Student Text
Preface

This publication is a student text on data flow in the IBM 3704 and 3705 Communications Controllers network control program (NCP), release 5 data flow. This publication does not include information on Advanced Communications Function (ACF) Network Control Program (NCP), which supports multiple, concurrent channel adapters in NCP mode or in multidomain networks.

Prerequisite knowledge of the IBM 3704 and 3705 Communications Controllers is required to understand this material. The prerequisite information may be obtained in the following:

*IBM 3704 and 3705 Communications Controllers Hardware* (SR20-4544)

*IBM 3704 and 3705 Communications Controllers NCP Programming* (SR20-4568)

*Advanced Function NCP and Related Host Traces* (SR20-4510)

*IBM 3704 and 3705 Control Program Generation and Utilities Guide and Reference Manual* (GC30-3008)

A quiz appears at the end of each major section, with the answers given in Appendix B. You will need the following handbook to answer the questions.

*IBM 3704 and 3705 Program Reference Handbook* (GY30-3012)

If you require additional information, please refer to:

*IBM 3704 and 3705 Communications Controllers, Network Control Program/VS Program Logic Manual* (SY30-3013).

*IBM 3704 and 3705 Communications Controllers Principles of Operation* (GC30-3004)
List of Illustrations

1.1. Program Levels
2.1. NCP Components
3.1. Task States
3.2. Dispatcher Entry Points
3.3. Dispatcher Execution Sequence
4.1. Data Transfer from NCP to Host Buffer Units
5.1. Subarea Index Table
5.2. Subarea Vector Table
5.3. Resource Vector Table
6.1. Session Hierarchy
6.2. SSCP to NCP Physical Services Command Sequence
6.3. Activate Physical and Activate Logical Commands
6.4. Bind and Start Data Traffic Commands
7.1. Segmentation Example
7.2. Boundary Network Node Path Flow
7.3. Boundary Network Node Inbound Path Flow
7.4. SNA and SDLC Switched Command Sequence
7.5. SNA and SDLC Commands to Terminate Switched Connections
7.6. CUB and LUB PIU Processing
8.1. Command Sequence to Activate and Remote
8.2. Remote Disk Format
8.3. Command Sequence to Dump a Remote
8.4. Command Sequence for Alternate Link
8.5. Command Sequence for Remote Power Off
9.1. Activate Link Command Flow
10.1. BSC/SS Processor Components
10.2. BSC/SS Processor Flow
10.3. BSC/SS Block-handler Control Block Relationships
10.4. MTA List Format
10.5. LCST Entry Relationships
## Contents

Preface .................................................................................. i

Hardware and Programming Structure .................................. 1.1
  Review of Hardware Facilities ............................................. 1.1
  Levels of Programming ....................................................... 1.1
  Interrupt Scheduling ......................................................... 1.3
  Hardware and Programming Structure Summary .................. 1.4

Network Control Program Overview and Data Flow ............... 2.1
  Identifying the Major Components ..................................... 2.1
  Network Control Program Supervisor .................................. 2.1
  Channel Adapter IOS ......................................................... 2.2
  Path Control ...................................................................... 2.2
  Network Control Program Physical Services ......................... 2.3
  Boundary Network Node ................................................... 2.3
  Link Scheduler .................................................................. 2.3
  SDLC Routines .................................................................. 2.4
  BSC/SS Processor ........................................................... 2.4
  Overview Summary ......................................................... 2.4

Network Control Program Supervisor .................................. 3.1
  Purpose of the Supervisor ................................................ 3.1
  Task Management .......................................................... 3.2
  Task States ...................................................................... 3.10
  Dispatching Tasks .......................................................... 3.12
  Supervisor SVC Services .................................................. 3.16
  Supervisor Summary ....................................................... 3.16

Channel Adapter IOS .......................................................... 4.1
  Channel Adapter Definition ............................................. 4.1
  Host Writes to the NCP ................................................... 4.2
  Host Reads from the NCP ................................................. 4.2
  Type 1 and 4 Channel Adapter ......................................... 4.4
  Type 2 or Type 3 Channel Adapter ................................... 4.5
  Channel Adapter IOS Performance .................................. 4.9
  Channel Adapter Summary .............................................. 4.11
  Channel Adapter IOS Quiz ............................................... 4.11

Path Control ........................................................................ 5.1
  Path Control Out ............................................................ 5.1
  Path Control In .............................................................. 5.5
  Path Control Summary .................................................... 5.5
  Path Control Quiz .......................................................... 5.6

Network Control Program Physical Services ......................... 6.1
  Purpose of NCP Physical Services .................................... 6.1
  Session Hierarchy .......................................................... 6.1
  Physical Services Block (PSB) .......................................... 6.4
  Physical Services Components ........................................ 6.4
  Network Control Program Physical Services Flow ............... 6.8
Network Control Program Physical Services Control Block (PSB) 6.9
Network Control Program Physical Services Summary 6.10
Network Control Program Physical Services Quiz 6.10

Boundary Network Node (BNN) 7.1
Function of the Boundary Network Node 7.1
BNN Control Blocks 7.1
Host to PU PIU Processing 7.3
Physical Unit-to-Host PIU Processing 7.13
SDLC Switched Support 7.17
Boundary Network Node (BNN) Summary 7.21
Boundary Network Node Quiz 7.23

Local/Remote Link 8.1
Activation of a Remote NCP 8.1
Station Control block (SCB) PU Type 4 8.1
Initializing the Remote 8.2
Host to Remote PIU Processing 8.3
Remote to Host PIU Processing 8.3
Remote NCP 8.3
Local/Remote Link Summary 8.8
Local/Remote Link Quiz 8.9

Data Link Control 9.1
What is Data Link Control 9.1
Data Link Control (DLC) Control Blocks 9.2
Link Scheduler Initiation 9.5
XIO LINK to the Link Scheduler 9.6
Link Poll to Path Control Inbound 9.8
Termination and Restart of an XIO Run Command 9.9
Data Link Control Review 9.9
Data Link Control Quiz 9.11

BSC/SS Processor 10.1
Definition of the BSC/SS Processor 10.1
BSC/SS Major Control Blocks 10.1
BSC/SS Processor Components 10.5
BTU Commands for BSC/SS Resources 10.8
BSC and SS Sessions 10.12
BSC/SS Flow 10.14
BSC/SS XIO Processing 10.16
Character Service Program (CSP) Flow 10.18
Block-handler Routines 10.20
Multiple Terminal Access (MTA) 10.24
BSC/SS Processor Summary 10.28
BSC/SS Processor Quiz 10.28

Service Aids and Diagnostics 11.1
Purpose of Service Aids and Diagnostics 11.1
Dynamic Panel Display 11.1
Line Test 11.1
Address Trace 11.2
Channel Adapter Trace 11.2
Line Trace 11.2
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error and Statistic Recording</td>
<td>11.2</td>
</tr>
<tr>
<td>Online Tests</td>
<td>11.3</td>
</tr>
<tr>
<td>Abend</td>
<td>11.3</td>
</tr>
<tr>
<td>Appendix A</td>
<td></td>
</tr>
<tr>
<td>Appendix B</td>
<td></td>
</tr>
<tr>
<td>Appendix C</td>
<td></td>
</tr>
<tr>
<td>Abbreviations</td>
<td></td>
</tr>
</tbody>
</table>
Objective
Upon completion of this topic the student should be able to identify the levels of programming and interrupt scheduling.

Review of Hardware Facilities
Before going on to the components of the SDLC network, this section reviews the hardware facilities which are used in the programming design, as well as dispatching code and techniques. In later sections the modules are related to an interrupt level or to a dispatched module. In either case a knowledge of the hardware and dispatcher is required.

Levels of Programming
Because the communications controller is an interrupt-driven unit, the NCP directing the operation of that unit is made up of smaller programs or levels. Interrupts can be caused by the channel, the communication lines, or the program itself.

The controller has five program levels. Program level 1 has the highest priority; program level 5 (referred to as the background level) has the lowest priority. Because level 5 has the lowest priority, level 5 code runs when levels 1 through 4 are not executing. For a complete description of the five levels of the controller and the interrupt facility, refer to IBM 3704 and 3705 Communications Controllers Principles of Operation (GC30-3004), Chapter 2: System Structure.

Figure 1.1 is a chart of the programming levels indicating the operations performed at each level, the starting address, and the means by which the level gets control. Note that when an attempt is made to execute an instruction at location X'0000', the NCP detects a 'branch to zero', regardless of the program level.
<table>
<thead>
<tr>
<th>Level</th>
<th>Operations Performed</th>
<th>Starting Address</th>
<th>Means of Getting Control</th>
</tr>
</thead>
</table>
| 5     | • Interpretation of commands from host.  
       • Control of polling and addressing.  
       • Decoding and execution of system examination and modification requests.  
       • Data handling functions.  
       • Block handling functions.  
       • Initiation and termination of line I/O.  
       • Panel functions.  
       • Boundary network node (BNN) processing.  
       • Physical services functions.  
       • Function management. | N/A | • Default from other four levels. |
| 4*    | • Buffer management.  
       • Queue management.  
       • Task dispatching.  
       • Supervisory services. | X'0180' | • PCI.  
       • SVC. |
| 3     | • Interval timer functions.  
       • Handling of panel functions.  
       • Channel adapter management.  
       • Communication processing deferred from level 2.  
       • Intermediate network node (INN) processing. | X'0100' | • PCI.  
       • Type 1, type 2, type 3 and type 4 CA.  
       • Interval timer.  
       • Panel INTERRUPT push button. |
| 2     | • Buffer service for communication lines.  
       • Character service for communication lines.  
       • Bit service for communication lines. | X'0080' | • Type 1, type 2, and type 3 scanner. |
| 1     | • Machine check handling.  
       • Program check handling.  
       • Adapter check handling.  
       • IPL procedure.  
       • Address trace facilities. | X'0010' | • IPL.  
       • Address exception check.  
       • Type 1, type 2, type 3 and type 4 CA checks.  
       • Type 1, type 2, and type 3 scanner checks.  
       • Address compare.  
       • Protection check.  
       • Input/output check. |
| x     | • Detection of branch to zero. | X'0000' | • Branch to zero. |

* Level 4 operations can also be performed at levels 1 and 3.

Figure 1.1. Program Levels
Interrupt Scheduling

Level 1, Address X'0010'

When a level 1 interrupt occurs, control is given to the level 1 router, which is located at address X'0010'. By examining the contents of external registers, the router determines the cause of the interrupt and passes control to one of the following handlers: the program exception check-handler, the address trace module, the channel adapter check-router, the communications adapter check-handler, or the abend module.

Level 2, Address X'0080'

When a level 2 interrupt occurs, control is given to address location X'0080'. The level 2 router determines if the interrupt was a normal character service request. The address of the router is located in the CCB. The level 2 router itself processes hardware error and exceptional conditions.

Level 3, Address X'0100'

When a level 3 interrupt occurs, control is given to address location X'0100'. By examining the external registers, the level 3 router determines the cause of the interrupt, then passes control to one of the following interrupt handlers: the channel adapter input/output supervisor, the communications-line timer service, the communications control program queue-handler (signaled by a PCI), or the panel support module.

Level 4, Address X'0180'

When a level 4 interrupt occurs, control is given to address location X'0180', the level 4 interrupt handler. An SVC interrupt occurs when a supervisor macro is issued in program level 5. The program issuing the macro specifies certain parameters. After decoding the SVC code, the supervisor nucleus loads these parameters into registers and calls the appropriate supervisor SVC routine to process the request. If the interrupt is a program-controlled interrupt (PCI), the interrupt handler branches to the address in the PCI vector table to process the request.

Level 5

All level 5 tasks are dispatched by the level 4 task dispatcher. The entry point of each task is provided as a field in the queue control block (QCB), which is scheduled by placing the QCB in one of the supervisor dispatching queues. The dispatching of level 5 tasks is covered later in the supervisor section.

Interrupt levels may be masked off to prevent interrupts. Levels 2 through 5 may be totally suppressed. Level 1 may be masked to ignore channel adapter and scanner interrupts for test purposes. If the level is not masked off and an
interrupt is pending, the interrupt is not allowed if any of the following conditions exist:

- A higher-priority interrupt request is present.
- The program level to be interrupted is already entered ('interrupt entered' latch is on).
- The program level to be interrupted is masked.
- A type 3 communication scanner cycle-steal request exists.
- A type 2 or 3 channel adapter cycle-steal request exists.

At the time an interrupt is honored, the 'interrupt entered' latch for that program level is turned on. The 'interrupt entered' latch is a hardware latch which signals the controller that the associated program level has been entered. As long as this latch is on, no other interrupts to this program level are honored. The general registers and condition latches for this level are safe from change by another interrupt. The 'interrupt entered' latch is turned off either by an EXIT instruction executed at this level or by a reset condition to the entire controller.

After each instruction is executed, the controller tests for priority conditions before executing the next instruction. The type 3 communications scanner and type 2 or 3 channel adapter cycle-steal requests occur between instructions. In addition, a higher-priority program level may need control. If level 3 code is executing ('interrupt entered' latch on) before executing each additional instruction the controller checks, in sequence, the 'level 1 entered' latch, 'level 1 pending' latch, 'level 2 entered' latch, 'level 2 pending' latch, and 'level 3 entered' latch. This sequence returns control to level 3 for another instruction execution.

If a second level 3 interrupt was pending, it is not checked in the sequence because the 'interrupt entered' latch is tested first. If the 'level 2 pending' latch was set, as in the previous example, level 2 code starts executing. The 'level 2 interrupt entered' latch is turned on and level 2 executes until an EXIT instruction turns off the 'interrupt entered' latch. When the between-instruction check is made after the level 2 EXIT instruction, the level 2 interrupt entered latch is off, so the 'level 2 interrupt pending' latch is checked. If that latch is on, the level 2 code executes again with the 'interrupted entered' latch turned on a second time. If the 'level 2 pending' latch is not on, the check returns control to level 3 where the 'interrupt entered' latch is still on. The level 3 code continues, unaware of the interrupt.

The IBM 3704 and 3705 Communications Controllers provide hardware support for five programming levels. The first four levels are interrupt-driven code, each having an absolute hardware address to begin instruction execution. The fifth level is dispatched under the control of the level 4 supervisor.
Network Control Program Overview and Data Flow

**Objective**
Upon completion of this topic, the student should be able to identify the major programming components of NCP, and the flow of control and data between major components.

**Identifying the Major Components**
This section identifies the major components of the network control program and the program level in which the components operate. This material serves as the foundation upon which the detail of future sections is built. The major components of the network are covered in the order of subsequent topics.

<table>
<thead>
<tr>
<th>CHANNEL ADAPTER ICS</th>
<th>PHYSICAL SERVICES PROCESSOR PSB</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHB COB</td>
<td>BOUNDARY NETWORK NODE (BNN)</td>
</tr>
<tr>
<td>PATH CONTROL SIT SVT RVT</td>
<td>CUB LUB</td>
</tr>
<tr>
<td></td>
<td>SCB</td>
</tr>
<tr>
<td></td>
<td>BSC/SS PROCESSOR LCB DV8</td>
</tr>
<tr>
<td></td>
<td>CICP &amp; BIT/CHAR. SERVICE ACB</td>
</tr>
<tr>
<td></td>
<td>LINK SCHED &amp; SDLC LKB ACB</td>
</tr>
</tbody>
</table>

Figure 2.1. NCP Components

**Network Control Program Supervisor**
The NCP supervisor serves primarily as the interface between the background tasks running in level 5 and the routines running in levels 1, 3, and 4. When levels 1, 3, or 4 require data to be processed by background tasks, the tasks are scheduled via the supervisor. The supervisor queues the data and schedules the correct background processing task. Conversely, as background tasks require initiation of input or output, manipulation of queues, management of buffers, and similar tasks, the task requests are presented to the supervisor. The supervisor then processes those requests as required.

The supervisor executes in level 4. The primary control blocks used by the supervisor are: byte direct addressables (XDB), halfword direct addressables (HDB),...
(XDH), word direct addressables (XDA), queue control blocks (QCB), path information units (PIU), and the SVC vector table. The supervisor code is executed and provides services for all of the routines identified in this section.

Channel Adapter IOS

The channel adapter module is used to monitor and control the hardware channel adapters within the 3704 or 3705 controller during a data transfer to or from the host. There are four types of channel adapters; however, only two types are used for programming purposes within the controller. Therefore, there are only two types of IOS modules: one to control the type 1 or type 4 adapter, the other to control the type 2 or type 3 adapter.

Type 1 or 4 Adapter

The 3704 supports a type 1 adapter only. A 3705 can have a type 1 or 4 adapter for NCP mode or PEP mode, or the type 1 or 4 adapter can be used for emulation programming for (EP), with a second adapter for NCP mode only. A type 1 or 4 channel adapter for operation in NCP mode uses a channel operation block (COB). The channel modules operate in program level 3 via a level 3 interrupt from the channel adapter hardware.

Type 2 or Type 3 Adapter

The 3705 operates in NCP mode with a type 2 channel adapter to a single processor or a type 3 channel adapter to two tightly coupled multiprocessors. A type 2 or type 3 channel adapter uses a channel control block (CRB). The channel module operates in program level 3 via a level 3 interrupt from the channel adapter hardware.

Path Control

Path control code is executed on outbound PIUs by a branch from the channel code. Path control inbound, which is different code, is executed on inbound PIUs by a branch from the link scheduler.

Path Control Out

'Path control out' directs the flow of path information units (PIUs) from the channel adapter IOS to its proper destination. 'Path control out' uses the destination address field (DAF) from the PIU to access entries in the subarea index table (SIT), subarea vector table (SVT), and resource vector table (RVT). The 'path control out' routine locates the appropriate path for the PIU and places the PIU on a queue control block (QCB) for processing by NCP physical services, boundary network node, link scheduler (SCB), or the BSC/SS processor. This module operates in program level 3 via a branch from the channel IOS.

After the boundary network node processing completes, an outbound PIU is passed to 'path control out delayed', which converts the PIU from FID1 to FID2 or FID3, segments the FID2 or FID3 as required, and places the PIU on the link outbound queue.

Path Control In

'Path control in' is divided into two parts: immediate and delayed. When a PIU is received on a link, 'path control in immediate' is invoked by a branch from the link scheduler. 'Path control in immediate' checks for a PIU source of a remote controller (SCB); if the PIU is from a remote, the PIU is immediately queued on the channel intermediate queue for the host. If the PIU is
not from a remote, the PIU is queued on the CUB link inbound queue and 'path control in immediate' exits from level 3.

The PIU queued on the CUB link inbound queue invokes a level 5 task of 'path control in delayed'. This task processes the PIU to identify the source logical unit or CUB physical services, then branches to an appropriate boundary network node connection point manager IN (CPM-IN) for additional processing.

NCP physical services interfaces with system services control point (SSCP) of the host to provide control functions for the NCP. NCP physical services provide functions such as activating or deactivating links, contacting physical units, and other control functions. These modules use the physical services control block (PSB). The physical services routines operate in program level 5 via the task dispatcher. ‘Path control out’ schedules physical services by PIU requests. Responses to SSCP are queued directly to channel adapter IOS.

The NCP physical services has a ‘connection point manager-in’ queue (inbound error-handler queue), which is invoked by the link scheduler at the completion of a ‘dial’, ‘answer’, ‘contact’, or break in a link.

Boundary Network Node

The boundary network node modules provide the interface to SDLC type 1 and type 2 devices. Remote 3704 and 3705 controllers are not included in this code, as PIUs destined for a remote are enqueued directly on a station control block (SCB) by 'path control out'. These modules control the session initiation and session status for the physical units and logical units attached to this 3704 and 3705 controller. These modules operate in program level 5 via task dispatching. Type 1 and 2 physical units are defined by the common physical unit control block (CUB); logical units are defined by the logical unit control block (LUB). BNN modules are scheduled when they receive a PIU from 'path control out'. BNN enqueues PIUs from the host on a link outbound queue for the link scheduler. BNN enqueues PIUs for the host on the channel intermediate queue.

Link Scheduler

The link scheduler executes in program level 3. The link scheduler is invoked for a specific link by an ‘activate link’ command. The link scheduler has two basic functions: data transfer or command processing.

The link scheduler uses the service order table (SOT) to locate the physical units for that specific link. Each physical unit is checked for active status. If the physical unit is active, the link outbound queue is checked for outbound PIUs to transmit. After any allowed outbound PIU traffic has been sent, the physical unit is polled for inbound PIUs. When all physical units have been checked for data service at least once, the link scheduler switches to control functions. One control function ('dial', 'answer', 'contact', 'discontact') is attempted for one physical unit before the link scheduler returns to data transfer mode.

If there are no outbound PIUs for a link and if no active physical unit has inbound PIUs in response to polling, after the control cycle the scheduler suspends polling for a user-specified pause. Data queued to be transmitted is sent, but polling is suspended.
The link scheduler uses the link control block (LKB) to schedule link operations and maintain link status. The LKB is generated by a LINE macro of an SDLC group. The common physical unit block (CUB) or station control block (SCB) is used to schedule the station control and maintain station status for any SDLC physical unit.

**SDLC Routines**

The SDLC routines are used for the actual transmission of data on the link. The adapter control block (ACB) is used for link control. These routines operate in program level 2 via an interrupt from the hardware scanner.

SDLC routines are initiated by the link scheduler, providing addresses of processing routines in the character control block (CCB) and enabling the link for interrupts to begin processing.

**BSC/SS Processor**

The BSC/SS processor supports the BSC/SS devices in NCP mode that are attached to this communications controller. The processor uses the line control block (LCB) and the device control block (DVB) to schedule and control commands issued to these devices. Command processors are used to define the commands and the work scheduler is used to schedule the necessary tasks to complete the command. Command decoders and initialization routines initialize the lines and control their operation; character-service routines handle the actual transmission of data across the line. Both types of routines use the adapter control block (ACB). The command processors, work scheduler, and scheduler tasks operate in program level 5 via task dispatching. The command decoders and initialization routines operate in program level 3 via a PCI level 3. The character service routines operate in level 2 via a hardware interrupt from the scanner.

Unless BSC or SS devices are also defined for NCP mode, BSC/SS processor support is not included in a network of SDLC terminals. The processor support routines are not included if BSC/SS devices are operated in emulation mode of a partitioned emulation program (PEP).

**Overview Summary**

There are four basic paths through the local controller from the host. The path that is taken depends upon the destination of the path information unit (PIU). The destination and sequences are as follows:

- Physical services destination
  Channel adapter IOS, path control out, physical services processor

- SDLC device or logical unit destination
  Channel adapter IOS, path control out, boundary network node, link scheduler, SDLC routines

- Remote controller destination
  Channel adapter IOS, path control out, link scheduler, SDLC routines

- BSC/SS processor destination
  Channel adapter IOS, path control out, BSC/SS processor, CICP
There are four paths through the local controller to the host, as follows:

- Physical services source
  Physical services processor, channel adapter IOS

- Type 1 or type 2 physical or logical unit source
  SDLC routine, link scheduler, path control in immediate, path control in delayed, boundary network node, channel adapter IOS

- Type 4 physical unit source
  SDLC routine, link scheduler, path control in immediate, channel adapter IOS

- BSC/SS processor source
  CICP, BSC/SS processor, channel adapter IOS
Network Control Program Supervisor

**Objective**
Upon completion of this topic the student should be able to identify and locate the supervisor dispatching queues, identify supervisor services, and explain how the services are requested.

**Purpose of the Supervisor**
The NCP supervisor serves primarily as the interface between background tasks running in level 5 and routines running in levels 1, 3, and 4. When levels 1, 3, or 4 require data or a stimulus to be processed by the background tasks, the task is scheduled via the supervisor. The supervisor queues the data and schedules the correct background processing task. Conversely, as background tasks require initiation of input or output, manipulation of queues, management of buffers, etc., the task requests are presented to the supervisor. The supervisor then processes those requests as required.

The supervisor can be entered from the level 4 interrupt handler or via a branch from levels 1, 3, or 4. The supervisor is entered from levels 1, 3, and 4 as a result of supervisor macros which expand to include a branch to the supervisor. This branch is created because of the SUPV=operand of any of the supervisor macros being coded YES. The supervisor routine is then being executed as level 1 or level 3 code rather than level 4 code because it was entered directly, not because of an interrupt. Level 5 always uses a level 4 SVC interrupt to request supervisor services. If level 3 has placed work on the supervisor dispatching queue, the level 4 PCI interrupt latch is set for future processing.

Entry to the level 4 interrupt handler at address X‘180’ is caused in one of two ways: a level 5 SVC macro or a level 4 PCI.

The level 5 SVC is created by an EXIT instruction. The EXIT instruction and two-byte SVC code immediately following are generated by a level 5 macro which is coded with an operand of SUPV=NO. In this case, the flow is through the level 4 interrupt handler, which uses the SVC code supplied by the level 5 macro expansion to index into the SVC vector table. This table contains pointers to the various supervisor macro routines. The SVC code is the first seven bits of the sixteen-bit field. The remaining nine bits are qualifiers of the SVC.

A level 4 PCI interrupt also causes the level 4 interrupt handler to get control. In this case, the level 4 interrupt passes control to one of three routines via a branch table.

Normally the first entry of the branch table points to the second entry and the second entry points to the third. The third entry always points to the dispatcher. A level 4 PCI interrupt normally causes the dispatcher to get control.

When the free buffer threshold is reached, the second entry is replaced with the address of the routine to generate a slowdown message. Each time the LEASE buffer routine is executed by a branch from level 3 or SVC from level 5, the count of remaining buffers is checked against the threshold value. If slowdown mode is required, the address of the slowdown message routine is

NCP Release 5 Data Flow 3.1
placed in the branch table and slowdown bits are set in the direct addressable area.

If an unconditional buffer request is made and no buffers are available, levels 4 and 5 can be disabled. Level 5 is disabled by masking off level 5, and the address of the buffer allocation routine is placed in the first entry of the dispatcher branch table.

The entry code at X'180' is entered for SVC and PCI interrupts. An IN X'7F' provides a bit to define whether a PCI or SVC caused the interrupt. The result causes the supervisor to go either to the SVC interrupt handler or to the PCI branch table.

**Task Management**

A task in the network control program (NCP) is defined as a portion of code and a queue of data upon which the code operates. In the NCP, tasks are executed in level 5 only. If one portion of code operates upon two or more separate queues of data, the task dispatcher handles this portion of code as two or more separate tasks. The background level (level 5) of the NCP is made up of several tasks that work together to schedule lines and process messages.

A task is defined at NCP generation when a queue control block (QCB) is assembled and linked to a unit of code. As queues become activated, their associated tasks are scheduled and initiated by the task dispatcher. Input queues (input to a task) are activated by the enqueuing of data to the queue. Enqueuing is provided by level 3 when a PIU is received over the communication lines or over the channel, or when the enqueuing is provided by one task passing control to another task. Pseudo-input queues (recording a stimulus for the task, but providing no data as input to the task) are activated by triggering the task upon the occurrence of some stimulus, such as a panel display request.

There are several control blocks used by the dispatcher. Before we cover the method used by the supervisor, the topics that follow will acquaint you with some of the control blocks.

*The IBM 3704 and 3705 Program Reference Handbook* (GY30-3012), can be used as a reference.

**Direct Addressables (XDB, XDH, XDA)** There are three fixed areas of special pointers or special fixed data. These areas are:

- **Byte direct addresses (XDB)**
  
  X'680' to X'6FF'

- **Halfword direct addresses (XDH)**
  
  X'700' to X'77F'

- **Word direct addresses (XDA)**
  
  X'780' to X'7FF'

A special form of instruction with a base register of zero allows an implied base to refer to these fields, with the displacement providing the offset from the beginning of the area. The instructions are:

*Insert Character*

\[ \text{IC} \ 5(0),16(0) \]
The 'insert character' instruction inserts the value at base location X'680' plus decimal 16 (X'10') into register 5 byte 0, for an effective address of X'690'. The true buffer size for this system, including the four-byte prefix, is at X'690'.

Store Character

STC 5(0),16(0)

This instruction stores the value in register 5 byte 0 at location X'690' (X'680' plus X'10').

Load Halfword

LH 6,84(0)

The 'load halfword' instruction places the current free buffer count from X'700' plus decimal 84 (X'54') into register 6, creating an effective address of X'754'.

Store Halfword

STH 6,96(0)

The 'store halfword' instruction uses the value in register 6 to set the value of the system abend code at X'760' (X'700' plus decimal 96). The NCP sets a value at X'760' to indicate the reason the failure occurred.

Load

L 6,96(0)

The 'load' instruction moves the address of the last byte of storage from X'7E0' to register 6.

Store

ST 6,68(0)

The 'store' instruction records a pointer to the first free buffer at location X'7C4'.

The direct addressables provide key status indicators and pointers to the system control blocks. As the various NCP routines are covered, related direct addressables fields which provide status indicators as an aid in debugging are referenced. These are some of the initial fields which may be of special interest:

Byte direct addressables (XDB) X'680' to X'6FF'

- X'685' Control byte for dispatcher flags
- X'687' BUILD macro buffer size
- X'689' Buffer pool and network status
- X'68A' General communications byte
- X'68B' Identifies program as NCP, EP, or PEP
- X'692' General communication byte
- X'693' SDLC subarea mask
- X'694' SDLC element mask
Halfword direct addressables (XDH) X'700' to X'77F'

- X'710' to X'72B' PEP emulation queue pointers
- X'744' to X'752' NCP level 4 task queue pointers
- X'754' Current free buffer count
- X'756' Free buffer threshold count plus one
- X'758' Number of communications lines
- X'75A' Level 5 system active queue control block
- X'760' System abend code
- X'770' Maximum byte count to host per host start I/O
- X'772' Pointer to the channel control block (CHB or COB)

Word direct addressables (XDA) X'780' to X'7FF'

- X'7BC' Lagging address register (LAR)
- X'7C4' Pointer to first free buffer
- X'7D0' Remembrance of the last buffer in the buffer pool
- X'7D4' Remembrance of the first buffer in the buffer pool
- X'7D8' Pointer to extended halfword direct addressables (HWE)
- X'7E0' Address of last byte of storage
- X'7E8' Pointer to the resource vector table (RVT)
- X'7F0' Pointer to the logical end of system free buffer pool

Queue Control Blocks (QCB) The queue concept is basic to an understanding of the data flow within the NCP. A queue is a group of either data blocks (PIU or BCU) or queue control blocks (QCBs) connected first through last by address pointers. First in, first out (FIFO) is the basic mode of queue manipulation; however, last in, first out (LIFO) mode is also used.

A queue control block (QCB) has two queue pointers. One points to the first element in the queue. The first element points to the second, the second points to the third, etc. The second queue pointer points to the last element on the queue. If both addresses are zero, there are no elements in the queue.

There are three types of queues: input, pseudo-input, and work. Each type of queue provides different program support.

**Input queues**

An input queue contains elements to be processed by the task identified by the QCB. Some of the fields are:

- X'00' Shifted address of first element queued
- X'02' Shifted address of last element queued
- X'04' Task state
- X'05' 1010 1xxx indicates this is an input QCB
- X'06' Shifted address of next QCB on this queue
Network Control Program Supervisor

X'08' Major control block displacement, provides the displacement from the beginning of the control block which contains this QCB to the first byte of the QCB.

X'09' Task dispatching priority

X'08' Full address of task entry point

Placing an element in a queue with ENQUE ACTV=YES puts a task in the pending state. If no task is active, the pending task becomes the active task.

_Pseudo-input Queue_

A pseudo-input queue contains no elements. It has the same format as the input queue, but the task is triggered by a stimulus rather than by the enqueueing of an element. An example of a pseudo-input queue is the panel queue. When the interrupt key is pressed on the 3704 or 3705 panel, a level 3 panel interrupt occurs. When level 3 determines that the interrupt was from the panel, level 3 branches to the level 4 supervisor routine which places the panel QCB on a dispatching queue.

The format of the pseudo-input queue is the same as the standard input queue. The only difference between an input queue and a pseudo-input queue is the means of dispatching the pseudo-input queue without data.

_Work queue_

A work queue does not have a task entry point. It is used as a queue to hold elements. The work queue is only eight bytes in length. The fields are as follows:

- X'00' Shifted address of first element queued
- X'02' Shifted address of last element queued
- X'04' Reserved
- X'05' 1010 0xxx indicates that this queue is a work QCB
- X'06' Shifted address of next QCB on this queue

_Path Information Unit (PIU)_ The element placed in a queue is either a queue control block (QCB), a block control unit (BCU) used in the BSC/SS code, or a path information unit (PIU). The placing of a PIU on a QCB normally triggers scheduling. The flow of the network control program is initiated by receiving a PIU from the channel or line and passing the address from one queue to the next for processing.

The PIU is received in one or more NCP buffers. The PIU is made up of a transmission header (TH), request/response header (RH), and request/response unit (RU). The PIU is that portion received from the host or from the lines.

In addition to the area specified on the BUILD macro BFRS operand, each NCP buffer requires a four-byte prefix for control purposes. The size of each buffer specified for the user is given in XDB at X'687'. The true buffer size is in XDB at X'690'. The buffer prefix field on each buffer is specified as follows:
X'00' Buffer prefix chain field, shifted address of the next buffer in the chain, or zero if the last in a chain.

X'02' Buffer prefix data offset field. This field provides the offset from the buffer prefix to the first byte of PIU text.

X'03' Buffer prefix data count field. This field specifies the quantity of data from the offset that is valid in the buffer.

In the first buffer of a PIU is an event control block (ECB). The ten-byte ECB immediately follows the buffer prefix. The first buffer prefix offset of X'0A' provides the offset past the ECB to the first byte of FID1 PIU. The PIU actually starts in the fifteenth byte of the buffer, including prefix.

Including the offset from the beginning of the buffer, the ECB fields are as follows:

X'04' Block status flags. Specifies if the PIU is in a queue.

X'05' Event status flags. Specifies if the event is satisfied and if the task is to be dispatched when this PIU is first in the queue.

X'06' ECB chain pointer. If multiple PIUs are queued on one QCB, this field is a shifted address of the next PIU on the queue chain. The queue manipulation macros use an offset of X'06' for chain addresses in QCBs and in the first buffer of a block.

X'08' PIU text count or set time interval

X'0A' Address of QCB for waiting task or hold area for blocks

X'0C' UIB type field. This field identifies the destination resource type. The values are as follows:

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000 0000</td>
<td>Communications controller</td>
</tr>
<tr>
<td>100x xxxx</td>
<td>Line</td>
</tr>
<tr>
<td>010x xxxx</td>
<td>Device</td>
</tr>
<tr>
<td>xxx1 xxxx</td>
<td>Input</td>
</tr>
<tr>
<td>xxxx 1xxx</td>
<td>Output</td>
</tr>
</tbody>
</table>

X'0D' UIB status. If nonzero this field indicates various errors.

If the type is FID0 or FID1, the next byte after the buffer prefix and unit information block is the first byte of the PIU. The FID2 buffer prefix offset specifies an offset of X'0E' and these four bytes (X'0E' through X'11') are not used. The FID3 buffer prefix offset is eight bytes (X'0E' through X'15') and the eight bytes are not used. The FID0 format is used only for text transfer between the host system and the BSC/SS router. FID1 is identified by a bit pattern of xx00 xxxx at X'0E'.

**FID0**

The type 0 PIU is a field identification type 0 (FID0). This format is used for all text transfers between a host application and a BSC/SS terminal. The FID0 is received from the host and sent to the BSC/SS converter. The FID0 is converted to a block control unit (BCU) for the BSC/SS processor. Text
from a BSC/SS terminal is received in a BCU format buffer, sent to the BSC/SS converter, and converted to a FIO0 before being sent to the host.

Including offsets from the beginning of the buffer, the format of the FIO0 is as follows:

Transmission header (TH)
- X'0E'  Transmission header. This field identifies the PIU as type 0 by xx00 xxxx in this byte.
- X'0F'  Reserved
- X'10'  Destination network address
- X'12'  Origin network address
- X'14'  Sequence number
- X'16'  Text count of the RH plus RU (excludes TH)

Request/response header (RH)
- X'18' - X'1A'  These fields are ignored in a FIO0 PIU.
- X'1B'  Request/response byte 3. This field is a pad byte to align the RU on a halfword boundary.

Request/response unit (RU)
- X'1C'  RU0 byte 0. BTU command field. This field is covered in IBM 3704 and 3705 Program Reference Handbook (GY30-3012), Section 3: BTU Command and Modifiers.
- X'1D'  RU0 byte 1. BTU command modifier
- X'1E'  and X'1F' RU0 bytes 2 and 3. BTU flags
- X'20'  RU0 byte 5. BTU system response. This field is covered in IBM 3704 and 3705 Program Reference Handbook (GY30-3012), Section 7: BTU Responses.
- X'21'  RU0 byte 6. BTU extended response.

FID1
The type 1 PIU is a field identification type 1 (FID1). This format is used for all control commands, and all text transfers between the host and boundary network node (BNN). If the PIU is transferring to a remote NCP, the FID1 is sent unchanged to the remote.
Including offsets from the beginning of the buffer, the format of the FID1 is as follows:

Transmission header (TH)

X'0E'  Transmission header. This field identifies the PIU as type 1 by xx01 xxxx in this byte. This byte also specifies whether this PIU is the first middle, last, or only PIU segment. PIU segmenting occurs when a PIU from the host has a length greater than that defined by the MAXDATA operand of a PU macro defines.

X'0F'  Reserved

X'10'  Destination network address

X'12'  Origin network address

X'14'  Sequence number

X'16'  Text count of the RH plus RU (excludes TH)

Request/response header (RH)

X'18'  Request/response byte 0. This byte specifies whether the PIU is a request or response, control or data, system control or function management, against flow or with flow, formatted or unformatted. PIU chaining by an application program is defined in this field, which specifies whether this is the first, middle, last, or only PIU element.

X'19'  Request/response byte 1. This field specifies that an FME is requested/sent, an RRN is requested/sent, and an exception response is requested/sent. VPACING and PACING use the pace bit in this field.

X'1A'  Request/response byte 2. This field specifies bracket protocol, EBCDIC or ASCII code, and change direction (HDX only).

Request/response unit (RU) - Network commands only.

X'1B'  RU1 byte 0. This field is covered in IBM 3704 and 3705 Program Reference Handbook (GY30-3012), Section 4: NCP Network Commands. The value of this field varies, based on the value of the RH byte 0.

X'1C'  RU1 byte 1. Used for function management requests.

X'1D'  RU1 byte 2. Request code for function management.

X'1E'  Network address for SCP function management requests. A command to activate a link or contact a device is addressed to NCP physical services in the DAF; the device to be controlled by the command is addressed by this field.

X'20'  The data beginning at this address varies, based on the type of command. For additional information, refer to IBM
Data PIU formats have user data starting at X'1B' immediately following the TH/RH.

**FID2**

The type 2 PIU is field identification type 2 (FID2). This format is used for all control commands and all text transfers between the boundary network node (BNN) routine and support of type 2 physical units (3770, 3600, 3650, 3660, 3790). The FID1 is received from the host and converted to a FID2 before being sent to the type 2 physical unit. A FID2 from a type 2 physical unit is converted to a FID1 by the BNN code before being sent to the host.

The FID2 is created from a FID1 by converting the two-byte OAF and DAF fields to one-byte fields and deleting the two-byte transmission header (TH) count field. This conversion provides a total of four bytes deleted from the FID1 requirements. Shifting the fields to the right places the following fields in the original FID1 buffer:

**Transmission Header (TH)**

- X'0E'  Reserved four-byte area
- X'12'  Transmission header. xx10 xxxx in this byte identifies the PIU as type 2.
- X'13'  Reserved
- X'14'  Destination
- X'15'  Origin
- X'16'  Sequence number

**Request/response header (RH).** Same as FID1

**Request/response unit (RU).** Same as FID1

**FID3**

The type 3 PIU is field identification type 3 (FID3). This format is used for some control commands and all text transfers between boundary network node (BNN) code for support of type 1 physical units (3767, SDLC 3270). The FID1 is received from the host and converted to a FID2 with the normal FID2 processing. The FID2 is converted to a FID3 before being sent to the type 1 physical unit. The FID3 commands directed to a 3270 are processed by the NCP and are not sent to the SDLC 3270.

