Throughput Analysis of SBC for MSC on KOMPSAT-2

H.P.Heo
Optical Payload Department, Korea Aerospace Research Institute
45 Eun-dong Yusung-gu, Taejon 305-600, Korea
hpyoung@kari.re.kr

J.P.Kong, Y.S.Kim, J.E.Park, Y.J.Chang, S.H.Lee
Optical Payload Department, Korea Aerospace Research Institute
45 Eun-dong Yusung-gu, Taejon 305-600, Korea
(kjp123, yskim1203, pje, yjchang, shlee) @kari.re.kr

Abstract: The MSC is a remote sensing instrument with very high performance that is to be installed on KOMPSAT2 satellite. The MSC consists of EOS (Electro-Optic Subsystem), PMU (Payload Management Unit) and PDTS (Payload Data Transmission Subsystem). PMU controls and monitors all the other payload units by sending commands and collecting telemetry. PMU is in charge of interfacing between payload system and satellite bus system. PMU gets commands from ground-station via OBC (On-Board Computer) that is a main controller of the satellite bus system and sends telemetry to the ground-station via OBC. There is a processor module, called SBC (Single Board Computer) in the PMU. The SBC is a main controller of the MSC system. The main roles of the SBC are payload mission management, command validation and execution, telemetry collection and monitoring, ancillary data handling, event reporting, power control of payload sub-units and communication with these units. Intel's 80486DX2 processor has been used for the SBC. Due to the fact that the SBC plays important roles for imaging mission execution and handles a lot of control data that is required for payload operation, it is required to make analysis of the CPU load when it is in maximum operation mode. In this paper, the analysis and measurement results of the SBC throughput in the maximum operation mode.

Keywords: MSC, SBC, Mission, Throughput,

1. Introduction to SBC Software

The MSC is a main payload on KOMPSAT-2 satellite. MSC system consists of EOS (Electro-Optic Subsystem), PMU (Payload Management Subsystem) and PDTS (Payload data Transmission Subsystem), and PMU includes SBC. SBC incorporates Intel 80486 as a main processor. Embedded software on the SBC board communicates with OBC which is a main controller of the spacecraft, and manages all the sub-units of the MSC system. SBC software handles all the information and activities for the mission operation using VxWorks real-time operating system.

SBC software has several functions and capabilities that enable MSC system to perform imaging missions. SBC software receives and executes all the command messages from OBC through mil-std-1553B communication channel. These command messages contain a lot of information for MSC system to execute required imaging missions. SBC software sends SOH (State Of Health) data to OBC every second. This includes not only simple state of all the units but also the important analog telemetry like main secondary voltages and temperature. SBC software deals with mission execution. MSC system has several predefined missions like real-time imaging, simultaneous imaging and playback, memory playback, OBC data playback, and so on. Every unit should be configured properly according to predefined configuration order before executing a mission. Some missions that will be executed within at least coming 24 hours are stored in the SBC memory. SBC software manages several system data tables that are required by different units for their assigned functions. Most of them are to be stored in flash memory and these can be updated to RAM by ground station commands.

SBC software incorporates VxWorks as a real-time operating system in order to handle all the simultaneous activities. SBC software consists of four main tasks and several modules to deal with controlling commands and data for imaging and all the state of health telemetry data and to perform interface with the other MSC units. The main structure of SBC software is depicted in fig.1.

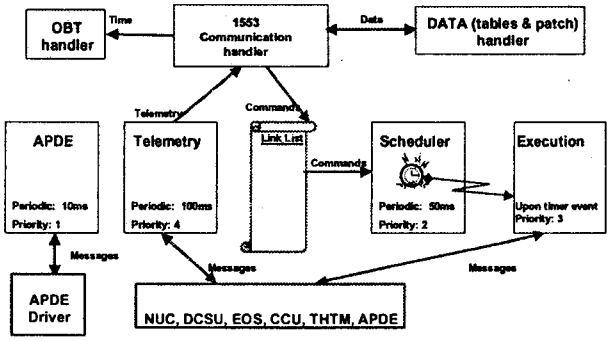

Fig.1. SBC Software Main Tasks and Main Modules

The 1553 communication handler receives command messages and parses them and transmits them to the appropriate task or module. Spacecraft OBC works as a BC (Bus Controller) and SBC works as an RT (Remote Terminal). The linked list module manages a command table that is sorted by execution time of each command by means of linked list data structure. Each command has its own execution time in half-second. A new command can be inserted into the linked list by the order of execution time and a command in the linked list can be deleted by request. Commands that should be executed in the next delta time can be fetched with respect to the current

