

MACROASSEMBLER MAS2

Range : MITRA 125 (mode 2)

Systems : MMT2

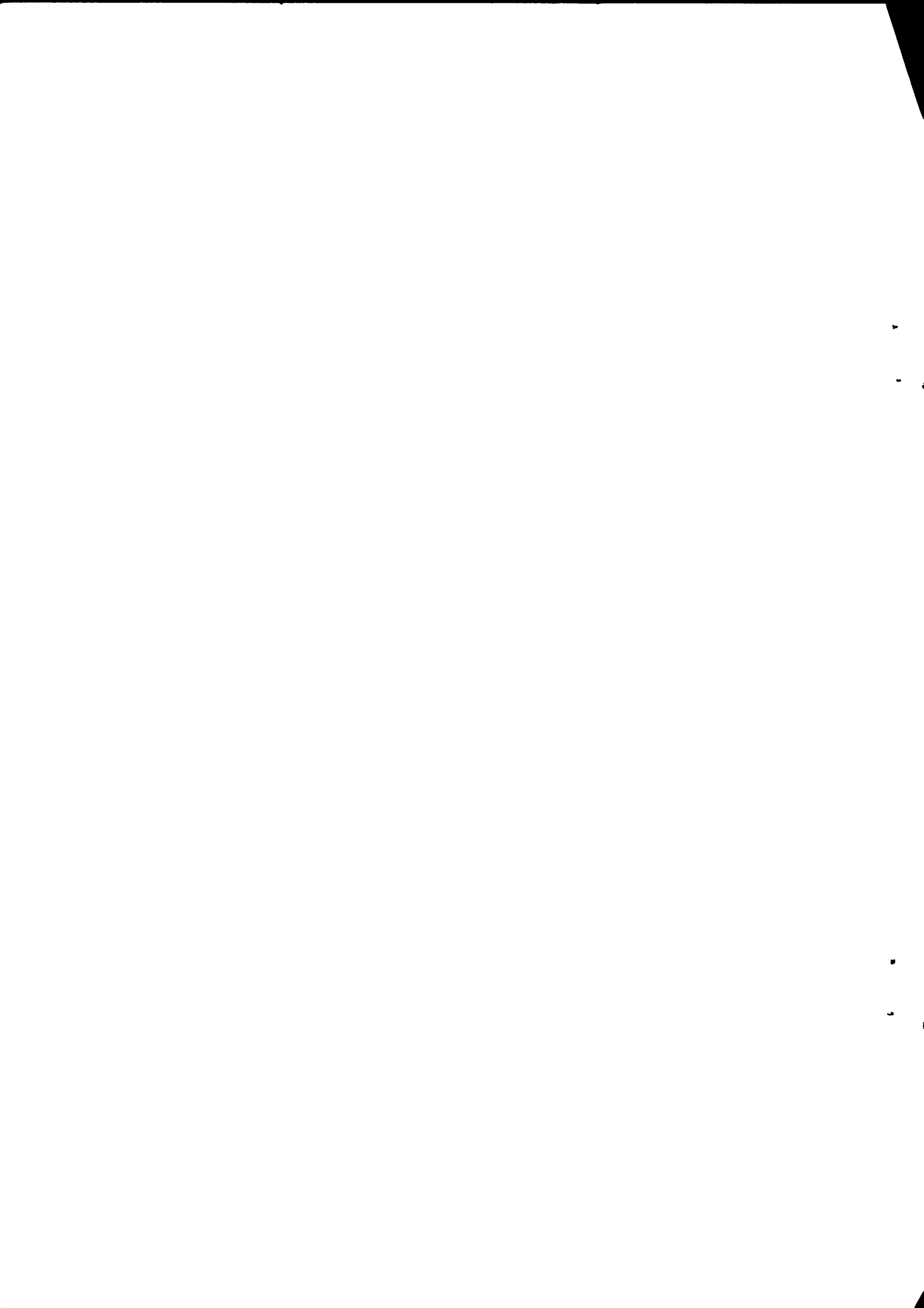
Purpose : This operator sheet furnishes elements required to
Use Macroassembler MAS2

Remarks : The information contained in this document refers to a
minimum index 2 product

Edition date : December 1977

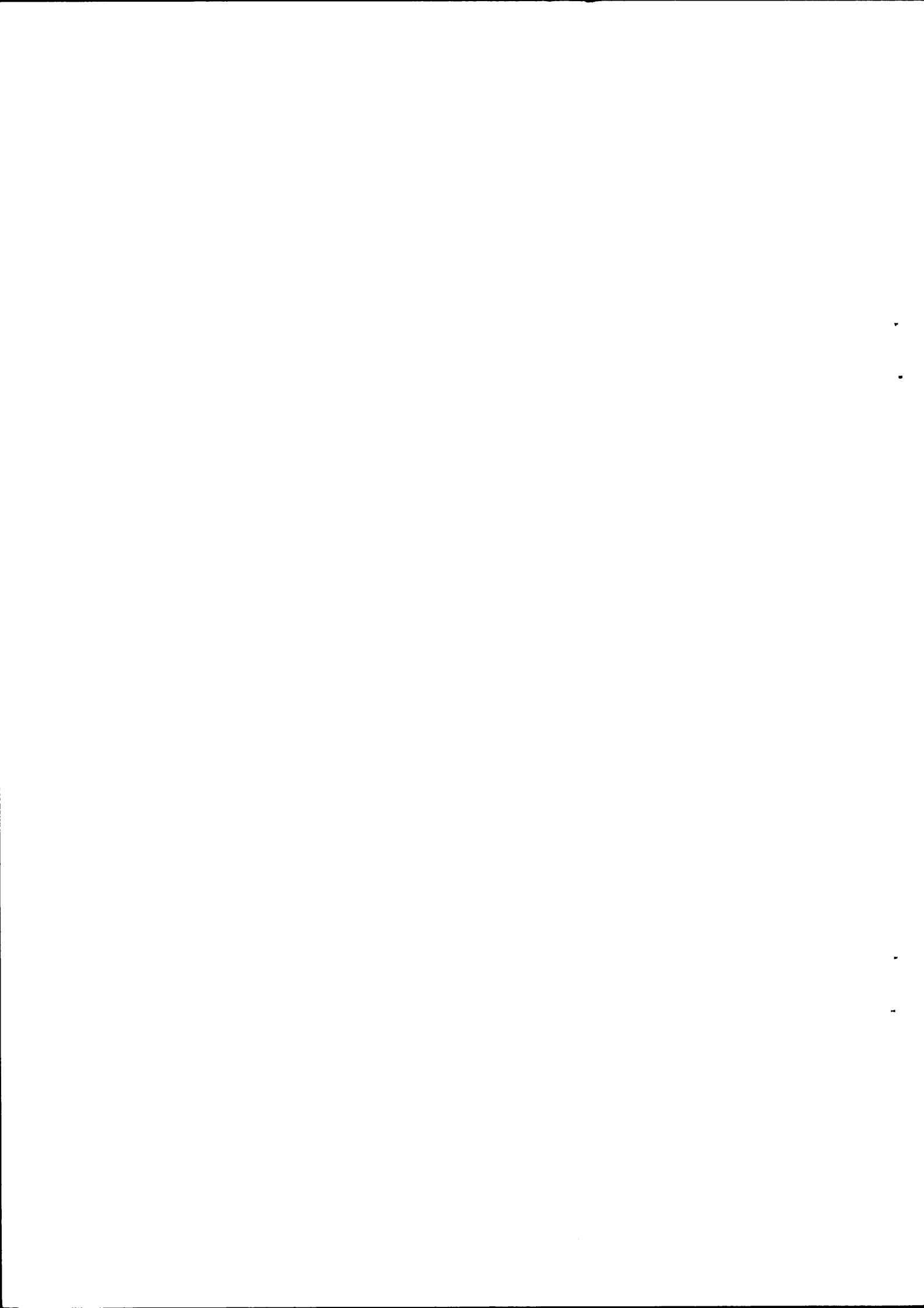
Document Reference

4769/O/EN



- SUMMARY -

1 LEVEL 2 MACROASSEMBLER	1.1
1.1 IDENTIFICATION	1.1
1.2 REQUIRED HARDWARE/SOFTWARE	1.1
1.3 FUNCTION	1.1
1.4 OPERATOR INTERFACE	1.1
1.4.1 Example	1.2
1.4.2 Initiation command	1.2
1.4.2.1 Format	1.2
1.4.2.2 Subsection OUT	1.2
1.4.2.3 Subsection IN	1.2
1.4.2.4 Subsection OPT	1.3
1.4.2.5 MNP Subsection	1.4
1.4.2.6 PAR Subsection	1.4
1.4.2.7 Subsection SDEC	1.4
1.4.2.8 Remarks	1.4
1.4.3 Pre-compiled SDEC creation	1.5
1.4.3.1 SDEC input flow structure	1.5
1.4.3.2 Command format	1.5
1.4.3.3 IN Subsection	1.5
1.4.3.4 OPT Subsection	1.5
1.4.3.5 MNP Subsection	1.5
1.4.3.6 PAR Subsection	1.6
1.4.3.7 Remarks	1.6
1.4.4 Label table	1.6
1.4.5 Subsection table	1.6
1.4.6 Error processing	1.7
1.4.6.1 Severity level	1.7
1.4.6.2 Errors	1.7
1.4.6.3 Standard processor processing	1.9
1.5 MEDIUM	1.10
1.5.1 Description	1.10



1 LEVEL 2 MACROASSEMBLER

1.1 IDENTIFICATION

Documentation: Operator sheet: 5473/02/EN
User's manual : 5419 /U1/EN

1.2 REQUIRED HARDWARE/SOFTWARE

Hardware : Memory size : 10.6 K words
: Peripherals : dialogue machine, disk

Software : MMT2 Monitor

Logical name of standard files used :

BT	Output RB (Relocatable Binary)
LS	Input source file
EDIT	Output assembly list and error numbers
FT1	Working file
MV	Virtual memory file

Note : these logical names are used in the initiation command (see related paragraph).

