EDUCATIONAL SERVICES

VAX/VMS
Training

VAX/VMS
Device Driver
I/O Architecture

digital

EY-2278E-MK-0001

I/O ARCHITECTURE

Prepared by Educational Services
of
Digital Equipment Corporation

First Edition, October 1984

Printed in U.S.A.

The manuscript for this book was created using DIGITAL Standard Runoff. Book production was done by Educational Services Development and Publishing in Nashua, NH.

CONTENTS

INTRODUCTION 11-3
OBJECTIVES 11-3
RESOURCES 11-4
ADDITIONAL READING 11-4
TOPICS 11-5
I/O ARCHITECTURE 11-10
    750 11-10
    780 11-10
OTHER CONSIDERATIONS 11-11
UNIBUS ADAPTER (UBA) 11-12
    UNIBUS Subsystem 11-14
    Physical Address Space 11-15
    Address Translation (VAX-11/780 Specific) 11-18
    SBI to UNIBUS Address Translation 11-19
    UNIBUS to SBI Address Translation (11/780) 11-20
    Direct Data Path (11/780) 11-21
    Buffered Data Paths (11/780) 11-22
    Line Division by Function 11-23
    Signal Description 11-24
    Priority Access 11-25
    SBI Bus Arbitration 11-25
    SBI to UNIBUS Device Write 11-27
    SBI to UNIBUS Device Read 11-27
    UNIBUS to SBI DMA Write Through Direct Data Path 11-28
    UNIBUS to SBI Read Through Direct Data Path 11-28
    UNIBUS DMA Write Through Buffered Data Paths 11-29
    UNIBUS to SBI Read Through Buffered Data Paths 11-30
    Interrupt Dispatching 11-31
    UNIBUS Interrupt Processing 11-32
    BRRVR Bit Settings 11-33
MASSBUS ADAPTER (MBA) 11-35
    MBA Operation 11-36
    MBA Structure 11-37
    MBA Address Translation 11-38
    MBA Address Space 11-39
    Location of MASSBUS Registers 11-40

FIGURES

11-1 VAX-11/780 Processor 11-7
11-2 VAX-11/750 Processor 11-7
11-3 VAX-11/730 Processor 11-8
11-4 Basic Bus Configuration for the VAX-11/780 and 11/750 11-9
11-5 UNIBUS Subsystem 11-14
11-6 VAX-11/780 Physical Address Space 11-15
11-7 VAX-11/750 Physical Address Space 11-16
11-8 VAX-11/730 Physical Address Space 11-17
11-9 SBI Address Space 11-18
11-10 SBI to UNIBUS Address Translation 11-19
11-11 UNIBUS to SBI Address Translation 11-20
11-12 Signal Lines on the SBI 11-24
11-13 SBI Bus Arbitration 11-26
11-14 CPU Writing to a Device Register 11-27
11-15 CPU Reading a Device Register 11-27
11-16 Data Sent from a Device to Memory in a DMA Operation Using the Direct Data Path 11-28
11-17 Data Read from Memory by a DMA Device Using the Direct Data Path 11-28
11-18 Data Written to Memory by a DMA Device Using a Buffered Data Path 11-29
11-19 Data Read from Memory by a DMA Device Using a Buffered Data Path 11-30
11-20 Format of SCB 11-31
11-21 Processing a Device Interrupt 11-32
11-22 Format of Several Internal UBA Registers 11-34
11-23 Simplified Block Diagram of the MBA 11-37
11-24 MBA Mapping a Virtual Address to a Page Frame Number 11-38
11-25 Structure of MBA Address Space 11-40

TABLES

11-1 Summary of NEXUS Values 11-11

INTRODUCTION

This module serves as an introduction to the primary hardware components that comprise the I/O architecture. Devices are connected to an adapter (on the UNIBUS or MASSBUS). The various adapters, CPU, and memory controllers are connected through a bus called the SBI. Adapters serve to interface between the SBI and various devices, and to translate addresses the devices understand (UNIBUS or MASSBUS) to addresses understood by the CPU and memory controllers (SBI addresses). This module describes the UNIBUS and MASSBUS adapters, as well as the SBI. An Appendix is provided which discusses the operaton of the UNIBUS itself. A second Appendix is provided which describes the DR32 adapter.

OBJECTIVES

Upon completion of this module, you will be able to:

1. Diagram the VAX-11/780 hardware architecture, basic bus configuration, and UNIBUS subsystem.
2. Diagram UNIBUS to SBI address translation (and vice versa).
3. Determine when it is appropriate to use the direct/buffered data paths.
4. Diagram MASSBUS to SBI address translation.

RESOURCES

1. VAX-11/780 Architecture Handbook
2. VAX-11/780 Hardware Handbook
3. Guide to Writing a Device Driver for VAX/VMS

ADDITIONAL READING

1. UNIBUS Adapter Technical Description, Chapters 1 and 2
2. PDP-11 Peripherals Handbook
3. VAX/VMS I/O User's Reference Manual

TOPICS

- VAX Processors (780/750/730)
  - Bus Configuration
  - I/O Architecture Differences
  - UBA Data
- Physical Address Space
- Address Translation

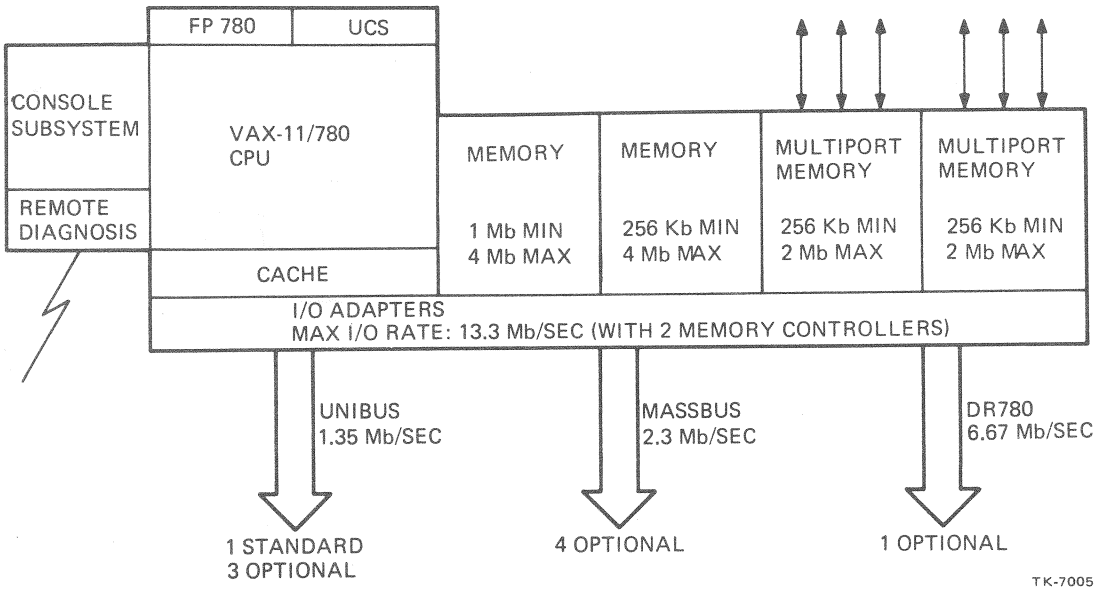


Figure 11-1 VAX-11/780 Processor

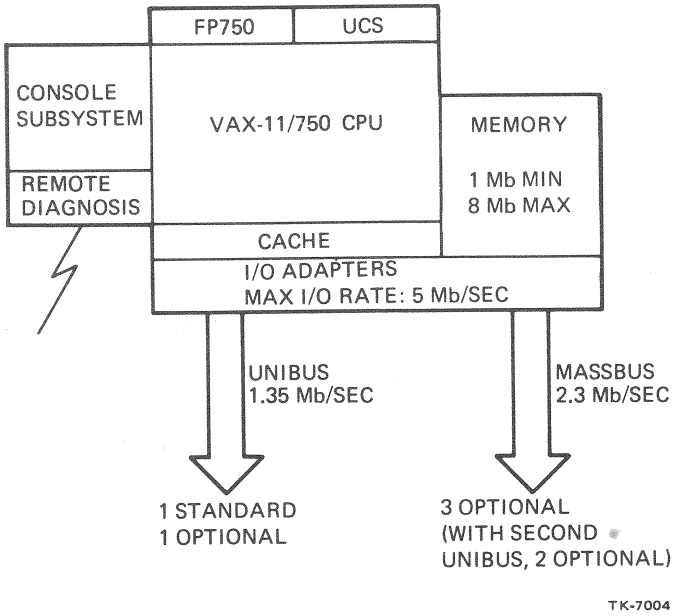


Figure 11-2 VAX-11/750 Processor

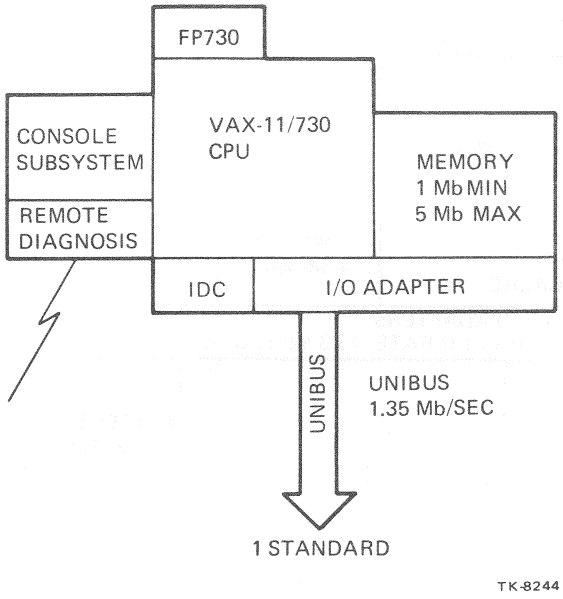


Figure 11-3 VAX-11/730 Processor

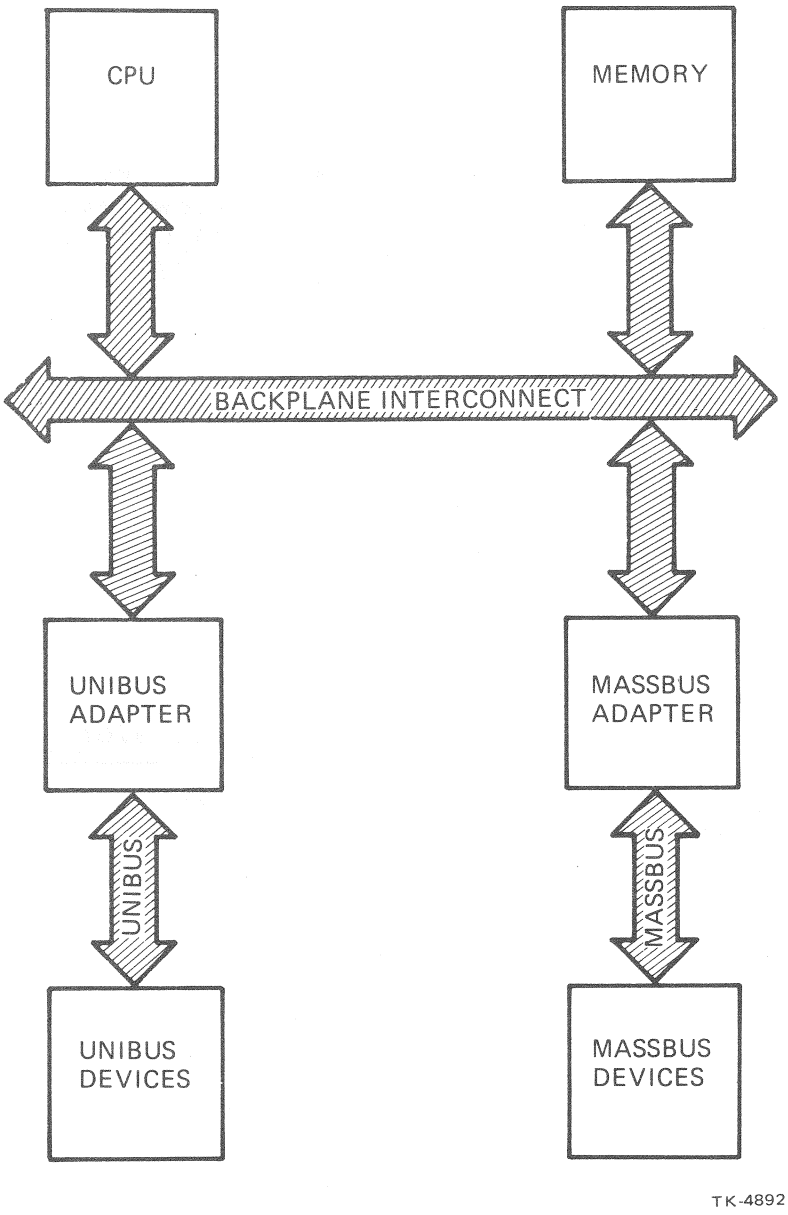


Figure 11-4 Basic Bus Configuration for the VAX-11/780 and 11/750

The Backplane Interconnect is the data path that links the central processor, the memory subsystem, and the hardware adapters provided for the UNIBUS and MASSBUS.

