DIGITAL EQUIPMENT CORPORATION

OPTION BULLETIN

pdp11

DX11B–PDP-11 TO IBM 360/370 CHANNEL INTERFACE

digital

FEATURES

- Interfaces to most models of the IBM 360 or 370 on the selector, multiplexer or block multiplexer channels
- Recognizes up to 128 IBM device addresses over the full range of 256 addresses.
- Operates in the byte multiplexed or burst mode
- NPR (DMA) operations to present status to 360, store 360 commands in the PDP-11, and transfer data
- Hardware recognition and presentation of the 360 device address plus hardware presentation of initial status
- Software interpretation and response to 360 commands
- Can be programmed to emulate a 2848, 2703 or 3705 control unit
- In off-line or powered-down mode, the DX11B is transparent to the S/360 and presents no load to the channel data and tag lines
- Built-in maintenance and protection features
- 250,000 byte/second data transfer rate (depending upon IBM model)

DESCRIPTION

The DX11B is a programmable interface between a PDP-11 UNIBUS™ and a S/360 or S/370 multiplexer or selector channel. The DX11B hardware handles the detection and response to all channel generated control signals. The DX11B hardware handles the Initial Selection Sequence operation without program intervention. It recognizes a wired (strapped) set of addresses, presents address, fetches a unique status (determined by the 360/370 command and device address) from a table in memory and stores an entry in a 128 entry tumble table. The status and tumble operations are by NPR (DMA). The tumble table entry contains status, IBM command and IBM address. Software interprets the command and responds to it. The commands recognized and the manner of response will depend upon the 360/370 control unit being emulated.

As soon as the hardware has stored the tumble table entry, it is ready to service another request from the 360/370. If both the PDP-11 and the 360/370 channel contend for the DX11B, the 360/370 channel wins and the PDP-11 is locked out. This protection feature makes sure the 360/370 channel is always master. It can cancel a previous request at any time.

The PDP-11 program loads DX11B registers to cause data transfers. Data transfer is by NPR. The length in bytes can be short (multiplex mode) or long (burst mode). Software determines which mode will be used. Burst mode is on selector channel or selector subchannel only.

The DX11B can be taken off-line or powered down. In either case, a relay closes to by-pass the SELECT-OUT line. The drivers and receivers on the 360/370 control and data lines present no bus loads when the DX11B is powered down. Thus, the DX11B is logically disengaged from the channel in the power-down or off-line mode. The DX11B contains power failure and timeout features. The power failure protection hardware interrupts the DX11 when an AC-low is detected. When in burst mode, timeout hardware interrupts the DX11 if the PDP-11 fails to respond in 5 seconds. In either case, the DX11B hardware stops all data transfers, presents UNIT CHECK status to the 360/370 and goes off-line. The programmer can disable the timeout feature during program debugging. The timeout will occur only while one of the DX11B-recognized devices is active (OPERATIONAL-IN is high).

The DX11B has a built-in channel simulator which is used by off-line diagnostics to verify the PDP-11 configuration up to and including the IBM cables. The latter can be checked by plugging one end into the DX11B and the other into the simulator.

Because NPR is used for data transfers, the DX11B is capable of data transfer rates in excess of 250,000 bytes/second. It is limited by the PDP-11 configuration and the rated capacity of the 360/370 channel to which it is attached.

PROGRAMMING

Programming Interfaces

Registers

The DX11B has the following set of programmable registers:

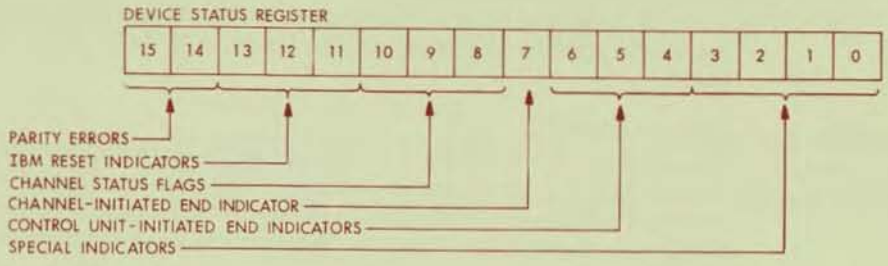
Address	Description
176200	Device Status (DXDS)
176202	Command and Address (DXCA)
176204	Control Unit Status (DXCS)
176206	Offset and Status (DXOS)
176210	Bus Address for NPR (DXBA)
176212	Byte Count for NPR (DXBC)
176214	Maintenance-Out (DXMO)
176216	Maintenance-In (DXMI)
176220	Control Bits (DXCB)
176222	NPR Data (DXND)
176224	Extra Signals (DXES1)
176226	Maintenance-Out Buffered (DXMOB)
176230	Extra Signals (DXES2)

REGISTERS

DXDS

DEVICE STATUS

This register contains all of the interrupt producing conditions along with various non-interrupt producing device status flags. This register is read only and is stored as the first tumble table entry before being reset.



BITS

Error Indicators

15 PARER — Parity Error

This bit is set if the DX11B detects an even-parity condition on the BUSOUT when either command or data information is sent by the channel (CH) to the DX11B. PARER will be set if bad Command-Out parity is detected during an ISS or if bad (BUSO) Data-Out parity is detected. PARER is not set on bad Address-Out parity during an ISS. In this case the DX11B will not recognize the Address from the CH.

14 NXM — PDP-11 Bus Timeout

This bit will set should the PDP-11 take longer than 20μs to complete any one Non-Processor Request Transaction (NPR). Such an occurrence might be the result of addressing a non-existent (memory) location. If set during a data transfer, this will terminate the sequence by setting CUDEND. Bus timeout is taken as equivalent to bus completion elsewhere so that a sequence may proceed to its normal ending point.

IBM Reset Indicators

13 SELRST — Selective Reset

This bit will be set by the channel execution of a Selective Reset Sequence as described in the *Channel (CH) to Control Unit (CU) OEM interface manual* published by IBM. This sequence is usually a response to a malfunction CU/device. When set, this bit causes a Program Interrupt (PI).

12 SYSRST — System Reset

This bit will be set by the channel execution of a System Reset sequence as described in the IBM document *Channel to Control Unit OEM Interface Manual.* When set this bit will cause a Program Interrupt.

11 INFDSC — Interface Disconnect

This bit is set when the channel performs a disconnect operation with the Control Unit.

Channel Status Flags

10 UCHKS — Unit Check Sent

Unit Check was included in status sent to the channel.

09 CHENDS — Channel End Sent

This bit is used to notify the emulator that CHEND status was sent in a status response.

08 BSYS — BSY Sent

BSY status bit was sent to the channel.

Channel Initiated End Indicator

07 CHIS — Channel Initiated (CHI) Selection Sequence End

This bit is set when a channel initiated sequence has been completed with the control unit. This bit becomes a zero when the DXDS is reset after the DXDS is entered in the tumble table.

Control Unit Initiated (CUI) End Indicators (PDP-11/DX11B)

06 ESEND — Ending Sequence End

This bit will set when a status byte is presented to the channel by a DX11 initialized sequence and/or when a stacked status is finally accepted.

This bit is most commonly associated with the Ending Status presentation type of sequence which normally follows a Data Transfer Sequence.

It may under some circumstances occur that the CHIS bit is set as a result of a CUI-ISS contention situation where the device address requested matched with the device address selected from the channel and the CU was requesting present status.

05 CHDEND — Channel Data End

This bit is set during a Data Transfer Sequence when the channel byte count overflows (Command-Out is sent in response to Service-In). This bit in its true state causes the CU to terminate the Data Transfer Sequence. This bit is also set when the CH terminates a Data Transfer Sequence by interface disconnect (INFDSC=1).

04 CUDEND — Control Unit Data End

This bit is set during a Data Transfer Sequence when the DXBC (byte count) register goes to its all zero state. When set, this bit causes a PI and also causes the DONE bit to assert. When this bit asserts, it causes the CU to terminate the Data Transfer Sequence.

Special Indicators

03 ISSREJ — Initial Selection Sequence Rejected

This bit is set when a channel-initiated selection sequence addressed to the CU was answered by the CU with a Control Unit Busy status indication and a short Control Unit Busy sequence. This can only occur if the CUBSY bit in the DXCS was set when the CH tried to initiate an ISS and the CU was in its idle phase.

02 CMDCHN — Command Chaining

This bit sets if the channel has indicated that another operation will probably follow for the CU/Device currently connected when the DX11B presents Device End status to the channel. Command chaining occurs when the current 360 Channel Command Word (CCW) has its command chaining bit set. The channel informs the DX11 of this by raising Suppress-Out at the same time as Service-Out.

01 STKSTB — Stack Status Copy

This bit is set when the Channel (CH) informs the DX11 that the status byte being presented on the BUS-IN cannot currently be accepted by the CH. This occurs when the CH responds to Status-In with Command-Out.
STKSTA may be set by the program if the DX11 is not active with the CH (LOCK=0). This is useful when initiating a DX11 request for status presentation. If Suppress-Out and STKSTA are both true, the DX11 drops its Request-In since the status contained is suppressible (once status has been stacked the CH also defines that status as suppressible). Since this bit is a copy of STKSTA, it is read only and is not reset with the rest of DXDS.

00 CMDREJ — Command Rejected

A Channel Initiated Selection Sequence command was ignored due to bad parity, a busy device, a pending status, or an illegal command for the device.

DXCA — COMMAND AND ADDRESS

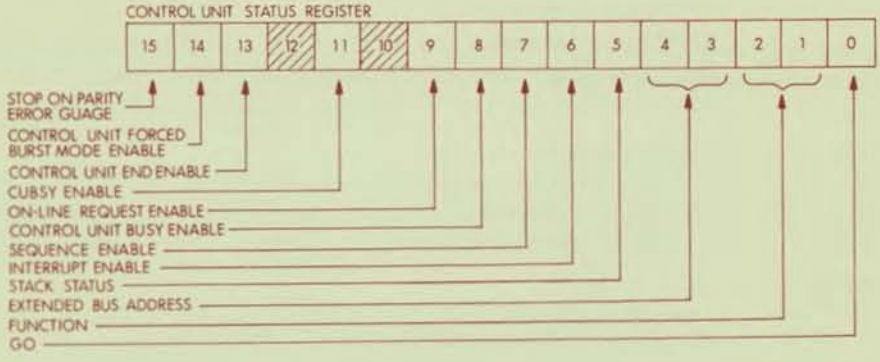
DXCA contains the control unit command register, CUCR, and the control unit address register, CUAR. These two bytes are the address and command as transmitted from the channel during an initial selection sequence. This register is stored as the second tumble table entry upon completion of a CH-CU interaction.

BITS		
15-08	CUCR	The odd byte of the DXCA register contains the Control Unit Command Register (CUCR). The CUCR cannot be loaded by the program. This byte contains the last command sent by the channel (even if it was rejected by the DX11).
07-00	CUAR	The even byte contains the Control Unit Address Register (CUAR). The CUAR is loaded with the device address from Bus-Out during an initial selection sequence. The CUAR is also loaded and cleared via PDP program control if LOCKO is a (0). The CUAR need not be loaded with some device address that is to be supported by the DX11 prior to going ON-LINE.

DXCS

CONTROL UNIT STATUS

This register contains the primary control command information bits and primary status indications for the DX11. The DXCS may not be modified by the PDP program when LOCKO equals one (except for DONE and INTEN).



BITS		
15	PARSTP	Stop on Parity Error Enable If this bit is set to a one and a parity error occurs on Bus-Out during a Data Transfer Sequence, then the sequence is terminated and CUDEND will be set. If this bit is not set, PARER may still become set but the sequence will end normally.
14	CUFBM	Control Unit Forced Burst Mode Enable This bit can be set by the program when CU Forced Burst mode is desired. This condition causes the CU to hold OPL-IN up from initial selection through the presentation of Channel End (except for TIO and IIIO).
13	ENDEN	Control Unit End Enable This bit is set and cleared only by the PDP-11 program. The purpose of setting this bit is to assert the control unit end bit (CUEND) in the device status presented to the channel during a control unit busy sequence; i.e., where the CUBSY bit is already set.
12		Reserved
11	BSYEN	CUBSY Enable This bit enables the setting of CUBSY immediately upon responding to an ISS (either CUI or CHI). This bit is set to a one by the program for emulating any single thread (shared) control unit such as 2848, 2803 and 2821. It is set to zero for multiple-thread control units such as 2703.
10		Reserved This bit is reserved for future use.

09	ONLINA	On-Line Request Enable This bit is writable (except when LOCKO is set) by the PDP program to either the one or the zero state. ONLINA indicates that the control unit has made or is making a request to go on-line to the 360 channel. It is a two-stage operation. This is the lower stage of going on- or off-line operation and is the stage loaded or cleared by program command (ONLINA: DXCS(09)). The upper stage is the operating on-line bit ONLINB (DXCB(02)). ONLINB follows the changes of ONLINA at a time when, as specified in the OEM channel manual (IBM), it is proper to make changes from on-line to off-line or from off-line to on-line. (The CH is considered on-line itself whenever Operational-Out is set.)
08	CUBSY	Control Unit Busy Enable Setting this bit will cause a channel initiated sequence to be answered by the DX11 with a Control Unit Busy Sequence. This bit causes the BSY bit to assert to the BUS-IN during the subsequent status presentation from the DX11. This bit is set and cleared by program and by the DX11 hardware if so enabled by BSYEN.
07	DONE	Sequence Done The DONE bit is the normal interrupt producing condition which the DX11 uses for its primary interrupt control (c.f., INTEN). If both DONE and INTEN are set an interrupt will be requested. Clearing DONE is required (of the program) only before making an attempt to change registers. With DONE reset, the DX11 will reset LOCKO if the DX11 is in either phase 0 or phase 7. Loading DONE is allowed only in phases 0 or 7.
06	INTEN	Interrupt Enable This bit is always writable. It is recommended that this bit always be set before setting ONLINA and that ONLINB (via ONLINA) be cleared prior to clearing this bit. This bit may be cleared or set by program control only.
05	STKSTA	Stack Status If set, STKSTA indicates that status was stacked. When cleared, it indicates that status was accepted. It may also be set voluntarily by a program that is presenting a suppressible (or low priority) status. It is also set automatically by the DX11 when the CH requires a status to be stacked and the DX11 will attempt to present it again.
04, 03	XBA	Extended Bus Address Bits These bits are the two extended most significant bits of the memory address register during data input/output. They are loaded and cleared under program control and may be caused to complement should the DXBA overflow from a DXBA increment of +2 during a data transfer. They are used only during a data sequence.
02, 01	FCTN	Function These two bits make up the DX11 function register. It is used by the program to select the CU operations desired: FCTN=0—reset the DX FCTN=1—input data transfer (from 360/370) FCTN=2—output data transfer (to 360/370) FCTN=3—present asynchronous status (to 360/370)
00	GO	When the GO bit is set, the function requested is performed. If FCTN=0 the reset operation is done on the DX11 and the DONE bit is left cleared. If FCTN is not zero, then Request-In (REQI) will be raised as the start of a Control Unit Initiated (CUI) sequence.
DXOS		OFFSET AND STATUS This is a two-byte register. The odd byte (CUOR) contains the offset address of the Status Pointer Word (SPW) table located in PDP-11 memory. The even byte (CUSR) contains the status byte that will be presented to the 360/370.

BITS		
15-08	CUOR	Control Unit Offset Register The CUOR contains the high order six bits of the SPW table and of the tumble table. It is program-loaded while in the off-line mode.
07-00	CUSR	Control Unit Status Register The CUSR contains the standard IBM status information bits which are transmitted to the channel.
07	ATTEN	Attention
06	STAMOD	Status Modifier
05	CUEND	Control Unit End
04	BSY	Busy The program should not directly set this bit. This bit is set only by a CU Busy Sequence or by being loaded as the status portion of the Status Pointer Word.
03	CHEND	Channel End
02	DEVEND	Device End
01	UCHECK	Unit Check
00	UEXCEP	Unit Exception

DXBA

BUS ADDRESS REGISTER
The bus address register is a 16-bit register which can be cleared and loaded under program control if LOCKO is not set (0). It is used during data transfers to point to the PDP-11 core location to/from which data will be transferred 16-bit words at a time. The DXBA register is also used during channel initiated sequences to fetch both the Status Pointer Word and the device status byte from PDP-11 core. During a data transfer the DXBA is preset by program to point to the first byte location where data is sent or stored. The DXBA register is incremented by two each time a PDP data word is fetched or stored in core during the data transfer process. Should the DXBA register overflow, the extended memory address bits (XBA) in the DXCS register will be caused to complement their states appropriately. The DXBA is also used to address the tumble table when information is to be stored there.

DXBA(00)

The low order bit of the DXBA is normally set to zero by program load. When this bit is placed on the UNIBUS address lines (ABUS), it is always represented as zero. This bit is also used to initiate the BALF flop when control is transferred to phases 5 or 6 to select the odd or even first byte of the first data word.

DXBC

BYTE COUNT
This register is used only during data transfers. It is loaded and cleared under program control and is set up prior to the data transfer involved. The DXBC is set to the negative of the number of bytes to be transferred. As each byte is actually transferred to/from the DX11, the DXBC register is incremented by one until all bytes are transferred, whereupon the DXBC equals zero. When the DXBC contents go to zero during a Data Transfer Sequence, the CUDEND bit of the DXDS will set, thereby terminating the data transfer sequence with the channel.

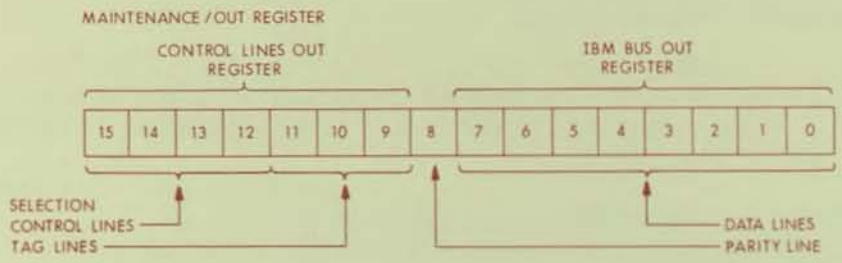
DXMO

MAINTENANCE-OUT

This register is used for holding the 360 Channel Bus-Out Data and Tags. This register is always directly readable by a PDP-11 program.

When the DX11 is On-line, the bits in this register are usually the same as what appears on the Bus-Out lines (hardwired cables to the Bus-Out Plug).

