EK-DTE20-UD-003

DTE20 TEN-ELEVEN INTERFACE UNIT DESCRIPTION

digital equipment corporation • marlborough, massachusetts

1st Edition, July 1975
2nd Edition (Rev), January 1976
3rd Edition (Rev), October 1976

The material in this manual is for informational purposes and is subject to change without notice. Digital Equipment Corporation assumes no responsibility for any errors which may appear in this manual.

Printed in U.S.A.

This document was set on DIGITAL's DECset-8000 computerized typesetting system.

CONTENTS

CONTENTS (Cont)

ILLUSTRATIONS

ILLUSTRATIONS (Cont)

TABLES

TABLES (Cont)

PREFACE

This manual contains three levels of DTE theory descriptions. The three levels are:

1. Overview – The overview introduces and identifies, in a simplified fashion, the basic hardware organization of the DTE Console Processor Interface. The major elements are presented without extensive details in order to provide a capsule view of the DTE structure.

2. Functional Description – This section describes the primary DTE function, which is to interface "front end" PDP-11 processors to the KL10 Central Processor. In such a system, several front end functions are thus provided, some of which are:

   a. Handling unit record equipment
   b. Handling communications equipment
   c. Diagnosing the KL10 Central Processor
   d. Bootstrapping the KL10 system.

   In addition to front end functions, the DTE features other capabilities. Some of these capabilities are:

   a. Examine and Deposit console functions

   b. Doorbell function, where the PDP-11 can interrupt the KL10 and vice versa

   c. High speed simultaneous two-way variable byte data transfer between the PDP-11 and KL10 memory.

   The functional description is the most comprehensive part of the DTE theory. Here, the basic elements of the DTE are described in the context of how they implement the primary DTE operations.

3. Logic Description – This section provides a detailed logic description of the DTE. The text is written to support the functional description. The logic description section is the most detailed part of the manual. This material is presented to expand the functional description so that the information provided in the functional description may be directly related to the engineering logic diagrams.

SECTION 1
OVERVIEW

1.1 INTRODUCTION

Each central processor in a KL10 system may have from one to four PDP-11 processors attached, each serving as a "front end" processor. Each PDP-11 is connected to the KL10 by a separate interface called the DTE20 Console Processor Interface, or simply the 10-11 Interface. The following are some of the possible front end functions:

1. Handling unit record equipment
2. Handling asynchronous communications equipment
3. Handling synchronous communications equipment
4. Providing a long term power line frequency clock
5. Diagnosing the KL10 Central Processor and other functional components in the system
6. Running a dedicated real-time data acquisition system
7. Bootstrapping the KL10 system.

In terms of basic features, the DTE20 generates parity for Deposit data and detects parity errors for both Examine data and byte transfers over the EBus. The DTE20 connects to the PDP-11 as a standard Unibus peripheral and communicates via interrupt or device address. Up to four DTE20s may be connected to a PDP-11. In a system consisting of four KL10 Central Processors, there may be four PDP-11/40 processors, where each processor can communicate with all KL10s in the system. It is possible to have up to four DTE20s on each PDP-11 in the KL10 system, and each KL10 processor may have 1, 2, 3, or 4 DTE20s connected to it via the EBus.

The DTE20 uses the NPR (Direct Memory Access) and BR (Vector Interrupt) features of the PDP-11. In addition, the DTE20 contains logic to detect PDP-11 core memory parity errors during NPR transfers, provided that the memory being accessed contains the parity option (MFU11 UP).

The DTE20 provides the following capabilities:

1. Console functions at Examine and Deposit, restricted or unrestricted.

2. Doorbell function, where the PDP-11 can interrupt the KL10 Central Processor and vice versa.

3. High speed simultaneous two-way transfer of variable byte data between the PDP-11 and KL10 memory.

4. Diagnostic bus for the PDP-11 to diagnose the KL10.

5. KL10-initiated bootstrap startup of the PDP-11 mechanism (diagnostic bus) to load the microcode into the CRAM, execute PDP-10 instructions, and start or stop the KL10 Central Processor.

The following terminology will give some perspective on the front end and its relationship to the DTE20.

PDP-11 Communication Region – This region consists of an area of KL10 core memory defined by the deposit relocation and protection word in the Executive Process Table (EPT). This area is written by the PDP-11 using protected deposits, and read by the KL10. It is used for coordination of status, preparing for byte transfer operations, and passing limited amounts of data. Each PDP-11 in the system has a separate communication region in the KL10 memory, which it alone can modify.

KL10 Communication Region – This region is defined solely by the KL10 software and is separate from the PDP-11 communication region. It can be written by the KL10, but may be read by the PDP-11 using protected Examines. This area is used to coordinate status, prepare byte transfer operations, and pass limited amounts of data (Figure 1-1).

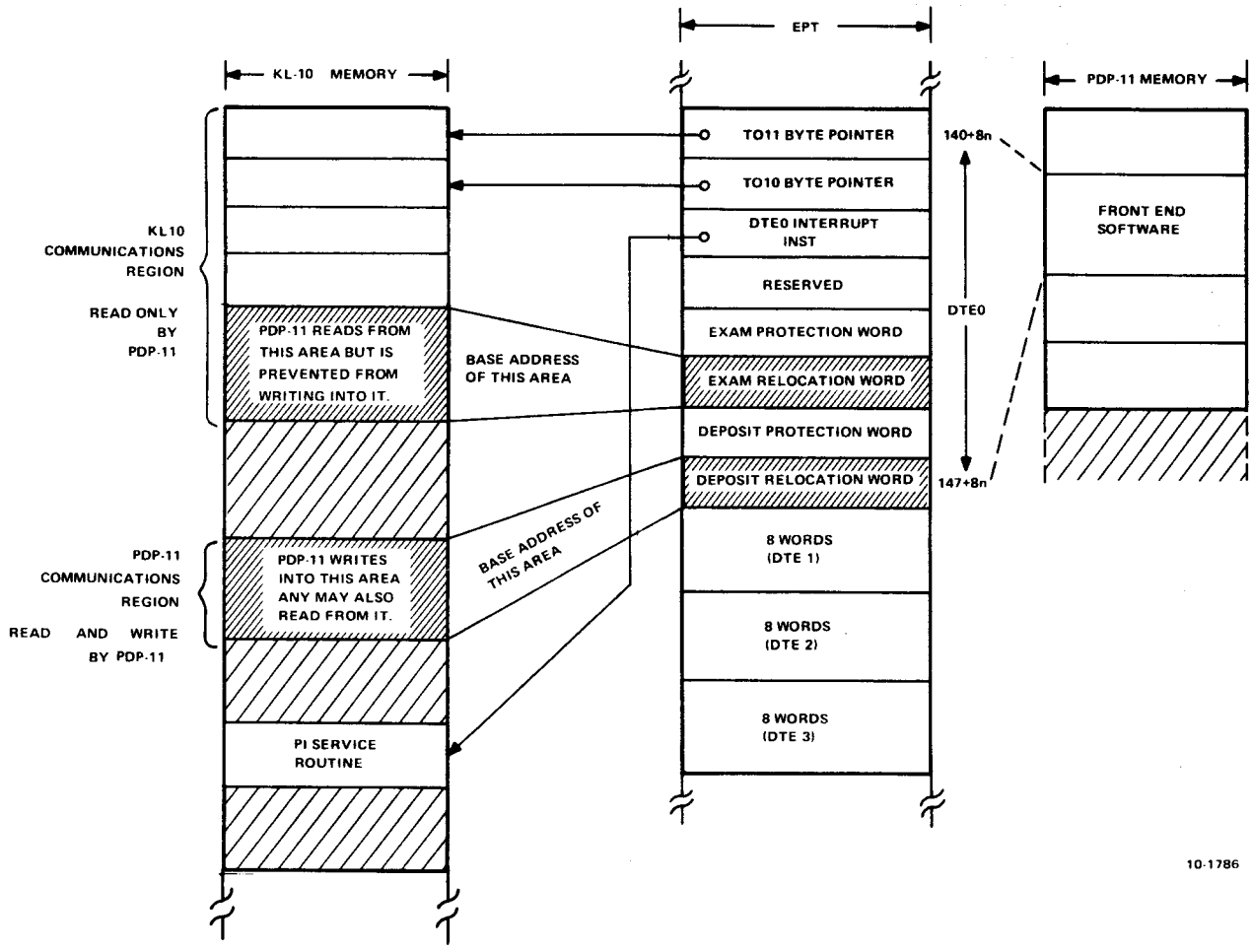


Figure 1-1 Overview Communications Region

Restricted Front End – A restricted front end is a PDP-11 system with a DTE20 that does not have diagnostic privileges. A restricted front end is prevented from using the diagnostic bus. A restricted front end can only access KL10 memory after the KL10 has performed a CONO (Conditions Out) to allow use of DTE PI0. After this has been done, the restricted front end can only examine or transfer bytes from the KL10 communication region and only deposit or transfer bytes to its own PDP-11 communication region.

Privileged Front End – A privileged front end is a PDP-11 attached to a KL10 via a DTE20 that can use the diagnostic bus and perform unrestricted Deposits.

Protected Examine or Deposit – A protected Examine or Deposit is an Examine or Deposit that is relocated and range checked by the KL10. The relocation and protection for Examine is separate from that of Deposit. A privileged front end can override the Examine and Deposit protection checks. A restricted front end cannot override these checks.

For addressing purposes, each controller is permanently assigned a unique device, or Controller Select (CS) code. A total of four Controller Select codes has been assigned because up to four controllers (interfaces) can be implemented in a KL10 system. Each interface is also assigned a physical number according to the physical slots in which the interface module will reside. These are indicated below. Both of these are hard-wired on the KL10 backplane. The specific Controller Select (CS) codes and physical number (n) assignments are as follows:

Interface	Controller Select (CS) Codes	Physical Number n_{10}
0	200	8
1	204	9
2	210	10
3	214	11

The device code is used to address the interface and the physical number is used to identify the interrupting interface.

Eight locations are assigned to each DTE20 in the KL10 Executive Process Table as follows:

Location	Name
140 + 8*n	To 11 Byte Pointer
141 + 8*n	To 10 Byte Pointer
142 + 8*n	DTE20 Interrupt Instruction
143 + 8*n	Reserved for DEC Hardware
144 + 8*n	Examine Protect Word
145 + 8*n	Examine Relocation Word
146 + 8*n	Deposit Protect Word
147 + 8*n	Deposit Relocation Word

NOTE: n = 0, 1, 2 or 3

Figure 1-2, API Word Format, illustrates the basic format of this word. The DTE20 allows the software to set the following fields of this 36-bit word:

Address Space Field 0-2
Unused bits 11-12
Address Field 13-35.

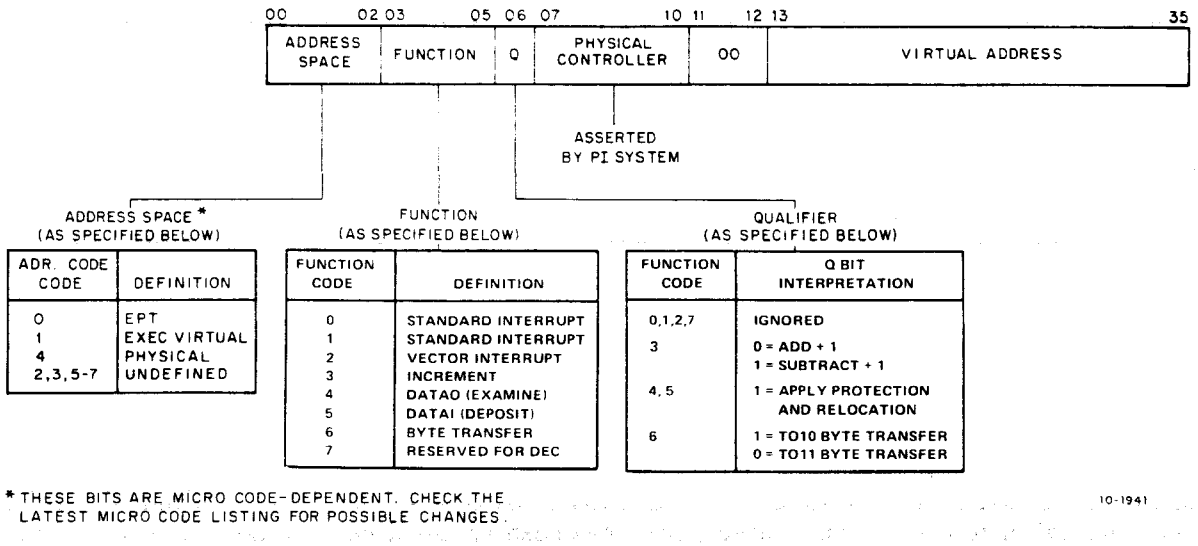


Figure 1-2 API Word Format

The Priority Interrupt (PI) board in the EBox supplies the physical controller number field [7-10]. The DTE20 asserts Qualifier (Q) bit 6, for all Examines and Deposits by a restricted front end, whether protected or not.

The DTE20 asserts Qualifier for all protected Examines and Deposits by a privileged front end and does not assert it if the privileged front end makes an unprotected Examine or Deposit.

1.2 BASIC PROGRAMMING OVERVIEW

To specify a 36-bit PDP-10 data word, three PDP-11 words are used. They are Deposit/Examine Data Word 1, Deposit/Examine Data Word 2, and Deposit/Examine Data Word 3.

To specify a 23-bit PDP-10 address, two PDP-11 words are used. They are Ten Address Word 1 and Ten Address Word 2. The high order part of Ten Address Word 1 is used for control. Ten Address Word 1 specifies whether an Examine or Deposit is to be done. For a privileged front end, the protect off bit in the Ten Address Word 1 can be set by the software to allow an unprotected Examine or Deposit. On unprotected operations, the space field specifies the type of address: Executive Process Table (EPT), Exec Virtual, or Physical Address, which may refer to core memory or ACs.

The Examine or Deposit function is started when the PDP-11 program writes the Ten Address Word. No program interrupts are generated on the KL10 or the PDP-11 side to signal completion of the Examine or Deposit. Therefore, the PDP-11 program must check for completion by looking at the status DEXDONE bit. The DTE20 clears DEXDONE when the PDP-11 writes Ten Address Word 2, so the software never needs to. Data in TENAD1, TENAD2, DEXWD1, DEXWD2, DEXWD3 remain intact after an operation. Therefore, the PDP-11 may perform repeated protected Examines or Deposits merely by writing the TENAD2 word each time. An Examine followed by a Deposit (changing only TENAD1 and TENAD2) will result in moving data from one KL10 core location to another. For unprotected operations, the PDP-11 must reload the protect off bit (PRTOFF) between each operation (Figures 1-3 and 1-4).

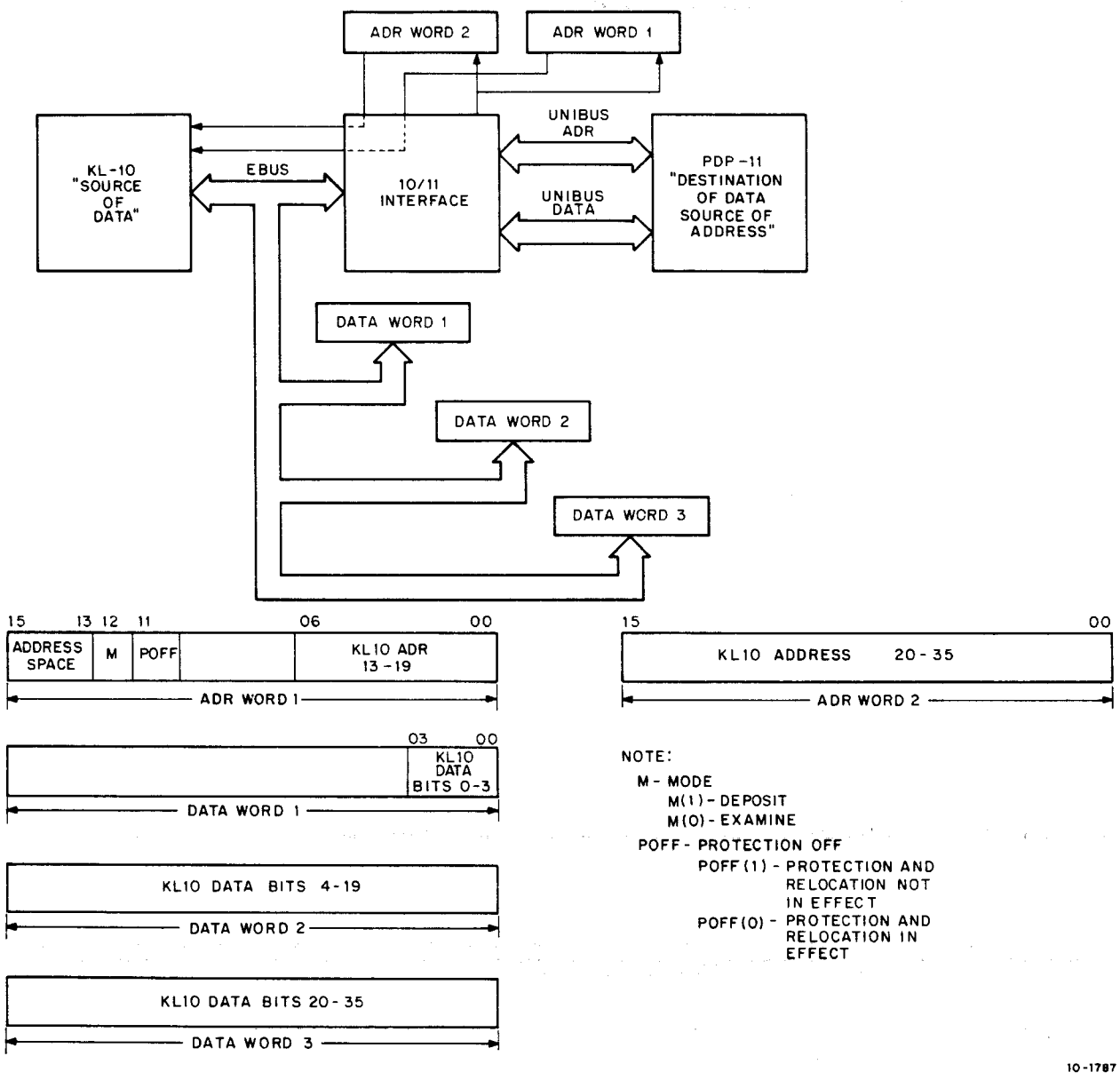


Figure 1-3 Examine Overview

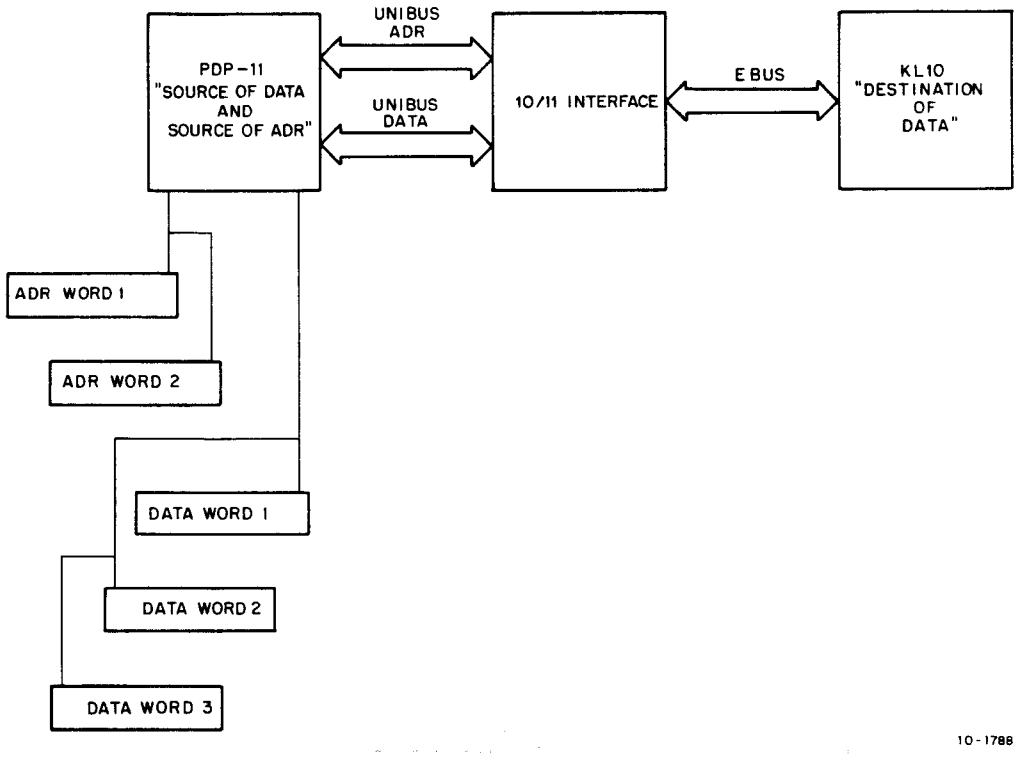


Figure 1-4 Deposit Overview

1.3 DOORBELL FUNCTION

The doorbell function allows each KL10 to interrupt each PDP-11 connected by a DTE20 and vice versa.

The doorbell consists of a programmable interrupt and a status bit. In order for the PDP-11 to interrupt the KL10, the PDP-11 sets the request 10 interrupt flip-flop (bit 08) in the PDP-11 status word. When this bit is set, the DTE20 generates an interrupt in the KL10 with a status bit set in the CONI word (bit 26) indicating that the PDP-11 CPU has programmed an interrupt of the KL10 (Figure 1-5).

This procedure works in a reversed but identical manner for the KL10 interrupting the PDP-11. The KL10 sets the 10 requesting 11 interrupt by doing a CONO to the DTE20. The PDP-11 discovers the cause for the interrupt by looking at bit TO10DB (bit 11) in status. Communication is done via a word (or words) in the communication region in KL10 memory. A word (or words) is chosen and Deposit and Examine features are used by the PDP-11 to gain access to these words (Figure 1-6).

This mechanism is used by either processor to indicate to the other processor that it is powering down. For example, if the KL10 determines that its power is disappearing, it will set a bit in a word that is assigned for power failure notification. The KL10 then interrupts the PDP-11. The PDP-11, as part of its standard routine, will always check for the KL10 power fail bit in the communication region. In this way, the PDP-11 is notified that the KL10 power is disappearing. In a similar way the PDP-11 could interrupt the KL10 on every tick of the power line clock (50 or 60 Hz).

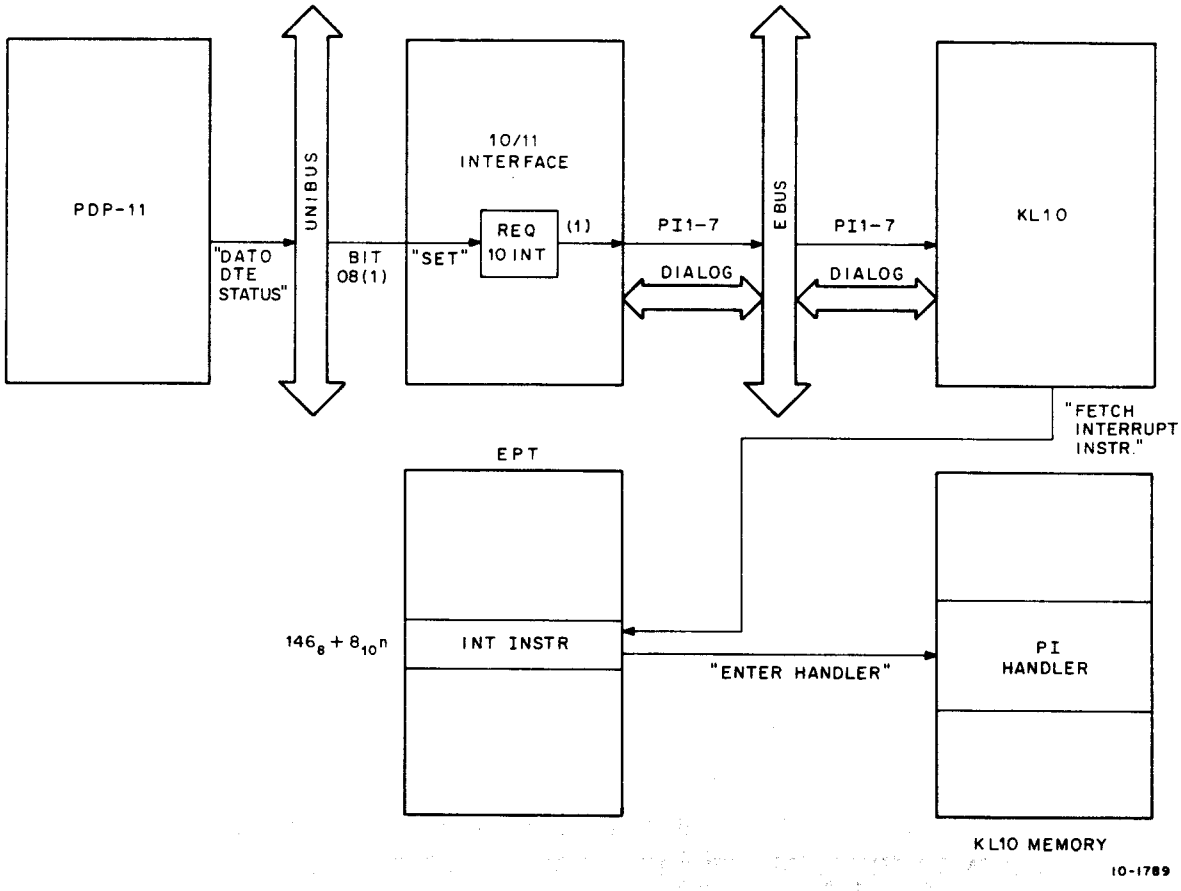


Figure 1-5 PDP-11 Rings KL10 Doorbell

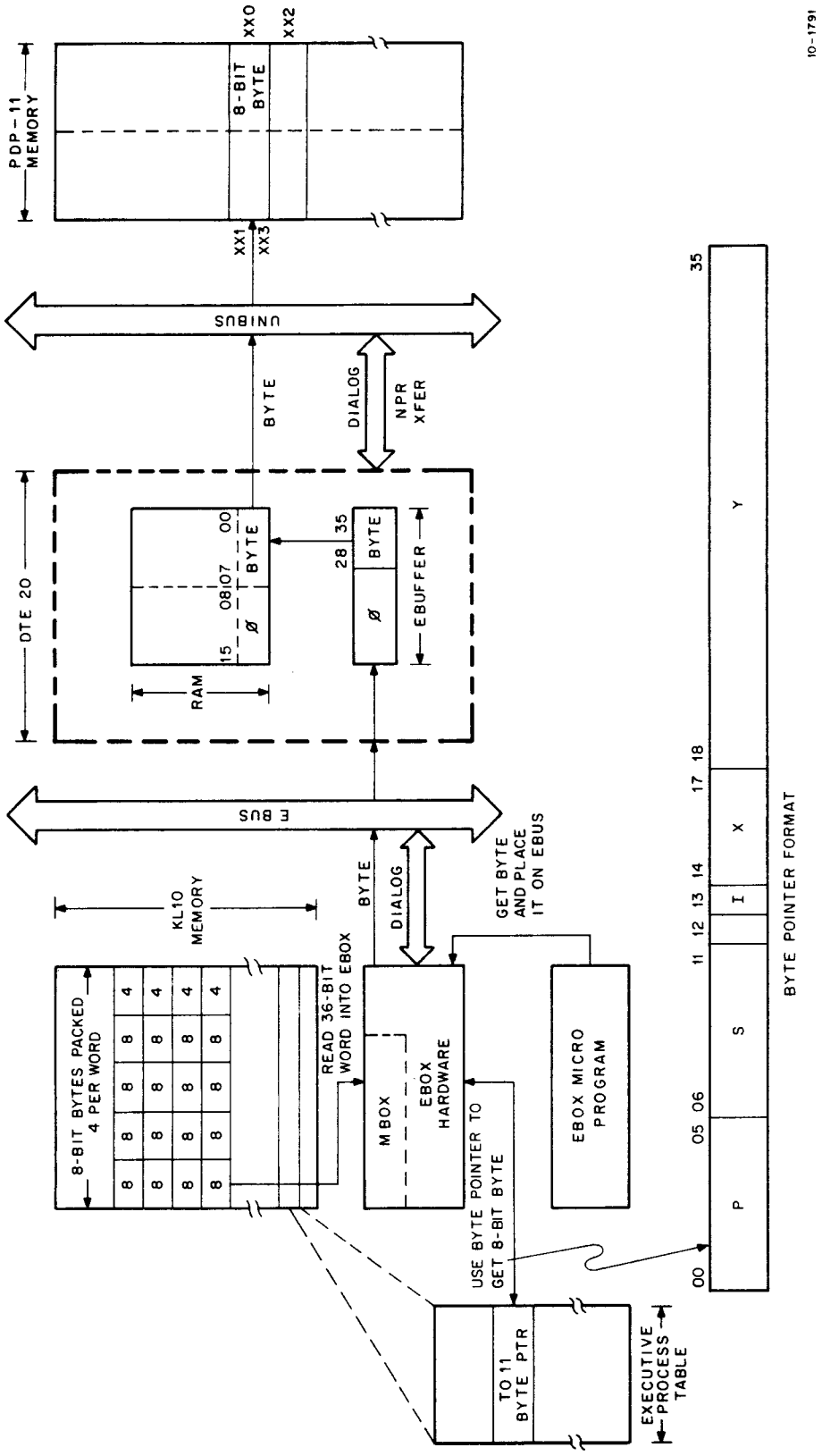


Figure 1-6 TO11 Byte Transfer Overview

1.4 BYTE TRANSFER FUNCTION

During the byte transfer function, the DTE20 transfers fields of information between the PDP-11 and the EBox. On the KL10 side, the fields are of variable length and are accessed through a PDP-10 byte pointer. On the PDP-11 side, the fields are either 8 bits wide and are stored in consecutive bytes or are 16 bits wide and are stored in consecutive words. If the field into which the information is being stored is narrower than the field from which it was read, as many of the rightmost bits as will fit are stored. If the field into which the information is being stored is wider than the field from which it was read, the information is right-aligned and padded with zeroes on the left.

To perform a transfer, the following actions must be done:

- The PDP-11 should specify the transfer rate (delay between transfers) and address bits 17-16 (this can be done once at system startup). If it is not specified, an undetermined transfer rate will occur to one of the four 32K memory regions.
- The PDP-11 must specify whether byte or word mode is to be used in the PDP-11.
- The sender must specify the address of the source string. The KL10 controls the address of the data either to or from the KL10 via byte pointers in the EPT. The PDP-11 controls the address on its side via two locations in the DTE (one word for each direction of transfer).
- The receiver must specify whether it alone (scatter write) or both CPUs are to receive normal termination interrupts (I bit = 1).

Information in the form of bytes may be stored in the PDP-11 as either one variable sized byte per PDP-11 16-bit word (1 to 16 bits of data) or one variable sized byte per 8-bit PDP-11 byte (1 to 8 bits of data). Byte addresses are specified in the KL10 using regular KL10 byte pointers in the EPT. Byte pointers are interpreted in Exec Virtual Address space.

CAUTION

The index field of the byte pointers should be zero. Otherwise, the EBox will index using the current contents of the Executive or User Index register at the time of the transfer. Indirection should not be used because the indirect word will not be incremented as with all byte pointer operations.

1.5 ERROR OVERVIEW

The DTE20 will generate/check parity on Deposit/Examine data (36 bits). It will not check or generate parity for CONI, CONO, DATAO, or API words. The software will check for errors by examining the termination words. The parity scheme also imposes one restriction on the byte pointer used for TO11 transfers. A byte size larger than 16 bits cannot be used unless the bits to the left of the rightmost 16 bits contain even parity. If a parity error occurs, the error termination bit status and the EBus parity error flag status will be set. If an Examine operation was in progress when a TO11 transfer operation has an error termination due to an EBus parity error, it is not possible for the software to determine if the Examine operation has a parity error. The EBus parity error is fatal, and is treated so by the Monitor. When a parity error occurs, the bad data is stored in the RAM and can be retrieved for error reporting. The DTE20 sometimes swaps the left and right bytes for byte mode prior to writing the bytes into the RAM. Therefore, the termination TO11 address word should be examined to determine if the left and right halves were swapped. If the termination address is even, the bytes were swapped. (This applies only to transfers in byte mode.)

1.6 DIAGNOSTIC OVERVIEW

The interface contains many features that enable diagnosing of the interface. It is designed to be diagnosed using three basic methods:

1. Without using or disturbing the EBus
2. With loopback on the EBus but without the KL10 or without the KL10 running
3. With the KL10 running.

The interface is primarily checked out in a single-step manner. Full speed operation may only be checked with a running KL10; DIAG1 contains the Diagnose 10/11 Interface bit. When DIAG1 is set, the following occurs. The interface clock is disabled and single step operation commences. Interrupts are inhibited from being sent to the KL10. The interface operates in the normal manner except that EBus operations never complete because no interrupts are issued to the KL10. Therefore, a bit is provided that enables setting EBus Complete, allowing the operation to continue.

The interface control is run by an up-counter and three decoders. The decoders are selected by the major state flip-flops. The up-counter is loadable by the rightmost four bits of DIAG Word 2. This enables any minor logic state to be executed. The major states are not loadable; however, they naturally cycle until a condition occurs that indicates the operation is ready to take place. These major state bits are readable.

1.6.1 Diagnosing the KL10

All KL10 diagnostic functions and console functions (except Deposit and Examine) are performed over the diagnostic portion of the EBus. This specification explains the operation of the diagnostic bus.

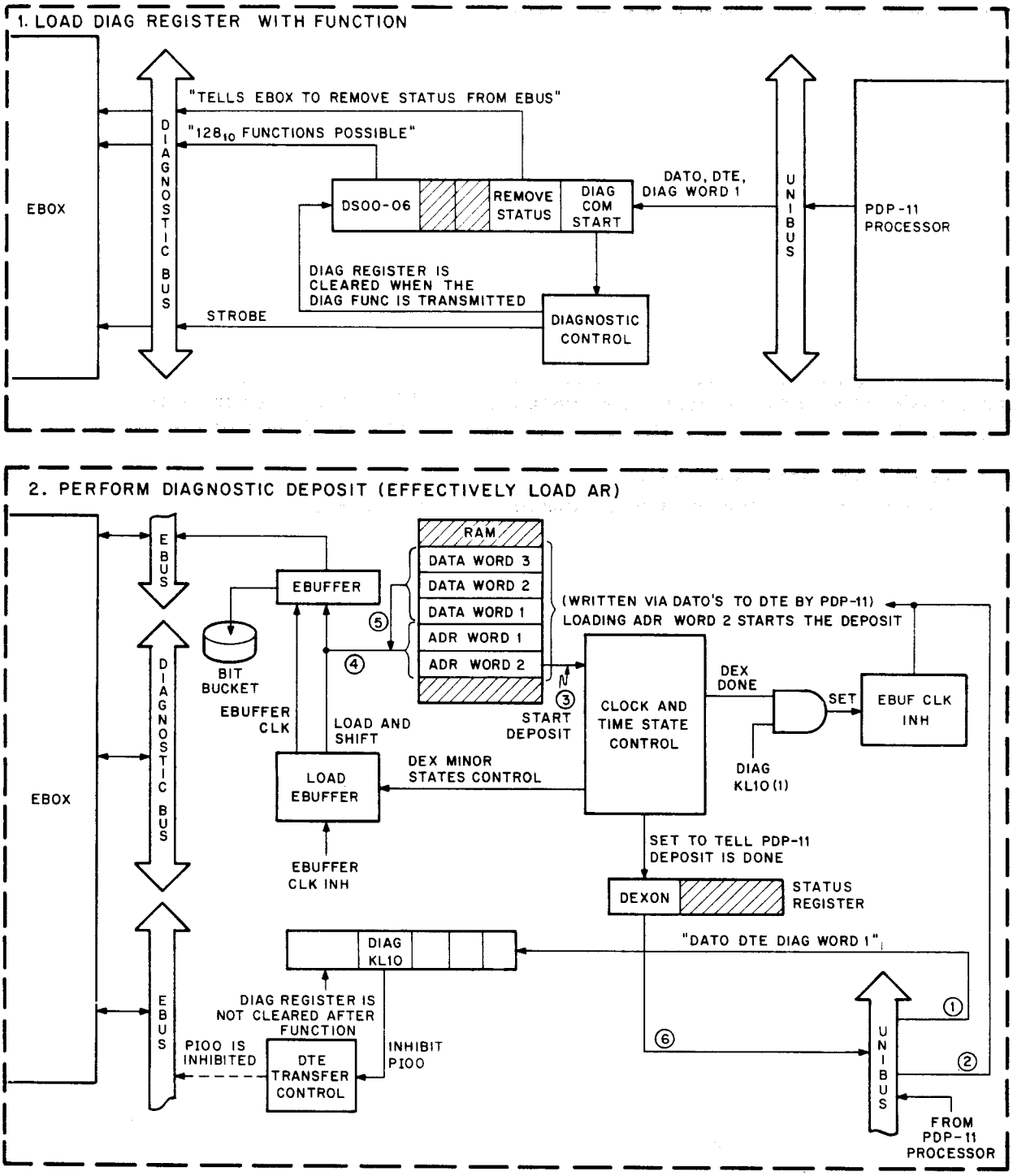
The diagnostic bus contains the following ten signal lines:

DS00-06	Diagnostic Select (DS) Lines – The PDP-11 sends encoded diagnostic functions to the KL10 on these lines. These lines can be read by the PDP-11 at any time, even while the rest of the EBus is active for other devices.
DIAG STROBE	Diagnostic Strobe – This line is asserted to indicate that the Diagnostic Select lines are stable, and that the indicated function should be performed.
DFUNC (Actual Mnemonic is Remove Status)	Diagnostic Function – When true, this causes the KL10 to disable the basic CPU status from the DS lines, switch the translator (only for the DS lines) to convert TTL to ECL, and put the EBus translator under control of DB bits 00 and 01.

1.6.1.1 Diagnostic Bus Control

Diagnostic CPU Status Read – All bits in DIAG Word 1 must be loaded with zeros. The CPU status may then be read from the DS lines after 1 μs has been allowed for the lines to settle (Figure 1-7).

Diagnostic Functions Only (i.e., no 36-bit transfers) – The desired function code bits should be set along with DIAG Command Start (DIAG1 PDCOMST) and Remove Status (DIAG1 [DFUNC]). This will result in the function being sent to the KL10. When DIAG Command Start is a zero, the function has been sent. All function bits must be loaded with the desired value each time a new command is sent. The DIAG Send bit has no effect upon this operation. DIAG KL10 must not be set or a 36-bit data transfer will take place. This operation should not take more than 2.0 μs.



NOTE:

4 Loading ADR1 is not necessary but loading ADR2 is required to begin the deposit operation

5 The clock and time state control loads and shifts DATAWORD 1, 2 and 3 into the EBUFFER which effectively discards ADR words 2 and 1

6 PDP-11 samples state, to determine when deposit is done; then may issue diagnostic function "LOADAR" causing EBOX to read the DATA from the EBUFFER into the AR. See 1

10-1792

Figure 1-7 Diagnostic Overview

1.6.1.2 Diagnostic Functions with 36-Bit Data Transfer

Sending Data to the KL10 – No other operations (i.e., byte string transfers) may be in progress while doing 36-bit diagnostic data transfers. The data should be loaded into DEX WD1-3 (same bit assignments as with a Deposit or Examine). DIAG KL10 should then be set and a Deposit operation should be started. When the transfer is complete (DEXDON SET), the diagnostic function should be loaded as described above, the DIAG KL10, DIAG Send, DIAG Command Start, and DIAG Function set. The operation is complete when DIAG Command Start is on a zero.

Receiving Data From the KL10 – The diagnostic function should be loaded with DIAG KL10 set, DFUNC set (Remove Status), DIAG Send clear, and DIAG Command Start set. When DIAG Command Start is clear, the function is complete and the data is in DEX WD1-3. No other operations (i.e., byte string transfers) may be in progress during this operation.

All the KL10 diagnostic functions are disabled when the privileged restricted mode switch is set to restricted mode. This bit can be tested by Reading Status (RM). When the switch is set to restricted mode, status (RM) is set (i.e., the device is restricted and cannot send diagnostic functions).

1.7 INTERFACE COMMUNICATION

The DTE20 can communicate directly with three devices in the system:

1. EBox via EBus
2. PDP-11 processor via Unibus
3. PDP-11 memory via Unibus.

This communication, when over the Unibus, is in a master-slave relationship (Figure 1-8). During any bus transfer, either the DTE20, the PDP-11 processor, or the PDP-11 memory has control of the bus. The controlling device is considered the bus master, and the device being controlled is considered the slave. Also, communication on the Unibus is interlocked between the DTE20 and either the PDP-11 processor or the memory. Each control signal issued by the master device must be acknowledged by a similar response from the slave device. Thus, communication is independent of bus length and of the response time between the master and the slave. When the DTE20 requests the bus, the handling of the request depends on the location of the interface in a priority structure. The following factors must be considered to determine the priority of the request:

1. The processor's priority is set under program control to one of eight levels using bits 7, 6, and 5 in the processor's Status register. These three bits set a priority level that inhibits granting bus requests (BR) on the same or lower levels.

2. Bus requests from external devices, i.e., DTE20 can be made on any one of five request lines. A non-processor request (NPR) has the highest priority and its request is granted by the processor between bus cycles of an instruction execution.

3. When more than one device is connected to the same bus request line, the one that is electrically closest to the PDP-11 processor has the higher priority.

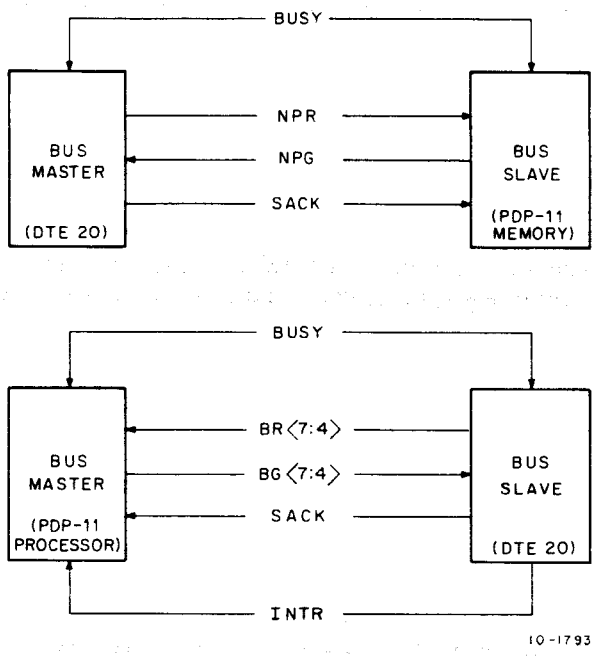


Figure 1-8 Master/Slave Relationship

1.7.1 Data Transfer

Direct memory access data transfers can be carried out by the DTE20 and memory without processor supervision.

