

SCE User Guide

(CRAY T90™ Series)

HDM-069-C

Cray Research/Silicon Graphics Proprietary



Record of Revision

REVISION	DESCRIPTION
----------	-------------

	August 1995. Original printing.
A	March 1996. This revision corresponds to the MT-T2.2.0 offline diagnostic release.
B	June 1996. This revision corresponds to the MT-T2.2.1 offline diagnostic release.
C	August 1997. This revision corresponds to the MT-T2.3.0 offline diagnostic release.

Any shipment to a country outside of the United States requires a letter of assurance from Cray Research, Inc.

This document is the property of Cray Research, Inc. The use of this document is subject to specific license rights extended by Cray Research, Inc. to the owner or lessee of a Cray Research, Inc. computer system or other licensed party according to the terms and conditions of the license and for no other purpose.

Cray Research, Inc. Unpublished Proprietary Information — All Rights Reserved.

Autotasking, CF77, CRAY, CRAY-1, Cray Ada, CraySoft, CRAY Y-MP, CRInform, CRI/TurboKiva, HSX, LibSci, MPP Apprentice, SSD, SUPERCLUSTER, UNICOS, and X-MPEA are federally registered trademarks and Because no workstation is an island, CCI, CCMT, CF90, CFT, CFT2, CFT77, ConCurrent Maintenance Tools, COS, CRAY-2, Cray Animation Theater, CRAY APP, CRAY C90, CRAY C90D, Cray C++ Compiling System, CrayDoc, CRAY EL, CRAY J90, CRAY J90se, CrayLink, Cray NQS, Cray/REELibrarian, CRAY S-MP, CRAY SSD-T90, CRAY T3D, CRAY T3E, CRAY T3E-900, CRAY T90, CrayTutor, CRAY X-MP, CRAY XMS, CSIM, CVT, Delivering the power . . ., DGauss, Docview, EMDS, GigaRing, HEXAR, IOS, ND Series Network Disk Array, Network Queuing Environment, Network Queuing Tools, OLNEL, RQS, SEGLDR, SMARTE, SUPERLINK, System Maintenance and Remote Testing Environment, Trusted UNICOS, UNICOS MAX, and UNICOS/mk are trademarks of Cray Research, Inc.

Silicon Graphics and the Silicon Graphics logo are registered trademarks and Origin and Origin2000 are trademarks of Silicon Graphics, Inc.

OpenWindows is a trademark of Sun Microsystems, Inc.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company, Limited.

X Window System is a trademark of The Open Group.

Requests for copies of Cray Research, Inc. publications should be directed to:

CRAY RESEARCH, INC.
Customer Service Logistics
1100 Lowater Road
P.O. Box 4000
Chippewa Falls, WI 54729-0078
USA

Comments about this publication should be directed to:

CRAY RESEARCH, INC.
Service Publications and Training
890 Industrial Blvd.
P.O. Box 4000
Chippewa Falls, WI 54729-0078
USA

SCE USER GUIDE

Description of this Document	vi
SCE OVERVIEW	1
Logical Configuration	1
Configuration Parameters	2
Partitioning	2
Physical Partitions	3
Logical Partitions	3
How SCE Creates Logical Partitions	3
Logical Partition Ownership	4
Creating Physical Partitions and Logical Partitions	5
Memory Degradation	6
How SCE Degrades Memory	6
Using SCE to Degrade Memory	9
Preferred Sequence for Memory Degradations	9
Special Conditions for Memory Degradation with CP Testers and CRAY T916 Mainframes	10
Section-to-bank Mapping for CP Testers	11
Section-to-bank Mapping for CRAY T916 Mainframes ..	13
How Special Conditions Affect Memory Degradation ...	15
Spare Chip Memory Management	17
How SCE Controls the Spare Chips	17
Guidelines for Using Spare Chips	18
Single-byte Errors	18
Bursting Memory Chips	18
Using Spare Memory Chips	19
Creating a New Flaw-chip Entry	20
Swapping Module Entries	23
Editing a Module Entry	23
Deleting a Module Entry	23

Deleting a Flaw-chip Entry	23
Viewing All Module Entries	23
Viewing Specific Module Entries	23
Viewing All Flaw-chip Entries	24
Viewing Specific Flaw-chip Entries	24
Special Conditions for Using Spare Chips with CP Testers and CRAY T916 Mainframes	24

USING SCE **25**

Starting SCE	25
From a UNIX Prompt	25
From the OpenWindows Workspace Menu	27
MWS Workspace Menu Options	27
SWS Workspace Menu Options	29
From MME	30
Creating a Configuration	31
Loading an Existing Configuration	31
Creating a New Configuration	31
Asserting the Configuration	34
Modifying the Configuration	35
Asserting the Modified Configuration	35

SCE EXAMPLES **37**

Creating a New CRAY T94 Configuration	37
Degrading Memory in a CRAY T932 Mainframe (with CM03 Modules)	42
Memory Degradation Example (with Memory Parameters in Default Mode)	44
Degrading Memory by Banks (Using Default Mode)	45
Degrading Memory by Subsections (Using Default Mode)	45
Degrading Memory by Sections (Using Default Mode) ..	45
Memory Degradation Example (with Memory Parameters in Custom Mode)	46
Degrading Memory by Banks (Using Custom Mode)	47
Degrading Memory by Subsections (Using Custom Mode)	47

Degrading Memory by Sections (Using Custom Mode) . . .	47
Degrading Memory in a CRAY T916 Mainframe (with CM02 Modules)	48
Using Default Mode	48
Using Custom Mode	48
Setting up a Partition for Maintenance	50
Flawing a Bad Memory Chip	56

Figures

Figure 1. Physical and Logical Partition Parameters in the Base Window	5
Figure 2. Forcing Address Bit 2 to a 1	7
Figure 3. Forcing Address Bit 5 to a 0	8
Figure 4. Forcing Address Bit 9 to a 0, Address Bit 5 to a 0, and Address Bit 2 to a 1	8
Figure 5. Memory Utilization Map for a CP Tester with a CM02 Module	11
Figure 6. Memory Utilization Map for a CP Tester with a CM03 or CM04 Module	12
Figure 7. Memory Utilization Map for a CRAY T916 Mainframe with CM02 Modules	13
Figure 8. Memory Utilization Map for a CRAY T916 Mainframe with CM03 or CM04 Modules	14
Figure 9. Bank Parameters in Custom Mode	15
Figure 10. MWS Workspace Menu Options to Start SCE with an FEI Channel	27
Figure 11. MWS Workspace Menu Options to Start SCE with the Simulator	28
Figure 12. SWS Workspace Menu Options to Start SCE with an FEI Channel	29
Figure 13. SWS Workspace Menu Options to Start SCE with the Simulator	30
Figure 14. Default Parameter Locations in the SCE Base Window	32
Figure 15. Default Parameter Locations in the SCE T90: Miscellaneous Configuration Window	33
Figure 16. SCE Base Window Settings for a New CRAY T94 Configuration (Example)	40
Figure 17. SCE T90: Miscellaneous Configuration Window Settings for a New CRAY T94 Configuration (Example)	41
Figure 18. Example of Degrading Memory	42

Figures (continued)

Figure 19.	Example Memory Utilization Map	43
Figure 20.	Example Memory Parameters (in Default Mode) . . .	44
Figure 21.	Example Memory Parameters (in Custom Mode) . . .	46
Figure 22.	Bank Memory Parameters (Example)	49
Figure 23.	System Information Parameters (Example)	51
Figure 24.	Physical Partition Parameters (Example)	52
Figure 25.	Logical Partition 0 Parameters (Example)	54
Figure 26.	Logical Partition 1 Parameters (Example)	55

Tables

Table 1.	Logical Partition Group Assignment	3
Table 2.	Command Line Options	26
Table 3.	Additional SCE Commands	26
Table 4.	Example System Information Parameter Settings . .	37
Table 5.	Example Physical Partition Parameter Settings	38
Table 6.	Example Logical Partition Parameter Settings	39
Table 7.	Example Section-to-bank Mappings	48
Table 8.	Example Configuration Components	50
Table 9.	System Information Parameter Settings (Example) .	51
Table 10.	Physical Partition Parameter Settings (Example) . . .	52
Table 11.	Logical Partition 0 Parameter Settings (Example) . .	53
Table 12.	Logical Partition 1 Parameter Settings (Example) . .	54
Table 13.	Bad Bit Location Information	56

Description of this Document

This document describes the CRAY T90™ series System Configuration Environment (SCE) application that is used to configure CRAY T90 series mainframes. This document also describes how to use SCE.

This document is one component of the SCE documentation set, which also includes the following document:

SCE Interface Reference, publication number HDM-182-C.

The *SCE Interface Reference* describes the SCE interface and all available menu button commands.

SCE OVERVIEW

The System Configuration Environment (SCE) is X Window System™ software that runs on a maintenance workstation (MWS) or system workstation (SWS) to perform the CRAY T90 series mainframe configuration tasks. This application:

- Creates and manages the mainframe logical configuration (Refer to the following “Logical Configuration” description for more detail.)
- Creates and manages the mainframe partitions (Refer to the “Partitioning” description on page 2 for more detail.)
- Degrades memory to enable you to continue system operation or perform system maintenance in the presence of memory system failures (Refer to “Memory Degradation” on page 6 for more detail.)
- Creates and manages a table of the flawed memory module chips that SCE uses to substitute spare chips for the flawed chips (Refer to the “Spare Chip Memory Management” description on page 17 for more detail.)



Logical Configuration

CRAY T90 series mainframes use logical configuration: the configuration is set by manipulating soft switch values rather than by physically rearranging the hardware. Logical configuration enables you to include or exclude modules from a system configuration without powering down any modules or the mainframe. You can then perform maintenance in one part of the system while another part runs the operating system (OS). You can also reconfigure areas of the system without affecting other areas.

SCE issues maintenance channel functions to create and manage the logical configuration. These functions manipulate sanity codes in the sanity tree for modules in the mainframe. The presence or absence of a sanity code for a module in the mainframe is the logical equivalent of powering the module up or down. Refer to the *Maintenance Channel* document, publication number HTM-006-B, for more information about sanity codes and the sanity tree.

NOTE: All configuration selections apply symmetrically within the CPUs in the selected partition; for example, selecting 4-bank mode within a CPU means that all requests on all memory ports use 4-bank addressing. It is not possible to use 4-bank addressing through one CPU and 8-bank addressing through the remaining CPUs in the partition.

Configuration Parameters

SCE enables you to modify individual parameters for the memory, CPU, shared, I/O, and channel configurations. Use the  menu button to access the parameters. For more information about the commands that are available from the  menu button, refer to the *SCE Interface Reference*, publication number HDM-182-C.

CAUTION

Do not change individual parameters in the SCE popup windows unless you use caution and have a thorough understanding of CRAY T90 series mainframe configuration.

Partitioning

SCE controls all partitioning of the mainframe. Partitioning the mainframe enables multiple customer functions or maintenance functions to occur in different areas of the mainframe at the same time. This enables one portion of the mainframe to run customer operations while you maintain or repair another portion; it also enables a customer to run more than one operating system (OS) at a time.

NOTE: At the time of this printing, UNICOS® does not support partitioning.

Physical Partitions

You can configure CRAY T90 series mainframes into one or two physical partitions, depending on the mainframe type. CRAY T94™ and CRAY T916™ mainframes have one physical partition, and CRAY T932™ mainframes can have one or two physical partitions. A CRAY T932 mainframe that is configured with two physical partitions can function as two separate mainframes.

Each physical partition has an associated sanity tree, which establishes connections between modules. CRAY T94 and CRAY T916 mainframes have one sanity tree. CRAY T932 mainframes have one or two sanity trees, depending on the number of IO modules in the mainframe and the number of physical partitions that you create in the mainframe.

Logical Partitions

Physical partitions are divided into sets of components called logical partitions. Each physical partition contains one to four logical partitions. This enables you to create up to four logical partitions for CRAY T94 and CRAY T916 mainframes and up to eight logical partitions for CRAY T932 mainframes. Physical partitions are divided into logical partitions through changes to the soft switch values.

How SCE Creates Logical Partitions

SCE creates logical partitions by assigning components of the mainframe to the same memory group, I/O group, and cluster group, as shown in Table 1. These groups determine how the modules can interact.


Table 1. Logical Partition Group Assignment

Logical Partition	Memory Group	I/O Group	Cluster Group
0	0	0	0
1	1	1	1
2	2	2	2
3	3	3	3

A memory group is a logical grouping of a portion of memory and CPUs. A CPU can access only the memory that is in the same memory group as the CPU.

An I/O group is a logical grouping of CPUs, LOSP channels, and VHISP channels. A CPU can access only the channels that are in the same I/O group as the CPU.

A cluster group is a logical grouping of shared register clusters and CPUs. A CPU can use only the clusters that are in the same cluster group as the CPU.

SCE ensures that components are placed in the proper groups as you use the base window to define your logical partitions. You can also change the group parameters by using the SCE subwindows (available through the  menu button), but this is not recommended. Do not modify the subwindow parameters unless you use caution and have a thorough understanding of CRAY T90 series mainframe configuration.

Logical Partition Ownership

Logical partition ownership determines the type of customer or maintenance operations that can occur in the logical partitions. Logical partitions that have an operating system (OS) owner are used by a customer operating system. Logical partitions with a maintenance system (MS) owner are used by the maintenance system.

An OS owner does not imply any particular operating system. OS-owned logical partitions have two attributes: online maintenance allowed (OMA) and concurrent maintenance allowed (CMA). Online maintenance is maintenance that the operating system controls. Concurrent maintenance is an offline environment in a restricted subset of the logical partition's components.

All components of the mainframe (CPUs, clusters, I/O channels, etc.) are either assigned or unassigned to a logical partition. Components that are assigned to a logical partition with an OS owner are not available for maintenance use. Components that are assigned to a logical partition with an MS owner are not available for customer use. Components that are currently unassigned are available for customer or maintenance use; these components must be assigned to a logical partition before they can be used.

