NOVA 1200 TO CDC 853 DISK CONTROLLER

by

Leo J. Slechta and Zoltran Strohoffer

UCC 75001 June, 1975

University Computer Center
University of Minnesota
Minneapolis, Minnesota
55455

PREFACE

In 1972 Professor Naftaly Minsky proposed a means of implementing a partial associative memory capability within a rotation device.* Under the direction of Professor Larry Kinney of the Electrical Engineering Department, two graduate students, Michael Burns and Ralph Benno, initiated the development of such a system to a Data General Nova 1200 minicomputer. A working model was completed by the University Computer Center System Engineering Group. This work was completed and documentation was developed by Leo Slechta and Zoltan Strohoffer. This report is a result of their efforts.

The authors wish to acknowledge the financial and moral support of Mr. Daniel B. Magraw and the Minnesota State Department of Administration through the Information Resources Development Fund.

Deep appreciation is expressed by UCC to Mr. Slechta and Mr. Strohoffer who extended themselves by working and completing the documentation on this project long after they had left the University environment.

---

*Rotation Storage Devices as Partial Associative Memory," University of Minnesota Computer, Information, and Control Sciences Department, Technical Report 72-4, April 28, 1972.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. GENERAL DESCRIPTION

1.1 INTRODUCTION

This document describes a disk controller designed for use with the NOVA 1200 computer and the CDC 853 disk drive. Design features include: the ability to process physically adjacent sectors on the same or different disk surfaces without the necessity of waiting one disk revolution, sector access error detection, and keyword writer protection.

1.2 CONTROLLER PHYSICAL DESCRIPTION

The disk controller is constructed with TTL integrated circuits and is contained on two boards that plug into the Nova 1200 mainframe (the 4042 board and the interface board) and 4 boards (the X board, the Y board, the IA - IC board, and the ID - IF board) that mount in a stand-along aluminum container called the I/O box. Nova 1200 to disk controller and disk controller to CDC 853 interconnections are shown in FIG 1-1.

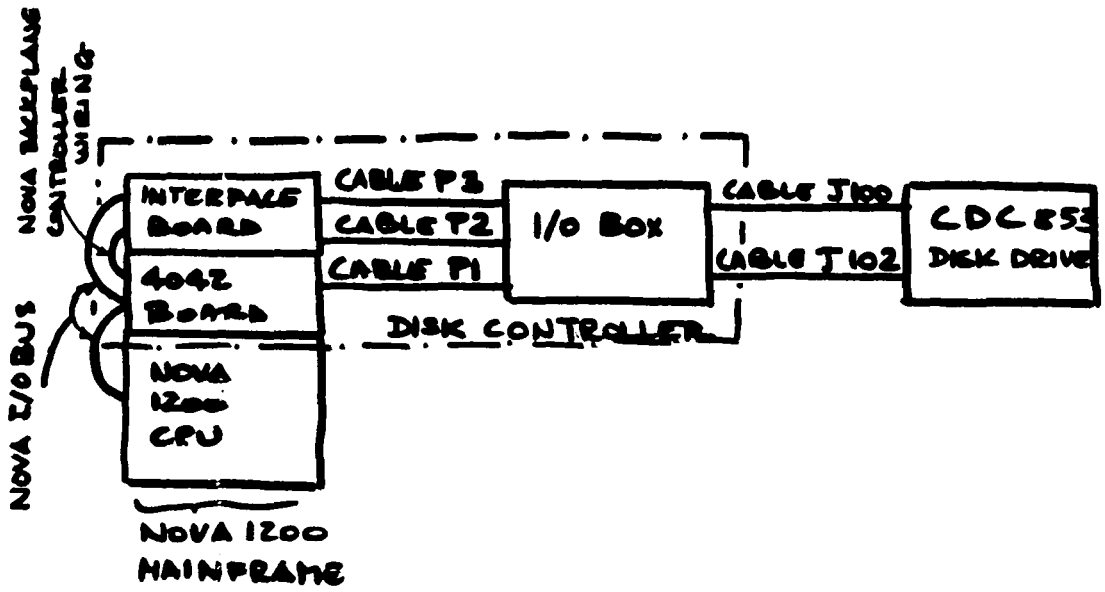


FIG 1-1 DISK STORAGE SYSTEM-FUNCTIONAL BLOCK DIAGRAM

1.3 DISK STORAGE SPECIFICATIONS

The 853 disk has a disk pack containing 10 recording surfaces. Each recording surface has 100 tracks and each track has 16 sectors. The cylinder, track, and sector are defined as follows:

1) Cylinder - all the recording surface under all 10 read/write heads at a given position.
2) Track - the recording surface under one read/write head at a given cylinder position.
3) Sector - the smallest subdivision of the disk pack. Each track is divided into 16 sectors.

Each sector contains an 85 word (16 bits/word) data field and all data transfers are one sector (85 words) in length. Refer to Table 1-1 for a list of disk storage specifications.

TABLE 1-1 DISK STORAGE SPECIFICATIONS

STORAGE CHARACTERISTICS	
DATA FORMAT:	16 bits/word*
	85 words/sector
	16 sectors/track
	10 tracks/cylinder
	100 cylinders
DATA CAPACITY	85 words/sector
	1,360 words/track
	13,600 words/cylinder
	1,360,000 words/853 disk
ACCESS TIME	
Head positioning time	165 milliseconds (maximum)
Cyl-to-cyl positioning time	30 milliseconds
TRANSFER RATE	
Data rate	14.4 usec/word*
Bit rate	.08 usec/bit

* The disk controller generates and writes 1 parity bit for each 8 bit byte written on the disk. So, for each 16 bit word transferred from the Nova to the 853, 18 bits are written on the disk.

1.4 ADDRESSING THE DISK PACK

Any sector on the disk pack can be accessed under program control from the computer. Sectors are addressed by a 16 bit word sent from the computer to the controller. This word is called the Sector Address Word and contains the sector number, the head (or surface) number, and the cylinder number. The least significant four bits of the Sector Address Word identify the sector within a track (1 of 16). The next higher four bits identify the head within a cylinder (1 of 10). Head numbers 10 to 15 are illegal. The upper 8 bits identify the cylinder (1 of 100). Cylinder addresses above 99 are illegal. The Sector Address Word is illustrated in FIG 1-2.

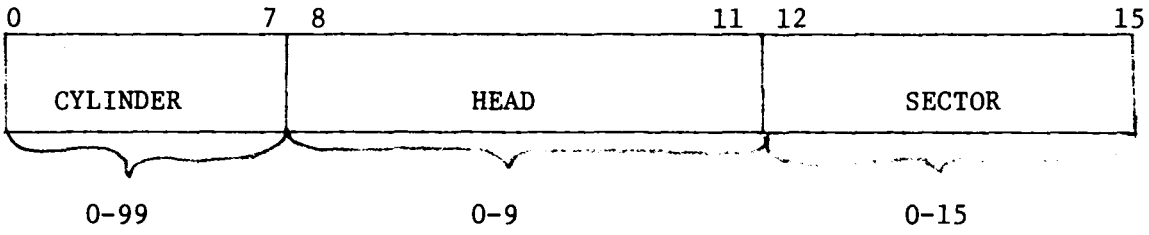


FIG 1-2 SECTOR ADDRESS WORD

1.5 SECTOR FORMAT

All sectors on the disk pack have a format that can be loosely divided into two portions; the address portion and the data portion. The address portion of the sector includes an address word identifying the sector and a keyword used to provide sector write protection. The data portion of the sector includes 85 data words and a checkword used by the disk controller to verify correct data transmission. The sector format is illustrated in FIG 1-3.

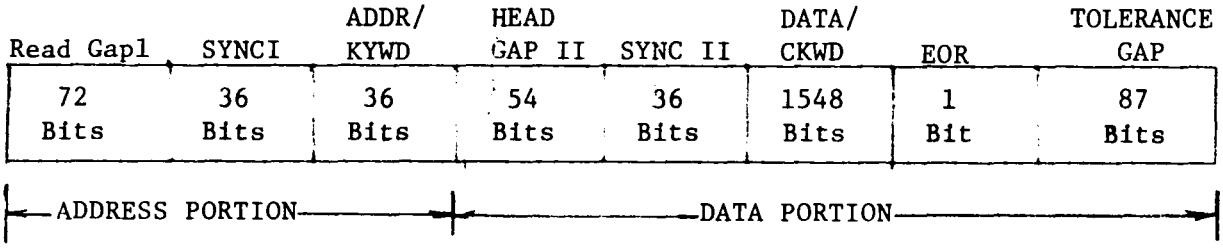


FIG 1-3 SECTOR FORMAT FIELDS

A detailed description of the sector format fields is given below:

1) Head Gap I (72 bits) - the Head Gap I field provides time for the read/write amplifiers to turn on and stabilize.
2) Sync I (36 bits) - the Sync Pattern I field contains all zeros except for the last bit which is a "1".
   A "1" denotes the end of the pattern and tells the disk controller that the read/write heads are entering the ADDR/KYWD field.
3) ADDR/KYWD (36 bits) - The Address/Keyword field contains a 16 bit address and a 16 bit keyword. Associated with each 16 bit word are two parity bits which are used to verify correct data recovery. The 16 bit address identifies the sector and is identical in format to the Sector Address Word. The 16 bit keyword is used to provide write protection for the sector being accessed. Only bits 4 through 11 of the keyword match the contents of the keyword portion of the FMK2 register (see sect. 1.7.1), can the accessed sector be written into or erased.
4) Head Gap II (54 bits) - The Head Gap II field provides the time needed to shut the read amplifiers off and turn the write amplifiers on when the disk controller is executing a write or an erase operation.
5) SYNC II (36 bits) - The Sync Pattern II field contains all zeros except for the last bit which is a "1'. A "1' denotes the end of the pattern and tells the disk controller that the read/write heads are entering the DATA/CHKWD field.
6) DATA/CHKWD (1548 bits) - The Data/Checkword field contains 85 16-bit data words followed by a 16-bit checkword. Associated with each of the 85 16=bit data words are two parity bits which together with the checkword are used to verify correct data recovery from the disk.
7) EOR (1 bit) - The End Of Record bit tells the disk controller to shut off the controller R/W logic.
8) Tolerance Gap (87 bits) - The Tolerance Gap is used to safely account for variations in the disk controller timing oscillator and the disk pack revolution speed.

1.6 FUNCTIONAL DESCRIPTION

The disk controller performs three operations: general seek, return to zero seek, and data transfers. To transfer a block (sector) of data, the programmer typically must perform a general seek operation to position the read/write heads over the correct cylinder and then perform a data transfer operation to access the desired sector. When performing a general seek, return to zero seek, or a data transfer operation, the disk controller monitors various disk and controller interruptable conditions and will interrupt the Nova 1200 if these interruptable conditions occur and if interrupts are enabled.

1.6.1 GENERAL SEEK OPERATION

The general seek operation positions and read/write heads over the cylinder specified by the contents of the cylinder portion of the Address Word Register (AWR). The AWR is loadable from the NOVA by executing the DOA AC, DSK instruction. The seek operation is executed when the NOVA executes a NIOP DSK or DOAP AC, DSK instruction. There are several interrupts (cylinder error, seek error, request buffer full, and DONE 20) that can be generated during a seek operation if the necessary interrupt enable conditions are present both in the NOVA and the disk controller. For a detailed explanation of the disk controller interrupt structure, see Section 1.6.4 and 1.7.1.

1.6.2 RETURN TO ZERO SEEK OPERATION

The return to zero seek operation positions the read/write heads over cylinder zero. The disk controller performs a return to zero seek operation when the NOVA 1200 executes a NIOP 21 or an IORST instruction and when the RESET key on the NOVA operator console is depressed. The return to zero seek operation is the only way that a seek error condition can be cleared. A DONE 20 interrupt can be generated following a return to zero seek operation if the necessary interrupt enable conditions are present in both the NOVA and the disk controller. For a detailed explanation of the disk controller interrupt structure, see Section 1.6.4 and 1.7.1.

1.6.3 DATA TRANSFER OPERATION

The data transfer operation performs the data transfer (read sector, write sector, write address, or erase sectors) specified by the contents of the function portion of the FUNCTION, MODE, KEYWORD register (FMK) at the sector specified by the contents of the head and sector portion of the AWR register. The AWR is loadable from the NOVA by executing the DOA AC, DSK instruction and the FMK register is loadable from the NOVA by executing the DOC AC, DSK instruction. The data transfer operation is initiated when the NOVA executes a NIOS DSK or DOAS AC, DSK or DOCS AC, DSK instruction.

When performing a read, write, or write address operation, the NOVA data channel is used to transfer data between the disk controller and NOVA memory. Before starting one of these data transfer operations, the MEMORY ADDRESS REGISTER (MAR) must be loaded with the starting address of the buffer in memory to (or from) which the data is to be transferred. The length of the buffer to be transferred is implicit to the data transfer operation performed. The read sector and write sector data transfer operation will result in an 85 word transfer and a write address data transfer operation will result in a 2 word transfer. The erase data transfer operation doesn't transfer any data from memory to the disk, it simply overwrites the data portion of the requested sector with all zeros. Although the programmer has no control over the length of a data transfer operation, the disk controller does contain a WORD COUNT register (WC) that the programmer can monitor. The WC contains a count of the number of words transferred since the first data word of the sector was transferred. The WC register is zeroed at the end of each sector (this means the WC register only contains meaningful information while the data transfer operation is in progress).

As explained in Section 1.5, each sector contains an address portion and a data portion. The address portion of the sector is written using the write address operation. To add some measure of protection against a programmer unwittingly destroying the address portion of a sector, the write address operation can only be performed with all interrupts disabled.

When performing a read sector, write sector, or erase sector operation, the disk controller locates the requested sector, compares the address/keyword information read from the address portion of the sector with the applicable contents of the AWR and FMK registers, and, assuming the comparisons are valid, performs the data transfer operation on the data portion of the sector. A keyword match is required only for a write or an erase operation. The disk controller performs even parity generation (write operation) and checking (read operation) on each 8 bit data byte transferred to and from the disk and generates (write operation) and checks (read operation) a longitudinal checkword on the data portion of the sector.

Most disk controllers are not capable of processing physically adjacent sectors consecutively, but can process an adjacent sector only after waiting for a complete disk revolution. This controller, however, is capable of processing adjacent sectors on the same or a different disk surface as long as all sectors to be processed are on the same cylinder. In other words, after processing sector 2, surface (head 3, processing may immediately continue at sector 3, surface 3 or sector 3, surface 9, etc. This controller property allows 160 sectors to be processed sequentially at the maximum data transfer rate.

In order to be capable of sequential sector processing as described above, the disk controller must accept accessing information (AWR, FMK, MAR) pertinent to the next sequential sector while processing the sector currently under the read/write heads. In general, access information for a new sector may be loaded into the controller when the read/write heads are over the data portion of the sector being processed. The disk controller generates an "address word request" interrupt (if interrupts are enabled) when the read/write heads have entered the data portion of the sector to inform the programmer that the controller is now ready for new accessing information.

There are numerous interrupts that can be generated during a data transfer operation if the necessary interrupt enable conditions are present in both the NOVA and the disk controller. For a detailed explanation of the disk controller interrupt structure, see section 1.6.4 and 1.7.1.

1.6.4 DISK CONTROLLER INTERRUPT STRUCTURE

The disk controller has a rather extensive interrupt structure which allows the programmer to closely monitor disk controller operation. In general, each interrupt can be enabled/disabled from the NOVA and each interrupt has associated with it a device code (20 or 21) that is used by the programmer to identify the interrupt source.

Device 20 interrupts are generated (if enabled) when a SEEK or RETURN TO ZERO SEEK operation completes. Device 20 interrupts are enabled by executing a MASK OUT instruction with DATA BIT 9 set to zero.

Device 21 interrupts are generated (if enabled) when a data transfer operation completes or when error conditions occur during the execution of a SEEK or data transfer operation. The 16 device 21 interrupts, which are divided into six subclasses, are enabled by executing a MASK OUT instruction with DATA BIT 8 set to zero. The six device 21 interrupt subclasses and their interrupt enable procedure are defined below:

START/END OF SECTOR FLAGS - If FMK register bit 13=1, enable "END OF RECORD" interrupt. If FMK register bit 12=1, enable "Address Word Request" interrupt.

PROTECTION FAULTS (PF) - If INT DIS register bit 12=0, enable PF interrupts.

ADDRESSING ERRORS (AE) - If INT DIS register bit 13=0, enable AE interrupts.

TIMING/DATA TRANSFER FAULTS (T/D) - If INT DIS register bit 14=0, enable T/D interrupts.

BUFFERING FAULTS (BF) - If INT DIS register bit 15=0, enable BF interrupts.

FATAL FAULTS - Fatal fault interrupts are always enabled.

The INT DIS register is loaded from the NOVA by executing a DOA AC, 21 instruction. NOTE: To enable a device 21 interrupt, the programmer must (1) enable device 21 interrupts (using the MASK OUT instruction) and (2) enable the required device 21 interrupt subclass(es) (using the DOA AC, 21 instruction).

To determine the cause of a device 21 interrupt, the programmer can read the contents of the STATUS REGISTER using the DIC AC, DSK instruction. The STATUS REGISTER is a 16 bit register which has a bit assigned to each of the 16 device 21 interrupts (i.e., the occurence of an interruptable condition sets the appropriate bit in the STATUS REG and then interrupts the NOVA if the proper enable conditions are met). For a definition of the STATUS REG bit assignments, see section 1.7.1.

1.7 INSTRUCTION REPERTOIRE

The general set of NOVA I/O instructions that are used to control the disk are divided into two types: (1) "Register transfer commands" that are used to read and/or load registers in the disk controller and (2) "Execution commands" that are used to initiate an operation in the disk controller. A summary of these I/O instructions is given in Table 1-2.

A detailed description of these instructions can be found in SECT 1.7.1 and 1.7.2. In most cases, an execution command can be combined with a register transfer command (See NOVA manual explanation of IOT instructions).

TABLE 1-2 DISK CONTROLLER IOT INSTRUCTION SUMMARY

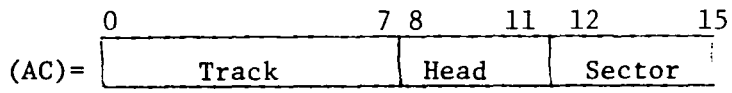
INSTRUCTION		DESCRIPTION
REGISTER TRANSFER COMMANDS		
DOA	AC,DSK*	load AWR register
DIA	AC,DSK*	read AWR register
DOB	AC,DSK*	load MAR1 register
DIB	AC,DSK*	read MAR2 register
DOC	AC,DSK*	load FMK1 register
DIC	AC,DSK*	read STATUS register
DOA	AC,21	load INT DIS register
DIA	AC,21	read WC register
EXECUTION COMMANDS		
NIOP	DSK*	execute SEEK operation
NIOP	21	execute RTZS operation
NIOS	DSK*	execute data transfer operation
NIOC	21	clear STATUS REG and device 21 INT REG flip-flops
NIOC	DSK*	CLEAR DEVICE 20 BUSY, DME, and INT REG flip-flops

* DSK=20 in NOVA 1200 assembly language.

1.7.1 REGISTER TRANSFER COMMANDS

DOA AC, DSK , where DSK=20_8, the device code.

Sends disk address from CPU accumulator AC according to the format



to address word register (AWR).

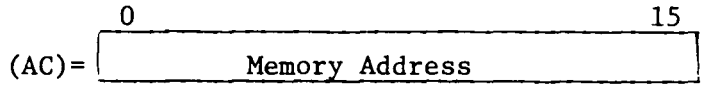
May be combined with P pulse, DOAP, which starts the seek operation (i.e., use DOAP only when a seek is actually required!)

DIA AC, DSK

Reads disk address word register into CUP accumulator AC.

DOB AC, DSK

Sends memory address from CPU accumulator AC according to the format



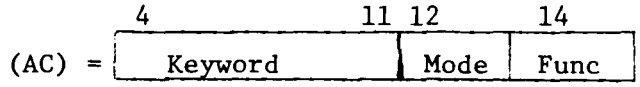
to memory address register.

DIB AC, DSK

Reads memory address into CPU accumulator AC. At the beginning of a data transfer operation, the memory address register must contain the address of the memory cell into which the first data word will be transferred (read) or from which the first data word will be taken (write).

DOC AC, DSK

Sends the function word to the disk according to the format



to FUNCTION-MODE-KEYWORD registers.

The functions (data transfer operations) are:

(bit 15 is at right)

00 - Read Data

01 - Write Data

10 - Write Address

11 - Erase

The modes are:

(bit 13 is at right)

00 - Leave mode unchanged

01 - Interrupt at end of sector only (EOR)

10 - Interrupt at beginning of data only (AWR)

11 - Both interrupts

If the keyword does not match the keyword stored on the addressed sector, the write and erase functions are not permitted. If the keywords match, all functions are permitted. This does not apply to address-write: it is used to write the keyword and sync pattern.

May be combined with the S pulse, DOCS, which starts the data transfer operation.

The address word request interrupt or status bit (AWR) indicates that the controller is ready to accept new accessing information and commands.

DIC AC, DSK

Reads the Status Register into the CPU accumulator AC. The status bits are defined as follows:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
AWR	WA	KF	CA	HA	SA		CH	PA	LD	BF	RF	NR	SE	DPU	EOR
	Protection Faults		Addressing Errors				Timing, Data Transfer Faults		Buffering Faults			Fatal			

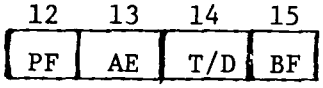
Starred (*) bits below are generated by the disk storage drive, the others are generated by the disk controller.

AWR	address word request (set with first data word of sector
WA	write address (comes up if interrupt is not disabled for write address function)
KF	keyword fault keyword does not match the one recorded on diskpack (during write, erase)
CA	cylinder address error
HA	head address error (head addresses do not match or if nonexistent address is requested)
SA	sector address error
SY	not used
CH	checkword error
PA	parity error
LD	lost data
BF	buffer full (trying to load MAR or FNC registers when already loaded)
RF	request buffer full (two seek requests!)
NR*	not ready - direct from disk drive = $\overline{\text{Sel. Unit Ready}}$
SE*	seek error - from disk when requesting a seek past 99 or 0.
DPU*	disk pack unsafe
EOR	end of record (after last word of sector)

The status register is cleared with CLEAR Pulse, device code 21. Progress of the Read/Write operation can be tested under program control (no interrupt is generated!) by using the BUSY 21 skip instruction. There is no hardware for DONE 21, so DONE 21 will always skip on zero and should not be used at all.

DOA AC, 21

Load the INT DIS register from CPU accumulator AC. Loading the INT DIS register with ones disables the 21 interrupt subclasses according to the following format:

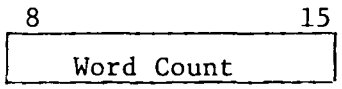


PF	protection faults (bits 1-2)
AE	addressing errors (bits 3-5)
T/D	timing, data transfer errors (bits 6-8)
BF	buffering errors (bits 9-11)

Bits 0, 15 and the fatal error bits cannot be disabled this way (see Section 1.6.4).

DIA AC, 21

Reads the Word Count Register into CPU accumulator AC



1.7.2 EXECUTION COMMANDS

NIOP DSK Initiate Seek Operation

Must be issued whenever a change in cylinder address is desired. Example: NIOP 2Ø: Does no I/O transfer, just starts a seek operation to the cylinder specified in the ADDRESS WORD register.

NIOP 21 Initiate RETURN to Zero Seek Operation

Sends RTZS command to disk drive. Heads will be retracted to cylinder zero. Example: 1) NIOP 21; 2) IORST (it always executes a zero seek command which should be taken into consideration when using it).

NIOS DSK Initiate Data Transfer Operation

This command begins the data transfer operation, and must be given for every sector transferred.

NIOC 21 Clear STATUS REGISTER

This command clears the device 21 INT REQ flip-flop and STATUS REGISTER bits 0, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 15. Note: STATUS REG bits 1, 12, 13, 14 cannot be cleared by the NIOC 21 command.

NIOC DSK Clear Device 20 Flip-Flop

This command clears the device 20 BUSY, DONE, and INT REQ flip-flops.

1.7.3 PROGRAMMING NOTES

The execution of the following instructions require careful attention by the programmer because their execution can cause potentially unexpected results to disk controller operation.

DICC 0, CPU or IORST

Resets all controller registers, and returns the heads to track 00.

It enables all interrupts to the CPU (it clears the interrupt disable flags of all I/O devices, disk controller included) except those generated by the disk storage drive itself in order to allow the drive to be (or be kept) powered off without interrupting the CPU.

HALT (instruction) or
STOP (from the console)

Executing these commands will result in erroneous data being written on the disk if the controller is actively processing a write data function. Halting the Nova disables the Data Channel. The controller will unsuccessfully attempt to use the Data Channel and a LOST DATA error will result. LOST DATA terminates all read-write operations and turn off the write head while the head is over the data portion of the sector. This results in garbage written on the disk.

This can be prevented by insuring that the read/write operation is completed before halting the Nova (use 21 BUSY SKIP or wait for END OF RECORD, bit 15 of the status word).

1.8 "READ/WRITE DISK PROGRAM" FLOWCHART

The flow chart illustrates the simple routine used to read or write any size buffer on the disk.

start

↓

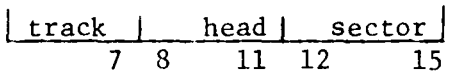
IORST — Master clear. Initiates seek to track zero. Enables all interrupts to CPU except the ones associated with the disk, device code 21.

↓

wait for done, device code 20 — check completion of seek operation (either flag check or interrupt)

↓

Load address word register with content of accumulators — DOA-, 20
format of register:



↓

initiate seek to specified track — if needed, may use DOAP-, 20 instead of DOA-, 20 as above.

↓

read back AWR (if wanted) — DIA-, 20 This reads back the register only. Will not tell where the heads are.

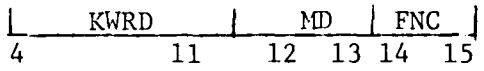
↓

wait for seek done — example for flag checking:

```
SKPDN  20   wait for flag bit
JMP   .-1   to set, clear it
NIOC   20   once it becomes set
```

↓

load 'function mode and keyword' register with content of accumulator — DOC-, 20
format of register



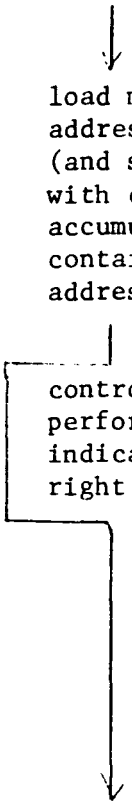
FNC operation

14	15	
0	0	read
0	1	write
1	0	address write
1	1	erase

MODE operation

12	13	
0	0	no change
0	1	intr. at EOR
1	0	intr. at AWR
1	1	both intr.'s

↓



load memory address register (and start function) with content of accumulator which contains core buffer address

DOB-, 20 load MAR
DOBS-, 20 load MAR and start to execute the function specified.

controller will access memory at location specified and perform depending on the function at the sector specified:

controller will perform function indicated at right

.a read operation
.a write operation if the key words match
.an erase operation if keywords match
.an address write if no interrupts with device code 21 are enabled.

after one word transferred, the memory address register will increment and transfers the next word until it reaches the end of the data portion of the sector at which time checkword and the EOR bit (bit 15 of status register) are generated. The AW bit (bit 0 of status register) indicates when the first data word is transferred between disk and controller (after address and keyword section of sector is handled).

wait for AW bit

for example:

```
DIC-, 20            read status register
MOVZL-, -, SNC      shift bit 0 to carry position
JMP .-2             and skip on nonzero carry:
NIOC  21            clear register when bit sets
```

Once the bit sets, new accessing information can be specified since sector address was transferred to second rank of the address word register. This is needed only when recording on non-adjacent sectors (the register was automatically incremented by AW bit).

The first rank of MAR must be rewritten with the new buffer's starting address (the second rank is incremented during data transfer!).

Also if new function, mode and keyword is selected - it can be loaded at this time.

give new start pulse if sequential accessing mode is used

for example NIOS 20
Remember: the controller transfers only one sector's worth of data (85 words) when it halts. There is no word count register in the conventional sense, that is saying how many words have to be buffered. It is the programmers duty to determine how many sectors his buffer is worth.

The controller's word count register, although readable, is used to count how many times the MAR must be incremented.

If no sequential accessing is used start pulse can be given after EOR. The start pulse is buffered and used only after EOR is detected.

wait for EOR

EOR (bit 15 of status register) indicates the transfer of a full sector (85 wds). It is possible to verify the sector now, to check other status bits, clear a core buffer area etc.

clear status register and give start pulse if not given earlier

etc.

2. HARDWARE DESCRIPTION

2.1 INTRODUCTION

This section contains a detailed explanation of disk controller operation at the logic implementation level. This explanation, together with the disk controller logic drawings (SECT 3) and the disk controller timing diagrams (SECT 4), should provide the reader with a clear understanding of how the disk controller works.

2.2 DISK CONTROLLER BLOCK DIAGRAM

The DISK CONTROL BLOCK DIAGRAM is contained on FIG 3-1. This diagram shows the organization of the disk controller. Included in each "block" are the Figure numbers of the associated logic drawings.

The INT DIS register is loaded from the NOVA I/O bus. The STATUS REG, which gets set from various points in the disk controller, together with the INT DIS register, is used to generate device 21 interrupts. The AWR register is loaded from the I/O bus and is used to supply the disk address word (cylinder, head, sector) to the disk (via the A/C bus). The FMK register, a double ranked register to allow processing of adjacent sectors, is loaded from the I/O bus and is used to provide function information to the R/W CONTROL logic and keyword information to the COMPARE LOGIC.

The COMPARE LOGIC is used to test the access information (found in AWR and FMK2) with the data read from the address portion of the sector under the read/write heads (READ BUF). The CUR CYL REG, ADDER, and A/C bus are used to provide the disk controller with new cylinder information (for SEEK operations) and new head information (for data transfer operations).

When performing a read operation, the PARITY CHECK logic, CHECKWORD GEN, and IN/OUT SHIFT register receive the serial bit stream read from the disk. The READ BUF is loaded from the IN/OUT SHIFT register as each word is assembled (in the IN/OUT SHIFT register). The contents of the READ BUF are then transferred to the NOVA memory via the I/O bus while the next word is being assembled.

When performing a write operation, the WRITE BUF is loaded from the NOVA memory via the I/O bus. The contents of the WRITE BUF are transferred in parallel to the PARITY GEN logic and the IN/OUT SHIFT register and then serially from the IN/OUT SHIFT register to the disk.

The MAR, a double ranked register to allow processing of adjacent sectors, is loaded from the I/O bus and is used to provide the address of the buffer in core memory to (or from) which data is to be transferred. The contents of the STATUS REG, AWR, WORD COUNT REG, and MAR can be read by the computer (via the I/O bus).

The IOT INST DECODE logic is used to execute IOT instruction control and register transfers between the disk controller and the NOVA. The DATA CHAN logic is used to control the transfer of information between NOVA memory and the READ BUF/WRITE BUF during data transfer operations. The SEEK CONTROL logic is used to control the execution of SEEK and Return to Zero Seek operations. The R/W CONTROL logic is used to control the execution of data transfer operations.

2.3 ADDRESS WORD REGISTER

Figure 3-2 contains the ADDRESS WORD REG (AWR). The AWR consists of a 4 bit sector field (A13), a 4 bit head field (B13), and an 8 bit cylinder field (C13 and D13). A13 is a presettable binary counter, B13 is a presettable decode counter, and C13/D13 are 4 bit latches. The AWR is loaded when the computer executes a DATA OUT A, DEVICE CODE 20 IOT instruction ($\overline{20\ DOA}$). The AWR can be read (by the computer) with the execution of a DATA IN A, DEVICE CODE 20 IOT instruction ($\overline{20\ DIA}$). The AWR is reset to zero by the signal $\overline{IORESET}$ and the cylinder portion of the AWR is reset to zero when a RTZS operation is performed (21 IOPLS). The signal $\overline{HEAD\ OVERFLOW}$ is generated whenever the head portion (B13) of the AWR is loaded with an invalid head address, i.e., 10 through 15. The sector/head portion (A13 and B13) of the AWR is incremented when the Read/Write heads enter the DATA/CHVWD field of the sector by the signal $\overline{FDW}$.

2.4 FUNCTION, MODE, KEYWORD REGISTER

Figure 3-3 contains the FUNCTION, MODE, KEYWORD REGISTER (FMK). The FMK register is double ranked to allow adjacent sector processing. The FMK1 (1st rank of FMK) is loaded when the computer executes a DATA OUT C, DEVICE CODE 20 IOT instruction ($\overline{20\ DOC}$) if FMK1 doesn't already contain valid data (FMK FULL is reset). The trailing edge of $\overline{20\ DOC}$ sets FMK FULL. When the information in FMK1 is to be used, it is transferred to FMK2 (2nd rank of FMK) and FMK FULL is reset. This FMK1 to FMK2 transfer is controlled by

the DTR SEQUENCER (FIG 3-24). If a load FMK attempt is made when the FMK1 contains valid data (FMK FULL is set), the signal $\overline{\text{BFSET}}$ generates a buffer full interrupt request. Whenever an IORESET command is issued, the function register is cleared to read (00) and the mode register is set to 11 (both end of sector and beginning of data interrupts generated).

2.5 MEMORY ADDRESS REGISTER

Figure 3-4 contains the MEMORY ADDRESS REGISTER (MAR). The MAR is double ranked to allow adjacent sector processing and contains the address of the computer memory cell into which the current data word is to be written or from which the next data word is to be obtained. The MAR1 (1st rank of MAR) is composed of four 4 bit latches and is loaded when the computer executes a DATA OUT B, DEVICE CODE 20 IOT instruction ($\overline{\text{20 DOB}}$) if MAR1 doesn't already contain valid data (MAR FULL is reset). The trailing edge of $\overline{\text{20DOB}}$ sets MAR FULL. When the information in MAR1 is to be used, it is transferred to MAR2 (2nd rank of MAR) and MAR FULL is reset. This MAR1 to MAR2 transfer is controlled by the DTR SEQUENCER (FIG 3-24). If a load MAR attempt is made when the MAR1 contains valid data (MAR FULL is set), the signal BFSET (FIG 3-3) generates a buffer full interrupt request.

MAR2 is composed of four 4 bit binary counters and is incremented once for each data word transfer (COUNT ADV, see DATA CHANNEL CONTROL LOGIC, FIG 3-16). The contents of MAR2 is gated onto the I/O bus at the beginning of every memory reference ($\overline{\text{ADD ENABLE}}$, see DATA CHANNEL CONTROL LOGIC, FIG 3-16) and when the computer executes a DATA IN B, DEVICE CODE 20 IOT instruction ($\overline{\text{20 DIB}}$).

2.6 WORD COUNT REGISTER

FIG 3-5 contains the WORD COUNT REGISTER. The WORD COUNT REGISTER is composed of two 4 bit binary counters and is used to keep track of the number of words transferred during the current read/write operation. The WORD COUNT REGISTER is incremented once for each data word transferred to or from memory (COUNT ADV, FIG 3-16) and cleared to zero at the beginning of each sector (R/W CLEAR, FIG 3-33) and by the IORESET command. The contents of the WORD COUNT REGISTER are read when the computer executes a DATA IN A, DEVICE CODE 21 IOT instruction ($\overline{\text{21 DIA}}$).

2.7 STATUS REGISTER

FIG'S 3-6, 3-7, 3-8, 3-9, and 3-10 contain the STATUS REGISTER. The 16 bit STATUS register, together with the DISABLE INTERRUPT REGISTER (FIG 3-11), allow the computer to keep close track of disk controller performance. The occurrence of any abnormal condition within the disk controller sets the appropriate bit position in the STATUS REGISTER. A device 21 interrupt is then sent to the computer if the appropriate bit of the DISABLE INTERRUPT REGISTER (FIG 3-11) is set and if the INT DIS flip flop (FIG 3-12) is clear.

Device 21 interrupts are grouped into six categories: protection faults disabled by bit 12 of the DISABLE INTERRUPT REGISTER; addressing errors, disabled by bit 13 of the DISABLE INTERRUPT REGISTER; timing and data transfer faults, disabled by bit 14 of the DISABLE INTERRUPT REGISTER; buffering faults, disabled by bit 15 of the DISABLE INTERRUPT REGISTER; fatal errors; always enabled; and start/end of sector flags, disabled by the mode portion of the FMK register (FIG 3-3).

Listed below are the interrupts (and their STATUS REGISTER bit positions) within each of the six device code 21 interrupt categories:

A) PROTECTION FAULTS (Disabled by bit 12 of DIS INTR REG)

WA, ST REG 1 (FIG 3-8) - Write Address error. Sets whenever a Write Address operation is contained in the function portion of the 2nd rank of the FMK (FIG 3-3) register. The only way to clear this interrupt is reload the FMK register with a different function. This interrupt can be disabled via bit 12 of the DIS INTR REG. Assuming that the programmer always enabled PROTECTION fault interrupts when he is doing computer-disk processing, he will get a WA interrupt if he inadvertently attempts to perform a Write Address operation. This provides the computer-disk system with a certain amount of data (on disk) destruction protection.

KF, ST REG 2 (FIG 3-8) - Keyword violation. Sets when the keyword portion of the FMK register doesn't equal the keyword portion of the ADDRESS/KYWD field read from the disk during a Write or Erase function. This keyword violation interrupt protects against non-authorized users writing on protected areas of the disk. Nonauthorized users are defined as those users who don't have the correct keyword.

B) ADDRESSING ERRORS (Disabled by bit 13 of DIS INTR REG)

CA, ST REG 3 (FIG 3-7) - Cylinder address error. Set when the cylinder portion of the AWR register (FIG 3-2) doesn't equal the cylinder portion of the ADDR/KYWD field read from the disk during a read, write, or erase operation.

HA, ST REG 4 (FIG 3-7) - Head Address error. Set when the head portion of the AWR register (FIG 3-2) doesn't equal the head portion of the ADDR/KYWD field read from the disk during a read, write, or erase operation. Also set when the head portion of the AWR register (FIG 3-2) is loaded with an invalid head address (i.e., 10 through 15).

SA, ST REG 5 (FIG 3-7) - Sector address error. Set when the sector portion of the AWR register (FIG 3-2) doesn't equal the sector portion of the ADDRESS/KYWD field read from the disk during a read, write, or erase operation.

C) TIMING/DATA TRANSFER ERRORS (Disabled by bit 14 of DIS INTR REG)

CH, ST REG 7 (FIG 3-8) - Checkword error. Set when the checkword portion of the DATA/CHKWD field read from the disk doesn't agree with the checkword generated by the disk controller when it reads in the data portion of the DATA/CHKWD field during a read operation.

PA, ST REG 8 (FIG 3-9) - Parity error. Set if a parity error occurs when reading the address portion of the sector (during a read, write, or erase operation) or when reading the data portion of the sector (during a read operation). Parity is checked on an 8 bit byte basis.

D) BUFFERING FAULTS (Disabled by bit 15 of DIS INTR REG)

LD, ST REG 9 (FIG 3-9) - Lost data error. Set when a disk controller initiated memory request isn't honored by the computer within the given time frame (14 usec). The effect of a lost data error is that a word of data is lost because it hasn't been transferred to or from memory when the disk controller is ready to process the next word in the data transfer operation.

BF, ST REG 10 (FIG 3-10) - Buffer full fault. Set when an attempt is made to load the FMK (FIG 3-3) or MAR (FIG 3-4) registers when the 1st rank of these registers already contains valid (unused) information.

BF, ST REG 11 (FIG 3-10) - Request buffer full fault. Set if a seek operation request (20 IOPLS, FIG 3-20) is made by the computer when the seek sequencer has another seek operation request active. Also set if a data transfer operation request ($\overline{20\ START}$, FIG 3-24) is made by the computer when the DTR sequencer has another data transfer operation request active.

E) FATAL ERRORS (cannot be disabled at DIS INTR REG level)

NR, ST REG 12 (FIG 3-10) - Disk not ready. Set whenever the disk has not reached operating speed or the R/W heads aren't loaded.

NE, ST REG 13 (FIG 3-10) - Seek error. Set whenever the disk R/W heads have been moved past track 0 in the reverse direction or track 99 in the forward direction. A seek error can only be cleared by executing a RTZS operation.

DPU, ST REG 14 (FIG 3-10) - Disk pack unsafe. Set whenever more than one disk R/W head is selected, when the erase gate is on while the write gate is off, or when the read, write or erase gate is on while the R/W heads aren't on cylinder.

F) START/END OF SECTOR FLAGS

AWR, ST REG 0 (FIG 3-6) - Address word request. Set when the first data word is written or read from the sector. At this point, the disk controller is ready to accept another read/write operation request and to load new read/write operation parameters into the AWR, FMK, and MAR. The AWR interrupt is enabled by setting MODE bit zero in the FMK register.

EOR, ST REG 15 (FIG 3-6) - End of record. Set when last data word is written or read from the sector. At this point, the disk controller has completed the read/write operation. The EOR interrupt is enabled by setting MODE bit 1 in the FMK register.

The contents of the STATUS REG is read when the computer executes a DATA IN C, DEVICE CODE 20 IOT instruction (20 DIC). The STATUS REG is cleared when the computer executes a DEVICE CODE 21 IOT instruction with the F field equal to 10 ($\overline{21\ CLEAR}$) or when the IORESET command is executed (See FIG 3-12).

2.8 DISABLE INTERRUPT REGISTER

FIG 3-11 contains the DISABLE INTERRUPT REGISTER. As explained in Sec. 2.7 on the STATUS REGISTER, the DISABLE INTERRUPT REGISTER is used to disable 4 of the 6 categories of device 21 interrupts. The DISABLE INTERRUPT REGISTER is loaded with the complement of the information contained in bit 12-bit 15 of an accumulator (1 in AC→disables, 0 in AC→enables) when the computer executes a DATA OUT A, DEVICE 21 IOT instruction (21 DOA). Execution of an IORESET command results in setting the DISABLE INTERRUPT REGISTER, i.e., enables all interrupts.

FIG 3-11 also contains the logic that generates the signal $\overline{\text{INT EN}}$. $\overline{\text{INT EN}}$ is used to set a Device 21 interrupt (FIG 3-12). $\overline{\text{INT EN}}$ is generated if we have an interruptable condition (any of the STATUS REGISTER bits set), if the appropriate DIS INTR REG bit is set, and if the INT DIS flip-flop (FIG 3-12) is reset.

2.9 INTERRUPT LOGIC (DEV 21)

FIG 3-12 contains the DEVICE 21 INTERRUPT LOGIC. A device 21 interrupt request is sent to the computer when the flip flop INT REQ (N6) is set by the occurrance of the signals $\overline{\text{INTEN}}$ (from the disk controller) and RQENB (from the computer). As explained in Sec. 2.8 on the DISABLE INTERRUPT REGISTER, the signal $\overline{\text{INTEN}}$ is generated if any of the 16 disk controller abnormal conditions occur (i.e., if the abnormal condition sets its associated bit in the STATUS REG), if the appropriate DIS INT REG bit is set, and if the INT DIS flip flop (N6, FIG 3-12) is reset.

When the computer has received the interrupt, the programmer will typically execute the instruction INTERRUPT ACKNOWLEDGE which generates the signal $\overline{\text{INT ACK}}$ used to gate device code 21 (gates D7 in FIG 3-12) onto the I/O bus. The execution of the INTERRUPT ACKNOWLEDGE instruction tells the programmer what device caused the interrupt.

The signal $\overline{\text{RESET STATUS}}$ is used to reset the STATUS REGISTER and the INT REG flip flop. $\overline{\text{RESET STATUS}}$ is generated when the computer executes a DEVICE CODE 21 IOT instruction with the F field equal to 10 (21 CLEAR) when the IORESET command is executed. All Device 21 interrupts can be disabled if the INT DIS flip-flop is set (by executing a MASK OUT IOT instruction with bit 8 = 1).

2.10 IOT INSTRUCTION DECODE LOGIC

FIG 3-13 contains the IOT instruction decode logic for the device 21 IOT instructions and FIG 3-14 contains the IOT instruction decode logic for device code 20 IOT instructions. See Appendix A of the manual "How to Use the NOVA Computers" for a general discussion of IOT logic and timing. The specific disk controller IOT instructions, how they are decoded, and how they are used is given below:

21DOA - This signal is generated when the CP executes a DATA OUT A IOT instruction with a device code of 21 and is used to load the lower 4 bits of the I/O bus into the INTERRUPT DISABLE register (FIG 3-11).

$\overline{21\ CLEAR}$ - This signal is generated when the CP executes an IOT instruction with a device code of 21 and the F field equal to 10. $\overline{21\ CLEAR}$ is used to clear the STATUS register (FIG 3-6 to FIG 3-10).

$\overline{21DIA}$ - This signal is generated when the CP executes a DATA IN A IOT instruction with a device code of 21 and is used to gate the WORD COUNT register (FIG 3-5) onto the I/O bus.

21IOPLS, $\overline{21IOPLS}$ - This signal is generated when the CP executes on IOT instruction with a device code of 21 and the F field equal to 11. 21 IOPLS is used to initiate a RTZS operation, set the BUSY flip-flop (FIG 3-15) and zero the cylinder portion of the AWR register (FIG 3-2).

IORESET, $\overline{IORESET}$ - This signal is generated when the CP executes the IORST instruction or the RESET switch on the front panel of the NOVA is depressed. IORESET is used to clear all flags and control registers in the disk controller.

$\overline{SELB}$ - This signal is generated when the CP executes a BUSY skip instruction with a device code of 21 and the disk controller is performing a R/W operation.

$\overline{20\ START}$ - This signal is generated when the CP executes an IOT instruction with a device code of 20 and the F field = 01. $\overline{20\ START}$ is used to initiate a R/W operation.

$\overline{20\ CLEAR}$ - This signal is generated when the CP executes an IOT instruction with a device code of 20 and the F field = 10. $\overline{20\ CLEAR}$ is used to clear the BUSY, DONE and INT REQ flip-flops (FIG 3-15).

$\overline{20\ IOPLS}$ - This signal is generated when the CP executes an IOT instruction with a device code of 20 and the F field = 11. $\overline{20\ IOPLS}$ sets the BUSY flip-flop (FIG 3-15) and initiates a SEEK operation.

$\overline{20\ DIA}$ - This signal is generated when the CP executes a DATA IN A IOT instruction with a device code of 20 and is used to gate the contents of the AWR (FIG 3-2) onto the I/O bus.

$\overline{20DOA}$ - This signal is generated when the CP executes a DATA OUT A IOT instruction with a device code of 20 and is used to load the contents of the I/O bus into the AWR (FIG 3-2).

$\overline{20\ DIB}$ - This signal is generated when the CP executes a DATA IN B IOT instruction with a device code of 20 and is used to gate the contents of the MAR2 (FIG 3-4) onto the I/O bus.

$\overline{20\ DOB}$ - This signal is generated when the CP executes a DATA OUT B IOT instruction with a device code of 20 and is used to load the contents of the I/O bus into the MAR1 (FIG 3-4) assuming MAR1 is empty.

$\overline{20\ DIC}$ - This signal is generated when the CP executes a DATA IN C IOT instruction with a device code of 20 and is used to gate the contents of the STATUS register (FIG 3-6 to FIG 3-10) onto the I/O bus.

$\overline{20\ DOC}$ - This signal is generated when the CP executes a DATA OUT C IOT instruction with a device code of 20 and is used to load the contents of the I/O bus into the FMK register (FIG 3-3) assuming the FMK register is empty.

2.11 BUSY, DONE, INTR LOGIC (DEV CODE 20)

See Appendix A of the Manual "How to Use the NOVA computers" for a general discussion of the Busy, Done, and Interrupt logic and timing. FIG 3-15 conains the BUSY, DONE and INTR logic associated with disk controller SEEK and RTZS operations. The basic sequence of events is as follows: $\overline{20\ IOPLS}$ (SEEK operation) or $\overline{21\ IOPLS}$ (RTZS operation) sets the BUSY flip-flop. When the SEEK or RTZS operation completes a $\overline{SET\ DONE}$ signal is generated. This $\overline{SET\ DONE}$ signal sets the DONE flip-flop. Setting DONE will in turn set the INT REQ flip-flip if the INT DIS flip-flop is reset. INT DIS is controlled by AC bit 9 (1 sets, 0 resets) when the CP executes the MASK OUT instruction. When the CP recognizes the interrupt condition, it executes an INTERRUPT ACKNOWLEDGE instruction which generates the signal $\overline{INT\ ACK}$. $\overline{INT\ ACK}$ is used

to gate device code 20 onto the I/O bus. The CP, having determined the source of the interrupt, executes an IOT instruction with a device code of 20 and F = 10 which generates the signal $\overline{\text{20 CLEAR}}$. $\overline{\text{20 CLEAR}}$ resets the BUSY, DONE, and INT REQ flip-flops. The CP may test the states of BUSY and DONE by executing the BUSY/DONE shik instructions with a device code of 20. IORESET clears the INT DIS, BUSY, and DONE flip-flops.

2.12 DATA CHANNEL CONTROL LOGIC

See appendix A of the manual "How to Use the NOVA Computers" for a general discussion of the Data Channel logic and timing. FIG 3-16 contains the disk controller DATA CHANNEL CONTROL logic and FIG 4-1 contains the disk controller DATA CHANNEL CONTROL timing.

The function of the Data Channel control logic is to control the transfer of information between the disk controller and core memory. The Data Channel Control logic is initiated when the signal $\overline{\text{MEM REQ}}$ (FIG 3-17) sets the DCH SYNC and DCH BUSY (FIG 3-17) flip-flops. The direction of the data transfer is determined by the levels of the signals $\overline{\text{DCHM0}}$ and $\overline{\text{DCHM1}}$. When the disk controller is doing a Write Address or Write function, memory reads are performed ($\overline{\text{DCHM0}}$ and $\overline{\text{DCHM1}}$ are both high) and when the disk controller is doing a Read function, memory writes are performed ($\overline{\text{DCHM0}}$ is low and $\overline{\text{DCHM1}}$ is high).

When RQENB makes a low to high transistion with DCH SYNC set, the DCH REQ flip-flop is set. Assuming no other devices are using the Data Channel, the CP responds with a DCHA signal. This signal is used to gate MAR2 (FIG 3-4) onto the DATA lines (the signal $\overline{\text{ADD ENABLE}}$)and to increment MAR2 (FIG 3-4) and the WC (FIG 3-5) register (trailing edge of the signal COUNT ADV.). Then depending on the direction of the transfer, the computer responds with a $\overline{\text{DCH SEL}\cdot\text{DCH0}}$ (for memory reads) or a $\overline{\text{DCH SEL}\cdot\text{DCHI}}$ (for memory writes) signal. $\overline{\text{DCH SEL}\cdot\text{DCH0}}$ is used to load the WRITE BUFFER (FIG 3-44). $\overline{\text{DCH SEL}\cdot\text{DCHI}}$ is used to gate the READ BUFFER (FIG 3-45) onto the DATA bus. The trailing edge of the logical or of the signals $\overline{\text{DCH SEL}\cdot\text{DCH0}}$ and $\overline{\text{DCH SEL}\cdot\text{DCHI}}$ is used to reset the DCH BUSY flip-flop.

2.13 MEMORY REQUEST LOGIC

FIG 3-17 contains the MEMORY REQUEST logic and FIG 4-2 contains the MEMORY REQUEST timing. The Memory Request logic drives the Data Channel logic (FIG 3-16) via the signal $\overline{\text{MEM REQ}}$, controls the WRITE BUFFER (FIG 3-44) to IN-OUT SHIFT REG. (FIG 3-46) data transfer via the signal $\overline{\text{LOAD SR}}$, and detects data transfer latency errors (LOST DATA errors).

