varian data machines /a varian subsidiary

VARIAN 70 SERIES
PROCESSOR MANUAL

VARIAN 70 SERIES

PROCESSOR MANUAL

See the addendum at the back of this manual.

varian data machines/a varian subsidiary
2722 michelson drive/irvine, california 92664

varian data machines

98 A 9906 023

JULY 1974

EN 82781

SECTION 4 THEORY OF OPERATION

TABLE OF CONTENTS

SECTION 1 GENERAL DESCRIPTION

SECTION 2 INSTALLATION

SECTION 3 OPERATION

SECTION 4 THEORY OF OPERATION *(continued)*

SECTION 5 MAINTENANCE

SECTION 6 MNEMONICS

LIST OF ILLUSTRATIONS

SECTION 4 THEORY OF OPERATION *(continued)*

LIST OF ILLUSTRATIONS *(continued)*

LIST OF ILLUSTRATIONS *(continued)*

LIST OF TABLES

SECTION 1
GENERAL DESCRIPTION

The **Varian 70 Series Processor Manual** describes the processor, I/O control, and control panel.

The manual is divided into six sections:

- Features and specifications of the processor, and related publications
- Installation and interconnection data
- Operation
- Theory of operation
- Maintenance
- Mnemonics list

Documents such as logic diagrams, schematics, and parts lists are supplied in a System Maintenance Manual. This manual is assembled when the equipment is shipped, and reflects the configuration of a specific system.

The following list contains the part numbers of other manuals pertinent to the Varian 70 series computers (the x at the end of each document number is the revision number and can be any digit 0 through 9):

72 System Handbook	98 A 9906 20x
73 System Handbook	98 A 9906 01x
74 System Handbook	98 A 9906 21x
Core Memory Manual	98 A 9906 03x
Semiconductor Memory Manual	98 A 9906 04x
Option Board Manual	98 A 9906 05x
Power Supply Manual	98 A 9906 06x
Microprogramming Guide	98 A 9906 07x
Writable Control Store Manual	98 A 9906 08x
Memory Map Manual	98 A 9906 10x
Test Programs Manual	98 A 9952 06x
VORTEX II Reference Manual	98 A 9952 24x

High-speed logic and use of a 60-nanosecond access-time control store permits the Varian 73 processor to execute a single microinstruction in 165 nanoseconds. A processor interface is provided that permits the internal control-store to be disabled by a writable control store allowing for special applications through user microprograms. Other features of the processor are:

- High performance through instruction pipelining
- Extendable architecture through microprogramming
- Independent operation of memory, I/O, and processor functions
- Multiple-bus structure
- 16 general-purpose registers
- Comprehensive set of shift, arithmetic, and logical functions

Table 1-1 lists the specifications of the processor.

Table 1-1. Processor Specifications

Parameter	Specification
Speed	165 nanoseconds (microinstruction execution time)
Word length	16 bits
Logic levels (internal)	High = +2.4 to +5.0V dc Low = −0.4 to +0.8V dc
Logic levels (I/O bus)	High = +2.8 to +3.6V dc Low = −0.4 to +0.8V dc
Power	Processor board: 5V dc at 12A Option board (with all options): 5V dc at 11A Control-panel board: 5V dc at 2.1A Termination shoes (on back of mainframe): 5V dc at 3.2A
Environment	0 to 50 degrees C, 10 to 90 percent relative humidity without condensation.
Forced-air cooling	For each 3W dc power, one cubic foot of air per minute applied above and below each circuit board.

SECTION 2
INSTALLATION

2.1 INSPECTION

The processor has been packed and inspected to ensure its arrival in good working order. To prevent damage, take care during unpacking and handling. Check the shipping list to ensure that all equipment has been received. Immediately after unpacking, inspect the equipment for shipping damage. Ascertain that wires and cables are neither loose nor broken, and that hardware is secure. If damage exists:

a. Notify the transportation company.

b. Notify Varian Data Machines.

c. Save all packing material.

2.2 PHYSICAL DESCRIPTION

The processor circuits are on the 15.6 by 19 inch processor board (p/n 44P0614), the I/O control circuits are on a portion of the 15.6 by 19 inch option board (p/n 44P0619), and the control-panel circuits are on the 6.5 by 17.5 inch control-panel board (p/n 44P0645). On the control-panel board, is mounted a switch assembly (p/n 44P0647) containing control-panel switches. The printed-circuit (PC) boards are illustrated in figures 2-1 and 2-2.

Figure 2-3 shows the locations of the processor, option, and control-panel boards in a typical mainframe installation. The control panel can be hinged open (figure 2-3) or completely removed from the mainframe as follows:

a. Pull out the control panel to the locked position (approximately 2 inches).

b. Lift up on the control panel a fraction of an inch. The panel can now be hinged down to its open position.

c. For complete removal of the control panel, unplug cables from connectors J1 and J2 on the control-panel board, press in the locking devices on sides of slides, and pull panel completely out of the mainframe.

d. To install the control panel back into the mainframe, insert slides into slide rails and slide the panel toward the mainframe (the slide locking devices must be pressed to move panel past the locked position).

2.3 INTERCONNECTION

Circuit-board connector functions are listed as follows:

a. Processor board

P1, memory and power
J2, writable control store
J3, writable control store
J4, option board
J5, I/O lines
J6, multiple processor and writable control store

b. Option board

P1, memory and power
J2, not used
J3, auxiliary I/O lines
J4, processor board
J5, I/O lines
J6, writable control store and processor
J7, priority memory access
J8, Teletype

c. Control-panel board

J1, I/O lines
J2, power
J3, switch assembly
J4, switch assembly

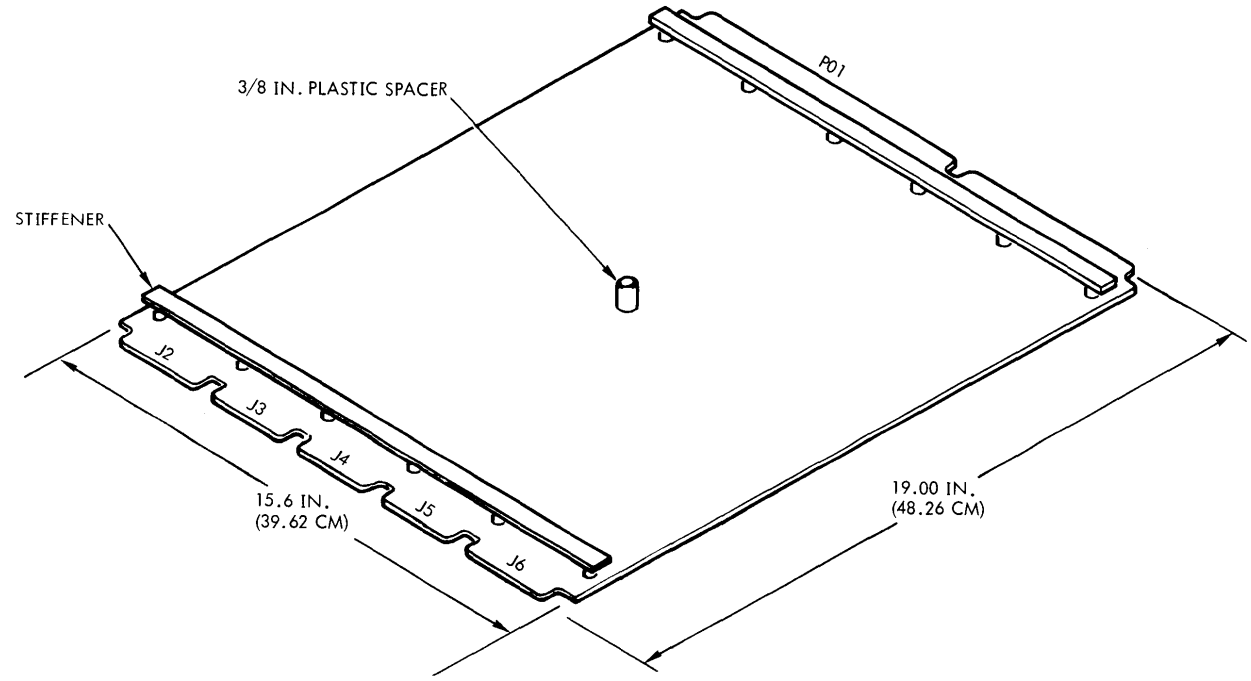
The pin assignments for these circuit-board connectors are given in the logic diagrams of each board (in System Maintenance Manual). Logic diagram part numbers are:

a. Processor logic diagram, p/n 91B0378.

b. Option logic diagram, p/n 91B0401.

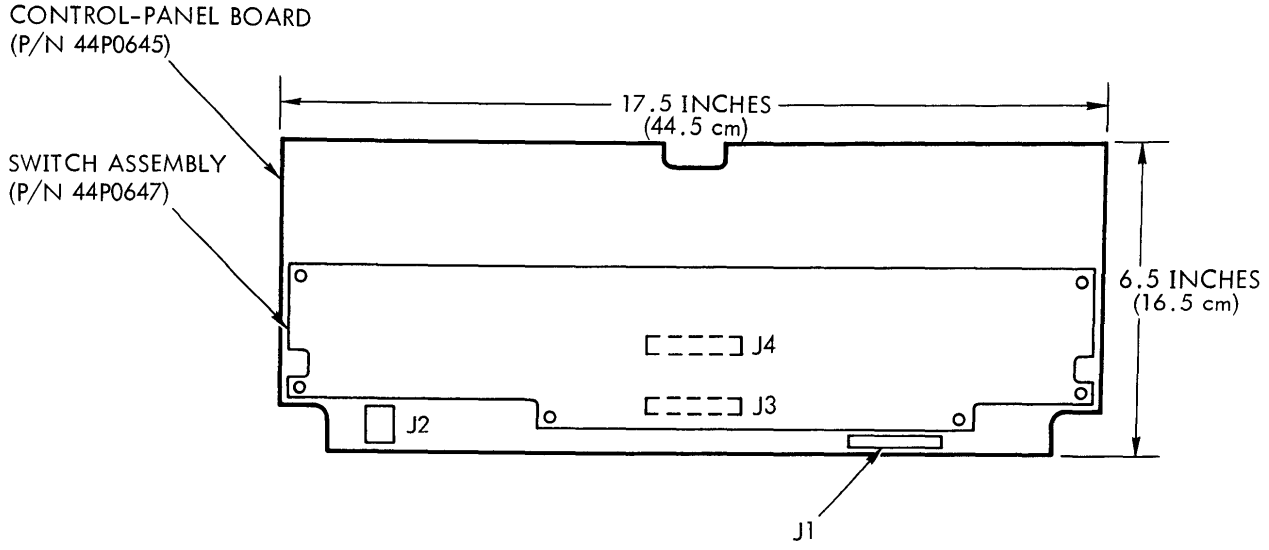
c. Control-panel logic diagram, p/n 91B0406.

Interconnection of the processor, option, and control-panel boards is shown in figure 2-4.



VTII-1478

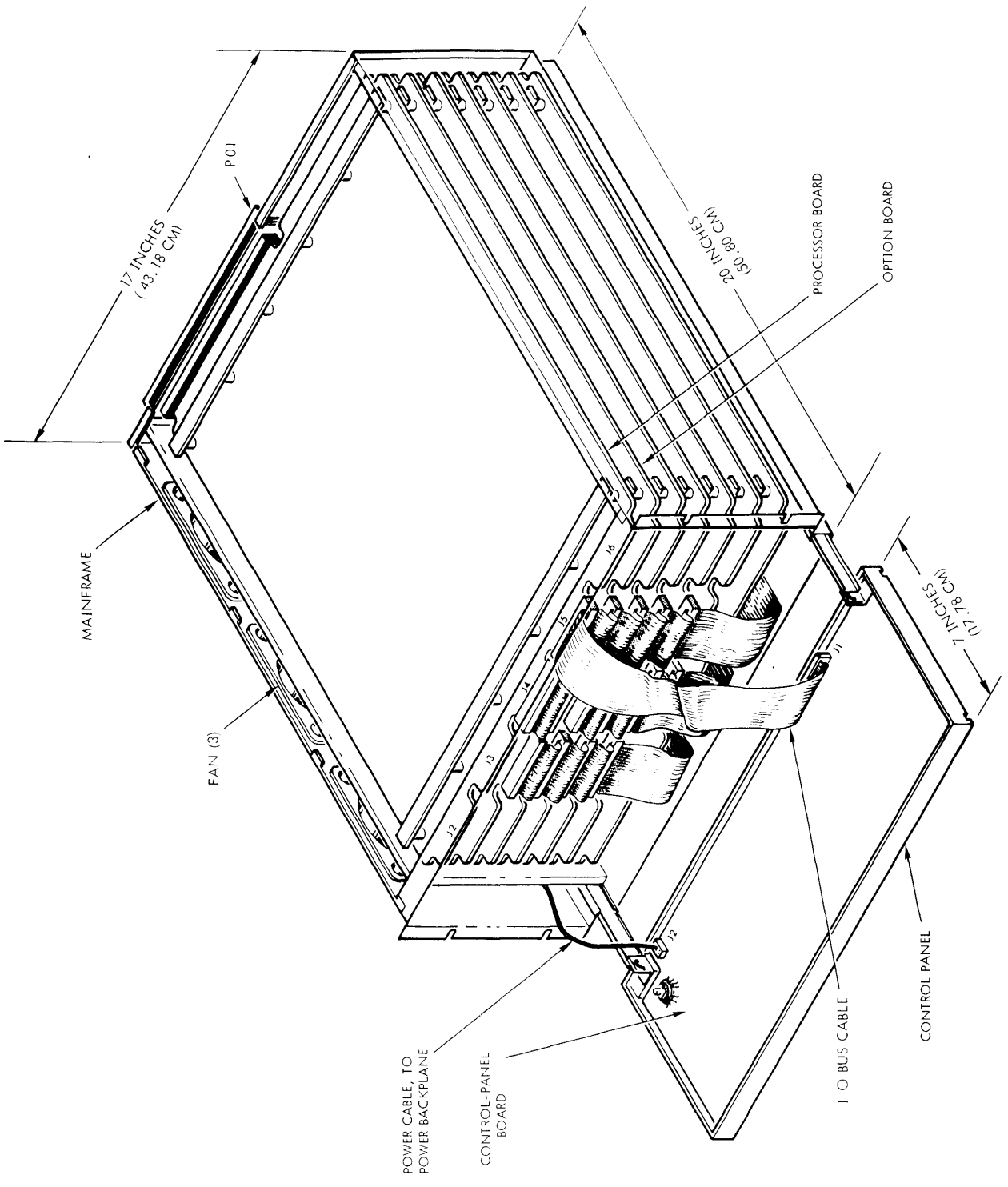
Figure 2-1. Processor or Option Board



NOTE: ELECTRICAL CONNECTIONS BETWEEN CONTROL-PANEL BOARD AND SWITCH ASSEMBLY ARE VIA J3 AND J4

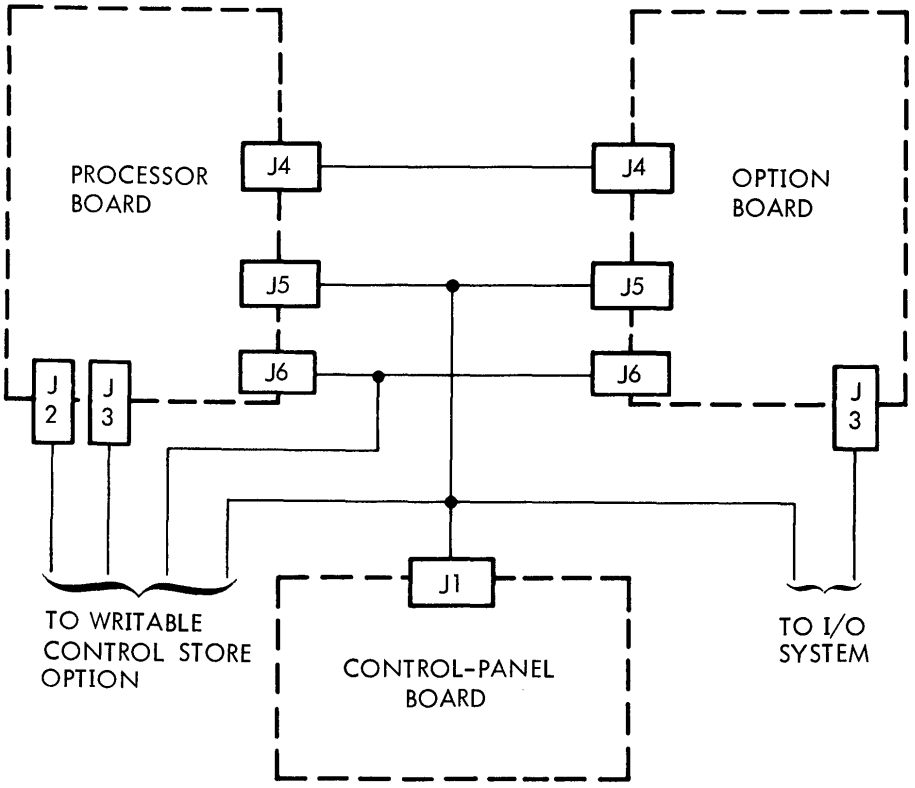
VTII-1670

Figure 2-2. Control Panel Board



VTII-1671

Figure 2-3. Typical Mainframe Installation (Seven-Inch Chassis)



VTII-1672

Figure 2-4. Interconnection Diagram

SECTION 3
OPERATION

This section describes the operation of the Varian 70 series computers.

3.1 SWITCHES AND INDICATORS

The Varian 74 control panel (figure 3-1) contains all of the switches and indicators needed for operating the system. Except for the POWER and BOOT SELECT switches, which are key-operated, all control-panel switches are pushbuttons. Control panels for the Varian 72 and 73 computers do not have the BOOT SELECT switch but otherwise have identical switches and indicators as shown in figure 3-1.

3.1.1 POWER Switch

The POWER switch is a key-operated, four-position switch that controls the ac line voltage to the computer power supply.

In the OFF position, the ac line voltage is removed from the input of the power supply and the optional data saver is disabled (if present). NOTE: When turning off power on systems with semiconductor memory, do not restore power for at least 30 seconds to ensure the refresher logic is operating.

In the HOLD position, the ac line voltage is applied to the power supply. All dc voltages are disabled except those required to maintain data in the semiconductor memory. Neither the computer nor the control panel is operational.

In the ON position, the ac line voltage is applied to the power supply. Both the computer and control panel are fully operational.

The CONSOLE DISABLE position is jumper-selectable to operate in two modes:

a. All control-panel pushbutton switches are disabled.

b. Only the STEP/RUN and RESET switches are disabled.

The jumper is factory-installed on the control-panel circuit board. With the POWER switch in the CONSOLE DISABLE position, the ac line voltage is applied to the power supply, the computer is operational, and the control-panel indicator lights are functional. The key can be removed from the POWER switch in any of the four positions.

To turn off the computer from the CONSOLE DISABLE position, turn the POWER switch ON, place the computer in the step mode (using STEP/RUN switch), and then turn the POWER switch to either HOLD (to maintain data in semiconductor memory) or OFF.

3.1.2 STEP/RUN Switch and STEP and RUN Indicators

The STEP/RUN switch is an alternate-action switch that switches the computer alternately to the step and run modes. In the step mode, the STEP indicator lights. In the run mode, the RUN indicator blinks until the START switch is pressed, at which time the RUN indicator comes on continuously.

When the computer is in the step mode, pressing the STEP/RUN switch places the computer in the run mode. The STEP indicator goes out and the RUN indicator blinks. When in the run mode, the computer is ready to be started (by pressing the START switch).

When the computer is in the run mode and has been started, pressing the STEP/RUN switch halts the computer after the current instruction has been executed and the next instruction fetched and loaded into the I register. The RUN indicator goes out and the STEP indicator lights. In addition, a halt instruction (after the computer has been started) halts the computer and causes the RUN indicator to blink.

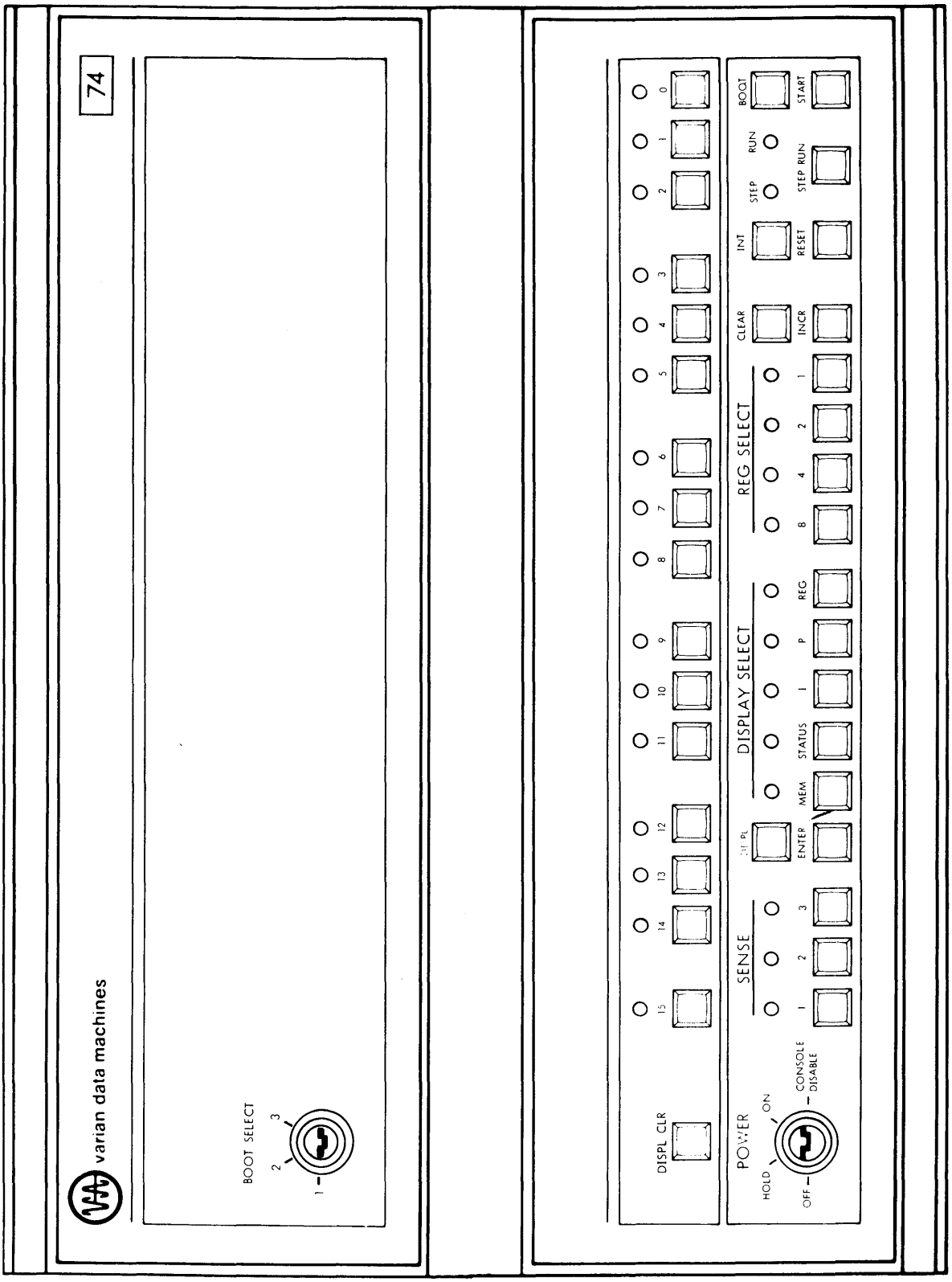
3.1.3 START Switch

When the computer is in the run mode but has not been started, pressing the START switch starts the program at the location specified by the contents of the program counter. The RUN indicator stops blinking and comes on continuously.

When the computer is in the step mode, pressing the START switch executes the instruction in the instruction register. Then it fetches the next instruction from the memory address specified by the contents of the program counter and loads it in the instruction register. The STEP indicator remains on.

3.1.4 BOOT Switch

The BOOT switch allows the bootstrap program to be loaded into the computer memory automatically. The bootstrap program enables the loading of the binary load/dump program into memory. When BOOT is pressed, the RUN indicator lights. Refer to the program execution portion of this section for bootstrap program loading procedures.



VTII-2127

Figure 3-1. Varian 74 Control Panel

3.1.5 Register-Entry Switches and Register-Display Indicators

The top row of control-panel lights comprises the 16 register-dislay indicators. They display the contents of the display register. This register, located on the control-panel circuit board, can be loaded from the register-entry switches on the control panel just below the 16 indicators. In addition, 16-bit data words can be loaded into the display register under control of the DISPLAY SELECT and REG SELECT switches (sections 3.1.7 and 3.1.9), allowing visual inspection of the contents of various registers and memory addresses.

Any of the sixteen bits can be set by pressing the corresponding register-entry switch. With a bit set, the corresponding display indicator lights. Pressing a register-entry switch for a bit already set has no effect. Bits can be reset only to zero by pressing the DISPL CLR switch (section 3.1.6).

For negative data, the sign bit (bit 15) is set (one).

3.1.6 DISPL CLR Switch

The contents of the display register can be cleared (reset to zero) by pressing the DISPL CLR switch. This turns off all sixteen display indicators.

3.1.7 DISPLAY SELECT Switches and Indicators

The five DISPLAY SELECT switches are used to select one of several registers for displaying its contents on the register display indicators and altering them from the register-entry switches. Pressing any DISPLAY SELECT switch cancels any previous selection, turns off the indicators for the previous selection, and lights the indicators for the new selection. Pressing DISPL CLEAR (section 3.1.6) clears the display register and turns off the display indicators.

3.1.7.1 MEM Switch

The MEM switch selects the memory for data entry or display. For entering data into memory and displaying the contents of memory refer to the manual operations (section 3.2).

3.1.7.2 STATUS Switch

The STATUS switch displays the status of various signals from the processor. To display the status of these processor signals:

a. Turn the POWER switch ON.

b. Place the computer in the step mode.

c. Press STATUS.

The register-display indicators now indicate the following:

Bit 15, Key register bit 15 (DCK15+)
Bit 14, Key register bit 14 (DCK14+)
Bit 13, Key register bit 13 (DCK13+)
Bit 12, Key register bit 12 (DCK12+)
Bit 11, Arithmetic and logic unit carry (DCNDC+)
Bit 10, Arithmetic and logic unit sign (DSGN+)
Bit 9, Arithmetic and logic unit output equals all ones (DEQ+)
Bit 8, Arithmetic and logic unit overflow (DOVF+)
Bit 7, Shift counter output bit 4 (DSC04+)
Bit 6, Shift counter output bit 3 (DSC03+)
Bit 5, Shift counter output bit 2 (DSC02+)
Bit 4, Shift counter output bit 1 (DSC01+)
Bit 3, Shift counter output bit 0 (DSC00+)
Bit 2, Arithmetic and logic unit output zero (DCNOZ+)
Bit 1, Supervisor mode (CESK+)
Bit 0, Not used

3.1.7.3 I Switch

The I switch selects the instruction (I) register for data display or entry. Pressing the I switch while the RUN indicator is off or blinking (step mode or halted) displays the contents of the instruction register on the register-display indicators. Changing the contents of the display register, by pressing the DISPLAY CLR switch and the register entry switches, automatically changes the contents of the instruction register. The instruction register contains the instruction to be executed next.

3.1.7.4 P Switch

The P switch selects the program (P) counter for data display or entry. Pressing the P switch while the RUN indicator is off or blinking (step mode or halted) displays the contents of the program counter on the register-display indicators. Changing the contents of the display register, by pressing the DISPLAY CLR and register-entry switches, automatically changes the contents of the program counter. The program counter contains the address of the next instruction to be fetched.

3.1.7.5 REG Switch

The REG switch enables one of the registers designated by the REG SELECT switches (section 3.1.9) to be selected for data display or entry.

3.1.8 DISPL and ENTER Switches

The DISPL switch is used with the MEM switch (section 3.1.7.1) for displaying memory data on the register-display indicators.

The ENTER switch is used with the MEM switch to load data into memory from the register-entry switches.

The procedures for displaying memory data and entering data into memory are described under manual operations (section 3.2).

3.1.9 REG SELECT Switches and Indicators

When the REG switch (section 3.1.7.5) is pressed, any desired general-purpose register (including A, B, or X) can be selected for displaying its contents on the register-display indicators or altering its contents from the register-entry switches. The register selection is accomplished by entering a binary code using the four REG SELECT switches designated 8, 4, 2, 1. A one bit is produced by pressing the appropriate REG SELECT switch; a zero bit is produced by not pressing the switch. A one bit causes the corresponding indicator to light. The binary codes for specific registers are listed in table 3-1. When the binary code has been entered, the register-display indicators automatically display the contents of the selected register. Changing the contents of the display register, using the DISPLAY CLR and register-entry switches (sections 3.1.5 and 3.1.6), automatically changes the contents of the selected register.

Table 3-1.

Binary Codes for Register Selection

REG SELECT Switches				Selected Register
8	4	2	1	
0	0	0	0	A
0	0	0	1	B
0	0	1	0	X
0	0	1	1*	
0	1	0	0*	
0	1	0	1*	
0	1	1	0*	
0	1	1	1*	
1	0	0	0*	

continued

REG SELECT Switches				Selected Register
1	0	0	1*	
1	0	1	0*	
1	0	1	1*	
1	1	0	0*	
1	1	0	1*	
1	1	1	0*	
1	1	1	1*	

* These codes select registers that are not used for programming with the instruction set of a Varian 70 system. They are available for WCS microprogramming use. With two exceptions, the contents of these registers can be displayed and altered using the control panel. However, such alteration should be done only for maintenance purposes or special applications. The register selected with 0100 always contains the contents of the instruction register. The registers selected with 0011 and 0101 always contain all zeros and all ones, respectively; the contents of these two registers cannot be altered from the control panel.

3.1.10 CLEAR and INCR Switches

The binary code for a selected register (section 3.1.9) is cleared (set to zero) by pressing the CLEAR switch. Each time the INCR switch is pressed, the binary code for a selected register is incremented by one, selecting the subsequent register.

3.1.11 INT Switch

The INT switch is used to interrupt the computer and is functional only in the run mode (RUN indicator on). Pressing the INT switch interrupts to memory address zero.

3.1.12 RESET Switch

Pressing the RESET switch:

a. Halts the computer

b. Stops I/O operation

c. Initializes both the computer and its peripheral devices

d. Leaves the computer in step mode

e. If the computer was in run mode, turn the RUN indicator off and the STEP indicator on.

f. Resets the overflow indicator (bit-8 register-display indicator with STATUS switch pressed, section 3.1.7.2).

3.1.13 SENSE Switches and Indicators

The three alternate-action SENSE switches permit the execution of predetermined program branching by the operator. When the program contains jump, jump-and-mark, or execution instructions that depend upon the setting of the SENSE switches, the jumps and executions occur only if the switch conditions are met.

Pressing a SENSE switch sets it and causes its associated indicator to light. Pressing the same switch again resets it, causing its indicator to go out.

For example, a program can be written so that the operator can obtain a partial total of a column of figures being added by use of the JSS1 (jump if SENSE switch 1 is set) instruction. The program writes individual entries as long as SENSE switch 1 is not set. When the operator wants a partial total, he sets the switch. The program then jumps to an instruction sequence that prints the desired information.

3.1.14 BOOT SELECT Switch

The BOOT SELECT switch is a key-operated switch available with the Varian 74 to select one of the three automatic bootstrap programs. After the selection is made, the bootstrap program is automatically loaded into memory by pressing the BOOT switch (section 3.1.4).

BOOT SELECT switch positions with corresponding automatic bootstrap programs are listed below:
Position 1 Teletype
Position 2 High-speed paper tape reader
Position 3 Disc memory

3.2 MANUAL OPERATIONS

Using the control-panel switches and indicators (section 3.1), data or instructions can be transferred manually to or from memory or a selected register, and stored programs can be executed manually.

Manual execution of a stored program is discussed in section 3.3.

3.2.1 Displaying Register Contents

To display the contents of the instruction register:

a. Place the computer in step mode

b. Press I

To display the contents of the program counter:

a. Place the computer in step mode

b. Press P

To display the contents of the A, B, or X or register:

a. Place the computer in the step mode

b. Press REG

c. Using the four REG SELECT switches, enter the appropriate binary code (0000 for A register, 0001 for B register, or 0010 for X register)

3.2.2 Displaying Memory Contents

To display the contents of a memory address:

a. Place the computer in step mode

b. Press P

c. Using the DISPL CLR and register entry switches, enter the desired memory address in the program counter

d. Press MEM

e. Press DISPL switch. The contents of the selected memory address are now displayed on the register-display indicators. The program counter is automatically incremented.

f. Repeated actuation of the DISPL switch displays the contents of consecutive memory addresses

3.2.3 Displaying Overflow Status

To display the overflow status:

a. Place the computer in step mode

b. Press STATUS

c. If register-display bit 8 is on, there is overflow.

3.2.4 Entering Data in Memory

To enter data in memory:

a. Place the computer in step mode

b. Press P

c. Using the DISPL CLR and register-entry switches, enter the memory address in the program counter.

d. Press MEM

e. Using the DISPL CLR and register entry switches, enter the data in the display register.

continued

f. Press ENTER to load the data in the previously addressed memory location. The program counter is automatically incremented.

g. Repeat steps e and f to enter data in consecutive memory addresses.

3.2.5 Entering Data in a Register

To enter data or instructions in a register:

a. Display the contents of the selected register as described in section 3.2.1.

b. Using the DISPLAY CLR and register-entry switches, enter the desired data or instruction in the selected register.

3.3 PROGRAM EXECUTION

To make a cold start (i.e., when a new system is being initialized or the contents of memory are unknown):

a. Turn the power on

b. Load the bootstrap program

c. Load the binary load/dump program

d. Load the object program

Instructions for steps a and b are provided below. Loading the binary load/dump and object programs is discussed in the Binary Load/Dump Program section of the applicable system handbook, with manual execution of a stored program explained below in section 3.3.3.

3.3.1 Power On

Turn on computer power by placing the POWER switch to ON. When power is first turned on, the following conditions apply:

a. Step mode (STEP indicator on)

b. Sense switches not set (SENSE indicators off)

c. Display register cleared (register display indicators off)

d. P switch on (P indicator on)

e. REG SELECT switches off (REG SELECT indicators off)

When power is removed and reapplied without actuation of the POWER switch (by loss and recovery of the ac line voltage), the same conditions apply, except the computer will be in the run mode (RUN indicator on) instead of the step mode.

3.3.2 Loading the Bootstrap Program

The bootstrap program permits the loading of the binary load/dump program into memory. Various input devices such as Teletype paper tape reader, high-speed paper tape reader, or disc memory unit can be used to load the binary load/dump program. The computer is wired at the factory to allow it to operate with a specific input device.

The automatic bootstrap program for the Teletype is included as a standard feature with Varian 72 and 73 computers. With the Varian 74, three automatic bootstraps are included: Teletype paper tape reader, high-speed paper tape reader, and disc memory. The desired bootstrap is selected with the BOOT SELECT switch. Switch positions 1, 2, and 3 select the Teletype, high-speed paper tape reader, and disc memory, respectively. Before the bootstrap program is loaded, the binary load/dump tape (if paper tape input is being used) should be inserted into the paper tape reader with the first binary frame at the reading station.

Addresses and instruction codes (octal) for the automatic bootstrap programs are listed in tables 3-2 and 3-3. When loading a bootstrap program manually, refer to table 3-4.

To load the automatic bootstrap program:

a. With the POWER switch in the ON position, place the computer in the run mode by pressing the STEP/RUN switch (RUN indicator blinking)

b. Press BOOT (RUN indicator is now on). This transfers the bootstrap program from the processor's control-store to computer memory. The binary load/dump program can now be loaded into memory automatically

To load the bootstrap program manually:

a. With the POWER switch in the ON position, place the computer in step mode (STEP indicator on)

b. Press P

c. Using the DISPL CLR and register entry switches, enter the starting memory address (007756) of the bootstrap program in the program counter

d. Press MEM

e. Using the DISPL CLR and register entry switches, enter the appropriate code of the next instruction in the console register (table 3-2)

f. Press ENTER to load instruction code into the memory address specified by the program counter. The program counter is incremented automatically.

g. Repeat steps e and f for each of the remaining bootstrap instructions

Table 3-2. Automatic Bootstrap Programs for High-Speed and Teletype Readers

Address	Instruction Code	Symbolic Coding
000200	102637* (102601)	READ CIB RDR
000201	004011	ASLB NBIT
000202	004041	LRLB 1
000203	004446	LLRL 6
000204	001020	JBZ SEL
000205	000214	(Memory address)
000206	055000	STA 0,1
000207	001010	JAZ LHLT + 1
000200	007000	(Memory address)
000211	0005144	IXR
000212	0005101	ENTR INCR 1
000213	100537* (102601)	SEL RDON
000214	101537* (101201)	SEL EXC IBFR,READ
000215	000200	(Memory address)
000216	001000	JMP *2
000217	000214	(Memory address)

*When using the Teletype reader, replace this code with the one in parentheses.

Table 3-3. Automatic Bootstrap Program for Disc Memory

Address	Instruction Code
001130	100416
001131	104016
001132	100216
001133	005001
001134	103116
001135	101016
001136	001141
001137	001000
001140	001135
001141	102516
001142	151167
001143	100021
001144	001130
001145	100021
001146	100316
001147	005102
001150	103216
001151	103120
001152	006010
001153	001130
001154	103121
001155	100020
001156	100016
001157	101416
001160	001157
001161	102516
001162	151167
001163	001016
001164	001130
001165	001000
001166	000600
001167	007760

Table 3-4. Manual Bootstrap Programs

Address	High-Speed Reader	Teletype Reader Code	Symbolic Coding
007756	102637	102601	READ CIB RDR
007757	004011	004011	ASLB NBIT - 7
007760	004041	004041	LRLB 1
007761	004446	004446	LLRL 6
007762	001020	001020	JBZ SEL
007763	007772	007772	(Memory address)
007764	055000	055000	STA 0,1
007765	001010	001010	JAZ LHLT + 1
007766	007000(a)	007000(a)	(Memory address)
007767	005144	005144	IXR
007770	005101	005101	ENTR INCR 1
007771	100537	102601	SEL RDON
007772	101537	101201	SEL EXC IBFR,READ
007773	007756	007756	(Memory address)
007774	001000	001000	JMP *-2
007775	007772	007772	(Memory address)

NOTE

The bootstrap loader routine is always loaded into the highest address of the first 4K memory increment, regardless of available memory. BLD II relocation and adaptation to the specified input device are described in the Binary Load/Dump Program section of the applicable system handbook.

(a) Replace this code with 007600 if the test executive of MAINTAIN II document 98 A 9952 060) is to be loaded and executed.

3.3.3 Executing a Stored Program

To execute a stored program manually:

a. Place the computer in step

b. Press P

c. Using the DISPL CLR and register entry switches, enter the address of the first program instruction in the program counter

d. Press I

e. Press DISPL CLR to clear the instruction register

f. Press START. This loads the instruction specified by the program counter into the instruction register

g. Press START again. This executes the instruction and loads the next program instruction into the instruction register

h. Repeat step g for each instruction in the program

To execute a stored program automatically: perform steps a through e above, place the computer in the run mode by pressing the RUN/STEP switch, and press START.

SECTION 4
THEORY OF OPERATION

4.1 GENERAL

This section describes the circuit operations of the processor board, I/O control portion of the option board, and control-panel board. Furthermore, the control-store word format and addressing are also described. For the most part, the circuit descriptions are written to functional block diagrams that accompany the text. In some cases, however, the reader may wish to refer to the appropriate logic diagram in the System Maintenance Manual. Logic diagram part numbers are:

a. Processor board, 91B0378
b. Option board, 91B0401
c. Control-panel board, 91B0406

For ease of reading, some mnemonics are written with the variable n in place of the actual bit numbers. For example, ALU data mnemonics DAL00+ through DAL15+ are written DALn+(0-15). Mnemonic conventions and definitions are provided in section 6.

4.2 FUNCTIONAL DESCRIPTION

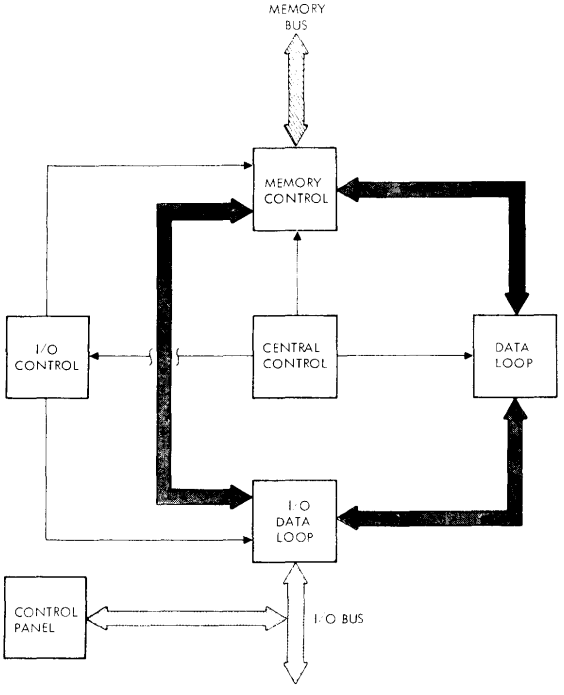
As illustrated in figure 4-1, the major functional sections of the processor are central control, data loop, memory control, I/O data loop, and I/O control. Except for the I/O control, which is located on the option board, these sections are on the processor board. Communication between the processor and the control panel is via the I/O bus. The processor bus structure is illustrated in figure 4-2. A brief description of the five major sections is provided in the following subsections.

4.2.1 Central Control

The central control is the heart of the processor. It contains the instruction register, control store, control store buffer, and control sequencing logic. The following functions are performed by the central control:

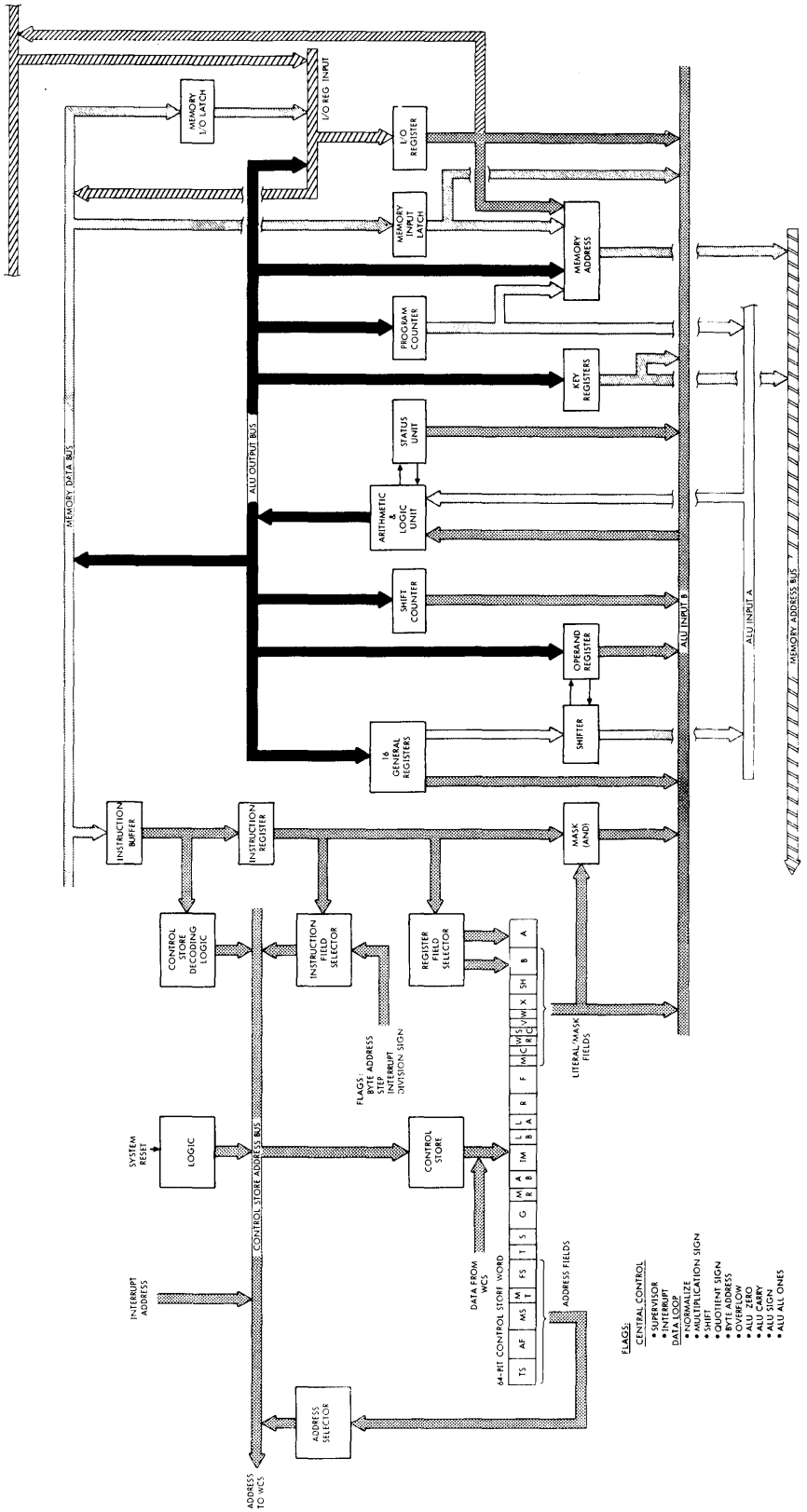
a. Initiates memory operations
b. Initiates I/O operations
c. Decodes instructions
d. Controls data transfers and manipulations
e. Tests internal data loop conditions
f. Responds to interrupts

The 16-bit instruction register receives instructions from an instruction buffer which is then free to accept new instructions. This double buffering of instructions provides a pipelining technique that allows the next instruction to be fetched during an otherwise unused memory cycle. The output of the instruction register can then be routed to the arithmetic and logic unit (ALU) or further decoding may be performed.