time. Access to this linked list command table is controlled by mutex semaphore. The scheduler task is required for checking if there are commands that are to be executed and fetching such command from linked list command table. This task will be executed every 100 millisecond. Scheduler task sends each command to the execution task through message queue. The execution task waits for new command from message queue all the time. New command is executed as soon as it arrives. Most of the commands will be executed by sending them to the appropriate units through serial communication channels. Therefore, this task calls suitable communication module to send them. The spacecraft sends time-mark signal every second through discrete line and sends OBT (On-Board Time) message through mil-std-1553 communication channel every second. There are many system data tables required for mission executions. Data handler module gets parts of or whole table data that should be updated from mil-std-1553B handler module. SBC software supports mission execution. One mission script takes care of performing one whole mission and consists of a series of related commands, which are to be executed in the designated sequence, with a specified time interval between commands. Dozens of mission scripts are defined in advanced and stored to flash memory. Mission script can be activated only when they are copied to the linked list command table by the mission execution command from OBC. Mission execution command will include the index of mission script and the absolute time when the mission should be executed. Designated mission script will be copied to linked list command table and time interval between commands will be translated into the absolute time. The telemetry task gathers all the state of health data from all the units by receiving relevant message through serial communication channel. This task sends telemetry data to OBC every second according to the telemetry format table. The ATS (Antenna Tracking Software) task controls the x-band antenna to communicate with ground station. This task calculates the angle of antenna and sends it to APDE through serial communication channel every 10 milliseconds. In order to calculate the next position of the antenna, the information about the position of satellite in the orbit, the attitude of satellite and the feedback value of current antenna position are required. There are six different software modules to communicate with six different units (EOS, THTM, NUC, APDE, DCSU and CCU). Each communication module deals with transmitting commands and gathering telemetry.

2. Throughput Requirement & Estimation

SBC software and hardware are designed to comply with the CPU load requirements (throughput) of 50% when it is operating in the maximum system operation mode. The design and operational margin for CPU throughput will be the rest of 50%.

The CPU of the SBC will be utilized maximally when the MSC system is in mission execution mode. Therefore, CEU is in imaging mode, DCSU is in compression mode, DLS is in RF transmission mode, NUC is doing nun-uniformity correction and APDE is controlling the x-band gimbal antenna, called APS (Antenna Pointing System). In this operation mode, all the tasks in the SBC software will have maximum loads. The ATS task will receive the instantaneous position of the APS from the APDE and reconstruct the position of the next cycle according to the TPF (Tracking Parameter File) and generate PWM (Pulse Width Modulation) command for next cycle and transmit it to the APDE. All these ATS task activities will be accomplished periodically in every 10ms. The scheduler task and the execution task receive 384 bytes spacecraft ancillary data and add 128 bytes MSC ancillary data and transmit them to the NUC module every second. Reporting SOH to the OBC by the scheduler task will be continued as does in the normal operation mode. The telemetry task will have much more load because it has to collect a lot of telemetry data from all the operating sub-units, like DCSU execution reports, CCU pipe-line status and CEU status telemetry. A lot of analog telemetry needs to be examined and checked by the telemetry task (therefore, every 100ms) in the mission execution mode.

Table1. CPU Usage Estimation @CDR Phase

Software Task	CPU Usage per second
Scheduler Task	30 ms
Execution Task	30 ms
ATS Task	150 ms
Telemetry Task	40 ms
Interrupt, Kernel & etc	-
Total	250 ms

The requirement says that the CPU usage should be less than 50% in this mission execution mode. The estimation of the CPU usage has been performed at the software design phase (@CDR; Critical Design Review) considering the performance of the Intel's 40486DX2 at 25MHz clock speed. The estimated CPU usage per second was 25% as shown in the following table 1.

3. Throughput Measurement Results

Throughput measurement of SBC software has been performed using SBC EM and sub-units simulation software at the software verification phase. All the external interface of the SBC was fulfilled by the simulators. The CPU time has been measured using the Windriver's 'Tornado Browser'. As shown in the figure 2, the scheduler task uses 2 %, the execution task uses 1%, the ATS task uses 13% and telemetry task uses 19%. Even though the interrupt handler and the kernel use about 6%, it should not be considered as real usage because most of the interrupt requests are issued by the serial communication for the link between the host and the target.