When the disk controller is performing a Write Address function, the Memory Request logic generates a 2 word memory read sequence. When the disk controller is performing a Write function, the Memory Request logic generates an 85 word memory read sequence. As shown in FIG 4-2, memory reads must be initiated one character count time before their contents are to be written on the disk (so that sufficient time is allowed for memory access and data channel latency). The WRITE BUFFER (FIG. 3-44), which is loaded with the word read from memory by the Data Channel logic, is transferred into the IN-OUT SHIFT register (FIG 3-46) at the trailing edge of the preceding character count time. If the data channel hasn't completed the preceding memory read when the WRITE BUFFER to IN-OUT SHIFT register transfer is attempted, a LOST DATA error results.

When the disk controller is performing a Read function the Memory Request logic generates an 85 word memory write sequence. As shown in FIG 4-2, memory writes are initiated at the beginning of the character count time following the completion of a disk to IN-OUT SHIFT register (FIG 3-46) assembly process. The IN-OUT SHIFT register to READ BUFFER (FIG 3-45) data transfer is free running and occurs just before the memory write is initiated and just after the disk to IN-OUT SHIFT register assembly is complete. If the data channel hasn't completed the preceding memory write when the IN-OUT SHIFT register to READ BUFFER transfer is attempted, a LOST DATA error results.

2.14 SYSTEM CLOCK

FIG 3-18 contains the SYSTEM CLOCK logic. The System Clock, a crystal controlled colpitts oscillator with a frequency of 5MHZ, is used as the basic disk controller timing source. The output of Q2, a non-symmetrical clipped sine wave is used to drive toggle flip-flop IA6. The output of IA6 (RAW CLOCK) is a 2.5 MHZ square wave.

RAW CLOCK is used to drive the SEEK (FIG 3-20, 3-21), RTZS (FIG 3-23) and DTR (FIG 3-24) sequencers via the Sequencer Clock (FIG 3-19) and the Read/Write Control logic via the WRITE CLOCK (FIG 3-19) and the Read/ Write Control logic via the WRITE CLOCK (FIG 3-37), BIT COUNTER (FIG 3-38), and the CHARACTER COUNTER (FIG 3-39).

2.15 SEQUENCER CLOCK

The SEQUENCER CLOCK logic and timing is contained on FIG 3-19. The Sequencer Clock is used to drive the SEEK (FIG 3-20, 3-21), RTZS (FIG 3-23) and the DTR (FIG 3-24) sequencers. The two flip-flops, whose outputs are used to form a 400 ns pulse (Ø1) and two 200 ns pulses (Ø1.5 and Ø2), form a divide by 3 counter. The Ø1, Ø1.5, and Ø2 clock pulses are used by the sequencers for gating and clocking purposes.

2.16 SEEK SEQUENCER

FIGS 3-20 and 3-21 contain the SEEK SEQUENCER logic and FIG 4-3 contains the SEEK SEQUENCER timing. The SEEK SEQUENCER provides the control and generates the enables that are needed to process a SEEK operation. The execution of a SEEK operation results in the moving of the read/write heads to the cylinder specified by the AWR register (FIG 3-2).

A SEEK operation is initiated when the computer executes a device code 20 IOT instruction with the F field equal to 11(P). The execution of this instruction results in the generation of the signal $\overline{20\ \text{IOPLS}}$ which is used to set the flip-flop SK REQ (on its trailing edge). SK REQ remains set until the SEEK operation has been completed. Should a SEEK operation request be made while a SEEK operation is in progress (i.e., SK REQ is set), the signal $\overline{\text{RFSET(1)}}$ will be generated and will cause a request buffer full interrupt request to be made.

The flip-flop RDCYL begins the SEEK SEQUENCER timing chain and is set at the trailing edge of hte next clock phase if the controller is inactive (R/W BUSY clear, FIG 3-24) and the disk drive is ON CYLINDER. The RDCYL flip-flop is used to generate the signal READ CYLINDER SEL (FIG 3-29) which tells the disk drive to gate the contents of its current address register onto the A/C bus. The stroke signal A/C→CCA is then used to load the A/C bus into the CURRENT CYLINDER ADDRESS REGISTER (FIG 3-22).

Setting RDCYL results in the direct setting of the flip-flop SK BUSY. SK BUSY remains set until the SEEK operation is completed and is used to disable R/W operations (FIG 3-24) when SEEK operations are active.

The next sequencer state, SKWT1, is set at the trailing edge of the next clock phase (RDCYL is cleared at the same time). SKWT1 is a "do nothing" state which exists to provide time for the signal ZC (zero compare, FIG 3-22) to become stable. If at the end of the next clock phase ZC is true (i.e., the heads are presently positioned at the requested cylinder), no seek is required, and a branch is made to SKDONE. If ZC is false, DIFF is set and the sequencer cycles through to SKWT2 where control stops until the seek is completed (ON CYLINDER signal is returned from disk).

During the DIFF state, the difference between the present position of the read/write heads and the desired position of the read/write heads (FIG 3-25) are gated onto the A/C bus (DIFF→A/C) and the signal DIFFERENCE SELECT (FIG 3-29) is used to tell the disk that the A/C bus contains the difference count.

During the CYL state, the desired position of the read/write heads is gated onto the A/C bus (CYL ADD→A/C) and the signal CYLINDER SEL (FIG 3-29) is used to tell the disk that the A/C bus contains the desired cylinder address.

During the CTRL state, the SEEK command (forward or reverse) is gated onto the A/C bus (FWD/RVS→A/C) and the signal. CONTROL SELECT (FIG 3-31) is used to tell the disk that the A/C bus contains command and control information.

During the SKWT2 state, the SEEK sequencer waits for the seek operation to complete. When the disk has completed the seek operation, it responds with the signal ON CYLINDER and the SEEK sequencer moves from the SKWT2 state to the SKDONE state.

During the SKDONE state, the signal $\overline{\text{SET DONE}}$ is generated [which is used to set the DONE flip-flop (FIG 3-15)] and the SK REQ and SK BUSY flip-flops are reset.

2.17 CYLINDER DIFFERENCE LOGIC

FIG 3-22 contains the CYLINDER DIFFERENCE logic. The CYLINDER DIFFERENCE logic is used to compute the difference between the present location of the read/write heads and the desired location of the read/write heads. It consists of a CURRENT CYLINDER ADDRESS REG, an adder, a complementer and a comparator.

As explained in Section 2.16 on the SEEK SEQUENCER, the CURRENT CYLINDER ADDRESS REG is loaded with the present location of the read/write heads during the RDCYL state. During the SKWT1 state, a one's complement substract is performed [the cylinder portion of the AWR (FIG 3-2) minus the CURRENT CYLINDER ADDRESS REG] and the result is checked for zero. If it is zero, the signal ZC is generated and used to tell the SEEK SEQUENCER that the read/write heads are already at the desired position.

If the result is not zero, the sign of the result is saved in the flip-flop SK DIR. The sign of the result determines the direction of the seek. If the difference is positive (desired cylinder greater than current cylinder), the seek must be in the forward direction. If the difference is negative (desired cylinder less than current cylinder), the seek must be in the reverse direction.

Since the disk drive requires the absolute difference between the desired and current cylinder location, a complementer is used to convert negative results to positive results when the difference ocunt is gated onto the A/C bus during the DIFF state.

2.18 RTZS SEQUENCER

FIG 3-23 contains the RTZS SEQUENCER logic and FIG 4-4 contains the RTZS SEQUENCER timing. The RTZS SEQUENCER provides the control and generates the enables that are needed to process a RTZS operation. A RTZS operation returns the read/write heads to cylinder 0 and is the only means of clearing a SEEK error interrupt.

An RTZS operation is initiated when the computer executes a DEVICE CODE 21 IOT instruction with the F field equal to 11(P) or when the command IORESET is generated. Either the execution of the DEVICE CODE 21 instruction (which results in the generation of the signal 21 IOPLS) or the IORESET

command will set the RTZ REQ flip-flop. One clock cycle later, RTZ CTRL is set and RTZ SEQ is reset.

During the RTZ CTRL state, the command RTZS (A/C 6T) is gated onto the A/C bus and the signal CONTROL SELECT is used to tell the disk drive that the A/C bus contains command and control information.

At the trailing edge of the next clock phase, RTZ CTRL is reset and RTZ WAIT is set. The RTZS sequencer remains in the RTZ WAIT state until the disk drive completes the RTZS operation (answers with ON CYLINDER). When ON CYLINDER appears, RTZ WAIT is reset and RTZ DONE is set. RTZ DONE remains set for 1 cycle and is used to generate the signal $\overline{\text{SET DONE}}$ (used to set the DONE flip-flop, FIG 3-15).

2.19 DTR SEQUENCER

FIG 3-24 contains the DTR SEQUENCER logic and FIG 4-5 contains the DTR SEQUENCER timing. The DTR SEQUENCER provides the control and the enables that are needed to process a read/write operation.

A read/write operation is initiated when the computer executes a DEVICE CODE 20 IOT instruction with the F field equal to S(01). The execution of this instruction results in the generation of the signal $\overline{\text{20 START}}$ which is used to set the flip-flop R/W REQ (on its trailing edge). R/W REQ is reset when the R/W sequencer starts its timing chain (i.e., when HEAD is set). If a read/write operation is requested while another read/write request is active (R/W REQ is set), the signal $\overline{\text{RFSET(2)}}$ will be generated and will cause a request buffer full interrupt request to be made.

The flip-flop HEAD starts the DTR timing chain and is set at the trailing edge of the next clock phase if the controller is inactive (SK BUSY clear, FIG 3-20), if the read/write heads are positioned over the requested cylinder (ON CYLINDER), and if a SEEK operation request is not pending (SK REQ, FIG 3-20).

During the HEAD state, the signal $\overline{\text{1RK→2RK}}$ is used to transfer MAR1 to MAR2 (FIG 3-4) and FMK1 to FMK2 (FIG 3-3). The signal $\overline{\text{CLR FULL}}$ is then used to clear the MAR FULL (FIG 3-4) and the FMK FULL (3-3) flip flops. The desired head address (from head portion of AWR, FIG 3-2) is gated onto the A/C bus (HEAD ADD→A/C) and the signal HEAD SELECT is used to tell the disk drive that the A/C bus contains the desired head address.

R/W BUSY is direct set when HEAD is set and remains set until the read/write operation is complete (STOP R/W, FIG 3-40). R/W BUSY is used to disable the SEEK SEQUENCER and provide the DEVICE CODE 21 BUSY skip instruction with the skip parameter (FIG 3-13).

At the end of the next clock phase, HEAD is reset and DTR is set. DTR is used to initiate the read/write control logic (FIG 3-33). DTR is cleared when the read/write operation is finished (STOP R/W, FIG 3-40).

2.20 A/C BUS

FIG 3-25 contains the A/C BUS ENABLE logic and FIG's 3-26, 27, and 28 contain the A/C BUS TRANSMITTER and RECEIVER logic. The A/C bus is a bi-directional bus used to transfer address and control information between the disk controller and the disk drive. One of six discrete "select" signals is used to tell the disk drive what information is presently on the A/C bus (see Table 2-1). FIGS 2-1 and 2-2 show the transmission scheme used by the A/C bus transmitters and receivers.

Table 2-1 A/C BUS TRUTH TABLE

	SELECT					
	RD CYC	DIFF	CYL	SECT	HEAD	CONTROL
A/C 0	2^0	2^0	2^0	2^0	2^0	Write Gate
A/C 1	2^1	2^1	2^1	2^1	2^1	Read Gate
A/C 2	2^2	2^2	2^2	2^2	2^2	SEEK FWD
A/C 3	2^3	2^3	2^3	2^3	2^3	---
A/C 4	2^4	2^4	2^4	--	--	Erase Gate
A/C 5	2^5	2^5	2^5	--	--	Seek Rev.
A/C 6	2^6	2^6	2^6	--	--	RTZ5

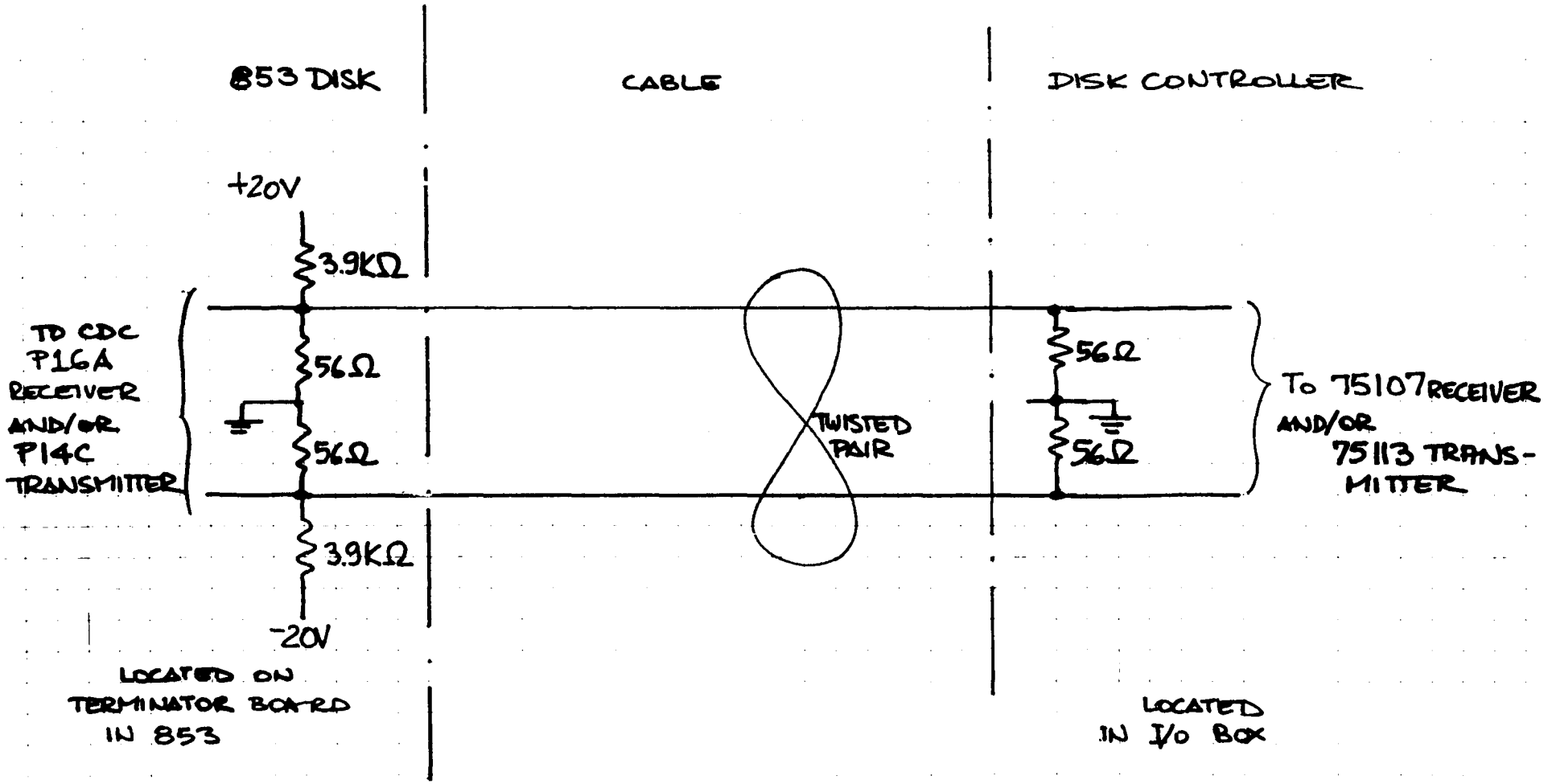
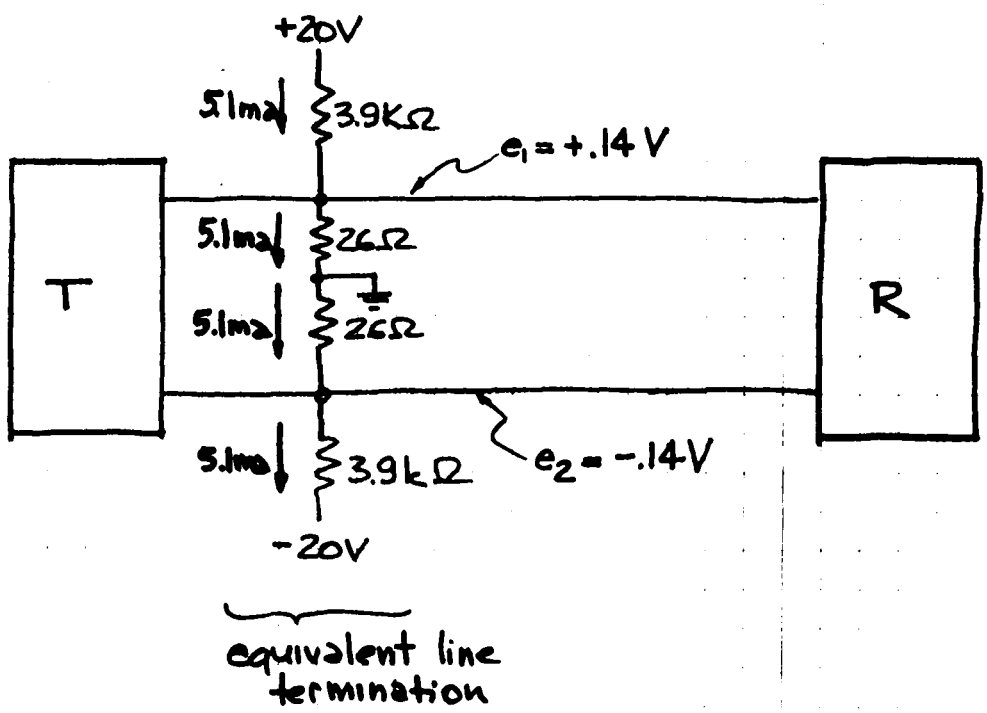
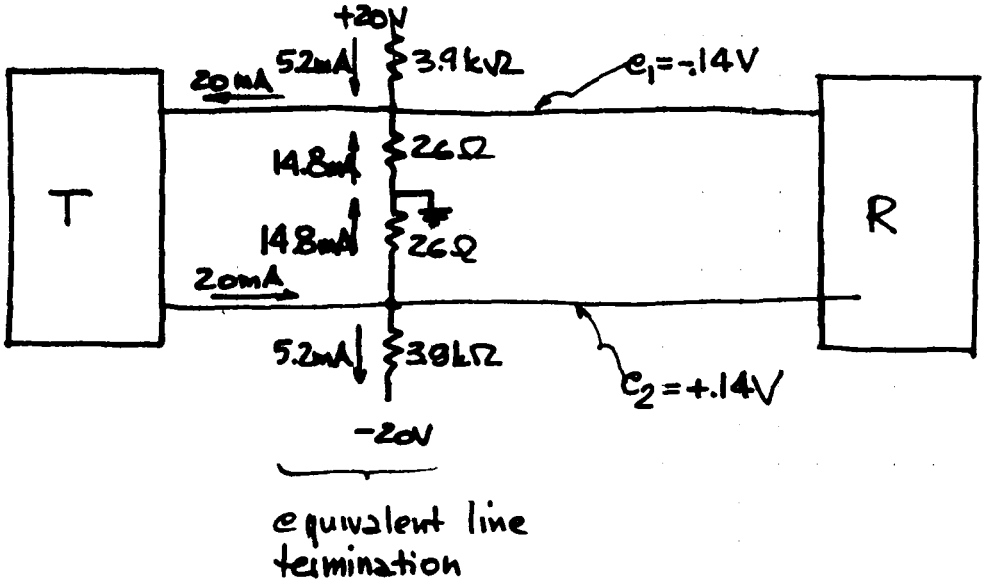


FIG. 2-1 DISK TO DISK CONTROLLER TRANSMISSION LINE

LOGICAL 0 ON TRANSMISSION LINE



LOGICAL 1 ON TRANSMISSION LINE



When a transmitter is turned on it sources 20 mA on one output & sinks 20mA on the other output

FIG-2-2 LOGIC LEVELS ON TRANSMISSION LINE

2.21 CONTROL TRANSMITTER/RECEIVERS

FIG'S 3-29, 30, and 31 contain the CONTROL TRANSMITTER/RECEIVER logic and FIGS 2-1 and 2-2 show the transmission scheme used by the CONTROL TRANSMITTER/RECEIVERS. The CONTROL TRANSMITTER/RECEIVER logic contains control and status discretes and, together with the A/C BUS TRANSMITTERS/ RECEIVERS, form the interface between the controller and the disk drive. These control and status discretes are listed below:

1) READ CYLINDER SELECT (FIG 3-29) - This signal tells the disk drive to gate its CURRENT ADDRESS REG onto the A/C bus.

2) CYLINDER SELECT (FIG 3-29) - This signal tells the disk drive that the A/C bus contains the address of the desired cylinder.

3) DIFFERENCE SELECT (FIG 3-29) - This signal tells the disk drive that the A/C bus contains the absolute difference between the present location of the read/write heads and the desired location of the read/write heads.

4) HEAD SELECT (FIG 3-29) - This signal tells the disk drive that the A/C bus contains the head address to be used for the next read/write operation.

5) ON CYLINDER (FIG 3-30) - This signal tells the controller that the read/write heads are stationary and positioned over a cylinder.

6) DPU (FIG 3-30) - This signal tells the controller that the "D"isk "P"ack is "U"nsafe. The DPU signal usually results from trying to perform a read/write operation when the read/write heads aren't "ON CYLINDER".

7) INDEX (FIG 3-30) - This signal tells the controller that the read/ write heads are entering "Sector 0". This signal occurs once for each revolution of the disk pack.

8) SECTOR MARKS (FIG 3-30) - This signal tells the controller that the read/write heads are entering the beginning of a sector. This signal occurs 16 times (there are 16 sectors per track) for each revolution of the disk pack.

9) SUR (FIG 3-30) - This signal tells the controller that the "S"elected "U"nit is "R"eady (the disk pack is selected and the read/write heads are loaded).

10) SEEK ERROR (FIG 3-30) - This signal tells the controller that the read/write heads have been moved past track 99 in the forward direction or track 0 in the reverse direction during a SEEK operation. This signal is cleared by executing a RTZS operation.

11) WRITE DATA (FIG 3-31) - This signal contains the information (in double-frequency format) to be written on the disk.

12) READ DATA (FIG 3-31) - This signal contains the information (in double-frequency format) that is read from the disk.

13) CONTROL SELECT (FIG 3-31) - This signal tells the disk drive that the A/C bus contains command and control information (see TABLE 2-1).

14) UNIT SELECT 2 (FIG 3-31) - This signal selects the disk drive. It is always active (the disk drive is always selected because there is only 1 disk drive on the controller).

2.22 I/O BOX INDICATOR LAMPS

The I/O BOX INDICATOR LAMPS, shown in FIG 3-32, are mounted on the I/O BOX and provide a means of monitoring disk controller and disk drive operation.

The SUR (Selected Unit Ready) lamp is on when the disk is selected and the Read/Write heads are loaded. The SEEK ERROR lamp is on when the disk drive has performed a SEEK operation that moved the Read/Write heads past track 99 in the forward direction or track 0 in the reverse direction. The DPU (Disk Pack Unsafe) lamp is on when more than one head is selected, both write and read gates are on, the erase gate is on without the write driver on, or when the read, write or erase gates are on while the heads aren't on cylinder. The ON CYLINDER lamp is on when the read/write heads are stationary and positioned over one of the recording tracks.

2.23 ON SECTOR LOGIC

FIG 3-33 contains the ON SECTOR logic, FIG 4-6 contains SECTOR MARK timing, and FIG 4-7 contains ON SECTOR timing. The ON SECTOR logic records read/write head position (relative to sector) information, resets the disk controller read/write control logic, and activates the disk controller read/write control logic when the necessary conditions are met.

Each track of the disk is divided into 16 sectors. The On Sector logic uses a signal generated by the disk drive called SECTOR MARKS to detect when the read/write heads are entering a new sector. This 10 u-sec pulse appears 16 times for each revolution of the disk pack. To provide a starting or reference point for the On Sector logic, the signal INDEX is generated by the disk pack concurrently with SECTOR MARKS once for every revolution of the disk pack. These two signals allow the On Sector logic to synchronize and increment a 4 bit counter called SC (Sector Counter). The SC is used to hold the number of the sector currently under the read/write heads.

SECTOR MARKS drives a 3 bit shift register (SMTI, SMT2, and SMT3). As shown in FIG 4-6, the outputs of the SMT shift register are used to generate 4 nonoverlapping pulses; $\overline{\text{R/W CLEAR}}$, $\overline{\text{R/W START}}$, $\overline{\text{PRESET SC}}$, and INC SC. $\overline{\text{R/W RESET}}$ is generated at every SECTOR MARKS and is used to reset the Read/Write Control logic. $\overline{\text{R/W START}}$ is generated at SECTOR MARKS if the read/write heads are entering the sector specified by the sector portion of the AWR (FIG 3-1) register (OS), if the disk is ready (SUR), and if a read/write request is active (DTR). $\overline{\text{R/W START}}$ is used to activate the Read/Write Control logic. $\overline{\text{PRESET SC}}$ is generated when SECTOR MARKS and INDEX are concurrently active and is used to synchronize the SC (presets SC to 15). INC SC is generated at every SECTOR MARKS and is used to increment the SC. FIG 4-7 shows the timing relationship of these four signals and their effect on the contents of SC.

2.24 FUNCTION DECODE LOGIC

The FUNCTION DECODE logic, shown in FIG 3-34, decodes the disk controller read/write function bits from the FMK1 register (FIG 3-3) into the following signals: $\overline{\text{READ}}$, WRITE, $\overline{\text{WRITE ADDRESS}}$, and ERASE + WRITE. These signals provide the Read/Write Control logic with function information and are generated according to the following truth table (See FIG 2-3).

FMK_{14}	FMK_{15}	READ/WRITE FUNCTION
0	0	READ
0	1	WRITE
1	0	WRITE ADDR
1	1	ERASE

FIG 2-3 READ/WRITE FUNCTION TRUTH TABLE

2.25 READ, WRITE, and ERASE GATING

The READ, WRITE, and ERASE GATING logic is contained in FIG 3-35. The Read, Write, and Erase Gating logic generates two signals (READ ENABLE and WRITE ENABLE) that provide the Read/Write Control logic with sector processing coarse timing information. These two signals, when "anded" with R/W ACT, are also used to control the information transfer mode of the read/write heads. When the signal READ GATE is active the heads read information from the disk. When the signals WRITE GATE and ERASE GATE are active, the heads write information on the disk pack.

As shown in FIG 4-8, READ ENABLE and WRITE ENABLE (and hence READ GATE, WRITE GATE and ERASE GATE) depend on the read/write function being performed. When doing a Read operation, READ ENABLE is true and WRITE ENABLE is false for the entire sector. This means that the heads read both the address portion and the data portion of the sector. When doing a Write or Erase operation READ ENABLE is true during the address portion of the sector and WRITE ENABLE is true during the data portion of the sector. This means that the heads write both the address and data portions of the sector.

Note that R/W ACT is true for the entire sector regardless of what read/write operation is being performed. R/W ACT is used to drive the signal CONT SEL. CONT SEL is a signal that tells the disk drive that valid information is on the READ GATE, WRITE GATE, and ERASE GATE lines.

2.26 READ/WRITE CONTROL CLOCKING SOURCES

Data recorded on the disk is in double frequency format. Each bit writen on the disk has associated with it a clock pulse and a data pulse (present if a 1 bit is recorded and absent if a 0 bit is recorded). This double frequency format clock pulse/data pulse information packet is shown in FIG 2-4.

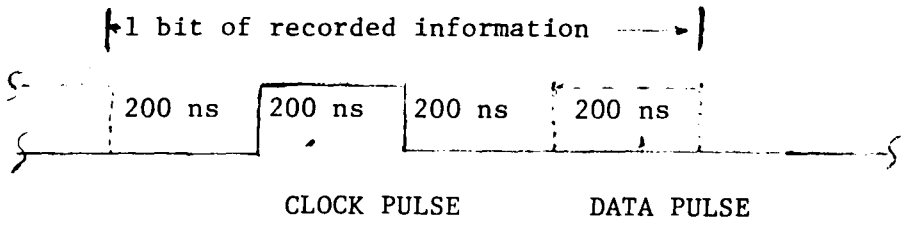


FIG 2-4 INFORMATION RECORDED IN DOUBLE FREQUENCY FORMAT

The Read/Write Control logic is driver by a 2 phase clocking system (Ψ_1 and Ψ_2) that is closely related to the double frequency format of the information recorded on the disk (See FIG 2-5).

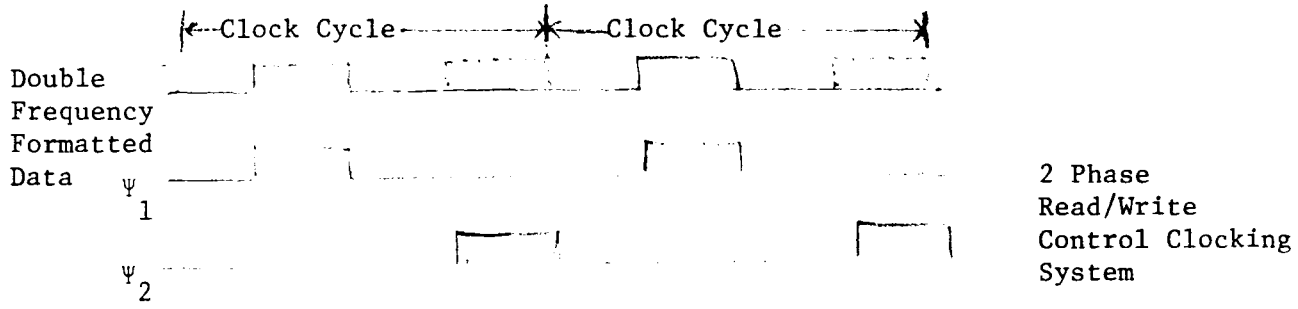


FIG 2-5 READ/WRITE CONTROL CLOCKING

This 2 phase clocking system is composed of two clocking sources, the READ CLOCK (FIG 3-36) and the WRITE CLOCK (FIG 3-37). Which clock source is used to drive the Read/Write Control logic depends on what read/write operation is being performed and what portion of the sector is under the read/write heads. A sector is divided into an address portion and a data portion.

The address portion of the sector contains address and keyword information. To verify sector access legality, the contents of the address portion of the sector is read and compared with the sector access parameters when the disk controller is performing a Read, Write or Erase function. The sector access parameters are contained in the keyword portion of the FMK register (FIG 3-3) and the AWR register (FIG 3-1). The Write Address function is used to write the address and keyword information on the address portion of the sector.

The data portion of the sector contains the stored data and is read when the disk controller is performing a Read function. When performing Write function, the disk controller writes data on the data portion of the sector. When performing a Write Address or Erase function, the disk controller writes zeroes on the data portion of the sector.

In general, the READ CLOCK drives the Read/Write Control logic when data is read from the disk and the WRITE CLOCK drives the Read/Write Control logic when data is written on the disk. FIG 4-8 shows the relationship between the read/write clocking sources and the read/write function being performed.

2.27 READ CLOCK

The READ CLOCK (FIG 3-36) is driven by the double frequency formatted information read from the disk (READ DATA). Transfer of control from the WRITE CLOCK to the READ CLOCK is always done when the read/write heads are 28 bits into a sector synchronization field (SYNC I field for the address portion of the sector, SYNC II field for the data portion of the sector). The sync fields contain a sequence of 35 zero bits followed by a 1 bit. Detection of this 1 bit tells the disk controller that the READ CLOCK is now synchronized with the data stream coming from the disk [i.e., the read/write heads are entering the ADDR/KYWD field (address portion of the sector) or the DATA/CHKWD field (data portion of the sector)].

The READ CLOCK must generate clock pulses and recover data from the double frequency formatted information stream read from the disk. For each double frequency formatted clock pulse / data pulse packet, the one shot LEAD fires on the leading edge of the clock pulse for 400 ns and the one shot TRAIL fires on the trailing edge of the clock pulse for 400 ns. The flip-flop DIS LEAD sets on the trailing edge of the LEAD pulse and together with the TRAIL pulse is used to inhibit LEAD from firing on the data pulse. DIS LEAD is reset after the TRAIL pulse and the data pulse have timed out. This enables LEAD for the next clock pulse/data pulse packet.

The Z_1 and Z_2 pulses are generated by decoding the outputs of LEAD and TRAIL and produce a 2 phase clocking system having the characteristics of Ψ_1 and Ψ_2 shown in FIG 2-5.

INPUT DATA, a flip-flop used to recover the data from the information stream is set when a data pulse is present in the clock pulse/data pulse packet and is reset at the next Z_1 time.

The Z_1 and Z_2 clocking pulses are not allowed to drive the Read/Write Control logic (via R_1 and R_2) until the sync bit is detected. The RCLK SYNC flip-flop is clocked by Z_2 and waits in the reset state for the sync bit to come along. When the sync bit (the 1 bit in the sync pattern) appears, RCLK SYNC sets and allows the Z_1 and Z_2 pulses to drive the Read/Write Control logic. FIG 4-10 shows detailed READ CLOCK timing.

2.28 WRITE CLOCK

FIG 3-37 contains the WRITE CLOCK. Both the WRITE CLOCK and the READ CLOCK (FIG 3-36) are idle until the On Sector logic (FIG 3-33) activates the Read/Write Control logic with the signal $\overline{\text{R/W START}}$ (FIG 3-33). $\overline{\text{R/W START}}$ sets the R/W ACT (FIG 3-40) and CLK CNTRL (FIG 3-37) flip-flops. CLK CNTRL is the signal that determines which clock source drives the Read/Write Control logic. When CLK CNTRL is true, the WRITE CLOCK is enabled and when CLK CNTRL is false, the READ CLOCK is enabled.

The logical "and" of R/W ACT and CLK CNTRL generates the signal $\overline{\text{ENA WCLK}}$ (FIG 3-37) which activates the WRITE CLOCK. The WRITE CLOCK is composed of two flip-flops, W1 and W2, driven by the System Clock signal RAW CLOCK (FIG 3-18). W1 is a mod 2 counter and W2 is a one stage shift register receiving the output of W1. The outputs of W1, W2, CLK CNTRL and R/W ACT are used to generate Ψ_1 and Ψ_2. 4-9 shows detailed WRITE CLOCK timing.

When the disk controller is performing a Read, Write, or Erase operation, the WRITE CLOCK remains enabled until the Read/Write heads are 28 bits into the SYNCI field. At this time ($CC5 \cdot BIT17 \cdot \Psi_2$), the CLK CNTRL flip-flop resets, disabling the WRITE CLOCK and enabling the READ CLOCK. See Section 2.27 for an explanation of control transfer from the WRITE CLOCK to the READ CLOCK.

After the ADDR/KYWD field has been read, the WRITE CLOCK is re-enabled. It remains enabled for the rest of the sector when a Write or Erase operation is being performed. If a Read operation is being performed, the WRITE CLOCK is again disabled 28 bits into the SYNC II field. If a Write Address operation is being performed, the WRITE CLOCK is enabled for the entire sector. FIG 4-8 shows WRITE CLOCK activity in relation to read/write function.

2.29 BIT COUNTER

FIG 3-38 contains the BIT COUNTER logic and FIG 4-11 contains the BIT COUNTER timing. The BIT COUNTER is a mod 18 counter used to provide the timing for the processing of each 18 bit word recorded or read from the disk pack. Typically each 18 bit word recorded on the disk pack consists of a 16 bit data word and 2 parity bits.

A BIT COUNTER consists of a 2 bitshift register (BIT 0, BIT17), a 4 bit binary counter (BC), on Excess 3 Gray Decoder (G2), and assorted control and decode logic. The BIT COUNTER is cleared to zero by $\overline{\text{R/W CLEAR}}$ (FIG 3-33) and initialized when the signal $\overline{\text{R/W START}}$ direct sets BIT 0. At the next clock phase (Ψ_2), BIT 0 is cleared and BC is enabled. BC increments through 16 counts and then at the end of BIT 16 time, BIT 17 is set and BC is disabled. At the next clock phase, BIT 0 is set, BIT 17 is reset, and the entire cycle repeats. See Table 2-2 for BIT COUNTER state table.

In addition to its normal mod 18 operation, the BIT COUNTER can be programmed to "SKIP 4 BITS" in its count cycle. This is accomplished by parallel loading BC with 12 (this represents BIT TIME 13 because BIT TIME is equal to the contents of BC + 1 when BC is enabled) at BIT TIME 9. The "SKIP 4 BITS" mode is used to program two bit count cycles of 14 bits each when the read/write heads are in the SYNC fields and the controller wants to transfer control from the WRITE CLOCK to the READ CLOCK. This allows the controller to enable the READ CLOCK Sync logic when the read/write heads are approximately 8 bits away from the sync bit (See Sec. 2-27 on READ CLOCK for explanation of READ CLOCK Sync logic).

TABLE 2-2 BIT COUNTER STATE TABLE

bit 0	BC				bit 17	BIT TIME	
06/pin9	J6/pin5	J6/pin9	J6/pin2	J6/pin12	G1/pin7		
0	0	0	0	0	0	--------	
1	0	0	0	0	0	0 (FLIP-FLOP06/9)	*
0	0	0	0	0	0	1	
0	1	0	0	0	0	2	
0	0	1	0	0	0	3	
0	1	1	0	0	0	4	
0	0	0	1	0	0	5	
0	1	0	1	0	0	6 (G2/pin4)	*
0	0	1	1	0	0	7	
0	1	1	1	0	0	8 (G2/pin1)	*
0	0	0	0	1	0	9 (FLIP-FLOPG1/2)	
0	1	0	0	1	0	10	
0	0	1	0	1	0	11	
0	1	1	0	1	0	12	
0	0	0	1	1	0	13	
0	1	0	1	1	0	14	
0	0	1	1	1	0	15	
0	1	1	1	1	0	16 (G2/pin9)	*
0	0	0	0	0	1	17 (FLIP-FLOPG1/7)	*

* These bit times are the only ones used by the disk controller.

2.30 CHARACTER COUNTER

FIG 3-39 contains the CHARACTER COUNTER logic. The CHARACTER COUNTER is a binary counter that tells the controller how far the read/write heads are into the sector. See FIG 4-8 on R/W CLOCK CONTROL for a definition of the sector subfields in relation to CHARACTER COUNT time.

The CHARACTER COUNTER consists of two 4 bit binary counters (13, 14), two BCD-decimal Decoders (J3, J4), and assorted decode logic. The CHARACTER COUNTER is cleared to zero by $\overline{\text{R/W CLEAR}}$ and then incremented at each BIT17$\cdot\Psi_2$ time until the end of the sector is reached (CC99). Specific Character Count times are decoded by taking the logical "and" of a J3 decoder output and J4 decoder output. See Table 2-3 for an explanation of this decode scheme.

TABLE 2-3 CHAR COUNT DECODER TRUTH TABLE

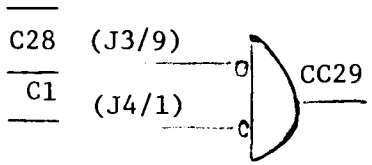
J3 CHAR COUNT DECODER

DECIMAL VALUE OF I3,I4 CHAR COUNT INPUTS				DECIMAL VALUE OF J3 OUTPUTS
32 ($\frac{J3}{12}$)	16 ($\frac{J3}{13}$)	8 ($\frac{J3}{14}$)	4 ($\frac{J3}{15}$)	
0	0	0	0	0(J3/1)
0	0	0	1	4(J3/2)
0	0	1	0	8(J3/3)
0	0	1	1	12(J3/4)
0	1	0	0	16(J3/5)
0	1	0	1	20(J3/6)
0	1	1	0	24(J3/7)
0	1	1	1	28(J3/9)
1	0	0	0	32(J3/10)
1	0	0	1	36(J3/11)

J4 CHAR COUNT DECODER

DECIMAL VALUE OF I3,I4 CHAR COUNT INPUTS				DECIMAL VALUE OF J4 OUTPUTS
-- ($\frac{J4}{12}$)	64 ($\frac{J4}{13}$)	2 ($\frac{J4}{14}$)	1 ($\frac{J4}{15}$)	
0	0	0	0	0(J4/1)
0	0	0	1	1(J4/2)
0	0	1	0	2(J4/3)
0	0	1	1	3(J4/4)
0	1	0	0	64(J4/5)
0	1	0	1	65(J4/6)
0	1	1	0	66(J4/7)
0	1	1	1	67(J4/9)
--	--	--	--	---
--	--	--	--	---

To decode a specific character count time, take logical and of a J3 output and a J4 output. The value of the decoded signed will be equal to the sum of thetwo inputs. For example:



or 28 + 1 = 29

2.31 READ/WRITE TIMING (1)

FIG 3-40 contains the READ/WRITE TIME(1) logic. This logic generates the three coarse timing signals used to control the reading and writing of information on the disk. These timing signals are listed below:

R/W ACT - Active when the read/write heads are over the sector being processed. Set at the beginning of the sector ($\overline{\text{R/W START}}$) when a read/write operation request is pending and normally reset when the read/write heads reach the end of the sector ($\overline{\text{EOR}}$). If a sector address error, a head address error, a cylinder address error, a keyword error, a lost data error, or a checkword error occurs during the processing of the sector, R/W ACT is cleared (STOP R/W) aborting the read/write operation.

ADDR TIME - Active when the read/write heads are over the address portion of the sector being processed. ADDR TIME is set at the beginning of the sector when a read/write operation is pending and reset at $CC7 \cdot BIT17 \cdot \Psi_2$ time.

DATA TIME - Active when the read/write heads are over the data portion of the sector being processed. DATA TIME is set at $CC8 \cdot BIT6 \cdot \Psi_2$ time and reset at the end of the sector ($\overline{\text{R/W CLEAR}}$).

FIG 4-8, R/W CLOCK CONTROL, shows the timing relationship between R/W ACT, ADDR TIME, and DATA TIME.

2.32 READ/WRITE TIMING (2 and 3)

Two timing generators (FIG 3-41) and associated decode logic (FIG 3-42) provide general timing information used to control the reading and writing of information on the disk. The "T" timing generator is a 4 bit shift register consisting of flip-flops TWMEM, TWLST, TAD/DT and TPULSE and the "L" generator is a 2 bit shift register consisting of flip-flops LAD/DT and LPULSE. The "T" timing generator is clocked on the trailing edge of $BIT17 \cdot \Psi_2$ time and the "L" timing generator is clocked on the trailing edge of $BIT0 \cdot \Psi_2$ time.

The first stage of the "T" generator (TWMEM) has an address cycle from the trailing edge of CC2·BIT17·Ψ_2 time to the trailing edge of CC4·BIT17·Ψ_2 time and a data cycle from the trailing edge of CC10·BIT17·Ψ_2 time to the trailing edge of CC85·BIT17·Ψ_2 time. Each of the succeeding stages of the T generator delays this basic cycle by one character count time. Note that the fourth stage of the T generator (TPULSE) has only a data cycle.

The first stage of the "L" generator (LAD/DT) has an address cycle from the trailing edge of CC6·BIT0·Ψ_2 time to the trailing edge of CC8·BIT0·Ψ_2 time and a data cycle from the trailing edge of CC12·BIT0·Ψ_2 time to the trailing edge of CC98·BIT0·Ψ_2 time. The second stage of the "L" generator (LPULSE) has only a data cycle and delays the basic cycle by one character count time.

FIG 4-12 contains timing diagrams for the timing generators and FIG 3-42 contains the logic used to generate timing signals derived from these generators.

2.33 WRITE DATA LOGIC

The WRITE DATA LOGIC (FIG 3-43) is used to generate the double frequency formatted serial data stream written on the disk ($\overline{\text{WRITE DATA}}$). $\overline{\text{WRITE DATA}}$ is enabled by the signals R/W ACT (FIG 3-40) and WRITE ENABLE (FIG 3-35). The four inputs to gate F7/8 generate the various fields of the sector being written on the disk. Gate 17/8 is enabled to write the EOR bit, gate I2/3 is enabled to write the sync bits, gate F1/12 is enabled to write the address, keyword and data fields, and gate I7/6 is enabled to write the checkword field. When none of these gates are active, zeroes are written on the disk.

2.34 WRITE BUFFER REGISTER

FIG 3-44 contains the WRITE BUFFER REGISTER logic. During a write or write address operation, 16 bit words are transferred from memory to the WRITE BUFFER and from the WRITE BUFFER to the IN/OUT SHIFT REGISTER (FIG 3-46). The DATA CHANNEL CONTROL logic (FIG 3-16) loads the WRITE BUFFER with the signal $\overline{\text{DCH SEL DCHO}}$.

2.35 READ BUFFER REGISTER

FUG 3-45 contains the READ BUFFER logic. The READ BUFFER is a 16 bit register that receives all information read from the disk. The READ BUFFER is loaded with the contents of the IN/OUT SHIFT REGISTER (FIG 3-46) at every BIT 0. Ψ_1 time and the DATA CHANNEL CONTROL logic (FIG 3-16) gates the contents of the READ BUFFER onto the I/O BUS ($\overline{\text{DCH SE2 DCHI}}$) when the read/write heads are over the data portion of the sector during a read operation. This READ BUFFER to I/O BUS transfer is part of the process necessary to write the data portion of the sector into memory.

2.36 IN/OUT SHIFT REGISTER

FIG 3-46 contains the IN/OUT SHIFT REGISTER. When writing information on the disk, the IN/OUT SHIFT REGISTER is used in a parallel in ($\overline{\text{LOAD SR}}$) - serial out (OUTPUT DATA) mode to disassemble the 16 bit (and 2 parity bits) words transferred from memory to the disk controller. When reading information from the disk, the IN/OUT SHIFT REGISTER is used in a serial in (INPUT DATA) - parallel out mode to assemble the serial bit stream read from the disk into 16 bit parallel words.

2.37 COMPARATORS (1 and 2)

FIG 3-47 and FIG 3-48 contain the COMPARATORS that are used, when the disk controller is performing a read/write operation, to verify that the correct sector is under the read/write heads and that the sector under the read/write heads may be written into (if a write or erase operation is being performed).

The CYLINDER comparator compares the cylinder portion of the AWR register (FIG 3-2) with READ BUFFER (FIG 3-45) bits 1-7. The HEAD comparator compares the head portion of the AWR register with READ BUFFER bits 8-11. The SECTOR comparator compares the sector portion of the AWR register with READ BUFFER bits 12-15. It is important to note that the only time that these comparators have meaningful results is when the first word of the ADDR/KYWD field of the sector under the read/write heads is in the READ BUFFER (CC7 time).

The KEYWORD comparator (FIG 3-48) compares the keyword portion of the FMK register (FIG 3-3) with READ BUFFER bits 5-11. It is important to note that the only time that the comparator has meaningful results is when the second word of the ADDR/KYWD field of the sector under the read/write heads is in the READ BUFFER (CC8 time).

2.38 CYL, HD, SECT, and KYWD ERROR LOGIC

FIG 3-49 contains the logic used to generate the cylinder, head, sector, and keyword errors. These errors are all saved in the STATUS REGISTER (FIG 3-6 to 3-10) and abort the read/write operation.

The CYLINDER, HEAD, and SECTOR comparators (FIG 3-47) are checked for valid comparisons during a read, write or erase operation (READ ENABLE) when the comparators contain valid results ($CC7 \cdot BIT17 \cdot \Psi_2$). These checks insure that the segment under the read/write heads is the segment desired by the programmer.

The KEYWORD comparator (FIG 3-48) is checked for a valid comparison during a write or erase operation (ERASE & WRITE) when the comparator contains valid results ($CC8 \cdot BIT17 \cdot \Psi_2$). This check insures that segments can only be written into when the programmer has the correct keyword.

2.39 CHECKWORD REGISTER

FIG 3-50 contains the CHECKWORD generation and test logic. The CHECKWORD REGISTER is a shift register used to generate an 18 bit longitudinal parity word called the checkword. This checkword is produced by taking the "exclusive or" of all data words contained in the DATA/CHKWD field.

When performing a write operation, the checkword is generated from the data written on the disk (OUTPUT DATA) and then is written into the last word of the DATA/CHKWD field. When performing a read operation, a checkword is generated using the data read from the disk (INPUT DATA) and then compared to the checkword read from the disk (by performing the "exclusive or" operation on the two checkwords). If the comparison fails, a checkword error occurs ($\overline{\text{CHKWD ERROR}}$).

2.40 PARITY GENERATION AND TEST

FIG 3-51 contains the PARITY GENERATION AND TEST logic and FIG 4-13 contains the parity check timing. Parity is generated when writing the ADDR/KYWD field during a write address operation and when writing the DATA/CHKWD field during a write operation. The parity generated is even with 1 bit generated for each 8 bit data byte and appears as bit 8 and bit 17 in each 18 bit serial word written on the disk.

Parity is checked when reading the ADDR/KYWD field during a read, write, or erase operation and when reading the DATA/CHKWD field during a read operation. The parity checker is a mod 2 counter that counts the number of ones in each "9 bit byte" read from the disk. If the number of ones is odd, a parity error results ($\overline{\text{PARITY ERROR}}$).

2.41 NOVA I/O BUS - DATA RECEIVE LOGIC

FIG 3-52 contains the NOVA I/O BUS DATA RECEIVE logic. The I/O BUS is a 16 bit bi-directional parallel bus used to transfer all information between the NOVA and the disk controller.

3.0 LOGIC DRAWINGS

3.1 NOTATION

This section contains the complete set of logic drawings (FIG 3-0 to FIG 3-52) for the disk controller. As explained in Sect. 1.2, the disk controller is contained on two boards that plug into the NOVA 1200 mainframe (the 4042 board and the Interface board) and 4 boards (the X board, the Y board, the IA-IC board, and the ID-IF board) that mount in the I/0 BOX. These 6 boards contain the TTL integrated circuits (chips) that make up the disk controller logic. Each chip is assigned an alphanumeric name that identifies its board and position when shown in the logic drawings. These names are assigned as follows:

1) 4042 BOARD (Data General 4042 logic) - $\overline{E1}$, $\overline{E2}$,..., $\overline{E48}$
2) 4042 BOARD (Disk Controller wire-wrapped logic) - A1 to A13, B1 to B13, C1 to C13, D1 to D13, and E1 to E13.
3) INTERFACE BOARD (Wire wrapped logic) - F1 to F8, G1 to G8, H1 to H8, I1 to I8, J1 to J8, K1 to K8, L1 to L8, M1 to M8, N1 to N8, 01 to 08, P1 to P8, and Q1 to Q8.
4) I/0 BOX (X Board) - X1 to X18
5) I/0 BOX (Y Board) - Y1 to Y18
6) I/0 BOX (IA-IC Board) - IA1 to IA6, IB1 to IB6, and IC1 to IC6
7) I/0 BOX (ID-IF Board) - ID1 to ID6, IE1 to IE6, and IF1 to IF6

See Appendix B for a physical layout of these boards and a list of chip type assignments.

Two cables, P2 and P3, connect the Interface Board with the I/0 BOX and one cable, P1, connects the 4042 Board with the I/0 BOX. Two cables, J100 and J102, connect the I/0 BOX with the 853 Disk. The 4042 Board and the Interface Board are connected together via 48 unused positions on the NOVA backplane. The NOVA I/0 BUS connects to both the 4042 Board and the Interface BOARD via the NOVA backplane.
and their interconnections.

FIG 1-1 shows these boards and their interconnections. To represent these interconnections in the logic drawings, the following notation is used:

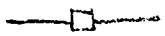
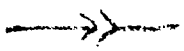
SYMBOL	MEANING
	Connection between the 4042 Board and the Interface Board via the NOVA backplane.
or	Connection to the NOVA I/O bus
	Wire wrap pin used as a connection point
	Cable termination point
DATA 6	Signal name, no bar implies an active "high" signal
$\overline{\text{DATA 6}}$	Signal name, bar implies an active "low" signal

FIG 3-0 of the logic drawings give examples of this logic notation.

3.2 LOGIC DRAWING LIST

3.3 LOGIC DRAWINGS

This section contains a complete set of logic diagrams (FIG 3-0 to 3052) for the Disk Controller.

