SFOF Mark IIIA Central Processing System Model Development

H. S. Simon
SFOF/GCF Development Section

Simulation models are currently being used for SFOF development at the Jet Propulsion Laboratory. The results of two modeling studies are described that were performed during the early stages of the SFOF Mark IIIA Central Processing System development.

I. Introduction

Simulation models are currently being developed to support the design and implementation of the SFOF Mark IIIA Central Processing System (CPS). See Ref. 1 for a functional description of the SFOF.

SFOF Mark IIIA modeling studies began in May 1969 when it was learned that NASA would be providing JPL with IBM 360 model 75 digital computers to form the nucleus of the SFOF Mark IIIA CPS.

The IBM computer system simulator program was selected for model development activities since it runs on the IBM 360/75 and the program itself applies specifically to computer systems.

This article describes the results of two modeling studies that were performed during the early stages of SFOF Mark IIIA development. The first study, completed in August 1969, evaluated the performance of two real-time operating systems. The second study, completed in November 1969, provided information on central processing unit (CPU) utilization and response times during a simulated 24-h time segment in the orbital phase of the *Mariner* Mars 1971 dual spacecraft mission.

II. Performance Evaluation of Two Real-Time Operating Systems

To meet the requirement to provide test and flight support for the *Mariner* Mars 1971 mission, within the time allotted for the SFOF Mark IIIA CPS development cycle, it was deemed necessary to use an existing operating system with real-time data handling capabilities. The two systems selected for consideration were:

(1) The Goddard Real-Time System (GRTS).

(2) The Houston Real-Time Operating System (RTOS).

The objective of this study was to evaluate the relative performance of the two operating systems using simulation models, each being subjected to similar loading conditions.

A. Hardware Configuration

The final SFOF Mark IIIA CPS hardware configuration had not been established at the time of this study. Therefore, the preliminary configuration, shown in Fig. 1, was defined in both models.

B. Software Configuration

The model routines represented the capabilities within each operating system at the time the study was performed. Timings for the execution of each of the RTOS model elements were derived from the Statistics Gathering System measurements provided by NASA, Houston. Timings for the GRTS model were derived from instruction counts with an assumed execution speed of 1 μs per instruction.

Additionally, real-time data processors and analysis programs were included in each model.

C. Sequence of Events

To exercise the models, a script was developed defining the inputs that were applied in a time-sequenced manner. The script simulated real-time data from four missions using assumed data loading conditions. It also simulated requests from user area inquiry devices to turn on non-real-time programs.

Telemetry and tracking data for each mission were routed to the system as messages of 600 and 1600 bits, respectively, at a simulated real-time input rate of 40,800 bits/s.

Two streams of high-rate video data were input to the system for processing. The resulting full frame of TV data was made available for outputting to the Mission Test and Video System at a rate of 2,496,000 bits/s.

