

ICs for Consumer Electronics MEGATEXT

ACQ Reference

MEGATEXT® ACQ Reference	
Revision History: 10.94	
Previous Releases: 08.93, 02.94	
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 General Overview	5
2 Acquisition Groups	6
2.1 Type 0	7
2.2 Type 2	7
2.3 Type 3	8
3 Pseudopacket Support	9
3.1 Request of Pseudopackets	9
3.2 Pseudopacket Enable	10
3.3 Pseudopacket Trace	10
4 Subpage Support	11
4.1 Subpage Collect Mode	11
4.2 Subpage Collect Mode with Overwrite	11
4.3 Selected Subpage Mode	11
4.4 Subpage Don't Care Mode	11
4.5 Subpage Request with Type 2	11
5 Treatment of Transmission Errors	12
5.1 Programmable Reception Threshold	12
5.2 1-Byte Hamming Errors	13
5.3 Parity Errors	13
5.4 3-Byte Hamming Errors	13
6 Other Features	14
6.1 VPS Support	14
6.2 PAL/NTSC Switch	14
6.3 WSS Data Reception	14
6.4 Programmable Framing Code	17
6.5 Page Trace and Page Request Status Table	17
6.6 Acquisition Page Control Bits	20
6.6.1 Disable Clear	20
6.6.2 Hold Enable	20
6.6.3 Check Off	20
6.6.4 End-of-Page Interrupt	20
6.6.5 Page_Being_Received	21
6.6.6 Page_Being_Looked_For	21
6.6.7 First_Reception	21
6.6.8 Compress Basic TOP Tables	21
6.6.9 Page-Selective Pseudopacket Disable	22
6.6.10 Pseudopacket Memory Overflow Indicator	22
6.6.11 Subpage Buffer Overflow Indicator	22
6.6.12 Subpage Modes	22
6.6.13 Remove Protection	22
6.6.14 Check Modes	23

Contents (cont'd)	Page
7 TOP/FLOF Support	24
7.1 Request of TOP Tables	24
8 Memory Management System	25
9 Restrictions of System	29
9.1 Flyback Mode	29
10 How to Acquire Data	30
11 Feature Table	37

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 General Overview

This paper describes and specifies the features of acquisition in the MEGATEXT® SDA 5273. It is realized with the support of software routines integrated in the MEGATEXT processing unit. Acquisition can be controlled by M3L registers and MCI commands. For details of M3L bits and MCI commands refer to “M3L Bus Registers Programmer’s Reference” and “MCI Description”.

To convert the incoming serial datalines into parallel format, a controllable line buffer is used, whose output data are processed by software. By programming this interface you can differ between two modes:

625 line mode (PAL) with flyback buffer

525 line mode (NTSC) with flyback buffer

In flyback mode the contents of the line buffer are first copied into a larger buffer in the RAM, before the received data and the request records are evaluated. The design of the flyback buffer and the evaluation of data is completely asynchronous. The start address and the length of the buffer can be configured by the user. To obtain maximum acquisition performance the length of the buffer must be at least 18 lines.

MEGATEXT acquisition routines have highest priority, so that other tasks have to wait. Depending on the acquisition parameters and the transmitted teletext data, the number of M3L Bus accesses is dynamically limited. It is absolutely necessary to consider the “break function” of the M3L Bus (ref. specification of the M3L Bus).

As the integrated memory manager calculates the physical memory address of a page, the user only has to “think in terms of page numbers” and not memory addresses.

For convenient evaluation of TOP pages several MCI commands are available.

2 Acquisition Groups

The number of page request circuits in hardware acquisition in former TTX processors was limited to eight. With these circuits it was possible to search eight pages in parallel. After reception, these circuits must be programmed again, to request other pages. The simultaneous search of non-page-related data and pages was not possible.

In the SDA 5273 software system the eight acquisition circuits are replaced by eight acquisition groups. An acquisition group contains two parameters: the search type and a do care mask. The do care mask of a group is valid for every request record of this group. A different number of different page request records may be attached to each group. So the number of pages which can be searched in parallel in one cycle is not limited to eight. The performance of the acquisition depends on the use of the search types and is in the range from 140 to 2048 "pages".

The sequence of the groups, their search types and the sequence in which the requests of a user page are programmed determines the priority for searching a page. This means that first a page is compared with the records of group 0. If it does not match, the page will be searched in the next group. Each page is stored once only.

Mixed use of search types is allowed (mixed mode acquisition). So it is possible to search non-page-related data, normal pages and TOP pages at the same.

2.1 Type 0

Type 0 is reserved for requesting pages and subpages without looking to the header control bits (C14 – C4). With this type it is possible to search all basic pages of all magazines (2048). This feature is realized by building an address table (table look up). The administration of this table and its addresses is done by the memory manager. The user only has to reserve memory space for it (4 Kbytes). Type 0 must be applied if in a big memory a large number of chapters should be requested. It is also recommended for smaller memories, if you want to save CPU time for other tasks than acquisition (i.e. display features, graphics, M3L access).

Using this search type you cannot determine the do care mask of the record because it is fixed to the bits belonging to magazine and page number.

The search of subpages can be done by page-related control parameters (ref. subpage support, acquisition page control bits). The definition of search type 0 is allowed once only.

You can add a page to this group with the MCI command “ADD_PAGE” (ref. MCI Description).

2.2 Type 2

If you have special demands on the do care mask of a page request record, use this search type. Each bit of the mask can be programmed independently. If you use this group type exclusively (eight times) and attach only one record to every group, you will be able to emulate the hardware acquisition of SDA 5243 for example.

The number of pages to be requested with this type depends on use of the other types and is also limited. It should be used only if its do care mask differs from the default masks of type 0.

Requests can be made by the MCI command “ADD_PAGE”.

In contrast to type 0, subpages are not related firmly to the corresponding basicpages. The request of subpages must be realized by programming the do care mask and the request record in an appropriate way.

2.3 Type 3

Type 3 is reserved for acquiring non-page-related and non-magazine-related data. This type can be used more than once and can be combined with every other type. The storage of each packet consumes 80 bytes of memory. Data, that are already defined in the world system teletext standard, are checked before storing. These are:

Packets: X/29 with designation codes: 0000, 0001, 0100

Packets: 8/30 with designation codes: X000, X100

The number of packets to be requested with this type depends on the use of the other types and is limited (ref. system restrictions). The request of these packets can be achieved by using the command "ADD_PACKET_29_30".