When the DX11 is Off-line, these bits may be written directly by a PDP programmed request. The programmed bits are held buffered in the DXMOB. When the DX11 is On-line but cabled to the Bus-Out Test Plug, these bits are also writable by PDP-11 programs. The on-line cabled mode is used to isolate the cables and Bus-Out receivers as an error source.



BITS

15-08 CONO — Control Out-Lines Register

This byte contains the following signals as strobed from either the Bus-Out lines or from CONOB:

Selection Control Lines

15 OPLO — Operational-Out

This line indicates that the channel is in operation.

Note: refer to IBM manual A22-6843 for detailed description of each line of the 360 bus.

14 HLDO — Hold-Out

13 SELO — Select-Out

This bit is set only if both hold-out and select-out are set. When set or cleared by a PDP-11 program, only the simulated select-out signal is affected (see DXMOB).

12 SUPO — Suppress-Out

Tag Lines

11 ADRO — Address-Out

10 CMDO — Command-Out

09 SRVO — Service-Out

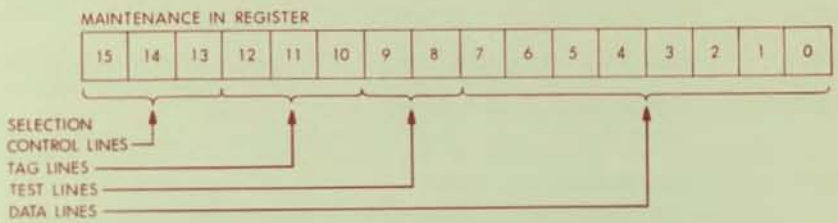
Parity Line

08 PARO — Parity-Out

This bit does double duty when written by a PDP program while the DX11 is in on-line cabled mode. At such times the state of the bit will be translated directly into the state of the Simulated Clock-Out line (of the Bus-Out Test Plug) and into the Parity-Out line. Clock-Out's primary purpose is to synchronize the control units to make changes in their ON/OFF-LINE state.

Another function of the bit is to allow the program generation of either normal (odd) Parity-Out or "bad" (even) Parity-Out. This feature is necessary in order to permit checking the parity generator within the main DX11 logic.

Data Lines 07-00	BUSO	IBM Bus-Out Register Bus-Out data bits 0 to 7, as seen either directly from the Bus-Out Cables or from BUSOB, if off-line. When written by the PDP-11 program, this byte is buffered in BUSOB.
DXMI		MAINTENANCE-IN This register is used for reading the Bus-In Tags and data originating mainly from the main DX11 logic. In this way the register represents the channel's view of the Bus-In. The output of these flops is enabled to the Bus-In lines either whenever OPLI=1 or whenever a fast CU-Busy is in progress. The register is normally read only, but for maintenance purposes programmed modification is permitted except for CLKO and PARI. When the DX11 is off-line, the data read by PDP-11 programmed references come directly from DXMI. When the DX11 is on-line, the data read by the PDP-11 program comes from the Test-In plug. Correct data will be seen then only if the plugs are cabled together.



BITS		
15-08	CONI	Buffered Control Lines in Register This byte contains the following signals:
Selection Control Lines		
15	OPLI	Operational-In
14	SELI	Select-In (not direct cleared)
13	REQI	Request-In
Tag Lines		
12	ADRI	Address-In
11	STAI	Status-In
10	SRVI	Service-In
Test Lines		
09	CLKO	Clock-Out signal from Bus-Out This line always comes from the cables, even when the DX is off-line.
08	PARI	Bus Parity-In (not a flip-flop—output of parity generation for BUSI)
Data Lines 07-00	BUSI	Buffered Bus-In Data Register This byte contains the data that is enabled to the Bus-In lines for transmission back to the IBM 360 channel. The output of this byte also is always input to a parity generator that produces the signal PARI.

DXCB	CONTROL BITS This is a 16-bit register containing control bits such as LOCKO, phase flip-flops and time state. It is a read only register used by the diagnostics to determine the condition of the DX11B.
DXND	NON-PROCESSOR REQUEST (NPR) DATA This is a 16-bit register to/from which NPR data moves. It is readable for diagnostic purposes.

DXES1		EXTRA SIGNALS This is a two-byte register containing the tumble table index and some miscellaneous signals. The odd byte contains the Tumble Table Index (TTNDX). It indicates the word-pair address for the next tumble table entry to be made by the DX11B. The even byte (MISC) contains miscellaneous control signals for DX11B maintenance diagnostic purposes.
15-08	TTNDX	Tumble table index byte This byte is the low order address of the tumble table entry to be used next. It is shifted left before being copied into the DXBA.
07-00	MISC	Miscellaneous control signals byte
07		Reserved
06	ODD	Copy of DXBA(00) (for future use)
05	NPRTO	NPR latency error. Bus grant not received within timeout interval.
04	DXTO	Program response latency error. While OPL-IN was up, the program did not interact for a 5-second period.
03	TIMDIS	Set to disable DXTO during program debugging.
02	SOSIEN	Fast NPR test enable. Cause simulated SRVO to follow SRVI.
01	MCLKEN	Maintenance clock enable When this bit is set the DX does not change time states until MCLKP is set:
00	MCLKP	Mainenance clock pulse If MCLKEN is set, setting this bit causes the DX to enable the next time state. One normal clock pulse will be issued with each setting of MNCLKF, the Clock pulse thus generated will reset the MNCLKF. The UNIBUS interface continues to run at normal speed at all times. Maintenance clock mode cannot be entered when on-line.
DXMOB		MAINTENANCE-OUT BUFFERED This is a 16-bit register containing the buffered setting of the Control-Out and Bus-Out lines. It is read only. It consists of the flip-flops that are set when the program writes to DXMO. When the DX11B is cabled back to itself and on-line, a comparison of the contents of DXMOB and DXMO is an indication of the operational condition of the DX11B drivers and receivers and of the IBM cables.
DXES2		EXTRA SIGNALS Contains interval signals for DX11B maintenance diagnostics.
15-02		Reserved
01	DSCRSP	Disconnect response Hardware controlled latch that enables 'Fast CU busy' and 'Propagate Select Out' during Phases 4 and 7. Immediately after an Interface Disconnect. This allows Operational-In to be dropped in Phases 4 on an Interface Disconnect to meet the 6 μsec timeout requirement.
00	IRS	IBM reset conditions stored Hardware controlled latch to cause an IBM reset condition (Interface Disconnect, System Reset, Selective Reset) to be stored in the tumble table when that condition was recognized during Phase 4 or 7.

Status Pointer Word (SPW) and Device Status Table (DST)

The Status Pointer Word (SPW) is accessed by the DX11B during an Initial Selection Sequence (ISS) to determine the status of the 360/370 device selected. During the ISS, the DX11B must present status for the command issued by the 360/370 for the device indicated by the DX11B when it presented Address-In. The SPW is used to perform that function.

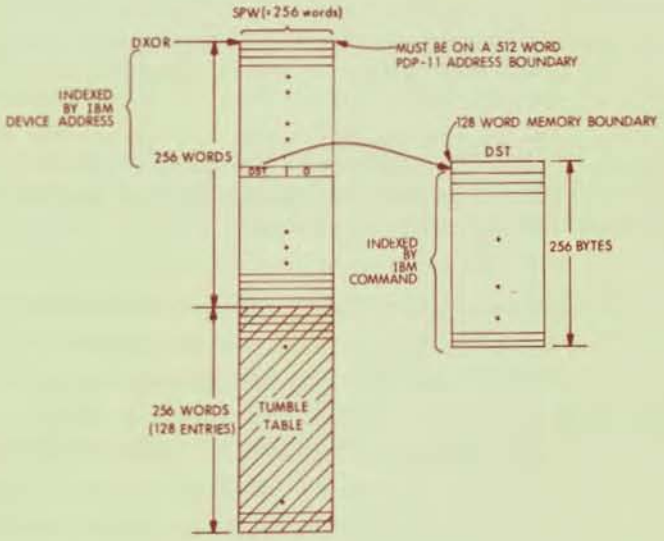
The SPW is a word table indexed by the Control Unit Offset Register (CUOR). A 256-word memory area is reserved for the SPW. Word 0 of the SPW contains the SPW entry for device 0, word 1 for device 1, . . . , word 255 for device 255. The SPW is always on a 512 word PDP-11 memory boundary.

The SPW is a two-byte entry. The even byte contains channel status information. When an access is made to the SPW, the even byte is loaded into the Control Unit Status Register (CUSR). The odd byte contains either 0 or an offset value. If=0, the content of CUSR is the status response to all commands for this device. No further NPR operations are required for status. If the odd byte contains an offset value and the even byte is zero, the odd byte is used as the pointer to a 256-byte Device Status Table (DST). The 360/370 command is used as an index into this table to retrieve a unique status for that command. The DST status is retrieved if the SPW even byte is zero and the odd byte is other than zero. If the status byte is retrieved, it is loaded into the CUSR to become the status byte that will be presented to the 360/370. Each device can have the same, a unique, or no DST, as determined by the requirements of the control unit and devices being emulated. For each DST specified, 128 words must be reserved. The meaning of each status bit is 360/370 device-dependent. The DST is always located on a 128 word PDP-11 memory boundary.

Tumble Table (TT)

This is the name given the circular buffer in which the DX11 stores an entry at the end of each interface activity. The interface activity can be channel (360/370) or Control Unit (PDP-11 emulation) initiated.

A 256-word area is reserved for the TT. Each entry consists of two words. The content of the two words will be the value of DXDS and DXCA, respectively, at the time the channel activity occurred. The entry is guaranteed to be other than zero. TTNDX is the index to the next TT entry to be used by the DX11B. The program must zero the slot after using it. Each entry made by the DX11B causes an interrupt and sets the DONE bit. The program can service all table entries on each interrupt. The program must keep its own TT pointer. The TT is physically located in the next 256 words above the 256 word PSW table.



DX11B PROGRAMMING TECHNIQUES

Interrupt and Tumble Table Servicing

Because the DX11B must respond to Initial Selection Sequences every 32 microseconds, a tumble table is used to indicate when one occurred. An interrupt is generated when the entry is made. If the program permits an interrupt on each entry, it will be impossible to service an interrupt in 32 microseconds. Besides, a subsequent table entry may supersede an operation specified by a previous entry. Therefore, it is desirable to service all tumble table entries when an interrupt occurs. The recommended interrupt procedure is:

Clear DONE bit.

Service all tumble table entries until a zero entry occurs, clearing the TT entries as they are processed.

Build a queue of operations to be performed.

Load the registers (DXCS, DXOS, DXBA and DXBC) with the required data to perform the first operation: set the GO bit.

As the queue of operations is built, a subsequent TT entry may cancel one of the operations in the queue. It is important to be able to delete an operation from this queue.

It must be remembered that the 360/370 channel is always master and the control unit cannot perform a function the channel has told it to terminate.

NOTE: Under no circumstances should the Interrupt Enable bit of the DX11B or any other PDP-11 peripheral be cleared while the DX11B is on-line. Otherwise a NO-SACK timeout may occur which in turn can cause DX11B NPRTO error.

Byte Multiplexed Versus Burst Mode

NPR operations are used for data transfers, Burst mode is defined as any data transfer longer than 32 microseconds. When byte multiplexed mode is desired, the program must break up a block of data into 4-byte segments and initiate a Control Unit Initiated (CUI) operation for each segment. CUI is initiated by raising the REQUEST-IN control line. DXBC should never be loaded with a value greater than minus 4 if multiplexed mode is desired.

Tumble Table Overflow Detection

Programming is used to detect Tumble Table Overflow. The TT can hold 128 entries. This will normally be more than enough. However, it is possible for an overflow to occur.

Overflow is defined as follows:

The program has a pointer to the tumble table entry it serviced last. When it uses the contents, it zeros the entry. When it is ready to service another entry, overflow has occurred if the previous entry (still indicated by program pointer) is other than zero.

Overflow is an irrecoverable error. The program should present UNIT CHECK status for each active device and go off-line.

PROGRAMMING CONSIDERATIONS

This paragraph contains a discussion of some of the general programming considerations that pertain to the DX11-B.

Hardware/Software Interlock

Because of a contention situation that can arise when both the channel and the control unit (through software request) attempt to use the interface, an interlock mechanism is necessary to protect information used by both parts of the system. The control unit can appear busy to channel activity when the control unit software must use the facilities. After system reset, while table initialization is in progress, the CUBSY flip-flop is set.

The general solution is simply to let CH requests always override CU requests. This is done by the LOCKO flip-flop which prevents further changes to the DX11-B registers once a selection sequence has begun. The program can later examine the interrupt conditions to determine if the program requests must be repeated. LOCKO gets DXCS, DXCA, DXOS, and DXBA. Only DXBC remains program-writable.

Boundary Considerations

a. The SPW, TT firmware is 512 words long and must begin on a 2000_8 address boundary

b. All DSTs are 256 bytes long and must begin on a 400_8 address boundary.

c. All data transfers should begin on an even boundary. On input operations, the following will occur for odd BA and odd BC, respectively:

   1. When starting an input on an odd address, the *previous* even address byte is clobbered.
   2. When ending an input on an even address, the *following* odd address byte is clobbered.

d. On output operations, as many as two words *following* the end of the data buffer can be pre-fetched. Therefore, buffers should not be assigned at the *end* of core. This prevents spurious NXMs.

Interrupt Request

When the DX11-B requires either a program interrupt or tumble table service, it sets the DONE bit, leaving LOCKO set. When the program is ready to try a CUI, it must clear DONE; then, if no new selection is in progress, the DX11-B will clear LOCKO.

NOTE
Clearing INTEN is discouraged during DX11-B operation.

The following rules should be followed:

a. The TT entry should be zeroed after being serviced.

b. On an INT, *all* nonzero TT entries should be serviced before dismissing INT (RTI).

c. Software should keep a pointer to current TT entry. This should follow the hardware pointer in relieving the entries the hardware places (hardware will guarantee a nonzero TT entry).

d. No software requests for data transfer or status should be made until all TT entries are serviced.

e. *Before* each TT entry is serviced, DONE should be cleared with a

BIC # DONE, DXCS

thus, the general INT service procedure is:

1. Clear DONE.
2. Service current TT entry. Update action to be performed for device whose address is in TT entry. Do not request data transfer or status at this time. (Note that this may *cancel* a previously queued request for this device.)
3. Clear TT entry.
4. Bump software pointer to next TT entry.

NOTE

If TT entry not 0, go back to Step 1. If it is 0, proceed.

5. When a zero TT entry is encountered, initiate *last* action pending for each device.
6. Dismiss interrupt (RTI).

Data Transfer

Data transfer sequences (DT) are always initiated by the DX11-B program. It is a software responsibility to ensure that a DT is valid at the point requested. Information supplied to the hardware includes Buffer Start Address (DXBA), byte Count (DXBC), Device Address (CUAR), and I/O direction (FCTN—input or output). The hardware will get control of the I/O interface and transfer the data in a single burst, after which it will generate a Data End interrupt. The last bit set is the GO bit. If LOCKOUT is set at this time, the effect is a NO OP and the result is no data transfer.

Several other events can happen as follows:

a. A bus out parity error can occur, setting PARER. This will terminate data transfer with PARSTP.

b. A timeout reference can occur, setting NXM.

c. The channel can indicate I/O stop, setting CHDEND.

d. An Interface Disconnect can occur.

If a CUI is used, it could be overridden by an ISS, in which case an interrupt would occur; but different bits would be set in the DXDS and copied into the tumble table.

Status Presentation

There are several cases in which presentation must be initiated by the program as follows:

a. When stack status is indicated by the channel. In this case, the DX11-B will automatically request presentation of the status again, until it is subsequently relieved or overridden via an ISS (only if BSYEN=1).

b. When ending status is initially available for the device (DEVEND, etc.).

c. When asynchronous status becomes available for the device (DEVEND or ATTEN).

d. When a device that had previously been interrogated while busy becomes free (CUEND or DEVEND).

e. At the termination of a data transfer (CHEND or CHEND+DEVEND).

The program loads the status and device address and requests status presentation (FCTN=3). If the status is accepted, the device status, device address and command (if any) are loaded into the tumble table and an interrupt is generated.

On-line/Off-line Control

The ONLINA flip-flop is written by the program to request a change of on-line status. The ONLINB flip-flop, in DXCB, can be read to verify that the transition occurred. The program clears the ONLINA flip-flop to attempt to put the DX11-B off-line. The ONLINB flip-flop will not clear if any channel activity is in progress.

NOTE

When the program sets ONLINA, the ONLINB flip-flop will only set if the hardware ON-LINE/OFF-LINE switch is in the enable position when channel conditions permit.

An on-line/off-line request can be made at any time. If channel activity occurs at the time ONLINA is cleared, it will not clear. This allows the program a chance to reconsider the off-line request in view of the new CHIS.

NOTE

An On-line/Off-line sequence should not be attempted in an interval less than 10 ms.