This type of transfer is called NPR level data transfer. Normally, NPR transfers are only made between memory and controllers. During NPR transfers, it is not necessary for the PDP-11 processor to transfer the information between the memory and DTE20. The bus structure is such as to enable device-to-device transfers. This allows special controllers to access other devices on the bus as well as memory.

The DTE20 can transfer data at high rates once it gains control of the bus. In addition, the processor's internal state is not affected by this type of transfer. Therefore, the processor can release the bus while an instruction is in progress. The DTE20 can transfer 16-bit or 8-bit bytes to memory at the same speed as the memory cycle time.

1.7.2 Interrupt Requests

Once the DTE20 has gained control of the bus, it can take full advantage of the power and flexibility of the processor by requesting an interrupt. Note that interrupt requests can be made only if bus control has been gained through a BR priority level. An NPR level request cannot be used for an interrupt request.

1.7.2.1 Processor Interrupt Procedure – Assume that the DTE20 is responding to a CONO from the EBox and this CONO is activating the doorbell feature for interprocessor communication. The DTE20 must then interrupt the PDP-11 to inform it that the EBox wishes to communicate. Refer to Figure 1-9. The following takes place:

1. Priorities permitting, the processor relinquishes bus control to the DTE20.

2. When the DTE20 gains control of the bus, it sends the PDP-11 processor an interrupt command and the starting address of the device service routine. This is called the Interrupt Vector Address. Immediately following this address is a word to be used as the new processor status (PS) word.

3. The processor pushes the current processor status word and then the program counter (PC) value onto the processor stack. The stack is pointed to by register 6.

4. The new PC and PS (the interrupt vector) are taken from the address specified by the DTE20 and the service routine is initiated.

5. The service routine can cause the processor to resume the interrupted process by executing the return from interrupt (RTI) instruction, which pops the two top words from the processor stack and transfers them back to the PC and PS registers.

1.7.3 Unibus Signal Lines

The PDP-11 Unibus comprises 56 lines. All devices including the processor are connected to these lines in parallel. The bidirectional nature of 51 signal lines permits signals to flow in both directions. The remaining five lines are used for priority bus control. Table 1-1 lists the data transfer signals.

Table 1-1
Data Transfer Signals

Name	Mnemonic	No. of Lines
Data	D (15:00)	16
Address	A (17:00)	18
Control	C0, C1	2
Master Sync	MSYN	1
Slave Sync	SSYN	1
Parity	PA, PB	2
Interrupt	INTR	1
		Total: 41

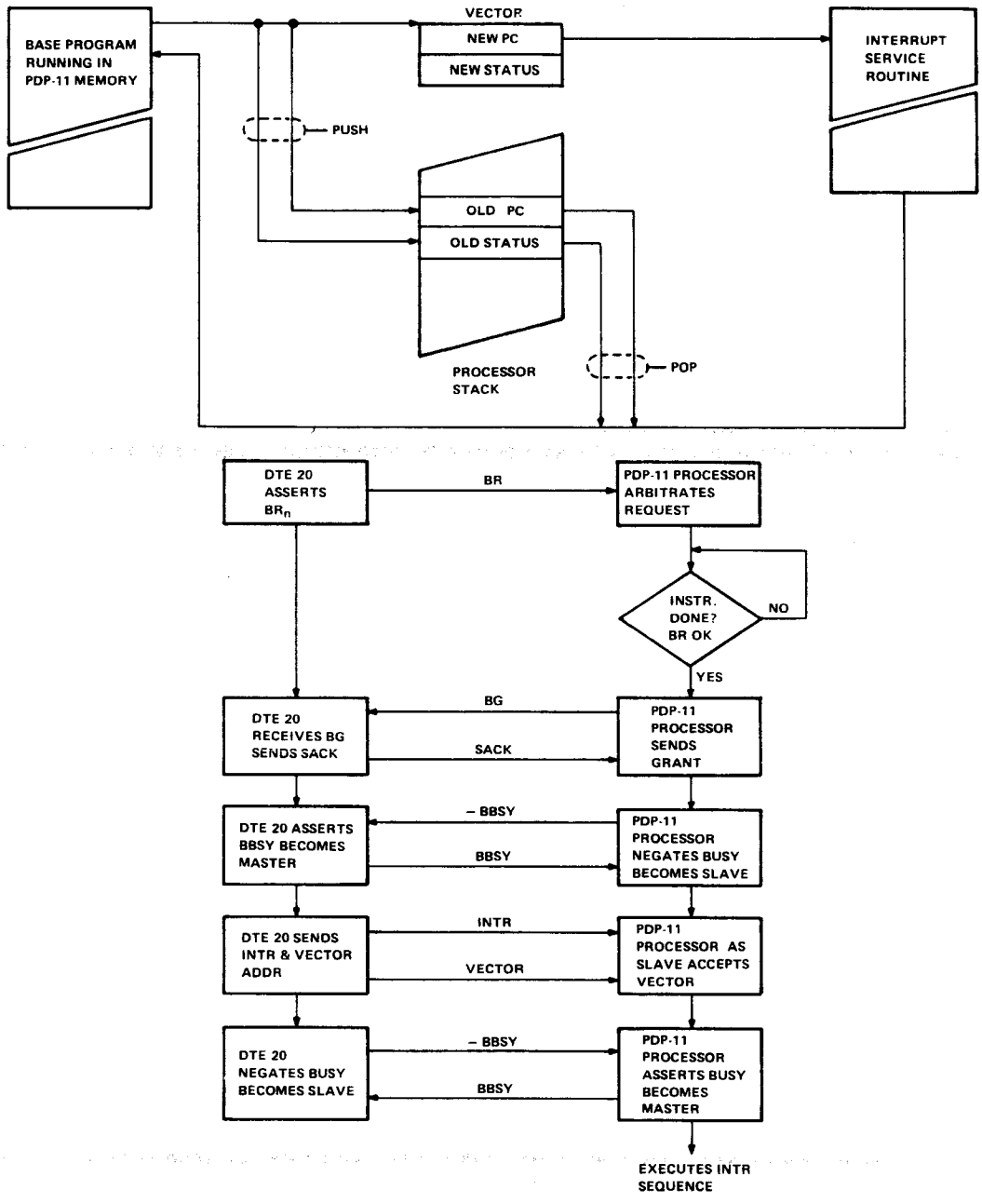


Figure 1-9 BR Sequence

Data Lines D(15:00) - The 16 data lines are used to transfer information between the DTE20 and either the PDP-11 processor or PDP-11 memory. The most significant bit is bit 15, the least significant is bit 00.

Address Lines A(17:00) - The 18 address lines are used by the DTE20 to select the PDP-11 memory address used in the current data transfer. The reason for 18 address lines is to extend the total memory capability to 262,144 bytes.

The extension bits are bits 17 and 16. The normal most significant bit is bit 15; the least significant bit is bit 00. Lines A(17:01) specify a unique 16-bit word. In byte operations, A00 specifies the byte being referenced. If a word is referenced as ADR (ADR must be even, because words can be addressed only on even boundaries), the low order byte can be referenced at ADR and the high order byte at ADR+1.

Only 16 bits are supplied by programs as memory references. In the processor, lines A17 and A16 are asserted (forced to 1) whenever the program attempts to reference an address between 160000 and 177777. Thus, the processor converts the 16-bit to a full 18-bit address.

Control Signals - The control signals are divided into three groups: signals that select data transfer operations, signals that allow the master and slave device to communicate, and signals used for parity checking.

1. Control Lines C(1:0) - These two bus signals are coded by the DTE20 as well as the PDP-11 processor to control the memory in one of four possible data transfer operations shown in Table 1-2.

Table 1-2
Data Transfer Operations

C1	C0	Operation
0	0	DATI -- data in
0	1	DATIP -- data in pause
1	0	DATO -- data out
1	1	DATOB -- data out byte

2. Master and Slave Synchronization - Master Synchronization (MSYN) is a control signal used by the master device to indicate to the slave device that address and control information is present. Slave Synchronization (SSYN) is the slave device response to the master.

3. Parity Error Indicators - The PA and PB are used to indicate that a memory parity error occurred on a memory read. The DTE sets the DPS5 MEM PAR ERR flag when a memory parity error is indicated during an NPR transfer initiated by the DTE.

Interrupt (INTR) – This signal is asserted by the DTE20 to start a priority interrupt in the processor.

1.7.3.1 Priority Transfer Lines – The Unibus contains 12 lines classified as priority transfer lines. Four of these are priority bus request lines BR(7:4) and four are the corresponding grant lines BG(7:4); NPG, NPR, SACK, and BBSY complement the priority transfer lines.

Each device of the same priority level passes a grant signal to the next device on the line, unless it has requested bus control; in this case, the requesting device blocks the signal from the following devices and assumes bus control. These 12 lines are described as follows:

1. Bus Request Lines BR(7:4) – These four bus signals are used by the DTE20 to request control of the bus.

2. Bus Grant BG(7:4) – These signals are the processor's response to a bus request. They are asserted only at the end of instruction execution, and in accordance with the priority determination.

3. Non-Processor Request (NPR) – This signal is the bus request from the DTE20 to the PDP-11 processor for a DMA-type bus cycle.

4. Non-Processor Grant (NPG) – This signal is the processor's response to an NPR.

5. Selection Acknowledge (SACK) – This signal is asserted by the DTE20 after receiving Bus Grant (BG). Bus control passes to the DTE20 as soon as the current bus master has completed its operation. If SACK is not received by the processor within 28 μs of issuing BG, a timeout occurs and the Bus Grant is cleared automatically by the processor.

6. Bus Busy (BBSY) – This signal is asserted by the master, either the DTE20 or PDP-11 processor, to indicate that the bus is being used.

1.7.3.2 Miscellaneous Control Lines

Initialization (INIT) – This signal is asserted by the processor when the START key on the console is pressed, when a RESET instruction is executed, or when a power failure sequence occurs. In the later case, INIT is asserted following the power fail service routine while power is going down, and again when power comes up.

1.7.4 EBus Signal Lines

The EBus consists of 60 signal lines. All devices, including the KL10, are connected to these lines in parallel. The bidirectional nature of 36 of the signals permits some information to flow in both directions. These are the data lines. The remaining 24 signal lines are used for control functions. Table 1-3 lists the signals necessary to effect a data transfer.

Data Lines D(00:35) – The 36 data lines are used to transfer information between the EBox and the DTE20. The most significant bit is bit 00, the least significant bit is bit 35.

Controller Select Lines CS(00:06) – These seven lines are used to select the desired controller for a data transfer. Each controller has a unique select code that is hard-wired on the backplane of the device.

Table 1-3
Data Transfer Signals

Name	Mnemonic	No. of Lines
Data	D (00:35)	36
Controller Sel	CS (00:06)	7
Function	F (00:02)	3
Demand	DEM	1
Acknowledge	ACK	1
Transfer	XFER	1

Function Lines F(00:02) – The function lines specify the type of data transfer or non-data transfer that is to take place. Table 1-4 lists the four implemented functions.

Table 1-4
Data Transfer Commands

F00	F01	F02	Operation
0	0	0	CONO
0	0	1	CONI
0	1	0	DATAO
0	1	1	DATAI

DEMAND (DEM) – This signal line causes the addressed controller to sample the CS lines and the F lines and to decode their meaning. Upon implementing the specified function, TRANSFER and ACKNOWLEDGE are asserted as a response, along with data being placed onto or taken from the EBus as specified by the decoded function.

ACKNOWLEDGE (ACK) – This signal line is necessary to tell the DIA20 I/O Bus Adapter not to respond to the current operation. If the DIA20 does not see ACKNOWLEDGE some period of time after DEMAND is asserted, it will try to perform the transfer. It does not decode the CS lines, as do the standard KL10 devices.

TRANSFER (XFER) – This line is asserted by the selected controller when it is ready to execute the specified function as decoded in F(00:02).

1.7.4.1 Priority Transfer Lines – To perform priority interrupts between the KL10 and its devices, the same basic set of signals is used in a slightly modified form. Table 1-5 lists the necessary signals.

Table 1-5
Priority Transfer Signals

Name	Mnemonic	No. of Signals
Controller Sel	CS (04:06)	3
Controller Sel	CS (00:03)	4
Function	F (00:02)	3
Demand	DEM	1
Acknowledge	ACK	1
Transfer	XFER	1

Controller Select CS(04:06) – During interrupt arbitration, these three lines represent the octal encode of the interrupting channel. The range is 0 through 7.

Controller Select CS(00:03) – These four lines specify the controller or device that the EBox will honor during this interrupt sequence. This is, of course, only a single device or controller, even though several may be interrupting on the same channel. This code will also correspond to the hard-wired physical device number of the appropriate controller or device.

Function F(00:02) – Two functions are generated during the interrupt dialogue; refer to Table 1-6. The first is a code of 4 in F(00:02). It specifies to the interrupting controllers that those being addressed by channel number in CS(04:06) should send their physical controller number by placing them onto the EBus upon sensing DEMAND. The second function is a code of 5 in F(00:02). It specifies to the interrupting controllers or devices that one has been selected. The selected one will see CS(00:03) as the same number as its physical controller number.

Table 1-6
Priority Transfer Commands

F00	F01	F02	Operation
1	0	0	PI SERVED
1	0	1	PI ADDRESS IN

ACKNOWLEDGE (ACK) – Same as for data transfers.

TRANSFER (XFER) – In the case of interrupts, the device selected for service by the EBox will place a special function on the EBus data lines D(00:35). Refer to Figure 1-2, API Word Format.

The vector interrupt locations for the PDP-11 are as follows: 774, 770, 764, 760 for the first, second, third, and fourth interfaces on a single PDP-11 respectively. The high-order PDP-11 address bits are listed in Table 1-7.

Table 1-7
PDP-11 Device Registers

Interface	Assignment
0	774400
1	774440
2	774500
3	774540

All of the necessary registers for implementing the specified types of data or non-data transfers have been included in the interface. In general, the majority of these registers are addressable by the PDP-11 for read or for write. They are not selectable from the EBox.

1.8 INTERFACE DATA AND CONTROL BUFFERING

To facilitate efficient interprocessor data transfers, with minimum intervention by either the EBox or PDP-11 processors, a storage medium in the form of a semiconductor Random Access Memory (RAM) containing 16 words × 16 bits per word of storage has been included as a part of the interface. The access time of the RAM is about 125 ns. The available RAM storage is sufficient to contain all of the necessary control and data words used to perform the DTE20's four major hardware operations as a free standing element. In addition, the RAM can be loaded and read under the direction of a diagnostic program resident in the PDP-11 processor and can also be used to control the DTE20 during diagnostic operations.

1.8.1 Addressable Register Summary

Table 1-8 provides a summary of all of the internal addressable locations within the DTE20. Sixteen locations are listed; twelve are RAM locations and the remaining four are register locations.

Diagnostic Word 3 – This word can be read at any time from the PDP-11 processor. It consists of interface control signals and addresses, along with the TO10 byte mode bit.

Status – This register consists of an EBox portion and a PDP-11 portion. The detailed bit assignments for both are given in Tables 1-13, 1-14, 1-15, and 1-16. Register bit assignments generally consist of Done and Error flags for the various operations, a flag for each processor to interrupt the other, and miscellaneous other flags.

NOTE

Each machine has its own separate copy of each status bit, except for the doorbells (e.g., there is a 10 TO10 Normal Termination flag and an 11 TO10 Normal Termination flag).

Diagnostic Word 2 – This word is similar to Diagnostic Word 3 in that it can be read by the PDP-11 processor at any time simply by addressing the appropriate register within the DTE20. Under diagnostic control, this word can be written to control the DTE20 minor states in a single pulse fashion. Also, because no real interrupts are sent to the EBox during diagnosis of the DTE20, a bit in this word can enable the appropriate flag to set to simulate a response from the EBox as if it responded to the interrupt. Finally, it can enable the current major state to lock out any changes in that state until such time as it is desired to do so.

The remaining registers are RAM locations and are summarized, together with the DTE programming information, in Tables 1-9 through 1-19 of this section.

Table 1-8
Addressable Register Summary

Register Name	Accessible By:	PDP-11 INSTR	DTE20 ADR	KL10 INSTR	FCN CODE
DIAG3	PDP-11	DATO, DATI	XXX36	–	–
STATUS	BOTH PDP-11 and KL10	DATO, DATI	XXX34	CONO, CONI	0,1
DIAG2	PDP-11	DATO, DATI	XXX32	–	–
DIAG1	PDP-11	DATO, DATI	XXX30	–	–
TO11 DATA	PDP-11	DATO, DATI	XXX26	–	–
TO10 DATA	PDP-11	DATO, DATI	XXX24	–	
TO11 ADR	PDP-11	DATO, DATI	XXX22	–	–
TO10 ADR	PDP-11	DATO, DATI	XXX20	–	–
TO11 BYTE CNT	PDP-11	DATO, DATI	XXX16	–	–
TO10 BYTE CNT	BOTH PDP-11 and KL10	DATO, DATI	XXX14	DATAO	2
ADDRESS WORD 2	PDP-11	DATO, DATI	XXX12	–	–
ADDRESS WORD 1	PDP-11	DATO, DATI	XXX10	–	–
DATA WORD 1	PDP-11	DATO, DATI	XXX06	–	–
DATA WORD 2	PDP-11	DATO, DATI	XXX04	–	–
DATA WORD 3	PDP-11	DATO, DATI	XXX02	–	–
DELAY COUNT	PDP-11	DATO, DATI	XXX00	–	–

Table 1-9
Deposit or Examine Word Formats

Word	Bits	Function
DEXWD1	15-04	Must be zero, reserved by DEC
	03-00	KL10 data bits 0-3
DEXWD2	15-00	KL10 data bits 4-19
DEXWD3	15-00	KL10 data bits 20-35
TENAD1	15-13	Address space
	12	Deposit bit 1 = deposit 0 = examine
	11	PRTOFF if 1 protection and relocation is off for examines and deposits for a restricted mode DTE
	10-09	Must be zero, reserved by DEC
	08-07	Must be zero, reserved by DEC
	06-00	High order KL10 address bits (13-19)
TENAD2	15-00	Low order KL10 address bits (20-35)

Table 1-10
TO10 Transfer Word Format

Word	Bits	Function
DLYCNT	15-14	Unibus address bits 17-16. Specifies two high-order bits of 18 bit PDP-11 address used in 18 bit byte transfer addresses. Transfer cannot cross a 32K boundary. TO10 and TO11 transfers must be in the same 32K bank.
	13-00	NEGATIVE DLY COUNT – The software specifies how many 500-nanosecond units of delay are to occur between each byte on byte transfers in either direction. The delay also applies before the first byte. During the transfer operation the DTE up counts a copy of this count in the ABC register, once each 500 nanoseconds, until bit 13 = 0. **NOTE** **The count is incremented by 1 and then the hardware tests bit 13 of ABC for = 0. Therefore both values of 17777 and 00000 are equivalent to no delay.**
TO10AD	15-00	Byte address of source string. This is updated by the DTE as each byte is transferred. At the end of a transfer, it points to the byte (word) which would have been transferred next from PDP-11 memory. The update is by +1 for byte mode and +2 for word mode.
TO10BC	15	If a 1, this bit interrupts both processors at the completion of the current transfer. If a 0, it interrupts the -10 only.
	14-12	Must be zero, reserved by DEC.
	11-00	Negative byte count.
TO10DT	15-08	High order byte PDP-11 byte mode: equal to 0 PDP-11 word mode: bits to become KL10 data bits 20-27
	07-00	Low order byte bits to become KL10 data bits 28-25

Table 1-11
TO11 Transfer Word Format

Word	Bits	Function
DLYCNT	15-14	Same as for TO10 transfer – see Table 1-10.
	13-00	Same as for TO10 transfer – see Table 1-10.
TO11	15-00	Byte address in PDP-11 memory of where to store next byte received from EBox. This word is updated as each byte (word) is transferred. At the end of a transfer, it points to the byte (word) that would have been transferred next. The update is by +1 for byte mode and +2 for word mode.
TO11BC	15	I Bit – If 0, on normal termination interrupt only the PDP-11. If 1, on normal termination interrupt both the PDP-11 and EBox. If an error occurs, the I bit is ignored and both the PDP-11 and EBox always get an error termination interrupt.
	14	Z Stop – If 1, stop on a null character received from the EBox, after storing it in PDP-11 memory. The TO11AD is not incremented so that the next transfer can start by overwriting the null character if desired.
	13	TO11BM – If 1, set byte mode in the DTE, if 0, set word mode in the DTE for TO11 transfer.
	12	Must be zero, reserved by DEC.
	11-00	Negative byte count
TO11DT	15-08	High order byte PDP-11 byte mode: KL10 bits 28-35 or 20-27 PDP-11 word mode: KL10 bits 20-27
	07-00	Low order byte PDP-11 word mode: KL10 bits 28-35 PDP-11 byte mode: KL10 bits 20-27 or 28-35

Table 1-12
DATAO DTE Function

Bits	Function
0-22	Must be zero, reserved by DEC
23	TO11IB – This is the "I" bit. If 1 set TO10IB. If 0, clear TO10IB. If 1, the EBox has set the "I" bit for a TO10 byte transfer. Both the EBox and the PDP-11 will be interrupted on normal termination. If 0, the EBox has not set the "I" bit for a TO10 byte transfer. Only the EBox will be interrupted on normal termination. The EBox may then reset the TO10 byte pointer, before reloading the TO10 byte count and performing a scatter read.
24-35	Negative Byte Count – The twos complement of the number of characters left to transfer until a TO10 normal termination occurs. A -1 will transfer one character to the EBox before a normal termination. A 0 will transfer 0 bytes before a normal termination.

Table 1-13
CONI DTE Function

Bits	Function
0-19	Read as zeros
20	RM – a 1 in this bit indicates that the DTE is in restricted mode, a 0 in this bit indicates that the DTE is in privileged mode.
21	DEAD11 – a 1 in this bit indicates that the PDP-11 power is not correct (the Unibus signal "AC LOW" is asserted) and that no transfers can take place.
22	TO11DB – a 1 in this bit indicates that the EBox has requested a PDP-11 doorbell interrupt and is waiting for the PDP-11 to take some action.
23-25	Read as zeros.
26	TO10DB – a 1 in this bit indicates that the PDP-11 has requested a doorbell interrupt and is waiting for the EBox to take some action.
27	TO11ER – a 1 in this bit indicates that an error occurred during a TO11 transfer.
28	Read as zero
29	TO11DN – a 1 in this bit indicates that a TO11 transfer was completed and an error did not occur. The "I" bit had been set by the PDP-11. The transfer is completed if: 1. The byte count became equal to zero. or 2. The PDP-11 had set the "Z" bit and a null character was encountered.
30	TO10DN – a 1 in this bit indicates that the byte counter for the TO10 transfer became equal to zero and an error did not occur.
31	TO10ER – a 1 in this bit indicates an error (PDP-11 memory parity or Unibus timeout error, but not EBus Parity error) occurred during the TO10 transfer.
32	PI0ENB – a 1 in this bit indicates that the DTE is enabled to perform examines, deposits and byte transfers at PI Level 0 by the EBox. **NOTE** **The DTE is automatically enabled if the PDP-11 is privileged even though this bit is 0.**
33-35	The current PI channel assignment for doorbell interrupts, byte transfer normal and error terminations.

Table 1-14
CONO DTE Bit Function

Bits	Function
18-21	Must be zero, reserved by DEC.
22	TO11DB – causes a doorbell interrupt in the PDP-11, setting the 10 request interrupt flag in the DTE status register. This flag can only be cleared by the PDP-11.
23	CR11B – clears the reload PDP-11 button in the DTE.
24	SR11B – sets the reload PDP-11 button in the DTE. Setting this bit in the status register initiates the ROM bootstrap in the PDP-11.
25	Must be zero, reserved by DEC.
26	CL11PI – clear the PDP-11 requesting 10 interrupt flag in the DTE status register.
27-28	Must be zero, reserved by DEC.
29	CLTO11 – clear both the TO11 normal termination, and also the TO11 error termination flags in the DTE.
30	CLTO10 – clears both the TO10 normal termination, and the TO10 error termination flags in the DTE.
31	PILDEN – loads the PI Interrupt Channel number from bits 33-35 and PI Level 0 enabled from bit 32.
32	PIOENB – enables PI0.
33-35	PI Channel Number. Loaded if bit 31 is equal to 1.

Table 1-15
DATO DTE Status Function

Bits	Function
15	DON10S – If 1, set TO10 normal termination status (TO10DN). This bit is provided for diagnostic purposes only. Setting it via a DATO does not terminate a transfer in progress.
14	DON10C – If 1, clear TO10 normal termination status (TO10DN).
13	ERR10S – If 1, set TO10 error termination status (TO10ER). This bit is provided for diagnostic purposes only. Setting it via a DATO does not terminate a transfer in progress.
12	ERR10C – If 1, clear TO10 error termination (TO10ER).
11	INT11S – If 1, set 10 request PDP-11 interrupt (TO11DB). This results in a PDP-11 vector interrupt.
10	INT11C – If 1, clear 10 request PDP-11 interrupt (TO11DB). This enables more doorbell interrupts to the PDP-11 to occur.
09	PERCLR – If 1, clear the PDP-11 memory parity error flag (11MPE).
08	INT10S – If 1, set request 10 interrupt (TO10DB). This results in a vectored interrupt to EPT location 142 + 8 * n.
07	DON11S – If 1, set TO11 normal termination flag (TO11DN). This bit is provided for diagnostic purposes only. Setting this bit does not terminate a transfer in progress.
06	DON11C – If 1, clear TO11 normal termination flag (TO11DN).
05	INTRON – If 1, enable DTE to generate PDP-11 BR requests. Clearing or setting this bit does not clear any interrupts waiting.
04	EBUSPC – If a 1, clear EBus parity error.
03	INTROF – If 1, disable DTE from generating PDP-11 BR requests. Clearing or setting this bit does not clear any interrupts waiting.
02	EBUSPS – If 1, set EBus parity error.
01	ERR11S – If 1, set TO11 error termination flag (TO11ER). This bit is provided for diagnostic purposes only. Setting it does not terminate a transfer in progress.
00	ERR11C – If 1, clear TO11 error termination flag (TO11ER).

Table 1-16
DATI DTE Status Function

Bits	Function
15	TO10DN – The TO10 byte count became 0 or the PDP-11 program set DON10S. TO10DN will not be set if an error termination occurred, i.e., TO10ER.
14	Read as zero, this bit is unused.
13	TO10ER – an NPR Unibus parity error (DIAG3 [NUPE]), PDP-11 memory parity (status [11MPE]), or a Unibus timeout (no bit) occurred during a TO10 byte transfer, or the PDP-11 program set the status bit (ERR10S). Status bit (TO10DN) will not be set, if an error termination occurred. Thus, PDP-11 programs must test for both TO10DN and TO10ER.
12	RAMIS0 – The data out of the RAM location is all 0s. This bit is provided on a read for diagnostic purposes only. It has no meaning and is unpredictable unless the DTE is being single stepped.
11	TO11DB – The 10 has requested (via CONO DTEN) a PDP-11 doorbell interrupt.
10	DEXWD1 – This bit is provided for diagnostic purposes only. It has no meaning and is unpredictable unless the DTE is being single stepped.
09	MPE11 – Indicates that the PDP-11 memory had a parity error during a data fetch for a TO10 byte transfer. Parity errors are detected only if the PDP-11 has the MF11UP or MP11LP memory parity option.
08	TO11DB – The PDP-11 has requested a -10 doorbell interrupt (INT10S) and the -10 has not yet cleared the bit (via CONO DTEN) using CL11PI.
07	TO11DN – The TO11 byte count became equal to 0 (TO11BC = 0), the transfer stopped on a MU11 character (status bit NULSTP = 1), or the PDP-11 program set status bit DON11S.
06	EBSEL – E Buffer Select. This bit is provided for diagnostic purposes only. It has no meaning and is unpredictable unless the DTE20 is being single stepped.
05	NULSTP – Null Stop. The TO11 transfer stopped because the stop bit was set (TO11BC [Z Stop]) = 1.
04	B PAR ER – EBus Parity Error. The DTE detected an EBus parity error during a TO11 byte transfer or examine transfer.
03	RM – If 1, the attached PDP-11 is in restricted mode. If 0, the attached PDP-11 is in privileged mode. The value of this bit is determined by the setting of the privileged switch on the DTE20.
02	DEXON – The last deposit or examine operation has been completed. No interrupt occurs. The PDP-11 must watch for this bit to be set after every deposit or examine. The DTE20 clears status bit DEXON whenever a deposit or examine is started (by loading TENAD2).
01	TO11ER – an error occurred during a TO11 byte transfer or the PDP-11 program sets the status bit ERR11S. Status bit TO11DN will not be set if an error termination occurred. Thus, programs must test for both TO11DN and TO11ER.
00	INTSON – Interrupts on. If 1, the DTE is enabled for PDP-11 BR requests. If 0, it is disabled (INTRON enables, INTROF disables).

Table 1-17
DATI/DATO DTE DIAG Word 1

Bits	Function	
	DS REMOVE STATUS FALSE and all DS bits = 0 (Any DS bits = 1 are considered illegal)	DS REMOVE STATUS TRUE
	Receive processor status bits.	Observe up to 128 DS bits asserted by the DTE.
15-12	Unused	
11	DS04-1 = KL Clock Error Stop. The KL10 internal clock (32 MHz) has frozen due to a hardware malfunction of one of the following: CRAM, DRAM, Fast Memory Parity Error, or Field Service test condition.	**NOTE** **Unless the DS Bits are asserted without DCOMST (Diagnostic Command Start), this will always read as zeros because DCOMST clears the register at the end of a diagnostic cycle.**
10	DS05-1 = RUN (1). The microcode examines this flag between functions. The microcode enters a Halt Loop if this flag is off. This flag is under control of the PDP-11, using two diagnostic functions. The KL10 cannot affect **RUN**.	
09	DS06-1 = HALT (1). This signal is set when the microcode enters the Halt Loop and clears the Signal when it leaves the Loop.	
08	DEX – Deposit or Examine major state. WRITE: Must be zero. READ: A 1, indicates interface major state is deposit or examine.	
07	TO10 – READ: a 1, indicates interface major state is TO10 transfer. 0 indicates not in TO10 transfer state. DFUNC (Remove Status) – WRITE: A 1, causes the EBox to stop sending basic status on the DS lines, so that a loopback test can be performed on the DS lines or a DIAG FUNC can be sent to the EBox via the DS lines. If any of the DS lines are set (by the DTE) the result is an "OR" of the bits set in the DTE and EBox status.	
06	TO11 – READ: A 1 means interface major state is TO11 transfer. WRITE: Must be zero.	
05	D1011 – Diagnose 10/11 interface. READ: If a 1, the DTE is in 10/11 diagnostic mode, i.e., it will diagnose itself. If a 0, it is not in 10/11 diagnostic mode. WRITE: If a 1, set DTE to 10/11 diagnostic mode. This mode is used to diagnose the DTE itself. If a 0, leave 10/11 diagnostic mode.	

Table 1-17 (Cont)
DATI/DATO DTE DIAG Word 1

Bits	Function
04	VEC04 – Vector Interrupt Address bit 4: READ: Vector interrupt address bit 4 PULSE-WRITE: If a 1, generate a single clock cycle. If 10/11 diagnostic mode (D1011 status bit) is also set.
03	DIKL10 – READ: This bit is read as zero. WRITE: If a 1, and the DTE is in privileged mode, put the DTE into KL10 diagnostic data transfer mode. Subsequent deposits and examines become diagnostic functions instead of accessing KL10 memory. If a 0, put the DTE in normal data transfer mode.
02	DSEND – READ: This bit is read as zero. WRITE: If a 1, send data (TO10) during a diagnostic bus transfer. If a 0, receive data (TO11) during a diagnostic bus transfer.
01	This bit is unused and must be zero on a write. It is read as zero.
00	DCOMST – Diagnostic command start. READ: If a 1, a diagnostic command is in progress. WRITE: If a 1, and the DTE is switched to privileged mode, diagnostic command start is set. If a 0, diagnostic command start is cleared.

Table 1-18
DATI/DATO DIAG Word 2

Bits	Function
15	RFMAD0 – RAM File Mixer Address 0. READ: The contents of RFM address 0. WRITE: Must be zero.
14	RFMAD1 – RAM File Mixer Address 1. READ: The contents of RFM Address 1. EDONES – WRITE: If a 1, set EBus done. If a 0, clear EBus done.
13	RFMAD2 – RAM File Mixer Address 2. READ: The contents of RFM Address 2. WRITE: Must be zero.
12	RFMADR3 – RAM File Mixer Address 3. READ: The contents of RFM address 3. WRITE: Must be zero.
11-07	Unused – READ: Read as zeros. WRITE: Must be zero.
06	DRESET – DTE Reset. READ: Read as zero. WRITE: If a 1, reset the DTE.
05	Unused – READ: Read as zero. WRITE: Must be zero.
04-01	READ: Read as zeros. WRITE: Loads 04, 03, 02, 01 into minor state counter 8, 4, 2, 1 for diagnostic use only. (During normal operation must be zero.)
00	Unused – READ: Read as zero. WRITE: Must be zero.

Table 1-19
DATI/DATO DIAG Word 3

Bits	Function
15	SWSLLT – Swap Select Left READ: CNT1 [N] Swap Select LT. WRITE: Must be zero.
14	DPS4 [N] Parity (1) H READ: DPS4 [N] Parity flop is on a one. Diagnostic use only. WRITE: Must be zero.
13-08	Captured Unibus Parity Error Information. READ: Ann indicates Unibus register address bit, Dnn indicates Unibus data bit, when a Unibus parity error is detected. TIME — UNIBUS DATA BITS Initially — D15 D14 D13 D12 D11 A00 1st Shift — D10 D09 D08 D07 D06 A00 2nd Shift — D05 D04 D03 D02 D01 A00 3rd Shift — D00 A04 A03 A02 A01 A00 4th Shift — D15 D14 D13 D12 D11 A00 . . . . . . WRITE: Must be zero.
07-06	Unused READ: Read as zeros. WRITE: Must be zeros.
04	DUPE – DATO Unibus Parity Error. READ: If 1, a DATO Unibus parity error has been detected by the DTE. CDD – Clear DUPE and DURE error flags.

Table 1-19 (Cont)
DATI/DATO DIAG Word 3

Bits	Function
03	WEP – Write even (bad) parity. READ: Read the status of the write even Unibus parity flip-flop. WRITE: If a 1, write even Unibus parity. Results in DTE generating even (bad) parity on all Unibus transfers which have parity. If a 0, the DTE will generate odd (good) parity on all subsequent Unibus transfers which have parity. This bit is provided for diagnostic purposes to check the parity networks.
02	DURE – DATO Unibus receive error. READ: A Unibus receiver error has occurred. WRITE: Must be zero.
01	NUPE – NPR Unibus parity error. READ: If a 1, a Unibus parity error has occurred on an NPR (byte) transfer. CNUPE – WRITE: Clear NUPE.
00	TO10BM – TO10 byte mode. READ: Read as zero. WRITE: If a 1, TO10 byte transfers are to be performed in byte mode from the PDP-11 memory. If a 0, TO10 byte transfers are to be done in word mode from PDP-11 memory.

1.9 BUS OPERATION

Functionally, the DTE20 operates in two ways. One way is an operation that fetches data; the other is an operation where the data must be loaded by an operating program. In the first method, a single bus operation may access the interface from the PDP-11 to read or write data (or control information) into the interface RAM file. This is done by using DATI and DATO instructions and involves the Unibus. This type of operation activates the Interface Control Logic long enough for the RAM access. No dialogue, other than the normal dialogue that would take place between the PDP-11 and any standard peripheral device, i.e., MSYN, SSYN, etc., occurs. Similarly, the PDP-11 can read or write status information to or from the interface using this same mode of operation. In the second method, the interface detects the loading of control information from the PDP-11, or from both the PDP-11 and KL10 processors, and begins processing this information. The interface initiates an internal timing sequence that includes the necessary interprocessor dialogue. The result of this operation is to transfer information between the two processors using both the EBus and the Unibus, where the direction of transfer is a function of the operation being performed.

The DTE20 performs four basic hardware operations:

DEX (Deposit or Examine)
TO11 Transfer/TO10 Transfer
Interprocessor Doorbell.
Diagnostic Functions

Multiple transfers take place for both TO11 and TO10 transfers; however, the source and destination differ. In the TO11 transfer, the source of the data is the KL10 memory and the destination is the PDP-11 memory. The situation is reversed for TO10 transfers: the source is the PDP-11 memory and the destination is the KL10 memory.

The DEX (Deposit or Examine) operation differs from a TO10 or TO11 transfer operation by the fact that a single transfer occurs. Also, no interrupt is generated to the PDP-11 upon completion of the operation.

The interprocessor doorbell is unique in that it uses only the "peripheral bus control logic" (i.e., the BR Control and the EBus Dialogue Logic) and does not initiate the internal time state logic.

Data entering the DTE20 on the EBox side does so in 36-bit words, which are converted into 16-bit words within the interface and stored in the RAM FILE. From there, the buffered data word can be transmitted to the PDP-11 under control of the Interface Control Logic by the NPR facility. Data entering the DTE20 on the PDP-11 side does so in 16-bit words that are stored in the RAM FILE. From there, the buffered word can be transmitted to the EBox under the control of the Interface Control Logic, using the interrupt control and the EBox side interrupt logic. This transfer also involves the E-Buffer register.

Two basic classes of transfers can be performed in terms of implementation. Table 1-8 contains a list of PDP-11 addresses given to select the appropriate RAM address within the interface, the functional name for that particular RAM slot, and a description of that slot's usage during the appropriate interface operations.

The first class is Deposit and Examine and deals with an address in KL10 memory and a data word that is either sent or received to or from KL10 memory as specified by the address word. These two operations involve five temporary RAM storage slots for address and data.

The second class is TO11 transfer, or TO10 transfer, and deals with a Delay Count word, a Byte Count word, an address word, and a data word. All of these words must have storage space in the RAM FILE. Before transfers for either class of operation begin, the necessary words that control the particular transfer must be supplied, by the processor, to the RAM FILE.

Refer to Figures 2-1 and 2-2. The PDP-11 can read or write all RAM locations and also the Diagnostic and Status registers via the Unibus. These registers have internal addresses (shown in brackets [ ]); for example the Status register is address [16]. To set up a transfer within the interface, the PDP-11 performs DATO instructions while addressing the appropriate RAM locations. The EBox cannot address any of the RAM locations and, in fact, is only required to supply one piece of control information. This is the TO10 Byte Count word, necessary in the TO10 transfer operation. It is supplied by performing a DATAO DTE X instruction in the EBox. The TO10 Byte Count word will be placed into a temporary Buffer register, E B HOLD, where it will reside until the Interface Control Logic can store it in its slot in the RAM FILE. The Status register contains bits that allow both processors to communicate. This is necessary to negotiate transfers. The feature is referred to as "the interprocessor doorbell," and is implemented via the appropriate interrupt logic, with the interrupt control and the Status register.

To summarize, both processors communicate and negotiate the particular operation to take place. Next, the appropriate RAM slots are loaded by the PDP-11 or EBox accordingly and the interface, using detection logic in the data control, starts up the interface via the Interface Control Logic. At this time, the interface contains all the necessary information in order to complete the specified transfer or transfers, including the ability to use either interrupt facility as necessary.

1.10 PROGRAMMING EXAMPLES

Generally, the PDP-11 program sets up the interface by reading status conditions and loading control parameters, and by performing the appropriate sequence of instructions.

The format is:

```
INSTR ADR1, ADR2
```

where ADR1 is the symbolic address of a particular PDP-11 general purpose register (one of eight) within the processor (R0-R7).

ADR2 is the symbolic address of the memory location or device being addressed by this instruction. **For example:**

```
MOV DLYCNT, @ RAM0.
```

In this example, the MOV instruction moves the word in symbolic location DLYCNT into the location addressed by the number in symbolic location RAM0. The @ symbol indicates indirect addressing.

If the contents of RAM0 is the appropriate DTE20 address for the Delay Count word, i.e., XXXX00, then the value in DLYCNT will be moved (loaded) into the DTE20 delay count slot in the RAM (Figure 1-10).

The interprocessor doorbell feature will allow the PDP-11 and KL10 processors to talk to each other via the DTE20 interrupt facilities. To interrupt the EBox, the PDP-11 performs the following:

```
MOV INT10S,@ STATUS; Ring the KL10 doorbell.
```

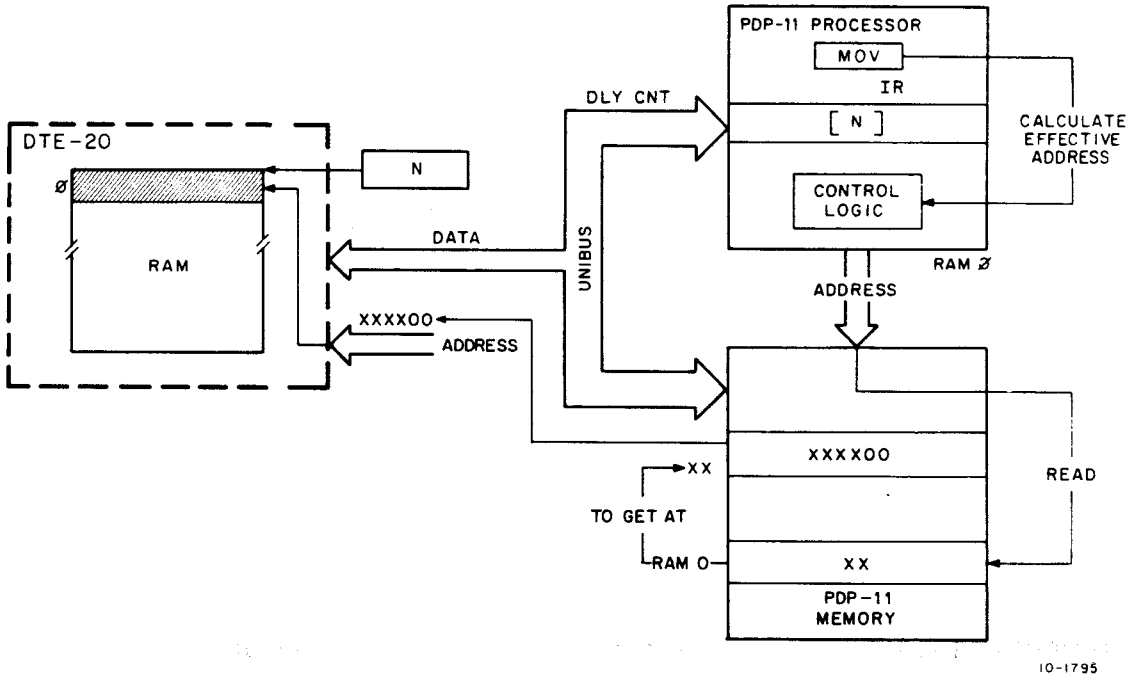


Figure 1-10 Load DLY Count

Assume location Instruction +2 contains a single one in bit position 08; this is necessary in order to set the flag in the DTE20 that causes a programmed interrupt to the EBox. Also, assume location Status contains the address of the Status register in the DTE20, XXXX34. The execution of the MOV instruction by the PDP-11 processor causes the appropriate flag in the interface to set. This action initiates a programmed interrupt to the EBox (Figure 1-11).

