SHEET INDEX AND ISSUE CONTROL

4401CD

CIRCUIT DESCRIPTION OF THE
INSTRUCTION EXECUTION LOGIC/B (IXL/B)
(410401)

REVISIONS

ISSUE	DATE	AUTH. NO.
1	3-18-75	23863R

SHEET NUMBER

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

SHEET ISSUE NUMBER

APPROVALS

PROJ. SUPV.	PROJ. DIR.	MFG.REL. COMPL.
FPR	3-31	bm

ENG.	DSGNR.	DRN.
SA/GH		

DATE	S-NUMBER
2/28/75	

R&D FILE NO.	E-NUMBER
6-167.172AA	4401CD

TELETYPE

TC482-1272

SHEET INDEX AND ISSUE CONTROL

4401CD

CIRCUIT DESCRIPTION OF THE
INSTRUCTION EXECUTION LOGIC/B (IXL/B)
(410401)

Sheet 2

REVISIONS

ISSUE	DATE	AUTH. NO.
1	3-18-75	23863R

SHEET NUMBER

21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

SHEET ISSUE NUMBER

APPROVALS

PROJ. SUPV.	PROJ. DIR.	MFG.REL. COMPL.
		DM

ENG.	DSGNR.	DRN.
SA/GH		

DATE	S-NUMBER
2/28/75	

R&D FILE NO.	E-NUMBER
6-167.172AA	4401CD

TELETYPE

TC482-1272

SHEET INDEX AND ISSUE CONTROL

4401CD

CIRCUIT DESCRIPTION OF THE
INSTRUCTION EXECUTION LOGIC/B (IXL/B)
(410401)

Sheet 3

REVISIONS

ISSUE	DATE	AUTH. NO.
1	3-18-75	23863R

SHEET NUMBER

41	42	43	44	45	46	47	48	49
1	1	1	1	1	1	1	1	1

SHEET ISSUE NUMBER

APPROVALS

PROJ. SUPV.	PROJ. DIR.	MFG.REL. COMPL.
		Bm

ENG. SA/GH	DSGNR.	DRN.
DATE 2/28/75	S-NUMBER	
R&D FILE NO. 6-167.172AA	E-NUMBER	4401CD

TELETYPE

TC482-1272

CIRCUIT DESCRIPTION OF THE INSTRUCTION EXECUTION LOGIC/B (IXL/B) (410401)

TABLE OF CONTENTS

SECTION I - USER INFORMATION

1. General Description

1.1 The Instruction Execution Logic (IXL) requires two card assemblies - the IXL/A and IXL/B inserted into card positions 1 and 2 of the Logic Module. The Instruction Execution Logic provides the logic required to execute a sequence of instructions for the 40C400 Controller.

1.2 The IXL is divided into Data Path Logic and Sequence and Control Logic. The Data Path Logic receives data from the Telebus, processes the data and returns the data to the Telebus. The IXL/A card comprises the logic for the data path. The Sequence and Control Logic contained on the IXL/B card decodes the instruction stream so as to sequence the data path operations. The Instruction Set is composed of Instructions, Figures 1 through 7, and the Operand Addressing modes, Figures 8, 9 and 10. Table 1 lists the abbreviations used to describe the Instruction Set. The Instructions are divided into the following categories:

1.2.1 Double Operand Instructions - The Double Operand Instructions, shown in Figure 1, require both Source and Destination Addressing, shown in Figures 8 and 9.

1.2.2 Single Operand Instructions - The Single Operand Instruction, shown in Figure 2, require only Destination Addressing.

1.2.3 Branch Instructions - Figure 3 lists the Branch Instructions.

1.2.4 Jump and Return Instructions - Figure 4 lists both Jump and Return Instructions.

1.2.5 Data Block Instructions (Control Instruction Group) - Figure 5 lists the Data Block Instructions.

1.2.6 Diagnostic Instructions (Control Instruction Group) - The Diagnostic Instructions are listed in Figure 6.

1.2.7 Status Register Instructions (Control Instruction Group) - Figure 7 lists the instructions that can set or clear the Status Register.

1.3 The IXL/B card executes an instruction by first obtaining the instruction from the Telebus via the IXL/A card. The instruction is stored and decoded into a series of steps called microinstructions. In turn the microinstructions are decoded into control signals by a microprogram stored in an array of Read Only Memories (ROM). The series of microsteps, each approximately 1 microsecond in time, sequences the data through the data path to perform the instruction. When an instruction is completed, the next instruction is fetched.

1.4 The IXL/B also contains logic to command Telebus operations and logic to generate timing pulses to synchronize the operation of the Controller.

2. Features

2.1 The IXL/B card contains the capability to implement a Self-Test feature. This feature is provided by four LEDs and a momentary switch called the Self-Test switch installed on the upper edge of the card. Named Diagnostic LEDs, these LEDs are used to indicate the successful completion of the Self-Test routine, or to indicate the card position code of a card which has failed the Self-Test. Depressing the Self-Test Switch intiates a self-test microinstruction sequence within the IXL. If the IXL functions properly all four Diagnostic LEDs are energized. Releasing the Self-Test Switch continues the testing of the Controller by means of the Self-Test routine contained in the System Memory area.

2.2 In addition to the conventional Instructions, the IXL/B microprogram permits special instructions to be easily incorporated which enhance the performance of 40C400 Controller. These include instructions which speed up Display functions; Rapid Block Transfer (RBT), Store (STO), Search (SRH).

3. Supporting Information

3.1 Refer to 4401SD for schematics of the IXL/B card.

3.2 Refer to 4400CD and 4400SD for a description and the schematics of the IXL/A card.

TABLE 1

LEGEND FOR INSTRUCTION SET DESCRIPTION

LEGEND FOR SYMBOLS USED

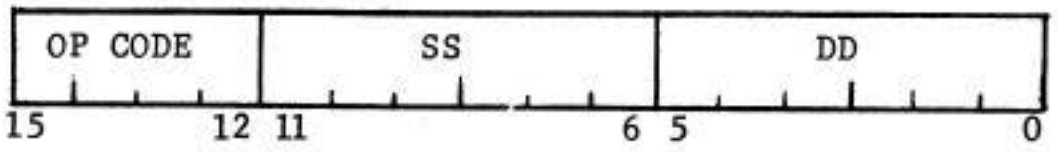
■	= 0 for word/1 for Byte
SS	= Source Field (6 bits)
DD	= Destination Field (6 bits)
JJ	= Jump Address Field (6 bits)
R	= General Register 0 to 7 (3 bits)
XXX	= Offset, +127 to -128 (8 bits)
YY	= Parameter or selected status bits (4 bits)
I	= Index Word
A	= Valid Address
N	= Program Variable
S	= Contents of Source
d	= Contents of Destination
PC	= Contents of Program Counter
SR	= Contents of Status Register
TR	= Contents of Trap Register
IR	= Contents of Interrupt Register
j	= Jump Address
n	= Number of words in the data block operation

CONDITION CODE SYMBOLS USED

*	= Conditionally Set/Cleared
-	= Unaffected
0	= Cleared
1	= Set

EXECUTION TIMES

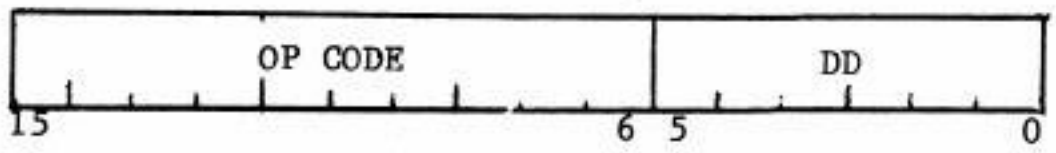
All times are in usec. Add 1.12 usec to source or destination operand addressing when accessing an I/O Device.



MNEMONIC	OCTAL CODE	INSTRUCTION	OPERATION	N Z V C	EXECUTION TIME
MOV(B)	■1SSDD	MOVE	d ← s	* * 0	7.84
CMP(B)	■2SSDD	COMPARE	d - s	* * *	6.72
BIT(B)	■3SSDD	BIT TEST (AND)	d ∧ s	* * 0	6.72
BIC(B)	■4SSDD	BIT CLEAR	d ← d∧($\overline{s}$)	* * 0	7.84
BIS(B)	■5SSDD	BIT SET	d ← d ∨ s	* * 0	7.84
XOR(B)	■6SSDD	EXCLUSIVE OR	d ← d ⊻ s	* * 0	7.84
ADD	07SSDD	ADD	d ← d + s	* * *	7.84
SUB	17SSDD	SUBTRACT	d ← d - s	* * *	7.84

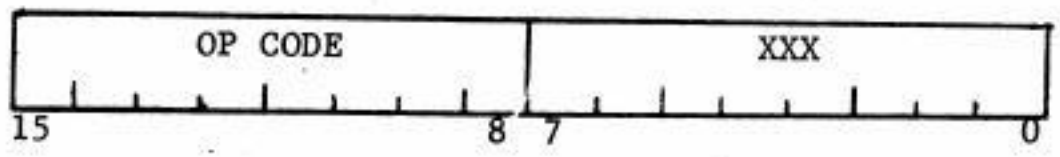
DOUBLE OPERAND INSTRUCTIONS

FIGURE 1



MNEMONIC	OCTAL CODE	INSTRUCTION	OPERATION	N Z C	EXECUTION TIME
CLR(B)	■002DD	CLEAR	d ← o	0 1 0	5.60
COM(B)	■003DD	ONE'S COMPLEMENT	d ← $\overline{d}$	* * 0	5.60
INC(B)	■012DD	INCREMENT	d ← d + 1	* * 0	5.60
DEC(B)	■013DD	DECREMENT	d ← d - 1	* * 0	5.60
TST(B)	■022DD	TEST	d	* * 0	4.48
ASL(B)	■023DD	ARITHMETIC SHIFT LEFT	d ← 2d	* * *	5.60
ASR(B)	■032DD	ARITHMETIC SHIFT RIGHT	d ← d/2	* * *	5.60
ROL(B)	■033DD	ROTATE LEFT	d ← 2d + c	* * *	5.60
ROR(B)	■042DD	ROTATE RIGHT		* * *	5.60
EXB	0043DD	EXCHANGE BYTES		* * 0	5.60
TSR	0052DD	TRANSFER STATUS REGISTER	d ← SR	- - -	5.60
TTR	0053DD	TRANSFER TRAP REGISTER	d ← TR	- - -	6.72

SINGLE OPERAND INSTRUCTIONS
FIGURE 2

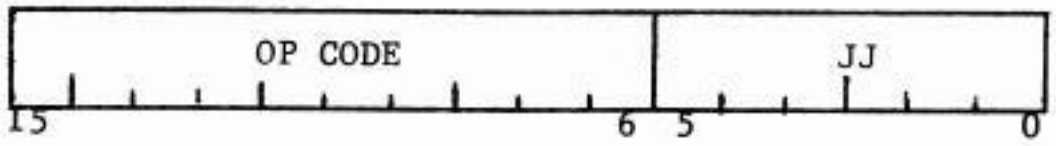


MNEMONIC	OCTAL CODE	INSTRUCTION		CONDITION
NOP	000400+XXX	No Operation		
BR	001400+XXX	Branch Unconditionally	Always	
BNE	002400+XXX	BR If Not Equal	$\neq 0$	Z = 0
BEQ	003400+XXX	BR If Equal	$= 0$	Z = 1
BGE	004400+XXX	BR If Greater Than or Equal	≥ 0	N = 0
BLT	005400+XXX	BR If Less Than	< 0	N = 1
BGT	006400+XXX	BR If Greater Than	> 0	NvZ = 0
BLE	007400+XXX	BR If Less Than or Equal	≤ 0	NvZ = 1
BPL	100400+XXX	BR If Plus	+	N = 0
BMI	101400+XXX	BR If Minus	-	N = 1
BHI	102400+XXX	BR If Higher	$>$	CvZ = 0
BLS	103400+XXX	BR If Lower Than or Same	$\leq$	CvZ = 1
UNUSED	104400+XXX	-		
UNUSED	105400+XXX	-		
BCC(BHS)	106400+XXX	BR If Carry Clear	$\geq$	C = 0
BCS(BLO)	107400+XXX	BR If Carry Set	$<$	C = 1

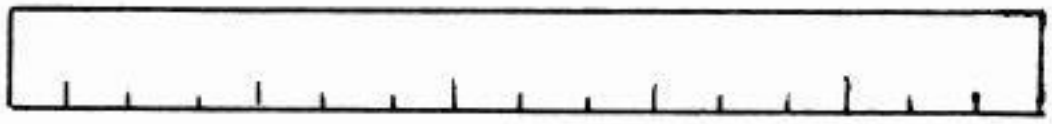
If the branch condition is not satisfied, execution time is 2.24 usec.

If the branch condition is satisfied, branch to location. Execution time is 3.36 usec. PC ← Address of branch instruction +2 + (2·offset).

