1982-01-18

THE CONTROLLER FOR PARALLEL TRANSFER DISK DRIVES IN THE PICAP II SYSTEM

Anders Mårtensson

INTERNAL REPORT
LiTH-ISY-I-0488
## CONTENTS

1 GENERAL INFORMATION

1.1 Introduction
1.2 General description
1.3 Performance characteristics
  1.3.1 Transfer modes
  1.3.2 Transfer rates
  1.3.3 Data storage capacity
  1.3.4 Connection of more drives
  1.3.5 EDC option
1.4 Physical outline

2 DESCRIPTION

2.1 Main functions
2.2 Main function blocks
  2.2.1 Drive sequencer
  2.2.2 PICAP control and address generator
  2.2.3 Processor bus interface
  2.2.4 Data paths
  2.2.5 FIFO monitoring circuits
  2.2.6 Sector counter

3 USER INTERFACE AND PROGRAMMING EXAMPLES

3.1 General
3.2 Control and status registers
  3.2.1 Registers for the PICAP address generator
  3.2.2 Drive parameter registers
  3.2.3 Command register
  3.2.4 Status register
  3.2.5 Control mode register
3.3 Control sequence at sequential block transfers
3.4 Some program examples
APPENDIX

Available software for the disk drive and the way to use it
1 GENERAL INFORMATION

1.1 Introduction

This document describes the physical and functional characteristics of the disk drive control unit used in the PICAP system. To read the document some knowledge of the PICAP II system is necessary. It is also recommended to get acquainted with the AMPEX PTD-9309, Parallel Transfer Drive.

Useful documents:

PTD-930x, Parallel Transfer Drive Product Description, AMPEX 3308829-01

PICAP II. General system description


1.2 General description

The controller for PTD in the PICAP system has the task to transfer data between one or more drives and the PICAP time shared bus (and thereby the PICAP memory). It is considered as a "processor" in PICAP and controls the bus accesses as well as the drive(s).

The transfers are initiated by the host computer which commands the controller to record or play back a block of data. One block usually means one image field but the size of a block is programmable. When ready, the controller interrupts the host computer. Sequential transfers of such blocks are possible and are maintained as long as the host sets up new parameters for the controller. The average (i.e. including
time for cylinder to cylinder seek) data rate is sufficient to transfer sequences of images in real time (i.e. 25 frames/sec of at least 512x512 images).

1.3 Performance characteristics

1.3.1 Transfer modes

Data can be transferred on eight or nine parallel channels. Thus nine of the R/W-heads of a disk drive can be used simultaneously. Two such groups of nine heads (i.e. disk tracks) are accessible. There is also a possibility to access a special track independently of the other.

1.3.2 Transfer rates

Eight channels maximum (burst) rate: 9.67 Megabytes/sec

Nine channels maximum (burst) rate: 10.88 Megabytes/sec

The average data rate depends on the seek time and on the formatting of the data on the disk, which in turn is programmable.

With two drives run in parallel the maximum data rate will be: 21.7 Megabytes/sec (= 174 Megabits/sec).

1.3.3 Data storage capacity

Each track on a disk surface can hold 20160 bytes of unformatted data. For each sector 51 bytes are used for "overhead" information. With one sector per track the maximum data capacity thus will be: 20109 bytes. 18 such tracks will provide 361962 bytes.
1.3.4 _ _ Connection of more drives

Provisions are made to facilitate the use of a pair of drives enabling access to 18 parallel channels.

There is also a possibility to install more drives in a daisy-chained fashion which can be used alternatingly with the first (pair of) drive(s).

1.3.5 _ _ EDC option

The controller can be supplied with Error Detection and Correction (EDC) circuitry. This will make use of convolutional coding on each channel separately.

1.4 _ _ Physical outline

The controller circuits occupy two boards in the PICAP cabinet. One board (board A) holds the data transfer parts e.g. line drivers, receivers, FIFOs, etc. The second board (board B) contains the control logic.

If two drives are to be connected and run in parallel, one more board must be installed (board C). This board is similar to the data board above (board A).

The drive is connected through four ribbon cables; three of them carry data and clock signals and the fourth transfers control information.
2 DESCRIPTION

2.1 Main functions (See block diagram BS1)

The controller has the capability to transfer data between
nine parallel drive channels and the PICAP time shared bus.
The transfers can be performed as single block transfers
(where one block usually means one image field) or in a block
sequential manner.

A microprogrammed sequencer provides the control and monitoring
functions for the disk drive in a number of modes e.g. seek,
record, playback.

