VAX-11/780 Console Interface Board

Technical Description

EK-KC780-TD.001

EK-KC780-TD-001

VAX-11/780

Console Interface Board

Technical Description

digital equipment corporation • maynard, massachusetts

1st Edition, March 1978

Copyright © 1978 by Digital Equipment Corporation

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.

The following are trademarks of Digital Equipment Corporation, Maynard, Massachusetts:

DEC	DECtape	PDP
DECCOMM	DECUS	RSTS
DECsystem-10	DIGITAL	TYPESET-8
DECSYSTEM-20	MASSBUS	TYPESET-11
		UNIBUS

CONTENTS

		Page
CHAPTER 1	**INTRODUCTION**	
1.1	SCOPE	1-1
1.2	CONSOLE SUBSYSTEM OVERVIEW	1-2
1.3	CONSOLE MODES	1-2
1.4	VAX-11/780 CPU STATUS	1-5
1.5	PANEL FUNCTIONS	1-5
1.6	ID (INTERNAL DATA) BUS OVERVIEW	1-7
1.6.1	ID Bus Structure and Operation	1-7
1.7	V BUS OVERVIEW	1-8
1.8	Q BUS OVERVIEW	1-10
CHAPTER 2	**CIB FUNCTIONAL DESCRIPTION**	
2.1	CIB OVERVIEW	2-1
2.2	CONSOLE/VAX-11 INTERACTION VIA THE CIB	2-1
2.2.1	VAX-11/780 CPU Status: Halted and Running	2-3
2.2.2	LSI-11 Operating Modes	2-3
2.2.3	Use of CIB Registers	2-5
2.3	ID BUS REGISTERS ON THE CIB	2-5
2.3.1	System Identification Register	2-5
2.3.2	Receiver Control/Status Register	2-5
2.3.3	TO ID Register - R/O	2-5
2.3.4	Transmit Control/Status Register	2-6
2.3.5	From ID Register - W/O	2-6
2.3.6	Use of the TO ID and FM ID Registers	2-6
2.4	Q BUS REGISTERS ON THE CIB	2-8
2.4.1	Read Only Memory	2-8
2.4.2	ID DATA LO and ID DATA HI Registers	2-8
2.4.3	Receiver Done Register - R/W	2-8
2.4.4	Transmitter Ready Register - R/W	2-8
2.4.5	TO ID LO and TO ID HI Registers - R/W	2-11
2.4.6	FM ID LO and FM ID HI Registers - R/O	2-11
2.4.7	ID Control Status Register, ID C/S	2-11
2.4.8	Machine Control Register, MCR	2-12
2.4.9	Miscellaneous Control and Status Register, MCS	2-14
2.4.10	V Bus Register	2-16
2.5	DIALOGUE: LSI-11 PROGRAM I/O - VAX-11/780 MACROCODE	2-16
2.5.1	LSI-11 Sends a Character to the VAX-11/780	2-16
2.5.2	VAX-11/780 Sends a Character to the LSI-11	2-18

CONTENTS

		Page
2.6	DIALOGUE: LSI-11 CONSOLE I/O MODE - VAX-11/780 MICROCODE	2-18
2.7	DIRECT ID BUS REFERENCE	2-21
2.7.1	ID Bus Reference with the Clock Running	2-21
2.7.2	ID Bus Reference with the Clock Stopped	2-21
2.7.3	ID Bus Reference without use of Console Command Language Instructions	2-22
CHAPTER 3	**CIB DETAILED LOGIC DESCRIPTION**	
3.1	CONSOLE INTERFACE BOARD LOGIC	3-1
3.2	ID BUS INTERFACE LOGIC	3-1
3.3	Q BUS ADDRESS DECODER AND TIMING LOGIC	3-1
3.3.1	Interface Address Decoder Logic	3-1
3.3.2	QMUX Logic	3-4
3.3.3	One of Twelve Decoder Logic	3-4
3.3.4	Strobe Signals Developed on a Q Bus Read Transfer	3-7
3.3.5	Clocking the Registers on a Q Bus Write Transfer	3-7
3.3.6	Transfer Synchronization Logic	3-7
3.3.7	Q Bus Transfers when the VAX-11/780 CPU is Stopped	3-13
3.4	ROM LOGIC	3-13
3.5	CIB REGISTER LOGIC	3-13
3.5.1	TO ID Register Logic	3-13
3.5.2	FM ID Register Logic	3-17
3.5.3	ID C/S Register Logic	3-17
3.5.4	MCR Register	3-20
3.5.5	MCS Register Logic	3-20
3.5.6	V Bus Register Logic	3-20
3.5.7	ID Data Register Logic	3-24
3.5.8	Interrupt Logic	3-25
3.6	TYPICAL CIB TRANSFER OPERATIONS	3-29
APPENDIX A	**Q BUS TECHNICAL DESCRIPTION**	
A.1	GENERAL	A-1
A.2	TRANSFER OPERATIONS ON THE Q BUS	A-3
A.2.1	Input Operations	A-4
A.2.2	Output Operations	A-5
A.2.3	Interrupts	A-5
A.2.4	Bus Initialization	A-8
A.2.5	Power-Up/Power-Down Sequence	A-8
A.2.6	Memory Refresh Operation	A-8

FIGURES

Figure No.	Title	Page
1-1	Console Subsystem Configuration	1-3
1-2	VAX-11/780 Control Panel	1-6
1-3	ID Bus Control Logic, Block Diagram	1-9
1-4	V Bus Block Diagram	1-9
2-1	CIB, Block Diagram	2-2
2-2	CIB Data Paths, Simplified Block Diagram	2-3
2-3	VAX-11/780 and LSI-11 Interaction and Operating Mode Combinations	2-4
2-4	ID Bus Registers on the CIB	2-7
2-5	Use of the TO ID and FM ID Registers When the VAX-11/780 CPU is Running	2-7
2-6	Lower Eight Q Bus Registers: Addresses and Bit Configurations	2-9
2-7	Upper Eight Q Bus Registers: Addresses and Bit Configurations	2-10
2-8	LSI-11 Sends a Character to the VAX-11/780 CPU, Flowchart	2-17
2-9	VAX-11/780 Sends a Character to the LSI-11, Flowchart	2-19
2-10	LSI-11 Response to a VAX-11/780 CPU Halt	2-20
3-1	ID Bus Address Decoder Logic	3-2
3-2	ID Data Multiplexer Logic	3-3
3-3	Q Bus Address Decoder Logic Interface Chips	3-4
3-4	QMUX Logic	3-5
3-5	One of Twelve Decoder Logic	3-6
3-6	Register Strobe Logic	3-8
3-7	Transfer Synchronizer Logic	3-9
3-8	DATO/B Transfer Timing on the Q Bus	3-10
3-9	Synchronized DATO Transfer Timing Diagram	3-11
3-10	DATI Transfer Timing on the Q Bus	3-12
3-11	Synchronized DATI Transfer Timing Diagram	3-12
3-12	ROM Logic	3-14
3-13	ROM Address Selection Logic	3-15
3-14	TO ID Register Logic	3-16
3-15	Development of CIBU CLK FM ID H	3-18
3-16	FM ID Register Logic	3-18
3-17	ID C/S Register Logic	3-19
3-18	Machine Control Register Logic	3-21
3-19	MCS Register Logic	3-22
3-20	V Bus Register Logic	3-23
3-21	V Bus Clock Timing Diagram	3-24
3-22	ID DATA HI and ID DATA LO Register Logic	3-25
3-23	VAX-11/780 CPU Interrupt Logic	3-26
3-24	Ready and Done Multiplexer Logic	3-27
3-25	LSI-11 Interrupt Logic	3-28
3-26	LSI-11 Sets TX READY, DATOB Transfer on Q Bus	3-30
3-27	VAX-11/780 Software Sends a Character to the FM ID Register for Transmission to the Console Terminal, Write Transfer on the ID Bus	3-31

FIGURES (CONT)

Figure No.	Title	Page
3-28	Interrupt Dialogue on the Q Bus	3-32
3-29	LSI-11 Interrupt Subroutine Reads the FM IDLO Register, DATI Transfer on Q Bus	3-33
3-30	LSI-11 Writes a Character to the TO IDLO Register for Transmission to the VAX-11/780 Software, DATO Transfer on the Q Bus	3-34
3-31	VAX-11/780 Interrupt Subroutine Reads the TO ID Register, Read Transfer on the ID Bus	3-35
A-1	DATI Bus Cycle	A-4
A-2	DATIO or DATIOB Bus Cycle	A-6
A-3	DATO or DATOB Bus Cycle	A-7
A-4	Interrupt Request/Acknowledge Sequence	A-9

TABLES

Table No.	Title	Page
1-1	Related Hardware Manuals	1-1
1-2	Console Functions	1-4
1-3	ID Bus Register Address Assignment	1-8
2-1	Clock Frequency Control	2-13
2-2	Uses of PROCEED	2-14
2-3	Panel Switch Sense Bit Functions	2-15
3-1	Generation of the CLK FM ID H Signal	3-17
3-2	Generation of the SDMS Clock H Signal	3-24
3-3	Interrupt Related Register Bit Functions	3-29
A-1	Q Bus Signals	A-1

CHAPTER 1
INTRODUCTION

1.1 SCOPE

This manual provides an overview of the VAX-11/780 Console Subsystem and a comprehensive description of the Console Interface Board on the functional and logical levels. A description of the Q bus is provided as an appendix. The manual will serve as a resource for appropriate branch and support level courses of the Field Service and Manufacturing training programs and as a field reference. Table 1-1 lists related hardware manuals.

Table 1-1 Related Hardware Manuals

Title	Document	Notes
Microcomputer Handbook	EB 06583	Available on hard copy.*
LSI-11, PDP-11/03 User's Manual	EK-LSI11-TM-003	Available on hard copy. Hard copy ships with device.*
LA36/LA35 DECwriter II User's Manual	EK-LA3635-OP-003	In Microfiche Library. **Available on hard copy. Hard copy ships with device.
KA780 Central Processor Technical Description	EK-KA780-TD-PRE	In Microfiche Library.**

*The documents can be ordered from:

Digital Equipment Corporation
444 Whitney Street
Northboro, MA 01532
Attention: Communication Services NR2/M15
Customer Services Section

**For information concerning Microfiche Libraries, contact:
Digital Equipment Corporation
Micropublishing Group, PK3-2/T12
129 Parker Street
Maynard, MA 01754

1.2 CONSOLE SUBSYSTEM OVERVIEW

Five major components make up the VAX-11/780 Console Subsystem: an LSI-11 microprocessor (KD11-F), which includes a 4K by 16 semiconductor RAM; a single floppy disk and a controller (RXV11); a terminal and one or two serial line unit interfaces (DLV11-E), one for communication with a remote terminal (optional); a VAX-11/780 CPU console interface (CIB), which includes 4K by 16 bits of ROM for the LSI-11; and a control panel on the VAX-11/780 cabinet. The Console Subsystem provides three major functions:

- Traditional *lights* and *switches* functions such as EXAMINE, DEPOSIT, HALT, START, and Single Instruction.

- Diagnostic and maintenance functions, including the capability to load diagnostic microcode into writable control store (WCS), control execution and monitor results, control single step clock functions, and examine key system points via a serial diagnostic bus (V bus).

- Materialize the terminals and floppy disk I/O registers in the processor register space. These console I/O registers are located on the CIB module; they are addressable both from the LSI-11, via the Q bus, and from the VAX-11/780 CPU, via the ID bus. This port is therefore used for terminal and floppy disk transfers to the VAX-11/780 CPU and all other software defined communications between the console and the VAX-11/780 CPU.

The user can perform the lights and switches and diagnostic and maintenance functions through a set of keyboard commands and responses at the terminal. The LSI-11 in turn interprets the commands and controls and monitors the VAX-11/780 CPU through a set of control/status and data registers on the CIB module in the Q bus I/O space. These registers connect to the ID bus, the V bus, and points within the VAX-11/780 CPU and on the control panel. Figure 1-1 shows the console subsystem configuration.

1.3 CONSOLE MODES

The LSI-11 normally operates in one of two modes, each of which is implemented through a set of MACRO-11 routines. When the LSI-11 is in the program I/O mode it functions as a character handler, passing characters between the console terminal and the VAX-11/780 CPU. The console terminal therefore functions as the VAX-11/780 VMS operator's terminal.

When the LSI-11 is in the console I/O mode, it interprets all console terminal output in order to perform the lights and switches and maintenance functions and implement the console command language (CCL) capability.

Table 1-2 lists the console functions implemented through the console command language.

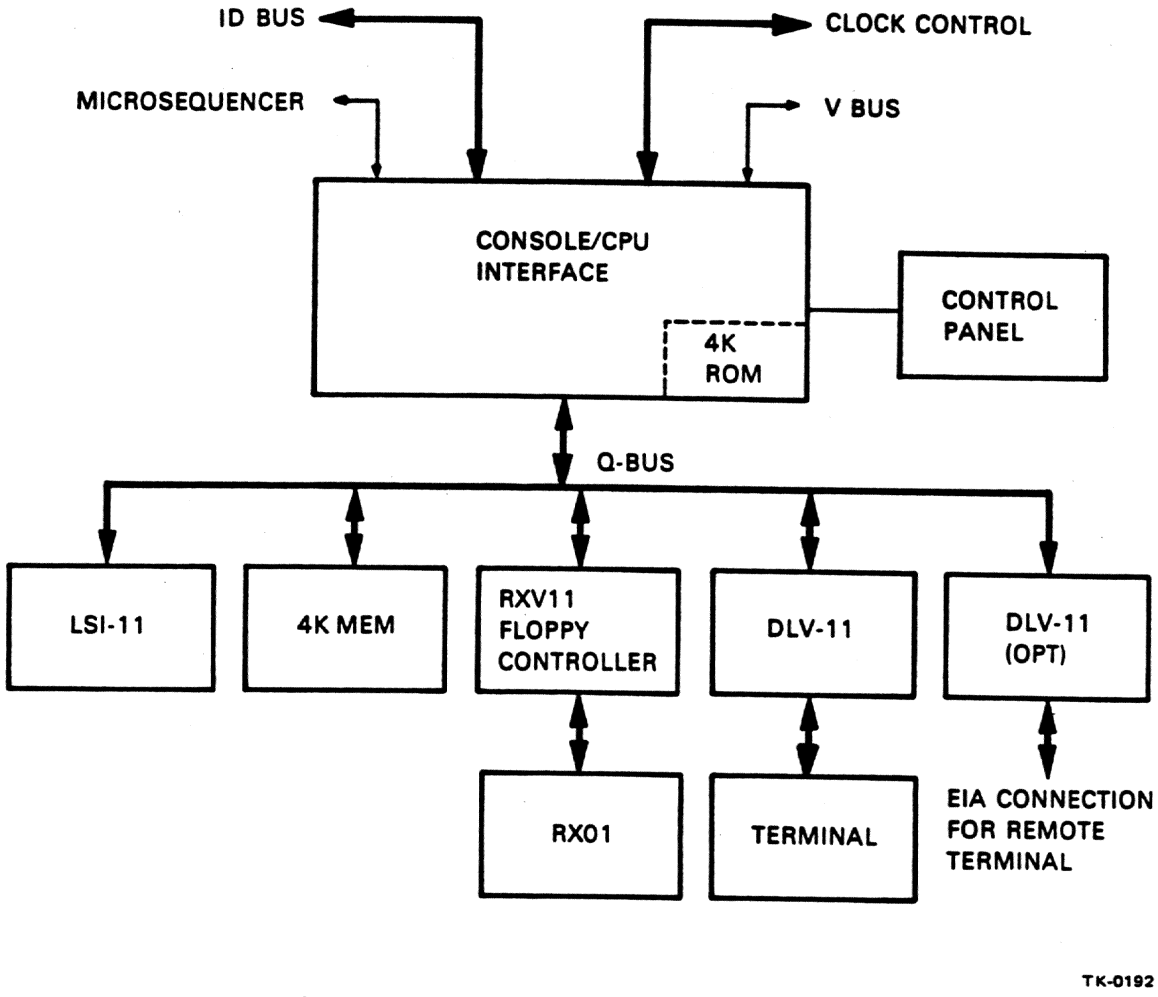


Figure 1-1 Console Subsystem Configuration

Table 1-2 Console Functions

Virtual examine byte, word, longword
Virtual deposit byte, word, longword
Examine general register
Deposit general register
Examine processor register
Deposit processor register
Continue
Initialize TB, cache, etc. (result of CPU initialize signal)
Quad clear
SBI unjam
Stop clock
Start clock
Step one time state
Step one SBI cycle (stops in CPT 0)
Select one of four clock frequencies
Assert VAX-11/780 CPU initialization signal
Interrupt VAX-11/780 CPU (terminal registers)
Halt at end of current instruction
Step single instructions
Stop clock upon microbreak match (in CPT0 of match state)
Force UPC<12> to WCS on microtrap
Force NOP on selected ROM fields
Clock V bus
Assert V bus loopback bit
Load V bus shift registers
Read V bus serial channels
Sense positions of auto-restart, boot, lock, and remote switches
Time-out from VAX-11/780 interrupt strobe signal, used to assert RUN light
Provide a write-only register on the ID bus (FM ID) (as a responder)
Provide a read-only register on the ID bus (TO ID) (as a responder)
Write to any ID bus address (requires console control and clock running)
Read any ID bus address (when clock is stopped or running)
Synchronize use of FM ID and TO ID via ready and done bits
Read clock states
Sense assertion of console acknowledge (reply to halt request)
Sense when system clock is stopped
Maintenance return (forced jump to UPC off top of microstack)
Turn floppy disk power on or off
Read ID bus address and direction lines (clock stopped only)
Materialize JMP into ROM at 163000 and 163002 or 173000 and 173002_8 (LSI-11 I/O address space)

Table 1-2 Console Functions (Cont)

The capability to read/write at any ID bus address permits register accesses to implement the following functions (and others):

- Push microstack
- Pop microstack
- Write microbreak
- Read microbreak
- Read WCS address
- Write WCS address
- Write WCS data
- Read WCS status

Specify defaults

- numeric radix
- addressing modes
- data length

Display console and CPU status

Control the number of fill characters to be added after special characters sent to the console terminal.

Command repetition

Execute commands from an indirect command file (an ASCII file containing console commands)

Invoke microdiagnostics

Implement the remote console access command set.

1.4 VAX-11/780 CPU STATUS

The VAX-11/780 CPU may be either halted or running. When it is halted, it enters the console command mode, executing a microcode loop which enables the console to perform the lights and switches and maintenance functions. When the VAX-11/780 CPU is running, it is running at the ISP level, executing macroinstructions.

1.5 PANEL FUNCTIONS

With the exception of the OFF position of the key switch and the POWER light, the lights and switches on the control panel are connected to other parts of the computer through the miscellaneous control status (MCS) register, a Q bus register on the Console Interface Board. The OFF switch controls a relay in the power supply. The POWER light is connected between +5 volts and ground. ATTN indicates that the VAX-11/780 CPU is not running and/or needs operator intervention. RUN indicates that the VAX-11/780 CPU is strobing interrupts, and thus not caught in a microcode loop. The REMOTE light reflects the position of the 5 position key switch, indicating which of the two console terminals is on line. Figure 1-2 shows the control panel.

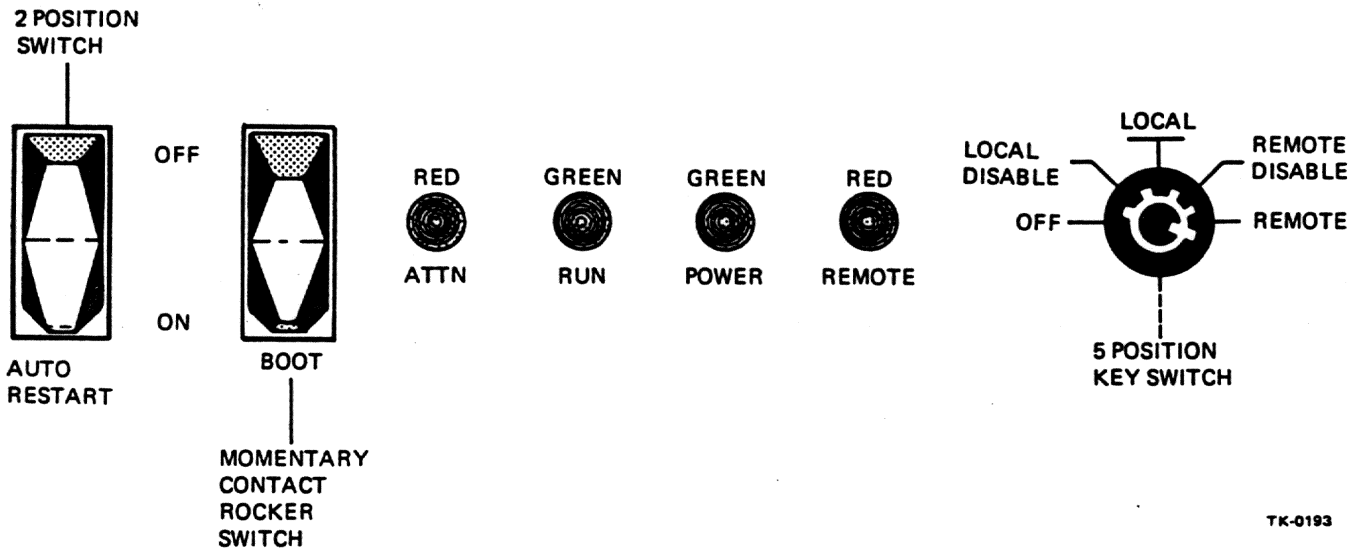


Figure 1-2 VAX-11/780 Control Panel

AUTO RESTART is a two position switch which controls bit 2 in the MCS register. BOOT is a momentary contact rocker switch, which when pressed sets MCS bit 11; LOCAL DISABLE, LOCAL, REMOTE DISABLE, and REMOTE control MCS bits 1 and 0. These MCS register bits in turn control and initiate specific LSI-11 operations, as follows:

- AUTO RESTART – when on, causes the LSI-11 processor to jump to a designated ROM location on power up.
- BOOT – causes the LSI-11 to boot VMS, the operating system. When the boot routine is completed, the console comes up in the program I/O mode.
- LOCAL – when the 5 position key switch is in LOCAL, it enables the LSI-11 to designate the local terminal as the console terminal or as the VAX-11/780 VMS operator's terminal (enabling either the console I/O or the program I/O mode, according to the wishes of the terminal operator).
- LOCAL DISABLE – this position of the switch causes the LSI-11 to select the local terminal as the VAX-11/780 VMS operator's terminal, while inhibiting console operation in the console I/O mode.
- REMOTE – this switch position causes the LSI-11 to select the remote terminal, allowing operation in both the console I/O and the program I/O modes.
- REMOTE DISABLE – when the key switch is in REMOTE DISABLE, it causes the LSI-11 to designate the remote terminal as the VAX-11/780 VMS operator's terminal, while disabling use of the terminal in the console I/O mode.

1.6 ID (INTERNAL DATA) BUS OVERVIEW

The ID bus is a high speed data path between the major functional areas of the VAX-11/780 CPU and provides the following functions:

1. Transfers data to and from the internal registers of the VAX-11/780 CPU and the Translation Buffer.
2. Transfers data in the form of displacement and short literals from the Instruction Buffer to the VAX-11/780 CPU's data paths and FPA.
3. Transfers data between the VAX-11/780 CPU's data paths and the FPA.
4. Transfers data from the internal registers to the CIB under console control during the maintenance operation.

1.6.1 ID Bus Structure and Operation

The ID bus consists of 32 data lines, 6 address lines, and 1 write control line. The address lines specify which internal register has been designated as the source or destination. Address assignments are listed in Table 1-3.

The write control line specifies directional control, indicating whether an internal register is to be read onto the bus or data is to be clocked from the bus into an internal register.

During a normal read operation, data is transferred from the addressed internal register to the Q register of the data paths via the ID bus. During a normal write operation, data is transferred from the D register of the data paths to the addressed internal register via the ID bus.

Table 1-3 ID Bus Register Address Assignment

00	IBUF DATA	20	USTACK
01	TIME OF DAY	21	UBREAK
02	- RSVD -	22	WCS ADDRESS
03	SYSTEM ID	23	WCS DATA/STATUS
04	CNSL RXCS	24	P0BR
05	CNSL RXDB (TO ID)	25	P1BR
06	CNSL TXCS	26	SBR
07	CNSL TXDB (FROM ID)	27	RSVD FOR SYS SPACE
08	DQ (ID MAINT ONLY)	28	KSP
09	NEXT INTERVAL REGISTER	29	ESP
0A	CLOCK CS	2A	SSP
0B	INTERVAL COUNTER	2B	USP
0C	CES	2C	ISP
0D	VECT	2D	FPDA
0E	SIR	2E	D.SV
0F	PSL	2F	Q.SV
10	TBUF DATA	30	T0
11	-RSVD-	31	T1
12	TBUF REG 0	32	T2
13	TBUF REG 1	33	T3
14	ACC REG 0	34	T4
15	ACC REG 1	35	T5
16	ACC MAINT REGISTER	36	T6
17	ACC CONTROL/STATUS	37	T7
18	SBI SILO	38	T8
19	SBI ERR REGISTER	39	T9
1A	SBI TIMEOUT ADDRESS	3A	PCBB
1B	SBI FAULT/STATUS	3B	SCBB
1C	SBI SILO COMPARATOR	3C	P0LR
1D	MAINTENANCE	3D	P1LR
1E	CACHE PARITY	3E	SLR
1F	-RSVD-	3F	RSVD FOR SYS SPACE

The ID bus may be controlled from the console interface logic in a maintenance mode operation as shown in Figure 1-3. This allows access to writable control store, the microstack (USTACK) and internal registers from the console. In maintenance mode operation only, the D and Q registers of the data paths may be addressed as internal registers over the ID bus. Note that the left and right sides of the address and direction lines are functionally identified, since they are separated only by buffer driver gates in the VAX-11/780 CPU.

When the Console Interface Board generates the ID MAINT signal, it initiates a maintenance operation, allowing the console to assert ID bus address and write signals (and data, if appropriate).

1.7 V BUS OVERVIEW

The V bus consists of eight serial data lines, a load signal line, a clock signal line, and a self test line. Each of the participating VAX-11/780 CPU modules contains at least one V bus shift register chip. The data input lines to the shift register monitor specific test points on the CPU module, as shown in Figure 1-4. The LOAD signal causes the shift register to parallel load from the test points when the VAX-11/780 CPU is in a stable condition. The clock signal can then be used to read the latched data serially from each of the shift registers into a register on the CIB. The LSI-11 must read the register before clocking in the next serial bit from each of the shift registers. See Sections 2 and 3 for more detail.

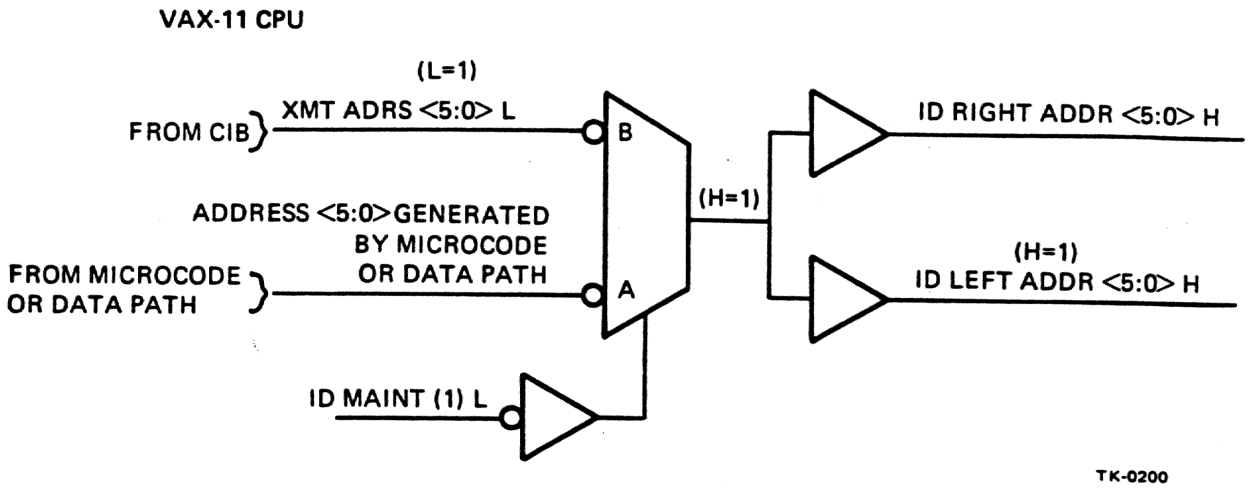


Figure 1-3 ID Bus Control Logic, Block Diagram

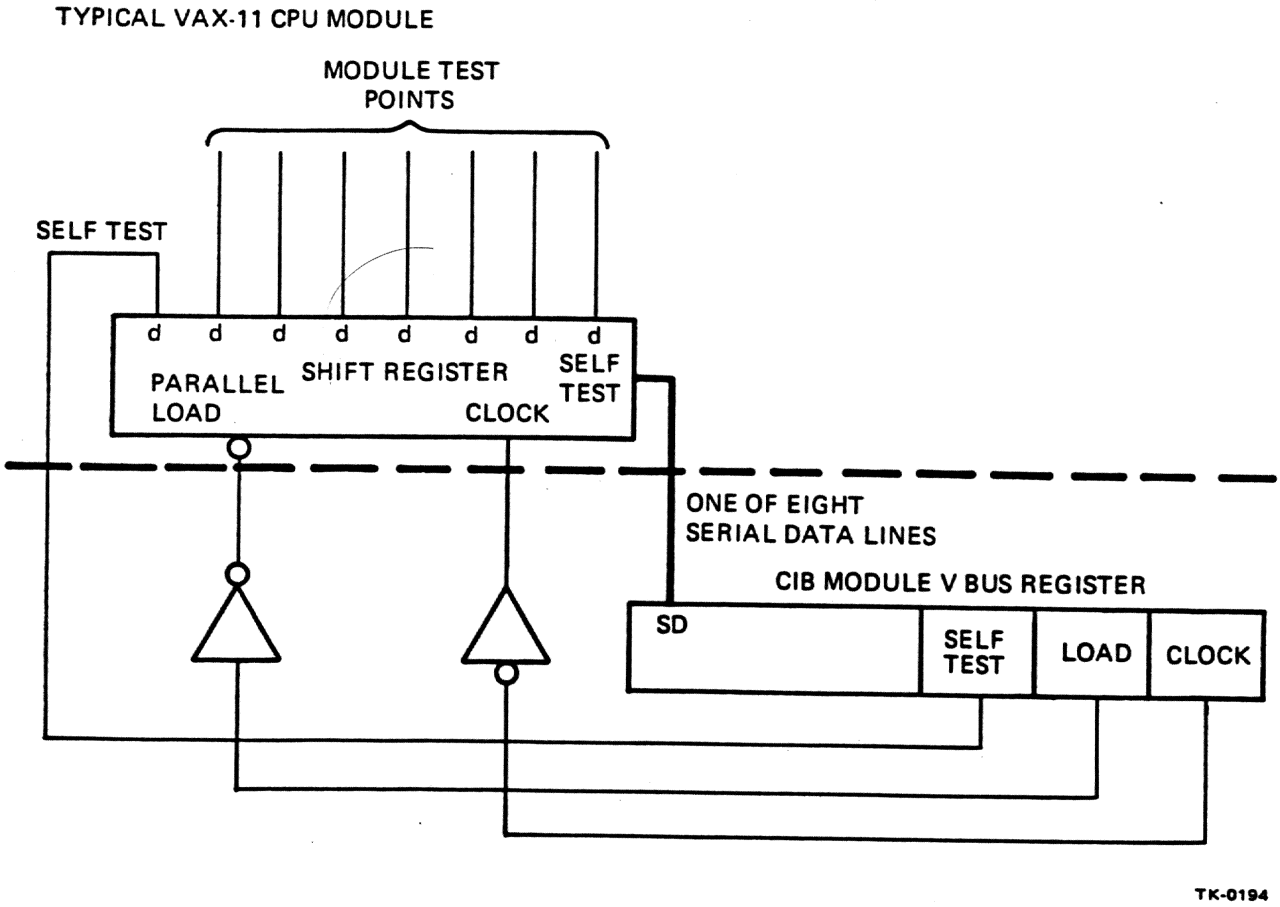


Figure 1-4 V Bus Block Diagram

1.8 Q BUS OVERVIEW

The Q bus (LSI-11 bus) connects the LSI-11 processor (and its ROM and RAM memories), the console terminal interfaces, and the floppy disk interface to the Console Interface Board, and thus to the VAX-11/780 CPU. The 16 address signals and 16 data signals share the same bus lines. Fourteen other LSI-11 signal lines are used in the VAX-11/780 configuration for control signals (note that the DMA control lines are not used).

A master-slave relationship defines communication between the processor and the other devices on the bus. Each control signal issued by a master device must be acknowledged by a slave device in order to complete a transfer. The LSI-11 processor must therefore become bus master in order to read or write any interface register or memory location on the Q bus. The Q bus permits an addressing structure in which control, status, and data registers for peripheral devices are directly addressed as memory locations. Therefore, all operations on these registers are performed by normal memory reference instructions. No system clock is used on the Q bus, and all communications on it are asynchronous. However, when one of the interface units such as the serial line interface for the console terminal must transfer data (i.e., a character) to or from the LSI-11 processor, it must interrupt the processor and thereby invoke a service routine which will handle the actual data transfer.