BRANCH INSTRUCTIONS
FIGURE 3

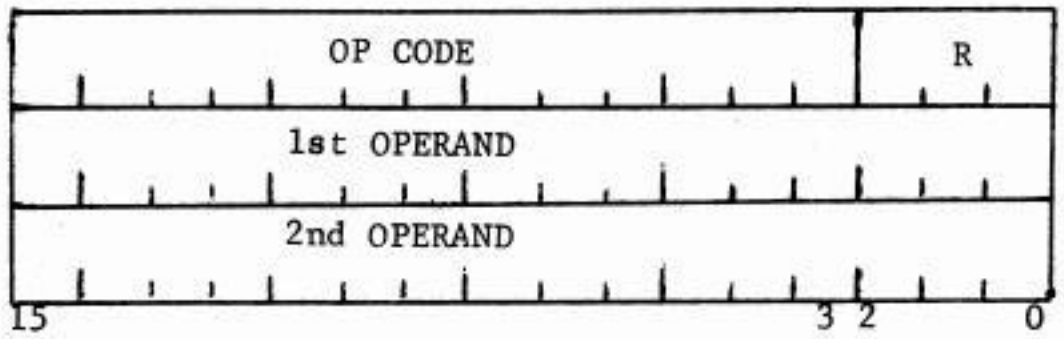


MNEMONIC	OCTAL CODE	INSTRUCTION	OPERATION	EXECUTION TIME
JMP	0072JJ	JUMP	PC ← j	5.60
JSR	0073JJ	JUMP TO SUBROUTINE	SP ← SP - 2 (SP) ← PC PC ← j	10.08



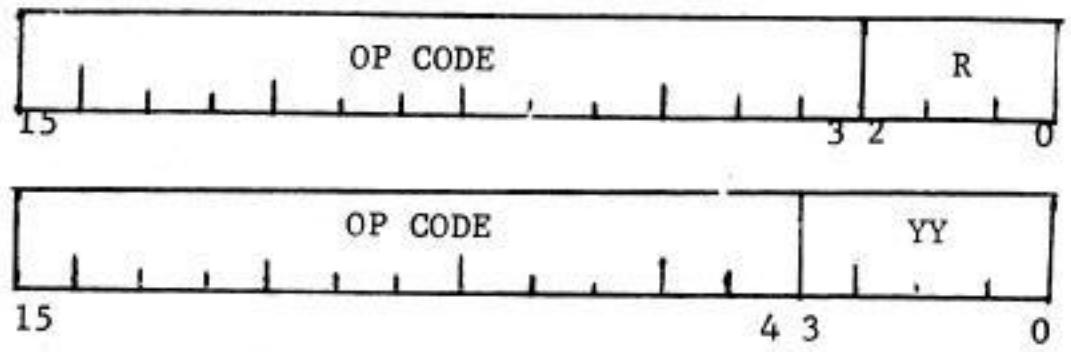
MNEMONIC	OCTAL CODE	INSTRUCTION	OPERATION	EXECUTION TIME
RTS	007267	RETURN FROM SUBROUTINE	PC ← (SP) SP ← SP+2	5.60
RTI	001140	RETURN FROM INTERRUPT	SR ← (SP) SP ← SP + 2 PC ← (SP) SP ← SP + 2	6.72
RFT	000120	RETURN FROM TRAP	PC ← TR	4.48

JUMP AND RETURN INSTRUCTIONS
FIGURE 4



MNEMONIC	OCTAL CODE	INSTRUCTION	OPERATION	EXECUTION TIME
RBT	001000	RAPID BLOCK TRANSFER	Moves a block of data from RAM to the Telebus. The first Operand defines block length. The second Operand defines the block address.	17.92 + 1.12•n
SRH	00202R	SEARCH	Search a block of data. RØ contains block addrs. Rn defines block length. First Operand defines bits tested. Second Operand defines bit sense	11.20 + 3.36n
STR	00402R	STORE	Modifies a block of data. RØ contains block addrs. Rn defines block length. First Operand defines bits cleared. Second Operand defines bit set.	8.96 + 5.60n

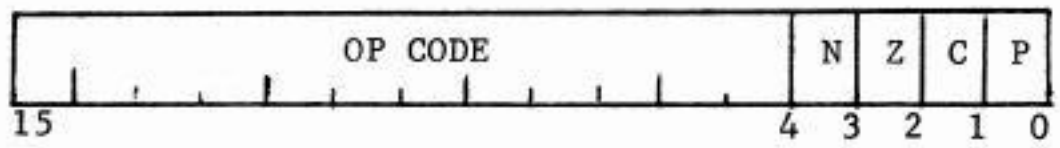
DATA BLOCK INSTRUCTIONS
FIGURE 5



MNEMONIC	OCTAL CODE	INSTRUCTION
SWT	000000+YY	Software Trap
RTT	000060	Return from Trap and Trap
RTC	000160	Return from Trap, Trap and Clear T
LDT	00216R	Load Trap Register
RIV	00502R	Move Interrupt Vector Register to a General Register
-	00504R	Pause (Contents of R decremented until zero each 1.12 usec) (If contents of R is zero, 2×10^{16} cycles required)
-	005060	Issue Grant and Load IR with the Interrupt Vector
DLA	002000+YY	LED Ø
DLB	003000+YY	LED 1
DLC	004000+YY	LED 2
DLD	005000+YY	LED 3

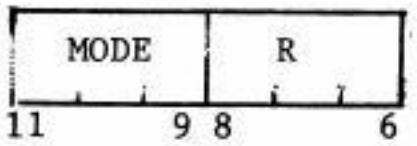
(Led on for period of cycles necessary to decrement 2YY to zero. If YY is zero, 2×10^{16} cycles required)

DIAGNOSTIC INSTRUCTIONS
FIGURE 6



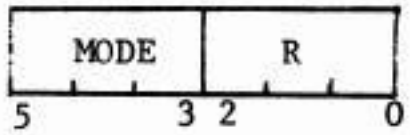
MNEMONIC	OCTAL CODE	INSTRUCTION	EXECUTION TIME
ZER N, Z, C, P	000100+ZZ	Clear the Selected Status Register Bits	4.48
SET N, Z, C, P	000140+ZZ	Set the Selected Status Register Bits	4.48

STATUS REGISTER INSTRUCTIONS
FIGURE 7



MODE	NAME	SYMBOLIC	DESCRIPTION	EXECUTION TIME
0	REGISTER	R	The Operand is contained in General Register R	0.00
1	REGISTER AUTO-DECREMENT	-(R)	Decrement R. The Address of the Operand is contained in R.	1.12
2	ABSOLUTE	A	The Address of the Operand is contained in the second word of the Instruction	1.12
3	IMMEDIATE	#N	The Operand is contained in the second word of the Instruction	0.00
4	REGISTER DEFERRED	(R)	The Address of the Operand is contained in General Register R.	0.00
5	REGISTER AUTO-INCREMENT	(R)+	The Address of the Operand is in R. After the Operand is accessed, R is incremented	0.00
6	RELATIVE	$A	The Address of the Operand is the sum of the second word of the Instruction and PC	3.36
7	INDEXED	±I(R)	The Address of the Operand is the sum of the second word of Instruction and R.	3.36

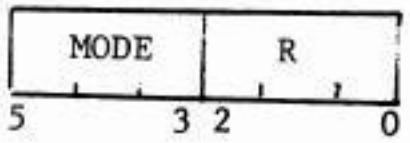
SOURCE OPERAND ADDRESSING
FIGURE 8



MODE	NAME	SYMBOLIC	DESCRIPTION	EXEC. TIME
0	REGISTER	R	The Operand is contained in R.	0.00
1	REGISTER AUTO-DECREMENT	-(R)	Decrement R. The Address of the Operand is in R.	2.24
2	ABSOLUTE	A	The Address of the Operand is contained in the last word of the Instruction	2.24
3	ABSOLUTE DEFERRED	@A	The Address of the Address of the Operand is contained in the last word of the Instruction	3.36
4	REGISTER DEFERRED	(R)	The Address of the Operand is in R.	1.12
5	REGISTER AUTO-INCREMENT	(R)+	The Address of the Operand is in R. After the Operand is accessed, increment R.	1.12
6	RELATIVE	$A	The Address of the Operand is the sum of the last word of the Instruction and the PC	4.48
7	INDEXED	±I(R)	The Address of the Operand is the sum of the second word of the Instruction and R	4.48

DESTINATION OPERAND ADDRESSING

FIGURE 9



MODE	NAME	SYMBOLIC	DESCRIPTION	EXEC. TIME
0	REGISTER DEFERRED	(R)	The Jump Address is contained in R.	0.00
1	REGISTER AUTO-DECREMENT DEFERRED	@-(R)	The contents of R are decremented. The Address of the location which contains the Jump Address is in R.	1.12
2	RELATIVE	$A	The Jump Address is the sum of the second word of the instruction and the PC.	1.12
3	INDEXED DEFERRED	@±I(R)	The sum of the second word of the instrument and R is the address of the location containing the jump address.	3.36
4	ABSOLUTE	A	The Jump Address is the second word of the instruction.	0.00
5	ABSOLUTE DEFERRED	@A	The second word of the instruction is the address of the location containing the Jump Address.	1.12
6	REGISTER AUTO-INCREMENT DEFERRED	@(R)+	The contents or R is the address of the location containing the Jump Address. The register is post incremented.	0.00
7	TEST	-		-

JUMP ADDRESSING

FIGURE 10

SECTION II - DETAILED DESCRIPTION

1. General Introduction

1.1 Purpose of Circuit

1.1.1 The CC4401 IXL/B Circuit Card Assembly contains the logic required to accept instructions and other external inputs and generally a proper sequence of control signals in response to each such instruction or external input. The control sequences, in turn, govern the detailed activity of the Data Path Card (IXL/A) and Telebus to execute the function specified by the instruction or specific external input.

1.2 Microprogram

1.2.1 The Instruction execution logic for any Programmable Controller, like the 40C400, usually consists of two subunits. One of these subunits, the Data Path Logic is required to perform storage, and arithmetic and logical manipulation of all operands and addresses as required by the instruction set of the controller. The other subunit is the Control Logic which interprets the instruction and issues a series of commands to the various logic elements in the data path (such as the registers, arithmetic logic or data switches or bus) in response to each instruction in the operating program. The 40C400 IXL/B control unit uses a technique called "microprogramming" to implement the proper sequencing of these controls. All control commands to the Data Path and Telebus originating from the Control Unit during a given IXL clock period constitute a control word or "micro-instruction". A sequence or collection of microinstructions is called a "microprogram" or microsubroutine and constitutes the actual step by step control of operations within the IXL.

1.3 Control Word

1.3.1 The microinstruction control word for the 40C400 is divided into a number of control fields each designed to control a specific logic element of the IXL as shown in Figure 11. A detailed account of these fields and their coding will be presented in subsequent sections.

1.4 Control Storage

1.4.1 The entire microprogram of the 40C400 consists of 256 control words stored in ROM arranged in groups or "microroutines". Each routine has a name and performs a specific function in the course of instruction execution as follows:

1.4.2 The FETCH microroutine accesses the next macro instruction and services the overhead functions for interrupts, traps or self tests.

1.4.3 The SOURCE microroutine accesses the first operand for all double operand instruction. The operand can be addressed in any one of 8 different addressing modes. See Figure 9.

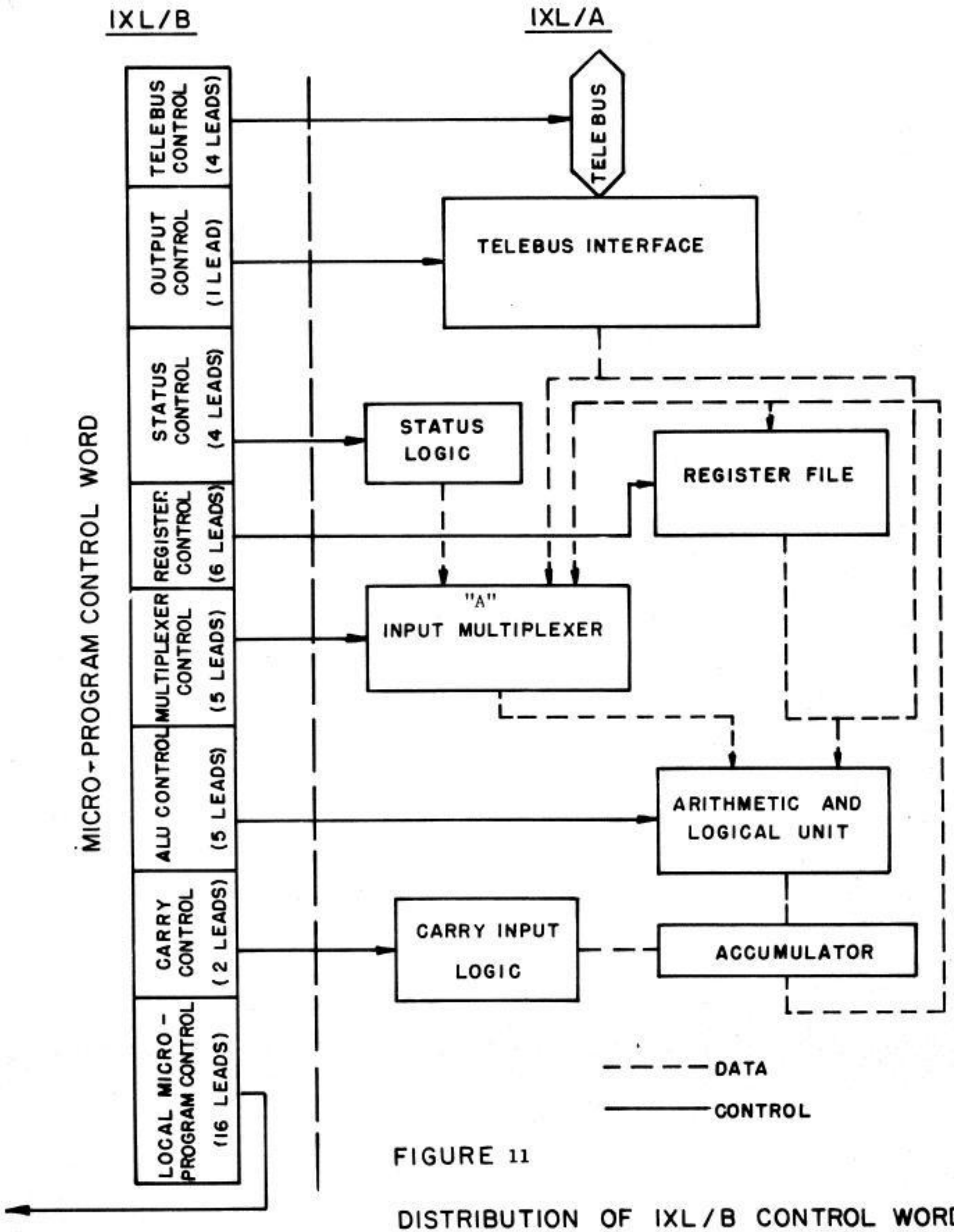


FIGURE 11

DISTRIBUTION OF IXL/B CONTROL WORD

1.4.4 The DESTINATION microroutine accesses the second operand for all double operand instructions or the only operand for single operand instructions. It also accesses the jump address for all jump instructions. Again, the operand can be accessed in any one of 8 different addressing modes. See Figure 10.

1.4.5 The SOURCE II and DST II microroutines are an extension of the SRC and DST routine which sign-extend and swap byte operands for subsequent manipulation by the ALU.

1.4.6 The EXECUTE microroutine controls the actual arithmetic or logical operations performed on the previously accessed operands.

1.4.7 The CONTROL microroutines perform the data manipulations required to execute the Control Type instructions such as RBT, SRCH and STORE.

1.4.8 The BRANCH microroutine performs the arithmetic calculations required to perform the branching instructions.

1.5 Major State Transition

1.5.1 For correct execution of a macro-instruction, the Control Unit must select one or more microroutines to be executed in sequence. The proper sequencing of these microroutines for any given macro-instruction is shown in Figure 12 in terms of a major state transition diagram. When the machine is executing a particular microroutine (such as FETCH) it is said to be in a major state. At any given time, the selection of the next major state, or microroutine depends on the present major state and the type of macro instruction being executed. All such conditions and their corresponding microprogram state transitions are shown in Figure 12.

