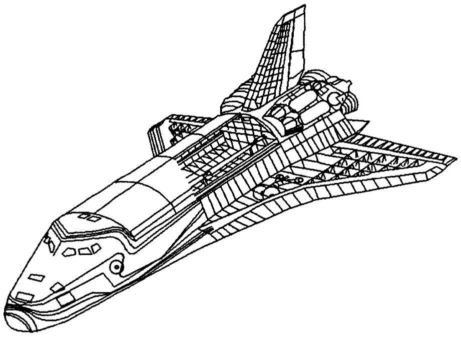
# Inside the ™MPX-32 Operating System

## A System Analysts Course

**Published by:**

*Simulation Technologies, Inc.*
*Desktop Publishing Division*
*44915 Laszlo Street*
*Lancaster, CA 93534*



First Printing, September 1985

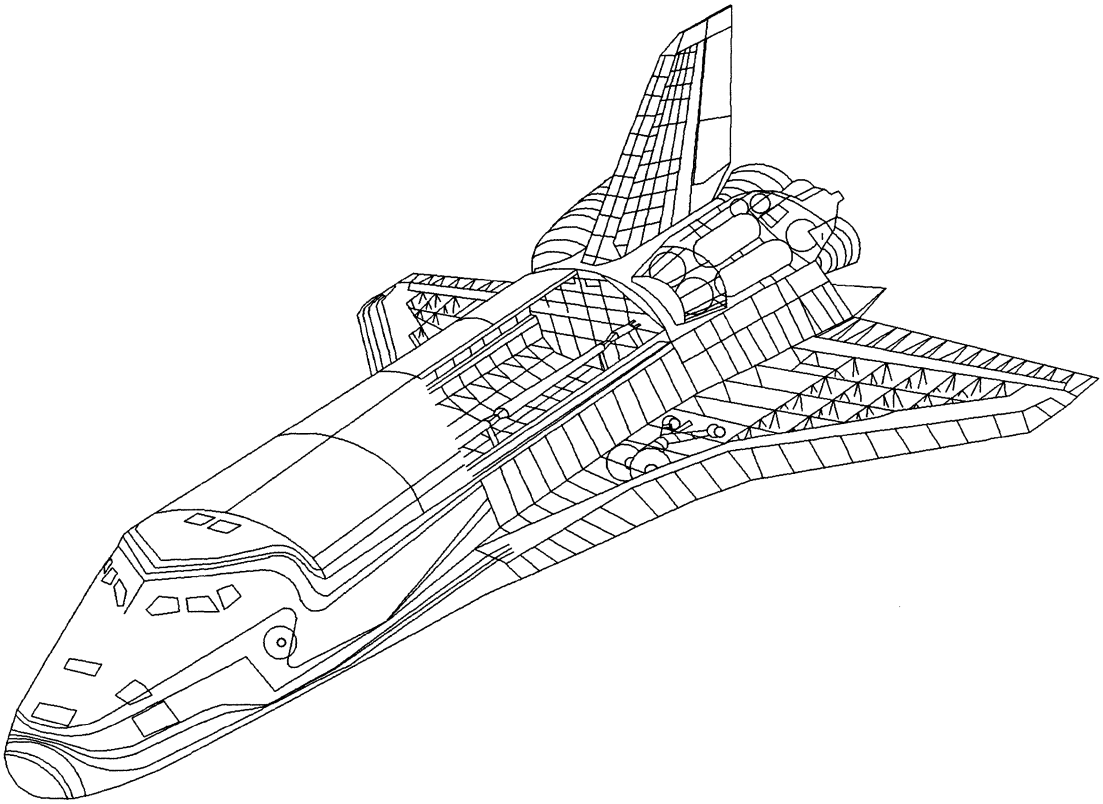
Printed in the United States of America

This book is designed to provide information about the MPX-32™ operating system. Every Effort has been made to make this book complete and as accurate as possible. But no warranty of suitability, purpose, or fitness is implied. The information is provided on an "as-is" basis. The author shall have neither liability nor responsibility to any person or entity with respect to any loss or damages in connection with or arising from the information contained in this book.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where these designations appear in the book and the author was aware of a trademark claim, the designations have been printed with initial capital letters - for example, MPX-32 or NP1.

# SIMULATION TECHNOLOGIES, INC.

## NASA / NORTHROP/ STERLING Training Course
## Student Workbook



SIMULATION TECHNOLOGIES, INC.
P.O. Box 2452
Lancaster, CA 93539-2452
(805) 949-6831

# Table of Contents

## SECTION 1 TRAINING INFORMATION

## SECTION 2 32/7X ARCHITECTURE

# TRAINING INFORMATION

# 1. STUDENT INFORMATION

The following information will describe the intent, goals and expectations for this 3 week training period. Please read this section carefully, then complete the trainee record found in this section of the training workbook.

## 2. TRAINING PHILOSOPHY

Simulation Technologies, Inc. believes that laboratory intensive training periods yield individuals better able to support and maintain Gould Inc. computer systems. In order for the student to successfully perform in the laboratory environment, it is imperative the material presented during the conference portion of the course be mastered.

It is presumed that all individuals enrolled in this course have at least 6 months of experience in maintaining and/or programming SERIES 32/7X and CONCEPT 32 computer systems. Though a review of basic system features has been included, it will be the students responsibility to seek remedial instruction from the instructor should a problem arise.

Your performance during this course will be primarily measured during the laboratory exercises. Should you not be able to attend a particular laboratory session, please inform the instructor.

# 3. COURSE OBJECTIVE

This course is designed for Systems Analysts and/or maintenance personnel charged with maintaining SERIES 32 and CONCEPT 32 computer systems, utilizing the MPX-32 1.X operating system.

This course will hopefully provide each individual information that will enable that individual to better perform his/her job. The course will focus on special techniques for problem isolation and remediation in the software. Though the course contains review material, it is intended for reference purposes, and is not intended to be introductory in nature. In addition, many of the topics listed in the hardware section will not be covered in detail, as this information will be fully covered in the software portion of the course.

*The success of this course depends upon each and everyone participation in the conference and laboratory sessions. We are here primarily to answer your questions and demonstrate techniques in fault isolation and remediation. It is imperative that you take excellent notes and ask numerous questions should you experience difficulty with the course material and/or concepts presented herein.*

This information contained in the workbook is to be used as a guideline in the course. If information is not of interest to the class, then alternate material may be prepared for class presentation. Though your supervisors have indicated to us those areas of primary interest, we will try to entertain other area you may find of interest. Please let us know as soon as possible during the course if you have a topic of particular interest and may apply to the group as a whole.

# COURSE SCHEDULE

MPX-32 REVISION 1.X

SIMULATION TECHNOLOGIES, INC.

| | MONDAY | TUESDAY | WEDNESDAY | THURSDAY | FRIDAY |
|---|---|---|---|---|---|
| 9:00 | Introduction | MPX Overview | TSM | Job Control Language | AJCLGEN |
| | Course Overview | | | | |
| 10:00 | 32/7X Architecture | MPX Overview | TSM | Job Control language | AJCLGEN |
| 11:00 | 32/7X Architecture | MPX Overview | TSM | Job Control language | AJCLGEN |
| 12:00 | LUNCH | LUNCH | LUNCH | LUNCH | LUNCH |
| 1:00 | 32/7x Architecture | MPX OVERVIEW | TSM | Job Control Language | COURSE REVIEW |
| | | | Lab Exercise | Lab Exercise | COMPLETE LAB |
| 2:00 | | | | | EXERCISES |
| 3:00 | | | | | |
| 4:00 | | | | | |

# 32/7X ARCHITECTURE

# 1. OVERVIEW

In order for the student to correctly operate the SERIES 32/7X computer system, a thorough knowledge of hardware fundamentals must be achieved. In other words, in order to generate efficient and functional programs you must know the limitations of the hardware.

In this section we will examine the components of the SERIES 32/7X hardware. At then end of this section, you should be able to answer the following questions.

- *How many general purpose registers are contained in each 32/7X central processing unit? How are these registers used?*
- *How many data lines are incorporated into the SelBUS?*
- *How much physical memory can the 32/7X CPU address?*
- *Is an RTOM required on a SERIES 32/7X if an interval timer is be available for CPU accounting?*
- *Does the SERIES 32/7X support an IPU?* Internal Processing Unit 2 are supported
- *How many external interrupt levels are available on each IOP? how many on each RTOM?*
- *How many interrupt levels can be configured on a SERIES 32/7X computer system?* 112

# 2. OBJECTIVES

In order for the student generate efficient programs on the SERIES 32/7X computer systems, a fundamental knowledge of system hardware features is essential. *Please insure the following objectives have been achieved prior to beginning the system characteristics laboratory exercises.*

- ¤ **The student will answer several questions concerning SERIES 32/7X hardware.**

- ¤ **The student will identify several components, subsystems and plug-in modules in the SERIES 32/7X computer.**

# 3. ABBREVIATIONS

| | |
|---|---|
| ° **HSD** | High Speed Data (Interface) 32 bit parallel |
| ° **CPU** | Central Processing Unit |
| ° **IOP** | Input/Output Processor |
| ° **FPA** | Floating Point Accelerator |
| ° **RTOM** | Real-time Option Module |
| ° **GPR** | General Purpose Register |
| ° **PSD** | Program Status Doubleword |
| ° **PSW** | Program Status Word |

# 4. DEFINITIONS

° **Program Status Doubleword**

2 32-bit registers in the CPU which describes the context of the currently executing task and/or the MPX-32 operating system. Such information includes privilege state, condition codes, program counter, mapping information, and interrupt blocking states.