Note that the serial line interfaces and the floppy disk interface cannot communicate directly with the Console Interface Board, nor can the CIB communicate directly with them. All transfers initiated from the interfaces begin with interrupts to the LSI-11 processor.

CHAPTER 2
CIB FUNCTIONAL DESCRIPTION

2.1 CIB OVERVIEW

The Console Interface Board connects the console subsystem to the VAX-11/780 central processor. The CIB contains interfaces for the Q bus (LSI-11 bus), the ID bus, and the V bus; registers accessible to each bus; and all the hardware necessary to implement the various console functions. In addition, the CIB contains a 4K by 16 bit ROM which provides the bulk of the console LSI-11 software, and internal buses and multiplexers. The buses and multiplexers connect the various registers and bus interfaces. Figure 2-1 shows the CIB in block diagram form.

The LSI-11 software has access (read and/or write access) to the ROM matrix and all of the registers shown (except SYSID). The Q bus address decoder logic on the CIB identifies the register addressed. When the LSI-11 writes to a CIB register, the data is routed via the Q bus transceivers directly to the register specified. On a read, the contents of the register addressed are asserted on the Q bus by the same transceivers.

The VAX-11/780 software has access to the five following registers:

1. SYSID register
2. Receiver Control/Status register (RXIE, RX DNE)
3. TO ID register
4. Transmitter Control/Status register (TXIE, TX RDY)
5. FM ID register.

On a read transfer, the contents of the register addressed are gated via the ID MUX logic to the ID bus transceivers. When the VAX-11/780 software writes to the FM ID register, or one of the ID control/status registers, the data is gated from the ID bus transceivers directly to the register addressed.

2.2 CONSOLE/VAX-11 INTERACTION VIA THE CIB

All data transfer operations between the VAX-11/780 CPU and the console LSI-11 are routed via the TO and FM ID registers on the CIB, as shown in Figure 2-2, with two exceptions. First, the LSI-11 may look at various points in the VAX-11/780 CPU via the V bus. Second, it may look at the data on the ID bus via the ID DATA HI and LO registers when the VAX-11/780 CPU clock is stopped.

However, the interaction of the console subsystem and the VAX-11/780 CPU is directly related to the states of the two processors. The VAX-11/780 CPU may be running or halted, and the LSI-11 may be in the program I/ mode or the console I/O mode.

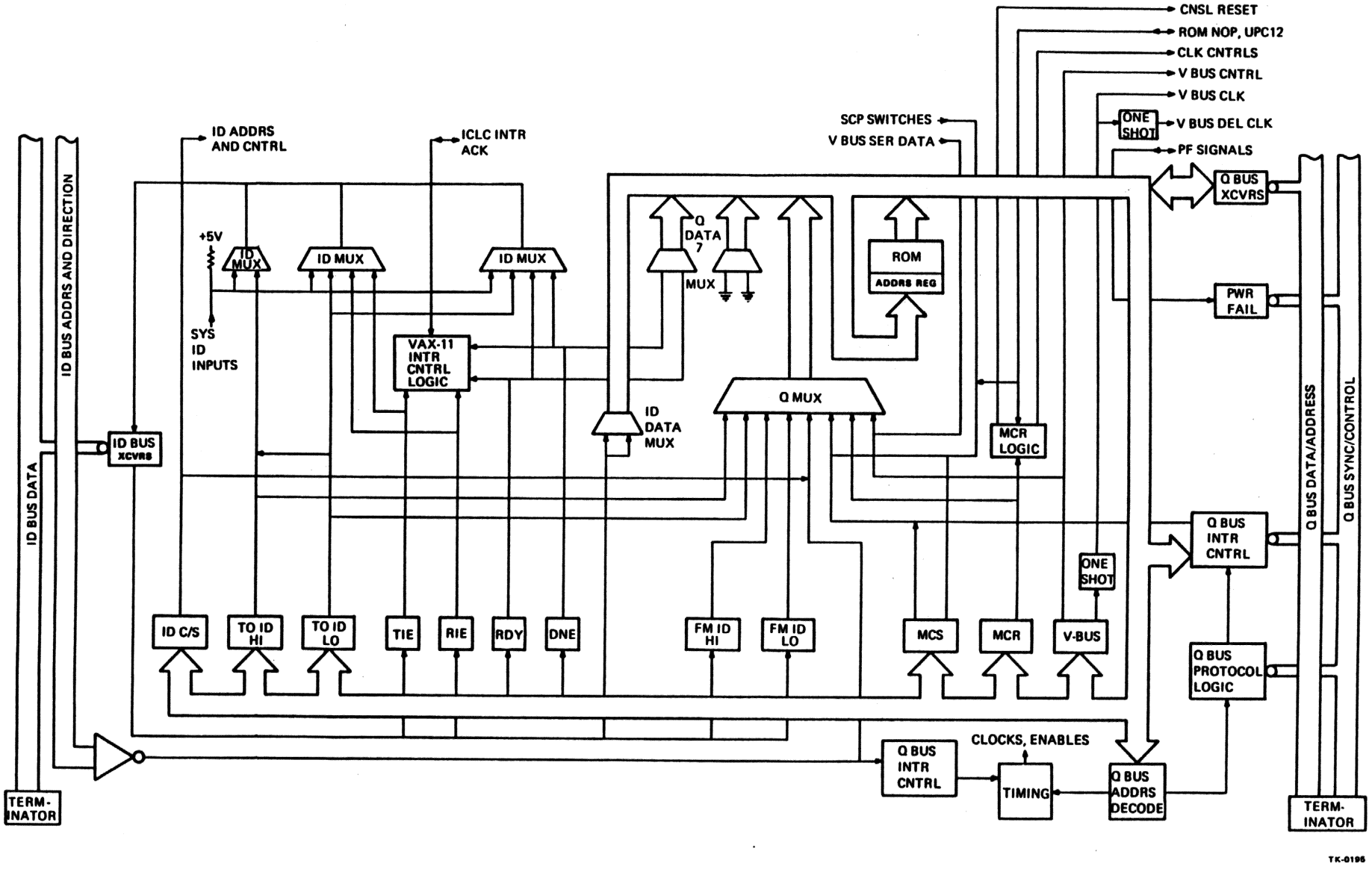


Figure 2-1 CIB, Block Diagram

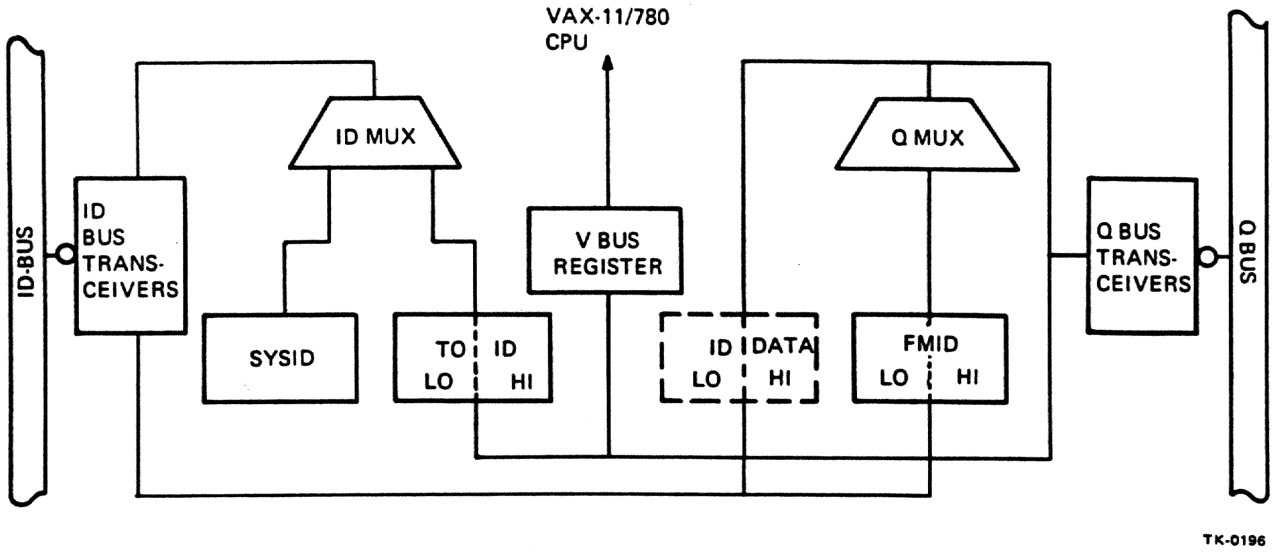


Figure 2-2 CIB Data Paths, Simplified Block Diagram

2.2.1 VAX-11/780 CPU Status: Halted and Running

When the VAX-11/780 CPU is halted, it enters the console command mode. The VAX-11/780 microprocessor jumps to the console wait loop and the attention (ATTN) indicator on the panel lights up. When the VAX-11/780 CPU is in this mode, the console subsystem may control the ID bus by asserting the ID MAINT signal. The LSI-11 software then provides the source of the ID bus address lines and the ID bus write control line. Note that the VAX-11/780 system clock may or may not be running when the VAX-11/780 CPU is in this mode.

When the VAX-11/780 CPU is not halted, it is running at the instruction set processor (ISP) level, executing macroinstructions, unless it is executing microdiagnostic or maintenance routines. The console software does not pass any commands to the ISP level software. When the LSI-11 is in the console I/O mode, it will not accept any output from ISP level software running in the VAX-11/780 CPU. Therefore, the VAX-11/780 software cannot communicate with the console floppy disk or the console terminal when the LSI-11 is in the console I/O mode. Figure 2-3 shows the various types of interaction of the two processors.

2.2.2 LSI-11 Operating Modes

The LSI-11 normally operates in either the program I/O mode or the console I/O mode. When the LSI-11 is in the program I/O mode, the VAX-11/780 CPU is always executing macroinstructions, and the LSI-11 passes the console terminal input, character by character, to the VAX-11/780 software. All data sent from the VAX-11/780 software to the terminal is passed by the LSI-11 software directly to the terminal.

When the LSI-11 operates in the console I/O mode, the LSI-11 software interprets all input from the console terminal, invoking specific console functions as appropriate.

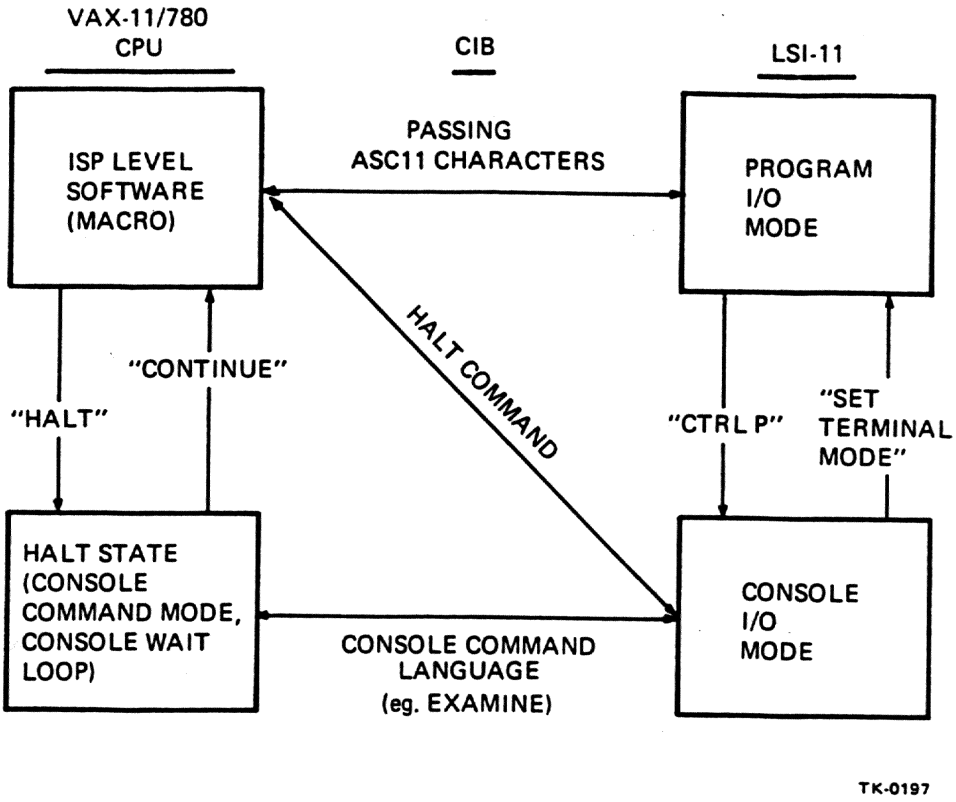


Figure 2-3 VAX-11/780 and LSI-11 Interaction and Operating Mode Combinations

When the VAX-11/780 CPU is executing macrocode, the console may be in either the program I/O mode or the console I/O mode. In the program I/O mode, the terminal operator can execute programs, and in other ways interact with the ISP level software. However, when the console is in the console I/O mode and the VAX-11/780 CPU is executing ISP level software, the range of functions which may be performed by the console is limited to those that require no direct response by the VAX-11/780 CPU (except HALT). These commands include the following:

SHOW
SET
HALT
WAIT DONE
HELP
EXAMINE/VBUS
CLEAR

Other console commands, such as EXAMINE, cannot be implemented until the VAX-11/780 CPU is halted (note that typing in HALT will do this).

When the VAX-11/780 CPU is executing macrocode, the operator at the console terminal can shift from the console I/O mode to the program I/O mode by typing "SET TERMINAL PROGRAM." He can shift from program I/O mode to console I/O mode by typing control-P (^P).

2.2.3 Use of CIB Registers

When VAX-11/780 ISP level software is running and the console is in the program I/O mode, communication between the console subsystem and the VAX-11/780 CPU is accomplished by means of interrupt transfers. Data is passed via the lower halves of the TO ID and FM ID registers; the status of the interrupt processes is controlled and monitored via the interrupt enable and disable and ready and done bits in the various control and status registers on the CIB.

In this respect the Console Interface Board functions like a conventional communications controller (e.g., a DL11), except that interrupts are generated on both sides of the interface and data is passed in parallel.

However, the console can make more extensive use of the facilities provided by the Console Interface Board when the VAX-11/780 CPU is in the console command mode and the LSI-11 is in the console I/O mode. Then the control signals developed on the ID control/status register (ID C/S), the machine control register (MCR), the miscellaneous control and status register (MCS) and the V bus register are all accessible to the LSI-11 as it interacts with the VAX-11/780 CPU.

2.3 ID BUS REGISTERS ON THE CIB

Of the five ID bus registers on the Console Interface Board, all but the SYS.ID register are dual ported, allowing VAX-11/780 CPU access via the ID bus and LSI-11 access via the Q bus. This section describes these registers as they appear on the VAX-11/780 CPU side of the interface. Figure 2-4 shows the addresses and bit configurations of the five registers.

2.3.1 System Identification Register (SYS.ID, ID03)

This register makes the system identification available in the processor register space. The 32 bits come out to pins for back panel switches. Pull-up resistors are provided on the CIB.

2.3.2 Receiver Control/Status Register (RXCS, ID04)

The VAX-11/780 software and LSI-11 software use this register to synchronize data transfers from the console subsystem to the VAX-11/780 CPU through the TO ID register. RXCS contains two bits.

RXCS<7> Receiver Done (RX DNE) - R/O

The LSI-11 sets this bit to indicate to the VAX-11/780 microcode that valid data is available in the TO ID register. This is a read only bit from the ID bus side; the CIB hardware clears the bit automatically when the VAX-11/780 microcode reads the TO ID register. System initialization also clears the bit.

RXCS<6> Receiver Interrupt Enable (RXIE) - R/W

When set, this read/write bit enables an interrupt at IPL 14_{16} and vector FC_{16} to the VAX-11/780 CPU each time the RX DNE bit makes a transition from 0 to 1. Each transition generates one interrupt. If RX DNE is already set and RXIE goes from 0 to 1, the interrupt will also occur. System initialization clears this bit.

2.3.3 TO ID Register (TO ID, RXDB, ID05) - R/O

This data buffer register serves two functions. First, it may be loaded by the LSI-11 with data from the console terminal, one ASCII character, to be read by the VAX-11/780 microcode. The data is valid only when the RX DNE bit is set, and when the microcode reads TO ID, the CIB hardware automatically clears RX DNE. Second, the LSI-11 may write to any ID bus address through the TO ID register by executing an ID maintenance cycle return function when the VAX-11/780 CPU is halted (see Paragraph 2.7 for further details). Note that the terms *TO* and *FROM* are used with respect to the VAX-11/780 CPU.

2.3.4 Transmit Control/Status Register (TXCS)

The VAX-11/780 CPU and the LSI-11 software use this register to synchronize data transfers from the VAX-11/780 microcode to the console subsystem through the FM ID register. TXCS contains the following two bits:

TXCS<7> Transmitter Ready (TX RDY) - R/O

The LSI-11 sets this read only bit to indicate to the VAX-11/780 microcode that it is ready to accept another character in the FM ID register. The CIB hardware clears this bit automatically when the microcode writes data into the FM ID register. System initialization clears the TX RDY bit.

TXCS<6> Transmit Interrupt Enable (TXIE) - R/W

When the VAX-11/780 microcode sets this read/write bit it enables an interrupt at IPL 14_{16} and vector $F8_{16}$ each time TX RDY goes from 0 to 1. Only one interrupt occurs for each 0 to 1 transition. If TX RDY is already set and the TXIE bit makes a 0 to 1 transition, the interrupt will also occur. TXIE is cleared by system initialization.

2.3.5 From ID Register (FM ID, TXDB, ID07) - W/O

Like the TO ID register, this data buffer register serves two functions. First, it may be loaded by the VAX-11/780 microcode with data to be passed to the console subsystem. It should be loaded in this way only when the TX RDY bit is set. The CIB hardware automatically clears TX RDY when the VAX-11/780 microcode writes to the FM ID register. Second, the LSI-11 may read any ID bus register through the FM ID register by executing an ID maintenance cycle when the VAX-11/780 CPU is halted.

2.3.6 Use of the TO ID and FM ID Registers

Under normal circumstances, when the VAX-11/780 CPU is running, executing ISP level software and the LSI-11 is in the program I/O mode, only the low order portions of the TO ID and FM ID registers are used, as shown in Figure 2-5.

The low order eight bits of the TO ID register (RXDB<7:0>) will contain data coded as an ASCII character to be passed from the LSI-11 to the VAX-11/780 CPU. Bits <11:8> specify the console unit at which the data originated. Logical unit 00 is reserved for the operator terminal.

The FM ID register is used in the same way when the VAX-11/780 CPU is running. Bits <7:0> contain the ASCII character to be passed to the LSI-11. Bits <11:8> specify one of the logical units in the console subsystem.

All references to the RXCS, TO ID, TXCS, and FM ID registers when the VAX-11/780 CPU is running result from microcode interpretation of MFPR and MTPR instructions. However, the VAX-11/780 microcode does not test the READY and DONE bits before referencing the TO ID and FM ID registers when executing MFPR and MTPR instructions. To do so would affect interrupt latency times. The macro level instructions should test the READY and DONE bits before the microcode references the data buffer registers.

When the VAX-11/780 CPU is halted, the microcode and the LSI-11 software may use all 32 bits of the TO ID and FM ID registers to transfer parameters and other information. The VAX-11/780 microcode has responsibility for testing the appropriate synchronizing bits before referencing the registers. Note that the READY or DONE bit must be set before the transfer can take place.

Note also that since the LSI-11 cannot read the TXIE and RXIE bits, it must disable interrupts to the VAX-11/780 CPU while the TO ID and FM ID registers are being used for examine functions and the like. The reader should also understand that the bits in the TXCS and the RXCS registers are totally divorced from the corresponding bits in the DLV-11E. When the LSI-11 is in the program I/O mode, it simply passes data to and from the CIB and to and from the DLV11-E.

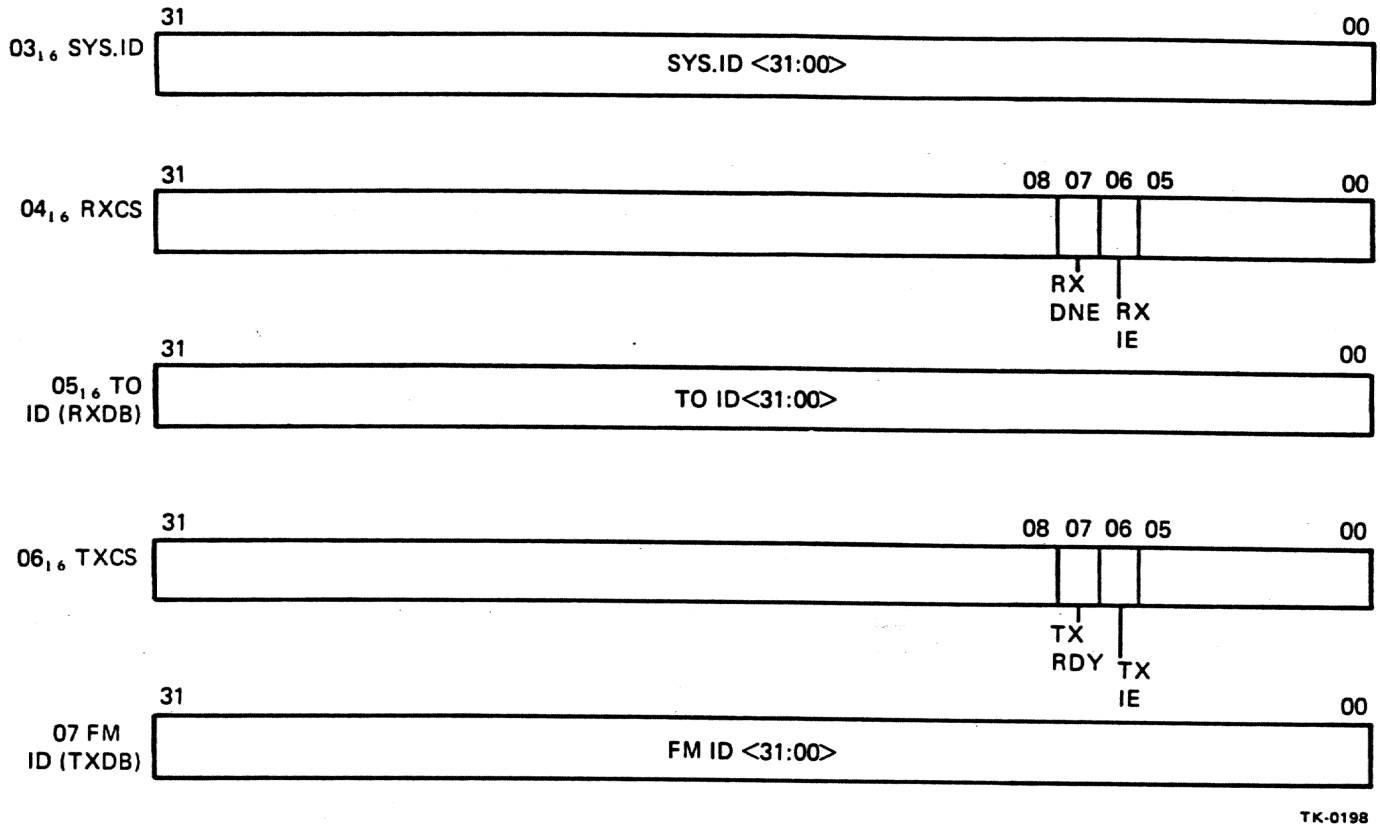


Figure 2-4 ID Bus Registers on the CIB

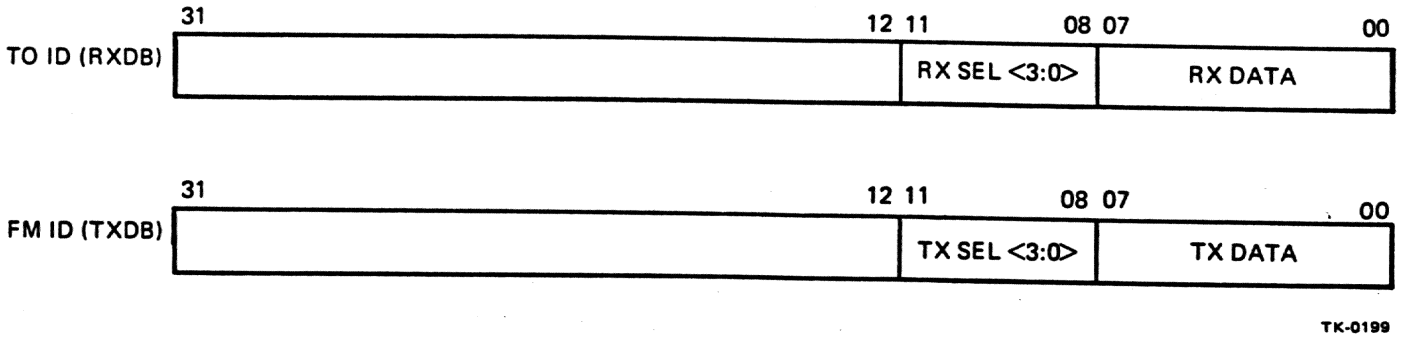


Figure 2-5 Use of the TO ID and FM ID Registers When the VAX-11/780 CPU is Running

2.4 Q BUS REGISTERS ON THE CIB

The Console Interface Board contains sixteen 16 bit registers which are addressable from the LSI-11. Six of these registers are dual ported, allowing VAX-11/780 CPU access as well as LSI-11 access. This section describes the registers as they appear on the LSI-11 side of the interface. Figures 2-6 and 2-7 show the addresses and bit configurations of the Q bus registers.

2.4.1 Read Only Memory

The Console Interface Board contains 4K words of ROM, starting at Q bus location 140000_8. This ROM contains the bulk of the LSI-11 console operating system, including the power up routines, drivers, look-up tables, and basic console command routines, such as EXAMINE, DEPOSIT, and HALT.

The first two words of the ROM are also addressable as registers at locations 163000_8 or 173000_8 and 163002_8 or 173002_8. These two words contain a JMP X instruction for the LSI-11, where X is the starting address of the power up routine in the ROM. These two words are located at ROM addresses 0 and 2, and they therefore appear also at Q bus addresses 140000_8 and 140002_8. The LSI-11 is jumpered to fetch at 163000 or 173000_8 upon power up, so that it can execute the jump to the power up routine.

2.4.2 ID DATA LO and ID DATA HI Registers ($163006/173006$, $163010/173010_8$)

If the VAX-11/780 system clock is stopped when the LSI-11 initiates an ID maintenance cycle in order to read an ID bus register, the LSI-11 reads the data through the ID DATA LO and ID DATA HI registers. These two Q bus registers allow the LSI-11 to look directly at the data currently asserted on the ID bus. They are read-only registers and the data is valid only when the VAX-11/780 CPU clock is stopped. ID DATA LO contains ID bus data bits <15:00>. ID DATA HI contains data bits <31:16>.

2.4.3 Receiver Done Register ($163014/173014_8$) - R/W

The receiver done register contains one bit in bit position 07, RX DONE. This bit is the backside of the RX DNE bit of the RXCS register on the ID bus (the same flip-flop is used for each) making this synchronization bit available to both the LSI-11 and the VAX-11/780 CPU. From the Q bus side, RX DONE is a read/write bit. The LSI-11 sets the bit to indicate to the VAX-11/780 microcode that valid data has been placed in the TO ID register. RX DONE is cleared automatically by a read reference to TO ID on the ID bus or by system initialization. The 1 to 0 transition of RX DONE will interrupt the LSI-11 if interrupts to the console are enabled.

2.4.4 Transmitter Ready Register ($163016/173016_8$) - R/W

The transmitter ready register also contains one bit in bit position 07, TX READY. This bit is the backside of the TX RDY bit of the TXCS register on the ID bus, and is thus available on both buses. From the Q side TX READY is a read/write bit. The LSI-11 sets this bit to indicate to the VAX-11/780 microcode that it is ready to accept another longword in the FM ID register. TX READY is cleared automatically when the VAX-11/780 CPU writes to FM ID on the ID bus, or when the system is initialized. The 1 to 0 transition of TX READY will interrupt the LSI-11 if enabled.

The following constraints apply to these ready and done bits:

- The VAX-11/780 interrupt enable bits, RXIE and TXIE, are not visible to the LSI-11.
- If the VAX-11/780 CPU clock is running, clocking RX DNE and TX RDY from Q bus data is done at CPT 60 - CPT 90. These bits will therefore be stable at CPT 0 when read on the ID bus. This feature enables the VAX-11/780 microcode to execute a tight loop, branching on the setting of RX DNE and TX RDY. The LSI-11 can set the bit from the Q bus while it is being referenced on the ID bus.
- RX DONE and TX READY are not set automatically by references on the Q bus to the TO ID and FM ID registers. These bits must be explicitly set by the LSI-11.

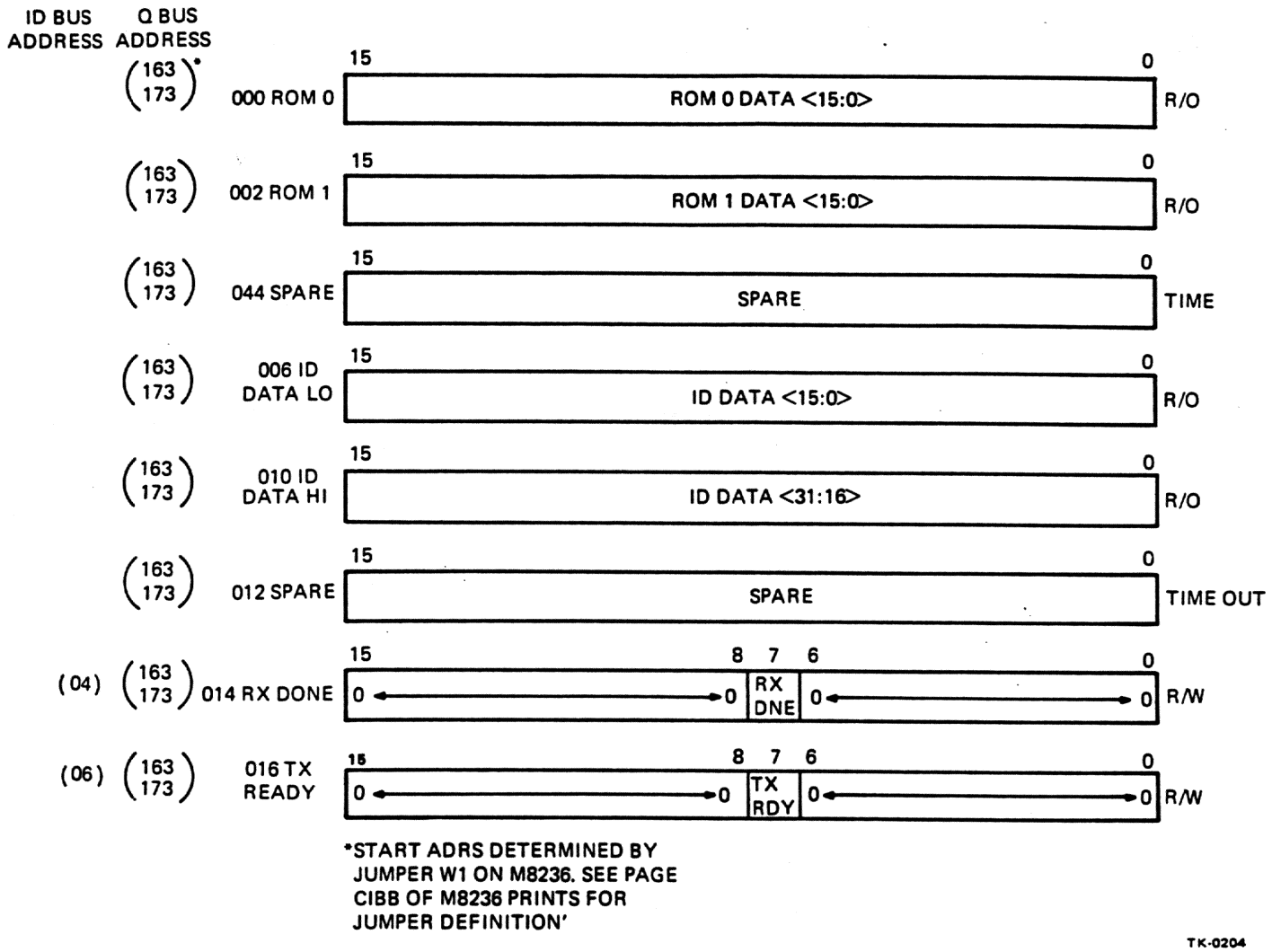


Figure 2-6 Lower Eight Q Bus Registers: Addresses and Bit Configurations (Address Bit 4 is False)

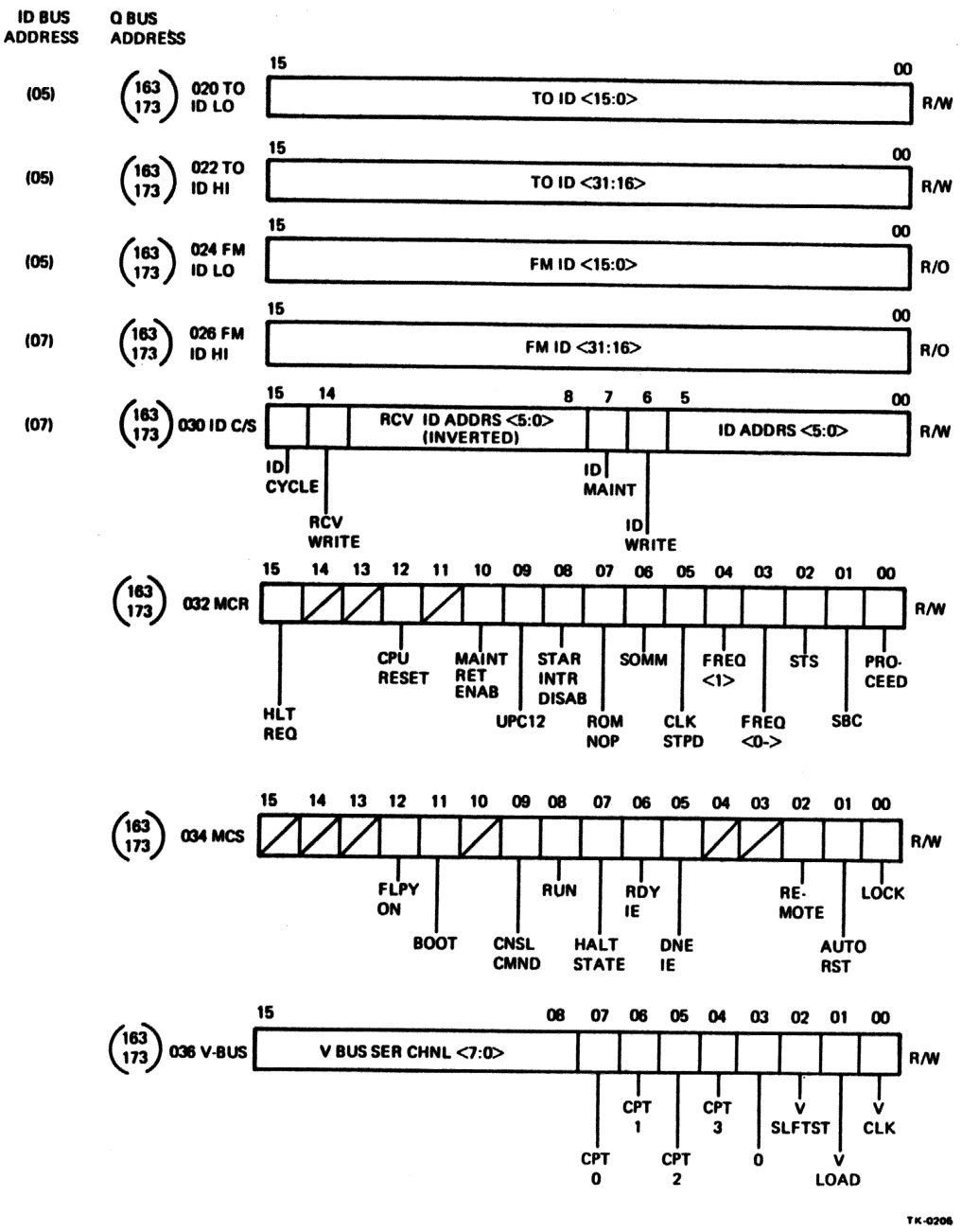


Figure 2-7 Upper Eight Q Bus Registers: Addresses and Bit Configurations (Address Bit 4 is True)

2.4.5 TO ID LO and TO ID HI Registers ($163020/173020_8$, $163022/173022_8$) - R/W

These two registers together contain 32 data bits which the LSI-11 can write via the Q bus when sending data to the VAX-11/780 CPU. When the microcode reads the TO ID register, it loads the data into the Q register in the VAX-11/780 CPU. The LSI-11 also places data in the TO ID LO and TO ID HI registers before performing an ID MAINT cycle to execute an ID bus write cycle. TO ID LO and TO ID HI are read/write registers from the Q bus side. They are readable for diagnostic purposes.