1.3 FUNCTION

The macroassembler permits obtaining a MITRA 125 Relocatable Binary from a source text written in MITRA 125 (or MITRA 105 or MITRA 15) assembly language.

1.4 OPERATOR INTERFACE

Below is an example of control cards required for assembly followed by details on initiation command format.

1.4.1 Example

In this example, it is assumed that MAS2 is running under BATCH2 sub-system control. Implicit assignments of logical names of files MV, FT1, EDIT, LS have been performed by BATCH2 sub-system.

RB file assignment :

AS/BT, < RB name > , DMOO,, DM, SQ, S, BN, W, STD, 104.

Or

AS/BT,, MTOO,, MT, SQ, S, BN, W, STD, 104.

Call of assembler :

CAL/MAS2/OPT; VMS=<n.vms > ?

/MNP; < RB name > /PAR; P,R,E,S,T,I,N,G,C.

1.4.2 Initiation command

1.4.2.1 Format :

CAL/MAS2/?

/OUT;LFN=< lfn > ?

/IN; [SDEC] ,LFN= < lfn > [- < lfn > ...] , TYPE= < t1 > ?

/OPT;VMS= < n.vms > , SWAP= < t2 > , EDIT= < lfn > , ERR= < lfn > ?

/MNP;NAME= < string > , VER= < number > , LEV= < number > ?

/PAR;P,R,E,S,T,I,N,G,C ?

[/SDEC;NAME= < string > , LFN= < lfn >] .

Only the MNP subsection is obligatory.

1.4.2.2 Subsection OUT

LFN= < lfn >>indicates the logical name of the file where the RB (BT implicitly) will be generated. This standard organization file is sequential.

1.4.2.3 Subsection IN

SDEC indicates presence of a file containing the pre-compiled SDEC.

LFN= < lfn > indicates the logical name of the file or the files where the source text will be read (LS implicitly)

TYPE= < t1 > indicates the 3 possible types of source text MITRA 15, MITRA 105, or MITRA 125 (implicitly MITRA 125)

Remarks : The source text can be read on 5 different files. These files must be sequential.

1.4.2.4 Subsection OPT

VMS= < n.vms > specifies the number of virtual memory pages (200 implicitly; 16 minimum; 1200 maximum).

SWAP= < t2 > indicates whether swapping on disk or not.

: Y indicates swapping on disk.

: N indicates transfer in memory (SSDS) (:Y implicitly).

EDIT= < lfn > indicates the logical name of the file where listing and table will be edited. (EDIT implicitly).

ERR= < lfn > indicates the logical name of the file where errors will be edited (EDIT implicitly)

1.4.2.5 MNP Subsection

NAME= < string > allows declaring the name given to the RB (6 EBCDIC characters) this parameter is obligatory.

VER= < number > specifies the version number (0 implicitly)

LEV= < number > specifies the level (0 implicitly)

1.4.2.6 PAR subsection (print options). If the subsection is not present, all the options are inhibited.

P	Paging pseudo-instructions print-out
R	"Remote" expansion print-out
E	User macroinstruction expansion print-out
S	Structure expansion print-out
T	Label and subsections tables print-out
I	Internal macroinstruction expansion print-out
N	NOLIST pseudo-instruction cancellation
G	Print-out of source lines skipped at assembly by pseudo-instructions: GOTO, IF, JMP.
C	Print-out of code generated from macroinstructions if expansion is not requested.

1.4.2.7 Subsection SDEC

NAME = <string> indicates pre-compiled SDEC name, this parameter is obligatory
LFN = <lfn> indicates logical name of the file where pre-compiled SDEC will
be read (SDEC implicitly).

Subsection SDEC and SDEC keyword of subsection IN must always be present
simultaneously.

1.4.2.8 Remarks

Logical file names have to be assigned to peripherals as well as
FTI for the working file and MV for virtual memory management. The batch
subsystem performs these assignments implicitly.

Any source file must be terminated by an end of file mark.

Assembly file is output on 126 characters.

At end of assembly, "XXXX END MAS2 LEV. <x>" message is printed out on the
dialogue machine (<x> being the severity level) and XXXX the group and task
numbers.

1.4.3 Pre-compiled SDEC creation

1.4.3.1 SDEC input flow structure

The input flow must begin with a SDEC pseudo-instruction and be termi-
nated by the two pseudo-instructions FIN and END.

It can be divided into a maximum of 5 files.

1.4.3.2 Command format

CAL/MAS2?

/OUT;SDEC ?

/IN; LFN= < lfn > [- < lfn > ...] ?

/OPT; VMS=<n.vms>, SWAP=:Y, EDIT=<lfn>, ERR=<lfn>?

/MNP; NAME=<string> ?

/PAR; P, R, S, T, I, N, G, C.

1.4.3.3 IN Subsection

LFN=<lfn> indicates the logical name of the file where the SDEC source
text will be read (LS implicitly).

Remark: SDEC source text can be read on 5 different files.

1.4.3.4 OPT Subsection

VMS=<n.vms> indicates number of virtual memory pages (200 implicitly, 16 minimum, 1200 maximum).

SWAP=:Y indicates swapping on disk.

EDIT=<lfm> indicates the logical name of the file where the listing and tables will be edited (EDIT implicitly).

ERR=<lfm> indicates the logical name of the file where errors will be edited (EDIT implicitly).

1.4.3.5 MNP Subsection

NAME=<string> permits declaration of the name given to SDEC (6 EBCDIC characters), this parameter is obligatory.

1.4.3.6 PAR Subsection

P	Paging pseudo-instructions print-out
R	"Remote" expansion print-out
S	Structure expansion print-out
T	Label table print-out
I	Macroinstruction expansion print-out
N	NOLIST pseudo-instruction cancellation
G	Print-out of source lines skipped at assembly by pseudo-instructions GOTO, IF, JMP.
C	Print-out of code generated from macroinstructions if expansion is not requested.

1.4.3.7 Remarks

The < lfm > FTI must be assigned to the working file.

The < lfm > MV must be assigned to the file which will contain the pre-compiled SDEC.

"TAILLE EN SECTEURS DU FICHLER SDEC : XXXX" message (SIZE of SDEC file expressed in sectors : XXXX) is printed out at the end of precompiled SDEC creation.

1.4.4 Label table

A table containing the list of labels is printed out after the end of each subsection. It is in alphabetical order on 6 columns.

The format is :

ETIQ Z LLLL

ETIQ : Label name

Z : Label type

C : CDS

L : LDS

P : LPS

S : Structure

A : Absolute

E : External

LLLL : Value

1.4.5 Subsection table

A subsection table is edited after pseudo-instruction END. It is in an alphabetical order and contains the actual subsection names and their type : C (CDS), L (LDS) or P (LPS).

1.4.6 Error processing

1.4.6.1 Severity level

0 : No error

1 : Probable error corrected by the assembler

2 : Obvious error

Errors 31 through 60 : no generation

Errors 61 through 90 : generation of 1, 2 or 4
or 4 "zero" bytes

3 : Abort current pass.

1.4.6.2 Errors

Level 1 errors

01 symbol too long. It is truncated

02 subsection redefinition. Label is ignored.

03 syntax error in a label. It is ignored

04 symbol defined twice. It is ignored.

05 syntactic element not essential. It is ignored.

06 location counter odd. It is incremented by one.

07 argument DEF or REF incorrect. It is ignored.

08 subsection name missing or incorrect. It is ignored.

09 argument field not the same as DUM. It is ignored.

10 LDS name associated with an LPS, incorrect. Last LDS is implicitly assumed.

11 start address in FIN incorrect. Address 0 is implicitly assumed.

12 section name in END incorrect. First section is implicitly assumed.

13 keyword reserved or keyword defined twice. Keyword is ignored.

14 FINST, or FIND, or FINR, or END pseudo-instruction found before label searched for by a GOTO, IF, JMP. Search is aborted.

15 FIN or END pseudo-instruction found before end of loop FIND. Loop is closed.

16 END pseudo-instruction found during macroinstruction expansion. Expansion is interrupted.

17 END found during "remote" generation. Generation is interrupted.

18 structure length error.

19 Continuation line expected during a macroinstruction call. Call is ignored.

20 FIN found before end of structure FINST. Structure is closed.

21 END pseudo-instruction found during RMT definition. Generation is interrupted.

22 expanded line too long. It is truncated.

23 keyword not initialized. It is initialized by an empty string.

24 error generated by pseudo-instruction ERROR,1.

Level 2 errors

Effect: abort line or argument without binary generation.

31 syntax error.

32 error in an expression.

33 label missing.

34 label definition error.

36 close an unopened base. (BASE pseudo-instruction missing).

37 error in an argument.

38 reserved name.

39 continuation line expected during a macro-definition
Abort macro-definition.

40 incorrect keyword transmitted to macro-definition

41 macro-definition of line impossible.

42 macroinstruction defined twice. The first definition is ignored

43 structure declaration incorrect. It is aborted.

45 field incompatible with the support item.

46 loop nesting level greater than 6.

