A GROUND-BASED MEMORY STATE TRACKER FOR SATELLITE ON-BOARD COMPUTER MEMORY

Alan Quan, Robert Angelino, Michael Hill, Ursula Schwuttke, Felipe Hervias
Jet Propulsion laboratory
California Institute of 1 ethnology
301-270
4800 Oak Grove Drive
Pasadena, CA 91109-8099
email: alan@puente.jp l.nasa.gov

ABSTRACT

1 he 1 C) PEX/POSEIDON satellite, currently in Earth orbit, will use radar altimetry to measure sea surface height over 90 percent of the world's ice-free oceans. In combination with a precise determination of the spacecraft orbit, the altimetry data will provide maps of ocean topography, which will be used to calculate the speed and direction of ocean currents worldwide. NASA's Jet Propulsion Laboratory (JPL) has primary responsibility for mission operations for TOPE X/POSEIDON. Software applications have been developed to automate mission operations tasks. This paper describes one of these applications, the Memory State Tracker, which allows the ground analyst to examine and track the contents of satellite on-board computer memory quickly and efficiently, in a human-readable format, without having to receive the data directly from the spacecraft. 1 his process is accomplished by maintaining a ground-based mirror-image of spacecraft On-board Computer memory.

I. Introduction

1 he TOPE X/POSEIDON satellite contains an embedded computer system for the control of the satellite engineering subsystems and science instruments. The On-board Computer (OBC) provides the executive, command, telemetry, and attitude control processes, which are driven by variables contained in system tables in (X3C memory. Control of the satellite is performed primarily through uplink manipulation of the variables in these system tables. OBC memory also contains other types of information including a relative time sequence instruction buffer and flight software code. Proper spacecraft sequence generation requires knowledge of the current values in OBC memory at any given time. Direct examination of OBC memory requires uplinking a command to perform a memory dump, and the data that is transmitted back to the ground must be processed and converted to a readable format. This process is impractical because it is slow and the data is subject to corruption during transmission.

To track the memory state, previous JF'L missions maintained a duplicate image on paper of the current contents of OBC memory. These copies would be updated by hand based on update commands that had been sent to the spacecraft. The format was in octal or hexadecimal, and had to be translated manually into a meaningful format for the human reader (e.g., decimal engineering units for system tables or command mnemonics and parameters for flight software) whenever an analyst wanted to know what was in memory. Maintenance and translation of this paper version was a time-consuming and tedious process subject to frequent human error. To complicate matters, different mission operations teams would often maintain their own separate versions of the expected memory image, resulting in redundant and inconsistent information.

1 he 10PEX Memory State Tracker (MS1) was developed to provide an automated approach for examining, updating, and archiving the state of OBC rnernory. MS1 is a software system that maintains a ground-based mirror image of the expected contents of the non-dynamic areas of 1OPEX OBC memory. Non-dynamic areas are memory locations that can only be changed by commands sent from the ground. Accurate and timely maintenance of the ground image is essential; in order for the ground image to be useful, it must be identical to the expected OBC memory image.

MS1 **crenelates reoorts** of the contents of the around image in a variety of formats and displays them in an easy-to-use graphical user-interface. These reports provide status and attributes of system tables, flight software, and other memory contents. Previously analysts had to wade through three or four types of documentation and data and perform manual conversions to get the same information. 1 his required an average of 15 minutes of effort per OBC value. 1 he Memory State Tracker provides this information for entire tables of OBC values (containing dozens of memory words) within seconds. In the past, the ability to perform memory state analysis was a specialization. The Memory State 1 racker now allows this task to be performed by any mission analyst. In addition, the MS1 ground image provides a single common reference point for all interested parties, instead of relying on multiple inconsistent versions of the expected memory image.

This paper describes the implementation and functions of the TOPEX Memory State Tracker.

II. TOPEX OBC Memory Organization

TOPEX OBC memory consists of 16 banks of 4K words, or 64K words of memory. Each word contains 18 bits. Memory contains a variety of types of data, including system tables, command buffers, and flight software. F light software code is stored in a contiguous range of memory. Other types of data are not necessarily stored contiguously. The specific start and end addresses of the different data areas is determined when the final binary memory image is linked and loaded into the OBC. A link map is generated at that time, which contains the starting addresses and ranges of the data segments. 1 hus the data type of any 'word can be determined based on the word's absolute address. 1 he major data types are:

1. System tables - Spacecraft parameter tables,
2. Flight Software
3. Relative Time Sequence commands
4. Telemetry Monitor (TMON) - Special system tables that specify logical tests to be performed on telemetry and memory values

III._ MS'T memory organization

The MS1 ground memory image is an ASCII file containing one record for each address of OBC memory. Records are ordered by memory address. Each record contains the following fields:

1. address - a six-character octal memory address.
2. value - a six-character octal value which is the contents of the memory location.
3. size - an integer specifying whether the memory word is part of a multi-word system table variable. If the word is the first word of a multi-word variable, it also indicates the number of words in the variable.
4. memory type - an integer denoting the type of data stored at the location, and whether the location is trackable (non-dynamic).
5. update number - an integer telling the number of times the location has been updated by MST.
6. update time - a character string specifying the time of last update of the location.

Only the non-dynamic areas of OBC memory are maintained ("tracked") in their current form. MS1" allows the user to specify dynamic memory areas that are not tracked via an ASCII mask file. At the user's request a mask file is read and the locations specified as dynamic (non-trackable) are marked as such in the ground image. Non-trackable locations are not fully displayed in the MS1 reports and are not examined during MRO compare (see Memory Readout Compare below). Even though only the non-dynamic **areas** of OBC memory are tracked by MST, all 64K locations of memory are represented in the ground image for ease of organization and retrieval and to accommodate possible future expansion of the non-dynamic locations. 1 he ASCII format allows the image file to be portable across computers.

I-o initialize the ground memory image, MS1 uses three files generated during the course of OBC memory initialization. 1 hese files exist independently of MST and are not modified in any way to accommodate MST. A binary image file is used to initialize the contents of OBC memory. MST obtains the contents of the memory locations from this file. The link map, generated by the linker program when the initial binary image is created, is used by MST to determine the data type of each location. MST recognizes three types of memory locations: Relative Time Sequence Buffer (R1S), system table (including TMON), and flight software code. A global symbol list, generated by hand, is used by MST to determine the sizes of system table variables. With this information MST creates the 64K records of the ground image.

