SIEMENS

# Geschäftsbereich Basisinformationssysteme

# KUNDENDIENST

## Basis-Datensystem 6.000

## TECHNICAL MANUAL 6.610

## VOLUME II

Best.-Nr. D45/50010

# DISKETTE CONTROLLER
# TDV 2114

## CONTENTS



TDV 2114
1426 · 10 · 76

# INTRODUCTION



TDV 2114
1428 - 10 - 76

# INTRODUCTION

## General

The diskette controller is a separate unit from the diskette storage drive and is the controlling interface between the drive and the main computer, Figure 1.

The system consists of a diskette storage drive which contains the drive mechanism, read/write heads, track positioning mechanism and the removable floppy disk (diskette). The associated control electronics is contained on a printed circuit board within the drive.

When the main computer requires to read data from or write data to a diskette it addresses the controller and transfers the instructions and commands for the particular mode. The controller then selects any one of four diskette drives and prepares the diskette to read or write. Once started, the controller logic is capable of transferring data between the diskette and the main computer memory independently of the main computer program.

The diskette controller logic is shown in Figure 2, and is divided up into four main functions:-

1) **Input/Output**
This logic controls the flow of data between the main computer and the controller and provides the initial indications to the internal processor that the main computer requires attention.

2) **Read**
This logic receives data from the diskette, synchronises it with the system clock, counts, checks and transfers it byte by byte to the processor.

3) **Write**
This logic receives data from the processor via the input/output logic and writes it byte by byte to the diskette.

4) **Processor (internal)**
This logic contains the fixed micro-program which controls and co-ordinates the reading and writing of data; initially sets up the diskette and transfers data and status to and from the input/output interface logic.

TDV 2114
1426 - 10 - 76

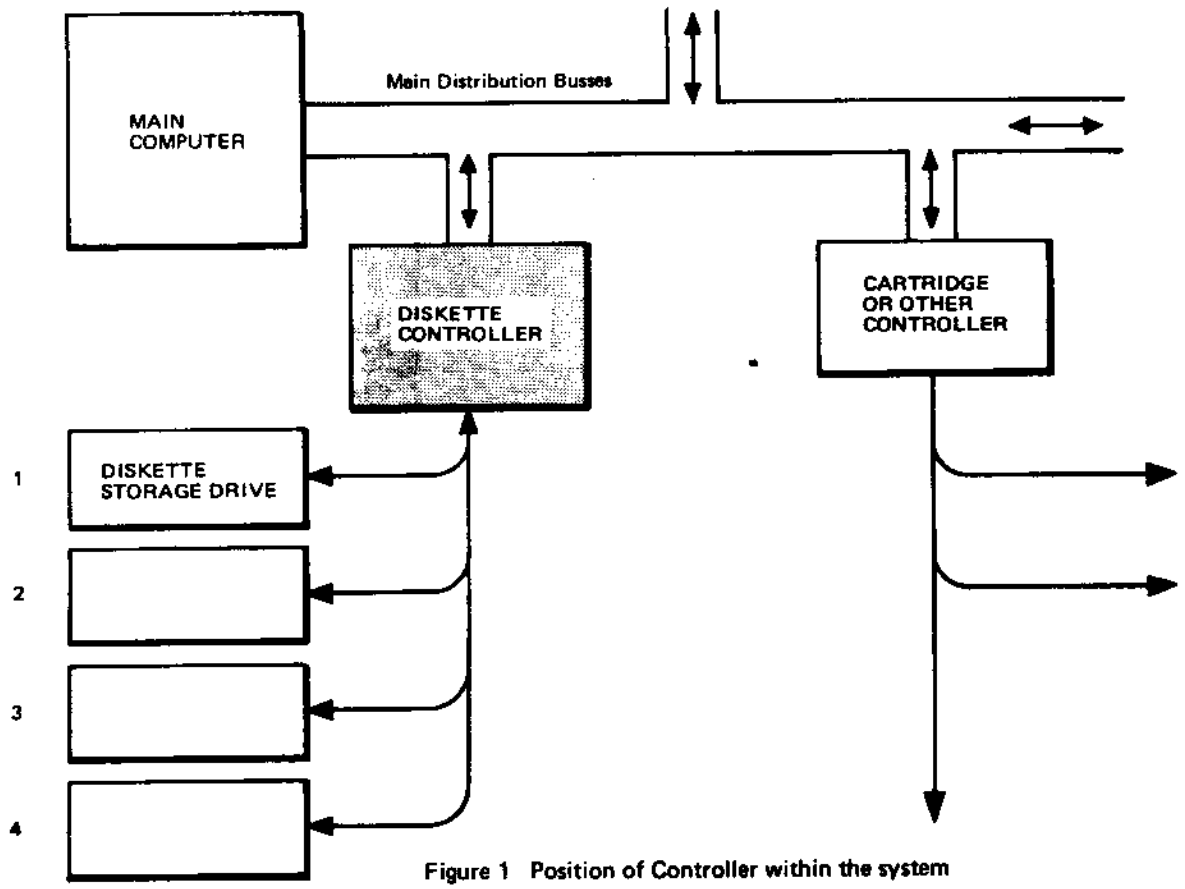



Figure 1 Position of Controller within the system

Data is checked after it is read from the diskette to detect whether it is from the correct sector and also to find out if an error has occurred. This checking function is described in the read and write mode logic.

The controller hardware logic is divided into three diagrams, each with an associated functional block diagram. The block diagrams are drawn to correspond with the logic layout to provide easy recognition between the diagram and the description.

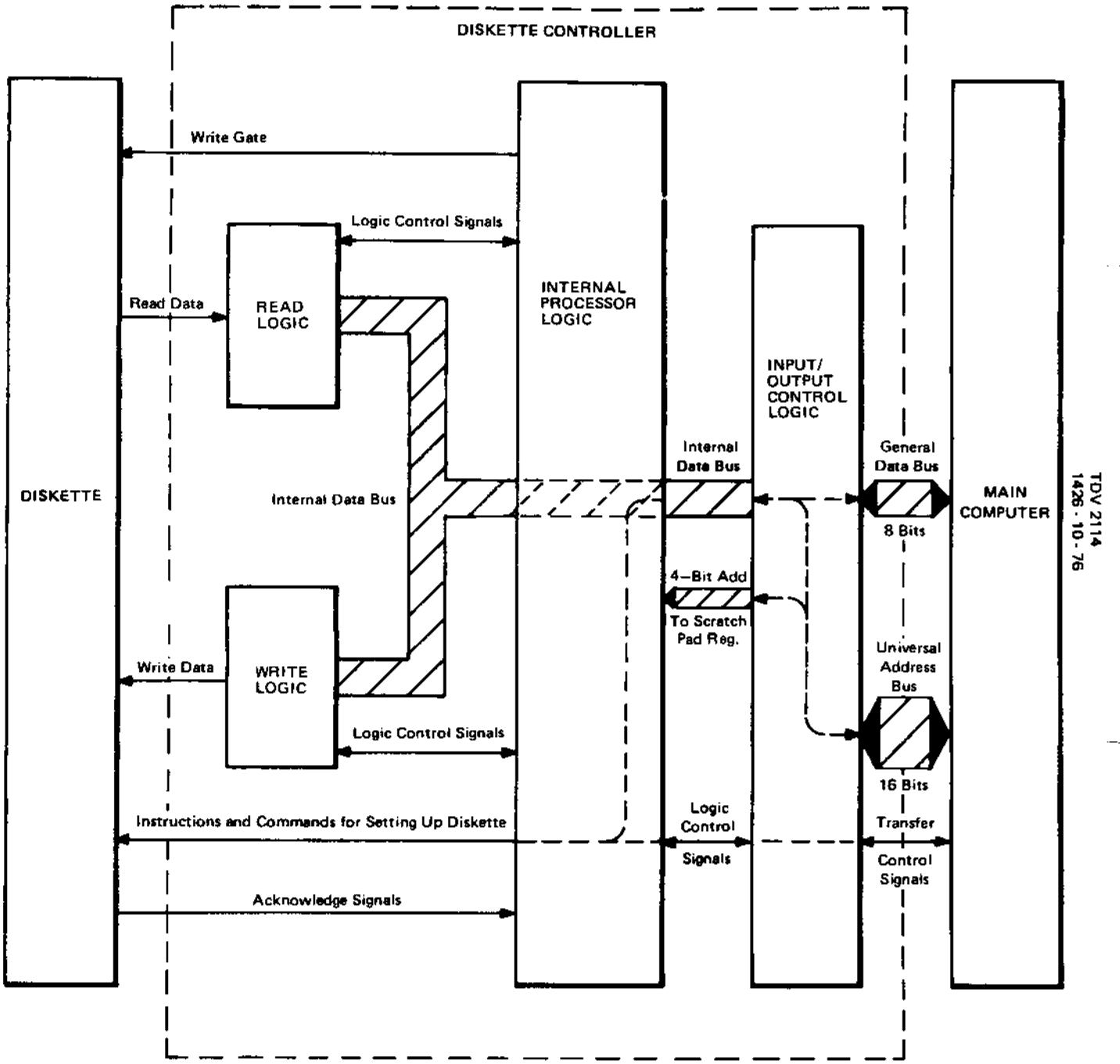



Figure 2 Block diagram of Controller

## Recording and Track Format

### Recording Format

The media characteristics, the logical data format, physical data format and the drive characteristics are compatible with the IBM 3740 Data Entry System and 3540 Diskette Input/Output Unit.

The diskette is initialised by writing each track from beginning to end without interruption. The beginning and end is coincident with the leading edge of the physical index hole. The drive uses the double frequency (2F) horizontal non return to zero (NRZ1) method of recording. Double frequency is the term given to the recording system which inserts a clock bit at the beginning of each bit cell time thereby doubling the frequency of recorded bits. The clock bit, as well as the data bit, are provided by the using system.

The presence of a flux transition represents a binary one and the absence of a transition represents a binary zero. The development of data bits is shown in Figure 3 and the timing between clock and data pulses is shown in Figure 4.

TDV 2114
1428 - 10 - 78

### Track Format

The diskette contains 77 tracks numbered from 00 to 76. The recording density varies from approximately 3200 bits per inch (6400 flux changes per inch) on the inside track to approximately 1700 bits per inch on the outside track.

The logical data format is comprised of an index track format, data track format, alternate and spare track format.

Each diskette is divided into one index track (number 00), 73 data tracks (number 01 to 73), two alternate tracks (numbers 74, 75) and one spare track (number 76).

Each track is divided into 26 sectors. The logical record length of a sector can be anywhere from 1 to 128 characters.

### Typical Track Index Format

Figure 5 shows a track Format, which is IBM compatible, using Index Recording Format with soft sectoring.

### Soft Sector Recording Format

In this Format, the using system may record one long record or several smaller records. Each track is started by a physical index pulse and then each record is preceded by a unique recorded identifier. This type of recording is called soft sectoring.

### Gaps

Each field on a track is separated from adjacent fields by a number of bytes containing no data bits. These areas are referred to as gaps and are provided to allow the updating of one field without affecting adjacent fields. As can be seen from Figure 5, there are four different types of gaps on each track.

### Gap 1 Post-Index Gap

This gap is defined as the 32 bytes between Index Address Mark and the ID Address Mark for Sector one (excluding the address mark bytes). This gap is always 32 bytes in length and is not affected by any updating process.

### Gap 2 ID Gap

The seventeen bytes between the ID Field and the Data Field is defined as Gap 2 (ID Gap). This gap may vary in size slightly after the Data Field has been updated.

### Gap 3 Data Gap

The thirty-three bytes between the Data Field and the next ID Field is defined as Gap 3 (Data Gap). As with the ID Gap, the Data Gap may vary slightly in length after the adjacent Data Field has been updated.

### Gap 4 Pre-Index Gap

The three hundred and twenty bytes between the last Data Field on a track and the Index Address Mark is defined as Gap 4 (Pre-Index Gap). Initially, this gap is nominally 320 bytes in length; however, due to write frequency tolerances and disk speed tolerances this gap may vary slightly in length. Also, after the data field of record 26 has been updated, this gap may again change slightly in length.

### Address Marks

Address Marks are unique bit patterns one byte in length which are used in this typical recording format to identify the beginning of ID and Data Fields and to synchronise the deserialising circuitry with the first byte of each field. Address Mark bytes are unique from all other data bytes in that certain bit cells do not contain a clock bit (all other data bytes have clock bits in every bit cell). There are four different types of Address Marks used. Each of these are used to identify different types of fields.

### Index Address Mark

The Index Address Mark is located at the beginning of each track and is a fixed number of bytes in front of the first record. The bit configuration for the Index Address Mark is Hex. FCD 7.

### ID Address Mark

The ID Address Mark byte is located at the beginning of each ID Field on the diskette. The bit configuration for this Address Mark is shown in Figure 6 (Hex. FEC7).

### Data Address Mark

The Data Address Mark byte is located at the beginning of each nondeleted Data Field on the diskette. The bit configuration for this Address Mark is Hex. FBC7.

### Deleted Data Address Mark

The Deleted Data Address Mark byte is located at the beginning of each deleted Data Field on the diskette. The bit configuration for this Address Mark is Hex. F8C7.

### CRC

Each field written on the diskette is appended with two Cyclic Redundancy Check (CRC) bytes. These two CRC bytes are generated from a cyclic permutation of the data bits starting with bit zero of the address mark and ending with bit seven of the last byte within a field (excluding the CRC bytes). When a field is read back from a diskette, the data bits (from bit zero of the address mark to bit seven of the second CRC byte) are divided by the same generator polynomial. A non-zero remainder indicates an error within the data read back from the drive while a remainder of zero indicates the data has been read back correctly from the diskette.

### Software

The internal processor contains a fixed programmed read only memory which controls all diskette operations. It is accessed automatically after the main computer program requests attention.

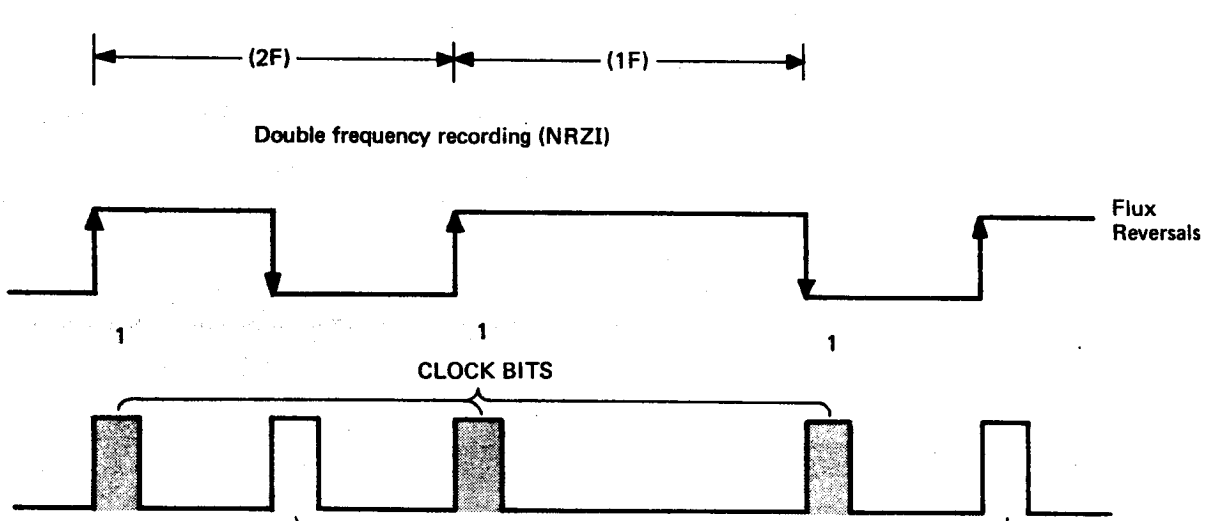
Certain decoding functions are electrically preprogrammed, therefore, it is essential that all software and decoder tables are available for the complete understanding of the controller hardware actions.

The following software information is contained in the software section of this manual.

1) Input/Output Instructions and commands
2) Controller micro-program instructions
3) Controller program flow charts
4) Controller program codes.

TDV 2114
1426 - 10 - 76

**Figure 3 Recording Format**



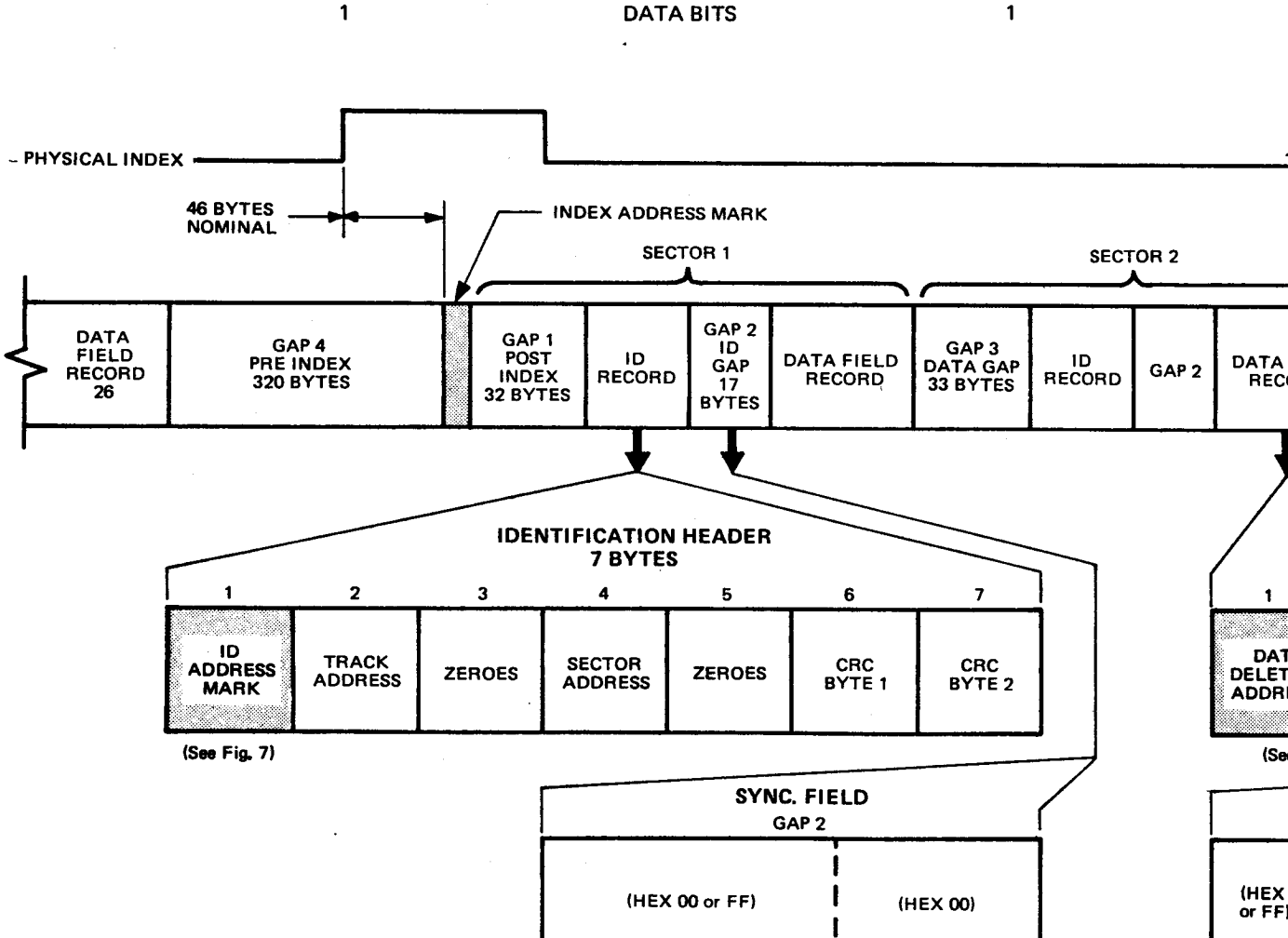
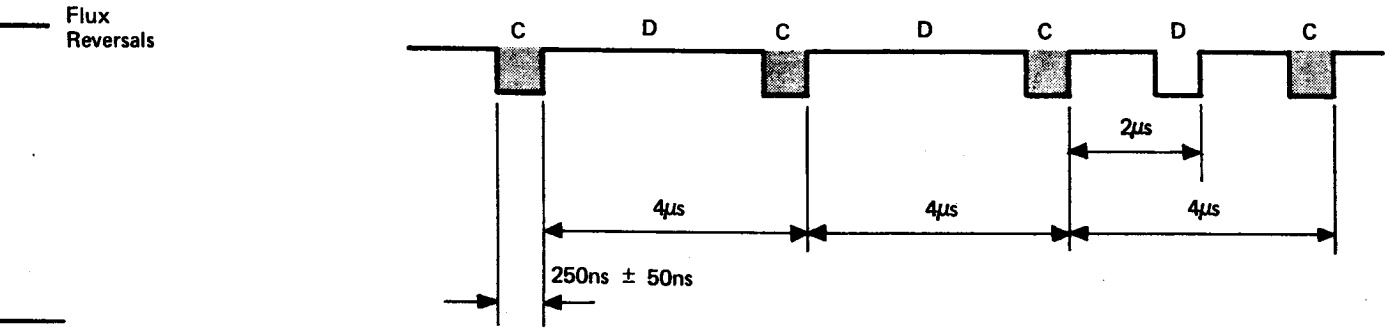






**Figure 5 Track Form**

A4

**Figure 4 Clock and Data Timing**



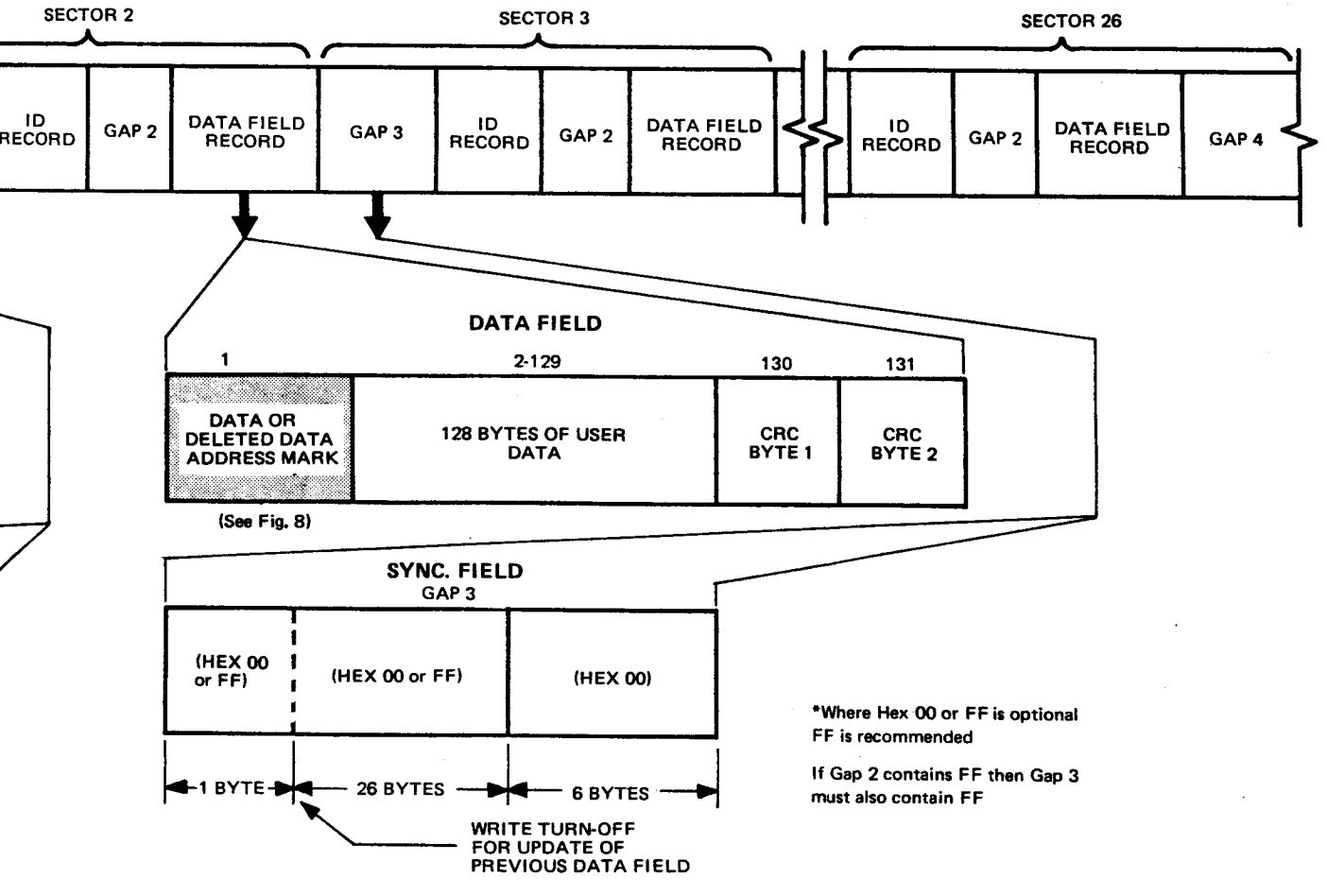
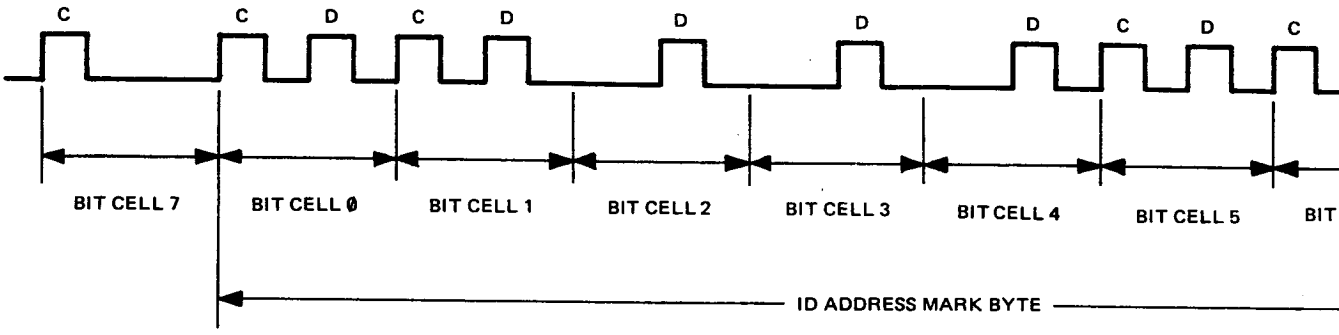
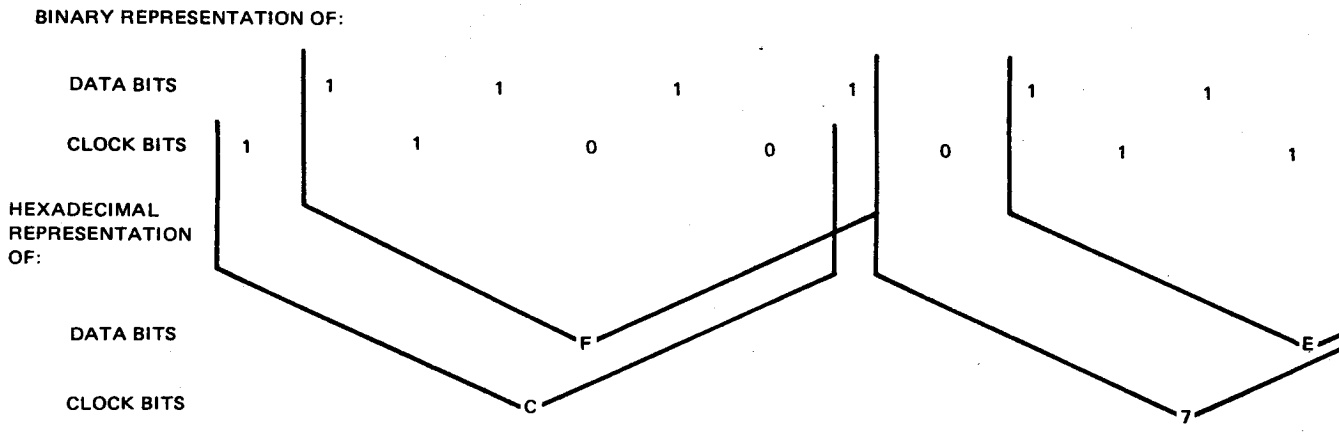



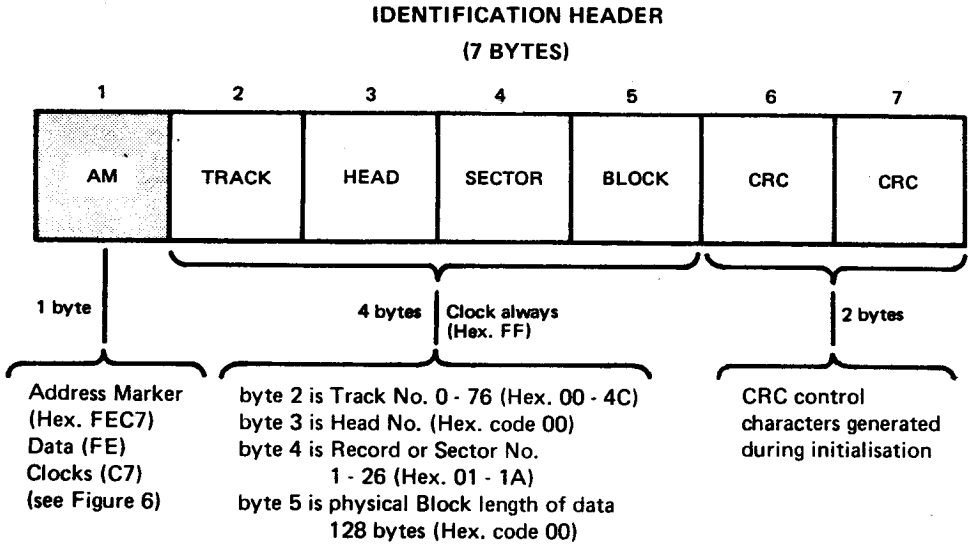

**Figure 5 Track Format**

A4

**Figure 6 ID Address Mark**



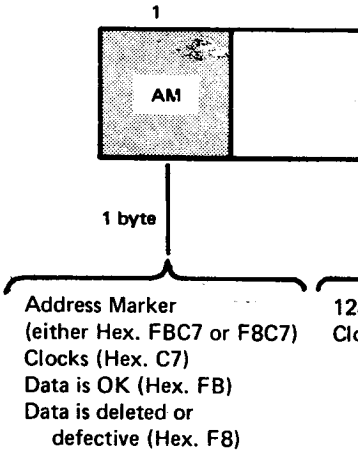
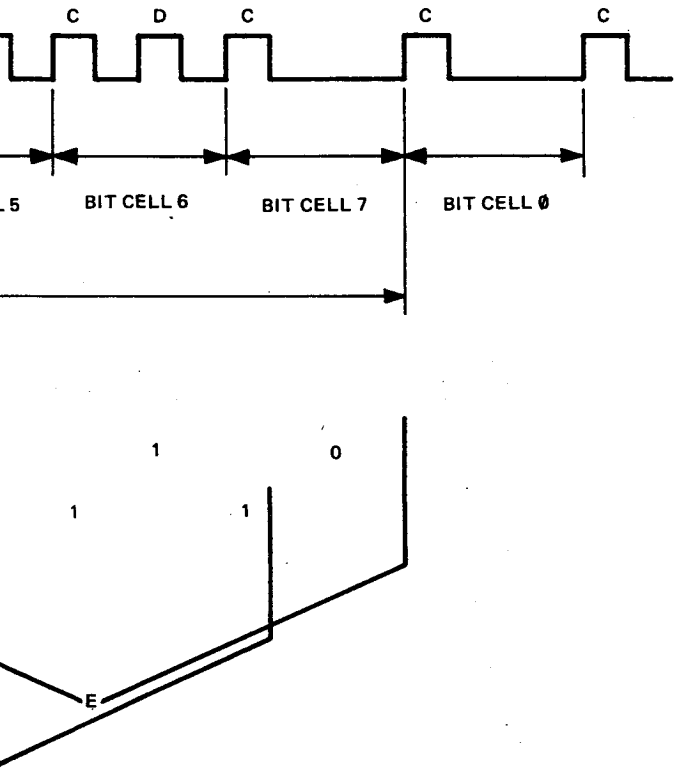




**Figure 7**



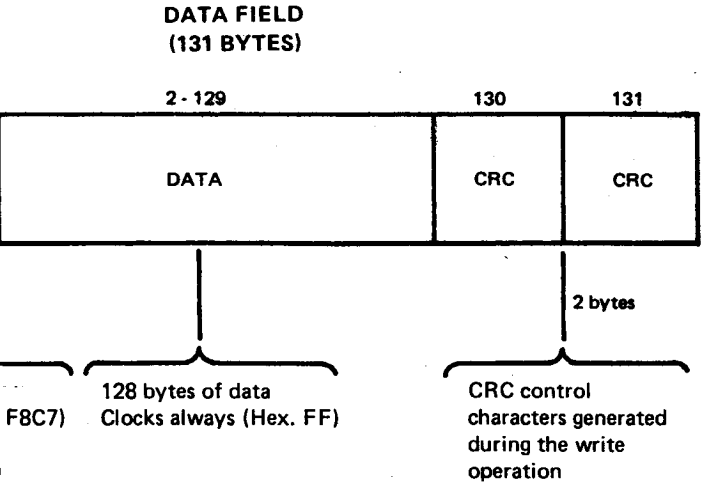




**Figure 8**

## Signal Definitions

A mnemonic is an abbreviation for the signal name, for example DATEN = Data Enable.

The signal name states the purpose or function of the signal.

The signal is said to be **true** or **false** depending on whether the statement in the signal name is true or false.

When the signal is **true** its logic value is 1.

When the signal is **false** its logic value is 0.

A bar above the mnemonic (e.g. $\overline{\text{BOUTE}}$) means an inversion of the logic value.

Example:

| | Logic Value | |
|---|---|---|
| **Signal state** | **BOUTE** | **$\overline{\text{BOUTE}}$** |
| True | 1 | 0 |
| False | 0 | 1 |

Inside the controller the logic signal levels are

| 0 to + 0.7 volts | + 2.5 to 5 volts |
|---|---|
| Logic 0 | Logic 1 |

## Other Connection Details

To avoid the circuit diagram being crowded with lines, many interconnections are merely indicated by the signal mnemonic and an arrow.

Example: ⟶▷ DATEN — To another place on the same diagram.

DATEN ▷⟶ — From another place on the same diagram.

⟶◇ — To or from another place on another diagram, but on the same board.

TDV 2114
1426 - 10 - 76

## Signal Tables

| Mnemonic | Signal name | Source | Destination/s | Effect of signal |
|---|---|---|---|---|
| ABS. M.B. | Absolute most sign. bit | Registers | – | Indicates most signification bit position |
| A = B1 | Input B1 (bit 0 - 3) is equal with A1 (bit 0 - 3) | ALU register U23 pin 14 | U32 pin 13 | Produces jump instruction JA = B |
| A = B2 | Input B2 (bit 4 - 7) is equal with A2 (bit 4 - 7) | ALU register U37 pin 14 | U32 pin 13 | Compares Acc. bits with internal data bus Produces jump instruction JA = B |
| AHREN | Address high register enable | Output decoder U39 pin 14 | U3 pin 11 | Clocks in address to high reg. from internal data bus for DMA transfers |
| AKBFE | Accumulator buffer enable | Inst. decoder U33 pin 9 | U25 pin 1 | Enable Acc. buffer |
| AKKEN | Accumulator enable | Inst. decoder U33 pin 6 | U27 pin 12 | Enables Acc. |
| ALREN | Address low register enable | Output decoder U39 pin 13 | U4 pin 11 | Clocks in address to low reg. from internal data bus for DMA transfers |
| BCOUNT | Block count | Write byte counter U54 pin 15 | U63 pin 9 | Enables JCODD after every byte. counted up by 16 REGCL pulses |
| BOUTE | Buffer output enable | Oscillator U76 pin 7 | U25 pin 2 | To ensure that the logic enables Acc. buffer in phase with system clock and also provide a 1:1 pulse width to give the Acc. buffer sufficient time to place its contents on the data bus. The Acc. buffer is enabled slightly ahead of SYSCL. |
| CDATA | Clock data | Write clock reg. U66 pin 11 | U52 pin 9 | Writes clock bits from write clock register to disk at 4μs intervals |
| CLEAR | Clear | **Interface signal** (P2 - A13) U22 pin 14 | Bistables and counters | After 500ms without new command it resets to 0 on processor interface and r/w logic. Unloads heads but does not return to track 0. Does not stop 2 x SYSCL so that a full SYSCL pulse can start at the correct time after CLEAR has gone |
| CLKBT | Clock bit | Read decoder U57 pin 9 | U43 pins 2,3 | When in either CLOCM or ID MODE produces input to read clock bistable when detecting clock pulses |
| | | | U43 pin 13 | Enables $\overline{K}$ input on read data bistable when detecting data pulses |
| CKSTR | Clock status register | Output decoder U39 pin 11 | U7 pin 11 | Clocks in data to status reg. from Acc. and Mem. buffer |
| CLOCM | Clock mode | Syncr. counter U71 pin 15 | U57 pin 11 | Read decoder input. Occurs after 15 SYSCL pulses when RDATA is at 4μs intervals |
| | | | U59 pin 7 | Enables ID counter to count to 110 and indicates the ID MODE. |
| CLOKE | Clock enable | Read decoder U57 pin 3 | U71 pin 7 | Enables syncr. counter but inhibits when no read pulse occurs. Maintains CLOCM during ID pattern |
| $C_n$+8 | Carry output from ALU | ALU register U37 pin 16 | U32 pin 14 | Produce jump instruction JCARY when a carry out signal from combined ALU register is indicated |
| COMCL | Combined clock | Output decoder U39 pin 15 | U49 pin 11 | Enable mode decoder. Selects from internal DB0 - 7 |
| | | | U49 pin 9 | (see also memory counter extension) |
| COMOR | Command or data registered | General address bus decode logic U61 pin 1 | U2 pin 3 | Sets bistable in input add. reg. to produce JCOLD when command instruction occurs |
| DATAB | Data bit | Read decoder U57 pin 2 | U43 pin 14 | Data pulse to set serial read input bistable |
| DATEN | Data enable | Mode decoder U49 pin 12 | U58 pin 15 | Resets SYNC STATUS bistable until CLOCM is established |
| | | | U63 pin 1 | Clears write/read counter at the start of read mode prior to SYNCR. |
| | | | U63 pin 10 | Sets JCODD at end of each data byte when w/r counter produces REGCL |
| | | | U65 pin 2 | Sets SELDT upon detection of A.M. when clock decoder circuit produces REGCL |
| | | | U67 pin 1 | Presets CRC generator before SYNCR occurs |
| | | | U64 pin 6 | Presets write/read counter to 8 after SYNCR for bit-counting to 15 |
| | | | U22 pin 12 | Resets read data and SYSCL timing logic until the CLOCK MODE is established, at the start syncr. MODE. |
| DATOP | Data operation | Read ser/par reg. U51 pin 13 | U55 pin 2 | Serial data stream, monitored by CRC during read operation |
| DBREN | Data bus register enable | Output decoder U32 pin 12 | U6 pin 11 | Clocks in from DB internal |

## Signal Tables

| Mnemonics | Signal name | Source | Destination/s | Effect of signal |
|---|---|---|---|---|
| DISCL | Disable clock (external) | TP2 | Oscillator U70 pin 5 | External signal applied to the test point to inhibit the oscillator |
| DMAPLS | DMA pulse | **Interface signal** P2 A20 | U55 pin 6 | With GRANT, enables the data transfer to or from main memory to proceed |
| DR2 | DMA request No. 2 | **Interface signal** U13 pin 6 | P1 A21 | Informs computer that disk is ready to accept data transfer when in the DMA mode. Computer sends GRANT |
| ENOMB | Enable output memory buffer | Inst. decoder U33 pin 5 | U20 pin 1 U34 pin 1 | Enables the memory buffer register contents to be placed on the internal data bus – but not during a 2-byte operation because of MBUFF |
| ENOSY | Enable output system | Interface logic U14 pin 11 | U6 pin 1 | Enables the output data register when writing to main memory during a DMA transfer |
| EXCLK | External clock | TP1 | Oscillator U70 pin 12 | External clock to replace oscillator for test purposes |
| FDSEN | Floppy disk system enable | Output decoder U39 pin 10 | U9 pin 11 | Clocks the floppy disk register for disk selection, direction and track position pulses |
| GECEN | General clock enable | Output decoder U75 pin 3 | U40 pin 10 | Anded with bit 3 to produce $\overline{\text{GENCL}}$ for an m.p. data/device transfer. Inhibited during a 2 byte operation because of $\overline{\text{IRENB}}$ |
| GENCL | General clock | Output decoder U41 pin 3 | U38 pin 15 | Clocks R1/R2 register for a DMA data transfer instruction |
| | | | U13 pin 11 U13 pin 3 | Clocks interrupt register for status transfer instruction (Controlled I/O). With bit 3 sets up data request for DMA transfers. Response from computer is GRANT |
| GRANT 2 | Grant No. 2 | **Interface signal** P1 B11 | U26 pin 1 U14 pin 13 | The computer acknowledgement to a DMA data request by No. 2. Priority . Enables DMA logic |
| $\overline{\text{GRANT 2}}$ | Grant No. 2 | Interface logic U26 pin 2 | U55 pin 5 | Anded with DMAPLS to allow a DMA transfer |
| | | | U11 pin 15 | Enables acknowledge signal to computer to send or accept data |
| | | | U12 pin 4 | Resets data request bistable each time a request is made |
| | | | U3 pin 1 U4 pin 1 | Enables the AHREN and ALREN to be gated to the main computer memory as a 16 bit word |
| IACK 6 | Interrupt acknowledge No.6 | **Interface signal** P1 A27 | U12 pin 9 | The computer acknowledgement to an interrupt request by interrupt level 6 to transfer status (controlled I/O). Resets interrupt register |
| IDMOD | Identification mode | Format decoder U73 pin 6 | U57 pin 12 | (READ mode) Input signal to the Syncr. decoder when the format ID marker is detected |
| | | | U73 pin 15 | Resets JTERR detection bistable after timing error |
| | | | U44 pin 16 | Produces jump instruction to indicate error if ID MOD remains in Syncr. gap between read before write mode |
| IDRES | Identification mode reset | Read decoder U57 pin 5 | U73 pin 3 | Resets ID MOD bistable on the first clock pulse after the missing clocks |
| INDEX | Index pulse (see JINDX) | Disk Interface JF 31 | U56 pin 6 | Indicates start of track, once per revolution of disk. Causes jump instruction. Start point for initialisation |
| INPUT | Input | Instruction decoder U33 pin 7 | U38 pin 1 U27 pin 10 U40 pin 2 | Produces read data and read clock register enabling signals with SEL bit 4, produces S register output enable signals (S → A) clock MD register to ALU for the 2 byte instruction (since IRENB is inhibited) |
| INCRL | Input register clock | Interface logic U12 pin 3 | U2 pin 9 | Clocks input addresses AB0 - AB2 into E register from computer |
| | | | U8 pin 11 | Clocks input data DB0 - DB7 into E register from computer |
| IOIN | I/O in | **Interface signal** P2A17 | U26 pin 13 | Signal from computer to enable status transfer (controlled) from status register |
| IOW | I/O write | **Interface signal** P2A19 | U1 pin 3 | Signal from computer to enable the clocking of the input address and data registers. (Controlled transfers) |
| IRCLK | Instruction register clock | IR enable logic U25 pin 11 | U32 pin 11 | Clocks 8 bistables in total:- Outputs 2, 5, 6, 9 to Inst. decoder Outputs 12, 15, 16, 19 to Jump selector Clocks on every SYSCL except in middle of 2 byte operation |

| Mnemonics | Signal name | Source | Destination/s | Effect of signal |
|---|---|---|---|---|
| IRENB | Instruction register enable | 2 byte logic U75 pin 11 | Normally low, except in 2 byte operation U25 pin 12 | With SYSCL, produces IRCLK enables IR register |
| | | | U18 pins 9, 10 | With SYSCL, MD bits 0 - 3 enables ALU selection |
| | | | U40 pin 1 | With SYSCL, register MD bits 0 - 5 |
| | | | U27 pin 13 | Primes Acc. |
| IRENB | Instruction register enable | 2 byte logic U53 pin 6 | Normally high except in 2 byte operation U67 pin 9 | Enables jump selector for a 2 byte operation |
| | | | U75 pin 1 | Inhibits GENCL during 2 byte operation |
| | | | U20 pins 9, 10 U34 pins 9, 10 | Enables memory data into memory buffer during 2 byte operation |
| | | | U40 pin 13 | Inhibits writing into S register whilst data is being taken from Mem. buffer during 2 byte operation. Will accept this data on next SYSCL. |
| IREQ 6 | Interrupt request No. 6 | Interface signal U13 pin 8 | P1A26 | Interrupt request from floppy to computer to read status register. Occurs when int. data bit 1 is set to 1 |
| JA = B | Jump if A = B | Instruction register U32 pin 12 | U44 pin 7 | Jump instruction when contents of Acc. equal internal data bits 0 - 7 compared in the ALU |
| JCARY | Jump if carry | Instruction register U32 pin 15 | U44 pin 6 | Jump instruction as a result of $C_n$+8 from combined ALU register |
| JCODD | Jump if command or data is detected | R/w count logic U65 pin 9 (TP 5) | U44 pin 21 | Jump instruction for every byte of data read or written and including the read AM byte (110) and CRC bytes |
| JCOLD | Jump if command register loaded | Instruction register U32 pin 19 | U44 pin 4 | Jump instruction when an interface command (37) is loaded into the E register address AB0 - 2 |
| | | | U1 pin 4 | Is low prior to command instruction and enables controlled I/O transfer |
| JCRCE | Jump if CRC error | R/w logic U68 pin 13 | U44 pin 22 | Jump instruction when the CRC generator indicates an error at the end of a read header or data block |
| JDATRQ | Jump if data requested | Interface logic U62 pin 6 | U44 pin 20 | Jump instruction when data is requested from or to main memory during a DMA transfer. Enables data to be transferred at correct time and initiates error if GRANT is not received |
| JNINX | Jump if not index pulse | Disk interface U56 pin 7 | U44 pin 18 | Jump instruction when the disk electronics detects that index marker (start of track) has not yet been detected |
| JNIR | Jump if input register loaded | Instruction register U32 pin 16 | U44 pin 5 | Jump instruction when address AB0 - 2 and data DB0 - 7 are clocked into the E register |
| JNRDY | Jump if not ready | Disk interface U56 pin 2 | U44 pin 2 | Jump instruction when disk is NOT physically able to accept or send data it selected, loaded, up-to-speed and heads loaded |
| JNTR0 | Jump if not track 00 | Disk interface U56 pin 5 | U44 pin 1 | Jump instruction when disk electronics detect that the heads are NOT on track 00 |
| JNWPT | Jump if not write protected | Disk interface U56 pin 10 | U44 pin 23 | Jump instruction when the disk electronics detect a NOT write protected disk |
| JPULS | Jump if data pulse detected | Read Syncr. logic U52 pin 6 (TP 3) | U44 pin 3 | Jump instruction which detects clock pulses from disk to establish SYNCR. (i.e. when 16 JPULS signals occur at 4µs intervals after the first CLOCM is established). After SYNCR the JPULS takes no active part in the disk logic sequences |
| JPRST | Jump reset sync. status | Read Syncr. logic U52 pin 10 (TP 4) | U44 Pin 17 | Jump instruction occurring with data/clock pulses from disk at 2µs intervals. Inhibited when CLOCM is established, prior to SYNCR. |
| JTERR | Jump if timing error | Read timing logic U52 pin 5 | U44 pin 19 | Jump instruction when 2 or more clock pulses are missing other than when looking for the address marker |
| | | | U77 pin 13 | Feedback line to latch JTERR until software resets the whole read logic |
| JWORE | Jump enable | Instruction decoder U33 pin 2 | U67 pin 10 | Enables jump selector to load new address into the program counter for 2 byte operation |