The controller is prepared to handle two disk drives accessed
in parallel. As an option more drives can be connected which
then might be used alternatingly.

The PICAP bus control is implemented as a firm-ware programmed
sequencer. It handles two sets of data related to a pair of
disk drives by means of two independant address generators.

Provisions are made to facilitate later installation of Error
Correction and Detection (EDC) circuits.

The host computer commands the controller by writing into
operation control registers via the PICAP processor bus. As
an acknowledgement of fulfilled tasks the controller interrupts
the host computer.
2.2 Main function blocks

2.2.1 Drive sequencer (Board B, see block diagram BSB1)

Each of the two drive sequencers has the task to supply the disk drive control lines with the appropriate signals to perform an operation. Examples of operations are: Set Head, Seek to cylinder n, Record one block, etc. The sequencer fulfills its task according to a microprogram which is initiated when the host computer writes into a command register.

The sequencer is built with the AM2910 microprogram controller. There is a program memory of 1K words of 24 bits. Either RAM or PROM can be used. The AM2910 condition code input is connected to a test input selector with 16 lines. Tests are made on disk drive status signals and on internal monitoring circuits.

The disk drive control lines are commanded by latch or pulse outputs from the sequencer. Other such outputs are used for the internal logic in the controller e.g. FIFO handling. For a detailed description of the disk drive control interface, see the "Parallel Transfer Drive, Product Description".

A number of registers and buffers hold the parameters that control the disk drive (see KSB10 and 3 USER INTERFACE ....). There are two sets of such registers, the second supplying the next parameters at sequential transfers. This means that before starting such transfers the host must load both sets of registers and then update one at each finished transfer.
2.2.2 PICAP control and address generator (Board B)

In order to transfer one word of data to or from the PICAP time shared bus a "hand shaking" procedure must be started and an address should be set up. The design permits two data sets with independent addresses to be handled virtually simultaneously. Block diagram BSB2 shows the address generator.

Two registers hold the actual addresses computed in the adder. The increment is fetched from either of two registers ("line" or "word" increment). The increments are common to the two sets of addresses. The start addresses are of course separate and there are also two chaining registers containing the next sets of start addresses at sequential transfers.

Four counters (two for each set of addresses) keep track of the number of times each increment register is used. The actual address is thus computed as \( \text{STARTADDRESS} + N \times \text{WORDINCR} + M \times \text{LINEINCR} \). Here \( N \) and \( M \) denotes number of words/line and number of lines respectively. These numbers are loaded into the counters.

The logic that controls the PICAP access is shown simplified on block diagram BSB3. It is mainly designed with state sequencers with PROMs and gets its input signals partly from the FIFO monitoring circuits and partly from the Processor Bus Interface (see 2.2.3) and of course also from the bus itself. Besides the PICAP control, these circuits also supply the FIFOs with necessary signals (FPOINT 0-3 and load signals or unload signals depending on the direction of transfers).

The PICAP accesses are performed in bursts of a length that is set by the FIFO monitoring circuits. In the record mode the FIFO is first filled to a predetermined value. When the drive has started to record, the FIFO is gradually emptied. As the FIFO contents goes below a certain level the PICAP
accesses are started again and the FIFO is filled. This procedure is repeated until the address generator logic signals that a whole block (one image field) has been transferred. A similar procedure takes place in the playback mode.

2.2.3 Processor bus interface (Board B)

This part has the purpose of transferring control data to or from the host computer. The processor bus is connected with the host through a special interface in PICAP which in turn connects to a HSD (High Speed Device) interface in the host computer.

Two main types of transfers are used:

1) DMA-transfers to or from the host.

2) Command transfer. This is a one word transfer directed to a command register in the controller. By writing into this register the host initiates a controller operation.

The parts that handle this communication are shown on block diagram BSB4. Transceivers connect the processor data bus to the internal bus. The control signals for the communication are handled by a microprogram sequencer.

8 address bits are supplied by the host at a transfer start. Four of these bits points to the actual processor and a comparator recognizes the pre-wired address. The remaining bits are used as an internal pointer in the processor. As far as the controller is concerned, there are two types of DMA-transfers. The first is used to access registers in the controller (called direct DMA). The internal address points to the actual register.
When for example the microprogram memory is to be loaded from the host, the 2nd type of transfer called indirect DMA is used. In this case the first word transferred from the host is a start address where three bits are used to point to a certain device in the controller and 12 bits form the start address in this device e.g. the microprogram store. The subsequent transfers are data words and to provide the address the counter is incremented at each such transfer. A special internal address is used to indicate the indirect DMA mode to the sequencer.