I/O ARCHITECTURE

750

CMI -

- uses no protocol
- CPU arbitrates
- no extended transfers
- 160 nanosecond clock
- no history silo
- 5 MB/sec bandwidth

780

SBI -

- has a protocol
- each nexus arbitrates
- allows extended transfers
- 200 nanosecond clock
- 16 cycle history silo
- 13.3 MB/sec bandwidth

OTHER CONSIDERATIONS

1. When writing a connect to interrupt program that needs to map device registers, the macro $IO750DEF should be used rather than $IO780DEF to define symbol IO750$AL_UBOSP (to find the start of UNIBUS address space).

   I/O address space (physically) starts at 20000000 (hex) for the 780, and F20000 (hex) for the 750. The starting physical address of UNIBUS address space is given by one of the following formulas:

   For 780:

   UB I/O space PFN = UB#*512!<<^X20100000+^0760000>/^X200>

   For 750:

   UBO I/O space PFN = <^XFC0000+^0760000>/^X200

2. Also, TR numbers differ for UBAs and MBAs on the 750 (important for SYSGEN CONNECT commands). See Table 11-1.

Table 11-1 Summary of NEXUS Values

	780 TR#	750 TR#	730 TR#
UBA 0	3	8	3
1	4	9	-
2	5	-	-
3	6	-	-
MBA 0	8	4	-
1	9	5	-
2	10	6	-
3	11	-	-

UNIBUS ADAPTER (UBA)

	78Ø	75Ø	73Ø
Number of UBAs	1 std, 3 optional	1 (integral)	Part of Memory Controller
Direct data path	1	1	1
Buffered data paths (buffer-size)	15 (8 bytes)	3 (4 bytes)	Ø
Mapping Registers	496	512 (496 used by VMS)	512 (496 used by VMS)
Physical Addr. Space	3Ø-bit	24-bit	24-bit
Interrupt Vectoring	Nondirect	Direct	Direct

NOTE

For more details on UNIBUS adapters, see the chapter on "The UNIBUS Adapter" in the Guide to writing a Device Driver for VAX/VMS, plus the UNIBUS Adapter Technical Description for the appropriate VAX (780/750/730).

UNIBUS ADAPTER (UBA) (Cont)

- Entire UNIBUS address space is directly mapped (physically); only upper 8K bytes (I/O page) are mapped virtually.
- Map registers allow UNIBUS to map into portions of SBI space.
- UNIBUS address selects a map register and specifies a byte offset.
- UBA map register selects a data path.
- Data path Ø is nonbuffered. Data paths 1-15 are quadword buffered.
- Map registers allow word-aligned UNIBUS data to be transferred to byte aligned SBI addresses.

UNIBUS Subsystem

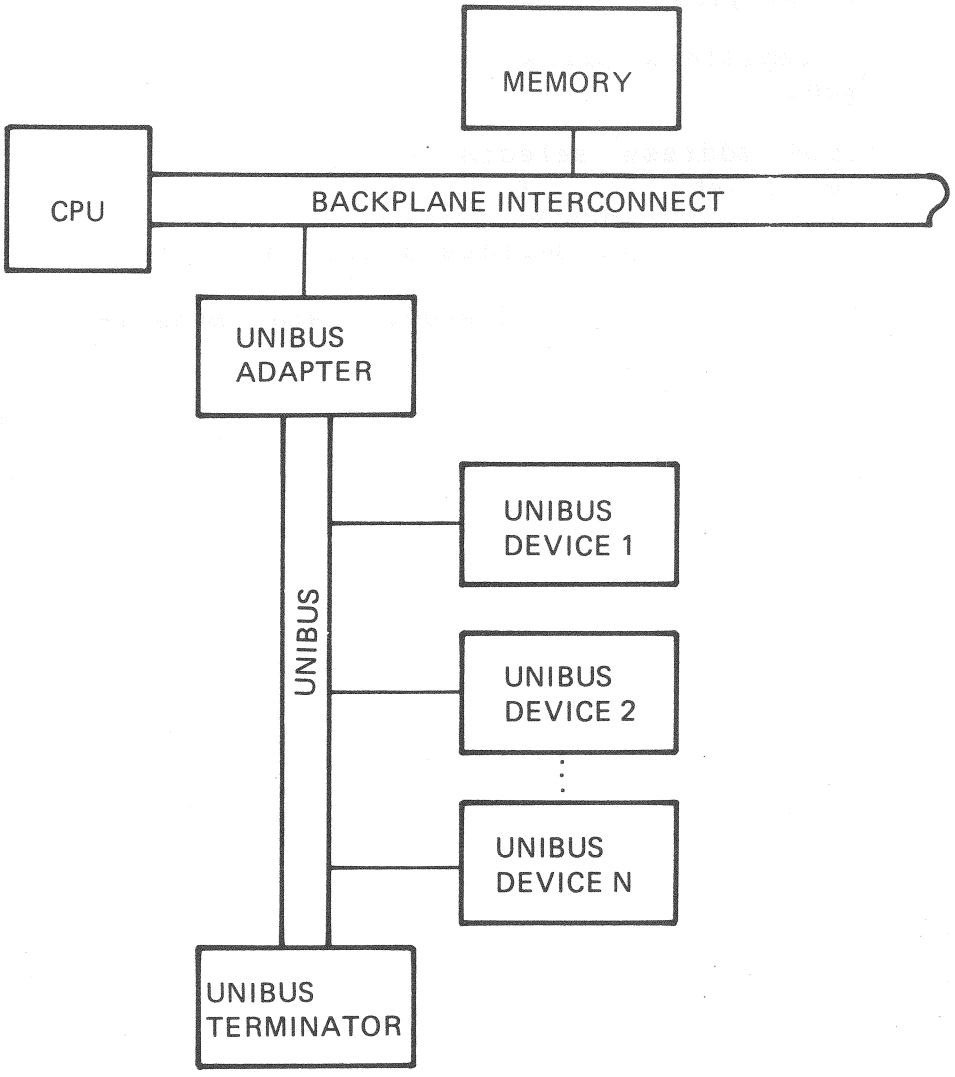


Figure 11-5 UNIBUS Subsystem

Note that devices performing buffered I/O operations can disregard the presence of the UBA.

Physical Address Space

Physical address space consists of two parts; one devoted to physical memory, and the other to device registers, as illustrated in Figure 11-6. Note that both physical and SBI addresses are provided.