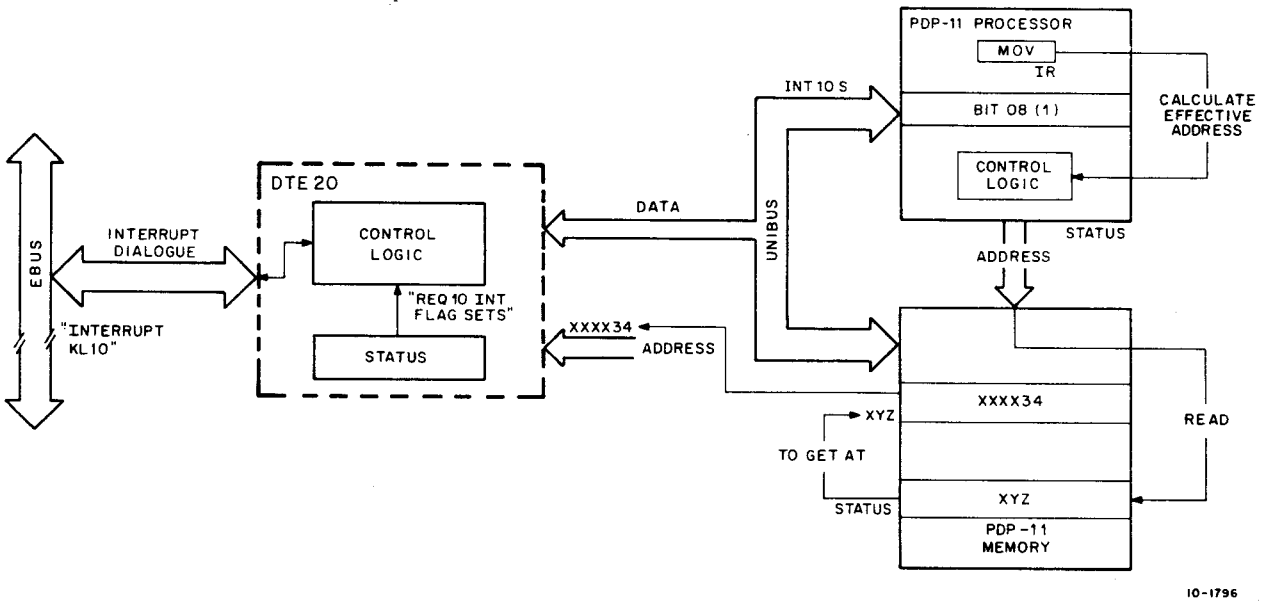


Figure 1-11 Ring KL10 Doorbell

Assume that the DTE20 has completed a series of transfers to the EBox and a flag in the DTE20 (TO11 TRANSFER DONE) sets. This indicates that the current series of data transfers is terminated and could require action from either or both processors. In the event the EBox is interrupted, the following could be the response:

```
CONI DTE20,BITS; Read the Interface Status register
.
. After evaluating the bits the service routine
. clears the interrupt
CONO DTE20,CLTO11; Clear TO11 DONE removing the interrupt
```

Once the priority interrupt logic has arbitrated the priorities and transferred control to the device service routine, this routine must determine what to do about the interrupt. Normally, it reads the Status register and tests the bit pattern, and then makes a determination based upon its findings. It must also turn off the interrupt to enable any other devices on the same channel to use it. Figures 1-12 and 1-13 show the simplified sequence.

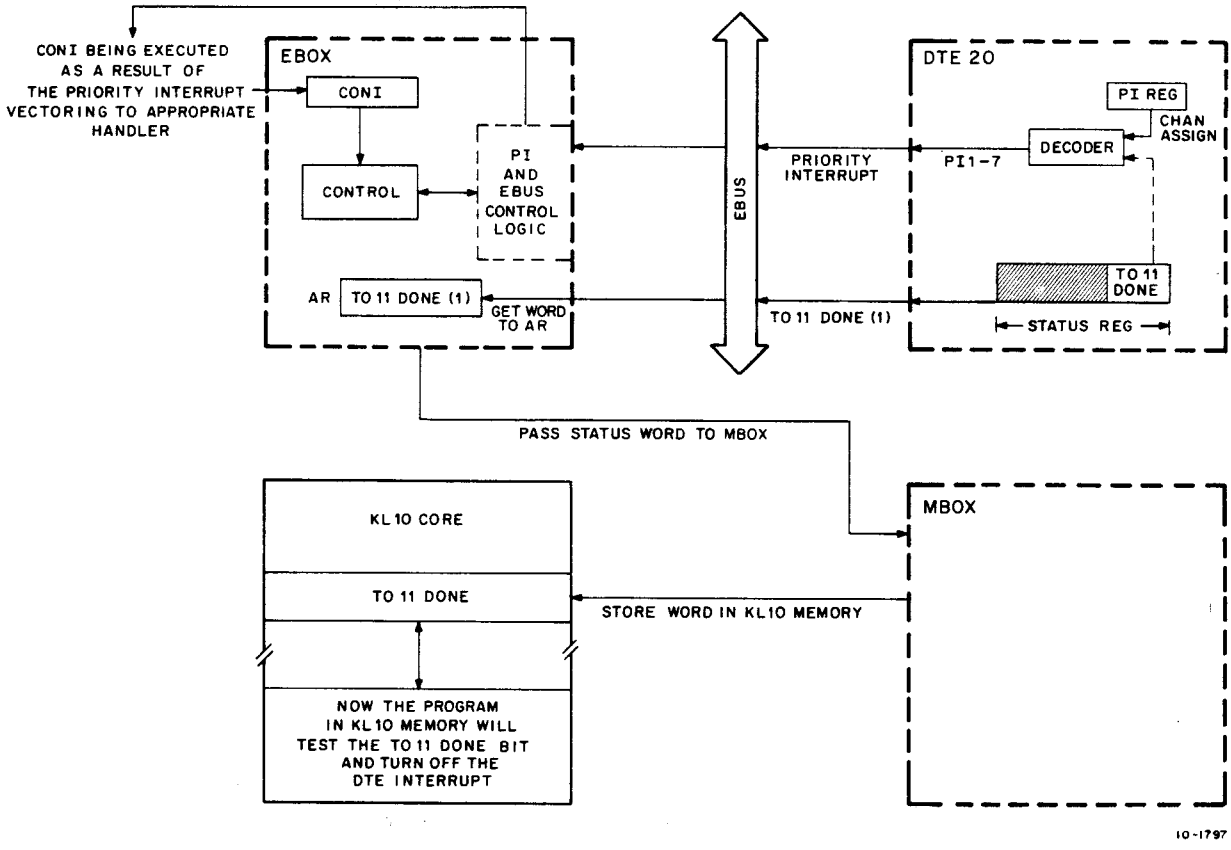


Figure 1-12 CONI Simplified

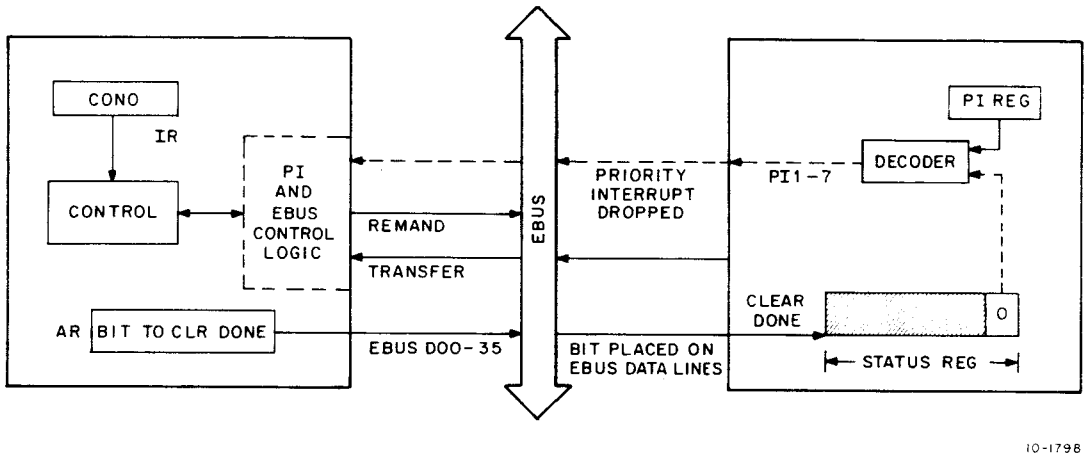


Figure 1-13 CONO Simplified

SECTION 2
FUNCTIONAL DESCRIPTION

2.1 BUS INTERFACING PDP-11 SIDE

Refer to Figures 2-1 and 2-2. The Instruction and Register Decoder PDP-11 side is a functional component. It connects to the Unibus control lines, the interrupt logic, and the NPR logic. It produces the appropriate functional internal register block address and passes data as appropriate during DATO and DATI instructions. Each addressable register has its internal address given in brackets [ ], e.g., Status register [16]. This number is the internal octal address formed from the Unibus address lines A(04:01), a subset of A (17:00). The lines C(1:0) determine the type of instruction being performed, and data passes to or from the internal register block on the data lines D(15:00).

A CROBAR signal is generated by the power controller when the KL10 system is powered up and asserts for approximately 5 seconds. It is passed to the DTE, where it disables the DTE from driving any signals onto the Unibus that might be incorrectly produced as power is asserted or disasserted.

During interface transfer operations, data to be passed between the KL10 and PDP-11 is buffered first in the RAM portion of the Interface Control Logic (Figure 2-2). The data passes through the RAM portion to or from the PDP-11 processor using the NPR facility. In this way, the PDP-11 processor is free to carry on other non-Unibus functions, while DTE20 uses the Unibus for a transfer to or from PDP-11 memory. The Interface Control Logic informs the NPR logic that it would like to make a transfer involving PDP-11 memory. The NPR logic asserts NPR and a dialogue takes place between the PDP-11 and the Instruction and Register Decoder. If the PDP-11 processor or some other device is not using the Unibus, the DTE20 will receive NPG as a response to having asserted NPR. It next asserts SACK and then tests for BBSY SSYN. When this condition is not present from another device, the interface asserts (BBSY). The Address and Byte Count (ABC) register, one of the workinq registers, at this time holds the PDP-11 address and places it on the Unibus address lines. Normally, the address is held in one of the RAM locations until it is needed for a transfer.

The two Status registers are a link between the two processors. All Status register bits are separate flip-flops except for the interprocessor doorbell. Both Status registers can be addressed by either the KL10 or the PDP-11 processor. By setting the TO10 doorbell or TO11 doorbell bit in the Status register, a handshaking will take place between the two processors using the interrupt dialogue. Thus, this mechanism, referred to as the "interprocessor doorbell feature" is a useful medium for communications between processors and transfers may be negotiated.

To the EBox, the DTE20 appears in most respects as a KL10 peripheral. The EBox can perform conditions out (CONO) or conditions in (CONI) instructions, which can set, clear, or read status information from within the interface. The internal register block is essentially transparent to the EBox, with two notable exceptions. Both a Status register (as mentioned previously) and the E B HOLD register can be accessed by the EBox. Data is passed between the DTE20 and the EBox using the EBus data lines D(00:35). Device addressing is handled in a slightly different manner than was the case with the PDP-11 processor, because the EBus has a different structure than the Unibus. Each DTE20 in the system has a discrete physical number assigned to it.

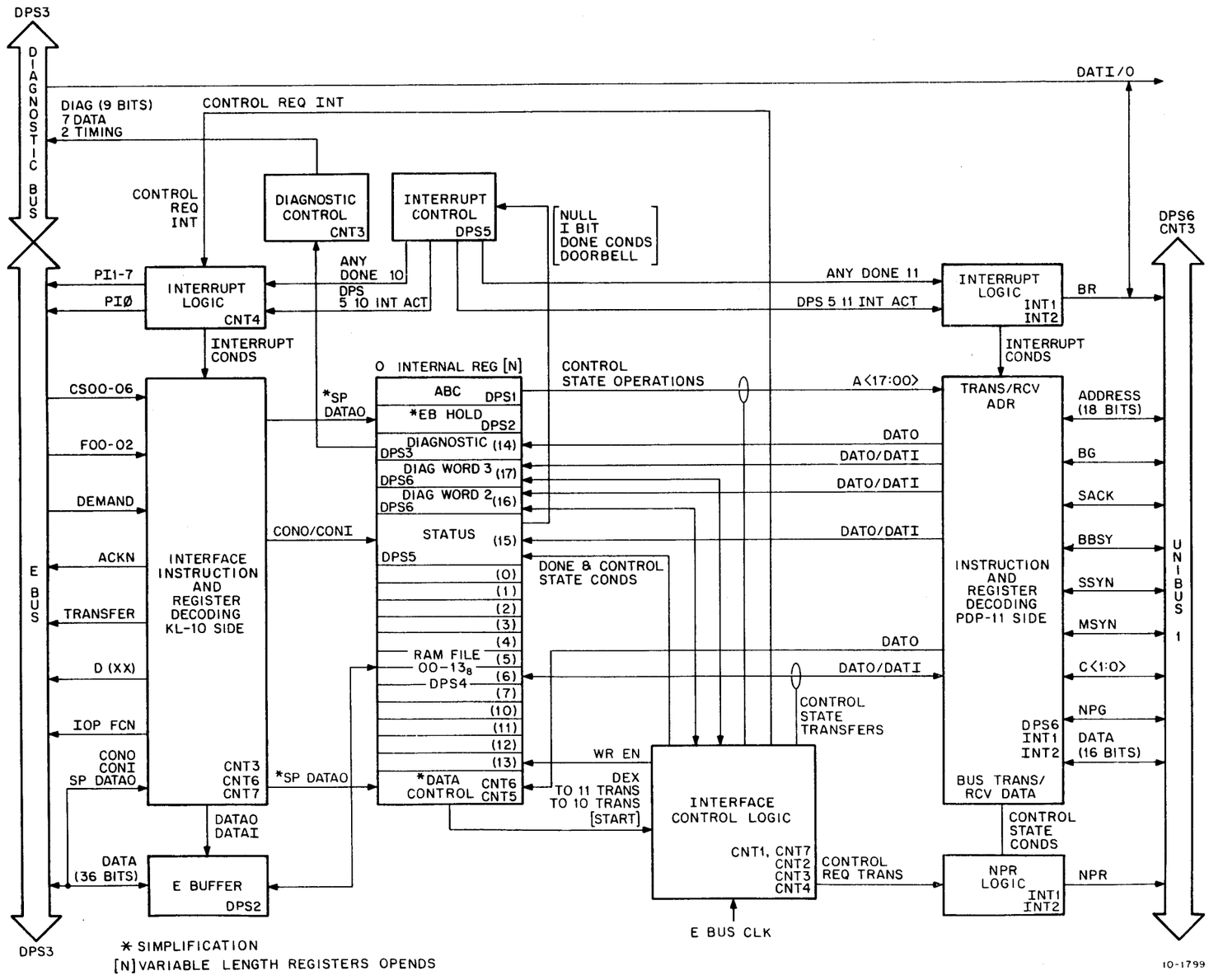


Figure 2-1 DTE Simplified Functional Block Diagram

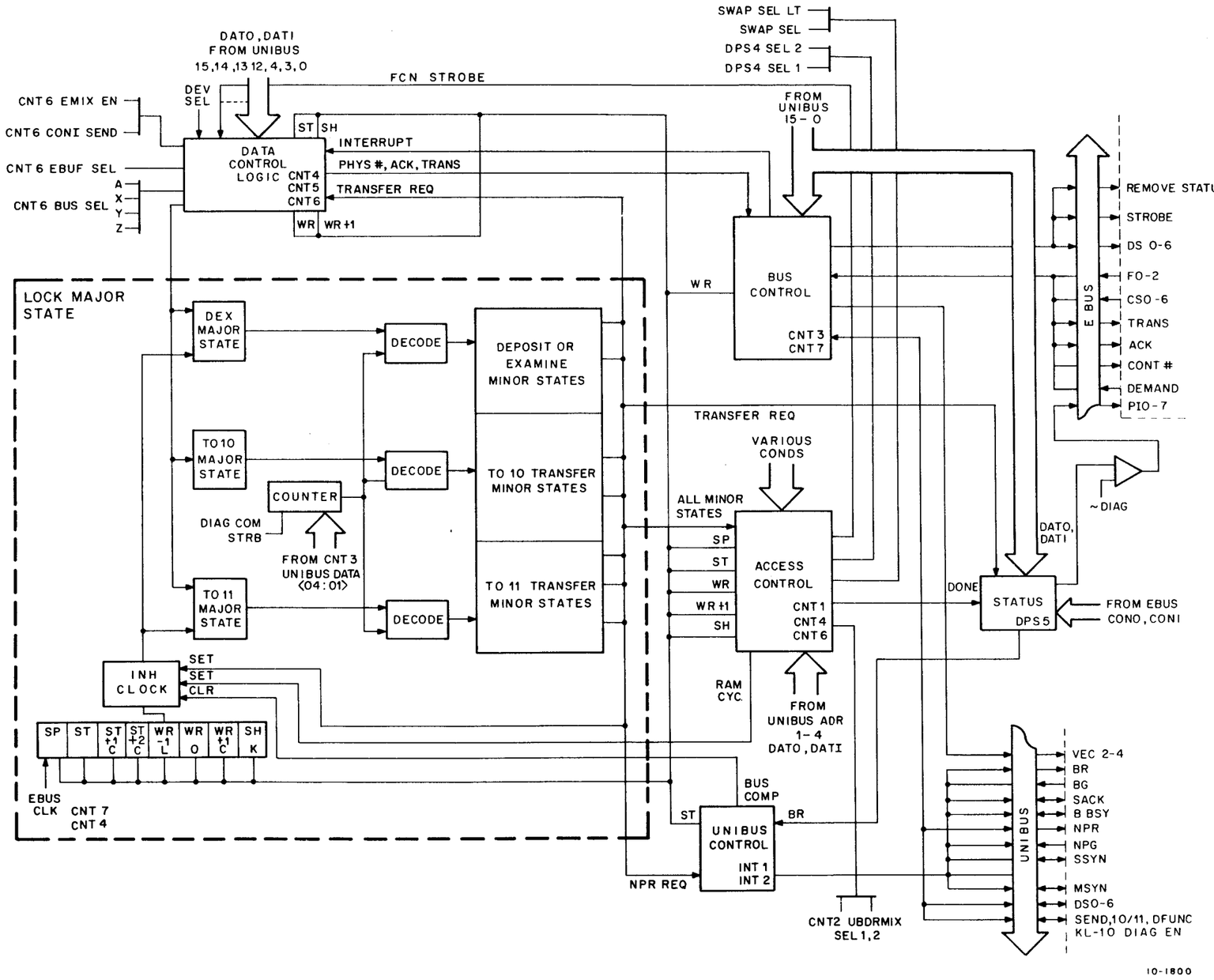


Figure 2-2 Simplified Control Block Diagram

The Controller Select (CS) lines are hard-wired into the logic of each interface. Thus, the EBox must assert the CS lines in a particular configuration and cause the interface to look at these lines (DEMAND) before a transaction can take place between the interface and the EBox. In addition, the type of instruction is given by the function lines F(00:02), while the Controller Select is valid.

2.2 BUS INTERFACING EBOX SIDE

Refer to Figures 2-1 and 2-2. The EBus connects functionally to the interface Instruction and Register Decoder. The decoder is logically driven from one of two sources.

Interrupt Logic
EBox (for instructions).

During interrupts to the EBox, the interrupt logic asserts PI0-7 as the active interrupt line and waits for some response from the EBox. When the EBox is ready to select those controllers and devices on a particular channel for arbitration, it asserts the Controller Select lines CS(04:06) as the channel number (priority interrupt channel number), function lines F(00:02) as PI level serviced, and then asserts DEMAND.

All controllers and devices interrupting on the interrogated channel place their physical controller or device number on the EBus data lines in a preassigned bit position (Dxx). Once again, the DTE20 waits for a response from the EBox. The physical controller numbers are given priority according to the weight of the bit position on the EBus. Physical number 0 has higher priority than physical number 1 and so forth.

In the EBox, the priority interrupt logic arbitrates the incoming physical number, asserts CS(00:03) the physical controller or device to be honored (this is one unique device), CS(04:06) channel being served, and F(00:02) PI address in, and, finally, asserts DEMAND. When the DTE20 is the selected device, it places a special function word on the EBus data lines that identifies the type of service required from the EBox. This function is called the API FCN. The interface asserts ACKNOWLEDGE and TRANSFER to the EBox. Some time later, the EBox drops DEMAND, which detaches the decoder. The EBox responds to the API FCN by executing a DATAO or DATAI instruction in microcode and data passes over the EBus data lines through the E-Buffer and eventually to or from the internal register block.

2.3 INSTRUCTION AND DATA TRANSFER IMPLEMENTATION OVERVIEW-EBOX SIDE

Data transfers over the EBus are implemented in a similar manner. Assume the EBox wishes to place a word of data into the EB HOLD register. The following dialogue occurs.

The EBox places the device select code on CS(00:06). Four possible DTE20 device codes: 200, 204, 210, and 214, are used for the first through the fourth interfaces, respectively. The EBox places DATAO on F(00:02), places the data on the EBus data lines D(00:35), and asserts DEMAND. The interface Instruction and Register Decoder implements the DATAO function and then asserts ACKNOWLEDGE and TRANSFER. At this time, the data is in the EB HOLD register. This completes the DATAO instruction.

By using a variation of the same procedure for a CONO instruction, a bit is set in the Status register that causes an interrupt to the PDP-11 processor. Refer to Figures 2-1 and 2-2. The dialogue between the DTE20 and the EBox takes place as with the DATAO instruction. The difference is that the function lines specify CONO in F(00:02). The EBox places a bit in bit position 23 on the EBus and asserts DEMAND. For detailed bit assignments, see Tables 1-12, 1-13, and 1-14.

The interface Instruction and Register Decoder takes the bit from the bus and sends it directly to the Status register flip-flop. Setting the appropriate flip-flop (DPS5 10 REQUEST INT, in this case) constitutes the "doorbell." The transition of this flip-flop to a 1 is passed into the interrupt control, where it raises the level DPS5 11 INT. This starts the interrupt logic on the PDP-11 side of the DTE20 and BR is asserted over the Unibus control lines. The interrupt condition is passed to the Instruction and Register Decoder, which carries on the appropriate dialogue with the PDP-11 processor. Upon receiving BR from the interface, the PDP-11 asserts BG, granting the bus. SACK is asserted and the DTE20 must test for (BBSY, SSYN) to ensure that no other device is still using the Unibus. As soon as the Unibus is free, BBSY is asserted and the vector address, as well as the interrupt line, is placed on the Unibus. Finally, the PDP-11 asserts SSYN, which terminates the bus dialogue.

In a similar manner, CONI reads the Status register. Once again, the dialogue takes place between the DTE20 and the EBox. The function code for CONI is transmitted via F(00:02), together with CS(00:06), followed by DEMAND. The selected Status bits are selected and passed through the Instruction and Register Decoder to the EBus. When the DEMAND level is removed, the decoder is decoupled and the operation is complete.

The E-Buffer is quite often being used and, therefore, cannot be guaranteed to be free when the EBox is loading the TO10 byte count into the DTE20. The E B HOLD logic was implemented to help enable the byte count loading. The EBox performs the DATAO to the DTE for this unique operation. Once the word (TO10 byte count) is stored in the E B HOLD register, the DATAO can be removed from the EBus lines and the normal sequencing can continue, as the E-Buffer is not busy with the byte count. A flag in the Data Control register indicates that the Byte Count word was placed in the E B HOLD register.

No provision is provided in the DTE20 to allow the EBox to perform DATAI instructions. Thus, the EBox responds with a function code of DATAI only in response to the reception of an appropriate API FCN from the DTE20. If a DATAI DTE is performed, 0 is returned.

2.4 INSTRUCTION IMPLEMENTATION PDP-11 SIDE

By executing DATO or DATI instructions from the PDP-11 processor, and giving the appropriate address, all of the previously mentioned available registers in the register block can be accessed. Assume that it is desired for the PDP-11 to ring the EBox doorbell. To accomplish this, the Status register must be addressed via the Unibus address lines A(17:00) as logical register (16) in bits A(04:01). Also, the necessary control lines must be asserted. These include:

BBSY – Assume control of bus
C(01:00) – Encoded DATO
MSYN – Tell device address and data if any are present.

Bit 08 must be placed on the Unibus data lines D(15:00) at the time the control lines are active.

The Instruction and Register Decoder passes bit 08 into the register block to the Status register, where a flip-flop named DPS5 REQUEST 10 INT will be set, and generates a level into the interrupt control. Any of the RAM locations can be read in the same manner by executing a DATI instruction from the PDP-11, along with the appropriate RAM address. For example, to read RAM location (13), the following lines are activated:

A(17:00) – The address of the desired RAM location
BBSY – Assume control of the bus
C(01:00) – Encoded DATI
MSYN – Tell device address is present.

In response to receiving these control signals, the Instruction and Register Decoder accesses the specified RAM location (13) and places the contents of the Unibus data lines D(15:00) and generates SSYN to terminate the operation. Some time later the PDP-11 detaches from the bus.

2.5 FUNCTIONAL OPERATIONS OVERVIEW

This section describes the four basic DTE20 hardware operations:

1. Diagnostic Operations
2. Deposit/Examine
3. TO11 Transfer/TO10 Transfer
4. Doorbell.

The description of each operation includes a flowchart of the various steps and a simplified functional block diagram similar to Figure 2-1. However, only those sections germane to the operation being described are included in the description.

2.5.1 Deposit Overview

Each of the four DTE20 operations has a preliminary phase of operation. This is the loading of specific control information, or in some cases data, into the appropriate RAM locations. It is always done before the interface begins any operations. For the Deposit operation, the following words are loaded into the indicated locations in the RAM by the PDP-11 processor:

RAM Address	Word Loaded
1	Data Word 3
2	Data Word 2
3	Data Word 1
4	Address Word 1
5	Address Word 2

For the exact format of these words, refer to Table 1-9. The purpose of the Deposit operation is to take information previously loaded into locations in the DTE RAM, in the form of three 16-bit words, and place them in a desired KL10 memory location. This address is in the form of the two words loaded in the interface. Address Word 1 contains a mode bit (12) that determines whether the operation is Deposit or Examine. If bit 12 is set, the operation is a Deposit and if clear, the operation is Examine. Bits 15-13 and bit 11 control the context of the KL10 address. For a privileged front end, the protection bit (bit 11) can be set by the software to perform an unprotected Deposit. For unprotected Deposits, the address space field (bits 15-13) specifies the type of address. Currently, three types of space may be specified as follows:

Bits 15-13	Space
0	Executive Process Table
1	Executive Virtual Address Space
4	Physical Address Space

Note: See API Function Word.

All other codes are reserved for future use by the hardware.

Address Word 2 contains, in bits 15 through 00, bits 20 through 35 of the KL10 physical address. The data word is a 36-bit word composed of three words loaded in the interface. Data Word 1 contains bits 00 through 03 of the KL10 data word in PDP-11 bits 03 through 00; the bits are right justified. Data Word 2 contains bits 04 through 19 of the KL10 data word in PDP-11 bits 15 through 00 and Data Word 3 contains bits 20 through 35 of the KL10 data word in PDP-11 bits 15 through 00. Loading the data words has no direct effect on the Interface Control Logic in terms of starting the operation. When Address Word 2 is loaded, the operation begins.

Refer to Figure 2-3. Assume that the five words have been previously loaded into the interface. The Data Control register senses the loading of the key word. When Address Word 1 was loaded, the mode bit set a flag in the Data Control register. It is this flag that causes the interface to perform a Deposit, instead of an Examine, operation. When Address Word 2 is loaded, the interface begins the Deposit operation and, therefore, all other locations relating to the Deposit must be loaded prior to loading Address Word 2. Refer to the flow for Deposit. The significant events that occur at various points in the flow are listed on the flow (right side) to highlight the event taking place. All operations that occur in the interface are synchronous with a series of progressive time states. These are listed to the left of the event (or in some cases the events) that occur during that particular state.

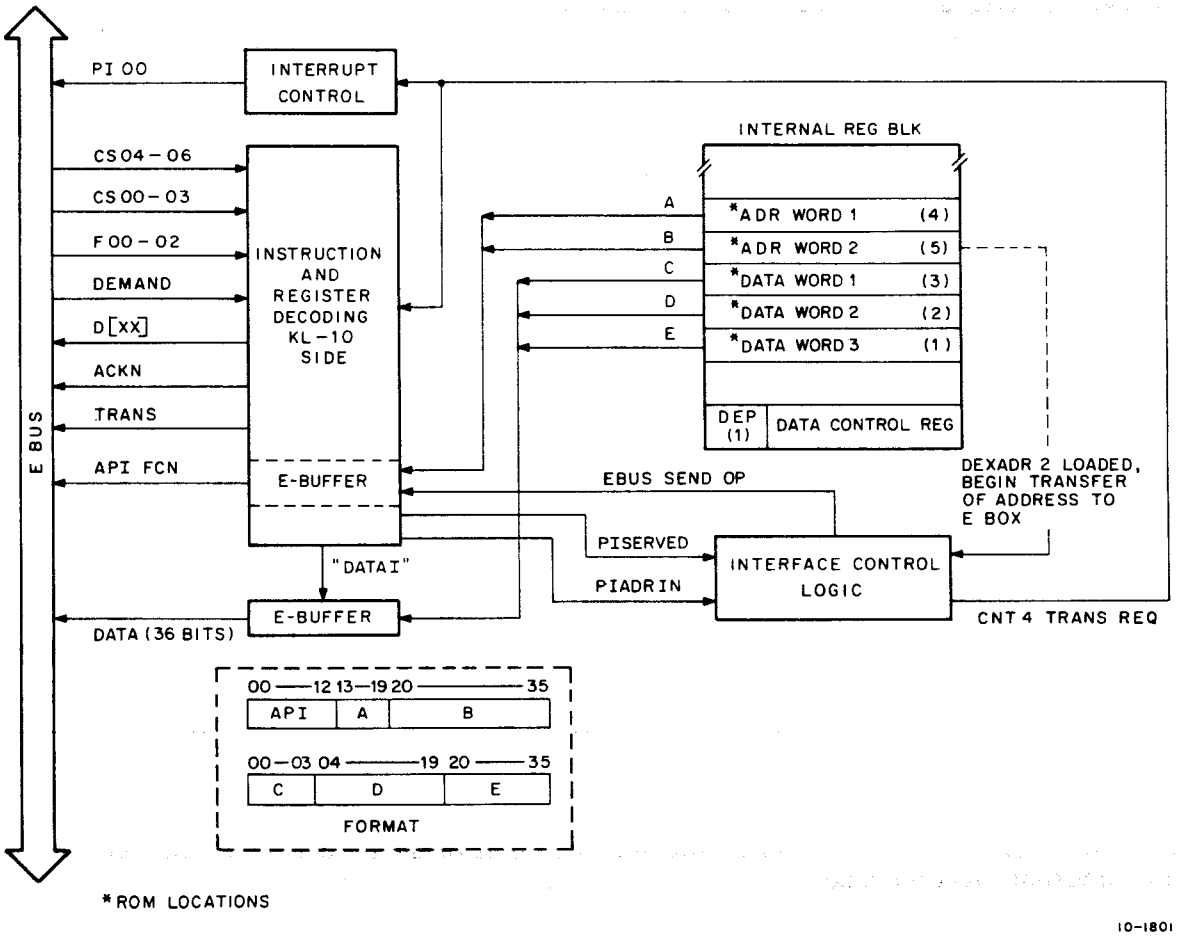


Figure 2-3 Deposit Simplified Functional Block Diagram

2.5.1.1 Deposit Operation – The format of the words used for the Deposit operation are illustrated on Figure 2-5, Deposit and Examine Words; in addition, a functional flow (Figure 2-6, Simplified Flow Deposit) is included. These should be referenced while reading the functional description.

The first part of this operation extracts the 22-bit KL10 address from two RAM locations, ADR Word 1 and ADR Word 2, and places the adjusted address in the E-Buffer (Figure 2-4).

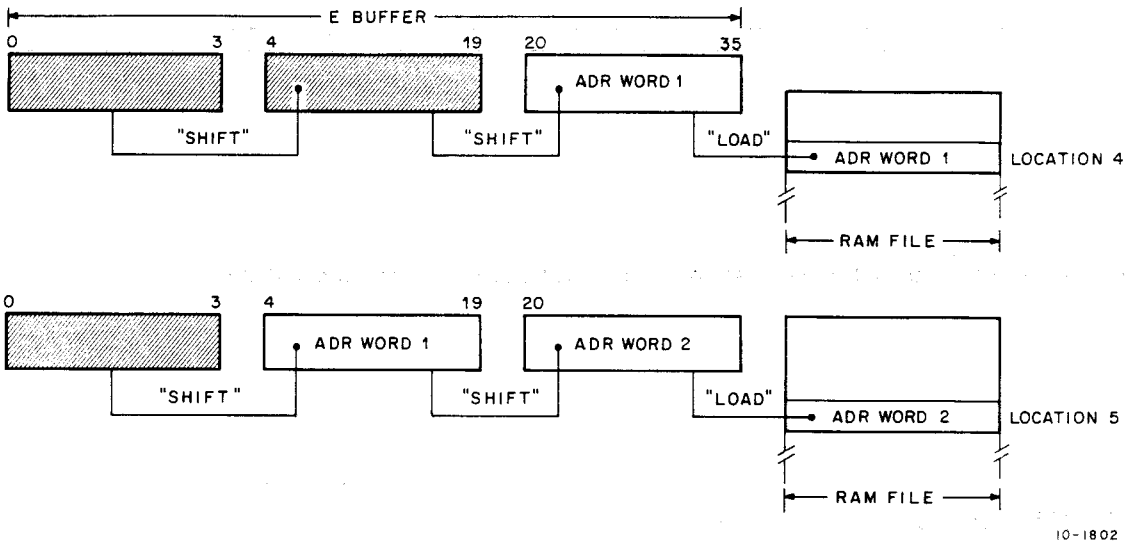


Figure 2-4 Address Word Setup

This is shown on Figure 2-3. The contents of RAM location (4) are read into the E-Buffer (shown in dotted lines). At this time, only what corresponds to bits 14-19 is loaded into the E-Buffer. The event occurs during "CNT 4 DEX ADR 1" time. Next, the contents of RAM location (5) are loaded into the position currently occupied by ADR Word 1. Simultaneously, that portion of the E-Buffer is shifted left. The flow indicates the final position of the address in the E-Buffer. Bits 0 through 03 contain zeros and bits 14 through 35 contain the KL10 physical address. To get this address to the EBox, it is necessary to use the interrupt facility in the interface. The Interface Control Logic generates a CNT 4 TRANS REQ into the interrupt control, which triggers the interrupt dialogue with EBus PI00.

The Interface Control Logic holds the interface in the "CNT 4 DEX ADR 2" state until the dialogue is at a point where the API FCN word has been taken by the EBox.

Refer to Figure 2-7. When the PI module detects that the interrupt request is true, it sends out a function that polls all devices on the highest interrupting channel to send the physical numbers. On the first interrupt, the PI module detects the highest level interrupt and starts servicing it. Once it determines which devices are interrupting on that channel, it decides which device should be served and sends out a second function: for a given physical controller on a given PI level, send its interrupt function code. The PI module then interrupts the main part of the processor and the processor executes the interrupt function code.

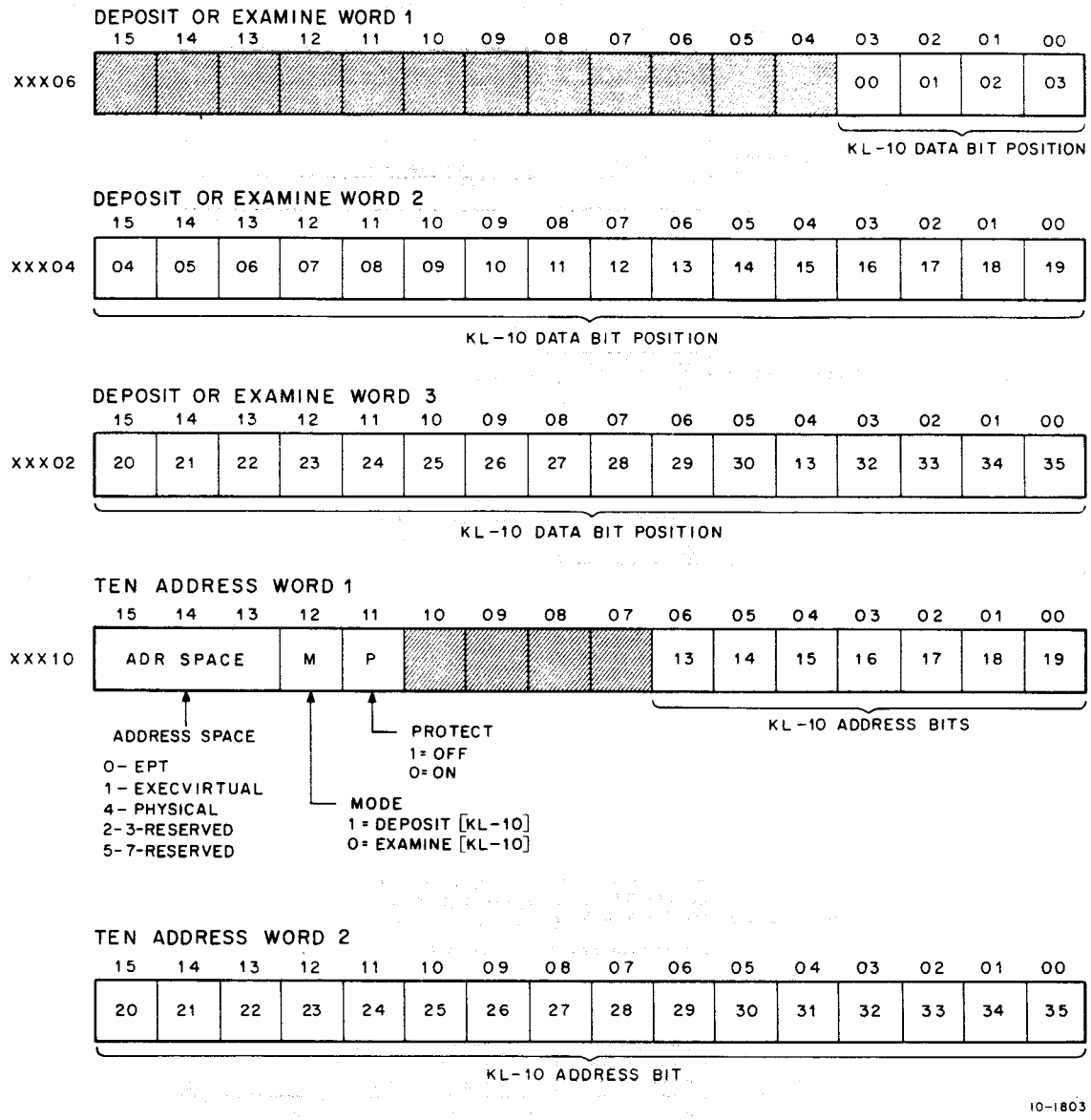


Figure 2-5 Deposit and Examine Words

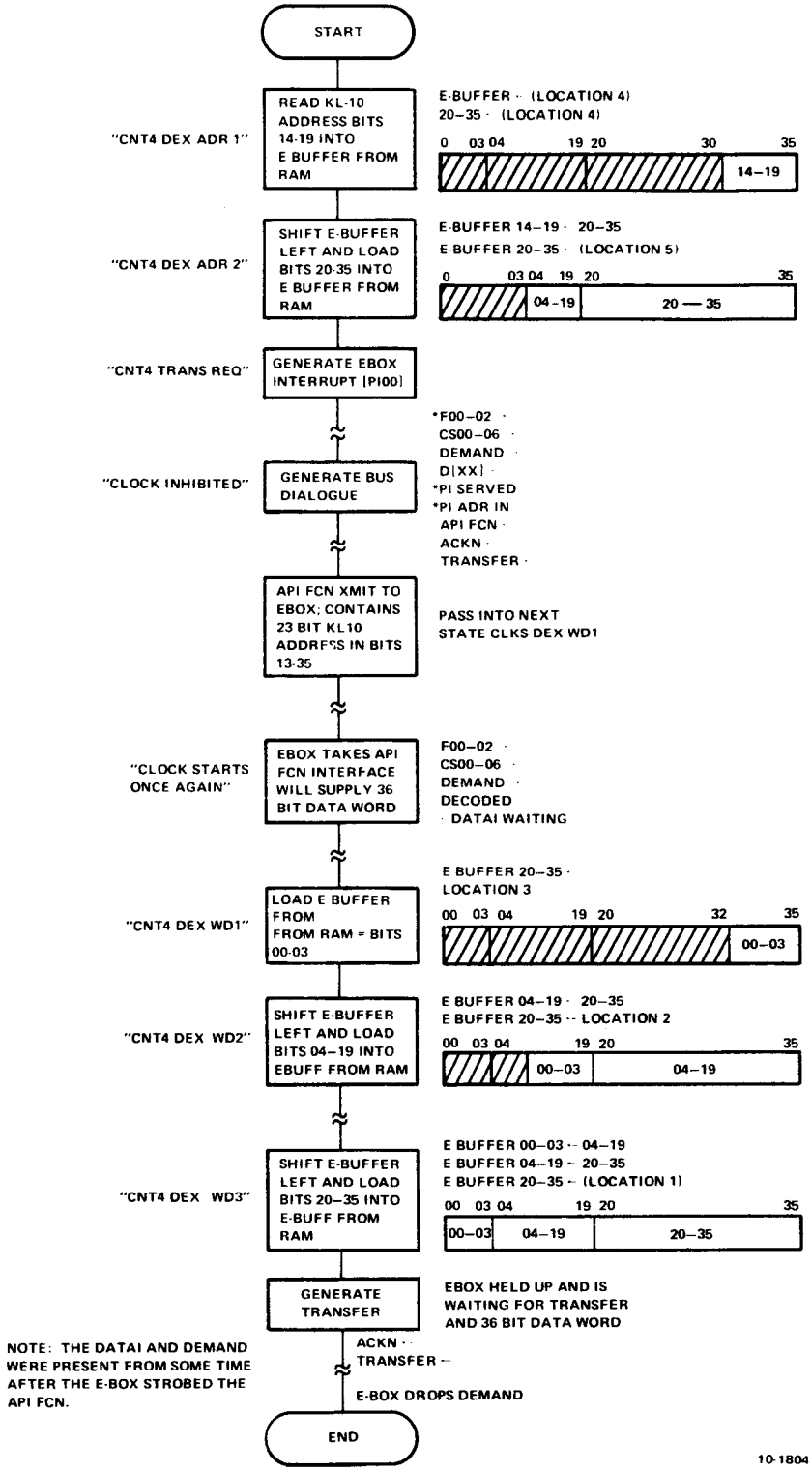


Figure 2-6 Simplified Flow Diagram

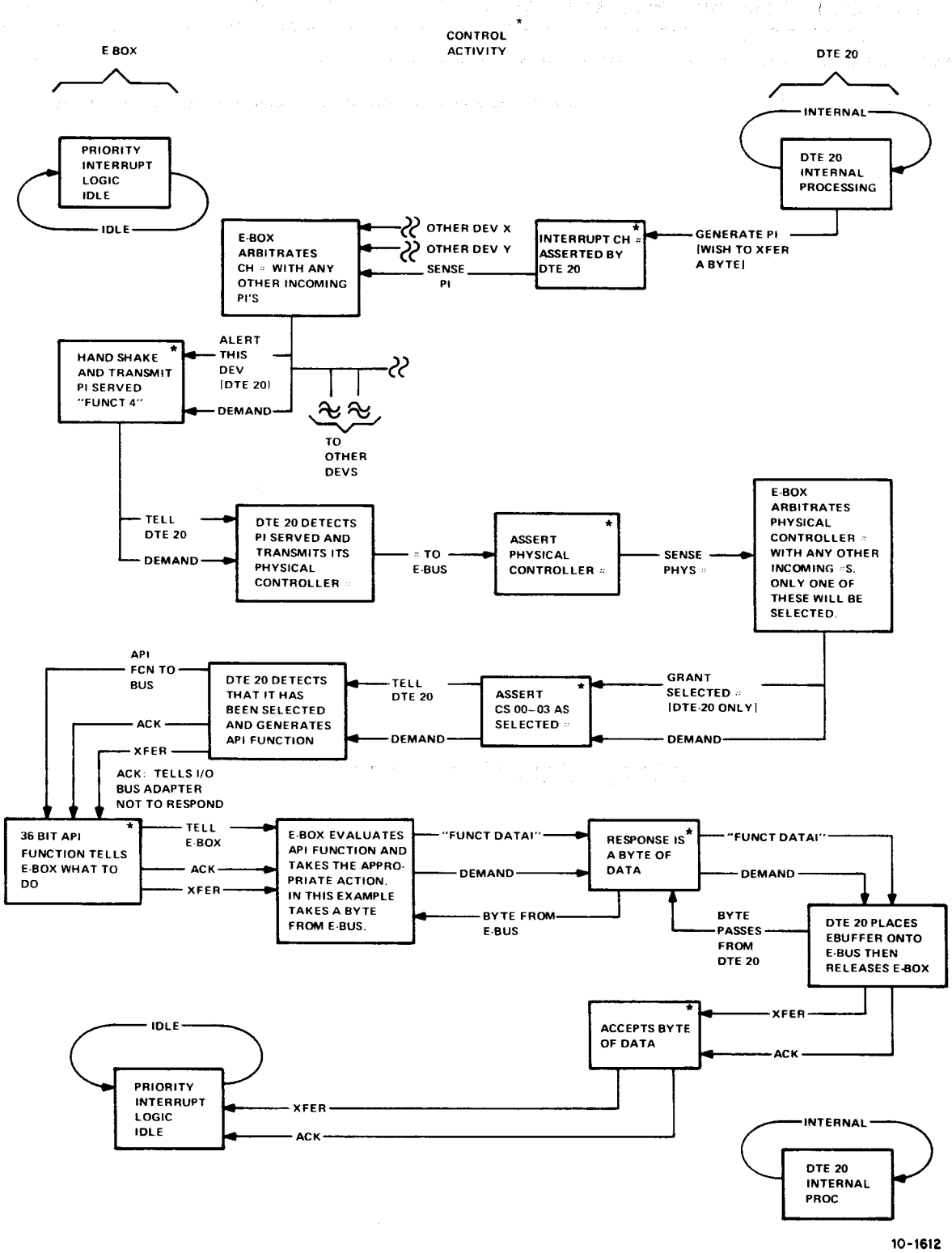


Figure 2-7 Interrupt Dialogue Overview

Upon detecting DEMAND for the second time with PI ADR IN, the DTE20 places the API FCN on the EBus data lines D(00:35) and asserts ACKNOWLEDGE and TRANSFER to the EBox. The API FCN contains the information indicated on Figure 1-1. The function code in bits 3-5 specifies the type of service required and bit 6 specifies whether protection and relocation should be put in effect. At this time, two events are happening: the DTE20 is passing into the next state to set up the 36-bit data word in the E-Buffer to pass along to the EBox, and the EBox is preparing to implement a DATAI instruction in the microcode. The EBox asserts F(00:02) as DATAI, CS(00:06) as selecting the DTE20 and asserts DEMAND. The EBox must now wait until the DTE20 generates ACKNOWLEDGE and TRANSFER before it can detach from the EBus. The interface reads the word from RAM location 3 into the E-Buffer, refers to the flow at "CNT 4 DEX WD1", and enters the next state "CNT 4 DEX WD 2". The second part of the data word bits 04-19 is now read from RAM location 2 and loaded into the E-Buffer and simultaneously the E-Buffer is shifted left adjusting the words (Figure 2-8).

