96728700

CONTROL DATA
CORPORATION

CDC® 1700 TRANSFORM WITH MICRO MEMORY

DE402-A

GENERAL DESCRIPTION
OPERATION
INSTALLATION AND CHECKOUT
THEORY OF OPERATION
DIAGRAMS
MAINTENANCE
EQUATION SUMMARY

HARDWARE MAINTENANCE MANUAL

REVISION RECORD

REVISION	DESCRIPTION
01	Preliminary manual released.
(10/75)	
A	Manual released.
(6/77)	
Publication No. 96728700	

REVISION LETTERS I, O, Q AND X ARE NOT USED

Address comments concerning this manual to:
Control Data Corporation
Publications and Graphics Division
4455 Eastgate Mall
La Jolla, California 92037

or use Comment Sheet in the back of this manual.

MANUAL TO EQUIPMENT LEVEL CORRELATION SHEET

This manual reflects the equipment configurations listed below.

EXPLANATION: Locate the equipment type and series number, as shown on the equipment FCO log, in the list below. Immediately to the right of the series number is an FCO number. If that number and all of the numbers underneath it match all of the numbers on the equipment FCO log, then this manual accurately reflects the equipment.

EQUIPMENT TYPE	SERIES	WITH FCOs	COMMENTS
AA109-A	02	ECO 14239/ 14241	
AB109-B	08		
AA123-B	03		
AA123-C	04		

LIST OF EFFECTIVE PAGES

New features, as well as changes, deletions, and additions to information in this manual, are indicated by bars in the margins or by a dot near the page number if the entire page is affected. A bar by the page number indicates pagination rather than content has changed.

PAGE	REV	PAGE	REV	PAGE	REV	PAGE	REV	PAGE	REV
Cover	--								
Title page	--								
ii	A								
iii/iv	A								
v/vi	A								
vii/viii	A								
ix	A								
x	A								
1-1	A								
2-1	A								
3-1	A								
4-1 thru 4-15	A								
5-1 thru 5-8	A								
6-1	A								
A-1	A								
Comment sheet	A								
Cover	--								

PREFACE

This manual is intended to be used in conjunction with the CONTROL DATA® Basic Micro-Programmable Processor Hardware Maintenance Manual to form a system manual. It contains the theory of operation, diagrams and maintenance information for the CONTROL DATA® DE402-A 1700 Transform with Read-Only Micro Memory. Information presented in this manual is intended for use by maintenance personnel in training and in the field.

The logic diagrams, parts data, and wire lists for this equipment are included in the field print package. Other documents that may be of use to the reader are listed below.

Refer to the Basic Micro-Programmable Processor Hardware Maintenance Manual for a listing of additional applicable documents and for a detailed description of the micro processor.

Title	Publication Number
1700 Transform with ROM Field Print Package	96751300
Basic Micro-Programmable Processor Hardware Maintenance Manual	39451400
CYBER 18 Computer Systems Installation Manual	39451500
CYBER 18 Processor with Core Memory (Macro Level) Reference Manual	88973500
CYBER 18-05/10 Computer Systems Hardware Maintenance Manual, Volumes 1 and 2	96767500 96767600
CYBER 18 Processor with MOS Memory (Macro Level) Hardware Reference Manual	96768300
CYBER 18 Processor (Micro Level) Reference Manual	88973400

CONTENTS

APPENDIX

FIGURES

TABLES

SCOPE

This manual contains the functional and physical descriptions of the 1700 transform with read-only micro memory. The 1700 transform module, when used with the basic micro processor, allows the 1700 instruction repertoire to be emulated more efficiently. This manual completes the CYBER 18 Processor Hardware Maintenance Manual when combined with the Basic Micro-Programmable Processor Hardware Maintenance Manual.

PHYSICAL DESCRIPTION

The 1700 transform module consists of one 11 by 14 inch printed circuit board, which plugs into slot R of the processor chassis.

ELECTRICAL DESCRIPTION

The 1700 transform module requires +5 V dc, which is obtained from the main central processor unit (CPU) power supply. This module draws approximately 4 amperes (average power consumption is about 20 watts).

The logic level of the 1700 transform module is transistor-transistor logic (TTL) level as follows:

Logical zero (low):	0.4 V dc or less
Logical one (high):	+2.4 to 5.25 V dc

ENVIRONMENTAL CONDITIONS

Refer to the Basic Micro-Programmable Processor Hardware Maintenance Manual for information on the environmental requirements of the 1700 transform.

OPERATION 2

Once installed, the 1700 transform module requires no operation or programming.

INSTALLATION AND CHECKOUT 3

Information for this section is contained in the CYBER 18 computer systems installation manual and in the systems level CYBER 18 computer systems hardware maintenance manual.

THEORY OF OPERATION 4

The 1700 transform with read-only micro memory is used to emulate the CDC® 1700 computer instruction repertoire when it is combined with the basic micro processor to form the 1700 enhanced processor. The emulation process utilizes both hardware and firmware for more efficient operation.

The firmware consists mainly of many micro-code subroutines that emulate 1700 macro instructions; therefore, it is also called the 1700 emulator. For each 1700 macro instruction, there exists a corresponding subroutine required to emulate it. To start the emulation, the macro instruction is read out from macro memory by a portion of the micro program. The macro instruction is then decoded by hardware; this hardware decoder is called the transform. The transform provides the micro program with the capability to select patterns of bits from the registers and the data transmission path of the processor to form the micromemory address. This micro-memory address selects the appropriate micro-code subroutine to emulate the macro instruction. More than one transform operation may be required to completely emulate a macro instruction. The transform also sets the parameters, generates the micro code needed for the arithmetic and logical operation (refer to MIR Encode) during the emulation process, and sets the contents of the N and K registers.

There are three types of transforms:

- MA transform
- K or N transform
- Combined MA and K transform

The transform commands are coded in the C field of the micro instruction as TMA/j, TN/j, GETMAK/j, and GETMAK/XT. The letter j is decoded from the lower four bits (MIR28 through MIR31) of the micro-instruction register for the MA transform and the lower three bits (MIR29 through MIR31) for the K and N transform. These bits specify the selector position of the MA transform and of the K and N transform. Table 4-1 lists the operations that result when the above transform commands are executed. Figure 4-1 is the block diagram of the 1700 transform module.

TABLE 4-1. TRANSFORM OPERATIONS

Mnemonics	Operation
TMA/j	Obtain mext micro-instruction pair from the address specified by MA transform (selector S5), setting j.
TK/j	Set K register to value specified by K transform (selector S8), setting j.
TN/j	Set N register to value specified by N transform (selector S8), setting j.
TMAK/j	An MA and K transform is executed based on the value of j.
GETMAK/j†	1. Output data from macro memory is gated into the instruction transform (IXT) register. 2. An MA and K transform is executed based on the value of j.
GETMAK/XT†	1. Output data from macro memory is gated into the IXT and IXT' registers. 2. One of eight MA transforms is executed based on the macro instruction loaded into the IXT register (selected from selector S5, positions 8 through 15). The K register is always transformed from S8, position 7. 3. The most significant 16 bits of the micro instruction register (MIR) are loaded with a micro command encoded from the macro instruction residing in the IXT register. This operation is referred to as MIR transform (XT/MIR). The least significant 16 bits of MIR are loaded from micro memory.

†These commands must be executed in the micro instruction following a read command.

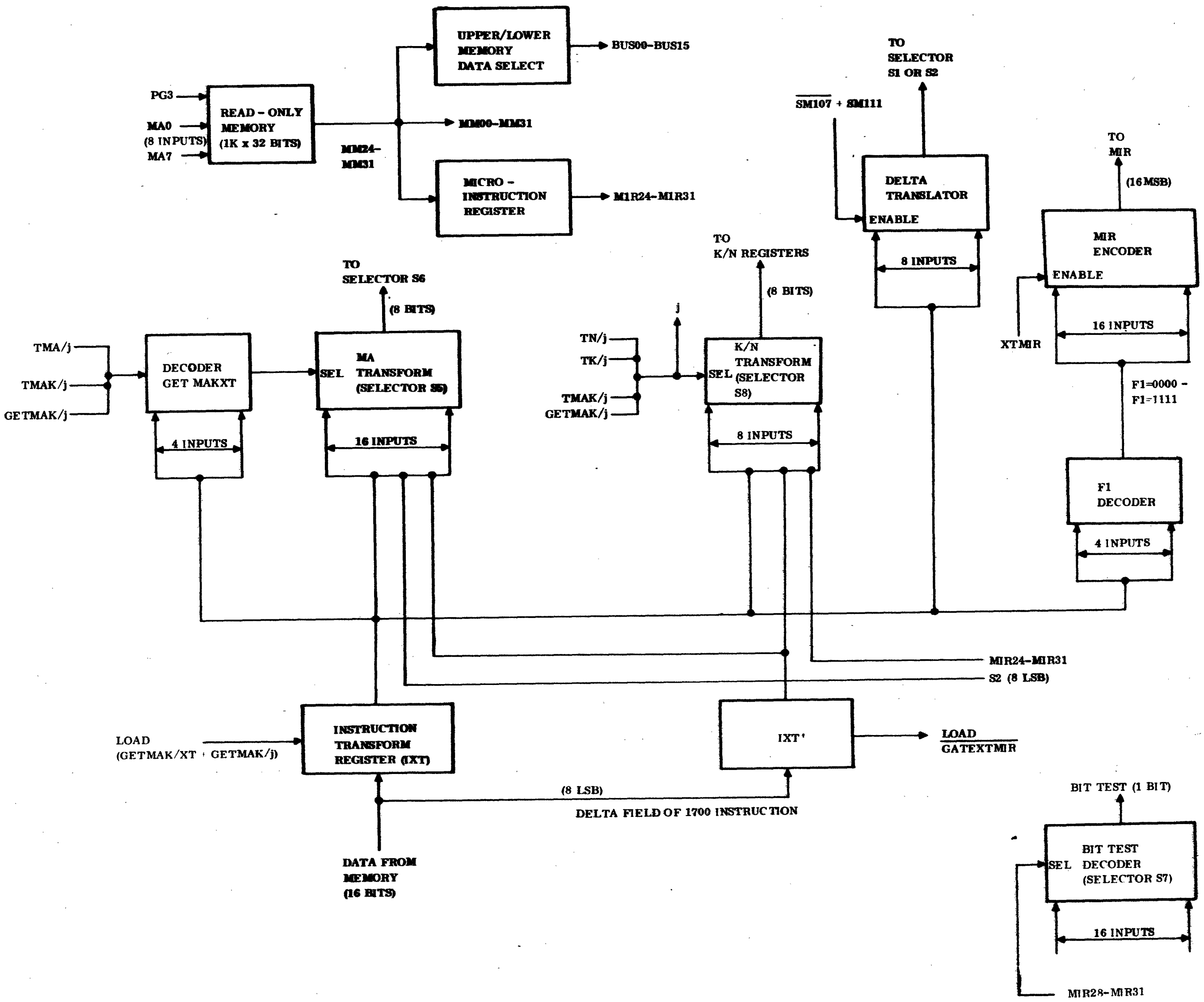


Figure 4-1. 1700 Transform Module Block Diagram

INSTRUCTION TRANSFORM (IXT) REGISTER

This is a 16-bit register that holds the macro instruction currently being emulated. It receives inputs directly from macro (main) memory. The IXT register outputs are sent to the MA Transform, K/N transform, delta translator, and via the F1 decoder to the MIR encoder to be transformed. Other IXT register outputs are also sent to the GETMAK/XT decoder to generate the proper select signals for the MA transform during a GETMAK/XT operation. This register is loaded by executing a macro memory read micro instruction followed by a micro instruction into GETMAK/j or GETMAK/XT in the C" (double prime) field. Otherwise the IXT register can be loaded by executing a micro instruction with C' (prime) equal to 011xxxx to generate a general purpose strobe at time T4.