The table below gives you an overview of the **internal** use of the do care bits for different search types. Do not forget that you cannot change the do care mask for type 0.

Type 0	0	0	0	0	0	M2	M1	M0
Type 2	0	0	0	0	0	M2	M1	M0
Type 3	R4	R3	R2	R1	R0	M2	M1	M0
Type 0	PT3	PT2	PT1	PT0	PU3	PU2	PU1	PU0
Type 2	PT3	PT2	PT1	PT0	PU3	PU2	PU1	PU0
Type 3	0	0	0	0	D3	D2	D1	D0
Type 0 (Subpage mode)	0	0	0	0	0	0	0	0
	0	MT2	MT1	MT0	MU3	MU2	MU1	MU0
Type 2	C4	MT2	MT1	MT0	MU3	MU2	MU1	MU0
Type 3	0	0	0	0	0	0	0	0
Type 0 (Subpage mode)	0	0	0	0	0	0	0	0
	0	0	HT1	HT0	HU3	HU2	HU1	HU0
Type 2	C6	C5	HT1	HT0	HU3	HU2	HU1	HU0
Type 3	0	0	0	0	0	0	0	0
Type 0	0	0	0	0	0	0	0	0
Type 2	C14	C13	C12	C11	C10	C9	C8	C7
Type 3	0	0	0	0	0	0	0	0

3 Pseudopacket Support

3.1 Request of Pseudopackets

Page-related pseudopackets (X/25, X/26, X/27, X/28) are stored in a special data section, the so-called P40 chain. The request of these packets can be controlled in general (for all pages) and for each page separately.

The allocation of memory is managed by the acquisition and depends on the transmitted data. The user only has to define the memory segment in which the pseudopackets of all received pages should be stored (MCI command "CREATE_FREE_P40_CHAIN").

With this dynamic memory allocation feature a lot of memory space is saved, because the acquisition consumes only memory for those packets that are really transmitted. The storage of one of these packets consumes 40 bytes.

The allocation of memory for magazine-related data (X/29) and non-page-related data (8/30) is done by using the command "ADD_PACKET_29_30". This command allocates 80 bytes of memory and requests the packet you want to store. The command "CREATE_FREE_P80_CHAIN" initializes the memory segment in which all of these packets are stored.

For removing and searching single pseudopackets several commands are available (ref. MCI Description). Because of the dynamic memory allocation it is possible that the memory space for the P40 chain may be exceeded. In this case an interrupt is given to the user. With the command "GET_FREE_CHAP_CHAIN" the memory space of the P40 chain can be increased again. If you remove a page, all related pseudopackets will be removed too.

All packets (except X/30 and X/31) are error checked and compressed before storing. The format of the stored pseudopacket data is explained in "MCI Description".

3.2 Pseudopacket Enable

With the M3L register “PSEUDO_PACKET_ENABLE” it is possible to enable or disable certain page-related pseudopackets for all pages.

0	0	0	0	EN_X_28	EN_X_27	EN_X_26	EN_X_25
---	---	---	---	---------	---------	---------	---------

EN_X_28 ... EN_X_25

1: Acquisition of associated packet enabled

0: Acquisition of associated packet disabled

The page selective pseudopacket enable is controlled by the “DIS_PP” bit in the acquisition control byte (see below).

3.3 Pseudopacket Trace

For every received pseudopacket number a flag is set in a RAM register.

Use command “READ_PSEUDO_PACKET_TRACE” to obtain the stored information. It is possible to reset the trace register automatically after reading it.

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

PPT_X_31 ... PPT_X_25

Each bit is associated with the pseudopacket number given in its symbol.

1: Pseudopacket received

0: Pseudopacket not received

4 Subpage Support

Basic pages are subpages with subpage code 0000_H, subpages are pages with any other subpage code. If search type 0 is used, the following four convenient subpage modes are supported by firmware. They can be set for every basic page number independently.

4.1 Subpage Collect Mode

In this mode the acquisition allocates memory space for as many chapters as subpages are transmitted. The allocation is done whenever a subpage is received that is not yet stored.

The number of subpages can be limited by setting a parameter (S_MAX). If there are more subpages transmitted than provided by S_MAX, a subpage buffer overflow interrupt is given to the user.

In this mode the user does not need to know which subpage numbers are transmitted in order to store them. The acquisition control bits of the corresponding basic page are copied automatically to the request record of all related subpages.

This mode can be chosen by use of acquisition control bits in the “ADD_PAGE” command.

4.2 Subpage Collect Mode with Overwrite

This mode is nearly the same as the one just described. The only difference is that in case of subpage buffer overflow, the buffer will be overwritten starting with the oldest subpage. In this case the user will not get an interrupt.

This mode can be chosen by use of acquisition control bits in the “ADD_PAGE” command.

4.3 Selected Subpage Mode

By use of MCI command “ADD_PAGE” the user can request a page with a specific subcode.

4.4 Subpage Don't Care Mode

The user does not want to reserve more than one chapter for acquisition of subpages. All received subpages (with the same basic page number) are now stored in the same chapter (running through subpages).

This mode can be chosen by use of acquisition control bits in the “ADD_PAGE” command.

4.5 Subpage Request with Type 2

You can of course request a subpage with the user-defined request and a corresponding do care mask (type 2). In this case the subpage is not related to the basic page and the modes just described are not supported.

The subpage is treated in the same way as any other basic page.

5 Treatment of Transmission Errors

All data received are checked online by acquisition before storing them. To obtain the highest signal quality, it has to be decided when, where and which kind of checks have to be done.

5.1 Programmable Reception Threshold

Using the M3L register "RECEPTION_THRESHOLD" you can program the checks to be done for magazine and row number of each packet. Furthermore you can decide whether a 1-bit framing code error should be accepted or not. The acquisition discards packets with bit errors relating to the reception-threshold register bits.

HCBCHECK	ACQRHEAD	ACQRMGRW	FCERR_1	MAGERR_2	MAGERR_1	ROWERR_2	ROWERR_1
----------	----------	----------	---------	----------	----------	----------	----------

HCBCHECK = 1 Header_control_byte_check

The last two hamming coded header bytes will not be hamming checked.