ORDERING INFORMATION

DEC No.	Description	Prerequisite
DX11-BA	360/370 channel interface w/115V 60Hz power supply and an H950 standard PDP-11 cabinet	PDP-11 (PDP-11/20 or PDP-11/15 w/ KH11-A Option & Comtex Software)
DX11-BB	360/370 channel interface w/230V 50Hz power supply and an H950 standard PDP-11 cabinet	same

APPLICATIONS

The DX11B can be used for any application where it is desirable to have a PDP-11 as a pre- or post-processor to either a S/360 or S/370. With its ability to emulate a 2848 display or a 2703 or 3705 communications controller, it is compatible with OS/GAM, BTAM, QTAM and TCAM and DOS/BTAM and QTAM. It can be used in systems designed for the following applications:

1. Front-end processors
2. Message switching
3. Store and forward
4. On-line inquiry
5. Data entry
6. Remote batch
7. Display cluster control
8. On-line data enscribers

NOTES

digital

DIGITAL EQUIPMENT CORPORATION, Corporate Headquarters: Maynard, Massachusetts 01754, Telephone: (617) 897-5111

SALES AND SERVICE OFFICES

DOMESTIC — ARIZONA, Phoenix and Tucson • CALIFORNIA, Los Angeles, Monrovia, Oakland, Ridgecrest, San Diego, San Francisco (Mountain View), Santa Ana, Sunnyvale and Woodland Hills • COLORADO, Englewood • CONNECTICUT, Fairfield and Meriden • DISTRICT OF COLUMBIA, Washington (Latham, Md.) • FLORIDA, Orlando • GEORGIA, Atlanta • ILLINOIS, Chicago (Rolling Meadows) • INDIANA, Indianapolis • IOWA, Bettendorf • KENTUCKY, Louisville • LOUISIANA, Metairie (New Orleans) • MASSACHUSETTS, Marlborough and Waltham • MICHIGAN, Detroit (Farmington Hills) • MINNESOTA, Minneapolis • MISSOURI, Kansas City and St. Louis • NEW HAMPSHIRE, Manchester • NEW JERSEY, Fairfield, Metuchen and Princeton • NEW MEXICO, Albuquerque • NEW YORK, Albany, Huntington Station, Manhattan, Rochester and Syracuse • NORTH CAROLINA, Durham/Chapel Hill • OHIO, Cleveland, Columbus and Dayton • OKLAHOMA, Tulsa • OREGON, Portland • PENNSYLVANIA, Philadelphia (Bluebell) and Pittsburgh • TENNESSEE, Knoxville • TEXAS, Austin, Dallas and Houston • UTAH, Salt Lake City • WASHINGTON, Bellevue • WISCONSIN, Milwaukee (Brookfield) • INTERNATIONAL — ARGENTINA, Buenos Aires • AUSTRALIA, Adelaide, Brisbane, Canberra, Melbourne, Perth and Sydney • AUSTRIA, Vienna • BELGIUM, Brussels • BOLIVIA, La Paz • BRAZIL, Puerto Alegre, Rio de Janeiro and São Paulo • CANADA, Calgary, Halifax, Montreal, Ottawa, Toronto and Vancouver • CHILE, Santiago • DENMARK, Copenhagen • FINLAND, Helsinki • FRANCE, Grenoble and Paris • GERMANY, Berlin, Cologne, Hannover, Hamburg, Frankfurt, Munich and Stuttgart • HONG KONG • INDIA, Bombay • INDONESIA, Djakarta • ISRAEL, Tel Aviv • ITALY, Milan and Turin • JAPAN, Osaka and Tokyo • MALAYSIA, Kuala Lumpur • MEXICO, Mexico City • NETHERLANDS, Utrecht • NEW ZEALAND, Auckland • NORWAY, Oslo • PHILIPPINES, Manila • PUERTO RICO, Santurce • SINGAPORE • SPAIN, Barcelona and Madrid • SWEDEN, Gothenburg and Stockholm • SWITZERLAND, Geneva and Zurich • TAIWAN, Taipei and Taoyuan • UNITED KINGDOM, Birmingham, Bristol, Dublin, Edinburgh, Leeds, London, Manchester and Reading • VENEZUELA, Caracas • YUGOSLAVIA, Ljubljana •

PRINTED IN USA ED 04088 75 03A/68 M 14 05

DATE: 12/11/79
PROJECT: DDCMP MULTIPLEXER
CHARGE NO: M416-E0300-37622

SYSTEM DESIGN SPECIFICATION

AND

USERS GUIDE

REV 1.1
BY
D.E. KORF

TABLE OF CONTENTS

1.0 INTRODUCTION

1.1 PURPOSE

The purpose of this document is to provide a design overview of KMC DDCMP multiplexer and its interaction with a host PDP11 processor. It is also intended to provide the PDP11 user with the required knowledge to interface the PDP11 host to the KMC multiplexer.

1.2 REQUIREMENT SUMMARY

Detailed below is a summary of the Requirements, Goals and non goals of the DDCMP multiplexer. The Functional Specification should be referred to for a complete definition of multiplexer functionality.

KMC DDCMP MULTI-LINE CONTROLLER

* WHAT ARE THE MAJOR OBJECTIVES OF MULTIPLEXER?

- Provide 8 lines at full or half duplex with speeds up to 9600 BPS.
- DDCMP communications protocol implemented in firmware providing high reliability and throughput with low host processor overhead.
- Provide communications between KMC11 and other synchronous interfaces that support DDCMP protocol in a point to point environment.
- 16 Bit NPR (DMA) transfers for minimum interference with host processor operation.
- Firmware based on KMC11-B so that the firmware is loadable.

* WHAT ARE ITS GOALS?

- Line level compatability with present DMC11.
- Support greater than seven (7) outstanding messages.

* WHAT THE MULTIPLEXER IS NOT

- Not a DMC11 modified to JUST scan 7 more lines.
- Will not provide support of line speeds greater than 9600 BPS.
- Will not support local on board diagnostics. (Note: Since KMC is loadable, diagnostics can be separately loaded into the KMC).
- Will not support automatic recovery from powerfail.

* DESIGN GOALS AND CONSIDERATIONS

- CSR control is the same as COMM IOP-DUP. Note commands differ but interface control is the same.
- CSR transfers from KMC to host PDP11 take priority over CSR transfers from PDP11 to KMC.
- Body of all messages reside in PDP11 host.
- All message headers reside in KMC.
- All inquiry control messages reside in KMC.
- System is primarily a state system.
- The states in the system are controlled by a series of queues, tables and buffers.
- Data reception takes priority over data transmission.
- Attempt to make the system somewhat self regulating (It won't accept more work than what it can handle).

1.3 DESIGN GOALS AND CONSIDERATIONS

Listed below are some design considerations of the DDCMP multiplexer in relation to other products, interfaces and internal control. The speed calculations define the maximum slice of time or state size that may be executed for each line per character per second.

SPEED CALCULATIONS

KMC = 5,000,000 instructions per second

$$\text{CHARACTER RATE} = \frac{8 \text{ lines X } 9600 \text{ BPS}}{8 \text{ bits per character}} = 9600 \text{ characters/second}$$

$$\text{INSTRUCTIONS/CHAR/SEC} = \frac{5{,}000{,}000 \text{ instr/sec}}{9{,}600 \text{ char/sec}} = 520.8 \text{ instr/char/sec}$$

520.8 instructions/character/sec for 1/2 duplex line.

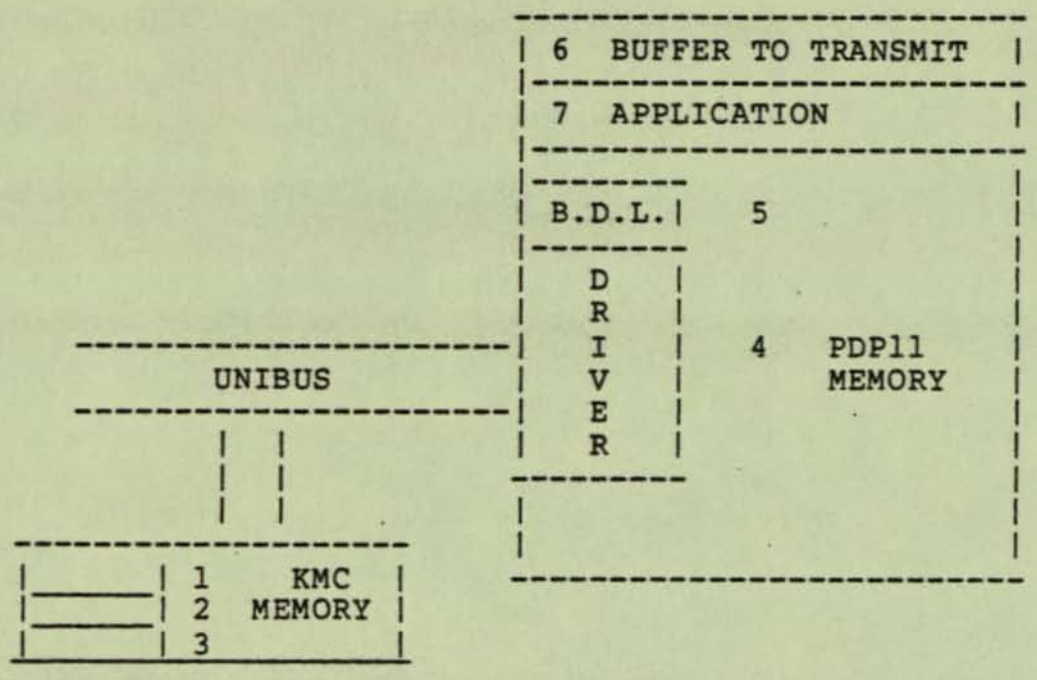
NOTE: The 520.8 instruction/character/sec is for a one way (1/2 duplex) line to arrive at a full duplex two way communication divide 520.8 by 2.

$$\text{FULL DUPLEX} = \frac{520.8}{2} = 260 \text{ instructions/character/second}$$

2.0 USERS VIEW OF SYSTEM

Below is a visual description of the system from the PDP11 users point of view. All major components of the system are identified.

VISUAL DESCRIPTION OF SYSTEM



1 - LINE STATUS TABLE IN KMC (LST)
CONTAINS CURRENT LINE DEFINITIONS

2 - LINE OUTPUT STATE TABLE IN KMC (LOST)
CONTAINS CURRENT STATE, HEADER AND
NEXT CHARACTER INFO

3 - INTERMEDIATE BUFFER IN KMC (IB)
CONTAINS BUFFER OF 8 CHARACTERS
TO BE TRANSMITTED

4 - DRIVER INTERFACE IN PDP11

5 - BUFFER DESCRIPTOR LIST AREA IN PDP11

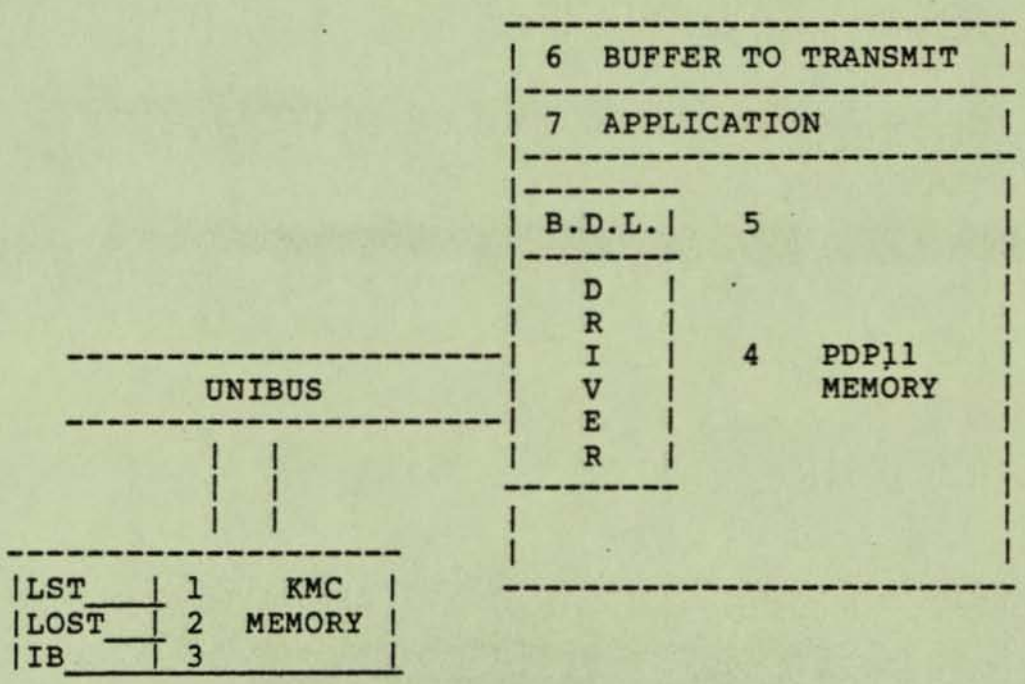
6 - DATA BUFFER TO BE TRANSMITTED

7 - APPLICATION PROGRAMS

2.1 USERS VIEW OF WHAT HAPPENS

The following is a description of the general processes that occur from a users point of view in the transmission of a message. The reception of a message has a similar process to that of transmission.

WHAT HAPPENS



Application program (7) passes to driver (4) a request for a message to be transmitted. It specifies the BDL MEMBER (5) that contains the address and character count of the buffer to be transmitted.

The driver (4) passes the information to the KMC.

The KMC first builds the header information in the L.O.S.T. Table (2).

The KMC then retrieves the BDL Member (5) which defines where in PDP memory the data buffer (6) is and how big it is.

The KMC then retrieves the first eight characters of the data buffer (6) and places them in the intermediate buffer (1).

The KMC then transmits the header information from the L.O.S.T. (6) table and updates the line status table (3).

After the header has been transmitted, a header CRC is transmitted.

Following the header CRC transmission, a character is retrieved from the intermediate buffer (1) and transmitted and a character from the data buffer (6) is requested to take its place (remember 7 other characters will be transmitted before the character requested is transmitted). The L.S.T. (3) is updated as is required.

The previous step is repeated until all characters have been transmitted. the data CRC is then generated and the LOST (2) is updated.

Upon reception of an ACK or NAK, (after seven retries), the BDL (5) member is set as free and a notification is sent to the driver (4) as to what has happened and the KMC tables and buffers 1,2,3 are updated.

The driver (4) then passes the KMC response status to the application program.

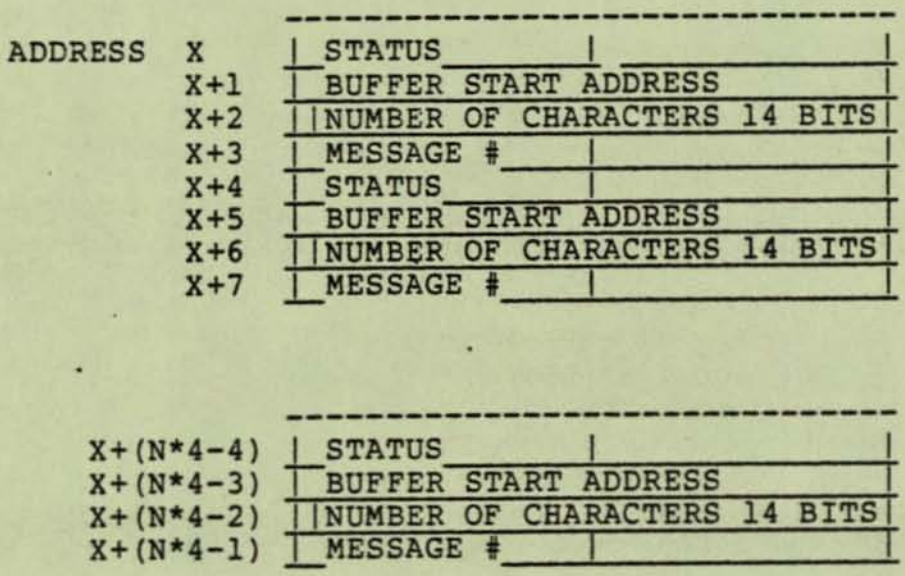
3.0 PDP11 INTERFACES

This section discusses the interfaces between the KMC and PDP11. There are three areas of interaction. The Buffer Descriptor List Definition Table in the PDP11, CSR Input Commands to the KMC from the PDP11 and CSR Response Commands from the KMC to the PDP11.

3.1 BUFFER DESCRIPTOR LIST (BDL) DEFINITIONS

In the PDP11 an area must be set aside to hold definitions of messages to be transmitted or received. What is a Buffer Descriptor List? It is a four word entry in PDP11 memory that defines to the KMC and the PDP11 where in PDP11 memory a data message is, how many characters are in the message, the line number and tributary of the line the message is to be transmitted or was received on, the DDCMP message number that was assigned by the KMC protocol handler and the status of the four word BDL entry (member). During initialization a BDL base address is passed to the KMC which defines where in memory the BDL Definition table starts. All references to this area of memory after setting the base is by member (four word entry) number. It is up to the user to define the size of this area. Its minimum size must be large enough to handle the number of input and output lines. For an eight line full duplex operation, the minimum size would be 64 words. That is eight lines for input at four words per line and eight lines for output at four words per line, assuming all lines are transmitting and receiving data at the same time. A definition of the Buffer Descriptor list and how it is organized follows.

PDP11 BUFFER DESCRIPTOR LIST DEFINITION TABLE



MESSAGE # = DDCMP Message Number from KMC Controller

STATUS = The status area is for use by the PDP11 for keeping track of which BDLs are in use and assigned to transmit and receive lines which are free.

BUFFER START ADDR = Start Address of Message Data to be Transmitted.

LENGTH = Character Count of Message to be Transmitted.

Length of BDL list is user defined. Maximum of 256 members.

3.2 CSR INPUT COMMAND TO KMC

The KMC and PDP11 communicate with each other via Control Status Registers (CSRs). Since both the KMC and PDP11 can read and write the CSRs, a control mechanism is required to prevent one CPU from writing while the other CPU is reading.

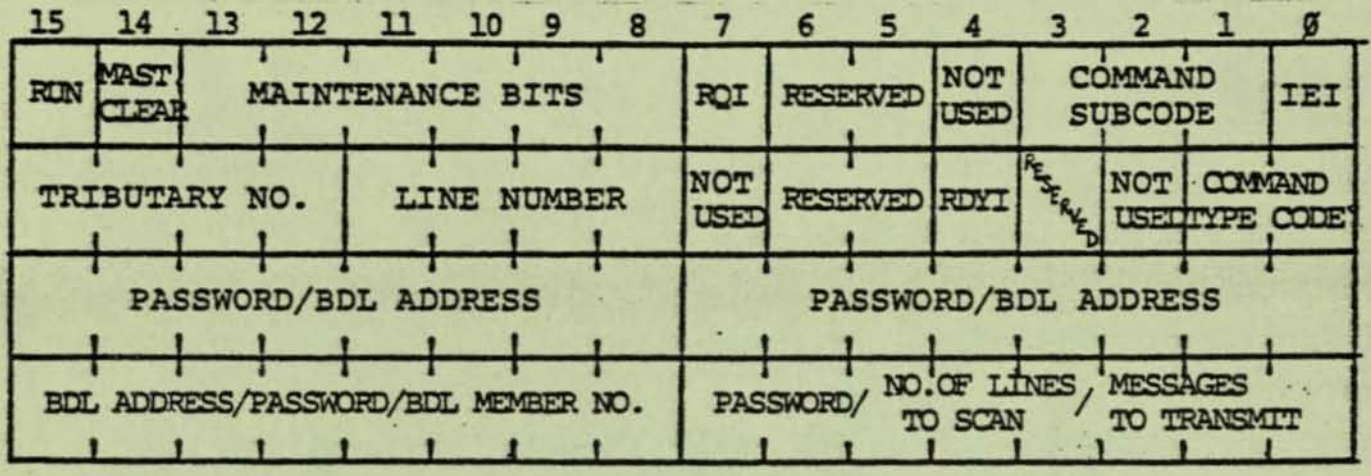
All input commands to the KMC are issued by an application program to a driver program which is completed in a series of steps. The driver program sets RQI to request the use of the CSR for transfer of data and then waits for the KMC to set RDYI. This wait can be implemented through a delay loop or can wait for an interrupt. The delay loop is not recommended. The KMC will then set RDYI when it is ready to accept an input. After the RDYI has been set by the KMC.