IXT' REGISTER

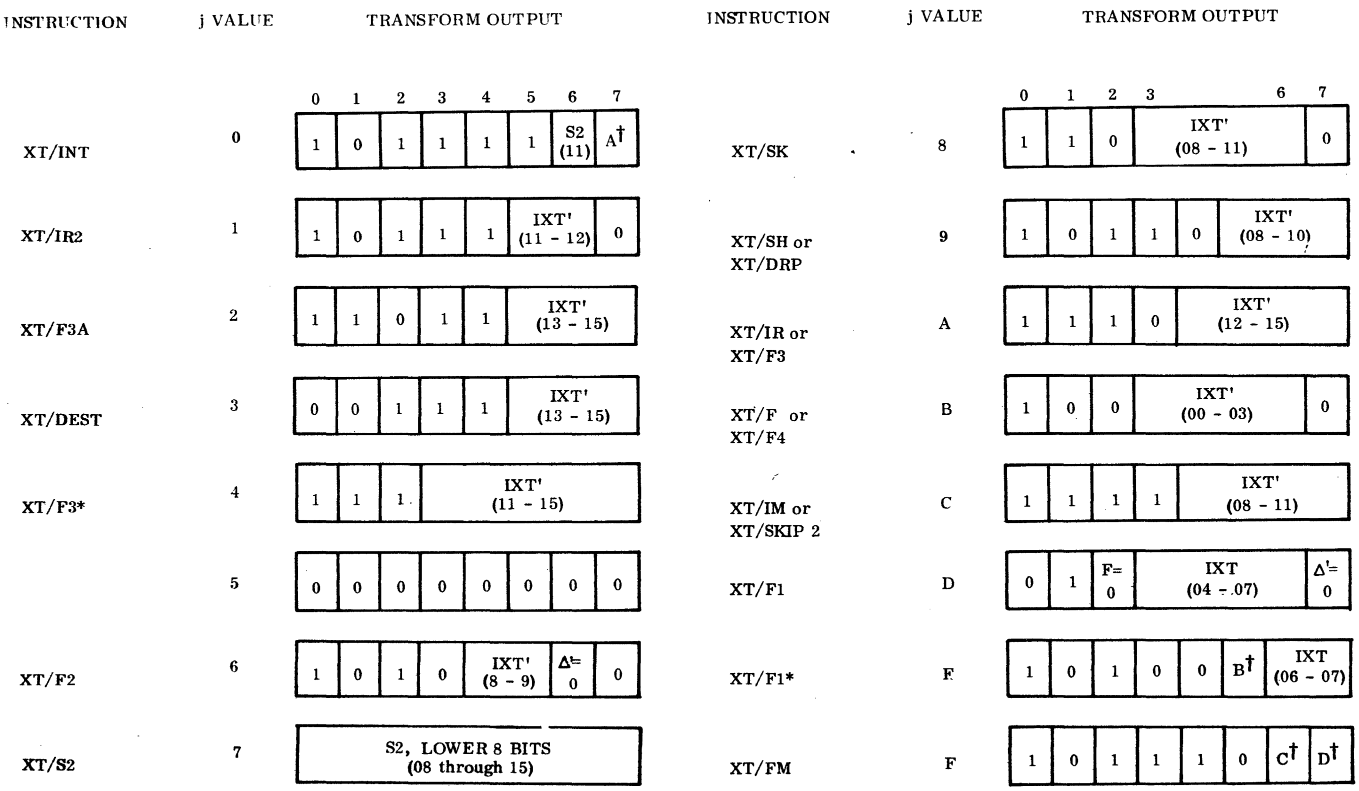
The IXT' register is an 8-bit register that holds the eight least significant bits (delta field) of the 1700 macro instruction. This register provides emulation of the 1700 enhanced instructions that have double word format. Inputs to the IXT' register are obtained directly from macro (main) memory. The macro memory data is gated into this register by application of GATEXTMIR, which is generated only during the GETMAK/XT command. IXT' register outputs are sent to the MA and K/N transforms to be transformed.

MA TRANSFORM

The MA transform (selector S5) is an 8-bit wide selector that is used to form micro-memory addresses. The MA transform provides selection of 16 different micro memory address (MA) trnasforms. These micro memory address transforms specify one of 256 64-bit emulation instructions contained within a micro memory page. The selection of each emulation instruction is determined by the j value of transform commands (TMA/j, TMAK/j, and GETMAK/j) or depends upon the macro instruction via transform hardware (MIR 28 through MIR 31, GETMAK/XT commands).

Figure 4-2 indicates 16 different instruction transforms. Transform selections (j value) 0 through 7 and 9 through C are applicable to emulation of enhanced 1700 instructions. Transform selections 8 through F are applicable to emulation of the basic 1700 instruction set. The patterns of bits are derived from hardwired connections tc +5 V and ground, IXT and IXT' registers, CPU selector S2, and special conditions (such as protect violation or indirect address mode) that are decoded from the macro instruction. Table 4-2 lists the 16 different MA transforms applied for different types of macro instructions and for different addressing modes. Whenever the GETMAK/XT command is executed, one of the eight MA transforms (8 through F) is selected, based on the macro instruction being emulated. Tables 4-3, 4-4, and 4-5 list the MA transforms for the basic 1700 instruction set storage reference (F ≠ 0), register reference, and inter-register reference instructions, respectively. The MA transforms for the enhanced instructions are selected by MA transform of instruction XT/F1 (j value D).

During GETMAK/XT operations, the MA transform select signals are derived from the TMA/j, TMAK/j, and GETMAK/j inputs to the GETMAK/XT decoder. During $\overline{\text{GETMAK/XT}}$ operations the MA transform select signals are derived from the MIR12 through MIR15 inputs obtained form the read only memory.



INSTRUCTION	j VALUE	0	1	2	3	4	5	6	7
XT/INT	0	1	0	1	1	1	1	S2 (11)	A†
XT/IR2	1	1	0	1	1	1	IXT' (11 - 12)		0
XT/F3A	2	1	1	0	1	1	IXT' (13 - 15)		
XT/DEST	3	0	0	1	1	1	IXT' (13 - 15)		
XT/F3*	4	1	1	1	IXT' (11 - 15)				
	5	0	0	0	0	0	0	0	0
XT/F2	6	1	0	1	0	IXT' (8 - 9)		Δ'= 0	0
XT/S2	7	S2, LOWER 8 BITS (08 through 15)							
XT/SK	8	1	1	0	IXT' (08 - 11)				0
XT/SH or XT/DRP	9	1	0	1	1	0	IXT' (08 - 10)		
XT/IR or XT/F3	A	1	1	1	0	IXT' (12 - 15)			
XT/F or XT/F4	B	1	0	0	IXT' (00 - 03)				0
XT/IM or XT/SKIP 2	C	1	1	1	1	IXT' (08 - 11)			
XT/F1	D	0	1	F= 0	IXT (04 - 07)				Δ'= 0
XT/F1*	F	1	0	1	0	0	B†	IXT (06 - 07)	
XT/FM	F	1	0	1	1	1	0	C†	D†

†A = (IXT=0500) + S209
= IIN instruction or false attempt
B = (F1=00xx) + $\overline{\text{MULTILEVEL INDIRECT MODE}}$
C = Protect violation
D = (Δ'= 0) + B

060

Figure 4-2. MA Transforms

TABLE 4-2. MA TRANSFORM APPLICATIONS

Instruction	MIR28-MIR31 = j	Application
XT/INT	0	Micro/macro interrupt
XT/IR2	1	Inter-register type 2 instruction
XT/F3A	2	Field instruction
XT/DEST	3	Register destination
XT/F3*	4	Miscellaneous instruction
	5	Not used
XT/F2	6	F2 (address mode) for enhanced instruction
XT/S2	7	Selector S2 (lower eight bits); normally used for the breakpoint panel
XT/SK	8	Skip instruction
XT/SH or XT/DRP	9	Shift instruction, or decrement and repeat instruction
XT/IR or XT/F3	A	Inter-register instruction with M not the origin, or miscellaneous instruction
XT/F or XT/F4	B	F (OP CODE) field, or OP CODE for storage reference type 2 and field instruction
XT/IM or XT/SKIP2	C	Inter-register with M origin, or skip isntruction type 2
XT/F1	D	F1 (address mode) field
XT/F1*	E	Alternate F1 field
XT/FM	F	Miscellaneous F1 field

TABLE 4-3. 1700 STORAGE REFERENCE TRANSFORMS DURING GETMAK/XT OPERATION

Mode	F1 (Binary)	Hexadecimal	Delta	Instruction	MIR Transform
Absolute	0000	0	≠0	XT/F	Δ → X,AB
Constant			=0	XT/F1	P + 1 → P, AB
Absolute	0001	1	≠0	XT/F	Δ + (00FF) → X, AB
Constant			=0	XT/F1*	P + 1 → P, AB
Absolute	0010	2	≠0	XT/F	Δ + (Q) → X, AB
Constant			=0	XT/F1*	P + 1 → P, AB
Absolute	0011	3	≠0	XT/FM	Δ + (00FF) → X, AB
Constant			=0	XT/F1*	P + 1 → P, AB
Indirect	0100	4	≠0	XT/F1*	Δ → X, AB
Storage			=0	XT/F1*	P + 1 → P, AB

TABLE 4-3. 1700 STORAGE REFERENCE TRANSFORMS DURING GETMAK/XT OPERATION (Contd)

Mode	F1 (Binary)	Hexadecimal	Delta	Instruction	MIR Transform
Indirect Storage	0101	5	≠0 =0	XT/F1* XT/F1*	Δ →X, AB P + 1 → P, AB
Indirect Storage	0110	6	≠0 =0	XT/F1* XT/F1*	Δ → X, AB P + 1 → P, AB
Indirect Storage	0111	7	≠0 =0	XT/F1* XT/F1*	Δ → X, AB P + 1 → P, AB
Relative 16-bit relative	1000	8	≠0 =0	XT/F XT/F1	P + Δ(SE) → X, AB P + 1 → P, AB
Relative 16-bit relative	1001	9	≠0 =0	XT/F1 XT/F1	P + Δ(SE) → X, AB P + 1 →P, AB
Relative 16-bit relative	1010	A	≠0 =0	XT/F1 XT/F1	P + Δ(SE) →X, AB P + 1 → P, AB
Relative 16-bit relative	1011	B	≠0 =0	XT/F1 XT/F1	P + Δ(SE) → X, AB P + 1→ P, AB
Relative indirect Relative indirect	1100	C	≠0 =0	XT/F1* XT/FM	P + Δ(SE) → X, AB P + 1 → P, AB
Relative indirect Relative indirect	1101	D	≠0 =0	XT/F1* XT/FM	P + Δ(SE) → X, AB P + 1 → P, AB
Relative indirect Relative indirect	1110	E	≠0 =0	XT/F1* XT/FM	P + Δ(SE) → X, AB P + 1 →P, AB
Relative indirect Relative indirect	1111	F	≠0 =0	XT/F1* XT/FM	P + Δ(SE) →X, AB P + 1 →P, AB

TABLE 4-4. 1700 REGISTER REFERENCE TRANSFORMS DURING GETMAK/XT OPERATION

F1 (Binary)	Instruction	MIR Transform	Comment
0000	XT/F1	NOP	Selective stop (Δ=0) Instruction enhanced (Δ≠0)
0001	XT/SK	P + 1 → P, AB	Skip
0010	XT/F1	P + Δ(SE) → F, AB	Input to A
0011	XT/F1	P + Δ(SE) → F, AB	Output from A
0100	XT/F1	NOP	Enable interrupt (Δ=0) Instruction enhanced (Δ≠0)
0101	XT/F1	NOP	Inhibit interrupt (Δ=0) Instruction enhanced (Δ≠0)

TABLE 4-4. 1700 REGISTER REFERENCE TRANSFORMS DURING GETMAK/XT OPERATION (Contd)

F1 (Binary)	Instruction	MIR Transform	Comment
0110	XT/F1	Q → X, AB	Set program protect (Δ=0) Instruction enhanced (Δ≠0)
0111	XT/F1	Q → X, AB	Clear program protect (Δ=0) Instruction enhanced (Δ≠0)
1000	†	†	Inter-register
1001	XT/F1	P + 1 → P, AB	Increase A
1010	XT/F1	P + 1 → P, AB	Enter A
1011	XT/F1	NOP	Pass (Δ=0) Instruction enhanced (Δ≠0)
1100	XT/F1	P + 1 → P, AB	Enter Q
1101	XT/F1	P + 1 → P, AB	Increase Q
1110	XT/F1	Δ(INT) → X, AB	Exit interrupt
1111	XT/SH	A → F	Shift

†See 1700 Inter-Register Transforms, table 4-5.