° **Program Status Word**

The first 32 bits of the PSD. PSW mode is used in the RTM operating System where no memory mapping is provided.

# 5. WHAT YOU NEED TO KNOW

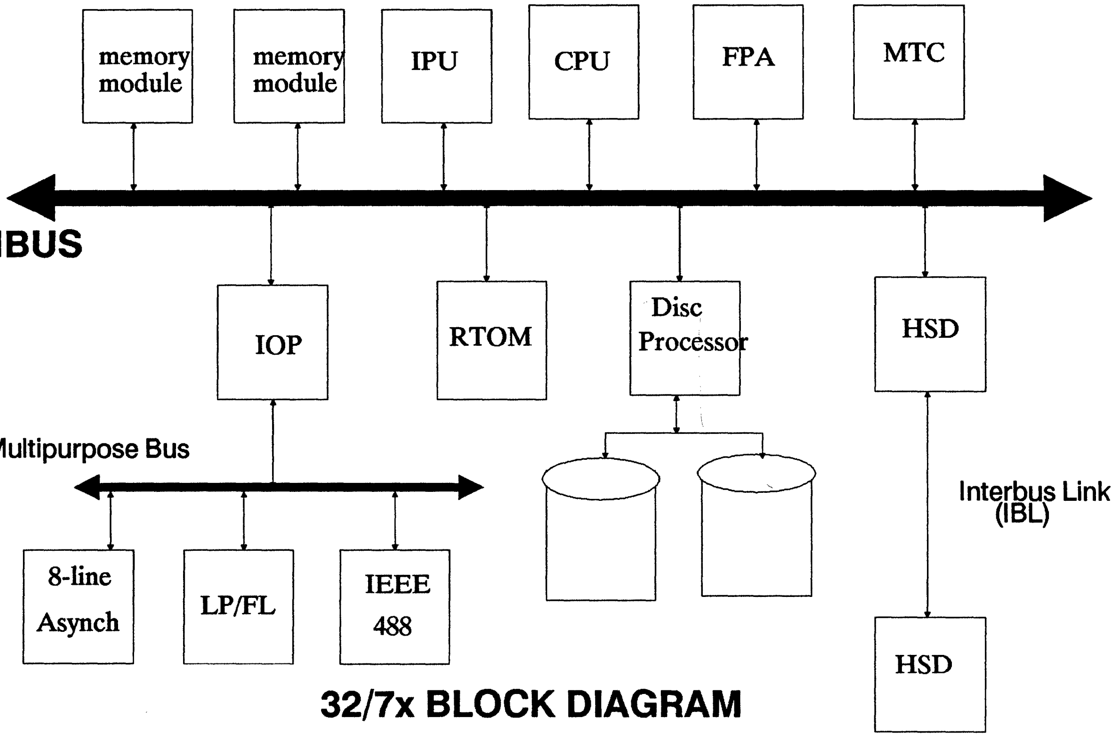
## 5.1. In a Nutshell

This section will present the basic features and concepts of the SERIES 32/7X computer. The information provided in this section is fundamental to all levels of programming including device handlers, timing tasks, and FORTRAN programs that utilize inline assembly code.

The 32/7X is a high-performance 32-bit computing system built around the SelBUS, a high-speed, synchronous, shared, bi-directional, multiplexed bus. The SelBUS provides the path for communication between all functional elements of the system. It runs vertically through the logic chassis and distributes information at a rate of 26.67 million bytes per second.

The modules that plug into the SelBUS include:

- **Central Processing Unit (CPU)**
- **High-speed Floating-Point Option (FPA)**
- **Real-time Option Module (RTOM)**
- **Input/Output Microprogrammable Processors (IOM)**
- **Input/Output Processor (IOP)**
- **Regional Processing Units (RPU)**



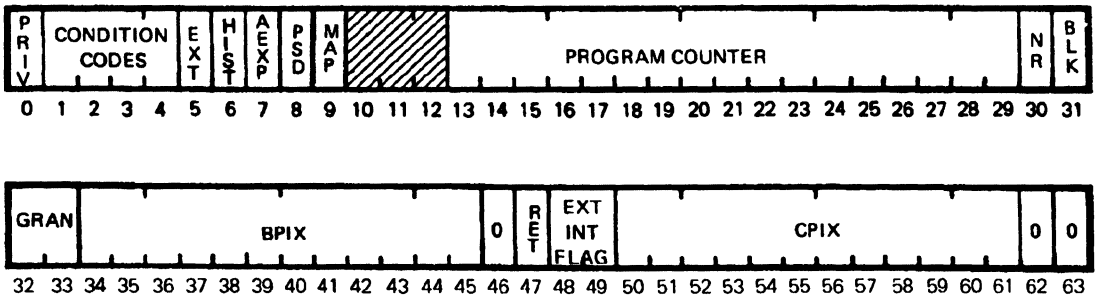
32/7x BLOCK DIAGRAM

## 5.1.1. SelBUS

- **184 bidirectional lines**
- **150 ns cycle time** [illegible]
- **32 data lines**
- **24 address lines**
- **26.67 MB transfer rate**
- **112 software controllable interrupt levels**

1 micro inst / word = 1 cycle

# PROGRAM STATUS DOUBLEWORD (PSD)



BIT 0 = 0 UNPRIVILEGED MODE
= 1 PRIVILEGED MODE

BITS 1-4 ARE CONDITION CODES
BIT 1 = CC1
2 = CC2
3 = CC3
4 = CC4

BIT 5 = 0 EXTENDED MODE (OFF) CEA
= 1 EXTENDED MODE (ON) SEA

BIT 6 = 0 LAST INSTRUCTION EXECUTED WAS NOT A RIGHT HALFWORD
= 1 LAST INSTRUCTION EXECUTED WAS A RIGHT HALFWORD

BIT 7 = 0 ARITHMETIC EXCEPTION TRAP MASK (OFF)
= 1 ARITHMETIC EXCEPTION TRAP MASK (ON)

*BIT 8 = 0 COMPUTER IS IN PSW MODE (DISPLAYED PSD ONLY)*
= 1 COMPUTER IS IN PSD MODE (DISPLAYED PSD ONLY)*

*BIT 9 = 0 UNMAPPED (DISPLAYED PSD ONLY) *
= 1 MAPPED (DISPLAYED PSD ONLY) *

BITS 10-12 ARE NOT USED
BITS 13-29 ARE LOGICAL WORD ADDRESS
BIT 30 NEXT INSTRUCTION IS A RIGHT HALFWORD
*BIT 31 BLOCKED (DISPLAYED PSD ONLY) *
BITS 32-33 INDICATE MAP GRANULARITY, 00=UNMAPPED AND ALL OTHERS =8K MAP GRANULARITY
BITS 34-45 PROVIDE A WORD INDEX INTO THE MASTER PROCESS LIST (MPL) FOR THE BASE PROCESS
BIT 46 NOT USED
BIT 47 RETAIN CURRENT MAP CONTENTS
BITS 48-49 INTERRUPT CONTROL FLAGS

| BITS | | |
|---|---|---|
| 48 | 49 | |
| 0 | 0 | OPERATE WITH UNBLOCKED INTERRUPTS |
| 0 | 1 | OPERATE WITH BLOCKED INTERRUPTS |
| 1 | 0 | RETAIN CURRENT BLOCKING MODE |
| 1 | 1 | RETAIN CURRENT BLOCKING MODE |

BITS 50-61 PROVIDE WORD INDEX INTO MASTER PROCESS LIST (MPL) FOR CURRENT PROCESS
BITS 62-63 NOT USED

* THESE BITS ARE USED FOR DISPLAY ONLY AND ARE NOT PRESENT IN THE PSD STORED IN MEMORY

## 5.1.2. Central Processing Unit

- **Implemented on three plug-in boards**
- **Two boards are Arithmetic/Logic Unit**
- **One board is Control Unit (firmware)**

## 5.1.3. Registers

- **8 high-speed, 32-bit general purpose registers.**
- **R0 - used to contain PSW during Branch & Link operations.**
- **R1,R2,R3 - used for indexing operations.**
- **R4 - used for masking operations.**
- **R5-R7 General purpose registers.**
- **all registers can be used arithmetic, logical and shift operations.**