Components are classified by three states: working, broken, or unavailable. *Working* indicates that a component is in working condition. *Broken* indicates that a component is not in working condition. *Unavailable* indicates that a component is not logically present in the mainframe. Unavailable CP, SR, and IO modules are not part of the sanity tree. Unavailable memory modules are handled with memory degrade options.

Creating Physical Partitions and Logical Partitions

The easiest way to create physical partitions and logical partitions is to use the settings available in the base window (refer to Figure 1). Click on the settings you want, and SCE generates a sequence of maintenance channel functions that is sent to the mainframe when the configuration is applied. This sequence builds the appropriate physical partitions and logical partitions.

The screenshot shows the 'System Configuration Environment (SCET90 1.5.0) - SIM [illusion] (87)' window. It is divided into several sections:

- System Information Parameters:** Includes System Type (T94), System Serial (7001), System Memory (128M), and Physical Partitions (1, 2).
- Physical Partition Parameters †:** Includes Current Physical Partition (0), Sanity Port (I/O A), Sparechips (Disabled/Enabled), and Logical Partitions (1, 2, 3, 4).
- Logical Partition Parameters:** Includes Current Logical Partition (0), Memory (128M), Owner (OS, Maintenance, In Use), Maintenance (On-Line Enabled, Concurrent Enabled), and Description (PP0, LPO).
- CPUs:** A grid of CPU identifiers (00-37) and associated parameters (Cluster Offset: 1, Cluster Range: 22, Auto BCD: Off/On).
- LOSPs, HISPs, and VHISPs:** Grids of memory addresses for Logical OS Partitions (LOSPs), Hardware Interrupt Service Programs (HISPs), and Virtual Hardware Interrupt Service Programs (VHISPs).

At the bottom of the window, it says 'New configuration.'

† Additional physical partition parameters are now located in the SCE T90: Miscellaneous Configuration window, which is shown in Figure 15. Choose **View -> Miscellaneous** to access the parameters for the maintenance channel, boundary scan channel, error logger channel, and support channel.

Figure 1. Physical and Logical Partition Parameters in the Base Window

Memory Degradation

SCE can also degrade memory, which enables you to continue system operation or perform system maintenance in the presence of memory system failures. You may degrade a system until a CPU addresses only one bank of memory in each of four memory sections (a total of two memory modules with only one-eighth of the memory on each of those modules in use).

When you degrade memory, you are bypassing areas in memory where failures occur; then, the system can continue to operate and you may perform system maintenance on the degraded portion of memory. By degrading memory, you are forcing selected section, subsection, and bank bits to a specific value. Therefore, the memory address that the CPU references is different from the address that memory receives.

Use the memory configuration parameters to degrade memory used in a mainframe. For more information about the memory configuration parameters, refer to the “View → Memory” description in the *SCE Interface Reference*, publication number HDM-182-C.

NOTE: All configuration selections apply symmetrically within the CPUs in the selected partition; for example, selecting 4-bank mode within a CPU means that all requests on all memory ports use 4-bank addressing. It is not possible to use 4-bank addressing through one CPU and 8-bank addressing through the remaining CPUs in the partition.

How SCE Degrades Memory

SCE degrades memory by configuring the CP modules to force certain address bits to a 0 or 1 and to shift the remaining bits 1 bit higher. SCE manipulates the section, subsection, and bank profile and select bits, which causes the CP modules to avoid addressing and using the degraded memory.

The section profile bit indicates whether or not a bit should be added to the section portion of the address bits. If the section profile bit is set to 1, SCE configures the CP modules to add a bit to the section portion of the address bits and shift the higher bits one position to the left. The section select bit sets the extra bit to 0 or 1.

The subsection profile bits indicate whether or not a bit should be added to the subsection portion of the address bits. If a subsection profile bit is set to 1, SCE configures the CP modules to add a bit to the subsection portion of the address bits and shift the higher bits one position to the left. The subsection select bit sets the extra bit to 0 or 1.

The bank profile bits indicate whether or not a bit should be added to the bank portion of the address bits. If a bank profile bit is set to 1, SCE configures the CP modules to add a bit to the bank portion of the address bits and shift the higher bits one position to the left. The bank select bit sets the extra bit to 0 or 1.

Figure 2, Figure 3, and Figure 4 show examples of how memory degradation affects the address bits.

Figure 2 shows an example of memory degradation by sections. In this example, the CP modules address sections 4, 5, 6, and 7 and bypass addressing sections 0, 1, 2, and 3.

In this example, SCE configures the CP modules to force address bit 2 to a 1 and to shift the higher bits one position to the left. (To do this, SCE sets the section 2^2 profile bit to 1 and the section 2^2 select bit to 1.)

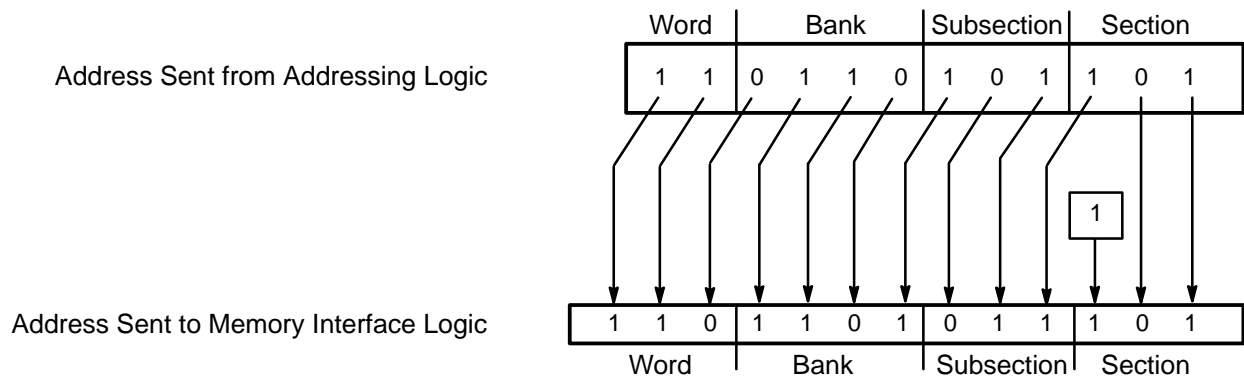


Figure 2. Forcing Address Bit 2 to a 1

Figure 3 shows an example of memory degradation by subsections. In this example, the CP modules address subsections 0, 1, 2, and 3 and avoid addressing subsections 4, 5, 6, and 7.

In this example, SCE configures the CP modules to force address bit 5 to a 0 and to shift the higher bits one position to the left. (To do this, SCE sets the subsection 2^2 profile bit to 1 and the subsection 2^2 select bit to 0.)

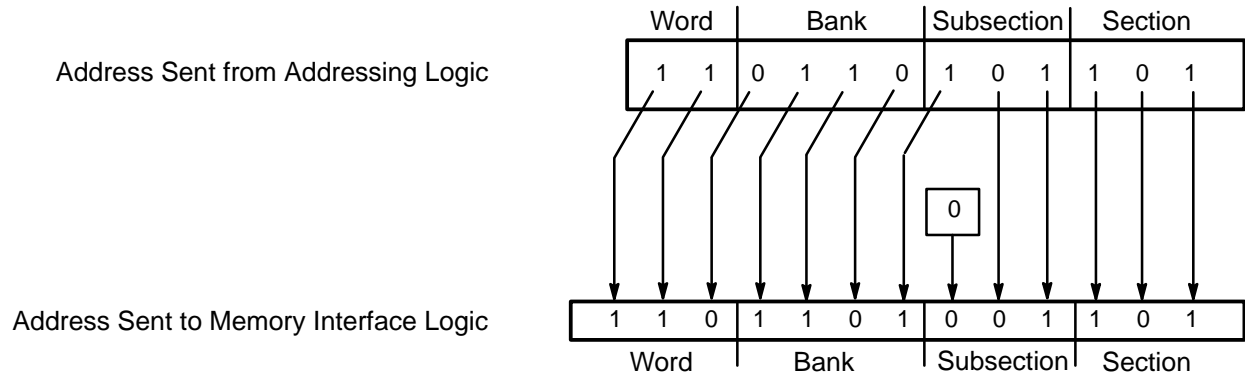


Figure 3. Forcing Address Bit 5 to a 0

Figure 4 shows an example of memory degradation by banks, subsections, and sections. The CP modules avoid addressing banks 10, 11, 12, 13, 14, 15, 16, and 17; subsections 4, 5, 6, and 7; and sections 0, 1, 2, and 3.

In this example, SCE configures the CP modules to force address bit 9 to a 0, address bit 5 to a 0, and address bit 2 to a 1. SCE configures the CP modules to shift the corresponding higher bits one position to the left. (To do this, SCE sets the bank 2^3 profile bit to 1 and the bank 2^3 select bit to 0, the subsection 2^2 profile bit to 1 and the subsection 2^2 select bit to 0, and the section 2^2 profile bit to 1 and the section 2^2 select bit to 1.)

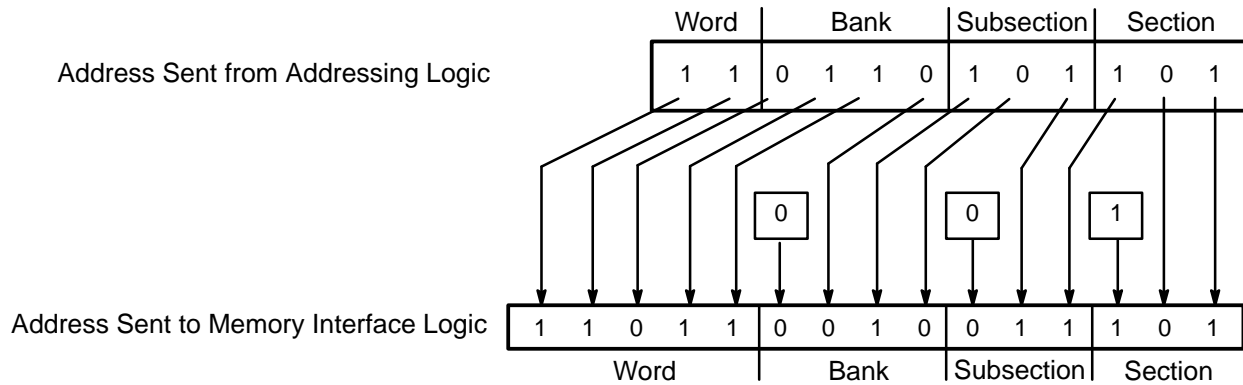


Figure 4. Forcing Address Bit 9 to a 0, Address Bit 5 to a 0, and Address Bit 2 to a 1

Each time an address bit is forced to a 0 or 1, the amount of memory available for use is reduced by one-half.

Using SCE to Degrade Memory

Choose **View → Memory** to access the SCET90 Memory window to perform memory degradation. This window provides two modes that you can use to perform memory degradation: default mode and custom mode.

In default mode, you simply specify which sections, subsections, and banks of memory you want to use. SCE forces the necessary address bits to degrade the memory that you do not want to use.

In custom mode, you can perform memory degradations for individual sections. For example, you could degrade 1 section to use 4 banks (for example, banks 0, 1, 2, and 3) while the other sections use the other 4 banks (banks 4, 5, 6, and 7).

Refer to the *SCE Interface Reference*, publication number HDM-182-C, for more information about the SCET90 Memory window and the parameters that are available in default and custom modes. Refer to “Degrading Memory in a CRAY T932 Mainframe (with CM03 Modules)” on page 42 and “Degrading Memory in a CRAY T916 Mainframe (with CM02 Modules)” on page 48 of this document for examples of how to use SCE to perform memory degradation.

Preferred Sequence for Memory Degradations

NOTE: If only one memory chip fails in a half-bank, you do not need to perform a memory degradation. Instead, you can simply reconfigure a spare chip to replace the flawed chip. Using a spare chip is not a form of memory degradation, and it will not cause more conflicts. (Refer to “Spare Chip Memory Management” on page 17 of this document for more information.)

If more than one chip fails in a half-bank, you can degrade memory by banks or sections for a CRAY T94 system or by banks, subsections, or sections for CRAY T916 and CRAY T932 systems.

For a given memory fault, there may be more than one way to degrade memory so that the mainframe can continue to operate. All degradations cause more conflicts for the remaining memory, so the preferred degradation (degrade) is the one that minimizes the number of conflicts. Memory degrades that occur farther away from CPUs cause fewer memory conflicts. This means that if no other reasons exist to start degrades elsewhere, you should degrade by banks, then by subsections (for CRAY T916 and CRAY T932 systems), and then by sections.

If a bank fails, degrade by banks, if possible. If a module fails, degrade by subsections, if possible. When you degrade by banks, degrade by the upper bank bit first, followed by the lower bank bit. When you degrade by subsections (CRAY T916 and CRAY T932 systems only), degrade in bit order, starting with the lowest bit.

Special Conditions for Memory Degradation with CP Testers and CRAY T916 Mainframes

There are special memory configuration conditions for CP testers and CRAY T916 mainframes that you should understand before you perform memory degradation for these systems. To configure these systems, SCE maps the sections to specific banks so it appears that there are 8 sections and 8 banks for systems with CM02 modules or 8 sections and 16 banks for systems with CM03 or CM04 modules.

NOTE: The section-to-bank mappings shown in the following paragraphs are the preferred configurations for CP testers and CRAY T916 mainframes because these mappings provide the best performance. You can use other mappings if you follow the patterns that the following paragraphs describe.

