VARIAN 73

PRELIMINARY

MICROPROGRAMMING GUIDE

August 1972

The information contained herein is of a preliminary nature for information and is subject to change without notice. Varian Data Machines assumes no liability for any consequences resulting from its use.

TABLE OF CONTENTS

TABLE OF CONTENTS (cont'd.)

TABLE OF CONTENTS (cont'd.)

FIGURES AND TABLES

V73 MICROPROGRAMMING GUIDE

1. MICROPROGRAMMING ELEMENTS

In the conventional processor, the control section normally consists of large assemblies of gates and flip-flops interconnected to form timing counters, sequencers and decoders to perform the following functions required by the specific instruction set:

- fetch instructions from memory
- decode machine instructions
- enable appropriate data paths
- change the state of the computer to that required by the next operation

In a microprogrammed processor, the control section is implemented in a less random fashion. All control signals are derived from information stored in a memory device (usually a read only memory). This memory, together with its buffers and control logic, form the control sections. The control words stored in the memory are known as microinstructions. Preparation of these instructions is known as microprogramming. These microinstructions bear no resemblance to the computer's own instruction set as they manipulate and control data at the most elementary level.

Advantages of microprogramming are:

- Provides an orderly method of implementing modifications and extensions to existing instruction sets.
- Permits easier troubleshooting through minimization of random logic.
- Permits optimum tailoring of computer systems to a specific task by implementing frequently used operations in microinstructions.

Design Considerations

Two fundamental choices must be made in the design of a microprogrammed control section:

- number of bits in control word
- number of computer states for each microinstruction

These choices must be made by trade-off decisions involving objectives of cost, performance and capability.

The Varian 73 CPU is designed to offer high speed performance (approximately twice that of the Varian 620/f-100), generalized organization to permit great application flexibility and modular expansion to allow orderly system growth.

High performance is achieved through the use of a microinstruction consisting of 64 bits. This permits a great degree of parallelism in the computer. Data path manipulations, instruction decoding, conditional testing, interrupt monitoring and memory cycle initiation may be simultaneously controlled.

Processor

High speed logic and use of a 60 nanosecond access time control store permit the Varian 73 CPU to execute a single microinstruction in 165 nanoseconds (since each microinstruction represents just one state of the processor, the most general capability is achieved.

A processor interface is provided to permit the internal read only memory to be disabled by an external writable control store to permit the Varian 73 to be adapted to special applications through user microprograms. This can result in significant performance improvement.

Varian 73 Microprogramming

Major operations performed by microinstructions are:

- Data path manipulation
- Address sequence
- I/O and memory control
- Specification of processor status
- Instruction register field selection
- Miscellaneous control functions

Details of these major operations are described in the following paragraphs (refer to figure

Data Path Manipulation

The types of data path manipulations are:

a. Source selection for each of the two inputs of the arithmetic and logic unit (ALU), including introduction of 16 bit data from the contents of the control store and masked instruction register. The sources are:

1. Any two of 16 general purpose registers. One of the two may be selected for shift 1 bit left or right or unshifted.
2. Operand register.
3. Memory input latch.
4. I/O register.
5. Operand register right byte with sign extended.
6. Operand register left byte with sign extended.
7. Operand register right byte with zeros in left byte.
8. Operand register right byte in left byte position. Zeros in right byte position.

9. Program counter.
10. 16 bit control store literal.
11. Instruction register masked by control store literal.
12. Processor status word.
13. All zeros register.
14. All ones register.

b. Selection of the ALU function (arithmetic or logical operations).

c. Destination selection for ALU output data. The destinations are any one of 16 general purpose registers and one of the following:

1. Operand register.
2. Program counter.
3. Shift counter.
4. Processor key register.
5. I/O key register.
6. Memory address register.*
7. I/O register.*
8. Memory data bus.*
9. Instruction buffer via memory bus.*

*These destinations are selected by specifying I/O or memory operations.

d. Incrementing of program or shift counters.

e. Selection of carry input to the ALU. The selections are:

1. No carry
2. Carry
3. Stored carry
4. Stored carry complement

f. Selection of shifting operations. The selections are:

1. Shift or rotate
2. Single or double word
3. Arithmetic or logical

4. Left or right
5. Setting and selecting flag bits used in multiply or divide algorithms.

Address Sequence

The next microinstruction address can be determined through one of the following operations:

a. Unconditional specification of a single address (or set of addresses dependent on the instruction register contents).
b. Specification of two addresses (or sets of addresses); the choice is dependent on the status of a selected flag.
c. Decoding of the instruction buffer contents.
d. Specification of a branch outside the normal 512 word control store.
e. Conditional override of all of the above operations in the presence of an interrupt.
f. Specification of a wait for the completion of I/O or memory operation.

I/O and Memory Control

A memory operation can be initiated and the following parameters specified:

a. Specification of address source.
b. Specification of data destination for read operations.
c. Specification of a read or write operations.
d. Specification of word or byte write operation.
e. Initiating of the memory cycle may be conditional on the pass or failure of a test of processor status flags.
f. A previously initiated operationscan be overriden to change the type of operation and the destination of data.

An I/O operation is initiated by specifying a starting address for the independent I/O control store.

Specification of Processor Status

Examples of processor status are:

a. Set, reset or sample the overflow bit.

b. Sample the ALU carry, zero detection and sign.

Instruction Register Field Selection

By selecting a 3 or 4 bit field from the instruction register as the A or B field of the next microinstruction, register specification can be done directly from the instruction register without requiring control store addresses for the decoding. The A and/or B fields, once set up, can be maintained for as many microinstructions as desired.

Miscellaneous Control Functions

Microinstructions can perform miscellaneous control functions such as:

a. Selection of interrupt classes to be enabled.

b. Transfer of instruction buffer contents to instruction register.

c. Set and reset supervisor mode.

d. Set and reset interrupt flag.

II. WRITABLE CONTROL STORE

Writable control store (WCS) provides a means for extending the V73 processor's read only memory control store to permit addition of new instructions, emulation of other instruction sets, development of micro diagnostics and optimum tailoring of the computer system to the application.

Unlike the read only memory which contains the V73 basic instruction set and cannot be altered, the writable control store consists of arrays of read/write memory which can be loaded from the computer system's main memory under control of I/O instructions. The capability of altering the contents of the writable control store permits full user access to the resources of the V73 computer.

The WCS is packaged on a standard 15.6 in. (39.62 cm) x 19.0 in. (48.26 cm) V73 style multi-layer printed circuit card. It plugs into the V73 backplane to interface with the memory buses and connects to the processor and option boards via flat ribbon cable. (See figure 10-6 of Varian 73 Systems Handbook). Power is supplied entirely from an external power supply via a power cable attached from the chassis rear. To minimize signal propagation delays the WCS module(s) should be located within 4 card slot positions of the processor and option boards. Up to three writable control store modules may be added to a V73 system.

The WCS is available with either 256 or 512 words of 64 bit central control store. The central control store controls manipulation of all of the V73 processor's data paths (except I/O) and is capable of initiating memory or I/O activity.

Optionally available are instruction decode and I/O control store arrays to permit efficient decoding of alternate instruction sets and variations in I/O bus discipline. The decode control store option consists of two 16 word by 16 bit memory arrays with the logic necessary to map instruction buffer register contents into central control store addresses. The I/O control store option consists of a 256 word by 16 bit memory array

which controls operation of the I/O data paths and I/O bus signals.

When operating under WCS control, the processor executes a micro instruction in 190 ns. instead of the normal 165 ns.

Another optional feature is the micro-subroutine return address stack. The return address stack provides a 16 address stack for storing micro-subroutine return addresses. Addresses can be pushed or popped from the stack under microprogram control to provide a nested subroutine capability.

III. MICROPROGRAM OPERATIONS

Micro Word Format

The detailed operation and applicable bit patterns of each field of the V73 micro-instruction word will be described. Use of the Varian micro assembler minimizes the need for a user to become intimately familiar with all aspects of V73 microprogramming to write effective microprograms. Obviously, the resources of the V73 may be fully exploited by the sophisticated user who becomes familiar with the functions of all microinstruction fields. The micro word will be described in this section from the functional point of view. Rather than examine each field in detail the functions:

- data path manipulation
- addressing
- I/O control
- memory control
- status control
- register field extraction
- miscellaneous control;

will be described in the succeeding sections.

Figure 1 shows the names and bit positions of each field in the 64 bit micro-instruction word. These names will be used in the following description.

TS | AF | MS | MT | FS | T | S | G | MR | AB | IM | LB | LA

ROM ADDRESSING FIELDS SEE TABLE 2.

[AB = 01 ∨ 10]

TS3-0 selects a four bit field from C2I masked by MR and replace the normal A or B field of the microinstruction

IØREQ

MT = IOAD7

T = 0: no test
T = 2: cond. met = TST cond.
T = 3: cond. met = $\overline{TST}$ cond.

Sel. D-ROM & Samp. Int.

TS0: Enable I/O INTRPT
TS1: Enable I/O INTRPT IF BINS
TS2: Enable MP INTRPT (OINP)
TS3: Enable CP INTRPT (STEP)

If: CEAFS+

FS selects a five bit field from C2I to be masked by MS for use as the least significant five bits of CEAD. See table 1.

The field select outputs are labeled CFS(4-0) in table 2.

Page Jump

CEAD12 = TS3
CEAD11 = TS2
CEAD10 = TS1
CEAD9 = TS0

T = 0

S = 0: IM is special control
S = 1: IM is memory control
S = 3: Test cond. IM is mem. control

REQ Iϕ

IOAD5 = TS3
IOAD4 = TS2
IOAD3 = TS1
IOAD2 = TS0

T ≠ 0

S = 0: IM is special control
S = 1: IM is memory control
S = 2: Test cond: IM is mem. control
S = 3: Test cond. IM is mem. control

Table 1

FSN	CFS4	CFS3	CFS2	CFS1	CFS0
FS0	1	1	1	1	1
FS1	1	1	1	1	CINTF-
FS2	1	C2IO1	1	DSB+	DBAD+
FS3	1	1	1	1	NSTP+
FS4	C2IO4	C2IO3	C2IO2	C2IO1	C2IO0
FS5	5	4	3	2	1
FS6	6	5	4	3	2
FS7	7	6	5	4	3
FS8	8	7	6	5	2
FS9	9	8	7	6	5
FSA	C2I10	9	8	7	6
FSB	11	C2I10	9	8	7
FSC	12	11	C2I10	9	8
FSD	13	12	11	C2I10	9
FSE	14	13	12	11	C2I10
FSF	15	14	13	12	C2I11

TABLE 2

ROM Addressing

Field	Condition	CEADn+ 8	7	6	5	4	3	2	1	0
IIA (3-0)+	If: CADEQ1+ then	1	1	1	1	1	1	1	1	1
DROM	If: CEAIA+ then						IIA3	IIA2	IIA1	IIA0
AF (4-0)	If: CACIDE+ then	DROM8	DROM7	DROM6	DROM5	DROM4	DROM3	DROM2	DROM1	DROM0
MS (3-0)∧CFS (3-0)	If: CEAFS+ then	AF4	AF3	AF2	AF1	AF0	MS3 ∧ CFS3	MS2 ∧ CFS2	MS1 ∧ CFS1	MS0 ∧ CFS0
MT∧CFS4	If: CEAFS+∧CRQIO- then					MT∧ CFS4				
TS (3-0)	If: CEATS+ then					TS3	TS2	TS1	TS0	

Definitions:

CADEQ1+ = System reset ∨ control panel int. (step)

CEAIA+ = [Int. req. & Int. enable on] ∧ [Sel. D-ROM & samp. Int.] ∧ [($\overline{\text{Sel. \& rst. int. flag}}$ ∨ set int. flag) ∧ CINTF+]

CACIDE+ = [Inh. D-ROM ∧ sel. D-ROM & samp. Int] ∧ [($\overline{\text{Sel. \& rst. int. flag}}$ ∨ set int. flag) ∧ CINTF+] ∧ CEAIA-

CEAFS+ = [T = 0 ∧ CEAIA- ∧ CACIDE-] ∨ [T = 2 & test cond. tr.] ∨ [T = 3 & test cond. fs.]

CEATS+ = {[Field sel. for A∨B fields] ∧ [Sel. D-ROM & samp. Int] ∧ Page jump ∧ T = 0 ∧ $\overline{\text{CRQIO}}$} ∨ Cond. test & cond. not met

7/26/72

(T = 0) ∧ (S = 0)

G3: SAMPLE OVERFLOW
G2: Sel. D-ROM & Samp Int.
G1: Sample cond. codes
G0: Enable MII → C2I

(T = 0) ∧ (S = 1)

0XX1 Unused
X00X No status change
X01X Set overflow
X10X Reset overflow
X11X Sample OVERFLOW
1XX0 Reset S/M Channel CC F/F
1XX1 Set S/M Channel CC F/F

(T = 0) ∧ (S = 2)

G3: Unused
G2: Page jump
G1: Sample cond. codes
G0: EN PMA START

(T ≠ 0) ∨ [(T = 0) ∧ (S = 3)]

G specifies test cond.

0 overflow cond.
1 I/O sense
2 SS3
3 SS2
4 SS1
5 620/f test
6 equal cond.
7 less than cond. (DAL 15)
8 carry cond.
9 zero cond.
A $\overline{DSB}$
B MIL15
C shift count = 0
D AL15
E DAL15 ≠ DAL14
F QS

(S = 0) ∧ IØREQ

Field Sel. for AVB Fields
1 AB = 1:
A → A
C2I(TS + 3) ∧ MR → B3
C2I(TS + 2) → B2
C2I(TS + 1) → B1
C2I(TS) → B0
AB = 2 B → B
C2I(TS + 3) ∧ MR → A3
C2I(TS + 2) → A2
C2I(TS + 1) → A1
C2I(TS) → A0
AB = 3
A → A
B → B

(MR = 0) ∧ (AB = 0) ∧ IØREQ

Control Store → A, B

(MR = 1) ∧ (AB = 0) ∧ IØREQ

A → A
1 → B3, B2, B1
(W ∧ DAL15) ∨ (W ∧ DOR01) → B0

IØREQ

AB1 = IOAD1
AB0 = Set FF

IØREQ

Controls AB field

IØREQ

MR = IOAD6

00: File B → Latch B
01: B-MX → Latch B
10: C2I ∧ $\overline{MSR}$ → Latch B
11: $\overline{MSR}$ → Latch B

LA = 2∨3

File A (00) → Byte flag (DBAD+)

00: File A → Latch A
01: P-Reg. → Latch A
10: File A (left 1 bit) → Latch A
11: File A (right 1 bit) → Latch A

S = 0

0000 No action
0001 Wait for mem. done
0010 Wait for I/O done
0011 Unused
0100 Request ALU → MII, MIL
0101 Unused
0110 Select & rst. intrpt. flag
0111 Set interrupt flag
1000 Load I/O key reg.
1001 Unused
1010 Reset supervisor key
1011 Set supervisor key
1100 Inh. D-ROM
1101 Unused
1110 Req. I/O ← inh. D-ROM
1111 Req. I/O

(S = 1) ∨ [(T = 0) ∧ (S = 2)]

00XX Override mem.
01XX ALU → L
10XX P → L
11XX MIL → L
} Start mem.

(S ≠ 0)

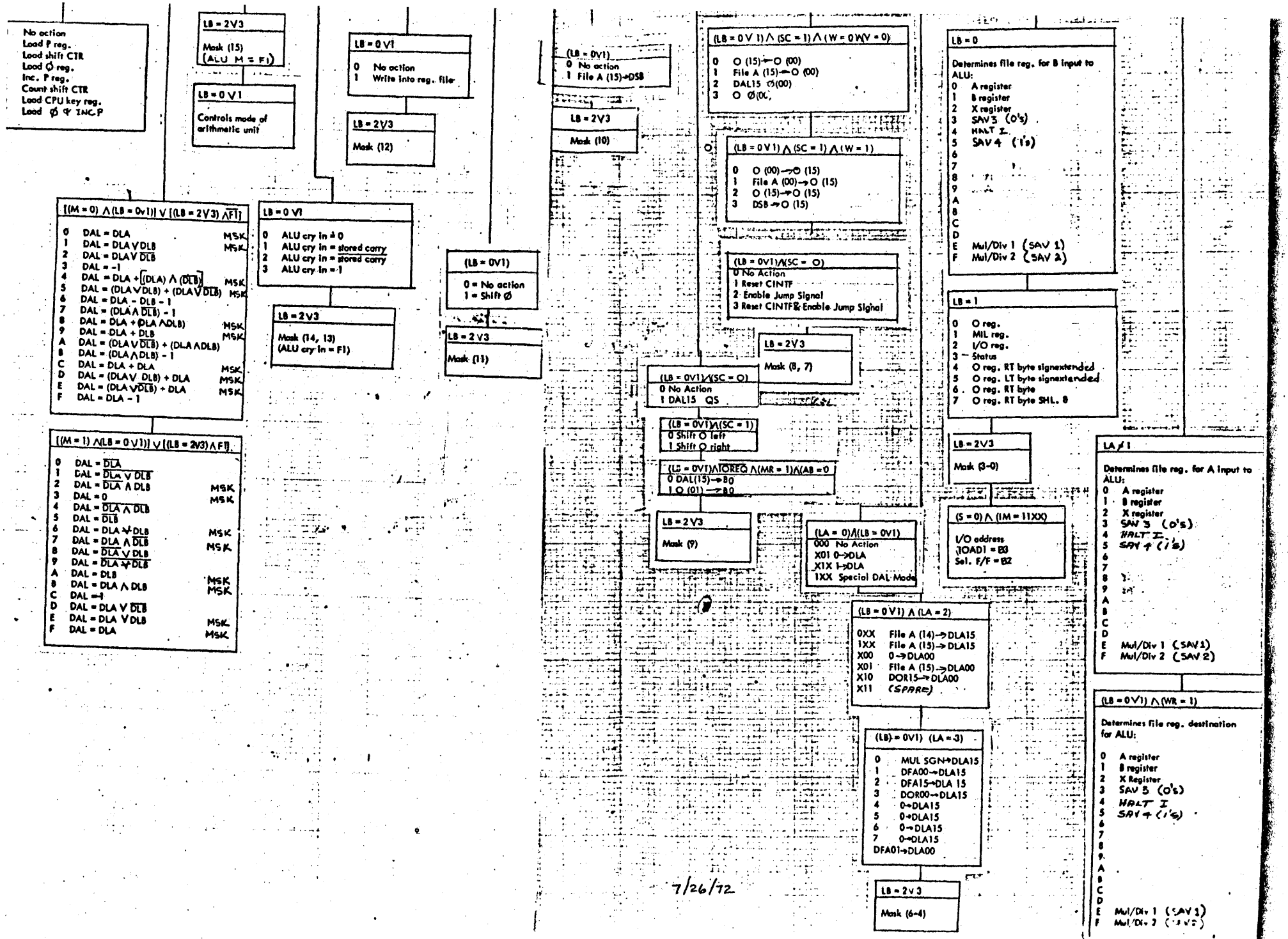
XX00 Read MM into MII & MIL
XX01 Read MM into MIL
XX10 Write word into MM
XX11 Write byte into MM

(T ≠ 0) ∧ (S = 2)

00XX Override mem. if $\overline{\text{test cond.}}$
01XX ALU → L
10XX P → L
11XX MIL → L
} start mem. if $\overline{\text{test cond.}}$

(S = 3)

00XX OVRDE mem. if test cond.
01XX ALU → L
10XX P → L
11XX MIL → L
} Start mem. if test cond.