HCBCHECK = 0 The last two hamming coded header bytes will be hamming checked.

ACQRHEAD = 1 ACQ_reset header related

If one not correctible hamming error is detected in any hamming coded header byte, the packet will be rejected and the ACQ will be reset. Reset of ACQ means: reset active header IDs and set end of page condition in the active destination chapter.

ACQRHEAD = 0 If one not correctible hamming error is detected in any hamming coded header byte, the packet will be rejected without ACQ reset.

ACQRMGRW = 1 ACQ_reset magazine and row related

If one not correctible hamming error is detected in the magazine or row number byte, the packet will be rejected and the ACQ will be reset. Reset of ACQ means: reset active header IDs and set end of page condition in the active destination chapter.

ACQRMGRW = 0 If one not correctible hamming error is detected in the magazine or row number byte, the packet will be rejected without ACQ reset.

FCERR_1 = 1 The packet received will not be stored if a 1-bit framing code error is recognized.

FCERR_1 = 0 A 1 bit framing code error is not considered.

MAGERR_2 = 1 The received packet will not be stored if there is a non-correctible error in the magazine number.

MAGERR_2 = 0 A multiple error in the magazine number is not considered.

MAGERR_1 = 1 The received packet will not be stored if a correctible 1-bit error is recognized in the magazine number.

MAGERR_1 = 0 A 1-bit error in the magazine number is not considered.

ROWERR_2 = 1 The received packet will not be stored if there is a non-correctible error in the row number.

ROWERR_2 = 0 A multiple error in the row number is not considered.

ROW_ERR_1 = 1 The received packet will not be stored if a correctible 1-bit error is recognized in the row number.

ROW_ERR_1 = 0 A 1-bit error in the row number is not considered.

5.2 1-Byte Hamming Errors

Depending on the reception threshold the packet received will not be stored if a non-correctible hamming error occurs in byte 6 – 13 (page number, subcode, control bits) of any header.

The same will happen if a non-correctible error is detected in byte 6 or 7 of any packet and in the designation code of a pseudopacket.

All other hamming checked data bytes are compressed to 4 data bits and one FALSE bit. If a page or a pseudopacket is received for the second time, wrong bytes will not be stored.

As long as they are not described differently, all 1-byte hamming checked data will be stored in the following format:

0	0	FALSE	0	D4	D3	D2	D1
---	---	-------	---	----	----	----	----

FALSE = 1 This byte contains a non-correctible error.

FALSE = 0 This byte is received and stored correctly.

D4, D3, D2, D1 Compressed data bits

5.3 Parity Errors

During first reception of a packet or a page, parity protected bytes will be discarded if an error is detected. Instead of the received data byte, a blank (20 hex) is stored. If the data are received for the second time and an error occurs, these bytes will not be stored.

As long as they are not described differently, all parity checked data will be stored in the following format:

FALSE	D7	D6	D5	D4	D3	D2	D1
-------	----	----	----	----	----	----	----

FALSE = 1 This byte contains a non-correctible error.

FALSE = 0 This byte is received and stored correctly.

D7 – D1 Compressed data bits

5.4 3-Byte Hamming Errors

3-byte hamming protected bytes are also checked before storage. To obtain improved quality of the received data, they will not be stored if they are received for the second time.

The format of the stored 3-byte hamming groups can be seen in “MCI Description”.

6 Other Features

6.1 VPS Support

The reception of VPS data is possible from datalines programmed by the M3L register “SINGLE_DATA_LINE”. The framing code for VPS reception is fixed to 8A99_H (equ. to BAH biphasic decoded). The “biphase” coded data are decoded and compressed by acquisition and stored in internal RAM. If there is any error in any data byte of the VPS line, the whole line will be discarded. All 13 bytes (decoded VPS words) can be read from the internal MTX RAM (see “MCI Description”).

6.2 PAL/NTSC Switch

In flyback mode it is possible to receive data transmitted in 525-line mode as well as in 625-line mode. Switching between these standards can be done by setting bit “ACQ_NTSC” in M3L register R 109.

6.3 WSS Data Reception

- WSS is short for Wide Screen Signaling. WSS is introduced with PAL-PLUS and contains data describing the type of transmission with respect to display format.
- WSS reception is done in the “**single line mode**” like VPS.
- To enable VPS or WSS reception the VPS mode has to be switch on. For more details please refer to the MCI description “slicer control command”.
- To switch over from VPS to WSS reception a new bit “WSS” was introduced in R98.

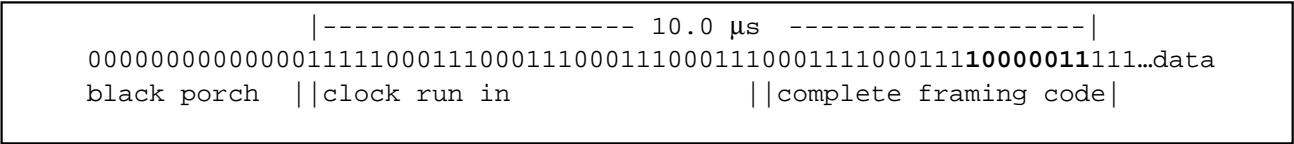
Mode	WSS (R98, D5)	SDL (R98, D4 ... D0)
VPS + PAL	0	16 _d
WSS + PAL	1	23 _d

As transmission line for WSS is specified line 23, for VPS line 16. The single data line number (SDL) has to be adopted respectively.

- In WSS mode the following bit pattern for **framing code** detection is used:

first “10000011” last

This is a reduced framing code compared to the WSS specification. Each 1 or 0 represents a period of 200 ns.