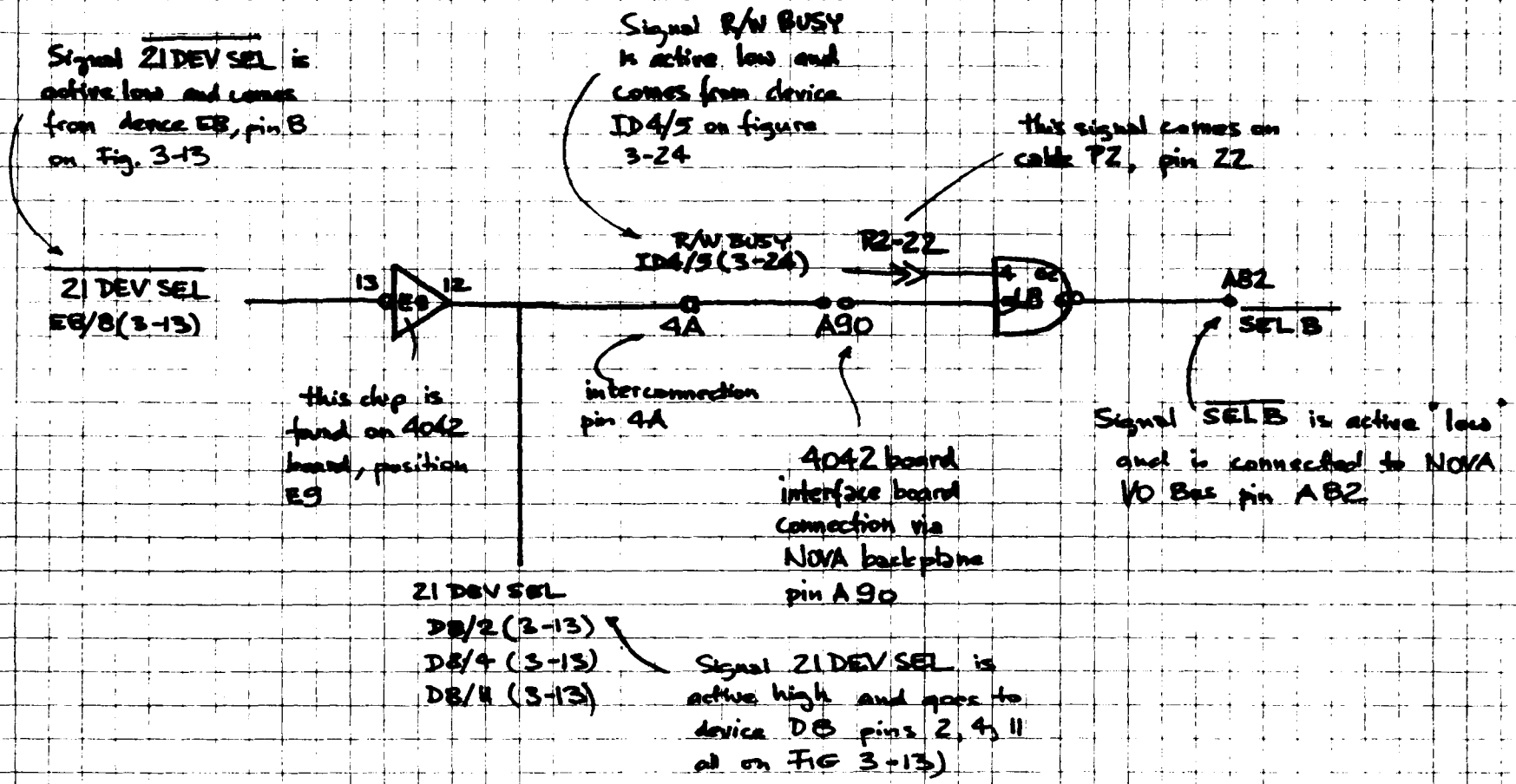


FIG. 3-0 LOGIC DIAGRAM NOTATIONS

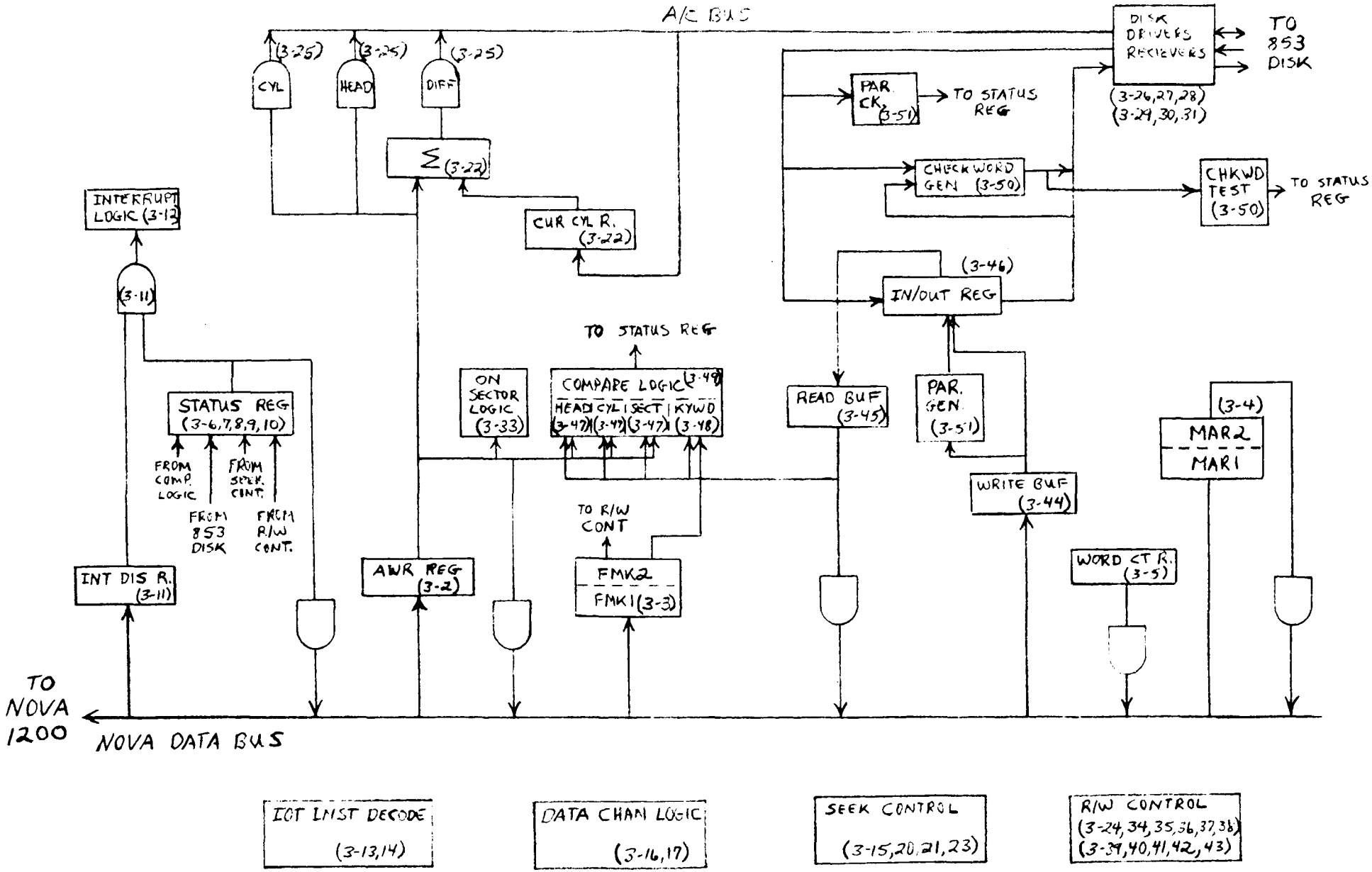


FIG 3-1 DISK CONTROLLER BLOCK DIAGRAM

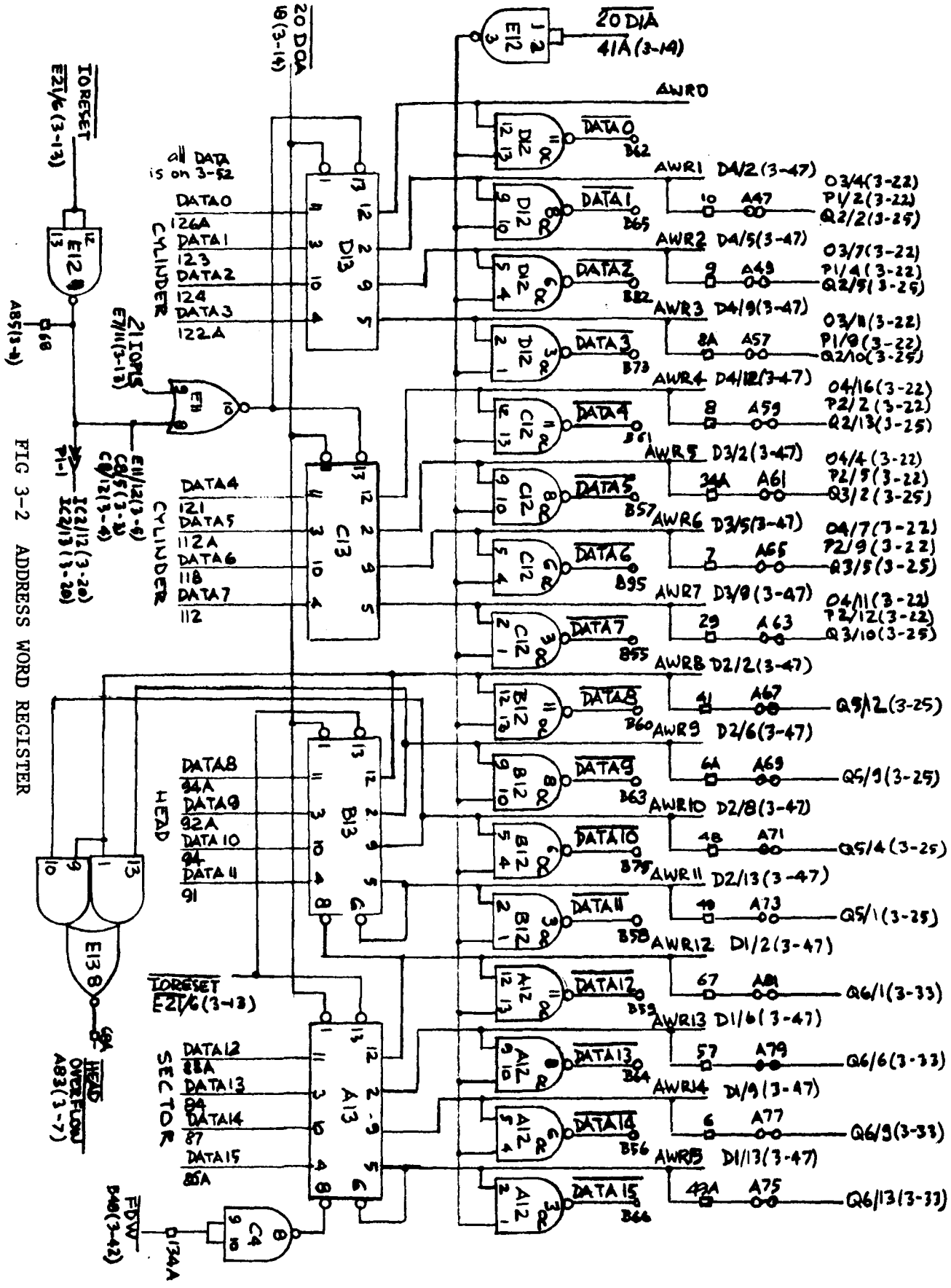


FIG 3-2 ADDRESS WORD REGISTER

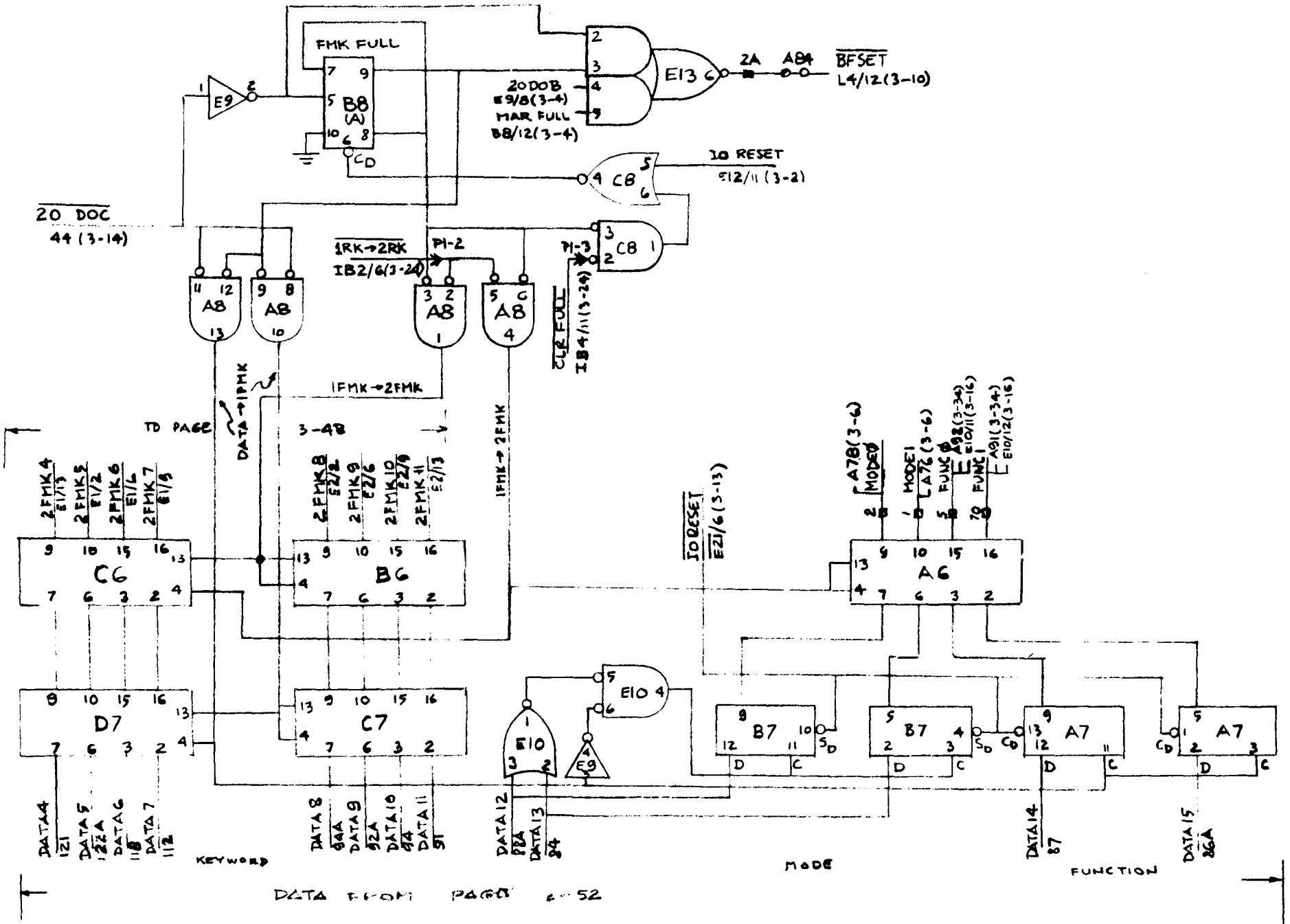


FIG 3-3 FUNCTION, MODE, AND KEYWORD REGISTER

REV 1

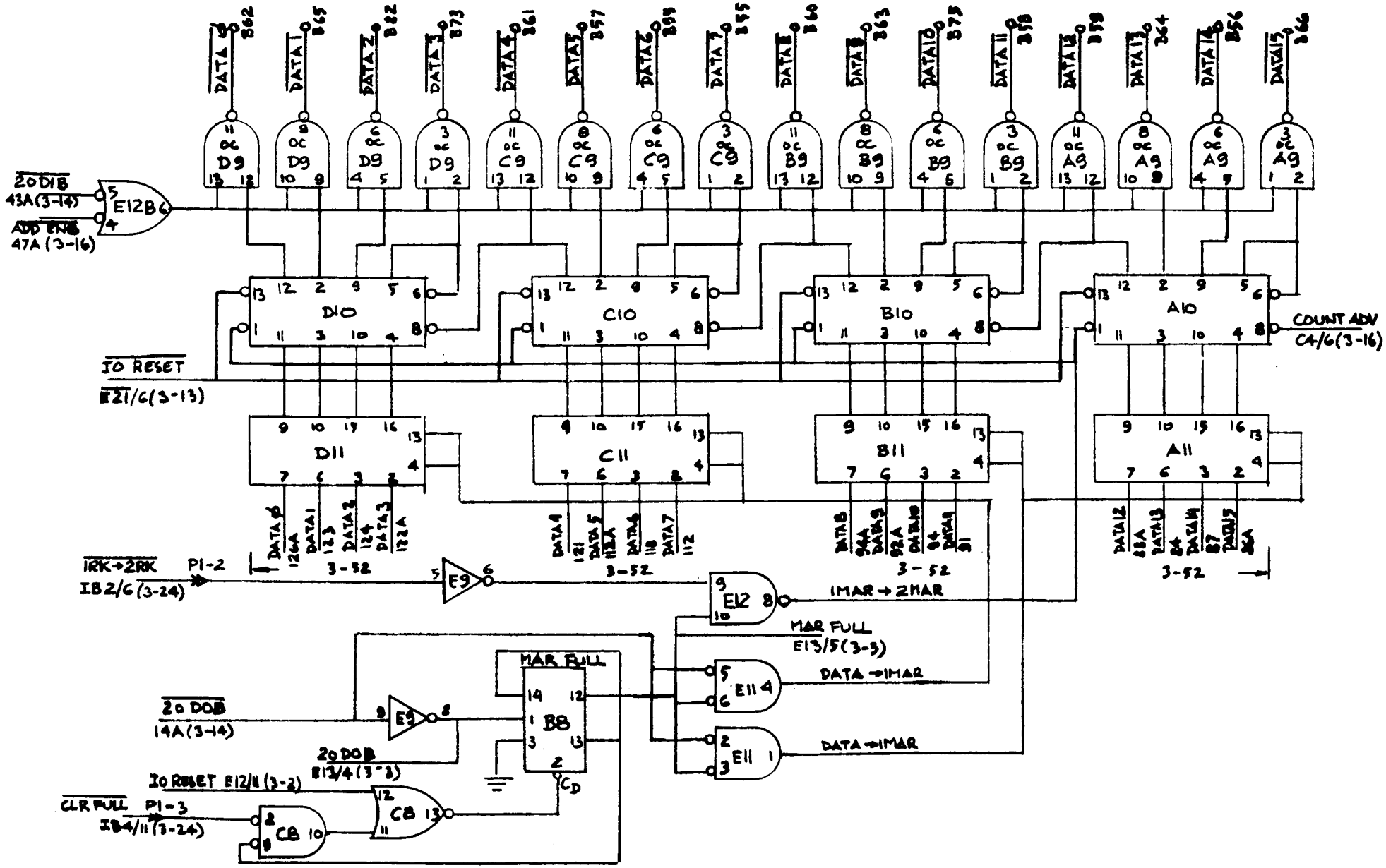


FIG 3-4 MEMORY ADDRESS REGISTER

REV.1

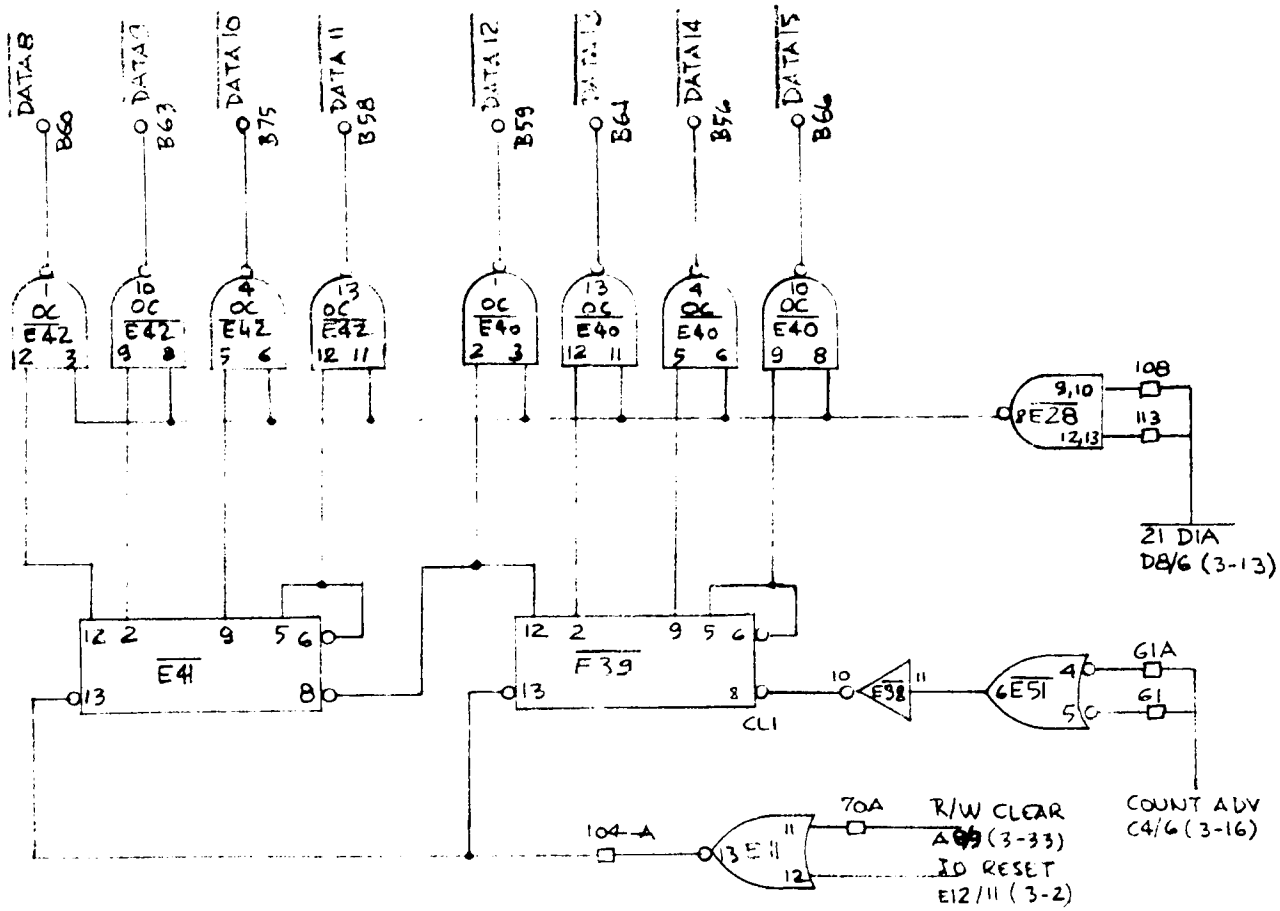


FIG 3-5 WORD COUNT REGISTER

REV. 1

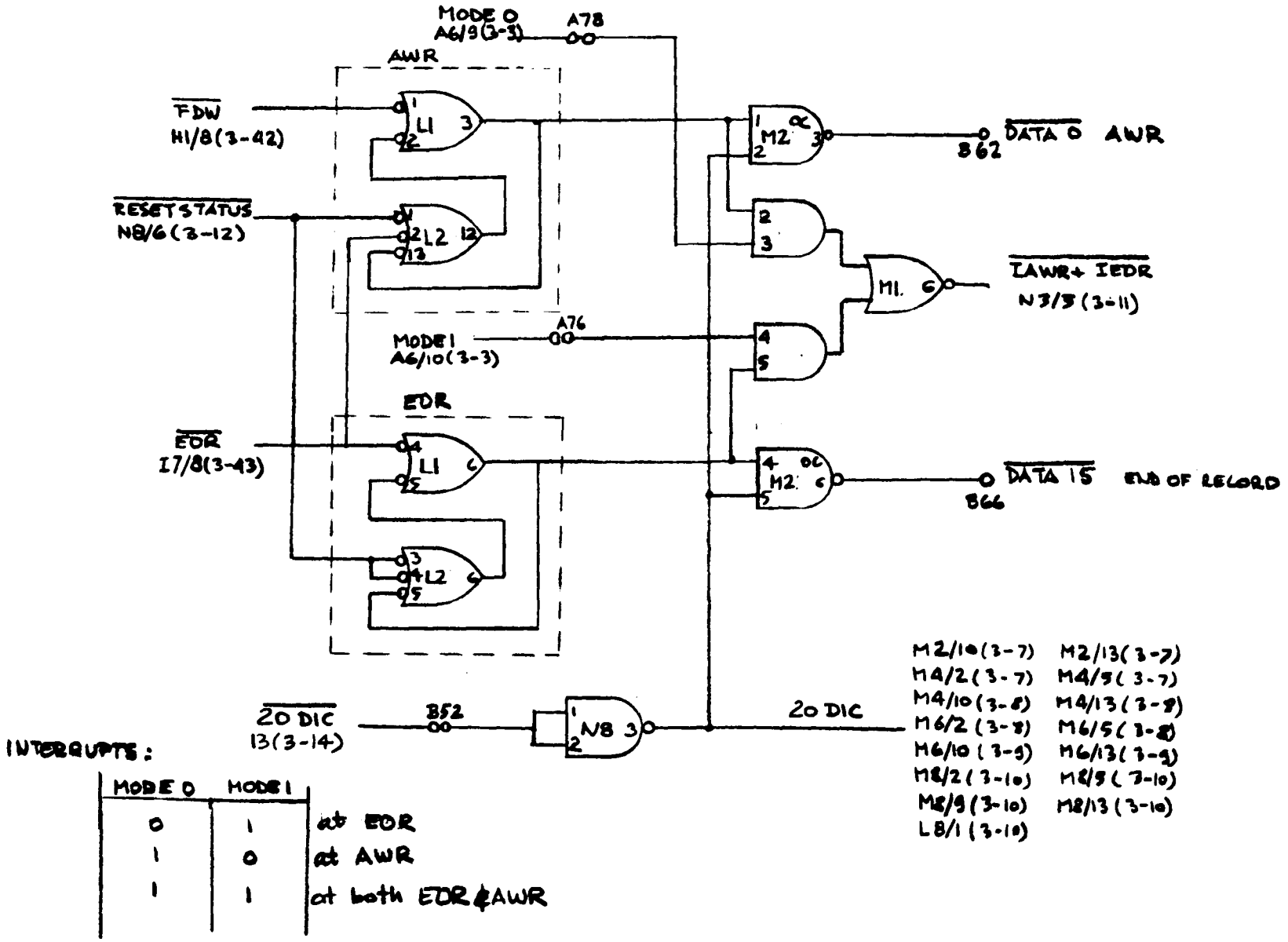


FIG 3-6 STATUS REGISTER (1)

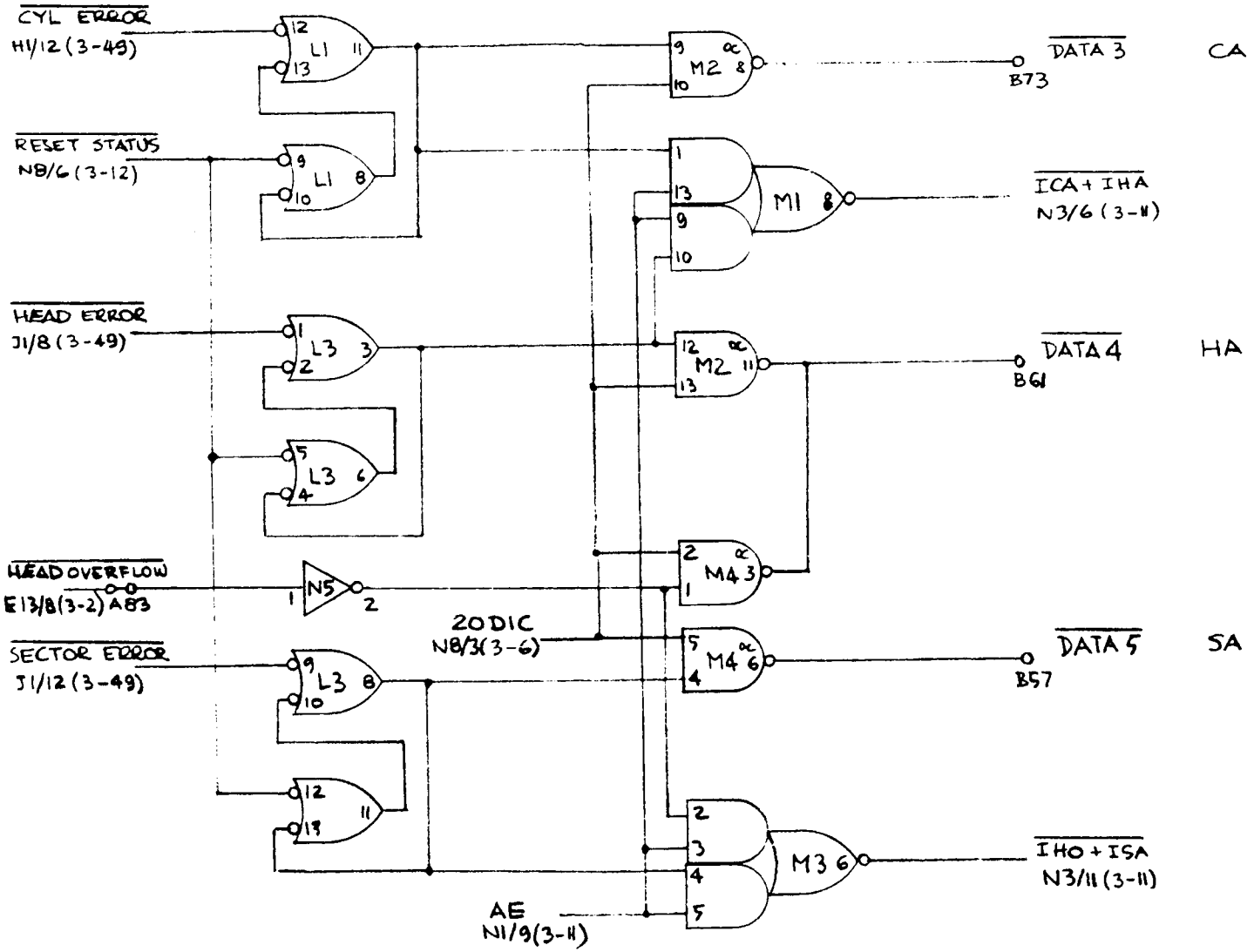


FIG 3-7 STATUS REGISTER (2)

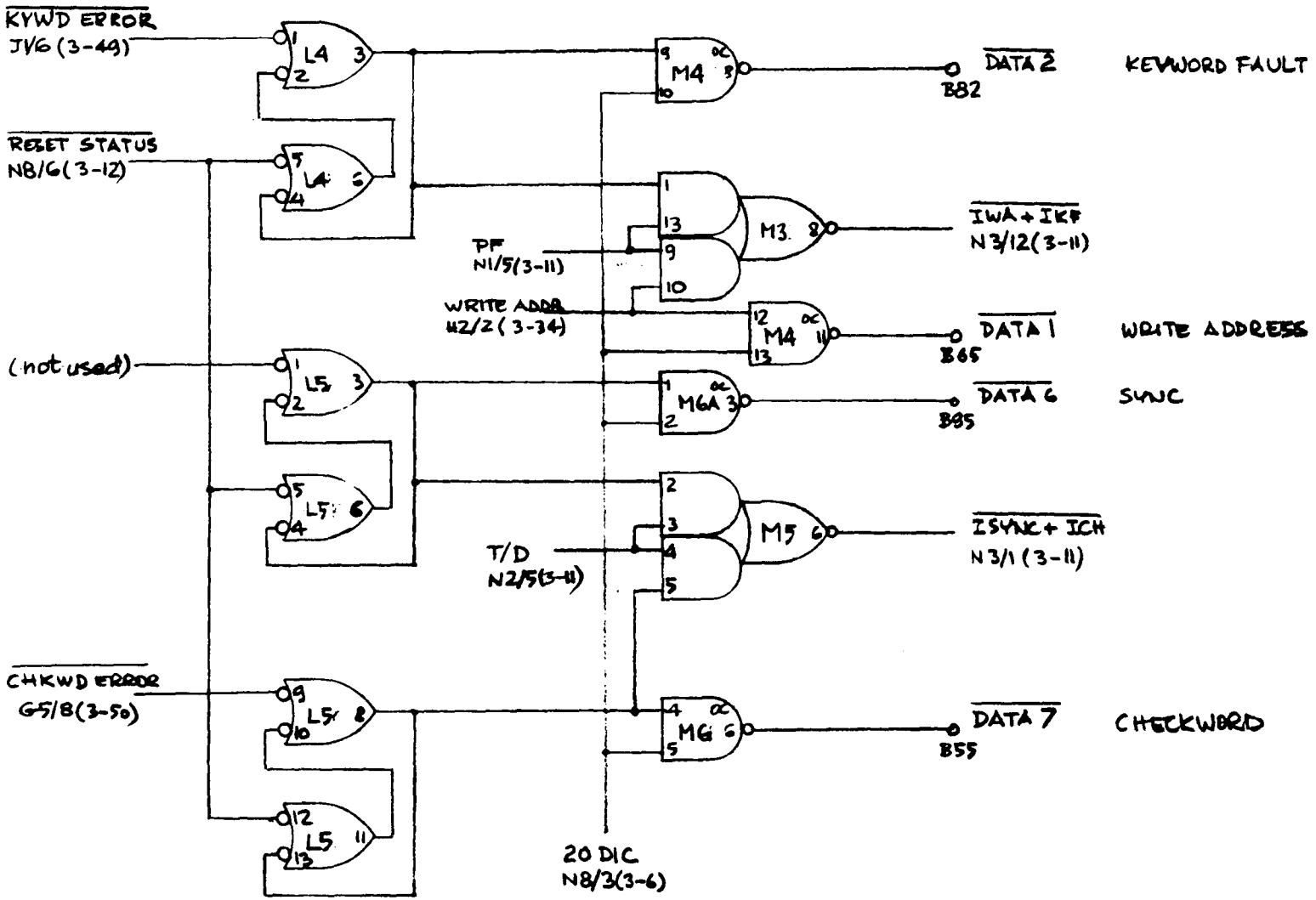
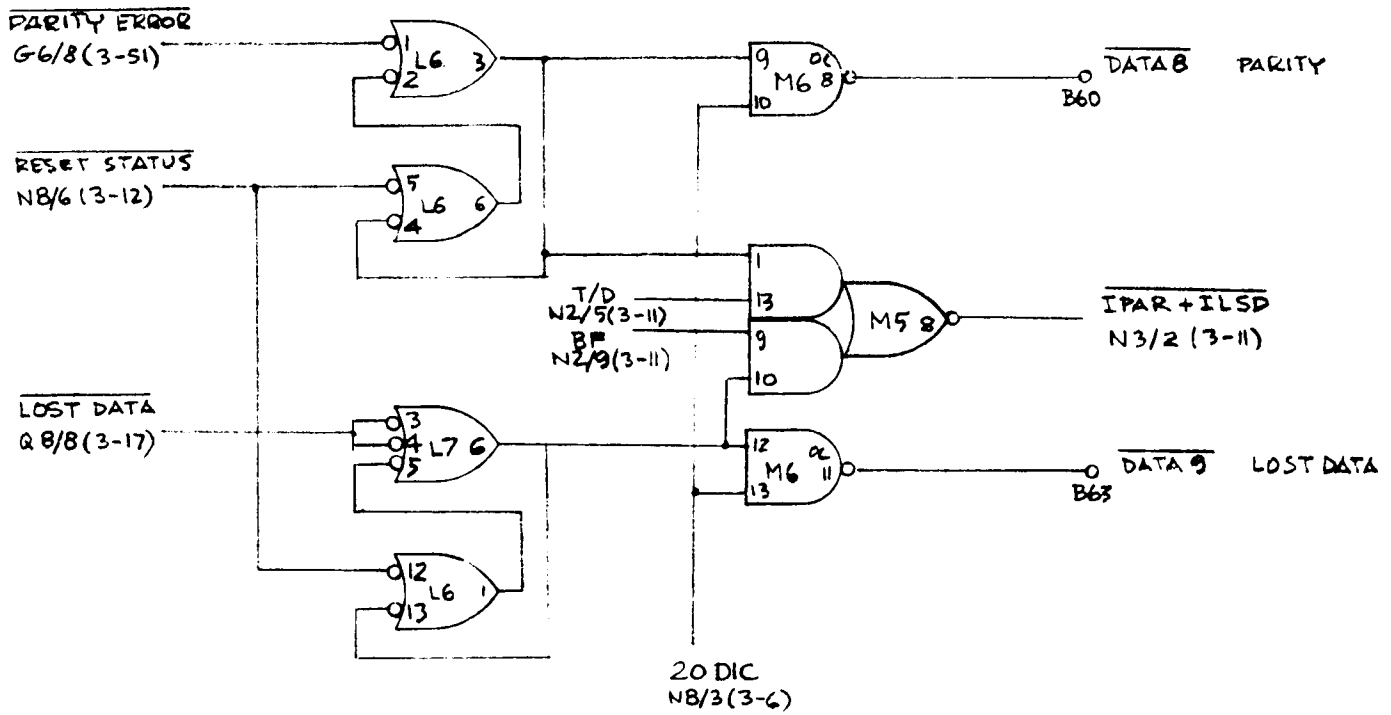


FIG 3-8 STATUS REGISTER (3)



REV.1

FIG 3-9 STATUS REGISTER (4)

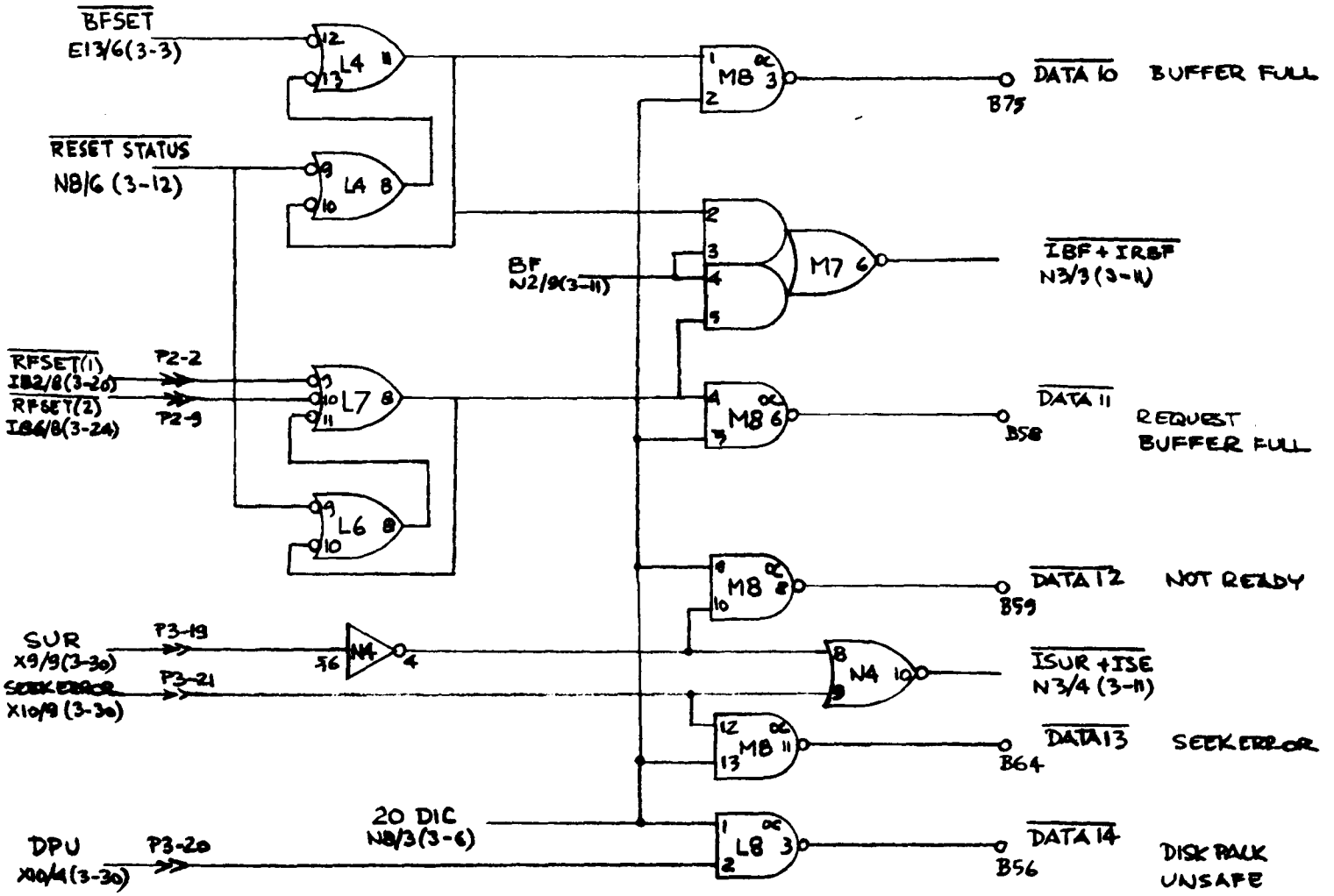
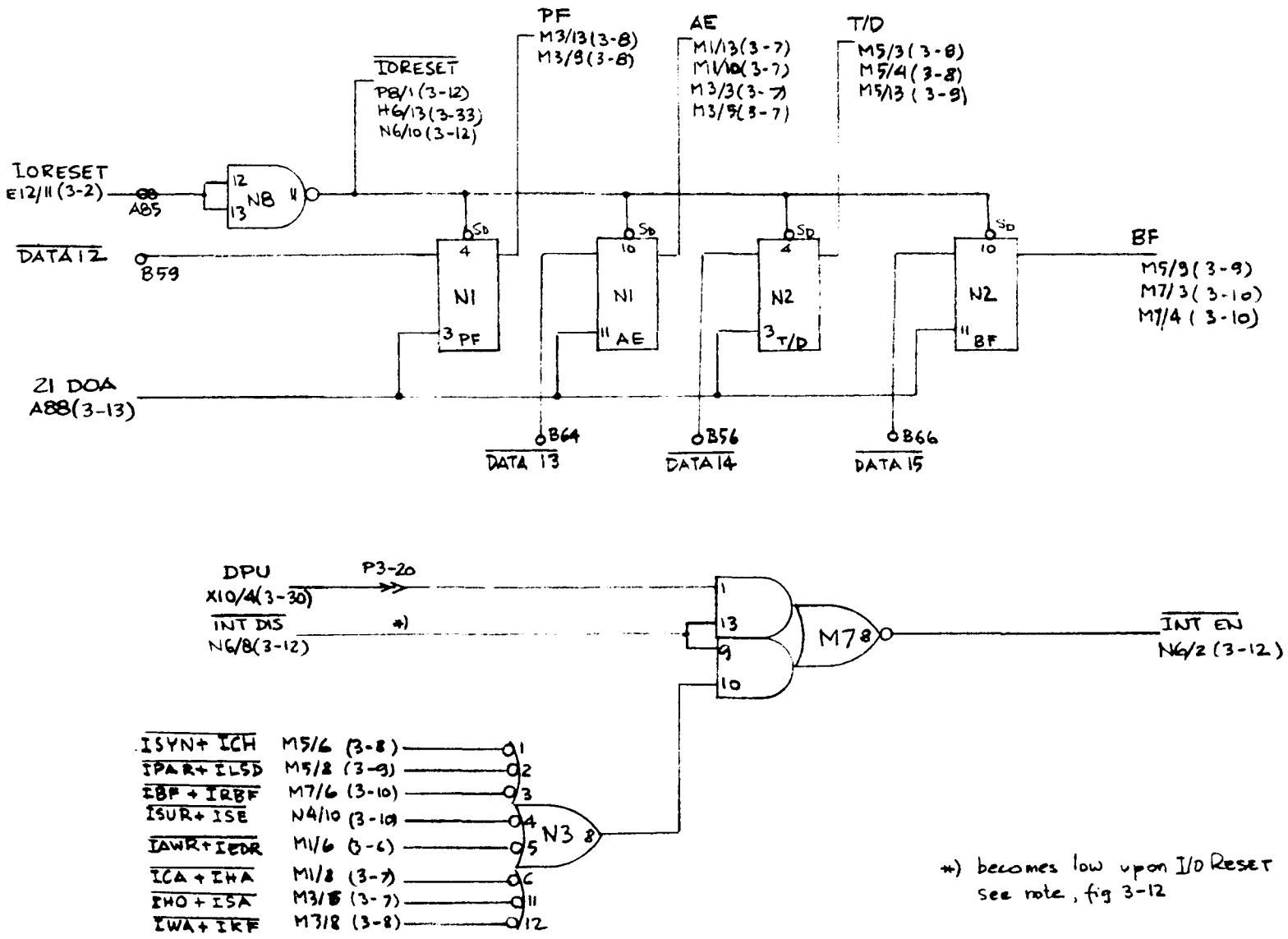


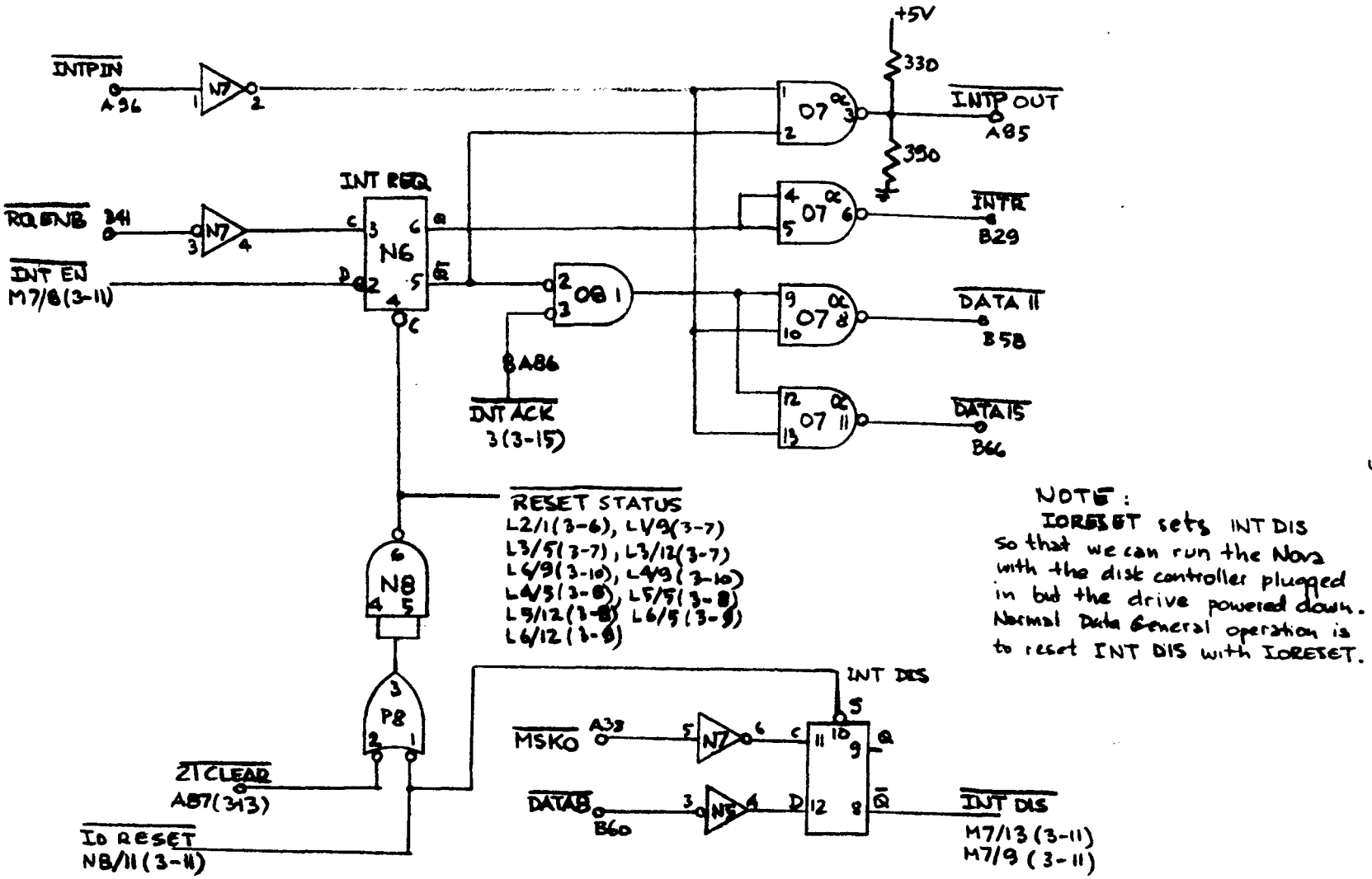
FIG 3-10 STATUS REGISTER (5)

REV.1



REV. 1

FIG 3-11 DISABLE INTERRUPT REGISTER



NOTE:
IORESET sets INT DIS so that we can run the Nova with the disk controller plugged in but the drive powered down. Normal Data General operation is to reset INT DIS with IORESET.