## Signal Tables

| Mnemonics | Signal name | Source | Destination/s | Effect of signal |
|---|---|---|---|---|
| LSB | Least significant bit | All registers | Registers and busses | Generally as LHS bit on horizontal registers. Generally lowermost bit on vertical registers. (See manufacturers data i.e. Texas, for clarification of position) |
| MBUFF | Memory buffer enable | 2 byte logic U72 pin 9 | U20 pin 2 U34 pin 2 | Enables output of memory buffer for 2 byte operation |
| | | | U75 pin 13 | Causes IRENB signals to go not true during 2 byte operation |
| M.D. Bits 0 - 5 | Memory data bits | ALU selection register U19 pins 15, 12 10, 7, 5, 2 | U23, U27 pins 8,3,4,7,5,6 | Selects ALU functions as shown in 74LS 181 Active High Table (Texas) M, S3, S2, $C_n$, S1, S0 |
| MIN | Memory in | **Interface signal** U11 pin 13 | P2A15 | Signal to computer main memory to place data on general bus during DMA transfer to floppy disk |
| MSB | Most significant bit | All registers | Registers and busses | General as RHS bit on horizontal registers. Generally uppermost bit on vertical registers. (See manufacturers data i.e. Texas, for clarification of position) |
| MW | Memory write | **Interface signal** U11 pin 11 | P2A16 | Signal to computer main memory to accept data on general bus from floppy disk during a DMA transfer |
| OSTEN | Output status enable | Interface logic U14 pin 3 | U7 pin 1 | Enables the contents of the status register to be placed on the general data bus following an interrupt request to transfer status. Also at the request from the main computer |
| OUTPT | Output | Instruction decoder U33 pin 4 | U27 pin 4 | Part of input selection to output decoder (when SEL bit 3 = 0) to produce mp output signals |
| | | | U41 pin 4 | Produces GENCL (when SEL bit 3 = 1) for 1 or 2-byte mp output signals |
| | | | U26 pin 11 | Enables writing to S register from Acc. or Memory. (When SEL bit 4 = 0) |
| PLOAD | Program counter load | Jump selector U44 pin 10 | U17 pin 9 U31 pin 9 | Output of jump selector to load the memory data bus into the program counter |
| PSEL 1 | Program select enable | Mode decoder U49 pin 2 | U29 pin 19 U15 pin 19 | Enables addressing from locations 0 to 511 in the read only memory as shown in the PSEL 2 table |
| PSEL 2 | Program select enable | Mode decoder U49 pin 6 | U29 pin 15 U53 pin 11 U15 pin 15 | Enables addressing to a specific part of the read only programmed memory. With a COMCL operation, DB0 (PSEL 1) is used with DB2 (PSEL 2) to extend the addressing from 511 to 1023 |
| $\overline{\text{PSEL 2}}$ | Program select enable | Mode decoder U53 pin 10 | | |

| PSEL 1 (DB0) | PSEL 2 (DB2) | ADD | CHIP |
|---|---|---|---|
| 0 | 0 | 0- 255 | U29 |
| 1 | 0 | 256- 511 | U29 |
| 0 | 1 | 512- 767 | U15 |
| 1 | 1 | 768-1023 | U15 |

| | | | | |
|---|---|---|---|---|
| RCMDIS | Read clock mode disable | Mode decoder U49 pin 9 | U72 pin 1 | Resets read data input enable signal to CLOCM counter |
| | | | U71 pin 1 | Resets clock mode counter (CLOCM). Produced as a result of a Syncr. fault. |
| RCREN | Read clock register enable | Input decoder U38 pin 6 | U46 pin 1 | Enables the data in the parallel read register to be clocked on to the internal data bus and transferred to the Acc. Occurs when JCODD is detected by software |
| RDATA | Read data | Disc interface JF pin 5 | U69 pin 13 U74 pin 5 | Raw data from disk is synchronised with SYSCL and either stretched or shortened to a standard 200ms pulse width |
| | | | R6 | Pull-up resistor for TTL compatibility and correct matching |
| RDATC | Read data clock | Read logic U60 pin 12 | U63 pin 5 | Produces a position pulse for every clock pulse being read. Allows counter to count only at this time |
| | | | U62 pin 3 | Produces clock pulses to clock CRC generator for every data bit (DATOP) |
| RDREN | Read data register enable | Input decoder U38 pin 7 | U50 pin 1 | Enables the data in the parallel read register to be clocked on to the internal data bus and transferred to the Acc. Occurs when JCODD is detected by software |
| READY | Floppy disk ready (see JRDY) | Disk interface JF pin 29 | U56 pin 3 | Causes jump instruction (see JNRDY) |

## Signal Tables

| Mnemonics | Signal name | Source | Destination/s | Effect of signal |
|---|---|---|---|---|
| REGCL | Register clock | R/W logic U62 pin 16 | U52 pin 11 U52 pin 13 | Clocks data and clock alternately every 2μs when writing on to disk. R/W counter produces pulses every 2μs by counting SYSCL pulses from 6 to 15 cyclically. |
| | | | U65 pin 11 (TP5) | Sets JCODD on the next SYSCL following the SYSCL which caused the counter to output (pin 15) after receiving 8 clock pulses when reading |
| | | | U66 pin 4 | Produces CRCE clock pulse when writing. Inhibited on alternate pulses by WDSEL, to clock only at 4μs intervals |
| $\overline{\text{REGCL}}$ | Register clock | R/W logic U53 pin 2 | U54 pin 2 | When writing, clocks 16 pulses into counter to indicate that 1 byte of information has been written (1 byte = 8 CKS + 8 DATA) WRITE MODE |
| | | | U46 pin 11 U50 pin 11 | After 8 clock pulses (1 byte), clocks the serial clocks/data from the serial read to the parallel read registers READ MODE |
| | | | U47 pin 2 U48 pin 2 | When writing, clocks the previously loaded parallel data, serially out of the write clock and data registers. Clocks out every 4μs since WDSEL alternately inhibits the REGCL (from U54) WRITE MODE |
| | | | U65 pin 3 | Sets the SELDT bistable when AM is detected when reading. Disabled by WMODE when writing READ MODE |
| RESCN | Reset counter | Read decoder U57 pin 7 | U59 pins 9, 10 | After the start of CLOCM pulses this signal resets ID counter to zero after nominally 8 SYSCL except when a clock pulse fails to appear in sync mode and clock mode. (Start of ID detection or possible JTERR). |
| RESDI | Reset data input | R1/R2 decoder U38 pin 9 | U12 pin 12 | Resets the input address register AB0-2 from computer. Results from JCOLD or JINLD instructing program to set busy status and reset input. Byte 2 in GENCL organises status transfer and reset signals |
| RESJC | Reset JCODD | R1/R2 decoder U38 pin 11 | U66 pin 10 | Resets JCODD immediately the software detects that JCODD has been set. Occurs after each JPULS during synchronisation, on detection of AM byte and after each byte read or written |
| RESTF | Reset status flip-flop | JCODD logic U66 pin 8 | U65 pin 13 | Produced by $\overline{\text{RESJC}}$ to reset the JCODD bistable |
| | | | U58 pin 1 | Resets JPULS until SYNCR is established and thereafter each time JCODD is detected. Setting or resetting the JPULS after SYNCR has no significance and is ignored by the software |
| RWCLR | Read/write clear | R/W logic U63 pin 3 | U64 pin 1 | Occurs when write mode is initiated from a JCODD count after the 11th byte following the header and at the start of the 6 byte syncr. gap. Also occurs for synch. fault or any read fault |
| | | | U62 pin 1 U65 pin 1 U66 pin 9 | Resets bistables in RDATA logic, R clock logic, REGCL line and RDATC logic. Resets JCODD and SELDT bistables and JPULS via $\overline{\text{RESTF}}$ |
| 2 BYTE | Second byte | Instruction decoder U33 pin 1 | U72 pin 14 | Decoded when an m.p. instruction code requires data or address information to be transferred on the internal data and address busses |
| | | | U74 pin 1 | Inversion to U75 pin 13 input to allow 'toggle' or 'no-change' of bistable |
| | | | U75 pin 12 | With output of U75, inhibits the IRENB signals during 2 byte operation |
| 7 CENB | Seven count enable | Mode decoder U49 pin 5 | U64 pin 3 | Initiated by software when the JCODD is set by the REGCL which resets the R/W counter to count in the 2nd byte of the CRC control byte. The R/W counter is preset to 1001(9). After 7 more RDATC pulses the REGCL is set and sets JCODD which informs the software that the complete CRC control word has been clocked into the CRC checker. |

## Signal Tables

| Mnemonics | Signal name | Source | Destination/s | Effect of signal |
|---|---|---|---|---|
| SEBEN | Scratch pad- and E-registers enable | Instruction decoder U33 pin 3 | U18 pins 1, 2 | Enables the instruction selection register to address the S register, for single bytes only, during E to S transfers of disk commands and initial addressing |
| $\overline{\text{SEBEN}}$ | Scratch pad- and E-registers enable | Instruction decoder U77 pin 8 | U28 pin 1 | Allows add. and data from E reg. to be clocked in S. register (SEL bit 4 = 0) |
| | | | U8 pin 1 | Enables E reg. contents to be placed on to internal data bus for transfer to S reg. |
| | | | U11 pin 1 | Enables E reg. (ADDRESS) contents to be placed on int. add. bus to S reg. |
| SELB 0 | Instruction selection register bit 0 | Instruction selection register U18 pin 3 | U44 pin 15 | Part of address selection C0 - C3 for jump instructions |
| | | | U39 pin 1 | Part of input selection for output decoder |
| | | | U38 pin 2 | Part of input selection for input decoder |
| | | | U21 pin 3<br>U35 pin 3 | Part of address selection for S register |
| SELB 3 | Instruction selection register bit 3 | Instruction selection register U18 pin 6 | U33 pin 14 | Part of instruction decoder |
| | | | U44 pin 11 | as bit 0 (pin 15) |
| | | | U27 pin 5<br>U40 pin 9 | Part of input selection for output decoder and GENCL logic |
| | | | U21 pin 6<br>U35 pin 6 | As bit 0 (pin 3) |
| SELB 4 | Instruction selection register bit 4 | Instruction selection register U32 pin 9 | U33 pin 13 | As bit 3 |
| | | | U75 pin 2 | Part of input to output decoder and selection of GENCL logic |
| | | | U38 pin 3 | As bit 0 (pin 2) |
| | | | U27 pins 9, 2 | Selects either write or read enabling signals for the S register |
| SELB 1, 2 | Instruction selection register bits 1 and 2 | Instruction selection register U18 pins 4, 5 | U44 pins 14, 13 | As bit 0 (pin 15) |
| | | | U39 pins 2, 3 | As bit 0 (pin 1) |
| | | | U21 pins 4, 5<br>U35 pins 4, 5 | As bit 0 (pin 3) |
| SELDT | Select data | Read logic U65 pin 5 | U60 pin 13 | Set high by REGCL on detection of the AM (110) at the start of the read cycle. Stays high during the read mode |
| $\overline{\text{SELDT}}$ | Select data | Read logic U65 pin 6 | U67 pin 5 | Low when reading. Preset to low when in WMODE to allow WSEL to inhibit alternate REGCL to CRC generator<br>Is high before the first REGCL is set by the AM, therefore with this REGCL, allows the first AM bit to be clocked into CRC. This is the only time U67 is used to clock CRC during read operation |
| SPRCS | Scratch pad register clock enable | Scratch pad logic U28 pin 11 | U21 pin 2<br>U35 pin 2 | Enables the scratch pad register (S) when either read or write are selected. (Chip enable signal CE) |
| SPROE | Scratch pad register output enable | Scratch pad logic U27 pin 8 | U28 pin 13 | Produces the SPRCS signal for the scratch pad read operation |
| | | | U42 pins 1, 19 | Enables and inverts the scratch pad outputs on to the internal data bus |
| SPRWE | Scratch pad register write enable | Scratch pad logic U41 pin 8 | U21 pin 3<br>U35 pin 3 | Enables the scratch pad register (S) to write in data. The SPRCS signal is also required for complete operation |
| SRCLK | Serial register clock | Read decoder U57 pin 6 | U41 pin 12 | Produces a low pulse for every read pulse after the CLOCK MODE is established. Also produces low pulse during and for the detection of the ID MODE. Provides a continuous read clock pulse for subsequent read logic. |
| SYNCR | Synchronize | Mode decoder U49 pin 19 | U57 pin 13 | After 16 counts by software of JPULS after CLOCM is established, and no JREST, the SYNCR signal on DB7 is clocked by COMCL. The SYNCR signal input into the read decoder U57 modifies the programmed matrix to allow the read logic to detect the IDMOD. |
| | | | U59 pin 1<br>U73 pin 1<br>U43 pins 1, 15<br>U45 pin 9<br>U51 pin 9<br>U60 pin 2 | The SYNCR signal inputs to the read data and clock register S and gates are enabling levels which allow the read mode to continue |
| SYSCL | System clock | Oscillator U70 pin 8 | All logic | Supplies 50ns negative going pulses every 200ns from a Xtal controlled oscillator circuit. The SYSCL pulses provide the time clock for the floppy disk logic. |

## Signal Tables

| Mnemonics | Signal name | Source | Destination/s | Effect of signal |
|---|---|---|---|---|
| 2.SYSCL | 2 x System clock | Oscillator<br>U70 pin 11 | U77 pin 1<br>U70 pin 10 | Supplies 50ns pulses every 100ns, (pulse width ratio 1:1) feeding a ÷2 and gating circuit to produce the SYSCL time clock pulses |
| | | | U76 pin 12 | Clocks in RDATA from disk and combines with the SYSCL to produce a standard 200ns data read pulse. This input circuit ensures that very short and also very long incoming disk pulse can be standardised. i.e. pulses ≪ 200ns and ≫ 200ns are held to 200ns. |
| TRACK 00 | Floppy disk on track 00<br>(see JTRCO) | Disk interface<br>JF pin 9 | U56 pin 5 | Produces jump instruction JNTRO when disk electronics detects that the head is not on track 00 |
| WCREN | Write clock register enable | Output decoder<br>U39 pin 7 | U48 pin 1 | Loads parallel data into write clock register (internal bus DB0-7) from Acc. or memory (2 byte operation) |
| WDATA | Write data | Disk interface<br>U36 pin 9 | JF pin 13 | Transfers write data and write clock pulses to the disk head |
| WDREN | Write data register enable | Output decoder<br>U39 pin 9 | U47 pin 1 | Loads parallel data into write data register (internal bus DB0-7) from Acc. or memory (2 byte operation) |
| WDSEL | Write data select | Write byte counter<br>U54 pin 14 | U52 pin 12<br>U53 pin 3 | The l.s.b. from the write byte counter to produce a pulse of 2 x REGCL.<br>Is inverted to produce $\overline{\text{WDSEL}}$ and when anded with the REGCL causes an alternate clocking/enabling of clock and data pulses through U52. |
| | | | U67 pin 4 | Produces CRCE clock pulses every 4μs by anding with REGCL for loading write data pulses into the CRCE generator |
| $\overline{\text{WDSEL}}$ | Write data select | Write byte counter<br>U53 pin 4 | U52 pin 10 | As described for WDSEL and forms a 4μs 1:1 pulse width ratio |
| | | | U47 pin 15<br>U48 pin 15 | With $\overline{\text{REGCL}}$, clocks out serial data and clock bits from the write registers for transfer to disk |
| WMODE | Write mode | R/W mode decoder<br>U63 pin 11 | U10 pin 6 | Write mode is initiated by software after reading the last byte of 11 in gap 2 of the header |
| | | | U63 pin 4 | Enables the R/W counter through the write mode operation. |
| | | | U63 pin 2 | Produces the R/W CLEAR signal to reset the R/W logic at start of write mode as determined by software and hardware errors. |
| | | | U64 pins 4, 5 | Presets the R/W counter to 6 (0110) at each load pulse to produce REGCL pulses of 200ns at 2μs intervals |
| | | | U54 pin 1 | Enables the write byte counter |
| | | | U66 pin 13 | Enables write clock pulses to be gated from the register to produce CDATA signal |
| | | | U77 pin 9 | Presets the $\overline{\text{SELDT}}$ signal low to allow WDSEL and REGCL signals to produce the CRC clock pulse |
| | | | U14 pin 4 | Enables main memory transfer to floppy disk when DMA pulse is granted. Produces $\overline{\text{MIN}}$ |
| $\overline{\text{WMODE}}$ | Write mode | R/W mode decoder<br>U77 pin 8 | U65 pin 4 | As WMODE on U77 pin 9 |
| | | | U14 pin 10 | Enables floppy disk transfer to main memory when DMA pulse is granted. Produces $\overline{\text{MW}}$ |
| | | | U14 pin 12 | With GRANT, enables the data output register |
| WMOD 1 | Write mode 1 | R/W mode decoer<br>U49 pin 16 | U63 pin 12 | Maintains WMODE and WDATA signals during the end of block writing and start of CRC 2 write bytes |
| | | | U66 pin 2 | Enables the gate to write the CRC data bytes at end of data block |
| | | | U67 pin 2 | Presets CRC generator at the start of the write data block |
| | | | U69 pin 2 | When $\overline{\text{WMOD2}}$ is high at the end of the write data block, a low is placed (from U69 pin 3) to the CRC check word enable input. The check word is gated out of CRC via pin 12 |
| $\overline{\text{WMOD 2}}$ | Write mode 2 | R/W mode decoder<br>U53 pin 12 | U52 pin 4 | Enables the gate to write the data bits of the check word from the CRC generator at the end of the write data block |
| | | | U69 pin 1 | Enables CRC output when it goes high at end of data block |
| WRTPT | Floppy disk write protected<br>(see JNWPT) | Disk interface<br>JF pin 7 | U56 pin 11 | Produces jump instruction JNWPT when diskette/electronics detects 'not write protected' |

## INPUT/OUTPUT CONTROL

## Input/Output Control

The input/output logic operates in 2 modes,

(a) the controlled transfer mode (I/O mode), and

(b) the direct memory access mode (DMA).

The **controlled transfer mode** is responsible for

(a) Setting up instructions, i.e. drive, sector etc.,

(b) Function commands, i.e. write data etc.,

(c) Status register information, i.e. ready, busy etc.

The **direct memory access mode** is responsible for

(a) Transferring data blocks from the main computer memory to the diskette (write mode)

(b) Transferring data blocks from the diskette to the main computer memory (read mode)

TDV 2114
1426 - 10 - 76

The basic difference between the two modes is that the main computer program controls the sequence and timing for the controlled transfers, whereas, in the DMA mode, all control is between the input/output logic and the main memory, with no main program interference. The main computer is kept in a waiting mode each time a DMA transfer takes place.

The input/output control logic responds to the appropriate interface signals but they alone cannot organise the flow across the interface. The processor program provides the enabling and clocking signals for the input and output registers in response to jump instruction signals. These signals are JINLD and JCOLD for controlled transfers and JDATRQ for DMA transfers.

Simplified input/output functions are shown in Tables 1 and 2 with Table 3, scratch pad locations, included for completeness.

The description of each type of transfer should be read in conjunction with its waveform. Table 2 should provide the most useful reference of the three tables.

The diskette storage drive interface signals are described fully in the Diskette Storage Drive, Original Equipment Manufacturer's Manual.

The input/output instruction and command address formats are given in the Software section of this Controller Manual.

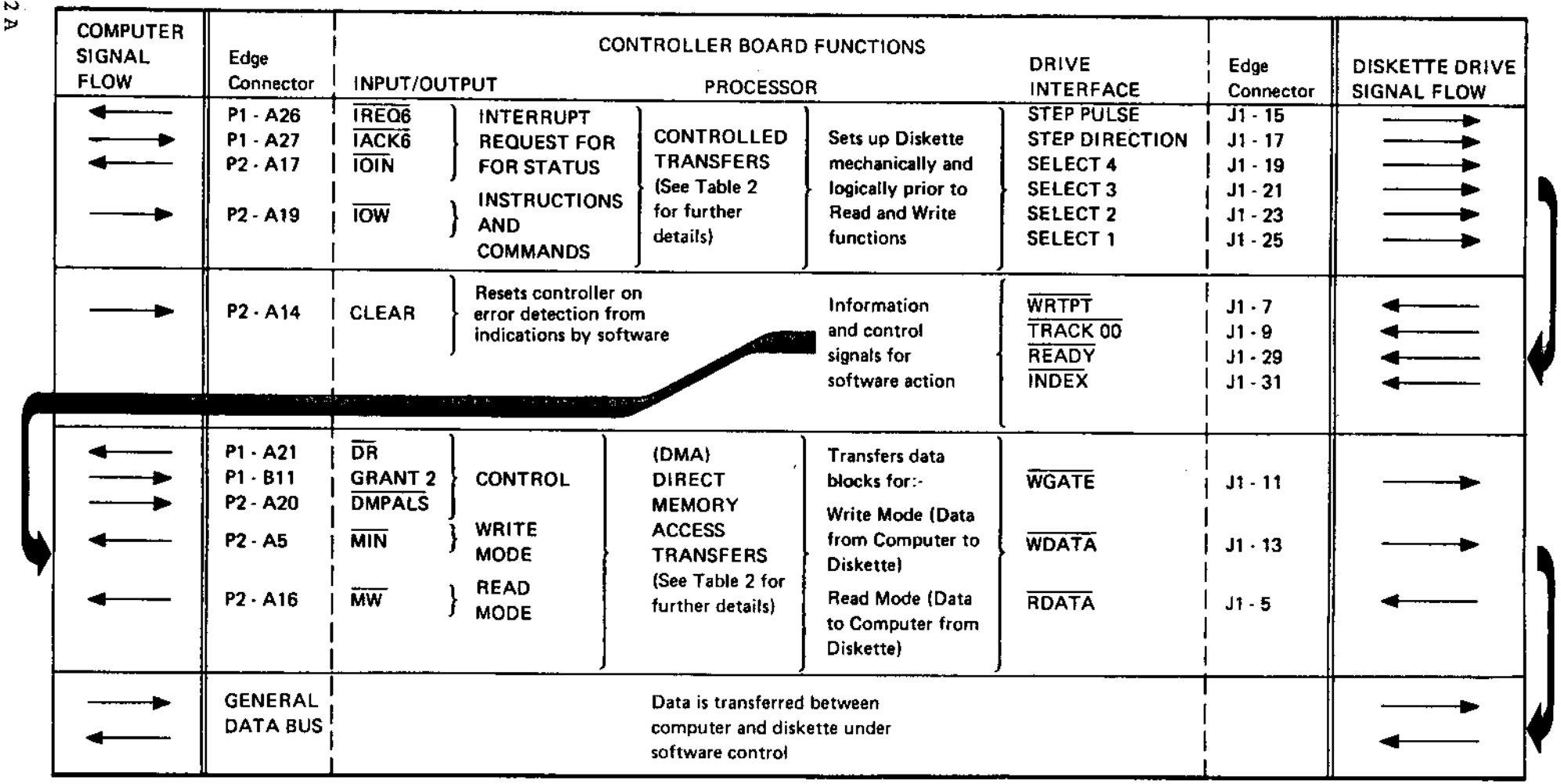



**Table 1 Generalised Sequence of Input/Output and Interface signals**

1426-02-79
TDV 2114

| Type of Transfer | Input/ Output Functions | Interface Signals | Logic Signals | General Data Bus DB 0 - 7 | Universal Address Bus AB 0 - 15 |
|---|---|---|---|---|---|
| CONTROLLED | INSTRUCTIONS | $\overline{IOW}$ | JINLD | Data to E Register<br>Data on DB 0 - 7<br>is Hex. OX DRIVA<br>XX RELTA<br>XX SECTA<br>XX ADDHI<br>XX ADDLO | Decode on AB 0 - 7<br>is Hex. 30 DRIVA<br>31 RELTA<br>32 SECTA<br>33 ADDHI<br>34 ADDLO<br>AB 0 - 2 to E Register |
| | COMMANDS | $\overline{IOW}$ | COMOR<br>JCOLD<br>JINLD | Data to E Register<br>Data on DB 0 - 7<br>is Hex. F1 WRTDA<br>F2 WTDEL<br>F8 WRTFO<br>E0 REDAT<br>E6 REDID<br>E3 SEEKT<br>EC RECAL | Decode on AB 0 - 7<br>is Hex. 37 for all commands<br>AB 0 - 2 to E Register |
| | STATUS | $\overline{IREQ6}$<br>$\overline{IACK6}$<br>IOIN | $\overline{OSTEN}$ | Status register to data bus DB 0 - 7<br>is Hex. XX SENST | Decode on AB 0 - 7<br>Is Hex. 37 SENST |
| DIRECT MEMORY ACCESS (DMA) | DMA (Read Mode) Data from diskette to Main Computer Memory) | $\overline{DR2}$<br>GRANT 2<br>$\overline{DMAPLS}$<br>$\overline{MW}$ | $\overline{ENOSY}$<br>WMODE | Data from data register to data bus DB 0 - 7 | (Instructions and Commands already given)<br>High and low address locations to bus AB 0 - 7 and AB 8 - 15 |
| | DMA (Write Mode) Data from Main Computer Memory to diskette) | $\overline{DR2}$<br>GRANT 2<br>$\overline{DMAPLS}$<br>$\overline{MIN}$ | $\overline{INRCL}$<br>WMODE | Data to E Register from data bus DB 0 - 7 | (Instructions and Commands already given)<br>High and low address locations to bus<br>AB 0 - 7 and AB 8 - 15 |

**Table 2 Input/Output Functions**

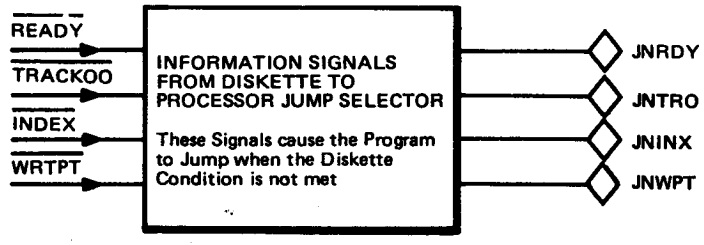
**Scratch Pad Register Locations**

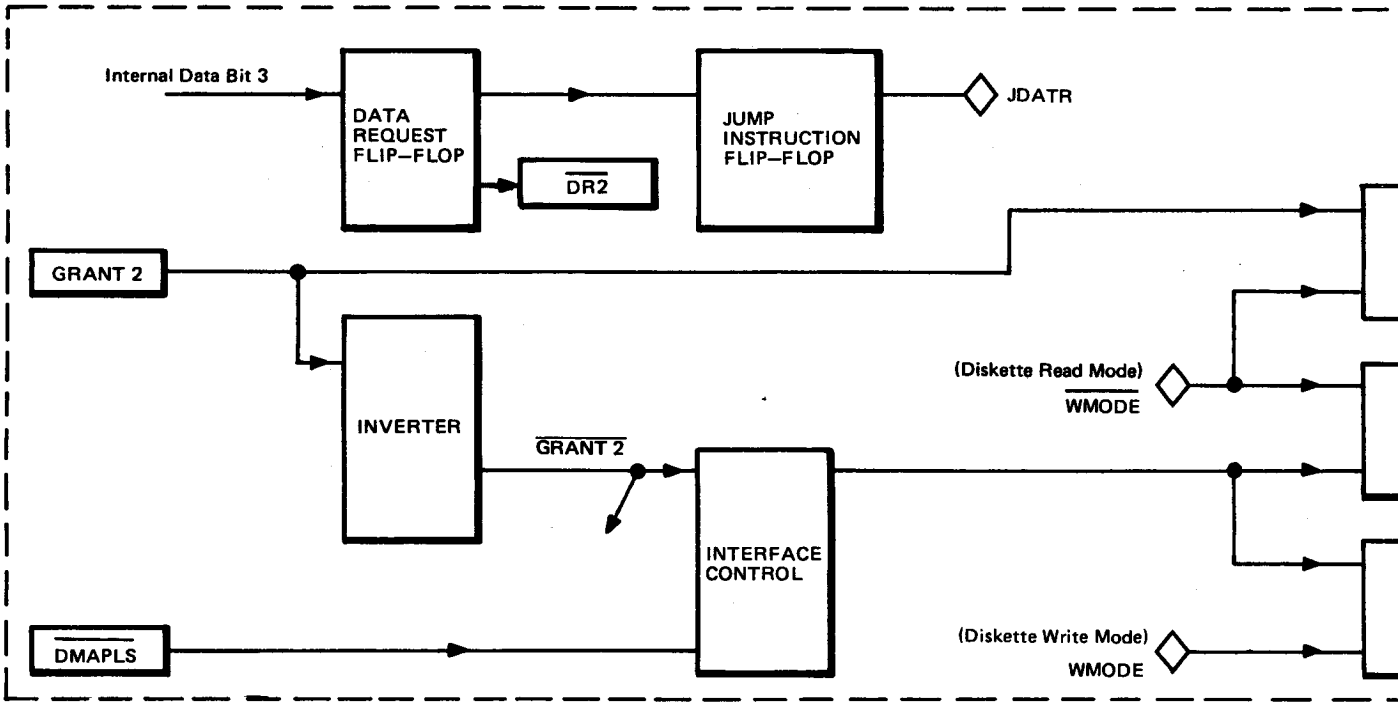
| Location | Contents |
|---|---|
| S0 : | Drive Address<br>Data during write operation |
| S1 : | Relative track Address<br>Direction of head movement |
| S2 : | Sector Address |
| S3 : | The most significant byte of the main memory address |
| S4 : | The least significant byte of the main memory address |
| S5 : | Block length<br>Clock data during write format operation |
| S6 : | This register contains information to let the software know if the correct header has been read during a read data field operation |
| S7 : | Command register |
| S8 : | Device No., step direction, step pulse |
| S9 & SA : | Counters to count the number of sync. errors and retries. Only S9 is employed with read operation between header and main block. SA is also employed as a general purpose register. |
| SB : | Counts missing sectors etc. Also employed during seek/recalibrate and write format operation |
| SC : | Contains the block length in steps of 128. (0 = 128, 1 = 256, 3 = 512) |
| SD : | Storage for counting index-strobes. Also employed as a general storage register during write operation |
| SE : | Contains the correct header to be written |
| SF : | Contains the information which is going to be sent to the status register |

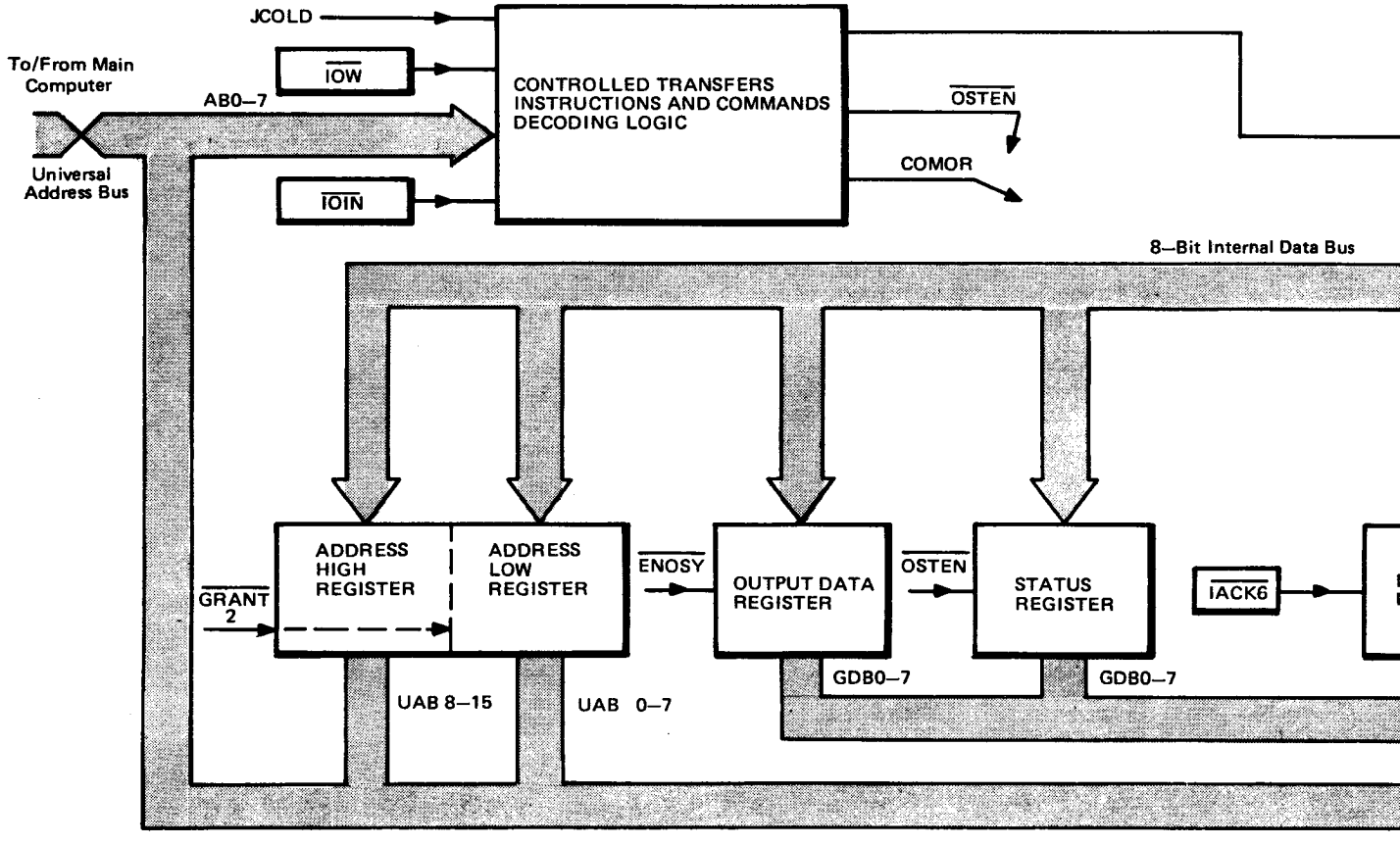
**Table 3 Scratch Pad Locations**

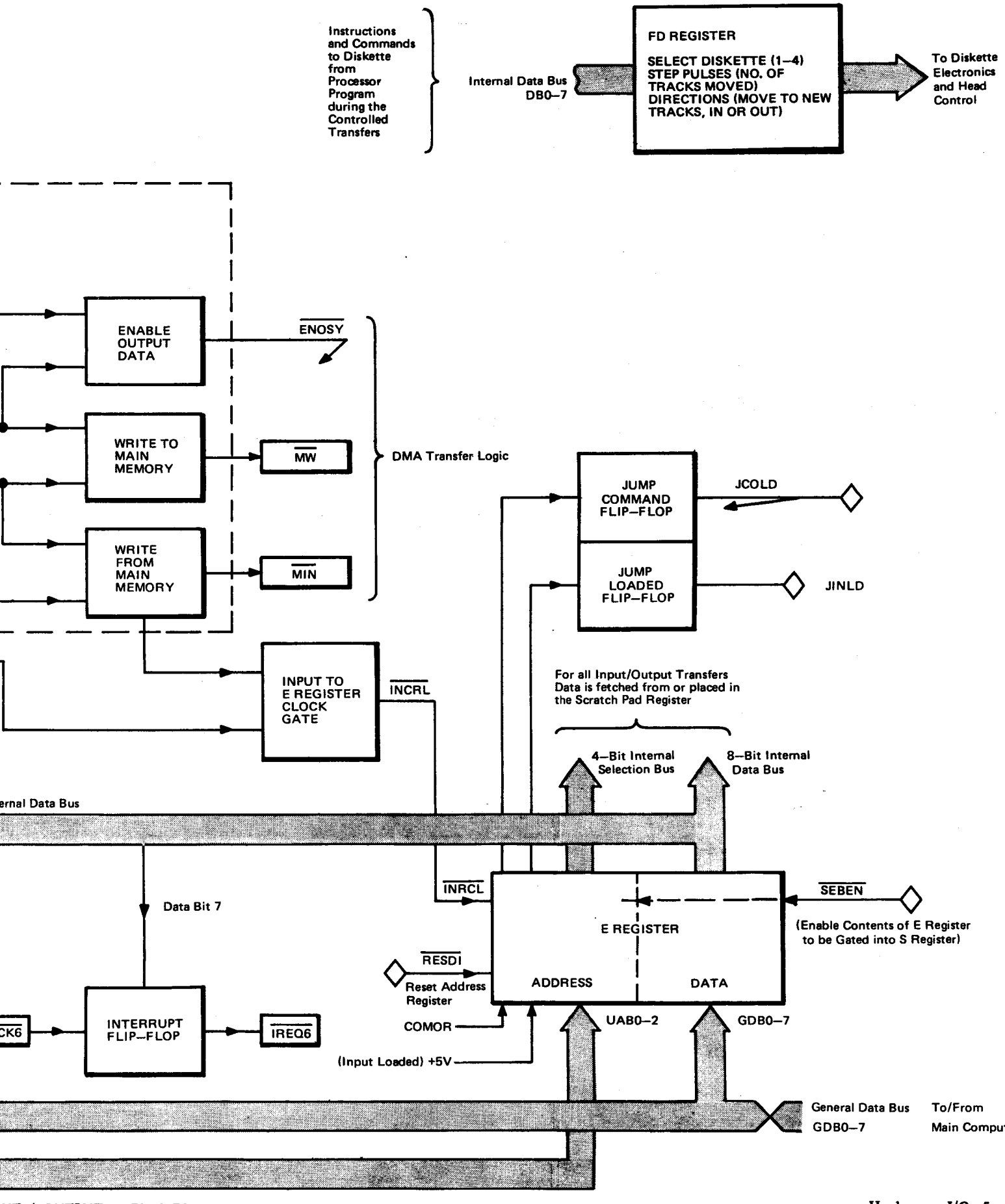


TDV 2114
1426 - 10 - 76


READY
TRACK00
INDEX
WRTPT
INFORMATION SIGNALS FROM DISKETTE TO PROCESSOR JUMP SELECTOR
These Signals cause the Program to Jump when the Diskette Condition is not met
JNRDY
JNTRO
JNINX
JNWPT


Internal Data Bit 3
DATA REQUEST FLIP–FLOP
DR2
JUMP INSTRUCTION FLIP–FLOP
JDATR
GRANT 2
INVERTER
GRANT 2
(Diskette Read Mode)
WMODE
INTERFACE CONTROL
DMAPLS
(Diskette Write Mode)
WMODE

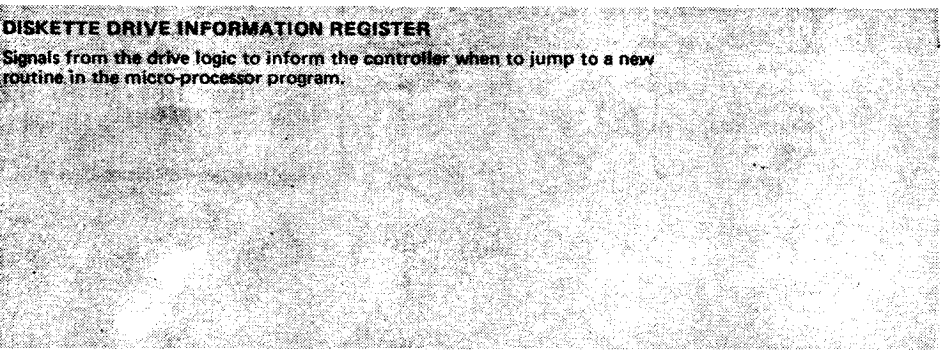

JCOLD
To/From Main Computer
IOW
AB0–7
CONTROLLED TRANSFERS INSTRUCTIONS AND COMMANDS DECODING LOGIC
OSTEN
COMOR
Universal Address Bus
IOIN
8–Bit Internal Data Bus
ADDRESS HIGH REGISTER
ADDRESS LOW REGISTER
ENOSY
OUTPUT DATA REGISTER
OSTEN
STATUS REGISTER
IACK6
GRANT 2
GDB0–7
GDB0–7
UAB 8–15
UAB 0–7




PUT / OUTPUT — Block Diagram

## DISKETTE DRIVE INFORMATION REGISTER

Signals from the drive logic to inform the controller when to jump to a new routine in the micro-processor program.



## DIRECT MEMORY ACCESS TRANSFERS

### READ MODE (Waveform 4)

The read mode is when data is read from the diskette and written into the main computer memory.

A DMA request occurs when data in the data output register is ready to be written to the main memory location, already specified by the address high and low register bytes.

The processor program transfers these high and low addresses from the S register via the accumulator and into the high and low address registers. The address bytes are then incremented by the accumulator and set back into the S register for the next DMA request. Data read from the diskette is transferred from the read logic, into the processor and clocked into the data output register.

A general clock instruction is then generated in the processor and sets bit 3 on the internal data bus into the data request flip-flop, which then generates the low interface signal DR2. A second flip-flop sets jump instruction JDATRQ to inform the processor software that a data request has been made. The read logic can then continue knowing that the existing data byte will be cleared from the output register.

The computer acknowledges DR2 by sending the GRANT2 signal which, (a) places the high and low address on to the address bus, (b) produces the low signal ENOSY which places the data on to the data bus, and finally, resets the data request flip-flop.

After approximately 250ns the computer sets the DMAPLS signal low, which with the GRANT signal, produces the memory write interface signal MW. This low signal allows the data to be written into the main memory location.

The interface signals, GRANT and DMAPLS, are automatically reset in a time to ensure that writing to the main memory is complete. When these signals are reset, the MW and ENOSY signals are also reset. The DMA transfer is then complete and a further transfer is set up for the next data byte read from the diskette.

### WRITE MODE (Waveform 5)

The write mode is when data is read from the main computer

The DMA request occurs when the processor program is ready location, already specified by the address high and low registe processor each time a data transfer takes place so that a new byte until the complete block is transferred. The start address controlled transfer instructions and stored in the S register un

Then follows the same sequence as in the read mode when DF required. The computer acknowledges this with the GRANT a written to the diskette, the WMODE signal is high which enab produces interface signal MIN and the input register clock, IN

The low MIN signal places data from the main memory locati on to the interface data bus. The data is then clocked into the the end of the DMA request. This occurs when the interface s in a time to ensure that reading from the main memory is con

When the E register is loaded by INRCL, the +5 logic level on and produces JINLD. The processor software then resets the the DMA transfer must be placed in address location 0 in the processor and clocks the data into the S register to complete

## INSTRUCTIONS (Waveform 1)

Any one of four diskettes can be selected by the main computer. Each diskette is selected by addressing the drive logic via the FD register.

The main computer program places the instruction on the data bus and the function code on the address bus. Interface signal IOW is then set low for approximately 500ns giving the decoding logic sufficient time to clock the data and address bits into the E register.

Address bits AB0 - 7 (Hex. 30 drive address) are decoded via a pre-programmed read only memory and a series of gates to produce the low going signal INRCL, since JCOLD is low (not a command) and IOW is low.

The input register clocking signal, INCRL, clocks in the data and address bits to the E register where they are held until the processor transfers the information to the scratch pad (S) register.

A fixed voltage of +5 volts on the input of the E address register is not buffered, and immediately after INCRL, sets the jump instruction flip-flop JINLD (input loaded). The processor program acknowledges the jump selector input JINLD and produces the SEBEN signal, which enables the data in the E register to be transferred into the S register location, as specified by the E address, SELB0 - 2.

The drive address (or drive select) is located in address 0 of the S register, a list of the instruction and command locations in the scratch pad is given in the S register location table.

When the information is in the scratch pad register the processor program transfers it to the FD register via the internal data bus and into the diskette control logic. A reset data input signal, RESDI, is produced by the program to clear the E register and allow a further input instruction or command to take place.

## COMMANDS (Waveform 2)

Commands to the diskette are given only when certain instructions have already been carried out. The only exception to this condition is the RECAL command which sets the track to zero.

The function code 37 is decoded from the address bus AB0 - 7 and initiates signals COMOR and the low clock pulse INRCL.

The command on the data bus is then clocked into the E register together with address bits AB0 - 2 and command signal COMOR.

Both jump instruction flip-flops, JINLD and JCOLD are set, which inform the processor software that a command has been loaded into the E register. The processor then initiates SEBEN and transfers the information to the S register for further processor operations. The JCOLD signal is also fed to the input of the interface decoding logic gate. The effect of JCOLD going high is to inhibit the input register clocking signal INRCL. This ensures that no further setting or resetting of JCOLD and JINLD could occur until after the command has been processed. The reset data input signal, RESDI, is produced by the processor program to clear the E register and allow a further instruction or command to take place.

## STATUS TRANSFER (W

The status byte is transfer either an interrupt reques or when demanded by the the main computer can oc independent of the proces

After each instruction or ted by the processor prog bus is set high and the ger produces a low IREQ6 sig This signal remains low ur computer with IACK6.

The main computer place address bus AB0 - 7 and s The signals are decoded to which loads the contents data bus. Prior to the data the computer disables the status has priority on inpu

## ADDRESS HIGH AND LOW REGISTERS

Addresses the main computer memory during DMA transfers.

## OUTPUT DATA REGISTER

Transfers data to the main computer memory during a DMA Read mode.

## STATUS AND INTERRUPT REGISTERS

The status register is an 8-bit store containing the follow

| | |
|---|---|
| Bit 0 = Busy | Bit 4 = Drive not ready |
| Bit 1 = Sector missing | Bit 5 = Non-valid comma |
| Bit 2 = CRC error | Bit 6 = Deleted record |
| Bit 3 = No address mark | Bit 7 = Operation compl |

Each bit is set to logic 1 by the processor program as an occurs. When one or more bits are set, the processor also signal. The interrupt priority is level 6, therefore, being interrupt may wait for at least 20ms before being servic

When both CRC Error and No Address Mark are set, the error during the read (or read before write) operation.

When both CRC Error and Sector Missing are set the ste error during data transfer between the main computer a (DMA operation).

### DRIVE SELECTION AND CONTROL REGISTER

Logic control signals fed to the drive electronics and positioning control as specified during the Controlled Instruction and Command Transfers.

### WRITE MODE (Waveform 5)

The write mode is when data is read from the main computer and written on to the diskette.

The DMA request occurs when the processor program is ready to receive data from the main computer memory location, already specified by the address high and low register bytes. The addresses are incremented by the processor each time a data transfer takes place so that a new location is sent to the computer memory for every byte until the complete block is transferred. The start addresses are initially sent to the processor during the controlled transfer instructions and stored in the S register until required for the DMA transfer.

Then follows the same sequence as in the read mode when DR2 informs the computer that a DMA request is required. The computer acknowledges this with the GRANT and DMAPLS interface signals. Since data is to be written to the diskette, the WMODE signal is high which enables the "write from main memory" gate. This gate produces interface signal MIN and the input register clock, INCRL.

The low MIN signal places data from the main memory location, specified by the high and low address registers, on to the interface data bus. The data is then clocked into the E register when the low INRCL signal is set high at the end of the DMA request. This occurs when the interface signals GRANT and DMAPLS, are automatically reset in a time to ensure that reading from the main memory is complete.

When the E register is loaded by INRCL, the +5 logic level on the E address register sets the input loaded flip-flop and produces JINLD. The processor software then resets the E address register to zero, because input data during the DMA transfer must be placed in address location 0 in the S register. The SEBEN pulse is then set low by the processor and clocks the data into the S register to complete the DMA transfer.

### STATUS TRANSFER (Waveform 3)

The status byte is transferred to the main computer when either an interrupt request is set by the processor program or when demanded by the main computer. A demand by the main computer can occur at any time and be quite independent of the processor program.

After each instruction or command an interrupt is requested by the processor program. Bit 7 on the internal data bus is set high and the general clock mode pulse GENCL produces a low IREQ6 signal from the interrupt flip-flop. This signal remains low until acknowledged by the main computer with IACK6.

The main computer places the function code 37 on the address bus AB0 - 7 and sets interface signal IOIN low. The signals are decoded to produce the low signal OSTEN which loads the contents of the status register on to the data bus. Prior to the data being clocked from the register, the computer disables the data bus to ensure that the status has priority on input to the computer.

### THE EXTERNAL (E) REGISTER

The E register is formed using 3 separate registers. The data register (U8) clocks in data from the bus which is then held until enabled on to the internal data bus. The address register (U2) is a 'D' type register which clocks in the address and is then immediately available at the register output. However, the address is prevented from being enabled on to the internal address bus by the 3-state buffer register (U11). This buffer register transfers the address bits when the data register is enabled by SEBEN.

