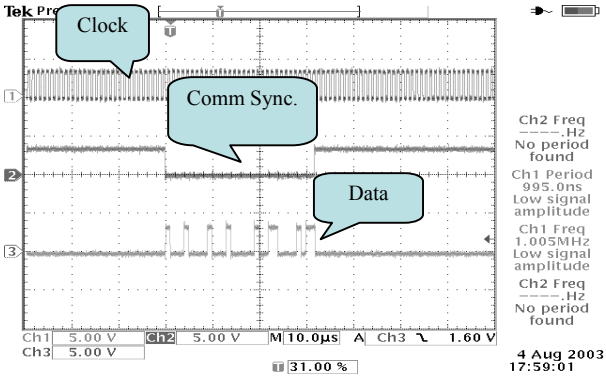

Fig. 7. Serial Comm. Test between CC and FPE

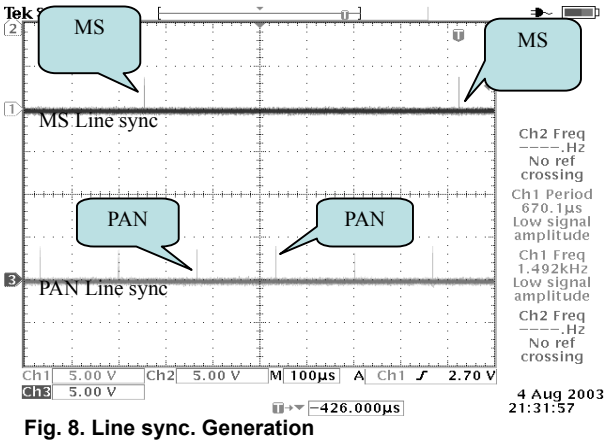

Fig. 8. Line sync. Generation

3. Conclusions

In this paper, the design concept and implementation results of the CC, which is responsible for controlling the PAN and the MS channel camera to take pictures of ground target, are shown in the hardware and the operation point of view respectively. In the hardware, the FPGA block was designed to play an important role in the FPE control and the system management and the various mode operations are considered for the low power consumption and the reliability in the operational software. This paper also shows that all the hardware including the FPGA block and the operation software were implemented and were tested successfully. As the next step, the integration between the CC and the FPEs should be carried out.

References

[1] Camera Controller Software Requirement Specification.
[2] Camera Controller Software Design Document.
[3] H. C. Park, 1994, VHDL, Seoul: Hansung Inc.
[4] A. P. Malvino, 1995, Electronic Principles, Los Angeles: McGraw-Hill.