TABLE 4-5. 1700 INTER-REGISTER TRANSFORMS DURING GETMAK/XT OPERATION

Instruction	Condition
XT/IR	M not origin register (I12 ≠ 0)
XT/IM	M is origin register (I12 = 0)

MIR Transform (MIR Fields)				Conditions							
						Origin			Destination		
F Bits 2-6	A Bits 7-9	B Bits 10-12	D Bits 13-15	LP I8	XR I9	A I10	Q I11	M I12	A I13	Q I14	M I15
ADDT 11001	-	-	-	0	0	X	X	0	X	X	X
A B 01110	-	-	-	1	0	X	X	0	X	X	X
A + B 01001	-	-	-	0	1	X	X	0	X	X	X
(-A) + (-B) 00001	-	-	-	1	1	X	X	0	X	X	X

TABLE 4-5. 1700 INTER-REGISTER TRANSFORMS DURING GETMAK/XT OPERATION (Contd)

Instruction	Condition
XT/IR	M not origin register (I12 ≠ 0)
XT/IM	M is origin register (I12 = 0)

MIR Transform (MIR Fields)				Conditions							
						Origin			Destination		
F Bits 2-6	A Bits 7-9	B Bits 10-12	D Bits 13-15	LP I8	XR I9	A I10	Q I11	M I12	A I13	Q I14	M I15
ADD+ 11010	P register 001	Zeros 001	P register[†] 001	X	X	X	X	1	X	X	X
-	Ones 110	-	-	X	X	0	X	0	X	X	X
-	A register 100	-	-	X	X	1	X	0	X	X	X
-	-	Ones 110	-	X	X	X	0	0	X	X	X
-	-	Q register 100	-	X	X	X	1	0	X	X	X
-	-	-	NOP 000	X	X	X	X	0	0	0	0
-	-	-	A register[†] 101	X	X	X	X	0	1	X	X
-	-	-	Q register[†] 011	X	X	X	X	0	0	1	X
-	-	-	F register 111	X	X	X	X	0	0	0	X

[†] NOP if protect violation detected.

K/N TRANSFORM

The K/N transform (selector S8) is an 8-bit wide selector that chooses one of eight instruction transforms to be loaded into the CPU K and N registers. Figure 4-3 indicates the eight K/N transform assignments. Bit patterns for creation of the transform outputs are derived from direct ground connections, eight bits from CPU selector S2 (S208 through S215), IXT and IXT' registers, and eight bits from the CPU micro instruction register (MIR24 through MIR31). The K/N transform is enabled by receipt of an S8ENABLE/ (low). The S8ENABLE/ (high) disables the K/N transform during the clear K register, clear N register, and clear N and page register commands. These clear commands allow all zeros to be loaded into the respective (K or N) register.

BIT TEST DECODER

The bit test decoder (selector S7) is a 1-bit wide selector that provides for up to 16 different conditions (external/-internal) to be tested. These tests determine whether the upper or lower micro instruction shall be executed from the next micro-instruction pair. The test bit is selected by the least significant four bits of the micro-instruction register (MIR28 through MIR31). The bit test output is sent to the T-field test multiplexer in the control 2 module and is tested if the T field of the micro instruction contains a BTU command. Table 4-6 list the conditions to be tested by the 1700 emulator during the emulation process.

MIR ENCODE

During GETMAK/XT command, the macro instruction is encoded from the F1 bits of the 1700 instruction. the F1 bit values select the various configuration of the upper 16 bits (MM00 through MM15) that are loaded into the CPU MIR register. These micro memory instructions control the arithmetic functions for read next instruction cycles and other required operations to provide efficient execution. The various types of MIR transforms, based on the macro instruction being emulated, are listed in tables 4-3, 4-4, and 4-5 for storage, register, and inter-register reference

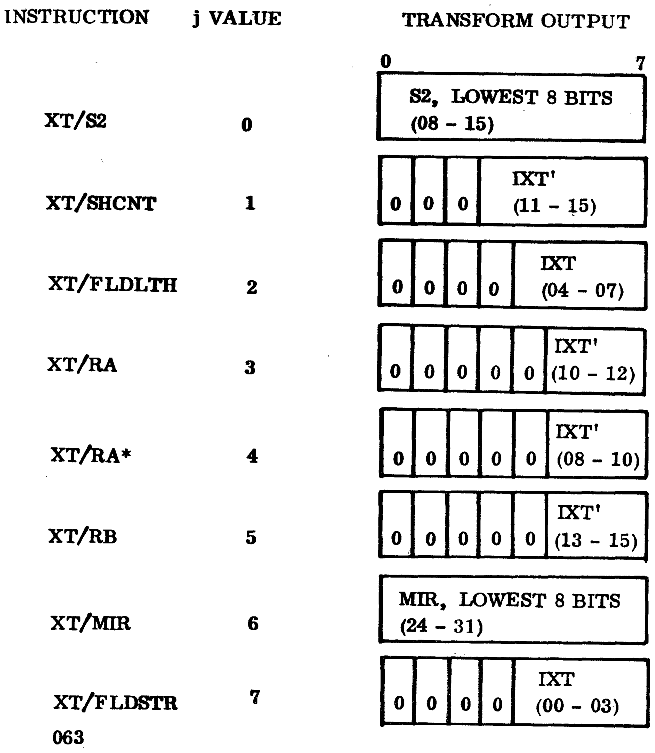
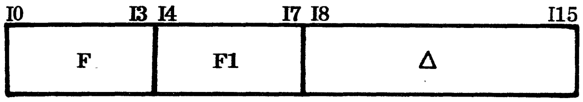


Figure 4-3. K/N Transforms

instructions, respectively. Figure 5-4 shows the MIR transform selections for storage reference instructions, and figure 5-5 shows the MIR transform selections for register reference instructions. These figures illustrate the selection conditions and not the sequential steps that select the MIR transform.

DELTA TRANSLATOR

To emulate certain basic and enhanced 1700 instructions, the delta field of the macro instruction figure 4-4 must be modified:



This modification is necessary before the delta field can be processed for operations indicated by MIR transforms. Table 4-7 lists the conditions and modified delta field.

The delta translator translates the delta field according to the type of macro instructions being emulated as indicated in table 4-4. The delta translator is enabled by conditions of SM107, SM111, F field, and F1 field:

- SM107 is not set (decimal arithmetic correction logic is disabled).
- SM111 is not set (the CPU file F1 output to selectors S1 and S2 is disabled and delta translator output to selectors S1 and S2 is enabled).
- F = 0 and F1 = 1000 are low (the inter-register reference instruction is not selected).

When an inter-register reference instruction is emulated, the delta translator is disabled. This causes the delta ($FFFF_{16}$) to be sent to selector S1 and S2.

READ-ONLY MICRO MEMORY

The micro memory of the 1700 transform module is a read-only memory that has been preprogrammed with the 1700 instruction emulator. This micro memory consists of 512 64-bit words (two pages). Each word consists of two micro instructions that are referred to as upper (32 bit) and lower (32 bit). Each word in the micro memory is addressed by the memory address bits (MA0 through MA7, PG3). The MA0 through MA7 specify one of 256 words (micro instruction pairs, 32-bit upper and 32-bit lower instruction within a page. The page (PG3) selects the page (page 0 or 1) in which the instruction resides. The output of the read-only memory is coupled to the upper/lower memory data select where the sequence of selection (upper or lower) is determined in accordance with the emulation required. The selected instruction is transferred via the CPU three-state bus to the X register.

1700 EMULATOR

The 1700 emulator is the firmware program that emulates both the basic and enhanced 1700 computer instruction set. The emulator also contains firmware for handling I/O operations, macro and micro interrupts, auto-data transfer (ADT) operations, and panel simulation. The 1700 emulator consists of micro-memory subroutines that are preprogrammed in the read-only memory. These subroutines are completely contained in the two pages (0 and 1, 1024 32-bit micro instructions) micro-memory. The 1700 emulator consists of many micro-code subroutines. Each subroutine micro code performs a specific task, such as generating and controlling the operations required to emulate a macro instruction. For example, the micro-code subroutine titled the decode-next-instruction subroutine is executed at the beginning of every new macro-instruction emulation. This subroutine resides in micro location 058_{16} and 059_{16} and includes the following micro instructions.

TABLE 4-6. EMULATION TEST CONDITIONS

Test Bit	Operation	Selector S7	
		Pin	Position
BTU00	Not assigned	8	0
I02	Execute upper micro instruction if I02 is a 1.	7	1
I07	Execute upper micro instruction if I07 is a 1.	6	2
I06	Execute upper micro instruction if I06 is a 1.	5	3
IND00FF	Execute upper micro instruction if STORE 00FF (index 1) status is true.	4	4
SM105/ $\overline{\text{(PROTECT FAULT}}$)	Execute lower micro instruction if storage protect fault is detected.	3	5
SELSTOP	Execute lower micro instruction if selective stop switch is set.	2	6
SELSKIP/	Execute lower micro instruction if selective skip switch is set.	1	7
SM108 (PARITY ERROR)	Execute lower micro instruction if storage parity error is detected.	23	8
BTU00	Not assigned	22	9
DELTA'=0	Execute upper micro instruction if delta equals 0 (LXT8 through IXT15 = 0)	21	10
$\overline{\text{EA=OPER}}$	Execute lower micro instruction if the effective address equals the operand.	20	11
EVENPAR	Execute upper micro instruction if memory parity line is true (even parity).	19	12
I00	Execute upper micro instruction if I00 is a 1.	18	13
MULTIND	Execute upper micro instruction if multilevel indirect address mode is selected.	17	14
SM105+SM108	Execute upper micro instruction if previous macro memory write cycle was aborted (caused either by parity error or protect fault).	16	15

- Increment to next instruction (INI)
- Read the next instruction (RNI) and check for interrupt. Go to process micro/macro interrupt subroutine if an interrupt occurred
- Transform $\overline{\text{GETMAK/XT}}$ on the next instruction (XNI) if there is no interrupt

At the end of every macro-instruction emulation, a jump command is used to branch to specific locations in the decode-next-instruction subroutine to start the emulation of the next macro instruction.

MISCELLANEOUS CIRCUITRY

PROTECT VIOLATION DETECTING CIRCUIT

To emulate the 1700 series computer protect function, a combination of hardware and firmware (1700 emulator) is

TABLE 4-7. DELTA TRANSLATIONS

Conditions	Delta (Δ)
a. (F=0) (F1 = 0xxx) b. Enhanced instructions: (F=0) (F1 = 4 + 5) (r = 0)[†]	Δ = 0 0 0 0 0 0 0 0 I08 I09 I10 I11 I12 I13 I14 I15
a. (F=0) (F1 = 1xxx) b. (F=0) (F1= 2 + 3) c. Enhanced instructions: (F=0) (F1 = 4 + 5) (r = 0)[†]	Δ(SE) (with sign extend) = $\underbrace{C\ C\ C\ C\ C\ C\ C\ C}_{\text{Constant = I08}}$ I08 I09 I10 I11 I12 I13 I14 I15
a. (F=0) (F1=1) b. (F=0) (F1= 0 + 6)	Δ(SK) (for skip instruction) = 0 0 0 0 0 0 0 0 0 0 0 0 I12 I13 I14 I15
a. (F=0) (F1=E)	Δ(INT) (for interrupt instruction = 0 0 0 0 0 0 0 1 I08 I09 I10 I11 I12 I13 I14 I15
a. (F=0) (F1=8)	Δ(FFFF) = 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

[†] Refer to the CYBER 18 processor reference manual for type 2 storage reference instructions, enhanced instructions, and field reference isntructions. Flag r is the relative address flag represented by IXT'08.

required. The 1700 computer incorporates a program protect function that inhibits access into protected programs by nonprotected programs. The function develops around a memory protect bit that is contained in each word of macro (main) memory. The protect violation conditions detected by the 1700 transform hardware are:

- An attempt to execute instructions that are not protected
- An attempt to execute a protected instruction following the execution of a nonprotected instruction
- An attempt to execute an unprotected miscellaneous instruction with the protect switch set

All other violations, such as an attempt by a nonprotected instruction to write into a protected storage location, are detected by the macro memory interface hardware and processor hardware.