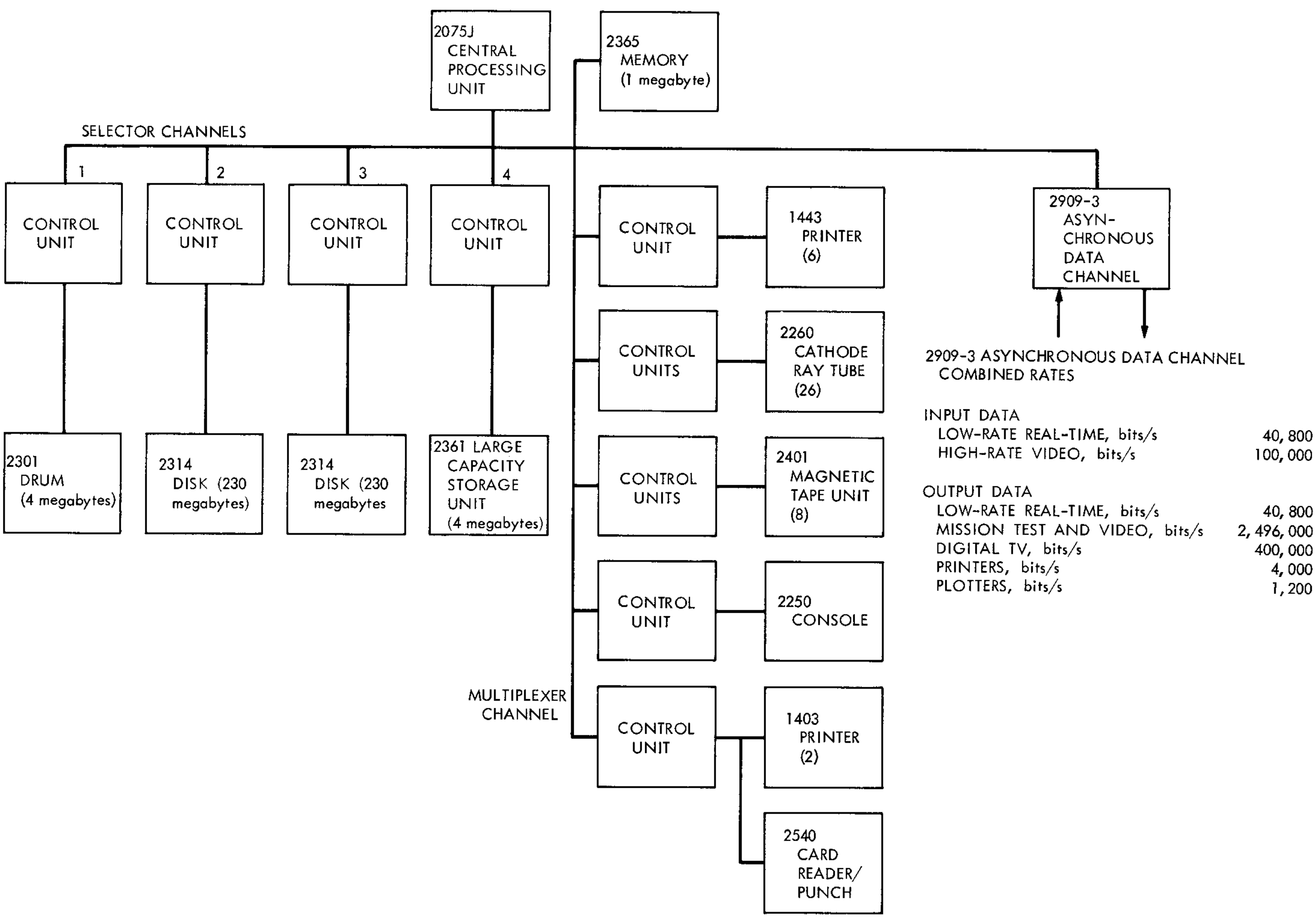

Fig. 1. Hardware configuration for performance evaluation of real-time operating systems

To realistically exercise the models, the script was written to impose a heavier than normal load on the systems. This was accomplished by turning on the following three large-scale computation-bound programs:

(1) Double-precision orbit determination program (DPODP).

(2) Star identification program (SIPM).

(3) Monitor criteria data program (MCDM).

These programs ran concurrently with the real-time data processors.

Additionally, the models were initialized to cause the following computation-bound programs to operate shortly after the start of the simulation runs:

(1) Orbit data generator program (ODGX).

(2) Engineering data records program (RECM).

(3) Communications measurements program (CMPM).

D. Results

Statistics obtained from two simulation runs are shown in Table 1. The statistics were derived from a 10-min computer run on the RTOS model and a 6-min run on the GRTS model. The script was deliberately set to give high CPU utilization to measure the effectiveness of each operating system.

Table 1. RTOS–GRTS comparison statistics

Utilization summary and task breakdown	RTOS	GRTS
CPU utilization summary (max values)		
Applications, %	87.2	69.0
Defined system tasks, %	0.5	12.6
GRTS supervisor service, %	—	12.6
Supervisor/mission and test video service, %	7.7	4.6
Memory interference, %	4.4	1.1
Total	99.8%	99.9%
Task breakdown		
System CPU utilization		
Interrupt input/output supervisor, %	2.0	5.3
Task management, %	2.8	4.4
Data management, %	2.7	25.2
Supervisor state CPU utilization (includes supervisor service requested by, and charged to, the application), %	10.2	35.0
Application		
Average response time for real-time task, ms	140	756
CPU utilization for real-time task (max value), %	2.4	6.3
DPODP CPU utilization (max value), %	75.8	58.2

E. Observations

In the RTOS model, approximately 88% of the available CPU time was spent processing application programs compared with 69% for the GRTS model.

RTOS required 8.2% of the available time to process system supervisor calls and to define system tasks compared to 29.8% for GRTS.