The FID3 is created from the FID2 by converting the six-byte transmission header (TH) of the FID2 to a two-byte TH of the FID3. The FID2 destination of one byte is converted to the low-order six bits of the last byte of the TH. The two leftmost bits specify the following:

- Bit 0 - 1 = to/from application, 0 = to/from SSCP
- Bit 1 - 1 = to/from logical unit, 0 = to/from physical unit
The first byte of the FID3 TH identifies the type of PIU. Deleting the four bytes of the TH from the FID2 makes four more alignment bytes available. Shifting the fields to the right provides the following fields in the original FID1 buffer:

Transmission header (TH)

X'0E'  Reserved eight-byte area

X'16'  Transmission header. This field identifies the PIU as a type 3 by xx11 xxxx in this byte.

X'17'  Application or SSCP indicator and local address

Request/response header (RH). Same as FID1

Request/response unit (RU). Same as FID1

Task States

At any given point in time, a task can be in any one of four logical states. The four states, under program control, are: active, pending, ready, or disconnected. Initially all tasks are in the 'ready' state. The state is specified in the QCB at an offset of X'04' for all conditions. A 'ready' task is available for execution, but there is no element in its queue or no stimulus to initiate is; therefore it is not in a dispatching queue.

When an element is placed in a queue by an ENQUE ACTV=YES or when a TRIGGER macro is executed, the task is changed from 'ready' to 'pending and disconnected', and is placed on the one of the dispatching queues. The 'pending' status makes it available for execution. The 'disconnect' status identifies the QCB as having been triggered; therefore, it will not be triggered again until the 'pending' status completes (a task should not be placed in the dispatching queue when it is already in the dispatching queue). Subsequent elements placed on a triggered QCB can specify an automatic trigger when the PIU is first on the queue, providing that the UIB field at an offset of X'05' has a value of x1xx xxxx. When the level 4 supervisor looks for a task to make 'active', it takes the first pending QCB off the highest priority dispatching queue and schedules the task routine specified in the QCB field.

Only one task can be 'active'. The active task may issue SVC requests to level 4 for services, but remains the active level 5 task until it completes. If the active task is waiting for supervisor services (SVC), the second bit in byte 4 of the QCB is a 1 (task in wait state). A task completes by issuing a SYSXIT macro. When the task ends, the 'active' state must be ended, and the task changed from 'disconnect' to 'ready' by a QPOST macro. If the first element on the QCB specifies the task is to be dispatched (offset X'05' ECB status byte of the queued PIU), the QCB is triggered to the end of the dispatching queue and made 'pending'.

When a task is active, the byte direct addressable (XDB) at X'0685' has a value of x1xx xxxx. Figure 3.1 illustrates task state migration. In figure 3.1, all tasks going to 'pending' or 'active' also are 'disconnected'.

3.10 NCP Release 5 Data Flow
Network Control Program Supervisor

<table>
<thead>
<tr>
<th>TASK STATE ➔</th>
<th>Disconnect</th>
<th>Ready</th>
<th>Pending</th>
<th>Active</th>
</tr>
</thead>
<tbody>
<tr>
<td>TASK MGMT MACROS ➔</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QPOST</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TRIGGER conditional</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TRIGGER unconditional</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ENQUE ACTV=YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QCB dispatched</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

TRIGGER turns on both the disconnect and pending to avoid a second trigger.
OPOST turns off the disconnect (not ready) bit.

Figure 3.1. Task States

The following are the bit settings of the QCB in byte four, which indicates the status:

- 0xx0 xxxx: Ready and not disconnected
- 1xx1 xxxx: Pending and disconnected
- 0xx1 xxxx: Active and disconnected
- 1xx0 xxxx: Not valid

**Task Dispatching Priorities**

Tasks in the network control program have one of four task-scheduling priorities: appendage, immediate, productive, and nonproductive. All tasks having the same priority are queued together.

Appendage tasks have the highest priority in the system. When the current active task relinquishes control, appendage tasks are dispatched from the appendage queue on a first-in, first-out (FIFO) basis. Appendage tasks are generally initiated by character service at the end of a line input or output operation. However, they can also be initiated by the supervisor or by level 5 tasks.

Immediate tasks have the second highest priority. Once processing for a line has started, all tasks necessary to initiate the input or output on the line are given the immediate priority.

Productive tasks have the third highest priority. A task is classified as productive if the end result of its execution is the initiation of output on either the channel or the communication line.
Nonproductive tasks have the lowest priority in the system. A task is classified as nonproductive if it is not capable of starting input or output operations. Nonproductive tasks are not dispatched when the system is in slowdown mode.

There are definite reasons for having task scheduling priorities:

1. Appendage tasks are used to handle an exceptional condition as soon as possible.

2. Immediate priority improves performance. Once a task associated with a line in the idle state receives control, the performance is better if all the tasks necessary to initiate the transfer on this line are dispatched in succession before dispatching tasks associated with any other lines. The immediate priority accommodates such tasks.

3. Productive tasks have a high potential for freeing buffers and a low potential for allocating buffers.

4. Nonproductive tasks have a low potential for freeing buffers and a high priority for allocating buffers. Hence, productive tasks should be executed before nonproductive tasks.

The priority of a task can be changed dynamically by the CHAP macro.

The task dispatching queues are in the halfword direct addressables (XDH) at the addresses given below. The left address points to the first QCB queued and the right address points to the last QCB queued.

- Appendage queue: X'74E' X'74C'
- Immediate queue: X'746' X'744'
- Productive queue: X'74A' X'748'
- Nonproductive queue: X'750' X'752'

The QCB identifies which task dispatching queue is to be used. At QCB plus an offset of 9, the indicator to the TRIGGER macro is specified as follows:

- 10xx xxxx: Productive
- 010x xxxx: Immediate
- 001x xxxx: Appendage
- 000x xxxx: Nonproductive

### Dispatching Tasks

When level 4 is entered by PCI or by SVC, the supervisor at CXABTST checks for PCI or SVC. An SVC goes to CXASUPV to decode the macro using the SVC decode table. A PCI uses the three branch-table entries for processing. If levels 4 and 5 are disabled for buffer allocation, the first entry is primed with CXALEAS, the buffer allocation routine. Normally, the first entry contains the address of the second entry. If the system is in slowdown, the address of the routine to generate the slowdown entry message is in the second entry (CXAEXSS). Normally, the second entry contains the address of the third entry. The third entry contains the address of the task dispatcher (CXADISP). Figure 3.2 illustrates the supervisor processing.
The CXADISP task dispatcher checks whether level 5 is enabled by testing the byte direct addressable (X'685') to see if dispatcher service is required. If service is not required the supervisor checks for an active task in byte X'685'. If this bit is on, an EXIT from level 4 returns control to the current active level 5 task. If there is no active task, the supervisor searches the dispatching queues.

The queues are scanned in a sequence of appendage, immediate, and productive. The first entry found is dequeued, the QCB address is placed in the
level 5 register 2, the task entry point is placed in level 5 register 0, and the level 4 supervisor executes an EXIT to allow level 5 to begin execution.

If no queue entry is found through the productive queue, the supervisor checks for slowdown. If the slowdown is indicated from an earlier LEASE macro condition, level 5 is disabled (OUT X'7E'). The supervisor executes an EXIT. If the system is not in slowdown, the supervisor checks the nonproductive queue and dispatches an entry if one is found. If no entry is found, level 5 is disabled and an EXIT at level 4 places the controller in the wait state.

Figure 3.3 illustrates the dispatching sequence.
Figure 3.3. Dispatcher Execution Sequence

From: CXABTST
       CXASUPV

Check System
      Mask

Is task active
      bit on?  Yes  Return control
          to active task.
                   No

Is appendage
      dispatching
      queue empty?  No  Dequeue 1st element
                      in appendage queue
                      and dispatch
                      associated task.
                   Yes

Dequeue 1st element
      in immediate queue
      and dispatch
      associated task.

Is immediate
      dispatching
      queue empty?  No
                   Yes

Dequeue 1st element
      in productive queue
      and dispatch
      associated task.

Is productive
      dispatching
      queue empty?  No
                   Yes

Is slowdown in
      progress bit
      on?  Yes  Disable
              level 5
                   No

Dequeue 1st element
      in nonproductive queue
      and dispatch
      associated task.

Is nonproductive
      dispatching
      queue empty?  No  Disable
                        level 5
                   Yes
Supervisor SVC Services

In addition to the task management routines, the supervisor provides queue management, buffer management, and supervisory services. All of these facilities are provided by macros. Macros at level 5 provide an SVC interrupt by an operand of SUPV=NO. Levels 1, 3, and 4 branch directly to the appropriate routine by using a macro coded with an operand of SUPV=YES. The SVC is an EXIT instruction in level 5 with a 16-bit EXIT qualifier, seven-bit SVC identifier, and nine-bit SVC qualifier.

All of these services are covered in the 3704/3705 Assembler Language and Macro Instructions Student Text (SR20-4512).

Supervisor Summary

The supervisor provides service facilities to level 1 and 3 routines by direct branch. The supervisor provides level 5 services facilities by SVC interrupts to level 4. Entry to level 4 by PCI normally causes the supervisor to search the dispatching queues for queued work to be dispatched in level 5.

If a partitioned emulation program (PEP) is defined by the BUILD macro operand of TYPGEN=PEP or TYPGEN=PEP-LR, a concurrent emulation and NCP program is generated. The NCP performance may be degraded by heavy emulation usage. All emulator code executes at levels 1, 2, and 3. Therefore, the emulation code has priority over the NCP dispatcher and level 5 dispatched routines.

When register 0 (instruction address register) addresses an EXIT instruction (X'B840'), the program level terminates. If the EXIT is in level 5, there is an additional 16-bit SVC qualifier.

Network Control Program Supervisor Quiz

Using the dump provided in Appendix C, answer the following questions. This is a self-evaluation quiz, so please finish the quiz before referring to the answers in Appendix B.

1. Which program levels, if any, are active?

2. For each inactive level, if any, identify how you determined that the code was not active.

Criterion

If you could not answer both questions, you should review this section.
Channel Adapter IOS

Objective
Upon completion this topic, the student should be able to identify the types of channel control blocks and program support for each type of channel adapter, and name the user specifications which affect performance at the channel.

Channel Adapter Definition
To transfer data across the channel interface, we must give the NCP definitions of the channel adapter type and the host buffers.

The CHANTYP operand in the BUILD macro defines which type of channel adapter is installed in the 3704 or 3705. This operand also selects the appropriate control block and IOS module to be included in the generation. If CHANTYP=TYPE1 or CHANTYP=TYPE4, the generation selects the channel operation block (COB) and the COB IOS module. If CHANTYP=TYPE2 or CHANTYP=TYPE3, the generation selects the channel control block (CHB) and the CHB IOS module. If both type 1 or 4 and type 2 or 3 channel adapters are installed for a partitioned emulation program (PEP), the operand is coded with the high-performance channel adapter (type 2 or type 3) as the first operand and the low-performance channel adapter (type 1 or type 4) as the second operand. This channel combination allows a high-performance channel adapter for NCP mode with the emulation program code using the type 1 or type 4 channel adapter.

Note: The type 4 CA cycle-steal support is added in ACF/NCP, and is not available in NCP release 5.

The HOST macro provides the correct values to be placed in the selected control block for proper channel operation. The MAXBFRU and UNITSZ operands define for the NCP the input area that the host allocates on any channel read operation. The maximum amount of data in one PIU that the NCP may transfer to the host in a single channel transfer equals (MAXBFRU x UNITSZ) - BFRPAD. This calculation uses all allocated host buffers to contain one PIU.

The maximum number of PIUs which may be sent as a single channel transfer depends upon the size of the PIUs. In the previous paragraph, the example illustrated how one PIU may use all buffers. If all PIUs sent to the host are small enough so that each PIU and the buffer pads fit in a single host buffer, as many PIUs may be sent in a single channel transfer as there are host buffers available (MAXBFRU quantity).

Normally the PIUs do not all fit in a single host buffer nor will all the host buffers be required for one PIU. A combination of PIU lengths occurs where some PIUs require one host buffer while others require multiple host buffers.

The INBFRS operand determines the number of buffers the NCP should LEASE for a host 'write' operation to the 3704 or 3705. When the number of INBFRS is totally depleted, the INBFRS quantity is LEASED to continue the 'read' from the host. Once NCP buffers are allocated for a host 'write' they are allocated until used by this or a future host 'write'. At the end of a host 'write' the unused INBFRS are not returned to the available buffer pool.
Host Writes to the NCP

The host channel program must always start with a control command. A 'write' operation from the host must start with a 'write start zero' (WS0) or a 'write start one' (WS1). The first 'write' must be a WS0. After the first 'write', the 'write start' commands alternate between WS0 and WS1 with the successful completion of each write channel program. The 'write start' commands are X'31' for WS0 and X'51' for WS1.

When the channel adapter receives the control command, the CA generates a level 3 interrupt into IOS. When IOS receives the 'write start' (WS) control command, the command determines that the host wants to write to the 3704 or 3705. The WS control command is compared to the expected WS command in the channel adapter control block (COB or CHB at offset X'0F'). If the two commands are equal, the expected WS command is flipped and the enqueue count and skip count are reset to zero. Data is transmitted until a complete PIU is received or until an unexpected control command signals an error condition on the channel interface. When a PIU is completely received, the PIU is passed to 'path control out', and the enqueue count is incremented by 1. If an unexpected control command is received, the enqueue count is added to the skip count. As each PIU is received a second time, rather than pass the PIU to 'path control out' again, the skip count is decremented until the count is zero.

The enqueue count and skip count fields are reset to zero by IOS when IOS receives the next WS control command which is equal to the expected WS command in the channel control block.

The host can send multiple PIUs to the NCP with one host write. PIUs are separated logically by using a CCW with command chaining between PIUs.

Host Reads from the NCP

The host 'read channel program' must start with a control command. The control command is either a 'read start zero' (RS0) or a 'read start one' (RS1). At the completion of the read channel program in the host, the RS0 command which must be sent first is changed to a RS1. The RS0 and RS1 commands alternate with the successful completion of each read channel program. The RS control commands are X'32' for RS0 and X'52' for RS1.

The host does not execute the read channel program without first receiving an attention interrupt from the 3704 or 3705 controller. The attention interrupt is the means by which IOS lets the host know that it has data to send across the channel.

Before the data is put on the channel intermediate queue, the data length is checked to ensure that the PIU fits in the host buffers. IOS sets up the channel adapter to present the attention interrupt to the host. The attention interrupt causes the host to execute its read channel program starting with an RS control command.

On receiving the RS control command, IOS compares the received RS command with the expected RS command in the channel control block at a displacement of X'0E'. If the RS command received is the expected command, IOS flips the expected RS command and purges any PIUs from the hold queue. IOS moves as much data as fits in the host buffers from the intermediate queue to the hold queue.
Before each PIU is sent to the host, a number of pad characters is sent to the host as a reserved area for host internal control. The count of pads is coded on the HOST macro BFRPAD operand. Following the pad, if the pad and PIU are less than or equal to the length of one host buffer, the IOS sends a complete PIU (UNITSZ value). IOS never lets the host CCW channel stop, but forces chaining to avoid a channel stop by the channel. If the end of a PIU forces chaining, the second PIU begins in the next host buffer, with leading pads sent before the PIU. If the original PIU had additional data beyond a single host buffer, the data continues into the subsequent host buffer.

When all PIUs in the hold queue have been sent, IOS presents ending status to the host. If more PIUs are available for the host, IOS adds an 'attention' to the status being sent back to the host. This 'attention' status indicates to the host that a new 'read' is needed for the 3704 or 3705 controller. The host responds with a new 'read channel program'.

A second method by which the channel IOS indicates to the host that it has PIUs for the host is to send a status modifier (SM) at the end of the 'write' portion of a write/read combination channel program. The SM tells the host to skip over a NOP that follows the 'write' CCWs and continue with the 'read' CCWs. These methods eliminate the need for excess asynchronous interrupts on the channel. At the end of the read CCWs IOS presents final status of channel end, device end, and unit exception. This facility is specified on the HOST operand of STATMOD=YES.

The following channel programs illustrate the host channel program for a 'read', 'write', and the 'write/read' sequence.

**Read Channel Program**

```
CCW 32 or 52,*,X'60',1
CCW 02,BUF1,X'60',L'BUF1
--
  -- Read Commands
--
CCW 02,BUFn,X'60',L'BUFn
CCW 03,*,0,1 NO-OP
```

**Write/Write Break Channel Program**

```
CCW 31 or 51,*,C'60',1
CCW 01, BUF1,X'60',L'BUF1
--
  -- Write and/or
-- Write break commands
--
CCW 09,BUFm,X'60',L'BUFm'
CCW 03,*,0,1 NO-OP
```

**Write/Write Break and Read Combination Channel Program**

```
CCW 31 or 51,*,X'60',1
CCW 01,BUF1,X'60',L'BUF1
--
  -- Write and/or
-- Write break commands
--
CCW 09,BUFn,X'60',L'BUFn
```
NOTE 1: This NO-OP is not essential for correct operation, although it may be desirable for compatibility when the status modifier option is selected. If the status modifier option is not selected, the 'write break CCW' may be command-chained to the 'read start CCW'. If status modifier is selected, the NO-OP should be included and should not be command-chained to the 'read start CCW'. If compatibility is desired, include the NO-OP in the channel program and turn the command chain flag on and off as needed.

Type 1 and 4 Channel Adapter

The type 1 or 4 channel adapter support requires an interrupt at level 3 for each four bytes transferred. As the number of INBFRS is depleted, the level 3 code branches into the level 4 supervisor routine to obtain a new supply of buffers equal to the INBFRS number. Then the 'read' (host 'write') operation continues to the completion of the host 'write' or another allocation of buffers. Buffers allocated to the channel and not used by the current NCP 'read' are held for a later 'read' operation.

Channel Operation Block (COB) The control block for a type 1 and type 4 channel adapter is the channel operation block (COB). The address pointer to the COB is in the halfword direct addressables (XDH) at X'772'. The control block has a negative displacement to minus X'30'. Some of the key addresses are identified as follows:

- X'20'  Channel intermediate QCB
- X'18'  Channel hold QCB
- X'08'  Constant of XXCXTCOB
X'00'  Channel condition flags
X'0E'  Next expected read start
X'0F'  Next expected write start
X'10'  to X'23'  External register input areas
X'28'  Address of first inbound buffer
X'30'  Pointer to current buffer
X'34'  Current inbound data address
X'38'  Address of first buffer of completed inbound PIU
X'3C'  Address of last buffer of completed inbound PIU
X'40'  Count of PIUs passed to path control
X'42'  Number of PIUs to skip on a retry
X'44'  Maximum data count for current inbound buffer
X'46'  Generation buffer lease count (HOST INBFRS=value)
Type 2 or Type 3 Channel Adapter

The type 2 or type 3 channel adapter uses cycle steal for data transfer operations. The facility requires IN or OUT control words (CW) which are similar to CCWs in the host.

**Host Writes** When the first 'write start zero' is received, IOS leases buffers, builds 'IN' control words (CWs) and sets up the channel adapter to accept data from the channel. The 'IN' control words are executed one at a time, causing the channel adapter to cycle steal the PIUs into the buffers. During the execution of the control words, no program intervention required.

The next level 3 interrupt into IOS is from one of three conditions; a channel stop, zero count override, or an unexpected 'write start' command.

The channel stop condition occurs when the channel adapter receives 'command out' to a 'service in' request. The channel stop condition signals the end of a PIU and causes a level 3 interrupt into IOS. IOS increments the enqueue count, passes the PIU to 'path control out', and sets up the channel adapter to continue receiving data.

A zero count override condition exists when all the control words (CWs) on the channel-in chain (CIC) have been executed and the host still has more data to transfer. At the completion of the last control word in the CIC, the channel adapter causes a level 3 interrupt into IOS. Since the data transfer has not completed for this PIU, IOS must rebuild the CWs in the CIC. IOS leases new buffers, chains them to the previous buffers and rebuilds the CWs. When the CWs are rebuilt, IOS sets up the channel adapter to continue transferring data, using the address of the first CW in the new CIC. This sequence occurs each time a zero count override is reached.

Receiving an unexpected control command is common to all adapter types and was covered earlier.

**Host Reads** When the 'read start' is received, it is compared against the expected 'read start' control command. If the 'read start' is as expected, IOS flips the expected RS command and purges any PIUs in the hold queue (the previous PIUs to the host). IOS builds the 'OUT' control words (CWs) necessary to send each PIU to the host. After building the 'OUT' control words for a PIU, including a buffer pad (HOST BFRPAD=value), IOS moves the PIU to the channel hold queue. When all of the data (or enough of the data to fill the host buffers) has been moved to the hold queue, the CWs on the channel-out chain (COC) are executed and the PIUs are sent to the host.
When all of the CWs on the channel-out chain (COC) have been executed, the channel adapter generates a level 3 interrupt into IOS. IOS presents ending status to the channel adapter for the host. If more PIUs are available for the host, IOS adds 'attention' to the status for the host. This 'attention' status indicates that a new start I/O is needed from the host.

**Channel Control Block (CHB)** The control block for a type 2 or type 3 channel adapter is the channel control block (CHB). The address pointer to the CHB is in the halfword direct addressables (XDH) at X'772'. The control block has a negative displacement to minus X'30'. Some of the key addresses are identified as follows:

- X'30'  PIU exception QCB
- X'20'  Channel intermediate pointers
- X'18'  Channel hold pointers
- X'08'  Constant of XXCXTCHB
- X'00'  Channel condition flags
- X'02'  Channel adapter select bit (first or second CA)
- X'0E'  Next expected 'read start'
- X'0F'  Next expected 'write start'
- X'10'  to X'25' External register input area
- X'34'  Address of first buffer of current PIU
- X'38'  Pointer to previous inbound buffer
- X'3C'  Address of first buffer on inbound CW chain
- X'40'  Address of last buffer on inbound CW chain
- X'44'  Address of complete PIU to pass to path control out
- X'48'  Address of last buffer of PIU to be enqueued
- X'4C'  Address of inbound CW area
- X'4E'  Address of first CW on inbound chain
- X'50'  Address of last CW on inbound chain
- X'52'  Address of last executed CW
- X'54'  Data count for last inbound buffer
- X'56'  Original data count in last executed CW
- X'58'  Inbound buffer lease count (HOST INBFRS=value)
- X'59'  Current buffer lease count
- X'5A'  Number of PIUs enqueued (enqueue count)
- X'5C'  Number of PIUs to skip for retry (retry count)
- X'60'  Address of last outbound block
- X'68'  Address of first CW on outbound chain
- X'6A'  Address of last CW on outbound chain
X'6C' Host read buffer size (HOST UNITSZ=value)
X'70' Host read CCW count (HOST MAXBFRU=count)
X'74' Buffer pad size (HOST BFRPAD=value)
X'76' Attention delay interval (HOST DELAY=value)
X'78' Channel timeout (HOST TIMEOUT=value)

Channel Words (CW) Channel words are coded in a four field format. The four fields in sequence specify:

1. CW type (IN, OUT, or OUT STOP)
2. Chaining or no chaining.
3. Quantity of data to be read or written
4. Address of data area

When IN CWs are built, all of the CWs are coded IN with the chaining bit ‘on’ in all CWs except the last. When the last CW completes without chaining, (1) a level 3 interrupt occurs to lease more buffers, (2) the last CW is chained to the new buffer chain, and (3) the CWs are rebuilt to point to the new buffers. The first CW points X'0E' offset into the buffer to reserve space for the event control block (ECB) and buffer prefix. All subsequent buffers are generated with an offset address of X'04' into the buffer to bypass the buffer prefix. The length field specifies the true buffer size less X'04'.

When a PIU is received, the host forces ‘channel end’, ‘device end’, ‘without unit exception’ by using a CCW with command chaining. This channel status stops channel transfer and generates a level 3 interrupt. The PIU is passed to ‘path control out’. The next available CW is modified for an offset address of X'0E' into the buffer, the count is modified to the remaining buffer length, and the channel is restarted. All of the delay is transparent to the host.

OUT channel command words are used with chaining until the last CW of a PIU is transmitted or until a host CCW is filled. The next data byte sent to the last-plus-1 position of a host CCW causes the channel to halt transfer on a zero count override to access the next CCW. This channel halt is avoided by forcing the next CCW access without letting the zero count be recognized. If the end of a PIU is reached, the next PIU must start, with pads, in a new host buffer. Both of these conditions use the OUT STOP command to send channel end, device end, without unit exception. Chaining is used for both OUT and OUT STOP until the last OUT STOP CW.

The first CW of each PIU sends pad characters. The second CW addresses the PIU at an offset of X'0E', following the event control block (ECB).

Figure 4.1 illustrates the NCP-to-host transfer using OUT and OUT STOP CWs. The NCP buffer size is 48 bytes (without the buffer prefix) and the host buffer size is 124 bytes. The first PIU is 154 bytes, the second PIU 63 bytes. In each PIU the pad is sent to the host for BFRPAD length (15 bytes in the example). The first NCP buffer of each PIU has a 10-byte ECB which is not sent to the host; only 38 bytes are transmitted from a first NCP buffer.
Figure 4.1. Data Transfer from NCP to Host Buffer Units

4.8 NCP Release 5 Data Flow
Type 1 or 4 Versus Type 2 or 3 Channel Adapter

There are many factors which affect channel performance, one factor being the type of channel adapter. The type 1 or type 4 channel adapter requires many additional communications controller cycles to execute the level 3 code after every four bytes of transfer. More commands are also processed in the channel, tying up the channel for greater periods of time than is the case with the type 2 or 3 channel adapter. If the controller is not heavily loaded, machine cycles are available for servicing the type 1 or 4 adapter, rather than having the controller in the wait state.

Host and NCP Buffer Sizes

Host and NCP buffer sizes should not need to be identical. For one thing, the NCP buffer has a 10-byte event control block (ECB) for control fields in the first buffer of a PIU, and this size does not match the host prefix requirements. OS VTAM requires a HOST macro BFRPAD of 28; DOS VTAM requires 15. The size of buffers should be related to the average size of the PIU in order to avoid unused space in large buffers for small PIUs and avoid excessive buffer chaining and unchaining of small buffers for large PIUs. Remember that CICS, IMS, TCAM, control commands, and probably many user applications have a response of 13 bytes, and even control commands are short. The minimum NCP buffer size of 44 (BUILD BFRS operand) should be sufficient for responses. The maximum size of 248 for NCP buffers or the default of 60 may be excessive if data requests are short. The host size should be determined as the same size as NCP, plus the difference between the NCP control requirement (ECB ten bytes) and the host buffer pad requirements as specified on the BFRPAD operand.

In addition to the size of PIUs as a factor in NCP buffer size, there is a critical factor of SDLC terminal buffer size specification. An operand of MAXDATA on the PU macro specifies the maximum PIU which can be sent to the device. There is an absolute requirement that the NCP buffer size must be at least TH less than the smallest MAXDATA value. This NCP buffer size should never be a problem unless the MAXDATA is coded in error. Type 1 physical units have a 261-byte physical buffer (five-byte FID3 TH/RH plus 256 bytes of text), and type 2 physical units have a 265-byte physical buffer for receiving PIUs (nine-byte FID2 TH/RH plus 256 bytes of text). The largest NCP buffer size is 248, six bytes less than the requirement for MAXDATA.

If PIUs are larger than the MAXDATA operand, PIUs are segmented. A segment is a TH-plus-1 or a multiple of full NCP buffers. Segmenting affects the NCP buffer size. Segmenting is covered later under the topic boundary network node.

VS1 VTAM requires the host buffer size to be an odd multiple of words. The HOST macro UNITSZ operand should not be divisible by 8 for VS1 VTAM.

Host Buffer Allocation

The NCP defines the number of host buffers on the HOST macro operand of MAXBFRU. The number multiplied by the host buffer size minus buffer pads (MAXBFRU x UNITSZ - BFRPAD) restricts the size of the largest PIU which can be sent to the host. There is no restric-
tion on the size of a PIU from the host to NCP. If channel IOS determines that a PIU exceeds the length of the host capacity, IOS sets an error response in the PIU and returns the PIU to the source.

Another consideration is the number of PIUs sent to the host by a single host read. If DELAY is coded as nonzero on the HOST macro, a timer is set when the first PIU is enqueued to the intermediate queue. 'Attention' is not sent until the timer expires, allowing additional PIUs to be enqueued, or until the number of PIUs fills the number of host buffers, whichever condition occurs first. If the host completes a write to the NCP and STATMOD=YES is coded on the HOST macro, any PIUs in the intermediate queue are sent before the timer event.

The delay technique has two benefits: (1) improvement in the host performance by reducing the number of 'attentions' and host buffer allocations; (2) improvement in NCP performance by reducing the number of channel initializations and termination processing of the intermediate queue to hold queue. When traffic is light, the PIU is delayed at the channel. When traffic is heavy, the delay is not used because the amount of data queued fills the host buffer allocation.

**NCP Buffer Allocation** The INBFRS operand defines the number of buffers to be allocated for host-to-NCP transfers. When the last allocated buffer is filled, the NCP obtains more buffers as required. If a large number of NCP buffers is allocated to the channel and not used promptly, it deprives other users of free buffers and may result in slowdown. If few NCP buffers are allocated, the NCP must lease buffers more frequently, taking required controller cycles.

**Delay** See Host Buffer Allocation.

**Status Modifier** The STATMOD=YES operand of the HOST macro allows the NCP to send PIUs to the host at the completion of a host 'write'. When a host 'write' completes, rather than send the 'attention' separately or as a part of the write status and waiting for a host 'read', the PIUs can be sent as a continuation of the host 'write CCW' chain. If the NCP has traffic for the host, the status modifier causes the host 'write CCWs' to chain to 'read CCWs'.

**Channel Timeout** If the HOST macro operand is coded TIMEOUT=NONE, the NCP sends 'attention' and waits indefinitely for the host to reply. If auto network shutdown support is included (BUILD ANS=YES), the operator can initiate auto network shutdown from the panel of the communications controller.

If the host does not reply to the 'attention', a timeout value provides automatic entry to auto network shutdown. All current pending line operations complete, resources are deactivated, and a 'network shutdown complete' message is placed on the channel queue.
There are four types of channel adapters. Types 1 and 4 require heavy program support, but are required for emulation programming. Types 1 and 4 have common code. Types 2 and 3 also have common code. Type 2 is used for single processors; type 3 allows a dual interface to tightly coupled multi-processors. Types 2 and 3 are high-performance, cycle-steal channel adapters. User definition of host and NCP buffer parameters and other channel-related operands on the HOST macro can have significant effect on performance.

Use appendix C to answer the following questions. Do not refer to the answers in Appendix B until you have completed all the questions.

1. What is the address of the channel control block?
2. Is the channel control block a COB or CHB?
3. Are any buffers on the channel queues?
4. How many buffers (INBFRS) are allocated for data from the host?
5. How many buffers (MAXBFRU) are allocated for data going to the host?
6. What is the pad size (BFRPAD) on PIUs going to the host?
7. What is the size (UNITSZ) of a host buffer?

Criterion
If you missed more than one question, you should review this material.
Path Control

Objective

Upon completion of this topic, the student should be able to identify the control blocks used by 'path control out', 'path control out delayed', 'path control in immediate', 'path control in delayed', and describe the flow of data in the modules.

Path Control Out

'Path control out' directs the flow of path information units (PIUs) from the channel adapter IOS to the proper destination. 'Path control out' uses the destination address field (DAF) from the PIU to access entries in the subarea index table (SIT), subarea vector table (SVT), and the resource vector table (RVT). The 'path control out' routine locates the appropriate path for the PIU and places the PIU on a queue control block (QCB) for processing by NCP physical services, boundary network node (CUB or LUB), link scheduler (SCB), or the BSC/SS processor. The 'path control out' module operates in program level 3 via a branch from the channel IOS. When a complete PIU is received from the host, the channel code branches to 'path control out'.

'Path control out' determines where the PIU is to be queued.

When the PIU destination is a type 4 physical unit, 'path control out' enqueues the PIU directly on the station control block (SCB) link outbound queue (LOB). From the link outbound queue the PIU is transmitted to the remote by the link scheduler.

When the PIU destination is a type 1 or type 2 physical unit, the PIU is enqueued on the common unit physical block (CUB) or logical unit block (LUB), depending upon the PIU destination.

A PIU that is destined for NCP physical services is placed on the NCP physical services block (PSB) process queue.

If the PIU is destined for a BSC or SS device, the PIU is passed to the BSC/SS router, which converts the PIU to a BTU and enqueues the BTU on a device block (DVB) or a nondevice input queue.

'Path control out' routes PIUs to their proper destination. To accomplish this routing, 'path control out' uses several tables (created during NCP generation) in conjunction with the DAF portion of the PIU. These tables are the subarea index table (SIT), subarea vector table (SVT), and resource vector table (RVT).

Subarea Index Table (SIT)

The subarea index table (SIT) consists of one-byte entries that correspond to the network subarea addresses. The content of each one-byte entry is a value used to index into the SVT. The NCP generation builds the SIT according to the MAXSUBA and SUBAREA operands of the BUILD macro and the SUBAREA operand of the type 4 PU macro.

The MAXSUBA operand of the BUILD macro determines the size of the SIT. If MAXSUBA is coded 15, there are MAXSUBA entries-plus-1, or 16 entries. The first entry is always one byte of zero. The remaining entries are filled according to the definitions of the network control program.
If the host is defined as subarea 1, the SIT table offset of 1 provides the one-byte offset into the SVT, which contains the address of the channel control block (COB or CHB). The second subarea index table entry always contains a X'02' to offset to the third entry in the SVT. All other entries are dependent upon the SUBAREA operands on the BUILD and type 4 PU macros.

If the BUILD macro SUBAREA=2 and two type 4 PU macros are coded SUBAREA=4 and SUBAREA=6, the third, the fifth, and seventh SIT entries provide offsets to the SVT table. The BUILD entry always contains X'01' to point to the second SVT entry. Each type 4 PU macro generates consecutive entries in the SVT; therefore, the SIT values for type 4 PU macros are third, fourth, etc., in the relative subarea position. Figure 5.1 illustrates the SIT.

```
SUB AREA ADDRESS
   1 2 3 4 5 6 7   F
  00 02 01 00 03 00 04 00 00

BUILD MAXSUBA=15, SUBAREA=2
PU PUTYPE=4, SUBAREA=4
PU PUTYPE=4, SUBAREA=6
```

Figure 5.1. Subarea Index Table

**Subarea Vector Table (SVT)** The subarea vector table is made up of four-byte entries. Each entry consists of a type field, which describes the type of subarea this entry represents, and the address of the table or control block representing that subarea. The NCP generation builds the SVT according to the TYPGEN operand of the BUILD macro and the number of type 4 PU macros included in the generation.

The type field identifies the entry as the address of the resource vector table (RVT), channel control block (COB-30 or CHB-30), or a type 4 PU (SCB) of a remote link. The first entry is a value of zero, and all SIT entries with undefined SUBAREA index to this entry. The second entry is always the address of the resource vector table (RVT). In a channel-attached controller, the third entry is the address of the channel control block (COB or CHB) minus 30. If TYPGEN=NCP or PEP, the next entry is a delimiter entry with X'FF' in the type field. If TYPGEN=NCP-LR or PEP-LR, each type 4 PU (SCB) generates an entry between the channel entry and the delimiter.

Figure 5.2 illustrates the SVT.
The resource vector table (RVT) is made up of four-byte entries. Each entry consists of a type field and the address of the control block represented by this entry. The NCP generation builds the RVT, with an entry for BSC/SS definitions of LINE, CLUSTER, TERMINAL, and COMP macros, and SDLC definitions of LINE, PU, and LU macros. If switched SDLC links are defined, the last entries are addresses of logical units in the logical unit pool. These addresses are generated by the LUPOOL macro.

The RVT is divided into two sections. The first section is for BSC/SS entries, and the second is for SDLC entries. Both sections have a delimiter entry with a type field of X'FF'. At RVT-4 is a two-byte field which contains the highest network element count in the table. The RVT-2 is a two-byte field which contains the highest BSC/SS network element count.
The first entry in the RVT has a type field of X'00' and the address of physical services control block (PSB). If there are BSC/SS devices, they begin in the second position and are delimited by a X'FF'. SDLC devices follow the BSC/SS delimiter entry, or if no BSC/SS devices are included, the SDLC devices follow the PSB entry. The format of the RVT is illustrated on Figure 5.3.

The RVT is located by an address pointer in the word direct addressables (XDA) at X'07E8' which points at the RVT-2. The SVT immediately precedes the RVT and the first entry contains an address of zero. The SIT immediately precedes the SVT. The length of the SIT is determined by the subarea mask at X'693' in the byte direct addressables (XDB).

**Path Control Out Flow** 'Path control out' receives control from the channel adapter IOS. The DAF of the FIDO or FID1 is used by path control to route the PIU properly. The first byte of the DAF contains the subarea address. The byte is shifted as required to delete any leftmost bytes of element address, leaving the true subarea value. This subarea address is used to vector into the SIT to the entry for that subarea. The one-byte SIT entry contains an index value to be used with the SVT. This value is used by path control to index into the SVT to the corresponding entry. The SVT entry contains flags describing the entry and a pointer to the control block representing that subarea.

The possible subarea entries in the SVT and their associated pointers are as follows:

- Invalid subarea (entry of zeros)
- Local NCP subarea (pointer to the RVT)
- Host subarea (pointer to the CHB or COB)
- Remote subarea (pointer to SCB)

The action taken by 'path control out' differs for the various subareas. If the PIU is destined for a type 4 physical unit, the PIU is enqueued on the station control block (SCB) link outbound queue.

If the PIU is for physical services, the element address is zero and the PIU is routed to physical services.

PIUs for type 1 or type 2 physical units are processed by a connection point manager out (CPM-OUT). The CPM-OUT is invoked by enqueuing the PIU on an appropriate CUB or LUB queue. The CPM-OUT branches to 'path control out delayed' for conversion of PIUs from FID1 to FID2 or FID3, segmenting as required, and enqueuing to a link outbound (LOB) queue.

If the RVT entry is in the BSC/SS section of the RVT, the PIU is routed to the BSC/SS system router via a branch instruction.
Path Control

Path Control In

'Path control in' is divided into two parts: 'immediate' and 'delayed'. When a PIU is received on a link, 'path control in immediate' is invoked by a branch from the link scheduler. 'Path control in immediate' checks for a PIU source of a remote controller (SCB). If from a remote, the PIU is immediately queued on the channel intermediate queue for the host. If from a type 1 or type 2 physical unit, the PIU is queued on the CUB inbound queue, and 'path control in immediate' exits from level 3.

'Path control in delayed' dequeues the PIU from the CUB inbound queue and determine which connection point manager in (CPM-IN) should process this PIU. 'Path control in delayed' (at level 5) branches to the appropriate CPM-IN.

Path Control Summary

All PIUs from the host are passed from level 3 channel adapter IOS to level 3 'path control out'. The PIU is validated and if the destination address subarea field is for a remote controller (SVT entry), the PIU is immediately placed on the SCB link outbound queue of the remote controller. All PIUs destined for this controller or for devices connected to this controller are checked against the resource vector table to locate the appropriate queue. After the boundary network node code has processed the PIU, it is passed to 'path control out delayed' for FID conversion, segmenting, and enqueuing to a link outbound queue.

All PIUs from the link scheduler are processed by 'path control in immediate'. The PIU is validated. If the source is a remote controller, the PIU is XPORTed to the channel intermediate queue. If the source is a type 1 or type 2 physical unit, the PIU is enqueued on the CUB link inbound queue, triggering path control in delayed. 'Path control in delayed' determines the session of the PIU and branches to the appropriate boundary network node connection point manager in (CPM=IN).

Additional information on 'path control in' is given later under physical unit-to-host PIU processing.
Path Control Quiz

Do not refer to the solution in Appendix B until you have finished the quiz. Appendix C provides the storage listing for use in answering the following questions.

1. What is the address of the subarea index table?
2. What is the address of the subarea vector table?
3. What is the address of the resource vector table?
4. How many bits of the 16-bit network address are used to identify the subarea?
5. What is the highest valid element defined in this NCP?
6. What is the subarea address of this NCP?
7. What is the path for a contact command for a type 4 PU?
8. What is the path for an 'activate physical' command for a type 4 PU?

Criterion If you missed more than one question, you should review this section.
Network Control Program Physical Services

Objective Upon completion of this topic, the student should be able to identify the session hierarchy, physical services components, control blocks, and flow of data in physical services modules.