FIG 3-12 INTERRUPT LOGIC (DEV 21)

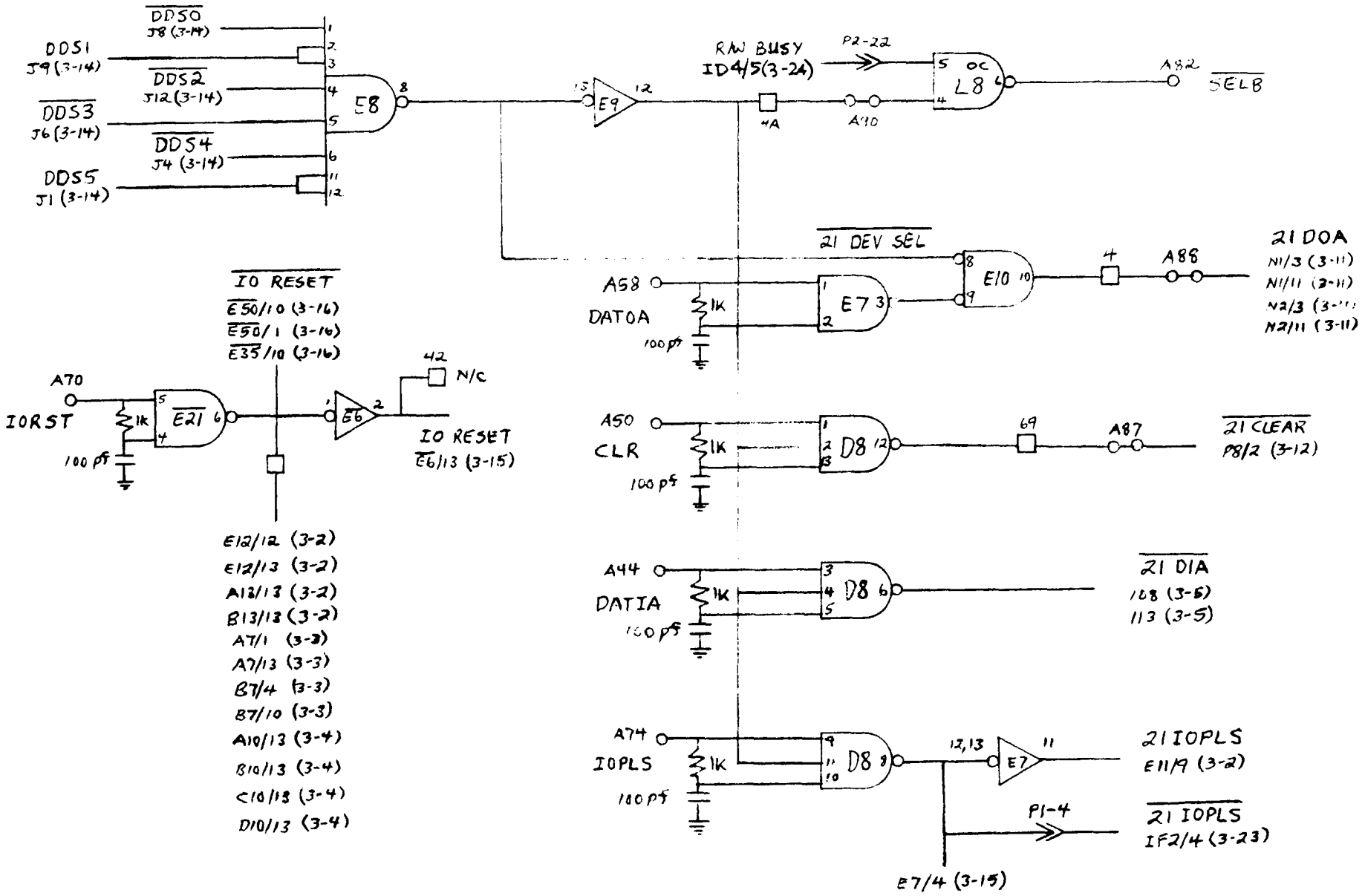


FIG 3-13 IOT INSTRUCTION DECODE (DEV 21)

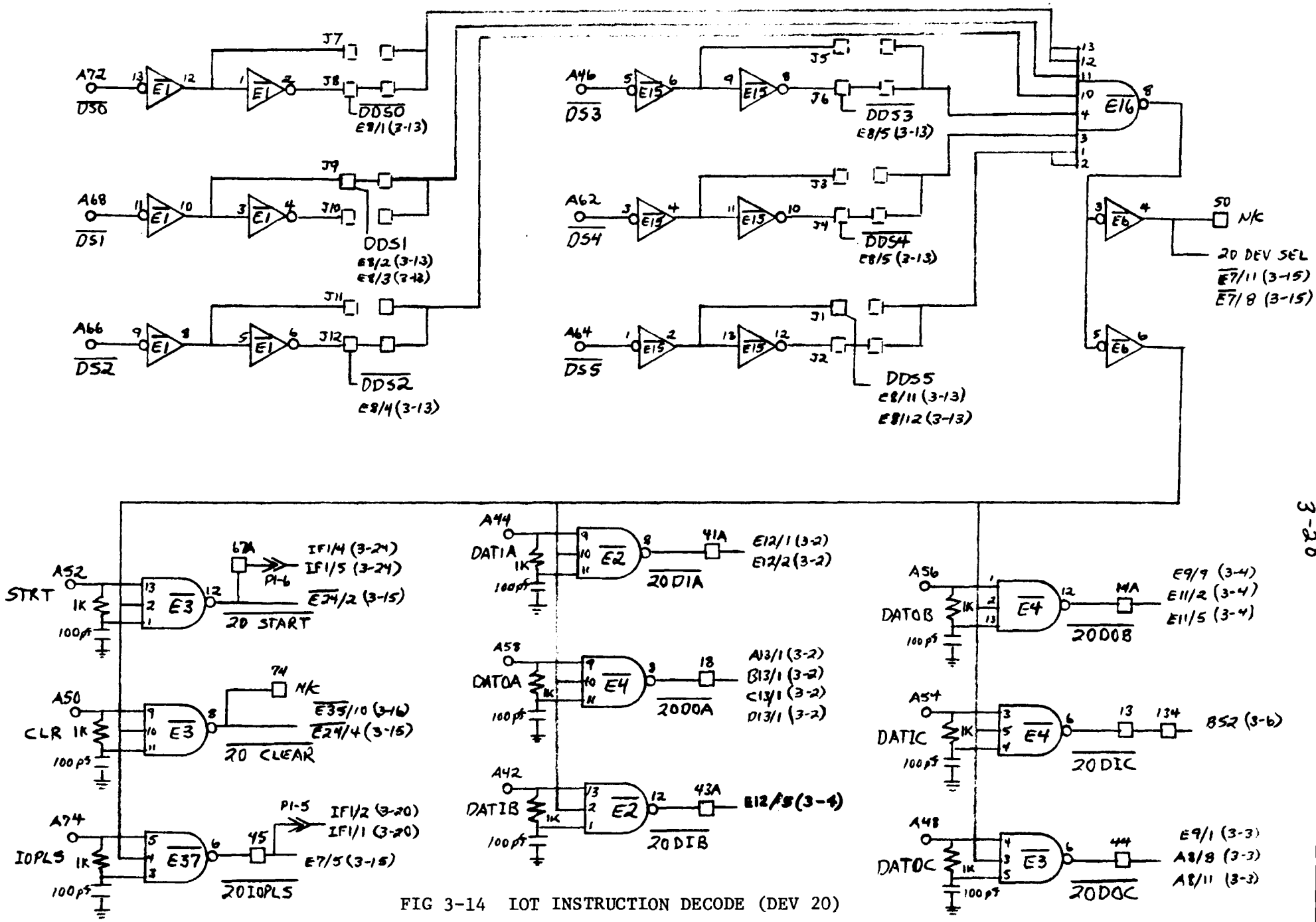
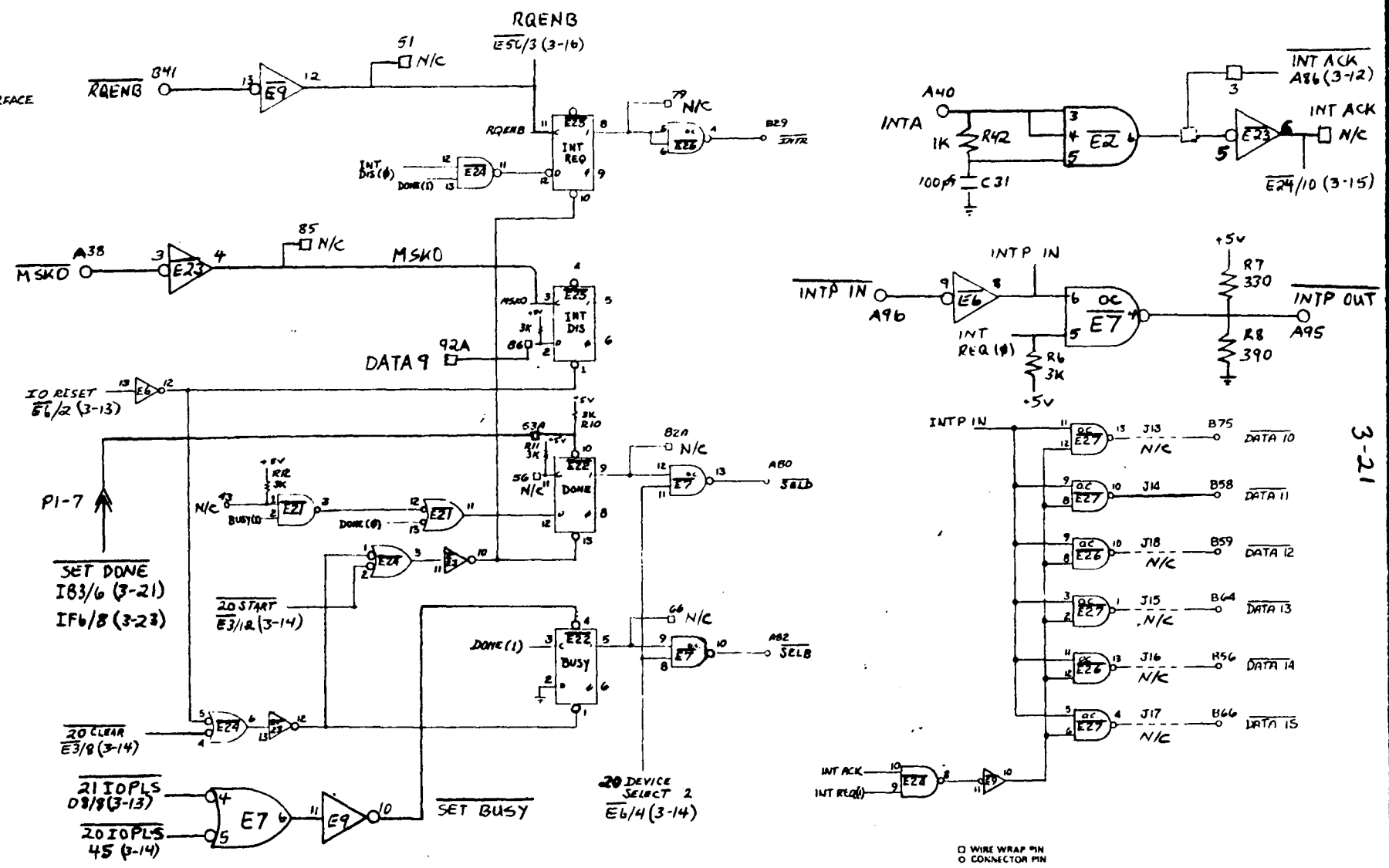


FIG 3-14 IOT INSTRUCTION DECODE (DEV 20)

REV. 2

INS AVAILABLE FOR INTERFACE

A47	10
A49	9
A57	8A
A59	8
A61	34A
A63	29
A65	7
A67	41
A69	6A
A71	48
A73	47
A75	49A
A76	1
A77	6
A78	2
A79	57
A81	67
A83	68A
A84	2A
A85	68
A86	3
A87	69
A88	4
A89	70A
A90	4A
A91	70
A92	5
B6	71
B11	72A
B13	12
B15	76
B19	83
B23	84A
B25	89
B27	93
B31	82
B34	131
B36	132A
B38	132
B40	133
B49	134A
B49	98A
B51	104
B54	134
B65	135
P.67	136A
P.67	1E5
P.67	136



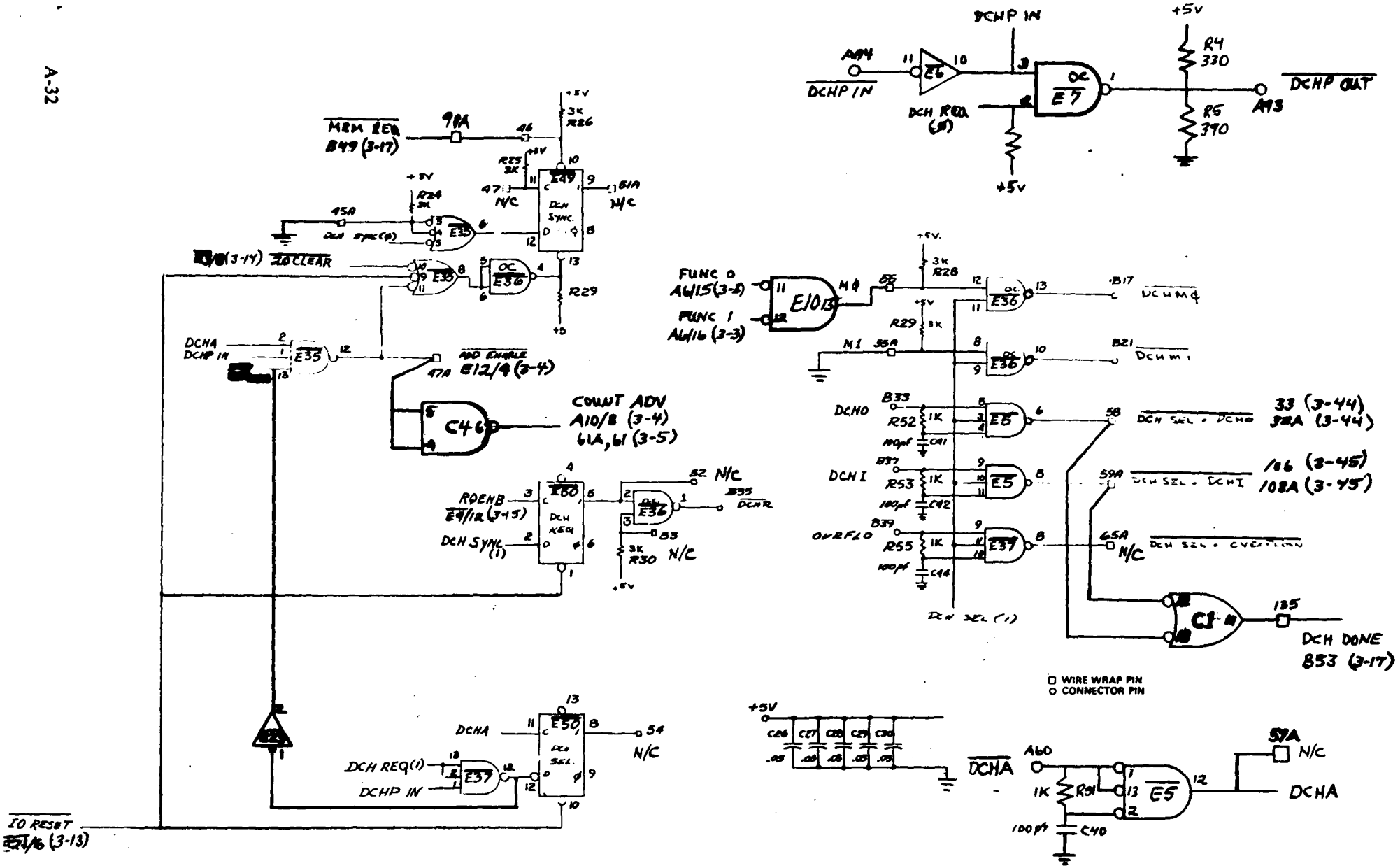
GENERAL PURPOSE INTERFACE: BUSY, DONE, INTERRUPT (4040)

3-21

3-15

FIG. 3-15 BUSY, DONE, INTR. LOGIC (DEV. CODE 20)

REV 2



GENERAL PURPOSE INTERFACE: DATA CHANNEL CONTROL (4042)

FIG. 3-16 DATA CHANNEL CONTROL LOGIC

3-16
REV. 2

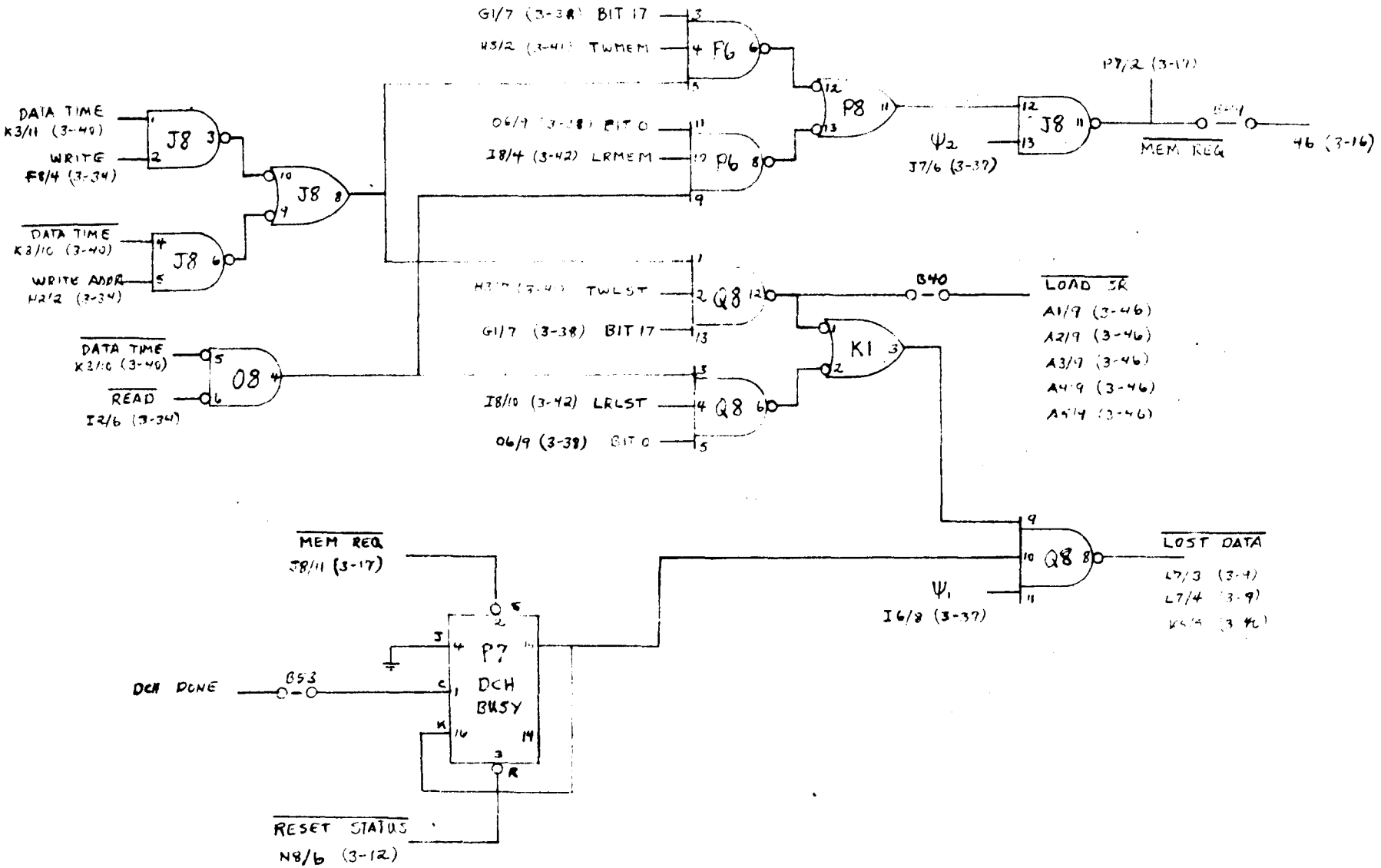
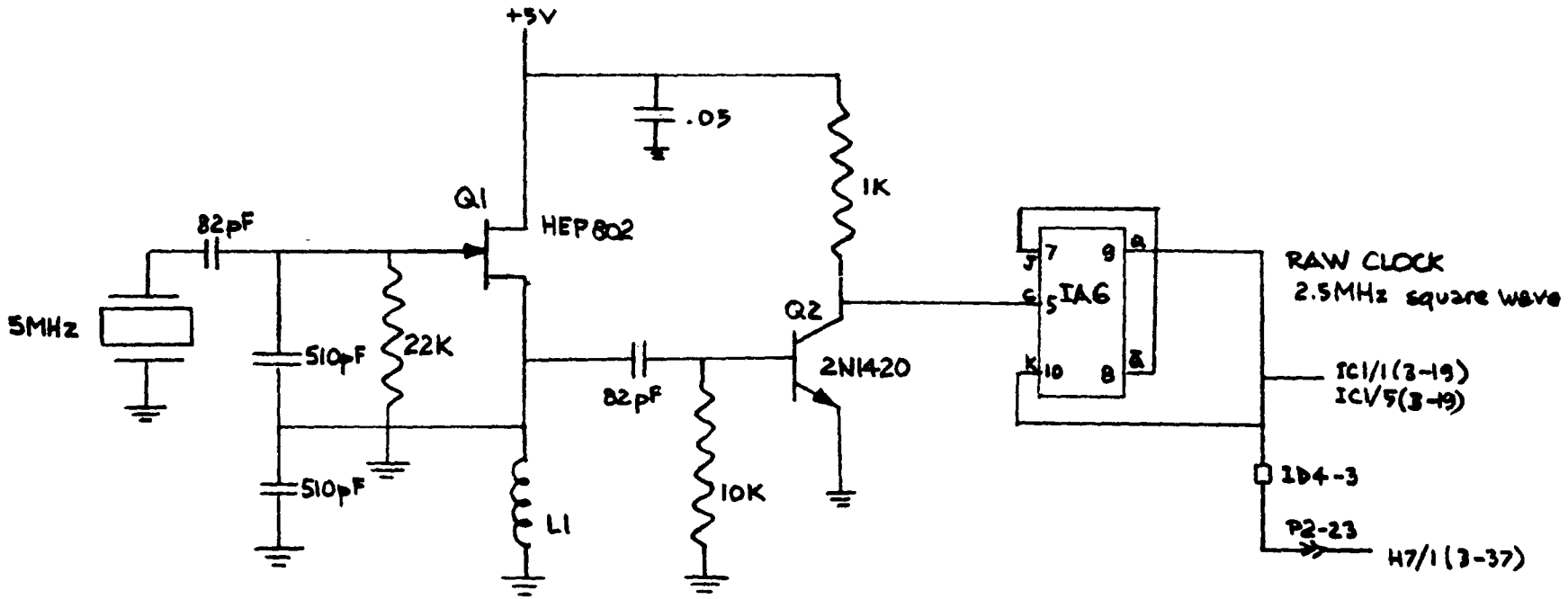


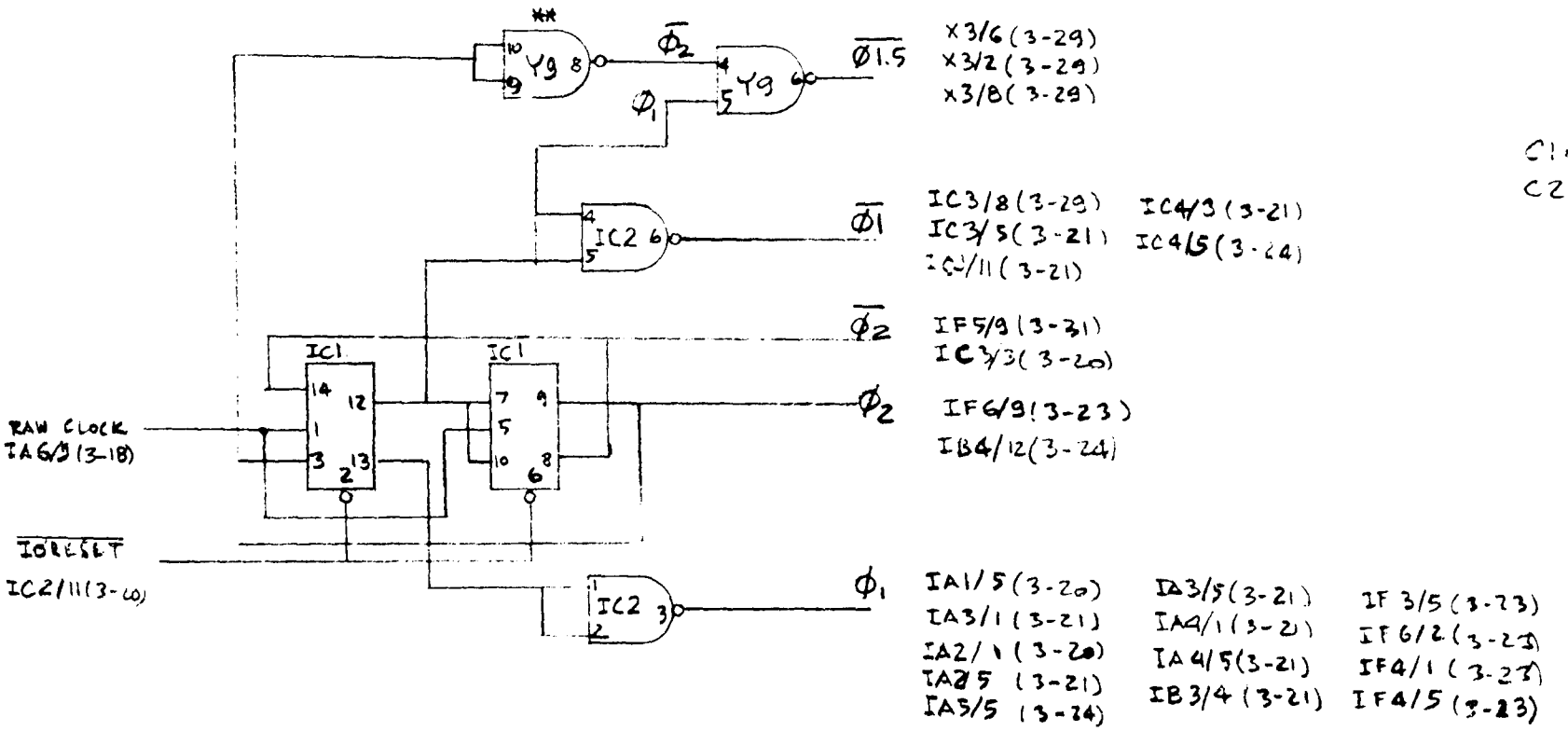
FIG 3-17 MEMORY REQUEST LOGIC

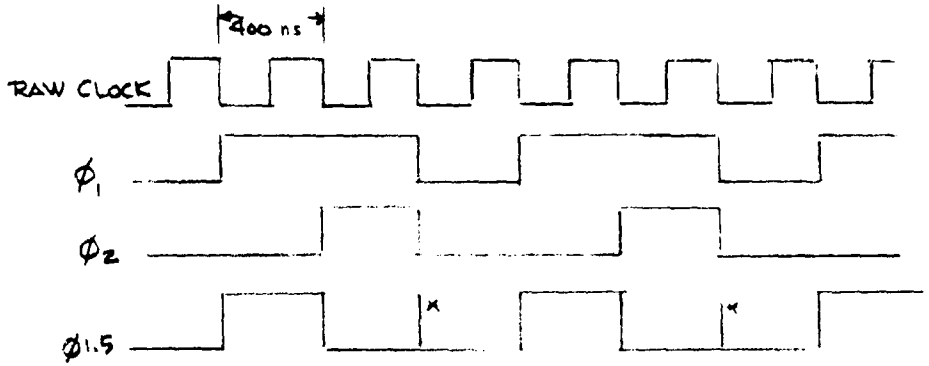
REV 1



REV.1

FIG 3-18 SYSTEM CLOCK





* spikes appear here if Φ2 is not delayed

** gate added to delay Φ2 to remove spikes

FIG 3-19 SEQUENCER CLOCK

REV 1

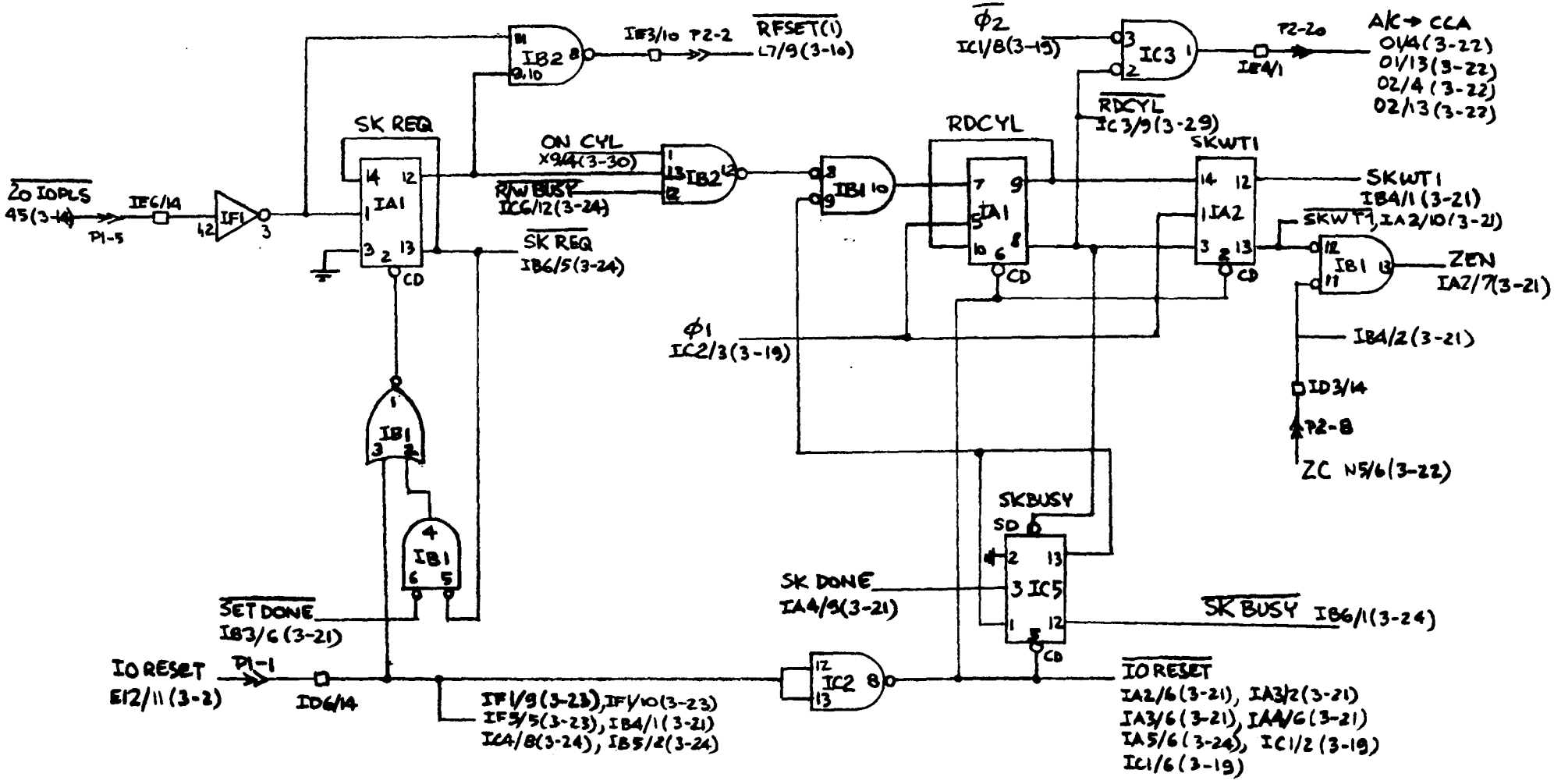


FIG 3-20 SEEK SEQUENCER (1)

REV.1

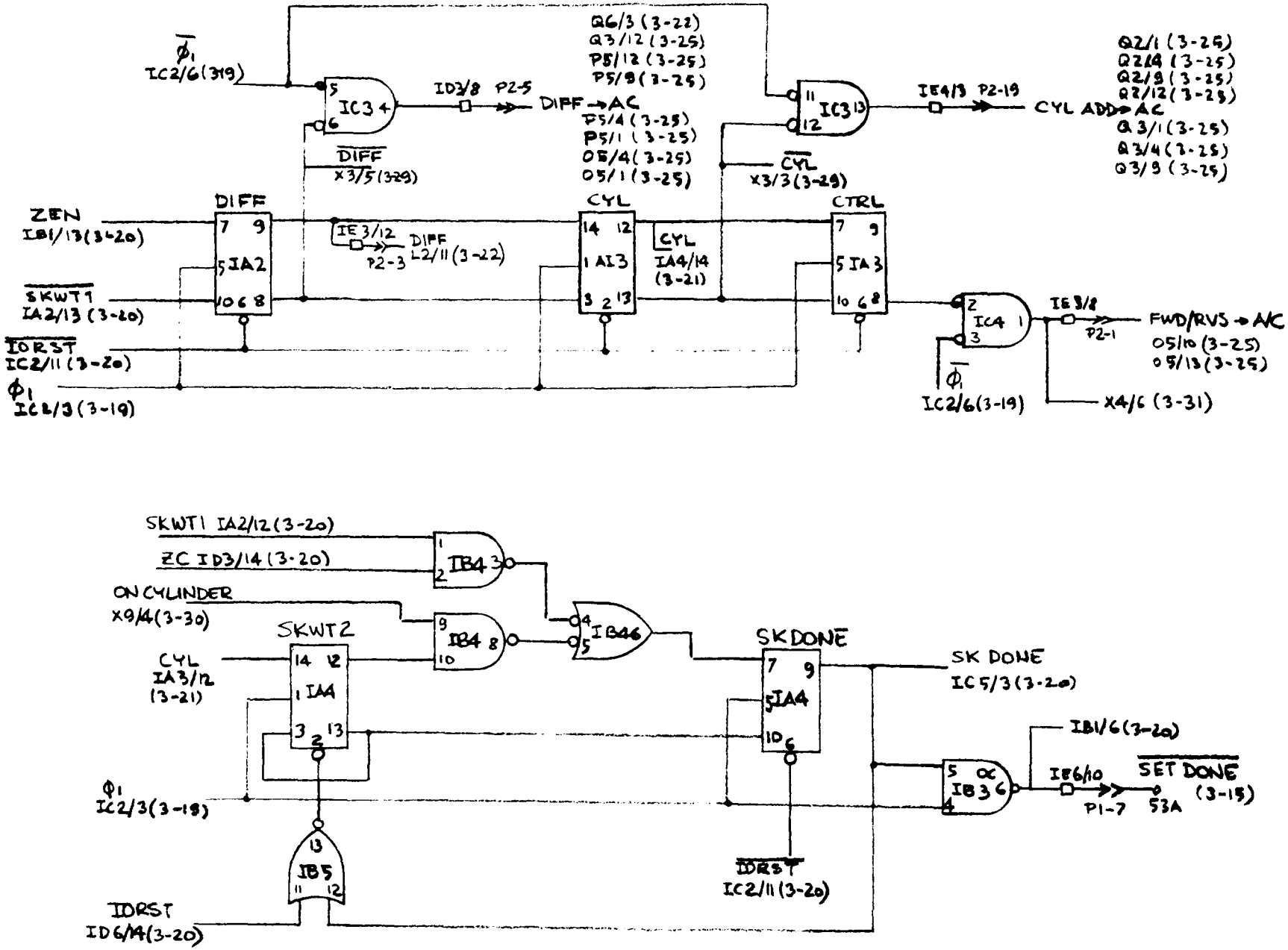


FIG 3-21 SEEK SEQUENCER (2)