- The start and stop values of the single **line framing code** window in R100 are calculated as follows:

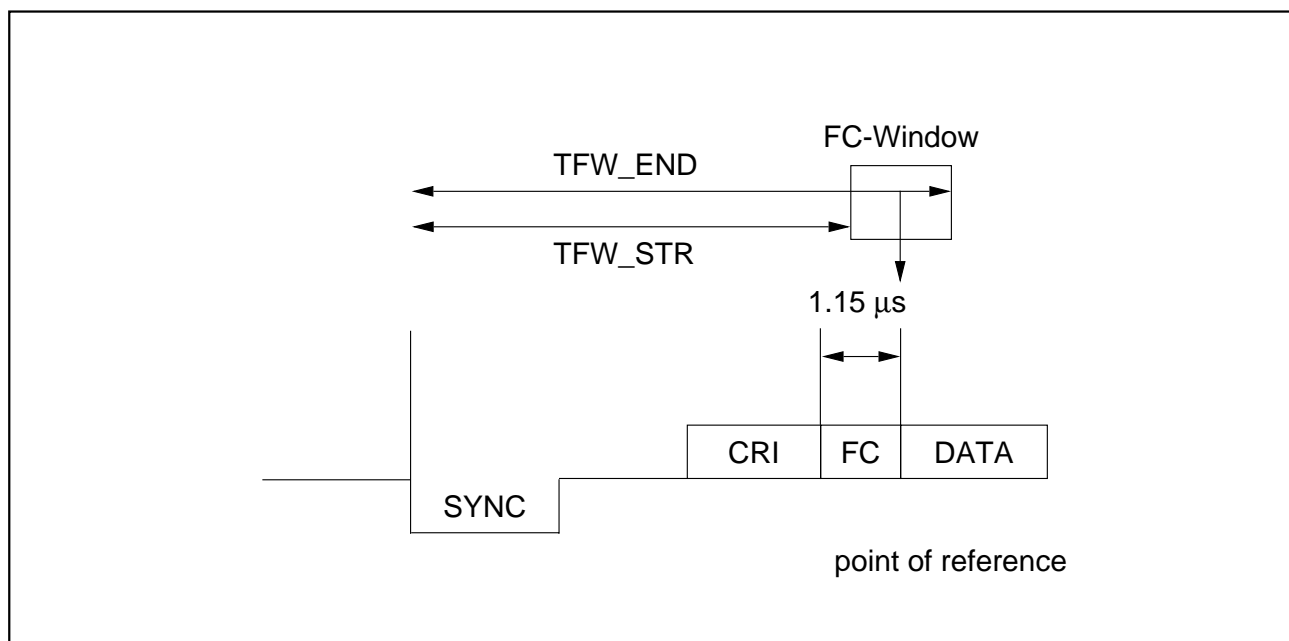
$$TFW_STR = 10.75 \mu s + XFW_STR [3:0] \times 0.66 \mu s$$

$$TFW_END = 10.3 \mu s + XFW_END [3:0] \times 0.66 \mu s$$

The timing references are the falling sync edge and the end of the framing code.

recommended value for R100 : F8_H in WSS mode

F0_H in VPS mode



● Data Evaluation

The data evaluation has to be done by software. For real time processing it is recommended to use downloadable Megatext-RISC software instead of software control by an external μC via serial bus. For downloadable firmware please contact:

Siemens Semiconductor, marketing department for consumer electronics

● Tasks of data evaluation:

In case of transmission errors caused by signal distortion (e.g. VCR replay) it is recommended to apply error correction algorithms:

- The majority of the 6 bits may decide about the result.
(010110) = 1 + 0 + 1 + 1 + 1 + 0 = 4 > 3 = Highlevel
- The data of three subsequently received data lines should have the same result.
- Interrupt generation in case of changes
- etc.

- format of stored WSS data:

Block/Byte/ Row/Col.		MSB							LSB
0/1/0/0		/B01 ₂	/B01 ₁	B01 ₃	B01 ₂	B01 ₁	1	1	1
0/1/0/1		B03 ₁	/B02 ₃	/B02 ₂	/B02 ₁	B02 ₃	B02 ₂	B02 ₁	/B01 ₃
0/1/0/2		B04 ₃	B04 ₂	B04 ₁	/B03 ₃	/B03 ₂	/B03 ₁	B03 ₃	B03 ₂
0/1/0/3		/B05 ₂	/B05 ₁	B05 ₃	B05 ₂	B05 ₁	/B04 ₃	/B04 ₂	/B04 ₁
0/1/0/4		B07 ₁	/B06 ₃	/B06 ₂	/B06 ₁	B06 ₃	B06 ₂	B06 ₁	/B05 ₃
0/1/0/5		B08 ₃	B08 ₂	B08 ₁	/B07 ₃	/B07 ₂	/B07 ₁	B07 ₃	B07 ₂
0/1/0/6		/B09 ₂	/B09 ₁	B09 ₃	B09 ₂	B09 ₁	/B08 ₃	/B08 ₂	/B08 ₁
0/1/0/7		B11 ₁	/B10 ₃	/B10 ₂	/B10 ₁	B10 ₃	B10 ₂	B10 ₁	/B09 ₃
0/1/0/8		B12 ₃	B12 ₂	B12 ₁	/B11 ₃	/B11 ₂	/B11 ₁	B11 ₃	B11 ₂
0/1/0/9		/B13 ₂	/B13 ₁	B13 ₃	B13 ₂	B13 ₁	/B12 ₃	/B12 ₂	/B12 ₁
0/1/0/10			/B14 ₃	/B14 ₂	/B14 ₁	B14 ₃	B14 ₂	B14 ₁	/B13 ₃

MEMO:

- VPS and WSS reception is not possible at the same time.
- Reception of PAL WST is possible parallel to WSS or VPS.
- The external software may use algorithms to compensate reception errors.

Example

WSS data:

1011 16:9 centre
0001 film mode
000 no subtitles
000 reserved

transmitted data sequence (first - last):

1 1 0 1 1 0 0 0 0 0 0 0 0

transmitted 200 ns pulses:

1111100011100011100011100011100011110001111000001111... (clock run in & framing code)
...111000111000000111111000111000000111000111000111000... (data)
...111000111000111000111000111000111

WSS data in MEGATEXT Memory (assumption: no transmission error):

Block/Byte I/ Row/Col.	HEX	Binary							
		MSB				LSB			
0/1/0/0	3F	0	0	1	1	1	1	1	1
0/1/0/1	0E	0	0	0	0	1	1	1	0
0/1/0/2	FC	1	1	1	1	1	1	0	0
0/1/0/3	38	0	0	1	1	1	0	0	0
0/1/0/4	70	0	1	1	1	0	0	0	0
0/1/0/5	1C	0	0	0	1	1	1	0	0
0/1/0/6	C7	1	1	0	0	0	1	1	1
0/1/0/7	71	0	1	1	1	0	0	0	1
0/1/0/8	1C	0	0	0	1	1	1	0	0
0/1/0/9	C7	1	1	0	0	0	1	1	1
0/1/0/10	71		1	1	1	0	0	0	1

6.4 Programmable Framing Code

The framing code for teletext packets is programmable with the MCI command “ACQ_CONTROL”. For programming of the data-entry window refer to the timing features described in “M3L-Bus Registers Programmer’s Reference”.