The PDP11 driver loads the CSRs with the command and its associated parameters. When the parameters have been loaded into the CSR, the RDYI is cleared to inform the KMC that the parameters may be read.

There are four (4) major input command types to the KMC. The command types and CSR structures are defined in the following figures.

KMC CONTROL INPUT COMMANDS

GENERAL FORMAT



COMMAND TYPES

COMMAND VALUE	COMMAND DESCRIPTION
0	Master Control
1	Line Control
2	Message Control
3	Status

MASTER CONTROL COMMANDS

COMMAND VALUE	COMMAND SUB CODE	COMMAND DESCRIPTION	PARAMETERS
0	0	Init System	N/A
0	1	Reset/Set to DDCMP Mode	LN #, TRIB #
0	2*	Enter MOP Mode	LN #, TRIB #, PASSWORD
0	3	Terminate Activity	LN #, TRIB #
0	4	Set Line Scan Length (No. of Lines to Scan)	LN #
0	5*	Set MOP Mode Password for Forced Entry	LN #, PASSWORD
0	6	Set BDL Base Address	BDL ADDRESS, TRANS., RCV.
0	7	Set Modem CSR Address (Note, also enables modem control for all lines.	CSR Address of modem control lower 18 bits

* Currently not planned for implementation.

LINE CONTROL COMMANDS

COMMAND VALUE	COMMAND SUBCODE	DESCRIPTION	PARAMETERS
1	0	Set Line Up	LN # (TRIB #)
1	1	Set Line 1/2 Duplex	LN #
1	2	Set Line Down	LN # (TRIB #)

MESSAGE CONTROL COMMANDS

COMMAND VALUE	COMMAND SUBCODE	DESCRIPTION	PARAMETERS
2	0	Transmit Message	LN #, BDL Member # of messages, TRIB #
2	1	BDL Member for use in Message Reception	BDL Member, LINE #

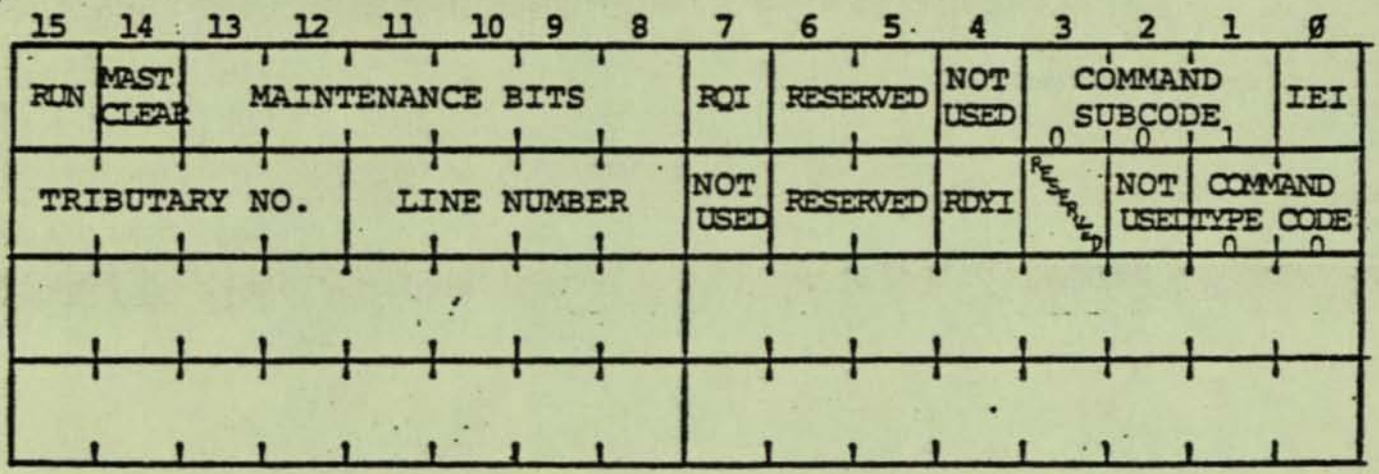
STATUS CONTROL COMMANDS

COMMAND VALUE	COMMAND SUBCODE	DESCRIPTION	PARAMETERS
3	0	Request Line Status	(LN #, (TRIB #1))
3	1	Read KMC Memory	KMC Data Memory Address
3	2	Write KMC Memory (Note an automatic read of the data store will take place for verification.	KMC Data Memory Address & Value to be stored.

*Currently not implemented.