Another function block handles the interrupt signalling to the host and also provides the error flag.

2.2.4 Data_paths (Board A)

The Disk Drive provides nine data channels in parallel plus a 10th channel that can be used for data or as an address or "book-keeping" channel (for details, see the PTD Product Description).

The 9 data tracks are formatted in accordance with chapter 5.1b in the PTD description. Unpacking and packing these data streams to fit into the PICAP data format is performed on the data board in the following way: The 32 bits of PICAP data is considered as four 8-bit words (bytes). Thus the PICAP data can be used as a stream of bytes which are assigned to the 9 drive channels in a circular manner one byte at a time. The sequence starts with the first byte being assigned to channel 0, the second to channel 1 and so on. Consequently the 9th byte is assigned to channel 8 and the 10th to channel 0.

There is also a possibility to use 8 channels instead of 9 and a third possibility of using only one channel. In these modes the packing/unpacking procedure above is of course changed accordingly.
The main components for transferring data between the disk drive and the PICAP system are: LINE RECEIVERS, LINE DRIVERS, DESKEW FIFOs, FIFOs SHIFTERS and REGISTERS.

The DESKEW FIFOs serve as buffers to compensate for the time lag which might exist between the data channels (max 128 bits). Thus the input data to these FIFOs might be clocked in at different instants while the clocking out is the same for all 9 channels.

The FIFOs (main FIFOs) provide enough buffering to allow some waiting for access to the PICAP bus. In record mode these FIFOs are loaded by control signals created in the PICAP and FIFO sequencer. They are unloaded (i.e. serially shifted out) by signals derived from the disk drive clock. In playback mode the FIFOs are serially shifted in by the drive clock and unloaded by signals from the PICAP and FIFO sequencer.

2.2.5 _ _ FIFO monitoring circuits (Board B)

To control and monitor the contents of the FIFOs described in 2.2.4, there is a counter that is incremented or decremented in pace with the loading or unloading of the FIFOs. A comparator and a PROM keeps track of the number of data in the FIFOs and signals the PICAP control logic (2.2.2) to stop or start accessing the PICAP bus. These circuits are placed on board B but the counter enable and up/down signals are derived on board A.

Three levels of the contents in the FIFOs are monitored: The lower level indicates that the FIFOs are "almost empty", the middle level is set to a value corresponding to "half full" and the third means "almost full". These levels are stored in PROMs and might be chosen differently for playback and record mode.
2.2.6 Sector_counter (Board B)

A 6-bit counter which is incremented by the sector clock pulses and reset by the index signal informs the controller of the angular position of the rotating disk. The counter is compared with a start sector value and an end sector value. The disk drive sequencer tests the comparator output for a match. These values are set by the host.
3 USER INTERFACE AND PROGRAMMING EXAMPLES

3.1 General

The user handles the PTD controller and PICAP interface by setting up parameters in a number of control registers. These registers are loaded via the processor bus in the PICAP system. This bus is connected to the host computer through an HSD (High Speed Device) interface. There are also status registers in the controller which can be monitored from the host via the processor bus.

When the controller has finished an operation it interrupts the host and if an error has occurred it sets an error flag. To get more precise information about the error type the host must read the error register in the controller.

The controller is designed to perform sequential operations. This means that e.g. transferring of a block of data to the PICAP memory is immediately followed by a new block transfer as soon as the appropriate data is found on the disk. For this purpose there are two sets of those registers that hold the PICAP address information and the disk drive parameters.

At the initiation of a sequential transfer both sets of registers must be loaded i.e. the parameters for the first and the second block transfer are to be supplied. The operation is started when the host writes a command word into the command register (this type of loading is performed with a special one word transfer operation of the HSD interface).

The controller performs the operation and when the first block has been transferred it interrupts the host and the second block transfer immediately starts. If the sequence is to be continued the host must load a new set of parameters each time an interrupt is received. If the host does not load a new set of parameters the controller stops the operation when
the transfer defined by the last set of parameters is finished. The host starts a new sequence by loading two sets of parameters again and sending start command. This initiation must not be performed until the last interrupt of the previous sequential transfer has been received.