The existence of the buffer register (U11) allows the COMOR and input loaded (+5V) signals to be clocked directly to the jump instruction flip-flops to inform the processor software that data has been loaded into the E register.

### STATUS AND INTERRUPT REGISTERS

The status register is an 8-bit store containing the following information:

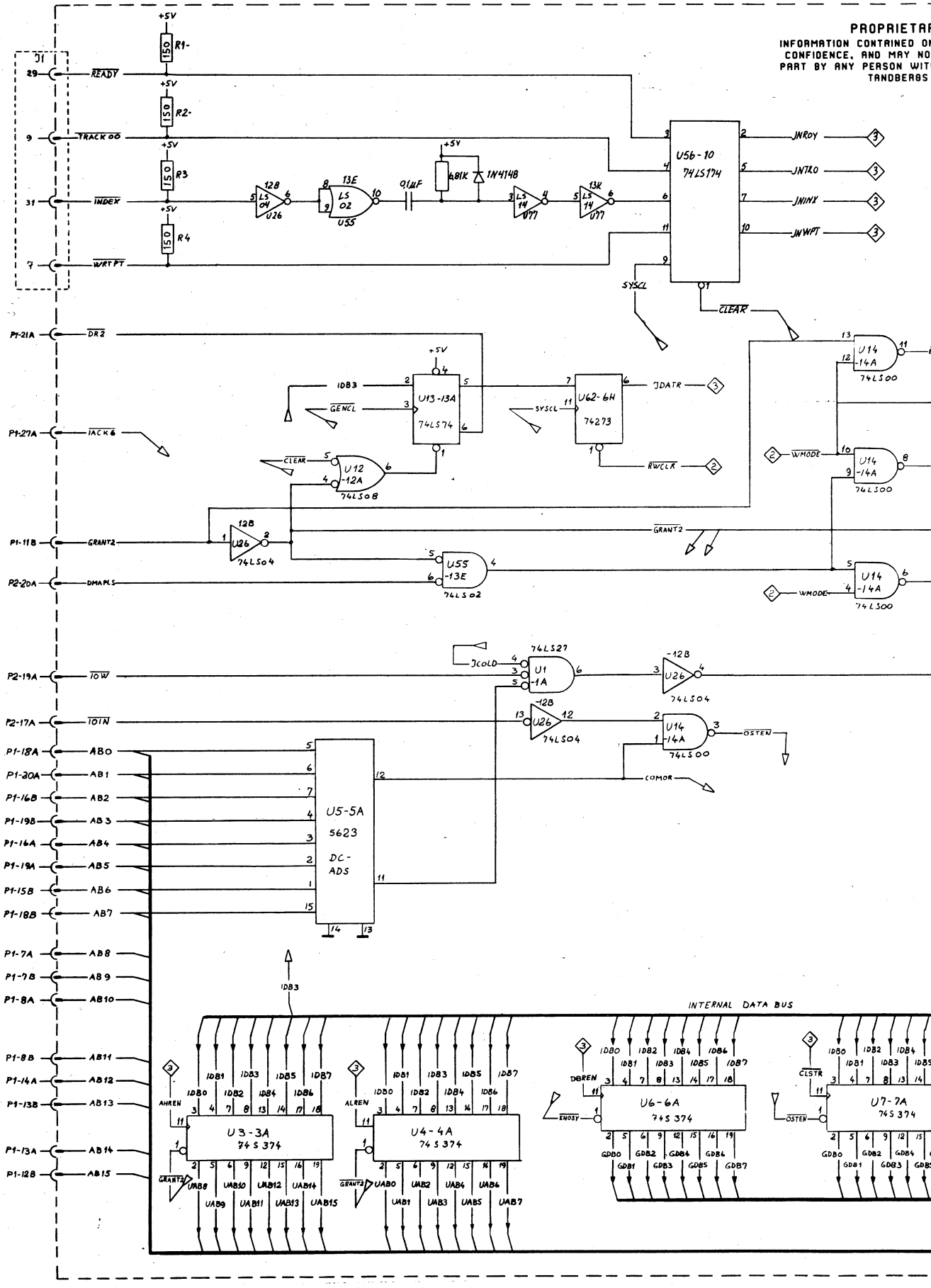
| | |
|---|---|
| Bit 0 = Busy | Bit 4 = Drive not ready |
| Bit 1 = Sector missing | Bit 5 = Non-valid command |
| Bit 2 = CRC error | Bit 6 = Deleted record |
| Bit 3 = No address mark | Bit 7 = Operation completed |

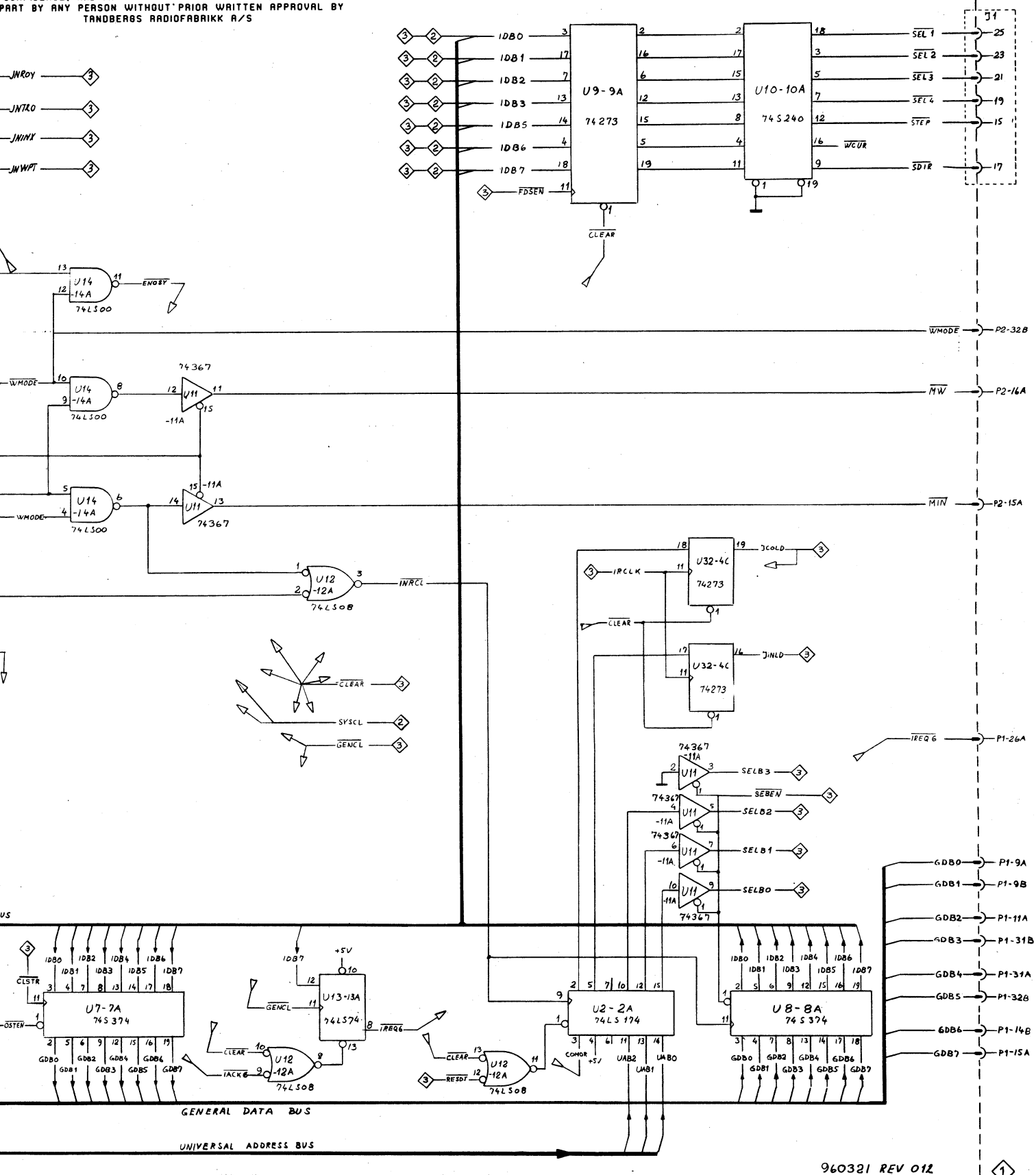
Each bit is set to logic 1 by the processor program as and when the error or function occurs. When one or more bits are set, the processor also sets the interrupt request signal. The interrupt priority is level 6, therefore, being a low priority rating, the interrupt may wait for at least 20ms before being serviced.

When both CRC Error and No Address Mark are set, the status is defined as a timing error during the read (or read before write) operation.

When both CRC Error and Sector Missing are set the status is defined as a timing error during data transfer between the main computer and the diskette controller (DMA operation).

INPUT / OUTPUT — Description





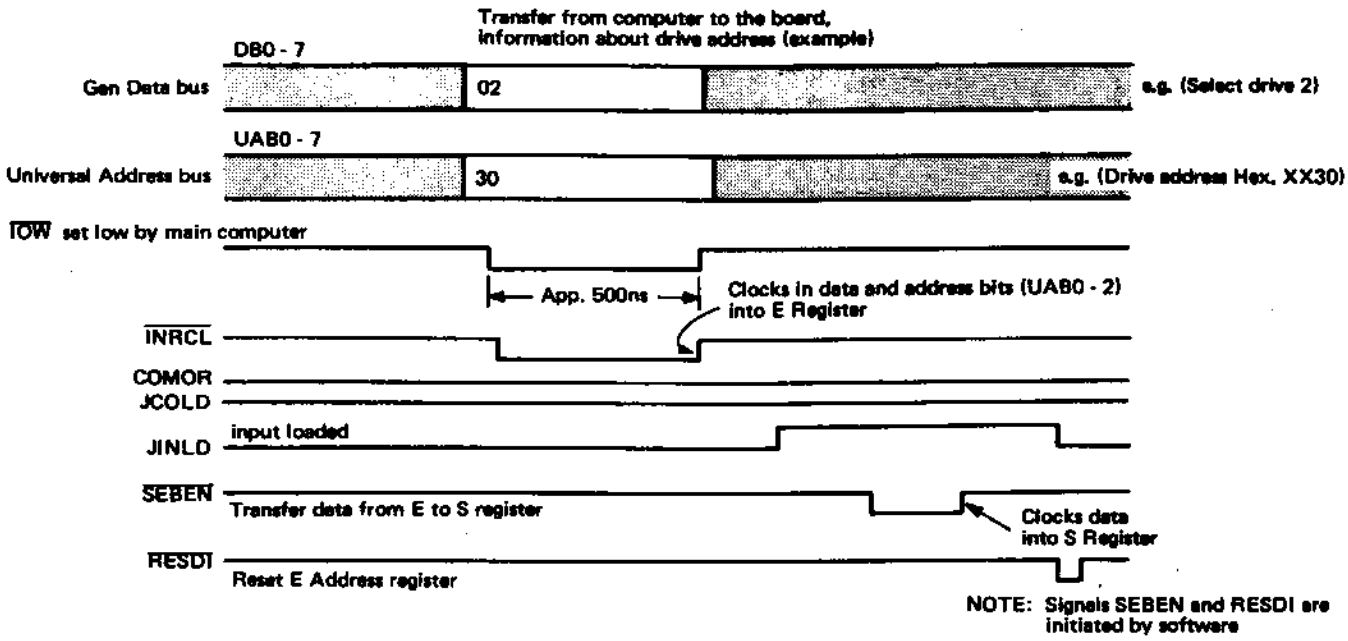

960321 REV 012

DISKETTE CONTROLLER IN/OUT

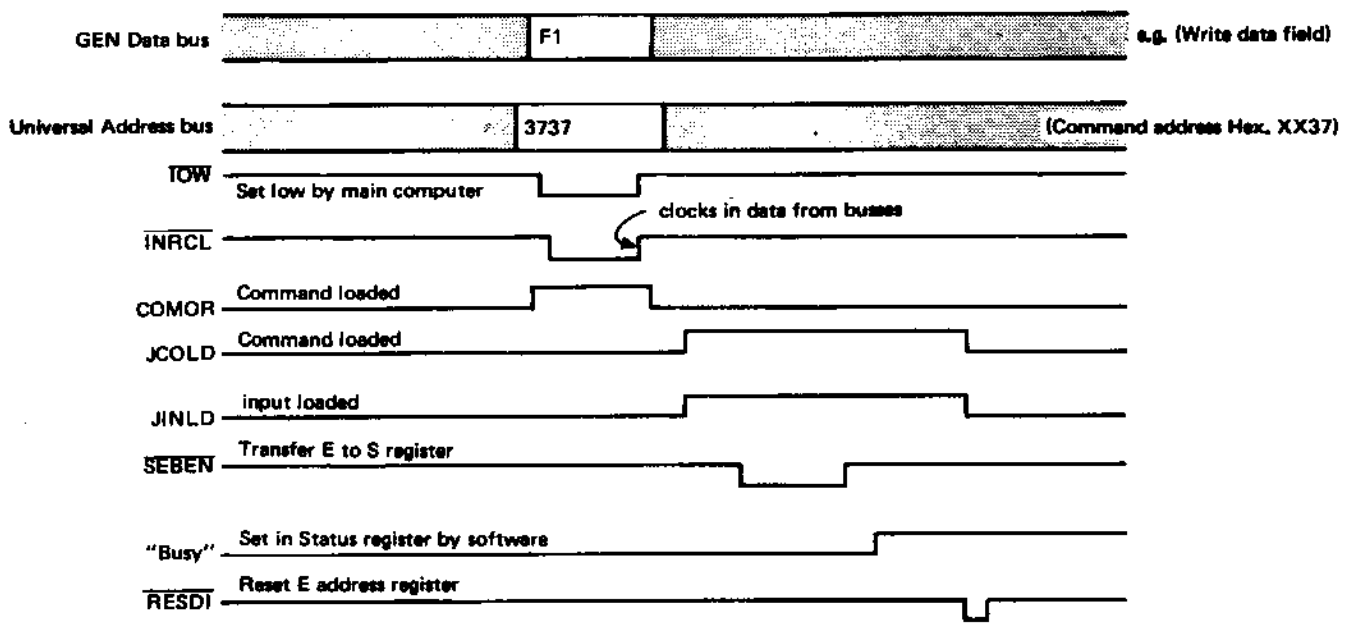
Proj. 296
SCHEMATIC 1 OF 3
4-2-77 BOB
19-9-78 FA

92942-1

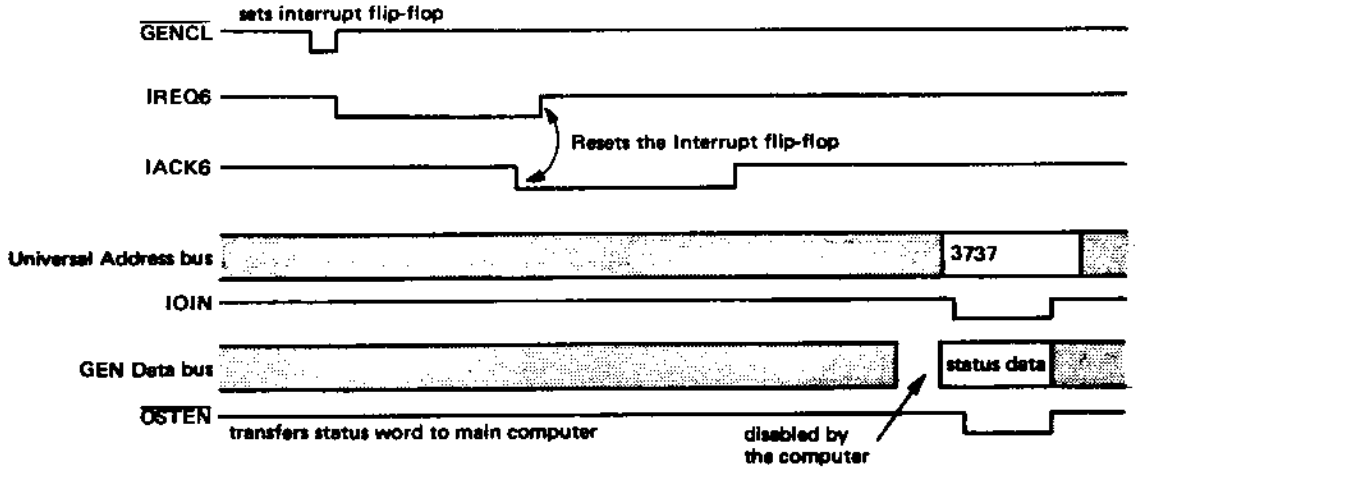
## Controlled transfer Waveform 1 Transfer Instruction




## Controlled transfer Waveform 2 Transferring a Command

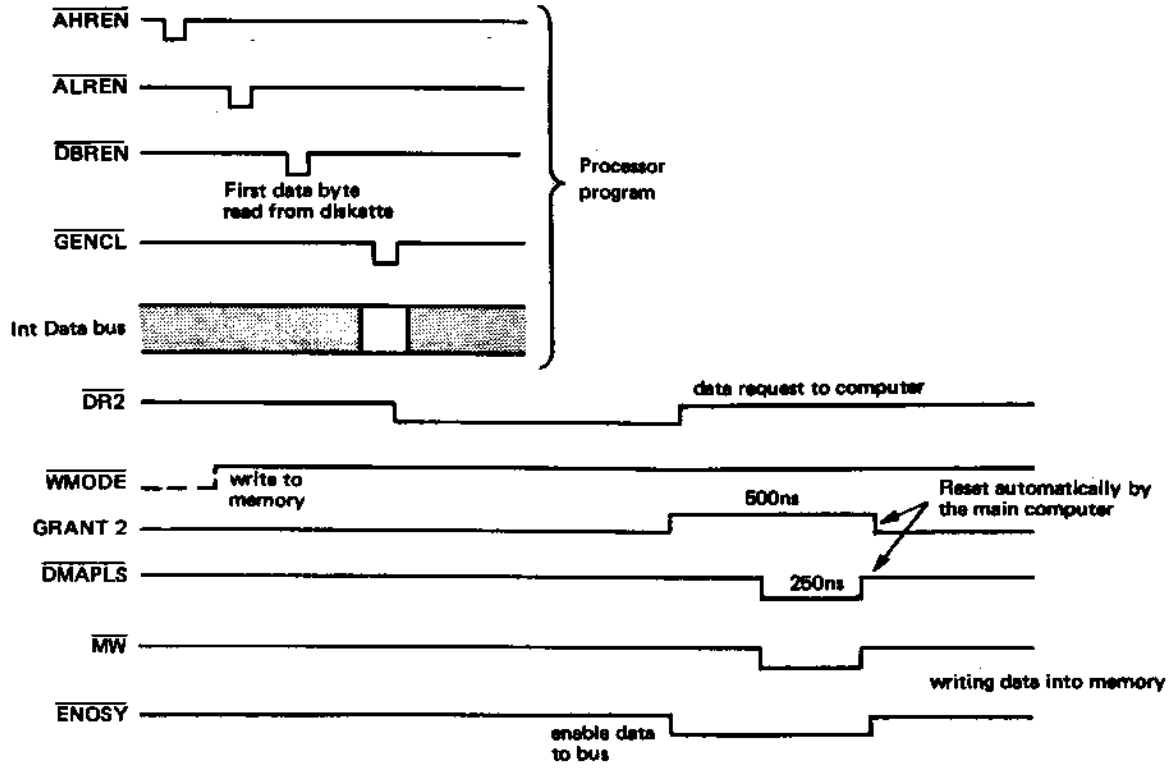



## Controlled transfer Waveform 3 Transferring Status

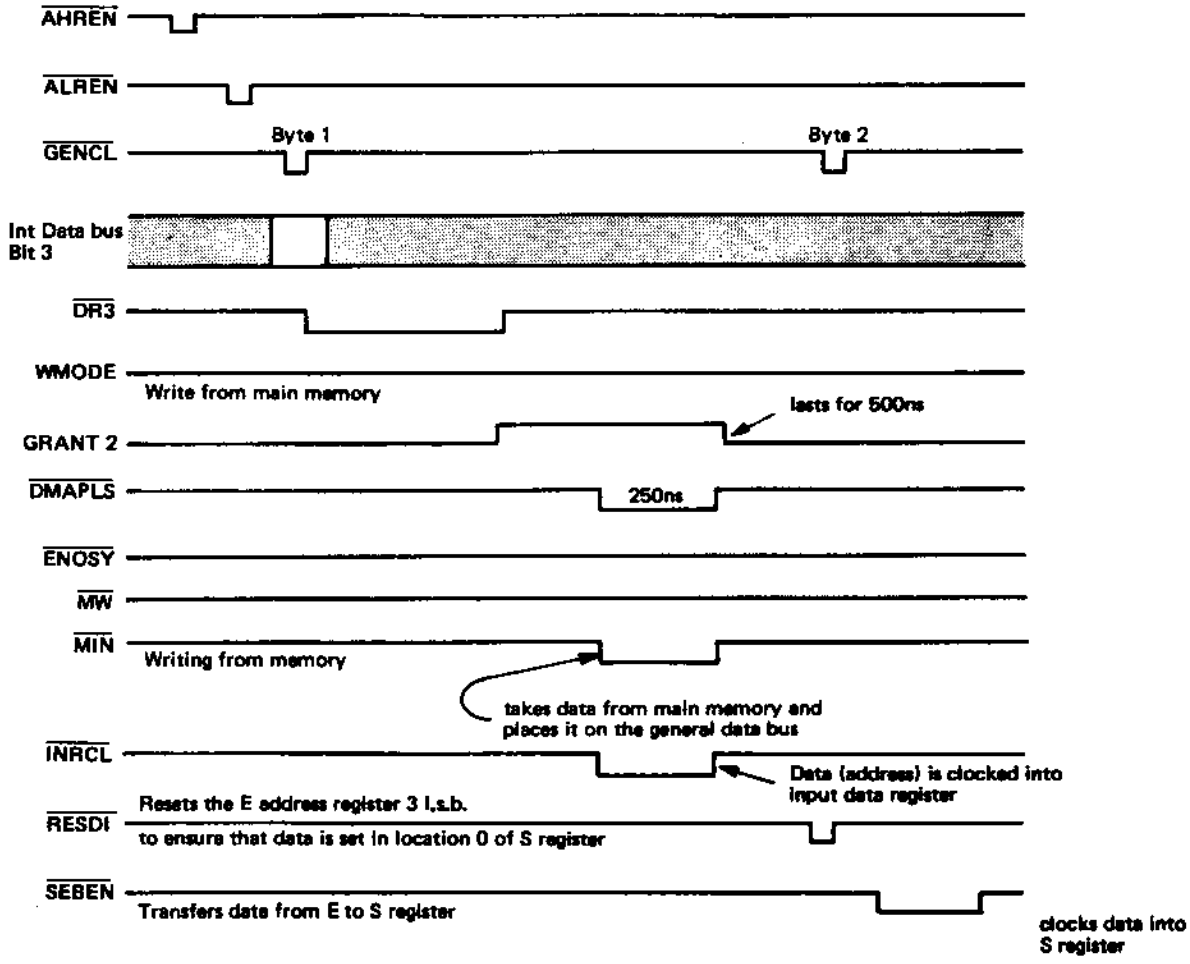



TDV 2114
1426 - 10 - 78

**DMA Transfer Waveform 4 Data from Diskette to Main Memory**

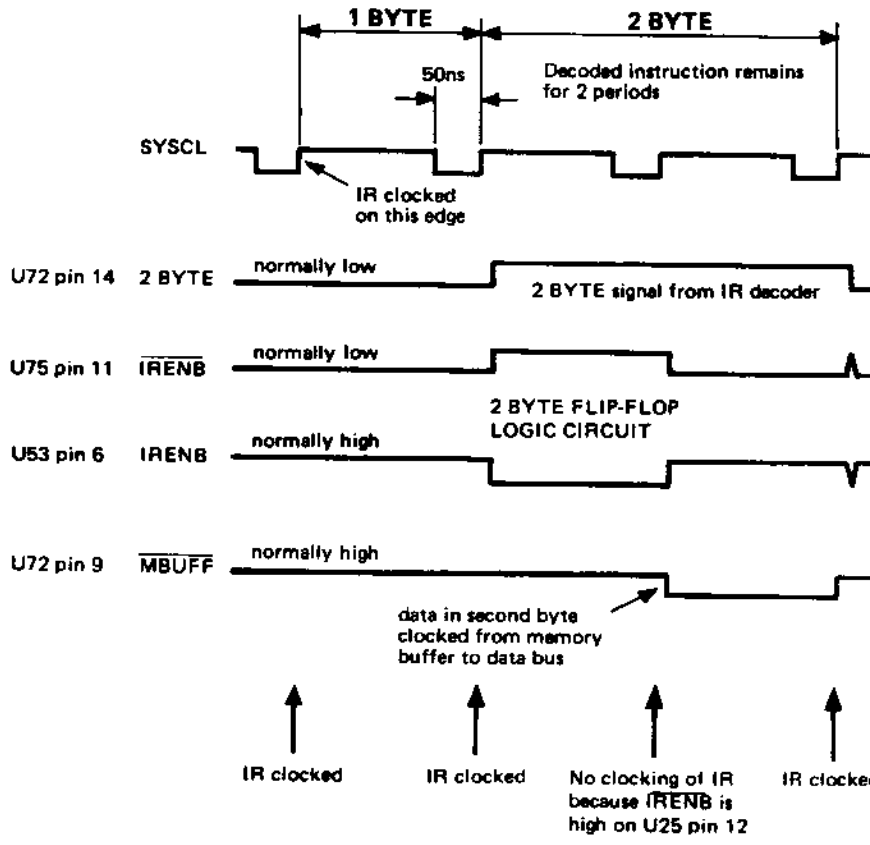



**DMA Transfer Waveform 5 Main Memory to Diskette**




TDV 2114
1428 - 10 - 76

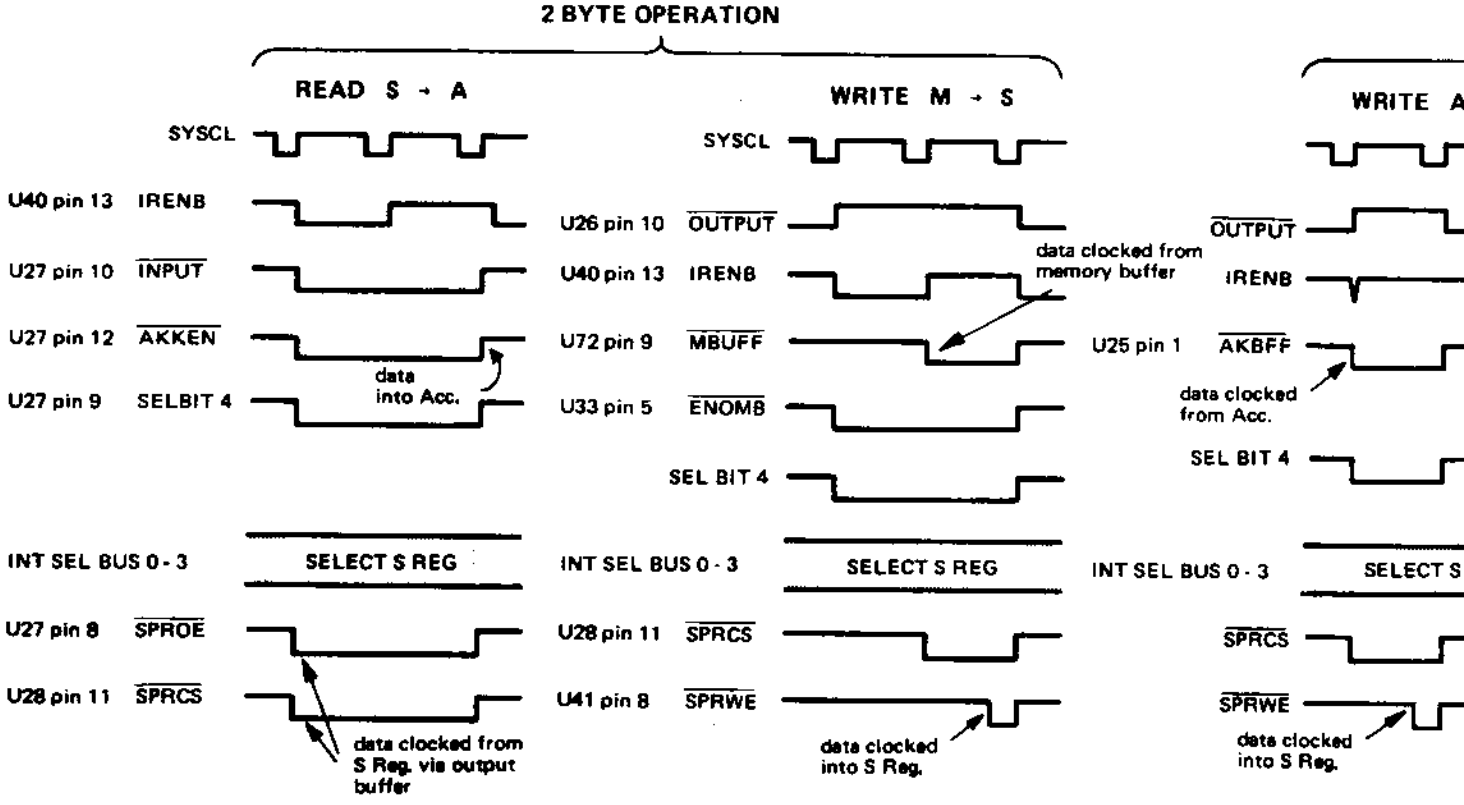
**Waveform 1 2-Byte operation**

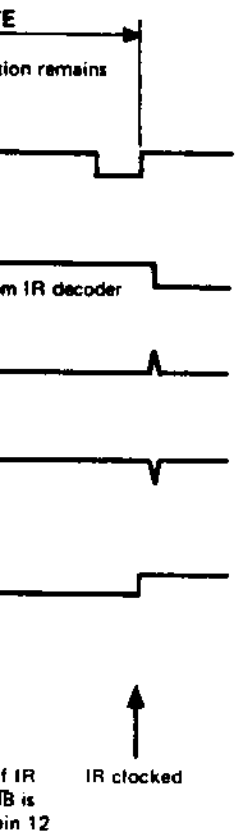



1426-02-79

**Waveform 2 Scratch Pad Clocking**

**SCRATCH PAD REGISTER WAVEFORMS**
**(into and from register decoder gates)**






RMS
)

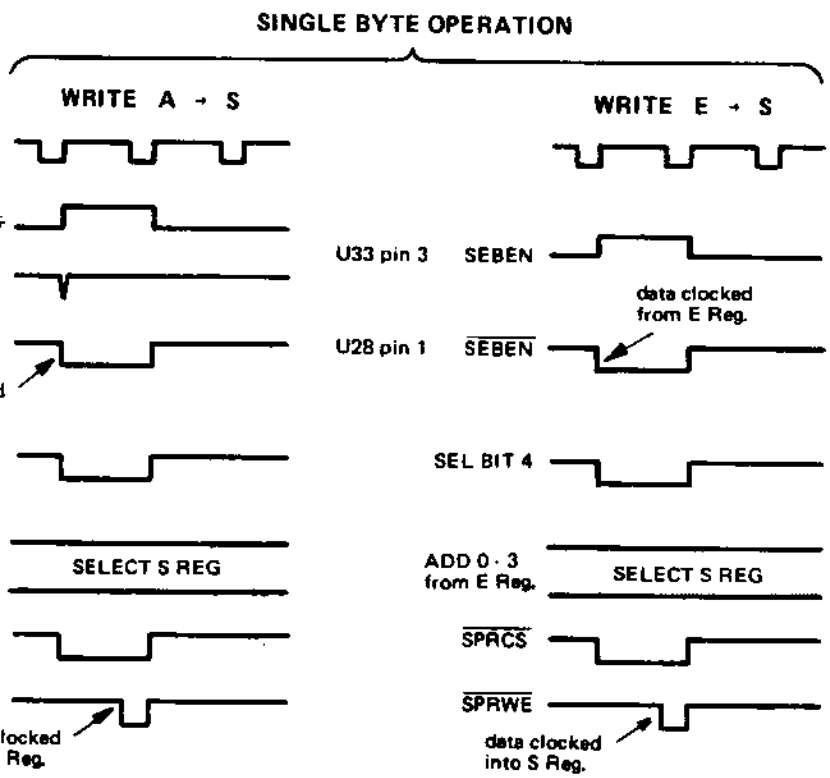



**Table 1**

**IR Decoder Version 5**

| No. | Pin 14 IRBIT4 | Pin 13 IRBIT5 | Pin 12 IRBIT6 | Pin 11 IRBIT7 | Pin 10 | Pin 9 AKBFE | Pin 7 INPUT | Pin 6 AKKEN | Pin 5 ENOMB | Pin 4 OUTPT | Pin 3 SEBEN | Pin 2 JUMPE | Pin 1 2BYTE | Hex not. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | FA |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 72 |
| 2 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | DA |
| 3 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | E3 |
| 4 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | FA |
| 5 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | FE |
| 6 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | CA |
| 7 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 9B |
| 8 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | EB |
| 9 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 72 |
| 10 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | CB |
| 11 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | E3 |
| 12 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | FA |
| 13 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | F9 |
| 14 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | CA |
| 15 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 9B |
| 16 | | | | | | | | | | | | | | |
| 17 | | | | | | | | | | | | | | |
| 18 | | | | | | | | | | | | | | |
| 19 | | | | | | | | | | | | | | |
| 20 | | | | | | | | | | | | | | |
| 21 | | | | | | | | | | | | | | |
| 22 | | | | | | | | | | | | | | |
| 23 | | | | | | | | | | | | | | |
| 24 | | | | | | | | | | | | | | |
| 25 | | | | | | | | | | | | | | |
| 26 | | | | | | | | | | | | | | |
| 27 | | | | | | | | | | | | | | |
| 28 | | | | | | | | | | | | | | |
| 29 | | | | | | | | | | | | | | |
| 30 | | | | | | | | | | | | | | |
| 31 | | | | | | | | | | | | | | |

**Table 2**

**Instruction Decoder**

| Operation | | Byte 1 | Byte 2 | IR Decoder Output | Comments |
|---|---|---|---|---|---|
| Operate M, A → A | 1 | 00 ALUSEL | Data (M7 – M0) | (2 BYTE), (ENOMB) | No clocking of acc. Only 1 byte operation when possible |
| Operate M, A → A | 2 | 00 ALUSEL | Data (M7 – M0) | AKKEN, (2 BYTE), (ENOMB) | Clocking of acc. Only 1 byte operation when possible |
| Operate D, A → A | 3 | 111 D4 → D0 | XXALUSEL | AKKEN, INPUT 2 BYTE | |
| Transfer M → D | 4 | 110 D4 → D0 | Data M7 – M0 | OUTPUT, 2 BYTE ENOMB | |
| Transfer A → D | 5 | 100 D4 – D0 | | AKBFE, OUTPUT | |
| Transfer E → S | 6 | 1010X S2 – S0 | | SEBEN | |
| Jump (un)conditionally | 7 | 1011 $C_3 - C_0$ | Jump Address M7 – M0 | JUMPE, 2 BYTE | |

**Table 3**

**Output Addresses**

| | |
|---|---|
| COMCL | A → co |
| COMCL | M → co |
| AHREN | A → A |
| AHREN | M → A |
| ALREN | A → A |
| ALREN | M → A |
| DBREN | A → D |
| DBREN | M → D |
| CLSTR | A → St |
| CLSTR | M → St |
| FDSEN | A → FI |
| FDSEN | M → FI |
| WDREN | A → W |
| WDREN | M → W |
| WDREN | A → W |
| WCREN | M → W |
| | A → S |
| | M → S |
| | IE → S |
| GENCL | A → Ge |
| GENCL | M → Ge |

| COMCL 7 | 6 | |
|---|---|---|
| SYNCR | Write Data 1 | W Da |
| SYNCR | Write Data 1 | W Da |
| GENCL | | |
| R2 | | |
| R2 | | |

**Table 6**

**Scratch Pad Function Signals — S-Register**

| Function | Data Transfer | Single Byte Operation | | 2-Byte Operation | |
|---|---|---|---|---|---|
| | | From IR Register | S Register Enable/ CLK signal | From IR Register | S Register Enable/ CLK signal |
| WRITE | Accumulator to S Register | OUTPUT AKBFE SELBIT4 (low) Address bits, SELB0-3 | SPRCS SPRWE | | |
| WRITE | Memory to S Register | | | 2 BYTE OUTPUT ENOMB SELBIT4 (low) Address bits, SELB0-3 | SPRCS SPRWE |
| WRITE | E Register to S Register | SEBEN SELBIT4 (low) Address bits 0 - 3 from E Register | SEBEN SPRCS SPRWE | | |
| READ | S Register to Accumulator | | | 2 BYTE INPUT AKKEN Address bits SELB0-3 | SPRCS SPROE |

**Table 3**

**Output Addresses (CPU → D)**

| | | |
|---|---|---|
| COMCL | A → combined clock | 10010000 |
| $\overline{\text{COMCL}}$ | M → combined clock | 11010000 |
| AHREN | A → Address High Reg. | 10010001 |
| $\overline{\text{AHREN}}$ | M → Address High Reg. | 11010001 |
| ALREN | A → Address Low Reg. | 10010010 |
| $\overline{\text{ALREN}}$ | M → Address Low Reg. | 11010010 |
| DBREN | A → Data Bus out Reg. | 10010011 |
| $\overline{\text{DBREN}}$ | M → Data Bus out Reg. | 11010011 |
| CLSTR | A → Status Reg. | 10010100 |
| $\overline{\text{CLSTR}}$ | M → Status Reg. | 11010100 |
| FDSEN | A → Floppy Select + current Reg. | 10010101 |
| $\overline{\text{FDSEN}}$ | M → Floppy Select + current Reg. | 11010101 |
| WDREN | A → Write Data Reg. | 10010110 |
| $\overline{\text{WDREN}}$ | M → Write Data Reg. | 11010110 |
| WDREN | A → Write Clock Reg. | 10010111 |
| $\overline{\text{WCREN}}$ | M → Write Clock Reg. | 11010111 |
| | A → S | $1000S_3S_2S_1S_0$ |
| | M → S | $1100S_3S_2S_1S_0$ |
| | (E → S) | $1010XS_2S_1S_0$ |
| GENCL | A → General clock | 10011XXX |
| $\overline{\text{GENCL}}$ | M → General clock | 11011XXX |

| COMCL | Byte 2 | | | | | | |
|---|---|---|---|---|---|---|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SYNCR | Write Data 1 | Write Data 2 | Read Data 1 | Read Data 2 | PSEL2 | 7CENB | PSEL1 |
| SYNCR | Write Data 1 | Write Data 2 | Read Data 1 | Read Data 2 | PSEL2 | 7CENB | PSEL1 |
| GENCL | | | | | | | |
| R2 | | | | set DR | $R_1$ | Int Reg | |
| $R_2$ | | | | set DR | $R_1$ | Int Reg | |

**Table 4**

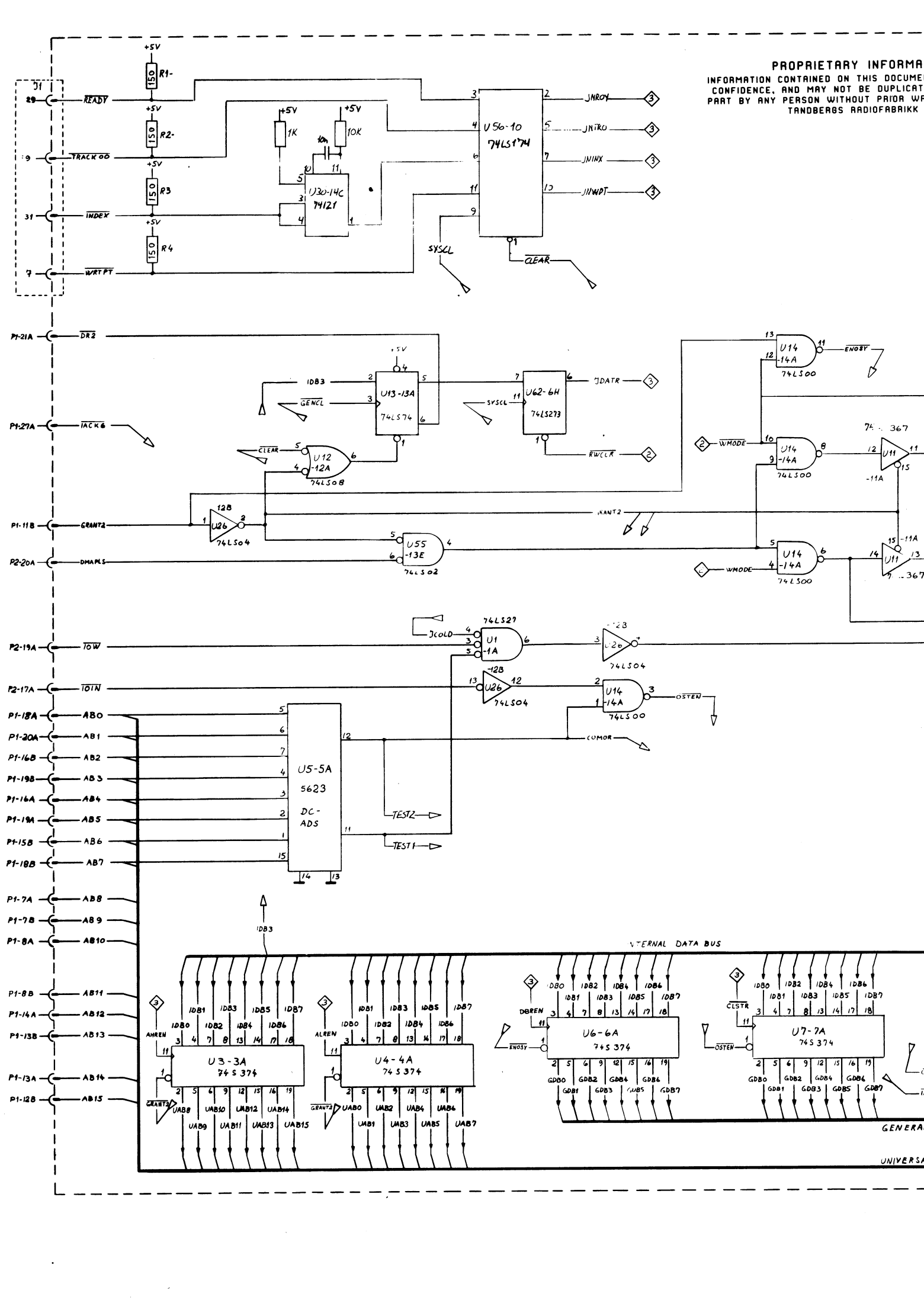
**Input Addresses (D → CPU)**

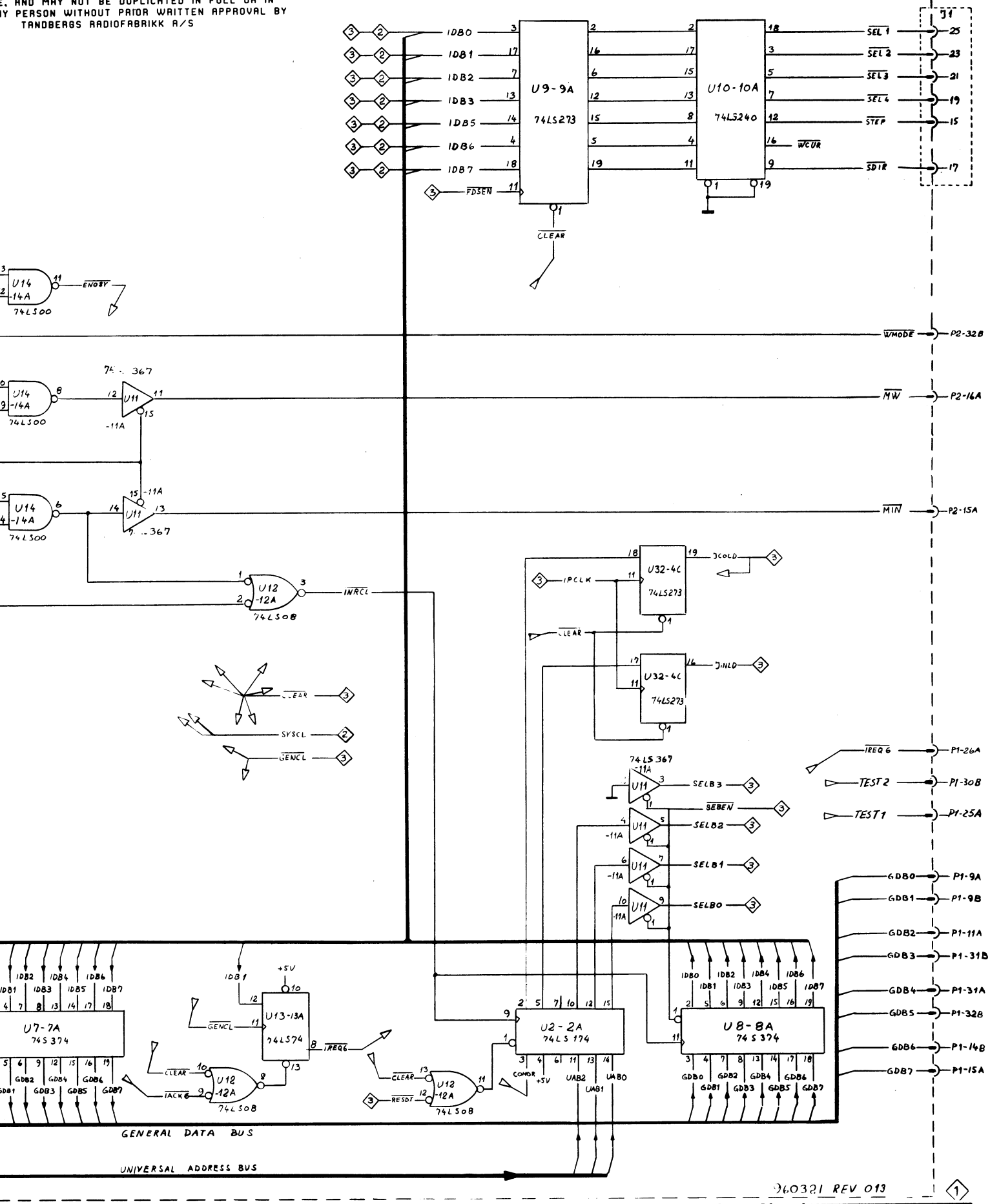
| | | |
|---|---|---|
| $\overline{\text{RCREN}}$ | Clock Register → A | 1111XXX0 |
| $\overline{\text{RDREN}}$ | Data Read Reg → A | 1111XXX1 |
| | S → A | $1110\,S_3S_2S_1S_0$ |

**Table 5**

**Scratch Pad Register Locations**

| Location | Contents |
|---|---|
| S0 : | Drive Address<br>Data during write operation |
| S1 : | Relative track Address<br>Direction of head movement |
| S2 : | Sector Address |
| S3 : | The most significant byte of the main memory address |
| S4 | The least significant byte of the main memory address |
| S5 : | Block length<br>Clock data during write format operation |
| S6 : | This register contains information to let the software know if the correct header has been read during a read data field operation |
| S7 : | Command register |
| S8 : | Device No., step direction, step pulse |
| S9 & SA : | Counters to count the number of sync. errors and retries. Only S9 is employed with read operation between header and main block. SA is also employed as a general purpose register. |
| SB : | Counts missing sectors etc. Also employed during seek/recalibrate and write format operation |
| SC : | Contains the block length in steps of 128. (0 = 128, 1 = 256, 3 = 512) |
| SD : | Storage for counting index-strobes. Also employed as a general storage register during write operation |
| SE : | Contains the correct header to be written |
| SF : | Contains the information which is going to be sent to the status register |

