



ICs for Consumer Electronics MEGATEXT

Command Interface

Edition 10.94

MEGATEXT® Command Interface	
Revision History: 10.94	
Previous Releases: 08.93	
Page	Subjects (changes since last revision)

Data Classification

Maximum Ratings

Maximum ratings are absolute ratings; exceeding only one of these values may cause irreversible damage to the integrated circuit.

Characteristics

The listed characteristics are ensured over the operating range of the integrated circuit. Typical characteristics specify mean values expected over the production spread. If not otherwise specified, typical characteristics apply at $T_A = 25\text{ °C}$ and the given supply voltage.

Operating Range

In the operating range the functions given in the circuit description are fulfilled.

For detailed technical information about “**Processing Guidelines**” and “**Quality Assurance**” for ICs, see our “**Short Form Catalog**”.

Edition 10.94

This edition was realized using the software system FrameMaker®.

**Published by Siemens AG, Bereich Halbleiter, Marketing-Kommunikation,
Balanstraße 73, 81541 München.**

© Siemens AG 1994. All Rights Reserved.

As far as patents or other rights of third parties are concerned, liability is only assumed for components, not for applications, processes and circuits implemented within components or assemblies.

The information describes the type of component and shall not be considered as assured characteristics.

Terms of delivery and rights to change design reserved.

For questions on technology, delivery and prices please contact the Semiconductor Group Offices in Germany or the Siemens Companies and Representatives worldwide (see address list).

Due to technical requirements components may contain dangerous substances. For information on the types in question please contact your nearest Siemens Office, Semiconductor Group.

Siemens AG is an approved CECC manufacturer.

Packing

Please use the recycling operators known to you. We can also help you - get in touch with your nearest sales office. By agreement we will take packing material back, if it is sorted. You must bear the costs of transport.

For packing material that is returned to us unsorted or which we are not obliged to accept, we shall have to invoice you for any costs incurred.

Contents	Page
1	Command Directory5
1.1	How to Use an MCI Command5
1.2	Command Handshake6
1.3	Error Code6
1.4	MCI Command Table7
1.5	Command Description8
1.5.1	ACQ_CONTROL8
1.5.2	ADD_PACKET_29_3011
1.5.3	ADD_PAGE13
1.5.4	ADD_ALL_PAGES16
1.5.5	CALL_SUBROUTINE_LEVEL519
1.5.6	CALL_SUBROUTINE_MAIN20
1.5.7	CLEAR_PAGE_TRACE21
1.5.8	COUNT_TOP_PAGENUMBER22
1.5.9	CREATE_FREE_CHAP_CHAIN24
1.5.10	CREATE_FREE_P40_CHAIN25
1.5.11	CREATE_FREE_P80_CHAIN26
1.5.12	EXECUTE_FLOF27
1.5.13	GET_FREE_CHAP31
1.5.14	MOVE_MEMORY_SEG32
1.5.15	NO_OPERATION39
1.5.16	READ_CLEAR_PAGE_TRACE39
1.5.17	READ_GROUP43
1.5.18	READ_PSEUDO_PACKET_TRACE46
1.5.19	REMOVE_PAGE47
1.5.20	REQUEST_ALL_PAGES50
1.5.21	RESET_INTQ52
1.5.22	SEARCH_PACKET_25_2854
1.5.23	SEARCH_PACKET_29_3057
1.5.24	SEARCH_PAGE59
1.5.25	SEARCH_TOP_PAGENUMBER62
1.5.26	SERIAL_PARALLEL_CONVERSION65
1.5.27	TIME_DISPLAY72
1.5.28	WRITE_GROUP74
1.5.29	WRITE_READ_DOUBLEWORD76
1.5.30	WRITE_TOP_TITLE83
1.6	MCI Command Parameter Overview87
2	Appendix89
2.1	Special RAM Addresses and Their Functions89
2.1.1	Initialization89
2.1.2	GROUP_INIT89
2.1.3	X_RAM_SIZE91
2.1.4	SOFTWARE_REFRESH91
2.1.5	SIGNAL_QUALITY91

Contents	Page
2.1.6	SUBROUTINE_CALL_ADDRESS94
2.1.7	VPS95
2.2	Stored Format of Page96
2.3	Stored Format of Linking List of Compressed BTT96
2.4	Stored Format of X/2597
2.5	Stored Format of 1-Byte Hamming Protected Data97
2.6	Stored Format of Packets with 3-Byte Hamming Checks98
2.7	Stored Format of Packets with 6 1-Byte Hamming Checks99
2.8	Stored Format of Packet 8/30 (000x)100
2.9	Stored Format of Packet 8/30 (001x)102
3	Glossary103

MEGATEXT[™] is a registered trademark of Siemens AG.

Purchase of SIEMENS I²C components conveys a licence under the Philips' I²C patent to use the components in the I²C-system provided the system conforms to the I²C specifications defined by Philips.

Copyright Philips 1983.

1 Command Directory

This chapter contains a list of commands from the MEGATEXT® Command Interface MCI in alphabetic order. The documentation for each command contains a statement about the command's purpose, a description of its input parameters with used parameter registers, and a description of its return values with used parameter registers. The documentation for some functions contains additional, important comments that a developer needs in order to use the command.

1.1 How to Use an MCI Command

1. Set all MCI input parameters as described.
2. Set the CMD_RUN bit in MCI input parameter register MCI3_1 bit position MCI3_1_0 to activate MCI handshake.
3. Transmit the MCI command to subaddress MCI_COMMAND (see M3L-Bus Registers Programmer's Reference).
4. As long as the MCI command is in execution, the CMD_RUN bit will be in the 1 condition.
5. Use the MEGATEXT interrupt request signal at pin INTQ or a polling technique to detect if the CMD_RUN bit goes into the 0 condition.
6. If the CMD_RUN bit is in the 0 condition, the MCI is ready to receive a new command.
7. Evaluate the return parameters especially the error code.
8. To poll other interrupt request sources use the RESET_INTQ MCI command.
9. To check a MEGATEXT request to the external controller, use either the INTQ pin for an interrupt technique or the INTQ_BIT located at MCI3_1_1 for a polling technique.

1.2 Command Handshake

An MCI command can be given only if the previous command is terminated. The CMD_RUN bit shows the command status. As long as it is in the 1 condition, a command is in execution. The external controller can check this bit before giving a new command. The user is responsible for setting the CMD_RUN bit before giving the MCI command.

COMMAND_STATUS = MCI3_1

0	0	0	0	0	0	INTQ_BIT	CMD_RUN
---	---	---	---	---	---	----------	---------

CMD_RUN

MCI command running flag

1: MCI command is in execution.

0: MCI command is terminated.

INTQ_BIT

Interrupt request at INTQ pin

The INTQ_BIT can be used to check MEGATEXT service requests with a polling technique instead of using the INTQ pin.

1: An interrupt request was given at INTQ pin.

0: No interrupt request was given at INTQ pin.

1.3 Error Code

Many commands return an error code to acknowledge the command result. It looks as follows:

ERROR_CODE = MCI3_2

ERR_7	ERR_6	ERR_5	ERR_4	ERR_3	ERR_2	ERR_1	ERR_0
-------	-------	-------	-------	-------	-------	-------	-------

ERR_7 thru ERR_0 Error code bits are explained in the command description. Unless stated otherwise, MCI3_2 is used for the error code register.

1.4 MCI Command Table

The following table shows the command name to command number assignment:

Command Name	Command Number
ACQ_CONTROL	0
ADD_PACKET_29_30	2
ADD_PAGE	4
ADD_ALL_PAGES	1
CALL_SUBROUTINE_LEVEL5	16
CALL_SUBROUTINE_MAIN	5
CLEAR_PAGE_TRACE	6
COUNT_TOP_PAGENUMBER	11
CREATE_FREE_CHAP_CHAIN	7
CREATE_FREE_P40_CHAIN	8
CREATE_FREE_P80_CHAIN	9
EXECUTE_FLOF	3
GET_FREE_CHAP	12
MOVE_MEMORY_SEG	17
NO_OPERATION	19
READ_FLOF_LINK_PAGES	22
READ_GROUP	23
READ_CLEAR_PAGE_TRACE	54
READ_PSEUDO_PACKET_TRACE	24
REMOVE_PAGE	31
REQUEST_ALL_PAGES	32
RESET_INTQ	37
SEARCH_PACKET_25_28	39
SEARCH_PACKET_29_30	40

MCI Command Table (cont'd)

Command Name	Command Number
SEARCH_PAGE	41
SEARCH_TOP_PAGENUMBER	45
SERIAL_PARALLEL_CONVERSION	47
TIME_DISPLAY	49
WRITE_GROUP	50
WRITE_READ_DOUBLEWORD	51
WRITE_TOP_TITLE	53

1.5 Command Description**1.5.1 ACQ_CONTROL**

Switches on/off TTX and VPS packet reception from the serial data input to the packet buffer and data acquisition from packet buffer to page and packet memories. In addition, the framing code reference, automatic request of TOP tables, VPS reception and page trace on/off switching can be controlled.

Input Parameters**Packet Buffer Control****PKT_BUF_ACQ_CONTROL = MCIO_0**

R_MPEX	R_MP	VPS_ENA	0	TTX_FC	ACQ_BUF_ON	R_AI	PAGE_TRACE
--------	------	---------	---	--------	------------	------	------------

PAGE_TRACE

Switches on/off page trace recording in the page trace memory.

1: Page trace recording is switched on.

0: Page trace recording is switched off.

R_AI

1: Automatic request of all Additional TOP Tables is enabled.

0: Automatic request of all Additional TOP Tables is disabled.

ACQ_BUF_ON

Controls switching on/off TTX data transfer to packet buffer and TTX data acquisition.

1: Data transfer to packet buffer and ACQ is switched on.

0: Data transfer to packet buffer and ACQ is switched off.

TTX_FC

Teletext framing code redefinition.

1: Framing code of input parameter register TTX_FRAMING_CODE is taken to redefine framing code reference for acquisition.

0: Framing code remains the same as before giving this command, independently of register TTX_FRAMING_CODE.

- VPS_ENA** Controls switching on/off VPS reception (ACQ_BUF_ON must be switched on).
 1: VPS reception is switched on.
 0: VPS reception is switched off.
- R_MP** 1: Automatic request of all Multipage TOP Tables is enabled.
 0: Automatic request of Multipage TOP Tables is disabled.
- R_MPEX** 1: Automatic request of all extended Multipage TOP Tables is enabled.
 0: Automatic request of extended Multipage TOP Tables is disabled.

TTX Framing Code

TTX_FRAMING_CODE = MCIO_1

TTX_FC_7	TTX_FC_6	TTX_FC_5	TTX_FC_4	TTX_FC_3	TTX_FC_2	TTX_FC_1	TTX_FC_0
----------	----------	----------	----------	----------	----------	----------	----------

TTX_FC_7 thru TTX_FC_0 The framing code is compared by data acquisition with the received teletext framing code as a reference for byte synchronization.

Return Values

None.

Error Code

ERR_7 – ERR_0	Description
0	Command execution was successful.
1-255	Not defined.

Comments

1. Before using this command to switch on packet reception, the packet buffer must have been declared in the M3L packet buffer registers PB_ADR, PB_LENGTH and the M3L memory allocation registers IAT or XAT.
2. If packet reception is switched on, the packet buffer is initialized to 0, the ACQ reset and started.
3. To ensure proper working of the ACQ, all ACQ-relevant input parameters must have been declared. For more details see document "ACQ Reference".
4. To receive VPS data ensure that the M3L registers SINGLE_DATA_LINE and the EXTRA_FRAMING_WINDOW are set correctly.
If teletext and not VPS data are transmitted in the VPS data line (16), errors will occur during teletext reception.
5. If the automatic TOP_TABLE request option is used, ACQ group 5 must be declared with a sub-code do care qualification. If memory overflow occurred during allocation of memory space, an interrupt is set. The ACQ control bits are automatically set in the following way:

DCR = 0	HEN = 0	0	IREQ = 1	PBR = 0	PBLF = 1	FIRST = 0	0
DIS_PP = 0	0	0	SM_1 = 0	SM_0 = 0	0	CHECK_1 = 1	CHECK_0 = 0 or CHECK_0 = 1

6. The check bits are automatically set so that additional information tables, multipage tables and multipage extension tables are checked in the way described in the TOP specification of the IRT.
7. If one of the bits R_AI, R_MP or R_MPEX is set, after reception of the BTT the acquisition is searching for link information in the Basic-TOP-Table-List allocates memory space and requests user-optional the Additional-Information-TOP-Tables, Multipage-TOP-Tables or Multipage-Extension-TOP-Tables. The pages will always be requested in PRQ group 5. Before enabling this option, ensure that the PRQ group record has been declared with search type 2. Use WRITE_GROUP for this command.
8. The configuration of the PAGE_TRACE memory can be done with the M3L registers 123-125.

1.5.2 ADD_PACKET_29_30

This command adds a packet data record for packets X/29 or 8/30 with the given packet request record and ACQ control data to a PRQ groups type 3 chain.

Input Parameters

Packet Control Bits

Packet control bits. For details of bit see document "ACQ Reference".

PACKET_CONTROL = MCIO_5

0	PCKHEN	CHECK_OFF	PCKIREQ	0	PCKBLF	0	0
---	--------	-----------	---------	---	--------	---	---

CHECK_OFF 1: Packets are stored without any checks (except row magazine number and designation code).
 0: Packets are checked before storing.
 (see Appendix "Stored Format of ...").

PRQ Group Number

The PRQ group number selects the group record to be used.

PRQ_GROUP = MCIO_1

0	0	0	0	0	PRQ_GROUP_2	PRQ_GROUP_1	PRQ_GROUP_0
---	---	---	---	---	-------------	-------------	-------------

PRQ_GROUP_2 thru PRQ_GROUP_0 PRQ group number bits, $0 \leq \text{PRQ_GROUP} \leq 7$.

PRQ Record

The PRQ record determines the packet request bits which will be compared with the incoming TTX data.

Before comparison, search type 3 PRQ records will be qualified with the PRQ do care bits in the associated group.

PRQ_RECORD_4 = MCII_5

ROW4	ROW3	ROW2	ROW1	ROW0	M2	M1	M0
------	------	------	------	------	----	----	----

PRQ_RECORD_3 = MCII_4

0	0	0	0	DES3	DES2	DES1	DES0
---	---	---	---	------	------	------	------

All above listed request bits have the same meaning as defined in the world system teletext specification.

Return Values

Actual number of elements in the free p80 chain in M3L-Bus register NF_P80.

Actual number of allocated elements of the p80 chain in the associated group record GROUPX_NA.

Only for adding the first element: pointer to the first element of the p80 chain in the associated group record GROUPX_FAP.

All return values in the control row of the previous and added p80 record.

Error Code

ERR_7 – ERR_0	Description
0	Command execution was successful.
1	No p80 element available in free p80 chain.
2	Given group contains unsupported type or group is not used.
3	The given packet number is not unique (there is another element in this group with the same packet number).
4 – 255	Not defined.

Comment

Before adding the first p80 element, ensure that a PRQ group record with search type 3 has been declared. Use WRITE_GROUP for this command.

1.5.3 ADD_PAGE

This command adds a page with the given page request record and ACQ control data to a specified PRQ group. The search type of the group is given with the command WRITE_GROUP. It is also possible to change the subpage mode of an existing type 0 subpage chain.

Input Parameters

ACQ Control Bits

Data acquisition control bits. For bit description see document “ACQ Reference”.

ACQ_CONTROL_1 = MCIO_5

DCR	HEN	0	IREQ	PBR	PBLF	FIRST	C_BTT
-----	-----	---	------	-----	------	-------	-------

ACQ_CONTROL_0 = MCIO_4

DIS_PP	P_OV_OC = 0	OV_OCCU = 0	SM_1	SM_0	REM_PROT	CHECK_1	CHECK_0
--------	-------------	-------------	------	------	----------	---------	---------

Maximum Number of Subpages

Maximum number of subpages which are linked to the basic page if search type 0 is chosen. For any other type these bits are ignored. For details of bit see document “ACQ Reference”.

NMAX_SUB = MCIO_3

S_MAX_7	S_MAX_6	S_MAX_5	S_MAX_4	S_MAX_3	S_MAX_2	S_MAX_1	S_MAX_0
---------	---------	---------	---------	---------	---------	---------	---------

$0 \leq S_MAX \leq 255$.

If $S_MAX = 255$ then the subpage collect mode is enabled.

PRQ Group Number

The PRQ group number selects the group record to be used.

PRQ_GROUP = MCIO_1

MOD_SUB_REQ	REQUEST_STAT	0	0	0	PRQ_GROUP_2	PRQ_GROUP_1	PRQ_GROUP_0
-------------	--------------	---	---	---	-------------	-------------	-------------

PRQ_GROUP_2 thru PRQ_GROUP_0 PRQ group number bits. $0 \leq PRQ_GROUP \leq 7$.

REQUEST_STAT

- 1: The (page request in progress) bit in the PAGE_REQUEST_STATUS_TABLE corresponding to the basic page number to be requested is set. The start address of the PAGE_REQUEST_STATUS_TABLE could be found in M3L registers R123 – R125 (PT_ADR). The REQUEST_STATUS_TABLE is located under the same chapter address as the page trace but with an offset of 512 (binary). The format of the bits is explained in document “ACQ Reference”.
- 0: The REQUEST_STATUS_TABLE is not influenced.

MOD_SUB_REQ**Modify subpage request. Only for groups of type 0.**

- 1: Modify subpage request. **No** page request is added. Only the subpage mode of the subpage chain given by PRQ_RECORD is changed. The use of this bit only makes sense if a group with search type 0 is chosen.
- 0: The page request specified by PRQ_RECORD is added and the subpage mode is switched to the mode given by NMAX_SUB, SM_1 and SM_0.

PRQ Record

The PRQ record determines the PRQ bits which will be compared with the incoming TTX header data.

Before comparison, search type 2 PRQ records will be qualified with the PRQ do care bits in the associated group.

PRQ_RECORD_4 = MCI1_5

0	0	0	0	0	M2	M1	M0
---	---	---	---	---	----	----	----

PRQ_RECORD_3 = MCI1_4

PT3	PT2	PT1	PT0	PU3	PU2	PU1	PU0
-----	-----	-----	-----	-----	-----	-----	-----

PRQ_RECORD_2 = MCI1_3

C4	MT2	MT1	MT0	MU3	MU2	MU1	MU0
----	-----	-----	-----	-----	-----	-----	-----

PRQ_RECORD_1 = MCI1_2

C6	C5	HT1	HT0	HU3	HU2	HU1	HU0
----	----	-----	-----	-----	-----	-----	-----

PRQ_RECORD_0 = MCI1_1

C14	C13	C12	C11	C10	C9	C8	C7
-----	-----	-----	-----	-----	----	----	----

All above listed request bits have the same meaning as defined in the world system teletext specification.

If search type 0 is chosen, bits C14-C4 have to be set to 0.

Return Values

Actual number of elements in the free chap chain in M3L-Bus register NF_CHAP.

Actual number of allocated elements of the chap chain in the associated group record GROUPX_NA.

Only for adding the first element: pointer to the first element of the chap chain in the associated group record GROUPX_FAP.

For type 0 these are the pointers in the PAGE_TLU (page table lookup).

Error Code

ERR_7 – ERR_0	Description
0	Command execution was successful.
1	No chap element available in free chap chain.
3	The given PRQ record is not unique (there is another element in this group with the same page number).
4	Modify subpage request (MOD_SUB_REQ) is used before any page is requested.
5	PAGE_REQUEST_STATUS_TABLE is not enabled although REQUEST_STAT is used.
2,6 – 255	Not defined.

Comments

1. Before adding the first chap element, ensure that a PRQ group record with the given search type has been declared. For this use command WRITE_GROUP.
2. If another page exists with the same PRQ record, remove it before giving the ADD_PAGE command.
3. If the associated group contains type 2, the page numbers are compared with respect to the GROUP_DO_CARE bits (see MCI command WRITE_GROUP). PRQ record comparison matches if all PRQ record bits qualified with do care are equal. PRQ record bits qualified with don't care must not be equal to lead to the match condition.
4. If the PAGE_REQUEST_STATUS_TABLE is used, one chapter must be reserved in the IAT/XAT register and the start address of this chapter must be given in PT_ADR (M3L register R123 – R125).

1.5.4 ADD_ALL_PAGES

This command supports the request of pages in binary order, or pages marked in the basic TOP table or pages recorded in the PAGE_TRACE.

If the TOP table should be evaluated, beginning from a given start page, TOP block pages, group pages or pages of a group can be searched and, if found, marked in the PAGE_REQUEST_STATUS_TABLE as requests to be intended. The allocation of memory space and so the request of these pages can be done by use of the command REQUEST_ALL_TABLES.

Input Parameters

Basic TOP Tables

The basic TOP table must have been received before giving this command and be located as the first page in ACQ group 6.

PRQ Record of First Requested Basic Page

Determines the PRQ bits of the first basic page to be requested. If this page number is not present in the PAGE_TRACE and PAGE_TRACE is selected, the first page to be requested is the page with the next highest (lowest) page number.

Start Page

The start page determines the first page number to be added.

PRQ_RECORD_I = MCIO_4

0	0	0	0	0	M2	M1	M0
---	---	---	---	---	----	----	----

PRQ_RECORD_0 = MCIO_3

PT3	PT2	PT1	PT0	PU3	PU2	PU1	PU0
-----	-----	-----	-----	-----	-----	-----	-----

All above listed request bits have the same meaning as defined in the world system teletext specification.

REQUEST_MODE

Determines the mode of the auto-increment (decrement) of the page number. It is not permissible to combine the four different auto-increment modes (TOP_DIRECT, TOP, PAGE_TRACE, BINARY).

REQUEST_MODE = MCIO_1

0	UP_DOWN	PSEUDO_PAGES	REQUEST_OVERFLOW	0	TOP	PAGE_TRACE	BINARY
---	---------	--------------	------------------	---	-----	------------	--------

- BINARY** The page numbers are binary incremented and the corresponding bit in the PAGE_REQUEST_STATUS_TABLE will be set until the number of pages given in N_BASIC_PAGES is reached.
- PAGE_TRACE** All pages which are marked in the page trace will also be marked in the PAGE_REQUEST_STATUS_TABLE until the number of pages given in N_BASIC_PAGES is reached.
- TOP** The pages to be marked in the PAGE_REQUEST_STATUS_TABLE are specified by the contents of the basic TOP table and the parameter TOP_TYPE. The maximum number of pages to be requested is given by N_BASIC_PAGES.
- REQUEST_OVERFLOW** 1: If the end (beginning) of the PAGE_TRACE, the BTT or the last page of the BINARY increment mode is reached before NUMBER OF PAGES, the request will go on at the beginning (end) of these tables or pages.
0: If the end (beginning) of the PAGE_TRACE or the last page of the BINARY increment mode is reached, the request of pages will be stopped.
- PSEUDO_PAGES** 1: Pseudo pages (pages with page number tens or page number units bigger than 9) are also requested.
0: Pseudo pages (pages with page number tens or page number units bigger than 9) are skipped.
- UP_DOWN** (only in binary or page trace mode)
1: Page numbers are requested in ascending order.
0: Page numbers are requested in descending order.