Purpose of NCP Physical Services The NCP physical services component is a collection of routines necessary for the control and/or modification of the communications network. NCP physical services are divided into two functional areas: (1) system control and (2) function management. The required services are selected via request codes in the PIU.

Session Hierarchy The requirement for physical services is based upon the session control of the network and the need to change network status. Before data can be transferred through the communication network, a physical and logical connection must be established between the origin and destination of the data request. This connection is referred to as a session. There are four types of sessions that are controlled by network commands. Figure 6.1 illustrates the four session types.

Figure 6.1. Session Hierarchy

SSCP and NCP Physical Services This session is initiated with an ‘activate physical’ command to NCP physical services from SSCP and is ended with a ‘deactivate physical’ command. The next command required is ‘start data traffic’ which enables data flow within a session. Data sent to physical services consists of requests to change the network status. Before any other sessions can be initiated, the links must be activated and physical units contacted.

An ‘activate link’ session control request is required to activate a link. The ‘activate link’ request causes the link scheduler to be initiated for this link.
Bit 1 of LKBSTAT (X'12') in the link control block (LKB) is set to 1 to indicate that an 'activate link' is in progress. For nonswitched links only, the modem is enabled. The LKBSTAT bit 0 is set to 1 to indicate an active link.

---

**Figure 6.2. SSCP to NCP Physical Services Command Sequence**

Switched links require an 'answer' or 'dial' command, and other switched commands which are covered later under switched support. A 'contact' command is required to contact a physical unit. Figure 6.2 illustrates the request sequence of a contact command. The contact request is acknowledged by physical services with a response to SSCP. The contact request also schedules a 'set normal response mode' (SNRM) SDLC command to the physical unit by setting the SNRM bit in the CUB plus X'1F'. On a timeout after an SNRM, the SNRM is retried on a user-specified basis. If a 'nonsequenced acknowledgement' (NSA) response is returned by the physical unit, a 'receive ready' (RR) SDLC command is sent to the physical unit, and a 'contacted' PIU is generated by NCP physical services and sent to SSCP. The link is marked active. The common physical unit block (CUB) CUBSSCF (X'1E') bit 2 (not operational bit) is turned off to indicate that the device is available for sessions to be established. Figure 6.3 illustrates the SSCP-PU and SSCP-LU activation sequence.
SSCP and PU Physical Services  The SSCP/PU session is established with an ‘activate physical’ command addressed to the common physical unit block (CUB) or station control block (SCB) defined by a PU macro. The session is ended by a ‘deactivate physical’ command. The SSCP/PU session must exist before any sessions can be established with logical units. The ‘activate link’ to the link and ‘contact’ command to the device must complete successfully before this session can be established. Type 2 and type 4 physical units receive and respond to the ‘activate physical’ command. The NCP processes this command for type 1 physical units (SDLC 3270 and 3767).

SSCP and LU  The SSCP/LU session is initiated with an ‘activate logical’ command addressed to the logical unit block (LUB) defined by a LU macro. The session is ended by a deactivate logical command. This session must exist before a APPL/LU (host application/logical unit) session can be established. This command is processed by type 1, type 2, and type 4 physical units, except for the SDLC 3270. The NCP performs the processing and issues all responses for all commands addressed to the SDLC 3270.

Host Application and LU  The APPL/LU (host application/logical unit) session is initiated with a ‘bind’ command addressed to the LUB. A ‘start data traffic’ command is required by some types of logical units before data flow can occur. The session is ended by an ‘unbind’. Figure 6.4 illustrates the APPL/LU activation sequence.
Network Control Program Physical Services

Figure 6.4. Bind and Start Data Traffic Commands

BSC and SS note:
Before data can be transferred, sessions must also be established between the host access and BSC/SS devices. These sessions are initiated and terminated within the NCP support for BSC/SS devices via BTU commands. This session level is covered separately in the BSC/SS Processor topic.

Physical Services Block (PSB)
The physical services block (PSB) contains the process queue control block for NCP physical services. The PSB also contains the network addresses of NCP physical services and the host 'system services control point' (SSCP). Other fields of interest are:

- X'00' NCP physical services process QCB
- X'24' Network address of NCP physical services
- X'26' Network address of SSCP
- X'28' Active link count
- X'2A' NCP physical services status
- X'3C' Auto network shutdown extension

Physical Services Components
The NCP physical services component interfaces with the 'system services control point' (SSCP) in the host to provide control functions for the NCP. Some of the functions provided on the basis of requests addressed to the NCP physical services by the host SSCP are:

- Activating and deactivating NCP physical services
- Activating and deactivating links
- Dial
• Answer
• Loading and dumping remote controllers
• Activating and deactivating nodes attached to this controller

NCP physical services is made up of three sections: connection point manager out (CPM-OUT), connection point manager in (CPM-IN), and function management (FM) router. NCP physical services also calls the system control router when necessary. The system control router is common to NCP physical services and NCP boundary network node physical services.

Physical Services Connection Point Manager Out (CPM-OUT)
Physical Services CPM-OUT receives a PIU addressed to NCP physical services. The PIU is validated and, according to the contents of the request/response header (RH) byte 0, CPM-OUT calls either the system control router or the function management router.

Physical Services Connection Point Manager In (CPM-IN)
CPM-IN validates a PIU and XPORTS it to the channel adapter IOS for the host SSCP. All physical services requests and responses are directed to the host SSCP, bypassing path control. When link commands (dial, answer, contact, etc.) complete, NCP CPM-IN is triggered to change status fields and build a PIU to inform the host.

System Control Router
The system control router receives control for a system control category PIU (from either NCP physical services CPM-OUT or boundary network node physical services). The PIU request unit (RU) request code is resolved and through a table lookup routine, the appropriate processor for that request code is given control. The values of bits in the RH and RU determine whether session control or function management gets control. The following identifies the commands and modules for the given RH/RU values:

\[
RH \text{ byte } 0 \ x11xxxx
\]

RU Byte 0

0D  Activate logical CSDBSIL
0E  Deactivate logical CSDBSTL
11  Activate physical (BNN) CSDBSIP
11  Activate physical (NCP) CSDBAPH
12  Deactivate physical (BNN) CSDBSTP
12  Deactivate physical (NCP) CSDBDPH
31  Bind CSDBSIA
32  Unbind CSDBSTA
A0  Start data traffic CSDBSDF
A1  Clear
A2  Set and test sequence numbers
A3  Request recovery
There are data commands addressed to the system router which have an RH byte 0 value of x01xxxx. The commands are:

\[ RH \text{ byte } 0 \ x01xxxx \]

RU byte 0 Command

07 Auto network shutdown complete
50 Initialization complete
51 Switch line to NCP mode (BSC/SS)
52 Switch line to EP mode (BSC/SS)

**Function Management (FM)** The function management router validates FM requests, selects a table of processors according to the RVT type field and, by using a table lookup routine, selects the appropriate processor according to the PIU RU request code. If the PIU RH byte 0 has a value of x00xxxx, the function manager is given control. The PIU RU byte 1 value determines which of four FM categories is used. The PIU RU byte 2 contains the request code. Some of the valid codes are as follows:

\[ RH \text{ Byte } 0 \ x10xxxx \]

RU byte 0 Command

04 Logical unit status
05 Ready to receive
80 Quiesce at end of chain
81 Quiesce complete
82 Release quiesce
83 Cancel
84 Chase
C0 Shutdown
C1 Shutdown complete
C2 Request shutdown
C8 Bid
C9 Signal

\[ RH \text{ Byte } 0 \ x00xxxx, \ RU \text{ Byte } 1 \ X'00' \]

RU Byte 2 Command

01 Change device transmission limit (BSC/SS)
02 Change line negative poll response (BSC/SS)
03 Change line session limit (BSC/SS)
04 Change line service seeking pause (BSC/SS)

RH Byte 0 x00xxxxx, RU Byte 1 X'02'

01 Contact
02 Discontact
03 Load initial
04 Load data
05 Load final
06 Dump initial
07 Dump data
08 Dump final
09 Remote power off

0A Activate link
0B Deactivate link
0E Dial
0F Abandon connection

11 Set state vector
   RU byte 5:
   01 Time and date
   02 Remote NCP/link association
   03 Set control vector/switched PU
   04 Set control vector/switched LU
   05 Set control vector/channel delay

14 Entering slowdown
15 Exiting slowdown
16 Answer
17 Abandon answer mode
18 Abandon dial
19 Assign network address
1A Free network addresses
80 Contacted
81 Inoperative
84 Off hook

RH Byte 0 x00xxxxx, RU Byte 1 X'03'

01 Execute test
02 Activate line trace
Physical services CPM-OUT receives control via an enqueue macro with the ACTV=YES operand. This macro is issued by 'path control out'. This queueing occurs when 'path control out' receives a PIU with a DAF destined for NCP physical services. The PIU is enqueued on the physical services outbound queue in the physical services block (PSB). The task entry pointer for the PSB QCB points to the NCP physical services CPM-OUT.

CPM-OUT gets the contents of the PIU RU byte 0. If not a X'11' request code ('activate physical'), CPM-OUT verifies the PIU OAF by comparing it to the network address in the PSB at offset X'26'.

Physical services CPM-OUT uses bits 1 and 2 of the PIU RH byte 0 to determine the type of request. Both bits 'off' signifies a function management request. If the PIU is a system control request, the system control router is called.

Function management performs more verification on a request by checking the sequence number of the PIU against the PSB offset of X'20'. CPM-OUT assumes that the request following the 'activate physical' from the SSCP to physical services must have a sequence number of 1 in its transmission header. Each subsequent function management request is expected to have a sequence number one greater than the previous request. The PSB is checked for 'session established' and 'data flow enabled' at PSB offset X'2A', testing for a value of 11xx xxxx. If all of the above tests are met, the function management router is called.

The system control router and the function management router both use a table lookup routine in conjunction with the PIU request code to select a processor. There are significant differences between the two routers.

The system control router first uses the DAF from the TH and the UIBITYPE byte of the PIU to set an indicator showing the destination type for this PIU. The indicators are as follows:

- X'80' Request is for NCP physical services
- X'00' Request is for BNN physical services
- X'40' Request is for a BNN logical unit

The indicator is used as the second byte of a two-byte table search argument. The request code from the RU1RCO byte of the PIU is used as the first byte of the search argument.
The search argument is compared to the first two bytes of each entry of the system control router table (SCRT). When a match is found, the routine pointed to in that entry is given control. X’FFFC’ indicates the end of the SCRT.

The function management router activates links, contacts physical units, and performs similar services.

Function management requests are divided into four subcategories. The type of subcategory is determined by the contents or the RU1BT1 byte of the PIU as follows:

- X’00’ BSC/SS service request
- X’02’ Physical configuration services request
- X’03’ Physical maintenance request
- X’06’ Session services request

Once the function management router determines which subcategory the requests are for, the RVTETYPE bytes within the RVT are used to select the proper table within that subcategory. An example of this table selection is the physical configuration subcategory which contains three tables:

1. Link configuration table
2. NCP configuration table
3. Station configuration table

Finally, the function management router uses the request code in the RU1RC2 byte of the PIU as a search argument for the selected table. When a match is found, the routine pointed to in that entry is given control. The function management router tables are delimited by a X’80’.

The physical services control block (PSB) can be located by the first entry in the resource vector table (RVT). The RVT-minus-2 address can be found at X’7E8’ in the word direct addressables (XDA). The following PSB fields are of special interest:

- X’00’ Shifted address of first element queued
- X’02’ Shifted address of last element queued
- X’04’ Task and queue status
- X’08’ CPM-OUT task address
- X’10’ Shifted address of first element queued
- X’12’ Shifted address of last element queued
- X’14’ Task and queue status
- X’18’ CPM-IN task address
- X’20’ Inbound sequence number
- X’22’ Outbound sequence number
- X’24’ Network address of NCP physical services
Network Control Program Physical Services

Physical services provides services for system control requests and function management requests. The initialization of NCP, activation of lines, initial contact of devices, etc., all are performed by physical services. Host control requests are sent to physical services in the PIU RU with the command type, command, and resource address of the element to be affected by the command.

Use the storage dump listing in Appendix C to answer the following problems.

Do not refer to the answers in Appendix B until you have answered all questions.

1. What is the address of the PSB?
2. What is the NCP load module name?
3. Has an ‘activate physical’ command been processed?
4. Has a ‘start data traffic’ command been processed?
5. How many links are active?
6. What is the next sequence number PIU expected from SSCP to physical services?
7. Are any PIUs queued for processing by physical services?
8. The buffer pool is formatted in Appendix C. Analyse the following buffers to determine PIU origin, destination, command, and element affected by the command:
   X'19CA8' through X'19E24'
   X'1A628' through X'1A7A4'
   X'1A8D4'
9. The buffer pool is formatted in appendix C. Analyse the following buffers to determine PIU origin, destination, command, and element affected by the command:
   X'1A038' and X'1A168'

Criterion
If you miss more than two questions, you should review this material.
Boundary Network Node (BNN)

**Objective**
Upon completion of this topic, the student should be able to identify the boundary network node components, control blocks, and flow of data in physical services modules.

**Function of the Boundary Network Node**
The NCP boundary network node (BNN) is the interface between the host SSCP and the link scheduler. The BNN processes PIUs containing control requests and data associated with sessions between:

- SSCP and the physical units (SSCP/PU)
- SSCP and the logical unit (SSCP/LU)
- Host application and the logical unit (APPL/LU).

The two major elements of BNN are ‘connection point manager’ (CPM) and ‘path control in’.

The boundary network node (BNN) of NCP can be divided into two sections. The first section consists of PIUs travelling to physical unit (PU) or logical unit (LU) on an SDLC link. The second section are PIUs travelling from a physical unit (PU) or logical unit (LU).

There are three distinct paths through the BNN for PIUs travelling in either direction. These paths relate to the session which can be established with PUs or LUs. The possible sessions are: SSCP/PU, SSCP/LU, and APPL/LU.

**BNN Control Blocks**
The BNN processes FID1, FID2, and FID3 PIUs. The formats of the PIU were covered in the Network Control Program Supervisor section. References are made to NCP physical services control block (PSB) (covered in the previous section on physical services) and the link control block (LKB), which is checked for link status (covered in detail in the link scheduler section). The two new control blocks used primarily by the BNN code are the common physical unit block (CUB) and the logical unit block (LUB).

**Common Physical Unit Block (CUB)**
The common physical unit block (CUB) is generated by a PU macro. The CUB represents the physical device for SDLC control and queuing of inbound and outbound PIUs for this physical and logical unit group. The CUB provides a link inbound queue control block at CUB offset X'00' to X'0F' for queuing of all inbound PIUs from the device and dispatching of ‘path control in delayed’. PIUs addressed to the CUB are queued on the QCB at CUB offset X'3C' to X'4B' for processing by SSCP/PU connection point manager out. After processing is complete and the PIUs for the physical or logical units are ready to be sent to the device, the PIUs are placed on the link outbound queue at CUB offset X'10' to X'13'. The key fields of the CUB are:

- X'00'  Shifted address of first element on the link inbound queue (all FID1 and FID2 PIUs from the device, including logical units)
- X'02'  Shifted address of last element on the link inbound queue
- X'08'  Address of task 'path control in delayed'
### Boundary Network Node (BNN)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'10'</td>
<td>Shifted address of first element on link outbound queue (all FID1 and FID2 PIUs to the device, including logical units)</td>
</tr>
<tr>
<td>X'12'</td>
<td>Shifted address of last element on link outbound queue</td>
</tr>
<tr>
<td>X'14'</td>
<td>Shifted address of first element on link outstanding queue (a maximum of seven PIUs sent on link but not acknowledged)</td>
</tr>
<tr>
<td>X'16'</td>
<td>Shifted address of last element on link outstanding queue</td>
</tr>
<tr>
<td>X'18'</td>
<td>Address of link control block (LKB)</td>
</tr>
<tr>
<td>X'1C'</td>
<td>Network address of CUB</td>
</tr>
<tr>
<td>X'1E'</td>
<td>Service-seeking and contact poll status</td>
</tr>
<tr>
<td>X'22'</td>
<td>Transmission counter</td>
</tr>
<tr>
<td>X'24'</td>
<td>Specification of CUB</td>
</tr>
<tr>
<td>X'24'</td>
<td>Address of physical services</td>
</tr>
<tr>
<td>X'3C'</td>
<td>Shifted address of first element on SSCP/PU queue</td>
</tr>
<tr>
<td>X'3E'</td>
<td>Shifted address of last element on SSCP/PU queue</td>
</tr>
<tr>
<td>X'44'</td>
<td>SSCP/PU CPM-OUT task address</td>
</tr>
<tr>
<td>X'4C'</td>
<td>Device status</td>
</tr>
<tr>
<td>X'4E'</td>
<td>Segment size in NCP buffers</td>
</tr>
<tr>
<td>X'50'</td>
<td>Segment size in bytes (maximum)</td>
</tr>
</tbody>
</table>

**Switched Extension**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'54'</td>
<td>Maximum LUVT entries</td>
</tr>
<tr>
<td>X'54'</td>
<td>LUVT address (last 18 bits)</td>
</tr>
</tbody>
</table>

**Logical Unit Control Block (LUB)** The logical unit control block (LUB) is generated by the LU or LUPOOL macros. LU macros must immediately follow the PU macro they are associated with and must be in LOCADDR operand sequence. The network addresses of the logical units are consecutively numbered from the physical unit. This addressing scheme is used in converting PIUs to or from the different FID formats. There are two queues in the LUB, one for SSCP/LU sessions and one for APPL/LU sessions. The SSCP/LU queue is at LU X'00' to X'0F'. The APPL/LU queue is at LU X'10' to X'1F'. The key fields of the LU are:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'00'</td>
<td>Shifted address of first element on the SSCP/LU queue</td>
</tr>
<tr>
<td>X'02'</td>
<td>Shifted address of last element on the SSCP/LU queue</td>
</tr>
<tr>
<td>X'08'</td>
<td>SSCP/LU CPM-OUT task address</td>
</tr>
<tr>
<td>X'10'</td>
<td>Shifted address of first element on the APPL/LU queue</td>
</tr>
<tr>
<td>X'12'</td>
<td>Shifted address of last element on the APPL/LU queue</td>
</tr>
<tr>
<td>X'18'</td>
<td>APPL/LU CPM-OUT task address</td>
</tr>
<tr>
<td>X'20'</td>
<td>Address of common physical unit block (CUB)</td>
</tr>
<tr>
<td>X'24'</td>
<td>Network address of this LU</td>
</tr>
</tbody>
</table>
Host to PU PIU Processing

BNN Queues for PIUs from the Host PIUs travelling from the host to the link scheduler are received by channel adapter IOS, passed to 'path control out', and enqueued to a processing queue. If a PIU is for a PU from SSCP, the PIU is enqueued on the SSCP/PU queue within the common physical unit block (CUB) CPQ1ECB at offset X'3C'. If the PIU is for an LU from SSCP, the PIU is enqueued on the SSCP/LU queue of the logical unit control block (LUB) LUL1ECB at offset X'00'. If the PIU is for an LU from an application program, the PIU is enqueued on the APPL/LU queue of the logical unit control block (LUB) LUA1ECB at offset X'10'.

Each of the PIUs for the three types of sessions is enqueued on an input QCB which has a task entry point of a 'connection point manager out' (CPM-OUT). Each type of session has a separate CPM-OUT because the processing is different for each type of session. The task pointer in the SSCP/PU processing QCB for the CUB points to the SSCP/PU CPM-OUT. The task pointer in the APPL/LU processing QCB for the LUB points to the APPL/LU CPM-OUT. The ENQUE macro issued in 'path control out' includes the ACTV=YES operand which causes the associated task to be triggered. When the task is dispatched, the appropriate CPM-OUT has control.

Connection Point Manager Out (CPM-OUT) The three types of CPM-OUT processors perform similar functions but are different enough to be covered individually.

SSCP/PU CPM-OUT

The PIU is validated as a FID1 format. Only a FID1 format is valid for the host-to-BNN routines.

The PIU origin address field (OAF) is compared to the network address of the SSCP, which is stored in the physical services block (PSB) by the 'activate physical' command from SSCP to NCP physical services. Only the SSCP in session with the NCP can create this SSCP/PU session or communicate over this path.

The CUB cannot accept any SSCP/PU commands unless the PU is operational. This operational status occurs by means of a command directed from SSCP to NCP physical services function management of a contact command. The contact command schedules a 'set normal response mode' (SNRM) SDLC command to the device. A 'nonsequenced acknowledgement' (NSA) reply indicates that the command was received by the device. Then a 'receive ready' (RR) SDLC command is sent to the device. A 'receive ready' (RR) response indicates that the physical unit is ready for session initiation. Bit 2
of CUBSSCF at offset X'1E' of the common physical unit block (CUB) is set to zero to indicate that the CUB is operational.

If the PIU is a control request, with an x11x xxxx in byte 0 (RH1BO) of the RH, the system control router is called. This is the same system control router which is used by NCP physical services. If the control command in byte 0 of the RU is X'11' (activate physical'), CPM-OUT checks for a session established at bit 0 of X'4C' (CUBPSTAT) in the CUB. If a session is already established, the request is rejected and returned to SSCP. If a session is not established, bit 1 of byte X'4C' CUBPSTAT is turned on to indicate that a session initiation request is being processed. CPM-OUT branches to 'path control out delayed' to convert the FID1 to a FID2 and enqueue the PIU for transmission to the CUB link outbound queue at CUB plus X'10' (CUBLOBH). If the physical unit is a type 1, the 'activate physical' command is processed by the NCP and not transmitted to the physical device. The response is created in the NCP for reply to the SSCP.

If the device is an SDLC 3270, all commands are processed by the NCP, and all replies on behalf of the 3270 are created by NCP and sent to the host.

If the PIU is not a control request (RH byte 0 value of x00x xxxx), the CUB is checked at offset X'4C' for bit 0 value of 1 to confirm that a session has been established. If a session has been established, CPM-OUT branches to BNN 'path control out delayed' to convert the FID1 to a FID2 and enqueue the PIU for transmission to the CUB link outbound queue at CUB plus X'10' (CUBLOBH).

**SSCP/LU CPM-OUT**

The SSCP/LU CPM-OUT processing performs the following functions:

The PIU is validated as a FID1 format. Only a FID1 format is valid for the host-to-BNN routines.

The PIU origin address field (OAF) is compared against the network address of the SSCP, which is stored in the physical services block (PSB) by the 'activate physical' command from the SSCP to NCP physical services. Only the SSCP in session with the NCP can create this SSCP/LU session or communicate over this path.

The SSCP/LU session cannot exist unless the SSCP/PU session is established. The CUB is checked for a 1-bit in bit 0 of X'4C' (CUBPSTAT), indicating an active SSCP/PU.

If the PIU is a control request with an x11 xxxx in byte 0 of the RH, the system control router is called. This is the same system control router which is used by NCP physical services. If the control command in byte 0 of the RU is X'0D' (activate logical'), the LUB is checked for an existing session at LUB plus X'28' (LUBCPSET) indicated by a 1 in bit 0. If no session exists, bit 3 in X'28' (LUBCPSET) is set to 1 to indicate that an 'activate logical' command is being processed. CPM-OUT branches to BNN 'path control out delayed' to convert the FID1 to a FID2 (or to a FID3 if the CUB is a type 1 physical unit) and to enqueue the PIU for transmission to the CUB link outbound queue at CUB plus X'10' (CUBLOBH).
If the PIU is not a control request (RH byte 0 value of x00xxxx), the LUB is checked at offset X'28' (LUBCPSET) for a bit 0 value of 1 to confirm that a session has been established. If a session has been established, CPM-OUT branches to BNN 'path control out delayed' to convert the FID1 to a FID2 (or to a FID3) and enqueue the PIU for transmission to the CUB link outbound queue at CUB plus X'10' (CUBLOBH).

**APPL/LU CPM-OUT**

The APPL/LU CPM-OUT processing performs the following functions:

The PIU is validated as a FID1 format. Only a FID1 format is valid for the host-to-BNN routines.

The APPL/LU CPM-OUT processor checks to verify that an 'activate logical' command established an SSCP/LU session by testing at LU plus X'28' (LUBCPSET) bit 0 for a value of 1.

If the PIU is a control request with an x11x xxxx in byte 0 of the RH, the system control router is called. This is the same system control router which is used by the NCP physical services. If the control command in byte 0 of the RU is X'31' ('bind') the LUB is checked for an active session bit 0 value of 1 in LUB plus X'2C' (LUBAPSET). If no 'bind' command has established a session, bit 3 of byte X'2C' of the LUB is set to 1 to indicate that a 'bind' is being processed. CPM-OUT branches to 'path control out delayed' to convert the FID1 to a FID2 (or FID3) and to enqueue the PIU for transmission to the CUB link outbound queue at CUB plus X'10' (CUBLOBH).

If the PIU is not a control request (RH byte 0 value of x00xxxx), the LUB is checked at offset X'2C' (LUBAPSET) for a bit 0 value of 1 to confirm that a session has been established.

**Pacing from APPL/LU CPM-OUT to LU**

Pacing or the lack of pacing can have a significant effect on the performance of the network. There are two key areas where pacing can be defined for the network: (1) a PIU can be paced between the host and APPL/LU CPM-OUT, and (2) from APPL/LU CPM-OUT and a logical unit. Pacing is always on an APPL/LU basis. The PACING operand provides control of PIU flow between the NCP and the logical unit. VPACING provides control of PIU flow between a VTAM host application and APPL/LU CPM-OUT on a logical unit basis. PACING is covered first because VPACING does not work unless PACING is also used.

If pacing is not defined for each logical unit, the PIUs are processed and placed on the link outbound queue as they are received by the NCP. The link scheduler dequeues and transmits PIUs to the logical units. In a physical unit there is a fixed number of physical buffers per logical unit. If the physical unit buffers are filled, subsequent PIUs transmitted by the link scheduler are rejected for lack of buffers. The link scheduler retransmits until PIUs are accepted or an error threshold is reach. This retransmission not only adds overhead by executing the link scheduler but also uses line capacity, thereby degrading line capacity. The physical unit buffers may be tied up by the PIUs for one logical unit, while the remaining logical units are waiting for PIUs because of a lack of physical unit buffers.
Pacing is defined by two operands of N and M. An operand of PACING=(N,M) specifies that N PIUs are to be sent to the logical unit before waiting for a pacing response. The M value defines which of the N PIUs carries the request for a pacing response.

There are five fields used for pacing control. The PIU has one bit for pacing control. If bit 7 of RH1B1 is 1 in a request between NCP and the logical unit, the request is for a pacing response by the M PIU. The pace bit 1 in a response between the logical unit and the NCP identifies a reply to a request for a pacing response, indicating that the logical unit is available for the next PIU. The other four fields are in the logical unit control block (LUB) in the following fields:

- X'2D'  LUBASSET bit 1, waiting (1) or not waiting (0) for a pacing response
- X'2E'  LUBM M pacing parameter
- X'2F'  LUBN N pacing parameter
- X'30'  LUBPC pacing count

If the LUBN field has a 0 value, pacing is not defined. If pacing is not defined, CPM-OUT processes PIUs as they arrive and places them on the link outbound queue. If LUBN has a non0 value, the following pacing processing occurs:

1. Pacing between the NCP and logical unit by APPL/LU CPM-OUT first checks LUBASSET for a ‘waiting for a pacing response’. If a wait is indicated, the PIU remains in the APPL/LU queue on the LUB and CPM-OUT EXITS. CPM-OUT is triggered again by CPM-IN when a pacing response is received.

2. If step 1 did not suspend processing, the LUBPC pacing counter is incremented by 1 and compared to LUBM. If LUBPC is equal to LUBM, the current PIU carries the pacing request. The RH1B1 bit 7 is set to 1. The PIU is queued to the CUB link outbound queue for transmission.

3. The pacing counter, LUBPC, is compared to LUBN. If the fields are not equal, the next PIU is processed at step 1. If the LUBPC and LUBN fields are equal, the LUBASSET ‘waiting for pacing response’ bit is set on.

Processing of PIUs loops through steps 1, 2, and 3 for each PIU until the LUBPC counter equals LUBN limit. An equal condition turns on the ‘waiting for pacing response’.

4. When a pacing response is received from the logical unit by CPM-IN, the following processing occurs:
   - The LUBASSET ‘waiting for pacing response’ is turned off.
   - The pace bit (pace response) in the PIU is turned off.
   - The LUBPC counter is reset to 0. If a pacing response returns before the LUBN limit is reached, the ‘waiting for pacing response’ bit is not 1, so a new pacing sequence is initiated by resetting the pacing counter. The pacing counter is reset when
PACING=(2,1), only one PIU is available for transmission, and the response arrives before the N limit is reached.

- The PIU is checked for response status. If the PIU which carried the pacing request did not request an FME or RRN response, the PIU was sent to the NCP merely as a pacing response. The buffer is returned to the free buffer pool. If the PIU which carried the pacing request also required a response (FME or RRM), the response to the host continues.

- The APPL/LU CPM-OUT queue is searched for an enqueued PIU. If a PIU is on the queue, CPM-OUT is triggered to begin the next pacing sequence.

**Pacing from SSCP to APPL/LU CPM-OUT**

The VTAM VPACING parameter can control PIU traffic on a per-logical-unit basis between the host and the NCP. The VPACING operand is required to avoid NCP buffer depletion by an unlimited number of PIUs sent to the NCP from the host. The only alternate to VPACING is for each application program to send a limited number of PIUs before waiting for an FME response. VPACING limits are controlled by SSCP, which eliminates the concern for controls in each host application.

One important consideration for VPACING is the requirement for concurrent PACING. The following VPACING logic should make clear that VPACING has no effect unless PIUs are held on the APPL/LU queue by the PACING scheduling.

VPACING is defined by two operands of N and M, the same as PACING. The VPACING=(N,M) specifies N PIUs are to be sent to the NCP before waiting for a pacing response. The M value specifies which of the N PIUs carries the request for a pacing response.

The M, N, count, and bit indicating 'waiting for VPACING response' from NCP are in the SSCP. Our concern is the manner in which NCP looks for a pacing request from SSCP and how NCP sends a pacing response. VPACING uses the same pacing bit in the RHIB1 of the PIU, which is used for PACING between the NCP and logical unit. VPACING also uses the logical unit control block (LUB) field of LUBASSET (X'2D'). Bit 3 of LUBASSET is used to indicate that a pace response is required by SSCP.

VPACING processing is easily incorporated into the previous processing example for PACING. Only two processing points are added by VPACING.

When a PIU is enqueued on the APPL/LU, the CPM-OUT is triggered as specified in PACING step 1. If the APPL/LU CPM-OUT is waiting for a pacing response from the logical unit, the new PIU is placed on the APPL/LU queue without any processing. If the APPL/LU CPM-OUT is not waiting on a logical unit pacing response, the first PIU on the queue is dequeued for the following processing:

1. If the pace bit in RHIB1 of the PIU is 1, the PIU is checked to determine if a response (FME or RRN) to the host is required by this PIU. If no response is required, a response PIU is created with the pace bit of 1, and the response PIU is XPORTed to the host. If a response is
required, the 'pace required by host' bit is set in LUBASSET for future use by the response. The pace bit in RH1B1 is turned off (0), VPACING outbound processing is complete, and step 1 of PACING can begin.

VPACING inbound processing occurs after step 4 of PACING is complete.

2. The LUBASSET field is checked for 'pace required by host'. If the bit is 1, it is changed to 0 and the pace bit in RH1B1 of the PIU response is set to 1 to be sent to the host.

VPACING logic depends upon a delay on the APPL/LU CPM-OUT queue for PACING. If PACING is not specified, CPM-OUT immediately processes PIUs, sends a pace response to the host, and queues the PIU on the link outbound queue.

The transfer of data in a session requires that a consecutive sequence number be maintained for PIU requests. A PIU from the host application to an LU contains a PIU sequence number field at TH1SNF (X'14'). The type 2 physical unit performs its own sequence checking. Type 1 physical unit number generation inbound and sequence checking outbound is performed by the NCP, using the type 1 LUB extension at LUB plus X'32'.

Control requests are asynchronous and are not sequence-numbered or sequence-checked. Only one asynchronous control command can be outstanding in a session.

**BNN Path Control Out Delayed** ‘Path control out delayed’ is common to all BNN session types. ‘Path control out delayed’ converts the FID1 PIU to FID2 or FID3 PIU. For APPL/LU sessions, the PIU is segmented if the length exceeds the physical unit line-buffer size. Finally, ‘path control out delayed’ issues an XIO LINK which causes the PIU to be enqueued on the common physical unit block (CUB) link outbound queue.

**FID1 to FID2 Conversion**

When the PIU is received by ‘path control out delayed’, the PIU is checked to ensure it is a valid FID1 PIU. Conversion does not change the request/response header (RH), request/response unit(RU) or text. The only change is to the transmission header (TH). The TH1DCF count field at offset X'16' and one byte from the OAF and DAF fields are deleted.

The TH1SNF sequence number field at offset X'14' is moved to X'16'.

Both the destination address field (DAF) and the origin address field (OAF) are two-byte fields in a FID1 PIU. The FID2 format provides only a single byte for each of these fields. The PIU has reached the destination point of the network address by being queued to the specific control block which defines the physical destination point. The full network address is no longer required. The origin and destination addresses need identify only the device local address and determine that the session is an SSCP or application.

If the FID1 origin address field (OAF) is from SSCP, the FID2 OAF field is set to a value of X'00'. If the PIU is from an application program, the field is set to X'01'. The FID2 OAF is at TH20AF at offset X'15' where the original
FID1 sequence number was located. The destination address field (DAF) is obtained from the LUB plus X'31'.

If a FID1 PIU is queued to an LUB, the network address of the CUB is located in the LUB at LUBCUB at offset X'20'. The CUB network address is at the CUB field of CUBRSE at offset X'1C'. A PIU sent to an LUB may be from SSCP (identified by X'00') or from the application program (identified by X'01') in the FID2 OAF field. A valid combination is one of the following:

- DAF X'00', OAF X'00' SSCP to CUB physical services (type 2 PU); type 1 PU commands are processed by NCP and the logical units start at zero)
- DAF X'nonzero', OAF X'00' SSCP to LU
- DAF X'nonzero', OAF X'01' host application to LU

The next conversion moves the FID1 TH1B1 field from offset X'0F' to X'13' unchanged. The TH1B0 from X'0E' is moved to X'12', with bit 3 set to 0 (FID1 indicator) and bit 2 set to 1 (FID2 indicator).

A four-byte gap has been created from X'0E' through X'11'. The buffer offset is incremented by 4 and the buffer data count field is decremented by 4 to adjust for the change. The PIU FID1-to-FID2 conversion is complete. If the PIU is destined for a type 2 physical unit and the PIU is from SSCP (a command), the PIU is placed on the common physical unit block (CUB) link outbound queue by an XIO LINK. The link outbound queue is in the CUB at CUBLOBH (link outbound header) and CUBLOBT (link outbound trailer) at offsets X'10' and X'12'. During normal execution, the link scheduler locates the PIU. Only an APPL/LU PIU requires additional processing. If the physical device is a type 1, the FID2 must be converted to a FID3 before being placed on the link outbound queue.

**FID2 Segmenting**

A PIU from an application differs in processing only if the converted PIU length exceeds the physical buffers of the device, as defined by the MAXDATA operand on a PU macro. If the PIU length is greater, the segmentation routine (CXDBSEG) divides the PIU into segments which are equal to or less than the length of the buffers in the physical device. The routine leases a buffer and copies the TH of the original PIU, setting the TH2B0 first segment, middle segment, last segment indicators as required. Segmenting is based upon the NCP buffer size. A segment length is based upon the data length to the size of one or a multiple of the NCP buffers. A middle or last segment always starts with a leased buffer containing the copied TH and continues with the text from the beginning of an NCP host buffer.

A first segment contains the TH, RH, and a portion of the RU, in multiples of full NCP buffers, the total length of which is less than or equal to MAXDATA size. A nonfirst segment is TH (copied from the first segment into a separate buffer) plus one or more full NCP buffers of less than or equal to MAXDATA size. If PIUs of more than MAXDATA length are used, the NCP buffer size should be selected to provide an efficient segment length. If segmenting is not normal, the segment length should not be a consideration in selecting an NCP buffer size.
Figure 7.1 illustrates a PIU which requires segmenting. The PIU from the host contains 553 bytes (540 bytes of RU). The physical unit definition is coded MAXDATA=265. The NCP buffers are defined as 60 bytes (plus a 4-byte pad). Segment size is in full NCP buffers. The segments sizes are:

- First segment, TH=6, RH=3, and RU=217, from the first four NCP buffers. The RU is made up of 37 bytes of the first buffer and 180 bytes of the second, third, and fourth buffers.

- Middle segment, TH=6 and RU=240. The TH is copied from the first buffer into a leased buffer. The RU is from buffers five, six, seven and eight.

- Last segment, TH=6 and RU=83. The TH is copied from the first buffer into a leased buffer. The RU is from buffers nine and ten.

As each segment is created, it is placed on the link outbound queue of an XIO LINK, just as in the processing of the nonsegmented PIUs. Pacing occurs on complete PIUs, not PIU segments. Keep in mind that the PU physical line buffers may be overrun.

Segmenting may not be supported by a specific terminal type. In addition, you should not confuse segmenting (TH indicated) between an NCP and a terminal with chaining (RH indicated) between a host application and a terminal.

**FID2 to FID3 Conversion**

After the normal processing of 'path control out delayed' is completed (except for placing the PIU on the link outbound queue), a last check is made for a type 1 or a type 2 physical unit. If the bit settings in the common physical
unit block (CUB) indicate that the physical unit is a type 1, the FID2 must be converted to a FID3.

Conversion of the FID2 to a FID3 format affects only the transmission header (TH) fields. Four more bytes in the original buffer are now reserved fields and only two bytes of TH are used. The first byte of FID3 TH contains the FID3 identifier at buffer offset of X'16'. The offset of X'17' contains two bits of information defining the session as follows:

- Bit 0 - 1=to/from application, 0=to/from SSCP
- Bit 1 - 1=to/from logical unit, 0=to/from physical unit

The remaining six bits contain the device local address of the destination of this PIU.

Summary of Host-to-Physical Unit Processing Figure 7.2 illustrates the flow of a PIU through the boundary network node (BNN) for a PIU from the host. There are three paths for BNN outbound processing. A PIU is enqueued to one of three queues for one of three paths through BNN.
1. A PIU from SSCP to a physical unit is enqueued on the CUB processing queue at CUB1ECB (X'3C'). This queuing triggers the SSCP/PU connection point manager out (CPM-OUT). If the PIU is a system control command, the PIU is passed to the system control router for processing, then is returned to SSCP/PU CPM-OUT. Type 1 PU commands are processed by NCP, and SSCP/PU CPM-IN is triggered for responses. A type 2 PU PIU is passed to 'path control out delayed' for conversion to a FID2. The PIU is passed to the link scheduler by placing the PIU on the CUB link outbound queue at CUB68BH (CUB plus X'10').
2. A PIU from SSCP to an LU is enqueued on the LUB session control point (SCP) process queue at LULIECB (offset X'00'). The queueing triggers the SSCP/LU connection point manager out (CPM-OUT). The PIU is passed to the system control router for processing and the PIU is returned to SSCP/PU CPM-OUT. Type 1 PU commands are processed by NCP, and SSCP/LU CPM-IN is triggered for responses. A type 2 PU PIU is passed to 'path control out delayed' for conversion to a FID2. The PIU is passed to 'path control out delayed' for conversion to a FID2 format, then to the link scheduler by being placed on the CUB link outbound queue at CUBLOBH (CUB plus X'10').

3. A PIU from an application to a logical unit is enqueued on the LUB application process queue at LUA1ECB (offset X'10'). This queueing triggers the APPL/LU connection manager out (CPM-OUT). If the PIU is a system control command, the PIU is passed to the system control router for processing. Type 1 PU commands are processed by the NCP, and APPL/LU CPM-IN is triggered for the reply. The type 1 physical unit data is sequenced-checked. Type 1 and type 2 physical unit PIUs are processed for host/NCP VPACING and NCP/LU PACING. The PIU is passed to 'path control out delayed' for conversion to a FID2 format and for segmenting, if it is required. Only APPL/LU data PIUs are segmented.