The most significant divergence between the two systems was in the area of data management. Here, RTOS required 2.7% of CPU time compared to 25.2% for GRTS. This significant difference was attributed to the data management concepts for each operating system.

The average task response time for the tracking and telemetry data was 140 ms for RTOS compared to 756 ms for GRTS. The major difference in response times between the two systems is the input/output time required by GRTS. In GRTS, real-time data is queued to a disk resident string before processing, while RTOS holds real-time input data in main memory until it is processed.

During the simulation runs, DPODP plus the system services that it calls received 75.8% of the CPU time under RTOS and 58.2% of the CPU time under GRTS. This means that DPODP would run approximately one-third faster under RTOS.

The total supervisor state utilization provides a measure of the total overhead of the system. For RTOS, the total system utilization was 10.2% compared to 35.0% for GRTS.

F. Conclusion

Based upon the assumptions made for these simulation runs, data was processed more efficiently under RTOS than under GRTS:

(1) Data management services required 9 times as much CPU time for the GRTS run.

(2) GRTS required 3.5 times more CPU time than RTOS to process system supervisor calls and to define system tasks.

(3) The navigation program (DPODP) ran 3 times faster under RTOS than under GRTS.

(4) The average real-time task response time was faster under RTOS by a factor of nearly 5.5.

III. *Mariner* Mars 1971 Simulation Study

The objective of this study, performed in November 1969, was to obtain CPU utilization and real-time task response time information from a simulation model that defined a preliminary SFOF Mark IIIA configuration. This information was obtained from several runs that exercised the model during a simulated 24-h segment of the orbital phase of the *Mariner* Mars 1971 dual spacecraft mission.

A. Hardware Configuration

The simulation model defined the preliminary hardware configuration shown in Fig. 2. The IBM 360 model 75J digital computer with 1 megabyte of main memory formed the nucleus of the hardware system. The configuration also included the following units:

(1) A multiplexer channel with nine control units for servicing the following set of assumed input/output devices: two 1443 line printers, two 1403-N1 line printers, eight 2401-1 magnetic tape units, 26 CRTs with keyboards, a 2250 console, and a 2540 card reader/punch.

(2) A selector channel with two control units for servicing two 2314 disk units (230 megabytes each), a 2361 large capacity storage unit (4 megabytes), and an electrical interface with the Univac 1108 computer.

(3) An asynchronous data channel with seven subchannels.