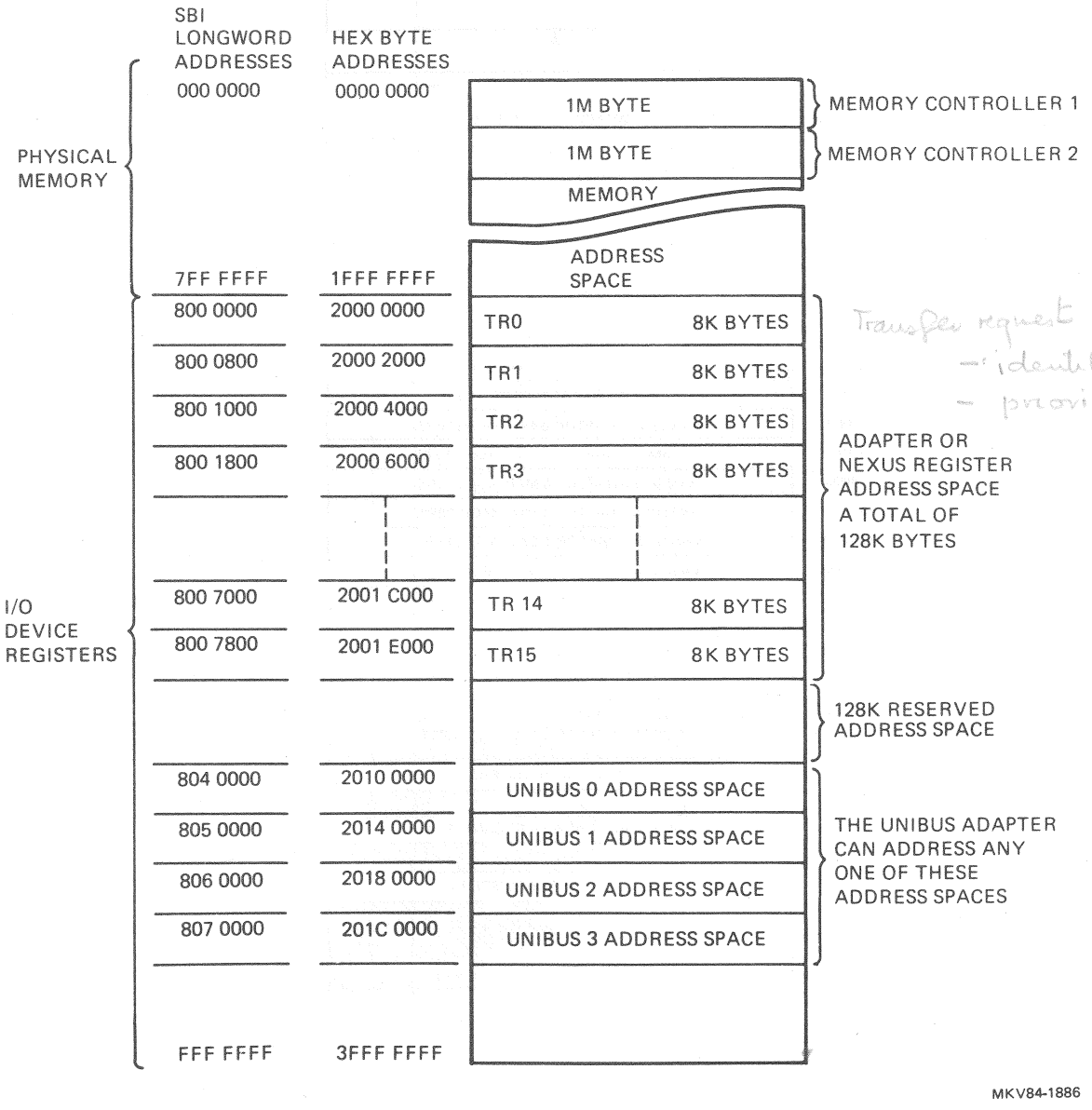


Figure 11-6 VAX-11/780 Physical Address Space

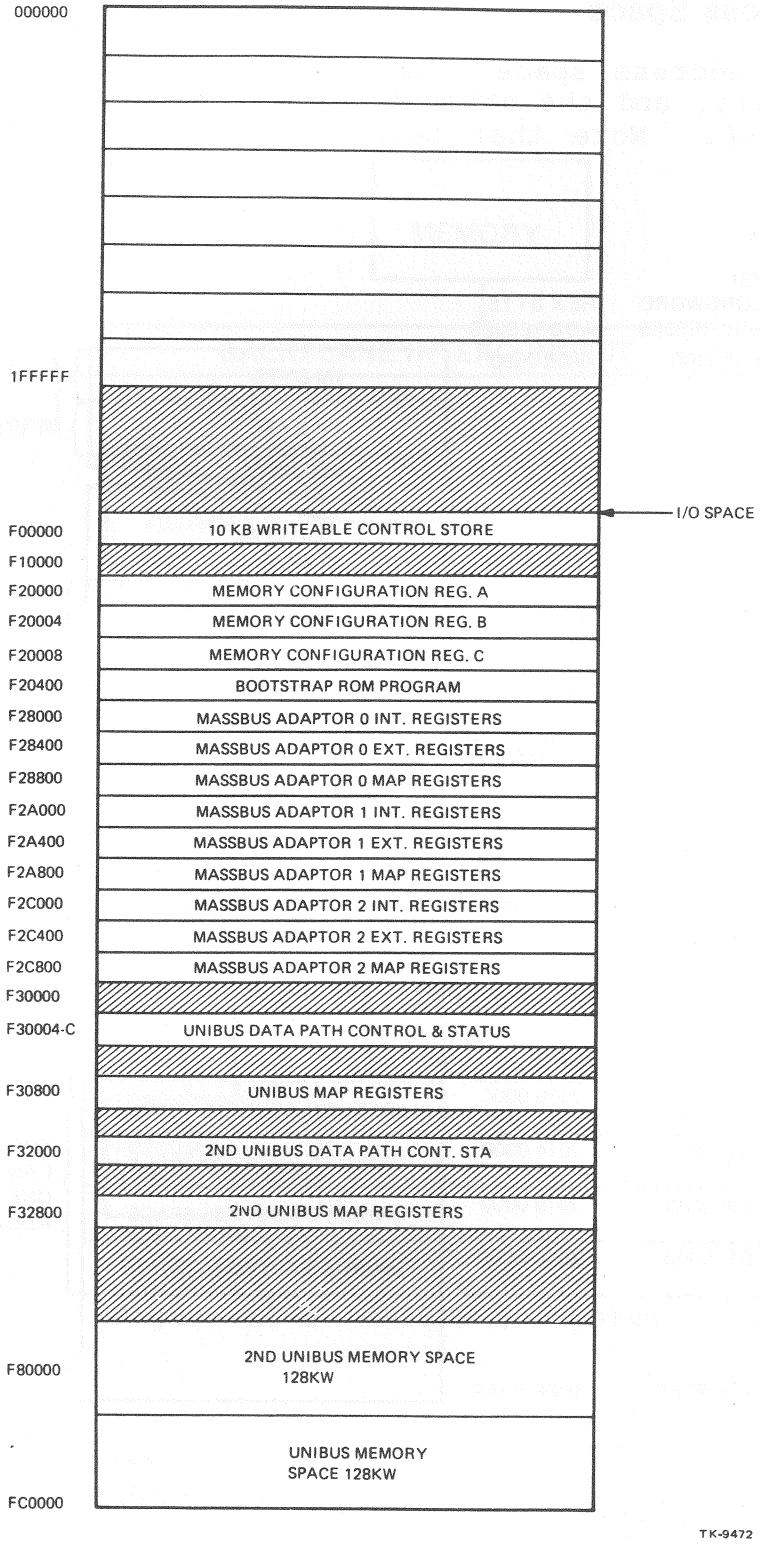


Figure 11-7 VAX-11/750 Physical Address Space

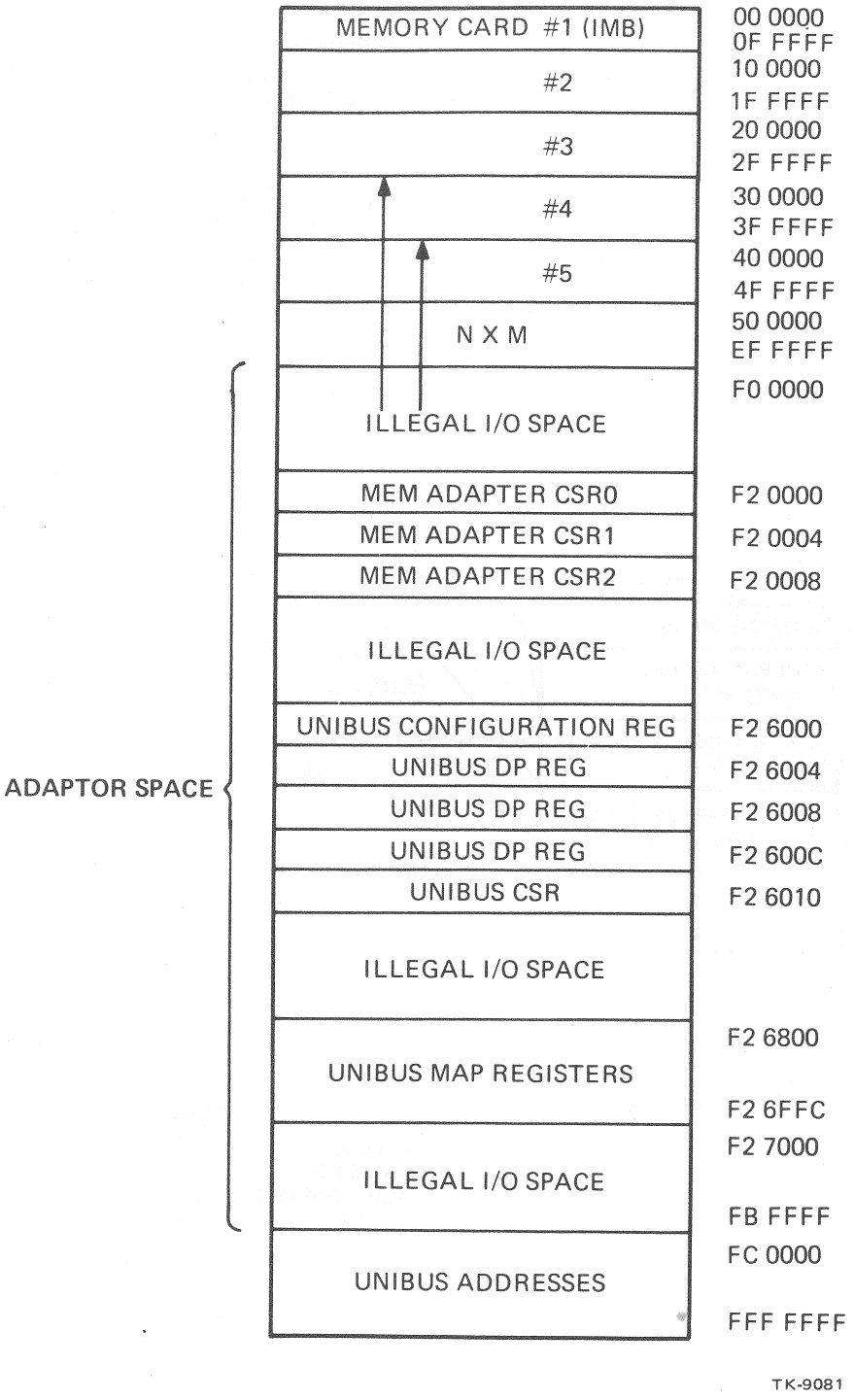


Figure 11-8 VAX-11/730 Physical Address Space

Address Translation (VAX-11/780 Specific)

Figure 11-9 shows the SBI address space controlled by the UBA. Note that there are two sections of memory that reference the UBA; one for internal UBA registers, the other for UNIBUS addresses.

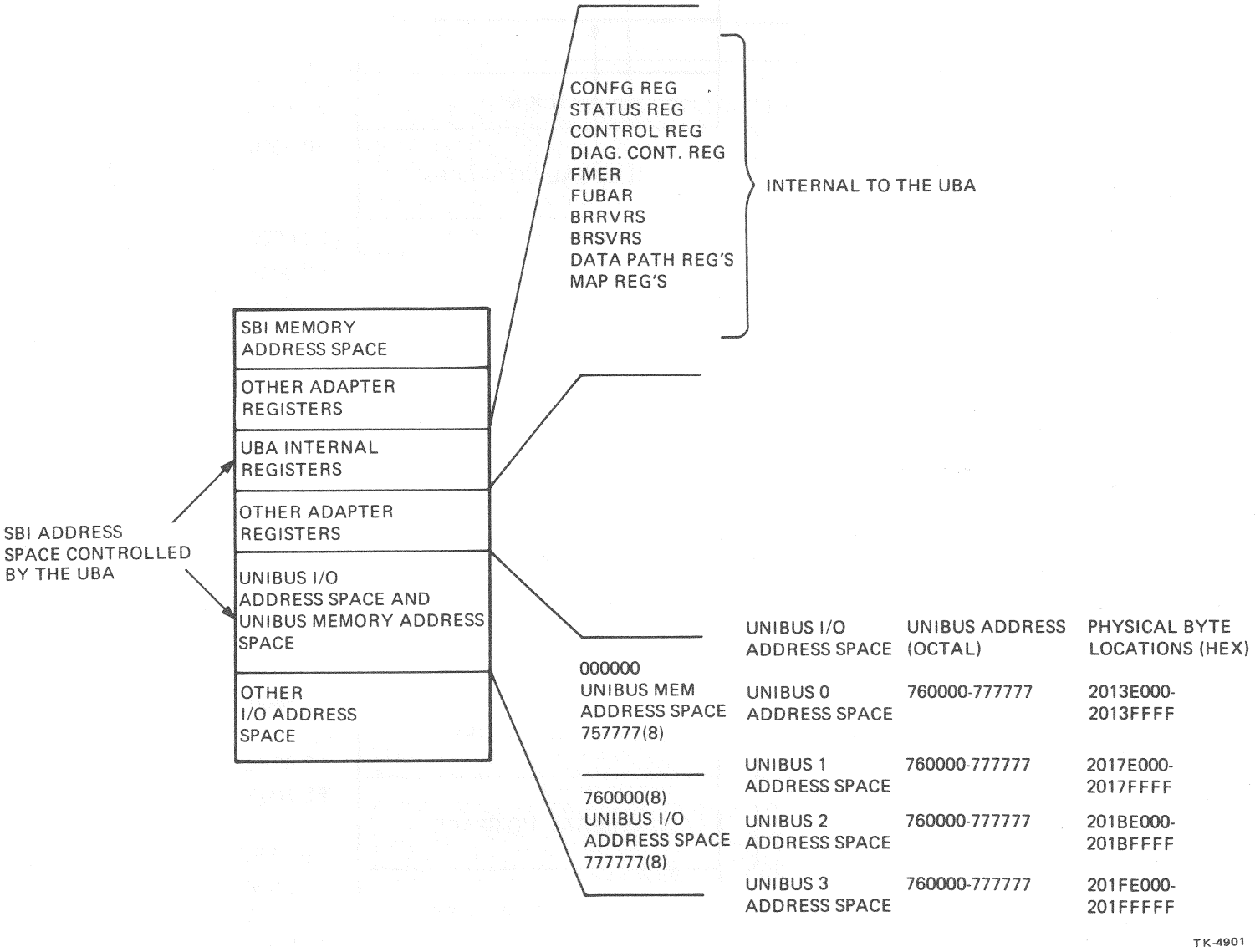


Figure 11-9 SBI Address Space

SBI to UNIBUS Address Translation

Figure 11-10 illustrates how 28-bit SBI addresses are converted into 18-bit UNIBUS addresses.