1.6 Microprogram Next State Address Encoding

1.6.1 Since each major state, or microroutine of the machine corresponds to the execution of a distinct block of control words in the 256 word ROM microprogram, a starting address for that particular section of ROM must be generated for each transition to its corresponding major state. It should be noted, however, that some major state microroutines divide into a number of smaller sub-blocks. Each sub-block in the routine performs a different variation of the required basic microroutine function. For example, the SOURCE and DESTINATION routines are actually divided into 8 sub-groups each; each performing the accessing of the SOURCE or DESTINATION operand in one of the 8 possible addressing modes. Thus, in addition to selecting the correct microroutine, the starting address generated upon major state transition must select the proper segment within the routine.

1.6.2 An overview of the 256 word microprogram layout is shown in Figure 13. For a detailed account of the step by step microprogram operation the reader is referred to a complete microprogram listing in a later section.

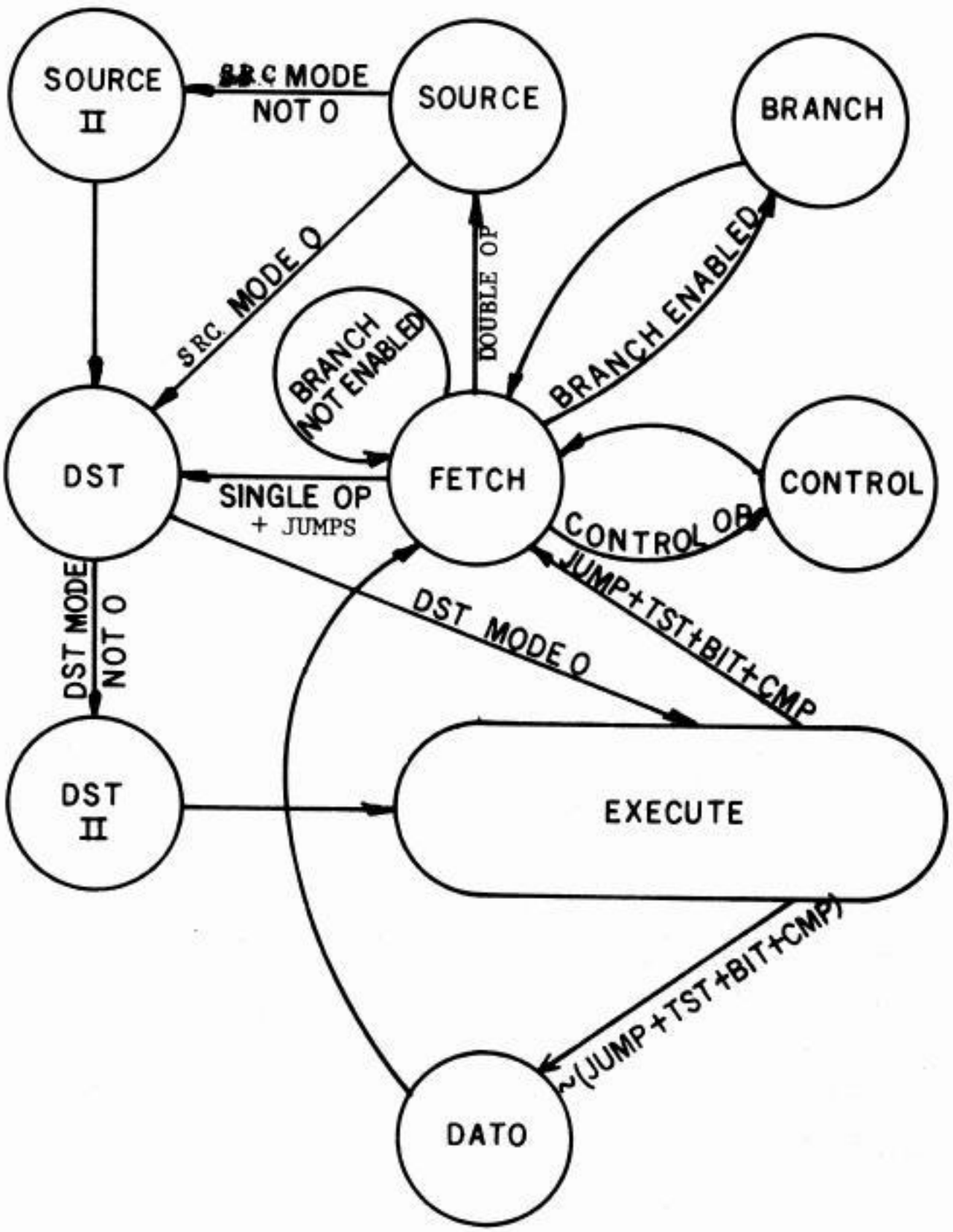


FIGURE 12 MAJOR STATE TRANSITION DIAGRAM

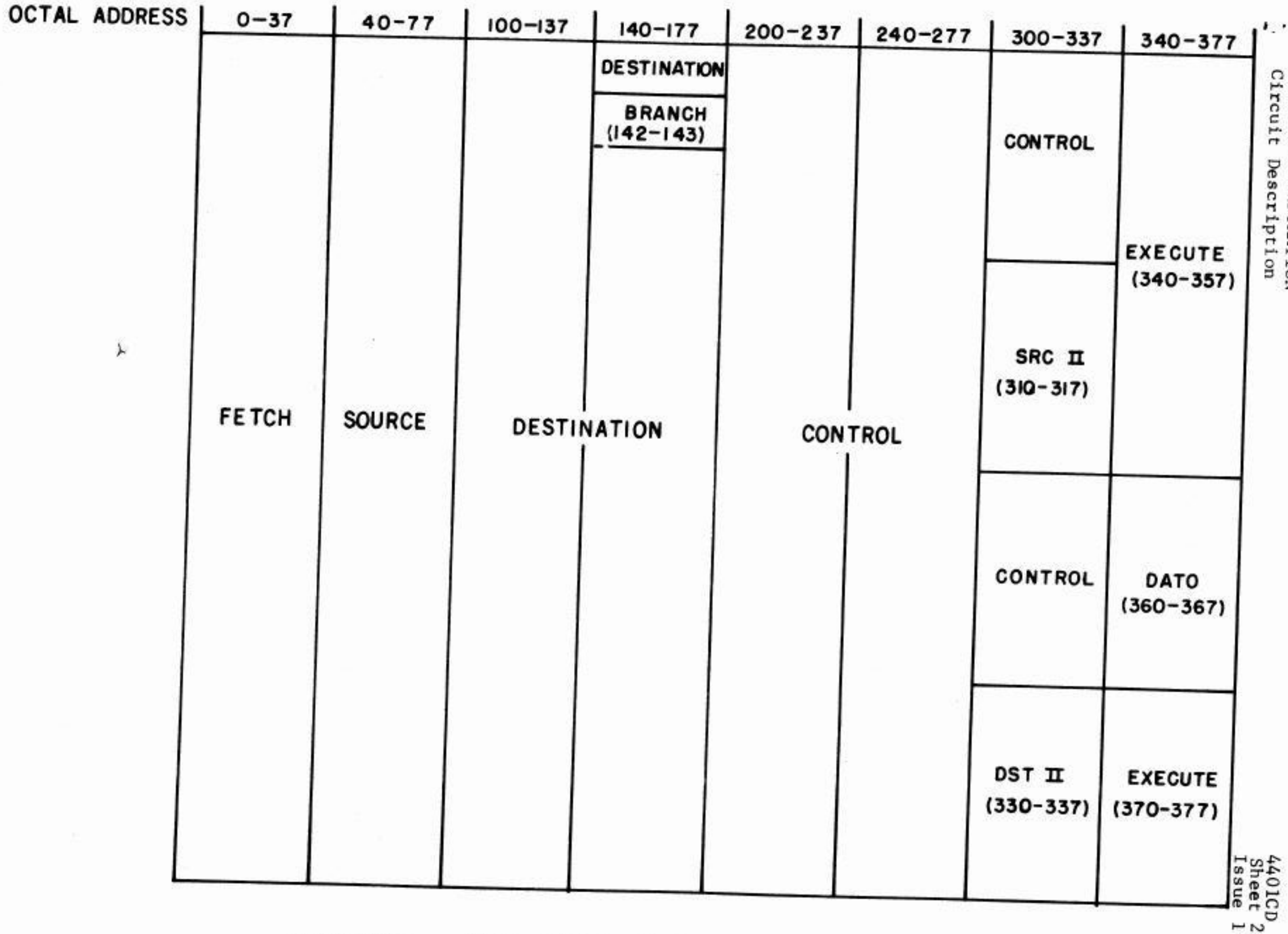


FIGURE 13 256 WORD MICRO-PROGRAM LAYOUT

2. Voltage and Logic Standards

2.1 The 410401 Card Assembly requires +5 Volts and +5 Volts Standby from the power supply. The voltage drop of the power busses on the 410401 Card is held to less than 2%. The power supply system should provide +5 Volts and +5 Volts Standby with sufficient tolerance to include the 2% due to card voltage drop and still provide +5 Volts $\pm$ 10% at the supply leads of all integrated circuit packages.

2.2 The following convention is used in the circuit description to give a unique designation to each circuit in an integrated circuit package.

MLB8 -- (4)
ML refers to micrologic
B8 refers to the package at board location B8
(4) refers to the pin number of the output of the circuit.

2.3 The terms "HIGH" and "LOW" are used in this description to indicate logic voltage levels; where "HIGH" represents any voltage between 2.4 and 5.5 volts and "LOW" represents any voltage between 0.0 and 0.4 volts.

2.4 Signal leads are labeled with a title description of its function. When they are superscribed with a bag (e.g., $\overline{\text{INTR}}$) it indicates that the signal is in the LOW state when the state described by the signal is present; otherwise it is in the HIGH state.

3. Block Diagram

3.1 (Refer to 4401SD Sheet 2)

The schematic documentation of the 410401 Circuit Card Assembly (4401SD) is broken up into a number of Functional Schematics (FS1-FS8). The operation of each functional block will be described in this section. The block diagram (4401SD Sheet 2) shows the logical relationship of the functional blocks to each other.

4. Circuit Card Operation

4.1 FS-1 - INSTRUCTION REGISTER (Refer to 4401SD Sheet 3). All macro-instructions which are executed by the IXL are first brought into and stored in a 16 bit register called the Instruction Register (MLC11, 12, 13). The instructions originate from memory and are transmitted via the Telebus and received by bus transceivers located on the IXL/A circuit card. From there, they are transmitted on the back (B) connector to the instruction register on the IXL/B card. The proper signal for strobing of the instruction is generated by gate MLB14-(12). Strobing of the instruction occurs when the FB1 and FB2 leads, part of the microprogram control word, are LOW and the ACC CLOCK lead, originating in the Timing Logic (FS-8) goes LOW.

4.2 FS-2 - INSTRUCTION DECODE AND MAJOR STATE CONTROL LOGIC (Refer to 4401SD Sheets 3 and 4).

4.2.1 The instruction decode ROM (MLB11) and outputs 2 and 3 of the branch ROM (MLC9) are used to decode certain attributes of the instruction currently being executed. These attributes are required by the address encoding logic and affect the microroutine starting address as follows:

4.2.1.1 BYTE - The instruction is a byte operation. This attribute affects starting address encoding in the SOURCE, SOURCE II, DESTINATION, DST II and DATO routines.

4.2.1.2 SUBTRACT - The instruction is a subtract instruction. This condition affects address encoding in the DOUBLE OP routine.

4.2.1.3 BYTE $\wedge$ $\overline{(\text{MOVB} \wedge \text{DMO})}$ - The instruction is a byte operation; but not a MOV destination mode 0. This attribute is needed to control the starting address in the DATO routine.

4.2.1.4 DMO - The instruction has a destination mode of 0. This condition affects the address encoding in the DATO routine.

4.2.1.5 JUMP - The instruction is either a JMP or JSR instruction. This attribute affects address encoding in the DESTINATION state.

4.2.1.6 ROR - The instruction is a ROR. Starting address of the SRC II and DST II routine is affected.

The Boolean equation for decoding these attributes is also shown on FS2. The detailed relationship between the above attributes and the microroutine starting addresses is described in the explanation of the M.P. address encoding logic (FS-3).

4.2.2 The Branch ROM (MLB11) in addition to decoding instruction attributes, compares the instruction bits which identify the branch type (IR9-IR11 and IR15) with the condition codes C, Z and N originating in the Status Register on IXL/A. Whenever the value of the condition codes satisfies the condition imposed by the branch type, a HIGH level is generated on the BR ENABLE output MLC9-(12). This output, in turn, is fed to the next state ROM (MLC10) and affects a state transition to the BRANCH routine if the instruction is a branch type.

4.2.3 The Next State ROM (MLC10) receives two groups of information. The first group (PSØ, PS1 and PS2) contains a code which conveys the present major state of the control unit. This code is part of the microprogram control word and is generated by the micro-control ROM MLA11 (10-12) on FS4 as a direct result of the microprogram address. The second group (BR ENABLE, IR7, IR8, DBLOP and ROR) conveys information necessary to decode the type of instruction currently being executed. The Next State ROM combines this information and generates a coded output on MLC10 (9-12) which signifies the next microroutine to be executed. This next state code (NSØ-NS3) is sent to the microprogram address encoding logic (FS3) and selects the proper address encoding for each

4.2.3 Continued

particular microroutine. 4401SD Sheet 5, Table 1, shows a detailed major state transition table, which describes, in symbolic and binary form, the coding of the MLC10 Next State ROM. The first part of the table indicates a microprogram address field and the corresponding microroutine or major state, located in the field. The present major state code (PSØ-PS2), generated from the address field by the control ROM MLA11, is shown alongside the name of the present state. For each present state, the table shows one or more transitions to a next state, whose name and code (NSØ-NS3) are shown at the right. Where there is more than one possible transition, as in FETCH, the particular next state depends on the instruction type currently being executed. The information necessary to decode instruction type is listed along the name of the type. Note that in some cases, more than one present major state is assigned to a particular coding of PSØ-PS2, for example in BRANCH, CONTROL and DATO. This is done to minimize the code since, in each such instance, the next state is the same. Also, the next state code is assigned in such a manner as to simplify the address encoding logic shown later. It can be seen that the transition table is simply a more detailed version of the transition diagram of Figure 12.

4.2.4 The gate MLB14-(6) decodes the $\overline{\text{DBLOP}}$ attribute which is required by the next state ROM and by the branch ROM to decode the ROR and JMP attributes.