2.4.6 FM ID LO and FM ID HI Registers ($163024/173024_8$, $163026/173026_8$) - R/O

These two registers form the Q bus side of the FM ID register. They permit the LSI-11 to read data loaded into the FM ID register by the microcode as a result of a write reference to ID bus address 07_{16}. In addition, the LSI-11 reads data placed in the FM ID LO and FM ID HI registers when performing an ID maintenance cycle to execute an ID bus read cycle. FM ID LO and FM ID HI are read only registers, from the Q bus side.

2.4.7 ID Control Status Register, ID C/S ($163030/173030_8$)

The LSI-11 software uses this register to monitor the ID bus address and direction lines and to control console generated ID bus functions directly. In combination with the ID DATA registers, the TO ID registers, and the FM ID registers, the ID C/S register permits the LSI-11 to read or write any ID bus register while the clock is running or in the single step mode. The ID C/S register also permits the LSI-11 to read any ID bus register when the clock is stopped. Note that writing to ID bus registers requires stepping the clock, if the clock is not running.

ID C/S register bit descriptions follow.

ID C/S <15> ID CYCLE - R/W

When the LSI-11 writes ID CYCLE as a 1, it causes the Console Interface Board to assert the ID MAINT signal for one clock cycle (CPT 0 to CPT 0) if the clock is running. ID MAINT will be asserted at the next occurrence of CPT 0 if the clock is being single stepped. The ID CYCLE bit is cleared automatically at the end of the ID bus cycle. This means that the LSI-11 will read the bit as a zero unless the clock is in the single time state mode. See the description of the ID MAINT bit for further explanation.

ID C/S <14> ID RCV WRITE - R/O, and ID C/S <13:08> ID ADRS <5:0> - R/O

These seven bits allow the LSI-11 to read the state of the ID bus left address and direction lines. They are valid only when the clock is stopped. Note that because of receiver gate inversions, these bits must be read as the complements of the logical states of the corresponding bus wires.

ID C/S <07> ID MAINT - R/W

The CIB hardware will assert the ID MAINT bit automatically, after the LSI-11 sets the ID CYCLE bit, at the next occurrence of CPT 0. ID MAINT will remain set for one clock cycle. The following CPT 0 will clear it. The ID MAINT signal steers the multiplexer in the VAX-11/780 CPU which selects the source of the ID bus address and direction lines. During the time that the ID MAINT signal is asserted, the ID bus address and direction (WRITE) lines will be sourced from bits <06:00> of the ID control/status register.

If the VAX-11/780 CPU clock is running, ID MAINT is a read-only bit and writing to it has no effect. However, it will be read as a 0 by the LSI-11 since it is asserted only for a 200 ns cycle.

If the VAX-11/780 CPU clock is not running (the CLOCK STOPPED bit in the MCR is set), the LSI-11 may set or clear the ID MAINT bit by writing a 1 or a 0 to it. This feature enables the LSI-11 to statically read ID bus registers via the ID DATA LO and ID DATA HI registers. The address and WRITE fields may be written by the LSI-11 together with ID MAINT or ID CYCLE, in the same instruction.

LSI-11 initialization clears both the ID MAINT and the ID CYCLE bits.

ID C/S <06> ID WRITE - R/W, and ID C/S <05:00> XMT ADRS <05:00> - R/W
These seven bits form the ID bus address and direction lines during an ID cycle invoked by ID CYCLE or ID MAINT, enabling the LSI-11 to control the ID bus. ID C/S <06>, the WRITE bit, should be set to invoke a write cycle and cleared to invoke a read cycle. The LSI-11 software loads the XMT ADRS bits in true form (high true). These bits are cleared by LSI-11 initialization.

When the LSI-11 writes to an ID bus register, the data is sourced from the TO ID register. For static reads (clock stopped) the LSI-11 reads the data in the ID DATA LO and ID DATA HI registers. On dynamic reads the data is available to the LSI-11 in the FM ID LO and FM ID HI registers on the following cycle.

2.4.8 Machine Control Register, MCR ($163032/173032_8$)
The LSI-11 software uses the machine control register to control and monitor several functions of the VAX-11/780 CPU. The following bit descriptions outline these functions.

MCR <15> Halt Request - R/W
The LSI-11 sets this bit to force the VAX-11/780 microcode to jump into the console wait loop. The VAX-11/780 CPU recognizes the assertion of HLT REQ and sets the halt pending flip-flop in the interrupt control logic of the VAX-11/780 CPU. When the CPU passes through the instruction register decode (IRD) state and the halt pending flip-flop is set, the microcode will set the console command mode (CNSL CMND MODE) bit and enter the console wait loop. This mode is indicated on the CIB by the assertion of HALT STATE (MCS bit <07>) and on the control panel by the lighting of the ATTN indicator. Later, when the microcode invokes the CONTINUE function, in response to a command from the LSI-11, the halt pending flip-flop and the CNSL CMND MODE bit are both reset, and the VAX-11/780 CPU enters the IRD state. If the Halt Request signal is still enabled on the CIB, the halt pending flip-flop will set again, but not until the VAX-11/780 CPU leaves the IRD state.

This means that the invocation of a CONTINUE function with HLT REQ asserted results in a single instruction execution. In order to cause the VAX-11/780 CPU to resume normal instruction execution, the LSI-11 software must first clear HLT REQ and then issue a CONTINUE command. The LSI-11 system initialization clears HLT REQ.

Note that the VAX-11/780 microcode can branch on the state of the CNSL CMND MODE bit. This enables the microcode to determine, when in various error routines, whether the routine was entered as a result of some console requested function or normal machine execution.

MCR <14:13> Reserved

MCR <12> CPU RESET - R/W
When the LSI-11 sets CPU RESET, it forces the assertion of the internal initialization signal (DC LO equivalent) within the VAX-11/780 CPU. Upon the negation of CPU RESET, the microcode enters the power-up initialization routine. LSI-11 system initialization clears this bit.

MCR <11> Reserved

MCR <10> Maintenance Return Enable - R/W
When the LSI-11 writes a 1 to this bit, it causes the VAX-11/780 CPU to perform a maintenance return function. The VAX-11/780 CPU pops the top element of the microstack and disables the J-field inputs. The result is a forced jump to the location addressed by the word at the top of the microstack. MAINT RET ENABLE remains asserted at the microsequencer approximately 100 ns, from CPT 150 to CPT 50. When the VAX-11/780 CPU is in the single time state mode, MAINT RET ENABLE will remain asserted from the time it was written as a one until the next occurrence of CPT 50. Note that the LSI-11 should not attempt a maintenance return function unless the VAX-11/780 CPU is in the console wait loop.

MCR <09> Trap to Writable Control Store - R/W
When the LSI-11 sets TRAP TO WCS, it forces bit 12 of the UPC to a one whenever a microtrap occurs, in turn forcing the trap into writable control store. This bit should be set or cleared only when the VAX-11/780 CPU is in the halt state or the clock is stopped. LSI-11 system initialization clears the TRAP TO WCS bit.

MCR <08> Star Interrupt Disable - R/W
When the LSI-11 sets this bit, it disables the TX RDY and RX DNE interrupts to the VAX-11/780 CPU, regardless of the state of the interrupt enable bits, TXIE and RXIE. It furthermore inhibits any change in the state of the interrupt pending flip-flops in the terminal interrupt control logic on the CIB.

STAR INTR DISAB allows the LSI-11 use of the TO ID and FM ID registers and the ready and done bits for console functions (when the LSI-11 is in the console I/O mode) without causing extraneous interrupts to the VAX-11/780 CPU.

MCR <07> ROM NO-OP - R/W
When the LSI-11 sets this bit, it generates the CLR UWORD and ABORT CYCLE signals in the microsequencer, producing the same effect as STALL. ROM NOP thus forces NOPs on the various subsystem control fields so that random patterns from WCS will not produce undesired side effects during testing. If the bit is set while the clock is stopped, the clock must be stepped to CPT 0 before the ROM NOP signal takes effect. This bit is cleared by LSI-11 system initialization.

MCR <06> Stop on Microbreak Match - R/W
If the VAX-11/780 CPU detects a match between the microbreak register and the UPC and the LSI-11 has set the SOMM bit, the clock will stop in CPT 0 of the cycle in which the match occurs. This in turn causes the assertion of the CLK STPD signal. The SOMM bit is cleared by LSI-11 system initialization.

MCR <05> Clock Stopped - R/O
This read only bit is formed by a signal which originates in the clock control logic. It is set when the clock is not running. CLK STPD is also used in several circuits on the Console Interface Board for determination of whether or not synchronization to the VAX-11/780 CPU clock is necessary. LSI-11 system initialization clears this bit.

MCR <04:03> Frequency Select <1:0> - R/W
The LSI-11 software sets these two bits to control the VAX-11/780 CPU clock frequency, as shown in Table 2-1.

Table 2-1 Clock Frequency Control

FR 1	FR 0	Frequency
0	0	10.0 Mhz (normal)
0	1	10.525 Mhz (5% short)
1	0	8.925 Mhz (12% long)
1	1	External Source

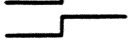
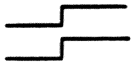
FR 1 and FR 0 should not be changed when the clock is running. LSI-11 system initialization clears both bits.

MCR <02> Single Time State - R/W
If the LSI-11 software sets the STS bit when the VAX-11/780 CPU clock is running the clock will stop in any of the four time states. As long as the STS bit is set, and regardless of the state of the Single Bus Cycle bit, writing a 1 to the PROCEED bit will step the clock one time state (e.g., from CPT 100 to CPT 150). LSI-11 system initialization clears the STS bit.

MCR <1> Single Bus Cycle - R/W
If the LSI-11 asserts this bit (and the STS bit is 0) while the clock is running, the clock will stop in CPT 0. As long as SBC is set and STS is cleared, writing a 1 to PROCEED will step the clock to the next CPT 0 (i.e., one ROM/SBI cycle). LSI-11 system initialization clears the SBC bit.

MCR <0> Proceed - W/O
When the LSI-11 software writes a 1 to the PROCEED bit, it will affect the clock in any of three ways, depending on the states of the STS and SBC bits, as shown in Table 2-2.

Table 2-2 Uses of PROCEED

PROCEED	STS	SBC	Clock
	0	0	start clock running immediately
	0	1	step clock one cycle
	1	0	step clock one time state
	1	1	step clock one time state

Writing a 1 to the PROCEED bit when the clock is running has no effect. LSI-11 system initialization clears the bit. Note that CPT 0 = SBI T1. In other words the SBI leads the CPU by one time state.

Constraints on Clock Control
The proper method for stopping the clock is to write a 1 to the SBC bit. In this way the stopped clock state is known to be CPT0.

Clearing the STS and SBC bits will not start the VAX-11/780 CPU clock. The LSI-11 software must write a 1 to the PROCEED bit after STS and SBC have been cleared in order to start the clock.

2.4.9 Miscellaneous Control and Status Register, MCS ($163034/173034_8$)
The miscellaneous control and status register enables the LSI-11 software to control and monitor some of the console subsystem functions and the interaction between the LSI-11 and the VAX-11/780 CPU. Descriptions of the bit functions follow.

MCS <15:13> Reserved

MCS <12> Floppy On - R/W
The FLPY ON bit controls power to the console floppy disk drive through a relay. When the LSI-11 sets the bit, it turns on power to the floppy. Clearing the bit turns the floppy off. The LSI-11 system initialization clears FLPY ON.

MCS <11> Boot - R/W
This read/write-one to clear bit is set by a 0 to 1 transition of the signal from the BOOT switch on the control panel. The bit will be cleared when the LSI-11 writes a 1 to the bit location and when the LSI-11 system is initialized.

MCS <10> Reserved

MCS <09> Console Command Mode - R/O
This read-only bit permits the LSI-11 to determine the state of the VAX-11/780 CPU. The CNSL CMND MODE bit is set by the VAX-11/780 microcode together with the assertion of HALT STATE. The bit remains set until cleared by the microcode while executing a CONTINUE function. When set this bit lights the ATTN indicator on the control panel.

MCS <08> Run - R/O
This read-only bit is the "1" side of a retriggerable one-shot. The one-shot is clocked by the VAX-11/780 interrupt strobe signal, and it will remain set as long as the VAX-11/780 CPU is strobing interrupts at least every 0.423 ms. When the VAX-11/780 CPU enters the console command mode, the RUN one-shot times out. Also, while the VAX-11/780 CPU is running and executing macrocode, the negation of RUN generally indicates some type of problem. For example, the microcode might be hung in a loop or there might be a hardware failure. The RUN one-shot is also used to light the RUN indicator on the control panel.

MCS <07> Halt State - R/O
The HALT STATE bit is the raw decoded output of a ROM field which is asserted if and only if the microcode is in the console wait loop. This bit is required because the CNSL CMND MODE bit, which is set upon entry into the console wait loop, will remain set even if the microcode leaves the loop. The LSI-11 software therefore is able to use the HALT STATE bit to determine whether or not the microcode has returned to the console wait loop following a console microcode function (other than CONTINUE). The LSI-11 must also test the HALT STATE bit to be sure that the microcode is in the console wait loop before performing a console ID maintenance cycle.

MCS <06> Transmit Ready Interrupt Enable
The LSI-11 sets the TX IE bit in order to enable interrupts from the VAX-11/780 CPU to the LSI-11 upon a 1 to 0 transition of the TX RDY bit. If the TX RDY bit is already 0, and the TX IE bit goes from a 0 to a 1, the interrupt will occur. LSI-11 initialization clears the TX IE bit.

MCS <05> Receiver Done Interrupt Enable
The LSI-11 sets the RX IE bit in order to enable interrupts from the VAX-11/780 CPU to the LSI-11 upon a 1 to 0 transition of the RX DNE bit. If the RX DNE bit is already 0, and the RX IE bit goes from a 0 to a 1, the interrupt will occur. LSI-11 initialization clears the RX DNE bit.

MCS <04:03> Reserved

MCS <02:00> Panel Switch Sense - R/O
These three bits sense the positions of control panel switches, as shown in Table 2-3.

Table 2-3 Panel Switch Sense Bit Functions

Bit	Related Panel Switch
MCS <02>	Auto Restart Switch (AUTO RST)
MCS <01>	Remote Mode (REMOTE)
MCS <00>	Lock (Lock = disable)

These bits are set when the corresponding switches are on. They are not affected by system initialization.

2.4.10 V Bus Register ($163036/173036_8$)

The V Bus register allows the LSI-11 software to perform the data load, serial shift, and self test functions on the V bus by writing to the V Bus register. The serial data retrieved from polling stable internal state VAX-11/780 CPU test points is then available when the LSI-11 software reads the register. A description of the V Bus register bits follows.

V-BUS <15:08> Serial Data Bits - R/O

These eight bit positions are loaded with serial data from shift registers on the eight V bus data channels which originate within each of the VAX-11/780 CPU subsystems.

V-BUS <07>: CPT 0 - R/O

V-BUS <06>: CPT 1 - R/O

V-BUS <05>: CPT 2 - R/O

V-BUS <04>: CPT 3 - R/O

These four bits enable the operator to determine the state of the VAX-11/780 CPU clock when the clock is stopped.

V-BUS <03> Reserved

V-BUS <02> SDMS Self Test - R/W

This signal feeds the data input to the most remote bit of the shift register connected to each of the V bus channels. The LSI-11 software uses the self test bit to assure correct operation of the V bus system by shifting a known bit pattern through each channel (as shown in Figure 1-4). The bit can be set or reset only by writing a 1 or a 0 to the bit location.

V-BUS <01> SDMS Load - R/W

The LSI-11 software sets this bit in order to parallel load machine state information into each of the shift registers on the V bus channels. If the LSI-11 asserts bits 1 and 0 with the same instruction, a load function is guaranteed. Bit 1 must be reset by the LSI-11 software for a shift to take place.

V-BUS <00> SDMS Clock - W/O

When the LSI-11 software writes a 1 to the SDMS CLOCK bit, it causes a 2 μs pulse to be applied to the clock inputs to all V bus channel shift registers. If the LSI-11 software asserts the SDMS CLOCK bit, and the SDMS LOAD bit is reset, a shift will take place on each V bus channel shift register, causing the next bit of data from each of the shift registers to be available in the upper byte of the V Bus register.

2.5 DIALOGUE: LSI-11 PROGRAM I/O - VAX-11/780 MACROCODE

When the LSI-11 is in the program I/O mode and the VAX-11/780 CPU is executing macrocode, the two processors use the TO ID and FM ID registers as the RXDB and TXDB, respectively. Interrupt service routines on each side handle the transfers.

2.5.1 LSI-11 Sends a Character to the VAX-11/780

When the console terminal operator types a character, the terminal interface interrupts the LSI-11. An LSI-11 interrupt service routine then loads the ASCII code for the character into the lower eight bits of the TO ID LO register on the CIB and then sets the RX DNE bit to interrupt the VAX-11/780 CPU. Figure 2-8 is a flowchart showing the dialogue between the VAX-11/780 macrocode, the VAX-11/780 microcode, and the LSI-11 software.

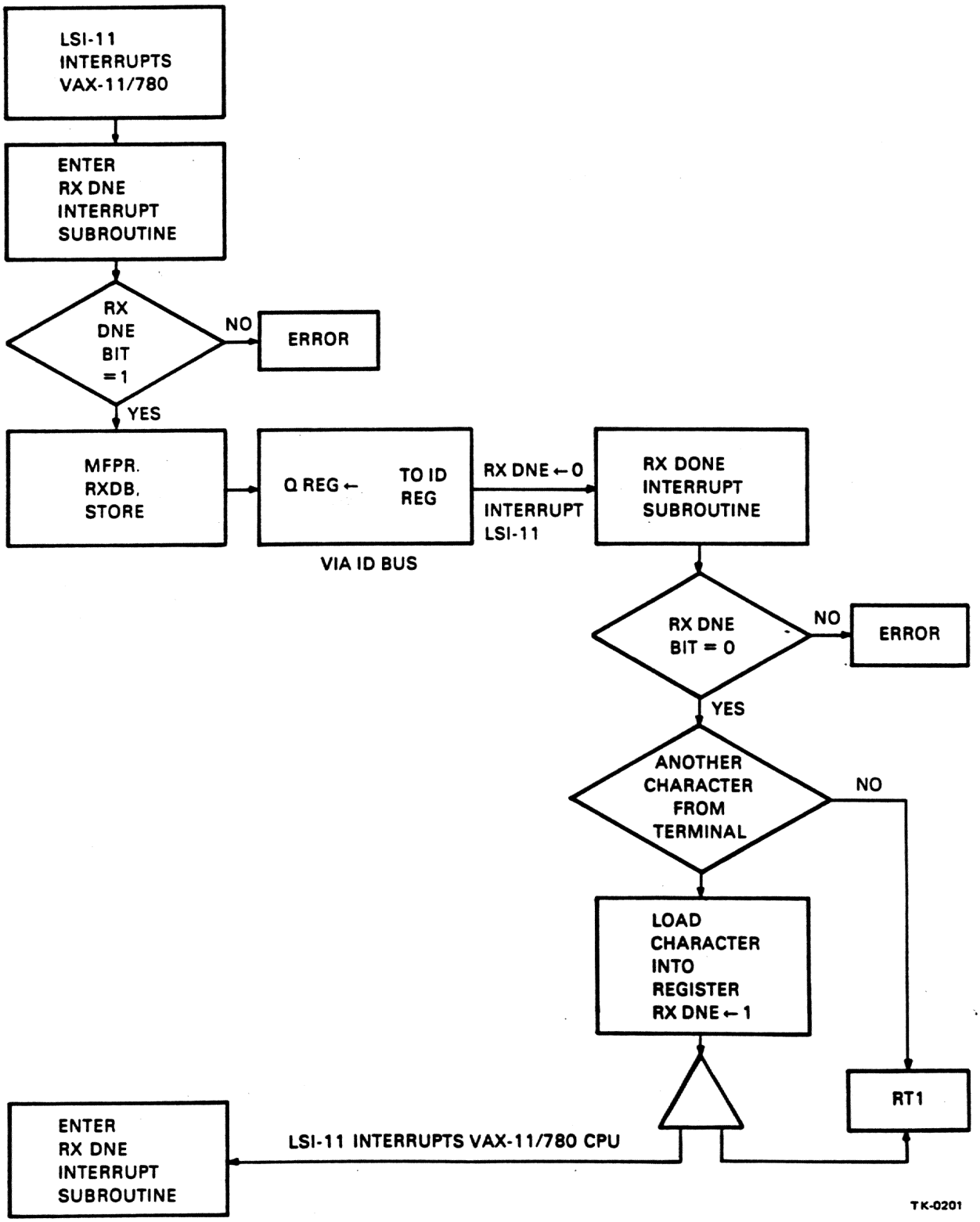


Figure 2-8 LSI-11 Sends a Character to the VAX-11/780 CPU, Flowchart

The VAX-11/780 CPU responds to the interrupt by invoking an interrupt subroutine. This routine tests the RX DNE bit on the CIB. If the bit is set, the subroutine executes an MFPR instruction to read the TO ID register (RXDB) and stores the data in the Q register in the VAX-11/780 CPU. This action causes the RX DNE bit on the CIB to reset, thus causing the CIB to send an interrupt signal to the LSI-11. This interrupt in turn invokes an LSI-11 interrupt subroutine which tests the RX DNE bit on the CIB. If the RX DNE bit is reset, as it should be, the subroutine determines whether or not the terminal is waiting to send another character to the VAX-11/780 CPU. If the terminal does have another character to send, the subroutine loads the ASCII equivalent into the TO ID register on the CIB and then sets the RX DNE bit, causing another interrupt to the VAX-11/780 CPU. The LSI-11 software then returns from its interrupt subroutine.

2.5.2 VAX-11/780 Sends a Character to the LSI-11

When the VAX-11/780 software is transmitting a series of characters to the LSI-11, the LSI-11 will set the TX RDY bit on the CIB each time it reads a character. The setting of TX RDY then generates an interrupt signal to the VAX-11/780 CPU, invoking a transmit interrupt subroutine, as shown in Figure 2-9.

The transmit interrupt subroutine executes an MTPR instruction to load the character into the TXDB (FM ID register on the CIB) and return from the interrupt to the main program. The VAX-11/780 microcode implements the MTPR instruction, loading the FM ID register from the Q register. The loading of FM ID automatically resets the TX RDY bit, generating an interrupt to the LSI-11. The LSI-11 receive interrupt subroutine checks the TX RDY bit and then reads the data in the FM ID register and sets the TX RDY bit.

The LSI-11 then returns from the receive interrupt subroutine. The setting of the TX RDY bit generates another interrupt signal to the VAX-11/780 CPU, notifying the VAX-11/780 software that the FM ID register is empty and ready for another character.

2.6 DIALOGUE: LSI-11 CONSOLE I/O MODE - VAX-11/780 MICROCODE

When the VAX-11/780 CPU is halted (in the console wait loop) and the LSI-11 is in the console I/O mode, the LSI-11 may direct VAX-11/780 microcode routines to perform various functions which implement console command language instructions, such as an examine virtual address instruction. The TO ID and FM ID registers on the Console Interface Board are used to pass the parameters needed by or supplied by these routines, and the transfers are interlocked through use of the ready and done bits. However, when the LSI-11 sets the TX RDY bit or the RX DNE bit, no corresponding interrupt to the VAX-11/780 CPU is generated. Instead, the VAX-11/780 microcode assumes the responsibility for determining that the TX RDY bit or the RX DNE bit is set before writing or reading data in the FM ID register or the TO ID register.

The LSI-11 software invokes the VAX-11/780 microroutines needed to execute a console command language instruction by writing the starting address of the microroutine to the microstack, thus pushing the address on the microstack. The LSI-11 software then writes a 1 to the MAINT RET ENABLE bit in the MCR register on the CIB, popping the microstack and forcing a jump to the address specified. All VAX-11/780 microroutines invoked from the console except CONTINUE must return to the console wait loop.

Three types of events will cause the VAX-11/780 CPU to enter the Halt State (console command mode).

1. The VAX-11/780 CPU may execute a HALT instruction which is coded within a Macro program listing.

2. The VAX-11/780 CPU may halt as the result of an error condition.

3. The console terminal operator may type a HALT command on the terminal while the LSI-11 is in the console I/O mode, causing the VAX-11/780 CPU to halt.

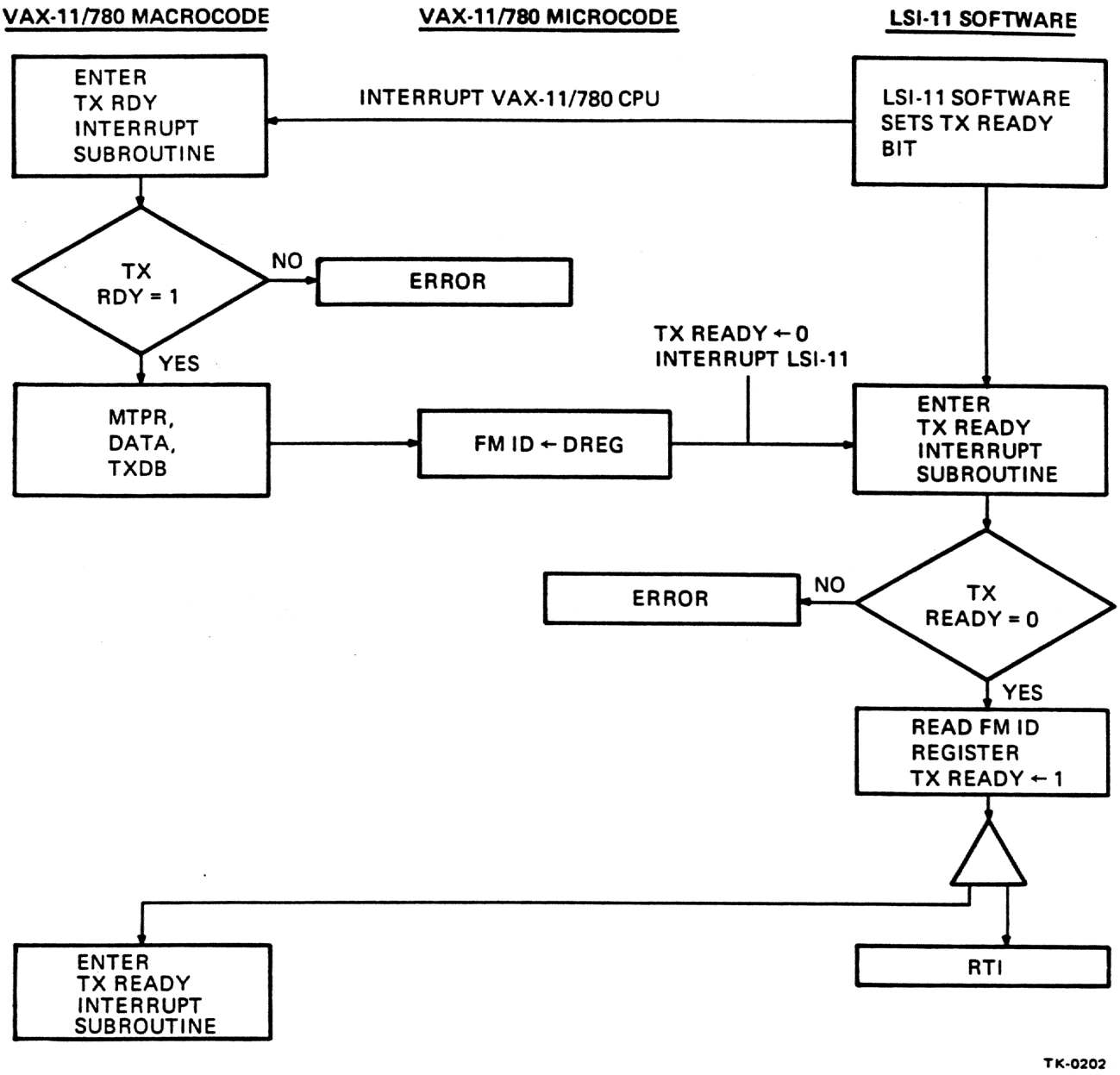


Figure 2-9 VAX-11/780 Sends a Character to the LSI-11, Flowchart

In any case, the microcode will load the D.SV register on the ID bus with a code which identifies the reason for the halt. If the LSI-11 is in the program I/O mode, and the VAX-11/780 CPU enters the halt state because of a programmed halt or an error condition, the LSI-11 software will sense the halt by reading the HALT STATE bit (MCR <10>) when it goes through a null loop, and thus branch out of the program I/O mode in order to display the reason for the halt on the terminal and enter the console I/O mode.

However, the LSI-11 software must determine whether or not the TO ID and FM ID registers contain valid data, and if so, save that data, before using those registers to read the contents of the D.SV register. The flowchart given in Figure 2-10 shows the steps taken by the LSI-11 software in response to a VAX-11/780 CPU halt.