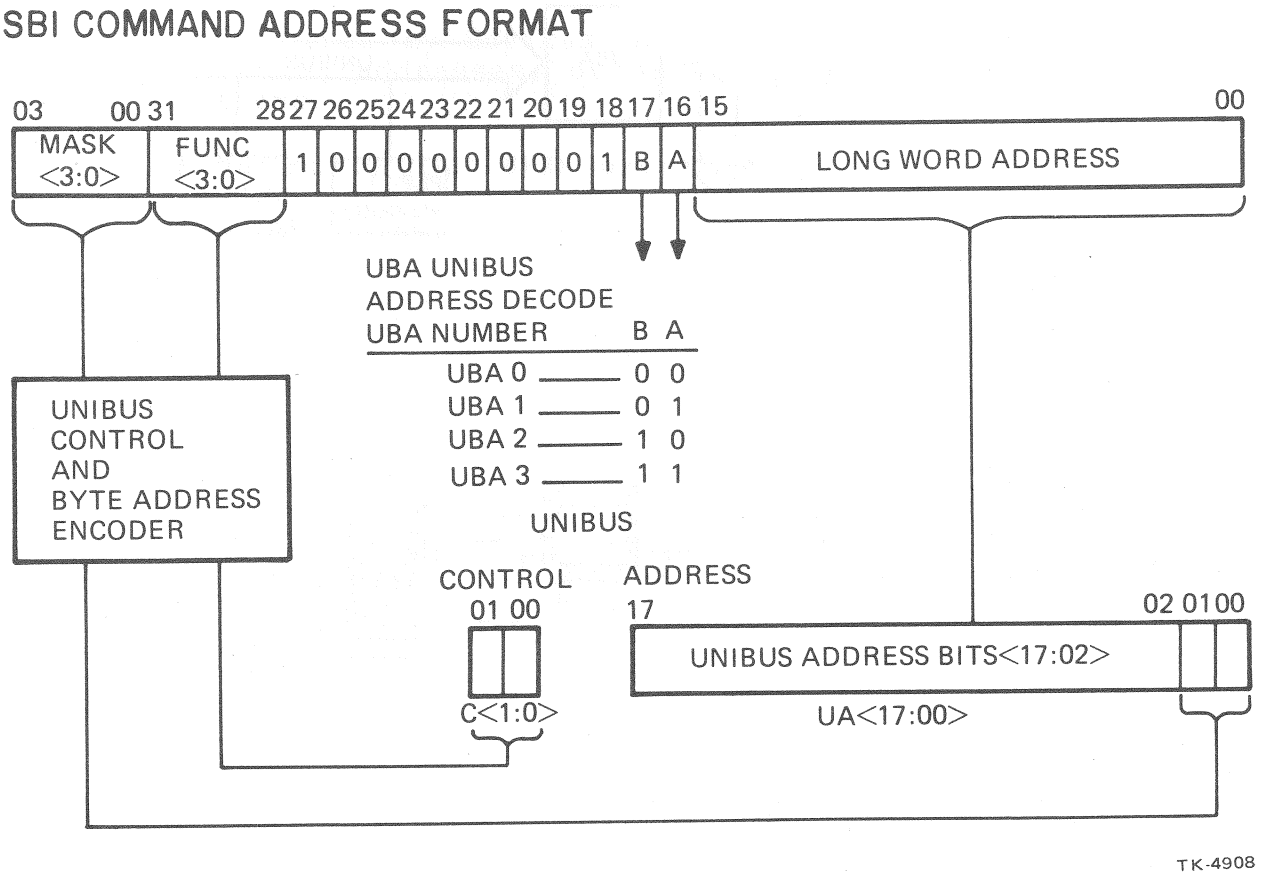
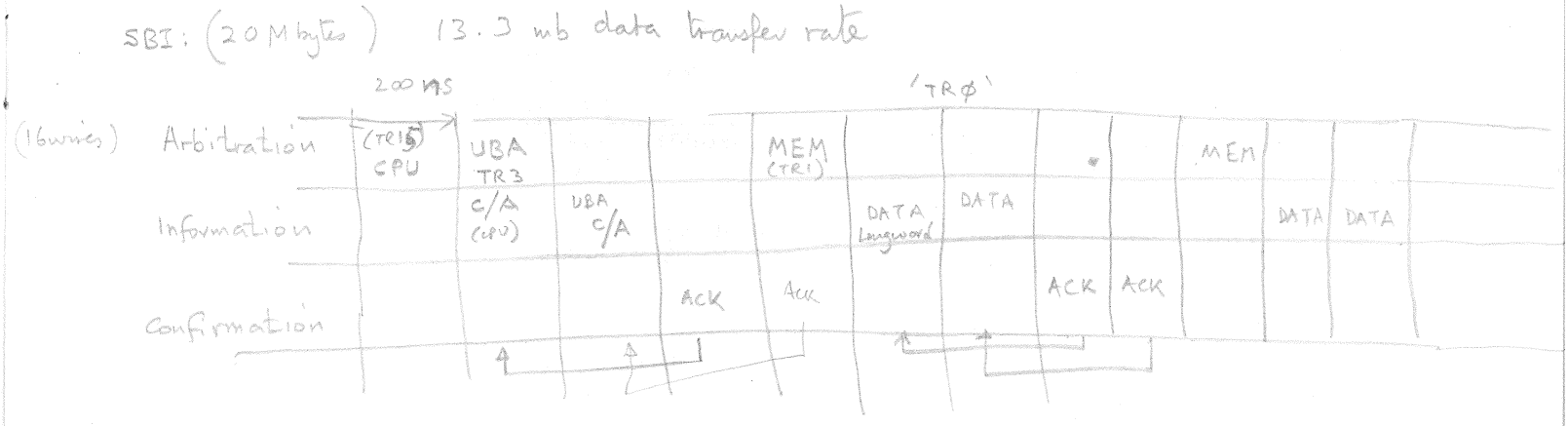


Figure 11-10 SBI to UNIBUS Address Translation



E.G. CPU Read to memory (CPU always reads quadword from memory)

UNIBUS to SBI Address Translation (11/780)

Figure 11-11 illustrates how 18-bit UNIBUS addresses are translated to SBI addresses using mapping registers.

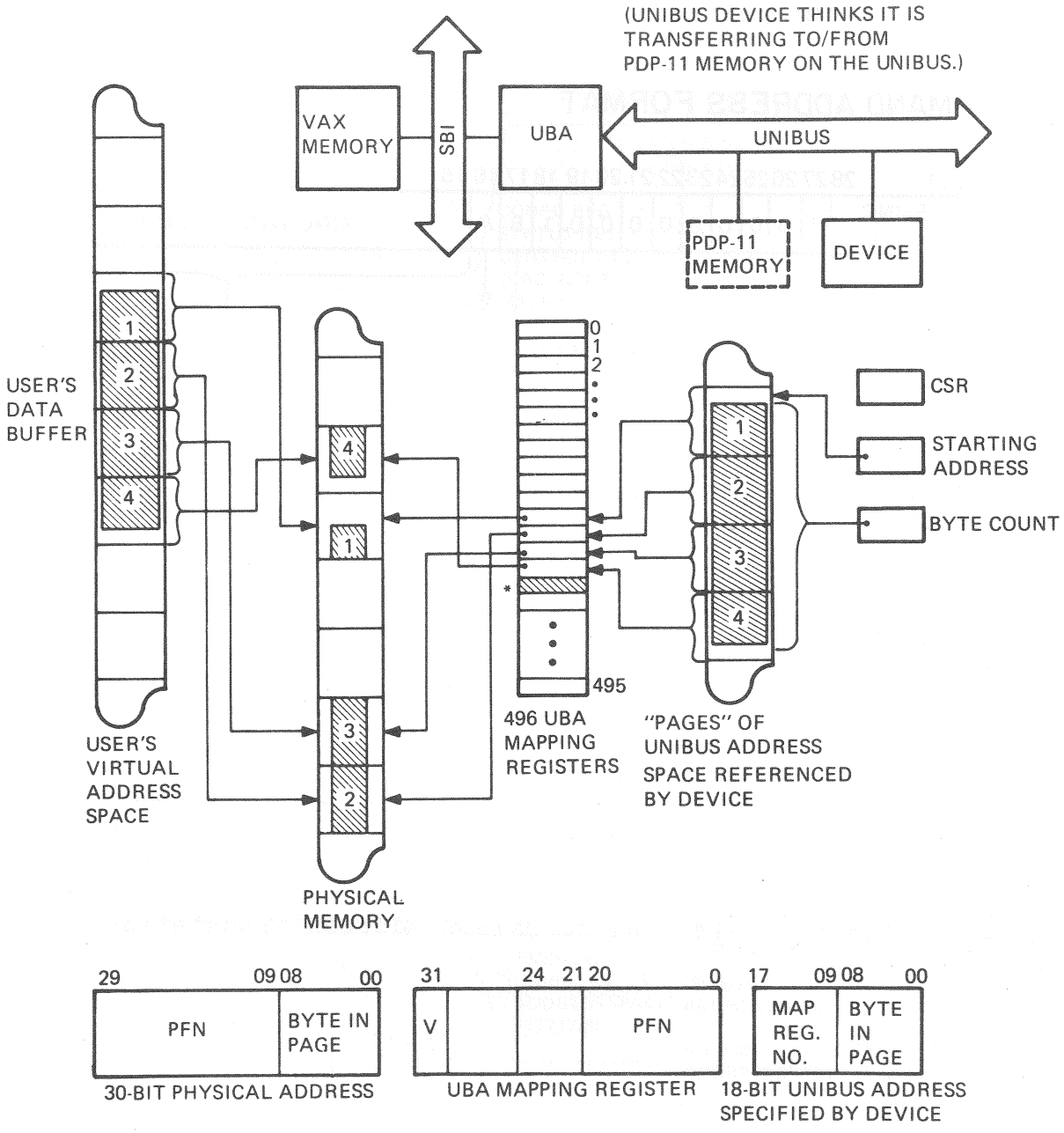


Figure 11-11 UNIBUS to SBI Address Translation

Direct Data Path (11/780)

- Can be assigned to more than one transferring UNIBUS device. UBA arbitrates between devices.
- Must be used by any device wanting to use interlock sequence (DATIP-DATOB) to the SBI.
- Must be used by devices not transferring to consecutive increasing addresses, or devices that mix read and write functions.
- Maximum throughput 800K bytes/sec.
- One UNIBUS transfer => one SBI transfer.
- Cannot transfer a word/byte to an odd SBI address. Therefore, FDT routines should check to make sure buffer is on a word boundary.
- Slower than buffered data path.

Buffered Data Paths (11/780)

- For fast DMA block transfer devices (maximum 1.5 megabytes/sec).
- Improves the effective UNIBUS bandwidth.
- Makes more efficient use of the SBI.
- Enables word aligned block transfer devices to begin and end on odd bytes of SBI memory.
- Enables 32-bit data transfers from random longword-aligned SBI addresses.
- Allows access to SBI I/O registers from UNIBUS devices.
- Constraints:
  - Assigned to one device at a time.
  - All transfers within a block must be to consecutive increasing addresses.
  - All transfers within a block must be the same function type (read or write).
  - DATIP not recognized.
  - The buffered data path must be purged after each transfer to:
    - flush prefetches on read
    - empty buffer on write
- Four UNIBUS transfers => one SBI transfer.

APPENDIX

Line Division by Function

Arbitration	transfer requests Ø-15 (Ø=hold)
Information	data, interrupt summary response, command/ address
Confirmation	busy, ack*, error *Acknowledgments are always sent two cycles after a command is received by the responder nexus.
Interrupt	interrupt request group, alert
Control	fail, dead, unjam, intlk

NOTE

TR = Transfer Request. SBI access priority increases from TR15 to TRØØ.