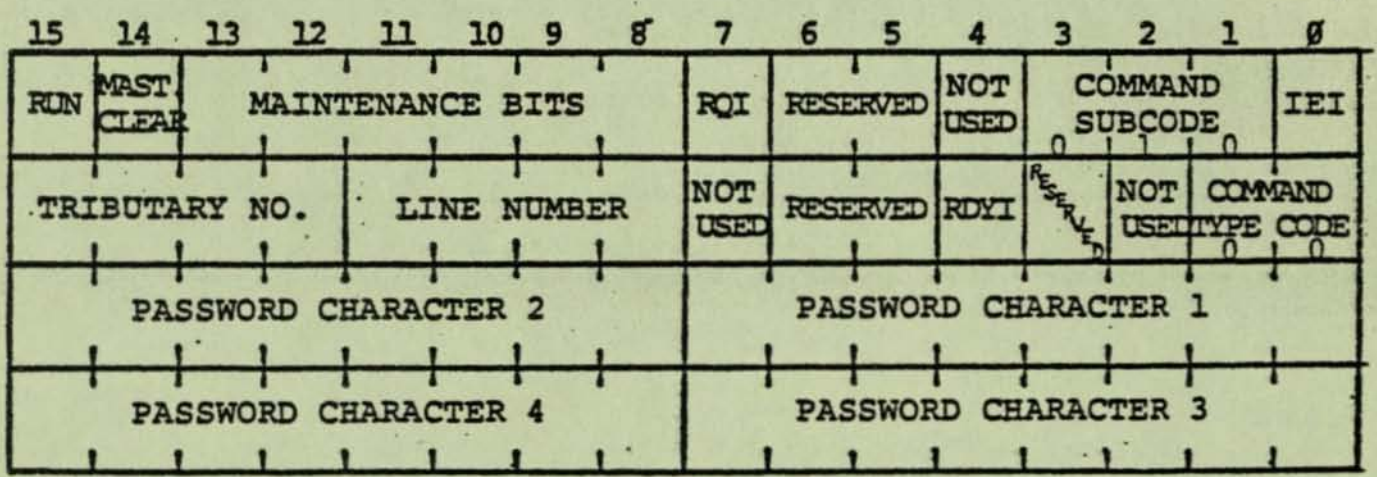
KMC CONTROL INPUT COMMANDS

RESET/SET DDCMP MODE



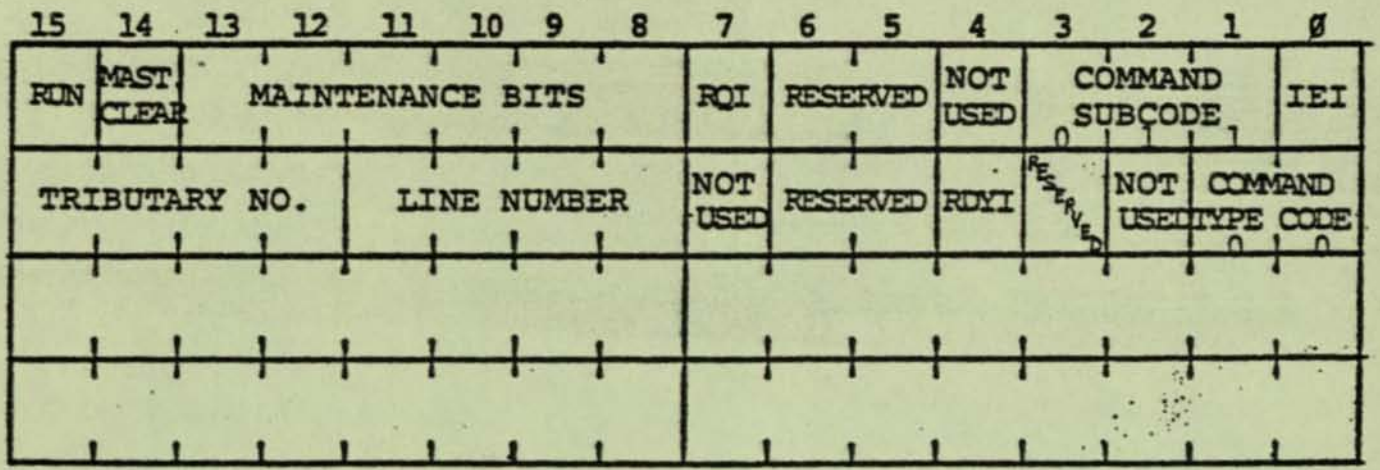
KMC CONTROL INPUT COMMANDS

ENTER MOP MODE



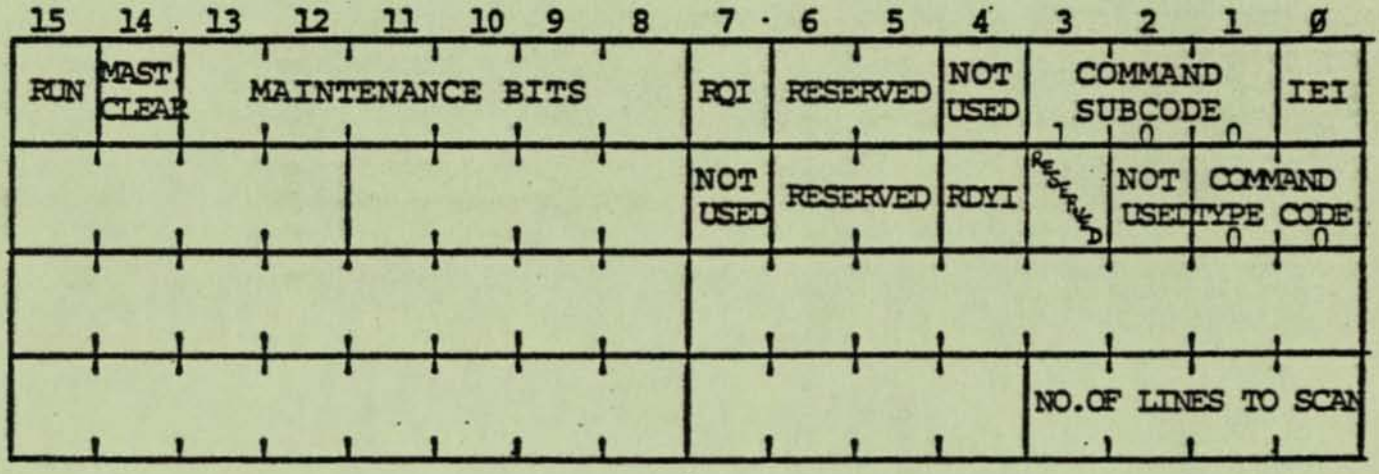
KMC CONTROL INPUT COMMANDS

TERMINATE ACTIVITY



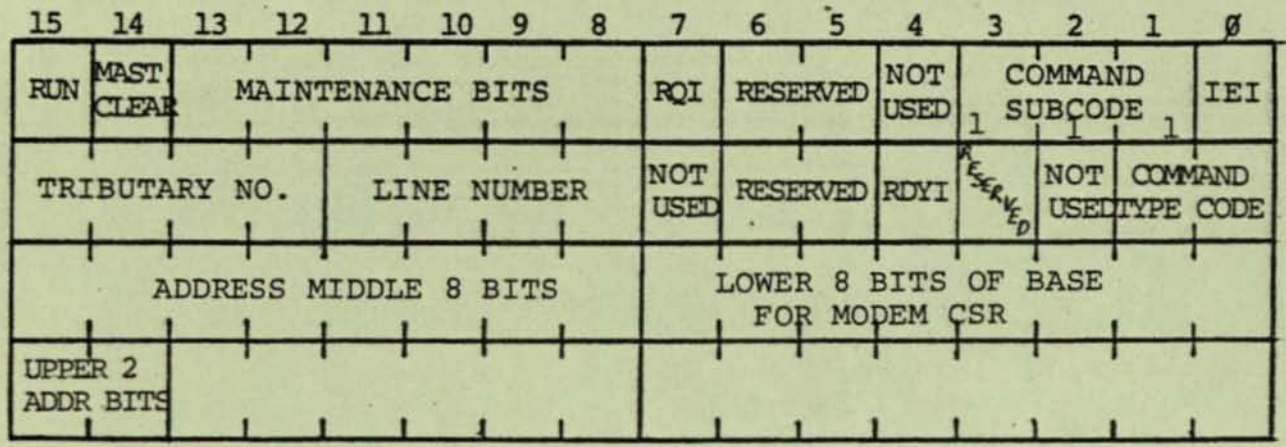
KMC CONTROL INPUT COMMANDS

SET LINE SCAN LENGTH



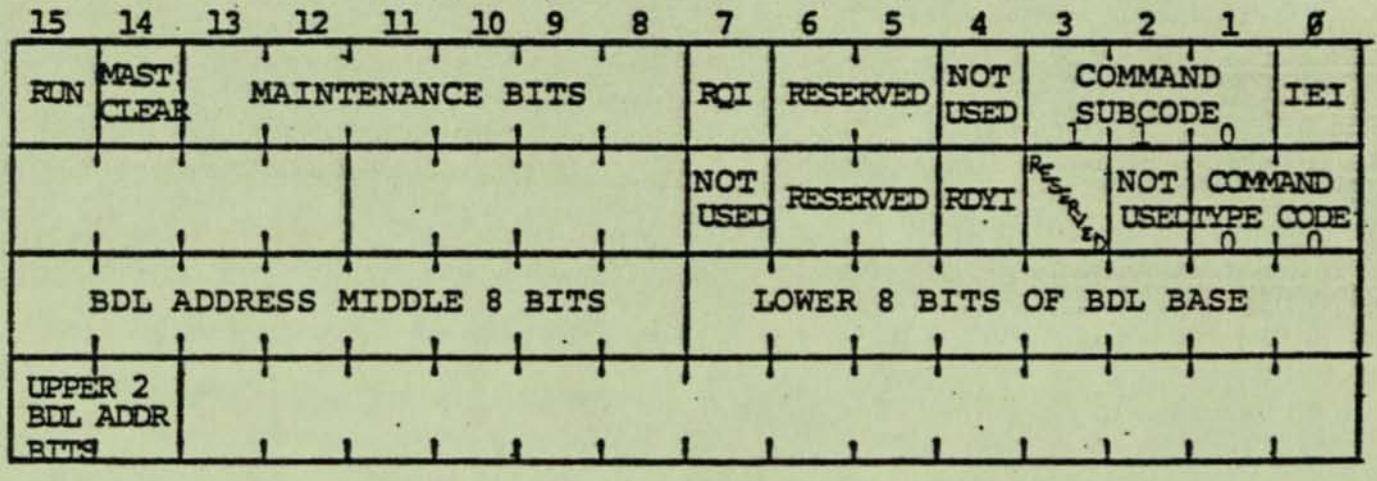
KMC CONTROL INPUT COMMANDS

MODEM CONTROL CSR ADDR



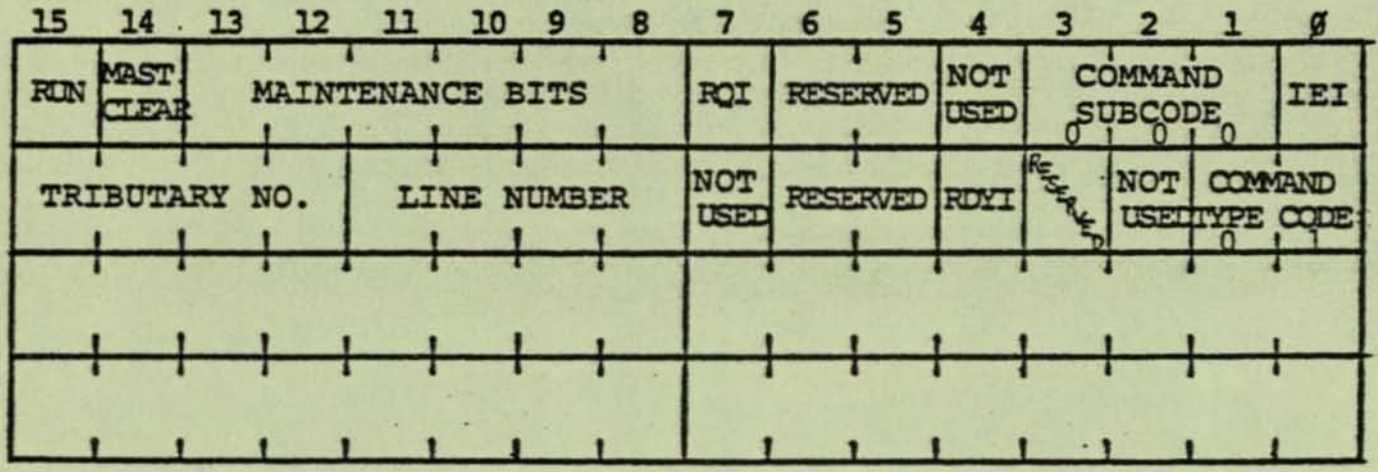
KMC CONTROL INPUT COMMANDS

BDL BASE ADDRESS



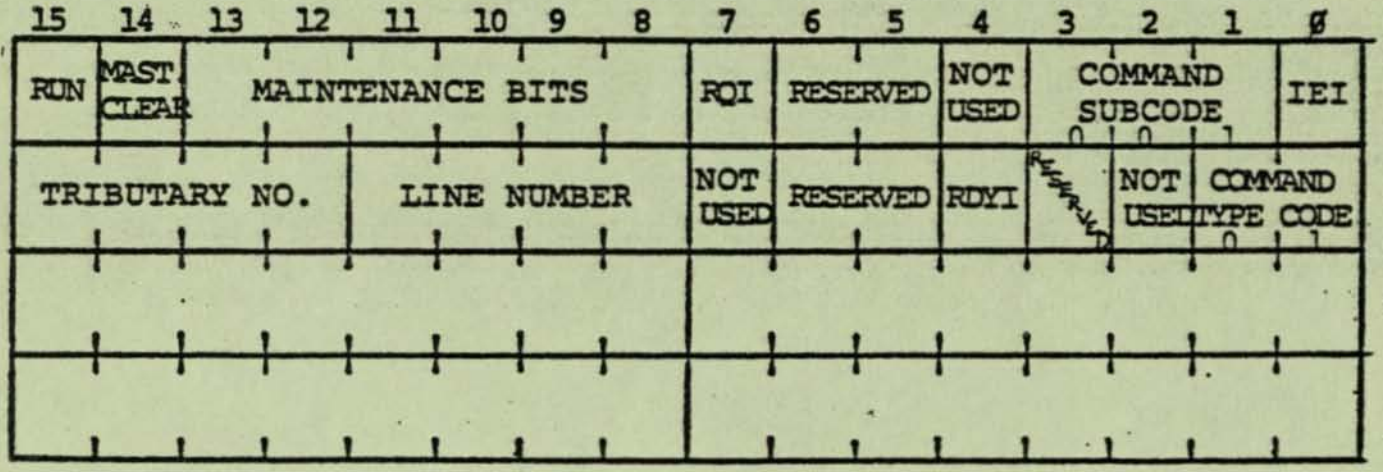
KMC CONTROL INPUT COMMANDS

SET LINE UP



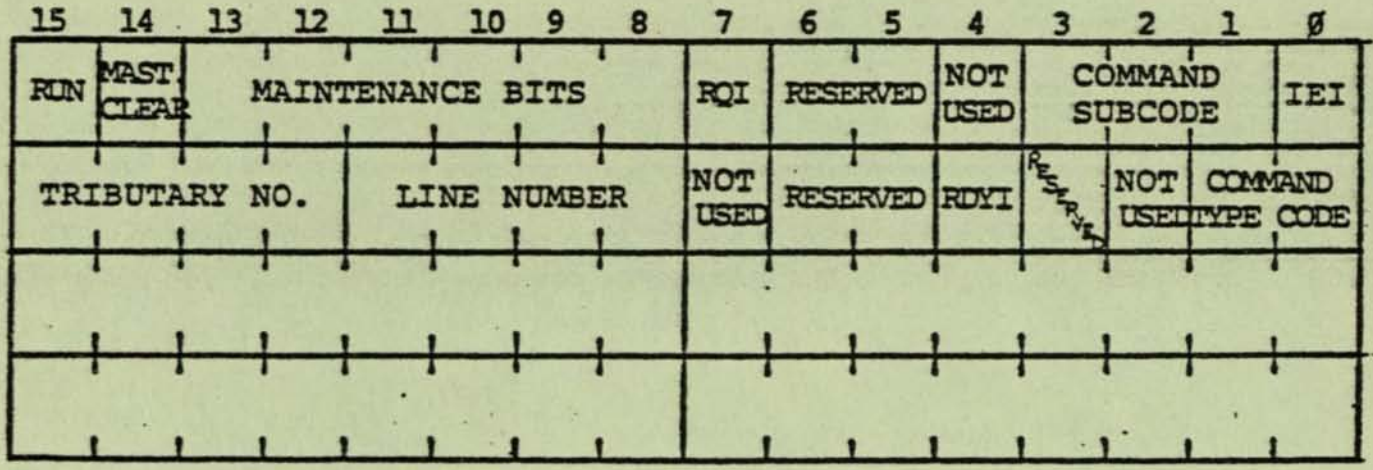
KMC CONTROL INPUT COMMANDS

SET LINE ½ DUPLEX



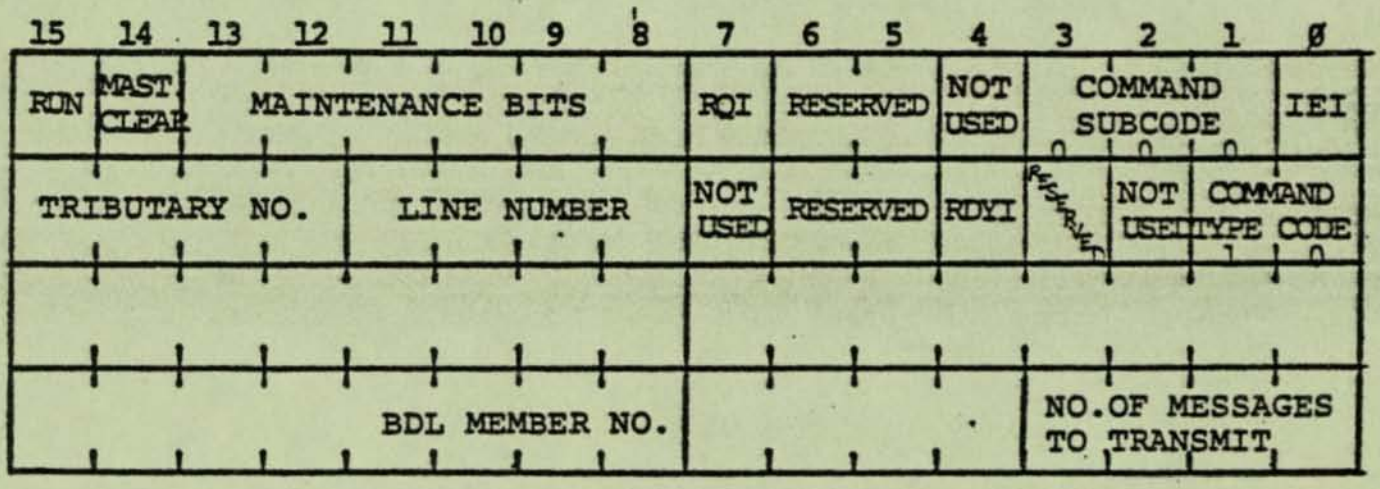
KMC CONTROL INPUT COMMANDS

SET LINE DOWN



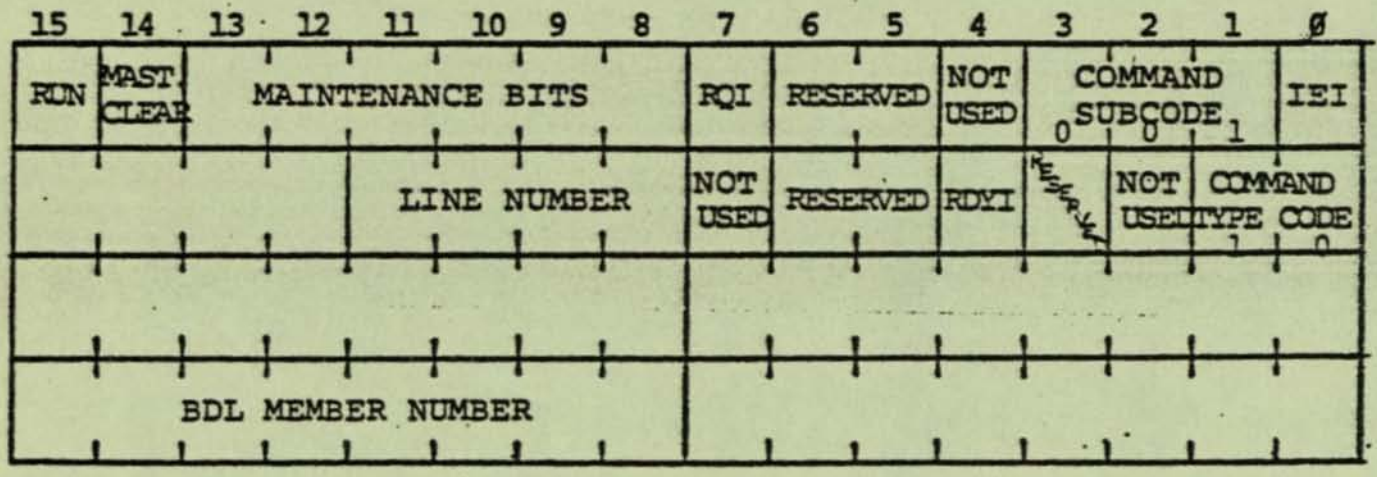
KMC CONTROL INPUT COMMANDS

TRANSMIT MESSAGE



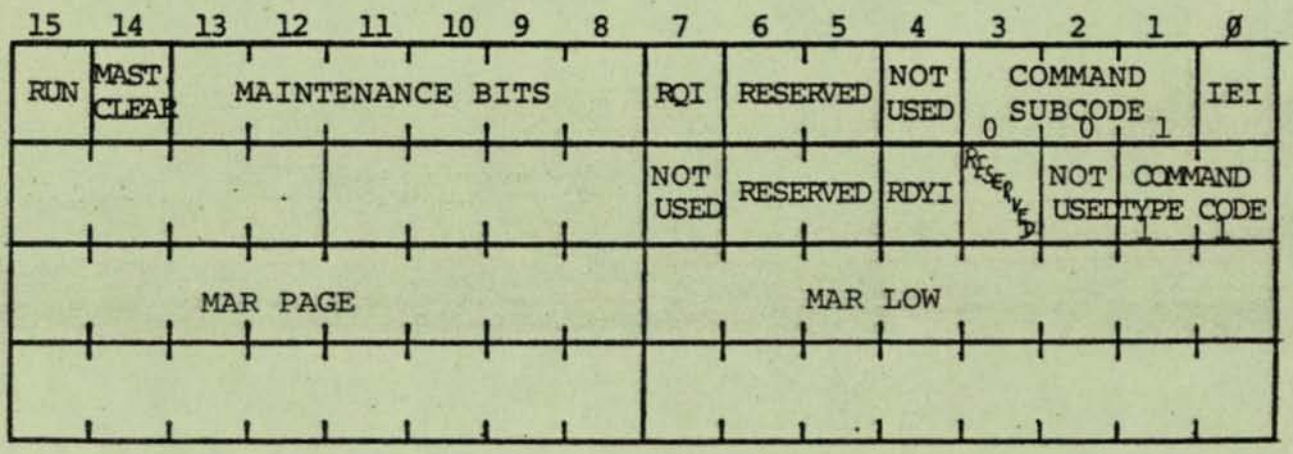
KMC CONTROL INPUT COMMANDS

BDL MEMBER FOR USE IN MESSAGE RECEPTION



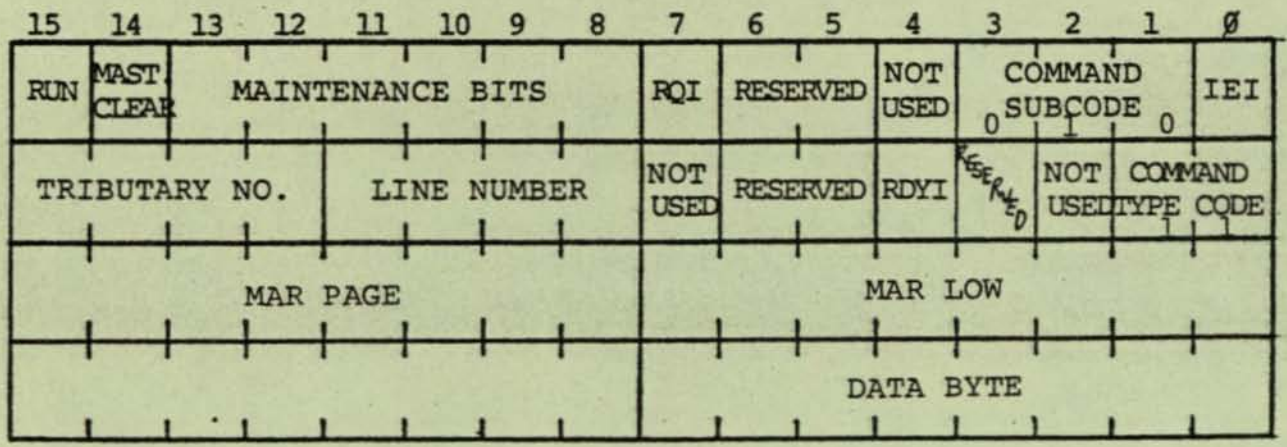
KMC CONTROL INPUT COMMANDS

READ KMC11 DATA MEMORY



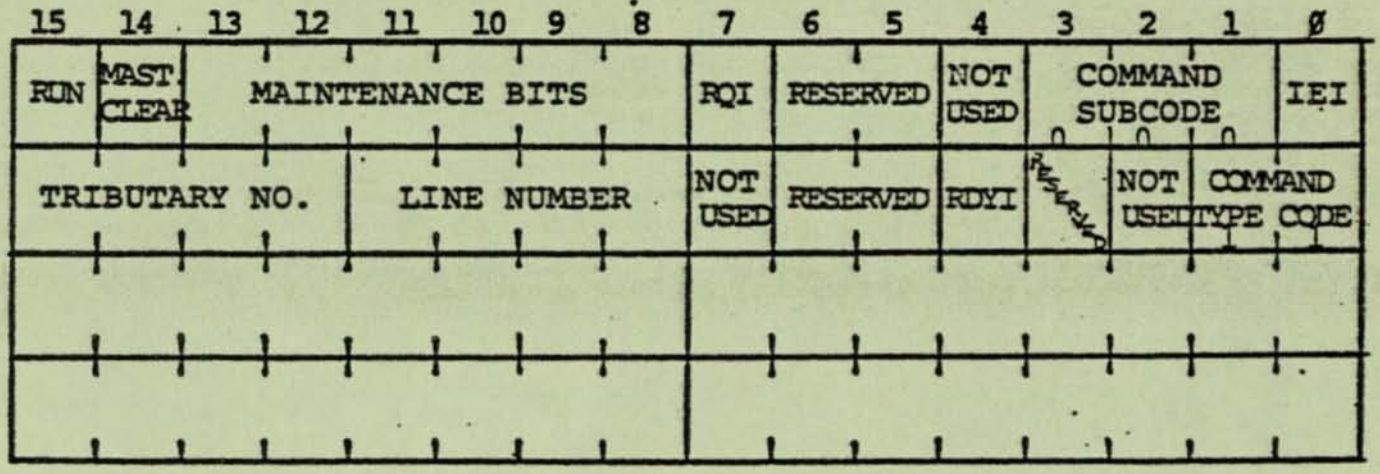
KMC CONTROL INPUT COMMANDS

WRITE KMC11 DATA MEMORY



KMC CONTROL INPUT COMMANDS

REQUEST LINE STATUS



3.3 CSR RESPONSE COMMANDS FROM KMC

Corresponding to the CSR input commands there are a series of output commands. Commands from the KMC to the PDP11 always take priority over an input command request. The method used for implementing output commands are as follows:

The KMC loads the CSRs with the parameters required for a given command;

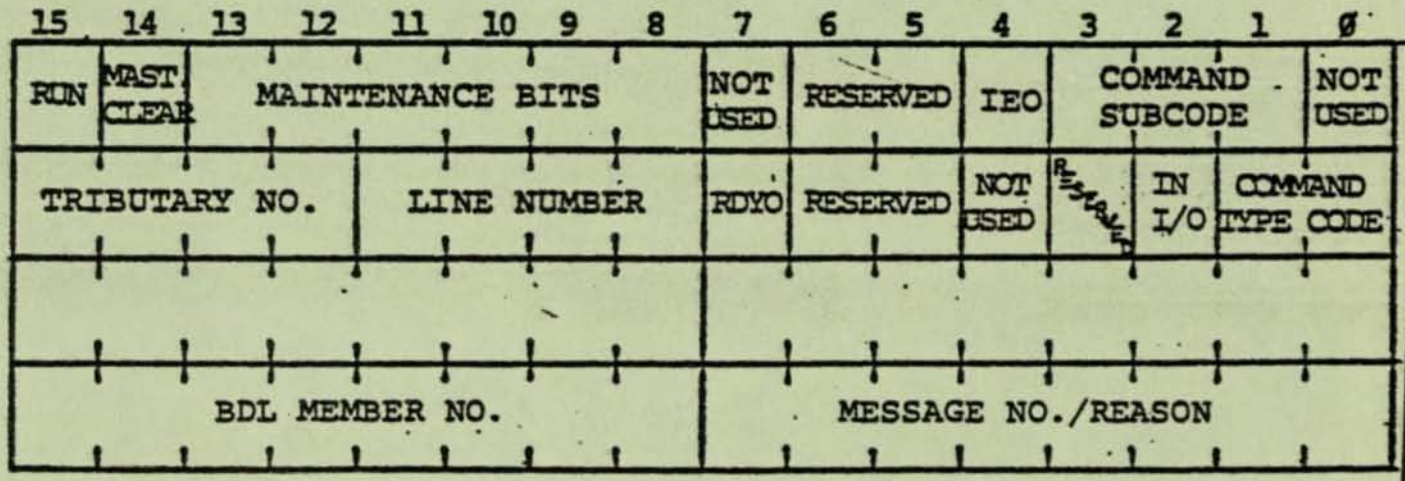
the RDYO bit is set and an interrupt is generated to the PDP11 indicating the PDP11 should read the CSRs;

the driver programs retrieves the CSR data and clears the RDYO bit to indicate the transfer is complete.

There are four (4) major output command types from the KMC. The command types and CSR structures are defined in the following figures.