INTERRUPT ENABLE

The 1700 computer systems require that the interrupt function be activated after on instruction has been executed following the enable interrupt instruction (EIN). A combination of hardware and firmware is required to prevent the interrupt function from being activated during the execution of the instruction following EIN if the instruction is either:

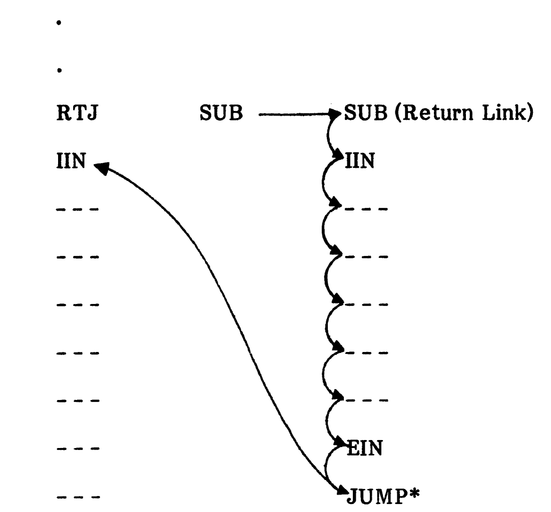
- A storage reference instruction with multilevel indirect address mode that requires more than one memory reference (more than one RNI cycle)
- An enhanced instruction with double-word format.

The implementation of the interrupt function is described below.

Status mode bit SM114 is first set by the 1700 emulator whenever the EIN is emulated to pre-enable the interrupt function. If the instruction following EIN is neither a storage reference instruction (F = 0) nor an enhanced instruction (except a type 2 skip instruction), the hardware enables the SETSM106 at the next RNI cycle which in turn sets SM106 to enable the CPU interrupt function. If the instruction following EIN is either a storage reference or enhanced instruction, SM106 is set after the 1700 emulator has executed a SUB- operation (increment P counter). The SUB- operation indicates the end of the macro instruction emulation. Once status mode bit SM106 is set, SM114 is cleared by the hardware.

1700 series computers also require that the inhibit interrupt instruction (IIN) tape precedence over enable interrupt instructions (EIN). If the IIN instruction is executed after one instruction following an EIN (as in the following example) the macro interrupt function must be disabled.

Main Program



The 1700 emulator implements the above example as follows: whenever an interrupt is detected, the interrupt transform (XT/INT, MA transform 0) is performed. The XT/INT transform forms the micro memory address of BC$_{16}$ of BE$_{16}$ for the micro interrupt (S211 = 0) or macro interrupt (S211 = 1), respectively. If the interrupt is a macro interrupt, the emulator ignores the macro interrupt and rereads the macro instruction. (This has the same effect as disabling the macro interrupt function.) If the interrupt is a micro interrupt, the emulator processes the micro interrupt as normal.

If the macro interrupt being emulated is not an IIN instruction and there is no false interrupt, the XT/INT transform forms micro memory address BD$_{16}$ or BF$_{16}$ for the micro interrupt or macro interrupt, respectively. The interrupt is then processed accordingly by the micro or macro interrupt subroutine residing at the above address.

XTBLKT4 SIGNAL GENERATION

During any operation except GETMAK/XT, the destination is decoded and strobed at time T4. However, during a GETMAK/XT transform operation, the previously read data is strobed into the destination register by the trailing edge of RESUME the read macro memory command. (Refer to the processor manuals for more details.) This allows the correct delta field to be used in emulation of the macro instruction.

BLKM100 SIGNAL GENERATION

During macro step mode, the macro halt interrupt must be blocked while either a memory reference instruction or an enhanced instruction is emulated; this instruction is enabled only at the end of instruction execution. The following example with step- by-step operation of the emulator shows the need for the block mask bit 100 (BLKM100/) signal.

Assume that the machine is in step mode and is idling.

1. A macro GO command is executed (enter I: with function control register bit 12 clear).
2. The GO command sets SM215 (clear macro halt interrupt).
3. The RNI cycle is executed to read the instruction. Assume that the instruction is a memory reference instruction with multilevel indirect addressing mode. Check for interrupt. Since SM215 is set, there is no macro halt interrupt.
4. Perform the GETMAK/XT transform operation.
5. Hardware clears SM215 of either the breakpoint controller or the I/O-TTY mode whenever the CPU is in step mode and GETMAK/XT is executed. This sets the macro halt interrupt (enabled by mask bit M100).
6. The emulator executes a multilevel indirect address subroutine to look for the effective address. The interrupt is also checked.
7. Since the macro interrupt is set, the emulator branches to the interrupt subroutine and the instruction cannot be completed.

To avoid the problem in step 7, BLKM100/ must be generated to block the macro halt interrupt from being set to step 5. BLKM100/ is set low during a GETMAK/XT operation whenever a memory reference instruction or an enhanced instruction is decoded.

At the end of the above macro instruction emulation, the emulator executes a SUB- operation (increment the P counter), which sets the BLKM100/ signal high. BLKM100/ is also set high by the master clear.

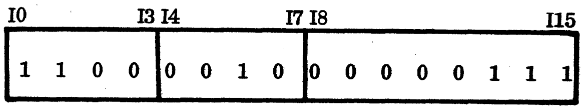
TYPICAL 1700 MACRO INSTRUCTION EMULATION

The following 1700 macro instruction, load A register, is selected to demonstrate the transform operation and the emulation of a macro instruction.

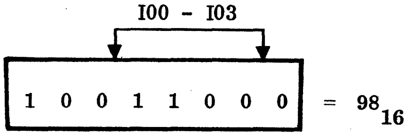
I0 I3	I4 I7	I8 I15
F = C	F1 = 2	Δ = 07

The above macro instruction loads the A register with the contents of the storage location specified by the effective address (EA). EA is Δ + (Q). Figure 4-4 shows the step-by-step emulation of this macro instruction in the form of a flow chart. Figure 4-5 lists the micro-code subroutines required to emulate the instruction. First, the above instruction must be read out from macro memory using the read next instruction (RNI) cycle. Refer to the decode-next-instruction subroutine at location 058_{16}. The interrupt is also checked. Assuming that there is no interrupt at this time, then the transform next instruction (XNI) cycle (the GETMAK/XT command) is executed. Execution of the GETMAK/XT command causes the following sequence of events.

1. Data from memory is loaded into the IXT and IXT' registers. The IXT register now contains:



2. The output of the IXT register is decoded into j value select signals to select the MA transform. Refer to figure 5-2 for MA transform selection. For the macro instruction used in this example, the XT/F MA transform is selected since j value equals B_{16}. The output of the MA transform becomes:



which is then applied to CPU selector S6 to designate the new micro-memory address (MA). The new micro-memory address, 98_{16}, points to the load-A-register subroutine of the 1700 emulator.

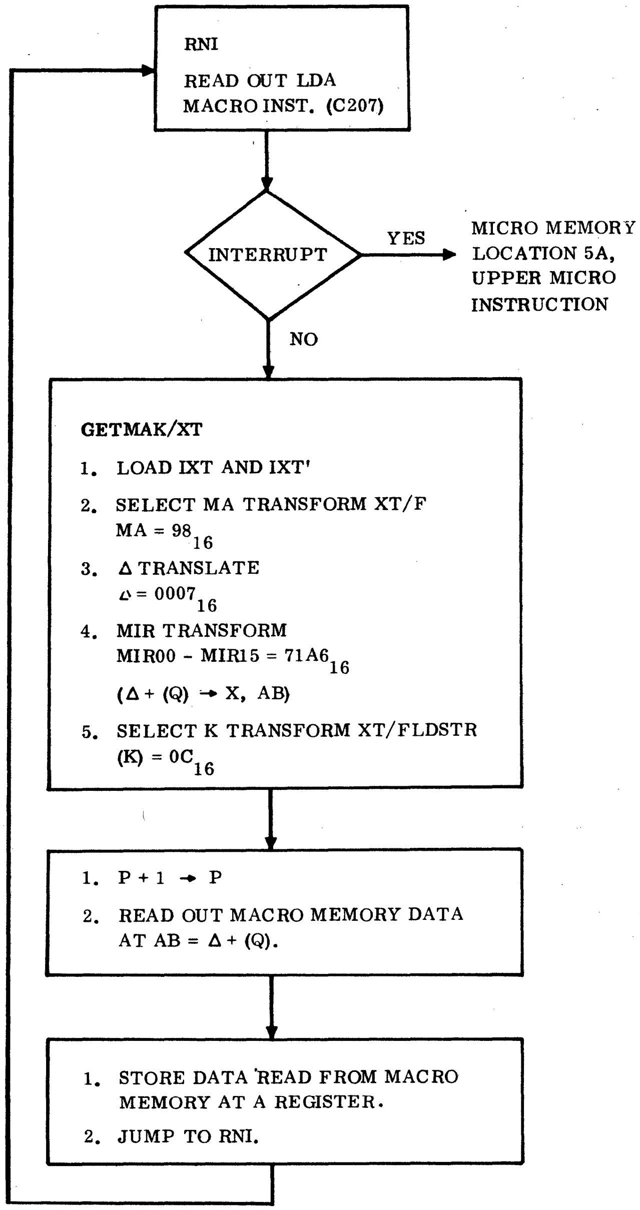
3. If required, the outputs of the IXT and IXT' registers are also sent out to the delta translator circuit to be modified. In this example the output of the delta translator contains the following:

```
Δ = 0 0 0 0 0 0 0 0 I08 I09 I10 I11 I12 I13 I14 I15
    0 0 0 0 0 0 0 0 0    0   0   0   0   0   0   0
```

Refer to table 4-7 for more detail.

4. Simultaneously with the operations in step 2 and 3, the output of IXT is encoded to form the upper 16 bits (MM00 through MM15) of micro memory. Refer to figure 5-4.

During the GETMAK/XT operation, the read-only memory chips containing the upper 16 bits of micro memory data are disabled to allow the output of the MIR encoder to be loaded into MIR at time T5.



067

Figure 4-4. Step-by-Step Emulation of Macro Instruction LDA

```
T  P/MA  MICRO-MEM LOCATION  F   A      B     D   S     C            MT    COMMENT

                    DECODE NEXT INSTRUCTION

                                                                           INCR. TO NEXT
0  058   6C79  0000  B  INI   SUB-  P   MEM   P                            INST., P = P + 1
1  058   D8D8  B307  G  -RNI                      READ  N=F2IREG     INTU  READ NEXT INST.
0  059   4AF7  2830  B  +     -B        INTA  F           GATE IXT   U     PROCESS ACTIVE
                                                                           INTERRUPT
1  059   58D8  302C  C  -XNI                      PAGE0 GITMAKXT     U     TRANSFORM ON
                                                                           NEXT INST.
                               •
                               •
                               •
                               •

                    LOAD A REGISTER   F = C

0  098   6C79  2300  G  +LDA  SUB-  P   MEM   P   READ               U     READ (EA),
                                                                           P + 1 TO P
0  099   9FDD  4058     +     A     MEM       A         RNI          J     (EA) TO A,
                                                                           GO TO RNI
```

Figure 4-5. Micro-Code Subroutines to Emulate LDA Micro Instructions

The upper 16 bits of MIR now contain:

MIR00															MIR15
0	1	1	1	0	0	0	1	1	0	1	0	0	1	1	0
M		F					A			B			D		

Where: F = 11000 indicates that this is an A+B operation

A = 110 selects the output of file 1 as the A input to the ALU. Since status mode bit SM111 is clear, the output of the delta translator is used as the A input.