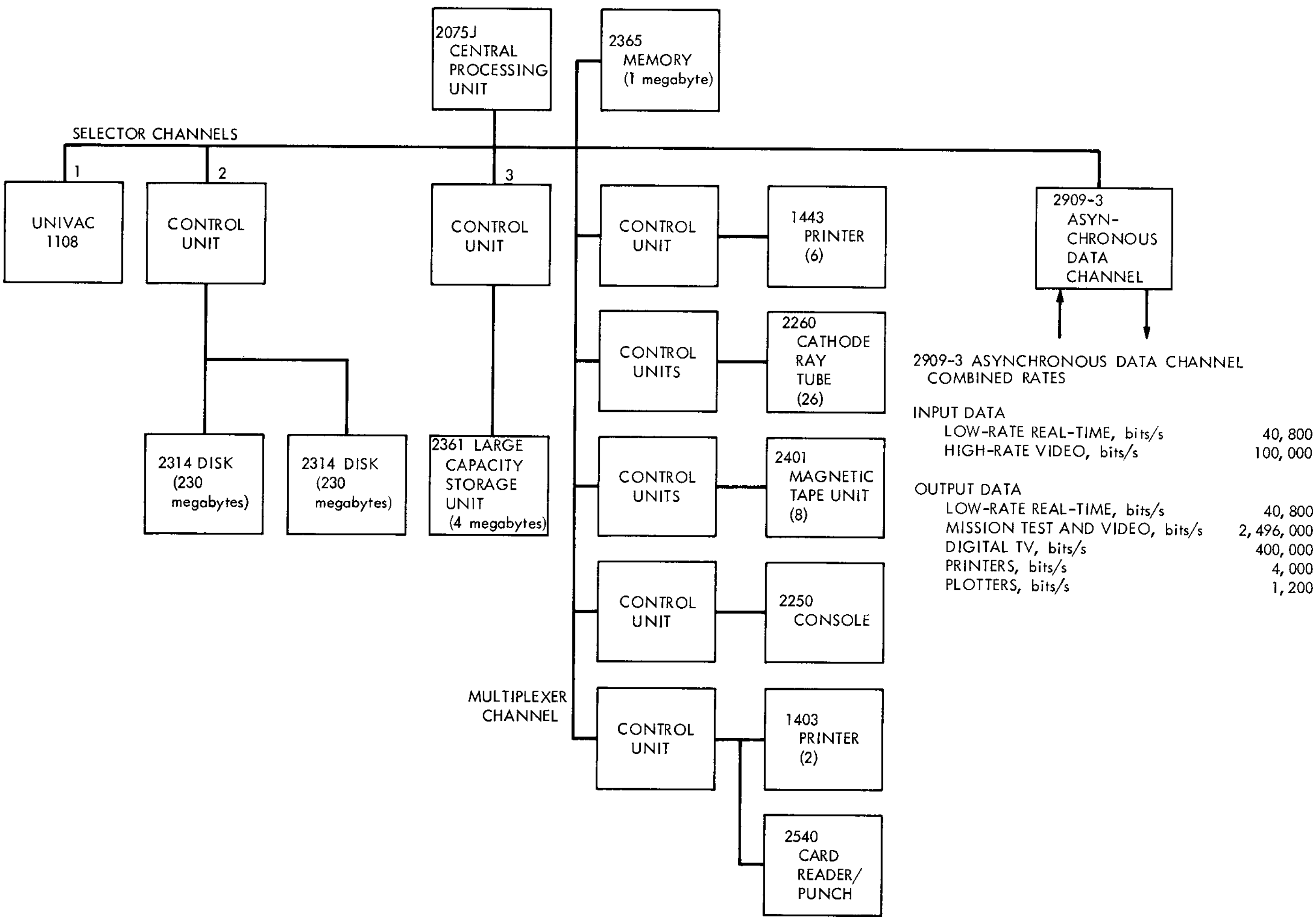

Fig. 2. Hardware configuration for *Mariner* Mars 1971 simulation study

B. Software Configuration

The model of the software system defined the capabilities of five functional software subsystems:

(1) Master control and user interface.

(2) SFOF command data.

(3) SFOF telemetry data.

(4) SFOF tracking data.

(5) SFOF monitor and operations control data.

The master control, or operating system portion of the master control and user interface, utilized the capabilities of Houston's RTOS version 11.0.

The user interface capabilities enabled each SFOF data software subsystem to interface with the operating system. In the model, all input data was routed directly by the operating system to the appropriate task and module for processing. Each task used the display formatting language for data conversion to the output device and the operating system's real-time access method to output the converted data.

The processing capabilities included in the model for the four remaining software subsystems (command, telemetry, tracking, and monitor and operations control data) are listed in Table 2.

C. Sequence of Events

The scripts used for this study were derived from a *Mariner* Mars 1971 project memorandum that described a suggested strategy to be used during the orbital phase of the mission. The period chosen was from 16:00 GMT, December 2, 1971, to 16:00 GMT, December 3, 1971. A mission profile corresponding to this 24-h period is shown in Fig. 3.

System utilization and response time information was obtained from two runs. All functions were performed in accordance with the scheduled, time-sequenced events as shown on the mission profile (Fig. 3).

Run 1. The entire 24-h period was simulated during which time all real-time data, including high-rate video, was logged and processed in the IBM 360/75 and distributed for display. Raw data was passed to the Univac 1108 as it was received.

Table 2. Processing capabilities of software subsystems

Type of data	Processing performed
Tracking	Data conversion Residual computation Store data for later use Display converted data Display residuals
Engineering telemetry	Frame synchronization Decommutation Data conversion Store data for recall Display frame synchronized data
Spectral science telemetry	Frame synchronization Decommutation Data conversion Store data for recall Display frame synchronized data
Video	Frame synchronization Decommutation Store data for recall Display video frame and/or plots
Monitor	Data conversion Store data for recall Display data

A peak utilization of the IBM 360/75 occurred at approximately 04:30 hours. During this time, the following streams of data were input to the IBM 360/75:

(1) Recorded science (8,100 bits/s).

(2) High-rate video (16,200 bits/s).

(3) Low-rate telemetry and tracking.

Run 2. The same 24-h period was simulated with identical input data streams as in Run 1. All processing was done in the IBM 360/75. In addition to the processing of all real-time data, the following science and navigation programs were turned on during the run:

(1) Spacecraft command generation program (COMGEN).