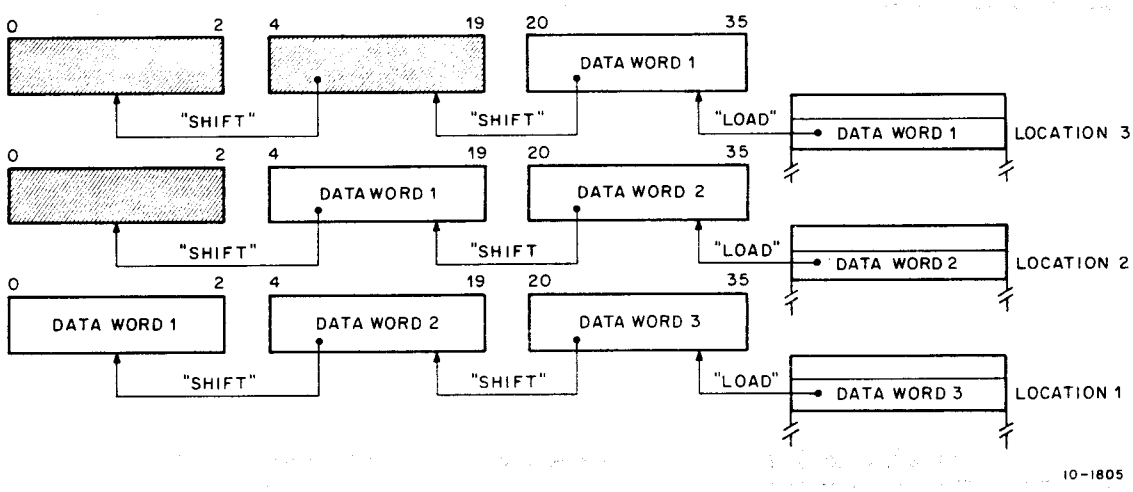


Figure 2-8 Deposit Data Word Setup

Referring to Figures 2-9 through 2-12 and the flow (Figure 2-13), bits 16-19 of the E-Buffer contain bits 00-03 of the previously loaded data word, and bits 20-35 of the E-Buffer contain bits 04-19 of the data word.

The DTE20 now passes into the final state of the Deposit operation "CNT 4 DEX WD 3" and loads the E-Buffer from RAM location 1 while shifting the E-Buffer left once again. At this point, the E-Buffer contains the entire 36-bit data word and this is placed onto the EBus. The DTE20 asserts ACKNOWLEDGE and TRANSFER and some time later the EBox drops DEMAND. For parity computation, see Section 3.

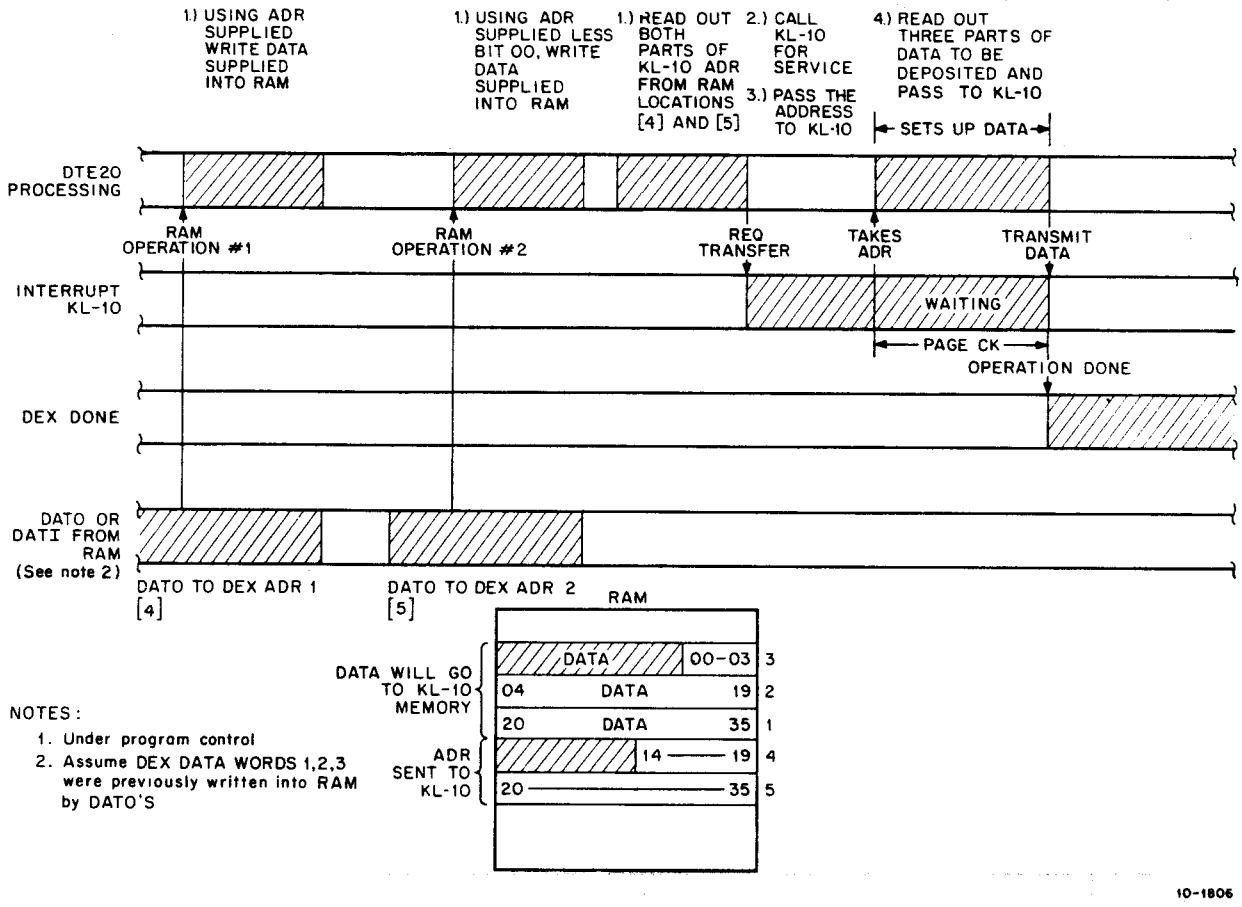


Figure 2-9 Deposit Operation Including Address Setup

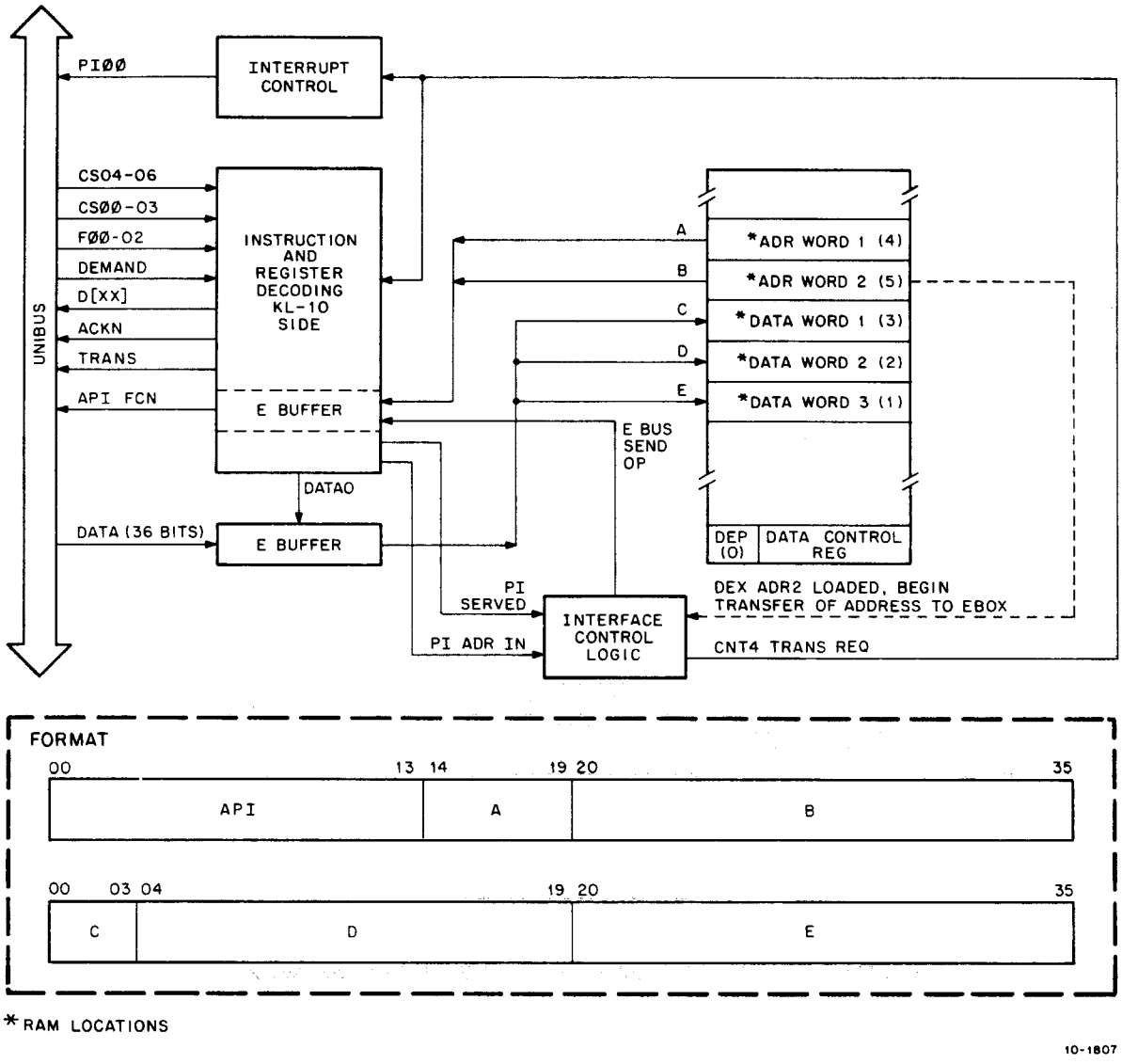


Figure 2-10 Examine Simplified Functional Block Diagram

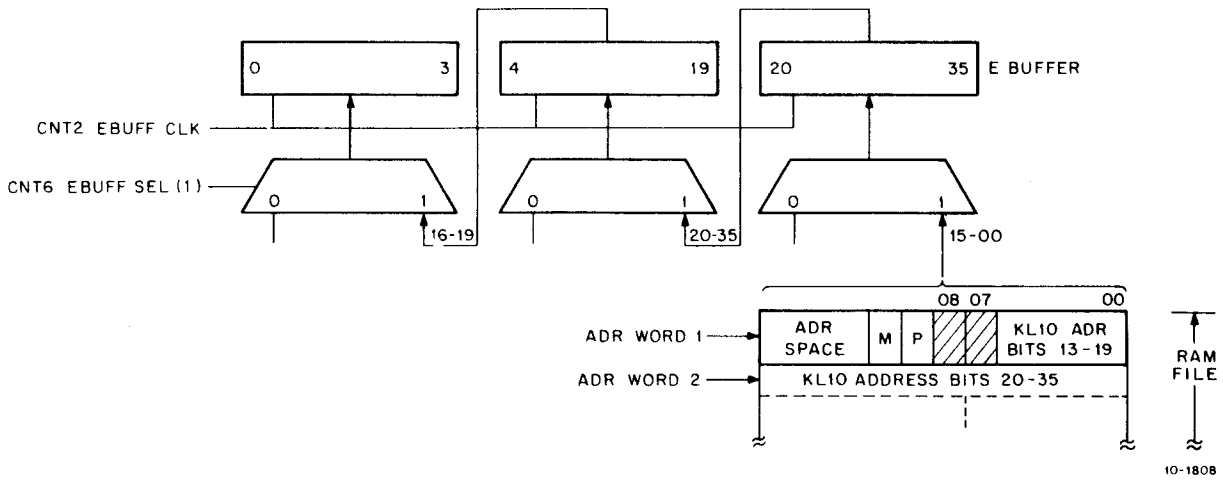


Figure 2-11 Loading the E-Buffer (Examine)

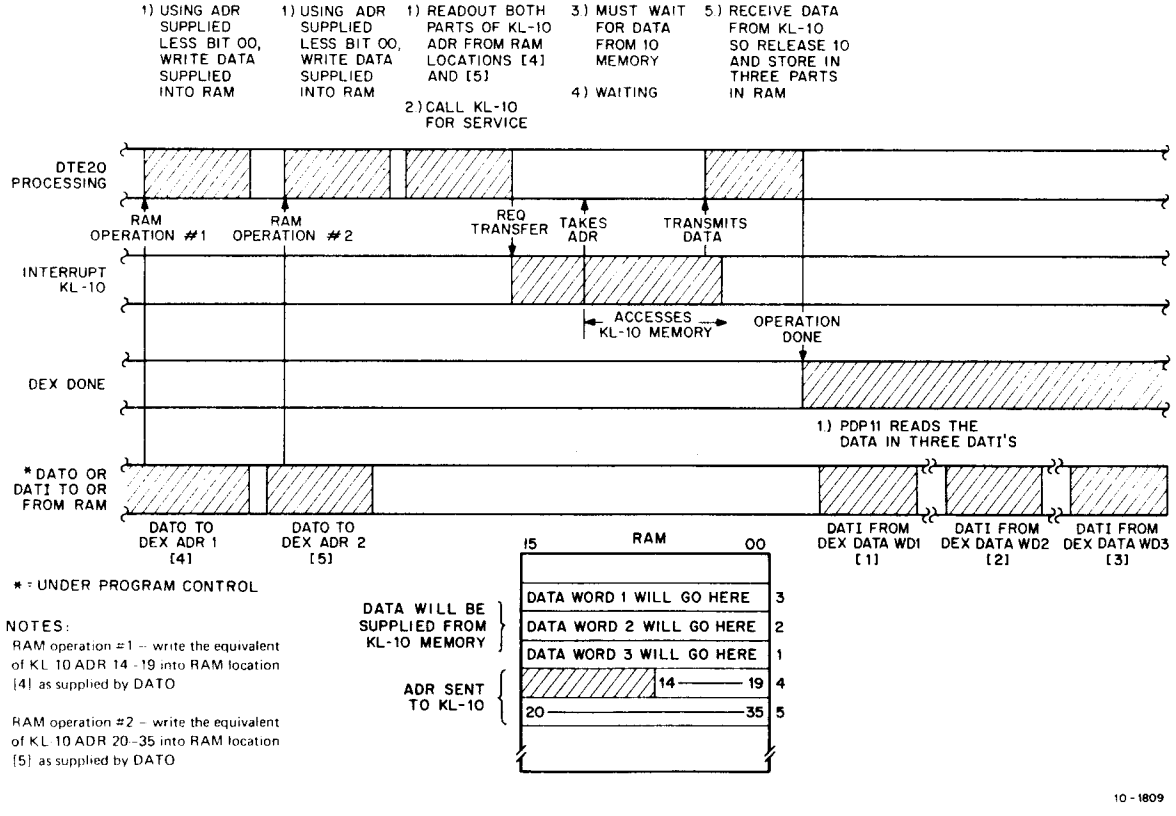


Figure 2-12 Examine Operation Including Address Setup

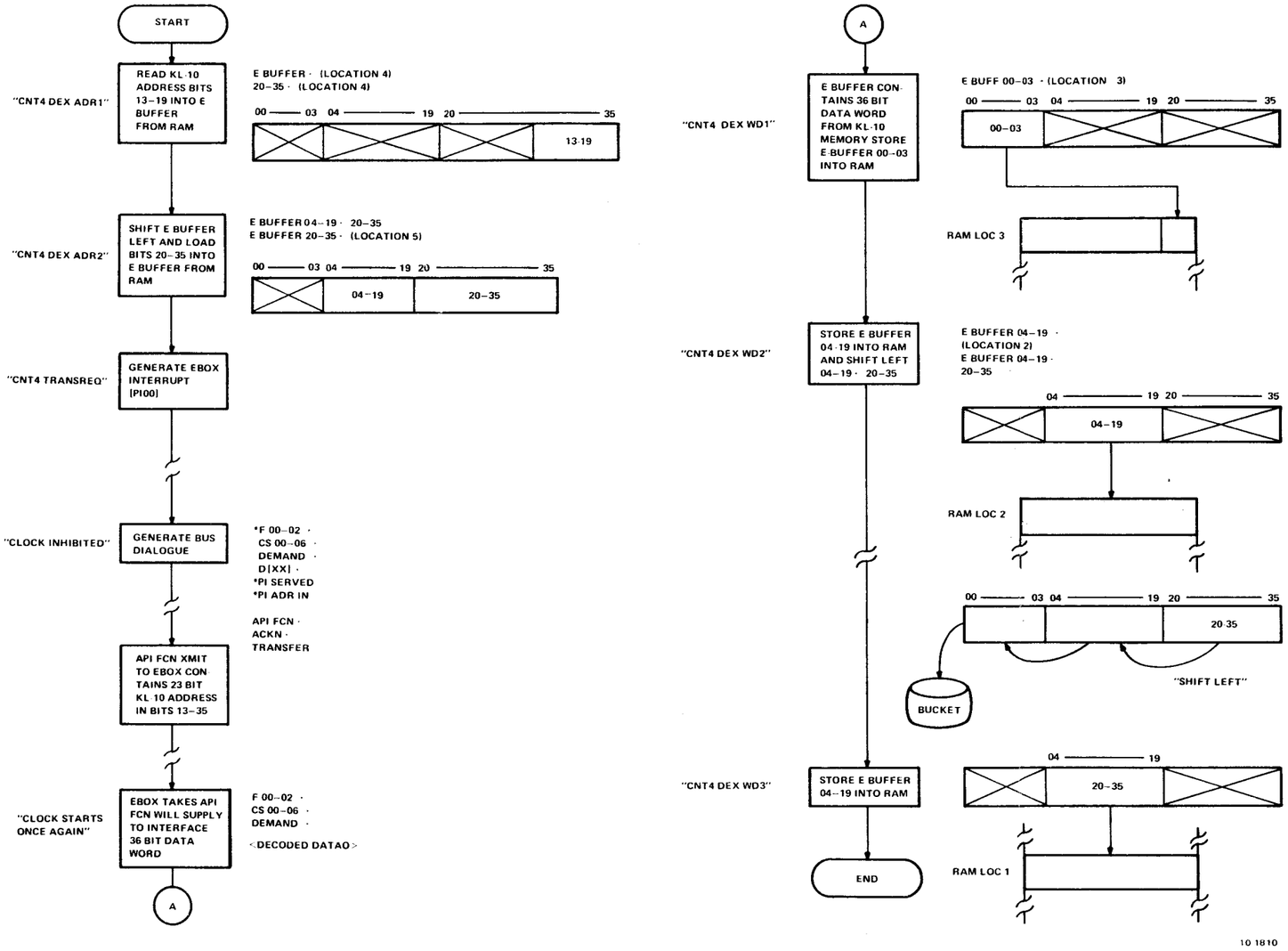


Figure 2-13 Simplified Flow Examine

2.5.2 Examine Overview

The preliminary phase for the Examine operation is as follows. The PDP-11 processor loads the following locations in the RAM:

RAM Address	Word Loaded
4	Address Word 1
5	Address Word 2

For the exact format of these words, see Table 1-9. The purpose of the Examine operation is to take information from KL10 memory in the form of one 36-bit word and place this information, in the form of three 16-bit words, into a specific PDP-11 memory location. The address is as with the Deposit operation, in the form of two words loaded into RAM locations in the interface. Address Word 1 and Address Word 2 are in the same format as for Deposit, the only exception being that the mode bit is not set. This indicates that the operation to be performed is not a Deposit, but rather an Examine. Because the EBox will produce the data word, nothing is loaded into RAM locations 1, 2, and 3 as with the Deposit operation. The interface begins the operation when it detects the loading of Address Word 2.

2.5.2.1 Examine Operation – The format of the words used for the Examine operation is illustrated on Figure 2-12, Examine Operation Including Setup of ADR. In addition, a functional flow Figure 2-13, Simplified Flow Examine, is included. These should be referenced while reading the functional description.

The first part of this operation extracts the 36-bit API word from two RAM locations (ADR Word 1 and ADR Word 2) and places the adjusted word in the E-Buffer.

NOTE
See Figure 1-9 for exact format.

This is shown in Figure 2-11. The contents of RAM location (4) are read into the E-Buffer (shown in dotted lines). At this time only what corresponds to bits 14-19 is loaded into the E-Buffer. The event occurs during "CNT 4 DEX ADR 1" time. Next, the contents of RAM location (5) are loaded into the position currently occupied by ADR Word 1 and simultaneously that portion of the E-Buffer is shifted left. The flow indicates the final position of the address in the E-Buffer. Bits 0 through 13 contain zeros and bits 14 through 35 contain the KL10 virtual address.

To get this address to the EBox, the interface interrupt facility is used. The Interface Control Logic generates a CNT 4 TRANS REQ into the interrupt control, which triggers the interrupt dialogue by asserting PI00. The Interface Control Logic holds the interface in the "CNT 4 DEX ADR 2" state until the dialogue is at a point where the API FCN has been taken by the EBox.

When the bus dialogue begins, the interface asserts PI00 as the interrupting line. In response, the EBox arbitrates the line with any others it samples and responds to the highest. Assuming the DTE20 to be the one, the EBox asserts F(00:02) as PI Served, CS(04:06) as the interrupting channel, and finally asserts DEMAND. The DTE20 compares its channel to the one asserted and finding it the same, asserts (DXX) the physical controller number (for the DTE20). Once again, upon receiving the physical controller numbers from those candidates selected by channel number, the EBox arbitrates among the physical numbers and selects the DTE20 as highest priority (lowest weighted value). The EBox asserts F(00:02) as PI ADR IN, CS(04:06) as the interrupting channel, CS(00:03) as physical controller to be honored, and finally asserts DEMAND.

Upon detecting DEMAND for the second time with PI ADR IN, the DTE20 places the appropriate API FCN on the EBus data lines D(00:35) and asserts ACKNOWLEDGE and TRANSFER to the EBox. The API FCN contains the 23-bit address for the EBox, as well as information telling the EBox which interface is talking to it, and a function code in bits 03-05 that effectively tells the EBox how to respond (i.e., with a DATAO or DATAI). In the case of an Examine operation, the EBox reads a 36-bit word from KL10 memory and executes a DATAO using the microcode. The result is that the 36-bit word is placed on the EBus data lines D(00:35) and loaded into the E-Buffer (Figure 2-14). The DTE20 detects this event and asserts ACKNOWLEDGE and TRANSFER.

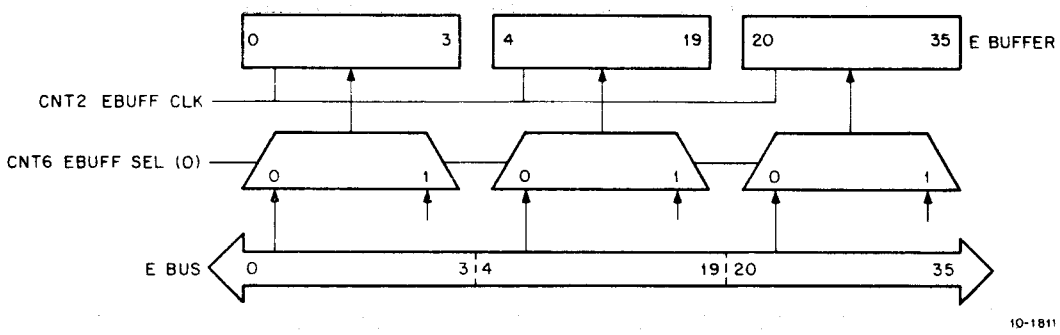


Figure 2-14 E-Buffer Loaded from the EBus

The interface enters the "CNT 4 DEX WD 1" state where the E-Buffer begins to be disassembled and written into the RAM. E-Buffer bits 00-03 are written into RAM location 1 and the next state is entered. During "CNT 4 DEX WD 2", the E-Buffer bits 04-19 are written into RAM location 2 and the E-Buffer is shifted left so that the next portion of the word will be positioned for writing. The last state is entered, "CNT 4 DEX WD 3" and the E-Buffer bits 04-19, which contain bits 20-35 of the Data word, are written into RAM location 3. This terminates the Examine operation.

2.5.3 TO11 Transfer Overview

The preliminary phase for the TO11 transfer is as follows: the PDP-11 processor loads the following locations in the RAM:

RAM Address	Word Loaded
0	DLY Count
7	TO11 Byte Count
11	TO11 PDP-11 Address

Bit 13 of the TO11 Byte Count word controls whether the DTE is in byte mode or word mode. When bit 13 is set, the DTE is in byte mode; when clear, the DTE is in word mode. If the DTE is in byte mode, transfers on the PDP-11 side are 8 bits long, while word mode transfers on the PDP-11 side are 16 bits long. Once bit 13 selects byte or word mode operation, the bit is written into the RAM and into a Control flip-flop. The bit written into the RAM is used only when the Byte Count word is read. Therefore, only the state of the last write operation is seen. This controls the positioning of the incoming byte in the RAM, which acts as a temporary buffer for the byte. The byte is always placed on the EBus in bits 28-35 for TO11 byte mode transfers.

The DTE writes into the first location specified in the TO11 address. Refer to Figures 2-15 through 2-17. The byte in EBus bits 28-35 is stored into bits 07–00 of the RAM on the first transfer. From there, it is passed over the Unibus (on bits 07-00) and is stored in PDP-11 memory. Before each succeeding transfer, both the TO11 address and the byte count are incremented. The byte count holds a negative number equal to the number of bytes to be transferred. Actually the byte count is read, tested (for a field equal to zero), and then incremented.

Again, a byte of data is placed onto the EBus in bits 28-35 by the EBox. This time, however, the byte is stored in bits 15-08 of the RAM. The TO11 address and byte count are incremented and the succeeding byte of data is transferred. The transfers continue until the byte count increments to zero. The byte count test for zero is performed before a data transfer, so an extra cycle is started after the last transfer of a particular transfer sequence. When the byte count is read and tested for zero, and the field is zero, the cycle is aborted. Then, the TO11 Done flag sets, which causes an interrupt to the PDP-11; optionally, the EBox can also be interrupted.

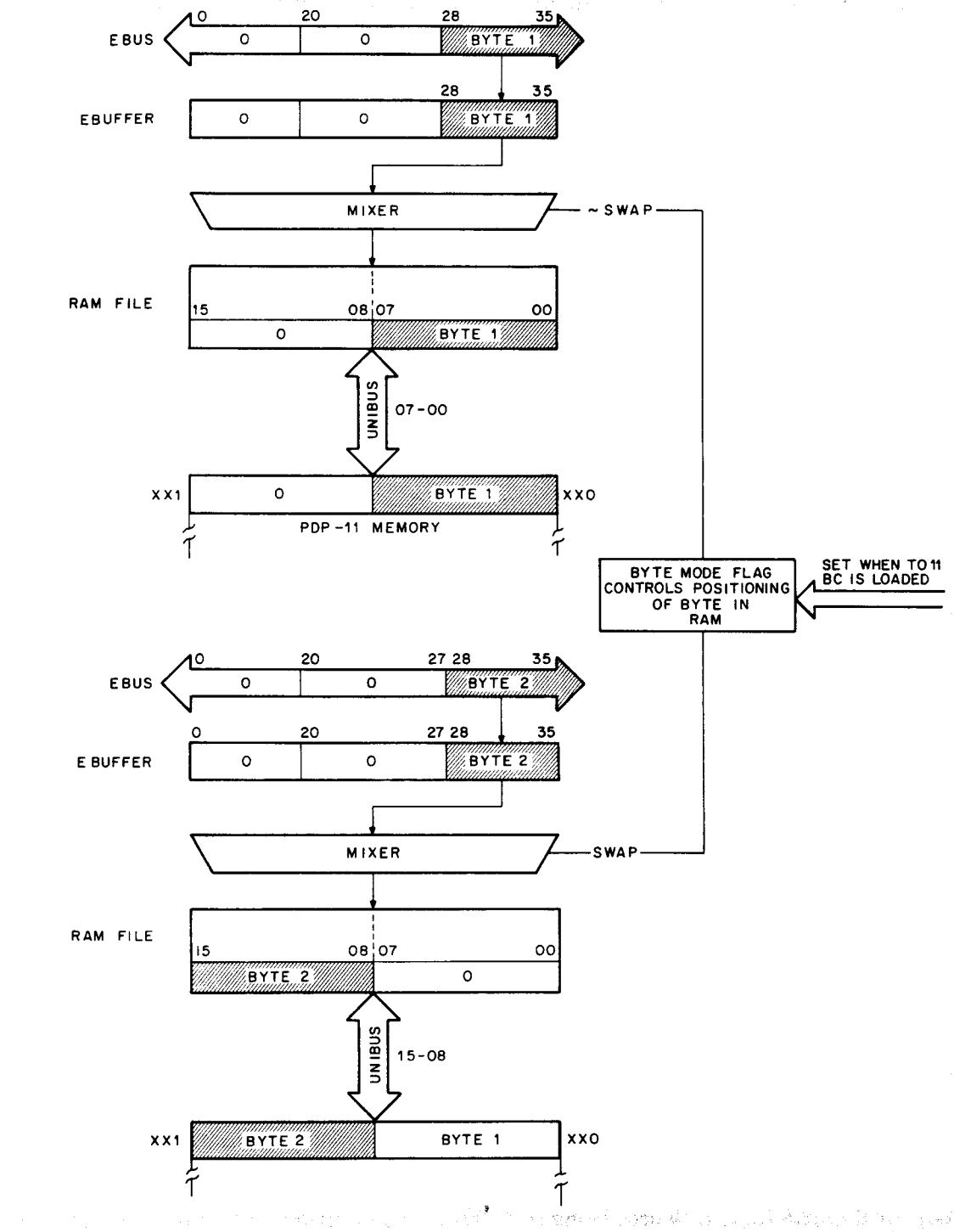
When bit 13 of the Byte Count word is a zero, the DTE is in word mode. Information is placed on the EBus by the EBox in bits 20-35 and is stored in the RAM in bits 15-00. From there, it is passed to PDP-11 memory over Unibus bits 15-00. When in word mode, the DTE writes into consecutive PDP-11 memory words, all of which are even locations. Writing is done in every PDP-11 word, as opposed to every PDP-11 byte (byte mode).

After each transfer, the TO11 address is incremented twice. This updates the address for the next transfer and transfers continue until the byte count reaches zero. Two special provisions are furnished in the Byte Count word: one is to allow transfer termination upon detection of a NULL character during the transfer. This causes an interrupt to the PDP-11 processor, provided that bit 14 of the TO11 Byte Count word is set prior to beginning the transfer. The second provision (I bit; bit 15 of the TO11 Byte Count word) allows the receiver of data (only) the option of being interrupted. So, without reloading all of the parameters, another transfer can be started just by changing the address. The transfer in progress continues from the new address.

In general, transfers are of the single-block type and the I bit is always set. Occasionally, multiple-block transfers (e.g., TO10) are implemented, resulting in more than one transfer. The last transfer, in such a case, always sets the I bit. Therefore, both the KL10 side and PDP-11 side recognize a Done Interrupt and, relative to the software, stay synchronized, believing the transfer is done. If the I bit is set, both the TO11 address and the TO11 Byte Count word must be reloaded. If it is not set, just the TO11 Byte Count word must be reloaded.

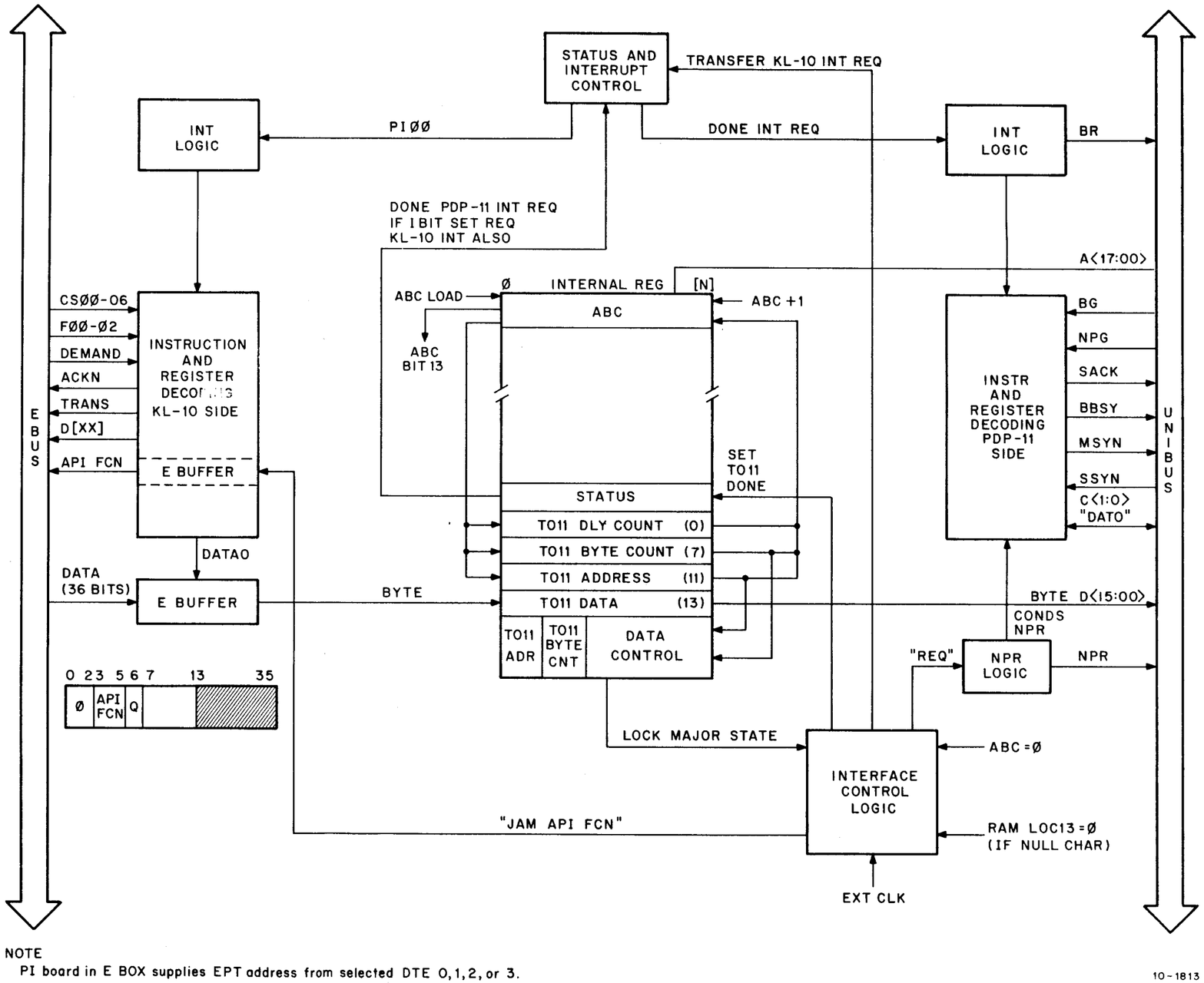
The high order PDP-11 address bits (16-17) are controlled by the delay count. Therefore, the high order bits of the Delay Count word are bits 16-17 of the address space. All transfers are constrained to being in the same 32K space, but they can operate in a greater than 32K core machine; hence, the 18 bits worth of addressing.

Similar to the TO11 Byte Count word, the TO11 Delay Count word holds a negative number. The DTE increments the word (toward zero) prior to any transfer of data. When bit 13 of the Delay Count word equals zero, the transfer begins. The net effect is to force the DTE to pause before starting each transfer, eliminating bursts of interrupts. During the transfers, RAM location (13) is used as a temporary buffer for the data (byte or word) being transferred. The transfer from RAM location (13) is via the NPR facility to PDP-11 memory.



NOTE: Initial TO11 Address = XX1
DTE utilizes bit 13 of the TO11 byte count word, as a byte mode flag.

Figure 2-15 TO11 Byte Mode



NOTE
PI board in E BOX supplies EPT address from selected DTE 0, 1, 2, or 3.