<u>Effect.</u>	abort line or argument with generation.
61	syntax error.
62	error in an expression
63	GEN pattern incorrect.
64	command field configuration not compatible with that of argument field.
65	incorrect command or command prohibited in the current subsection.
66	error in argument field.
67	field overflow.
68	argument with extended addressing not supported by a word.
69	argument with extended addressing based by odd Z.
70	error in an instruction.
71	incorrect keyword or too many keywords when a macro-instruction is called. This call is ignored.
72	Keyword unknown when a macro-instruction is called. This call is ignored.
73	sectioning pseudo-instruction occurring during an expansion not present at initial definition.
74	error in an item.
75	error generated by pseudo-instruction ERROR,2.
77	argument of a HERE pseudo-instruction unknown.

Level 3 errors

<u>Effect :</u>	Abort current pass.
90	error generated by pseudo-instruction, ERROR,3.
91	location counter greater than $2^{16} - 1$
92	END missing
93	insufficient number of virtual memory pages.
94	END found in a subsection.
95	insufficient memory space for an assembly.
96	SDEC name incorrect, or SDEC parameter found both in IN subsection and OUT subsection.
97	SDEC file structure incorrect (in pre-compiled SDEC utilization,
98	SDEC file structure incorrect (in pre-compiled SDEC creation).
99	SWAP=:N parameter used in pre-compiled SDEC creation.

1.4.6.3 Standard processor processing

- Initiation incorrect.

"ABNORMAL START" message is printed out on "EDIT".

Severity level is 6.

- Error in initiation command analysis.

"ANAC ERROR n" message is printed out on "EDIT", n being the report given by ANAC.

Severity level is 5.

- Error not specifically processed by processor.

" * (aaaa) ERROR XXXX llll" message is printed out on "EDIT", with:

aaaa error code

XXXX function corresponding to error

llll lfn in which error was located.

Severity level is 4.

- End of processor message (output on dialogue machine).

"ggtt END XXXXXX LEV. N"

ggtt group and task number

XXXXXX name of processor

n severity level

1.5 MEDIUM

1.5.1 Description

RMI of MAS2.

It is made up of a single file terminated by an end of file mark.



LINKAGE EDITOR LED2

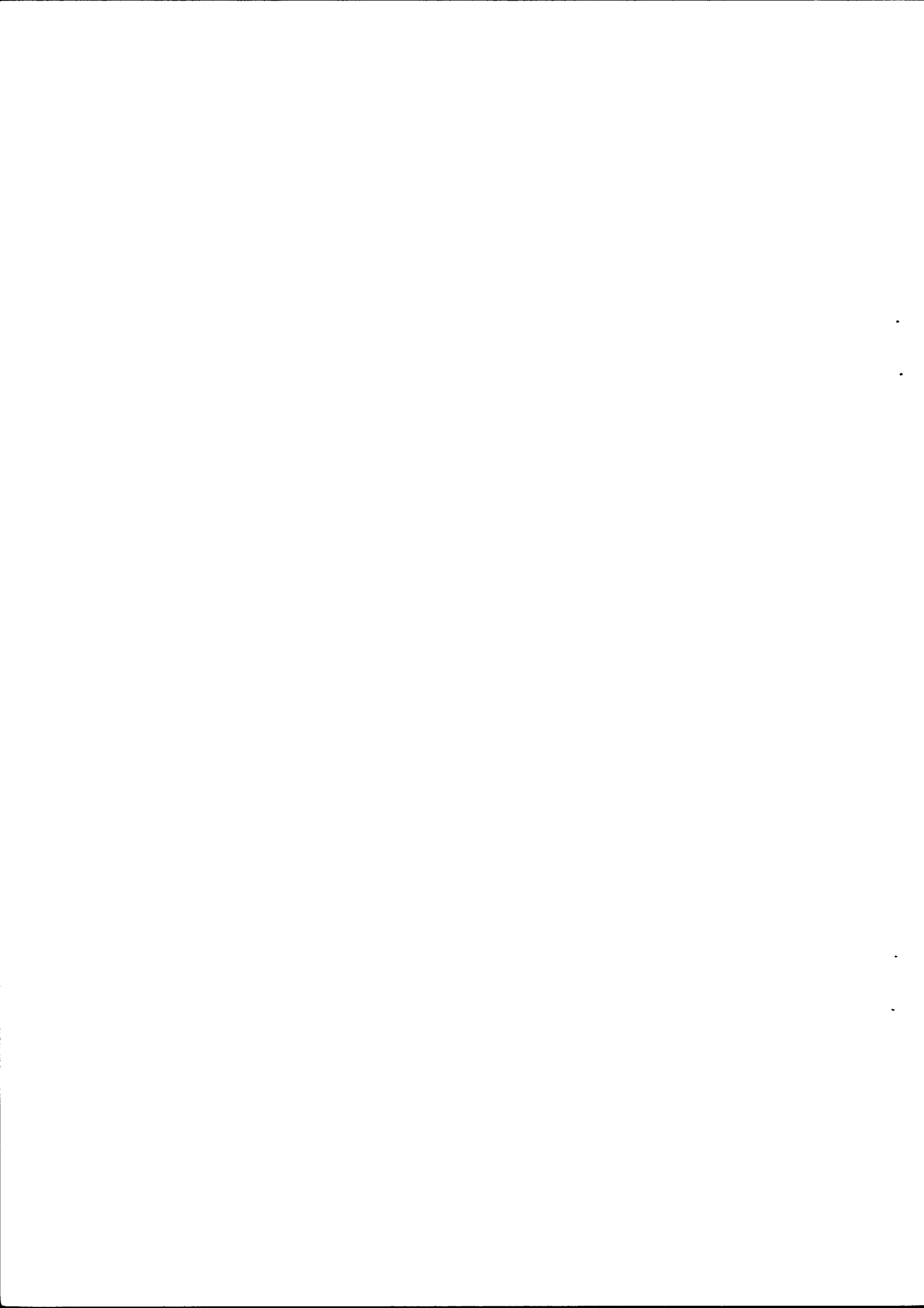
Range : MITRA 125 (Mode 2)

Sytstems : MMT2

Purpose : This operator sheet provides information enabling operation of the LED2 linkage editor.

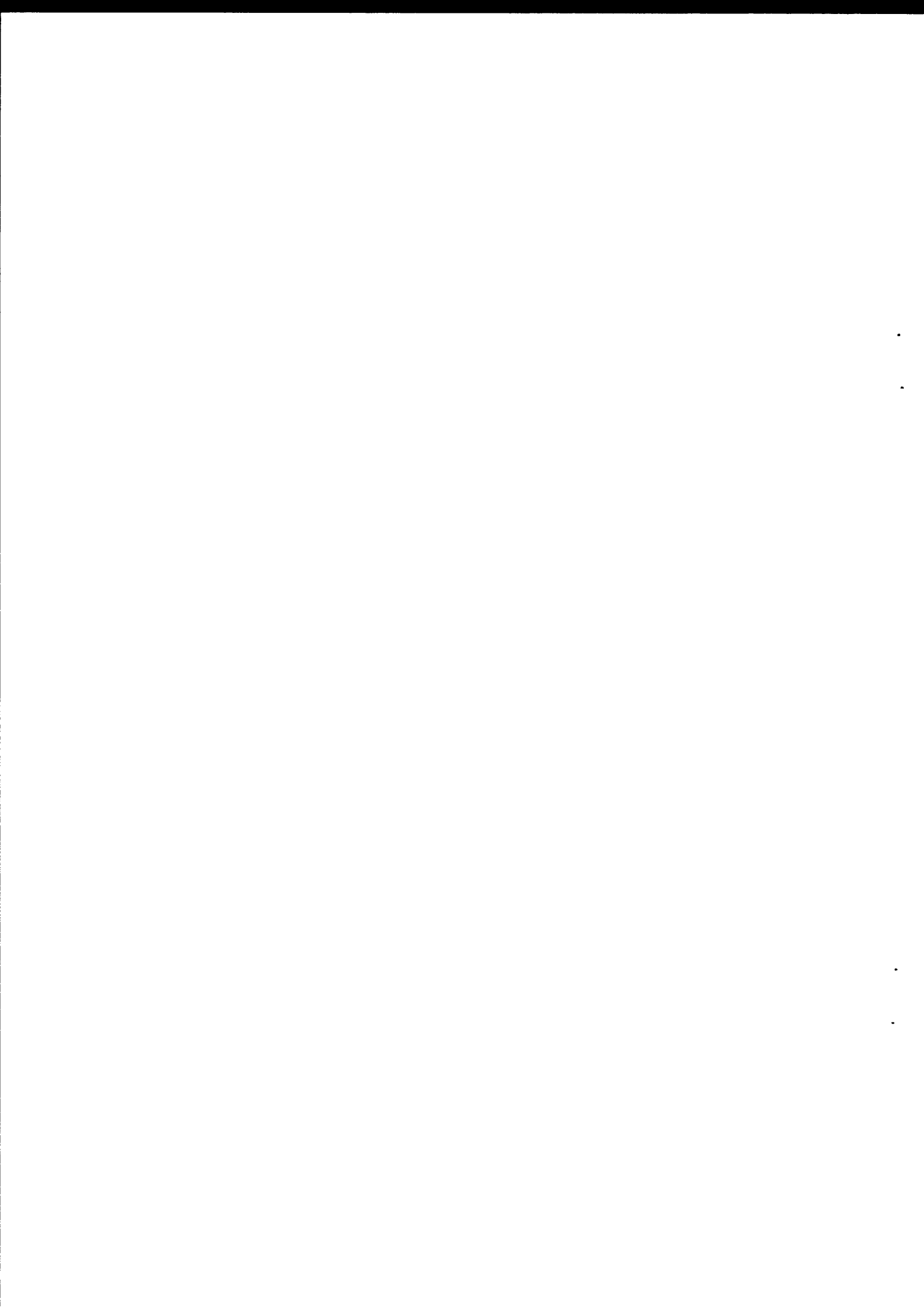
Remarks : The information contained in this manual is valid for products having a minimum index of 2.