Comparing with the estimated throughput performed at CDR phase, the measured result shows that the CPU

load of the telemetry task is very high (19%). It is caused by the fact that the amount of the telemetry from the sub-units is increased after CDR, in addition to that, important telemetry monitoring activities occupy the CPU relatively long time. On the whole, SBC software on the 80486 CPU of the SBC EM is working with the duty ratio of about 35%.

However, measured throughput using the 'Tornado Browser' might have much difference compared with the real situation because it is too much affected by the test environment, for example, the target server which is not flight software is involved somehow, the communication between the target and the host which is driven by 'interrupt' for each character reception/transmission and CPU time was checked by the software which might add some uncertainty.

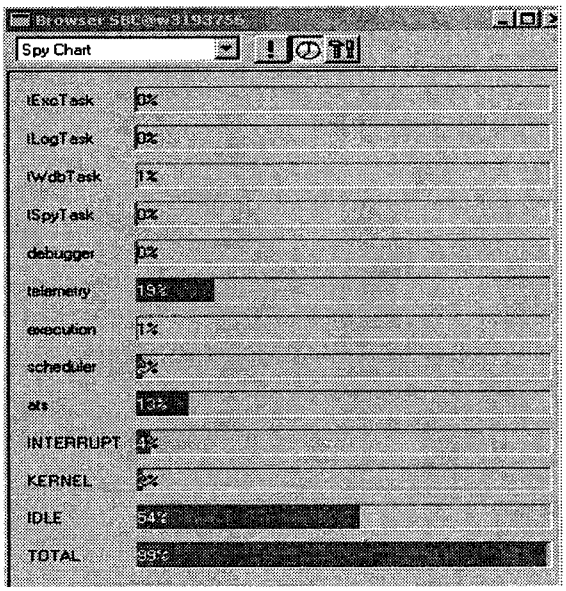

Fig.2. SBC CPU Time Measurements

Another way of throughput measurement has been achieved without using any test software. SBC EM was used for the measurement and exactly the same flight software was installed on it. All the external interfaces were simulated. In order to measure the throughput, the flight software has been slightly modified using code patch scheme. The CPU changes the status of a discrete signal before and after execution of some specific code of which we want to measure the CPU time. By measuring the transition of the discrete signal which will be driven by the software using an oscilloscope, the CPU time of specific task or module has been measured. In order to be very accurate in this measurement, special care has been taken because it needs to be guaranteed that no task switching should be occurred and no interrupt should be issued and no shared resource should be used within the code.

The CPU time of the telemetry task in one cycle (100ms) has been measured. Just after the telemetry task acquires the semaphore for synchronization before starting its work, it toggles the discrete hardware signal. After finishing its work in one cycle, before beginning to waits for the synchronization semaphore, it toggles the hardware discrete signal again. For the measurement at the maximum operating mode, the software was in the mission execution mode. A lot of sub-units telemetry were collected and monitored to check if each of them is in the normal range. In order to make the workload a little bit bigger, some erroneous telemetry that are being monitored by the telemetry task were injected by the sub-units simulation software.

The measurement result, as shown in the fig 3, says that 30% (30ms) of the CPU time was occupied by the telemetry task in each 100ms task cycle. The rest of the 70% will be idle or occupied by the other task.

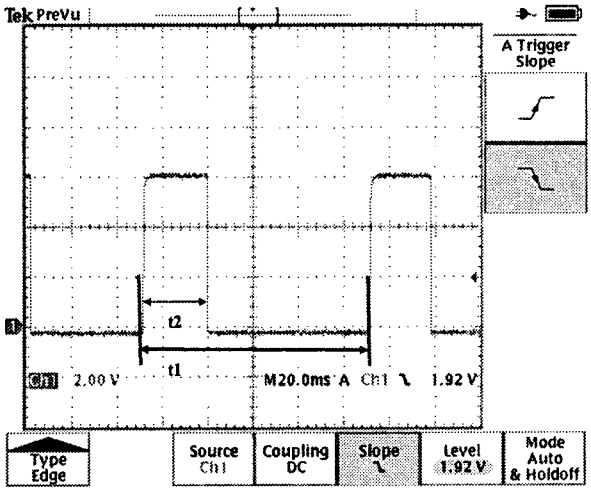

Fig.3. CPU Time of Telemetry Task (t1:100ms, t2:30ms)

The measurement was repeated after setting the software in the normal operation mode (not in the maximum operation mode). As shown in the fig 4, the CPU time of the telemetry task was reduced to about 22% (22ms) because the amount of the telemetry to be monitored was relatively smaller than in the mission execution mode.