KMC CONTROL OUT COMMANDS

GENERAL FORMAT



COMMAND TYPES

COMMAND VALUE	DESCRIPTION
0	Positive Responses Control
1	Negative Responses Control
2	Message Reception Control
3	Status Responses (Does Not Use General Format for Out Commands)

POSTIVIE RESPONSE CONTROL

COMMAND VALUE	COMMAND SUBCODE	DESCRIPTION	PARAMETERS
0	0	Init Complete	N/A
0	1	Set/Reset Complete	LN #, TRIB #
0	2	Activity Terminated	LN #, TRIB #
0	3	MOP Mode Entered*	LN #, TRIB 3
0	4	Message Acknolwedged	Message #, BDL #

NEGATIVE RESPONSE CONTROL

COMMAND VALUE	COMMAND SUBCODE	DESCRIPTION	PARAMETERS
1	3	Error Threshold Reached	LN #, TRIB #
1	5	Messaged Naked	LN #, TRIB #, BDL Member
1	6	Returned too many receiver buffers for line	LN #, MEMBER #

MESSAGE RECEPTION CONTROL

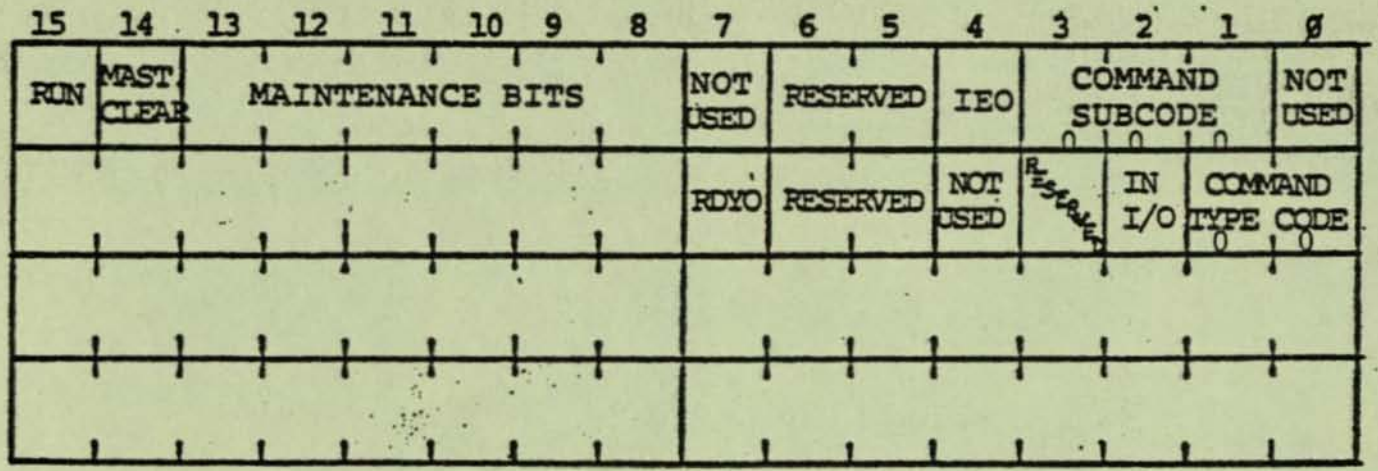
COMMAND VALUE	COMMAND SUBCODE	DESCRIPTION	PARAMETERS
2	0	Message Received	LN #, TRIB #, BDL MEMBER MSG #
2	1	Request BDL Member for Reception of Message	LN #

STATUS

COMMAND VALUE	COMMAND SUBCODE	DESCRIPTION	PARAMETERS
3	0	1st 4 bytes of line status table for specific line.	LIN# TRIB Bytes 0-3 of LST for line.
3	1	Last 4 bytes of line status table for specific line.	LIN# TRIB Bytes 4-7 of LST for line.
3	2	Dump of specified KMC Data Memory	Data Memory Address & Contents

KMC CONTROL OUT COMMANDS

INITIALIZATION COMPLETE

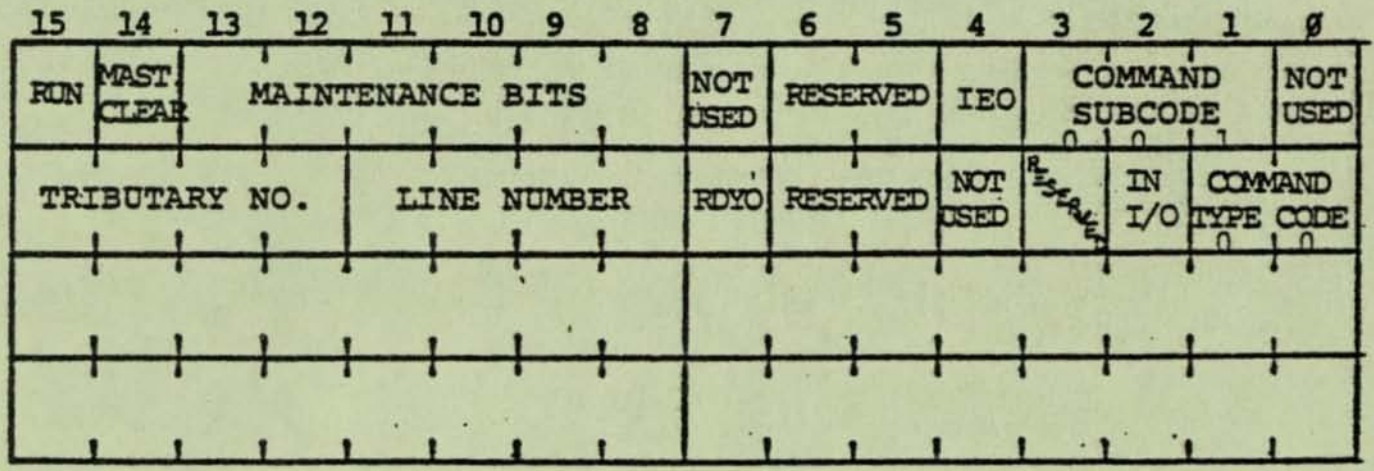


COMMAND TYPES

COMMAND VALUE	DESCRIPTION
Ø	Positive Responses Control
1	Negative Responses Control
2	Message Reception Control
3	Status Responses - (Does Not Use General Format for Out Commands)

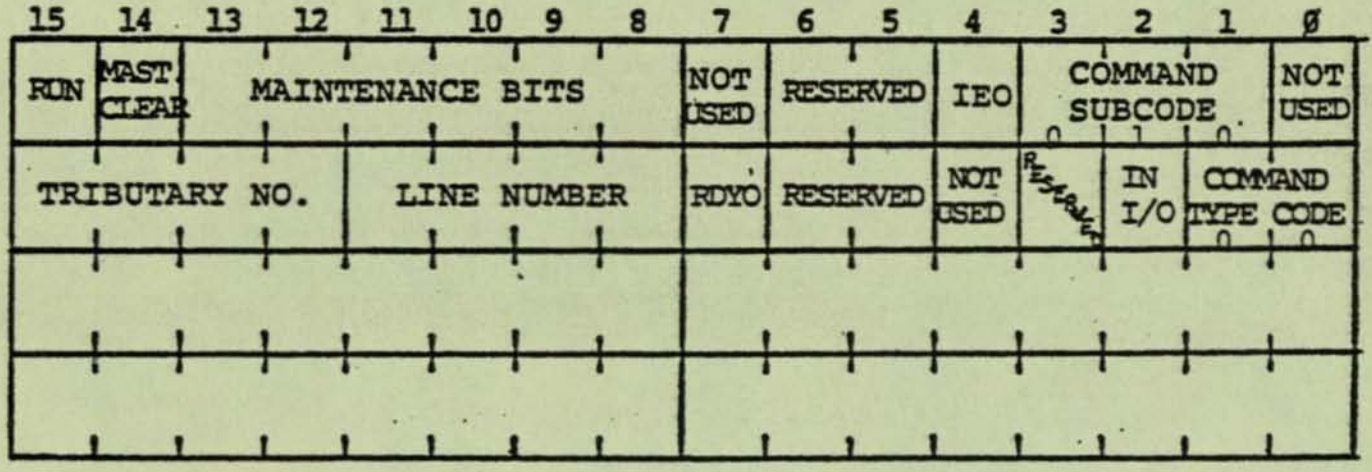
KMC CONTROL OUT COMMANDS

SET/RESET COMPLETE



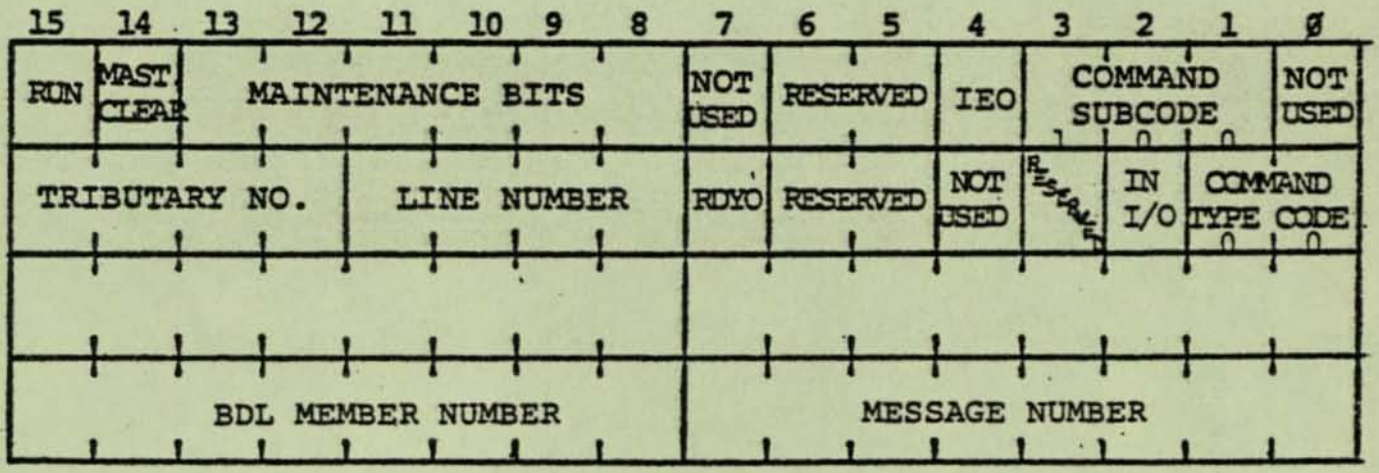
KMC CONTROL OUT COMMANDS

ACTIVITY TERMINATED



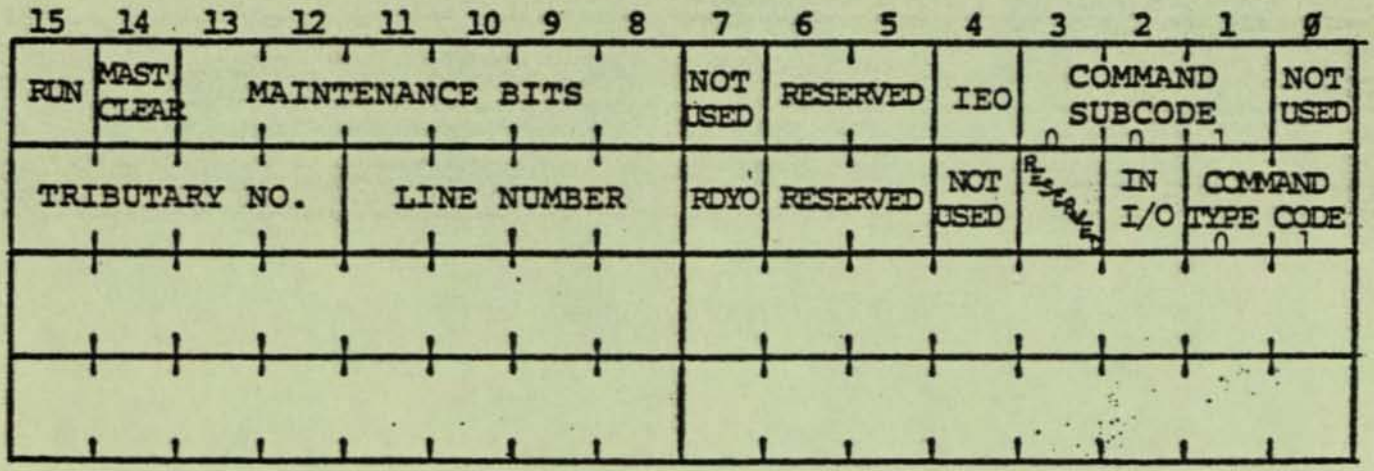
KMC CONTROL OUT COMMANDS

MESSAGE ACKNOWLEDGED



KMC CONTROL OUT COMMANDS

SET/RESET FAILED



KMC CONTROL OUT COMMANDS

ERROR THRESHOLD REACHED

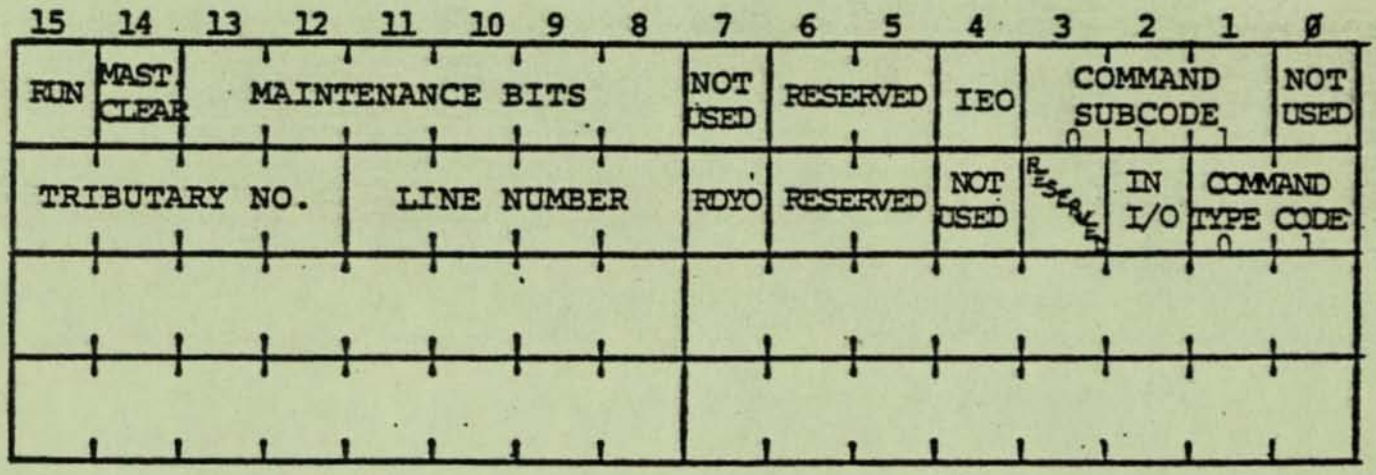
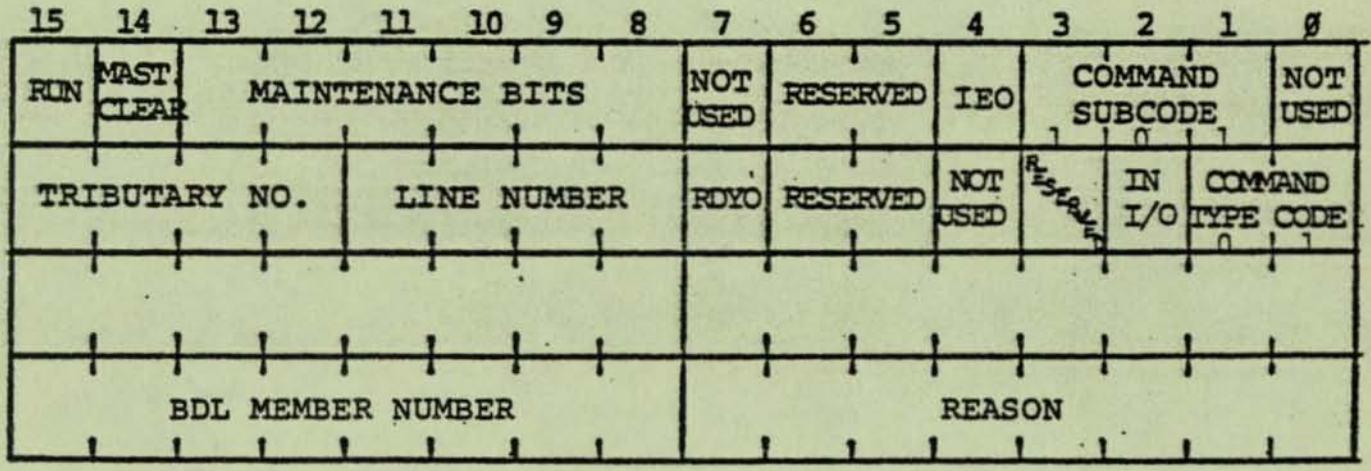