B = 100 selects the Q register as the B input to the ALU.

D = 110 transfers the results of the add operation to X and the macro memory address AB registers.

When executed, the above 16 bits form the effective address Δ + (Q).

5. K transform 7 (XT/FLDSTR) is always selected during the GETMAK/XT operation (that is, the K register is loaded with $0C_{16}$). However, since SM113 is not set by the emulator, the contents of the K register is not used for emulation of this instruction.

After the GETMAK/XT transform operation is completed, the upper micro instruction of the load A register subroutine at location 98_{16} is loaded into the MIR and executed. From this micro instruction, the P counter is incremented by 1 and data from macro memory at address Δ + (Q), determined by the content of the macro memory address register, AB, is read out. The upper micro instruction is executed next. From this micro instruction, the data read from macro memory is loaded into the A register and a jump is made to the RNI cycle. This completes the emulation of the load A register (LDA) macro instruction.

EXTERNAL INTERFACE

Table 4-8 provides a glossary of terms for external interface signals. This glossary tabulates the mnemonics found on the logic diagrams the term that the mnemonic represents, and the application of the signal(s).

The field print package logic diagrams contain diagonal (/) and bar (‾‾) symbols to indicate active low conditions. The diagonal is used with external signals and the bar is used with internal signals.

TABLE 4-8. GLOSSARY OF TERMS (EXTERNAL SIGNALS)

Mnemonic	Term	Application
BLKM100	**Block mask interrupt bit 100**	Disables emulator branch to interrupt subroutine during macro halt
BTU	**Bit test unit**	One bit used to modify any CPU word
BUS00 - BUS15	**Bus lines**	Three-state bus lines of CPU
CARRYOUT 0 - CARRYOUT 3	**Carry-out bits**	Carry output from ALU of processor
CLRMIR	**Clear memory instruction register**	Clear signal to clear transform MIR
CPU PROT	**Central processing unit protect**	Discrete signal that indicates that CPU is in a protect status; only protected instructions will be processed
DELTA00 - DELTA15	**Delta translator bits**	Delta translator output lines to selector 1 of CPU
DFM01 - DFM16	**Data from memory bits**	Macro memory word bits
ENMML ENMMLB	**Enable micro-memory lower bits**	Signal used to enable the output of lower micro-memory bits MM00 - MM15
ENMMU ENMMUB	**Enable micro-memory upper bits**	Signal used to enable the output of upper micro-memory bits MM16 - MM31
EVENPAR	**Even parity**	Signal used to test for even or odd parity
GATEAB	**CPU memory address register clock**	Clocks store 00FF to bit test if store status is true
GATEIXT	**Gate instruction transform**	Gates macro memory bits into IXT register
GATEIXT-	**Gate instruction transform prime**	Gates macro memory bits DFM01 - DFM08 into IXT' register
GATEMIR	**Gate micro instruction register**	Clocks the micro memory bit MM24 - MM31 into the micro instruction register of the 1700 transform
GETMAK	**Gate transform of MA and K registers**	Gates macro memory to IXT register and performs transform of MA and K registers
ILLCHAR0 - ILLCHAR3	**Illegal character**	Determines selection of delta translator outputs
IND00FF	**Indicator of ALU equals 00FF**	Enables execution of upper micro instruction
MA0 - MA7	**Memory address bits**	Selects one of 256 micro-memory instructions residing in either page 0 or 1
MCDELAYED	**Delayed master clear**	Clears transform protect bit and clocks the block mask bit 100 signal
MEMPROTBT	**Memory protect bit**	Enables the transform protect function
MODE11	**Sequential address mode**	Determines selection of K/N transforms
MM00 - MM31	**Micro-memory bits**	Micro-memory bits from read only memory that contain the emulation instruction

TABLE 4-8. GLOSSARY OF TERMS (EXTERNAL SIGNALS) (Contd)

Mnemonic	Term	Application
PG0 - PG3	Page bits	Enables selection of upper or lower micro-memory instruction
POWERFAIL	Power failure	Activates program interrupt to inhibit processing if a power failure occurs
PROTECT	Protect switch input	Activates the program protect function
READCYCLE	Read cycle	Enables the transform memory instruction register
RPINT16	Program interrupt 16	Enables micro-interrupt control programs
SELGETMAK	Select GETMAK	Selects gate transform of MA and K registers
SELSKIP	Select skip	Enables skip operation
SELSTOP	Select stop	Activates stop condition
SETSM106	Set status mode bit 106	Enables interrupt function
SETSM209	Sets status mode bit 209	Sets protect fault detection function
SM105	Protect fault	Set if a protect violation occurred
SM108	Memory parity error	Set if a macro memory parity error exists
SM111	Enable F1	Enables transform input to CPU selector S1 and S2 when low.
SM114	Pre-enable interrupt function	Enables interrupt function by permitting interrupt bit SM106 to be set.
SM209	Protect fault detection bit	Enables protect fault detection
SM213	Delta translator select bit	Enables delta translator selection conditions 0 and 1.
S208 - S215	CPU selector 2 inputs	Enables breakpoint selections
S5-0 - S5-7	Transform selector S5 bits	Memory address register transform selection
S8-0 - S8-7	Transfer selector S8 bits	K and N register transform selection
S8ENABLE	K/N transform enable	Enables K/N transform selector S8
T3	Time T3 from CPU	Clock pulse T3 to enable destination register selection by T4 pulse
XTBLKT4	Transfer block destination timing pulse T4	Block selection of destination register at time T4
XTPAGE0	Transform page 0	Enable additional read-only memory if included in system

DIAGRAMS 5

1700 TRANSFORM FUNCTIONAL BLOCK DIAGRAM

Each block of figure 5-1 contains one or more numbers in the upper right corner. These numbers correspond to the logic diagram sheet where the function is located. Input and output signals, including the control signals to each block, are also included. This block diagram supplements the logic diagrams.

INSTRUCTION TRANSFORM (IXT) REGISTER

This register is a 16-bit register that holds the macro instruction currently being emulated. The register consists of D-type flip-flops A3, J2, C5, and B5. The macro instruction residing in the register is received from main memory via lines DFM01 through DFM16 (DFM01 is the least significant bit and DFM16 is the most significant bit). The outputs from the IXT register, I00 through I15 (I00 is the least significant bit and I15 is the most significant bit), are coupled to the MA transform (selector S5), K/N transform (selector S8), delta translator, and F1 and GETMAK/XT decoders. Data-from-memory (DFM) bits are transferred through the associated register elements when the gate instruction transform (GATEIXT/) and gate instruction transform prime (GATEIXT-1) are active (high). When both are active, the bit condition at input D is placed on the output lines by the leading edge (low to high transition) of the gate pulse ($\overline{\text{IXT + IXT'}}$).

IXT' REGISTER

This is an 8-bit register that holds the eight least significant bits of the macro instruction (DFM00 through DFM08) from main memory. The register consists of D-type flip-flops B4 and C4. The data present at the D inputs is placed on the output lines (IXT'08 through IXT'15) on the leading edge of the $\overline{\text{GATEXTMIR}}$ signal. This signal is only generated during the GETMAK/XT command. The outputs are coupled to the MA and K/N transforms (S5 and S8 respectively).

MA TRANSFORM

The MA transform (selector S5) is an 8-bit wide selector that generates micro-memory addresses. This selector consists of eight 16-to-1 multiplexers L2, L3, L4, L5, L7, L9, L10, and L12. These multiplexers permit 16 different micro-memory address (MA) transforms to be specified (figure 4-2). The 8-bit output S5-0 through S5-7 specifies one of 256 64-bit micro memory instructions residing within each page of read-only memory. Selection of the MA transforms is determined by the command (TMA/j, TMAK/j, and GETMAK/j) j value, or derived from the macro instruction during GETMAK/XT commands.

During TMA/j, TMAK/j, and GETMAK/j transform operations, select signals S5-S0 through S5-S3 directly correspond to MIR28 through MIR31 that are derived from the micro memory bits (MM28 through MM31). During the GETMAK/XT transform operations, select signals S5-S0 through S5-S3 are generated based on the macro instruction being emulated. Multiplexer K12 selects one of the above cases depending upon the high or low state of the GETMAK/XT signal applied to K12-1. During GETMAK/XT operations, the output of the IXT register is first coded to select the MA transforms for storage reference instructions (figure 5-2) when F ≠ 0 and then for register and inter-register reference instructions (figure 5-3, table 4-5) when F = 0. The F = 0 signal applied to pin 1 of multiplexer D10 selects the MA transforms as follows:

F ≠ 0 S5-S0 through S5-S3 are generated from the storage reference instructions

F = 0 S5-S0 through S5-S3 are generated from the register reference and inter-register reference instructions

The logical equations for MA transform selections are contained in appendix A.

The MA transform selection shown in figure 5-3 is performed simultaneously by the combination circuit. These flow charts indicate the selection conditions rather than the sequential steps in selecting the MA transforms.

K/N TRANSFORMS

The K/N transforms (selector S8) consists of eight 8-to-1 multiplexers K1, K2, K3, K4, K5, K6, K7, and K9 that are enabled by a low state of the S8ENABLE/ applied to pin 7 of each multiplexer. The 8 bit output of selector S8 is used to choose the sources for loading the K/N transform assignments (figure 4-3). The high state of S8ENABLE/ disables S8 during the clear N register (CLRN), clear K register (CLRK), and clear N and page register (CLRNP) commands. These clear commands allow all 0s to be loaded into the N or K register. The input selection signals S8-S0 through S8-S2 are derived from MIR29 through MIR31 by S8 control signal generator H9 when MODE11/ and MIR28 inputs to OR gate J10 are both high.

Table 5-1 indicates that if MODE11/ is low (sequential address mode, MIR00 and MIR01 = 11), the S8 control signal generator output (S8-S0 through S8-S2) selects the MIR transform (XT/MIR, S8 position 6) to load MIR24 through MIR 31 bits directly into the K or N register. If MODE11/ is high and $\overline{\text{MIR28}}$ (at H10 pin 3) is low, the S8 control signal generator outputs (S8-S0 through S8-S2) select field starts the transform (XT/FLDSTR, S8 position 7) to load into the K or N register. When both MODE11/ and $\overline{\text{MIR28}}$ are high, the