(2) Double-precision orbit determination program (DPODP).

(3) Planetary observation geometry and science instrument sequence program (POGASIS).

(4) Infrared interferometer spectrometer program (IRIS).

(5) Predicts (PRDX).

(6) Central computer and sequencer (CC&S) update.

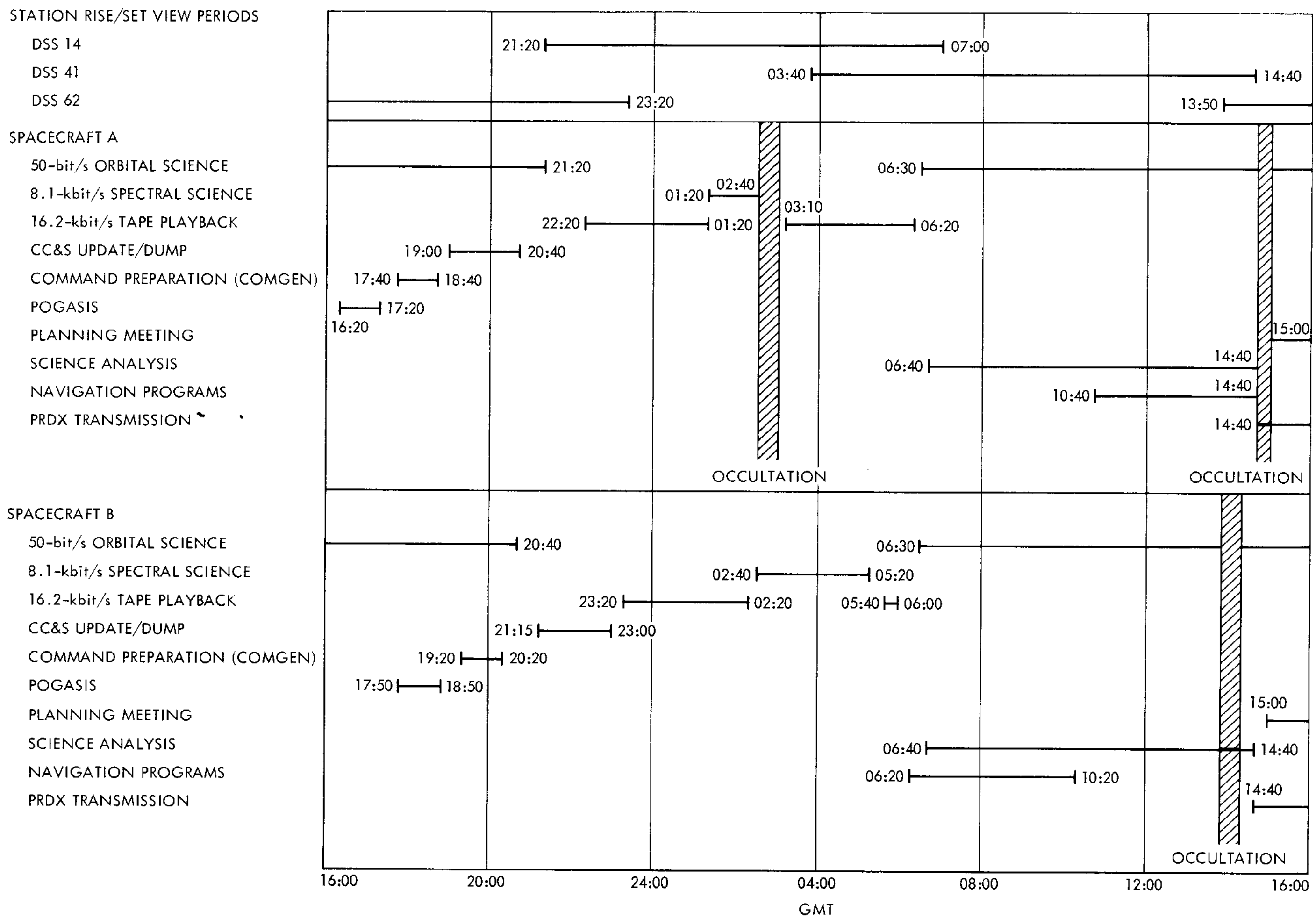

Fig. 3. Dual-mission profile (orbital phase)

Two scripts were utilized during Run 2 to evaluate segments of the orbital phase.