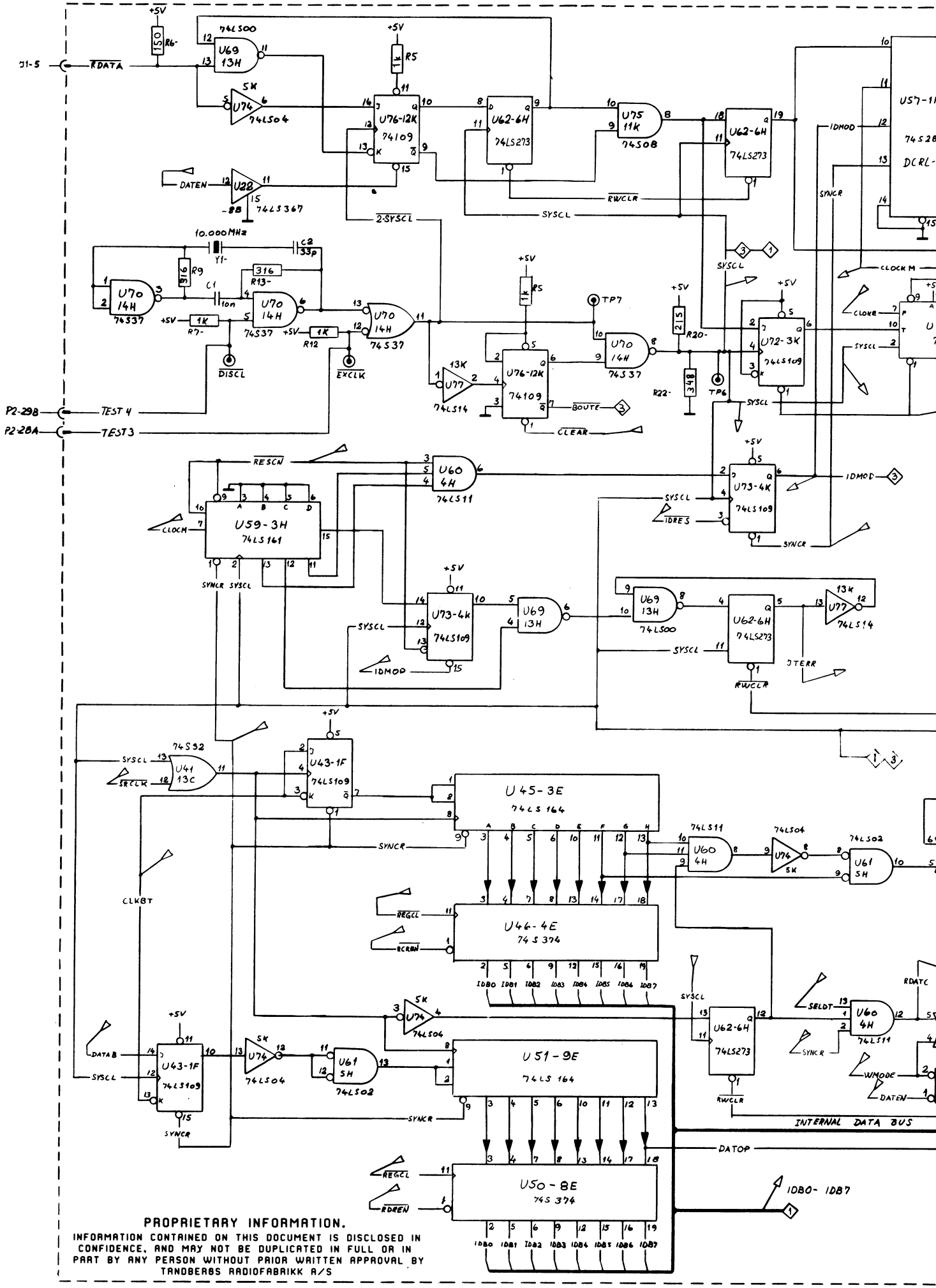


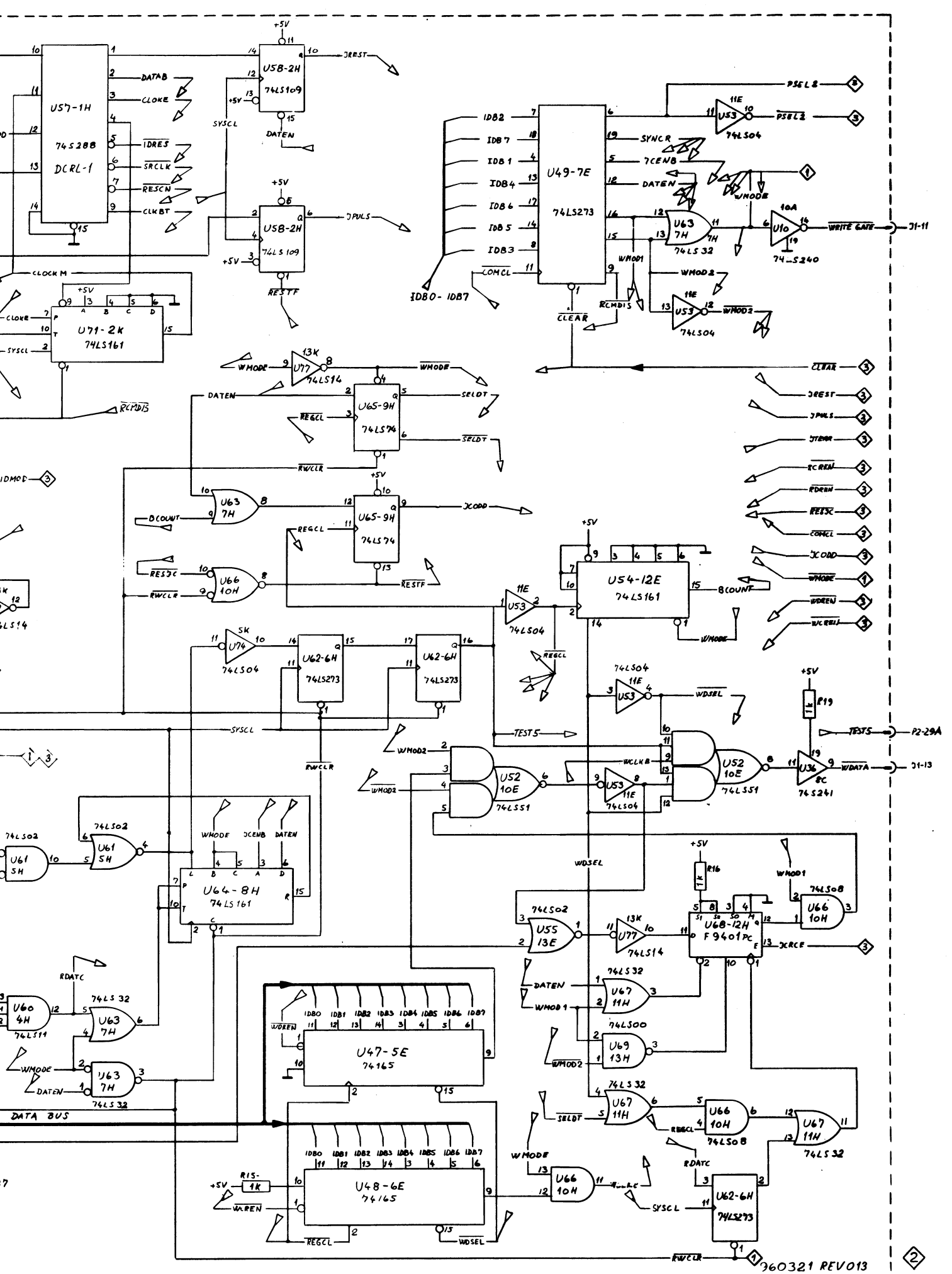



960321 REV 013

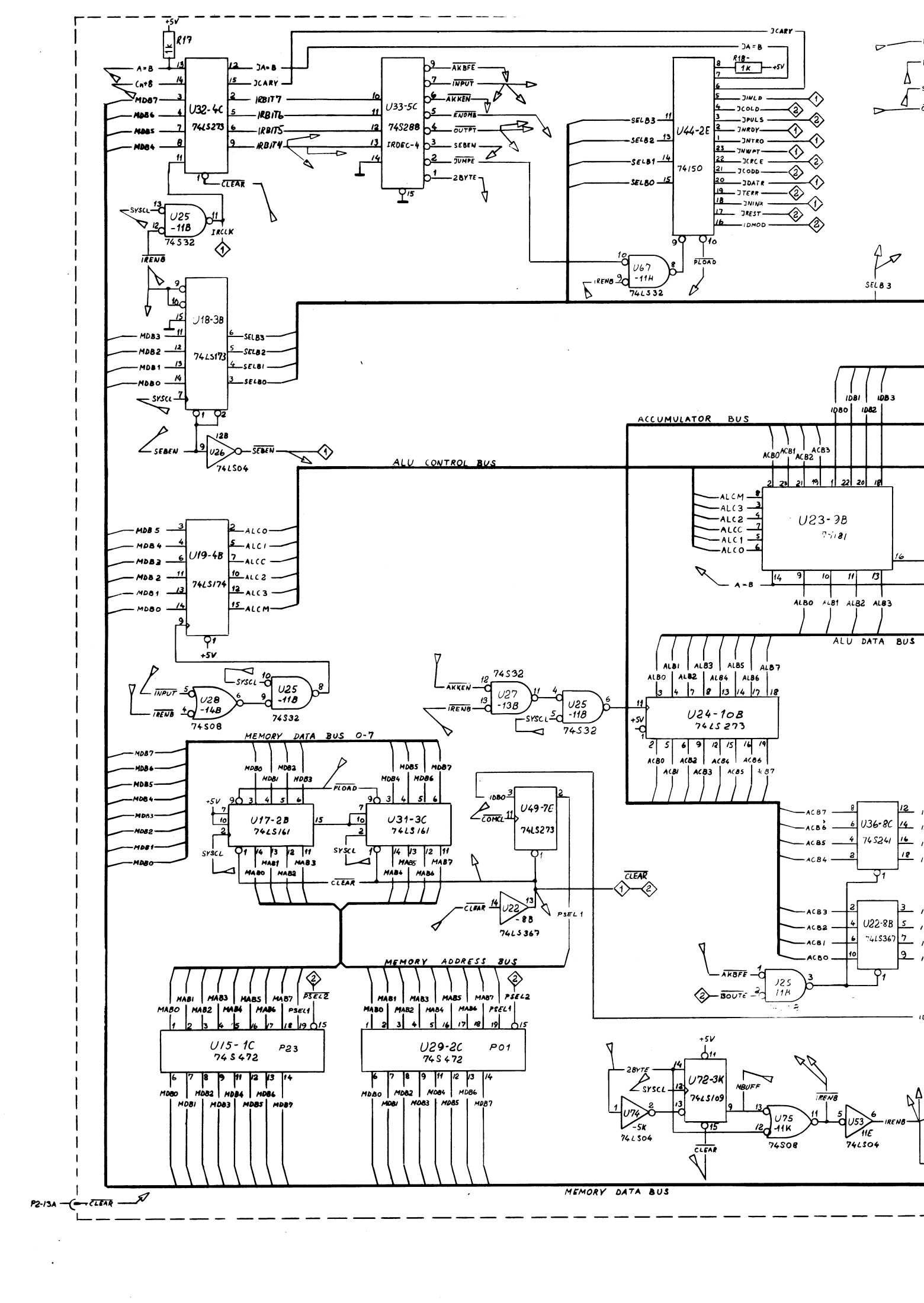
| | | | | | |
|---|---|---|---|---|---|
| | Use: DISKETTE CONTROLLER | Part name: SCHEMATIC | | Position: A1 | Sheet no.: 1-3 |
| 24.4.79 | Model no.: TDV2114 | Part no.: (Ordering no.) 960321 | Rev. level: 0-13 | Drawings no.: 92942 | Corr. no.: 2 |

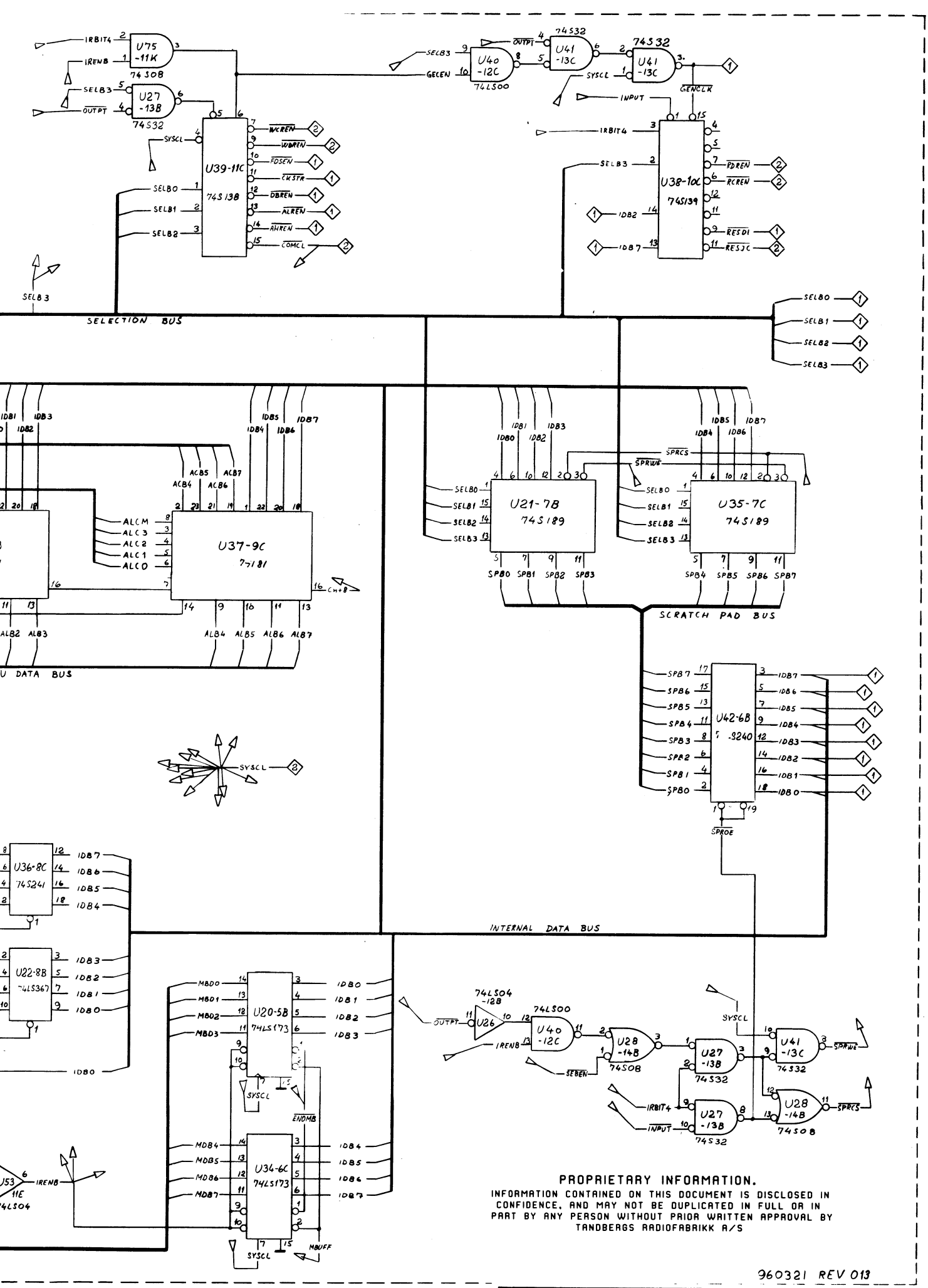
TANDBERG DATA — Tlf. 232080 – Boks 9, Korsvoll – Oslo 8






| | | | | | | |
|---|---|---|---|---|---|---|
| | | Use: DISKETTE CONTROLLER | Part name: SCHEMATIC | | Position: A1 | Sheet no.: 2 - 3 |
| | | Model no.: TDV 2114 | Part no. (Ordering no.): 960321 | Rev level: 0 - 13 | Drawing no.: 92942 | Corr. no.: 2 |
| 19.9.78 FØ | | TANDBERG DATA | | | TIf. 232000 – Boks 9, Kjørvoll – Oslo 8 | |


ACCUMULATOR BUS
ALU CONTROL BUS
ALU DATA BUS
MEMORY DATA BUS 0-7
MEMORY ADDRESS BUS
MEMORY DATA BUS
U32-4C 74LS273
U33-5C 74S288
U44-2E 74150
U67 -11H 74LS32
U18-3B 74LS173
U19-4B 74LS174
U23-3B 74181
U24-10B 74LS273
U25 -11B 74S32
U26 74LS04
U27 -13B 74S32
U28 -14B 74S08
U17-2B 74LS161
U31-3C 74LS161
U49-7E 74LS273
U22 -8B 74LS367
U36-8C 74S241
U22-8B 74LS367
U15-1C 74S472 P23
U29-2C 74S472 P01
U72-3K 74LS109
U74 -5K 74LS04
U75 -11K 74S08
U53 -11E 74LS04
R17 1K
R18 1K
P2-13A CLEAR

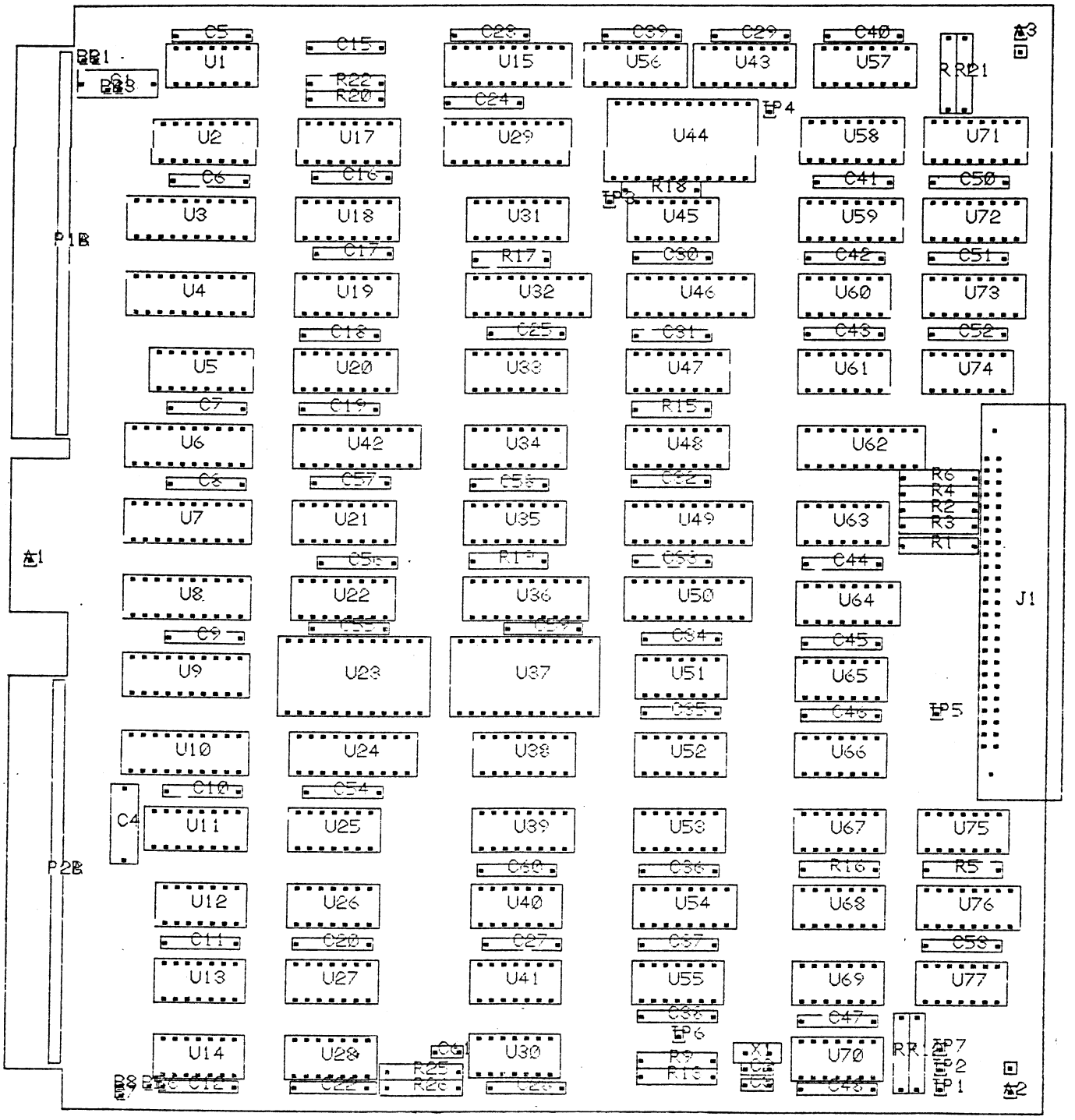





960321 REV 013

| | | | | | | |
|---|---|---|---|---|---|---|
| | Use: DISKETTE CONTROLLER | Part No.: SCHEMATIC | | | Position: A1 | Sheet no.: 3-3 |
| 24.4.79 Fa | Model no: TDV 2114 | Part no (Ordering no.): 960321 | Rev level: 0-13 | Drawing no.: 92942 | | Corr no.: 2 |

TANDBERG DATA — Tlf. 232080 – Boks 9, Korsvoll – Oslo 8




Board seen from component side

## READ MODE

## Read Mode

The diskette is set to the read mode after a sequence of instructions and commands have been carried out between the controller and the main computer. The sequence selects the drive, sets the recording head to the correct track, informs the program which part of the track is to be read and where the information is to be stored in the main computer.

Raw data is read from the diskette when the head is loaded and the Write Gate (J1 - 11) is at a high level. The program then enters the read synchronising mode and all subsequent actions are mainly software controlled.

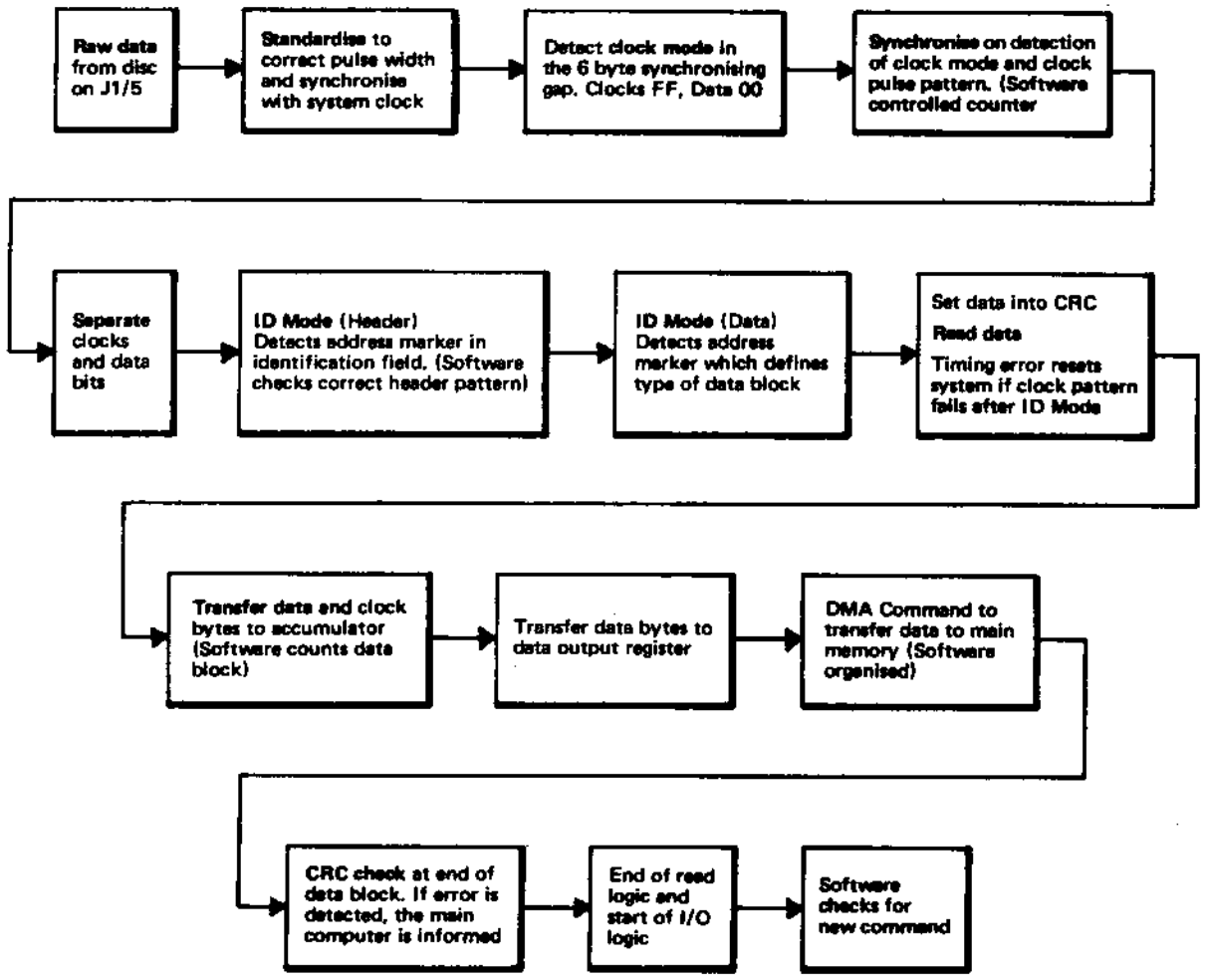
The read sequence is described briefly in the following flow schematic and can also be identified by referring to the block descriptions of the logic diagrams.

The header format in the following diagram shows the read mode sequence as it occurs on the track.

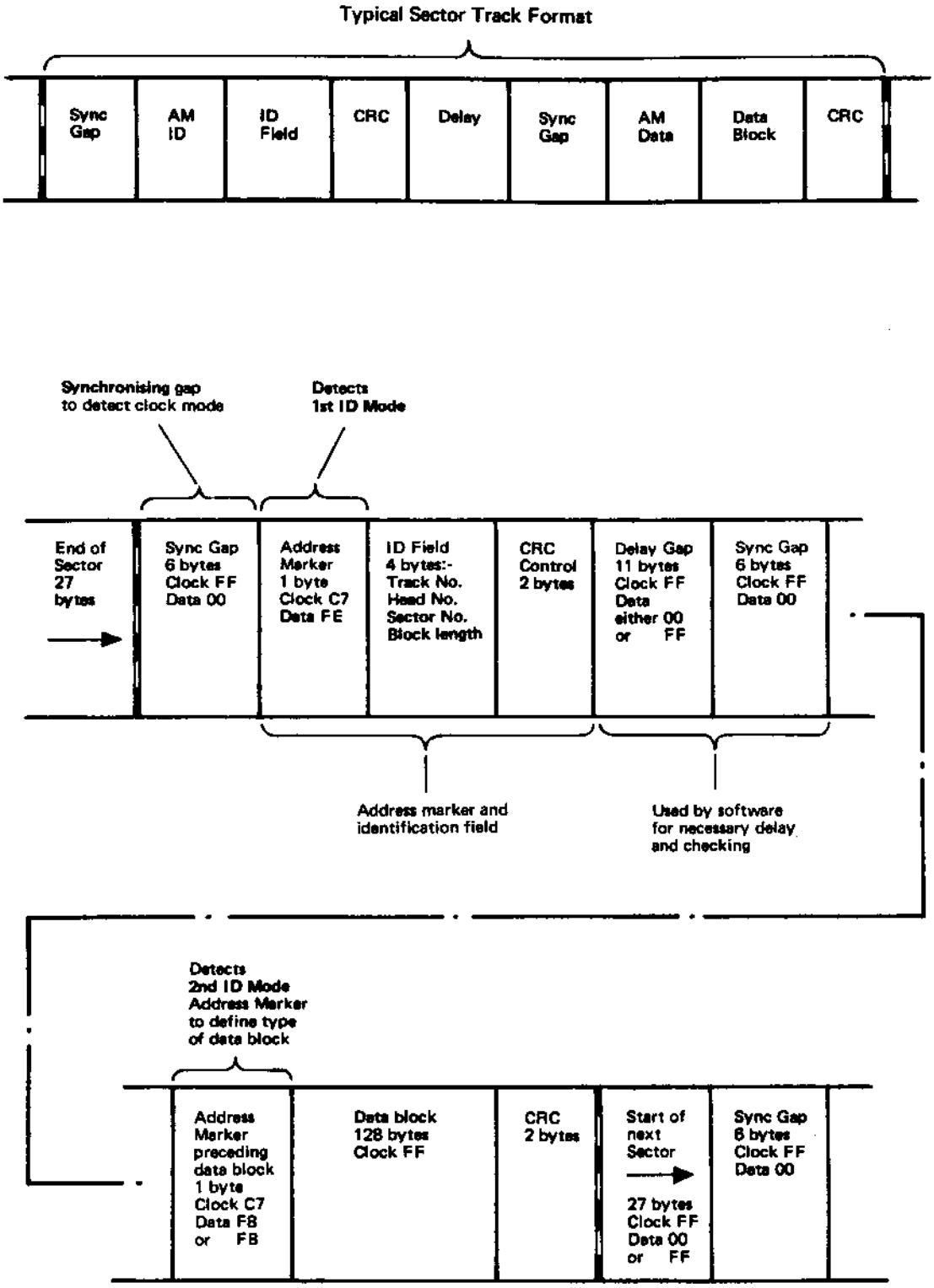
The signals shown on the read mode block diagram are those which are associated with the main functions and sequences and can be easily identified in the text and on the logic diagram.

The block layout conforms approximately to the position of the logic gates and registers on the logic diagram.

TDV 2114
1426 - 10 - 76

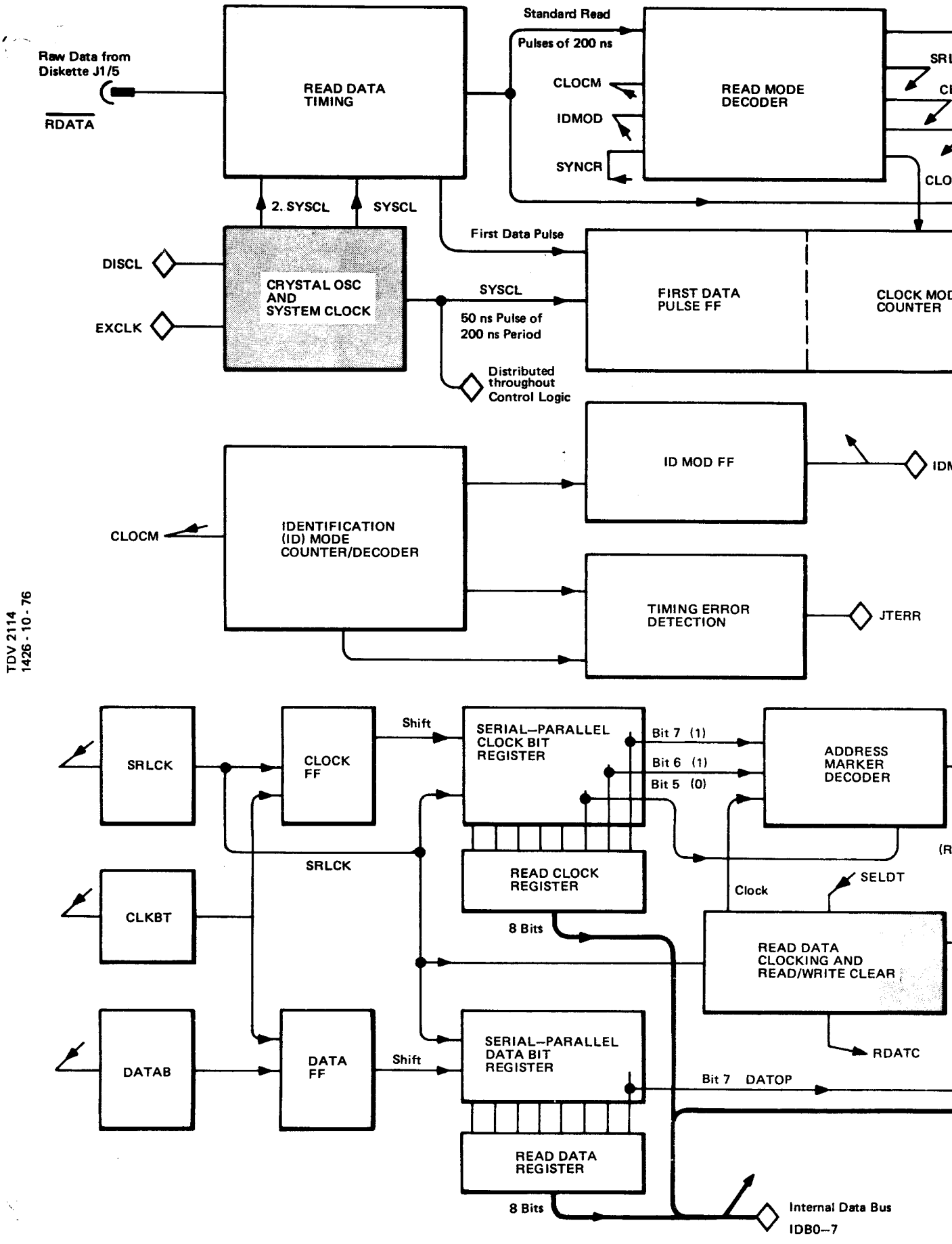



READ SEQUENCE SCHEMATIC

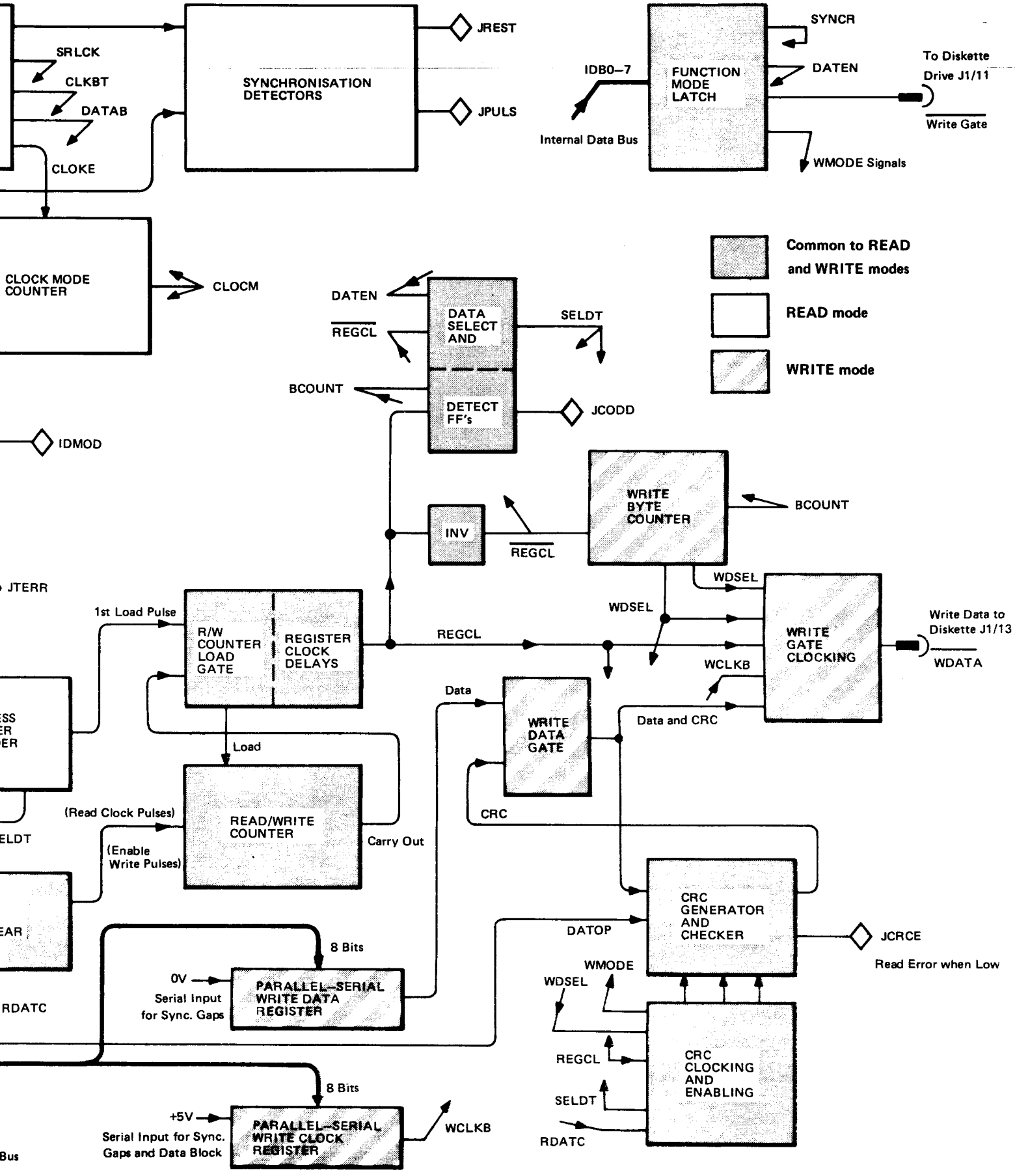



TDV 2114
1426 - 10 - 76

**TRACK FORMAT**


Raw Data from Diskette J1/5
RDATA
READ DATA TIMING
Standard Read
Pulses of 200 ns
CLOCM
IDMOD
SYNCR
READ MODE DECODER
2. SYSCL
SYSCL
First Data Pulse
DISCL
EXCLK
CRYSTAL OSC AND SYSTEM CLOCK
SYSCL
50 ns Pulse of 200 ns Period
Distributed throughout Control Logic
FIRST DATA PULSE FF
CLOCK MOD COUNTER
ID MOD FF
IDM
CLOCM
IDENTIFICATION (ID) MODE COUNTER/DECODER
TIMING ERROR DETECTION
JTERR
TDV 2114
1426 - 10 - 76
SRLCK
CLOCK FF
Shift
SERIAL–PARALLEL CLOCK BIT REGISTER
Bit 7 (1)
Bit 6 (1)
Bit 5 (0)
ADDRESS MARKER DECODER
SRLCK
READ CLOCK REGISTER
8 Bits
CLKBT
Clock
SELDT
READ DATA CLOCKING AND READ/WRITE CLEAR
RDATC
DATAB
DATA FF
Shift
SERIAL–PARALLEL DATA BIT REGISTER
Bit 7 DATOP
READ DATA REGISTER
8 Bits
Internal Data Bus
IDB0–7

READ MODE




**EAD MODE — Block Diagram**

**READ DATA TIMING (Waveform 1)**
Raw data is generated in the diskette drive logic and fed into the controller on connector J1 pin 5. This circuit produces standard read data pulses of 200ns and also synchronises these pulses with the system clock (SYSCL) at the start of every read operation. A read pulse is therefore equal to the period of the system clock. A clock frequency of twice the system clock (2.SYSCL) is used to ensure detection of raw data with pulse widths down to 130ns.
The DATEN signal resets the read data and system clock timing until the read mode is established, at the start of the synchronisation mode. This can be seen on the waveform for the synchronisation detector circuit.

**CRYSTAL OSCILLATOR AND SYSTEM CLOCK (Waveform 2)**
A 10MHz crystal controlled oscillator and a divide by two circuitprovide the clock timing pulses for the diskette controller logic. These pulses (SYSCL) are approximately 50ns wide with a period of 200ns.
With a low input of TP2 (DISCL), the oscillator can be inhibited to allow an external clock on TP1 to control and step the whole control logic.
Line termination resistors on the system clock output are used for impedance matching and ensure a sharp cut-off on the trailing edge of the clock pulse.
The accumulator buffer enable signal is derived from U76 pin 7 and is a 200ns period of pulse width ratio 1:1. This ensures that the buffer is enabled ahead of the SYSCL pulse and is ready to accept the data at the correct time. The buffer signal also remains on for several nano seconds after the SYSCL pulse to allow the contents placed on the internal data bus to settle.

**IDENTIFICATION (Waveform 5)** *
**(ID MODE)**

**TIMING ERROR (Waveform 6)** *

**SERIAL CLOCK PULSE (SRCLK)**
For every information clock bit, a serial clock input pulse is generated from the function decoder. The only exception is when reading the address marker when the serial clock will occur with the data bits for as long as the ID Mode signal stays high. (Function code 94).
When the serial clock pulse is low and the SYSCL pulse goes low a clock pulse is generated to clock the serial input registers. This signal is also inverted and fed to the serial clock flip-flop for use in decoding and further data counting logic.

**CLOCK BIT PULSE (Waveform 7)** *

**READ DATA REGISTERS (Waveforms 7 and 8)**
For every data bit read from the disk a DATAB pulse is produced from the read mode decoder. The only exception to this is during a clock mode (CLOCM pulse) or an Identification mode (IDMOD pulse). The DATAB pulse sets the data flip-flop when the CLKBT signal is high.
The inverters between the data flip-flop and the serial-parallel data bit register provide a delay so that the data from the flip-flop output is held high long enough on the register input to ensure being clocked into the register by the serial clock pulse.
The data bits are clocked serially into the register and after synchronisation and ID Mode detection the m.s.b., DATOP, is continuously carried from the register and clocked into the CRC generator by RDATC.
The read/write counter produces a REGCL pulse after every 8 bits of data read, i.e. each time a byte has filled up the register. The data byte is then clocked into the read data register and waits for the software, via JCODD, to initiate the read data enable signal. Each byte is transferred to the accumulator on the internal data bus.
Software instructions continue to operate on the data byte, transferring each byte from the accumulator to the data output register and then to the main computer. The flow of data to the main computer is described in the Input/Output control logic.
Each byte read from the diskette is counted by the software in the processor which records the number of times that JCODD is set. In the data block a total byte count of 128 is required and is then followed by 2 CRC bytes. The software automatically detects, by reading the header bytes, if a block contains 128, 256 or 512 bytes of data. A block length of 128 is the normal requirement.
When the last bit in the CRC byte is clocked into the generator, the software tests for a read error with jump instruction signal JCRCE. If the jump instruction signal is high, the main computer is informed via the status word and the read function can be repeated. This is accomplished by a new read command from the main computer.

**READ MODE DECODER (Function table 1)**
The read mode decoder is a pre-programmed read only memory which is permanently enabled and uses four input select lines, the fifth input (pin 14) being held low. The function table indicates the logic states which occur during the read operation and provides the information necessary to construct and verify the various read mode waveform diagrams.
The functions dependent upon the decoder are as follows:
a) The clock mode (CLOCM)
b) Synchronisation (SYNCR)
c) Header identification mode (IDMOD)
d) Timing error (JTERR)
e) Reading clock pulses
f) Reading data pulses

**CLOCK MODE PULSE GENERATOR (Waveform 3)**
The clock mode generator is a counter, which will only give an output (CLOCM) when it is clocked by 15 consecutive SYSCL pulses. After the first read data pulse from the diskette, it is necessary to detect a pattern of clock pulses occurring at 4μs intervals with no data pulses present. This pattern is the synchronising gap ahead of the track headers and data blocks and is the only pattern which will allow the clock mode counter the time to develop a stream of CLOCM pulses. When 16 consecutive CLOCM pulses are detected (controlled by software), synchronisation will follow.
The waveform diagrams show the development of the CLOCM pulse in 3 stages as follows:
a) the premature resetting of the counter when read information pulses occur at 2μs intervals.
b) the production of a CLOCM pulse when read information pulses occur at 4μs intervals.
c) the pattern of read clock pulses and CLOCM pulses necessary for synchronisation.
After the first data pulse, the output from U72 pin 6 remains high during the read operation to maintain the counter U71 in the enabled state. The CLOKE signal from the read mode decoder prevents the counter from recycling to zero after the CLOCM pulse occurs. The next read information clock pulse produces the preset load signal on pin 9 of the counter which sets the counter to binary 1 and allows the sequence to continue as shown in waveform diagram (stage b).
The CLOCM pulse width is approximately 1μs, but during the detection of an address marker in the identification mode the CLOCM pulse width will increase due to the absence of a clock pulse.

**SERIAL-TO-PARALLEL CLOCK REGISTER (Waveform 7)**
The clock bit register is a serial bit input shift register to an 8 bit parallel output register. Serial shifting is controlled by the SRCLK and CLKBT input signals whilst the output is available at all times and called for when the register is full. When the register is full, but the information is not required, the data is overwritten and ignored.
The outputs taken from the three most significant bits are used for address marker decoding.

**ADDRESS MARKER DECODER (Waveform 7)**
By referring to diagrams in the Read Mode Logic introduction, it is shown that four types of address marker can occur before reading data. These can be in the identification header field, before the data block and also for an index mark.
The clock codes used are all identical on the three most significant bits, i.e. 110. This pattern is used to detect all address markers.
The waveform diagram shows the address marker code C7, FE being stepped through the clock and data serial registers. The ID Mode signal goes high after the first missing clock pulse, as described in the ID Mode counter circuit, and is reset at the next valid clock pulse; but the address marker code is not detected until the whole byte is loaded into the register.
After synchronisation, the serial read logic is enabled and is ready to decode an address marker. The first clock byte which is loaded into the serial clock register and holds this pattern causes the output of the A.M. decoder gate to go high. The logic 0 on bit 5 in the register is anded with the decoder output gate to provide the first load pulse into the Read/write counter and also establish the first REGCL signal.

**READ DATA CLOCKING (Waveform 8)**
The data byte of the address marker is the first byte of information which is read into the CRC generator. This byte and all subsequent bytes are read serially into the CRC until the point is reached when the software checks for a CRC error. This occurs at the end of the identification field and at the end of the read data block.
The m.s.b. of data (DATOP) in the serial/parallel data bit register is clocked into the CRC generator when the 1st REGCL occurs, which is approximately 200ns before the SELDT signal goes true, as shown in the waveform diagram. The next bit of data (bit 6) and subsequent bits, having been shifted serially by new data entering the register, are clocked into the CRC generator by the read data clock signal (RDATC).
The RDATC pulse occurs at every serial clock pulse after the SELDT signal goes high on the input of the select data gate.

ead only memory which
ect lines, the fifth input
dicates the logic states
vides the information
ead mode waveform

as follows:

orm 3)
will only give an output
SYSCL pulses. After
necessary to detect a
ls with no data pulses
head of the track
which will allow the
am of CLOCM pulses.
ed (controlled by

of the CLOCM pulse in

n read information pulses

d information pulses

M pulses necessary for

2 pin 6 remains high
nter U71 in the enabled
decoder prevents the
ero after the CLOCM
information clock pulse
nal on pin 9 of the
nter to binary 1 and
inue as shown in

approximately 1μs,
an address marker in
CLOCM pulse width
ence of a clock pulse.

put register. Serial
out is available at
the information is

narker decoding.

at four types of
ion header field,

10. This pattern is

through the clock
clock pulse, as
pulse; but the
egister.
an address
holds this
bit 5 in the
into the Read/

(Waveform 8)
ss marker is the first
is read into the CRC
subsequent bytes are
until the point is
checks for a CRC
d of the identification
read data block.
P) in the serial/parallel
into the CRC
GCL occurs, which is
e the SELDT signal
waveform diagram. The
subsequent bits,
by new data entering
to the CRC generator
al (RDATC).
t every serial clock pulse
s high on the input of

D MODE — Description

**SYNCHRONISATION DETECTORS (Waveform 4)**

This circuit consists of two flip-flops which are being set and reset by software until the sync mode is established. From this point, the JREST flip-flop stays high and the JPULS continues to be set and reset, but is not used again by the software jump instructions.

Synchronisation will occur when a regular pattern of clock mode pulses (CLOCM) and information read clock pulses cause the JPULS flip-flop to be set and reset for 16 consecutive periods of 4μs intervals without JREST going high. This condition is counted by the software in the micro-processor (mp) and produces the SYNCR pulse from internal data bus bit 7.

The micro-processor (mp) instruction responsible for activating the synchronous mode is shown in hexadecimal notation D0 in the mp instruction code table. Bit 7 is latched in the function mode latch U49 pins 18 and 19.

The waveform diagram shows the circuit conditions for synchronisation to occur. At the point when the first CLOCM pulse is established is the time when the software, in conjunction with the read mode decoder, enter the synchronisation mode.

For the read operation to continue correctly, it is necessary to read the track header and identify the address marker by entering the identification mode. As can be seen from the header format, this mode follows immediately after the synchronisation gap.

**FUNCTION MODE LATCH**

When a mode signal is required to be maintained in a logic high or low state, an internal data bus bit is clocked into the function mode latch by the action of software initiating a combined clock function. Each time the COMCL signal is selected, the program will place the correct bits on the data bus to ensure that a bit which has been previously set remains set.