PSW/PSD Mode Relative Trap/Interrupt Priorities

| RELATIVE PRIORITY | LOGICAL PRIORITY | INTERRUPT VECTOR LOCATION (IVL) | TCW ADDRESS ** | IOCD ADDRESS ** | DESCRIPTION |
|---|---|---|---|---|---|
| 00 | | 0F4 | | | Power Fail Safe Trap |
| 01 | | 0FC | | | System Override Trap (Not used) |
| 02 | | 0E8* | | | Memory Parity Trap |
| 03 | | 190 | | | Nonpresent Memory Trap |
| 04 | | 194 | | | Undefined Instruction Trap |
| 05 | | 198 | | | Privilege Violation Trap |
| 06 | | 180 | | | Supervisor Call Trap |
| 07 | | 184 | | | Machine Check Trap |
| 08 | | 188 | | | System Check Trap |
| 09 | | 18C | | | MAP Fault Trap |
| 0A | | | | | Not Used |
| 0B | | | | | Not Used |
| 0C | | | | | Not Used |
| 0D | | | | | Not Used |
| 0E | | 0E4 | | | Block Mode Timeout (Watchdog) Trap |
| 0F | | 1A4* | | | Arithmetic Exception Trap |
| 10 | 00 | 0F0 | | | Power Fail Safe Interrupt |
| 11 | 01 | 0F8 | | | System Override Interrupt |
| 12 | 12 | 0E8* | | | ***Memory Parity Trap |
| 13 | 13 | 0EC | | | Attention Interrupt |
| 14 | 14 | 140 | 100 | 700 | I/O Channel 0 Interrupt |
| 15 | 15 | 144 | 104 | 708 | I/O Channel 1 Interrupt |
| 16 | 16 | 148 | 108 | 710 | I/O Channel 2 Interrupt |
| 17 | 17 | 14C | 10C | 718 | I/O Channel 3 Interrupt |
| 18 | 18 | 150 | 110 | 720 | I/O Channel 4 Interrupt |
| 19 | 19 | 154 | 114 | 728 | I/O Channel 5 Interrupt |
| 1A | 1A | 158 | 118 | 730 | I/O Channel 6 Interrupt |
| 1B | 1B | 15C | 11C | 738 | I/O Channel 7 Interrupt |
| 1C | 1C | 160 | 120 | 740 | I/O Channel 8 Interrupt |
| 1D | 1D | 164 | 124 | 748 | I/O Channel 9 Interrupt |
| 1E | 1E | 168 | 128 | 750 | I/O Channel A Interrupt |
| 1F | 1F | 16C | 12C | 758 | I/O Channel B Interrupt |
| 20 | 20 | 170 | 130 | 760 | I/O Channel C Interrupt |
| 21 | 21 | 174 | 134 | 768 | I/O Channel D Interrupt |
| 22 | 22 | 178 | 138 | 770 | I/O Channel E Interrupt |
| 23 | 23 | 17C | 13C | 778 | I/O Channel F Interrupt |
| 24 | 24 | 190* | | | ***Nonpresent Memory Trap |
| 25 | 25 | 194* | | | ***Undefined Instruction Trap |
| 26 | 26 | 198* | | | ***Privilege Violation Trap |
| 27 | 27 | 19C | | | Call Monitor Interrupt |
| 28 | 28 | 1A0 | | | Real-Time Clock Interrupt |
| 29 | 29 | 1A4* | | | ***Arithmetic Exception Interrupt |
| 2A | 2A | 1A8 | | | External/Software Interrupts |
| 2B | 2B | 1AC | | | External/Software Interrupts |
| 2C | 2C | 1B0 | | | External/Software Interrupts |
| 2D | 2D | 1B4 | | | External/Software Interrupts |
| 2E | 2E | 1B8 | | | External/Software Interrupts |
| 2F | 2F | 1BC | | | External/Software Interrupts |
| 30 | 30 | 1C0 | | | External/Software Interrupts |
| 31 | 31 | 1C4 | | | External/Software Interrupts |
| THRU | THRU | THRU | | | THRU |
| 77 | 77 | 2DC | | | External/Software Interrupts |

| | |
|---|---|
| * | Vector Locations Shared With Traps |
| ** | For Nonextended I/O Devices |
| *** | PSW Function-Now External/Software Interrupts-For PSD Mode |
| **** | IPU Related Traps |

All Interrupts Are Externally Generated

## 5.1.4. Memory

- The 32/7X computer uses volatile 600 ns MOS memory.
- Memory organized into 36-bit words (32-bits data and 4 bits parity).
- Memory Bus controller (MBC) provides interface between the SelBUS and the memory bus.
- Each memory chassis can support up to eight memory modules.

## 5.1.5. Memory Classes

CLASS E

- **First 128KW of physical memory of any speed.**

CLASS H

- **600ns memory with physical address greater than 128KW.**

CLASS S

- **Memory with access time greater than 600ns with physical address greater than 128KW (shared memory).**

CPU 1 in SEL A
CPU 2 as IPU in SEL A
CPU 3 in SEL B
CPU 4 as IPU in SEL B

## 5.1.6. Units of Memory

| | | |
|---|---|---|
| Byte | = 8 bits | |
| Halfword | = 2 bytes | |
| Word | = 2 halfwords | |
| Doubleword | = 2 words | |
| File | = 8 words | X32 |
| Page | = 512 words | |
| Protection Granule | = 1 page | |
| Map block | = 8 KW | |

## 5.1.7. IOM

- **Individual SelBUS plug-in modules that follow class 0, 1, 2, E, or F SelBUS protocols.**

- **Provide input/output capabilities to the system.**

- **Operate at one of 23 jumpered SelBUS priorities.**

- **Examples - Magnetic Tape Controller (MTC), Terminal/Line printer, card reader (TLC), Asynchronous Data Set (ADS), High-speed Data Interface (HSD).**

## 5.1.8. IOP / MP bus

- **Multiplexed channel providing interface to Multi-purpose Bus**

- **16 software controllable interrupt levels**

- **32-bit programmable interval timer**

## 5.1.9. RTOM (Real-time Option Module)

- **Provides 16 basic interrupt levels on 32/7X computer system per RTOM**
- **4 external interrupt levels**
- **32-bit programmable interval timer**
- **Provides real-time clock for software accounting and time-of-day**

## 5.1.10. Traps And Interrupts

Traps and interrupts report asynchronous or synchronous events to the software. Traps are normally error conditions that are generated internally (to the CPU) and interrupts are requests that are generated externally (RTOM, IOP, IOM).

Traps in order of priority:

- **Power Fail**
- **Memory Parity**
- **Nonpresent Memory**
- **Undefined Instruction**
- **Privileged Violation**
- **System Override**
- **Supervisor Call (software generated)**
- **Machine Check**
- **System Check**
- **MAP Fault**
- **Block Mode Timeout (Watchdog)**
- **Arithmetic Exception**
- **Start / Stop IPU processing**

Interrupts consist of the following:

- **Any external event scheduled through the RTOM.**
- **Input/Output (I/O) termination interrupts.**
- **Software request interrupt control instructions.**

# 6. IN-CLASS QUIZ

1. Which CPU general purpose register will contain the current PSW during a branch & Link operation?

    a. R0

    b. R1

    c. R4

    d. R6/R7

2. What general purpose register is used to contain a mask for mask operations?

    a. R1

    b. R2

    c. R3

    d. R4

3. Which bits of the PSD contain the logical address of the next instruction to be executed?

    a. Bits 0-31

    b. Bits 0-4

    c. Bits 12-31

    d. Bits 32-63

4. How does the CPU determine whether a task can legally execute a privileged instruction?

a. By insuring the CPU itself is operating in the PSD mode.

b. All tasks are authorized to execute all Macro level instructions.

c. By checking bit 6 of the current PSD/PSW.

d. By checking bit 0 of the current PSD/PSW.

5. What is the primary difference between PSD and PSW modes of operation?

a. In the PSD mode (used for MPX) the CPU is not capable of performing memory mapping operations.

b. In the PSW mode (used for RTM) the CPU is not capable of performing memory mapping operations.

c. In the PSW mode (used for RTM) interrupts are not recognized by the CPU.

d. In the PSD mode (used for MPX) traps and interrupts are implemented, whereas RTM only uses traps.

# 7. LABORATORY EXERCISE

## 7.1. 32/7X ARCHITECTURE

The student will identify the following components, subsystems, and plug-in modules in the SERIES 32/7X computer system.

- **SelBUS**
- **System Control Panel**
- **Three board Central Processor Unit (CPU)**
- **Three Board Internal Processor Unit (IPU)**
- **Real-time Option Module (RTOM)**
- **Input/Output Processor (IOP)**
- **Magnetic Tape Controller (MTC)**
- **Disc Processor**
- **Floating Point Accelerator (FPA)**
- **8-line Asynch**
- **Line printer/floppy disc controller**
- **Memory Bus controller**
- **MOS memory modules**
- **Shared memory chassis**
- **High-speed data Interface (HSD)**

In order to accomplish this laboratory exercise you must be familiar with Gould SEL part/model numbers or be able to trace cables and connectors from peripherals and subsystems to their respective processors/controllers.

# 8. HOMEWORK QUIZ

1. List four of the major plug-in modules available on the SERIES 32/7X computer system and describe the functional characteristics of each.

a.

b.

c.

d.

2. If bit 4 is set in the PSD after an arithmetic operation, what does this indicate?

a. The result of the arithmetic operation was zero.

b. The result of the arithmetic operation generated a privilege violation trap.

c. The current program counter will point to the next halfword instruction to be executed.

d. Bit 4 is not used in PSD mode.

3. If bit 5 in the PSD is currently set, what does this indicate?

a. 24 bits will be used from the index register during indexed operations in order to address logical extended memory.

b. Arithmetic exception handling by MPX-32 is disabled.

c. The operating system executed a "CEA" instruction.

d. A device or interrupt handler executed a "SEA" instruction during I/O processing.