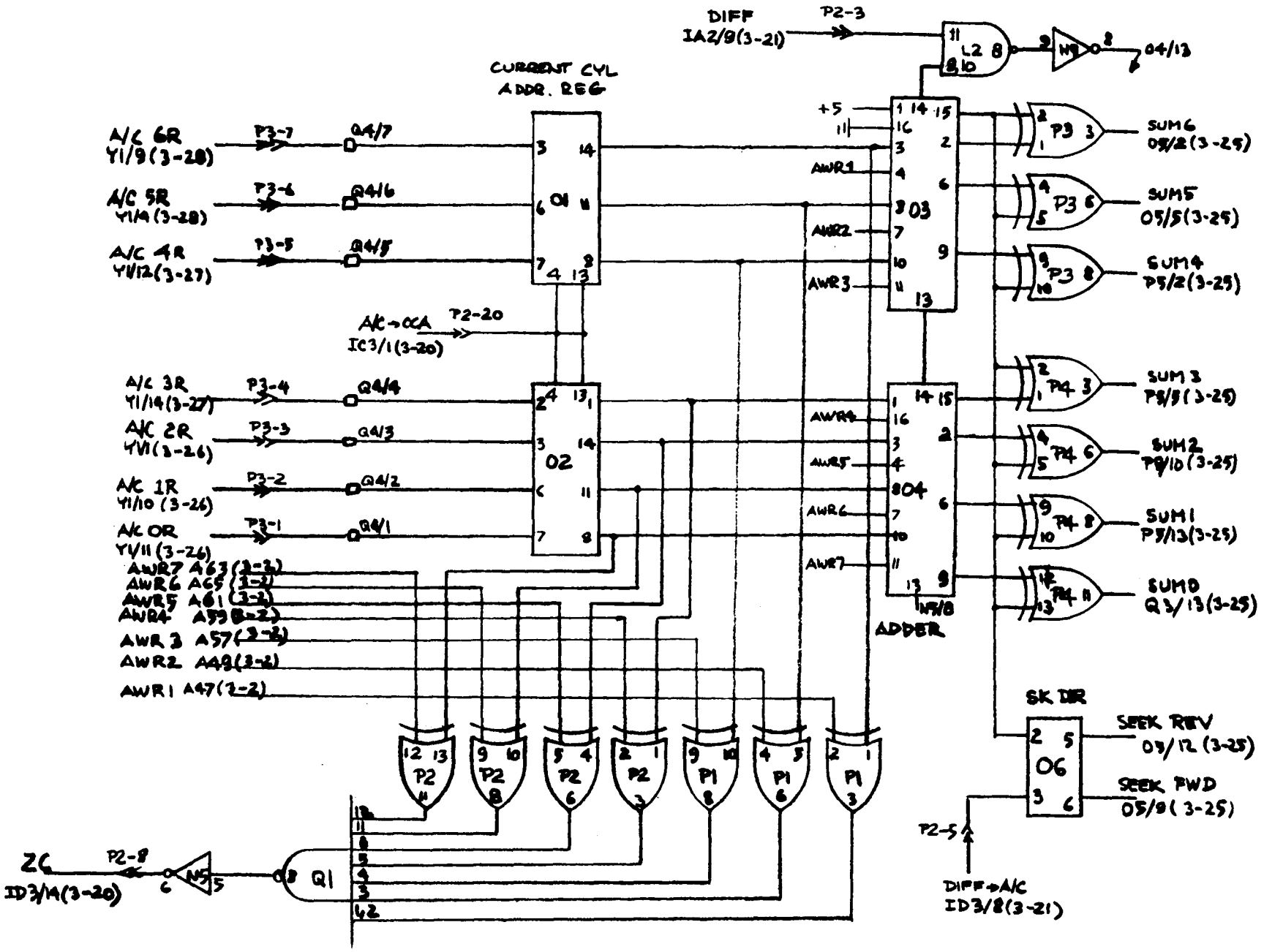


FIG 3-22 CYLINDER DIFFERENCE LOGIC

REV.1

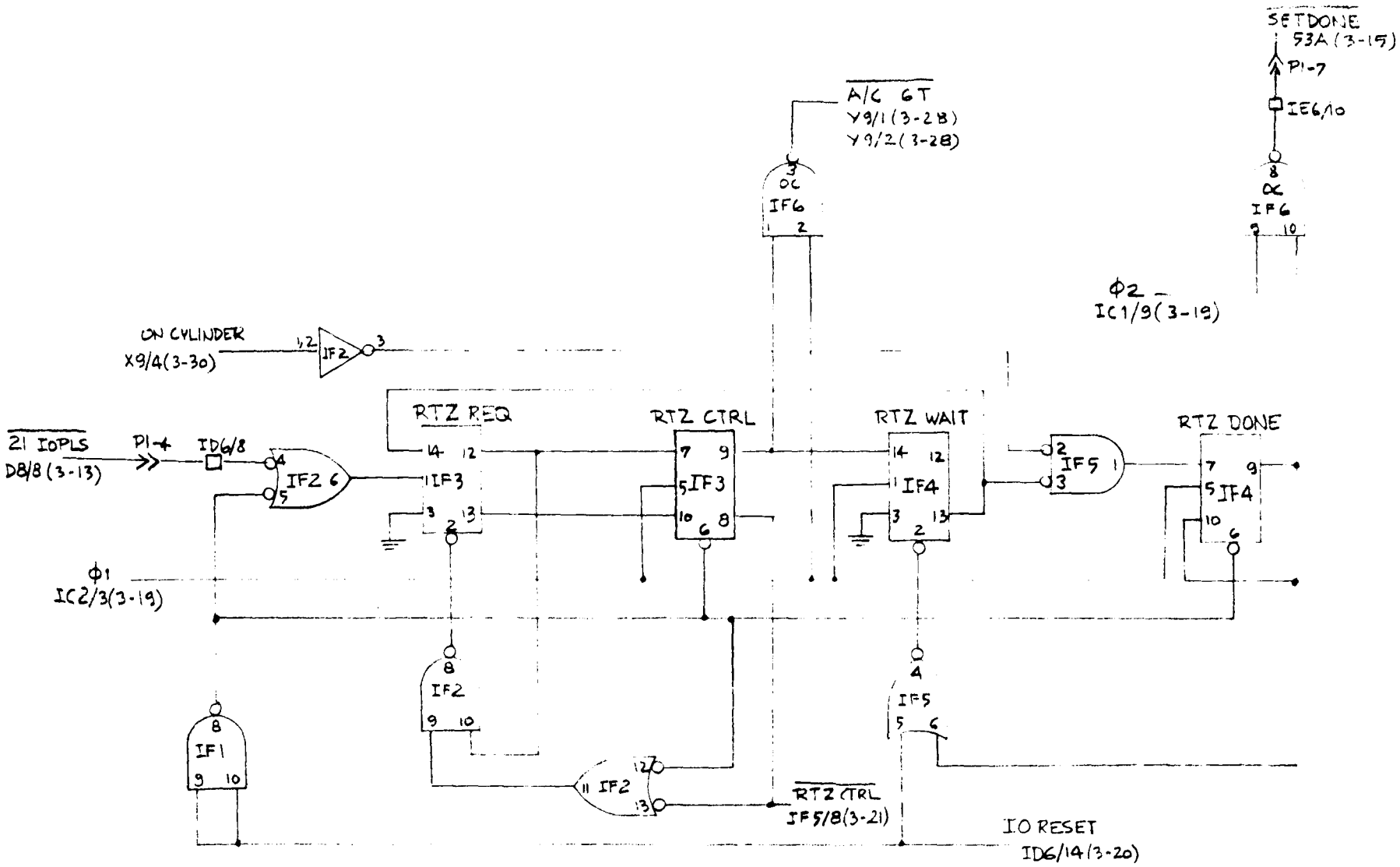


FIG 3-23 RTZS SEQUENCER

REV 1

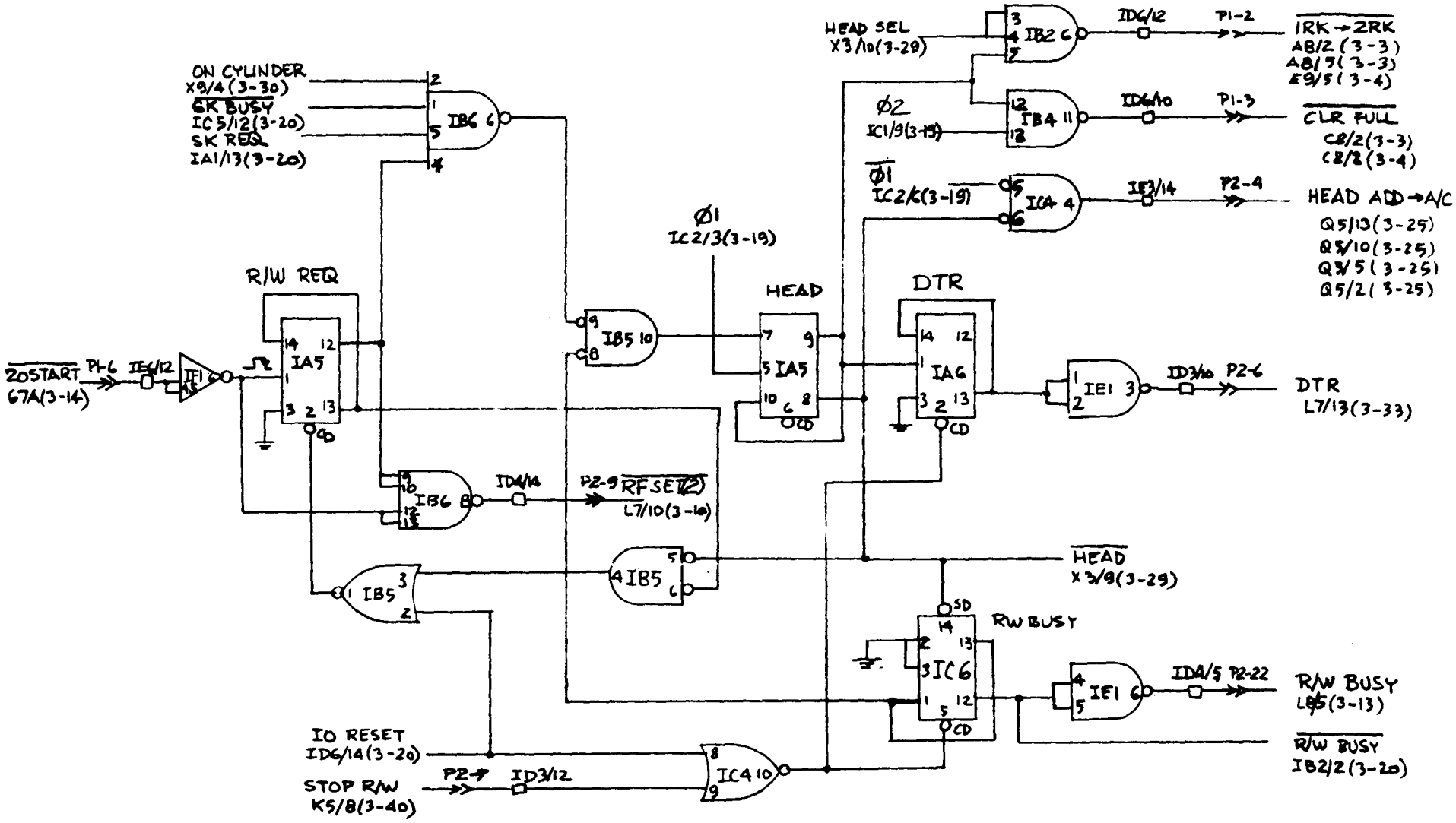


FIG 3-24 DTR SEQUENCER

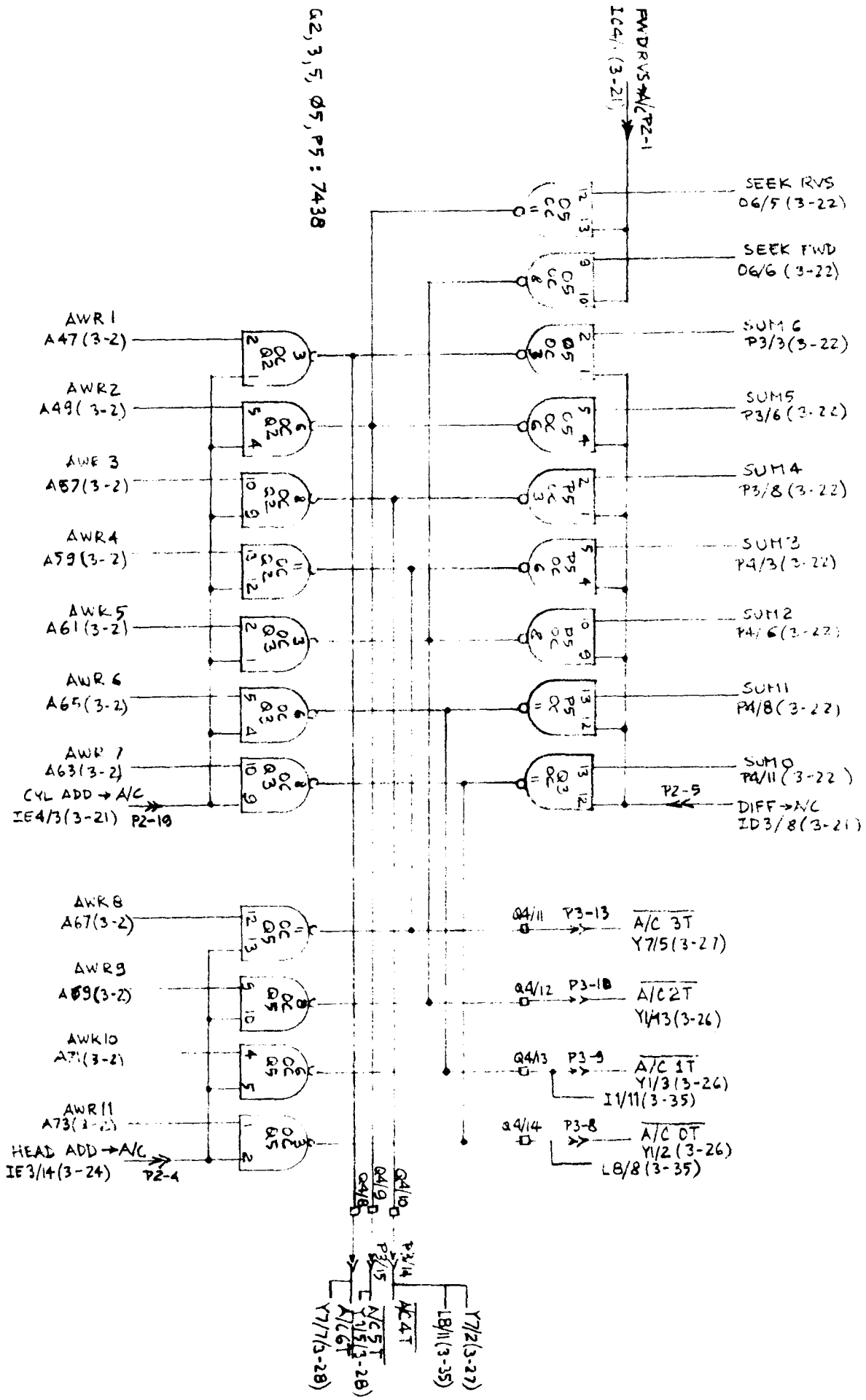


FIG 3-25 A/CT BUS ENABLES

REV. 1

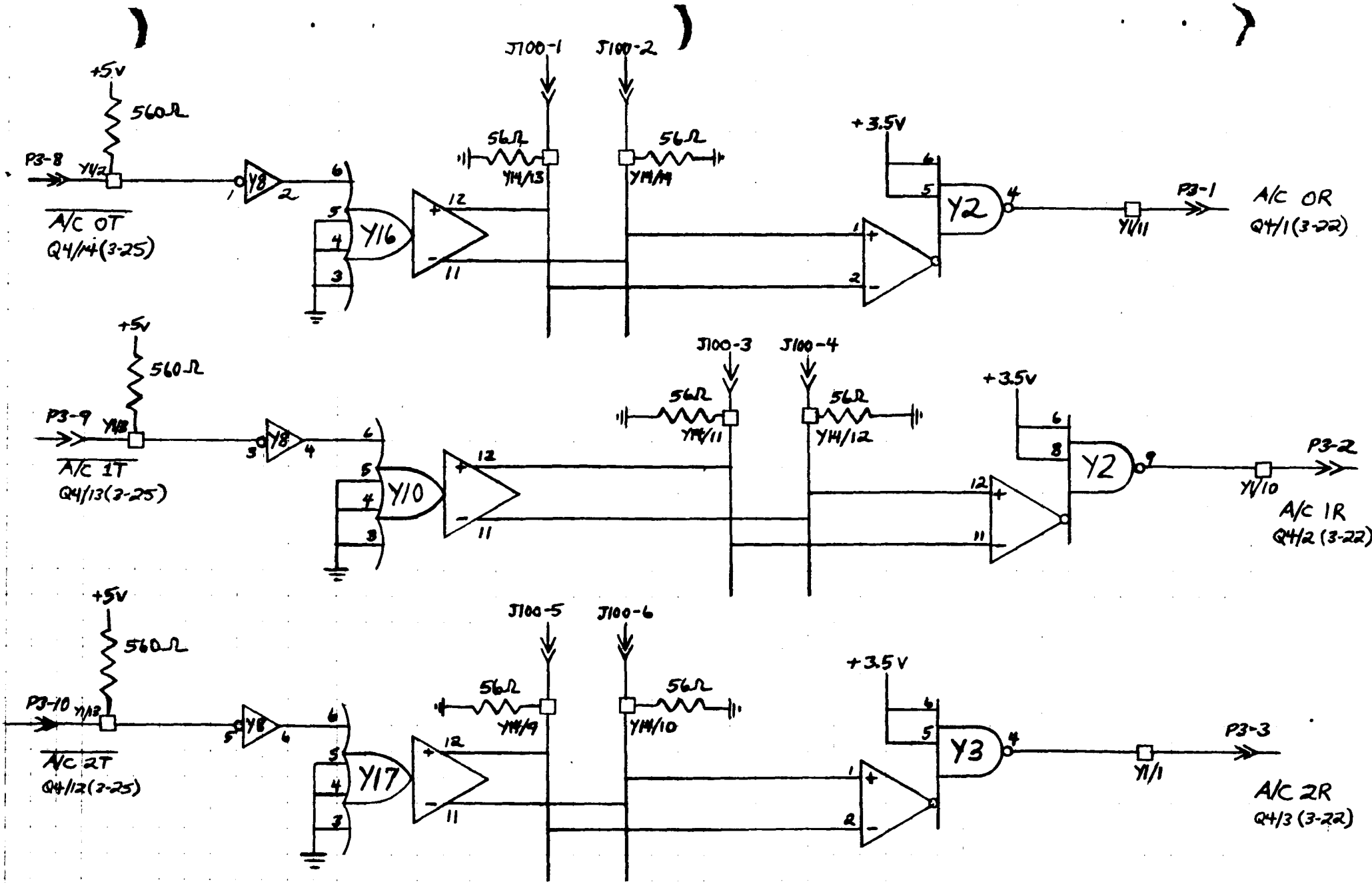


FIG. 3-26 A/C BUS TRANSMITTERS RECEIVERS, page 1 of 3

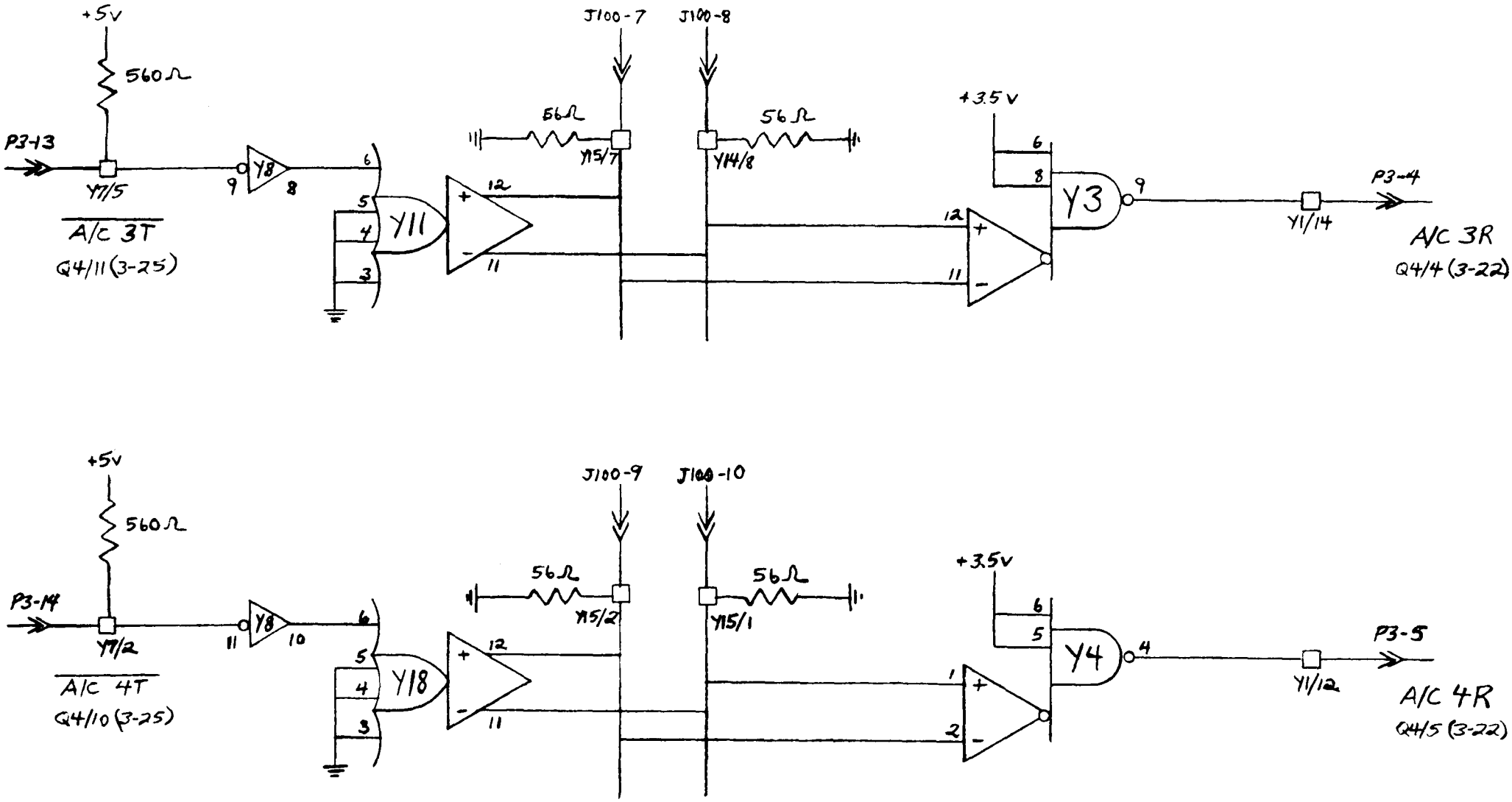


FIG. 3-27 A/C BUS TRANSMITTERS, RECEIVERS, page 2 of 3

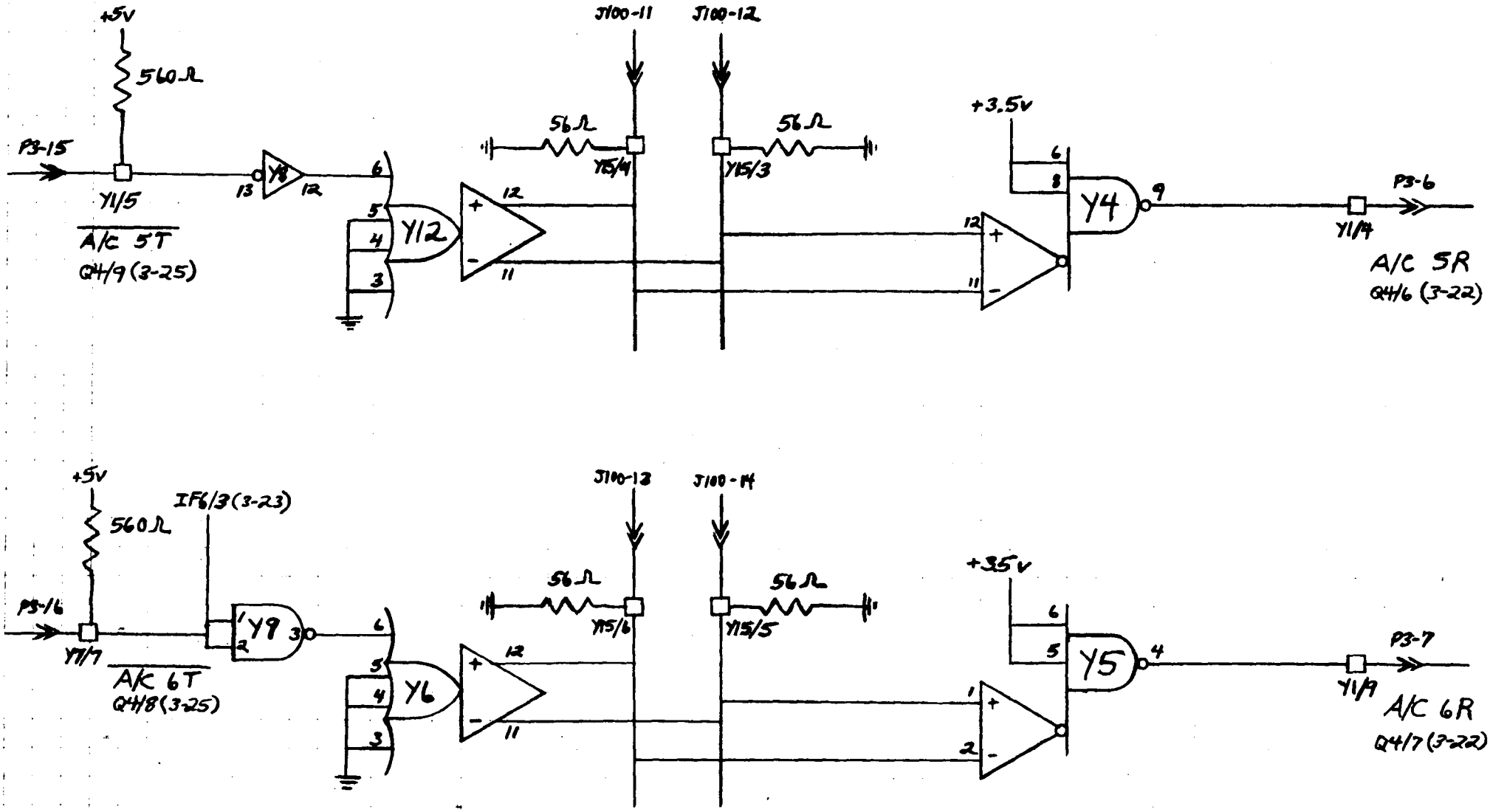
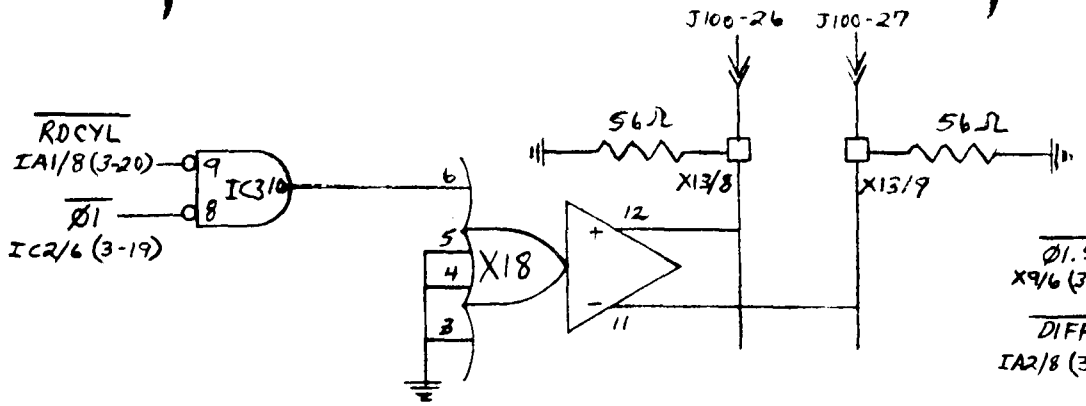
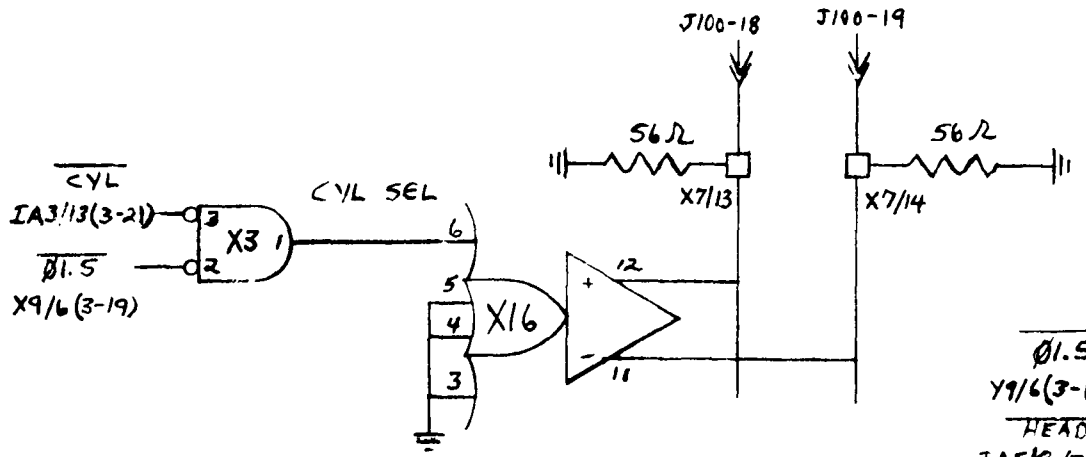


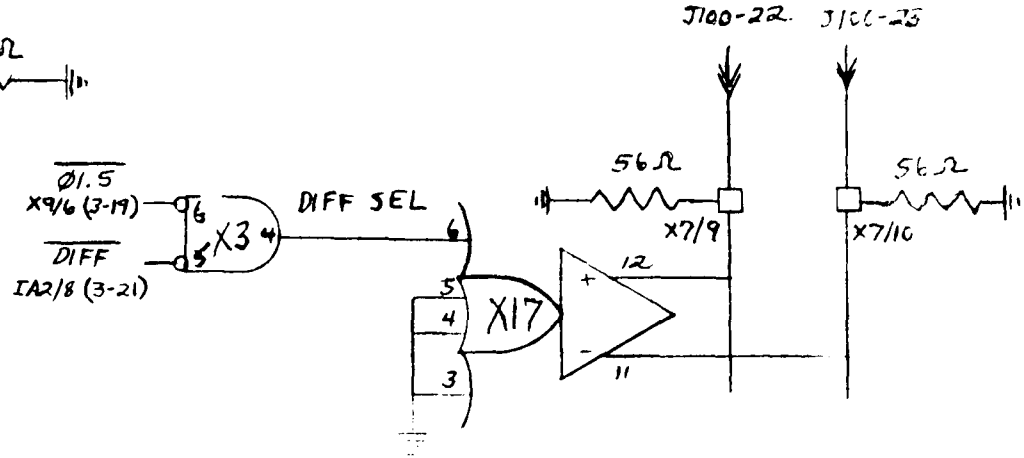
FIG 3-28 A/C BUS TRANSMITTERS, RECEIVERS, page 3 of 3



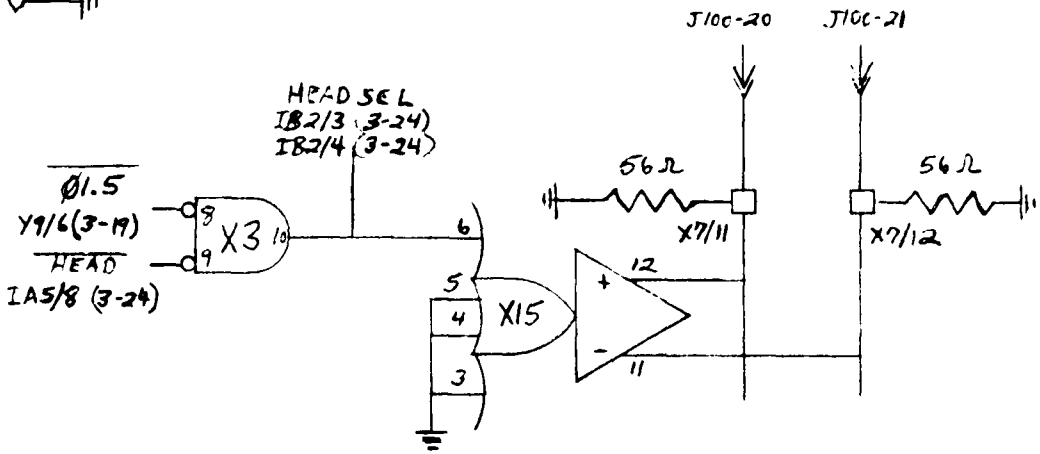
READ CYLINDER SELECT



CYLINDER SELECT



DIFFERENCE SELECT



HEAD SELECT

FIG 3-29 CONTROL, TRANSMITTERS/RECEIVERS, page 1 of 3

REV 1

D

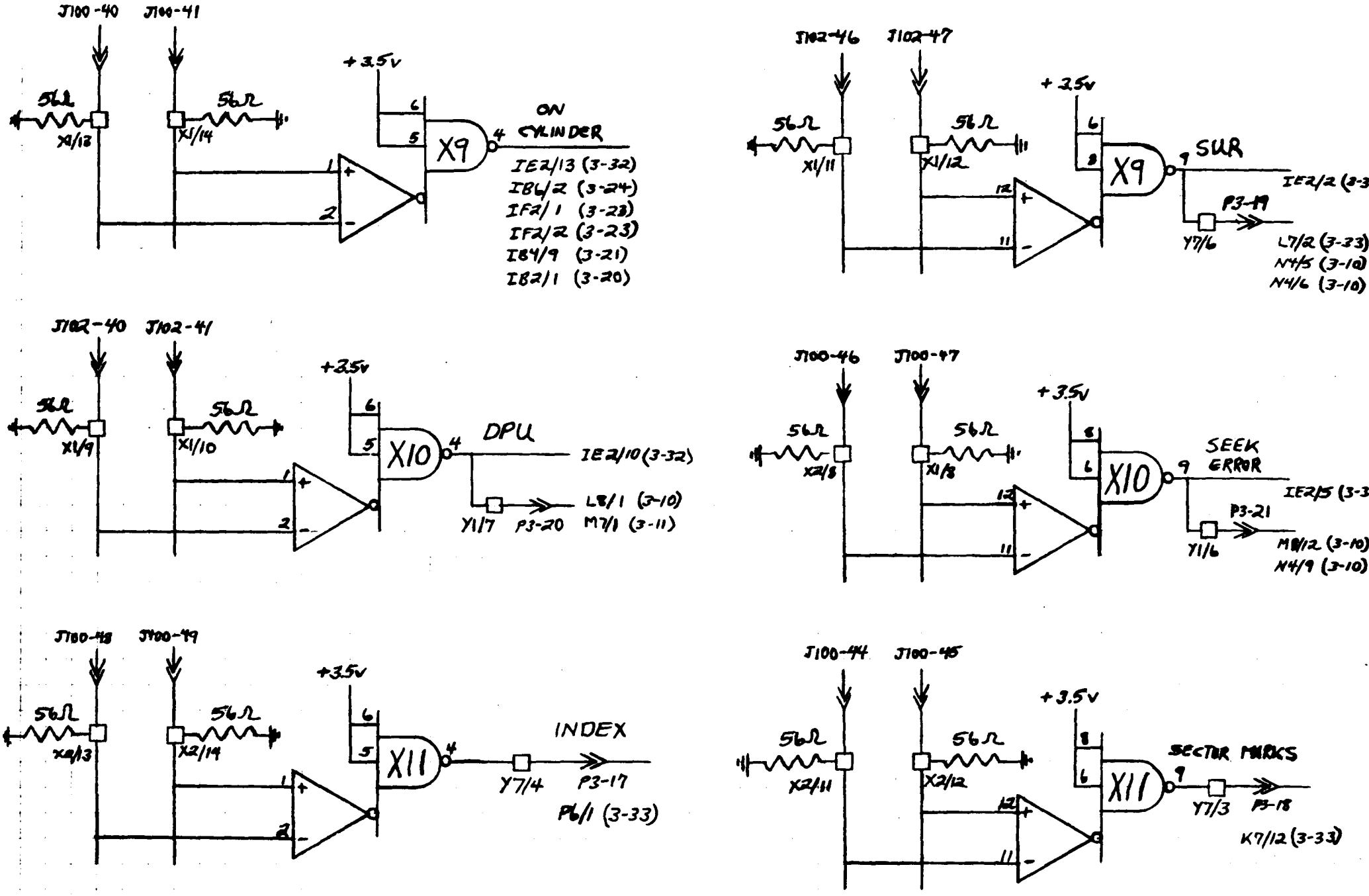


FIG. 3-30 CONTROL, TRANSMITTERS/RECEIVERS, page 2 of 3

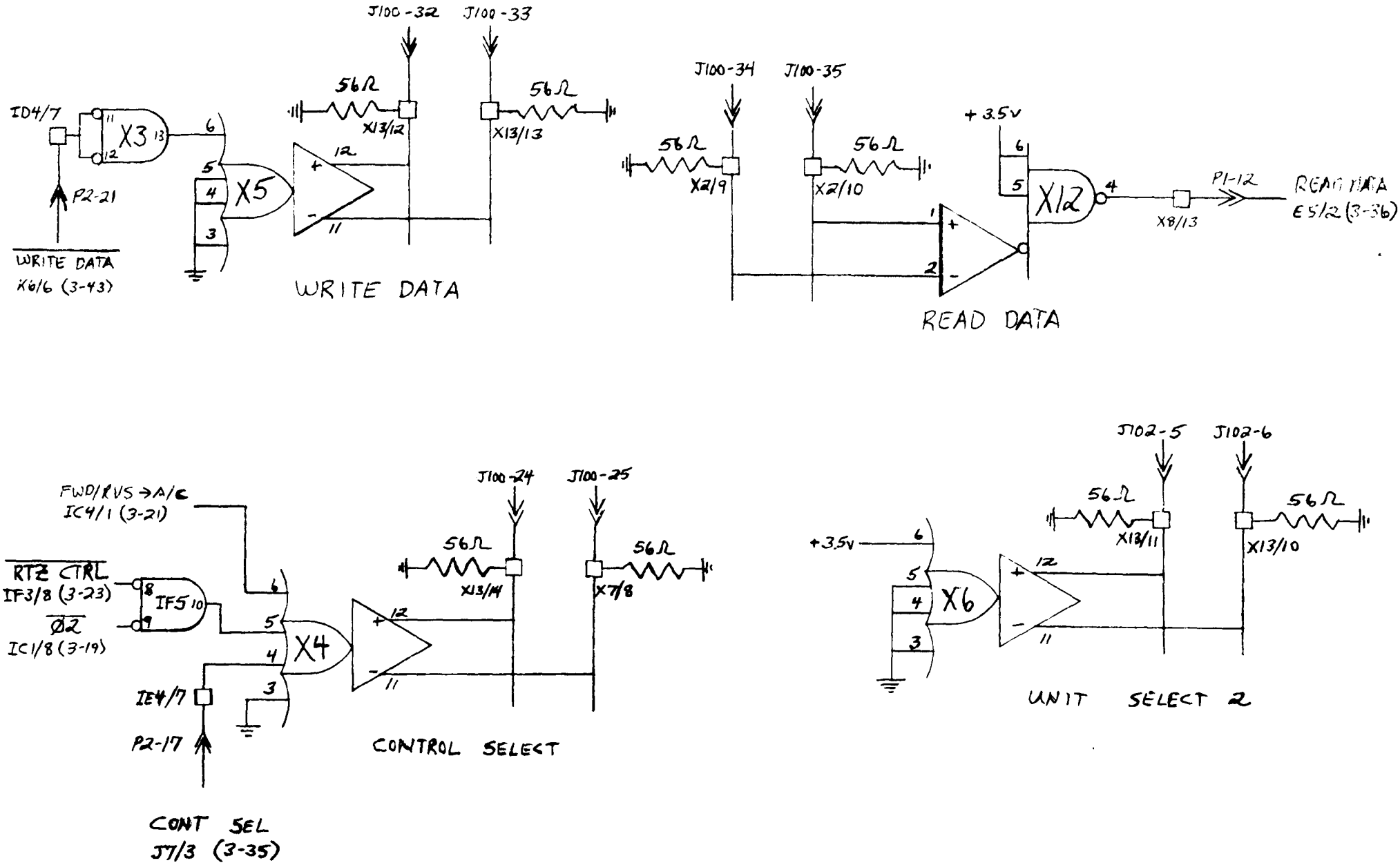
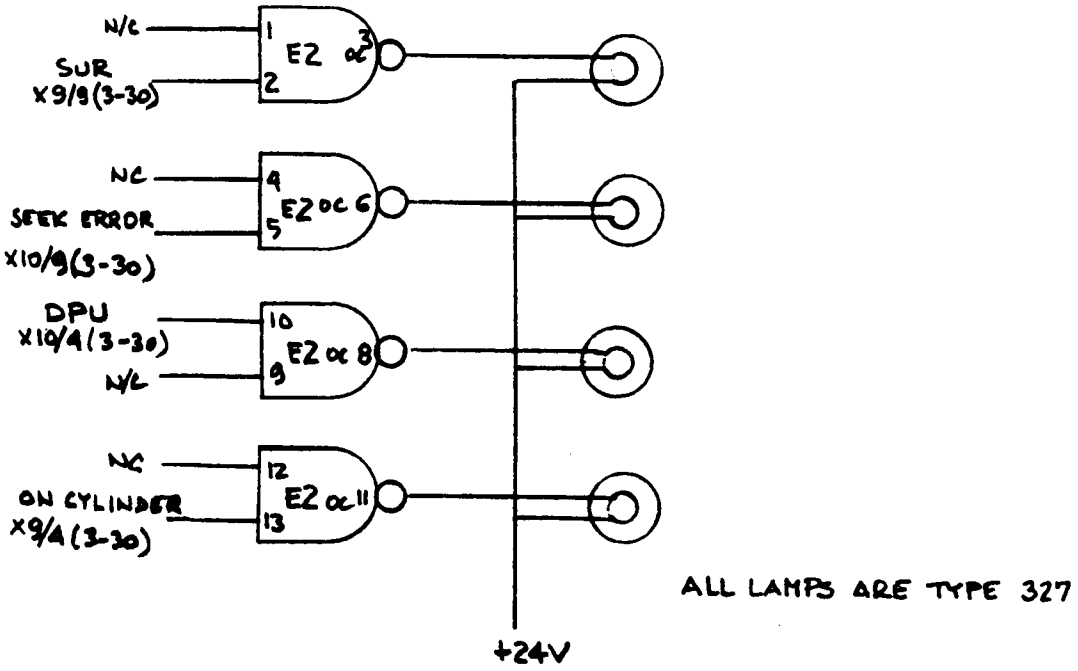


FIG 3-31 CONTROL, TRANSMITTERS/RECEIVERS,



ALL LAMPS ARE TYPE 327

NC : no connection

FIG 3-32 I/O BOX INDICATOR LAMPS

REV.1

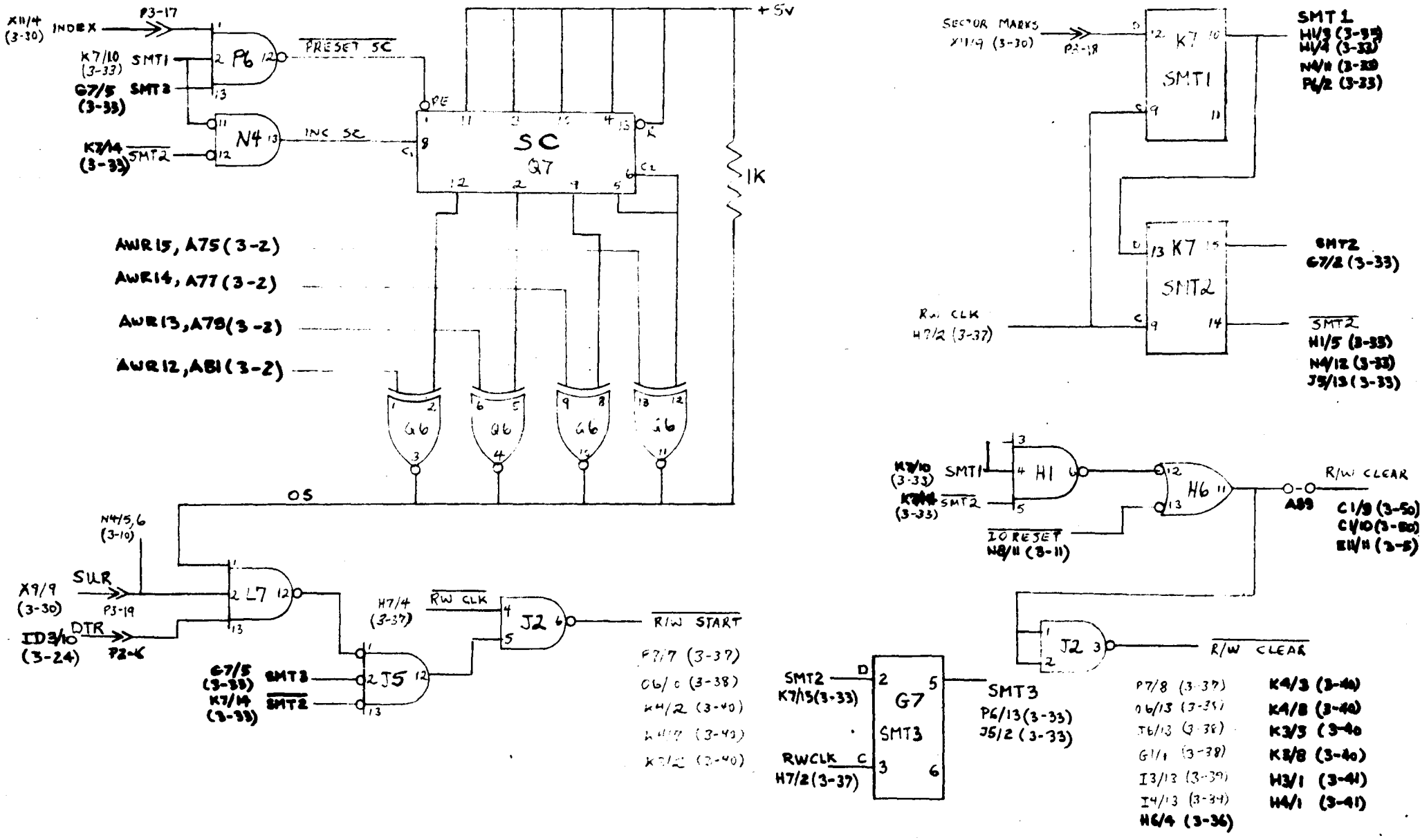
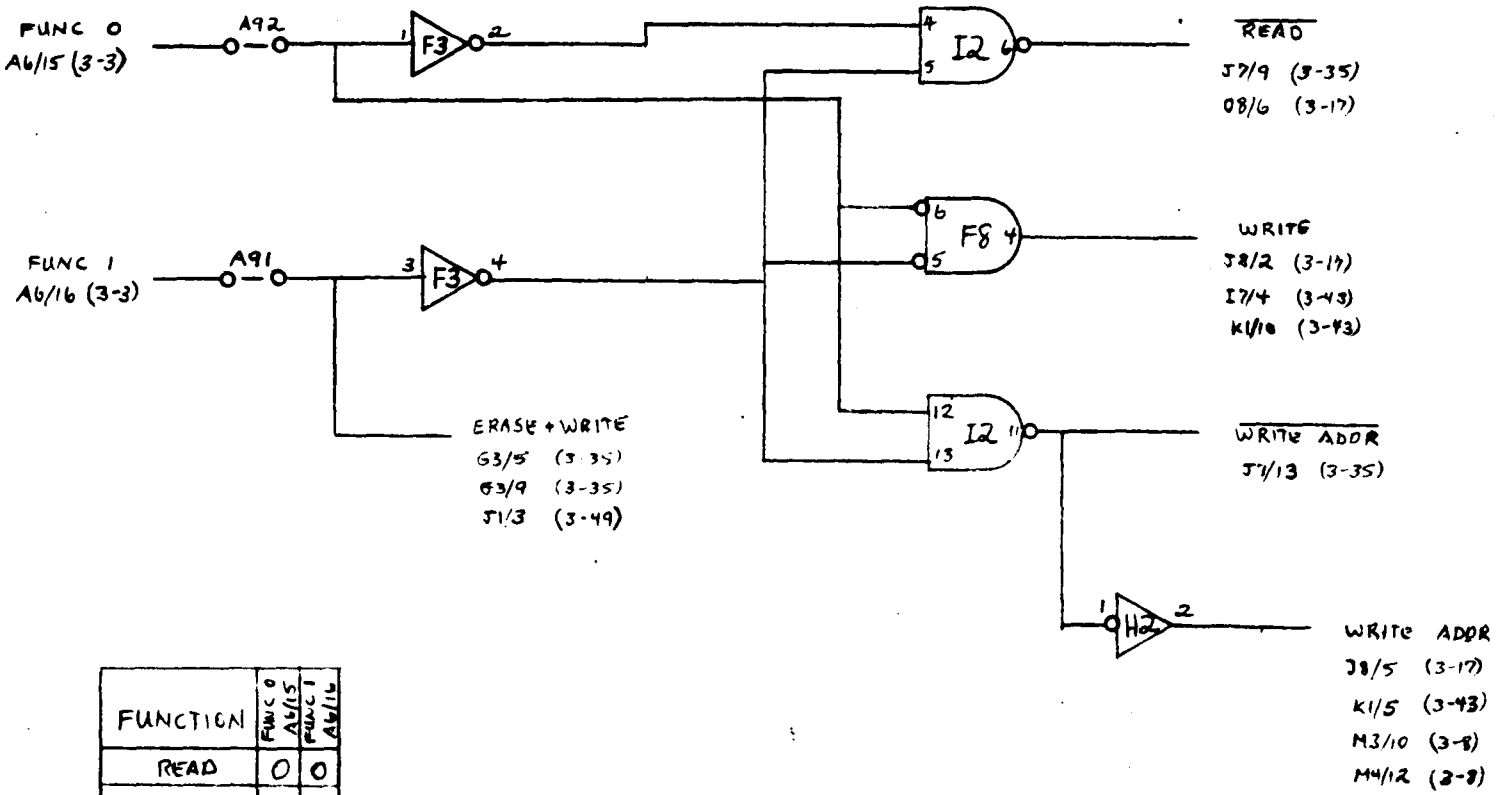


FIG 3-33 ON SECTOR LOGIC

REV 2



FUNCTION	FUNC 0 A6/15	FUNC 1 A6/16
READ	0	0
WRITE	0	1
WRITE ADDR	1	0
ERASE	1	1

FIG 3-34 FUNCTION DECODE LOGIC

REV 1

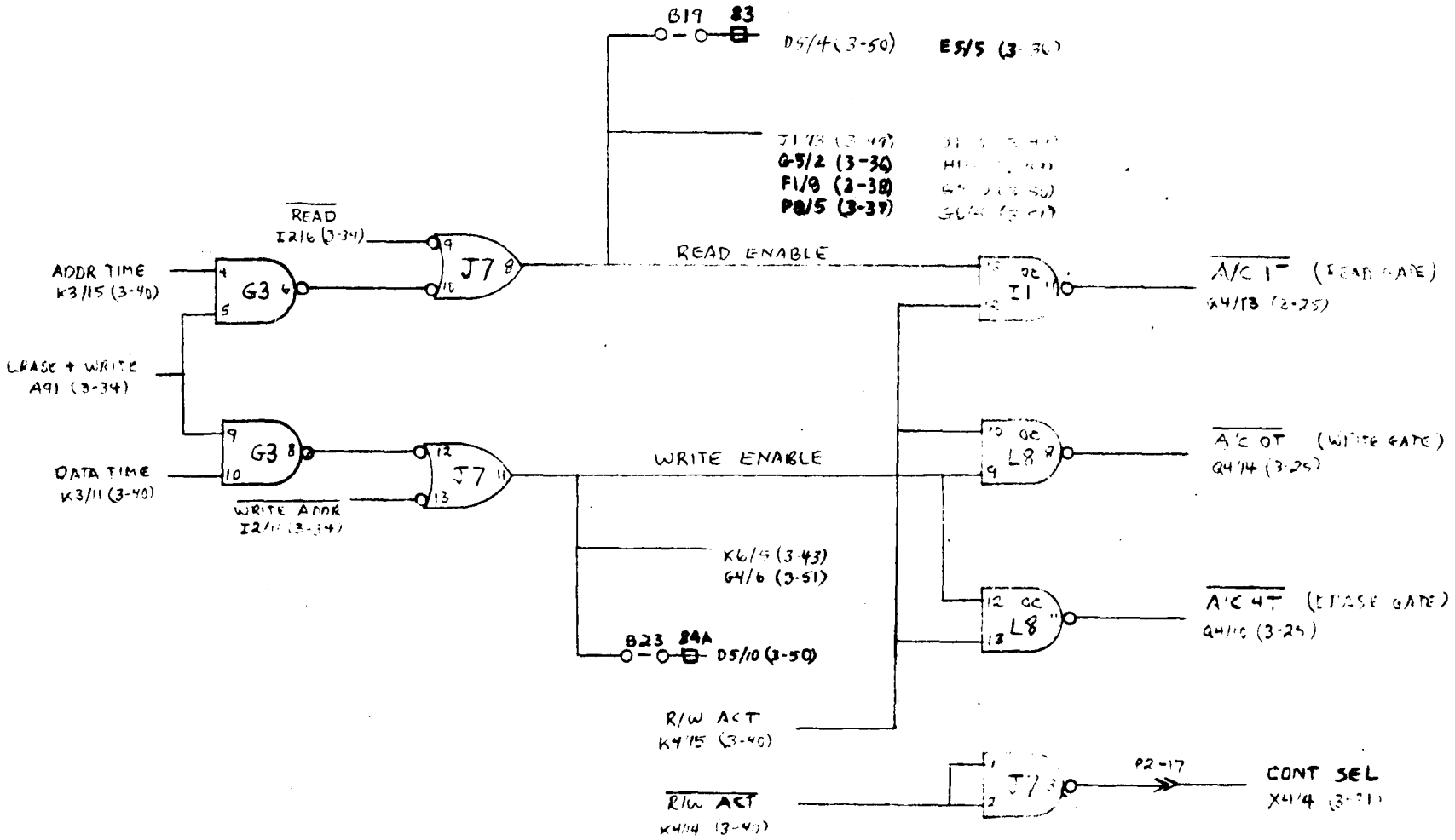


FIG 3-35 READ, WRITE, AND ERASE GATING

REV 2

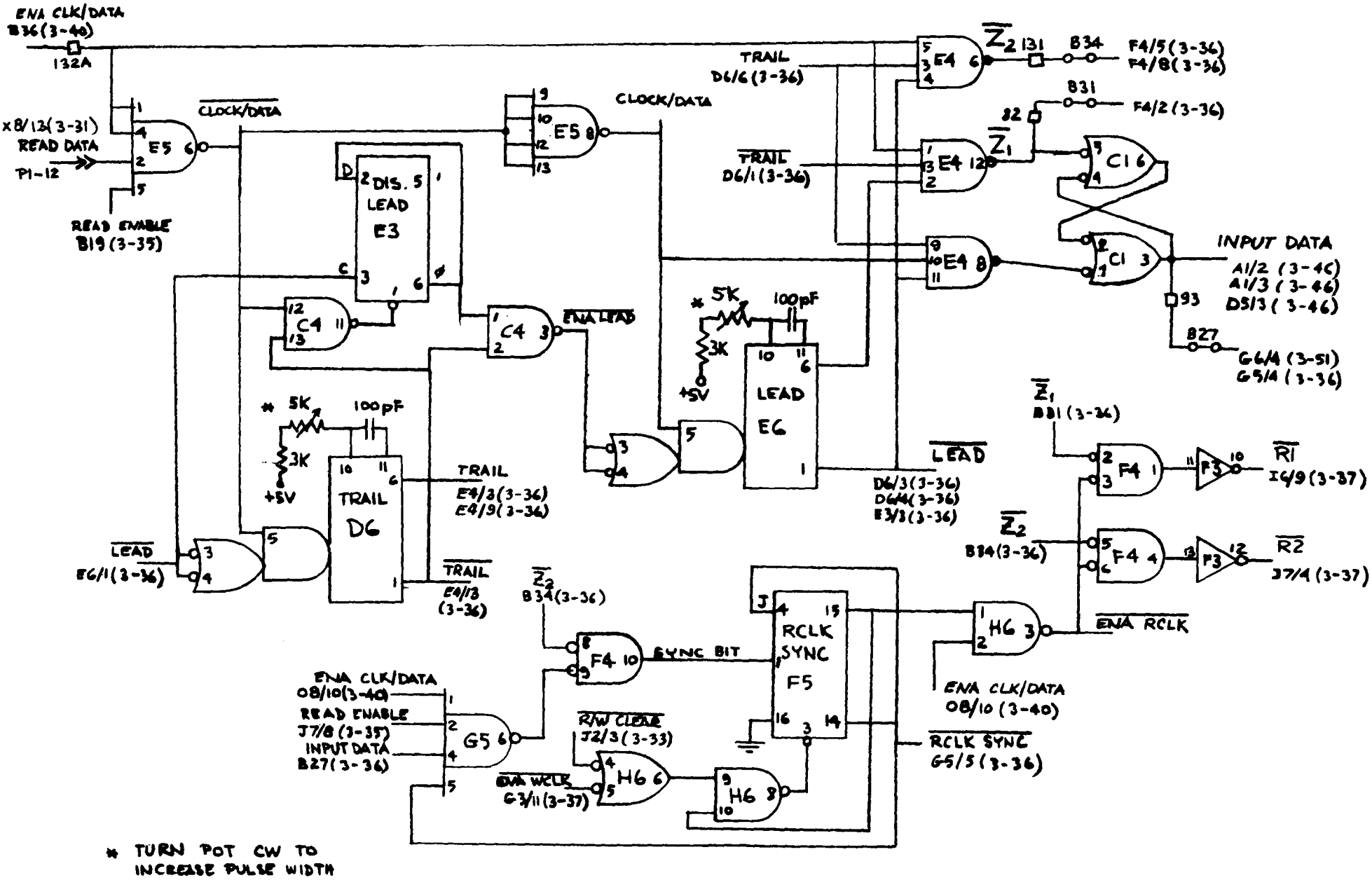


FIG 3-36 READ CLOCK

REV. 1

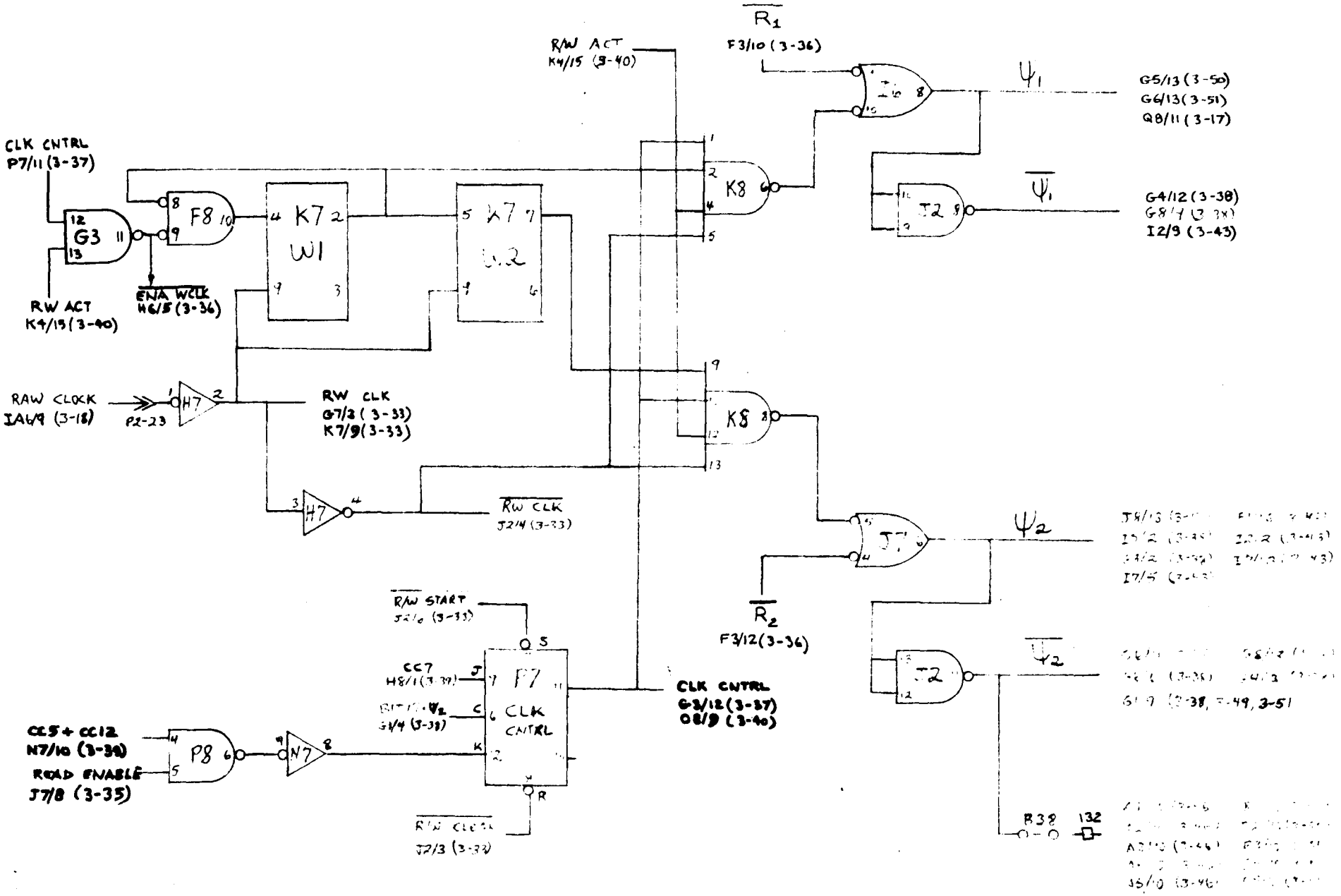


FIG 3-37 WRITE CLOCK

REV 2

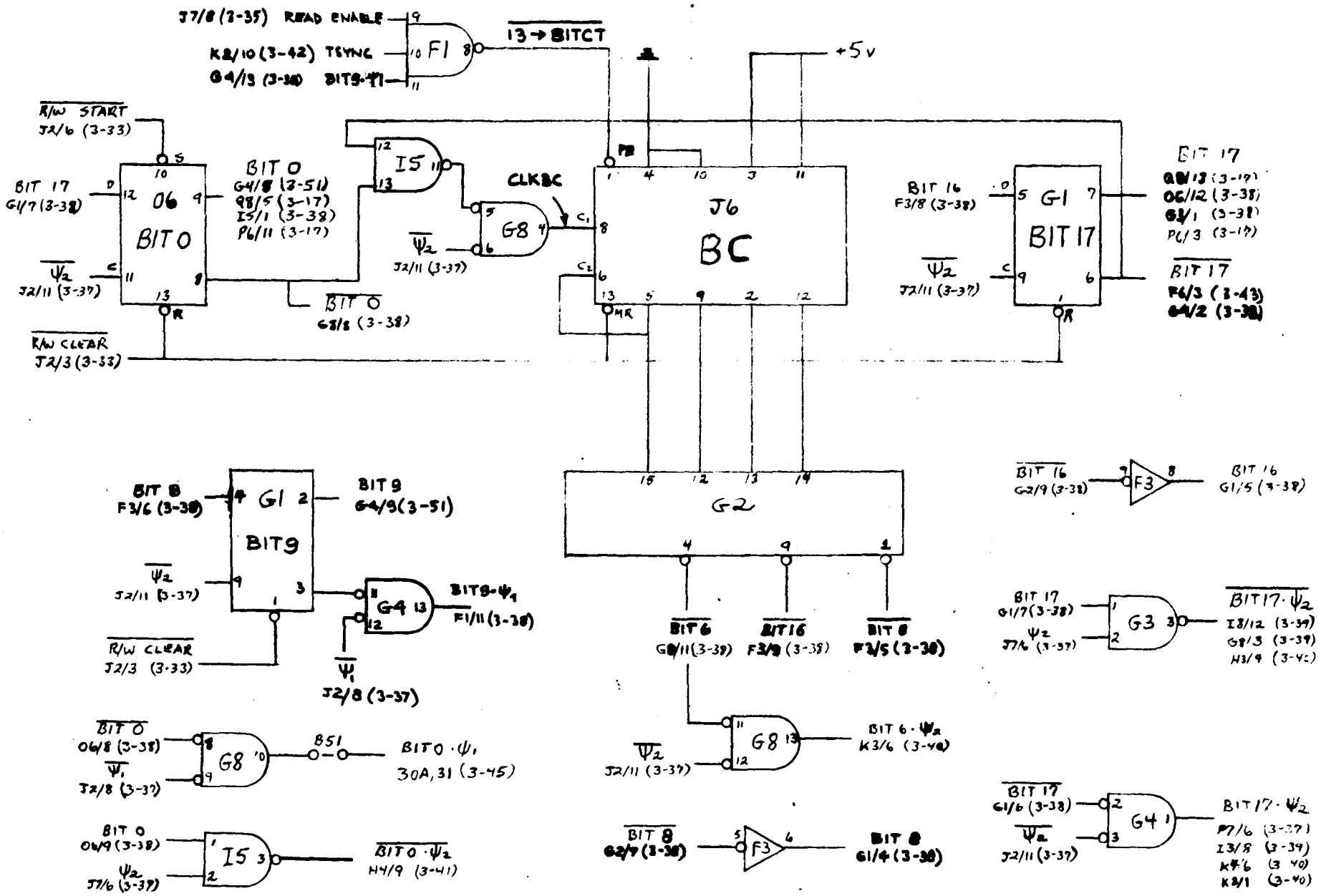


FIG 3-38 BIT COUNTER

REV 2

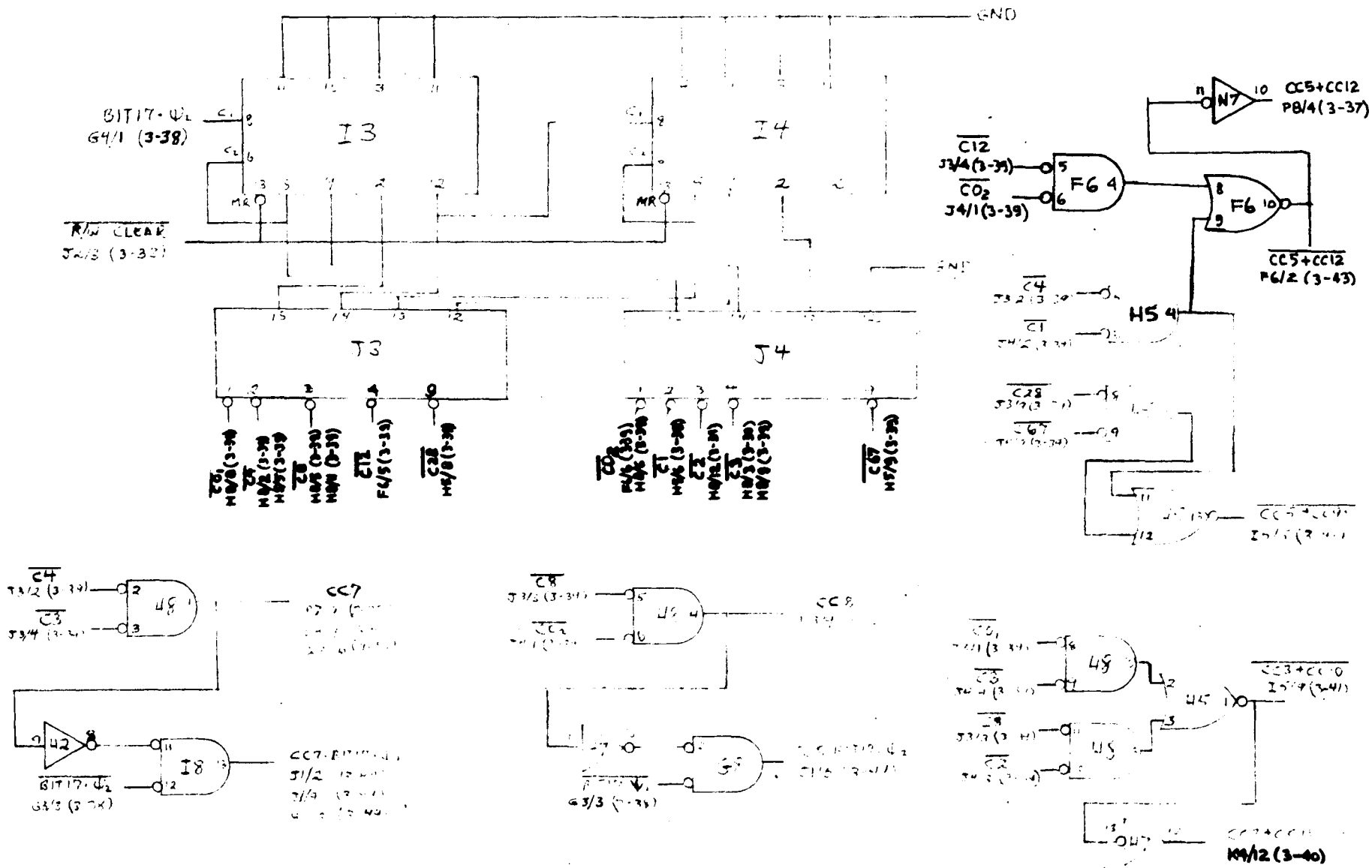


FIG 3-39 CHARACTER COUNTER

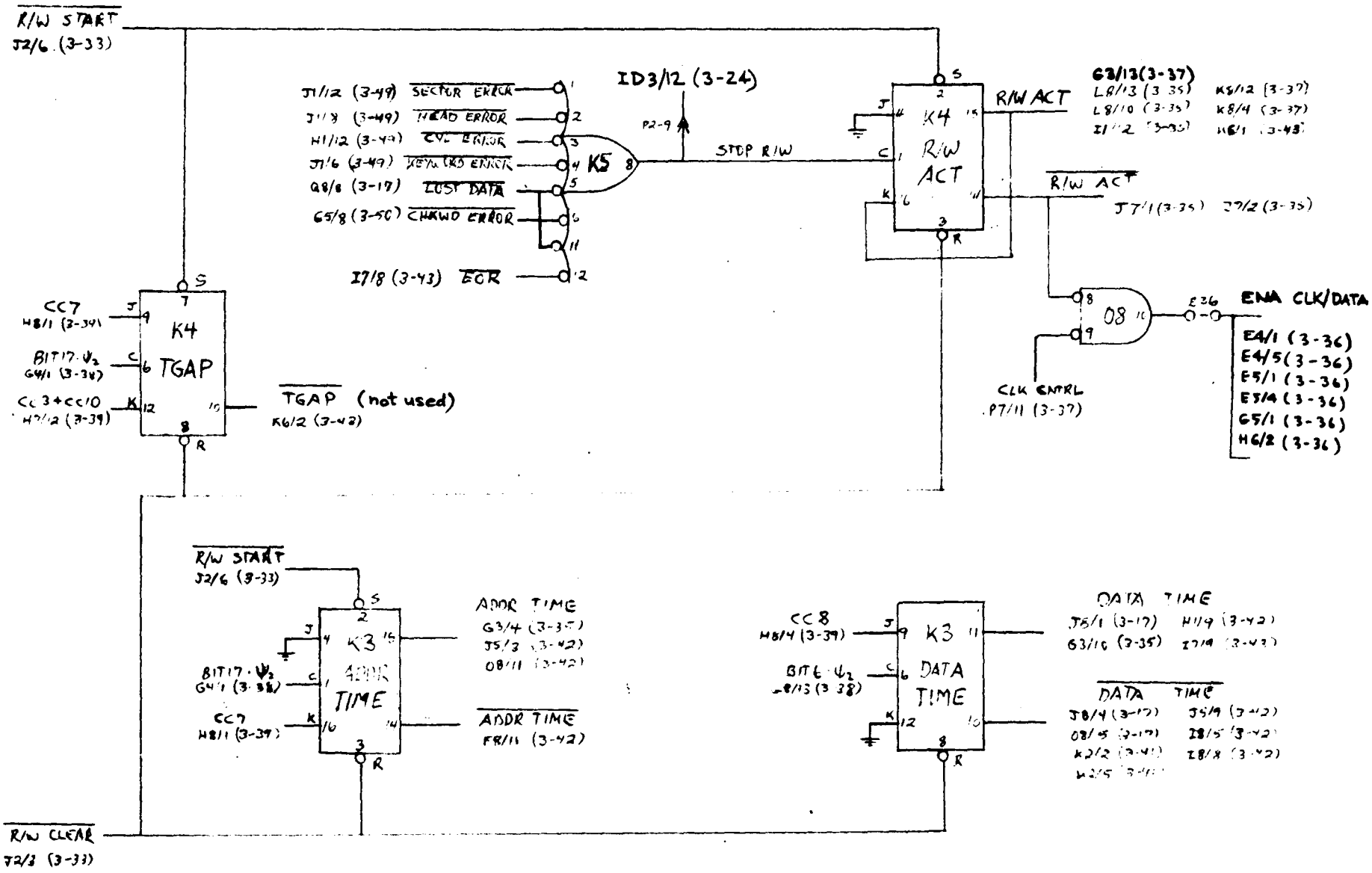


FIG 3-40 READ/WRITE TIMING (1)