VTII-1500A

Figure 4-1. Processor Functional Block Diagram



VT13-280A

Figure 4-2. Processor Bus Structure

4.2.2 Data Loop

The data loop provides data transfer paths, data manipulation circuits, storage registers, and counters (figure 4-2). The data loop performs the following functions:

a. Selects both of the ALU inputs from the following sources:
   1. 16 general purpose registers
   2. Operand register
   3. Memory input latch, in memory control section
   4. I/O register, in I/O data loop section
   5. Status word (signals displayed by control-panel STATUS switch)
   6. Instruction register (masked), in central control section
   7. Program counter
   8. Control store literal which consists of a 16-bit mask field from the control store buffer.

b. Performs arithmetical and logical operations on the ALU inputs.

c. Performs single and double length, bidirectional, open or closed, arithmetical or logical shifts in accordance with the contents of the shift counter.

d. Stores and selects the desired test conditions such as ALU output zero, overflow, carry, SENSE switches, etc.

The ALU performs arithmetic and logical functions under control of the control store buffer. The ALU output is applied to the memory control and I/O data loop sections.

4.2.3 Memory Control

The memory control performs tasks initiated by the central control, I/O control, or options (on option board). It acknowledges acceptance of these tasks to the requesting section and signals completion. Once a request is accepted, no further requests are acknowledged (one exception to this rule permits the central control to override a previous request before it has been completed). Priority memory access (PMA) requests have a higher priority than I/O requests, and I/O requests have a higher priority than central control requests.

The following functions are performed by the memory control:

a. Accepts tasks from central control and stores the following information to complete the task:
   1. Read/write
   2. Word/byte
   3. Address source

b. Accepts tasks from options

c. Accepts tasks from I/O control

d. Acknowledges receipt of tasks

e. Resolves priority of simultaneous requests

f. Defers requests if higher priority devices, such as PMA, request use of the memory bus.

g. Provides asynchronous operation and drivers/receivers for the memory bus.

h. Signals completion of scheduled task.

Since the memory control operates asynchronously, the central control is free to perform other non-memory operations while the scheduled task is being executed. The memory control's ability to accept tasks from the I/O control permits direct memory access (DMA) operations to cycle steal without interfering with non-memory operations in the remainder of the processor.

4.2.4 I/O Data Loop

The I/O data loop (see figure 4-2 for I/O data paths) contains a multiplexor, I/O register, and drivers and receivers. Three sources of data are applied to the I/O data loop: data from the I/O bus, data from the ALU, and data from the memory I/O latch. The input data is selected by the I/O multiplexor under control of I/O control signals and transferred onto the bidirectional I/O bus.

In addition to being applied to the I/O drivers, the output of the I/O register is applied to the data loop and memory control sections.

4.2.5 I/O Control

The I/O control operates under control of an independent I/O control store and performs I/O operations initiated either by the central control or peripheral device activity. This permits I/O operations to proceed with minimum impact on other internal processor functions. The I/O control performs the following functions:

a. Programmed I/O initiated by the central control.

b. DMA trap-in/trap-out operations (up to 372,900 words per second, with semiconductor memory).

c. High-speed DMA trap-in/trap-out operations (up to 969,600 words per second, with semiconductor memory).

d. I/O interrupts

4.3 CENTRAL CONTROL CIRCUITS

This section describes the hardware circuits of the central control. To understand the functions of the central control, one must also become familiar with the microprogramming

operations (firmware). Listed below are sections of this manual containing microprogramming information:

a. Control-Store Word Format (section 4.8)

b. Control-Store Addressing (section 4.9)

c. Microprogramming Examples (section 4.11)

Other microprogramming documents to be used in conjunction with this manual are:

a. The Microprogramming Guide (98 A 9906 07x) describes capabilities, techniques, and coding of microprogramming.

b. The Micro-Word Flowcharts (98A0887) are contained in the System Maintenance Manual.

The circuits of the central control are shown in the block diagram of figure 4-3. Page numbers of the processor logic diagrams are provided in parentheses for each circuit block.

4.3.1 Instruction Register

The 16-bit instruction register receives instructions M1In + (0-15) from the instruction buffer (in the memory control section) which is then free to accept new instructions. This double buffering of instructions provides a pipelining technique that allows the next instruction to be fetched during an otherwise unused memory cycle. Refer to section 3 of Microprogramming Guide for further information on the instruction pipeline.

Data is clocked into the instruction register by clock signals that result from gating the full clock KKC2 + with CTEG0 + and CBG0 +. The register clocks occur whenever the control-store buffer T field contains 00 (CTEQ0 + is high) and the G field bit 0 is true (CBG0 + is high). The register generates instructions C2In(0-15) in both true and complement form.

4.3.2 Instruction Field Selector

The instruction field selector consists of ten 8-input multiplexors which can select a field of up to five bits from the instruction register to form the low order bits of the control store address. This is accomplished by using the FS field of the control store buffer CBFSn + (0-3) to specify which of 16 possible fields are to be extracted and applied to the control store address lines CEADn– (0-8). Since there are not enough bits in the 16 bit instruction register to provide 16 5-bit fields, the remaining bits (multiplexor inputs) are either grounded to provide a source of logical ones to the control-store address or connected to various processor flags for conditional branching. All outputs from the instruction field selector are masked by bits in the MS and MT fields of the control store buffer, thus providing a generalized method for generating addresses. That is, addresses may be formed which either depend on the contents of the instruction register or may be arbitrarily specified by selecting a field position containing logical ones and then masking off the desired logical zero bits using the MS and MT fields.

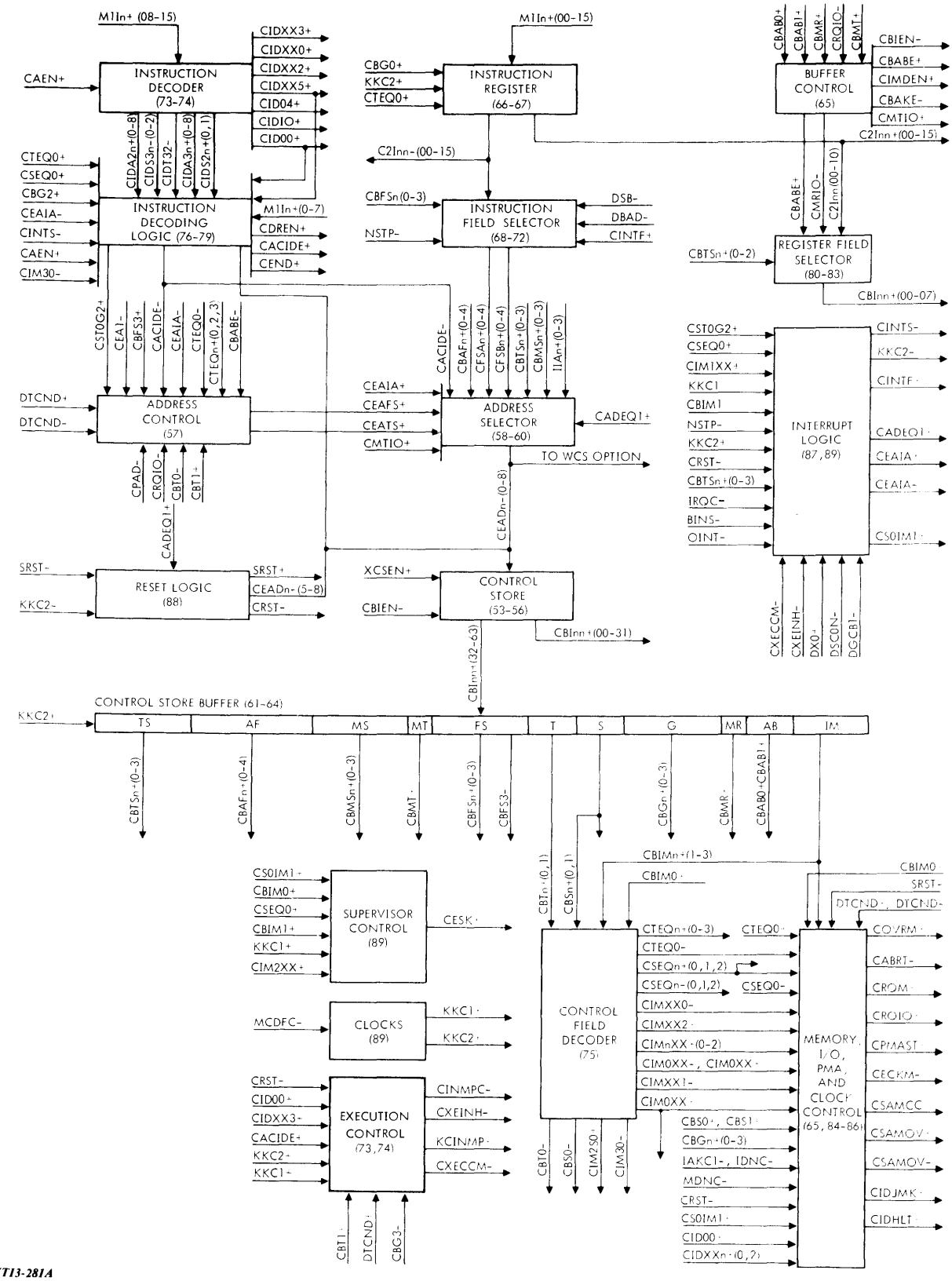
Bits 0, 1, and 2 of the FS field select one input of each multiplexor. The bit-3 active low and high signals of the FS field (CBFS3– and CBFS3 +) enable the five A multiplexors CFSAn + (0-4) or the five B multiplexors CFSBn + (0-4). The multiplexor truth table is shown in table 4-1.

Table 4-1. Instruction Multiplexor Truth Table (IC Type SN74151)

CBFSn+ (2-0)			CBFS3±	INPUT DATA								OUTPUT CFSxn+
PINS			PIN	PINS								PIN
9	10	11	7	4	3	2	1	15	14	13	12	6
			H									H
L	L	L	L	L								H
L	L	L	L	H								L
L	L	H	L		L							H
L	L	H	L		H							L
L	H	L	L			L						H
L	H	L	L			H						L
L	H	H	L				L					H
L	H	H	L				H					L
H	L	L	L					L				H
H	L	L	L					H				L
H	L	H	L						L			H
H	L	H	L						H			L
H	H	L	L							L		H
H	H	L	L							H		L
H	H	H	L								L	H
H	H	H	L								H	L

NOTE: 1. H = HIGH, L = LOW, AND NO H OR L = IRRELEVENT INPUT.
2. THE FIVE A MULTIPLEXORS CFSAn+(0-4) ARE ENABLED WITH A LOW CBFS3–, AND THE FIVE B MULTIPLEXORS CFSBn+(0-4) WITH A LOW CBFS3+.

VTII-1601A



VT13-281A

Figure 4-3. Central Control Block Diagram

4.3.3 Instruction Decoder

The instruction decoder provides preliminary decoding of the two most-significant 4-bit fields of the instruction buffer M1In + (8-15). Decoding is accomplished with ROMs (read only memories) that are programmed for optimum performance. As illustrated in figure 4-4, the ROMs are divided into two sections, A2 and A3. The A2 ROM decodes M1In + (8-11) and produces 9-bits of output data CIDA2n + (0-8) plus control signals. The A3 ROM decodes M1In + (12-15) and produces 9-bits of output data CIDA3n + (0-8) plus control signals. The two 9-bit data words are applied to the instruction decoding logic where they are combined with the two least-significant 4-bit fields of the instruction buffer.

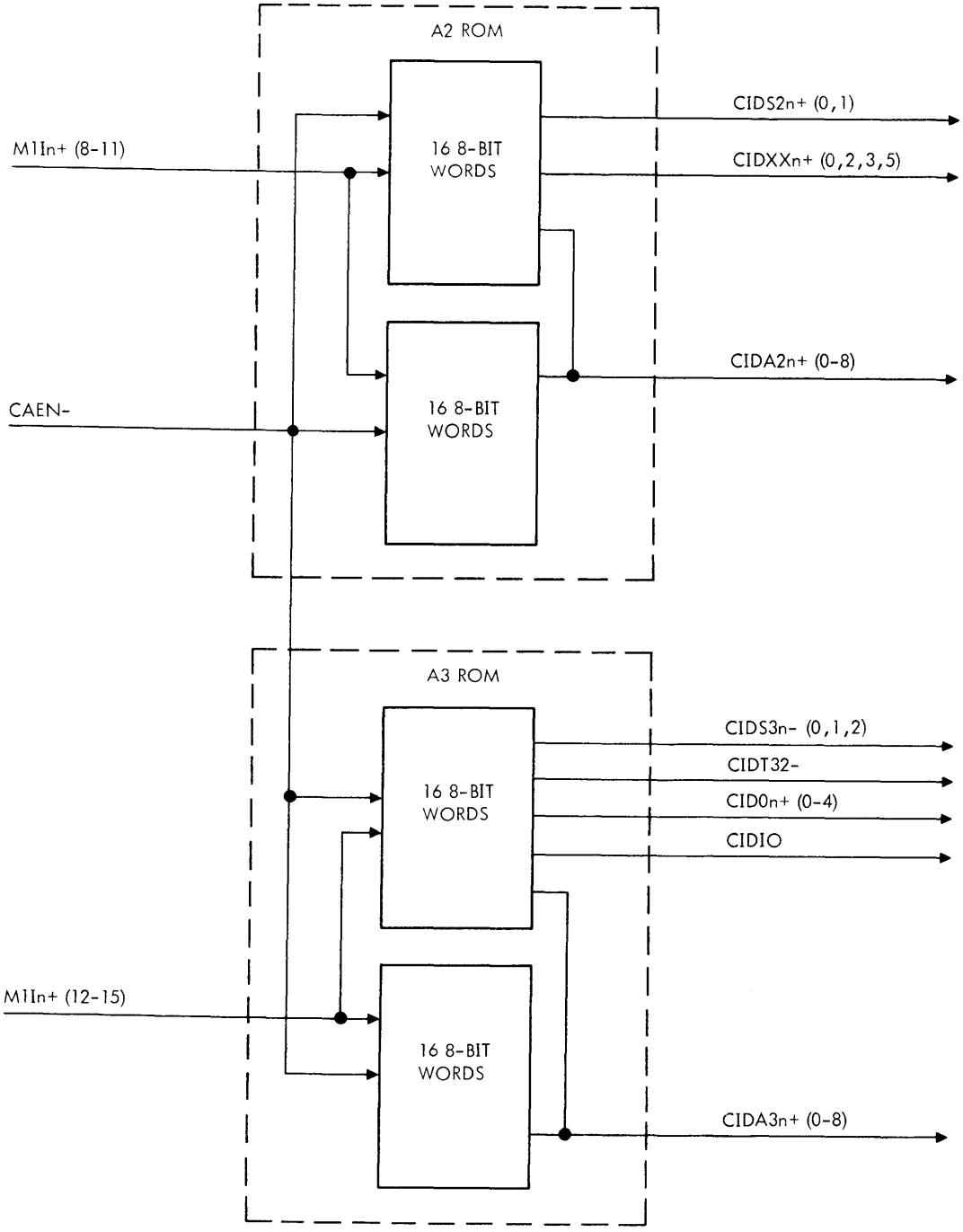


Figure 4-4. Instruction Decoder ROMs Block Diagram

A low CAEN– enables the ROMs. Decoded halt and jump-and-mark signals (CIDHLT + and CIDJMK +) are derived from ROM control signals CID00 + , CIDXX0, and CIDXX2 + . Truth tables for the A2 and A3 ROMs are shown in tables 4-2 and 4-3.

Table 4-2. A2 ROM Truth Table

INPUT BITS	OUTPUTS (CID...)														
11 10 9 8	S21+	S20+	XX3+	XX0+	XX2+	XX5+	A28+	A27+	A26+	A25+	A24+	A23+	A22+	A21+	A20+
L L L L	L	L	L	L	H	L	L	L	H	L	L	L	L	L	L	L
L L L H	L	L	L	L	H	L	L	L	H	L	L	L	L	L	L	L
L L H L	L	L	L	L	L	L	L	L	H	L	L	H	L	L	L	L
L L H H	L	L	L	L	L	L	L	L	H	L	L	H	L	L	L	L
L H L L	L	L	L	L	L	H	L	L	H	L	H	L	L	L	L	L
L H L H	L	L	L	L	L	H	L	L	H	L	H	L	L	L	L	L
L H H L	L	L	L	H	L	L	L	L	H	L	H	H	L	L	L	L
L H H H	L	L	L	H	L	L	L	L	H	L	H	H	L	L	L	L
H L L L	L	L	L	L	L	L	L	L	H	H	L	L	L	L	L	L
H L L H	L	L	L	L	L	L	L	L	H	H	L	L	L	L	L	L
H L H L	L	H	L	L	L	L	H	L	H	H	L	H	H	L	L	L
H L H H	L	H	L	L	L	L	H	L	L	H	L	H	H	L	L	L
H H L L	L	L	H	L	L	L	L	L	L	H	H	L	H	L	L	L
H H L H	L	H	L	L	L	L	L	L	H	H	H	L	L	L	H	H
H H H L	L	L	L	L	L	L	L	L	H	H	H	H	H	H	L	L
H H H H	L	L	H	L	L	L	L	L	H	H	H	H	H	H	L	L

NOTE: L = LOW, H = HIGH

VTII-1603

Table 4-3. A3 ROM Truth Table

INPUT BITS	OUTPUTS (CID...)														
15 14 13 12	S32–	S31–	S30–	T32–	D04+	D00+	DI/O+	A38+	A37+	A36+	A35+	A34+	A33+	A32+	A31+	A30+
L L L L	L	H	H	H	L	H	L	L	L	L	L	L	L	L	L	L
L L L H	H	H	H	L	L	L	L	H	H	L	L	L	L	L	H	L
L L H L	H	H	H	L	L	L	L	H	H	L	L	L	L	L	H	L
L L H H	H	H	H	L	L	L	L	H	H	L	L	L	L	L	H	L
L H L L	H	H	H	L	H	L	L	H	H	L	L	L	L	L	H	L
L H L H	H	H	H	L	L	L	L	H	H	L	L	L	L	L	L	H
L H H L	H	H	H	L	L	L	L	H	H	L	L	L	L	L	L	H
L H H H	H	H	H	L	L	L	L	H	H	L	L	L	L	L	L	H
H L L L	H	H	H	L	L	L	H	H	H	L	L	L	L	L	L	L
H L L H	H	H	H	L	L	L	L	H	H	L	L	L	L	L	H	L
H L H L	H	H	H	L	L	L	L	H	H	L	L	L	L	L	H	L
H L H H	H	H	H	L	L	L	L	H	H	L	L	L	L	L	H	L
H H L L	H	H	H	L	L	L	L	H	H	L	L	L	L	L	H	L
H H L H	H	H	H	L	L	L	L	H	H	L	L	L	L	L	H	L
H H H L	H	H	H	L	L	L	L	H	H	L	L	L	L	L	H	L
H H H H	H	H	H	L	L	L	L	H	H	L	L	L	L	L	H	L

VTII-1604

NOTE: L = LOW, H = HIGH

4.3.4 Instruction Decoding Logic

The instruction decoding logic produces the 9-bit address for the control store CEADn– (0-8) by combining the two least significant 4-bit fields of the instruction buffer M1In + (0-7) and the two 9-bit outputs from the instruction decoder CIDA2n + (0-8) and CIDA3n + (0-8). This logic and the instruction decoder are completely general purpose, and can be used to optimize any instruction set's performance by changing the control store contents or by using a writable control store.

Formation of the decoded control-store address is accomplished with open-collector drivers. Section 4.9 provides a detailed description of control-store addressing. The block diagram in figure 4-5 shows the inputs and outputs for the driver circuits. The outputs are the control store address signals, and the inputs are data from the instruction buffer and instruction decoder plus enabling signals generated by the instruction decoding logic. The enabling signals are described as follows:

a. CACIDA + and CACIDB + enable decoded addresses to be applied to the control store, and are high except when inhibited by an interrupt (CDREN +, CAEN +, CEND + are high). A low CIM30– inhibits address decoding.

b. CILC2 + transfers the CIDA2n + (4-8) signals to bit positions 4 through 8 of the control store address. CILC2 + is controlled by signals generated within the instruction decoding logic.

c. CILC2X + transfers the CIDA2n + (0-3) signals to bit positions 0 through 3 of the control store address. CILC2X + is controlled by CID00 +.

d. CILC0 + transfers the M1In + (0-3) signals to bit positions 0 through 3 of the control store address. CILC0 + is controlled by CIDS32–, CIDS20 +, and CIDS30–.

e. CILC1 + transfers the M1In + (4-7) signals to bit positions 0 through 3 of the control store address. CILC1 + is controlled by CIDS32–, CIDS21 +, and CIDS31–.

4.3.5 Address Control

The address control applies two enabling signals (CEAFS + and CEATS +) to the address selector.

CEAFS + is true (high) when one of the following conditions occur:

a. The T field contains binary 10 (CBT1 + high and CBT0– low) indicating a true condition is being tested, and the true condition is met (DTCND + is high).

b. The T field contains 00 (CTEQ0 + is high) indicating no conditional testing is being performed, the interrupt address is not enabled (CEAIA– is high), and the decoder address is not enabled (CACIDE– is high).

c. The T field contains binary 11 (CTEQ3 + is high) indicating that a false condition is being tested, and the false condition is met (DTCND– is high).

CEATS + is true (high) when one of the following conditions occur:

a. Conditional testing is being performed (CTEQ0– is high), and field selection is not enabled (CEA1– is high).

b. No conditional testing is being performed (CTEQ0 + is high), no register field selection (CBABE– is high), the TS field is not enabling interrupts (CST0G2– is high), no I/O request (CRQIO– is high), and there is no page jump (CPAD– is high).

4.3.6 Address Selector

Under control of the address control circuits, the address selector selects the next control store address from one of the following sources:

a. Instruction field selector.

b. AF field of control store buffer.

c. Alternate test address (TS field).

d. Interrupts.

e. RESET switch.

The control store address lines CEADn– (0-8) from the address selector are logically ORed with address lines from the instruction decoding logic and reset logic. The OR function is accomplished by driving the address lines with open-collector integrated circuits.

When an instruction field selection occurs, a high CEAFS + enables a 5-bit field from the instruction field selector CFSAn(0-4) or CFSBn(0-4) to be used as the least significant bits of the control store address CEADn– (0-4). Bit 4 is also enabled by a high CMTIO + which indicates that no I/O operation has been requested and the mask bit is true (CBMT + is high). Bits 0 through 3 are masked by the MS field CBMSn + (0-3).

When no instruction decoding is being performed, a high CACIDE– enables the AF field CBAFn + (0-4) to be used as the most significant bits of the control store address CEADn– (4-8).

When an alternate test address is selected, a high CEATS + enables the TS field CBTSn + (0-3) to be used as control store address bits 1 through 4.

When an interrupt address is required, a high CEAIA + enables interrupt addresses IIAn + (0-3) from the I/O section to be used as the least significant bits of the control store address CEADn– (0-3).

Pressing the control-panel RESET switch, produces a high CADEQ1 + which sets all bits of the control store address to ones (low level). Bits 0 through 4 are set by the address selector, and bits 5 through 8 by the reset logic.

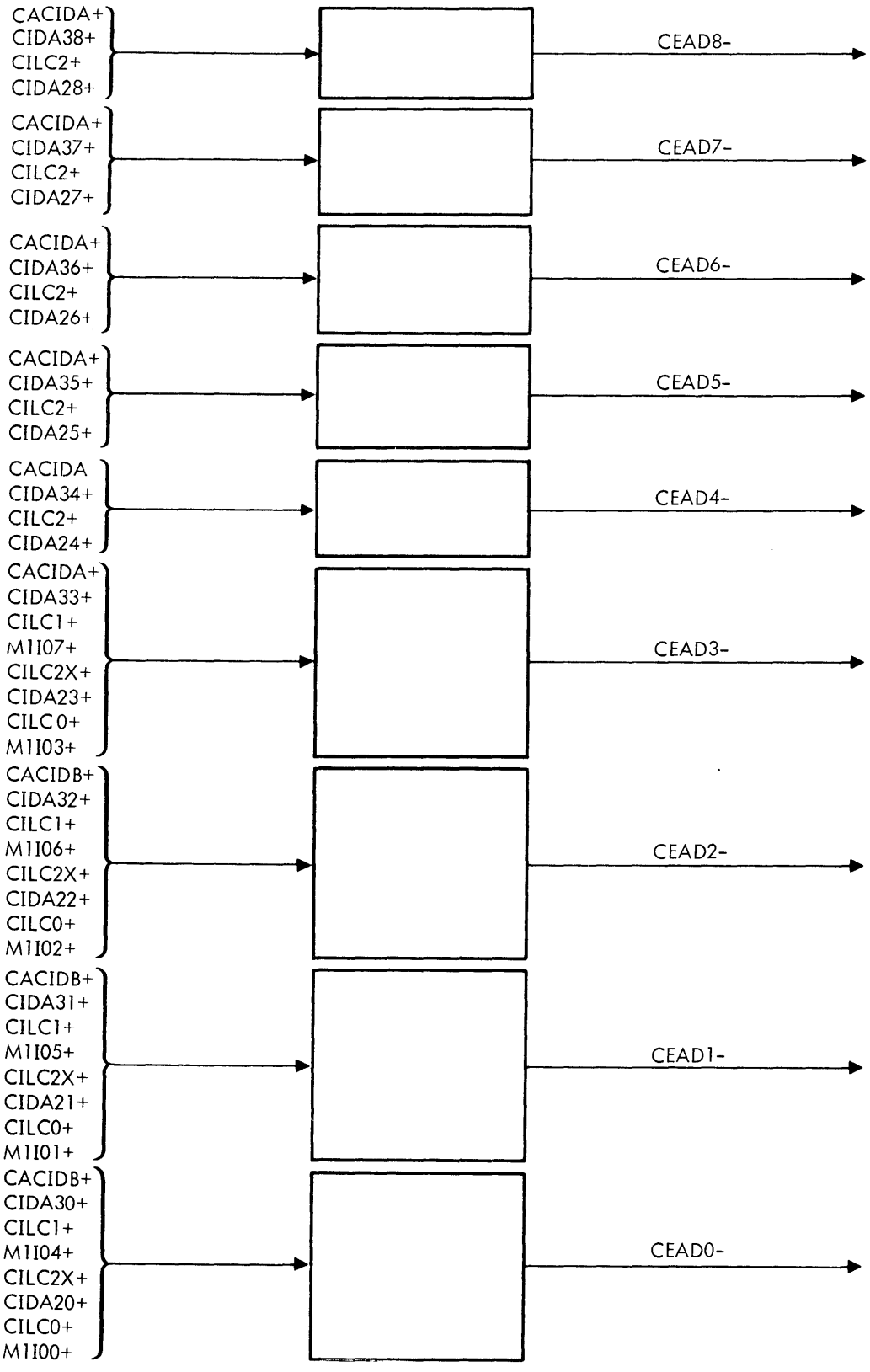


Figure 4-5. Decoding Logic Driver

4.3.7 Control Store

The control store is a 512-word by 64-bit high-speed ROM. It contains the microprogram which emulates the computer's instruction set. Control store output bit 0 through 31 are applied to the control store buffer in the data loop section, and output bits 32 through 63 are applied to the control store buffer in the central control. Since the control store outputs are tristate (or open collector), they can be disabled to allow an external control store to provide the inputs to the control store buffer. Interface signals for the external control store are routed through processor board connectors P2 and P3.

A low XCSEN + disables the control store when an external control store is used (XCSEN + is high with no external control store). A high CBIEN– disables control store bits 0 through 7 and allows these bits to be applied to the control store buffer from the register field selector.

For a list of the binary codes contained at each address of the control store, refer to document 81L1709-001 in the System Maintenance Manual.

4.3.8 Buffer Control

The buffer control decodes certain fields of the control store buffer.

When the AB field does not contain 00 (CBAB0 + or CBAB1 + is high) and there is no I/O transfer requested (CRQIO + is low), a high CBABE + enables the register field selector to provide input data to the A and B fields. A high CBIEN– disables the control store output bits 0 through 7 when the register field selector is enabled (CBABE + is high) or the MR field contains 1 (CBMR + is high). CBAKE– is an enabling signal for the A field of the control store buffer.

CMTIO + is the repowered MT-field bit (CBMT +) which is applied to the address selector when there is no I/O request (CRQIO– high).

CMRIO– is generated by gating CBMR + with CRQIO– to mask the most significant bit in the register field selector.

CIMDEN + is an enabling signal for the file control in the data loop section.

4.3.9 Register Field Selector

The register field selector, which contains four 8-input multiplexors, permits the selection of a 3- or 4-bit field from bits 0 through 10 of the instruction register C2In + (0-10) and applies it to the A and B fields of the control store buffer. This allows the A and B fields to be loaded directly from the instruction register thus saving control store addresses and improving performance for typical instruction sets.

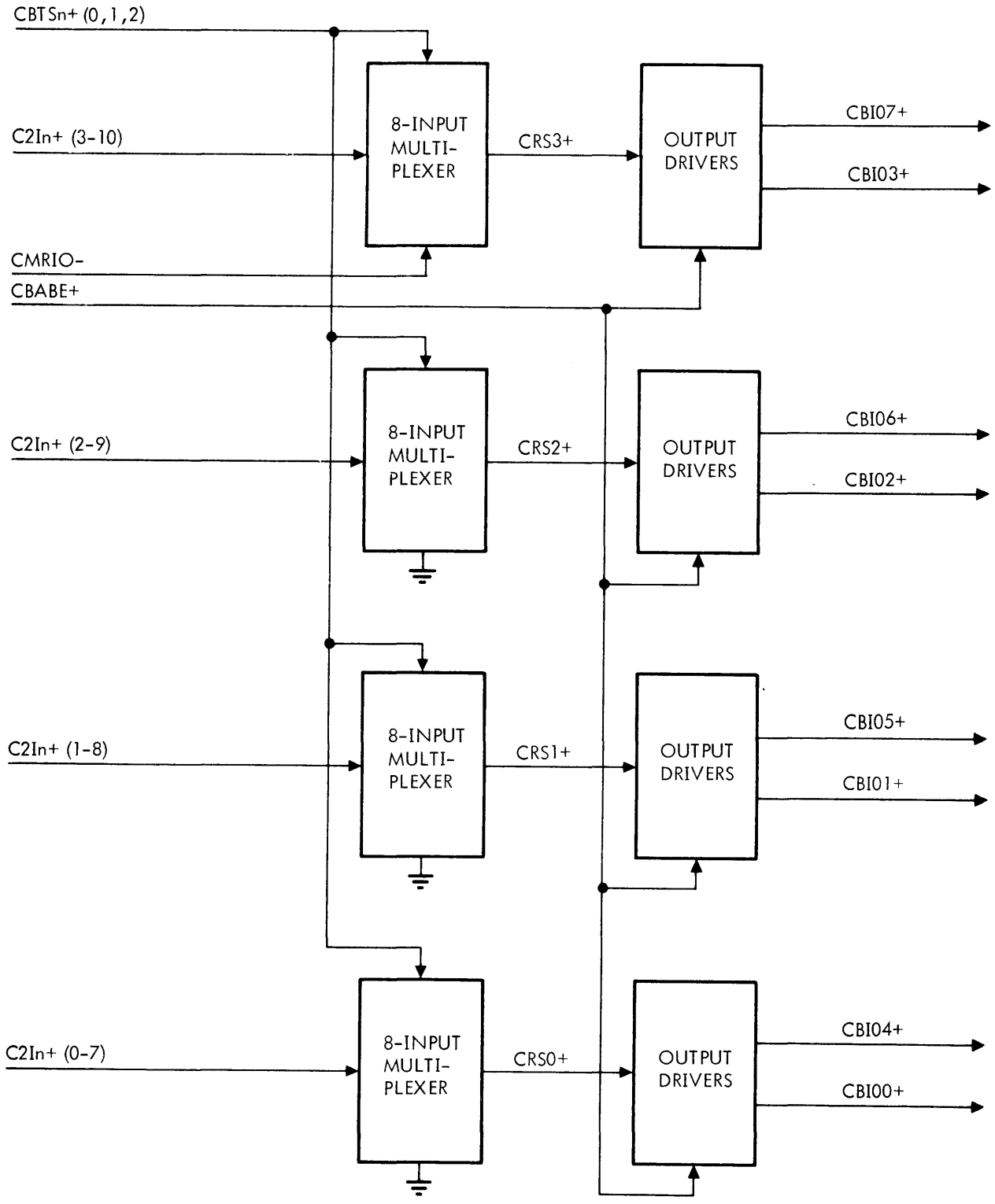
A block diagram of the register field selector is illustrated in figure 4-6. Field selection is controlled by three bits of the TS field CBTSn + (0-2). CMRIO– masks the most significant output bit (CBI03 + or CBI07 +) to permit the selection of either a 3- or 4-bit field. The multiplexor outputs are transferred through open-collector drivers by CBABE + . Table 4-4 is the truth table for the multiplexors.

4.3.10 Control Store Buffer (Bits 32-63)

The central control portion of the control store buffer holds the control fields of the last microinstruction fetched from the control store. Input and output signals for the various fields are shown in the block diagram of figure 4-7. The 32 input bits are simultaneously loaded into eight 4-bit registers on the negative transition of the full clock KKC2 + . For a detailed description of the fields in the control store word, refer to section 4.8.

4.3.11 Control Field Decoder

The control field decoder (figure 4-8) decodes the signals from the T, S, and IM fields of the control store buffer. A block diagram of the control field decoder is illustrated in figure 4-8, and truth tables are provided in tables 4-5, 4-6, and 4-7.



VTII-1606A

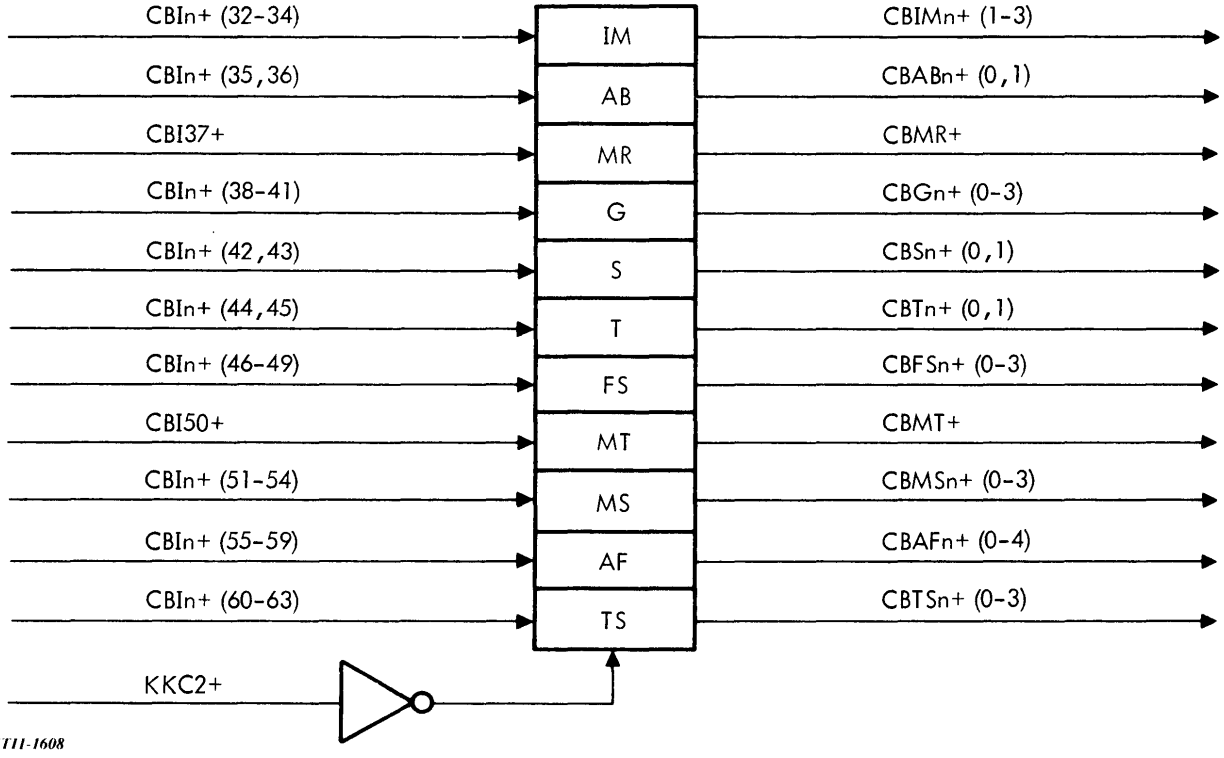
Figure 4-6. Register Field Selector Block Diagram

Table 4-4. Register Multiplexor Truth Table (IC Type SN74151)

CBTSn+ (0-2)			CMRIO- OR GRD	INPUT DATA (C2In+)								OUTPUT CRSn+ (0-3)
PINS			PIN	PINS								PIN
9	10	11	7	4	3	2	1	15	14	13	12	6
			H									H
L	L	L	L	L								H
L	L	L	L	H								L
L	L	H	L		L							H
L	L	H	L		H							L
L	H	L	L			L						H
L	H	L	L			H						L
L	H	H	L				L					H
L	H	H	L				H					L
H	L	L	L					L				H
H	L	L	L					H				L
H	L	H	L						L			H
H	L	H	L						H			L
H	H	L	L							L		H
H	H	L	L							H		L
H	H	H	L								L	H
H	H	H	L								H	L

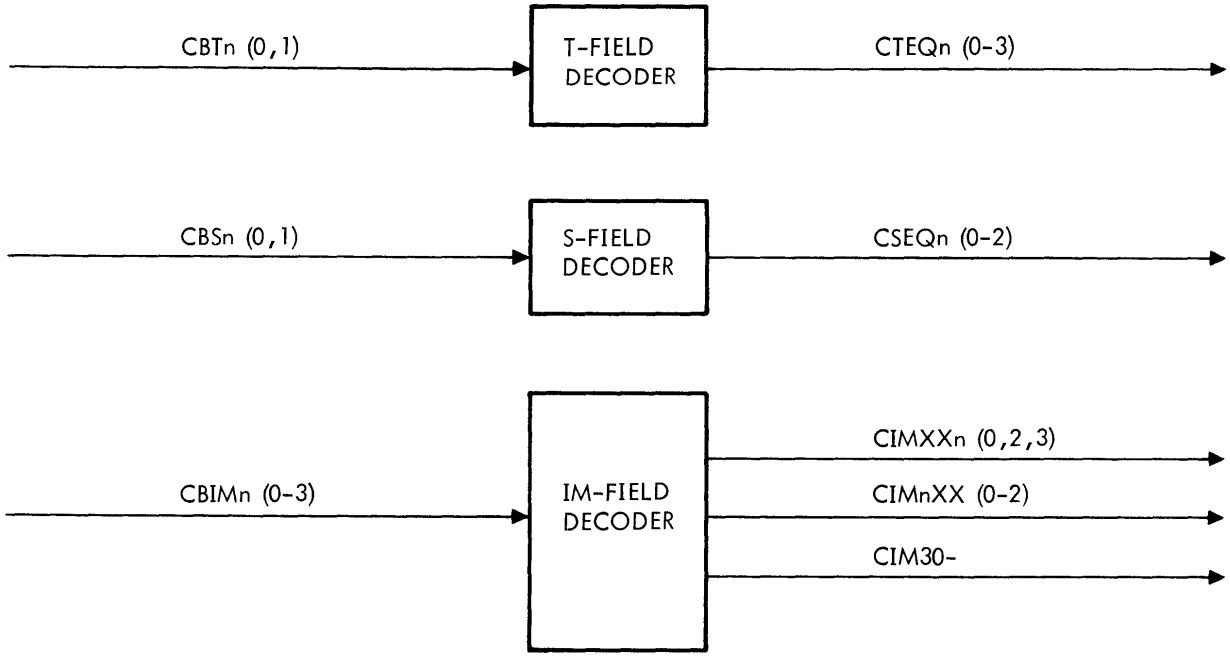
NOTE: H = HIGH, L = LOW, NO H OR L = IRRELLEVANT INPUT

VTII-1607



VTII-1608

Figure 4-7. Control Store Buffer (Bits 32-63)



VTII-1609

Figure 4-8. Control Field Decoder Block Diagram

Table 4-5. T-Field Decoder Truth Table

INPUTS		OUTPUTS			
CBT1+	CBT0+	CTEQ0+	CTEQ1+	CTEQ2+	CTEQ3+
L	L	H	L	L	L
L	H	L	H	L	L
H	L	L	L	H	L
H	H	L	L	L	H

NOTE: L = LOW, H = HIGH

VTII-1631

Table 4-6. S-Field Decoder Truth Table

INPUTS		OUTPUTS		
CBS1+	CBS0+	CSEQ0+	CSEQ1+	CSEQ2+
L	L	H	L	L
L	H	L	H	L
H	L	L	L	H
H	H	L	L	L

NOTE: L = LOW, H = HIGH

VTII-1610

Table 4-7. IM-Field Decoder Truth Table

IM FIELD CODES (CBIMn, 0-3)					OUTPUT SIGNALS (TRUE STATES)
n BITS	3	2	1	0	
	X	X	L	L	CIMXX0-
	X	X	L	H	CIMXX1+
	X	X	H	L	CIMXX2+
	X	X	H	H	CIMXX3+
	L	L	X	X	CIM0XX+
	L	H	X	X	CIM1XX+
	H	L	X	X	CIM2XX+
	H	H	X	L	CIM30-

NOTE: L = LOW, H = HIGH, AND X IS IRRELEVANT

VTII-1611A

4.3.12 Memory, I/O, and Clock Control

Signals applied to the memory control section are: CRQM+, COVRM+, CABRT-, CECKM+, and CECKM-.

The central control requests a memory operation by generating a high CRQM+. CRQM+ is high except during one of the following conditions:

a. The central control overrides the memory request (CSEQ0- and CIM0XX+ are high).

b. The IM field does not contain 0100 when the S field contains 00 (CIMXX0-, CIM1XX-, and CSEQ0+ are high).

The central control can change its memory request by generating a high COVRM+ (override signal). COVRM+ goes high when the S field does not contain 00 (CSEQ0- is high), and the two most-significant bits of the IM field are zero (CIM0XX+ is high).

In order for the central control to successfully initiate a memory request, the abort signal CABRT- must remain high. The memory request is aborted (CABRT- is low) if any of the following conditions occur:

a. A system reset condition occurs (SRST+ is high).

b. The S field contains binary 11 (CBS0+ and CBS1+ are high) and the conditional test is not met (DTCND- is high).

c. The S field contains binary 10 (CSEQ2+ is high), the T field does not contain 00 (CTEQ0- is high), and, the conditional test is met (DTCND+ is high).

CECKM +, and its complement CECKM–, enable clock signals which are generated in the memory control section and used in the central control. CECKM + is normally high but may be inhibited (low) during one of the following conditions:

a. An I/O transfer is requested (CRQIO + is high) but not acknowledged (IAKC1– is high).

b. A wait for a memory acknowledgment occurs and no memory acknowledgment signal is received (MDNC– is high). The wait for a memory acknowledgment occurs when CBIM0 +, CBIM1 +, CSEQ0 +, and CIM0XX + are high.

c. A wait for an I/O done condition. This occurs when CIMXX2 +, CIM0XX +, CSEQ0 +, and IDNC– are high.

d. A system reset condition occurs (SRST– is low).

Signals applied to the I/O control section are: CRQIO +, CIDJMK–, and CIDHLT +.

The high CRQIO + is an I/O transfer request signal that is applied to the I/O control section when the IM field contains 1110 or 1111 and the S field contains 00 (CSEQ0 + is high). CIDJMK + is a decoded jump-and-mark signal and CIDHLT + is a decoded halt signal; they are derived by signals from the instruction decoder (CID00 +, CIDXX0 +, and CIDXX2 +).

CSAMCC + and CSAMOV + (and its complement CSAMOV–) are applied to the status and test logic in the data loop section. A high CSAMCC + enables a clock signal DTKL– for the ALU flags DCNOZ +, DCNDC +, DSGN +, and DEQ +. CSAMOV + goes high whenever an overflow condition is to be sampled by the overflow flag. A high CPMAST + is generated to start a special mode of the PMA option. CPMAST + is high when the T field contains 00 (CTEGO + is high), the S field contains binary 10 (CSEQ2 + is high), and bit 0 of the G field is true (CBG0 + is high).

4.3.13 Reset Logic

The reset logic contains a flip-flop that synchronizes the reset signal (SRST–) from the computer control panel. When the RESET switch is pressed, the flip-flop output CRST– is low and is applied to the interrupt logic to produce a high CADEQ1 +. The CADEQ1 + signal is routed back to the reset logic to set all control-store address bits to ones (low level).

4.3.14 Interrupt Logic

The interrupt logic receives the following types of interrupts:

a. Control panel (NSTP–)

b. I/O (IRQC–)

c. I/O with memory protection installed (BINS–)

d. Option board (OINT–)

Each of the above interrupt signals are masked with a bit from the TS field CBTSn + (0-3). The occurrance of an interrupt signal and its associated mask bit enable the interrupt logic to generate various control signals.

An interrupt from the control panel is initiated when the STEP/RUN switch is placed in the step position. This applies a low NSTP– to the interrupt logic where it is synchronized with the full clock KKC2 + and then gated with bit 3 of the TS field.

When an I/O interrupt is initiated, the I/O control section applies a low IRQC– to the interrupt logic. Any I/O interrupt is accepted by the interrupt logic when bit 0 of the TS field is set (CBTS0 + is high). When TS-Field bit 1 is set (CBTS1 + is high) and bit 0 is reset (CBTS0 + is low), interrupts are accepted only if the memory protection option is installed. BINS– is low when this option is installed.

When an interrupt from the option board is initiated, a low OINT– is applied to the interrupt logic where it is gated with bit 2 of the TS field.

The signals generated by the interrupt logic are: CINTS–, CEAIA–, CINTF +, CADEQ1 +, and CS0IM1 +.

CINTS– is low when the interrupt flag CINTF + is high, IM field contains 011x (x is irrelevant bit), and the S field contains 00. The last two conditions occur when CBIM1 +, CIM1XX +, and CSEQ0 + are high.

CEAIA– is low when all the following conditions occur:

a. An interrupt is accepted.

b. The S and T fields contain 00 and the G-field bit 2 is set (CST0G2 + is high).

c. An interrupt flag has not been selected, or it has been selected but not set (CINTS– is high).