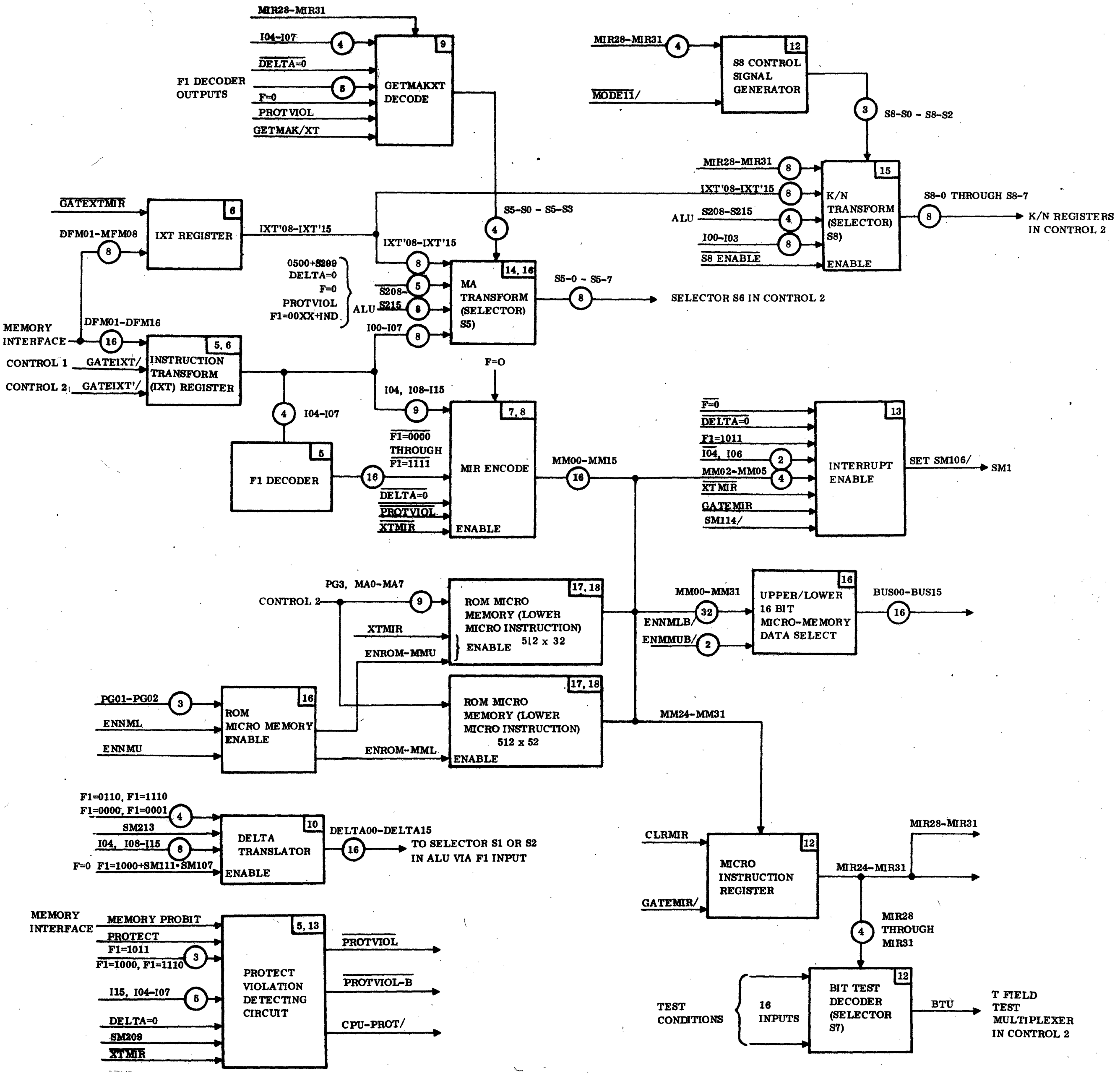
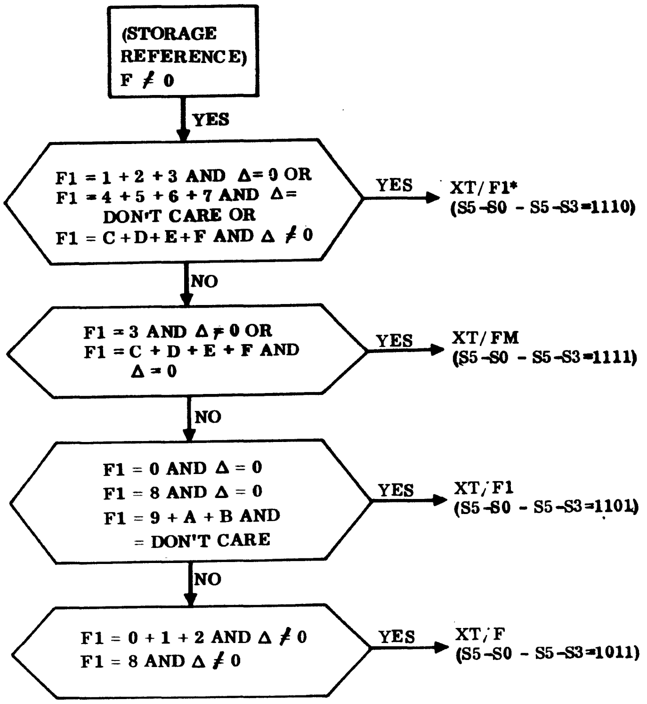


Figure 5-1. 1700 Transform with Read-Only Micro Memory Functional Block Diagram



061

Figure 5-2. MA Transform Selection for 1700 Storage Storage Reference Instructions

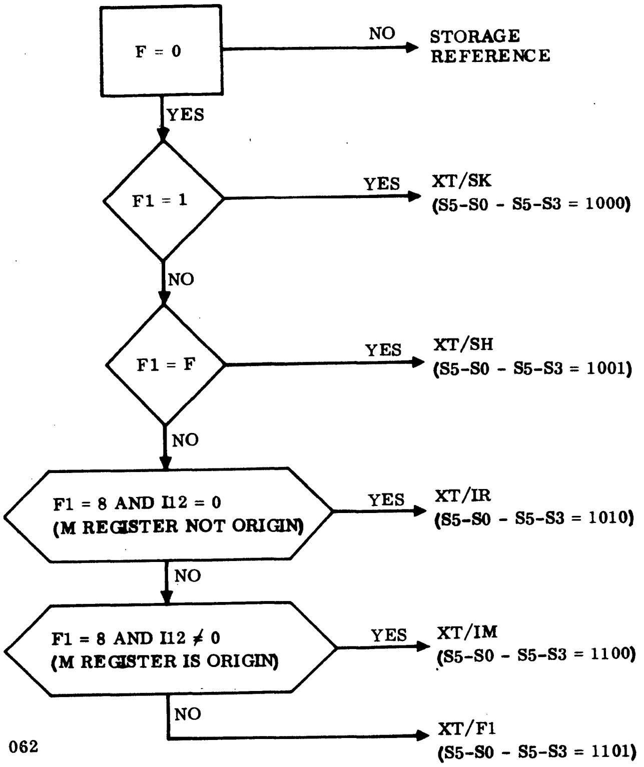
TABLE 5-1. K/N TRANSFORM POSITION SELECTION

MODE11/	$\overline{\text{MIR28}}$	S8-S2	S2-S1	S8-S0
1	1	MIR29	MIR30	MIR31
1	0	1	1	1
0	X	1	1	0

selection bits (S8-S0 through S8-S2) are dependent upon the state of MIR bits (MIR29 through MIR31).

BIT TEST DECODER

The bit test decoder (selector S7) is a 16-to-1 multiplexer (H12) that tests up to 16 external and/or internal conditions. These test conditions are selected by the state of MIR bits (MIR29 through MIR31).



062

Figure 5-3. MA Transform Selection for 1700 Register Reference and Inter-Register Reference Instructions

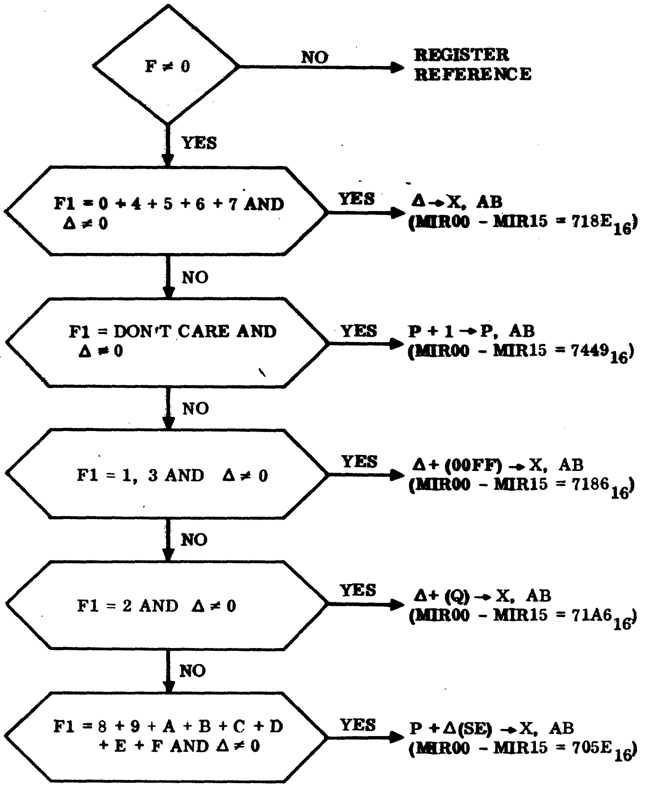
MIR ENCODE

During GETMAK/XT command, the macro instruction is encoded to form the upper 16 bits (MM00 through MM15) of micro memory. These most significant bits are loaded into the CPU micro instruction register via upper/lower micro memory data selectors DE13, E13, FG13, and G13 and the three-state bus. These bits of MIR control the arithmetic functions for read next instruction (RNI) cycles and other required operations to provide efficient emulation. Three types of MIR transform instructions are created (storage reference, register reference, and inter-register reference) based on the macro instruction being emulated.

The four 2-to-1 multiplexers E5, E6, G5, and G6, select one of two MIR transforms, either storage reference instruction (F≠0, figure 5-4) or register and inter-register reference instructions (F=0, figure 5-5). These multiplexers are enabled by the MIR transform signal (XT/MIR) that is only generated during GETMAK/XT operations. If a protect violation occurs, the D field of MIR encode is set to 0000 (no operation, NOP). Refer to appendix A for encoding equations applicable to storage, register, and inter-register reference instructions.

DELTA TRANSLATOR

The delta translator consists or 2-to-1 multiplexers A9, A10, C9, and C10 and associated AND and OR gates A6, A7, B7, C6, and D6. These elements translate the delta field of the 1700 instruction in accordance with the instruction F1 field selections. The multiplexers are enabled by AND gate B7 when the SM111 and SM107 are not set and F = 0 and F1 = 1000 are low. This enable allows the bit conditions



064

Figure 5-4. MIR Transform of 1700 Storage Reference Instructions

present on the multiplexer 0 input lines to be placed on the A, B, C, D output lines. The select signal input at pin 1 is generated as follows:

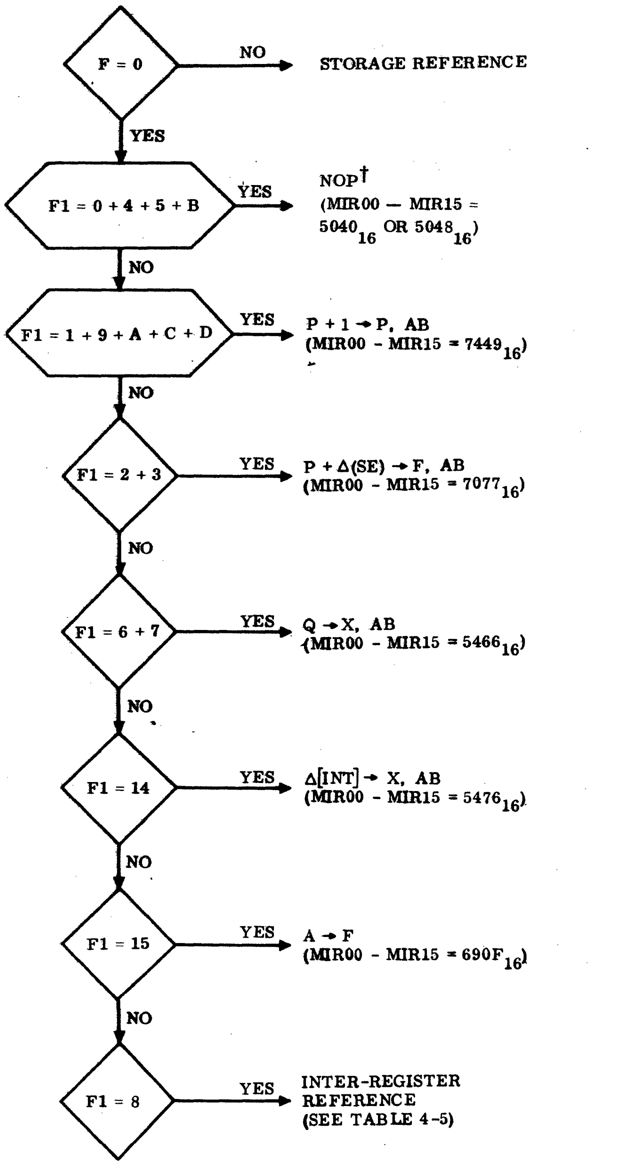
$\overline{SM213} \cdot F = 0 \cdot \overline{F1 = 0+1+6+E}$

If the select signal at output of AND gate A7 is high, it allows Δ (SK) skip and Δ (INT) interrupt instruction to be selected from the 1 inputs to the multiplexers. The Δ and Δ(SE) sign extended isntructions are selected from the 0 inputs when the AND gate A7 output is low. Status mode bit SM213 is set by the emulator only during emulation of enhanced instructions (that is, type 2 storage reference and field reference instructions).