Signal Description

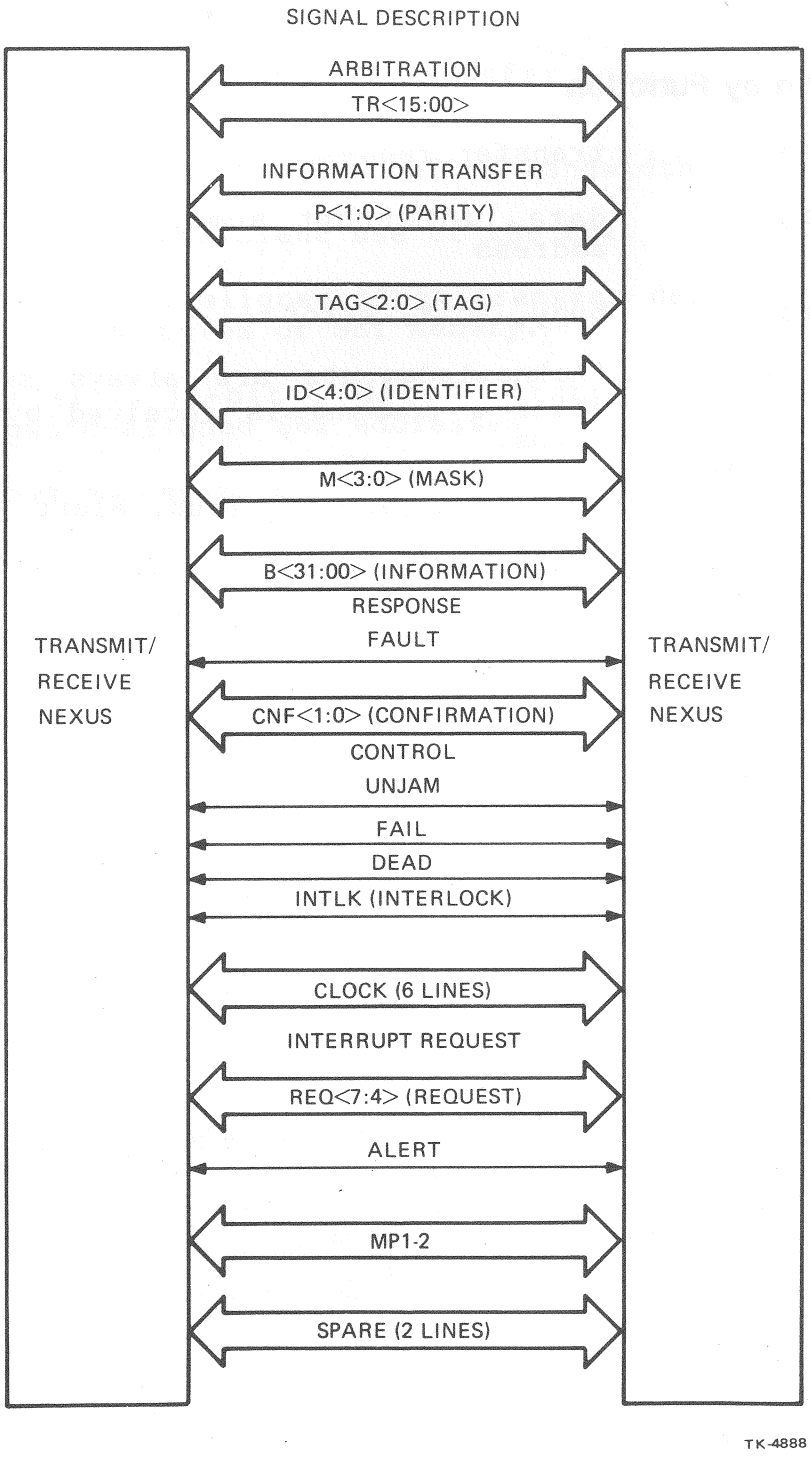
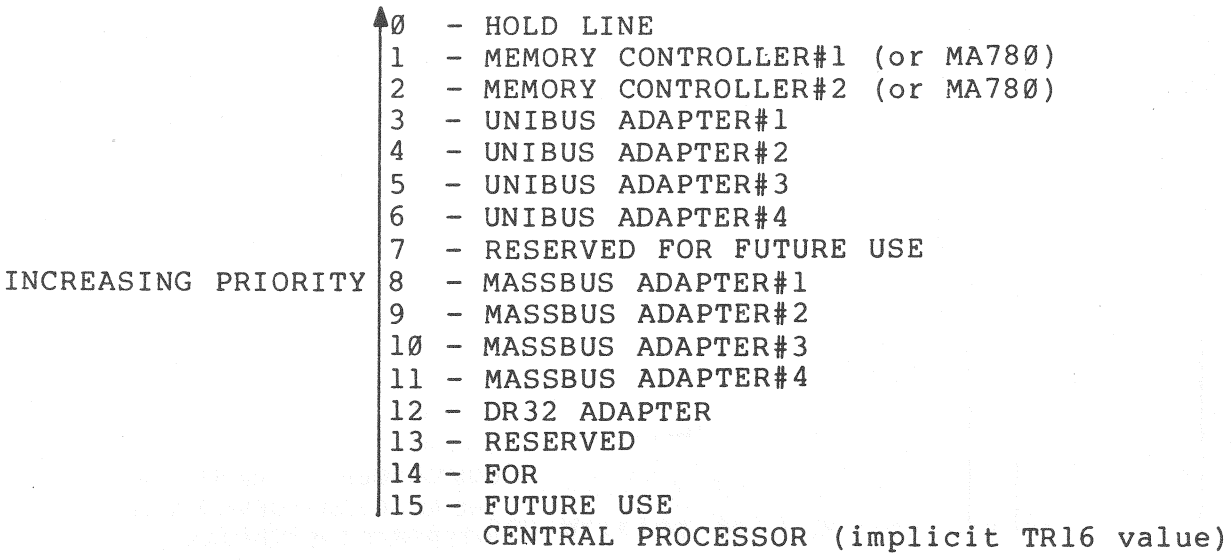


Figure 11-12 Signal Lines on the SBI

Priority Access

Ownership of the SBI is granted to the nexus with the highest priority (lowest TR number) that requests it. Standard TR number assignments for the SBI are shown below.



SBI Bus Arbitration

- The CPU gets the SBI only if no one else wants it.
- UBA priority is greater than MBA priority, since the UBA would otherwise never gain access to the SBI on a system with multiple MBAs (that could keep the SBI always busy).
- A nexus owning the SBI may assert the hold line (TRØ) to continue ownership of the SBI to perform extended reads or writes (see Figure 11-13).
- A nexus TR number is not fixed. A configuration with two memory controllers, and an MA78Ø, would have its first UBA at TR#4. The MA78Ø would be at TR#3.
- Relative positioning of the nexus never changes.

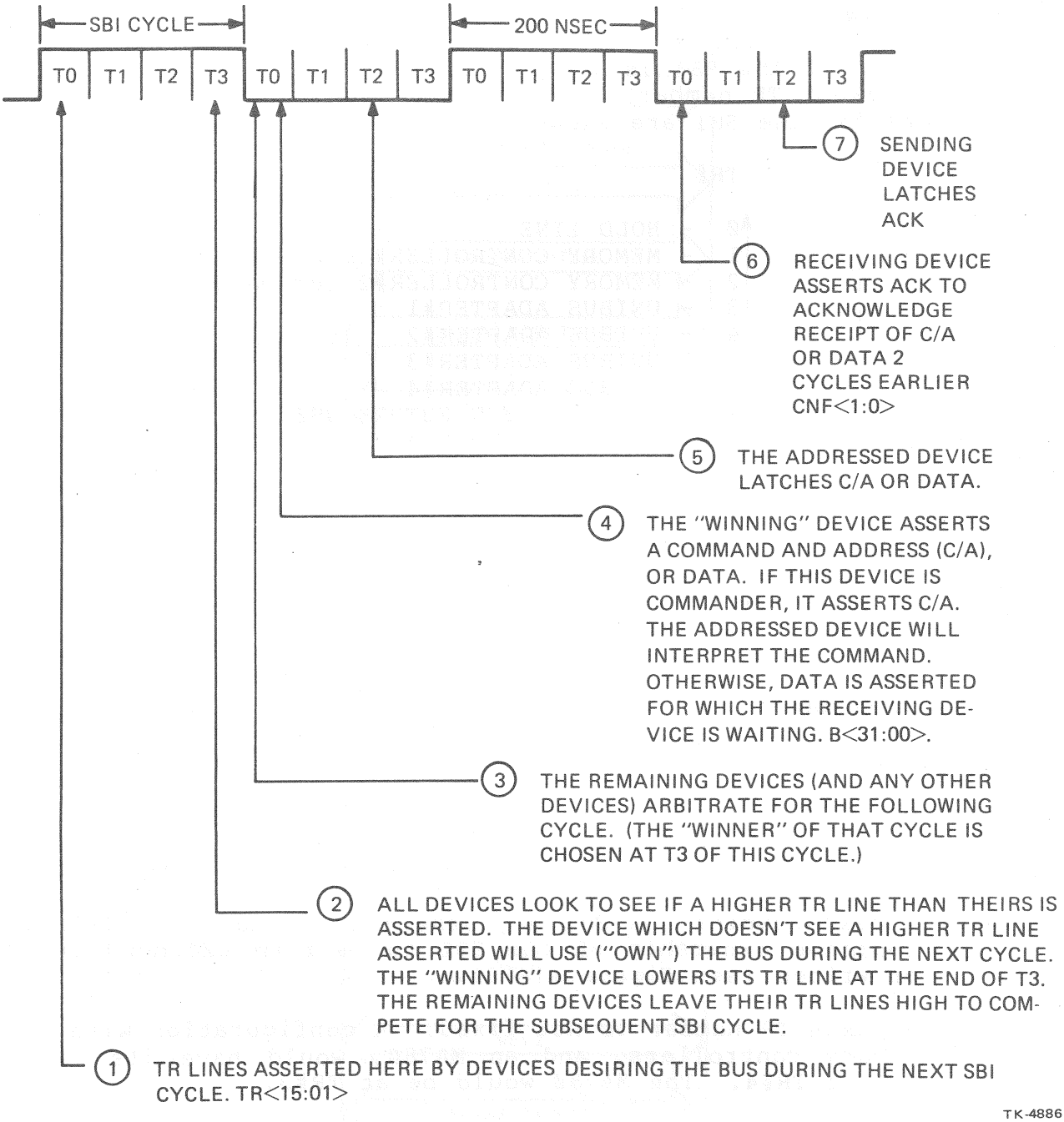


Figure 11-13 SBI Bus Arbitration

SBI to UNIBUS Device Write

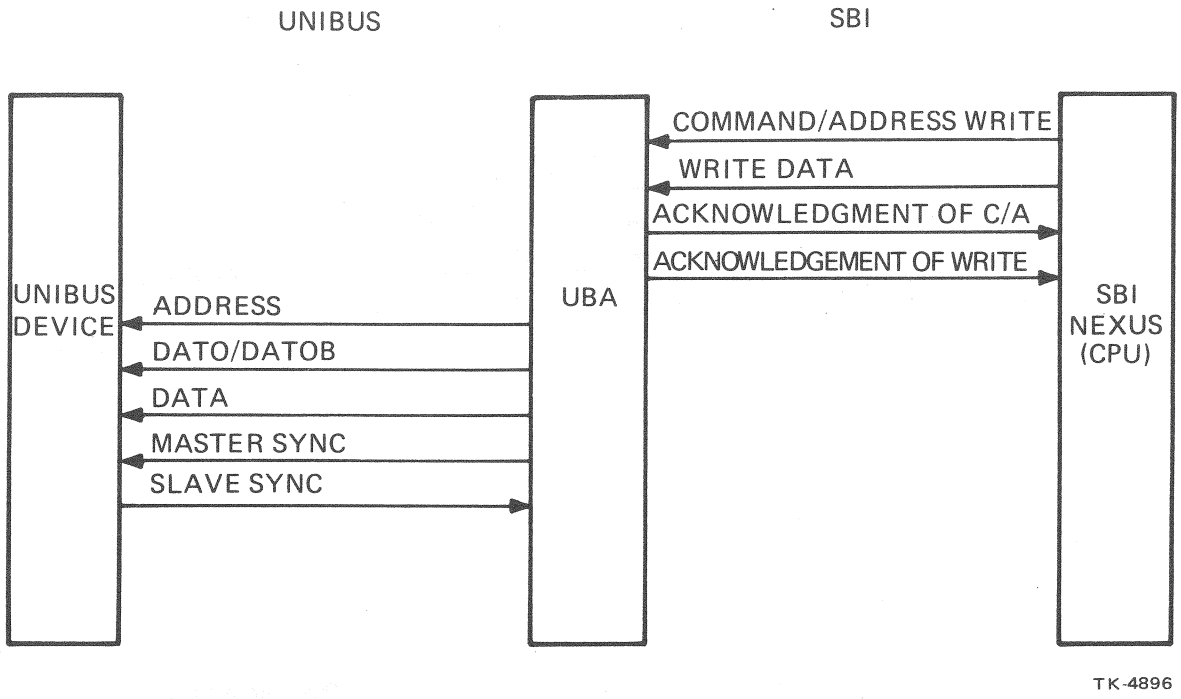
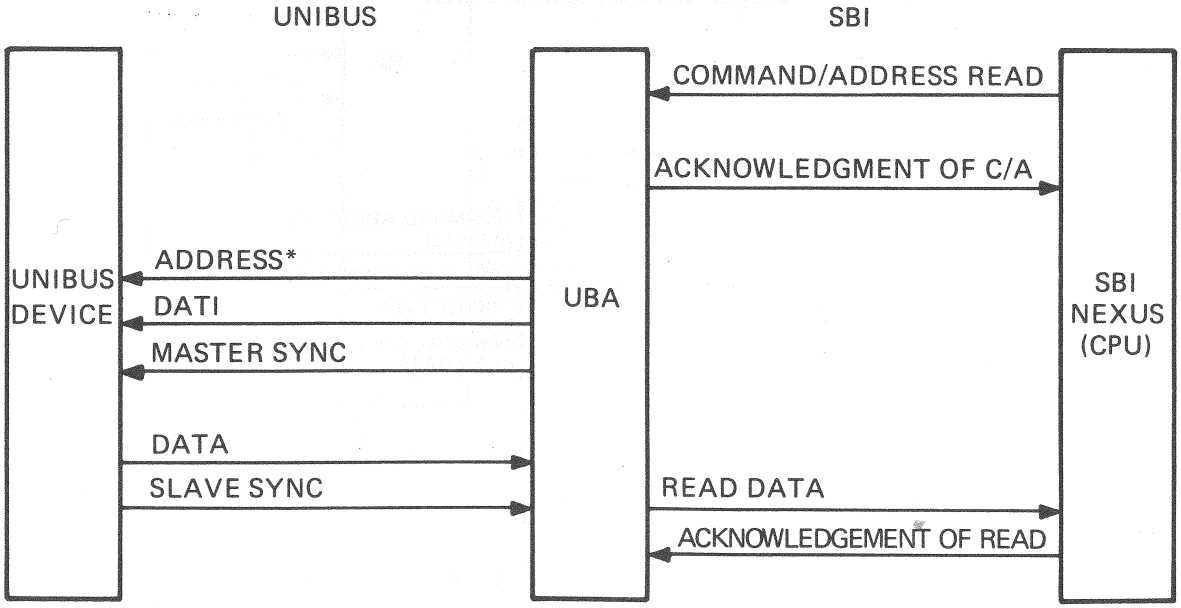