4.3 FS-3 Micro-Program Next State Address Encoding Logic (Refer to 4401SD Sheet 4)

4.3.1 The next state code (NSØ-NS3) generated by the MLC10 ROM controls the selection of microroutine starting addresses as follows. Each bit of the starting address for a given microroutine forms an input to a 1 of 8 multiplexer. Since 8 bits of address must be generated to produce an entire next microroutine address, 8 channels of multiplexing are required to generate the whole address set. For each particular next state code, a set of 8 inputs is chosen to be gated through to the next state microprogram address bus (NS ADDRSØ-NS ADDRS7). Each microroutine starting address, generated in this manner, is not a fixed value but is dependent on certain attributes from the instruction itself. The binary value of these attributes cause a modification of the entry point in a given microroutine. For example, the starting address for the SOURCE routine is selected by NS code = 5 (octal) which gates the "D5" inputs of multiplexers MLB6 through MLB9 and MLC6-C7 to the "NS ADDRS" bus. A number of attributes modify the SOURCE routine starting address as follows:

a) BYTE on address bit 1 (NS ADDRS1)
b) IR9 on address bit 2 (NS ADDRS2)
c) IR10 on address bit 3 (NS ADDRS3)
d) IR11 on address bit 4 (NS ADDRS4)

Thus there are really 16 different sub-block starting addresses, or encoding combinations, for this particular routine. A pair of different microprogram sub-blocks, in this case, controls the execution of one of the eight source operand addressing modes, one sub-block of each pair in the word mode and one in the byte mode, as seen in the detailed microprogram listing. Note that the attributes IR9-IR11 correspond directly with that part of the instruction word which specifies the source addressing mode.

4.3.2 Table 2 of 4401SD Sheet 5 shows the address encoding selection for each of the nine major states or microroutines of the Control Unit. The SRC II and DST II state encoding has been combined to form a single input into the 1 of 8 address selector multiplexers. The starting addresses for these two microroutines differ only in bit position 4 (SRC II bit address bit 4 is LOW, and DST II address bit 4 is HIGH). Since the Present State code lead PS1 is LOW in the SOURCE state and HIGH in the DESTINATION state, this lead is used as an attribute in address bit position 4. It should be noted that the encoding for address bits 6 and 7 (NS ADDRS6 and NS ADDRS7) is done by dual 4-1 multiplexer MLC8 since the upper address bits have the same value for more than 1 microroutine. For example the FETCH and SOURCE routines both have address bits 6 and 7 LOW; the DESTINATION and BRANCH have address bit 6 HIGH and address bit 7 LOW, etc. The microroutines are subdivided into four such groups, and thus it only becomes necessary to select the group, instead of the exact routine. The next state code bits NS2 and NS3 are assigned so that they select such a group of these routines.

4.3.3 Multiplexer MLB10 is used to select 1 of 2 sets of attributes and combines them into 1 set for use by the EXECUTE microroutine. Since the "routine address determining" bits for a Double-Operand Instruction are IR12, IR13 and IR14 and for a Single Operand instruction are IR6 and IR9-11, MLB10 selects between the two sets. The resulting outputs are called EXECØ through EXEC3 and are gated to the microprogram address bits Ø, 1, 2 and 4, upon entry into the EXECUTE microroutine. The Subtract instruction disables the outputs and causes LOWs to be applied to the particular microroutine address positions. Both the "DBL OP" and "SUBTRACT" leads originate in the Instruction decode logic (FS-2).

4.3.4 Gates MLB15 and MLB16 generate the attributes required for handling interrupts, traps and self tests in the FETCH state, since each of these conditions require the execution of a different FETCH microprogram sub-block. For example the "INTR" condition, along with LOWs in the "T" and "P" status register bits, will generate a HIGH on microprogram address bit 2, if there is no "TRAP" or self TEST pending; causing an entry into the FETCH microroutine at octal address 4. Normal entry into FETCH is made at octal address 2 since the absence of all external conditions causes a HIGH level on the MLB15-(13) gate, ($\overline{TEST}$ $\overline{TRAP}$ $\overline{INTR}$ $\overline{T}$ $\overline{P}$), which is gated, via the MLB7 multiplexer, into microprogram address bit 1. Similarly the self TEST and TRAP conditions cause an entry at octal address 20 and 10 respectively, due to HIGH level generated on the $\overline{TEST}$ or TRAP*$\overline{TEST}$ leads which are gated into levels 4 and 3 of the microprogram address prior to the execution of the FETCH microroutine.

4.3.5 The exact address generated for each specific entry into a microroutine can be read from the Address Encoding Table of 4401SD Sheet 5. By imposing, upon each address bit, the attribute specified in the table, the total address is derived. If the specific attribute is a " H " or " L ", the corresponding address bit is fixed at entry into that microroutine. If the attribute is an Instruction Register bit, such as IR9, the address is dependent upon the value of that bit when the microroutine is executed. If the attribute is a more complex boolean expression, it must be evaluated to establish the microroutine next state address.

4.4 FS-4 Micro-Program Addressing Logic and Micro-Storage Array No. 1

4.4.1 Microprogram Addressing Logic

4.4.1.1 Two 4 bit presettable counters, MLA6 and MLA7, make up an 8 bit micro-program address register. The outputs of this register drive the address lines of the entire 256 word ROM microprogram storage array. The lower 4 bits of the counter receives 2 sources of preset data via the MLA5 quad 2-1 multiplexer. One source comes from the next state address encoding logic on FS3, labeled NS ADDRSØ through NS ADDRS3, and the other source comes from the ROM microprogram control word and is labeled MPBR ADDRSØ through MPBR ADDRS3. The upper 4 bits of address can be preset with encoding logic outputs NS ADDR4 thru NS ADDR7.

4.4.1.2 Whenever transfer to a new microroutine is to be executed, the address register is preloaded with the routine starting address on the NSADDR leads generated by the address encoding logic. Within a given microroutine, however, sequential addresses are generated on the MPBR ADDRS leads by the micro-ROM itself, and are loaded into the lower 4 bits of the micro-program address register one after another at each IXL cycle as the microroutine proceeds until a new next state starting address is required. This activity is controlled by the leads labeled SEQØ and SEQ1, which are also part of the microprogram ROM outputs. Thus when SEQØ and 1 are both HIGH, a load of 8 bits from the NS ADDRS leads occurs because MLC17-(1) and MLC17-(4) are both LOW, and when SEQØ is LOW and SEQ1 is HIGH a load of the lower 4 bits from the MPBR ADDRS ROM outputs occurs because MLC17-(1) is LOW and MLC17-(4) is HIGH.

4.4.1.3 Conditional Loading - Whenever lead SEQ1 is LOW the MLA6 preset is conditioned by the CNT EN lead. This lead originates on the IXL/A card and indicates the success or failure of a microprogram level test. The test indicates that the result of the ALU operation currently being executed is either zero or non-zero. The SEQØ lead is sent to the IXL/A card and selects a zero test when in a LOW state and a non-zero test when in a HIGH state. When the particular test fails, the IXL/A sends a LOW level on the CNT EN lead. This condition, together with SEQ1 lead being LOW, causes the address register to index (count) rather than preset to a next address. Using this technique the microprogram can do conditional branching within a microroutine. This feature is frequently employed within the microroutines for the SEARCH, STORE and other instructions in the CONTROL group. The address register is clocked every IXL cycle with the MP CLOCK lead which originates in the Timing Logic (FS-7). A HIGH condition on the PWR RESET lead going to MLA6 and MLA7 causes these registers to be reset to all LOWs. This condition will initiate micro-program execution at address 0, which corresponds to the power up microroutine. (See microprogram listing.)

4.4.2 Micro-Storage Array No. 1

4.4.2.1 The succession of microprogram addresses, from the address register MLA6-MLA7, feeds the ROM microprogram arrays No. 1 and 2 which, in turn, generate the 44 bit micro-control words required to control the IXL and TELEBUS. Micro-ROMs MLA9 thru MLA13 generate the following control fields:

4.4.2.1 Continued)

a) Control ROM MLA9 generates the MPBR ADDRESS field. This 4 bit field specifies the low-order part of the next microprogram address location to be executed on the next IXL clock cycle.

b) Control ROM MLA10 generates a 4 bit BUS CONTROL field. The leads MPCØ-MPC2 specify the bus operation (e.g., load-address and read) to be executed during the upcoming TELEBUS cycle. The lead MPINTG specifies that the interrupt grant (INTG) lead of the TELEBUS is to be activated.

c) Control ROM MLA1Ø -(10-12) generate the 3 bit present state code (PSØ-PS2) which converts the microprogram address fields, corresponding to a specific microroutine, into the codes specified by the NEXT State Table (4401SD Sheet 5). Control ROM MLA1Ø-(9) generates a single "OUT ENABLE" which controls the IXL/A TELEBUS output gating.

d) Control ROM MLA12 generates a 4 bit STATUS CONTROL code field. These leads control the loading, setting or clearing of the N, C, Z, P and T status bits residing in a register on the IXL/A card.

e) Control ROM MLA13-(9, 10) generate a 2 bit ALU carry input control code. This code specifies the value of the ALU input carry and the value of the sign extension during the upcoming IXL cycle. Control ROM MLA13-(11,12) generates the two sequence control leads (SEQØ and SEQ1) required to control the cycle by cycle activity of the microprogram address register, described in 4.4.1.

4.4.2.2 Detailed listing of the above control codes and their specific functions are found in the tables accompanying the 4401SD Sheet 6. An explanation of the functions controlling the IXL/A card can be found in the 4400CD.

4.5 FS-5 Micro-Program Storage Array No. 2 and General Purpose Register Addressing Logic.

4.5.1 M.P. Storage Array No. 2

4.5.1.1 This part of the microprogram storage consists of MLA3 and MLA14 through MLA18 and generates the following control fields:

a) Control ROM MLA8 generates the 4 LED control leads LEDØ through LED3. These leads, when programmed in a HIGH state, light their corresponding LED during the upcoming IXL cycle. The feature is used for the IXL self test operation and by program execution of the LED macro-instructions.

b) Control ROM MLA14-(9-11) generates a 3 bit code (FBØ-FB2) which controls the loading of the macro-instruction (see 4.1), clearing and setting of the TRAP condition and sampling or clearing of the Odd byte condition as described in detail in Section 4.6.

c) Control ROM MLA14(12) and MLA15 provide the ALU function code (MPSØ-MPS3 and M) to the IXL/A card. The function code specifies the logical or arithmetic operation to be performed by the ALU during the upcoming IXL

4.5.1.1 Continued

c) continued

cycle. Whenever the "M" lead is LOW, the operation is of an arithmetic nature, whereas when "M" is HIGH the operation is of a logical type.

d) Control ROM MLA16 and MLA17 (9-11) generate the IXL/A Register File Control field. The leads MP REG SELØ through MP REG SEL4 select the specific register of the file to be accessed during the upcoming IXL cycle. The leads MP WUB and MP WLB specify that the upper or lower byte (or both), of the selected register are to be loaded on the upcoming cycle. These control leads are routed to the Register File addressing logic described in Section 4.5.2.

e) The Control ROMs MLA17-(12) and MLA18 generate a 5 bit IXL/B input multiplexer control code MUX SELØ through MUX SEL4. This code selects a specific operand or value to be selected for presentation to the IXL/A ALU A operand bus during the upcoming IXL cycle. The choices of operands include the TELEBUS input, the Accumulator, and the Status Register. Additional multiplexed operand sources are the constant values +1 and -2, sign extended operands and byte exchanged operands from the above sources.

4.5.1.2 A detailed listing of the functions of the above control codes are to be found in the accompanying tables on 4401SD Sheet 8. An explanation of the functions controlling the IXL/A card can be found in the 4400CD.

4.5.2 Register File Addressing Logic - Register file addresses, sent to the IXL/A circuit card, originate from three sources in the IXL/B control logic.

Whenever a double operand instruction is being executed, instruction register bits IR6, IR7 and IR8 can specify the 1 of 8 general purpose registers to be used for the source operand or the source operand address. Whenever a double or single operand instruction or control instruction is being executed, instruction register bits IRØ, IR1 and IR2 can specify 1 of 8 general purpose registers to be used for the destination operand or destination operand address.

During the course of execution of a microroutine, the microprogram, itself, may specify the implied use of any of the 8 general purpose registers in a given instruction, such as the use of Register 0 for the block starting address of a SEARCH or STORE instruction. Additionally, the microprogram may address any of the remaining 8 special purpose registers of the IXL/A register file. These registers are used to handle various overhead functions as follows:

a) PC (register 8) used as a macro-instruction program counter.

b) SRC (register 9) used to temporarily store the SOURCE operand until execution of arithmetic or logical operation.

c) TEMP (register 10) used as a temporary register for storage of one of the mask operands during the execution of the SEARCH and STORE instruction.

4.5.2 Continued

d) X (register 11) used as a temporary register for storage of intermediate results during the execution of the SEARCH and STORE instructions.

e) IV (register 14) is used to store the last interrupt vector generated by a device on the TELEBUS. This feature is a diagnostic aid.

f) TRAP (register 15) is used to save the value of the program counter when a TRAP condition is serviced. The PC is restored when a RETURN FROM TRAP instruction is executed.

The three-way register selection, described above, is done by the MLB12 and MLB13 4 into 1 multiplexers. The MP REG SEL4 lead, from the microprogram control word, selects "instruction selected" registers when in a LOW state and implied microprogram registers when in a HIGH state. When "instruction selected" registers are addressed the MP REG SELØ lead, from the control word, selects the SOURCE instruction register field when in a LOW state and the DESTINATION instruction register field when in a HIGH state.

When implied microprogram registers are addressed, the leads MP REG SELØ through MP REG SEL3 contain the binary address of the registers to be accessed during the upcoming IXL cycle. The 4 MLB12 and MLB13 outputs R0-R3 are routed via the back-bus (connector B) to the IXL/A circuit card.

4.6 FS-6 - Bus Control, Interrupt, Trap and Self Test Logic -(See 4401SD Sheet 9)

4.6.1 Bus Control Logic - TELEBUS transceiver MLB5 receives the 4 micro-program TELEBUS control leads MPCØ-MPC2 and MP INTG, buffers these leads with "open collector" drivers and sends them out to the TELEBUS to control the BUS operations. Pins 7 and 9 are strapped to ground to permanently enable the BUS outputs.

The MLC14 decoder monitors the TELEBUS data/address lines and serves to decode two groups of TELEBUS addresses. The first group is address range 160000 to 161776 (octal) which defines the I/O device addresses falling into the "I/O A" selection group. The second range is 162000 to 163776 (octal) and defines the "I/O B" device selection group. Whenever an address falling into either of the two device ranges appears on the TELEBUS, the "I/O A" or "I/O B" leads are asserted LOW on MLC14-(9) and (7) respectively. These signals are inverted by the MLC15-(6) and (3) gates and converted to TELEBUS "open collector" signals by MLB4-(11) and MLC5-(15) respectively; from where they are routed along the BUS to the various I/O devices.

4.6.2 Interrupt and Trap Logic

The interrupt request, INTR, is received from the TELEBUS via the MLC5 (pin 6) bus transceiver. The received signal, R INTR, is transmitted to the MLB18 quad D flip-flop where it is synchronized with the positive edge of the ACC CLOCK signal from the timing logic. This is done to avoid critical races in the subsequent address encoding logic which has the synchronized INTR lead as an input to gate MLB16-(5).