Number of Basic Pages to Be Requested

Determines the maximum number of basic pages to be requested.

N_BASIC_PAGES_1 = MCI1_1

0	0	0	0	NBP_11	NBP_10	NBP_9	NBP_8
---	---	---	---	--------	--------	-------	-------

N_BASIC_PAGES_0 = MCI1_0

NBP_7	NBP_6	NBP_5	NBP_4	NBP_3	NBP_2	NBP_1	NBP_0
-------	-------	-------	-------	-------	-------	-------	-------

NBP_11 – NBP_0 $0 \leq \text{NBP}_{11} \text{ thru } \text{NBP}_0 \leq 2047$.

TOP_TYPE = MCI0_0

These bits are only relevant if TOP is selected in REQUEST_MODE.

0	0	0	0	0	BL	GR	PA
---	---	---	---	---	----	----	----

The different request modes are shown in the following table:

BL GR PA	Description of BL, GR and PA Bits
000	Not defined.
001	Beginning from the given start page all pages of the current group are requested.
010	Beginning from the given start page all group pages of the current block are requested.
011	Beginning from the given start page all group pages of the current block and all pages of these groups are requested.
100	Beginning from the given start page all block pages of the BTT are requested.
101	Not defined.
110	Beginning from the given start page all block and all group pages are requested.
111	Beginning from the given start page all pages are requested.

Return Values

Error Code

ERR_7 – ERR_0	Description
0	Command execution was successful.
1	Basic TOP table not received.
2	PAGE_REQUEST_STATUS_TABLE is not enabled.
3 – 255	Not defined.

Comments

1. Before using this command it must be ensured that memory space for the request table is reserved in the IAT/XAT register. The start address of the table is given in M3L register PT_ADR (R123 – R125) plus an offset of 768 (binary address scheme is used). The format of the table is explained in document “ACQ Reference”.
2. Block pages are pages of the basic TOP table with code 1,2,3,4 or 5.
3. Group pages are pages of the basic TOP table with code 6 or 7.
4. Normal pages are pages of the basic TOP table with code 8,9,A or B.
5. Before the BTT is received for the first time, all bytes of this chapter are automatically set to 20_H.

1.5.5 CALL_SUBROUTINE_LEVEL5

Calls the subroutine given by its calling address. This subroutine is processed with an higher interrupt priority (MEGATEXT interrupt level 5) and cannot be interrupted by acquisition or timing interrupts.

Input Parameters

Subroutine Calling Address

The subroutine calling address is the binary ROM or IRAM address given in IRAM address CALL_SUB_ADR as listed in the Appendix.

Return Values

Depends on the called subroutine.

Error Code

ERR_7 – ERR_0	Description
0	Command execution was successful.
1 – 255	Not defined.

Comment

The CALL_SUB_ADR bits must be written over M3L-Bus data port 0 or 1.

1.5.6 CALL_SUBROUTINE_MAIN

Calls the subroutine given by its calling address. This subroutine is executed in the main program of the MEGATEXT firmware and has lower priority than all other interrupt routines.

Input Parameters

Subroutine Calling Address

The subroutine calling address is the binary ROM or IRAM address given in IRAM address CALL_SUB_ADR as listed in the Appendix.

Return Values

Depends on the called subroutine.

Error Code

ERR_7 – ERR_0	Description
0	Command execution was successful.
1 – 255	Not defined.

Comment

The CALL_SUB_ADR bits must be written over M3L-Bus data port 0 or 1.

1.5.7 CLEAR_PAGE_TRACE

This command clears the page trace memory, i.e. all bits be set to 0.

Input Parameters

None.

Return Values

None.

Comment

This command is a very simple version of READ_CLEAR_PAGE_TRACE and is only included because of software compatibilities with older versions.

1.5.8 COUNT_TOP_PAGENUMBER

This command counts TOP pages of a certain type.
It is always assumed that the basic TOP table is the first page of ACQ group 6.

Input Parameters

TOP Page Type Selection

This is the page type which should be counted.

TOP_PAGE_TYPE = MCIO_5

0	0	0	0	TP_3	TP_2	TP_1	TP_0
---	---	---	---	------	------	------	------

TP_3 thru TP_0 TOP page type bits

They define the type of page in the basic TOP table (block, group, subtitle, program review, normal page) and have the same meaning as described in the TOP specification of the IRT.

Search Start Page Number

This is the page number which specifies the current block or group.

PAGE_NUMBER_1 = MCIO_4

0	0	0	0	0	M2	M1	M0
---	---	---	---	---	----	----	----

PAGE_NUMBER_0 = MCIO_3

PT3	PT2	PT1	PT0	PU3	PU2	PU1	PU0
-----	-----	-----	-----	-----	-----	-----	-----

All above listed bits have the same meaning as defined in the world system teletext specification.

Return Values

Number of Found Pages

Number of pages named in basic TOP table for the selected search page type.

N_PAGE_BTT_1 = MCIO_1

0	0	0	0	0	0	NPBT_9	NPBT_8
---	---	---	---	---	---	--------	--------

N_PAGE_BTT_0 = MCIO_0

NPBT_7	NPBT_6	NPBT_5	NPBT_4	NPBT_3	NPBT_2	NPBT_1	NPBT_0
--------	--------	--------	--------	--------	--------	--------	--------

NPBT_9 thru NPBT_0 Number of pages bits
0 ≤ NPBT_9 thru NPBT_0 ≤ 800.

Number of pages of the selected search type counted in the current block given by PAGE_NUMBER.

N_PAGE_BLO_1 = MCI1_4

0	0	0	0	0	0	NPBL_9	NPBL_8
---	---	---	---	---	---	--------	--------

N_PAGE_BLO_0 = MCI1_3

NPBL_7	NPBL_6	NPBL_5	NPBL_4	NPBL_3	NPBL_2	NPBL_1	NPBL_0
--------	--------	--------	--------	--------	--------	--------	--------

NPBL_9 thru NPBL_0 Number of pages bits
 $0 \leq \text{NPBL}_9 \text{ thru } \text{NPBL}_0 \leq 800.$

Number of pages of the selected search type counted in the current group given by PAGE_NUMBER.

N_PAGE_BGR_1 = MCI1_1

0	0	0	0	0	0	NPGR_9	NPGR_8
---	---	---	---	---	---	--------	--------

N_PAGE_BGR_0 = MCI1_0

NPGR_7	NPGR_6	NPGR_5	NPGR_4	NPGR_3	NPGR_2	NPGR_1	NPGR_0
--------	--------	--------	--------	--------	--------	--------	--------

NPGR_9 thru NPGR_0 Number of pages bits
 $0 \leq \text{NPGR}_9 \text{ thru } \text{NPGR}_0 \leq 800.$

Error Code

ERR_7 – ERR_0	Description
0	Command execution was successful.
1	Basic TOP table not received.
2 – 255	Not defined.

Comments

1. Before the BTT is received for the first time, all bytes of this chapter are automatically set to 20_H.
2. Received bytes with hamming errors are counted like pages with code 0 (page not in transmission cycle).

1.5.9 CREATE_FREE_CHAP_CHAIN

This command creates the free chap chain.

Input parameters

Address pointer to first free chap chain element as registered in M3L register FFP_CHAP.

Unallocated memory as registered in M3L-Bus register IAT.

Unallocated memory as registered in M3L-Bus register XAT.

Memory size of XRAM as registered in IRAM register SIZE_XRAM.

Requested number of free elements of free chap chain. This is identical to M3L register NF_CHAP.

Return Values

Actual number of free elements of the chap chain in M3L-Bus register NF_CHAP.

Error Code

ERR_7 – ERR_0	Description
0	Creation of chap chain was successful.
1	Number of requested elements > number of available elements.
2 – 255	Not defined.

Comments

1. The CREATE_FREE_CHAP_CHAIN command first allocates free chapters in IRAM and then in XRAM in ascending order.
2. Only one free chap chain can be created.
3. Each chap chain element allocates one chapter of memory. All chapters start addresses are aligned at addresses with row/column = 0/0.
4. After this command the M3L register NF_CHAP must be interpreted as actual number of free elements of the free chap chain.
5. Before acquisition is started again, the user has to erase the page trace and the page request status stable with the command "READ_CLEAR_PAGE_TRACE".

How to use this command:

1. Allocate customized memory with IAT and/or XAT.
2. Set address pointer to first free chap chain element in M3L register FFP_CHAP.
3. Set FFP_40 and FFP_80 to zero.
4. Set requested number of free chap elements in M3L register NF_CHAP.

1.5.10 CREATE_FREE_P40_CHAIN

This command creates the free p40 chain.

Input Parameters

Address pointer to first free p40 chain element as registered in M3L register FFP_P40.

Unallocated memory as registered in M3L-Bus register IAT.

Unallocated memory as registered in M3L-Bus register XAT.

Memory size of XRAM as registered in IRAM register SIZE_XRAM.

Requested number of free elements of free p40 chain. This is identical to M3L register NF_P40.

Return Values

Actual number of free elements of the p40 chain in M3L-Bus register NF_P40.

Error Code

ERR_7 – ERR_0	Description
0	Creation of p40 chain was successful.
1	Number of requested elements > number of available elements.
2 – 255	Not defined.

Comments

1. Before creation of the free p40 chain, the free chap chain has to be created!
2. The CREATE_FREE_P40_CHAIN command first allocates free p40 elements in IRAM and then in XRAM in ascending order.
3. Only one free p40 chain can be created.
4. Each list element allocates one row (only rows 0 to 24 are used) of memory.

How to use this command:

5. Allocate customized memory with IAT and/or XAT.
6. Set address pointer to first free p40 chain element in M3L register FFP_P40.
7. Set requested number of free p40 elements in M3L register NF_P40.

1.5.11 CREATE_FREE_P80_CHAIN

This command creates the free p80 chain.

Input Parameters

Address pointer to first free p80 chain element as registered in M3L register FFP_P80.

Unallocated memory as registered in M3L-Bus register IAT.

Unallocated memory as registered in M3L-Bus register XAT.

Memory size of XRAM as registered in IRAM register SIZE_XRAM.

Requested number of free elements of free p80 chain. This is identical to M3L register NF_P80.

Return Values

Actual number of free elements of the p80 chain in M3L-Bus register NF_P80.

Error Code

ERR_7 – ERR_0	Description
0	Creation of p80 chain was successful.
1	Number of requested elements > number of available elements.
2 – 255	Not defined.

Comments

1. If no p80 chain is desired, set NF_P80 to zero.
2. The CREATE_FREE_P80_CHAIN command first allocates free p80 elements in IRAM and then in XRAM in ascending order.
3. Only one free p80 chain can be created.
4. Each chain element allocates two rows (only rows 0 to 23 are used) of memory.
5. Take notice of the maximum number of p80 elements as described in document “ACQ Reference”.
6. Byte 0 to 5 of row 4 and row 5 of block 0 of the internal RAM are reserved for very small p80 chains. If the number of requested p80 elements is not bigger than six these rows may also be used for storing p80 elements. The address pointer to the first free p80 chain element in M3L register FFP_P80 should then be: BLOCK0 BYTE0 ROW4 COLUMN0.

How to use this command:

1. Allocate customized memory with IAT and/or XAT.
2. Set address pointer to first free p80 chain element in M3L register FFP_P80.
3. Set requested number of free p80 elements in M3L register NF_P80.

1.5.12EXECUTE_FLOF

This command reads and returns the linked information given by a page number in PRQ_RECORD and the corresponding packet 27/0000 and a “FLOF color”.

It is also possible to change automatically the display chapter and to request the five new linked FLOF pages. In this case the old display chapter and the old unused FLOF links (up to four pages) are removed if they are not linked from the new display chapter.

Input Parameters

FLOF Color

FLOF_COLOR = MCIO_0

0	0	0	0	0	FLOF_C_2	FLOF_C_1	FLOF_C_0
---	---	---	---	---	----------	----------	----------

The meaning of the FLOF_C bits is shown in the following table:

FLOF_C_2 – FLOF_C_0	Color	Description
000	Red	Link to next page (see also FLOF_MODE).
001	Green	Link to next page (see also FLOF_MODE).
010	Yellow	Link to next page (see also FLOF_MODE).
011	Cyan	Link to next page (see also FLOF_MODE).
100	Not defined.	Not defined.
101	Not defined.	Link to index page (see also FLOF_MODE).
110	Not defined.	Not defined.
111	Not defined.	Not defined.

PRQ Group Number

The PRQ group number selects the group record to be used.

PRQ_GROUP = MCIO_1

REM_PROT	0	0	0	0	PRQ_GROUP_2	PRQ_GROUP_1	PRQ_GROUP_0
----------	---	---	---	---	-------------	-------------	-------------

PRQ_GROUP_2 thru PRQ_GROUP_0 PRQ group number bits, 0 ≤ PRQ_GROUP ≤ 7.

REM_PROT:1 Pages protected by the bit REM_PROT (this bit can be set with the other ACQ control bits) are also removed.

REM_PROT:0 Pages are only removed if they are not remove protected.

FLOF Mode

The FLOF mode bits are only defined if FLOF_COLOR 000, 001, 010, 011 or 101 is selected.

FLOF_MODE = MC10_2

0	0	0	0	0	MODE_2	MODE_1	MODE_0
---	---	---	---	---	--------	--------	--------

MODE_2 – MODE_0	Description
000	The found linked page number is returned to L_PAGE_NUMBER.
001	The found linked page number is returned to L_PAGE_NUMBER. The linked page is automatically requested and the display chapter is changed to the new page number. The pointer of the new display chapter is returned to AP_CHAP.
010	Not defined.
011	The found linked page number is returned to L_PAGE_NUMBER. The linked page is automatically requested and the display chapter is changed to the new page number. The pointer of the new display chapter is returned to AP_CHAP. The five new linked pages are automatically requested.
100	Not defined.
101	The found linked page number is returned to L_PAGE_NUMBER. The linked page is automatically requested and the display chapter is changed to the new page number. The pointer of the new display chapter is returned to AP_CHAP. The old display chapter and the four unused linked pages of this page are automatically removed, if they are not linked from the new display chapter.
110, 111	Not defined.

PRQ Record

The PRQ record determines the page number which should be FLOF processed.

PRQ_RECORD_4 = MC11_5

0	0	0	0	0	M2	M1	M0
---	---	---	---	---	----	----	----

PRQ_RECORD_3 = MC11_4

PT3	PT2	PT1	PT0	PU3	PU2	PU1	PU0
-----	-----	-----	-----	-----	-----	-----	-----

PRQ_RECORD_2 = MCI1_3

0	MT2	MT1	MT0	MU3	MU2	MU1	MU0
---	-----	-----	-----	-----	-----	-----	-----

PRQ_RECORD_1 = MCI1_2

0	0	HT1	HT0	HU3	HU2	HU1	HU0
---	---	-----	-----	-----	-----	-----	-----

PRQ_RECORD_0 = MCI1_1

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

All above listed display page number bits have the same meaning as defined in the world system teletext specification.

Return Values

Linked Page Numbers

The linked page number is returned in MCI1.

L_PAGE_NUM_4 = MCI1_5

0	0	0	0	0	M2	M1	M0
---	---	---	---	---	----	----	----

L_PAGE_NUM_3 = MCI1_4

PT3	PT2	PT1	PT0	PU3	PU2	PU1	PU0
-----	-----	-----	-----	-----	-----	-----	-----

L_PAGE_NUM_2 = MCI1_3

0	MT2	MT1	MT0	MU3	MU2	MU1	MU0
---	-----	-----	-----	-----	-----	-----	-----

L_PAGE_NUM_1 = MCI1_2

0	0	HT1	HT0	HU3	HU2	HU1	HU0
---	---	-----	-----	-----	-----	-----	-----

L_PAGE_NUM_0 = MCI1_1

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

All above listed linked page number bits have the same meaning as defined in the world system teletext specification. The bits C4, C5, C6 are used as the related magazine number for calculating the actual magazine number given in M2-M0.

Address Pointer to New Display Page

This is the pointer to the new display chapter.

AP_CHAP_2 = MCI3_5

EXT_MEM = 0	0	0	0	BYT_5	BYT_4	BYT_3	BYT_2
EXT_MEM = 1		CHP_10	CHP_9	CHP_8	CHP_7	CHP_6	CHP_5

AP_CHAP_1 = MCI3_4

BYT_1	BYT_0	BLK_2	BLK_1	BLK_0	0	0	0
CHP_4	CHP_3	CHP_2	CHP_1	CHP_0			

Error Code

ERR_7 – ERR_0	Description
0	Command execution was successful.
1	No chap element available in free chap chain.
2	No packet x/27 found (FLOF tree ends here or packet still not received).
3 – 255	Not defined.

Comments

1. The REQUEST_STATUS_TABLE is not influenced by the automatic request of the FLOF pages.
2. Do not forget that pages which are added in the same group by use of the command ADD_PAGE or REQUEST_ALL_PAGES can be removed with the command EXECUTE_FLOF.
3. Pages which are automatically requested get the ACQ control bits 04_H, 00_H.
4. Do not forget to reserve enough memory space for the automatic request of FLOF pages (at least five chapters).

1.5.13 GET_FREE_CHAP

This command takes a free chap chain element and links it to the free p40 chain.

Input Parameters

None.

Return Values

Actual number of free elements of the free p40 chain in M3L-Bus register NF_P40.

Actual number of free elements of the free chap chain in M3L-Bus register NF_CHAP.

Error Code

ERR_7 – ERR_0	Description
0	Command execution was successful.
1	No chap elements available in free chap chain.
2 – 255	Not defined.

Comment

This command is useful to increase the free p40 chain after getting the INTQ request by interrupt source “pseudo packet overflow”.

1.5.14 MOVE_MEMORY_SEG

This command moves any internal or external source RAM memory byte segment to an internal or external destination RAM memory byte segment. Find forward, find backward and substitution capabilities are supported.

Input Parameters

Source Segment Start Address

MOVE_SOURCE_SEG_START_2 = MCI0_5

EXT_MEM = 0	0	0	0	BYT_5	BYT_4	BYT_3	BYT_2
EXT_MEM = 1		CHP_10	CHP_9	CHP_8	CHP_7	CHP_6	CHP_5

MOVE_SOURCE_SEG_START_1 = MCI0_4

BYT_1	BYT_0	BLK_2	BLK_1	BLK_0	ROW_4	ROW_3	ROW_2
CHP_4	CHP_3	CHP_2	CHP_1	CHP_0			

MOVE_SOURCE_SEG_START_0 = MCI0_3

ROW_1	ROW_0	COL_5	COL_4	COL_3	COL_2	COL_1	COL_0
-------	-------	-------	-------	-------	-------	-------	-------

EXT_MEM

Internal/external memory selection.

0: Values in the MOVE_SOURCE_SEG_START registers are related to internal RAM memory.

BYT_5 thru BYT_0 **Byte position of internal memory doubleword**
Only one BYT bit must be selected.

BLK_2 thru BLK_0 **Block of internal memory**
 $0 \leq \text{block address} \leq 3$.

ROW_4 thru ROW_0 **Internal memory row address**
 $0 \leq \text{row address} \leq 25$.

COL_5 thru COL_0 **Internal memory column address**
For $0 \leq \text{row address} \leq 24$ is $0 \leq \text{column address} \leq 39$.
For row address = 25 is $0 \leq \text{column address} \leq 23$.

1: Values in the MOVE_SOURCE_SEG_START registers are related to external RAM memory.

CHP_10 thru CHP_0 **External memory chapter address**
 $0 \leq \text{chapter address} \leq 2047$.

ROW_4 thru ROW_0 **External memory row address**
 $0 \leq \text{row address} \leq 25$.

COL_5 thru COL_0 **External memory column address**
For $0 \leq \text{row address} \leq 24$ is $0 \leq \text{column address} \leq 39$.
For row address = 25 is $0 \leq \text{column address} \leq 23$.

Source Segment End Address**MOVE_SOURCE_SEG_END_2 = MCI0_2**

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

MOVE_SOURCE_SEG_END_1 = MCI0_1

0	0	0	0	0	ROW_4	ROW_3	ROW_2
---	---	---	---	---	-------	-------	-------

MOVE_SOURCE_SEG_END_0 = MCI0_0

ROW_1	ROW_0	COL_5	COL_4	COL_3	COL_2	COL_1	COL_0
-------	-------	-------	-------	-------	-------	-------	-------

ROW_4 thru ROW_0 Row address for end of memory source segment $0 \leq \text{row address} \leq 25.$ **COL_5 thru COL_0 Column address for end of memory source segment**For $0 \leq \text{row address} \leq 24$ is $0 \leq \text{column address} \leq 39.$ For row address = 25 is $0 \leq \text{column address} \leq 23.$ **Destination Segment Start Address****MOVE_DESTINATION_SEG_START_2 = MCI1_5**

EXT_MEM = 0	0	0	0	BYT_5	BYT_4	BYT_3	BYT_2
EXT_MEM = 1		CHP_10	CHP_9	CHP_8	CHP_7	CHP_6	CHP_5

MOVE_DESTINATION_SEG_START_1 = MCI1_4

BYT_1	BYT_0	BLK_2	BLK_1	BLK_0	ROW_4	ROW_3	ROW_2
CHP_4	CHP_3	CHP_2	CHP_1	CHP_0			

MOVE_DESTINATION_SEG_START_0 = MCI1_3

ROW_1	ROW_0	COL_5	COL_4	COL_3	COL_2	COL_1	COL_0
-------	-------	-------	-------	-------	-------	-------	-------

EXT_MEM**Internal/external memory selection**

0: Values in the MOVE_DESTINATION_SEG_START registers are related to internal RAM memory.

BYT_5 thru BYT_0 Byte position of internal memory doubleword

One byte of a memory doubleword can be selected.

BLK_2 thru BLK_0 Block of internal memory $0 \leq \text{block address} \leq 7.$ **ROW_4 thru ROW_0 Internal memory row address** $0 \leq \text{row address} \leq 25.$ **COL_5 thru COL_0 Internal memory column address**For $0 \leq \text{row address} \leq 24$ is $0 \leq \text{column address} \leq 39.$ For row address = 25 is $0 \leq \text{column address} \leq 23.$

- 1: Values in the MOVE_DESTINATION_SEG_START registers are related to external RAM memory.
- CHP_10 thru CHP_0 External memory chapter address**
0 ≤ chapter address ≤ 2047.
- ROW_4 thru ROW_0 External memory row address**
0 ≤ row address ≤ 25.
- COL_5 thru COL_0 External memory column address**
For 0 ≤ row address ≤ 24 is 0 ≤ column address ≤ 39.
For row address = 25 is 0 ≤ column address ≤ 23.