IV. Graphical User-Interface

The MST Graphical User-Interface (GUI) was created with ease of use in mind. The layout of the interface is designed to allow easy navigation into all functions provided by MST (See figure 1). Menus are not used because they hide information. Buttons are used to

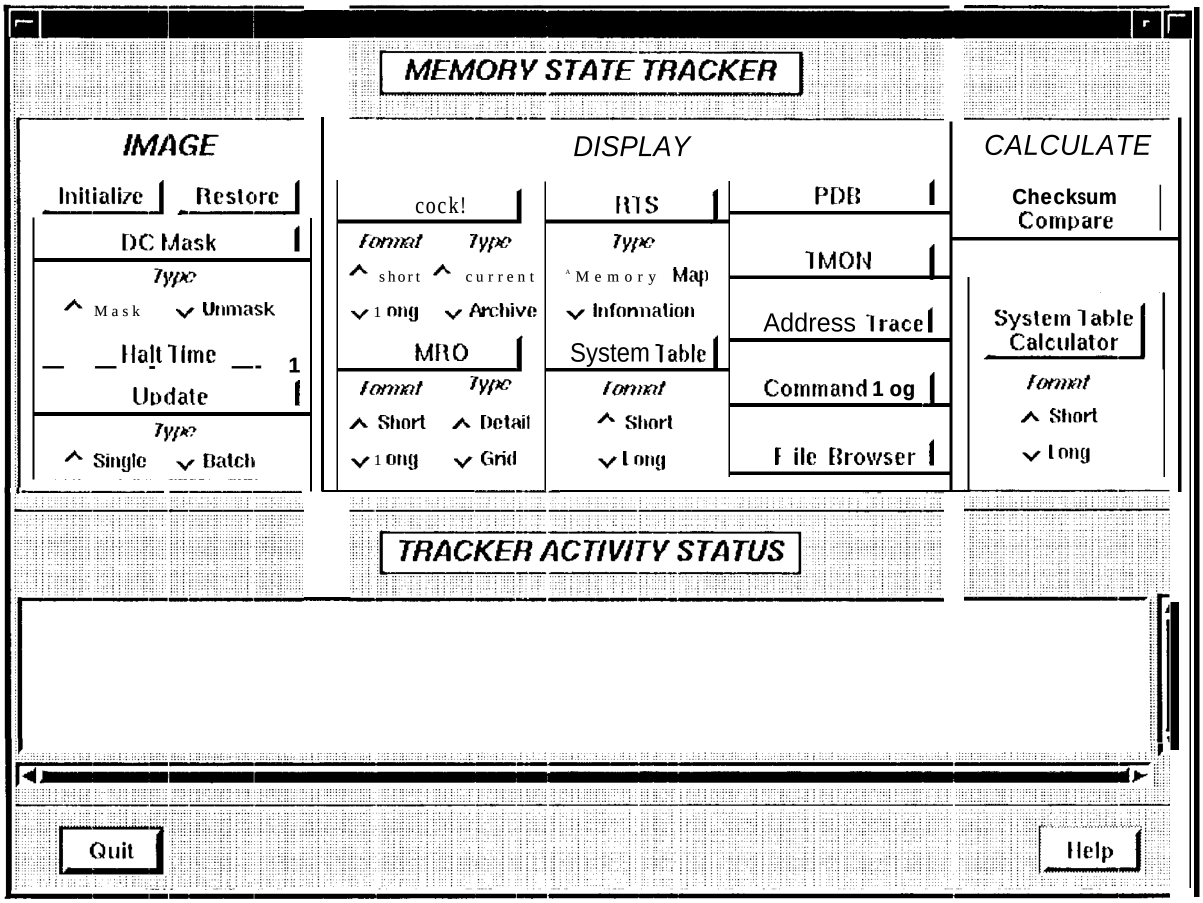
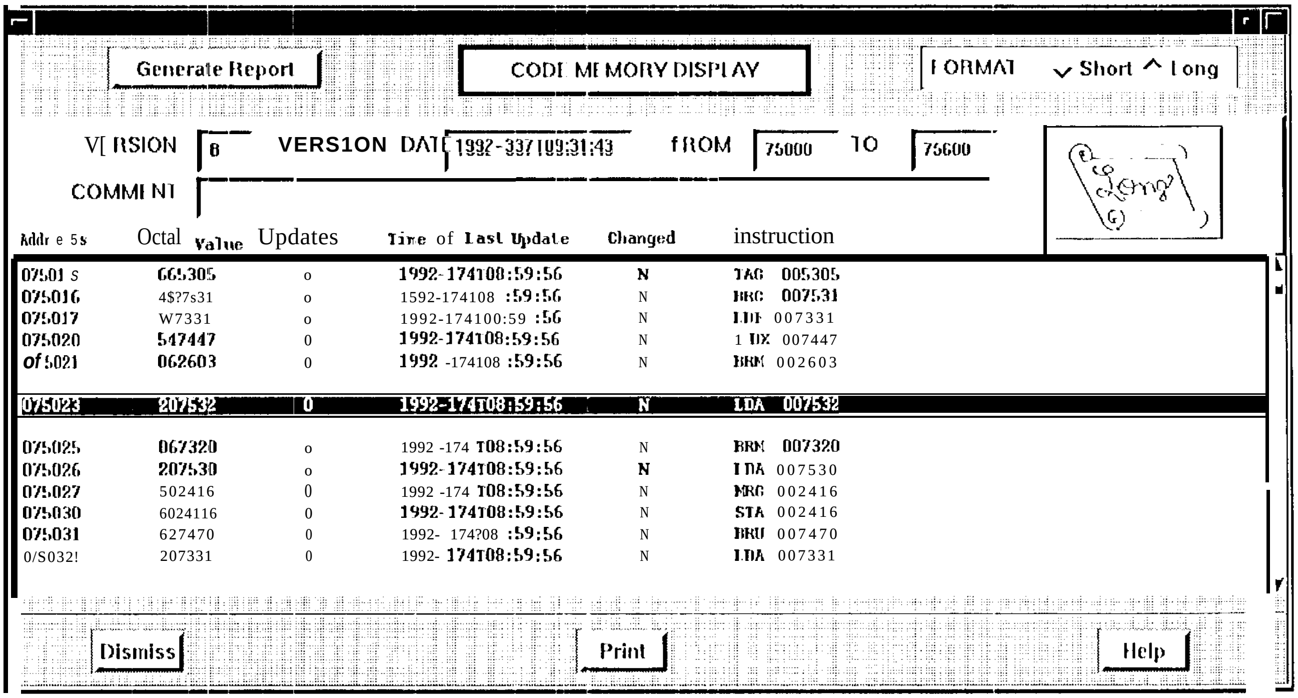