The internal data bus bits 0 to 7 are clocked into the function mode latch from either the accumulator or the programmed memory. The functions of each data bit are as follows:

DB0 Program select section 1 (PSEL 1)
1 Count 7 enable (C7ENB)
2 Program selec section 2 (PSEL 2)
3 Clock mode disable (CMDIS)
4 Data enable (DATEN)
5 Write mode 2 (WMOD2)
6 Write mode 1 (WMOD1)
7 Synchronise (SYNCR)

The functions are associated with the following modes

| Mode | Signals |
|---|---|
| Program addressing | PSEL 1, PSEL 2 |
| Read | DATEN, SYNCR, C7ENB |
| End of Read | RCMDIS |
| Write | WMOD2, WMOD1 |

ITEMS MARKED * ON FACING PAGE

**REGISTER CLOCK (REGCL) (Waveform 8)**

The first REGCL signal is used to inform the software program that an address marker has been loaded into the clock and data registers and that this information is to be transferred to the processor and then to the main computer. This action is initiated by the jump instruction flip-flop JCODD.

The REGCL signal is inverted and performs the following functions to allow the software to operate on the read logic.

a) Clocks the address marker clock and data bytes in the serial/parallel registers to the read clock and data registers.
b) Sets the select data flip-flop to produce the SELDT signal which then remains high during the read mode and enables the select data clock gate. The output from this gate is used to clock the read/write counter and the CRC generator.

The first REGCL pulse which sets JCODD is also fed to the CRC clocking logic. At this time only, is the REGCL used for clocking the CRC during the read mode.

The two flip-flops in the line between the read/write counter load gate and the REGCL signal are used to delay the REGCL signal to change the write data pulse from 200ns to 400ns. This may be necessary with diskette drives from some manufacturers. An OR-gate will also be required in the circuit modification.

**READ/WRITE COUNTER (Waveform 8)**

The pulse produced from the decoder output gate on detection of an address marker loads (presets) the read/write counter to binary 8, since the signals WMODE and 7CENB are low and DATEN is high. The counter is then incremented by 1 on each successive read data clock pulse (RDATC) up to a count of 15. The carry out pulse from the counter is fed back to load (preset) the counter back to binary 8 and also produce a REGCL pulse.

At this point, a completely new byte has been read into the clock and data serial/parallel registers, and also the previous AM data byte has been clocked serially out of the register and into the CRC generator.

The REGCL pulse again sets the jump instruction flip-flop JCODD. The software detects this condition, resets the flip-flop and transfers the data byte from the read register to the processor and then to the main computer. When reading an address marker, both the clock and data bytes are transferred to the main computer. When reading the data block, only the data bytes are transferred, (software controlled).

The signal 7CENB is set high when reading the second CRC byte after the data block. The counter is preset to binary 9 (1001) which allows only seven counts to be made before the carry out signal initiates REGCL. Only seven counts are required to clock out the last byte in the serial/parallel data register, since the m.s.b. has already been clocked into the CRC on the last clock pulse in the previous byte.

If the binary counter is not preset to binary 9, an extra clock pulse would insert an unwanted bit into the CRC generator. The REGCL and therefore JCODD would be delayed by one clock pulse causing both hardware and software errors to occur.

**CRC GENERATOR AND CHECKER (Waveform 9)** *

TDV 2114
1426 - 10 - 76

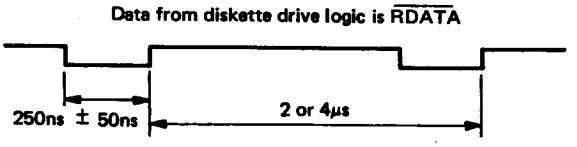
**IDENTIFICATION MODE (Waveform 5)**
There are two identification modes which occur during a read mode, firstly, the header address marker in the ID field and secondly, the data definition address marker preceding the data block. An index address marker also produces an ID Mode but is ignored by software. The address markers producing an ID Mode are shown and defined in the waveform diagram.
The bit pattern of an address marker is arranged so that the ID Mode signal from the ID flip-flop will go high when the first information clock pulse is missing and be reset on the next clock pulse. In the waveform diagram it is shown that the clock mode and synchronisation are both established. The ID Mode cannot occur during non-synchronisation because the flip-flop is inhibited by the low SYNCR signal.
At the start of every clock mode pulse, the ID counter is clocked by the SYSCL and normally reaches a count of about 8 before being reset by the information clock pulse. This resetting action can be verified by the read mode decoder function table, where it can be seen that code 14 will reset the counter. If, however, the information clock pulse is missing, then the counter will count to ten.
The binary code of ten (1010) is decoded by U60 and sets the ID Mode bistable U73. The bistable will remain set until reset by the next clock pulse, (function table code 05).
At the same time that an ID Mode (address maker) is being detected by the ID counter, the bit pattern is also being read into the serial to parallel read circuit. This sequence is described later and shows that the complete pattern must be read before the address marker can be identified fully.

**TIMING ERROR (Waveform 6)**
A timing error occurs when two or more consecutive information pulses are missing, the first one being a missing clock pulse.
The effect of a missing clock pulse is to establish the ID Mode signal. However, because the next pulse is missing, the ID counter will carry on counting until the carry out pulse from the counter sets the timing error detection flip-flop.
The ID counter reset signal is a result of code 14 on the function table, but since the pulse is missing the reset signal stays high.
This allows the count to continue and also enables the timing error flip-flop to be set by the carry out signal.
The timing error gate is now enabled by a high from the flip-flop and a high from the counter which sets the timing error latch and produces the timing error jump signal.
When a timing error occurs before the data block transfer has started, the software detects the jump instruction and resets the whole read logic. A new read mode is then re-synchronised and starts to re-read the format. The first time the software tests for a timing error is after the ID Mode has been set.
The signals responsible for resetting the read logic are DATEN and SYNCR, since without these, the read counters and flip-flops cannot function.
The clock mode circuit is not reset after a timing error, nor is the oscillator circuit, since both of these are required for immediate re-synchronisation. Only when the system is cleared by the CLEAR pulse are these circuits reset.
When the data block is being transferred, the software is checking at every byte for a timing error. If an error occurs, the software will stop transferring data and reset the read logic with signals DATEN, SYNCR and RCMDIS. The software will then give the correct status pattern (timing error) and wait for a new instruction.

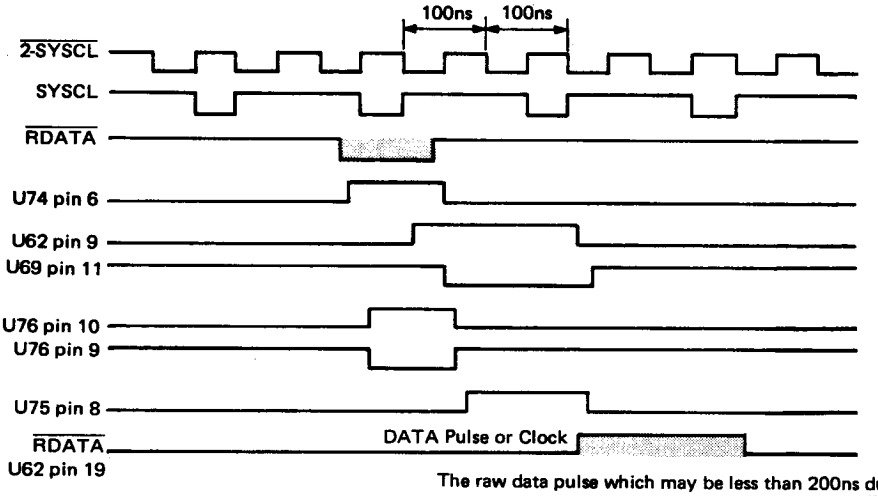
**CLOCK BIT PULSE (CLKBT) (Waveform 7)**
For every information clock bit, a clock bit pulse is generated from the function decoder. The only exception is when reading the address marker when the clock bit will not occur when the ID Mode signal is high. (Function code 94).
During normal data transfer, the clock bit signal is fed to the clock and data flip-flops preceding the serial-to-parallel clock and data registers respectively. The CLKBT signal is used to clock in clock bits and reset the data flip flop. During the ID Mode, the CLKBT pulse is used to load zero's into the clock bit register.
The clock flip-flop is used to load the clock register but also as an extension to the register to provide the '9th' clock bit at the start of the first valid data byte after synchronisation.
To ensure that a complete data byte is clocked into the serial/parallel register it is necessary for the serial clock input pulse to be one system clock time ahead of the first data bit, that is, it takes 9 serial clock pulses to load the first 8 bits of data. This occurs at the start of reading data, thereafter the last data bit of each byte is clocked into the register by the first clock pulse in the following information byte.
By referring to the waveform diagram it can be seen that it is necessary to use the first clock pulse in the following byte of information in order to fill the register completely.
On the waveform diagram, the first CLKBT signal is shown to set a logic 1 into the clock register, because the output of the clock flip-flop is high and cannot change with a low CLKBT signal on both inputs, (Texas function table). At this point, the input to the data serial register is low and is therefore clocked into the register.

**CRC GENERATOR/CHECKER (Waveform 9)**
The Cyclic Redundancy Check (CRC) Generator/Checker is a programmable device which operates on serial data streams and provides a means of detecting transmission errors. To check an incoming message (reading) for errors, both the data and check bits are entered serially via the D input. The read data signal is DATOP and is fed from the m.s.b. of the serial/parallel data bit register. The CRC checker is not in the data path, but only monitors the data read from the diskette.
The CRC is preset before each check is made, this is during the synchronisation mode when DATEN is being pulsed and WMOD1 remains low. Also during reading, the check word enable input (CWE) is high, since both WMOD1 and WMOD2 signals are low.
The data byte which forms part of the address marker is the first byte of data to be entered into the CRC. Data cannot be entered before the address marker is detected because the REGCL and RDATC signals have not been produced. Both these signals are required for clocking in data to the CRC.
The m.s.b. of the AM data byte is clocked into the CRC when the first REGCL is produced. This first clock pulse to the CRC is generated from the WRITE CRC CLOCK GATE and (1st BIT READ GATE). The SELDT signal on U67 pin 5 is still high, and with REGCL, provides a clock pulse from the CRC CLOCK GATE. The data on the D input of the CRC is clocked in on the negative going edge of the CRC clock pulse.
Referring to the waveform diagram, the SELDT signal on U67 pin 5 goes low immediately after the 1st REGCL and thereafter inhibits CRC clocking via the write clock gate. It is from this point onward that RDATC is produced and is used to clock all subsequent bits of data into the CRC via the read CRC clock flip-flop and CRC clock gate.
When the last bit of the second CRC byte is clocked into the checker, the JCODD signal, produced to indicate the end of the byte, is detected (counted) by software and instructs the program to test for a CRC error. This error signal is jump instruction JCRCE which if low indicates no error and if high indicates an error. The error signal (JCRCE) is normally high during the read function, but at the point when tested, it must go low in the JCODD time to indicate that no error has occurred. The JCRCE signal is reset to high at the start of each new read function or write function.
When the JCRCE signal remains high at the test time, the software program inserts a logic 1 into bit 2 of the status register and sets the Interrupt Request flip-flop. The main computer checks the status register at each input/output instruction and also as instructed by the main program. On detection of an error, the processor program resets the whole read logic and the main computer program may either accept the information and label it as suspect or reject it and repeat the read function. The interrupt request signal is acknowledged and status read according to the current priorities of the main computer program.
To ensure that the correct number of bytes (and bits) are entered into the CRC before an error check is made, the JCODD signal is counted by software and the signal 7CENB is set by program at the correct time. The 7CENB signal is described in the read/write counter and also shown in the CRC checking waveform diagram.

## Waveform 1 Read Data Timing and Standardisation of data pulses

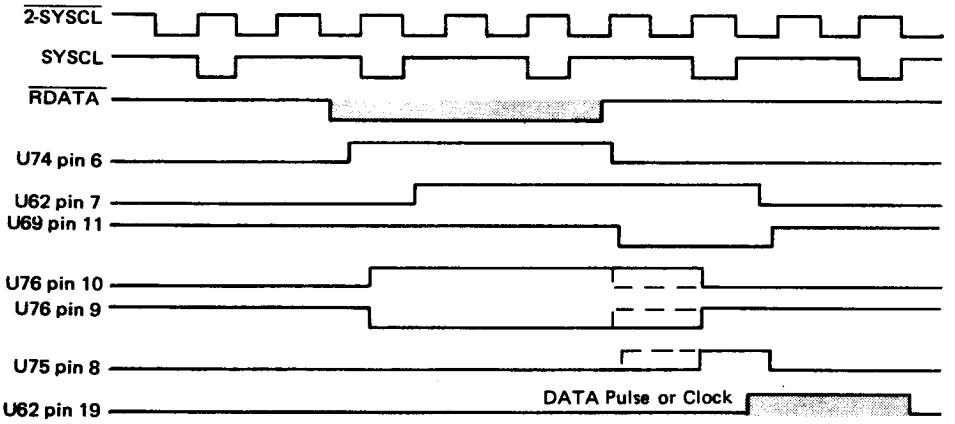



The RDATA minimum pulse width is 150 ns therefore a faster clock pulse 2SYSCL with a period of 100ns is required.




The raw data pulse which may be less than 200ns due to variation between different drives. Therefore the RDATA pulse is stretched to a 200ns standard pulse width.

TDV 2114
1426 - 10 - 76




When RDATA is a long pulse much greater than 200ns, the clock circuit will still hold it to 200ns.

**Waveform 2 Generation of System Clock (SYSCL)**

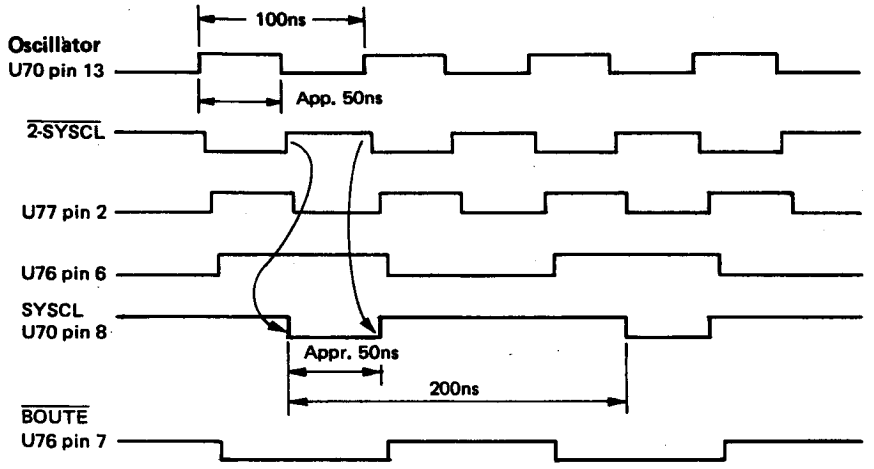


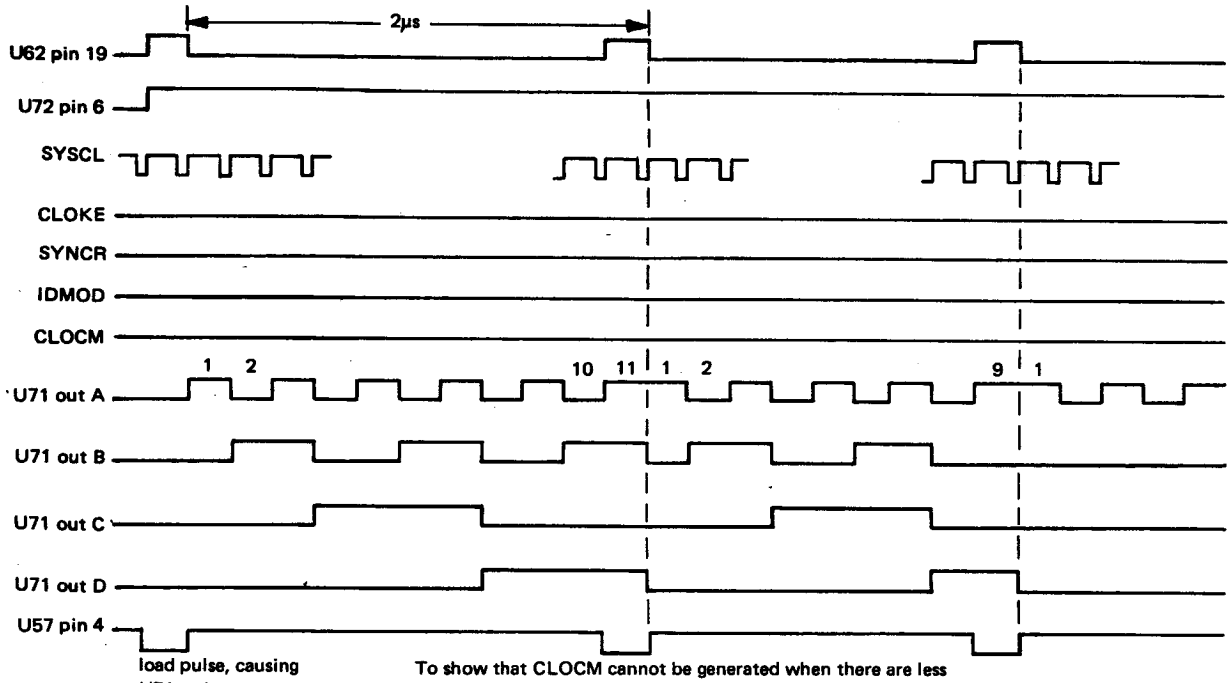

**Table 1 Read Mode Decoder Function Table**

| | INPUT | | | | | OUTPUT | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Pin 14 | Pin 13 SYNCR | Pin 12 IDMOD | Pin 11 CLOCM | Pin 10 (DELRI)* | Pin 9 CLKBT | Pin 7 RESCN | Pin 6 SRCLK | Pin 5 IDRES | Pin 4 | Pin 3 CLOKE | Pin 2 DATAB | Pin 1 | Hex. Notation | |
| NOT SYNCHRONISED | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | BC | Refer to this section for read mode before synchronisation (only signals on pin 1, pin 3 and pin 4 are used) |
| | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | B7 | |
| | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | B8 | |
| | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 14 | |
| | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | BC | |
| | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 05 | |
| | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | B8 | |
| | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 94 | |
| SYNCHRONISED | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | FC | Refer to this section for read mode after synchronisation |
| | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | FF | |
| | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | F8 | |
| | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 14 | |
| | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | FC | |
| | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 05 | |
| | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | F8 | |
| | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 94 | |

*(DELRI) Delayed read input

## Waveform 3 Generation of Clock Mode (CLOCM)

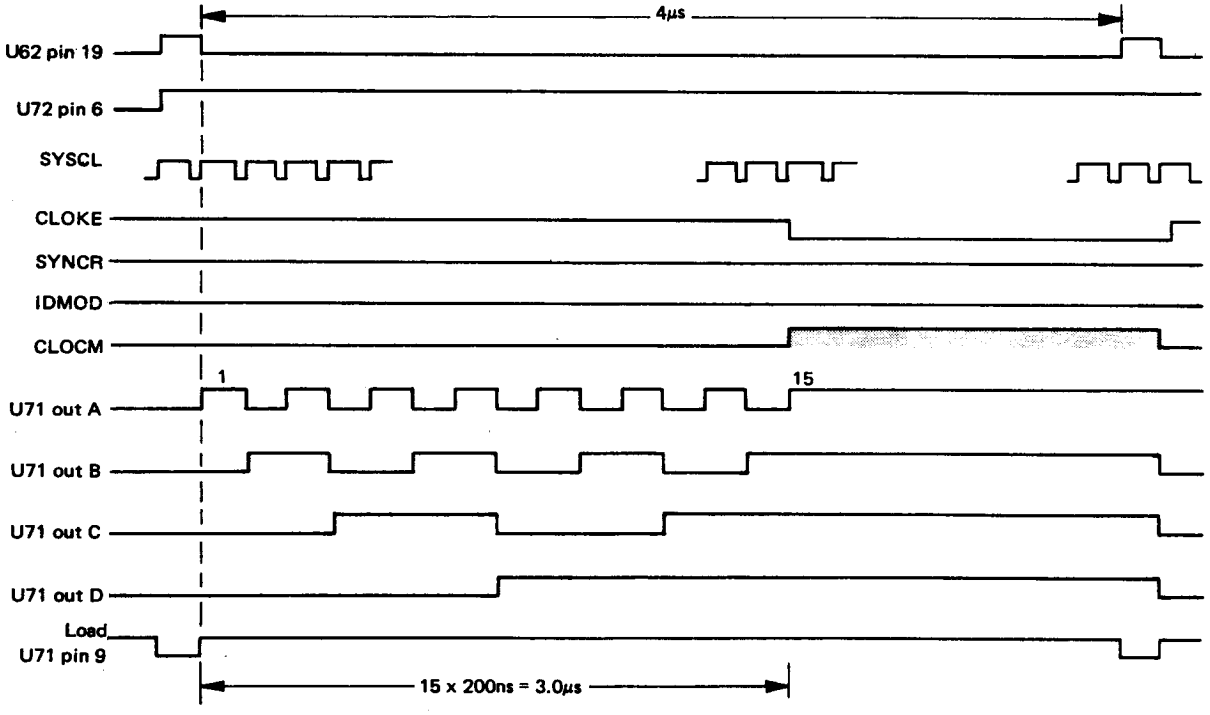
Counting of U71 (CLOCM) before synchronisation




load pulse, causing U71 to be reset to 1

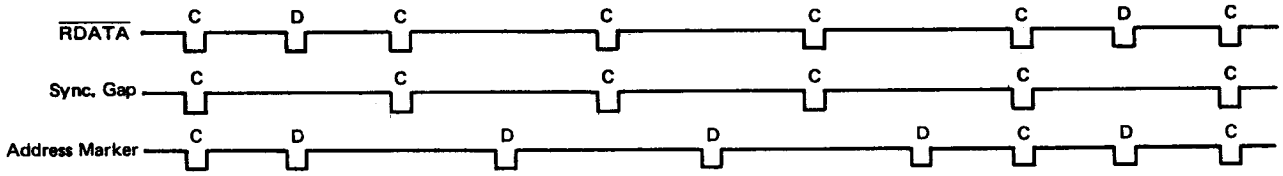
To show that CLOCM cannot be generated when there are less than 15 binary counts on U71. A load signal U71 pin 9 is produced by the U57 decoder for every data/clock pulse being read.

Counting of U71 (CLOCM) before synchronisation



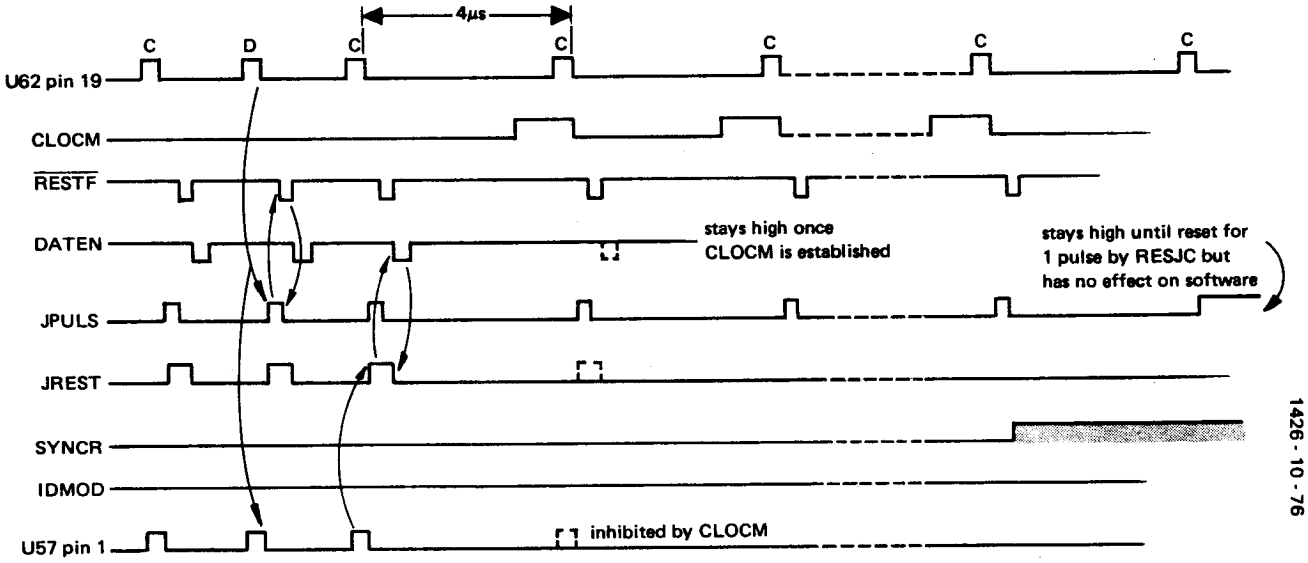

Counter U71 is used to inform the software if the time between 2 pulses is more or less than 2µs

## Waveform 4 Establishing Synchronisation



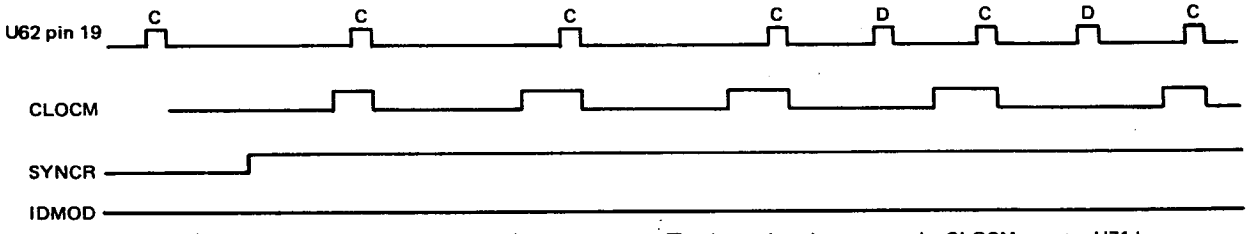

Synchronisation : Checking for an area where there are 16 pulses with 4us period, these pulses must be clock pulses contained in the sync. gap.

The Sync. gap is where data is 00 and clocks FF. If data is 00 but not in the sync. gap, the system will synchronise, but if not followed by an address marker will immediately re - sync. The worst case that may happen is when detecting an address marker with three clock pulses missing (but not more).




TDV 2114
1426 - 10 - 76

Data reading after Synchronisation.



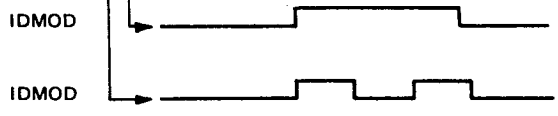

Shows that a pattern is established between clock pulses and CLOCM.

The data pulses do not reset the CLOCM counter U71 because the ID mode is low. This can be checked from the U57 decoder matrix sheet.

After synchronisation, the logic is waiting for an Address Marker, which can be one of the following:

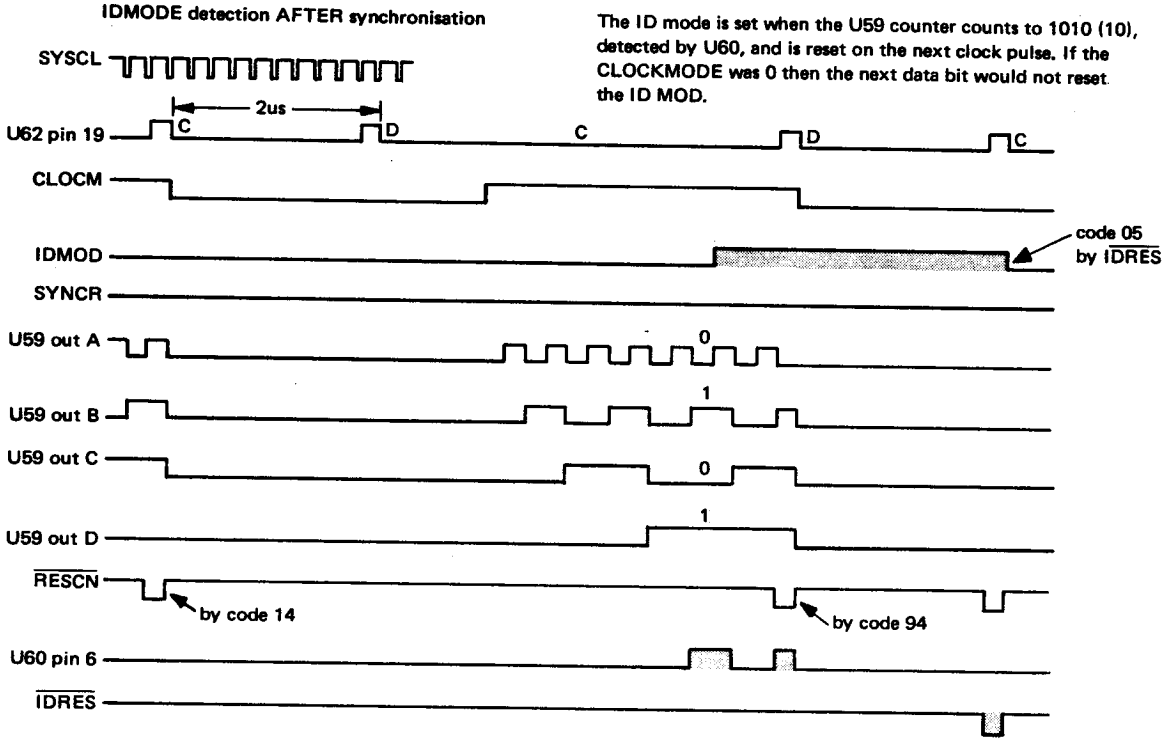
| | | | | |
|---|---|---|---|---|
| Data — | FE header | FB or F8 data block | | FC index marker |
| Clocks — | C7 | C7 | C7 | D7 |

| DATA | Clock | C | D | C | D | C | D | C | D | C | D | C | D | C | D | C | D |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FE | C7 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| FB | C7 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| F8 | C7 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| FC | D7 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 |



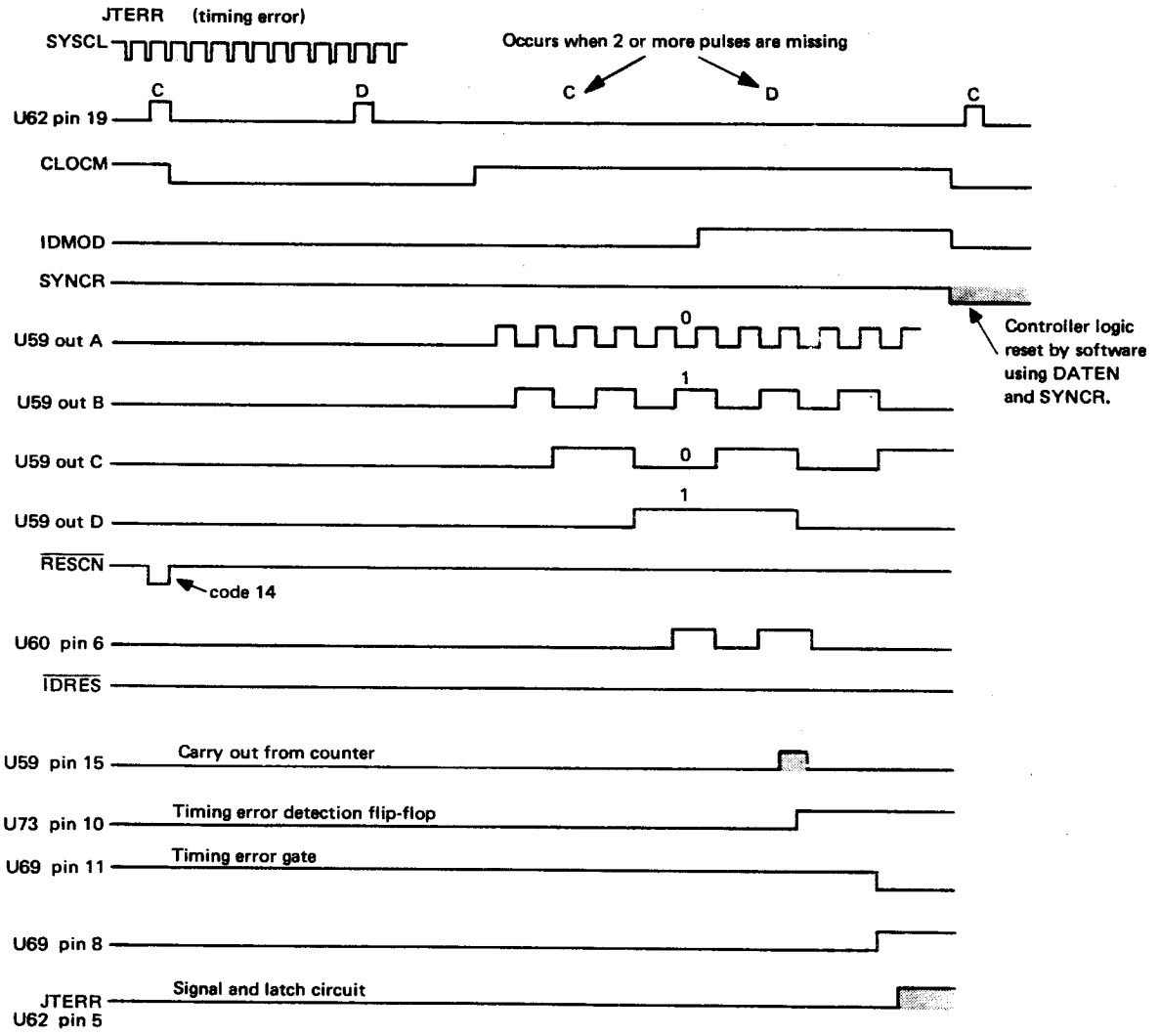

The clock pattern causes the IDMOD to differ. Again determined by the U57 decoder matrix

## Waveform 5 Detecting ID Mode after Synchronisation

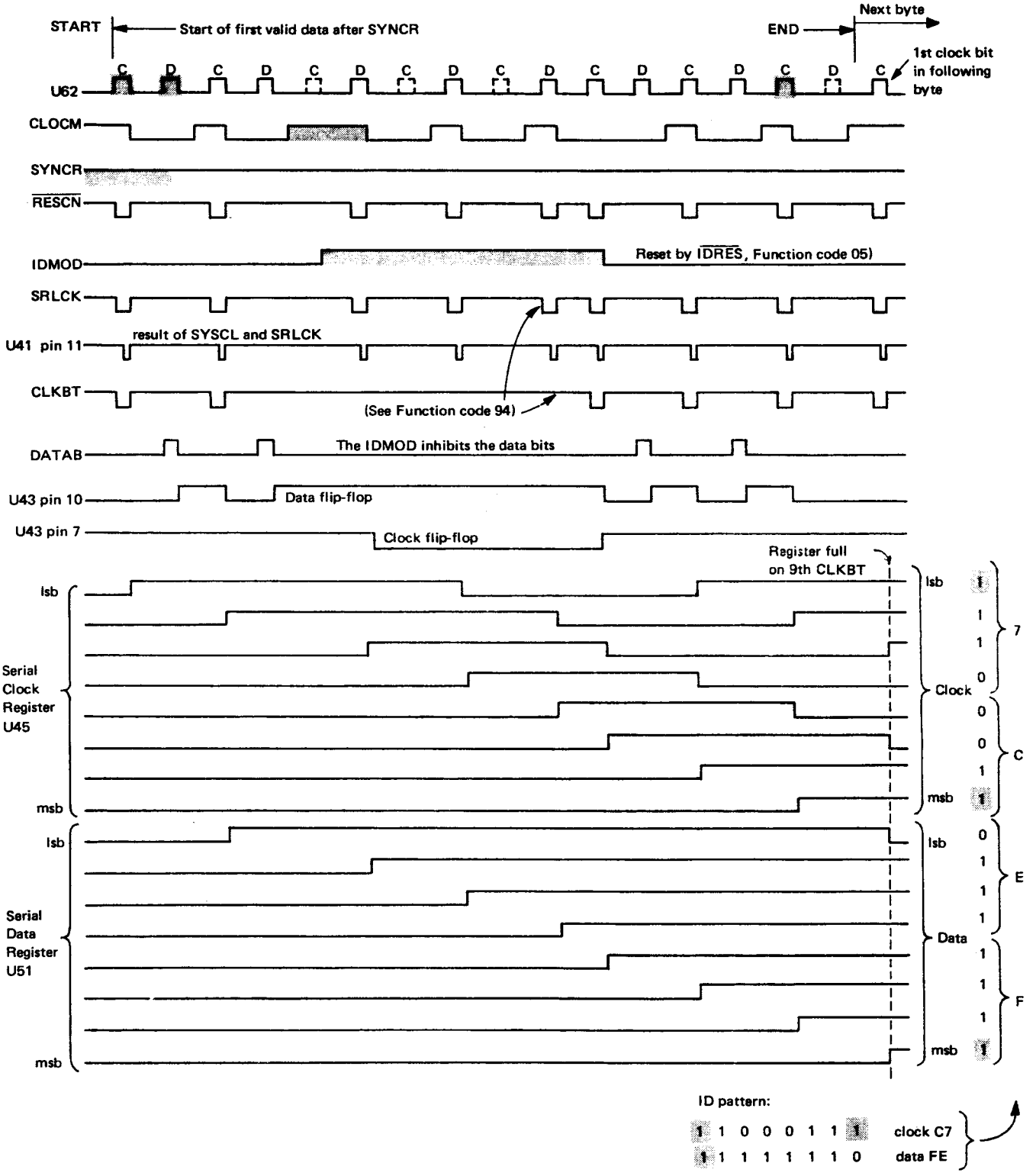



TDV 2114
1426 - 10 - 76

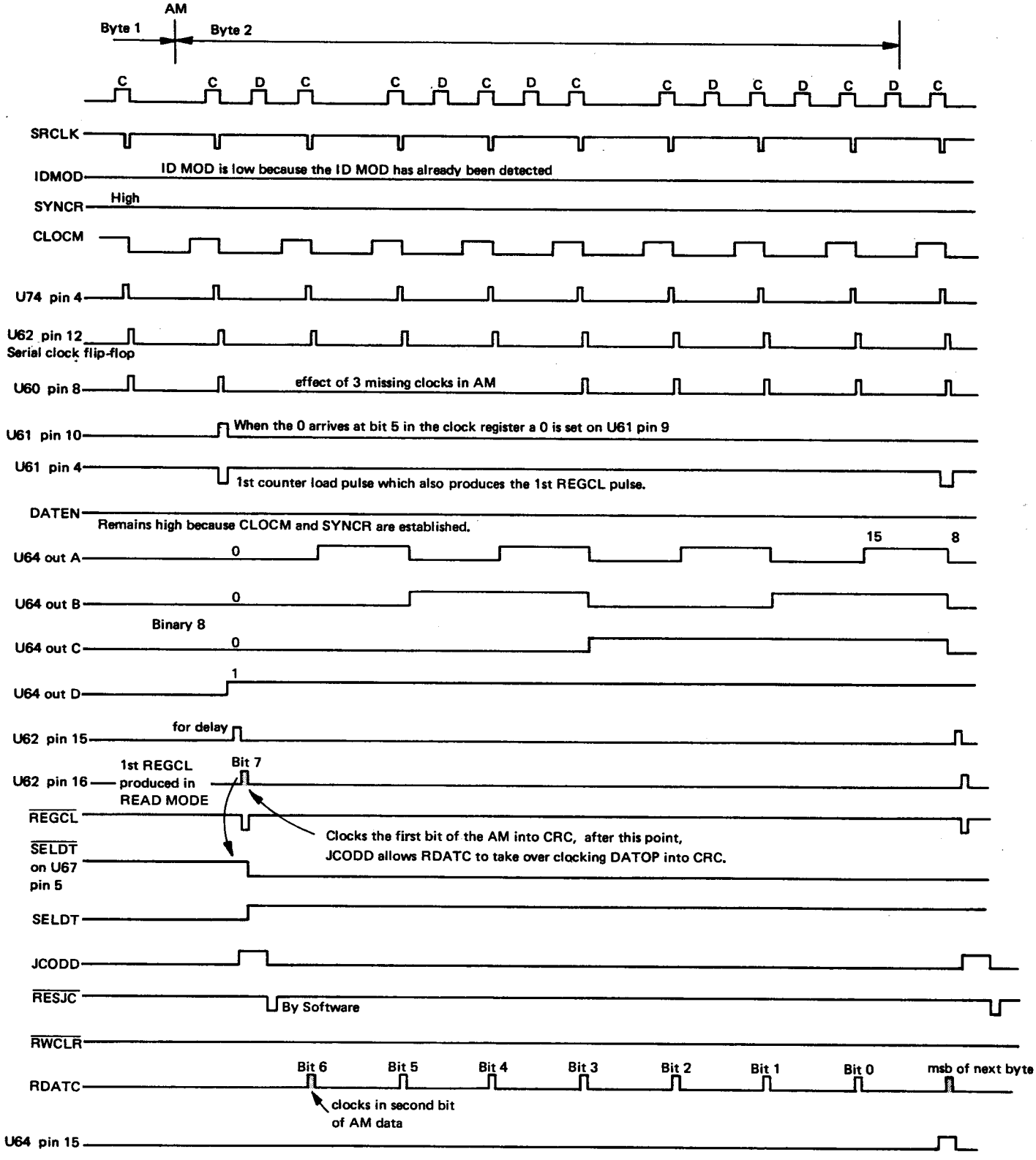
**Waveform 6 Timing Error**



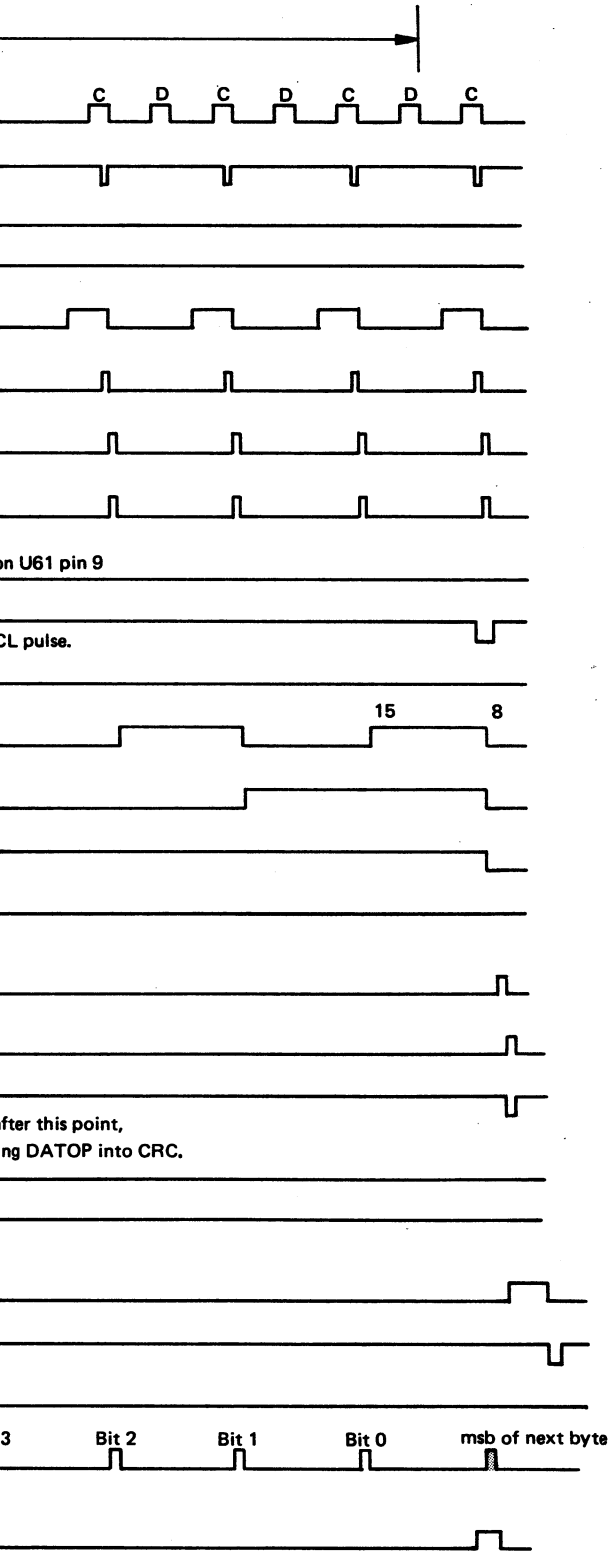
**Waveform 7 Detecting Address Marker**



**Waveform 8 Data clocking after AM**



AM



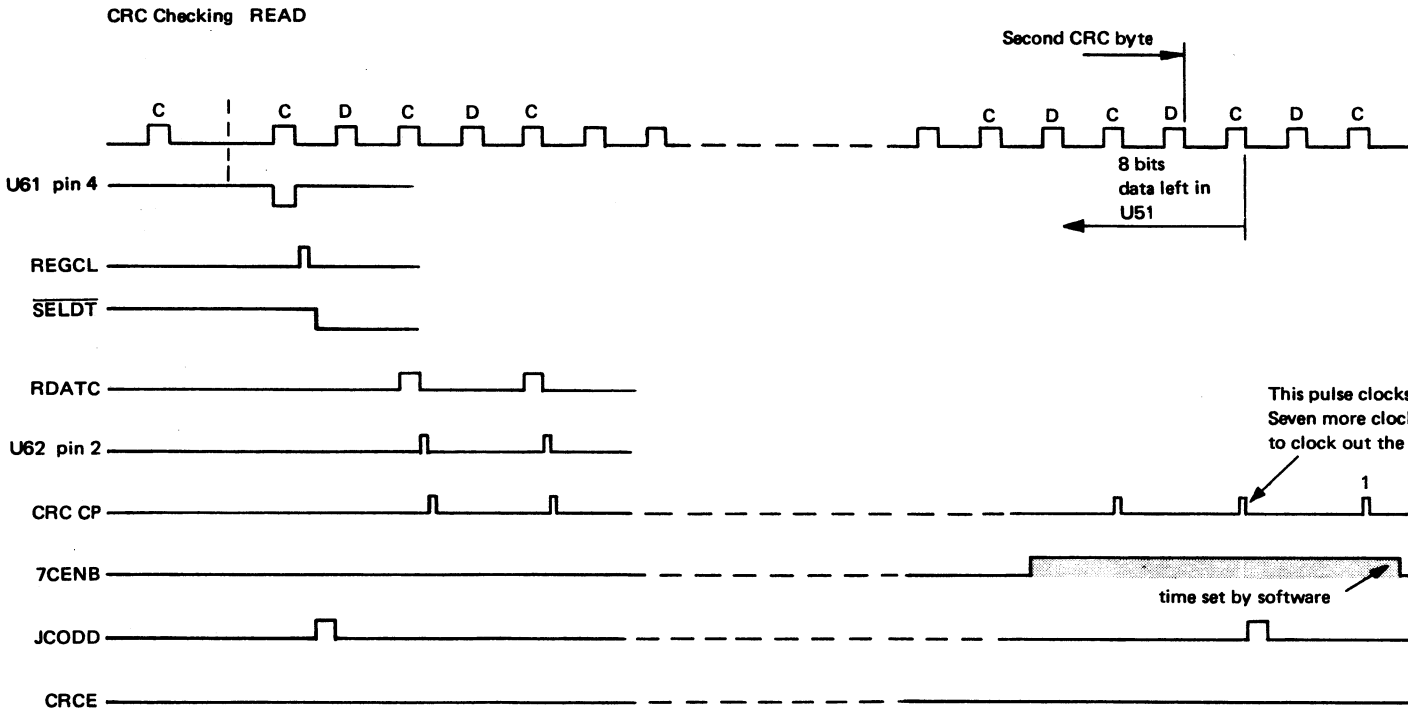

**Data Clocking:**

The system is waiting for an AM (clock is 110 XXXXX) then U64 is loaded to 8, REGCL goes true, which again sets SELDT true, and JCODD true. REGCL also transfers the data and clock pattern to their registers.