Of course there is also the possibility to perform single block transfers. The choice between single and sequential transfers is indicated by a bit in the command register. However, (for hardware reasons), the two sets of parameter registers must be loaded also in the case of single transfers. They should be loaded with the same parameters. Only one interrupt is given at such transfers.
3.2 Control and status registers

3.2.1 Registers for the PICAP address generator

As mentioned earlier the PICAP address generator is capable of supplying two sets of addresses for the use of two disk drives in parallel. Thus there are two sets of start address registers. However, in this description, the case of parallel run is not considered. The address is computed as follows:

\[
\text{ADDR} = \text{STARTADDR} + \text{LINEINCR} \times L + \text{WORDINCR} \times W
\]

The following registers for the address generator must be loaded before a transfer is started.

ADDRESS GENERATOR PARAMETERS:

- \text{INCL} \rightarrow \text{Increment between "lines"} <16:31>
- \text{INCW} \rightarrow \text{"words"} <16:31>
- \text{NLIN} \rightarrow \text{(Number of "lines")-2} <16:31>
- \text{NWORD} \rightarrow \text{(Number of "words")-2} <16:31>

NB: The \text{NLIN} and \text{NWORD} parameters (the "-2" is for hardware reasons).

Address generator parameters must be loaded in sequence and in the above order. Internal address = 1

PICAP START ADDRESS REGISTERS:

- \text{ADDR} \rightarrow \text{Address register} <8:31>
- \text{ADDR} \rightarrow \text{Start address} <12:31>
- \text{R*/W} \rightarrow \text{Read from PICAP=0, write=1} <8>
  \rightarrow \text{Not used} <9:11>

There are two such registers; for the first block transfer and for the next block transfer. Internal address = 6

These two registers should be loaded before a transfer is initiated. At sequential transfers the ADDR register should be updated after each interrupt.
3.2.2 Drive parameter registers

As for the PICAP START ADDRESS REGISTER there is a pair of disk drive parameter registers, one for the first block transfer and one for the next.

DCTRL > Drive parameter register <0:31>

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>NWCYL</td>
<td>New cylinder=1</td>
<td>&lt;0&gt;</td>
</tr>
<tr>
<td>CYL</td>
<td>Cylinder no</td>
<td>&lt;1:10&gt;</td>
</tr>
<tr>
<td>SECS</td>
<td>Start sector-1</td>
<td>&lt;11:16&gt;</td>
</tr>
<tr>
<td>SECL</td>
<td>Last sector</td>
<td>&lt;17:22&gt;</td>
</tr>
<tr>
<td>HDGRP</td>
<td>Head group</td>
<td>&lt;23:25&gt;</td>
</tr>
<tr>
<td>HD</td>
<td>Head no</td>
<td>&lt;26:29&gt;</td>
</tr>
<tr>
<td></td>
<td>Spare</td>
<td>&lt;30:31&gt;</td>
</tr>
</tbody>
</table>

Internal address = 2

The NWCYL bit should be set if the cylinder number differs from previous transfer.

CYL is a number 0 - 814.

SECS is the start sector -1 value.

With the current formatting of the disk the start sector no is dependant on the CYL no.

SECL the last sector in the block transfer. (Not always used.)

HDGRP is the headgroup and HD is the head no in accordance with the PTD Description.

These two registers should be loaded before a transfer is started. At sequential transfers one register should be updated after each interrupt.
3.2.3 _ _ Command register

This register is loaded to start an operation. The HSD command transfer is used to write into this register.

COMD \ Command word \ <0:31>

PTREN1 \ PICAP transfer enable disk1 \ <0>
PTREN2 \ -"- \ -"- \ -"- \ -"-2 \ <1>
STDSKC1\ Start disk control 1 \ <2>
STDSKC2\ -"- \ -"- \ -"- \ 2 \ <3>
\ Spares \ <4:17>
SEQTRAN\ Sequential transfers=1 \ <18>
MAP10 \ Mapping address disk ctrl 1 (MSB) \ <19>
MAP11 \ -"- \ -"- \ -"- \ -"- \ <20>
MAP12 \ -"- \ -"- \ -"- \ -"- \ <21>
MAP13 \ -"- \ -"- \ -"- \ -"- \ <22>
MAP14 \ -"- \ -"- \ -"- \ -"- \ <23>
REC1 \ Record on disk 1 = 1, Playback = 0 \ <24>
REC2 \ -"- \ " \ "- \ -"- \ <25>
MAP20 \ Mapping address disk ctrl 2 \ <26>
MAP21 \ -"- \ -"- \ -"- \ -"- \ <27>
MAP22 \ -"- \ -"- \ -"- \ -"- \ <28>
MAP23 \ -"- \ -"- \ -"- \ -"- \ <29>
MAP24 \ -"- \ -"- \ -"- \ -"- \ <30>
\ Used by the host driver \ <31>

Explanation of command register fields:

PTREN, : when this bit is set the PICAP data transfers are started.