Figure 1 - Main MS1 window



V[RSION 8 VERS1ON DAT[1992-337109:31:43 fROM 75000 TO 75600

COMMI NT

Addr e 5s	Octal Value	Updates	Time of Last Update	Changed	instruction
07501 s	665305	o	1992-174108:59:56	N	TAG 005305
075016	4$?7s31	o	1592-174108 :59:56	N	BRC 007531
075017	W7331	o	1992-174100:59 :56	N	LDF 007331
075020	547447	0	1992-174108:59:56	N	1 DX 007447
of 5021	062603	0	1992 -174108 :59:56	N	BRM 002603
075023	207532	0	1992-174T08:59:56	N	LDA 007532
075025	067320	o	1992 -174 T08:59:56	N	BRM 007320
075026	207530	o	1992-174T08:59:56	N	IDA 007530
075027	502416	0	1992 -174 T08:59:56	N	MRG 002416
075030	6024116	0	1992-174T08:59:56	N	STA 002416
075031	627470	0	1992- 174?08 :59:56	N	BRU 007470
0/S032!	207331	0	1992- 174T08:59:56	N	LDA 007331

Figure 2- Code Display

denote the functions that are available. This allows all MST functions to be visible to the user in the main window. Since the buttons represent functions, they are grouped into three sections: the image section, the display section, and the calculate section. 1 he image section of buttons contains all functions dealing with manipulation of the ground image (initializing, updating, restoring, masking). 1 he display section of buttons contains all functions dealing with displaying the ground image in different formats, The calculate section of buttons contains functions which require calculations.

Color is used to denote functionality as well. Any field having a white background color is an editable input field. Any field with a medium blue background color is a non-editable display field. Any field with a light blue background color denotes a list containing user-selectable items. These lists allow the user to request information or functions on specific data items.

A Help button appears in every window to allow the user to request context sensitive help. Selecting the Help button will pop up an index window and a Help window. 1 he Help window contains a description of the contents and operation of the window in which the selected Help button appears. The index window contains a scrollable list of all Help categories. Selecting any category in the index window will cause the Help window to be loaded with the Help informaion on that category.

V. MST Functions

1 he TOPEX Memory State Tracker performs a number of functions that allow the user to update, view, and compare the contents of the ground memory image. 1 hese functions are activated by buttons in the MOTIF-based Graphical User-interface. The various MS1 functions are described below.

Update and Restore

The ground image is updated using the same commands that are used to update the spacecraft OBC memory. TOPEX commands are in ASCII format and consist of a time tag, a command mnemonic, and one or more numeric parameters. Command files contain header information, one or more commands, and a terminating string. The commands are ordered by time tag. 1 o update OBC memory, an ASCII command file is created, translated into binary, then uplinked to the satellite for execution by the OBC. MS1 uses this same ASCII command file to update the ground memory image.

Before executing the MS1 update function, the user specifies the "halt time" which indicates the time tag of the last command in the command file that will be executed on the ground image. This is necessary in order to ensure that a proper "snapshot" of OBC memory is available for a specific time. Next the user specifies one or more command files to be executed by MST. MST reads and parses the commands in these files, and executes the changes on the ground image. For each address specified in a command, the corresponding record of the ground image file is updated. 1 he value field is updated, the update number field is incremented, and the update time field is modified to match the time tag associated with the command. If the location to be updated is part of the flight software code, the entire code area of the ground image is first archived to a file before the location is updated. This allows all previous versions of the flight software to be retrieved and viewed at any time (see Code Display below). All commands performed on the ground image are echoed to a command log file.

1 he command log file serves a dual purpose. It provides a history of all executed commands and it is also used to restore the ground image to a previous state. To execute the restore function, the user specifies the previous time to which the image is to be restored. To restore the image to the previous time, first the current ground image is replaced by a copy of the initial baseline image which reflects the state of OBC memory before any updates were performed. l-he command log then serves as a command file which is executed by MST. All commands in the command log whose time tags are less than or equal to the restore time are executed on the ground image. This causes the ground image to be restored to the state it had at the specified restore time.

Code Display

l-he flight software code area of the ground image can be displayed to the user in two different formats. The user specifies the address range of the code to be displayed and can view either the current version of the code or a previous archived version. 1 he short format code report is simply a dump of the octal values of the specified code address range. 1 he long format report (see figure 2) shows the octal address, octal value, number of updates made to the location, time of last update, and whether the location has 'changed since the previous version of the code. 1 he last column of the report shows the original code mnemonic and operand(s) from which the octal value was translated. This symbolic information is obtained from a translation table which contains command mnemonics and their corresponding opcodes.