Substitution Parameters

SUBSTITUTION_PATTERN_DISABLE = MCI1_1

SUBS_DIS_7	SUBS_DIS_6	SUBS_DIS_5	SUBS_DIS_4	SUBS_DIS_3	SUBS_DIS_2	SUBS_DIS_1	SUBS_DIS_0
------------	------------	------------	------------	------------	------------	------------	------------

SUBSTITUTION_PATTERN = MCI1_0

SUBS_7	SUBS_6	SUBS_5	SUBS_4	SUBS_3	SUBS_2	SUBS_1	SUBS_0
--------	--------	--------	--------	--------	--------	--------	--------

SUBSTITUTION_PATTERN_DISABLE
(SUBS_DIS_7 thru SUBS_DIS_0)

Bits SUBS_DIS_7 thru SUBS_DIS_0 enable the associated SUBSTITUTION_PATTERN bit for substitution. So a selection of bits to be substituted by the given SUBSTITUTION_PATTERN is possible.

0: Substitution is enabled.
1: Substitution is disabled.

SUBSTITUTION_PATTERN
(SUBS_7 thru SUBS_0)

Substitutes one byte which fulfills a match condition. Match conditions depend on the specific command and are explained there. Only bits which are enabled by the associated SUBSTITUTION_PATTERN_DISABLE bit will be substituted.

Find Parameters**FIND_RANGE_MAX = MCI2_4**

FIND_MAX_7	FIND_MAX_6	FIND_MAX_5	FIND_MAX_4	FIND_MAX_3	FIND_MAX_2	FIND_MAX_1	FIND_MAX_0
------------	------------	------------	------------	------------	------------	------------	------------

FIND_RANGE_MIN = MCI2_3

FIND_MIN_7	FIND_MIN_6	FIND_MIN_5	FIND_MIN_4	FIND_MIN_3	FIND_MIN_2	FIND_MIN_1	FIND_MIN_0
------------	------------	------------	------------	------------	------------	------------	------------

or

FIND_PATTERN_DONT_CARE = MCI2_4

FIND_DNC_7	FIND_DNC_6	FIND_DNC_5	FIND_DNC_4	FIND_DNC_3	FIND_DNC_2	FIND_DNC_1	FIND_DNC_0
------------	------------	------------	------------	------------	------------	------------	------------

FIND_PATTERN = MCI2_3

FIND_PATT_7	FIND_PATT_6	FIND_PATT_5	FIND_PATT_4	FIND_PATT_3	FIND_PATT_2	FIND_PATT_1	FIND_PATT_0
-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------

FIND_RANGE_MIN
(FIND_MIN_7 thru FIND_MAX_0)

Defines the minimum value of the range of reference values for the find option.

$0 \leq \text{FIND_MIN_7 thru FIND_MIN_0} \leq 255$.

$\text{FIND_MIN_7 thru FIND_MIN_0} \leq \text{FIND_MAX_7 thru FIND_MAX_0}$.

FIND_RANGE_MAX
(FIND_MAX_7 thru FIND_MAX_0)

Defines the maximum value of the range of reference values for the find option.

$0 \leq \text{FIND_MAX_7 thru FIND_MAX_0} \leq 255$.

$\text{FIND_MIN_7 thru FIND_MIN_0} \leq \text{FIND_MAX_7 thru FIND_MAX_0}$.

FIND_PATTERN_DONT_CARE
(FIND_DNC_7 thru FIND_DNC_0)

This parameter is a mask which qualifies the FIND_PATTERN bits FIND_PATT_7 thru FIND_PATT_0 as do care or don't care.

A 1 of a mask bit means don't care of the corresponding bit in FIND_PATTERN.

A 0 of a mask bit means do care of the corresponding bit in FIND_PATTERN.

FIND_PATTERN
(FIND_PATT_7 thru FIND_PATT_0)

Defines the reference value for the find option.

$0 \leq \text{FIND_PATT_7 thru FIND_PATT_0} \leq 255$.

FIND_CONTROL = MCI3_3

WINDOWS	BINARY	FIND_STOP	MOV_DIS	SUB_EN	FIND_RANGE	FIND_BACK	FIND_EN
---------	--------	-----------	---------	--------	------------	-----------	---------

Find enable FIND_EN

FIND_EN = MCI3_3_0.

1: The find capability is enabled.

	<p>If the move capability is enabled by the MOV_DIS bit, only found bytes will be moved.</p> <p>If the substitution capability is enabled by the SUB_EN bit, only found bytes will be substituted.</p> <p>0: The find capability is disabled.</p>
Find backward FIND_BACK	<p>FIND_BACK = MCI3_3_1</p> <p>The FIND_BACK bit defines the find direction.</p> <p>1: The find direction goes from higher to lower addresses.</p> <p>0: The find direction goes from lower to higher addresses.</p>
Find method FIND_RANGE	<p>FIND_RANGE = MCI3_3_2</p> <p>1: Find reference value is the value range given in FIND_RANGE_MIN and FIND_RANGE_MAX.</p> <p>0: Find reference value is the pattern given in FIND_PATTERN and FIND_PATTERN_DONT_CARE.</p>
Find stop FIND_STOP	<p>FIND_STOP = MCI3_3_5</p> <p>1: Find function is executed until the first occurrence of the match condition.</p> <p>0: Find function is executed until the whole memory segment is processed.</p>
Substitution enable SUB_EN	<p>SUB_EN = MCI3_3_3</p> <p>1: All bytes of the selected memory segment are substituted with the reference value of the SUBSTITUTE_PATTERN input parameter if FIND_EN = 0.</p> <p>Only found bytes are substituted if FIND_EN = 1.</p> <p>0: Substitution is disabled.</p>
Move disable MOV_DIS	<p>MOV_DIS = MCI3_3_4</p> <p>1: Moving any byte of the selected byte segment is disabled. The find capability is not influenced.</p> <p>0: Each byte is moved to the destination address.</p>
Binary	<p>1: The binary address scheme is assumed. The row/column bits ROW_4 thru ROW_0 and COL_5 thru COL_1 of MOVE_SOURCE_SEG_START and MOVE_SOURCE_SEG_END are interpreted as binary address bits. COL_0 has don't care value.</p> <p>0: The row/column address scheme is assumed.</p>
Windows	<p>1: The address given in MOVE_SOURCE_SEG_START is interpreted as the top left corner of a window and the address in MOVE_SOURCE_SEG_END the bottom right corner. The column and row address in MOVE_SOURCE_SEG_END must be bigger than the column and row address in MOVE_SOURCE_SEG_START.</p> <p>0: Contiguous row/column address scheme is assumed.</p>

Return Values

Address of Found Byte: MOV_SOURCE_SEG_START

If the find function is enabled, the address of the found byte is given in the MOVE_SOURCE_SEG_START parameter.

Address of Actual Destination: MOV_DESTINATION_SEG_START

If the find function is enabled, the destination address after found data is given in the MOV_DESTINATION_SEG_START parameter.

The value of the found byte is returned in MCI2_0.

FIND_STATUS = MCI3_1

x	x	x	SEG_END	FOUND	x	x	CMD_RUN
---	---	---	---------	-------	---	---	---------

Found Match Condition Flag: FOUND FOUND = MCI3_1_3.

- 1: A found match condition occurred.
- 0: No found match condition occurred.

Segment End Flag: SEG_END

SEG_END = MCI3_1_4.

- 1: The memory segment end is reached.
- 0: The memory segment end is not yet reached.

Comments

1. In internal or external memory a byte segment can be 1 to 1024 bytes and must be inside a chapter.
2. Source segment size must fit into the destination chapter.
3. If the source memory is internal memory, the following segments can be selected:
 - Destination is internal memory:
Selection of one byte segment by the BYT_5 to BYT_0 bits is possible. The segment length is determined by the MOV_SOURCE_SEG_START and MOV_SOURCE_SEG_END parameters. The destination byte segment can be at any byte position of a doubleword.
 - Destination is external memory:
Selection of only one byte segment by the CHP_10 to CHP_0 bits is possible. The segment length is determined by the MOV_SOURCE_SEG_START and MOV_SOURCE_SEG_END parameters.
4. If the source memory is external memory, one byte segment can be selected:
 - Destination is external memory:
The segment length is determined by the MOV_SOURCE_SEG_START and MOV_SOURCE_SEG_END parameters.
 - Destination is internal memory:
Selection of one byte segment by the BYT_5 to BYT_0 bits is possible. The segment length is determined by the MOV_SOURCE_SEG_START and MOV_SOURCE_SEG_END parameters. The destination byte segment can be at any byte position of a doubleword.

5. Minimum source segment start address of a chapter is row/column = 0/0.
 Maximum source segment start address of a chapter is row/column = 25/23.
 Minimum source segment end address of a chapter is row/column = 0/0.
 Maximum source segment end address of a chapter is row/column = 25/23.
 Minimum destination segment start address of a chapter is row/column = 0/0.
 Maximum destination segment start address of a chapter is row/column = 25/23.
6. Find function:
 After a byte is found, the command execution stops if FIND_STOP = 1. The address of the found byte is returned in the MOV_SOURCE_SEG_START parameter.
 If the move capability is enabled, the found byte will be moved to the destination address before command execution stops.
 If the substitution capability is enabled, the found byte will be substituted before moving.
7. Find methods:
 To find a byte with a certain value, two methods can be used:
 Find a byte which matches a range of reference values given in FIND_RANGE_MIN and FIND_RANGE_MAX.
 Find a byte which matches a reference value given in FIND_PATTERN. The reference value can be qualified with a don't care mask given by DONT_CARE_FIND_PATTERN.
8. Find direction:
 If FIND_BACK is 1 and the MOVE_MEMORY_SEG command is given, the find function starts on the address given in the MOV_SOURCE_SEG_START parameter. The find function stops at address row/column = 00/00. If FIND_BACK is 0 and the MOVE_MEMORY_SEG command is given, the find function starts on the address given in the MOV_SOURCE_SEG_START parameter. The find function stops at the address given in MOV_SOURCE_SEG_END parameter.
9. The find back bit must be set to 1 only if FIND_EN = 1.
10. The WINDOWS and BINARY bits must not be set to 1 at the same time.

1.5.9 NO_OPERATION

This command evokes an MCI bus interrupt service request without executing any functions.

Input Parameters

None.

Return Values

None.

Comment

This command only consumes PU time.

1.5.10READ_CLEAR_PAGE_TRACE

PAGE_TRACE and PAGE_REQUEST_STATUS_TABLE are combined in the same chapter. The start address of this chapter is given in M3L register PT_ADR. This chapter is divided into the following four different chapter segments with binary address organization:

- Address 0 – 255: In this chapter segment all transmitted basic pages are marked.
- Address 256 – 511: In this chapter segment all transmitted basic pages with subpages are marked.
- Address 512 – 767: In this chapter segment all pages are marked whose request is in progress.
- Address 768 –1023: In this chapter segment all pages whose request in a given group is intended are marked.

Beginning at a start page number this command searches the next (previous) page number which fits a given SEARCH_MODE and returns the page number with the page trace or page request status information. Dependent on the WRITE_READ and SET_RESET bits the corresponding bits in the selected memory segments are set or reset.

It is also possible to clear the complete PAGE_TRACE memory and the PAGE_REQUEST_STATUS_TABLE.

Input Parameters

Search Start Page Number

This is the page number where the search is being started from if ALL is set to 0.

PAGE_NUMBER_1 = MC10_4

0	0	0	0	0	M2	M1	M0
---	---	---	---	---	----	----	----

PAGE_NUMBER_0 = MC10_3

PT3	PT2	PT1	PT0	PU3	PU2	PU1	PU0
-----	-----	-----	-----	-----	-----	-----	-----

All above listed bits have the same meaning as defined in the world system teletext specification.

Page_Search_Type**SEARCH_MODE = MCI0_1**

DIRECTION	ALL	SET_RESET	WRITE_READ	PRQ_TO_BE_INTENDED	PRQ_IN_PROGRESS	SUB_PAGE	PAGE
-----------	-----	-----------	------------	--------------------	-----------------	----------	------

PAGE

1: Search is stopped at the next page which is marked in the page trace.

0: Search is not influenced by the page trace.

SUB_PAGE

1: Search is stopped at the next page which is marked in the subpage trace.

0: Search is not influenced by the subpage trace.

- PRQ_IN_PROGRESS** 1: Search is stopped at the next page which is marked in the PAGE_REQUEST_STATUS_TABLE as a page whose request is in progress.
0: Search is not influenced by the PAGE_REQUEST_STATUS_TABLE.
- PRQ_TO_BE_INTENDED** 1: Search is stopped at the next page whose request is intended.
0: Search is not influenced by the PAGE_REQUEST_STATUS_TABLE.
- WRITE_READ** 1: SET_RESET is active.
0: No bit will be overwritten, but the found search type will be returned to the output parameter register.
- SET_RESET** 1: If WRITE_READ = 1, all selected bits are set to 1.
0: If WRITE_READ = 1, all selected bits are set to 0.

ALL

- 1: All bits of the selected chapter segment (PAGE, SUB_PAGE, PRQ_IN_PROGRESS or PRQ_TO_BE_INTENDED) are selected.
0: Only one bit of a given memory segment (PAGE, SUB_PAGE, PRQ_IN_PROGRESS or PRQ_TO_BE_INTENDED) is selected. This bit represents the status of a page number specified in PAGE_NUMBER.

DIRECTION

Determines the search direction if ALL is set to 0.

- 1: Forward. Search is made to higher page numbers related to the given page number.
0: Backward. Search is made to lower page numbers related to the given page number.

Return Values

Page Number of Found Page

This is the page number of the found page. Depending on the search direction and the OVERFLOW bit it is > or < than the start page number.

PAGE_NUMBER_1 = MCIO_4

0	0	0	0	0	M2	M1	M0
---	---	---	---	---	----	----	----

PAGE_NUMBER_0 = MCIO_3

PT3	PT2	PT1	PT0	PU3	PU2	PU1	PU0
-----	-----	-----	-----	-----	-----	-----	-----

All above listed bits have the same meaning as defined in the world system teletext specification.

Page Request Status**PAGE_REQUEST_STATUS = MCIO_0**

OVERFLOW	0	0	0	PRQ_TO_BE_INTENDED	PRQ_IN_PROGRESS	SUB_PAGE	PAGE
----------	---	---	---	--------------------	-----------------	----------	------

PAGE

- 1: The found page was transmitted by broadcast.
- 0: The found page was not transmitted by broadcast.

SUB_PAGE

- 1: The broadcast transmitted a subpage for the found page number.
- 0: The broadcast did not transmit a subpage for the found page number.

PRQ_IN_PROGRESS

- 1: A page request for the found basic page is already executed.
- 0: A page request for the found basic page is not executed.

PRQ_TO_BE_INTENDED

- 1: A page request for the found basic page should be executed.
- 0: A page request for the found basic page should not be executed.

OVERFLOW

- 1: Search was started again at the beginning (end) of the table. The found page number is < (>) than the start page number.
- 0: Search was stopped at a higher (lower) page number before the end (beginning) of the table was reached.

Error Code

ERR_7 – ERR_0	Description
0	Command execution was successful.
1	PAGE_REQUEST_STATUS_TABLE is not enabled.
2 – 255	Not defined.

Comments

- Before using the PAGE_TRACE or the PAGE_REQUEST_STATUS_TABLE, a chapter must be allocated in the IAT/XAT register and the start address of this chapter must be given in PT_ADR.
- If only single bits in the PAGE_REQUEST_STATUS_TABLE should be set (independent of the page trace), use the command ADD_ALL_PAGES or direct M3L access.

1.5.17 READ_GROUP

This command reads a PRQ group record from the given PRQ group.

Input Parameters

PRQ Group Number

The PRQ group number selects the group record to be used.

PRQ_GROUP = MCI0_5

GR_UNUSE	0	0	0	0	PRQ_GROUP_2	PRQ_GROUP_1	PRQ_GROUP_0
----------	---	---	---	---	-------------	-------------	-------------

PRQ_GROUP_2 thru PRQ_GROUP_0 PRQ group number bits, $0 \leq \text{PRQ_GROUP} \leq 7$.

GR_UNUSE

Group control bit to change a used group to unused.

1: The group record is read and signed as unused.

0: The group record is only read.

Return Values

Search Type Number

The search type number determines the format of the PRQ table.

PRQ_TYPE = MCI0_5

GR_USED	0	0	0	0	PRQ_TYPE_2	PRQ_TYPE_1	PRQ_TYPE_0
---------	---	---	---	---	------------	------------	------------

PRQ_TYPE_1 – PRO_TYPE 0

Page request type number bits, $0 \leq \text{PRQ_TYPE} \leq 3$.

GR_USED

For details of this bit see MCI command WRITE_GROUP.

Number of Allocated Pages or Packets

GROUPX_NA_2 = MCI0_4

NA15	NA14	NA13	NA12	NA11	NA10	NA9	NA8
------	------	------	------	------	------	-----	-----

GROUPX_NA_1=MCI0_3

NA7	NA6	NA5	NA4	NA3	NA2	NA1	NA0
-----	-----	-----	-----	-----	-----	-----	-----

NA15 thru NA0

For search types 0 and 2 they state the number of allocated pages,

for search type 3 they state the number of allocated packets, $0 \leq \text{NA15 thru NA0} \leq 65535$.

GROUPX_FAP Pointer

This is the pointer to the first allocated element of the groups chain.

GROUPX_FAP_2 = MCI0_2

EXT_MEM = 0	0	0	0	BYT_5	BYT_4	BYT_3	BYT_2
EXT_MEM = 1		CHP_10	CHP_9	CHP_8	CHP_7	CHP_6	CHP_5

GROUPX_FAP_1 = MCI0_1

BYT_1	BYT_0	BLK_2	BLK_1	BLK_0	ROW4	ROW3	ROW2
CHP_4	CHP_3	CHP_2	CHP_1	CHP_0	ROW4	ROW3	ROW2

GROUPX_FAP_0 = MCI0_0

ROW1	ROW0	0	0	0	0	0	NIL
ROW1	ROW0						NIL

NIL 1: Pointer value is invalid. There is no succeeding page or packet in the chain.
0: Pointer shows to the first allocated page or packet.

PRQ Do Care Bits

GROUP_DO_CARE_4 = MCI1_5

type 0	x	x	x	x	x	x	x	x
type 2	0	0	0	0	0	M2	M1	M0
type 3	ROW4	ROW3	ROW2	ROW1	ROW0	M2	M1	M0

GROUP_DO_CARE_3 = MCI1_4

type 0	x	x	x	x	x	x	x	x
type 2	PT3	PT2	PT1	PT0	PU3	PU2	PU1	PU0
type 3	0	0	0	0	DES3	DES2	DES1	DES0

GROUP_DO_CARE_2 = MCI1_3

type 0	x	x	x	x	x	x	x	x
type 2	C4	MT2	MT1	MT0	MU3	MU2	MU1	MU0
type 3	x	x	x	x	x	x	x	x

GROUP_DO_CARE_1 = MCI1_2

type 0	x	x	x	x	x	x	x	x
type 2	C6	C5	HT1	HT0	HU3	HU2	HU1	HU0
type 3	x	x	x	x	x	x	x	x

GROUP_DO_CARE_0 = MCI1_1

type 0	x	x	x	x	x	x	x	x
type 2	C14	C13	C12	C11	C10	C9	C8	C7
type 3	x	x	x	x	x	x	x	x

For details of these bits see MCI command WRITE_GROUP.

Error Code

ERR_7 – ERR_0	Description
0	Command execution was successful.
1 – 255	Not defined.

Comment

If the group records are to be newly organized with the WRITE_GROUP command, they must be signed as unused by the READ_GROUP command input parameter GR_UNUSE.

1.5.18READ_PSEUDO_PACKET_TRACE

This command returns the pseudo packet trace bits from the PSEUDO_PACKET_TRACE register. A clear PSEUDO_PACKET_TRACE register option is available.

Input Parameters

Pseudo Packet Trace Control

PP_TRACE_CONTROL = MCIO_0

0	0	0	0	0	0	0	PP_TR_CLR
---	---	---	---	---	---	---	-----------

PP_TR_CLR Controls clearing of the pseudo packet trace memory.
1: After reading the pseudo packet trace memory it will be cleared to 0.
0: The pseudo packet trace memory will only be read.

Return Values

Pseudo Packet Trace

Acquisition traces all pseudo packet numbers and indicates them with the associated PPT_X bit in register PSEUDO_PACKET_TRACE.

PSEUDO_PACKET_TRACE = MCIO_1

D7	PPT_X_31	PPT_X_30	PPT_X_29	PPT_X_28	PPT_X_27	PPT_X_26	PPT_X_25
----	----------	----------	----------	----------	----------	----------	----------

PPT_X_31 thru PPT_X_25 Each bit is associated with the pseudo packet number given in its symbol.
1: Pseudo packet received.
0: Pseudo packet not received.

D7 Reserved. D7 is undefined.

Comment

None.

1.5.11 REMOVE_PAGE

This command removes a page with all related subpage and packet chains from PRQ groups of type 0 and 2. The page must be named by its PRQ record.

Input Parameters

PRQ Group Number

The PRQ group number selects the group record to be used.

PRQ_GROUP = MCIO_1

REM_PROT	REQUEST_STAT	SUBPAGE_1	SUBPAGE_0	0	PRQ_GROUP_2	PRQ_GROUP_1	PRQ_GROUP_0
----------	--------------	-----------	-----------	---	-------------	-------------	-------------

PRQ_GROUP_2 – PRQ_GROUP_0 PRQ group number bits, $0 \leq \text{PRQ_GROUP} \leq 7$.

The SUBPAGE_1 – 0 bits are only considered if a group with search type 0 is selected. The following table explains their meaning:

SUBPAGE_1, SUBPAGE_0	Comment
00	Only the page given by magazine number, page number and subpage code is removed.
01	The subpage chain related to the basic page number is completely removed.
10	The subpage chain related to the basic page number is removed without the first page of the chain (and control bits).
11	Not defined.

REM_PROT: 1 Pages protected by the bit REM_PROT (this bit can be set with the other ACQ control bits) are also removed.

REM_PROT: 0 Pages are only removed if they are not remove protected.