The CUB is checked for type 1 PU or type 2 PU. If the CUB is a type 1 PU, the PIU is converted to a FID3 format. The PIU is passed to the link scheduler by placing the PIU on the CUB link outbound queue at CUBLOBH (CUB plus X'10').

Figure 7.3 illustrates the processing path of a PIU going from a type 1 or type 2 physical unit to the host. The type 4 physical unit path from 'path control in immediate' is an XPORT to the channel intermediate queue on the COB or CHB.
Boundary Network Node (BNN)

![Diagram of Boundary Network Node Inbound Path Flow]

Figure 7.3 Boundary Network Node Inbound Path Flow

**Path Control In Immediate** When a PIU has been received on a link the link scheduler is initiated in level 3 by a PCI. The link scheduler branches to the level 3 'path control in immediate'. Path control validates the PIU. If received from a remote, the PIU must be a FIDO or FID1, and XPORTed to the channel intermediate queue of the COB or CHB. If received from a type 1 or type 2 PU, the PIU must be a FID2 or FID3 and must be placed on the common physical unit block (CUB) link inbound queue at CUB1ECB (offset X'00'). The ENQUE ACTV=YES triggers 'path control in delayed' in level 5. 'Path control in delayed' is triggered to convert a FID2 or FID3 PIU to FID1 and schedule the correct connection point manager in (CPM-IN).

**Path Control In Delayed** Conversion from FID2 or FID3 to FID1 format occurs within the first NCP buffer of the PIU. When the response to the poll is received, the communications control interrupt program (CICP) leases a buffer and sets up the appropriate offset for the type of device polled. A remote controller sends a FID0 or FID1 which requires an offset of X'0E'. A type 2 physical unit sends a FID2 which requires an offset of X'12'. A type 1 physical unit sends a FID3, which requires an offset of X'16'.

If the PIU received is a FID3, the conversion is to a FID2 and the FID2 is converted to a FID1. The conversion from a FID3 to a FID2 obtains some of the basic information to rebuild the FID2 from the control blocks, as well as from the FID3. The CUB is known, as the device was selected from the service order table for polling.

The FID2 TH2B0 is moved from X'12' to X'0E'; bit 2 is set to 0 (FID2 indicator) and bit 3 is set to 1 (FID1 indicator). TH2B1 is moved from X'13' to X'0F'.

7.14 NCP Release 5 Data Flow
The origin address field (OAF) at TH2OAF must be converted from a one-byte address to the two-byte network address. The specific origin (local device address) is obtained from the OAF. For nonswitched links, the local address is added to the CUB network address (CUB plus X'1C') to develop the OAF network address. The resulting address is verified using the resource vector table (RVT) to locate the LUB, and to verify the CUB address pointer at LUB plus X'20'.

For switched links, the local address is used as a displacement into the logical unit vector table (LUVT). The pointer to the LUVT is at CUB plus X'54'. The resulting address is verified through the RVT as in the nonswitched physical units.

The destination address field (DAF) at TH2DAF (X'14') is converted from a one-byte address to the two-byte network address. The only values are X'00' and X'01'. If the value is X'00', the FID2 is destined to SSCP. The SSCP address can be obtained from the physical services control block (PSB) at PSBADRPC (X'26'). If the value is X'01', the FID2 is destined to the application in session with the LU. The application network address is in the logical unit control block (LUB) at LUBNAPL (X'2A'). The DAF field is stored at TH1DAF (X'10') of the PIU.

The TH2SNF sequence number is moved to the TH1SNF (X'16' to X'14'). The PIU text count was accumulated as the PIU was received in U2TCNT (X'08'). This value is the total PIU length, from which the FID2 TH length is subtracted to calculate the RH/RU count placed in TH1DCF at X'16'. Once the conversion is complete, 'path control in delayed' calls one of three connection point manager in (CPM-IN) routines.

Connection Point Manager In (CPM-IN) There are three connection point manager in (CPM-IN) routines. 'Path control in delayed' determines which of the three CPM-IN routines to call, depending upon the session type (SSCP/PU, SSCP/LU, or APPL/LU).

SSCP/PU CPM-IN

When the SSCP/PU CPM-IN is called, the physical unit is checked for an established or pending session, and the PIU is checked for a request or response status. If the PIU field RH1B0 at X'18' has a value of 0xxx xxxx, the PIU is a request from the PU and the PIU is XPORTed to the channel intermediate queue on the COB or CHB. If the bit has a value of 1, the PIU is a response and must be checked for response indicators. If the RH1B0 at X'18' of the PIU has a value of x1lx xxxx, the RU1B0 at X'1B' contains the request code of 'activate (or deactivate) physical'. The response may be a positive or negative response, based upon RH1B1 bit 3. A response requires that status be set. A positive response to 'activate physical' turns on 'session established' and turns off the 'processing session initiation' bit in the CUB CUBPSTAT byte. A 'deactivate physical' response turns off the 'session established' and 'processing session termination request' bits of the same byte. A negative response requires the bit indicating that a command is in process be set to 0. The response is XPORTed to the channel intermediate queue of the COB or CHB.
SSCP/LU CPM-IN

When the SSCP/LU CPM-IN is called, the logical unit is checked for an established or pending session at LUBCPSET (X'28'), and the PIU is checked for a request or response status. If the PIU field RH1B0 at X'18' has a value of 0xxx xxxx, the PIU is a request from the LU and the PIU is XPORTed to the channel intermediate queue on the COB or CHB. If the bit is a 1, the PIU is a response and must be checked for response indicators. If the RH1B0 at X'18' of the PIU has a value of x11x xxxx, the RU1B0 at X'1B' contains the request code of 'activate' or (deactivate) logical'. A positive response (RH1B1 bit 3 of 0) to an 'activate logical' requires that the 'processing activate' bit turned off and 'session established' bit turned on in LU field LUBCPSET at X'28'. A positive response to a 'deactivate logical' requires that the 'session established' and 'deactivate in progress' bits be turned off in LUBCPSET. A negative response requires the appropriate bit of 'activate (or deactivate) in progress' be turned off. The response is then XPORTed to the channel intermediate queue of the COB or CHB.

APPL/LU CPM-IN

When an APPL/LU CPM-IN is called, the logical unit is checked for an established or pending session at LUBAPSET (X'2C') and the PIU is checked for a request or response status. If the field RH1B0 at X'18' has a value of 0xxx xxxx, the PIU is a request from the LU. The PIU is XPORTed to the channel intermediate queue on the COB or CHB. If the bit is a 1, the PIU is a response and must be checked for response indicators. If the RH1B0 at X'18' of the PIU has a value of x11x xxxx, the RU1B0 at X'1B' contains the request code of 'bind', 'unbind', or 'start data traffic'. A positive response (RH1B1 bit 3 of 0) to a 'bind' requires the 'processing bind' bit turned off and the 'session established' bit turned on in LUBAPSET (X'2C'). A positive response to an 'unbind' requires the 'session established' and 'processing unbind' bits of LUBAPSET turned off. The 'start data traffic' response does not set bits in the LUB but is required by the device to verify that the response to the 'bind' was processed by SSCP.

The PIU is checked at RH1B1 for a 'pace' bit. If the pace bit is 1, the logical unit has responded to a pacing request sent to the logical unit by BNN CPM-OUT. The LUBASSET (X'2D') field of the LUB bit indicating 'awaiting pacing from the LU' is set to 0, the pace bit in the PIU is set to 0, the pacing counter (LUB plus X'30') is reset to 0, and BNN CPM-OUT is triggered to send another PIU to the device. The PIU is checked for FME or RRN response to be sent to the host. If the FME or RRN bits are not 1, the PIU is a 'stand-alone pacing response'; the buffer is returned to the NCP buffer pool, and CPM-IN exits.

If the FME or RRN bits are 1 and the PIU is to be sent to the host, the LUB field of LUBASSET is then checked for a 'pace required by host'. If this bit is 1, the pace bit in the RH1B1 field of the PIU is set to 1 and the LUB 'pace required by host' bit is set to 0.

With all response checking now completed, the response PIU is XPORTed to the channel intermediate queue of the COB or CHB.

Summary of Physical Unit-to-Host Processing Figure 7.3 illustrates the flow of a PIU through the boundary network node (BNN) for a PIU from the physical unit. There are three paths for inbound processing. All three
paths are the same until 'path control in delayed' enqueues the PIU to one of three connection point managers in (CPM-IN). The CPM-IN XPORTs the PIU to the channel intermediate queue of the CHB or COB.

The following sequence is followed for PIUs going to the host:

1. A PIU is passed from the link scheduler to 'path control in immediate' at level 3. 'Path control in immediate' checks to see if the PIU is from a type 1, type 2 or type 4 PU. A PIU from a type 4 physical unit is checked for FID2 or FID1 and XPORTed to the channel intermediate queue of the CHB or COB. A PIU from a type 1 or type 2 physical unit is enqueued to the link inbound queue of the CUB, triggering 'path control in delayed'.

2. 'Path control in delayed' is a dispatched level 5 task triggered by the PIU enqueued from 'path control in immediate'. The PIU is converted from FID2 or FID3 to FID1 and passed to one of the three connection point managers in (CPM-IN), depending upon the session type.

3. Connection Point Manager In (CPM-IN)

The three CPM-IN routines called from 'path control in delayed' are based on one of the three following types of session.

**SSCP/PU CPM-IN**

The SSCP/PU CPM-IN processes control responses to reflect correctly the session status of the SSCP/PU session, and XPORTs the PIU to the channel intermediate queue of the COB or CHB.

**SSCP/LU CPM-IN**

The SSCP/LU CPM-IN processes control responses to reflect correctly the status of the SSCP/LU session, and XPORTs the PIU to the channel intermediate queue of the COB or CHB.

**APPL/LU CPM-IN**

The APPL/LU CPM-IN processes control responses to reflect correctly the status of the APPL/LU session. Data PIU requests are sequence-checked. Data PIU responses are checked for pacing responses from the device. If a pacing response is found, the APPL/LU CPM-OUT is triggered for another PIU to be sent to the device. If the LUB bit indicating a host pacing response is required, CPM-IN sets the pacing bit in this response to the host. The PIU is XPORTed to the channel intermediate queue of the COB or CHB.

**SDLC Switched Support**

The NCP generation of SDLC switched support includes defining a group of lines for dialout, dialin, or dialin/out operations. The macro instructions that define switched SDLC operations are GROUP, LINE, PU, and LUPOOL. The PU macro specifies the number of LUs required during a connection by the operand of MAXLU. When a connection is made, the required LUs are obtained as required from the pool of LUs defined by the LUPOOL macro.

The switched SDLC support generates an extension on the CUB at offset X'54' of four bytes. The leftmost byte provides a count of entries in the logical unit vector table (LUVT). The last 18 bits provide an address pointer to the the LUVT table.
The LUVT table contains a four-byte entry for each logical unit defined for this physical unit (PU MAXLU=count). Each entry contains the following:

- X'00'  Local address of the logical unit
- X'01'  LUVT flags: 1xxx xxxx last entry in table, x1xx xxxx entry in use
- X'00'  LUB address pointer (last 18 bits)

This is the only table which is added for switched SDLC support.

The NCP provides three modes of operation for switched SDLC links:

1. Manual dial. The NCP enables the link and allows the operator to dial out.
2. Autodial. The NCP enables the link and performs the dial operation using the dial digits provided with the command.
3. Answer. The NCP enables the link and allows the remote stations to call in. The link remains in answer mode until the SSCP terminates it. If the SSCP issues a dial command to the link, the answer mode is temporarily suspended until the dialed connection is broken.

Logical unit control blocks (LUBs) are dynamically assigned to logical units when a switched connection is made. For this reason, a number of dummy LUBs must be allocated during NCP generation. Using the 'assign network address' command, the SSCP assigns LUBs from a pool to the physical unit which has a connection. When the SSCP breaks the connection, the SSCP issues a 'free network addresses' command to return the LUBs to the pool.

Additional commands from SSCP to NCP physical services provide the control of switched link support. The function management data indicator (x00x xxxx) in RH byte 0 and 02 in RU byte 1 indicate a request to physical configuration services. The commands for the control of switched links include the following:

- X'0E'  Dial. Causes the NCP to initiate an outbound call on a switched SDLC link. For auto dial, the NCP performs the dial operation with the dial digits provided in the command. For manual dial, the NCP enables the link and the operator performs the dial operation.
- X'0F'  Abandon connection. Causes the physical unit to terminate a switched connection.

RU, byte 5 = X'04'. Changes dynamic fields in the logical unit control block (LUB) and completes initialization of the logical unit vector table (LUVT).

RU, byte 5 = x'03'. Changes dynamic fields in the common physical unit block (CUB) which are associated with the specified physical unit.

RU, byte 5 = x'02'. Associates a remote NCP's subarea with a particular SDLC link.

X'16'  Answer. Causes the NCP to put the specified link in answer mode. Answer mode enables the link to accept incoming calls.
X'17'   Abandon answer mode. Causes the NCP to discontinue answer mode on the specified link.

X'18'   Abandon dial. Causes the NCP to halt the dialing operation over the specified link.

X'19'   Assign network addresses. Assigns a set of network addresses to a specified physical unit (SDLC switched link only).

X'1A'   Free network addresses. Causes the NCP to free the network addresses that were assigned to a physical unit (SDLC switched link only).

X'84'   Off hook. Informs the SSCP that a physical connection has been established between the NCP and a physical unit. The PIU contains the station ID.

Appendix A provides the command sequence required to create the connections and sessions.

Figure 7.4 illustrates the command sequence and SDLC sequences of a switched connection.
The switched SDLC link connection is broken by the following sequence of commands to terminate the connection:

1. The SSCP issues a 'deactivate logical' command for each of the logical units. This command terminates the SSCP/LU session.

2. The SSCP issues a 'free network addresses' command to release the assigned LUBs and return them to the LU pool.

3. A 'deactivate physical' command terminates the session between the SSCP and the physical unit. If the physical unit is a type 1 device, the NCP does not transmit the command to the device, but responds to the 'deactivate physical' command. Type 2 and type 4 devices receive the command and reply.
4. The 'discontact' command causes the NCP to send a 'set disconnect response mode' (SDRM) SDLC command to the station. The station replies with an NSA and the connection is broken.

5. The 'abandon connection' command causes the NCP to disable the link and return it to 'on hook' status. If the link was previously in answer mode, the NCP reenables the link.

Figure 7.5 illustrates the network commands and SDLC sequences for breaking a switched SDLC connection.

Figure 7.5. SNA and SDLC Commands to Terminate Switched Connections

**Boundary Network Node (BNN) Summary**

All PIUs in a session involving an SDLC link, except for a local/remote link, are processed by BNN routines. These routines handle session control requests and responses, and convert PIUs to the required format. Data transfers in an application/logical unit session are processed by pacing routines. On output to a physical unit, the buffer requirements of the physical unit segments PIUs as required. The NCP performs sequence-number processing on PIUs to and from type 1 physical units.

The user definition of pacing is vital to system performance. VPACING schedules PIUs on a logical unit basis between the host and the NCP to avoid buffer depletion. PACING schedules PIUs on a logical unit basis between the NCP and the physical unit to avoid depleting physical unit buffers and having one logical unit lock out other logical units. VPACING logic requires a definition of PACING also. In order to operate correctly, VPACING requires a delay on the CPM-OUT queue created by PACING requests.
Segmenting breaks up PIUs when the length of a PIU exceeds MAXDATA. A first segment is TH, RH, and RU to full NCP buffers of equal to or less than MAXDATA size. A nonfirst segment is TH (copied from the first segment into a separate buffer) plus one full NCP buffer or a multiple of buffers of less than or equal size of MAXDATA. If PIUs of more than MAXDATA length will be received, the NCP buffer size should be selected to provide an efficient segment size.

Figure 7.6 illustrates the flow of PIUs through the NCP for CUB and LUB devices. The numbered text that follows identifies the components and processing:

1. Channel IOS branches to 'path control out'. Using the DAF to access the SIT, SVT, and RVT, path control enqueues the PIU to a CUB or LUB.

2. The enqueuing triggers the BNN CPM-OUT.

3. If the PIU is a session control request, the system control router gets control via a branch.

4. The system control router selects the proper subroutine and returns.
5. BNN CPM-OUT processes the PIU and calls ‘BNN path control out’.

6. The PIU (FID2) is placed on the link outbound queue (LOB) and XIO is issued to the link.

7. The link scheduler locates the PIU on the LOB, then sets up the CCBL2 and ICW.

8. CSP handles the ‘transmit’ or ‘receive’.

9. When level 2 ends, level 2 sets up CCPQON/OFF and issues a PCI to level 3 to return.

10. The ‘command ender’ routine uses CCBL3 to continue level 3 link scheduler processing.

11. The link scheduler branches to ‘BNN path control in immediate’.

12. ‘Path control in immediate’ enqueues the PIU to the link inbound queue on the CUB.

13. ‘Path control in delayed’ selects the proper CPM-IN, using the FID1 or FID2 origin to locate the CUB or LUB queue.

14. The CPM-IN processes the PIU and XPORTs the PIU to the channel queue to be sent to the host.

Use the storage listing in Appendix C to answer the following problems. Do not check the answers in Appendix B until you have answered all of the questions.

1. Which CUBs have a pending SNRM?

2. Which CUBs are in session?

3. What are the SDLC address/polling characters of the CUBs in session?

4. What are the network addresses of the CUBs in session?

5. What are the addresses of the LUB control blocks of the CUB at X'18C1C'?

6. Do the LUBs in question 5 have an SSCP/LU session?

7. Do the LUBs in question 5 have an application/LU session?

8. Locate and write down the pacing values of the LUBs at X'183C0', X'183F4', and X'18428'.

9. What are the local addresses of the LUBs in question 8?

10. Using the formatted buffers in Appendix C, analyze the buffers at the following addresses:

   X'1A7F0'
   X'1A83C'
   X'1A920' through X'1ACFC'
11. Using the formatted buffers in Appendix C, analyze the buffers at the following addresses:

X'1A628'
X'1A674'
X'1B040'
X'1ADE0'
X'1AE2C'
X'1B0D8'
X'1AE78'
X'1B254'
X'1B124'
X'1B170'
X'1B1BC'
X'1B2A0'

Criterion
If you missed more than two questions, you should review this section.
Local/Remote Link

Objective

Upon completion of this topic, the student should be able to identify the local/remote link control block, the flow of data to activate, load, communicate, and closedown a remote NCP, and operands which affect performance.

Activation of a Remote NCP

At generation time, a PU macro is used to define a remote (link-attached) NCP to a local (channel-attached) NCP. The path to a remote requires the same activation as type 1 and type 2 physical units. Once the link is active, 'path control out' XPORT's an outbound PIU to the PU generated station control block (SCB) by queuing the PIU to the SCB link outbound queue. Inbound PIUs are passed from the link scheduler to 'path control in immediate' at level 3. When 'path control in immediate' determines that the PIUs are from a remote to the host, the PIU is XPORTed to the channel intermediate queue of the CHB or COB.

Station Control Block (SCB) PU Type 4

If you compare the station control block (SCB) PU type 4 with the CUB PU type 1 and type 2, the fields are seen to be identical for the length of the type 4 SCB. The type 1 and type 2 CUB has an extension added for a QCB for outbound PIU processing. PIUs with the remote subarea identification are not processed but XPORTed by 'path control out' directly to the SCB link outbound queue at SCBLOBH (X'10'). PIUs for the host are XPORTed directly to the channel intermediate queue of the CHB or COB. PIUs directed to the remote are enqueued to the SCB link inbound queue at SCB offset X'00'. Some of the main SCB fields are as follows:

- X'00' Shifted address of the first element queued
- X'02' Shifted address of the last element queued
- X'08' SSCP/SCB CPM-IN task address
- X'10' Shifted address of the first element on the link outbound queue
- X'12' Shifted address of the last element on the link outbound queue
- X'14' Shifted address of the first element on the link outstanding queue. One to seven PIUs are transmitted on the link, but not acknowledged.
- X'16' Shifted address of the last element on the link outstanding queue
- X'18' SDLC addressing character of remote
- X'18' Address of link control block (LKB)
- X'1C' Network address of resource (local subarea and element address). Remote subarea and element address is in the SVT.
- X'1E' Service seeking and contact poll commands
- X'20' Remote status
- X'22' Transmission counter
- X'24' Address of physical services
**Initializing the Remote**  
PIUs from the host can be directed only to the remote subarea specified in the PU macro SUBAREA operand. A local type 4 PU station control block (SCB) has no queue or task to process a PIU. The format of the type 1 or type 2 CUB of basically the same format has an extension input QCB for queuing PIUs from the host. All PIUs which are for a remote or which refer to a remote are directed to the local NCP physical services or are queued directly to the SCB link outbound queue for processing by the link scheduler.

---

**Figure 8.1 Command Sequence to Activate a Remote**

Before PIUs can be directed to a remote, the link must be activated, contact established, and the remote loaded. Figure 8.1 illustrates the command sequence which is required between SSCP and the local NCP physical services, and between the NCP and the remote.

The sequence of commands from SSCP to physical services is identical to the type 1 or type 2 PU. The response from the remote may be either a 'request initialization' (RQI) or the 'nonsequenced acknowledge' (NSA) of the type 1 or type 2 PU. The contacted command from physical services in response to the RQI informs SSCP that the remote requires loading; the NSA specifies a warm start capability.
In response to an RQI, the SSCP obtains the remote load module and sends the load initial, load data, load final, and a second contact command. Physical services schedules a ‘set initialization mode’ (SIM) command and receives a ‘nonsequenced acknowledge. (NSA). The ‘load initial’, ‘load data’, and ‘load final’ are transmitted to the remote. Physical services acknowledges receipt of the ‘contact’ by sending a ‘set normal response mode’ (SNRM) to the remote and a FME response to the SSCP, to acknowledge that the ‘contact’ command was received. Now that the remote is operational, it can reply to the SNRM with an NSA. The NSA response results in a contacted command being sent to the SSCP.

All of the responses from the remote are processed by the SSCP/PU connection point manager in, with a task address at SCBTSKEP (SCB plus X’08’). All responses from a remote directed to the host are XPORTed directly to the channel queue. Some replies and status are enqueued by link control to the SCB.

Now that the remote is loaded, the same SSCP and application sessions are established in the remote as are established in the local. An SSCP/PSB ‘activate physical’, ‘start data traffic’, ‘set state vector’, ‘activate link’, and other session command sequences must be established between the SSCP and remote elements.

**Host to Remote PIU Processing**

PIUs from the host are received by channel IOS and passed to ‘path control out’. ‘Path control out’ at level 3 validates the FID0 or FID1, verifies that the local/remote link is active, and XPORTs the PIU to the SCB link outbound queue (SCB plus X’10). The link scheduler locates and transmits the PIU to the remote.

**Remote to Host PIU Processing**

The PIUs received on a link are passed at level 3 from the link scheduler to ‘path control in immediate’. ‘Path control in immediate’ has a pointer to the control block which provided the poll request. The station type is at the control block address plus X’24’. If the station is a type 1 or type 2 PU, the PIU is enqueued to the BNN connection point manager in (CPM-IN) queue. If the device is a type 4 PU SCB and the PIU is normal link traffic, the PIU is validated as FID0 or FID1 and XPORTed to the channel intermediate queue of the CHB or COB. If the remote had a failure, the SCB connection point manager in (CPM-IN) is triggered for error recovery.

**Remote NCP**

The remote NCP has basically the same facilities as the local NCP. The remote controller does not have a channel adapter and therefore does not have the channel adapter IOS. The link is serviced by the link scheduler and outbound PIUs are passed to ‘path control out’. ‘Path control out’ enqueues the PIUs to physical services, BNN CPM-OUT, or the BSC/SS processor. The same session control sequence is required among SSCP, applications and remote elements as was required in the local.

**Loading a Remote NCP**

The remote 3704 or 3705 controller includes a diskette which contains programs used to test the remote hardware and to load and dump the remote NCP. The diskette is prewritten with the configuration data set (CDS) file. This file must be configurated before the remote controller is used. The CDS defines the link to be monitored for communication and the pointers to the diskette data sets.
Loading and dumping of a remote NCP is performed by the load/dump program that resides on the diskette. This program is loaded into the high 8K of storage when one of the following occurs:

- Power is turned on
- The load pushbutton on the remote console is pressed
- The remote NCP terminates abnormally
- An error occurred during a load or dump
- Host issues a load or dump network command

Before loading the load/dump program into high storage, the NCP checks to see if the high 8K of storage should be saved and written on the disk. Also, checks are made to see if any diagnostics or initial tests are to be executed. Figure 8.2 illustrates the format of the remote disk files.

![Remote Disk Format](image)

Figure 8.2 Remote Disk Format

When the load/dump program is loaded into storage, control is passed to program level 1. External register X'6B' contains IPL flags; general register 6 of program level 3 contains a line address for the load/dump program to monitor. This line must have been defined in the remote configuration data set (CDS) file. A byte in the CDS file entry determines if this line is to be used for loading and dumping of the remote NCP. This check prevents unauthorized loading and dumping of a remote controller.

After the load/dump program is initialized, the program executes in levels 2 and 3 performing link scheduler and SDLC functions. Level 1 is reentered when control is passed to the remote NCP after it is loaded.

If a 'load' is to be performed, after the link is activated and the remote contacted, the host sends PIUs containing the remote version of the NCP to the local NCP. Physical services in the local determine that the PIU is a function management request and call the function manager. The FM router uses the
RU of the PIU to select the remote PIU decoder routine from the appropriate FM table.

The remote PIU decoder (CSDKRPD) determines that the request is a 'load initial'. It sets up the station control block for the remote and sends a 'set initialization mode' (SIM) SDLC command to the remote. The load/dump program in the remote controller responds with the 'nonsequenced acknowledgement' (NSA). The NSA ends the run command in the local (CSDKRNT) and passes control to the SIM terminator (CSDKRST). The SIM terminator checks that an NSA was received and issues an XIO LINK to send the load initial PIU to the remote controller. The 'load data' and 'load final' commands that follow are all processed through the local NCP's physical services to the remote PIU decoder (CSDKRPD), which issues XIO LINK commands and sends them to the remote controller. Figure 8.1 illustrates the sequence of commands for loading a remote NCP.

After the load final PIU is sent, a contact is sent by the host SSCP to the local NCP. The local NCP issues a 'contact poll' to the remote controller (send SNRM). On receiving the SNRM, the remote load/dump program passes control to the remote NCP which has been loaded. The remote NCP responds with an NSA to the local NCP. The local NCP sends a contacted response PIU back to the SSCP indicating that the remote is loaded.

**Dumping a Remote NCP** If a printout of remote storage is to be made, the SSCP sends a dump request to the local NCP physical services, which forwards the 'dump initial', 'dump data', and 'dump final' network commands to the remote controller. Figure 8.3 illustrates the command sequences for a dump.

The processing of the dump commands is similar to the load process using the remote PIU decoder (CSDKRPD) and the remote SIM terminator (CSDKRST). The dump data requests are sent to the remote load/dump program which returns the requested data area. The local NCP returns the dump data PIUs to the SSCP for writing to a host disk dump file. After the 'dump final' command is sent to the remote, a 'discontact' command is sent to the local NCP to stop normal polling of the remote controller.
**Local/Remote Link**

<table>
<thead>
<tr>
<th>HOST</th>
<th>CHANNEL</th>
<th>LOCAL CONTROLLER</th>
<th>LINK</th>
<th>REMOTE CONTROLLER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operator issues</td>
<td>Dump Initial</td>
<td>SIM</td>
<td>Dump Initial</td>
<td>Load/Dump program is loaded</td>
</tr>
<tr>
<td></td>
<td>Dump Initial</td>
<td>Dump Initial</td>
<td>Dump Initial</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Dump Data</td>
<td>Dump Data</td>
<td>Dump Data</td>
<td>Returns storage size, storage keys and Regs.</td>
</tr>
<tr>
<td></td>
<td>Dump Data</td>
<td>Dump Final</td>
<td>Dump Data</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Dump Final</td>
<td>Dump Data</td>
<td>Dump Data</td>
<td>Returns data area requested.</td>
</tr>
<tr>
<td></td>
<td>Discontact</td>
<td>Dump Data</td>
<td>Dump Final</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FME</td>
<td>Stops normal polling of the remote controller</td>
<td>Informs remote dump is complete</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 8.3 Command Sequence to Dump a Remote**

**Link Failure to a Remote** If a load is to be performed due to a permanent link failure, the SSCP activates the alternate link. Once the alternate link has been activated, the load and dump process is the same as described above. Figure 8.4 illustrates the command sequence for a recovery from a link failure and loading of a remote NCP.
Remote Power Off

The SSCP may power off a remote controller by issuing a ‘remote power off’ command to the local NCP physical services. Function management remote PIU decoder (CSDKRPD) sets up the SCB for the remote to send a SIM to the remote NCP. Upon receiving the SIM, the remote NCP link scheduler causes an abend condition to load the load/dump program. Figure 8.5 illustrates the command sequence.
The load/dump program responds to the local NCP with a 'nonsequenced acknowledgment' (NSA), causing the remote SIM terminator in the local to get control. The remote SIM terminator issues an XIO LINK to send the 'remote power off' command to the remote controller. The load/dump program checks for a 'remote power off' command and, finding it, issues an OUT X'79' instruction to power off the controller.

**Local/Remote Link Summary**

The type 4 PU station control block (SCB) represents the remote controller on a link. Control commands required to enable, load, dump, or power off a remote controller are all directed through physical services in the local NCP. Once the remote is active, the sessions must be established for the remote physical services, CUBs and LUBs as for the local. Session commands and data PIUs directed to the remote are XPORTed by 'path control out' in the local to the link outbound queue of the remote in level 3. PIUs from the remote are validated by 'path control in immediate' and XPORTed to the channel intermediate queue on the COB or CHB in level 3.
Local/Remote Link Quiz

Answer the following questions. Do not refer to the answers in Appendix B until you have finished the quiz.

1. What type of physical unit defines a remote/local link?
2. What type of control block is generated for a remote/local link?
3. What are the two possible SDLC responses to an SNRM (other than a timeout)?

Criterion

If you made any errors, you should review this section.
Data Link Control

Upon completion of this topic, the student should be able to identify the control blocks, flow of data and operands which affect performance of link operations.

Data link control (DLC) provides the scheduling and control for link operations. Data link control is made up of three main parts:

1. Communications interrupt control program (CICP)
2. Link scheduler
3. Synchronous data link control (SDLC)

The CICP interfaces with the background tasks and drives the link scheduler. The link scheduler schedules, initiates, and ends all SDLC link operations. The SDLC transfers the data between the data buffers and the hardware scanner.

This topic covers data link control in relation to a full-duplex (FDX) link.

The flow of control from initiation of the link scheduler to termination of the link scheduler involves the components covered in the previous sections. The physical services process the commands to ‘activate link’ and ‘contact’ the physical devices on the line. The ‘activate link’ initiates the link scheduler. As each ‘contact’ completes, the run terminator gets control, but the link scheduler is reinitiated for a new run command. After the contact commands the session initiation commands to physical units must be sent on the link to establish the session with the physical units. The session commands and responses are processed as data by the data link control support.

The first pass through the service order table (SOT) issues a ‘contact poll’ (from a ‘contact’ command) for each physical device. After the first pass the run terminator reissues the XIO RUN, which remains in effect until a permanent error or deactivate link command.

Once the link is operative, the service order table is used to locate a link outbound queue of a CUB or SCB. If no element is queued and the device session is established, the ‘receive’ leg is scheduled with a poll. With the ‘receive’ leg now committed, the send ACB can search for a service order table entry with PIU to send to a station other than to the polled station.

The first link outbound queue (in service order table sequence) sends one to seven PIUs depending on several factors. If there is only one PIU, only one is sent before going to the next SOT entry. In addition, there are two operands on the CUB or SCB which qualify the number of PIUs sent on a link. MAX-OUT specifies that one to seven frames may be sent before an SDLC response is required. PASSLIM specifies the maximum number of frames sent before going to the next entry in the SOT. The type 4 SCB PASSLIM may be set to 254 frames maximum on a full duplex link. On a full duplex link, after each frame the ‘receive’ link is checked for a busy condition. If the ‘receive’ leg is released, a poll is sent between frames to a device other than the one currently being transmitted to.
The LINE macro SERVLIM operand (default of 4) specifies the number of passes through the service order table for polling and addressing before special handling is scheduled. Special handling is a search for a command (contact, discontact, deactivate link, etc.). If one is found, one command is attempted before returning to normal data traffic scheduling.

Any pass through the service order table without a PIU to send or with no incoming traffic from polling causes special handling to be scheduled immediately.

The PAUSE operand defines a time value for one pass through the service order table. If the time value has expired before the end of the service order table is reached, normal processing continues. If the time value has not expired, the link scheduler suspends service on the link until (1) the time expires or (2) a PIU is enqueued to a CUB to be transmitted on this link. If the link scheduler is triggered for sending a PIU, the PIU is sent, but no polling occurs until the time has expired.

Each line is initially disabled for interrupts to level 2. When the line is enabled from level 3, the CCB is primed on each interrupt with the address of the next character service routine at CCB plus X'00' (CCBL2). When the sequence at level 2 is complete, a PCI to level 3 gives control to the routine specified at CCB plus X'4C' (CCBL3). The level 3 processing passes input to 'path control in immediate' and schedules the next poll. Output PIUs are retained on the link outstanding queue until an SDLC response confirms a good transmission; then the buffers are released.

There are several control blocks generated from a LINE macro definition at NCP generation. In addition to the LINE macro, the GROUP macro- and SERVICE macro-generated control blocks are used by data link control. The following control blocks and key fields are of special interest:

**Line Group Table (LGT)** The line group table (LGT) is generated by the GROUP macro. SDLC groups generate a shorter LGT (X'17' bytes) than the BSC/SS groups. Some of the primary fields are as follows:

- X'00' Line group type. An X'8C' value is an SDLC primary station. An X'8E' indicates an SDLC secondary system.
- X'0C' LCD/PCF transmit initial value
- X'0D' LCD/PCF receive initial value
- X'10' Command decode vector table address

**Link Control Block (LKB)** The link control block (LKB) contains fields for scheduling link operations and for maintaining link status information. The LKB is generated for each link from the LINE macro. The resource vector table (RVT) contains a pointer to the LKB. Some of the primary fields are as follows:

- X'00' Shifted address to first element queued.
- X'02' Shifted address to last element queued.
- X'08' Enable terminator task address at generation time. When the link is enabled by XIO LINK, the address is replaced by the address
of the run terminator. This task address is the only SDLC task pointer which changes from the generated address.

X'10' Network address of the link. This network address is used at PIU plus X1E for ‘activate link’ and ‘deactivate link’ commands processed by physical services.

X'12' Status of link. Bit 0 indicates an ‘active link’, bit 1 an ‘activate link in progress’, bit 2 indicates ‘deactivate link in progress’.

X'13' Link type. Specifies this link is leased, switched, one or more type 1 PUs, one or more type 2 PUs, one or more type 4 PUs, and whether the link is primary or secondary.

X'24' Address of adapter control block (ACB).

**Adapter Control Block (ACB)** The adapter control block (ACB) is generated by a LINE macro. The ACB contains line control information and the status of input or output operations for SDLC links. At X'24' the link control block (LKB) has an address pointer to the only ACB for a half-duplex link or to the ‘receive’ ACB for a full-duplex link. At ACB plus X'22' is the address of the ‘transmit’ ACB for a full-duplex line. The ACB can be located from the line vector table (LNVT) using the line address. The ACB contains the link XIO block (LXB) from X'00' to X'23' and the character control block (CCB) from X'24' to X'5C'. The fields are covered under the LXB and CCB which follow.

**Link XIO Block (LXB)** The link XIO block (LXB) is generated as the first X'24' bytes of the adapter control block and contains the status of link operations. Some of the primary fields are as follows:

- X'00' Immediate control command flags
- X'01' I/O commands. The only valid commands are X'8D' (enable), X'30' (run SDLC link), and X'83' (disable), X'8F' (dial).
- X'06' Command ending status and completion code status
- X'0E' Shifted address of first buffer of data received
- X'10' Shifted address of final buffer of data received
- X'14' Pass count
- X'18' Pointer to link control block
- X'1A' Received block size (number of data characters stored)
- X'1C' Pointer to current service order table (SOT) for ‘receive’
- X'1E' ‘Contact poll’ command executed (see X'20')
- X'1E' Duplex link pointer to ‘receive’ leg (‘transmit’ leg ACB only)
- X'20' Pointer to current service order table (SOT) for ‘transmit’.
- X'22' Duplex link pointer to ‘transmit’ leg (‘receive’ leg ACB only)

**Character Control Block (CCB)** The character control block (CCB) is
generated as bytes X'24' through X'5C' of an adapter control block and contains line control operations. Some of the primary fields as offsets from the ACB are as follows:

- **X'24'** Address of current level 2 character service routine
- **X'26'** Pointer to character service state address table
- **X'30'** Line address for type 2 scanner. Bit control block (BCB) address if type 1 scanner.
- **X'34'** Pointer to the line group table (LGT)
- **X'3C'** Address of current data byte being sent or received
- **X'40'** Address of current buffer
- **X'4C'** Address of next level 3 routine to be executed
- **X'5B'** Address expected in response (SDLC address/poll character)

**Line Vector Table (LNVT)** The line vector table (LNVT) is generated from the CSB macro and initialized by the LINE macro. A different format is created for type 1 than for type 2 or 3 scanners.

**Type 1 Line Vector Table (LNVT)**

The type 1 line vector table (LNVT) generates an entry of X'10' bytes for each possible line address for a type 1 scanner. A 3705 generates 64 entries from address X'800' to X'BFF'. A 3704 generates 32 entries starting at X'800'. A line address of X'00' to X'3F' is multiplied by X'10', and added to X'800' to calculate the BCB address. The BCB is used by the level 2 routines in program support for the interface control word (ICW) used by a type 2 scanner. An undefined line address has the rightmost bit set to 1 in the first halfword. If the bit is 0, the first halfword points to the adapter control block (ACB).

**Type 2 or Type 3 Line Vector Table (LNVT)**

The type 2 or type 3 line vector table (LNVT) generates a two-byte entry for each possible line address (maximum=96) for each defined scanner (CSB macro). A single scanner generates 96 halfword entries from X'800' to X'8BF'. Each subsequent CSB macro reserves an additional 96 halfwords. An undefined line address has the rightmost bit set to 1 in a halfword entry. A bit of 0 indicates that the halfword contains the address of the adapter control block (ACB) for this line. The first X'20' entries from X'800' to X'83F' are always invalid because the first scanner has only 64 lines starting at line address X'20'.

If a line address is known for a type 2 or type 3 scanner, the LNVT entry can be calculated by multiplying the line address by 2 and adding X'800'. The LNVT allows the level 2 routines to find the ACB (and CCB) for a line when only the line address is known.

**Bit Control Block (BCB)** The bit control block (BCB) is a X'10' byte control block which provides the same facility for a type 1 scanner as the ICW hardware provides for a type 2 scanner. A BCB is generated for each LINE macro defined for a type 1 scanner, and is placed as a valid entry in the type 1 line vector table (LNVT).
Service Order Table (SOT)  The service order table (SOT) is generated by a SERVICE macro to identify the sequence of service to devices on a line. A pointer in the link XIO control block (LXB) at X'1C' points to the current entry in the table for service. All SDLC links, except the link of a local/remote or SDLC-switched, have a service order table. The table contains the following entries:

- X'00'  Halfword of zero
- X'02'  Maximum number of entries
- X'03'  Number of entries in use
- X'xx'  Four-byte entries with the leftmost 14 bits a negative offset to SOT Header. The rightmost 18 bits are the address of a SCB.
- X'xx'  Last four-byte entry has an offset and address of zero.

Link Scheduler Initiation

At the termination of the 'activate link' process, the enable terminator receives control and issues the XIO LINK macro with the run command stored in the LXBCMAND field of the LXB. The XIO macro causes an SVC level 4 interrupt into the supervisor. The supervisor uses the SVC code to vector into the branch table for a pointer to the CICP at entry CXECMDCO. The CICP passes control to level 3 via a PCI level 3 interrupt. Level 3 is used to eliminate any interference while setting up to start the command.