No action
Load P reg.
Load shift CTR
Load Ø reg.
Inc. P reg.
Count shift CTR
Load CPU key reg.
Load Ø ↑ INC P
LB = 2V3
Mask (15)
(ALU M = F1)
LB = 0 V1
Controls mode of arithmetic unit
LB = 0 V1
0 No action
1 Write into reg. file
LB = 2V3
Mask (12)
[(M = 0) ∧ (LB = 0v1)] V [(LB = 2V3) ∧F1]
0 DAL = DLA MSK
1 DAL = DLA V DLB MSK
2 DAL = DLA V $\overline{DLB}$
3 DAL = -1
4 DAL = DLA + [(DLA) ∧ ($\overline{DLB}$)] MSK
5 DAL = (DLA V DLB) + (DLA V $\overline{DLB}$) MSK
6 DAL = DLA - DLB - 1
7 DAL = (DLA ∧ $\overline{DLB}$) - 1
8 DAL = DLA + (DLA ∧ DLB) MSK
9 DAL = DLA + DLB MSK
A DAL = (DLA V $\overline{DLB}$) + (DLA ∧ DLB)
B DAL = (DLA ∧ DLB) - 1
C DAL = DLA + DLA MSK
D DAL = (DLA V DLB) + DLA MSK
E DAL = (DLA V $\overline{DLB}$) + DLA MSK
F DAL = DLA - 1
[(M = 1) ∧ (LB = 0 V1)] V [(LB = 2V3) ∧ F1]
0 DAL = $\overline{DLA}$
1 DAL = $\overline{DLA}$ V DLB
2 DAL = $\overline{DLA}$ ∧ DLB MSK
3 DAL = 0 MSK
4 DAL = $\overline{DLA \wedge DLB}$
5 DAL = $\overline{DLB}$
6 DAL = DLA ∀ DLB MSK
7 DAL = DLA ∧ $\overline{DLB}$ MSK
8 DAL = $\overline{DLA}$ V DLB
9 DAL = $\overline{DLA \forall DLB}$
A DAL = DLB
B DAL = DLA ∧ DLB MSK MSK
C DAL = -1
D DAL = DLA V $\overline{DLB}$
E DAL = DLA V DLB MSK
F DAL = DLA MSK
LB = 0 V1
0 ALU cry in ≠ 0
1 ALU cry in = stored carry
2 ALU cry in = stored carry
3 ALU cry in = -1
LB = 2V3
Mask (14, 13)
(ALU cry in = F1)
(LB = 0V1)
0 = No action
1 = Shift Ø
LB = 2 V3
Mask (11)
(LB = 0V1)
0 No action
1 File A (15)→DSB
LB = 2V3
Mask (10)
(LB = 0 V 1) ∧ (SC = 1) ∧ (W = 0 W V = 0)
0 O (15)←O (00)
1 File A (15)←O (00)
2 DAL15 Ø(00)
3 O Ø(0(,
(LB = 0 V 1) ∧ (SC = 1) ∧ (W = 1)
0 O (00)→O (15)
1 File A (00)→O (15)
2 O (15)→O (15)
3 DSB→O (15)
(LB = 0V1)∧(SC = O)
0 No Action
1 Reset CINTF
2 Enable Jump Signal
3 Reset CINTF& Enable Jump Signal
LB = 2V3
Mask (8, 7)
(LB = 0V1)∧(SC = O)
0 No Action
1 DAL15 QS
(LB = 0V1)∧(SC = 1)
0 Shift O left
1 Shift O right
(LB = 0V1)∧OREG ∧(MR = 1)∧(AB = 0
0 DAL(15)→B0
1 O (01) →B0
LB = 2V3
Mask (9)
(LA = 0)∧(LB = 0V1)
000 No Action
X01 0→DLA
X1X I→DLA
1XX Special DAL Mode
(LB = 0 V1) ∧ (LA = 2)
0XX File A (14)→DLA15
1XX File A (15)→DLA15
X00 0→DLA00
X01 File A (15)→DLA00
X10 DOR15→DLA00
X11 (SPARE)
(LB) = 0V1) (LA = 3)
0 MUL SGN→DLA15
1 DFA00→DLA15
2 DFA15→DLA 15
3 DOR00→DLA15
4 0→DLA15
5 0→DLA15
6 0→DLA15
7 0→DLA15
DFA01→DLA00
LB = 2 v 3
Mask (6-4)
LB = 0
Determines file reg. for B input to ALU:
0 A register
1 B register
2 X register
3 SAV 3 (0's)
4 HALT I
5 SAV 4 (1's)
6
7
8
9
A
B
C
D
E Mul/Div 1 (SAV 1)
F Mul/Div 2 (SAV 2)
LB = 1
0 O reg.
1 MIL reg.
2 I/O reg.
3 Status
4 O reg. RT byte signextended
5 O reg. LT byte signextended
6 O reg. RT byte
7 O reg. RT byte SHL. 8
LB = 2V3
Mask (3-0)
(S = 0) ∧ (IM = 11XX)
I/O address
IOADI = B3
Sel. F/F = B2
LA = 1
Determines file reg. for A input to ALU:
0 A register
1 B register
2 X register
3 SAV 3 (0's)
4 HALT I
5 SAV 4 (1's)
6
7
8
9
A
B
C
D
E Mul/Div 1 (SAV 1)
F Mul/Div 2 (SAV 2)
(LB = 0V1) ∧ (WR = 1)
Determines file reg. destination for ALU:
0 A register
1 B register
2 X Register
3 SAV 3 (0's)
4 HALT I
5 SAV 4 (1's)
6
7
8
9
A
B
C
D
E Mul/Div 1 (SAV 1)
F Mul/Div 2 (SAV 2)
7/26/72

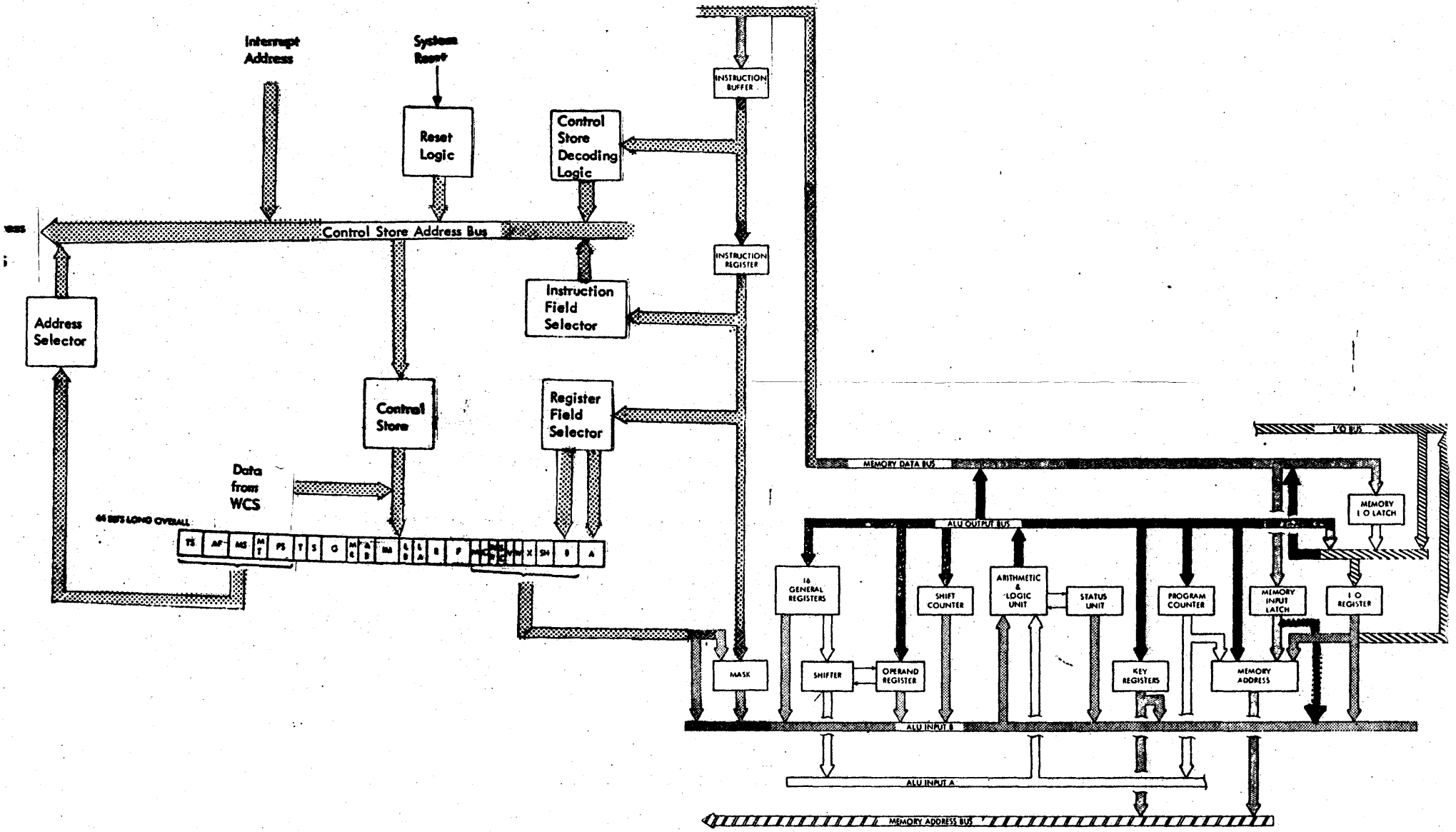


Figure A.

V73 Processor
- Simplified Block Diagram

Data Path Manipulation

The V73 processor's data paths are shown in figure A.
Major data buses are:

- Arithmetic and logic unit (ALU) output bus
- ALU input A bus
- ALU input B bus
- Memory data bus
- Memory address bus
- I/O bus (E-bus)
- I/O register input bus

ALU Operations

The ALU is the only element which drives the ALU output bus. It performs a large variety of functions under control of the F, M and C fields of the micro instruction. The M field specifies the ALU's mode of operation: logical (M = 1) or arithmetic (M = 0). The C field specifies the carry input to the ALU.

- Carry in equals zero (C = 00)
- Carry in equals stored carry (C = 01)
- Carry in equals stored carry complement (C = 10)
- Carry in equals one (C = 11)

The F field selects the particular operation to be performed. The following table (Table 1) lists the ALU output where A is the ALU input A bus data and B is the ALU input B bus data. Arithmetic operations are modified by the carry input selection.
Symbols used are:

$\forall$	Exclusive OR
V	Logical OR

$\wedge$	Logical AND
+	Two's complement addition
-	Two's complement subtraction

A bar over an input term (e.g. $\overline{A}$) refers to the one's complement of that input.

F Field Bit		Arithmetic		Logical
3 2 1 0		(M = 0)*		(M = 1)
0 0 0 0	(1)	A		$\overline{A}$
0 0 0 1	(1)	$A \vee B$		$\overline{(A \vee B)}$
0 0 1 0		$A \vee \overline{B}$	(1)	$\overline{A} \wedge B$
0 0 1 1		-1 (All ones)	(1)	0 (Zero)
0 1 0 0	(1)	$A + [(A \wedge \overline{B}]$		$\overline{A \wedge B}$
0 1 0 1	(1)	$(A \vee B) \quad (A \vee \overline{B})$		$\overline{B}$
0 1 1 0		A - B - 1	(1)	$A \not\vee B$
0 1 1 1		$(A \wedge \overline{B})$ - 1	(1)	$A \wedge B$
1 0 0 0	(1)	$A + (A \wedge B)$		$\overline{A} \vee B$
1 0 0 1	(1)	A + B		$\overline{A \not\vee B}$
1 0 1 0		$(A \vee \overline{B}) + (A \wedge B)$	(1)	B
1 0 1 1		$(A \wedge B)$ - 1	(1)	$A \wedge B$
1 1 0 0	(1)	A + A		-1 (All ones)
1 1 0 1	(1)	$(A \vee B) + A$		$A \vee \overline{B}$
1 1 1 0		$(A \vee \overline{B}) + A$	(1)	$A \vee B$
1 1 1 1		A - 1	(1)	A

*When in the arithmetic mode the selected carry is added two's complement to the result above for arithmetic operations (M = 0) carries selectable are:

C Field	Carry
00	0 (Zero)
01	Stored carry
10	Stored carry complemented
11	1 (One)

NOTE 1: If LB = 10 or 11, the M and C fields are part of a literal 16 bit mask. In this case, the ALU mode is automatically forced to the state of the F field bit 1 and the carry is a zero.

NOTE 2: If LA = 00 and LB = 0X, the ALU function may be forced in accordance with the contents of certain instruction register bits. For complete specification see the section on Miscellaneous Control.

TABLE 1. Summary of ALU Operations

ALU Input A Bus

Selection of the ALU input A bus is performed by the LA and A fields and is modified for certain shift operations by the SH field.

To select the program counter for the ALU input A bus, the LA field is set to a code of 01. No other fields are required to make this selection.

Any of the 16 general registers may be applied to the ALU input A bus by setting the LA field to 00 and placing the binary code of the register number in the A field (0000 through 1111).

Any of the 16 general registers may be applied to the ALU input A bus shifted either left (LA = 10) or right (LA = 11) by one bit position. The registers to be used is specified by the A field.

When shifting left (LA = 10) the selection of what to apply to bits 15 and 00 of the ALU input A bus is made by the SH field. Bit 15 (the sign bit) will be equal to the selected register's bit 14 if the SH field is 0XX. (NOTE: An X in any bit position designates that it's state is irrelevant to the operation discussed). Bit 15 will be equal to the selected register's bit 15 if the SH field is 1XX. This permits either

logical or arithmetic shifts to be performed. (In arithmetic shifts the sign bit is unaffected by the shift operation). Bit 00 of the ALU input A bus is selected to be a zero if the SH field is X00. This is for open single length shifts. Bit 00 will be equal to the selected register's bit 15 of the SH field is X01. This permits circular single length shifts. Bit 00 of the ALU input A bus will be equal to the operand register's bit 15 if the SH field is X10. This permits double length shifts.

When shifting right (LA = 11) the selection of what to apply to bit 15 of the ALU input A bus is made by the SH field. Bit 15 will be equal to the selected register's bit 00 if SH equals 001. This permits single length closed shifts. If SH equals 010 bit 15 will be equal to the selected register's bit 15 to permit arithmetic shifts. If SH equals 011 bit 15 will be equal to the operand register's bit 00 permitting double length shifts. If SH equals 1XX bit 15 will be equal to zero. If SH equals 000 bit 15 will be set to the logical state of the processor's multiply sign flag.

The ALU input A bus will be forced to all zeros when the LA field equals 00 and the 16 bit literal mask is not to be used (LB not equal to 10 or 11) if the SH field equals X01.

The ALU input A bus will be forced to all ones when the LA field equals 00 and the 16 bit literal mask is not to be used if the SH field equals X1X.

The ALU input A bus selections are summarized in Table 2.

ALU Input A Bus Source	Fields			
	LA	SH	LB	A
Program counter	01	XXX	XX	XXXX
General register (any one of 16)	00	Neither X01 nor X1X	0X	Specifies register
General register (any one of 16)	00	XXX	1X	Specifies register
All zero's input	00	X01	0X	XXXX
All one's input	00	X1X	0X	XXXX
General register shifted left	10	See below	0X	Specifies register
Bit 15 = register bit 14		0XX		
Bit 15 = register bit 15		1XX		
Bit 00 = zero		X00		
Bit 00 = register bit 15		X01		
Bit 00 = operand register bit 15		X10		
General register shifted right	11	See below	0X	Specifies register
Bit 15 = multiply sign flag		000		
Bit 15 = register bit 00		001		
Bit 15 = register bit 15		010		
Bit 15 = operand register bit 00		011		
Bit 15 = zero		100		

Table 2. ALU Input A Bus Selections

ALU Input B Bus

Selection of the ALU input B bus is performed by the LB and B fields.