Interrupt flag CINTF + is set by the interrupt service microroutine. It indicates that an I/O interrupt has occurred and is in process. CINTF + is also used at the completion of an interrupt to determine the status of the instruction pipeline. The interrupt flag flip-flop is set (CINTF + is high) when the S field contains 00 and the IM field contains 0111; it is reset when the S field contains 00 and the IM field contains 0110. The clock for the interrupt flag flip-flop is controlled by decoding the three most significant bits of the IM field, and is generated when the IM field contains 011x with the least significant bit irrelevant (x). The least significant bit of the IM field is applied to the data input of the flip-flop. The flip-flop is directly reset when the LB field contains 0x, the X field contains x1, and DSCON– is high.

A high CADEQ1 + is applied to the address selector and reset logic to set the control store address bits to ones (low level). CADEQ1 + goes high when one of the following conditions occur:

a. A reset signal is received from the control panel (CRST– is low).

b. A control panel interrupt is received, bit 3 of TS field is set (CBTS3 + is high), and CEAIA– is low.

4.3.15 Supervisor Control and Clocks

The supervisor control contains a flip-flop which is set (CESK + is high) when the S field contains 00 (CSEQ0 + is high) and the IM field contains 1011 (CIM2XX +, CBIM1 +, and CBIM0 + are high). The flip-flop is reset when the S field contains 00 and the IM field contains 1010 (CIM2XX + and CBIM1 + are high, and CBIM0 + is low).

The clocks circuit generates two clocks (KKC1 + and KKC2 +) from the full clock MCDFC– .

4.3.16 Execution Control

The execution control performs the following functions:

a. Decodes the execution type instruction (003XXX), and stores the decoded information during the instruction execution.

b. Provides a signal to the memory protection option indicating that an execution type instruction has been decoded.

c. Inhibits I/O and control-panel step interrupts until the completion of instruction execution.

The CINMPC + flip-flop is set (CINMPC + high) at the completion of instruction decoding (CACIDE + high) if the instruction decoder outputs CID00 + and CIDXX3 + are both high. This flip-flop is reset when the instruction being decoded is not an execution type instruction.

The CXECCM + flip-flop is set (CXECCM– low) when the condition is met for a jump, jump and mark, or execution test (CBT1 +, CBG3–, and DTCND + are high). The low CXECCM– enables the CXEINH + flip-flop to be set (CXEINH– low) when the next instruction is decoded. This causes the CXECCM + flip-flop to be cleared (CXECCM– high).

The CXEINH + flip-flop is reset during the next instruction decoding time (CACIDE + high).

4.4 DATA LOOP CIRCUITS

The circuits of the data loop are shown in the block diagram of figure 4-9. Page numbers of the processor logic diagram are provided in parentheses for each circuit block.

4.4.1 Control Store Buffer (Bits 0-31)

This portion of the control store buffer holds the first 31 bits of the last microinstruction fetched from the control store. Bits 0 through 30 control data loop operations and bit 31 is used in the central control. Input and output signals for the various fields are shown in the block diagram of figure 4-10. The 32 input bits are loaded into the control store buffer on the negative transition of the full clock KKC1 + . The A and B fields require enabling signals from the buffer control circuit in the central control section. A low CBAKE– enables the A field and a high CABIO1– enables the B field. For a detailed description of the fields in the control store word, refer to section 4.8.

4.4.2 Register Files A and B

The register files A and B comprise 16 general purpose 16-bit registers that can be used in the emulation of an arbitrary instruction set, as well as providing operational registers and working storage for internal operations. Each file consists of four 16-word by 4-bit random-access memories. During a read operation, each file can be independently addressed. This permits simultaneous access to the contents of two arbitrary registers thus eliminating the need for a time consuming sequential access. During a writing operation, the files are identically addressed so that duplicate copies of the written data are available.

A block diagram of the register files are illustrated in figure 4-11. Both register files receive their input data DALn + (0-15) from the arithmetic and logic unit (ALU). The input data are loaded (written) into one of the 16 registers in files A and B with a low DWCN– . One register out of 16 is selected with 4-bit address codes DFAAn + (0-3) and DBSn– (0-3) applied to each file. These address codes and the corresponding registers are listed in table 4-8. Loaded data are available on the output lines of the register files. As an example, if an address code of 0001 and a low DWCN– are generated, the data on the output lines of files A and B are the contents of the B register.

Data on the output lines of file A DFAn– (0-15) are transferred (read) to latch A with a low DENFA– . Data on the output lines of file B DFBn– (0-15) are transferred (read) to latch B with a low DENFB– . In addition, the complements of file A output bits 0 and 15 (DFA0 + and DFA15 +) are applied to latch A for use in various shifting operations.

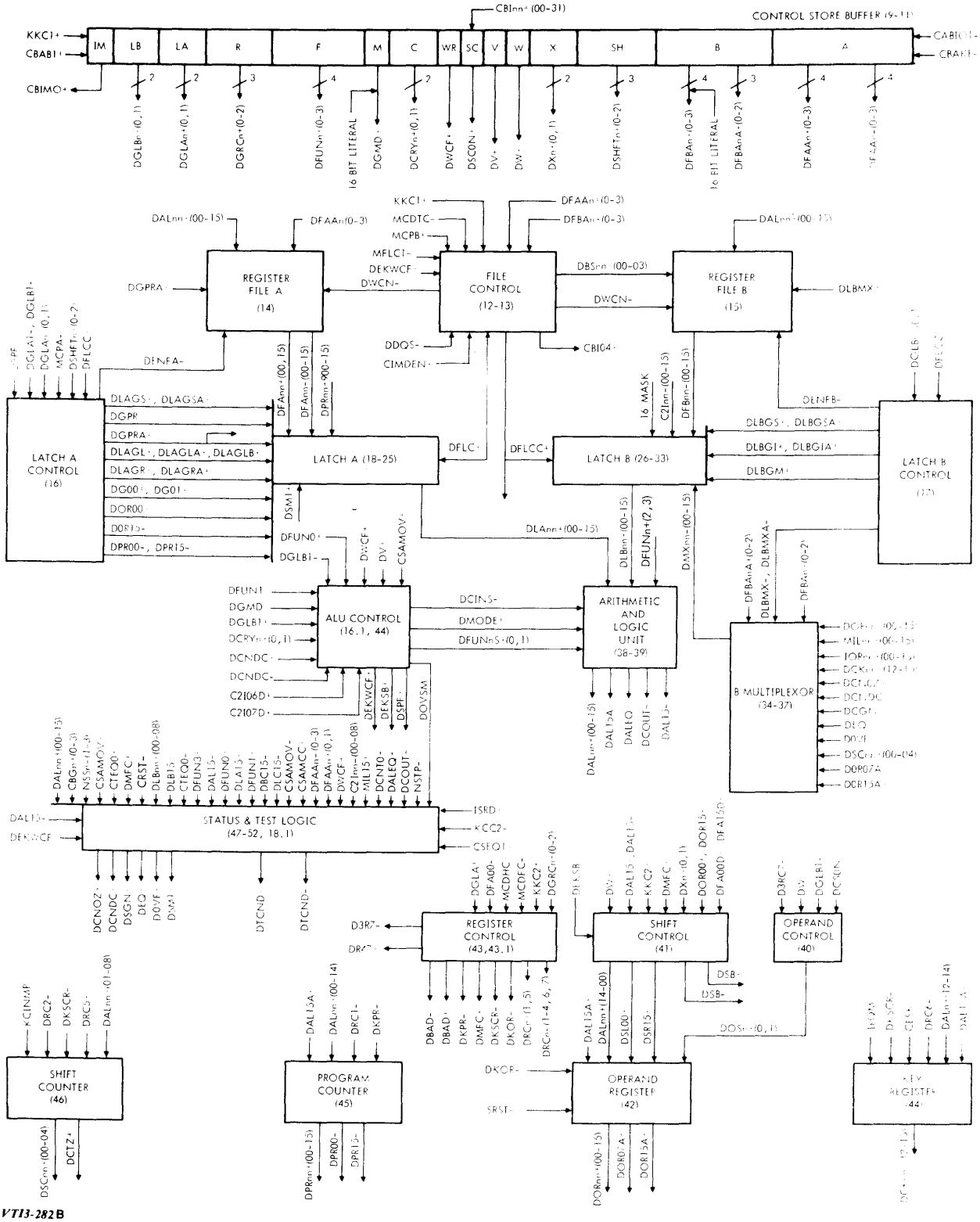


Figure 4-9. Data Loop Block Diagram

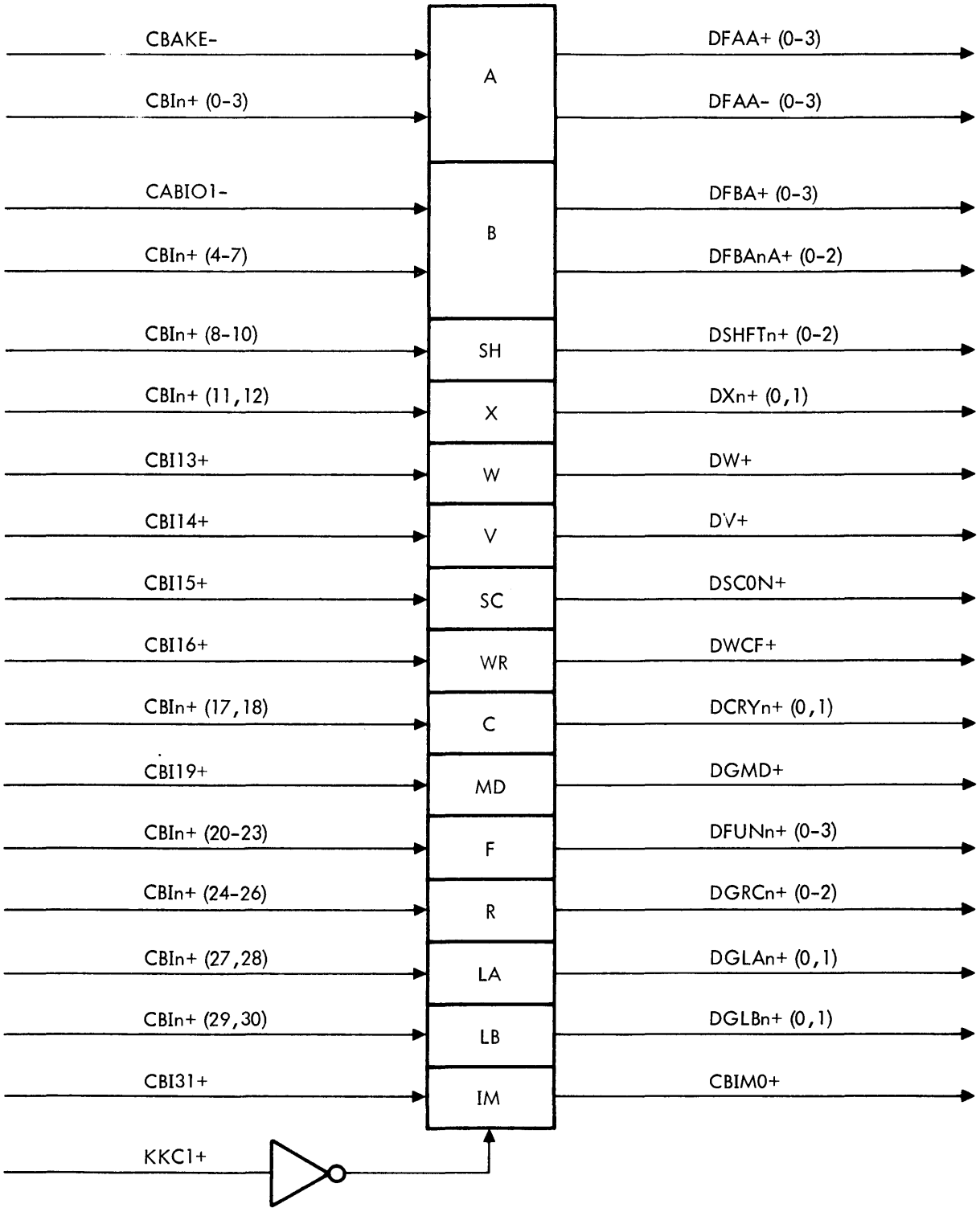


Figure 4-10. Control Store Buffer (Bits 0-31)

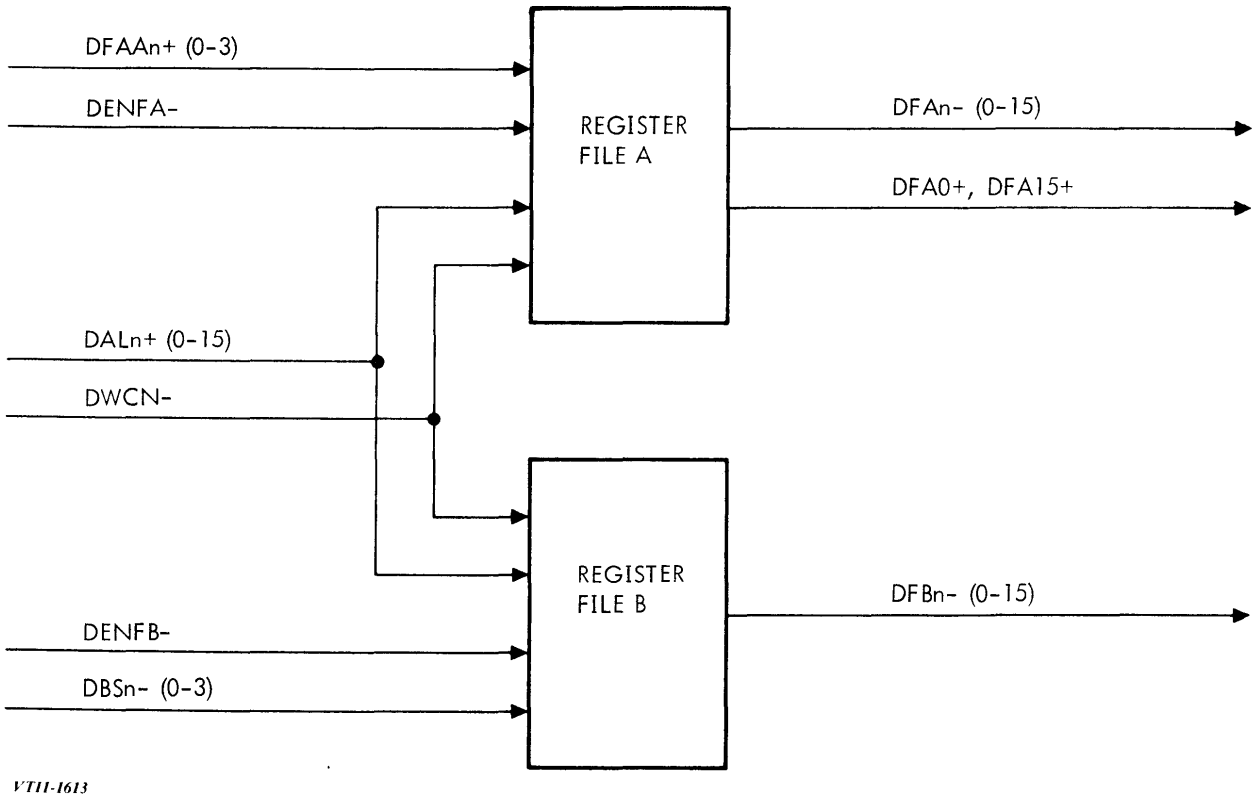


Figure 4-11. Register Files Block Diagram

Table 4-8. Register File Address Codes

Address Codes	Register
0000	A register
0001	B register
0010	X register
0011	Contents are always all zeros.
0100	Instruction register.
0101	Contents are always all ones.
0110	Not used
0111	Not used
1000	Not used
1001	Not used
1010	Not used
1011	Not used
1100	Not used
1101	Not used
1110	Used for temporary storage
1111	Used for temporary storage

4.4.3 File Control

The file control provides register file B with 4-bit address codes DBSn + (0-3) by inverting B-field signals DFBAn + (0-3). During the writing phase (DFLC + is high) of each microinstruction, file-B addresses are switched to the same codes as the file-A addresses. This ensures that identical data is loaded into the same address of both files when DWCN– goes low.

A high DEKWCF +, applied to the file control, produces a low DWCN– that enables data to be written into files A and B.

During the latter portion of the processor cycle, high DFLC + and DFLCC + signals are generated to prevent input data from being loaded into latches A and B.

During multiplication and division operations, CBI04 + is controlled by DDQS– to produce address 1110 or 1111 for register file B. Depending on the state of the W-field bit (DW +), DDQS– is controlled either by the ALU bit 15 or operand register bit 1.

4.4.4 Latch A Control

Latch A control decodes LA field signals DGLAn + (0,1) and generates enabling signals for latch A and register file A. The following signals are applied to latch A:

a. DGPR + and DGPRA + load the contents of the program counter DPRn + (0-15) into latch A.

b. DLAGL + and DLAGLA + load register file A data DFAn + (0-15) which are shifted left one bit position, into latch A.

c. DLAGR + and DLAGRA + load register file A data, which are shifted right one bit position, into latch A.

d. DLAGS + and DLAGSA + load unshifted register file A data into latch A.

A low DENFA– is generated by latch A control to transfer register file A data to latch A. DENFA– goes low when the contents of the program counter are not being transferred to latch A (DENF2 + low), and the conditions specified by the LA, LB, and SH fields are met (DGLA1–, DGLB1–, DLAGS + and DSHFT0 + are high). These conditions specify the output of latch A is zero.

DFLCC +, from the file control, goes high during the latter portion of the processor cycle to inhibit the decoding function of the latch A control.

4.4.5 Latch A

Latch A provides data selection and buffering for the A input of the ALU. The sources of latch A input data are the program counter DPRn + (0-15) and register file A DFAn– (0-15). Latch A buffers the output of the register file A from the ALU input during a file A write operation. A block diagram of the latch A circuits is illustrated in figure 4-12. The following are the types of data that can be loaded into latch A along with corresponding enabling signals from latch A control:

a. Contents of the program counter are loaded into latch A with a high DGPR + and DGPRA +.

b. Unshifted data from register file A are loaded into latch A with a high DLAGS + and DLAGSA +.

c. Register file A data shifted left one bit position are loaded into latch A with a high DLAGL +, DLAGLA +, and DLAGLB +.

d. Register file A data shifted right one bit position are loaded into latch A with a high DLAGR + and DLAGRA +.

DFLC + is a timing signal from the file control circuit. A low DFLC + enables latch A to sample and select the appropriate input data. During the latter portion of the processor cycle, DFLC + goes high causing latch A to

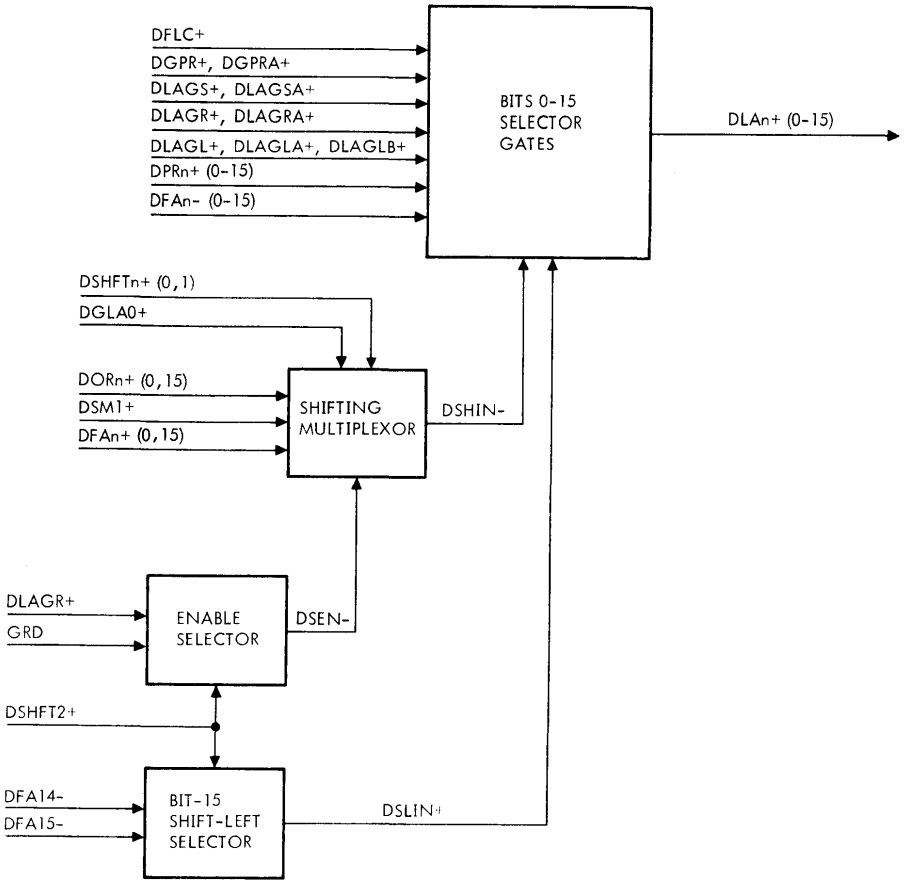


Figure 4-12. Latch A Block Diagram

disregard further changes to its other inputs as occurs when data are written into the register file.

DSHIN– and DSLIN + (figure 4-12) are additional inputs to the bit 0 and 15 selector gates, and are required for implementing shifting operations. DSHIN– is applied to the bit 0 and 15 selector gates and is generated by a multiplexor circuit (refer to table 4-9 for the multiplexor truth table). DSLIN + is applied only to the bit 15 selector gate and is selected from either DFA14– or DFA15–. When DSHFT2 + is high, DFA15– is connected to the DSLIN + line. When DSHFT2 + is low, DFA14– is connected to the DSLIN + line.

A low DSEN– is used to enable the shifting multiplexor. When DSHFT2 + is high, a ground signal is connected to the DSEN– line to enable the multiplexor. When DSHFT2 + is low, DSEN– is controlled by DLAGR +.

4.4.6 Latch B Control

Latch B control decodes LB field signals (DGLB0 + and DGLB1 +) and generates enabling signals for latch B. register file B, and B multiplexor. The following signals are applied to latch B.

a. DLBGI + and DLBGIA + load the contents of the instruction register C2In– (0-15) into latch B.

b. DGLB1– and DLBGM + are produced from bit 1 of the LB field (DGLB1 +) and are used to load the 16-bit control store literal into latch B. The control store literal is the mask field of the control store buffer (fields B, SH, X, W, V, SC, WR, C, and M).

c. DLBGS + and DLBGSA + load data from either the B multiplexor or the register file B.

A low DENFB– transfers register file B data to latch B. DENFB– goes low when register file B is selected as the input for latch B, and remains high when other latch B inputs are selected. DLBMX– and DLBMXA– are complements of bit 0 of the LB field and are used to enable the B multiplexor.

Table 4-9. Shifting Multiplexor Truth Table

DGLA0+	DSHFT1+	DSHFT0+	DSEN–	GRD	DFA15+	DOR15+	OPEN	DSM1+	DFA00+	DFA15+	DOR00+	DSHIN–
			H									H
L	L	L	L	L								H
L	L	H	L		L							H
L	L	H	L		H							L
L	H	L	L			L						H
L	H	L	L			H						L
L	H	H	L				H					L
H	L	L	L					L				H
H	L	L	L					H				L
H	L	H	L						L			H
H	L	H	L						H			L
H	H	L	L							L		H
H	H	L	L							H		L
H	H	H	L								L	H
H	H	H	L								H	L

NOTE: H = HIGH, L = LOW, AND NO H OR L INDICATES IRRELEVANT INPUT

VTII-1615

4.4.7 Latch B

Latch B provides data selection and buffering for the B input of the ALU. The sources of latch B input data are: register file B, B multiplexor, instruction register, and 16-bit control store literal. Latch B buffers the output of the register file B from the ALU input during a file writing operation. The following are the types of data that can be loaded into latch B along with corresponding enabling signals from latch B control:

a. Register file B data DFBn– (0-15) or B multiplexor data DMXn– (0-15) are loaded into latch B with a high DLBGS + and DLBGSA + . The data selection is determined by latch B control (DENFB– or DLBMX– and DLBMXA–).

b. Contents of the instruction register C2In– (0-15) are loaded into latch B with a high DLBGI + and DLBGIA + .

c. The 16-bit control store literal (fields B, SH, X, W, V, SC, WR, C, and M) is loaded into latch A with a high DLBGM + and DGLB1– .

DFLCC is a timing signal from the file control circuit. A low DFLCC enables latch B to sample and select the appropriate input data. During the latter portion of the processor cycle, DFLCC + goes high causing latch B to disregard further changes to its other inputs.

4.4.8 B Multiplexor

When specified by the LB field, the B multiplexor provides input data to latch B. As illustrated in the block diagram of figure 4-13, the B multiplexor receives eight data inputs (a through h), LB,-field enabling signals (DLBMX– and DLBMXA–), and B-field data-selection signals DFBAn + (0-2) and DFBAnA + (0-2). The B multiplexor output data DMXn– (0-15) is applied to latch B.

The eight data inputs and their sources are listed as follows:

a. Operand register DORn + (0-15).

b. Memory input latch MILn + (0-15).

c. I/O register IORn + (0-15).

d. Status word consisting of key register bits DCKn + (12-15), shift-counter bits DSCn + (0-4), supervisor-key flag CESK + , and arithmetic flags (DCNDC + , DSGN + , DEQ + , DOVF + , and DCNOZ +).

e. Right byte of operand register DORn + (0-7) with the sign bit DOR07A + extended. The operand register right byte remains in the right byte position of the B multiplexor output, and the sign bit is placed in all bit positions of the left byte.

f. Left byte of operand register DORn + (8-15) with the sign bit DOR15A + extended. The operand register left byte is placed in the right byte position of the B multiplexor output, and the sign bit DOR15A + is placed in all bit positions of the left byte.

g. Right byte of the operand register DORn + (0-7) without the sign bit extended. The operand register right byte is placed in the right byte position of the B multiplexor output, and a zero (GRD) is placed in all bit positions of the left byte (DMX08– through DMX15– are high).

h. Right byte of the operand register DORn + (0-7) and zero data (GRD). The operand register right byte is placed in the left byte of the B multiplexor output, and a zero (GRD) is placed in all bit positions of the right byte (DMX00– through DMX07– are high).

Data input selection is accomplished with bits 0 through 2 of the B field. B-field codes and corresponding data inputs they select are listed in table 4-10.

4.4.9 Arithmetic and Logic Unit

The ALU performs logical or arithmetical operations on data from latches A and B. The ALU output data DALn + (0-15) are directed to the following sections of the processor:

a. The data multiplexor in the memory control for writing the ALU data into memory.

b. The address multiplexor in the memory control for applying the ALU data into the address latch.

c. The I/O multiplexor in the I/O data loop for programmed and multiplexor channel I/O operations through the I/O register.

d. The program counter to provide jump instruction addresses.

e. The shift counter for initializing the shift count.

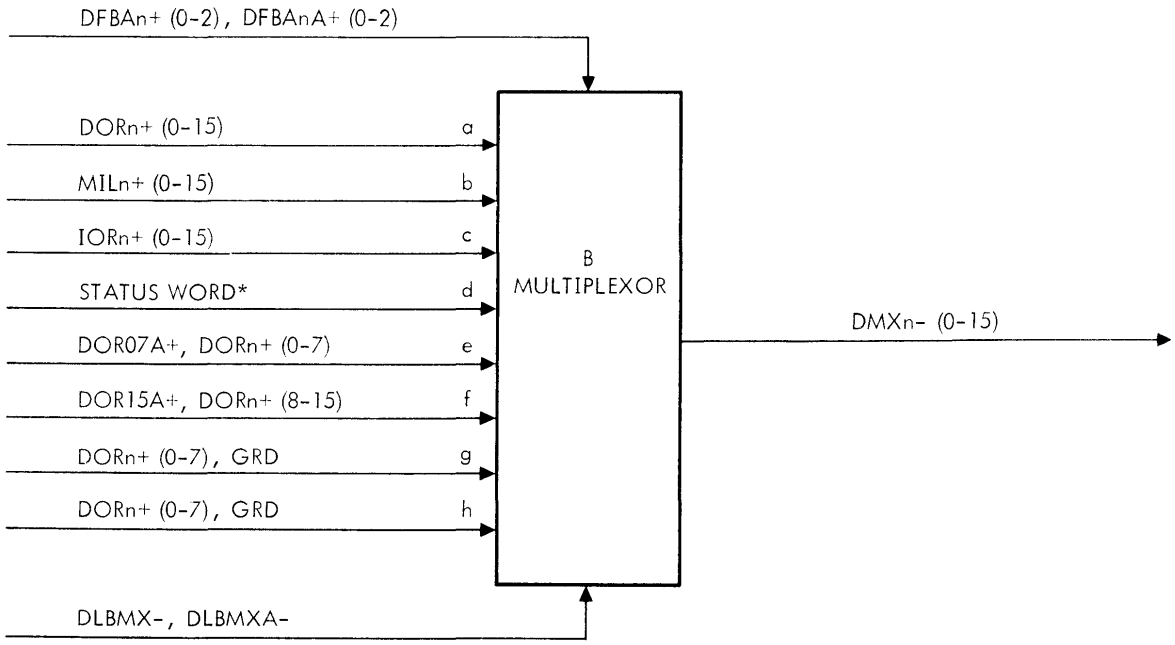
f. The operand register for general data loop operations and those requiring double length shifts.

g. The register files A and B for updating the contents of the registers.

h. The key register for use with memory map operations.

Figure 4-14 is a block diagram of the ALU. The ALU performs logical operations when DMODE + is high and arithmetical operations when it is low. During an arithmetical operation, a low carry-in signal (DCINS–) adds a 1 to the ALU output. Internal carry generation and propagation signals, DSGn + (1-4) and DSPn + (1-4), are applied to the carry generator to determine whether internal carry signals DC0n + (1-3) are to be generated. Function control signals (DFUN0S + , DFUN1S + , DFUN2 + , DFUN3 +) comprise 16 4-bit codes that determine the particular logical or arithmetical ALU operation. Table 4-11 lists the 16 logical and 16 arithmetical operations that can be performed by the ALU.

When the ALU output bits are all ones a high DALEQ + is applied to the status and test logic. A low DCOUT– is applied to the status and test logic when the ALU generates a carry.



*STATUS WORD CONSISTS OF KEY-REGISTER BITS (DCKn+, 12-15), SHIFT-COUNTER BITS (DSCn+, 0-4), SUPERVISOR-KEY FLAG (CESK+), AND ARITHMETIC FLAGS (DCNDC+, DSGN+, DEQ+, DOVF+, AND DCNOZ+).

VTII-1616

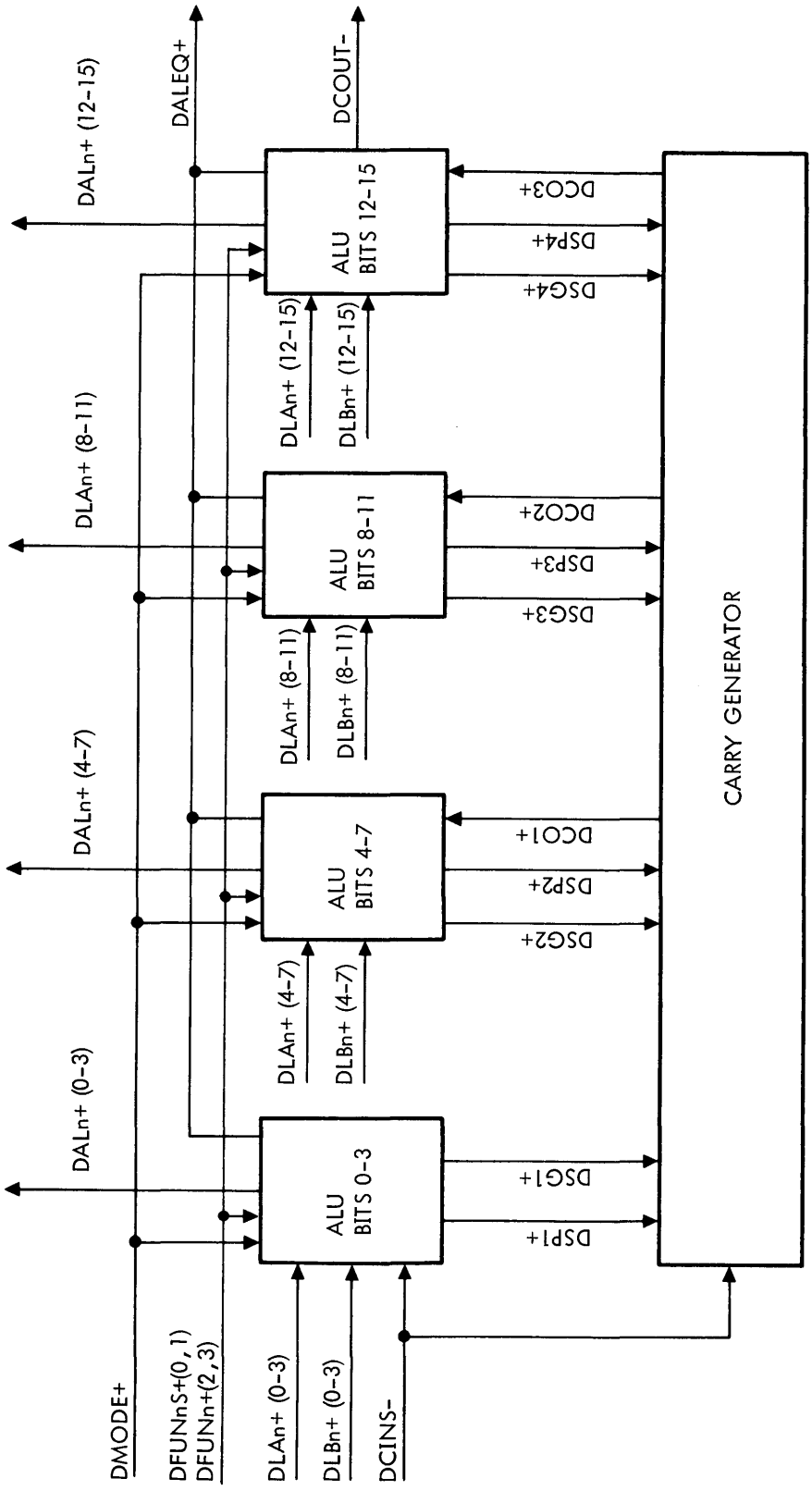
Figure 4-13. B-Multiplexor Block Diagram

Table 4-10. B-Field Codes

DFBA2+ DFBA2A+	DFBA1+ DFBA1A+	DFBA0+ DFBA0A+	SELECTED DATA INPUT*
L	L	L	a
L	L	H	b
L	H	L	c
L	H	H	d
H	L	L	e
H	L	H	f
H	H	L	g
H	H	H	h

NOTE: H = HIGH, L = LOW

*DATA INPUTS ARE DESIGNATED a THROUGH h AS SHOWN IN FIGURE 4-13



VTII-1618A

Figure 4-14. ALU Block Diagram

Table 4-11. ALU Operations

DFUN3+	DFUN2+	DFUN1S+	DFUN0S+	LOGICAL OPERATIONS DMODE+ HIGH	ARITHMETICAL OPERATIONS DMODE+ LOW	
					DCINS- HIGH	DCINS- LOW
L	L	L	L	F = $\overline{A}$	F = A	F = A PLUS 1
L	L	L	H	F = $\overline{A \vee B}$	F = A∨B	F = (A∨B) PLUS 1
L	L	H	L	F = $\overline{A} \wedge B$	F = A∨$\overline{B}$	F = (A∨$\overline{B}$) PLUS 1
L	L	H	H	F = 0	F = MINUS 1 (2's COMPLEMENT)	F = ZERO
L	H	L	L	F = $\overline{A \wedge B}$	F = A PLUS A	F = A PLUS A∧B PLUS 1
L	H	L	H	F = $\overline{B}$	F = (A∨B) PLUS A	F = (A∨B) PLUS A∧$\overline{B}$ PLUS 1
L	H	H	L	F = A∀B	F = A MINUS B MINUS 1	F = A MINUS B
L	H	H	H	F = A∧$\overline{B}$	F = A∧B MINUS 1	F = A∧$\overline{B}$
H	L	L	L	F = $\overline{A}$∨B	F = A PLUS A	F = A PLUS A∧B PLUS 1
H	L	L	H	F = $\overline{A \forall B}$	F = A PLUS B	F = A PLUS B PLUS 1
H	L	H	L	F = B	F = (A∨$\overline{B}$) PLUS A	F = (A∨$\overline{B}$) PLUS A∧B PLUS 1
H	L	H	H	F = A∧B	F = A∧B MINUS 1	F = A∧B
H	H	L	L	F = 1	F = A PLUS A	F = A PLUS A PLUS 1
H	H	L	H	F = A∨$\overline{B}$	F = (A∨B) PLUS A	F = (A∨B) PLUS A PLUS 1
H	H	H	L	F = A∨B	F = (A∨$\overline{B}$) PLUS A	F = (A∨$\overline{B}$) PLUS A PLUS 1
H	H	H	H	F = A	F = A MINUS 1	F = A

NOTES:

1. IN THE ABOVE TABLE, F REPRESENTS THE ALU OUTPUT (DAL00+ THROUGH DAL15+), AND A AND B REPRESENT THE ALU INPUTS (DLA00+ THROUGH DLA15+, AND DLB00+ THROUGH DLB15+).
2. THE SYMBOL ∨ DESIGNATES THE INCLUSIVE ØR OPERATION.
3. THE SYMBOL ∀ DESIGNATES THE EXCLUSIVE ØR OPERATION.
4. THE SYMBOL ∧ DESIGNATES THE AND OPERATION.
5. H = HIGH, L = LOW

VTII-1619

4.4.10 ALU Control

The ALU control contains two multiplexor circuits which provide the ALU control signals. One multiplexor is controlled by DGLB1 +, the other by DSPF +. DGLB1 + is high when the LB field contains binary 10 or 11, indicating signals from the control store buffer are used for masking instead of controlling. A high DSPF + indicates a special ALU function used for the register transfer and modification instructions that have codes beginning with 005. DSPF + goes high when the SH field contains binary 100 through 111, and the LA and LB fields contain 00 or 01. The ALU control signals consist of the ALU mode control DMODE +, ALU carry-in DCINS–, and ALU function controls DFUN0S + and DFUNS1 +.

DMODE + is controlled by the M-field bit (DGMD +) in the non-masked condition (DGLB1 + low), and by bit 1 of the F field (DFUN1 +) in the masked condition (DGLB1 + high). DCINS– is controlled by instruction register bits 6 and 7 (C2I06D + and C2I07D +) for the special ALU function (DSPF + high). During the normal ALU function (DSPF + low) in the non-masked condition, DCINS– (carry input) is controlled by the C field as follows:

a. When the C field contains 00 (DCRY0 + and DCRY1 + low), DCINS– goes high indicating there is no carry input.

b. When the C field contains 01 (DCRY0 + high and DCRY1 low), DCINS– goes low if DCNDC + (stored carry) is high and high if DCNDC + is low. This makes the carry in equal to the stored carry.

c. When the C field contains binary 10 (DCRY0 + low and DCRY1 + high), DCINS– goes low if DCNDC + is low and high if DCNDC + is high. This makes the carry in equal to the complement of the stored carry.

d. When the C field contains a binary 11 (DCRY0 + and DCRY1 + high), DCINS– goes low indicating a carry input.

DFUN0S + and DFUN1S + are controlled by instruction-register bit 7 (C2I07D– and C2I07D +) for the special ALU functions, and by F-field bits 0 and 1 for the normal ALU function.

In addition to the ALU control signals, DEKWCF +, DEKSB +, and DOVSMS + are also generated by the ALU control. In the non-masked condition, DEKWCF + is controlled by the WR-field bit (DWCF +) and DEKSB + by the V-field bit (DV +). DEKWCF + is sent to the file control for writing data into files A and B, and DEKSB + to the shift control to enable the setting of the dividing-sign flip-flop (DSB +). DOVSMS + is an enabling signal for setting the overflow flip-flop in the status and test logic. This signal is controlled by instruction-register bit 6 (C2I06D +) for the special ALU function, and by CSAMOV + for the normal ALU function.

4.4.11 Register Control

A decoder circuit in the register control decodes the three R-field bits DGRCn + (0-2) to produce seven control signals that control various operations for the shift counter, program counter, operand register, and key register. Table 4-12 is a truth table showing the states of the seven control signals for each R-field code. Control signals DRC3–, DRC4–, and DRC7– are used in the register control to produce D3RC7– and DR47 +. When the R field contains 011 or 111, a low D3RC7– loads the operand register.

Table 4-12. Register-Control Decoder Truth Table

R-FIELD INPUT BITS				DECODED OUTPUTS						
GROUND	DGRC2+	DGRC1+	DGRC0+	DRC1–	DRC2–	DRC3–	DRC4–	DRC5+	DRC6–	DRC7–
L	L	L	L	H	H	H	H	L	H	H
L	L	L	H	L	H	H	H	L	H	H
L	L	H	L	H	L	H	H	L	H	H
L	L	H	H	H	H	L	H	L	H	H
L	H	L	L	H	H	H	L	L	H	H
L	H	L	H	H	H	H	H	H	H	H
L	H	H	L	H	H	H	H	L	L	H
L	H	H	H	H	H	H	H	L	H	L

NOTE: L = LOW, H = HIGH

VTII-1620

When the R field contains 100 or 111, a high DR47+ increments the program counter at half-clock time. The R field specifies the following operations:

a. R-field containing 000 causes no operation.

b. R-field containing 001 loads the program counter.

c. R-field containing 010 loads the shift counter.

d. R-field containing 011 loads the operand register.

e. R-field containing 100 increments the shift counter.

f. R-field containing 101 loads the key register.

g. R-field containing 110 increments the program counter.

h. R-field containing 111 loads the operand register and increments the program counter.

A byte-address flip-flop (DBAD+) stores file-A bit 0 (DFA00−) at half-clock time (MCDHC+) to prevent losing the bit during shifting operations (DGLA1 + high).

The processor clock KKC2+ is repowered to provide clocks for the operand register and shift counter. These clocks are DKOR− and DKSCR−. Clock MCDFC− is also repowered to produce the data loop clock DMFC+. The data-loop clock is also used to generate the program-counter clock DKPR−. The program-counter clock can occur at either full-clock time (MCDFC−) to load the program counter or at half-clock time to increment the program counter.

4.4.12 Shift Control

The shift control contains two multiplexors that select the data to be shifted into the end bits of the operand register during shifting operations. Both multiplexors are controlled by the X field. The shift-left multiplexor selects data (DSL00+) for bit 0 of the operand register from bit 15 of the ALU, file A, or operand register. The shift-right multiplexor selects data (DSR15+) for bit 15 of the operand register from bits 0 or 15 of the operand register, bit 0 of file A, or the divide sign bit (stored file-A bit 15, DSB+). Tables 4-13 and 4-14 are truth tables for the shift-left and shift-right multiplexors.

4.4.13 Operand Control

The operand control selects the operating mode of the operand register based on the SC and W fields. The four operating modes consist of: hold (no activity), shift left, shift right, and load in ALU data. Table 4-15 lists the operating modes resulting from the various states of output signals DOS0+ and DOS1+.

Table 4-13. Shift-Left Multiplexors Truth Table

CONTROL		INPUT DATA				OUTPUT
DX1+	DX0+	DOR15+	DFA15D+	DAL15−	GROUND	DSL00+
L	L	L	X	X	X	L
L	L	H	X	X	X	H
L	H	X	L	X	X	L
L	H	X	H	X	X	H
H	L	X	X	L	X	L
H	L	X	X	H	X	H
H	H	X	X	X	L	L

NOTE: L = LOW, H = HIGH, AND X = IRRELEVANT

VTII-1621

Table 4-14. Shift-Right Multiplexors Truth Table

CONTROL		INPUT DATA				OUTPUT
DX1+	DX0+	DOR00+	DFA00D+	DOR15+	DSB+	DSR15+
L	L	L	X	X	X	L
L	L	H	X	X	X	H
L	H	X	L	X	X	L
L	H	X	H	X	X	H
H	L	X	X	L	X	L
H	L	X	X	H	X	H
H	H	X	X	X	L	L
H	H	X	X	X	H	H

NOTE: L = LOW, H = HIGH, AND X = IRRELEVANT

VTII-1622

Table 4-15. Operand Register Modes

INPUTS				OUTPUTS		MODE
D3RC7-	DW+	DGLB1-	DSCON+	DOS0+	DOS1+	
H	X	X	X			
H	X	X	L	L	L	HOLD
H	L	H	H	L	H	SHIFT LEFT
H	H	H	H	H	L	SHIFT RIGHT
L	X	X	X	H	H	LOAD IN ALU DATA

NOTE: L = LOW, H = HIGH, AND X = IRRELEVANT

VTII-1623

4.4.14 Operand Register

The 16-bit operand register has load and shift capabilities to enable double length shifts in conjunction with shifting modes of latch A. The register can be loaded with data from the ALU, and its output DORn + (0-15) can be selected by the B multiplexor as an ALU input. The four operating modes of the operand register are given in the preceding section.

The operand register consists of four 4-bit shift registers. Loading and shifting operations occur on the positive-going transition of clock DKOR- (unless DOS0 + and DOS1 + are both low).

4.4.15 Key Register

The 4-bit key register is loaded from the four most significant bits of the ALU output under control of the R field (DRC6-). The register is used with a memory map to designate which one of 16 memory protection partitions is active for processor-initiated memory operations.

In addition to the logical high and low states, the key-register output DCKn + (12-15) has an open-circuit state. This allows the output lines to be directly connected to outputs of the I/O key register in the memory control section. The key register output is in the open-circuit state when an I/O request occurs (IRQM + high) or when the supervisor control flip-flop is set (CESK + high). The key register is loaded at full-clock time with the positive-going transition of clock DKSCR- .

4.4.16 Program Counter

The 16-bit program counter performs loading and counting functions to permit address generation of the next instruction to be fetched. Contents of the counter DPRn + (0-15) can be selected as an address source by the address multiplexor for memory control or by latch A for the ALU input.

The counter can be loaded from the ALU or incremented under control of the R field (DRC1- and DR47 +). When DRC1- is low, the positive-going transition of clock DKPR- loads in ALU data at full clock time. When DR47 + is high, the positive-going transition of clock DKPR- increments the counter contents at half-clock time.

4.4.17 Shift Counter

The shift counter, under control of the R field, performs loading and counting functions used for shifting, multiplication, and division. When the desired shift count is reached, a high carry signal DCTZ + is applied to the status and test logic. The five output bits DSCn + (0-4) are applied to the B multiplexor as part of the status word.