4.6.2 Continued

The trap request, TRAP, is received from the TELEBUS via the MLC5 (Pin 1) bus transceiver. The received signal, R TRAP, is sent through the MLC17 NOR gate to 3 of the inputs of the MLB17 dual 1-4 selector. The feed back select code (FBØ-FB2), originating from the microprogram control word ROM MLA14-(9-11), enables passage of the trap condition through the multiplexer when the FB1 and FB2 leads are not both HIGH. The output of the multiplexer is clocked into pin 12 of the MLB18 quad D flip-flop on the positive transition of the ACC CLOCK signal from the timing logic (FS-7). Once the D flip-flop is set from an external trap condition, the value of the flip-flop is maintained through subsequent cycles of the ACC CLOCK because the flip-flop output PRE TRAP is looped back to provide the second input of the MLC17-(13) NOR gate. To terminate or reset such an external trap condition the microprogram FBØ, FB1 and FB2 leads are all switched to a HIGH level, allowing a LOW Level to be clocked into the flip-flop on the subsequent ACC CLOCK cycle. This is done by the first micro-step in the trap microroutine, since execution of this micro-step indicates that the above trap condition has been recognized and is being serviced by the micro-program.

The microprogram can also initiate a software trap, without an external trap condition. This is done by programming the FBØ and FB1 leads HIGH, and the FB2 lead LOW, which allows a HIGH level to be clocked into the MLB18-(10) D flip-flop. The "RETURN FROM TRAP WITH TRAP" (RTT or RTC) macro-instructions makes use of this feature since, at the end of its execution, the trap condition is reinitiated. This microprogram initiated trap condition is delayed by one IXL clock cycle since the PRE TRAP lead from MLB18-(10) is again synchronized with the positive edge of the IG CLOCK in the MLC16-(14) D flip-flop. This allows any software trap condition, generated at the end of RTT or RTC instructions to be delayed past the first step of the FETCH routine; allowing the next user instruction to be fully executed before the trap condition is again serviced. This mode of operation is used by the SODA debug system to emulate single instruction execution and the instruction trace techniques. The output of the MLC16-(14) flip-flop is called the $\overline{\text{TRAP}}$ lead and generates an input to the address encoding logic of FS-3, where it causes an entry into the TRAP sub-block of the FETCH routine at the end of the execution of the current instruction.

4.6.3 Odd Address Logic

Whenever the microprogram places byte operand addresses on the TELEBUS during the execution of a SOURCE or DESTINATION addressing mode, the low order bit (D/A0) of the address is tested for a "one" (LOW on TELEBUS). If this condition occurs, the byte operand subsequently delivered by the memory or I/O device exists on the upper 8 bits of the TELEBUS. The microprogram must swap the byte operand onto the lower 8 bits of the IXL/A data path to properly perform the required arithmetic or logical function. The controls for these operations is generated in ODD BYTE sub-block of the SRC II and DST II microroutines. Entry into these sub-blocks is controlled by the ODD BYTE lead from the MLB18-(14) D flip-flop which conditions the next state address encoding logic to the proper address of the sub-blocks.

4.6.3 Continued

The D/AØ bus lead is gated into the flip-flop via the MLB17 multiplexer whenever the FBØ lead is LOW and FB1 lead is HIGH. The odd byte condition is cleared from two places in the microprogram as follows:

a) In the execution of the ODD BYTE sub-block of the SRC II routine. At this point the odd address condition generated during the accessing of the source operand has been recognized.

b) In the execution of the instruction FETCH routine. At this point any odd address condition generated during the accessing of the destination operand has been satisfied. It should be noted that, for the destination operand address, the odd byte condition must be preserved until the DATO routine is executed. Since an odd destination address requires that the byte result must be re-swapped onto the upper 8 bits of the TELEBUS for depositing into the correct memory or I/O device location. This is done by the controls generated during the execution of the ODD BYTE sub-block of the DATO microroutine.

The clearing of the ODD BYTE condition in the MLB18-(14) flip-flop occurs when the FBØ lead is HIGH and FB1 lead is LOW. The state of the ODD BYTE condition is preserved whenever the FBØ and FB1 leads are both HIGH or both LOW.

4.6.4 Self Test Logic

An IXL self test is originated by depressing the switch SW1. The switch output level is synchronized with the positive going edge of the IG CLOCK lead to avoid races in the subsequent address encoding logic. The test flip-flop, MLC16-(2) delivers a LOW signal to the MLB16-(3) gate and a HIGH signal to the MLB15-(1) gate in the address encoding logic. When the self test condition is active, the microprogram TEST sub-block is selected upon entry into the FETCH microroutine (at the end of the present instruction). The TEST sub-block control sequence sets up and tests various conditions in the IXL/A data paths. Each time a test condition is successfully met, one of the LEDs on the IXL/B card is activated via the microprogram. If every condition of the self-test is met, all 4 LEDs will appear to be active during the depression of SW1; since the microprogram executes the self test repeatedly.

4.7 Micro-Program Symbolic Listing

Table 2 contains the complete symbolic listing of the 256 word microprogram. Each horizontal entry in the listing constitutes an entire microprogram control word. The first three columns contain the name of the microroutine, the names of the sub-blocks within the routine and the binary microprogram address. The remaining columns each contain the mnemonic symbols representing a given control operation performed by the microprogram control field specified in the heading. Table 2A is a legend containing an index of the symbols and their corresponding control operations.

TABLE 2A

LEGEND OF MICRO-PROGRAM LISTING MNEMONICS

ALU FUNCTION

The expressions listed in the table are the arithmetic or logical values appearing on the ALU output bus. The expressions are written in terms of the A and B operand busses of the ALU (see the table of ALU operations in the 4400CD).

"A" - MUX SELECT

BUS - The contents of the TELEBUS are gated onto the A operand inputs of the ALU.

BUS SX - The sign extended contents of the TELEBUS are gated onto the A operand bus of the ALU.

BSWBSX - The contents of the TELEBUS are byte - swapped and sign extended and gated onto the A operand bus of the ALU.

ACC - The contents of the Accumulator are gated onto the A operand bus of the ALU.

ACC SX - The sign extended contents of the Accumulator are gated onto the A operand bus of the ALU.

ACC SWB - The byte swapped contents of the Accumulator are gated onto the A operand bus of the ALU.

ACC SR - The contents of the Accumulator are shifted right and gated onto the A operand bus of the ALU.

STATUS - The contents of the Status Register are gated onto the A operand bus of the ALU.

+1 - The constant 000001 (octal) is gated onto the A operand bus of the ALU.

-2 - The constant 177776 (octal) is gated onto the A operand bus of the ALU.

REG. CONTROL

"B" - REGISTER - The contents of the designated register is placed on the B operand bus of the ALU.

WLB - A "1" symbol in the column indicates that the lower byte of the designated register is written into.

TABLE 2A
(Continued)

WHB - A "1" symbol in this column indicates that the high byte of the designated register is written into.

CARRY

CIN

"0" - The carry input of the ALU is "0".
"1" - The carry input of the ALU is "1".
ACC15 - The carry input of the ALU is Accumulator bit 15.
C - The carry input of the ALU is the Status Register "C" bit.

SX

A7 - The value of the sign extension is derived from bit 7 of the multiplexer input bus.
C - The value of the sign extension is derived from the Status Register "C" bit.

OUT EN

A "1" indicates that the contents of the "B" operand bus are gated out onto the TELEBUS.

BUS

CONTROL

LAR - Sends the "load-address-and-read" bus command.
LDA - Sends the "load-address" bus command.
WWI - Sends the "write-word" bus command.
WBI - Sends the "write-byte" bus command.
RDI - Sends the "read and increment" bus command.
RDV - Sends the "read from device" bus command.
RDW - Sends the "read word" bus command.

G - A "1" indicates the "INTG" bus lead is active during this cycle.

LED - A binary combination used to activate the 4 LED's on the IXL/B card.

Bit Ø activates LEDØ
Bit 1 activates LED1
Bit 2 activates LED2
Bit 3 activates LED3

TABLE 2A
(Continued)

STATUS CNTRL

CLR T - Clear the T bit of the Status Register.
SET T - Set the T bit of the Status Register.
XFR - Load the following bits of the Status Register:

"C" bit loaded with "0"
"N" bit loaded with ALU output bit 15
"Z" bit loaded if ALU output is 000000 (octal)

ADD - Load the following bits of the Status Register:

"C" bit loaded with ALU carry output
"N" bit as in XFR
"Z" bit as in XFR

SUB - Load the following bits of the Status Register:

"C" bit loaded with inverted ALU carry output
"N" bit as in XFR
"Z" bit as in XFR

BUS - Load the entire status register from the output of the ALU.

SEQ CNTRL

BR - Branch to the next control word. The next control word is located at the address designated by the concatanation of the 4 "BRANCH ADDRESS" bits and the upper 4 bits of the existing microprogram address.

BR Z=1 - Branch to the next control word (as in BR) if the ALU output is 000000 (octal), otherwise index address counter for next address.

BR Z=0 - Branch to the next control word (as in BR) if the ALU output is not 000000 (octal), otherwise index address counter for next address.

DONE - Enter next major state microroutine address as defined by the address encoding logic.

F.B.

CT - Clear the trap condition
ST - Sets the trap condition (software trap)
COB - Clear odd byte condition
OBE - Enables sampling of D/AØ lead to test for odd byte condition
LIR - Loads the macro-instruction register

MICRO-PROGRAM SYMBOLIC LISTING

TABLE 2B

MICRO OUTINE	SUB BLOCK	BINARY ADDRESS	ALU FUNCTION	'A'-MUX SELECT	REG. CONTROL 'B'- REG.	REG. CONTROL WLB	REG. CONTROL WHB	CARRY CIN	CARRY SX	OUT EN	STATUS CNTRL	BUS CNTRL	BUS G	LED	SEQ CNTRL	BRANCH ADDRESS	F.B.
ETCH	PWR UP	00000000	A	-2	SRC	1	1	0	A7	0	CLRT	NØP	0	0000	BR	0001	CT
		00001	$\bar{A}$	+1	SRC	0	0	0	A7	0	BUS	NØP	0	0000	BR	1011	NØP
	FETCH	00010	A+B+1	+1	PC	1	1	1	A7	1	NØP	LAR	0	0000	BR	0011	CØB
		00011	A+A	BUS SX	SRC	1	1	0	A7	0	NØP	NØP	0	0000	DØNE	1111	LIR
	INTR	00100	A+B	-2	SP	1	1	0	A7	0	NØP	NØP	1	0000	BR	0101	NØP
		00101	A+B	-2	SP	1	1	0	A7	1	NØP	LDA	1	0000	BR	0110	NØP
		00110	A	STATUS	PC	1	1	0	A7	1	NØP	WWI	1	0000	BR	0111	NØP
		00111	$\bar{A}$	+1	SP	0	0	0	A7	1	BUS	LDA	1	0000	BR	1101	NØP
	TRAP	01000	B	BUS	PC	0	0	0	A7	0	SETT	NØP	0	0000	BR	1001	CT
		01001	A	ACC	TRAP	1	1	0	A7	0	NØP	NØP	0	0000	BR	1010	NØP
		01010	A+A	-2	SRC	1	1	0	A7	0	NØP	NØP	0	0000	BR	1011	NØP
		01011	B	BUS	SRC	0	0	0	A7	1	NØP	LAR	0	0000	BR	1100	NØP
	TRAP (CONT)	01100	A	BUS	PC	1	1	0	A7	0	NØP	NØP	0	0000	BR	0010	NØP
		01101	B	BUS	PC	0	0	0	A7	1	NØP	WWI	1	0000	BR	1110	NØP
		01110	A	BUS	IV	1	1	0	A7	0	NØP	RDV	1	0000	BR	1111	NØP
		01111	B	BUS	IV	0	0	0	A7	1	NØP	LAR	0	0000	BR	1100	NØP
	TEST	10000	A+1	-2	R3	1	1	1	A7	0	BUS	NØP	0	0001	BRZ=1	0000	CT
		10001	A+1	ACC SX	R0	1	1	1	A7	0	ADD	NØP	0	0001	BRZ=1	0011	NØP
		10010	A+B+CIN	BUS	PC	0	0	0	A7	1	NØP	NØP	0	0001	BR	0100	LIR
		10011	A+A+1	+1	PC	1	1	1	A7	0	NØP	NØP	0	0001	BR	0010	NØP
	TEST (CONT)	10100	A+A	ACC SX	RD	1	1	0	A7	0	CLRT	NØP	0	0001	BR	0101	NØP
		10101	A-B	ACSWB	RS	1	1	1	A7	0	NØP	NØP	0	0001	BR	0110	NØP
		10110	A∧B	STATUS	R3	1	1	0	A7	0	NØP	NØP	0	0001	BR	0111	NØP
		10111	A+B	+1	RD	1	1	0	A7	0	XFR	NØP	0	0001	BRZ=1	0000	NØP
	TEST (CONT)	11000	A+B	ACC SX	R3	1	1	AC15	C	0	NØP	NØP	0	0010	BR	1001	NØP
		11001	A+B+CIN	ACC	RD	1	1	C	A7	0	NØP	NØP	0	0010	BRZ=0	1011	NØP
		11010	A∨B	ACC	RO	1	1	0	A7	0	BUS	NØP	0	0010	BR	1100	NØP
		11011	A+B+1	BUS SX	R3	1	1	1	A7	1	NØP	NØP	0	0010	BR	1010	LIR
	TEST (CONT)	11100	0	ACC	R4	1	1	0	A7	0	NØP	NØP	0	0010	BRZ=0	1100	NØP
		11101	A⊕B	ACC SX	RD	1	1	AC15	C	0	NØP	NØP	0	0010	BR	1110	NØP
		11110	A∨B	ACC SR	RS	1	0	0	A7	0	NØP	NØP	0	0010	BR	1111	CØB
		11111	B	BUS	RO	0	0	0	A7	1	NØP	NØP	0	0010	DØNE	1111	LIR