REV 2

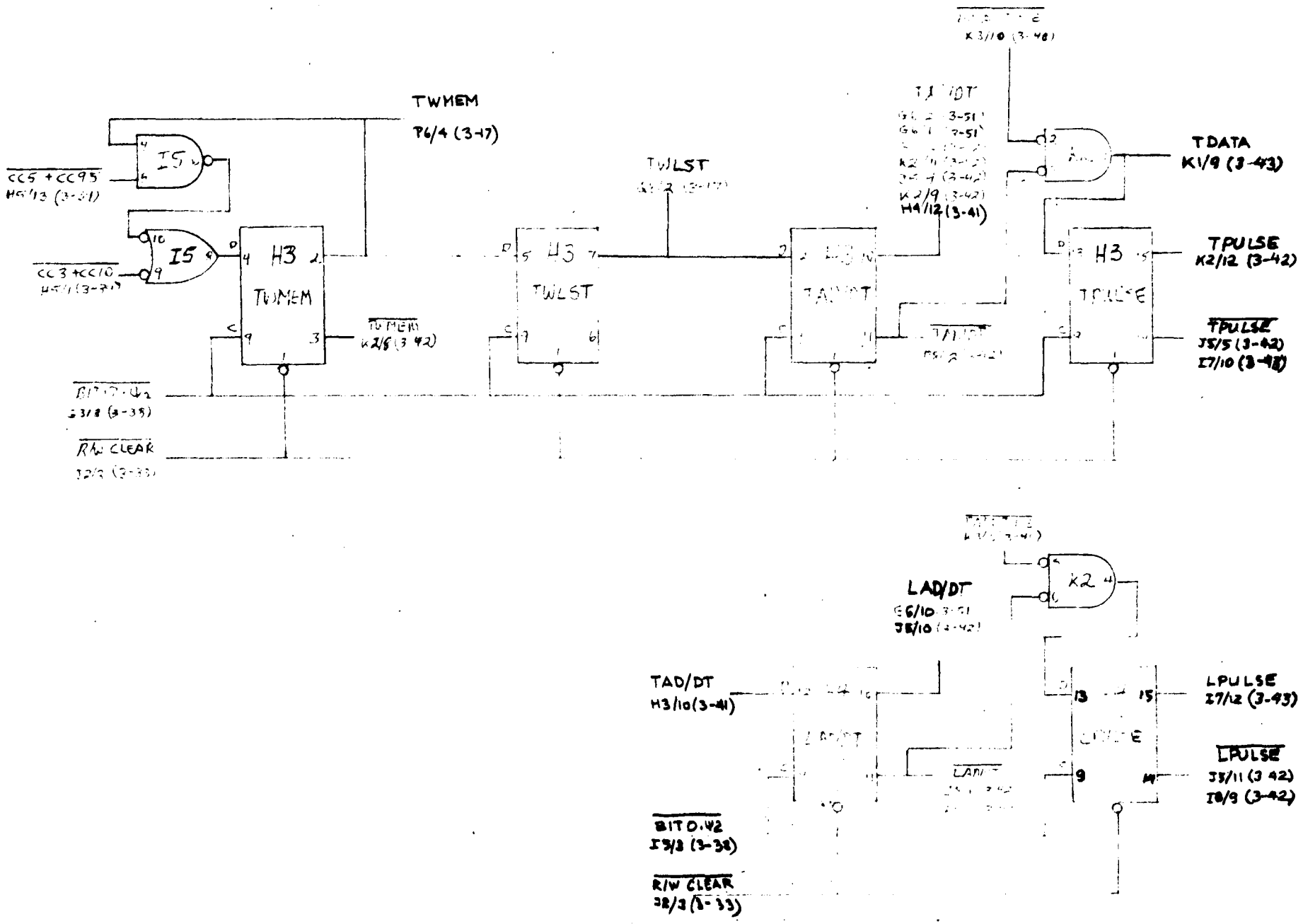


FIG 3-41 READ/WRITE TIMING (2)

REV 2

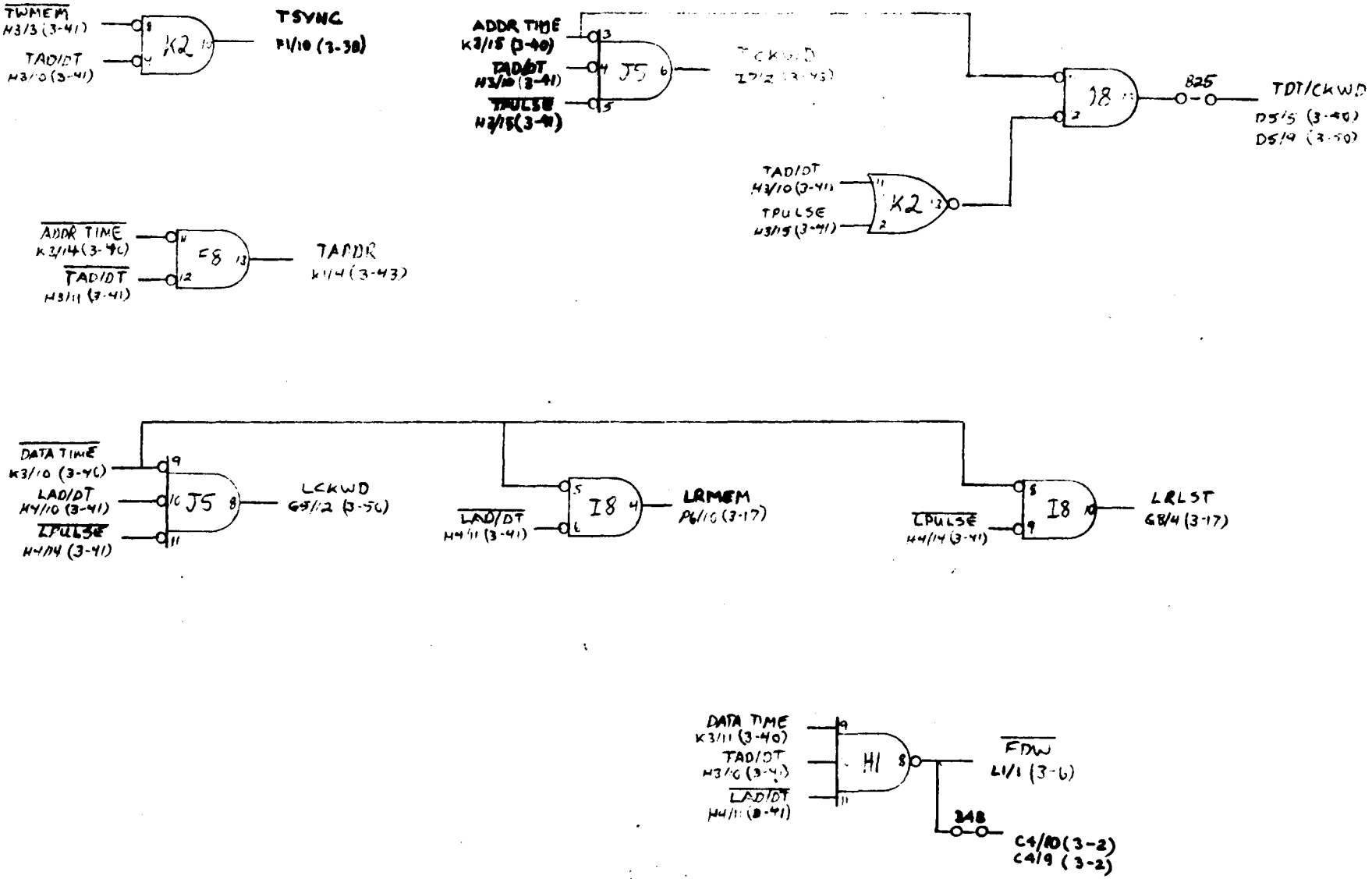


FIG 3-42 READ/WRITE TIMING (3)

REV 2

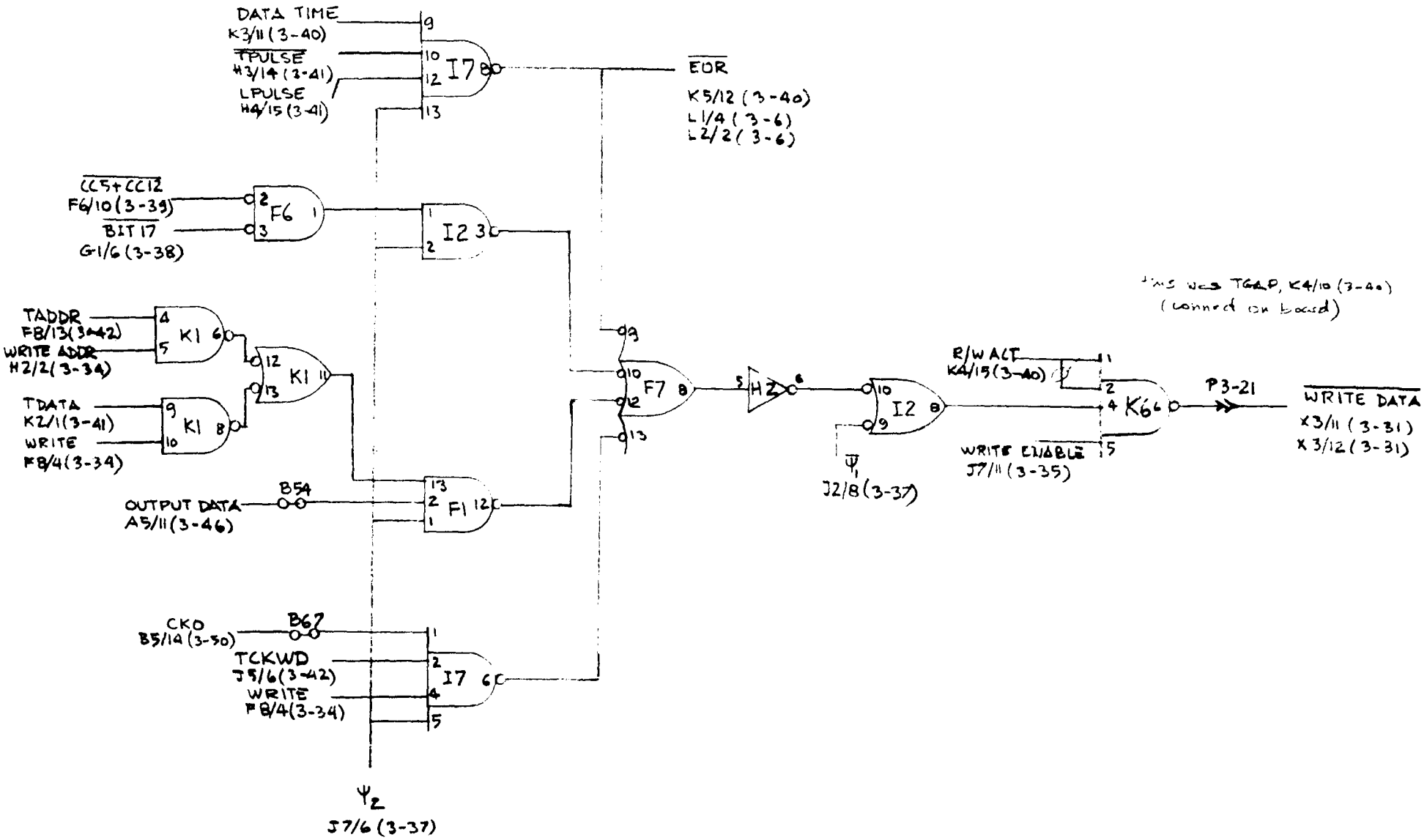
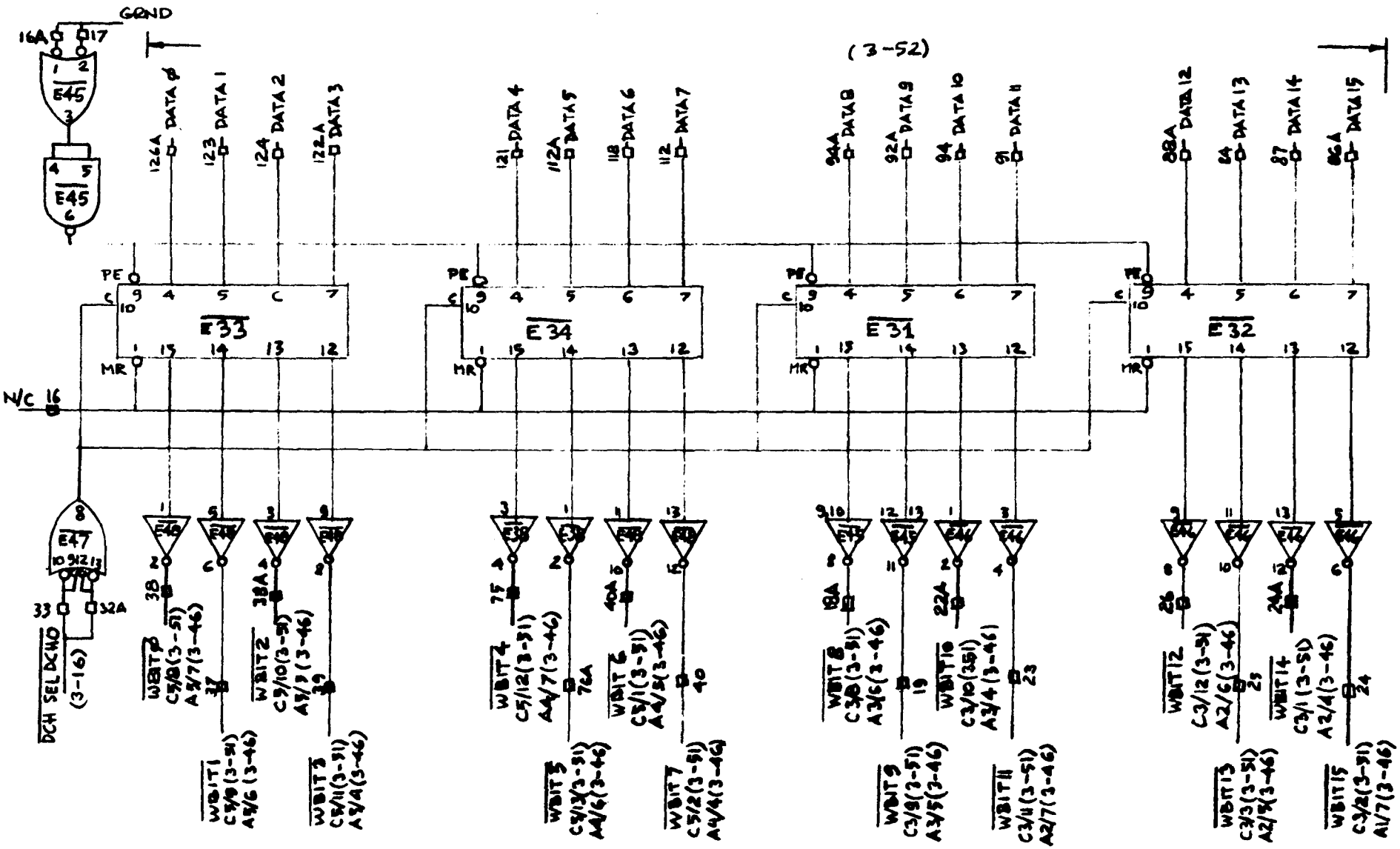


FIG 3-43 WRITE DATA LOGIC

REV. 2



E31 THROUGH E34: SN74195

(10) CLOCK: ⌐¯|_

(9) PE : HIGH = SHIFT
LOW = PARALLEL LOAD

FIG 3-44 WRITE BUFFER REGISTER

REV. 1

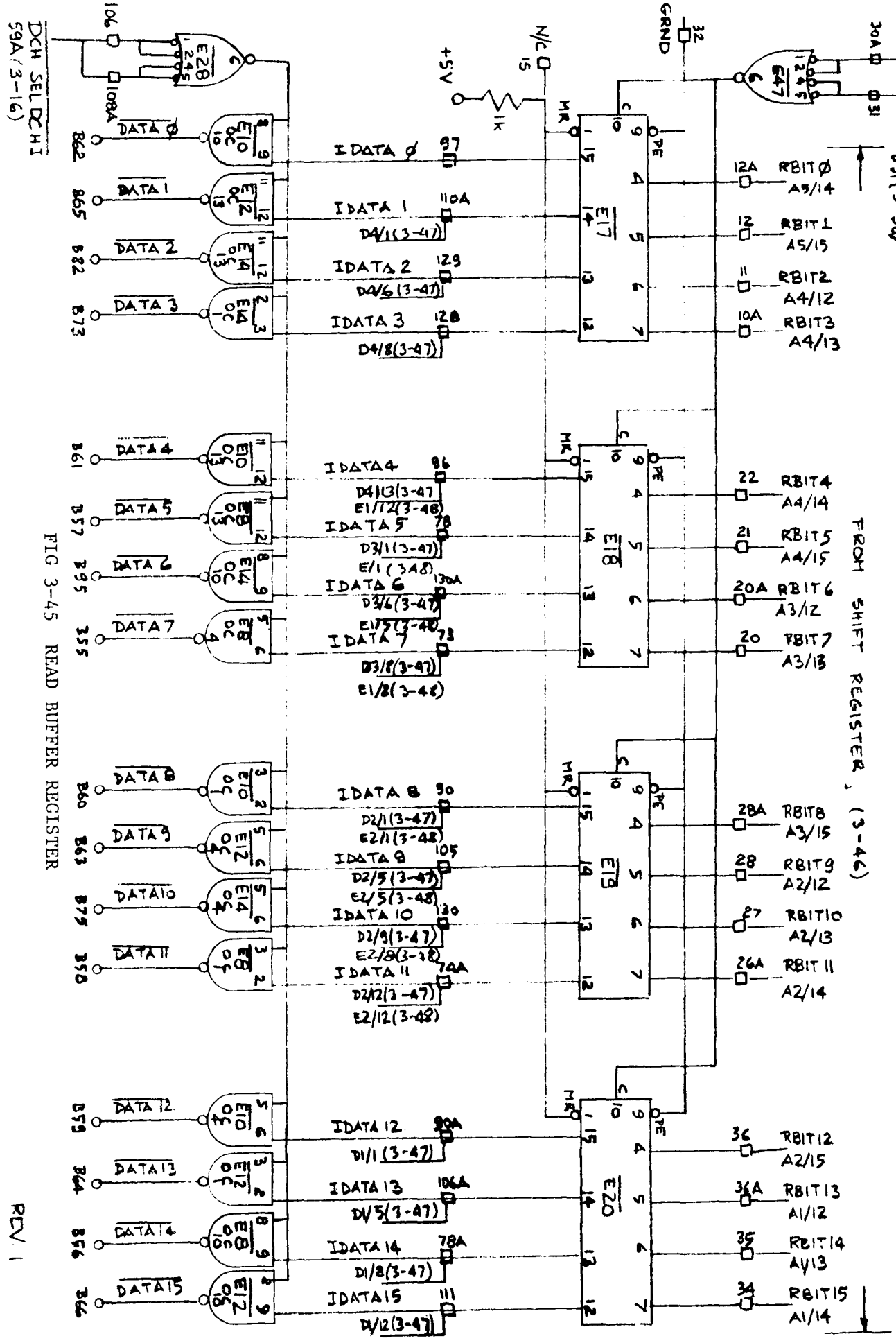
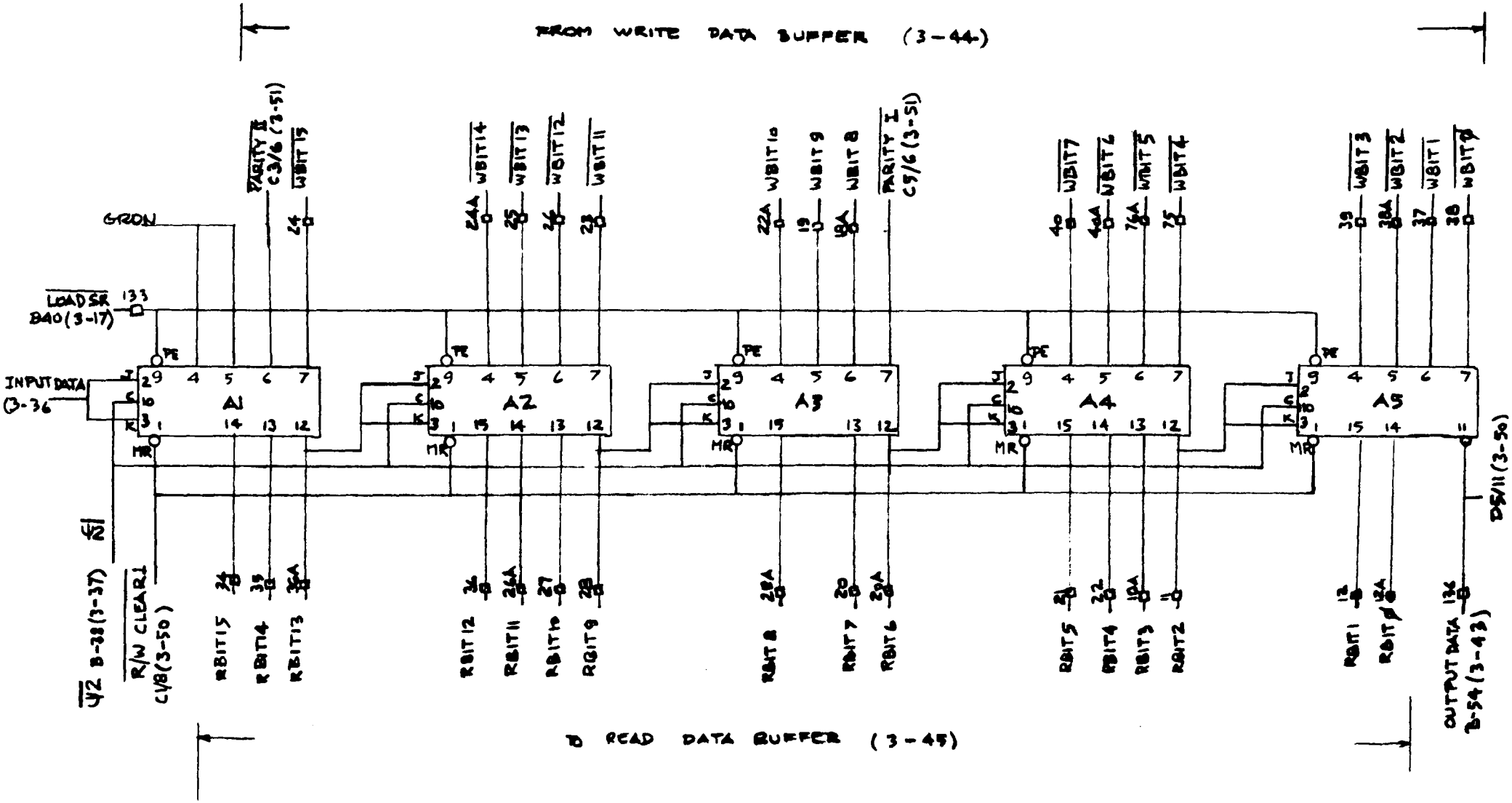


FIG 3-45 READ BUFFER REGISTER

REV 1



A1 THROUGH A5 : SN74195

CLOCK(10):

PE(9) : HIGH = SHIFT
LOW = PARALLEL LOAD

FIG 3-46 IN/OUT SHIFT REGISTER

REV. 1

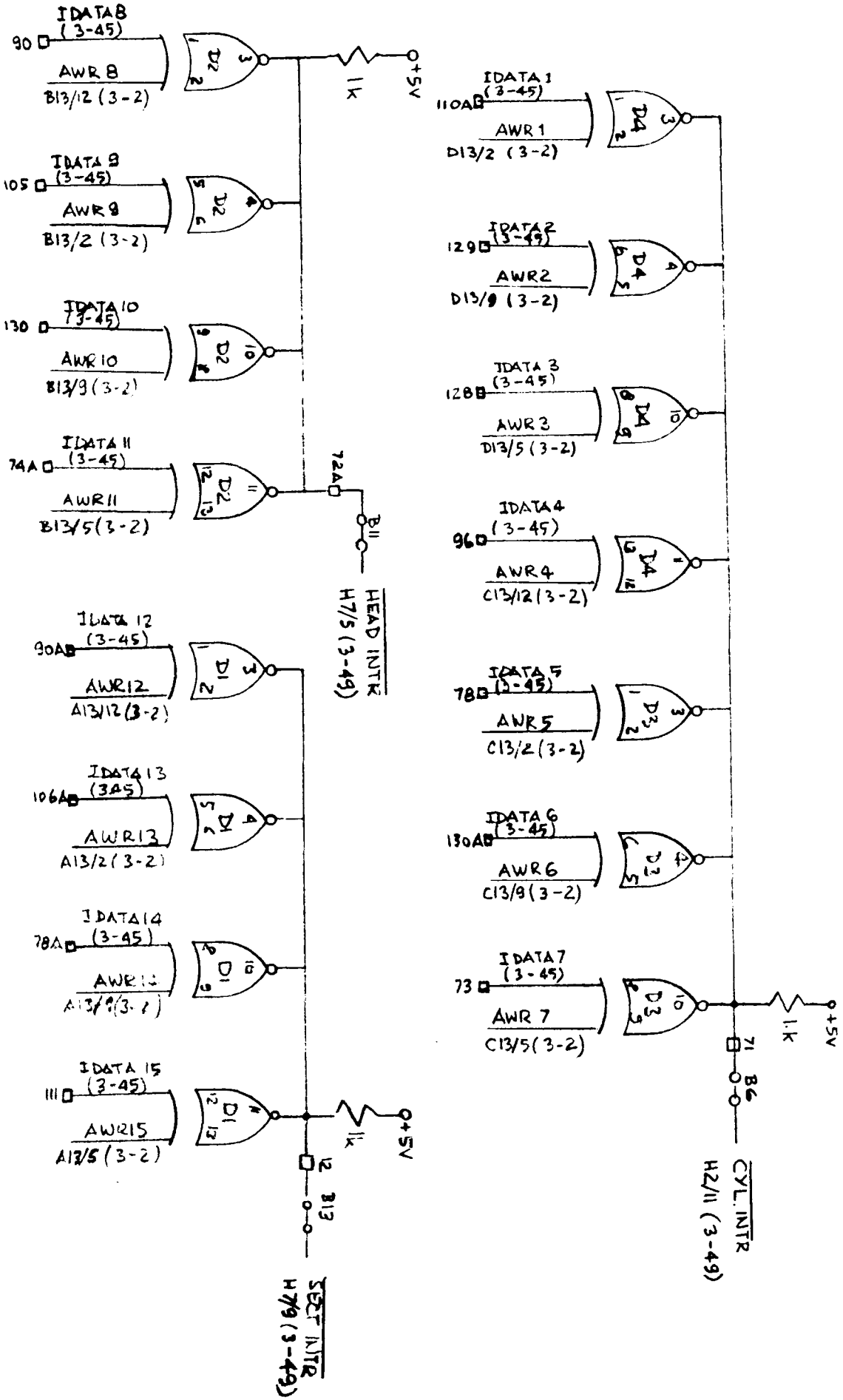


FIG 3-47 COMPARATORS (1)

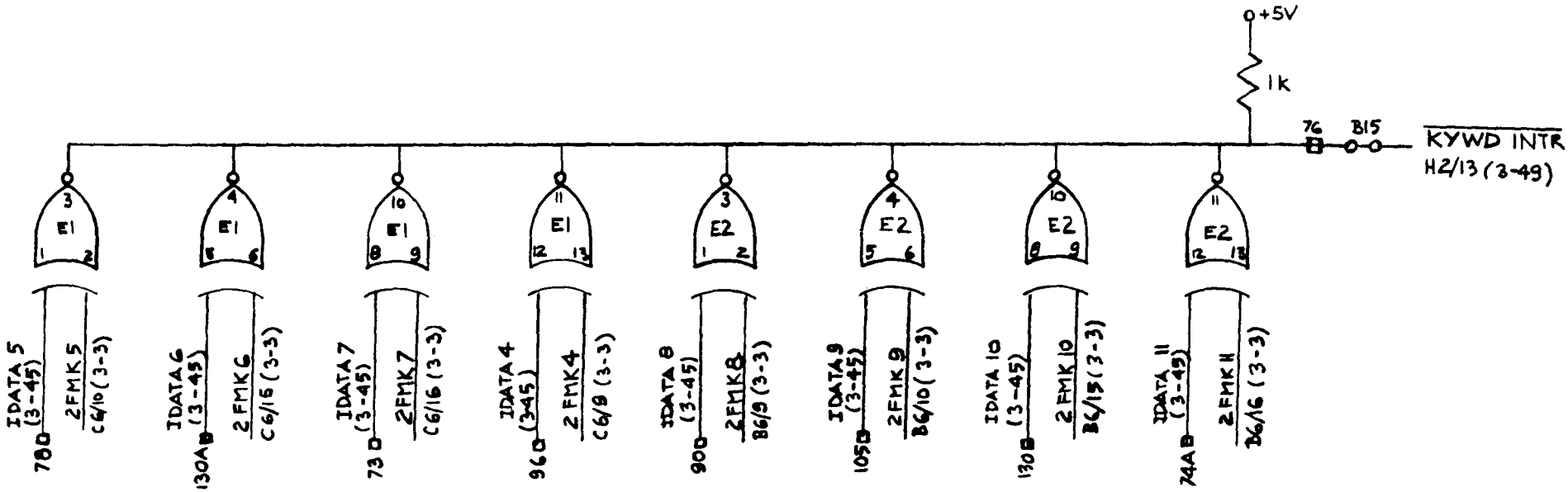
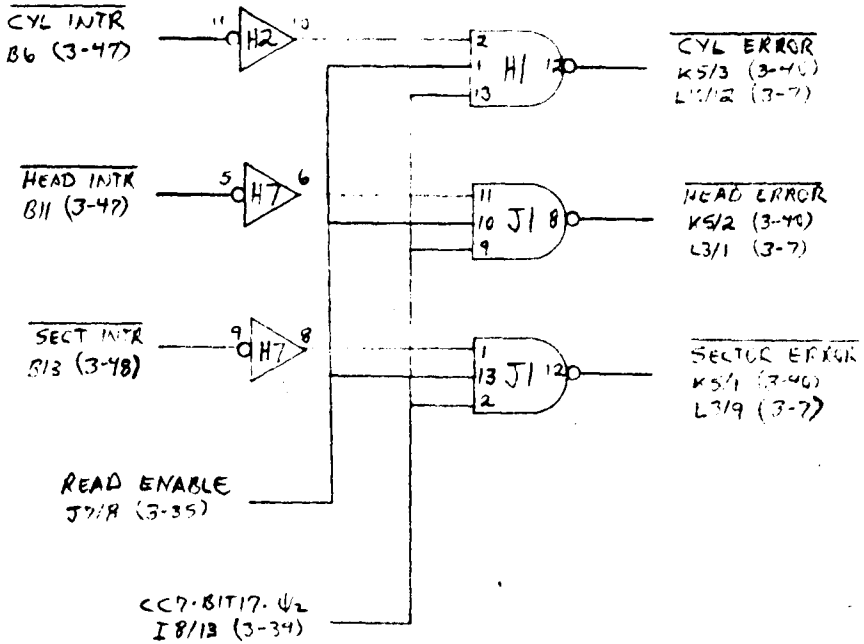


FIG 3-48 COMPARATORS (2)

REV. 1



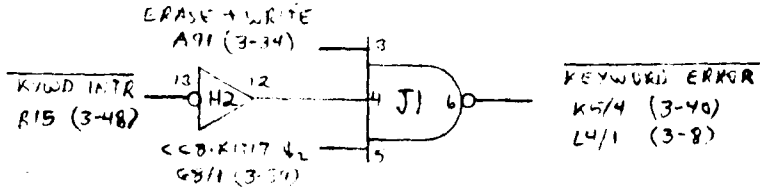


FIG 3-49 CYL, HD, SECT, KYWD

REV 2

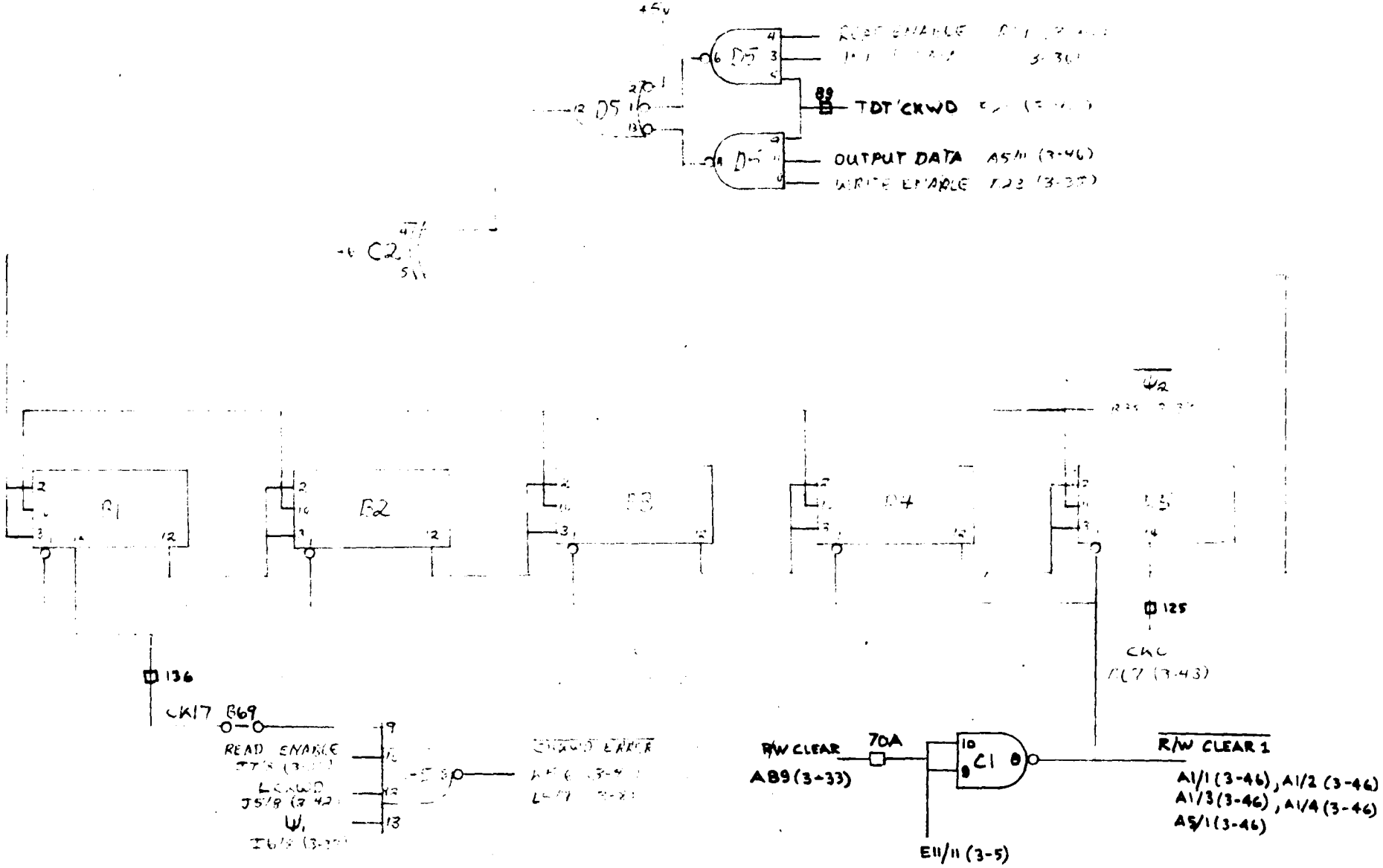


FIG 3-50 CHECKWORD REGISTER

REV 2

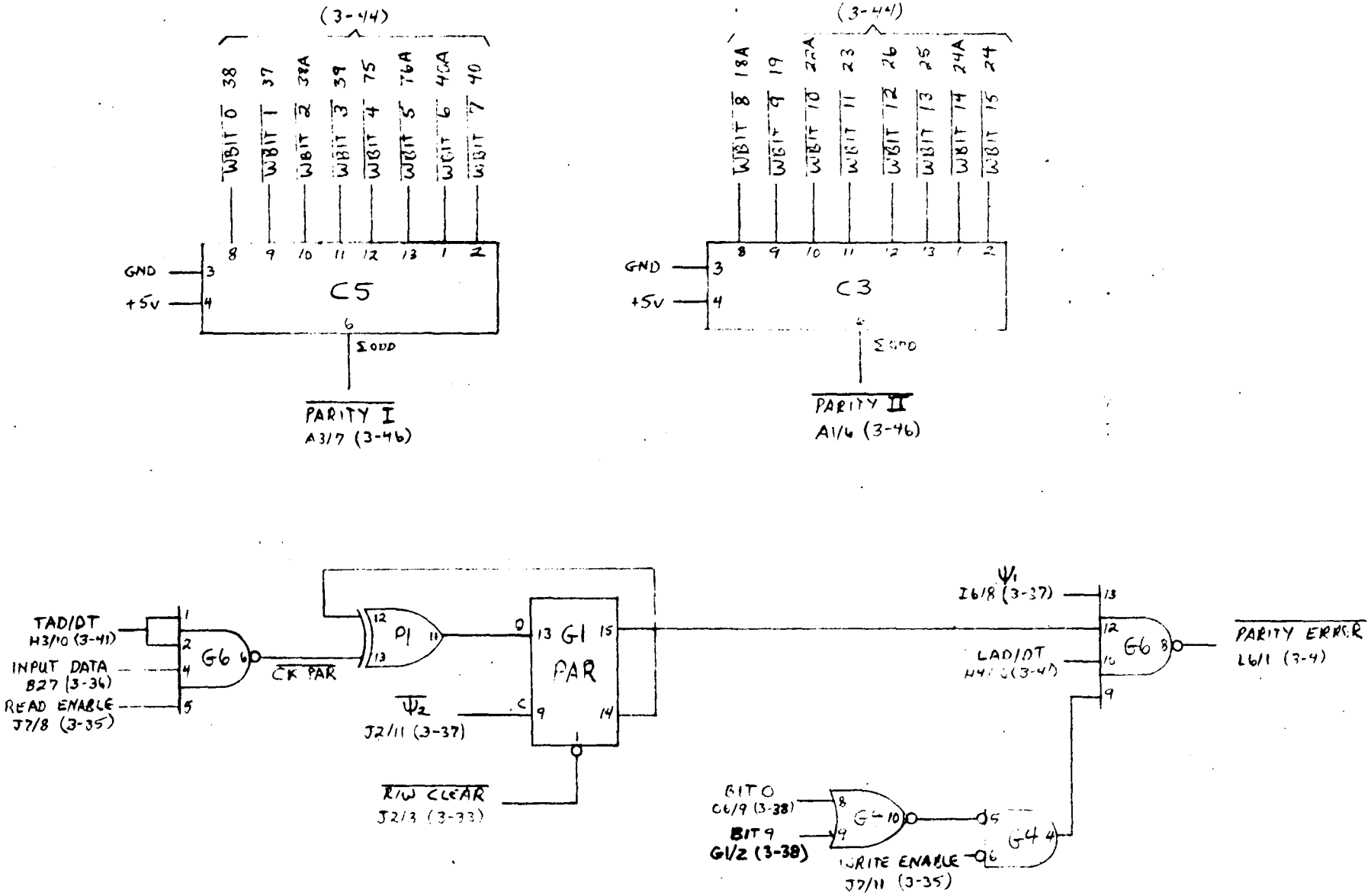


FIG 3-51 PARITY GENERATION AND TEST

REV 2

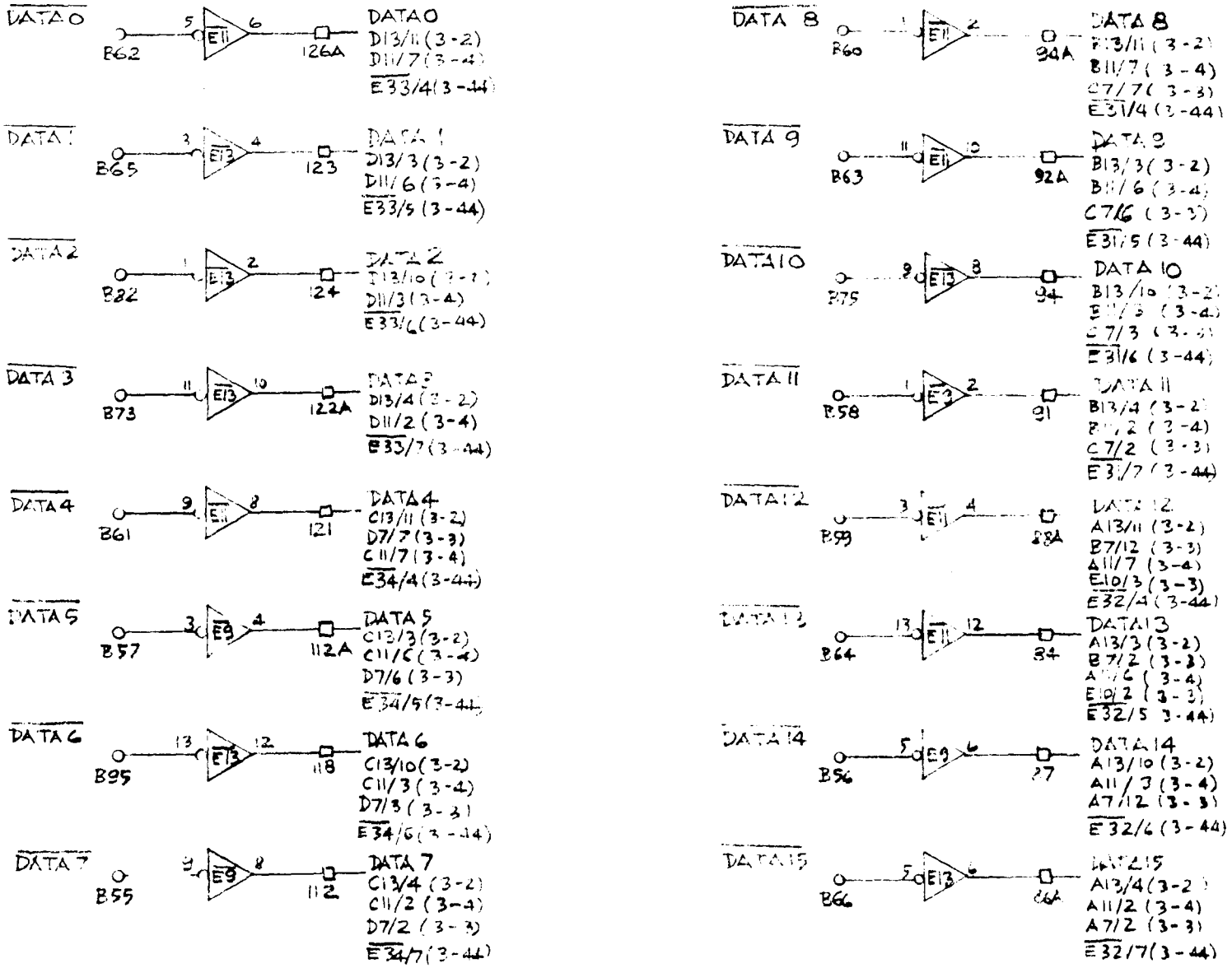


FIG 3-52 NOVA I/O BUS DATA RECEIVE LOGIC

REV.1

4. TIMING DIAGRAMS

Section 4 contains the disk controller timing diagrams.

4.1 TIMING DIAGRAM LIST

4.2 TIMING DIAGRAMS

The disk controller timing diagrams appear starting on the next page.

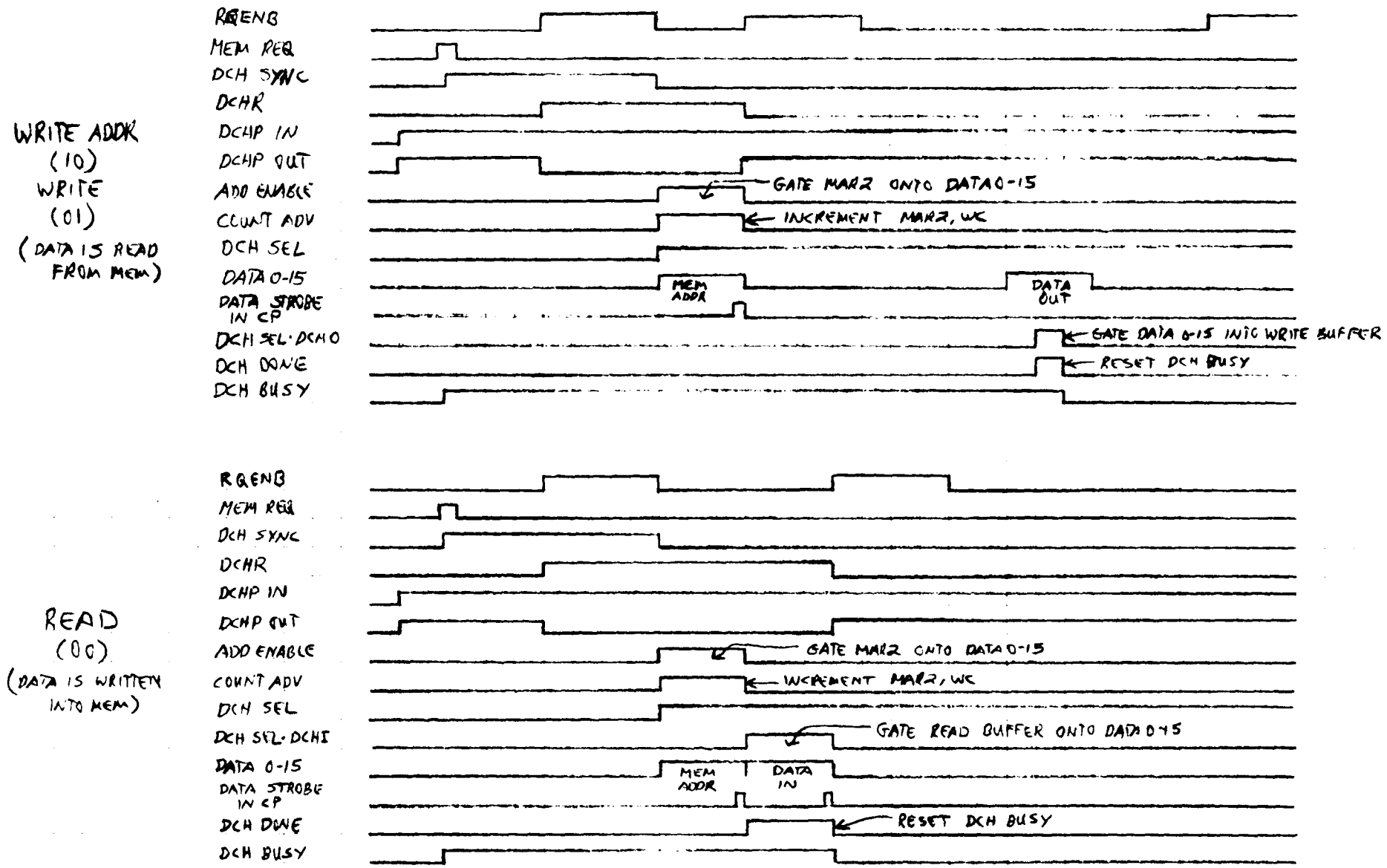
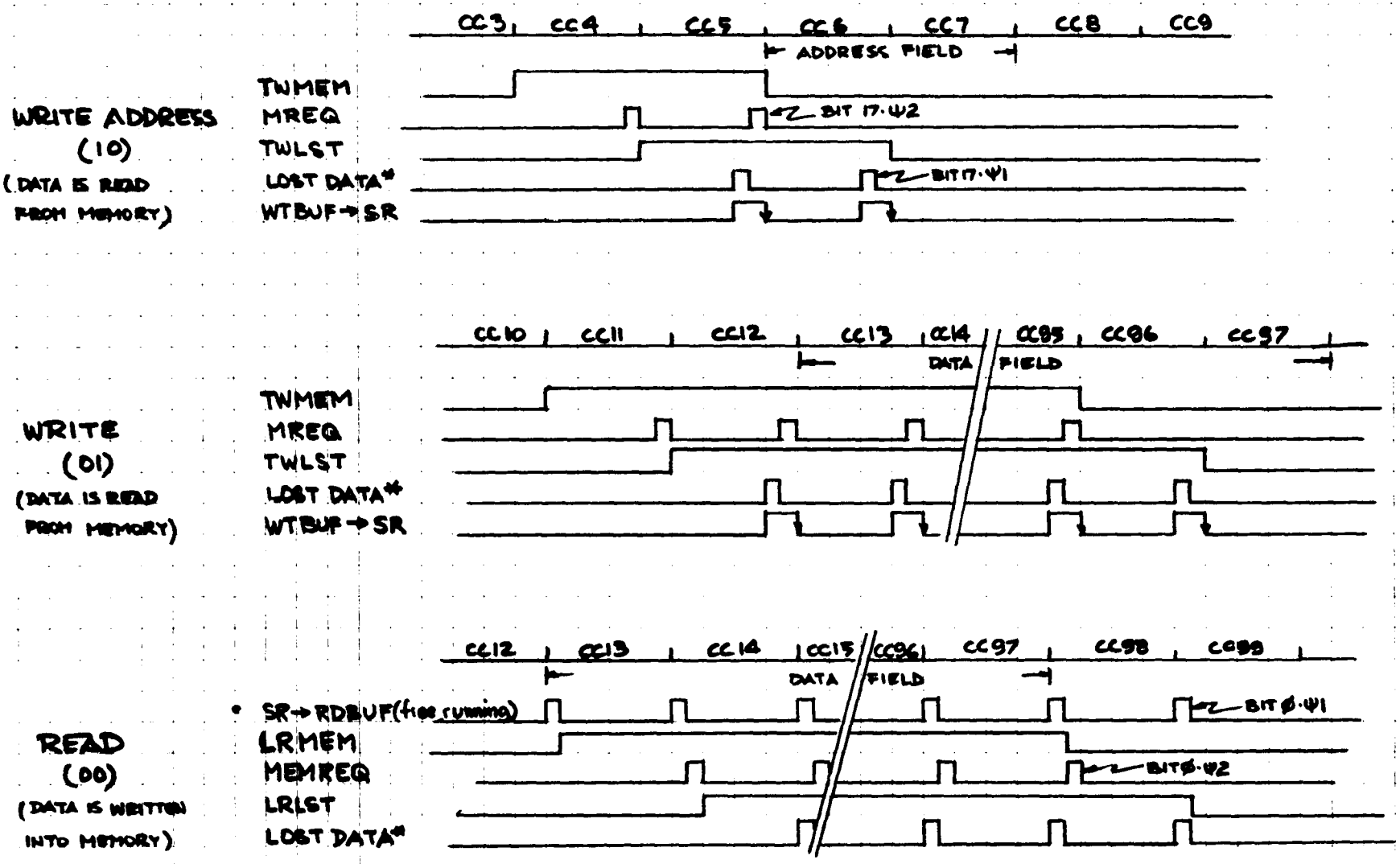


FIG. 4-1 DATA CHANNEL TIMING



* LOSS OF DATA IS CHECKED FOLLOWING EACH MEMORY REQUEST. THEY SHOULD NOT OCCUR DURING NORMAL OPERATION, IN WHICH CASE THE PULSE TRAIN WILL NOT BE SEEN.

