



WISSENSCHAFTLICHE
DATEN-
VERARBEITUNG G.M.B.H.

W D V

MAGNETIC TAPE UNITS

Index

Description	page 1 - 3
MGTST	page 4
MAGR	page 5
MAGRE	page 5
MAGW	page 5
MAGRS	page 6
MAGWS	page 6
MAGWE	page 6
MAGSP	page 7
MAGRW	page 7
MAG1 ; MAG2	page 7
MAGSN	page 7
LWCNT	page 8
FMTST	page 9

SOFTWARE DESCRIPTION

The following XIO instructions are used:

WRITE	DC	ADR	This instruction writes a block of words to magnetic tape,
	DC	/FE22	commencing at the ADR address in core. A small gap is left
			at the beginning.
WRTLN	DC	ADR	Used when there is a defective part of the mag. Tape at the
	DC	/FE2A	beginning of the tape. A long gap is left so that these de-
			fective parts are skipped.
READ	DC	ADR	This instruction reads the next block into core storage
	DC	/FE23	commencing at the core address ADR.

The above instructions are followed by:

WCNT	DC	N	Places the value of N in the word count register. N+1 words
	DC	/FE21	are then read from magnetic tape or written to tape into or
			from ADR, ADR-1, ADR-2,.....ADR-N.

The values in the wordcount registers can be loaded into the accumulator with the instruction:

LWCNT	DC	0	The 0,1 and 2 bits are not used. After the transfer of
	DC	/FF28	N+1 words, the contents of the register are 1. If the read
			operation finds only M words, the contents of the register
			are N-M.

BACKSPACE

BSP	DOC	00	Backspaces the tape until the beginning of the preceding
	DC	/FE24	block is reached.

REWIND

RWND DC 0 Rewind the mag. Tape to the beginning.
 DC /FE25

SENSE BITS

Using the following instruction, the bits referring to mag.
Tape units are loaded into the accumulator:

SENSE DC 0 The bits 0, 3, 4, 5, 6, 7, 8, 9 are used and have the fol-
 DC /FF20 lowing meaning:

- 0 - Signifies end of the last read or writes operation.
This bit is reset to zero with the next word instruction.
- 3 - A parity error occurred on the last read or write operation.
Bit 3 is reset with the next read or write operation to zero.
- 4 - Bit 4 is 1 during a read, write or rewind or backspace.
Otherwise bit 4 is zero.
- 5 - The unit is not ready. Press to REMOTE key.
- 6 - The unit is in rewind operation.
- 7 - The end of the magnetic tape has been reached or passed. (EOT)
Must be reset with the WCNT instruction.
- 8 - The FILE PROTECT ring is not in. Read only is possible.
- 9 - The magnetic tape is at the START position. (BOT)

FORTTRAN SUBROUTINES

These programmes are used with the CALL statement and with the
ONE WORD INTEGER specification.
Standard precision and extended precision are differentiated
as follows:

If the floating values $X(1), X(2), \dots, X(N)$ are to be transferred in standard precision then: EQUIVALENCE ($X(1), IX(2)$) must be used and the variables $IX(1), IX(2), \dots, IX(2N)$ replace the variables $X(1), X(2), \dots, X(N)$.

If the floating values above are to be transferred using extended precision, then

EQUIVALENCE ($X(1), IX(3)$) is used
and $IX(1), \dots, IX(3N)$ replace the variables X .

ERRORS

- (1) The computer stops within the accumulator extension /6666: -
either the unit is not ready or a write operation has been attempted with file protect ring out. If the count is not ready - press the REMOTE key. In the case of the ring being inserted and the programme start key being pressed thereafter, the unit is again tested and if all is well the programme proceeds.
- (2) Accumulator /8000. MAGSP (see later) has reached tape end or tape beginning. Continues when the programme start key is depressed. To reset bit 4 press STOP-FORWARD-STOP-REMOTE-STOP-REVERSE-STOP-REMOTE.

All subroutines wait for the current operation to be completed before execution.

FORTTRAN SUBROUTINES (cont.)

MGSTST

Calling Sequence: //XEQ MGSTST

Tests the MAGNETIC TAPE UNIT.

Writes 100 blocks, then backspaces 100 blocks and reads these blocks.

Error messages are written on the console typewriter.

This write/read process is repeated along the magnetic tape until the console entry switch is switched on. (CONSOLE - Switch 0)

The error messages on the typewriter can be suppressed using console entry switch 1.