6.5 Page Trace and Page Request Status Table

Page trace must be enabled by the user with the MCI command “ACQ_CONTROL”. After enabling the page trace the page trace memory is updated with every received header. The position of the memory can be programmed with the M3L register “PAGE_TRACE_REGISTER”. The following information is stored in the page trace table: basic pages which are transmitted by broadcast and transmitted basic pages which contain subpages.

The trace bits are stored in **binary**, increasing RAM addresses.

The **Page Request Status Table (PRST)** is used by the command “REQUEST_ALL_PAGES”. The PRST is divided into two parts.

The first part (address 512 – 767) reflects all pages whose request is in progress. For each page with subcode 0000_H 1 bit is reserved. These bits can optionally be set with the command “ADD_PAGE” and reset with “REMOVE_PAGE”.

In the second part (address 768 – 1023) all pages whose request is intended can be marked. This could be done by simple M3L-Bus transfer or by the command “READ_CLEAR_PAGE_TRACE” or with the very convenient command “ADD_ALL_PAGES” which evaluates the TOP tables or the PAGE TRACE (see “MCI Description”).

The command “REQUEST_ALL_PAGES” evaluates these two parts of PRST. 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.

START_OF_PTT:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P (8/07)	P (8/06)	P (8/05)	P (8/04)	P (8/03)	P (8/02)	P (8/01)	P (8/00)
:	:	:	:	:	:	:	:
P (8/FF)	P (8/FE)	P (8/FD)	P (8/FC)	P (8/FB)	P (8/FA)	P (8/F)	P (8/F8)
P (1/07)	P (1/06)	P (1/05)	P (1/04)	P (1/03)	P (1/02)	P (1/01)	P (1/00)
:	:	:	:	:	:	:	:
P (1/FF)	P (1/FE)	P (1/FD)	P (1/FC)	P (1/FB)	P (1/FA)	P (1/F)	P (1/F8)
P (2/07)	P (2/06)	P (2/05)	P (2/04)	P (2/03)	P (2/02)	P (2/01)	P (2/00)
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
P (7/FF)	P (7/FE)	P (7/FD)	P (7/FC)	P (7/FB)	P (7/FA)	P (7/F9)	P (7/F8)

START_OF_PTT + 256:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SP (8/07)	SP (8/06)	SP (8/05)	SP (8/04)	SP (8/03)	SP (8/02)	SP (8/01)	SP (8/00)
:	:	:	:	:	:	:	:
SP (8/FF)	SP (8/FE)	SP (8/FD)	SP (8/FC)	SP (8/FB)	SP (8/FA)	SP (8/F)	SP (8/F8)
SP (1/07)	SP (1/06)	SP (1/05)	SP (1/04)	SP (1/03)	SP (1/02)	SP (1/01)	SP (1/00)
:	:	:	:	:	:	:	:
SP (1/FF)	SP (1/FE)	SP (1/FD)	SP (1/FC)	SP (1/FB)	SP (1/FA)	SP (1/F)	SP (1/F8)
SP (2/07)	SP (2/06)	SP (2/05)	SP (2/04)	SP (2/03)	SP (2/02)	SP (2/01)	SP (2/00)
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
SP (7/FF)	SP (7/FE)	SP (7/FD)	SP (7/FC)	SP (7/FB)	SP (7/FA)	SP (7/F9)	SP (7/F8)

START_OF_PTT + 512:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RIP (8/07)	RIP (8/06)	RIP (8/05)	RIP (8/04)	RIP (8/03)	RIP (8/02)	RIP (8/01)	RIP (8/00)
:	:	:	:	:	:	:	:
RIP (8/FF)	RIP (8/FE)	RIP (8/FD)	RIP (8/FC)	RIP (8/FB)	RIP (8/FA)	RIP (8/F)	RIP (8/F8)
RIP (1/07)	RIP (1/06)	RIP (1/05)	RIP (1/04)	RIP (1/03)	RIP (1/02)	RIP (1/01)	RIP (1/00)
:	:	:	:	:	:	:	:
RIP (1/FF)	RIP (1/FE)	RIP (1/FD)	RIP (1/FC)	RIP (1/FB)	RIP (1/FA)	RIP (1/F)	RIP (1/F8)
RIP (2/07)	RIP (2/06)	RIP (2/05)	RIP (2/04)	RIP (2/03)	RIP (2/02)	RIP (2/01)	RIP (2/00)
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
RIP (7/FF)	RIP (7/FE)	RIP (7/FD)	RIP (7/FC)	RIP (7/FB)	RIP (7/FA)	RIP (7/F9)	RIP (7/F8)

START_OF_PTT + 768:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RI (8/07)	RI (8/06)	RI (8/05)	RI (8/04)	RI (8/03)	RI (8/02)	RI (8/01)	RI (8/00)
:	:	:	:	:	:	:	:
RI (8/FF)	RI (8/FE)	RI (8/FD)	RI (8/FC)	RI (8/FB)	RI (8/FA)	RI (8/F)	RI (8/F8)
RI (1/07)	RI (1/06)	RI (1/05)	RI (1/04)	RI (1/03)	RI (1/02)	RI (1/01)	RI (1/00)
:	:	:	:	:	:	:	:
RI (1/FF)	RI (1/FE)	RI (1/FD)	RI (1/FC)	RI (1/FB)	RI (1/FA)	RI (1/F)	RI (1/F8)
RI (2/07)	RI (2/06)	RI (2/05)	RI (2/04)	RI (2/03)	RI (2/02)	RI (2/01)	RI (2/00)
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
RI (7/FF)	RI (7/FE)	RI (7/FD)	RI (7/FC)	RI (7/FB)	RI (7/FA)	RI (7/F9)	RI (7/F8)

P (i/j) = 1 Page j of magazine i occurred

SP (i/j) = 1 Any subpage to page number j of magazine i occurred

RIP (i/j) = 1 Request of page j of magazine i already done

RI (i/j) = 1 Request of page j of magazine i intended

6.6 Acquisition Page Control Bits

Acquisition page control bits can be written by use of the MCI commands “ADD_PAGE” and “ADD_PACKET_29_30”. They are used for page selective control of acquisition features and must not be mixed up with the bits in M3L register “ACQUISITION_CONTROL”, which are applicable for the control of acquisition in general.