4. If the SelBUS transfers 32-data bits per bus cycle and the bus cycle time is 150 ns., what is the bandwidth (transfer rate) of the SelBUS?

a. 32 Megabytes per second.

b. 26.67 Megabits per second.

c. 150 Megaflops per second.

d. 26.67 Megabytes per second.

5. Since the CPU and IPU are physically identical processors, how does the CPU know its the CPU and the IPU know its the IPU?

a. First come - First serve!

b. A jumper is inserted into the processor designated as the IPU.

c. Since both processors are "intelligent", they decide whom is the CPU and whom is IPU.

d. The SERIES 32/7X does not support an IPU.

# 9. INSTRUCTORS NOTES

## 9.1. Tips to Remember

### 9.1.0.1. briefs

For more information on the SERIES 32/7X computer system refer to the following documentation:

- **SERIES 32/77 Technical Manual**
- **SERIES 32/77 Reference Manual**
- **CONCEPT 32 Interface guide**

### 9.1.0.2. Gotcha's

# MPX-32 OVERVIEW

# 1. OVERVIEW

The MPX-32 operating system is an interrupt driven, multi-tasking executive that operates in a hardware mapped environment. MPX-32 evolved from the RTM (Real-time Monitor) operating system but unlike RTM, MPX-32 supports 16MB of physical memory.

After completing this section you should be able to answer the following questions:

- *Is the 16MB memory capacity restriction of the SERIES 32 computer system a function of MPX-32 or the SERIES 32 hardware?*
- *What is the difference between context switching and swapping?*
- *Are files under MPX-32, blocked or unblocked by default?*
- *What is meant by a blocked file?*
- *What is the purpose of memory mapping? Do we really need it?*

# 2. OBJECTIVES

In order for the student to interact correctly with MPX-32 and its' associated utilities and service calls, the student must be familiar with the programming environment MPX-32 provides.

*Please insure the following objectives have been achieved prior to beginning the MPX-32 Overview laboratory exercises.*

¤ **1. The student will identify critical data structures associated with MPX-32 and an a designated task.**

¤ **2. The student will correctly answer several questions on MPX-32 logical and physical memory organization.**

# 3. ABBREVIATIONS

| | |
|---|---|
| ° **LFC** | Logical File Code |
| ° **MPX** | Mapped Program Executive |
| ° **TSA** | Task Service Area |

# 4. DEFINITIONS

° **MPX**

Mapped Program Executive. A disc-oriented, multi-tasking executive which effectively coordinates the concurrent execution of multiple real-time, batch, and interactive tasks in a hardware mapped environment.

° **TASK**

A logically organized collection of routines in loadable format which can be scheduled for CPU execution time.

° **Physical Memory**

The Physical memory configured in the system. A task will occupy a predetermined amount of physical memory during its normal life cycle.

° **Logical Memory**

A tasks address space as represented by the CPU map registers. This space includes MPX-32 and the tasks TSA, Code and data segments, and extended address space. Logical memory allows the CPU to have instant access to the physical memory locations occupied by the currently executing task.

° **Buffered I/O**

Most CLASS 'E' I/O devices can only access the first 128KW of physical memory. If your task resides in memory above 128KW (and normally it will) and wishes to perform I/O to a CLASS 'E' device, that data must be buffered through the operating system which always resides in the first 128KW of physical memory.

° **Context Switching**

The process whereby a tasks either gains or relinquishes CPU control. The process requires the loading of the CPU map registers (if mapping is required), the loading of the general purpose registers, and the loading of the PSD.

° **Swapping**

The process whereby a task either gains or relinquishes physical memory. When a task is "SWAPPED OUT", the task has been physically removed from memory and temporarily stored on disc until physical memory becomes available. When a task is "SWAPPED IN", the task is reloaded from the temporary disc space back to physical memory.

° **Task Service Area (TSA)**

The Task Service Area (TSA) is a data structure unique to each task. The TSA contains information required by the CPU and by MPX-32 to identify your logical address space, general purpose register stack, PSD, and other critical task information.

# 5. WHAT YOU NEED TO KNOW

## 5.1. In a Nutshell

MPX-32 is a collection of routines logically organized in bootable format (also known as an operating system). The purpose of MPX-32 is to provide certain functions and services that most users require on a continuous basis:

- **Task activation /termination services**
- **Logical and Physical I/O services**
- **Conversion Utilities**
- **Time-of-day service**

## 5.1.1. Task Generation

Task generation is a fairly simple process as follows:

- **Generate appropriate source code (FORTRAN, Assembler, etc.)**
- **Compile or assemble the source code into object code.**
- **CATALOG the object code into a relocatable and loadable format called a loadmodule.**
- **It is now ready for loading and execution.**

## 5.1.2. Task Activation Methods

A task can be activated in several ways as follows:

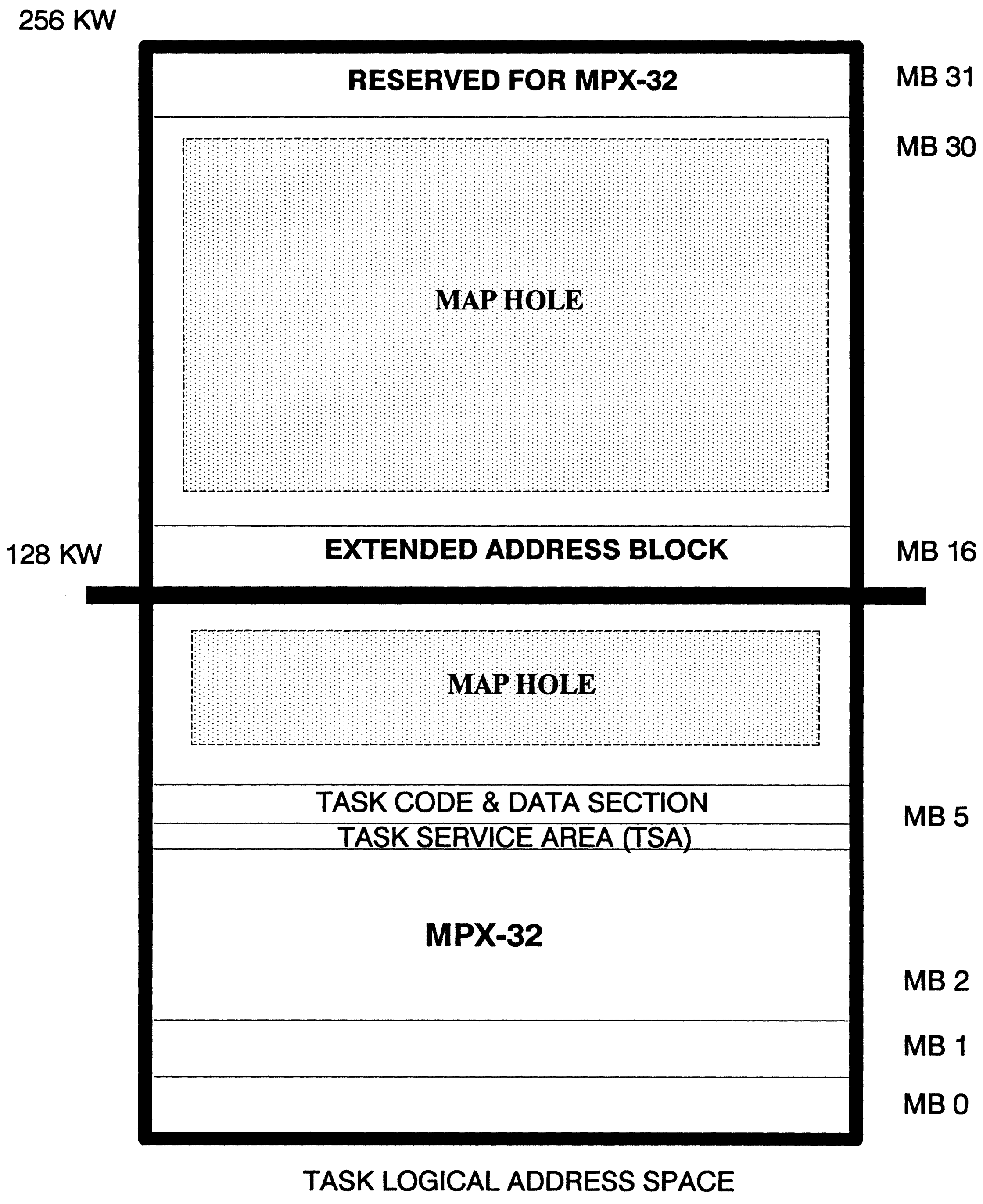
| | |
|---|---|
| ◦ **Job Control Language** | - **ACTIVATE / EXECUTE** |
| ◦ **Terminal Services Manager** | - **RUN / EXECUTE** |
| ◦ **System Service Calls** | - **M.ACTV / M.PTSK** |
| ◦ **Timers** | |
| ◦ **OPCOM** | - **ACTIVATE / ESTABLISH** |

## 5.1.3. Task Priorities

Tasks execute at a realtime priority or time distributed priority.

| | |
|---|---|
| ◦ **Realtime** | **1 - 54** |
| ◦ **Time Distributed** | **55 - 63** |