To select one of the 16 general registers for the ALU input B bus the LB field is set to a code of 00. The register used is specified by the B field.

Other special registers are specified with the LB field equal to 01. The operand register is selected by a code of 0000 in the B field. The memory input register is selected by a code of 0001 in the B field. The I/O register is selected by a code of 0010 in the B field. The processor status word is selected by a code of 0011 in the B field. B field codes of 0100, 0101, 0110 and 0111 are used to perform byte operations on the contents of the operand register. These are summarized in Table 3.

Any 16 bit literal constant may be applied to the ALU input B bus by selecting a code of 11 for the LB field. When this code exists the M, C, WR, SC, V, W, X, SH and B fields are treated as the one's complement of the desired binary constant.

The contents of the instruction register may be applied to the ALU input B bus by selecting a code of 10 for the LB field. When this code exists the M, C, WR, SC, V, W, X, SH and B fields are used to mask their corresponding instruction register bits. If a bit in the micro instruction word mask is a one, the corresponding bit of the instruction register appears with ALU input B bus. A summary of ALU input B selections appears in Table 3.

?zero?

ALU Input B Bus Source	Fields		
	LB	B	Remarks
General register (any one of 16)	00	Specifies register	
Operand register full word	01	0000	
Operand register right byte with sign extended	01	0100	
Operand register left byte with sign extended	01	0101	
Operand register right byte with zero's in left byte	01	0110	
Operand register right byte in left byte position; zero's in right	01	0111	
Memory input register	01	0001	
I/O register	01	0010	
Processor status word	01	0011	
16 bit literal constant consisting of the one's complement of fields M, C, WR, SC, V, W, X, SH and B	10	Part of constant	Note
Instruction register masked by 16 bit literal constant consisting of fields M, C, WR, SC, V, W, X, SH and B. A one in the mask fields forces the corresponding ALU input bit to a zero. ? Contradicts previous page	11	Part of mask	Note

NOTE: When the 16 bit literal or mask is used, the ALU mode is forced to the arithmetic mode if the F field bit 1 is a zero and to the logical mode if the F field bit 1 is a one. A carry of zero is forced. The ALU output may not be written into any general register.

Table 3. ALU Input B Bus Selections

Destinations of ALU Output Bus

Data from the ALU may be directed to a variety of destinations. Some of these are under direct control of the central control micro instruction. Others are under control of the memory and I/O control sections of the processor. These sections accept tasks from the central control and thus indirect control of ALU data destination is achieved.

Direct control may cause ALU data to be written into any of the 16 general registers under control of the WR, LB and A fields. If the WR field is a 1, the ALU output data will be written into the register designated by the A field unless the WR field is being used as part of the 16 bit literal (LB = 1X).

Other directly controlled destinations of ALU data are under control of the R field. They are:

- The program counter (R = 001)
- The operand register (R = 011 or 111)
- The shift counter (R = 010)
- The CPU key register (R = 110)

Destinations of ALU data controlled indirectly by the memory control are:

- Memory Data Bus: When a write to memory operation is specified, (S field not equal to 00 and IM field equal to XX1X) the ALU output data is applied to the memory data bus during the resulting memory cycle.
- Memory Address Register: When any memory cycle is initiated with the IM field equal to 01XX, the ALU output data will be loaded into the memory address register and applied to the memory address bus.

- Memory Input Register and Instruction Buffer: When a special transfer is initiated by setting the S field equal to 00 and the IM field equal to 0100 the ALU output data is transferred to the instruction buffer and the memory input register via the memory bus. This operation requires two micro steps.

The I/O register may be loaded with ALU output data under control of the I/O control. This operation is initiated by a request to the I/O mode by setting the S field equal to 00 and the IM field equal to 111X. The I/O operation specified by the TS, MR and AB fields will determine the timing and exact type of operation to be performed.

Table 4 summarizes the ALU output data destinations.

Destination	Control Fields					
	R	WR	A	S	IM	LB
DIRECT CONTROL						
General register (any 1 of 16)		1	Specifies register			0X
Program counter	001					
Operand register	011 or 111					
Shift counter	010					
CPU key	110					
INDIRECT MEMORY CONTROL						
NOTE: Transfer occurs only if cycle is successfully initiated)						
Memory data bus				Not 00	XX1X	
Memory address register				Not 00	01XX	
Memory input register & instruction buffer				00	0100	
INDIRECT I/O CONTROL						
I/O register				00	111X	
NOTE: Transfer is under direct control of I/O control. Operation is specified by TS, AB, MR fields and contents of I/O control store.						

Table 4. ALU Output Data Destinations

Operand Register Shift Operations

As previously noted, the ALU input A bus may have any of the 16 general registers applied shifted left or right one bit position. In addition, the operand register may be shifted left or right independently or in conjunction with shifting of any general register. This can occur any time the 16 bit literal or mask is not in use.

When the LB field is equal to 0X (no literal/mask) the SC, W and X fields define operand register shifting.

When the SC field equals 0 no shifting takes place. When the SC field equals 1,the operand register is shifted left if the W field equals 0 and right if the W field equals 1.

For left shifts the next contents of the operand register bit 00 is specified by the X field. If X equals 00 operand register bit 15 is copied to bit 00 to permit independent circular shifting. If X equals 01 bit 15 of the general register specified by the A field is copied to bit 00. This permits double length circular shifting. If X = 10 the complement of the ALU output bit 15 is copied to bit 00. If X = 11 the operand register bit 00 is set to zero.

For right shifts the next contents of the operand register bit 15 is specified by the X field. If X equals 00 operand register bit 00 is copied to bit 15 to permit independent circular shifting. If X equals 01 bit 00 of the general register specified by the A field is copied to bit 15 to permit double length circular shifting. If X equals 10 the operand register bit 15 is maintained at its current state to permit independent arithmetic shifting. If X equals 11 the divide sign flag (DSB) is copied to bit 15.

Table 5 summarizes the operand register shift operations.

	Control Field				
	LB	SC	W	X	A
No shifting		0			
No shifting	1X				
Shifting of operand register	0X	1			
Left shifting			0		
Bit 00 = operand register bit 15				00	
Bit 00 = general register bit 15				01	Specifies register
Bit 00 = ALU bit 15 complement				10	
Bit 00 = zero				11	
Right shifting			1		
Bit 15 = operand register bit 00				00	
Bit 15 = general register bit 00				01	Specifies register
Bit 15 = operand register bit 15				10	
Bit 15 = DSB (divide sign) flag				11	

Table 5. Operand Register Shift Operations

Program and Shift Counter Counting

The program counter (16 bits) and shift counters (8 bits) may be loaded from the ALU output as previously described. They may be incremented under control of the R field.

To increment the program counter set the R field to 100 or 111. To increment the shift counter set the R field to 101. Either counter will automatically cycle back to zero when it is incremented while it contains all ones.

This overflow from the shift counter may be tested by the micro instruction to cause a branch in the control flow. Shift counter operations characteristically are performed by first loading the two's complement of the desired number of shifts and ~~that~~ then testing the shift count overflow while shifting. The shift continues until the counter overflows where upon a branch is made to the next desired micro instruction.

The shift counter is incremented at the conclusion of the micro instruction. The program counter is incremented during the micro instruction. This is done for timing reasons and permits use of the incremented program counter value for a memory cycle initiated by the same micro instruction. Both program counts and shift counter are loaded at the conclusion of the micro instruction.

Data Loop Flags

A number of one bit flags exist in the V73 processor's data loop. These serve to optimize performance for certain operations. Their names and method of control are summarized below.

DNZT — Normalize flag

Set after any micro instruction during which the ALU output ~~but~~ bit 15 logical state is different from ALU bit 14. It will be ~~sent~~ reset ? after any micro instruction during which ALU output bus bits 15 and 14 are alike. This flag may be tested by a micro instruction

and used to cause a branch to either of two alternate micro instructions.

DSM1 — Multiply sign

Set after any micro instruction during which any of the following three conditions existed:

- ALU output bit 15 and ALU input A bit 15 were both equal to 1.
- ALU output bit 15 and ALU input B bit 15 were both equal to 1.
- ALU input A bit 15 and ALU input B bit 15 were both equal to 1.

This flag may be applied to the ALU input A bus during right shift operations previously described.

DSB — Shift flag

Copies bit 15 of the general register specified by the A field whenever the literal/mask is not being used if the V field equals 1 (LB = 0X and V = 1). This flag may be shifted into the operand register bit 15 as previously described. It may be tested by a micro instruction to cause a branch to either of two micro instructions.

DQS — Quotient sign

Copies bit 15 of the ALU output following any micro instruction in which the literal/mask is not being used if the W field equals 1 and the SC field equals 0 (LB = 0X and W = 1 and SC = 0).

DBAD — Byte address flag

Copies bit 00 of the general register specified by the A field whenever the general register is specified as a shifted input to the ALU input A bus. This flag may be used to determine the address of the next micro instruction and for memory byte write operations (S field not equal to 00 and IM field equal to XX11) determines which byte of the addressed memory location is to be altered. If DBAD equals 0, the left byte is selected. If DBAD equals 1, the right byte is selected.

DOVF — Overflow flag

The overflow flag may be set or reset unconditionally or may sample data loop conditions under control of the T, S and G fields. It is automatically reset by system reset or a micro instruction in which the G field specifies testing of the 620/f test flag with bit 00 of the instruction register set and the test condition is met.

The overflow flag appears as bit 08 of the processor status word.

Operation	Fields				Conditions Bit 15		
					ALU Input		ALU Output
	T	S	G	F	A	B	
Set overflow	00	01	X01X				
Reset overflow	00	01	X10X				
Sample overflow	00	01	X11X				
(ADD)				1XXX			
SET					0	0	1
					1	1	0
DON'T SET*					1	0	X
					0	1	X
(SUBTRACT)				0XXX			
SET					1	0	0
					1	0	1
DON'T SET*					0	0	X
					1	1	X

Also, reset by system reset or a micro instruction specifying test of the 620/f test condition with the instruction register bit 00 on in which the test passes.

Overflow may be sampled to be set if S = 00 and G = 1XXX. It will not be reset even if no overflow exists.

*If set previously, overflow will remain set regardless of sampling conditions.

Table 6. Overflow Flag Control

DCNOZ — ALU zero flag
This flag will be set if sampling is specified when all ALU output bus bits are zero.

DCNDC — ALU carry flag
This flag will be set if sampling is specified and a carry out is generated from the ALU.

DSGN — ALU sign flag
Samples bit 15 of the ALU output bus when sampling is specified.

DEQ — ALU all one's flag
This flag will be set if sampling is specified when all ALU output bus bits are one.

DCNOZ, DCNDC, DSGN and DEQ are sampled after any micro instruction in which the S field equals X0 and the T field is equal to 00 and the G field is equal to XX1X.

These flags may be tested by a micro instruction to cause branching to either of two alternate micro instruction addresses.

They appear as bits in the processor status word.

Processor Status Word

The processor status word may be applied to the ALU input B bus when the LB field equals 01 and the B field equals 0011. Processor status bits are assigned as follows:

Bit	Function
00	Not used (logic 1)
01	Supervisor mode flag
02	ALU zero flag
03	Shift counter bit 00
04	Shift counter bit 01
05	Shift counter bit 02
06	Shift counter bit 03
07	Shift counter bit 04
08	Overflow flag
09	ALU all ones flag
10	ALU sign flag
11	ALU carry flag
12	Processor key register bit 12
13	Processor key register bit 13
14	Processor key register bit 14
15	Processor key register bit 15

MICRO INSTRUCTION ADDRESSING

General

Unlike conventional computer instructions which execute sequentially until altered by a branch instruction, each micro instruction of the V73 must specify the address of the next micro instruction. Through various means of specifying this address the functions of instruction decoding, conditional testing, interrupt handling and normal micro instruction sequencing are accomplished.

The V73 standard 620/f compatible instruction set resides in the first 512 word page of central control store. Writable control store modules of 512 word page size may be added until a total of 8 pages is reached. Logical provision is made for 16 pages but physical limitations of present technology do not permit more than 3 pages.

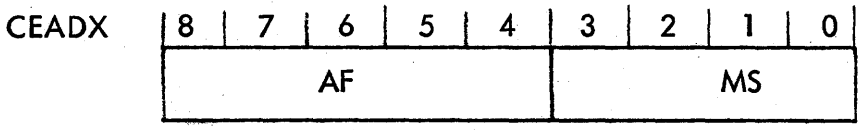
Page Number

Each control store address consists of a four bit page number and a 9 bit word address within that page. The page number is initially set to zero by system reset. Thereafter, page numbers are altered by specification of a page jump by setting the T field equal to 00, the S field equal to 10 and the G field equal to X1XX. The TS field specifies the page number and the word address is specified by other fields as described on the following pages.

ADDRESSING MODES (Refer to figures 2 and 3)

Normal Addressing

Normal addressing is used to arbitrarily specify the next micro instruction address. No conditional testing is involved, no interrupts are active or they are disabled and decoder addressing is not specified. The FS and TS fields are set equal to 0000 and the MT field equals 0 so the low order address contribution (bits 0-3) is governed entirely by the MS field. The high order bits (4-8) are supplied by the AF field.



Control Store Address -- Normal Addressing

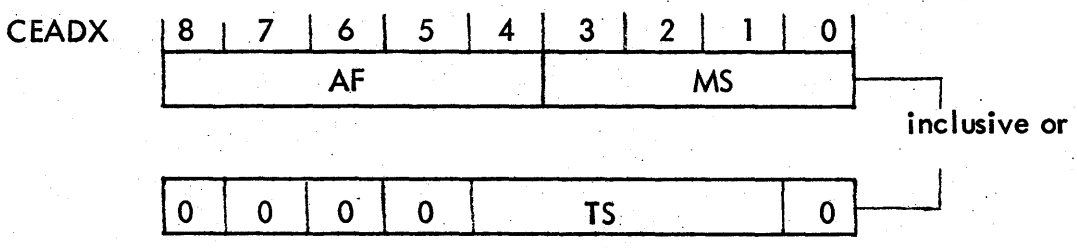
No reset
No interrupts
No decoding
FS = 0000
MT = 0
TS = 0000 or T = 0

Normal Addressing with TS Field

The TS field may be used to form bits 1 through 4 of the control store address when none of the following conditions is true:

a. Register field extraction (AB field equals 01 or 10)
b. Interrupts allowed (S and T fields both 00; G field equals X1XX)
c. I/O request (S field equals 00; IM field equals 111X)
d. Page jump (T field equals 00; S field equals 10; G field equals X1XX)

The address is formed by the inclusive OR of the TS field into bits 1 through 4 of the address obtained with normal addressing (FS field equals to 0000; no decoding; no interrupts, MT field equals 0).



Control Store Address
Normal Addressing with TS Field

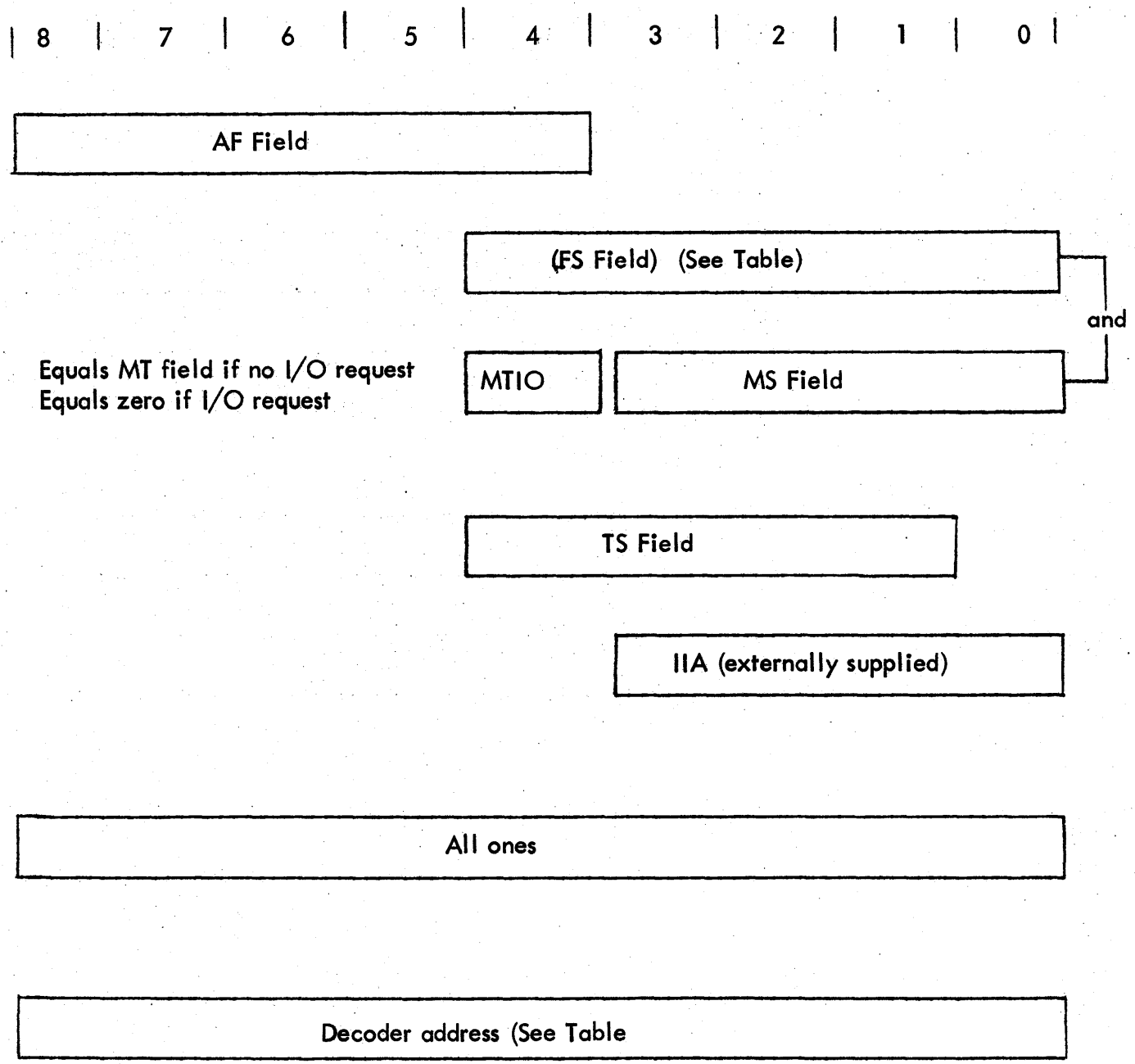


Figure 2. Control Store Address Components

	Field Select or Normal Addressing	Field Select or Normal Addressing if no Testing, no Reg. Field Select, no I/O Request, no Decode or Interrupt or Page Jump	Testing		Decode & No Interrupt	Interrupt	System Reset
			Pass	Fail			
AF	X	X	X	X		X	
S) Λ MS, MT	X	X	X				
TS		X		X			
IIA						X	
ll Ones							X
Decoder					X		