REQUEST_STAT: 1 The (page request in progress) bit in the PAGE_REQUEST_STATUS_TABLE corresponding to the basic page number to be removed is removed too. The start address of the PAGE_REQUEST_STATUS_TABLE could be found in M3L registers R123 – R125 (PT_ADR). The REQUEST_STATUS_TABLE is located under the same chapter address as the page trace but with an offset of 512 (binary).

REQUEST_STAT: 0 The REQUEST_STATUS_TABLE is not influenced.

PRQ Record

The PRQ record determines the PRQ bits which will be searched for.

PRQ_RECORD_4 = MCI1_5

0	0	0	0	0	M2	M1	M0
---	---	---	---	---	----	----	----

PRQ_RECORD_3 = MCI1_4

PT3	PT2	PT1	PT0	PU3	PU2	PU1	PU0
-----	-----	-----	-----	-----	-----	-----	-----

PRQ_RECORD_2 = MCI1_3

C4	MT2	MT1	MT0	MU3	MU2	MU1	MU0
----	-----	-----	-----	-----	-----	-----	-----

PRQ_RECORD_1 = MCI1_2

C6	C5	HT1	HT0	HU3	HU2	HU1	HU0
----	----	-----	-----	-----	-----	-----	-----

PRQ_RECORD_0 = MCI1_1

C14	C13	C12	C11	C10	C9	C8	C7
-----	-----	-----	-----	-----	----	----	----

All above listed request bits have the same meaning as defined in the world system teletext specification. If search type 0 is chosen, bits C14-C4 have to be set to 0.

Return Values

Actual number of elements in the free chap chain in M3L-Bus register NF_CHAP,
actual number of elements in the free p40 chain in M3L-Bus register NF_P40.

Actual number of allocated elements of the chap chain in the associated group record GROUPX_NA.

Only for removing the last element:

Pointer to the first element of the chap chain in the associated group record GROUPX_FAP will be set to zero. For type 0 these are the pointers in the PAGE_TLU (page table lookup).

Error Code

ERR_7 – ERR_0	Description
0	Command execution was successful.
1	No page with the given PRQ record found.
2	Type of group given is not supported.
3	Page is not removed because of remove protection (see ACQ control bits). This error code is only relevant if SUBPAGE_1-0 = 00.
5	Page trace not enabled.
4 – 255	Not defined.

Comments

1. For search type 0 the PRQ_RECORD magazine, page and subpage bits are relevant. Qualification of these bits with the PRQ record do care mask will be ignored. For search type 2 the PRQ record do care bits of the related group are relevant.
2. If the same page was requested in different groups, the page request in progress bit will also be reset if only one of these multiple requested pages is removed (see also ADD_PAGE).

1.5.20REQUEST_ALL_PAGES

This command compares the list of pages whose request is in progress with the list of the pages which are intended to be requested. All pages whose request is in progress and not intended are removed. All pages whose request is intended and not in progress will be requested.

Input Parameters

ACQ Control Bits

Data acquisition control bits. For details of bit see document “ACQ Reference”.

ACQ_CONTROL_1 = MCIO_5

DCR	HEN	0	IREQ	PBR	PBLF	FIRST	0
-----	-----	---	------	-----	------	-------	---

ACQ_CONTROL_0 = MCIO_4

DIS_PP	P_OV_OC = 0	OV_OCCU = 0	SM_1	SM_0	REM_PROT	CHECK_1	CHECK_0
--------	-------------	-------------	------	------	----------	---------	---------

These ACQ_CONTROL registers are copied to each requested basic page.

Maximum Number of Subpages

Maximum number of subpages which are linked to the basic page if search type 0 is chosen. For any other type these bits are ignored. For details of bit see document “ACQ Reference”.

NMAX_SUB = MCIO_3

S-MAX_7	S-MAX_6	S-MAX_5	S-MAX_4	S-MAX_3	S-MAX_2	S-MAX_1	S-MAX_0
---------	---------	---------	---------	---------	---------	---------	---------

$0 \leq S_MAX \leq 255$

If S_MAX = 255 then the subpage collect mode is enabled.

PRQ Group Number

The PRQ group number selects the group record to be used.

PRQ_GROUP = MCIO_1

REM_PROT	0	0	0	0	PRQ_GROUP_2	PRQ_GROUP_1	PRQ_GROUP_0
----------	---	---	---	---	-------------	-------------	-------------

PRQ_GROUP_2 thru PRQ_GROUP_0

PRQ group number bits, $0 \leq PRQ_GROUP \leq 7$

REM_PROT

- 1: Pages protected by the bit REM_PROT (this bit can be set with the other ACQ control bits) are also removed.
- 0: Pages are only removed if they are not remove protected.

Return Values

Actual number of elements in the free chap chain in M3L-Bus registers NF_CHAP.

Actual number of allocated elements of the chap chain in the associated group record GROUPX_NA.

Pointer to the first element of the chap chain in the associated group record GROUPX_FAP. For type 0 these are the pointers in the PAGE_TLU (page table lookup).

Error Code

ERR_7 – ERR_0	Description
0	Command execution was successful.
1	No chap element available in free chap chain.
2 – 255	Not defined.

Comments

1. Before using this command, ensure that a PRQ group record with the given search type has been declared. Use WRITE_GROUP for this command.
2. Ensure that the number of basic pages to be requested fits the available memory of the free chap chain.
3. Ensure that the page trace is enabled before using this command.
4. Notice that the PAGE_REQUEST_STATUS_TABLE represents pages in any group, but the remove or request of pages is only done in the group specified by PRQ_GROUP. So pages which are requested in another group will not be removed.

1.5.21 RESET_INTQ

This command resets the interrupt signal at the INTQ pin to High and resets INTQ_BIT to Low. Then it returns the INTQ interrupt request source flags.

Input Parameters

None.

Return Values**Interrupt Signal: INTQ**

Interrupt signal at INTQ pin is set to High.

Interrupt Request Bit: INTQ_BIT

MEGATEXT interrupt request bit INTQ_BIT is set to Low.

Interrupt Request Source Flags

Interrupt request source flags are used to determine the origin of the interrupt request. The bit mapping of the interrupt request source registers is as follows:

IRQS_5 = MCIO_5

IRQSB_47	IRQSB_46	IRQSB_45	IRQSB_44	IRQSB_43	IRQSB_42	IRQSB_41	IRQSB_40
----------	----------	----------	----------	----------	----------	----------	----------

IRQS_4 = MCIO_4

IRQSB_39	IRQSB_38	IRQSB_37	IRQSB_36	IRQSB_35	IRQSB_34	IRQSB_33	IRQSB_32
----------	----------	----------	----------	----------	----------	----------	----------

IRQS_3 = MCIO_3

IRQSB_31	IRQSB_30	IRQSB_29	IRQSB_28	IRQSB_27	IRQSB_26	IRQSB_25	IRQSB_24
----------	----------	----------	----------	----------	----------	----------	----------

IRQS_2 = MCIO_2

IRQSB_23	IRQSB_22	IRQSB_21	IRQSB_20	IRQSB_19	IRQSB_18	IRQSB_17	IRQSB_16
----------	----------	----------	----------	----------	----------	----------	----------

IRQS_1 = MCIO_1

IRQSB_15	IRQSB_14	IRQSB_13	IRQSB_12	IRQSB_11	IRQSB_10	IRQSB_9	IRQSB_8
----------	----------	----------	----------	----------	----------	---------	---------

IRQS_0 = MCIO_0

IRQSB_7	IRQSB_6	IRQSB_5	IRQSB_4	IRQSB_3	IRQSB_2	IRQSB_1	IRQSB_0
---------	---------	---------	---------	---------	---------	---------	---------

The states of the interrupt request source flags are as follows:

1: Indicates the interrupt source which activates the INTQ signal.

0: Indicates that this source does not make a request.

IRQSB	Interrupt Request Source	Related MCI Commands
0	MCI command execution terminated. This is signaled by the CMD_RUN bit in MCI register MCI3_1 too.	
1	End of page: given if any EOP criterion was found by the ACQ and IREQ control bit is set.	
2	Pseudo packet overflow. Memory allocation failed during pseudo packet acquisition. There is no memory space to store pseudo packets.	GET_GET_FREE_CHAP
3	Subpage buffer overflow. Memory allocation failed during subpage acquisition in buffered subpage collect mode.	
4	Chap chain overflow. Memory allocation failed during page acquisition. There is no memory space to store pages.	REMOVE_PAGE
5	Non-page-related packet detected. Will be set by ACQ if non-page-related packet (X/29 ... X/31) was found the first time after the packet being looked for bit (PCKBLF) was set.	
6	Subpage buffer overflow. Will be set by ACQ if reserved subpage buffer is exhausted.	
7 – 47	Reserved.	

Comment

INTQ_BIT is located in MCI register MCI3_1 bit MCI3_1_1. It allows the external controller polling of MEGATEXT requests instead of using the INTQ interrupt request.

1.5.12 SEARCH_PACKET_25_28

This command searches for TTX packets with packet numbers from 25 to 28 in packet chains of pages from PRQ groups of search type 2.

Input Parameters**PRQ Group Number**

The PRQ group number selects the group record to be used.

PRQ_GROUP = MCI0_1

0	0	0	0	0	PRQ_GROUP_2	PRQ_GROUP_1	PRQ_GROUP_0
---	---	---	---	---	-------------	-------------	-------------

PRQ_GROUP_2 thru PRQ_GROUP_0 PRQ group number bits, $0 \leq \text{PRQ_GROUP} \leq 7$.

PRQ Record

The PRQ record determines the PRQ bits to be searched for.

PRQ_RECORD_4 = MCI1_5

ROW4	ROW3	ROW2	ROW1	ROW0	M2	M1	M0
------	------	------	------	------	----	----	----

PRQ_RECORD_3 = MCI1_4

PT3	PT2	PT1	PT0	PU3	PU2	PU1	PU0
-----	-----	-----	-----	-----	-----	-----	-----

PRQ_RECORD_2 = MCI1_3

C4	MT2	MT1	MT0	MU3	MU2	MU1	MU0
----	-----	-----	-----	-----	-----	-----	-----

PRQ_RECORD_1 = MCI1_2

C6	C5	HT1	HT0	HU3	HU2	HU1	HU0
----	----	-----	-----	-----	-----	-----	-----

PRQ_RECORD_0 = MCI1_1

C14	C13	C12	C11	C10	C9	C8	C7
-----	-----	-----	-----	-----	----	----	----

DESIGNATION = MCI1_0

0	0	0	0	DES3	DES2	DES1	DES0
---	---	---	---	------	------	------	------

All above listed request bits have the same meaning as defined in the world system teletext specification.

DES3 thru DES0

Designation code bits of packets X/26 thru X/28.

These bits are only relevant for $26 \leq \text{ROW4 thru ROW0} \leq 28$.

PRQ Record Do Care Mask

The PRQ record do care mask determines the PRQ bits (row and designation) which will be qualified with do care in the PRQ_RECORD registers.

PRQ_RECORD_DO_CARE_4 = MCI2_5

ROW4	ROW3	ROW2	ROW1	ROW0	0	0	0
------	------	------	------	------	---	---	---

DESIGNATION = MCI2_0

0	0	0	0	DES3	DES2	DES1	DES0
---	---	---	---	------	------	------	------

Each bit masks the related bits of the PRQ_RECORD registers.

1: Do care of the corresponding PRQ_RECORD bit during search.

0: Don't care of the corresponding PRQ_RECORD bit during search.

Return Values

Address Pointer to Found Packet

This is the pointer to the first found packet if only the designation code does not match.

This is the pointer to the found packet if both the packet and designation code match.

AP_CHAP_2 = MCI2_5

EXT_MEM = 0	0	0	0	BYT_5	BYT_4	BYT_3	BYT_2
EXT_MEM = 1		CHP_10	CHP_9	CHP_8	CHP_7	CHP_6	CHP_5

AP_CHAP_1 = MCI2_4

BYT_1	BYT_0	BLK_2	BLK_1	BLK_0	ROW5	ROW4	ROW3
CHP_4	CHP_3	CHP_2	CHP_1	CHP_0			

AP_CHAP_0 = MCI2_3

ROW2	ROW1	ROW0	0	0	0	0	0
------	------	------	---	---	---	---	---

Detected Designation Codes

The search is continued until all packets which match with the given packet are found. The scanned designation codes of these packets are returned in the following registers.

SCANNED_DESIGNATION_1 = MCI2_1

DC_15	DC_14	DC_13	DC_12	DC_11	DC_10	DC_9	DC_8
-------	-------	-------	-------	-------	-------	------	------

SCANNED_DESIGNATION_0 = MCI2_0

DC_7	DC_6	DC_5	DC_4	DC_3	DC_2	DC_1	DC_0
------	------	------	------	------	------	------	------

DC_15 thru DC_0 Each bit represents one designation code. The following table shows the relationship:

Bit	Designation Code
DC_0	0
DC_1	1
—	—
DC_14	14
DC_15	15

Error Code

ERR_7 – ERR_0	Description
0	Packet found.
1	No page with the given PRQ record found.
2	Given group contains unsupported type (e.g. type 3).
3	No packet with the given PRQ record found.
4 – 255	Not defined.

Comment

For search type 0 distinction between basic and subpages is made with the subpage bits. If the subpage bits are set to 0, the basic page will be addressed.

1.5.13 SEARCH_PACKET_29_30

This command searches for a given packet request record for packets 29 and 30 in chains from PRQ groups of type 3.

Input Parameters**PRQ Group Number**

The PRQ group number selects the group record to be used.

PRQ_GROUP = MCI0_1

0	0	0	0	0	PRQ_GROUP_2	PRQ_GROUP_1	PRQ_GROUP_0
---	---	---	---	---	-------------	-------------	-------------

PRQ_GROUP_2 thru PRQ_GROUP_0 PRQ group number bits, $0 \leq \text{PRQ_GROUP} \leq 7$.

PRQ Record

The PRQ record determines the PRQ bits to be searched for.

PRQ_RECORD_4 = MCI1_5

ROW4	ROW3	ROW2	ROW1	ROW0	M2	M1	M0
------	------	------	------	------	----	----	----

PRQ_RECORD_3 = MCI1_4

0	0	0	0	DES3	DES2	DES1	DES0
---	---	---	---	------	------	------	------

All above listed request bits have the same meaning as defined in the world system teletext specification.

DES3 thru DES0 Designation code bits of packets X/29 thru X/31.

Return Values**Address Pointer to Found Packet**

This is the pointer to the found packet.

AP_CHAP_2 = MCI2_5

EXT_MEM = 0	0	0	0	BYT_5	BYT_4	BYT_3	BYT_2
EXT_MEM = 1		CHP_10	CHP_9	CHP_8	CHP_7	CHP_6	CHP_5

AP_CHAP_1 = MCI2_4

BYT_1	BYT_0	BLK_2	BLK_1	BLK_0	ROW5	ROW4	ROW3
CHP_4	CHP_3	CHP_2	CHP_1	CHP_0			

AP_CHAP_0 = MCI2_3

ROW2	ROW1	ROW0	0	0	0	0	0
------	------	------	---	---	---	---	---

Received Packet Address and Designation Code

This is the row, magazine and designation code of the found packet

P_ADR_1 = MCI2_2

ROW4	ROW3	ROW2	ROW1	ROW0	M2	M1	M0
------	------	------	------	------	----	----	----

P_ADR_0 = MCI2_1

0	0	0	0	DES3	DES2	DES1	DES0
---	---	---	---	------	------	------	------

PACKET_CONTROL = MCI2_0

Packet control bits. For details of bit see document “ACQ Reference”

0	PCKHEN	0	PCKIREQ	0	PCKBLF	0	0
---	--------	---	---------	---	--------	---	---

Error Code

ERR_7 – ERR_0	Description
0	Packet found.
1	Given packet was not found.
2	Unsupported search type in given group or group is not used.
3 – 255	Not defined.

Comment

Search comparison is done with PRQ_RECORD and DO_CARE bits of given group record.

1.5.14 SEARCH_PAGE

This command searches for a given page in PRQ groups of type 0 and 2. Searching is done for the PRQ record which has been written by the ADD_PAGE command.

Input Parameters

PRQ Group Number

The PRQ group number selects the group record to be used.

PRQ_GROUP = MCI0_1

0	0	0	0	ALL_GROUPS	PRQ_GROUP_2	PRQ_GROUP_1	PRQ_GROUP_0
---	---	---	---	------------	-------------	-------------	-------------

PRQ_GROUP_2 thru PRQ_GROUP_0 PRQ group number bits, $0 \leq \text{PRQ_GROUP} \leq 7$.

ALL_GROUPS:1

Beginning from the group given by PRQ_GROUP_2 – 0 the given page number is searched in every group with the same or with a higher group number until it is found.

ALL_GROUPS:0

The page is searched only in the group given by PRQ_GROUP_2 – 0.

PRQ Record

The PRQ record determines the PRQ bits to be searched for.

PRQ_RECORD_4 = MCI1_5

0	0	0	0	0	M2	M1	M0
---	---	---	---	---	----	----	----

PRQ_RECORD_3 = MCI1_4

PT3	PT2	PT1	PT0	PU3	PU2	PU1	PU0
-----	-----	-----	-----	-----	-----	-----	-----

PRQ_RECORD_2 = MCI1_3

C4	MT2	MT1	MT0	MU3	MU2	MU1	MU0
----	-----	-----	-----	-----	-----	-----	-----

PRQ_RECORD_1 = MCI1_2

C6	C5	HT1	HT0	HU3	HU2	HU1	HU0
----	----	-----	-----	-----	-----	-----	-----

PRQ_RECORD_0 = MCI1_1

C14	C13	C12	C11	C10	C9	C8	C7
-----	-----	-----	-----	-----	----	----	----

All above listed request bits have the same meaning as defined in the world system teletext specification.

PRQ Record Do Care Mask:

The PRQ record do care mask determines the PRQ bits which will be qualified with do care in the PRQ_RECORD registers.

For this command the input parameters are implicitly given by the search type.

For search type 2 the PRQ record do care bits of the related group are relevant.

For search type 0 all do care bits of the bits are don't care.

If the subpage mode of the given basic page number is switched to subpage don't care mode (type 0), the subpage bits of the record will not be considered.

Return Values

Page Number

This is the page number of the found page.

If a group with type 0 is selected and the subcode which is searched for is not found, the next higher subpage code to the given basic page number will be returned.

PAGE_NUMBER_4 = MCI1_5

0	0	0	0	0	M2	M1	M0
---	---	---	---	---	----	----	----

PAGE_NUMBER_3 = MCI1_4

PT3	PT2	PT1	PT0	PU3	PU2	PU1	PU0
-----	-----	-----	-----	-----	-----	-----	-----

PAGE_NUMBER_2 = MCI1_3

C4	MT2	MT1	MT0	MU3	MU2	MU1	MU0
----	-----	-----	-----	-----	-----	-----	-----

PAGE_NUMBER_1 = MCI1_2

C6	C5	HT1	HT0	HU3	HU2	HU1	HU0
----	----	-----	-----	-----	-----	-----	-----

All above listed request bits have the same meaning as defined in the world system teletext specification.

Address Pointer to Found Page

This is the pointer to the found page.

AP_CHAP_2 = MCI3_5

EXT_MEM = 0	0	0	0	BYT_5	BYT_4	BYT_3	BYT_2
EXT_MEM = 1		CHP_10	CHP_9	CHP_8	CHP_7	CHP_6	CHP_5

AP_CHAP_1 = MCI3_4

BYT_1	BYT_0	BLK_2	BLK_1	BLK_0	0	0	0
CHP_4	CHP_3	CHP_2	CHP_1	CHP_0			

ACQ Control Bits

Data acquisition control bits. For details of bit see document “ACQ Reference”.

ACQ_CONTROL_1 = MCI3_3

DCR	HEN	0	IREQ	PBR	PBLF	FIRST	C_BTT
-----	-----	---	------	-----	------	-------	-------

ACQ_CONTROL_0 = MCI2_2

DIS_PP	P_OV_OC	OV_OCCU	SM_1	SM_0	REM_PROT	CHECK_1	CHECK_0
--------	---------	---------	------	------	----------	---------	---------

PRQ Group Number

This is the group number in which the search page was found.

PRQ_GROUP = MCI0_1

0	0	0	0	0	PRQ_GROUP_2	PRQ_GROUP_1	PRQ_GROUP_0
---	---	---	---	---	-------------	-------------	-------------

PRQ_GROUP_2 thru PRQ_GROUP_0 PRQ group number bits, $0 \leq \text{PRQ_GROUP} \leq 7$.

Error Code

ERR_7 – ERR_0	Description
0	Page with the given PRQ record found.
1	No page with the given PRQ record found.
2	Given group contains unsupported type or group is not used.
3	The returned page number and the corresponding address pointer is the page number (address pointer) of a requested page with the next highest subcode (only in groups of type 0).
4 – 255	Not defined.

Comments

2. The status of the found page can be checked with return parameter. ACQ_CONTROL_1 and ACQ_CONTROL_0.
1. For search type 2 the PRQ_RECORD input parameter bits are qualified with the do care bits of the given group record.

1.5.25SEARCH_TOP_PAGENUMBER

Beginning at a start page number this command searches the next (or actual or previous) page number (or block page number or group page number or subtitle page number or TV program schedule page number) in the basic TOP table.

It is always assumed that the basic TOP table is the first page of ACQ group 6.

Input Parameters

Search Start Page Number

This is the page number where the search is started from.

SEARCH_PAGE_NUM 1 = MCIO_4

0	0	0	0	0	M2	M1	M0
---	---	---	---	---	----	----	----

SEARCH_PAGE_NUM_0 = MCIO_3

PT3	PT2	PT1	PT0	PU3	PU2	PU1	PU0
-----	-----	-----	-----	-----	-----	-----	-----

All above listed bits have the same meaning as defined in the world system teletext specification.

TOP Page Search Direction

TOP_SEARCH_DIR = MCIO_2

0	0	0	0	0	0	TPD_1	TPD_0
---	---	---	---	---	---	-------	-------

TPD_1, TPD_0 TOP page search direction bits
These have the following meaning:

TPD_1, TD_0	TOP Page Search Direction
00	Backward. Search is made to lower page numbers related to the given page number.
01	Actual. The page type in the basic TOP table at the given page number will be returned in parameter PAGE_TYPE_FOUND.
10	Forward. Search is made to higher page numbers related to the given page number.

TOP Page Type Selection

This is the page type which will be searched for.