FIG. 4-2. MEMORY REQUEST TIMING

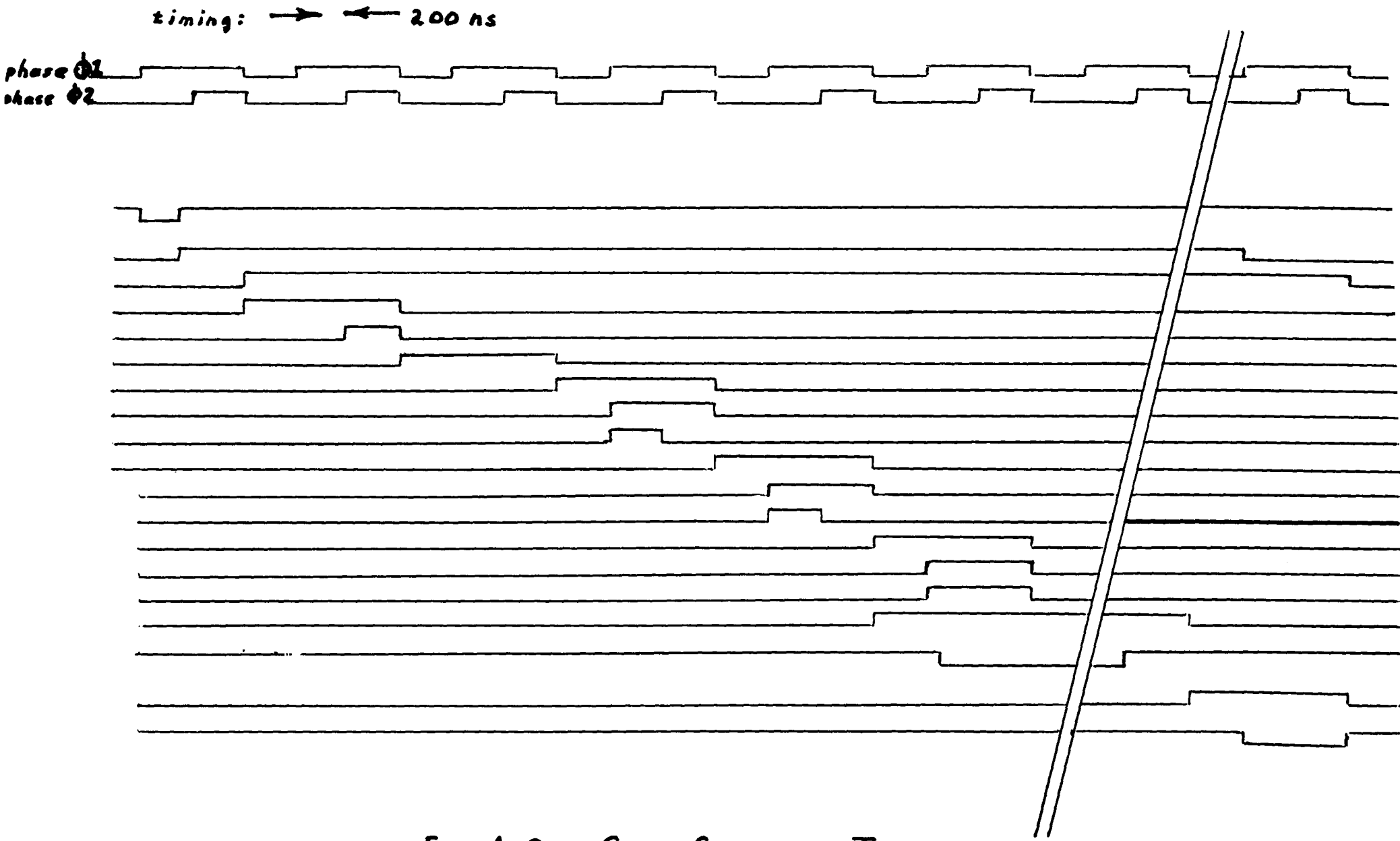


FIG. 4-3 SEEK SEQUENCER TIMING

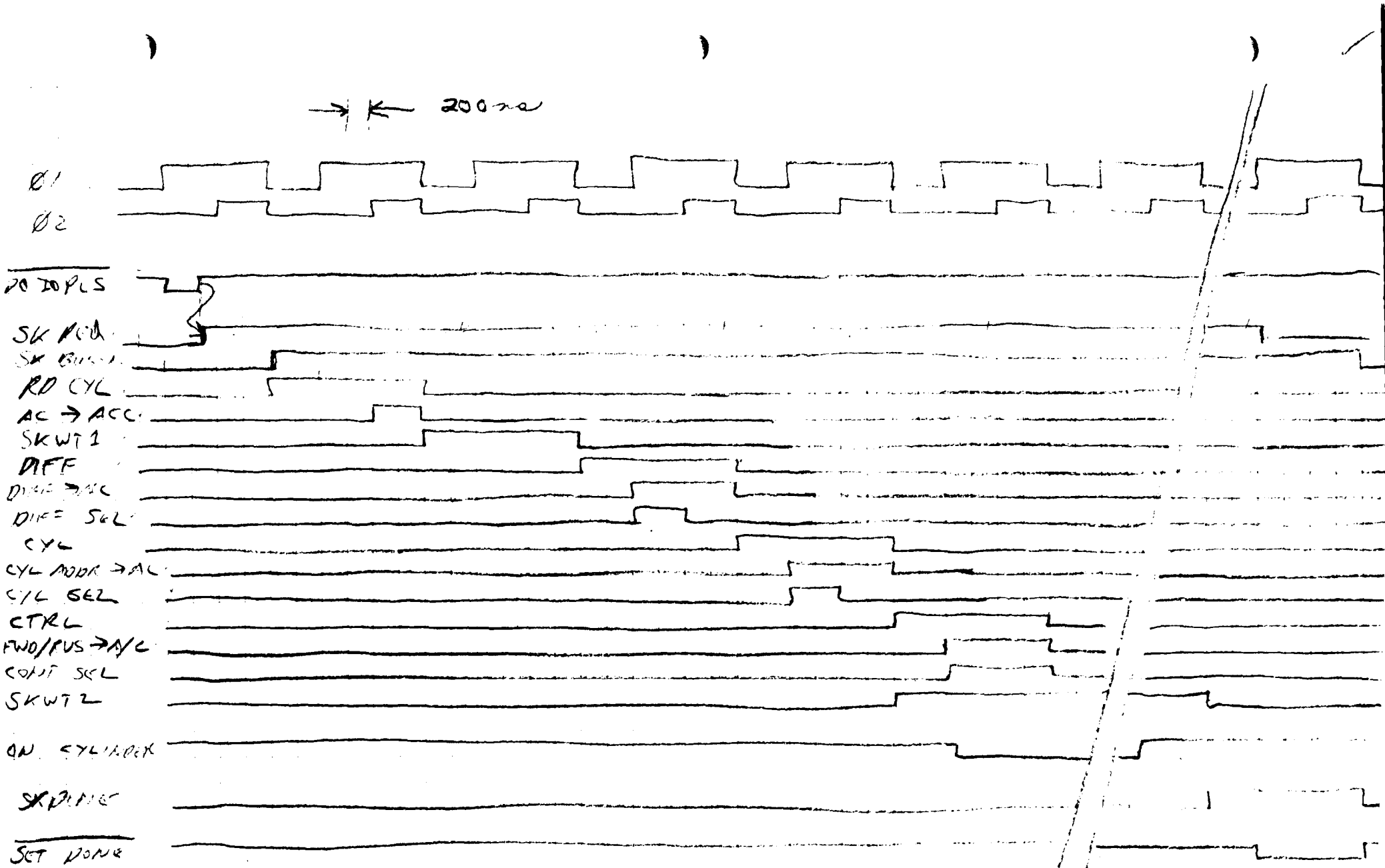


FIG 4-3 SEEK SEQUENCER TIMING

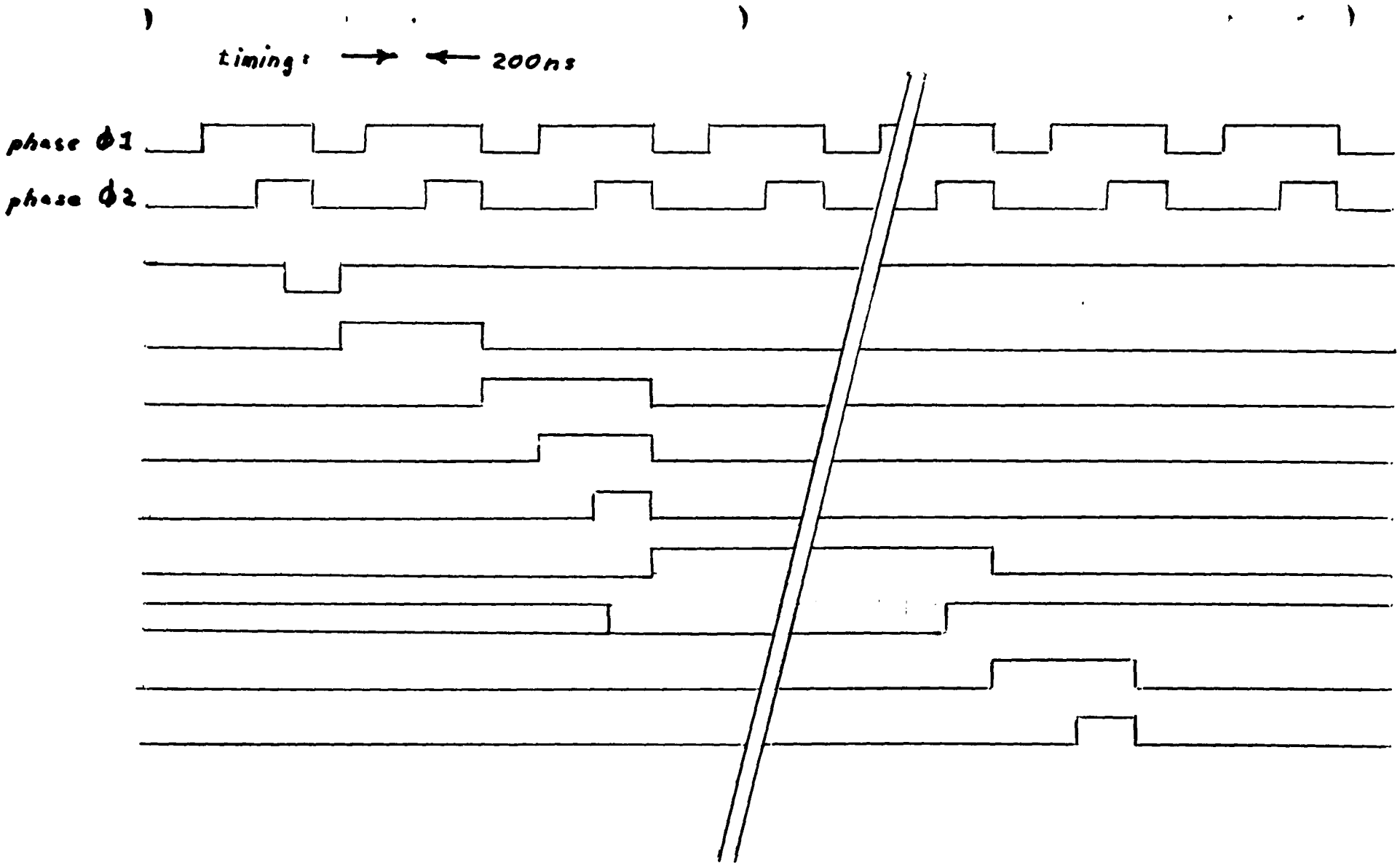


FIG. 4-4 RTZS SEQUENCER TIMING

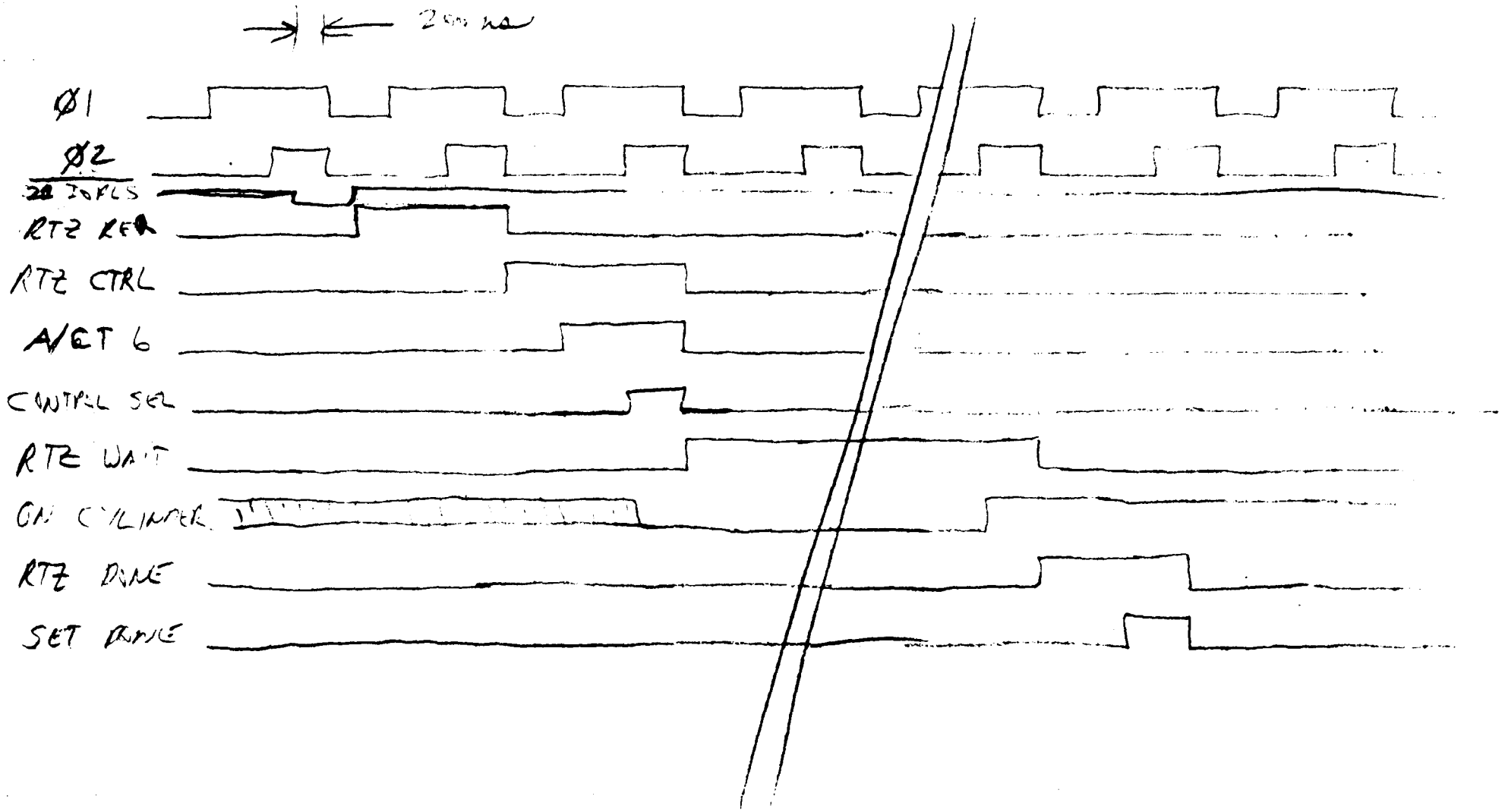


FIG 4-4 RTZ TIMING

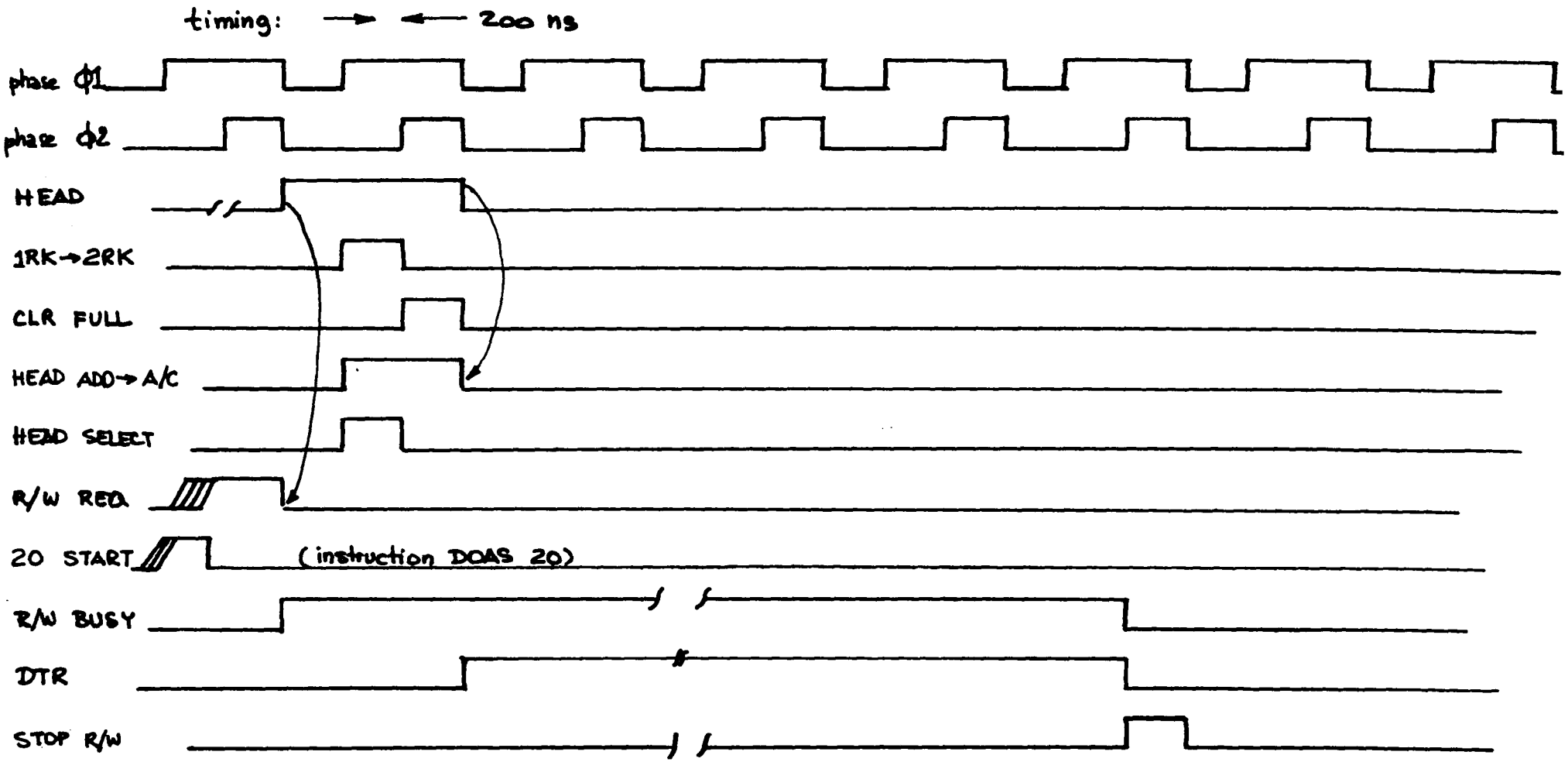


FIG. 4-5 DTR SEQUENCER (READ/WRITE TIMING)

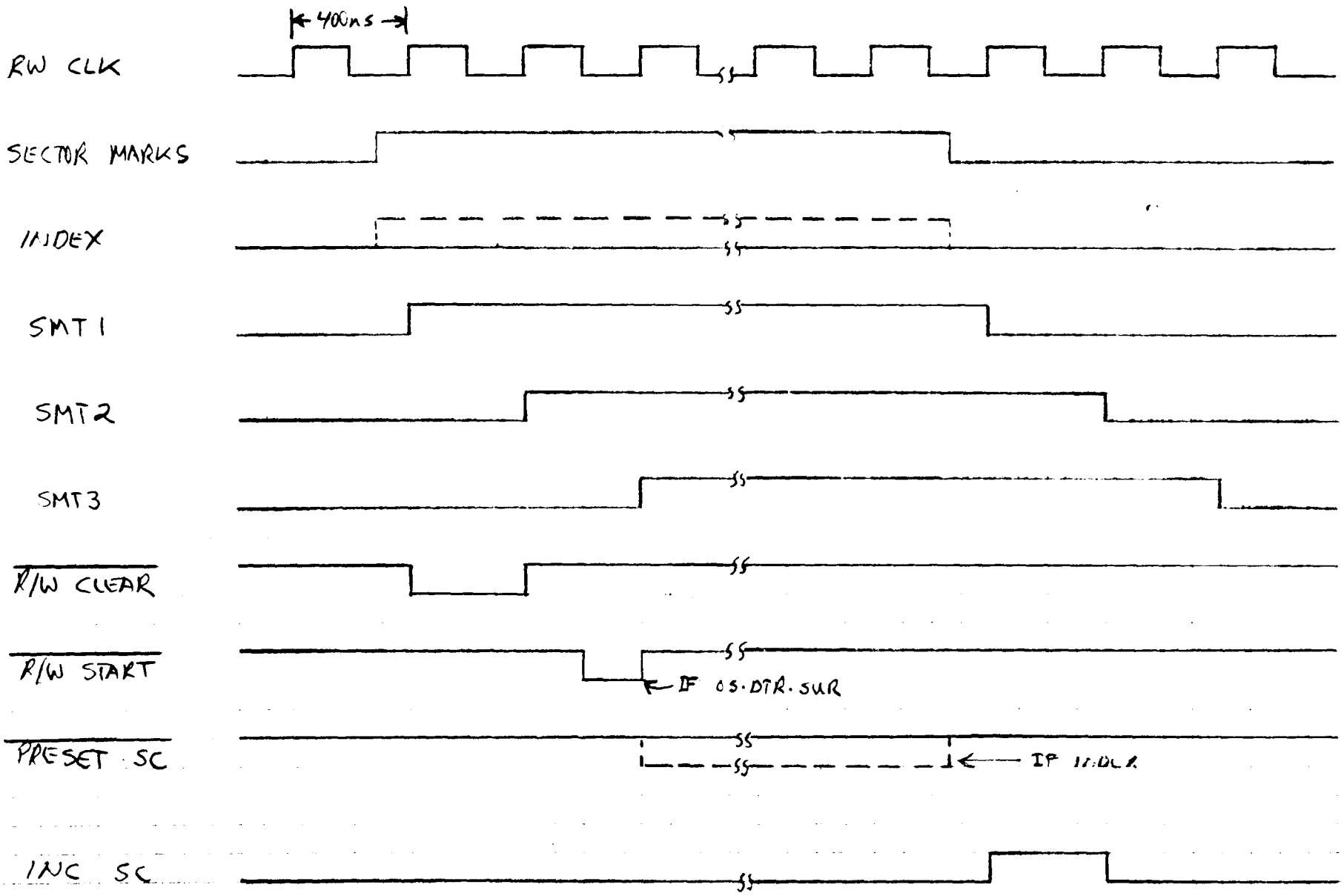


FIG 4-6 SECTOR MARK TIMING

REV 2

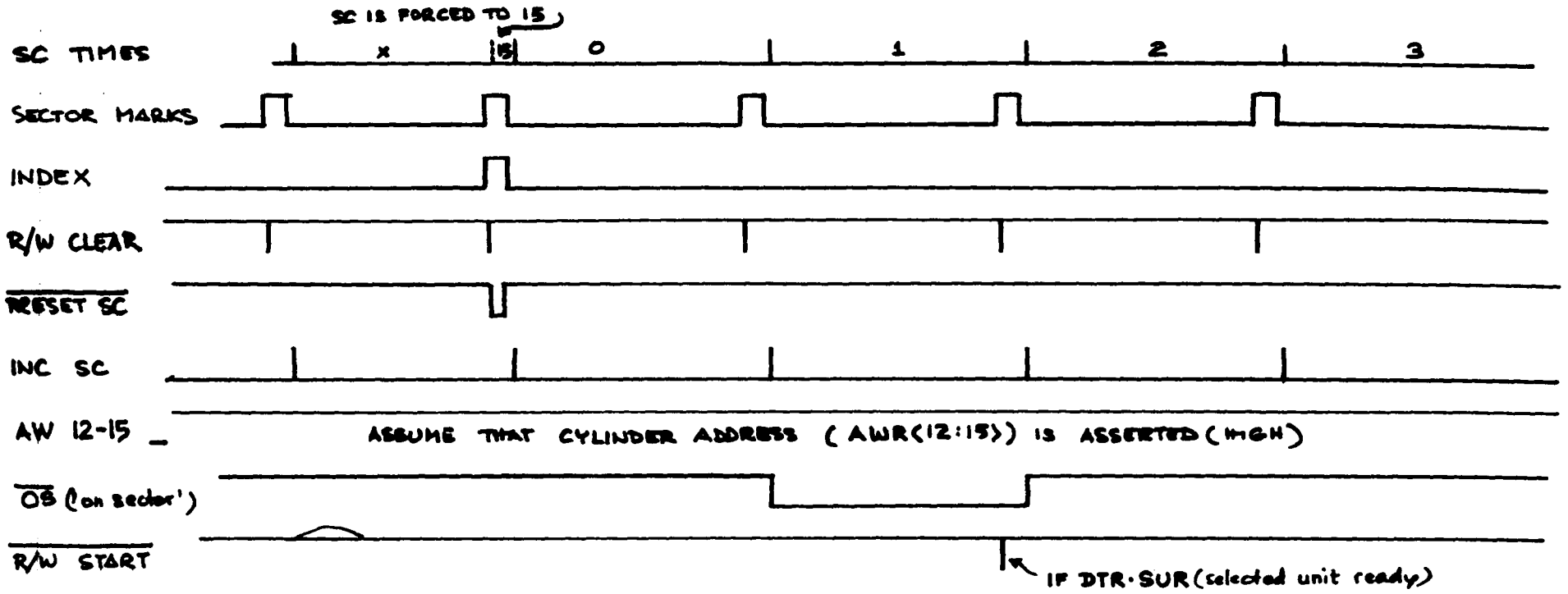


FIG. 4-7 SECTOR COUNTER & 'ON SECTOR' TIMING

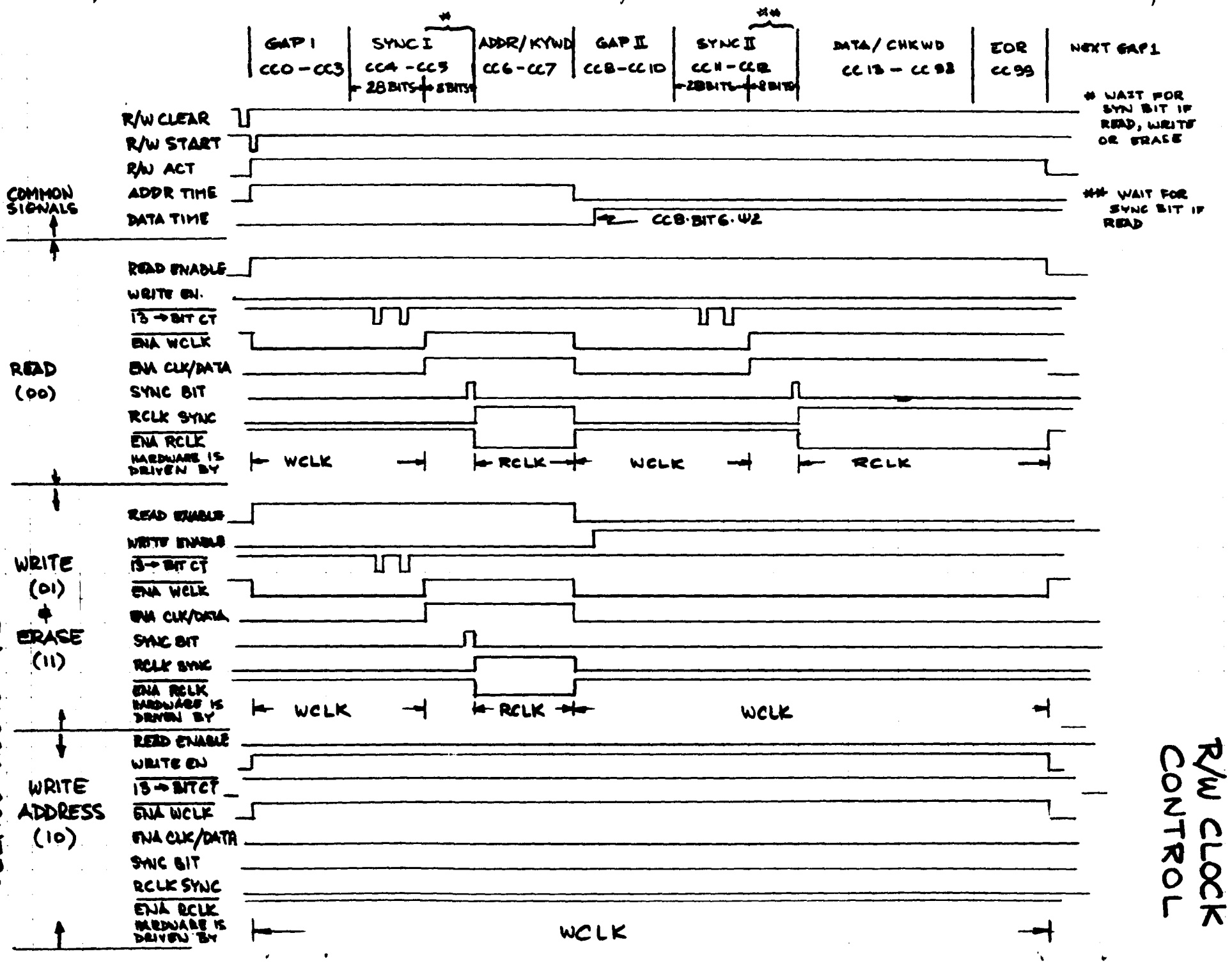


FIG. 4-B R/W CLOCK CONTROL

R/W CLOCK CONTROL

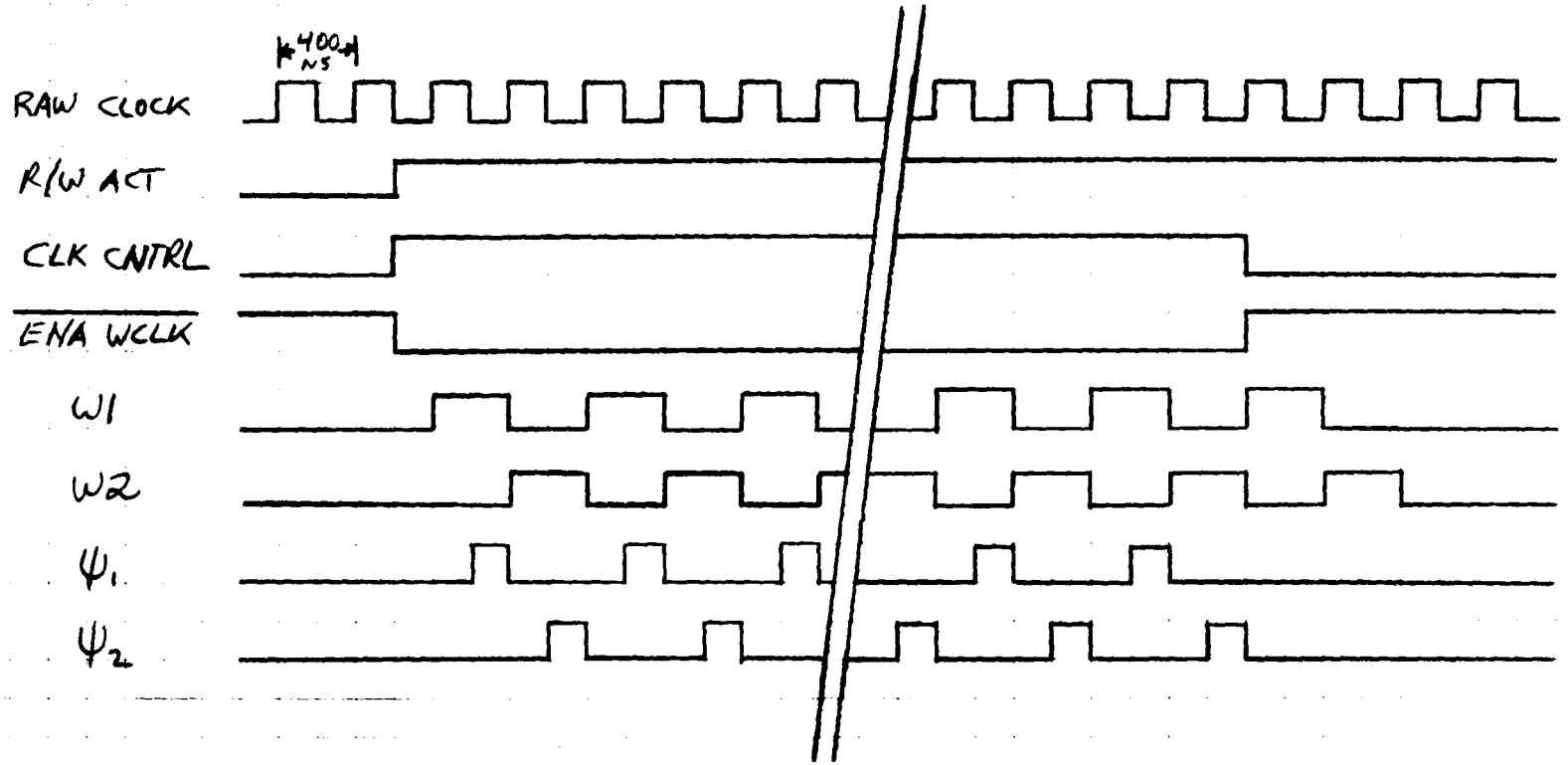


FIG4-9 WRITE CLOCK, DETAILED TIMING

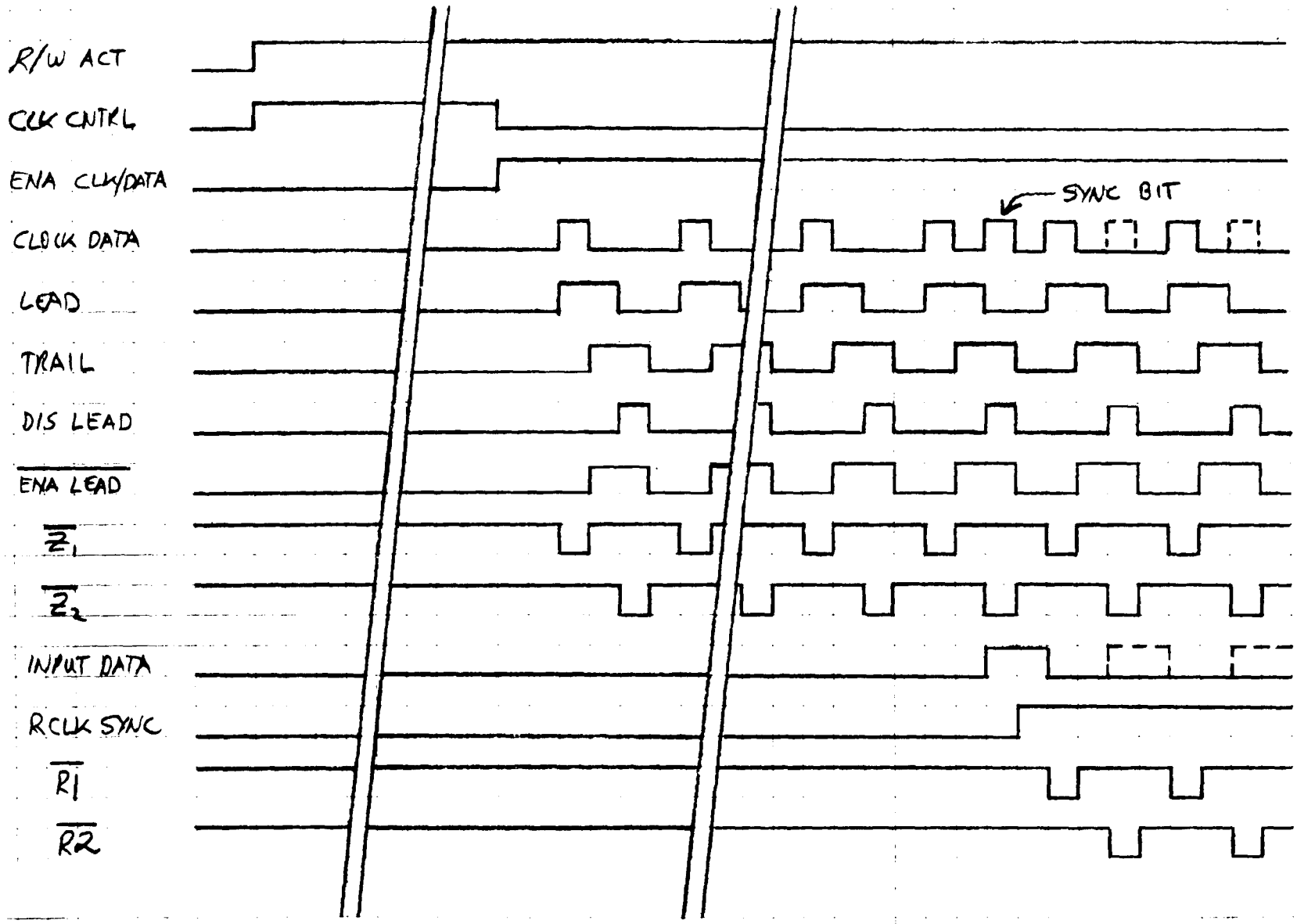


FIG. 4-10 READ CLOCK, DETAILED TIMING

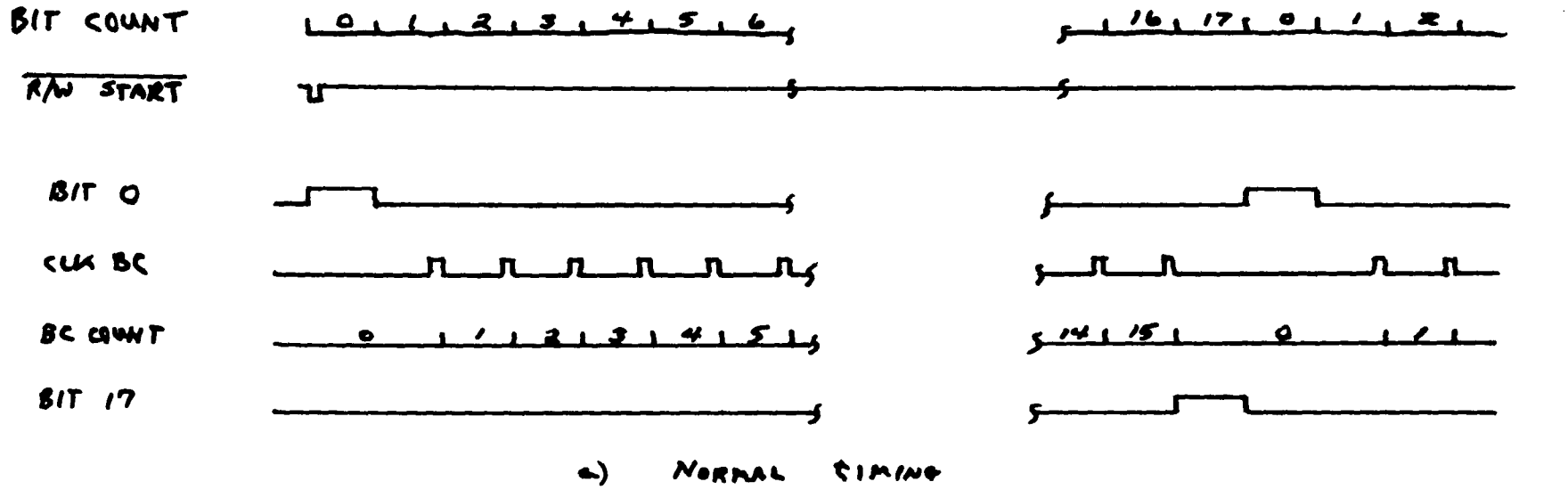


FIG. 4-11 BIT COUNTER TIMING REV 1

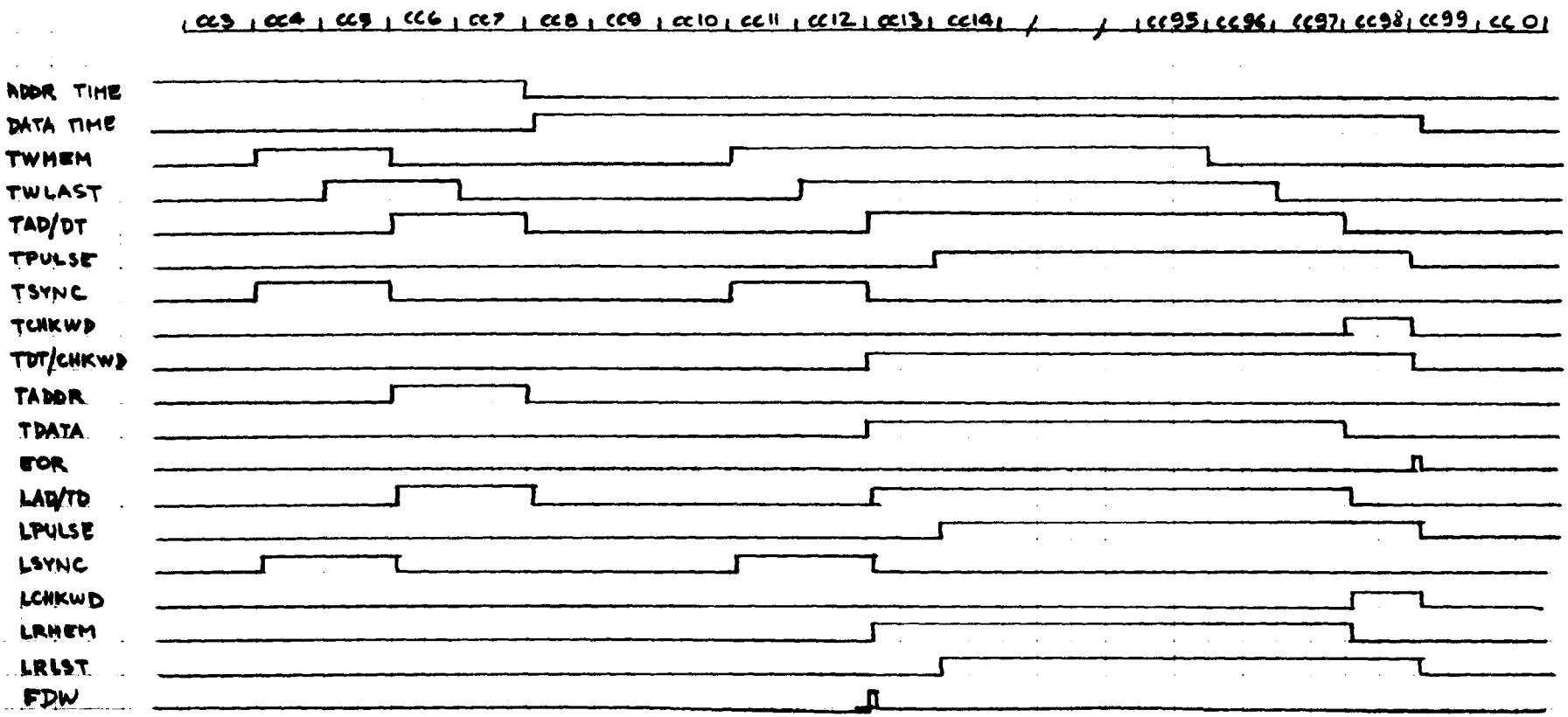


FIG. 4-12 READ/WRITE TIMING, B

REV. 1

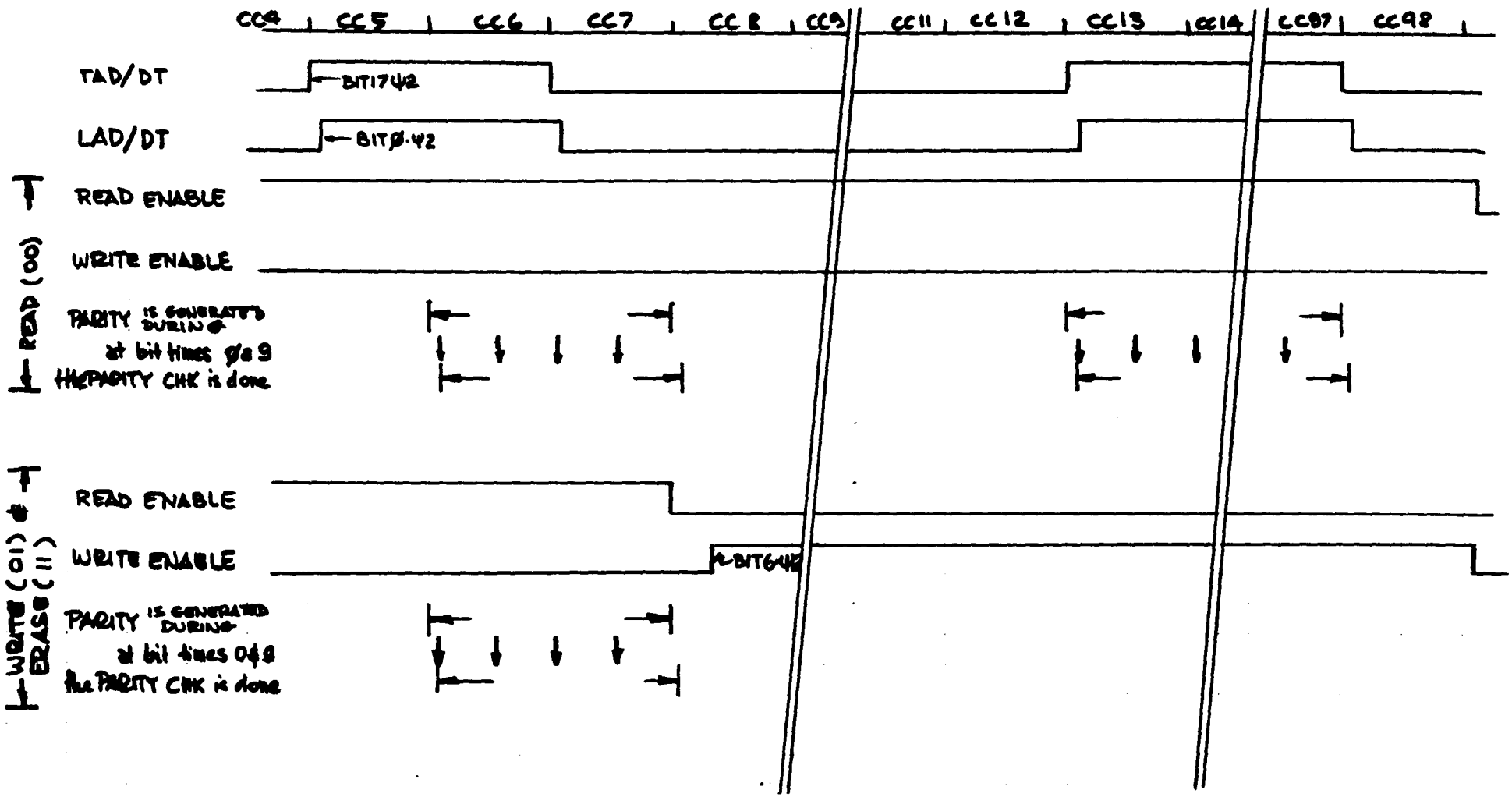


FIG.4-13 PARITY CHECK TIMING

A. DISK CONTROLLER INTERCONNECTIONS

A.1 INTRODUCTION

Appendix A contains the pin assignments and the signal names for all the cables that are used to interconnect the disk controller, the 853 disk, and the NOVA 1200.

Section A.2 contains the 4042 Board - Interface Board Interconnection Table which gives signal names, sources, destinations, and NOVA Backplane pin assignments of all signals going between the 4042 Board and the Interface Board (via the 48 unused pins on the NOVA backplane).

Section A.3 and A.4 contain the P1 cable signal names, sources, destinctions, and pin assignments. Cable P1 connects the 4042 Board to the I/O Box.

Section A.5 and A.6 contain the P2 cable signal names, sources, destinations, and pin assignments. Section A.7 and A.8 contain the P3 cable signal names, sources, destinations, and pin assignments. Cables P2 and P3 connect the Interface Board to the I/O Box.

Section A.9 contains the J100 cable signals names, pin assignments in the cable, and connection points within the I/O BOX. Section A.10 contains the J100 cable termination boards located in the 853 disk. Section A.11 contains the J102 cable signal names, pin assignments in the cable, and connection points within the I/O BOX. Section A.12 contains the J102 cable termination boards located in the 853 disk. Cables J100 and J102 connect the I/O BOX to the 853 disk.