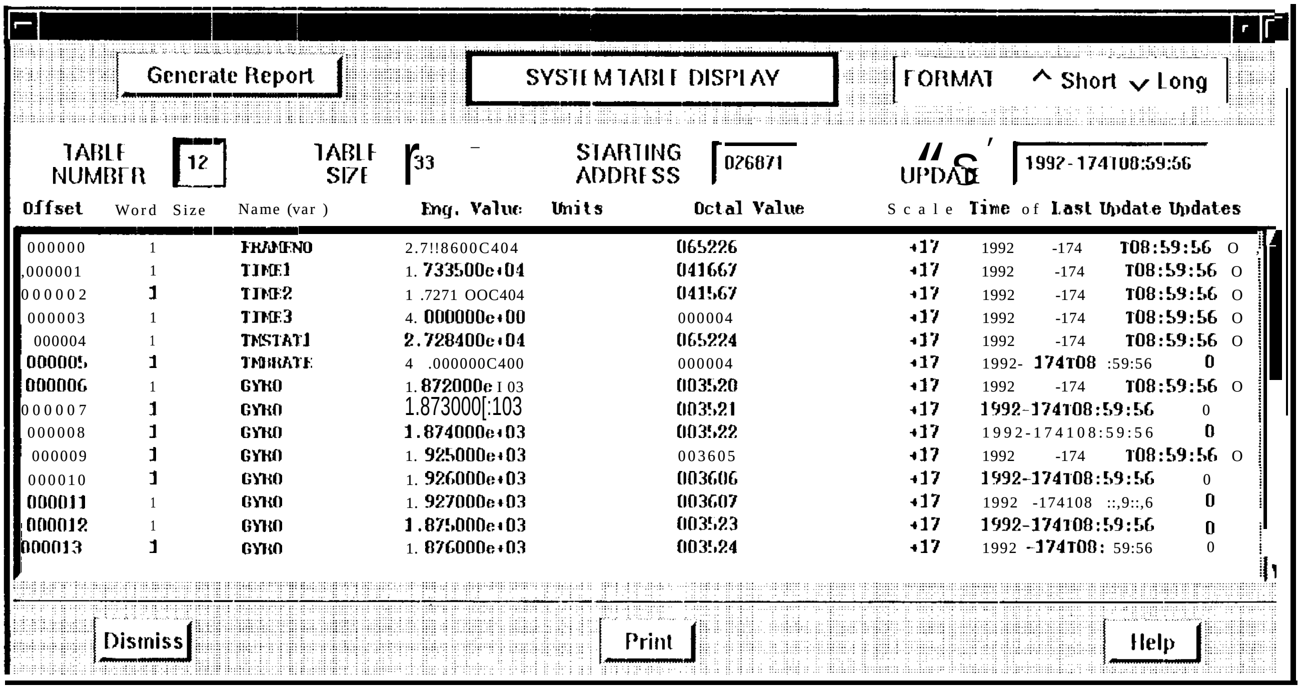


TABLE NUMBER 12 TABLE SIZE 33 STARTING ADDRESS 026871 UPDATE 1992-174T08:59:56

Offset	Word Size	Name (var)	Eng. Value	Units	Octal Value	Scale	Time of Last Update	Updates
000000	1	FRAMENO	2.7!!8600C404		065226	+17	1992 -174 T08:59:56	O
,000001	1	TIME1	1. 733500e+04		041667	+17	1992 -174 T08:59:56	O
000002	1	TIME2	1 .7271 OOC404		041567	+17	1992 -174 T08:59:56	O
000003	1	TIME3	4. 000000e+00		000004	+17	1992 -174 T08:59:56	O
000004	1	TMSTAT1	2.728400e+04		065224	+17	1992 -174 T08:59:56	O
000005	1	TMBRATE	4 .000000C400		000004	+17	1992- 174T08 :59:56	0
000006	1	GYRO	1.872000e I 03		003520	+17	1992 -174 T08:59:56	O
000007	1	GYRO	1.873000[:103		003521	+17	1992-174T08:59:56	0
000008	1	GYRO	1.874000e+03		003522	+17	1992-174108:59:56	0
000009	1	GYRO	1. 925000e+03		003605	+17	1992 -174 T08:59:56	O
000010	1	GYRO	1. 926000e+03		003606	+17	1992-174T08:59:56	0
000011	1	GYRO	1. 927000e+03		003607	+17	1992 -174108 ::,9::,6	0
000012	1	GYRO	1.875000e+03		003523	+17	1992-174T08:59:56	0
000013	1	GYRO	1. 876000e+03		003524	+17	1992 -174T08: 59:56	0

Figure 3- System Table Display

System Table Display

The 10PEX satellite engineering subsystems and science instruments are controlled by the OBC. The OBC is driven by parameters grouped into system tables in OBC memory. It is essential that ground analysts have complete knowledge of the current contents of system tables in order to direct spacecraft functioning. System tables are used by the analyst to update parameters to OBC processes. So for example, a new TOPEX ephemeris would require an update to the ephemeris table (TOPEX system table number 33).

MST maintains a copy of all non-dynamic system tables in the ground image. To display the current state of a system table from the ground image, the user specifies the system table number in the system table display window (see figure 3). A system table variable can occupy more than one memory word. 1 he system table report contains the following information for each variable in the system table:

1) offset within the system table

2) number of words (or addresses) occupied by the variable

3) variable name

4) decimal engineering unit value.

5) unit of measure

6) the 6-character octal values of all words contained in the variable.

7) the scaling factor (used to perform the conversion from octal to engineering units and vice versa)

8) the time of last update

9) the number of times the variable has been modified

1 he memory address of a system table is obtained from the system table map, which is generated when the OBC binary image is linked and loaded. The offset, octal value, time of last update, and number of updates are obtained from the ground image. The number of words in the variable, variable name, unit of measure, and scaling factor are obtained from a separate global symbol file called a global list. The global list contains this information for all non-dynamic system table variables. The engineering unit value must be calculated based upon the octal value and the scaling factor, which specifies the position of the decimal point. Information about the system table itself, including the starting address of the table, the number of words in the table, and the time of the last location updated in the table is also displayed.

Dynamic system tables (whose values can be changed by the OBC without a ground command) are not fully reported. This is because the global list does not contain information for variables from dynamic system tables. If a dynamic system table is selected for display, only the information available from the ground image (off set, octal value, time of last update, and number of updates) will be displayed. All other fields are filled with X's, to indicate that the table is dynamic.

Memory Readout and Compare

Portions of TOPEX OBC memory can be downloaded from the spacecraft and viewed in octal form in a special report called the Memory Readout File (MRO). The MRO is generated by the ground spacecraft team, not by MS1. MS1 can read the MRO and compare the portion of memory contained in it to the same memory locations in the ground image and report any discrepancies (miscompares). This process is useful for confirming that the ground image is in fact identical to OBC memory. Any memory locations contained in the MRO which are dynamic (non trackable, as specified by a mask file) are not examined during the comparison.

To give the user an overall view of how the MRO compares to the corresponding memory range, MST can display a graphical representation of the miscompares throughout the entire 16 bank, 64K memory image (see figure 4). Each bank is represented by a button, with 16 buttons arranged in a grid. The label of a button denotes the bank number, and the color of a button denotes the percent miscornpares that were found in that bank during the MRO compare. Green indicates no miscompares, while red denotes between 75 and 100 percent miscompares. Yellow, amber, and orange correspond to 1-24, 25-49, and 51-74 percent miscompares respectively.

When the user selects one of the buttons, a second display appears which shows a breakdown of the percent miscompares within the 4K bank represented by the button (see figure 5). Each button in this display corresponds to a 64-word range within the bank, with a total of 64 buttons. Each button is labelled with the octal starting address of the 64-word range. The color coding scheme for the buttons is the same as in the bank grid display.