TOP_PAGE_TYPE_U = MCIO_1

0	0	0	0	TPU_3	TPU_2	TPU_1	TPU_0
---	---	---	---	-------	-------	-------	-------

TOP_PAGE_TYPE_L = MCIO_0

0	0	0	0	TPL_3	TPL_2	TPL_1	TPL_0
---	---	---	---	-------	-------	-------	-------

TPU_3 thru TPU_0 and TPL_3 thru TPL_0

TOP page type bits

They define the type of a page in the basic TOP table (block, group, subtitle, program review, normal page) and have the same meaning as described in the TOP specification of the IRT.
The type to be searched is smaller than or equal to the upper limit (TPU) and larger than or equal to the lower limit (TPL).

Return Values

Page Number of Found Page

This is the page number of the found page. It is > or < than the start page number.

PAGE_NUMBER_1 = MCIO_4

0	0	0	0	0	M2	M1	M0
---	---	---	---	---	----	----	----

PAGE_NUMBER_0 = MCIO_3

PT3	PT2	PT1	PT0	PU3	PU2	PU1	PU0
-----	-----	-----	-----	-----	-----	-----	-----

All above listed bits have the same meaning as defined in the world system teletext specification.

TOP Page Type of Found Page

This is the page type of the found page. See TOP specification of the IRT. If the search direction is backward the EOBT, LBL and LGR bits are not relevant.

PAGE_TYPE_FOUND = MCIO_5

EOBT	LBL	LGR	0	TPTF_3	TPTF_2	TPTF_1	TPTF_0
------	-----	-----	---	--------	--------	--------	--------

TPTF_3 thru TPTF_0**TOP page type of found page number****LGR****End of block reached**

The LGR bit will be set, if the returned page number is in the last group of the current block.

1: Last group reached.

0: Last group not reached.

LBL**Last block reached**

The LBL bit will be set, if the returned page number is in the last block of the BTT.

1: Last block reached.

0: Last block not reached.

EOBT**End of basic TOP table reached**

For forward search this is the end of the BTT.

1: End of basic TOP table reached.

0: End of basic TOP table not reached.

Error Code

ERR_7 – ERR_0	Description
0	Command execution was successful.
1	Basic TOP table or additional information table not received.
2	No page number found.
3 – 255	Not defined.

Comments

1. Before the BTT is received for the first time, all bytes of this chapter are automatically set to 20_H.
2. If the search direction is actual, the EOBT bit should always be interpreted as the end of the BTT.

1.5.26 SERIAL_PARALLEL_CONVERSION

This command switches on/off the serial/parallel conversion (S/P-C) and controls TTX page display representation like: single/double page display, status row position, subtitle/newsflash display, user-defineable character string byte selection, rolling header control and language character set selection.

Input Parameters

S/P-C Mode

SPC_MODE_1 = MCIO 5

0	MIX_CON	DIS_24	0	SPC_1	EN_B4	EN_B3	SPC_0
---	---------	--------	---	-------	-------	-------	-------

SPC_MODE_0 = MCIO_4

PCS_CHAR_SET	NAT_OPT_OFF	OUT_TR_ENA	ROLL_OFF	STAT_BOT	NEW_SUB	NAT_OPT_1	NAT_OPT_0
--------------	-------------	------------	----------	----------	---------	-----------	-----------

SPC_1, SPC_0 Controls S/P-C switch on/off and for different page modes as shown in the following table:

SPC_1, SPC_0	Description
00	S/P-C is switched off.
01	S/P-C is switched on and serial/parallel conversion will be continuous. Single page mode. Only the page which is selected by the basic display page registers will be converted.
10	S/P-C is switched on and serial/parallel conversion will be continuous. Single page mode. Only the page which is selected by the extended display page registers will be converted.
11	S/P-C is switched on and serial/parallel conversion will be continuous. Double page mode. Both pages which are selected by the basic and extended page registers will be converted.

EN_B3

The EN_ B3 bit commands the S/P-C to use byte 3 of either the basic or extended display memory for display features like X/26 support and user-defineable character string copy. The user-defineable character display words are strings of length 8 located in the internal RAM at addresses UDC0 and UDC1.

- 1: Byte 3 of basic and extended display memory is approved for the S/P-C.
- 0: Byte 3 of basic and extended display memory is locked for the S/P-C.

- EN_B4** The EN_B4 bit commands the S/P-C to use byte 4 of either the basic or extended display memory for display features like user-defineable character string copy. The user-defineable character display words are strings of length 8 located in the internal RAM at addresses UDC0 and UDC1.
1: Byte 4 of basic and extended display memory is approved for the S/P-C.
0: Byte 4 of basic and extended display memory is locked for the S/P-C.
- DIS_24** The DIS_24 bit commands the S/P-C to copy row 24 of the display page which is determined by the basic page registers to row 24 of the display memory or not. This is normally useful for showing the FLOF labels.
1: Copy of row 24 is disabled.
0: Copy of row 24 is enabled but the display row 24 flag of packet X/27 is considered as described in the WSTN.
- MIX_CON** The MIX_CON bit commands the S/P-C to set the transparent foreground bit (TRF) of the character display word each time the conceal bit in the display word is set. The MIX_CON bit must be set by the controller, if the concealed characters should be hidden in MIX mode.
1: TRF in the character display word (CDW) is set each time the conceal bit (in CDW) is set.
0: TRF in the character display word (CDW) is never set.
- NAT_OPT_1, NAT_OPT_0** National option character set.
These bits control the default national character sets addressed by the page header control bits C14, C13 and C12 as shown in the following table.

Control Bits	NAT_OPT_1, NAT_OPT_0			
C12, C13, C14	00	01	10	11
	Char. set 6	Char. set 38	Char. set 40	Char. set 55
000	English	Polish	English (US)	English
001	German	German	German	German
010	Swedish	Swedish	Swedish	Swedish
011	Italian	Italian	Italian	Italian
100	French	French	French	French
101	Portuguese	Serbocroat	Portuguese	Portuguese
110	Czechoslovak	Czechoslovak	Czechoslovak	Turkish
111	English	Rumanian	English	English

- NEW_SUB** The NEW_SUB bit commands the S/P-C to consider or not the page header control bits C5 (news flash) and C6 (sub-title).
1: The S/P-C considers the control bits C5 and C6. If one or both are set to 1, the box mask register 1 (BOXMR_1 of display control words) and box

display word 1 (BOXDW_1 of display control words) are used by the display generator. Otherwise BOXMR_0 and BOXDW_0 are used.

0: The S/P-C ignores the control bits C5 and C6. In this case BOXMR_0 and BOXDW_0 are used.

In each case the S/P-C converts serial box on and box off control information into parallel for the affected characters in the basic display memory by setting the BX bit in the CDWs.

STAT_BOT

0: The status row 24 of the actual displayed chapter will be written by the S/P-C to the basic display memory row 0/column 0. Normal text starts on row 1/column 0.

1: The status row of the displayed chapter will be written by the S/P-C to the basic display memory row 24/column 0. Normal text starts on row 0/column 0.

ROLL_OFF

The ROLL_OFF bit commands the S/P-C to display the page header or not.

1: The S/P-C does not copy the page header to the display memory.

0: The S/P-C copies the page header to the display memory. Because the S/P-C refreshes the page header continuously, the result is a “rolling header”.

OUT_TRA_ENA

The OUT_TRA_ENA bit commands the S/P-C to influence the outer screen display word or not.

1: The S/P-C considers the control bits C5 and C6. If one or both are set to 1, the transparent background and the transparent foreground bit in the outer screen display word are set to 1.

0: The S/P-C considers the control bits C5 and C6. If one or both are set to 1, the transparent background and the transparent foreground bit in the outer screen display word are set to 0.

NAT_OPT_OFF

The NAT_OPT_OFF bit commands the S/P-C to consider the language control bits C12, C13, C14 or not.

1: Translation of the national option characters is switched off.

0: Translation of the national option characters is enabled.

PCS_CHAR_SET

This bit enables the S/P-C to use user-defined PCS characters instead of hardwired ROM characters of MEGATEXT. The user can exchange the complete G0 set and the G2 set or parts of them. These parts are defined by registers G0_WINDOW_START, G0_WINDOW_END, G2_WINDOW_START and G2_WINDOW_END. The equivalent PCS character set (or a part of it) of G0 (G2) must be downloaded between PCS addresses 20_H – 7F_H (80_H – FF_H) before the S/P-C is started. The translation of national option characters which are inside the defined window is automatically switched off. Do not forget to set the EN_B4 bit, if PCS characters should be used by the S/P-C.

1: The PCS character set is enabled for the S/P-C.

0: The PCS character is disabled for the S/P-C. Only characters stored in the character ROM are used.

G0_WINDOW_START = MCIO_3

G0_WIN_S_7	G0_WIN_S_6	G0_WIN_S_5	G0_WIN_S_4	G0_WIN_S_3	G0_WIN_S_2	G0_WIN_S_1	G0_WIN_S_0
------------	------------	------------	------------	------------	------------	------------	------------

G0_WIN_S_7 thru G0_WIN_S_0 define the lowest address of the G0 set to be substituted by PCS characters. This value must be bigger than 20_H.

G0_WINDOW_END = MCIO_2

G0_WIN_E_7	G0_WIN_E_6	G0_WIN_E_5	G0_WIN_E_4	G0_WIN_E_3	G0_WIN_E_2	G0_WIN_E_1	G0_WIN_E_0
------------	------------	------------	------------	------------	------------	------------	------------

G0_WIN_E_7 thru G0_WIN_E_0 define the highest address of the G0 set to be substituted by PCS characters. This value must be smaller than 7F_H.

G2_WINDOW_START = MCIO_1

G2_WIN_S_7	G2_WIN_S_6	G2_WIN_S_5	G2_WIN_S_4	G2_WIN_S_3	G2_WIN_S_2	G2_WIN_S_1	G2_WIN_S_0
------------	------------	------------	------------	------------	------------	------------	------------

G2_WIN_S_7 thru G2_WIN_S_0 define the lowest address of the G2 set to be substituted by PCS characters. This value must be bigger than 20_H.

G2_WINDOW_END = MCIO_0

G2_WIN_E_7	G2_WIN_E_6	G2_WIN_E_5	G2_WIN_E_4	G2_WIN_E_3	G2_WIN_E_2	G2_WIN_E_1	G2_WIN_E_0
------------	------------	------------	------------	------------	------------	------------	------------

G2_WIN_E_7 thru G2_WIN_E_0 define the highest address of the G2 set to be substituted by PCS characters. This value must be smaller than 7F_H.

Display Page Parameters**Basic Display Page Registers**

These registers define the address pointer of the display page which will be converted into the basic display memory (internal RAM block 2). Use command SEARCH_PAGE to get the address pointer. Selection on which half of the screen the page appears can be done by bit BES of the page position word (PPW) display control register.

Address Pointer to Basic Display Page

This is the pointer to the page to be displayed.

AP_CHAP_2 = MCI1_5

EXT_MEM = 0	0	0	0	BYT_5	BYT_4	BYT_3	BYT_2
EXT_MEM = 1		CHP_10	CHP_9	CHP_8	CHP_7	CHP_6	CHP_5

AP_CHAP_1 = MCI1_4

BYT_1	BYT_0	BLK_2	BLK_1	BLK_0	0	0	0
CHP_4	CHP_3	CHP_2	CHP_1	CHP_0			

AP_CHAP_0 = MCI1_3

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

Extended Display Page Registers

These registers define the address pointer of the display page which will be converted into the extended display memory (internal RAM block 3). Use command SEARCH_PAGE to get the address pointer. Selection on which half of the screen the page appears can be done by bit BES of the page position word (PPW) display control register.

Address Pointer to Extended Display Page

This is the pointer to the page to be displayed.

AP_CHAP_2 = MCI2_5

EXT_MEM = 0	0	0	0	BYT_5	BYT_4	BYT_3	BYT_2
EXT_MEM = 1		CHP_10	CHP_9	CHP_8	CHP_7	CHP_6	CHP_5

AP_CHAP_1 = MCI2_4

BYT_1	BYT_0	BLK_2	BLK_1	BLK_0	0	0	0
CHP_4	CHP_3	CHP_2	CHP_1	CHP_0			

AP_CHAP_0 = MCI2_3

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

Inhibit Update Windows

The inhibit update windows defined by the following registers will be not affected by the S/P-C. The display surrounding a windows will be updated by the S/P-C continuously. Inside the windows user-definable or the time information can be displayed. A window is closed if row (START_COORDINATE) = 25.

COORDINATE_BASIC_2 = MCI1_2

0	START_ROW_4	START_ROW_3	START_ROW_2	START_ROW_1	START_ROW_0	START-COL_5	START-COL_4
---	-------------	-------------	-------------	-------------	-------------	-------------	-------------

COORDINATE_BASIC_1 = MCI1_1

START-COL_3	START-COL_2	START-COL_1	START-COL_0	0	END_ROW_4	END_ROW_3	END_ROW_2
-------------	-------------	-------------	-------------	---	-----------	-----------	-----------

COORDINATE_BASIC_0 = MCI1_0

END_ROW_1	END_ROW_0	END_COL_5	END_COL_4	END_COL_3	END_COL_2	END_COL_1	END_COL_0
-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------

These registers define a window in the basic display memory.

END_ROW_4 thru END_ROW_0, Row/column address bits for window end position.

END_COL_5 thru END_COL_0

START_ROW_4 thru START_ROW_0, Row/column address bits for window start position.

START_COL_5 thru START_COL_0

Following address restrictions must be noticed:

END_ROW (COORDINATE_BASIC) \geq START_ROW (COORDINATE_BASIC).

END_COLUMN (COORDINATE_BASIC) \geq START_COLUMN (COORDINATE_BASIC).

COORDINATE_EXTENDED_2 = MCI2_2

0	START_ROW_4	START_ROW_3	START_ROW_2	START_ROW_1	START_ROW_0	START_COL_5	START_COL_4
---	-------------	-------------	-------------	-------------	-------------	-------------	-------------

COORDINATE_EXTENDED_1 = MCI2_1

START_COL_3	START_COL_2	START_COL_1	START_COL_0	0	END_ROW_4	END_ROW_3	END_ROW_2
-------------	-------------	-------------	-------------	---	-----------	-----------	-----------

COORDINATE_EXTENDED_0 = MCI2_0

END_ROW_1	END_ROW_0	END_COL_5	END_COL_4	END_COL_3	END_COL_2	END_COL_1	END_COL_0
-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------

END_ROW_4 thru END_ROW_0, Row/column address bits for window end position.

END_COL_5 thru END_COL_0

START_ROW_4 thru START_ROW_0, Row/column address bits for window start position.

START_COL_5 thru START_COL_0

Following address restrictions must be noticed:

END_ROW (COORDINATE_BASIC) \geq START_ROW (COORDINATE_BASIC).

END_COLUMN (COORDINATE_BASIC) \geq START_COLUMN (COORDINATE_BASIC).

Return Values

Error Code

ERR_7 – ERR_0	Description
0	Command execution was successful.
1 – 255	Not defined.

Comments

1. After giving the command with SPC_1, SPC_0 = 00, there is a pause until the last S/P-C conversion cycle is terminated before the CMD_RUN bit is set to 0.
2. An S/P-C cycle normally is triggered by acquisition. If acquisition is switched off, an S/P-C cycle can be started by setting **bit 0 of byte 2 in block 0, row 0, column 17** to 1. This bit is reset by the S/P-C after execution, so it has to be set again if S/P-C should be started again.
3. If the translation of the national option characters is switched on, the address of 13 characters of each language is translated into an address of the MTX character ROM. The following table shows the correlation of the transmitted code, the language and the resulting MTX character ROM address.

	2/3	2/4	4/0	5/b	5/c	5/d	5/e	5/f	6/0	7/b	7/c	7/d	7/e
Czechoslovak	023	00d	11a	008	011	00f	185	003	11d	111	181	00a	006
English (UK)	123	124	040	12c	13d	12e	12d	023	02d	13c	01f	13e	138
English (US)	023	024	040	05b	05c	05d	05e	05f	060	07b	07c	07d	07e
French/Belgian	11d	187	110	11f	11e	009	186	023	11c	112	18f	00b	119
German	023	124	127	085	09f	10a	05e	14c	14a	115	000	00c	17b
Italian	123	124	11d	14a	119	12e	12d	023	009	110	18d	11c	184
Polish	023	18a	117	01d	103	168	118	18e	180	012	004	178	010
Portuguese	119	124	121	111	11d	185	18e	00a	13f	00c	18b	11c	110
Rumanian	023	024	106	082	104	084	093	183	007	112	005	114	186
Serbian/Croatian	023	13d	08a	088	10f	162	105	11f	11a	118	011	172	006
Swedish/Finnish	023	024	08d	085	09f	086	10a	14c	11d	115	000	116	00c
Turkish	01e	182	094	104	09f	089	10a	091	183	005	000	119	00c

1.5.27TIME_DISPLAY

This command controls the onscreen time display.

Input Parameters

Time Display Mode

The time display is controlled by the following register:

TIME_DISPLAY_CONTROL = MCIO_5

0	0	0	0	0	0	TIME_EXT	TIME_ON
---	---	---	---	---	---	----------	---------

Variable time positions can be chosen. All attributes of the character display words for time position are user-defineable.

TIME_ON Switches time display on/off
1: The time display will be switched on.
0: The time display will be switched off.

TIME_EXT Time display memory destination
Determines the display memory the time will be written to.
1: The time will be written in the extended display memory.
0: The time will be written in the basic display memory.

Time Position

Row/column position for the time display onscreen.

TIME_POSITION_1 = MCIO_4

0	0	0	0	0	TIME_ROW_4	TIME_ROW_3	TIME_ROW_2
---	---	---	---	---	------------	------------	------------

TIME_POSITION_0 = MCIO_3

TIME_ROW_1	TIME_ROW_0	TIME_COL_5	TIME_COL_4	TIME_COL_3	TIME_COL_2	TIME_COL_1	TIME_COL_0
------------	------------	------------	------------	------------	------------	------------	------------

TIME_COL_5 thru TIME_COL_0 Time column position bits
 $0 \leq \text{TIME_COL_5 thru TIME_COL_0} \leq 39$.

TIME_ROW_5 thru TIME_ROW_0 Time row position bits
 $0 \leq \text{TIME_ROW_5 thru TIME_ROW_0} \leq 24$.

The time position registers are only relevant for TIME_1, TIME_0 = 01 and 11.
In S/P-C double page display mode the time will be displayed only in page 0.

Time Attributes

These are the character attributes for the time display.

TIME_ATTRIBUTE_1 = MCIO_2

TA_23	TA_22	TA_21	TA_20	TA_19	TA_18	TA_17	TA_16
-------	-------	-------	-------	-------	-------	-------	-------

TIME_ATTRIBUTE_0 = MCIO_1

TA_15	TA_14	TA_13	TA_12	TA_11	0	TA_9	0
-------	-------	-------	-------	-------	---	------	---

All above listed time attribute bits have the same meaning as defined for character display words. If bits TA_11 (double height) and/or TA_12 (double width) are set automatically, the double space of character display words are written to the display memory.

Return Values

Error Code

ERR_7 – ERR_0	Description
0	Command execution was successful.
1 – 255	Not defined.

Comments

1. If time display is switched on in video mode, it must be ensured that all character display word attributes of the whole screen are in a user-defined condition. The setting of the attributes of the display words not used by the time display and the mask registers must be according to the display application.
2. Control characters are interpreted as blanks.
3. In parallel mode a magazine number must be defined from which the time information should be read. The definition of the magazine is given by the current display chapter. The display chapter is defined by the address pointer to the basic display page in command SERIAL_PARALLEL_CONVERSION.

1.5.28WRITE_GROUP

This command writes a given PRQ group record to the given PRQ group.

Input Parameters

Search Type Number

The search type number determines the format of the PRQ table.

PRQ_TYPE = MCI0_2

0	0	0	0	0	PRQ_TYPE_2	PRQ_TYPE_1	PRQ_TYPE_0
---	---	---	---	---	------------	------------	------------

PRQ_TYPE_2 thru PRQ_TYPE_0 Page request type number bits
0 ≤ PRQ_TYPE ≤ 3.

PRQ Group Number

The PRQ group number selects the group record to be used.

PRQ_GROUP = MCI0_1

0	0	0	0	0	PRQ_GROUP_2	PRQ_GROUP_1	PRQ_GROUP_0
---	---	---	---	---	-------------	-------------	-------------

PRQ_GROUP_2 thru PRQ_GROUP_0 PRQ group number bits
0 ≤ PRQ_GROUP ≤ 7.

PRQ Do Care Bits

The PRQ do care bits qualify the PRQ record bits of the PRQ records belonging to this group with do or don't care. The qualification is evaluated by the PRQ logic before comparison with the incoming TTX data. Search types 0 and 1 do not use these PRQ do care bits.

GROUP_DO_CARE_4 = MCI1_5

type 0	x	x	x	x	x	x	x	x
type 2	0	0	0	0	0	M2	M1	M0
type 3	ROW4	ROW3	ROW2	ROW1	ROW0	M2	M1	M0

GROUP_DO_CARE_3 = MCI1_4

type 0	x	x	x	x	x	x	x	x
type 2	PT3	PT2	PT1	PT0	PU3	PU2	PU1	PU0
type 3	0	0	0	0	DES3	DES2	DES1	DES0

GROUP_DO_CARE_2 = MCI1_3

type 0	x	x	x	x	x	x	x	x
type 2	C4	MT2	MT1	MT0	MU3	MU2	MU1	MU0
type 3	x	x	x	x	x	x	x	x

GROUP_DO_CARE_1 = MC11_2

type 0	x	x	x	x	x	x	x	x
type 2	C6	C5	HT1	HT0	HU3	HU2	HU1	HU0
type 3	x	x	x	x	x	x	x	x

GROUP_DO_CARE_0 = MC11_1

type 0	x	x	x	x	x	x	x	x
type 2	C14	C13	C12	C11	C10	C9	C8	C7
type 3	x	x	x	x	x	x	x	x

The logic function of the do care bits is:

1. The related PRQ record bit is weighted as do care by the PRQ logic.
2. The related PRQ record bit is weighted as don't care by the PRQ logic.

Return Values

Error Code

ERR_7 – ERR_0	Description
0	Command execution was successful.
1	Not defined.
2	Search type number greater than 3.
3 – 255	Not defined.

Comments

3. The PRQ groups GROUPX_NA (number of allocated elements) will not be affected.
4. For search type 0 the GROUPX_FAP will be set to the PAGE_TLU start address.
For search type 2 and 3 the GROUPX_FAP will be set to zero.
5. For search type 0 the PAGE_TLU is initialized, that means all pointers of this table are set to zero.
6. After giving this command, the group will be automatically signed as used. See also the GR_UNUSE bit of the READ_GROUP command.

1.5.29WRITE_READ_DOUBLEWORD

This command supports comfortable address modes in the internal RAM. It is possible to write single bits in a double word. The read/modify/write of these bits is done by the command. Using the multiplex mode and the DO_CARE_MASK you can compress distributed bits in a doubleword to neighboring bits (and the other way round).