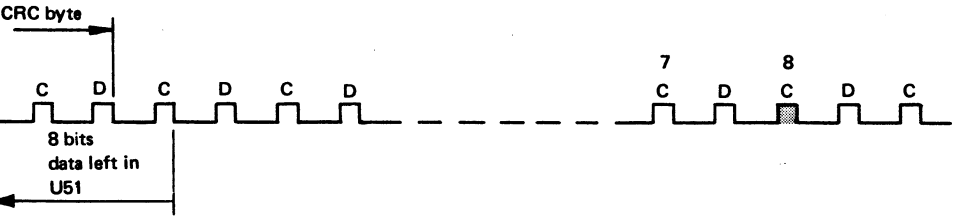
The U64 counter is clocked up one bit at a time each time a clock pulse arrives. When the counter is at 15, no pulse appears on the output, R, because 'T' is low. A high on R occurs on the next clock which sets JCODD, clocks data and clock pattern to register and loads the counter to 8.

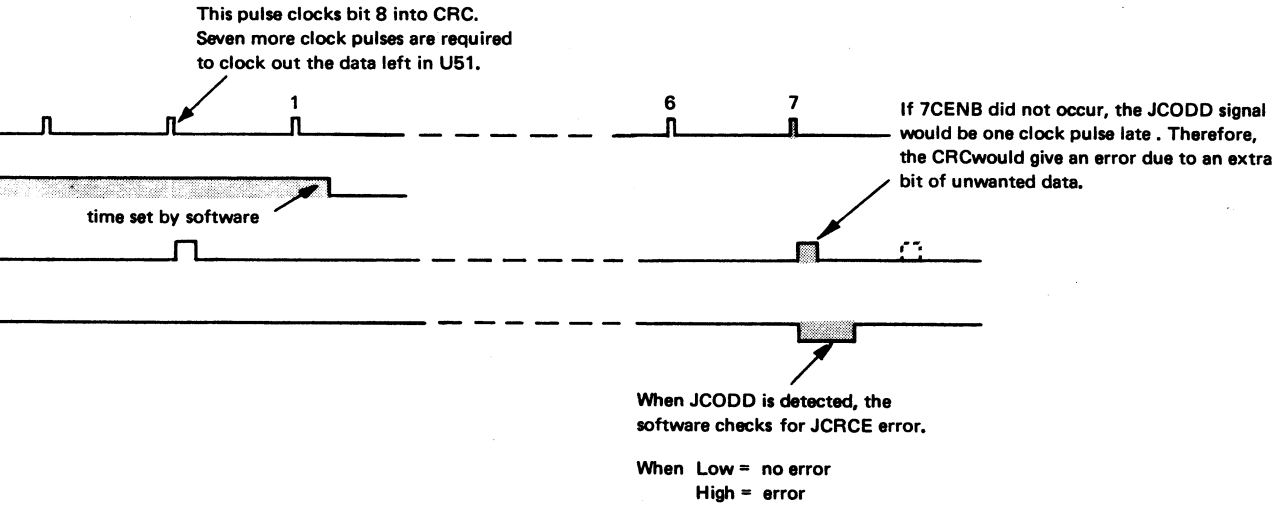
Each time JCODD is set, the software detects it, resets the signal with RESJC and reads data and clock pulses into Acc.

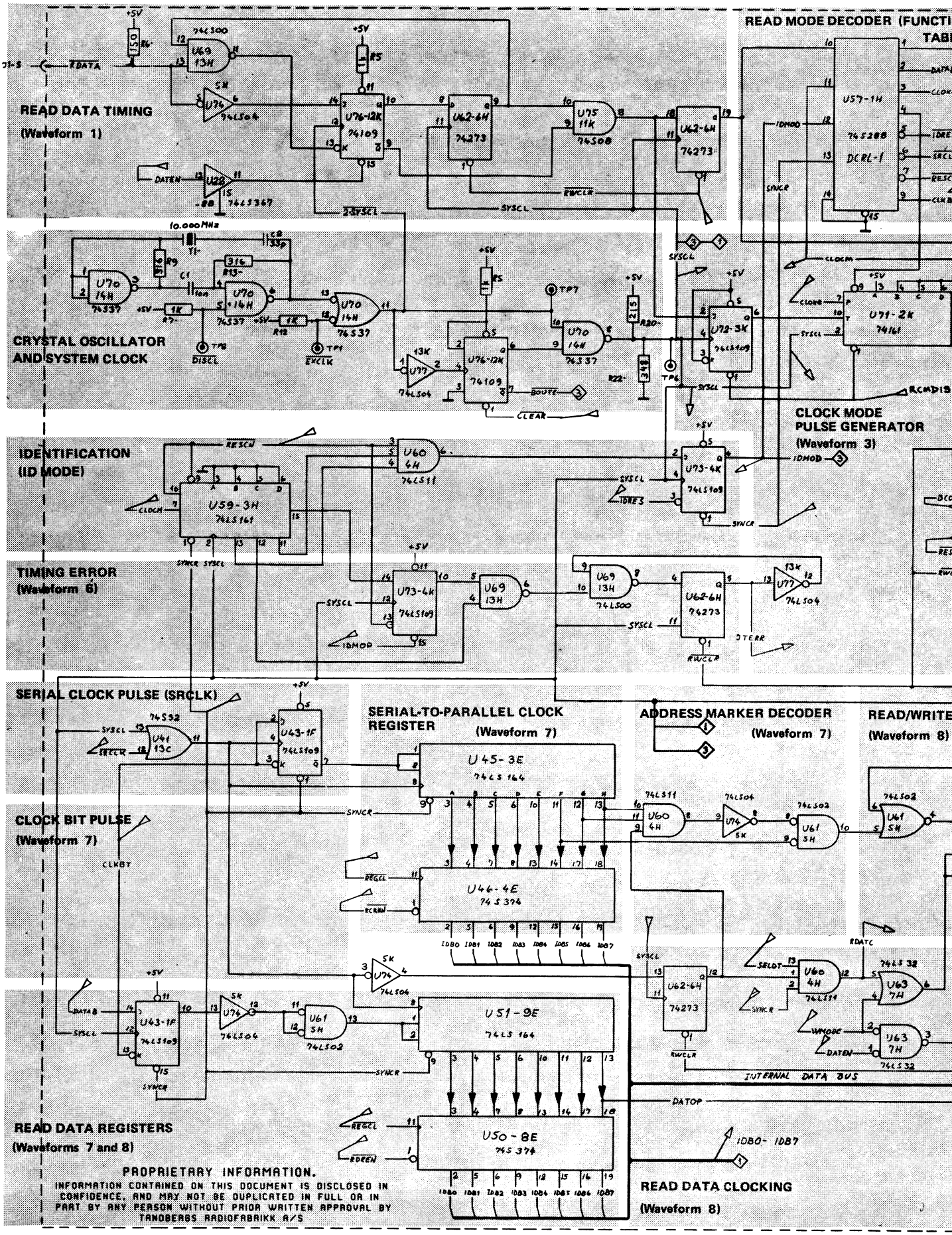
**Waveform 9 End of Read, CRC check**



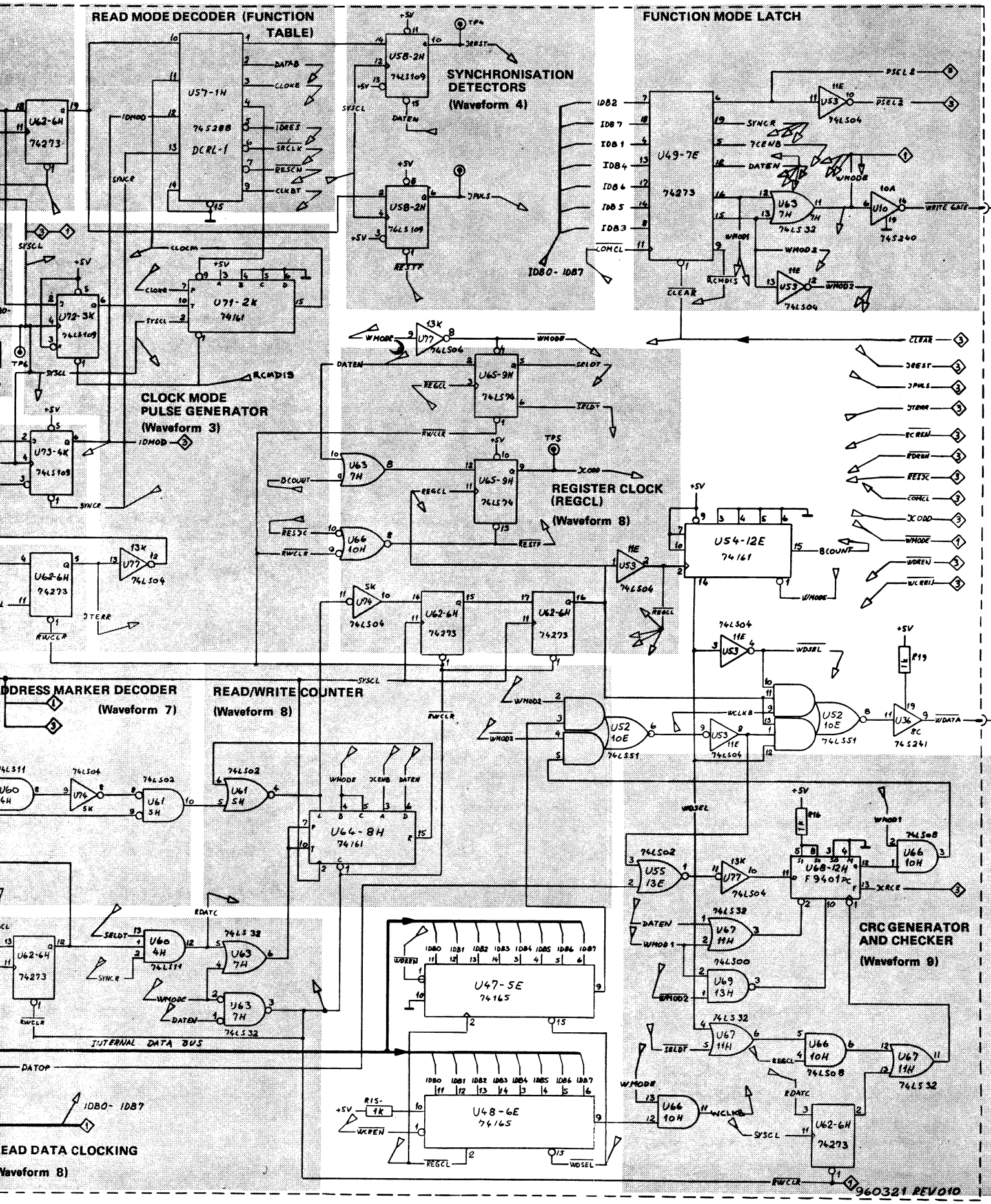
## m 9 End of Read, CRC check






READ MODE — Logic Diagram




**READ MODE — Logic Diagram**

| Position | Name | Part No. | Rev. No. |
|---|---|---|---|
| A1 | Diskette Controller | 960321 | 012 |

## WRITE MODE

## Write Mode

The diskette is set to the write mode after a sequence of instructions and commands have been carried out between the controller and the main computer.

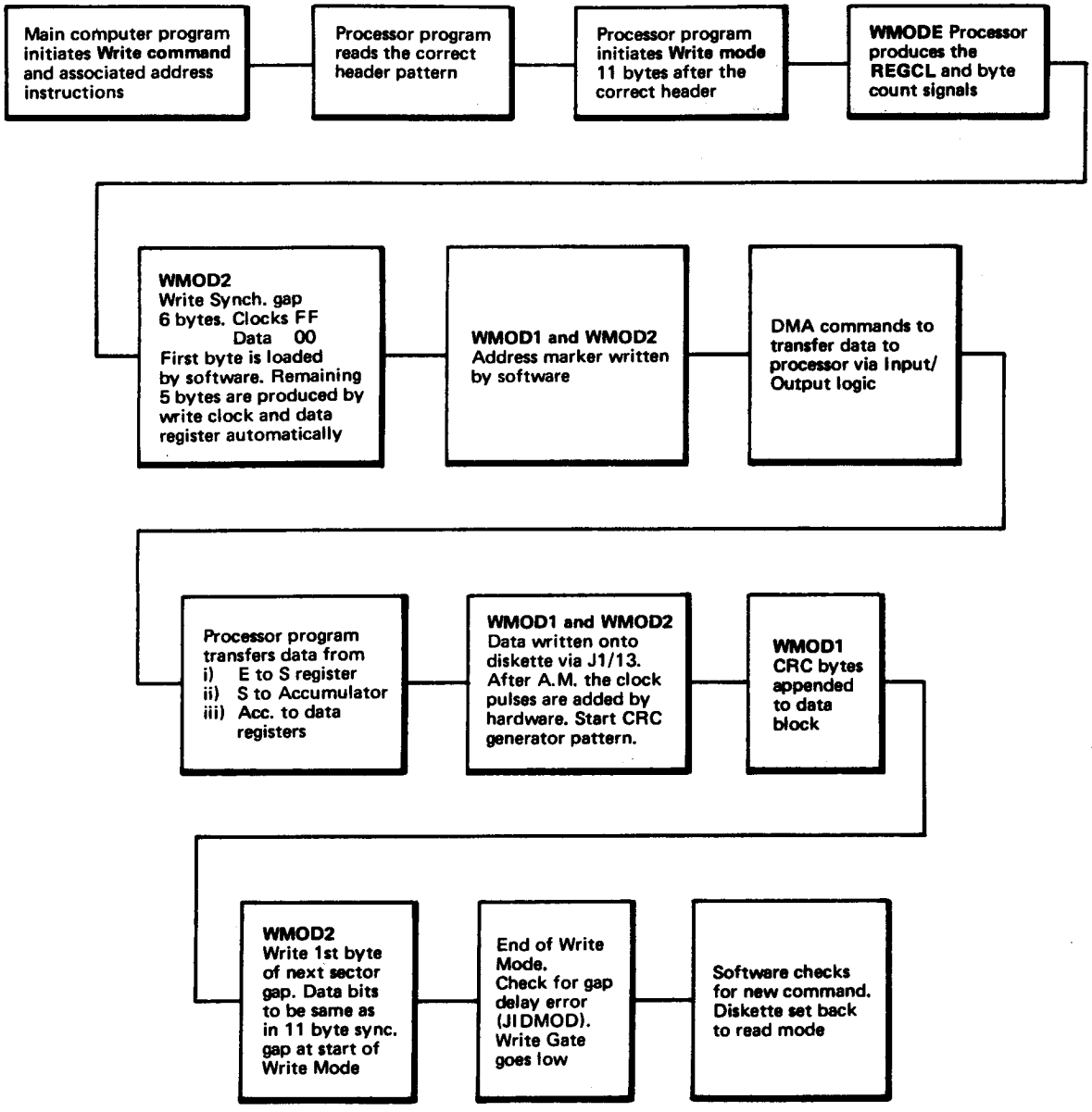
A seek track command is given by the main computer to move the head to the correct track, followed by a write command. The internal processor then starts the write operation by reading headers until the correct one is detected. The software then counts the first 11 bytes in the following gap and starts writing.

The processor program sets the Write Gate (J1 - 11) to a low level and this signal remains low for the duration of the write mode.

The write sequence is described briefly in the following schematic and can also be identified by referring to the block descriptions of the logic diagrams. The header format diagram also shows the write mode sequence as it occurs on the track.

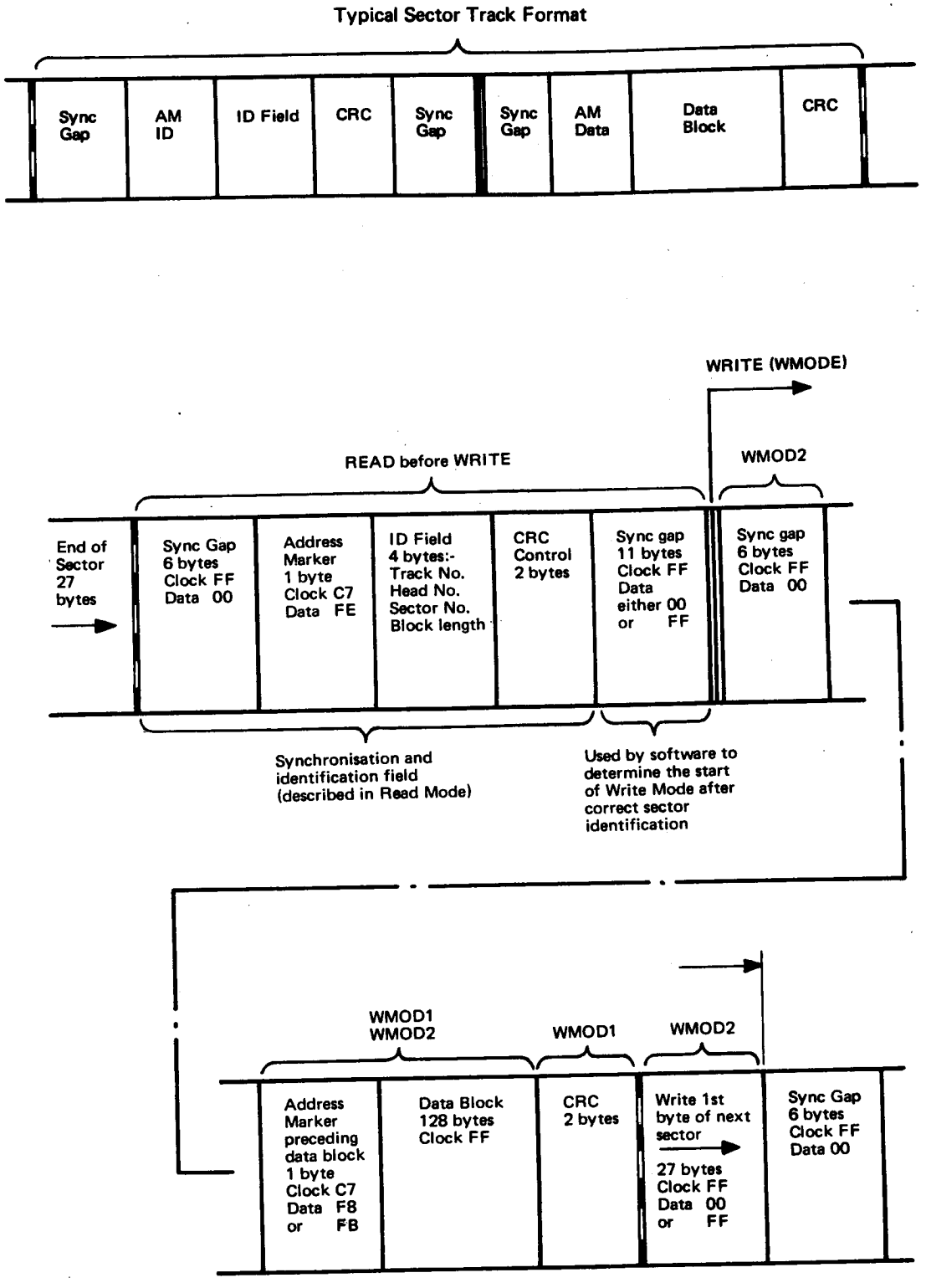
The signals shown on the write mode block diagram are those which are associated with the main functions and sequences and can be easily identified in the text and on the logic diagram.

The block layout conforms approximately to the position of the logic gates and registers on the logic diagram.




TDV 2114
1426 - 10 - 76

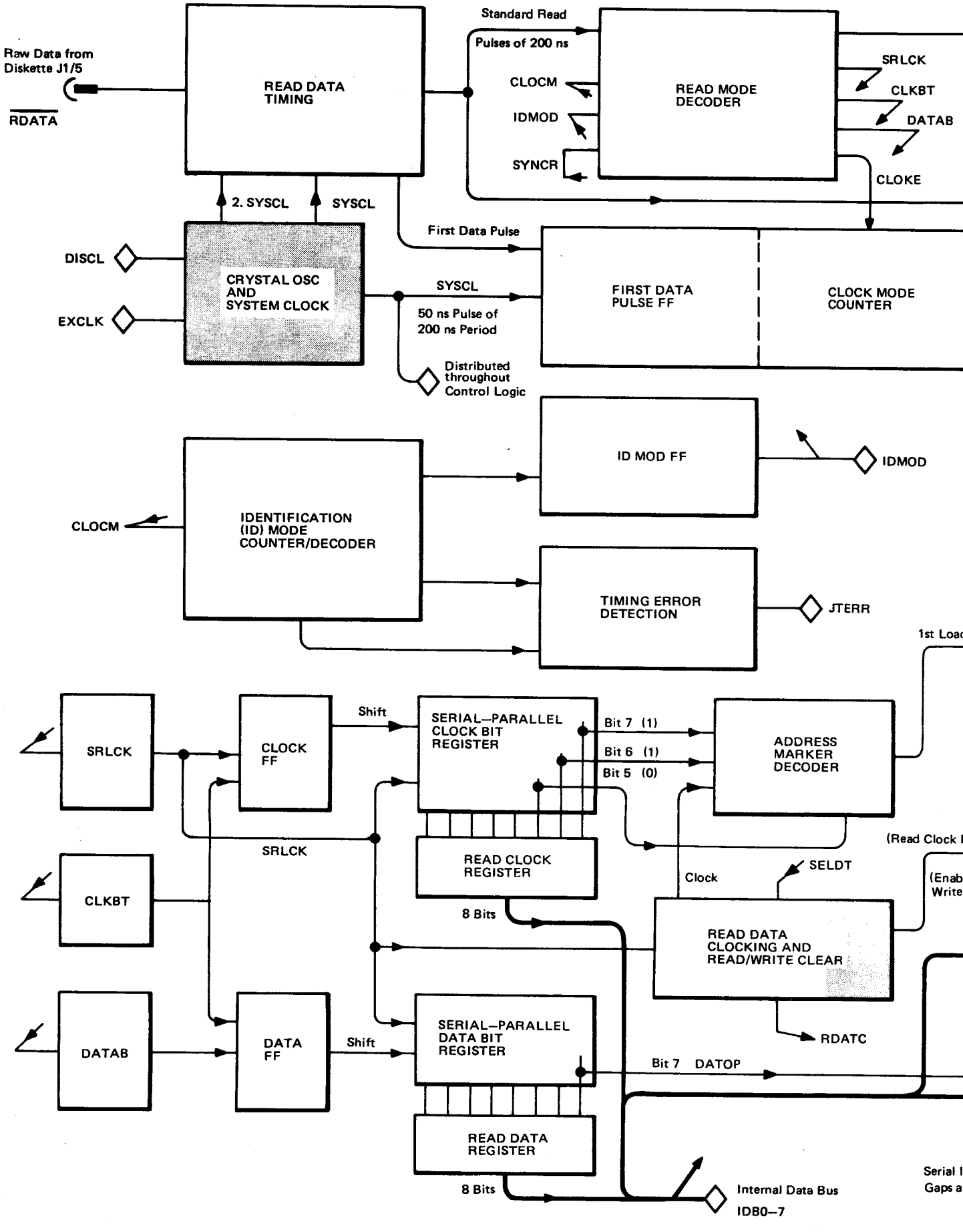
**WRITE SEQUENCE SCHEMATIC**




**TRACK FORMAT**

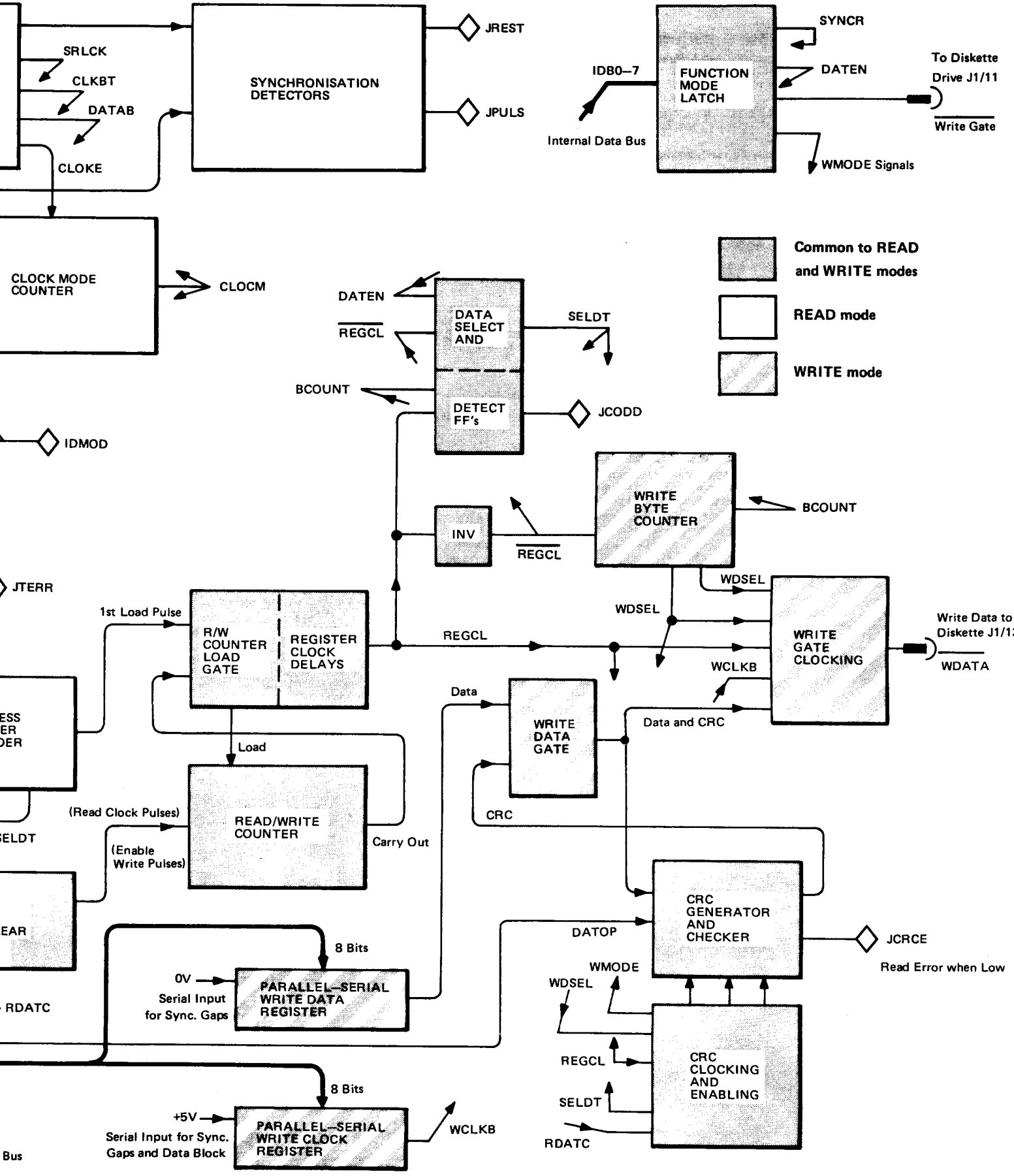


TDV 2114
1426 - 10 - 76




TDV 2114
1426 - 10 - 76

**WRITE MODE – Bloc**




**WRITE MODE – Block Diagram**

**WRITING 6 BYTE SYNCHRONISATION GAP (Waveform 1)**

At the end of the read before write synchronising gap, the software loads the write data register with all logic 0's and the write clock register with all logic 1's. This information is sent from the processor accumulator and loaded by the low going register enabling signals WDREN and WCREN. This software action occurs for the 1st byte only, after which, the registers are hardware loaded from fixed logic voltage levels.

On the 1st REGCL, the most significant clock bit in the write clock register is written to the diskette. The sequence of clock and data bits being separated and clocked to the diskette is described in the write gate clocking logic.

As each bit is shifted from the register, the fixed logic level, either 0V or +5V, serially loads a new bit into the register.

The CRC generator pattern is not started until after this 6 byte gap is written, since the WMOD1 signal, being low, holds the CRC preset input low.

**WRITING ADDRESS MARKER (Waveform 3)**

The software in the processor program detects the last JCODD signal in the 6 byte synchronising gap and sets WMOD1 high. The write clock and data registers are loaded with the address marker bytes, the clock is C7 and the data is either F8 or FB.

The REGCL pulse, after the JCODD signal, enables the first clock bit from the write clock register to be written to the diskette. The clocking of data and clock bits of the address marker continues with each REGCL and is as described in the write gate clocking logic. After 15 REGCL pulses, the BCOUNT signal sets JCODD to inform the software that the address marker is written. From this point onwards, the data block is written.

**WRITING THE NEXT SYNCHRONISING GAP (Waveform 5)**

After writing the 2 CRC bytes the write mode is virtually completed except for writing the first byte of data and clock bits for the next synchronising gap.

The WMOD2 signal goes high to enable the write data gate. The CRC check word enable input is therefore reset to a high level which prevents any further output from the CRC.

The data written must correspond with the data written in the 11 byte synchronising gap immediately prior to the start of the write mode. If the data to be written is FF, that is, all logic 1's, then the software loads the write data register with all 1's. If the data is 00, then the software loads 00 into the register.

The clock bits written in this byte are always all 1's, therefore the +5V logic level on the input of the write clock register provides the necessary clock bit serial shift.

At the end of this first byte in the next synchronising gap, the JCODD signal informs the processor software that writing is complete. WMOD2 signal is set low which sets the Write Gate signal on J1 pin 11 to a high level. This action ends the write mode and informs the diskette electronics that the writing heads can be unloaded if no further instructions are received within approximately 0.5ms.

The processor software sets the interrupt flip-flop and waits for the main computer program to read the status register.

**END OF WRITE MODE**

When the Write Gate signal goes high, the diskette is set back to the read data mode and is ready for new software instructions. If the track format gap between the last written byte and the 1st byte of data read is too long, causing the IDMODE to be delayed by more than 5μs, the software clears the read logic and starts the re-synchronising cycle.

This error detection signal is JIDMOD which may arise due to diskette speed variation or when the write current decoy has not occurred in the normal time.

**WRITE REGISTER CLOCK**

For the duration of the write
REGCL pulses of 200ns at a
clock bit intervals. The pulse
clock bits to the diskette.
The R/W clear pulse resets th
write mode. After 15 SYSCL
produces the first REGCL pu
The WMODE, DATEN and 7
nals preset the counter input
(binary 6) and load the coun
each subsequent load pulse.
The counter therefore counts
pulses between each carry ou
produce a series of REGCL p
200ns at 200μs intervals.

**READ/WRITE CLEAR**

At the end of the 11 byte read mode gap, the software sets the DATEN signal low for two system clock pulses before initiating the WMODE signal. This produces a low pulse at the output of the R/W clear gate to reset those counters and flip-flops which are in common use for both the read and write modes.
A R/W clear pulse is also generated by software when an error occurs during synchronisation and when reading the identification field format. The processor program uses a combined clock instruction to set both DATEN and WMODE low for one or two system clock pulses. The program then repeats the read function.

**WRITE DATA A**
**WRITING DATA**

When the data bi
all logic 1, therefo
transferred from
write data register
automatically fro
serial shift input
The data bits are
every 4μs and gat
input of the data
fed to the input o
for the generation
Data and clock bi
diskette and the n
monitored and co
JCODD flip-flop
pletion of the dat
and inhibits the d

**FUNCTION MODE LATCH**

After reading the 11 byte synchronization gap at the end of the header format, the processor program initiates a combined clock instruction (Hex. D0) to set the WMODE signal high and the Write Gate signal on J1 pin 11 low. The WMOD2 signal is also set high and remains high except when writing the CRC bytes from the generator at the end of the write data block. The WMOD1 signal is set high after writing the initial 6 byte synchronizing gap, and remains high for the duration of the data block and the CRC characters.

The processor program enables these WMODE signals when the software counts the correct number of jump instruction signals (JCODD) for each track sequence.

**WRITE BYTE COUNTER (Waveform 1)**

The write byte counter counts from 1 to 15 before being automatically reset to zero by the next (16th) input pulse (REGCL). The carry out signal is the BCOUNT pulse which indicates that 1 byte of data has been written from the write registers. Sixteen REGCL input pulses are required to write 1 byte of data, since 1 byte comprises 8 data bits and 8 clock bits.

The BCOUNT pulse sets the jump instruction flip-flop (JCODD). This JCODD pulse informs the processor software that 1 byte has been written. It is the number of times that the JCODD flip-flop is set which allows the program to detect the end of each track format and so control the write mode.

The l.s.b. of the write byte counter is the write data select signal (WDSEL) which, with the REGCL pulse, is used to control the timing of data and clock bits to the diskette.

**WRITE REGISTER CLOCK (REGCL) (Waveform 1)**

For the duration of the write mode, the R/W Counter produces REGCL pulses of 200ns at a rate to simulate the 2μs read data bit-to-clock bit intervals. The pulses control the timing of write data and clock bits to the diskette.

The R/W clear pulse resets the counter to zero at the start of the write mode. After 15 SYSCL pulses, the counter carry output produces the first REGCL pulse and loads the counter to binary 6. The WMODE, DATEN and 7CENB signals preset the counter inputs to 0110 (binary 6) and load the counter to 6 for each subsequent load pulse.

The counter therefore counts 10 SYSCL pulses between each carry out pulse to produce a series of REGCL pulses of 200ns at 200μs intervals.

**WRITE GATE CLOCKING (Waveforms 1 and 2)**

The first pulse to be written to the diskette occurs when the first REGCL is generated from the read/write counter. A high level then exists on each input of the 3-input write clocking gate. The WCLKB is high from the m.s.b. of the already loaded write clock register, the inverted WDSEL signal is high because the l.s.b. of the write byte counter is still low, therefore in the time that the REGCL goes high a clock pulse is written to the diskette via J1 pin 13.

As each subsequent REGCL occurs, the WDSEL signals from the write byte counter provide an alternating clocking sequence for clock and data bits to the diskette. The two 3-input clocking gates are alternately enabled every 2μs to allow either a clock pulse or data pulse to be written. The waveform shows the WDATA pattern during the synchronising gap when clock bits are logic 1 and data is all zero.

**CRC GENERATOR (Waveforms 3 and 4)**

**START OF CRC**

When the WMOD1 signal goes high, the preset input on the CRC also goes high and allows the CRC clock pulse to start clocking in the write data bits. The CRC clock pulse is coincident with the write data register shift clock pulses and occurs when WDSEL and REGCL are both high, that is, on the second REGCL after the JCODD signal. Each bit of data written to the diskette is also fed back to the input of the CRC. No CRC output occurs until the end of the write data block, but the check pattern is being generated continuously within the CRC.

The data bits are clocked into the CRC on the negative going edge of the CRC clock pulse which occurs every 4μs. This time interval allows clock bits to be written between each data bit to the diskette. The check word enable (CWE) input must be held high whilst data is being entered into the CRC, this is ensured by the low signal of WMOD2 on the input of the CWE enable gate.

**APPENDING CRC BYTES (Waveform 4)**

When the WMOD2 signal goes low, at the end of the write data block, the CRC gate in the write data gate logic is enabled and the data gate is inhibited. The chck word enable (CWE) input goes low and allows the 16-bit CRC internal register (holding the CRC generator pattern) to be clocked out and written to the diskette.

The WMOD1 signal enables the CRC output gate and CRC data is gated via the write data gate logic to the 3-input data clocking gate. Two CRC bytes are therefore appended to the data block and are unique to that data block.

**WRITE DATA AND CLOCK REGISTERS**

**WRITING DATA BLOCK**

When the data block is written, the clock bit pulses are all logic 1, therefore only the write data bytes are transferred from the computer and loaded into the write data register. The clock bit pulses are loaded automatically from the fixed +5V logic level on the serial shift input of the write clock register.

The data bits are serially shifted out of the register every 4μs and gated through the write data gate to the input of the data clocking gate. Every data bit is also fed to the input of the CRC. Clock bits are not used for the generation of the CRC pattern.

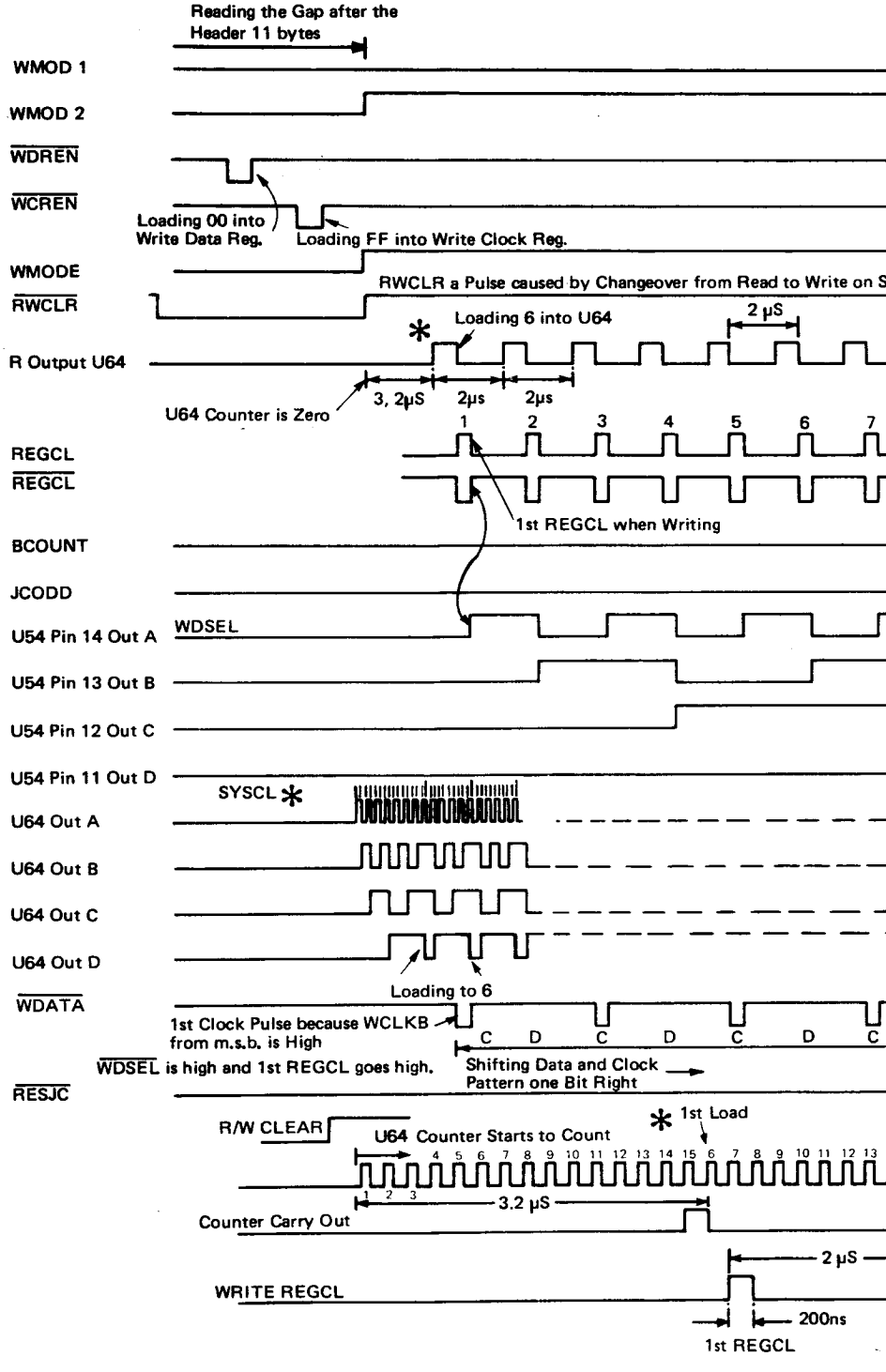
Data and clock bits are alternately written to the diskette and the number of bytes transferred is monitored and controlled via the write byte counter, JCODD flip-flop and the processor software. On completion of the data block the WMOD2 signal goes low and inhibits the data through the write data gate.

**LEAR**

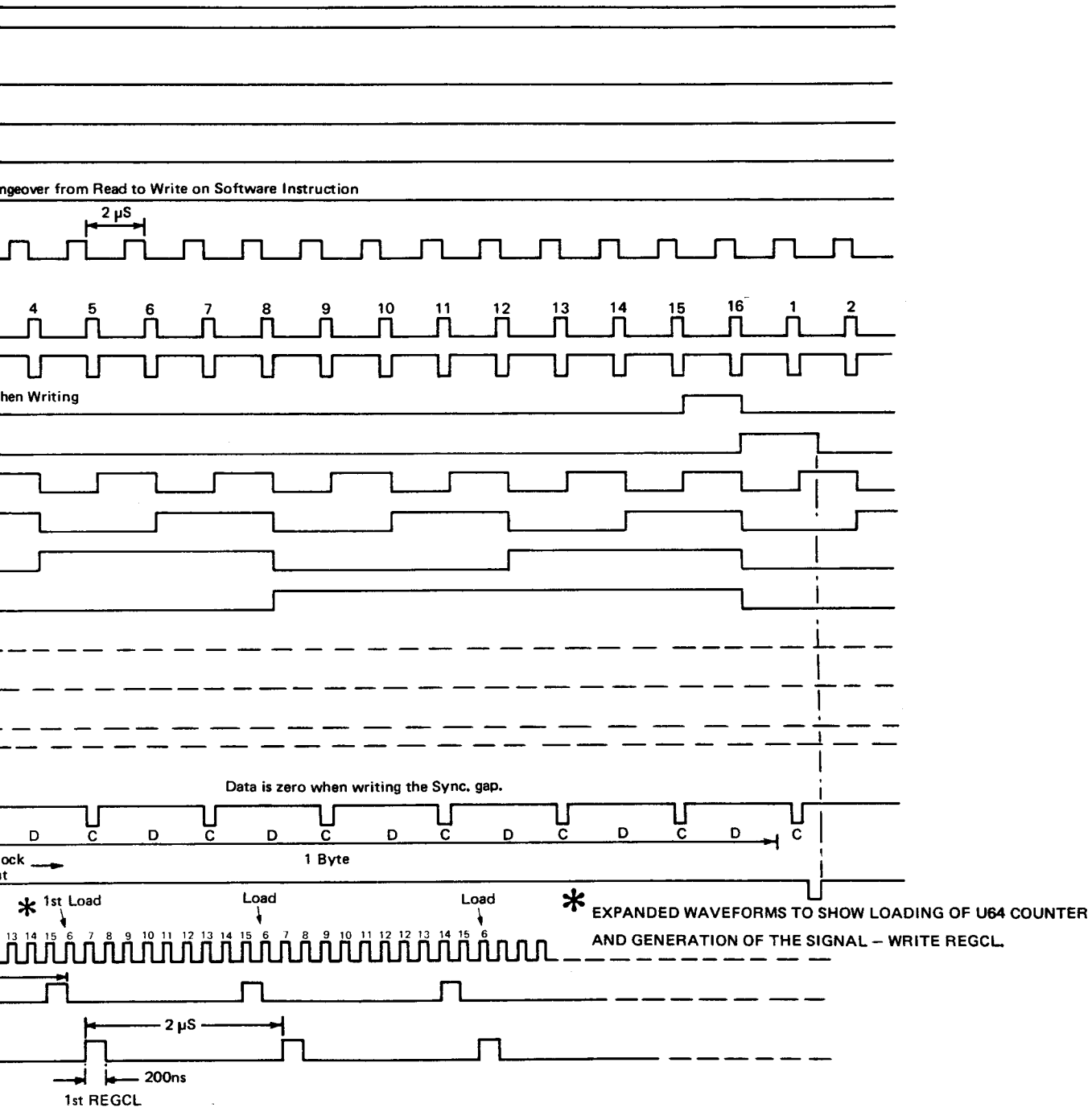
11 byte read mode gap, the DATEN signal low lock pulses before ODE signal. This pro- at the output of the reset those counters ich are in common use and write modes. is also generated by error occurs during nd when reading the ld format. The processor ombined clock instruc- DATEN and WMODE o system clock pulses. repeats the read

**WRITE MODE — Description**

Waveform 1 Wr

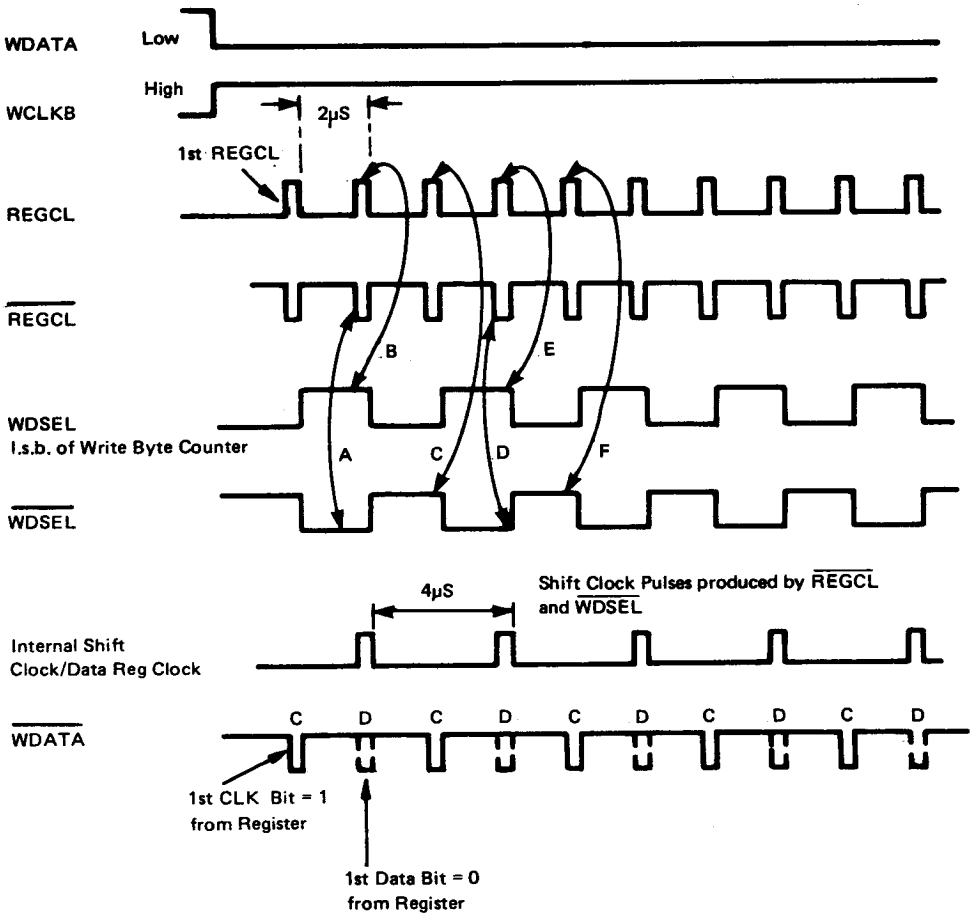


**Waveform 1 Writing 6 byte Sync. Gap**




* EXPANDED WAVEFORMS TO SHOW LOADING OF U64 COUNTER AND GENERATION OF THE SIGNAL – WRITE REGCL.