Section-to-bank Mapping for CP Testers

For CP testers with CM02 memory modules, SCE uses the following section-to-bank mapping (refer to Figure 5):

- Section 0 goes to banks 0 and 4
- Section 1 goes to banks 0 and 4
- Section 2 goes to banks 1 and 5
- Section 3 goes to banks 1 and 5
- Section 4 goes to banks 2 and 6
- Section 5 goes to banks 2 and 6
- Section 6 goes to banks 3 and 7
- Section 7 goes to banks 3 and 7

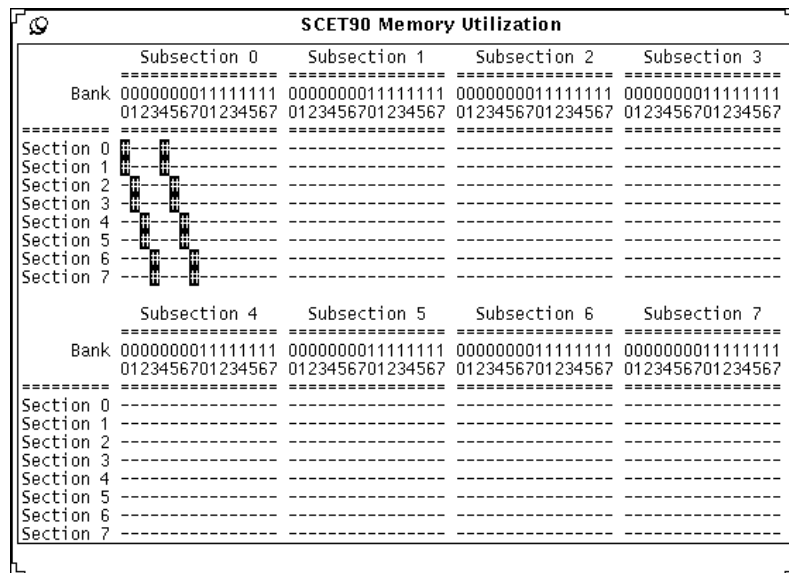


Figure 5. Memory Utilization Map for a CP Tester with a CM02 Module

For CP testers with CM03 or CM04 memory modules, SCE uses the following section-to-bank mapping (refer to Figure 6):

- Section 0 goes to banks 0, 4, 10, and 14
- Section 1 goes to banks 0, 4, 10, and 14
- Section 2 goes to banks 1, 5, 11, and 15
- Section 3 goes to banks 1, 5, 11, and 15
- Section 4 goes to banks 2, 6, 12, and 16
- Section 5 goes to banks 2, 6, 12, and 16
- Section 6 goes to banks 3, 7, 13, and 17
- Section 7 goes to banks 3, 7, 13, and 17

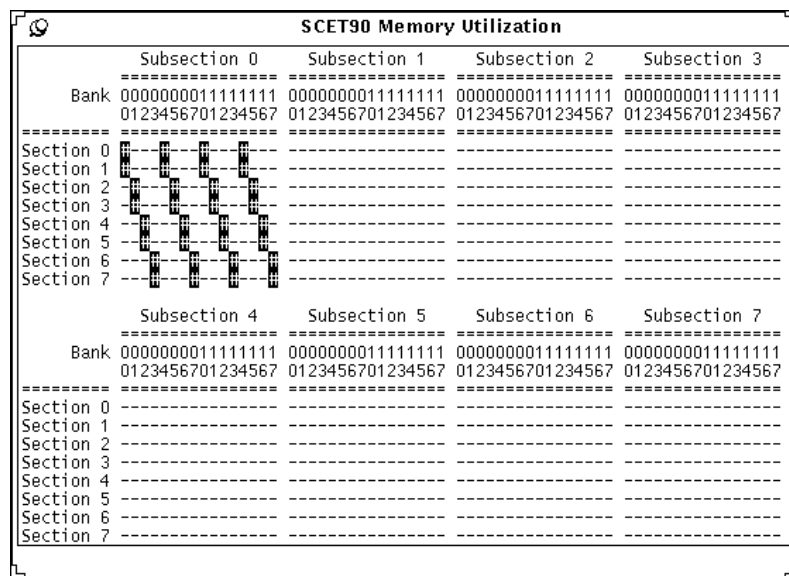


Figure 6. Memory Utilization Map for a CP Tester with a CM03 or CM04 Module

Section-to-bank Mapping for CRAY T916 Mainframes

For CRAY T916 mainframes with CM02 memory modules, SCE uses the following section-to-bank mapping (refer to Figure 7):

- Section 0 goes to banks 0, 1, 4, and 5
- Section 1 goes to banks 0, 1, 4, and 5
- Section 2 goes to banks 2, 3, 6, and 7
- Section 3 goes to banks 2, 3, 6, and 7
- Section 4 goes to banks 0, 1, 4, and 5
- Section 5 goes to banks 0, 1, 4, and 5
- Section 6 goes to banks 2, 3, 6, and 7
- Section 7 goes to banks 2, 3, 6, and 7

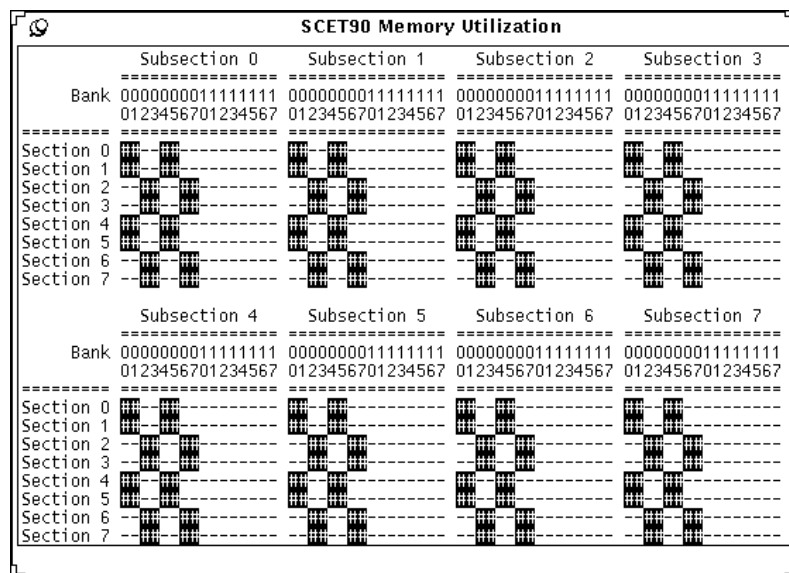


Figure 7. Memory Utilization Map for a CRAY T916 Mainframe with CM02 Modules

For CRAY T916 mainframes with CM03 or CM04 memory modules, SCE uses the following section-to-bank mapping (refer to Figure 8):

- Section 0 goes to banks 0, 1, 4, 5, 10, 11, 14, and 15
- Section 1 goes to banks 0, 1, 4, 5, 10, 11, 14, and 15
- Section 2 goes to banks 2, 3, 6, 7, 12, 13, 16, and 17
- Section 3 goes to banks 2, 3, 6, 7, 12, 13, 16, and 17
- Section 4 goes to banks 0, 1, 4, 5, 10, 11, 14, and 15
- Section 5 goes to banks 0, 1, 4, 5, 10, 11, 14, and 15
- Section 6 goes to banks 2, 3, 6, 7, 12, 13, 16, and 17
- Section 7 goes to banks 2, 3, 6, 7, 12, 13, 16, and 17

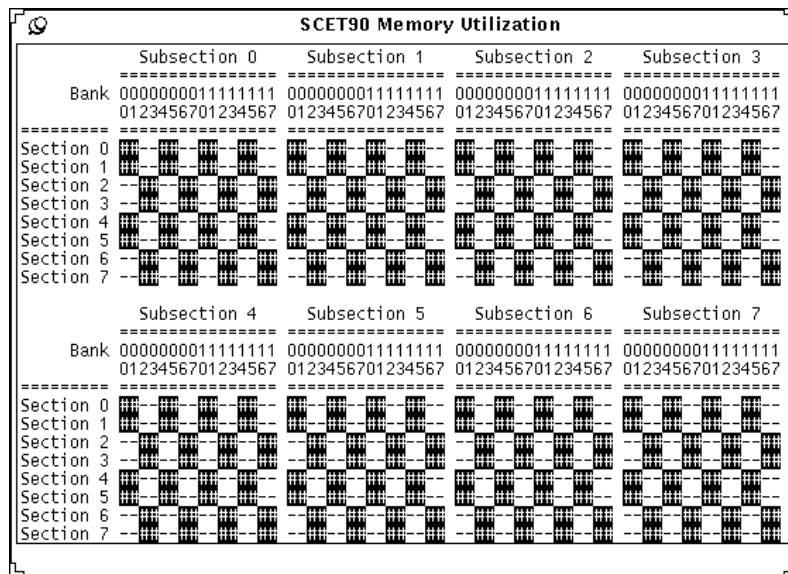


Figure 8. Memory Utilization Map for a CRAY T916 Mainframe with CM03 or CM04 Modules

How Special Conditions Affect Memory Degradation

When you want to perform memory degradation by banks for a CP tester or a CRAY T916 mainframe, you should use custom mode because you will need to change the section-to-bank mappings for the memory that you want to degrade. Custom mode displays how the sections are configured to specific banks. For example, Figure 9 shows the section-to-bank mappings for a CRAY T916 mainframe with CM02 memory modules.

Banks: ▾					
	2 [^] 3	2 [^] 2	2 [^] 1	2 [^] 0	
Section 0: ▾	0	-	0	-	0,1,4,5
Section 1: ▾	0	-	0	-	0,1,4,5
Section 2: ▾	0	-	1	-	2,3,6,7
Section 3: ▾	0	-	1	-	2,3,6,7
Section 4: ▾	0	-	0	-	0,1,4,5
Section 5: ▾	0	-	0	-	0,1,4,5
Section 6: ▾	0	-	1	-	2,3,6,7
Section 7: ▾	0	-	1	-	2,3,6,7

Figure 9. Bank Parameters in Custom Mode

For CP testers with CM02 modules, bank configuration in custom mode is as follows:

16 of 16 banks	(not valid)
8 of 16 banks	(not valid)
4 of 16 banks	(not valid)
2 of 16 banks	(all banks)
1 of 16 banks	(degraded to use half of the banks)

For CP testers with CM03 or CM04 modules, bank configuration in custom mode is as follows:

16 of 16 banks	(not valid)
8 of 16 banks	(not valid)
4 of 16 banks	(all banks)
2 of 16 banks	(degraded to use half of the banks)
1 of 16 banks	(degraded to use a fourth of the banks)

For CRAY T916 systems with CM02 modules, bank configuration in custom mode is as follows:

16 of 16 banks	(not valid)
8 of 16 banks	(not valid)
4 of 16 banks	(all banks)
2 of 16 banks	(degraded to use half of the banks)
1 of 16 banks	(degraded to use a fourth of the banks)

For CRAY T916 systems with CM03 or CM04 modules, bank configuration in custom mode is as follows:

16 of 16 banks	(not valid)
8 of 16 banks	(all banks)
4 of 16 banks	(degraded to use half of the banks)
2 of 16 banks	(degraded to use a fourth of the banks)
1 of 16 banks	(degraded to use an eighth of the banks)

Refer to “Degrading Memory in a CRAY T916 Mainframe (with CM02 Modules)” on page 48 for an example of degrading memory by banks in a CRAY T916 mainframe.

Spare Chip Memory Management

As described in the *Memory Module (CM02)* document, publication number HTM-004-0, you can flaw failing memory chips. You can then have SCE configure the mainframe to use spare memory chips that are physically located in the same row of chips in the memory stack as the flawed chips.

Use the SCE T90: Spare Chip window to create and maintain a table of modules with flawed chips. SCE uses this table to generate the appropriate direct memory access (DMA) commands to replace the flawed chips with spare chips.

To flaw a chip, you need the following information: bank, subsection, section, and bad bit. This information can come from the error logging program (`xelog`), the MME error log, or a diagnostic test program.

NOTE: When you use spare memory chips, you should set the Debug Level setting to `normal`. This setting causes SCE to display the flaw map in the standard output window when you assert a configuration. When you view the flaw map during an assertion, you will see any errors that are related to the spare chip assignment. For example, if you enter 2 flaws in the same stack (which invalidates the first flaw), SCE will not display an error until you assert the configuration.

How SCE Controls the Spare Chips

SCE uses two hidden files in the `usr/cfg/` directory to implement spare-chip functionality. The `.mmm` file contains a memory module map that defines which modules are in the system. The `.scm` file contains a spare-chip map that describes which chips are flawed (the information that you provide to SCE). The `.mmm` and `.scm` files are in a binary format; you cannot manually edit these files.

SCE uses the memory module map, spare-chip map, and memory degradation information to build flaw maps. When you apply a configuration, SCE generates the flaw maps and writes each logical partition with a flaw map. All spare-chip memory management activity is transparent to the OS.

Guidelines for Using Spare Chips

Use the following guidelines to determine whether you should use SCE's spare-chip management function to flaw a chip so that a spare chip replaces the flawed chip.

Single-byte Errors

A single-byte error (SBE) means that 1 to 2 bits are failing on the same memory chip.

During a 24-hour period, a memory chip that exhibits a consistent or increasing failure rate is a possible candidate for flawing. Monitor the frequency of failures over a 7-day period; at least 250 hits per day over a 7-day period should occur before you consider flawing the chip. However, use the spare chip feature before system performance is compromised.

NOTE: You should log all SBEs as they occur for individual modules. If a module is returned to Central Repair for a double-bit error or meets the criteria to be returned for flaws, this log and the flaw map should accompany the module. The log enables Central Repair to change chips that do not currently meet the criteria for flawing but that may fail more seriously in the future.

Bursting Memory Chips

A burst is defined as multiple SBEs that are logged rapidly in a short time. Bursts are not counted in specific numbers; for example, *more than* 500 hits in 1 second would be considered a burst.

During a 24-hour period, a memory chip that experiences one burst is not a candidate for flawing. If the chip experiences more than three bursts during a 24-hour period, consider flawing the chip.

During a 7-day period, a memory chip that experiences one burst is not a candidate for flawing. If the burst repeats on more than 3 days during a 7-day period, consider flawing the chip.