MICRO-PROGRAM SYMBOLIC LISTING

MICRO OUTINE	SUB BLOCK	BINARY ADDRESS	ALU FUNCTION	'A'-MUX SELECT	REG. CONTROL 'B'- REG.	WLB	WHP	CARRY CIN	S X	OUT EN	STATUS CNTRL	BUS CNTRL	G	LED	SEQ CNTRL	BRANCH ADDRESS	F.B.
ØURCE	REG DIR WORD	00100000	B	BUS	RS	0	0	0	A7	0	NØP	NØP	0	0000	PC	0001	NØP
		00001	A	ACC	SRC	1	1	0	A7	0	NØP	NØP	0	0000	DØNE	1111	NØP
	REG DIR BYTE	00010	B	BUS	RS	0	0	0	A7	0	NØP	NØP	0	0000	BR	0011	NØP
		00011	A	ACC SX	SRC	1	1	0	A7	0	NØP	NØP	0	0000	DØNE	1111	NØP
	REG A-DEC WORD	00100	A+B	-2	RS	1	1	0	A7	0	NØP	NØP	0	0000	PC	0101	NØP
		00101	B	BUS	RS	0	0	0	A7	1	NØP	LAR	0	0000	DØNE	1111	NØP
	REG A-DEC BYTE	00110	A+B+1	-2	RS	1	1	1	A7	0	NØP	NØP	0	0000	BR	0111	NØP
		00111	B	BUS	RS	0	0	0	A7	1	NØP	LAR	0	0000	DØNE	1111	ØBE
	ABSOL WORD	01000	A+B+1	+1	PC	1	1	1	A7	1	NØP	LAR	0	0000	BR	1001	NØP
		01001	A	BUS	PC	0	0	0	A7	0	NØP	LAR	0	0000	DØNE	1111	NØP
	ABSOL BYTE	01010	A+B+1	+1	PC	1	1	1	A7	1	NØP	LAR	0	0000	BR	1011	NØP
		01011	A	BUS	PC	0	0	0	A7	0	NØP	LAR	0	0000	DØNE	1111	ØBE
	IMMED WORD	01100	A+B+1	+1	PC	1	1	1	A7	1	NØP	LAR	0	0000	DØNE	1111	NØP
		01101															
	IMMED BYTE	01110	A+B+1	+1	PC	1	1	1	A7	1	NØP	LAR	0	0000	DØNE	1111	NØP
		01111															
	REG DEF WORD	10000	B	BUS	RS	0	0	0	A7	1	NØP	LAR	0	0000	DØNE	1111	NØP
		10001	A	ACC	SRC	1	1	0	A7	0	NØP	NØP	0	0000	BR	0011	NØP
	REG DEF BYTE	10010	B	BUS	RS	0	0	0	A7	1	NØP	LAR	0	0000	DØNE	1111	ØBE
		10011	B	BUS	SRC	0	0	0	A7	1	NØP	LAR	0	0000	DØNE	1111	NØP
	REG AI WORD	10100	A+B+1	+1	RS	1	1	1	A7	1	NØP	LAR	0	0000	DØNE	1111	NØP
		10101	A	ACC	SRC	1	1	0	A7	0	NØP	NØP	0	0000	BR	0111	NØP
	REGAI BYTE	10110	A+B	+1	RS	1	1	0	A7	1	NØP	LAR	0	0000	DØNE	1111	ØBE
		10111	B	BUS	SRC	0	0	0	A7	1	NØP	LAR	0	0000	DØNE	1111	ØBE
	REL WORD	11000	A+B+1	+1	PC	1	1	1	A7	1	NØP	LAR	0	0000	BR	1001	NØP
		11001	A+B	BUS	PC	0	0	0	A7	0	NØP	NØP	0	0000	BR	0001	NØP
	REL BYTE	11010	A+B+1	+1	PC	1	1	1	A7	1	NØP	LAR	0	0000	BR	1011	NØP
		11011	A+B	BUS	PC	0	0	0	A7	0	NØP	NØP	0	0000	BR	0101	NØP
	INDEX WORD	11100	A+B+1	+1	PC	1	1	1	A7	1	NØP	LAR	0	0000	BR	1101	NØP
		11101	A+B	BUS	RS	0	0	0	A7	0	NØP	NØP	0	0000	BR	0001	NØP
	INDEX BYTE	11110	A+B+1	+1	PC	1	1	1	A7	1	NØP	LAR	0	0000	BR	1111	NØP
		11111	A+B	BUS	RS	0	0	0	A7	0	NØP	NØP	0	0000	BR	0101	NØP

TABLE 2C

MICRO-PROGRAM SYMBOLIC LISTING

MICRO OUTINE	SUB BLOCK	BINARY ADDRESS	ALU FUNCTION	'A'-MUX SELECT	REG. CONTROL 'B'- REG.	WLB	WHB	CARRY CIN	SX	OUT EN	STATUS CNTRL	BUS CNTRL	G	LED	SEQ CNTRL	BRANCH ADDRESS	F.B.
ESTINATION	REG DIR WORD	01000000 00001	B	BUS	RD	0	0	0	A7	0	NØP	NØP	0	0000	DØNE	1111	NØP
	REG DIR BYTE	00010 00011	B A	BUS ACC SX	RD TEMP	0 0	0 0	0 0	A7 A7	0 0	NØP NØP	NØP NØP	0 0	0000 0000	BR DØNE	0011 1111	NØP NØP
	REG A-DEC WORD	00100 00101	A+B B	-2 BUS	RD RD	1 0	1 0	0 0	A7 A7	0 1	NØP NØP	NØP LAR	0 0	0000 0000	BR DØNE	0101 1111	NØP NØP
	REG A-DEC BYTE	00110 00111	A+B+1 B	-2 BUS	RD RD	1 0	1 0	1 0	A7 A7	0 1	NØP NØP	NØP LAR	0 0	0000 0000	BR DØNE	0111 1111	NØP ØBE
	ABSOL WORD	01000 01001	A+B+1 A	+1 BUS	PC PC	1 0	1 0	1 0	A7 A7	1 0	NØP NØP	LAR LAR	0 0	0000 0000	BR DØNE	1001 1111	NØP NØP
	ABSOL BYTE	01010 01011	A+B+1 A	+1 BUS	PC PC	1 0	1 0	1 0	A7 A7	1 0	NØP NØP	LAR LAR	0 0	0000 0000	BR DØNE	1011 1111	NØP ØBE
	ABSOL DEFRD WORD	01100 01101	A+B+1 A	+1 BUS	PC PC	1 0	1 0	1 0	A7 A7	1 0	NØP NØP	LAR LAR	0 0	0000 0000	BR BR	1101 1001	NØP NØP
	ABSOL DEFRD BYTE	01110 01111	A+B+1 A	+1 BUS	PC PC	1 0	1 0	1 0	A7 A7	1 0	NØP NØP	LAR LAR	0 0	0000 0000	BR BR	1111 1011	NØP NØP
	REG DEF WORD	10000 10001	B A	BUS ACC	RD TEMP	0 1	0 1	0 0	A7 A7	1 0	NØP NØP	LAR NØP	0 0	0000 0000	DØNE BR	1111 0011	NØP NØP
	REG DEF BYTE	10010 10011	B B	BUS BUS	RD TEMP	0 0	0 0	0 0	A7 A7	1 1	NØP NØP	LAR LAR	0 0	0000 0000	DØNE DØNE	1111 1111	ØBE NØP
	REG AI WORD	10100 10101	A+B+1 A	+1 ACC	RD TEMP	1 1	1 1	1 0	A7 A7	1 0	NØP NØP	LAR NØP	0 0	0000 0000	DØNE BR	1111 0111	NØP NØP
	REG AI BYTE	10110 10111	A+B B	+1 BUS	RD TEMP	1 0	1 0	0 0	A7 A7	1 1	NØP NØP	LAR LAR	0 0	0000 0000	DØNE DØNE	1111 1111	ØBE ØBE
	REL WORD	11000 11001	A+B+1 A+B	+1 BUS	PC PC	1 0	1 0	1 0	A7 A7	1 0	NØP NØP	LAR NØP	0 0	0000 0000	BR BR	1001 0001	NØP NØP
	REL BYTE	11010 11011	A+B+1 A+B	+1 BUS	PC PC	1 0	1 0	1 0	A7 A7	1 0	NØP NØP	LAR NØP	0 0	0000 0000	BR BR	1011 0101	NØP NØP
	INDEX WORD	11100 11101	A+B+1 A+B	+1 BUS	PC RD	1 0	1 0	1 0	A7 A7	1 0	NØP NØP	LAR NØP	0 0	0000 0000	BR BR	1101 0001	NØP NØP
	INDEX BYTE	11110 11111	A+B+1 A+B	+1 BUS	PC RD	1 0	1 0	1 0	A7 A7	1 0	NØP NØP	LAR NØP	0 0	0000 0000	BR BR	1111 0101	NØP NØP

TABLE 2D

MICRO-PROGRAM SYMBOLIC LISTING

MICRO ROUTINE	SUB BLOCK	BINARY ADDRESS	ALU FUNCTION	'A'-MUX SELECT	REG. CONTROL 'B'- REG.	REG. CONTROL WLB	REG. CONTROL WHB	CARRY CIN	CARRY SX	OUT EN	STATUS CNTRL	BUS CNTRL	BUS G	LED	SEQ CNTRL	BRANCH ADDRESS	F.B.
DESTINATION (cont'd)	REG DEF	01100000	B	BUS	RD	0	0	0	A7	0	NØP	NØP	0	0000	BR	0001	NØP
		00001	A	ACC	SRC	1	1	0	A7	0	NØP	NØP	0	0000	DØNE	1111	NØP
BRANCH	BRANCH	00010	A+B	ACC	PC	1	1	0	A7	0	NØP	NØP	0	0000	DØNE	1111	NØP
		00011															
DESTINATION (cont'd)	REG A DEC	00100	A+B	-2	RD	1	1	0	A7	0	NØP	NØP	0	0000	BR	0101	NØP
		00101	B	BUS	RD	0	0	0	A7	1	NØP	LAR	0	0000	BR	0110	NØP
		00110	A	BUS	SRC	1	1	0	A7	0	NØP	NØP	0	0000	DØNE	1111	NØP
		00111															
	REL	01000	A+B+1	+1	PC	1	1	1	A7	1	NØP	LAR	0	0000	BR	1001	NØP
		01001	A+B	BUS	PC	0	0	0	A7	0	NØP	NØP	0	0000	BR	0001	NØP
		01010															
		01011															
	INDEX DEFRD	01100	A+B+1	+1	PC	1	1	1	A7	1	NØP	LAR	0	0000	BR	1101	NØP
		01101	A+B	BUS	RD	0	0	0	A7	0	NØP	NØP	0	0000	BR	1110	NØP
		01110	A	ACC	SRC	1	1	0	A7	0	NØP	NØP	0	0000	BR	1111	NØP
		01111	B	BUS	SRC	0	0	0	A7	1	NØP	LAR	0	0000	BR	0110	NØP
	ABSOL	10000	A+B+1	+1	PC	1	1	1	A7	1	NØP	LAR	0	0000	BR	0001	NØP
		10001	A	BUS	SRC	1	1	0	A7	0	NØP	NØP	0	0000	DØNE	1111	NØP
		10010	A-B-$\overline{C_{IN}}$	ACC	RS	1	1	C	A7	0	NØP	NØP	0	0100	BR Z=0	1110	NØP
		10011	A+A	-2	R2	1	1	0	A7	0	BUS	NØP	0	0100	BR	0110	NØP
	ABSOL DEFRD	10100	A+B+1	+1	PC	1	1	1	A7	1	NØP	LAR	0	0000	BR	0101	NØP
		10101	A	BUS	PC	0	0	0	A7	0	NØP	LAR	0	0000	BR	0001	NØP
		10110	A+B	ACC	R1	1	1	0	A7	0	NØP	NØP	0	0100	BR	0111	NØP
		10111	A⊕B	ACC	RD	0	0	0	A7	0	NØP	NØP	0	0100	BR Z=1	1001	NØP
	REG AUTO IN	11000	A+B+1	+1	RD	1	1	1	A7	1	NØP	LAR	0	0000	BR	0001	NØP
		11001	B	ACC	RS	0	0	0	A7	0	NØP	NØP	0	1000	BR	1010	NØP
		11010	A+A+1	ACC	RS	0	0	1	A7	0	XFR	NØP	0	1000	BR	1011	NØP
		11011	A+A	ACC	RS	0	0	0	A7	0	NØP	NØP	0	1000	BR	1101	NØP
	TEST	11100	A	ACC SR	R2	1	1	0	A7	0	SHR	NØP	0	0100	BR	0010	NØP
		11101	A	ACC SX	RS	0	0	ACC15	C	0	NØP	NØP	0	1000	BR	1111	NØP
		11110	A	ACC	RS	0	0	0	A7	0	NØP	NØP	0	0100	DØNE	1111	NØP
		11111	A	ACSWB	RS	0	0	0	A7	0	NØP	NØP	0	1000	DØNE	1111	NØP

TABLE 2E

TABLE 2F

MICRO-PROGRAM SYMBOLIC LISTING

MICRO ROUTINE	SUB BLOCK	BINARY ADDRESS	ALU FUNCTION	'A'-MUX SELECT	REG. CONTROL 'B'- REG.	WLB	WHB	CARRY CIN	SX	OUT EN	STATUS CNTRL	BUS CNTRL	G	LED	SEQ CNTRL	BRANCH ADDRESS	F.B.
CTRL	SOFTWR TRAP	100 00000	B	BUS	PC	0	0	0	A7	0	SET T	NØP	0	0000	BR	0001	NØP
		00001	A	ACC	TRAP	1	1	0	A7	0	NØP	NØP	0	0000	BR	0010	NØP
		00010	A+A	-2	SRC	0	0	0	A7	0	NØP	NØP	0	0000	BR	0011	NØP
		00011	A-B	ACC	SRC	1	1	1	A7	0	NØP	NØP	0	0000	BR	0100	NØP
	SOFTWR TRAP (CONT)	00100	A+B	-2	SRC	1	1	0	A7	0	NØP	NØP	0	0000	BR	0101	NØP
		00101	B	BUS	SRC	0	0	0	A7	1	NØP	LAR	0	0000	BR	0111	NØP
	RTRN FRM TRAP & TRAP	00110	B	BUS	TRAP	0	0	0	A7	0	NØP	NØP	0	0000	BR	1111	NØP
		00111	A	BUS	PC	1	1	0	A7	0	NØP	NØP	0	0000	DØNE	1111	NØP
	CCØPC	01000	A	ACC-SR	SRC	1	1	0	A7	0	NØP	NØP	0	0000	BR	1001	NØP
		01001	$A \wedge \overline{B}$	STATUS	SRC	0	0	0	A7	0	BUS	NØP	0	0000	DØNE	1111	NØP
	RTRN FRM TRAP	01010	B	BUS	TRAP	0	0	0	A7	0	CLRT	NØP	0	0000	BR	1011	NØP
		01011	A	ACC	PC	1	1	0	A7	0	NØP	NØP	0	0000	DØNE	1111	NØP
	CCØP S	01100	A	ACC-SR	SRC	1	1	0	A7	0	NØP	NØP	0	0000	BR	1101	NØP
		01101	$A \vee B$	STATUS	SRC	0	0	0	A7	0	BUS	NØP	0	0000	DØNE	1111	NØP
	RTRN FRM TRAP & TRAP (CLRT)	01110	B	BUS	TRAP	0	0	0	A7	0	CLRT	NØP	0	0000	BR	1111	NØP
		01111	A	ACC	PC	1	1	0	A7	0	NØP	NØP	0	0000	DØNE	1111	S-TRAP
	RBT	10000	A+B+1	+1	PC	1	1	1	A7	1	NØP	LAR	0	0000	BR	0001	NØP
		10001	A	BUS	SRC	1	1	0	A7	0	NØP	NØP	0	0000	BR	0010	NØP
	RBT (CONT)	10010	A+B+1	+1	PC	1	1	1	A7	1	NØP	LAR	0	0000	BR	0011	NØP
		10011	A	BUS	TEMP	0	0	0	A7	0	NØP	LAR	0	0000	BR	0100	NØP
	RBT (CONT)	10100	A	BUS	TEMP	1	1	0	A7	0	NØP	NØP	0	0000	BR	0101	NØP
		10101	A+B+1	+1	TEMP	1	1	1	A7	0	NØP	NØP	0	0000	BR	0110	NØP
		10110	A+B	-2	TEMP	1	1	0	A7	1	NØP	WWI	0	0000	BR	0111	NØP
		10111	B	BUS	TEMP	0	0	0	A7	1	NØP	LAR	0	0000	BR	1000	NØP
	RBT (CONT)	11000	A	BUS	TEMP	0	0	0	A7	0	NØP	LDA	0	0000	BR	1001	NØP
		11001	A+B+1	-2	SRC	1	1	1	A7	0	NØP	RDI	0	0000	BRZ=1	1011	NØP
		11010	A+B+1	-2	SRC	1	1	1	A7	0	NØP	RDI	0	0000	BRZ=0	1010	NØP
		11011	A	BUS	SRC	0	0	0	A7	0	NØP	NØP	0	0000	BR	1100	NØP
	RTI	11100	A+B+1	+1	SP	1	1	1	A7	1	NØP	LAR	0	0000	BR	1101	NØP
		11101	$\overline{A}$	BUS	SP	0	0	0	A7	0	BUS	NØP	0	0000	BR	1110	NØP
		11110	A+B+1	+1	SP	1	1	1	A7	1	NØP	LMR	0	0000	BR	1111	NØP
		11111	A	BUS	PC	1	1	0	A7	0	NØP	NØP	0	0000	DØNE	1111	NØP