TABLE OF CONTENTS

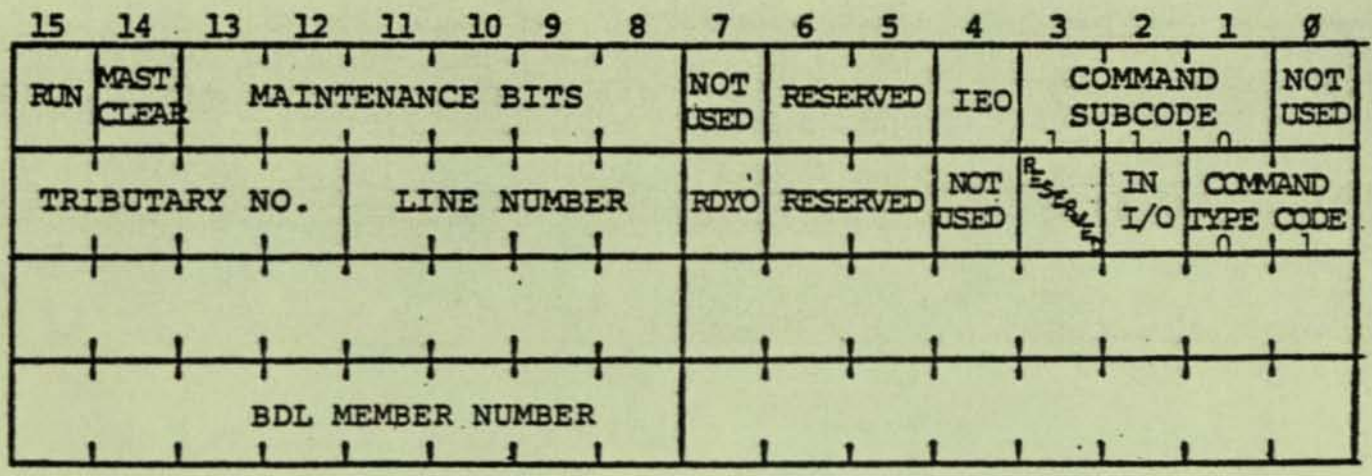
KMC CONTROL OUT COMMANDS

MESSAGE NACKED



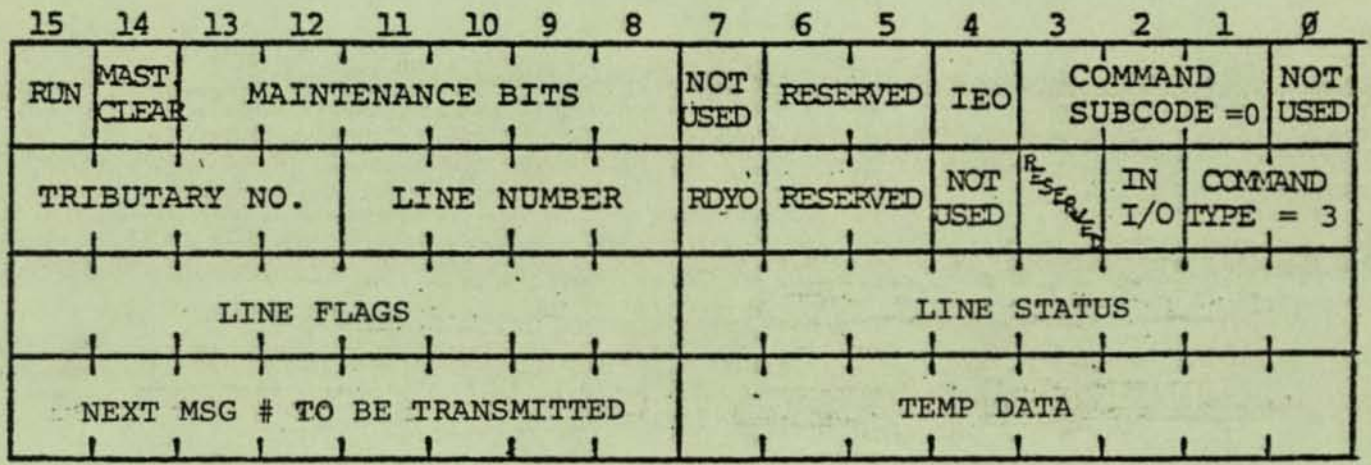
KMC CONTROL OUT COMMANDS

TO MANY RECEIVE BUFFERS

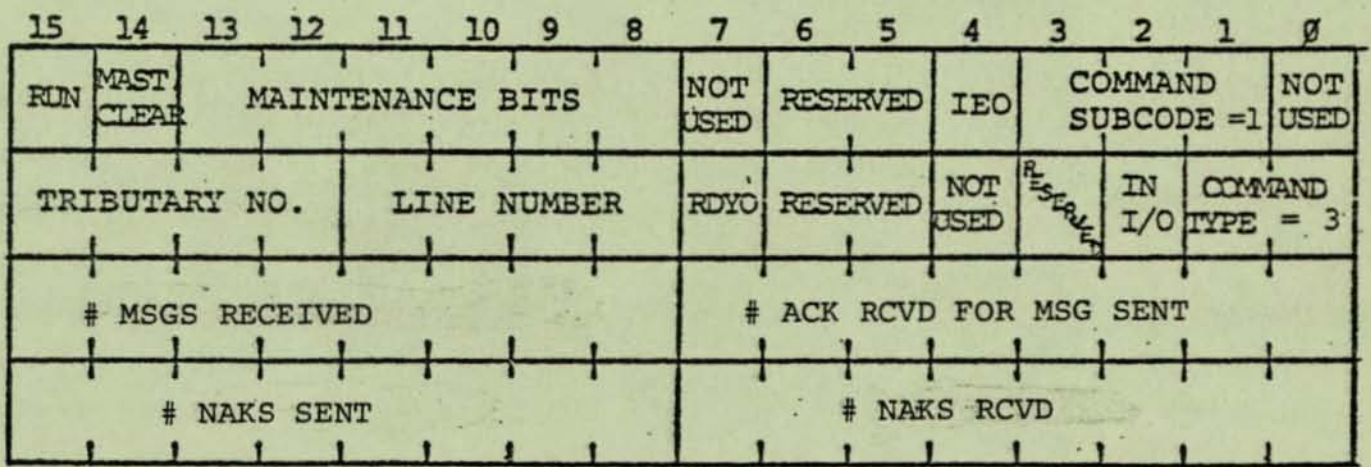


KMC CONTROL OUT COMMANDS

STATUS RESPONSES FORMAT 1

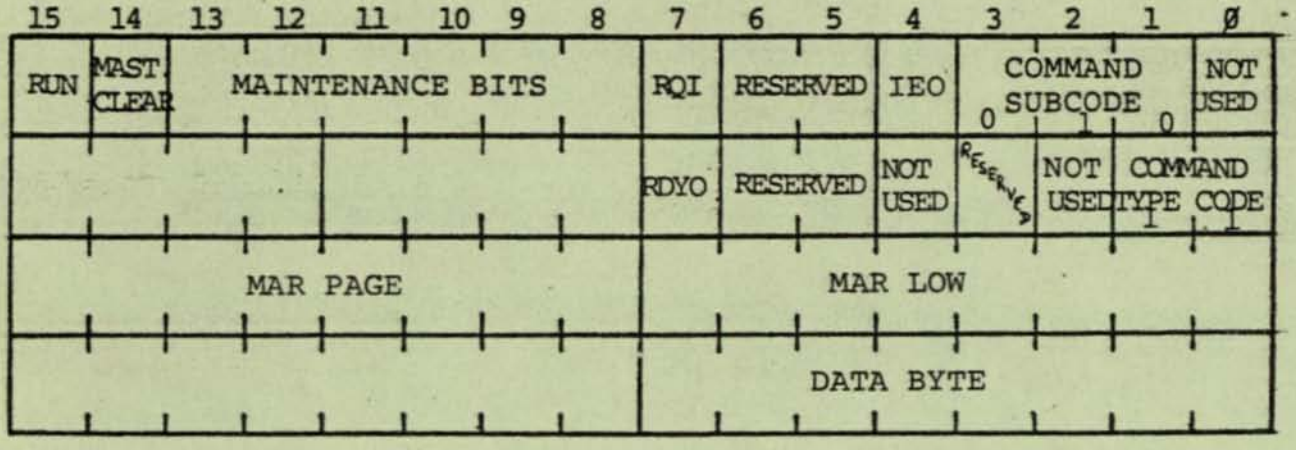


STATUS RESPONSES FORMAT 2



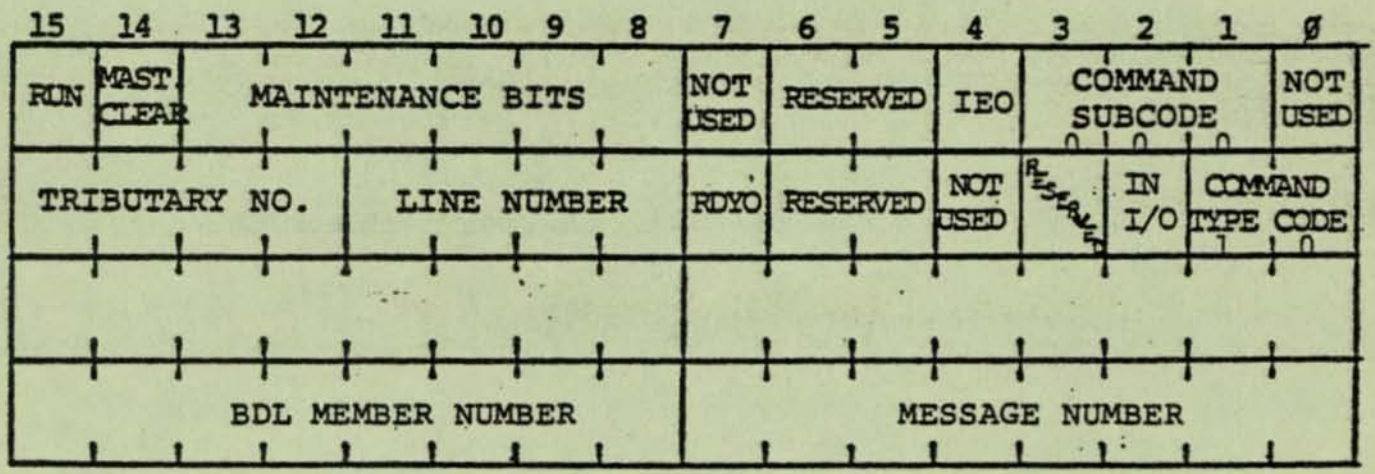
KMC CONTROL OUTPUT COMMANDS

READ KMC11 DATA MEMORY



KMC CONTROL OUT COMMANDS

MESSAGE RECEIVED



4.0 KMC DESIGN

4.1 The KMC DDCMP multiplexer is a state driven system, based on a series of queues, tables and buffers. Each state per line is less than or equal to 260 KMC instructions. The following sections describe the KMC memory organization, tables and queues used in the operation and control of the multiplexer.
The secion starts out with a detailed example of a message transmission and steps involved. This is followed by a memory map, and block diagram of the sytem and definition of tables. The last section contains a definition of various states the system may be in.

4.2 DETAILED EXAMPLE OF MESSAGE TO BE TRANSMITTED

1. Application program passes to driver BDL Base address.

2. PDP11 driver passes BDL base to KMC.

3. Application program request a transmission of a message providing PDP11 Driver with BDL member control address of message data and number of messages to be transmitted.

4. PDP11 Driver requests CSR transfer.

5. KMC acknowledges request when input que is available.

6. KMC retries CSR/Retransmit data and places data in input.

7. KMC command interpreter sees input data in queue. It determines from queue data line data is to be transmitted on. If line is busy, it leaves data in que and waits to check it next time around. It then dertermines if a control message is in the control out que. If so, it is prepared for transmission. Else it sets the line to output state 1 and places CSR data into output mesage character address buffer. The command interpreter state is then set back to 0.

8. The KMC then waits until select flag is set. If set, the KMC then builds the message header.

9. The KMC then builds a BDL address and places a read request into the output NPR queue and set BDL flag.

10. The KMC then performs THE NPR states which request the BDL Read.

11. After completion of the read, the message address contained in the BDL is placed in the Output message character address table.

12. An output request is placed in the NPR queue to set the BDL as in unuse by the KMC.

13. The KMC then requests that the first four words 8 character of the message be stored in the intermediate buffer.

14. After eight characters are buffered, the LST is updated, the header of the message is then transmitted.

15. After the header and CRC of the header is transmitted, the first character of the data message is transmitted.

16. After every even character of the message has been transmitted from the intermediate buffer, an NPR INPUT QUEUE request is made to get two more characters from the PDP11 for transmission until entire message has been retrieved.

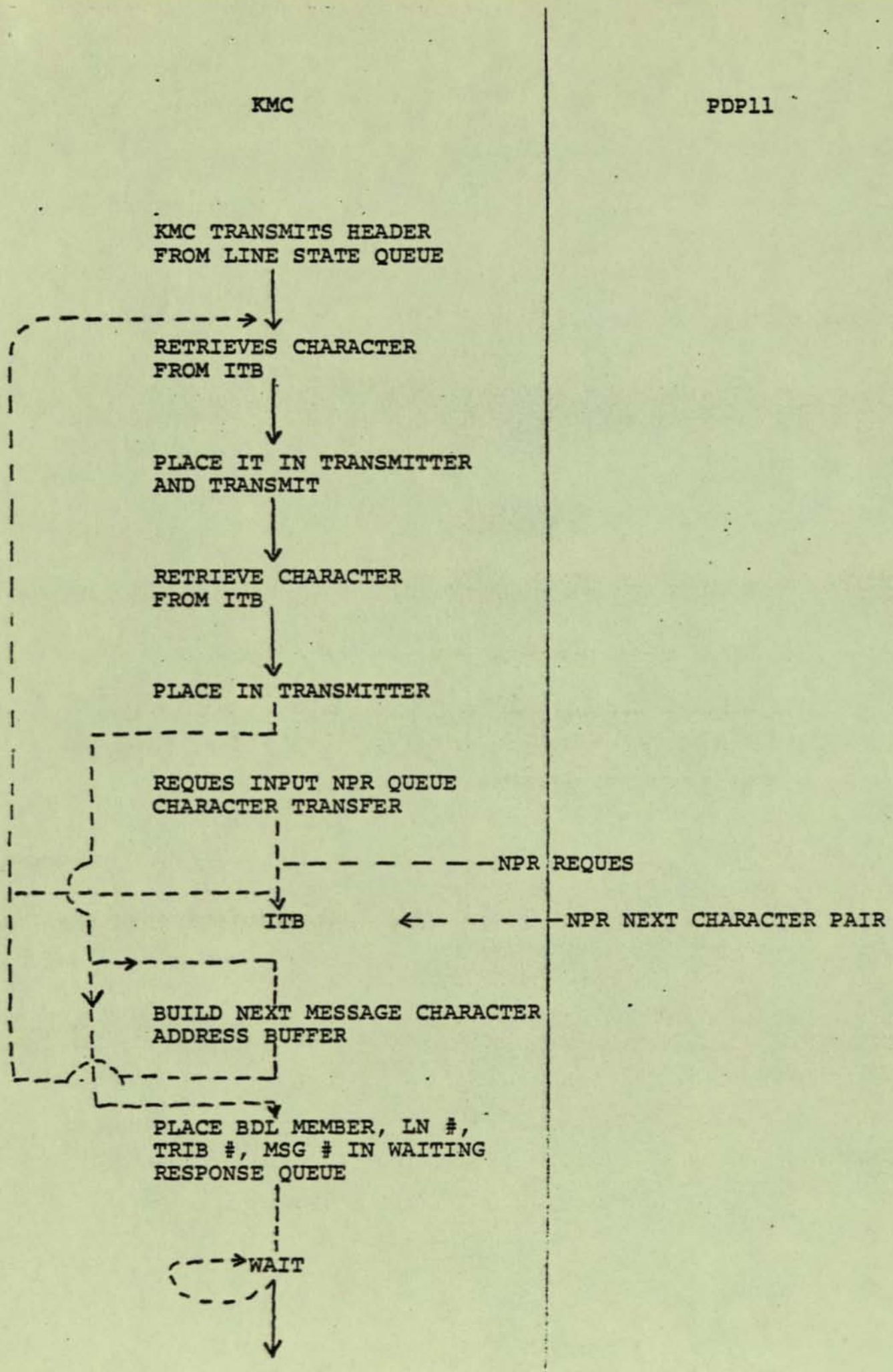
17. At this point in time, if multiple messages (pipelined messages) are to be transmitted, the next BDL address is calculated the BDL and next message flags are set and the address of the next message is retrieved. An NPR INPUT queue request is then made for a charcter pair of the next message while the current message is finishing its transmision.

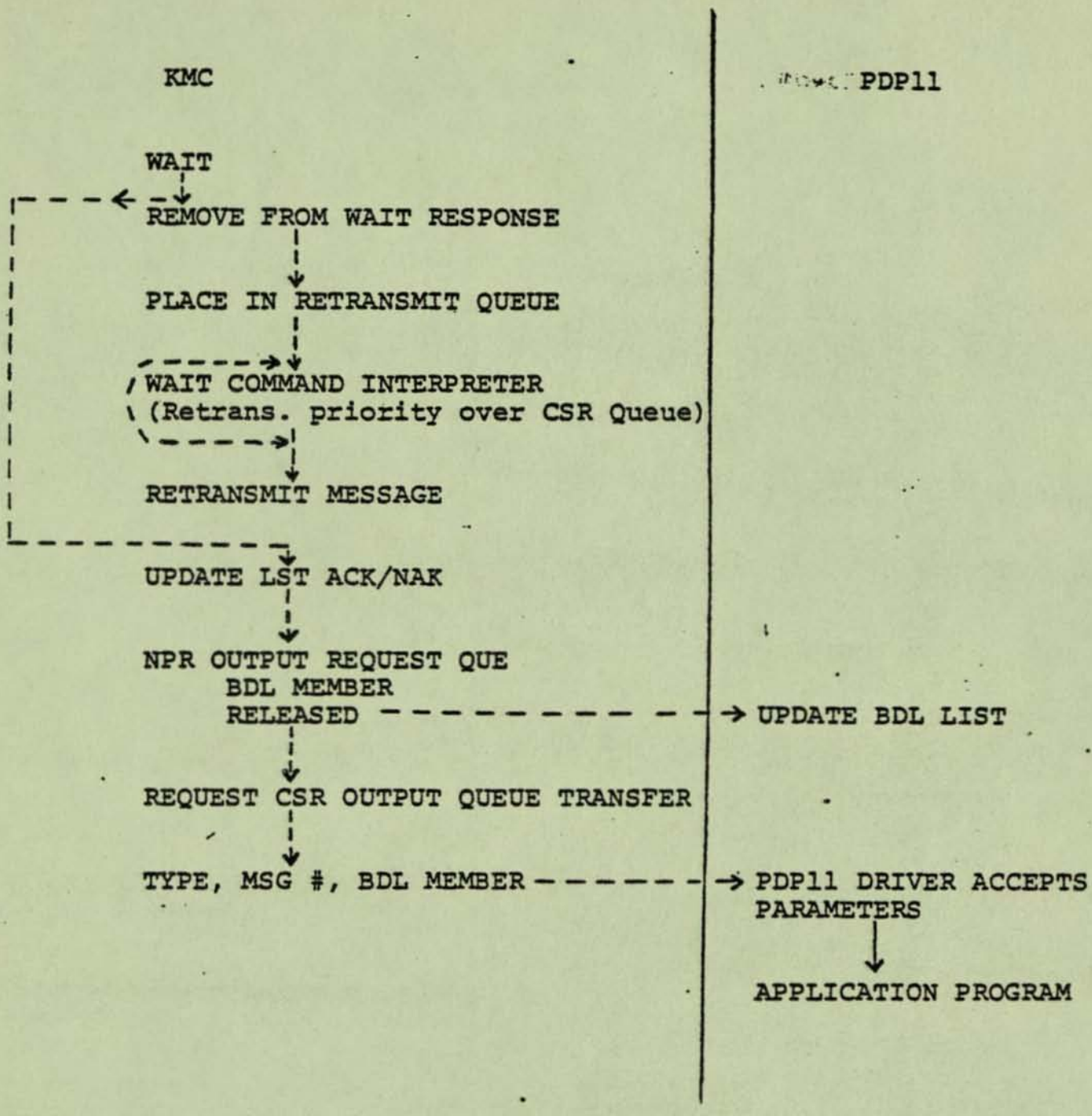
18. Upon completion of the transmission of the message, the message pending flag is set and the BDL member #, line # and tributary # are placed in the message waiting response que.

19. If the message is acked, then the entry in the message pending que is cleared. A request is placed in the OUT NPR Queue to release the BDL. A transmit complete CSR output request is placed in the CSR output que.

20. If the message is nacked, the entry in the message pending que is removed and placed in the retransmit que. An error counter is then incremented.