Using Spare Memory Chips

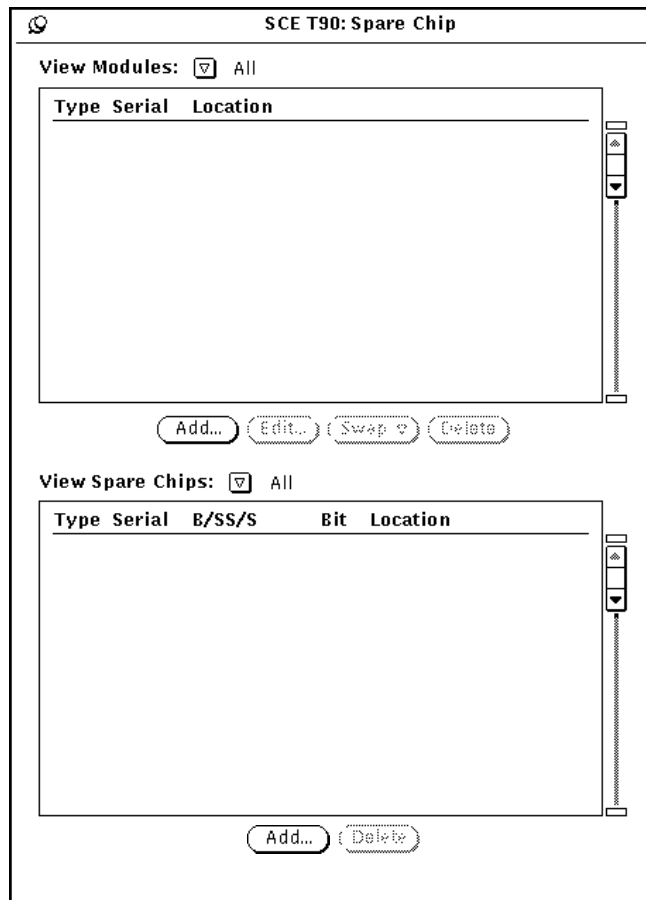
SCE includes the following function, which enable you to flaw a failing chip to use the spare memory chips:

- Creating a new flaw-chip entry
- Swapping module entries
- Editing a module entry
- Deleting a module entry
- Deleting a flaw-chip entry
- Viewing all module entries
- Viewing specific module entries
- Viewing all flaw-chip entries
- Viewing specific flaw-chip entries

Creating a New Flaw-chip Entry

Perform the following procedure to create a new flaw-chip entry in the table.

1. Choose **View -> Spare Chip** in the SCE base window. The SCE T90: Spare Chip window appears:



Before you can enter the bank, subsection, section, and bad bit information; you must create an entry for the memory module.

- In the SCE T90: Spare Chip window, click on the **Add...** button that is located below the View Modules scroll box to add a module entry to the table. SCE displays the SCE T90: Module Add window:

The screenshot shows a dialog box titled "SCET90: Module Add". It contains the following fields and controls:

- Module Serial:** A dropdown menu with "0" selected.
- Module Type:** Three buttons: "CM02", "CM03", and "CM04".
- Module Location:** Two buttons: "Spare" and "Mainframe".
- Mainframe Type:** Four buttons: "Tester", "T94", "T916", and "T932".
- Mainframe Serial:** A dropdown menu with "0" selected.
- Mainframe Slot:** Four buttons: "C2", "C3", "C4", and "C5".
- Apply:** A button at the bottom center.

- In the Module Serial field, enter the serial number of the module.
- Click on Module Type: **CM02**, **CM03**, or **CM04** to indicate the memory module type.
- Specify the location of the module:
 - If the module is a spare module that is not installed in the mainframe, click on Module Location: **Spare**.
 - If the module is installed in the mainframe, click on Module Location: **Mainframe**. Next, specify the Mainframe Type that you are using (click on **Tester**, **T94**, **T916**, or **T932**). Then, enter the serial number of the mainframe in the Mainframe Serial field, and click on the Mainframe Slot in which the memory module is located.
- Click on the **Apply** button that is located in the SCE T90: Module Add window to insert the module information into the table.

Once you have created a module entry, you can add flawed-chip information to the table.

- In the SCE T90: Spare Chip window, click on the button that is located below the View Spare Chips scroll box to add a flaw-chip entry to the table. SCE displays the SCE T90: Spare Chip Add window:

SCET90: Spare Chip Add

Location:

Bank:

0	1	2	3	4	5	6	7
10	11	12	13	14	15	16	17

Subsection:

0	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---

Section:

0	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---

Bad Bit:

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49
50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69
70	71	72	73	74	75				


- From the Location: , choose the serial number for the mainframe or the spare module on which the flawed chip is located.
- Click on the appropriate Bank, Subsection, Section, and Bad Bit settings.

NOTE: The subsection parameter does not apply to CRAY T94 mainframes because they do not have subsections.



- Click on the button that is located in the SCE T90: Spare Chip Add window to insert the flawed-chip information into the table.

SCE substitutes a spare chip for the flawed chip the next time you apply the configuration if the spare-chip option is enabled for the physical partition that contains the flawed chip.


Swapping Module Entries

SCE enables you to swap two module entries, which represents swapping the physical modules. To swap module entries, select one module that you want to swap. Then, choose the module with which you want to swap from the  menu button.


Editing a Module Entry

Click on the  button in the SCE T90: Spare Chip window to change a module entry. SCE displays a Module Edit window that displays the current values for the module entry. Change the values and click on the  button in the Module Edit window.


Deleting a Module Entry

To delete a module entry, select the entry in the View Modules scroll box and click on the  button that is located below the View Modules scroll box.


Deleting a Flaw-chip Entry


To delete a flaw-chip entry, select the entry in the View Spare Chips scroll box and click on the  button that is located below the View Spare Chips scroll box.

Viewing All Module Entries

SCE displays all module entries in the View Modules scroll box when you choose All from the View Modules: .

Viewing Specific Module Entries

SCE displays only module entries for a specific mainframe in the View Modules scroll box when you choose the mainframe serial number from the View Modules: .

SCE displays only module entries for spare modules in the View Modules scroll box when you choose Spare from the View Modules: .

Viewing All Flaw-chip Entries

SCE displays all current flaw-chip entries in the View Spare Chips scroll box when you choose All from the View Spare Chips: .

Viewing Specific Flaw-chip Entries

SCE displays only the flaw-chip entries for a specific module in the View Spare Chips scroll box when you choose the module from the View Spare Chips: .

Special Conditions for Using Spare Chips with CP Testers and CRAY T916 Mainframes

Special memory configuration conditions occur for CP testers and CRAY T916 mainframes that you should remember when you use spare chips with these systems.

A CP tester has only one memory module, but the memory configuration appears to be 8 sections and 8 banks. When you enter spare-chip information for a tester, SCE maps the section information to an odd or even value.

A CRAY T916 mainframe has 16 memory modules, but the memory configuration appears to have 8 sections, 8 subsections, and 4 banks. Sections 2, 3, 4, and 5 are on the same module. Sections 2 and 3 go to 4 of the 8 banks, and sections 4 and 5 are mapped to the other 4 banks. When you enter spare-chip information for a CRAY T916 mainframe, SCE restricts the section values to 0, 1, 2, or 3. Section 4 maps to section 2; section 5 maps to section 3, section 6 maps to section 0; and section 7 maps to section 1. SCE displays the original and mapped sections in the spare-chip list.

USING SCE

The following procedure provides a general overview of the process for using SCE. This section includes related information for each step of the process.

1. Starting SCE
2. Creating a configuration
3. Asserting the configuration
4. Modifying the configuration
5. Asserting the modified configuration
6. Repeating Step 4 and Step 5 (as needed to adjust the configuration)

Starting SCE

You can start SCE from a UNIX® prompt, from the OpenWindows™ Workspace menu, or from any environment of the Mainframe Maintenance Environment (MME) application.

NOTE: For information about starting SCE from a Service Center through a hub, refer to the *Remote Support* document, publication number HMM-106-A.

From a UNIX Prompt

You can start SCE from a UNIX prompt with the following command:

```
sce [-copy num] [-default filename] [-kill]
    [-remote host | -client | -server]
    [-sim]
```

Table 2 describes the available command line options.

Table 2. Command Line Options

Option	Description
-client	Start SCE client only
-copy <i>num</i>	Connect to maintenance software assigned the copy number specified by <i>num</i> NOTE: Copy numbers are necessary only when you run multiple copies of SCE on the same MWS or SWS (for example, when you run several SCE copies with the simulator or when you use SCE to support multiple CRAY T90 series mainframes that are connected to the same MWS or SWS).
-default <i>filename</i>	Load configuration file <i>filename</i> and apply the configuration
-kill	Kill all maintenance system processes (MME, LME, SCE) before starting SCE
-remote <i>host</i>	Start SCE client only and connect to server on <i>host</i>
-server	Start SCE server only
-sim	Start SCE with the simulator

Table 3 lists additional SCE commands that you can enter at the UNIX prompt. Refer to the online manual (man) pages for detailed information about these commands.

Table 3. Additional SCE Commands

Command	Description
sce_assert	Assert a configuration
sce_assume	Assume a configuration
sce_cpreset	Reset a hung CPU
sce_oscfg	Create an operating system (OS) configuration file

From the OpenWindows Workspace Menu

You can start SCE from the OpenWindows Workspace menu on either an MWS or an SWS.

MWS Workspace Menu Options

Figure 10 shows the OpenWindows Workspace menu options that you should choose on an MWS to start SCE with an FEI channel. Choose any copy number.

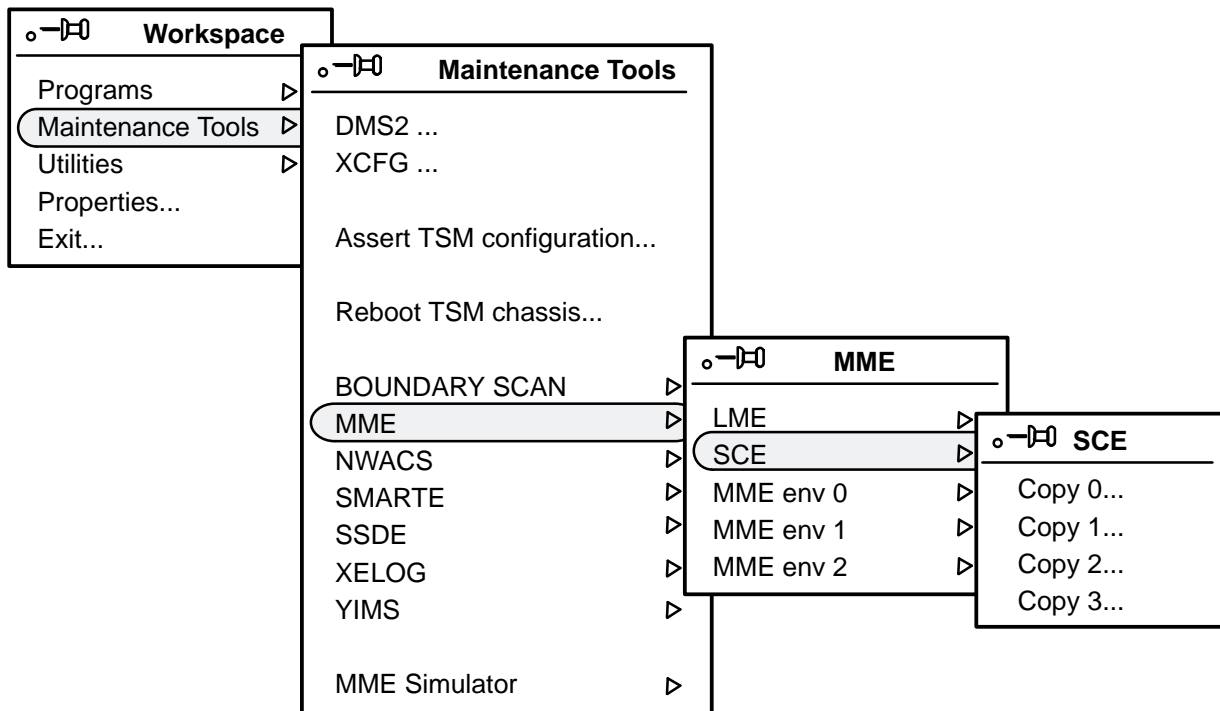


Figure 10. MWS Workspace Menu Options to Start SCE with an FEI Channel

Figure 11 shows the OpenWindows Workspace menu options that you should choose on an MWS to start SCE with the simulator.