Figure 3. Enabling of Control Store Address Components

Field Select Addressing

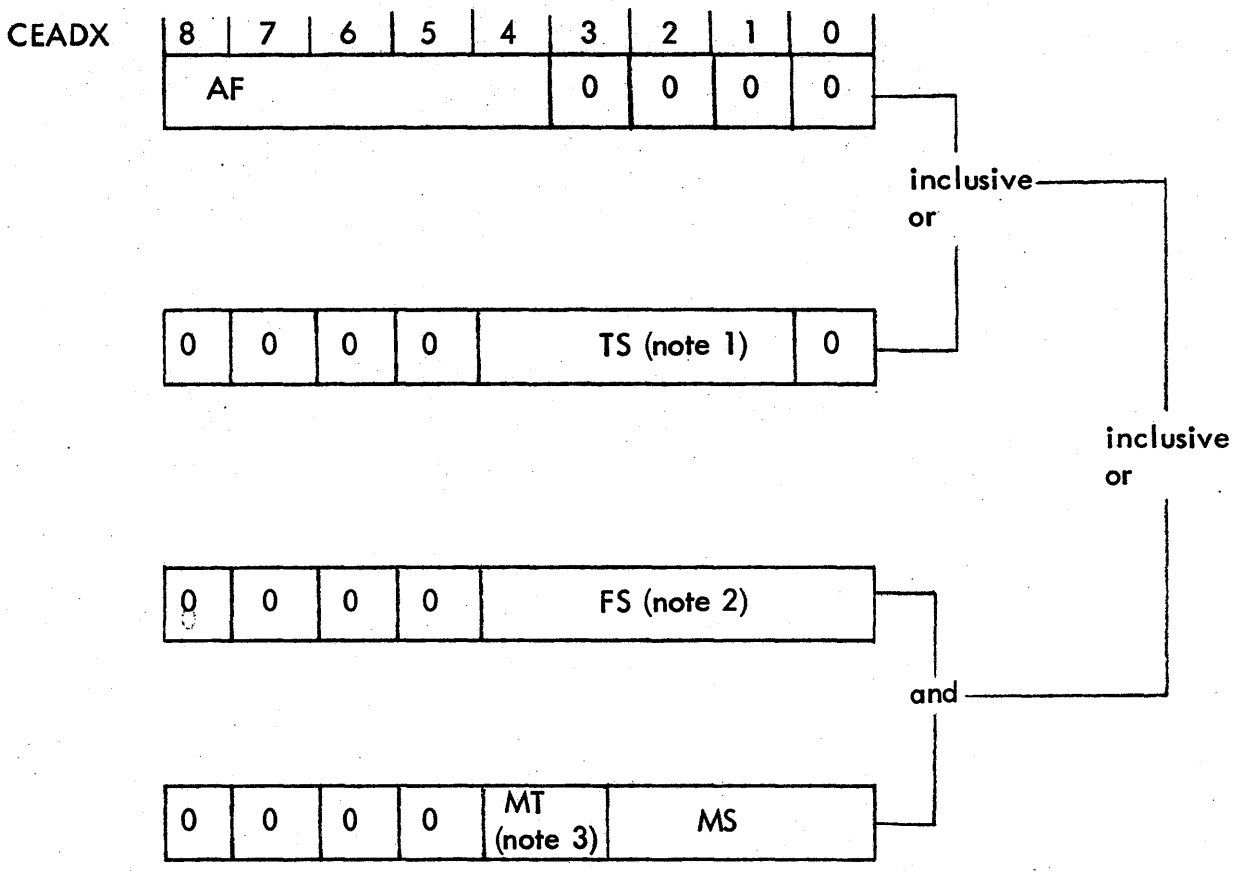
The contents of the instruction register and a number of processor flags may be used to form a control store address. Any one to five bit contiguous field from the instruction register may be used too in forming the low order five bits of control store address. Thus, up to a 32 way branch may be performed based on instruction register contents. This permits detailed instruction decoding. In addition, the interrupt flag, byte address flag, shift flag and console step mode may be selected to alter the control store address.

Field select addressing is used any time the FS field is not equal to 0000. The field selector address contribution for all values of the FS field is shown in Table 7. Any bit of the field select contribution may be forced to a zero by use of the MS and MT fields. The field masks bits 0-3 of the field select contribution. The MT field masks bit 4. A zero in any bit of the MS and MT fields forces the contribution of the corresponding field select bit to zero. When an I/O request is issued (S field equal to 00 and IM field equal to 111X) the MT field is and as part of the I/O operation specification. In this case, the MT field is ignored and bit 4 of the field select address contribution is masked to zero.

The field select address contribution is shown in Table 7 for all values of the FS field.

High order address bits 4 through 8 are provided by the AF field.

The TS field is logically OR'ed into the control store address bits 1 through 4 under the same conditions as normal addressing into TS field. Thus, the composite field select address is formed as follows: (see the following page)



Note 1: TS field is not used in bits 1-4 of address formation when:

a. Register field extraction (AB field equals 01 or 10)

b. Interrupts allowed (S and T fields both 00; IM field equals 111X)

c. I/O request (S field equals 00; IM field equals 111X)

d. Page jump (T field equals 00; S field equals 10; G field equals X1XX)

e. Test addressing is specified (T field not equal 00)

Note 2: (FS) is the contents of the field specified by the FS field as given in Table 7.

Note 3: MT is replaced by a zero when I/O request is present (S field equals 00; IM field equals 111X)

It will be noted that normal addressing and normal addressing with TS field are special cases of field select addressing -- i.e. the FS field equals 0000 and the MT field equals 0.

Control Store Address Bit	4	3	2	1	0	FS Field
	One	One	One	One	One	0
	One	One	One	One	CINTF-	1
	One	01	One	DSB+	DBAD+	2
	One	One	One	One	NSTP+	3
	04	03	02	01	00	4
	05	04	03	02	01	5
	06	05	04	03	02	6
	07	06	05	04	03	7
	08	07	06	05	04	8
	09	08	07	06	05	9
	10	09	08	07	06	A
	11	10	09	08	07	B
	12	11	10	09	08	C
	13	12	11	10	09	D
	14	13	12	11	10	E
	15	14	13	12	11	F

Table 7.
Field Select Address Contribution

Numbers 00 through 15 refer to instruction register bits

CINTF- is the interrupt flag (complement)

DBAD+ is the byte address flag

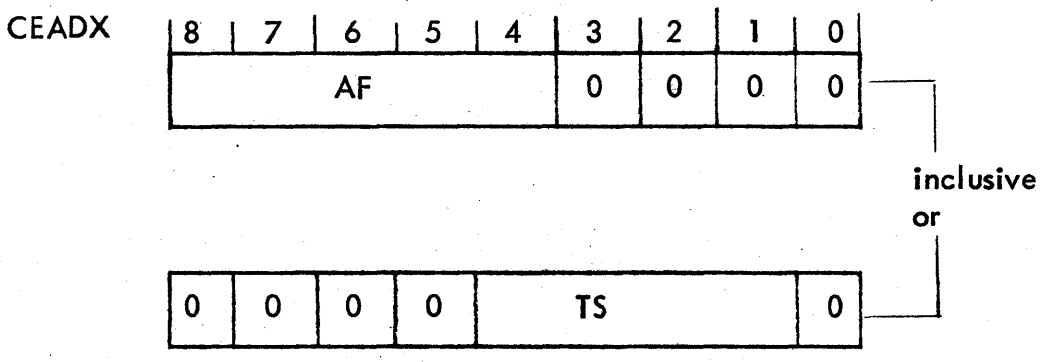
DSB+ is the shift flag

NSTP+ is true when the console is in the STEP mode

Test Addressing

Two addresses must be specified when test operations are performed -- one for use if the test passes and one for use if it fails. Testing is specified whenever the T field is not equal to 00. If the test is to pass when the condition tested is true, the T field must be equal to 10. If the test is to pass when the condition tested is false, the T field must be equal to 11. The condition to be tested is specified by the G field. Test conditions are shown in Table 8.

The address used if the test passes is identical to that formed by field select addressing. The address used if the test fails is made up of the AF and TS fields as shown below.



Control Store Address --
Test Fails

Condition	G Field
Overflow flag	0000
I/O sense response flag	0001
Sense switch 3	0010
Sense switch 2	0011
Sense switch 1	0100
620/f Test The contents of the instruction register determine a set of conditions to be simultaneously tested. Refer to Varian 73 System Handbook pp. 16-25, 26 for a detailed description of instruction register bit assignments.	0101
ALU all one's flag	0110
ALU sign flag	0111
ALU carry flag	1000
ALU zero flag	1001
Shift flag	1010
Memory input register bit 15	1011
Shift counter overflow	1100
General register 0 bit 15	1101
Normalize flag	1110
Quotient sign flag	1111

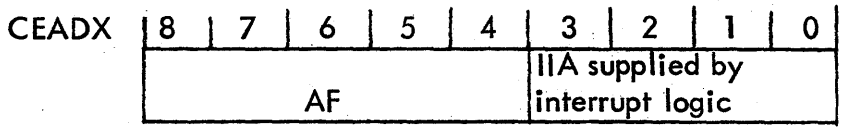
T = 10 test pass if condition is true

T = 11 test pass if condition is false

Table 8. Test Conditions

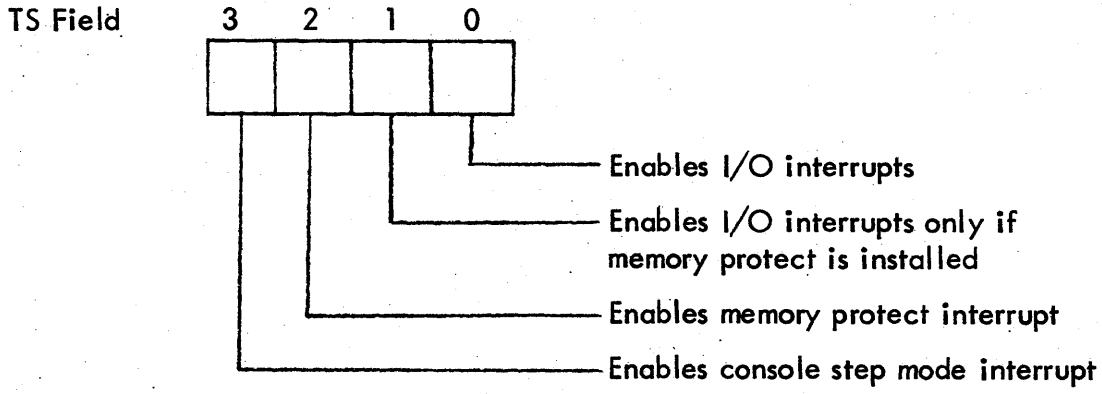
Interrupt Addressing

When interrupts are allowed (both T and S fields equal to 00 and G field equals X1XX) and an interrupt is active and that class of interrupt is enabled by an appropriate bit in the TS field. The low order form bits of the control store address (0 through 3) supplied by the interrupt logic and the high order bits (4 through 8) are supplied by the AF field.



Control Store Address Interrupts

The TS field enables interrupts wherever there are ones as follows:



Page Jump

The micro instruction specifies a branch to a location in another 512 word page by executing a page jump. In this case, a 13 bit address is generated which sets a new active page number and specifies an address within that page. The page number is specified by the TS field. The word address is specified by field select addressing.

CEADX

12	11	10	9	8	7	6	5	4	3	2	1	0
TS				Address as moduled by field select addressing								

Control Store Address
Page Jump

Page jump is specified by the T field equal to 00; the S field equal to 10; and the G field equal to X1XX.

Reset

When system reset is active as a result of pressing the console RESET button or following a power up or preceding a power failure, the control store address is forced to all ones.

Decode Addressing

Preliminary decoding of instructions in the instruction buffer is performed by the instruction decoder control store and instruction decode logic. These elements translate the 16 bit instruction into a 9 bit control store address in accordance with the contents of the instruction decoder control store.

The instruction decoder control store consists of two 16 word by 16 bit memory arrays. The Varian 73 processor implements this with programmable read only memories (PROMS). An option to the writable control store permits selection of read/write memory arrays to permit alternate decoding strategies.

The first decode control store array uses instruction buffer bits 12 through 15 as an address. The second decode control store array uses instruction bits 08 through 11 as an address. The formats for these two control store arrays are shown in figure

Decode addressing is enabled by the T and S fields both equal to 00 and the G field equal to X1XX. If an interrupt is present, decoding is inhibited and interrupt addressing is used.

Decode addressing will be inhibited if the IM field equals 11X0. If decode addressing is so inhibited and no interrupts are present field select addressing is used.

The possible components of a decoded address are shown in figures 4 and 5. The 9 bits labeled CIDA3X obtained from the first decode control store are always used in decode addressing.

The most significant 5 bits (4-8) labeled CIDA2X are included in the control store address bits 4 through 8 by an inclusive or if either of the following bit combinations exist in the first decoder output:

$\overline{T32}$ equals zero

or

$\overline{S32}$ equals zero

The least significant 4 bits (0-3) labeled CIDA2X are included in the control store address bits 0 through 3 by an inclusive or if either of the following bit combinations exist in the first decoder output:

$\overline{T32}$ equals zero and 00 equals one

or

$\overline{S32}$ equals zero and 00 equals one

The contents of instruction buffer bits 04 through 07 are included in the control store address bits 0 through 3 by an inclusive or if either of the following bit combinations exist:

$\overline{S31}$ equals zero

or

$\overline{S32}$ equals zero and S21 equals one

The contents of instruction buffer bits 00 through 03 are included in the control store address bits 0 through 3 by an inclusive or if either of the following bit combinations exist:

$\overline{S30}$ equals zero

or

$\overline{S32}$ equals zero and S20 equals one

One exception to this is the contribution of instruction buffer bits 04 through 07 is that the contribution to control store address bit 2 will be the contents of instruction buffer bit 03 if the first decoder's bit 00 equals one and the second decoder's bit XX5 equals one.

Decode addressing is used to perform a preliminary instruction decoding function. It permits instruction classes to be discriminated with the detailed decoding performed later by field select addressing after the instruction buffer is transferred to the instruction register.

The meaning of other bits in the two decode control store words is shown in figure These signals are available at a processor connector and are used by Varian 73 options to detect certain instruction classes.