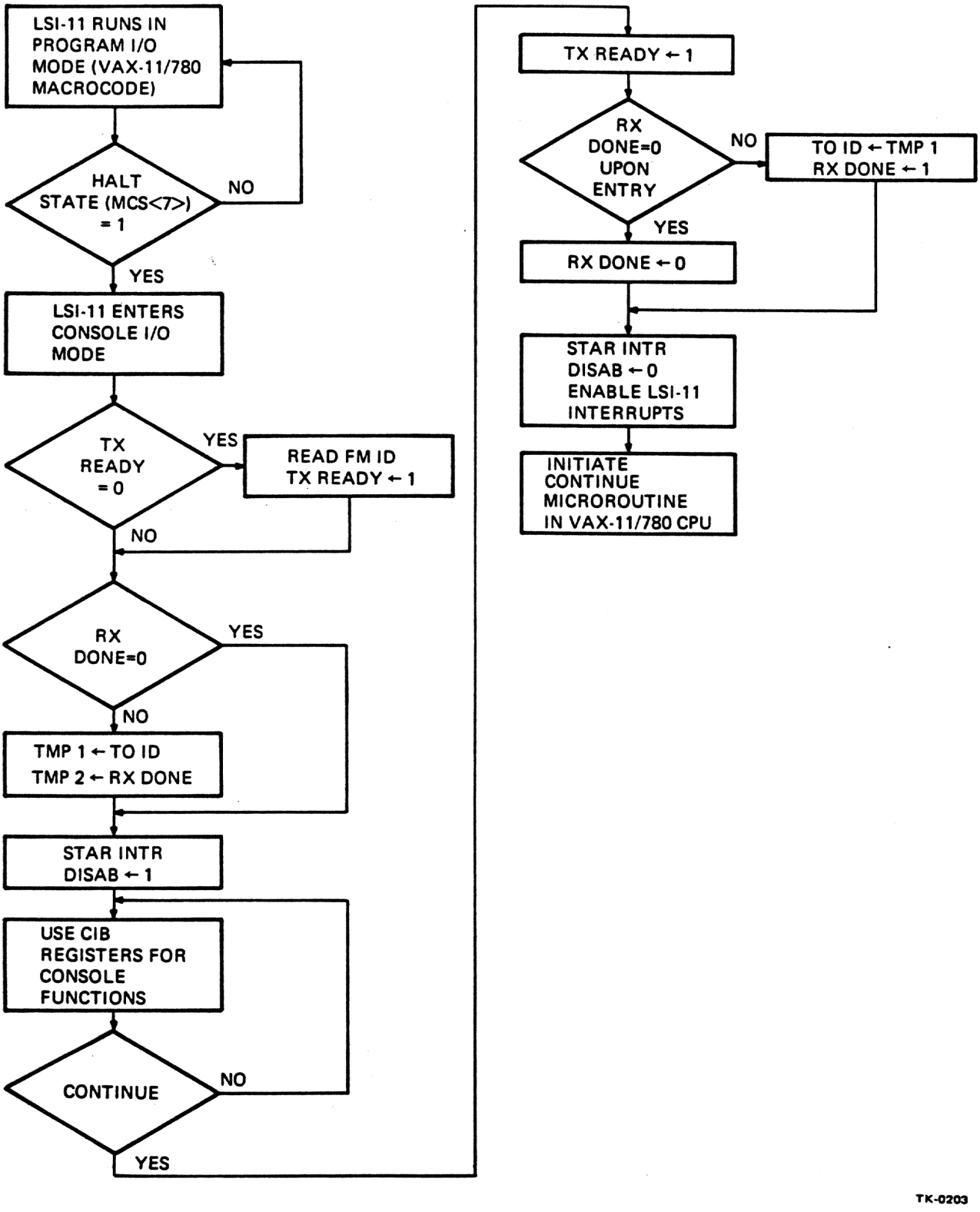


Figure 2-10 LSI-11 Response to a VAX-11/780 CPU Halt

If the TX RDY bit is 0 at the time of the halt, the LSI-11 software reads the FM ID register and either saves or acts on the data and then sets the TX RDY bit.

Then, if the RX DNE bit is 2, indicating that the TO ID register contains data from the LSI-11 which the VAX-11/780 microcode has not yet read, the LSI-11 must save the data and the state of the RX DNE bit.

The LSI-11 then sets the STAR INTR DISABLE bit (MCR <08>), freezing the state of the VAX-11/780 interrupt control logic on the CIB. The TO ID and FM ID registers and the READY and DONE bits are available to the LSI-11 software for other functions. At this point, the LSI-11 software can read the D.SV register and the operator can execute the various console command language instructions.

Some time later, when the console terminal operator issues a CONTINUE command, the LSI-11 software takes the following steps.

1. Sets TX RDY.

2. Reloads data saved from the TO ID register and sets the RX DNE bit, if RX DNE was set at the time of the halt.

3. Reenables interrupts to the VAX-11/780 CPU, enables interrupts to the LSI-11, and executes CONTINUE by pushing the starting address of the continue routine onto the microstack and then setting the MAINT RET ENABLE bit.

This sequence is necessary for two reasons. It prevents false interrupts to the VAX-11/780 CPU, and it ensures that interrupts do occur when the single instruction step mode is used.

2.7 DIRECT ID BUS REFERENCE

When the LSI-11 is in the console I/O mode and the VAX-11/780 CPU is in the console wait loop, the LSI-11 software references certain ID bus registers in the execution of some of the console command language instructions.

2.7.1 ID Bus Reference with the Clock Running

When the VAX-11/780 system clock is running, the LSI-11 software can write to any ID bus location by loading the data to be written into the TO ID LO and TO ID HI registers and then writing the register address and setting the WRITE and ID CYCLE bits in the ID C/S register. Note that the ID CYCLE bit may be set on a following instruction. In either case the ID maintenance function causes the data in the TO ID register to be written into the register addressed.

The LSI-11 software reads ID bus registers similarly, except that the WRITE bit of the ID C/S register must be 0. At the completion of the ID bus cycle, the data from the register addressed is available in the FM ID LO and FM ID HI registers.

2.7.2 ID Bus Reference with the Clock Stopped

The LSI-11 can read any ID bus register with the clock stopped by placing the desired address in the ID C/S register, clearing the WRITE bit, and setting the ID MAINT bit. However, since all ID bus register strobe signals are disabled in CPT 0 it is necessary to step the clock to the time state in which the desired strobe will enable the data onto the ID bus. As long as ID MAINT is set, the clock is in the correct time state, and the WRITE bit = 0, the addressed register may be read through the ID DATA LO and ID DATA HI registers. The address may be changed even while ID MAINT is set, enabling data from a newly addressed register onto the ID bus as long as the clock is in a time state which enables the appropriate strobe. The ID MAINT bit should be cleared before the clock is started again.

The LSI-11 software cannot write to ID bus registers when the clock is stopped without stepping through time states. But the LSI-11 software can accomplish write functions with the clock stopped by first ensuring that the clock is in CPT 0, then loading the data to be written into the TO ID LO and TO ID HI registers, and then setting the WRITE and ID MAINT bits and loading the desired address into the ID C/S register. Invoking a single bus cycle via PROCEED will then write the data into the registeraddressed and clear the ID MAINT bit. The LSI-11 software can accomplish read functions in a similar manner.

2.7.3 ID Bus Reference without use of Console Command Language Instructions

The console terminal operator may circumvent the LSI-11 console I/O mode software by moving the HALT/ENABLE switch on the LSI-11 control panel within the cabinet to the HALT position. This causes the LSI-11 to jump to the ODT (Octal Debugging Technique) routine, permitting read and write access to any Q bus register, and thus, indirectly, to the ID bus. The LSI-11 will also jump to the ODT routine if the operator presses the BREAK key on the console terminal. See Section 2, Chapter 2 of the *Microcomputer Handbook* for details of ODT.

CHAPTER 3
CIB DETAILED LOGIC DESCRIPTION

3.1 CONSOLE INTERFACE BOARD LOGIC

The Console Interface Board logic falls into four categories: ID bus interface, Q bus interface, ROM logic, and register logic, each of which is described in one of the four following subsections. A fifth subsection, consisting primarily of flowcharts, traces typical transfer operations at the detailed level.

3.2 ID BUS INTERFACE LOGIC

When the VAX-11/780 CPU asserts an address on the ID bus, the ID bus address decoder logic on the CIB becomes active only if the address falls within the range of 03_{16} to 07_{16}, as shown in Figure 3-1. Notice that address bits 3, 4, and 5 must be false.

Two full 32 bit registers can be read from the ID bus, SYS.ID and TO ID. When either of these registers is addressed on a read, or the CIB is writing to an ID bus register on an ID MAINT cycle, CIBM ID ENDAT ALL L and CIBM ENDAT (7,6) L go true, enabling the transmit function of the ID bus transceivers between CPT 100 and CPT 0. Normally, the state of CIBM SYS.ID DECODE L generates CIBM ID MUX SEL L, governing the output of the 2:1 multiplexer shown in Figure 3-2, and determining whether the contents of SYS.ID or TO ID are asserted on the bus. However, when the CIB performs an ID MAINT cycle write transfer, the CIBM SYS.ID DECODE L signal is overridden, and the TO ID register is automatically selected by the multiplexer. A 4:1 2-bit multiplexer selects one of four sources for data bits 7 and 6 on all read transfers initiated by the VAX-11/780 CPU.

When the VAX-11/780 CPU writes to bit 6 of the RXCS or TXCS registers, the bit will set or clear with the assertion of CIBM CLK RXIE L or CIBM CLK TXIE L at CPT 100, as shown in Figures 3-1 and 3-23. See Paragraph 3.5.2 for an explanation of the logic involved in a write transfer to the FM ID register.

3.3 Q BUS ADDRESS DECODER AND TIMING LOGIC

The Q bus address decoder logic consists of three closely related circuits: the interface chips, the QMUX circuit, and a one of twelve decoder circuit.

3.3.1 Interface Address Decoder Logic

When the LSI-11 processor initiates a read or write transfer to one of the CIB registers, it asserts the address on BDAL <15:00> L and then asserts BBS7 L and BSYNC L. If the asserted address falls within the range of the jumpered address inputs to the DC005 interface chips, these chips will assert MATCH H, as shown in Figure 3-3. Note that the jumper, W1, determines whether MATCH H is asserted in response to addresses in the range of $1730XX_8$ or $1630XX_8$, as follows.

W1 installed: Address $1730XX_8$
W1 removed: Address $1630XX_8$

Note: W1 is normally installed.

The DC005 chips also gate the address to the tristate BUS QDATA <15:00> H lines. Bits <12:01> are latched as CIBJ LADRS <12:01> H with the receipt of BSYNC L. The ANDing of the MATCH H and BSYNC L signals within the DC004 interface chip enables CIBJ TRPLY L after a slight delay.

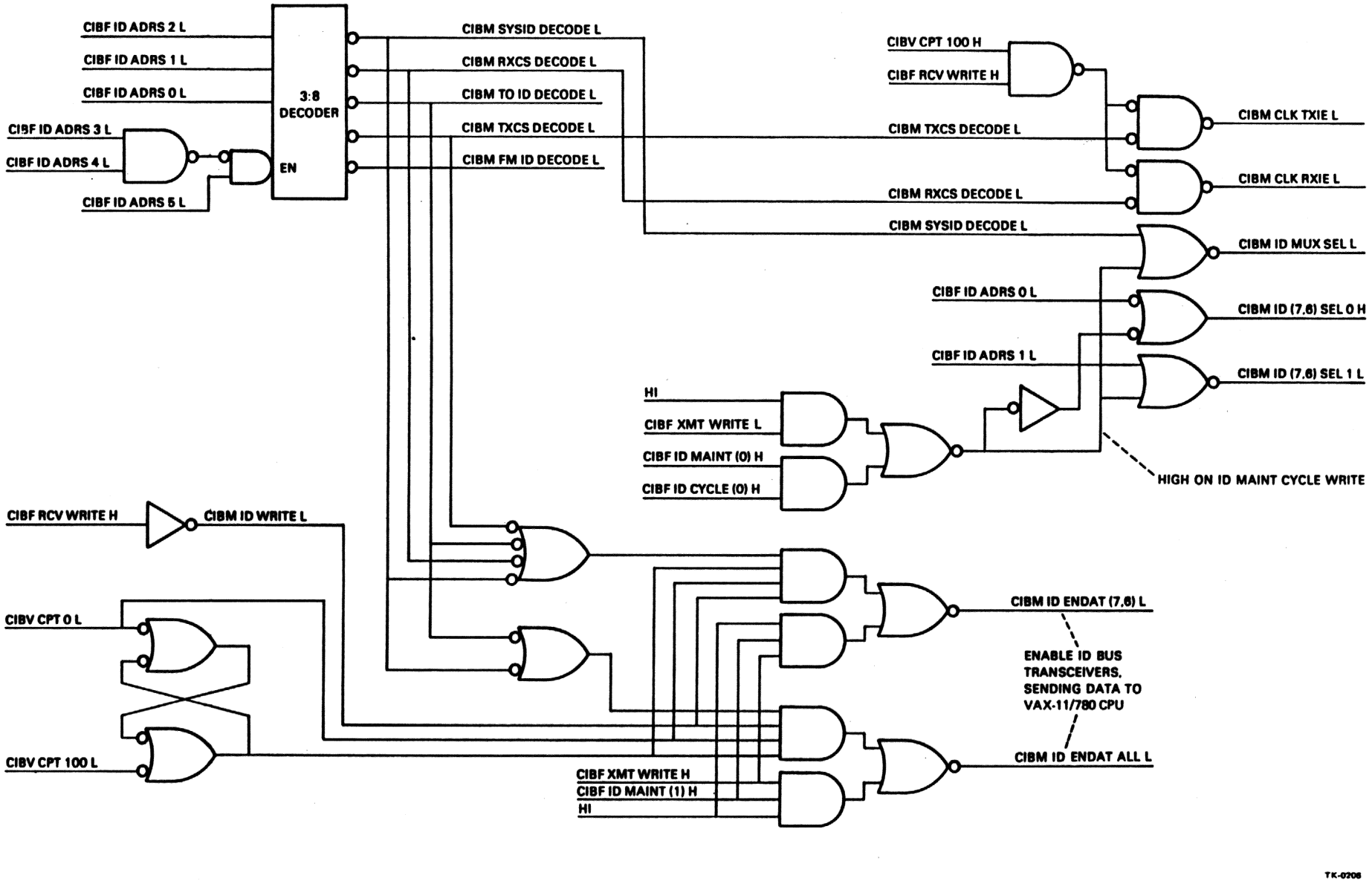


Figure 3-1 ID Bus Address Decoder Logic

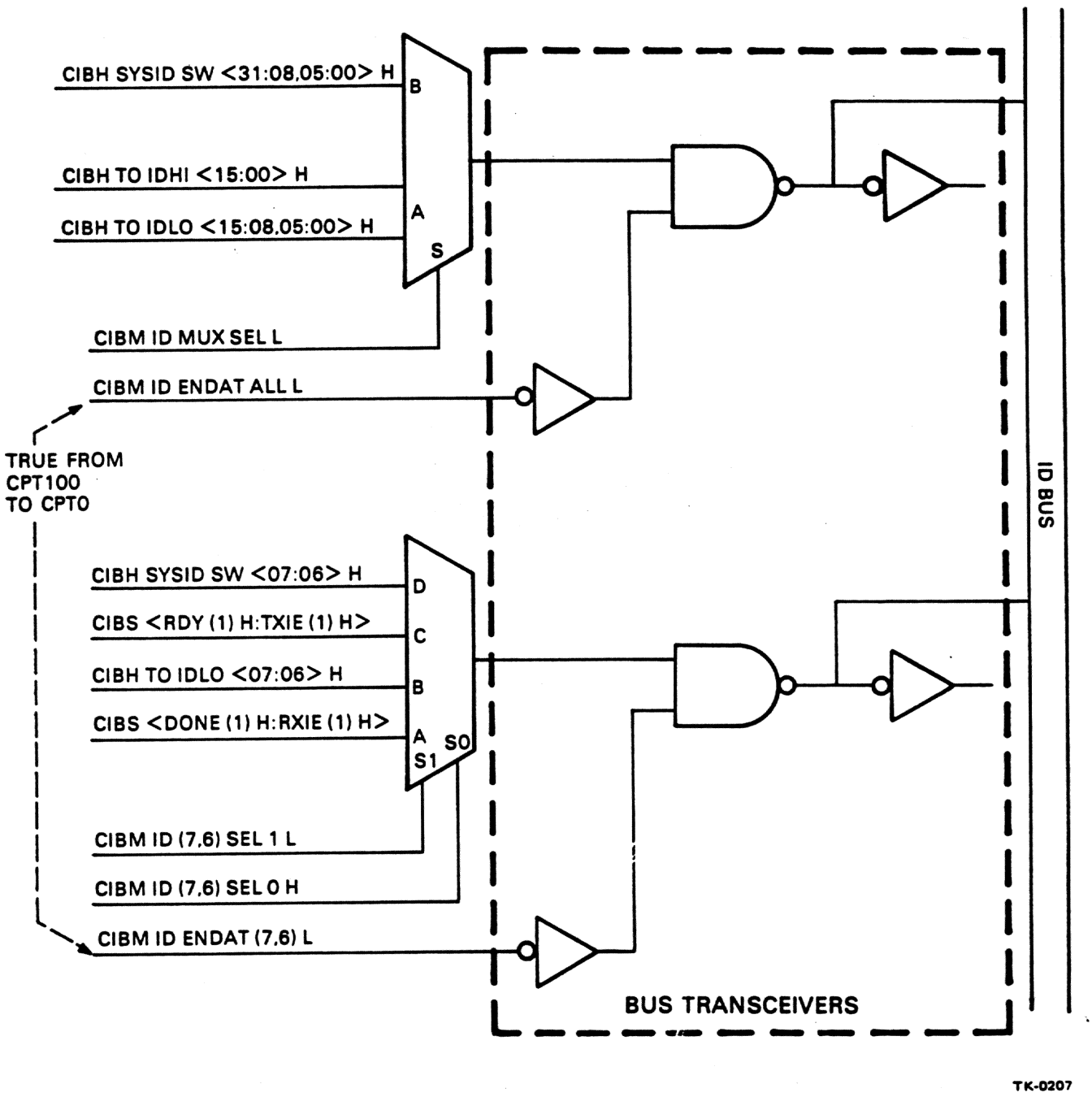


Figure 3-2 ID Data Multiplexer Logic

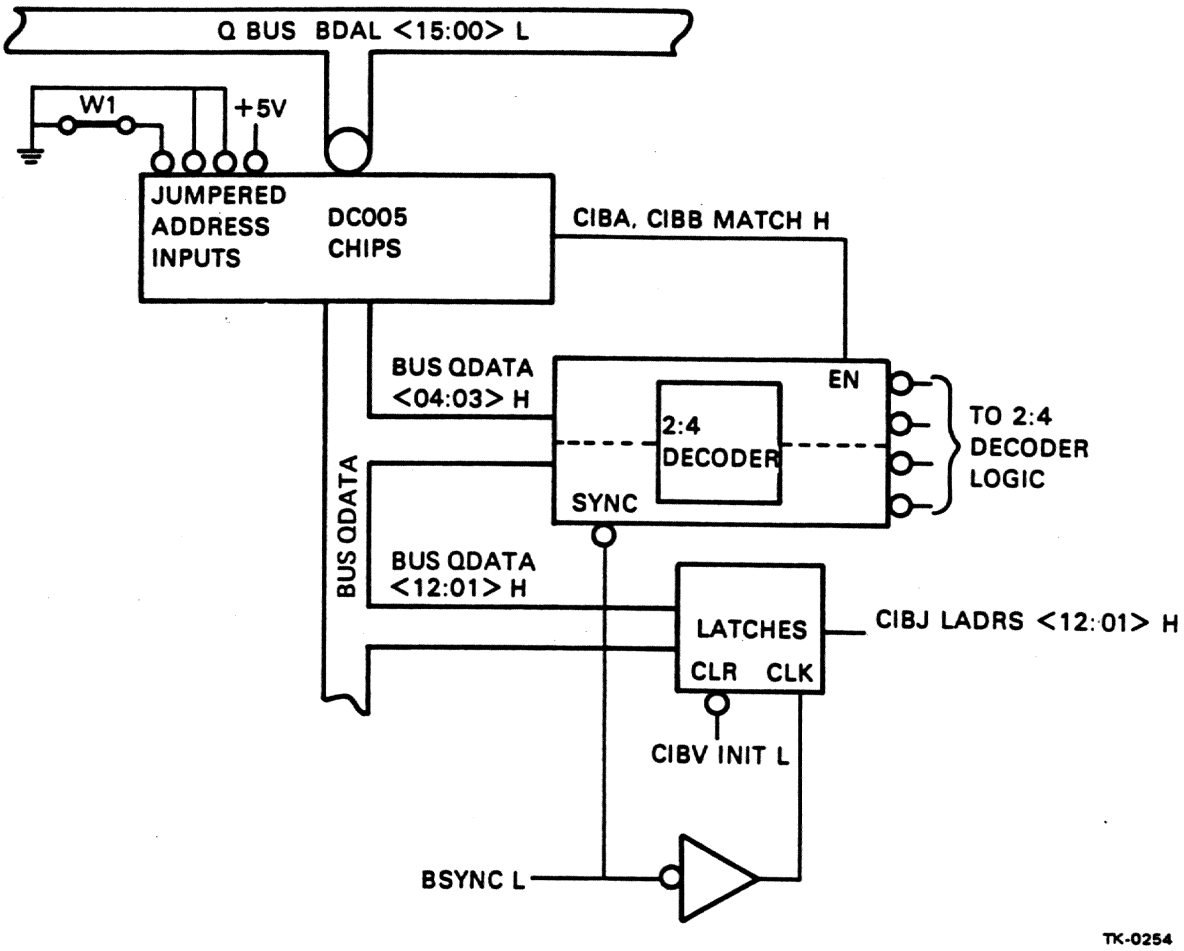


Figure 3-3 Q Bus Address Decoder Logic Interface Chips

3.3.2 QMUX Logic

Eight registers form input signals to the QMUX, as shown in Figure 3-4.

When the LSI-11 processor performs a read transfer to one of these registers, latched address bits <03:01> are used to select outputs of the register addressed, as shown in Figure 3-4. When CIBR QMUX STROBE L is asserted, it enables the contents of the selected register on the tristate output of the QMUX. Refer to Paragraph 3.3.4 for further details.

3.3.3 One of Twelve Decoder Logic

The LSI-11 processor can also read six of the CIB registers which do not feed the QMUX and it can perform write transfers to a total of eight CIB registers. The one of twelve decoder logic shown in Figure 3-5 selects the register addressed.

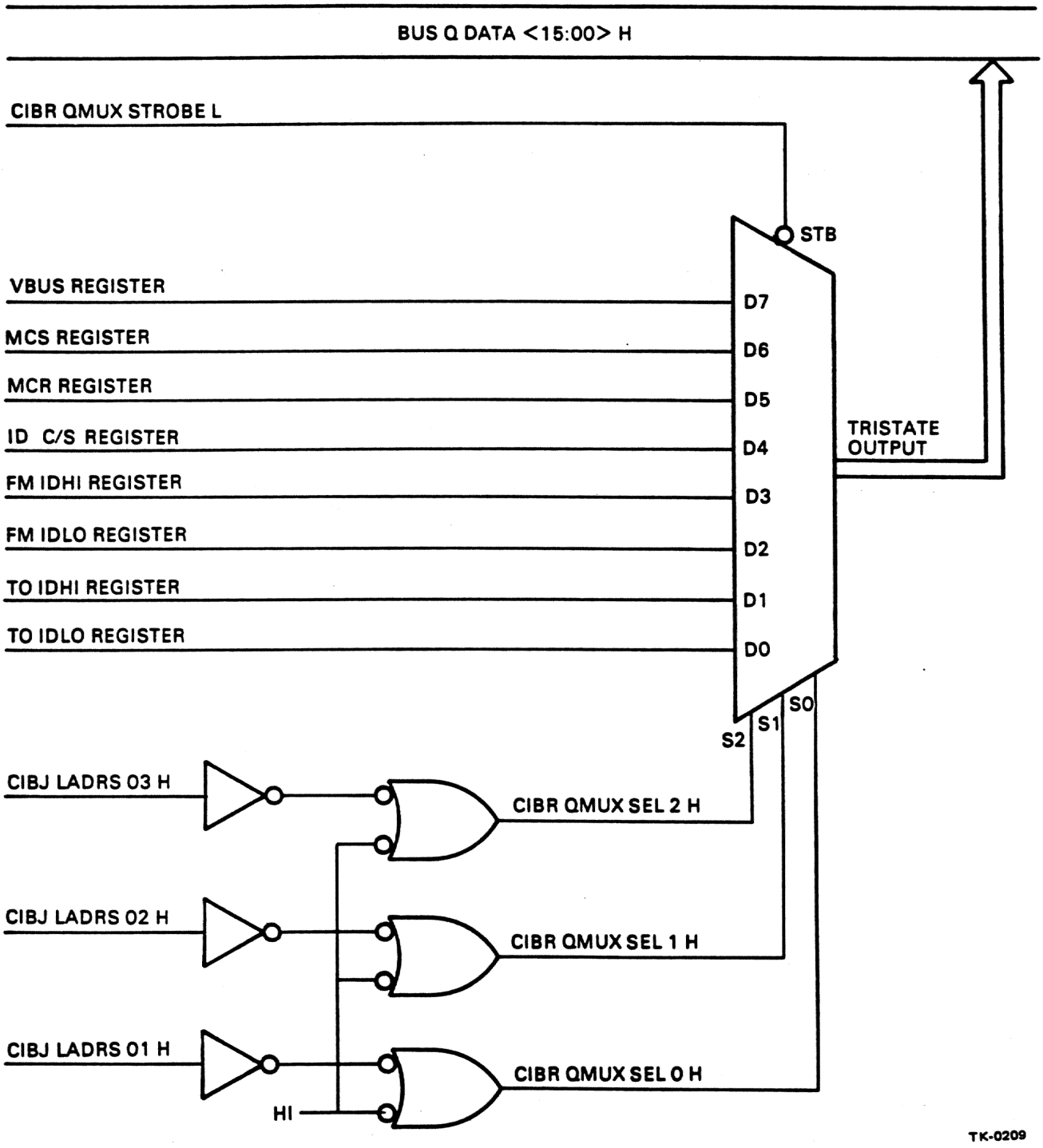


Figure 3-4 QMUX Logic

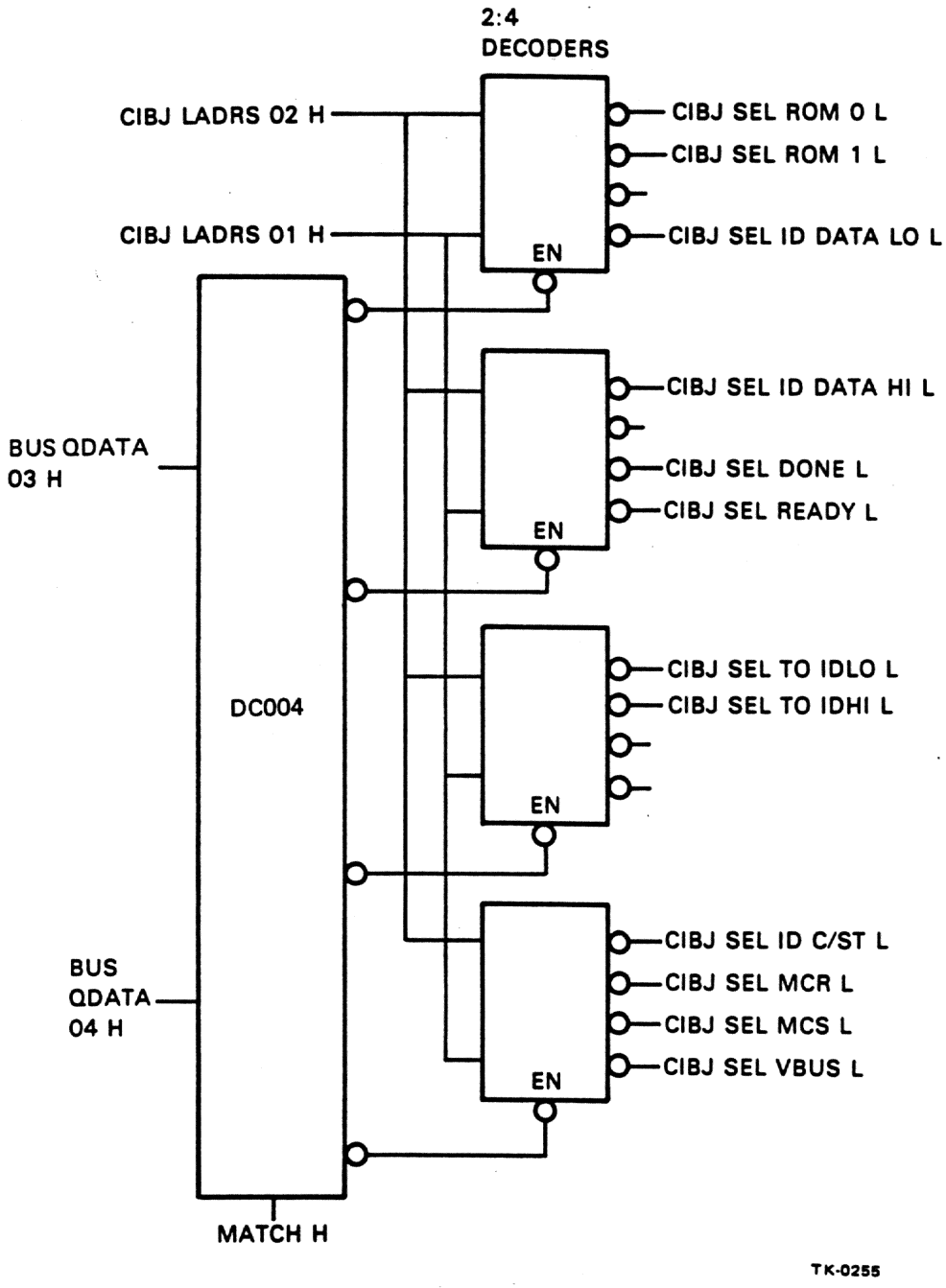


Figure 3-5 One of Twelve Decoder Logic

A decoder on the DC004 chip interprets the address bits on BUS QDATA <04:03> H, enabling one of four low true outputs when MATCH H is asserted. These signals in turn enable one of four 2:4 decoders. As shown in Figure 3-5, CIBJ LADRS <02:01> H select 1 of the 4 outputs of the enabled 2:4 decoder, enabling 1 of 12 select register signals. The following signals will enable a strobe signal gating the contents of the selected register onto the BUS QDATA <15:00> H lines when the LSI-11 processor performs a read transfer.

CIBJ SEL ROM 0 L
CIBJ SEL ROM 1 L
CIBJ SEL ID DATA LO L
CIBJ SEL ID DATA HI L
CIBJ SEL DONE L
CIBJ SEL READY L

When the LSI-11 processor performs a write transfer to one of the CIB registers, only the one of twelve decoder logic is used, enabling one of the following signals:

CIBJ SEL DONE L
CIBJ SEL READY L
CIBJ SEL TO IDLO L
CIBJ SEL TO IDHI L
CIBJ SEL ID C/ST L
CIBJ SEL MCR L
CIBJ SEL MCS L
CIBJ SEL VBUS L

See the descriptions of the specific registers for further details on read and write transfers from the LSI-11 processor to the CIB registers.

3.3.4 Strobe Signals Developed on a Q Bus Read Transfer

On a Q bus read transfer the LSI-11 processor negates the address after the CIB address logic has had time to decode and latch it. The LSI-11 then asserts BDIN L. The DC004 chip in turn enables CIBJ INWD L. This triggers a one-shot which sets a flip-flop on the trailing edge of the pulse, as shown in Figure 3-6.

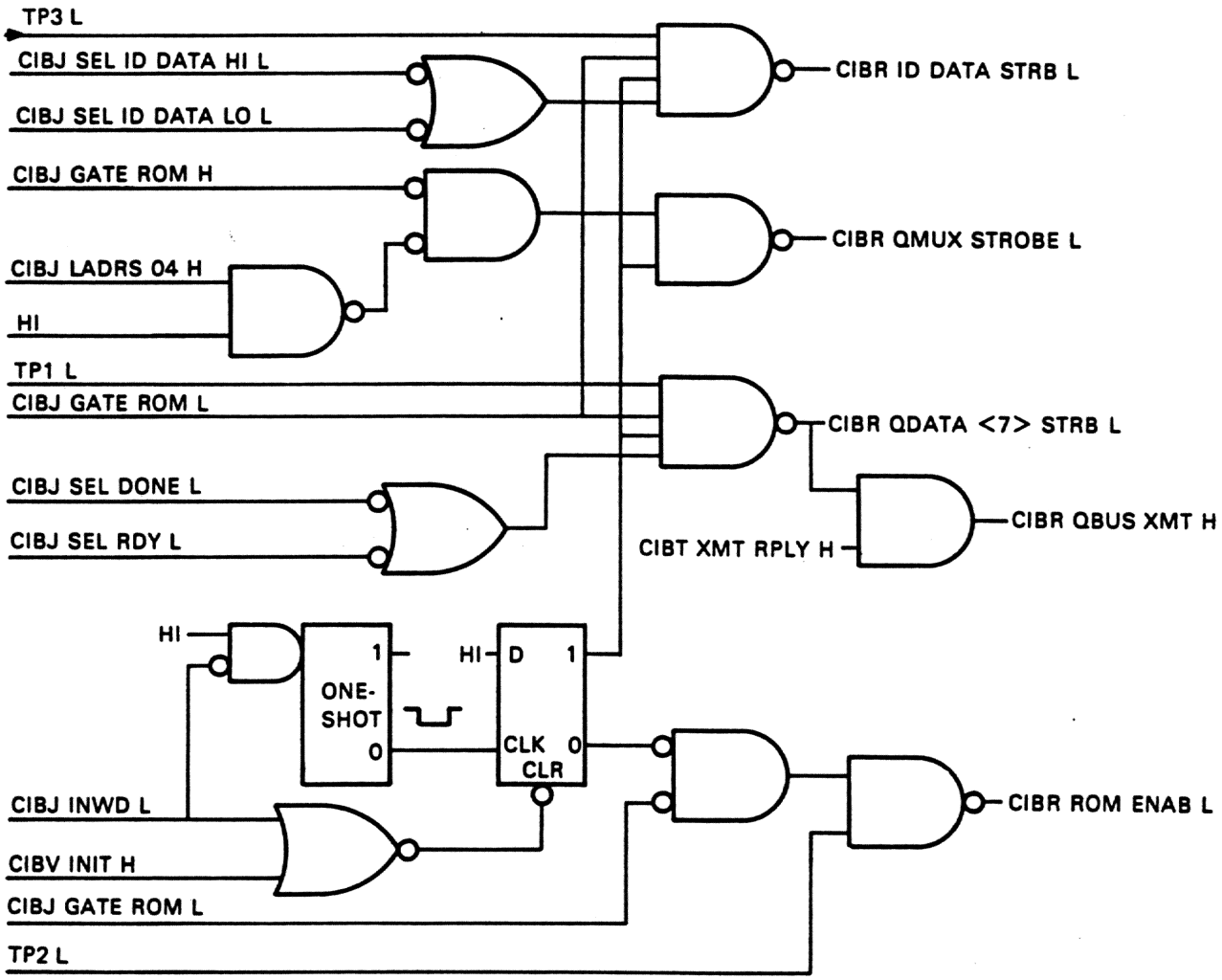
If CIBJ GATE ROM H is false and CIBJ LADRS 04 H is true, CIBR QMUX STROBE L is asserted, enabling the tristate output of the QMUX onto the BUS QDATA lines. The logic for CIBR ID DATA STRB L, CIBR QDATA <7> STRB L, and CIBR ROM ENAB L is similar.