Edition date : December, 1977



SUMMARY

1 MITRA 125 LINKAGE EDITOR	1.1
1.1 Identification	1.1
1.2 Hardware/software requirements	1.1
1.3 Function	1.1
1.4 Operator interface	1.2
1.4.1 Initialisation command	1.2
1.4.1.1 Format	1.2
1.4.1.2 Subsection OUT	1.2
1.4.1.3 Subsection IN	1.2
1.4.1.4 Subsection OPT	1.2
1.4.1.5 Subsection MNP	1.3
1.4.1.6 Subsection SDP	1.3
1.4.2 Errors	1.3
1.4.3 Standard processing	1.7
1.5 Medium	1.8
1.5.1 Description	1.8



1.4 Operator interface

1.4.1 Initiation command

1.4.1.1 Format

```
CAL / LED2/?  
/OUT; LFN = <lfn>, S | L?  
/IN; LFN= <lfn> ?  
/OPT; COM=<lfn>, EDIT= <lfn>, ERR= <lfn>, VMS=<number>, ?  
SWAP= : Y | :N, FC ?  
/MNP; NAME= <name of RMI>, VER= <number>, LEV= <number> ,<type> ?  
/SDP; [SP|SV], PV, PR, FLRHT= <file right>, ?  
SYRHT= <system right>, BDSZ= <binding length>, ?  
WKSZ= <working area length>, X, ?  
PRTB= <PRTB length>, TPTE= <TPTE length>, ?  
TSTE= <TSTE length>, USL= <stack length> [ / ].  
Only subsection MNP and NAME parameter is mandatory.
```

1.4.1.2 Subsection OUT

<lfn> indicates the logic name of the RMI file (IMT by default).
This is a standard binary file.
S or L indicates whether this file is sequential or partitioned
(sequential implicitly).

1.4.1.3 Subsection IN

<lfn> indicates the logical name of the RB file (BT implicitly). This
is a standard binary file, and always sequential.

1.4.1.4 Subsection OPT

COM indicates the logical name of the file where subsequent commands
shall be read (LS implicitly). (It is significant only if FC is present).

EDIT indicates the logical name of the file where lists will be edited
(EDIT implicitly).

ERR indicates the logical name of the file in which error messages will
be edited (EDIT implicitly).

VMS specifies the number of virtual memory pages (200 implicitly; 500 max.).



1. MITRA 125 LINKAGE EDITOR

1.1 Identification

Catalogue number: RMI of LED2: 57875
Documentation : Operator sheet 5551 02/EN
User's manual 5427 U2/EN

1.2 Hardware/software requirements

Hardware : Control memory MC2
Memory size 8 Kbytes
Peripherals: dialogue machine, disc

Software : MMT2 monitor

Labels of standard files used:

IMT	RMI output
BT	RB input
LS	Command file
EDIT	Message output
EDIT	Error message output
FTI	Working file
MV	Virtual memory file

Note: These names are used in the initiation command
(see related paragraph).

1.3 Function

The linkage editor is a processor designed to satisfy undefined elements (forward references, external references, calls to user libraries) of relocatable binary (RB) formatted object modules obtained from the assembler and/or the various compilers, and to convert these modules globally into an RMI (relocatable memory image) formatted binary program which can be accepted by the group generator.

1.4 Operator interface

1.4.1 Initiation command

1.4.1.1 Format

```
CAL / LED2/?  
/OUT; LFN = <lfn>, S | L?  
/IN; LFN= <lfn> ?  
/OPT; COM=<lfn>, EDIT= <lfn>, ERR= <lfn>, VMS=<number>, ?  
SWAP= : Y | :N, FC ?  
/MNP; NAME= <name of RMI>, VER= <number>, LEV=<number> ,<type> ?  
/SDP; [SP|SV], PV, PR, FLRHT= <file right>, ?  
SYRHT= <system right>, BDSZ= <binding length>, ?  
WKSZ= <working area length>, X, ?  
PRTB= <PRTB length>, TPTE= <TPTE length>, ?  
TSTE= <TSTE length>, USL= <stack length> [ / ].  
Only subsection MNP and NAME parameter is mandatory.
```

1.4.1.2 Subsection OUT

<lfn> indicates the logic name of the RMI file (IMT by default).
This is a standard binary file.
S or L indicates whether this file is sequential or partitioned
(sequential implicitly).

1.4.1.3 Subsection IN

<lfn> indicates the logical name of the RB file (BT implicitly). This
is a standard binary file, and always sequential.

1.4.1.4 Subsection OPT

COM indicates the logical name of the file where subsequent commands
shall be read (LS implicitly). (It is significant only if FC is present).

EDIT indicates the logical name of the file where lists will be edited
(EDIT implicitly).

ERR indicates the logical name of the file in which error messages will
be edited (EDIT implicitly).

VMS specifies the number of virtual memory pages (200 implicitly; 500 max.).

SWAP indicates whether there is swapping on disc or not
 : Y indicates swapping
 : N indicates transfer in memory (in SSDS)
 (: Y implicitly).

FC indicates the presence of a COM command file.

1.4.1.5 Subsection MNP

NAME gives the name of the RMI (mandatory parameter)
 VER gives the version number (0 implicitly)
 LEV specifies level (0 implicitly)

The RMI type can take on one of the following values: PG, RT, DA, BD, SP
 (PG implicitly) (PG = Program, RT = re-entrant program, DA = Data,
 BD = binding, SP = shared subroutine).

1.4.1.6 Subsection SDP

SP	Shared subroutine mode
SP, PV	Privileged shared subroutine mode
SV	Supervisor mode
PV	Privileged program mode
PR	Memory protection
FLRHT	File right
SYRHT	System right
BDSZ	Binding area length (non-implemented in V2)
WKSZ	Working area length
PRTB	Binding PRT length (non-implemented in V2)
TPTE	TPT extension length
TSTE	TST extension length
USL	User stack length

1.4.2 Errors

Error Number	Meaning	And	Action
1	Command error: command name unknown		Immediate abort
2	ICT overflow		Immediate abort
3	Command: subsection name unknown in a command		immediate abort
4	Command: incorrect utilization of positional subsections		Immediate abort

- 17
- 5 Command: non-successive occurrences of a repetitive subsection
Immediate abort
 - 7 Command: mandatory subsection missing
Immediate abort
 - 8 Command: presence of exclusive factors in the command
Immediate abort
 - 9 Command: incorrect utilization of positional parameters
Immediate abort
 - 10 Command: mandatory parameter missing from a subsection present
Immediate abort
 - 11 Command: separator incorrect
Immediate abort
 - 12 Command: key word unknown
Immediate abort
 - 13 Command: value incorrect
Immediate abort
 - 14 Command analysis error
 - 16 Linkage edition impossible (RB severity level excessive)

 - 29 Name of subsection in + BR command unknown
Immediate abort
 - 30 Branch number incorrect
Immediate abort
 - 31 Number of mother branch or location incorrect
Immediate abort
 - 32 Subsection name incorrect
Immediate abort
 - 33 Name appears twice
Immediate abort
 - 34 Incompatibility between the mode and overlay structure
Immediate abort

 - 65 Table overflow
Immediate abort

 - 100 RB severity level too high
Immediate abort

 - 101 Initiation section multi-defined
Last definition accepted
Module name is read-out
 - 102 Initiation section not defined
Initiation on first section
 - 103 Initiation section not recognized
Initiation on first section

- 104 Initiation on section not generated in RMI
 - Initiation on first section
- 105 Initiation on data subsection
 - Initiation on first section
- 106 CDS not at beginning of RB
 - Immediate abort
 - Module name read-out
- 107 Subsection defined twice
 - Name read-out
 - Abort at end of first pass
- 108 LPS whose LDS has not been previously defined
 - Previous LDS is associated or CDS if no LDS;
 - otherwise: immediate abort
- 109 LPS whose associated LDS is actually a program section
 - Name read-out
 - Abort at end of first pass
- 110 Call in DA type editing
 - Abort at end of first pass
- 111 CLS in type SP editing
 - BRK 0 generated instead of CLS
- 112 CLQ in RT type editing
 - BRK 0 generated instead of CLQ
- 113 External multi-defined
 - First definition is retained
 - Abort at end of first pass
- 114 External used in REF and already declared in REFX
 - Abort at end of first pass
- 115 External used in DEF, already declared in DEFX
 - Second declaration ignored
- 116 External used in DEFX, and already declared otherwise
 - Abort at end of first pass
- 117 External used in DEFX, already declared in DEF
 - Second declaration ignored
- 118 Length/G greater than 32K
 - Immediate abort
- 119 Name of subsection EXT/XREF already defined
- 120 Subsection not in library
- 121 POOL defined twice in the same subsection
- 122 Fill-in of uncreated POOL
- 123 Length of a direct access POOL greater than 256 bytes
- 124 Declaration of COMMON prohibited without previous positioning