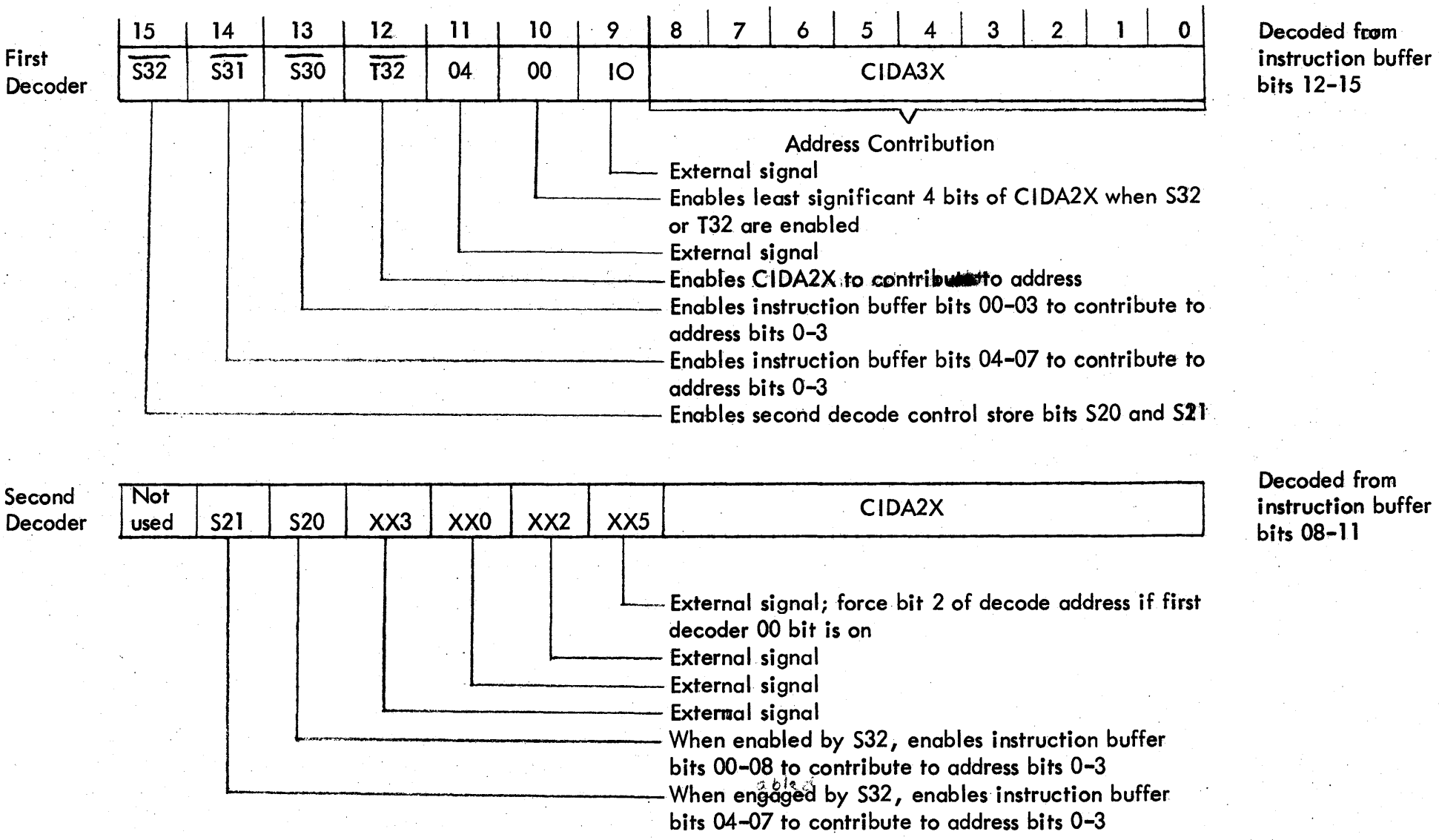
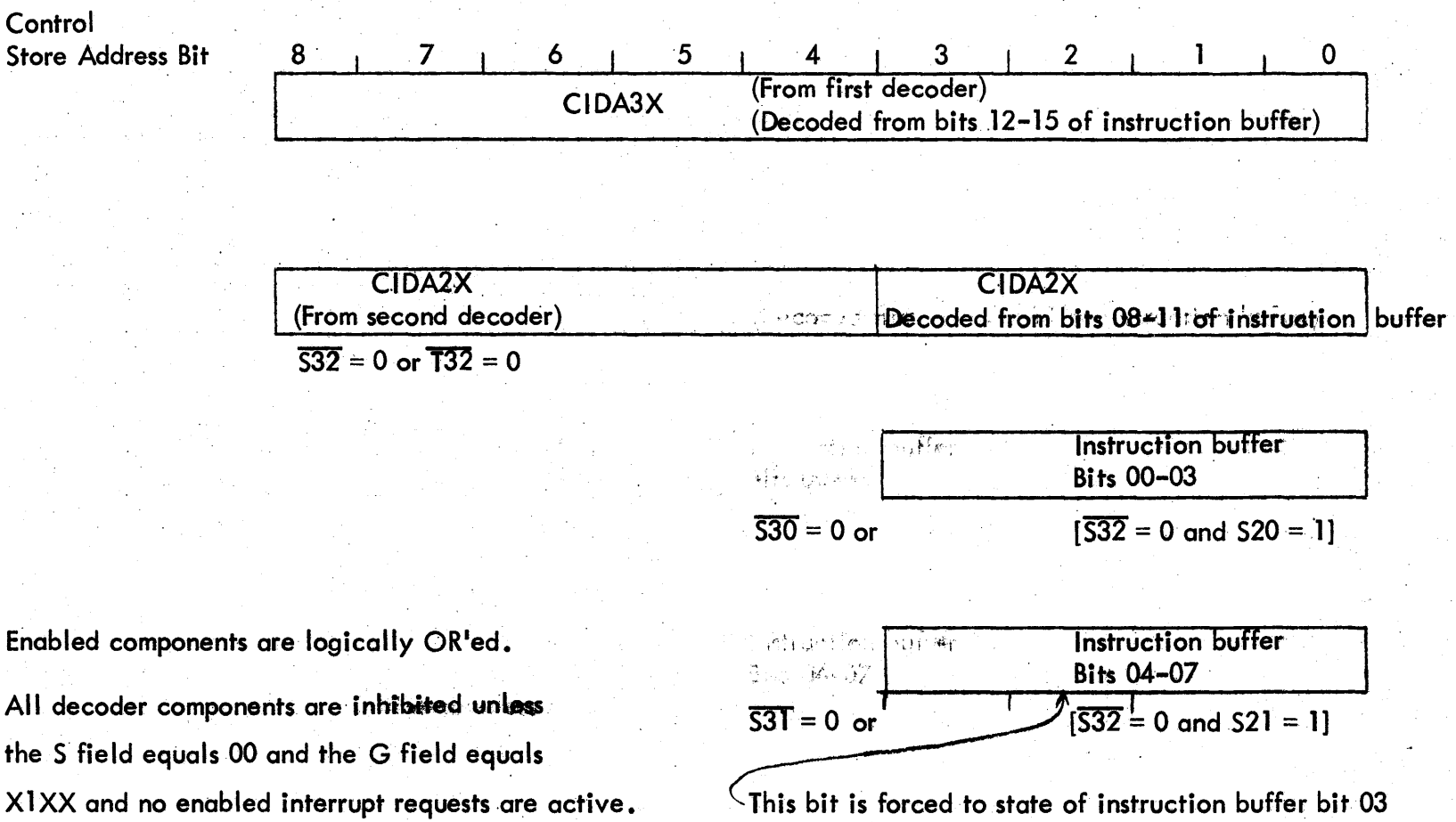


Figure 4.
Decode Control Store Format

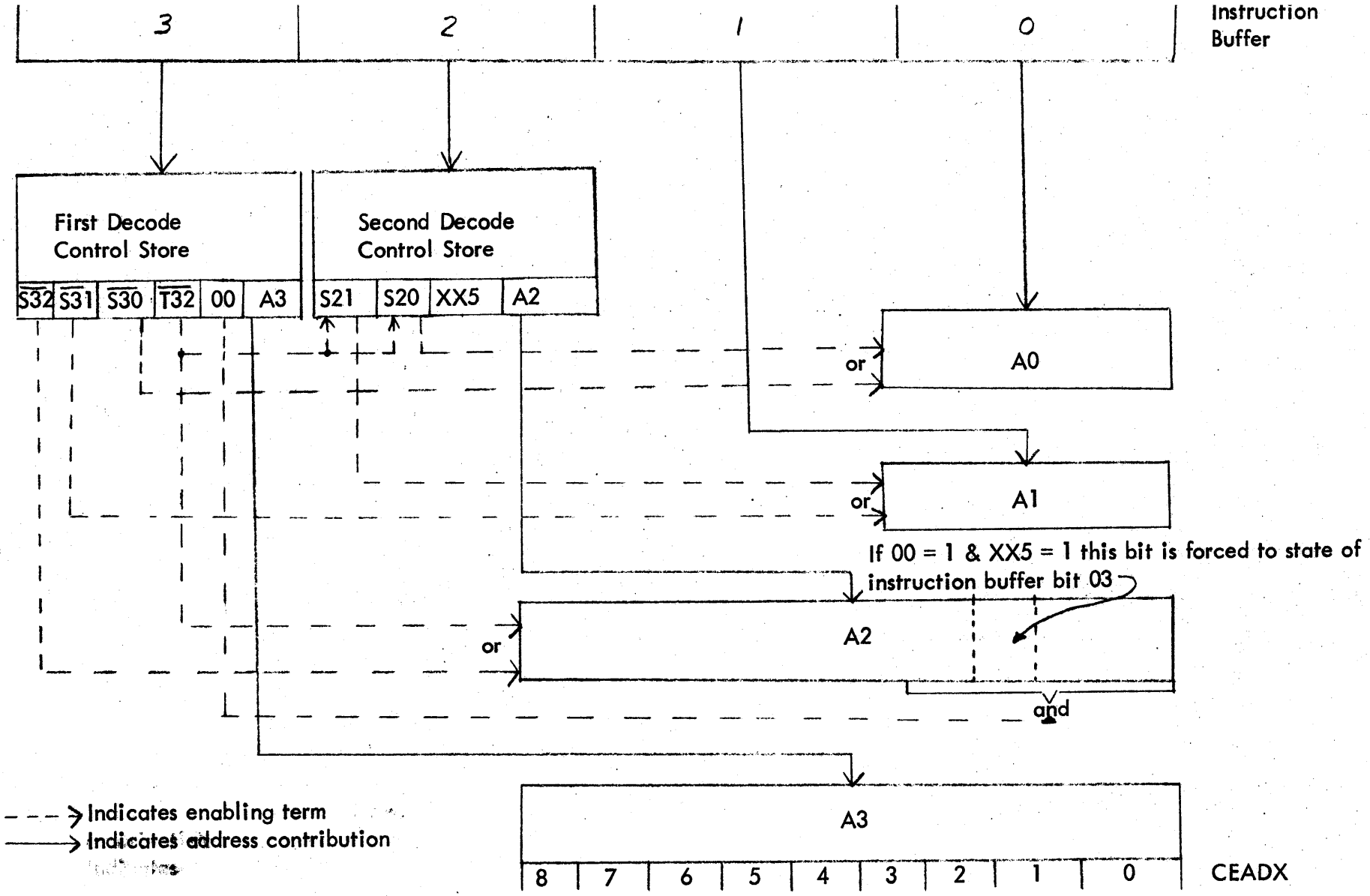


Enabled components are logically OR'ed.

All decoder components are inhibited unless the S field equals 00 and the G field equals X1XX and no enabled interrupt requests are active.

In addition, decoding may be inhibited by the IM field equal to 11X0.

Figure 5.
Decoder Address Components



1. Enabled contributions combined by inclusive or
2. Decode addressing enabled by S = 00; T = 00; and G = X1XX
3. Decode addressing is inhibited by interrupts or IM = 11X0

Figure 6.
Decode Addressing

I/O CONTROL

Microprogram Initiation

The micro instruction can initiate I/O activity by signaling an I/O request while forming a starting address for the independent I/O control store. An I/O request is made by setting the S field equal to 00 and the IM field equal to 111X. (If the IM field equals 1110, decode addressing is inhibited).

The I/O control store starting address is specified by the MT, MR and TS fields.

7	6	5	4	3	2	1	0
MT	MR	TS				AB1*	0

I/O request
S = 00
IM = 111X

I/O Control Store
Starting Address

*AB1 is most significant bit of the AB field

The micro instruction can wait for completion of I/O activity by specifying a wait for I/O done. This is coded by setting the S field equal to 00 and the IM field equal to 0010. Execution of this and subsequent micro instructions will be inhibited until the I/O sequence is completed. If the I/O is busy performing a sequence and an I/O request is issued, execution of the micro instruction specifying new I/O activity will be inhibited until the I/O completes its current sequence.

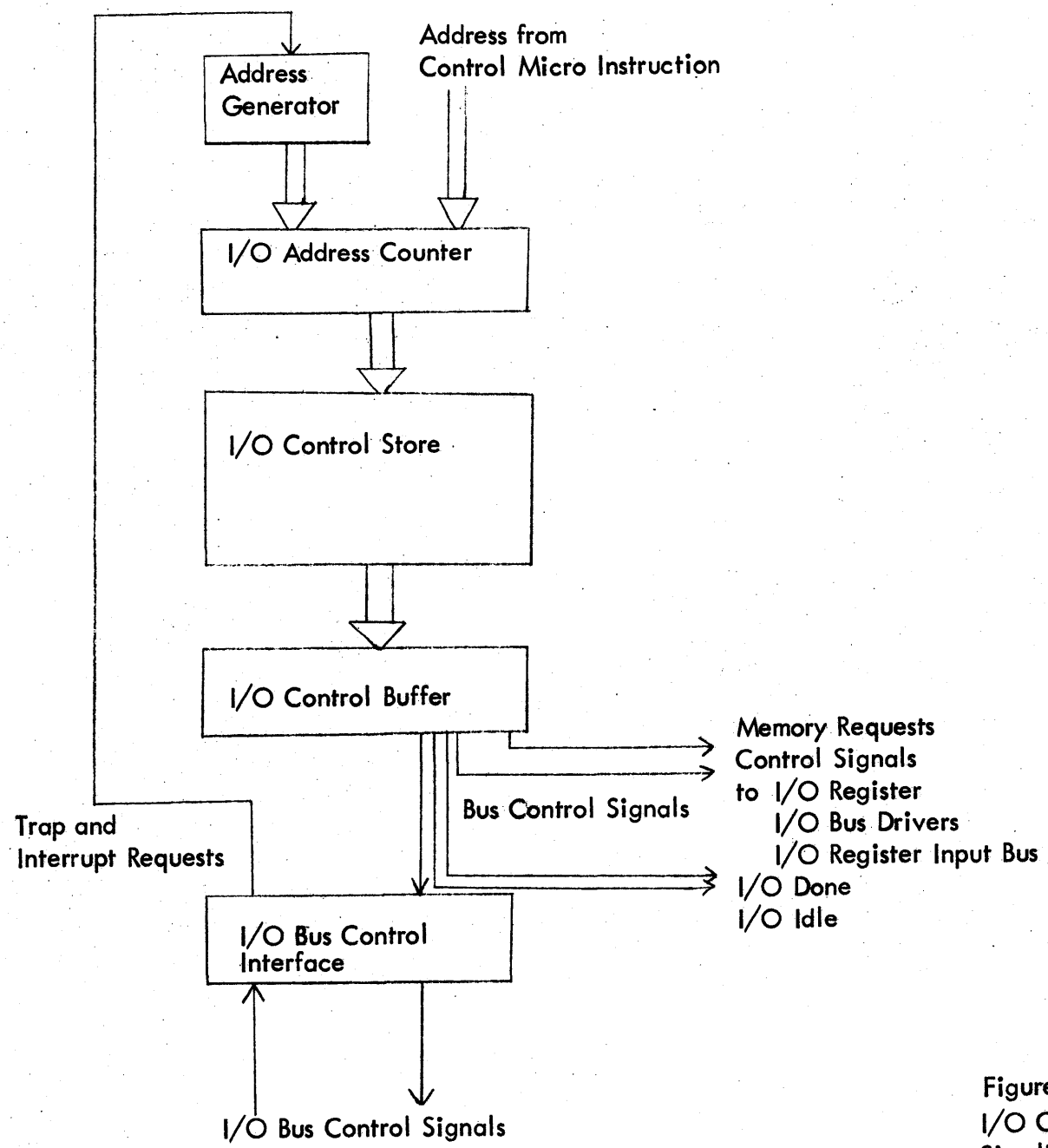


Figure 7.
I/O Control
Simplified Block Diagram

I/O Micro Programming

The I/O control section performs I/O sequences initiated from either the Varian 73 processor microprograms or external DMA trap requests or interrupts.

It performs the following functions in accordance with the sequence of I/O micro instructions stored in the I/O control store:

- Control the source of data applied to the I/O register input bus.
- Control loading on byte shifting of the I/O register.
- Initiate memory cycle requests to the Varian 73 memory control section.
- Initiate I/O bus control signals.
- Wait for completion of external events such as memory cycles, new processor microprogrammed requests, external control signals, etc.
- Signal completion of I/O activity to the processor's central control section.

I/O control store formats are shown in figure 8.

The I/O address counter is automatically incremented at completion of each micro instruction unless a "WAIT" or "IDLE" state is entered. This counter is cleared to zero by system reset.

I/O micro instructions are executed from sequential addresses until the end of the sequence whereupon the I/O becomes idle and ready to accept new requests.

The format of the I/O micro instruction is shown in figure 8.