## 5.1.4. Physical vs. Logical Memory



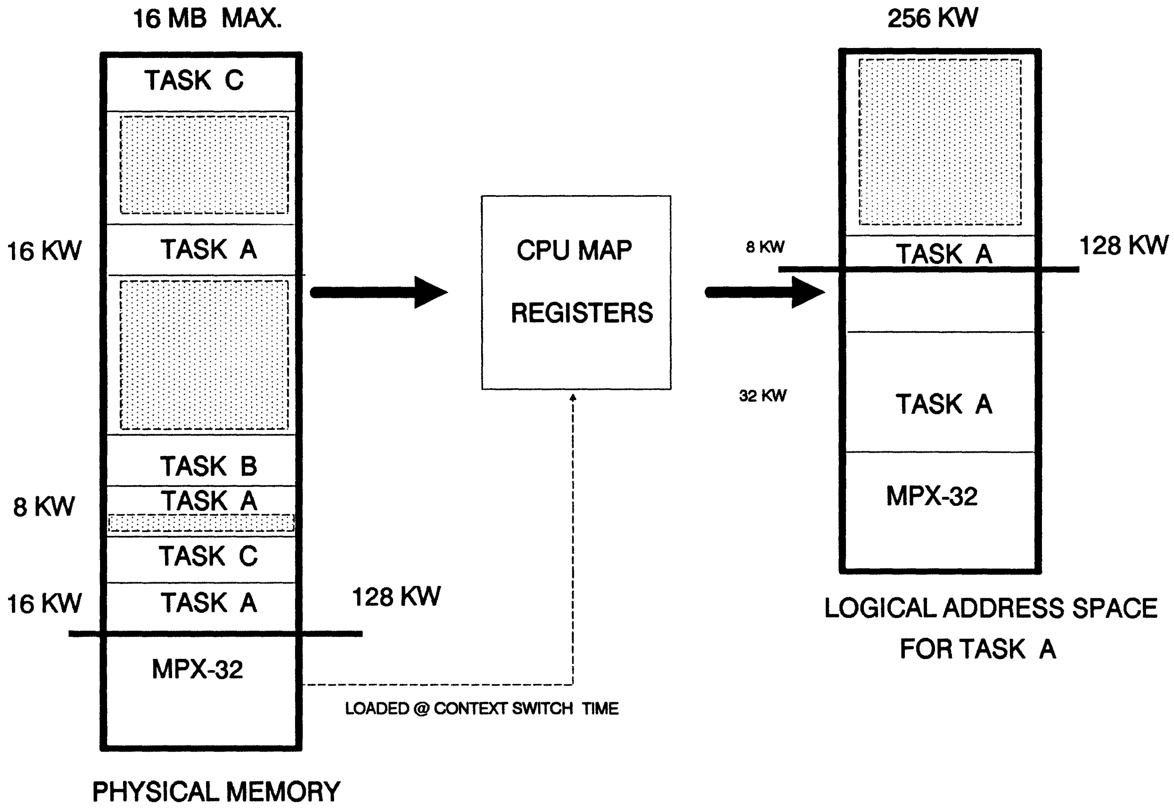
MPX-0001

MB = MAP BLOCK #

## 5.1.5. MPX Memory Mapping

Memory mapping in MPX-32 is a process which requires cooperation by both the hardware and the software. The hardware always follows the protocol established by the CPU firmware. The hardware expects certain data structures to be present in predefined locations. The software therefore must insure that the data structures (map register information, new PSD, general purpose registers) are created correctly and reside in the proper memory locations.

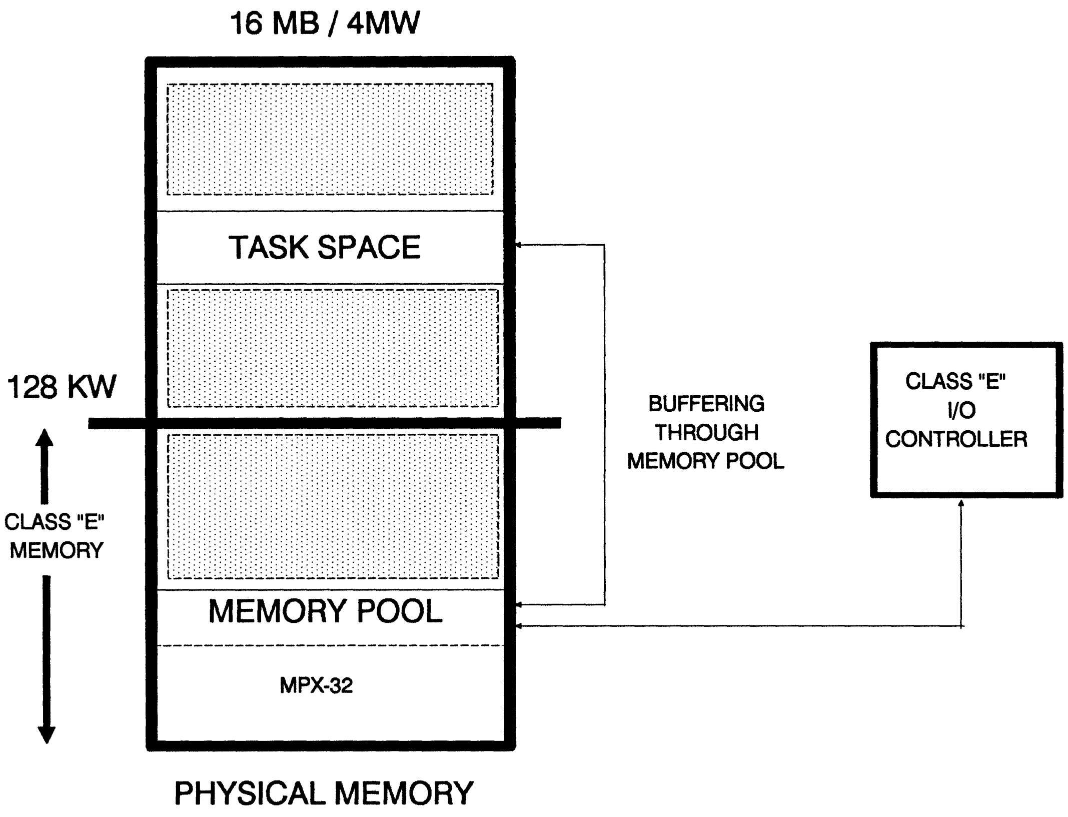
One the CPU has been initialized with the proper context information, the task may begin/continue execution.



# MPX MEMORY MAPPING

MPX-0002

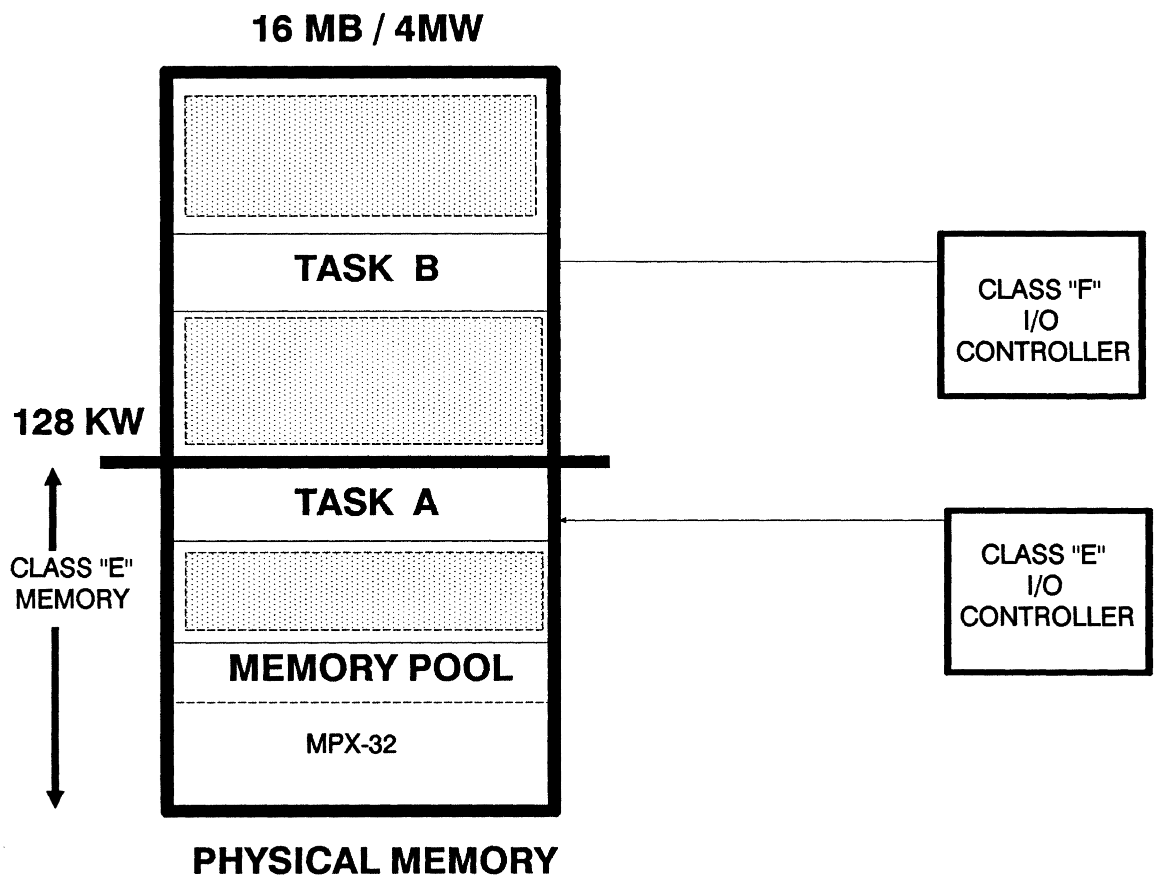
## 5.1.6. MPX Buffered I/O



MPX-32 BUFFERED I/O

MPX-0003

## 5.1.7. MPX Unbuffered I/O



# MPX-32 UNBUFFERED I/O

MPX-0004

# 6. IN-CLASS QUIZ

1. You must always buffer your I/O operation for Class 'E' I/O controllers?

   a. True

   b. False

2. Class 'F' I/O controllers can access 16MB of physical memory and therefore don't require the data to be buffered in Class 'E' memory!