3.3.5 Clocking the Registers on a Q Bus Write Transfer

When the LSI-11 processor performs a write transfer to a CIB register, it replaces the address on the BDAL lines with data to be written and asserts BDOUT L, after the CIB has decoded and latched the address. BDOUT L enables CIBJ OUTHB L or CIBJ OUTLB L or both on the DC004 chip, depending on the states of address bit 0 and BWTBT. These signals are ANDed with a synchronizing flip-flop signal (see Paragraph 3.3.6) to produce CIBT CLK HB L and CIBT CLK LB L. And these signals are in turn ANDed with one of the select signals from the one of twelve decoder logic (Figure 3-5) to produce 1 of the 15 register clock signals shown on sheet CIBU of the print set. Note that the data to be written is not latched until the clock signal(s) develop a rising edge.

3.3.6 Transfer Synchronization Logic

The VAX-11/780 CPU and the LSI-11 processor may happen to perform data transfers to the same register simultaneously. The synchronization logic shown in Figure 3-7 ensures that the transfer of unstable data will not result by making sure that CIBT XMT RPLY H and BRPLY L are not asserted until the CPT 50 following the CPT 150 which follows the assertion of CIBJ TRPLY by the DC004 interface chips.



NOTE:
TP1, TP2, AND TP3 ARE TEST POINTS WHICH MAY BE GROUNDED BY THE MAINTENANCE PERSON TO DISABLE THE RELATED CIRCUITS.

TK-0256

Figure 3-6 Register Strobe Logic

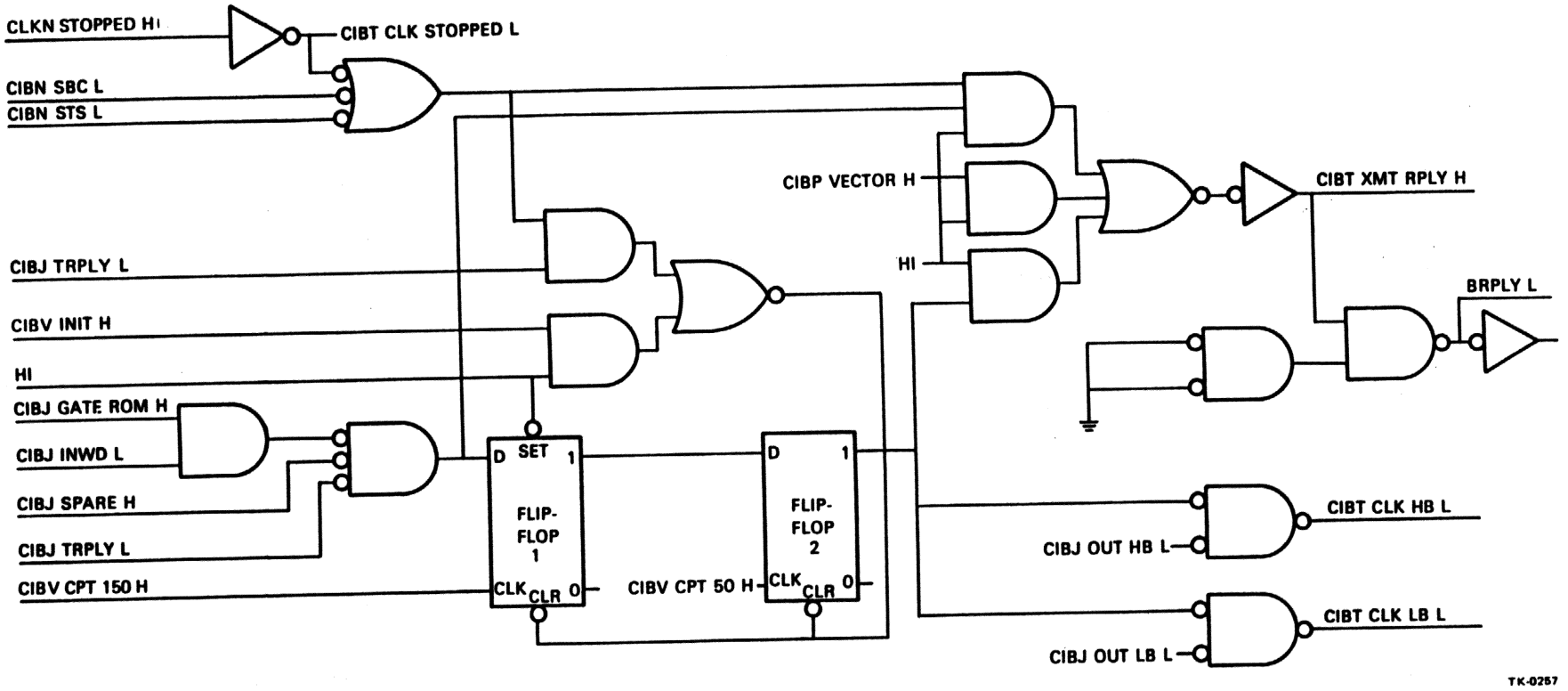


Figure 3-7 Transfer Synchronizer Logic

The two flip-flops shown in Figure 3-7 make up the core of the synchronization logic. A read or write transfer to any Q bus addressable CIB register will cause the first flip-flop to set on the CPT 150 following the assertion of CIBJ TRPLY L, if the VAX-11/780 CPU clock is running. The second flip-flop sets 100 ns later with the assertion of CPT 50. The output of the second flip-flop qualifies CIBT XMT RPLY H and BRPLY L.

If the transfer is a write, the rising edge of the output of the second flip-flop or the rising edge of CIBJ OUT HB L and/or CIBJ OUT LB L will produce a rising edge on CIBT CLK HB L and/or CIBT CLK LB L and thus on the clock signal(s) for the selected register, latching the data to be written.

Figure 3-8 shows the sequence of Q bus signals developed on a write transfer. Figure 3-9 shows how the timing developed by the synchronizing flip-flops ensures proper coordination with the VAX-11/780 CPU clock. Note that a delay between the LSI-11 write and the VAX-11/780 CPU read is built in. The data must be valid on the ID bus between CPT 131 and CPT 188.

If the transfer is a read to any Q bus readable register except TX READY or RX DONE, CIBT XMT RPLY H is ANDed with CIBR QDATA (7) STRB L (false) to qualify CIBR QBUS XMT H. This signal enables the transmit function of the transceiver gates on the DC005 chips, gating the data from the BUS QDATA <15:00> H lines to the Q bus. For further discussion of transfers to TX READY and RX DONE, refer to Paragraph 3.5.8. Figure 3-10 shows the sequence of Q bus signals developed on a read transfer. Figure 3-11 shows how the synchronization flip-flops affect the timing of the transfer. Note that a minimum of 100 ns delay between a VAX-11/780 CPU write and an LSI-11 read is certain.

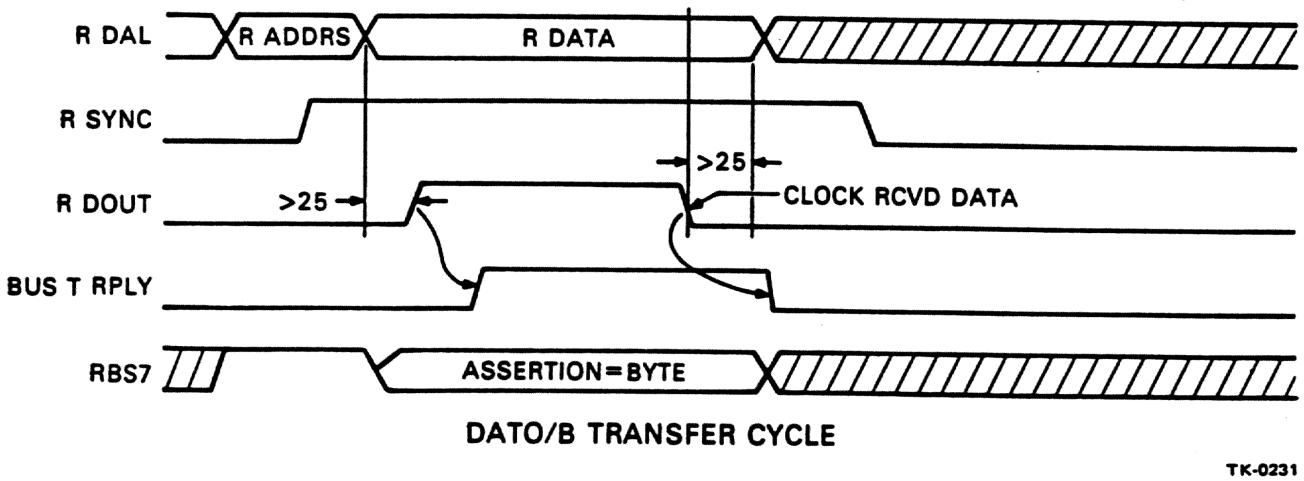


Figure 3-8 DATO/B Transfer Timing on the Q Bus

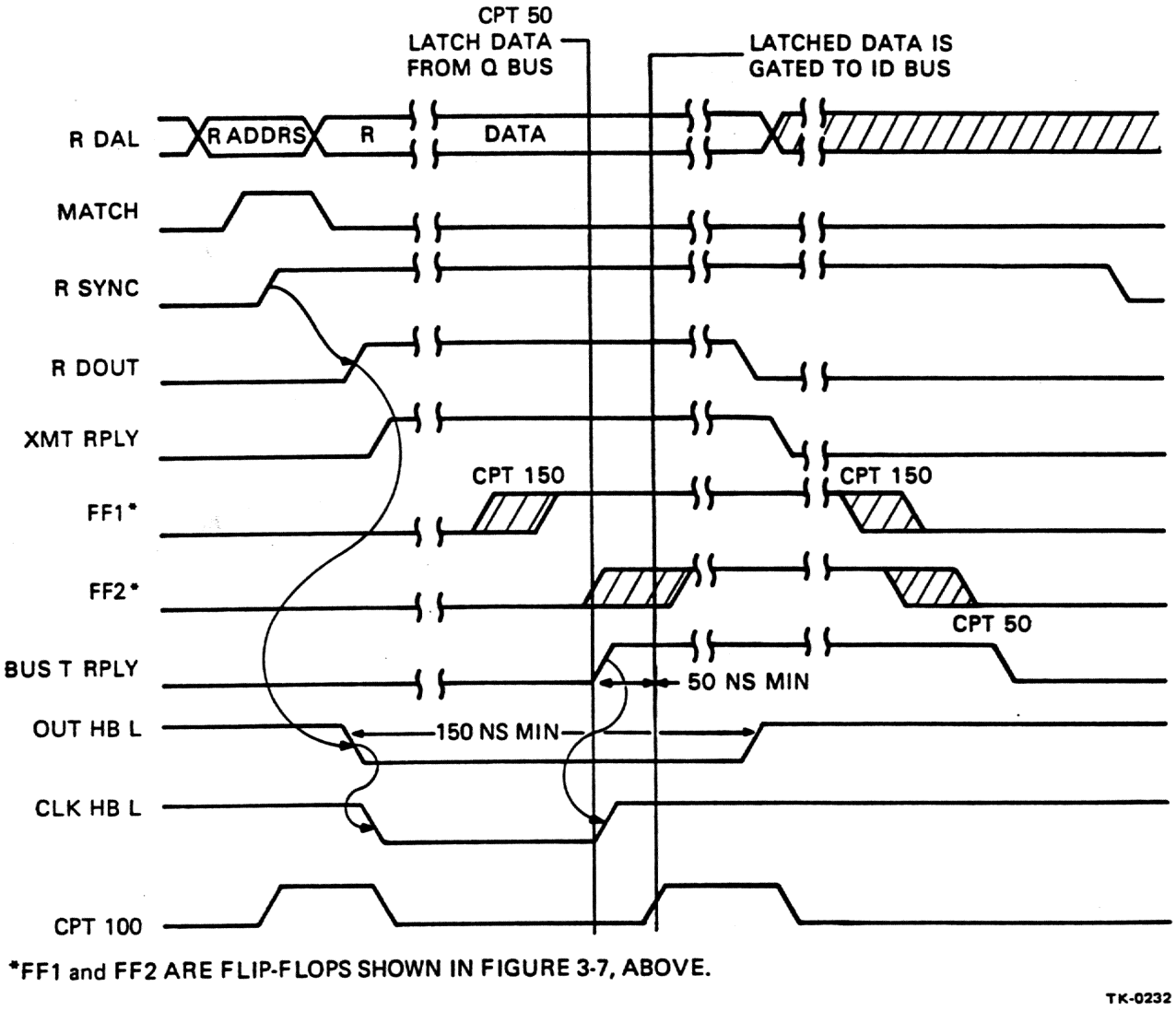


Figure 3-9 Synchronized DATO Transfer Timing Diagram

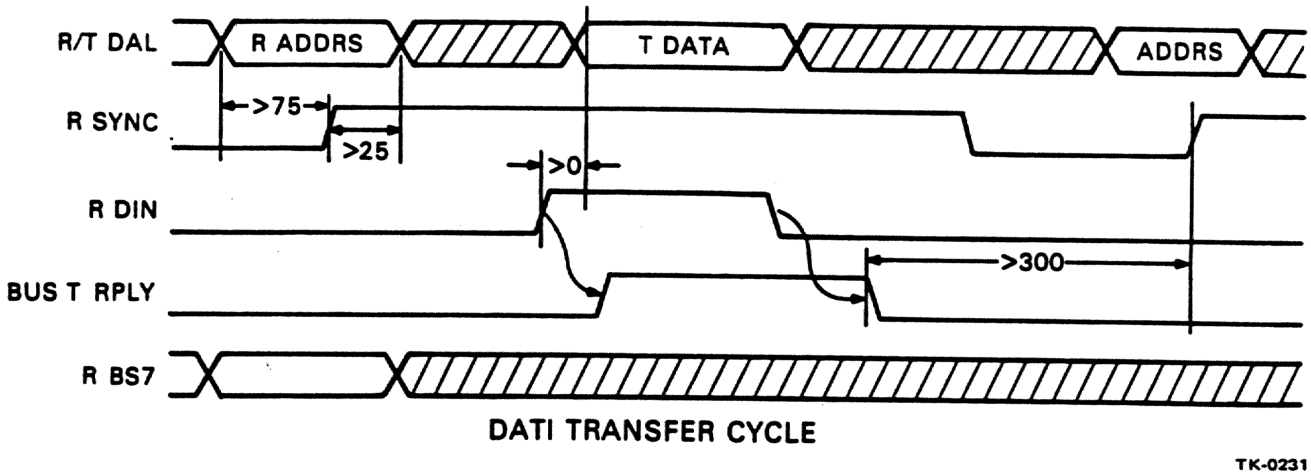


Figure 3-10 DATI Transfer Timing on the Q Bus

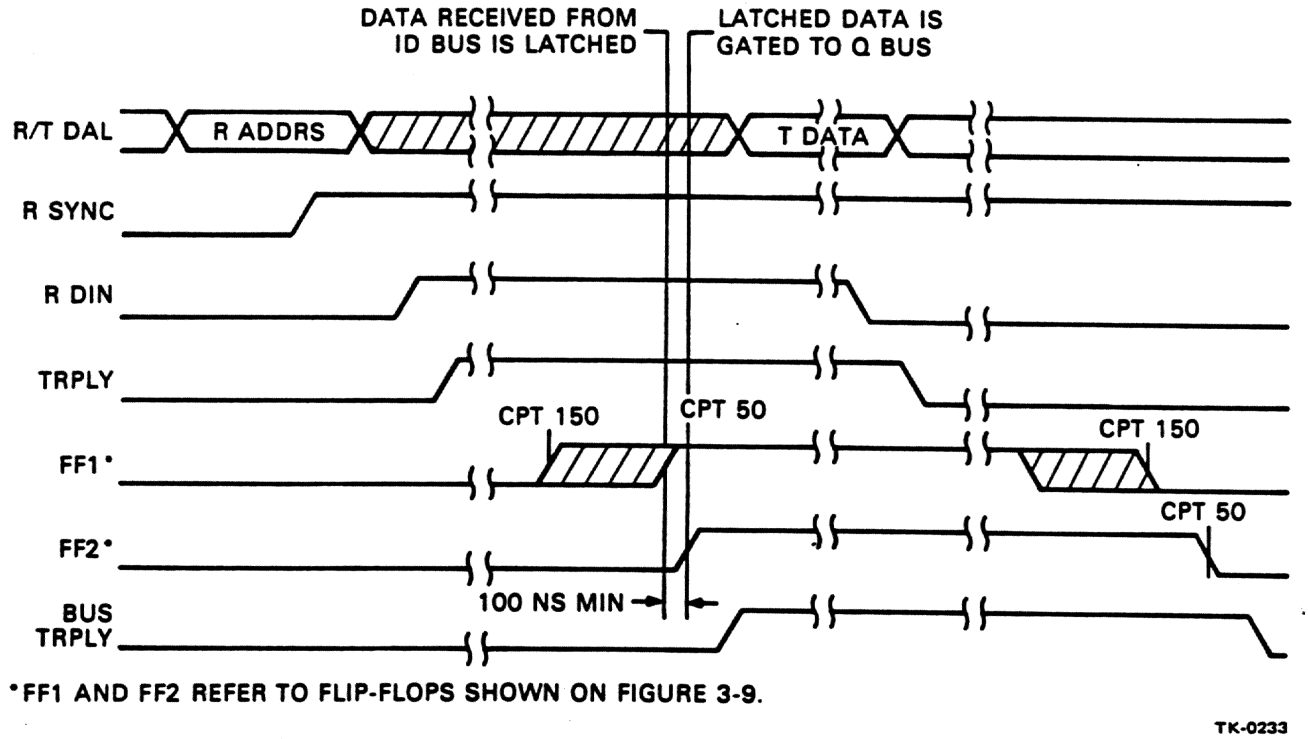


Figure 3-11 Synchronized DATI Transfer Timing Diagram

3.3.7 Q Bus Transfers when the VAX-11/780 CPU is Stopped

When the VAX-11/780 CPU clock is stopped, the synchronizing flip-flops are both cleared by the ANDing of CLKN STOPPED H and CIBJ TRPLY L (false). If the transfer is a write, CIBT CLK HB (and/or LB) L will be asserted low with the enabling of CIBJ OUTHB L and/or CIBJ OUTLB L. The D input to the first synchronizing flip-flop is also the signal which is ANDed with CLKN STOPPED H or CIBN SBC L or CIBN STS L, to produce CIBT XMT RPLY H and BRPLY L, as shown in Figure 3-8. This circuit becomes active when the DC004 interface chip asserts CIBJ TRPLY L in response to the initiation of a Q bus transfer, thus circumventing the synchronizing flip-flops when the clock is stopped. Note that on a write transfer the signal which clocks the data into the register will not develop a positive edge until CIBJ OUTHB L and/or CIBJ OUTLB L go high following the negation of BDOUT L by the LSI-11.

3.4 ROM LOGIC

The first two ROM locations have two sets of addresses: 163000(or 173000)/140000 and 163002(or 173002)/140002 (octal). These locations contain a JMP X instruction and they are addressed on power up through addresses $163000/173000_8$, $163002/173002_8$. When the LSI-11 places $163000/173000_8$ or $163002/173002_8$ on the Q bus address lines, CIBJ SEL ROM 0 or 1 L will be generated to perform the following functions.

1. Disable CIBR ROM AD <9:10> H.
2. Select ROM Bank 0 or 1, depending on the state of CIBJ LADRS 11 H.
3. Enable CIBJ GATE ROM L, generating CIBR ROM ENAB L after a delay following the receipt of BDIN L (CIBJ INWD L), enabling the tristate output of the selected ROM bank onto the BUS QDATA <15:00> H lines.

Figure 3-12 shows the ROM address logic.

When the LSI-11 processor subsequently reads ROM addresses in the 14XXXX range, BSYNC L sets the flip-flop shown in Figure 3-13 to generate CIBJ GATE ROM H and enable the output of one of the four ROM banks, selected by the latched address signals (CIBJ LADRS <12:01> H).

3.5 CIB REGISTER LOGIC

Not all of the CIB registers are discrete registers in the typical hardware sense. The illustrations which accompany the following descriptions should be useful in identifying the components and operations of the registers.

3.5.1 TO ID Register Logic

The TO ID register is the buffer used for data transmission from the LSI-11 to the VAX-11/780 CPU. The LSI-11 can write and read the contents of the register at Q bus locations $163020/173020_8$ and $163022/173022_8$, 16 bits at time. The VAX-11/780 CPU can read but not write the TO ID register at ID bus address 05_{16}. Figure 3-14 shows the TO ID register logic.

When the LSI-11 writes to the TO ID register one or two of four clock signals are needed to handle these transfers, as shown in Figure 3-14.

Note that the BUS QDATA signals are latched in the register flip-flops on the rising edge of a clock signal at CPT 50.

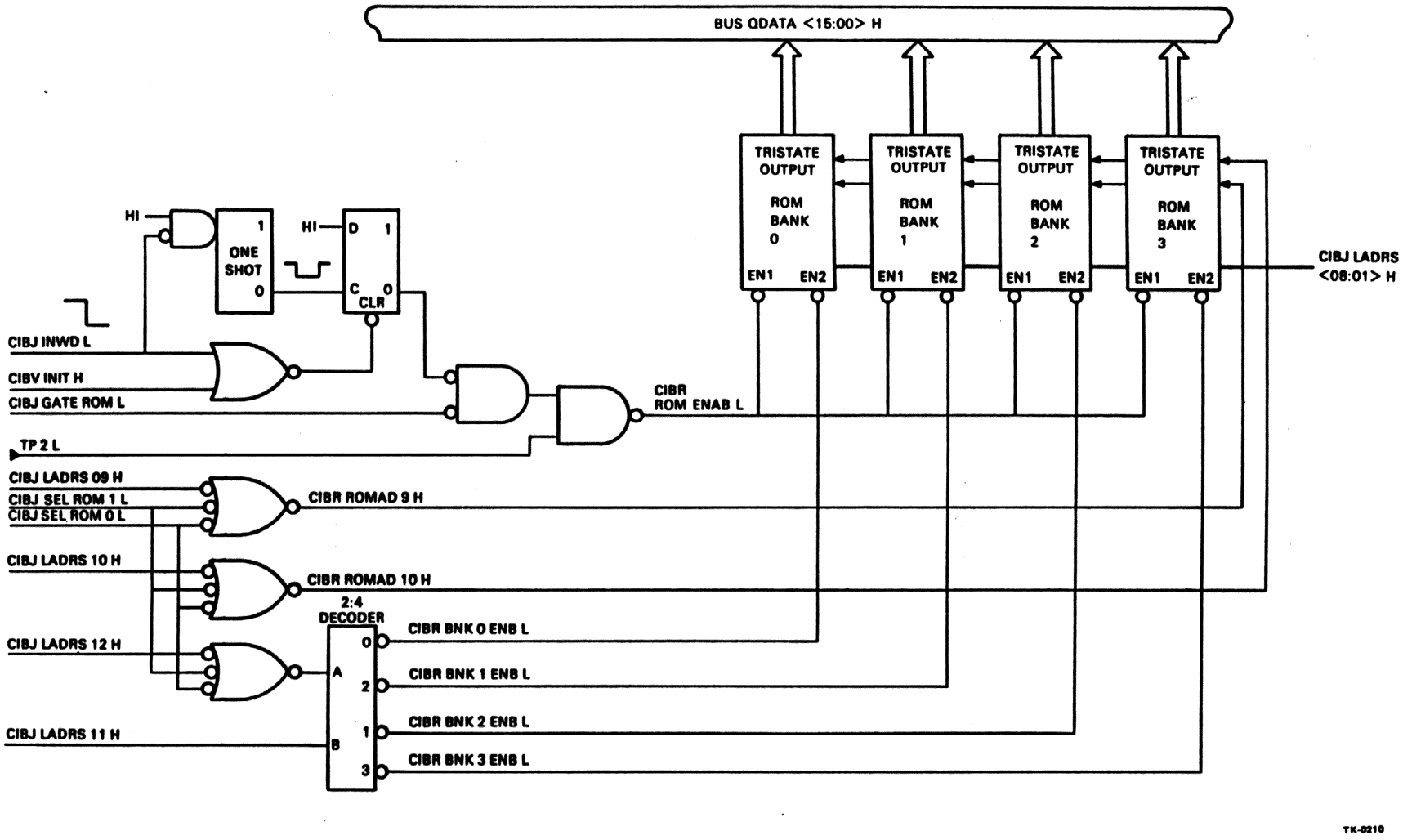


Figure 3-12 ROM Logic

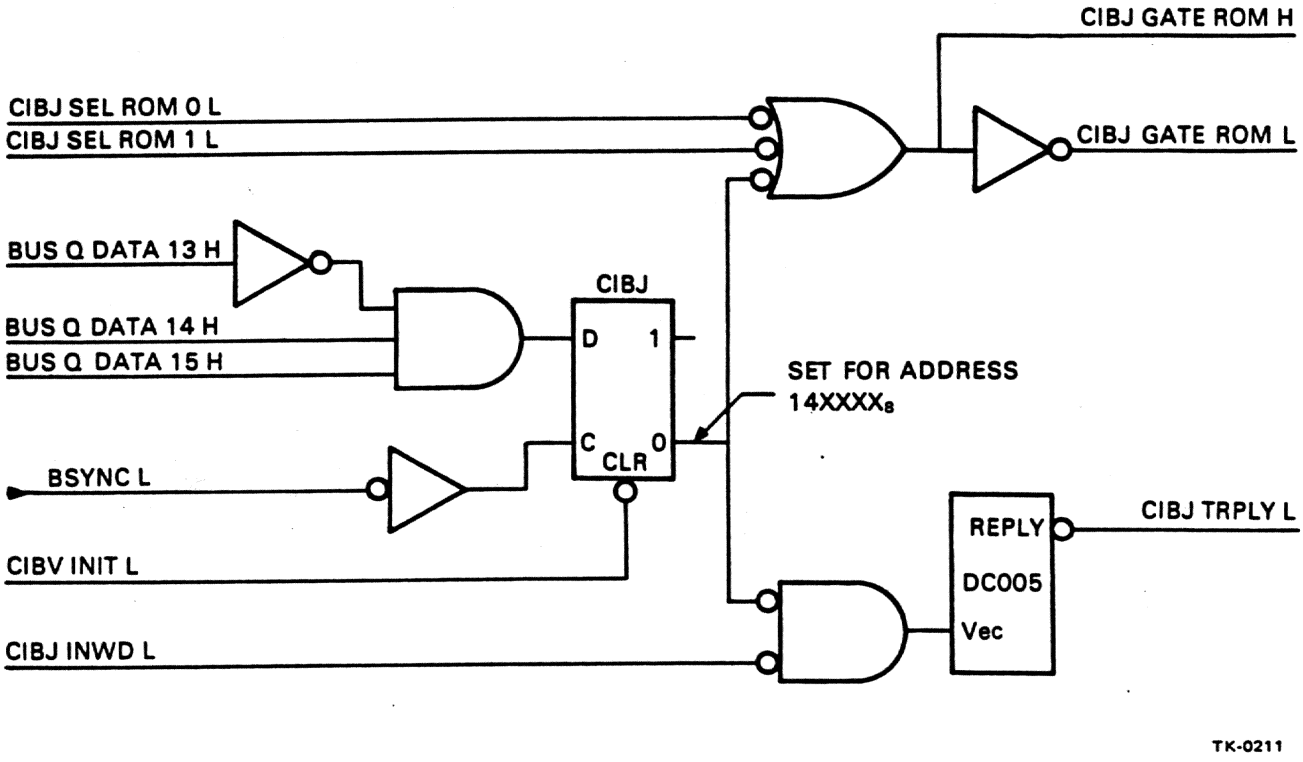


Figure 3-13 ROM Address Selection Logic

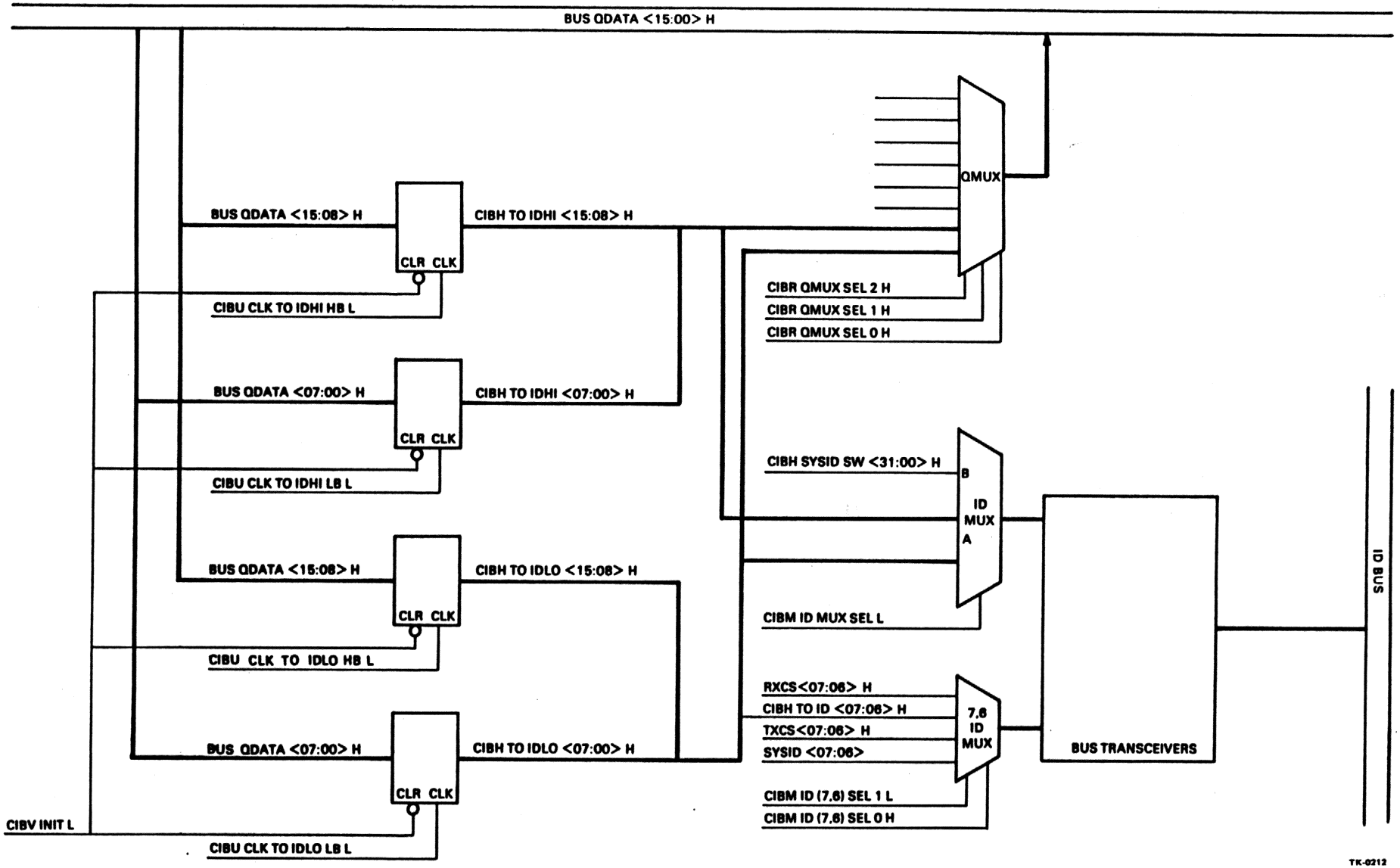


Figure 3-14 TO ID Register Logic

When the VAX-11/780 CPU places the address of the TO ID register on the ID bus address lines to read the register, the address decoder logic asserts CIBM ID MUX SEL L selecting the A inputs to the 2:1 ID multiplexer shown in Figure 3-2. Notice that bits 7 and 6 are handled separately in the 4:1 multiplexer. The decoding of the TO ID register address also enables CIBM ENDAT ALL L and CIBM ENDAT (7, 6) L at CPT 100, enabling the data through the ID bus transceivers.