TABLE 2G

MICRO-PROGRAM SYMBOLIC LISTING

MICRO ROUTINE	SUB BLOCK	BINARY ADDRESS	ALU FUNCTION	'A'-MUX SELECT	REG. CONTROL 'B'- REG.	WLB	WHB	CARRY CIN	SX	OUT EN	STATUS CNTRL	BUS CNTRL	G	LED	SEQ CNTRL	BRANCH ADDRESS	F.B.
CONTROL (cont'd)	LED Ø	101 00000	A-1	ACC	SRC	0	0	0	A7	0	NØP	NØP	0	0001	BRZ=0	0000	NØP
		00001	B	BUS	SRC	0	0	0	A7	0	NØP	NØP	0	0001	DONE	1111	NØP
	SEARCH	00010	A+B+1	+1	PC	1	1	1	A7	1	NØP	LAR	0	0000	BR	0011	NØP
		00011	A	BUS	SRC	1	1	0	A7	0	NØP	NØP	0	0000	BR	0100	NØP
	SEARCH (CONT)	00100	A+B+1	+1	PC	1	1	1	A7	1	NØP	LAR	0	0000	BR	0101	NØP
		00101	A	BUS	TEMP	1	1	0	A7	0	NØP	NØP	0	0000	BR	0110	NØP
		00110	$\bar{A} \wedge B$	+1	RD	1	1	0	A7	0	XFR	NØP	0	0000	BRZ=1	1100	NØP
		00111	A+B	ACC	RO	1	1	0	A7	1	NØP	LDA	0	0000	BR	1000	NØP
	SEARCH (CONT)	01000	A	BUS	SRC	0	0	0	A7	0	NØP	RDI	0	0000	BR	1001	NØP
		01001	$A \wedge B$	BUS	SRC	0	0	0	A7	0	NØP	NØP	0	0000	BR	1010	NØP
		01010	$A \nwedge B$	ACC	TEMP	0	0	0	A7	0	NØP	NØP	0	0000	BRZ=0	1101	NØP
		01011	A+B	-2	RD	1	1	0	A7	0	XFR	RDI	0	0000	BRZ=0	1001	NØP
	SEARCH (CONT)	01100	A+B	ACC	RO	1	1	0	A7	0	NØP	NØP	0	0000	DONE	1111	NØP
		01101	A-B-1	+1	RD	0	0	0	A7	0	NØP	NØP	0	0000	BR	1100	NØP
	LD TRAP	01110	B	BUS	RD	0	0	0	A7	0	SET T	NØP	0	0000	BR	1111	NØP
		01111	A	ACC	TRAP	1	1	0	A7	0	NØP	NØP	0	0000	DONE	1111	NØP
	LED 1	10000	A-1	ACC	SRC	0	0	0	A7	0	NØP	NØP	0	0010	BRZ=0	0000	NØP
		10001	B	BUS	SRC	0	0	0	A7	0	NØP	NØP	0	0010	DONE	1111	NØP
		10010															
		10011															
		10100															
		10101															
		10110															
		10111															
		11000															
		11001															
		11010															
		11011															
		11100															
		11101															
		11110															
		11111															

MICRO-PROGRAM SYMBOLIC LISTING

MICRO ROUTINE	SUB BLOCK	BINARY ADDRESS	ALU FUNCTION	'A'-MUX SELECT	REG. CONTROL 'B'- REG.	WLB	W'+B	CARRY CIN	SX	OUT EN	STATUS CNTRL	BUS CNTRL	G	LED	SEQ CNTRL	BRANCH ADDRESS	F.B.
CONTROL (Contd)	LED 2	110 00000	A-1	ACC	SRC	0	0	0	A7	0	NØP	NØP	0	0100	BRZ=0	0000	NØP
		00001	B	BUS	SRC	0	0	0	A7	0	NØP	NØP	0	0100	DØNE	1111	NØP
	STORE	00010	A+B+1	+1	PC	1	1	1	A7	1	NØP	LAR	0	0000	BR	0011	NØP
		00011	A	BUS	SRC	1	1	0	A7	0	NØP	NØP	0	0000	BR	0100	NØP
	STORE (CONT)	00100	A+B+1	+1	PC	1	1	1	A7	1	NØP	LAR	0	0000	BR	0101	NØP
		00101	A	BUS	TEMP	1	1	0	A7	0	NØP	NØP	0	0000	BR	0110	NØP
		00110	B	BUS	RO	0	0	0	A7	1	NØP	LAR	0	0000	BR	1101	NØP
		00111	A	ACC	X	1	1	0	A7	0	NØP	NØP	0	0000	BR	1001	NØP
SRC II	WORD	01000	A	BUS	SRC	1	1	0	A7	0	NØP	RDV	0	0000	DØNE	1111	NØP
		01001	B	BUS	X	0	0	0	A7	1	NØP	WWI	0	0000	BR	1100	NØP
	BYTE	01010	A	BLSSX	SRC	1	1	0	A7	0	NØP	RDV	0	0000	DØNE	1111	NØP
	ODD BYTE	01011	A	BSWBSX	SRC	1	1	0	A7	0	NØP	RDV	0	0000	DØNE	1111	CØB
		01100	A+B	-2	RD	1	1	0	A7	0	NØP	RDW	0	0000	BRZ=1	0001	NØP
		01101	A∧B	BUS	SRC	0	0	0	A7	0	NØP	NØP	0	0000	BR	1111	NØP
		01110	A	ACCSX	TEMP	0	0	ACC15	C	0	NØP	NØP	0	0000	DØNE	1111	NØP
		01111	A∨B	ACC	TEMP	0	0	0	A7	0	NØP	NØP	0	0000	BR	0111	NØP
CONTROL (Contd)	LED 3	10000	A-1	ACC	SRC	0	0	0	A7	0	NØP	NØP	0	1000	BRZ=0	0000	NØP
		10001	B	BUS	SRC	0	0	0	A7	0	NØP	NØP	0	1000	DØNE	1111	NØP
	MOV IV REG	10010	B	BUS	IV	0	0	0	A7	0	NØP	NØP	0	0000	BR	0011	NØP
		10011	A	ACC	RD	1	1	0	A7	0	NØP	NØP	0	0000	DØNE	1111	NØP
	WAIT	10100	A+B+1	-2	RD	1	1	1	A7	0	NØP	RDW	0	0000	BRZ=0	0100	NØP
		10101	A	BUS	RD	0	0	0	A7	0	NØP	NØP	0	0000	DØNE	1111	NØP
	READ IV	10110	A+A	ACCSX	IV	0	0	0	A7	0	NØP	NØP	1	0000	BRZ=0	0110	NØP
		10111	A	BUS	IV	0	0	0	A7	0	NØP	NØP	1	0000	BR	1001	NØP
DST II	WORD	11000	A	BUS	TEMP	0	0	0	A7	0	NØP	RDV	0	0000	DØNE	1111	NØP
		11001	A	BUS	IV	1	1	0	A7	0	NØP	RDV	1	0000	DØNE	1111	NØP
	BYTE	11010	A	BUSSX	TEMP	0	0	0	A7	0	NØP	RDV	0	0000	DØNE	1111	NØP
	ODD BYTE	11011	A	BSWBSX	TEMP	0	0	0	A7	0	NØP	RDV	0	0000	DØNE	1111	NØP
	RORAWD	11100	A	BUS	TEMP	0	0	0	A7	0	NØP	RDV	0	0000	DØNE	1111	NØP
	RORABY	11101															
		11110	A	BUS SX	TEMP	0	0	ACC15	C	0	NØP	RDV	0	0000	DØNE	1111	NØP
	RORAOB	11111	A	BSWBSX	TEMP	0	0	ACC15	C	0	NØP	RDV	0	0000	DØNE	1111	NØP

MICRO-PROGRAM SYMBOLIC LISTING

MICRO ROUTINE	SUB BLOCK	BINARY ADDRESS	ALU FUNCTION	'A'-MUX SELECT	REG. CONTROL 'B'- REG.	WLB	WHB	CARRY CIN	SX	OUT EN	STATUS CNTRL	BUS CNTRL	G	LED	SEQ CNTRL	BRANCH ADDRESS	F.B.
EXECUTE	SUB	111 00000	A-B	ACC	SRC	1	1	1	A7	0	SUB	NØP	0	0000	DØNE	1111	NØP
	MØV	00001	B	ACC	SRC	1	1	0	A7	0	XFR	NØP	0	0000	DØNE	1111	NØP
	CMP	00010	A-B	ACC	SRC	1	1	1	A7	0	SUB	NØP	0	0000	DØNE	1111	NØP
	BIT	00011	A∧B	ACC	SRC	1	1	0	A7	0	XFR	NØP	0	0000	DØNE	1111	NØP
	BIC	00100	A∧$\bar{B}$	ACC	SRC	1	1	0	A7	0	XFR	NØP	0	0000	DØNE	1111	NØP
	BIS	00101	A∨B	ACC	SRC	1	1	0	A7	0	XFR	NØP	0	0000	DØNE	1111	NØP
	XØR	00110	A⊻B	ACC	SRC	1	1	0	A7	0	XFR	NØP	0	0000	DØNE	1111	NØP
	ADD	00111	A+B	ACC	SRC	1	1	0	A7	0	ADD	NØP	0	0000	DØNE	1111	NØP
	CLR	01000	0	ACC	SRC	1	1	0	A7	0	XFR	NØP	0	0000	DØNE	1111	NØP
	CØM	01001	$\bar{A}$	ACC	SRC	1	1	0	A7	0	XFR	NØP	0	0000	DØNE	1111	NØP
	INC	01010	A+1	ACC	SRC	1	1	1	A7	0	XFR	NØP	0	0000	DØNE	1111	NØP
	DEC	01011	A-1	ACC	SRC	1	1	0	A7	0	XFR	NØP	0	0000	DØNE	1111	NØP
	TST	01100	A	ACC	SRC	1	1	0	A7	0	XFR	NØP	0	0000	DØNE	1111	NØP
	ASL	01101	A+A	ACC	SRC	1	1	0	A7	0	ADD	NØP	0	0000	DØNE	1111	NØP
	ASR	01110	A	ACC SR	SRC	1	1	ACC15	C	0	SHR	NØP	0	0000	DØNE	1111	NØP
	RØL	01111	A+A+CIN	ACC	SRC	1	1	C	A7	0	ADD	NØP	0	0000	DØNE	1111	NØP
DATO	WØRD	10000	B	BUS	SRC	0	0	0	A7	1	NØP	WWI	0	0000	DØNE	1111	NØP
		10001	B	BUS	SP	0	0	0	A7	1	NØP	LDA	0	0000	BR	0101	NØP
	BYTE	10010	B	BUS	SRC	0	0	0	A7	1	NØP	WBI	0	0000	DØNE	1111	NØP
	ØDD BY	10011	A	ACC SWB	SRC	1	1	0	A7	0	NØP	NØP	0	0000	BR	0010	NØP
	DMØ*WD	10100	A	ACC	RD	1	1	0	A7	0	NØP	NØP	0	0000	DØNE	1111	NØP
		10101	B	BUS	PC	0	0	0	A7	1	NØP	WWI	0	0000	BR	0111	NØP
	DMØ*BY	10110	A	ACC	RD	1	0	0	A7	0	NØP	NØP	0	0000	DØNE	1111	NØP
		10111	B	BUS	SRC	0	0	0	A7	0	NØP	NØP	0	0000	BR	1110	NØP
EXECUTE (CONT)	RØR	11000	A	ACC SR	SRC	1	1	C	A7	0	SHR	NØP	0	0000	DØNE	1111	NØP
	SWAB	11001	A	ACC SWB	SRC	1	1	0	A7	0	XFR	NØP	0	0000	DØNE	1111	NØP
	TSR	11010	A	STATUS	SRC	1	1	0	A7	0	NØP	NØP	0	0000	DØNE	1111	NØP
	TTR	11011	B	ACC	TRAP	0	0	0	A7	0	NØP	NØP	0	0000	BR	1100	NØP
		11100	A	ACC	SRC	1	1	0	A7	0	NØP	NØP	0	0000	DØNE	1111	NØP
		11101	A	ACC SX	SRC	0	0	0	A7	0	XFR	NØP	0	0000	DØNE	1111	NØP
	JMP	11110	A	ACC	PC	1	1	0	A7	0	NØP	NØP	0	0000	DØNE	1111	NØP
	JSR	11111	A+B	-2	SP	1	1	0	A7	0	NØP	NØP	0	0000	BR	0001	NØP

TABLE 21

4.7.1 Example of microprogram sequencing.

All executions of macro-instructions begin in the FETCH microroutine, or major state. Normal entry into FETCH is made at address 2 (octal) if no interrupts, traps or tests are pending. The following control sequence is generated.