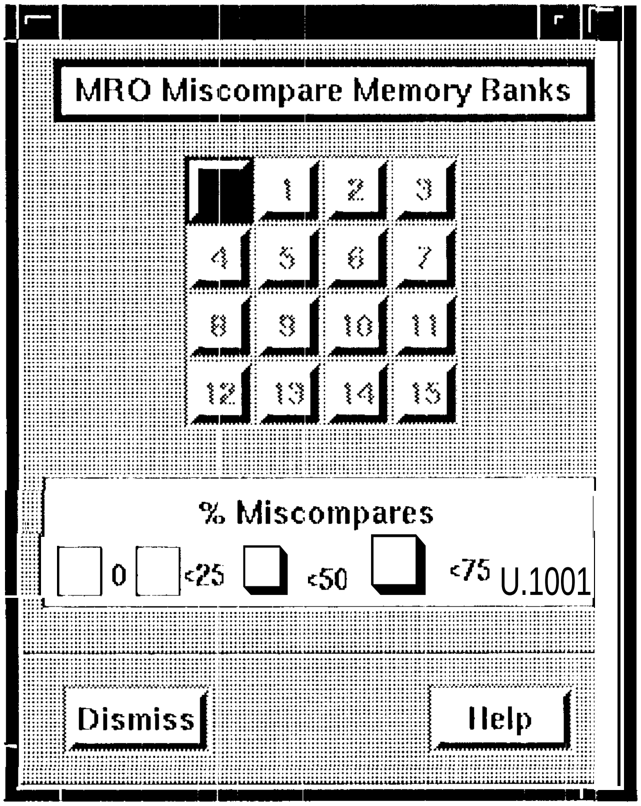
To display the 64 word range that a button represents, the user selects the button corresponding to that range. This causes a third display to appear (see figure 6) which shows in tabular form the memory address, the MRO and ground image values at that address, the octal difference between the two values (if any), and the type of data at that address (if it is one of the types that MS? tracks), This hierarchical data display and navigation enables the analyst to quickly ascertain and examine any miscompares which could potentially be a problem.

Relative Time Sequence (RTS) Display

Relative Time Sequence (RTS) commands are fixed sequences of commands that carry out functions which must be performed repeatedly. RTS's can be activated from the ground via real-time commands, by the flight software, or by Telemetry Monitor (TMON) reaction commands. An RTS command consists of 3 memory words: the time delay to wait before execution of the command in the sequence and two words containing the command to be executed. RTS commands are kept in an area of memory called the RTS buffer, which consists of 3 sets of contiguous 512 memory words, or 512 RTS commands. OBC memory can hold a maximum of 73 RTS's. Also contained in memory is an RTS index which has 6 words per RTS. The index words contain the starting address of the W-S in the RTS buffer, the number of commands allocated for the RTS, the number of commands actually loaded for the RTS, and the remaining three words are dedicated to the run-time status of the RTS. Access and modifications to any RTS requires consulting the RTS index to find the correct memory location and size of the desired RTS. In order to know which RTS's are loaded and their sizes and locations one must examine and decipher the RTS Index. To know what commands are in an RTS, one must read the RTS buffer and translate the numeric time delay tags and commands into mnemonics.

Since RTS'S can be loaded or overwritten, but not deleted, fragmentation can occur, because if an existing RTS has to be extended beyond the number of commands originally allocated for that RTS, the RTS has to be moved to the end of the RTS buffer and all the locations it occupied previously have to be marked as abandoned.

MS1 allows update and fast translation and display of the RTS buffer. There are two displays for this purpose: a memory map of the RTS buffer and a detailed display of all loaded RTS'S. The RTS memory map graphically displays available space in the RTS buffer (see figure 7) based on information in the RTS index. Empty command slots are denoted by blank spaces, periods denote loaded commands, plus signs denote allocated commands, and the capital letter 'X' denotes abandoned spaces. This information is impor-