READ-ONLY MICRO MEMORY

The read-only micro memory (figure 5-6) consists of 512 micro memory words (64 bit) formatted in two pages (0 and 1).

Each micro memory word consists of two 32-bit instructions (upper and lower). The upper instructions are contained in four 512-by 8-bit read-only memory (ROM) chips E9, G9, E12, and G12. The lower instructions are contained in four 512-by 8-bit ROM chips E7, G7, E10, and G10. The outputs of the corresponding upper and lower instruction bit ROM chips are OR wired to form a 32-bit micro memory three-state bus. The upper instruction ROM chips are enabled by ENROM-MMU, and the lower instruction ROM chips are enabled by ENROM-MML. These enables provide individual selection of the upper or lower micro instruction. ROM chips E9, G9, E7, and G7 (for MM00 through MM15) are disabled ($\overline{XTMIR}$ at input pin 19 low) during micro-instruction register operation to permit 16-bit outputs of the MIR encoder (instead of a ROM instruction) to be loaded into the CPU micro instruction register (control 2).



† NOP = THE D FIELD OF THE MICRO INSTRUCTION MUST BE 0 0 0.

065

Figure 5-5. MIR Transforms of 1700 Register Reference Instructions

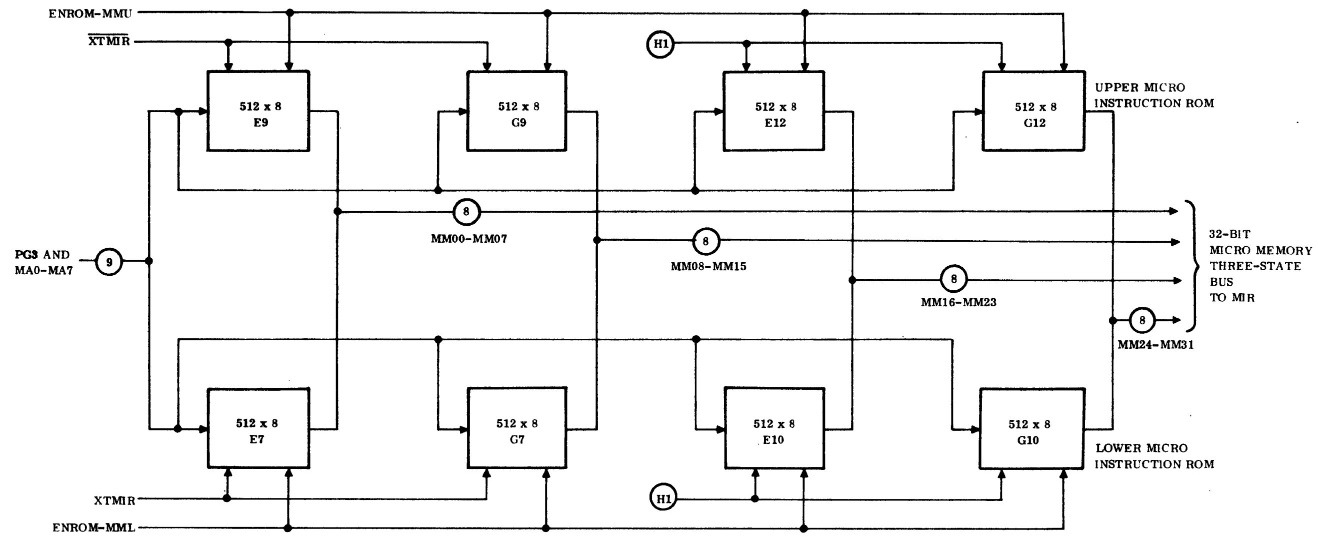


Figure 5-6. Simplified Read-Only Micro Memory Block Diagram

Each word in the ROM is addressed by memory address bits MA0 through MA7 and page bit PG3. The MA0 through MA7 bits specify one of 256 micro memory word pairs (upper and lower) within a page. The PG3 bit selects the page (0 or 1) from which the instruction is to be extracted (low = page 0, high = page 1).

UPPER/LOWER MICRO-MEMORY DATA SELECT

During a read micro-memory operation, the upper or lower 16 bits of data from micro memory can be transferred to the X register via the main CPU three-state bus. Whenever the B' code contains either a read micro memory upper or read micro memory lower command, an associated ENMMUB or ENMMLB is generated. These ENMMUB and ENMMUL signals enable the micro memory select multiplexers D13, E13, FG13, and G13 and select the micro-memory bits (MM00 through MM15 or MM16 through MM31) that will be placed on the three-state bus lines. The ENMMUB is the select signal that selects upper micro-memory bits (MM00 through MM15) when low and selects the lower micro-memory bits (MM16 through MM31) when high. The selected 16-bit output from these multiplexers is placed on the main CPU three-state bus lines (BUS00 through BUS15).

MISCELLANEOUS CIRCUITRY

PROTECT VIOLATION DETECTING CIRCUIT

The protect violation conditions detected by the 1700 transform hardware set the protect violation ($\overline{\text{PROTVIOL}}$) when an attempt is made to execute the following instructions.

- When the instruction is unprotected:

  -Any inter-register instruction with bit 15 set (F = 0, F1 = 8, and I15 = 1 at B2-10)

  -Enable interrupt (EIN, inhibit interrupt (IIN), set protect bit (SPB), and clear protect bit (CPB) instructions (F = 0, F1 = 01XX, and $\Delta \neq 0$ at B2-11)

  -Enable interrupt transform (EXI) instruction (F = 0 and F1 = E at B2-9)

These violations are detected only if the memory protect bit and protect ($\overline{\text{MEMPROBIT}}$ · PROTECT) signal at A2-2 are high. This indicates that the instruction is unprotected and the protect switch is set. (The protect switch is set via function control register bit 08 in the breakpoint controller or panel simulation firmware when the breakpoint controller is not present.)

- When a protected instruction follows the execution of an unprotected instruction - If the previous instruction is an unprotected instruction, then CPU protect flip-flop H5 is set during the read next instruction (RNI) cycle ($\overline{\text{XTMIR}}$ at J7-1 is low) by the GATEMIR clock. When both the enable fault detection (SM209) and CPU protect are set, the output at H7-12 will be low if the next instruction is a protected instruction. This causes the PROTFAULT at H7-8 to set. At GATEMIR time, flip-flop H5 is reset and flip-flop J4 is simultaneously set to keep PROTFAULT signal set for another memory cycle.

One exception is that if an interrupt caused the sequence of executing a protected instruction following the execution of a nonprotected instruction, this is not a protect violation. The emulator handles this exception by clearing the SM209 bit whenever an interrupt is detected. This disables the protect system and prevents PROTFAULT at H7-8 from setting. SM209 is set by hardware at time GATEMIR whenever a GETMAK/XT operation is executed.

- When execution of an unprotected miscellaneous instruction with the protect switch set is attempted- All miscellaneous instructions are privileged instructions. $\overline{\text{PROTVIOL-B}}$ is generated by:

$$\overline{\text{PROTVIOL-B}} = \overline{\text{F = 0 · F1 = B · DELTA = 0·}}$$
$$\overline{\text{MEMPROBIT · PROTECT}}$$

All other protect violations are detected by memory interface and processor hardware.

INTERRUPT ENABLE

Implementation of the interrupt function requires a combination of hardware and firmware. Status mode bit SM114 is first set by the 1700 emulator whenever the enable interrupt (EIN) instruction is emulated to pre-enable the interrupt function. If the instruction following EIN is neither a storage reference instruction ($F \neq 0$) nor an enhanced instruction, the hardware causes SETSM106 at J5-8 to be low at the next RNI cycle (XTMIR is high). The logic equation at J5-13 for this condition is:

$$\text{J5-13} = (\text{XTMIR} \cdot \text{GATEMIR}) \cdot \text{F} = 0 \cdot (\text{I04} + \overline{\text{I05}} + \text{DELTA=0'}) \cdot (\overline{\text{F1 = B}} + \text{DELTA} = 0')$$

If the instruction following EIN is a storage reference or enhanced instruction, SM106 is set after the 1700 emulator has executed a SUB- operation lincrement P counter). The SUB- operation indicates the end of the macro-instruction emulation. The SUBoperation is decoded at J5-12 if the F field contains 1011 the following:

$$\text{J5-12} = \text{MM02} \cdot \overline{\text{MM03}} \cdot \text{MM04} \cdot \text{MM05} \cdot \text{GATEMIR}$$

Once status bit SM106 is set, SM114 is cleared by the hardware. If the macro instruction being emulated happens to be an inhibit interrupt (IIN) instruction (instruction = 0500_{16}) or a false interrupt (S209 = 1), then the $\overline{\text{0500 + S209}}$ signal at J3-8 is low.

$$\text{J3-8} = \text{UNP-PROT} \cdot \overline{\text{S209}} + (\overline{\text{F = 0 · F = 5}}) \cdot \text{DELTA}' = 0$$

XTBLKT4 SIGNAL GENERATOR

The block transform at T4 signal (XTBLKT4) is generated by flip-flop J9 whenever GATEXTMIR is active low. Normally J9 is clocked by time T3 to set the XTBLKT4 output high to allow destination to be strobed at time T4. When GATEXTMIR is low, this signal is disabled to block the destination register strobe at T4.

BLKM100 SIGNAL GENERATOR

To avoid emulator branches to the interrupt subroutine during macro halt interrupt, a BLKM100 signal must be generated. This blocks the macro halt interrupt from being set whenever the CPU is in step mode and GETMAK/XT is executed. The $\overline{\text{BLKM100}}$ is set low during GETMAK/XT operation whenever a memory reference or enhanced instruction is decoded at set input H5-10.

$$\text{H5-10} = (\text{XTMIR} \cdot \text{GATEMIR}) \ (\overline{\text{I04}} + \text{I05} + \overline{\text{DELTA = 0}}) + \overline{\text{DELTA=0}} \cdot \text{F1} = \text{B}) + \overline{\text{F = 0}}$$