   a. True

   b. False

3. State the difference (s) between logical and physical memory.

   a. Physical Memory

   b. Logical Memory

4. When would a logical and physical address be the same?

   a. When the CPU is running in the mapped mode and the 'SEA' instruction has been executed.

   b. When the logical program counter points into Class 'E' memory.

   c. When the CPU map registers have been initialized with MPX in Class 'E' memory and your task in Class 'H' memory.

d. While executing within MPX-32, provided MPX-32 physically resides in Class 'E' memory.

5. Why is the last map block in your logical address space reserved for MPX-32?

a. Buffered I/O operations.

b. For TSA remap operations.

c. Intertask communications.

d. Nobody really knows.

# 7. LABORATORY EXERCISE

## 7.1. Pre-Laboratory Exercise

# 8. HOMEWORK QUIZ

1. State the purpose and usage of swapping?

2. Your tasks TSA logically resides right above MPX-32. Where does it reside in physical memory?

   a. Always in Class ’E’ memory since it must be in the first 128Kw of logical memory.

   b. Anywhere except in MPX-32’s address space.

   c. Logically above 128KW and physically below 128Kw.

   d. Always in logical extended memory.

3.

   a.

   b.

   c.

   d.

4.

   a.

   b.

c.

d.

5.

a.

b.

c.

d.

# 9. INSTRUCTORS NOTES

## 9.1. Tips to Remember

For more information on MPX-32 refer to the following documentation:

- **MPX-32 Reference Manual, Vol I, Chapters 1,2,3**

### 9.1.0.1. Briefs

It is important to remember the difference between logical and physical memory. Physical memory is what is actually configured in the system (upto 16MB). Logical memory is what is reflected in the CPU map registers (upto 256KW).

Without Memory Mapping, our physical address space would be limited to 128KW of physical memory.

Your task may reside anywhere in physical memory if the CPU is running in the mapped mode; this includes your TSA, Code, Data and any memory partitions (common blocks) your task may be using.

MPX-32 is mapped into everybodies task space so that you may execute system service calls.

### 9.1.0.2. Gotcha's

If the CPU halts, the front panel display always shows the program counter and the instruction causing the halt. The program counter is a logical address not a physical address (unless it points into MPX-32. In order to find physical address you need to manually convert the logical address to a physical address.

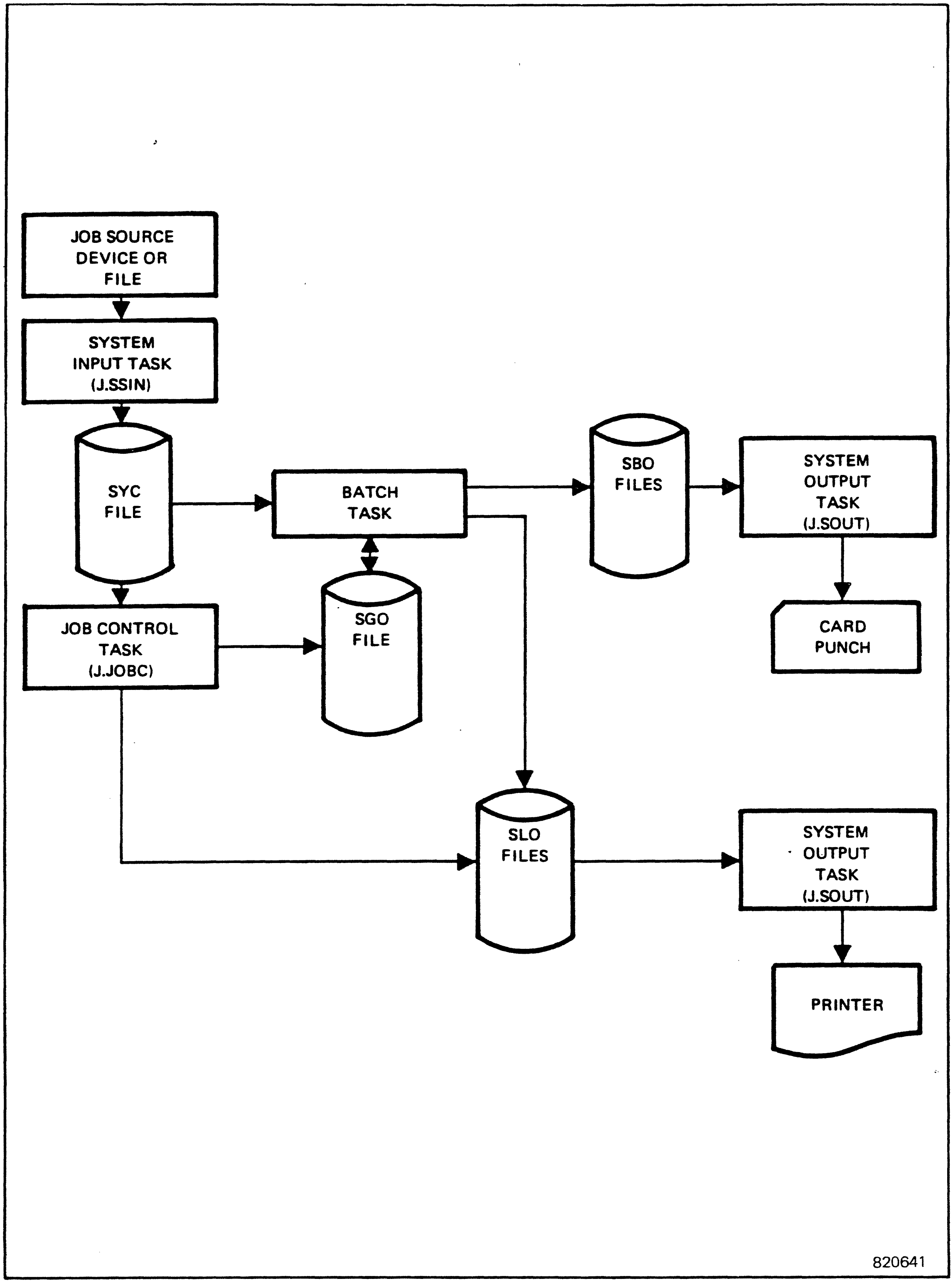

JOB SOURCE DEVICE OR FILE
SYSTEM INPUT TASK (J.SSIN)
SYC FILE
BATCH TASK
SBO FILES
SYSTEM OUTPUT TASK (J.SOUT)
SGO FILE
JOB CONTROL TASK (J.JOBC)
CARD PUNCH
SLO FILES
SYSTEM OUTPUT TASK (J.SOUT)
PRINTER
820641

Figure 6-1. Data Flow for a Job

# JOB CONTROL LANGUAGE (JCL)

# 1. OVERVIEW

- Batch (background) processing is a method of performing task execution while interactive processing continues.

- There are several methods whereby the programmer can invoke the background processor (J.JOBC). Theses methods will be described in detail.

- Batch processing and interactive job processing are very similar though a few important differences exist. All of the TSM commands are valid during batch processing, but not all batch mode commands are valid for interactive processing.

# 2. OBJECTIVES

In order for the student to build and maintain batch files, a knowledge of JCL commands, batch processing, and system spooled files is required. *The following objectives must be achieved prior to beginning the JCL laboratory exercises.*

¤ **1. The student will correctly answer a set of questions on batch processing.**

¤ **2. The student will correctly write a series of JCL commands to perform a specified function.**

¤ **3. The student will state the four (4) special system files, how they are created and allocated and the purpose of each.**

# 3. DEFINITIONS

- **PERMANENT DISC FILE**

  A collection of disc sectors logically linked to an entry in the System Master Directory (SMD). Permanent disc files may be identified by username and filename.

- **TEMPORARY DISC FILE**

  A collection of disc sectors not linked to an entry in the System Master Directory (SMD). Temporary files are only identified by logical file code and are deleted when the associated task terminates or closes the file.

- **J.SSIN**

  System input spooling task. This task is a pre-processor for J.JOBC; it allocates and builds each SYC file for the batch stream.

- **J.JOBC**

  Job Control Processor. This task is responsible for all batch processing under MPX-32 1.X.