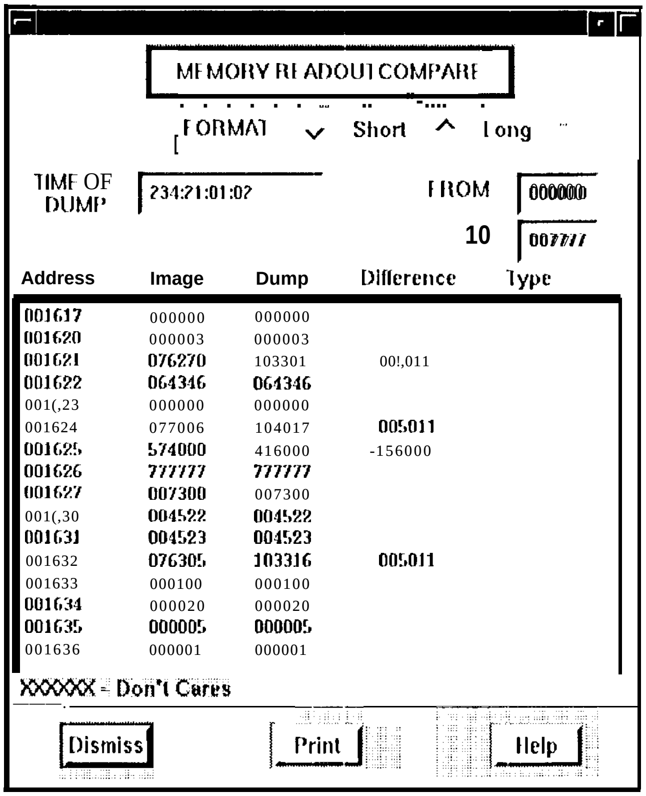
MRO Miscompare Memory Banks

■	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

% Miscompares

□ 0 □ <25 □ <50 □ <75 U.1001

Figure 4- MRO Bank Grid



MEMORY READOUT COMPARE

FORMAT ✓ Short ^ Long

TIME OF DUMP 234:21:01:02

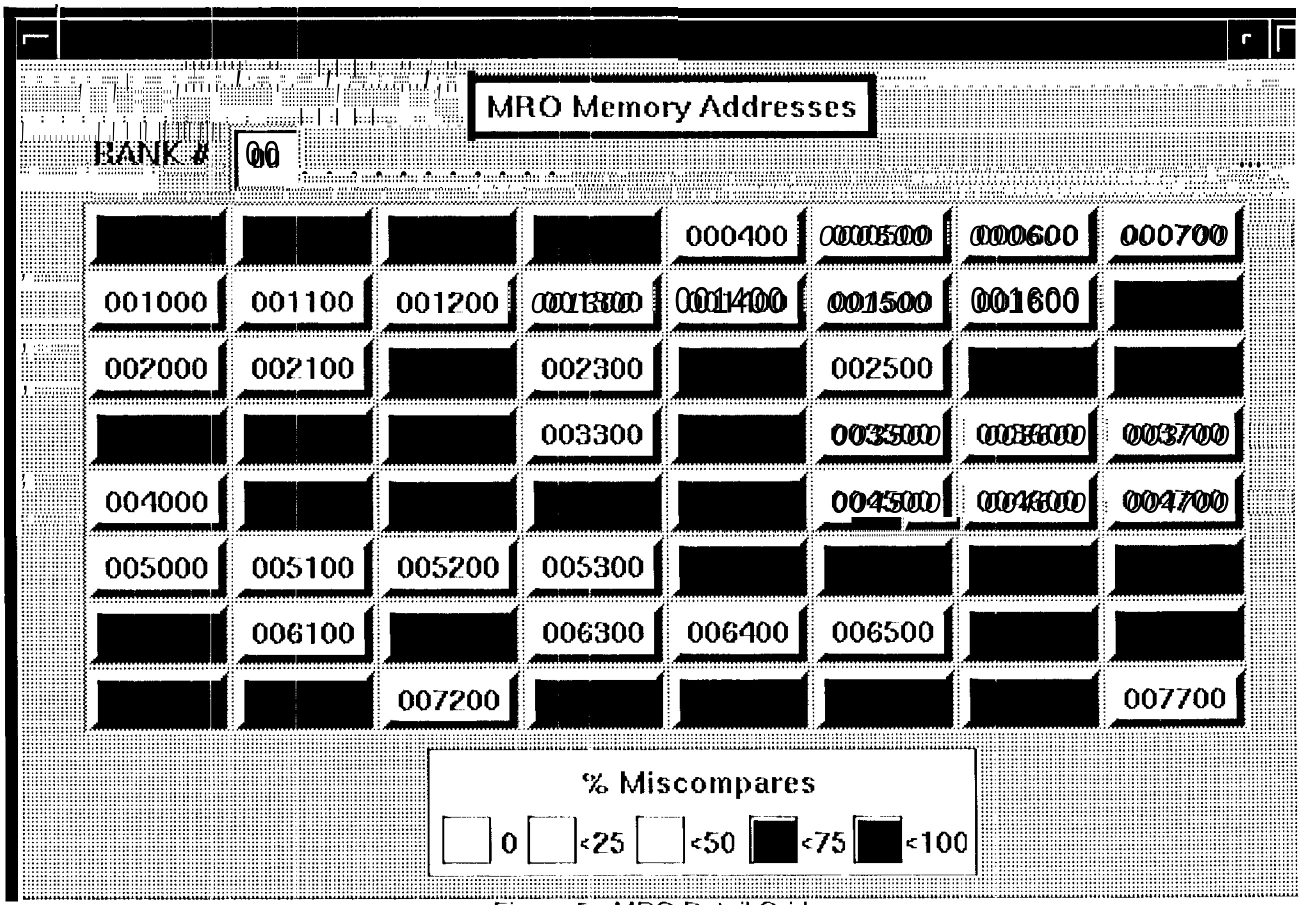
FROM 000000

10 007777

Address	Image	Dump	Difference	Type
001617	000000	000000		
001620	000003	000003		
001621	076270	103301	00!,011	
001622	064346	064346		
001(,23	000000	000000		
001624	077006	104017	005011	
001625	574000	416000	-156000	
001626	777777	777777		
001627	007300	007300		
001(,30	004522	004522		
001631	004523	004523		
001632	076305	103316	005011	
001633	000100	000100		
001634	000020	000020		
001635	000005	000005		
001636	000001	000001		

XXXXXX = Don't Cares

Figure 6 - MHU Detail Report



MRO Memory Addresses

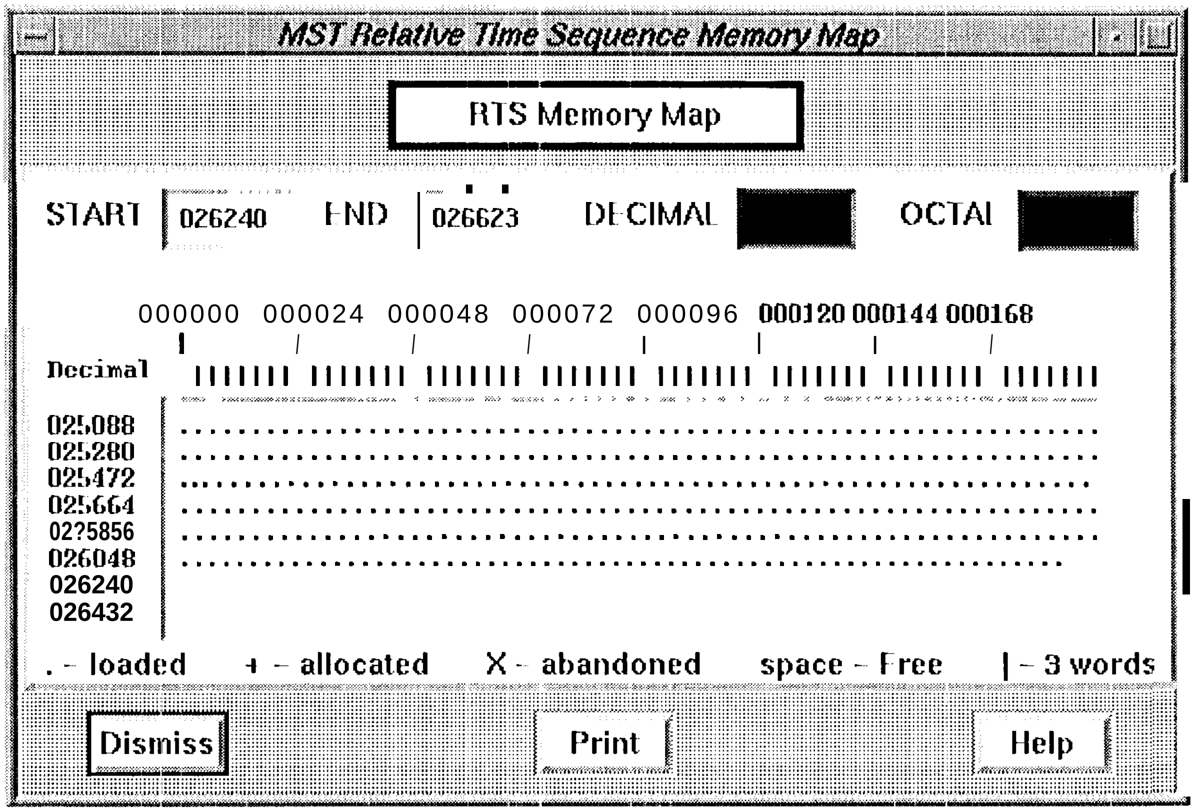
BANK # 00

■	■	■	■	000400	000500	000600	000700
001000	001100	001200	001300	001400	001500	001600	■
002000	002100	■	002300	■	002500	■	■
■	■	■	003300	■	003500	003600	003700
004000	■	■	■	■	004500	004600	004700
005000	005100	005200	005300	■	■	■	■
■	006100	■	006300	006400	006500	■	■
■	■	007200	■	■	■	■	007700