- 125 Location of FORTRAN COMMON POOL greater than 256 bytes
- 126 Initialization of COMMON prohibited without previous declaration
- 127 Initialization of COMMONs located at WA/SDS or blank COMMON prohibited
- 128 Overlap between COMMON initializations
- 129 Declared COMMON length less than initialized length
- 130 Extension of length of already declared COMMON
- 131 Overflow of FORTRAN COMMON POOL
- 132 Expression referencing a non-existent COMMON
- 200 Editing without a section in a type other than DA
 - Immediate abort
- 201 CLS in SP type
 - BRK 0 generated
- 202 CLQ in RT type
 - BRK 0 generated
- 204 CLS concerning a data subsection
 - BRK 0 generated
- 205 CLS of a section whose number is greater than 255
 - BRK 0 generated
- 206 Expression concerning non-existent CDS
 - Absolute 0 value generated on the defined field
- 207 Non-homogeneous expression concerning several subsections
 - Absolute 0 value generated on the defined field
- 208 Field overflow
 - Absolute 0 value generated on the defined field
- 209 Branch expression incorrect - non computable
 - BRK 0 generated
- 210 Branch to a displacement greater than 255
 - BRK 0 generated
- 211 Non-homogeneous operand expression
 - BRK 0 generated
- 212 Operand expression does not concern the data subsection associated with current LPS
 - BRK 0 generated
- 213 Expression of mandatory word field different from the word
 - Absolute zero value generated on defined field
- 214 Odd expression and even expression mandatory
 - Absolute zero value generated on defined field
- 215 Expression with generalized addressing and odd operand
 - BRK 0 generated

- 216 DATAB expression not multiple of 16
Zero word generated
- 217 Forward reference not defined
Absolute zero value generated on defined field
- 218 External reference not satisfied
Absolute zero value generated on defined field
- 219 Expression referring to more than ten subsections
Absolute zero value generated on defined field
- 220 Operand field overflow
BRK 0 generated
- 221 Expression referring to non-existing POOL
- 222 Expression referring to non-existing COMMON
- 223 GENOVM: REFX already defined
BRK 0 generated
- 224 GENCLS not satisfied
0 word generated

1.4.3 Standard processing

- initiation incorrect

"ABNORMAL START" message is printed out on "EDIT".

The severity level is 6.

- Initiation command analysis error

The "ANAC ERROR n" message is printed out on "EDIT", where n is the report given by ANAC.

The severity level is 5.

- Error not specifically treated by processor

The message "* (aaaa) ERROR XXXX 1111" is printed out on "EDIT", where

aaaa error code

XXXX Function corresponding to error

1111 Erroneous information

The severity level is 4.

- End of processor message (read-out on dialogue machine)

ggtt END XXXXXX LEV. N"

ggtt group and task number

XXXXXX processor name

n severity level

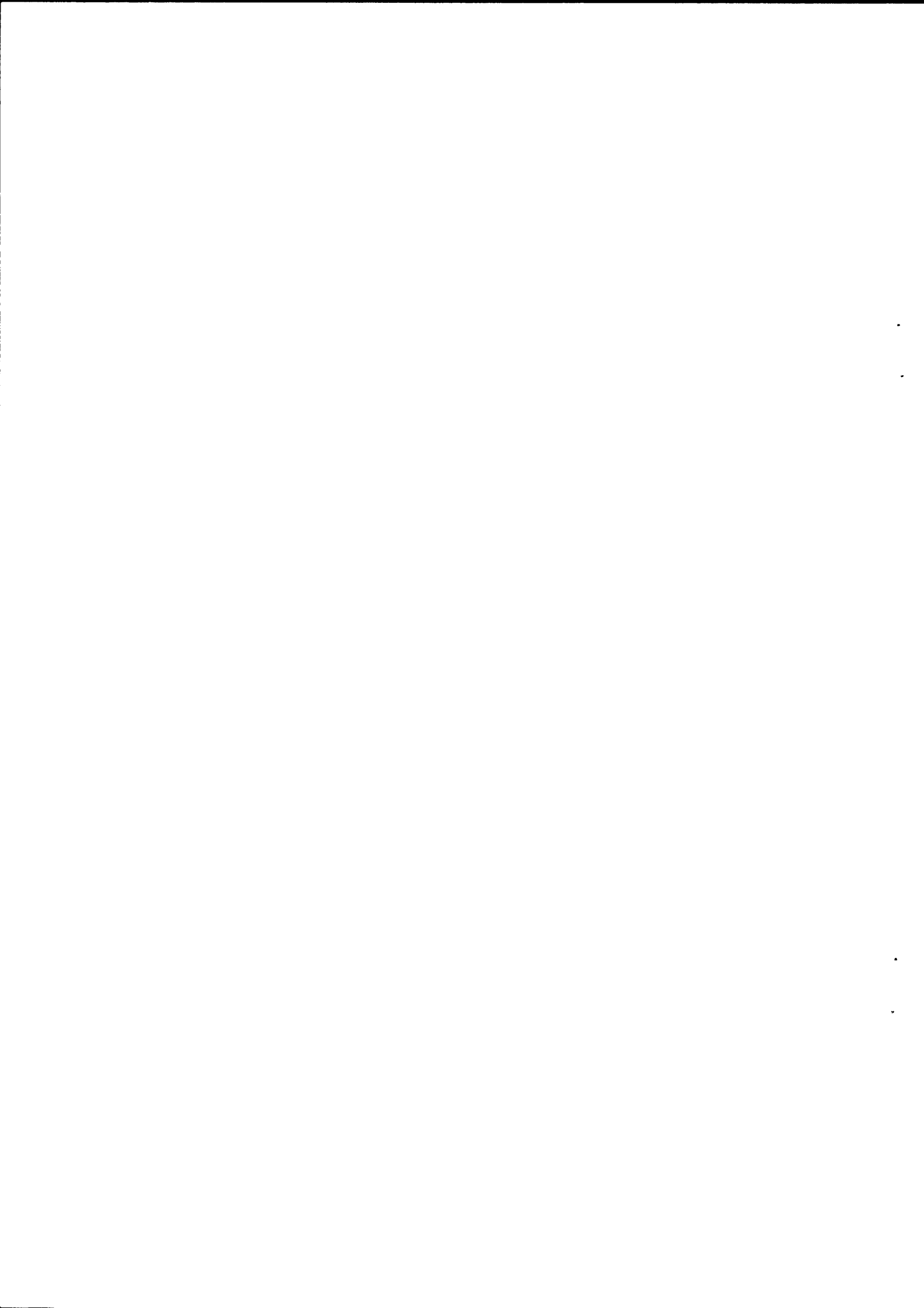
1.5 Medium

1.5.1 Description

RMI of LED2

This consists of a single file terminating with a "end of file" mark.







GROUP GENERATOR GRG2

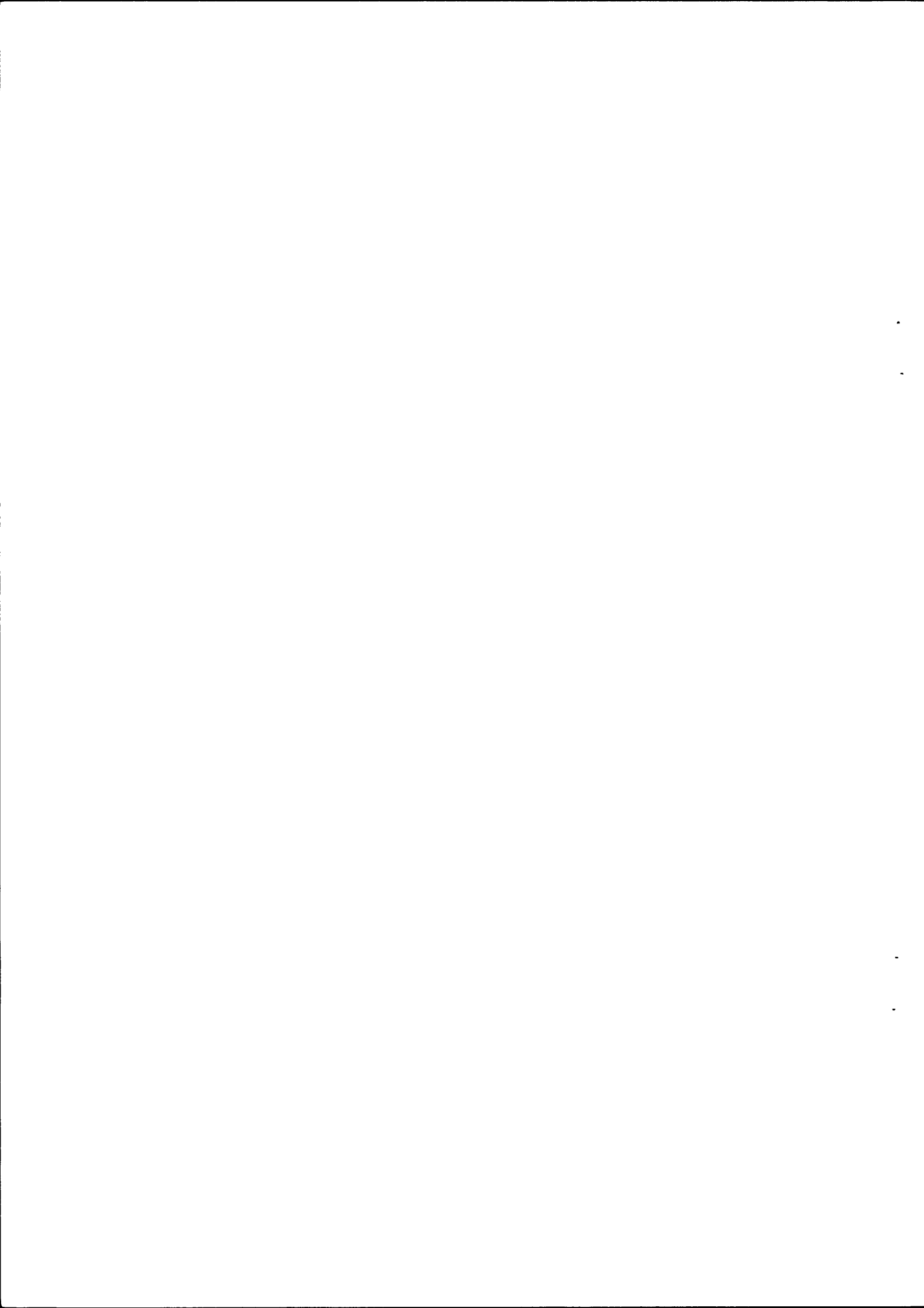
Range : MITRA 125 (mode 2)

Systems : MMT2

Purpose : This operator sheet provides the elements required for processing the Group Generator GRG2.