3.5.2 FM ID Register Logic

The FM ID register is a write-only register from the VAX-11/780 CPU side and a read-only register from the LSI-11 side. The FM ID register is made up of positive edge triggered flip-flops. The clock input signal (CIBU CLK FM ID H) develops a positive edge under only two types of conditions, as shown in Table 3-1. Figure 3-15 shows the logic involved.

Table 3-1 Generation of the CLK FM ID H Signal

FM ID DECODE L	ID WRITE L	ID MAINT (0) H	XMT WRITE H	CPT 100	CLK FM IDH	COMMENTS
L	L	X	X	⌐└⌐	⌐└⌐	VAX-11 CPU writes to FM ID register.
X	X	L	L	⌐└⌐	⌐└⌐	CIB initiates a Maintenance Read Function.
H	X	H	X	X	H	On all other signal combinations no positive edge is developed.
H	X	X	H	X	H	
X	H	H	X	X	H	
X	H	X	H	X	H	

Note: X = don't care condition

The ID bus data, therefore, will be latched from the ID bus transceivers only when the VAX-11/780 CPU writes data to the FM ID register or when the CIB performs a maintenance read cycle on the ID bus. Figure 3-16 shows the FM ID register logic.

The output of the FM ID register feeds the QMUX. When the LSI-11 performs a DATI transfer to the high or low portion of the register, the contents of the FM IDLO register or the FM IDHI register are enabled on the BUS QDATA <15:00> H lines and sent to the LSI-11 with BRPLY L, at CPT 50.

3.5.3 ID C/S Register Logic

The ID C/S register enables the LSI-11 to control and monitor activity on the ID bus. When the LSI-11 performs a read transfer to the register, the Q bus address decoder logic and the QMUX select lines enable the output of the ID C/S register through the QMUX and on the BUS QDATA <15:00> H lines to be asserted on the Q bus together with BRPLY L.

ID C/S bits <14:08> are not latched on the CIB, but fed directly from ID bus receivers to the QMUX. Figure 3-17 shows the ID C/S register logic.

The setting of ID CYCLE causes ID MAINT to set at the next occurrence of CPT 0. ID MAINT (1) H then resets ID CYCLE at CPT 150 automatically, unless the VAX-11/780 CPU clock is stopped. The LSI-11 writes to the ID MAINT bit through the direct set and direct clear inputs to the flip-flop.

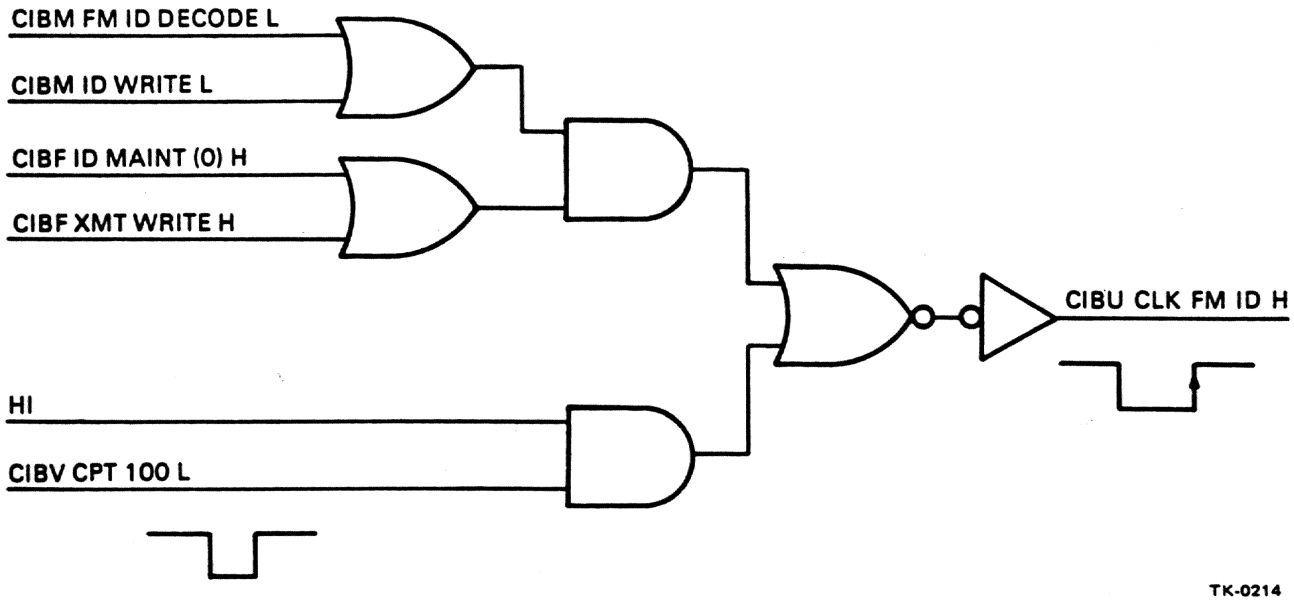


Figure 3-15 Development of CIBU CLK FM ID H

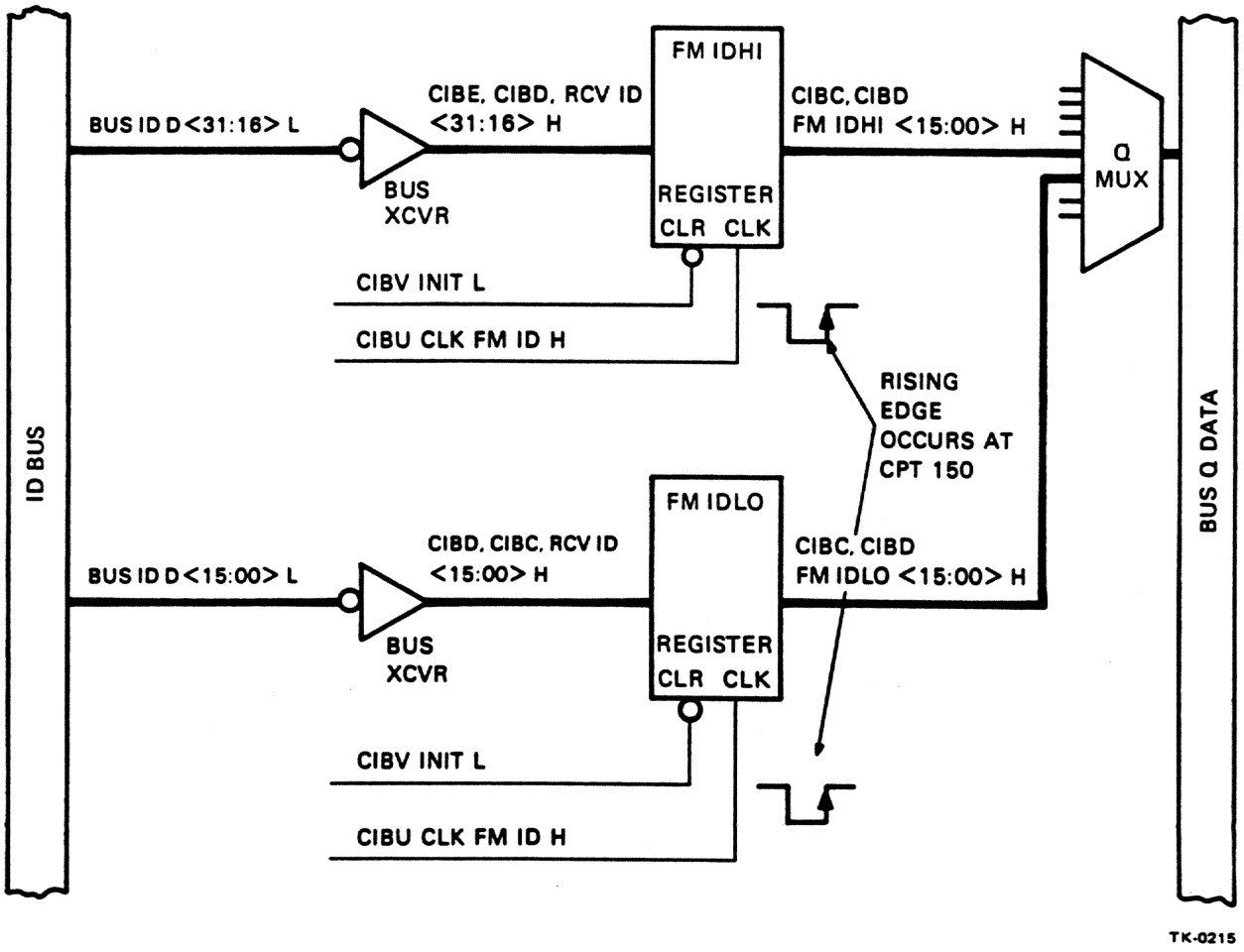


Figure 3-16 FM ID Register Logic

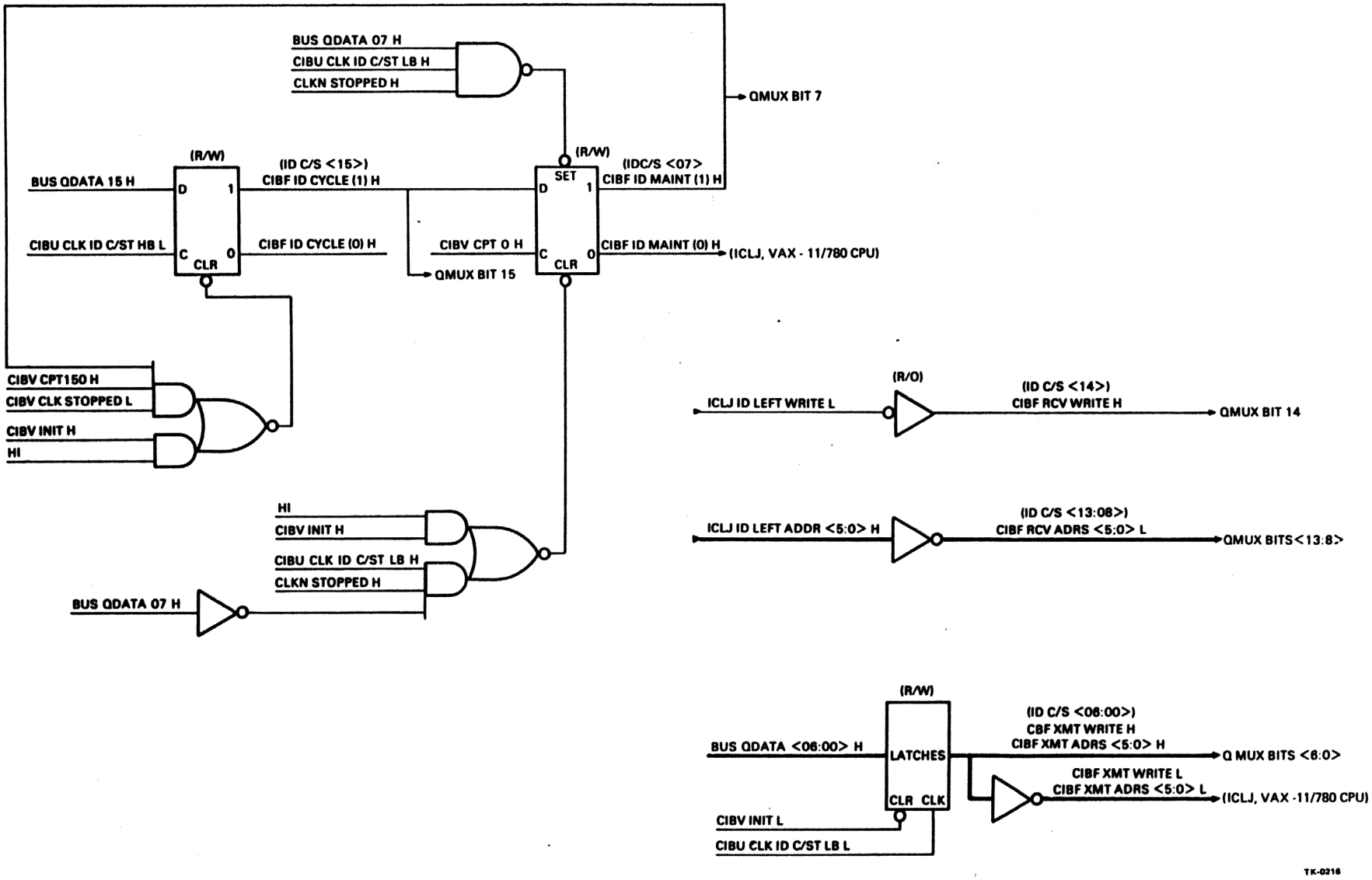


Figure 3-17 ID C/S Register Logic

3.5.4 MCR Register

The machine control register (MCR) enables the LSI-11 to control and monitor certain VAX-11/780 CPU functions. The LSI-11 reads the MCR register through the QMUX. Figure 3-18 shows the MCR logic.

When the LSI-11 processor writes a 1 to MCR bit 10, MAINT RTN EN, a second flip-flop sets when CPT 150 H goes high, enabling CIBN DMAINT RTN H. The 1 output of this second flip-flop also loops back to clear the MAINT RTN EN flip-flop, putting a low level on the D input to the second flip-flop. CIBN DMAINT RTN is thus disabled after one bus cycle at the next occurrence of CPT 150.

MCR bit 5 is not latched on the CIB but formed directly from CLKN STOPPED H.

MCR bit 0, PROCEED, does not feed the QMUX and thus cannot be read. As soon as the LSI-11 writes a 1 to PROCEED, the VAX-11/780 CPU clock starts. CIBT STOPPED L then goes high immediately, clearing the flip-flop.

3.5.5 MCS Register Logic

The LSI-11 uses the miscellaneous control status (MCS) register to monitor conditions on the control panel (SCP) and in the VAX-11/780 CPU and to enable or disable interrupts from the CIB to the LSI-11. Figure 3-19 shows the MCS Register logic.

The LSI-11 reads the MCS register through the QMUX. Notice that LOCK, REMOTE, AUTO-RESTART, CNSL ACK, RUN, and ATTN are read-only bits.

MCS bit 8, the RUN signal, is developed from a series of two one-shots. The 1 side of the first one-shot feeds negative inputs of both one-shots. When the VAX-11/780 CPU starts running, ICLD ISTR H goes high, firing both one-shots and enabling CIBN RUN H. At this point the 1 output of the first one-shot disables the negative inputs to both one-shots. The first one-shot therefore times out after 211.5 μs, whether or not ICLD ISTR H cycles again, thus reenabling the negative inputs to both one-shots. The second one-shot carries for 423 μs without retriggering, however, so that RUN remains asserted. ICLD ISTR H will be asserted again, before the second one-shot times out, if the VAX-11/780 CPU is still running. This fires the first one-shot, retriggers the second one-shot, and keeps CIBN RUN asserted.

Notice also that the LSI-11 can clear, but not set, the BOOT bit.

The LSI-11 reads the MCS register regularly when going through a null loop and on power up in order to take appropriate action in response to a change in the status of the VAX-11/780 CPU or the control panel.

3.5.6 V Bus Register Logic

The LSI-11 can read the V bus register at any time, and it can set and clear the self test and load bits at any time, but it should write a 1 to the clock bit (bit 0) only when the VAX-11/780 CPU clock is stopped and the CPU is in a stable state. Setting the clock bit at any other time may result in the receipt of invalid data on the eight V bus channels. Figure 3-20 shows the V bus register logic.

The LSI-11 reads the V bus register through the QMUX. Bits <15:04> are not latched, but fed directly from the VAX-11/780 CPU to the QMUX. CPT 0, 50, 100, and 150 H do not form a part of the V bus; they are included on the V bus register so that the maintenance person can determine the state of the VAX-11/780 CPU clock during the reading of the V bus lines.

The CIBU CLK VB LB L signal clocks the data to be written into bits 0, 1, and 2 on a write transfer. Notice that bit 0, the SDMS CLOCK bit, can be written as a 1 only, and is self clearing, but that the output, CIBH SDMS CLOCK H is normally high, and goes low when the bit is clocked, for the period of the one-shot, as shown in Table 3-2. Figure 3-21 shows the timing involved.

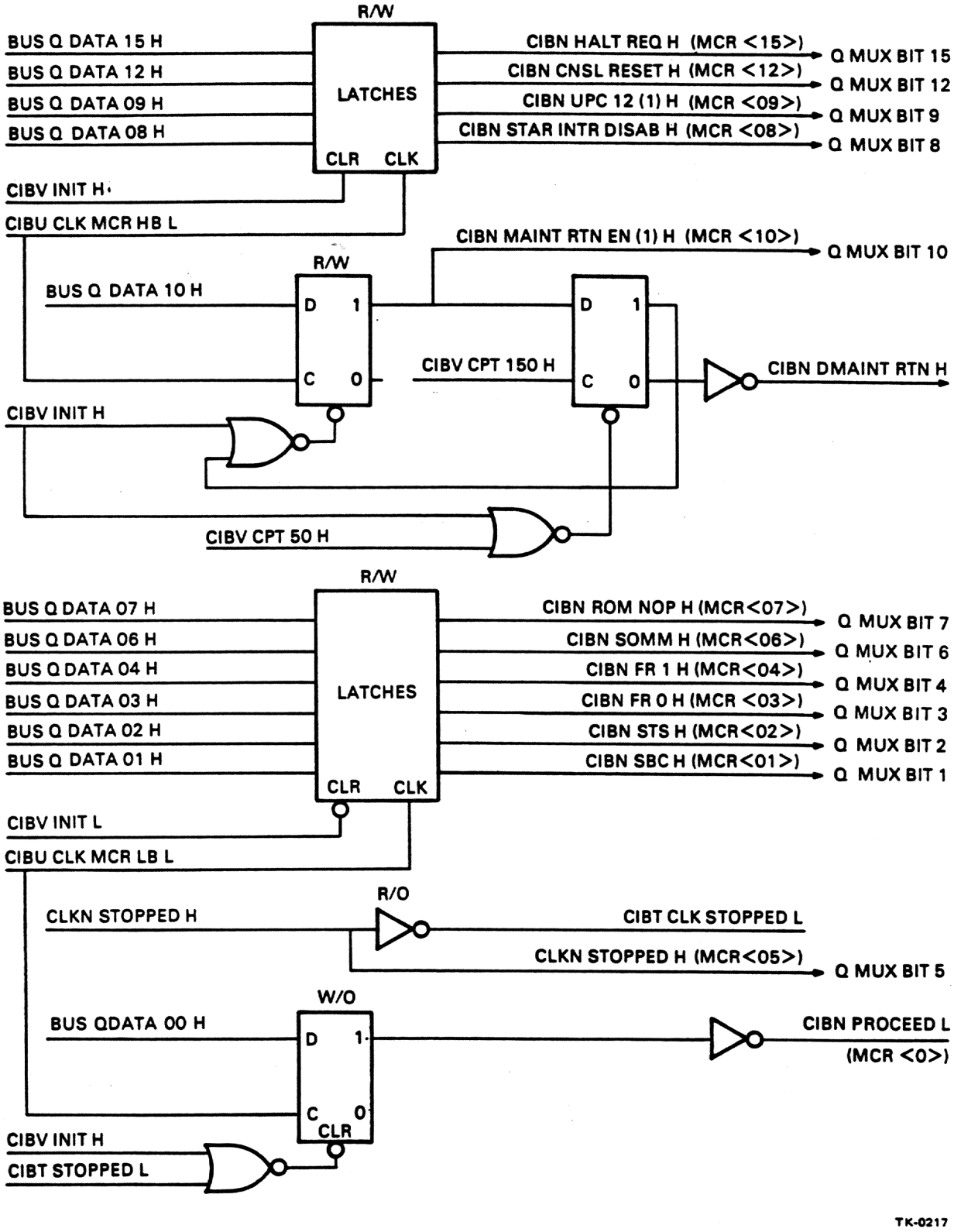


Figure 3-18 Machine Control Register Logic

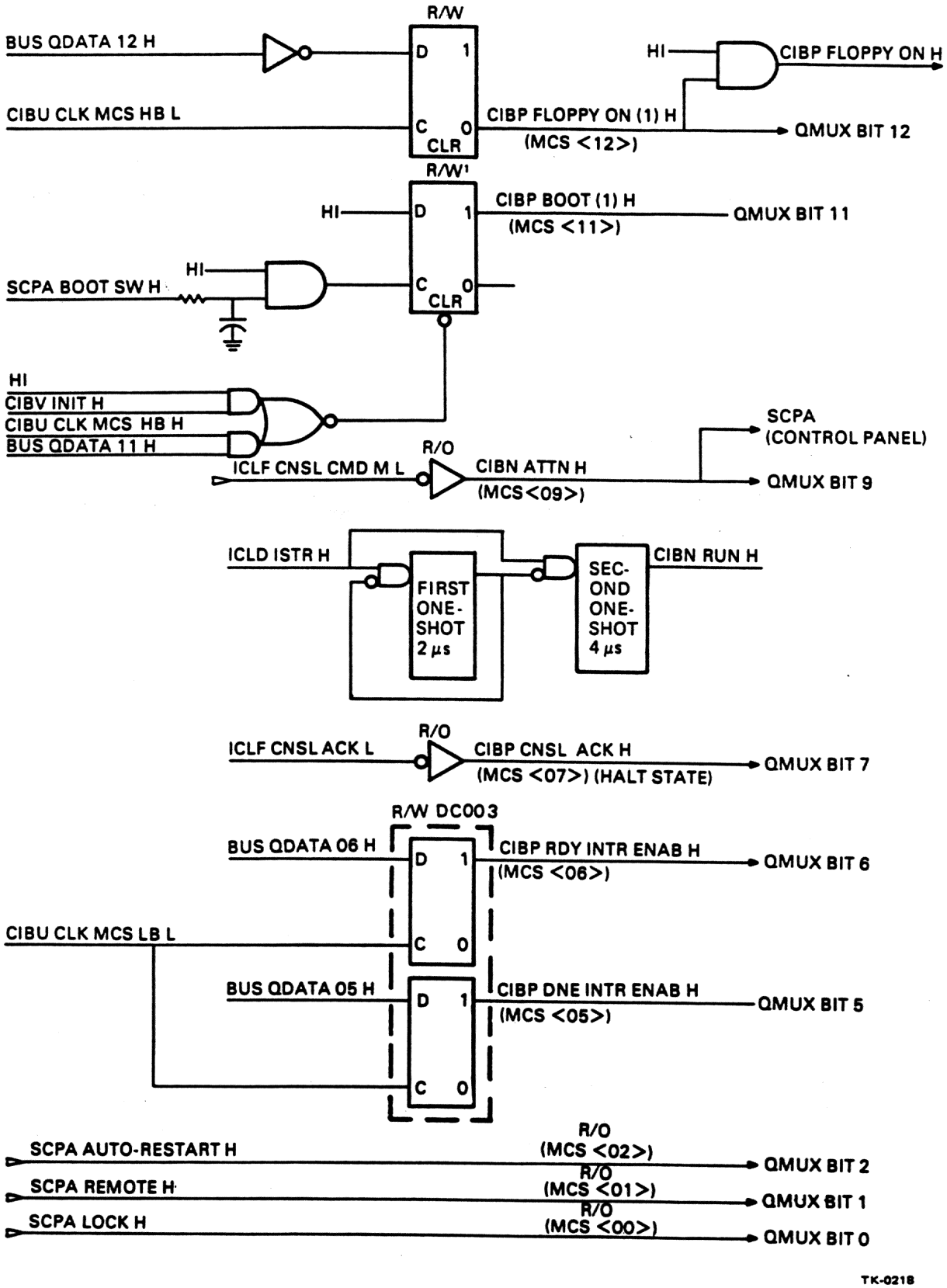


Figure 3-19 MCS Register Logic

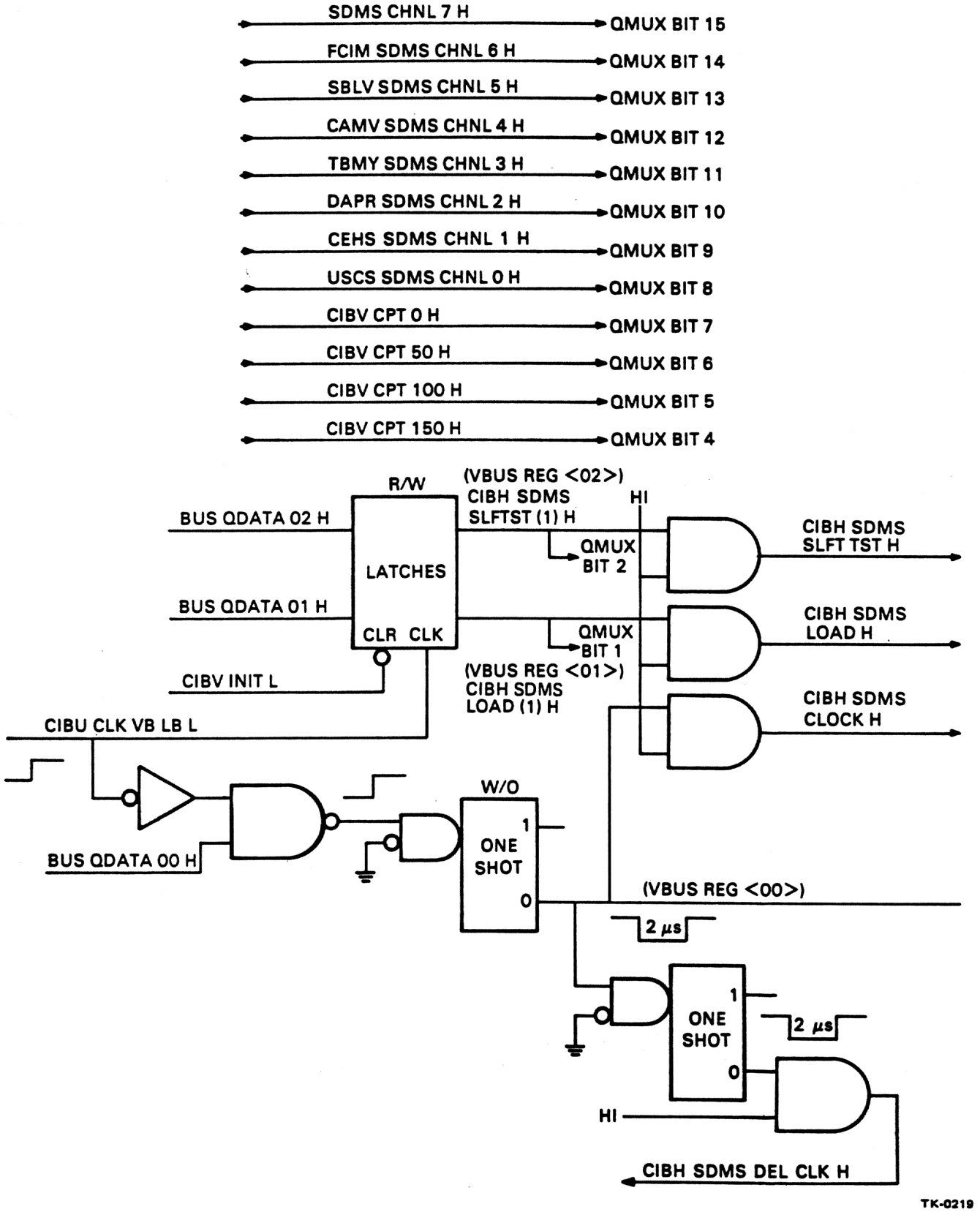


Figure 3-20 V Bus Register Logic

Table 3-2 Generation of the SDMS Clock H Signal

CIBU Clock V Bus L B L	Bus Q Data 00	One-Shot Input (Pin 2)	One-shot Output 0 CIBH SDMS Clock H	Comment
H	L	H	H	Normal condition
H	H	H	H	
L	H	↓	H	LSI-11 writes a 1 to bit 0
H	H	↑	⊓	Negative pulse on output clocking shift register on leading edge (through receivers)

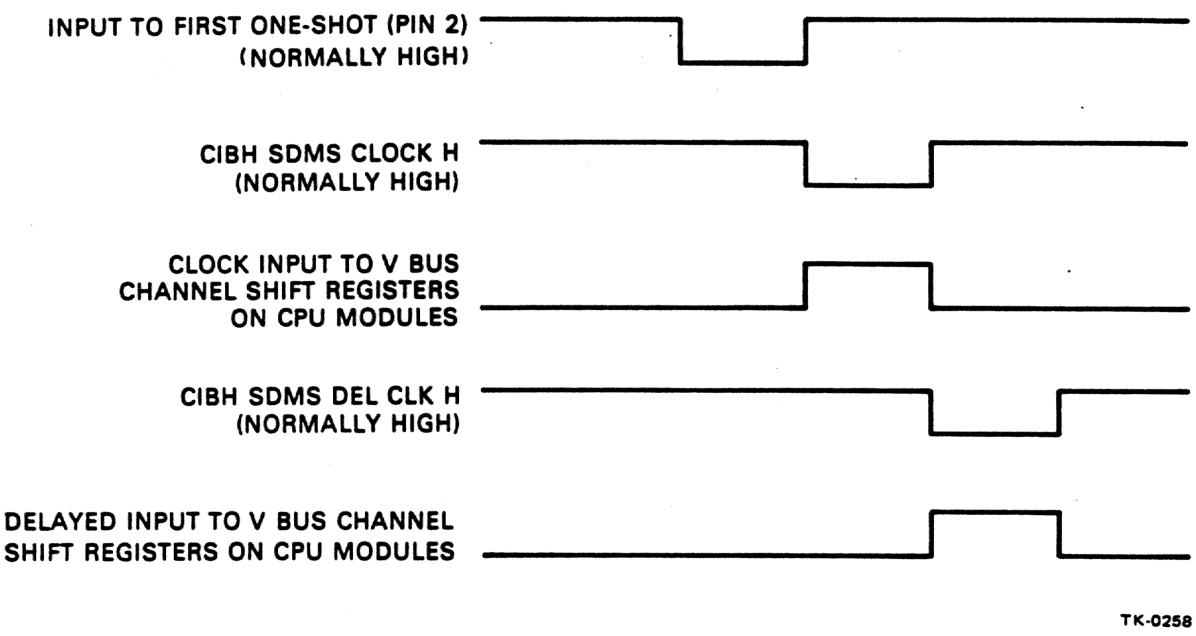


Figure 3-21 V Bus Clock Timing Diagram

When the one-shot fires, the CIBH SDMS CLOCK H signal stays low for about 2 μs. The CPU modules which use this signal to clock the V bus shift registers invert the signal to develop a positive pulse on the clock inputs to the shift register chips. The positive edge of the one-shot 0 output also triggers a second one-shot, providing the CIBH SDMS DEL CLK H. The delay produced is 2 μs.

3.5.7 ID Data Register Logic

The ID data register provides a window to the ID bus for the LSI-11. It is unique in that it contains no storage elements and is a read-only register, as shown in Figure 3-22.

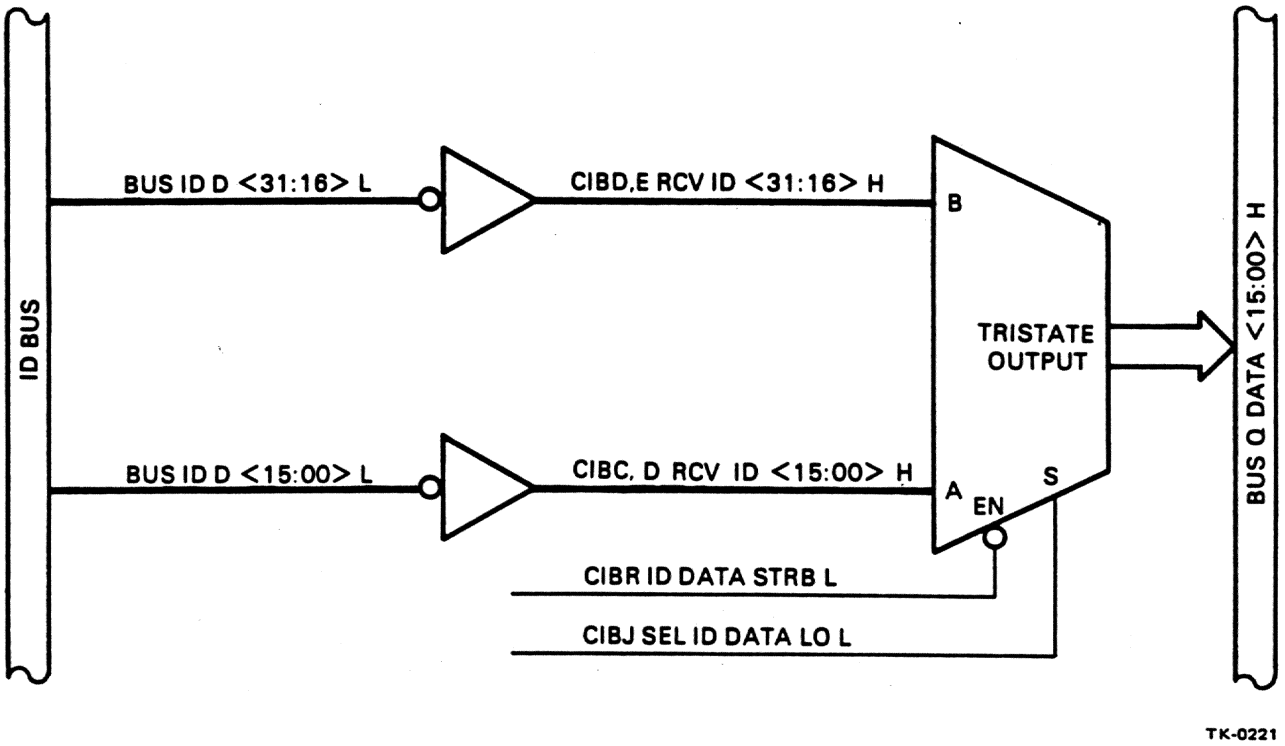


Figure 3-22 ID DATA HI and ID DATA LO Register Logic

When the LSI-11 performs a read transfer to either the ID DATA LO register or the ID DATA HI register, the Q bus address decoder logic enables CIBJ SEL ID DATA (LO or HI) L, selecting the A or B inputs to the 2:1 multiplexer shown in Figure 3-22. CIBR ID DATA STRB L enables the tristate output of the multiplexer on the BUS QDATA <15:00> H lines after a delay following the receipt of BDIN from the LSI-11.