Figure 11-14 CPU Writing to a Device Register

SBI to UNIBUS Device Read



*SEE SBI TO UNIBUS ADDRESS TRANSLATION DIAGRAM

Figure 11-15 CPU Reading a Device Register TK-4893

UNIBUS to SBI DMA Write Through Direct Data Path

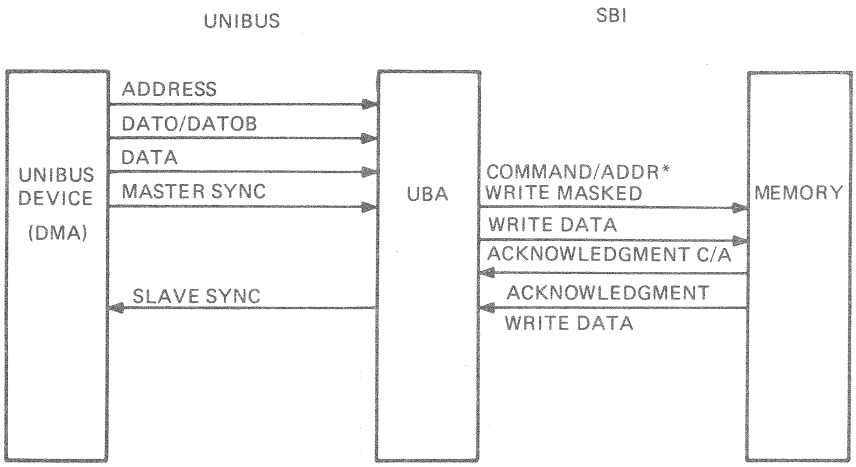


Figure 11-16 Data Sent from a Device to Memory in a DMA Operation Using the Direct Data Path

UNIBUS to SBI Read Through Direct Data Path

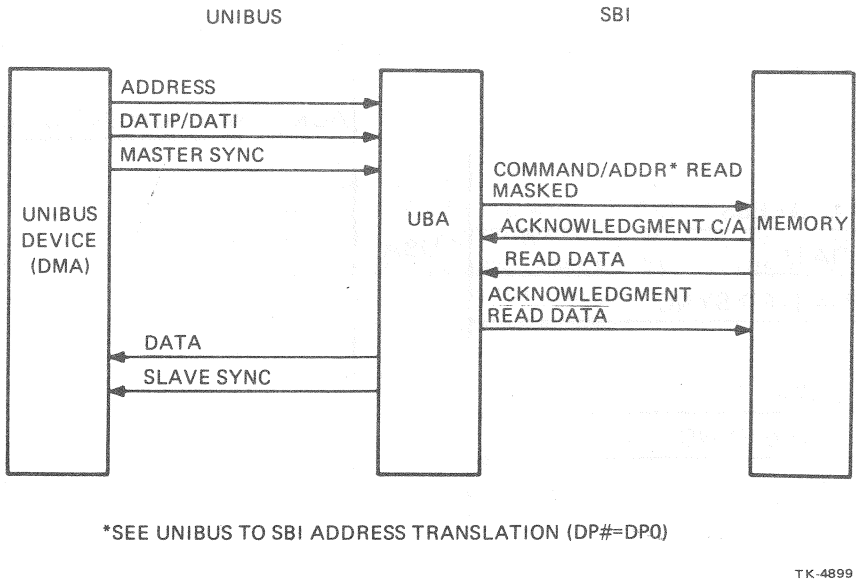


Figure 11-17 Data Read from Memory by a DMA Device Using the Direct Data Path

UNIBUS DMA Write Through Buffered Data Paths

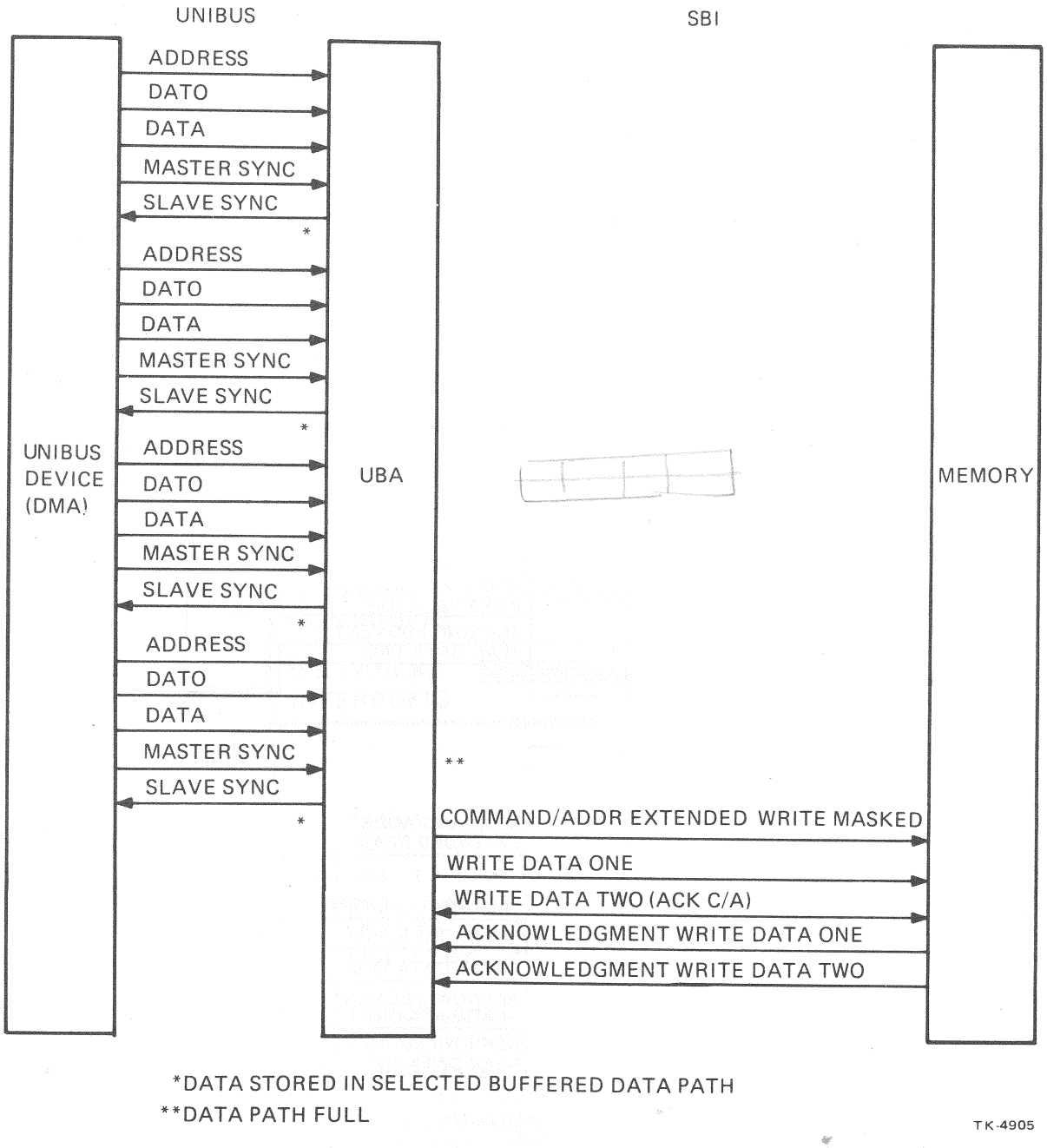


Figure 11-18 Data Written to Memory by a DMA Device Using a Buffered Data Path

Note that both UNIBUS and SBI bandwidths are increased (when compared to four transfers with the direct data path).