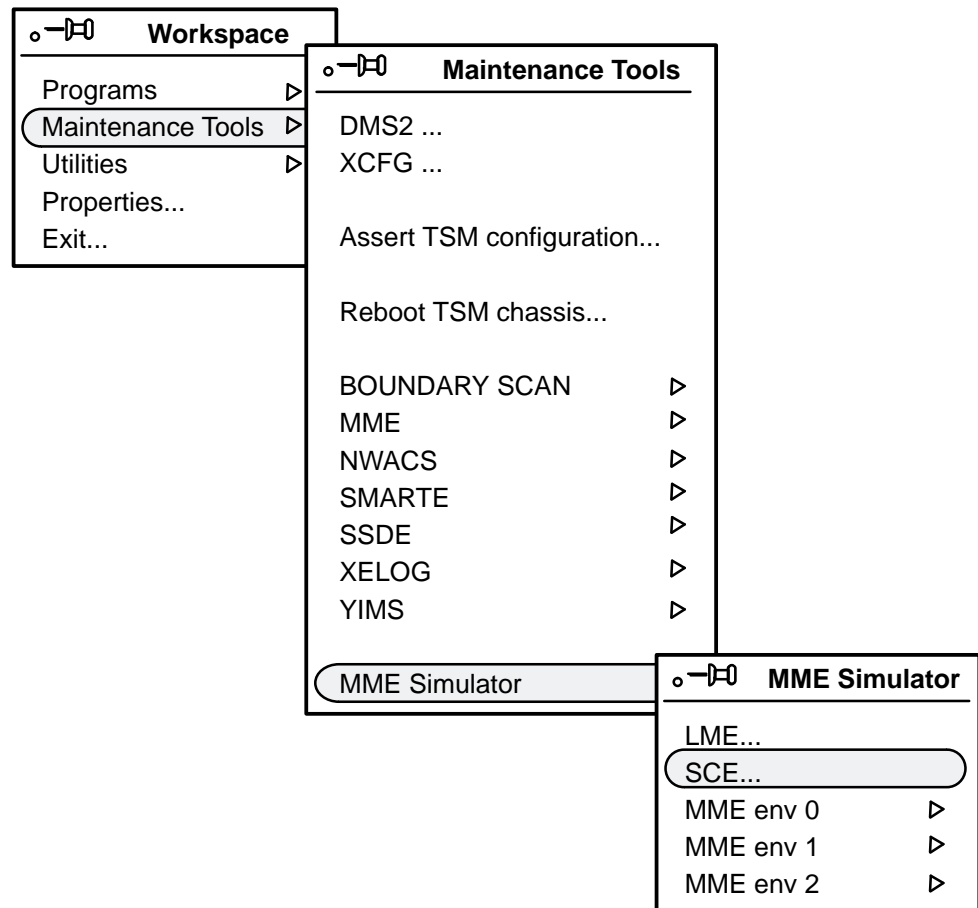


Figure 11. MWS Workspace Menu Options to Start SCE with the Simulator

SWS Workspace Menu Options

Figure 12 shows the OpenWindows Workspace menu options that you should choose on an SWS to start SCE with an FEI channel. Choose any copy number.

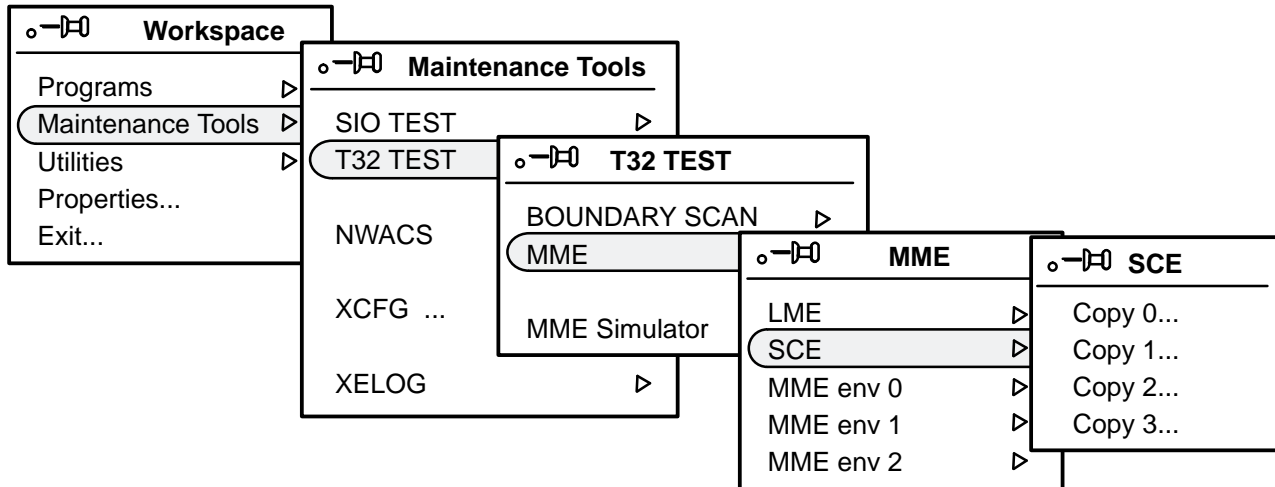


Figure 12. SWS Workspace Menu Options to Start SCE with an FEI Channel

Figure 13 shows the OpenWindows Workspace menu options that you should choose on an SWS to start SCE with the simulator.

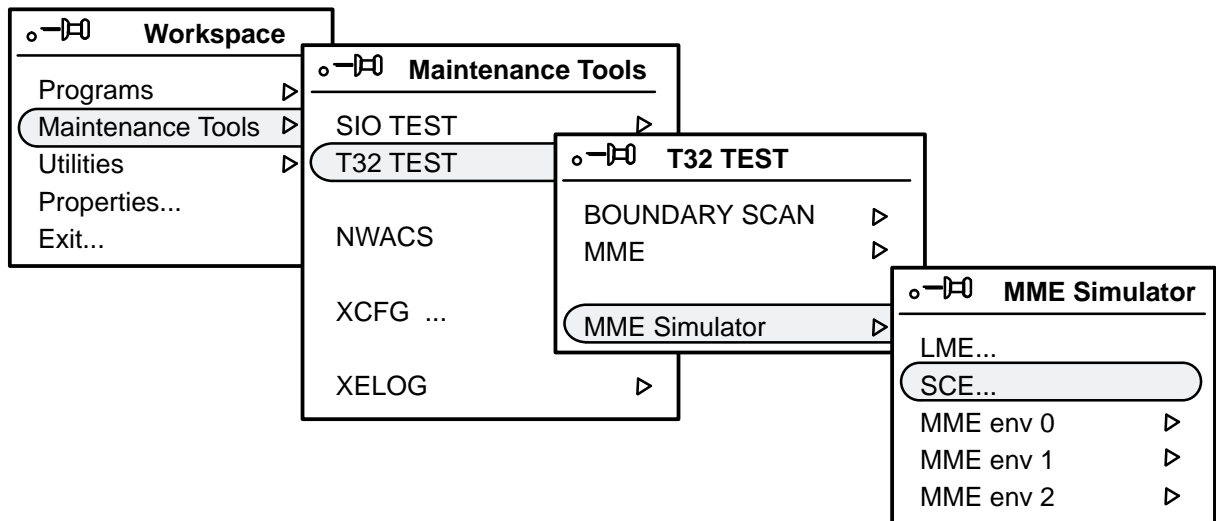


Figure 13. SWS Workspace Menu Options to Start SCE with the Simulator

From MME

Choose **Utilities** → **Configuration** from the MME base window to start SCE from MME.

Creating a Configuration

When you first start SCE, you need to set the parameters for the initial mainframe configuration. There are two ways to create the configuration: you can load a previously saved configuration or create a new configuration.

Loading an Existing Configuration

Use the File → Load command to load an existing configuration file.

Creating a New Configuration

You normally use only the SCE base window to create a new configuration because the base window contains all the parameters necessary to create most typical configurations. The following procedure describes how to use the SCE base window to create a new configuration.

1. Choose **File → New → T94** to create a new CRAY T94 mainframe configuration, choose **File → New → T916** to create a new CRAY T916 mainframe configuration, or choose **File → New → T932** to create a new CRAY T932 configuration.

SCE loads a set of default values, which appear in the SCE base window. Figure 14 on page 32 shows the SCE base window with default values for a CRAY T94 mainframe configuration.

2. Modify the system information parameters to indicate the serial number, amount of system memory, and number of physical partitions.
3. Set up the configuration information for the current physical partition.
4. Modify the logical partition parameters to create each logical partition.

For each logical partition, you will specify the owner of the partition (operating system or maintenance), the CPUs in the partition, the cluster information for the partition, the LOSP channels in the partition, the HISP channels in the partition, and the VHISP channels in the partition.

- Use the **File → Save** command to save the configuration. This enables you to reload the configuration later with the File → Load command.

The screenshot shows the 'System Configuration Environment (SCET90 1.5.0) - SIM [illusion] (87)' window. It features a menu bar with 'File', 'View', 'Edit', 'Utilities', 'Assume', 'Assert', and 'Reset'. The main area is divided into three sections:

- System Information Parameters:** Includes 'System Type: T94', 'System Serial: 7001', 'System Memory: 128M', and 'Physical Partitions: 1 2'.
- Physical Partition Parameters †:** Includes 'Current Physical Partition: 0', 'Sanity Port: I/O A', 'Sparechips: Disabled Enabled', and 'Logical Partitions: 1 2 3 4'.
- Logical Partition Parameters:** Includes 'Current Logical Partition: 0', 'Memory: 128M', 'Owner: OS Maintenance In Use', 'Maintenance: On-Line Enabled Concurrent Enabled', 'Description: PP0, LP0', and a 'CPUs' table. Below this are tables for 'LOSPs', 'HISPs', and 'VHISPs'.

At the bottom of the window, it says 'New configuration.'

† Additional physical partition parameters are now located in the SCE T90: Miscellaneous Configuration window, which is shown in Figure 15. Choose **View → Miscellaneous** to access the parameters for the maintenance channel, boundary scan channel, error logger channel, and support channel.

Figure 14. Default Parameter Locations in the SCE Base Window

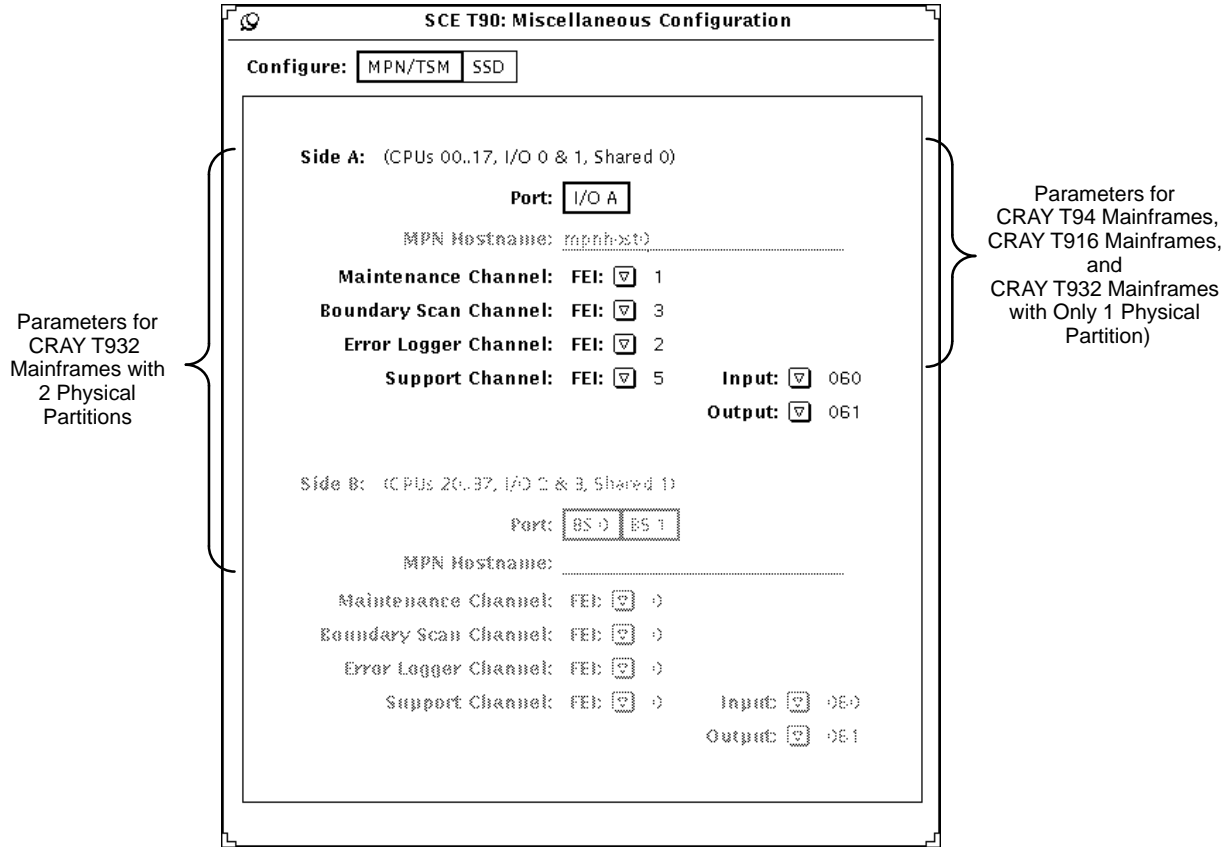


Figure 15. Default Parameter Locations in the SCE T90: Miscellaneous Configuration Window

NOTE: Figure 15 shows the SCE T90: Miscellaneous Configuration window with the parameters for IO01 modules. Click on IO Type: **CageRing** in the SCE T90: I/O Configuration window to display the parameters for IO02 modules.

Asserting the Configuration

Once you have set the parameters for the initial configuration, you need to assert the configuration to configure the CRAY T90 series mainframe. Asserting the configuration builds the sanity tree(s) for your specified configuration.

CAUTION


Asserting a configuration halts all operating system and user activity in the mainframe. Data is lost if an operating system is running in the mainframe when you assert a configuration.

Choose **Assert → Physical Partition 0**, **Assert → Physical Partition 1**, or **Assert → Both Partitions** to assert the configuration. SCE asserts the configuration.

NOTE: There is another option for updating a configuration: you can assume the configuration. Assuming a configuration sends the configuration data to SCE's clients without updating the sanity tree.


For descriptions of the actions that SCE performs to assert or assume a configuration, refer to the “Assert → Physical Partition 0,” “Assert → Physical Partition 1,” “Assert → Both Partitions,” “Assume → Physical Partition 0,” “Assume → Physical Partition 1,” and “Assume → Both Partitions” descriptions in the *SCE Interface Reference*, publication number HDM-182-C.

Modifying the Configuration

You may need to modify the configuration to degrade memory, to modify the available partitions to perform maintenance, or to use the spare memory chips. You can modify the configuration by changing parameters in the SCE base window or by changing the parameters in the windows that you can access with the  menu button commands.

CAUTION

Do not change individual parameters in the SCE popup windows unless you use caution and have a thorough understanding of CRAY T90 series mainframe configuration.

For more information about the commands that are available from the  menu button, refer to the *SCE Interface Reference*, publication number HDM-182-C.