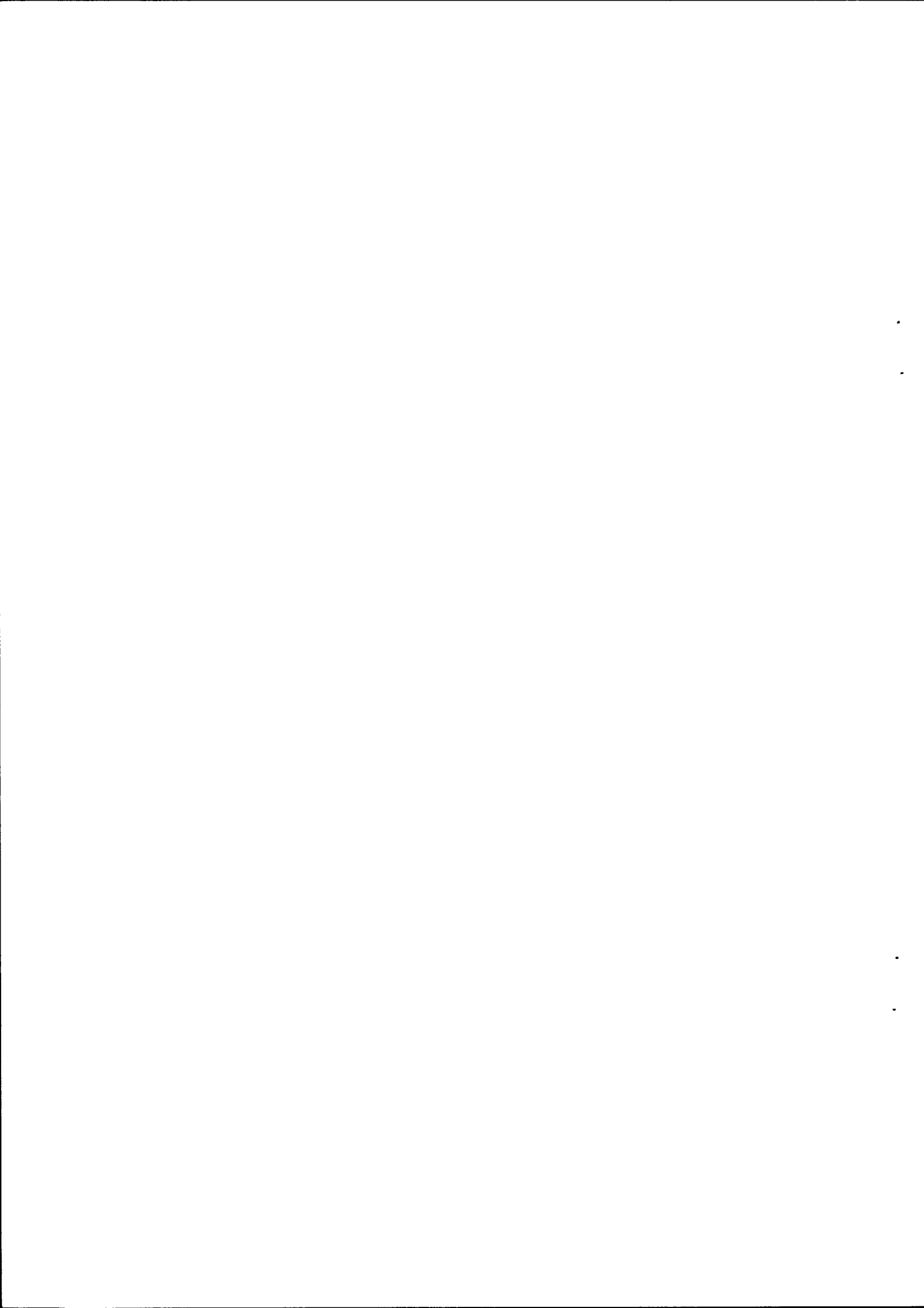
Remarks : The information contained in this manual is valid for products having a minimum index of 2.

Edition date : December 1977



+ SUMMARY +

1 GROUP GENERATOR	1.1
1.1 IDENTIFICATION	1.1
1.2 REQUIREMENTS	1.1
1.3 FUNCTIONS	1.1
1.4 OPERATOR INTERFACE	1.2
1.4.1 Start command	1.2
1.4.1.1 Format	1.2
1.4.1.2 FUNC Subsection	1.2
1.4.1.3 OUT Subsection	1.2
1.4.1.4 OPT Subsection	1.3
1.4.1.5 MNP Subsection	1.3
1.4.1.6 SDP Subsection	1.3
1.4.1.7 RST Subsection	1.5
1.4.2 Commands defining the environment	1.5
1.4.2.1 SYS file definition	1.5
1.4.2.2 SPS file definition	1.5
1.4.2.3 SDS File definition	1.5
1.4.2.4 Definition of the first specific character of the monitor sections	1.6
1.4.2.5 Definition of the initiation RMI name (not implemented)	1.6
1.4.3 Commands defining RMIs at input	1.6
1.4.4 End of COM file command	1.7
1.4.5 Generation examples	1.7
1.4.5.1 Creation of the SYS file and generation of a resident system EMI	1.7
1.4.5.2 Creation of the SDS file and generation of an RMI to be loaded under Z	1.8
1.4.5.3 Creation of the SPS file and generation of a RMI to be loaded under Q	1.8
1.4.5.4 Generation of a user RMI or a system RMI	1.9
1.4.5.5 Generation of a user EMI and extension of the SPS file	1.10
1.4.5.6 Error messages	1.10
1.4.5.7 Standard processing	1.12



1. GROUP GENERATOR

1.1. IDENTIFICATION

DOCUMENTATION : Operator sheet 4657 O2/EN
 User's manual : 4761 U1/EN

1.2. REQUIREMENTS

HARDWARE Memory size : 9.2 K words + tables
 Required peripherals : Dialogue machine,
 Disk

SOFTWARE Monitor MMT2
 Logical names of standard files used:

FTI	Working file
MV	Virtual memory file
IMI	RMI file
IME	EMI file
LS	Command file
EDIT	Message and/or Error Print-out file.

Note: These logical names are used in the start command (see this paragraph).

1.3. FUNCTIONS

The group generator is a processor designed to :

- . Describe a given environment (creation of SYS, SDS, SPS files).
- . Satisfy undefined references (section calls, references external to the RMI).
- . Generate an Executable Memory Image (EMI) directly loadable and executable.

1.4. OPERATOR INTERFACE

1/1/68

1.4.1 Start command

1.4.1.1. Format

CAL/GRG2/?

/FUNC; MOR | MOV | SPS | SDS | PRG | XPRG ?

/OUT; LFN= < lfn > , L?

/OPT; COM= < lfn > , EDIT= < lfn > , ERR= < lfn > ,?

VMS= < nbpgs > , SWAP= :Y | :N , FC ?

/MNP; NAME= < name of EMI > , VER= < version number > ,?

LEV= < level > ?

/SDP; SP | SV, PV, PR, FLRHT= < file right > ,?

SYRHT= < system right > ., BDSZ= < binding size > ,?

WKSZ= < working area size > ,?

PRTB= < binding area size > ,?

TPTE= < TPT extension > , TSTE= < TST extension > ,?

USL= < stack size > ?

/RST; SWE= B | S, PWF= B | S, OPT= B [/] .

Only the FUNC subsection is mandatory.

1.4.1.2. FUNC Subsection

It is mandatory. One and only one of the 6 following functionalities is possible:

- . MOR : Creation of an SYS file and, optionally, generation of a monitor root EMI
- . MOV : Generation of a monitor overlay branch EMI
- . SPS : Creation of an SPS file and generation of a subroutine EMI
- . SDS : Creation of an SDS file and optionally, generation of given EMI.
- . PRG : Generation of a user or system EMI
- . XPRG : Generation of a user EMI and extension of the SPS file

1.4.1.3. OUT Subsection

LFN indicates the logical name of the partitioned file where the EMI will be generated (IME implicitly).

L indicates that the file is partitioned (L implicitly)

1.4.1.4. OPT Subsection

COM indicates the logical name of the file where subsequent commands will be read (LS implicitly).

EDIT indicates the logical name of the file where messages will be printed out (EDIT implicitly).

ERR indicates the logical name of the file where errors will be printed out (EDIT implicitly).

VMS specifies the number of virtual memory pages (200 implicitly).

SWAP indicates the swapping of the group generator's work area onto disk if it is equal to :Y or into memory in segment SDS if it is equal to :N (:Y implicitly).

FC indicates that a COM command file is present (FC implicitly).

1.4.1.5. MNP Subsection

NAME indicates the EMI name which is also the partition name where the EMI will be stored. If omitted, the EMI is not generated.

VER indicates the version number (0 implicitly).

LEV indicates the level number (0 implicitly).

1.4.1.6. SDP Subsection

Parameters SP, SV and PV indicate the EMI execution mode. Possible combinations are :

SV

SP

SP,PV

-

PV

PR indicates that access to protected memory locations is authorized.

FLRHT indicates the program access right with respect to files.

. If this parameter is present, its value may not be greater than the value specified at Linkage Editing.

. If this parameter is missing and if it was not specified at Linkage Editing, its implicit value is zero (minimum right).

SYRHT indicates the program access right with respect to the system.

. If this parameter is present, its value may not be greater than the value specified at Linkage Editing.