When DRC2- is low, ALU data DALn + (0-7) consisting of the two's complement of the required shift count is loaded into the shift counter on the positive-going transition of clock DKSCR- . At full-clock time whenever DRC5 + is high, the counter is incremented by one until a carry is generated from the eighth most significant bit. This causes the counter to be reset and the carry signal DCTZ + to go high. The counter is also reset when KCINMP + goes low.

4.4.18 Status and Test Logic

The status and test logic stores the results of certain operations under control of the control store buffer. Five status signals for the B multiplexor and one for latch A are generated by the status and test logic:

a. The multiplication sign flag DSM1 + can be applied to latch A. During multiplication, DSM1 + goes high when the product is negative.

b. DCNOZ + goes high when the ALU output is zero. This condition is sampled by DTKL- .

c. DCNDC + goes high when the ALU generates a carry. This condition is sampled by DTKL- .

d. DSGN + goes high when the ALU output is negative. This condition is sampled by DTKL- .

e. DEQ + goes high when the ALU output bits are all ones. This condition is sampled by DTKL- .

f. DOVF + goes high when the overflow flip-flop is set.

The status and test logic also applies a test-result signal DTCND + to the central control to determine the control store address during the processing of conditional instructions. DTCND + goes high when the selected test condition is met and low when it is not met. When selected by the G-field signals CBGn + (0-3), the following can be tested:

a. Overflow. When this condition occurs, the overflow flip-flop is set (DOVF + high).

b. I/O sense response. When this condition occurs, a high ISRD + is applied to the status and test logic.

c. Sense switches 1, 2, and 3. When these switches are set, low NSSn- (1-3) signals cause the corresponding CBSSn + (1-3) signals to go high in the status and test logic.

d. Jump, Jump-and Mark, and Execution tests. When a condition is met for one of these instructions, a high DFTT + is generated in the status and test logic.

e. Equal ALU inputs. When the data inputs to the ALU are equal, a high DALEQ + causes DEQ + to go high in the status and test logic.

f. Sign of ALU output. A high DAL15A + produces a high DSGN + in the status and test logic indicating a negative sign. A low DAL15A + produces a low DSGN + indicating a positive sign.

continued

g. ALU carry. When the ALU generates a carry, a low DCOUT– produces a low DCNDC– in the status and test logic.

h. ALU output is zero. When the output of the ALU equals zero, a low DZTS– and a high DCNOZ + are generated in the status and test logic.

i. DSB flag. When bit 15 of register file A is a one, a high DSB + is sent to the status and test logic. A low DSB + indicates bit 15 of register file A is a zero.

j. Sign of memory input latch. A high MIL15 + indicates the data from the memory input latch has a negative sign. A low MIL15 + indicates a positive sign.

k. Shift-counter overflow. When this condition occurs, a high DCTZ + is applied to the status and test logic indicating the shift counter is reset to zero.

l. Output sign of A register (general-purpose register 0). The status and test logic generates a high DASGN + for a negative sign and a low DASGN + for a positive sign.

m. Normalized shift completed. When a normalized shift is complete, the status and test logic generates a high DNZT +. This occurs when the ALU output bits 14 and 15 have opposite states.

4.5 MEMORY CONTROL CIRCUITS

The circuits of the memory control are shown in the block diagram of figure 4-15. Page numbers of the processor logic diagram are provided in parenthesis for each circuit block.

4.5.1 Address Multiplexor

The 4-input 16-bit address multiplexor selects a memory address from one of the following sources:

a. I/O register, IORn + (0-15)

b. ALU, DALn + (0-15)

c. Program counter, DPRn + (0-15)

d. Input latch, MILn + (0-15)

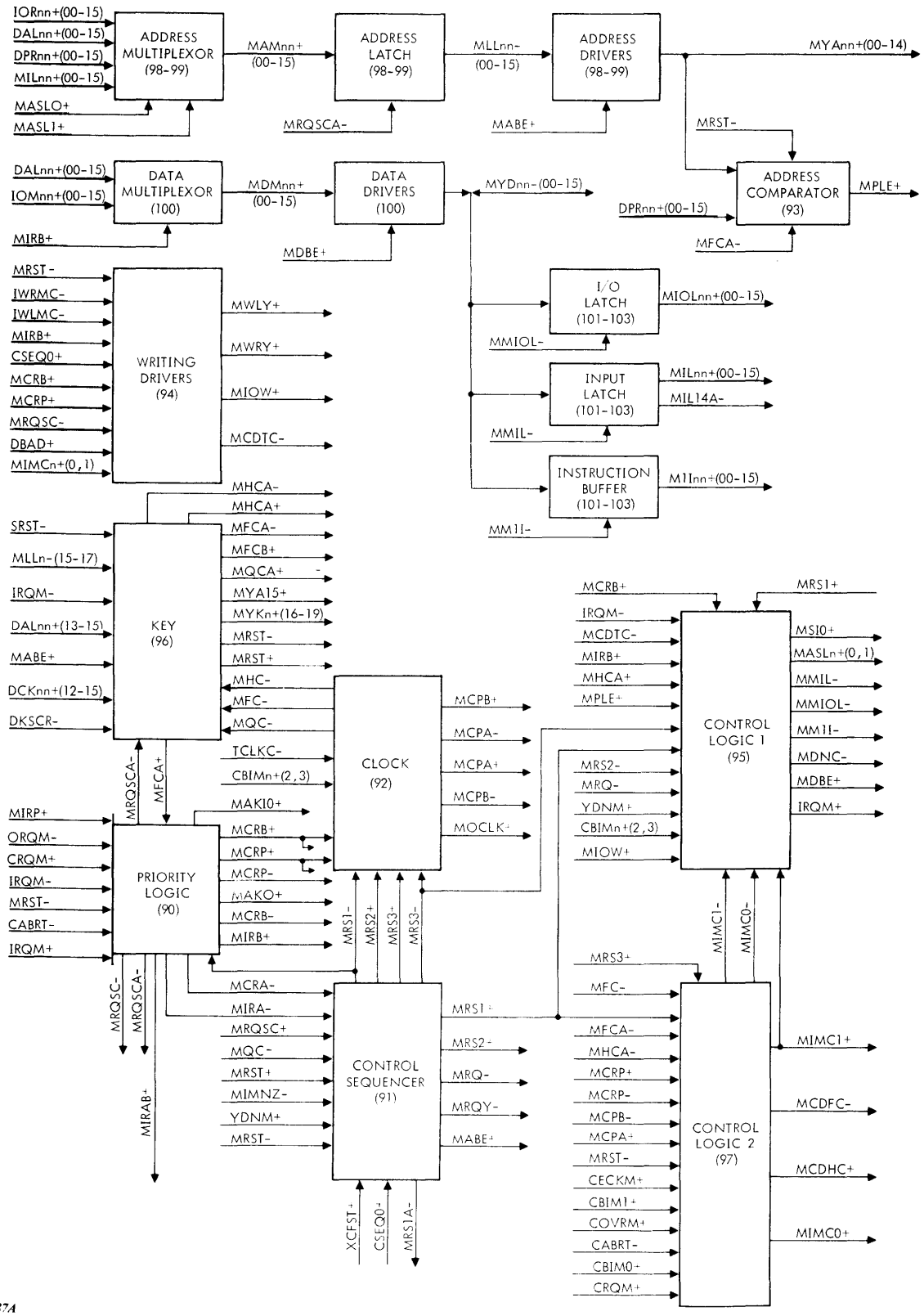
Address selection is controlled by MASL0 + and MASL1 + from control logic 1. The address multiplexor applies the selected address MAMn + (0-15) to the address latch. Table 4-16 is a truth table for the address multiplexor.

Table 4-16. Address Multiplexor Truth Table

CONTROL		INPUT DATA				OUTPUT
MASL1+	MASL0+	IORn+ (0–15)	DALn+ (0–15)	DPRn+ (0–15)	MILn+ (0–15)	MAMn+ (0–15)
L	L	L	X	X	X	L
L	L	H	X	X	X	H
L	H	X	L	X	X	L
L	H	X	H	X	X	H
H	L	X	X	L	X	L
H	L	X	X	H	X	H
H	H	X	X	X	L	L
H	H	X	X	X	H	H

NOTE: L = LOW, H = HIGH, AND X = IRRELEVANT

VTII-1624



VTI3-287A

Figure 4-15. Memory Control Block Diagram

4.5.2 Address Latch

The 16-bit address latch stores the selected memory address during the memory cycle. A low sample clock MRQSCA– from the priority logic loads the selected address into the latch where it is inverted and applied to the address drivers.

4.5.3 Address Drivers

The open-collector address drivers interface the address latch to the memory address bus. A high enabling signal MABE + from the control sequencer gates the latch output MLLn– (0-15) through the drivers where it is inverted and applied to the address bus as MYAn + (0-14) (address bit 15 is not used on the address bus unless a writable control store is present).

4.5.4 Address Comparator

The address comparator prevents erroneous operation in the event a store type instruction is being currently processed and its effective address contains the next instruction to be fetched (the processor fetches the next instruction before the current instruction is completed). Addresses from the program counter DPRn + (0-15) and the address drivers MYAn + (0-15) are compared by the comparator. When the addresses are equal, a high MPLE + signal is sent to control logic 1 so that the instruction already fetched into the instruction buffer can be updated during the store instruction's execution.

4.5.5 Data Multiplexor

The 16-bit data multiplexor selects memory write data from either the ALU or I/O multiplexor for DMA operation. Data selection is controlled by I/O request signal MIRB + from the priority logic. A high MIRB + selects I/O data IOMn + (0-15); a low MIRB + selects ALU data DALn + (0-15).

4.5.6 Data Drivers

The open-collector data drivers interface the data multiplexor to the bidirectional memory data bus. A high enabling signal MDBE + from control logic 1 gates the data multiplexor output MDMn + (0-15) through the drivers where it is inverted and applied to the memory data bus as MYDn – (0-15).

4.5.7 I/O Latch

Data from the memory data bus is loaded into the 16-bit I/O latch with a low MMIOL– from control logic 1. The I/O latch inverts the data and applies it to the I/O multiplexor in the I/O data loop section.

4.5.8 Input Latch

Data from the memory data bus is loaded into the 16-bit input latch with a low MMIL– from control logic 1. The input latch inverts the data and applies it to the address multiplexor and the B multiplexor in the data loop section.

4.5.9 Instruction Buffer

Data from the memory data bus is loaded into the 16-bit instruction buffer with a low MM1I– from control 1 logic. The instruction buffer inverts the data and applies it to the central control section as M1In + (0-15). This data is applied to the instruction decoder and instruction decoding logic for preliminary decoding, and to the instruction register for detailed decoding.

4.5.10 Writing Drivers

The writing drivers apply left- and right-byte controls MWLY + and MWRY + to the memory. When a full word is written into memory, MIMC1 + is high and MIMC0 + is low resulting in high MWLY + and MWRY + signals. When a byte is written into memory, MIMC0 + is high and the stored-byte address DBAD + determines which byte control is high. A high MCRB + enables the byte controls when the processor has priority, and a high MIRB + enables them when the I/O control has priority. IWLMC– and IWRMC– provide left- and right-byte information from the I/O control. A high MIOW indicates I/O data is written into memory.

When the processor has priority and the S field equals 0 (MCRP + and CSEQ0 + are high), MCDTC– goes low on the positive-going transition of clock MRQSC– indicating that a special data transfer from the ALU to the instruction buffer is taking place.

4.5.11 Key Logic

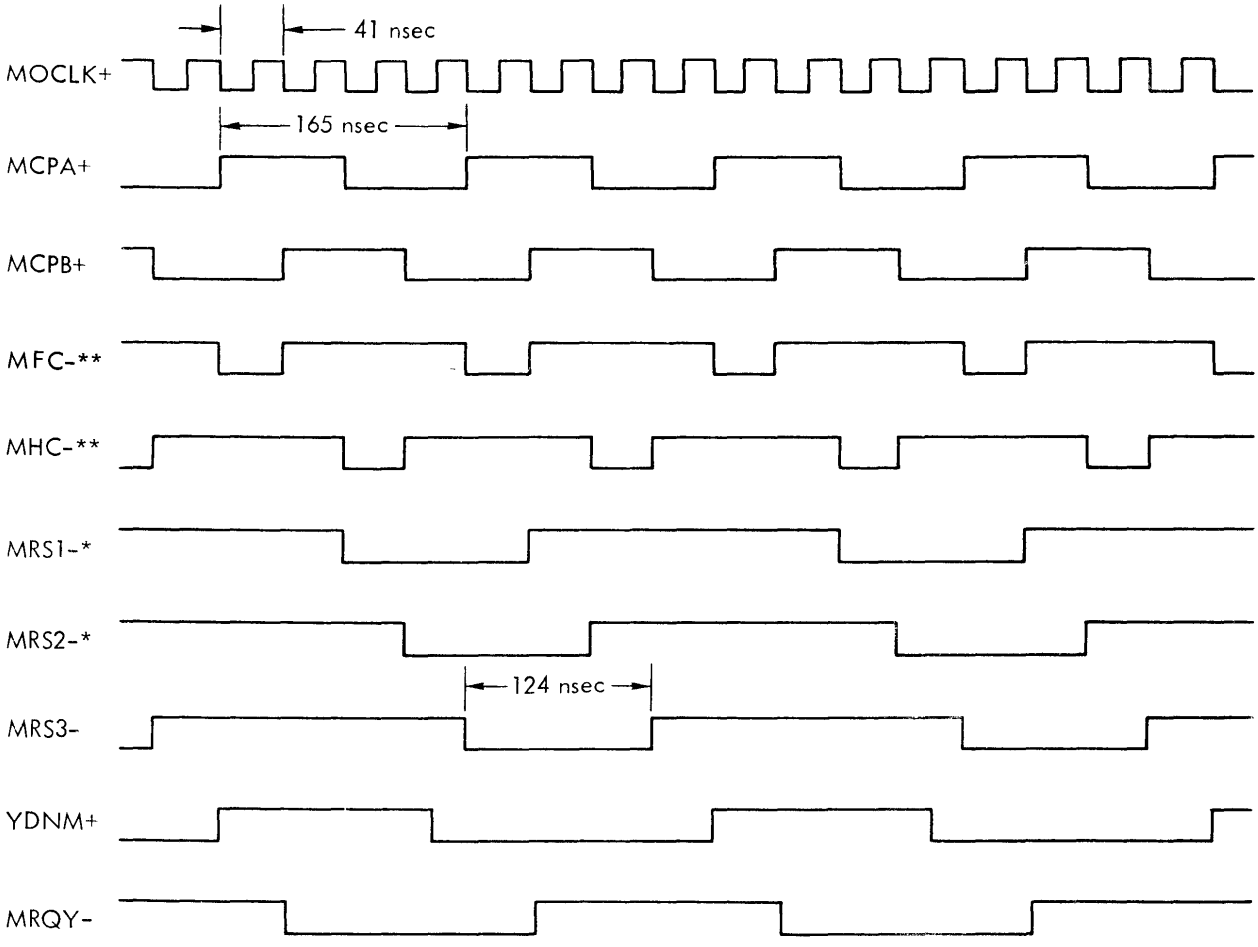
The key logic provides key bus data MYKn + (16-19) for memory mapping, and repowered clock and reset signals. The key bus data comes from either an I/O key register in the key logic or from the key register in the data loop. When CIMXX0– and CIM2S0– are low ALU data bits 12 through 15 are loaded into the I/O key register on the positive-going transition of clock DKSCR–. The key register output is enabled with an I/O request (IRQM– low). Key-bus data (I/O or processor) are gated onto the key bus with a high MABE +.

4.5.12 Clock Generator

The clock generator provides clock signals (figure 4-16) for all sections of the processor, except I/O control, and for options on the option board. A crystal oscillator circuit produces the symmetrical squarewave MOCLK + (24.2424 MHz). This 41.25-nanosecond period clock is sent to the

option board and is also used to clock flip-flops that produce the phase A and B clocks MCPA + and MCPB +. The phase clocks have a 165-nanosecond period representing the basic processor control state in which a single microinstruction is processed. Full and half clocks MFC– and MHC– are produced by decoding MCPA + and MCPB + The periods of MFC– and MHC– may vary in 41 nanosecond increments depending on when the memory acknowledgment signal YDNM + is applied to the control sequencer. During indirect addressing and DMA operations, the periods are increased an additional 41 nanoseconds (one MOCLK + cycle) by a delay flip-flop in the clock generator.

The crystal oscillator circuit can be disconnected (by removing a jumper on the processor card) and replaced with an external signal (TCLKC–) connected to J2-43 of the processor board. This external signal is normally supplied by the writable control store option.



*PULSE WIDTH DEPENDS ON WHEN YDNM+ GOES LOW

**PERIOD MAY VARY IN 41 NANOSECOND INCREMENTS DEPENDING ON RELATIONSHIP OF YDNM+ GOING LOW AND PHASE CLOCKS MCPA+ AND MCPB+. DURING INDIRECT ADDRESSING AND DMA OPERATIONS, THE PERIOD IS INCREASED AN ADDITIONAL 41.25 NANOSECONDS

VTII-1625

Figure 4-16. Clock and Sequencer Waveforms

4.5.13 Control Sequencer

The control sequencer generates sequencing signals when a memory request is received from the processor or I/O. A low MCRA– (processor request) or MIRA– (I/O request) from the priority logic produces a high MABE + that gates the memory address onto the address bus. This occurs at the beginning of the sequencer cycle so that the address has time to stabilize.

A three flip-flop counter clocked by MQC– and MQC + generates the three sequencing signals MRS1–, MRS2–, and MRS3– (figure 4-16). The pulse width of MRS3– is always 124 nanoseconds; however, MRS1– and MRS2– pulse widths depend on when the memory acknowledgment signal is received (YDNM + goes low).

4.5.14 Priority Logic

The priority logic receives requests for memory operations from the PMA (on option board), I/O control and central control. PMA has the highest priority and central control has the lowest.

When the PMA requests a memory operation, a low ORQM– is applied to the priority logic. This inhibits the sampling of requests from I/O control and central control, and causes MAKO + to go high indicating the PMA has priority.

An I/O control request indicates a DMA operation is to be performed and applies a high IRQM + to the priority logic. A high IRQM + inhibits the sampling of central control requests and, if there is no PMA request, causes MIRB + to go high indicating the I/O control has priority.

A central control request indicates that the processor is requesting a memory operation. When a central control request occurs, high CRQM + and CABRT– signals are applied to the priority logic. If there are no requests from the PMA or I/O control, MCRB + goes high indicating the central control has priority.

4.5.15 Control Logic 1

The control logic 1 produces various control and enabling signals.

Signals MASL1 + and MASL0 + are sent to the address multiplexor to control the address selection. The states of these signals are determined by the I/O control request IRQM– for I/O memory requests or by bits 2 and 3 of the IM field for processor memory requests.

Low MMIL–, MMIOL–, and MM1I– signals load data into the input latch, I/O latch, and instruction buffer, respectively. In addition to other requirements, one of the following two conditions is required before the three loading signals can go low: the memory acknowledgment signal YDNM + is low, or a special data transfer occurs from the ALU to the instruction buffer and input latch (MHCA +, MRS3–, and MRQ– high).

MMIL– goes low when the processor has priority (MCRB + and MRS1– high) and an input-latch loading operation is specified (MIMC1– high). MMIOL– goes low when the I/O control has priority (MIRB + and MRS1– high). MM1I– goes low when IM-field bits 0 and 1 are both zero (MIMC1– and MIMC0– high) and the processor has priority (MCRB + and MRS1– high). Another condition that causes MM1I– to go low is when the contents of the program counter and the memory address are equal, and MIMC1 + is high.

A low MDNC– is applied to clock control circuits in the central control to inhibit processor clocks until a memory acknowledgment is received if a wait for memory done is specified. MDNC– goes high to disable the clocks when the processor has priority (MCRB + high) and the control sequencer is in state 2 (MRS2 + high).

A high MDBE + is applied to the data drivers to transfer data onto the memory data bus. MDBE + goes high when one of the following conditions occur:

a. I/O control has priority (MIRB + high) and an I/O writing operation is specified (MIOW + high).
b. Processor has priority (MCRB + high), and a processor writing operation is specified (MIMC1– low) or the special data transfer from the ALU to the instruction buffer is specified (MCDTC– low).

4.5.16 Control Logic 2

The control logic 2 stores IM-field data and generates processor clocks. Two flip-flops (MIMC0 and MIMC1) store the contents of the IM-field bits 0 and 1 when the processor is requesting a memory operation (COVRM + and CABRT– are high for a memory request, or MRS1 + and MRS3 + are low for an override condition). The flip-flip outputs are used by the writing drivers and control logic 1 circuits.

Full and half clock signals (MCDFC– and MCDHC +) for the processor are produced by decoding appropriate states of phase A and B clock (MCPA and MCPB), and are inhibited if one of the following conditions occur:

a. The processor has priority but the memory is busy (MCRP + and MRS2 + are high).

b. The processor is denied memory access because of a higher priority request (MCRP–, CRQM +, and CABRT– are high).

c. Central control generates a low clock-disabling signal (CECKM– low).

After one of the above disabling conditions occur, a delay flip-flop allows an additional MCDHC + pulse to be generated (figure 4-17).

4.6 I/O DATA LOOP

The circuits of the I/O data loop are shown in the block diagram of figure 4-18. Page numbers of the processor logic diagram are provided in parentheses for each circuit block.

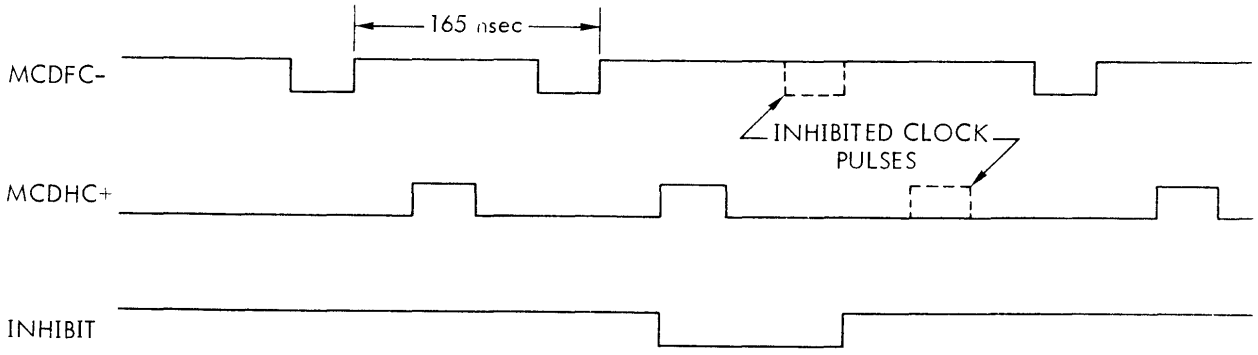


Figure 4-17. Processor Clock Inhibiting

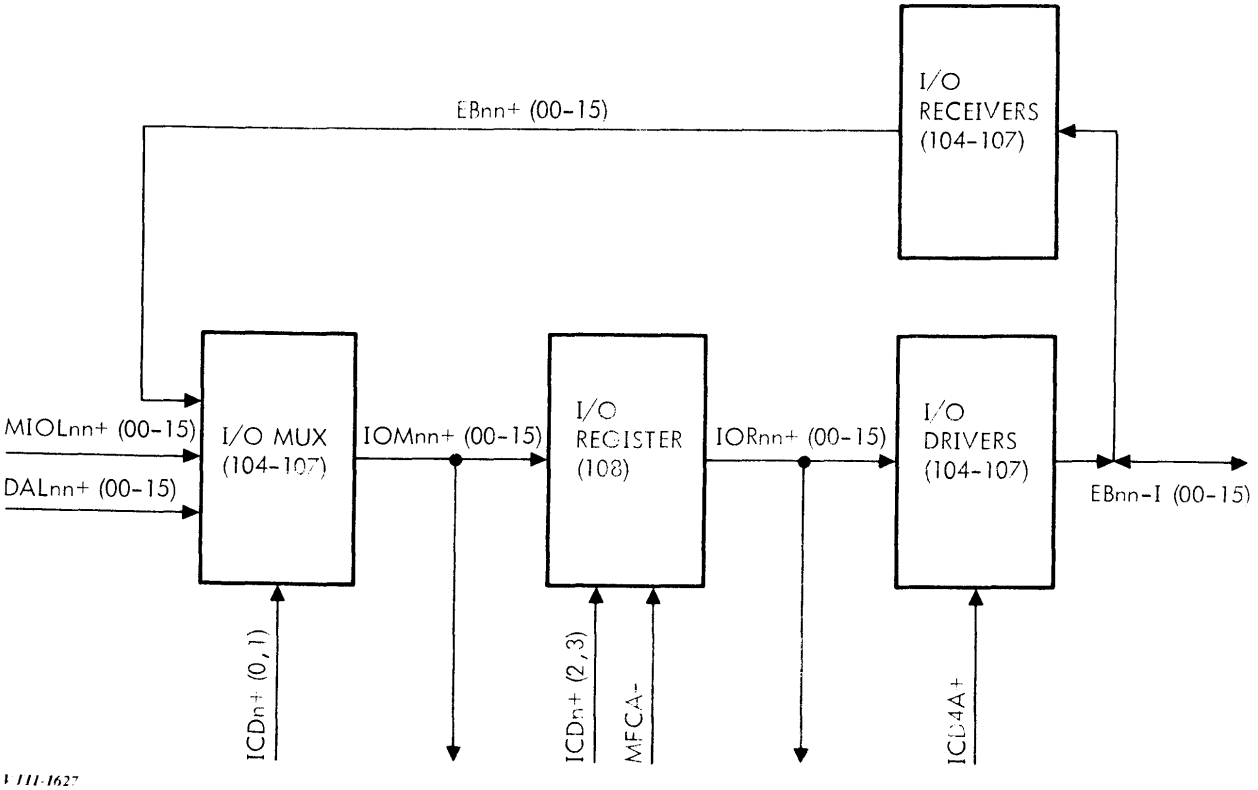


Figure 4-18. I/O Data Loop Block Diagram

4.6.1 I/O Multiplexor

The I/O multiplexor, which consists of eight dual 4-to-1 multiplexor circuits (figure 4-19), selects data for the I/O register and for the data multiplexor in memory control. The data is selected from the following sources:

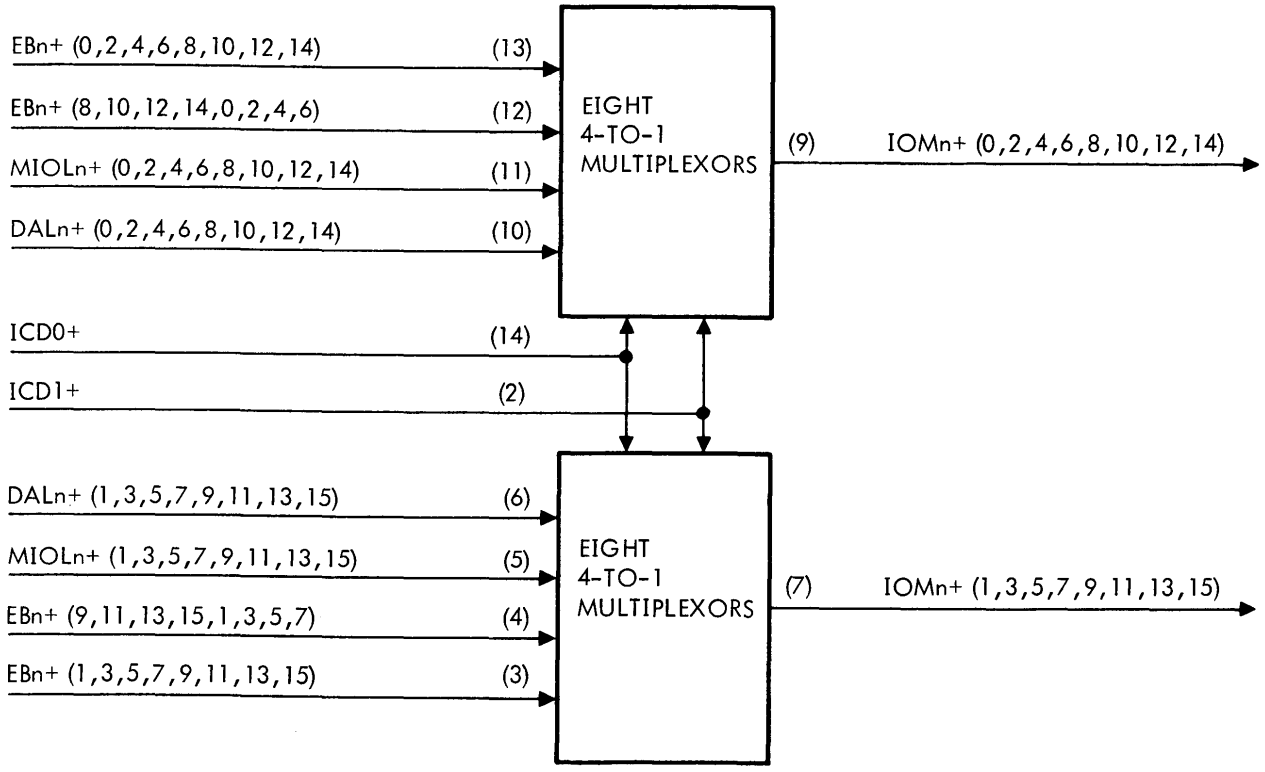
a. I/O latch, MIOLn + (0-15)

b. ALU, DALn + (0-15)

c. I/O receivers, EBn + (0-15)

Data from the I/O receivers are applied to the I/O multiplexor in normal and byte-shifted forms, that is, each of the multiplexor output bits can be selected from corresponding bits in either byte of the I/O data. For example, multiplexor output bit 0 can be selected from I/O data bits 0 or 8, the output bit 1 from bits 1 or 9, etc.

Selection of input data to the multiplexor is controlled by I/O control signals ICD0 + and ICD1 +. Table 4-17 is a truth table for the I/O multiplexor.



NOTE: IC PIN NUMBERS ARE IN PARENTHESES

VTII-1628

Figure 4-19. I/O Multiplexor Block Diagram

Table 4-17. I/O Multiplexor Truth Table

CONTROL		INPUT DATA				OUTPUT
PINS		PINS				PINS
2	14	6/10	5/11	4/12	3/13	7/9
L	L	L	X	X	X	L
L	L	H	X	X	X	H
L	H	X	L	X	X	L
L	H	X	H	X	X	H
H	L	X	X	L	X	L
H	L	X	X	H	X	H
H	H	X	X	X	L	L
H	H	X	X	X	H	H

NOTES: 1. L = LOW, H = HIGH, AND X = IRRELEVANT

2. REFER TO FIGURE 4-19 FOR THE SIGNALS ASSOCIATED WITH PIN NUMBERS IN THE ABOVE TABLE

VTII-1629

4.6.2 I/O Register

The 16-bit I/O register stores data from the I/O multiplexor and applies it to the I/O drivers, to the B multiplexor in the data loop, and to the address multiplexor in memory control. Data is clocked into the register on the positive-going transition of full clock MFCA-- or at MHCA + when the I/O requests a memory cycle (IRQM + high). The I/O register can operate in four modes that are selected by I/O control signals ICD2 + and ICD3 +. A hold mode inhibits the effect of the clock so that the register outputs are unchanged. On a single clock transition, byte transfer modes transfer the left byte to the right byte position and the right byte to the left byte position (see figure 4-20). A loading mode loads data selected by the I/O multiplexor into the register. Table 4-18 lists the operating modes resulting from the various states of ICD2 + and ICD3 +.

Table 4-18. I/O Register Modes

ICD2 +	ICD3 +	Mode
L	L	Hold
L	H	Left Byte to Right Byte
H	L	Right Byte to Left Byte
H	H	Load in multiplexor data

4.6.3 I/O Drivers and Receivers

Data is transferred to and from the bidirectional I/O data bus with I/O drivers and receivers. These drivers and receivers also invert the data.

When enabled with a high ICD4A + signal from the I/O control, the I/O drivers transfer data from the I/O register to the I/O data bus.

The I/O receivers apply I/O bus data to the I/O multiplexor.

4.7 I/O CONTROL

This section consists of circuit descriptions followed by the various I/O operations. The I/O circuits are grouped as follows:

a. Microinstruction Sequencing

b. Decoding and Sequencing Control

c. Interrupt and DMA Control

d. I/O Data-Loop Control

e. Processor-I/O Control

f. Drivers, Receivers, and Miscellaneous Control

The I/O operations consist of:

a. Programmed I/O

b. Interrupt

c. Normal DMA

d. High-Speed DMA

The I/O Control circuits are located on the option board (44P0619). Logic diagrams for these circuits are on sheets 3.0 through 3.11 of drawing 91B0401 (provided in the System Maintenance Manual). Page numbers of the option

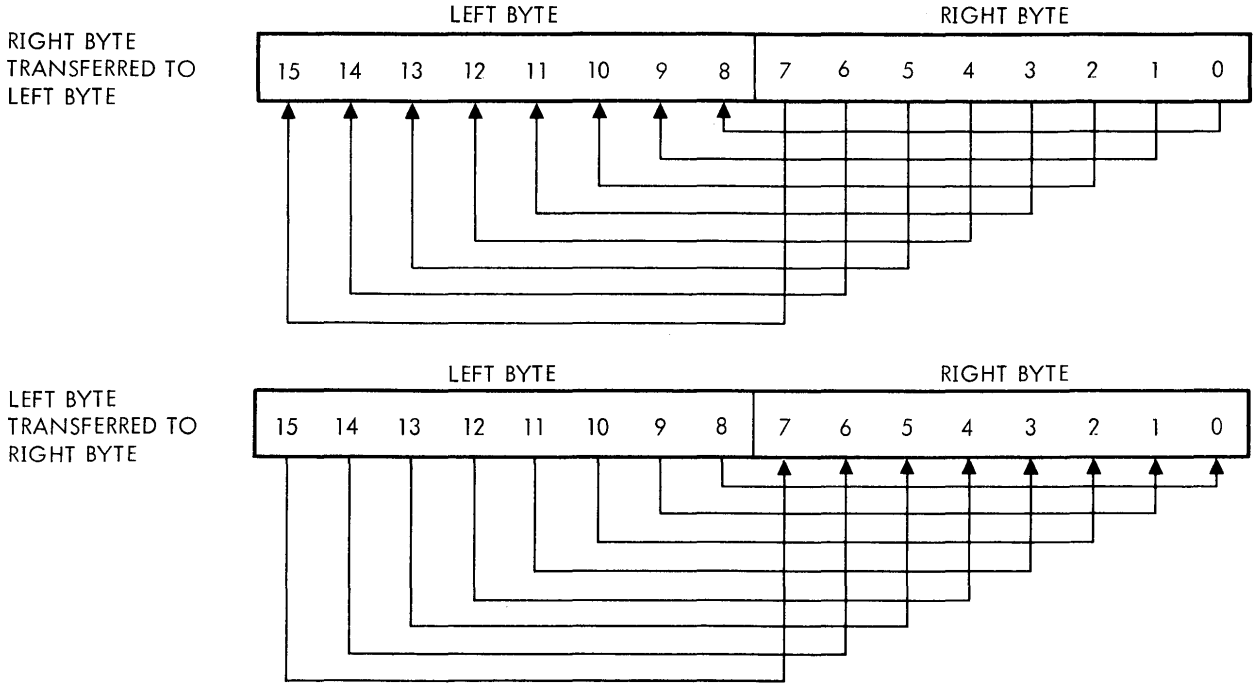


Figure 4-20. Byte-Transfer Modes of I/O Register

board logic diagrams are provided in parentheses in the block diagrams that follow. For a description of the external I/O interface signals, refer to the Input/Output section of the applicable system handbook.

4.7.1 Microinstruction Sequencing

All I/O operations are performed through a sequence of microinstructions. Implementation of these sequences is accomplished with the address selector, address counter, I/O control store, and buffer. Figure 4-21 shows how these circuits are related to each other. A description of each circuit is provided in the following subsections.

4.7.1.1 Address Selector

Under control of the control logic, the address selector selects an I/O control-store address to initiate an I/O operation. The address is selected from either central control or the trap-address generator (for DMA operations). A low IIIDLE– enables the address selector. The address selector selects an address from central control when IMXAD + is high and from the trap-address generator when IMXAD + is low. The selected 8-bit address is applied to the address counter.

4.7.1.2 Address Counter

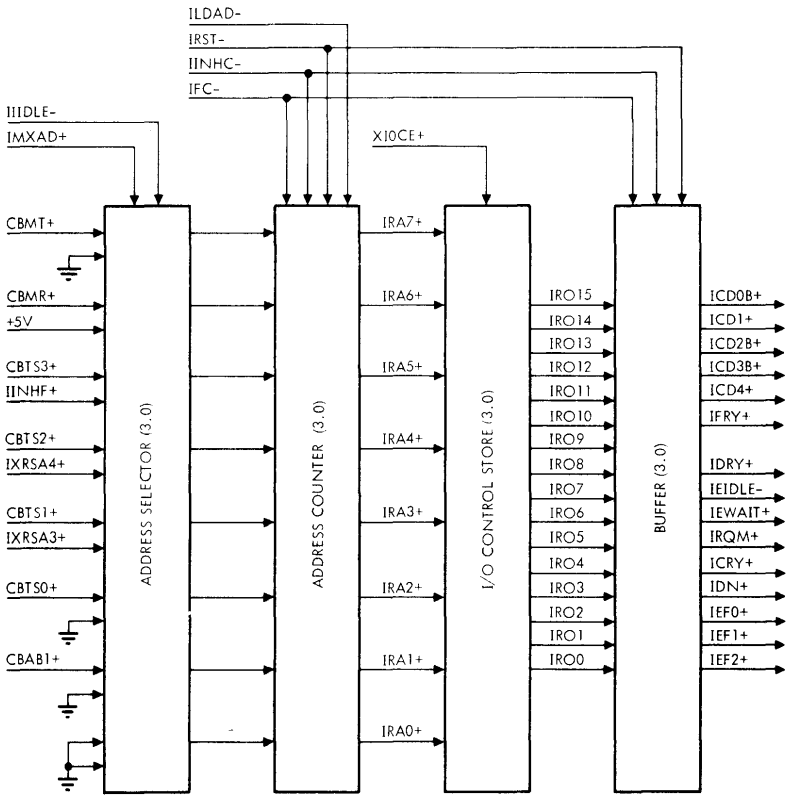
The address counter provides sequential addresses for the I/O control store. An I/O operation begins when the selected address (from the address selector) is loaded into the address counter with the positive-going transition of IFC– (when ILDAD– is low). The counter is then incremented with each positive-going transition of IFC– (when IINHC– is high) to provide the sequential addresses for an entire I/O operation. The next to last microinstruction of any I/O operation produces a low IEIDLE– to prevent further incrementing of the address counter.

4.7.1.3 I/O Control Store

The I/O control store is a 256-word by 16-bit high-speed ROM (figure 4-22). It contains the microinstructions that control timing and sequencing for all I/O operations. When an address IRAn + (0-7) is received, the I/O control store applies the addressed data IROn(0-15) to the buffer. The I/O control store can be disabled with a low XIOCE– when an external I/O control store is used. For a list of the binary codes contained at each address of the I/O control store, refer to documents 49A0195-000 and 49A0195-001 in the System Maintenance Manual.

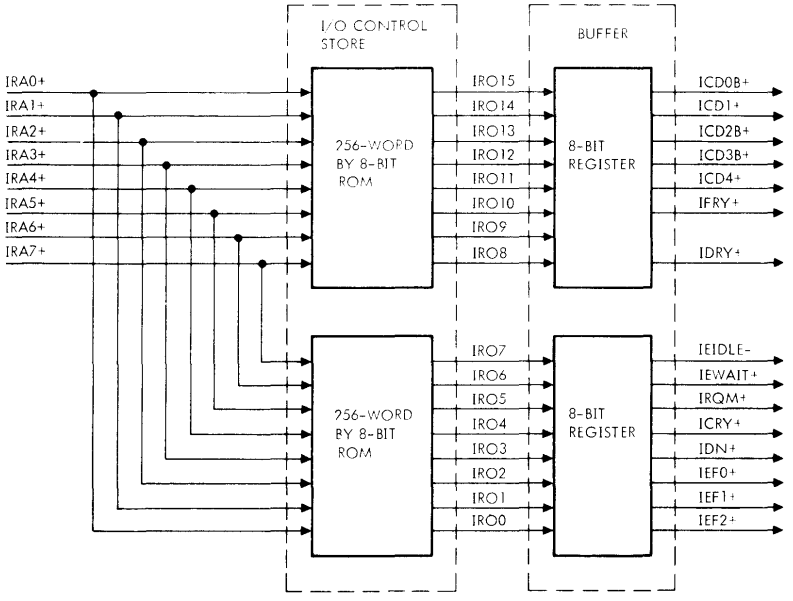
4.7.1.4 Buffer

The buffer consists of a 16-bit register that permits overlapping of microinstruction executions and access of the next microinstruction. Its outputs control all I/O operations and enable new I/O tasks when the current operation is completed. The format for the I/O control-store word is defined in figure 4-23. The buffer output states and control store addresses for the various I/O operations are listed in tables 4-19 through 4-26.