Asserting the Modified Configuration

CAUTION

Asserting a configuration halts all operating system and user activity in the mainframe. Data is lost if an operating system is running in the mainframe when you assert a configuration.

Choose **Assert → Physical Partition 0**, **Assert → Physical Partition 1**, or **Assert → Both Partitions** to assert the configuration. SCE asserts the modified configuration. Asserting the configuration rebuilds the sanity tree(s) for your modified configuration.

NOTE: There is another option for updating a configuration: you can assume the configuration. Assuming a configuration sends the configuration data to SCE's clients without updating the sanity tree.

For descriptions of the actions that SCE performs to assert or assume a configuration, refer to the “Assert → Physical Partition 0,” “Assert → Physical Partition 1,” “Assert → Both Partitions,” “Assume → Physical Partition 0,” “Assume → Physical Partition 1,” and “Assume → Both Partitions” descriptions in the *SCE Interface Reference*, publication number HDM-182-C.

SCE EXAMPLES

This section provides the following examples to illustrate the functions that you can perform with SCE:

- Creating a new CRAY T94 configuration
- Degrading memory in a CRAY T932 mainframe
- Degrading memory in a CRAY T916 mainframe
- Setting up a partition for maintenance
- Flawing a bad memory chip

Creating a New CRAY T94 Configuration

This example shows how to create a configuration for a CRAY T94 mainframe with an IO01 module. This example configuration includes all possible modules and channels. The configuration includes 1 physical partition, 1 logical partition, and 128 Mwords of memory. The mainframe serial number is 7001.

1. Choose **File** → **New** → **T94** to create a new CRAY T94 mainframe configuration. SCE creates a default set of parameters.
2. Modify the system information parameters to indicate the serial number, amount of system memory, and number of physical partitions.

Table 4 shows the system information parameter settings that are necessary to configure the CRAY T94 mainframe for this example.

Table 4. Example System Information Parameter Settings

Parameter	Setting
System Type	T94
System Serial	7001
System Memory	128M
Physical Partitions	1 (for CRAY T94 mainframe)

3. Set up the configuration information for the current physical partition.

Table 5 shows the physical partition parameter settings for this example.

Table 5. Example Physical Partition Parameter Settings

Parameter	Setting
Current Physical Partition †	0 (CRAY T94 mainframes have only one physical partition)
Sanity Port †	<i>Left at the default for this example</i>
Sparechips †	<i>Left at the default for this example</i>
Logical Partitions †	1
Maintenance Channel ‡	<i>Left at the default for this example</i>
Boundary Scan Channel ‡	<i>Left at the default for this example</i>
Error Logger Channel ‡	<i>Left at the default for this example</i>
Support Channel ‡	<i>Left at the default for this example</i>
Input ‡	<i>Left at the default for this example</i>
Output ‡	<i>Left at the default for this example</i>

† These parameters are located in the SCE base window.

‡ These parameters are located in the SCE T90: Miscellaneous Configuration window. Choose **View -> Miscellaneous** to access this window.

4. Modify the logical partition parameters to create each logical partition.

Table 6 shows the logical partition parameter settings that are necessary for this example.

Table 6. Example Logical Partition Parameter Settings

Parameter	Setting
Current Logical Partition	0 (to define parameters for logical partition 0)
Memory	128M
Owner	OS
Maintenance	Online and concurrent maintenance enabled
Description	<i>Left at the default for this example</i>
CPUs	0 ,1, 2, and 3
Cluster Offset	1
Cluster Range	22 (selects clusters 1, 2, 3, 4, 5, 6, 7, 10, 11, 12, 13, 14, 15, 16, 17, 20, 21, and 22)
Auto BCD	<i>Left at the default for this example</i>
LOSPs	100/101, 102/103, 104/105, 106/107, 110/111, 112/113, 114/115, 116/117, and 060/061
HISPs	400, 401, 402, 403, 404, 405, 406, and 407
VHISPs	020, 021, 022, and 023

Figure 16 shows the SCE base window with the correct parameters for this example. Figure 17 shows the SCE T90: Miscellaneous Configuration window with the correct parameters for this example.

System Configuration Environment (SCET90 1.5.0) – SIM [illusion] (87)

File ▾
View ▾
Edit ▾
Utilities ▾
Assume ▾
Assert ▾
Reset ▾

System Type: T94
System Serial: 7001 ▲ ▾
System Memory: 128M
Physical Partitions: 1 2

Current Physical Partition: 0
Sanity Port: I/O A
Sparechips: Disabled Enabled
Logical Partitions: 1 2 3 4

Current Logical Partition: 0
Memory: 128M
Owner: OS Maintenance In Use
Maintenance: On-Line Enabled Concurrent Enabled
Description: PP0, LP0

CPUs:

00	04	10	14	20	24	30	34
01	05	11	15	21	25	31	35
02	06	12	16	22	26	32	36
03	07	13	17	23	27	33	37

Cluster Offset: 1 ▲ ▾
Cluster Range: 22 ▲ ▾
Auto BCD: Off On

LOSPs:

100/101	120/121	140/141	160/161
102/103	122/123	142/143	162/163
104/105	124/125	144/145	164/165
106/107	126/127	146/147	166/167
110/111	130/131	150/151	170/171
112/113	132/133	152/153	172/173
114/115	134/135	154/155	174/175
116/117	136/137	156/157	176/177
060/061	062/063	064/065	066/067

HISPs:

400	410	420	430
401	411	421	431
402	412	422	432
403	413	423	433
404	414	424	434
405	415	425	435
406	416	426	436
407	417	427	437

VHISPs:

020	030
021	031
022	032
023	033
024	034
025	035
026	036
027	037

New configuration.

Figure 16. SCE Base Window Settings for a New CRAY T94 Configuration (Example)

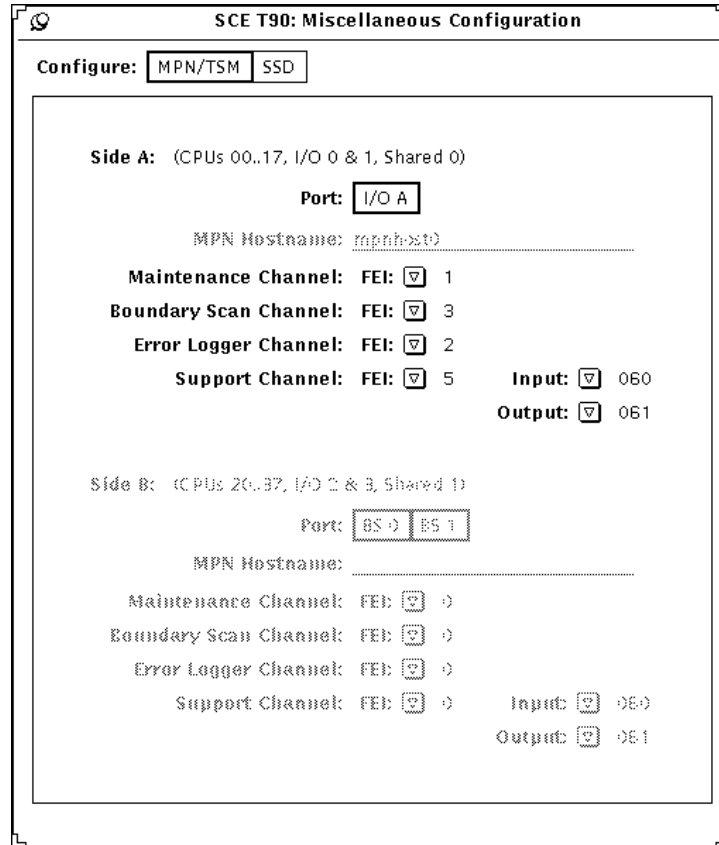


Figure 17. SCE T90: Miscellaneous Configuration Window Settings for a New CRAY T94 Configuration (Example)

NOTE: The SCE T90: Miscellaneous Configuration window provides information about the LOSPX channel connections to FEI-3 boards in the support system chassis; it does not configure these connections. If you change the LOSPX channel connections to the support system chassis, you must update the `xcfg` application because the `xcfg` application configures the proper drivers for the FEI-3 boards.

Degrading Memory in a CRAY T932 Mainframe (with CM03 Modules)

This example shows you how to degrade memory by banks, subsections, and sections. This example degrades memory in a CRAY T932 mainframe (with CM03 modules) to use banks 0, 1, 2, 3, 4, 5, 6, and 7; subsections 0, 1, 2, and 3; and sections 4, 5, 6, and 7.

Figure 18 shows how this example adds 3 address bits to force bit 9 to a 0, bit 5 to a 0, and bit 2 to a 1. These settings degrade banks 10, 11, 12, 13, 14, 15, 16, and 17; subsections 4, 5, 6, and 7; and sections 0, 1, 2, and 3. These areas in memory are no longer used. After you perform these memory degradations, only banks 0, 1, 2, 3, 4, 5, 6, and 7; subsections 0, 1, 2, and 3; and sections 4, 5, 6, and 7 are used.

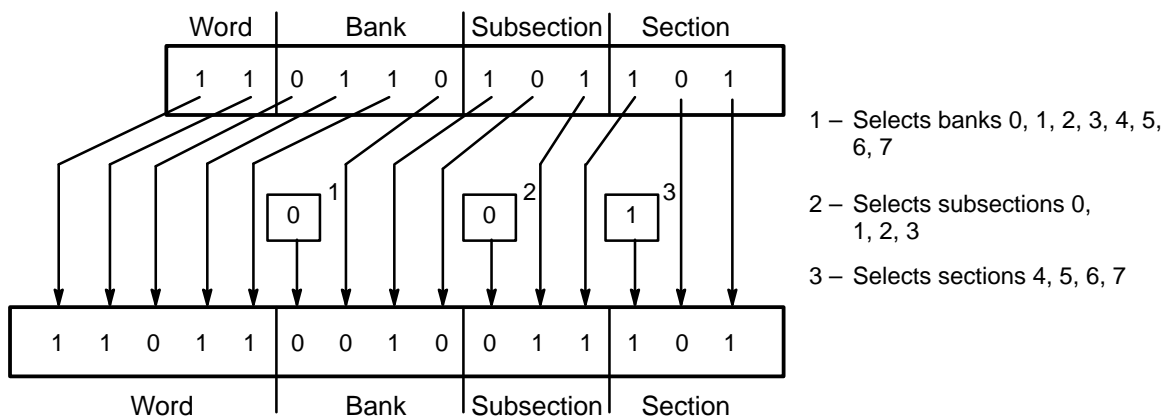


Figure 18. Example of Degrading Memory

Figure 19 shows the memory utilization map for this example.

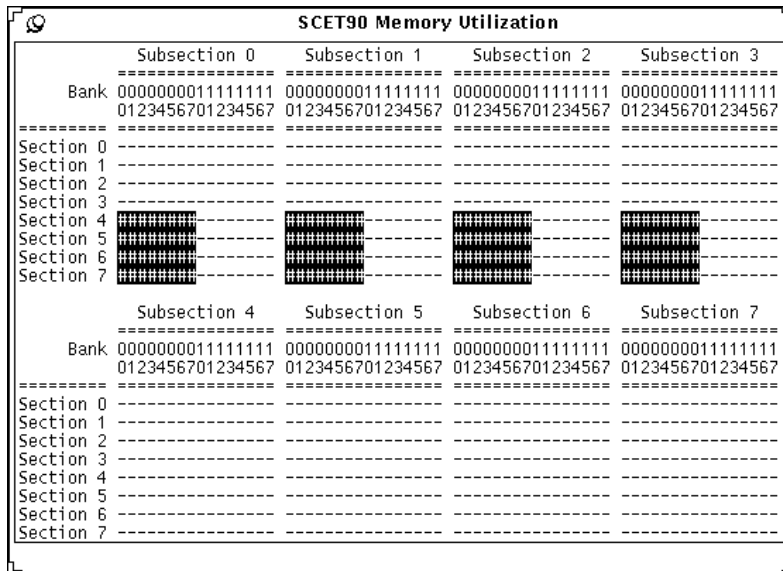


Figure 19. Example Memory Utilization Map

SCE provides two modes in the SCET90: Memory Configuration window that you can use to degrade memory: default mode and custom mode. Figure 20 on page 44 shows how to set the parameters for this example in default mode. Figure 21 on page 46 shows how to set the parameters for this example in custom mode.

Before you can perform any of the procedures shown in this subsection, you must open the SCET90 Memory window and view the memory degradation parameters. To do this, perform the following actions:

1. Choose **View -> Memory** to open the SCET90 Memory Window.
2. Click on Configure: to select the memory degradation parameters.

Memory Degradation Example (with Memory Parameters in Default Mode)

Click on Degrade Mode: Default to set the parameters in default mode. Figure 20 shows the correct parameter settings in default mode for this example.

SCET90 Memory

Configure: Modules Degrades Partitions

Physical Partition:

Memory Type: CM03 – Homogeneous

Degrade Mode: Default Custom

Sections: All 1/2

0,1,2,3
4,5,6,7

Subsections: All 1/2 1/4 1/8

0,2,4,6	0,1,4,5	0,1,2,3
1,3,5,7	2,3,6,7	4,5,6,7

Banks: All 1/2 1/4 1/8 1/16

0,1,2,3,4,5,6,7
10,11,12,13,14,15,16,17
0,1,2,3,10,11,12,13
4,5,6,7,14,15,16,17
0,1,4,5,10,11,14,15
2,3,6,7,12,13,16,17
0,2,4,6,10,12,14,16
1,3,5,7,11,13,15,17

Figure 20. Example Memory Parameters (in Default Mode)

The following procedures describe how you use the Sections, Subsections, and Banks settings in default mode for this example.

Degrading Memory by Banks (Using Default Mode)

1. Click on Banks: to specify that you want to use half of the available banks.
2. Click on to specify that you want to degrade memory by banks to use only banks 0, 1, 2, 3, 4, 5, 6, and 7.