Input Parameters**Do_Care_Mask**

This mask selects the bits of a doubleword to be read or written.

DO_CARE_5 = MCI1_5

DO_C_47	DO_C_46	DO_C_45	DO_C_44	DO_C_43	DO_C_42	DO_C_41	DO_C_40
---------	---------	---------	---------	---------	---------	---------	---------

DO_CARE_4 = MCI1_4

DO_C_39	DO_C_38	DO_C_37	DO_C_36	DO_C_35	DO_C_34	DO_C_33	DO_C_32
---------	---------	---------	---------	---------	---------	---------	---------

DO_CARE_3 = MCI1_3

DO_C_31	DO_C_30	DO_C_29	DO_C_28	DO_C_27	DO_C_26	DO_C_25	DO_C_24
---------	---------	---------	---------	---------	---------	---------	---------

DO_CARE_2 = MCI1_2

DO_C_23	DO_C_22	DO_C_21	DO_C_20	DO_C_19	DO_C_18	DO_C_17	DO_C_16
---------	---------	---------	---------	---------	---------	---------	---------

DO_CARE_1 = MCI1_1

DO_C_15	DO_C_14	DO_C_13	DO_C_12	DO_C_11	DO_C_10	DO_C_09	DO_C_08
---------	---------	---------	---------	---------	---------	---------	---------

DO_CARE_0 = MCI1_0

DO_C_07	DO_C_06	DO_C_05	DO_C_04	DO_C_03	DO_C_02	DO_C_01	DO_C_00
---------	---------	---------	---------	---------	---------	---------	---------

The logic function of the do care bits is:

1: The related bit position will be overwritten if write mode is chosen.

0: The related bit position will not be overwritten.

Data Register

These registers contain the data to be read or written.

DATA_5 = MCI0_5

DATA_47	DATA_46	DATA_45	DATA_44	DATA_43	DATA_42	DATA_41	DATA_40
---------	---------	---------	---------	---------	---------	---------	---------

DATA_4 = MCI0_4

DATA_39	DATA_38	DATA_37	DATA_36	DATA_35	DATA_34	DATA_33	DATA_32
---------	---------	---------	---------	---------	---------	---------	---------

DATA_3 = MCIO_3

DATA_31	DATA_30	DATA_29	DATA_28	DATA_27	DATA_26	DATA_25	DATA_24
---------	---------	---------	---------	---------	---------	---------	---------

DATA_2 = MCIO_2

DATA_23	DATA_22	DATA_21	DATA_20	DATA_19	DATA_18	DATA_17	DATA_16
---------	---------	---------	---------	---------	---------	---------	---------

DATA_1 = MCIO_1

DATA_15	DATA_14	DATA_13	DATA_12	DATA_11	DATA_10	DATA_09	DATA_08
---------	---------	---------	---------	---------	---------	---------	---------

DATA_0 = MCIO_0

DATA_07	DATA_06	DATA_05	DATA_04	DATA_03	DATA_02	DATA_01	DATA_00
---------	---------	---------	---------	---------	---------	---------	---------

Data Pointer

This register defines the block and row column address in the internal RAM where data should be read from or written to.

WRITE_DW_DATAPOINTER_1 = MCI2_4

0	0	BLK_2	BLK_1	BLK_0	ROW_4	ROW_3	ROW_2
---	---	-------	-------	-------	-------	-------	-------

WRITE_DW_DATAPOINTER_0 = MCI2_3

ROW_1	ROW_0	COL_5	COL_4	COL_3	COL_2	COL_1	COL_0
-------	-------	-------	-------	-------	-------	-------	-------

Mode Register**MODE_0 = MCI2_0**

0	0	0	0	0	MPX	READ	R_C_AUTO_INC
---	---	---	---	---	-----	------	--------------

R_C_AUTO_INC**Selects auto-increment mode**

If it is switched on, row is incremented if column overflows 39 and column is set to 0. After row/column 25/23 the block address will be incremented.

- 1: Auto-increment is switched on.
- 0: Auto-increment is switched off.

READ**Selects read/write mode**

- 1: Data will be read from RAM to the data registers.
- 0: Data will be written from data registers to RAM.

MPX**Selects between multiplex and normal write mode**

- 1: Multiplex mode is switched on. If read mode is chosen, every bit which is qualified by the do care mask will be multiplexed to the least significant bits of the data register. All other bits keep their value. For example, if DO_C_8, DO_C_5 and DO_C_2 are set, bits 8, 5 and 2 of the actual

block/row/column address will be copied to DATA_00, DATA_01 and DATA_02 of the data register.

If write mode is chosen, the multiplexer will work the other way round. DATA_00, DATA_01 and DATA_02 will be copied to bits 8, 5 and 2 of the actual row/column address.

- 0: Multiplex mode is switched off. If read mode is chosen, each qualified bit will be read from the actual RAM address and copied to the same bit position in the data register. If write mode is chosen, each qualified bit is copied from the data register to the current RAM address.

Example of write with multiplex:

Bit Position	RAM Contents (old)	Data	Do Care		RAM Contents (new)
0	0	1	0		0
1	0	0	0		0
2	0	1	1	→	1
3	0	0	0		0
4	0	1	0		0
5	0	0	1	→	0
6	0	0	0		0
7	0	0	0		0
8	0	0	1	→	1
9	0	0	0		0
10	0	0	0		0
11	0	0	1	→	0
12	0	0	0		0
13	0	0	0		0
14	0	0	1	→	1
15	0	0	0		0
16	0	0	0		0
17	0	0	0		0
18	0	0	0		0
19	0	0	0		0
20	0	0	0		0
21	0	0	0		0
22	0	0	0		0
23	0	0	0		0
24	0	0	0		0
25	0	0	0		0

Example of write with multiplex (cont'd):

Bit Position	RAM Contents (old)	Data	Do Care		RAM Contents (new)
26	0	0	0		0
27	0	0	0		0
28	0	0	0		0
29	0	0	0		0
30	0	0	0		0
31	0	0	0		0
32	0	0	0		0
33	0	0	0		0
34	0	0	0		0
35	0	0	0		0
36	0	0	0		0
37	0	0	0		0
38	0	0	0		0
39	0	0	0		0
40	0	0	0		0
41	0	0	0		0
42	0	0	0		0
43	0	0	0		0
44	0	0	0		0
45	0	0	0		0
46	0	0	0		0
47	0	0	0		0

Bit_Position_Auto_Increment_Mode

MA_ROT_2 = MCI3_5

0	0	0	MA_ROT_14	MA_ROT_13	MA_ROT_12	MA_ROT_11	MA_ROT_10
---	---	---	-----------	-----------	-----------	-----------	-----------

MA_ROT_14 thru MA_ROT_10 Defines the number of auto-mask rotations after which the next row/column auto-increment will be made. If it is set to 0, auto-mask rotation is switched off.

MA_ROT_1 = MCI3_4

0	0	0	MA_ROT_9	MA_ROT_8	MA_ROT_7	MA_ROT_6	MA_ROT_5
---	---	---	----------	----------	----------	----------	----------

MA_ROT_9 thru MA_ROT_5 Defines the actual number of auto-mask rotations after which the next row/column auto-increment (if switched on by AUTO_INC) will be made. MA_ROT_9 thru MA_ROT_5 will be automatically decremented after each mask rotation by the command (see output parameter). After reaching 0 the number of auto-rotations given by MA_ROT_2 will be copied to MA_ROT_1 and WRITE_DW_DATAPOINTER will be row/column auto-incremented.

MA_ROT_0 = MCI3_3

0	0	0	MA_ROT_4	MA_ROT_3	MA_ROT_2	MA_ROT_1	MA_ROT_0
---	---	---	----------	----------	----------	----------	----------

MA_ROT_4 thru MA_ROT_0 Defines the number of bit rotations (right) of the DO_CARE mask after each execution of this command (if mask auto-rotate is switched on).

Return Values

Do_Care_Mask

DO_CARE_5 = MCI1_5

DO_C_47	DO_C_46	DO_C_45	DO_C_44	DO_C_43	DO_C_42	DO_C_41	DO_C_40
---------	---------	---------	---------	---------	---------	---------	---------

DO_CARE_4 = MCI1_4

DO_C_39	DO_C_38	DO_C_37	DO_C_36	DO_C_35	DO_C_34	DO_C_33	DO_C_32
---------	---------	---------	---------	---------	---------	---------	---------

DO_CARE_3 = MCI1_3

DO_C_31	DO_C_30	DO_C_29	DO_C_28	DO_C_27	DO_C_26	DO_C_25	DO_C_24
---------	---------	---------	---------	---------	---------	---------	---------

DO_CARE_2 = MCI1_2

DO_C_23	DO_C_22	DO_C_21	DO_C_20	DO_C_19	DO_C_18	DO_C_17	DO_C_16
---------	---------	---------	---------	---------	---------	---------	---------

DO_CARE_1 = MCI1_1

DO_C_15	DO_C_14	DO_C_13	DO_C_12	DO_C_11	DO_C_10	DO_C_09	DO_C_08
---------	---------	---------	---------	---------	---------	---------	---------

DO_CARE_0 = MCI1_0

DO_C_07	DO_C_06	DO_C_05	DO_C_04	DO_C_03	DO_C_02	DO_C_01	DO_C_00
---------	---------	---------	---------	---------	---------	---------	---------

DO_CARE_47 thru DO_CARE_0 Defines the next bit positions data should be read from or written to.

Data Register

DATA_5 = MCIO_5

DATA_47	DATA_46	DATA_45	DATA_44	DATA_43	DATA_42	DATA_41	DATA_40
---------	---------	---------	---------	---------	---------	---------	---------

DATA_4 = MCIO_4

DATA_39	DATA_38	DATA_37	DATA_36	DATA_35	DATA_34	DATA_33	DATA_32
---------	---------	---------	---------	---------	---------	---------	---------

DATA_3 = MCIO_3

DATA_31	DATA_30	DATA_29	DATA_28	DATA_27	DATA_26	DATA_25	DATA_24
---------	---------	---------	---------	---------	---------	---------	---------

DATA_2 = MCIO_2

DATA_23	DATA_22	DATA_21	DATA_20	DATA_19	DATA_18	DATA_17	DATA_16
---------	---------	---------	---------	---------	---------	---------	---------

DATA_1 = MCIO_1

DATA_15	DATA_14	DATA_13	DATA_12	DATA_11	DATA_10	DATA_09	DATA_08
---------	---------	---------	---------	---------	---------	---------	---------

DATA_0 = MCIO_0

DATA_07	DATA_06	DATA_05	DATA_04	DATA_03	DATA_02	DATA01	DATA_00
---------	---------	---------	---------	---------	---------	--------	---------

Data Pointer

WRITE_DW_DATAPOINTER_1 = MC12_4

0	0	0	BLK_1	BLK_0	ROW_4	ROW_3	ROW_2
---	---	---	-------	-------	-------	-------	-------

WRITE_DW_DATAPOINTER_0 = MC12_4

ROW_1	ROW_0	COL_5	COL_4	COL_3	COL_2	COL_1	COL_0
-------	-------	-------	-------	-------	-------	-------	-------

BLK_1 thru BLK_0, ROW_4 thru ROW_0, COL_5 thru COL_0 Defines the next block/row/column address data shall be read from or written to.

MA_ROT_1 = MC13_4

0	0	0	MA_ROT_9	MA_ROT_8	MA_ROT_7	MA_ROT_6	MA_ROT_5
---	---	---	----------	----------	----------	----------	----------

MA_ROT_9 thru MA_ROT_5 Defines the actual number of auto-mask rotations after which the next row/column auto-increment (if switched on by AUTO_INC) will be made.

Error Code

ERR_7 – ERR_0	Description
0	Command execution was successful.
1 – 255	Not defined.

Comment

This command can be preferred to write (read) very fast display character words, PCS characters or graphic pixels into (from) the memory.

1.5.15WRITE_TOP_TITLE

This command writes the TOP title from the additional TOP table for a given page number and row/column to position the display memory. The additional TOP table must be located in ACQ group 5.
Packet X/26 of the additional TOP table is considered as well as the received control bits c14, c13 and c12 and the NAT_OPT bits set in command SERIAL_PARALLEL_CONVERSION.

Input Parameters

DISPLAY_MEMORY_ADDRESS

This is the display destination pointer to which the TOP title is to be written.

ROW_COL_POSITION_2 = MCIO_2

0	0	0	0	0	0	0	1
---	---	---	---	---	---	---	---

ROW_COL_POSITION_1 = MCIO_1

1	1	0	BLK_1	BLK_0	0	0	0
---	---	---	-------	-------	---	---	---

ROW_COL_POSITION_0 = MCIO_0

ROW_1	ROW_0	COL_5	COL_4	COL_3	COL_2	COL_1	COL_0
-------	-------	-------	-------	-------	-------	-------	-------

COL_5 thru COL_0 Column address of first TOP title character to be written.

ROW_4 thru ROW_0 Row address of first TOP title character to be written.

BLK_1, BLK_0 Selects display memory to which TOP title is to be written.

Mode

The page number addresses the TOP title to be written in the additional TOP table.

MODE_1 = MCIO_5

0	0	0	APPEND_CHAR	FILL_BLANKS	READ	CUT_BLANKS	PAGE_NUMBER_ENABLE
---	---	---	-------------	-------------	------	------------	--------------------

PAGE_NUMBER_ENABLE 1: If no additional information to the page number specified in PAGE_NUMBER is present, this page number will be changed to ASCII format and written to the position specified in ROW_COL_POSITION.
0: If no additional information is present, nothing will be written to the display memory.

CUT_BLANKS 1: Blanks behind the last TOP title character to be written will be truncated.
0: Blanks behind the last TOP title character to be written will not be truncated and written to the specified.

- READ

1: Only the additional information is returned. Nothing will be written in the display memory.
0: The TOP title will be written to the specified address and the additional information will be returned.
- FILL_BLANKS

1: Instead of a page number or a TOP title, blanks will be written to the specified address. If this bit is set, all other mode bits are ignored. The number of blanks to be written is given in NUMBER_OF_BLANKS. The attributes of these blanks are given in TOP_TITLE_ATTRIBUTES.
0: FILL_BLANKS is switched off.
- APPEND_CHAR

1: A character specified by TOP_TITLE_CHARACTER appended to the written page number or TOP title.
0: APPEND_CHARACTER is switched off.

Page Number

The page number addresses in the additional TOP table the TOP title to be written.

PAGE_NUMBER_1 = MCIO_4

0	0	0	0	0	M2	M1	M0
---	---	---	---	---	----	----	----

PAGE_NUMBER_0 = MCIO_3

PT3	PT2	PT1	PT0	PU3	PU2	PU1	PU0
-----	-----	-----	-----	-----	-----	-----	-----

All above listed page number bits have the same meaning as defined in the world system teletext specification.

Number of Blanks

This register defines the number of blanks to be written.

NUMBER_OF_BLANKS = MCI1_3

0	0	NU_5	NU_4	NU_3	NU_2	NU_1	NU_0
---	---	------	------	------	------	------	------

NU_5 thru NU_0 < 39d

Top Title Attributes

These registers define the character attributes for the TOP title string to be written.

TOP_TITLE_ATTRIBUTE_1 = MCI1_2

TT_23	TT_22	TT_21	TT_20	TT_19	TT_18	TT_17	TT_16
-------	-------	-------	-------	-------	-------	-------	-------

TOP_TITLE_ATTRIBUTE_0 = MCI1_1

TT_15	TT_14	TT_13	0	0	0	TT_9	0
-------	-------	-------	---	---	---	------	---

All above listed TOP title attribute bits have the same meaning as defined for character display words.

TOP Title Characters

This register defines the character to be appended to a written TOP title.

TOP_TITLE_CHARACTER = MCI1_0

CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
-----	-----	-----	-----	-----	-----	-----	-----

With CH7 thru CH0 any character from the MEGATEXT character set can be selected.

Return Values

Additional Information

The first 8 bytes (magazine number, page number tens, page number units, link information, direct selection) of the given page number are returned. For details of these bits refer to the TOP specification of the IRT. The additional information consists of 8 bytes ADDI_BYTE_7 thru ADDI_BYTE_0 with ADDI_BYTE_0 is the leftist. The FALSE bit indicates whether the byte is received correctly.

ADDI_BYTE_7 = MCI2_5

0	0	FALSE	0	AI_7_3	AI_7_2	AI_7_1	AI_7_0
---	---	-------	---	--------	--------	--------	--------

ADDI_BYTE_6 = MCI2_4

0	0	FALSE	0	AI_6_3	AI_6_2	AI_6_1	AI_6_0
---	---	-------	---	--------	--------	--------	--------

ADDI_BYTE_5 = MCI2_3

0	0	FALSE	0	AI_5_3	AI_5_2	AI_5_1	AI_5_0
---	---	-------	---	--------	--------	--------	--------

ADDI_BYTE_4 = MCI2_2

0	0	FALSE	0	AI_4_3	AI_4_2	AI_4_1	AI_4_0
---	---	-------	---	--------	--------	--------	--------

ADDI_BYTE_3 = MCI2_1

0	0	FALSE	0	AI_3_3	AI_3_2	AI_3_1	AI_3_0
---	---	-------	---	--------	--------	--------	--------

ADDI_BYTE_2 = MCI2_0

0	0	FALSE	0	AI_2_3	AI_2_2	AI_2_1	AI_2_0
---	---	-------	---	--------	--------	--------	--------

ADDI_BYTE_1 = MC13_5

0	0	FALSE	0	AI_1_3	AI_1_2	AI_1_1	AI_1_0
---	---	-------	---	--------	--------	--------	--------

ADDI_BYTE_0 = MC13_4

0	0	FALSE	0	AI_0_3	AI_0_2	AI_0_1	AI_0_0
---	---	-------	---	--------	--------	--------	--------

The ADDI bits have the same meaning as described in document “TOP-Verfahren für Fernsehtext” published by IRT Institut für Rundfunktechnik.

Next Column Address

This is the column address behind the TOP title character last written.

ROW_COL_POSITION_2 = MC10_2

0	0	0	0	0	0	0	1
---	---	---	---	---	---	---	---

ROW_COL_POSITION_1 = MC10_1

1	1	0	BLK_1	BLK_0	0	0	0
---	---	---	-------	-------	---	---	---

ROW_COL_POSITION_0 = MC10_0

ROW_1	ROW_0	COL_5	COL_4	COL_3	COL_2	COL_1	COL_0
-------	-------	-------	-------	-------	-------	-------	-------

BLK_1, BLK_0 Block address of the TOP title character last written.

ROW_4 thru ROW_0 Row address of the TOP title character last written.

COL_5 thru COL_0 Column address behind the TOP title character last written.
If this address is equal to the column address in ROW_COL_POSITION_0 and CUT_BLANKS = 1, the TOP title string consists only of blanks.

Error Code

ERR_7 – ERR_0	Description
0	Command execution was successful.
1	Basic TOP table not received.
2	Additional information not found.
3 – 255	Not defined.

Comment

Before the additional information table is received for the first time, all bytes of this chapter are automatically set to 20_H.

1.6 MCI Command Parameter Overview

Command Name	MCIO_5	MCIO_4	MCIO_3	MCIO_2	MCIO_1	MCIO_0
ACQ_CONTROL					TTX_FRAMING_COD	PKT_BUF_ACQ_CON
ADD_PACKET_29_30	PACKET_CONTROL				PRQ_GROUP	
ADD_PAGE	ACQ_CONTROL_1	ACQ_CONTROL_0	NMAX_SUB		PRQ_GROUP	
ADD_ALL_PAGES		PRQ_RECORD_1	PRQ_RECORD_0		REQUEST_MODE	TOP_TYPE
COUNT_TOP_PAGENUMBER	TOP_PAGE_TYPE	PAGE_NUMBER_1	PAGE_NUMBER_0		N_PAGE_BIT_1	N_PAGE_BIT_0
EXECUTE_FLOF				FLOF_MODE	PRQ_GROUP	FLOF_COLOR
MOVE_MEMORY_SEG	MOVE_SOURCE _SEG_START_2	MOVE_SOURCE _SEG_START_1	MOVE_SOURCE _SEG_START_0	MOVE_SOURCE _SEG_END_2	MOVE_SOURCE _SEG_END_1	MOVE_SOURCE _SEG_END_0
READ_CLEAR_PAGE_TRACE		PAGE_NUMBER_1 (i:o)	PAGE_NUMBER_0 (i:o)		SEARCH_MODE	PAGE_REQUEST _STATUS
READ_FLOF_LINK_PAGES	L_PAGE_NUMBER_3	L_PAGE_NUMBER_2	L_PAGE_NUMBER_1	L_PAGE_NUMBER_0	PRQ_GROUP	FLOF_COLOR
READ_GROUP	PRQ_GROUP (i) PRQ_TYPE (o)	GROUPX_NA_2	GROUPX_NA_1	GROUPX_FAP_2	GROUPX_FAP_1	GROUPX_FAP_0
READ_PSEUDO_PACKET_TRACE					PSEUDO_PACKET _TRACE	PP_TRACE_CONTROL
REMOVE_PACKET_25_28					PRQ_GROUP	
REMOVE_PAGE					PRQ_GROUP	REMOVE_MODE
REQUEST_ALL_PAGES	ACQ_CONTROL_1	ACQ_CONTROL_0	NMAX_SUB		PRQ_GROUP	
RESET_INTQ	IRQS_5	IRQS_4	IRQS_3	IRQS_2	IRQS_1	IRQS_0
SEARCH_PACKET_25_28					PRQ_GROUP	
SEARCH_PACKET_29_31	PACKET_CONTROL				PRQ_GROUP	
SEARCH_PAGE					PRQ_GROUP	
SEARCH_TOP_PAGENUMBER	PAGE_TYPE_FOUND	PAGE_NUMBER_1 (i:o)	PAGE_NUMBER_0 (i:o)	TOP_SEARCH_DIR	TOP_PAGE_TYPE_U	TOP_PAGE_TYPE_L
SERIAL_PARALLEL_CONVERSION	SPC_MODE_1	SPC_MODE_0	G0_WINDOW_START	G0_WINDOW_END	G2_WINDOW_START	G2_WINDOW_END
TIME_DISPLAY	TIME_DISPLAY_CONTROL	TIME_POSITION_1	TIME_POSITION_0	TIME_ATTRIBUTE_1	TIME_ATTRIBUTE_0	
WRITE_GROUP				PRQ_TYPE	PRQ_GROUP	
WRITE_READ_DOUBLEWORD	DATA_5	DATA_4	DATA_3	DATA_2	DATA_1	DATA_0
WRITE_TOP_TITLE	MODE	PAGE_NUMBER_1	PAGE_NUMBER_0	ROW_COL POSITION_2 (i:o)	ROW_COL_ POSITION_1 (i:o)	ROW_COL_ POSITION_0 (i:o)