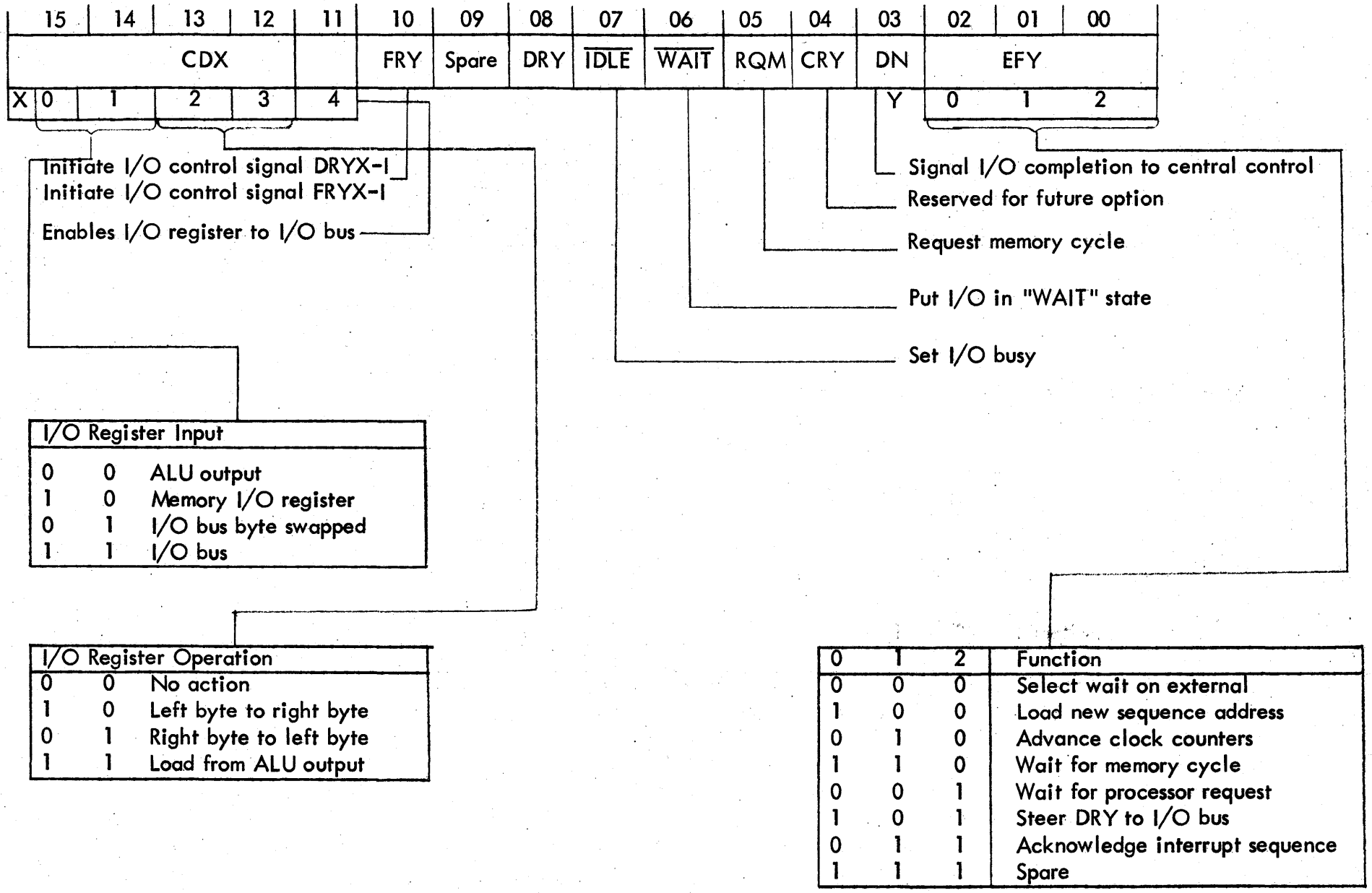


Figure 8.
I/O Micro Instruction Format

As the address counter is loaded with its starting address, the I/O control buffer is loaded with the contents of I/O control store location corresponding to the last contents of the address register. Following a system reset this will be the contents of I/O control store address zero. At all other times it will be the ending address of the previous I/O sequence. In either case, the standard data will cause bits $\overline{\text{IDLE}}$ and DN to become true.

$\overline{\text{IDLE}}$ true indicates the I/O control is not idle and further requests are to be ignored. Al long as $\overline{\text{IDLE}}$ is true, the I/O address counter and I/O control buffer are enabled.

At each succeeding micro instruction time the address counter is incremented and the I/O control buffer is loaded with the contents of the address designated by the address counter. The 16 bits of the I/O control buffer control all I/O functions. Their use is described below:

CD0 Control the processor's
CD1 I/O data loop multiplexor (IOMXX+)

CD		
1	0	I/O Register Input
0	0	ALU
0	1	Memory I/O register
1	0	I/O bus byte swapped
1	1	I/O bus

CD2 Control the processor's
CD3 I/O register

CD		
3	2	
0	0	No action
0	1	Shift right (left byte to right byte)
1	0	Shift left (right byte to left byte)
1	1	Load from ALU

These bits do not directly control the I/O register. The I/O register may also be controlled by $\overline{\text{IDLE}}$ (when the I/O is idle the register is continuously loaded from the ALU).

CD4	Enables the processor's I/O register onto the E-bus.
FRY	Initiates an I/O function ready (FRYX-I) signal. FRYX-I is terminated 247.5 ns. later by signal IIIT-.
Spare	Not used.
DRY	Initiates an I/O bus data ready (DRYX-I) signal. DRYX-I is terminated 247.5 ns. later by signal IEDRYN+ derived from IIIT-.
$\overline{\text{IDLE}}$	Determines idle/busy status of I/O control. While busy the I/O can accept no new requests.
WAIT	Places the I/O control in a "wait" state by inhibiting address counter and ROM buffer clocks until receipt of a designated signal. The I/O may wait for any of the following: • new processor request • processor interrupt flag reset • data memory cycle complete • external wait signal Selection of the specific condition is determined by the function bits EF2, EF1 and EF0 of the I/O control buffer.
RQM	Requests a DMA memory cycle from the processor's memory control.
CRY	Channel request. Reserved for future option.
DN	Results in an I/O done signal (IDNC- low) to signal the processor of completion of the I/O sequence.
EF2	Function bits which control: • selection of "wait" condition • advance of interrupt clock counters • steering of DRY • acknowledge interrupt requests • loading of new sequence addresses

EF			
2	1	0	
0	0	0	Select wait on external signal IEXW+
0	0	1	Load new sequence address from CPU if CRQIO+
0	1	0	Advance IUCX and IUCF clock counters
0	1	1	Select wait for memory cycle complete
1	0	0	Select wait on CPU request
1	0	1	Steer DRY to DRYX-I
1	1	0	Acknowledge interrupt sequence request from CPU
1	1	1	Not used

Any I/O sequence continues through successive ROM addresses until address counter and ROM buffer clocks are inhibited by either of two conditions:

$\overline{\text{IDLE}}$ becomes false signifying end of sequence or

WAIT becomes true signaling that the current sequence must stop to wait for some external event such as:

- memory cycle
- new processor request
- interrupt flag set
- external wait line active

For programmed I/O sequences signal DN will become active and at the next micro instruction time $\overline{\text{IDLE}}$ will become active also. $\overline{\text{IDLE}}$ causes I/O sequencing to stop.

The I/O sequence is thus completed leaving the address counter loaded with an address whose contents $\overline{\text{IDLE}}$ and DN. This will be the first data loaded into the ROM buffer when clocks are re-enabled.

MEMORY CONTROL

General

Memory cycles may be initiated by micro instructions either unconditionally or depending on the results of a test. The micro instruction specifies the type of operation and address source to be used. When the cycle is initiated, the memory control section handles the

complete operation. This permits the microprogram to initiate, for example, an operand fetch and then do other functions while that cycle occurs. The microprogram can detect completion of the memory cycle by specifying a wait for memory done.

A special transfer may be initiated which uses the memory data bus but which does not cycle any memory units. This permits transfer of ALU output data to the instruction buffer and memory input register.

An active memory cycle may have the type of operation changed by the micro instruction following the one which initiated it. This is designated "override" and may occur conditionally on the result of a test.

If the memory control is busy when a micro instruction specifies a new memory cycle, execution of that micro instruction is deferred until completion of the current cycle.

Memory operations are summarized in table

Unconditional Cycle Initiation

A memory cycle is unconditionally initiated or overriden when the S field equals 01 or if the S field equals 10 and the T field equals 00.

The IM field specifies the type of operation and the address source. Permitted operations are:

IM = XX00	Read data from memory into the instruction buffer and memory input register (instruction fetch).
IM = XX01	Read data from memory into the memory input register (operand or address fetch).
IM = XX10	Write the full word output of the ALU into memory.

IM = XX11 Write the byte from the ALU specified by the byte address flag (DBAD) into the corresponding memory byte. The other memory byte at the designated word address is unaffected. If DBAD is false, the left byte is written. If DBAD is true, the right byte is written (override).

The operation may be changed by the following micro instruction by specifying the new operation with the IM field equal to 00XX. This permits, for example, conversion of a store cycle into a fetch or an instruction fetch into an operand fetch.

The data to be written to memory must be maintained at the ALU output by the micro instruction(s) following initiation until the cycle is complete.

The source to be used for loading the memory address register is specified as follows:

IM = 01XX ALU output
IM = 10XX Rrogram counter
IM = 11XX Memory input register (specification of this source)

Conditional Cycle Initiation

A memory cycle may be initiated (or overridden) on not depending on the results of a test specified by the G field. Conditions tested were described previously in the section on Test Addressing.

If the T field is not equal to 00 and the S field equals 10, the cycle will be initiated (or overridden) if the tested condition is false.

If the S field is equal to 11, the cycle will be initiated (or overridden) if the tested condition is true.

In either case, the IM field specifies the operation to be performed and the address source to be used as described in the previous section.

Special Transfer

ALU output data may be transferred to the instruction buffer and memory input register by using the memory data bus. This does not involve activation of any memory module. To initiate this transfer the S field must be equal to 00 and the IA field equal to 0100. The ALU output data must be set up by the initiating micro instruction and maintained for one more micro instruction.

Function	Control Fields S	T	IM	G
UNCONDITIONAL INITIATION	01 or 10	00		
CONDITIONAL INITIATION				
Condition True	11			Specifies Cond.
Condition False	10	Not 00		Specifies Cond.
EITHER				
Operation				
Read memory data into instruction buffer and memory input register			XX00	
Read memory data into memory input register			XX01	
Write ALU word output			XX10	
Write ALU byte output			XX11	
Address Source or Override				
Override operation			00XX	
ALU output			01XX	
Program counter			10XX	
Memory input register			11XX	
SPECIAL TRANSFER (ALU output to instruction buffer and memory input register)	00		0100	

Table 9. Memory Operations

CENTRAL CONTROL FLAGS

Certain flags in the processor's central control may be set or reset by microprogram control to store internal conditions. They are:

CSMF — Channel flag (reserved for a possible future option)

Provides an external control signal at J4-22 under microprogram control. Set or reset when the T field equals 00 and the S field equals 01 in accordance with the G field:

SET	G = 1XX1
RESET	G = 1XX0

CESK — Supervisor key

Provides an external control signal at J6-2 under microprogram control. It also, when set, forces processor initiated memory cycles to be from key zero. This is used for the memory future map option. The actual contents of the key register are preserved and will reappear when the supervisor flag is reset. It is set or reset when the S field equals 00 in accordance with the IM field:

SET	IM = 1011
RESET	IM = 1010

It appears as bit 01 of the processor status word.

CINTF — Interrupt flag

This flag, which may be set or reset under microprogram control, may be used to suppress interrupts when set. It is set when the S field equals 00 in accordance with the IM field:

SET	IM = 1011
SELECT & RESET	IM = 1011

When the IM flag is selected and reset, all interrupts will be blocked and the flag will be reset.

The interrupt flag may be reset without affecting interrupts when the LB field equals 0X (literal/mask not in use) and the SC field equals 0 (no shifting of operand register) if the X field equals X1.

REGISTER FIELD CONTROL

Many types of instruction words contain fields which specify registers which contain operand data. If all combinations of operations on all possible registers had to be specified by individual micro instructions, the control store size would be quite large.

The Varian 73 permits three or four bit fields to be selected from the instruction register and stored and maintained in the control buffer register specification fields. This permits a single micro instruction to handle all combinations of registers for any operation.

This register field extraction is performed independently of the field select addressing function and both may be used simultaneously.

The A and B fields of the micro instruction contained in control store are copied into their corresponding positions in the control buffer any time the AB field equals 00 and the MR field equals 0. This is the normal mode of operation.

When the S field equals 00 and no I/O request is active, the AB field equals 01 or 10; the TS field specifies a four bit field of the instruction register to be loaded into the control buffer's A or B field. The field not being loaded will be maintained at its last value. A code of AB equals 01 and loads the field selected into the B field. A code of AB equals 10 and loads the field selected into the A field.

The MR bit is used to mask the most significant bit of the selected field. If MR equals zero, the most significant bit of the selected field will be treated as a zero. If MR equals one, the most significant bit of the selected field will be loaded into the designated field.

The A and B fields can be maintained in their current state by specifying an AB field equal to 11 while the S field equals 00 and no I/O request is present.

If no I/O request is present, the AB field equals 00 and the MR field equals 1, the control buffer A field will be maintained at its current value and the B field will be forced to either of two addresses depending on data loop conditions and the W field.

W field equal to 1

Operand register bit 01 = 1; B = 1111

Operand register bit 01 = 0; B = 1110

W field equal to 0

ALU bit 15 = 1; B = 1111

ALU bit 15 = 0; B = 1110

This function is used by the Varian 73 microprograms for multiply and divide.

Register field control operations are summarized in tables 10 and 11.

Function	Control Fields S	AB	MR	TS	W
Load A & B fields from control store	00	00	0		
Inhibit loading of A field & place selected 4 bit field (masked) from instruction register into B field	00	01	Masks most significant bit of B field	Selects field	
Inhibit loading of B field & place selected 4 bit field (masked) from instruction register into A field.	00	10	Masks most significant bit of A field	Selects field	
Inhibit loading of A & B fields	00	11			
Inhibit loading of A field & force B field to 1110 if ALU output bit 15 = 0 or to 1111 if ALU bit 15 = 1		00	1		0
Inhibit loading of A field & force B field to 1110 if operand register bit 01 = 0 or to 1111 if operand register bit 01 = 1		00	1		
All functions are inhibited if an I/O request is issued.					

Table 10. Register Field Control

TS Field	Bits selected from instruction register for register file				
000	03	02	01	00	
001	04	03	02	01	
010	05	04	03	02	
011	06	05	04	03	
100	07	06	05	04	
101	08	07	06	05	
110	09	08	07	06	
111	10	09	08	07	

Table 11. Register Field Selection

VARIAN 73 REGISTER USAGE

The Varian 73 standard instruction set makes use of certain general registers. The user microprogrammer is responsible for preserving his results when they are altered and restoring those necessary to their original condition. Registers usage are as follows:

Register 0	Used to emulate the V73 A register.
Register 1	Used to emulate the V73 B register.
Register 2	Used to emulate the V73 X register.
Register 3	Forced to all zeros by the halt microprogram. Used by V73 microprograms as a source of zeros.
Register 4	Used by the halt microprogram to save the contents of the instruction register. Not used by the V73 standard instruction set while running.
Register 5	Forced to all ones by the halt microprogram. Used by V73 microprograms as a source of ones.
Registers E & F	Used by certain V73 instructions as temporary holding registers. Need not be restored as their use does not extend beyond the instructions duration.

All other registers are used by V73 microprograms and must be considered to be temporary usage registers.

MISCELLANEOUS CONTROL

Transfer Instruction Buffer to Instruction Register

The contents of the instruction buffer will be transferred to the instruction register when the T and S fields are both equal to 00 and the 6 field equals XXX1.

PMA Start

A special mode of operation of the PMA may be invoked by setting the T field equal to 00 and the S field equal to 10 with the 6 field equal to XXX1.

Enable Jump Signal

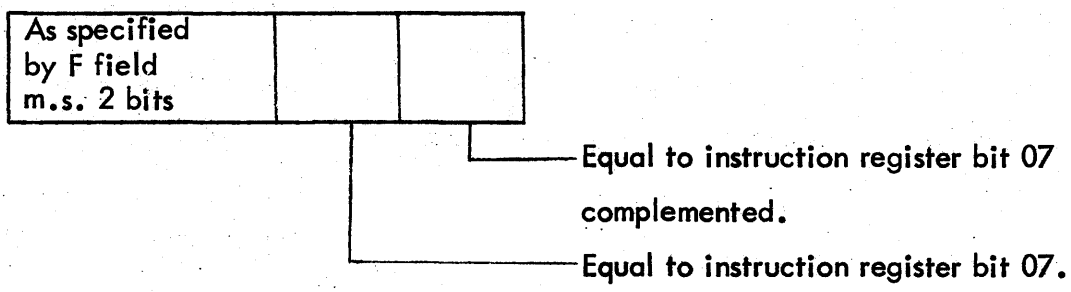
A signal is sent to the memory protect option designating a jump instruction by setting the LB field to 0X and the SC field equal to 0 with the X field equal to 1X. If the X field equals 11, the interrupt flag will be reset also.

Reset Interrupt Flag

The interrupt flag will be reset if the LB field equals 0X and the SC field equals 0 with the Y field equal to X1.

Enable Special ALU Mode

The ALU mode, carry input and overflow sampling may be forced in accordance with the contents of the instruction register by setting the LA and LB fields both equal to 0X with the SH field equal to 1XX. In this case, the ALU function control will be as follows:



The carry input to the ALU will be a one if the instruction register bit 07 is a one or bit 06 is a zero.

DETAILED MICRO INSTRUCTION DEFINITION

General

Figures 2 and 3 define the functions of each micro instruction field. All of these have been functionally described in previous section. The following description lists the functions of each field. Detailed coding is given in the figures.

TS Field

- Provides a component of the address used when a test fails.
- Provides a component of the address used when not 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 micro instructions.
    - Selects interrupts which are to be enabled.
    - Provides a portion of the I/O sequence starting address for I/O requests.
    - Provides the page number for page jump operations.

AF Field

- Provides a contribution to the most significant five bits of control store address for all except decode addressing.

MS Field

- Provides the low order from bits of address for normal addressing.
- Masks the low order four bits of the field select address component.

MT Field

- Masks the most significant bit of the field select address component.
- 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 use in field select addressing.

T Field

- Specifies no testing.
- Specifies testing with pass on condition true.
- Specifies testing with pass on condition false.

S Field

- Defines unconditonal or conditional memory control.

G Field

- Specifies condition to be tested for testiaddressing and conditional memory control.
- Controls status sampling and control of overflow flag.
- Controls transfer of instruction buffer to instruction register.
- Controls transfer of instruction buffer to instruction register.
- Controls selection of decode addressing.
- Controls selection of page jump.
- Controls selection of PMA start.

MR Field

- Masks the most significant bit of the instruction register four bit field selected by the TS field when the AB field so designates.
- Controls, in conjunction with the AB and W fields, the forcing 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 source of data for A and B fields of next micro instruction. May specify:
  - Control store
  - Register field selection from instruction register
  - Previous values
  - Forced 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.
  - Specifies I/O requests
  - Controls interrupt flag
  - Controls supervisor mode
  - Controls special transfer of ALU output to instruction buffer and memory input register.