Degrading Memory by Subsections (Using Default Mode)

1. Click on Subsections: to specify that you want to use half of the available subsections.
2. Click on to specify that you want to degrade memory by subsections to use only subsections 0, 1, 2, and 3.

Degrading Memory by Sections (Using Default Mode)

1. Click on Sections: to specify that you want to use half of the available sections.
2. Click on to specify that you want to degrade memory by sections to use only sections 4, 5, 6, and 7.

Memory Degradation Example (with Memory Parameters in Custom Mode)

Click on Degrade Mode: Custom to set the parameters in custom mode. Figure 21 shows the correct parameter settings in custom mode for this example.

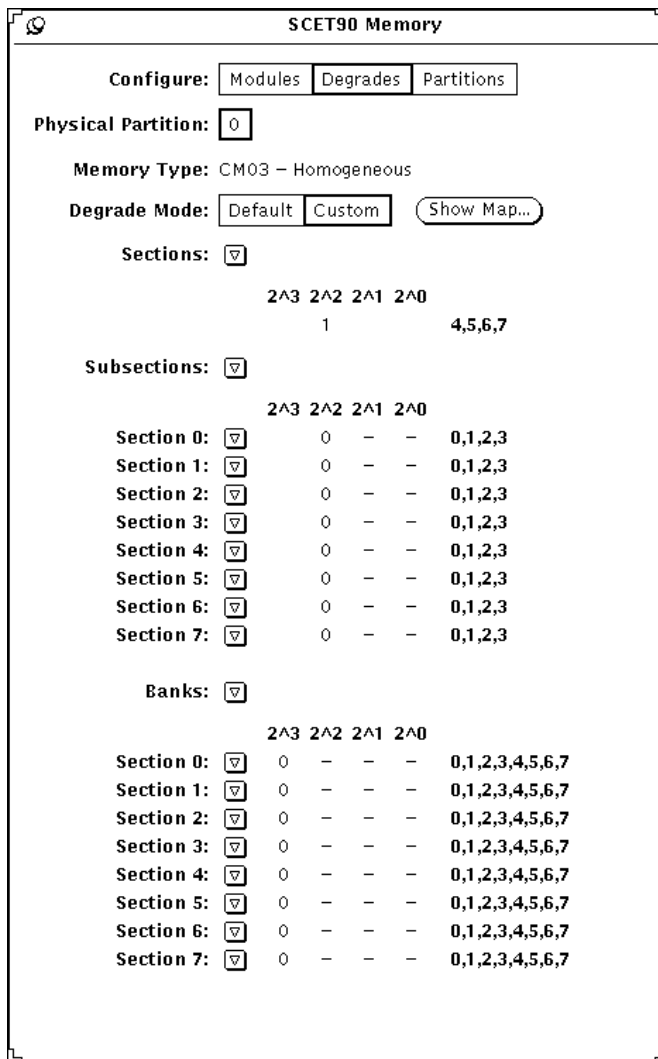


Figure 21. Example Memory Parameters (in Custom Mode)

The following procedures describe how to perform the memory degradations in custom mode for this example.

Degrading Memory by Banks (Using Custom Mode)

For this example, you can degrade memory by banks three different ways (the easiest way is shown first):

- Choose Banks: 8 of 16 → 0,1,2,3,4,5,6,7 to specify that you want to degrade memory to use only banks 0, 1, 2, 3, 4, 5, 6, and 7.
- In the Banks area of the window, for each section (Section 0 through Section 7), choose 8 of 16 → 0,1,2,3,4,5,6,7, to specify that you want to degrade memory to use only banks 0, 1, 2, 3, 4, 5, 6, and 7 for the section.
- In the Banks area of the window, set bit 2³ for each section to 0.

Degrading Memory by Subsections (Using Custom Mode)

For this example, you can degrade memory by subsections by using one of three different ways (the quickest method is shown first):

- Choose Subsections: 4 of 8 → 0,1,2,3 to specify that you want to degrade memory to use only subsections 0, 1, 2, and 3.
- In the Subsections area of the window, for each section (Section 0 through Section 7), choose 4 of 8 → 0,1,2,3 to specify that you want to degrade memory to use only subsections 0, 1, 2, and 3 for the section.
- In the Subsections area of the window, set bit 2² for each section to 0.

Degrading Memory by Sections (Using Custom Mode)

For this example, you can degrade memory by sections by using one of two different ways:

- Choose Sections: 4 of 8 → 4,5,6,7 to specify that you want to degrade memory to use only banks 4, 5, 6, and 7.
- In the Sections area of the window, set bit 2² to 1.

Degrading Memory in a CRAY T916 Mainframe (with CM02 Modules)

This example shows how to degrade memory by banks in a CRAY T916 mainframe (with CM02 memory modules) to use only the upper banks. Table 7 lists the section-to-bank mappings for the full configuration and the degraded configuration that this example uses.

Table 7. Example Section-to-bank Mappings

Section	Banks	
	Full Configuration	Degraded Configuration
0	0, 1, 4, 5	4, 5
1	0, 1, 4, 5	4, 5
2	2, 3, 6, 7	6, 7
3	2, 3, 6, 7	6, 7
4	0, 1, 4, 5	4, 5
5	0, 1, 4, 5	4, 5
6	2, 3, 6, 7	6, 7
7	2, 3, 6, 7	6, 7

This example shows how to degrade memory in default mode and in custom mode.

Using Default Mode

In default mode, click on Banks: 1,2 and then click on 4,5,6,7. SCE automatically configures the mainframe to use the section-to-bank mappings that the “Degraded Configuration” column of Table 7 lists.

Using Custom Mode

In custom mode, perform the following steps in the Banks portion of the SCE T90: Memory Configuration window (Figure 22 shows a window with the correct settings for this example):

1. Choose Section 0: 2 of 16 → 4,5 to map section 0 to banks 4 and 5.
2. Choose Section 1: 2 of 16 → 4,5 to map section 1 to banks 4 and 5.

3. Choose Section 2: 2 of 16 → 6,7 to map section 2 to banks 6 and 7.
4. Choose Section 3: 2 of 16 → 6,7 to map section 3 to banks 6 and 7.
5. Choose Section 4: 2 of 16 → 4,5 to map section 4 to banks 4 and 5.
6. Choose Section 5: 2 of 16 → 4,5 to map section 5 to banks 4 and 5.
7. Choose Section 6: 2 of 16 → 6,7 to map section 6 to banks 6 and 7.
8. Choose Section 7: 2 of 16 → 6,7 to map section 7 to banks 6 and 7.

Banks: <input type="checkbox"/>					
	2 [^] 3	2 [^] 2	2 [^] 1	2 [^] 0	
Section 0: <input type="checkbox"/>	0	1	0	-	4,5
Section 1: <input type="checkbox"/>	0	1	0	-	4,5
Section 2: <input type="checkbox"/>	0	1	1	-	6,7
Section 3: <input type="checkbox"/>	0	1	1	-	6,7
Section 4: <input type="checkbox"/>	0	1	0	-	4,5
Section 5: <input type="checkbox"/>	0	1	0	-	4,5
Section 6: <input type="checkbox"/>	0	1	1	-	6,7
Section 7: <input type="checkbox"/>	0	1	1	-	6,7

Figure 22. Bank Memory Parameters (Example)

Setting up a Partition for Maintenance

This example shows how to create a logical partition in which you can run maintenance software while another partition of the mainframe continues to run the operating system. Table 8 shows the physical components that each logical partition contains.

Table 8. Example Configuration Components

Partition	Memory	CPUs	Clusters	Channels
Logical partition 0 (OS owner)	64 Mwords	0, 1	1, 2, 3, 4, 5, 6, 7, 10, 11 (octal)	LOSP: 100/101 102/103 104/105 106/107 060/061 HISP: 400 401 402 403 VHISP: 020 021
Logical partition 1 (Maintenance owner)	64 Mwords	2, 3	12, 13, 14, 15, 16, 17, 20, 21, 22 (octal)	LOSP: 110/111 112/113 114/115 116/117 HISP: 404 405 406 407 VHISP: 022 023

The following procedures indicate the steps that you need to perform to create this example configuration:

1. Modify the system information parameters to indicate the serial number, amount of system memory, and number of physical partitions.

Table 9 and Figure 23 show the system information parameter settings that are necessary for this example.

Table 9. System Information Parameter Settings (Example)

Parameter	Setting
System Type	T94
System Serial	7001
System Memory	128M
Physical Partitions	1 (for CRAY T94 mainframe)

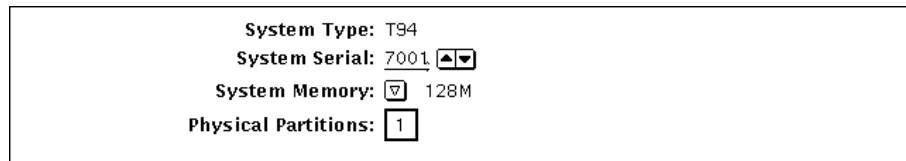


Figure 23. System Information Parameters (Example)

2. Set up the configuration information for the current physical partition.

Table 10 and Figure 24 show the physical partition parameter settings for this example.

Table 10. Physical Partition Parameter Settings (Example)

Parameter	Setting
Current Physical Partition †	0 (CRAY T94 mainframes have only one physical partition)
Sanity Port †	Left at the default for this example
Sparechips †	Left at the default for this example
Logical Partitions †	2 (to create the two logical partitions)
Maintenance Channel ‡	Left at the default for this example
Boundary Scan Channel ‡	Left at the default for this example
Error Logger Channel ‡	Left at the default for this example
Support Channel ‡	Left at the default for this example
Input ‡	Left at the default for this example
Output ‡	Left at the default for this example

† These parameters are located in the SCE base window.

‡ These parameters are located in the SCE T90: Miscellaneous Configuration window. Choose **View -> Miscellaneous** to access this window.

Physical Partition Parameters in the SCE Base Window

Current Physical Partition:

Sanity Port:

Sparechips: Disabled Enabled

Logical Partitions:

Physical Partition Parameters in the SCE T90: Miscellaneous Configuration Window

SCE T90: Miscellaneous Configuration

Configure:

Side A: (CPUs 00..17, I/O 0 & 1, Shared 0)

Port:

MPN Hostname:

Maintenance Channel: FEI:

Boundary Scan Channel: FEI:

Error Logger Channel: FEI:

Support Channel: FEI: Input: Output:

Figure 24. Physical Partition Parameters (Example)

NOTE: The SCE T90: Miscellaneous Configuration window provides information about the LOSPX channel connections to FEI-3 boards in the support system chassis; it does not configure these connections. If you change the LOSPX channel connections to the support system chassis, you must update the `xcfg` application because the `xcfg` application configures the proper drivers for the FEI-3 boards.

3. Modify the logical partition parameters to create each logical partition.

For this example, you need to create two logical partitions (logical partition 0 and logical partition 1). Table 11 and Figure 25 show the logical partition parameter settings for logical partition 0.

Table 11. Logical Partition 0 Parameter Settings (Example)

Parameter	Setting
Current Logical Partition	0 (to define parameters for logical partition 0)
Memory	64 Mwords (SCE divided memory in half and configured half to each of the two logical partitions)
Owner	OS
Maintenance	Online and concurrent maintenance enabled
Description	<i>Left at the default for this example</i>
CPUs	0 and 1
Cluster Offset	1
Cluster Range	11 ₈ (selects clusters 1, 2, 3, 4, 5, 6, 7, 10, and 11)
Auto BCD	<i>Left at the default for this example</i>
LOSPs	100/101, 102/103, 104/105, 106/107 060/061
HISPs	400, 401, 402, 403
VHISPs	020, 021

Current Logical Partition:

Memory: 64M

Owner:

Maintenance:

Description: PPO, LPO

CPUs:

00	04	08	12	16	20	24	28	32	36
01	05	11	15	21	25	31	35		
02	06	10	14	22	26	30	34		
03	07	13	17	23	27	33	37		

Cluster Offset: 1

Cluster Range: 11

Auto BCD:

LOSPs:

100/101	120/121	140/141	160/161
102/103	122/123	142/143	162/163
104/105	124/125	144/145	164/165
106/107	126/127	146/147	166/167
110/111	130/131	150/151	170/171
112/113	132/133	152/153	172/173
114/115	134/135	154/155	174/175
116/117	136/137	156/157	176/177
060/061	062/063	064/065	066/067

HISPs:

400	410	420	430
401	411	421	431
402	412	422	432
403	413	423	433
404	414	424	434
405	415	425	435
406	416	426	436
407	417	427	437

VHISPs:

020	030
021	031
022	032
023	033
024	034
025	035
026	036
027	037

New configuration – apply/assert required.

Figure 25. Logical Partition 0 Parameters (Example)

Table 12 and Figure 26 show the logical partition parameter settings for logical partition 1.

NOTE: To switch to logical partition 1, click on Current Logical Partition: .