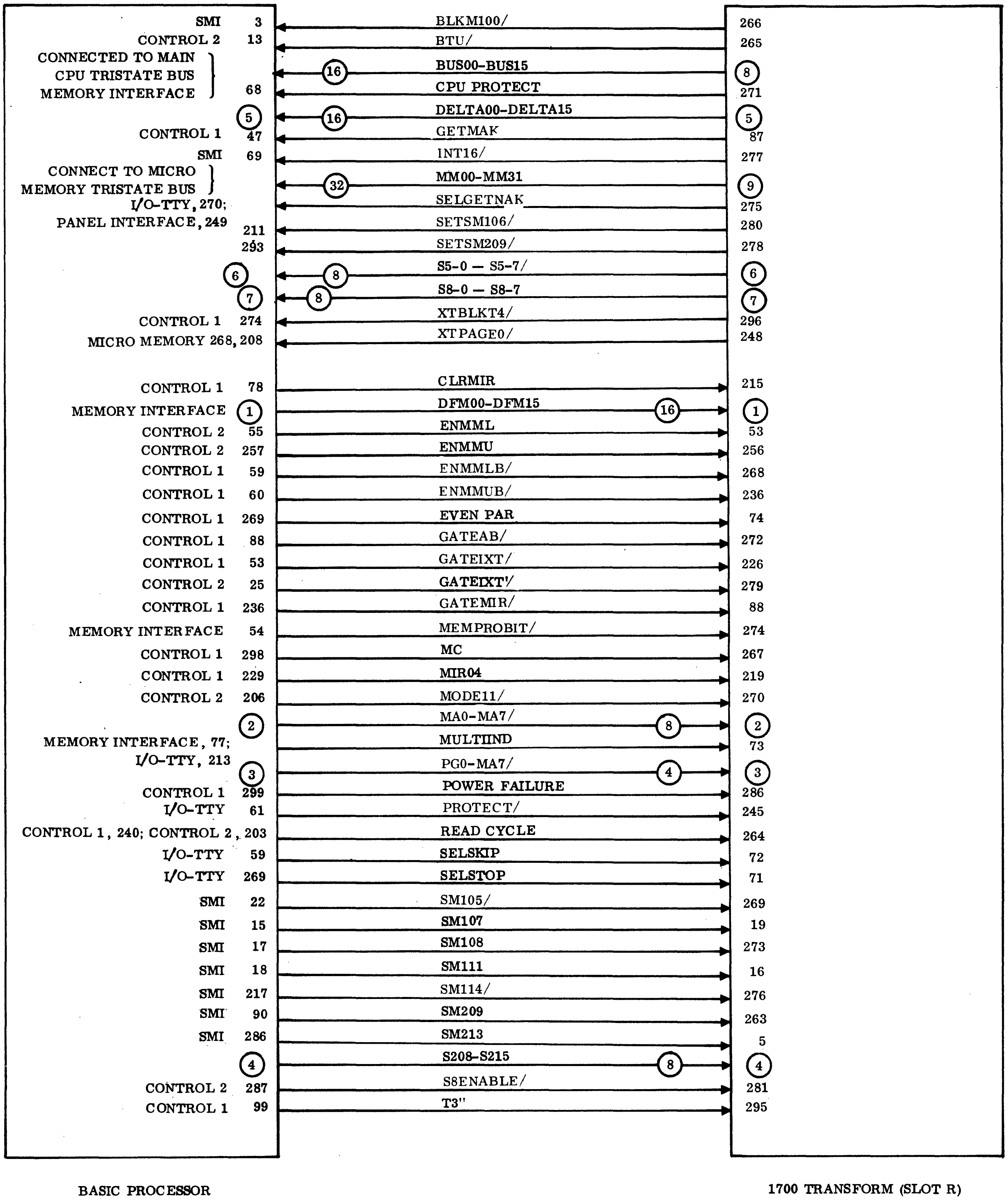
At the end of the instruction emulation, the emulator executes a SUB- operation (increment the P counter) that sets the $\overline{\text{BLKM100}}$ signal high. $\overline{\text{BLKM100}}$ is also set high by a master clear delayed signal (MCDELAYED/).

1700 TRANSFORM/PROCESSOR INTERFACE SIGNALS

Figure 5-7 shows all the connections between the 1700 transform and the processor. Signal names and logic board pin numbers are included. This diagram should be useful for troubleshooting to the board level.

LOGIC DIAGRAMS

The logic diagrams for the 1700 transform are located in the field print package for the 1700 transform with read-only memory.



073

Figure 5-7. Interconnecting Diagram Between 1700 Transform Module and Basic Processor (Sheet 1 of 2)

① DFM01-DFM16

MEMORY INTERFACE		TRANSFORM
16	DFM1	208
24	2	3
18	3	4
19	4	207
20	5	229
22	6	228
23	7	227
25	8	230
97	9	206
92	10	205
91	11	203
98	12	204
93	13	297
95	14	299
99	15	300
89	DFM16	298

② MA0/-MA7/

CONTROL 2		TRANSFORM
273	MA0/	231
68	1/	257
276	2/	258
69	3/	247
282	4/	246
97	5/	244
96	6/	253
95	MA7/	255

③ PG0/-PG3/

CONTROL 2		TRANSFORM
283	PG0/	54
258	1/	254
30	2/	249
267	PG3/	232

④ S208-S215

ALU		TRANSFORM
31	S208	94
241	09	95
232	10	293
231	11	96
18	12	97
16	13	98
225	14	99
19	S215	100

⑤ DELTA00-DELTA15

TRANSFORM		ALU MODULE PIN NUMBER
211	DELTA0	281
7	1	286
209	2	287
6	3	290
8	4	254
212	5	262
210	6	268
213	7	272
223	8	236
20	9	243
21	10	245
220	11	249
221	12	203
224	13	216
222	14	217
225	DELTA15	220

⑥ S5-0/ - S5-7/

TRANSFORM		CONTROL 2 PIN NUMBER
289	S5-0/	66
290	1/	274
90	2/	290
91	3/	275
291	4/	299
92	5/	298
292	6/	284
93	S5-7/	286

⑦ S8-0 - S8-7

TRANSFORM		CONTROL 2 PIN NUMBER
283	S8-0/	70
85	1/	71
285	2/	272
89	3/	269
86	4/	87
287	5/	88
288	6/	92
282	S8-7/	90

⑧ BUS00-BUS15

	TRANSFORM
BUS00	59
01	60
02	61
03	62
04	35
05	43
06	44
07	34
08	11
09	50
10	64
11	67
12	75
13	76
14	10
BUS15	9

⑨ MM00-MM31

	TRANSFORM
MM00	28
01	31
02	27
03	26
04	25
05	24
06	23
07	22
08	55
09	56
10	57
11	58
12	63
13	65
14	66
15	70
16	40
17	41
18	49
19	48
20	47
21	46
22	45
23	42
24	77
25	78
26	79
27	80
28	81
29	82
30	83
MM31	84

Figure 5-7. Interconnecting Diagram Between 1700 Transform Module and Basic Processor (Sheet 2 of 2)

PREVENTIVE MAINTENANCE

The 1700 transform module consists of one printed circuit board that plugs into the central processor unit (CPU) chassis.

All preventive maintenance for the transform is covered by the preventive maintenance procedures in the Basic Micro-Programmable Processor Maintenance Manual. This consists of cleaning any accumulated dirt or dust from the printed circuit board while performing preventive maintenance is required.

CAUTION

The 1700 transform printed wiring assembly contains electrostatic sensitive devices and is identified with a red solder mask. Exercise extreme care in handling to avoid damage. Common practices, such as touching a grounded surface before handling, and storing in an antistatic or conductive bag for storage or transfer, repairing only at properly equipped and grounded work station, etc., should be strictly followed.

CALIBRATION AND ADJUSTMENT

None is required.

TROUBLESHOOTING

Troubleshoot to the board level using the MSMP17 and ODS diagnostic tests.

MIR ENCODE EQUATIONS

1700 STORAGE REFERENCE INSTRUCTIONS (F≠0)

MM00 = 0

MM01 = 1

MM02 = $\overline{\text{PROTVIOL}}$

MM03 = 1

MM04 = 0

MM05 = DELTA'=0

MM06 = 0

MM07 = $\overline{\text{I04 + DELTA=0}}$

MM08 = $\overline{\text{I04 + DELTA=0}}$

MM09 = I04 + DELTA=0

MM10 = F1=0010 · $\overline{\text{DELTA=0}}$ + I04 · DELTA=0

MM11 = I04 · $\overline{\text{DELTA=0}}$

MM12 = F1=01xx + F1=0000 + DELTA=0

MM13 = $\overline{\text{DELTA=0}}$ · $\overline{\text{PROTVIOL}}$

MM14 = $\overline{\text{DELTA=0}}$ · $\overline{\text{PROTVIOL}}$

MM15 = $\overline{\text{DELTA'=0}}$ · $\overline{\text{PROTVIOL}}$

1700 REGISTER REFERENCE AND INTER-REGISTER REFERENCE INSTRUCTIONS (F=0)

MM00 = 0

MM01 = 1

MM02 = F1=0001 + F1=1101 + F1=1001 + F1=1010 + F1=1100 + F1=0010 + F1=0011 + F1=1111 + (I08 · I09 + I12) · F1=1000

MM03 = $\overline{\text{F1=1111}}$ · ($\overline{\text{I08}}$ + $\overline{\text{I09}}$ + I12 + $\overline{\text{F1=1000}}$)

MM04 = F1=1111 + I08 · $\overline{\text{I09}}$ · $\overline{\text{I12'}}$ · F1=1000

MM05 = F1=0001 + F1=1101 + F1=0110 + F1=0111 + F1=1110 + F1=1001 + F1=1010 + F1=1100 + (I08 · $\overline{\text{I09}}$ + I12) · F1=1000

MM06 = $\overline{\text{I12}}$ · ($\overline{\text{I08}}$ + I09) · F1=1000

MM07 = F1=1111 + $\overline{\text{I12}}$ · F1=1000

MM08 = $\overline{\text{I10}}$ · $\overline{\text{I12'}}$ · F1=1000

MM09 = $\overline{\text{F1=1000}}$ · $\overline{\text{F1=1111}}$ + I12 · F1=1000

MM10 = F1=0010 + F1=0011 + F1=0110 + F1=0111 + F1=1110 + $\overline{\text{I12}}$ · F1=1000

MM11 = F1=0010 + F1=0011 + F1=1110 + $\overline{\text{I11}}$ · $\overline{\text{I12'}}$ · F1=1000

MM12 = F1=0001 + $\overline{\text{F1=1000}}$ · $\overline{\text{F1=1110}}$ · I04 + I12 ·

MM13 = F1=0010 + F1=0011 + F1=1111 + F1=0110 + F1=0111 + F1=1110 + (I15 · $\overline{\text{I14}}$ + I13) · $\overline{\text{I12}}$ · F1=1000

MM14 = F1=0010 + F1=0011 + F1=1111 + F1=0110 + F1=0111 + F1=1110 + (I14 + I15) · $\overline{\text{I12}}$ · $\overline{\text{I13}}$ F1=1000

MM15 = F1=0001 + F1=1101 + F1=0010 + F1=0011 + F1=1111 + F1=1001 + F1=1010 + F1=1100 + (I12 + I13 + I14 + I15) · F1=1100

MA TRANSFORM SELECTION EQUATIONS

1700 STORAGE REFERENCE INSTRUCTIONS (F≠0) DURING GETMAK/XT TRANSFORM OPERATION†

S5-S0 = 1

S5-S1 = $\overline{\text{F1=0000 + F1=1000 · DELTA=0 + (I06 + I07)}}$ I01 · $\overline{\text{I05}}$

S5-S2 = $\overline{\text{(F1=0000 + F1=1000 + F1=0010 + F1=0001 ·}}$ $\overline{\text{DELTA=0}}$

S5-S3 = I04 · I05 · $\overline{\text{DELTA=0}}$ + $\overline{\text{I04}}$ · $\overline{\text{I05}}$ + $\overline{\text{I04}}$ · $\overline{\text{I05}}$ · $\overline{\text{DELTA=0   (I06 + I07)}}$

1700 REGISTER REFERENCE AND INTER-REFERENCE INSTRUCTIONS (F=0) DURING GETMAK/XT OPERATION†

S5-S0 = 1

S5-S1 = $\overline{\text{I12}}$ · F1=1000

S5-S2 = $\overline{\text{I12}}$ · F1=1000 + $\overline{\text{F1=1111}}$ · $\overline{\text{F1=0001}}$

S5-S3 = $\overline{\text{F1=0001 + F1=1000}}$

† For all other transform operations, the S5 select signals, S5-S0 through S5-S3, equal MIR28 through MIR31, respectively.

COMMENT SHEET

MANUAL TITLE CDC® DE402-A 1700 Transform with Micro Memory Hardware Maintenance Manual

PUBLICATION NO. 96728700 REVISION A

FROM NAME:

BUSINESS
ADDRESS:

COMMENTS: This form is not intended to be used as an order blank. Your evaluation of this manual will be welcomed by Control Data Corporation. Any errors, suggested additions or deletions, or general comments may be made below. Please include page number.

CUT ALONG LINE

STAPLE

STAPLE

FOLD

FIRST CLASS
PERMIT NO. 333

LA JOLLA CA.

BUSINESS REPLY MAIL

NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.

POSTAGE WILL BE PAID BY

CONTROL DATA CORPORATION
PUBLICATIONS AND GRAPHICS DIVISION
4455 EASTGATE MALL
LA JOLLA, CALIFORNIA 92037

FOLD

STAPLE

STAPLE