4042 BOARD - INTERFACE CARD INTERCONNECTION TABLE

BACKPLANE PIN #	SIGNAL	4042 BOARD CONNECTIONS	INTERFACE BOARD CONNECTIONS
A47	AW1	10,D13/2(3-2),D4/2(3-47)	P1/2(3-22),03/4(3-22), Q2/2(3-25)
A49	AW2	9,D13/9(3-2),D4/5(3-47)	P1/4(3-22),03/7(3-22), Q2/5(3-25)
A57	AW3	8A,D13/5(3-2),D4/9(3-47)	03/11(3-22),P1/9(3-22), Q2/10(3-25)
A59	AW4	8,C13/12(3-2),D4/12(3-47)	04/16(3-22),P2/2(3-22), Q2/13(3-25)
A61	AW5	34A,C13/2(3-2),D3/2(3-47)	04/4(3-22),P2/5(3-22), Q3/2(3-25)
A63	AW7	29,C13/5(3-2),D3/9(3-47)	04/11(3-22),P2/12(3-22), Q3/10(3-25)
A65	AW6	7,C13/9(3-2),D3/5(3-47)	04/7(3-22),P2/9(3-22), Q3/5(3-25)
A67	AW8	41,B13/12(3-2),D2/2(3-47)	Q5/12(3-25)
A69	AW9	6A,B13/2(3-2),D2/6(3-47)	Q5/9(3-25)
A71	AW10	48,B13/9(3-2),D2/8(3-47)	Q5/4(3-25)
A73	AW11	49,B13/5(3-2),D2/13(3-47)	Q5/1(3-25)
A75	AW15	49A,A13/5(3-2),D1/13(3-47)	Q6/13(3-33)
A76	MODE 1	1,A6/10(3-3)	M1/4(3-6)
A77	AW14	6,A13/9(3-2),D1/9(3-47)	Q6/9(3-33)
A78	MODE 0	2,A6/9(3-3)	M1/3(3-6)
A79	AW13	57,A13/2(3-2),D1/6(3-47)	Q6/6(3-33)
A81	AW12	67,A13/12(3-2),D1/2(3-47)	Q6/1(3-33)
A83	$\overline{\text{HEAD}}$ OVER-FLOW	68A,E13/8(3-2)	N5/1(3-7)
A84	$\overline{\text{BFSET}}$	2A,E13/6(3-10)	L4/12(3-10)

A85	IORESET	68,E12/11(3-2),E11/12(3-5), E11/8(3-2), C8/5(3-3), C8/12(3-4)	N8/13(3-11),N8/12(3-11)
A86	$\overline{\text{INT ACK}}$	3,$\overline{\text{E2}}$/6(3-15)	08/3(3-12)
A87	$\overline{\text{21 CLEAR}}$	69,D8/12(3-13)	P8/2(3-12)
A88	21DOA	4,E10/10(3-13)	N2/11(3-11),N2/3(3-11), N1/11(3-11), N1/3(3-11)
A89	R/W CLEAR	70A,C1/9(3-50),C1/10(3-50), E11/11(3-5)	H6/11(3-33),J1/2(3-33), J1/1(3-33)
A90	21 DEV SEL	4A,E9/12(3-13),D8/2(3-13), D8/4(3-13)D8/11(3-13)	L8/4(3-13)
A91	FUNC 1	70,A6/16(3-3),E10/12(3-16)	F3/3(3-34),G3/5(3-35), G3/9(3-35),J1/3(3-49)
A92	FUNC 0	5,A6/15(3-3),E10/11(3-16)	F3/1(3-34),F8/6(3-34), I2/12(3-34)
B6	$\overline{\text{CYL INTR}}$	71,D4/3(3-47),D4/4(3-47),D4/10(3-47),D4/11(3-47),D3/3(3-47), D3/4(3-47), D3/10(3-47), RESISTOR AT A2/8	H2/11(3-49)
B11	$\overline{\text{HEAD}}$ INTR	72A,D2/3(3-47),D2/4(3-47), D2/10(3-47), D2/11(3-47), RESISTOR AT A2/9	H7/5(3-49)
B13	$\overline{\text{SECT}}$ INTR	72,D1/3(3-47),D1/4(3-47), D1/10(3-47),D1/11(3-47), RESISTOR AT A1/9	H7/9(3-49)
B15	$\overline{\text{KYWD}}$ INTR	76,E1/3(3-48),E1/4(3-48) E1/10(3-48),E1/11(3-48), E2/3(3-48),E2/4(3-48) E2/10(3-48),E2/11(3-48), RESISTOR AT A1/8	H2/13(3-49)
B19	READ ENABLE	83,D5/4(3-50),E5/5(3-36)	J7/8(3-35),J1/13(3-49), J1/10(3-49),H1/1(3-49), G5/10(3-50),P8/5(3-37), G6/5(3-51),I1/13(3-35), G5/2(3-36),F1/9(3-38)
B23	WRITE ENABLE	84A,D5/10(3-50)	J7/11(3-35),L8/9(3-35), L8/12(3-35),K6/5(3-43), G4/6(3-51)

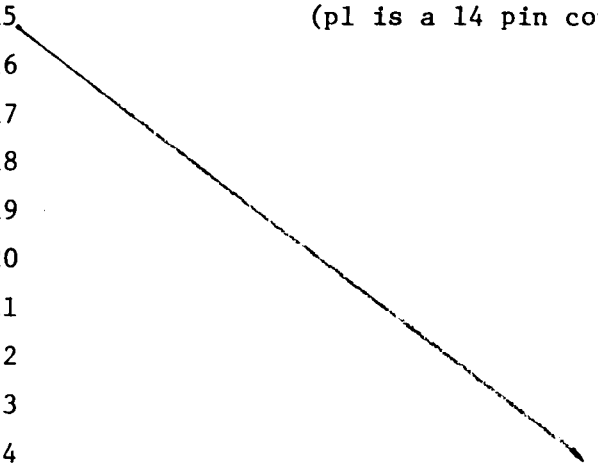
B25	TDT/ CKWD	89,D5/5(3-50),D5/9(3-50)	08/13(3-42)
B27	INPUT DATA	93,A1/2(3-46),A1/3(3-46), D5/3(3-50),C1/3(3-36), C1/4(3-36)	G5/4(3-36),G6/4(3-51)
B31	$\overline{Z_1}$	82,E4/12(3-36),C1/5(3-36)	F4/2(3-36)
B34	$\overline{Z_2}$	131,E4/6(3-36)	F4/5(3-36), F4/8(3-36)
B36	ENA CLK/ DATA	132A,E5/1(3-36),E5/4(3-36), E4/5(3-36),E4/1(3-36)	08/10(3-40),G5/1(3-36), H6/2(3-36)
B38	$\overline{\psi_2}$	132,A1/10(3-46),A2/10(3-46), A3/10(3-46),A4/10(3-46), A5/10(3-46),B1/10(3-50), B2/10(3-50),B3/10(3-50), B4/10(3-50),B5/10(3-50)	J2/11(3-37),06/11(3-38), G8/12(3-38),G8/6(3-38), G4/3(3-38),G1/9(3-38)
B40	$\overline{\text{LOAD SR}}$	133,A1/9(3-46),A2/9(3-46), A3/9(3-40),A4/9(3-46), A5/9(3-46)	Q8/12(3-17),K1/1(3-17)
B48	$\overline{\text{FDW}}$	134A,C4/9(3-2),C4/10(3-2)	H1/8(3-42)
B49	$\overline{\text{MEM REQ}}$	98A,46(3-16)	J8/11(3-17),P7/2(3-17)
B51	BITO.ψ_1	104,31(3-45),30A(3-45)	G8/10(3-38)
B52	$\overline{\text{2ODIC}}$	134,13(3-14)	N8/1(3-6),N8/2(3-6)
B53	DCH DONE	135,C1/11(3-16)	P7/1(3-17)
B54	OUTPUT DATA	136A,A5/11(3-46),D5/11(3-50)	F1/1(3-43)
B67	CKO	125,B5/14(3-50),C2/5(3-50)	I7/1(3-43)
B69	CK17	136,B1/15(3-50)	G5/9(3-50)

A.3 P1 CABLE TABLE - DG 4042 BOARD END

P1-x pin	DESTINATION (other terms follow comma)	SIGNAL NAME
1	C8/2	IO RESET
2	A8/2	$\overline{\text{1RK→2RK}}$
3	C8/2	$\overline{\text{CLR FULL}}$
4	E7/4	$\overline{\text{21 IOPLS}}$
5	E7/5	$\overline{\text{20 IOPLS}}$
6	between E8/8 & D8/14	$\overline{\text{20 START}}$
7	between E8/7 & D7/1	$\overline{\text{SET DONE}}$
8	between E8/1 & E9/14	GRND
9	N/C	----
10	between D6/8 & C8/14 (not used)	
11	D5/2 (not used)	
12	E5/2	READ DATA
13	between E8 and front of board	GND (white cond. of twisted pair)
14	between E5 and front of board	GND (black cond. of twisted pair)
15	(p1 is a 14 pin connector!)	
16		
17		
18		
19		
20		
21		
22		
23		
24		

A.4 P1 CABLE TABLE - I/O BOX END

P1-x pin	DESTINATION (other terms follow comma)	SIGNAL NAME
1	ID6/14	IO RESET
2	ID6/12	$\overline{\text{IRK→2RK}}$
3	ID6/10	$\overline{\text{CLR FULL}}$
4	ID6/8	$\overline{\text{21 IOPLS}}$
5	IE6/14	$\overline{\text{20 IOPLS}}$
6	IE6/12	$\overline{\text{20 START}}$
7	IE6/10	$\overline{\text{SET DONE}}$
8	IE6/8	GND
9	N/C	----
10	N/C	----
11	N/C	----
12	X8/13	READ DATA
13	RIGHT TOP MOUNTING BOLT OF ID, IE, IF BOARD	GND
14	RIGHT TOP MOUNTING BOLT OF ID, IE, IF BOARD	GND
15	(p1 is a 14 pin connector!)	
16		
17		
18		
19		
20		
21		
22		
23		
24		



A.5 P2 CABLE TABLE - DG INTERFACE BOARD END

P2-x pin	DESTINATION (other terms follow comma)	SIGNAL NAME
1	Ø5/10, Ø5/13	FWD/RVS→A/C forward reverse
2	L7/9	RFSET(1)
3	L2/11	DIFF
4	Q5/10, Q5/13, Q5/2, Q5/5	HEAD ADD→A/C
5	Ø6/3, Q3/12, P5/12, P5/9, P5/4, P5/1, 05/4, 05/1	DIFF→A/C
6	L7/13	DTR
7	K5/8	STOP R/W
8	N5/6	ZC
9	L7/10	$\overline{\text{RFSET(2)}}$
10	N/C	----
11	N/C	----
12	N/C	----
13	N/C	----
14	N/C	----
15	N/C	----
16	N/C	----
17	J7/3	CONT SELECT control set
18	K8/13 (not used)	ERROR OVERRIDE
19	Q3/9, Q3/4, Q3/1, Q2/12 Q2/9, Q/2/4, Q2/1	CYL ADD→A/C
20	Q2/4	A/C→CCA current cylinder addr.
21	K6/6	$\overline{\text{WRITE DATA}}$
22	L8/5	R/W BUSY
23	H7/1	RAW CLOCK
24	between K8 and L8	GRND

A.6 P2 CABLE TABLE - I/O BOX END

P2-x pin	DESTINATION		SIGNAL NAME
1	IE3/8		FWD/RVS→A/C—{forward / reverse}
2	IE3/10		$\overline{\text{RFSET(1)}}$
3	IE3/12		DIFF
4	IE3/14		HEAD ADD→A/C
5	ID3/8		DIFF→A/C
6	ID3/10		DTR
7	ID3/12		STOP R/W
8	ID3/14		ZC
9	ID4/14		$\overline{\text{RFSET(2)}}$
10	N/C		----
11	N/C		----
12	N/C		----
13	N/C		----
14	N/C		----
15	N/C		----
16	N/C		----
17	IE4/7		CONT SELECT—{control / sel}
18	IE4/5	(not in use)	ERROR OVERRIDE
19	IE4/3		CYL ADD→A/C—{cylinder / address}
20	IE4/1		A/C→CCA—{current / cyl. address}
21	ID4/7		$\overline{\text{WRITE DATA}}$
22	ID4/5		R/W BUSY
23	ID4/3		RAW CLOCK
24	ID4/1		GRND

A.7 P3 CABLE TABLE - D.G. INTERFACE BOARD END

P3-x pin	DESTINATION (other terms follow comma)	SIGNAL NAME
1	Q4/1	A/C ØR
2	Q4/2	A/C 1R
3	Q4/3	A/C 2R
4	Q4/4	A/C 3R
5	Q4/5	A/C 4R
6	Q4/6	A/C 5R
7	Q4/7	A/C 6R
8	Q4/14	$\overline{\text{A/C ØT}}$
9	Q4/13	$\overline{\text{A/C 1T}}$
10	Q4/12	$\overline{\text{A/C 2T}}$
11	N/C	----
12	N/C	----
13	Q4/11	$\overline{\text{A/C 3T}}$
14	Q4/10	$\overline{\text{A/C 4T}}$
15	Q4/9	$\overline{\text{A/C 5T}}$
16	Q4/8	$\overline{\text{A/C 6T}}$
17	P6/1	INDEX
18	K7/12	SECTOR MARKS
19	L7/2	SUR selected unit ready
20	M7/1	DPU diskpack unsafe
21	N4/9	SEEK ERROR
22	between K1 and L1	GND
23	N/C	----
24	N/C	----

A.8 P3 CABLE TABLE - I/O BOX END

P3-x pin	DESTINATION (other terms follow comma)	SIGNAL NAME
1	Y1/11, Y2/4	A/C ØR
2	Y1/10, Y2/9	A/C 1R
3	Y1/1, Y3/4	A/C 2R
4	Y1/14, Y3/9	A/C 3R
5	Y1/12, Y4/4	A/C 4R
6	Y1/4, Y4/9	A/C 5R
7	Y1/9, Y5/4	A/C 6R
8	Y1/2	$\overline{\text{A/C ØT}}$
9	Y1/3	$\overline{\text{A/C 1T}}$
10	Y1/13	$\overline{\text{A/C 2T}}$
11	N/C	----
12	N/C	----
13	Y7/5	$\overline{\text{A/C 3T}}$
14	Y7/2	$\overline{\text{A/C 4T}}$
15	Y1/5	$\overline{\text{A/C 5T}}$
16	Y7/7	$\overline{\text{A/C 6T}}$
17	Y7/4	INDEX
18	Y7/3	SECTOR MARKS
19	Y7/6, IE2/2	SUR selected unit ready
20	Y1/7, IE2/10	DUP diskpack unsafe
21	Y1/6	SEEK ERROR
22	Y1,Y14/7	GRND
23	N/C	----
24	N/C	----

A.9 CABLE J100 SIGNALS

J100 CABLE		WIRING WITHIN IO BOX			SIGNAL NAME
853 END	VO BOX END	COLOR CODE	POINT 1	CHIP PINS	
A1	1	ORG/BLK	Y14/13	Y2/2, Y8/13	A/C bit Ø
A2	2	BLK/GRN	Y14/14	Y2/1, Y8/12	
A3	3	GRN/JEL	Y14/11	Y2/11, Y8/8	A/C bit 1
A4	4	GRAY/BLK	Y14/12	Y2/12, Y8/3	
A5	5	WHT/GRAY	Y14/9	Y3/2, Y3/13	A/C bit 2
A6	6	BRN/BLK	Y14/10	Y3/1, Y3/12	
A7	7	BLU/WHT	Y15/7	Y3/11, Y3/8	A/C bit 3
A8	8	RED/GRN	Y14/8	Y3/12, Y9/9	
A9	9	YEL/GRAY	Y15/2	Y4/2, Y10/13	A/C bit 4
A10	10	BRN/JEL	Y15/1[x]	Y4/1, Y10/12	
B1	11	PUR/GRAY	Y15/4	Y4/11, Y10/8	A/C bit 5
B2	12	Yel/RD	Y15/3	Y4/12, Y10/9	
B3	13	BLK/GRAY	Y15/6	Y5/2, Y11/8	A/C bit 6
B4	14	RD/BRN	Y13/3	Y5/1, Y11/9	
C5	18	WHT/BLU	X7/13	X5/12	CYLINDER SELECT
C6	19	YEL/BLU	X7/14	X5/13	
C7	20	BLK/BLU	X7/11	X15/9	HEAD SELECT
C8	21	PUR/BLU	X7/12	X15/8	
C9	22	GRAY/RD	X7/9	X16/12	DIFFerence SELECT
C10	23	GRN/PUR	X7/10	X16/13	
D1	24	RD/BLU	X13/14	X16/9	CONTROL SELECT
D2	25	GRN/3RN	X7/8	X16/8	
D3	26	BLU/BRN	X13/8	X18/12	READ CYLinder SELECT
D4	27	RD/BRN	X13/9	X18/13	

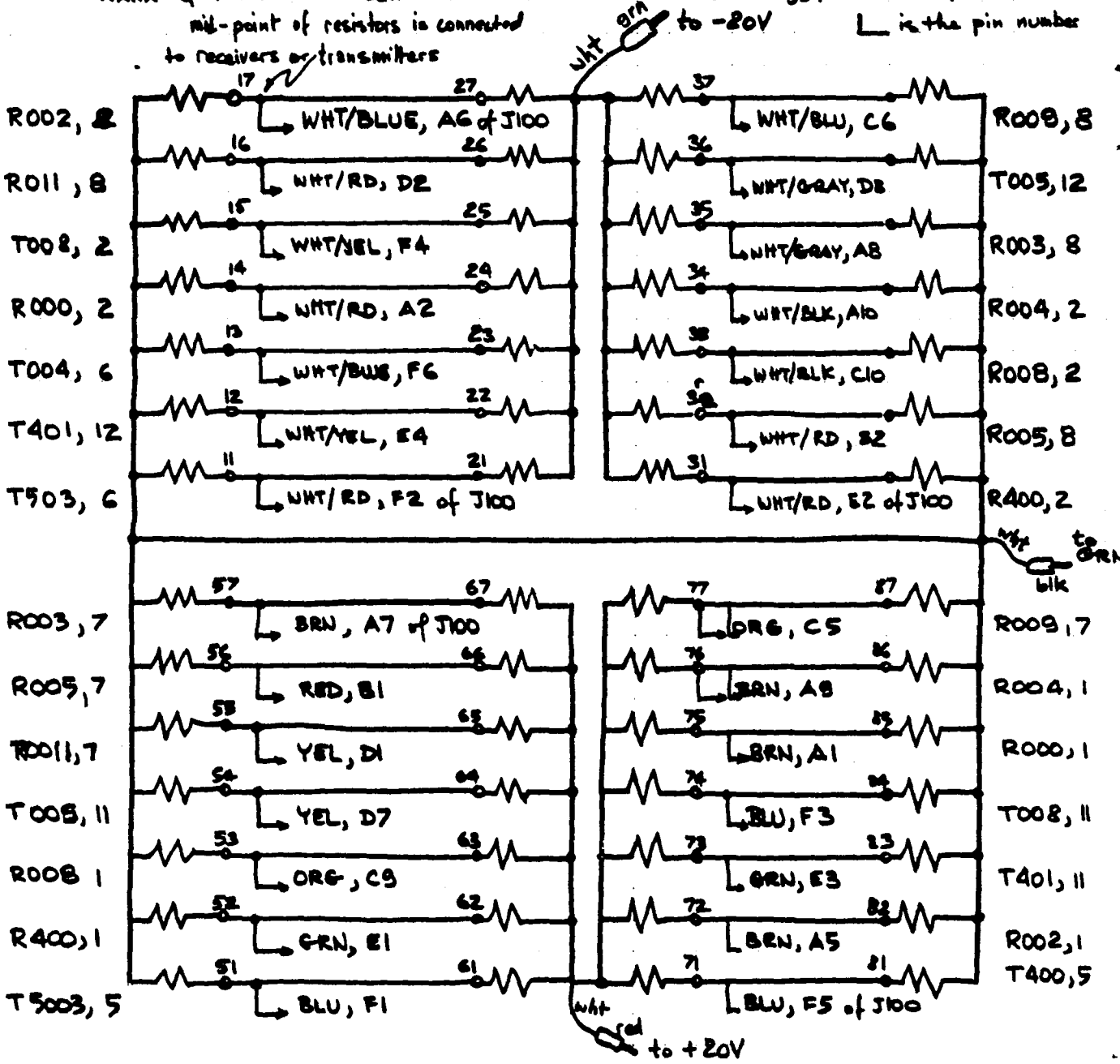
J100 CABLE		WIRING WITHIN IO BOX			SIGNAL NAME
853 END	VO BOX END	COLOR CODE	POINT 1	CHIP PINS	
E1	32	YEL/BRN	X13/12	X17/12	WRITE DATA
E2	33	BLK/BRN	X13/13	X17/13	
E3	34	BLU/PUR	X2/9	X12/2	READ DATA
E4	35	PUR/BRN	X2/10	X12/1	
E9	40	GRN/BLK	X1/13	X9/2	ON CYLINDER
E10	41	GRAY/WHT	X1/14	X9/1	
F3	44	ORG/WHT	X2/11	X11/11	SECTOR MARK
F4	45	PUR/GRAY	X2/12	X11/12	
F7	46	PUR/GRN	X2/8	X10/11	SEEK ERROR
F8	47	RD/PUR	X1/8	X10/12	
F5	48	GRAY/YEL	X2/13	X11/2	INDEX
F6	49	BLU/BLK	X2/14	X11/1	

A.10 J100 I/O CABLE TERMINATION (IN 853 CHASSIS)

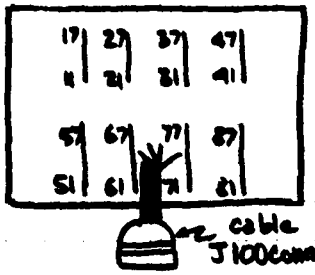
LARGE RESISTOR BOARD

Rxxx & Txxx are receivers or transmitters in 853 drawings. number after comma ⌊ is the pin number

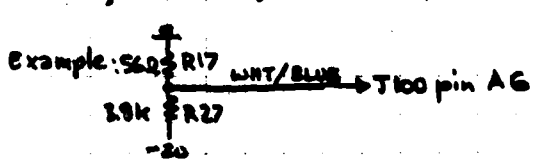
mid-point of resistors is connected to receivers or transmitters



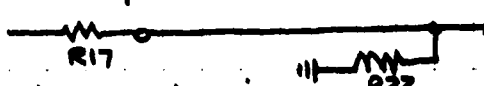
Position of resistors on PC board



Resistor values:
R11-R17, R41-R47, R51-R57, R81-R87 : 56Ω
R21-R27, R31-R37, R61-R67, R71-R77 : 3.9kΩ

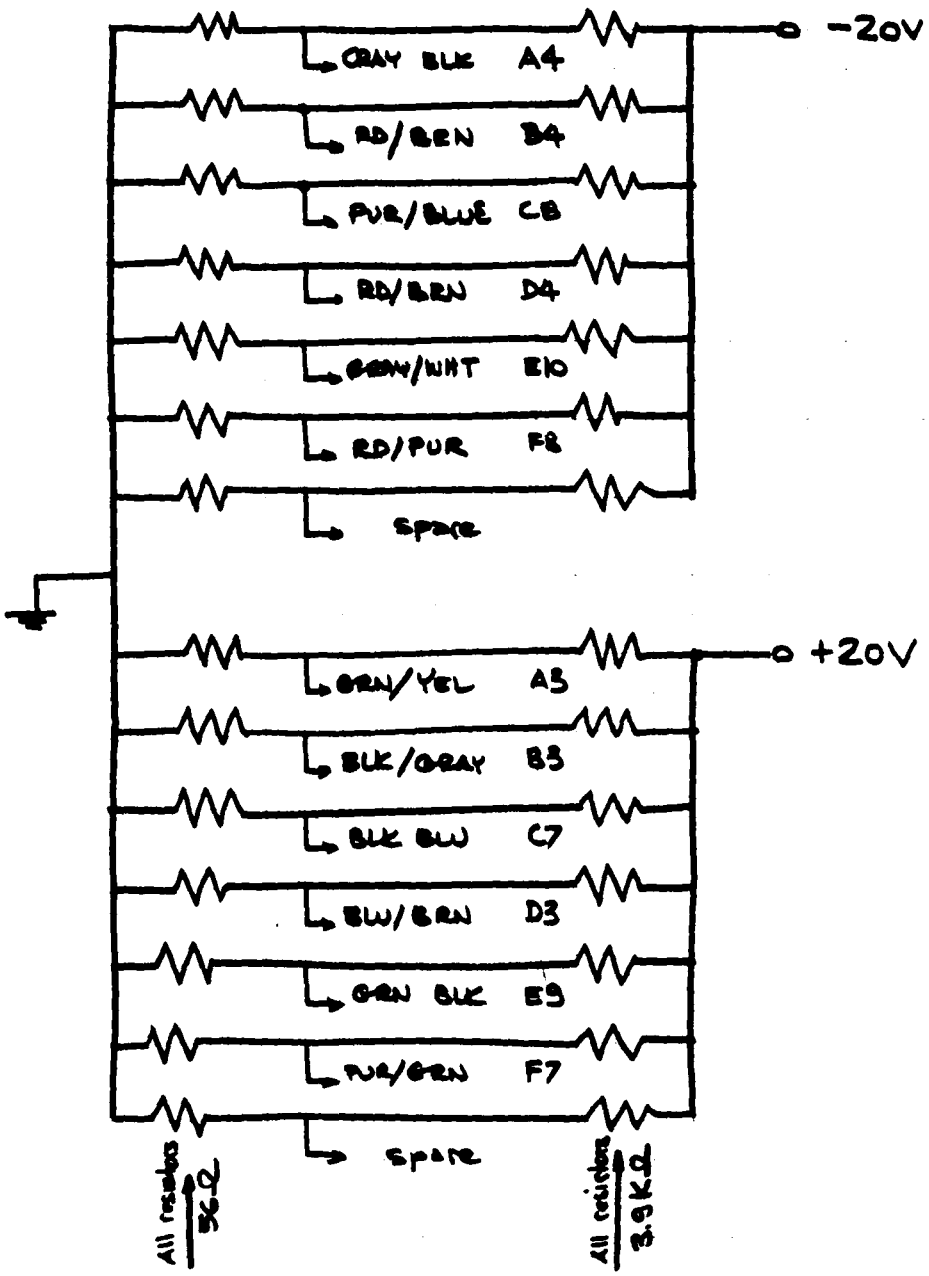


Note: The block of resistors R21-R27 is wired differently than shown:



The above drawing displays otherwise the physical layout of the card.

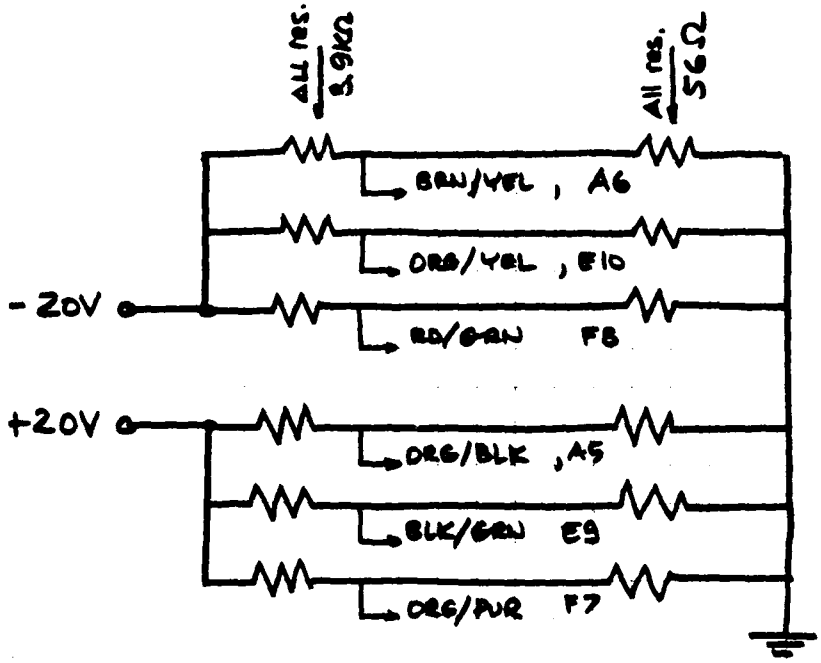
A.10 J100 I/O CABLE TERMINATION
SMALL RESISTOR BOARD



A.11 CABLE J102 SIGNALS

J102 CABLE		WIRING WITHIN I/O BOX			SIGNAL NAME
853 END	I/O BOX END	COLOR CODE	POINT I	GMP PINS	
A5	5	ORG/BLK	X13/11	X17/8	UNIT SELECT 2
A6	6	BRN/YEL	X13/10	X17/9	
E9	40	BLK/GRN	X1/9	X10/2	DISK PACK UNSAFE
E10	41	ORG/YEL	X1/10	X10 1	
F7	46	ORG/PUR	X1/11	X9/11	SELECTED UNIT READY
F8	47	RED/GRN	X1/12	X9/12	

A.12 J102 I/O CABLE TERMINATION



RESISTORS LOCATED ON SMALL P.C. BOARD AT TERMINATION OF CABLE ON 853 CHASSIS

B. DISK CONTROLLER COMPONENT ASSIGNMENTS

B.1 INTRODUCTION

Appendix B contains the IC chip assignments for the 4042 Board (Sect B.2), the Interface Board (Sect B.3), and the I/O BOX (Sect B.4). All chips used in the disk controller are TTL. FIG's B.1, B.2, and B.3 show the layout of the 4042 Board, INterface Board, and the I/O BOX respectively.

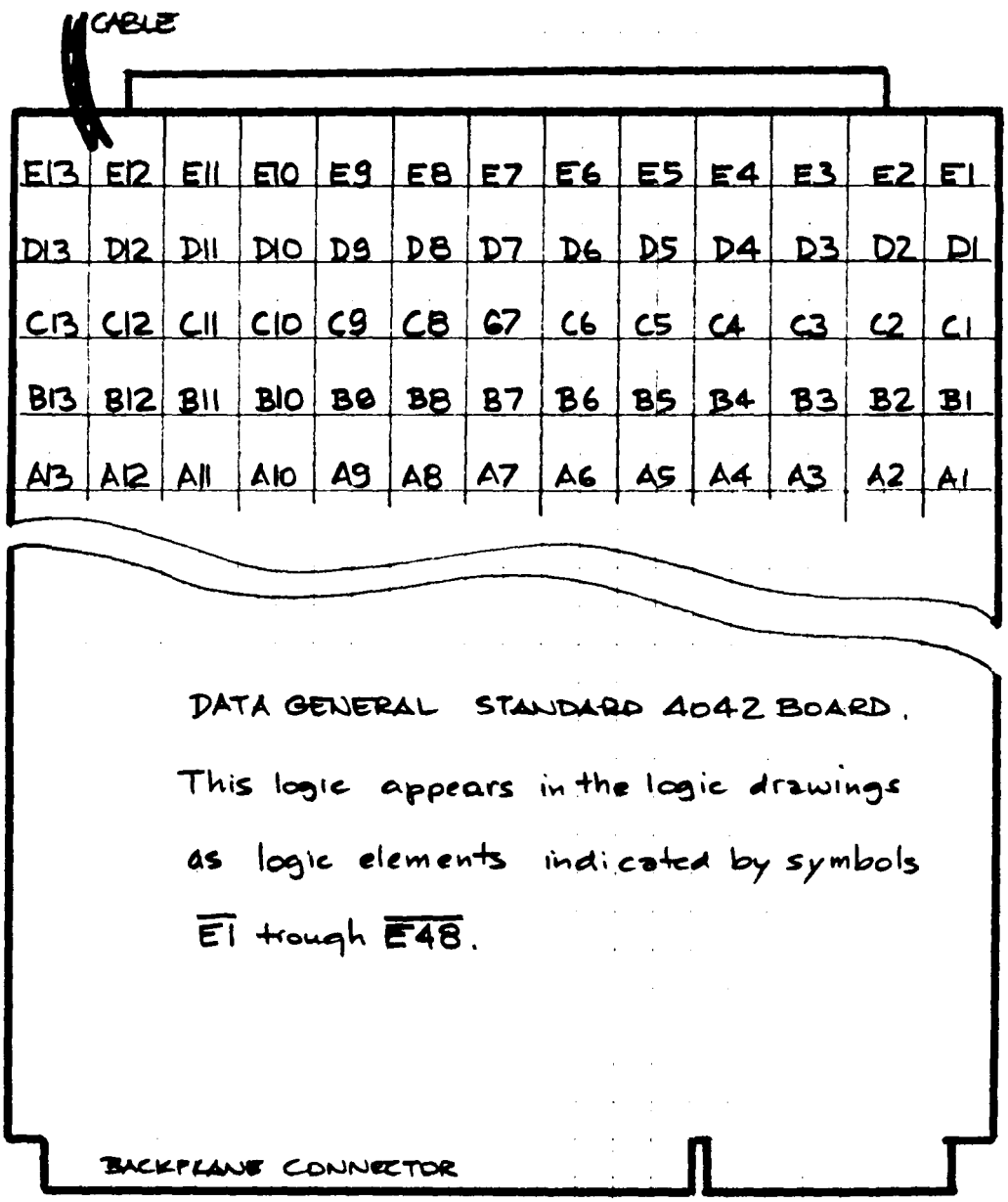


FIG. B-1 4042 BOARD LAYOUT

TOP VIEW

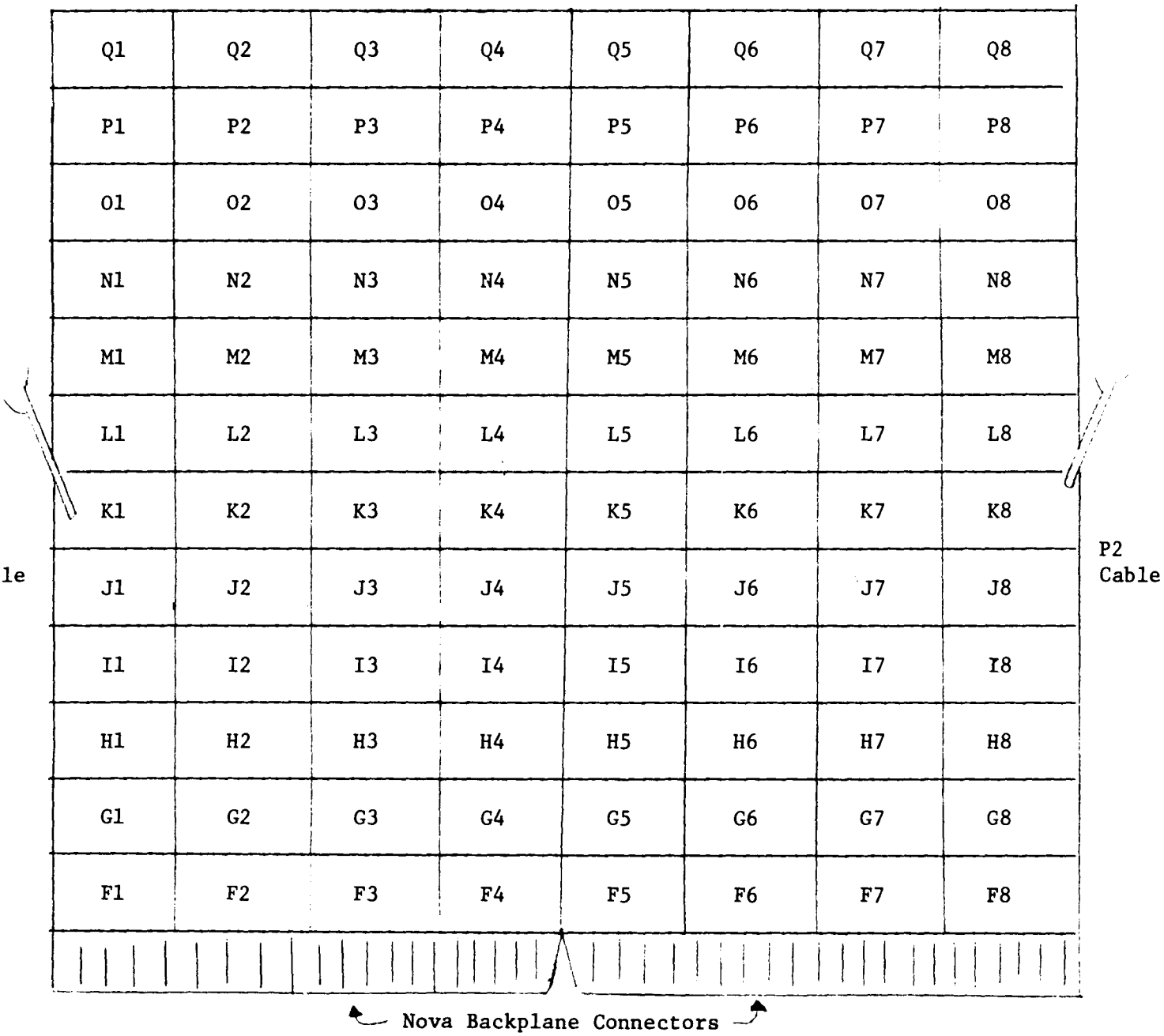
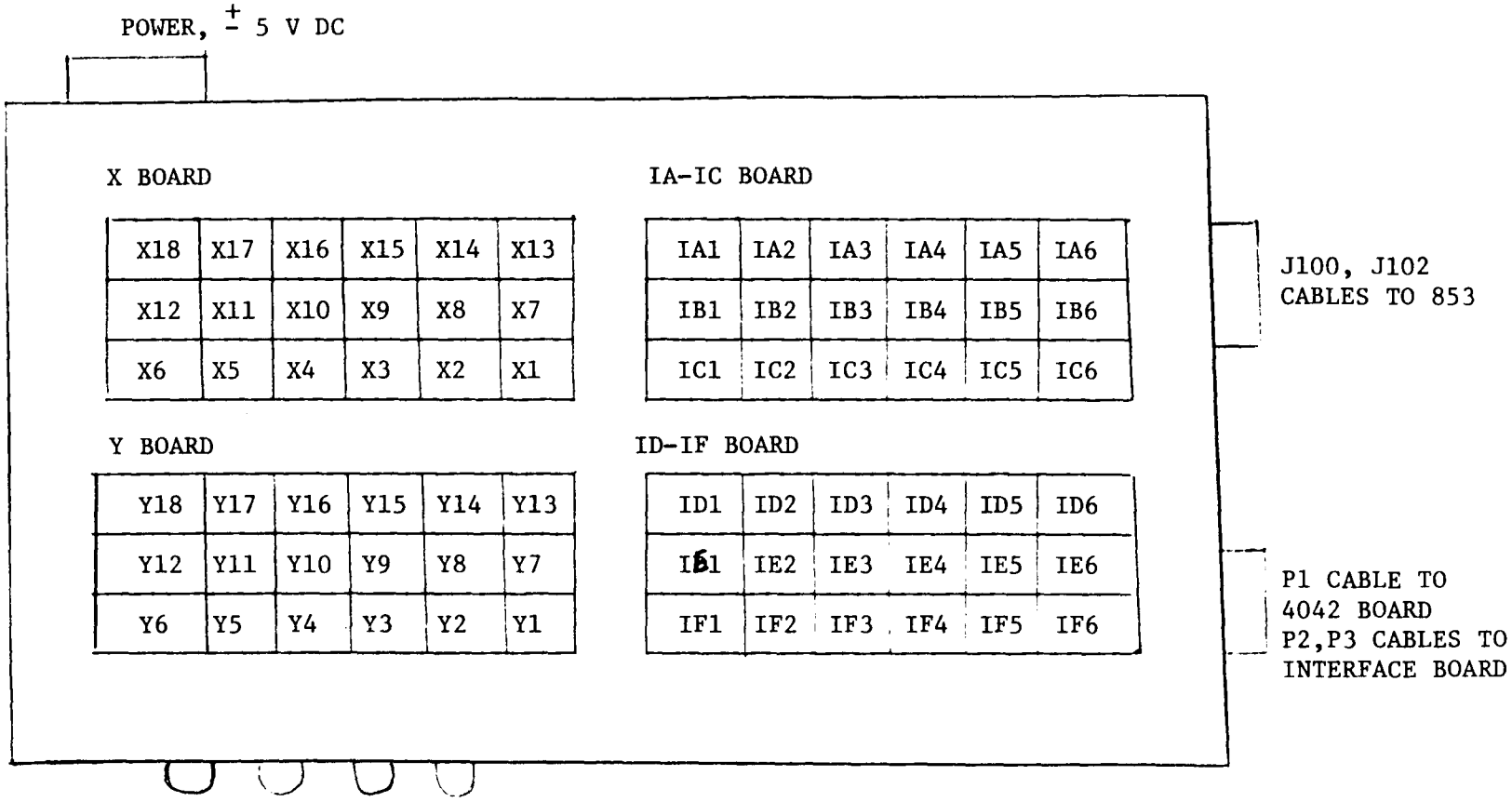


FIG. B.2 INTERFACE BOARD LAYOUT

TOP VIEW



I/O BOX INDICATORS

FIG B.3 I/O LAYOUT

B.2 4042 BOARD COMPONENT ASSIGNMENTS

The following IC's are located on the 4042 BOARD. See FIG B.1 for the 4042 BOARD layout.

LOCATION	COMPONENT	DESCRIPTION
A1	74195	4 bit shift register
A2	74195	4 bit shift register
A3	74195	4 bit shift register
A4	74195	4 bit shift register
A5	74195	4 bit shift register
A6	7475	Quad D Latch
A7	7474	Dual D Flip-Flop
A8	7402	Quad 2 input NOR
A9	7438	Quad 2 input OC NAND buffer
A10	8293	4 bit Binary Counter
A11	7475	Quad D Latch
A12	7438	Quad 2 input OC NAND buffer
A13	8293	4 bit Binary Counter
B1	74195	4 bit shift register
B2	74195	4 bit shift register
B3	74195	4 bit shift register
B4	74195	4 bit shift register
B5	74195	4 bit shift register
B6	7475	Quad D Latch
B7	7474	Dual D Flip-Flop
B8	7473	Dual J-K Flip-Flop
B9	7438	Quad 2 input OC NAND buffer
B10	8293	4 bit Binary Counter
B11	7475	Quad D Latch
B12	7438	Quad 2 input OC NAND buffer
B13	8292	4 bit Decade Counter
C1	7437	Quad 2 input NAND buffer
C2	7486	Quad 2 input Exclusive OR
C3	74180	8 bit Parity Generator
C4	7402	Quad 2 input NOR
C5	74180	8 bit Parity Generator

LOCATION	COMPONENT	DESCRIPTION
C6	7475	Quad D Latch
C7	7475	Quad D Latch
C8	7402	Quad 2 input NOR
C9	7438	Quad 2 input OC NAND buffer
C10	8293	4 bit Binary Counter
C11	7475	Quad D Latch
C12	7438	Quad 2 input OC NAND buffer
C13	8293	4 bit Binary Counter
D1	8242	Quad 2 input OC Exclusive NOR
D2	8242	Quad 2 input OC Exclusive NOR
D3	8242	Quad 2 input OC Exclusive NOR
D4	8242	Quad 2 input OC Exclusive NOR
D5	7410	Triple 3 input NAND
D6	74121	One Shot
D7	7475	Quad D Latch
D8	7410	Triple 3 input NAND
D9	7438	Quad 2 input OC NAND buffer
D10	8293	4 bit Binary Counter
D11	7475	Quad D Latch
D12	7438	Quad 2 input OC NAND buffer
D13	8293	4 bit Binary Counter
E1	8242	Quad 2 input OC Exclusive NOR
E2	8242	Quad 2 input OC Exclusive NOR
E3	7474	Dual D Flip flop
E4	7410	Triple 3 input NAND
E5	7413	Dual NAND Schmitt Trigger
E6	74121	One Shot
E7	7400	Quad 2 input NAND
E8	7430	8 input NAND
E9	7404	Hex Inverter
E10	7402	Quad 2 input NOR
E11	7402	Quad 2 input NOR
E12	7437	Quad 2 input NAND buffer
E13	7451	Dual 2 wide 2 input AND-OR-NOT

B.3 INTERFACE BOARD COMPONENT ASSIGNMENTS

The following IC's are located on the INTERFACE BOARD. See FIG B.2 for the INTERFACE BOARD layout.

LOCATION	COMPONENT	DESCRIPTION
F1	7410	Triple 3 input NAND
F2	----	
F3	7404	Nex Inverter
F4	7402	Quad 2 input NOR
F5	7476	Dual J-K Flip-Flop
F6	7402	Quad 2 input NOR
F7	7413	Dual NAND Schmitt Trigger
F8	7402	Quad 2 input NOR
G1	74175	Quad D Flip-flop
G2	7444	Excess 3 Gray-Decimal Decoder
G3	7400	Quad 2 input NAND
G4	7402	Quad 2 input NOR
G5	7420	Dual 4 input NAND
G6	7420	Dual 4 input NAND
G7	7474	Dual D Flip-flop
G8	7402	Quad 2 input NOR
H1	7410	Triple 3 input NAND
H2	7404	Hex Inverter
H3	74175	Quad D Flip-flop
H4	74175	Quad D Flip-flop
H5	7402	Quad 2 input NOR
H6	7400	Quad 2 input NAND
H7	7404	Hex Inverter
H8	7402	Quad 2 input NOR
I1	7438	Quad 2 input OC NAND buffer
I2	7400	Quad 2 input NAND
I3	8293A	4 BIT BINARY COUNTER
I4	8293A	4 BIT BINARY COUNTER
I5	7400	Quad 2 input NAND
I6	7437	Quad 2 input NAND Buffer
I7	7420	Dual 4 input NAND
I8	7402	Quad 2 input NOR

LOCATION	COMPONENT	DESCRIPTION
J1	7410	Triple 3 input NAND
J2	7437	Quad 2 input NAND Buffer
J3	7442	BCD - Decimal Decoder
J4	7442	BCD - Decimal Decoder
J5	8875A	Triple 3 input NOR
J6	8293A	4 BIT Binary Counter
J7	7437	Quad 2 input NAND buffer
J8	7400	Quad 2 input NAND
K1	7400	Quad 2 input NAND
K2	7402	Quad 2 input NOR
K3	7476	Dual J-K Flip-flop
K4	7476	Dual J-K Flip-flop
K5	7430	8 input NAND
K6	7440	Dual 4 input NAND buffer
K7	74175	Quad D Flip-flop
K8	7420	Dual 4 input NAND
L1	7400	Quad 2 input NAND
L2	7410	Triple 3 input NAND
L3	7400	Quad 2 input NAND
L4	7400	Quad 2 input NAND
L5	7400	Quad 2 input NAND
L6	7400	Quad 2 input NAND
L7	7410	Triple 3 input NAND
L8	7438	Quad 2 input OC NAND buffer
M1	7451	Dual 2 wide 2 input AND-OR-NOT
M2	7438	Quad 2 input OC NAND buffer
M3	7451	Dual 2 wide 2 input AND-OR-NOT
M4	7438	Quad 2 input NAND Buffer
M5	7451	Dual 2 wide 2 input AND-OR-NOT
M6	7438	Quad 2 input OC NAND buffer
M7	7451	Dual 2 wide 2 input AND-OR-NOT
M8	7438	Quad 2 input OC NAND buffer

LOCATION	COMPONENT	DESCRIPTION
N1	7474	Dual D flip-flop
N2	7474	Dual D flip-flop
N3	7430	8 input NAND
N4	7402	Quad 2 input NOR
N5	7404	Hex Inverter
N6	7474	Dual D Flip-flop
N7	7404	Hex Inverter
N8	7437	Quad 2 input NAND buffer
01	7475	Quad D Latch
02	7475	Quad D Latch
03	7483	4 bit adder
04	7483	4 bit adder
05	7438	Quad 2 input OC NAND buffer
06	7474	Dual D flip-flop
07	7438	Quad 2 input OC NAND buffer
08	7402	Quad 2 input NOR
P1	7486	Quad 2 input Exclusive OR
P2	7486	Quad 2 input Exclusive OR
P3	7486	Quad 2 input Exclusive OR
P4	7486	Quad 2 input Exclusive OR
P5	7438	Quad 2 input OC NAND buffer
P6	7410	Triple 3 input NAND
P7	7476	Dual J-K Flip flop
P8	7400	Quad 2 input NAND
Q1	7430	8 input NAND
Q2	7438	Quad 2 input OC NAND buffer
Q3	7438	Quad 2 input OC NAND buffer
Q4	----	P3 CABLE TERMINATION POINT
Q5	7438	Quad 2 input OC NAND buffer
Q6	8242A	Quad 2 input OC Exclusive NOR
Q7	8293A	4 bit Binary Counter
Q8	7410	Triple 3 input NAND

B.4 I/O BOX COMPONENT ASSIGNMENTS

The following IC's are located in the I/O Box. See FIG B.3 for the I/O BOX layout.

LOCATION	COMPONENT	DESCRIPTION
IA1	7473	Dual J-K Flip-Flop
IA2	7473	Dual J-K Flip-Flop
IA3	7473	Dual J-K Flip-Flop
IA4	7473	Dual J-K Flip-Flop
IA5	7473	Dual J-K Flip-Flop
IA6	7473	Dual J-K Flip-Flop
IB1	7402	Quad 2 input NOR
IB2	7410	Triple 3 input NAND
IB3	7438	Quad 2 input OC NAND buffer
IB4	7400	Quad 2 input NAND
IB5	7402	Quad 2 input NOR
IB6	7420	Dual 4 input NAND
IC1	7473	Dual J-K Flip-Flop
IC2	7437	Quad 2 input NAND buffer
IC3	7402	Quad 2 input NOR
IC4	7402	Quad 2 input NOR
IC5	EF114	Dual J-K Flip-Flop
IC6	EF114	Dual J-K Flip-Flop
ID1		
ID2		
ID3	----	P2 CABLE TERMINATION POINT
ID4	----	P2 CABLE TERMINATION POINT
ID5		
ID6	----	P1 CABLE TERMINATION POINT
IE1		
IE2		
IE3	----	P2 CABLE TERMINATION POINT
IE4	----	P2 CABLE TERMINATION POINT
IE5		
IE6	----	P1 CABLE TERMINATION POINT

LOCATION	COMPONENT	DESCRIPTION
1F1	7437	Quad 2 input NAND buffer
IF2	7400	Quad 2 input NAND
IF3	7473	Dual J-K Flip-Flop
IF4	7473	Dual J-K Flip-Flop
IF5	7402	Quad 2 input NOR
IF6	7438	Quad 2 input OC NAND buffer
X1	RESISTORS	J100, J102 CABLE TERMINATORS
X2	RESISTORS	J100 CABLE TERMINATORS
X3	7402	Quad 2 input NOR
X4	75113	Line Driver
X5	75113	Line Driver
X6	75113	Line Driver
X7	RESISTORS	J100, J102 CABLE TERMINATORS
X8	RESISTORS	Source of 3.5 VOLTS
X9	75107	Dual line Receiver
X10	75107	Dual line Receiver
X11	75107	Dual line Receiver
X12	75107	Dual line Receiver
X13	RESISTORS	J100 CABLE TERMINATORS
X14	----	+5v, -5v, -6v POWER DIST POINT
X15	75113	Line Driver
X16	75113	Line Driver
X17	75113	Line Driver
X18	75113	Line Driver
Y1	----	P3 CABLE TERMINATION POINT
Y2	75107	Dual Line Receiver
Y3	75107	Dual Line Receiver
Y4	75107	Dual Line Receiver
Y5	75107	Dual Line Receiver
Y6	75113	Line Driver
Y7	----	P3 CABLE TERMINATION POINT
Y8	7404	HEX INVERTER
Y9	7400	Quad 2 input NAND
Y10	75113	Line Driver
Y11	75113	Line Driver
Y12	75113	Line Driver
Y13	----	+5v, -5v, -6v POWER DIST POINT

LOCATION	COMPONENT	DESCRIPTION
Y14	RESISTORS	J100 CABLE TERMINATORS
Y15	RESISTORS	J100 CABLE TERMINATORS
Y16	75113	Line Driver
Y17	75113	Line Driver
Y18	75113	Line Driver