The CICP running in level 3 checks to see if the link is busy by checking the receive-CCB control field (CCBCTL) phase bits for 00. Finding the link not busy, CICP zeros out the status fields in the LXB. No check is made to see whether command initialization should be delayed. If no delay is required, the receive-CCB is checked for any outstanding status.

Next, the transmit-CCB is checked for any outstanding status. With no outstanding status on either leg, the link's PCF field is set to zero to prevent any level 2 interrupts on this line from changing any fields that will be set up now. CICP vectors into the command decode vector table, using the LGTCMD pointer from the line group table (LGT) and the command from the LXB. The pointer at the vector is loaded into the instruction address register (IAR), causing a direct branch to the link scheduler at entry point CXELNKS1. The entry point is the 'run' command initialization entry into the link scheduler. Here the phase bits (CCBCTL) are set to indicate command active, then a branch is taken to the scheduler to schedule run command activity.

When scheduling run command activity, the link scheduler decides whether to schedule a poll or a data transmission. The first test determines whether the 'transmit' leg of the link is busy. If not, the 'receive' leg is checked to see whether it is busy. With both legs of the link idle, the scheduler branches to the poll subroutine to schedule a poll operation. The poll operation is started by scanning the service order table (SOT) for a station control block (SCB) or common physical unit block (CUB) to poll. The scheduler first checks the service-seeking control flags (SCBSSCP) and the service-seeking output control flags (SCBOCF) to be sure that this entry has not already been polled or that a second level error recovery program is not in progress. If the station
has not been polled and there is no second level error recovery in progress, the scheduler proceeds to poll the station.

Before the actual transmission of the poll frame, a test is made for the station type. Is the station a type 1, type 2 or type 4 physical unit? Type is checked so that the ‘receive’ buffer used to store the response can be set up properly for the type of FID. A branch is then taken to the ‘receive initialization’ subroutine (RCVINIT) to set up the ‘receive’ leg to handle the response from polling. This routine prepares the ‘receive’ leg to monitor for flags, then returns to the scheduler. The scheduler sets up the ‘transmit’ leg to send an ‘RR’ poll command (CCBCFLD). The CCB, if not already transmitting continuous flags, is set up to transmit a flag character. Returning from the transmit initialization subroutine, the scheduler sets the data length field in the CCB (CCBCHAR) to zero for the poll, and exits from the program level.

The SDLC character service program sends a flag byte on the link. The program also prepares the CCBL2 pointer to send the address field from the CCB (CCBADDR) at the next level 2 interrupt. When the complete poll frame has been sent, the scanner is set up to transmit continuous flags until the link scheduler finds the next service order table (SOT) entry that can be polled. The ACB is then queued to the ACB queue and a PCI level 3 interrupt is issued. The PCI level 3 interrupt causes the link scheduler to get control again, via the CCBL3 pointer, to poll the next entry in the service order table (SOT).

The XIO LINK macro is used to put PIUs on the link outbound queue (LOB) to be transmitted down the link. The XIO LINK macro stores the pointer to the PIU in the LOB and checks for an active ‘run’ command. With a ‘run’ command already active, XIO LINK does not have to trigger the link scheduler. During its normal scan, the link scheduler finds the PIU on the LOB and sends it down the link.

Entry to the link scheduler is at CSELNKX for normal scan. Before sending the PIU, the link scheduler checks to see if the ‘receive’ leg is busy as a result of the last poll. If the leg is busy, the link scheduler tries to select a CUB or SCB with data on its LOB. The current service order table (SOT) select pointer from the ‘transmit’ leg (LXBSEL) is used to get the station’s SOT pointer. A test of the station’s output control flags (SCBOCF) is made to see if the station is ready and data is waiting. If the station is not ready, the link scheduler advances to the next SOT select entry. With the station ready the link scheduler branches to the SENDPIU subroutine.

The SENDPIU subroutine first checks the basic link unit (BLU) outstanding count (SCBOCL) defined by the MAXOUT operand and the pass count (CCBPASCT) defined by the PASSLIM operand. If the counts have been exceeded, SENDPIU returns to the calling routine (SELECT). SELECT increments the select pointer and continues to the next SOT entry except for a CUB or SCB MAXOUT condition. A local/remote link SOT pointer is not incremented until the PASSLIM operand value is reached. If the counts have not been exceeded, the PIU is sent.

SENDPIU increments the BLU outstanding count, takes the PIU off the LOB queue and puts it on the link outstanding queue (LOS). Next an ‘I’ format BLU command is built and passed to the XMTINIT subroutine along with the
ending process pointer (CSELNKX). A branch is taken to the XMTINIT subroutine to initiate the transmission. The XMTINIT subroutine stores the ending processor address in the CCBL3 field and the BLU command in the CCBCFLD.

Level 2 interrupts are now disabled and a test is made to see if flags are already being transmitted. If so, XMTINIT loads the CCBL2 field with the address to the SDLC send address routine (CSBDLXZ). Level 2 interrupts are enabled again and a ‘transmit’ time-out is started. XMTINIT now returns to SENDPIU to complete the setup of the CCB for transmission. In the CCB, SENDPIU stores the character count (CCBCHAR), the pointer to the current buffer (CCBSTART), and the pointer to the first data character (CCBDATA). SENDPIU returns to the calling routine (SELECT). SELECT stores the SOT pointer in the LXB (LXBSEL) and EXITs from the program level.

The SDLC character-service routines take over to transmit the PIU on the link. The link scheduler subroutine (XMTINIT) previously setup the CCBL2 pointer to point to the ‘transmit address’ routine (CSBDLXA). When the scanner hardware finishes sending a flag character, a level 2 interrupt is generated to the ‘transmit address’ routine. This routine initializes the BCC field (CCBBCC), than passes the address field (CCBAFLD) and the next CCBL2 pointer (CXBDLXC) to the BCC accumulation routine. The BCC accumulation routine sends the address to the scanner, accumulates the BCC character, stores the CCBL2 pointer passed to it in CCBL2, and then EXITs from the program level.

The next level 2 interrupt is to the ‘transmit control field’ routine (CXBDLXCI). This routine sends the control character from CCBCFLD and tests the character count (CCBCHAR) for zero. If the count is not zero, CCBL2 is set up to transmit data (CXBDLXI). If it is zero, the CCBL2 is set up to transmit the first BCC character (CXBDLXBI). On the next level 2 interrupt, with the CCBL2 pointer set to CXBDLXI, the character-service routine sends the data out on the link. This routine loops until all the data has been sent on the link. With no more data to send, this routine sets up the CCBL2 pointer to transmit the BCC character that has been accumulated in the CCB.

The next level 2 interrupt transmits the rightmost byte of the BCC field (CCBBCC) in the CCB. The CCBL2 is set up to transmit the leftmost byte of the BCC field on the next level 2 interrupt. After the second BCC character is transmitted, CCBL2 is set up (CXBDLXFF) to transmit a flag to end the frame.

On the next level 2 interrupt, a check made for half-duplex to see whether a turnaround is needed on the line. If no turnaround is needed, ‘frame transmitted’ status is stored in the CCB (CCBCMPCD) and the line is set to transmit continuous flags (LCD/PCF=9D). A branch is taken to QACBL3 to queue the ACB to the ACB queue for level 3 processing and a PCI level 3 is set. Returning from QACBL3, the CCBL2 pointer is set to ignore interrupts from the line (CSBDLIDL) before EXITing from the program level. At this point, one PIU has been sent on the link.
The PCI level 3 interrupt gives control back to the link scheduler at entry CXELNKK, via the CCBL3 pointer. The 'transmit leg busy' flag is reset and a check is made to see if the last frame transmitted was an 'I'-format frame. If not, the frame must have been in 'S' or 'NS' format, so execution returns to the scheduler at entry CXELNKSX to continue link activity.

A poll frame has been sent down the link. The correct offset for the type of FID has been stored in the CCB (CCBOFSET), based on the secondary station type, and the 'receive' leg CCBL2 pointer has been set to monitor for flags. The CCBL3 pointer has been set up with the normal read-end processor address (CXELNKR).

When the first flag character is received, the 'monitor for flags' routine sets CCBL2 to receive the address (CXBDLRA). The hardware in the scanner handles any other flags received without causing any level 2 interrupts. The next level 2 interrupt comes with the first nonflags character received. This character should be the address field of the frame. The 'receive address' routine (CXBDLRA) checks the character received to see if it is the address expected. If it is not, then the link is reset to monitor for flags again. If the character is the expected address, the address is stored in the CCB (CCBAFLD). The BCC (CCBBCC) is initialized next and the CCBL2 pointer is set to receive the control field (CXBDLRC). Before EXITing from the program level, the BCC is accumulated for the address field which was received.

The next level 2 interrupt gives control to the control field routine (CXBDLRC). If the character received was a flag, a format error has occurred. The status is set to indicate format exception and the ACB is queued back for level 3 processing. If the ACB cannot be queued back for processing, 'block overrun' status is set and the remainder of the frame is flushed. If the character received is the control field, tests are made for the frame format. If the frame is an 'I' format, a buffer is leased and initialized. The CCBL2 pointer is set to receive data (CXBDLRI) before EXITing from the program level.

The 'receive data' routine accepts characters until a flag is received. The characters are stored in the buffer leased by the control field routine (CXBDLRC). If more buffers are needed, buffers are leased one at a time. When the flag character is received, the pointer to the data buffers is stored in the LXB (LXBDATAP). Next, three checks are made, testing for: (1) correct BCC for this frame, (2) the expected address, and (3) final frame. If this is not the final frame, the ACB is queued back to level 3 to process this frame and the CCB is set up to receive the next frame, starting with the address field (CXBDLRA). If this was the final frame (P/F bit on), the transmitting terminal sets 'poll/final' status and queues the ACB back to level 3 to process the frame.

The return to the link scheduler is via CXELNKR from the CCBL3 pointer. If the frame received is an 'I' format, a branch occurs to PROCIFMT to process the frame. PROCIFMT computes the frame length and sets the data offset and counts in each buffer. The last buffer count is adjusted for the BCC characters that were stored. The total data count is stored in the NCP buffer prefix and a test for station type is made: type 4 or not type 4. As a
result of the text, a branch is taken to the appropriate path control routine to route the PIU to its destination. Before returning to normal scheduling, the N(C) count is updated by 1.

Termination and Restart of an XIO Run Command

The 'run' command is ended by triggering the link process queue in the LKB. The task pointer in the link process queue is for the 'run terminator' task. There are only six valid reasons for ending the run command:

1. Reset immediate ('deactivate link in progress', or 'contact' command)
2. Permanent link error (hardware or XMIT error)
3. Station counters overflow
4. Buffer pool end
5. Valid response or ERPs exhausted on 'contact poll'
6. Unrecoverable station error during poll

When the SDLC character-service routines uses PCI to return to level 3 for processing, the link scheduler checks the link status to see if 'run termination' is required. If 'run termination' is required, (CXELNKSS) the 'stop run' command is set in both of the CCB's (CCBCTL) for a full-duplex link. When both legs of the link become idle, the link scheduler ENDRUN subroutine triggers the run terminator task.

Based on the error status received, the ENDRUN subroutine flushes the LOB and LOS queues. For hardware or transmit error status, the ENDRUN subroutine flushes the LOB and LOS for all stations on the link. All PIUs on the station's LOB and LOS are set with 'path error' status, and put on the link inbound queue for that station. The link inbound queue gets triggered along with the run terminator task. For this type of error, the 'run' command is not reissued.

For other exceptions, only the current stations LOB and LOS queues are flushed. Again, all the PIUs on the current station's LOB and LOS are set with 'path error' and are put on the station's link inbound queue. The link inbound queue is triggered along with the run terminator task, but in this case the run terminator reissues the RUN command when the task finishes its processing.

The run terminator determines the reason for termination and the appropriate routine is called to handle the status. In an example of a permanent link error, the link is set inactive (LKBSTAT), the 'active links count' is decremented by 1, an 'inoperative' request is built and sent to the SSCP, and an MDR record is returned to the host. All the stations on the link are checked for FM data requests and if any are found they are returned to the SSCP with an error indication. All stations are left with 'inoperative' and 'poll skip' flags on.

Data Link Control Review

The link scheduler is initiated for this link by an 'activate link' command addressed to NCP physical services. NCP physical services identifies the link to be activated in the PIU RU1NA field, and an 'enable' is processed for nonswitched links.

The link scheduler has a three-part cycle:
1. SERVLIM data passes are made through the service order table as long as one PIU is sent or received per pass. The first pass without a PIU transfer invokes part 2.

2. The physical units are searched for a contact command to be processed. The search begins with the first physical unit following the last unit serviced for a contact command and ends when a command has been serviced or all physical units have been scanned.

3. If in the last data pass (see point 1), the time specified in the PAUSE operand had not expired, the link scheduler waits until (a) a PIU is enqueued for a PU on this link and then transmits only; or (b) the time expires to begin polling.

The flow of an 'activate link' for an SDLC link is illustrated in Figure 9.1. The numbered items that follow identify the flow of the 'activate link' command and processing that takes place.

---

**Figure 9.1 Activate Link Command Flow**

---

1. At channel stop, IOS passes the PIU to path control via a branch.
2. Using the PIU DAF to access the SIT, SVT and RVT, path control enqueues the PIU to NCP physical services.
3. The PSB task is dispatched (PSB CPM-OUT), which calls the function management router.
4. Using RU1BT1 and RU1RC2 (bytes 1 and 2 of the RU), the 'function management router' selects and calls the 'activate link' processor.
5. 'Activate link' enqueues the PIU to the LKB, sets the LKB task pointer to 'enable terminator', sets 'activate link in progress', and issues 'enable XIO'.

6. Command decoder selects the proper initialization routine for 'enable'.

7. CCBL3 is set for the proper return from level 2, the ICW or BCB is set to 'data terminal ready', and CCBL2 is set to the proper level 2 routine to wait for 'data set ready'.

8. At 'data set ready', level 2 enqueues the ACB on the ACB queue (CCPQON) and issues a PCI to level 3.

9. CCPSAVE contains the address of the command ender, which gives control to the CCBL3 pointer.

10. The LKB is triggered, which schedules the 'enable terminator' task. The enable terminator task changes the LKB task pointer to 'run terminator', sets 'link active', and issues 'run XIO'.

11. NCP physical services CPM-IN sends a response to the channel queue for routing to the host.

12. The command decoder resolves the link scheduler as the initialization routine for 'run XIO'.

13. The link scheduler begins polling and selection for this link until the termination of the run command.

14. Should the 'run' command terminate, the 'run terminator' is dispatched because of the LKB task pointer.

Data Link Control Quiz

Answer all of the following questions before checking the answers in Appendix B.

1. What command invokes the link scheduler for a specific link?

2. What two conditions restrict the number of passes through the service order table for data service?

3. When a level 2 scanner interrupt occurs, where can you find the address of the level 2 routine to be executed?

4. When a level 3 PCI is issued by a level 2 scanner routine, where can you find the address of the level 3 routine to be executed?

5. Where is the identifier of the current physical unit to be serviced by a contact poll?

6. Where can you find the identification of the current 'contact poll' command being executed?

7. What are the addresses of the LKBs of the active links?

8. What are the addresses of the LKBs of the switched links and non-switched links?

9. What are the addresses of the LKBs of the half-duplex links and the full-duplex links?
Data Link Control

Criterion

If you missed more than one question, you should review this section.
BSC/SS Processor

Objective

Upon completion of this topic, the student should be able to identify the main components of the BSC/SS processor, the major control blocks, and flow of data in the modules.

Definition of the BSC/SS Processor

The BSC/SS processor is that part of the NCP that processes requests for BSC/SS resources. Instead of the PIU, the basic unit of work is the basic transmission unit (BTU). Therefore, the BSC/SS processor must convert a FIDO PIU received from the host to a BTU and convert a BTU destined for the host to a FIDO PIU.

Processing within the BSC/SS processor is totally different from SDLC support for SDLC resources in boundary network node support. The routing of information to a BSC/SS resource includes the system router, command processor, work scheduler, I/O line task (including I/O line subtasks), character-service routines, and, if a type 1 scanner is defined, bit-service routines. Also, in addition to using a BTU instead of a PIU, many of the queues and control blocks are different.

This topic presents the data format, control blocks, components, and data flow used in the BSC/SS processor for communicating with BSC/SS resources and BSC/SS supporting routines.

BSC/SS Major Control Blocks

Block Control Unit (BCU) When the BSC/SS router receives a FIDO PIU, the PIU/BTU converter builds the block control unit (BCU). The BCU consists of the first buffer prefix, event control block, workarea, and the basic transmission unit (BTU). The format preceding the BTU is similar to the PIU prefix area. The BTU contains 14 bytes of control information from the FIDO and may contain text. The BCU may be contained in one buffer or in many buffers, depending on the size of the buffers and the amount of text in the BTU.

The major areas of the block control unit (BCU) are:

- X'00' Buffer prefix
- X'04' Event control block
- X'0C' Work area
- X'14' Basic transmission unit

The following are the fields of the basic transmission unit (BTU), which is contained within the block control unit (BCU). Offsets are from the beginning of the NCP buffer.

- X'14' Origin address field (always host resource, FIDO X'12')
- X'16' Destination address field (always nonhost resource, FIDO X'10')
- X'18' Sequence number (FIDO X'14')
- X'1A' System and extended BTU response, (FIDO X'20')
X'1C'  BSC/SS BTU command and modifier, (FIDO X'1C')
X'1E'  Function flags (FIDO X'1E')
X'20'  Text length (FIDO X'16' minus RH length of 3)
X'22'  User data (FIDO X'22')

BTU commands and modifiers are covered later in this topic.

**Resource Vector Table (RVT)** The BSC/SS portion of the RVT contains an entry for each LINE, CLUSTER, TERMINAL, and COMP macro in the BSC/SS portion of the NCP generation. Each LINE macro causes an entry to be built describing the type of entry and containing a pointer to the LCB representing that line. Each TERMINAL, COMP, or BSC/SS CLUSTER macro causes an entry to be built describing the type of entry and containing a pointer to the DVB representing that entry. The entries are built as the macros are encountered in the generation.

The format of the resource vector table (RVT) was described in the section on path control.

**Device Base Control Block (DVB)** The device base control block (DVB) contains an input QCB for the device input queue and a work QCB for the device work queue, as well as all parameters needed to operate a device. One DVB is built at NCP generation time for each CLUSTER, TERMINAL, and COMP macro coded (except for CLUSTER macros coded without the GPOLL operand). The DVB may have one or more external extensions, depending on the type of device and the features of the device represented.

Some of the key fields of the DVB are as follows:

- X'00'  Device work QCB
- X'08'  Device input QCB
- X'18'  Block-handler status and address pointer
- X'1C'  Device resource ID
- X'1E'  Device features
- X'2C'  Service-seeking control block
- X'32'  Polling/addressing extension
- X'36'  Polling extension

There are variable extensions to the DVB, depending upon the options selected when the generation definition is coded. Following are the control block extensions to the DVB (offsets to the extensions, if included, are in the DVB from X'27' to X'2A'). The format and values of the extensions can be found in the *IBM 3704 and 3705 Program Reference Handbook* under 'DVB'):

- BHR  Block handler routine extension
- BUE  Switched backup extension
- CGP  Cluster general poll extension
- CIE  Callin extension
COE  Callout extension
DAE  Device addressing extension

**Line Control Block (LCB)**  At NCP generation time, a line control block (LCB) is built for each BSC/SS line connected to the controller. The LCB contains information required for scheduling line operations. The LCB also has fields for maintaining line significant status information and three queue control blocks: (1) line I/O queue, (2) line work queue, and (3) the suspended sessions queue when the LCB represents a multipoint line. Depending upon the line type, the LCB may have nonswitched point-to-point, multipoint, or switched extension.

Some of the key fields of the line control block (LCB) are:

- X'00'  Line I/O QCB
- X'14'  Line work QCB
- X'24'  Pointer to the adapter control block (ACB)
- X'28'  Pointer to the line type command table (LTCT)
- X'2C'  Pointer to the device (DVB) currently connected over the line
- X'34'  Subtask sequence pointer
- X'36'  LCB features, status, etc.
- X'42'  Resource ID
- X'44'  Multipoint extension and BSC/SS session definitions
- X'44'  Switched extension

**Line Type Command Table (LTCT)**  The LTCT contains the system command table, the offset table, and a collection of subtask sequence tables. The system command table is a table of all valid BTU command/modifier combinations. The line work scheduler finds the position in the system command table corresponding to the command and modifiers specified in the BTU. The corresponding position of the offset table gives the offset to the appropriate entry in the subtask sequence table.

BTU commands and modifiers are covered later.

**Adapter Control Block (ACB)**  At NCP generation, an ACB is built for each line defined in the NCP. A BSC/SS ACB contains an input/output block (IOB) and a character control block (CCB). All ACBs are located in the first 64K of storage.

The ACB fields are as follows:
- X'03'  Retry count for dialout
- X'02'  Address of dialout line for auto call
- X'00'  Input/output block (IOB)
- X'24'  Character control block (CCB)

**Input/output Block (IOB)**  The input/output block (IOB) contains the command and modifier to indicate the I/O operation to be performed. The IOB also contains status fields to indicate the outcome of the operation, and pointers to the beginning point and ending point of data sent or received, if any data is present.
Some of the key fields of the input/output block are:

- X'00'  Flags, I/O command and modifiers
- X'0E'  Pointer to first buffer in the block
- X'10'  Pointer to last buffer in the block
- X'18'  Pointer to the line control block (LCB)
- X'20'  Partitioned emulation (PEP) flags

**Character Control Block (CCB)** The character control block (CCB) contains current information on the physical operation of the line and the data being transferred to or from the line. Some of the contents of the CCB are a pointer to the translate/decode table, a CCBL2 pointer, a CCBL3 pointer, and counters that maintain the position of data being accessed within buffers.

Some of the key fields of the CCB are as follows:

- X'24'  Address of current level 2 character-service routine (CCBL2)
- X'30'  Line address (type 2 or 3 scanners) or bit control block (BCB) address (type 1 scanner)
- X'34'  Pointer to line group table (LGT)
- X'38'  Current operational status
- X'46'  Leftmost byte of transmit translate table (rightmost byte is character to be translated)
- X'4C'  Address of next level 3 routine (CCBL3)

**Line Vector Table (LNVT)** The line vector table (LNVT) is generated from the CSB macro and initialized by the LINE macro. A different format is created for type 1 than for type 2 or 3 scanners.

**Type 1 LNVT**

The type 1 line vector table (LNVT) generates an entry of X'10' bytes for each possible line address for a type 1 scanner. An operand of MODEL=3705 generates 64 entries from address X'800' to X'BFF'. An operand of MODEL=3704 generates 32 entries starting at X'800'. A line address of X'00' to X'3F' is multiplied by X'10', then added to X'800' to calculate the BCB address. The BCB is used by the level 2 routines in program support of the ICW used by a type 2 scanner. An undefined line address has the rightmost bit set to 1 in the first halfword. If the bit is 0, the first halfword points to the adapter control block (ACB).

**Type 2 or 3 LNVT**

The type 2 or 3 line vector table (LNVT) generates a two-byte entry for each possible line address (96) for each defined scanner (CSB macro). A single scanner generates 96 halfword entries from X'800' to X'8BF'. Each subsequent CSB macro reserves an additional 96 halfwords. An undefined line address has the rightmost bit set to 1 in a halfword entry. A bit of 0 indicates that the halfword contains the address of the adapter control block (ACB) for
this line. Because the first scanner has 64 lines starting at line address X'20', the first X'20' entries from X'800' to X'83F' are set as invalid.

If a line address is known for a type 2 or 3 scanner, the LNVT entry can be calculated by multiplying the line address by 2 and adding X'800'. The LNVT allows the level 2 routines to find the ACB (and CCB) for a line when only the line address is known.

**Bit Control Block (BCB)** The bit control block (BCB) is a X'10' byte control block which provides the same facility for a type 1 scanner as the ICW hardware provides for a type 2 scanner. A BCB is generated for each LINE macro defined for a type 1 scanner and is a valid entry of the type 1 line vector table (LNVT).

**Service Order Table (SOT)** The BSC/SS service order table (SOT) is generated by a SERVICE macro to identify the sequence of service given to devices on a multipoint line. A pointer in the line control block (LCB) at X'4C' points to the current entry in the table for service. All BSC/SS multipoint lines have a service order table, which contains the following entries:

- X'00'  Maximum number of entries
- X'01'  Number of entries in use
- X'02'  Reserved
- X'04'  Four-byte entries of address pointers to the DVBSTAT field of the DVB for a device in the SOT. More than one entry can point to the same DVB.
- X'xx'  Last four-byte entry has a negative offset to the first entry in the SOT in the first two bytes, and zeros in the last halfword.

Processing within the BSC/SS processor is totally different than in the SDLC support for SDLC resources. The routing of information to a BSC/SS resource includes the system router, command processor, work scheduler, input/output line task (including input/output subtasks), character-service routines, and if required, bit-service routines.
This section describes the BSC/SS processor components, providing information about the functions each component performs, the control blocks each component uses, and the manner in which each component passes control to the next. Figure 10.1 illustrates the components and processing flow.

**System Router** The system router receives control and data PIUs from 'path control out' via a branch. The system router branches to the PIU/BTU converter and, after conversion, control is returned to the system router. The system router resolves the BCU resource ID using the BSC/SS portion of the RVT. From the RVT, the system router obtains resource type information and the address of the control block representing this resource. The control block may be a device base control block (DVB) or a line control block (LCB). A DVB represents a terminal, component, or a BSC/SS cluster (these are considered to be device-type resources). An LCB represents a line (nondevice).

If the resource is not a device, the system router enqueues the BCU on the nondevice input queue. There is only one nondevice input queue, the address of which is in the extended halfword direct addressable (HWE) at X'2C'.

If the resource is a device, the system router enqueues the BCU on the input queue of the DVB representing that resource. If the command and modifier
for the device indicate a critical control command (bit 1=1), the BTU is
enqueued on the devices input queue ahead of data and noncritical control
commands. The ENQUE macro used contains the ACTV=YES operand
which results in a trigger.

**Nondevice Command Processor**  If the system router enqueues a BCU
on the nondevice input queue, the nondevice command processor is triggered.
The nondevice command processor dequeues the BCU from the nondevice
input queue and processes the nondevice command contained within the
BCU.

**Device Command Processor**  If the system router enqueued a BCU on
the input queue for a DVB, the device command processor is triggered. The
device command processor validates the BTU command and modifiers,
dequeues the BCU from the DVB input queue, enqueues it to the DVB work
queue, and triggers the line work scheduler.

**Line Work Scheduler**  The line work scheduler uses the BTU command
and modifiers as a search argument against the line-type command table
(LTCT). In the LTCT, the chain of subtasks necessary to process the BTU
command and modifiers is found. The line work scheduler also dequeues the
BCU from the DVB work queue, enqueues it to the line I/O queue, and
triggers the line I/O task.

**Line I/O Task**  The line I/O task is made up of the sequencer and line I/O
subtasks.

The sequencer simply gives control sequentially, as required, to the line I/O
subtasks contained in the selected chain.

The line I/O subtask chains are made up of pairs of subtask initiators and
subtask terminators (different pairs according to the BTU command and
modifiers), plus a chain terminator (read or write version).

Each I/O subtask initiator stores an IOB command in the IOB (contained in
the ACB for a BSC/SS line) and issues an XIO macro to pass control to the
communications control program (CCP), which runs in level 3.

After the CCP and level 2 processing is completed for a given IOB command,
each I/O subtask terminator gets control when the line I/O sequencer is
triggered by the CCP. The I/O subtask terminator checks to see if the
command completed successfully; if so, the terminator passes control back to
the sequencer, which gives control to the next I/O subtask initiator or the
chain terminator for this chain.

The chain terminator updates the response field in the BCU and issues a
TPPOST macro which branches to the BTU/PIU converter. The converter
then passes the PIU to the channel adapter I/O supervisor.

**Communications Control Program**  The BSC/SS CCP is made up of (1)
the command decoder, (2) the command initializer, (3) the command ender,
and (4) the BSC/SS service-seeking module.

The command decoder receives control from the XIO macro issued in an I/O
subtask initiator. The decoder selects the proper initialization routine by
using the command that was placed in the IOB. The command decoder then
passes control to the command initializer.
The command initializer initializes the CCB (contained in the BSC/SS ACB) and the communication scanner in whatever way is necessary to accomplish the level 2 processing for the IOB command. When the command initializer is finished, level 2 interrupts begin to occur on this line for the IOB command.

When level 2 has finished processing the IOB command, the command ender receives control via a level 3 PCI initiated by level 2. The command ender checks whether a good completion occurred; if so, it triggers the line I/O task.

Character Service Program (CSP) The CSP processes level 2 interrupts from the communications scanner. Processing initially begins according to how the command initializer sets up the CCB level 2 pointer. From then on, CSP updates the CCBL2 pointer as required. For a type 2 scanner (CSB macro-coded with an operand of TYPE=TYPE2), the CSP moves a character at a time into the scanner’s ICW for a given line (for ‘write’ operations) or removes a character at a time from the ICW for a given line (for a ‘read’ operation). For a type 3 scanner (CSB macro coded with an operand of TYPE=TYPE3), the data characters are transferred by cycle steal to the end of an NCP buffer or end of block. When CSP processing is complete for an IOB command, a level 3 PCI is issued to pass control back to the CCP.

Bit Service Bit-service routines are included in the NCP for type 1 communications scanner support only. The bit-service routines emulate the type 2 scanner hardware (serializer/deserializer, ICW, and related functions) not included in the type 1 scanner. To accomplish this emulation, the bit control block (which takes the place of the ICW) is included in the NCP if a CSB macro is coded with an operand of TYPE=TYPE1 For each level 2 bit-service interrupt, the bit-service routines move/remove one bit at a time to the type 1 scanner, and cause the scanner to present a character-service level 2 interrupt when required.

BTU Commands for BSC/SS Resources

The basic transmission unit (BTU) is the unit of transfer within the BSC/SS processor. Data that passes between the host and the BSC/SS processor must be converted between the PIU and the BTU formats. In the buffer, the BTU is contained within the block control unit (BCU).

For data transfer to or from the BSC/SS resources, the BSC/SS processor uses three units of transfer: block, message, and transmission.

A block is the smallest unit recognized by the network control program. For SS devices, the data between two end-of-block (EOB) characters; for BSC devices, the data between a start-of-text (STX) or start-of-header (SOH) character and an end-of-transmission block (ETB).

A message for SS devices is the same as a transmission, that is, the data between a start-of-data (circle D) and end-of-block (EOB), end-of-transmission (EOT); for BSC devices a message is the data between a start-of-text (STX) or start-of-header (SOH) character and ended by an end-of-text (ETX) character.

A transmission for SS devices is the same as a message. For BSC devices, a transmission is ended by an end-of-transmission (EOT).

For large amounts of data coming from a terminal, the BSC/SS processor can run in subblocking mode. The BSC/SS processor passes data to the host as a
subblock before receiving an EOB, ETX, or EOT. Subblocking is in full multiples of NCP buffers, as specified on the TRANSFER operand of the LINE macro, and CUTOFF specifies the number of subblocks before the data is flushed. The BSC/SS processor automatically sends data to the host on an EOB or ETX.

There are five BTU commands that are used with all BSC/SS devices. The commands are: ‘invite’, ‘contact’, ‘read’, ‘write’, and ‘disconnect’. Even though the physical operation may be different for each device, the same commands are used. Each of these commands is discussed in detail later in this manual.

The ‘invite’ and ‘contact’ commands establish a BSC/SS session. The ‘invite’ command implies a ‘read’. The ‘read’ and ‘write’ commands transfer data between the host and a device. The ‘disconnect’ command ends a session. By the use of command modifiers a number of commands can be combined into one request from the host.

There are two other commands the host can send to the BSC/SS processor: ‘control’ and ‘test’. The ‘control’ command is used to alter or examine the status of a line or device. The ‘test’ command is used to test a BSC/SS device or line. The online line test (OLLT) tests BSC/SS lines; the online terminal test (OLTT) tests BSC/SS terminals. For these tests, the text portion of the BTU contains interpretive commands.

The BTU response information is contained within the system response byte of the BTU and the extended response byte of the BTU. The system response byte identifies a response as an error response or normal response. The system response byte also contains the phase (0, 1, 2, or 3) to which this response applies and the system response code. The extended response byte contains the initial status of the line and the final status of the line.

The BTU commands given below may be found in IBM 3704 and 3705 Program Reference Handbook (GY30-3012), Section 3: BTU commands and modifiers.

**Contact Command** The ‘contact’ has a BTU command of X'06' with no modifiers. The 'contact' does not imply any data transfer, but only assures that a connection is available for data transfer. When a response to a ‘contact’ is received by the host, the host may then issue either a ‘read’ or ‘write’ command. Control commands, such as ‘reset device queues’ or ‘reset immediate’ can request termination of a ‘contact’ command.

**Invite Command** ‘Invite’ has a BTU command of X'05', with several modifiers available to qualify the ‘invite’, as follows:

- X'00' Invite normal. Unit of data for this command (block, message or transmission) is specified on the NCP macro defining the device. Default is block.
- X'01' Invite block. Unit of data for this command is a block. Ended by EOB.
- X'02' Invite message. Unit of data for this command is a message. Ended by ETX (BSC) or EOT (SS). Message and transmission are the same for SS.
X'03' Invite transmission. Unit of data for this command is a transmission. Ended by EOT (BSC).

X'04' Invite transmission with disconnect. Executed as an ‘invite transmission’ command followed by a ‘disconnect’ command.

X'05' Invite with auto restart. Executed as unbounded series of ‘invite with disconnect’ commands. This command must be terminated with a ‘reset’ command.

X'06' Invite perpetual. Valid only for clusters. Executed as an unbounded series of ‘invite transmission’ commands with no intervening ‘disconnect’ commands.

If an ‘invite’ is pending (no response from the terminal), and data is available to send to the device, the ‘invite’ can be terminated by control commands of ‘reset invite’, ‘reset conditional’, or ‘reset at end of command’.

A ‘write’ to a BSC 3270 occurs without a reset of the ‘invite perpetual’.

**Read Command** ‘Read’ has a BTU command of X'01' with several modifiers, as follows:

X'00' Read normal. Unit of data for this command (block, message, transmission) is specified on the NCP macro which defines the device. Default is block.

X'01' Read block. Unit of data for this command is the block. Ends with an EOB.

X'02' Read message. Unit of data for this command is the message. Ends with an ETX (BSC) or EOT (SS). The message and transmission are the same for SS.

X'03' Read transmission. Unit of data for this command is a transmission. Ends with an EOT (BSC).

X'04' Read transmission with disconnect. Executed as a ‘read transmission’ command followed by a ‘disconnect’ command.

X'05' Read with invite. Executed as a ‘read transmission with disconnect’ followed by an ‘invite normal’ command.

The read command can be terminated by a ‘reset device queues’, ‘reset immediate’, ‘reset conditional’, or ‘reset at end of command’.

**Write Command** ‘Write has a BTU command of X'02' with several modifiers, as follows:

X'00' Write normal. Unit of data is one block. Ended by an EOB.

X'01' Write with end-of-message. Unit of data is one block followed by the appropriate control sequence for an end of message.

X'02' Write with end of transmission. Unit of data is one block followed by the control sequence for end of transmission.

X'03' Write with disconnect. Executed as a ‘write transmission’ command followed by a ‘disconnect command’.
X'06' Write with read. Executed as a 'write with end of transmission' followed by a 'read' command.

X'07' Write with invite. Executed as a 'write with end of transmission' followed by a 'disconnect' command and then an 'invite' command.

X'08' Write with contact. Executed as a 'contact' command followed by a 'write normal' command. Ended with an EOB.

X'09' Write with contact. Executed as a 'contact' command followed by a 'write with end of message'. Ended with ETX (BSC) or EOT (SS).

X'0A' Write with contact. Executed as a 'contact' command followed by a 'write with end of transmission'. Ended with EOT.

X'0B' Write with contact and disconnect. Executed as a 'contact' command followed by a 'write with end of transmission' followed by a 'disconnect' command.

X'0E' Write with contact and read. Executed as a 'contact' command followed by a 'write with end of transmission' followed by a 'read normal' command.

The 'write' command can be terminated by 'reset device queues', 'reset immediate', 'reset conditional', or 'reset at end of command'.

**Disconnect Command** 'Disconnect' has a BTU command of X'07' with several modifiers, as follows:

- X'00' Disconnect normal. No modifier
- X'01' Disconnect with invite. Executed as a 'disconnect normal' followed by an 'invite normal' command.
- X'02' Disconnect with end of call. For switched lines, this modifier results in the physical connection between the terminal and the communications controller being broken. For nonswitched lines, this modifier is the same as 'disconnect normal'.
- X'03' Disconnect with end of call and invite. Executed as a 'disconnect with end of call' followed by an 'invite' command.

The 'disconnect' command is reset by 'reset immediate'.

**Control and Test Commands** The control commands have a BTU command of X'08' with many modifiers. The test commands have a BTU command of X'03' with many modifiers. A listing of commands and modifiers is given in *IBM 3704 and 3705 Program Reference Handbook* (GY30-3012). Section 3: BTU Commands and Modifiers.
BSC/SS Processor

BSC and SS Sessions

The ability of the NCP BSC/SS processor to conduct multiple sessions on the same multipoint line depends upon the fact that data transfer does not occur continuously for the duration of a session. For example, for inquiry/response applications, the elapsed time between receiving a response from the host processor and entering the next inquiry typically exceeds the time required for transmission of the inquiry and response. This elapsed time is the result of operator 'think' time. The interval during which the terminal is not using the line can profitably be used to service other terminals on the same line.

The number of concurrent sessions to be conducted on a line depends upon several factors. Among these are (1) the relative amount of time a terminal in use does not need the line, and (2) the permissible delay between the time the operator is ready to use the terminal and the time the line is available to that terminal. The number of concurrent sessions on a line is specified by the user in the SESSION operand of the LINE macro. This value is called a session Limit.

The sequence by which the BSC/SS processor attempts to establish sessions on a multipoint line is determined by the service order table associated with the line. This table is defined by the SERVICE macro, directly following the LINE macro.

Logical Connections

A session is active when the BSC/SS processor is communicating with, or is ready to communicate with, the associated device. If the NCP is not communicating with, or is not ready to communicate with, the associated device, the session is either suspended (but within an active session) or inactive.

In most applications it is necessary to limit the amount of time a session is permitted to be active in order to prevent a device, once in session, from monopolizing the line. The period during which a session is active is called a logical connection. The length of a logical connection is the maximum number of transmissions that may be transferred in either direction between the BSC/SS processor and the device during the logical connection. The limit is specified in the XMITLIM operand of the CLUSTER, TERMINAL, or COMP macro representing the device. The user can indicate that the XMITLIM specifies the number of blocks, rather than transmissions, by coding ENDTRNS=EOB on the macro.

Once a session has been established, the BSC/SS processor repolls the device for each subsequent transmission solicited from the device. You may have the program repeat the polling operation one or more times, if you wish to allow the device more time in which to respond. The number of polling operations allowed during this period is specified in the POLIMIT operand of the LINE macro. The value specified in the POLIMIT operand is called the negative response limit. The best performance results occur with a value of 1.

Once the negative response limit is reached, the BSC/SS processor can proceed in one of three ways:

1. NOWAIT. The BSC/SS processor breaks the logical connection and cancels the read request that caused the polling. The host is informed.
2. **WAIT.** The BSC/SS processor maintains the logical connection, holding the line; informs the host the negative poll limit has been reached, and waits for a new command from the host.

3. **QUEUE.** The logical session is suspended, the command is queued for the next logical connection for this device, and the host is notified that the negative poll limit was reached.

Note that the specific operand may be host-system dependent; refer to the appropriate system Programmer's Guide for additional information.

Most types of I/O errors that occur during an active session cause suspension of that session. The host is notified of the error.

**Session-Servicing and Service-Seeking** The activity of attempting to establish a new session on a BSC or SS line is called service-seeking. Service-seeking occurs by searching all DVBs for an 'invite' or 'contact pending' bit in DVBSTAT, in the sequence of the service order table (SERVICE macro). If either the 'invite' or 'contact pending' bits have a value of 1, the device is polled or addressed (if it is a polled device) or otherwise enabled for communication. These bits are 0 once a session is established, but the 'connection exists' bit indicates that 'read' or 'write' commands may be processed during a logical connection. The 'disconnect received' bit in DVBSTAT, set when a 'disconnect' command is received, indicates that the session is to be terminated.