At Address 2 - Place the constant 000001 (octal) on the A operand bus, and register PC (program counter) on the B operand bus. Perform the ALU operation A+B+1, causing PC+2 to be generated at the output of the ALU. Place the contents of the B operand bus (PC) onto the TELEBUS and generate a "Load-Address-and-read" command via the bus controls. Write the contents (PC+2) of the ALU output back into both bytes of the PC register. Clear any previous "odd byte" condition and branch to microprogram address 3.

At Address 3 - Sign extend bit A7 of the TELEBUS contents (result of previous LAR) and place it on the A operand bus. Perform the ALU operation A+A; this operation shifts the operand left (multiplies by 2). Write the contents of the ALU output into both bytes of the SRC register. Also load the contents of the TELEBUS into the macro-instruction register of IXL/B. The microroutine is DONE at this point, and an entry into the next state microroutine is executed. The address of the next microroutine is generated by the next state address encoding logic and depends on the type of instruction which has been clocked into the instruction register. The above IXL/A operations are done in case the instruction clocked into the instruction register happens to be a BRANCH; in this case the correct branch offset has been placed in the SRC register, and is used later in the BRANCH microroutine. If the upcoming instruction is not a BRANCH, the contents of the SRC register are later ignored. For this example, we will assume the instruction is an "ADD R1, R2" (not a BRANCH). The decoding of this instruction by the IXL/B logic causes a next state address generation of 40 (octal), which is the address of the REG DIR sub-block of the SOURCE microroutine.

At Address 40 - Place the contents of the R_S register (the register designated by IR6-IR8) on the B operand bus. Transfer the B operand through the ALU and into the accumulator (the output of the ALU is always clocked into the accumulator). Branch to microprogram address to address 41.

4.7.1 Continued

At Address 41 - Place the contents of the accumulator onto the A operand bus, transfer through the ALU and write into both bytes of the SRC register (this register is used to hold the source operand until execution of the arithmetic operation). The microroutine is DONE at this point. The address encoding logic generates a next state address of 100 (octal); this corresponds to the REG DIR sub-block of the DESTINATION microroutine.

At Address 100 - Place the contents of the R_D register (the register designated by IRØ-IR2) on the B operand bus. Transfer the B operand through the ALU and into the Accumulator. The DESTINATION microroutine is DONE at this point. The address encoding logic generates a next state address of 347 (octal); this corresponds to the ADD sub-block of the EXECUTE microroutine.

At Address 347 - Place the contents of the accumulator on the A operand bus, and the contents of the SRC register on the B operand bus. Perform the operation A+B in the ALU and write the results back into both bytes of the SRC register. The EXECUTE microroutine is DONE at this point, and the next state address encoding logic generates an address of 364 (octal). This corresponds to the DMO WD sub-block of the DATO microroutine.

At Address 364 - Place the contents of the accumulator on the A operand bus and transfer it through the ALU. Write the result back into both bytes of the R_D register. The DATO microroutine is DONE at this point. A next state address of 2, 4, 10 or 20 is generated to reinitiate the FETCH routine in the normal, interrupt, trap or test modes respectively.

Thus the "ADD R1, R2" instruction requires 7 micro-steps or 7 X 1.12 or 7.84 microseconds.

4.8 FS-7 Timing Logic and Restart Timer

4.8.1 General

4.8.1.1 The Timing Logic within the IXL/B card assembly generates seven waveforms to synchronize the operation of 40C400 Controller. Three of the waveforms are Telebus signals, namely I/O Clock, Bus Enable, and Memory Clock. The four remaining waveforms are used by the IXL, namely Microprogram Clock (MP CLK), Interrupt Grant Clock (IG CLK), Accumulator Clock (ACC CLK), and Write Pulse.

4.8.1.2 The Timing Logic normally operates with a period of 1.116 Microseconds for all waveforms. The Logic deviates from this period when the Telebus command is to read from a device and when the Controller is running under standby power. When reading from a device, the I/O Clock is maintained at a period of 1.117 uS. with the six other waveforms changed to 2.232 uS. for one period only. Under standby power, the Timing Logic generates only the I/O CLK waveforms at a period 17.856 uS.

The Timing Logic automatically change modes when the read from device command (RDV) is instituted by the microprogram or when the power switches to standby operation.

4.8.1.3 The timing waveforms are derived from the Crystal Oscillator (XTAL OSCILLATOR) oscillating at 14.336 MHz. with a period of 69.75 Nanoseconds.

4.8.1.4 The waveforms have a period of 1.116 Microseconds phased in increments of 69.75 uS. To obtain the 1.116 uS. period, the XTAL OSCILLATOR drives two divide-by-16 counters, the XCLK Counter (XCLK CNTR) and the Timing Ring Counter (TR CNTR). The XCLK CNTR is a four stage binary counter; the fourth stage generates I/O CLK. The TR CNTR is an eight stage twisted ring counter and generates six of the seven waveforms. Synchronization of the two counters and suppression of unwanted ring counter counting modes is the function of the TF Synchronizer gate.

4.8.1.5 Ten two input NAND and NOR gates decode the outputs of the TR CNTR and produce the Bus Enable, Memory Clock, MP CLK, IG CLK, ACC CLK and the Write Pulse.

4.8.1.6 To change the period of the six waveforms during a Read From Device (RDV) Telebus command, the RDV command is decoded from Telebus leads C0, C1, C2. The RDV DECODE circuit decodes the command. A signal $\overline{\text{RDV EN}}$ is generated within the IXL/B whenever DEVICE A ENABLE or DEVICE B ENABLE is actuated and is used with the RDV DECODE to trigger the "pulse stretching" of all timing signals except for the I/O CLK.

4.8.1.7 Provisions have been made to maintain +5V power to the XTAL OSCILLATOR and those integrated circuit packages which generate the I/O CLK when a set is equipped with a secondary power source. When power fails, the +5V Standby is to continue operation from the secondary source. During power failure, the I/O counter (I/O CNTR) is enabled and divides the I/O CLK by sixteen. During standby operation, the I/O CLK is active low on the Telebus for 17.3 uS. and becomes high for 0.56 uS. When normal power is restored, the I/O CNTR becomes passive and the I/O CLK returns to a period of 1.116 uS.

4.8.1.8 The Restart Timer is an analog timing circuit that actuates the INITIALIZE Telebus lead. The momentary closing of a remote Restart switch is differentiated by an R-C circuit to trigger the Restart Timer. In turn the Timer drives the INITIALIZE lead to a Low level for 50 milliseconds.

4.8.1.9 Refer to 4401SD-9 for the schematic of the Timing logic and 4401SD-10, -11 and -12 for the Timing Charts TC1, TC2, TC3 and TC4.

4.8.2 Supporting Information

4.8.2.1 Schematic 4401SD

4.8.3 Crystal Oscillator and Multiplexer

4.8.3.1 The Crystal Oscillator (XTAL OSCILLATOR) is contained in one hybrid circuit package, MLC1. The output drives a two input multiplexer (MLC2) normally selected to the XTAL OSCILLATOR. The multiplexer output (MLC2-(4) is XCLK, the basic clock signal for the timing generation.

4.8.3.2 The multiplexer MLC2 is used to replace the XTAL OSCILLATOR signal with an external source during production card testing or during use of a Test Panel. In normal operation, the signal leads OSC EN and MUX EN are held at a High voltage level by the resistors R4 and R6 connected to +5V Standby supply. To insert an external frequency source, the signal lead MUX EN (B3) is driven to a low level by external circuitry. The external frequency source is then connected to the signal lead OSC EN (B2). The multiplexer MLC2 selects the external source to become the signal XCLK.

4.8.3.3 The XTAL OSCILLATOR, package MLC2, and resistors R4 and R6 are powered from the +5V Standby supply. In standby operation, the signal XCLK is not interrupted.

4.8.4 XCLK Counter

4.8.4.1 The XCLK Counter, MLB1, is a four stage binary counter driven by the signal XCLK. The output of the last stage, MLB1-(11) is titled I/O CLK and is a square wave of frequency 896 kHz. Another XCLK CNTR output MLB1-(15) is entitled XCLK CNTR CARRY.

4.8.4.2 The time relationship of the XCLK Counter input and outputs are shown in TC-1 (4401SD-11) and TC-2 (4401SD-12). The signal CLK I/O is shown to be a sub-multiple of the XCLK frequency. The signal XCLK CNTR CARRY becomes active HIGH for one XCLK period prior to the HIGH TO LOW transition of the CLK I/O signal. In turn XCLK CNTR CARRY acts as a clock for the RDV Sequence Control.

4.8.4.3 The XCLK CNTR is powered from the +5 Volt Standby supply. The unused control leads of the SCLK CNTR is held HIGH by the resistor R2 and R3, connected also to +5 Volt Standby.

4.8.5 Timing Ring Counter

4.8.5.1 The Timing Ring Counter (TR CNTR) generates all timing waveforms except for I/O CLK. The division of timing signals is to allow standby or battery operation, and to allow a one cycle pause during Read from Device (RDV) Telebus operation. The I/O CLK signal is never interrupted while the TR CNTR outputs may be interrupted.

4.8.5.2 The Timing Ring Counter (TR CNTR) is an eight stage twisted ring counter composed of MLA2 and MLA1 integrated circuit packages. A twisted ring counter consists of a shift register with the output of the last stage inverted and connected to the first stage input. The TR CNTR consists of 8 D-Type flip-flops connected in series. The TF Synchronizer gate, MLB3, inverts the last stage output, TF7, and drives the first stage input, MLA2-(4).

4.8.5.3 The output signals of the TR CNTR are symmetrical pulses and are the 16th submultiple of CLK. The timing diagram TC-1 (4401SD-11) shows the TR CNTR outputs (TFØ through TF7); each output is displaced one XCLK period from the preceding output. Note that the TF Synchronizing gate synchronizes the TR CNTR with the I/O CLK signal, MLB1-(11), by gating or blocking the first stage input, MLB3-(4). In effect, the I/O CLK signal being in phase slows the TF signal.

4.8.6 Counter Decoders

4.8.6.1 The outputs (TFØ through TF7) are decoded by two input NOR and NAND gates to generate the required timing signals. The signals are generated as follows:

4.8.6.1.1 $\overline{\text{MEM CLK}} = \text{TF3} \wedge \overline{\text{TF6}}$

4.8.6.1.2 IG CLK $= \overline{\text{TFØ}}$

4.8.6.1.3 ACC CLK $= \overline{\overline{\text{TFØ}} \wedge \overline{\text{TF3}}}$

4.8.6.1.4 $\overline{\text{BUS EN}} = \text{TF2}$

4.8.6.1.5 MP CLK $= \text{TF4} \vee \overline{\text{TF3}}$

4.8.6.1.6 WRITE PULSE $= \overline{\text{TF4} \vee \overline{\text{TF3}}}$

4.8.6.2 The $\overline{\text{I/O CLK}}$ signal is derived through three stages of inversion, MLC3-(11), MLC3-(8) and MLC3-(6). A signal I/O CLK ENABLE is used to gate $\overline{\text{I/O CLK}}$ during standby operation. The integrated circuit package MLC3 is powered from +5 Volt Standby.

4.8.7 RDV Decode and RDV Sequence Control

4.8.7.1 The RDV Decode and RDV Sequence Control logic detects the RDV command coincident with either DEVA EN or DEVB EN Telebus signals and procedes to stop the TR CNTR for one bus cycle of 1.116 Microsecond. The pause is to allow response of alower I/O devices attached to the Telebus.

4.8.7.2 The RDV Decode gate decodes the Telebus controls generated by the Microprogram Storage Arrays MPCØ, MPC1, MPC2. The MLA4-(12) lead goes HIGH for RDV detected. The signal RDV EN is generated whenever either Telebus DEV ENABLE signal is propagated. The coincidence of MLA4-(12) and RDV EN sets the FFØ flip-flop when the CLK CTR CARRY LOW to HIGH transition occurs. In turn the RDV FFØ goes LOW which clears MLA2 and stops the TR CNTR. At the same transition, RDV FF1 is set to 1, MLA3-(15) = HIGH, blocking the RDV FFØ input. The next LOW to HIGH transition of XCLK CTR CARRY will clear RDV FFØ allowing TR CNTR to commence counting.

4.8.7.3 The timing relationship of the RDV sequence is shown in TC-2 (4401SD-12). The XCLK CNTR CARRY is the synchronizing or clocking signal that sequences the RDV pause.

4.8.8 I/O Counter

4.8.8.1 For operation under standby conditions, a clock is required operating at a lesser frequency than normal operation. The clock is to exercise the dynamic memories to retain the stored data. The IXL provides the I/O CLK during standby or battery operation operating at 56 KHz. The I/O Counter (I/O CTR) provides the logic and counting to generate the sub-multiple I/O CLK.

4.8.8.2 During normal operation, the I/O CTR (MLC4) is held at count 15. Since I/O CTR is a four stage counter, the carry output (I/O CLK ENABLE) is always HIGH, thereby enabling MLC3-(18) and allowing I/O CLK to propagate through MLC3-(6) to the Telebus. The I/O CTR is held at count 15 by loading each flip-flop with logic 1 at each X CLK CTR CARRY cycle. The load signal, MLC4-(9), is derived from a NAND gate MLB2-(8), with the inputs, MLB2-(9,10) held HIGH by resistor R1. When power fails the signal MLB2-(8) previously LOW becomes HIGH. When clocked through the flip-flop MLA3-(10) to synchronize the signal, the load command MLC4-(9) goes HIGH. The I/O CTR procedes to count and the I/O CLK ENABLE goes LOW blocking I/O CLK in a LOW state. Every 16 counts the I/O CLK ENABLE goes HIGH for one XCLK CTR CARRY cycle.

4.8.8.3 The timing diagram for standby power operation is shown in TC-3 (4401SD-13). The XCLK CTR CARRY becomes the synchronizing or clocking signal by controlling the ENP, MLC4-(7) lead of the I/O CTR. Note that the I/O CLK signal is the only Telebus signal defined for standby operation.

4.8.8.4 To remain in operation during standby conditions, the integrated circuit packages MLA3, MLB1, MLC1, MLC2, MLC3 and MLC4 are powered from the +5 Volt Standby supply.

4.9 Restart Timer

4.9.1 The Restart Timer is used to momentarily actuate the Telebus INITIALIZE to a low state for approximately 50 milliseconds. A HIGH to LOW transition at the input, RESTART (B55), is differentiated by the circuit of R16, R17 and C28 to trigger the timer, MLC18, with a HIGH to LOW pulse. Once triggered the timer output, POR, is actuated HIGH for approximately 50 milliseconds. The POR signal is inverted and drives the INITIALIZE lead LOW by the bus transceiver MLC5.