Table 12. Logical Partition 1 Parameter Settings (Example)

Parameter	Setting
Current Logical Partition	1 (to define parameters for logical partition 1)
Memory	64 Mwords (SCE divided memory in half and configured half to each of the two logical partitions)
Owner	Maintenance
Description	<i>Left at the default for this example</i>
CPUs	2 and 3
Cluster Offset	12
Cluster Range	11g (selects clusters 12, 13, 14, 15, 16, 17, 20, 21, 22)
Auto BCD	<i>Left at the default for this example</i>

Table 12. Logical Partition 1 Parameter Settings (Example) (continued)

Parameter	Setting
LOSPs	110/111, 112/113, 114/115, 116/117
HISPs	404, 405, 406, 407
VHISPs	022, 023

Current Logical Partition:

Memory: 64M

Owner:

Maintenance:

Description: PPO, LP1

CPUs:

00	04	08	12	16	20	24	28	32	36
01	05	11	15	21	25	31	35		
02	06	10	14	22	26	30	34		
03	07	13	17	23	27	33	37		

Cluster Offset: 12

Cluster Range: 11

Auto BCD:

LOSPs:	HISPs:	VHISPs:
<input type="text" value="100/101"/> <input type="text" value="120/121"/> <input type="text" value="140/141"/> <input type="text" value="160/161"/>	<input type="text" value="400"/> <input type="text" value="410"/> <input type="text" value="420"/> <input type="text" value="430"/>	<input type="text" value="020"/> <input type="text" value="030"/>
<input type="text" value="102/103"/> <input type="text" value="122/123"/> <input type="text" value="142/143"/> <input type="text" value="162/163"/>	<input type="text" value="401"/> <input type="text" value="411"/> <input type="text" value="421"/> <input type="text" value="431"/>	<input type="text" value="021"/> <input type="text" value="031"/>
<input type="text" value="104/105"/> <input type="text" value="124/125"/> <input type="text" value="144/145"/> <input type="text" value="164/165"/>	<input type="text" value="402"/> <input type="text" value="412"/> <input type="text" value="422"/> <input type="text" value="432"/>	<input type="text" value="022"/> <input type="text" value="032"/>
<input type="text" value="106/107"/> <input type="text" value="126/127"/> <input type="text" value="146/147"/> <input type="text" value="166/167"/>	<input type="text" value="403"/> <input type="text" value="413"/> <input type="text" value="423"/> <input type="text" value="433"/>	<input type="text" value="023"/> <input type="text" value="033"/>
<input type="text" value="110/111"/> <input type="text" value="130/131"/> <input type="text" value="150/151"/> <input type="text" value="170/171"/>	<input type="text" value="404"/> <input type="text" value="414"/> <input type="text" value="424"/> <input type="text" value="434"/>	<input type="text" value="024"/> <input type="text" value="034"/>
<input type="text" value="112/113"/> <input type="text" value="132/133"/> <input type="text" value="152/153"/> <input type="text" value="172/173"/>	<input type="text" value="405"/> <input type="text" value="415"/> <input type="text" value="425"/> <input type="text" value="435"/>	<input type="text" value="025"/> <input type="text" value="035"/>
<input type="text" value="114/115"/> <input type="text" value="134/135"/> <input type="text" value="154/155"/> <input type="text" value="174/175"/>	<input type="text" value="406"/> <input type="text" value="416"/> <input type="text" value="426"/> <input type="text" value="436"/>	<input type="text" value="026"/> <input type="text" value="036"/>
<input type="text" value="116/117"/> <input type="text" value="136/137"/> <input type="text" value="156/157"/> <input type="text" value="176/177"/>	<input type="text" value="407"/> <input type="text" value="417"/> <input type="text" value="427"/> <input type="text" value="437"/>	<input type="text" value="027"/> <input type="text" value="037"/>
<input type="text" value="060/061"/> <input type="text" value="082/083"/> <input type="text" value="084/085"/> <input type="text" value="086/087"/>		

New configuration – apply/assert required.

Figure 26. Logical Partition 1 Parameters (Example)

HDM-069-C

Cray Research/Silicon Graphics Proprietary

55

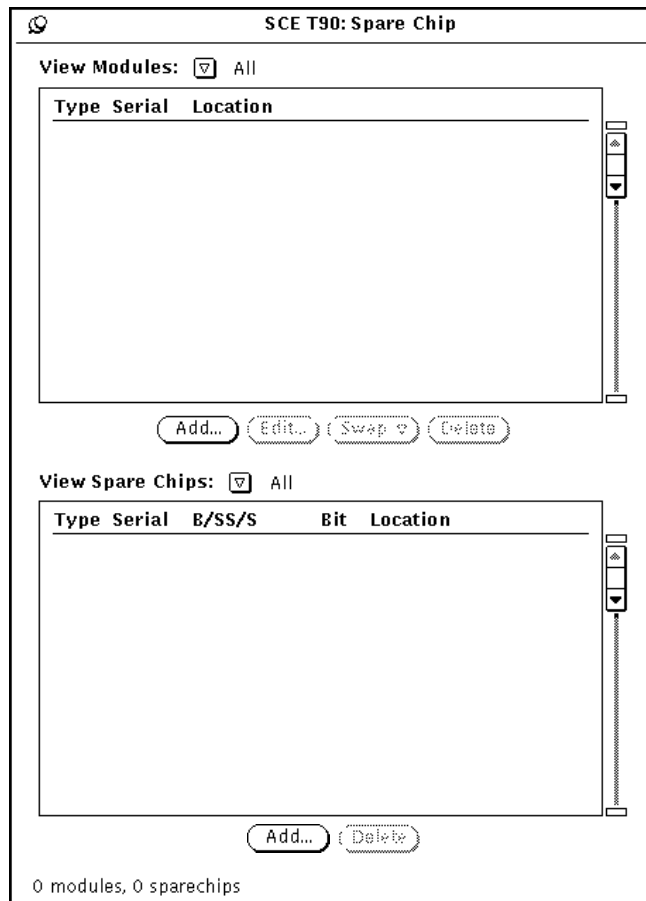
Flawing a Bad Memory Chip

The following example shows how to enter the bad bit information that Table 13 lists so that SCE will replace the flawed chip with a spare chip.

Table 13. Bad Bit Location Information

Component	Value
Module location	In a CRAY T94 mainframe (serial number 7001 [slot C5])
Module	CM02 module (serial number 117)
Bank	3
Subsection	0
Section	7
Bad bit	51

1. Choose **View -> Spare Chip** in the SCE base window. The SCE T90: Spare Chip window appears:



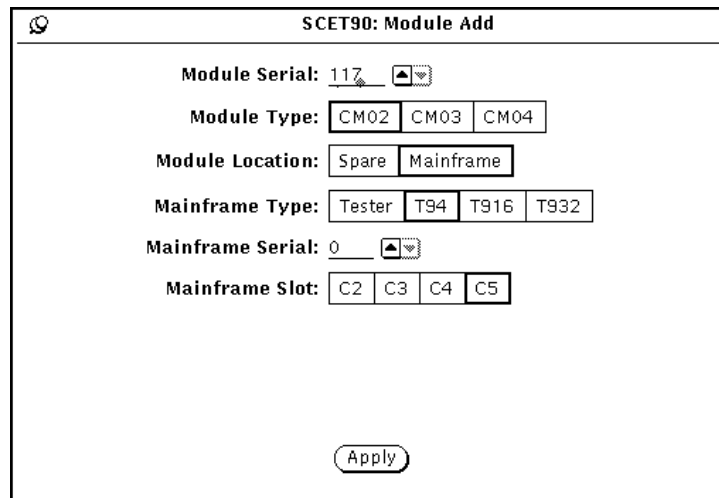
2. In the SCE T90: Spare Chip window, click on **Add...** (located below the View Modules scroll box) to add a module entry to the table. SCE displays the SCE T90: Module Add window:

The screenshot shows a window titled "SCET90: Module Add". It contains the following fields and controls:

- Module Serial:** A text field containing "0" and a dropdown arrow.
- Module Type:** Three buttons labeled "CM02", "CM03", and "CM04".
- Module Location:** Two buttons labeled "Spare" and "Mainframe".
- Mainframe Type:** Four buttons labeled "Tester", "T9-1", "T9-16", and "T9-32".
- Mainframe Serial:** A text field containing "0" and a dropdown arrow.
- Mainframe Slot:** Four buttons labeled "C2", "C3", "C4", and "C5".
- Apply:** A button at the bottom center.

3. In the Module Serial field, enter **117** (the serial number of the module in this example).
4. Click on Module Type: **CM02**.
5. Perform the following actions to enter the module location information for this example:
 - a. Click on Module Location: **Mainframe**.
 - b. Click on Mainframe Type: **T9-1**.
 - c. Enter **7001** in the Mainframe Serial field.
 - d. Click on Mainframe Slot: **C5**.

The following snap shows the SCE T90: Module Add window with the correct values for this example:

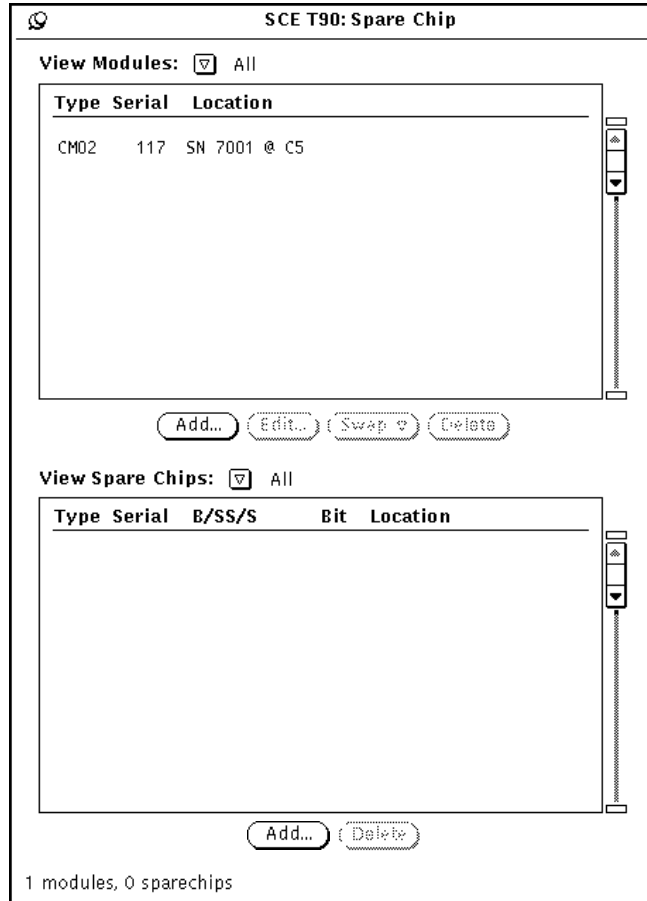


The screenshot displays a window titled "SCET90: Module Add". It contains several configuration fields:

- Module Serial:** A text field containing "117" and a small upward/downward arrow icon.
- Module Type:** A set of three buttons labeled "CM02", "CM03", and "CM04".
- Module Location:** A set of two buttons labeled "Spare" and "Mainframe".
- Mainframe Type:** A set of four buttons labeled "Tester", "T94", "T916", and "T932".
- Mainframe Serial:** A text field containing "0" and a small upward/downward arrow icon.
- Mainframe Slot:** A set of four buttons labeled "C2", "C3", "C4", and "C5".

At the bottom center of the window is an "Apply" button.

6. In the SCE T90: Module Add window, click on **Apply** to insert the module information into the table. SCE adds the module information to the View Modules scroll box in the SCE T90: Spare Chip window, as shown in the following snap:



- In the SCE T90: Spare Chip window, click on (located below the View Spare Chips scroll box) to add a flaw-chip entry to the table. SCE displays the SCE T90: Spare Chip Add window:

SCET90: Spare Chip Add

Location:

Bank:

0	1	2	3	4	5	6	7
10	11	12	13	14	15	16	17

Subsection:

0	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---

Section:

0	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---

Bad Bit:

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49
50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69
70	71	72	73	74	75				

- In the SCE T90: Spare Chip Add window, from the Location: , choose **Mainframe -> SN 7001** to indicate that the bad chip is on a module in the mainframe.
- Click on Bank: .

NOTE: The subsection parameter does not apply to CRAY T94 mainframes because they do not have subsections.

- Click on Section: .
- Click on Bad Bit: .

The following snap shows the SCE T90: Spare Chip Add window with the correct values for this example:

SCET90: Spare Chip Add

Location:

Bank:

0	1	2	3	4	5	6	7
10	11	12	13	14	15	16	17

Subsection:

0	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---

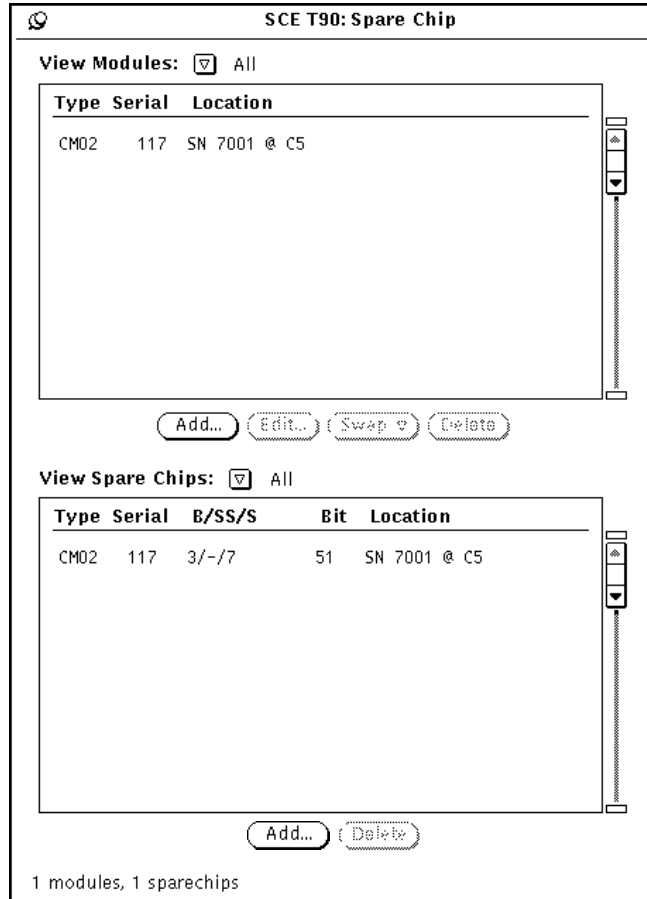
Section:

0	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---

Bad Bit:

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49
50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69
70	71	72	73	74	75				

- In the SCE T90: Spare Chip Add window, click on **Add** to insert the flawed-chip information into the table. SCE adds the failing chip information to the View Spare Chips scroll box in the SCE T90: Spare Chip window:



SCE will substitute a spare chip for the flawed chip the next time you assert the configuration if the spare chip option is enabled for the physical partition that contains the flawed chip.