If no comments are written on the typewriter during non-suppressed operations, the magnetic tape is ready for use.

If the programme waits with accumulator /1111 after an error message, the programme start key is pressed to restart the programme.

MAGR

Calling Sequence: CALL MAGR (IA, N, IE)

READS BLOCKS. The read operation of the next tape block is started using the table IA(1),..... IA(N).

If a parity error occurred during the preceding read or write operation IE =1. Otherwise IE =0.

Core Storage: 54 words.

MAGRE

Calling Sequence: CALL MAGRE (IA, N, IE)

The table IA(1),..... IA(N) of the next available tape block is read. This operation is repeated a maximum of 10 times.

If there is a parity error IE =1. Otherwise IE =0.

Core Storage: 78 words.

MAGW

Calling Sequence: CALL MAGW (IA, N, IE)

Initiates the write operation instead of the read operation as in MAGR.

Parity errors are delt with in the same way.

Core Storage: 64 words.

MAGRS

Calling Sequence: CALL MAGRS (IA, N, IE)

This programme works like MAGR, but waits for READY.

(WAIT FOR READY)

If a parity error is found in the just initialized function, then IE receives the value 1 as divergent from MAGR.

With error free operation, IE receives the value 0.

MAGWS

Calling Sequence: CALL MAGWS (IA, N, IE)

This programme works like MAGW, but waits for READY.

(WAIT FOR READY)

IE is equal to MAGRS.

MAGWE

Calling Sequence: CALL MAGWE (IA, IB, N, IE)

The table IA is written to tape and afterwards the table IB is read from magnetic tape.

If the corresponding values are not equal, the operation is repeated 10 times, after which the error blocks are overwritten by a long gap.

If the table values do not correspond, IE =1. Otherwise IE =0.

Core Storage: 119 words.

FORTTRAN SUBROUTINES (cont.)

MAGSP

Calling Sequence: CALL MAGSP (N)

The magnetic tape is spaced N blocks forward (N positive) or N blocks backwards (N negative).

Core Storage: 70 words.

MAGRW

Calling Sequence: CALL MAGRW

The magnetic tape initiates rewind operation.

Core Storage: 36 words.

MAG1;MAG2

Calling Sequence : CALL MAG1 CALL MAG2

Starts the first or second magnetic tape unit. All other operations are executed by the operation of the unit itself.

Core Storage: 36 words.

MAGSN

Calling Sequence: CALL MAGSN (M)

The status bits of the magnetic tape unit are stored in location M and stay in the accumulator after return to the calling programme. Using the statement 1 IF (MAGSN(M)) 2,1,1 the programme stays at statement number 1 until bit 0=1 and the last read or write operation is completed.

With the instruction IF(MAGSN(M)*8)2,1,1 the programme goes to statement 2 if a parity error occurs. Otherwise to 1.

Core Storage: 14 words.

LWCNT

Calling Sequence: CALL LWCNT(M)

The contents of the word count register are stored in location M and remain in the accumulator when return is made to the calling programme. The value of M corresponds to the number of words which failed to be read after a read operation is completed. After a write operation is completed, the programme executes the instruction IF (LWCNT(M))1,1,2 which causes a branch to statement 2 if the tape block is less than it should be: otherwise a transfer to 1 occurs.

Core Storage: 16 words.

FMST

Calling Sequence: // XEQ FMST

With this programme 'File-marks' (EOF) may be written, as well as read in both tape - directions.

This programme may be used for TEST, if file-marks may be written or read, as well as for the manual singular writing and reading of the same beyond the usual Software - service.

After the XEQ-call, the programme goes to PAUSE 1010 and asks DATSW. 0, if programme should run or go to EXIT; then DATSW. 15 which tape unit should be served.

____ Immediately after XEQ-call: DATSW. 0 ON for programme running.
DATSW.15 ON for example MAG1

Afterwards the console - switches (DATSW.) will determine the further operation:

Write file-mark (WEOF)	DATSW. 1 ON push PROGR.START
Search file-mark forward (REOF)	DATSW. 2 ON push twice PROGR.START (because of the pauses 1111 and 2222)
Search file-mark backw. (REOB)	DATSW. 2 OUT twice PROGR.START (see REOF)

Should the programme be finished, so DATSW. 0 should be put to OUT, then the programme goes after the last reading- or writing-operations to EXIT.