21. Before another CSR request is honored from the 11, the retransmit que is emptied and goes back for retransmission.

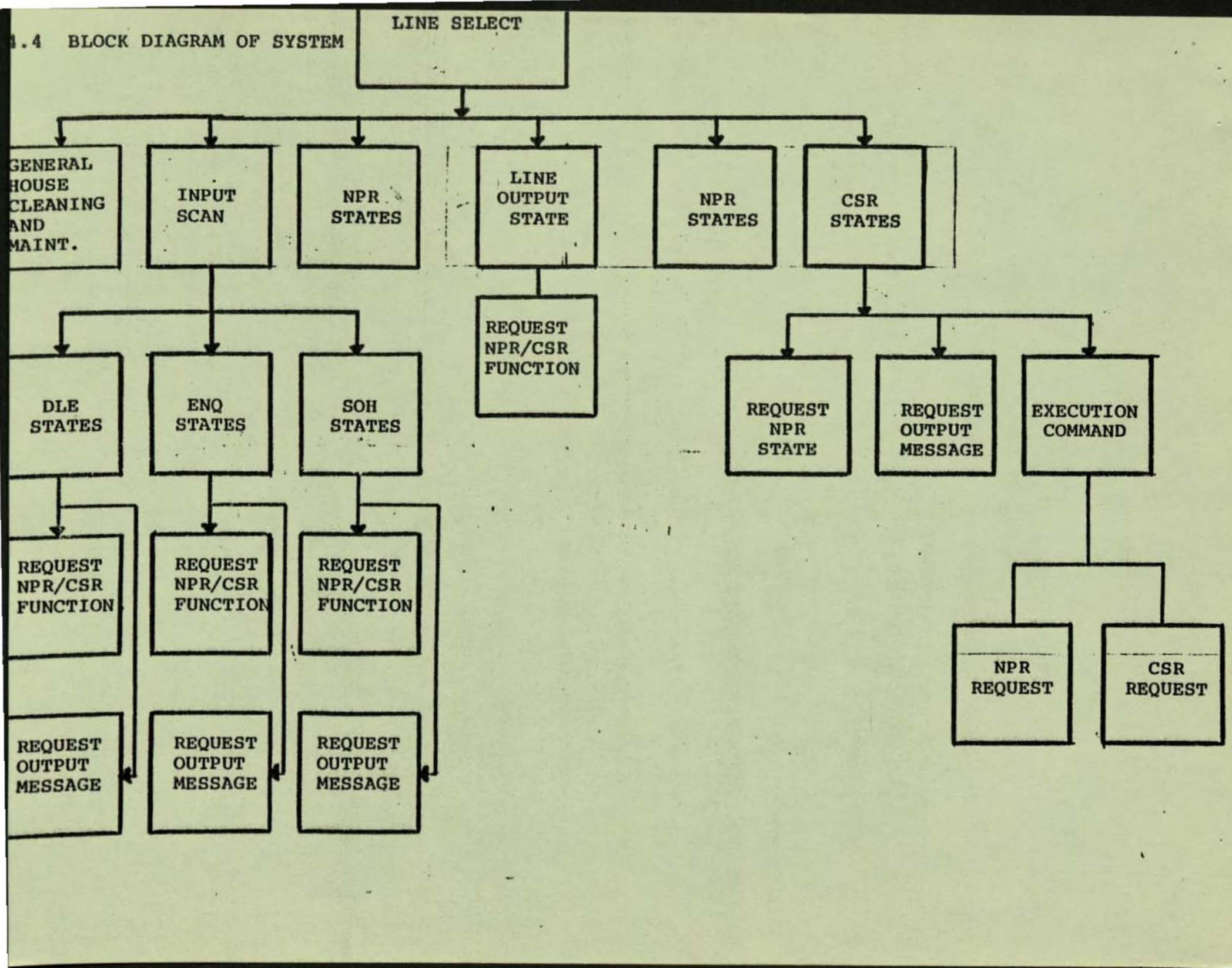

KMC
PDP11
KMC TRANSMITS HEADER FROM LINE STATE QUEUE
RETRIEVES CHARACTER FROM ITB
PLACE IT IN TRANSMITTER AND TRANSMIT
RETRIEVE CHARACTER FROM ITB
PLACE IN TRANSMITTER
REQUES INPUT NPR QUEUE CHARACTER TRANSFER
NPR REQUES
ITB
NPR NEXT CHARACTER PAIR
BUILD NEXT MESSAGE CHARACTER ADDRESS BUFFER
PLACE BDL MEMBER, LN #, TRIB #, MSG # IN WAITING RESPONSE QUEUE
WAIT


KMC
PDP11
WAIT
REMOVE FROM WAIT RESPONSE
PLACE IN RETRANSMIT QUEUE
WAIT COMMAND INTERPRETER
(Retrans. priority over CSR Queue)
RETRANSMIT MESSAGE
UPDATE LST ACK/NAK
NPR OUTPUT REQUEST QUE
BDL MEMBER
RELEASED
UPDATE BDL LIST
REQUEST CSR OUTPUT QUEUE TRANSFER
TYPE, MSG #, BDL MEMBER
PDP11 DRIVER ACCEPTS
PARAMETERS
APPLICATION PROGRAM

4.3 MEMORY DEFINITION

0	- 777	LINE STATUS TABLE (LST)
1000	- 1077	LINE OUTPUT STATE TABLE (LOST)
1100	- 1177	LINE INPUT STATE TABLE (LIST)
1200	- 1277	INTERMEDIATE TRANSMIT BUFFERS(ITB)
1300	- 1466	SPARE
1467	- 1477	CSR INPUT BUFFER
1500	- 1677	CONTROL MESSAGE OUTPUT QUE
1700	- 2077	CONTROL MESSAGE INPUT QUE
2100	- 2207	PRIMARY OUTPUT MSG ADDR BUFFER
2210	- 2317	SECONDARY OUTPUT MSG ADDR BUFFER
2320	- 2427	PRIMARY INPUT MSG ADDRESS BUFFER
2430	- 2547	SECONDARY INPUT MSG ADDRESS BUFFER
2550	- 2777	SPARE
3000	- 3377	NPR OUTPUT REQUEST QUE
3400	- 3777	MSG WAITING RESPONSE QUEUE 1ST HALF
4000	- 4377	NPR INPUT REQUEST QUEUE
4400	- 4777	CSR OUTPUT QUEUE
5000	- 5110	TIMERS
5111	- 6977	SPARE
6400	- 6777	MESSAGES WAITING RESPONSE QUEUE 2ND HALF
7000	- 7640	RETRANSMIT QUE
7640	- 7661	KMC DATA VARIABLES

1.4 BLOCK DIAGRAM OF SYSTEM



MEMORY LAYOUT

LINE STATUS TABLE

	MEM ADDR TRIB # LINE #			
LINE 0	0	0	0	1 2 3 4 5 6 7
TRIB 1	0	1	0	
TRIB 2	0	2	0	
TRIB 3	0	3	0	
TRIB 4	0	4	0	
TRIB 5	0	5	0	
TRIB 6	0	6	0	
TRIB 7	0	7	0	
LINE 1	1	0	0	1 2 3 4 5 6 7
TRIB 1	1	1	0	
TRIB 2	1	2	0	
TRIB 3	1	3	0	
TRIB 4	1	4	0	
TRIB 5	1	5	0	
TRIB 6	1	6	0	
TRIB 7	1	7	0	
	.			
	.			
	.			
	.			
LINE 7	7	0	0	1 2 3 4 5 6 7
TRIB 1	7	1	0	
TRIB 2	7	2	0	
TRIB 3	7	3	0	
TRIB 4	7	4	0	
TRIB 5	7	5	0	
TRIB 6	7	6	0	
TRIB 7	7	7	0	

LINE TRIBUTARY ENTRIES

0 = LINE/TRIB STATUS
1 = LINE/TRIB FLAGS
2 = FLAGS
3 = NEXT MESSAGE TO BE TRANSMITTED
4 = NUMBER OF ACKS RECEIVED
5 = NUMBER OF MESSAGES RECEIVED
6 = NUMBER OF NAKS TRANSMITTED
7 = NUMBER OF NAKS RECEIVED

LINE STATUS BYTE MEANINGS

BYTE	NAME	BIT #	DEFINITION	
0	LINE STATUS	BIT 0	0 = LINE DOWN	1 = LINE UP
		BIT 1	0 = DDCMP MODE	1 = MOP
		BIT 2	0 = XMIT NOT ACTIVE	1 = ACTIVE
		BIT 3	0 = RECEPTION NOT ACTIVE	1 = ACTIVE
		BIT 4	0 = NO START PENDING	1 = START PENDING
		BIT 5	0 = SPARE	1 = SPARE
		BIT 6	0 = FULL DUPLEX	1 = 1/2 DUPLEX
		BIT 7	SELECT FLAG FOR 1/2 DUPLEX TURN AROUND 0 = NO XMIT	
1	LINE FLAGS 1	BIT 0	0 = NO START RCVD	1 = START RCVD
		BIT 1	0 = NO NAK RCVD	1 = NACK RCVD
		BIT 2	0 = NO REP RCVD	1 = REP RCVD
		BIT 3	0 = NO XMIT CONTRO MSG	1 = XMIT CONTROL MSG
		BIT 4	0 = NO ACK RCVD	1 = ACK RCVD
		BIT 5	0 = NO STACK RCVD	1 = STACK RCVD
		BIT 6		
		BIT 7	0 = NO START PENDING	1 = START PENDING
2	TEMPORARY DATA		MSG # FROM ACK OR NAK	
3	NBRNXT		NEXT MESSAGE NUMBER TO BE TRANSMITTED	
4	NBRACK		NUMBER OF ACKs RECEIVED FOR MESSAGE SENT	
5	NBRRCV		NUMBER OF MESSAGES RECEIVED	
6	NAKRCV		NUMBER OF NAKs RECEIVED (NO OVERFLOW)	
7	NAKXMT		NUMBER OF NAKs TRANSMITTED (NO OVERFLOW)	

4.5 TABLES AND QUES AND BUFFERS

The following sections provide definitions of the Tables, Queues and Buffers used internally by the KMC11.

4.5.1 LINE STATUS TABLE (LST)

The line status table is used to reflect the current status and definition for each line. The table is configured of 64 entries at 8 bytes of information per entry. The entries consist of 8 lines with 8 tributaries per line. The line table entries can be thought of as a series of row entries. The line number and line tributary number make up the memory address of where the table entry for a given line is to be found. The line status table contains the following information:

* Current line status up/down
* Current mode DDCMP/MOP
* Transmitter state
* Receive state
* Full or 1/2 duplex line
* Select flag
* Next message number to be sent
* Number of ACKs received for message sent
* Number of messages sent
* Number of NAKs transmitted
* Number of NAKs received

4.5.2 LINE OUTPUT STATE TABLE (L.O.S.T.)

The Line Output State Table consists of eight subtables. One for each output line. Each subtable contains eight entries which contain the current state of the line and the header of the current output message except CRC values. The first byte of each subtable is the current state of the line. This state value is used as an indexed branch to execute the =next series of instructions to keep the line operating properly. The remainder of the table contains the header data of the current message being transmitted.

LINE OUTPUT STATE TABLE

	LINE TRIB						
		STATE	BASE BDL	BASE LOW	BASE UP	WORK BDL	BDL	# MSGS
ADDR	1000	1	2	3	4	5	6	7
	1010	STATE	HDR1	HDR2	HDR3	HDR4	CRC1/NEXT CHAR IN	CRC2/NEXT CHAR AVAIL.
	.							
	.							
	.							
	1077	"	"	"	"	"	"	"

LINE OUTPUT STATE TABLE CONTROL MSG

1	2	3	4	5	6	7
STATE	ACK #	LSTL	LSTH	INDEX	NAK COUNT	

4.5.3 LINE INPUT STATE TABLE (L.I.S.T.)

The Line Input State Table is configured exactly as the Line Output State Table except that it is used for input rather than output.

LINE INPUT STATE TABLE

STATES							
ADDR	1100	1	2	3	4	5	6	7
		STATE	HDR1	HDR2	HDR3	HDR4	CRC1/NEXT CHAR IN	CRC2/NEXT CHAR AVAIL.
	.							
	.							
	.							
	1170	"	"	"	"	"	"	"

4.5.4 CSR QUEUE

The CSR functions have been broken down into Input and Output States and Ques. There can be only one CSR state at any given time. If there is both an input request and an output request, the KMC output request takes priority. There are eleven (11) output queue entries and one input que entry. See Section 3.2.3.3 on control in and control out commands for specific CSR control information.

CSR OUTPUT QUE ENTRIES

BYTE 0 = SUBCOMMAND
1 = LINE & TRIB #
2 = DATA BSEL 3
3 = DATA BSEL 4
4 = DATA BSEL 5
5 = DATA BSEL 6
6 = COMMAND TYPE

CSR INPUT QUEUE

BYTE 0 = COMMAND (BITS 0,1)
1 = SUBCOMMAND (BITS 2,3,4)
2 = LINE # (BITS 0-3) & TRIB # (BITS 4-7)
3 = BDL ADDRESS MIDDLE BYTE/DATA
4 = BDL ADDRESS LOWER BYTE/DATA
5 = BDL ADDRESS UPPER 2 BITS (6,7)/MEMBER NO. BITS 0-7
6 = # OF MESSAGES TO BE SENT.

4.5.5 NPR INPUT REQUEST

The NPR input request queue is the only interface that allows NPR input transfers between the KMC and the PDP11. There are two basic types of input NPR transfers in the system: message data and B.D.L. data. To request a BDL transfer, the line number is entered into the NPR Que with BDL bit set (bit 4) and with either the primary or secondary message character buffer bit set (bit 5). Bits 0-3 indicate the line number that is requesting the transfer and point to the message character buffer used. Bit 4 if on requests that the contents be placed not in the intermediate buffer for the line, but in the message character address buffer. Bit 5 indicates whether the BDL data is to be in the primary or secondary message character address buffer. For a simple message character input only, the line number is entered and the retrieved data characters are placed in the intermediate buffer for the requesting line.

4.5.6 NPR OUTPUT REQUEST QUEUE

The NPR output request que operates similar to that of the input request. The queue is structured rather differently; there are three bytes per entry verses the one (1) in the Input Request Que. The first byte the line number has the same definition as that of an Input Request. The second two bytes are the character pair or data that is to be stored.

NPR CHARACTER INPUT REQUEST QUEUE

LINE #	LINE #	LN#	LN#	LINE #
ENTRY 1	ENTRY 2	ENT.3	ENT. 15	ENTRY 16

NPR CHARACTER OUTPUT REQUEST QUEUE

	CHARACTER A	CHARACTER B	ADDR
ENTRY 1	"	"	"
ENTRY 2	"	"	"
ENTRY 3			
ENTRY 16	CHARACTER A	CHARACTER B	ADDR

4.5.7 INPUT/OUTPUT MESSAGE CHARACTER ADDRESS BUFFERS

The Output Message Character Address Buffers are the main interface to the NPR queue. These buffers contain the address of where the next two characters for transfer from the PDP11 are stored. It also contains the character count of the message and the count of the number of characters retrieved from the PDP11. The tributary number and current message transmit number are also stored in this buffer. An example of its use is after a character pair is transmitted, the line number is placed in the input NPR queue. The input NPR queue accesses the output message character address buffer to find the address of the next character pair to be transmitted.

There are four message character address buffers - two (2) for input a`nd two (2) for output. Th`=e reason for redundancy is to save time during pipelined input or output messages. The "next" Input/Output Message Character Address buffer is readied during use of the primary buffer. As soon as the primary buffer is complete, a buffer flip flop takes place and the secondary is now the primary buffer interface for transmitting or receiving characters.

OUTPUT DATA CHARACTER ADDRESS BUFFER

BYTE	0	1	2	3	4	5	6	7	8
	TRIB 3	BDL ADDR		CHAR COUNT		# OF CHAR TRANSFERRED		BDL #	# OF MSGS
LINE 1									
LINE 2									
LINE 3									
.									
LINE 8									

INPUT DATA CHARACTER ADDRESS BUFFER

BYTE	0	1	2	3	4	5	6	7	8
	BDL #	BDL ADDR			CHAR COUNT		# OF CHAR RECEIVEDS		ODD BYTE RECEIVED
LINE 1									
LINE 2									
LINE 3									
.									
LINE 8									

WAITING RESPONSE BUFFER

	LINE #	TRIB #	MSG #	BDL MEMBER
ENTRY 1				
ENTRY 2				
ENTRY 3				
.				
.				
.				
ENTRY 100	LINE #	TRIB #	MSG #	BDL MEMBER

4.5.8 MESSAGE WAITING RESPONSE QUEUE

After a message has been transmitted, its line number, tributary number, message number and BDL number are entered into the waitin response queue. This queue is a holding area containing all pertinent information concerning the message until the message disposition can be determined. If the message is acked, the BDL member is released, the queue entry cleared, and the PDP11 is notified of the transmit. If the message was naked and seven retries have not occurred, the queue entry will be cleared and th data will be entered into the retransmit que.

4.5.9 RETRANSMIT QUE

This queue contains information required to retransmit a message that was NAKED or failed to get an acknowledgement. This queue i examined and takes priority over the CSR input request que from the PDP11.

4.5.10 INTERMEDIATE TRANSMIT BUFFER

There is one intermediate buffer per line in the system. This buffer area provides an input buffering function between characters being sent from the PDP11. The ITB keeps eight (8) characters buffered ahead of the transmitter such that if a transfer of a character is late in coming from the PDP11, the KMC will not have to wait and will transmit the seven (7) previously buffered characters before the one that arrived late. The ITB is used in conjunction with the NPR INPUT request que.

INTERMEDIATE TRANSMIT BUFFER

ADDR	120	0	1	2	3	4	5	6	7
	121	DATA	DATA	DATA	DATA	DATA	DATA	DATA	DATA
	122		"	"	"	"	"	"	"
	.								
	.								
	.								
	.								
	127		"	"	"	"	"	"	"

INTERMEDIATE RECEIVE BUFFER

ADDR	130	0	1	2	3	4	5	6	7
	131	DATA	DATA	DATA	DATA	DATA	DATA	DATA	DATA
	132								
	.								
	.								
	.								
	.								
	137								

4.5.11 CONTROL MESSAGE QUEUE

The control message queues are used to hold control messages that are to be transmitted as soon as the line is free. There are two (2) queues - one for input message reception and the second is for message transmission. Position in the queue defines line and tributary, thus entry nine in the que is for line one, tributary one. To determine the line and tributary number, the queue entry is divided by eight. The integer from the division gives the line number and the remainder is the tributary number. Each queue entry is two bytes long. The first byte defines the message type and the second the subtype.

LINE 0 Type	LINE 0 Q,S Bits SubType	LINE 1 Type	LINE 1 SubType	LINE 2 Type	LINE 3 SubType	LINE 4 Type
LINE 4 SubType	LINE 5 Type	LINE 5 SubType	LINE 6 Type	LINE 6 SubType	LINE 7 Type	LINE 7 SubType

4.6 SYSTEM STATES

The following sections contain a list of the various states the multiplexer may be in. It should be noted that it is possible and very likely that the KMC will be in several different states at any given point in time.