VTII-1647A

Figure 4-21. Microinstruction-Sequencing Circuits



VTII-1648A

NOTE: CONTROL SIGNALS ARE NOT SHOWN

Figure 4-22. I/O Control Store and Buffer Block Diagram

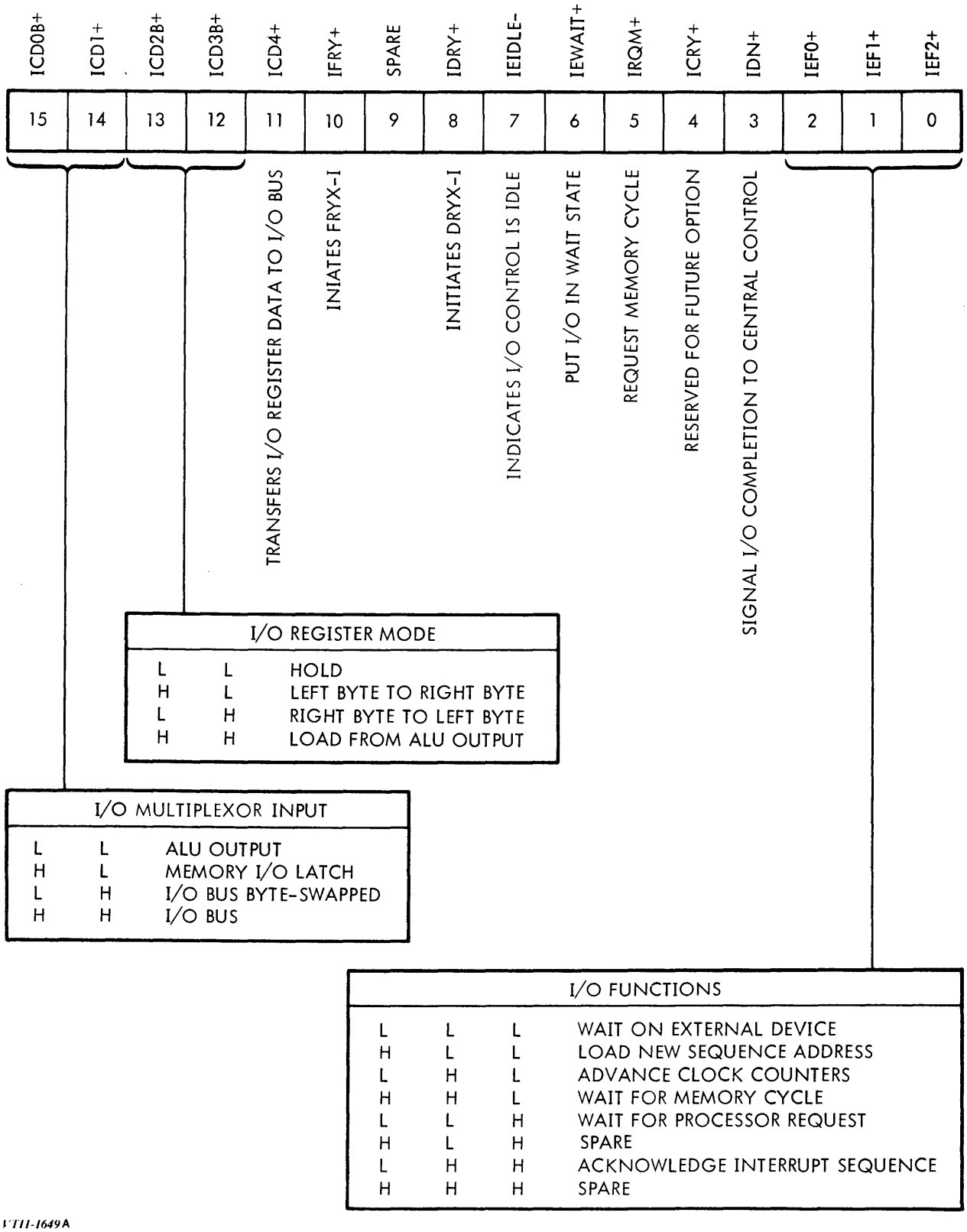


Figure 4-23. I/O Control-Store Word Format

Table 4-19. Programmed I/O for SEN, EXC, and EXC2 Instructions

I/O CONTROL STORE ADDRESS		BUFFER OUTPUTS															
DECIMAL	HEXADECIMAL	ICD0B+	ICD1+	ICD2B+	ICD3B+	ICD4+	IFRY+	SPARE	IDRY+	IEIDLE-	IEWAIT+	IRQM+	ICRY+	IDN+	IEF0+	IEF1+	IEF2+
0	0	L	L	L	L	L	L	L	L	H	L	L	L	H	L	L	L
4	4	L	L	H	H	L	L	L	L	H	L	L	L	H	L	L	L
5	5	L	L	L	L	L	L	L	L	H	L	L	L	H	L	L	L
6	6	L	L	L	L	H	L	L	L	H	L	L	L	H	L	L	L
7	7	L	L	L	L	H	H	L	L	H	L	L	L	H	L	L	L
8	8	L	L	L	L	H	H	L	L	H	L	L	L	H	L	L	H
9	9	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L
10	A	L	L	L	L	L	L	L	L	H	L	L	L	H	L	L	L

NOTES: 1. L = LOW, H = HIGH

2. ADDRESS ZERO IS THE RESET STATE

VTII-1650A

Table 4-20. Programmed I/O, Input-Data Transfer

I/O CONTROL STORE ADDRESS		BUFFER OUTPUTS															
DECIMAL	HEXADECIMAL	ICD0B+	ICD1+	ICD2B+	ICD3B+	ICD4+	IFRY+	SPARE	IDRY+	IEIDLE-	IEWAIT+	IRQM+	ICRY+	IDN+	IEF0+	IEF1+	IEF2+
12	C	L	L	H	H	L	L	L	L	H	L	L	L	H	L	L	L
13	D	L	L	L	L	L	L	L	L	H	L	L	L	H	L	L	L
14	E	L	L	L	L	H	L	L	L	H	L	L	L	H	L	L	L
15	F	L	L	L	L	H	H	L	L	H	L	L	L	H	L	L	L
16	10	L	L	L	L	H	H	L	L	H	L	L	L	H	L	L	L
17	11	H	H	L	L	L	L	L	L	H	L	L	L	H	L	L	L
18	12	H	H	L	L	L	L	L	H	H	L	L	L	H	L	L	L
19	13	H	H	H	H	L	L	L	H	H	L	L	L	L	L	L	L
20	14	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L
21	15	L	L	L	L	L	L	L	L	H	L	L	L	H	L	L	L

NOTE: L = LOW, H = HIGH

VTII-1651

Table 4-21. Programmed I/O, Output-Data Transfer

I/O CONTROL STORE ADDRESS		BUFFER OUTPUTS															
DECIMAL	HEXADECIMAL	ICD0B+	ICD1+	ICD2B+	ICD3B+	ICD4+	IFRY+	SPARE	IDRY+	IEIDLE-	IEWAIT+	IRQM+	ICRY+	IDN+	IEF0+	IEF1+	IEF2+
28	1C	L	L	H	H	L	L	L	L	H	L	L	L	L	L	L	L
29	1D	L	L	L	L	L	L	L	L	H	L	L	L	H	L	L	L
30	1E	L	L	L	L	H	L	L	L	H	L	L	L	H	L	L	L
31	1F	L	L	L	L	H	H	L	L	H	L	L	L	H	L	L	L
32	20	L	L	H	H	H	H	L	L	H	L	L	L	H	L	L	L
33	21	L	L	L	L	H	L	L	L	H	L	L	L	H	L	L	L
34	22	L	L	L	L	H	L	L	H	H	L	L	L	H	L	L	L
35	23	L	L	L	L	H	L	L	H	H	L	L	L	H	L	L	L
36	24	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L
37	25	L	L	L	L	L	L	L	L	H	L	L	L	H	L	L	L

NOTE: L = LOW, H = HIGH

VTII-1652

Table 4-22. DMA, Trap-Out Operation

I/O CONTROL STORE ADDRESS		BUFFER OUTPUTS															
DECIMAL	HEXADECIMAL	ICD0B+	ICD1+	ICD2B+	ICD3B+	ICD4+	IFRY+	SPARE	IDRY+	IEIDLE-	IEWAIT+	IRQM+	ICRY+	IDN+	IEF0+	IEF1+	IEF2+
64	40	H	H	L	L	L	L	L	L	H	L	L	L	L	L	H	H
65	41	H	H	L	L	L	H	L	L	H	L	L	L	L	L	L	L
66	42	H	H	H	H	L	H	L	L	H	L	L	L	L	L	L	L
67	43	L	L	L	L	L	L	L	L	H	H	H	L	L	H	H	L
68	44	H	L	H	H	L	L	L	L	H	H	L	L	L	H	H	L
69	45	L	L	L	L	H	L	L	L	H	L	L	L	L	L	L	L
70	46	L	L	L	L	H	L	L	H	H	L	L	L	L	L	L	L
71	47	L	L	L	L	H	L	L	H	H	L	L	L	L	L	H	L
72	48	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L
73	49	L	L	L	L	L	L	L	L	H	L	L	L	H	L	L	L

NOTE: L = LOW, H = HIGH

VTII-1653

Table 4-23. DMA, Trap-In Operation

I/O CONTROL STORE ADDRESS		BUFFER OUTPUTS															
DECIMAL	HEXADECIMAL	ICD0B+	ICD1+	ICD2B+	ICD3B+	ICD4+	IFRY+	SPARE	IDRY+	IEIDLE-	IEWAIT+	IRQM+	ICRY+	IDN+	IEF0+	IEF1+	IEF2+
80	50	H	H	L	L	L	L	L	L	H	L	L	L	L	L	H	H
81	51	H	H	L	L	L	H	L	L	H	L	L	L	L	L	L	L
82	52	H	H	H	H	L	H	L	L	H	L	L	L	L	L	L	L
83	53	L	L	L	L	L	L	L	L	H	L	L	L	L	L	L	L
84	54	L	L	L	L	L	L	L	L	H	L	L	L	L	L	L	L
85	55	H	H	L	L	L	L	L	L	H	H	H	L	L	H	H	L
86	56	H	H	L	L	L	L	L	L	H	L	L	L	L	L	L	L
87	57	H	H	L	L	L	L	L	H	H	H	L	L	L	H	H	L
88	58	H	H	L	L	L	L	L	H	H	L	L	L	L	L	H	L
89	59	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L
90	5A	L	L	L	L	L	L	L	L	H	L	L	L	H	L	L	L

NOTE: L = LOW, H = HIGH

VTII-1654

Table 4-24. High-Speed DMA, Trap-Out Operation

I/O CONTROL STORE ADDRESS		BUFFER OUTPUTS															
DECIMAL	HEXADECIMAL	ICD0B+	ICD1+	ICD2B+	ICD3B+	ICD4+	IFRY+	SPARE	IDRY+	IEIDLE-	IEWAIT+	IRQM+	ICRY+	IDN+	IEF0+	IEF1+	IEF2+
96	60	H	H	H	H	L	H	L	L	H	H	H	L	L	L	H	H
97	61	L	L	L	L	L	L	L	L	H	L	L	L	L	L	L	L
98	62	H	L	H	H	H	L	L	H	H	H	L	L	L	L	H	H
99	63	L	L	L	L	H	L	L	H	H	L	L	L	L	L	H	L
100	64	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L
101	65	L	L	L	L	L	L	L	L	H	L	L	L	H	L	L	L

NOTE: L = LOW, H = HIGH

VTII-1655

Table 4-25. High-Speed DMA, Trap-In Operation

I/O CONTROL STORE ADDRESS		BUFFER OUTPUTS															
DECIMAL	HEXADECIMAL	ICD0B+	ICD1+	ICD2B+	ICD3B+	ICD4+	IFRY+	SPARE	IDRY+	IEIDLE-	IEWAIT+	IRQM+	ICRY+	IDN+	IEF0+	IEF1+	IEF2+
112	70	H	H	H	H	L	H	L	L	H	H	H	L	L	L	H	H
113	71	H	H	L	L	L	L	L	L	H	L	L	L	L	L	L	L
114	72	H	H	L	L	L	L	L	H	H	H	L	L	L	H	H	H
115	73	L	L	L	L	L	L	L	L	H	L	L	L	L	L	H	L
116	74	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L
117	75	L	L	L	L	L	L	L	L	H	L	L	L	H	L	L	L

NOTE: L = LOW, H = HIGH

VTII-1656

Table 4-26. I/O Interrupt Operation

I/O CONTROL STORE ADDRESS		BUFFER OUTPUTS															
DECIMAL	HEXADECIMAL	ICD0B+	ICD1+	ICD2B+	ICD3B+	ICD4+	IFRY+	SPARE	IDRY+	IEIDLE-	IEWAIT+	IRQM+	ICRY+	IDN+	IEF0+	IEF1+	IEF2+
220	DC	H	H	L	L	L	L	L	L	H	L	L	L	H	L	H	H
221	DD	H	H	L	L	L	H	L	L	H	L	L	L	H	L	L	L
222	DE	H	H	H	H	L	H	L	L	H	L	L	L	L	L	L	L
223	DF	L	L	L	L	L	L	L	L	H	H	L	L	L	L	L	L
224	E0	L	L	L	L	L	L	L	H	H	L	L	L	L	L	L	L
225	E1	L	L	L	L	L	L	L	H	H	L	L	L	L	L	H	L
226	E2	L	L	L	L	L	L	L	L	H	L	L	L	L	L	L	L
227	E3	L	L	L	L	L	L	L	L	H	L	L	L	L	L	L	L
228	E4	L	L	L	L	L	L	L	L	H	L	L	L	L	L	L	L
229	E5	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L
230	E6	L	L	L	L	L	L	L	L	H	L	L	L	H	L	L	L

NOTE: L = LOW, H = HIGH

VTII-1657

4.7.2 Decoding and Sequencing Control

The decoding and sequencing control (figure 4-24) decodes the I/O function field of the I/O control-store word, and supplies control signals to the microinstruction-sequencing circuits and to several other I/O control circuits.

4.7.2.1 Address-Counter Loading Control

This circuit provides the loading signal (ILDAD–) for the address counter. ILDAD– is active (low) when the buffer specifies an idle condition (IEIDLE + high) or a waiting condition is specified and a new address is to be loaded (IEWAIT + and ID1 + high). Loading of the address counter occurs when ILDAD– is low and the I/O clock (IFC–) is not inhibited by the clock inhibitor signal (IINHC–).

4.7.2.2 Address Selector Control

This circuit provides the selector signal (IMXAD +) for the address selector circuit. When IMXAD + is low, a trap address is selected (for normal or high-speed DMA). IMXAD + goes low during a trap sampling time if an external device (usually a buffered interlace controller) requests a DMA operation. Trap sampling for a normal DMA operation occurs when IESMP + is high, and for high-speed DMA operations when IEIDLE + and ICD are high. At all other times, the selector signal IMXAD + is high and a central control address is selected.

4.7.2.3 Clock Inhibitor

This circuit provides a clock inhibitor signal (IINHC–) that inhibits loading of the address counter and buffer. IINHC– is active (low) when there is no I/O activity (IEIDLE + and IIIDLE– high) or the I/O control is waiting for the central control to complete an operation. The waiting conditions are:

a. Waiting for the central control to respond (CRQIO + high)

b. Waiting for the memory control to respond to a request for a trap memory cycle (MAKIO + high).

c. Waiting for the central control to complete an interrupt service routine (IINTF + high).

4.7.2.4 I/O Function Decoder

This circuit provides various control signals by decoding the three I/O function bits IEFn + (0-2).

4.7.2.5 Inhibit-Idle Control

This circuit can override the clock inhibitor IINHC– by generating a low inhibit-idle signal IIIDLE–. The inhibit-idle signal is also used to enable the address selector. IIIDLE– is activated (low) during one of the sample times (IESMP + or ICD + high) if one of the following conditions occur:

a. A normal trap operation (DMA) is requested (ITRPN + high).

b. Central control requests an interrupt operation (IDCIR + and CRQIO + high).

c. Programmed I/O operation is requested (IDCIR– and CRQIO + high).

d. A high-speed trap operation (DMA) is requested (ITRPF + high).

4.7.3 Interrupt and DMA Control

The interrupt and DMA control (figure 4-25) performs the following functions:

a. Generates and controls the interrupt clocks (IUCX-I and IUCF-I)

b. Supplies input signals to the sample control logic

c. Generates interrupt acknowledgments (IUAX-I and IUAF-I)

d. Generates the I/O interrupt flag (IINTF +)

4.7.3.1 Interrupt Clock Generator

This circuit generates the ICA– and ICD + clocks which are sent to I/O bus drivers to become the normal and high-speed clocks IUCX-I and IUCF-I.

The period of the normal interrupt clock (IUCX-I) is jumper selectable at either 660 or 990 nanoseconds. Selection depends on the length of the priority chain in the system (for a description of the priority system, refer to the Input/Output Section of the applicable system handbook).

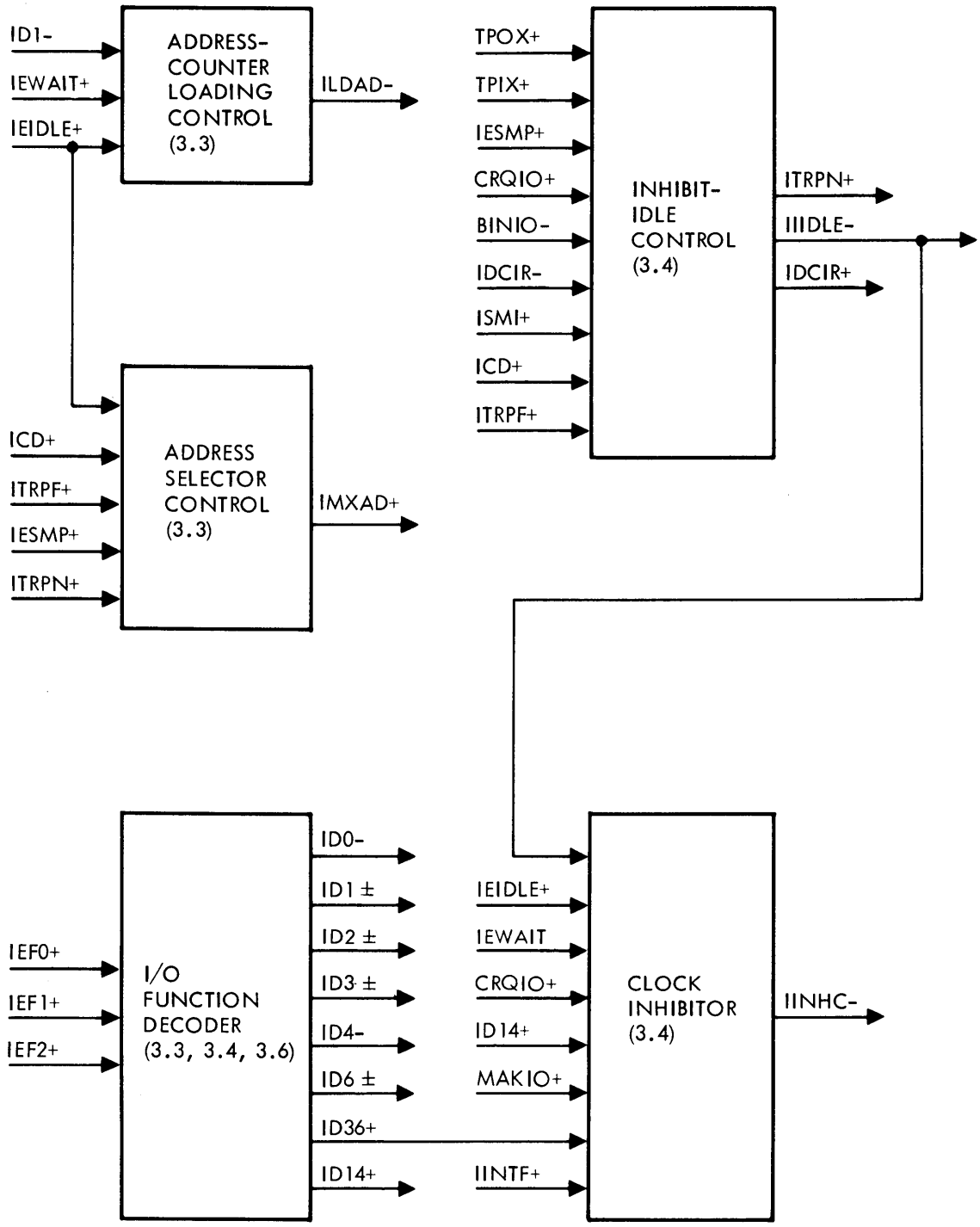
The period of the high-speed clock (IUCF-I) is fixed at 330 nanoseconds. This clock is used during high-speed DMA operations.

4.7.3.2 Sample Control Logic

This circuit provides the following signals

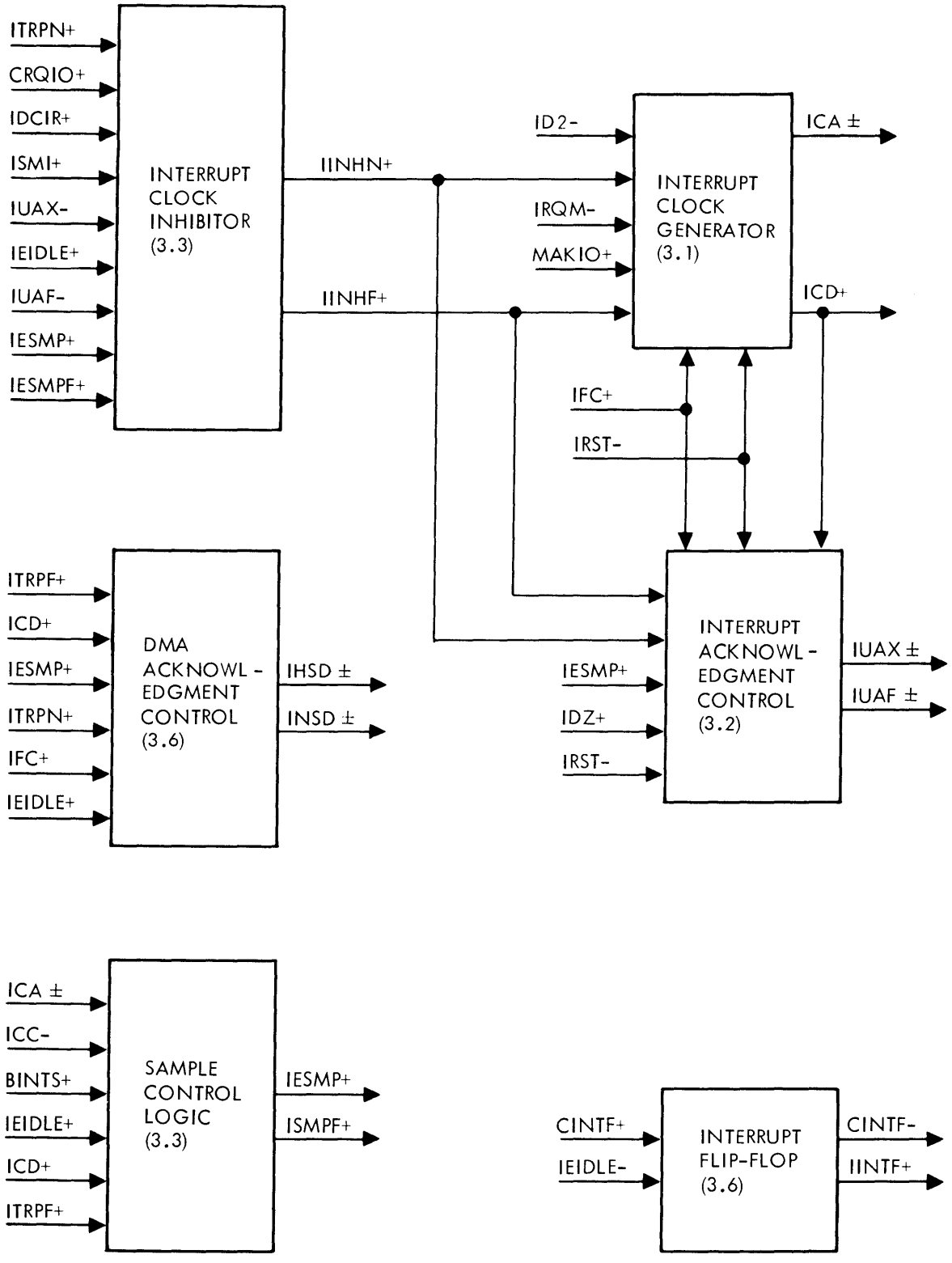
a. IESMP + . When there is no I/O activity, (IEIDLE + high) IESMP + samples normal DMA or interrupt requests to determine if they occur during the proper phase (ICA– high) of the interrupt clock (IUCX-I). IESMP + also tests for I/O requests from the central control.

b. ISMPF + . When there is no I/O activity (IEIDLE high), ISMPF + samples high-speed DMA requests to determine if they occur during the proper phase (ICD + high) of the high-speed interrupt clock (IUCF-I).



VTI1-2158

Figure 4-24. Decoding and Sequencing Control



VTI1-2159

Figure 4-25. Interrupt and DMA Control

4.7.3.3 Interrupt Acknowledgment Control

This circuit consists of two flip-flops that provide the normal and high-speed interrupt acknowledgments IUAX and IUAF. These signals are used in the I/O control circuits and are also routed through I/O drivers to the I/O bus (IUAX-I and IUAF-I)

4.7.3.4 Interrupt Flip-Flop

When CINTF + goes high, the interrupt flip-flop is set (IINTF + high) indicating the central control has completed its interrupt service routine and that the I/O control can continue processing and complete its interrupt routine.

4.7.3.5 Interrupt Clock Inhibitor

This circuit provides two interrupt clock inhibitors: a high IINHN + inhibits the normal clock (IUCX-I) and a high IINHF + inhibits the high-speed clock (IUCF-I).

IINHN + goes high during a sampling time (IESMP + high) if one of the following conditions occur:

a. A normal DMA operation is requested (ITRPN + high).

b. An interrupt sequence is acknowledged by the central control (CRQIO + high).

Both the above conditions are inhibited if a high-speed DMA operation is requested. This is accomplished by disabling IESMP + during the high-speed sampling time (ICD + and ITRPF + high). IINHN + also remains high (if a normal DMA or interrupt is recognized) as long as the interrupt acknowledgment is set (IUAX- low).

IINHF + goes high if a high-speed DMA request is made (ITRPF + high) during a sampling time (IEIDLE + and ICD + high). Normal DMA operations cannot override high-speed DMA operations. When a high-speed request is accepted, IINHF + remains high as long as the high-speed interrupt acknowledgment is set (IUAF- low).

4.7.3.6 DMA Acknowledgment Control

This circuit consists of two flip-flops that indicate if a DMA request has been sampled and accepted by the I/O control. A high INSD + acknowledges normal DMA requests and a high IHSD + acknowledges high-speed DMA requests.

4.7.4 I/O Data-Loop Control

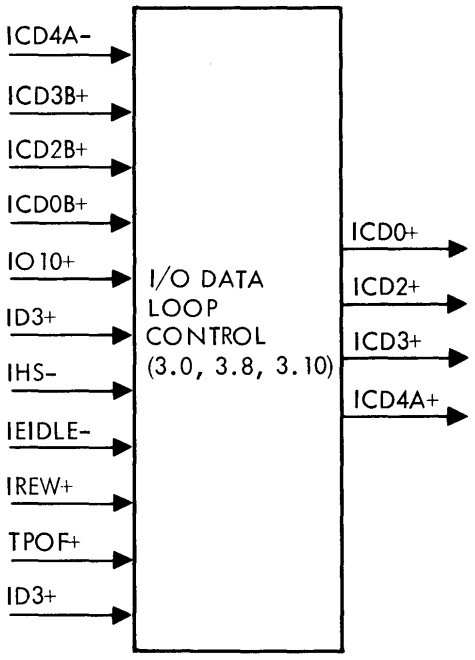
This circuit (figure 4-26) provides control signals for the I/O data loop.

ICD0 + is used in the selection of input data for the I/O multiplexor (section 4.6.1).

ICD2 + and ICD3 + determine the operating modes of the I/O register (section 4.6.2).

ICD4A + enables the I/O drivers (section 4.6.3) to transfer data from the I/O register to the I/O bus.

Functions of the above signals are summarized in the I/O control-store word format in figure 4-23.



VTI1-2160

Figure 4-26. I/O Data Loop Control

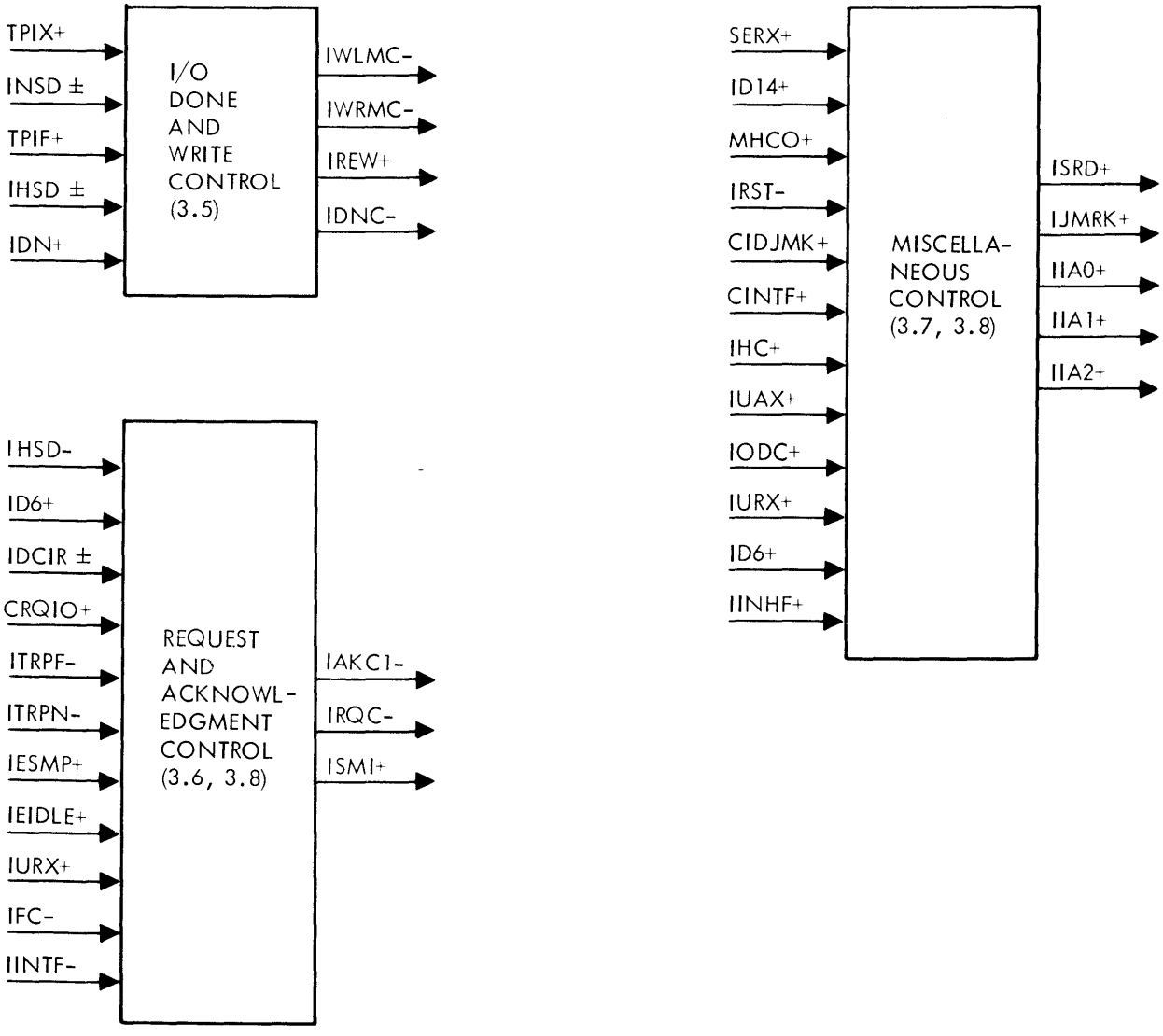
4.7.5 Processor-I/O Control

These circuits (figure 4-27) supply address and control signals to the central control.

4.7.5.1 I/O Done and Write Control

An I/O done condition exists when IDNC- goes high. This condition occurs during a programmed I/O or interrupt sequence to signal the central control that data is now valid and can be interrogated. IDNC- is held low during high-speed or normal DMA operations by a low IHSD- or INSD-.

During DMA operations, two write control line signals IWLMC- and IWRMC- are sent to the memory control. The write control lines are enabled (IWLMC- and IWRMC- low) either by high TPIX + and INSD + for normal DMA or high TPIF + and IHSD + during high-speed DMA.



VTII-2161

Figure 4-27. Processor I/O Control

4.7.5.2 Request and Acknowledgment Control

When an interrupt is requested (IURX + high), a low IRQC- is sent to the central control. IRQC- is reset to the high state when the I/O interrupt flag is set (IINTF- low). A high IURX + also causes ISMI + to go high indicating a program interrupt is pending.

When the I/O control has responded to a processor request (CRQIO + high), an acknowledgment signal (IAKC1- low) is sent to the central control. A low IAKC1- is generated when one of two conditions are acknowledged:

a. Process request of an interrupt operation (CRQIO + and IDCIR + high)

b. Processor request of a programmed I/O operation (CRQIO + high and IDCIR + low)

4.7.5.3 Miscellaneous Control

For interrupt operations, interrupt address bits IIAn + (0-2) are used as a portion of the central control store address CEADn- (0-8).

The sense response signal ISRD + is sent to the central control to indicate the status of a programmed I/O sense operation. If the sensed condition is true (SERX-I low), ISRD + is high. Conversely, if the sensed condition is false (SERX-I high), ISRD + is low. Refer to the Input/Output Section of the applicable system handbook for further information on programmed I/O sense operations.

If a jump and mark instruction is located at the interrupt address during an interrupt operation, IJMRK + goes high and transfers IUJX-I onto the I/O bus. IJMRK + goes high when the processor interrupt flag CINTF + and a decoded jump and mark signal CIDJMK + are high. At the completion of the I/O interrupt routine (IUAX + low), IJMRK returns to the low state.

4.7.6 Drivers/Receivers and Miscellaneous Control

These circuits are shown in figure 4-28.

4.7.6.1 Drivers/Receivers

These circuits are the I/O bus interface for the various I/O control signals.

4.7.6.2 Buffers and Decoder

These circuits perform the following functions:

a. Generates I/O full clocks IFC + and IFC- by buffering the processor full clock MFCO-.

b. Generates I/O reset signals IRST + and IRST- by buffering the system reset signal SRST-.

c. Generates the I/O half clocks IHC + and IHC- by buffering the processor half clock MHCO-.

d. Generates the decoded interrupt signal IDCIR- by decoding CBMT + and CBMR + .

4.7.6.3 Function and Data Ready Control

This circuit controls the length of time, the function ready and data ready signals are active. For timing waveforms, refer to the Input/Output Section of the applicable system handbook.

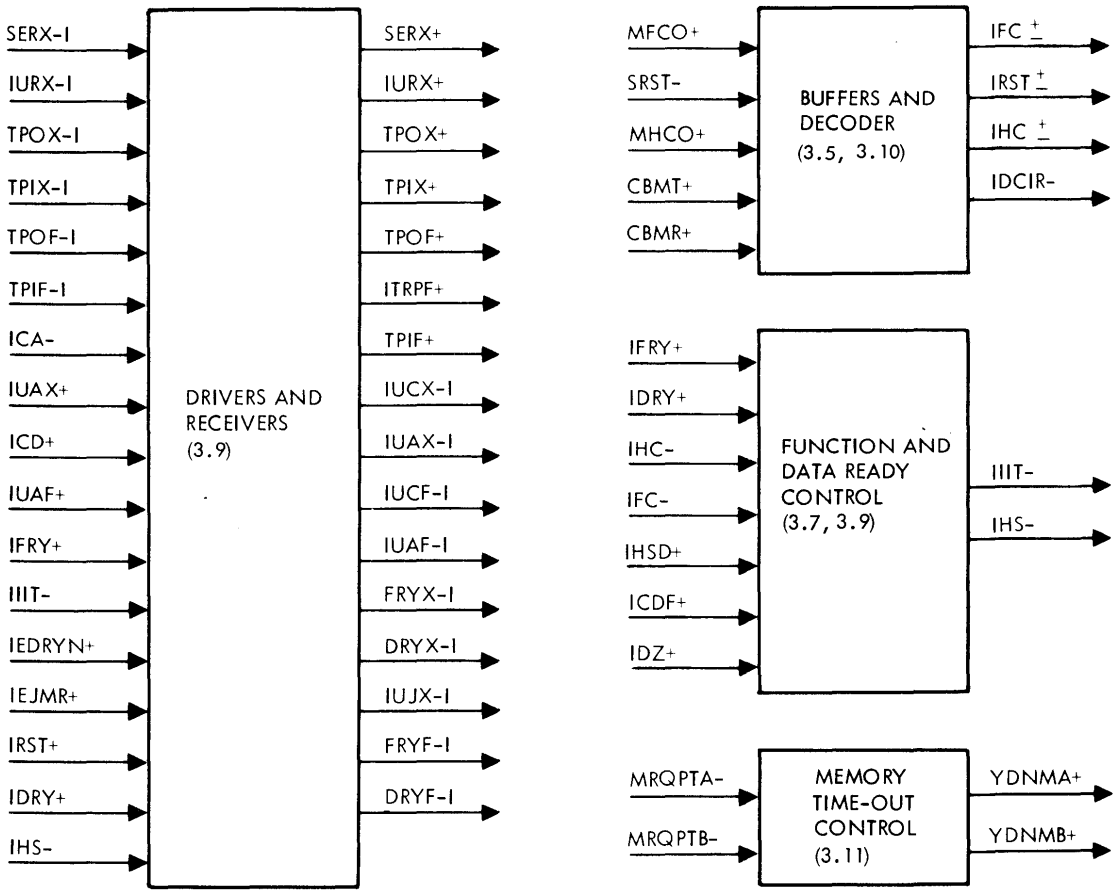
During normal I/O operations requiring data transfers onto the I/O bus, the pulse width of function ready or data ready signals (FRYX-I or DRYX-I) is shortened. This permits data to remain on the I/O bus for some period of time after the removal of one of these signals (see figure 4-29). A peripheral controller can then use the trailing edge (low-to-high transition) of the function ready or data ready signal to interrogate the data from the processor. Pulse width of DRYX-I or FRYX-I is shortened at the trailing edge of the I/O half clock IHC- when IIT- is low.

During high-speed DMA operations (figure 4-30) only the pulse width of DRYF-I is shortened during an output-data transfer. DRYF-I is shortened at the trailing edge of the I/O half clock IHC- when ID2- is low (IHS- low). No pulse-width change is required during an input-data transfer.

4.7.6.4 Memory Time-Out

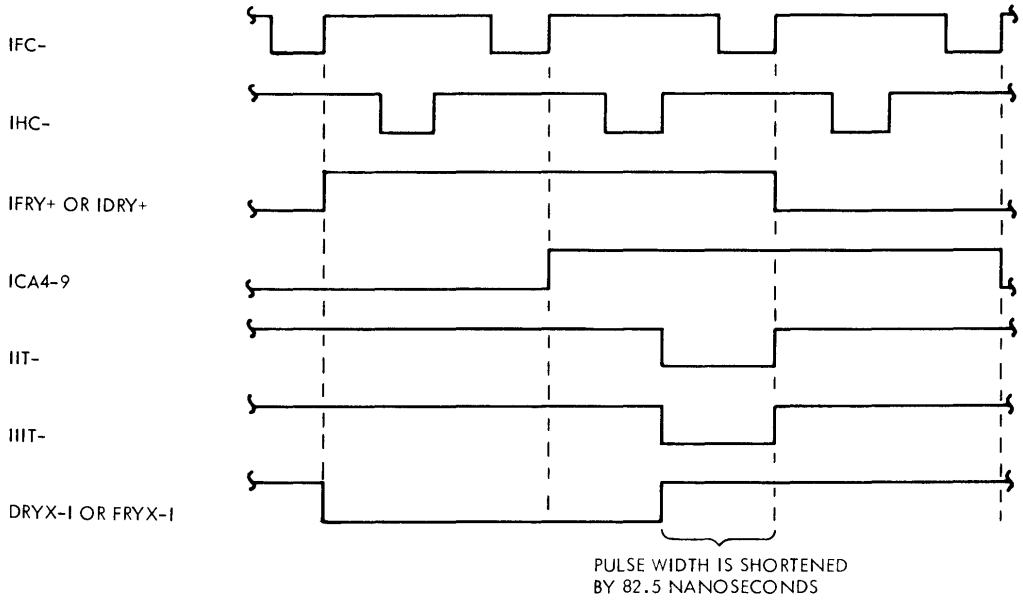
A memory time-out circuit is provided on the option board to terminate requests to non-existent memory locations. These requests may result from programming errors and, if wrap-around addressing is not employed, the system would otherwise hang-up waiting for a memory acknowledgment. The memory time-out circuit monitors all requests (MRQPTA- and MRQPTB-) on both memory buses and generates a memory acknowledgment signal (YDNMA + or YDNMB +) to the appropriate bus if a request has not been received from the memory within two milliseconds. This interval is considered long enough to avoid false time-outs which might result from higher priority port accesses. Timing waveforms for a time-out and non time-out conditions are illustrated in figure 4-31.

The memory time-out circuit can be disabled with jumper connections on the option board.



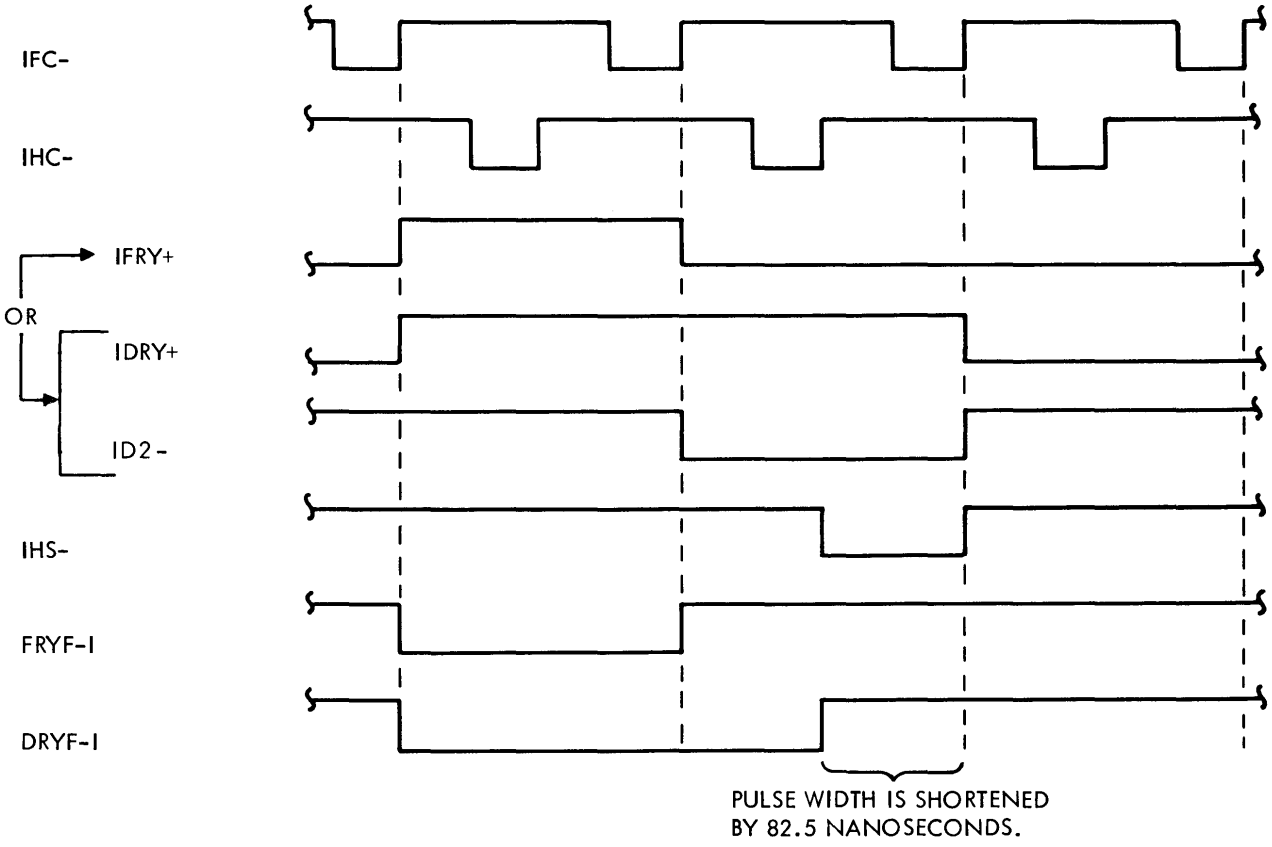
VTI1-2143

Figure 4-28. Drivers/Receivers and Miscellaneous Control

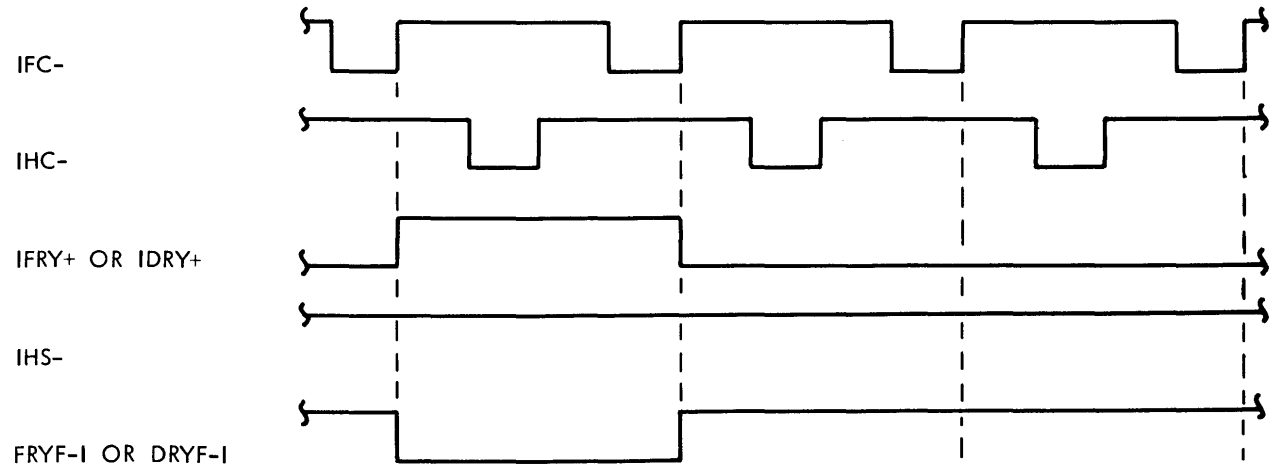


VTI1-2144

Figure 4-29. Normal I/O Operations



A. OUTPUT-DATA TRANSFER



B. INPUT-DATA TRANSFER

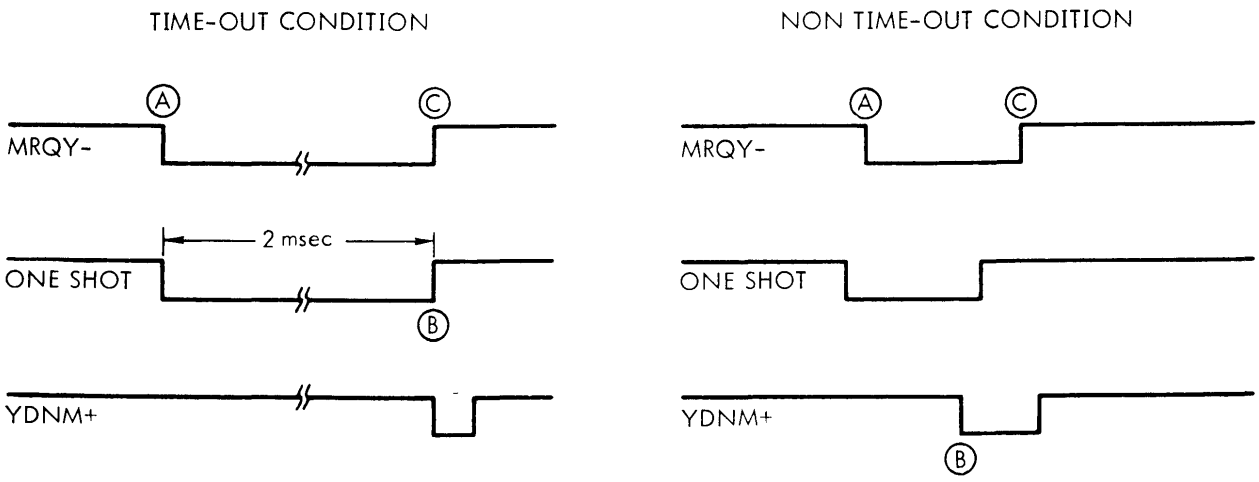
VTII-2145

Figure 4-30. High-Speed DMA Operations

4.7.7 Programmed I/O Operation

Timing waveforms for a programmed I/O operation are illustrated in figure 4-32. A programmed I/O operation is initiated when a high CRQIO + is received from the processor. This occurs when the processor IM and S fields contain 111x and 00, respectively. Other processor fields are interpreted by the I/O control as starting addresses for the I/O control store (section 4.7.1). The starting address is formed as follows:

Processor Field Bit	Address Bit
CBMT +	7
CBMR +	6
CBTS3 +	5
CBTS2 +	4
CBTS1 +	3
CBTS0 +	2
CBAB1 +	1
Zero	0



A. MEMORY REQUEST (MRQY-) BECOMES ACTIVE CAUSING ONE SHOT (2 msec) TO START TIME-OUT AND REMOVING THE CLEAR INPUT FROM A FLIP-FLOP IN THE TIME-OUT CIRCUIT

B. TWO-MILLISECOND TIME-OUT EXPIRES PROVIDING A CLOCK TO FLIP-FLOP. SINCE MEMORY ACKNOWLEDGEMENT (YDNM+) IS STILL HIGH, FLIP-FLOP IS SET, DRIVING YDNM+ LOW WHICH CLAMPS ONE SHOT TO A CLEAR STATE PREVENTING FURTHER TRIGGERING

C. REQUESTING DEVICE RECOGNIZES MEMORY ACKNOWLEDGEMENT SIGNAL AND REMOVES REQUEST CAUSING FLIP-FLOP TO BE CLEARED, THUS REMOVING MEMORY ACKNOWLEDGEMENT SIGNAL

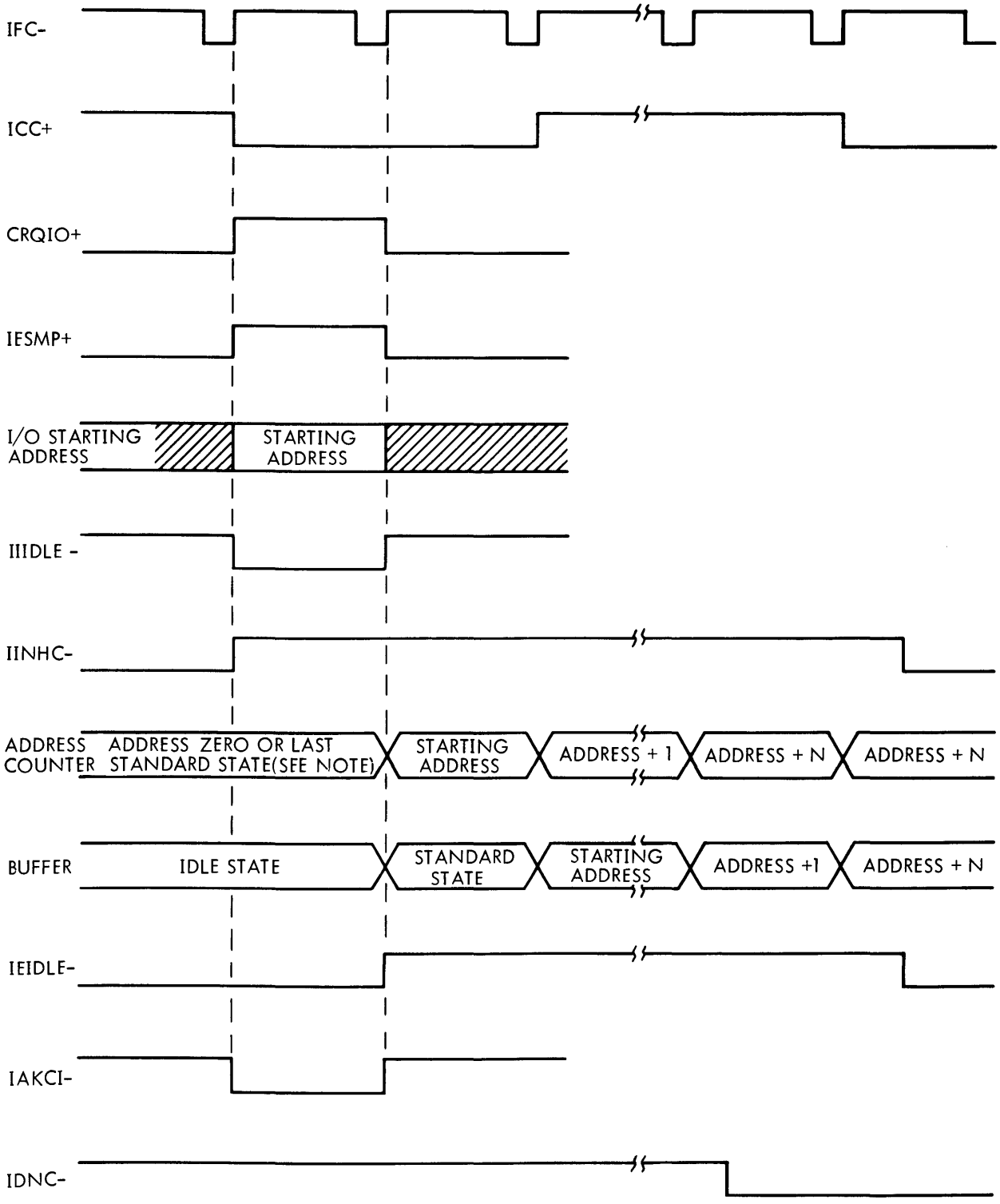
A. MEMORY REQUEST (MRQY-) BECOMES ACTIVE CAUSING ONE SHOT TO START TIME-OUT AND REMOVING THE CLEAR INPUT FROM A FLIP-FLOP IN THE TIME-OUT CIRCUIT

B. MEMORY ACKNOWLEDGEMENT SIGNAL IS RECEIVED TERMINATING THE ONE-SHOT OUTPUT PULSE SINCE YDNM+ IS LOW THE FLIP-FLOP IS NOT SET

C. REQUESTING DEVICE RECOGNIZES MEMORY ACKNOWLEDGEMENT SIGNAL AND REMOVES REQUEST CAUSING FLIP-FLOP TO BE HELD RESET

VTTT-1661

Figure 4-31. Memory Time-Out Waveforms



NOTE: STANDARD STATE OCCURS WHEN I/O CONTROL ACCEPTS A REQUEST (IEIDL- AND IDN+ HIGH)

VTI1-1658A

Figure 4-32. Programmed I/O Control Waveforms

Upon receipt of a h... CRQIO+, the I/O inhibit idle signal IIIDLE– goes low if a... the following conditions occur:

a. The interrupt-clock counter is in the sampling state (ICA– and ICC– high, resulting in IESMP + high).

b. The inhibiting signal of the memory protection option is not present (BINIO– high).

c. The two most-significant address bits (CBMT + and CBMR +) are not both high, indicating the request is not an interrupt sequence (IDCIR– high).