Script 1 covered the period from 16:00 to 23:00 hours, following the one-hour daily planning meeting. During the first three hours (16:00–19:00), POGASIS was run for each spacecraft. COMGEN was run for spacecraft A. During the next two hours, COMGEN was run for spacecraft B and the CC&S updates for Spacecraft A were transmitted to the site. During the final two hours (21:00 to 23:00), CC&S updates for spacecraft B were transmitted to the site.

Script 2 covered the period from 06:00 to 15:00 hours, prior to the planning meeting. During this time, the computation-bound navigation and science analysis programs for both spacecraft were turned on.

D. Results

The results of each run are shown graphically in Figs. 4 and 5. A summary of the utilization and response time statistics, for each run, is presented in Table 3.

E. Observations

During Run 1, the CPU utilization reached a peak value of 30.36% when the IBM 360/75 was required to simultaneously process (1) a stream of 8,100-bits/s science data, (2) a stream of 16,200-bits/s video data, and (3) streams of low-rate engineering telemetry and tracking data. The average CPU utilization over the 24-h period for all processing was 6.39%, while the average utilization required for processing low-rate, real-time data only was 2.45%.

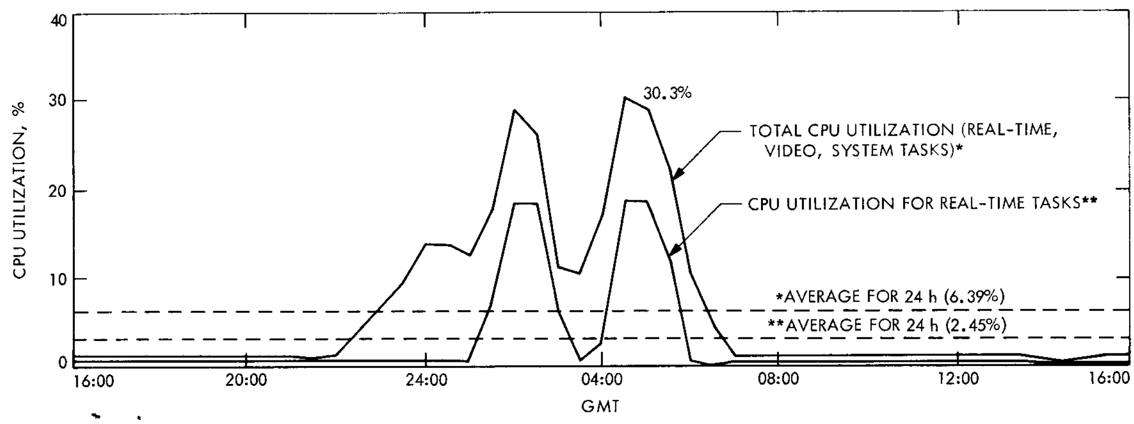

Fig. 4. CPU utilization during Run 1

During Run 2, the average CPU utilization for all jobs, over the 24-h period was 24.05%. An examination of average CPU utilization for various segments of the orbital phase reveals the following figures:

Post-planning meeting period	16:00–23:00	12.72%
Video and high-rate science playback period	23:00–06:00	14.88%
Science–navigation analysis program period	06:00–15:00	40.77%

The peak CPU utilization, which reached essentially 100%, occurred during the running of the computation-bound, science and navigation analysis programs for both spacecraft.

The average real-time task response time is defined as "the period of time starting with a real-time attach statement and ending when the task exits." This includes queue time, input/output time, and execution of the task. For this study, real-time tasks included telemetry processing, tracking residuals, and monitor data processing. Video data was excluded.

During the 9-h period that the science and navigation programs were running, the average real-time task response time was 58.680 ms. This relatively fast time demonstrated the efficiency of the operating system's task management function.

F. Conclusion

The results of this study showed that, during the orbital phase of the mission, the average CPU utilization over a 24-h period was 24.05%, leaving an average of 75.95% of the CPU for background processing. Assuming that DPODP will not be run in the IBM 360/75, the average CPU utilization should drop to less than 20%.