- **JOB**

  A logical collection of tasks combined with job control commands that begin with a $JOB card and end with a $$ card.

- **M.SID**

System Input Directory. A system file used by J.SSIN and J.JOBC to synchronize batch processing. This file contains entries for each spooled SYC file.

° **M.SOD**

System Output Directory. A system file used to unspool SLO temporary files to designated destinations. The file is similar to M.SID in that it contains pointers to each SLO file. This file is used by the System Output Task (J.SOUT) for unspooling after each job completes.

° **SYSTEM CONTROL FILE (SYC) FILE**

A temporary disc file that provides intermediate storage for Job Control statements, object code, and data for a batch job. A separate job file is dynamically created on the SYC for each task initiated in a user's job file, and when the last task in the job completes execution, the job's SYC is deleted.

° **SYSTEM LIST OUTPUT (SLO) FILE**

A temporary disc file used for listed output. SLO files generated by real-time task are output to destination peripheral devices when the files are deallocated. SLO files generated by a batch job are output to destination peripheral devices upon job completion.

- **SYSTEM GENERAL OBJECT (SGO) FILE**

SGO files are used for the accumulation of object code within batch jobs and command files. A separate SGO file is allocated for each job and exists for the duration of the job. The SGO file is a temporary file which is deallocated upon job completion.

- **SYSTEM BINARY OUTPUT (SBO) FILE**

An SBO file is a temporary file used for punched output. SBO files generated by real-time tasks are output to destination peripheral devices when the files are deallocated. SGO files generated by a batch job are output to destination devices upon job completion. The SBO and SGO files are the same format and both are used to accumulate object code to be used by CATALOG, LIBED or user developed tasks.

# 4. WHAT YOU NEED TO KNOW

## 4.1. In a Nutshell

## 4.1.1. Job Control Flow

Each job is spooled to a separate System Control (SYC) disc file by J.SSIN prior to J.JOBC's invocation. J.SSIN performs input spooling for all batch processing, and J.JOBC is the job control command processor for MPX-32 1.X.

Each Job begins with a $JOB card, is followed by additional job control statements and is terminated by a $$ card. During input spooling to the SYC file, J.SSIN verifies that the first input card contains a $JOB card beginning in column 1. If J.SSIN determines that the input file is a valid job, input spooling continues until the $$ card is processed. Each job control statement is copied from the batch input file to the SYC file except the $$ card and other batch commands processed by J.SSIN ($OBJECT, $SELECTXX, $$$). Once the input spooling is complete the SYC file is queued, and entry is made in M.SID, and J.JOBC is sent a run request for final job processing.

When J.JOBC is invoked the next job definition is retrieved from M.SID and entered into the batch stream for processing. J.JOBC obtains the corresponding SYC file for the job and begins processing the $JOB control statement. Job processing continues until the $EOJ card is detected in the SYC file.

Upon detection of the $EOJ card in the SYC file, J.JOBC terminates job processing and begins job post-processing. J.JOBC can process several jobs simultaneously.

### 4.1.2. Job Post-processing

- **Close and deallocate the current SYC, SGO, and SBO temporary files.**
- **Unspool the SLO temporary file to the output destination identified on the $JOB card. The destination may be a permanent disc file or device (i.e. Line printer, tape, card punch). Each SLO file is queued, and entry made in M.SOD, and a run request is sent to J.SOUT (System Output Task) for output processing.**

Should multiple job definitions be queued in M.SID, J.JOBC continues loading jobs in the batch stream until the job queue is full. The size of the job queue is determined at SYSGEN time.

The job control process is shown in Figure 1.

## 4.1.3. JCL Examples

The following job control example will accomplish the following:

- **The job name is EXAMPLE**
- **The job ownername is BELDEN**
- **The listed output destination is a permanent disc file named LO.TEST (This is where the echoed job control statements plus the FORTRAN outputs will be unspooled when the job completes).**
- **The job will execute a task named FORTRAN**

Send listed output file

```
$JOB EXAMPLE BELDEN SLOF = LO.TEST
$NOTE THIS IS AN EXAMPLE OF JCL
$OPTION 2 3 4 5 19
$FORTRAN
        PROGRAM MAIN
        INTEGER*4  I
        DO I = 1,10
           ICOUNT = ICOUNT + I
        END DO
        STOP
        END
$EOJ
$$
```

- **J.JOBC echoes (copies) each job control statement to the SLO file during job processing. The SYC file is accessed and copied sequentially in a downward direction only.**
- **If conditional job control statements have been supplied, those statements not processed will be echoed to the SLO file and will be preceded by an open parenthesis "(".**

# 5. COMMAND SUMMARY

- **$ACTIVATE** *
- **$ALLOCATE**
- **$ASSIGN1**
- **$ASSIGN2**
- **$ASSIGN3**
- **$ASSIGN4**
- **$DEBUG**
- **$DEFNAME**
- **$EOJ**
- **$EXECUTE**
- **$GOTO**
- **$IFF**
- **$IFT**

- $JOB
- $NOTE
- $OBJECT *
- $OPTION
- $RESETF
- $SCAN
- $SELECTD *
- $SELECTF *
- $SELECTLD *
- $SELECTLF *
- $SELECTS *
- $SETF
- $USERNAME
- $$
- $$$ *

## 5.0.1. Don't Forget

Remember which commands are valid only in the batch mode. Any command which provides pre-input spooling will not be available in the interactive environment because J.SSIN is only invoked during batch processing.

The $ACTIVATE command is only recognized by J.JOBC, therefore this command is not valid during interactive job processing.

# 6. IN-CLASS QUIZ

1. Which of the following JCL Command Lines would not be valid in the *Interactive environment*?

a. $SELECT MYFILE<br>
b. $USERNAME BELDEN<br>
c. $OBJECT<br>
d. $SCAN 72

2. Which of the following JCL commands would be used to assign the Logical file code (LFC) SYC to the System Control file in the blocked mode, in either the batch or interactive environments?

a. $ASSIGN3 SYC = TEMP,,U<br>
b. A2 SYC = SYC,,U<br>
c. $A2 SYC = SYC<br>
d. $ASSIGN4 SYC = SYC

3. Which JCL command signals the end of batch input spooling when continuous batch mode has been specified with the OPCOM MODE SCBT command?

~~a.~~ $$<br>
b. $EOJ<br>
[✓ c.] $$$<br>
d. $SCAN

4. How is the number of batch streams determined for a given system?

a. With the M.CDJS system service call.
[✓ b.] It is determined at SYSGEN time.
c. By using a hexidecimal calculator.
d. By the OPCOM BATCH command.

5.~~4.~~ Write a JCL segment that will invoke the FORTRAN compiler and perform the following:

- **Source input for the compiler will be from the file "MYINPUT"**
- **Compiler options 2 3 4 and 5 should be set**
- **Generated Object output (SGO) should be to the file "MYOBJECT"**
- **Direct the jobs SLO output directly to the Line printer (bypass output spooling).**
- **Direct the compilers output to the file "MYOUT"**
- **The job will run from username BELDEN, password SECRET**
- **Write a message to the Operators Console if the compiler aborts. The message should state the compiler aborted.**

5. List the four system spooled files and described in excruciating detail the method of creation, method of identification and how each of the files is allocated and deallocated, and the purpose of each file.

a.
b.
c.
d.

# 7. LABORATORY EXERCISES

## 7.1. Pre-laboratory Exercise

The following JCL laboratory assignments will be completed in the batch mode. When you complete the assignments please show your assignment sheets to your instructor. You may then complete the assignments in the interactive mode if desired.

PLEASE ENTER YOUR ASSIGNED LOGON NAME HERE - OWNERNAME: ____________________

List below where JCL commands are found in the standard documentation.

________________________________

### 7.1.1. Exercise

Your instructor will describe the laboratory exercise to be performed in this section.

# 8. HOMEWORK QUIZ

1. What task is responsible for interactive job processing?

a. J.JOBC<br>
b. J.SSIN<br>
c. MPX-32<br>
d. J.TSM

2. Why are the $SELECTXX commands not valid during interactive job processing?

a. These commands are only recognized by the batch processor J.JOBC.

[✓ b] These commands are only recognized by the input spool task J.SSIN and J.SSIN is not available for interactive job processing.

c. Theses commands are valid during interactive job processing.

d. These commands are only available for selecting system spooled files in the batch environment.

3. Which of the following ASSIGN commands would be used to assign an LFC to the Magnetic tape?

logical file code

a. ASSIGN1<br>
b. ASSIGN2<br>
[✓ c.] ASSIGN3<br>
d. ASSIGN4

4. When making LFC assignments how do you indicate that a file is to be accessed in the blocked mode?

a. Files are assigned blocked by default.

b. The is no provision for Unblocked files during batch processing.

c. Only devices may be accessed unblocked.

d. By using the ASSIGN4 command.

# 9. INSTRUCTOR NOTES

## 9.1. Tips to Remember

### 9.1.0.1. Briefs

Interactive job processing is covered in greater detail in the Terminal Services Manager (TSM) chapter.

For additional information on Job Control Language and batch processing refer to the following Documentation:

- **MPX-32 1.X Reference Manual- Volume I**

### 9.1.0.2. Gotcha's

If the ownername on the job card does not exist in the M.KEY file, J.SSIN will terminate input processing.