The low IIIDLE– causes the I/O clock inhibitor IINHC– to go high, enabling the clocks for the address counter and buffer (section 4.7.1).

At the next IFC– clock pulse, the address counter is loaded with the starting address received from the processor. The address is supplied by the address selector when IMXAD + is high. IMXAD + is high unless a DMA operation is requested (ITRPN + or ITRPF + high). When the starting address is loaded into the counter, the buffer is also loaded with the contents of the I/O control-store address corresponding to the last data in the address register. Following a system reset (IRST–low), the buffer receives data from address zero of the control store; at all other times, the data are from the ending address of the previous I/O operation. In either case, the data are identical and signals IEIDLE– and IDN + go high. A high IEIDLE– indicates that the I/O control is not idle and further requests are ignored. As long as IEIDLE– is high the address counter and buffer are enabled since IINHC– is latched high by a low IEIDLE + . At each succeeding IFC– clock pulse, the address counter is incremented and the buffer is loaded with the contents of the address designated by the address counter.

Any I/O operation continues through successive addresses of the I/O control store until address-counter and buffer clocks are inhibited by either of two conditions:

a. IEIDLE– goes low indicating the end of an I/O operation.

b. IEWAIT + goes high indicating that the current operation must stop to wait for one of the following external events to occur:
   1. Memory cycle (MAKIC + high).
   2. New processor ... (CRQIO + high).
   3. Interrupt (IINTF + high).
   4. External wait line is active (IEXW + high).

At the conclusion of a programmed I/O operation, IDNC– goes low and, on the next clock pulse, IEIDLE– goes low inhibiting clocks for the address counter and buffer (IINHC– low).

Upon completion of an I/O operation, the address counter contains data that cause IEIDLE– and IDN + to go high. These are the first data loaded into the buffer when clocks are reenabled for the next I/O operation.

4.7.8 Interrupt Operations

The interrupt operations are divided into seven sub-operations. Each sub-operation is shown on the interrupt flow diagram in figure 4-33. Timing waveforms are given for the overall operation (figure 4-34) as well as for each sub-operation.

SUB-OPERATION 1
Refer to figure 4-35 for timing waveforms of sub-operation 1. An interrupt operation is initiated when a device (such as the real time clock or priority interrupt module) generates an interrupt request (IURX-I low) on the positive-going transition of the interrupt clock IUCX-I. The I/O control then sends a low IRQC– to the central control and waits for a response.

SUB-OPERATION 2
Refer to figure 4-36 for timing waveforms of sub-operation 2. After some period of time (determined by the type of operation being performed by the central control prior to the interrupt request), the central control executes a micro-instruction that tests for an I/O interrupt (see section 4.3.14 for testing method). Following the test, the central control executes the first microinstruction of the interrupt service routine (see interrupt sequence in Micro-Word Flowcharts 98A0887) and generates a high CRQIO + . This microinstruction also tests the interrupt request. If the interrupt request signal IRQC– was activated as a result of noise or disabled by a macro instruction, the central control returns to its normal processing operations (shown as EXIT on the interrupt flow chart).

The central control waits (by executing the IWAIT microinstruction at 0D7) for the I/O control to respond to the high CRQIO + , and continues to send the starting address 0DC (see table 4-26) of the interrupt service routine to the address selector.

SUB-OPERATION 3
Refer to figure 4-37 for timing waveforms of sub-operation 3. The I/O control samples CRQIO + during the time IESMP + is high. Following the sampling, the I/O control performs the following:

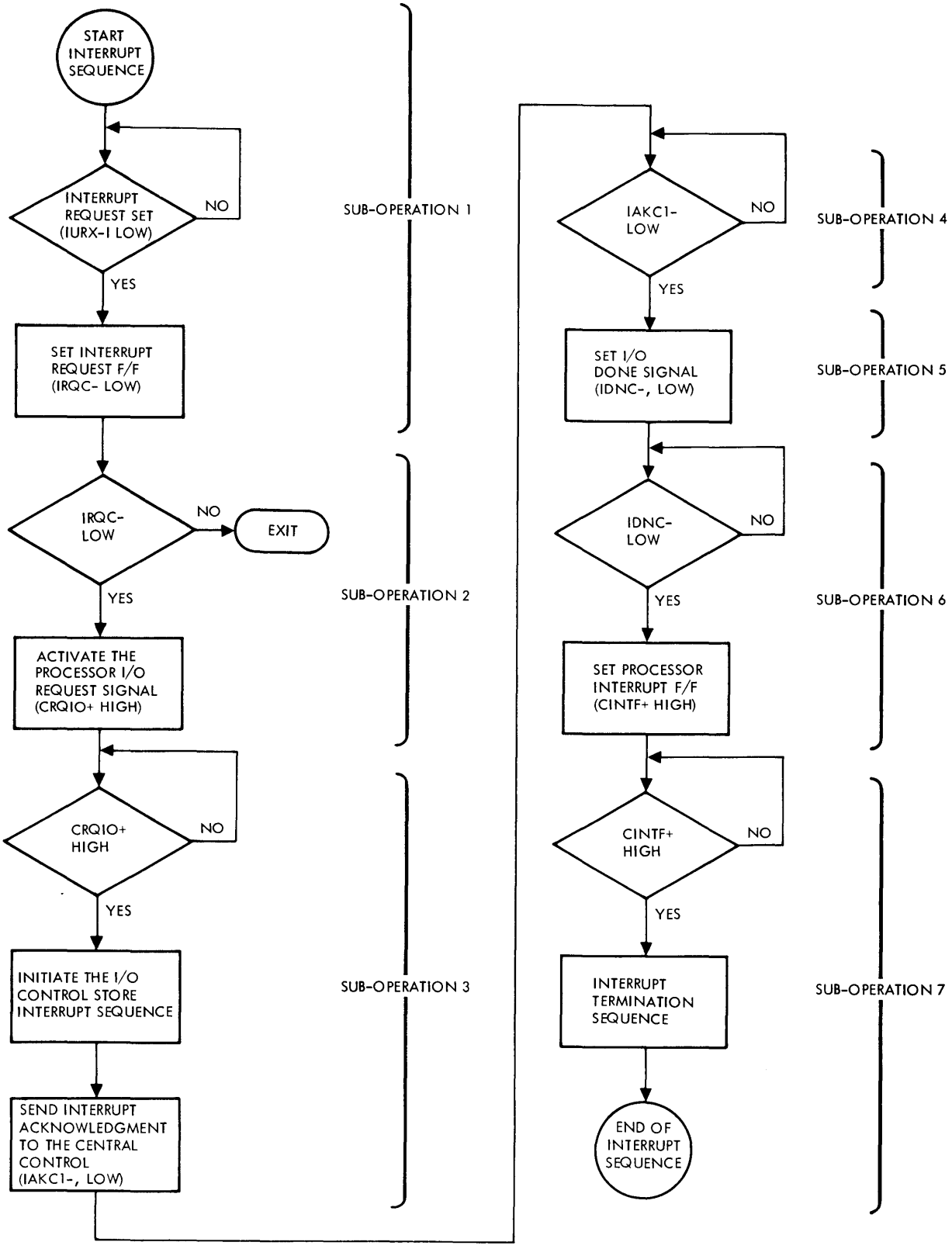
a. Sends an interrupt acknowledgment (IUAX-I low) to the I/O bus.

b. Loads the starting address (0DC) for the I/O interrupt operation into the address counter (IINHC– high).

c. Loads the buffer with the last standard state address (zero in the case of a reset or previously inactive I/O control.

d. Sends an acknowledgment (IAKC1– low) to the central control.

At this time, the I/O control is committed and must complete the interrupt operation. The only other termination that can occur is a system reset.



VTI1-2146

Figure 4-33. Interrupt Flow Diagram

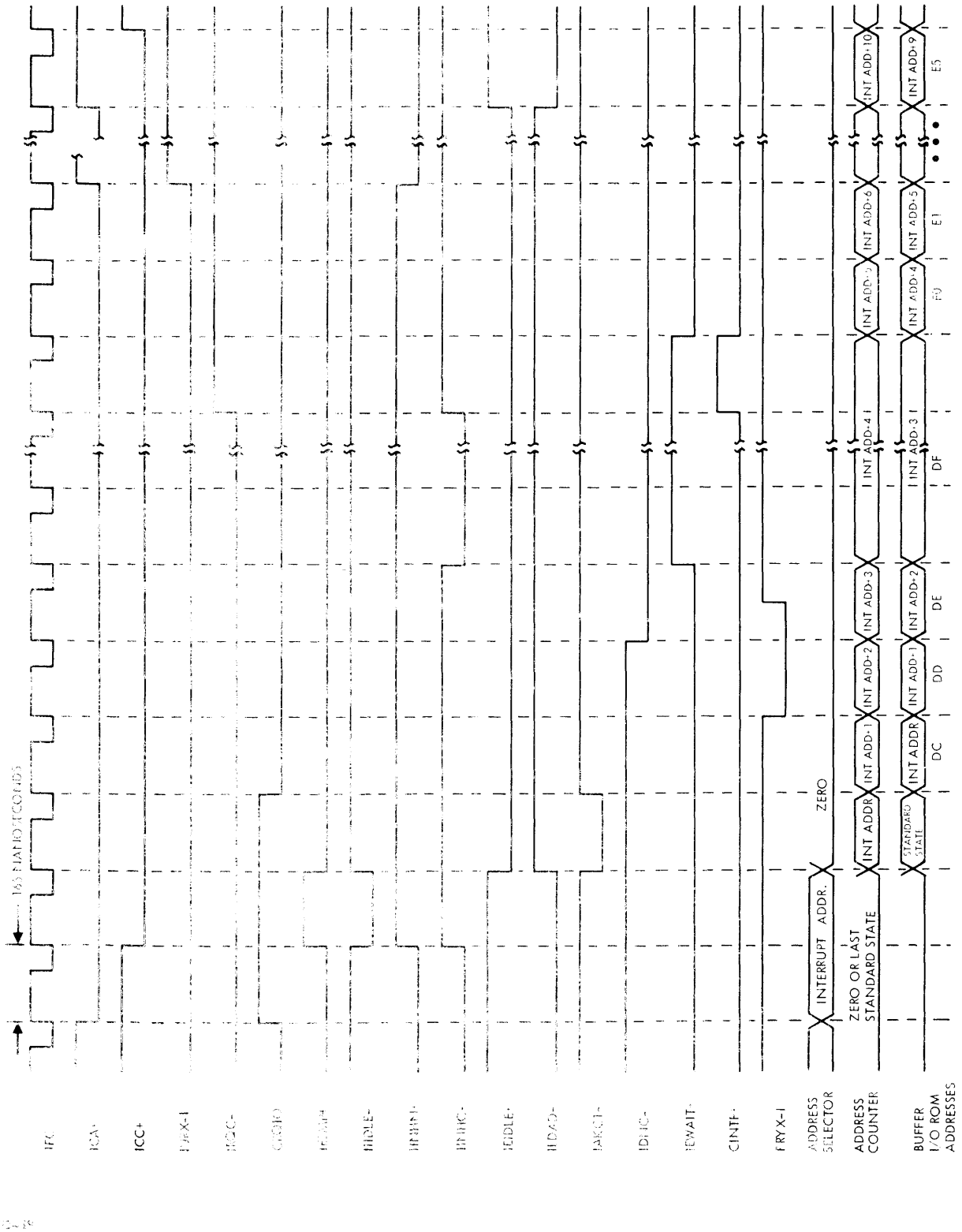
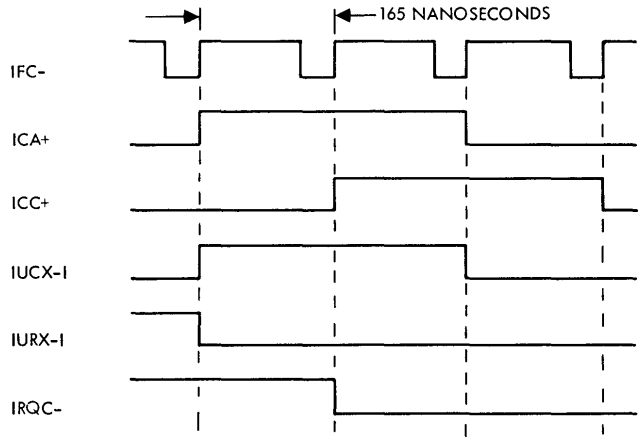


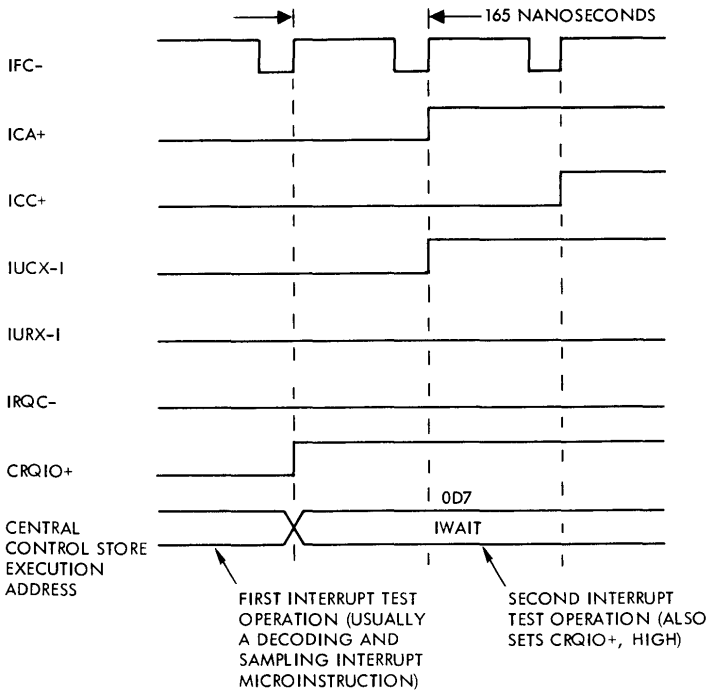
Figure 4-31. Interrupt Timing Waveforms



NOTE: TIMING ASSUMES AN INTERRUPT CLOCK RATE OF 660 NANOSECONDS (IUCX-I).

VT11-2162

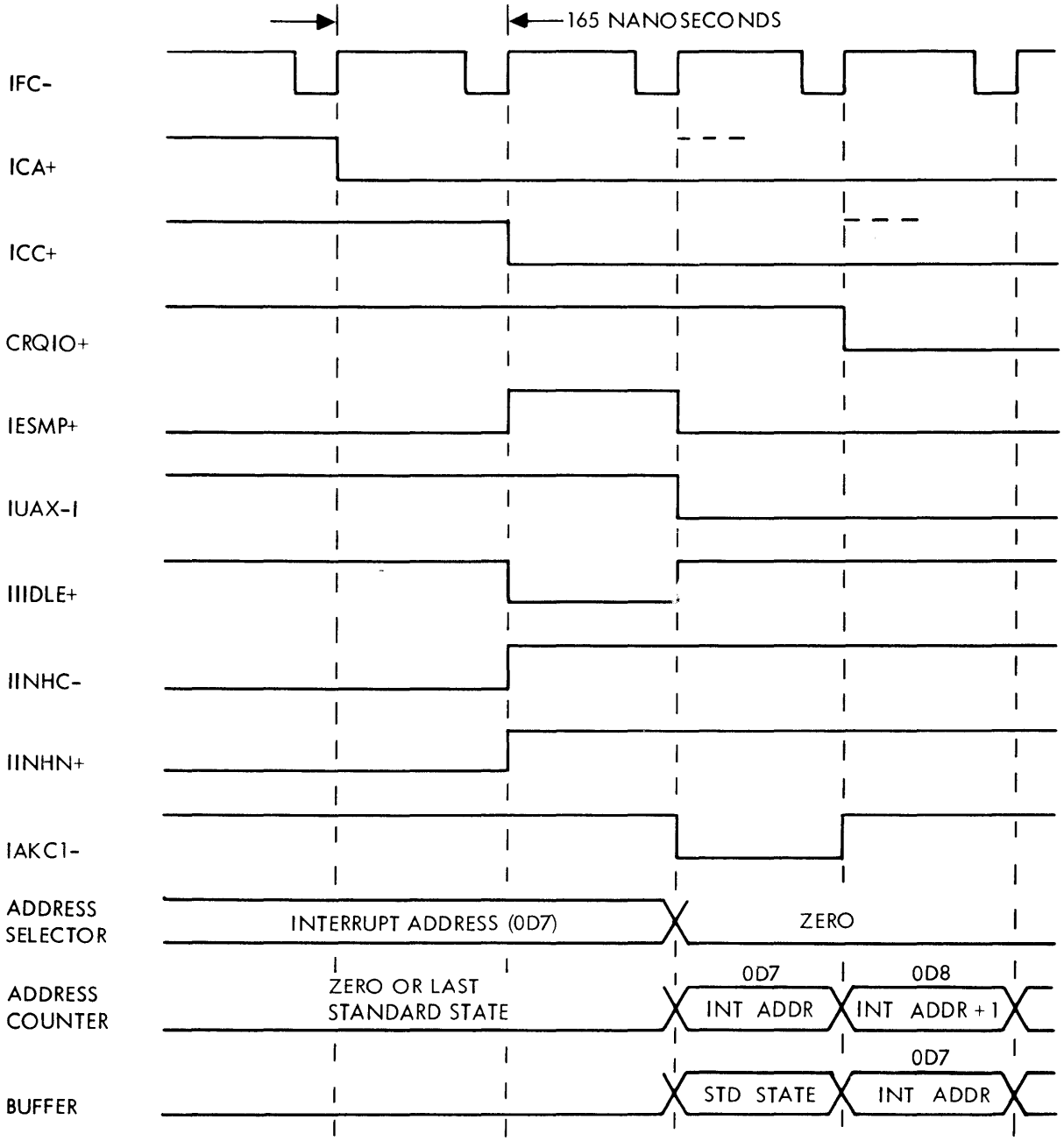
Figure 4-35. Sub-Operation 1 Timing Waveforms



NOTE: TIMING ASSUMES AN INTERRUPT CLOCK RATE OF 660 NANOSECONDS (IUCX-I).

VT11-2163

Figure 4-36. Sub-Operation 2 Timing Waveforms



VTII-2164

Figure 4-37. Sub-Operation 3 Timing Waveforms

SUB-OPERATION 4
Refer to figure 4-38 for timing waveforms of sub-operation 4. The low IAKC1- enables the central control to advance to its next microinstruction (address 0D1) in the interrupt service routine. In this condition, the central control waits until the interrupt address is accepted and transferred to the I/O register. At this time, the buffer (in the I/O control) is loaded with contents of address 0DC (table 4-26) and the address counter is incremented by one. This process of incrementing the address counter and loading the buffer continues until a waiting operation (IEWAIT + high) is specified by the I/O control microinstruction.

SUB-OPERATION 5
Refer to figure 4-38 for timing waveforms of sub-operation 5. When the I/O microinstruction at address 0DE (table 4-26) is loaded into the buffer, an I/O done signal (IDNC- low) is sent to the central control and the interrupt address is loaded into the I/O register. The interrupt operation then advances to the next microinstruction (address 0DF) and waits (IEWAIT + high) for the processor to complete its interrupt service routine.

SUB-OPERATION 6
Refer to figure 4-39 for timing waveforms of sub-operation 6. The central control responds to the low IDNC- by initiating a memory cycle to fetch the instruction at the interrupt address. This is accomplished using the I/O register contents as the memory address. When the central control advances to the first micro word of the instruction at the interrupt address, the interrupt flag is set (CINTF + high). A high CINTF + is sent to the I/O control indicating the central control has completed its interrupt service routine and is in the process of executing the interrupt macro routine.

SUB-OPERATION 7
Refer to figure 4-40 for timing waveforms of sub-operation 7. The I/O control responds to the high CINTF + by generating the I/O interrupt flag (IINTF + high). The high IINTF + enables the I/O control to complete its interrupt service routine by removing the processor I/O request (IRQC- high) and the clock inhibitor (IINHC- high). When returned to the idle condition (IEIDLE + high), the I/O control is available for new I/O operations.

4.7.9 Normal DMA Operations

Timing waveforms for a normal DMA operation are illustrated in figure 4-41. A normal DMA operation is initiated when a trap-in (TPIX-I) or trap-out (TPOX-I) request is received from the I/O bus. The external device (usually a buffered interlace controller) activates the trap signal on the positive-going edge of the interrupt clock (IUCX-I).

The trap request produces a high ITRPN + which causes IINHN + to go high. When the clock counter (ICA + and ICC +) reaches the sample state (both flip-flops reset) a high IESMP + permits recognition of the trap by making IIIDLE- low and enabling the address counter and buffer clocks with a high IINHC-. At the same time, a low IMXAD + causes the address selector to select the trap address. This address is generated by trap-address generation logic (in I/O control) that converts all trap requests to unique addresses.

At the clock pulse following IESMP + going high, the address counter is loaded with the starting address of the DMA operation. The address counter previously contained either address zero or the last address of the previous I/O operation. In either case this data, which is loaded into the buffer, causes the I/O control to become busy (IEIDLE- high).

When the I/O control becomes busy, the loading signal ILDAD- goes high (inactive) permitting the address counter to increment. At the same time, the I/O clock inhibitor IINHC- is latched off (high) and the sampling control IESMP + is turned off (low). The address selector selection then reverts to the processor address source, an acknowledgment (IUAX-I low) is sent to the I/O bus, the clock counter stops with ICA + and ICC + low, and IIIDLE- goes high.

At each succeeding clock pulse, the address counter is incremented as a new control word is loaded into the buffer. For a trap-in operation, the memory address is first obtained and loaded into the I/O register (in I/O data loop). The I/O control then enters a waiting state and requests a processor memory cycle (IEWAIT + and IRQM + high). When the request is acknowledged (MAKIO + high), the I/O control leaves the waiting state and obtains the trap-in data. This data is transferred from the I/O bus to the memory data bus via the I/O multiplexor (in I/O data loop) and the data multiplexor (in memory control).

At the conclusion of the memory cycle, the clock counter (in I/O control) is advanced with a code of 010 by buffer signals IEFn + (0-2). The next clock pulse loads all zeros into the buffer; thus, placing the I/O control back in its idle state (IEIDLE- low). Clocks for the address counter and buffer are inhibited until further I/O requests are received.

Trap-out operations are similar to trap-in; however, the I/O control obtains the memory address and then transfers memory data to the I/O bus via the I/O latch (in memory control), I/O multiplexor, and I/O register (in I/O data loop).

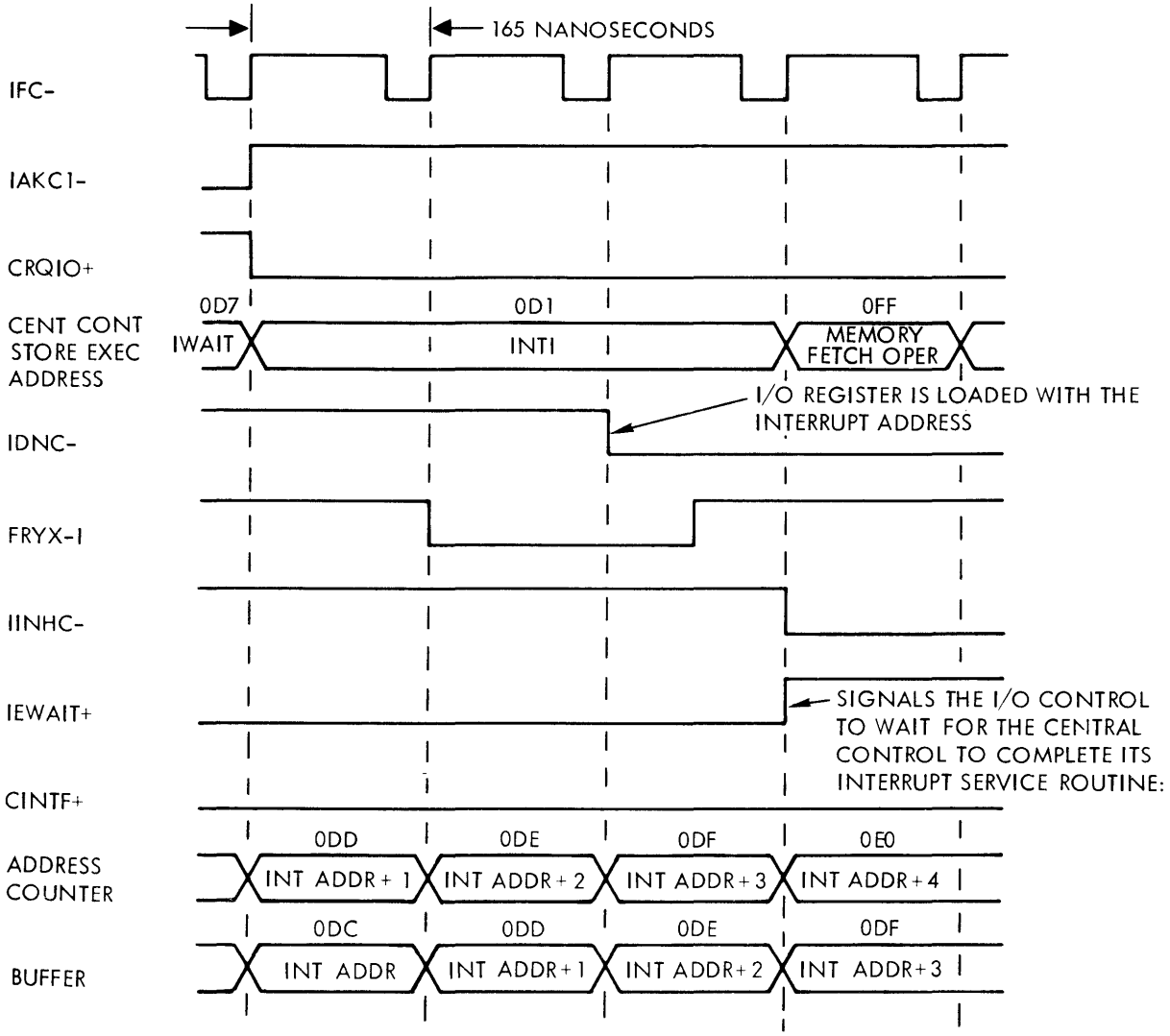


Figure 4-38. Sub-Operations 4 and 5 Timing Waveforms

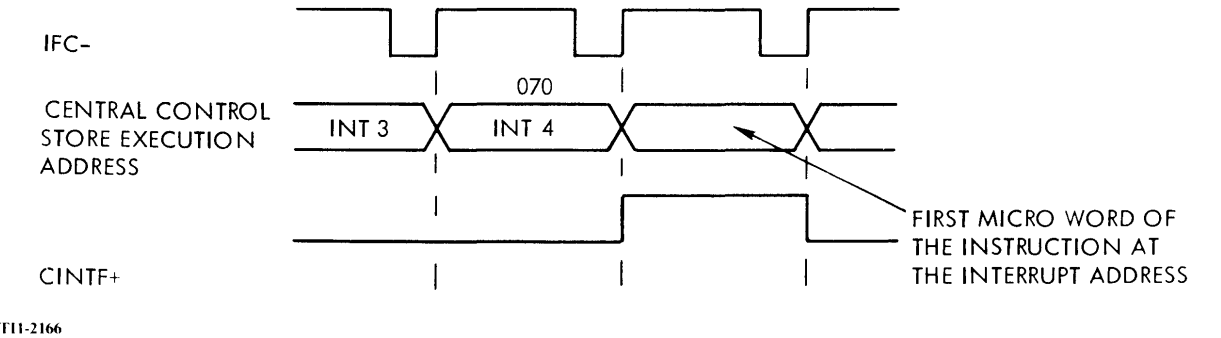
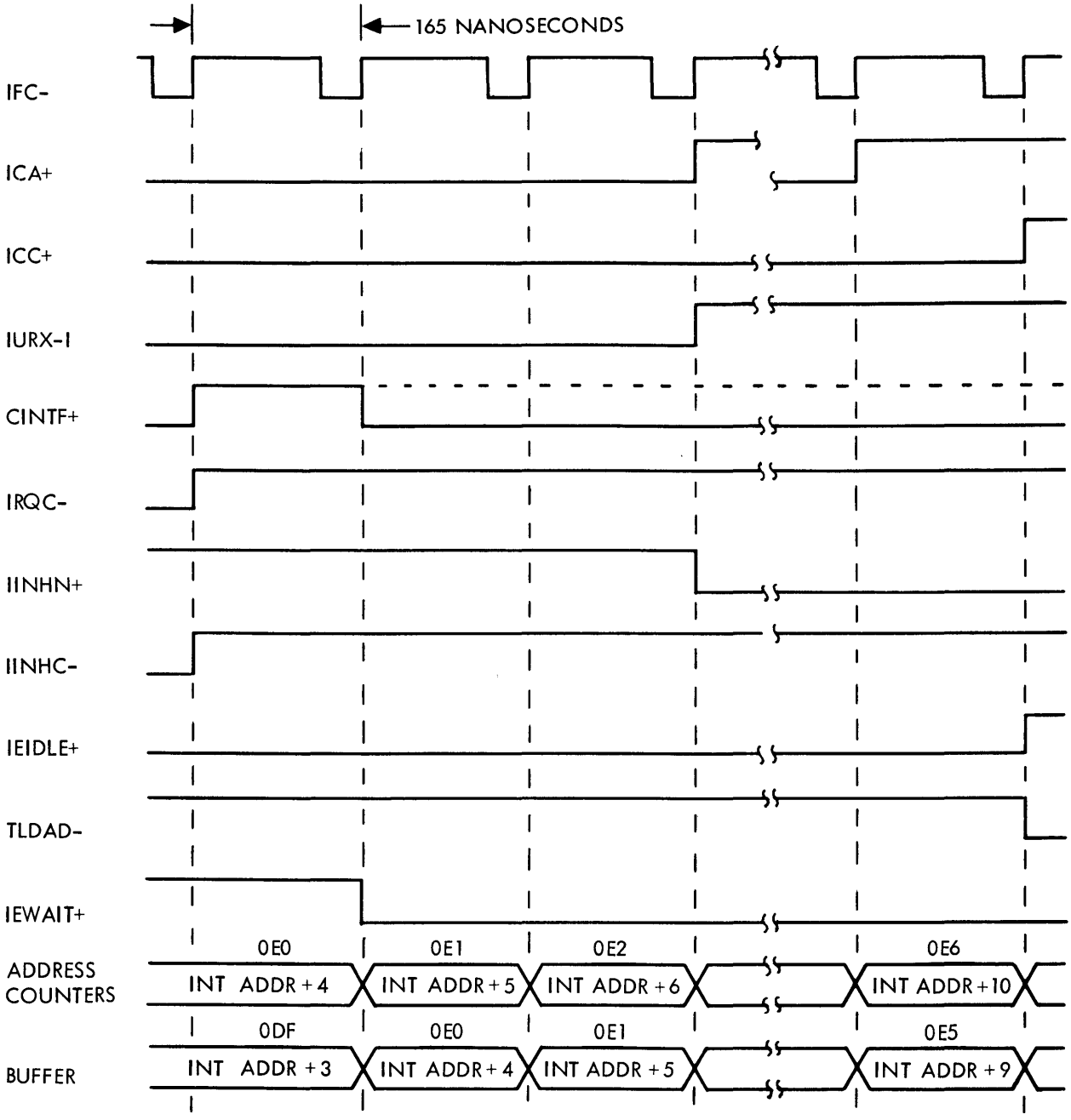


Figure 4-39. Sub-Operation 6 Timing Waveforms



VTII-2167

Figure 4-40. Sub-Operation 7 Timing Waveforms

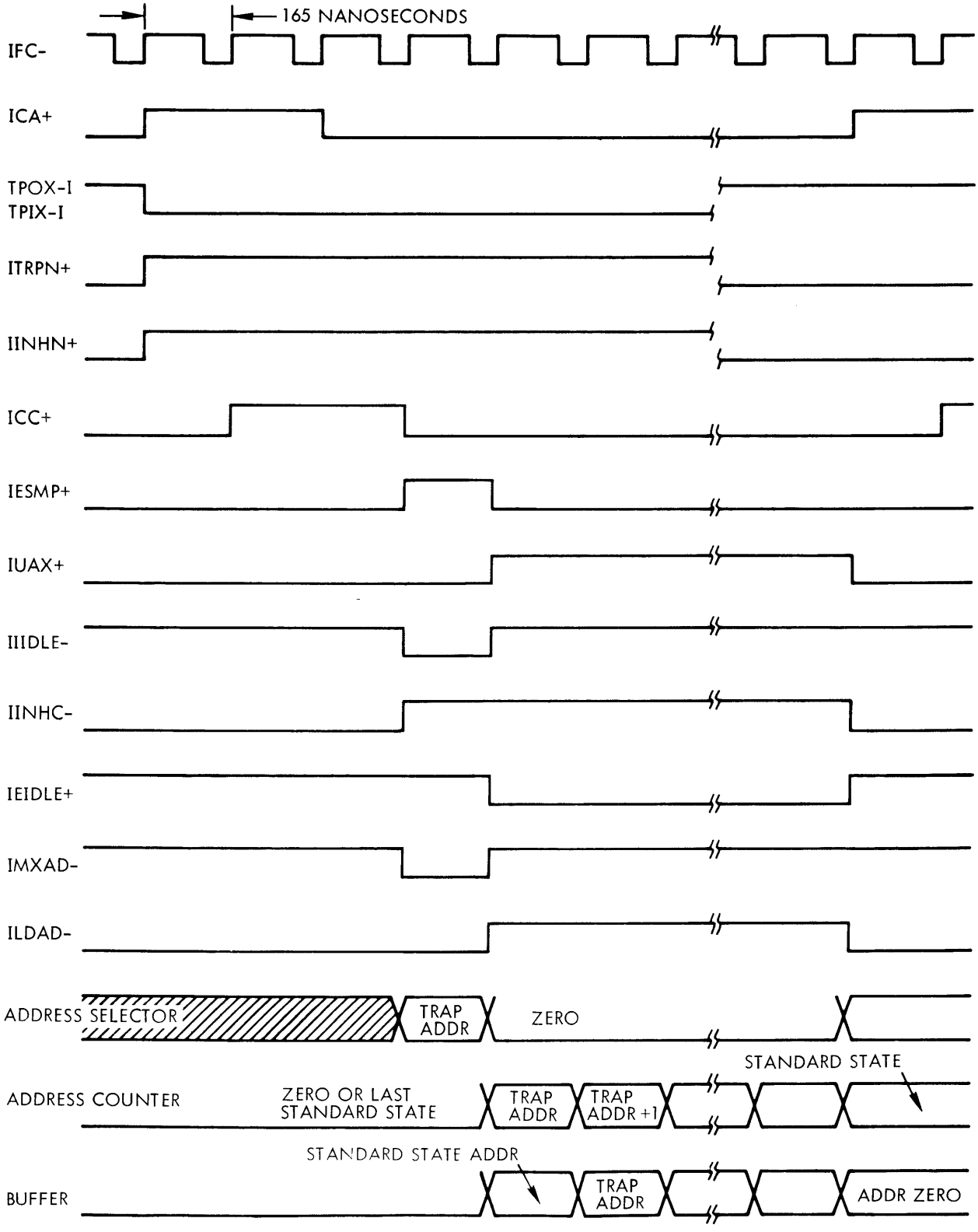
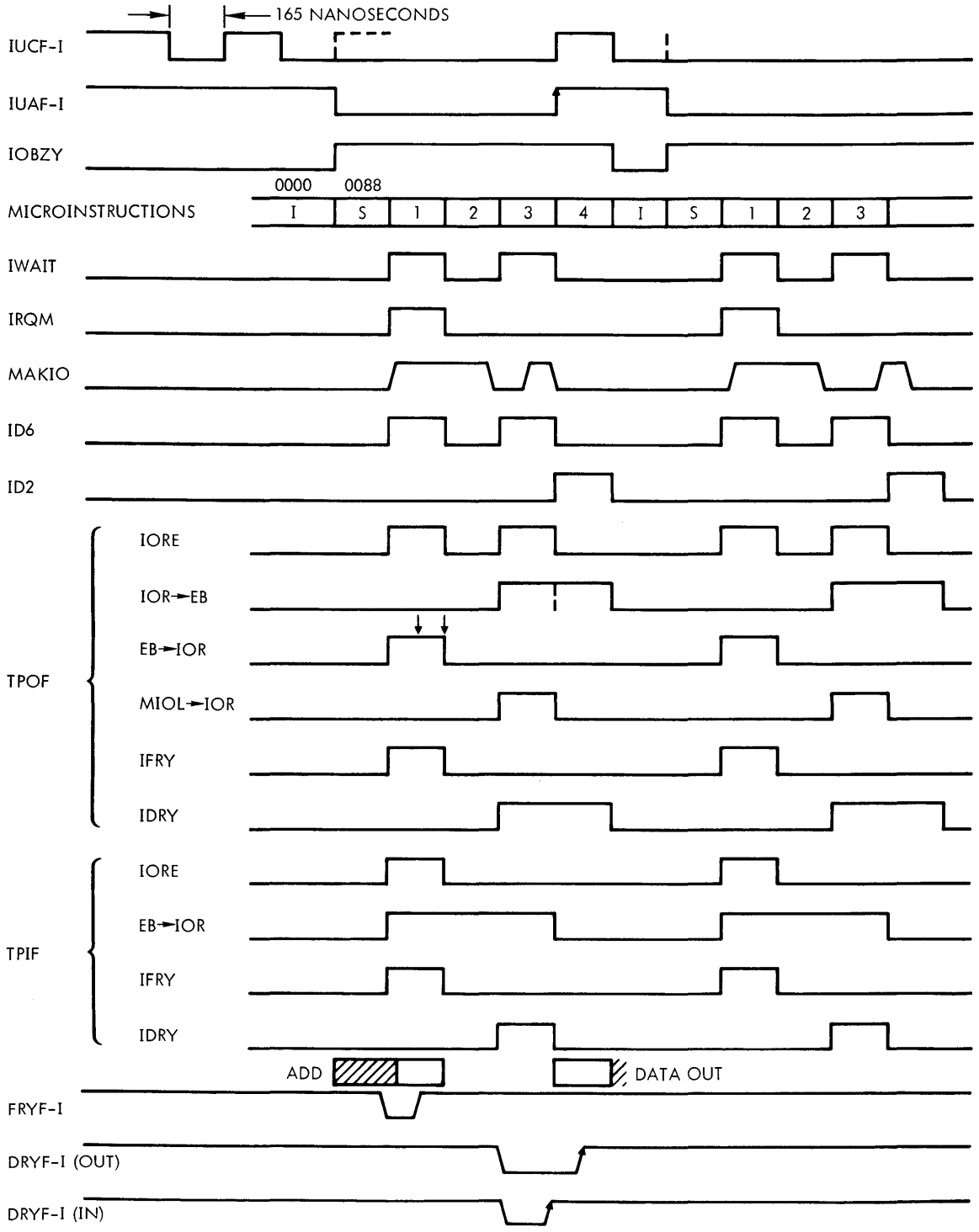


Figure 4-41. Normal DMA Waveforms

4.7.10 High-Speed DMA Operations

Timing waveforms for a high-speed DMA operation are illustrated in figure 4-42. A high-speed DMA operation is similar to the normal DMA; however, a faster interrupt clock (IUCF-I) and I/O timing are used. Furthermore, a separate set of lines are used for functions and data ready, interrupt acknowledgment, trap-in, and trap-out.



VTII-1660

Figure 4-42. High-Speed DMA Waveforms

4.8 CONTROL-STORE WORD FORMAT

Figure 4-43 shows the 64-bit word format of the control store. In the Microprogramming Guide, all fields of the 64-bit word are designated with two letters making them consistent with the mnemonics used in the microprogramming assembler (MIDAS). This is done by adding the letter F to all the one-letter designated fields except the A and B fields. The A and B fields are designated AA and BB, respectively.

Listed in descending bit order, each field of the word is described in the following paragraphs.

TS Field

- Provides a component of the control-store address used when a conditional test is not met.
- Provides a component of the address used when not conditional testing if the field is not used for any of the following:
  - Selects a field from the instruction register for use in the A or B fields of subsequent microinstructions.
  - Selects interrupts which are to be enabled.
  - Provides a portion of the I/O-sequence starting address for I/O requests.
  - Selects another control store when a page-jump microinstruction is executed (section 4.9).

AF Field

- Contributes to the five most-significant bits of the control-store address except for decoding addressing (section 4.9).

MS Field

- Provides the four least-significant bits of the control-store address for normal addressing (section 4.9).
- Masks the four least-significant bits for field-selection addressing (section 4.9).

MT Field

- Masks bit 4 of the control store address for field-selection addressing.
- Provides a portion of the I/O-sequence starting address for I/O requests.

FS Field

- Selects a five bit field from the instruction register for field-selection addressing.

T Field

- Specifies no conditional testing.
- Specifies conditional testing of a true condition.
- Specifies conditional testing of a false condition.

S Field

- Defines unconditional or conditional memory control.
- Defines page-jump control.

G Field

- Specifies condition to be tested for conditional-test addressing and conditional-memory control.
- Controls status sampling and control of overflow flag.
- Controls transfer of instruction buffer to instruction register.
- Controls selection of decoding addressing.
- Controls selection of page-jump operation.

MR Field

- Masks the most-significant bit of the instruction-register 4-bit field selected by the TS field when specified by the AB field.
- Controls (in conjunction with the AB and W fields) the transfer of address 1110 or 1111 into the B field depending on data-loop conditions.
- Provides a portion of the I/O-sequence starting address for I/O requests.

AB Field

- Specifies the following types of data for A and B field of next microinstruction.
  - Control-store output
  - Field selection from instruction register
  - Previous values contained in A and B fields
  - Value for B field depending on data-loop conditions
- Provides a portion of the I/O-sequence starting address for I/O requests.