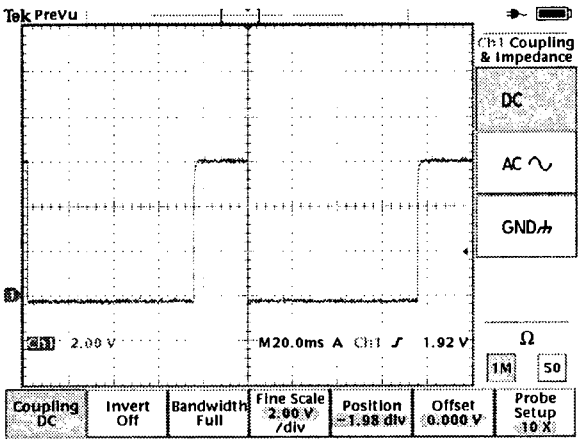

Fig.4. CPU Time of Telemetry Task (not in maximum mode)

By measuring the workload of most busy task (the telemetry task), the overall CPU time of the SBC software can be estimated based on the measurement using Tor-

nado Browser. Even in the worst case, considering the uncertainty of the measurement, the CPU will work not more than 60%.

4. Testing SBC using Popular Benchmark

SBC board has been tested using a few popular benchmarking codes in order to evaluate the performance of the hardware objectively. The CPU time has been measured by toggling the discrete hardware signal before and after execution the benchmark functions.

Ackerman's function has been run on the SBC EM. Ackerman's function is named after Wilhelm Ackermann, a mathematical logician who worked Germany during the first half of the 20th century. It is good for testing the efficiency of procedure calling mechanism. The measurement result of the Ackerman's function on the SBC EM says that it takes 42 ms, as shown in the fig 5.

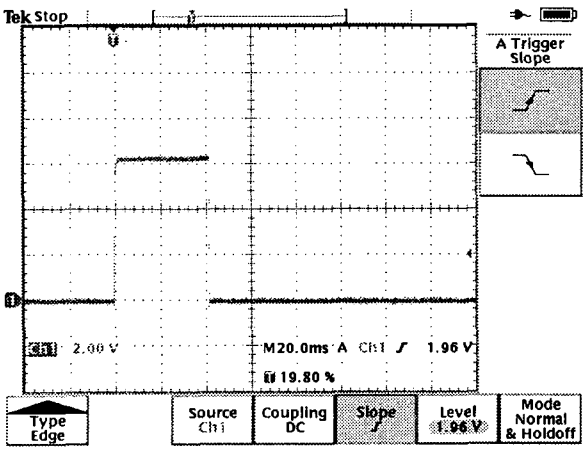

Fig.5. Ackerman Function (42ms)

The Sieve of Eratosthenes benchmark has been run on the SBC EM. This benchmark only measures integer CPU performance. The measurement result of the Sieve's function on the SBC EM says that it takes 1.6s, as shown in the fig 6.

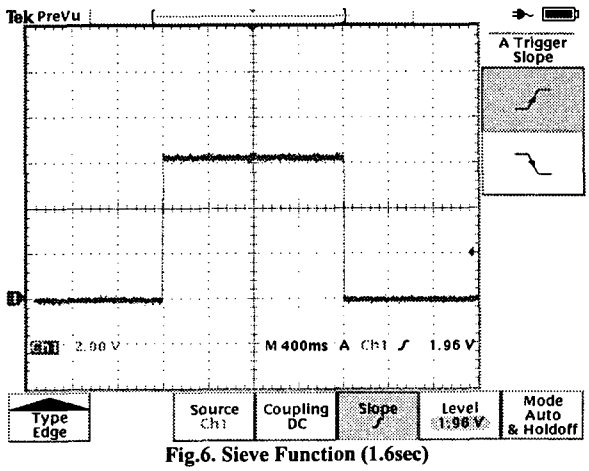

Fig.6. Sieve Function (1.6sec)

Above benchmark results has been used to compare that of another processor board which was developed for space application.

5. Conclusion

The workload of the SBC software in the maximum operation mode on the SBC EM board has been measured and evaluated to verify that SBC software and hardware has enough throughputs. It can be understand from the measurement result that the CPU of the SBC will work not more than 60% even in the maximum operation mode. It has about 40% performance margin. In addition to that, acquired benchmark results can be used to evaluate the throughput of the SBC compared with another space-born processor board

References

[1] MSC PMU Software Requirement Specification
[2] MSC Operation Requirement Specification
[3] MSC PMU Interface Requirement Specification
[4] Windriver Tornado User's Guide