Remember that if any subpage collect mode is chosen, all stored subpages will automatically obtain the same page control bits as the corresponding basic page. In other words, if you program an end-of-page interrupt request to the basic page, all its subpages will also produce an interrupt.

Not all control bits are available in each mode and by each command. Moreover, some of them must always be set to 0 or 1 by the user. The following table gives you an overview of the use.

MCI Command	Acquisition Page Control Bits							
ADD_PAGE	DCR	HEN	0	IREQ	PBR = 0	PBLF = 1	FIRST = 0	C_BTT
ADD_PACKET_29_31	0	HEN	CHECK_OFF	IREQ	x	PBLF = 1	x	0
ADD_PAGE	DIS_PP	P_OV_OC = 0	OV_OCCU = 0	SM_1	SM_0	REM_PROT	CHECK_1	CHECK_0
ADD_PACKET_29_31	x	x	x	x	x	x	x	x

6.6.1 Disable Clear

DCR = 1 Disable Clear
DCR = 0 Clear before first reception

If this bit is set to 0, the corresponding page will be cleared automatically before first reception.

6.6.2 Hold Enable

HEN = 1 Automatic hold
HEN = 0 No automatic hold

If this bit is set, the corresponding page (or non-page-related packet) will be stored once only.

6.6.3 Check Off

CHECK_OFF = 1 The check of non-page-related pseudopackets is done as described in the WSTN.
CHECK_OFF = 0 The automatic check of non-page-related pseudo packets is switched off.

6.6.4 End-of-Page Interrupt

IREQ = 1 EOP interrupt requested
IREQ = 0 EOP interrupt not requested

If this bit is set, an interrupt “end of page” will be given to the external controller as soon as the end-of-page condition is recognized.

It is not useful to request an interrupt for every requested page, because it is possible to request up to 2048 pages at the same time.

The end-of-page condition will occur if a page with the same magazine number and a different page number (including the subpage code) is transmitted.

If non-page-related data are requested, the interrupt will be given as soon as the request packet is found.

The following table gives you an overview for interpretation of the bits PBLF, PBR and FIRST.

PBLF	PBR	FIRST	EVENT
1	0	0	A new page request is given and the header is still not found.
0	1	1	The requested header is found for the first time. As long as PBR = 1, no end-of-page condition occurred.
0	0	1	An end-of-page condition occurs for the first time.
0	1	0	The requested header is found for the second time. As long as PBR = 1, the page is being received.
0	0	0	An end-of-page condition occurs for the second time.

6.6.5 Page_Being_Received

PBR = 1 Page being received

PBR = 0 Page not being received

This bit is a status bit of the acquisition. As long as it is set to 1, the requested page is received. It has always to be set to 0 when a new page is requested.

6.6.6 Page_Being_Looked_For

PBLF = 1 Page being looked for

PBLF = 0 Header found

This bit has to be set to 1 whenever a new page request record is programmed. It is reset by acquisition as soon as the header of this page is found. It can also be an indicator whether a non-page-related packet is found.

6.6.7 First_Reception

FIRST = 1 Page being received for the first time

This bit is an internal status bit and has to be reset by the user after giving a new request.

6.6.8 Compress Basic TOP Tables

C_BTT = 1 Compression of basic TOP tables on

C_BTT = 0 Compression of basic TOP tables off

If this bit is set and parallel magazine mode is given by broadcast (c8), the acquisition will interpret the received page as a basic TOP table (do not forget to set the correct check bits (see below)). The link information will be copied in a chapter, which is indicated in the M3L register "TOP_COM_BAS". The format of the link information can be seen in "MCI Description".

6.6.9 Page-Selective Pseudopacket Disable

DIS_PP = 1 Storage of pseudopackets to this page is disabled

DIS_PP = 0 Storage of pseudopackets to this page is enabled

With this bit it is possible to disable the reception of pseudopackets X/25, X/26, X/27 and X/28 for each page (subpage) separately.

6.6.10 Pseudopacket Memory Overflow Indicator

P_OV_OC = 1 Pseudopacket memory overflow occurred

This bit will be set by acquisition if a memory overflow occurs while allocating memory for pseudopacket storage.

With the command "GET_FREE_CHAP" another chapter can be attached to the p40 chain.

It has to be reset by the user after evaluating it.

6.6.11 Subpage Buffer Overflow Indicator

OV_OCCU = 1 Subpage buffer memory overflow occurred

This bit will be set by acquisition if the number of transmitted subpages in subpage collect mode is larger than S_MAX. In this case an interrupt is given to the external controller.

6.6.12 Subpage Modes

The four different subpage modes can be adjusted by using the bits SM1, SM0 and S_MAX (7:0).

SM_1	SM_0	S_MAX	Mode
0	0	S_MAX = 0	Not defined
		0 < S_MAX < 255	Subpage collect mode without overwrite and limited buffer size
		S_MAX = 255	Subpage collect mode with unlimited buffer size
0	1	S_MAX = 0	Not defined
		0 < S_MAX < 255	Subpage collect mode with overwrite and limited buffer size
		S_MAX = 255	Subpage collect mode with unlimited buffer size
1	0	XXX	Selected subpage mode
1	1	XXX	Subpage don't care mode

6.6.13 Remove Protection

REM_PROT = 1 This page is remove protected and can only be removed if the appropriate bit in the command "REMOVE_PAGE" is set.

REM_PROT = 0 The remove protection is switched off.

6.6.14 Check Modes

The check bits indicate the checks to be done by acquisition.