IM Field

- Specifies non-memory operations consisting of:
  - I/O requests
  - Interrupt flag (set or reset)
  - Supervisor mode (set or reset)
  - Special transfer of ALU output to instruction buffer and memory input latch.
- Specifies memory operations consisting of:
  - Conditional or unconditional start or override of memory cycle
  - Address source for memory cycle
  - Operation to be performed (reading or writing)

LB Field

- Specifies data source for B input of ALU from:
  - General-purpose registers (register file B)
  - B multiplexor
  - Instruction register 16-bit control store literal (fields M, C, WR, SC, V, W, X, SH, B)
- Specifies WCS control functions

LA Field

- Specifies data source for A input of ALU from:
  - General-purpose registers (register file A)
  - General-purpose registers (file A) shifted left
  - General-purpose registers (file A) shifted right
  - Program counter

R Field

- Specifies destinations for ALU output data to:
  - Program counter
  - Operand register
  - Shift counter
  - Key register (in data loop)
- Specifies incrementation for:
  - Program counter
  - Shift counter

F Field

- Specifies ALU operations.

M Field

- Specifies ALU mode as arithmetical or logical.

C Field

- Specifies ALU carry input as:
  - one
  - zero
  - stored carry
  - stored carry complement
- Forms part of the 16-bit control-store literal.

WR Field

- Specifies the loading of ALU output data into a general-purpose register specified by A field.
- Forms part of the 16-bit control-store literal.

SC Field

- Specifies shifting of operand register.
- Forms part of the 16-bit control-store literal.

V Field

- Controls the transfer of general-purpose register sign to shift flag.
- Forms part of 16-bit control-store literal.

W Field

- Controls the transfer of ALU output sign to quotient sign flag.
- Determines direction of operand register shifting.

(continued)

- Determines contents of B field for next microinstruction in conjuction with AB field.
- Forms part of 16-bit control-store literal.

X Field

- Determines that operand register bit 0 is to be one of the following during a shift operation:
  - Operand register bit 15
  - General-purpose register bit 15
  - ALU output bit 15
  - Zero
- Determines that operand register bit 15 is to be one of the following during a right-shift operation:
  - Operand register bit 0
  - General-purpose register bit 0
  - Operand register bit 15
  - Shift flag
- Resets interrupt flag.
- Sends jump signal to external memory protection option.
- Forms part of 16-bit control-store literal.

SH Field

- Sets A-input of ALU to all ones or all zeros.
- Selects special ALU function.
- Determines that general-purpose register bit 15 is to be one of the following during a left-shift operation:
  - General-purpose register bit 15 (remains the same)
  - General-purpose register bit 14
- Determines that general-purpose register bit 0 is to be one of the following during a left-shift operation:
  - Zero
  - General-purpose register bit 15
  - Operand register bit 15
- Determines that general-purpose register bit 15 is to be one of the following during a right-shift operation:
  - Multiplication-sign flag
  - General-purpose register bit 0
  - General-purpose register bit 15
  - Operand register bit 0
  - Zero
- Forms part of the 16-bit control-store literal.

B Field

- Specifies one of 16 general-purpose registers which can be applied to the B-input of the ALU.
- Specifies one of the following eight registers that can be applied to the B-input of the ALU:
  - Operand register
  - Memory input latch
  - I/O register
  - Processor status word
  - Operand-register right byte with sign extended
  - Operand-register left byte with sign extended
  - Operand-register right byte without sign extended
  - Operand-register right byte shifted left eight-bit positions.
- Forms part of the 16-bit control-store literal.

A Field

- Specifies one of 16 general-purpose registers which can be applied to the A-input of the ALU.
- Specifies one of 16 general-purpose registers that can receive ALU output data.

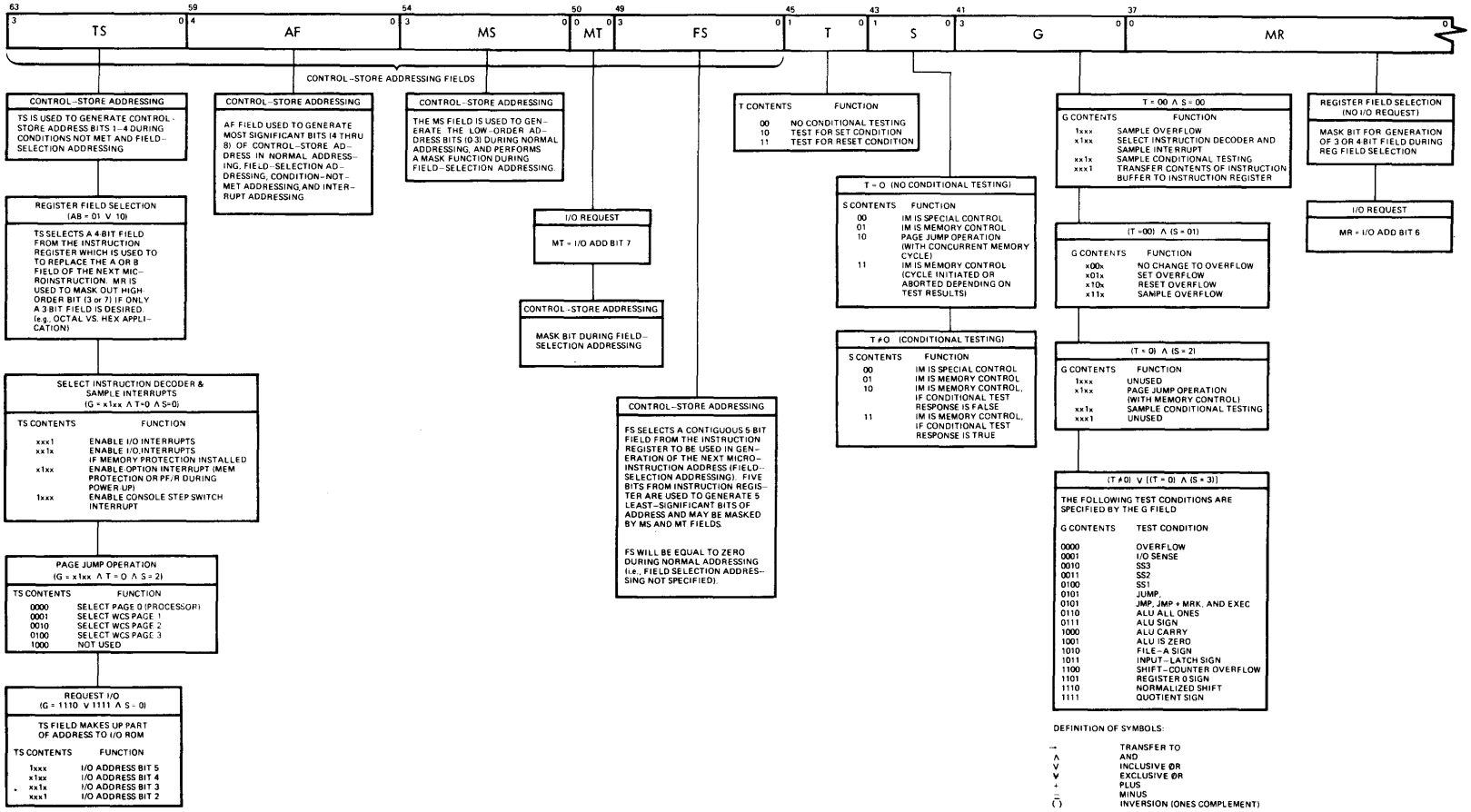
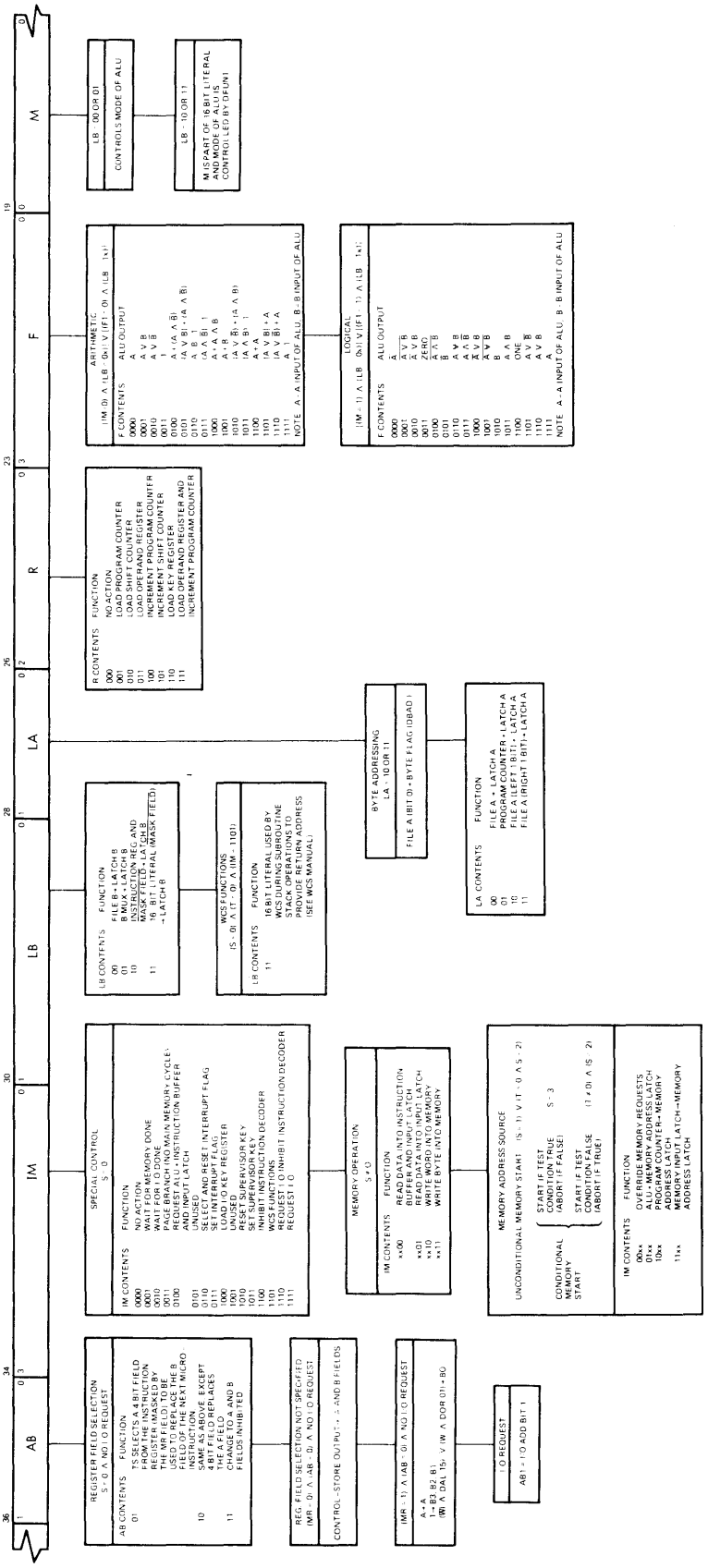
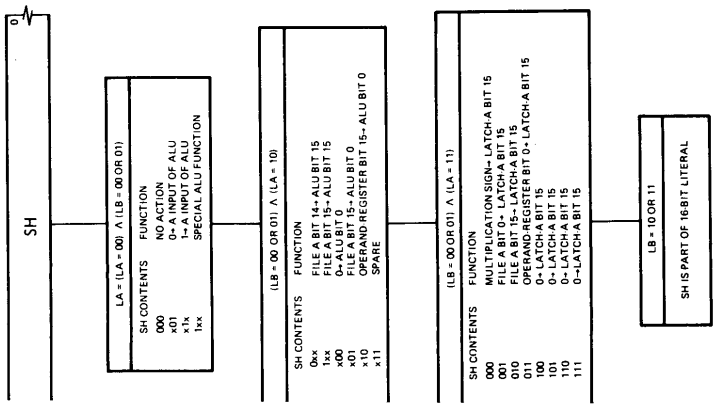


Figure 4-43. Control-Store Word Format *(Sheet 1 of 4)*



VTI3-284B

Figure 4-43. Control-Store Word Format *(Sheet 2 of 4)*



varian data machines
THEORY OF OPERATION

4.9 CONTROL-STORE ADDRESSING

Unlike conventional computer instructions, which execute sequentially until altered by a branch instruction, each microinstruction of the Varian 70 series computers must specify the address of the next microinstruction. By specifying this address, the functions of instruction decoding, conditional testing, interrupt handling, and normal microinstruction sequencing are accomplished.

The computer's basic instruction set is contained in the 512 64-bit words of the control store. Up to three 512-word control stores can be added to extend the computer's instruction set.

The 9-bit address CEADn– (0-8) selects one of the 512 addresses in the control store (read-only or writable). If a microinstruction specifies an address in another control store, a page-jump microinstruction is executed (T field contains 00, S field contains 10, and G field contains x1xx). The particular control store is selected by the 4-bit TS field. When a system reset condition occurs, all bits of the control store address are set to one and page zero (read-only control store) is selected.

The following addressing modes comprise control store addressing:

a. Normal

b. Normal/TS field

c. Field selection

d. Conditional testing

e. Interrupt

f. Decoding

4.9.1 Normal Addressing

4.9.2 Normal/TS Field Addressing

The normal/TS field mode can be used when none of the following conditions are specified:

a. Register-field extraction (AB field contains 01 or 10).

b. Interrupt allowed (S and T fields contain 00, and G field contains x1xx).

c. I/O request (S field contains 00, and IM field contains 111x).

d. Page-jump microinstruction specified (T field contains 00, S field contains 10, and G field contains x1xx).

The address is formed by the inclusive OR of the TS field and bits 1 through 4 of the normal address (figure 4-45).

4.9.3 Field-Selection Addressing

In the field-selection mode of addressing, the instruction register contents C2In– (0-15) and various processor flags are used to produce the five least significant bits of the control store address CEADn– (0-4). This permits up to 32 branching operations based on instruction register contents. The processor flags consist of: interrupt flag (CINTF +), shift flag (DSB–), byte address flag (DBAD–), and control-panel step flag (NSTP–).

Field selection addressing is used when the FS field does not contain 0000 (when the FS field contains 0000, normal addressing or normal/TS field addressing are used). Table 4-27 lists the signals that produce control store address bits 0 through 4 for all values of the FS field. The listed signals are applied to the instruction field selector. Masking of the five address bits is performed by the MS and MT fields causing a zero bit in these fields to produce a zero in a corresponding address bit (CEADn– high). Address bits 0 through 3 are masked by the MS field and bit 4 by the MT field.

When an I/O request is issued (S field contains 00 and IM

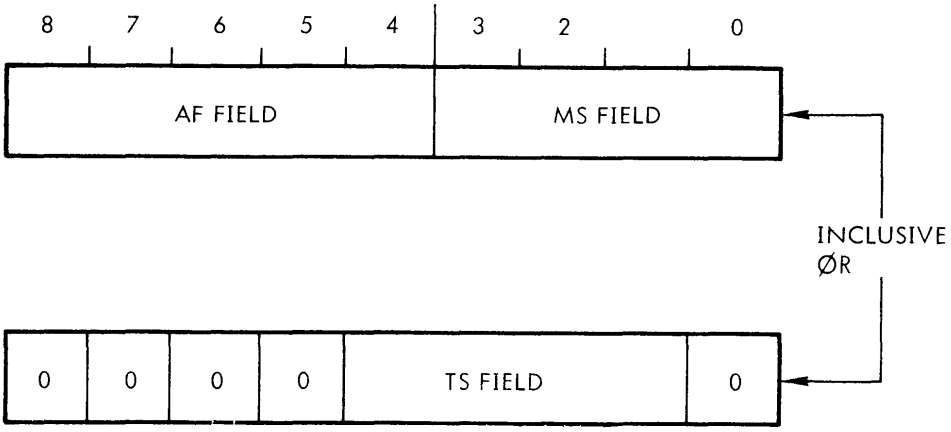


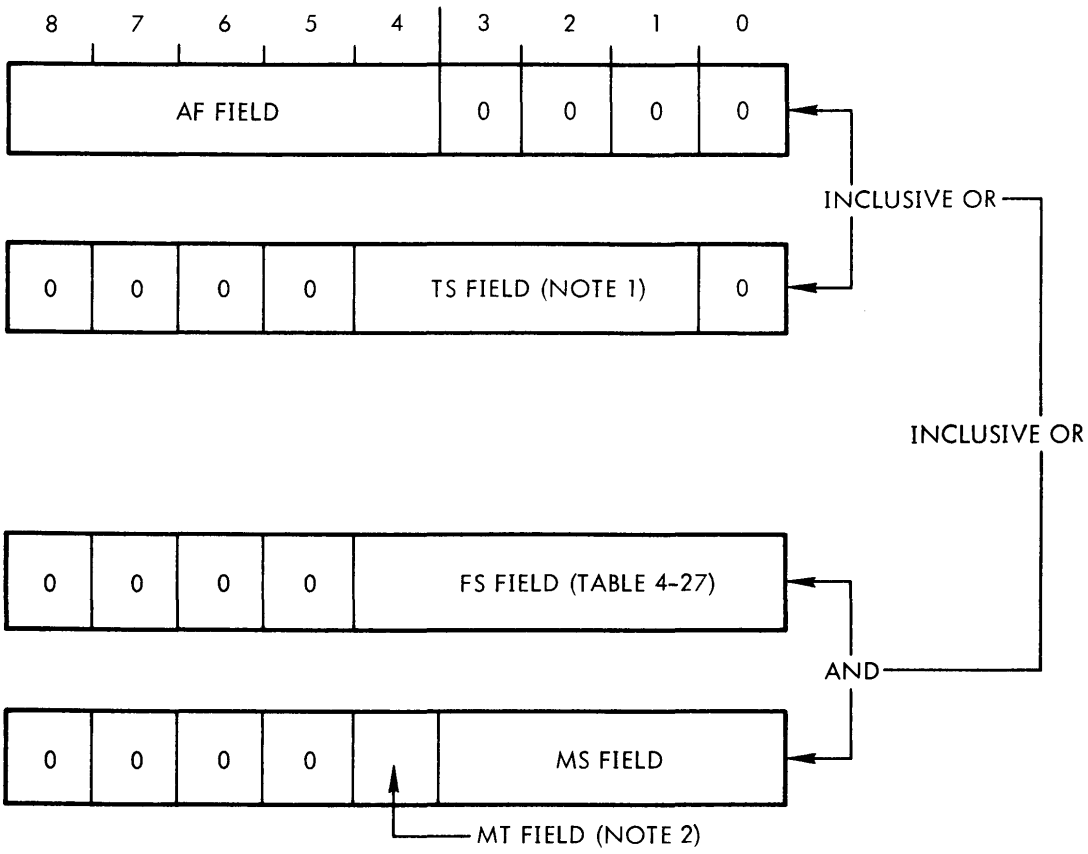
Figure 4-45. Normal/TS Field Addressing

Table 4-27. Field-Selection Addressing Contribution

FS FIELD BITS	SIGNAL SOURCE OF CONTROL-STORE ADDRESS BITS 0-4*				
3 2 1 0	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
0 0 0 0	0	0	0	0	0
0 0 0 1	0	0	0	0	CINTF+
0 0 1 0	0	C2I01-	0	DSB-	DBAD-
0 0 1 1	0	0	0	0	NSTP-
0 1 0 0	C2I04-	C2I03-	C2I02-	C2I01-	C2I00-
0 1 0 1	C2I05-	C2I04-	C2I03-	C2I02-	C2I01-
0 1 1 0	C2I06-	C2I05-	C2I04-	C2I03-	C2I02-
0 1 1 1	C2I07-	C2I06-	C2I05-	C2I04-	C2I03-
1 0 0 0	C2I08-	C2I07-	C2I06-	C2I05-	C2I04-
1 0 0 1	C2I09-	C2I08-	C2I07-	C2I06-	C2I05-
1 0 1 0	C2I10-	C2I09-	C2I08-	C2I07-	C2I06-
1 0 1 1	C2I11-	C2I10-	C2I09-	C2I08-	C2I07-
1 1 0 0	C2I12-	C2I11-	C2I10-	C2I09-	C2I08-
1 1 0 1	C2I13-	C2I12-	C2I11-	C2I10-	C2I09-
1 1 1 0	C2I14-	C2I13-	C2I12-	C2I11-	C2I10-
1 1 1 1	C2I15-	C2I14-	C2I13-	C2I12-	C2I11-

*THE LISTED SIGNAL SOURCES ARE INPUT TO THE INSTRUCTION FIELD SELECTOR (SECTION 4.3). THE SIGNALS (WHEN SELECTED) ARE INVERTED BY THE INSTRUCTION FIELD SELECTOR TO BECOME CFSAn+ (0-4) OR CFSBn+ (0-4)

VTII-1664



NOTES:

1. THE TS FIELD IS NOT USED TO PRODUCE ADDRESS BITS 1 THROUGH 4 IF ONE OF THE FOLLOWING CONDITIONS IS TRUE:
   A. REGISTER FIELD EXTRACTION (AB FIELD CONTAINS 01 OR 10)
   B. INTERRUPTS ALLOWED (S AND T FIELDS CONTAIN 00,IM FIELD CONTAINS 111x)
   C. I/O REQUEST (S FIELD CONTAINS 00, IM FIELD CONTAINS 111x)
   D. PAGE JUMP (T FIELD CONTAINS 00, S FIELD CONTAINS 10, G FIELD CONTAINS x1xx)
   E. CONDITIONAL TESTING IS SPECIFIED (T FIELD DOES NOT CONTAIN 00)
2. WHEN AN I/O REQUEST IS ISSUED (S FIELD CONTAINS 00, IM FIELD CONTAINS 111x, THE MT FIELD IS IGNORED AND ADDRESS BIT 4 IS MASKED TO ZERO

VTII-1665

Figure 4-46. Formation of Field-Selection Addressing

4.9.4 Conditional-Test Addressing

Two addresses must be specified when conditional test operations are performed: one to be used if the condition is met and one to be used if the condition is not met. Conditional-test addressing is specified when the T field does not contain 00. The T field contains binary 10 if the condition to be met is true (for example, Jump if Sense Switch 2 Set), and a binary 11 if the condition to be met is false (for example, Jump if Sense Switch 2 Not Set). The condition to be tested is specified by the G field (table 4-28).

If a condition is met, the address is produced by field selection addressing. If a condition is not met, the address is produced from the AF and TS fields as shown in figure 4-47.

4.9.5 Interrupt Addressing

Control-store address bits 0 through 3 are produced by the interrupt logic (in I/O control) and bits 4 through 8 by the AF field, if all of the following conditions are true:

a. Interrupts are allowed (T and S fields contain 00, and G field contains x1xx).

b. An interrupt is active.

c. The proper class of interrupts is enabled by an appropriate bit in the TS field (table 4-29).

Figure 4-48 shows the interrupt addressing format along with codes for the three types of interrupt conditions.

Table 4-28. Test Conditions

Condition	G Field
Overflow	0000
I/O sense response	0001
Sense switch 3	0010
Sense switch 2	0011
Sense switch 1	0100
Jump, Jump and Mark, and Execution instructions	0101
ALU output all ones	0110
ALU output sign	0111
ALU carry	1000
ALU output is zero	1001
Stored sign of register file A (DSB+)	1010
Sign of memory input latch	1011
Shift-counter overflow	1100
Sign of general-purpose register 0	1101
Normalized shift	1110
Quotient sign	1111

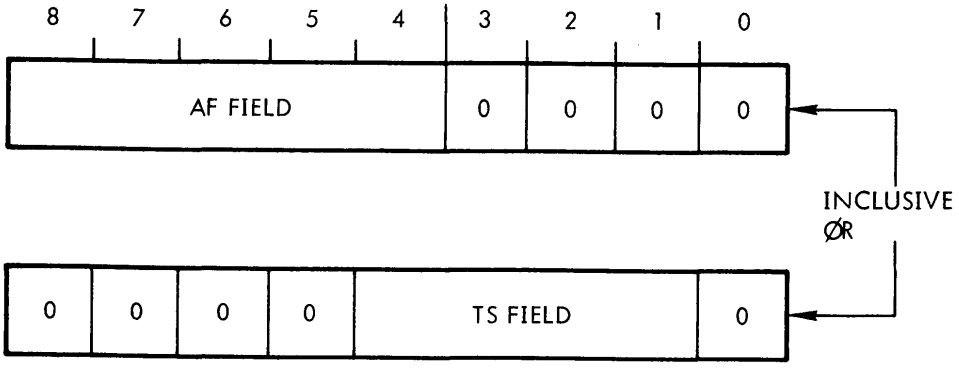
Table 4-29. TS-Field Interrupt Functions

TS-Field Bits				Interrupts
3	2	1	0	
x	x	x	1	Enables I/O interrupts
x	x	1	0	Enables I/O interrupts only if memory protection option is installed
x	1	x	x	Enables interrupt from memory protection option
1	x	x	x	Enables control-panel step interrupt

Note: x indicates irrelevant bit.

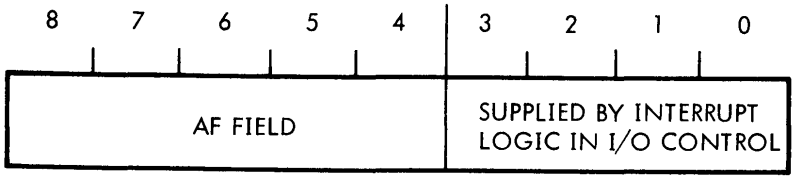
4.9.6 Decoding Addressing

Decoding addressing performs preliminary instruction decoding to determine the classification of an instruction. Detailed decoding is performed later by field-selection addressing after instruction buffer contents are transferred to the instruction register. The preliminary instruction decoding is implemented by the instruction decoder and instruction decoding logic. These elements translate the 16-bit instruction from the instruction buffer to the 9-bit control store address CEADn– (0-8).



VTII-1666

Figure 4-47. Addressing When Condition is Not Met



AF FIELD	INTERRUPT LOGIC	INTERRUPT CONDITION
0 1 1 0 1	0 0 0 0	ABORT INTERRUPT FOR DMA
0 1 1 0 1	0 0 0 1	START OF INTERRUPT SEQUENCE
0 1 1 0 1	0 1 1 1	INTERRUPT WAITING FOR DEVICE TO SEND MEMORY ADDRESS

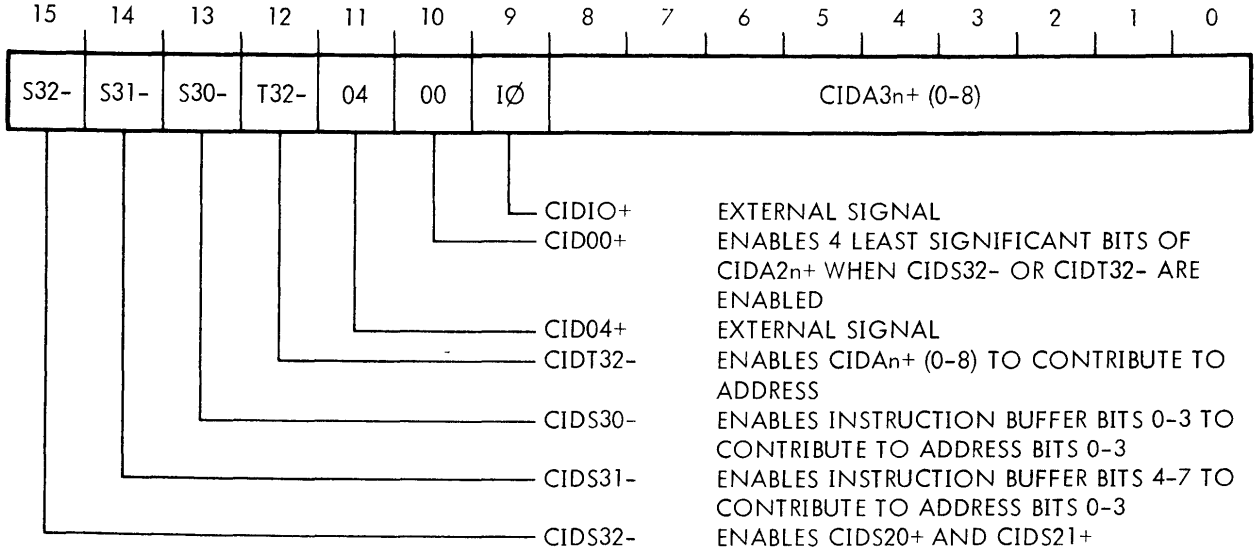
VTII-1667

Figure 4-48. Interrupt Addressing

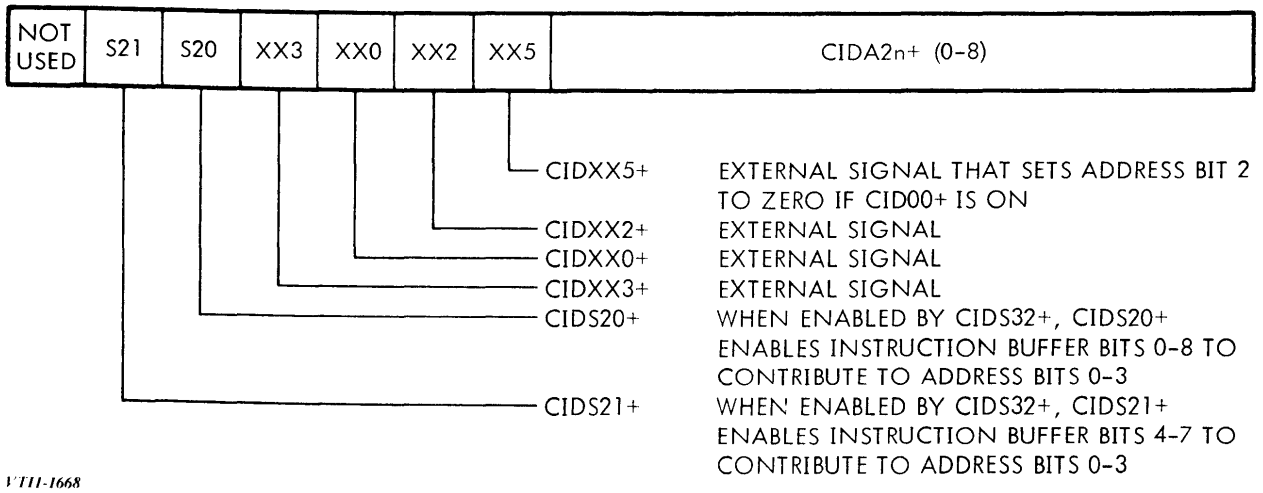
The instruction decoder consists of two ROMs each containing sixteen 16-bit words. An option to the WCS option provides selection of read/write memory arrays to permit alternate decoding methods. The first ROM decoder uses instruction buffer bits 12 through 15 as an address; the second ROM decoder uses bits 8 through 11. Output-signal formats for the two ROM decoders are shown in figure 4-49.

Decoding addressing is used when the T and S fields contain 00 and the G field contains x1xx. If an interrupt is present, decoding addressing is inhibited and interrupt addressing is used. Decoding addressing is also inhibited if the IM field contains 11x0. When the IM field contains 11x0 and no interrupts are present, field-selection addressing is used.

FIRST ROM DECODER DECODED FROM INSTRUCTION BUFFER BITS 12-15



SECOND ROM DECODER DECODED FROM INSTRUCTION BUFFER BITS 8-11



VTII-1668

Figure 4-49. Output Formats of ROM Decoders

Figure 4-50 shows the decoding address components. The nine bits from the first ROM decoder are always used in the address. The five most-significant bits from the second ROM decoder are inclusively ORed into address bits 4 through 8 when CIDT32- or CIDS32- from the first decoder is true (low). The four least-significant bits from the second ROM decoder are inclusively ORed into address bits 0 through 3 when either one of the following conditions exist in the first decoder:

a. CIDT32- is true (low) and CID00 + is ture (high).
b. CIDS32- is true (low) and CID00 + is true (high).

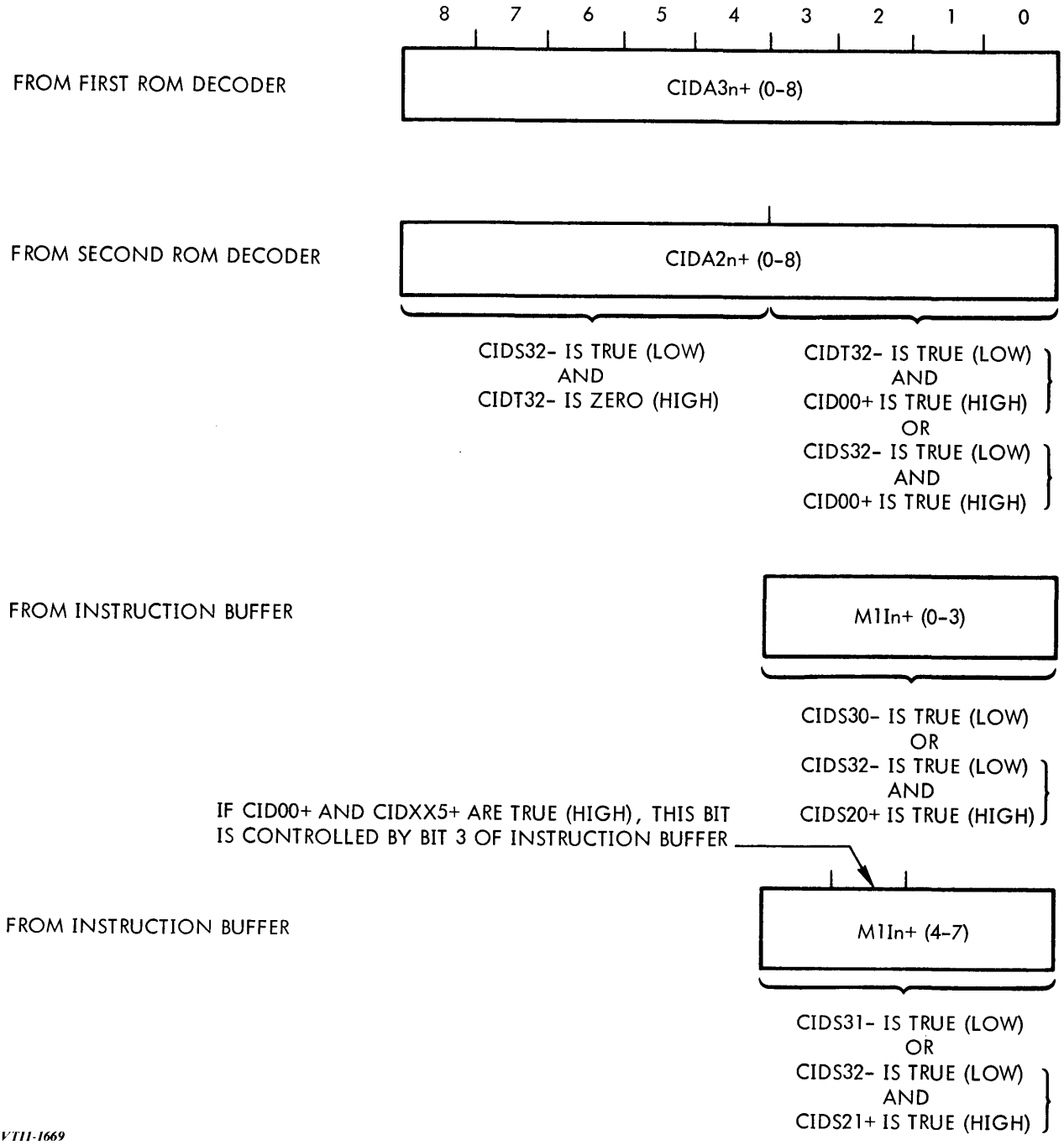


Figure 4-50. Decoding Address Components

Instruction buffer bits 0 through 3 are inclusively ORed into address bits 0 through 3 when either one of the following conditions exist:

a. CIDS30– is true (low)

b. CIDS32– is true (low) and CIDS20 + is true (high).

Instruction buffer bits 4 through 7 are inclusively ORed into address bits 0 through 3 when either one of the following conditions occur:

a. CIDS31– is true (low).

b. CIDS32– is true (low) and CIDS21 + is true (high).

One exception to these conditions for instruction buffer bits 4 through 7 is that address bit 2 is controlled by instruction buffer bit 3 when CID00 + and CIDXX5 + are true (high).

4.10 CONTROL-PANEL CIRCUITS

The control-panel circuits (figure 4-51), located on the control-panel board (44P0645), interface with the processor through the I/O bus. The interface circuits consist of the I/O control buffer, I/O bus drivers, and I/O terminators. Commands are decoded from the I/O bus by the I/O control buffer. The I/O control buffer enables either display register data or pulse-register data onto the I/O bus, or gates I/O bus data to the display-select register or the toggle register. The 16-bit display register provides temporary storage for either switch status or I/O bus data from the processor. The 4 bit pulse register sends signals to the processor that indicate when the DISPL, ENTER, START, or BOOT switches have been pressed. The 4-bit toggle register stores the states of the three sense switches and the STEP/RUN switch. In conjunction with the run mode control, the pulse and toggle registers control the STEP and RUN indicators. When the computer is halted or in the run mode, the blink oscillator causes the RUN indicator to blink. The display-select register signals the processor when one of the internal machine registers is to be altered or displayed from the control panel.

The interrupt control circuit applies interrupts to the processor when the INT switch is pressed. The 4 bit file-address register designates which one of the 16 register files are to be altered or displayed from the control panel. Signals from control-panel switches are applied to pull-up and debounce circuits. Debounce circuits eliminate the effects of extra signal transitions caused by switch contacts. Data display drivers and light emitting diodes receive data-inputs from the display register.

Detailed descriptions of the control-panel circuits are provided in the following subsections. Page numbers of logic diagram 91B0406 (System Maintenance Manual) are in parentheses in the circuit blocks of figure 4-51.

4.10.1 File Address Register

The file address register stores the 4-bit addresses that select one of the 16 register files for control-panel alteration or display. The register can be directly set by the four REG SELECT switches (NFA1–, NFA2–, NAF4–, and NFA8–), cleared by the CLEAR switch (NFACLR–), and incremented by the INCR switch (NFAINC–). NFAINC– is debounced by a Schmitt trigger circuit as well as the debounce circuit.

The contents of the file address register can be gated onto the I/O bus with the input-function transfer NFI + from the I/O control buffer. When the instruction register is selected for display, NIR + is high and an input-function transfer clears the file address register and EB02-1 is low (true).

4.10.2 Toggle Register

The toggle register stores the states of the three SENSE switches (NS1–, NS2–, and NS3–) and the STEP/RUN switch (NSP–). Each input is exclusively ORed with a complement output of its corresponding register flip-flop. The truth table for the toggle register is in table 4-30. The register can be cleared with a system reset (NRST– low).

The toggle register clock N021– comes from either of two sources:

a. A low NTCK– is generated by the clock generator when any of the four switches are pressed (SENSE switches or STEP/RUN switch).

b. A high output-function transfer NFO + from the I/O control buffer and a high data ready NDRY + from the I/O bus produce a low NTCK1– clock.

NFO + permits the program to change the state of the three SENSE-switch flip-flops in the toggle register. This is accomplished by receiving I/O bus signals NEB0n + (1-3) and enabling them with NFO + to simulate the switch being pressed (using open-collector gates).

Table 4-30. Toggle Register Truth Table

Switch	Before Clock NSi-	NSiF +	NOii +	After Clock NSiF +
Not pressed (open)	H	L	L	L
Not pressed (open)	H	H	H	H
Pressed (closed)	L	L	H	H
Pressed (closed)	L	H	L	L

Note: H = high; L = low; i = 1, 2, or 3; and ii = 17, 18 or 19.

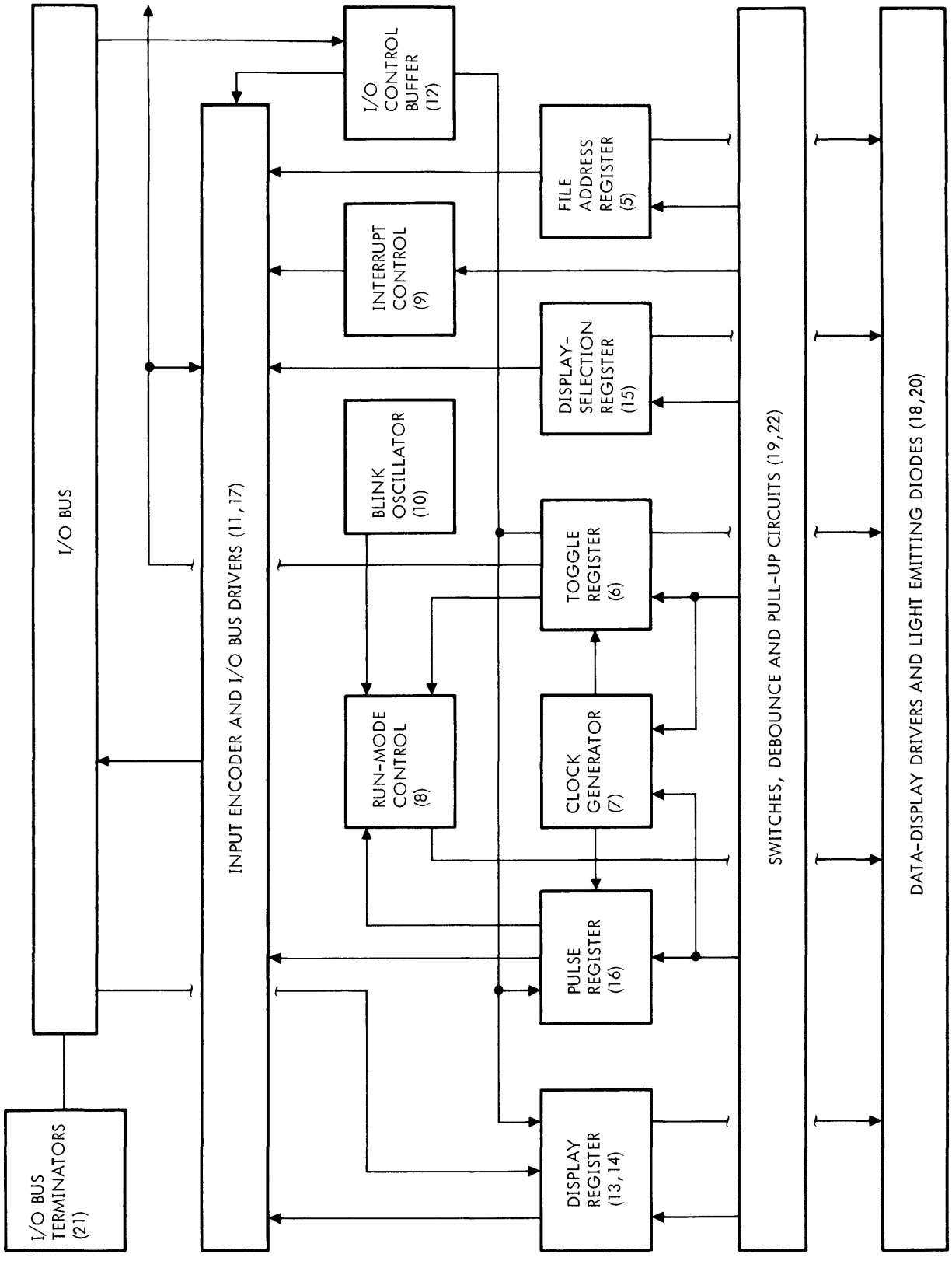


Figure 4-51. Control-Panel Circuits Block Diagram

4.10.3 Clock Generator

The clock generator consists of two one-shot circuits that generate clocks (NTCK– and NPC1 +) for the toggle and pulse registers whenever any of the four switches associated with each register are pressed. The one-shot circuits are inhibited when NTCK1– goes low. This occurs during an output-function transfer so that only one clock (from data ready NDRY +) is applied to the toggle register. The one-shot circuits provide time for switch bounce to dampen prior to clocking the registers.

4.10.4 Run-Mode Control

The run-mode control generates a run signal NRUN– that controls the operation of the RUN indicator. When a system reset occurs, a low NRST1– resets the run-mode flip-flop producing a high NRUN– that prevents the RUN indicator from lighting.

When the computer is placed in the run mode with the STEP/RUN switch, a high NSPF– enables the blink-oscillator signal NOSC + to make NRUN– go high and low at a low frequency causing the RUN indicator to blink.

When the START button is pressed (and the computer in step mode), a high NSF + sets the run-mode flip-flop. This provides a low NRUN– that lights the RUN indicator continuously.

When the computer is halted with a halt instruction, a low NFIT– resets the run-mode flip-flop. The high false output of the flip-flop and a high NSPF– enables NOSC + causing the RUN indicator to blink. The run-mode flip-flop is also set when the BOOT switch is pressed (NEB– low) or the option start signal OPSTRT– goes low.

4.10.5 Interrupt Control

When the INT switch is pressed, a low NINT– initiates a control-panel interrupt (figure 4-52). NINT– is debounced by a Schmitt trigger circuit as well as the debounce circuit. High NINT + and NIN1– cause the interrupt-control flip-flop to set (NINF + high) on the next interrupt clock IUCX-I. If the control panel haa priority (PR10X-I low) and the flip-flop is set, the interrupt request IURX-I is generated. When the flip-flop is set, NIN1– goes low to prevent the flip-flop from being set more than once each time the INT switch is pressed. When the interrupt clock occurs, the control panel has priority (PR10X-I low), IUAX-I is true (low), and the flip-flop is reset. The interrupt-control flip-flop is cleared by system reset NRST1– and held in the cleared state by a low NSPF– (step mode).

4.10.6 Blink Oscillator

The blink oscillator consists of two one-shot circuits coupled in a loop to produce a low-frequency squarewave NOSC + for blinking the RUN indicator.

Also included is a Schmitt trigger circuit, used in conjunction with a debounce circuit, to ensure that the system reset SRST– is free from oscillations due to bouncing of the RESET switch contacts.

4.10.7 Input Encoder and I/O Bus Drivers

During an input-function transfer, the input encoder supplies encoded signals to bits 4 through 8 of the I/O bus to determine the operation selected from the control panel. The input function codes are listed in table 4-31. The bootstrap function can generate any one of three codes depending on jumpers A and B:

a. If no jumpers are installed, bits 6, 7, and 8 are encoded for a Teletype bootstrap.

b. If only jumper A is installed, the bits are encoded for a paper-tape reader bootstrap.

c. If jumpers A and B are installed, the bits are encoded for a disc bootstrap.