10-1813

Figure 2-16 TO11 Transfer Simplified Functional Block Diagram

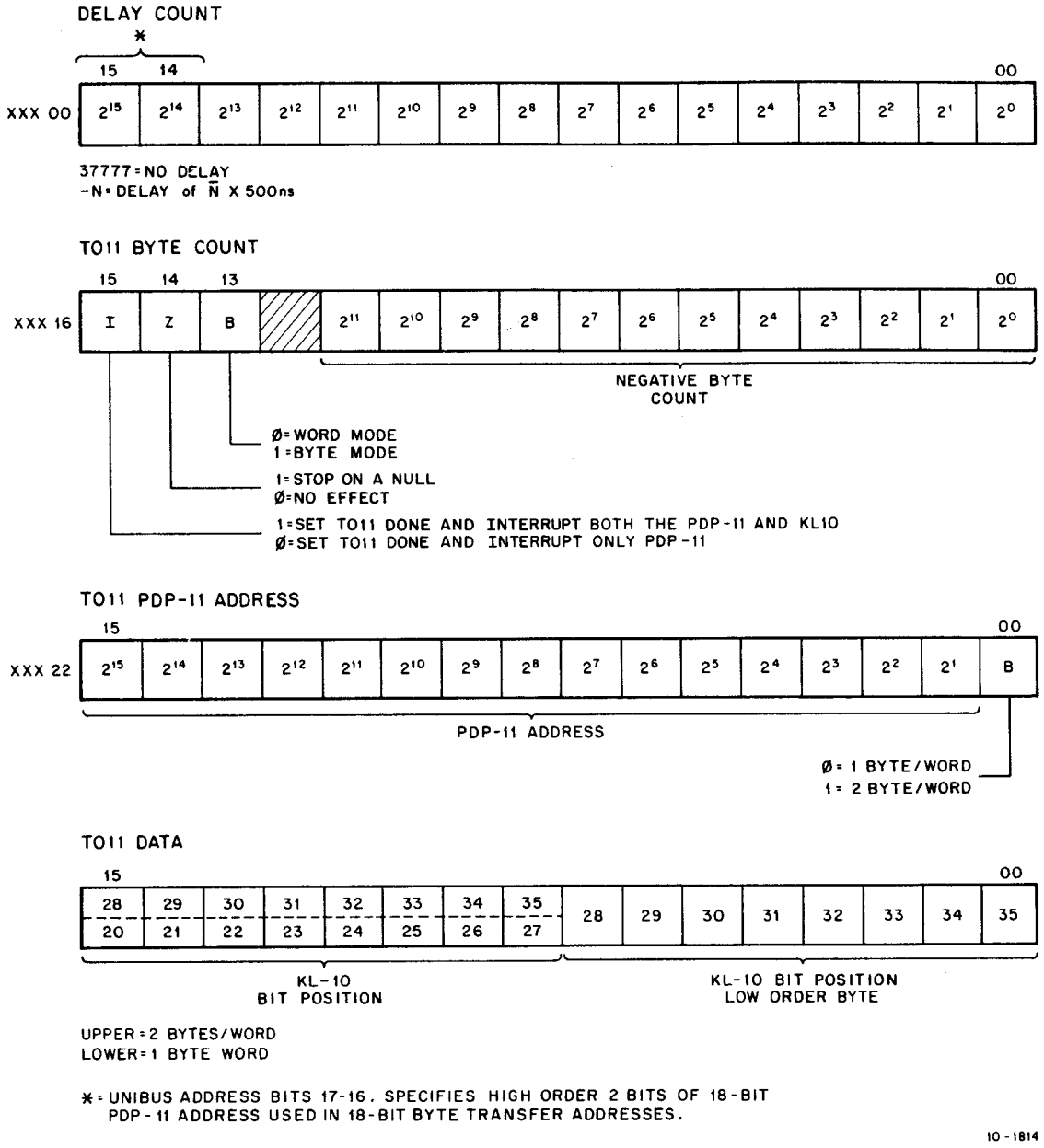


Figure 2-17 TO11 Transfer Words

EBus parity is sent with the data on TO11 transfers. The parity computation is performed on the output of data written to the RAM. Therefore, if no parity error is indicated, the data was written into the RAM correctly (excluding double bit failures). This is because the parity is checked after the write strobe occurs and the output has changed on the RAM, reflecting the data written into the cells. If a byte pointer of larger than 16 bits is used (on the KL10 side), that portion of the byte greater than 16 bits is stripped off the byte (does not get into the RAM), causing erroneous parity errors. A software restriction specifies a 16-bit byte pointer maximum size; this permits 16-bit byte pointers to be used safely on all TO11 transfers.

2.5.3.1 TO11 Transfer Operation – The format of the words used for the transfer operation is illustrated on Figure 2-18, TO11 Transfer. In addition, a functional flow, Figure 2-19, Simplified Flow TO11 Transfer, is included. Reference these while reading the functional description.

The first part of the operation is to read the TO11 Delay Count from RAM location (0) and load it into the Address and Byte Count (ABC) register. Here again, the high order address bits (16-17) are read out of the RAM and put into the address bits (on the data path module) for Unibus bits 16-17. The next flow state is CNT4 TO11 DLY INC. This state is maintained until the high order bit is incremented to zero; when this first occurs, the entire word becomes zero. Next, the TO11 Byte Count must be read from RAM locaton (7) into the ABC register and tested for a zero field. If the register iszero, the transfer terminates and an interrupt is given to the PDP-11 processor. If the ABC register field is non-zero, the CNT4 TO11 BYTE COUNT INC state is entered, which increments the ABC register by 1.

The flow proceeds to a point where the updated byte count is written back into RAM location (7). Note that, referring to the flow, the content of ABC goes to the RAM (7). At this time, a byte of data is required from the EBox, so the DTE enters the CNT4 TO11 I/O FCN state. This initiates an EBus dialogue, which results in the byte of data being fetched and placed into the E-Buffer.

A new KL10 API function is used to perform TO10 and TO11 transfers. It differs from the API function used in DEX operation. In the Examine operation, the function contained an address, as well as information about which DTE was interrupting and a function code. In the case of a TO11 transfer, the data address for use on the KL10 side is contained in a byte pointer (in KL10 core) and is not supplied by the DTE exclusively on the interrupt function. However, the fact that a TO11 transfer is in progress and requires a byte of data must be conveyed to the EBox. A combination of the function code and the Q-bit indicates to the EBox the required service type. A function code of 6 (in bits 03-05) specifies a byte transfer operation. If Q=0, a TO11 byte transfer is specified; Q=1 specifies a TO10 byte transfer.

At this time, the byte of data is in the E-Buffer. The flow enters the next state: CNT4 TO11 SHIFT, where the byte of data is adjusted for storage in the RAM. The rightmost E-Buffer bits do not have a direct data path available to the RAM; therefore, the byte must be shifted. The shift is performed during the TO11 address read operation from the RAM; no extra time is required to perform the shift. Referring to the flow, RAM location (11) is addressed and the ABC register is loaded with its contents, the TO11 address. During this period, the E-Buffer is also shifted. It is shifted left, placing the byte that was in bits 20-35 into bits 04-19. This is the normal position for writing into the RAM from the E-Buffer.

When writing into the high order bits in the PDP-11 (in byte mode), the two halves of the 16-bit word are swapped. During PDP-11 Unibus data operations, the transmitter of data (on byte operations) is responsible for putting bytes in the correct word halves, and should be thought of as putting "this half on that half, not this byte on that byte." On the KL10 side, a byte is generally right justified and the processor hardware takes care of all swapping. Bytes are thought of as being right justified. The PDP-11 recognizes left and right bytes. But all bytes from the KL10, because it uses byte pointers, come over

right justified. So when an 8-bit byte is transferred, it is right justified and would appear so in PDP-11 memory. Yet byte modes operate "right half, left half, etc." If a destination is a left half, for example, the word halves must be swapped. This is one reason the TO11 address is read prior to writing the data into the RAM. The destination address must be known before the data can be written into the RAM. This takes care of the swapping. The result is that a single byte is now positioned in bits 15-08 on the input to location (13), the TO11 data slot in the RAM. The adjusted byte is written during CNT4 TO11 EBUFF STORE. The DTE performs a standard PDP-11 NPR transfer to the data location. The Unibus dialogue begins. Refer to the *PDP-11 Peripherals Handbook* (EB 05961 76), Chapter 5 and Appendices A through D, for a complete description of the Unibus dialogue and a summary of Unibus addresses, miscellaneous data, and instructions.

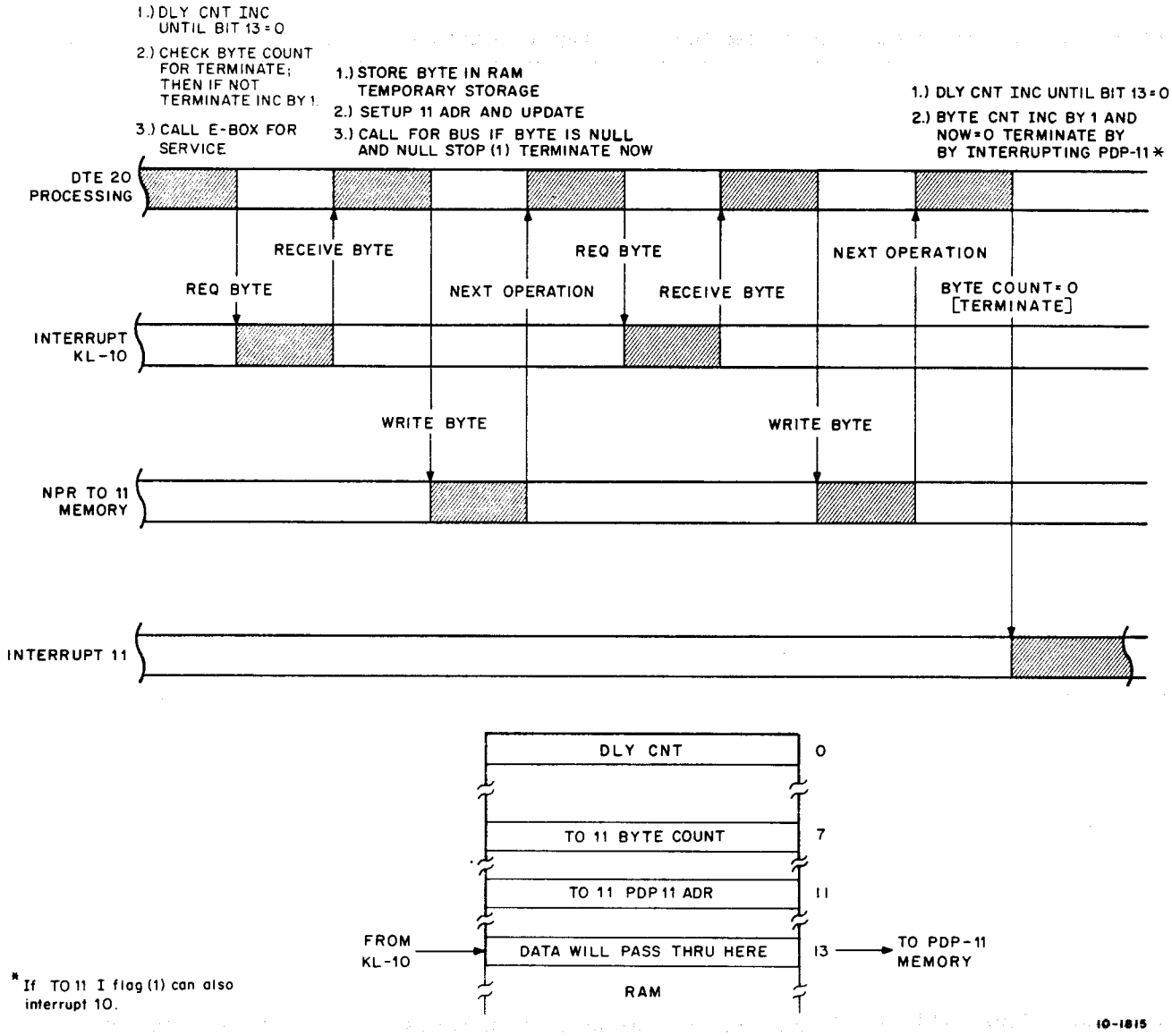


Figure 2-18 TO11 Transfer

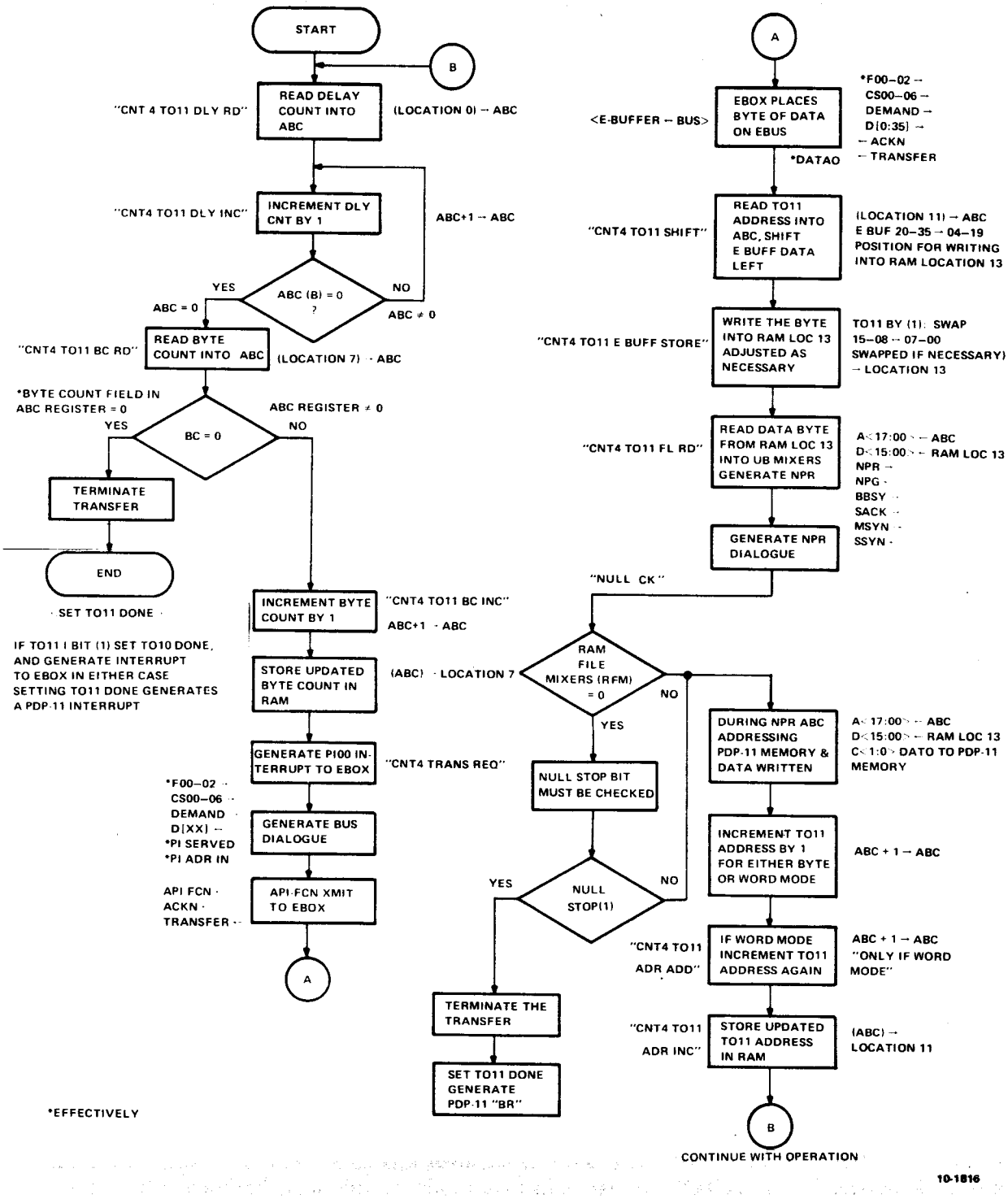


Figure 2-19 Simplified Flow TO11 Transfers

In addition to the NPR transfer, another mode of operation is possible: terminate transfer on NULL character. The NULL STOP flag must be set to accommodate this transfer. When the NULL character is read (on NULL STOP), the data transfer occurs. A zero word is written into core, the transfer takes place and then the operation terminates. The TO11 address in the DTE is not incremented. It must be loaded, however. If another byte count is loaded, to continue the transfer after the NULL character, the NULL character is overridden. This mode does not dispose of the normal NPR transfer; the NPR transfer still takes place on the NULL character.

The final state is CNT4 TO11 ADR INC. In byte mode, the address is incremented (in the ABC register) upon the return from the NPR transfer. In word mode, the address must be incremented twice. Once the ABC register is updated with the new address, the operation ends.

2.5.4 TO10 Transfer Overview

The preliminary phase for the TO10 transfer is as follows, the PDP-11 processor loads the following locations in the RAM:

RAM Address	Word Loaded
0	TO10 DLY Count
10	TO10 PDP-11 Address

The KL10 program loads the following location in the RAM:

6	TO10 Byte Count

In general, the TO10 transfer is quite similar to the TO11 transfer. With the TO10 transfer, the direction is opposite that of the TO11 transfer, and no NULL character is employed. Both states are identical except for reading the byte count from different RAM locations. After the byte count is read, the operations differ. On TO11 transfers, the KL10 is interrupted and an NPR occurs; on the TO10 transfer, an NPR fetches the data and then the KL10 is interrupted.

The least significant bit of the DIAG Word 3 is used to enable the transfer of 8-bit bytes from PDP-11 memory to the EBox. This word must be loaded into the RAM prior to starting the transfer of a block of bytes. The state of DIAG Word 3 bit 00 is stored in the TO10 Byte Mode flag within the DTE. This flag controls the positioning of the incoming byte (by the LSB of the address in byte mode) in the RAM, which acts as a temporary buffer for the byte.

Example: The first byte is read from an even-byte boundary. Refer to Figure 2-20, TO10 Byte Mode. The first byte enters the interface over Unibus lines 07-00 and is stored in the RAM in bits 07-00. Bits 15-08 are loaded with zeros. Then, the byte is loaded into the E-Buffer as follows: bits 15-00 of the RAM (the location holding the byte) is loaded into E-Buffer bits 20-35. From the E-Buffer, the byte is placed onto the EBus as indicated. The 8-bit byte occupies bits 28-35, which corresponds to the position that bytes occupy during TO11 transfers in byte mode. In this way, both are compatible.

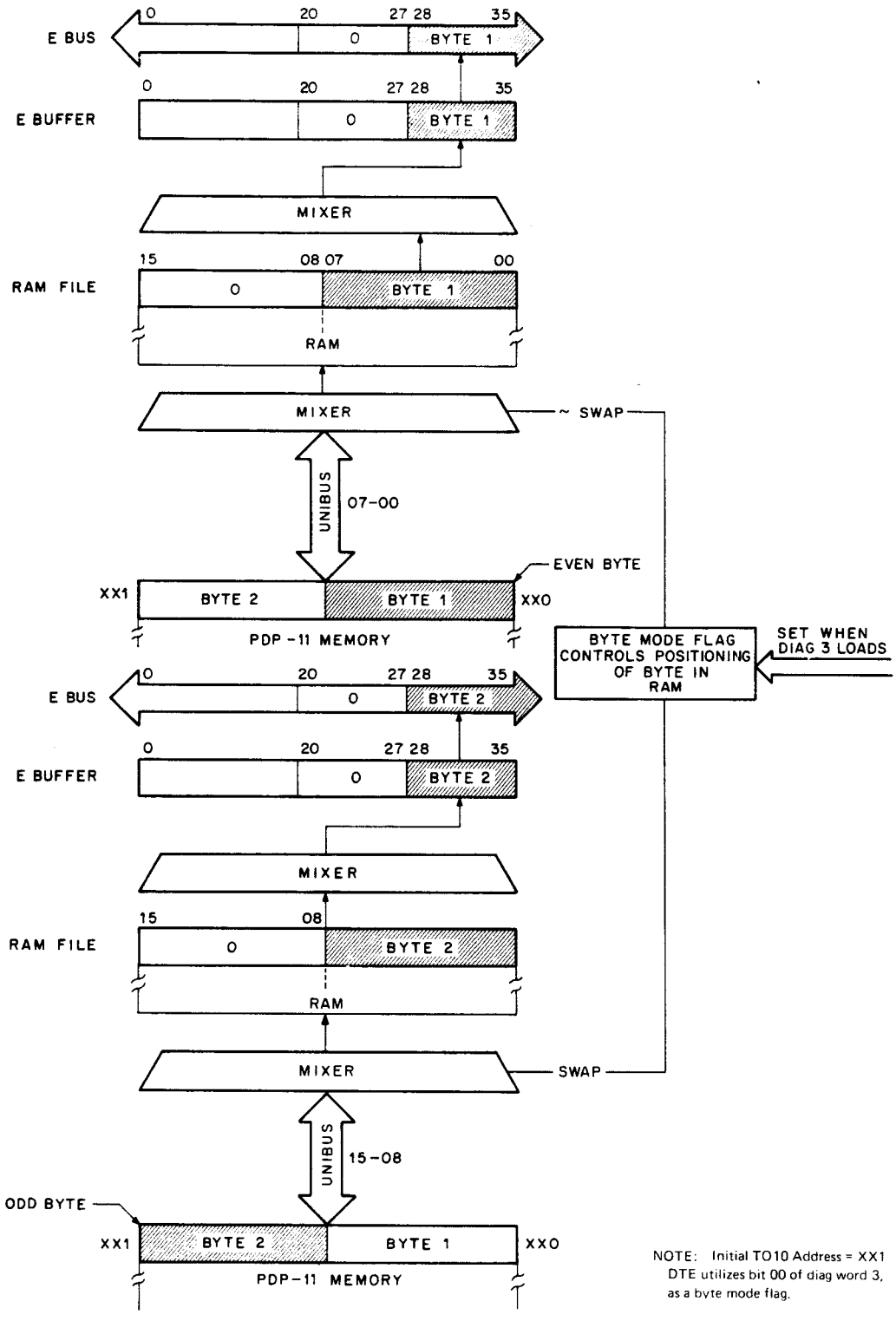


Figure 2-20 TO10 Byte Mode

The address is incremented by 1, in preparation for the next transfer. This is determined by testing the state of the TO10 Byte Mode flag. In Figure 2-20, notice the first byte is read from PDP-11 memory location XX0 and the second byte is not read using the updated address XX1. Remember, in byte mode, the PDP-11 memory appears as if composed of consecutive 8-bit bytes. These are in pairs; the low order byte in bits 07-00, and the high order byte in bits 15-08. The second byte enters the interface over Unibus lines 15-08 and is stored in an appropriate RAM location in bits 07-00. This is accomplished by a swapping technique. Bits 15-08 are swapped with bits 07-00 on the input to the RAM. This assures that the E-Buffer always receives the byte in bits 28-35. Finally, byte 2, the E-Buffer bits 20-35 containing the word 0, is placed onto the EBus for transfer to the EBox. Transfers continue until the byte count increments to zero. At that time, the TO10 Done flag sets, generating an interrupt to the EBox. Optionally, the PDP-11 may also be interrupted.

If the least significant bit (bit 00) is a 1 when DIAG Word 3 is loaded into the DTE, the interface is considered to be in word mode. Information is placed onto the Unibus in bits 15-00 as a 16-bit word. From there it is temporarily stored in a RAM location in bits 15-00 for transfer to the E-Buffer. The word is transferred to E-Buffer bits 20-35 and then to the EBus bits 20-35 for transfer to the EBox. See Figures 2-21 and 2-22, TO10 Transfer Simplified Functional Block Diagram and TO10 Transfer Words, respectively.

After each transfer, the address must be incremented by two, because in word mode, the PDP-11 memory appears as 16-bit words in consecutive even locations. Therefore, if the initial location is XX0, the second word comes from XX2. As previously indicated, the number of words to be transferred is controlled by a byte count. The byte count comprises a negative number equal to the number of consecutive bytes to be transferred. It increments toward zero, just as in the TO11 transfer. No Z bit is in the TO10 Byte Count word (as in the TO11 Byte Count word) to generate an interrupt to the PDP-11 upon detection of a NULL character. However, a provision is included to interrupt the PDP-11 upon setting the TO11 Done flag.

The TO10 Delay Count is the same as that previously described for the TO11 transfer. During transfers, location (12) in the RAM is used as a temporary buffer for the word or byte being transferred. The transfer from PDP-11 memory to the RAM is via the NPR facility.

The format of the words used for the transfer operation is illustrated on Figure 2-23, TO10 Transfer. In addition, a functional flow, Figure 2-24, Simplified Flow TO10 Transfer, is included. These should be referenced while reading the functional description.

The first part of the operation is to read the TO10 Delay Count and load it into the ABC register. Then the ABC register is incremented and the result is tested for ABC bit 13=0. As with a TO11 transfer, the delay count is a 16-bit word, and ABC bit 13 equal to zero indicates the zero condition. The Interface Control Logic holds the CNT4 TO10 DLY INC state until this condition is satisfied.

When the delay count reaches zero, the next state is entered and the byte count is read from RAM location (6) into the ABC register, where it is tested for bit 13 being equal to zero. If it contains a zero, the transfer is terminated and an interrupt is given to the EBox. In addition, if the I bit is set, the PDP-11 also is interrupted. If the ABC register bit 13 is non-zero when checked, then CNT4 TO10 BC ADD is entered and the contents of the ABC register are updated by 1. The flow passes to a point where the updated byte count can be stored back into its slot in RAM location (6).

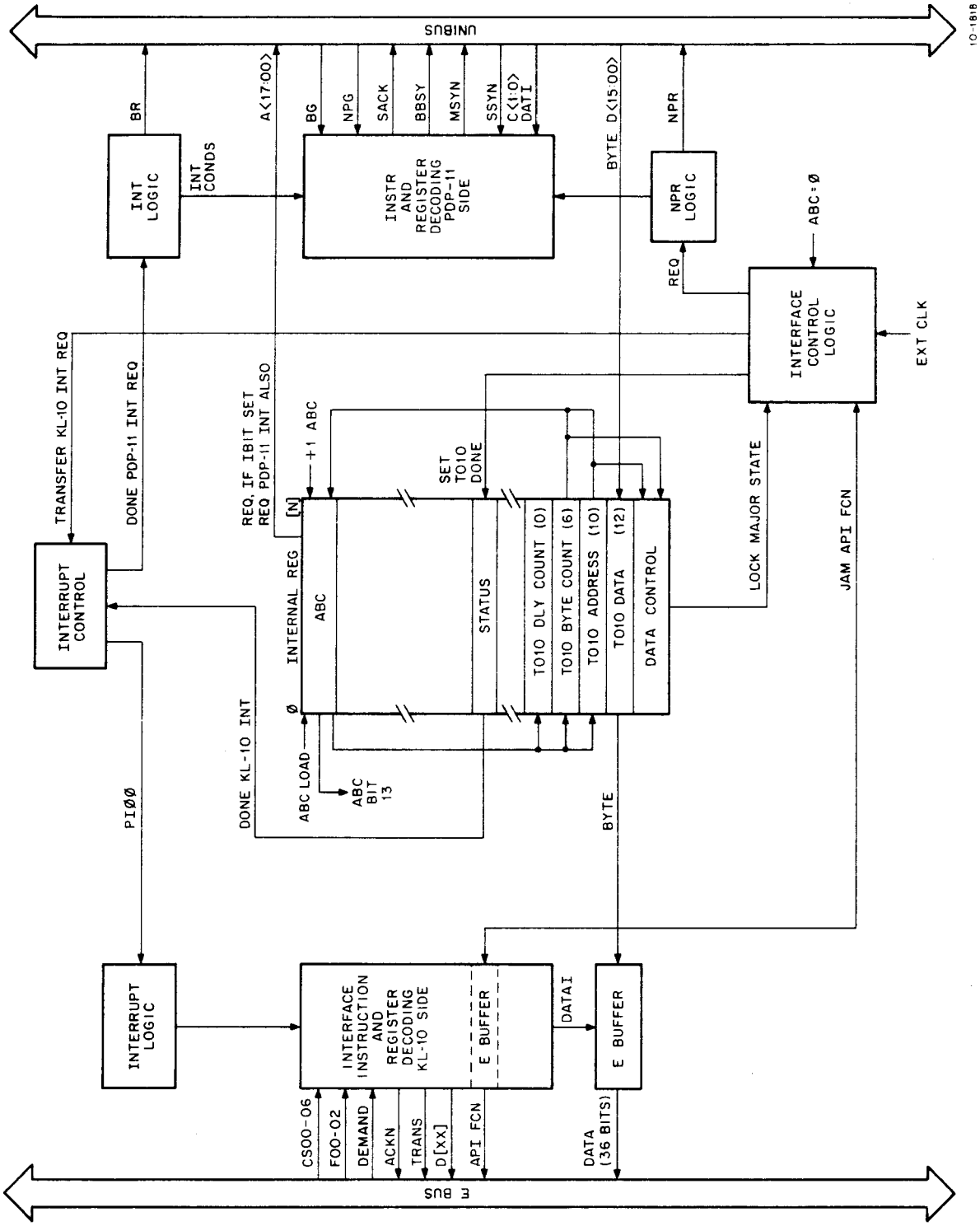


Figure 2-21 TO10 Transfer Simplified Functional Block Diagram

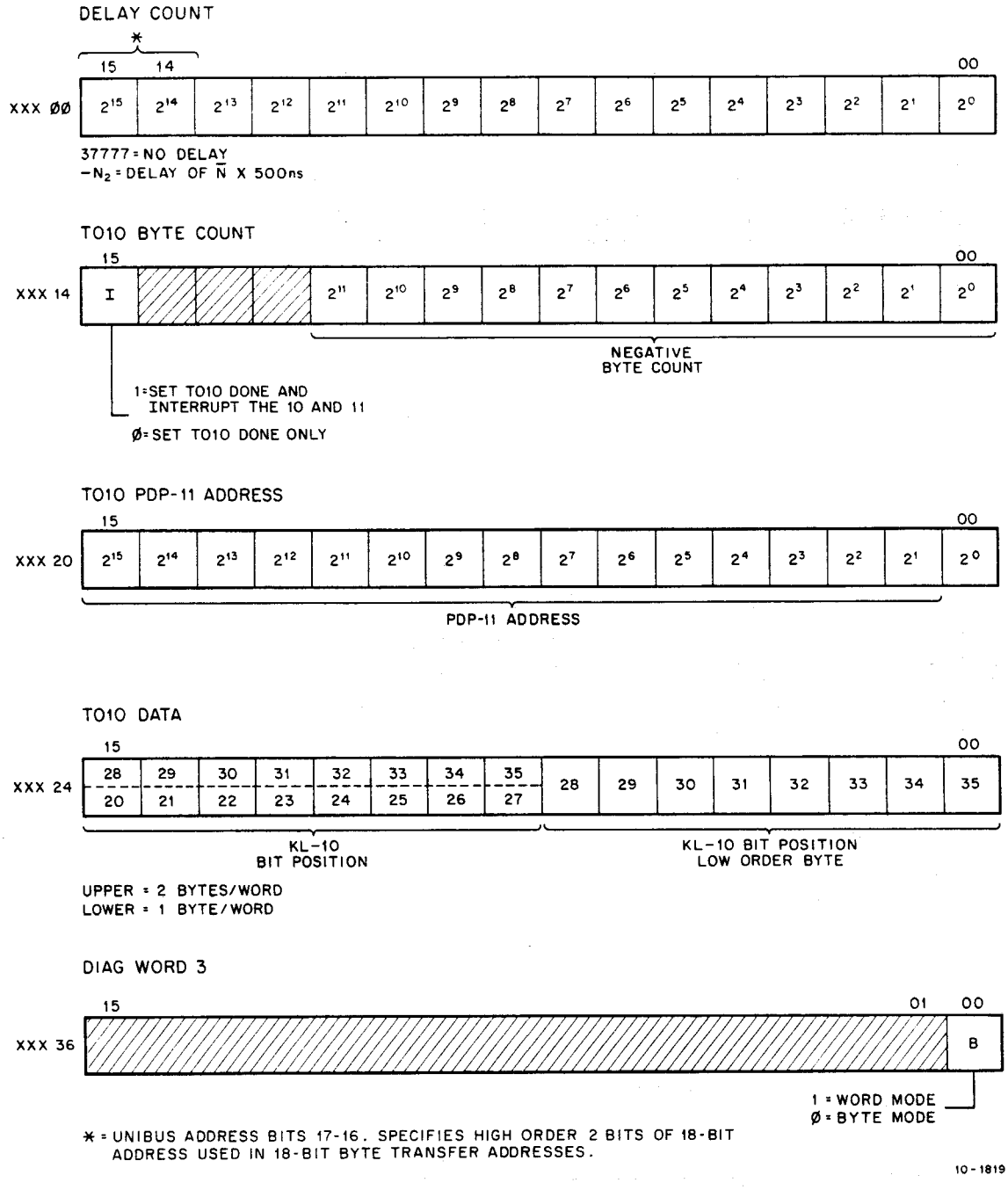


Figure 2-22 TO10 Transfer Words

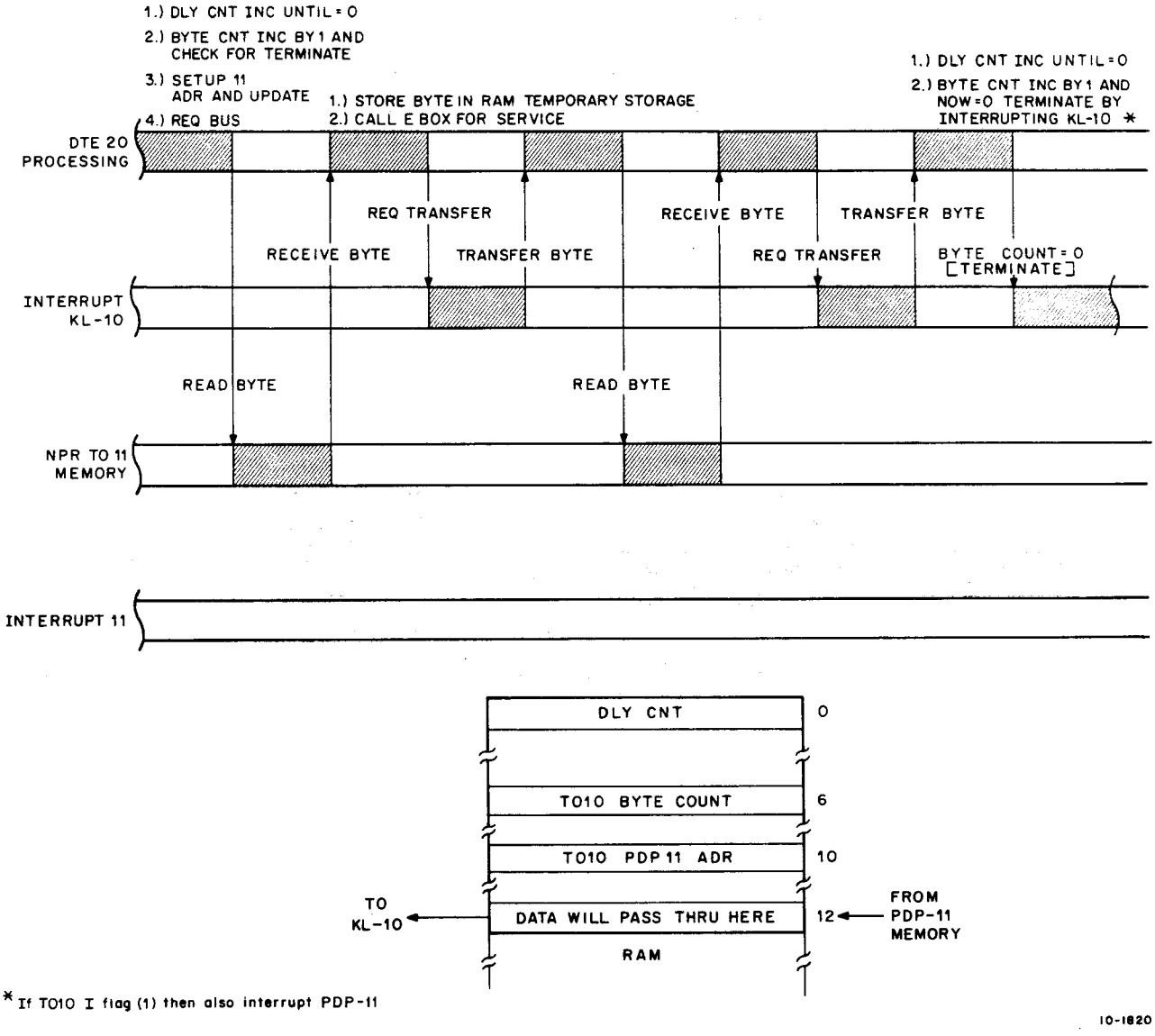


Figure 2-23 TO10 Transfer

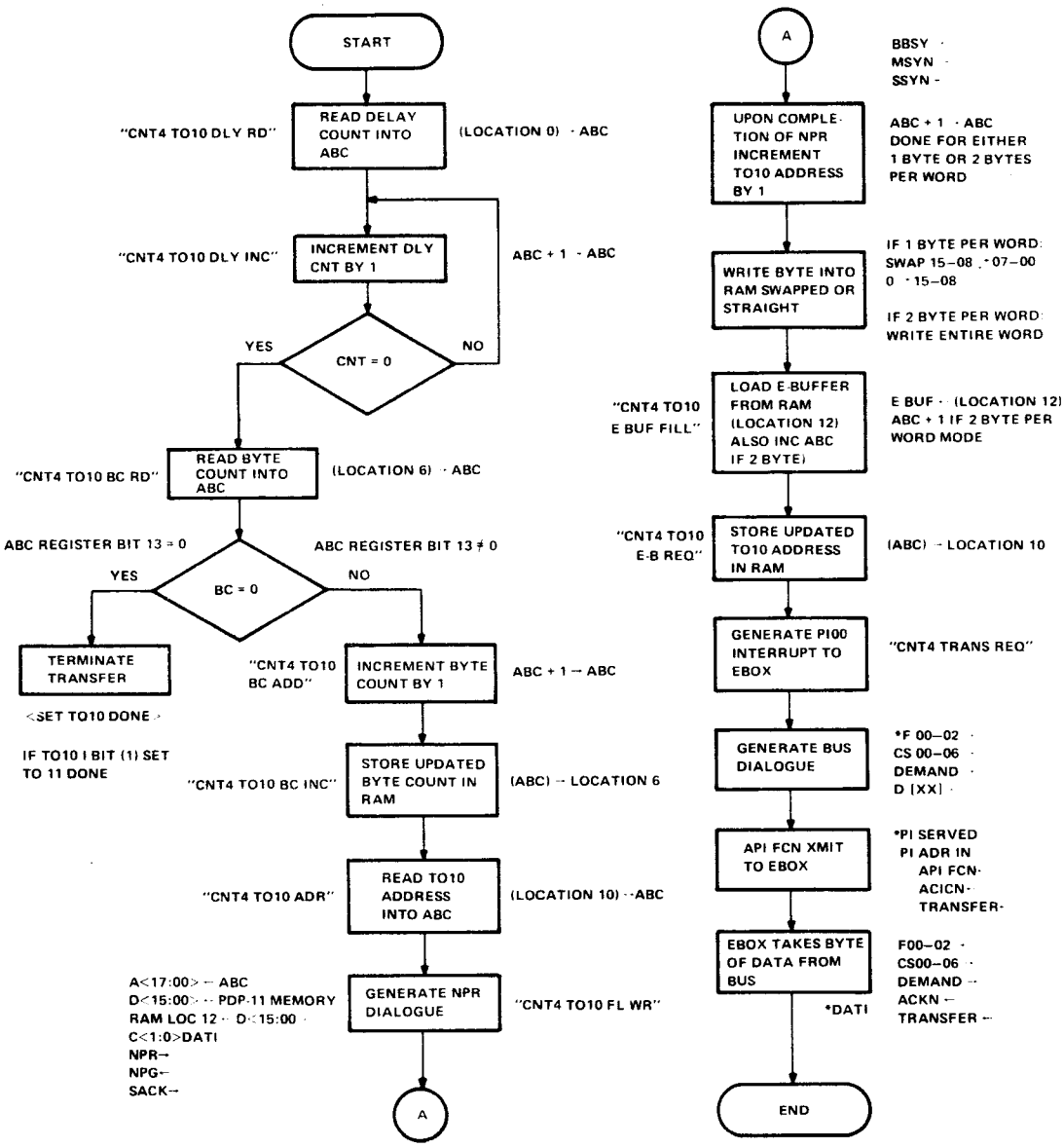


Figure 2-24 Simplified Flow TO10 Transfer

At this time, a byte of data is required from the PDP-11. The Interface Control Logic implements an NPR request to PDP-11 memory using the TO10 address currently stored in RAM location (10). First, the state CNT4 TO10 ADR is entered, the address is read from the RAM into the ABC register, and then an NPR is issued. The dialogue is shown on the flow and the connections are shown on the block diagram. When the dialogue is completed, a byte of data from the addressed location in PDP-11 memory is at the input to RAM location (12), where it is temporarily stored. It is written into this location, as in a TO11 transfer, either swapped or straight. Also, the contents of the ABC register, which were temporarily stored in RAM location (6), must be incremented by 1 and read into the E-Buffer for transfer to the EBox. The ABC register is incremented again and the updated address is placed into its slot in the RAM location (10).

An interrupt request is sent to the interrupt control from the DTE control logic at this time and the DTE/EBox dialogue ensues (refer to Section 2.5.3, TO11 Transfer CNT4 TO11 I/O FCN, for details). As a result of the dialogue, the EBox performs a DATAI and takes the data from the E-Buffer. Then, the DTE releases the EBox and the operation ends.

SECTION 3
LOGIC DESCRIPTIONS

3.1 DATA PATH ORGANIZATION

This section introduces the DTE20 interface by describing the logical elements that comprise the two major portions of the interface and supporting diagrams, DTE20 Data Paths, Figure 3-1, and DTE20 Control, Figure 3-2. As shown on Figure 3-1, the DTE20 data path essentially contains the following:

1. Random Access Semiconductor Memory (RAM) 16 words × 16 bits
2. Bus Drivers and Receivers for both the Unibus and EBus
3. Bus Drivers and Receivers for the Diagnostic Bus
4. Four Addressable Registers

    a. Diagnostic Word 1
    b. Status Register
    c. Diagnostic Word 2
    d. Diagnostic Word 3

5. Three Non-Addressable Registers

    a. E-Buffer Register
    b. Address and Byte Count Register
    c. E-Buffer Hold Register

6. Various Mixers and Combinational Logic.

3.1.1 Random Access Memory (RAM) (Figure 3-12)

The RAM serves the 10-11 Interface both as a data buffer and as a control buffer. It is pre-formatted so that a given RAM location always contains specific functional information. Twelve of the 16 available words of RAM storage are used. The RAM can be addressed by the PDP-11 processor via the Unibus lines A(17:00), and can be read or written using DATI or DATO instructions, respectively. The KL10 processor cannot directly access the RAM for read or write.