CHECK_1	CHECK_0	Mode
0	0	Normal page check: Header bytes 6 to 13 are 1 byte hamming checked. All other bytes of the page are parity checked.
0	1	No parity check for normal page: Header bytes 6 to 13 are 1 byte hamming checked. All other bytes of the page are stored unchecked.
1	0	1-byte hamming check (TOP page): Header bytes 6 to 13 are 1 byte hamming checked. Header bytes 14 to 45 are parity checked. All bytes in packet 1 – 22 are 1 byte hamming checked.
1	1	Mixed 1-byte hamming check: Header bytes 6 to 13 are 1 byte hamming checked. Header bytes 14 to 45 are parity checked. Bytes 0 – 7 and 20 – 27 in packet 1 – 22 are 1 byte hamming checked. Bytes 8 – 19 and 28 – 39 in packet 1 – 22 are parity checked.

7 TOP/FLOF Support

7.1 Request of TOP Tables

The request of Basic TOP Tables can be realized by programming the user-defined-group with the correct page number (i.e. "1F0/3fxx_H"). In contrast to former systems, it is possible to control the checks for each page separately. If the bits CHECK_(1:0) in the Acquisition Page Control byte are set to "010", the Basic Top Table will be received and checked in the accurate way. The Additional TOP Table, the Multipage TOP Table and the Multipage Extension Table can be received in a similar way. With the input parameters of the command "ACQ_CONTROL" it is possible to determine whether the request of all the TOP management pages should be done automatically or not.

If the do care bits of the magazine number of the Basic TOP Page are set to 0, the pages of all magazines will be mixed together. However, if the Compress TOP Table bit (C_BTT) is set, the acquisition will look for the c8 bit in the header of the page (parallel magazine mode). If it is set, the Basic TOP Tables of all magazines will be compressed to two pages. The Page Linking Table and Basic TOP Table list of all magazines will be stored in a special chapter in the IAT/XAT section. The user is responsible for allocation of memory for this feature.

For convenient evaluation of these two pages several MCI commands are available. These commands always assume that the BTT is requested as the first page in ACQ group 6 and all other linked TOP pages are located in group 5. For detailed information see "MCI Description".

With the command "EXECUTE_FLOF" it is possible to read the linked pages (in X/27/0000) of a given page number and/or request these pages and/or change the display chapter and/or remove the unused linked pages of the next highest FLOF level. For detailed information see also "MCI Description".

8 Memory Management System

The dynamic allocation of memory space is supported by firmware. The memory (internal and external) is divided into four different parts. The user-specific part includes data for system administration and temporary memory for the firmware. The remaining three parts will be used for data storage. These sections are split because of different record sizes (see below). The user is responsible for programming the limits of these segments correctly.

The user-specific segment covers the internal memory and the first 24 chapters of external memory at the most. By use of a memory allocation table (M3L register "IAT, XAT") each of these chapters can be attached to this memory segment and excluded for data storage. The allocation table below shows you an **example** of how to program the register "IAT, XAT". The compulsory chapters must not be used for other purposes. If you do not allocate chapters which are marked "optional", you will not be able to use the corresponding feature. The start address of the flyback buffer, the page trace table, the compressed TOP table and the p80 chain can be programmed with the corresponding M3L registers.

Data memory is organized with so-called "chained lists". We distinguish between list elements with three different fixed record sizes: 40 bytes, 80 bytes and 1 Kbyte. All elements are connected with pointers which point to the next element. After power-on all lists must be initialized with a "CREATE" command. This command sets up a list with empty elements only. They can be filled with data by using an "ADD" command and be erased with a "REMOVE" command. Use "SEARCH" to obtain the physical memory address of the data you stored with the "ADD" command.

The "CREATE_FREE_CHAP_CHAIN" command initializes the memory segment which should be used for storage of pages and subpages. With this command memory is only reserved for a certain number of pages and subpages, but no page request is started yet. The request of a page could be done with an "ADD" command (i.e. "ADD_PAGE"). After using the "CREATE" command all old data are destroyed. Before using this command it is necessary to write the M3L register "FFP_CHAP". This register contains the start address "FFP" (First Free Pointer) of the "chap chain" and the number of free chapters in the chain. The start address of the chain has to point to a chapter which is not allocated by the "IAT/XAT" register. The allocation of memory starts with "FFP" and continues with increasing chapter number. The sequence of the chapters is fixed in the following way: internal RAM block 0/byte 0, byte 1 ... , block 1, block 2, block 3, external RAM chapter 0, chapter 1 ... etc. If there are blanks in the "IAT/XAT" register, the equivalent chapters will be filled with teletext data after switching on the acquisition. Allocated chapters are skipped.

The "CREATE_FREE_P40_CHAIN" command initializes the memory segment which should be used for pseudopacket acquisition. The main difference between this command and "CREATE_FREE_CHAP_CHAIN" is in the record size, it is only 40 bytes. The M3L register "FFP_P40" contains the start pointer of the first free element and the total number of free records. The allocation of memory starts with "FFP" and continues with increasing row number (row-column addressing scheme). Row 25 of a chapter, which contains only 24 bytes, is skipped. After reaching a chapter border (row 24) the next chapter will be used for pseudopacket allocation.

The “CREATE_FREE_P80_CHAIN” command initializes the memory segment which should be used for acquisition of non-page-related and non-magazine-related data (single-packet acquisition). The record size is fixed at 80 bytes. The first 40 bytes are reserved for data storage, the second 40 bytes contain the pointers to the next element. The M3L register “FFP_P80” contains the start pointer to the first free element and the total number of free records. The allocation of memory starts with “FFP”, which has to be an even row address and has to point to a chapter also reserved by the “IAT/XAT” register. It continues with increasing double rows (row-column addressing scheme). Row 24 and row 25 of a chapter are skipped. After reaching a chapter border (row 23) the next chapter will be used for single-packet allocation. The p80 chain chapters must be coherent.

The user is responsible for correct programming of the “FFP” registers, meaning that the start pointers and the length of the lists must be given in such a manner that memory segments do not overlap. The programming of “FFP” and the “IAT/XAT” registers must be done **before** using the “CREATE” commands.

While acquiring data it is possible, that memory space for pseudopacket acquisition may not be sufficient. By using the command “GET_FREE_CHAP” the free chap chain is shortened by one chapter which is given to the p40 chain. So the memory space for one page or subpage is exchanged for memory space for 25 pseudopackets. The other way round – giving memory space from the p40 chain to the chap chain – is also possible.