The activity of servicing existing sessions is called session-servicing. Session-servicing occurs by sequentially servicing all of the DVBs queued on the suspended session queue in the LCB for that line. Servicing a session consists of establishing a logical connection, then sending or receiving data (or both) until the logical connection ends.

Session-servicing and service-seeking alternate in a sequence of operations called a service cycle. A service cycle consists of service-seeking and session-servicing if at least one session exists. If no sessions exist, only service-seeking is performed. If the existing sessions equal the session limit, only session-servicing is performed.

The maximum number of devices with which the program attempts to establish a session during each service-seeking operation is called the service-seeking limit. To specify the service-seeking limit, the SERVLIM operand of the LINE macro should be coded with the maximum number of devices with which the program is to attempt service-seeking during one service-seeking operation. Service-seeking attempts are in service order table (SOT) sequence for this count, even if the DVBs searched are currently in session; each DVB is scanned for an 'invite' or 'contact pending'. If response time is poor for existing sessions, you may improve performance by coding SERVLIM with a low value; the default is one-half the entries in the service order table.

You may also specify whether service-seeking or session-servicing is to have priority. This option is specified by coding SERVPR=OLD if session-servicing is to have priority, or SERVPR=NEW if service-seeking is to have priority. If response time is poor for existing sessions, you may improve performance significantly by coding SERVPR=OLD.
Nonproductive polling and the associated processing overhead can be mini-
mized by specifying a service-seeking pause. The pause is in effect only when 
there are no established sessions and only service-seeking occurs for this line. 
The pause is specified in the PAUSE operand of the LINE macro. When the 
first session is established, the pause becomes inoperative until all active 
sessions have terminated for this line.

Session information can be changed by commands from the host. The control 
command (X'08') with the following modifiers can be used for dynamic 
tuning of the network:

- X'84'  Change line service-seeking pause
- X'85'  Change line negative poll response limit
- X'86'  Change session limit
- X'8C'  Change device transmission limit

The fields which are changed by these commands are in the line control block 
(LCB) in the multipoint extension. These values can also be changed from 
the 3704 or 3705 control panel.

In specifying session limits, special consideration must be given to devices 
which use general polling. The BSC 3270 uses the general poll ('invite 
perpetual') to obtain input. A response to a general poll may include data 
from all 3277s on the cluster controller, exceeding the session limit. If the 
session limit is reached by a general poll of one cluster, other clusters on the 
same line are not polled. BSC 3270s should have a session limit equal to the 
total entries in the service order table; one per cluster controller plus one per 
3277.

Write operations to BSC 3270s are queued to the DVB which represents the 
terminal. Read operations occur when the DVB representing the cluster 
controller is processed for service-seeking.

The scheduling of the BSC/SS request for execution begins with the device 
command processor, operating from the device input queue, and is composed 
of a main routine and several subroutines. The device command processor 
decodes the command, makes various error checks, depending upon the 
command, and, if no errors are found, accepts the command. If errors are 
found, the proper response code is moved to the BTU and is returned to the 
host.

Accepting the command, the device command processor enqueues the BCU 
on the device work queue (DVB X'00'). Since the work queue has no execut-
able code associated with it, no task is triggered as a result of the enqueuing. 
If the line work scheduler is idle, the device command processor must trigger 
the line work scheduler to ensure that the input/output operation is initiated.

The device command processor also processes control commands directed to 
a device. If the command is a control command, the device command proc-
cessor enqueues the BCU to the device work queue, provided the control 
command is noncritical and the device is in session. If the command is 
critical, or if the device is not in session, the device command processor passes 
control to the control router.
The nondevice command processor, operating from the nondevice input queue, processes all control commands that are not directed to a device. The processor dequeues the BCU from the nondevice input queue and uses the resource vector table (RVT) to determine the address of the line control block (LCB) representing that line.

If the control command is supported in the system, the nondevice command processor calls the control router. The control router scans the supported control command tables looking for a match. When a match is found, the control router branches to the routine. When the control command routine has finished its processing, the control router triggers the line work scheduler.

The same line work scheduler gets control regardless of the line type. A different subroutine of this task exists for each type of line: point-to-point (which also supports switched callin), switched callout, and multipoint. The line work scheduler assigns a subtask sequence chain to the request by decoding the command, using the line type command table (LTCT). The LTCT contains an offset table and a collection of subtask sequence tables. The offset table corresponds to the system modifier table (a table of all valid command/modifier combinations). The line work scheduler finds the position in the system command table corresponding to the command and modifiers specified in the BTU. The corresponding position in the offset table of the LTCT gives the offset to the appropriate entry in the subtask sequence table. Each entry in the subtask sequence table is a series of pointers to the I/O subtasks necessary to process a particular command. Each pointer is the fullword address of an I/O subtask.

Once the initial pointer to the required subtask sequence has been established by the line work scheduler, the offset into the subtask sequence is stored in the LCB (LCBSSP) for the line. The line work scheduler then enqueues the request BCU on the line I/O queue.

The next task to get control is the line I/O task which was triggered by the line work scheduler. The line I/O task consists of a line I/O sequencer and a series of initiator/terminator subtasks. The subtasks associated with a line I/O task vary, depending upon the command being processed at any given time, and the type of line, switched or nonswitched.

When the line I/O QCB is activated, the line I/O sequencer receives control. The line I/O sequencer updates the subtask sequence pointer passed to it by the line work scheduler and gives control to the next subtask in the sequence. The initiator/terminator subtasks are structured to perform a series of I/O operations. A complete sequence of subtasks executes all the I/O operations necessary to perform a requested function. Initiator subtasks structure the line's input/output block (IOB) by inserting the required I/O command and modifier codes and initializing other appropriate fields, then issuing an XIO macro to start the I/O operation.

When the I/O operation completes, the terminator subtask checks the I/O completion status, initiates any error recovery procedures, and prepares the line I/O task for the next operation. If a terminator does not initiate any action that requires supervisor dispatching (such as issuing an XIO or TRIGGER macro), control returns to the supervisor via the SYSXIT macro to allow other level 5 tasks to compete for level 5 system time.
**BSC/SS XIO Processing**

BSC/SS XIO processing is handled by the communications control program (CCP). The CCP for BSC/SS processing consists of the command decoder (CXECMDC), command initializers (CSECMDI), the command ender (CXECEND) and the BSC/SS service-seeking module (CXESVSK). The CCP routines initiate and terminate data transmissions on the line; the character service program accomplishes the actual data transmission.

After an XIO is issued to a communications line by the level 5 I/O subtask, the communications control program (CCP) gets control from the supervisor. One of three decode routines is entered, depending upon which type of XIO was executed. The three types of XIO commands are:

- Normal IOB (CSEMDC0)
- Set mode (CSECMDCI)
- Immediate control (CXECMDC2)

**Normal IOB Command (CSEMDC0)** With information about the command in the line IOB, the nucleus routine for decoding normal XIO commands performs a number of initialization steps common to all such commands.

Certain IOB fields (status fields, input block size, and immediate control field) are set to zero, and the connection between the adapter control block (ACB) and line vector table (LNVT) is validated. If the ACB/LNVT connection is invalid, the XIO SVC is abended.

The character control block (CCB) is checked to see if that block is already busy executing a command or subblocked operation. The phase bits in CCBCTL are nonzero if the line is busy. If the CCB is found to be busy, the XIO SVC is abended unless the new command is appropriate for completing an outstanding subblocked operation.

The line adapter (ACB) is placed in the NO-OP state unless the ACB is completing a subblocked operation. This state permits level 2 to be enabled while command initialization steps are performed to the CCB, without interference from level 2 interrupt processing on this particular line.

At the start of command initialization, CCB fields are checked for any outstanding status conditions. Such conditions prevent the new command from being executed at this time. The new command is ended, using the outstanding status as its ending status and using the phase set to indicate the clearing of outstanding status.

The command control byte (CCBNCFL) is checked to see whether command initialization can proceed immediately or must be suspended until (1) something is received from the terminal or (2) a timeout completes.

As far as the decoder is concerned, there are three classes of commands: normal, common control, and subblock mode.

The first class consists of the normal data transfer commands, such as 'read initial' and 'write with end of transmission', since their execution is dependent on the particular type of line control. The initialization routine to be branched to is located through the command decode table pointer of the line.
group table (LGT) for the line. The initialization routines for these commands reside in the CSECMDC CSECT.

The common control commands include "enable", "dial", and "disable". Their functions are common to different types of line control, so the initializing routines are the same for all common control commands, which reside in the CSECMDC CSECT.

The subblock mode commands are normally accepted only if the line is busy. The decoder branches to the subblock command initializer, which resides in the CXECMDI CSECT.

The function of the command initializer routines is to examine the command and set up the adapter control block (ACB) with the proper values to handle the I/O on the lines. This module also sets initial timeouts and sets up the interface control word (ICW). Upon completion of level 2 operations, if the expected ending status is satisfied, control returns to the address contained in CCBL3.

**Set Mode Command (CSECMDC1)** The entry point into the communications control program (CCP) for an XIO 'set mode' is CXECMDC1. This code validates the 'set mode' command, vectors into the set mode command decode vector table, using parameters passed from level 5, then branches to the routine for execution. When the set mode function has completed, control returns to the level 5 routine via the supervisor.

**Immediate Control Command (CXECMDC2)** The entry point into the communications control program (CCP) for an XIO immediate is CXECMDC2. Several different types of resets are provided. Some resets are conditional and some are unconditional. The purpose of the resets is to terminate an IOB command operation or an ongoing subblocking operation. If the reset is successful, storing of received data is halted; if the line is subblocking, the receive buffers are released and the data is lost. If the line is transmitting when the reset is executed, the transmission is terminated in an orderly way.

The reset routine contains a branch tree to determine exactly what type of operation is to be reset: receive text, receive control data, transmit text, or transmit control data. For all 'reset immediate' routines, the linkage through the command ender is established via the CCBL3 pointer to the common reset end routine in CXECEND (CXECENDY), and by zeroing expected status. Then if the reset operation is completed without any hardware errors, the IOB command is ended with IOB status set to 'special' and 'reset', and phase set 'on' in the error flags byte. The phase of 'reset' is always 'control'.

There are two other types of 'immediate control' XIO commands. The first causes the break signal (SS only) to be sent if the line is currently executing a 'read' type IOB command. The second type places the line in monitor mode, provided the line is not executing an IOB command. If not busy or if handling a subblock between commands, the line is set to monitor mode, in which the line triggers the LCB's input/output task if an ending status condition occurs.
While in monitor mode, the line is busy to all IOB XIO commands issued to it. The result of an XIO being issued is to abend the task that executed the XIO, if monitor mode has not ended due to ending status or reset.

The function of the character service program (CSP) is to maintain the line discipline while transmitting or receiving data. There are two types of CSP: one for BSC line control, one for SS line control.

Each CSP is made up of routines to handle the line discipline in addition to the transmission and reception of data. Each CSP routine sets up the CCBL2 pointer for the next function needed to complete the I/O operation.

When the I/O operation has completed, the last level 2 CSP routine that gets control queues the ACB to the CCP ACB queue (CCPQOFF) for processing to the level 3 command ender via a PCI level 3 interrupt.

When the CSP routines have finished processing, the last routine issues a PCI level 3 after queueing the ACB to the ACB queue. The PCI level 3 interrupt passes control to the command ender routine in the CCP in level 3. The command ender removes the ACB from the ACB queue, and, if the queue is empty, resets the PCI level 3.

The command ender compares the ending status of the current command with the expected status stored in the CCBESTAT field. If the two agree, and no error bits were flagged, the routine exits to the routine pointed to by CCBL3. The CCBESTAT and CCBL3 fields were both set during command initialization. If CCBESTAT=0, the command ender automatically accepts the results.

If the ending status does not agree with the expected status, or if any error bits were flagged, the error recovery program (ERP) setup routine schedules the appropriate ERPs.

The ERP setup routine uses the phase bits (CCBCTL) and CCBEND1 to vector to the correct ERP branch table within the command ender. The ERP setup routine branches to the ERP routine, where a check is made for retry limits. If the limit has not been reached, the ERP routine branches to the initialization routine to retry the operation. If the limit is reached, the command is ended and the ERP routine triggers the line I/O task. The line I/O task gives control to the I/O terminator subtask to check whether the second level ERP limit has been reached. If the second level ERP limit has not been reached, the I/O terminator subtask reschedules the I/O initiator subtask. Error recovery continues either until the limit is reached or the I/O operation has completed successfully. In either case the I/O terminator subtask TPPOSTs the appropriate response back to the host.

Figure 10.2 illustrates the flow of the BSC/SS Processor. The command sequence is identified in the numbered points following the figure.
Figure 10.2. BSC/SS Processor Flow

1. Channel IOS branches to path control.
2. Path control uses the DAF to access the SIT, SVT, and RVT in order to identify a BSC/SS resource. Path control branches to the system router, which converts the FIDO PIU to a BTU.
3. The system router enqueues the BTU to either (1) a DVB queue, or (2) if the destination is a line, to the nondevice input queue.
4. The enqueuing triggers the DVB or nondevice processor.
5. The device command processor moves the BTU to the DVB work queue.
6. The line work queue is unconditionally triggered.
7. The line work scheduler moves the BTU to the line I/O queue and selects the proper subtask sequence.
8. The line I/O task sequences the initiators and terminators via branches.
9. The initiators issue the XIO.
10. Command decode selects the command initiator.
11. The command initiator sets up the line ICW, CCB, and CCBL2.
12. CSP handles the transmission or receive.
13. The end of the command at level 2 initiates a PCI to level 3 to the command ender.
14. The line I/O queue is triggered.
15. The terminator subtask (or error routine, if necessary) returns control to the line I/O task sequence (item 8).
16. The chain terminator TPPOSTs the BTU to the convert routine to change the BTU to a FIDO PIU.
17. The convert routine XPORTs the PIU to the channel queue to be sent to the host.

Refer to the following publication for additional information:

*IBM 3704 and 3705 Communications Controllers, Network Control Program/VS, Program Logic Manual* (SY30-3013). See:

- Appendix B: BSC/SS Control Command Cross Reference Table
- Appendix C: Sequences of I/O Subtasks for BSC/SS Processing in Level 5
- Appendix D: Command Sequence Charts (identifies the CCBL2 and CCBL3 Routines)
- Appendix F: Online tests

**Block-handler Routines**

The BSC/SS processor provides three points at which user-written routines or IBM-supplied routines may be executed for the manipulation of data. These data manipulation routines are called block-handler routines (BHR).

Block-handler routines are data-oriented. The routines are given access to blocks that contain data at the following times:

1. Before the output is sent to a device: blocks accompany 'write' commands only (execution points 1 and 2).
2. After input is received from a device: blocks accompanying commands of 'read', 'invite', 'write conversational', 'write with read modifier' (in read phase), 'write with contact and read modifiers' (in read phase) (execution points 2 and 3).
3. When the block is in error (execution points 2 and 3).

Block-handler routines (BHR) are grouped into units called block-handlers (BH). A block-handler is designated at NCP generation to be executed at one of the three points listed below. Up to three block-handlers (a possibility of one for each execution point) are grouped to form a block-handler set (BHS). Each device can be assigned a BHS at NCP generation. The BHS can be flagged as initially executable or it can be activated later by a control command. A control command can also be used to assign a BHS dynamically to a device, or to change the BHS association specified at NCP generation.

The BCU being edited by the BHRs resides on a different queue at each of the three points of execution. At point 1, the BCU is on the device input...
queue (DVB); at point 2, the BCU is on the line input/output queue; at point 3, the BCU is on the point 3 BHR queue extension to the DVB.

The three execution points are as follows:

**Point 1**

The point 1 entry is used for BCUs to be written to the BSC or SS device. Point 1 BHRs are executed after the BTU is received from the host and before the line has been scheduled for the I/O operation. The device command processor is the interface with the BHR mechanism for point 1. No BCUs are in error at this point.

**Point 2**

Point 2 BHR is invoked during execution of certain initiator and terminator subtasks by the BHEXIT macro. Since BHRs are data-oriented, the only initiator subtask that invokes BHRs at point 2 is the write initiator. All terminator subtasks that represent termination of a read command invoke BHRs at this point. The subtasks include the common read terminator, the display service-seeking terminator, and the chain terminator (read entry point only). The following subtasks invoke BHRs at point 2 for BCUs that are in error: the ‘write terminator’, the ‘read terminator’ routine, the ‘common read terminator’, the ‘display service-seeking terminator’, the ‘contact terminator’, and the ‘error retry’ routine.

**Point 3**

The TPPOST routine is the interface for point 3. At this point all processing on the BCU has been completed and the BCU is ready to be sent to the host. The TPPOST routine puts the BCU on the point 3 BHR queue of the DVB. When the BHRs have completed processing, an XIO macro instruction is used to send the BCU to the host.

**Block-handler Control Blocks** Figure 10.3 shows the relationships of the control blocks associated with block-handler routines. The paragraphs that follow explain the function of each block. The BHR extension to the DVB exists for those devices specified to have block-handler routines associated with them. The extension reserves space for a pointer to the block-handler set. If a block-handler set is defined at generation time, the address of a block-handler set is assigned at the same time. The pointer is changed if the block-handler set for this device is changed via an optional control command. The BHR extension also contains the QCB for a BHR queue, which is used at point 3 if the device macro is coded with an operand of PT3EXEC=YES.

The block-handler set (BHS) contains pointers to the block-handler driver tables (BHD) that are to be executed at each of the three-points (or the BHS entry contains zero if no block-handler is defined for a point).

The block-handler driver table (BHD) defines the block-handler routines that are to be executed for each block-handler. Each entry contains a pointer to the BHR, control information related to the BHR, and a one-byte parameter or an address to a parameter list.

A block-handler set table (BST) has an entry for each block-handler set (BHS) defined in the NCP. Each entry contains control flags, plus the
address of the block-handler set (BHS). This table is used in modifying block-handler sets associated with particular devices.

BHR Extension to the DVB

Terminals that have BHRs associated with them have a DVB with a BHR extension. This extension contains a pointer to the block-handler set (BHS) for this terminal, and also contains the point 3 BHR QCB (if PT3EXEC=YES was coded on the CLUSTER, TERMINAL, or COMP macro).

The DVB address relating to BHRs is:

- X'28' Offset to BHR extension

The two fields in the block-handler extension to the DVB (BHR) are:

- X'00' Pointer to the block-handler set (BHS)
X'04'  Point 3 QCB

**Block-handler Set (BHS)**

The block-handler set (BHS) contains pointers to the one, two, or three block-handlers that are to be executed for this set. If a block-handler is not defined, the address pointer contains zeros. If a block-handler is defined, the pointers are addresses of a block handler driver table.

The following are the fields of a block-handler set:

- X'00'  Pointer to point 1 BHD
- X'04'  Pointer to point 2 BHD
- X'08'  Pointer to point 3 BHD

**Block-handler Driver Table (BHD)**

The block-handler driver table (BHD) defines the block-handler routines (time and date, edit, and user block-handler) that are to be executed, at a point, for a block-handler. The BHD is created by the STARTBH, DATETIME, EDIT, UBHR, ENDH Macro grouping. Each entry in the BHD contains a pointer to the BHR, control information, and parameter information.

The BHD contains one entry for each coded macro of DATETIME, EDIT, or UBHR, with the following fields:

- X'00'  Pointer to the block-handler routine
- X'04'  Pointer to parameter list for edit

The parameter list for the EDIT BHD contains the following fields:

- X'00'  Backspace character
- X'01'  Flags
- X'02'  Record descriptor masking configuration

**Block-handler Set Table (BST)**

The block-handler set table (BST) contains an entry for each block-handler set defined in the NCP generation. The address of the BST is in XDA at X'7F4'. This table is used for dynamic block-handler set association.

The block-handler set table (BST) contains one entry for each block-handler set (BHS) defined. Each entry contains an address of a block-handler set (BHS).

**User Block-handler Routines**  User block-handler routines are identified in a block-handler by the UBHR Macro. The user routine must be preassembled in the library identified by the USERLIB (and QUALIFY) operand of the BUILD Macro.

The user routine is written with 3704 and 3705 communications controller instructions, assembler instructions, and internal macros. *IBM 3704 and 3705 NCP Instructions and Supervisor Macros* (SR20-4512) provides user coding information.
At entry to a user routine, register 2 contains the address of the QCB which contains the block to be processed. The NCP abends if a valid BCU is not available when the user code returns control to the NCP.

The multiple terminal access (MTA) feature of the network control program allows the communications controller to communicate with several common types of SS terminals over a single switched network port. When a terminal calls in over a line identified at NCP generation as an MTA line, the NCP identifies the type of terminal and the transmission code being used, and initializes the line's adapter control block (ACB) accordingly. The NCP then communicates with the terminal normally until the session ends.

The types of terminals supported by MTA are:

- IBM 2741
- Western Union TWX
- IBM 2740 transmit control (with or without checking)
- IBM 1050
- IBM 2740 basic (with or without checking)

The NCP terminal identification procedure always tests for terminal type (of terminals defined in the NCP system), in the order listed above.

**Multiple Terminal Access (MTA) Control Blocks**

In order to identify the type of terminal and to establish the appropriate operating parameters (speed and transmission code) once the terminal is identified, the NCP uses several tables. This section describes the function and relationships of these tables.

**MTA GROUP, LINE, and TERMINAL**

The actual line interface definition requires a GROUP, LINE, and TERMINAL macro definition. The TERMINAL macro has an operand of TERM=MTA. These macros create the line group table (LGT), line control block (LCB), adapter control block (ACB), and device base control block (DVB) which are used for the initial connection. The incoming MTA call is received using these control block definitions.

**MTA List**

The MTA list is a table of one-byte entries, each entry representing one of the five terminal types that can call in on an MTA line. The list consists of a group of entries for each combination of terminal types on MTA lines in the telecommunication subsystem. The entries in a group are always in the order in which the NCP tests for terminal type. The following values represent the given terminal type in the MTA list:

- X'00' 2741
- X'01' TWX
- X'02' 2740 transmit control
- X'03' 1050
- X'04' 2740 basic
A group of entries is delimited by a byte containing the value X'FF'.

The MTA identification routine uses the MTA list to determine which types of terminals to test for. The initial offset into the list is in IOBSTOFS field of the line's IOB.

The multiple terminal access (MTA) identification routine uses the MTA list as an index into the code for testing the terminal type. The routine sets a timeout at the beginning of each test. If the terminal does not respond before the timeout expires, the MTA routine is reentered and the next MTA list entry is used to test for the next terminal type. If the delimiter entry X'FF' is reached, the routine disconnects the line and ends the command.

The sequence for terminal checking occurs in the following manner:

The MTA identification routine checks for both 2741 and TWX at the same time (if either terminal type is specified in the MTA list). The routine sends an EOT to the device and sets a timeout. If the terminal responds with an EOA, the routine assumes the terminal is a 2741. If leading graphics are present and the response is the WRU character, the routine assumes a TWX terminal.

To test for 2740 transmit control, the MTA routine sends a slash-space (/b) character sequence. If the terminal responds with an EOA, the routine assumes the terminal is a 2740 with transmit control. If the response is a NAK, a 1050 terminal is implied.

To test for a 1050, the MTA routine must poll the terminal. The routine fetches and sends the polling characters for each device on the 1050 polling list until it receives a response. If the response ends in EOT, the routine disconnects the line and ends the command.

The 2740 basic test consists of transmitting a BID message to the terminal. The message, sent in both EBCD and correspondence code, prints at the terminal and indicates that the operator is to enter the MTA sign-on sequence (covered later in this manual).

When the terminal type (but not the transmission code) has been identified, the control pointer from the original line group table (LGT) can be updated to the stand-alone line group table (LGT). The two tables were created by GROUP macros defined for each MTA terminal type.

As an example, consider a telecommunication subsystem with three MTA lines. One line has all five terminal types; the second has 2741 and 1050 terminals; and the third has TWX, 2740 transmit control, and 1050 terminals. The MTA list for this NCP has the format shown in Figure 10.4.
The MTA list is defined by MTALIST macros coded for NCP generation.

**MTA Line Group Table (LGT)**

A GROUP macro, creating a line group table (LGT), is defined for the MTA line, plus one line-group table per MTA terminal type which calls in. The MTA identification routine initially uses the group definition associated with the MTA line. As soon as the terminal type and transmission code are identified, the pointer is changed to the specific LGT for the terminal type.

**Line Control Selection Table (LCST)**

The line control selection table (one per NCP) is used by the multiple terminal access (MTA) identification routine to initialize the line’s character control block (CCB), once the routine has identified the type of terminal calling in. The table is also used to establish CCB parameters when the NCP calls a device on an MTA line.

The LCST may contain up to sixty-three 16-byte entries, each representing a particular set of operating parameters for some MTA device (or devices) in the telecommunication subsystem. The first entry in the LCST is used by the MTA identification routine during the identification process and does not represent a particular type of device.

The parameters in an LCST entry are those that can vary for terminal type, transmission code, or individual device. These parameters include such variables as line speed, carriage return rate, translate table addresses, size of print line, and error retry limits.

A series of MTALCST macros is coded for NCP generation to define the LCST entries.

Once the MTA identification routine has identified the terminal type calling in, the routine determines which LCST entry to use by referring to a list of valid LCSTs for that terminal type. One list exists for each possible combination of terminal type and transmission code. The terminal operator must
enter a sign-on sequence to identify the correct terminal/code list and entry within the list. Each list contains up to ten halfword pointers to valid LCSTs for the combinations which that list represents. The sign-on sequence may include two identical digits, representing the number of the list entry to be used, relative to the beginning of the list for this terminal/code type. If the number is omitted, the routine assumes the first entry is to be used.

As an example, assume that the terminal has been identified as a 1050. The terminal operator enters sign-on sequence, /"44 CR EOB. The /" is unique for each type code, which identifies the code as BCD. Now that the terminal type and code are known, the digits '44' indicate that the MTA identification routine is to use entry 4 in the list of LCST pointers for 1050 BCD terminals. All 1050 terminals are checked terminals; however, the EOB, rather than EOT, provides the definition of a terminal with checking verses nonchecking features. Once the appropriate pointer to an LCST entry is located, the control block fields can be filled in. The relationship is shown in Figure 10.5.

![Figure 10.5. LCST Entry Relationships](image)

The relationship of the pointers in the list to the LCST entries which the pointers represent is established during NCP generation, according to the parameters specified for the MTA lines and devices. MTATABL macros are used to define these lists for NCP generation.

**1050 Polling List**

When testing for a 1050 terminal, the MTA identification routine must poll the terminal. For this purpose, a single polling list exists in the NCP for all 1050 terminals on all MTA lines. The polling list contains a halfword pointer to the polling characters for each such device. The MTA identification routine goes through each entry in the polling list until it receives a positive response from the device or until it exhausts the list. In the latter case, the routine assumes that the device is not a 1050 and goes on to test for the next terminal type. Each polling attempt which is not successful requires a polling
timeout; if many sets of 1050 polling characters are in the list, with a one-minute timeout per polling attempt, the sign-on could take excessively long.

The entries in the 1050 polling list are specified with the MTAPOLL NCP generation macro.

The BSC/SS processor is that part of the NCP that processes requests for BSC or SS resources. The first BSC/SS processor component to receive control is the BSC/SS system router.

When ‘path control out’ passes a PIU to the BSC/SS system router, the system router branches to the PIU/BTU converter to convert the PIU to a BCU. From this point the BSC/SS processor component flow is as follows:

Upon receiving the BCU from the converter, the system router enqueues the BCU for the device command processor or nondevice command processor for line-oriented control commands. The device command processor passes control to the line work scheduler. The line work scheduler selects a chain of subtasks and passes control to the line I/O task. The communications control program gets control and sets up for the start of level 2 activity. The character-service program (or the bit-service routines, if present) transfers data between the NCP buffers and the communications scanner. Upon completion of the level 2 activity, control returns to the CCP and then to the line I/O task. The BCU is converted to a FID0 PIU and XPORTed to the channel queue.

The storage listing in Appendix C is required for the following quiz. Do not refer to the solution in Appendix B until you have answered all of the questions.

1. The BSC/SS processor receives what type of FID from ‘path control out’?
2. What is the FID converted to before BSC/SS processing modules may process the buffers?
3. The system router enqueues the buffers from the host on one of two types of queues. What are the queues?
4. How many BSC and/or SS lines are defined in Appendix C?
5. How many BSC and/or SS resources are defined in Appendix C?
6. Using the formatted buffers in Appendix C, analyze the buffers from X'19E24' through X'19F08', X'19FA0' through X'1A414', and X'1A5DC' to determine PIU origin, destination, command, and element affected by the command.
7. Using the formatted buffers in Appendix C, analyze the buffers from X'1B208', X'1B2EC' through X'1B3D0', and X'1B4B4' to determine PIU origin, destination, command, and element affected by the command.

Criterion

If you missed more than two questions, you should review this tonic.
Service Aids and Diagnostics

Objective

Upon completion of this topic, the student should be able to identify the NCP service aids and diagnostic aids.

Purpose of Service Aids and Diagnostics

Service aid facilities and panel support routines provide the means for isolating and/or interrogating NCP failures. The NCP supports several functions to aid in problem determination and diagnostics. This topic explores the aids and diagnostic facilities available with the NCP, describes their implementation, and discusses their output.

For additional information, refer to *IBM Advanced Function NCP and Related Host Traces* (SR20-4510).

Dynamic Panel Display

The NCP allows you to display dynamically the following types of information on the 3704/3705 control panels:

- Communication scanner interface control word (ICW)
- Contents of external registers
- Contents of a halfword of 3704/3705 controller storage

Dynamic display functions are selected by setting the display/function select and storage address/register data switches on the panel and pressing the interrupt key.

NCP uses a group of routines to process level 3 interrupts from the panel. The panel control block (PCB) is the common data area for all panel routines.

The panel routines are provided in one of the following:

*Guide to Using the IBM 3704 Communications Controller Control Panel* (GA27-3086)

*Guide to Using the IBM 3705 Communications Controller Control Panel* (GA27-3087)

Line Test

The line test facility allows the user to address, poll, dial, and transmit to or receive from a terminal. Testing is initiated by entering variables through the 3704/3705 control panel. The status of the line resulting from the test is displayed in the panel lights. The line test control block (LTS) contains control information for panel test operations. See one of the two control panel guides mentioned above (under ‘Dynamic Panel Display’) for operating instructions.

The line test facility is included in the NCP whenever SS or BSC devices are defined for the generation.
Address Trace

The address trace facility allows the user to select any combination of up to four registers and storage halfwords, contents of which are to be recorded each time data is loaded from or stored into a specified 3704/3705 storage address at a specified program level. The NCP records the trace data in a trace table within controller storage. The contents of the trace table can be displayed on the control panel or examined in a dump listing. The address trace control block (ATB) has the address trace control information within it.

Operating procedures are given in one of the two control panel guides identified above (see the section on 'Dynamic Panel Display').

The TRACE operand of the BUILD macro specifies whether the address trace facility is to be included in the NCP and specifies the size of the trace table.

Channel Adapter Trace

Channel adapter trace is an optional diagnostic and debugging aid that stores certain fields from the channel control block, type 2 or 3 channel adapter (CHB) or channel operations block, type 1 or 4 channel adapter (COB), in a trace table. An entry is made for channel adapter spurious interrupts, channel adapter level 3 interrupts, and level 1 interrupts caused by channel adapter errors.

The trace is included in the NCP by reassembling SYSCG006 and specifying the TRACE operand which indicates the number of trace entries desired.

This trace cannot be activated or deactivated, only included or excluded. The trace involves significant overhead, especially with a type 1 or type 4 channel adapter. The trace should not be included except in cases where a suspected or known channel error must be isolated.

Line Trace

The line trace facility is a diagnostic and debugging aid that stores certain fields from the ICW (or bit control block) each time a level 2 interrupt occurs on a designated communication line. Line trace is activated and deactivated by network control commands from the host. Only one line at a time may be traced. If the line is duplex, both legs are traced. The fields traced are the line control definer (LCD), primary control field (PCF), secondary control field (SCF), and the parallel data field (PDF). A timer field is also included. The line trace control block (LTCB) contains pertinent information about the trace.

An explanation of the line trace fields is available in *Advanced Function NCP and Related Host Traces* (SR20-4510).

Error and Statistic Recording

NCP has the ability to create records for the miscellaneous data recorder (MDR). There are two types of records:

1. MDR records are built in the check-record pool (CRP) for adapter checks, program checks and unresolved interrupts, then are and sent to the host on the next level 3 timer interrupt.

2. MDR records are dynamically built for line statistics and permanent line errors and are immediately sent to the host.
**Online Tests**
The online tests (a user selected option) provide the IBM customer engineer with online maintenance capability. Testing is performed by one or two routines depending on the type of resource to be tested. The online line tests (OLLT) check BSC/SS lines and SDLC links. The online terminal test (OLTT) checks BSC/SS devices. Both tests are controlled by the terminal online test executive program (TOLTEP) which resides in the host.

To include this facility in NCP, code the OLT=YES operand in the BUILD macro.

**Abend**
Programming errors detected during execution of supervisory and nonsupervisory code of the NCP cause an abnormal end of program execution. The examination of abend codes within an NCP dump can help in locating the error. The optional abend service aid extends detection of programming errors to the NCP supervisor, thus causing the program to terminate before a supervisor error can be propagated into nonsupervisory portions of the program. The abend service aid stores an abend code at X'760' in controller storage and the controller is hard-stopped.

To include the abend service aid for programming levels 1 through 4 of the NCP, code ABEND=YES in the BUILD macro.
Appendix A

PIU Command Sequence

The following SRL references may be of assistance:

IBM 3704 and 3705 Program Reference Handbook (GY30-3012), Section 4: NCP Network Commands

IBM 3704 and 3705 Communications Controllers, Network Control Program/VS Program Logic Manual (SY30-3013), Appendix A: Network Commands

This section describes the command sequence to be followed for activation and session initiation for switched SDLC. Each entry in the 'switched' sequence is marked with an asterisk; you can determine the 'nonswitched' SDLC sequence by skipping those entries. In addition, this section identifies the general processing within the NCP and specifies the NCP control block changes made to record command processing.

If the correct operation takes place, the following command sequence occurs on a PIU trace:

1. Initialization complete -- from NCP physical services to SSCP
   
   This message, generated at the completion of NCP initialization, is placed on the channel intermediate queue of the channel control block (COB or CHB) for transmission as the first message to the host. No response is requested.

2. Activate physical -- from SSCP to NCP physical services
   
   This command is enqueued on the physical services block (PSB) connection point manager (CPM-OUT) queue. The processing sets the 'session established' bit to 1 in PSBPSTAT. A response is provided to SSCP from PSB CPM-IN and XPORTed to the channel intermediate queue of the channel control block (CHB or COB).

Configuration restart specifies a warm or cold restart. The cold restart results in a new initial program load or in initialization of a loaded NCP which has not been previously activated. Cold restart results in commands being directed only to the network addresses which are to be activated (ISTATUS=ACT). A warm restart to a previously activated NCP has a network command addressed to every network resource: an 'activate' to network addresses to be initially active, and a 'deactivate' to each network address to be inactive. The active or inactive status of each network addressable resource is maintained in the disk configuration data set of VTAM. A warm restart allows an NCP with partitioned emulation program (PEP) to be restarted without affecting the emulation lines by reloading.

The response in the request/response unit (RU) contains the name of the NCP. This name is the NEWNAME operand value from the BUILD macro, which was obtained from the physical services block (PSB) at PSBLDID.
NOTE: This command completes the first level of sessions between SSCP and NCP.

3. Start data traffic -- from SSCP to NCP physical services

This command is enqueued on the PSB CPM-OUT queue. The 'data flow enabled' bit and 'data flow active' bits of PSBPSTAT are set to 1. A response is provided to SSCP from PSB CPM-IN and XPORTed to the channel intermediate queue of the CHB or COB.

4. Set control vector -- from SSCP to NCP physical services

This command is optional (from NCP requirements) during activation. In the request unit (RU) byte 5, a value of 01 identifies this as the command which provides the time and date to be stored in the time and date control block. A response from PSB CPM-IN is sent to the host via the channel queue.

5. Set control vector -- from SSCP to NCP physical services

This command is optional (from NCP requirements) during activation. In the request unit (RU) byte 5, a value of 05 identifies this as the command which changes the channel delay from the user-coded value of zero for the duration of the bring-up sequence. A response from NCP CPM-IN is sent to the host.

The same command is used to change the channel delay back to the original value after initialization is complete; the present VTAM support of this command resets the delay after the 'activate link' command responses are received.

6. Activate link -- from SSCP to NCP physical services

A command for each defined link is sent to NCP physical services CPM-OUT. Each command identifies the network address of the link to be activated in the request unit (RU) field of RU1NA. The PSB CPM-OUT triggers the link control block (LKB) which defines the link. The processing initiates the link scheduler code (level 3 NCP code) to search the service order table for that link for work to be done. For a nonswitched link, the modem is enabled. For a switched link, the modem is not enabled until a 'dial' or 'answer' command is processed. The active status of the link is provided in the LKB in LKBSTAT. The task address at LKB LKWTSKEP is changed from the run initiator to the address of the run terminator. (The change of the task address is referred to later under 'dial', 'answer', and 'contact' commands.) Finally, when processing is complete, the PSB CPM-IN is triggered to send a response to SSCP.

The link scheduler is now active for this line definition. A timer queue is initiated for the PAUSE operand value before the link scheduler searches the service order table (SOT) for work to be performed. If the link scheduler completes a pass through the service order table in less than PAUSE value time, the LKB is placed on the timer queue. Service is suspended on this link until the time value expires or outbound data is enqueued for this link.
A negative response to an ‘activate link’ command normally indicates a modem problem, as only the ‘enable’ is processed on the link interface.

An activate line command to a BSC/SS line is followed by a set mode command for each device on the line. If the line is not multipoint, a buffer is allocated for input to the DVB work QCB. The BSC/SS BTU commands of Invite, Contact, Read, Write, and Discontact are valid.

7. * Answer or dial -- SSCP to physical services CPM-OUT (switched line only)

On a switched link the line must be enabled for answering incoming calls or provided with a telephone number for outgoing calls. Either command is logical following an ‘activate link’ command. The ‘dial’ or ‘answer’ is addressed to physical services with the link identified in the request unit (RU) field of RU1NA. The ‘dial’ or ‘answer’ request is acknowledged by physical services CPM-IN and a connection is then attempted.

The connection for ‘answer’ enables the link for an incoming call. The ‘dial’ connection consists of using the autocall unit to dial the telephone number provided in the ‘dial’ command request unit (RU), starting at RU byte 9. When the connection is established, an SDLC command of ‘exchange ID’ (XID) is transmitted to the physical unit with an address of FF (general poll). The XID response provides the terminal ID, and the LKB task at LKWTSKEP (run terminator task) is triggered. The task determines that a connection has been made, triggers physical services CPM-IN to send an ‘off-hook’ command to SSCP, and restarts the link scheduler run initiator.

The terminal ID is a 48-bit value with the following fields:

0     Reserved
x     Physical unit type (1 or 2)
00    Reserved
xxx   ID block, hardware by device type (example: 3790 006)
xxxxx  ID number, hardware or control program specified

The ‘dial’ or ‘answer’ status is indicated by bit settings in the LKB field of LKBSWST.

If a failure occurs during a callin or dial callout, the NCP physical services creates an ‘inoperative’ command with the failing link network address identified in the request unit (RU) field of RU1NA. An explanation of the ‘inoperative’ command is covered later in this appendix.

8. * Off-hook -- physical services to SSCP (switched link only)

‘Off-hook’ informs the SSCP that a physical connection has been established as a result of an ‘answer’ or ‘dial’ command. The network address of the link is carried in the request unit (RU) at RU1NA, and the terminal ID received by the SDLC XID response is sent in the RU in bytes 5-10. No response is requested by physical services.
9. * Set control vector PU -- SSCP to physical services (switched link only)

The original definition of the physical unit on this switched link was given to provide an unformatted control block for any switched physical unit calling in or being called. The 'set control vector PU' (RU byte 5 with a value of 3) provides the values for the CUB control block. The data provided to initialize the control block starting at RU byte 6 is as follows (see Appendix A of PLM):

```
byte 6    SDLC station address
byte 7    Physical unit type
byte 8    Reserved
byte 9    MAXOUT value
byte 10   PASSLIM value
byte 11   Immediate or deferred error recovery
byte 12-13 Reserved
byte 14-15 MAXDATA value
```

Physical services initializes the CUB control block and sends an acknowledgement to SSCP.