STDSKC \ starts the drive sequencer program at a microprogram address originating from the MAP field.

SEQTRAN: 0 at single block transfers
1 at sequential transfers.
3.2.4 Status register

This register contains information about fault conditions in the controller. It can only be read by the host.

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Fault in drive 1 (FAULTDSK1)</td>
</tr>
<tr>
<td>1</td>
<td>Seek error in drive 1 (SEEK ERROR1)</td>
</tr>
<tr>
<td>2</td>
<td>Temp warning in drive 1 (TEMP WARN1)</td>
</tr>
<tr>
<td>3</td>
<td>Error flag in drive 1 (ERROR 1)</td>
</tr>
<tr>
<td>4</td>
<td>Drive ready flag in drive 1 (DSK READY1)</td>
</tr>
<tr>
<td>5</td>
<td>On cylinder (ON CYL1)</td>
</tr>
<tr>
<td>6</td>
<td>Unit ready flag in drive 1 (UNIT RDY1)</td>
</tr>
<tr>
<td>7</td>
<td>Write protected flag in drive 1 (WRIT PROT1)</td>
</tr>
<tr>
<td>8:31</td>
<td>Not used</td>
</tr>
</tbody>
</table>

Internal address = 5

3.2.5 Control mode register

This register is used when initiating the controller after power off or failure and for test purposes.

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0:3</td>
<td>Not used</td>
</tr>
<tr>
<td>4</td>
<td>Step drive sequencer 1 (STEP1)</td>
</tr>
<tr>
<td>5</td>
<td>Step drive sequencer 2 (STEP2)</td>
</tr>
<tr>
<td>6</td>
<td>Halt drive sequencer 1 (HALT1)</td>
</tr>
<tr>
<td>7</td>
<td>Halt drive sequencer 2 (HALT2)</td>
</tr>
<tr>
<td>8:15</td>
<td>Not used</td>
</tr>
<tr>
<td>16</td>
<td>Reset drive sequencer 1 (RESETD1)</td>
</tr>
<tr>
<td>17</td>
<td>Reset PICAP ctrl logic (RESPIC)</td>
</tr>
<tr>
<td>18:31</td>
<td>Not used</td>
</tr>
</tbody>
</table>

All bits are active high (= 1)

Internal address = 0
The STEP and HALT bits are used to test the microprogram sequencer.

The RESETD1 and RESPIC signals are used when an error has occurred within the controller.

The RESETD1 is also used when the microprogram is loaded to the drive sequencer.

3.3 Control sequence at sequential block transfers

At initiation:

1) Load address generator parameters (4 words)
2) Load startaddress and next address (2 words)
3) Load first and next drive parameters (2 words)
4) Send command word

At interrupt:

1) Load address register (1 word)
2) Load drive parameter register (1 word)

When the last address and drive parameters have been loaded, wait for the next to the last interrupt and the last interrupt.
3.4 Some program examples

The following program performs a single block transfer from the disk to the PICAP memory. One block means one field i.e. 256 lines of 512 pixels. Some comments might be necessary:

The table SECTAB gives the start sector number for different cylinders. The three least significant bits in the cylinder no are used to point in this table.

The MAP no (8) points to a start address in the drive sequencer microprogram.

The PAR array contains the PICAP address generator parameters: Line increment (= 129), Word increment (= 1), Number of lines -2 (256-2), (Number of words/per line)-2 (128-2).

The line increment is chosen to pick out one field from the PICAP memory i.e. every other line is used. Thus on the first line (if PMSTRT=0) the last address will be 127. To get the third line the next address should be 256, which gives the line increment = 256-127 = 129.

The subroutines MIP:WR1, MIP:RE, MIP:CM and MIP:WR2 are assembler routines used to write or read the registers in the controller via the HSD interface.
SUBROUTINE VD: READ1(ICYL, HDGRP, PMSTRT)

READS ONE HALF CYLINDER FROM THE VIDEO DISC
PARAMETERS: ICYL = THE DISC CYLINDER NO (0 - 814)
           HDGRP = HEAD GROUP (0 OR 1).
           PMSTRT = START ADDRESS IN PICTURE MEMORY.