## Waveform 2 Write Gate Clocking




A Clock and Data Bits Shifted 1 Bit to the Right from the Write Registers

B Data Bit written via Data Clocking Gate

C Clock Bit written via Clock Clocking Gate

D Clock and Data Bits Shifted 1 Bit to the Right from the Write Registers

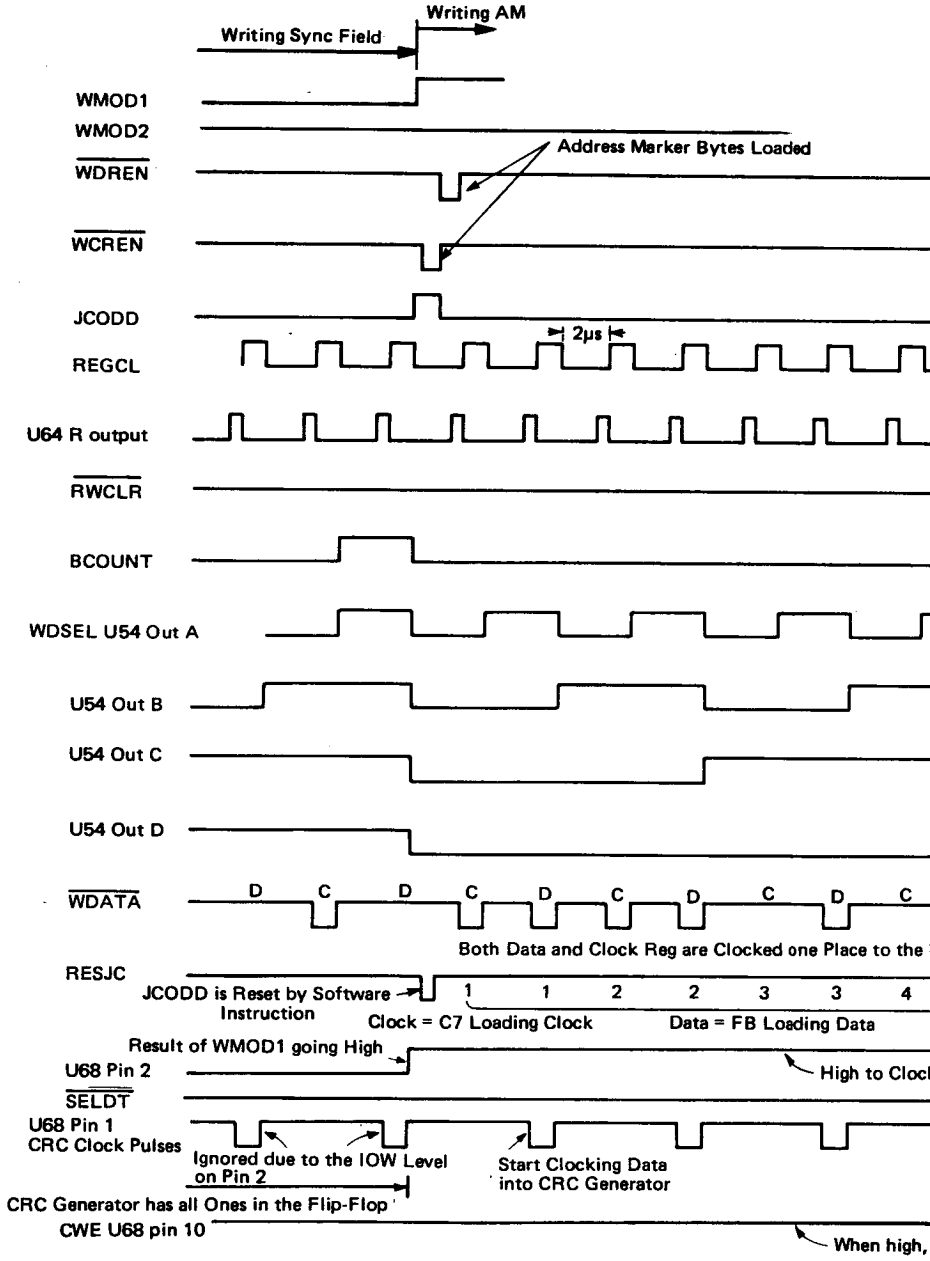
E Data Bit written via Data Clocking Gate

F Clock Bit written via Clock Clocking Gate

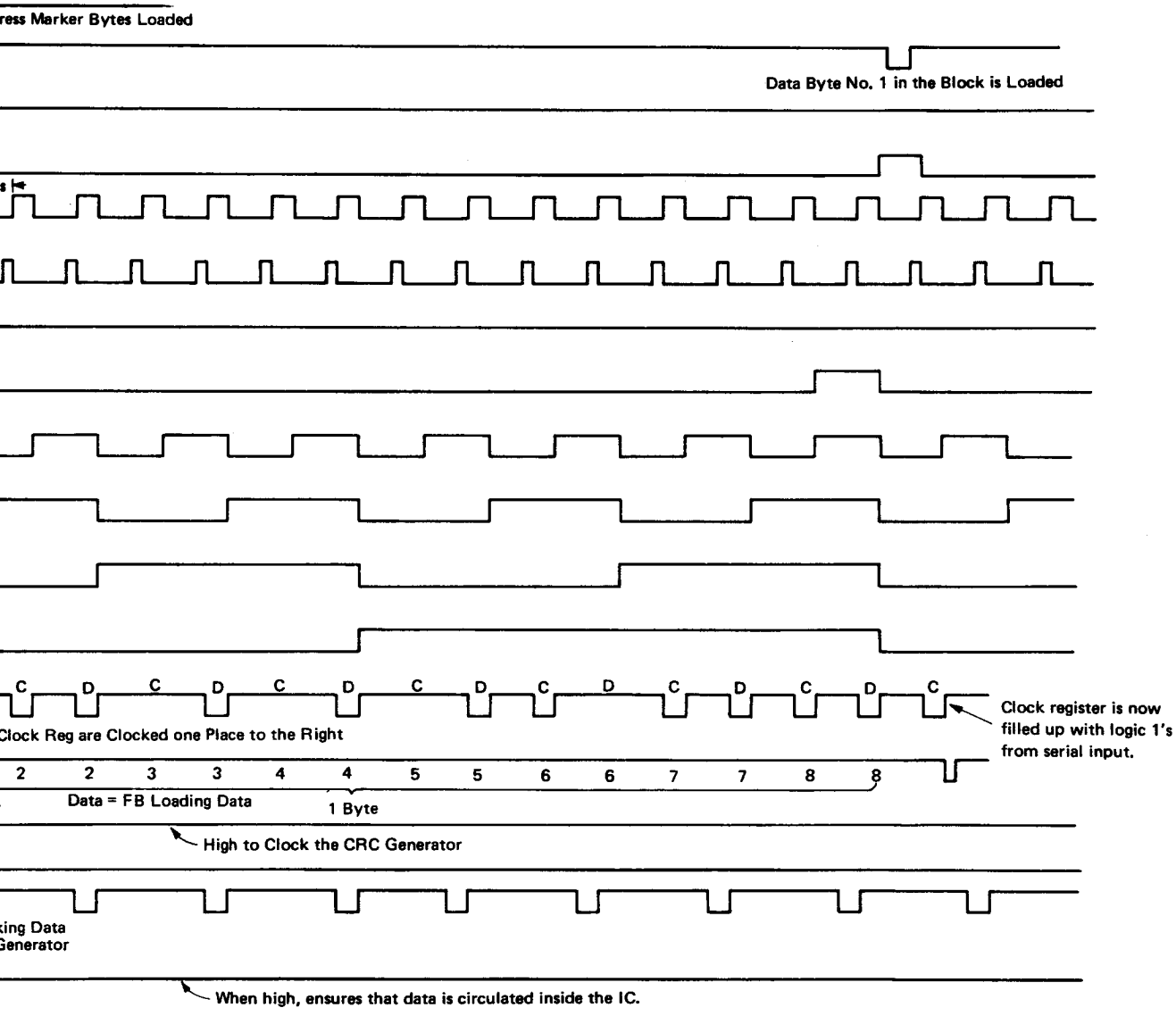
TDV 2114
1426-10-76

TDV 2114
1426-10-76

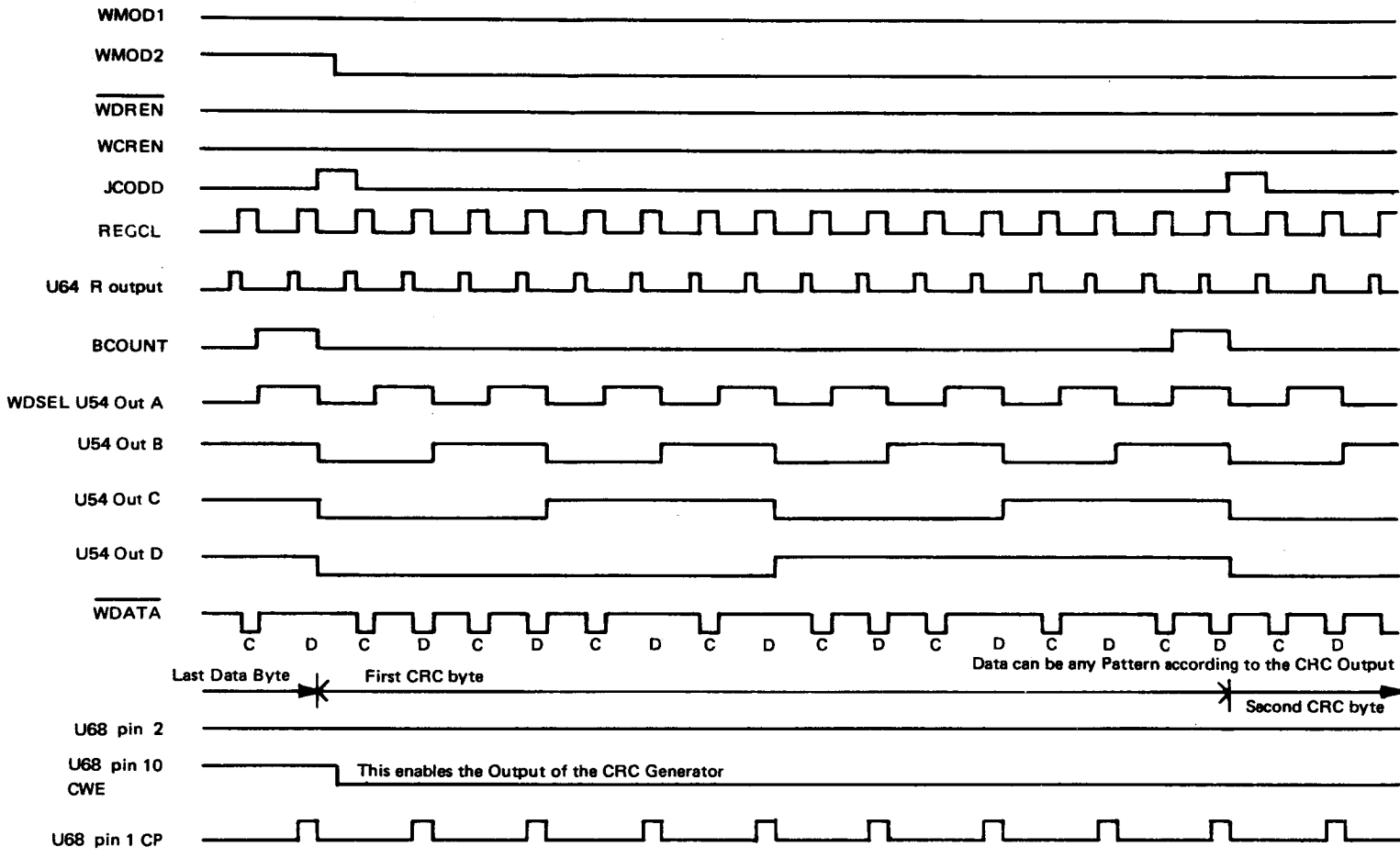
**Waveform**

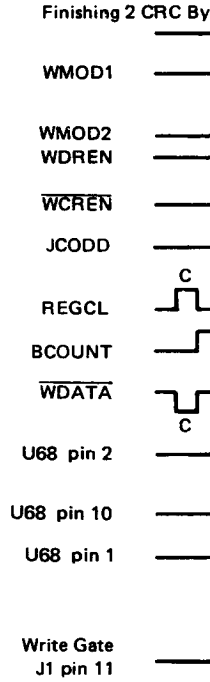


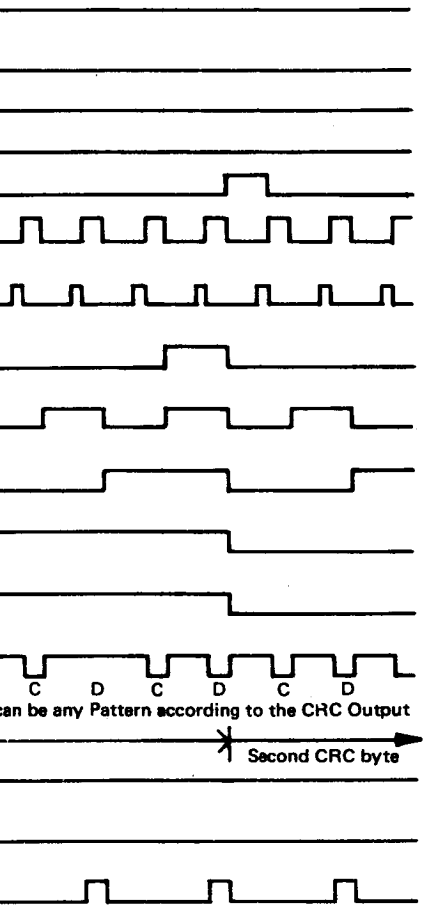
**Waveform 3 Write AM and start of data**



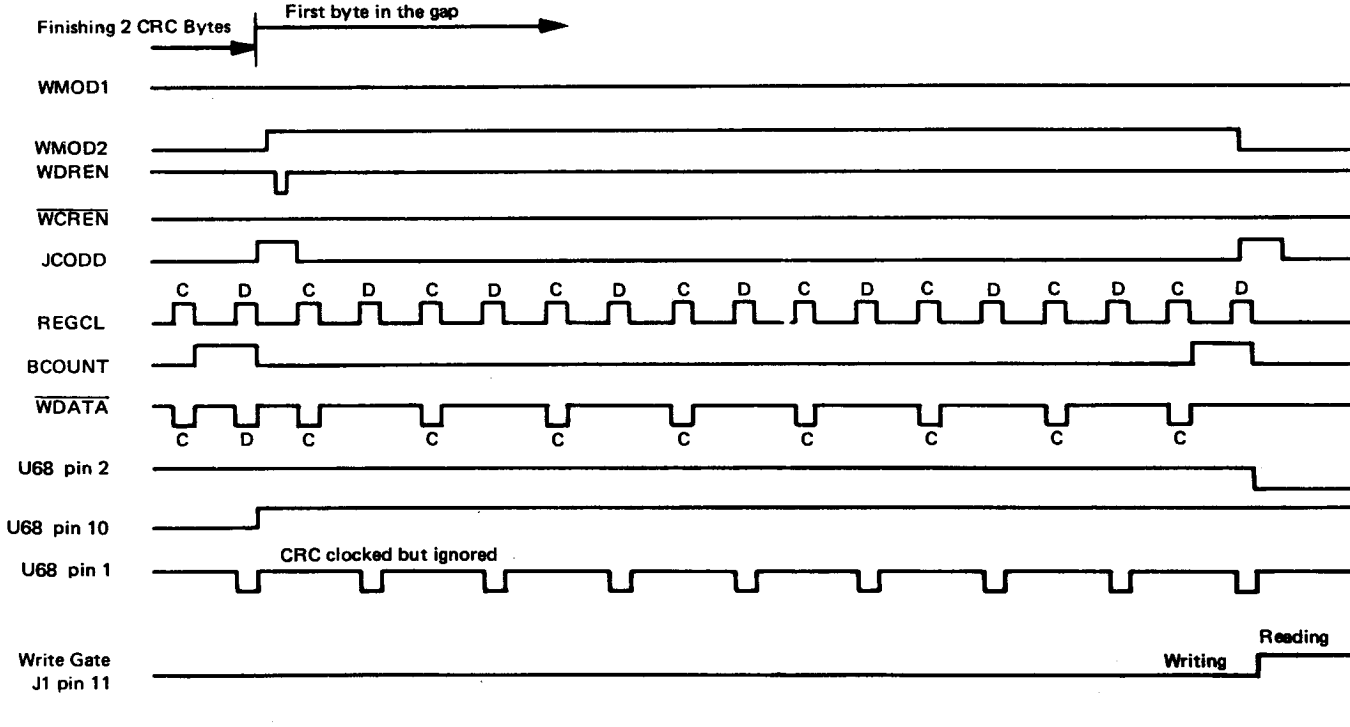
**Waveform 4 Write CRC**

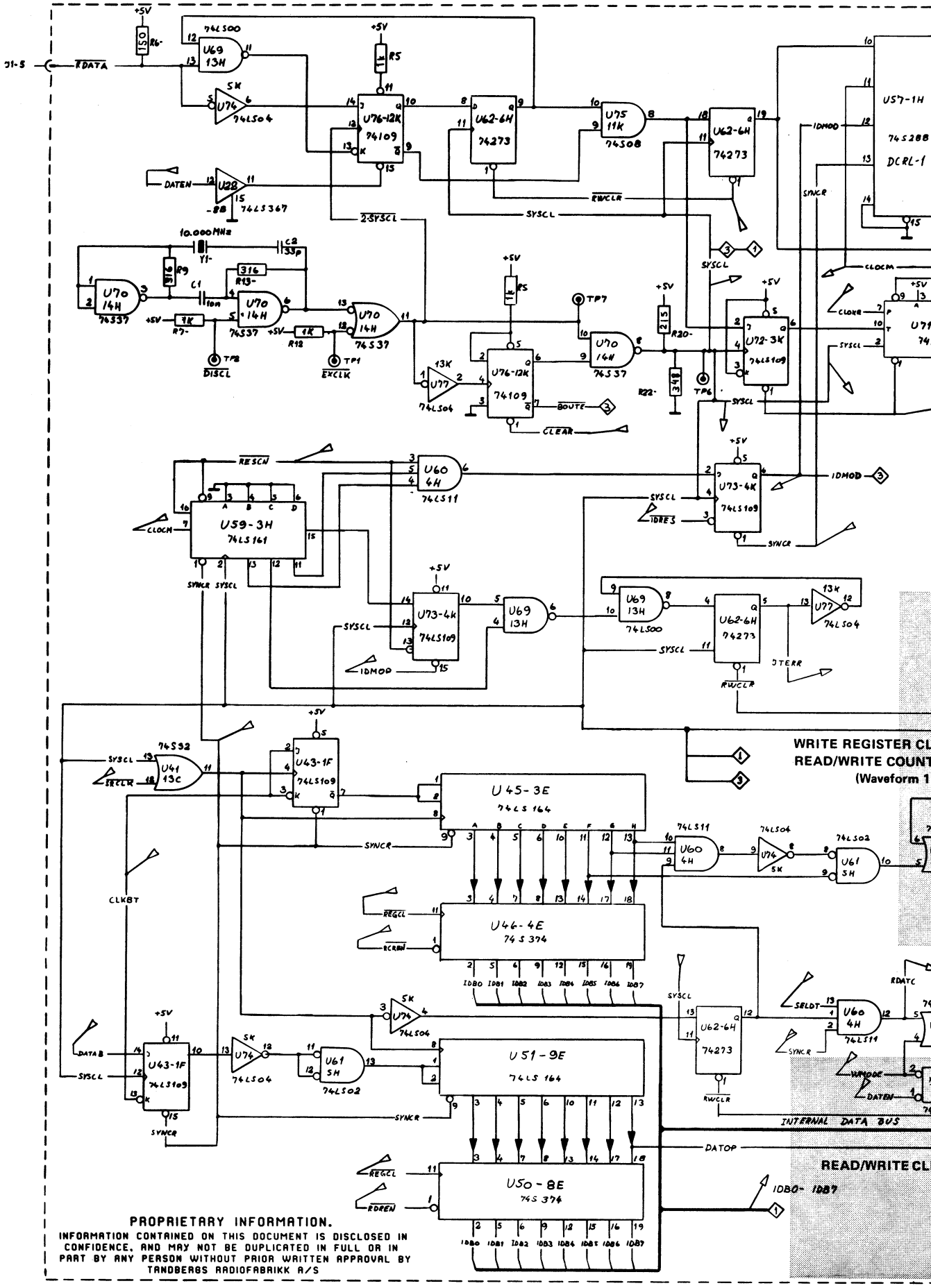


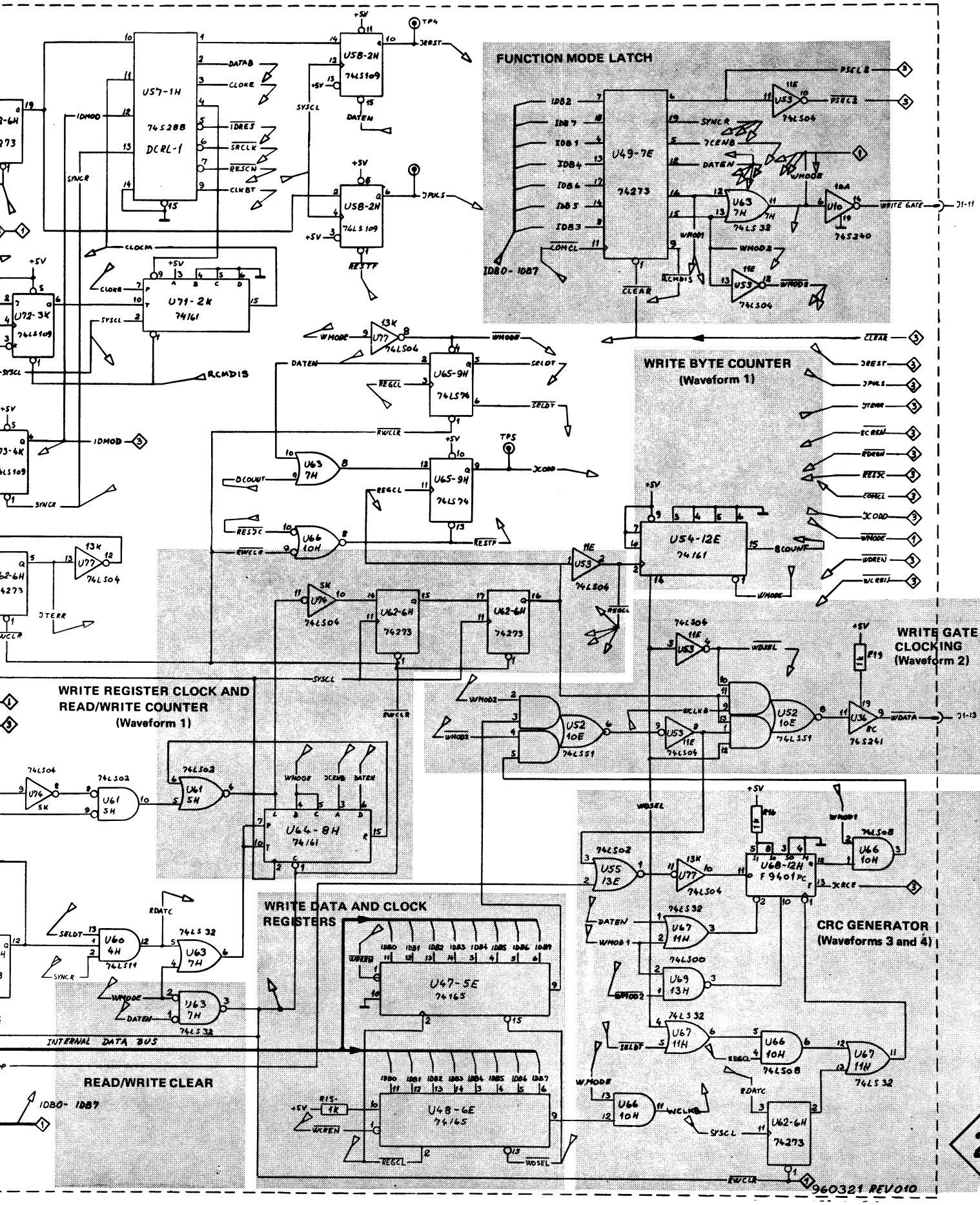






**Waveform 5 End of CRC, start of next gap**




RDATA
DATEN
10.000MHz
DISCL
EXCLK
SYSCL
RWCLR
CLEAR
BOUTE
IDMOD
RESCN
SYNCR
CLOCM
DTERR
WRITE REGISTER CL
READ/WRITE COUNT
(Waveform 1
CLKBT
REGCL
RDREN
DATOP
INTERNAL DATA BUS
READ/WRITE CL
IDB0- IDB7
PROPRIETARY INFORMATION.
INFORMATION CONTAINED ON THIS DOCUMENT IS DISCLOSED IN CONFIDENCE, AND MAY NOT BE DUPLICATED IN FULL OR IN PART BY ANY PERSON WITHOUT PRIOR WRITTEN APPROVAL BY TANDBERGS RADIOFABRIKK A/S




**WRITE MODE — Logic Diagram**

| Position | Name | Part No. | Rev. No. |
|---|---|---|---|
| A1 | Diskette Controller | 960321 | 012 |

## INTERNAL PROCESSOR



TDV 2114
1428 - 10 - 76

## Internal Processor

The processor program operates from a fixed program held in the Read Only Memory (ROM). The program is accessed by the program counter which increments a new address for every system clock pulse. When a valid jump instruction occurs, the memory output is loaded into the program counter which accesses a new section in the ROM. In this way, the program is able to output new instructions and also address the jump selector until it detects the next major change in the program.

The program is divided into sections which control the functions of the diskette. These functions include setting up the diskette with the input/output instructions and commands, synchronising and reading data from the diskette, writing data to the diskette and controlling all transfers between the diskette and the main computer.

TDV 2114
1426 - 10 - 76

The transfer of data within the processor is divided (buffered) into certain logic blocks. These and their associated busses are shown in the following block diagram. The internal selection and address bus and the internal data bus are the only two busses which communicate with the remaining controller logic, namely, the input/output and the read/write logic.

At the start of the input program, the scratch pad register location 8 is set to zero and the status register is reset to hexadecimal 80, i.e. operation completed. The accumulator is cleared and the program is then set up to detect the first instruction loaded into the E register. When the program detects the jump instruction signal JINLD in the jump selector, the program counter is loaded with a new address and the program is accessed to a new instruction.

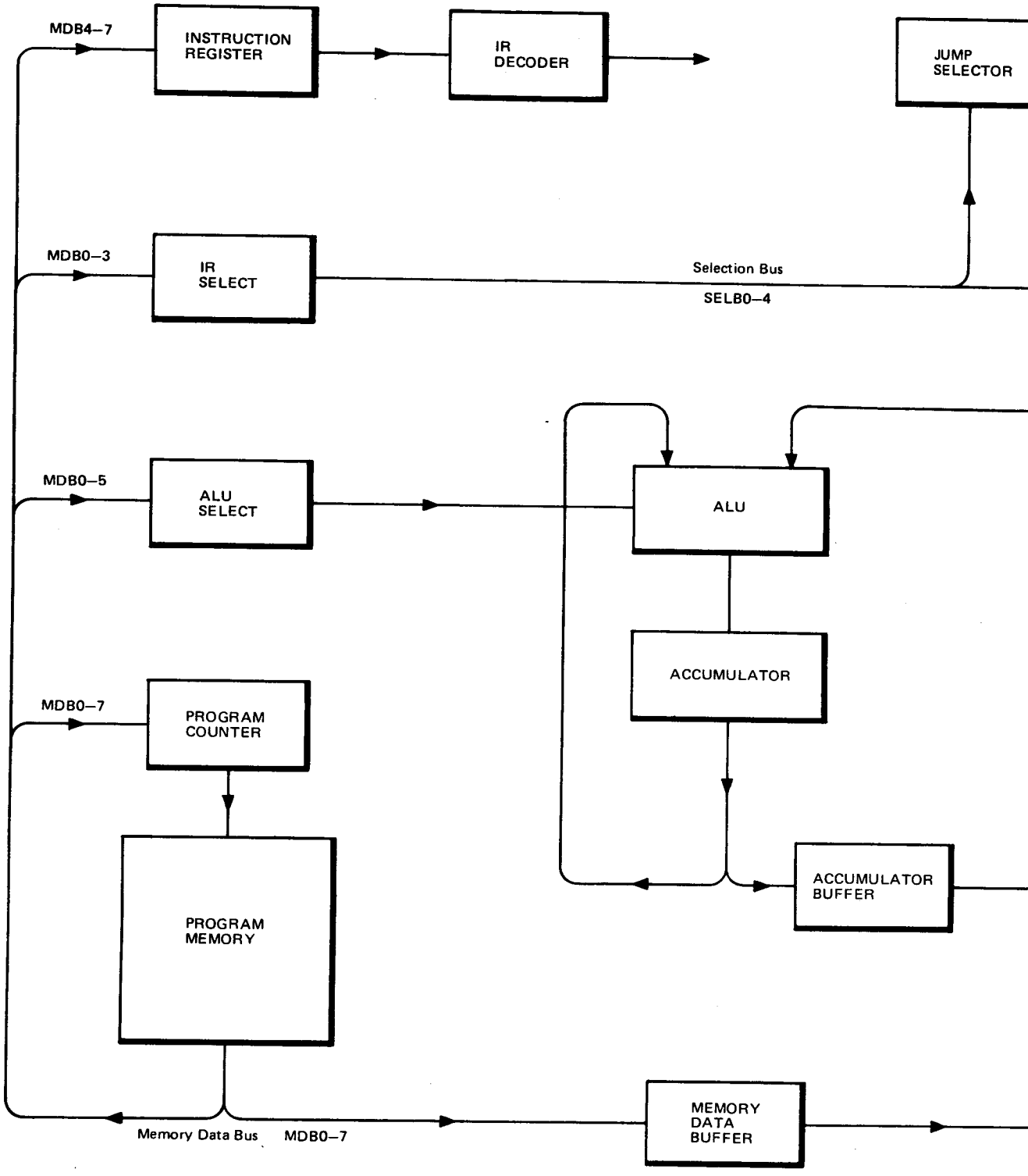
A full step by step account of the controller program is given in the Diskette Software Information section together with detailed block schematic flow charts of the program. This information is crossed referred with named labels so that each section in the program can be identified easily between the mnemonic list and the schematic block flow charts.

## Processor Timing

The processor timing is controlled by the low going SYSCL pulses derived from the crystal controlled oscillator in the read mode input logic. The pulses are 50ns wide in 200ns periods. The SYSCL pulses either clock the registers directly or are AND-ed with enabling signals on input gating logic. For every SYSCL, a new instruction or data is placed on the program memory data bus and executed within 200ns.

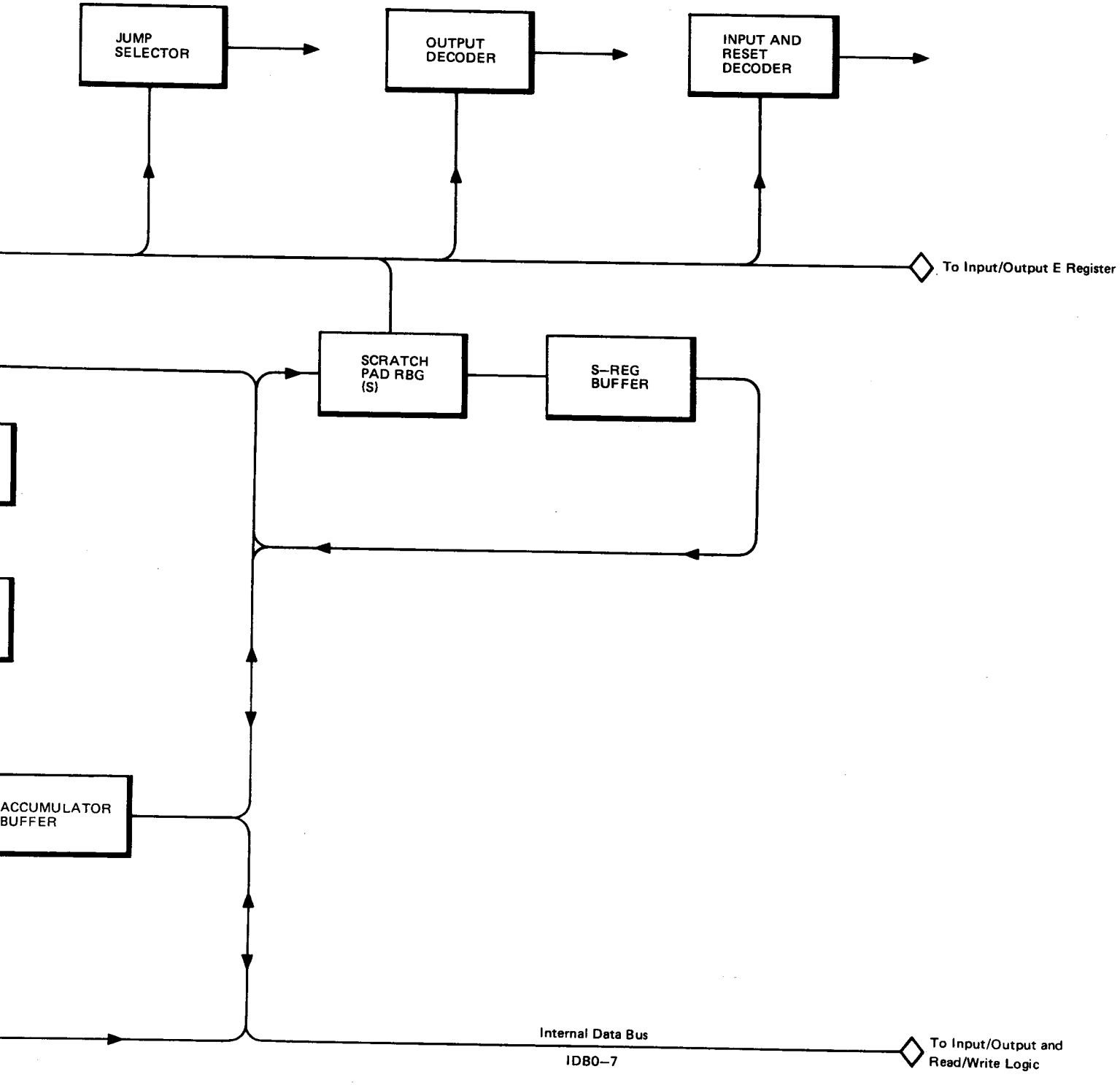
The general clock and combined clock pulses are signals derived from decoding certain instructions in the output decoder logic and are functional rather than timing pulses.

When the processor does not receive any new command from the computer for approximately 500μs from the last command or instruction the computer program clears the FD register in the input/output logic. This disables the selected drive and causes the read/write heads to be unloaded.




TDV 2114
1426 · 10 · 76

INTERNAL PROCESS




**INTERNAL PROCESSOR – Block Diagram**

## INSTRUCTION REGISTER AND DECODER (Tables 1 and 2)

The instruction byte is placed on the memory data bus and routed to three different instruction registers in the same system clock time, these are:

(a) the decoder register : bits 4 - 7
(b) Address selection register : bits 0 - 3
(c) ALU function register : bits 0 - 5

The significance of the instruction bits is shown in the following tables and also listed fully in the Diskette Software Information section.

The instruction bits 4 - 7 are clocked into the four D type flip-flops on the positive going edge of the IRCLK pulse. The Q outputs from the flip-flops are connected directly to the instruction decoder. These four bits define the seven major functions of the processor logic.

The instruction decoder is a read only memory which is electrically pre-programmed to the format listed in Table 1. Sixteen binary input codes provide a combination of signals for seven major processor functions, which are listed in Table 2. The decoder output signals are true when either a logic 0 in Table 1 corresponds with the 'NOT' signal or a logic 1 corresponds with the signal with no bar, for example:-

Binary 5 input 00101 produces the hexadecimal notation of FE which is 1111 1110.
Therefore only the SEBEN signal is true.

## ADDRESS SELECTION REGISTER

This register is a 4-bit D type with a third state high impedance output. It is capable of driving the outputs directly on to the internal selection bus. Memory data bits 0 - 3 are clocked into the flip-flops on the positive going edge of the SYSCL and address the following logic blocks.

(a) The jump instruction selector,
(b) the output decoder,
(c) the input decoder, and.
(d) the scratch pad register.

The internal selection bus is also connected to the E register for transferring interface input addresses to the scratch pad register. When this occurs the SEBEN signal from the instruction register decoder sets the output control pins on the address selection register (U18) high. The register output is then disabled to the high impedance state to ensure that the E to S address transfer cannot be affected by any spurious signals on the rest of the bus.

## ALU FUNCTION SELECT REGISTER

The ALU register contains six D type flip-flops which accept memory data bits 0 - 5. These bits are clocked in on the positive going edge of the SYSCL and contain the information to select the ALU function. Bits 1, 2, 4 and 5 select the function, bit 0 defines whether the function is arithmetic or logic and bit 3 defines whether the arithmetic operation has a carry input.

The processor software table lists the functions used and further information about the selection bits can be obtained from the ALU Manufacturer's data sheet.

## ADDRESSING THE MEMORY PROGRAM

The memory has a maximum of 1024 locations, requiring a 9-bit address bus. The 8-bits are system clocked from the program counter register whilst the 9th bit is provided by an extra flip-flop and clocked by the combined clock. The memory is divided into four sections. Signals PSEL1 (9th bit) and PSEL2 are used, together with the counter output to define the location addresses in the four sections.

Address locations up to 255 are accessed directly by the program counter, but above 255, a combined clock instruction, hexadecimal notation D0, must be given. This instruction defines the internal data bits 0 and 2 which address the higher locations. Signals, PSEL1 and PSEL2 are outputs from the function mode decoder U49.

The addressing is as follows:

| PSEL1 DB0 | PSEL2 DB2 | LOCATION | IC | PROGRAM SECTION |
|---|---|---|---|---|
| 0 | 0 | 0 - 255 | U29 | 0 |
| 1 | 0 | 256 - 511 | U29 | 1 |
| 0 | 1 | 512 - 767 | U15 | 2 |
| 1 | 1 | 768 - 1023 | U15 | 3 |

The data byte which contains the relevant bits 0 and 2 is placed on the internal data bus via the memory data buffer. The combined clock instruction is defined as a 2-byte operation. One byte from the memory program is decoded by the instruction registers to produce the logic control signals, whilst the following (second) byte from the memory is data and is gated through the memory data buffer on to the internal data bus.

The program memory instructions and data bytes are listed fully in the Diskette Software Information section and should be referred to frequently during the processor control logic description.

The program counter is incremented with every system clock pulse SYSCL. During input operations, the system operates in a loop (software) waiting for new instructions and a command. After each cycle in the loop the accumulator is incremented by 1, when the accumulator reaches 255, the selected drive (if any) is disabled (head unload) and the accumulator reset. Then a new loop section takes place, and so on, until a command is transferred. Each loop sections lasts for approximately 0.5ms. When a controlled transfer instruction arrives from the computer, the jump instruction JINLD is set. The program detects JINLD after repeatedly selecting the address to the jump selector during the input software loop. The output signal from the jump selector is the low pulse PLOAD.

The second byte from the memory program for the JINLD instruction is hex. 12, which is loaded into the program counter and over-writes the existing contents. Therefore, on the next SYSCL, the counter addresses the new location and continues to increment the address until either the program is reset to start or a new jump instruction occurs.

## JUMP SELECTOR

The jump instruction selector selects any one of sixteen inputs from a 4-bit binary address on the internal selection bus. Several inputs may be present at the same time but the program only addresses/selects the jump instruction required for the particular current program sequence.

The jump selector is addressed only for a 2-byte operation. This occurs when both JUMPE and IRENB signals are low in order to set the selector strobe low.

The low output signal PLOAD is used to preset, or load, the program counter with data from the memory data bus. This data is the new 'jump' address for the program memory.

Input signals such as JCODD are used by the program to count the number of times that a read or write byte has occurred. The new jump address then provides the instruction which transfers data from the scratch pad register, increments and compares in the ALU logic, and then replaces the data back into the scratch pad. It is the use of these jump instructions which enable the program to control the processor sequence and co-ordinate the logic functions.

## OUTPU

The outp
which ar
from the
the read
logic.
Each de
when the
address
IRENB
general
operatio
ferred to
signal is
associate
are show

## ARITHMETIC AND LOGIC PROCESSOR

The arithmetic and logic processor comprises the arithmetic logic uni
accumulator buffer. These three elements operate on the processor da
to the registers defined by the program.

Memory data bits 0 - 5 select the arithmetic or logic function which t
are listed in the diskette controller instructions contained in the Soft

The data input to the ALU is divided into two parts; the 'A' inputs fr
from the internal data bus. The A and B inputs are compared, anded,
defined by the ALU function. Data which is transferred to the accum
ALU whether or not an arithmetic function is to be carried out. Dire
ALU are clocked back on to the internal data bus through the accum
occur when data is read from the read data register to the accumulato
computer.

For 2-byte operations, the accumulator clocking is inhibited by the I
gate. This ensures that data clocked from a register on to the internal
routed to the ALU and operated upon with the existing information
line.

The accumulator buffer clocking signal, BOUTE, is derived from the
BOUTE signal goes low approximately 500ns before the SYSCL to en
bus is cleared at the start of the SYSCL pulse.

The ALU output signals A = B1, A = B2 and $C_n + 8$ are routed to jun
JA = B and JCARY respectively. These signals occur during the ALU
by the software program.

## ONE AND TWO BYTE PROGRAM OPERATION (Waveform 1)

The output from the program memory is placed on the 8-bit memor
instruction byte, but for some processor functions, a second byte of
instruction byte in order that the function has data on the internal d
Controller functions which require a data byte with the instruction a
Certain second bytes of data are derived from sources other than the
The instruction byte (byte 1) is clocked into three instruction registe
byte for either decoding or buffering to selection busses. The data by
the memory data buffer or loaded into the program counter during a
When the instruction requires a 2-byte operation it is defined in byte
instruction register produces the 2-byte signal from the instruction d
signal is to inhibit the instruction register clock pulse for one SYSCL
of data to be clocked into the memory buffer or program counter wh
instruction decoder signals are therefore valid for two SYSCL period
when the memory data is on the internal data bus and operated upon

A 2-byte operation does not always require data from the memory. T
is transferred via the internal data bus to the accumulator from eithe
clock register or the scratch pad register. The second period is requir
buffer, to be gated on to the data bus so that the ALU can execute t

## JUMP SELECTOR

The jump instruction selector selects any one of sixteen inputs from a 4-bit binary address on the internal selection bus. Several inputs may be present at the same time but the program only addresses/selects the jump instruction required for the particular current program sequence.

The jump selector is addressed only for a 2-byte operation. This occurs when both JUMPE and IRENB signals are low in order to set the selector strobe low.

The low output signal PLOAD is used to preset, or load, the program counter with data from the memory data bus. This data is the new 'jump' address for the program memory.

Input signals such as JCODD are used by the program to count the number of times that a read or write byte has occurred. The new jump address then provides the instruction which transfers data from the scratch pad register, increments and compares in the ALU logic, and then replaces the data back into the scratch pad. It is the use of these jump instructions which enable the program to control the processor sequence and co-ordinate the logic functions.

## OUTPUT DECODER (Table 3)

The output decoder produces the logic signals which are associated with the transfer of data from the memory buffer or accumulator to the read and write logic and the input/output logic.

Each decoder output signal is clocked by SYSCL when the OUTPUT signal is low and the internal address bits 0 - 4 enable the required input. The IRENB signal is AND-ed with bit 4 to inhibit the general clock pulse, GECEN, during a 2-byte operation to allow memory data to be transferred to the internal data bus. The COMCL signal is also a 2-byte operation. The data bits associated with the GECEN and COMCL signals are shown in Table 3.

## INPUT DECODER (Table 4)

The input decoder produces the enablin associated with the transfer of data fro and read clock register to the accumula enabled by the low INPUT signal from with bits 0 and 4 selecting either one o The input instructions are 2-byte opera Table 4.

## RESET DECODER

The reset decoder is enabled by the low the output decoder logic, with internal selecting either one of two outputs. Wh both the outputs are inhibited.

The RESDI signal resets the address reg output E register and the RESJC signal flip-flop in the read/write logic.

## ARITHMETIC AND LOGIC PROCESSOR

The arithmetic and logic processor comprises the arithmetic logic unit (ALU), the accumulator and the accumulator buffer. These three elements operate on the processor data and then transfer the information to the registers defined by the program.

Memory data bits 0 - 5 select the arithmetic or logic function which the ALU is to perform. These functions are listed in the diskette controller instructions contained in the Software Information section.

The data input to the ALU is divided into two parts; the 'A' inputs from the accumulator and the 'B' inputs from the internal data bus. The A and B inputs are compared, anded, subtracted and operated upon as defined by the ALU function. Data which is transferred to the accumulator is always routed through the ALU whether or not an arithmetic function is to be carried out. Direct transfers to the accumulator via the ALU are clocked back on to the internal data bus through the accumulator buffer. Transfers such as these occur when data is read from the read data register to the accumulator prior to a DMA transfer to the main computer.

For 2-byte operations, the accumulator clocking is inhibited by the IRENB signal on the input enabling gate. This ensures that data clocked from a register on to the internal data bus has sufficient time to be routed to the ALU and operated upon with the existing information held on the accumulator-to-ALU bus line.

The accumulator buffer clocking signal, BOUTE, is derived from the crystal controlled oscillator. The BOUTE signal goes low approximately 500ns before the SYSCL to ensure that the data on the accumulator bus is cleared at the start of the SYSCL pulse.

The ALU output signals A = B1, A = B2 and $C_n + 8$ are routed to jump instruction flip-flops to produce JA = B and JCARY respectively. These signals occur during the ALU functions and are processed accordingly by the software program.

## SCRATCH PAD REGISTER (S REGI (Waveform 2)

The scratch pad register is a 16 byte re as an input/output data buffer and a g contains certain functional informatio particular location, for example, locat significant byte of the main memory a diskette is operating. A list of the loca

The locations are addressed from the i are derived from either the function a E register. When selection is made fro write only function (controlled by SE

The four main functions of the scratch shown in Table 6. Signals from the ins decoder are routed to the S register de enabling and clocking signals to either processor program.

A waveform for each data transfer is g signals which actively effect the clocki writing data from the E register to the and IRENB signals do not affect the fu on the waveform diagram.

When data is written into the S registe pad and appears as the complement at buffer enables and inverts the output re-establish the correct binary levels. T low SPROE signal. The output buffer capable of driving directly onto the bu output when in the disabled condition

## ONE AND TWO BYTE PROGRAM OPERATION (Waveform 1)

The output from the program memory is placed on the 8-bit memory data bus and always contains one instruction byte, but for some processor functions, a second byte of data must immediately follow the instruction byte in order that the function has data on the internal data bus on which to operate.

Controller functions which require a data byte with the instruction are defined as 2-byte operations. Certain second bytes of data are derived from sources other than the memory.

The instruction byte (byte 1) is clocked into three instruction registers which accept certain bits of the byte for either decoding or buffering to selection busses. The data byte (byte 2) is either clocked into the memory data buffer or loaded into the program counter during a jump instruction.

When the instruction requires a 2-byte operation it is defined in byte 1, which, when decoded via the instruction register produces the 2-byte signal from the instruction decoder. The effect of this 2-byte signal is to inhibit the instruction register clock pulse for one SYSCL period to allow the second byte of data to be clocked into the memory buffer or program counter when the next SYSCL arrives. The instruction decoder signals are therefore valid for two SYSCL periods, the second period being the time when the memory data is on the internal data bus and operated upon by the decoded instruction.

A 2-byte operation does not always require data from the memory. The exception to this is when data is transferred via the internal data bus to the accumulator from either the read data register, the read clock register or the scratch pad register. The second period is required for data from the accumulator buffer, to be gated on to the data bus so that the ALU can execute the particular mode selected.

e of sixteen
ternal selection
me time but
instruction
sequence.
byte operation.
signals are low

set, or load,
mory data
the program

e program to
rite byte has
des the
ratch pad
U logic, and
pad. It is the
the program
dinate the

## OUTPUT DECODER (Table 3)

The output decoder produces the logic signals which are associated with the transfer of data from the memory buffer or accumulator to the read and write logic and the input/output logic.

Each decoder output signal is clocked by SYSCL when the OUTPUT signal is low and the internal address bits 0 - 4 enable the required input. The IRENB signal is AND-ed with bit 4 to inhibit the general clock pulse, GECEN, during a 2-byte operation to allow memory data to be transferred to the internal data bus. The COMCL signal is also a 2-byte operation. The data bits associated with the GECEN and COMCL signals are shown in Table 3.

## INPUT DECODER (Table 4)

The input decoder produces the enabling signals which are associated with the transfer of data from the read data register and read clock register to the accumulator. The decoder is enabled by the low INPUT signal from the instruction decoder, with bits 0 and 4 selecting either one of the two outputs.

The input instructions are 2-byte operations and are shown in Table 4.

## RESET DECODER

The reset decoder is enabled by the low GENCL signal from the output decoder logic, with internal data bus bits 2 and 7 selecting either one of two outputs. When bit 2 is logic 0, both the outputs are inhibited.

The RESDI signal resets the address register of the input/output E register and the RESJC signal resets the JCODD flip-flop in the read/write logic.

## OR

prises the arithmetic logic unit (ALU), the accumulator and the nts operate on the processor data and then transfer the information

netic or logic function which the ALU is to perform. These functions ructions contained in the Software Information section.

nto two parts; the 'A' inputs from the accumulator and the 'B' inputs B inputs are compared, anded, subtracted and operated upon as ich is transferred to the accumulator is always routed through the ction is to be carried out. Direct transfers to the accumulator via the al data bus through the accumulator buffer. Transfers such as these data register to the accumulator prior to a DMA transfer to the main

clocking is inhibited by the IRENB signal on the input enabling m a register on to the internal data bus has sufficient time to be with the existing information held on the accumulator-to-ALU bus

, BOUTE, is derived from the crystal controlled oscillator. The 500ns before the SYSCL to ensure that the data on the accumulator pulse.