. If this parameter is missing and was not specified at Linkage Editing, its implicit value is zero (minimum right).

15

BDSZ indicates the length in bytes of the "binding" area.

- . If this parameter is present, its value must be greater than or equal to the value specified at Linkage Editing.
- . If this parameter is missing, and if it was not specified at Linkage Editing, its implicit value is zero.

(Binding is not implemented in the MMT2-V2 monitor).

WKSZ indicates the length of the work area in bytes actually requested.

WKSZ,X indicates the minimum length of the work area in bytes.

- . If this parameter is present, its value must be greater than or equal to the value specified at Linkage Editing.
- . If this parameter is missing, and if its value was not specified at Linkage Editing, its implicit value is zero.

PRTB indicates the SRD number to be reserved in the PRT of the "binding" area.

- . If this parameter is present, its value must be greater than or equal to the value specified at Linkage Editing.
- . If this parameter is missing, and if it was not specified at Linkage Editing, its implicit value is zero.

TPTE indicates the length in bytes of the TPT extension.

- . If this parameter is present, its value must be greater than or equal to the value specified at Linkage Editing.
- . If this parameter is missing, and if it was not specified at Linkage Editing, its implicit value is zero.

TSTE indicates the length in bytes of the TST extension.

- . If this parameter is present, its value must be greater than or equal to the value specified at Linkage Editing.
- . If this parameter is missing, and if it was not specified at Linkage Editing, its implicit value is zero.

USL indicates the length in bytes of the user stack.

- . If this parameter is present, its value must be greater than or equal to its value specified at Linkage Editing.
- . If this parameter is missing, and if it was not specified at Linkage Editing, its implicit value is 32.

Parameters FLRHT, SYRHT, BDSZ, WKSZ, PRTB, TPTE, TSTE and USL have meaning only when a user EMI is generated.

1.4.1.7. RST Subsection

SWE= B | S indicates that if a software error occurs, task is restarted from section 1 (option B) or sequentially (option S). Otherwise, task is restarted according to task group conditions.

PWF= B | S indicates that if a power failure occurs, task is restarted from section 1 (option B) or sequentially (option S). Otherwise, task is restarted according to task group conditions (not implemented in the V2 monitor).

OPT= B indicates that if an operator trap occurs, task is restarted from section 1 (option B). Otherwise, task is sequentially restarted.

1.4.2. Commands defining the environment

These commands are read on the COM file. Each command can only appear once.

1.4.2.1. SYS file definition

+SYS / PFN= < pfn name > , PVN= < pvn name > [/] .

This command defines < pfn name > as being the physical name of the SYS file and < pvn name > as being the physical name of the volume to which the SYS file belongs.

1.4.2.2. SPS file definition

+SPS / PFN= < pfn name > , PVN= < pvn name > [/] .

This command defines < pfn name > as being the physical name of the SPS file and < pvn name > as being the physical name of the volume to which the SPS file belongs.

1.4.2.3. SDS file definition

+SDS / PFN= < pfn name > PVN= < pvn name > [/] .

This command defines < pfn name > as being the physical name of the SDS file and < pvn name > as being the physical name of the volume to which the SDS file belongs.

1.4.2.4. Definition of the first specific character of the monitor sections.

`+XCAR / CAR= C1 [-C2-C3-C4-C5] [/] .`

This command defines the first alphabetical character of the names of the monitor sections that cannot be called in user mode (characters I, J and X being implicit).

A maximum of five alphabetical characters can be transmitted in the command.

This command may only be used for the MOR functionality.

1.4.2.5. Definition of the initiation RMI name (not implemented)

`+STRT / NAME= <RMI name> [/] .`

This command defines the name of the RMI to be initiated when the input stream comprises several RMIs.

1.4.3. Commands defining RMIs at input

These commands are read on the COM file.

`+IN/LFN= < lfn > , S | L, NAME= < RMI name > , ADR= < adr > [/] .`

This command defines the input RMIs supplied to the group generator.

There must be as many + IN commands as input RMIs.

This command must appear after commands +SYS, +SPS, +SDS, +CAR, +STRT

LFN= < lfn > indicates the logical name of the file where the RMI will be read. Its implicit value is RMI.

S | L indicates whether the file is sequential (parameter S) or partitioned (parameter L). Its implicit value is S.

NAME= < RMI name > indicates the name of the RMI. This parameter is mandatory.

ADR= < adr > indicates the displacement of the first RMI relative to the beginning of the SDS segment for the SDS functionality. This address can also be defined during RMI loading in SDS. If not, the implicit value is zero.

1.4.4. End of COM file command

%EOD

This command indicates the end of the COM file commands.

1.4.5. Generation examples

All these examples are executed under BATCH 2 which performs the implicit assignments to MV, FT1, EDIT, LS, etc....

1.4.5.1. Creation of the SYS file and generation of a resident system EMI

```
%JOB/O,IMER.
%BLBA/& 200
%FI / BT, DMOO/CMFL, SQ, 500, TYPE=BN, STD, 1, 104, 1, BT.
%AS / BT, BT, DMOO,, DM.
%CAL/MAS2/ ?
%/MNP ; NAME = BTSYS.
%FI / IMT, DMOO / CMFL, SQ, 500, TYPE=BN, STD, 1, 104, 1, IMT.
%AS / IMT, IMT, DMOO,, DM.
%CAL/LED2/ ?
%/MNP ; NAME = IMTSYS, PG ?
%/SDP ; SV, PR.

%FI / IME, DMOO / CMFL, PA, 500, TYPE=BN, FIX, 1, 256, 2, IME.
%FI / FSYS, DMOO / CMFL, SQ, 20, TYPE=BN, FIX, 1, 16, 0.
%AS / OUT, IME, DMOO,, DM.
%CAL/LIB2/NEWL/OUT;OUT,L/OPT;NAT=IME.
%AS/IME,IME, DMOO,, DM.
%CAL/GRG2/ ?
%/FUNC ; MOR ?
%/MNP ; NAME = IMESYS ?
%/SDP ; SV, PR.
+SYS / PFN = FSYS, PVN = DMOO.
+IN / IMT, S, NAME = IMTSYS.
%EOD
```

1.4.5.2. Creation of the SDS file and generation of an RMI to be loaded under Z

```

ZJOB/O/,CSDS.
ZBLBA/& 200.
ZFI/BTSDS, DMOO / CMFL, SQ, 500, TYPE=BN, STD, 1, 104, 1, BT.
ZAS/BT, BTSDS, DMOO,,DM.
ZCAL/MAS2/ ?
Z/MNP ; NAME = BTSDS.

ZFI / IMTSDS, DMOO / CMFL, SQ, 500, TYPE=BN, STD, 1, 104, 1, IMT.
ZAS / BT, BTSDS, DMOO,, DM.
ZAS / IMT, IMTSDS, DMOO,, DM.
ZCAL/LED2/ ?
Z/MNP ; NAME = IMTSDS, DA.

ZFI / IMESDS, DMOO / CMFL, PA, 500, TYPE=BN, FIX, 1, 256, 2, IME.
ZFI / FSDS, DMOO / CMFL, SQ, 20, TYPE=BN, FIX, 1, 16, 0.
ZAS / OUT, IMESDS, DMOO, DM.
ZCAL/LIB2/NEWL/OUT;OUT,L/OPT;NAT=IME.
ZAS/IME,IMESDS,DMOO,,DM.
ZCAL/GRG2/ ?
Z/FUNC ; SDS ?
Z/MNP ; NAME = IMESDS.
+SDS / PFN = FSDS, PVN = DMOO.
+IN / IMT, S, NAME = IMTSDS.
ZEOD.

```

1.4.5.3. Creation of the SPS file and generation of a RMI to be loaded under Q

```

ZJOB/O, IMEQ.
ZFI /..BTSPS, DMOO / CMFL, SQ, 500, TYPE=BN, STD, 1, 104, 1, BT.
ZAS / BT, BTSPS, DMOO,, DM.
ZCAL / MAS2 / ?
Z/MNP ; NAME=BTSPS.
ZFI / IMTSPS, DMOO / CMFL, SQ, 500, TYPE=BN, STD, 1, 104, 1, IMT.
ZAS / IMT, IMTSPS, DMOO,, DM.
ZCAL / LED2 / ?
Z/MNP ; NAME = IMTSPS, SP ?