- Specifies memory operations.
  - Conditional or unconditional start or override of memory cycle
  - Address source for memory cycle
  - Operation to be performed

LB Field

- Specifies source for ALU input B bus from:
  - General registers (specification A field)
  - General registers shifted left (field)
  - General registers shifted right (field)
  - Program counter

R Field

- Specifies destinations for ALU output data to:
  - Program counter
  - Operand register
  - Shift counter
  - Processor key register
- Specifies counter incrementation for:
  - Program counter
  - Shift counter

F Field

- Specifies ALU function.

M Field

- Specifies ALU mode as arithmetic or logical.

C Field

- Specifies ALU carry input as:
    - one
    - zero
    - stored carry
    - stored carry complement
- Forms part of 16 bit literal/mask field.

WR Field

- Specifies writing of ALU output data into general register specified by A field.
- Forms part of 16 bit literal/mask field.

SC Field

- Specifies shifting of operand register.
- Forms part of 16 bit literal/mask field.

V Field

- Controls copying of general register sign to shifted flag.
- Forms part of 16 bit literal/mask field.

W Field

- Controls copying of ALU sign to quotient sign flag.
- Determines direction of operand register shifting.
- In conjunction with AB field determines forcing of B field for next micro instruction.
- Forms part of 16 bit literal/mask field.

X Field

- Determines bit 00 input to operand register for shifts from the following:
  - Operand register bit 15
  - General register bit 15
  - ALU bit 15
  - Zero
- Determines bit 15 input to operand register for right shifts from the following:
  - Operand register bit 00
  - General register bit 00
  - Operand register bit 15
  - Shift flag
- Resets interrupt flag.
- Sends jump signal to external memory protect option.
- Forms part of 16 bit literal/mask field.

SH Field

- Forces ALU input A to all ones or all zeros.
- Forces special ALU mode.
- Determines general register bit 15 input for left shifts from either:
    - General register bit 15
    - General register bit 15
- Determines general register bit 00 input for left shifts from the following:
    - Zero
    - General register bit 15
    - Operand register bit 15
- Determines general register bit 15 input for right shifts from the following:
    - Multiply sign flag
    - General register bit 00
    - General register bit 15
    - Operand register bit 00
    - Zero
- Forms part of the 16 bit literal/mask field.

B Field

- Specifies one of 16 general registers which may be applied to the ALU input B bus.
- Specifies one of 8 special registers which may be applied to the ALU input (B)bus:

- Operand register
- Memory input register
- I/O register
- Processor status word
- Operand register right byte with sign extended
- Operand register left byte with sign extended
- Operand register right byte with sign extended
- Operand register right byte shifted left
- Eight bits

- Forms part of the 16 bit literal/mask field.

A Field

- Specifies one of 16 general registers which may be applied to the ALU input A bus.
- Specifies one of 16 general registers which may receive ALU output data.

IV. USE OF WRITABLE CONTROL STORE

INTRODUCTION

The purpose of the V73 writable control store option is to provide means for expanding the instruction architecture and performance capabilities of the V73 microprogrammed processor.

Incorporated in the WCS are diagnostic capabilities to assist in efficient debugging of micro code. The WCS may be loaded, tested and controlled by a second computer of the Varian 73 or 620 type. The control allows single micro step execution of code, the inspect and changing of micro store locations and branch control using the teletype of the second computer.

The WCS may also be controlled by its host computer. The various stores may be inspected and changed and a single branch to extended store initiated. For diagnostic purposes an optional display and control unit allows single stepping the clock and displays the current micro store address.

FUNCTIONAL DESCRIPTION

General Specification

Central Control Store	Controls all processor activities. Modular in 256 word increments to 512 words by 64 bits on one board.
Decode Control Store (optional)	Allows efficient instruction set changes and new instruction architectures to be formulated.
I/O Control Store (optional)	Permits variation in I/O rates and discipline.
WCS Control	Controlled over I/O bus and by processor page select functions.
Loading	Loaded from either memory bus or input/output. Memory overlapped loading with execution from ROM or a different module.

Return Stack (optional)	Hardware micro subroutine return stack.
Performance	190 nsec. processor cycle time will use 380 nsec. main memory demand rate.
Power	+5V ± 5% @ 20 amps per fully configured module.

I/O COMMAND STRUCTURE

External Commands

EXC 076 Initialize	Program reset allows auxiliary diagnostic computer to reset WCS without having to reset the auxiliary CPU. Deselects all WCS RAM's and terminates any memory load operations in progress. Enables free running mode of the fine clock.
EXC 176 Run Free	Enables free run of the processor fine clock. Clock control commands are generally used with auxiliary diagnostic computer.
EXC 276 Step CCS	Run until next CPU full clock. Fine clock is inhibited following the cycles half clock.
EXC 376 Step I/O	Run until next I/O address register change (full clock). Fine clock is inhibited before next half clock.
EXC 476 Command 76	Sets mode whereby subsequent output 76 transfers are commands to WCS.
EXC 576 Data 76	Clears command 76 mode and subsequent output 76 transfers are set up for data transfer 75 commands.
EXC 676 Memory Load	Initializes main memory to selected control store transfers starting at the main memory location specified in the main memory address register. The control store load starts at the address specified in the WCS address register. The mode is reset when the number stored in the RAM transfer counter is decremented to zero.

Sense Commands

SEN 076 WCS Busy

This function senses that the WCS cannot perform I/O data transfers (transfer device 75) without disturbing a current function. Conditions are:

WCS Busy = (Clock running) (I/O selected module = processor selected module) (I/O selected module is doing a memory load).

Interrupts

Interrupts are: Memory load complete
Stack overflow/underflow
Trace

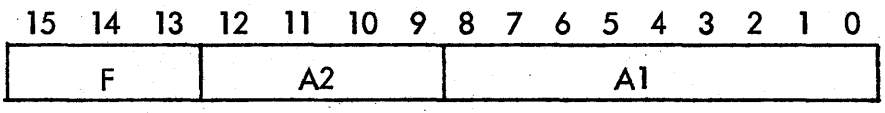
Interrupts may or may not be implemented depending on future design decisions.

Control Output Transfer

OAR, OBR, OME 76

The output function word is a command to the WCS if command 76 mode was established. Otherwise the word is set up information for subsequent data transfer device 75 sequences.

The word format for command mode is as follows:



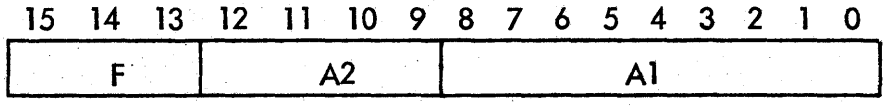
The A1 field specifies up to 9 bits of address in the WCS address register.

The A2 field specifies the page address (which corresponds to the module address for I/O purposes).

The F field specifies the command; command definitions are:

15	14	13	
0	0	0	Branch to central control store location specified by bits 0-12.
0	0	1	Enable decoder control store at the module specified by bits 9-12.
0	1	0	Not used.
0	1	1	Enable I/O RAM at the module address specified by bits 9-12. This enable is buffered until I/O IDLE occurs at which time it becomes active. The address presented to the RAM is for the last word in a programmed output sequence and the RAM contents should be the same as the ROM. Also, this word should be the last word of any sequence that would enable a different RAM or ROM.
1	0	0	Not used.
1	1	1	Not used.

The word format for data mode output 76 transfers is as follows:



The A1 field specifies up to 9 bits of address in the WCS address register.

The A2 field specifies the page address (same as I/O module select address).

The F field has different significance for a subsequent data transfer out to device 75 or a data transfer in from device 75.

Data Transfer Out 75

15	14	13	
0	0	0	Select CCS and load WCS address register.
0	0	1	Select instruction decode A store and load add register.
0	1	0	Select instruction decode B store and load add register.
0	1	1	Select I/O control store and load WCS add register.
1	0	0	Select main memory add register and load WCS add register.
1	0	1	Select and load RAM transfer counter register.
1	1	0	Unassigned.
1	1	1	Unassigned.

Data Transfer In 75

15	14	13	
0	0	0	Same as data transfer out 75.
0	0	1	Same as data transfer out 75.
0	1	0	Same as data transfer out 75.
0	1	1	Same as data transfer out 75.
1	0	0	Select CCS address.
1	0	1	Select pseudo instruction register.
1	1	0	Unassigned.
1	1	1	Select I/O address.

Data Transfers

OAR, OBR, OME 75

Output data to device 75 transfers data to the WCS control store or register previously selected. After each transfer to a control store the WCS address register is incremented. Transfers to and from the 64 bit CCS are always started on a CCS word boundary.

INA, INB, IME 75

Input data from device transfers data from the WCS control store or register to the 620 A register, B register or memory. After each transfer from a control store the WCS address register is incremented.

Control RAM Selection Disciplines

Control RAM's can be uniquely selected using the command 76 mode. Selection of the central control and decoder control stores can also be made under control of the processor page jump micro command.

Whenever a page jump is made, the select status of the decoder stores is set to enable the store associated with that particular WCS module. This change occurs only if the optional decoder RAM is present. If the change is undesired, the micro code of the new environment may be programmed to I/O select the RAM's desired.

The I/O selection is performed by I/O control only. The selection is buffered until the I/O idle state. The address presented to the RAM at this time is the last address of the EXC I/O micro routine, which contains the standard state contents of 0088_H. The just enabled RAM must also have 0088_H at this address, and this address must be the last step in the RAM control sequence which would select other I/O ROM/RAM control stores.

Impact of Memory Control on Processor System Configuration

The memory port of the WCS controls the memory request exactly as a processor does, and therefore, should not be connected to the same port as the processor. The PMA option can then be connected to either port and request the memory bus from either the WCS or the processor depending which memory bus it is connected to.

Physical Description

The WCS is packaged on a standard V73 mainframe printed circuit board (approximately 15.6" x 19"). The maximum component height is a maximum of .4", allowing modules on .6" centers.

Interface Requirements

The WCS interfaces to the main memory and to the I/O option and CPU boards. It also connects to the E-bus cable. The standard connector assignments across the front edge of the card are:

J2, J3	CPU board interconnect
J5	Standard I/O interconnect
J6	I/O option board interconnect

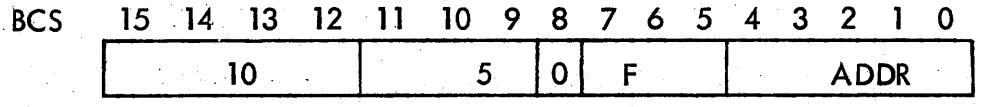
Signal assignments to these pins are given in tables 4-1 through 4-3.

Power Control

The auxiliary power supply for WCS functions the same as for semiconductor main memory. The power will remain on when the V73 console power switch is in the hold, on and console disable positions and is off in the power off positions.

Entry to Writable Control Store

Entry to a microprogram contained in writable control store is by either of two means -- an I/O instruction or a branch to control store (BCS) instruction. Formats for these instructions are given below:



Branch to Control Store

The microprogram control enters the first writable control store module (page number 1) at a word address of:

CEADX

8	7	6	5	4 3 2 1 0
0	0	0	0	ADDR

The F field may contain anything desired by the user.

I/O instructions are OAR, OBR, or OME 76 with a data word of:

15	14	13	12 11 10 9	8 7 6 5 4 3 2 1 0
1	0	0	PAGE	ADDR

The microprogram control enters the selected writable control store module by the page field at the word address designated by the ADDR field.

Exit from Writable Control Store

The writable control store can specify a branch to any address of any page by execution of a page jump as described in the section on Addressing Modes.

V. PREPARATION OF MICROPROGRAMS

General

Microprogram preparation consists of the following sequence:

- Define the operations to be performed.
- Prepare a flowchart of the operation detailing the use of the Varian 73's resources at each step.
- If memory utilization is high, prepare a timing diagram showing the time relationships of micro instructions to memory activity. Insure that memory is being used efficiently.
- Code the micro instructions using pre-defined OP codes.
- Assemble the microprogram using the Varian micro assembler.
- Simulate and trace the operation of the microprogram using the Varian micro simulator to verify correct operation.
- When corrections are completed, load the microprogram into the writable control store using the Varian micro loader/utility program.
- Prepare the macro program which calls the microprogram using the Varian DAS MR assembler.

Microprogram Assembler

The Varian micro assembler, which runs under VORTEX, permits users to code microprograms using instruction mnemonics to define the various fields of the micro instruction. Control store addresses can be referenced symbolically. Constants can be specified in octal or hexadecimal modes. For ease in checkout and documentation, comments can be added between symbolic statements.

The micro assembler has the following special features:

- Use of labels in all addressing modes.

- Ability to define 16 bit literal/mask fields.
- Provides automatic relative addressing.
- Allows user to selectively continue with pass 2 of assembly if pass 1 errors are encountered.
- Allows user-defined MACROS.
- Allows fixed fields in user-defined formats.

Micro Word Specification

The user has the option of specifying in each micro instruction a large number of tasks. The following checklist is useful in planning micro instructions.

Data Path Manipulation

ALU Input B

- General registers (one of 16)
- Special registers
  - Operand register
  - Memory input register
  - Processor status word
  - I/O register
  - Operand register right byte sign extended
  - Operand register left byte sign extended
  - Operand register right byte sign extended
  - Operand register right byte in left byte position
- Masked instruction register*
- 16 bit literal*

ALU Input A

- General registers (one of 16)
- Program counter
- All ones
- All zeros
- General registers shifted left*
  - General register bit 14 to bit 15
  - General register bit 15 to bit 15
  - Zero to bit 00
  - General register bit 15 to bit 00
  - Operand register bit 15 to bit 00
- General registers shifted right*
  - Multiply sign flag to bit 15
  - General register bit 00 to bit 15
  - General register bit 15 to bit 15
  - Operand register bit 00 to bit 15
  - Zero to bit 15

ALU Operation

- Arithmetic or logical functions
- Carry input
  - Zero
  - One
  - Stored carry
  - Stored carry complement

Loading and Counting of Special Registers

- Load from ALU output
  - Operand register
  - Program counter

- Shift counter
- Processor key register

- Count
  - Program counter
  - Shift counter

Writing of General Register with ALU Output Data*

- Operand register shifting*
  - No shift
  - Shift left
    - Operand register bit 15 to bit 00
    - General register bit 15 to bit 00
    - ALU sign complement to bit 00
    - Zero to bit 00
  - Shift right
    - Operand register bit 00 to bit 15
    - General register bit 00 to bit 15
    - Operand register bit 15 to bit 15
    - Shift flag to bit 15

Miscellaneous Data Loop Control*

- Copy general register (A) bit 15 to DSB

*When literal/mask is specified ALU functions are restricted; carry input is zero; operand register shifting is prohibited; general register shifting is limited; miscellaneous data loop control is prohibited.

Control Functions

Addressing

- Normal
- Field selection
  - Field (any 1-5 bit instruction register field)
  - Mask (any of field selected bits)
  - Base address
- Testing
  - Fail address
  - Pass address
    - Normal
    - Field selection
  - Condition tested (one of 16)
  - Pass or fail on condition true
- Interrupt or decode
  - Base address for interrupt
  - Class of interrupts enabled
    - Console step
    - I/O
    - I/O only if memory protect
    - Memory protect
- Page jump
  - Page
  - Word address by field selection

I/O and Memory Control

- Special control
  - No action
  - Wait for memory done
  - Wait for I/O done
  - Special transfer of ALU output to instruction buffer and memory input register

- Set or select and reset interrupt flag
- Load I/O key register
- Set or reset supervisor key
- Inhibit decode
- Request I/O (I/O address must be specified

- Memory operation
  - Instruction fetch
  - Operand fetch
  - Word store
  - Byte store
- Address source
  - ALU output
  - Program counter
  - Memory input register
  - Override previous operation
- Start or override memory cycle
  - Unconditionally
  - Conditional on tested condition true
  - Conditional on tested condition false

Register Field Control

- Loud from control store
- Maintain previous values
- Select from instruction register
  - A or B field
  - 3 or 4 bits
- Force depending on ALU sign or operand register bit 01

Status Control

- Set, reset or sample overflow
- Sample ALU condition codes

Miscellaneous Control

- Transfer instruction buffer to instruction register
- Start PMA
- Set or reset selector/multiplexor flag

I/O and Decoder Control Store Microprogramming

The I/O and decode control stores may not be assembled by the micro assembler. Source data must be prepared for these in loadable format.

Microprogram Simulator

Every product development always includes much time dedicated to the verification that the developed product does indeed interface with and correctly solve the stated problem. Sometimes when this point in time is reached, the developed product does not exactly interface with or correctly solve the stated problem. Many of the inconsistencies could have been uncovered and corrected if a simulation model was constructed. A simulation model or simply simulator can be used in helping to determine a product's performance before its design is fixed and in determining performance/cost trade-offs. The simulator can also be utilized to evaluate future modifications and extensions.

Software Summary

The fundamental program blocks of the simulator are:

a. Simulation control, inputs the simulator commands and directs their execution.
b. Simulator command execution represents the actual execution of the simulator commands.
c. Micro instruction execution, executes a micro instruction by simulating its effect.
d. Simulation information accumulator and list output.

NOTE: The I/O functions of the V73 are not simulated.

Summary of User Controlled Functions

A Alter/display simulator registers.
B Begin simulated execution.
C Change/display control ROM words.
D Dump control ROM to line printer.
E Change/display decode ROM (A/B) words.

H Set control ROM HALT address.

I Initialize program.

M Return to operating system.

R Read ROM data (ROM assembler output).

S Set program for single step/RUN operation.

T Trace output (list or inhibit tracing on LO).

The simulator permits a user developed microprogram to be debugged off-line with full visibility of the effects of each micro instruction registers, status flags and memory bus activity.