1.6 MCI Command Parameter Overview (cont'd)

Command Name	MC11_5	MC11_4	MC11_3	MC11_2	MC11_1	MC11_0
ACQ_CONTROL						
ADD_PACKET_29_30	PRQ_RECORD_4	PRQ_RECORD_3				
ADD_PAGE	PRQ_RECORD_4	PRQ_RECORD_3	PRQ_RECORD_2	PRQ_RECORD_1	PRQ_RECORD_0	
ADD_ALL_PAGES					N_BASIC_PAGES_1	N_BASIC_PAGES_0
COUNT_TOP_PAGENUMBER		N_PAGE_BLO_1	N_PAGE_BLO_0		N_PAGE_BGR_1	N_PAGE_BGR_0
EXECUTE_FLOF	PRQ_RECORD_4	PRQ_RECORD_3	PRQ_RECORD_2	PRQ_RECORD_1	PRQ_RECORD_0	
MOVE_MEMORY_SEG	MOVE_DESTINATION _SEG_START_2	MOVE_DESTINATION _SEG_START_1	MOVE_DESTINATION _SEG_START_0		SUBSTITUTION _PATTERN_DISABLE	SUBSTITUTION _PATTERN
READ_CLEAR_PAGE_TRACE						
READ_FLOF_LINK_PAGES	PRQ_RECORD_4 (i) L_PAGE_NUM_4 (o)	PRQ_RECORD_3 (i) L_PAGE_NUM_3 (o)	PRQ_RECORD_2 (i) L_PAGE_NUM_2 (o)	PRQ_RECORD_1 (i) L_PAGE_NUM_1 (o)	PRQ_RECORD_0 (i) L_PAGE_NUM_0 (o)	
READ_GROUP	GROUP_DO_CARE_4	GROUP_DO_CARE_3	GROUP_DO_CARE_2	GROUP_DO_CARE_1	GROUP_DO_CARE_0	
READ_PSEUDO_PACKET_TRACE						
REMOVE_PACKET_25_28	PRQ_RECORD_4	PRQ_RECORD_3	PRQ_RECORD_2	PRQ_RECORD_1	PRQ_RECORD_0	
REMOVE_PAGE	PRQ_RECORD_4	PRQ_RECORD_3	PRQ_RECORD_2	PRQ_RECORD_1	PRQ_RECORD_0	
REQUEST_ALL_PAGES						
RESET_INTQ						
SEARCH_PACKET_25_28	PRQ_RECORD_4	PRQ_RECORD_3	PRQ_RECORD_2	PRQ_RECORD_1	PRQ_RECORD_0	DESIGNATION
SEARCH_PACKET_29_31	PRQ_RECORD_4	PRQ_RECORD_3				
SEARCH_PAGE	PRQ_RECORD_4	PRQ_RECORD_3	PRQ_RECORD_2	PRQ_RECORD_1	PRQ_RECORD_0	
SEARCH_TOP_PAGENUMBER						
SERIAL_PARALLEL_CONVERSION	AP_CHAP_2	AP_CHAP_1	AP_CHAP_0	COORDINATE_ BASIC_2	COORDINATE_ BASIC_1	COORDINATE_ BASIC_0
TIME_DISPLAY						
WRITE_GROUP	GROUP_DO_CARE_4	GROUP_DO_CARE_3	GROUP_DO_CARE_2	GROUP_DO_CARE_1	GROUP_DO_CARE_0	
WRITE_READ_DOUBLEWORD	DO_CARE_5	DO_CARE_4	DO_CARE_3	DO_CARE_2	DO_CARE_1	DO_CARE_0
WRITE_TOP_TITLE				TOP_TITLE_ ATTRIBUTE_1	TOP_TITLE_ ATTRIBUTE_0	

2 Appendix

2.1 Special RAM Addresses and Their Functions

Because it costs a lot of ROM code in the MEGATEXT CPU to handle all the parameters of the MCI commands, it is necessary to read and write some input and output parameters direct to certain RAM addresses (by use of M3L-Bus register DATA_PORT_0 or DATA_PORT 1). In the following these addresses and their corresponding features are explained.

2.1.1 Initialization

After power-on the firmware initializes some parts of the RAM and some hardware registers. This is done in the following sequence:

- 1. Set GPO (General Purpose Output) to 1 (M3L register 82 bit 5). Only the hardware register is set (not the corresponding RAM register).
- 2. Wait 2 ms.
- 3. Hold M3L Bus.
- 4. Initialize whole RAM block 0 doublewords with 000000H 000000H.
- 5. Outer screen and inner screen are set to transparent (inner screen and outer screen mask and data registers bit 14 and bit 15).
- 6. Initialize whole RAM block 2 doublewords with 000000H 700020H.
- 7. Initialize M3L registers with date code of this firmware (see M3L register reference). Version C13: 03 15 93.
- 8. Refresh of external RAM is switched on.
- 9. Initialize framing code of ACQ interface to e4H.
- 10.Release clock line of M3L Bus.

2.1.2 GROUP_INIT

These registers define whether an ACQ group is used or not. They must be set to not used (0) by the controller and to used (1) by the command WRITE_GROUP. And the pointer to the first chapter of these groups must be set to zero before the first page is added.

GROUP_0_USED = BLOCK 0, BYTE 5, ROW 7, COLUMN 0

GROUP_0_USED	0	0	0	0	0	0	0
--------------	---	---	---	---	---	---	---

GROUP_0_NIL = BLOCK 0, BYTE 0, ROW 7, COLUMN 0

0	0	0	0	0	0	0	NIL_0
---	---	---	---	---	---	---	-------

GROUP_1_USED = BLOCK 0, BYTE 5, ROW 7, COLUMN 2

GROUP_1_USED	0	0	0	0	0	0	0
--------------	---	---	---	---	---	---	---

GROUP_1_NIL = BLOCK 0, BYTE 0, ROW 7, COLUMN 2

0	0	0	0	0	0	0	NIL_1
---	---	---	---	---	---	---	-------

GROUP_2_USED = BLOCK 0, BYTE 5, ROW 7, COLUMN 4

GROUP_2_USED	0	0	0	0	0	0	0
--------------	---	---	---	---	---	---	---

GROUP_2_NIL = BLOCK 0, BYTE 0, ROW 7, COLUMN 4

0	0	0	0	0	0	0	NIL_2
---	---	---	---	---	---	---	-------

GROUP_3_USED = BLOCK 0, BYTE 5, ROW 7, COLUMN 6

GROUP_3_USED	0	0	0	0	0	0	0
--------------	---	---	---	---	---	---	---

GROUP_3_NIL = BLOCK 0, BYTE 0, ROW 7, COLUMN 6

0	0	0	0	0	0	0	NIL_3
---	---	---	---	---	---	---	-------

GROUP_4_USED = BLOCK 0, BYTE 5, ROW 7, COLUMN 8

GROUP_4_USED	0	0	0	0	0	0	0
--------------	---	---	---	---	---	---	---

GROUP_4_NIL = BLOCK 0, BYTE 0, ROW 7, COLUMN 8

0	0	0	0	0	0	0	NIL_4
---	---	---	---	---	---	---	-------

GROUP_5_USED = BLOCK 0, BYTE 5, ROW 7, COLUMN 10

GROUP_5_USED	0	0	0	0	0	0	0
--------------	---	---	---	---	---	---	---

GROUP_5_NIL = BLOCK 0, BYTE 0, ROW 7, COLUMN 10

0	0	0	0	0	0	0	NIL_5
---	---	---	---	---	---	---	-------

GROUP_6_USED = BLOCK 0, BYTE 5, ROW 7, COLUMN 12

GROUP_6_USED	0	0	0	0	0	0	0
--------------	---	---	---	---	---	---	---

GROUP_6_NIL = BLOCK 0, BYTE 0, ROW 7, COLUMN 12

0	0	0	0	0	0	0	NIL_6
---	---	---	---	---	---	---	-------

GROUP_7_USED = BLOCK 0, BYTE 5, ROW 7, COLUMN 14

GROUP_7_USED	0	0	0	0	0	0	0
--------------	---	---	---	---	---	---	---

GROUP_7_NIL = BLOCK 0, BYTE 0, ROW 7, COLUMN 14

0	0	0	0	0	0	0	NIL_7
---	---	---	---	---	---	---	-------

2.1.2 X_RAM_SIZE

These registers define the size of the external RAM in Kbytes. They must be set during initialization before the command CREATE_FREE_CHAP_CHAIN is executed.

SIZE_XRAM_1 = BLOCK 0, BYTE 1, ROW 0, COLUMN 31

0	0	0	0	0	X_SIZE_10	X_SIZE_9	X_SIZE_8
---	---	---	---	---	-----------	----------	----------

SIZE_XRAM_0 = BLOCK 0, BYTE 1, ROW 0, COLUMN 31

X_SIZE_7	X_SIZE_6	X_SIZE_5	X_SIZE_4	X_SIZE_3	X_SIZE_2	X_SIZE_1	X_SIZE_0
----------	----------	----------	----------	----------	----------	----------	----------

$0 \leq X_SIZE \leq 2048$

2.1.3 SOFTWARE_REFRESH

This register switches on or off the refresh of external RAM and defines the refresh cycle time in units of the line period of the CVBS signal (television lines).

SOFTWARE_REFRESH = BLOCK 0, BYTE 0, ROW 6, COLUMN 31

REF_7	REF_6	REF_5	REF_4	REF_3	REF_2	REF_1	REF_0
-------	-------	-------	-------	-------	-------	-------	-------

REF_7 thru REF_0 Defines the number of television lines which are counted between two refresh bursts. A refresh burst means that the next (in binary order) 256 row addresses of external RAM are addressed. The handling of the refresh addresses of external RAM is managed by the firmware. The time that passes between two bursts is the product of the television line period (typ. 64 μ s) and the number given in SOFTWARE_REFRESH. If this value is set to 64d every 4096 μ s, the next 256 row addresses of external RAM are refreshed. With this value it is possible to refresh all known DRAMs up to a size of 16M \times 1.

REF_7 thru REF_0 = 0d0 means refresh is switched off.

2.1.4 SIGNAL_QUALITY

Each time the PU returns to the beginning of the loop of the main program, some status information from the chip is written to a special address in internal RAM. This information can be read by the external controller any time.

MEGA_STATUS = BLOCK 0, BYTE 0, ROW 7, COLUMN 19

x	x	x	x	x	x	COI	ACQ_EN
---	---	---	---	---	---	-----	--------

ACQ_EN **Acquisition Sync Signal Reference**

1: HPLL lock condition is very good.

0: HPLL lock condition is bad.

COI **Coincidence Indicator**

1: HPLL phase difference is less than 4% of line period.

0: HPLL is not locked.

To get more information about the **average** quality of the CVBS signal, two further signal quality routines can be switched on.

This can be done by loading the parameter SIGNAL_LINES and TELETEXT_LINES with any value but zero.

SIGNAL_LINES_1 = BLOCK 0, BYTE 4, ROW 0, COLUMN 32

SI_L_15	SI_L_14	SI_L_13	SI_L_12	SI_L_11	SI_L_10	SI_L_9	SI_L_8
---------	---------	---------	---------	---------	---------	--------	--------

SIGNAL_LINES_0 = BLOCK 0, BYTE 3, ROW 0, COLUMN 32

SI_L_7	SI_L_6	SI_L_5	SI_L_4	SI_L_3	SI_L_2	SI_L_1	SI_L_0
--------	--------	--------	--------	--------	--------	--------	--------

SIGNAL_LINES_1 – 0 defines an interval in which several items of signal quality information are accumulated. After this time period has expired, the information will be written to a special RAM register (see below) where it can be read out. The following status information is evaluated:

ACQ_EN: acquisition sync signal reference (1 bit)

COI: coincidence indicator (1 bit)

SYNC_ERRORS: number of horizontal sync distortions (4 bits)

Every 4th television line this information is accumulated until the number of television lines given by SIGNAL_LINES_1 – 0 is reached.

Example: SIGNAL_LINES is set to 100d and the value read from register COI equals 500. This means that in every second measured television line the HPLL was not locked (lock condition is very bad).

TELETEXT_LINES_1 = BLOCK 0, BYTE 4, ROW 0, COLUMN 36

TTL_L_15	TTL_L_14	TTL_L_13	TTL_L_12	TTL_L_11	TTL_L_10	TTL_L_9	TTL_L_8
----------	----------	----------	----------	----------	----------	---------	---------

TELETEXT_LINES_0 = BLOCK 0, BYTE 3, ROW 0, COLUMN 36

TTL_L_7	TTL_L_6	TTL_L_5	TTL_L_4	TTL_L_3	TTL_L_2	TTL_L_1	TTL_L_0
---------	---------	---------	---------	---------	---------	---------	---------

TELETEXT_LINES_1 – 0 defines the number of received teletext lines in which 1-bit hamming errors, 2-bit hamming errors and 1-bit framing code errors should be counted. After this time period has expired, these values will be written to special RAM registers where they can be read out. In each teletext line the hamming check of the magazine and the row number is evaluated. Teletext lines with more than 1-bit framing code errors are not considered.

The number of TELETEXT_LINES is independent of the number of SIGNAL_LINES.

SQ_ACQ_EN equals the accumulated value of the status information of every 4th television line.

SQ_ACQ_EN_2 = BLOCK 0, BYTE 5, ROW 0, COLUMN 33

ACQ_EN_23	ACQ_EN_22	ACQ_EN_21	ACQ_EN_20	ACQ_EN_19	ACQ_EN_18	ACQ_EN_17	ACQ_EN_16
-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------

SQ_ACQ_EN_1 = BLOCK 0, BYTE 4, ROW 0, COLUMN 33

ACQ_EN_15	ACQ_EN_14	ACQ_EN_13	ACQ_EN_12	ACQ_EN_11	ACQ_EN_10	ACQ_EN_09	ACQ_EN_08
-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------

SQ_ACQ_EN_0 = BLOCK 0, BYTE 3, ROW 0, COLUMN 33

ACQ_EN_07	ACQ_EN_06	ACQ_EN_05	ACQ_EN_04	ACQ_EN_03	ACQ_EN_02	ACQ_EN_01	ACQ_EN_00
-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------

0 < ACQ_EN_23 thru 00 < SIGNAL_LINES_15 thru 0

SQ_COI equals the accumulated value of the status information of every 4th television line.

SQ_COI_2 = BLOCK 0, BYTE 5, ROW 0, COLUMN 34

COI_23	COI_22	COI_21	COI_20	COI_19	COI_18	COI_17	COI_16
--------	--------	--------	--------	--------	--------	--------	--------

SQ_COI_1 = BLOCK 0, BYTE 4, ROW 0, COLUMN 34

COI_15	COI_14	COI_13	COI_12	COI_11	COI_10	COI_09	COI_08
--------	--------	--------	--------	--------	--------	--------	--------

SQ_COI_0 = BLOCK 0, BYTE 3, ROW 0, COLUMN 34

COI_07	COI_06	COI_05	COI_04	COI_03	COI_02	COI_01	COI_00
--------	--------	--------	--------	--------	--------	--------	--------

0 < COI_23 thru 00 < SIGNAL_LINES_15 thru 0

SQ_SYNC_ERR equals the accumulated value of the status information of every 4th television line.

SQ_SYNC_ERR_2 = BLOCK 0, BYTE 5, ROW 0, COLUMN 35

SYNC_ERR_23	SYNC_ERR_22	SYNC_ERR_21	SYNC_ERR_20	SYNC_ERR_19	SYNC_ERR_18	SYNC_ERR_17	SYNC_ERR_16
-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------

SQ_SYNC_ERR_1 = BLOCK 0, BYTE 4, ROW 0, COLUMN 35

SYNC_ERR_15	SYNC_ERR_14	SYNC_ERR_13	SYNC_ERR_12	SYNC_ERR_11	SYNC_ERR_10	SYNC_ERR_09	SYNC_ERR_08
-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------

SQ_SYNC_ERR_0 = BLOCK 0, BYTE 3, ROW 0, COLUMN 35

SYNC_ERR_07	SYNC_ERR_06	SYNC_ERR_05	SYNC_ERR_04	SYNC_ERR_03	SYNC_ERR_02	SYNC_ERR_01	SYNC_ERR_00
-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------

0 < SYNC_ERR_23 thru 00 < (SIGNAL_LINES_15 thru 0) × 16

SQ_1_BIT_FRCE equals the accumulated value of the 1-bit framing code errors of every received teletext line.

SQ_1_BIT_FRCE_2 = BLOCK 0, BYTE 5, ROW 0, COLUMN 37

1_BIT_FRCE_23	1_BIT_FRCE_22	1_BIT_FRCE_21	1_BIT_FRCE_20	1_BIT_FRCE_19	1_BIT_FRCE_18	1_BIT_FRCE_17	1_BIT_FRCE_16
---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------

SQ_1_BIT_FRCE_1 = BLOCK 0, BYTE 4, ROW 0, COLUMN 37

1_BIT_FRCE_15	1_BIT_FRCE_14	1_BIT_FRCE_13	1_BIT_FRCE_12	1_BIT_FRCE_11	1_BIT_FRCE_10	1_BIT_FRCE_09	1_BIT_FRCE_08
---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------

SQ_1_BIT_FRCE_0 = BLOCK 0, BYTE 3, ROW 0, COLUMN 37

1_BIT_FRCE_07	1_BIT_FRCE_06	1_BIT_FRCE_05	1_BIT_FRCE_04	1_BIT_FRCE_03	1_BIT_FRCE_02	1_BIT_FRCE_01	1_BIT_FRCE_00
---------------	---------------	---------------	---------------	---------------	---------------	---------------	---------------

0 < 1_BIT_FRCE_23 thru 00 < (SIGNAL_LINES_15 thru 0)

SQ_1_BIT_HAM equals the accumulated value of the 1-bit hamming errors of row and magazine number of every received teletext line.

SQ_1_BIT_HAM_2 = BLOCK 0, BYTE 5, ROW 0, COLUMN 38

1_BIT_HAM_23	1_BIT_HAM_22	1_BIT_HAM_21	1_BIT_HAM_20	1_BIT_HAM_19	1_BIT_HAM_18	1_BIT_HAM_17	1_BIT_HAM_16
--------------	--------------	--------------	--------------	--------------	--------------	--------------	--------------

SQ_1_BIT_HAM_1 = BLOCK 0, BYTE 4, ROW 0, COLUMN 38

1_BIT_HAM_15	1_BIT_HAM_14	1_BIT_HAM_13	1_BIT_HAM_12	1_BIT_HAM_11	1_BIT_HAM_10	1_BIT_HAM_09	1_BIT_HAM_08
--------------	--------------	--------------	--------------	--------------	--------------	--------------	--------------

SQ_1_BIT_HAM_0 = BLOCK 0, BYTE 3, ROW 0, COLUMN 38

1_BIT_HAM_07	1_BIT_HAM_06	1_BIT_HAM_05	1_BIT_HAM_04	1_BIT_HAM_03	1_BIT_HAM_02	1_BIT_HAM_01	1_BIT_HAM_00
--------------	--------------	--------------	--------------	--------------	--------------	--------------	--------------

$0 < 1_BIT_HAM_{23} \text{ thru } 00 < (\text{SIGNAL_LINES}_{15} \text{ thru } 0) \times 2$

SQ_2_BIT_HAM equals the accumulated value of the 2-bit hamming errors of row and magazine number of every received teletext line.

SQ_2_BIT_HAM_2 = BLOCK 0, BYTE 5, ROW 0, COLUMN 39

1_BIT_HAM_23	1_BIT_HAM_22	1_BIT_HAM_21	1_BIT_HAM_20	1_BIT_HAM_19	1_BIT_HAM_18	1_BIT_HAM_17	1_BIT_HAM_16
--------------	--------------	--------------	--------------	--------------	--------------	--------------	--------------

SQ_2_BIT_HAM_1 = BLOCK 0, BYTE 4, ROW 0, COLUMN 39

1_BIT_HAM_25	1_BIT_HAM_24	1_BIT_HAM_23	1_BIT_HAM_22	1_BIT_HAM_21	1_BIT_HAM_20	1_BIT_HAM_19	1_BIT_HAM_18
--------------	--------------	--------------	--------------	--------------	--------------	--------------	--------------

SQ_2_BIT_HAM_0 = BLOCK 0, BYTE 3, ROW 0, COLUMN 39

1_BIT_HAM_07	1_BIT_HAM_06	1_BIT_HAM_05	1_BIT_HAM_04	1_BIT_HAM_03	1_BIT_HAM_02	1_BIT_HAM_01	1_BIT_HAM_00
--------------	--------------	--------------	--------------	--------------	--------------	--------------	--------------

$0 < 1_BIT_HAM_{23} \text{ thru } 00 < (\text{SIGNAL_LINES}_{25} \text{ thru } 0) \times 2$

2.1.5 SUBROUTINE_CALL_ADDRESS

This address can be the calling address of any subroutine stored in the ROM of MEGATEXT or in internal RAM.

CALL_SUB_ADR_2 = BLOCK 0, BYTE 2, ROW 2, COLUMN 30

0	1	0	RAM = 1	1	1	1	1
0	1	0	RAM = 0	1	1	1	1

CALL_SUB_ADR_1 = BLOCK 0, BYTE 1, ROW 2, COLUMN 30

1	1	0	BLK_1	BLK_0	ADR_10	ADR_9	ADR_8
1	1	0	ADR_12	ADR_11	ADR_10	ADR_9	ADR_8

CALL_SUB_ADR_0 = BLOCK 0, BYTE 0, ROW 2, COLUMN 30

ADR_7	ADR_6	ADR_5	ADR_4	ADR_3	ADR_2	ADR_1	ADR_0
ADR_7	ADR_6	ADR_5	ADR_4	ADR_3	ADR_2	ADR_1	ADR_0

RAM

Control bit for IRAM/ROM-memory selection

1: RAM memory is selected.
0: ROM memory is selected.

BLK_1, BLK_0

Block address bits

$0 \leq \text{BLK_} \leq 3$.

ADR_12 thru ADR_0

Binary address bits

$0 \leq \text{ADR_} \leq 8$ Kwords for ROM programs. For IRAM programs the maximum size depends on IRAM memory allocation.

2.1.6 VPS

The VPS data are error checked and written to a buffer in the MEGATEXT RAM. Their actual values can be read any time. The received data are biphase decoded and only written if the whole VPS line is received without any errors. The status word shows the number of detected errors. It must be set to any other value than 0, if the user wants to know whether a new VPS line is received or not. Clock-run-in and framing code are not stored.