Table 3. Summary of utilization and response time statistics

Utilization and response time	Run 1	Run 2			Summary of all tasks (24-h period)
	Real-time and video processing (24-h period)	Post-planning meeting period (16:00–23:00)	Playback period (23:00–06:00)	Science–navigation period (06:00–15:00)	
Average CPU utilization, %	6.39	12.72	14.88	40.77	24.05
Real-time processing, %	2.95	1.01	6.80	1.06	2.97
Video processing, %	1.81	—	5.42	—	1.81
Total non-real-time processing, %	—	10.87	—	39.44	17.96
Navigation programs, %	—	10.87	—	15.80	9.09
Science programs, %	—	—	—	23.64	8.87
System, %	2.53	0.88	6.15	0.92	2.57
Average response time for real-time task, ms	79.779	42.615	164.165	58.680	82.316

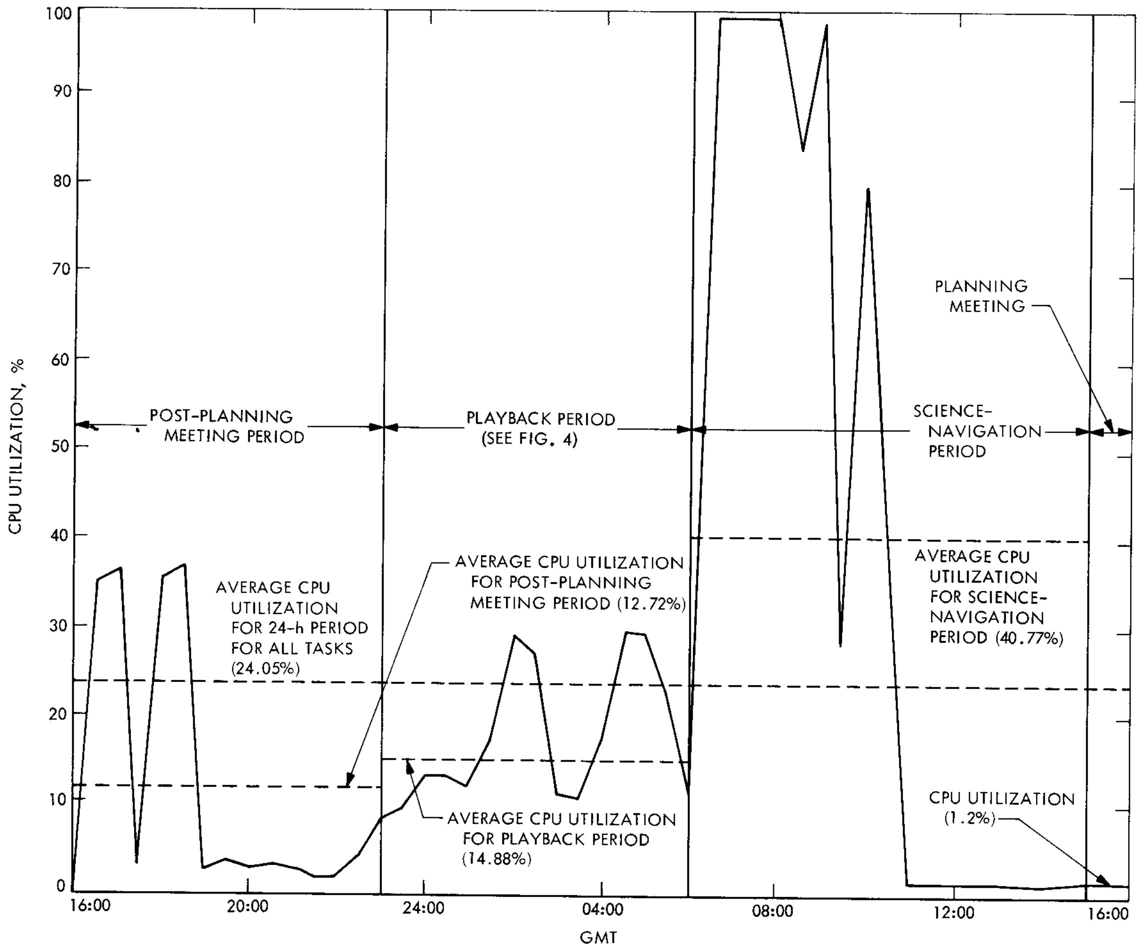

Fig. 5. CPU utilization during Run 2

Reference

1. Simon, H. S., "Functional Design of the Space Flight Operations Facility for the 1970–1972 Era," in *The Deep Space Network*, Space Programs Summary 37-66, Vol. II, pp. 90–94. Jet Propulsion Laboratory, Pasadena, Calif., Nov. 30, 1970.