The jobname and ownername are optional on the $JOB card during interactive processing. You may have them present but they are not used or verified.

If you wish to have access to the SGO file during interactive processing you must have a $JOB card, otherwise no SGO file is allocated during interactive processing.

You always have access to the SYC, SLO, SGO and SBO files during batch processing. The SYC file is allocated in read-only mode and each record in the file is accessed sequentially. You may read from or write to the SLO, SGO and SBO files during batch processing. In order to do this your task must allocate the specific file using the $ASSIGN2 command.

During interactive job processing the user terminal or command file is used as the SYC file. SGO file access is provided with the $JOB card. No access to SBO or SLO is provided during interactive job processing.

If you do not require an SGO file during interactive job processing, do not execute the JOB command.

The text editor RUN and BATCH commands are identical and either command may be used to invoke the batch processing pre-input spooler (J.SSIN).

# MPX-32 MEMORY MANAGEMENT

# 1. OVERVIEW

The simulation software utilizes several complex methods of sharing data and code between tasks executing in different processors and between task executing on different computers linked through shared memory.

In this section we will examine several of the concepts involved in memory management and how they are implemented in the simulation.

# 2. OBJECTIVES

In order for the student to understand how the simulation software executive and application routines share data between 4 processors, an understanding of memory management concepts is essential.

*Please insure the following objectives have been achieved prior to beginning the Memory Management laboratory exercises.*

¤ **1. The Student will correctly answer several questions on memory partitions including DATAPOOL and GLOBAL COMMON.**

# 3. ABBREVIATIONS

° **CSECT**

Write protected memory containing code and pure data. The task must be created using the ASSEMBLER CSECT directive.

° **DSECT**

Read/Write memory containing impure data. If the task is not sectioned the DSECT contains all the tasks code and data.

# 4. DEFINITIONS

° **STATIC PARTITION**

Partitions created at SYSGEN time. They are non-swappable partitions.

° **DYNAMIC PARTITION**

Partitions created by File Manager. They are swappable when allocation count is 0 and the use count = 0.

° **DATAPOOL**

A special type of memory partition where each symbol is not order dependent. Each symbol in the partition is defined through a DATAPOOL dictionary.

# 5. WHAT YOU NEED TO KNOW

## 5.1. In a Nut Shell

Pay close attention to the following discussion on DATAPOOL. The information on memory mapping presented earlier will be important in your understanding how DATAPOOL operates. Though GLOBAL commons are not used in the simulation they are used in the disk-to-disk backup system on SEL C/D.

## 5.2. GLOBAL and DATAPOOL Memory

Intertask shared memory is provided under MPX-32 through GLOBAL and DATAPOOL memory partitions. There are up to one hundred Global regions (GLOBAL00- GLOBAL99) plus a DATAPOOL partition available to the user.

- **Global and DATAPOOL partitions can be defined either via SYSGEN or through the CREATEM command in the File Manager utility.**

- **Partitions created at SYSGEN are considered permanently allocated; they are assigned both physical and logical memory attributes with apply to any task that references the partitions. This type of allocation is called static allocation. The static Global and DATAPOOL partitions are defined in integral numbers of protection granules.**

- Both GLOBAL and DATAPOOL partitions are located in an integral number of physical map blocks starting on a map block boundary and ending logically at the top of the users execution space (128KW).

- Write Protection is available to prevent the user from storing into a common area to which he does not have write access.

## 5.2.1. Static vs. Dynamic common

- Statically allocated common is fixed in physical memory even when no task is sharing it through is map. Dynamically allocated common is deallocated when it allocation count equals zero.

- Statically allocated common is allocated in increments of 512 words, while dynamically allocated common is allocated in map block increments (8KW on 32/7X computer and 2KW on a CONCEPT/32).

- Statically allocated common is invoked on a system-wide basis. Dynamically allocated common is based on a subsystem concept, where a single task issues an M.SHARE request system service.

- Dynamically allocated common can be excluded from a task via the M.EXCL system service. The users can elect to subsequently include another dynamically allocated common area via M.INCL. Statically allocated partitions are not supported.

- All logical references to common, whether statically or dynamically allocated, are resolved by the Cataloger (CATALOG). This is possible because the logical address of a system common partition is fixed when the partition is defined.

- Load modules from one MPX-32 configuration are compatible with another configuration, even if GLOBAL or DATAPOOL are allocated different physical addresses.

**The only compatibility requirement is that both systems employ the same logical conventions.**

## 5.2.2. Static vs. Dynamic Memory Partitions

| Characteristics | Static | Dynamic |
|---|---|---|
| Logical Address | Fixed at SYSGEN | Fixed by File Manager |
| Physical Address | Fixed at SYSGEN | Variable |
| Allocation Unit | 512 words | 8K words (32/7x)<br>2K words (CONCEPT/32) |
| Time of Allocation | SYSGEN | Run time via M.SHARE |
| Time of Deallocation | Never | When Allocation count=0 |
| Inclusion | Automatic via Activation | Run time via M.INCL |
| Exclusion | Automatic via Exit or M.EXCL | Automatic via Exit of M.EXCL |
| Owner Names or task numbers | None | Established by M.SHARE caller |
| Swapping | Never Swapped | Swappable when user count = 0 |

## 5.2.3. Memory Partition Applications

| Characteristics | Global | DATAPOOL | Ext. Common | CSECTS |
|---|---|---|---|---|
| Cataloger resolves references | YES | YES | NO | YES |
| Compiler resolves references thru extended bases | NO | NO | YES | N/A |
| Must be Logically below 128KW | YES | YES | NO | YES |
| Variables are order dependent | YES | NO | YES | N/A |
| Static | YES | YES | YES | NO |
| Dynamic | YES | YES | YES | YES |

# 6. SHARED MEMORY EXAMPLE

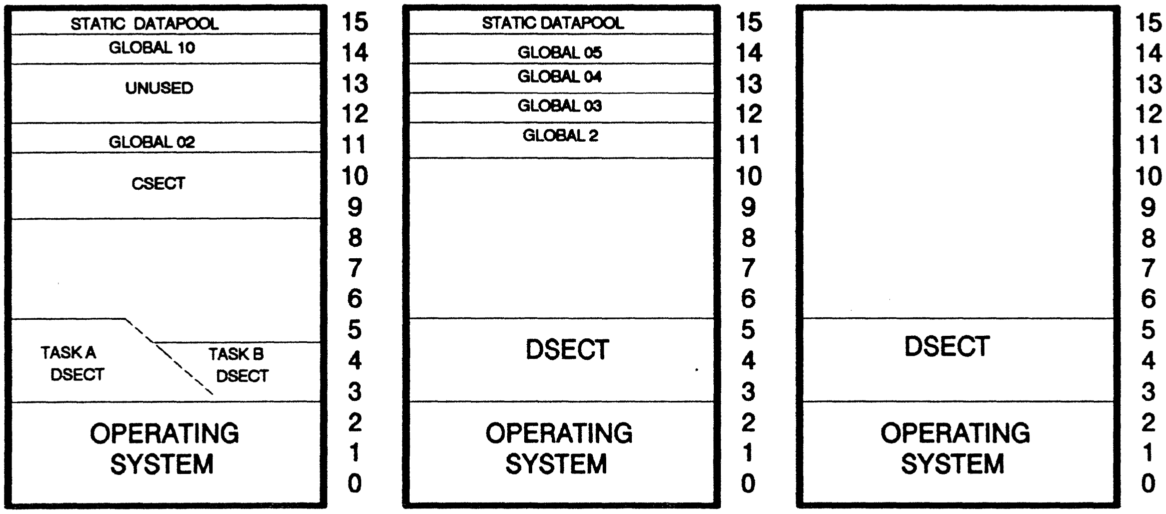
- Tasks A, B, and C all reference a static DATAPOOL partition.

- Task A has used an M.SHARE service for dynamic GLOBAL10 and Task B has used M.INCL for GLOBAL10. Thus GL10 is mapped at the same location in each logical address space.

- Tasks A and B use M.INCL for GL02, and Task C has used M.SHARE for GL02, to use for intertask communication. Thus GL02 is mapped into the same location in each logical address space.

- Task D shares no memory or code with other task. Map blocks 7 and up are available to the task.

- Tasks A and B are shared. They have CSECT mapped at the same location in each logical address space.



TASKS A + B
(SHARE CODE)

TASK C
(DOES NOT SHARE CODE)

TASK D
(DOES NOT SHARE CODE
OR COMMON)

# ALLOCATION OF MEMORY PARTITIONS AND CODE

MPX-0008

# 7. IN-CLASS QUIZ

1. Please state 4 major differences between Static and Dynamics Memory Partitions.

a.

b.

c.

d.

2. Please state 4 major differences between DATAPOOL and GLOBAL common.

a.

b.

c.

d.

3. When are references to DATAPOOL symbols resolved in a program?

a. During Compilation/Assembly

b. During Task loading and execution

c. During CATALOG

d. During LIBED

# 8. LABORATORY EXERCISE

## 8.1. Pre-Laboratory Exercise

# 9. INSTRUCTORS NOTES

## 9.1. Briefs

## 9.2. Gotcha's