% Miscompares

□ 0 □ <25 □ <50 ■ <75 ■ <100

Figure 5 - MRO Detail Grid



MST Relative Time Sequence Memory Map

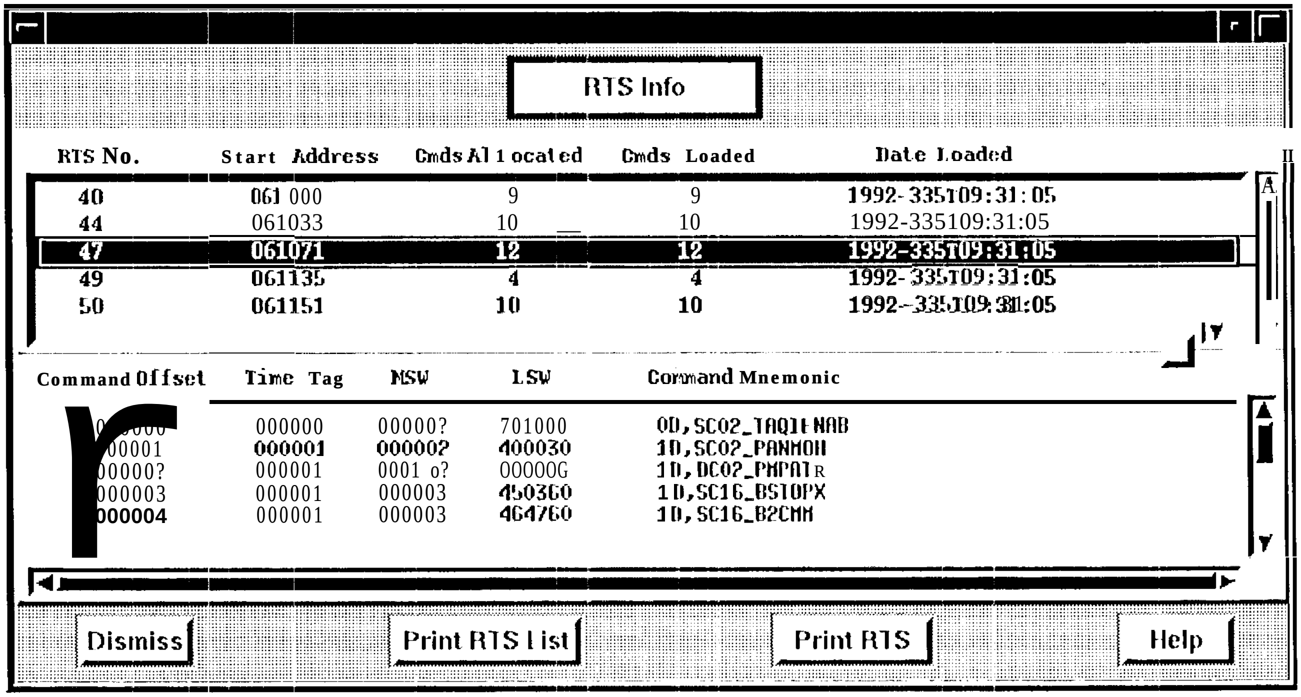
RTS Memory Map

START 026240 END 026623 DECIMAL OCTAL

```
            000000  000024  000048  000072  000096  000120 000144 000168
Decimal
025088 ............................................................
025280 ............................................................
025472 ............................................................
025664 ............................................................
02?5856 ...........................................................
026048 .........................................................
026240
026432
```

. - loaded + - allocated X - abandoned space - Free | - 3 words

Figure i' - RTS Memory Map



RTS Info

RTS No.	Start Address	Cmds Al 1 ocat ed	Cmds Loaded	Date Loaded
40	061 000	9	9	1992-335T09:31:05
44	061033	10	10	1992-335109:31:05
47	061071	12	12	1992-335T09:31:05
49	061135	4	4	1992-335T09:31:05
50	061151	10	10	1992--335T09:31:05

Command Offset	Time Tag	MSW	LSW	Command Mnemonic
0 000	000000	00000?	701000	0D,SC02_TAQ1ENAB
00001	000001	00000?	400030	1D,SC0?_PANMOH
00000?	000001	0001 o?	00000G	1D,DC0?_PMPA1R
000003	000001	000003	450360	1D,SC16_BSTOPX
000004	000001	000003	464760	1D,SC16_B2CMM

Figure 8- RTS Detail Report

tant because the analyst has to know if there is enough space to load new RTS's (or change an existing one) and when to unfragment the R1S buffer because there are too many abandoned command slots. Unfragmentation involves clearing the R1"S index and buffer, and re-loading all RTS's.

1 he detailed display shows the contents of all loaded RTS's (see figure 8). The window is split into two parts: the top section contains a list of all R1S's and the bottom section contains the commands that make up an individual RTS. Each entry in the RTS list has the following information: the RTS number, the RTS buffer starting address, the number of commands loaded, the number of commands allocated, and the date the RTS was loaded.

1o obtain information about a particular RTS, the user mouse-selects the desired RTS in the list. MST then retrieves the R1S from the R1S buffer area of the ground image and displays it in the lower area of the RTS report window. This detailed information includes the command offset, the time delay tag, the most significant command word, least significant command word, and the original command mnemonic. The time tag and command words are in octal. 1 he command mnemonic is not contained in the ground image. It is obtained from a separate ASCII file associated with the RTS that is created when the RTS is initially loaded into the ground image via the MS1 update function.

TMON Report

1 he 1MON or 1 elemetry MONitor consists of a group of special system tables that are used to monitor the values of spacecraft telemetry and OBC memory locations. These values are critical to the operation of the spacecraft and thus must be monitored regularly. MST allows the user to view formatted reports of these 1MON system tables as they exist in the ground image. The tables define 1 MON "groups", which specify the locations of data values, logical tests to be performed on these values, and actions to be performed if the data fails the tests. This information in its raw form in the system tables is not readily comprehensible by the human user. MST extracts the information from the ground image and translates it into a human-readable form. If the action to be performed upon test failure is a Relative 1 ime Sequence of commands (RI-S), MST retrieves the specified R1S from the ground image and displays it following the T MON group that invokes it.

Memory Address Trace

MS1 provides an address trace function that allows the user find out the data type of any address. 1 he user simply specifies an address and MST displays the data type associated with that address (system table, code, R1 S). If the address is in a system table or code, an appropriate display window appears with the system table or code segment that contains the address.

For example, if the user enters the address of a code location, MS1 will retrieve the record from the ground image associated with that address. The type field will indicate that the data type is code, which will be displayed to the user in a message in the main window of the GUI. Next MST will consult the link map file to determine which flight software module this address is contained in, and load the source code for that module into a file browser for viewing by the user. If the location is of type system table, MST will display the system table window with the system table that contains the address.