VPS_DATA = BLOCK 0, BYTE 2, ROW 1, COLUMN 0 to COLUMN 13

Column Position	VPS Data Word
0	Source identification
1	Source identification
2	Sound data special identification
3	Signal content identification
4	ASCII plain text channel
5	Routing
6	Routing
7	Reports/commands
8	VPS extra information
9	VPS extra information
10	VPS extra information
11	VPS extra information
12	Reserved for data protection
13	Status word

2.2 Stored Format of Page

False = Protection bit (result of 1-byte hamming or parity check) (1 = failed)

Di = Data bit i of byte j

The bits in column 4 - 7 in row 0 have the same meaning as defined in the world system teletext specification. They represent the header bits of received packet 0.

All bytes in column 8 - 39 of row 0 and all bytes in row 1 are either parity or hamming checked (corresponding to the world system teletext specification). The FALSE bit represents the check result (1 = failed).

Row	Column	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
R0	C0 – C2	Reserved for internal use							
R0	C3	0	0	0	0	0	M2	M1	M0
R0	C4	PT3	PT2	PT1	PT0	PU3	PU2	PU1	PU0
R0	C5	C4	MT2	MT1	MT0	MU3	MU2	MU1	MU0
R0	C6	C6	C5	HT1	HT0	HU3	HU2	HU1	HU0
R0	C7	C13	C12	C11	C10	C9	C8	C7	C6
R0	C8 – C39	False False	D6 0	D5 0	D4 0	D3 D3	D2 D2	D1 D1	D0 D0
R1 – R24	C0 – C39	False False	D6 0	D5 0	D4 0	D3 D3	D2 D2	D1 D1	D0 D0
R25	C0 – C23	Reserved for internal use							

2.3 Stored Format of Linking List of Compressed BTT

The chapter referenced by M3L register 120 - 122 (TOP_COM_BAS) contains the page_linking_table and the TOP table list of the basic TOP table of each magazine. Data stored in this chapter are hamming checked and compressed to one nibble per byte. The most significant nibble is zero if the byte is stored errorless.

Row	Column	
R1	C0 – C39	Linking list of magazine 8 (field 1 – 5)
R2	C0 – C39	Linking list of magazine 8 (field 5 – 10)
R3	C32 – C39	TOP table list of magazine 8
R4	C0 – C39	Linking list of magazine 1 (field 1 – 5)
R5	C0 – C39	Linking list of magazine 1 (field 5 – 10)
R6	C32 – C39	TOP table list of magazine 1

Row	Column	
R7	C0 – C39	Linking list of magazine 2 (field 1 – 5)
R8	C0 – C39	Linking list of magazine 2 (field 5 – 10)
R9	C32 – C39	TOP table list of magazine 2
R10	C0 – C39	Linking list of magazine 3 (field 1 – 5)
R11	C0 – C39	Linking list of magazine 3 (field 5 – 10)
R12	C32 – C39	TOP table list of magazine 3
R13	C0 – C39	Linking list of magazine 4 (field 1 – 5)
R14	C0 – C39	Linking list of magazine 4 (field 5 – 10)
R15	C32 – C39	TOP table list of magazine 4
R16	C0 – C39	Linking list of magazine 5 (field 1 – 5)
R17	C0 – C39	Linking list of magazine 5 (field 5 – 10)
R18	C32 – C39	TOP table list of magazine 5
R19	C0 – C39	Linking list of magazine 6 (field 1 – 5)
R20	C0 – C39	Linking list of magazine 6 (field 5 – 10)
R21	C32 – C39	TOP table list of magazine 6
R22	C0 – C39	Linking list of magazine 7 (field 1 – 5)
R23	C0 – C39	Linking list of magazine 7 (field 5 – 10)
R24	C32 – C39	TOP table list of magazine 7

2.4 Stored Format of X/25

Packet X/25

F_j = Protection bit (result of parity check) of byte j (1 = failed)

Di_j = Data bit i of byte j

Column	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
C0 – C39	F _j	D6 _j	D5 _j	D4 _j	D3 _j	D2 _j	D1 _j	D0 _j

2.5 Stored Format of 1-Byte Hamming Protected Data

The 1-byte hamming check of data bytes transmitted in TOP tables is done online before storing the data byte. The data bits are compressed to the four LSBs. Bit 5 of this byte shows whether the check was failed or not.

F_j = Protection bit (result of hamming check) of byte j (1 = failed)

Di_j = Data bit i of byte j

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	F _j	0	D3 _j	D2 _j	D1 _j	D0 _j

2.2 Stored Format of Packets with 3-Byte Hamming Checks

Packet X/26/ 000 thru 1111, packet X/27/ 0100 thru 0111, packet X/28/ 0000, 0001, 0011, packet X/29/ 0000, 0001, 0100

F_Gi = Protection bit (result of 3-byte hamming check) of 3-byte group i (1 = failed)

Di_Gj = Data bit i of 3-byte group j

Column	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
C0	Reserved for internal use							
C1	D7_G0	D6_G0	D5_G0	D4_G0	D3_G0	D2_G0	D1_G0	D0_G0
C2	D15_G0	D14_G0	D13_G0	D12_G0	D11_G0	D10_G0	D9_G0	D8_G0
C3	D3_G1	D2_G1	D1_G1	D0_G1	0	F_G0	D17_G0	D16_G0
C4	D11_G1	D10_G1	D9_G1	D8_G1	D7_G1	D6_G1	D5_G1	D4_G1
C5	0	F_G1	D17_G1	D16_G1	D15_G1	D14_G1	D13_G1	D12_G1
C6	D7_G2	D6_G2	D5_G2	D4_G2	D3_G2	D2_G2	D1_G2	D0_G2
C7	D15_G2	D14_G2	D13_G2	D12_G2	D11_G2	D10_G2	D9_G2	D8_G2
C8	D3_G3	D2_G3	D1_G3	D0_G3	0	F_G2	D17_G2	D16_G2
C9	D11_G3	D10_G3	D9_G3	D8_G3	D7_G3	D6_G3	D5_G3	D4_G3
C10	0	F_G3	D17_G3	D16_G3	D15_G3	D14_G3	D13_G3	D12_G3
C11	D7_G4	D6_G4	D5_G4	D4_G4	D3_G4	D2_G4	D1_G4	D0_G4
C12	D15_G4	D14_G4	D13_G4	D12_G4	D11_G4	D10_G4	D9_G4	D8_G4
C13	D3_G5	D2_G5	D1_G5	D0_G5	0	F_G4	D17_G4	D16_G4
C14	D11_G5	D10_G5	D9_G5	D8_G5	D7_G5	D6_G5	D5_G5	D4_G5
C15	0	F_G5	D17_G5	D16_G5	D15_G5	D14_G5	D13_G5	D12_G5
C16	D7_G6	D6_G6	D5_G6	D4_G6	D3_G6	D2_G6	D1_G6	D0_G6
C17	D15_G6	D14_G6	D13_G6	D12_G6	D11_G6	D10_G6	D9_G6	D8_G6
C18	D3_G7	D2_G7	D1_G7	D0_G7	0	F_G6	D17_G6	D16_G6
C19	D11_G7	D10_G7	D9_G7	D8_G7	D7_G7	D6_G7	D5_G7	D4_G7
C20	0	F_G7	D17_G7	D16_G7	D15_G7	D14_G7	D13_G7	D12_G7
C21	D7_G8	D6_G8	D5_G8	D4_G8	D3_G8	D2_G8	D1_G8	D0_G8
C22	D15_G8	D14_G8	D13_G8	D12_G8	D11_G8	D10_G8	D9_G8	D8_G8
C23	D3_G9	D2_G9	D1_G9	D0_G9	0	F_G8	D17_G8	D16_G8
C24	D11_G9	D10_G9	D9_G9	D8_G9	D7_G9	D6_G9	D5_G9	D4_G9
C25	0	F_G9	D17_G9	D16_G9	D15_G9	D14_G9	D13_G9	D12_G9
C26	D7_G10	D6_G10	D5_G10	D4_G10	D3_G10	D2_G10	D1_G10	D0_G10
C27	D15_G10	D14_G10	D13_G10	D12_G10	D11_G10	D10_G10	D9_G10	D8_G10
C28	D3_G11	D2_G11	D1_G11	D0_G11	0	F_G10	D17_G10	D16_G10
C29	D11_G11	D10_G11	D9_G11	D8_G11	D7_G11	D6_G11	D5_G11	D4_G11
C30	0	F_G11	D17_G11	D16_G11	D15_G11	D14_G11	D13_G11	D12_G11
C31	D7_G12	D6_G12	D5_G12	D4_G12	D3_G12	D2_G12	D1_G12	D0_G12

Column	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
C32	D15_G12	D14_G12	D13_G12	D12_G12	D11_G12	D10_G12	D9_G12	D8_G12
C33	0	0	0	0	0	F_G12	D17_G12	D16_G12
C34	Reserved for internal use							
C35	Reserved for internal use							
C36	Reserved for internal use							
C37	Reserved for internal use							
C38	Reserved for internal use							
C39	Reserved for internal use							

2.6 Stored Format of Packets with 6 1-Byte Hamming Checks

Packet 27 (0000 thru 0011)

F_j = Protection bit (result of 1-byte hamming or parity check) of byte j (1 = failed)

Di_j = Data bit i of byte j

Column	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
C0	Reserved for internal use							
C1	0	0	F_12	F_11	F_10	F_9	F_8	F_7
C2	D3_8	D2_8	D1_8	D0_8	D3_7	D2_7	D1_7	D0_7
C3	D3_10	D2_10	D1_10	D0_10	D3_9	D2_9	D1_9	D0_9
C4	D3_12	D2_12	D1_12	D0_12	D3_11	D2_11	D1_11	D0_11
C5	0	0	F_18	F_17	F_16	F_15	F_14	F_13
C6	D3_14	D2_14	D1_14	D0_14	D3_13	D2_13	D1_13	D0_13
C7	D3_16	D2_16	D1_16	D0_16	D3_15	D2_15	D1_15	D0_15
C8	D3_18	D2_18	D1_18	D0_18	D3_17	D2_17	D1_17	D0_17
C9	0	0	F_24	F_23	F_22	F_21	F_20	F_19
C10	D3_20	D2_20	D1_20	D0_20	D3_19	D2_19	D1_19	D0_19
C11	D3_22	D2_22	D1_22	D0_22	D3_21	D2_21	D1_21	D0_21
C12	D3_24	D2_24	D1_24	D0_24	D3_23	D2_23	D1_23	D0_23
C13	0	0	F_30	F_29	F_28	F_27	F_26	F_25
C14	D3_26	D2_26	D1_26	D0_26	D3_25	D2_25	D1_25	D0_25
C15	D3_28	D2_28	D1_28	D0_28	D3_27	D2_27	D1_27	D0_27
C16	D3_30	D2_30	D1_30	D0_30	D3_29	D2_29	D1_29	D0_29
C17	0	0	F_36	F_35	F_34	F_33	F_32	F_31
C18	D3_32	D2_32	D1_32	D0_32	D3_31	D2_31	D1_31	D0_31
C19	D3_34	D2_34	D1_34	D0_34	D3_33	D2_33	D1_33	D0_33
C20	D3_36	D2_36	D1_36	D0_36	D3_35	D2_35	D1_35	D0_35

Column	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
C21	0	0	F_42	F_41	F_40	F_39	F_38	F_37
C22	D3_38	D2_38	D1_38	D0_38	D3_37	D2_37	D1_37	D0_37
C23	D3_40	D2_40	D1_40	D0_40	D3_39	D2_39	D1_39	D0_39
C24	D3_42	D2_42	D1_42	D0_42	D3_41	D2_41	D1_41	D0_41
C25	0	0	0	0	0	0	0	F_43
C26	0	0	0	0	D3_43	D2_43	D1_43	D0_43
C27	D7_44	D6_44	D5_44	D4_44	D3_44	D2_44	D1_44	D0_44
C28	D7_45	D6_45	D5_45	D4_45	D3_45	D2_45	D1_45	D0_45
C29	Reserved for internal use							
C30	Reserved for internal use							
C31	Reserved for internal use							
C32	Reserved for internal use							
C33	Reserved for internal use							
C34	Reserved for internal use							
C35	Reserved for internal use							
C36	Reserved for internal use							
C37	Reserved for internal use							
C38	Reserved for internal use							
C39	Reserved for internal use							

2.7 Stored Format of Packet 8/30 (000x)

F_j = Protection bit (result of hamming or parity check) of byte j (1 = failed)

Di_j = Data bit i of byte j

Column	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
C0	Reserved for internal use							
C1	0	0	F_7	0	D3_7	D2_7	D1_7	D0_7
C2	0	0	F_8	0	D3_8	D2_8	D1_8	D0_8
C3	0	0	F_9	0	D3_9	D2_9	D1_9	D0_9
C4	0	0	F_10	0	D3_10	D2_10	D1_10	D0_10
C5	0	0	F_11	0	D3_11	D2_11	D1_11	D0_11
C6	0	0	F_12	0	D3_12	D2_12	D1_12	D0_12
C7	D7_13	D6_13	D5_13	D4_13	D3_13	D2_13	D1_13	D0_13
C8	D7_14	D6_14	D5_14	D4_14	D3_14	D2_14	D1_14	D0_14
C9	D7_15	D6_15	D5_15	D4_15	D3_15	D2_15	D1_15	D0_15
C10	D7_16	D6_16	D5_16	D4_16	D3_16	D2_16	D1_16	D0_16

Column	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
C11	D7_17	D6_17	D5_17	D4_17	D3_17	D2_17	D1_17	D0_17
C12	D7_18	D6_18	D5_18	D4_18	D3_18	D2_18	D1_18	D0_18
C13	D7_19	D6_19	D5_19	D4_19	D3_19	D2_19	D1_19	D0_19
C14	D7_20	D6_20	D5_20	D4_20	D3_20	D2_20	D1_20	D0_20
C15	D7_21	D6_21	D5_21	D4_21	D3_21	D2_21	D1_21	D0_21
C16	D7_22	D6_22	D5_22	D4_22	D3_22	D2_22	D1_22	D0_22
C17	D7_23	D6_23	D5_23	D4_23	D3_23	D2_23	D1_23	D0_23
C18	D7_24	D6_24	D5_24	D4_24	D3_24	D2_24	D1_24	D0_24
C19	D7_25	D6_25	D5_25	D4_25	D3_25	D2_25	D1_25	D0_25
C20	F_26	D6_26	D5_26	D4_26	D3_26	D2_26	D1_26	D0_26
C21	F_27	D6_27	D5_27	D4_27	D3_27	D2_27	D1_27	D0_27
C22	F_28	D6_28	D5_28	D4_28	D3_28	D2_28	D1_28	D0_28
C23	F_29	D6_29	D5_29	D4_29	D3_29	D2_29	D1_29	D0_29
C24	F_30	D6_30	D5_30	D4_30	D3_30	D2_30	D1_30	D0_30
C25	F_31	D6_31	D5_31	D4_31	D3_31	D2_31	D1_31	D0_31
C26	F_32	D6_32	D5_32	D4_32	D3_32	D2_32	D1_32	D0_32
C27	F_33	D6_33	D5_33	D4_33	D3_33	D2_33	D1_33	D0_33
C28	F_34	D6_34	D5_34	D4_34	D3_34	D2_34	D1_34	D0_34
C29	F_35	D6_35	D5_35	D4_35	D3_35	D2_35	D1_35	D0_35
C30	F_36	D6_36	D5_36	D4_36	D3_36	D2_36	D1_36	D0_36
C31	F_37	D6_37	D5_37	D4_37	D3_37	D2_37	D1_37	D0_37
C32	F_38	D6_38	D5_38	D4_38	D3_38	D2_38	D1_38	D0_38
C33	F_39	D6_39	D5_39	D4_39	D3_39	D2_39	D1_39	D0_39
C34	F_40	D6_40	D5_40	D4_40	D3_40	D2_40	D1_40	D0_40
C35	F_41	D6_41	D5_41	D4_41	D3_41	D2_41	D1_41	D0_41
C36	F_42	D6_42	D5_42	D4_42	D3_42	D2_42	D1_42	D0_42
C37	F_43	D6_43	D5_43	D4_43	D3_43	D2_43	D1_43	D0_43
C38	F_44	D6_44	D5_44	D4_44	D3_44	D2_44	D1_44	D0_44
C39	F_45	D6_45	D5_45	D4_45	D3_45	D2_45	D1_45	D0_45

2.8 Stored Format of Packet 8/30 (001x)

F_j = Protection bit (result of hamming or parity check) of byte j (1 = failed)

Di_j = Data bit i of byte j

Column	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
C0	Reserved for internal use							
C1	0	0	F_7	0	D3_7	D2_7	D1_7	D0_7
C2	0	0	F_8	0	D3_8	D2_8	D1_8	D0_8
C3	0	0	F_9	0	D3_9	D2_9	D1_9	D0_9
C4	0	0	F_10	0	D3_10	D2_10	D1_10	D0_10
C5	0	0	F_11	0	D3_11	D2_11	D1_11	D0_11
C6	0	0	F_12	0	D3_12	D2_12	D1_12	D0_12
C7	0	0	F_13	0	D3_13	D2_13	D1_13	D0_13
C8	0	0	F_14	0	D3_14	D2_14	D1_14	D0_14
C9	0	0	F_15	0	D3_15	D2_15	D1_15	D0_15
C10	0	0	F_16	0	D3_16	D2_16	D1_16	D0_16
C11	0	0	F_17	0	D3_17	D2_17	D1_17	D0_17
C12	0	0	F_18	0	D3_18	D2_18	D1_18	D0_18
C13	0	0	F_19	0	D3_19	D2_19	D1_19	D0_19
C14	0	0	F_20	0	D3_20	D2_20	D1_20	D0_20
C15	0	0	F_21	0	D3_21	D2_21	D1_21	D0_21
C16	0	0	F_22	0	D3_22	D2_22	D1_22	D0_22
C17	0	0	F_23	0	D3_23	D2_23	D1_23	D0_23
C18	0	0	F_24	0	D3_24	D2_24	D1_24	D0_24
C19	0	0	F_25	0	D3_25	D2_25	D1_25	D0_25
C20	F_26	D6_26	D5_26	D4_26	D3_26	D2_26	D1_26	D0_26
C21	F_27	D6_27	D5_27	D4_27	D3_27	D2_27	D1_27	D0_27
C22	F_28	D6_28	D5_28	D4_28	D3_28	D2_28	D1_28	D0_28
C23	F_29	D6_29	D5_29	D4_29	D3_29	D2_29	D1_29	D0_29
C24	F_30	D6_30	D5_30	D4_30	D3_30	D2_30	D1_30	D0_30
C25	F_31	D6_31	D5_31	D4_31	D3_31	D2_31	D1_31	D0_31
C26	F_32	D6_32	D5_32	D4_32	D3_32	D2_32	D1_32	D0_32
C27	F_33	D6_33	D5_33	D4_33	D3_33	D2_33	D1_33	D0_33
C28	F_34	D6_34	D5_34	D4_34	D3_34	D2_34	D1_34	D0_34
C29	F_35	D6_35	D5_35	D4_35	D3_35	D2_35	D1_35	D0_35
C30	F_36	D6_36	D5_36	D4_36	D3_36	D2_36	D1_36	D0_36
C31	F_37	D6_37	D5_37	D4_37	D3_37	D2_37	D1_37	D0_37
C32	F_38	D6_38	D5_38	D4_38	D3_38	D2_38	D1_38	D0_38

Column	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
C33	F_39	D6_39	D5_39	D4_39	D3_39	D2_39	D1_39	D0_39
C34	F_40	D6_40	D5_40	D4_40	D3_40	D2_40	D1_40	D0_40
C35	F_41	D6_41	D5_41	D4_41	D3_41	D2_41	D1_41	D0_41
C36	F_42	D6_42	D5_42	D4_42	D3_42	D2_42	D1_42	D0_42
C37	F_43	D6_43	D5_43	D4_43	D3_43	D2_43	D1_43	D0_43
C38	F_44	D6_44	D5_44	D4_44	D3_44	D2_44	D1_44	D0_44
C39	F_45	D6_45	D5_45	D4_45	D3_45	D2_45	D1_45	D0_45

3 Glossary

ACQI

Acquisition interface. This is a serial/parallel converter with a consecutive FIFO which outputs byte-wise the input data to the PU.

BDM

Basic display memory. Three to five chapters of internal RAM memory used exclusively to store display characters and attributes for the inner screen area. BDM can be exchanged with EDM with the BES bit in the page position word (display control register).

Block

Internal memory segment of 1024 consecutive doublewords. The first doubleword of a block starts at row/column address 0/0.

Chapter

Internal or external memory of 1024 consecutive bytes aligned to row/column = 0/0.

Displayed page memory

Page memory which is actually converted by S/P-C into basic display memory.

Displayword (DPWORD)

Word length = 24 to 48 bits, depending on the selected display mask registers.

Doubleword

Word length = 48 bits.

EDM

Extended display memory. EDM can be exchanged with BDM with the BES bit in the page position word (display control register). Five chapters of internal RAM memory used exclusively to store display characters and attributes for the outer screen area.

IRAM5273.PDF

Program definition file in ASM5273 source code for PU programs. Defines memory mapping.

MEGATEXT controller

Microcontroller connected over M3L Bus with SDA 5273.

MCI

MEGATEXT command Interface. Commands for the PU can be given over the MCI. MCI comprises the M3L registers MCI_COMMAND, MCI0_5 to MCI3_1, the M3L-bit CMD_RUN and the interrupt signal at INTQ pin.

NPRP	Non-page-related packet.
OSD mode	Display shows only a text or graphic picture whose source normally is a microcontroller driven display generator.
Page_TLU	Table lookup used for PRQ groups of type 0. Has pointers to destination addresses for TTX pages.
Page memory	Internal or external memory segment reserved for storing TTX data.
Page request data	Page-related information bits as defined in the world system teletext specification.
Page request record	Complete set of page-related information bits.
PB	Packet buffer. The ACQI output data are written to PB by the PU.
PRQ	Page request bits or packet request bits.
PRQ_GROUP	Memory segments with different PRQ record priority. They describe PRQ_TYPE, FAP_CHAP and care bits for the PRQ records.
PU	Processing unit. On-chip SDA 5273 RISC processor.
S/P-C	Serial/parallel attribute conversion. Realized with firmware.
Teletext mode	Display shows only a teletext picture whose source is a teletext fed display generator.
Trace memory	Stores traced basic page numbers of the incoming teletext data and a flag which indicates the presence of related subpages.
VBI	Vertical blanking interval.
Video mode	Display shows only a video picture whose source normally is the monitor's video input (IF output, SCART, CVBS or RGB input).
Word	Word length = 24 bits .