UNIBUS to SBI Read Through Buffered Data Paths

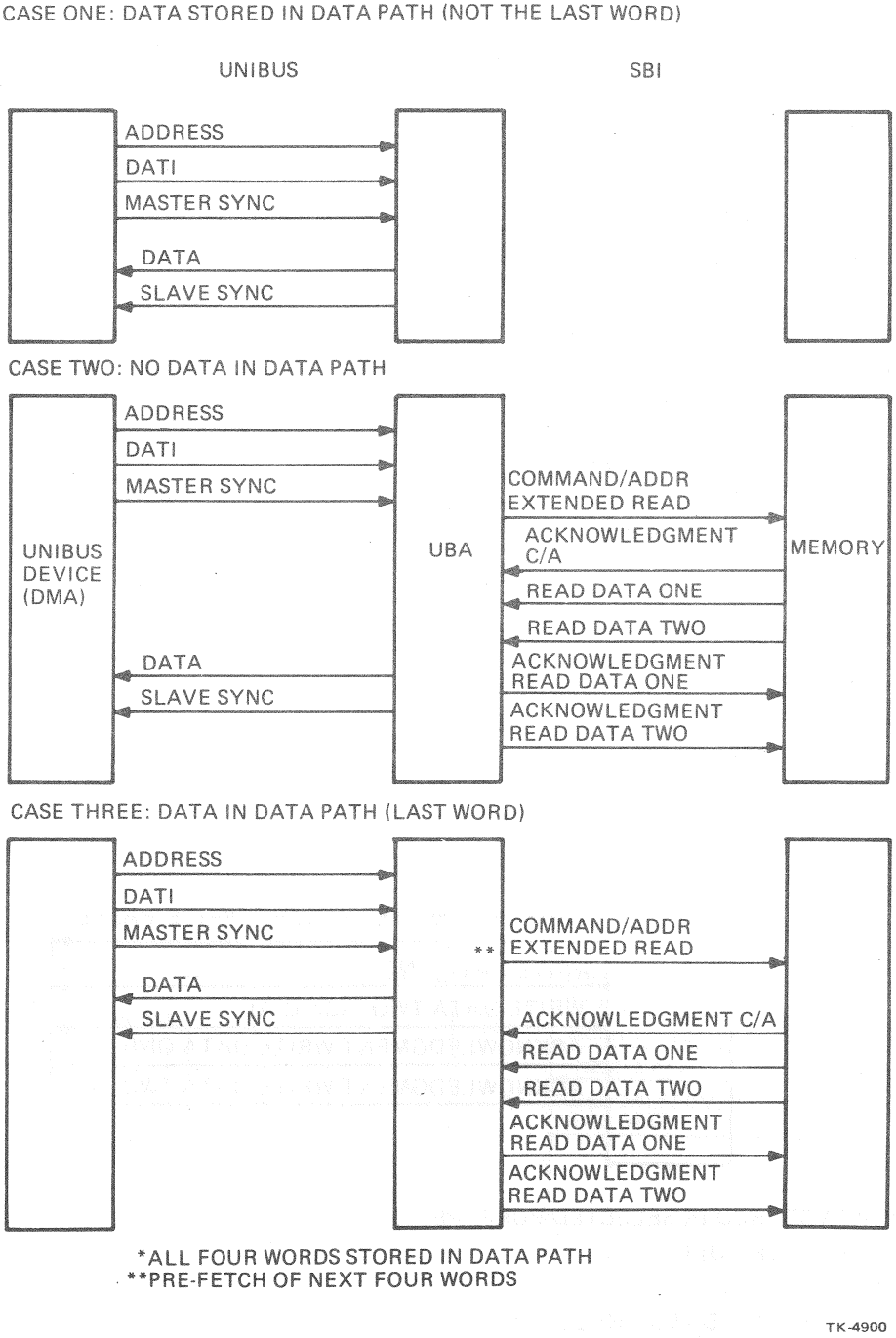


Figure 11-19 Data Read from Memory by a DMA Device Using a Buffered Data Path

Interrupt Dispatching

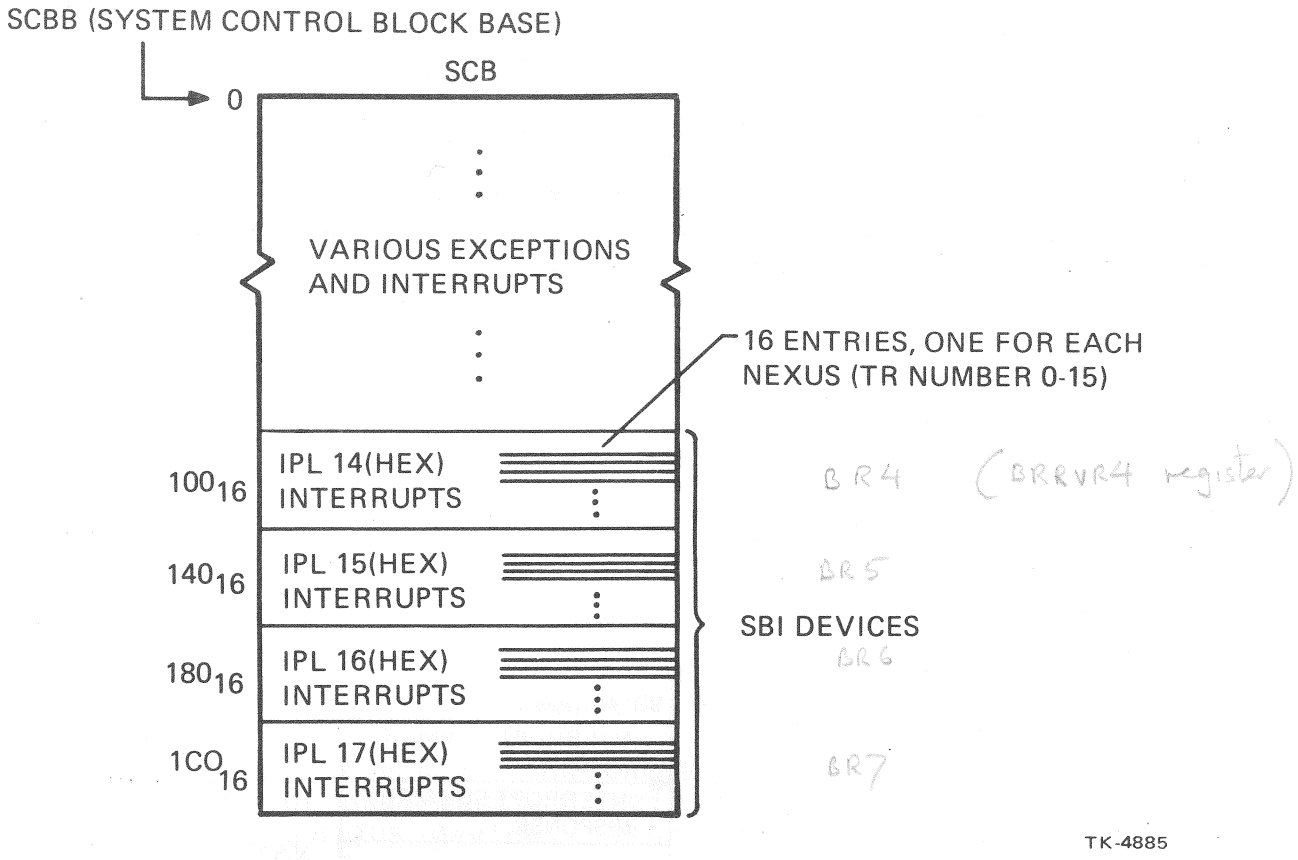
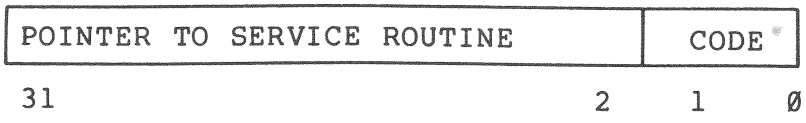


Figure 11-20 Format of SCB

The SCB is a block of longword vectors used by the microcode to dispatch interrupts (software and hardware) and exceptions. The microcode uses the TR number and IPL to index the SCB. The SCBB is an internal register.

Each entry in the SCB contains a pointer in bits <31:02> to a longword aligned interrupt service routine. For UNIBUS or MASSBUS devices, this is the address of a JSB instruction in the ADP. Bits <01:00> encode the stack to be used to process the event.



Code	
ØØ	Use kernel stack; if currently on interrupt stack, use interrupt stack.
Ø1	Use interrupt stack; if this is an exception, raise to IPL 1F(hex).
1Ø	Use WCS; if no WCS, the operation is UNDEFINED (on VAX-11/78Ø, HALT).
11	UNDEFINED; reserved to DIGITAL (on VAX-11/78Ø, HALT).

UNIBUS Interrupt Processing

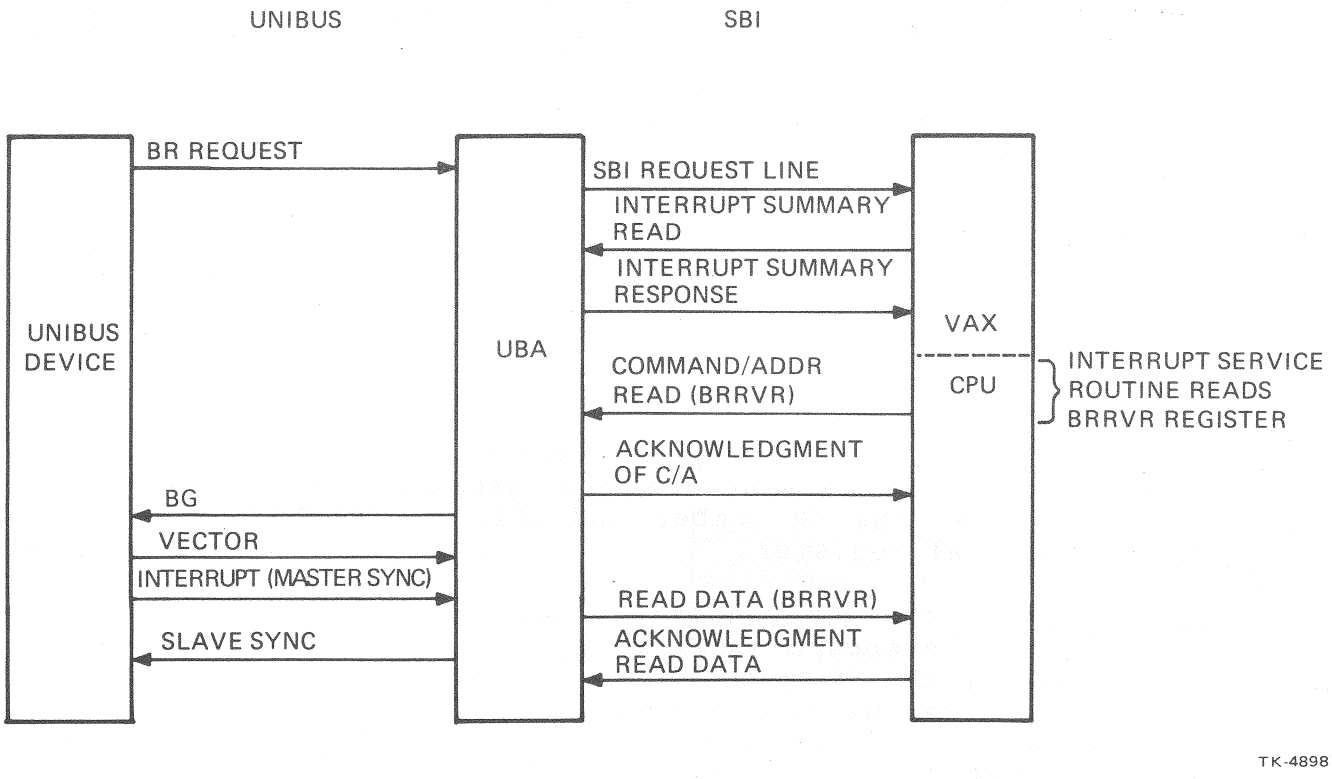


Figure 11-21 Processing a Device Interrupt

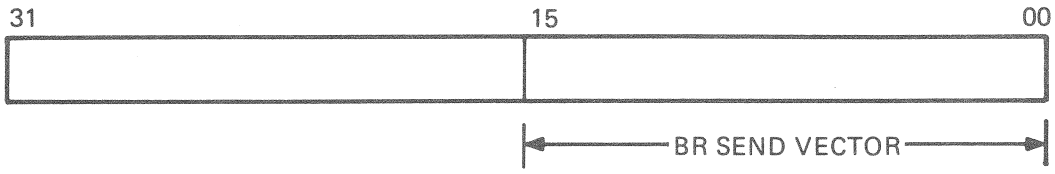
BRRVR Bit Settings

Bit 31	Bit <15:00>	
Ø	Ø	No service required.
Ø	V	UNIBUS service as indicated by vector V.
1	Ø	UNIBUS adapter service required. Read configuration register and status register to determine the service required (see Figure 11-24).
1	V	UNIBUS and UNIBUS adapter service required. 1. Save vector V. 2. Read UBA configuration register and status register. 3. Perform UBA service as indicated by configuration and status register. 4. Index into UNIBUS device service routine with vector V.

V is the vector field of the BRRVR received from the UNIBUS device. Zero is the null vector, indicating that a vector was not received from the UNIBUS device. VMS only recognizes vectors in the range Ø-774 (octal).

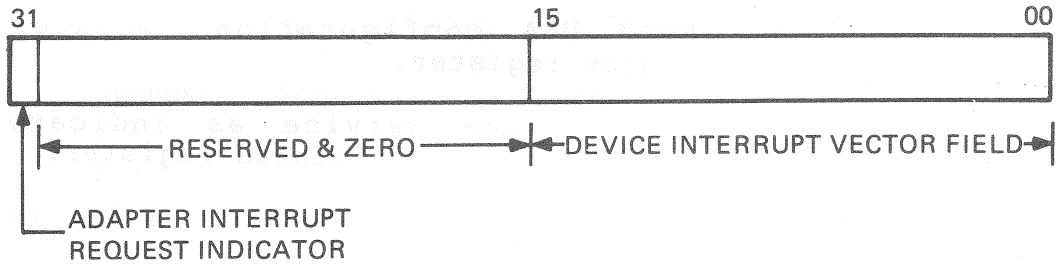
BRSVR

BR SEND VECTOR REGISTER



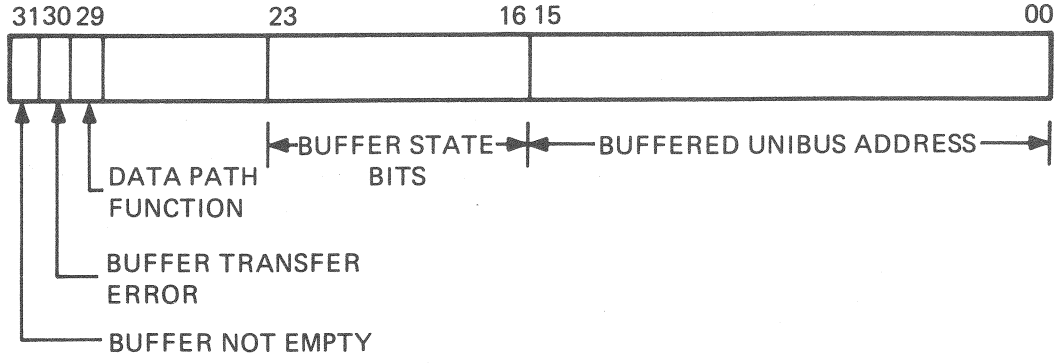
BRRVR

BR RECEIVE VECTOR REGISTER



DPR

DATA PATH REGISTER



TK-4902

Figure 11-22 Format of Several Internal UBA Registers

MASSBUS ADAPTER (MBA)

- Standard SBI interface.
- Standard MASSBUS interface (54 signal lines).
- Up to four MBAs per system.
- Each MBA supports up to eight device controllers.
  - Each disk controller supports one drive.
  - Each tape controller supports up to eight tape drives.
  - Only one controller may transfer data at a time.
- Data transfers up to 128K bytes.
- Does scatter-read/gather-write.
- 32-byte silo.
- Maximum transfer rate: 2 millon bytes/second.
- Prefetches on memory to MBA transfers.
- 192 mapping registers (longwords).
  - Bits 21-30 are reserved, and cannot be written.
  - Contain no byte offset field (the MBA VAR register contains the byte offset).
  - Contain no data path field (the MBA has a single data path).