HyperText Navigation

1 he code (long format only), system table, and MRO detailed reports display items (addresses or variables) in a list format. 1 he user can obtain more information about any item listed in the reports. When the user selects an item by clicking on it with the mouse, a file browser appears that displays the software module which contains the address (for code addresses) or defines the system table variable (for system table entries).

For example, if the user is viewing the detailed report of the MRO compare and notices that a few locations of type code or system table do not match with the ground image, he/she can select one of these locations with the mouse to obtain more information about it. If the location is of type code, MS1 determines the software module which contains the location and loads the source code of the module into a file browser for display to the user. If the location is part of a system table variable, MS1 loads the software module in which the system table variable is defined.

This feature provides information about the flight software context of a memory address in a matter of seconds. Performing this lookup manually would take significantly longer.

Checksum

1he TOPEX OBC calculates a checksum of the code area of memory and sends it in telemetry on a regular basis. MS1 can also calculate a checksum of the code range of the ground image and compare it with the OBC generated checksum. When the user invokes the MST checksum function, MST first reads the OBC checksum from a TOPEX report generated from telemetry. This is a standard report used by the Spacecraft Performance Analysis Team (i.e., not specially tailored to MST). Next MST calculates the octal checksum of the code range of the ground image, and then connpare the two values to see if they match. The result of the comparison is displayed to the user. Comparison of OBC and MST checksums is another way to verify that the ground image is identical to OBC memory.

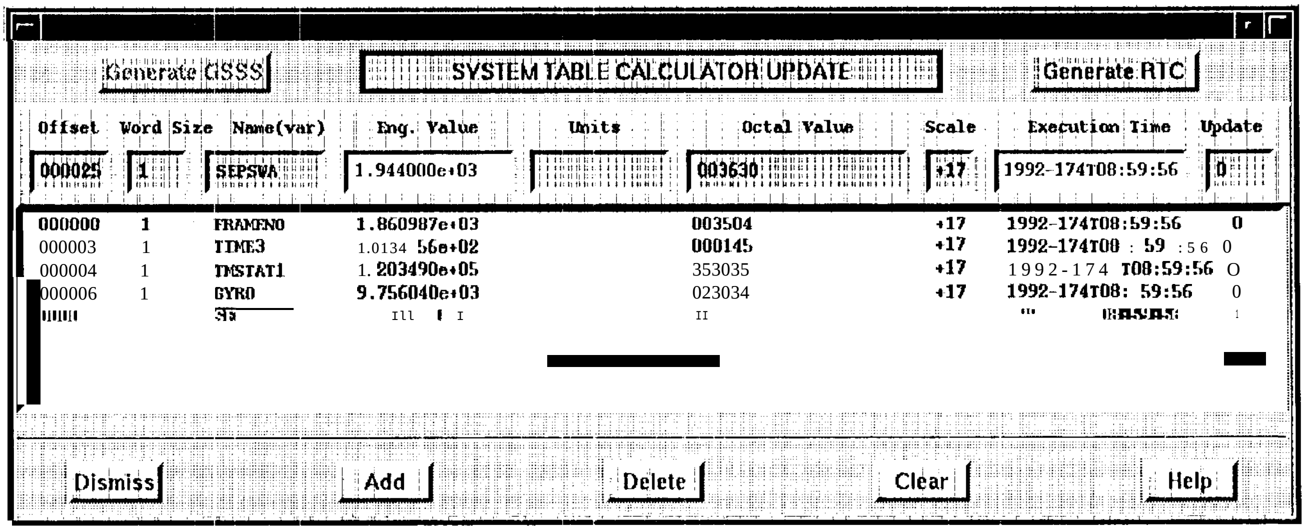
System Table Calculator

When an analyst wants to update a system table variable, he/she has to perform the following steps by hand:

1) Find the system table location to be updated.
2) Convert the desired engineering unit value to octal or hexadecimal using the correct scaling factor associated with that variable.
3) Generate the correct update command with the command mnemonic, converted octal or hex representation, time tag specifying the time at which the command is to be performed, and of course the correct syntax.

Spacecraft maneuvers require generating hundreds of these system table variable updates. Each step in this process is subject to human error.

1 he System Table Calculator function performs all of the above steps automatically. The only input that is needed is the desired engineering unit value and the time to update the table entry. The system table calculator display window is identical to the system table display except it has an extra button called "View Changes" and it behaves differently when a table entry is selected. The first time the user selects a table entry the system table calculator update window will appear with the selected entry separated into its constituent parts (see figure 9). The user enters the engineering unit and the update time and selects the "Add" button to add the modified table entry into a scrollable buffer window directly below the partitioned fields. After the user makes all desired changes, he/she selects the "Generate RTC" button to generate an uplinkable real-time command file containing the update command. The System 1 able Calculator eliminates many of the errors which can creep into this process and generates real-time commands much faster than a human can.



Offset	Word Size	Name(var)	Eng. Value	Units	Octal Value	Scale	Execution Time	Update
000025	1	SEPSVA	1.944000e+03		003630	+17	1992-174T08:59:56	0
000000	1	FRAMENO	1.860987e+03		003504	+17	1992-174T08:59:56	0
000003	1	TIME3	1.0134 56e+02		000145	+17	1992-174T08 : 59 :56	0
000004	1	TMSTAT1	1. 203490e+05		353035	+17	1992-174 T08:59:56	O
000006	1	GYRO	9.756040e+03		023034	+17	1992-174T08: 59:56	0
[illegible]		[illegible]	[illegible]		[illegible]		[illegible]	[illegible]

Figure 9- System Table Calculator Update Window

VI. Summary

A tool of the scope of the Memory State 1 racker has not been previously implemented at JPL. MST allows fast and easy access to the contents of OBC memory by maintaining and displaying a ground mirror-image of the OBC memory image.

MST obtains memory information from the ground image and from other ASCII files that exist independently of MST. Wherever possible, MS1 uses files that have been generated by ground processes for other purposes, thereby eliminating the need for special files whose formal, is designed solely for use by MST, This eliminates redundancy in data products.

1 he various reports generated by MS1 are readily available to the user via an intuitive Graphical User-Interface. Access to the contents of OBC memory is much faster than without the use of a memory state tracker, allowing ground personnel to perform the tasks of spacecraft analysis and sequence generation more accurately and efficiently.

VII. Acknowledgments

The software described in this paper was implemented by the Jet Propulsion Laboratory, California Institute of 1 ethnology under contract with the National Aeronautics and Space Administration. The authors wish to acknowledge support from TOPEX/POSEIDON Spacecraft Performance Analysis 1 earn (SPAT), Bill Mitchell of Fairchild Space Company, Kathy Shafer of Fairchild Space Company, and Monica B. Rivera, formerly of JPL.