3.1.2 EBus Drivers

The EBus data lines are fed from the E-Buffer mixer. The mixer can supply three basic types of information:

1. Status Information
2. Data
3. Interrupt Information.

The second source is the Diagnostic register that is addressed as Diagnostic Word 1. This register can supply 128 functions to the KL10 (EBox) as well as enable certain 10-11 Interface functions.

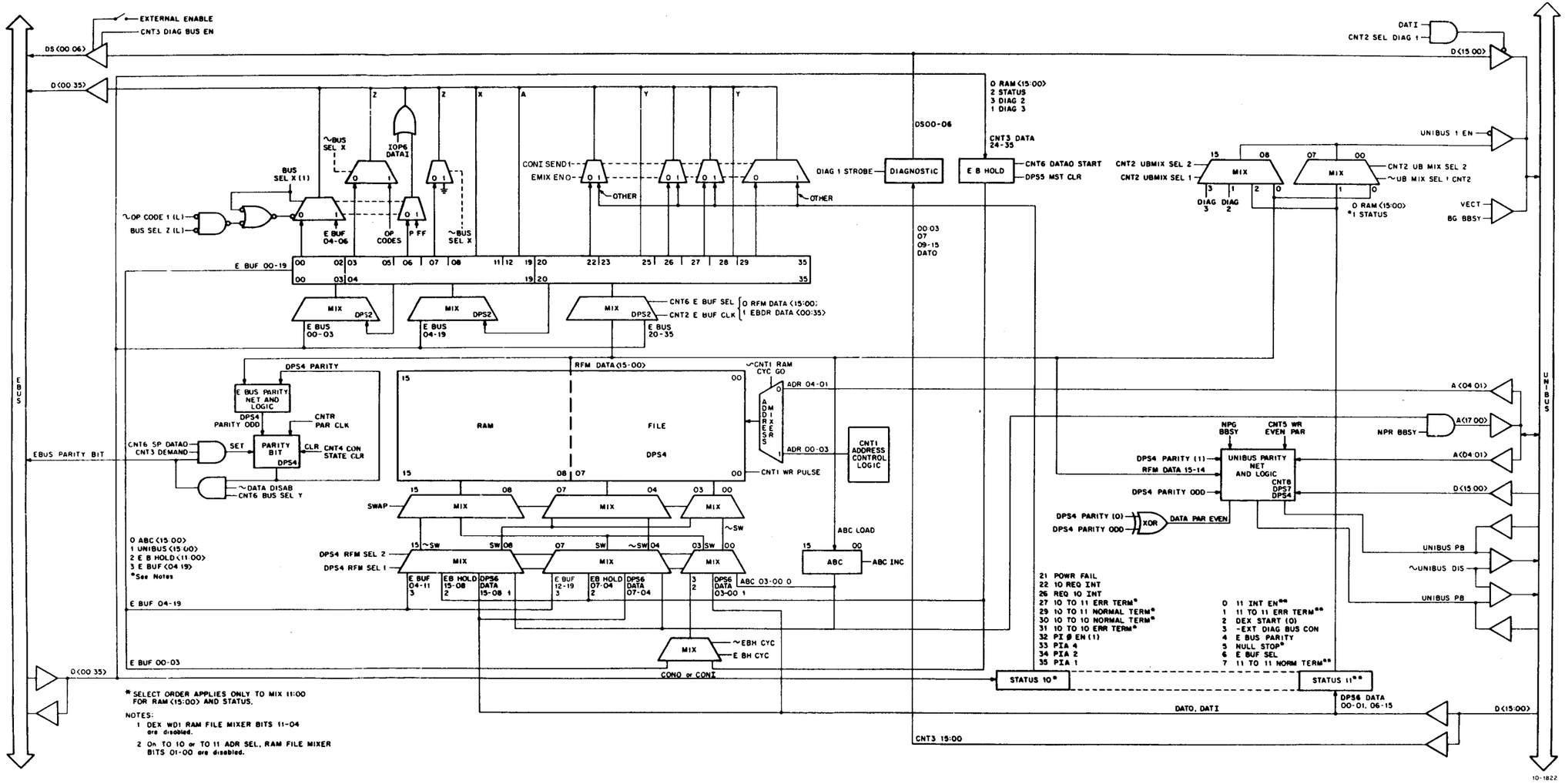


Figure 3-1 DTE20 Data and Address Paths Block Diagram

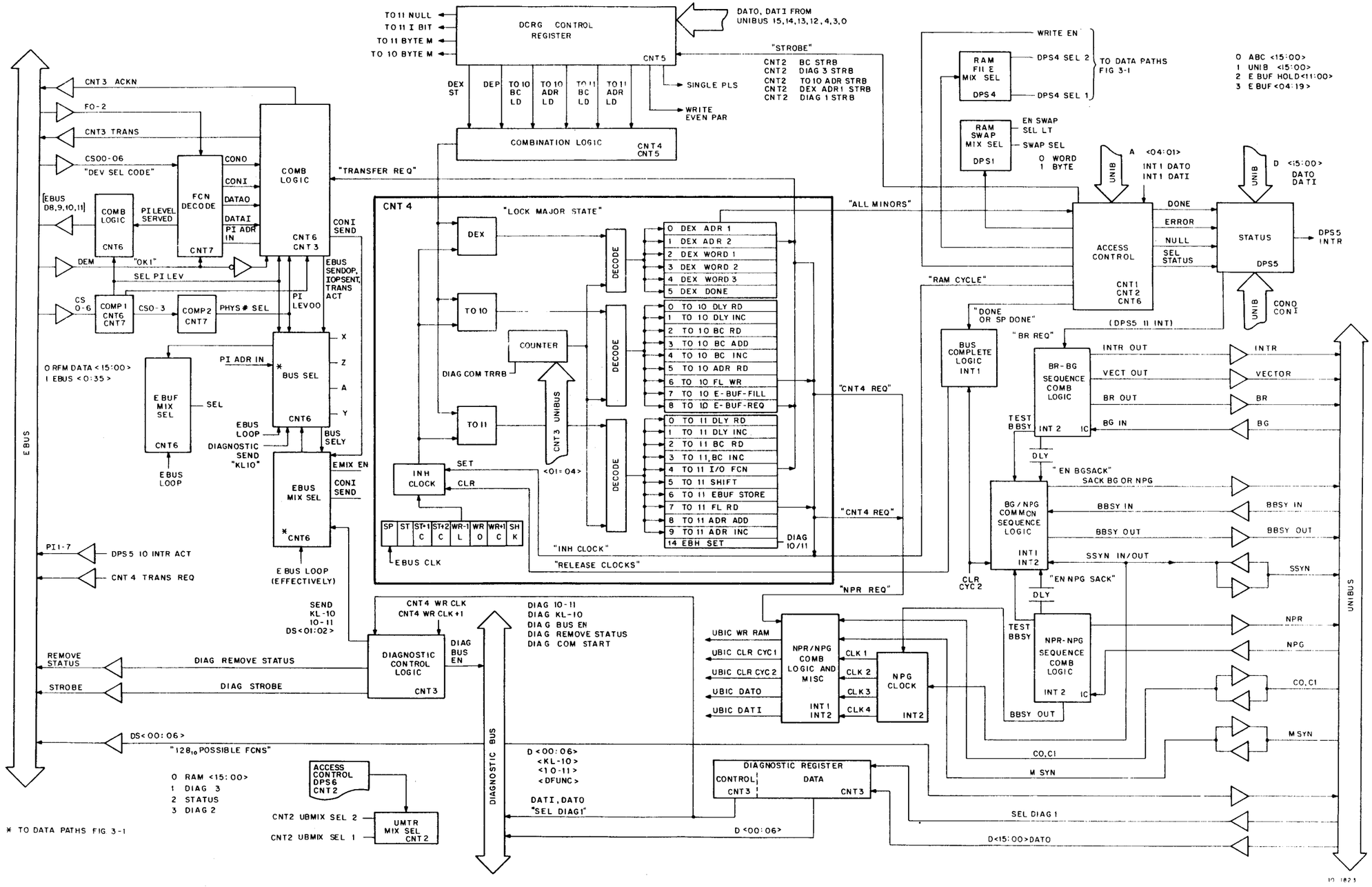


Figure 3-2 Detailed Control Block Diagram

3.1.3 EBus Receivers
The EBus receivers can supply information to three destinations in the interface:

1. E-Buffer Input Mixers
2. Status Register
3. E-Buffer Hold Register.

3.1.4 Diagnostic Bus Drivers
The diagnostic bus, on the EBus side, is a logical extension of the EBus. Its operation is under the control of PDP-11 diagnostic software. When the diagnostic bus is not actively used for diagnostic read, write, or console functions, its lines contain processor status information; the bus is never disabled. Reading vital processor status information (e.g., is the machine halted) is performed by just reading the specific Diagnostic register location. The register itself is not read, the bus is. When the register is written into, those bits are placed on the bus. Then the bus bits are read.

The EBUS REMOVE DS STATUS signal is asserted when a diagnostic function to the DTE is performed. The signal controls removing the processor status information from the DS lines.

3.1.5 Four Addressable Interface Registers
Twelve DTE locations reference the RAM and require a RAM cycle. This cycle is a larger and more complex interfacing cycle to perform than a regular cycle. Four DTE locations do not access the RAM, but are accessed by a simpler mechanism. The four locations are Diagnostic Word 2, Diagnostic Word 3, Diagnostic register, and the Status register. These locations should be used first in diagnosing problems because the simpler mechanism will probably fail less often than the more complex cycle mechanism.

3.1.6 Three Non-Addressable Registers
In order to perform the necessary manipulations on internal data and control information, the 10-11 Interface uses three working registers:

1. E-Buffer Register
2. Address and Byte Count Register
3. E-Buffer Hold Register.

3.1.7 Miscellaneous Mixers and Combinational Logic
The remainder of the logic in the Data Paths Logic provides the interconnections among the various elements pointed out in Sections 3.1.1 through 3.1.6 to facilitate proper interface data manipulation.

3.2 CONTROL SECTION ORGANIZATION
Refer to Figure 3-2. The 10-11 Interface control section essentially contains the following:

1. Access Control Logic
2. KL10 Interrupt Dialogue Logic
3. PDP-11 NPR Dialogue Logic
4. PDP-11 BR Dialogue Logic
5. Control State Timing Logic
6. DATA Register Control Logic
7. Diagnostic Control Logic.

3.2.1 Access Control Logic
The access control is driven from two basic sources: the first source is the clock and state control, which provides all the minor states from the three major states: DEX, TO11 transfer, and TO10 transfer. These minor states are gated against conditions arising in the interface and become the clocks

to provide synchronized control levels or pulses that are fed to various places within the interface and from the access control. Some of the functions performed from the Access Control Logic are:

RAM File Address Selection
WRITE Control
RAM File Mixer Selection
E-Buffer Loading
ABC Register Loading and Increment
Unibus No. 1 Enable
Addressable Register Selection
Generation of DONE Conditions and NULL STOP.

The second source is from the Unibus. The high-order four of the least significant five PDP-11 address lines A(04:01) are passed to the access control along with the DATO/I level. Decoders in the access control decode the address lines and DATO/I into a discrete signal name depending upon its weight; i.e., 00 decodes CNT2 SEL DLY CNT. The decoding of a RAM address enables a RAM cycle, which will implement the appropriate write or read operation. Upon completion, the interface logic shuts down.

3.2.2 KL10 EBus Dialogue

The majority of this logic is shown on CNT6 and CNT7 prints. The logic consists of a decoder for function as well as comparators for physical controller number and PI channel, control flip-flops to synchronize the events, and bus selection and mixer selection logic. The logic can be triggered by two basic events:

1. Control state interrupts; i.e., those that are interface generated. Also status interrupts; i.e., programmed interrupts.

2. The second type of event is the execution of an input/output instruction to the DTE20.

3.2.3 PDP-11 NPR Dialogue

The NPR Control Logic is shown on the INT1 and INT2 prints. This logic is used by the DTE for sending or receiving data to or from PDP-11 memory via the Unibus. The address for this transfer is always placed in the ABC register for use during the NPR transfer, but it is not an ABC register generated address.

All control logic necessary to sequence an NPR transfer is on the M8554 (INT) module. Therefore, if either an NPR or an interrupt does not function properly, this module is probably at fault.

3.2.4 PDP-11 BR Dialogue

The BR (bus request) Control Logic is shown on the INT1 and INT2 prints. The DTE uses this logic when requesting the following interrupts to the PDP-11:

TO11 DONE
TO10 DONE
DOORBELL.

3.2.5 Control State Timing Logic (Figure 3-2)

The control state timing is responsible for the implementation of all the major and minor state timing levels. By monitoring the Data Control Logic, the Control State Logic determines when to begin a specific sequence; e.g., DEX, TO10 transfer, or TO11 transfer. The logic mainly comprises an external clock input (EBus clock), an 8-stage ring-counter that serves as the main clock, and a 3-stage ring-counter that serves as a major State register. Also, a 4-stage BCD counter governs the minor states for the DEX and TO10/TO11 transfers.

3.2.6 Data Control Register Logic

The Data Control register logic is shown on the CNT4 and CNT5 prints. The register contains operational Status flags:

Operation	Pair
DEX	CNT5 DEP FF, CNT5 DEX START FF
TO11 Transfer	CNT5 TO11 BC LD FF, CNT5 TO11 ADR LD FF
TO11 Transfer	CNT5 TO10 BC LD FF, CNT5 TO10 ADR LD FF

The remaining five flip-flops remember the following:

TO11	• If (1), 8-bit byte transfers
	• If (0), 16-bit byte transfers
TO10 Byte	• If (1), 8-bit byte transfers
	• If (0), 16-bit byte transfers
TO11 NULL STOP	• Stops transfers upon detecting NULL character (0)
TO11 I Bit	• Interrupt the PDP-11 to get a new Byte Count word when TO11 DONE (1)
Single PLS	• A diagnostic function; allows stopping the 10-11 interface after each clock pulse
TO10 I Bit	• Interrupt both the PDP-11 and KL10 when TO10 DONE sets

The remainder of the logic controls the KL10 interrupt dialogue, and assures the proper EBus selection and EBus Mixer selection during interrupts.

3.3 BASIC BUS TRANSACTIONS

To allow communication between the KL10 processor and the PDP-11 processor, the doorbell feature is included as part of the 10-11 Interface. Both processors can use this feature. To ring the doorbell, the appropriate processor must perform a specific input/output instruction, which will set the Doorbell flag in the interface. Table 3-1 lists the essential steps necessary in the process.

3.3.1 PDP-11 Rings Doorbell

Refer to Figures 3-1 and 3-2, and Table 3-1. As indicated in Table 3-1, to ring the KL10's doorbell (i.e., generate a KL10 interrupt) the PDP-11 must execute a DATO instruction to the 10-11 Interface while addressing the Status register, and bit 08 on the Unibus will set the appropriate flag. In addition, at some previous time the KL10 must have assigned a channel number to the interface, via a CONO instruction. A standard Unibus dialogue continues. Refer to the *PDP-11 Peripherals Handbook* (EB 05961 76), Chapter 5, for a complete description of the dialogue.

3.3.2 Interrupt Dialogue

Refer to Figure 3-2. The setting of DPS5 REQ 10 INIT leads to the generation of DPS5 10 INTR ACT. This signal is passed to the EBus as the appropriate interrupt line, i.e., PI 1-7. The EBox must arbitrate this with other incoming interrupts from other devices. The following dialogue selects the device that the KL10 serves at this time (in this case, the interface).

Table 3-1
Typical Doorbell Sequence Abbreviated

Processor	Function	Action	Reason	Bit	Flag
KL10	Ring PDP-11 Doorbell	CONO	Negotiate byte XFER	22	SET DPS510 REQ INT
PDP-11	Read Interface Status Register	DATI	Determine reason for INTERRUPT	ALL Relevant	N.A.
PDP-11	Turn OFF Doorbell	DATO	Answer the doorbell and perform preliminary setup for BYTE XFER	10	CLR DPS510 REQ INT
			Transfer is negotiated here		
PDP-11	Ring KL10 Doorbell	DATO	Negotiate BYTE XFER	08	SET DPS5 REQ 10 INT
KL10	Read Interface Status Register	CONI	Determine Reason for INTERRUPT	ALL Relevant	N.A.
KL10	Turn OFF Doorbell	CONO	Answer Doorbell and perform preliminaries	26	CLR DPS5 REQ 10 INT

Having arbitrated the priorities, and wishing to select as candidates those devices on channel (N), the EBox asserts the following:

CS04-06; Channel No. (N)
F00-02; PI Level Served
DEMAND; Activate Decoders.

The 10-11 Interface Data Control (CNT3, CNT6 and CNT7) and Bus Control (CNT3 and CNT7) Logic handle the decoding and dialogue. Upon detecting DEMAND, the interface compares its assigned channel number to that provided by the EBox as well as decodes function to be PI LEVEL SERVED. The result causes the interface, as well as any other devices on the same channel to assert their physical controller numbers EBus D (8, 9, 10, 11). Each control will place its physical controller number in a predefined bit position on the EBus. The EBox then strobes the EBus. Then, the EBox must decide which physical controller it wishes to service. In this case, assume the interface is to be serviced. The EBox asserts the following:

CS00-03; Physical Controller No.
CS04-06; Channel No. (N)
F00-02; PI Address In
DEMAND; Activate Decoders.

The interface, upon detecting DEMAND for the second time with PI ADDRESS IN and physical controller number and channel number (N), asserts the appropriate bus select levels and EBus Mixer select levels, which cause the API function to be placed onto the EBus drivers. The API function format in this case is shown in Figure 3-3.

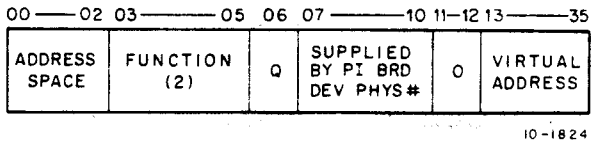


Figure 3-3 API Word Format Vector Interrupt

Refer to Figure 3-4 (sheet 2 of 2); two tables are shown. The first shows the bus select levels X, Z, A, Y, and data disable. These levels ultimately select some combination of the E-Buffer or the E-Mixer as input to the EBus drivers. There are two possible states for each bus select level; High (H) or Low (L). As indicated, the inputs to EBus driver bits 0-2, 3-5, 20-22, 26, 27, and 29-35 are mixed. Either the E-Buffer or some form of control or status information enabled into the EBus Mixer is provided as input. Note that the inputs to EBus driver bits 8-11, 12-19, 13-25 and bit 28 come directly from the corresponding E-Buffer bits.

The second table (Figure 3-4, sheet 2 of 2) shows the E-Mixer selects, a select line, and the enable line provided. The four types of input provided are:

E-Buffer
Status
Interrupt Information
Nothing.

For convenience, the EBus driver select and EBus Mixer selection logic are shown in a simplified form on Figure 3-4 (sheet 1 of 2), EBox Interrupt Dialogue. The interrupt sequence is indicated on the illustration by steps 1 through 7. The detailed breakdown on the EBus and E-Buffer Mixers is illustrated on Figure 3-5. On Figure 3-4, note that at 7, the signal EBUS SEND OP (send API word) enables Bus Select Z. This causes EBus bits 3-5 to be taken from the EBus Mixer bits 3-5 rather than the alternate, which would be 0s in bits 3-5. On the second table at the top of Figure 3-4, (sheet 2 of 2), the EBus Mixer selection, with Bus Select Z(L) asserted, enables the EBus Mixer. The source input to the EBus Mixer is a function of Bus Select X(L). When Bus Select X(L) is true, the EBus Mixer selects E-Buffer bits 04-06; but during EBus SEND OP, Bus Select X(L) is false and the selection is from E-Buffer bits 00-02. For function codes 4 or 5 the term "~OP CODE1" is true, enabling the EBus Mixer to select the E-Buffer bits 00-02. For functions 2 and 6, the EBus Mixer is disabled and the address space field of the API word being placed on the EBus is forced to 0. This will be interpreted by the EBox as the Executive Process Table (EPT). The EPT contains the TO10 and TO11 byte pointer words used while performing function 6 and the vector interrupt instruction used while performing function 2. Also indicated on Figure 3-4, ~OP CODE1 (true for Examine and Deposit API function codes 4 and 5) enables Bus Select A, which in turn enables Bus Select Y. This provides the virtual address in E-Buffer bits 13-35 as input to the EBus. Bus Select X(L), when false, enables the Protection Off flip-flop in the DTE20 to become the Q bit (EBus bit 6) in the API word.

3.3.3 KL10 Rings Doorbell

Refer to Figures 3-1, 3-2, 3-4, and Table 3-1. As indicated in Figure 3-1, to ring the PDP-11's doorbell (i.e., generate a PDP-11 BR interrupt), the KL10 must execute a CONO instruction to the 10-11 Interface while selecting the controller as the device, and bit 23 of the EBus will set the appropriate flag. Refer to Figure 3-2. The CONO is implemented as follows. The EBox asserts the following:

CS(04-06); Device Select
F(00-02); CONO
DEMAND; Activate Decoders.

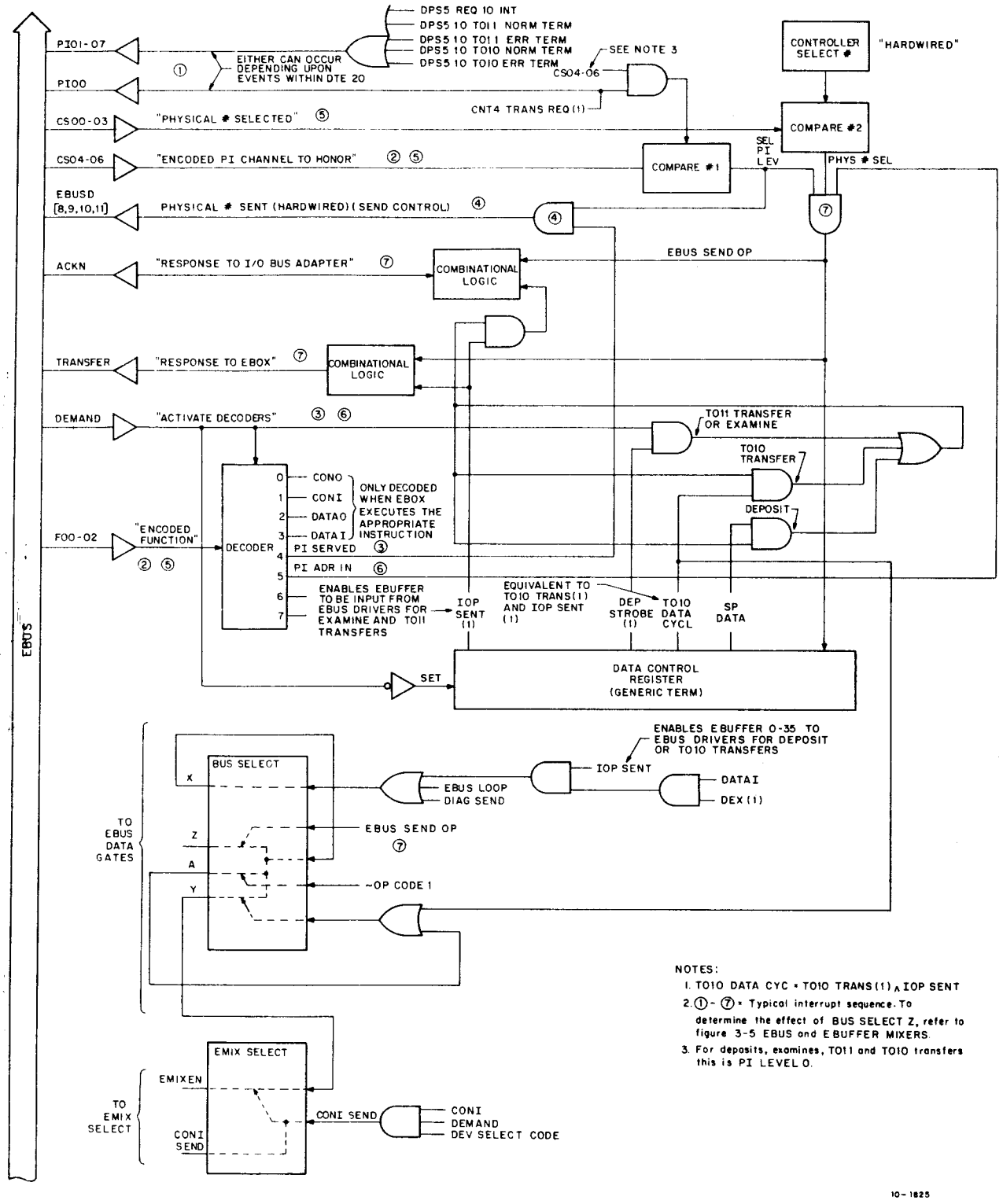
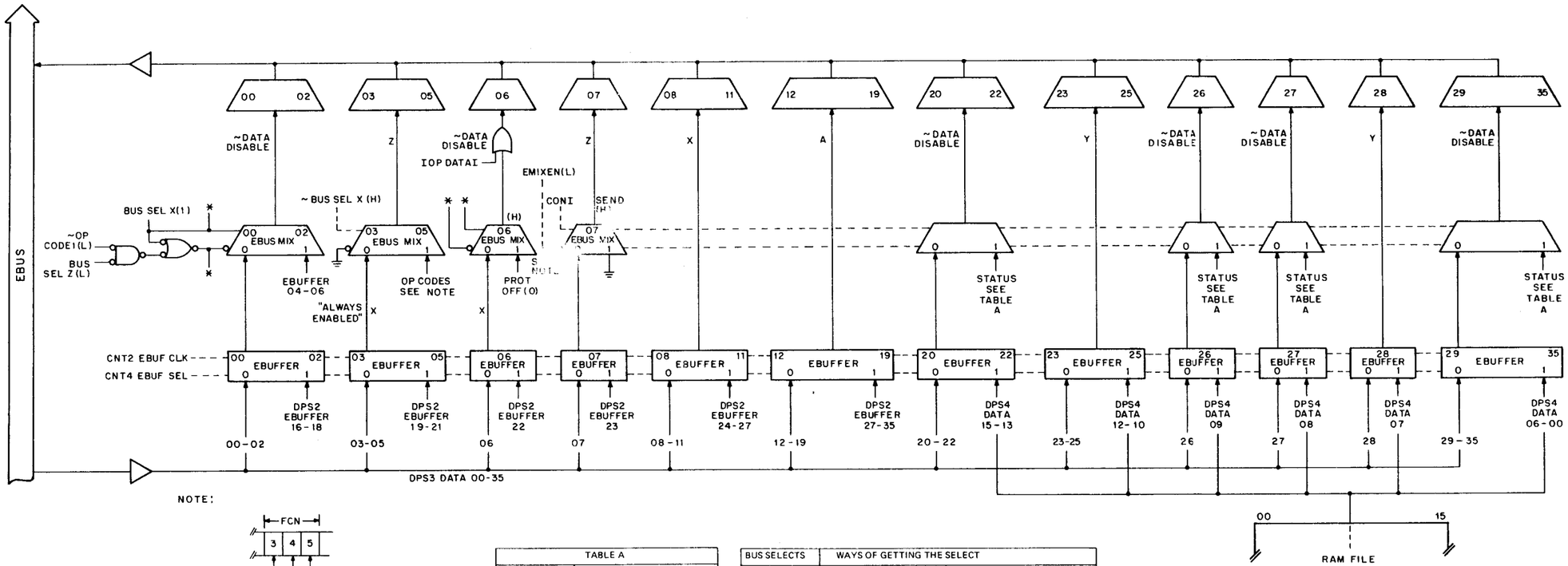


Figure 3-4 EBox Interrupt Dialogue (Sheet 1 of 2)

EBUS DATA GATES	EBUS												
BUS SELECTS	0–2	3–5	6	7	8–11	12–19	20–22	23–25	26	27	28	29–35	FUNCTION
BUS SEL X(L)					EBUFF 8–11								
BUS SEL Z(L)		EBUS MIX 3–5		EBUS MIX 7									
BUS SEL A (L)						EBUFF 12–19							
BUS SEL Y(L)								EBUFF 23–25			EBUFF 28		
~ DATA DISABLE	EMIX 0–2	ANDED WITH BUS SEL Z(L)	EMIX 6	ANDED WITH BUS SEL Z(L)	ANDED WITH BUS SEL X(L)	ANDED WITH BUS SEL A(L)	EBUS MIX 20–22	ANDED WITH BUS SEL Y(L)	EBUD MIX 26	EBUS MIX 27	ANDED WITH BUS SEL Y(L)	EBUS MIX 29–35	
DATA DISABLE	0	0	0	0	0	0	0	0	0	0	0	0	

EBUS MIXER SELECTION					EBUS DATA GATES												
CONI SEND(H)	EMIX EN(L)	BUS SEL X(L)	BUS SEL SEL Z(L)	NOPCODE 1(L)	0–2	3–5	6	7	8–11	12–19	20–22	23–25	26	27	28	29–35	FUNCTION
		TRUE	TRUE	TRUE	EBUFF 0–2	EBUFF 3–5	EBUFF 6	EBUFF 7									
		FALSE	TRUE	TRUE	EBUFF 4–6	TRANS ACT, OPCODE 1, OPCODE 2	PROT OFF(0)	0									
							IOP 6 DATAI										
TRUE	TRUE										STATUS		STATUS	STATUS		STATUS	
FALSE	TRUE										EBUFF 20–22		EBUFF 26	EBUFF 27		EBUFF 29–35	
		TRUE				EBUFF 3–5		EBUFF 7									
		FALSE				TRANS ACT, OPCODE 1, OPCODE 2		0									

Figure 3-4 EBox Interrupt Dialogue (Sheet 2 of 2)



NOTE:

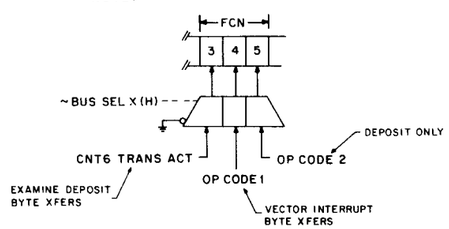


TABLE A	
E BUS MIXER BIT	TYPE OF STATUS
20	~EXT DIAG BUS EN
21	INT1 PWR FAIL
22	DPS5 10 REQ INT
26	DPS5 REQ 10 INT
27	DPS5 10 TO11 ERR TERM
29	DPS5 10 TO11 NORM TERM
30	DPS5 10 TO10 ERR TERM
31	DPS5 10 TO10 NORM TERM
32	DPS3 PI0 EN
33	DPS3 PIA 4 (1)
34	DPS3 PIA 2 (1)
35	DPS3 PIA 1 (1)

BUS SELECTS	WAYS OF GETTING THE SELECT
Z	1.) CNT6 EBUS SEND OP 2.) BUS SEL X
X	1.) CNT4 EBUS LOOP 2.) CNT6 DIAG SEND 3.) CNT6 IOP SENT (1) ^ [CNT7 DATAI ^ CNT4 DEX (1)]
Y	1.) BUS SEL X 2.) BUS SEL A 3.) CNT6 TO10 DATA CYC = CNT4 TO10 TRANS (1) ^ CNT6 IOP SENT
Z	1.) ~CNT6 OP CODE 2.) BUS SEL X
DATA DISABLE	1.) ~EXT DIAG BUS CON EBUS DATA DISABLE

CNT6 E BUFFER SELECT	
0	E BUFFER 00–19 ← E BUFFER 16–35 E BUFFER 20–35 ← RAM DATA 15–00
1	E BUFFER 00–35 ← EBUS 00–35

10-1826

Figure 3-5 EBus and E-Buffer Mixers

The interface, upon detecting DEMAND, decodes the device select as the 10-11 Interface and the function as CONO, and generates a strobe that leads the EBus bus (23) into the STAT 10 REQ INT FF (Figure 3-1). This flag leaves the Status register as BR REQUEST. Finally, the interface asserts ACKNOWLEDGE (ACK) and TRANSFER. This completes the dialogue on the KL10 side of the interface.

3.3.4 Interrupt Dialogue

Refer to Figure 3-2, Figure 3-7A, Bus Request Simplified, and Figure 3-7B, BR Timing. The signal BR REQUEST from the Status register causes the Unibus control logic to issue a BR to the PDP-11 processor. Refer to the *PDP-11 Peripherals Handbook* (EB 05961 76), Chapter 5, for a complete discussion of the interrupt dialogue.

The major difference in the implementation of instructions versus interrupts, as it applies to the bus dialogue, is in terms of the use made of the Controller Select lines. The device select codes available for the DTE20s are shown on Figure 3-6, at the right margin. The first interface would be assigned the code of 200, the next 204 and so forth. The physical number assignments are only used during interrupts. Assuming the KL10 is issuing a CONI to the DTE20, the dialogue is as follows:

The EBox asserts the following:

CS(00-06); Device Code to Select Device
F(00-02); Function is CONI
DEMAND; Activate Decoders.

The interface, upon detecting DEMAND, decodes the device select code as the appropriate DTE20 and the function as CONI. Refer to Figure 3-6. The combination of the decoded device select and function (CONI) enables E-Mixer Select 1 only. This selects those relevant bits from the Status register and enables them onto the EBus. The table at the top of Figure 3-4, (Sheet 2 of 2), indicates that bits 20-22, 26, 27, and 29 through 35 are enabled.

An important point to remember here is that the vector interrupts (KL10 device code and PDP-11 address assignments) are controlled by back panel wiring. No jumpers are used to differentiate them from slot to slot.

Each interface transfers to its preassigned vector address within PDP-11 memory. Refer to Table 3-2 for the various possibilities. After having received the interrupt line, the PDP-11 will assert SSYN. The interface, upon receiving SSYN, will drop BBSY as well as the vector address and interrupt line. This completes the dialogue.

Table 3-2
Vector Addresses

Interface No.	Vector Interrupt	PDP-11 Device Register
0	774	774400
1	770	774440
2	764	774500
3	760	774540

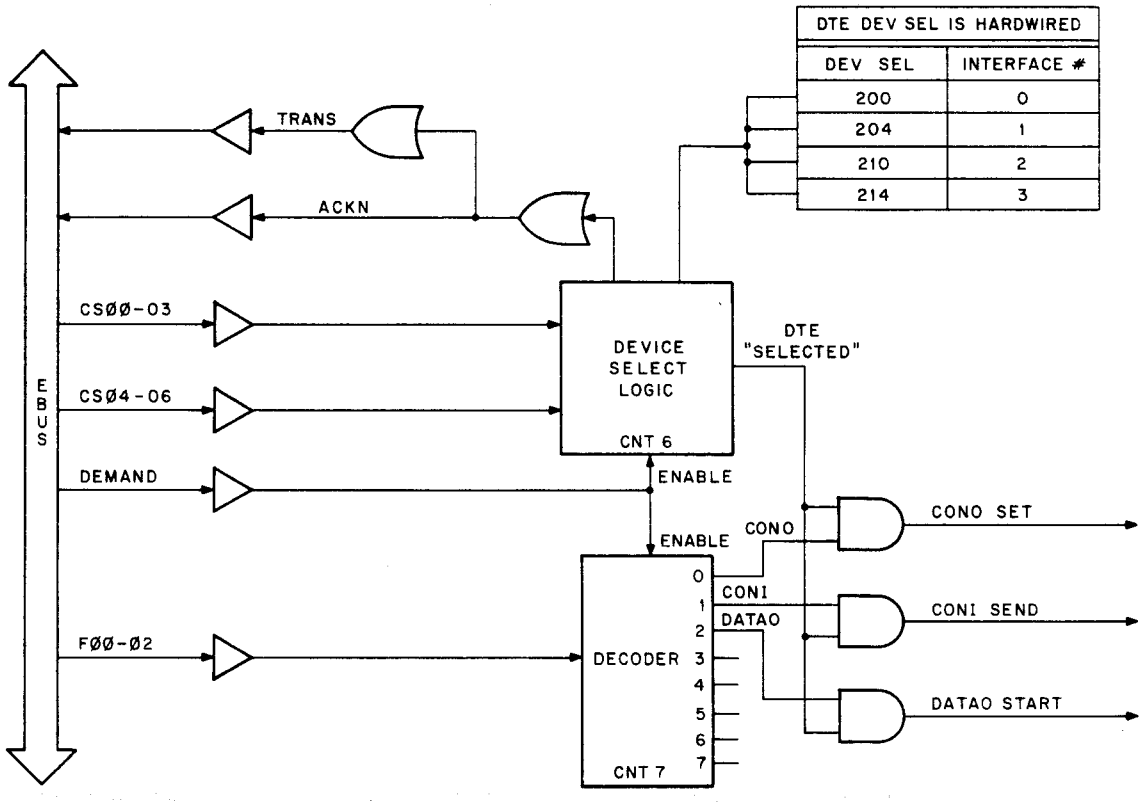
3.3.5 Doorbell Summary

In negotiating a transfer between the PDP-11 and the KL10, the doorbell is used to set up the necessary communications between the two processors. One of the operations that is performed by the PDP-11, with respect to the KL10, is to read information from a predefined area in the KL10 main memory. It is this information that establishes how the PDP-11 operating system proceeds during various types of transfers. The block of information contains such items as pointers to sources of data or control information, destination addresses, and flags.

3.4 INTERFACE STATUS

3.4.1 General Information

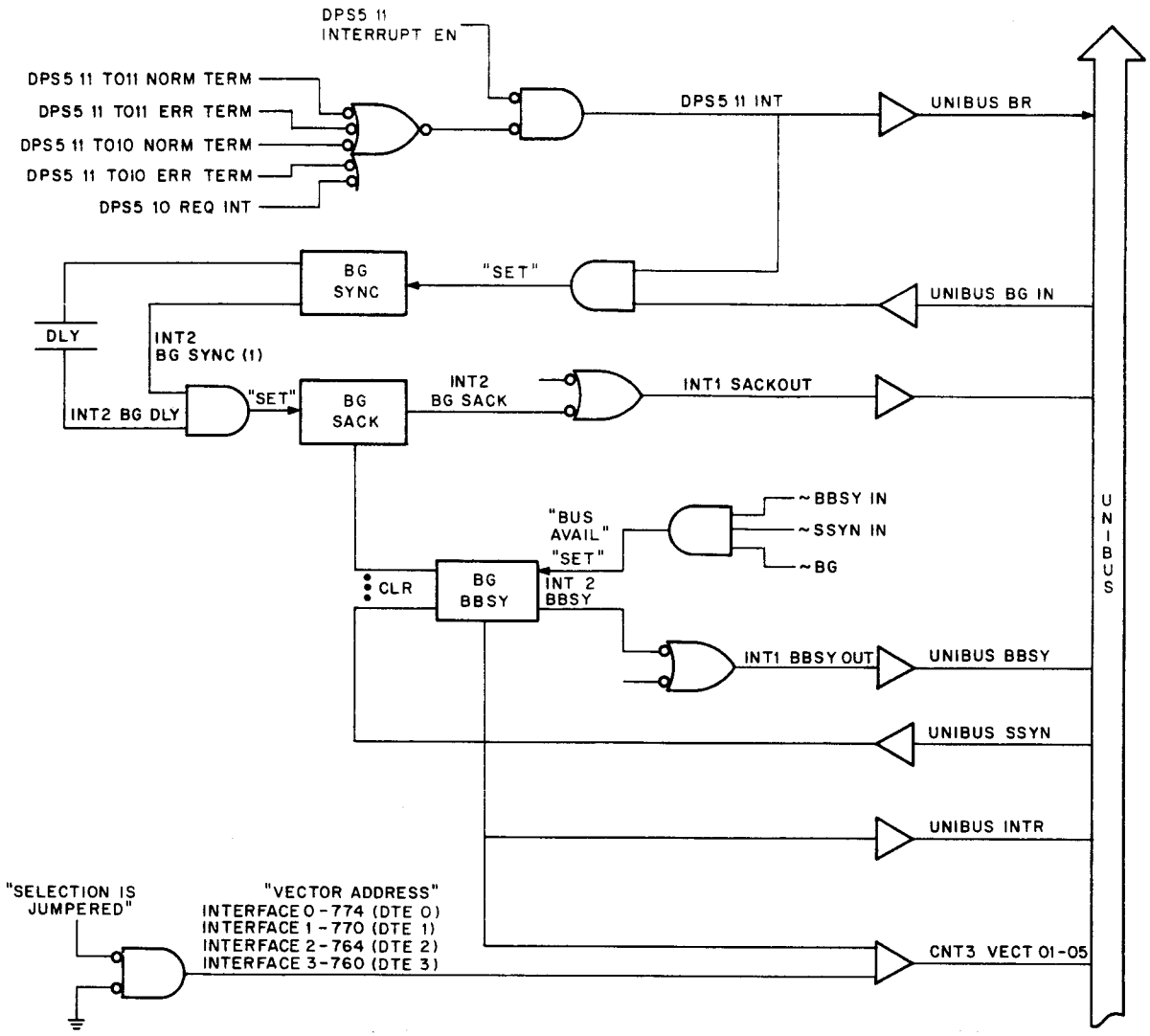
Figures 3-8 and 3-9 indicate the bit configurations for the CONO and CONI instructions, respectively. Tables 1-13 and 1-14 list the purpose of each bit. Similarly, Figures 3-10 and 3-11 indicate the PDP-11 DATO and DATI bit configurations. The 10-11 Interface Status register plays an important role in most, if not all, interface bus operations.