10. Contact -- from SSCP to NCP physical services

The 'contact' command is addressed to NCP physical services PSB CPM-OUT. The network address to be contacted is provided in the request unit (RU) at RUINA. The common unit physical block (CUB) is located and the 'set normal response mode' bit at CUBSSCP is set to 1. The PSB CPM-IN sends a response to SSCP acknowledging the PIU, but not acknowledging that the device was contacted.

The link scheduler, which was started for this link by the 'activate link' command, placed the LKB on a timer queue. When the timer interrupt occurs, the link scheduler searches the CUBs on that link for work. Each CUB is initially defined as being in the disconnect mode (CUB field of CUBSSCP) and the poll skip flag at CUBSSCF is on. Normal servicing is still indicated as being disconnected; however, the link scheduler looks for command processing after the normal servicing sequence.

When the 'set normal response mode' bit is found, the link scheduler sends an SDLC command of SNRM on the link to the device defined by the CUB. If a timeout occurs before a response, the 'set normal response' bit is left 'on'. With an SNRM bit 'on', an attempt is made to contact one of the CUBs on this link each time this link is serviced for 'contact poll' commands. If a response is received to the SDLC SNRM, the LKB task at LWTSKEP of the LKB is triggered. This task is the run terminator task set up by the 'activate link' command. The 'run terminator' triggers the PSB CPM-IN to send in the 'contacted' command from NCP PSB to SSCP. Also, the link scheduler is restarted (as if a new 'activate link' were issued), and again the task address of the run terminator is in the LKB task address.
NOTE: If the physical unit contacted is type 4 (remote NCP), the bring-up sequence depends upon the response to the SNRM SDLC command. If 'request initialization' (RQI) was received, the 'load initial', 'load data' (repeated), 'load final' take place. The 'contact' command is retried until an SNRM response of 'nonsequenced acknowledgement' (NSA) is received. When an NSA is received, initialization of the remote begins with the first item of this list ('activate physical' to the remote NCP physical services).

11. Contacted -- NCP physical services to SSCP

The 'contacted' command was initiated by a 'nonsequenced acknowledgement' (NSA) SDLC reply from a physical unit as a response to the 'set normal response mode' (SNRM). This command provides the network address of the physical device contacted in the request unit (RU) at RU1NA.

This information sent to SSCP allows the physical unit to be sent the next command (an 'activate physical') to establish the next level of session.

12. Activate physical -- SSCP to CUB physical unit process queue

The 'activate physical' command is enqueued to the CUB physical unit process queue. This command is the first command not addressed to NCP physical services and is the first which may be sent on a link to a physical unit. If the device is a type 2 physical unit, the command is transmitted to the physical unit. If the device is a type 1 physical unit, the command is processed by the NCP and is not sent to the physical unit.

The processing of the command results in setting the 'processing session initiating request' bit of CUBSTAT in the CUB control block to 1. The command format is modified from FIDI to FID2 and placed on the CUB link outbound queue (CUBLOBH); this is the queue searched by the link scheduler (started by an 'activate link' command) for data to be transmitted to the physical unit.

When the command has been processed by the physical unit and the response received by the link scheduler polling the physical unit, the response is enqueued on the CUB link inbound queue, triggering the CUB link inbound task. The task converts the PIU from FID2 to FID1 and checks for the type of response. If a positive response is received, the 'processing session initiation request' bit is set to 0, and the 'session established' bit is set to 1 in CUBSTAT of the CUB. If a negative response is received, the 'processing session initiation request' bit is set to 0. The response is XPORTed to the channel queue to be sent to SSCP.

The response request/response unit (RU) contains the name of the control program generation name for type 2 physical units.

For the type 1 PU, the 'session established' bit is set to 1 by the physical unit processing queue task, and the response to SSCP is created by the NCP.
NOTE: This command completes the second level of session (SSCP/CUB).

13. * Assign network addresses -- SSCP to physical services (switched link only)

The logical unit definitions for a switched SDLC link are created by the LUPOOL macro, and the network addresses assigned are the highest addresses in the NCP; that is, the last entries in the resource vector table (RVT). The logical units are not assigned to any switched link or physical unit at generation time. The CUB has a switched extension of four bytes which contains the maximum count of entries and an address pointer to a logical unit vector table (LUV). The definition on the PU macro of MAXLU creates the LUV and pointer to the LUV. This command initializes the LUV with the addresses of LUBs from the LUPOOL.

The fields in the 'assign network addresses' request unit are:

- bytes 3-4 Network address of the physical unit
- byte 5 Number of LU addresses to be assigned
- byte 6 X'80'
- bytes 7-n Network (LU) addresses to be assigned

The number of addresses to be assigned may not exceed the entries in the LUV table (MAXLU operand of PU), and the addresses of LUs assigned are allocated from available entries in the logical unit pool (LUPOOL). When the logical unit block (LUB) address entry is provided in the LUV, the address pointer to the CUB is provided at LUBCUB field in the LUB.

Physical services builds a response to SSCP when the command has been processed and the network addresses have been assigned in the LUV.

14. * Set control vector LU -- SSCP to physical services (switched link only)

The 'set control vector LU' command to NCP physical services provides the LU network address in the request unit (RU). A separate 'set control vector LU' command must be processed for each logical unit (LUB) to be used during a switched connection.

The command provides the following data:

- byte 6 - LUB network address
- byte 7 - n pacing count
- byte 8 - m pacing count
- byte 9 - Dispatching priority of APPL/LU CPM-OUT task (BATCH operand of LU)

The logical unit block (LUB) is now initialized with appropriate definitions and pointers which are generated for nonswitched LUBs.
15. Activate logical -- SSCP to LU/SSCP process queue

The 'activate logical' command is enqueued on the LU/SSCP process queue of the logical unit control block (LUB). The LUB CPM-OUT task checks the command type, turns the 'processing activate logical' bit to 1 in LUBCPSET of the LUB, converts the command from FID1 to FID2 (or FID3), and places the command on the CUB link outbound queue for the link scheduler to find and transmit. Except for SDLC 3270, an 'activate logical' is processed by the link-attached physical unit. All commands for the SDLC 3270 are processed by the network control program.

The PIU response to polling the physical unit is enqueued to the CUB link inbound queue. The CUB link inbound task dequeues the FID2 or FID3, converts it to a FID1, and branches to the CPM-IN task of LU/SSCP to process the input. A positive response requires the 'processing activate logical' bit to be set to 0 and the 'session established' bit to be set to 1 in LUBCPSET of the LUB. The response is then XPORTed to the channel queue to be sent to SSCP in the host.

NOTE: This command completes the third level of sessions (SSCP/LU). No additional session is started until an application program is connected to be logical unit.

16. Initiate self -- from LU to SSCP (Logical unit initiated logon only)

The 'initiate self' command is received from the polled physical unit and placed on the CUB link inbound queue. The CUB link inbound task dequeues the PIU, converts the FID2 or FID3 to FID1, and determines whether the PIU is from a defined LU which has a LU/SSCP session. The PIU is XPORTed to the channel queue to be sent to SSCP. No processing occurs. The host receives the PIU and processes the request. The request unit (RU) contains (1) the name of the application (VTAM APPL statement label, TCAM message handler label) to which logical unit wants to be connected, or (2) text used as an entry to the interpret table.

The 'initiate self' is required only if the connection is initiated from the network logical unit. A host application initiates the connection with a 'bind' command.

17. Bind command -- host application to LU

The 'bind' command is sent from the host application to the APPL/LU process queue of the logical unit block (LUB). The APPL/LU process queue task dequeues the request, sets to 1 the 'processing bind' bit of LUBAPSET of the LUB, converts the FID1 to FID2 or FID3, and places the PIU on the CUB link outbound queue for the link scheduler to transmit.

The response to 'bind' command is received and queued on the CUB link inbound queue. The CUB link inbound task dequeues the FID2 or FID3, converts it to a FID1, and branches to the APPL/LU CPM-IN for processing. If the response is positive, the 'processing bind' is set to 0 and 'session established' bit is set to 1 in the LUBAPSET field of the
LUB. The response is sent to the host application by XPORTing it to the channel intermediate queue of the CHB or COB.

NOTE: This command completes the fourth and last level of sessions (application/LU).

18. Start data traffic -- from host application to LU

The 'start data traffic' command is required by specific logical unit types. If 'start data traffic' is not required, data and subsequent commands immediately follow the 'bind' command.

The 'start data traffic' and all subsequent commands and data transfers are placed on the LUB APPL/LU process queue for converting from FID1 to FID2 or FID3 and placed on the CUB link outbound queue for transmission to the SDLC terminal. If the device is a type 1 physical unit, the sequence number processing is performed by NCP. If the PIU is text, the PIU is checked for VPACING from the host, and for PACING control from NCP to the logical unit. Data traffic is also segmented as required by the MAXDATA operand of the PU.

All text and data from the logical unit are received and placed on the CUB link inbound queue, converted to FID1, processed to identify which logical unit (or the CUB) the FID1 is from to locate the LUB control block, and processed by type. A command or command response is processed as required and text is checked for PACING responses or requirements to carry a VPACING response to the host. After required processing, the PIU is placed on the channel queue for transmission to the host.

NOTE: Two things which may not be initially apparent may occur during data transfer:

1) An 'isolated pacing response' (IPR) is sent from devices to the NCP, or from the NCP to the host. These IPR responses have the FME/RRN bits of 0, but pacing bit of 1 to request more data. This condition occurs whenever a outbound PIU carries a pacing request without requiring an FME/RRN response.

2) A pacing response resets the pacing counter. Therefore, if pacing of (2,1) is coded, the first PIU carries the pacing request. If the response is returned before a second PIU is processed, the second PIU becomes a 'first' PIU (carrying the pacing request) because the pacing counter was reset. If data traffic is not grouped in some manner (as in chaining), it appears that pacing is (1,1).

This command completes the initialization of the session. The last level of session could be ended by a 'terminate self' from the network logical unit followed by an 'unbind' or an 'unbind' initiated by the host application. A new 'bind' from a different or the same host application could initiate a new fourth level session without ending other levels. The switched support requires a full sequence of 'unbind', 'deactivate logical', 'free network addresses' (free LUBs to LUPOOL and clear LUV pointers), 'deactivate physical', 'discontact' (which sends SDLC 'set normal response mode' (SDRM) for a 'nonsequenced
acknowledgment' (NSA), and 'abandon connection'; and then a new 'dial' or 'answer' command may be issued for that switched interface.

A PIU trace provides the above sequence, and a formatted control block dump of NCP provides the bit settings to identify the levels of commands in process or completed.

19. Inoperative -- from NCP physical services to SSCP

The 'inoperative' command may be required at any point in the command sequences after the 'activate link' command. After the 'activate link' command, an 'answer', 'dial', or (nonswitched link) 'contact' command is issued by SSCP. If the request is for a valid network address in proper sequence, that command is immediately acknowledged with a positive response. The method of indicating an abnormal end or break in the processing on a link is for NCP physical services to send an 'inoperative' command to SSCP.

The 'inoperative' command identifies the network address in the request unit (RU) field of RU1NA of the link or resource. If the current command is to the link, the link address is carried in the request unit (RU) field of RU1NA and byte 5 of the RU contains a value of X'02'. If the current command is to a resource on the link, the resource network address is in RU1NA and byte 5 of the RU contains a X'01'.

No response is requested from SSCP; however, the host is expected to provide a sequence of commands to terminate, retry, or alternate path alternatives to the failing resource.
Appendix B

Problem Solutions

The following answers to the problems are based upon the generated values in the NCP dump listing in Appendix C.

Network Control
Program Supervisor

1. No program levels were active.

2. General register 0 of groups 0, 1, and 2 are a halfword beyond an EXIT instruction, X'B840'. Group 3 register 0 is a fullword beyond the X'B840' and SVC qualifier; however, the true indicator of an active level 5 task is at X'0685', bit 1.

Channel Adapter IOS

1. The channel block is at X'6F48'. The halfword address pointer is at X'0772'. The address is at offset X'00' of the control block; the prefix area begins at X'6F18'.

2. A CHB, identified by XXCXTCHB (or XXCXTCOB for COB) at the address minus 8 of the channel control block (X'6F40').

3. Yes, at X'6F30' (CHB-X'I8') in the channel hold queue is the last buffer sent to the host. No buffers are in the intermediate queue.

4. 04, from X'6FAO' at CHB plus X'58'.

5. 20 (X'0014'), from X'6FB8' at CHB plus X'70'.

6. 28 (X'001C'), from X'6FBC' at CHB plus X'74'.

7. 156 (X'009C'), from X'6FB4' at CHB plus X'6C'.

Path Control

1. X'17A40', the SIT is immediately before the SVT, with a count of entries equal to the count of the maximum subareas plus 1. The address of the SIT is in the extended halfword direct addressables (HWE) at offset X'48'.

2. X'17A50', the SVT is immediately before the RVT, with a word of zeros as the first entry. All nonfirst entries are nonzero. The SVT end delimiter is a X'FF' in the leftmost byte of the last entry. The address of the SVT is in the extended halfword direct addressables (HWE) at offset X'4C'.

3. X'17A64', the address of the RVT minus 2, is at X'07E8'. The minus 4 offset of the RVT address is at X'17A60'.

4. 4, from the mask at X'0693' or X'0694'.

5. X'65', from the RVT count field at X'17A60'.
6. 3, from the leftmost bits used for identifying a subarea based on the
network address of any resource, such as the PSB plus X'24', SCB or
CUB plus X'1C', or LUB plus X'24'.

7. A contact command for a type 4 PU is the same as for a type 1 or 2 PU.
The command is addressed to NCP physical services via SIT, SVT, and
RVT with the type 4 PU 'local' RVT address in the RU1NA field.

8. An 'activate physical' command for a type 4 PU is addressed to the
NCP Physical Services of the 'remote' NCP, using SIT and SVT control
blocks to locate the SCB link outbound queue.

**Network Control Program Physical Services**

1. The PSB is at X'19C60'. The address is the first RVT entry.
2. NCP04M from X'19C8C', PSB plus X'2C'.
3. Yes, bit 0 is 1 at X'19C8A', PSB plus X'2A'.
4. Yes, bit 1 is 1 at X'19C8A', PSB plus X2A'.
5. 8, from X'19C88', PSB plus X'28'.
6. X'17', from address X'19C82', PSB plus X'22'. The counter contains
   the sequence number previously received (X'0016').
7. No, from offset 0 of the PSB at X'19C60'.
8. The following identifies the buffer address and command within the
   buffer:

   X'19CA8'  'Initialization complete' from NCP to SSCP
   X'19CF4'  'Activate physical' to NCP from SSCP
   X'19D40'  'Start data traffic' to NCP from SSCP
   X'19D8C'  'Set control vector' (time and date, RU byte 5, 01)
   X'19DD8'  'Set control vector' (channel delay to 0, RU byte 5, 05)
   X'19E24'  'Activate link' (network address X'3001')
   X'1A628'  'Activate link' (network address X'303C')
   X'1A674'  'Answer' (network address X'303C')
   X'1A6C0'  'Set control vector', channel delay reset (RU byte 5, 05)
   X'1A70C'  'Contact' (network address X'3019')
   X'1A758'  'Contact' (network address X'302D')
   X'1A7A4'  'Contact' (network address X'302F')
   X'1A8D4'  'Contacted' (network address X'302F')

9. The first buffer contains an 'activate link' response. X'1A038' in RH
   byte 0 has a value of X'8F' with bit 5 indicating sense data is included.
   The four bytes of sense data are inserted following the RH and prior to
the RU. The system sense data of X'8002' is specified in *IBM 3704 and 3705 Program Reference Handbook* (GY30-3012), Section 8: NCP# Exception Responses as Path error: link failure. The network address of the link was X'3005' on the request, but has been overlaid in the response. A PIU trace would show sequence number X'0005' outbound with the network address and the response could be associated by the sequence number.

X'1A168' is an additional indicator of the link failure. The record maintenance statistics identifies the network address of X'3005' as the failing resource.

**Boundary Network Node**

1. The CUBs at X'1836C' and X'1879C' have outstanding 'set normal response mode' (SNRM) from a 'contact' command. At offset X'1F' is a value of X'41'; the 4 is the 'set normal response mode' bit.

2. The CUBs at X'18830' and X'18C1C' are the CUBs with an established SSCP/PU session. At offset X'4C' is a value of X'80'; the leftmost bit is the 'session established' bit.

3. At CUB plus X'18' is the SDLC addressing/polling character. The CUB at X'18830' has an SDLC addressing/polling character of X'C5'. The CUB at X'18C1C' has an SDLC addressing/polling character of X'C4'.

4. The network address is at the CUB plus X'1C'. The network address of the CUB at X'18830' is X'302F', subarea 3 and element 2F. The network address of the CUB at X'18C1C' is X'303D', subarea 3 and element 3D.

5. The LUB of the X'18C1C' CUB is at addresses X'18CEC'. The LUB pointer can be located by the address at CUB plus X'54', at X'18C70'. The value X'14018C74' specifies a maximum of X'14' entries in the LUV at X'18C74'. The only active LUB in the LUV table is at X'18CEC'.

6. Yes, the LUB has a session with SSCP with a X'80' value at offset X'28' at address X'18D14'.

7. No, the LUB does not have a host/LU, indicated by a value of X'00' at address X'18D18'.

8. Pacing is (2,2), (1,1), and (7,7); the pacing values are at offset X'2E' and X'2F'.

9. The identifier of the LUB local address is at offset X'31'. The local address of the LUBs are 1, 2, and 3.

10. To be understood, the following buffers must be grouped according to channel allocation. In groups of four buffers per allocation, the first channel buffers are: X'1A758', X'1A920', X'1AA50', X'1AC18', and X'1AD94'.

The following entries identify the buffer address and command:

- X'1A7F0' 'Activate physical', network address X'302F'
Appendix B

X'1A83C'  ‘Activate Logical’ (network address X'3030')
X'1A920'  ‘Activate Logical’ (network address X'3031')
X'1A96C'  ‘Activate Logical’ (network address X'3032')
X'1A9B8'  ‘Activate Logical’ (network address X'3033')
X'1AA04'  ‘Activate Logical’ (network address X'3034')
X'1AA50'  ‘Activate Logical’ (network address X'3035')
X'1AA9C'  ‘Activate Logical’ (network address X'3036')
X'1AAE8'  ‘Activate Logical’ (network address X'3037')
X'1AB34'  ‘Bind’ (network address X'3030')
X'1AB80'  Unformatted user data (network address X'3030')
X'1ABCC'  Unformatted user data (network address X'3030')
X'1AC18', X'1AC64', X'1ACB0', and X'1ACFC'  Unformatted user data, network address X'3030'. This is one PIU in four buffers. This is the first PIU transmitted to the terminal. The format is FID2 (buffer offset X'12'). The terminal type is an SDLC 3270 which requires NCP to process all commands. Therefore, the previous commands were all in FID1 format with the response in the same buffer used for the request.

11. The following buffers provide the sequence for a switched SDLC sequence.

X'1A628'  ‘Activate link’ (network address X'303C')
X'1A674'  ‘Answer’ (network address X'303C')
X'1B040'  ‘Off hook’ (network address X'303C')
X'1ADE0'  ‘Set control vector’ (network address X'303D')
X'1AE2C'  ‘Contact’ (network address X'303D')
X'1B0D8'  ‘Contacted’ (network address X'303D')
X'1AE78'  ‘Activate physical’ (network address X'303D', buffer offset X'0E' for FID2)
X'1B254'  ‘Activate physical’ response
X'1B124'  ‘Assign network addresses’ (to NCP physical services). The CUB is identified at offset X'1E' as X'303D'. At offset X'20' is a value of 01 to specify only one address being assigned. At offset X'22' is the assigned network address of X'303E'. This is a very unusual network address assignment of consecutive network addresses. The last defined CUB in the generation was the switched CUB with address X'303D', and the first available LUB in the LUPOOL had an address of X'303E'.
X'1B170'  ‘Set control vector LU’ (network address X'303E')
Local/Remote Link

1. Type 4 physical unit.
2. Station control block (SCB).
3. ‘Nonsequenced acknowledgement’ (NSA) or ‘request initialization’ (RQI).

Data Link Control

1. Activate link to NCP physical services with the link network address in the RU at RUINA.
2. (1) The SERVLIM operand specifies the number of passes through the service order table (SOT) before suspending data transfer for command processing. (2) If the pass through the service order table occurs in less time than the value specified in the PAUSE operand, link service is suspended until the pause expires or until outbound data is queued for transmission.
3. In the character control block (CCB) at CCBL2 (offset X'24').
4. In the character control block (CCB) at CCBL3 (offset X'4C').
5. In the link XIO block (LXB) at offset X'20'.
6. In the link XIO block (LXB) at offset X'1C'.
7. The active links are LKBs at X'18B0C', X'18BE8' and X'18CC4', identified at offset X'12' with a value of X'80'.
8. The switched links are the LKBs at X'18BE8' and X'18CC4', identified at the offset X'13' bit value of x1xx xxxx.
9. Half-duplex or full-duplex (one or two scanner addresses) for the link is identified in the ACB (CCB) at offset X'53', CCBTYPE. A value of x1xx xxxx specifies two line adapters. The LKBs at X'18BE8' and X'118CC4 are half-duplex. The LKBs at X'18B0C' is full-duplex.

BSC/SS Processor

1. FIDO
2. BTU
3. (1) Device queue of a device block (DVB) or (2) the nondevice input queue (pointer at HWE plus X'2C').
4. Seven lines, indicated by the leftmost bit of 1 in the resource vector table.
5. Fifteen devices, indicated by the second bit of 1 in the resource vector table.
6. The buffers are grouped in INBFRS quantity (4) at the following buffer addresses:

X'19E24', X'19FA0, X'1A1B4', X'1A2E4' and X'1A414

Buffers X'19F54', X'1A0D0' through X'1A168' were not allocated initially to the channel. The buffers contain the following information:

- **X'19E24'** 'Activate link' (network address X'3001')
- **X'19E70'** 'Set mode' (network address X'3002')
- **X'19EBC'** Allocated for input on the DVB work QCB for resource X'3002'. The DVB at X'17C48' contains a shifted address of X'67AF', or a nonshifted value of X'19EBC'. The chain pointer of buffer X'19EBC' contains X'0000'.
- **X'19F08'** 'Activate link' (network resource X'3003')
- **X'19F54'** Allocated as the first buffer in the system save area pool. At address X'075C' in the halfword direct addressables (XDH) is a value of X'67D5' shifted address of buffer X'19F54'.
- **X'19FA0'** 'Set mode' (network resource X'3004')
- **X'19FEC'** Allocated for input on the DVB work QCB for resource X'3004'
- **X'1A038'** 'Activate link' (network resource X'3005'). This command failed as indicated exception response in RH byte 1 xxx1xxxx. Sense data is included as indicated in RH byte 0 xxxxxx1xx. Four bytes of sense data is inserted following the RH. The RU is offset four bytes. This buffer contains the response, which does not include the resource identifier. In a PIU trace the request and response is associated by the sequence number field. A 'record maintenance statistics' PIU is generated by the NCP for SSCP to record the failure (see buffer X'1A168').
- **X'1A084'** 'Activate link' (network resource X'3008')
- **X'1A0D0'** Work buffer for 'activate link' failure (buffer X'1A038').
- **X'1A11C'** Work buffer for 'activate link' failure (buffer X'1A038')
- **X'1A168'** 'Record maintenance statistics for 'activate link' failure initiated in buffer X'1A038' for network resource X'3005'.
- **X'1A1B4'** 'Set mode' (network resource X'3009')
- **X'1A200'** Allocated for input on the DVB work QCB for resource X'3009'
Appendix B

X'1A24C'  'Activate line' (network resource X'300A')
X'1A298'  'Set mode' (network resource X'300B')
X'1A2E4'  BSC/SS 'contact' command (network resource X'300B')
X'1A330'  'Activate link' (network resource X'300C')
X'1A37C'  'Set mode' (network resource X'300D')
X'1A3C8'  BSC/SS 'invite' command in BCU (BTU) format (network resource X'300D').
X'1A414'  'Activate link' (network resource X'300E'). This command failed as indicated exception response in RH byte 1 xxx1xxx. Sense data is included as indicated in RH byte 0 xxxx1xx.

7. The buffers contain the following information:

X'1B208'  Unformatted user data, attempted logon from X'300B'
X'1B2EC'  'Reset immediate' (network resource X'300B')
X'1B338'  'Write with end of transmission' (network resource X'300B')
X'1B384'  'Read transmission' (network resource X'300B')
X'1B3D0'  'Write with end of transmission' (network resource X'300B')
X'1B4B4'  'Record maintenance statistics' (network resource X'300B')
THE FOLLOWING IS A DUMP OF A LOCAL NCP

PSB (PHYSICAL SERVICES BLOCK) 19C60

00000000 00A80000 0080F5D0 00000000 00000000 00A80000 0880F460 00000000
00060016 30001000 0008C000 D5C3D7F0 F4D4D450 00000000 02000000 00000000

SIT (SUBAREA INDEX TABLE) 17A40

00020001 00000000 00000000 00000000

SVT (SUBAREA VECTOR TABLE) 17A50

00000000 80A17A60 49E06F18 FF040404 00650016 00219C60

NCP Dump Listing

Appendix C
<table>
<thead>
<tr>
<th>LCB</th>
<th>17C94</th>
<th>0001</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>00000000 1C800000 0080FC0 00000000</td>
<td>00000000 00000000 18A80000 0A40D3B0</td>
</tr>
<tr>
<td></td>
<td>00000000 00007210 8001028C 61017C48</td>
<td>00000000 01100000 01000200 01000000</td>
</tr>
<tr>
<td></td>
<td>00013001 40000000 00017C49 67FB67PB</td>
<td>03A90000 00000000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ACB</th>
<th>07210</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>008D0082 00000000 00000000 20000000</td>
</tr>
<tr>
<td></td>
<td>FF000000 4C901680 793F0980 0000C800</td>
</tr>
<tr>
<td></td>
<td>04000000 90000000 02800000 4D500300</td>
</tr>
<tr>
<td></td>
<td>008D0082</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LGT</th>
<th>07760</th>
</tr>
</thead>
<tbody>
<tr>
<td>1D2A6A86 6A2C6548 CECA8AC0 4847100A</td>
<td>00007C03 02020000 1C261F37 7C342P2F</td>
</tr>
<tr>
<td>626D405D FF3D003E 010D2C5E 370E004F</td>
<td>4C7F6114</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LCB</th>
<th>17D2C</th>
<th>0003</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>00000000 1C800000 0080FC0 00000000</td>
<td>00000000 00000000 18A80000 0A40D3B0</td>
</tr>
<tr>
<td></td>
<td>00000000 00007270 8001028C 61017C40</td>
<td>00000000 01100000 01000200 01000000</td>
</tr>
<tr>
<td></td>
<td>00013003 40000000 00017C49 62920000</td>
<td>00017D94 00017DC0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ACB</th>
<th>07270</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>008D0082 00000000 00000000 00000000</td>
</tr>
<tr>
<td></td>
<td>FF000000 4C901680 793F0980 0000C800</td>
</tr>
<tr>
<td></td>
<td>04000000 90000000 02800000 4D500300</td>
</tr>
<tr>
<td></td>
<td>008D0000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LGT</th>
<th>07760</th>
</tr>
</thead>
<tbody>
<tr>
<td>1D2A6A86 6A2C6548 CECA8AC0 4847100A</td>
<td>00007C03 02020000 1C261F37 7C342P2F</td>
</tr>
<tr>
<td>626D405D FF3D003E 010D2C5E 370E004F</td>
<td>4C7F6114</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LCB</th>
<th>17F18</th>
<th>0005</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>00000000 1C800000 0040FC0 00000000</td>
<td>00000000 00000000 08A80000 0A40D3B0</td>
</tr>
<tr>
<td></td>
<td>00000000 0000727D 10001490 42000000</td>
<td>00000000 01200000 16002000 06010000</td>
</tr>
<tr>
<td></td>
<td>00033005 00000000 00000000 0017D7B8</td>
<td>02030200 00002000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ACB</th>
<th>072D0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>008D0000 000006F4 00000000 00000000</td>
</tr>
<tr>
<td></td>
<td>FF000000 3F5F1680 794501FF 00004800</td>
</tr>
<tr>
<td></td>
<td>04000000 04000000 03000000 4C040800</td>
</tr>
<tr>
<td></td>
<td>6A666A2C</td>
</tr>
</tbody>
</table>

<p>| LGT | 0732E |</p>
<table>
<thead>
<tr>
<th>LCB</th>
<th>07EC0</th>
<th>0008</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0000000 1CA80000 0080CF0C 00000000</td>
<td>00000000 00000000 18A80000 0A40D3B0</td>
</tr>
<tr>
<td></td>
<td>00000000 00007369 88012B8C 41101774</td>
<td>00000000 01100000 00000000 00000000</td>
</tr>
<tr>
<td></td>
<td>00013000 40000000 00177674 00000000</td>
<td>80A00000 00000000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ACB</th>
<th>07364</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0080D080 00000000 00000000 00000000</td>
</tr>
<tr>
<td></td>
<td>FF000000 4C9016C0 794B0380 0000C80B</td>
</tr>
<tr>
<td></td>
<td>04000000 0A000F00 2B800000 4D500200</td>
</tr>
<tr>
<td></td>
<td>6B026B02</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LGT</th>
<th>073C2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>002A6B02 6B026B02 C0C08AC0 48471006</td>
</tr>
<tr>
<td></td>
<td>6E6D405D FF3D003E 7C0D2C3D 7C0E004F</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LCB</th>
<th>17P58</th>
<th>000A</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>6D66E966 1CA80000 0080CF0C 00000000</td>
<td>00000000 00000000 00000000 00000000</td>
</tr>
<tr>
<td></td>
<td>00000000 00007369 88012B8C 41101774</td>
<td>00000000 00400000 01000000 00000000</td>
</tr>
<tr>
<td></td>
<td>00003000 50000000 00177674 00000000</td>
<td>00A00000 00000000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ACB</th>
<th>073F8</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>00282850 0000000C 00000000 1E33B066</td>
</tr>
<tr>
<td></td>
<td>00000000 00007369 88012B8C 41101774</td>
</tr>
<tr>
<td></td>
<td>00003000 50000000 00177674 00000000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LGT</th>
<th>07456</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>002A6B02 6B026B02 C0C08AC0 48471006</td>
</tr>
<tr>
<td></td>
<td>6E6D405D FF3D003E 7C0D2C3D 7C0E004F</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LCB</th>
<th>17FF0</th>
<th>000C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>00000000 1CA80000 0080CF0C 00000000</td>
<td>00000000 00000000 18A80000 0A40D3B0</td>
</tr>
<tr>
<td></td>
<td>00000000 0000748C 88012B8C 41101774</td>
<td>00000000 01280000 01000000 00010000</td>
</tr>
<tr>
<td></td>
<td>00013000 40000000 00177674 00000000</td>
<td>00180000 0001827C</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ACB</th>
<th>07548C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>008D0B00 00000000 00000000 00000000</td>
</tr>
<tr>
<td></td>
<td>FF000000 4C9016C0 72100680 0000C80B</td>
</tr>
</tbody>
</table>

---

Appendix C
<table>
<thead>
<tr>
<th>Address</th>
<th>Memory</th>
<th>Address</th>
<th>Memory</th>
<th>Address</th>
<th>Memory</th>
<th>Address</th>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>04000000</td>
<td>0B000000</td>
<td>2B800000</td>
<td>4D500020</td>
<td>217D0600</td>
<td>00000100</td>
<td>0A500000</td>
<td>0100222A</td>
</tr>
<tr>
<td>6B5A6BA2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LGT 074EA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22A6B5A</td>
<td>6B26BC2</td>
<td>C0C08AC0</td>
<td>78771106</td>
<td>79D8FP01</td>
<td>00000000</td>
<td>1C381F84</td>
<td>7C808889</td>
</tr>
<tr>
<td>8A8D405D</td>
<td>803DE085</td>
<td>910D9391</td>
<td>9140B100</td>
<td>3400008D</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LCB 182F8</td>
<td>0000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00000000</td>
<td>1CA80000</td>
<td>0040CFC0</td>
<td>00000000</td>
<td>00000000</td>
<td>00000000</td>
<td>09480000</td>
<td>0A40D3B0</td>
</tr>
<tr>
<td>00000000</td>
<td>00000000</td>
<td>00000000</td>
<td>46000000</td>
<td>00000000</td>
<td>01200000</td>
<td>16000300</td>
<td>04010000</td>
</tr>
<tr>
<td>0000300E</td>
<td>00000000</td>
<td>00A00000</td>
<td>0201803C</td>
<td>02090800</td>
<td>00000800</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACP 0751C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>008D0000</td>
<td>000006F4</td>
<td>00000000</td>
<td>00000000</td>
<td>00000000</td>
<td>00000000</td>
<td>608E0000</td>
<td>001803C</td>
</tr>
<tr>
<td>00000000</td>
<td>31C21640</td>
<td>793301FF</td>
<td>00004800</td>
<td>0888F400</td>
<td>75780000</td>
<td>000009C</td>
<td>00000000</td>
</tr>
<tr>
<td>04000000</td>
<td>008C0000</td>
<td>06000000</td>
<td>4C040700</td>
<td>38CD001E</td>
<td>00002000</td>
<td>00000000</td>
<td>4C1666D6</td>
</tr>
<tr>
<td>67CC6E48</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LGT 07578</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4C1666D6</td>
<td>67CC6E48</td>
<td>8D488C6</td>
<td>C851A07</td>
<td>79F8AA03</td>
<td>070F3202</td>
<td>10261037</td>
<td>1002101F</td>
</tr>
<tr>
<td>10701061</td>
<td>107C102D</td>
<td>3B111003</td>
<td>10688000</td>
<td>00300000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Address</td>
<td>Value 1</td>
<td>Value 2</td>
<td>Value 3</td>
<td>Value 4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00000000</td>
<td>00000000</td>
<td>00000000</td>
<td>00000000</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0480C810</td>
<td>00000000</td>
<td>00000000</td>
<td>00000000</td>
<td>3011087</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0000100</td>
<td>12000540</td>
<td>40C1C12D</td>
<td>00056060</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* ...*
<table>
<thead>
<tr>
<th>CUB</th>
<th>1836C</th>
<th>0019</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0000000 0080000 008103F0 0000000</td>
<td>00000000 00000000 C901880C 3019A141</td>
</tr>
<tr>
<td></td>
<td>0000000 02000000 00000000 000000700</td>
<td>00000000 00000000 00A0000 00000000</td>
</tr>
<tr>
<td></td>
<td>00880000 08180000 00000000 00000000</td>
<td>01090000 00000000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CUB</th>
<th>1879C</th>
<th>0020</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0000000 0080000 008103F0 0000000</td>
<td>00000000 00000000 C201880C 302DA141</td>
</tr>
<tr>
<td></td>
<td>0000000 04000000 00000000 000000100</td>
<td>0001000 00000000 00000000 00000000</td>
</tr>
<tr>
<td></td>
<td>00A80000 1E810ADD 00000000 000000300</td>
<td>01090000 00000000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CUB</th>
<th>18A30</th>
<th>002F</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0000000 00A80000 008103F0 0000000</td>
<td>00000000 00000000 C501880C 302DP001</td>
</tr>
<tr>
<td></td>
<td>0000000 02000000 88440000 00000700</td>
<td>00C0000 00000000 00000000 00000000</td>
</tr>
<tr>
<td></td>
<td>00A80000 1E810ADD 00000000 00800300</td>
<td>01090000 00000000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CUB</th>
<th>18A84</th>
<th>0039</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0000000 00A80000 008103F0 0000000</td>
<td>00000000 00000000 C301880C 3038A001</td>
</tr>
<tr>
<td></td>
<td>0000000 02000000 00000000 00000100</td>
<td>0001000 00000000 00000000 00000000</td>
</tr>
<tr>
<td></td>
<td>00A80000 1E810ADD 00000000 00000300</td>
<td>01090000 00000000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CUB</th>
<th>18B40</th>
<th>003B</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0000000 00A80000 008103F0 0000000</td>
<td>00000000 00000000 F3018BES 303BA001</td>
</tr>
<tr>
<td></td>
<td>00800000 24000000 00000000 00000000</td>
<td>0001000 00000000 00000000 00000000</td>
</tr>
<tr>
<td></td>
<td>00A80000 1E810ADD 00000000 00000000</td>
<td>00000000 14018998</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LUVT</th>
<th>18B98</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0000000 00000000 00000000 00000000</td>
<td>00000000 00000000 00000000 00000000</td>
</tr>
<tr>
<td></td>
<td>0000000 00000000 00000000 00800000</td>
<td>00000000 00000000 00000000 00000000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CUB</th>
<th>18C1C</th>
<th>003D</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0000000 00A80000 008103F0 0000000</td>
<td>00000000 00000000 C4018CC4 303D0001</td>
</tr>
<tr>
<td></td>
<td>00810002 22000000 44440000 00140700</td>
<td>00407000 00000000 00000000 00000000</td>
</tr>
<tr>
<td></td>
<td>00A80000 1E810ADD 00000000 80000300</td>
<td>01090000 14018C74</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LUVT</th>
<th>18C74</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>01418C2C 00000000 00000000 00000000</td>
<td>00000000 00000000 00000000 00000000</td>
</tr>
<tr>
<td></td>
<td>0000000 00000000 00000000 00800000</td>
<td>00000000 00000000 00000000 00000000</td>
</tr>
<tr>
<td>LUB</td>
<td>183C0</td>
<td>001A</td>
</tr>
<tr>
<td>------</td>
<td>--------</td>
<td>----------</td>
</tr>
<tr>
<td></td>
<td>0000000</td>
<td>00A8000</td>
</tr>
<tr>
<td>LUB</td>
<td>183F4</td>
<td>001B</td>
</tr>
<tr>
<td></td>
<td>0000000</td>
<td>00A8000</td>
</tr>
<tr>
<td>LUB</td>
<td>18428</td>
<td>001C</td>
</tr>
<tr>
<td></td>
<td>0000000</td>
<td>00A8000</td>
</tr>
<tr>
<td>LUB</td>
<td>1845C</td>
<td>001D</td>
</tr>
<tr>
<td></td>
<td>0000000</td>
<td>00A8000</td>
</tr>
<tr>
<td>LUB</td>
<td>18490</td>
<td>001E</td>
</tr>
<tr>
<td></td>
<td>0000000</td>
<td>00A8000</td>
</tr>
<tr>
<td>LUB</td>
<td>184C4</td>
<td>001F</td>
</tr>
<tr>
<td></td>
<td>0000000</td>
<td>00A8000</td>
</tr>
<tr>
<td>LUB</td>
<td>184F8</td>
<td>0020</td>
</tr>
<tr>
<td></td>
<td>0000000</td>
<td>00A8000</td>
</tr>
<tr>
<td>LUB</td>
<td>1852C</td>
<td>0021</td>
</tr>
<tr>
<td></td>
<td>0000000</td>
<td>00A8000</td>
</tr>
<tr>
<td>LUB</td>
<td>18560</td>
<td>0022</td>
</tr>
<tr>
<td></td>
<td>0000000</td>
<td>00A8000</td>
</tr>
<tr>
<td>LUB</td>
<td>18594</td>
<td>0023</td>
</tr>
<tr>
<td></td>
<td>0000000</td>
<td>00A8000</td>
</tr>
<tr>
<td>LUB</td>
<td>185C8</td>
<td>0024</td>
</tr>
<tr>
<td></td>
<td>0000000</td>
<td>00A8000</td>
</tr>
<tr>
<td>Location</td>
<td>Address</td>
<td>Data</td>
</tr>
<tr>
<td>----------</td>
<td>---------</td>
<td>------</td>
</tr>
<tr>
<td>LUB 185FC</td>
<td>0025</td>
<td>00000000 00A80000 00810A00 00000000 0001836C 302A0000 00000000 0000101</td>
</tr>
<tr>
<td>LUB 18630</td>
<td>0026</td>
<td>00000000 00A80000 00810A00 00000000 0001836C 30250000 00000000 0000101</td>
</tr>
<tr>
<td>LUB 18664</td>
<td>0027</td>
<td>00000000 00A80000 00810A00 00000000 0001836C 30260000 00000000 0000101</td>
</tr>
<tr>
<td>LUB 18698</td>
<td>0028</td>
<td>00000000 00A80000 00810A00 00000000 0001836C 30270000 00000000 0000101</td>
</tr>
<tr>
<td>LUB 186CC</td>
<td>0029</td>
<td>00000000 00A80000 00810A00 00000000 0001836C 30290000 00000000 0000101</td>
</tr>
<tr>
<td>LUB 18700</td>
<td>002A</td>
<td>00000000 00A80000 00810A00 00000000 0001836C 302A0000 00000000 0000101</td>
</tr>
<tr>
<td>LUB 18734</td>
<td>002B</td>
<td>00000000 00A80000 00810A00 00000000 0001836C 302B0000 00000000 0000101</td>
</tr>
<tr>
<td>LUB 18768</td>
<td>002C</td>
<td>00000000 00A80000 00810A00 00000000 0001836C 302C0000 00000000 0000101</td>
</tr>
<tr>
<td>LUB 18770</td>
<td>002E</td>
<td>00000000 00A80000 00810A00 00000000 0001879C 302E0000 00000000 0000101</td>
</tr>
<tr>
<td>LUB 18884</td>
<td>003C</td>
<td>00000000 00A80000 00810A00 00000000 00018830 30300000 8001001 80090101</td>
</tr>
<tr>
<td>LUB</td>
<td>188C4</td>
<td>0031</td>
</tr>
<tr>
<td>------</td>
<td>--------</td>
<td>------</td>
</tr>
<tr>
<td></td>
<td>00000000 00A80000 00810A00 00000000</td>
<td>00020001 00000000 00000000 00000000</td>
</tr>
</tbody>
</table>
|      | 0018830  30310000 80000000 00000101 | * ············································
| LUB  | 18904  | 0032 |
|      | 00000000 00A80000 00810A00 00000000 | 00030001 00000000 00000000 00000000 |
|      | 0018830  30320000 80000000 00000101 | * ············································
| LUB  | 18944  | 0033 |
|      | 00000000 00A80000 00810A00 00000000 | 00040001 00000000 00000000 00000000 |
|      | 0018830  30330000 80000000 00000101 | * ············································
| LUB  | 18964  | 0034 |
|      | 00000000 00A80000 00810A00 00000000 | 00050001 00000000 00000000 00000000 |
|      | 0018830  30340000 80000000 00000101 | * ············································
| LUB  | 189C4  | 0035 |
|      | 00000000 00A80000 00810A00 00000000 | 00060001 00000000 00000000 00000000 |
|      | 0018830  30350000 80000000 00000101 | * ············································
| LUB  | 18A04  | 0036 |
|      | 00000000 00A80000 00810A00 00000000 | 00070001 00000000 00000000 00000000 |
|      | 0018830  30360000 80000000 00000101 | * ············································
| LUB  | 18A44  | 0037 |
|      | 00000000 00A80000 00810A00 00000000 | 00080001 00000000 00000000 00000000 |
|      | 0018830  30370000 80000000 00000101 | * ············································
| LUB  | 18A88  | 0039 |
|      | 00000000 00A80000 00810A00 00000000 | 00090001 00000000 00000000 00000000 |
|      | 0018830  30390000 80000000 00000101 | * ············································
| LUB  | 18C2C  | 0032 |
|      | 00000000 00A80000 00810A00 00000000 | 00100000 00000000 00000000 00000000 |
|      | 0018C1C  30320000 80000000 00000101 | * ············································
| LUB  | 18D2C  | 0033 |
|      | 00000000 00A80000 00810A00 00000000 | 00110000 00000000 00000000 00000000 |
|      | 0018D00  30330000 80000000 00000101 | * ············································
| LUB  | 18D6C  | 0040 |
|      | 00000000 00A80000 00810A00 00000000 | 00120000 00000000 00000000 00000000 |
|      | 0018D00  30340000 80000000 00000101 | * ············································
|      | 00000000 00A80000 00810A00 00000000 | 00120000 00000000 00000000 00000000 |
|      | 0018D00  30350000 80000000 00000101 | * ············································
<p>|      | 00000000 00A80000 00810A00 00000000 | 00120000 00000000 00000000 00000000 |</p>
<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Addr 1</th>
<th>Addr 2</th>
<th>Address 3</th>
<th>Address 4</th>
<th>Address 5</th>
<th>Address 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>18DEC</td>
<td>0041</td>
<td>0000000 00810A00 00000000 00000000</td>
<td>0000000 00810A00 00000000 00000000</td>
<td><em>:--------------------------------------</em></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18DEC</td>
<td>0042</td>
<td>0000000 00810A00 00000000 00000000</td>
<td>0000000 00810A00 00000000 00000000</td>
<td><em>:--------------------------------------</em></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18E2C</td>
<td>0043</td>
<td>0000000 00810A00 00000000 00000000</td>
<td>0000000 00810A00 00000000 00000000</td>
<td><em>:--------------------------------------</em></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18F6C</td>
<td>0044</td>
<td>0000000 00810A00 00000000 00000000</td>
<td>0000000 00810A00 00000000 00000000</td>
<td><em>:--------------------------------------</em></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18EAC</td>
<td>0045</td>
<td>0000000 00810A00 00000000 00000000</td>
<td>0000000 00810A00 00000000 00000000</td>
<td><em>:--------------------------------------</em></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18EEC</td>
<td>0046</td>
<td>0000000 00810A00 00000000 00000000</td>
<td>0000000 00810A00 00000000 00000000</td>
<td><em>:--------------------------------------</em></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18E2C</td>
<td>0047</td>
<td>0000000 00810A00 00000000 00000000</td>
<td>0000000 00810A00 00000000 00000000</td>
<td><em>:--------------------------------------</em></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18F6C</td>
<td>0048</td>
<td>0000000 00810A00 00000000 00000000</td>
<td>0000000 00810A00 00000000 00000000</td>
<td><em>:--------------------------------------</em></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18FAC</td>
<td>0049</td>
<td>0000000 00810A00 00000000 00000000</td>
<td>0000000 00810A00 00000000 00000000</td>
<td><em>:--------------------------------------</em></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18F2C</td>
<td>004A</td>
<td>0000000 00810A00 00000000 00000000</td>
<td>0000000 00810A00 00000000 00000000</td>
<td><em>:--------------------------------------</em></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
LUB 1902C 004B
0000000 0080000 00810A 00 0000000 0000000 0000000 00810760 00000000 *
0000000 3040000 0000000 0000000 0000000 0000000 0000000 00000000 *