3.5.8 Interrupt Logic

The interrupt logic enables the LSI-11 and the VAX-11/780 CPU to interrupt each other. It involves two ID bus registers and three Q bus registers. Like other registers on the CIB, the RXCS and TXCS registers permit access by the VAX-11/780 CPU, and the RX DONE, TX READY, and MCS (described above) registers permit access by the LSI-11 via the address decoder mechanisms.

When the VAX-11/780 CPU writes a 1 to bit 6 of the RXCS register or the TXCS register, the ID bus address decoder logic clocks the bit at CPT 100, enabling interrupts to the VAX-11/780 CPU. Then, when the LSI-11 writes a 1 to the READY or DONE bit, the latched signal is ANDed with the corresponding interrupt enable bit to clock an interrupt pending flip-flop and enable an interrupt signal to the VAX-11/780 CPU, if the CIBN STAR INTR DISAB H signal is false.

Later, when a VAX-11/780 interrupt subroutine reads the TO ID register or writes the FM ID register, the DONE or READY flip-flop will clear automatically when the corresponding ID bus address decoder signal is enabled, as shown in Figure 3-23. ICLC CNSL RCV ACK L (which indicates the halt state in the VAX-11/780 CPU), CIBV INIT H, or CIBN CNSL RESET H will reset the interrupt pending flip-flops, preventing inadvertent interrupts.

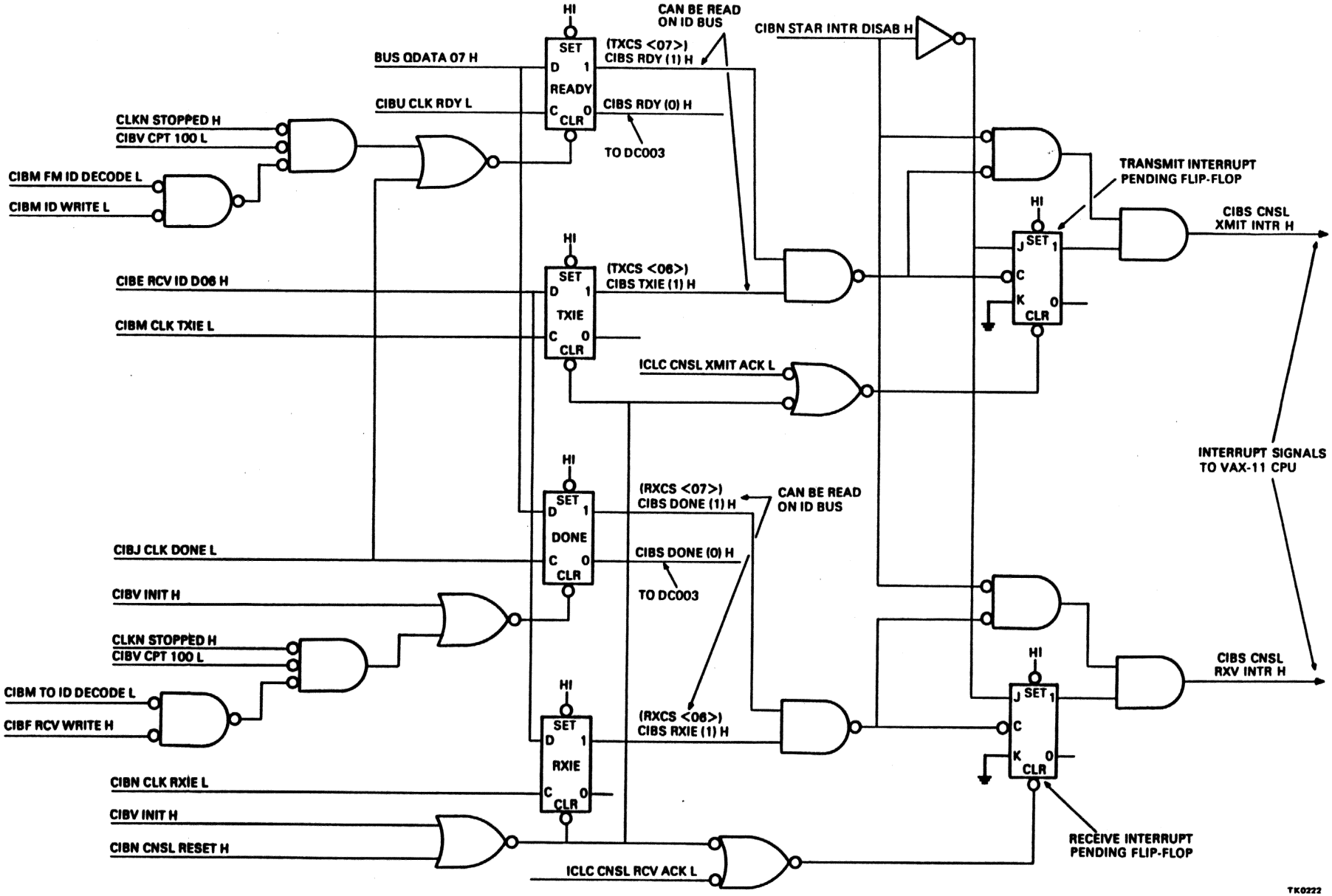


Figure 3-23 VAX-11/780 CPU Interrupt Logic

Also, the READY and DONE bits are included in two sets of registers so that they are accessible to both the LSI-11 and the VAX-11/780 CPU. From the ID bus side they are read-only bits which clear automatically. From the Q bus side they are read/write bits.

The LSI-11 processor reads the READY and DONE bits through a 2:1 tristate multiplexer as shown in Figure 3-24. Note that the other data inputs to the multiplexer are grounded. CIBR QMUX SEL 0 H selects the A or B input to the multiplexer. CIBR QDATA <7> L enables the data onto the tristate BUS QDATA lines (see Paragraph 3.3.4 for an explanation of the way this signal is developed), and thus to the data inputs to the corresponding DC005 interface chip. The transmit enable signal for this DC005 chip is CIBT XMT RPLY H (not CIBR QBUS XMT H, as for the other DC005 chips). Figure 3-7 shows the development of this signal.

When the VAX-11/780 CPU reads or writes one of the two CIB data buffer registers, clearing the corresponding READY or DONE bit, the interrupt logic generates a transmit interrupt or receive interrupt signal to the LSI-11 on the DC003 interface chip, as shown in Figure 3-25, if the corresponding interrupt enable bit is set.

Table 3-3 shows the functions of the five interrupt related registers.

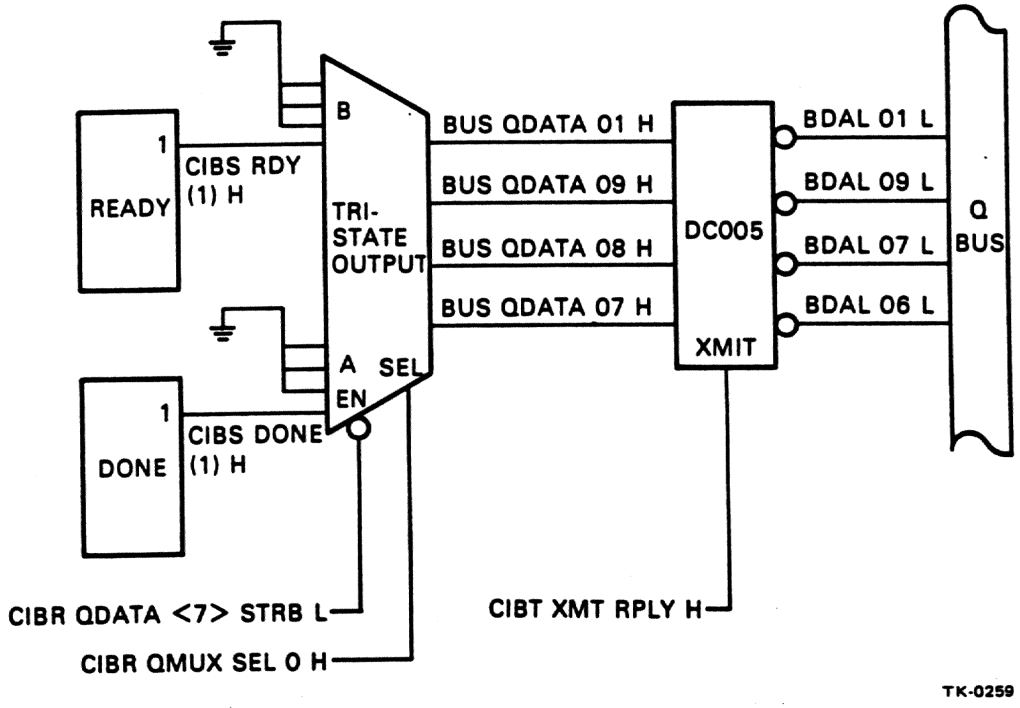


Figure 3-24 Ready and Done Multiplexer Logic

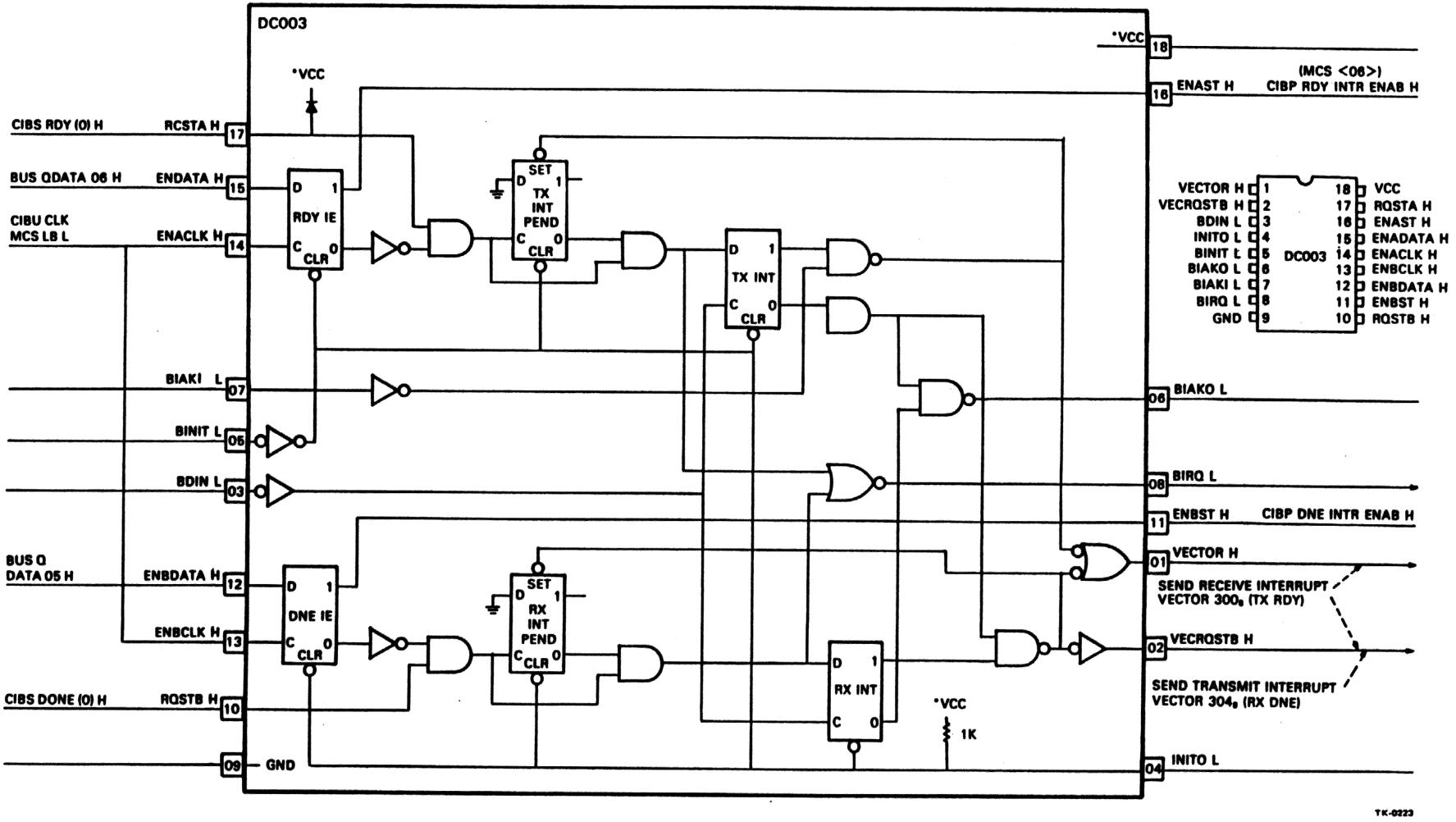


Figure 3-25 LSI-11 Interrupt Logic

Table 3-3 Interrupt Related Register Bit Functions

Register Name	Ready and Done Bits	Interrupt Enable Bits	Comments
RXCS (ID bus 04_{16})	RX DNE-R/O (bit 7)	RXIE R/W (bit 6)	If both bits are set, interrupt the VAX-11.
	same flop-flop		
RX DONE (Q bus $163014_8/173014_8$)	RX DNE-R/W (bit 7)		If this bit is reset and MCS <05> is set, interrupt the LSI-11.
TX CS (ID bus 06_{16})	TX RDY-R/O (bit 7)	TXIE R/W (bit 6)	If both bits are set, interrupt the VAX-11.
	same flip-flop		
TX RDY (Q bus $163016_8/173016_{16}$)	TX READY-R/W (bit 7)		If this bit is reset and MCS <06> is set, interrupt the LSI-11.
MCS (Q bus $163034_8/173034_8$)	RDY IE-R/W (bit 6)	DNE IE-R/W (bit 5)	Enable or disable interrupts to the LSI-11. These bits are implemented on the DC003 chip.

3.6 TYPICAL CIB TRANSFER OPERATIONS

When the VAX-11/780 software and the LSI-11 transfer data and control signals via the Console Interface Board, the transfer operations occur in predictable and logical sequences. For example, when the VAX-11/780 CPU is running macrocode and the console terminal operator types "set terminal program," switching from the console I/O mode to the program I/O mode, the following sequence will occur.

1. The LSI-11 sets the READY bit (7) in the CIB TX READY register, informing the VAX-11/780 that the console terminal is ready to accept a character (Figure 3-26).

2. The VAX-11/780 writes to the FM ID register, clearing the READY bit in the TX READY register, interrupting the LSI-11 (Figure 3-27).

3. The LSI-11 and CIB hold an interrupt dialogue on the Q bus (Figure 3-28).

4. An LSI-11 interrupt routine reads the character in the FM ID register and transfers the character to the console terminal (Figure 3-29).

5. The console terminal completes printing the character and interrupts the LSI-11 (not shown).

6. The LSI-11 sets READY in the TX READY register, thus completing one transfer (not shown).

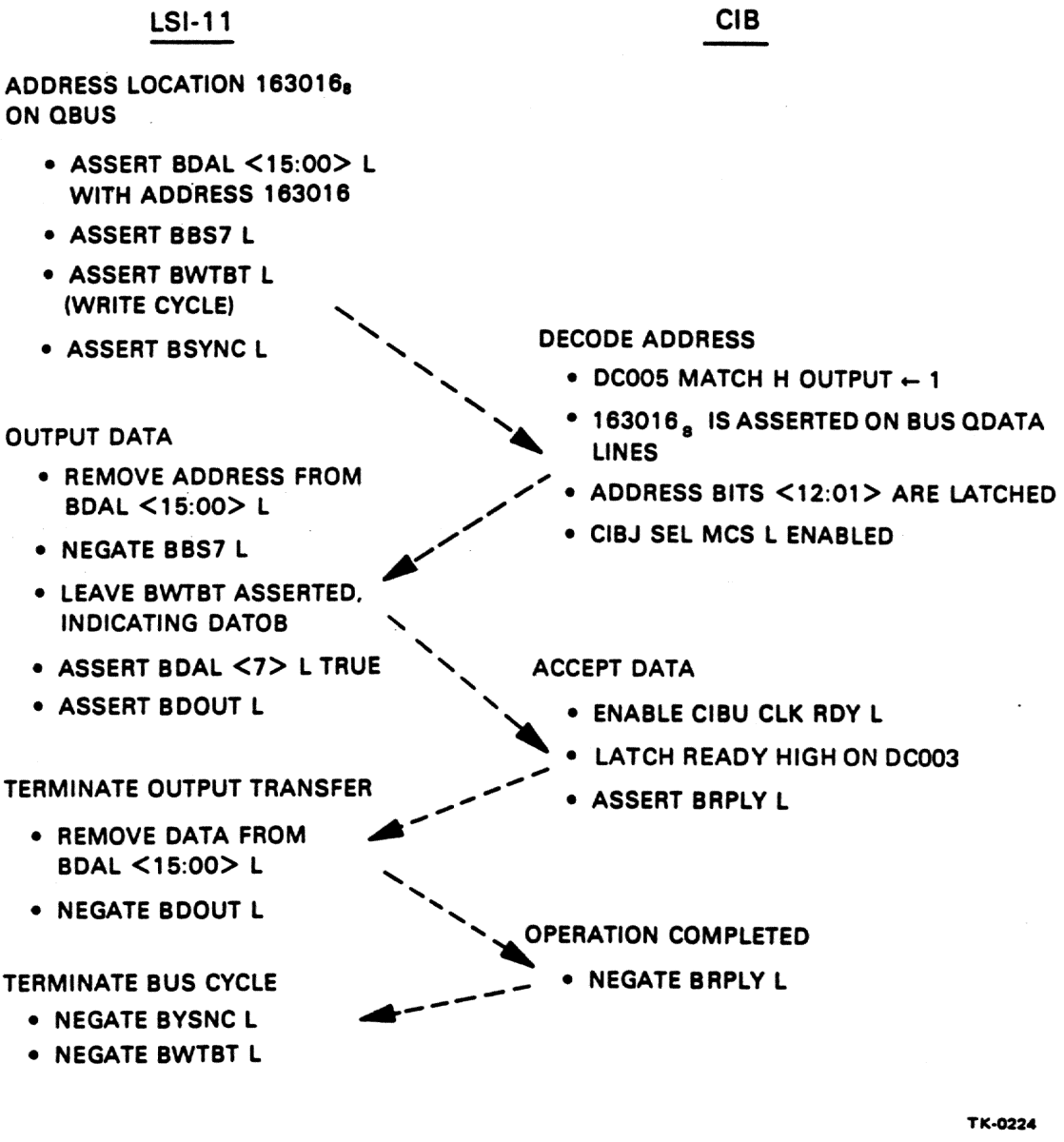


Figure 3-26 LSI-11 Sets TX READY, DATOB Transfer on Q Bus

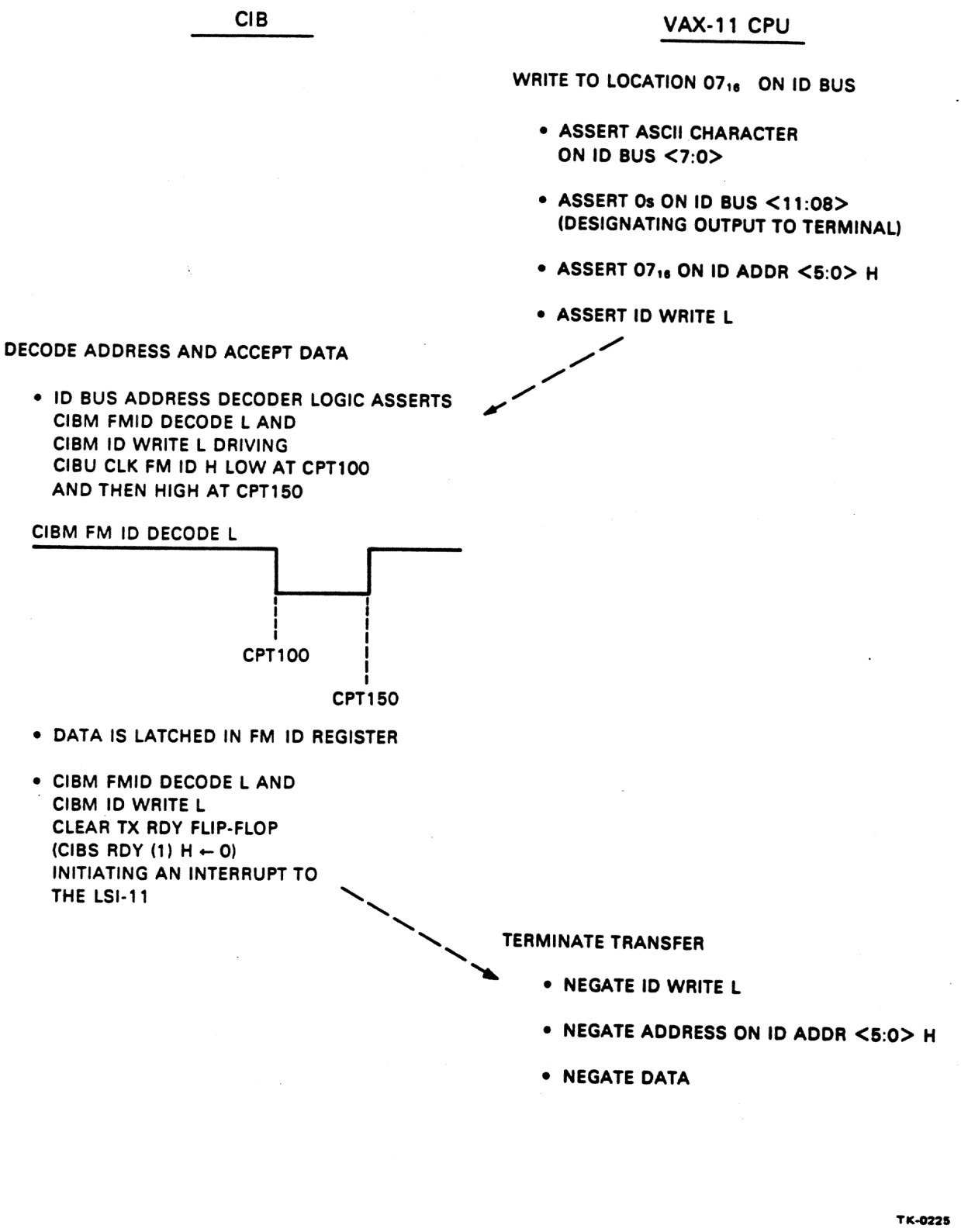


Figure 3-27 VAX-11/780 Software Sends a Character to the FM ID Register for Transmission to the Console Terminal, Write Transfer on the ID Bus

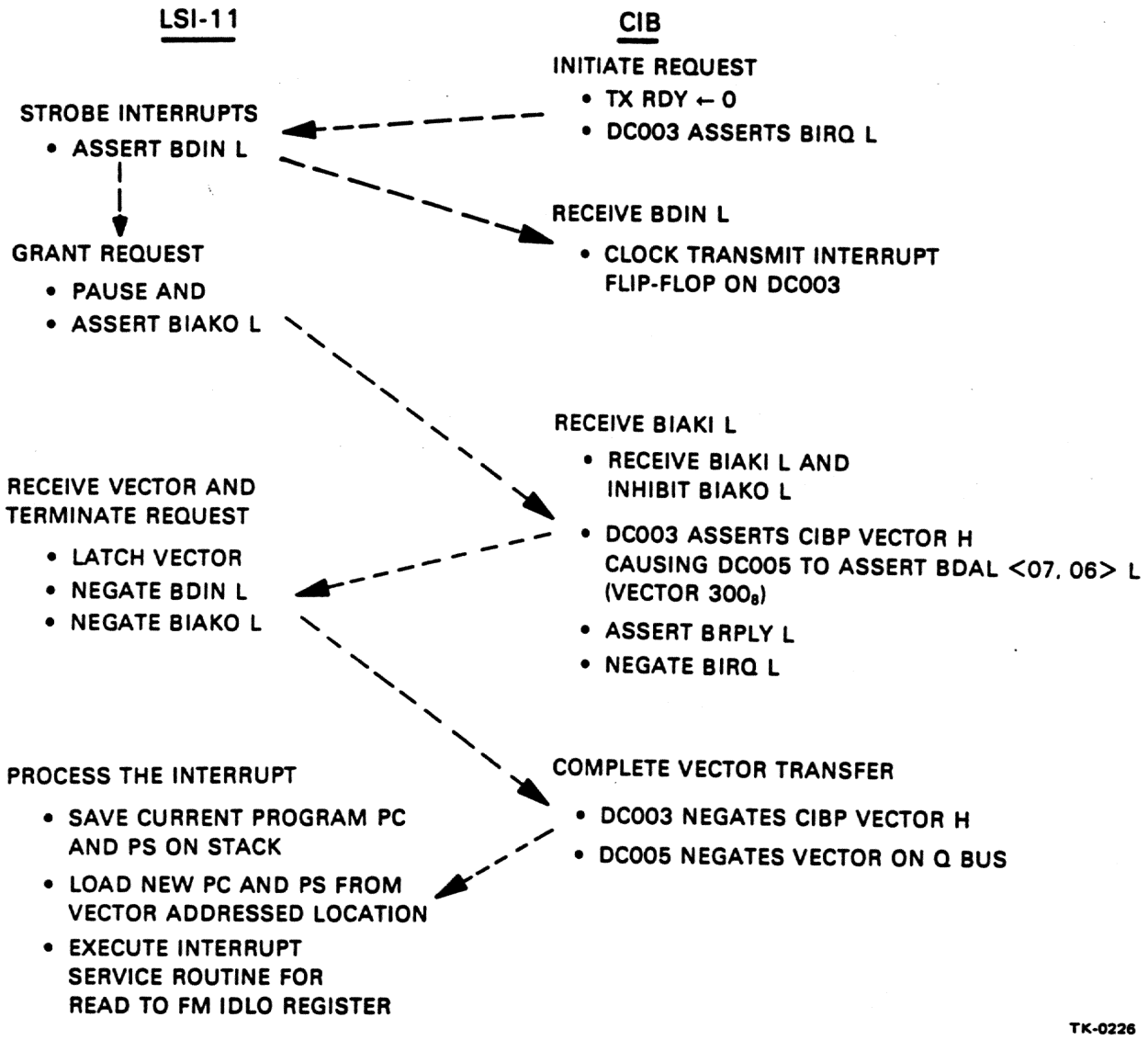


Figure 3-28 Interrupt Dialogue on the Q Bus

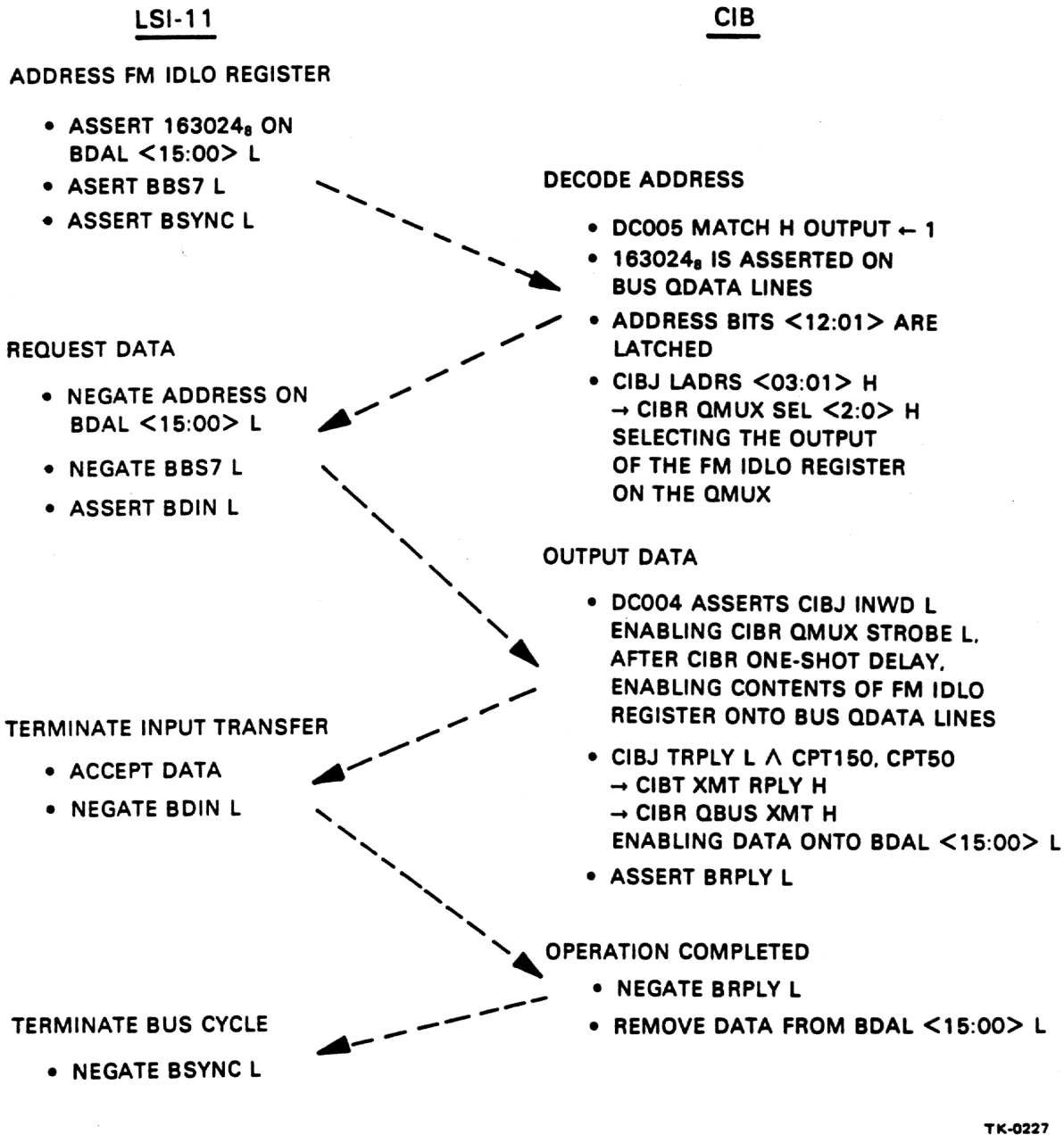


Figure 3-29 LSI-11 Interrupt Subroutine Reads the FM IDLO Register, DATI Transfer on Q Bus

Some time later, another sequence might occur, as follows:

1. LSI-11 writes to the TO IDLO register (Figure 3-30).
2. LSI-11 sets DONE (not shown).
3. VAX-11/780 CPU reads the TO ID register (Figure 3-31).

The flowcharts which follow trace the operations basic to each transfer.