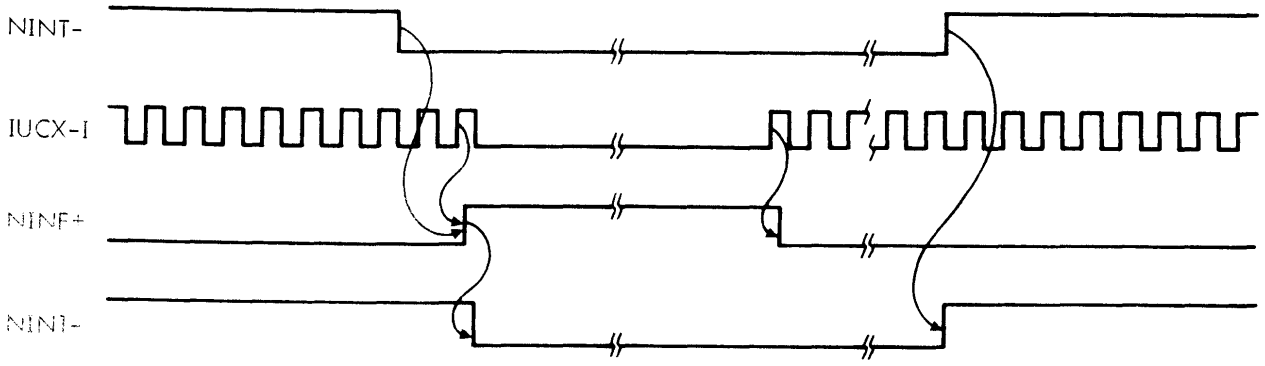


Figure 4-52. Interrupt Control Waveforms

Table 4-31. Input Encoder Truth Table

EB08-I	EB07-I	EB06-I	EB05-I	EB04-I	FUNCTION
H	H	H	H	L	PROGRAM COUNTER SELECTED
H	H	H	L	H	REGISTER FILE OR INSTRUCTION REGISTER SELECTED
H	H	L	X	X	MEMORY SELECTED
H	L	H	X	X	START
H	L	L	X	X	MEMORY SELECTED AND ENTER
L	H	H	X	X	TELETYPE BOOTSTRAP
L	H	L	X	X	MEMORY SELECTED AND DISPL
L	L	H	X	X	PAPER-TAPE READER BOOTSTRAP
L	L	L	X	X	DISC BOOTSTRAP

NOTE: L = LOW, H = HIGH, AND X = IRRELEVANT

VTII-1675A

The I/O drivers enable the display-register outputs to be transferred to the I/O bus. When the I/O bus is idle, a high IEIDLA + enables the data through the I/O drivers onto the I/O bus (EB00-I through EB15-I). The I/O drivers are inhibited with a low display clear NDCLR– or a low system reset NRST1– . Under conditions other than I/O operations, the I/O drivers are enabled for an input-data transfer (ND1 + high).

4.10.8 I/O Control Buffer

The I/O control buffer contains two flip-flops (NCBRST– and NFOS +) and a 4-bit function register. When the device address signal NFAD– (octal 77) is decoded (goes low), the NCBRST– flip-flop is set removing the clear signal from the function register. When the function ready FRYX-I goes true (low), the function register is clocked permitting it to sample inputs (see waveforms in figure 4-53).

The outputs of the function register are used when the computer is halted to provide the normal display. The output signals are:

a. Input function NFI + indicates that the processor determines the operation to be performed.

b. Output function NFO + indicates the processor can set switches after a power failure restart.

c. Input data function NDI + indicates the contents of the display register are transferred to the processor.

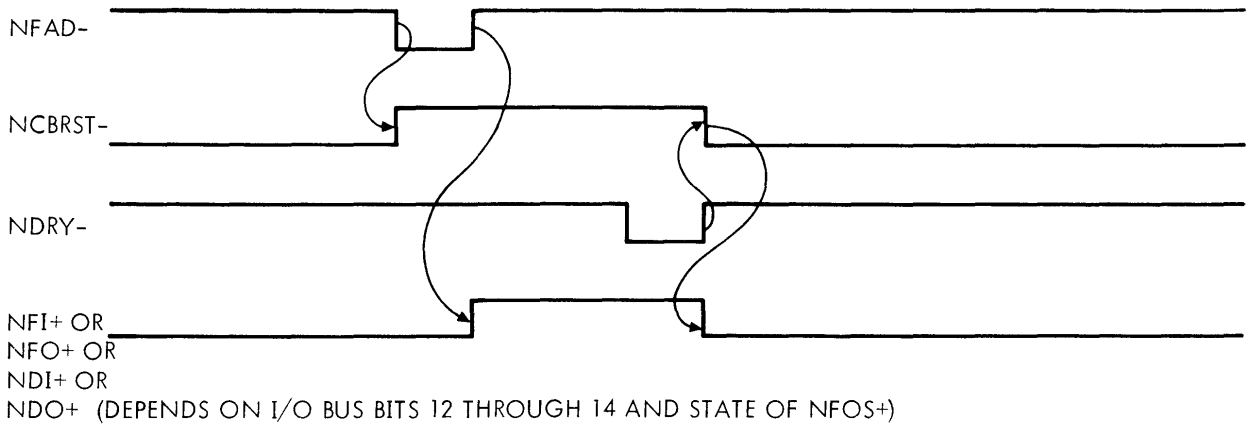
d. Output data function NDO + indicates the processor is sending data to the display register.

Except for the input function, these functions can also be used under program control.

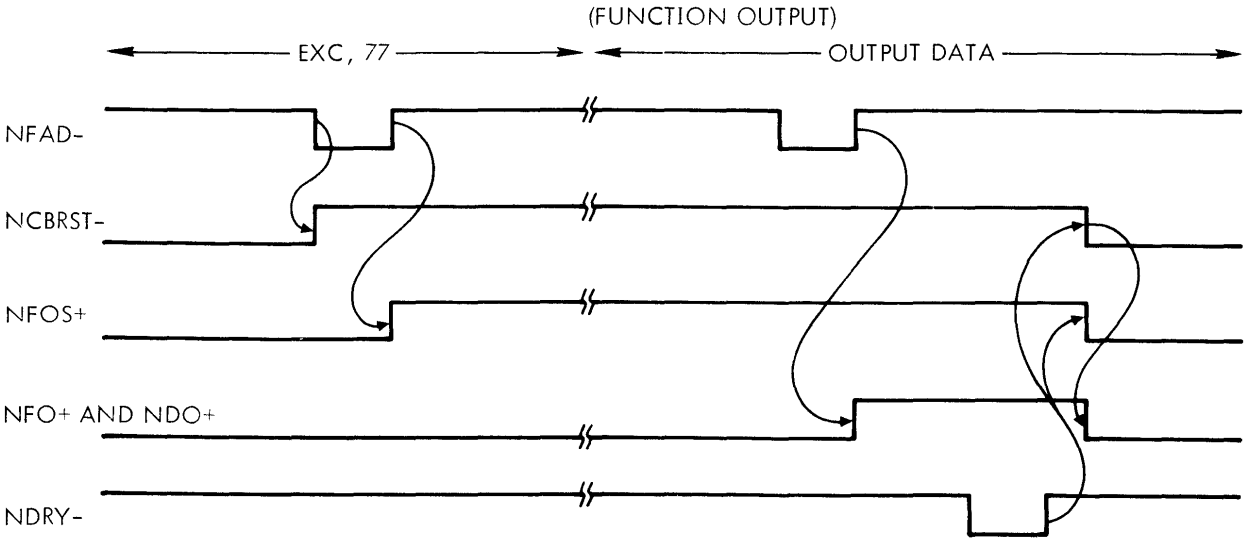
4.10.9 Display Register

I/O data (EB00-I through EB15-I) are applied to the display register inputs. The register outputs (NDF00– through NDF15–) are applied to the register-display indicators. A low NENAB– enables the register. The register is clocked by NCLOCK + .

When the I/O bus is idle (IEIDLA + high) the register clock pulses are generated continuously by a flip-flop that is toggled by the interrupt clock IUCX + . In addition to being applied to the register display indicators, the register outputs are circulated through the I/O data paths to the register inputs.



A



B

VTII-1676

Figure 4-53. I/O Control Buffer Waveforms

4.10.10 Display-Selection Register

The display-selection register contains five latch circuits whose inputs and outputs are connected to the five DISPLAY SELECT switches as follows:

a. NMEM– to MEM switch

b. NST– to STATUS switch

c. NIR– to I switch

d. NPR– to P switch

e. NRF– to REG switch

Pressing any of these switches, sets the corresponding latch (output is low). The previously selected latch is unlatched when NSRX– goes low. This occurs whenever an even number of latches are selected. The low NSRX– unlatches all latches except the one whose switch is pressed. When system reset occurs (NRST1– low), all latches are unlatched.

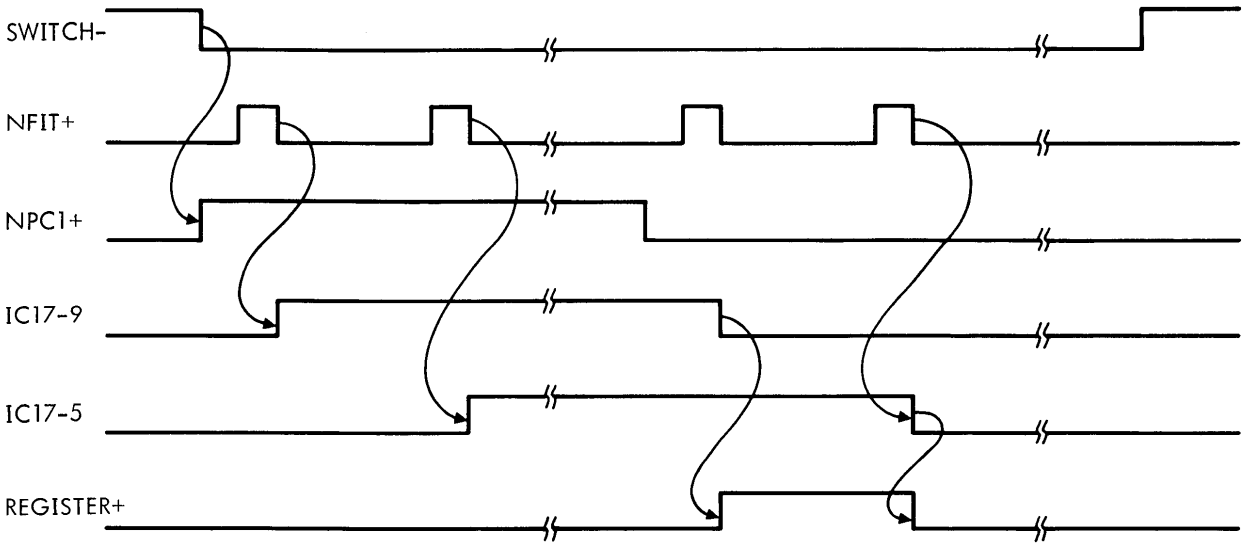
4.10.11 Pulse Register

The pulse register is set whenever the DISPL, ENTER, START, or BOOT switches are pressed. DISPL and ENTER are inhibited from being set when the computer is not in the step mode. BOOT is inhibited from being set when the computer is not in the run mode. A high NDF+ indicates DISPL switch is pressed, a high NEF+ indicates ENTER switch is pressed, a high NSF+ indicates START switch is pressed, and a high NEB+ indicates BOOT switch is pressed. Figure 4-54 shows the timing waveforms for the pulse generator.

4.10.12 Data-Display Drivers and Light-Emitting Diodes

The data-display drivers, which are connected to the outputs of the display register, consist of inverters that drive the light-emitting diodes.

The light-emitting diodes are the indicators for the control panel. A low signal applied to the cathode causes the diode to light; a high signal causes it to go out.



NOTES:
1. THE TERM SWITCH- REPRESENTS SIGNALS NDP-, NEN-, NSTR-, OR NBS-
2. THE TERM REGISTER+ REPRESENTS SIGNALS NDF+, NEF+, NSF+, AND NEB+
3. IC17-9 AND IC17-5 ARE IC PIN NUMBERS ON SHEET 16 OF LOGIC DIAGRAM 91B0406-A

VTII-1677

Figure 4-54. Pulse Register Waveforms

4.10.13 Switches, Debounce, and Pull-Up Circuits

The control-panel switches (except POWER) are mounted on a switch assembly (p/n 44P0647) which in turn is mounted on the control panel board. With the power switch in the CONSOLE DISABLE position: all switches are disabled if jumper terminals C and E are connected together, and only STEP/RUN and RESET switches are disabled if jumper terminals C and D are connected together. A switch is enabled when jumper terminal C is grounded.

Debounce circuits eliminate signal bounce caused by switch contacts.

Pull-up circuits keep the switch-signals at +5V dc when switches are open.

4.10.14 I/O Bus Terminators

The I/O bus terminators provide logic levels of 0 and +3 volts for the I/O bus signals. They also provide impedance matching for the I/O cable.

4.11 MICROPROGRAMMING EXAMPLES

This section provides examples of instruction implementation using Varian microprogramming.

4.11.1 LDA Instruction

In this example, direct addressing is used. The description is divided into the following sections:

a. Standard state 1 of the microprogram (SS1M).

b. Standard state 2 of the microprogram (SS2M).

c. Standard state 3 of the microprogram (SS3M).

d. First microinstruction of the single-word addressing sequence (SWA10).

e. Second microinstruction of the single-word addressing sequence (SWA20).

f. First LDA microinstruction (LDA1).

g. Second LDA microinstruction (LDA2).

SS1M

Initially the instruction pipeline is assumed to be empty so a new instruction must be fetched from main memory. The first microinstruction is obtained from control store location 13E (all addresses are given in hexadecimal). This location has the label SS1M, which is one of the microprogram's standard states.

The microinstruction fields at 13E are:

TS	AF	MS	MT	FS	T	S	G
0000	01001	0010	0	0000	00	01	0000

M	AB	IM	LB	LA	R	F	M
0	00	1000	00	00	000	0000	00

C	WR	SC	V	W	X	SH	B	A
0	0	0	0	0	00	000	0000	0000

The function of this microinstruction is to initiate an instruction fetch from the memory address specified by the program counter. Note that the S field equal to 01 specifies unconditional initiation of the memory cycle. The IM field specifies use of the program counter for an address source and the instruction buffer and memory input register as destinations for data received from memory. The FS, MT, TS and T fields contain all zeros so normal mode addressing is specified. The next control store address will be 092. No other fields of the microinstruction are pertinent.

SS2M

Location 092 is another microprogram standard state labeled SS2M. It continues the process of filling the pipeline by initiating another instruction fetch using the incremented contents of the program counter.

The microinstruction fields at 092 are:

TS	AF	MS	MT	FS	T	S	G
0000	00010	1101	0	0000	00	01	0000

M	AB	IM	LB	LA	R	F	M
0	00	1000	00	00	100	0000	0

C	WR	SC	V	W	X	SH	B	A
00	0	0	0	0	00	000	0000	0000

Again the S field is equal to 01 and the IM field is equal to 1000 specifying another instruction fetch using the program counter. In this case, however, the R field equals 100 specifying that the program counter will be incremented before it is used as an address. This microinstruction is not immediately executed since the previous microinstruction initiated memory activity and the memory control remains busy until the first instruction from memory is loaded into the instruction buffer and the memory input latch (in memory control). At this time, execution of the current microinstruction is completed and the next microinstruction from location 02D becomes active. Normal addressing occurs again due to FS, TS, MT and T fields being zero. No other fields of the microinstruction are pertinent.

SS3M

Location 02D is another microprogram standard state labeled SS3M. It causes decoding of the instruction fetched from memory while checking for interrupts. It also transfers contents of the instruction buffer into the instruction register to make room for the next instruction from memory.

The microinstruction fields at 02D are:

TS	AF	MS	MT	FS	T	S	G
1110	01101	0110	0	0000	00	00	0101

M	AB	IM	LB	LA	R	F	M
0	00	0110	00	00	000	0000	0

C	WR	SC	V	W	X	SH	B	A
00	0	0	0	0	00	000	0000	0000

This microinstruction manipulates no data paths nor does it initiate any memory cycles. Its sole purpose is to check for interrupts and, if there are none, cause a branch to the required microsequence. The T and S fields are both equal to 00 and the G field bit 0 is a one causing data transfer from the instruction buffer to the instruction register. The G field bit 2 is a one, thus enabling interrupts and decoder addressing. The TS field defines the interrupts which are enabled: all except I/O interrupts unless the memory protection option is installed. The IM field specifies selection of the interrupt flag. If this flag is set, interrupts are suppressed. The flag is reset by this microinstruction. If an interrupt is active and the interrupt flag is not set, the next control store address is 0DX where X designates the four bits supplied by the interrupt logic. This produces a branch to the interrupt microprogram sequence.

Assuming no interrupts are present, the new control store address is determined by the decoder logic. The instruction fetched from memory is assumed to be 10F9 (hexadecimal) or 010371 (octal). This is a Varian 70 series LDA instruction with direct addressing of location 00F9 (hexadecimal). The most significant four bits of the instruction buffer address the first decoder control store at location 1. The next four bits address the second decoder control store at location 00. The decoder control store contents are:

1st decoder

Control store location 1	B12 = 1 B8-B0 = 110000010

2nd decoder

Control store location 0	A8-A0 = 010000000

Since B12 equals 1, the B8-B0 and A8-A0 address components are logically ORed to produce an address of 182.

SWA10

Location 182 contains the first microinstruction of the single-word addressing sequence (SWA10) for the instruction fetched from memory. It forms the effective address by masking bits 00 through 10 from the instruction register. It also initiates the operand fetch.

The microinstruction fields at 182 are:

TS	AF	MS	MT	FS	T	S	G
0000	10010	1111	0	0000	00	01	0000

M	AB	IM	LB	LA	R	F
0	00	0101	10	00	011	1010

M	C	WR	SC	V	W	X	SH	B	A
1	11	1	1	0	0	00	000	0000	0000

16-bit mask literal

The LB field equals 10 so the ALU B input has the contents of the instruction register masked by the 16 bits of the M, C, WR, SC, V, W, X, SH and BB fields (a zero in the mask enables the corresponding instruction register bit). The mask equals F800 so the low order 11 bits of the instruction are used.

The ALU mode is determined by the F field (1010) in conjunction with the LB field (forces logical mode) resulting in an ALU function of the ALU = B.

The R field equals 011 so the ALU output is loaded into the operand register.

The S field equals 01 so unconditional memory control is specified by the IM field (0101) to fetch an operand into the memory input latch using the ALU output for an address source. This microinstruction is completed when the memory cycle initiated by the microinstruction at 092 is completed.

The FS, TS, T, and MT fields all contain zeros so normal addressing is used and the AF and MS fields specify the next control store address of 12F.

SWA20

Location 12F contains the second microinstruction of the single-word addressing sequence (SWA20). It decodes bits

13 through 15 of the instruction register contents to determine the class of the single-word addressing instruction.

The microinstruction fields at 12F are:

TS	AF	MS	MT	FS	T	S	G
0000	11110	1100	1	1111	00	00	0000

M	AB	IM	LB	LA	R	F	M
0	00	0000	00	00	000	0000	0

C	WR	SC	V	W	X	SH	B	A
00	0	0	0	0	00	000	0000	0000

No data manipulation or memory control operations are performed by this microinstruction. It serves only to branch to the specific microsequence for the class of single-word addressing instruction contained in the instruction register. Field-selection addressing is used for this decoding (FS field is not equal to 0000). The FS field is equal to 1111 so the selected field is bits 11 through 15 of the instruction register. The composite address formation is illustrated:

		8	7	6	5	4	3	2	1	0
AF field contribution:		1	1	1	1	0	0	0	0	0
	or:	1	1	1	1	0	0	0	0	0
TS field contribution:		0	0	0	0	0	0	0	0	0
Field selected from instruction register: (I = 10F9)		0	0	0	0	0	0	0	1	0
	and:	0	0	0	0	0	0	0	0	0
Mask consisting of MT and MS fields		0	0	0	0	1	1	1	0	0

Final effective address produced by inclusive OR:

1 1 1 1 0 0 0 0 0

The address of the next microinstruction is then 1E0.

LDA1

Location 1EO is the first microinstruction specifically for the LDA instruction (LDA1).

This microinstruction increments the program counter and initiates another instruction fetch from main memory.

TS	AF	MS	MT	FS	T	S	G
0000	01011	0101	0	0000	00	01	0000

M	AB	IM	LB	LA	R	F	M
0	00	1000	00	00	100	0000	0

C	WR	SC	V	W	X	SH	B	A
00	0	0	0	0	00	000	0000	0000

The RF field equal to 100 specifies that the program counter is incremented during this microinstruction.

The S field equals 01 so unconditional memory control is specified by the IM field (1000) to fetch an instruction into the instruction buffer and input latch using the program counter for an address source. (Note that the program counter is incremented during the microinstruction so the new value is used for the memory cycle).

Normal addressing is used to specify the next microinstruction address (T, TS, FS, MT fields are all zero). The AF and MS fields define the address as 0B5.

LDA2

Location 0B5 is the second microinstruction specifically for the LDA instruction (LDA2). This microinstruction performs the following: transfers the contents of the memory input latch to the A register (R0), transfers the instruction buffer containing the next instruction to the instruction register to make room for the instruction whose fetch was initiated by the microinstruction 1E0, decodes the instruction buffer to determine the starting address of the next microsequence, and checks for interrupts.

The microinstruction fields at 0B5 are:

TS	AF	MS	MT	FS	T	S	G
1111	01101	0110	0	0000	00	00	0101

M	AB	IM	LB	LA	R	F	M
0	00	0110	01	00	000	1010	1

C	WR	SC	V	W	X	SH	B	A
00	1	0	0	0	00	000	0001	0000

The ALU B input is specified by the LB field (equal to 01) to be one of the special registers. The B field (equal to 0001) defines the memory input register as the source.

The ALU operation is specified to be in the logical mode (MF = 1) with the ALU output equal to the ALU B input (FF = 1010).

The WR bit equals a one so the ALU output data is written into the register specified by the A field (A = 0000) which is the A register. This is the execution phase of the LDA instruction.

The S and T fields are both equal to 00 and the G field bit 0 is a one so the instruction buffer contents are transferred to the instruction register. The G field bit 2 is a one so the instruction decoder is enabled and interrupts are checked.

The IM field equal to 0110 with the SF field equal to 00 selects and resets the interrupt flag. If the flag is set, the decoded address and interrupts are suppressed and the next instruction is fetched from location 0D0. All interrupt classes are enabled as the TS field contains all ones. If

an interrupt is active and the interrupt flag is off, only the decoded address is suppressed and the next microinstruction is fetched from the address specified by the A field and the interrupt logic. This address is 0DX where X is the address supplied by the interrupt logic (X = 0).

If no active enabled interrupts exist, the next microinstruction is fetched from the address specified by the decoder logic. If the instruction buffer contains another single-word addressing instruction, the next address will be 182 (SWA10) and the sequence will be repeated.

Figures 4-55 and 4-56 show a flowchart and the flow diagram of the microinstruction sequence described. Note that the pipeline effect of buffering instructions permits efficient use of the memory. (A 330-nanosecond semiconductor memory was assumed).

4.11.2 STA Instruction

This example illustrates the use of the equal-address flag MPLE+ (see address comparator in section 4.5.4) during execution of the STA instruction with direct addressing into location n + 1 (where n is location of STA). The operation is divided into the four sequences as shown in figure 4-57.

SWA40

Microinstruction SWA40 (in location 181) initiates a memory request to store contents of the A register in memory location n + 1.

STA2

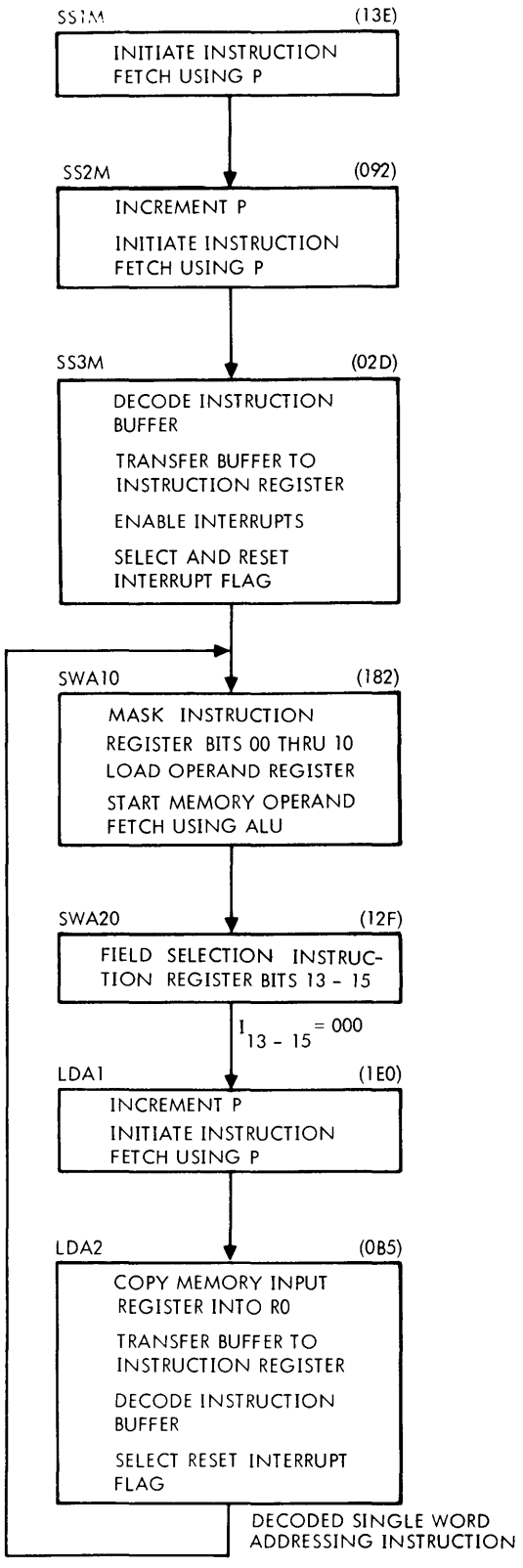
During the memory store operation, microinstruction STA2, (in location 11A) tests for equal addresses from the program counter and memory address lines. At the completion of STA2, MPLE+ is set indicating the addresses are equal.

STAX

Microinstruction STAX (in location 1DA) specifies a second memory request operation to maintain the instruction pipeline. The central control continues executing STAX until the memory store operation initiated by SWA40 is completed. At the completion of the memory store operation, contents of the A register are stored in both the instruction buffer and memory location n + 1 (due to MPLE+).

SS3M

The central control now initiates the instruction fetch operation and enters standard state 3 (SS3M) to decode the next instruction (the one just loaded into the instruction buffer).



VTII-1938A

Figure 4-55. Flowchart for LDA Instruction

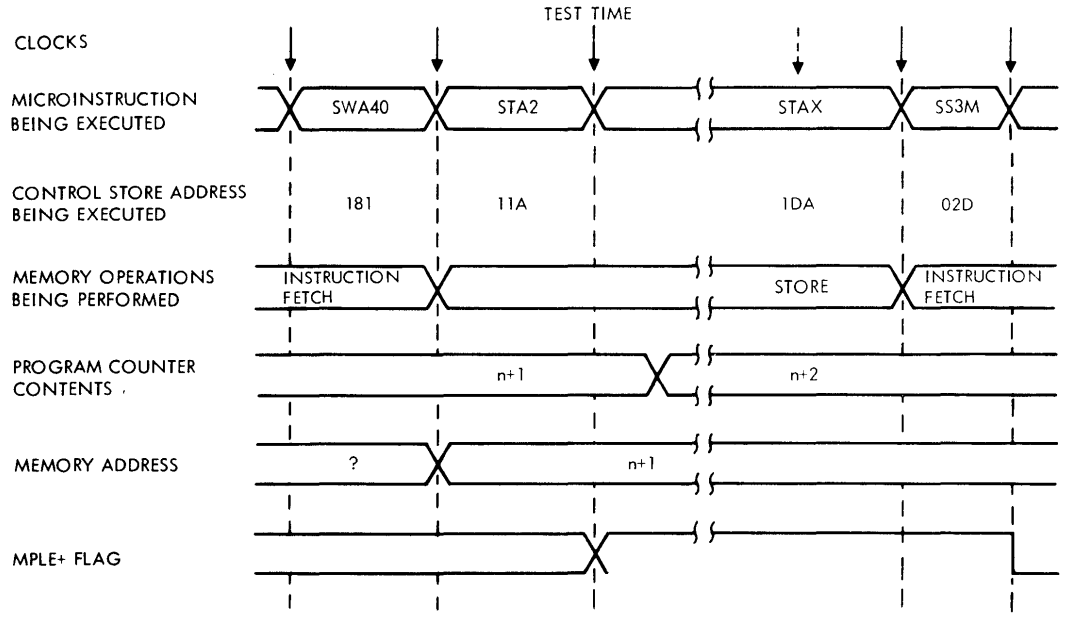
	IDENT (HEX. ADDR.)	SS1M (13E)	SS2M (92)	SS3M (2D)	SWA10 (182)	SWA20 (12F)	LDA1 (1E0)	LDA2 (0B5)
MEMORY	FUNCTION		FETCH LDA	FETCH NEXT INST.	FETCH NEXT INST.	FETCH OPERAND	FETCH OPERAND	FETCH THIRD INST.
	REQUEST	IF	IF		OF		IF	
	ADDRESS	P	P		ALU		P	
ALU	INPUT A							
	INPUT B				I ∧ 07FF			MIR
	OUTPUT				TRNB			TRNB
	DESTINATION							R0
STATUS	SAMPLE							
	TEST							
ADDRESSING	MODE			DECODE	FIELD SELECTION 113-115			DECODE
	ADDRESS	SS2M	SS3M	FROM DECODER	SWA20	LDA1+X WHERE X = 0,4,8,...28	LDA2	FROM DECODER
OTHER	SPECIAL ACTIONS		INCP	ENABLE INTERRUPTS IBR→I			INCP	IBR→I ENABLE INTERRUPTS

NOTE:

Timing diagram shows the start-up and execution of a sequence of single-word addressing instructions (330 nanosecond memory cycle time is assumed).

VTII-2084

Figure 4-56. Flow Diagram of LDA Instruction



VTII-2168

Figure 4-57. STA Instruction Timing Diagram

SECTION 5
MAINTENANCE

Maintenance personnel should be familiar with the contents of this manual before attempting processor troubleshooting. The Varian MAINTAIN II test program system (Test Programs Manual, 98 A 9952 06x) contains an instructions test program used to isolate a malfunction to a particular group of instructions. Further diagnosis can then be made by referring to this manual.

5.1 TEST EQUIPMENT

The following test equipment and tools are recommended for processor maintenance:

a. Oscilloscope, Tektronix type 547 with dual-trace plug-in unit, or equivalent.

b. Multimeter, Triplett type 630 or equivalent.

c. Soldering iron, 15-watt pencil type.

5.2 CIRCUIT BOARDS

The processor and option boards are four-layer PC boards. The two outer layers provide signal interconnections for the circuit components. The two inner layers provide low-impedance ground and power-voltage distribution, and 90-ohm microstrip transmission lines for all signals. The ICs contained on the boards consist of LSI read only memories; MSI multiplexors, decoders, and registers; and SSI gates and flip-flops. The control-panel circuit board contains only two outer layers for signal interconnections of the circuit components.

If it has been determined that circuit-board repair is required, it is recommended that the Varian Data Machines customer service department be contacted so that a new circuit board can be installed in the user's system and the faulty one returned to the factory for repairs. However, if the user decides to perform his own repairs, extreme caution should be used so that the circuit board is not permanently damaged. Approved repair procedures should be followed such as the ones described in document IPC-R-700A prepared by the Institute of Printed Circuits.

5.3 CIRCUIT-COMPONENT IDENTIFICATION

For IC components, the processor and option boards have location coordinates that are used in the logic diagrams as reference designations. For example, a flip-flop designated C8 in the processor logic diagram is in the IC package at location row-C column-8 on the processor board. For discrete components, the reference designations used in the logic diagrams appear on the circuit board adjacent to each component.

For the control-panel board, all reference designations used in the logic diagrams appear on the circuit board adjacent to each component.

Parts lists in the System Maintenance Manual provide a cross reference between Varian and the manufacturers part numbers.

SECTION 6
MNEMONICS

This section presents an alphabetized list of processor signal mnemonics with definitions (table 6-1), and describes certain conventions used in developing the mnemonics.

Polarity

Plus or minus signs are included at the end of each processor mnemonic. The plus sign indicates the signal is at a high logical level when its function is being performed. The minus sign indicates the signal is at a low logical level when its function is being performed. A signal that is the logical inversion of another uses the same mnemonic with an opposite sign; these signals are complements of each other.

I/O bus signal mnemonics end with – I.

Source Identification

The first letter of a signal mnemonic indicates the functional section in which it originates:

First Letter	Source
D	Data loop
C	Central control
M	Memory control
I	I/O control
N	Control panel
W	Power supply
L	Common logic
T	TTY
A	PMA
B	Memory protection
R	Real-time clock
F	Power failure/restart
Y	Memory

Table 6-1. Mnemonics

Mnemonic	Description
CABRT–	Implements the disabling of a memory request
CACIDA +	Transfers contents of instruction decoder onto control-store address bus.
CACIDB +	Transfers contents of instruction decoder onto control-store address bus.
CACIDE +	Enables instruction decoding operations.
CADEQ1	System reset or control-panel interrupt (STEP).
CBABE +	Transfers contents of register-field selector onto control-store output bus (bits 0-7).
CBABn +	Two bits, AB field
CBAFn +	Five bits, AF field
CBAKE–	Transfers bits from control-store output bus to A field
CBFSn +	Four bits, FS field
CBGn +	Four bits, G field
CBIEN–	Disables control-store output bits 0-7
CBIMn +	Three bits, IM field
CBIn +	64 bits, control-store outputs
CBMR +	One bit, MR field
CBMSn +	Five bits, MS field
CBMT +	MT-field bit
CBSn +	Two bits, S field
CBTn +	Two bits, T field
CBTSn +	Four bits, TS field
CCPRQ +	Detects control-panel interrupt forcing control-store address to low state
CDREN +	Partial enable for instruction decoder
CEADn–	Nine bits, control-store address
CEAFS +	Transfers contents of instruction field selector to control-store address
CEAIA +	Enables interrupt request
CEATS +	Selects TS field for control-store address
CEA1–	Allows instruction field selection when T is not zero

Table 6-1. Mnemonics *(continued)*

Mnemonics	Description
CECKM–	Clock control signal
CESK +	Supervisor control signal
CFSAn +	Five bits, instruction-field selector A output
CFSBn +	Five bits, instruction-field selector B output
CIDA2n +	Nine bits, instruction decoder A2 output
CIDA3n +	Nine bits, instruction decoder A3 output
CIDIO +	Output signal from instruction decoder, high for I/O operation
CIDJMK +	Processor decoded jump and mark
CIDR +	Special decoding of register-to-register instructions
CIDS2n +	Two bits, instruction decoder output
CIDS3n +	Three bits, instruction decoder output
CIDT32–	Instruction decoder output
CIDXXn +	Four bits 0, 2, 3, 5, instruction decoder output
CINTF +	Processor interrupt flip-flop
CINTS–	When low, disables the instruction decoding logic
COVRM +	Implements override of memory request
CRQIO	Processor I/O request
CRQM +	Requests memory service (in conjunction with CABRT–)
CSAMCC	Enables conditional code control
CSAMOV	Enables overflow control
CSEQn +	Three bits, decoded S fie'd
CTEQn +	Four bits, decoded T field
C2In–	16 bits, instruction register output
DALn +	16 bits, ALU outputs
DASBN +	When high, the last entry to the register file A address 0000 was negative
DAZR–	When low, the last entry to the register file A address 0000 was zero
DBAD +	Byte address
DBCn–	16 bits, latch B feedback path
DBSn +	Four bits, gated address for register file B
DBZB–	When low, the last entry to the register file B address 0001 was zero
DCIN–	Decoded carry-in to bit 0 of ALU
DCKn +	Four bits, output of data loop key register
DCNDC +	At the last condition sampled, the carry-out of the ALU was true
DCNOZ +	At the last condition sampled, the output of the ALU was zero
DCRY0 +	Carry control LSB, bit 13 of mask field
DCRY1 +	Carry control MSB, bit 14 of mask field
DCTZ +	Shift count equals zero
DC0n +	Carry-in signals from carry generator to ALU
DDQS–	Quotient bit input to DFBA0 +
DEQ +	At the last condition sampled, the A and B inputs to the ALU were equal
DFAAn +	Four bits, address for register file A
DFAn–	16 bits, output of register file A
DFBAn +	Four bits, address for register file B
DFBn–	16 bits, output of register file B
DFLCn +	Latch strobe, enables latch feedback path, disables all gating paths
DFTT +	When high, the condition is met for a jump, jump and mark, or execution test
DFUN0 +	ALU function control LSB
DFUN1 +	ALU function control bit 1
DFUN2 +	ALU function control bit 2
DFUN3 +	ALU function control MSB

Table 6-1. Mnemonics *(continued)*

Mnemonic	Description
DGLA0 +	Latch A control LSB
DGLA1 +	Latch A control MSB
DGLB0 +	Latch B control LSB
DGLB1 +	Latch B control MSB
DGMD +	ALU mode control bit - bit 15 of mask
DGPR +	Gates the P register through latch A
DGn +	Decoding signals of lower two bits of SH field
DGRC0 +	Register control LSB (R field)
DGRC1 +	Register control middle bit (R field)
DGRC2 +	Register control MSB (R field)
DIMEN +	Enables multiplication sign bit to bit 15 of latch A
DKABB	Clock to control-store buffer bits 8 through 31 and 0 through 3
DKOR–	Operand register clock
DKPR–	Program counter clock
DKSCR–	Shift counter clock
DLAGL +	Gates file A data shifted left through latch A
DLAGR +	Gates file A data shifted right through latch A
DLAGS +	Gates file A data straight through latch A
DLBGI +	Gates I register through latch B
DLBGM +	Gates mask field through latch B
DLBGS +	Gates file B data straight through latch B
DLBMX +	Gates B multiplexor through latch B
DLBn +	16 bits, latch B outputs
DLCn +	16 bits, latch A feedback path
DLTZ +	The current contents of the I register bits 3-8 and bit 0 are zero
DMFC +	Data loop full clock distribution
DMODE +	Decoded mode control to ALU
DMXn–	16 bits, output of B multiplexor
DORn +	16 bits, output of operand register
DOSn +	2 bits, decoded control signals to operand register
DOVF +	When high, the overfow flag is set
DOVSMP +	There is overflow, if the sample condition is on (set DOVF +)
DPRn +	16 bits, output of program counter
DRCS +	Increments shift counter
DRC1 +	Loads program counter
DRC2–	Loads shift counter
DRC3–	Loads operand register
DRC4 +	Increments program counter
DRC6–	Loads data-loop key register
DRC7–	Loads operand register and increments program counter
DR47 +	DRC4 or DRC7
DSCON +	Enables shift or operand register - (bit 11 of mask field)
DSCn +	Five bits, shift counter outputs
DSGN +	At the last condition sampled, the output of the ALU was negative
DSGn +	Carry-generating signals from ALU to carry generator
DSHFTn +	Three bits, controls latch A bits 15 and 0 for shift operations
DSL00 +	Multiplexed shift left input to operand register
DSM1–	Transfers multiplication sign bit to bit 15 of latch A
DSPn +	Carry propagating signals from ALU to carry generator
DSROV +	Enables set or reset of overflow flag according to contents of control-store buffer
DSR15 +	Multiplexed shift right input to operand register bit 15

Table 6-1. Mnemonics *(continued)*

Mnemonics	Description
DTCND +	When high, the current test being gated through the test multiplexor is true
DTKL–	When low, enables sample test flags
DV +	V field bit
DW +	W field bit
DWCF +	Enables data to be written into register files (bit 12 of mask field and WR field bit)
DWCN–	When low, enables data to be written into register files
DXn +	Two bits, control operand register (X field bits)
DXZR–	When low, the last entry to the X register (file address 0010) was zero
DZTS–	When low the ALU output is zero
D3RC7–	DRC3 or DRC7, either loads operand register
IAKC1	I/O acknowledgment of processor request (CRQIO +)
IMAn +	Eight bits, I/O multiplexor outputs
IRAn +	Eight bits, outputs of address counter (in I/O control)
IROn +	16 bits, outputs of I/O control-store
IBAD	Byte address
ICA +	Interrupt clock generator, flip-flop A
ICB +	Interrupt clock generator, flip-flop B
ICC +	Interrupt clock generator, flip-flop C
ICD +	Interrupt clock generator, flip-flop D
ICD0B +	Control signal for I/O multiplexor (in I/O data loop)
ICD1 +	Control signal for I/O multiplexor (in I/O data loop)
ICD2B +	Control signal for I/O register (in I/O data loop)
ICD3B +	Control signal for I/O register (in I/O data loop)
ICD4A +	Enables I/O drivers (in I/O data loop)
ICRY +	Reserved for future use
IDCIR +	When high, the processor is able to service interrupt
IDN +	I/O done flag sent to processor
IDNC–	I/O done signal sent to processor
IDRY +	DRYX-I, IUJX-I and DRYF-I timing signal
IEFn +	Three bits, least signification bits of I/O control-store word (buffer output)
IEIDLE–	Idle enable flag
IEJMR +	Enable IUJX-I
IESMP +	Sample period of normal interrupt clock
IEWAIT +	Wait enable flag
IEXW +	Exit external wait
IFC +	Full clock, 165 nanosecond period
IFRY +	FRYX-I, FRYF-I, and DRYF-I timing signal
IHC +	Half clock, 165 nanosecond period, 180 degree phase shifted from IFC
IHSD +	High-speed DMA flip-flop
IIIDLE–	Inhibit idle
IIIT–	Inhibit timing
IINHC–	Clock inhibitor
IINHF +	Interrupt clock inhibitor, fast rate
IINHN +	Interrupt clock inhibitor, normal rate
IINTF +	I/O interrupt flag
IJMRK +	Decoded interrupt for a jump and mark instruction
ILDAD–	Loads address into address counter
IMAn +	Eight bits, I/O multiplexor outputs
IMXAD +	I/O control multiplexor selector
INSD +	DMA flip-flop
IODC +	Reserved for future use
IOWB	Word/byte
IRAn +	Eight bits, outputs of address counter (in I/O control)

Table 6-1. Mnemonics *(continued)*

Mnemonics	Description
IROn +	16 bits, outputs of I/O control store
IRQC +	Interrupt request from I/O control to processor
IRQM +	I/O requested by the memory
IRST–	I/O reset
ISMF +	Reserved for future use
ISMI	Program interrupt pending
ISRD +	Sense flip-flop
ITRPF +	Fast trap request
ITRPN +	Normal trap request
IUAF +	Interrupt acknowledgment fast
IUAX +	Interrupt acknowledgment normal
IURX +	Program interrupt
IWLMC–	I/O write, left byte
IWRMC–	I/O write, right byte
IXRSA3 +	Bit 3 of external-request sequence address
IXRSA4 +	Bit 4 of external-request sequence address
KKC2 +	165 nanosecond clock driver
MABE +	Transfers address to memory address bus
MAKIO +	Memory control acknowledges I/O
MAKO +	When high, the PMA request is accepted
MAMn +	16 bits, memory address multiplexor output
MASLn +	Two bits, selection signals for memory address multiplexor
MCDFC–	165 nanosecond full clock, gated
MCDHC–	165 nanosecond half clock, gated
MCPA +	Phase A of clock generator
MCPB +	Phase B of clock generator
MCRA +	When high, the processor request is accepted on positive edge of full clock
MCRB +	When high, the processor request is accepted on negative edge of full clock
MCRP +	When high, the priority logic enables the processor's memory request
MDBE +	When high, enables memory data bus drivers
MDMn +	16 bits, memory data multiplexor output
MDNC–	Memory done
MFC–	165 nanosecond full clock
MHC–	165 nanosecond half clock
MILn +	16 bits, output of memory input latch
MIMCn +	Two bits, stores bits 0 and 1 of IM field to specifies read/write operation
MIMNZ–	Goes low when IM field bits 2 and 3 are not both zero
MIOLn +	16 bits, output of memory I/O latch
MIOW +	When high, a memory write operation for I/O data
MIRA–	When low, an I/O memory request is accepted on the positive edge of the full clock
MIRB +	When high, an I/O memory request is accepted on negative edge of full clock
MIRP +	When high, an I/O memory request is enabled
MMIL–	Loads memory data into memory input latch
MMIOL–	Loads memory data into memory I/O latch
MMII–	Loads memory data into instruction buffer
MPLE +	Address comparator output
MQC–	Oscillator clock
MRQ–	Memory request from control sequencer
MRQSC +	Memory request sampling clock
MRS1 +	Memory sequencing flip-flop 1
MRS2 +	Memory sequencing flip-flop 2

Table 6-1. Mnemonics *(continued)*

Mnemonics	Description
MRS3 +	Memory sequencing flip-flop 3
MWLY +	Left-byte control from memory writing drivers
MWRY +	Right-byte control from memory writing drivers
MYAn +	15 bits, memory address bits
M1In +	16 bits, instruction register outputs
SERX +	Sense response
SMRQ +	Reserved for future use
XCSEN +	When high, enables the control-store (read only)

ADDENDUM 1

Varian 70 Series Processor Manual

98 A 9906 023

This addendum contains changes to the Varian 70 Series Processor Manual.

PAGE	ACTION
4-68	Replace the sentence in the first block under the TS field with the following: TS is used to generate control store address bits 1 through 4 during conditions not met, field selection addressing, and normal/TS field addressing.
4-68	Replace the heading in the last block under the TS field with the following: Request I/O (IM = 1110 V 1111 ∧ S = 0)
4-68	Replace the last entry in the first block under the G field with the following: xxx1 Transfer contents of instruction buffer to instruction register (S = 0 does not apply).
4-69	Add the following to the first entry (AB = 01) in the first block under the AB field: Inhibits change to A field.
4-69	Add the following to the second entry (AB = 10) in the first block under the AB field: Inhibits change to B field.

ISSUED: OCTOBER 1974

PAGE	ACTION
4-70	The third entry in the first block under the X field should contain the X contents of 10 instead of 01.
4-70	Replace the heading in the first block under the SH field with the following: (LA = 00 OR 01) $\wedge$ (LB = 00 OR 01)
4-86	Replace the third and fourth sentences of the paragraph immediately following the SS3M fields with the following: The T field is equal to 0 and the G field bit 0 is a one causing data transfer from the instruction buffer to the instruction register. With the S field equal to 00 and the G field bit 2 equal to one, interrupts and decoder addressing are enabled.

varian data machines /a varian subsidiary