LUB 1906C 004C
0000000 0080000 00810A 00 0000000 0000000 0000000 00810760 00000000 *
0000000 3040000 0000000 0000000 0000000 0000000 0000000 00000000 *

LUB 190AC 004D
0000000 0080000 00810A 00 0000000 0000000 0000000 00810760 00000000 *
0000000 3040000 0000000 0000000 0000000 0000000 0000000 00000000 *

LUB 190EC 004E
0000000 0080000 00810A 00 0000000 0000000 0000000 00810760 00000000 *
0000000 3040000 0000000 0000000 0000000 0000000 0000000 00000000 *

LUB 1912C 0050
0000000 0080000 00810A 00 0000000 0000000 0000000 00810760 00000000 *
0000000 3050000 0000000 0000000 0000000 0000000 0000000 00000000 *

LUB 1916C 0051
0000000 0080000 00810A 00 0000000 0000000 0000000 00810760 00000000 *
0000000 3050000 0000000 0000000 0000000 0000000 0000000 00000000 *

LUB 191AC 0052
0000000 0080000 00810A 00 0000000 0000000 0000000 00810760 00000000 *
0000000 3050000 0000000 0000000 0000000 0000000 0000000 00000000 *

LUB 191EC 0053
0000000 0080000 00810A 00 0000000 0000000 0000000 00810760 00000000 *
0000000 3052000 0000000 0000000 0000000 0000000 0000000 00000000 *

LUB 1922C 0054
0000000 0080000 00810A 00 0000000 0000000 0000000 00810760 00000000 *
0000000 3053000 0000000 0000000 0000000 0000000 0000000 00000000 *

LUB 1926C 0055
0000000 0080000 00810A 00 0000000 0000000 0000000 00810760 00000000 *
0000000 3054000 0000000 0000000 0000000 0000000 0000000 00000000 *
<table>
<thead>
<tr>
<th>Address</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LUB 192EC 0056</td>
<td>00000000 00A80000 00810A00 00000000</td>
<td>08810760 00000000</td>
</tr>
<tr>
<td>LUB 1932C 0057</td>
<td>00000000 00A80000 00810A00 00000000</td>
<td>08810760 00000000</td>
</tr>
<tr>
<td>LUB 1936C 0058</td>
<td>00000000 00A80000 00810A00 00000000</td>
<td>08810760 00000000</td>
</tr>
<tr>
<td>LUB 193AC 0059</td>
<td>00000000 00A80000 00810A00 00000000</td>
<td>08810760 00000000</td>
</tr>
<tr>
<td>LUB 193EC 005A</td>
<td>00000000 00A80000 00810A00 00000000</td>
<td>08810760 00000000</td>
</tr>
<tr>
<td>LUB 1942C 005B</td>
<td>00000000 00A80000 00810A00 00000000</td>
<td>08810760 00000000</td>
</tr>
<tr>
<td>LUB 1946C 005C</td>
<td>00000000 00A80000 00810A00 00000000</td>
<td>08810760 00000000</td>
</tr>
<tr>
<td>LUB 194AC 005D</td>
<td>00000000 00A80000 00810A00 00000000</td>
<td>08810760 00000000</td>
</tr>
<tr>
<td>LUB 194EC 005E</td>
<td>00000000 00A80000 00810A00 00000000</td>
<td>08810760 00000000</td>
</tr>
<tr>
<td>LUB 1952C 005F</td>
<td>00000000 00A80000 00810A00 00000000</td>
<td>08810760 00000000</td>
</tr>
<tr>
<td>LUB</td>
<td>1956C</td>
<td>0060</td>
</tr>
<tr>
<td>-------</td>
<td>--------</td>
<td>------</td>
</tr>
<tr>
<td></td>
<td>00000000 00A80000 00810A00 00000000</td>
<td>00000000 00A80000 08810760 00000000</td>
</tr>
<tr>
<td></td>
<td>00000000 30600000 00000000 00000000</td>
<td>00010000 00000000 00000000 00000000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LUB</th>
<th>195AC</th>
<th>0061</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>00000000 00A80000 00810A00 00000000</td>
<td>00000000 00A80000 08810760 00000000</td>
</tr>
<tr>
<td></td>
<td>00000000 30610000 00000000 00000000</td>
<td>00010000 00000000 00000000 00000000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LUB</th>
<th>195EC</th>
<th>0062</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>00000000 00A80000 00810A00 00000000</td>
<td>00000000 00A80000 08810760 00000000</td>
</tr>
<tr>
<td></td>
<td>00000000 30620000 00000000 00000000</td>
<td>00010000 00000000 00000000 30000000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LUB</th>
<th>1962C</th>
<th>0063</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>00000000 00A80000 00810A00 00000000</td>
<td>00000000 00A80000 08810760 00000000</td>
</tr>
<tr>
<td></td>
<td>00000000 30F30000 00000000 00000000</td>
<td>00010000 00000000 00000000 00000000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LUB</th>
<th>1966C</th>
<th>0064</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>00000000 00A80000 00810A00 00000000</td>
<td>00000000 00A80000 08810760 00000000</td>
</tr>
<tr>
<td></td>
<td>00000000 30640000 00000000 00000000</td>
<td>00010000 00000000 00000000 00000000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LUB</th>
<th>196AC</th>
<th>0065</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>00000000 00A80000 00810A00 00000000</td>
<td>00000000 00A80000 08810760 00000000</td>
</tr>
<tr>
<td></td>
<td>00000000 30650000 00000000 00000000</td>
<td>00010000 00000000 00000000 00000000</td>
</tr>
<tr>
<td>Page 86</td>
<td>3705 DUMP</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-----------</td>
<td></td>
</tr>
<tr>
<td>1BD60</td>
<td>00000000 00000000 00000000 00000000</td>
<td>00000000 00000000 00000000 00000000</td>
</tr>
<tr>
<td>1BDA0</td>
<td>00000000 00000000 00000000 00000000</td>
<td>00000000 00000000 00000000 00000000</td>
</tr>
<tr>
<td>LINE 1BDC0 SAME AS ABOVE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1BDE0</td>
<td>00000000 00000000 00000000 00000000</td>
<td>00000000 00000000 00000000 00000000</td>
</tr>
<tr>
<td>1BEE0</td>
<td>00000000 00000000 00000000 00000000</td>
<td>00000000 00000000 00000000 00000000</td>
</tr>
<tr>
<td>1BE20</td>
<td>00000000 00000000 00000000 00000000</td>
<td>00000000 00000000 00000000 00000000</td>
</tr>
<tr>
<td>1BE40</td>
<td>00000000 00000000 00000000 00000000</td>
<td>00000000 00000000 00000000 00000000</td>
</tr>
<tr>
<td>LINE 1BE60 SAME AS ABOVE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1BE80</td>
<td>6FB30000 00000000 0000178 BA200400</td>
<td>B9200080 7194870 7688B920 00220198</td>
</tr>
<tr>
<td>1BEA0</td>
<td>EF048520 A8028508 10819104 C5381D81</td>
<td>8500BF20 020011A8 A875716C F0069906</td>
</tr>
<tr>
<td>1BEC0</td>
<td>717C108 9806F130 98BCA813 6FC6F110</td>
<td>989C59C2 D0A512C 1189834 D088A850</td>
</tr>
<tr>
<td>1BEF0</td>
<td>2502488 220ABE20 00EC0398 80283710</td>
<td>4730887 018A16C 51A5081 9502102</td>
</tr>
<tr>
<td>1BF00</td>
<td>1189808 6188E106 BB2000D0 73942088</td>
<td>80015134 A858D520 6F9218E 5185E103</td>
</tr>
<tr>
<td>1BF20</td>
<td>2980BB20 80102381 21899108 2182A83E</td>
<td>BF200200 21027198 218221B8 6D181798</td>
</tr>
<tr>
<td>1BF40</td>
<td>C7B9B088 BB208220 D510A804 BB208000</td>
<td>2181282 A1888B20 90A02381 BB2C8760</td>
</tr>
<tr>
<td>1BF60</td>
<td>23859128 6FECB920 006C0198 A182D188</td>
<td>51048020 81008154 BF200400 57446188</td>
</tr>
<tr>
<td>1BF80</td>
<td>D1365174 A8CD719C F9101314 67887108</td>
<td>C10B138 D0025174 A8B1717C F161612C</td>
</tr>
<tr>
<td>1BFA0</td>
<td>D98C816 E89B920 00386174 B9200740</td>
<td>672A38FB 610CPO33 9815B920 0006B164</td>
</tr>
<tr>
<td>1BFC0</td>
<td>B9200F40 A8176788 EF04E886 A061F802</td>
<td>A865B20 20007594 23886188 99200142</td>
</tr>
<tr>
<td>1BF20</td>
<td>618A1382 8143B800 96C40000 00008925</td>
<td>138A3397 33983398 8806A05F 170107E7</td>
</tr>
<tr>
<td>CHAIN POINTER</td>
<td>1A168</td>
<td>1A084</td>
</tr>
<tr>
<td>---------------</td>
<td>-------</td>
<td>-------</td>
</tr>
<tr>
<td>19FA0</td>
<td>6850A1A4</td>
<td>00140001</td>
</tr>
<tr>
<td></td>
<td>60000008</td>
<td>50000000</td>
</tr>
<tr>
<td></td>
<td>C3E7C6C9</td>
<td>D5C9E340</td>
</tr>
<tr>
<td>19FE4</td>
<td>00000000</td>
<td>00000000</td>
</tr>
<tr>
<td></td>
<td>·········</td>
<td>·········</td>
</tr>
<tr>
<td>1A038</td>
<td>68210A14</td>
<td>00000000</td>
</tr>
<tr>
<td></td>
<td>020A0000</td>
<td>00000000</td>
</tr>
<tr>
<td></td>
<td>00000000</td>
<td>00000000</td>
</tr>
<tr>
<td>1A084</td>
<td>6860A010</td>
<td>00000000</td>
</tr>
<tr>
<td></td>
<td>00000000</td>
<td>00000000</td>
</tr>
<tr>
<td></td>
<td>00000000</td>
<td>00000000</td>
</tr>
<tr>
<td>1A000</td>
<td>68780000</td>
<td>0101A0D8</td>
</tr>
<tr>
<td></td>
<td>00000000</td>
<td>00000000</td>
</tr>
<tr>
<td></td>
<td>00000000</td>
<td>00000000</td>
</tr>
<tr>
<td>1A11C</td>
<td>68340000</td>
<td>0101A124</td>
</tr>
<tr>
<td></td>
<td>00000000</td>
<td>00000000</td>
</tr>
<tr>
<td></td>
<td>00000000</td>
<td>00000000</td>
</tr>
<tr>
<td>1A168</td>
<td>680E0A2B</td>
<td>0000680E</td>
</tr>
<tr>
<td></td>
<td>002A0005</td>
<td>00000000</td>
</tr>
<tr>
<td></td>
<td>00000000</td>
<td>00000000</td>
</tr>
<tr>
<td>1A1B4</td>
<td>68930A14</td>
<td>00000000</td>
</tr>
<tr>
<td></td>
<td>60000008</td>
<td>00000000</td>
</tr>
<tr>
<td></td>
<td>00000000</td>
<td>00000000</td>
</tr>
<tr>
<td>1A200</td>
<td>0001E000</td>
<td>00000000</td>
</tr>
<tr>
<td></td>
<td>00000000</td>
<td>00000000</td>
</tr>
<tr>
<td></td>
<td>00000000</td>
<td>00000000</td>
</tr>
<tr>
<td>1A24C</td>
<td>68A60A10</td>
<td>00000000</td>
</tr>
<tr>
<td></td>
<td>00000000</td>
<td>00000000</td>
</tr>
<tr>
<td></td>
<td>00000000</td>
<td>00000000</td>
</tr>
<tr>
<td>1A298</td>
<td>68CC0A14</td>
<td>00000000</td>
</tr>
<tr>
<td></td>
<td>1001300B</td>
<td>1001000A</td>
</tr>
<tr>
<td>ADDR</td>
<td>DATA</td>
<td>LENGTH</td>
</tr>
<tr>
<td>-------</td>
<td>------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>1A5DC</td>
<td>CHAIN POINTER 1A414</td>
<td>110</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1A628</td>
<td>CHAIN POINTER 1A674</td>
<td>110</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1A674</td>
<td>CHAIN POINTER 1A6CO</td>
<td>110</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1A6C0</td>
<td>CHAIN POINTER 1A70C</td>
<td>110</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1A70C</td>
<td>CHAIN POINTER 1A758</td>
<td>110</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1A758</td>
<td>CHAIN POINTER 1A888</td>
<td>110</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1A7A4</td>
<td>CHAIN POINTER 1A8D4</td>
<td>110</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1A7F0</td>
<td>CHAIN POINTER 1A83C</td>
<td>110</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1A83C</td>
<td>CHAIN POINTER 1A920</td>
<td>110</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1A888</td>
<td>CHAIN POINTER 1A7A4</td>
<td>110</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1A8D4</td>
<td>CHAIN POINTER 1A7FO</td>
<td>110</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Address</td>
<td>Contents</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>--------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>1AC18</td>
<td>68191236 80806BC4 01490010 08001E00 03030200 01013AC0 039100F5 C3114040</td>
<td></td>
</tr>
<tr>
<td></td>
<td>370S DUMP</td>
<td></td>
</tr>
<tr>
<td>1AC64</td>
<td>6820048 6871114C 613C4D4D 5C1D4011 40P101D40 40114F40 1D40D7C6 4040F140</td>
<td></td>
</tr>
<tr>
<td>1AC80</td>
<td>68C40048 601D40D7 C64040F2 4078404D 5D46F1F5 5D46F1F5 5D46F1F5 5D46F1F5</td>
<td></td>
</tr>
<tr>
<td>1ACFC</td>
<td>6860048 67C93E3C 6340C9D5 C6D69D4 401D4040 1D4040C4 401D4040 1D4040C4</td>
<td></td>
</tr>
<tr>
<td>1AD48</td>
<td>68060AOE 00000000 00000000 00000000 00000000 00000000 00000000 00000000</td>
<td></td>
</tr>
<tr>
<td>1AD94</td>
<td>68D70033 D7F01D40 401D940 1D40D7C6 401D4040 1D4040C4 401D4040 1D4040C4</td>
<td></td>
</tr>
<tr>
<td>1ADB0</td>
<td>6C23010 00000000 00100000 6B781200 03C40200 07078000 00001000 00000000</td>
<td></td>
</tr>
<tr>
<td>1AE2C</td>
<td>68B20A10 00000000 00100000 6B9B1200 00000000 00000000 00000000 00000000</td>
<td></td>
</tr>
<tr>
<td>1AE78</td>
<td>6C360212 00000000 00160000 64001F00 00000000 00000000 00000000 00000000</td>
<td></td>
</tr>
<tr>
<td>1AE98</td>
<td>6C100A0D 00000000 00000000 6B11C000 00000000 00000000 00000000 00000000</td>
<td></td>
</tr>
<tr>
<td>1AF10</td>
<td>683F1202 80010000 00002200 00000000 00000000 00000000 00000000 00000000</td>
<td></td>
</tr>
<tr>
<td>Address</td>
<td>Pointer</td>
<td>Pointer Value</td>
</tr>
<tr>
<td>---------</td>
<td>---------</td>
<td>---------------</td>
</tr>
<tr>
<td>00000000 00000000 00000000 00000000</td>
<td>1AFF4</td>
<td>1AFF4</td>
</tr>
<tr>
<td>00000000 00000000 00000000 00000000</td>
<td>1ADEO</td>
<td>01010000 00000000 00000000 00000000</td>
</tr>
<tr>
<td>00000000 00000000 00000000 00000000</td>
<td>1AE8</td>
<td>01010000 00000000 00000000 00000000</td>
</tr>
<tr>
<td>00000000 00000000 00000000 00000000</td>
<td>01010000 00000000 00000000 00000000</td>
<td>00000000 00000000 00000000 00000000</td>
</tr>
<tr>
<td>00000000 00000000 00000000 00000000</td>
<td>01010000 00000000 00000000 00000000</td>
<td>00000000 00000000 00000000 00000000</td>
</tr>
<tr>
<td>00000000 00000000 00000000 00000000</td>
<td>01010000 00000000 00000000 00000000</td>
<td>00000000 00000000 00000000 00000000</td>
</tr>
<tr>
<td>00000000 00000000 00000000 00000000</td>
<td>01010000 00000000 00000000 00000000</td>
<td>00000000 00000000 00000000 00000000</td>
</tr>
<tr>
<td>00000000 00000000 00000000 00000000</td>
<td>01010000 00000000 00000000 00000000</td>
<td>00000000 00000000 00000000 00000000</td>
</tr>
<tr>
<td>00000000 00000000 00000000 00000000</td>
<td>01010000 00000000 00000000 00000000</td>
<td>00000000 00000000 00000000 00000000</td>
</tr>
<tr>
<td>00000000 00000000 00000000 00000000</td>
<td>01010000 00000000 00000000 00000000</td>
<td>00000000 00000000 00000000 00000000</td>
</tr>
<tr>
<td>00000000 00000000 00000000 00000000</td>
<td>01010000 00000000 00000000 00000000</td>
<td>00000000 00000000 00000000 00000000</td>
</tr>
<tr>
<td>00000000 00000000 00000000 00000000</td>
<td>01010000 00000000 00000000 00000000</td>
<td>00000000 00000000 00000000 00000000</td>
</tr>
<tr>
<td>00000000 00000000 00000000 00000000</td>
<td>01010000 00000000 00000000 00000000</td>
<td>00000000 00000000 00000000 00000000</td>
</tr>
<tr>
<td>Address</td>
<td>Chain Pointer</td>
<td>Contents</td>
</tr>
<tr>
<td>---------</td>
<td>---------------</td>
<td>----------</td>
</tr>
<tr>
<td>1B254</td>
<td>1B124</td>
<td>6C90A17</td>
</tr>
<tr>
<td></td>
<td></td>
<td>00000000</td>
</tr>
<tr>
<td>1B2A0</td>
<td>1B468</td>
<td>6D10A0F</td>
</tr>
<tr>
<td></td>
<td></td>
<td>00000000</td>
</tr>
<tr>
<td>1B2EC</td>
<td>1B338</td>
<td>6CCD0A14</td>
</tr>
<tr>
<td></td>
<td></td>
<td>00000000</td>
</tr>
<tr>
<td>1B338</td>
<td>1B384</td>
<td>6C50A14</td>
</tr>
<tr>
<td></td>
<td></td>
<td>00000000</td>
</tr>
<tr>
<td>1B384</td>
<td>1B208</td>
<td>6D810A2B</td>
</tr>
<tr>
<td></td>
<td></td>
<td>00000000</td>
</tr>
<tr>
<td>1B3D0</td>
<td>1B2E4</td>
<td>68B90000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>00000000</td>
</tr>
<tr>
<td>1B41C</td>
<td>1B2E4</td>
<td>68070000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>00000000</td>
</tr>
<tr>
<td>1B468</td>
<td>1B41C</td>
<td>6C80A2B</td>
</tr>
<tr>
<td></td>
<td></td>
<td>00000000</td>
</tr>
<tr>
<td>1B4B4</td>
<td>1B54C</td>
<td>6D530000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>00000000</td>
</tr>
<tr>
<td>1B500</td>
<td>1B54C</td>
<td>6D200000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>00000000</td>
</tr>
<tr>
<td>Address</td>
<td>Chain Pointer</td>
<td>Address</td>
</tr>
<tr>
<td>---------</td>
<td>---------------</td>
<td>---------</td>
</tr>
<tr>
<td>1B890</td>
<td>1B8DC</td>
<td>1B928</td>
</tr>
<tr>
<td>6E370000</td>
<td>00000000</td>
<td>6E540000</td>
</tr>
<tr>
<td>00000000</td>
<td>00000000</td>
<td>00000000</td>
</tr>
<tr>
<td>00000000</td>
<td>00000000</td>
<td>00000000</td>
</tr>
<tr>
<td>1B8DC</td>
<td>1B928</td>
<td>1B974</td>
</tr>
<tr>
<td>6E340000</td>
<td>00000000</td>
<td>6E5D0000</td>
</tr>
<tr>
<td>00000000</td>
<td>00000000</td>
<td>00000000</td>
</tr>
<tr>
<td>00000000</td>
<td>00000000</td>
<td>00000000</td>
</tr>
<tr>
<td>1B928</td>
<td>1B974</td>
<td>1B9C0</td>
</tr>
<tr>
<td>6E5D0000</td>
<td>00000000</td>
<td>6E530000</td>
</tr>
<tr>
<td>00000000</td>
<td>00000000</td>
<td>00000000</td>
</tr>
<tr>
<td>Offset</td>
<td>CHAIN POINTER</td>
<td>CHAIN POINTER</td>
</tr>
<tr>
<td>--------</td>
<td>----------------</td>
<td>----------------</td>
</tr>
<tr>
<td>3705</td>
<td>1BF18</td>
<td>1BF64</td>
</tr>
<tr>
<td>1BEC</td>
<td>6FC6F110 9BC9CC2C DDA512C 11889834 8028371O 47309887 C188A16C 51A85081 73942088 80015134 A858D520</td>
<td>DDB8A850 250E2488 220ABB20 00EC0398</td>
</tr>
<tr>
<td>1BF64</td>
<td>6FECB920 006C018A 81D2188 51048020 81005154 BF200400 57465188 D1365174</td>
<td>81005154 BF200400 57465188 D1365174</td>
</tr>
<tr>
<td>1BF60</td>
<td>674B90 81005154 BF200400 57465188 D1365174</td>
<td>81005154 BF200400 57465188 D1365174</td>
</tr>
<tr>
<td>1C8A</td>
<td>672A88FB 610CF033 9815B920 00096164</td>
<td>B920040 A8176788 8F048886 A861802</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>ACB</td>
<td>Adapter control block</td>
<td></td>
</tr>
<tr>
<td>APPL</td>
<td>Application</td>
<td></td>
</tr>
<tr>
<td>ATB</td>
<td>Address trace block</td>
<td></td>
</tr>
<tr>
<td>BCB</td>
<td>Bit control block</td>
<td></td>
</tr>
<tr>
<td>BCU</td>
<td>Block control unit</td>
<td></td>
</tr>
<tr>
<td>BSC</td>
<td>Binary synchronous</td>
<td></td>
</tr>
<tr>
<td>BNN</td>
<td>Boundary network node</td>
<td></td>
</tr>
<tr>
<td>CCB</td>
<td>Character control block</td>
<td></td>
</tr>
<tr>
<td>CCP</td>
<td>Communications control program</td>
<td></td>
</tr>
<tr>
<td>CSP</td>
<td>Character service program</td>
<td></td>
</tr>
<tr>
<td>CCW</td>
<td>Channel control word</td>
<td></td>
</tr>
<tr>
<td>CDS</td>
<td>Configuration data set</td>
<td></td>
</tr>
<tr>
<td>CE</td>
<td>Channel end</td>
<td></td>
</tr>
<tr>
<td>CHB</td>
<td>Channel control block</td>
<td></td>
</tr>
<tr>
<td>CIC</td>
<td>Channel input chain</td>
<td></td>
</tr>
<tr>
<td>COB</td>
<td>Channel operation block</td>
<td></td>
</tr>
<tr>
<td>COC</td>
<td>Channel output chain</td>
<td></td>
</tr>
<tr>
<td>CPM</td>
<td>Connection point manager</td>
<td></td>
</tr>
<tr>
<td>CRP</td>
<td>Check record pool</td>
<td></td>
</tr>
<tr>
<td>CSP</td>
<td>Character service program</td>
<td></td>
</tr>
<tr>
<td>CUB</td>
<td>Common unit physical block</td>
<td></td>
</tr>
<tr>
<td>CW</td>
<td>Channel word</td>
<td></td>
</tr>
<tr>
<td>DAF</td>
<td>Destination address field</td>
<td></td>
</tr>
<tr>
<td>DLC</td>
<td>Data link control</td>
<td></td>
</tr>
<tr>
<td>DVB</td>
<td>Device base control block</td>
<td></td>
</tr>
<tr>
<td>ECB</td>
<td>Event control block</td>
<td></td>
</tr>
<tr>
<td>EP</td>
<td>Emulation program</td>
<td></td>
</tr>
<tr>
<td>FID</td>
<td>Format identification</td>
<td></td>
</tr>
<tr>
<td>FIFO</td>
<td>First in/first out</td>
<td></td>
</tr>
<tr>
<td>FM</td>
<td>Function management</td>
<td></td>
</tr>
<tr>
<td>FME</td>
<td>Function management end</td>
<td></td>
</tr>
<tr>
<td>HWE</td>
<td>Extended halfword direct addressables</td>
<td></td>
</tr>
<tr>
<td>IAR</td>
<td>Instruction address register</td>
<td></td>
</tr>
<tr>
<td>ICW</td>
<td>Interface control word</td>
<td></td>
</tr>
<tr>
<td>IPL</td>
<td>Initial program load</td>
<td></td>
</tr>
<tr>
<td>LCB</td>
<td>Line control block</td>
<td></td>
</tr>
<tr>
<td>LGT</td>
<td>Line group table</td>
<td></td>
</tr>
<tr>
<td>LKB</td>
<td>Link control block</td>
<td></td>
</tr>
<tr>
<td>LNV T</td>
<td>Line vector table</td>
<td></td>
</tr>
<tr>
<td>LOBQ</td>
<td>Link outbound queue</td>
<td></td>
</tr>
<tr>
<td>LOSQ</td>
<td>Link outstanding queue</td>
<td></td>
</tr>
<tr>
<td>LTCB</td>
<td>Line trace control block</td>
<td></td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>-------------</td>
<td>---------------------------------------</td>
<td></td>
</tr>
<tr>
<td>LTCT</td>
<td>Line type command table</td>
<td></td>
</tr>
<tr>
<td>LU</td>
<td>Logical unit</td>
<td></td>
</tr>
<tr>
<td>LUB</td>
<td>Logical unit control block</td>
<td></td>
</tr>
<tr>
<td>LUV</td>
<td>Logical unit vector table</td>
<td></td>
</tr>
<tr>
<td>LXO</td>
<td>Link XIO block</td>
<td></td>
</tr>
<tr>
<td>NCP</td>
<td>Network control program</td>
<td></td>
</tr>
<tr>
<td>NSA</td>
<td>Nonsequenced acknowledgement</td>
<td></td>
</tr>
<tr>
<td>OAF</td>
<td>Origin address field</td>
<td></td>
</tr>
<tr>
<td>OLLT</td>
<td>Online line test</td>
<td></td>
</tr>
<tr>
<td>OLT</td>
<td>Online terminal test</td>
<td></td>
</tr>
<tr>
<td>PC</td>
<td>Path control</td>
<td></td>
</tr>
<tr>
<td>PCB</td>
<td>Panel control block</td>
<td></td>
</tr>
<tr>
<td>PCI</td>
<td>Program-controlled interrupt</td>
<td></td>
</tr>
<tr>
<td>PEP</td>
<td>Partitioned emulation program</td>
<td></td>
</tr>
<tr>
<td>PIU</td>
<td>Path information unit</td>
<td></td>
</tr>
<tr>
<td>PSB</td>
<td>Physical services block</td>
<td></td>
</tr>
<tr>
<td>QCB</td>
<td>Queue control block</td>
<td></td>
</tr>
<tr>
<td>RH</td>
<td>Request/response header</td>
<td></td>
</tr>
<tr>
<td>RQI</td>
<td>Request initialization</td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>Receive ready</td>
<td></td>
</tr>
<tr>
<td>RS</td>
<td>Read start (RS0, RS1)</td>
<td></td>
</tr>
<tr>
<td>RU</td>
<td>Request/response unit</td>
<td></td>
</tr>
<tr>
<td>RVT</td>
<td>Resource vector table</td>
<td></td>
</tr>
<tr>
<td>SC</td>
<td>System control</td>
<td></td>
</tr>
<tr>
<td>SCB</td>
<td>Station control block</td>
<td></td>
</tr>
<tr>
<td>SCRR</td>
<td>System control router table</td>
<td></td>
</tr>
<tr>
<td>SDLR</td>
<td>Synchronous data link control</td>
<td></td>
</tr>
<tr>
<td>SID</td>
<td>Send identification</td>
<td></td>
</tr>
<tr>
<td>SIT</td>
<td>Subarea index table</td>
<td></td>
</tr>
<tr>
<td>SM</td>
<td>Status modifier</td>
<td></td>
</tr>
<tr>
<td>SNRM</td>
<td>Set normal response mode</td>
<td></td>
</tr>
<tr>
<td>SOT</td>
<td>Service order table</td>
<td></td>
</tr>
<tr>
<td>SS</td>
<td>Start-stop</td>
<td></td>
</tr>
<tr>
<td>SSCP</td>
<td>System services control point</td>
<td></td>
</tr>
<tr>
<td>SVC</td>
<td>Supervisor call</td>
<td></td>
</tr>
<tr>
<td>SVT</td>
<td>Subarea vector table</td>
<td></td>
</tr>
<tr>
<td>TH</td>
<td>Transmission header</td>
<td></td>
</tr>
<tr>
<td>UC</td>
<td>Unit check</td>
<td></td>
</tr>
<tr>
<td>UE</td>
<td>Unit exception</td>
<td></td>
</tr>
<tr>
<td>WS</td>
<td>Write start (WS0, WS1)</td>
<td></td>
</tr>
<tr>
<td>XDA</td>
<td>Word direct addressables</td>
<td></td>
</tr>
<tr>
<td>XDB</td>
<td>Byte direct addressables</td>
<td></td>
</tr>
<tr>
<td>XDH</td>
<td>Halfword direct addressables</td>
<td></td>
</tr>
</tbody>
</table>
Index

Abend 11-3
Adapter control block (ACB) 9-3, 10-3
Address trace 11-2
Bar vector table (see line vector table) BHR extension 10-22
Bit control block (BCB) 9-4, 10-5
Bit service 10-8
Block control unit (BCU) 10-1
Block-handler control blocks 10-21
Block-handler driver table (BHD) 10-23
Block-handler routines 10-20
Block-handler set (BHS) 10-23
Block-handler set table (BST) 10-23
Boundary network node 2-3, 7-1
Type 1 or 4 adapter 2-2
Type 2 or 3 adapter 2-2
BSC/SS flow 10-14
BSC/SS processor 2-4, 10-1
BSC/SS sessions 10-12
BSC/SS XIO processing 10-16
BTU commands for BSC/SS resources 10-8
Channel adapter IOS 2-2, 4-1
Channel adapter performance 4-9
Channel adapter trace 11-2
Channel control block (CHB) 4-6
Channel operation block (COB) 4-4
Channel word (CW) 4-7
Character control block (CCB) 9-4, 10-4
Character service program (CSP) 10-8, 10-18
Commands Appendix 1
Communications control program (CCP) 10-7
Common physical unit block (CUB) 7-1
Configuration data set 8-5
Connection point manager
Physical services 6-5
Boundary network node 7-3, 7-15
Contact command 10-9
Control command 10-11
CPM (see connection point manager)
CUB (see common physical unit block)
Data link control block (LKB) 9-1
Device base control block (DVB) 10-2
Device command processor 10-7
Diagnostics 11-1
Direct addressables 3-2
Disconnect command 10-11
Dynamic panel display 11-1
Error and statistic recording 11-2
Function management 6-6
Input/output block 10-3
Intermediate node path control (see path control)
Invoke command 10-9
Line control block (LCB) 10-3
Line control selection table (LCST) 10-26
Line group table (LGT) 9-2, 10-24, 10-26
Line I/O task 10-7
Line test 11-1
Line trace 11-2
Line type command table (LTCT) 10-3
Line vector table (LNVT) 9-4, 10-4
Line work scheduler 10-7
Link control block (LKB) 9-2
Link scheduler 2-3, 9-5
LINK XIO block (LXB) 9-3
Local/remote link 8-1
Logical connections 10-12
Logical unit block (LUB) 7-2
Logical unit vector table (LUV) 7-17
LUB (see logical unit block)
LUV (see logical unit vector table)
MTA group, line and terminal 10-24
MTA line group table (LGT) 10-26
MTA list 10-24
Multiple terminal access (MTA) 10-24
Network control program supervisor 2-1, 3-1
Nondevice command processor 10-7
Online tests 11-3
Path control 2-2, 5-1
Path control in delayed 7-14
Path control in immediate 7-14
Path control out 2-2, 5-4
Path control out delayed 7-8
Path information unit (PIU) 3-5, Appendix 1
Physical services 2-3, 6-1
Physical services control block (PSB) 6-9
Polling list, IBM 1050 10-27
PSB (see physical services control block)
PUB (see physical unit block)
Read command 10-10
Read start (RSO, RSI) 3-1
Remote link (see local/remote link)
Request/response header (RH) 3-5, 6-6
Request/response unit (RU) 3-5, 6-5
Resource vector table 5-3, 10-2
SDLC routines 2-4
SDLC switched support 7-17
Service aids 11-1
Service order table (SOT) 9-5, 10-5
Service-seeking 10-13
Sessions (BSC/SS) see BSC/SS sessions
Session hierarchy 6-1
SSCP and NCP physical services 6-1
SSCP and PU physical services 6-3
SSCP and LU logical services 6-3
Host application and LU logical services 6-3
Session-servicing 10-13
Station control block (SCB) 8-1
Status modifier 4-10
Subarea index table (SIT) 5-1
Subarea vector table (SVT) 5-2
SVC services 3-16
Switched SDLC link 7-17
System control 6-5
System router 10-6
Task dispatching priorities 3-11
Task management 3-2
Task states 3-10
Test command 10-11
Transmission header (TH) 3-5
User block-handler routines 10-23
Write command 10-10
Write start (WSO, WSI) 3-1
XDA (see direct addressables)
XDB (see direct addressables)
XDH (see direct addressables)