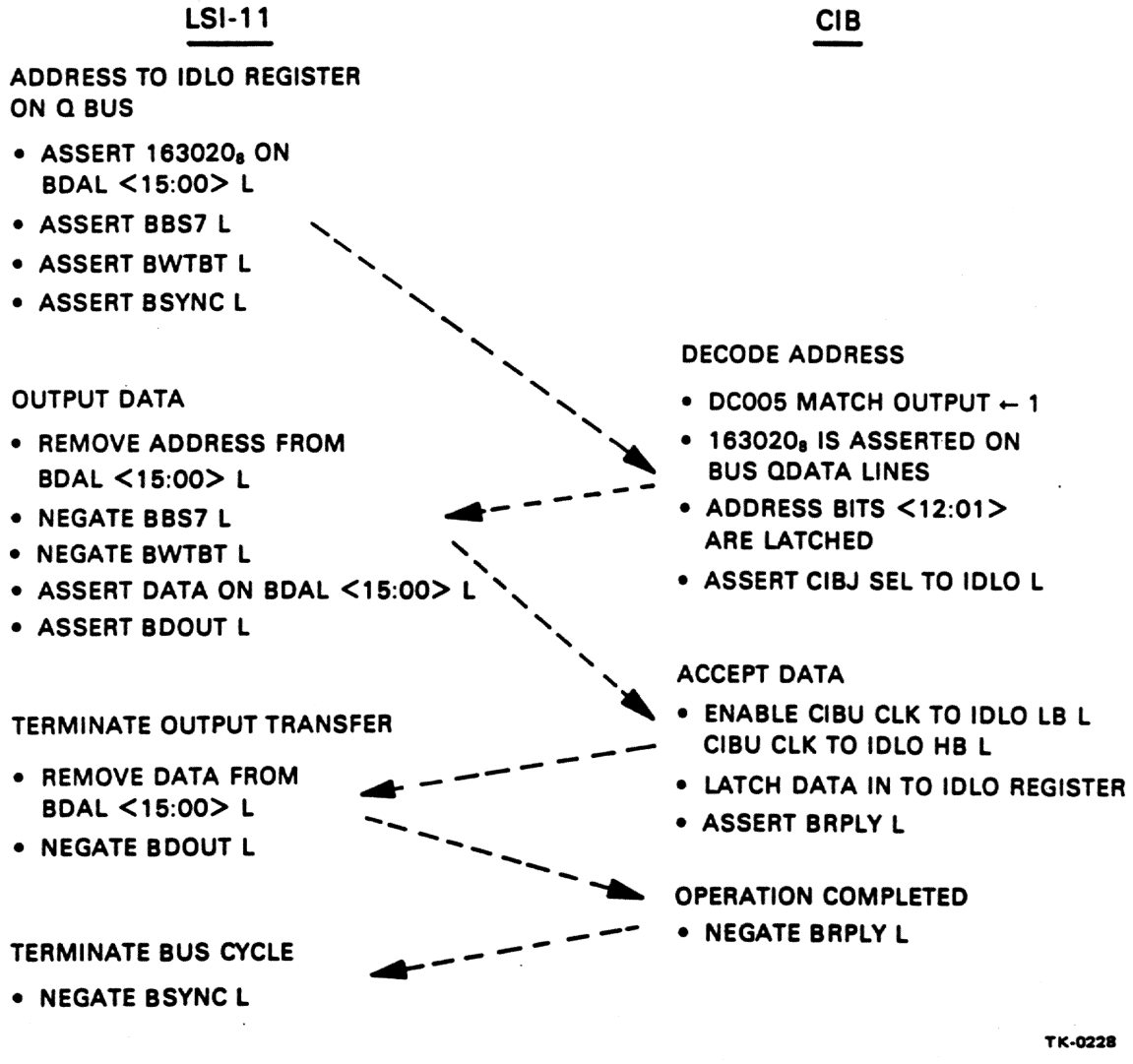


Figure 3-30 LSI-11 Writes a Character to the TO IDLO Register for Transmission to the VAX-11/780 Software, DATO Transfer on the Q Bus

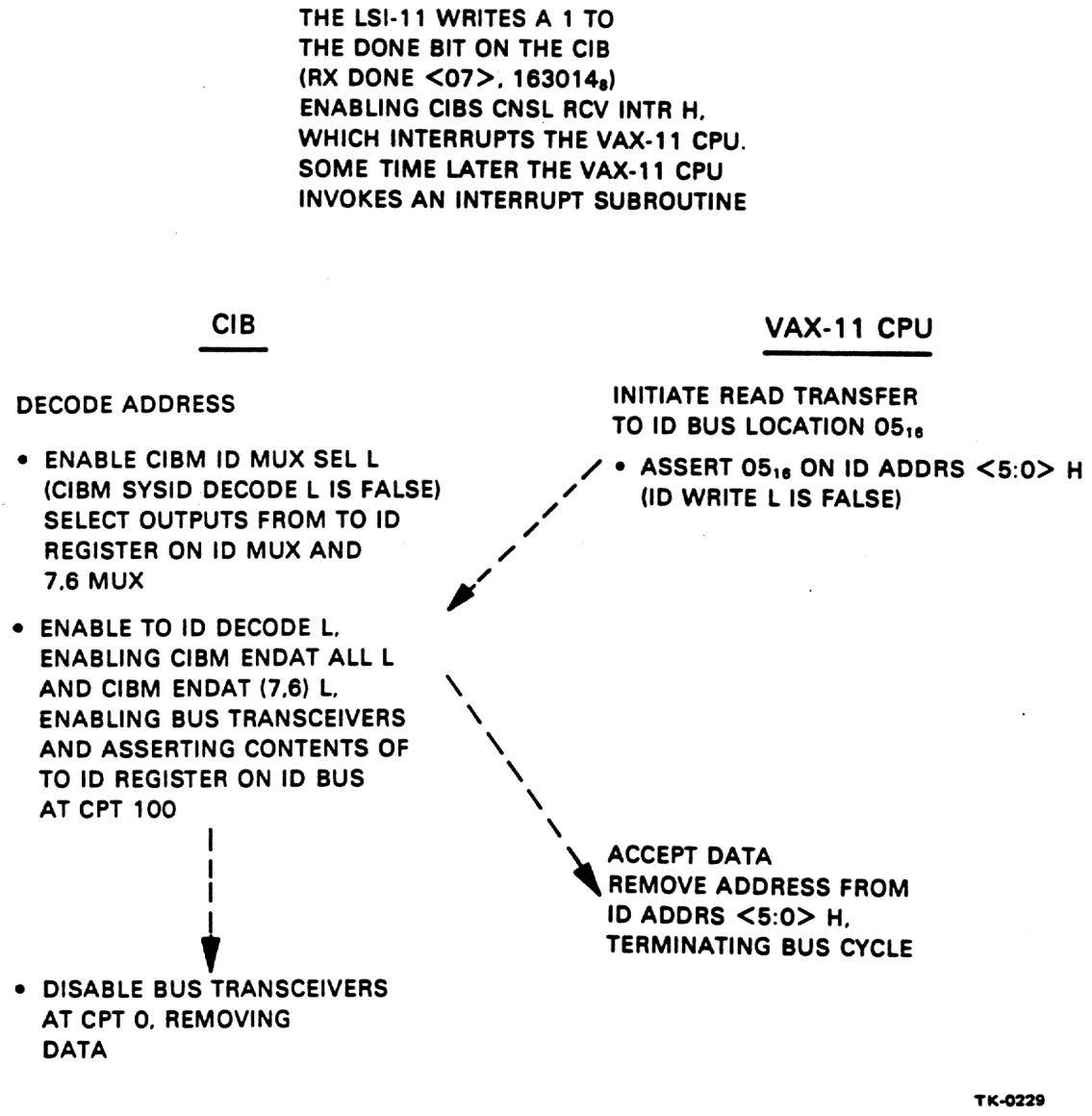


Figure 3-31 VAX-11/780 Interrupt Subroutine Reads the TO ID Register, Read Transfer on the ID Bus

APPENDIX A
Q BUS TECHNICAL DESCRIPTION

A.1 GENERAL

The Q bus is the I/O and memory bus for the LSI-11 and the VAX-11/780 console subsystem. The Q bus multiplexes address and data signals on the 16 BDAL lines. In addition individual control signals are used to sequence I/O transfers and interrupt transfers. Table A-1 lists the important Q bus signal lines.

Table A-1 Q Bus Signals

I/O Transfer Control Signals

Name	Description
BSYNC L	Synchronize – The bus master (LSI-11 processor) asserts BSYNC L to indicate that it has placed an address on BDAL <15:00> L. The transfer is in progress until BSYNC L is negated.
BDIN L	Data Input – The LSI-11 asserts BDIN L for two types of operations: 1. When it is asserted during BSYNC L time, BDIN L specifies an input transfer with respect to the processor. It requires BRPLY L as a response. The processor asserts BDIN L when it is ready to accept data from the slave device. 2. When the processor asserts BDIN L without BSYNC L, it is requesting an interrupt vector from an interrupting device.
BDOUT L	Data Output – When the LSI-11 processor asserts BDOUT L, valid data is on the bus for an output transfer from the processor to an I/O slave device. The slave device deskews BDOUT L (pauses) before latching the data. The slave device responding to the BDOUT L signal must assert BRPLY L to complete the transfer.

Table A-1 Q Bus Signals (Cont)

Name	Description
	I/O Transfer Control Signals
BWTBT L	Write/Byte – The LSI-11 processor uses BWTBT L to control bus cycles in two ways: 1. The processor asserts BWTBT L on the leading edge of BSYNC L to indicate that an output sequence (DATO or DATOB) is to follow. 2. The processor asserts BWTBT L together with BDOUT L, on a DATOB cycle, for byte addressing.
BRPLY L	Reply – A slave device asserts BRPLY L in response to BDIN L and BDOUT L on data transfers and in response to BIAKO L during interrupt transfers. BRPLY indicates that the slave has asserted input data on the bus, accepted output data from the bus, or asserted an interrupt vector on the bus.
	Interrupt Control Signals
BIRQ L	Interrupt Request – A device asserts this signal when its interrupt enable and interrupt request flip-flops are set. BIRQ L informs the processor that a device has data to send to the processor (input) or that the device is ready to accept output data from the processor. If the processor's PS word bit 7 is 0, the processor responds by acknowledging the request, asserting BDIN L and BIAKO L.
BIAKO L and BIAKI L	Interrupt Acknowledge Output and Interrupt Acknowledge Input – The processor asserts this signal in response to an interrupt request (BIRQ L). The processor asserts BIAKO L which is routed via the Q bus to the BIAKI L pin of the first device on the bus. If this device is requesting an interrupt (asserting BIRQ L), it will block the passing of BIAKO L to the next device and then place the interrupt vector on the bus. At the same time the device will negate BIRQ L and assert BRPLY L. If the device is not asserting BIRQ L, it passes BIAKI L to the next device via its own BIAKO L pin and the BIAKI L pin of the lower priority device.
	Address and Data Signals
BDAL <15:00> L	These 16 lines form the data/address path. Address information is first placed on the bus by the bus master (processor). The processor then either receives input data from or transmits output data to the addressed slave device or memory location over the same 16 bus lines.
BBS7 L	Bank 7 Select – The bus master asserts BBS7 L when an address in the upper 4K bank (address in the 28K–32K range) is placed on the bus. BSYNC L is then asserted, and BBS7 L remains active for the duration of the addressing portion of the bus cycle.

Table A-1 Q Bus Signals (Cont)

Initialization, Power Fail Signals	
Name	**Description**
BPOK H	Power OK – The power supply asserts this signal when primary power is normal. If BPOK H is negated during processor operation, the processor initiates a power fail trap sequence.
BDCOK H	DC Power OK – The power supply asserts this signal when there is sufficient dc voltage available to sustain reliable system operation.
BINIT L	Initialize – The processor asserts BINIT L to initialize or clear all devices connected to the Q bus. The signal is generated in response to a power up condition (the negated condition of BDCOK H).
	Halt and Refresh Signals
BHALT L	Processor Halt – When BHALT L is asserted, the processor responds by halting normal program execution. External interrupts are ignored, but memory refresh interrupts are enabled if W4 on the processor module is removed. When the processor is in the halt state, it executes the ODT microcode, invoking console device (terminal) operation.
BREF L	Memory Refresh – This signal can be asserted by a processor microcode-generated refresh interrupt sequence (when enabled) or by an external device. BREF L forces all dynamic MOS memory units to be activated for each BSYNC L/BDIN L bus transaction.

A.2 TRANSFER OPERATIONS ON THE Q BUS

A master-slave relationship defines communication between the processor and the other devices on the Q bus. Each control signal issued by a master device must be acknowledged by a slave device in order to complete a transfer.

Every processor instruction requires one or more I/O operations. The first operation required is a data input transfer (DATI), which fetches an instruction from the location addressed by the program counter (PC or R7). This operation is called a DATI bus cycle. If no additional operands are referenced in memory or in an I/O device, no additional bus cycles are required for instruction execution. However, if memory or a device is referenced, additional DATI, data input/output (DATIO or DATIOB), or data output transfer (DATO or DATOB) bus cycles are required. In addition, the processor can service interrupt requests only prior to an instruction fetch (DATI bus cycle) and if the processor's priority is zero. (PS word bit 7 is 0.)

The following paragraphs describe the types of bus cycles. Note that the sequences for I/O operations between processor and memory or between processor and I/O device are identical. DATO (or DATOB) cycles are equivalent to write operations, and DATI cycles are equivalent to read operations. In addition, DATIO cycles include an input transfer followed by an output transfer. The DATIO cycle provides an efficient means of executing an equivalent read-modify-write operation by making it unnecessary to assert an address a second time.

A.2.1 Input Operations

The sequence for a DATI operation is shown in Figure A-1. DATI cycles are asynchronous and require a response from the addressed device or memory. The addressed memory or device responds to the input request (BDIN L) by asserting BRPLY L. If BRPLY is not asserted within 10 μs (max) after BDIN L is asserted, the processor terminates the cycle and traps through location 4.

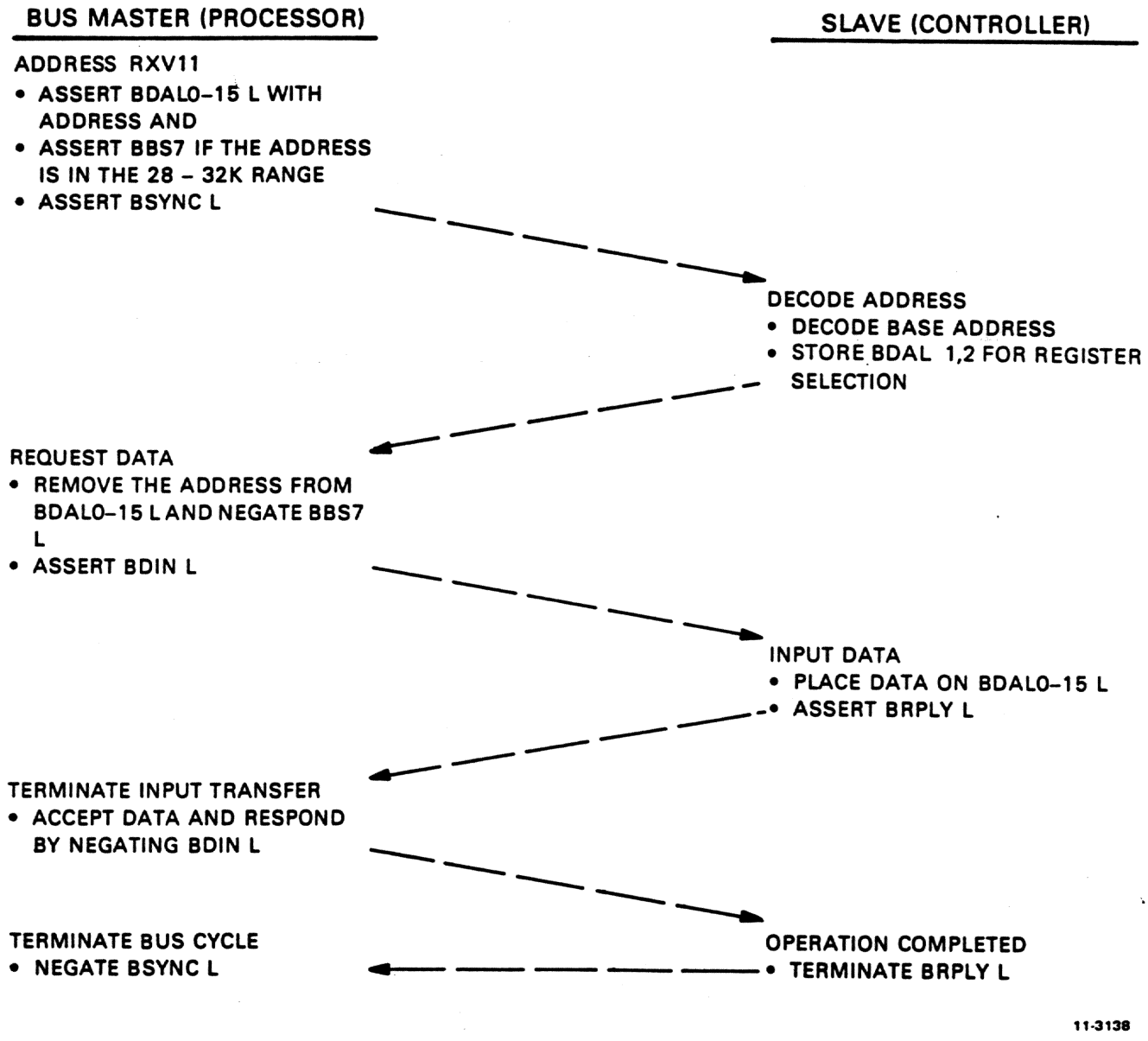


Figure A-1 DATI Bus Cycle

Note that BWTBT L is not asserted during the address time, indicating that an input data transfer is to be executed.

A DATIO cycle is equivalent to a read-modify-write operation. An addressing operation and an input word transfer are first executed in a manner similar to the DATI cycle; however, BSYNC L remains in the active state after completing the input data transfer. This causes the addressed device or memory to remain selected, and an output data transfer follows without any further addressing. After completing the output transfer, the master (processor) terminates BYSNC L, completing the DATIO cycle. The actual sequence required for a DATIO cycle is shown in Figure A-2. Note that the output data transfer portion of the bus cycle can be a byte transfer; hence, this cycle is shown as DATIOB.

A.2.2 Output Operations

The sequence required for a DATO or the equivalent output byte (DATOB) bus cycle is shown in Figure A-3.

As on the input operations, failure to receive BRPLY L within 10 μs after asserting BDOUT L is an error and results in a processor time-out trap through location 4.

Note that BWTBT L is asserted during the addressing portion of the cycle to indicate that an output data transfer is to follow. If a DATOB is to be executed, BWTBT L remains active for the duration of the bus cycle; however, if a DATO (word transfer) is to be executed, BWTBT L is negated during the remainder of the cycle.

A.2.3 Interrupts

Interrupts are requests made by peripheral devices which cause the processor to temporarily suspend its present (background) program execution to service the requesting device. Each device which is capable of requesting an interrupt has a service routine which is automatically entered when the processor acknowledges the interrupt request. After completing the service routine execution, program control is returned to the interrupted program.

A device can interrupt the processor only when interrupts are enabled. The processor's priority in the PS word is 4 when external interrupts are disabled and 0 when external interrupts are enabled. Device priority is highest for devices electrically closest to the processor along the bus.

Any device that can interrupt the processor can also interrupt the service routine execution of a lower priority device if the processor's priority is 0 during that execution; hence, interrupt nesting to any level is possible with this interrupt structure. Each device normally contains a control status register (CSR), which includes an interrupt enable bit. A program must set this bit before an interrupt request can actually be made by a device.

An interrupt vector associated with each device is hard-wired into the device's interface/control logic. This vector is an address pointer that allows automatic entry into the service routine without device polling.

When an interrupt request is issued via the external event signal line, the processor automatically services the request via location 100_8; it does not input a vector address as done for other external interrupt devices.

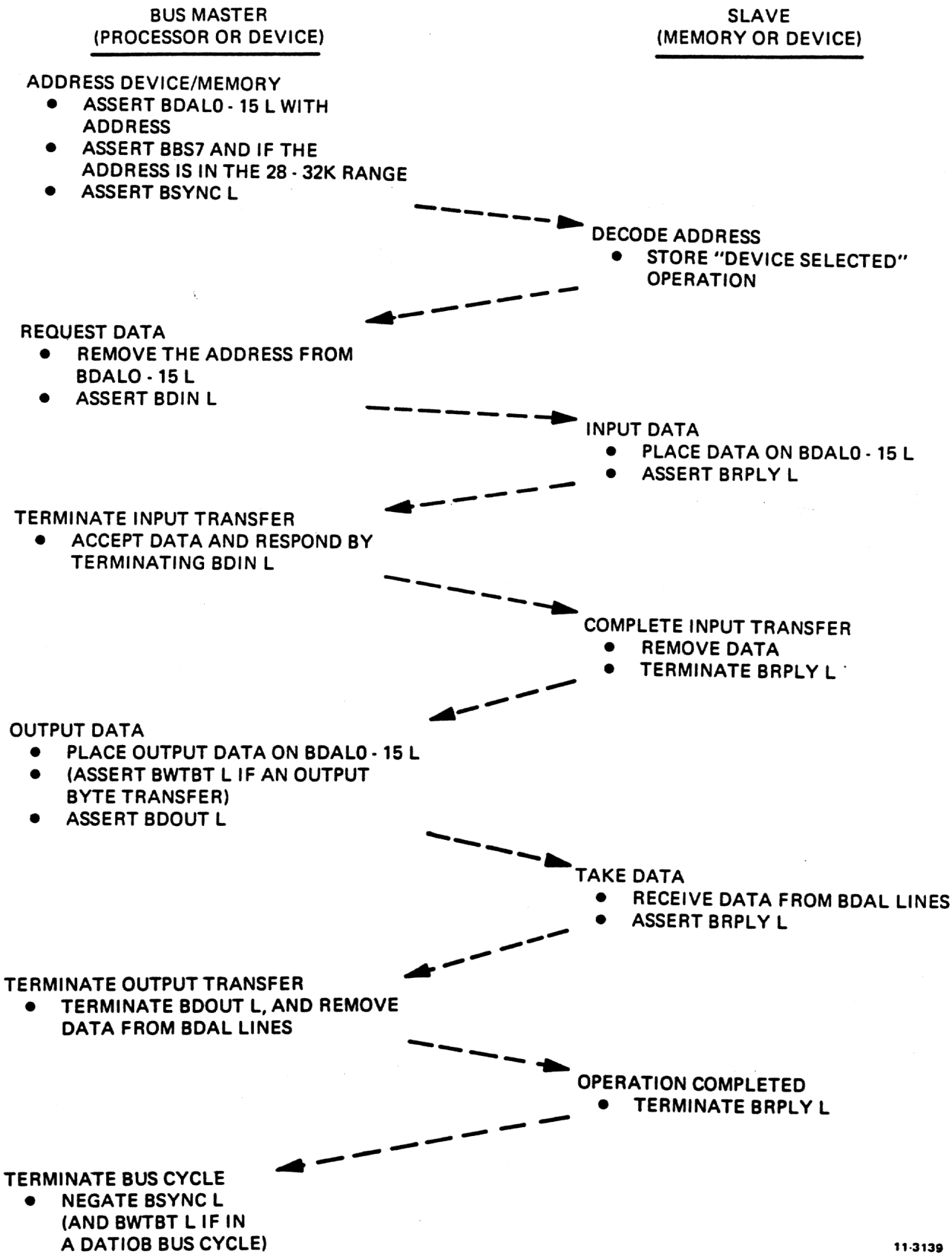


Figure A-2 DATIO or DATIOB Bus Cycle

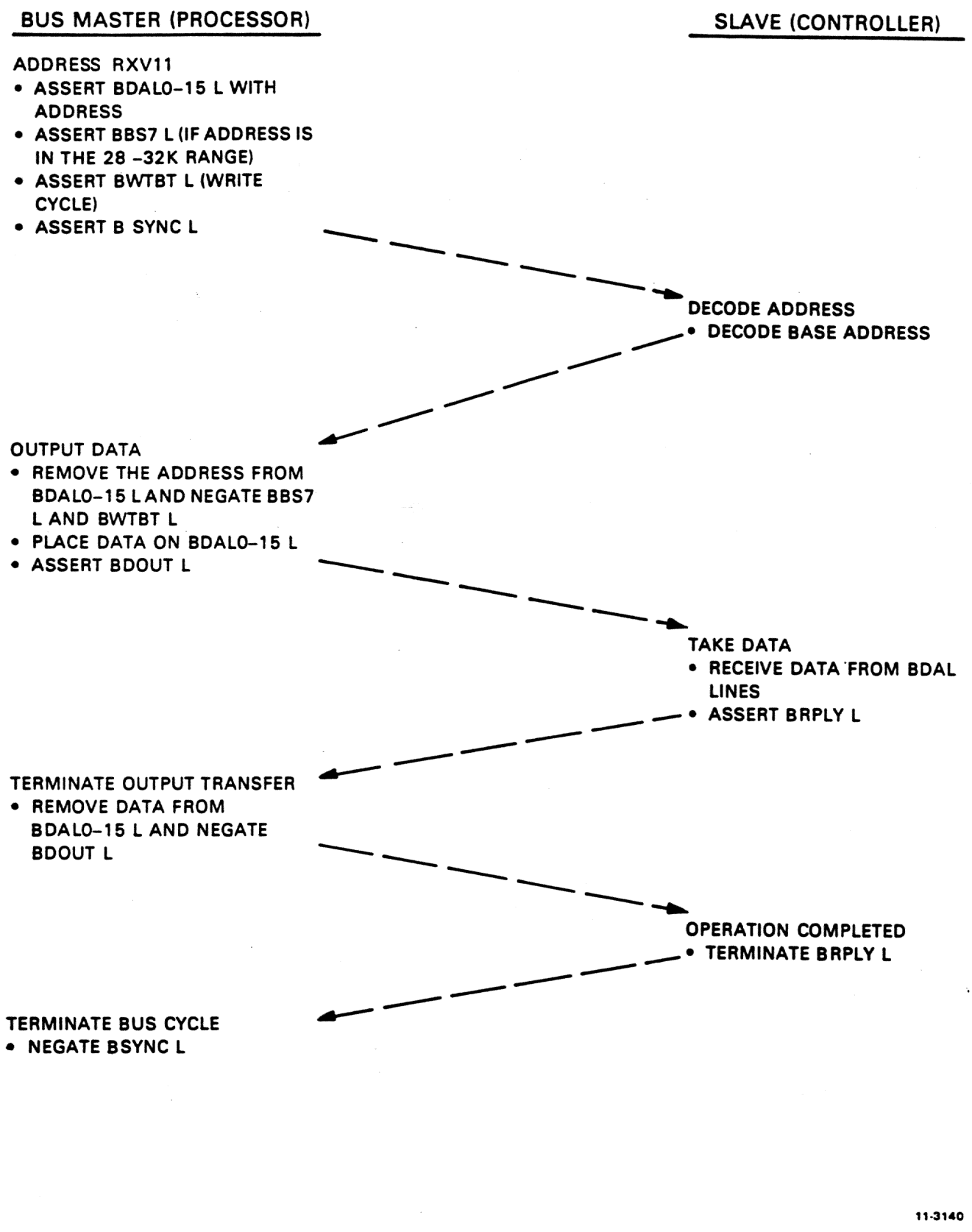


Figure A-3 DATO or DATOB Bus Cycle

The interface control and data signal sequence required for interrupts is shown in Figure A-4. A device requests interrupt service by asserting BIRQ L. The processor can acknowledge interrupt requests only between instruction executions. It does this by generating an active (low) BDIN L signal, enabling the device's vector response. The processor then asserts the BIAKO L signal. The first device on the bus receives this daisy-chained signal at its BIAKI L input. If it is not requesting service, it passes the signal via its BIAKO L output to the next device, and so on, until the requesting device receives the signal. The device that does not pass the BIAKO L signal responds by asserting BRPLY L (low) and placing its interrupt vector on data/address bus lines BDAL0–15 L. Automatic entry to the service routine is then executed by the processor as previously described.

A.2.4 Bus Initialization

Devices along the I/O bus are initialized whenever the system dc voltages are cycled on or off, or when a RESET instruction is executed. Initialization during the power-on/power-off sequence is described in Paragraph A.2.5. When the RESET instruction is executed, the processor responds by asserting BINIT L for approximately 10 μs. Devices along the bus respond to the BINIT L signal, as appropriate, by clearing registers and presetting or clearing flip-flops.

A.2.5 Power-Up/Power-Down Sequence

Power status signals BPOK H and BDCOK H must be asserted or negated in a particular sequence as dc operating power is applied or removed. Initially, BDCOK H and BPOK H are passive (low). As dc voltages rise to operating levels, BINIT L is asserted by the processor module. Approximately 3 ms (min) after +5 V and +12 V power are normal, an external signal source, or the H780 power supply in PDP-11/03 systems, produces an active BDCOK H signal; the processor responds by negating BINIT L, and waits for BPOK H. The BPOK H signal, produced by an external signal source or the H780 power supply, goes true (high) 70 ms (min) after BDCOK H goes high. The processor responds by executing the user-selected power-up routine; if BHALT L is asserted, the console microcode is executed.

During a power-down sequence, the external signal source first negates BPOK H, causing the processor to execute the power-fail trap (PC at 024_8, PS at 026_8).

Approximately 3 ms (max) later, the processor initializes the bus by asserting BINIT L in response to the external signal negation of BDCOK H.

A.2.6 Memory Refresh Operation

A complete refresh operation requires 64 BSYNC/BDIN transactions which must be completed every 2 ms. The processor (or other device controlling the refresh operation) first asserts BREF L for each BSYNC/BDIN transaction during the addressing portion of each refresh operation. BREF L causes all dynamic MOS memory devices to be simultaneously enabled and addressed, overriding local bank selection circuits. Refresh is then accomplished by executing 64 BSYNC/BDIN transactions, in a manner similar to the DATI bus cycle, incrementing the "row" address (bits 1–6) once for each transaction. Address bit 0 is not significant in the refresh operation. When refresh is controlled by processor microcode, the operation takes approximately 130 μs.

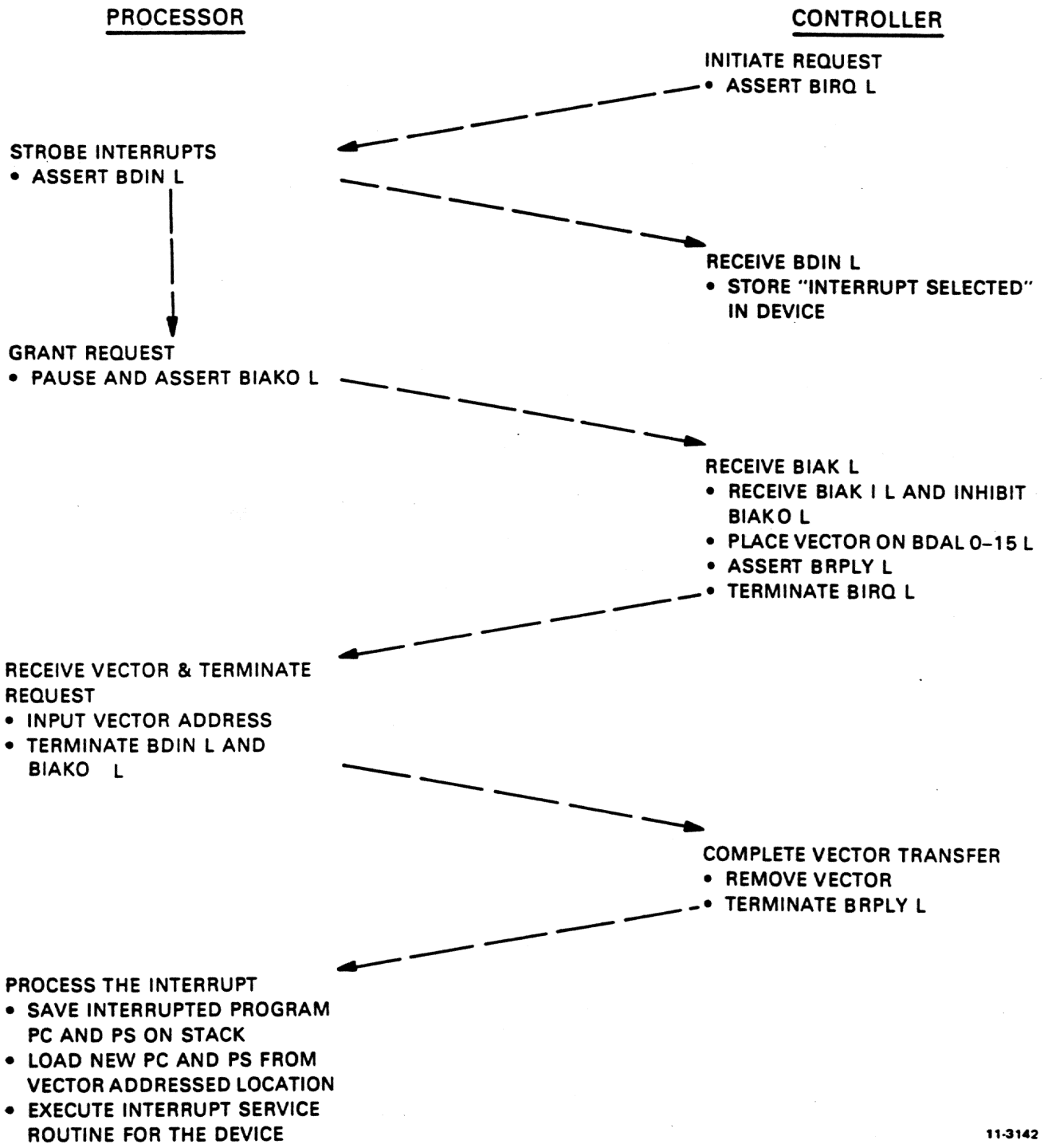


Figure A-4 Interrupt Request/Acknowledge Sequence

VAX-11/780 CONSOLE INTERFACE BOARD
TECHNICAL DESCRIPTION
EK-KC780-TD-001

Reader's Comments

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 or errors have you found in the manual? _____

_____

_____

_____

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

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

_____

_____

_____

☐ Please send me the current copy of the *Technical Documentation Catalog*, which contains information on the remainder of DIGITAL's technical documentation.

Name _____ Street _____
Title _____ City _____
Company _____ State/Country _____
Department _____ Zip _____

Additional copies of this document are available from:

Digital Equipment Corporation
444 Whitney Street
Northboro, Ma 01532
Attention: Communications Services (NR2/M15)
Customer Services Section

Order No. EK-KC780-TD-001

— — — — — — — — — — — — Fold Here — — — — — — — — — — —

— — — — — — — — — Do Not Tear - Fold Here and Staple — — — — — — — —

FIRST CLASS
PERMIT NO. 33
MAYNARD, MASS.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

Postage will be paid by:

Digital Equipment Corporation
Technical Documentation Department
Maynard, Massachusetts 01754