MBA Operation

The MBA consists of an SBI/MBA interface, internal registers, control paths, and datapaths, as illustrated in Figure 11-23. Data is sent as words, in the order of the first word (bits 15 to Ø), followed by the second word (bits 31 to 16) of a longword. The 32-byte silo is used to buffer data, and data transfers along the SBI, to or from main memory, occur in 8-byte increments. There are four 16-bit MASSBUS transfers per SBI transaction. The MBA accepts only longword aligned reads and writes to its external or internal registers.

The MBA address space has two sets of registers: internal and external. The eight MBA internal registers are physically located in the MBA. The external registers are located in the MASSBUS drives, and are drive-dependent. The internal registers are used to monitor the MBA and operating status conditions, as well as to control certain phases of the data transfers between the SBI and the MASSBUS device, such as:

- maintaining a byte count to ensure that all data to be transferred is accounted for.
- converting virtual addresses to physical addresses for referencing data in memory (see Figure 11-24).

MBA Structure

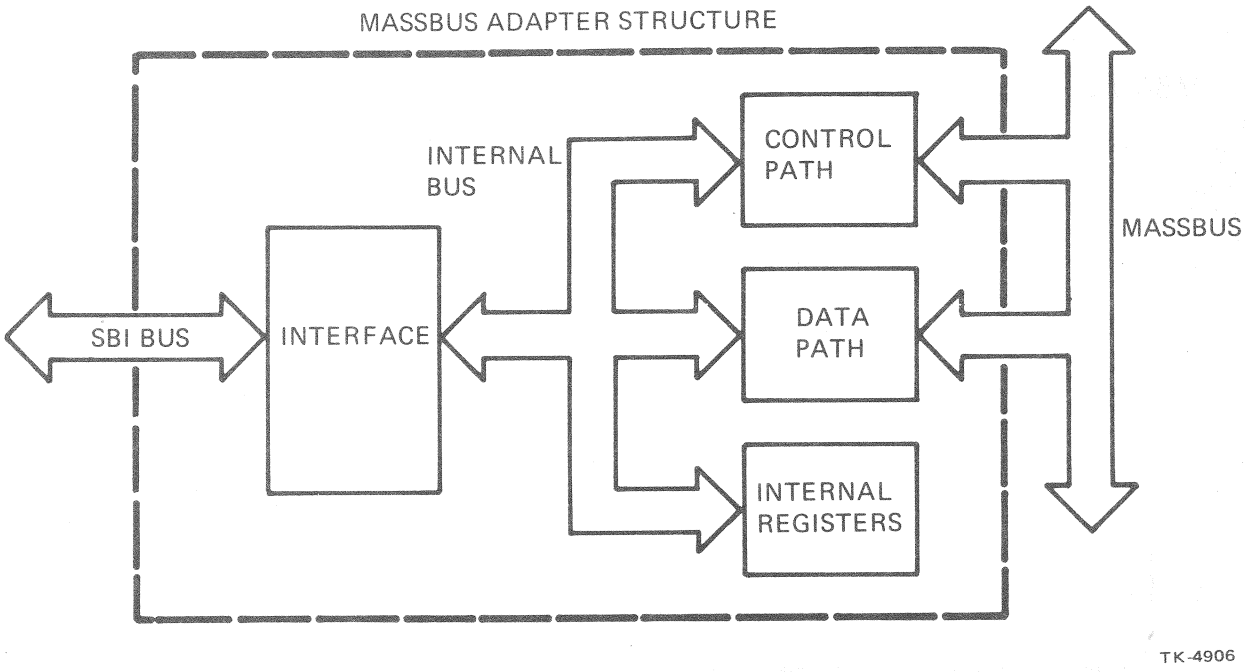


Figure 11-23 Simplified Block Diagram of the MBA

The MBA Internal Registers are:

1. MBA Configuration Register (CSR)
2. MBA Control Register (CR)
3. MBA Status Register (SR)
4. MBA Virtual Address Register (VAR)
5. MBA Byte Count Register (BCR)
6. MBA Diagnostic Register (DR)
7. MBA Selected Map Register (SMR)*
8. MBA Command Address Register (CAR)*

* Read-only, valid only during data transfers.

MBA Address Translation

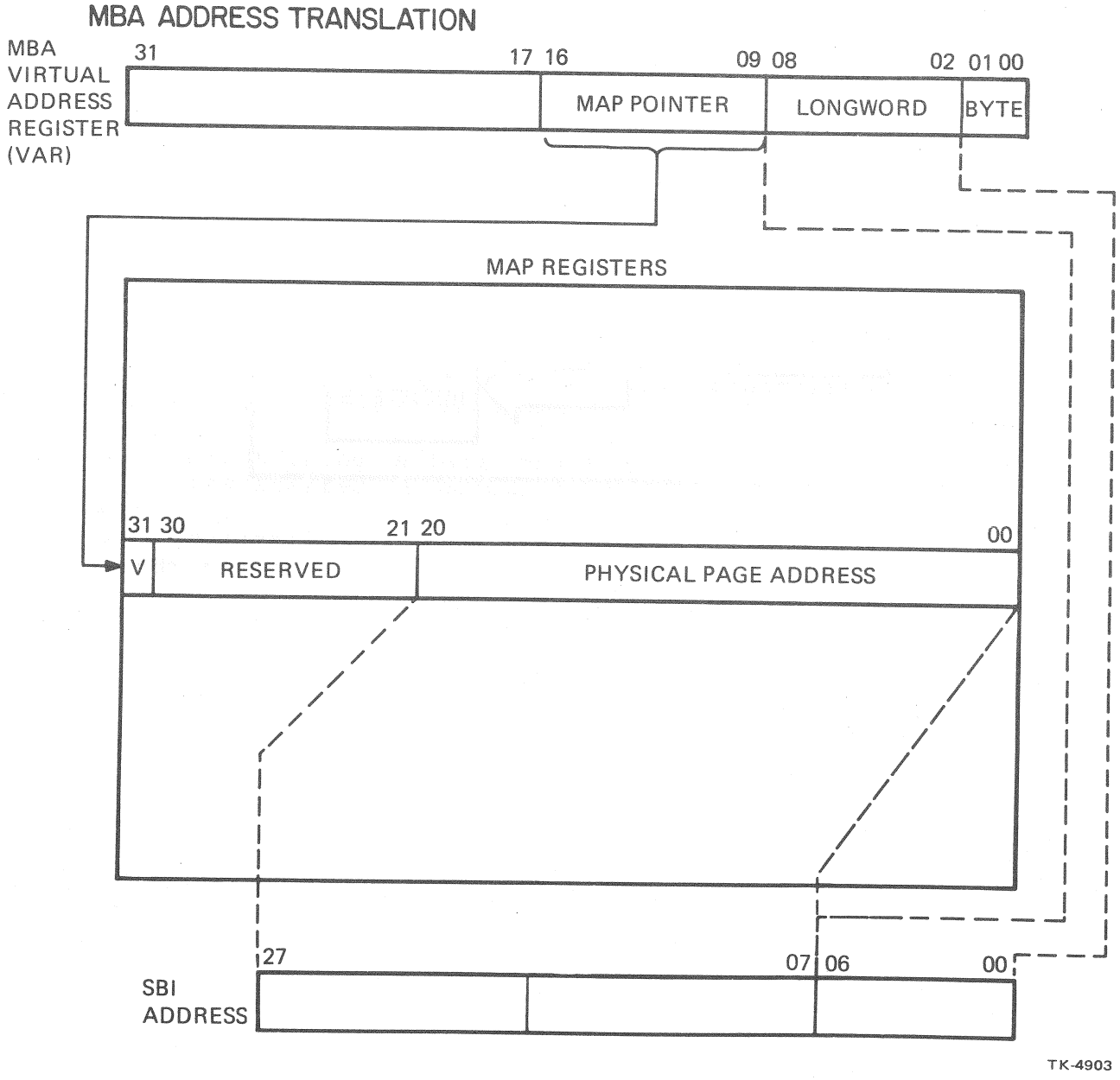


Figure 11-24 MBA Mapping a Virtual Address to a Page Frame Number

MBA Address Space

- Each MBA has a 2048-longword physical address space.
- Bits 10 and 11 select a region of MBA nexus address space:

Bits 11 and 10		Meaning
0	0	MBA internal registers
0	1	MBA external registers
1	0	Map register

- If external register:
  - Bits 0-6 select the register
  - Bits 7-9 select the unit (or subcontroller)
- If map register:
  - Bits 0-9 contain map register index
- Bits 13-16 specify TR number for MBA.
- The address of the start of MBA space (where MBA internal registers start) depends on the TR number at which the MBA is installed.

Location of MASSBUS Registers

Assume the MBA is installed at TR 1Ø. Figure 11-25 illustrates how MBA address space is structured:

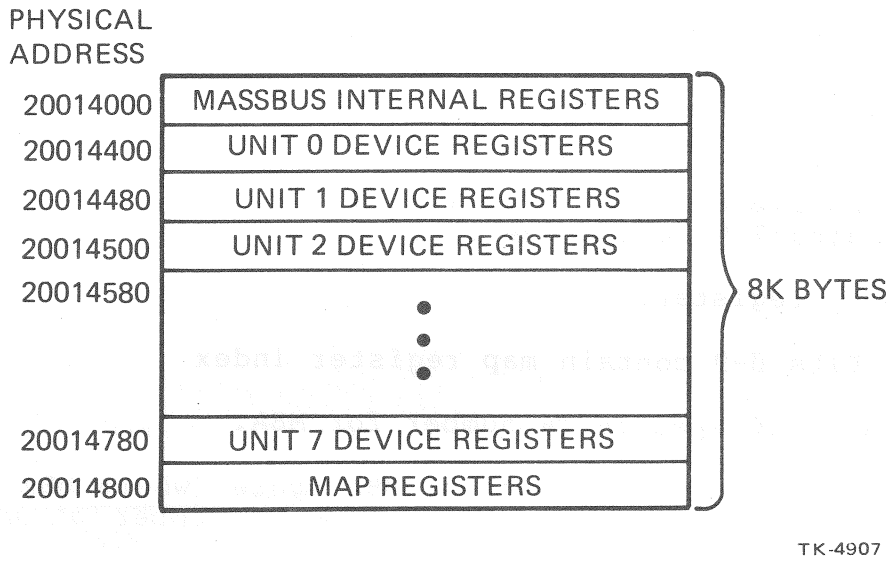


Figure 11-25 Structure of MBA Address Space

To address map register 6 in the MBA at TR 1Ø, the driver constructs the address 2ØØ14Ø86.

- 2ØØ14ØØØ indicates TR1Ø
- 8ØØ indicates map register (bits 1Ø and 11)
- 6 indicates map register 6

To address a device register, the driver uses:

2ØØ14ØØ .OR. (device unit) .OR. (register offset)

(The 4ØØ indicates device register in bits 1Ø and 11.)

digital

EY-2278E-MK-0001
Printed in U.S.A.