2 and $C_n + 8$ are routed to jump instruction flip-flops to produce signals occur during the ALU functions and are processed accordingly

## SCRATCH PAD REGISTER (S REGISTER) (Tables 5 and 6) (Waveform 2)

The scratch pad register is a 16 byte random access memory which acts as an input/output data buffer and a general storage register. Each location contains certain functional information which is associated only with that particular location, for example, location 4 always contains the least significant byte of the main memory address in whichever mode the diskette is operating. A list of the location contents is given in Table 5.

The locations are addressed from the internal selection bus, bits 0 - 3 and are derived from either the function address selection register or the E register. When selection is made from the E register, the scratch pad is a write only function (controlled by SEBEN).

The four main functions of the scratch pad register, 3 write and 1 read are shown in Table 6. Signals from the instruction register and instruction decoder are routed to the S register decoder gates which produce the enabling and clocking signals to either write or read as determined by the processor program.

A waveform for each data transfer is given in diagram 2 and shows those signals which actively effect the clocking of data. For example, when writing data from the E register to the S register, the INPUT, OUTPUT and IRENB signals do not affect the function and therefore are not shown on the waveform diagram.

When data is written into the S register it is inverted within the scratch pad and appears as the complement at the output. The S register output buffer enables and inverts the output data on to the internal data bus to re-establish the correct binary levels. The output buffer is enabled by the low SPROE signal. The output buffer is a 3-state element which is capable of driving directly onto the bus and providing a high impedance output when in the disabled condition.

## PERATION (Waveform 1)

is placed on the 8-bit memory data bus and always contains one or functions, a second byte of data must immediately follow the tion has data on the internal data bus on which to operate.

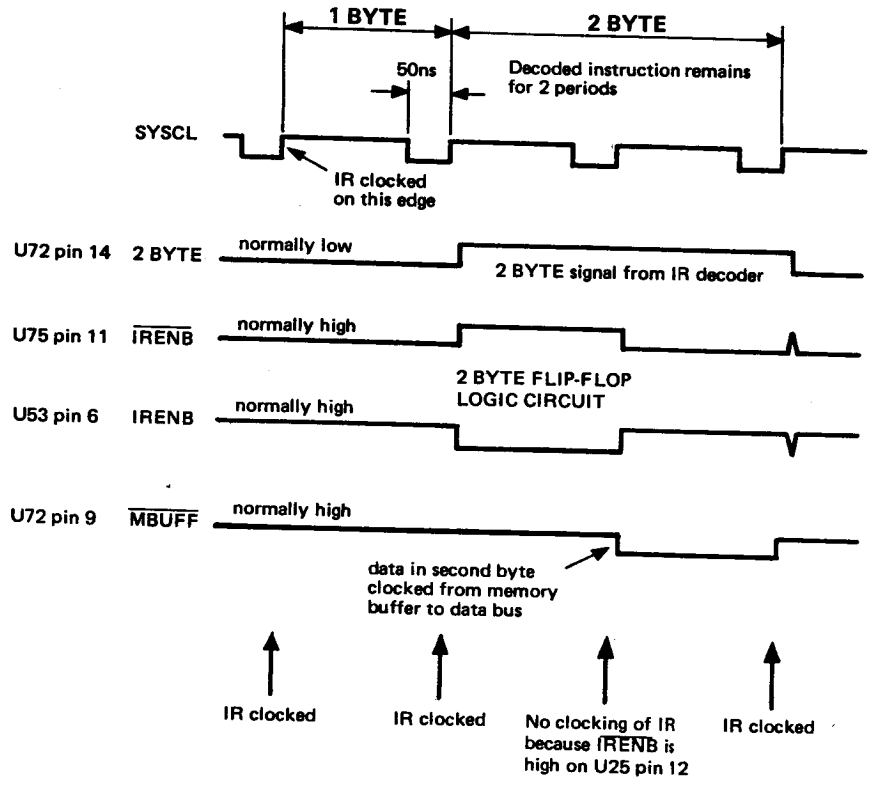
ata byte with the instruction are defined as 2-byte operations. ed from sources other than the memory.

ed into three instruction registers which accept certain bits of the o selection busses. The data byte (byte 2) is either clocked into the program counter during a jump instruction.

operation it is defined in byte 1, which, when decoded via the e signal from the instruction decoder. The effect of this 2-byte ster clock pulse for one SYSCL period to allow the second byte buffer or program counter when the next SYSCL arrives. The re valid for two SYSCL periods, the second period being the time nal data bus and operated upon by the decoded instruction.

equire data from the memory. The exception to this is when data to the accumulator from either the read data register, the read er. The second period is required for data from the accumulator so that the ALU can execute the particular mode selected.

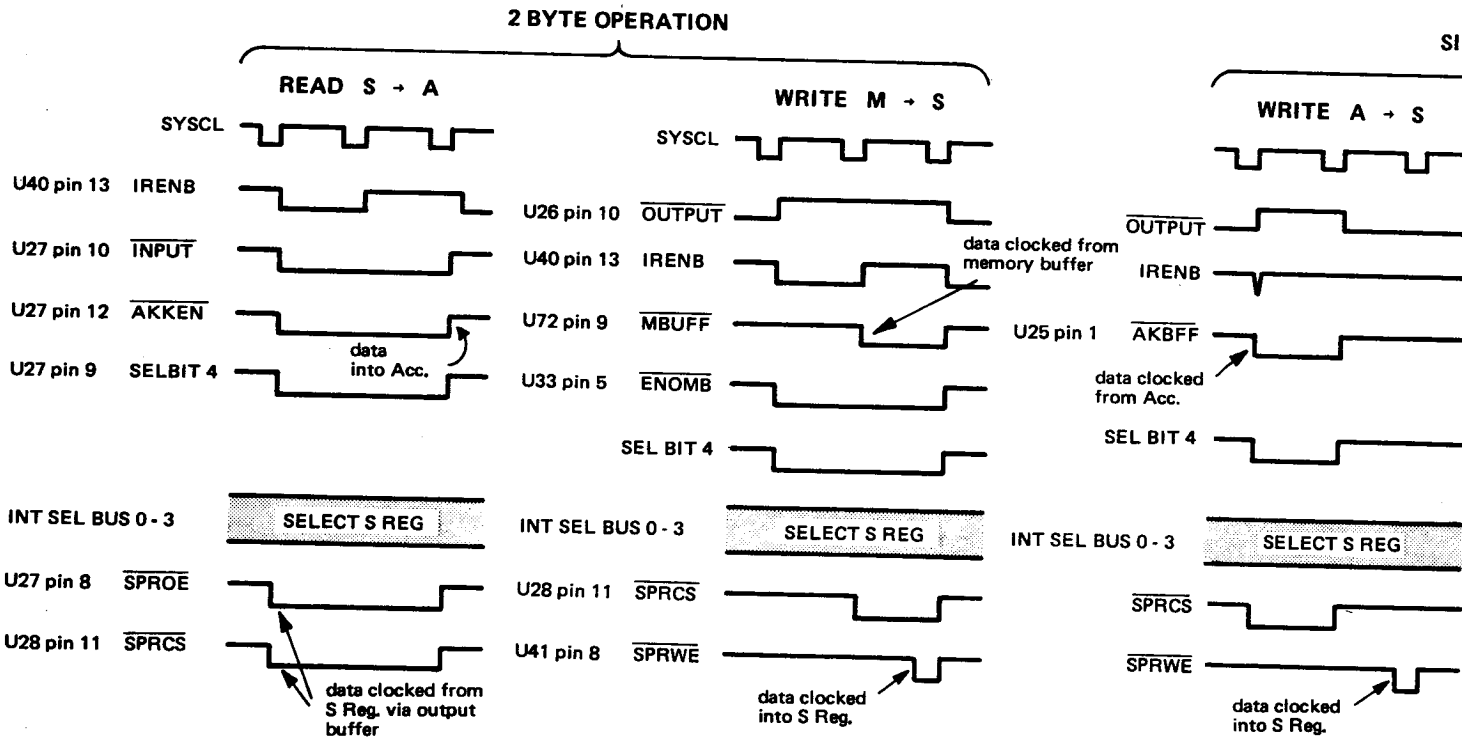
**Waveform 1 2-Byte operation**

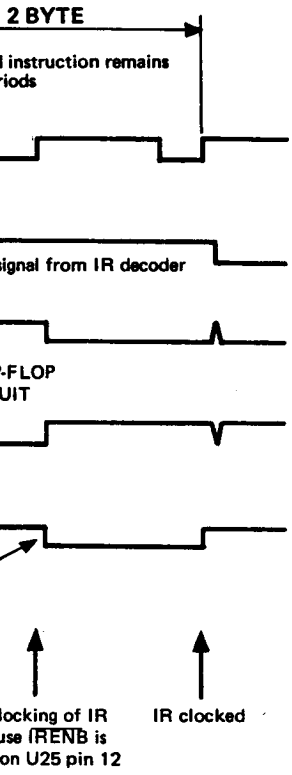



TDV 2114
1426 - 10 - 76

**Waveform 2 Scratch Pad Clocking**

**SCRATCH PAD REGISTER WAVEFORMS**
**(into and from register decoder gates)**






AVEFORMS
der gates)

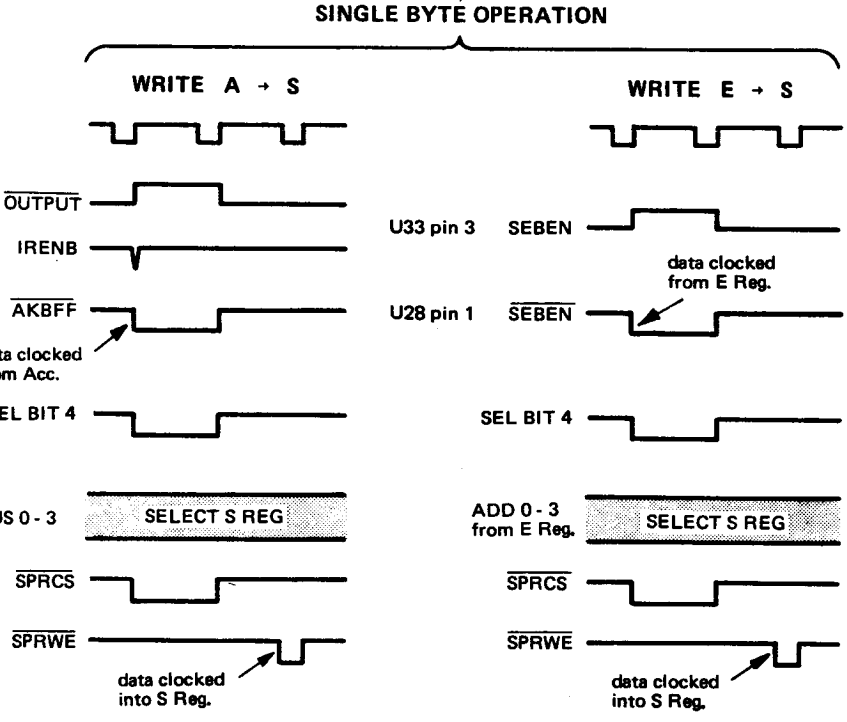



**Table 1**

**IR Decoder Version 5**

| No. | Pin 14 IRBIT4 | Pin 13 IRBIT5 | Pin 12 IRBIT6 | Pin 11 IRBIT7 | Pin 10 | Pin 9 AKBFE | Pin 7 INPUT | Pin 6 AKKEN | Pin 5 ENOMB | Pin 4 OUTPT | Pin 3 SEBEN | Pin 2 JUMPE | Pin 1 2BYTE | Hex not. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | FA |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 72 |
| 2 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | DA |
| 3 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | E3 |
| 4 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | FA |
| 5 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | FE |
| 6 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | CA |
| 7 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 9B |
| 8 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | EB |
| 9 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 72 |
| 10 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | CB |
| 11 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | E3 |
| 12 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | FA |
| 13 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | F9 |
| 14 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | CA |
| 15 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 9B |
| 16 | | | | | | | | | | | | | | |
| 17 | | | | | | | | | | | | | | |
| 18 | | | | | | | | | | | | | | |
| 19 | | | | | | | | | | | | | | |
| 20 | | | | | | | | | | | | | | |
| 21 | | | | | | | | | | | | | | |
| 22 | | | | | | | | | | | | | | |
| 23 | | | | | | | | | | | | | | |
| 24 | | | | | | | | | | | | | | |
| 25 | | | | | | | | | | | | | | |
| 26 | | | | | | | | | | | | | | |
| 27 | | | | | | | | | | | | | | |
| 28 | | | | | | | | | | | | | | |
| 29 | | | | | | | | | | | | | | |
| 30 | | | | | | | | | | | | | | |
| 31 | | | | | | | | | | | | | | |

**Table 2**

**Instruction Decoder**

| Operation | | Byte 1 | Byte 2 | IR Decoder Output | Comments |
|---|---|---|---|---|---|
| Operate M, A → A | 1 | 00 ALUSEL | Data (M7 – M0) | (2 BYTE), ($\overline{\text{ENOMB}}$) | No clocking of acc. Only 1 byte operation when possible |
| Operate M, A → A | 2 | 00 ALUSEL | Data (M7 – M0) | $\overline{\text{AKKEN}}$, (2 BYTE), ($\overline{\text{ENOMB}}$) | Clocking of acc. Only 1 byte operation when possible |
| Operate D, A → A | 3 | 111 D4 → D0 | XXALUSEL | $\overline{\text{AKKEN}}$, $\overline{\text{INPUT}}$ 2 BYTE | |
| Transfer M → D | 4 | 110 D4 → D0 | Data M7 – M0 | $\overline{\text{OUTPUT}}$, 2 BYTE $\overline{\text{ENOMB}}$ | |
| Transfer A → D | 5 | 100 D4 – D0 | | $\overline{\text{AKBFE}}$, $\overline{\text{OUTPUT}}$ | |
| Transfer E → S | 6 | 1010X S2 – S0 | | SEBEN | |
| Jump (un)conditionally | 7 | 1011 $C_3 - C_0$ | Jump Address M7 – M0 | $\overline{\text{JUMPE}}$, 2 BYTE | |

**Table 3**

**Output**

| |
|---|
| COMCL |
| $\overline{\text{COMCL}}$ |
| AHREN |
| $\overline{\text{AHREN}}$ |
| ALREN |
| $\overline{\text{ALREN}}$ |
| DBREN |
| $\overline{\text{DBREN}}$ |
| CLSTR |
| $\overline{\text{CLSTR}}$ |
| FDSEN |
| $\overline{\text{FDSEN}}$ |
| WDREN |
| $\overline{\text{WDREN}}$ |
| WDREN |
| WCREN |
| |
| |
| |
| GENCL |
| $\overline{\text{GENCL}}$ |

| |
|---|
| COMCL 7 |
| SYNCR |
| SYNCR |
| GENCL |
| R2 |
| $R_2$ |

**Table 6**

**Scratch Pad Function Signals – S-Register**

| Function | Data Transfer | Single Byte Operation | | 2-Byte Operation | |
|---|---|---|---|---|---|
| | | From IR Register | S Register Enable/ CLK signal | From IR Register | S Register Enable/ CLK signal |
| WRITE | Accumulator to S Register | $\overline{\text{OUTPUT}}$ $\overline{\text{AKBFE}}$ SELBIT4 (low) Address bits, SELB0-3 | $\overline{\text{SPRCS}}$ $\overline{\text{SPRWE}}$ | | |
| | Memory to S Register | | | 2 BYTE $\overline{\text{OUTPUT}}$ $\overline{\text{ENOMB}}$ SELBIT4 (low) Address bits, SELB0-3 | $\overline{\text{SPRCS}}$ $\overline{\text{SPRWE}}$ |
| | E Register to S Register | SEBEN SELBIT4 (low) Address bits 0 - 3 from E Register | $\overline{\text{SEBEN}}$ $\overline{\text{SPRCS}}$ $\overline{\text{SPRWE}}$ | | |
| READ | S Register to Accumulator | | | 2 BYTE $\overline{\text{INPUT}}$ $\overline{\text{AKKEN}}$ Address bits SELB0-3 | $\overline{\text{SPRCS}}$ $\overline{\text{SPROE}}$ |

**Table 3**

**Output Addresses (CPU → D)**

| | | |
|---|---|---|
| COMCL | A → combined clock | 10010000 |
| $\overline{\text{COMCL}}$ | M → combined clock | 11010000 |
| AHREN | A → Address High Reg. | 10010001 |
| $\overline{\text{AHREN}}$ | M → Address High Reg. | 11010001 |
| ALREN | A → Address Low Reg. | 10010010 |
| $\overline{\text{ALREN}}$ | M → Address Low Reg. | 11010010 |
| DBREN | A → Data Bus out Reg. | 10010011 |
| $\overline{\text{DBREN}}$ | M → Data Bus out Reg. | 11010011 |
| CLSTR | A → Status Reg. | 10010100 |
| $\overline{\text{CLSTR}}$ | M → Status Reg. | 11010100 |
| FDSEN | A → Floppy Select + current Reg. | 10010101 |
| $\overline{\text{FDSEN}}$ | M → Floppy Select + current Reg. | 11010101 |
| WDREN | A → Write Data Reg. | 10010110 |
| $\overline{\text{WDREN}}$ | M → Write Data Reg. | 11010110 |
| WDREN | A → Write Clock Reg. | 10010111 |
| $\overline{\text{WCREN}}$ | M → Write Clock Reg. | 11010111 |
| | A → S | $1000S_3S_2S_1S_0$ |
| | M → S | $1100S_3S_2S_1S_0$ |
| | (E → S) | $1010XS_2S_1S_0$ |
| GENCL | A → General clock | 10011XXX |
| $\overline{\text{GENCL}}$ | M → General clock | 11011XXX |

| COMCL | Byte 2 | | | | | | |
|---|---|---|---|---|---|---|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SYNCR | Write Data 1 | Write Data 2 | Read Data 1 | Read Data 2 | PSEL2 | 7CENB | PSEL1 |
| SYNCR | Write Data 1 | Write Data 2 | Read Data 1 | Read Data 2 | PSEL2 | 7CENB | PSEL1 |
| GENCL | | | | | | | |
| R2 | | | | set DR | $R_1$ | Int Reg | |
| $R_2$ | | | | set DR | $R_1$ | Int Reg | |

**Table 4**

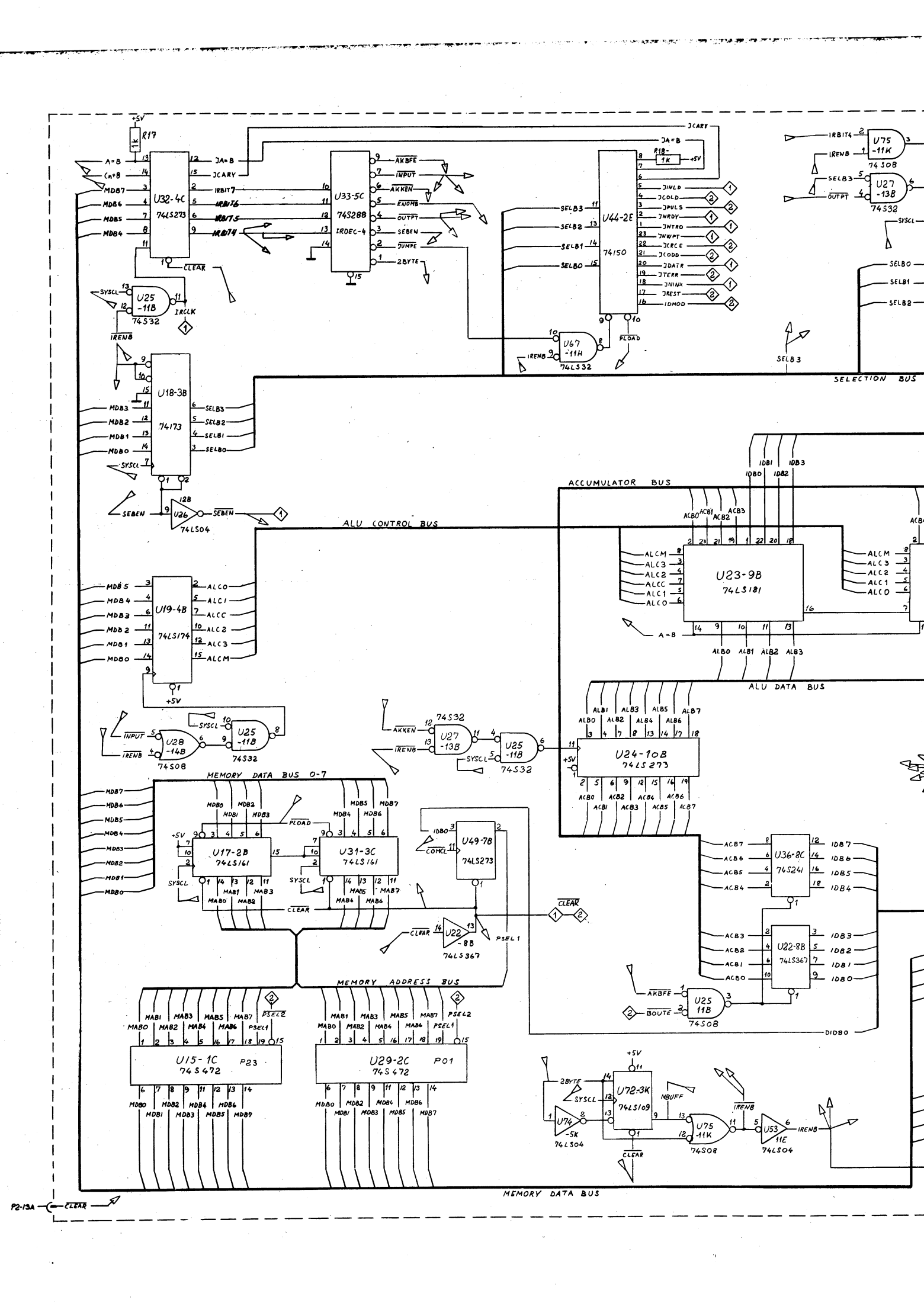
**Input Addresses (D → CPU)**

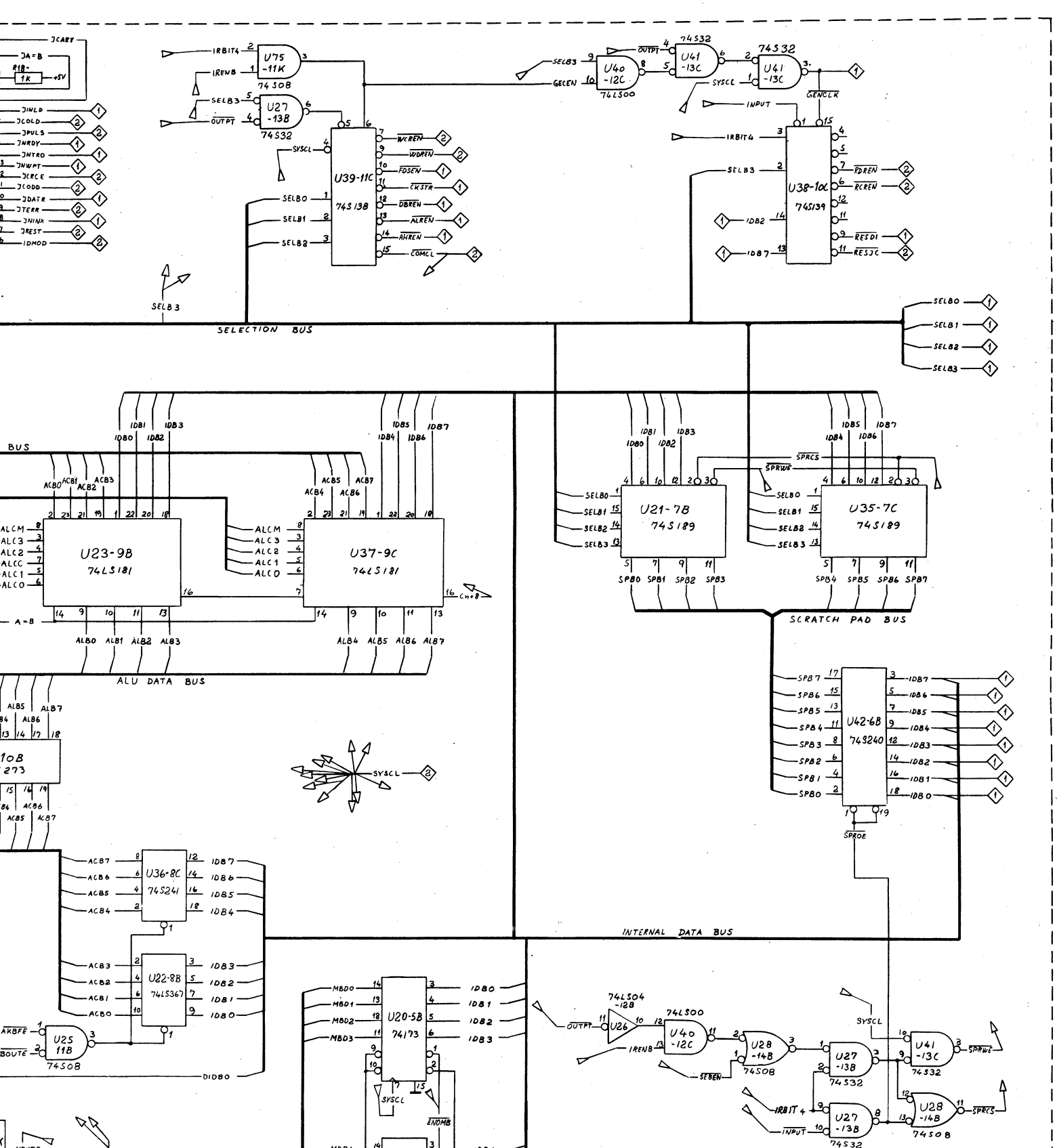
| | | |
|---|---|---|
| $\overline{\text{RCREN}}$ | Clock Register → A | 1111XXX0 |
| $\overline{\text{RDREN}}$ | Data Read Reg → A | 1111XXX1 |
| | S → A | $1110\ S_3S_2S_1S_0$ |

**Table 5**

**Scratch Pad Register Locations**

| Location | Contents |
|---|---|
| S0 : | Drive Address<br>Data during write operation |
| S1 : | Relative track Address<br>Direction of head movement |
| S2 : | Sector Address |
| S3 : | The most significant byte of the main memory address |
| S4 : | The least significant byte of the main memory address |
| S5 : | Block length<br>Clock data during write format operation |
| S6 : | This register contains information to let the software know if the correct header has been read during a read data field operation |
| S7 : | Command register |
| S8 : | Device No., step direction, step pulse |
| S9 & SA : | Counters to count the number of sync. errors and retries. Only S9 is employed with read operation between header and main block. SA is also employed as a general purpose register. |
| SB : | Counts missing sectors etc. Also employed during seek/recalibrate and write format operation |
| SC : | Contains the block length in steps of 128. (0 = 128, 1 = 256, 3 = 512) |
| SD : | Storage for counting index-strobes. Also employed as a general storage register during write operation |
| SE : | Contains the correct header to be written |
| SF : | Contains the information which is going to be sent to the status register |


R17
1k
+5V
U32-4C
74LS273
U33-5C
74S288
U44-2E
74150
U25 -11B
74S32
U67 -11H
74LS32
U75 -11K
74S08
U27 -13B
74S32
U18-3B
74173
U26
74LS04
U19-4B
74LS174
U23-9B
74LS181
U24-10B
74LS 273
U36-8C
74S241
U22-8B
74LS367
U28 -14B
74S08
U17-2B
74LS161
U31-3C
74LS161
U49-7B
74LS273
U15-1C P23
74S472
U29-2C P01
74S472
U72-3K
74LS109
U74 -5K
74LS04
U53 11E
74LS04
SELECTION BUS
ACCUMULATOR BUS
ALU CONTROL BUS
ALU DATA BUS
MEMORY DATA BUS 0-7
MEMORY ADDRESS BUS
MEMORY DATA BUS
P2-13A
CLEAR




960321 REV 012

DISKETTE CONTROLLER MP-CONTROL
SCHEMATIC 3OF3
4-2-77 BoB

92942-1

### Introduction

This section contains sufficient software information to enable the Engineer to step through the program for either learning or diagnostic procedures.

The information is as follows:

1) Input/Output instructions and commands
2) Controller micro-program instructions
3) Controller micro-program flow charts
4) Controller micro-program codes

### Input/Output instructions and commands

The following tables give a summary of the instructions and commands for the diskette controller. Basically, there are 6 output instructions, 1 input instruction, and 7 output commands (including 1 which is optional).

The output instructions are transferred prior to any commands. These instructions contain information about drive address, track address, sector address, record length etc; information which is necessary before any command can be executed.

The output commands are used to execute a write/read or seek operation. When an operation is complete, the controller sets the necessary bits in the status register and the interrupt line is set true. The status register can then be read by an input instruction.

After a command instruction has been transferred, the next output or command instruction must not be given until the interrupt line is set true by the controller.

### Controller m-p instructions

The following tables provide details of the program code mnemonics and define single and 2-byte operations.

### Controller m-p flow charts

The flow charts divide the program into major sections as shown in flow chart 1.

These sections describe the input sequence and lead into the read and write functions. The CRC generator and check functions can be identified and also the setting of the interrupt line.

Write formats 1, 2 and 3 deal with initialisation which is controlled from the main computer.

### Controller program m-p codes

The program codes are numbered in two sections 0 - 511 giving a total of 1024 codes. The program hexadecimal number is prefixed either 0, 1, 2 or 3 which defines the four areas addressed by the hardware counter using signals PSEL1 and PSEL2.

TDV 2114
1426 · 10 · 76

## Output Instructions (CPU - Controller)

| Mnemonic | Name | Address Bus (Hex.) | Data transfer Bus (DB0 - 7) (Positive logic) |
|---|---|---|---|
| DRIVA | Drive Address | xx30 | 7 0<br>0 0 0 0 D4 D3 D2 D1<br>A logic 1 in one of D1-D4 selects one of four drives.<br>Must be given before each command and only one drive can be selected at a time. |
| RELTA | Relative Track Address | xx31 | 7 0<br>R T6 T5 T4 T3 T2 T1 T0<br>The binary number T0→ T6 is the difference between the current and the new track address (T6 most significant bit). R indicates the direction of the movement. To move the head inwards (towards a track with higher number), R is set to 1. Must be given before SEEKT. |
| SECTA | Sector Address | xx32 | 7 0<br>0 0 S5 S4 S3 S2 S1 S0<br>S0 - S5 is the binary sector number, with S5 most significant bit. An all zero number is not valid.<br>Must be given before each REDAT, WRTDA or WTDEL command. |
| ADDHI | Address Register High | xx33 | 7 0<br>A15 A14 A13 A12 A11 A10 A9 A8<br>This instruction is used to transfer the most significant byte of the memory start address to the controller. Must be given before each REDAT, WRTDA or WTDEL command. A15 is the most significant bit of the memory address. |
| ADDLO | Address Register Low | xx34 | 7 0<br>A7 A6 A5 A4 A3 A2 A1 A0<br>This instruction is used to transfer the least significant byte of the memory address to the controller.<br>Must be given before each REDAT, WRTDA or WTDEL command. A0 is the least significant bit of the memory address. |
| CLKBT | Clock Byte | xx35 | 7 0<br>C7 C6 C5 C4 C3 C2 C1 C0<br>This instruction is used to transfer the clock bit pattern of the first byte to be written in a write format operation.<br>Must be given before the WRTFO command. |

| Mnemonic | Name | Address Bus (Hex.) | Data transfer Bus (DB0 - 7) |
|---|---|---|---|
| DATBT | Data Byte | xx36 | 7 … 0<br>D7 D6 D5 D4 D3 D2 D1 D0<br>This instruction is used to transfer the data bit pattern of the first byte to be written in a write format operation. Must be given before the WRTFO command. |

## Input Instruction (Controller → CPU)

| Mnemonic | Name | Address Bus (Hex.) | Data transfer Bus (Positive logic) |
|---|---|---|---|
| SENST | Sense Status | xx37 | 7 … 0<br>OPC DR NVC NR NA CRE SM BSY<br>OPC = Operation completed<br>DR = Deleted Record *<br>NVC = Non-Valid command<br>NR = Drive not Ready<br>NA = No Address Mark<br>CRE = CRC Error<br>SM = Sector missing<br>BSY = Busy<br>NA/CRE = Timing error in read operation.<br>Comment: The drive is ready when the power is on, the floppy diskette inserted and the drive is selected.<br>The Non-Valid command is set whenever the CPU transfers an unrecognisable instruction or the CPU gives a write command and the diskette is file protected.<br>The interrupt line is set true whenever one or more bits in the sense status register is set true, except for the busy status. The SENST instruction can be given at any time. SENST must be read before every new output inst./command because inst./com. must not be given when BSY is true.<br>* DR is set when a deleted address mark is detected. Data is transmitted as usual. After operation is finished both DR and OPC are true.<br>SM/CRE = Timing error during read/write data transfer operation. |

TDV 2114
1426 - 10 - 76

| Mnemonic | Name | Address Bus (Hex.) | Data transfer Bus (DB0 - 7) (Positive logic) |
|---|---|---|---|
| WRTDA | Write Data field | xx37 | 7 0<br>1 1 1 1 0 0 0 1<br>Hex.: F1<br>Prior to the transfer of this command, it is necessary to carry out the ADDHI, ADDLO, and SECTA instructions. |
| WTDEL | Write Deleted Data | xx37 | 7 0<br>1 1 1 1 0 0 1 0<br>Hex.: F2<br>With this command the controller writes a deleted ID mark for the data field.<br>Prior to the transfer of this command, it is necessary to carry out the ADDHI, ADDLO, and SECTA instructions. |
| WRTFO | Write format (optional) | xx37 | 7 0<br>1 1 1 1 1 0 1 1<br>Hex.: FB<br>This command is employed to initialise an entire track. Prior to the transfer of this command, it is necessary to carry out the ADDHI, ADDLO, CLKBT and DATBT instructions.<br>In the execution of this command, the data and clock bit patterns for the whole track must be provided for by the main memory/CPU.<br>The controller starts by writing the last 45 bytes of the pre-index gap. The clock bit pattern for the first of these bytes must be loaded into the S5 register (with a CLKBT operation) and the data bit pattern for the same byte must be loaded into the S6 register (with a DATBT operation) before the WRTFO command is given. The clock bit pattern for the next byte must be in the first memory location (the location number is given by the ADDHI and ADDLO instructions, and the data bit pattern for this byte must be in the next (+1) memory location. Then comes the next clock bit pattern and so on.<br>Nominally, there are 5208 bytes of clocks and 5208 bytes of data to be written into one track. The write operation stops 1 byte after the leading edge of the index strobe has been detected. Nominally, this is after the first 275 bytes of the pre-index gap. Due to speed tolerances, the main memory should contain at least 100 bytes of gap data and 100 bytes of gap clocks, to compensate for worst case speed variation. |

| Mnemonic | Name | Address Bus (Hex.) | Data transfer Bus (DB0 - 7) (Positive logic) |
|---|---|---|---|
| REDAT | Read Data field | xx37 | 7 0<br>1 1 1 0 0 0 0 0<br>Hex.: E0<br>Prior to the transfer of this command, it is necessary to carry out the ADDHI and ADDLO and SECTA instructions. |
| REDID | Read ID | xx37 | 7 0<br>1 1 1 0 0 1 1 0<br>Hex.: E6<br>This command is used to read an ID-mark. 5 bytes are transferred to CPU:<br>Byte Contents<br>1. ID Address Mark<br>2. Track address (binary)<br>3. Binary zero<br>4. Sector address (binary)<br>5. Record length (normally zero).<br>Prior to the transfer of this command, it is necessary to carry out the ADDHI and ADDLO instruction |
| SEEKT | Seek Track | xx37 | 7 0<br>1 1 1 0 0 0 1 1<br>Hex.: E3<br>This command is used to move the head to a new track. Prior to this command, it is necessary to carry out the RELTA instruction |
| RECAL | Recalibrate | xx37 | 7 0<br>1 1 1 0 1 1 0 0<br>Hex.: EC<br>This command moves the head to track 00. |

TDV 2114
1426 - 10 - 76

# Diskette Controller mp - Instructions

NOTE: NOT ALL THE FOLLOWING INSTRUCTIONS ARE USED BUT CAN BE MADE AVAILABLE BY SOFTWARE IF REQUIRED

| Operation | Mnemonics | Symbolic | Byte 1 7 6 5 4 3 2 1 0 | Hex. not. | Byte 2 7 6 5 4 3 2 1 0 | Hex. not. | Comments |
|---|---|---|---|---|---|---|---|
| ACC operation 1 byte | Clear A | 0 → A | 0 1 1 1 1 0 0 1 | 79 | | | |
| | Set one | 1 → A | 0 1 0 0 0 1 1 1 | 47 | | | |
| | Inv. A | $\overline{A}$ → A | 0 1 0 0 0 0 0 1 | 41 | | | |
| | Inc. A | A+1 → A | 0 1 0 0 0 0 0 0 | 40 | | | |
| | Dec. A | A-1 → A | 0 1 1 1 1 1 1 0 | 7E | | | |
| | Shift L | A+A → A | 0 1 0 0 1 1 1 0 | 4E | | | |
| | Shift Linc | A+A+1 → A | 0 1 0 0 0 1 1 0 | 46 | | | |
| Memory to ACC operation 2 bytes | Copy M | M → A | 0 1 0 1 0 0 1 1 | 53 | Mem. data | XX | |
| | Copy $\overline{M}$ | $\overline{M}$ → A | 0 1 1 0 0 1 0 1 | 65 | Mem. data | XX | |
| | Add M | A+M → A | 0 1 1 0 1 0 1 0 | 6A | Mem. data | XX | |
| | Addinc M | A+M+1 → A | 0 1 1 0 0 0 1 0 | 62 | Mem. data | XX | |
| | Sub M | A-M → A | 0 1 0 1 0 1 0 0 | 54 | Mem. data | XX | |
| | Subdec M | A-M-1 → A | 0 1 0 1 1 1 0 0 | 5C | Mem. data | XX | |
| | OR M | A∪M → A | 0 1 0 1 1 1 1 1 | 5F | Mem. data | XX | |
| | XOR M | A + M → A | 0 1 0 1 1 1 0 1 | 5D | Mem. data | XX | |
| | XORi M | $\overline{A + M}$ → A | 0 1 1 0 0 0 1 1 | 63 | Mem. data | XX | |
| | OR $\overline{M}$ | A∪$\overline{M}$ → A | 0 1 1 0 0 1 1 1 | 67 | Mem. data | XX | |
| Device to ACC operation 2 bytes (INPUT instruction) | COPY D | D → A | 1 1 1 D4 – D0 | – | 0 0 0 1 0 0 1 1 | 13 | Device addresses: Read Clock register: F0 (1111 0000) Read Data register F1 (1111 0001) S-register E0 → EF. (1110 S3 – S0) |
| | COPY $\overline{D}$ | $\overline{D}$ → A | 1 1 1 D4 – D0 | – | 0 0 1 0 0 1 0 1 | 25 | |
| | Add D | A+D → A | 1 1 1 D4 – D0 | – | 0 0 1 0 1 0 1 0 | 2A | |
| | Addinc D | A+D+1 → A | 1 1 1 D4 – D0 | – | 0 0 1 0 0 0 1 0 | 22 | |
| | Sub D | A-D → A | 1 1 1 D4 – D0 | – | 0 0 0 1 0 1 0 0 | 14 | |
| | Subdec D | A-D-1 → A | 1 1 1 D4 – D0 | – | 0 0 0 1 1 1 0 0 | 1C | |
| | And D | A∩D → A | 1 1 1 D4 – D0 | – | 0 0 1 1 0 0 1 1 | 33 | |
| | OR D | A∪D → A | 1 1 1 D4 – D0 | – | 0 0 0 1 1 1 1 1 | 1F | |
| | XOR D | A+D → A | 1 1 1 D4 – D0 | – | 0 0 0 1 1 1 0 1 | 1D | |
| | XORi D | $\overline{A+D}$ → A | 1 1 1 D4 – D0 | – | 0 0 1 0 0 0 1 1 | 23 | |
| | AND $\overline{D}$ | A∩$\overline{D}$ → A | 1 1 1 D4 – D0 | – | 0 0 1 1 0 1 0 1 | 35 | |
| | OR $\overline{D}$ | A∪$\overline{D}$ → A | 1 1 1 D4 – D0 | – | 0 0 1 0 0 1 1 1 | 27 | |
| Test operation No clocking of acc 2 bytes | TEQ M | A = M → ? | 0 0 0 1 1 1 0 0 | 1C | Mem. data | XX | Test if A = B, with JA = B instr. |
| | TGTR M | A > M → ? | 0 0 0 1 1 1 0 0 | 1C | Mem. data | XX | Test if A > M with JCary inst. A > M if JCary = 0 |
| | TLSS M | A < M → ? | 0 0 0 1 0 1 0 0 | 14 | Mem. data | XX | Test if A < M with JCary inst. A < M if JCary = 1 |
| Jump operation 2 bytes | JUNC | Jump uncond. | 1 0 1 1 0 0 0 0 | B0 | Address | XX | |
| | JEQ | Jump if A = M | 1 0 1 1 0 0 0 1 | B1 | Address | XX | |
| | JCary | Jump if carry = 1 | 1 0 1 1 0 0 1 0 | B2 | Address | XX | |
| | JINLD | Jump if INLD = 1 | 1 0 1 1 0 0 1 1 | B3 | Address | XX | Jump if input reg. loaded |
| | JCOLD | Jump if COLD = 1 | 1 0 1 1 0 1 0 0 | B4 | Address | XX | Jump if command reg. loaded |
| | JPULS | Jump if PULS = 1 | 1 0 1 1 0 1 0 1 | B5 | Address | XX | Jump if puls is set |
| | $\overline{JNRDY}$ | Jump if $\overline{RDYS}$ = 1 | 1 0 1 1 0 1 1 0 | B6 | Address | XX | Jump if diskette is not ready |
| | JNTRO | Jump if $\overline{TRCO}$ = 1 | 1 0 1 1 0 1 1 1 | B7 | Address | XX | Jump if head is not on track 00 |
| | JNWPT | Jump if WPRT = 1 | 1 0 1 1 1 0 0 0 | B8 | Address | XX | Jump if diskette is not write protected |
| | JCRCE | Jump if CRCE = 1 | 1 0 1 1 1 0 0 1 | B9 | Address | XX | Jump if CRC error |
| | JCODD | Jump if CODD = 1 | 1 0 1 1 1 0 1 0 | BA | Address | XX | Jump if clock identification is detected (data) or 8 bit clock set |
| | JDATR | Jump if DATR = 1 | 1 0 1 1 1 0 1 1 | BB | Address | XX | Jump if Data Request is set |
| | JTERR | Jump if TERR = 1 | 1 0 1 1 1 1 0 0 | BC | Address | XX | Jump if read timing error is set |
| | JNINX | Jump if INDX = 1 | 1 0 1 1 1 1 0 1 | BD | Address | XX | Jump if index has not been detected |
| | JIDMD | Jump if IDMOD=1 | 1 0 1 1 1 1 1 1 | BF | Address | XX | Jump if IDMOD=1 |
| | JREST | Jump if JREST = 1 | 1 0 1 1 1 1 1 0 | BE | Address | XX | Jump if Reset sync is set |

TDV 2114
1426 - 10 - 76

| Operation | Mnemonics | Symbolic | Byte 1 7 6 5 4 3 2 1 0 | Hex. not. | Byte 2 7 6 5 4 3 2 1 0 | Hex. not. | Comments |
|---|---|---|---|---|---|---|---|
| Transfer operation<br>M → D<br>2 bytes | MTHADR | M → HADRG | 1 1 0 1 0 0 0 1 | D1 | Address | XX | High Address Reg. |
| | MTLADR | M → LADRG | 1 1 0 1 0 0 1 0 | D2 | Address | XX | Low Address Reg. |
| | MTDBUSR | M → TDBUSR | 1 1 0 1 0 0 1 1 | D3 | Data | XX | Data bus output reg. |
| | MTSTAR | M → STATR | 1 1 0 1 0 1 0 0 | D4 | Data | XX | Data to status reg. |
| | MTFLOPR | M → FLOPR | 1 1 0 1 0 1 0 1 | D5 | Data | XX | Data to diskette drive reg.<br>Bit 0 = select 1<br>Bit 1 = select 2<br>Bit 2 = select 3<br>Bit 3 = select 4<br>Bit 5 = step pulse<br>Bit 7 = step direction |
| | MTWDATR | M → WDATR | 1 1 0 1 0 1 1 0 | D6 | Data | XX | Data to Write reg. |
| | MTWCLKR | M → WCLKR | 1 1 0 1 0 1 1 1 | D7 | Data | XX | Data to Write clock reg. |
| | MTSREG | M → SREG | 1 1 0 0 S3 – S0 | CX | Data | XX | Data from memory to S-reg. |
| Transfer operation<br>A → D<br>1 byte | ATHADR | A → HADR | 1 0 0 1 0 0 0 1 | 91 | | | Acc. to high address reg. |
| | ATLADR | A → LADR | 1 0 0 1 0 0 1 0 | 92 | | | Acc. to low address reg. |
| | ATBUSR | A → DBUSR | 1 0 0 1 0 0 1 1 | 93 | | | Acc. to data bus out reg. |
| | ATSTAR | A → STATR | 1 0 0 1 0 1 0 0 | 94 | | | Acc. to status reg. |
| | ATFLOPR | A → FLOPR | 1 0 0 1 0 1 0 1 | 95 | | | Acc. to diskette drive reg.<br>Bit 0 = select 1<br>Bit 1 = select 2<br>Bit 2 = select 3<br>Bit 3 = select 4<br>Bit 5 = step pulse<br>Bit 7 = step direct |
| | ATWDATR | A → WDATR | 1 0 0 1 0 1 1 0 | 96 | | | Acc. to write data reg. |
| | ATWCLKR | A → WCLKR | 1 0 0 1 0 1 1 1 | 97 | | | Acc. to write clock reg. |
| | ATSREG | A → SREG | 1 0 0 0 S3 – S0 | 8X | | | Acc. to S-reg. |
| Transfer operation<br>E → S<br>1 byte | TESREG | E → SREG | 1 0 1 0 X X X X | AX | | | Transfer data from E-reg. to S-reg. The addresses are supplied automatically. S-reg;<br>Drive address : 30<br>Rel. trac. Addr : 31<br>Sec. Address : 32<br>Add. H. Reg. : 33<br>Add. L. Reg. : 34<br>Command : 37 |
| A → D<br>1 byte operation | ATCOMB | A → COMBR | 1 0 0 1 0 0 0 0 | 90 | | | Bit 0 = Page 1 (PSEL 1)<br>Bit 1 = Count 7 enable<br>Bit 2 = Page 2 (PSEL 2)<br>Bit 3 = Read data 2 (disable line)<br>Bit 4 = Read data 1 (daten)<br>Bit 5 = Write data 2<br>Bit 6 = Write data 1<br>Bit 7 = Sync R |
| M → D<br>2 byte operation | MTCOMB | M → COMBR | 1 1 0 1 0 0 0 0 | D0 | Mem. data | XX | |
| A → D<br>1 byte operation | ATGENR | A → GENR | 1 0 0 1 1 0 0 0 | 98 | | | Bit 0 = Not used<br>Bit 1 = Inter. Reg.<br>Bit 2 = R1 (RESDI)<br>Bit 3 = Set DR (Data Request)<br>Bit 4 = Not used<br>Bit 5 = Not used<br>Bit 6 = Not used<br>Bit 7 = R2 (RESJC) |
| M → D<br>2 byte operation | MTGENR | M → GENR | 1 1 0 1 1 0 0 0<br>(ENOMB, OUTPUT and 2 BYTE) | D8 | Mem. data | XX | |
| | | | The OUTPUT necessary for GENCL means output (data) from the mp to the various registers on the board.<br>Does not imply a computer to board transfer. | | | | R1 R2<br>0 0 Don't care<br>0 1 Don't care<br>1 0 RESJC<br>1 1 RESDI |