FCN	SIGNAL	BASIC ACTION
CONO	CONO SET	TOGETHER WITH APPROPRIATE EBUS BITS PERFORM THE FUNCTIONS INDICATED IN TABLE 1-14
CONI	CONI SEND	CAUSES THE INFORMATION INDICATED IN TABLE 1-13 TO BE PLACED ONTO THE EBUS
DATAO	DATAO START	TOGETHER WITH APPROPRIATE EBUS BITS, PERFORM THE FUNCTIONS INDICATED IN TABLE 1-12

10-1827

Figure 3-6 KL10 Instruction Dialogue



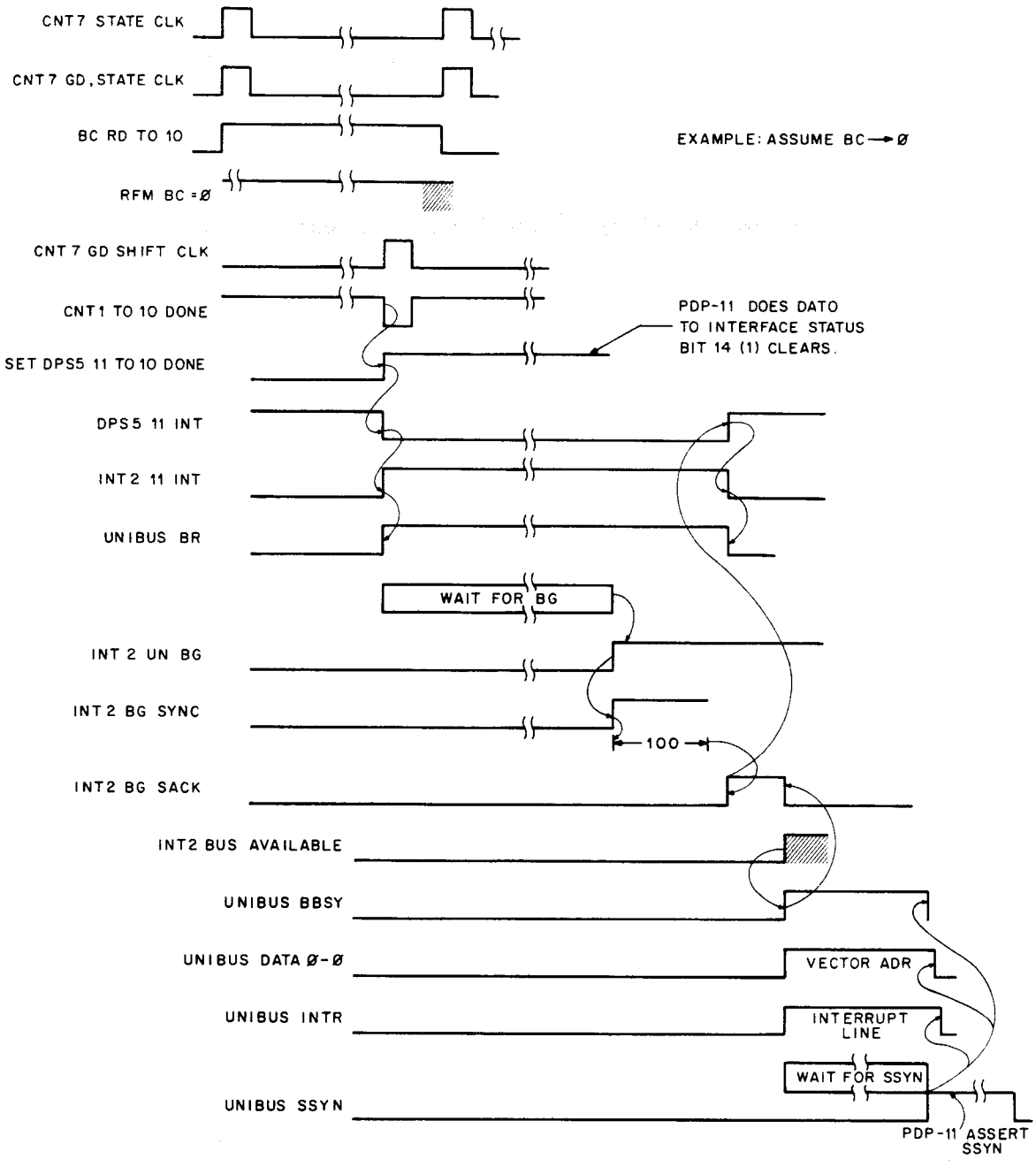
TYPE OF BR	REASON FOR BT
DPS5 11 TO11 ERR TERM (1)	CNT2 TO11 ERR SET IS ASSERTED FOR ONE OF THE FOLLOWING REASONS: 1.) AN EBUS PARITY ERROR OCCURRED DURING A TO11 TRANSFER 2.) A UNIBUS ERROR OCCURRED DURING A NPR TRANSFER DURING A TO11 TRANSFER
DPS5 11 TO11 NORM TERM (1)	CNT1 TO11 DONE SET IS ASSERTED FOR ONE OF THE FOLLOWING REASONS: 1.) BC=0 AND A TO11 BYTE COUNT READ WAS PERFORMED 2.) TO11 NULL STOP WAS TRUE AND RAM=0 DURING A TO11 FILE READ OPERATION DPS5 NULL STOP SETS AS A RESULT

TYPE OF BR	REASON FOR BT
DPS5 11 TO10 ERR TERM (1)	CNT2 TO10 ERR SET IS ASSERTED FOR ONE OF THE FOLLOWING REASONS: 1.) A PDP-11 MEMORY PARITY ERROR OCCURRED DURING AN NPR TRANSFER 2.) A UNIBUS ERROR OCCURRED DURING A TO10 TRANSFER
DPS5 11 TO10 NORM TERM (1) CNT5 TO11 I BIT (1) THIS BIT MUST HAVE BEEN SET PRIOR TO THE TO10 TRANSFER	CNT1 TO10 DONE SET IS ASSERTED FOR THE FOLLOWING REASON. 1.) BC=0 AND A TO10 BYTE COUNT READ WAS PERFORMED
DPS5 10 REQ INT (1) (DOOR BELL)	SET BY CONO DTE STATUS WITH BIT 22 (1) MUST BE CLEARED BY PDP-11 WITH DATO AND UNIBUS BIT 10(0)

10-1828

A. Bus Request Simplified

Figure 3-7 Bus Request (Sheet 1 of 2)



B. BR Timing

Figure 3-7 Bus Request (Sheet 2 of 2)

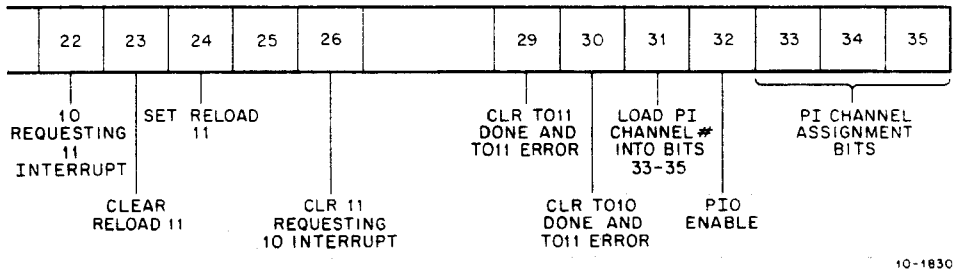


Figure 3-8 CONO Interface Bit Assignments

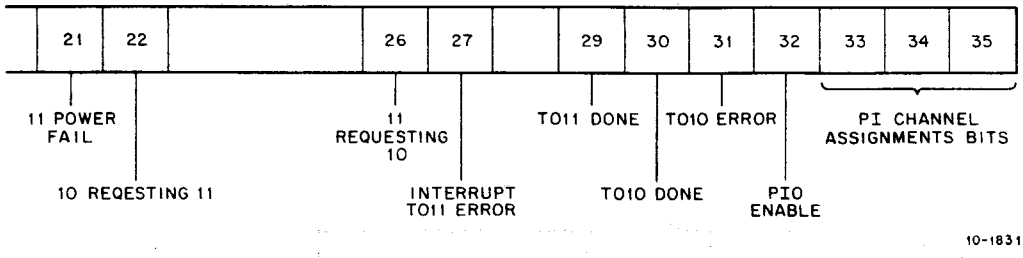


Figure 3-9 CONI Interface Bit Assignments

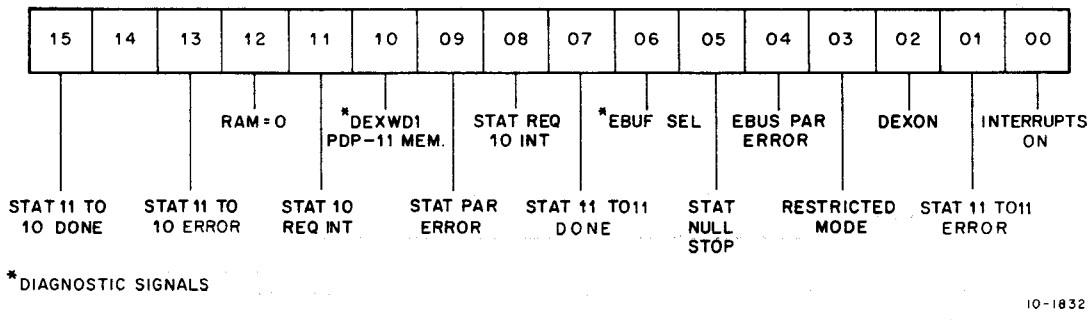


Figure 3-10 PDP-11 Status Word - DATI Configuration

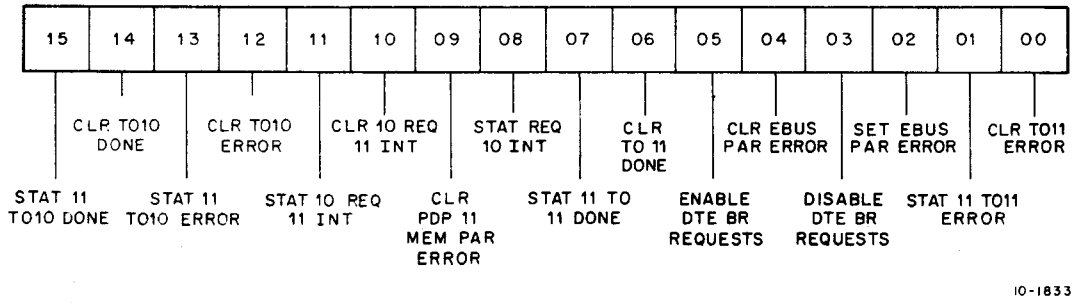


Figure 3-11 PDP-11 Status Word - DATO Configuration

For example, to implement the interprocessor doorbell feature, one of the two processors must execute an instruction that sets the Doorbell flag in the Status register (CONO from the EBox, and DATO from the PDP-11). Refer to Table 3-1, Section 3.3. Upon completion of either a TO10 transfer or TO11 transfer, the hardware sets appropriate Status flags to generate error or normal termination interrupts to one or both processors when done. Usually, one of the processors is unconditionally interrupted, and if an additional flag (I Bit) is set, the second processor is also interrupted. The reasons for the interrupts will become clear with some background information on just how the TO11 and TO10 transfers are set up.

3.4.2 STAT 10 TO10 DONE and I Bit

In the case of TO10 transfer, the KL10 loads a negative byte count into the RAM FILE in a predefined location (Figure 3-12). The PDP-11 must now load, or have loaded previously, the TO10 address into a similarly predefined RAM location. Upon detection by the interface that both the byte count and TO10-11 address have been loaded, the transfer starts. The interface reads from the RAM and increments the byte count, and checks to see if the result is zero. If the result is zero, no data is transferred and a flag in the Status register DPS5 10 TO10 DONE is set, causing an interrupt to the EBox. Figure 3-13 shows those Status flags that will cause 10 interrupts. If the byte count after incrementation was not equal to zero, the 10-11 Interface reads a byte of data pointed to by the TO10 – PDP-11 addresses using the NPR facility. It then updates this address by one or by two, depending on whether one or two bytes per PDP-11 word are being transferred. Next, the interface interrupts the EBox, carries on some dialogue, and transmits to the API FUNCTION 6. This informs the EBox that it must take the byte of data. When carrying out continuous transfers, each time the byte count becomes zero, the EBox must supply the interface with a new byte count before transfers can continue. An optional bit (the DPS5 10 I Bit) enables using multiple byte counts between transfers. When a transfer is complete and DPS5 10 TO10 DONE is asserted, the PDP-11 will be interrupted if DPS5 10 I Bit is also true. In this case, it is necessary for the byte count as well as the TO10 PDP-11 address to be reloaded. In addition, the TO10 Byte Count Load flag and the TO10 Address Load flag will be cleared.

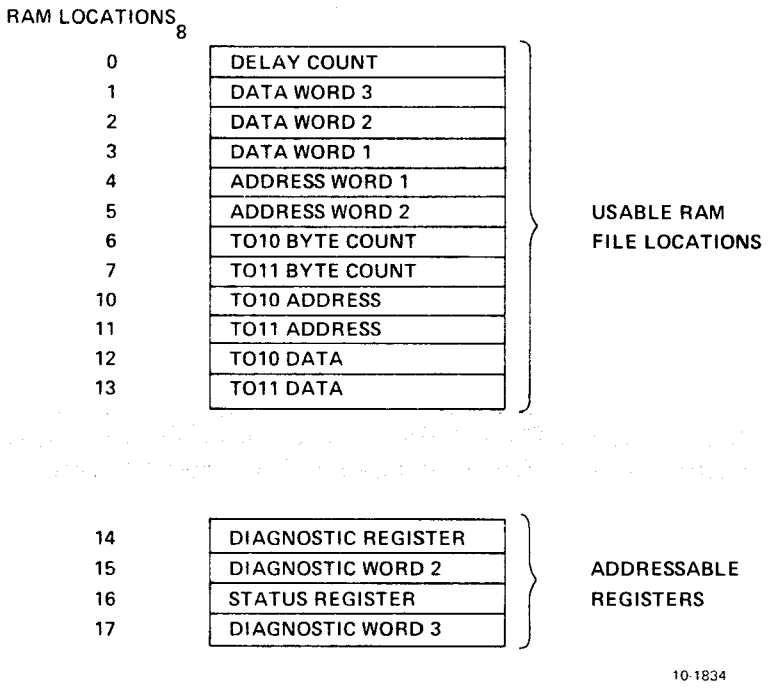


Figure 3-12 RAM Words and Registers

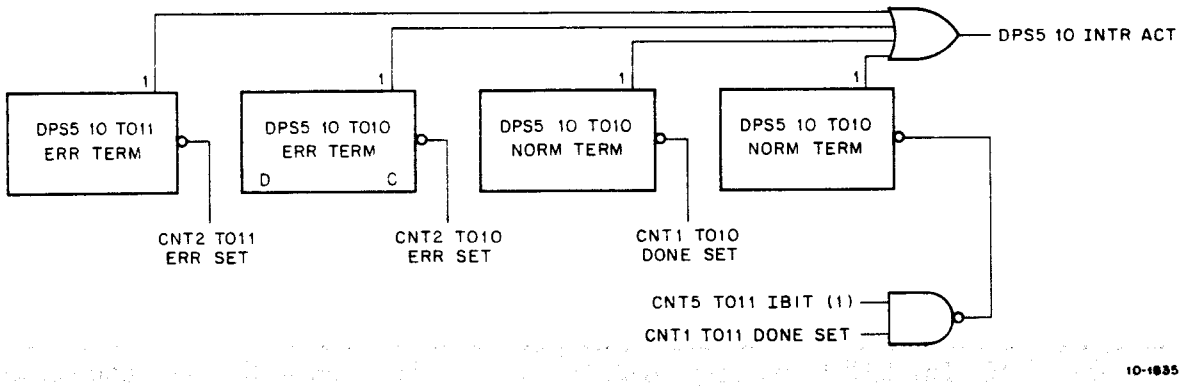


Figure 3-13 Hardware-Generated 10 Interrupts Simplified

3.4.2.1 TO10 Byte Transfers

Basically, TO10 transfers are performed in the following manner. Flow diagram Figure 3-14 complements the outlined procedure:

1. The PDP-11 writes the Delay Count word (DLYCNT) with the negative number of 500 ns intervals to delay between byte transfers across the DTE in bits 13-00. Bits 15-14 specify Unibus address bits 17-16 for both TO10 and TO11 byte transfers. The DLYCNT (15-00) may be written at system startup, because it is never reset by the hardware.

2. The PDP-11 writes address of source string [TO10AD (15-00)].

3. The PDP-11 sets the PDP-11 byte or word mode bit (DIAG3 TO10MB).

4. The KL10 allocates core to receive the string.

5. The KL10 sets up the byte pointer in EPT to receive the data.

6. The KL10 sets negative byte count and indicates that both CPUs are to receive the normal termination interrupt. This starts the transfer.

7. The DTE interface and EBox work as shown in Figure 3-14 during the transfer.

3.4.3 STAT 11 TO11 DONE and I Bit

In the case of a TO11 transfer, both the negative byte count and TO11 PDP-11 address are loaded into the specified RAM FILE locations (Figure 3-12) before the transfer can take place. Again, upon detecting that both the byte count and TO11 PDP-11 address have been loaded, the interface starts up. The byte count is incremented and checked to determine if it is equal to zero.

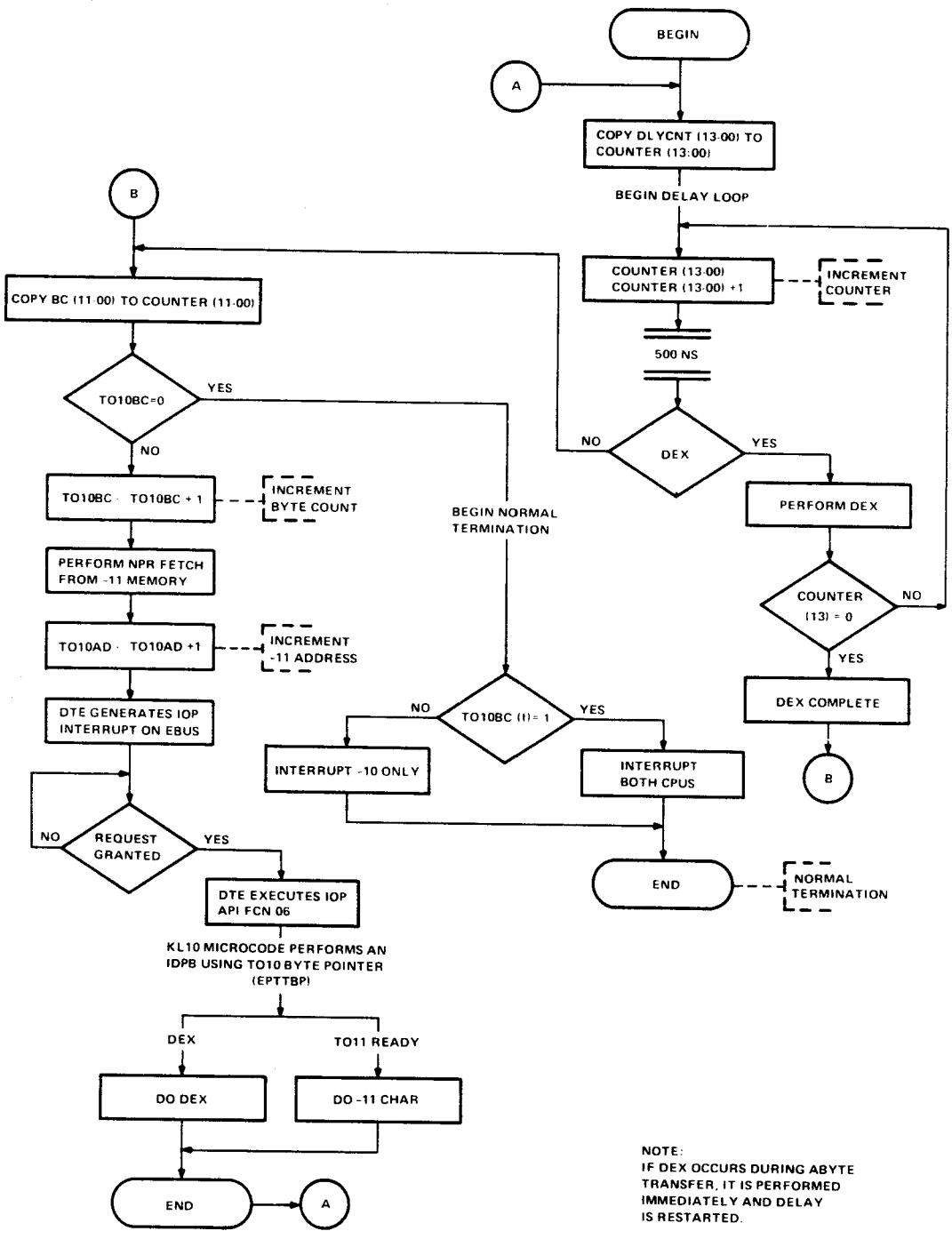


Figure 3-14 TO10 Byte Transfer Flow Diagram

If the result is zero, no data is transferred and a flag in the Status register (DPS5 11 TO11 DONE) is set, causing an interrupt to the PDP-11. Figure 3-16 shows those Status flags, which, when set by hardware conditions, will cause PDP-11 interrupts. If the byte count after incrementation was not equal to zero, the interface interrupts the EBox for a byte of data, using a hardware dialogue and API FUNCTION 6. The interface next performs an NPR to PDP-11 memory and updates the TO11 address by one or two depending on whether one or two bytes per word are being transferred. When carrying out continuous transfers, each time the byte count becomes zero, the PDP-11 must supply a new byte count and possibly a new TO11 PDP-11 address. When a transfer is complete and DPS5 11 TO11 DONE is asserted, the EBox will also be interrupted if CNT5 TO11 I Bit is true. Refer to Figures 3-15 and 3-16. Upon receiving interrupts, the corresponding processor must execute the appropriate instruction to remove the interrupt by clearing the flag. In the case of DPS5 10 TO10 DONE (Figure 3-15), the EBox performs a CONO to the Status register with bit 30(1). This will clear DPS5 10 TO10 NORM TERM and remove the interrupt. Similarly, to remove the PDP-11 interrupt caused by DPS5 11 TO11 DONE, the PDP-11 performs a DATO to the Status register with Unibus bit 6(1) and 7(0), to clear DPS5 11 TO11 DONE.

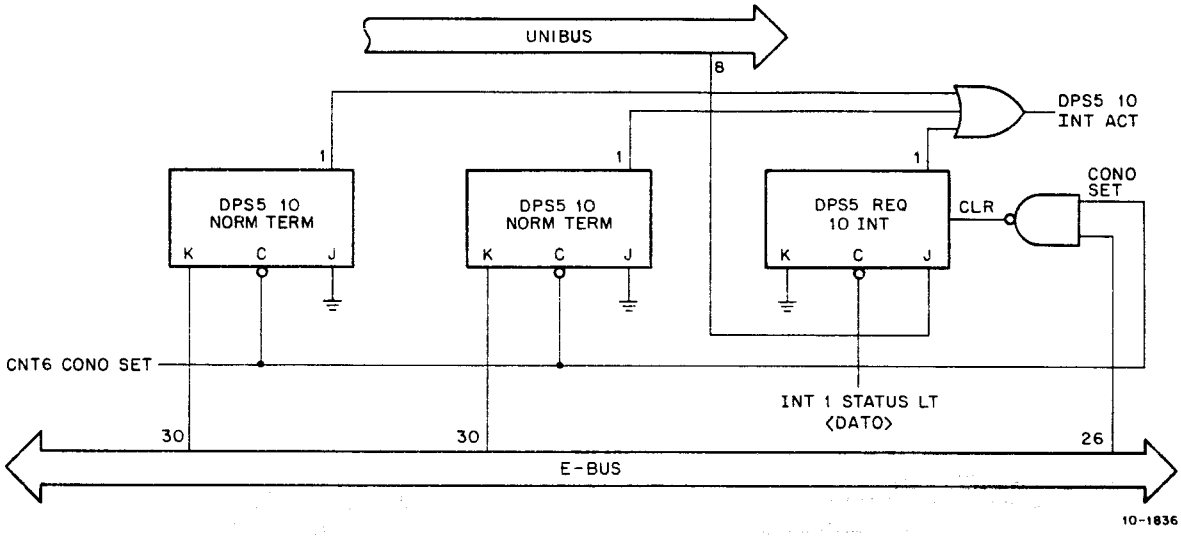


Figure 3-15 Program Generation and Clearing of 10 Interrupts Simplified

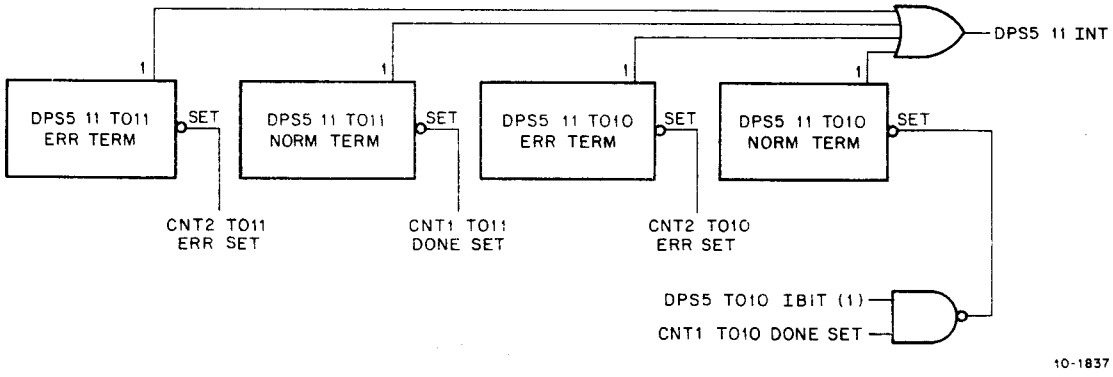


Figure 3-16 Hardware-Generated 11 Interrupts Simplified

3.4.3.1 TO11 Byte Transfers – Basically, TO11 transfers are performed in the following manner. Flow diagram Figure 3-17 complements the outlined procedure:

1. The PDP-11 writes the Delay Count word (DLYCNT) with the negative number of 500 ns intervals to delay between byte transfers across the DTE in bits 13-00. Bits 15-14 specify Unibus address bits 17-16 for both TO10 and TO11 byte transfers. The DLYCNT (15-00) may be written at system startup, because it is never reset by the hardware.

2. The KL10 writes address of source string (byte pointer in EPTEBP).

3. The PDP-11 allocates core to receive string.

4. The PDP-11 writes the TO11 receive data [TO11AD (15-00)].

5. The PDP-11 writes the TO11 Byte Count word (TO11BC), which sets the following:

   a. Negative byte count [TO11BC (11-00)]

   b. Indicates that both CPUs are to receive the normal termination interrupt [TO11BC (INT10) = 1]

   c. Whether termination is also to occur on transfer of a NULL character [TO11BC (ZSTOP)], PDP-11 byte or word mode [TO11BC (TO11BM)]

   d. Writing TO11BC starts the transfer, providing TO11AD has also been written since the last normal or error termination.

6. The DTE interface and EBox work as shown in Figure 3-17 during the transfer.

3.4.4 Status Error Conditions

Three error types can cause a transfer error condition that can be detected by the hardware. They are:

1. Unibus timeout errors
2. PDP-11 memory parity errors
3. Ebus parity errors.

All of these errors set a general error bit [status (TO10ER) or status (TO11ER)]. Also, all of the errors except Unibus timeout errors have an additional error bit. A bus timeout is most likely caused by an NXM (non-existent memory error) reference, although it can be caused by a variety of hardware problems.

3.4.5 NULL Stop

The DTE interface can automatically terminate TO11 transfers when a NULL character is detected. This facilitates the easy handling of string transfers that terminate in a NULL character (e.g., ASCII, MSG). When the DTE extracts the previously loaded byte from the RAM FILE TO11 data slot, it is checked for a NULL character. If it is found, DPS5 NULL STOP and TO11 DONE are both set, generating a bus request (BR) to the PDP-11. The PDP-11 then reads the DTS Status register, finds the NULL bit set and performs a DATO with bit 06 set. This clears the NULL STOP flag.

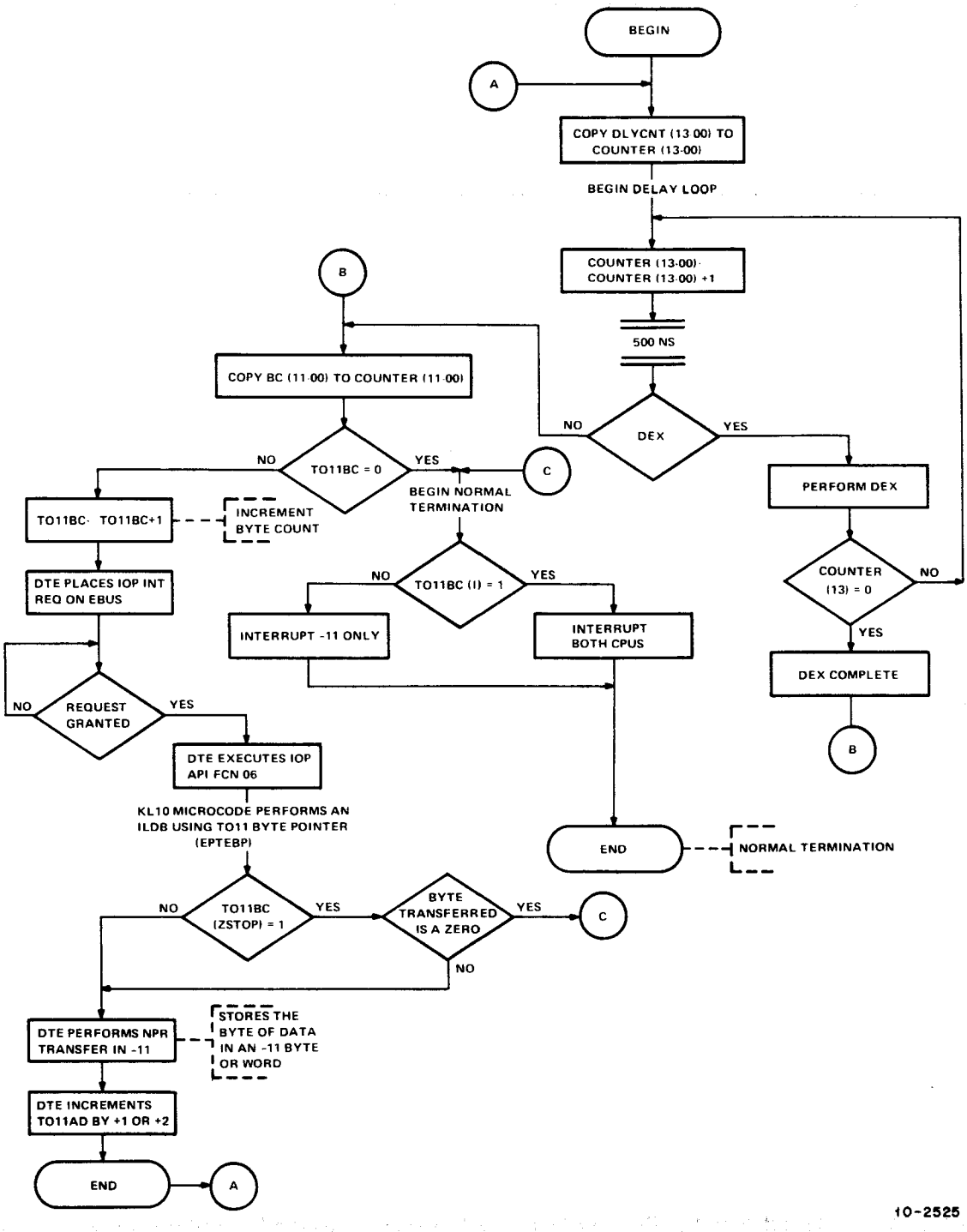


Figure 3-17 TO11 Byte Transfer Flow Diagram

3.5 RAM OPERATIONS OVERVIEW

To perform any of the three basic interface operations, the PDP-11 processor must write control and/or data information into predefined locations in the RAM. Also, for some of these operations, the EBox must supply a portion of that information. Table 1-8 contains a list of all of the RAM words, together with an explanation of their basic functions and the PDP-11 address that addresses that location in the DTE interface.

Table 3-3 gives a synopsis of the four types of interface operations. For each operation, it provides the following information:

PDP-11 Addresses Generated
RAM Addresses Generated
Name of Each RAM Location Used.

Table 3-3
Generalized RAM Storage

PDP-11 Address	RAM Location	Functional Name	Operation
2, 4, 6, 10, 12	1, 2, 3, 4, 5	Deposit or Examine Word 3, 2, 1 Ten Address Word 1, 2.	Examine
2, 4, 6, 10, 12	1, 2, 3, 4, 5	Deposit or Examine Word 3, 2, 1 Ten Address Word 1, 2.	Deposit
0, 14, 20, 24	0, 6, 10, 12	Delay Count, TO10 Byte Count, TO10 11 Address, TO10 Data.	TO10 Transfer
0, 16, 22, 26	0, 7, 11, 13	Delay Count, TO11 Byte Count, TO11 11 Address, TO11 Data.	TO11 Transfer

3.5.1 RAM Access and Control

Refer to Figure 3-19, Interface Address and Access Control Simplified. The interface address and access control consists of the RAM, its input and output mixers, Address Selection Logic, input address decoding, outputs to the Data Control register logic, and those timing elements necessary to implement RAM read or write operations.

The RAM is addressed via the Unibus, using the high-order four of the least significant five PDP-11 address lines A(04:01). All incoming Unibus addresses are checked to determine if they are interface addresses and if they are RAM addresses. The interface also contains four addressable registers (Section 3.1.5). When the address check detects an interface address and it is a RAM address, RAM Cycle Sync sets, enabling a RAM cycle to take place (Table 3-4). During the period of time that the Unibus address lines are asserted, decoders provide discrete levels to the Data Control register flip-flops. Three pairs of flip-flops synchronize the start of one of the interface operations. The select lines and Data Control register flip-flops are listed in Table 3-5.

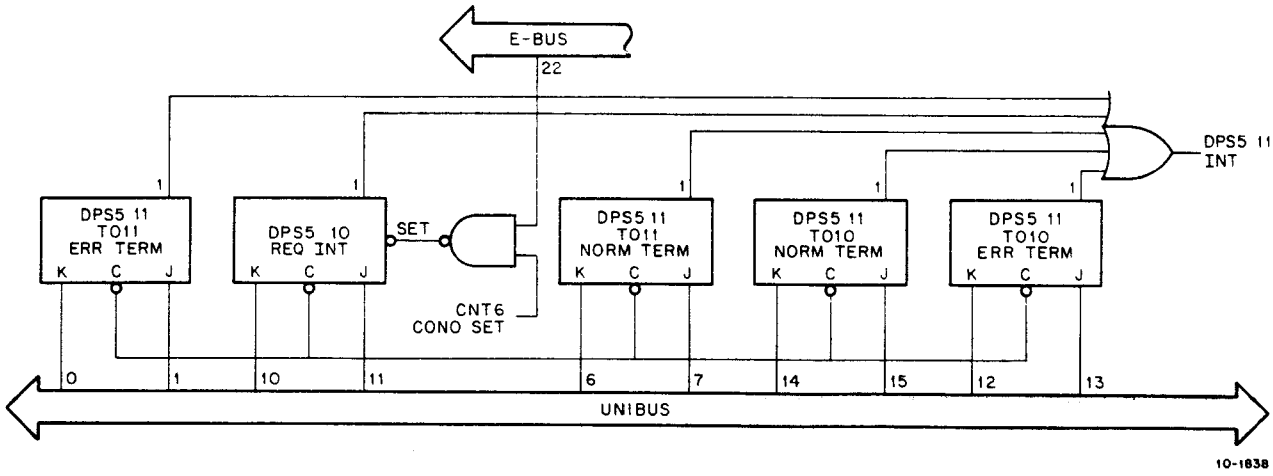


Figure 3-18 Program Generation and Clearing of 11 Interrupts Simplified

Table 3-4
RAM Cycle Functions

RAM Cycle Sync	RAM Cycle	Function
0	0	IDLE
1	0	Synchronizing a request
1	1	Reading or writing in RAM
0	1	Return SSYN

Table 3-5
Select Lines and Data Control

Interface Operation	CNT 2 Select Level	CNT5 Data Register Control FF
DEX	SELECT ADR1	CNT5 DEP Unibus bit 12(1)
DEX	SELECT ADR2	CNT5 DEX START
TO11 TRANSFER	SELECT TO11 BC	CNT5 TO11 BC LD
TO11 TRANSFER	SELECT TO11 ADR	CNT5 TO11 ADR LD
TO10 TRANSFER	SELECT TO10 ADR	CNT5 TO10 ADR LD

NOTE

TO10BC is set as a result of DATAO from the EBox. This word may also be referenced by the PDP-11 for verification in case of error. However, it is not normally loaded via the PDP-11 processor.

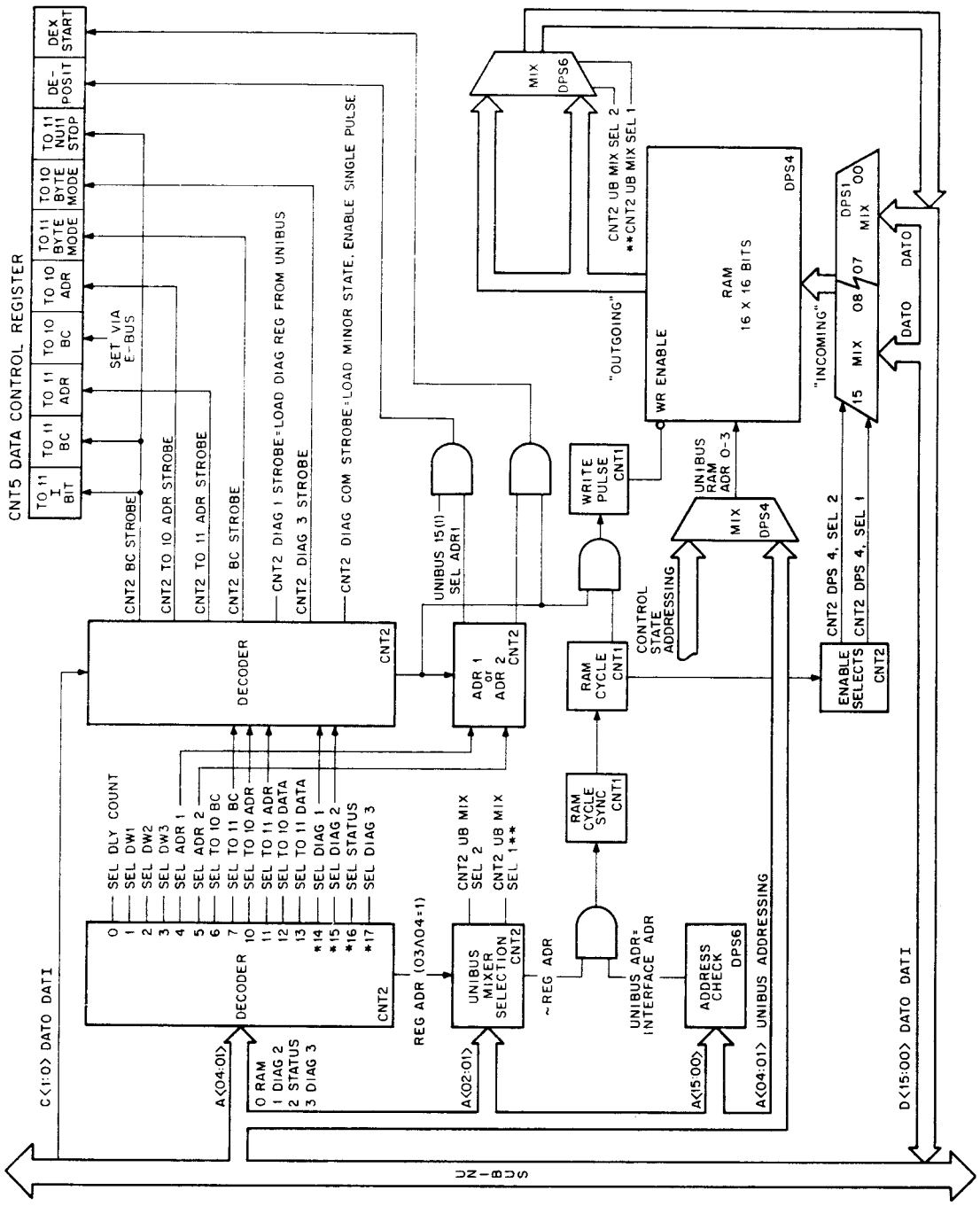


Figure 3-19 Interface Address and Access Control Simplified

In addition, four other Data Control register flip-flops (generic term) are used during 10-11 interface operations:

1. TO11 I Bit - This bit enables the interface to interrupt the EBox upon completion of the TO11 transfer operation.

2. TO11 NULL STOP - This bit enables the interface to stop the transfer upon detecting a NULL character, and generates an interrupt to the PDP-11.

3.-4. TO11 BYTE MODE, TO10 BYTE MODE - These bits define the byte transfers as consisting of 8-bit or 16-bit bytes. TO11 BYTE MODE is set with Unibus bit 13(1) and CNT2 BC STRB. TO10 BYTE MODE is set with Unibus bit 0(1) and CNT2 DIAG 3 STRB.

3.5.2 Write Access Overview

During DATO operations, the RAM FILE is input from the Unibus via the RAM input mixers. The RAM cycle assures that the RFM SEL 1 and 2 mixer select lines allow the input to come directly from the Unibus D(15:00). Because the operation is DATO, the RAM cycle enables a WRITE pulse at the appropriate time, to write the word from the Unibus into the selected RAM location.

3.5.3 Read Access Overview

During DATI operations, the RAM FILE output is enabled to the Unibus via the UB mixers. The addressing and selection is the same as for DATO except that no WRITE pulse is generated. Whenever a RAM address is decoded in the Access Control Logic, the UB mix SEL 1, 2 select lines allow the addressed locations contents to the Unibus.

3.5.4 Write Access Timing

Refer to Figure 3-20, DATO to RAM LOCATION 1. This timing diagram shows the DATO operation beginning with the assertion of the Unibus control and address until the 10-11 Interface generates SSYN, and releases the bus.