This simulator provides the basic facilities for inputting, modifying and outputting the contents of the simulated read only memory, tracing and address halt of the micro instructions.

The microprogram simulator runs under VORTEX. A sample of one step of the TRACE output listing is shown in figure 9.

Initial Condition Selection

After loading, the simulator program is automatically entered and outputs the following to the teletype:

```
Varian 73 simulator
*
```

An * indicates that the program is in the simulator executive awaiting a user command.

All simulator dialogue is entered through the teletype and is interactive between the simulator and the user. The simulator command is recognized by the first character which is entered on the teletype.

Some of the simulator commands are totally defined by its single character identification (i.e. no additional parameters or terminators are necessary). For these simulator commands, the simulator will output a carriage return and line feed when the command is interpreted. This is denoted in the following discussion of the simulator commands, thus:

(C/R)

Some of the simulator commands require, or optionally require, that parameters also be specified in addition to the single character identification. For these simulator commands, the operator must enter the parameters and the carriage return; the simulator will output a line feed. This is denoted in the following discussion of the simulator commands, thus:

(c/r)

All numeric values denoted in the following discussion of the simulator commands are hexadecimal (0-F). Numeric values which are entered by the operator are right-justified with unspecified leading bit positions containing zeros.

Two methods of correcting typographical errors are available to the operator. An entire line can be deleted by typing the control/C character. A line feed and a carriage return are output to indicate that the line has been deleted. A character just entered can be deleted by typing the backarrow character. The backarrow character is printed on the teletype page printer as a visual indicator of the deletion. As many backarrows as necessary can be entered; each deletes one character (but not beyond the beginning of the line).

Each simulator command is checked for syntax errors as the characters are input on a character-by-character basis. When an error is detected by the simulator, a backarrow character is output to the teletype page printer in the character position following the character causing the error. The effect of this backarrow character is the same as if the operator had entered it with the effect as described on this page.

```
ROM LOC = 0090

  TS   AF   MS   MT   FS    T    S    G    M   AB  IMC   LB   LA
  0F   12   00   00   00   02   03   05   00   00   0C   00   00

   R    F   MD    C    W   OS    V    Y    X   TC    B    A
  04   00   00   00   00   00   00   00   01   00   00   00

NEXT ROM ADDRESS =0120

LATCH A 0000

LATCH B 0000

CIN 0

ALU OUTPUT 0000

COUT 1

R0 0000  R1 8000  R2 0000  R3 0000

R4 0000  R5 FFFF  R6 0000  R7 0000

R8 0000  R9 0000  RA 0000  RB 0000

RC 0000  RD 0000  RE 0000  RF 0001

PREG  SREG  OREG  KREG  IOKR  IRG1  IRG2  STUS  IODR   DS    QS
02D5  0000  7FFF  0000  0000  02D7  0201  1800  0000  0001  0001

**** MEMORY OPERATIONS DATA ****

MCCO 1
MOPC 1
MADS 2
MBYC 0
MIL  02D7
IRG1 02D7

ADDR  DATA (MAIN MEM)
02D7  0AC1

TSMX   0000  0000  0000  0000  0000  0001  0000  0000

       0001  0001  0001  0000  0000  0000  0000  0001
```

Figure 9.

Writable Control Store Utility

The writable control store utility runs under VORTEX or as a free-standing program with its own I/O drivers. It permits flexible control of the writable control store to perform:

- Microprogram loading of assembler produced object microprograms.
- Display and alterations of writable control store contents.
- Dump of control store contents to an output device.
- In a diagnostic mode, when controlling another computer's writable control store, the following additional functions can be performed:
  - Clock control (single step, halt, free running.
  - Read out of next control store address.

Macro Definition

Varian 73 programs may be assembled by the DAS MR assembler which include macro instructions which are implemented either as stand instruction set sequences or as micro-programs contained in writable control store.

Through the use of macro definition directives both implementations may be defined. At assembly time, the correct definition may be invoked through the use of conditional assembly directives.

This permits debugging of programs with special instructions using first subroutines containing standard V73 instructions; then, after writable control store microprograms are prepared and debugged, the microprogrammed implementations of the new instructions.

Refer to the Varian Software Handbook for details of the DAS MR assembler usage.

VI. MICROPROGRAMMING EXAMPLE

General

As an example of instruction implementation using Varian 73 microprogramming, the steps of a single word addressing load accumulator (LDA) in the direct address mode will be traced.

SS1M

Initially the instruction pipeline is assumed to be empty so a new instruction must be fetched from main memory. The first micro instruction studied will be that 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 micro instruction fields at 13E are:

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

IM	LB	LA	R	F	M	C	WR	SC	V	W	X	SH
1000	00	00	000	0000	0	00	0	0	0	0	00	000

B	A
0000	0000

The function of this micro instruction 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 micro instruction 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 micro instruction fields at 092 are:

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

IM	LB	LA	R	F	M	C	WR	SC	V	W	X	SH
1000	00	00	100	0000	0	00	0	0	0	0	00	000

B	A
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 an an address. This micro instruction will not be immediately executed as the previous micro instruction initiated memory activity and the memory interface will remain busy until the first instruction from memory is loaded into the instruction buffer and the memory input register. At that time, the current micro instruction completes and the next micro instruction from location 02D becomes active (normal addressing again due to FS, TS, MT and T fields zero. No other fields of the micro instruction 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 copies the instruction buffer into the instruction register to make room for the next instruction from memory.

The micro instruction fields at 02D are:

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

IM	LB	LA	R	F	M	C	WR	SC	V	W	X	SH
0110	00	00	000	0000	0	00	0	0	0	0	00	000

B	A
0000	0000

This micro instruction 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 micro sequence. The T and S fields are both equal to 00 and the G field bit 0 is a one causing transfer of the instruction buffer to the instruction register. The Gfield 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 protect option is installed. The IM field specifies selection of the interrupt flag. If this flag were set, interrupts would be suppressed. The flag is reset by this micro instruction. If an interrupt were active and the interrupt flag had not been set, the next control store address would be 0DX where X designates the four bits supplied by the interrupt logic. This would produce a branch to the interrupt micro sequence.

Assuming no interrupts are present, the new control store address will be determined by the decoder logic. The instruction fetched from memory is assumed to be 10F9 (hexadecimal) or 010371 (ootal). This is a V73 "LDA" instruction with direct addressing of location 00F9 (hexadecimal). The most significant four bits of the instruction buffer address the first decode control store at location 1. The next four bits address the second decode control store at location 00. The decode control store contents are:

1st decode

Control store location 1 $\overline{T32} = 0$ A3 = 110000010

2nd decode

Control store location 0 A2 = 010000000

Since $\overline{T32}$ equals 0, the A3 and A2 address components are logically OR'ed to produce an address of 182.

SWA10

Location 182 contains the first micro instruction 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 micro instruction fields at 182 are:

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

IM	LB	LA	R	F	M	C	WR	SC	V	W	X	SH
0101	10	00	011	1010	1	11	1	1	0	0	00	000

B	A
0000	0000

The LB field equals 10 so the ALU B input bus will have the contents of the instruction register masked by the 16 bits of the M, C, WR, SC, V, W,X, SH and B 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 copied into the operand register.

The S field equals 01 so unconditional memory control is specified by the IM field (0101) to be fetch an operand into the memory input register using the ALU output for an address source. This micro instruction will complete when the memory cycle initiated by the micro instruction at 092 completes.

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 micro instruction of the single word addressing sequence (SWA20). It decodes bits 13-15 of the instruction register contents to determine the class of the single word addressing instruction.

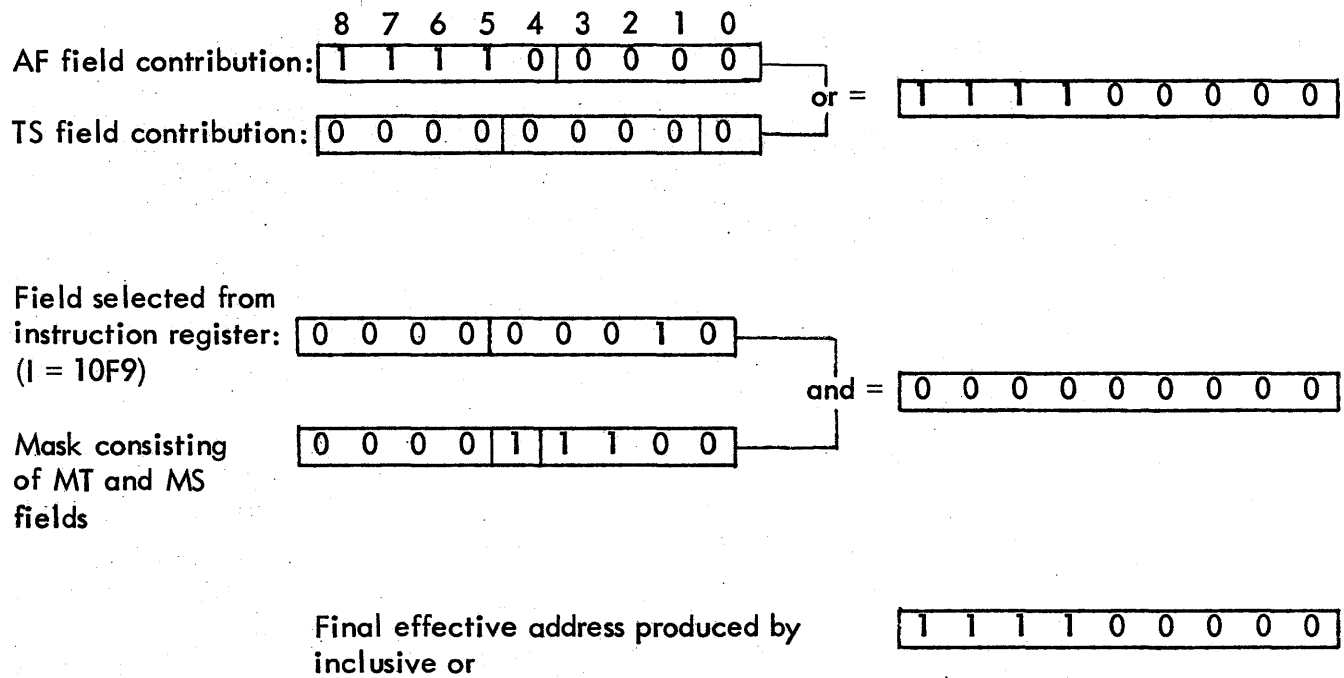
The micro instruction fields at 12F are:

TS	AF	MS	MT	FS	T	S	G	MR	AB
0000	11110	1100	1	1111	00	00	0000	0	00

IM	LB	LA	R	F	M	C	WR	SC	V	W	X	SH
0000	00	00	000	0000	0	00	0	0	0	0	00	000

B	A
0000	0000

No data manipulation or memory control operations are performed by this micro instruction. It serves only to branch to the specific micro sequence for the class of single word addressing instruction contained in the instruction register. Field select addressing is used to perform 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 below:
(see the following page)



The address of the next micro instruction is then IEO.

LDA1

Location IEO is the first micro instruction specific to the "LDA" instruction (LDA1). This micro instruction increments the program counter and initiates another instruction fetch from main memory.

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

IM	LB	LA	R	F	M	C	WR	SC	V	W	X	SH
1000	00	00	100	0000	0	00	0	0	0	0	00	000

B	A
0000	0000

The R field equals 100 specifying that the program counter will be incremented during this micro instruction.

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

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

LDA2

Location 0B5 is the second micro instruction specific to the "LDA" instruction (LDA2). This micro instruction transfers the contents of the memory input register to the accumulator (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 micro instruction 1E0; decodes the instruction buffer to determine the starting address of the next micro sequence and checks for interrupts.

The micro instruction fields at 0B5 are:

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

IM	LB	LA	R	F	M	C	WR	SC	V	W	X	SH
0110	10 (01?)	00	000	1010	1	00	1	0	0	0	00	000

B	A
0001	0000

The ALU B input is specified by the LB field (equal to ~~10~~ 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 (M = 1) with the ALU output equal to the ALU B input (F = 1010.

The WR bit equals a one so the ALU output data will be written into the register specified by the A field (A = 0000) which is the accumulator ("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 copied into the instruction register. The G field bit 2 is a one so the instruction decoder is enabled and interrupts are checked.

The IM field equals 0110 and with the S field equal to 00 the interrupt flag is selected and reset. This will suppress interrupts if the flag is set. All interrupt classes are enabled as the TS field contains all ones. If an interrupt was active and the interrupt flag was off, the decode address would be suppressed and the next micro instruction would be fetched from the address specified by the AF field and the interrupt logic. This would be 0DX where X is the interrupt logic supplied address.

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

Continuation

Figures 10 and 11 show a flowchart and timing diagram of the micro instruction sequence described. Note that the pipeline effect of buffering instructions permits efficient use of the memory. (A 330 ns. semiconductor memory was assumed).

Memory Activity	IF #1	IF #2	OF #1	IF #3	OF #2

Address of Active Micro Instruction	13E	092	02D	182	12F	IE0	0B5	182	12F
Label of Active Micro Instruction	SS1M	SS2M	SS3M	SWA 10	SWA 20	LDA1	LDA2	SWA 10	SWA 20
Memory Activity Specified & Address Source	IF, P	IF, P		OF, ALU		IF, P		OF, ALU	
Program Counter Operation		INC				INC			
Data Path Operation				I 00-10 ↓ ALU ↓ OR			MI ALU R0	I 00-10 ↓ ALU ↓ OR	
Addressing Mode Used	Norm	Norm	Decode	Norm	Field Select I 13-15	Norm	Decode	Norm	Field Select I 13-15
Other Operations			IB ↓ I				IB ↓ I		

Figure 10. Timing Diagram LDA Instruction
Timing Diagram showing start-up and execution of a sequence of single word addressing instructions (330 ns. memory cycle time assumed)

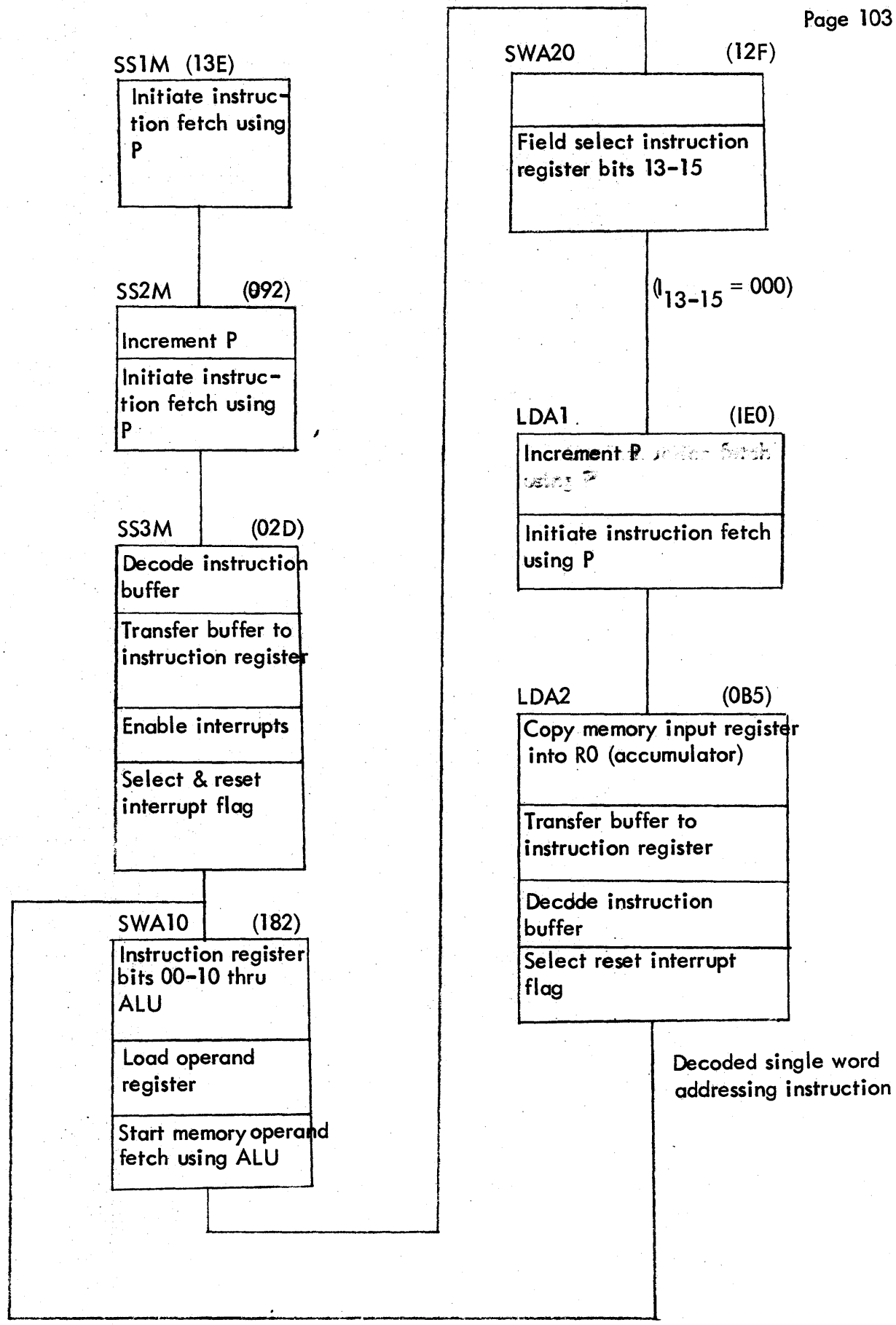


Figure 11. Flowchart LDA Instruction
Flowchart showing start-up and execution of
a sequence of single word addressing instructions