INTEGER PAR(4)
INTEGER PDSK(2)
INTEGER PMADR(2), PMSTRT
INTEGER HDGRP
INTEGER SECPNT, SECTOR, SECTAB(0:7)

DATA SECTAB /0,24,48,8,32,56,16,40/ !STARTSECTORS
DATA PAR /129,1,254,126/ !ADDR.GEN. PAR.
           !FOR FIELD TRANSFERS
DATA ICOM /Z'A0000800'/ !NOT SEQTRAN, MAP=8

PMADR(1) = PMSTRT
PMADR(2) = PMADR(1)

PDSK(1) = Z'80000038' + ISHFT(HDGRP, 6) !NEW CYL, HD=14
SECPNT = IAND(ICYL, Z'7') !SECTORPOINTER
SECTOR = SECTAB(SECPNT) !STARTSECTOR
PDSK(1) = PDSK(1) + ISHFT(ICYL, 21) + ISHFT(SECTOR, 15)
PDSK(2) = PDSK(1)

CALL MIP: WR2(1Z1, PAR, 4, 1Z2, PDSK, 2, 0) !LOAD ADDR.GEN.
           !PARS AND DISK
           !PARS
CALL MIP: WR1(1Z6, PMADR, 2, 0) !LOAD STRT ADDR. REGS.

CALL MIP: CM(ICOM, 0) !START COMMAND
BLOCK DIAGRAMS
NOTE:
Only function blocks for one drive shown

Block diagram BS1. Main function blocks (simplified)
TEST INPUTS

Blockdiagram BSB1. Drive sequencer
Block diagram BSB2. PICAP and FIFO control
The registers LINE INCR., WORD INCR., Num. of lines, Num. of words are connected in series to the internal bus and must be loaded in the given order.

Abbreviations:
LD = load register
OE = 3-state output enable
V = 3-state output
CNT = count

Block diagram BSB3. PICAP address generator
Block diagram BSB4. Processor bus interface
APPENDIX

AVAILABLE SOFTWARE FOR THE DISK DRIVE AND THE WAY TO USE IT

1. Normal run of the drive

1.1 _ Initiation of the controller

_CVD:INIT_ The program _CVD:INIT_ loads the microprogram _MIPMIP_ stored on the system area. It also resets the drive controller and the PICAP access logic. Finally it gives the command "REZERO" to the disk drive.

This program should be used after a power off/power on operation of the drive or the controller (PICAP).

_MIPINIT_ There is also a TSM macro command 'MIPINIT', which could be used with the same effect.

_CVD:MOVE_ 1.2 _ The program _CVD:MOVE_

This program is automatically started at a restart operation of the SEL computer.

The program first performs the same initiation procedure as _CVD:INIT_ and then with the interval of about half an hour it performs random seeks on the disk.

The purpose is to prevent the heads from being positioned at the same cylinder for too long periods.
The program is suspended (for example at initiation with MIPINIT) by the program VD:SUSP.

It is resumed again with the program VD:RSUM.

### 1.3 Routines for record and playback of the disk

The subroutines VD:READ and VD:WRITE reads/writes one cylinder of the disk. The amount of data transferred is 256 kbytes (i.e. one 512x512 image).

**Format for call:**

```plaintext
CALL VD:READ (ICYL, IPART)
CALL VD:WRITE (ICYL, IPART)
```

- **ICYL** = Cylinder no on the disk (0-814)
- **IPART** = Partition no in PICAP memory (0-15)

These routines are used when only one half cylinder (i.e. one group of heads used) of the disk is to be transferred.

**Format:**

```plaintext
CALL VD:READ1 (ICYL, IHDGRP, IPMSTA)
```

- **ICYL** = Cylinder no
- **IHDGRP** = Head group (0 or 1)
- **IPMSTA** = Start address in Picture Memory (0-F000HEX)

### 1.4 Assembler routines for communication with the controller via the HSD interface

A number of such routines have been written by Björn Gudmundsson to facilitate easy communication handling over the PICAP processor bus.
The special routines for the disk drive controller have the prefix 'MIP'.

For details of these programs refer to source code listings.

Brief description:

MIP:WR1 and MIP:WR2 writes in one or two different internal addresses

MIP:CM writes in the command register (interrupt is expected)

MIP:WRCM writes in an internal register plus the command register (interrupt is expected)

MIP:STRT starts sequential transfers

MIP:CONT continues sequential transfers.
2. _Test run of the drive and the controller_

For test purposes of the controller function (and to a certain extent of the drive) there is a test program developed.

For details of this program, reference is made to the maintenance part of the controller description.