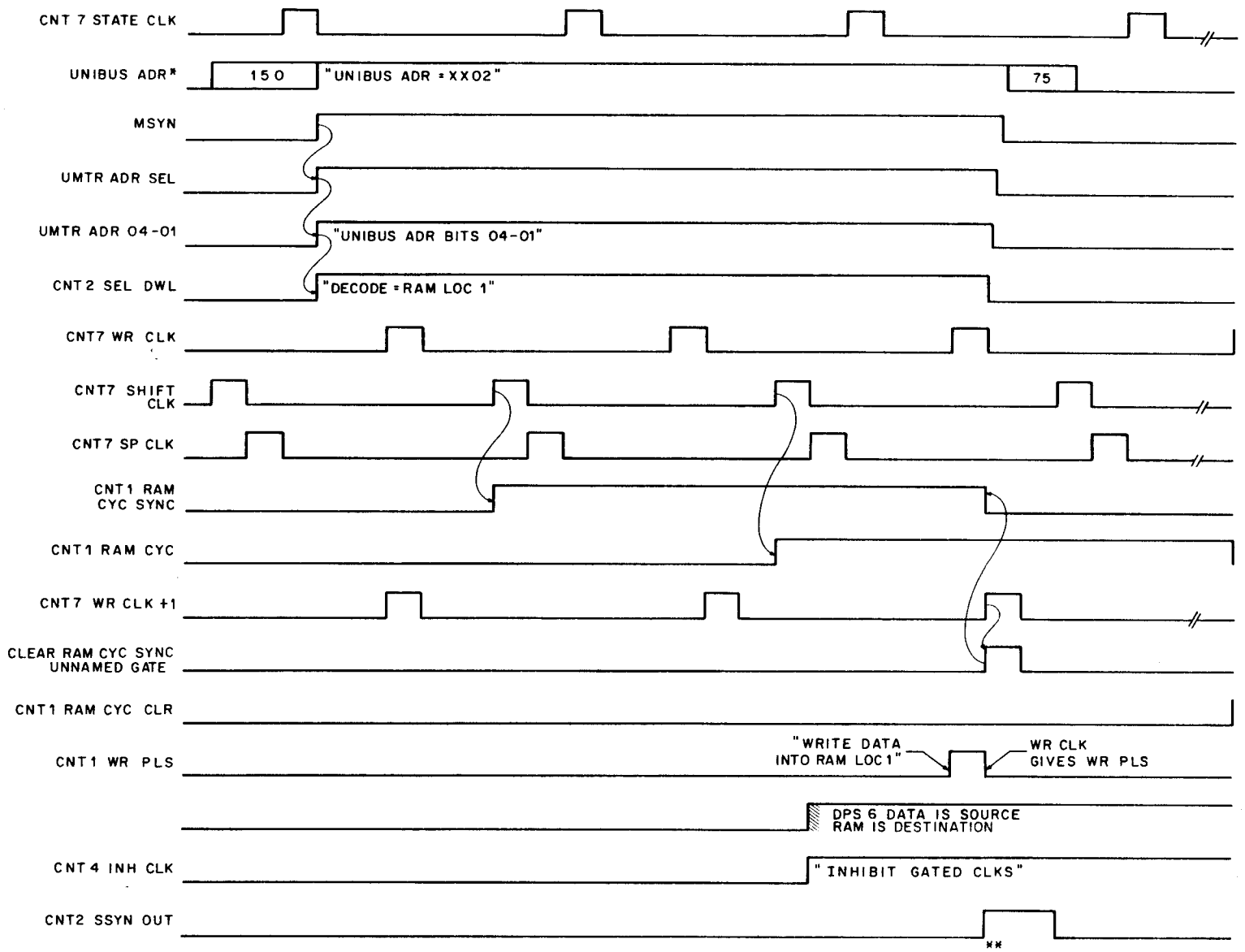
The PDP-11 asserts the following:

A(17:00) = XXXX02
C(1:0) = DATO
D(15:00) = DATA.

It waits approximately 150 ns, and asserts MSYN. At the interface, the address is checked to determine if it is a valid interface address. When this is true, DPS1 ADR SEL is asserted. The low-order five bits with bit 15 stripped off will be decoded as CNT2 DW1, i.e., A(04:01). The current shift clock sets CNT1 RAM CYCLE SYNC, which enables the next shift clock to set RAM cycle. RAM cycle enables INHIBIT CLOCK to set on the next SP clock. This is necessary to prevent any gated clocks from effecting the state of the interface. In addition, RAM CYCLE (1), selects the proper mixer combination (RFM SEL 1) for the DATA on Unibus D(15:00) to be admitted to the RAM. During the cycle, A(17:00) decodes to the proper RAM location. The 10-11 Interface always strips bit 15 from the incoming address, thus the address 02 becomes 01. RAM CYCLE (1) enables the WR clock to write the DATA into the RAM. WR clock +1 removes CNT1 RAM CYCLE SYNC and asserts SSYN. Finally, the third WR clock clears RAM cycle and removes SSYN. Sometime later the PDP-11 drops C(1:0) A(17:00) D(15:00).

3.5.5 Read Access Timing

By comparing Figures 3-20 and 3-21 it can be seen that the only difference between a WRITE access to the RAM (DATO) and a READ access from the RAM (DATI) is the generation of CNT1 WR PLS which writes into the RAM. A complete timing diagram for a DATI to the RAM is included as Figure 3-21.



*Unibus address leads MSYN by 150ns min and lags it by 75ns min.

**"Tell PDP-11 to drop MSYN" (device detach from bus)
<DATO to RAM loc 1 from PDP-11>

NOTE
Reflects ECO REV.#6.

10-1839

Figure 3-20 DATO to RAM Location 1 Timing

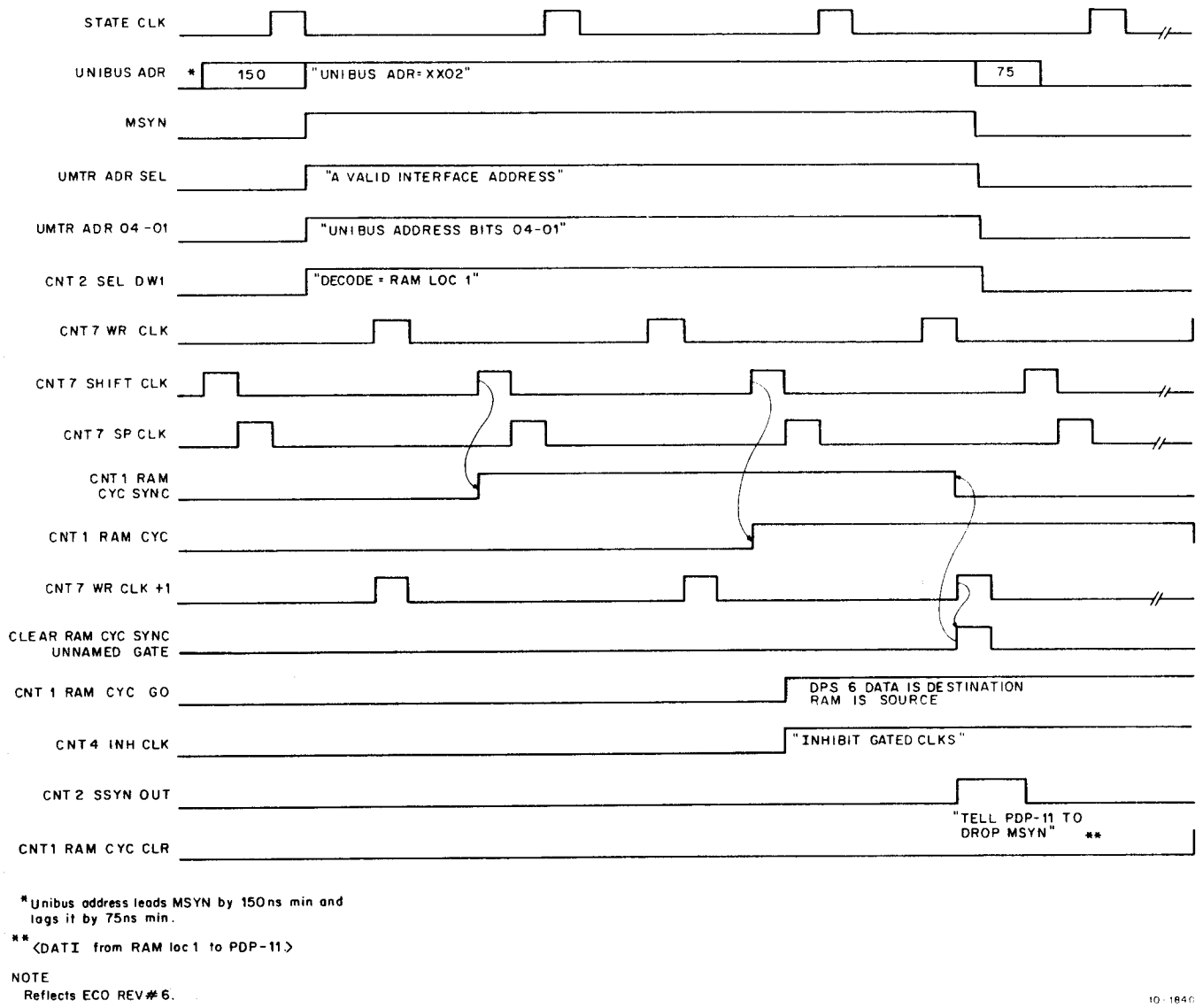


Figure 3-21 DATI to RAM Location 1 Timing

3.6 INTERFACE TIMING AND CONTROL

The 10-11 Interface timing and control section generates all the timing levels necessary to implement all interface operations. It includes the following:

A Major State Generator
A Minor State Generator
A 3-Phase Gated Clock
A 5-Phase Free Running Clock
Miscellaneous Combinational Logic.

3.6.1 Clock and Major State Control

Refer to Figure 3-22. The interface contains an externally driven, free-running clock. This clock has five phases. The first clock to occur after power up is always SP clock, followed by state clock, state clock +1, state clock +2, write clock, write clock -1, write clock +1, shift clock and the cycle repeats. Four of the eight phases will generate gated phases, providing inhibit clock is false. The gated phases are responsible for advancing major and minor states, shifting or loading DATA, or writing into the RAM. The five basic un-gated phases are necessary for synchronization at all events within the interface. Refer to Figure 3-23. The interface contains three major time states. The first to occur is always DEX followed by TO11 transfer which is followed by TO10 transfer. This sequence repeats in a cyclic fashion until STATE HOLD is present. STATE HOLD enables the appropriate time state to be locked into place and enables the minor state decoder to function. The Data Control Logic senses the conditions to lock the major state, and relays those levels to the State Hold Logic.

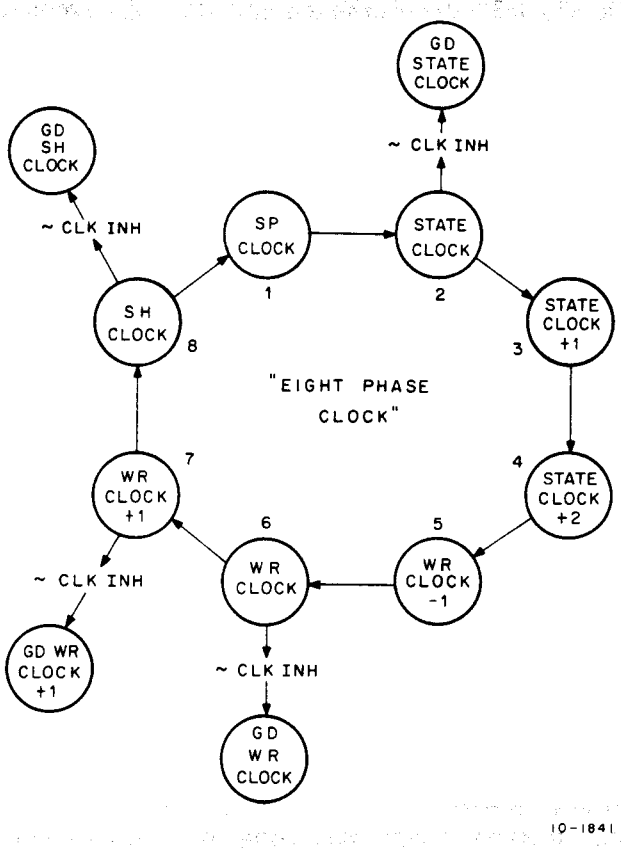


Figure 3-22 Free Running and Gated Clocks

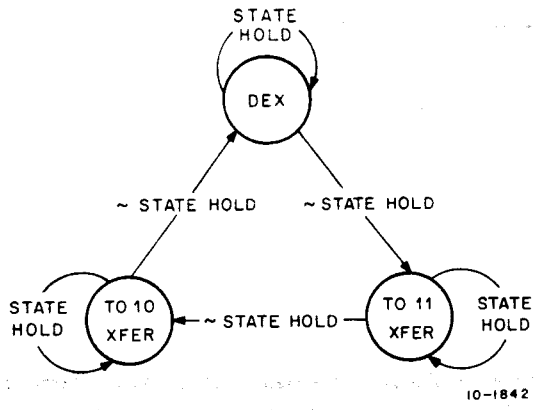


Figure 3-23 Major State Sequencing

3.6.2 Minor State Control and Inhibit Clock

Refer to Figure 3-24. Assume that the necessary conditions are present from data control and register gating (CNT5) at state hold to enable the locking of the TO11 transfer major state. At each GD STATE CLOCK the major state generator changes gates. When it next enters the TO11 transfer major state, this information is passed to the State Hold Logic, together with the input (from CNT5). The next SP clock sets STATE HOLD. This action locks the interface in the TO11 transfer major state until such time as the transfer is completed, provided that no Deposit or Examine operations have been started. If the interface is locked in the TO11 major state and is in the minor state "CNT4 TO11 DLY CNT INC", setting DEX START releases the major state and a Deposit or Examine cycle may be taken upon completion of the DEX operation. TO11 transfer sets once again. Having locked the major time state in place, the minor state generator begins to produce a series of sequential minor timing levels. At each GD STATE CLOCK, the minor state generator advances to the next minor state until INHIBIT CLOCK is asserted. This mechanism allows stopping the GD CLOCKS and locking the appropriate minor state. The INHIBIT CLOCK function is used during interrupts to the EBox and during BR or NPR, transfers to the PDP-11, as well as RAM cycles, and for certain diagnostic functions. It facilitates interlocked operation between the interface and the EBox or PDP-11. A more detailed diagram at the minor state decoding is shown on the DTE20 control drawing, Figure 3-2 and a detailed timing diagram of the clock is shown on Figure 3-25, Basic Clock Timing.

3.7 CONTROL STATE TRANSFER PDP-11

Refer to Figure 3-2. During both the TO11 and TO10 major states, data transfers are performed to or from PDP-11 memory. The NPR facility allows this to happen with a minimum of effort on the part of the PDP-11 processor. The abbreviated TO11 transfer sequence is:

1. Read and update delay count (from RAM).

NOTE
DEX cycle stealing is allowed at this time.

2. Read and update byte count (from RAM).
3. Send interrupt to EBox for a byte of data.
4. Read TO11 PDP-11 address from RAM.
5. Adjust the byte of data for storage.
6. Store byte temporarily in RAM.
7. Initiate NPR transfer to PDP-11 memory, using the TO11 PDP-11 address.
8. Update the TO11 PDP-11 address as appropriate.

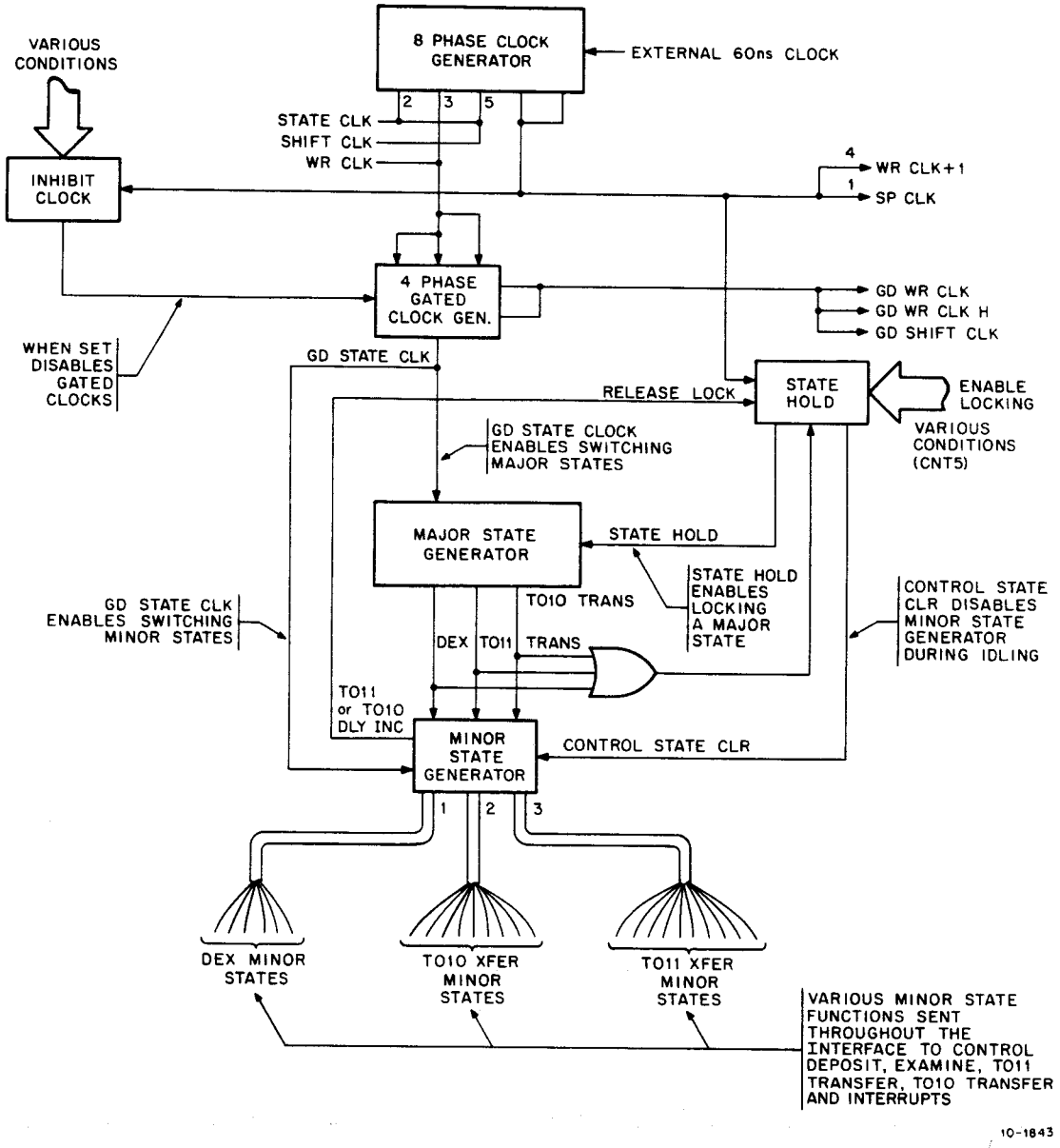


Figure 3-24 Basic Clock and State Control Simplified

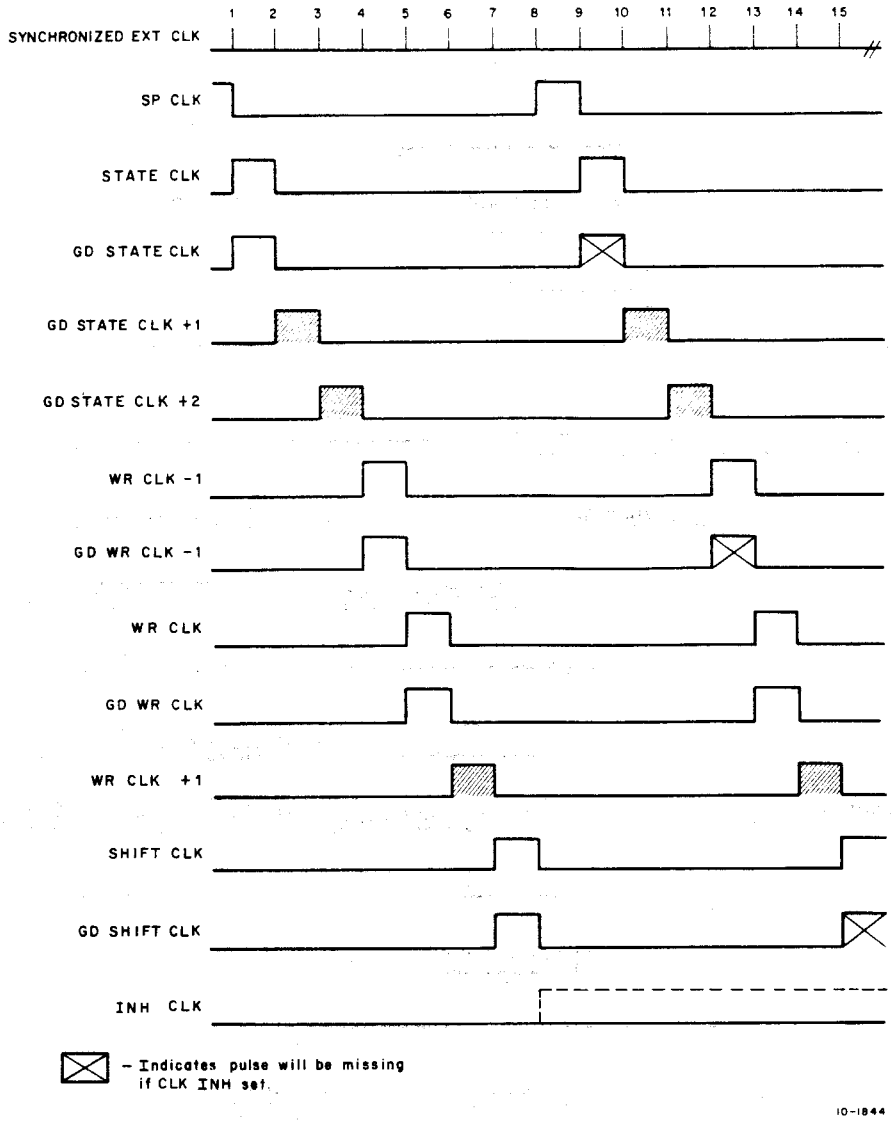


Figure 3-25 Basic Clock Timing

Step 7 above occurs during the "TO11 FILE READ" minor state. The signal CNT4 REQUEST indicates that it is a control state byte of request. During this transfer, the GD CLOCKS are inhibited. The NPR/NPG combinational logic, NPG clock and BG/NPG Common Sequence Logic as well as Bus Complete Logic is used. Refer to Figure 3-1. The source of the Unibus address lines, A(17:00) during the NPR transfer is the ABC register. Only when the DTE20 has gained control of the Unibus is this address placed on the Unibus address lines, as indicated by the gating signal NPR BBSY. In the case of a TO11 transfer, the RAM is the data source. It is selected in the Unibus mixers by a select code of 00 in U-B MIX SEL 2 and 1. Direct control over the U-B MIXER SELECTS comes from the access control, and is merely a function of the current address lines. The default case is U-B MIX SEL 2(0) and U-B MIX SEL 1(0). Refer to Figure 3-26, NPR Sequence. The NPR transfer is started when CNT4 REQUEST is asserted. This places NPR on the Unibus and enables the DTE20 to look for a response (NPG). Receiving NPG sets NPG SYNC and 100 ns later, the DTE20 asserts NPG SACK. If

the PDP-11 processor does not receive SACK within 10 μs of issuing NPG, a timeout occurs and NPG is removed automatically by the PDP-11. Next, BBSY and SSYN are tested and if found negated, BBSY is asserted. NPG BBSY gates the contents of ABC (the address) to the Unibus address lines. NPR BBSY starts up the NPR CLOCK, initially for one pulse. This is necessary to generate MSYN and to enable successive NPR CLOCKS, upon receiving SSYN when the PDP-11 memory operation is completed. The reception of SSYN generates CLK RUN which allows a series of clock pulses to finish the operation. The second clock sets WR RAM, but this function is only useful for a TO10 transfer NPR. In that case, the memory reference would have read a byte and sent it to the DTE20. The WR RAM pulse would then write the data into the selected RAM location. The third clock sets CLR CYC 1, which removes MSYN and enables the fourth clock to set CLR CYC 2. CLR CYC 2 drops BBSY and sets BUS DONE. This allows re-synchronization with the main clock (state clock), which now sets BUS COMPLETE. BUS COMPLETE releases the inhibit clock function and the next minor state is entered. Finally, removing the inhibit clock also removes BUS DONE and BUS COMPLETE. The NPR operation during a TO10 transfer works in an almost identical manner. The impetus is from "TO10 FILE WRITE" minor state, which generates CLKS REQUEST. This triggers the NPR sequence. The main difference is that the byte is read from PDP-11 memory and then sent to the DTE20 to be written into the appropriate RAM address.

3.8 DTE DIAGNOSTIC MODE OPERATION

In the third mode, (MODE 3), the interface is controlled by a diagnostic program resident in the PDP-11 processor. In this mode the interface can be single stepped through its internal timing sequence, or placed into a time state and locked there for some period of time as controlled by the diagnostic program. In KL10 diagnostic mode, the interface can be set up from the PDP-11 to transmit any of up to 128 functions over the diagnostic bus to the EBox in the KL10, where the appropriate function will be performed in response to the function code sent via the interface.

In normal use,the PDP-11 can address the diagnostic section over the Unibus by using DATO and DATI instructions and in this way exercise the interface logic and the EBox as well. Also the EBus can be made to loop back upon itself. Thus, the EBus logic can be effectively checked out within the KL10. Refer to Figure 3-27.

3.8.1 Diagnosing the 10-11 Interface

The interface contains many features that enable diagnosing of the interface. It can be diagnosed in three basic ways:

1. Without using or disturbing the EBus
2. With loopback on the EBus but without the KL10 or without the KL10 running
3. With the KL10 running.

The interface is checked out primarily in a single-step manner. Full speed operation may only be checked with a running KL10. DIAG Word 1 contains the Diagnose 10/11 Interface bit. When this bit is set, the following occurs:

The interface clock is disabled and single step operation begins. Interrupts are inhibited from being sent to the KL10. The interface operates in the normal manner except that EBus operations never complete because no interrupts are issued to the KL10. Therefore, a bit has been provided that allows setting of the EBus complete, allowing the operation to continue.

The interface control is run by an up-counter and three decoders. The decoders are selected by the major state flip-flops. The up-counter is loadable by the rightmost 4 bits of DIAG Word 2. This enables any minor logic state to be executed. The major states are not loadable. However, they naturally cycle until a condition occurs that indicates that an operation is ready to take place. These major state bits are readable. The diagnostic can lock any major state on by bit 07 of DIAG Word 2.

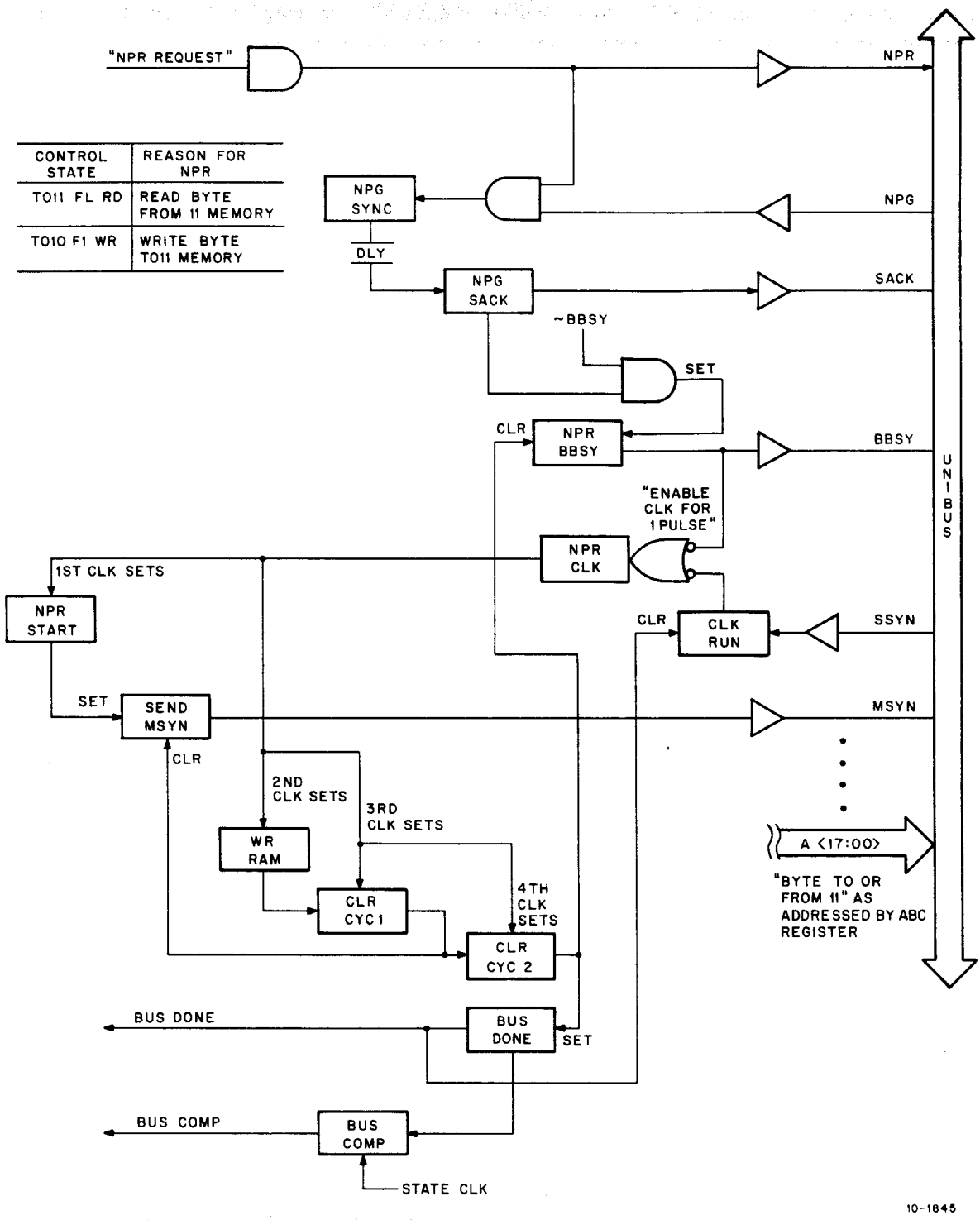


Figure 3-26 NPR Sequence

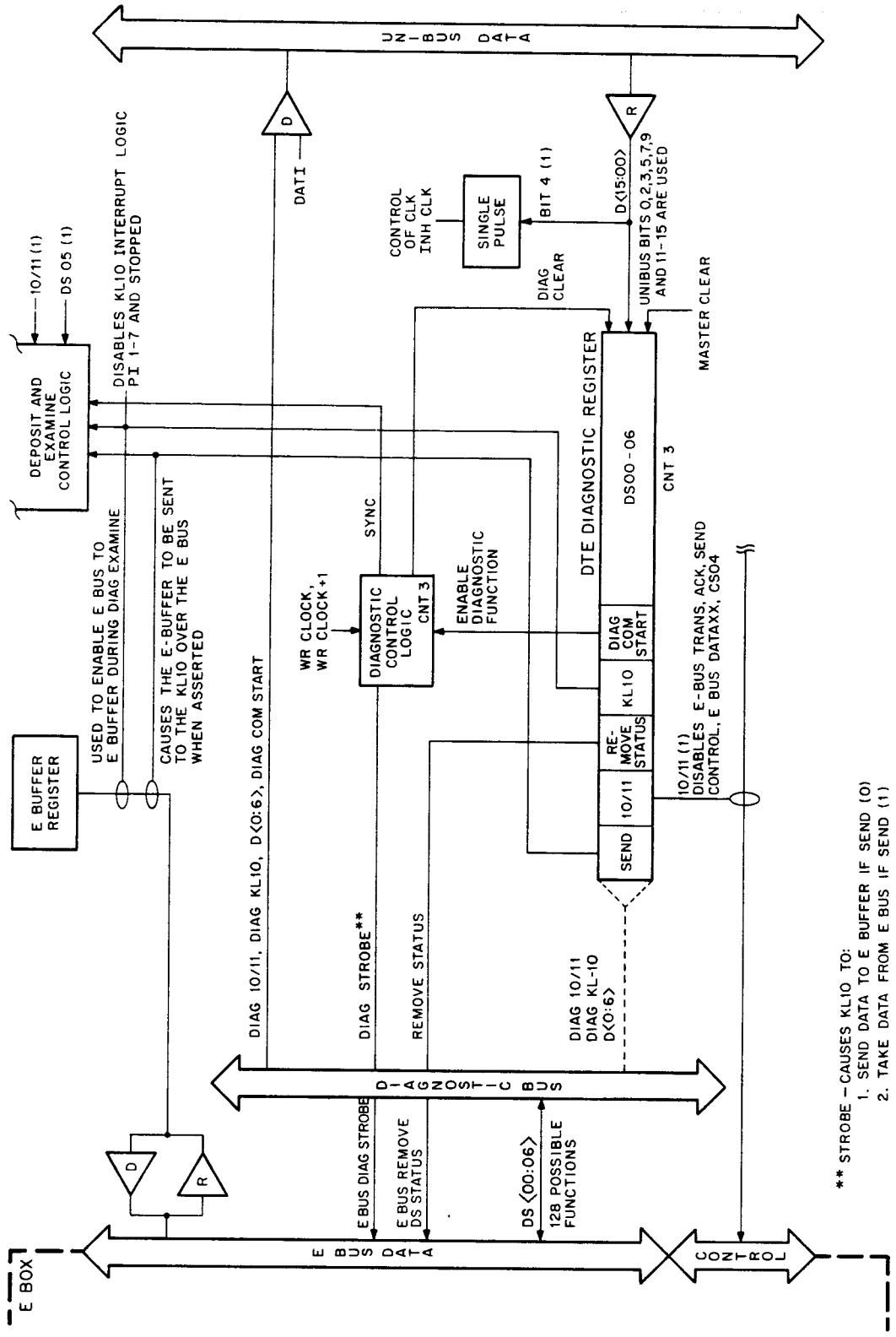


Figure 3-27 Simplified Diagnostic Functional Diagram

3.8.2 Diagnostic Control Logic

Most of the diagnostic control logic along with the diagnostic bus is contained on the bus control print. The diagnostic bus contains 10 lines.

DS 00-06	Diagnostic select lines. The PDP-11 sends encoded diagnostic functions to the KL10 on these lines. These lines can be read by the PDP-11 at any time even while the rest of the EBus is active for other devices.
STROBE	Diagnostic strobe. This line is asserted to indicate that the diagnostic select lines are stable, and that the indicated function should be performed.
REMOVE STATUS	When true, causes the KL10 to disable the basic CPU status from the DS lines, switch the translator (only for DS lines) to convert TTL to ECL, and put the EBus translator under control of DS bits 00 and 01.

Four control bits are associated with the diagnostic logic. These may be set via the Unibus by a DATO to the Diagnostic register.

1. Diagnostic 10-11 controls:

   a. State hold in conjunction with DS00

   b. Inhibit clock, in conjunction with single pulse

   c. EBus dialogue, i.e., ACKNOWLEDGE, TRANSFER, DATAXX (PHYS CONT NO.)

   d. E-Mixer Select in conjunction with both DS02 which controls MIX SEL 1 and DS01 which controls MIX SEL 2

   This bit (although not part of the diagnostic bus) is used to check out the 10-11 Interface without disturbing the KL10.

2. Diagnostic KL10 controls:

   a. E-Buffer Select, in conjunction with DEX Address 2

   b. Bus Selection (X, Y, Z, A), in conjunction with DIAG SEND PI01 -07 and KL10 stopped

   c. Can prevent CLK inhibit and/or CLKS TRANSFER REQ.

   This bit can be used with the 10/11 bit to check out the 10-11 interface and EBus logic.

3. Diagnostic Command Start controls:

   a. Enables strobe for one diagnostic cycle, then clears Diagnostic Command Start, as well as the Diagnostic register in the DTE upon transmission of the diagnostic function to the KL10.

4. DIAG Send controls:

    a. Enables diagnostic data transfers via the EBus to the KL10. When clear, diagnostic data transfers from the KL10 are possible.

3.8.3 Diagnostic Programming Synopsis

The following is a brief synopsis of how to proceed to program a 36-bit data transfer both to the KL10 and to the PDP-11. Also, Figure 3-28 presents an overview of the diagnostic bus operations.

1. Diagnostic Write - The DIAG KL10 bit should first be set. If it is not set, the interface will hang up waiting for interrupt service. A Deposit operation should be set up and performed. The address portion of the sequence may be loaded with any value as this information is not used. The interface will then load the E-Buffer with the data that was previously placed into the RAM Deposit data locations (DEX WD 1-3). Next, the appropriate diagnostic select bits, DIAG Send and DIAG Command Start, should be set. This will cause the data to be sent over the EBus to the KL10.

2. Diagnostic Read - A command with the appropriate diagnostic select bits should be sent. The DIAG KL10 bit and function code (DS00-06) to control the transfer. The diagnostic program must sample the DEX DONE bit before attempting to read or load information. When this bit is set, the data may be read from the interface using the same bit assignments as with an Examine or Deposit.

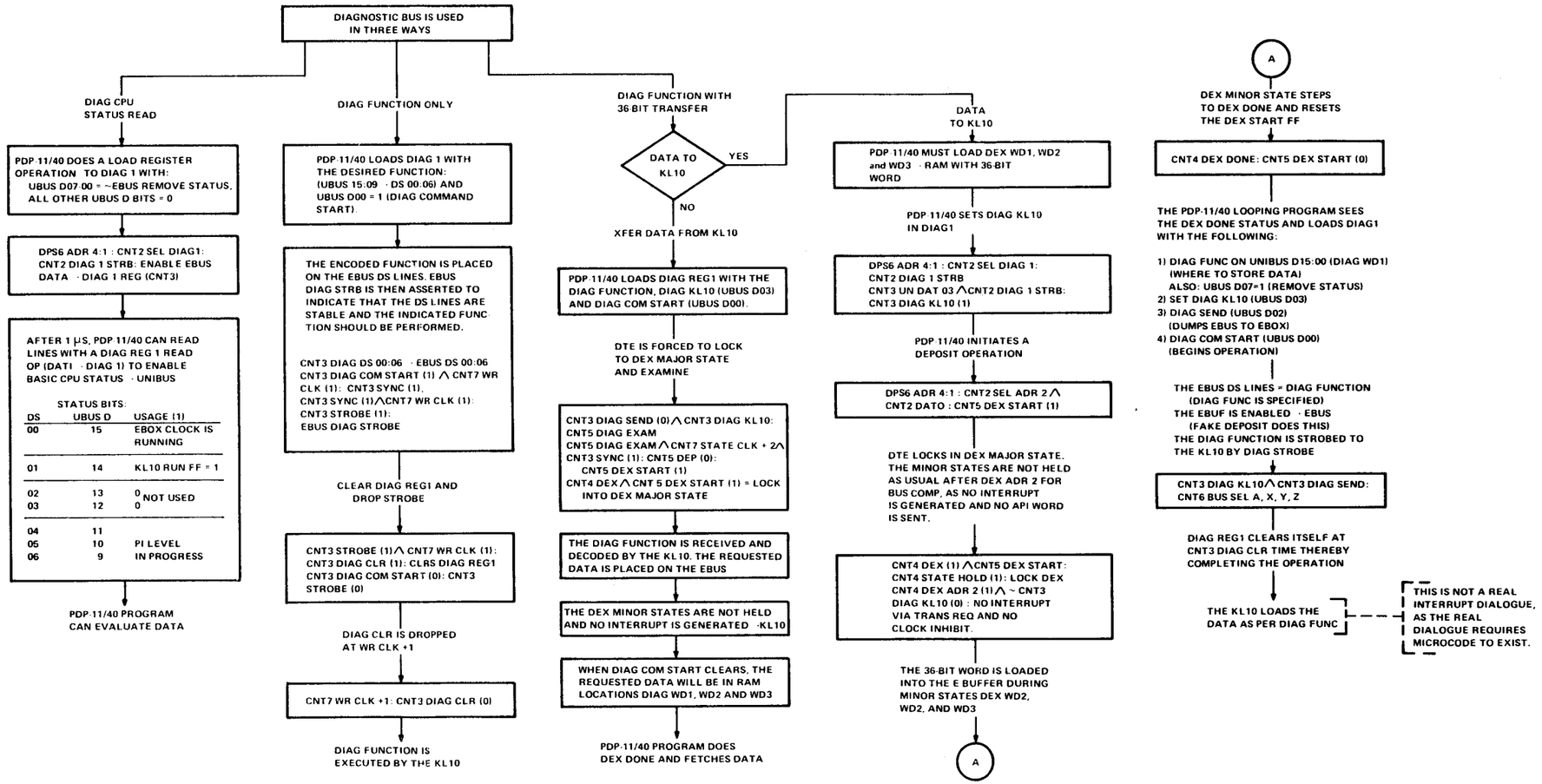


Figure 3-28 DTE/Diagnostic Bus Operation Flow

APPENDIX A
ABBREVIATIONS AND MNEMONICS

A	Address
ABC	Address and Byte Count
AC LO	AC Line Low
ACT	Active
ACK	Acknowledge
ACKN	Acknowledge
ADR	Address
API	Arithmetic Processor Interrupt
BBSY	Bus Busy
BC	Byte Count
BG	Bus Grant
BR	Bus Request
C	Control
CLK	Clock
CLR	Clear
CNT	Control
CON	Condition
CONI	Conditions In
CONO	Conditions Out
CS	Control Select
CYC	Cycle
D	Data
DAT	Data
DATAI	Data In
DATAO	Data Out
DATI	Data In
DATO	Data Out
DC LO	DC Line Low
DEP	Deposit
DEPST	Deposit
DEX	Deposit and Examine
DIAG	Diagnostic
DIS	Disable
DLY	Delay
DPS	Data Path and Status
DS	Device Select

EBH	Execution Buffer Hold
EBUF	Execution Buffer
EBUS	Execution Bus
EMIX	Execution Buffer Mixer
EN	Enable
ERR	Error
EXAM	Examine
F	Function
GD	Good
H	High
I BIT	Interrupt Bit
INC	Increment
INH	Inhibit
INIT	Initialize
INTER	Interrupt
INTR	Interrupt
IOP	Input/Output Pulse
L	Low
LD	Load
LT	Left
MIX	Mixer
MST	Master
MSYN	Master Sync
NPG	Non-Processor Grant
NPR	Non-Processor Request
NULL	ASCII Code of Zero
OP	Operation
PA	Parity
PAR	Parity
PB	Parity
PHY	Physical
PI	Priority Interrupt
PIA	Priority Interrupt Assignment
PLS	Pulse
PWR	Power
RAM	Random Access Memory
RD	Read
RDY	Ready
REC	Receive
REG	Register
REQ	Request
RFM	Ram File Mixers
RG	Register
ROM	Read Only Memory

SACK	Slave Acknowledge
SEL	Select
SHF	Shift
SP	Special
SSYN	Slave Sync
ST	State
STRB	Strobe
SW	Switch
TO10	To the KL10 (direction)
TO11	To the PDP-11 (direction)
TRAN	Transfer
TRANS	Transfer
UB	Unibus
UN	Unibus
WD	Word
WR	Write
XFER	Transfer

DTE MANUAL INDEX

Reader's Comments

DTE20 Ten-Eleven Interface
Unit Description
EK-DTE20-UD-003

Your comments and suggestions will help us in our continuous effort to improve the quality and usefulness of our publications.

What is your general reaction to this manual? In your judgment is it complete, accurate, well organized, well written, etc.? Is it easy to use? _____

_____

_____

_____

What features are most useful? _____

_____

_____

_____

What faults do you find with the manual? _____

_____

_____

_____

Does this manual satisfy the need you think it was intended to satisfy? _____

Does it satisfy *your* needs? _____ Why? _____

_____

_____

_____

Would you please indicate any factual errors you have found. _____

_____

_____

_____

Please describe your position. _____

Name _____ Organization _____

Street _____ Department _____

City _____ State _____ Zip or Country _____

CUT OUT ON DOTTED LINE