```

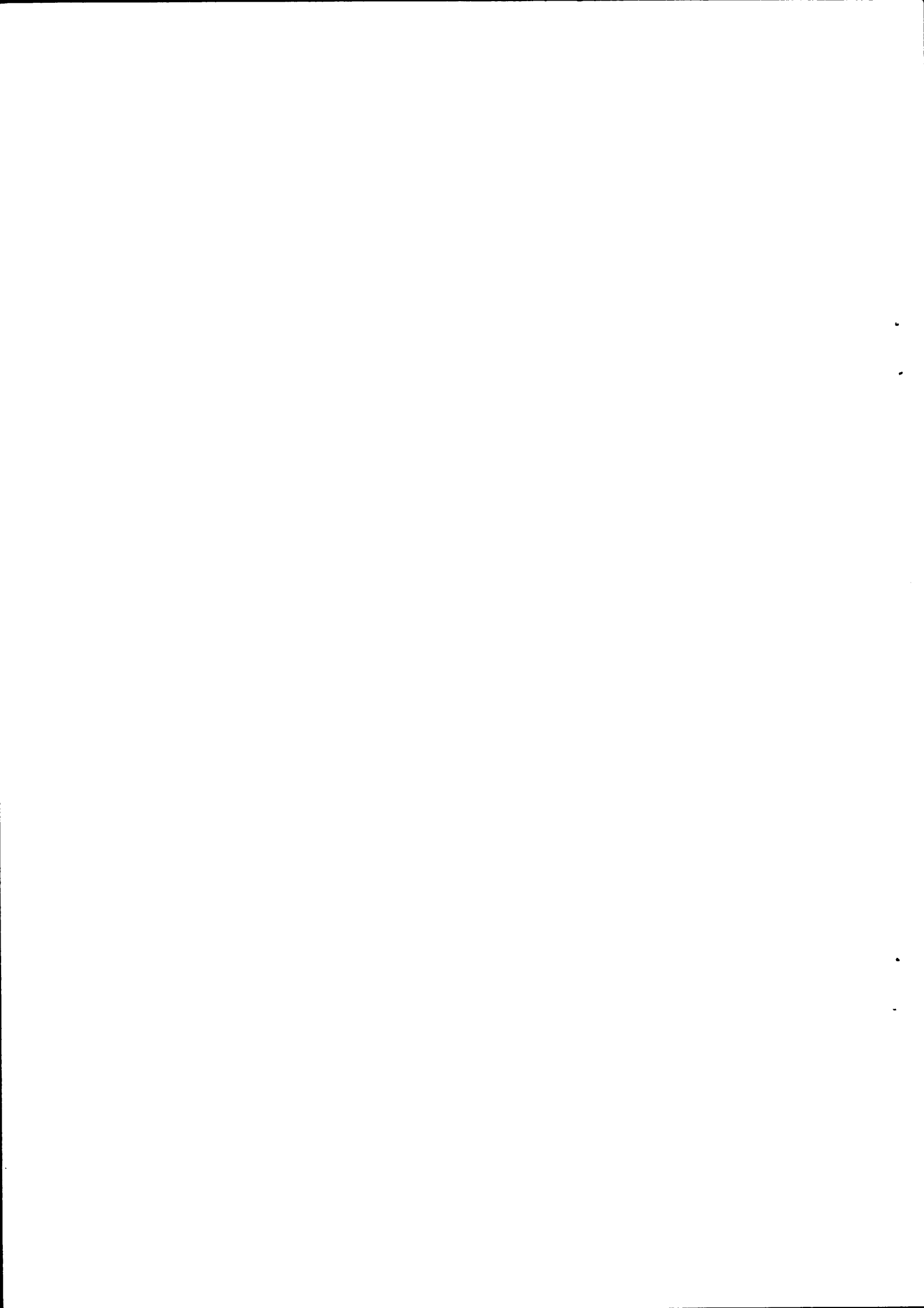
% / SDP ; SP.
% FI / IMESPS, DMOO / CMFL, PA, 500, TYPE=BN, FIX, 1, 256, 2, IME.
% FI/FSPS, DMOO / CMFL, SQ, 20, TYPE=BN, FIX, 1, 16, 0.
%AS / OUT, IMESPS, DMOO,, DM.
%CAL / LIB2 / NEWL / OUT ; OUT, L / OPT ; NAT=IME.

%AS / IME, IMESPS, DMOO,, DM.
%CAL/GRG2/?
%/FUNC ; SPS ?
%/MNP ; NAME = IMESPS ?
%/SDP ; SP.
+SYS / PFN = FSYS, PVN = DMOO.
+SDS / PFN = FSDS, PVN = DMOO.
+SPS / PFN = FSPS, PVN = DMOO.
+IN / IMT, S, NAME = IMTSPS.
%EOD.

1.4.5.4. Generation of a user EMI or a system EMI

%JOB. / 0, IMEPG
%FI/ BT, DMOO / CMFL, SQ, 500, TYPE=BN, STD, 1, 104, 1, BT.
%AS / BT, BT, DMOO,, DM.
%CAL/MAS2/MNP; BT.
%FI / IMT, DMOO / CMFL, SQ, 500, TYPE=BN, STD, 1, 104, 1, IMT.
% AS / IMT, IMT, DMOO,, DM.
%CAL/LED2/ ?
%/MNP ; NAME = IMT, PG.
%FI / IME, DMOO / CMFL, PA, 500, TYPE=BN, FIX, 1, 256, 2, IME.
%AS / IME, IME, DMOO,, DM.
%CAL/LIB2/OUT;IME,L ?
%/OPT; NAT = IME.
+NEWL.
%EOD.

%CAL/GRG2/?
%/FUNC ; PRG ?
%/MNP ; NAME = IME2.



```

+SYS / PFN = FSYS, PVN = DMOO.
+SPS / PFN = FSPS, PVN = DMOO.
+SDS / PFN = FSDS, PVN = DMOO.
+IN /IMT, S, NAME = IMT.
ZEOD.

```

1.4.5.5. Generation of a user EMI and extension of the SPS file

```

ZJOB / 0, IMEPG
ZFI / BT, DMOO / CMFL, SQ, 500, TYPE=BN, STD, 1, 104, 1, BT.
ZAS / BT, BT, DMOO,, DM.
ZCAL/MAS2/MNP ; BT.
ZFI /IMT, IMT, DMOO / CMFL, SQ, 500, TYPE=BN, STD, 1, 104, 1, IMT.
ZAS / .IMT, IMT, DMOO,, DM.
ZCAL/LFD2/ ?
Z/MNP ; NAME = IMT. PG.
ZFI /IMF, DMOO /CMFL, PA, 500, TYPE=BN, FLX, 1, 256, 2, IME.
ZAS /IME, IMF, DMOO,, DM.
ZCAL/LIB2/OUT;IME, L?
Z/OPT; NAT = IME.
+NEWL.
ZEOD.
ZCAL/GRG2/?
Z/FUNC ; XPRG ?
Z/MNP ; NAME = IME.
+SYS / PFN = FSYS, PVN = DMOO.
+SPS / PFN = FSPS, PVN = DMOO.
+SDS / PFN = FSDS, PVN = DMOO.
+IN / IMT, S, NAME = IMT.
ZEOD.

```

1.4.5.6. Error messages

- WARNING : (first definition is accepted)

DOUBLE DEFINITION DE DEFY (DOUBLE DEFINITION OF DEFY)

DOUBLE DEFINITION DE SECTION SOUS-PROGRAMME (DOUBLE DEFINITION OF A SUBROUTINE SECTION)

DOUBLE DEFINITION DE SECTION MONITEUR (DOUBLE DEFINITION OF A MONITOR SECTION)

- SERIOUS ERRORS : (generation of a DATA 0 or a BRK 0)

REFX XXXXXX INCONNU ADR YYYY (REFX XXXXXX UNKNOWN ADDRESS YYYY)

CALL XXXXXX INCONNU ADR YYYY (CALL XXXXXX UNKNOWN ADDRESS YYYY)

EXPRESSION NON HOMOGENE ADR YYYY (NON-HOMOGENEOUS EXPRESSION ADDRESS YYYY)

DEBORDEMENT DE CHAMP ADR YYYY (FIELD OVERFLOW ADDRESS YYYY)

EXPRESSION ADRESSAGE ETENDU IMPAIRE ADR YYYY (ODD EXTENDED ADDRESSING EXPRESSION ADDRESS YYYY)

EXPRESSION BASEE NON MULTIPLE DE 16 ADR YYYY (BASED EXPRESSION NOT MULTIPLE OF 16 ADDRESS YYYY)

EXPRESSION ERRONEE ADR YYYY (INCORRECT EXPRESSION ADDRESS YYYY)

- FATAL ERRORS : (Abort)

ERREUR PACK NO &XX (PACK ERROR NO &XX)

ERREUR ANAC NO &XX (ANAC ERROR NO &XX)

ERREUR ORDRE OU UNICITE COMMANDE (ORDER OR COMMAND UNIQUENESS ERROR)

ERREUR LFN POUR LECTURE COMMANDE (LFN ERROR FOR COMMAND READ)

COMMANDE ABSENTE (NO COMMAND)

MODE D'EXECUTION DE LA COMMANDE ILLEGAL// A LA FONCTION (COMMAND EXECUTION MODE ILLEGAL WITH RESPECT TO THE FUNCTION)

TYPE D'IMT INCOMPATIBLE / A LA FONCTION (RMI TYPE INCOMPATIBLE WITH THE FUNCTION)

MODE D'EXECUTION DE L'IMT INCOMPATIBLE / MODE D'EXECUTION DE LA COMMANDE (RMI EXECUTION MODE INCOMPATIBLE WITH COMMAND EXECUTION MODE)

NIVEAU DE SEVERITE DE L'IMT>3 (RMI SEVERITY LEVEL>3)

DROITS FICHER INCOMPATIBLES (FILE RIGHTS INCOMPATIBLE)

PARTITION CSVAT, EDT OU CLQAT VIDE (PARTITION CSVAT, EDT, OR CLQAT EMPTY)

PROGRAMME > 32 K (PROGRAM > 32 K)

ENVIRONNEMENT > 32 K (ENVIRONMENT > 32 K)

FICHER IMT INCORRECT (RMI FILE INCORRECT)

DEBORDEMENT DES TABLES (TABLE OVERFLOW)

1.4.5.7. Standard processing

- initiation incorrect.

The "ABNORMAL START" message is printed out on "EDIT"
Severity level is 6.

- Error in initiation command analysis.

The "ANAC ERROR n" message is printed out on "EDIT", n being the report
given by ANAC. Severity level is 5.

- Error not specifically handled by the processor.

The "(aaaa) ERROR XXXX &lll" message is printed out on "EDIT", with

aaaa	error code
XXXX	function corresponding to the error
llll	&fn on which error was located

Severity level is 4.

- End of processor message (output on dialogue machine).

"ggtt END XXXXXX LEV. n"

ggtt	group and task number
XXXXXX	processor name
n	severity level