The “ADD” and “REMOVE” commands do not influence the size of the chains but only their contents.

Possible Memory Allocations for Internal RAM

Block/Byte Position	Purpose	Comment
0 /0	M3L registers temporary system memory DRCS memory	Compulsory
0 /1	M3L registers temporary system memory DRCS memory	Compulsory
0 /2	M3L registers temporary system memory DRCS memory	Compulsory
0 /3	M3L registers temporary system memory DRCS memory	Compulsory
0 /4	M3L registers temporary system memory DRCS memory	Compulsory
0 /5	M3L registers temporary system memory DRCS memory	Compulsory
1 /0	Graphic memory	Optional
1 /1	Graphic memory	Optional
1 /2	Graphic memory	Optional
1 /3	Graphic memory	Optional
1 /4	Graphic memory	Optional
1 /5	Graphic memory	Optional
2 /0	Display memory	Compulsory
2 /1	Display memory	Compulsory
2 /2	Display memory	Compulsory
2 /3	Display memory (with DRCS)	Optional
2 /4	Display memory (with DRCS)	Optional
2 /5	Packet buffer	PB_ADR (M3L)
3 /0	Graphic memory	Optional
3 /1	Graphic memory	Optional
3 /2	Graphic memory	Optional
3 /3	Graphic memory	Optional
3 /4	Graphic memory	Optional
3 /5	Graphic memory	Optional

Possible Memory Allocations for External RAM

Chapter	Purpose	Comment
0 (external)	Table Look Up (TLU)	Optional
1 (external)	Table Look Up (TLU)	Optional
2 (external)	Table Look Up (TLU)	Optional
3 (external)	Table Look Up (TLU)	Optional
4 (external)	Not defined	Optional
5 (external)	Not defined	Optional
6 (external)	Not defined	Optional
7 (external)	Not defined	Optional
8 (external)	Not defined	Optional
9 (external)	Page Trace Table (PTT) + Page Request Status Table (PRST)	PT_ADR (M3L)
10 (external)	Compressed Basic TOP Table	TOP_COM_BAS (M3L)
11 (external)	P80 chain	Optional
12 (external)	P80 chain	Optional
13 (external)	P80 chain	Optional
14 (external)	P80 chain	Optional
15 (external)	Not defined	Free
16 (external)	Not defined	Free
17 (external)	Not defined	Free
18 (external)	Not defined	Free
19 (external)	Not defined	Free
20 (external)	Not defined	Free
21 (external)	Not defined	Free
22 (external)	Not defined	Free
23 (external)	Not defined	Free

9 Restrictions of System

9.1 Flyback Mode

In flyback mode the software acquisition will be able to search and store every requested page within one cycle, if the user observes the following conditions when programming the acquisition:

N_P80	*	4	<	1000
-------	---	---	---	------

and

+	N_CHAP_TYP2	*	4	<	1000
	N_MAX_SUB	*	4		

with

- N_P80 total number of all P80 elements
- N_CHAP_TYP2 total number of chapters in all groups of type 2
- N_MAX_SUB maximum number of subpages in biggest subpage buffer

The worst case is given if a subpage number is neither requested in a group of type 0 nor in a group of type 2. Such subpage headers must be transmitted one after the other (without any normal row). In this case and if the conditions just explained are violated, a system crash will **not** occur, however transmitted data may get lost and the user will have to wait one more broadcast cycle to receive this page.

The formulas above show that use of a group of type 2 is only sensible for small memories (< 256 Kbyte). Also the user-defined group should be used only if it is absolutely necessary (i.e. Basic TOP Page).

So if your system has high memory capacity, save PU time, devote memory space to the table look up (4 Kbyte) and use group type 0 instead of 2. You will see that the performance of the acquisition only depends on the size of the biggest subpage buffer and the length of the user-defined groups.

10 How to Acquire Data

This chapter contains a practical example of how to program the acquisition of MEGATEXT. It is only meant to give you an overview and should not substitute detailed study of “MCI Description” and “M3L-Bus Registers Programmer’s Reference”. Depending on your system configuration (i.e. memory size, graphic capabilities, TOP processing) you can of course choose another configuration of the acquisition.

This example is programmed with “MTLI”, a program for easy control of MEGATEXT by a PC and a SIMON interface. External memory of 128 Kbytes is expected.

Syntax	WRITE MEGATEXT	register output byte
Syntax	WRRRC port_reg byte_mask chapter row column output byte	
	with	
port_reg	Megatext address pointer start register (R48, R56)	
byte_mask	Bit 7 = EXT_MEM	= Bit 7 in Address_Pointer_x0
	Bit 6 = 0	= Bit 6 in Address_Pointer_x0
	Bit [5:0]	= Megatext byte select if EXT_MEM = 0
chapter	Chapter Address, 3 bits if EXT_MEM = 0, other 11 bits	
row	5 bits	
column	6 bits	

```

; *****
print "Timing and initialization"
; *****
WRITE MEGATEXT    0d82  0x07           ;reg 82    output_pin_control
WRITE MEGATEXT    0d83  0x38           ;reg 83    rgb_control
WRITE MEGATEXT    0d85  0x04           ;reg 85    display_vco

WRITE MEGATEXT    0d108 0x00           ;reg 108   acquisition_timing_1
WRITE MEGATEXT    0d109 0x00           ;reg 109   acquisition_timing_0

WRITE MEGATEXT    0d112 0x02           ;reg 112   system_clock_control
WRITE MEGATEXT    0d113 0x00           ;reg 113   sync_source_selection
WRITE MEGATEXT    0d114 0x00           ;reg 114   display_pll_control
WRITE MEGATEXT    0d115 0xbf           ;reg 115   black_level_clamp
WRITE MEGATEXT    0d116 0x00           ;reg 116   display_timing
WRITE MEGATEXT    0d117 0x00           ;reg 117   v_delay_setting

; *****
print "Refresh of external RAM is switched on"
; *****
WRRC 48 01 0 6 31 64                  ;byte_0 block_0 row_6 column_31

; *****
print "set XRAM_SIZE to 512 kbyte"
; *****
WRRC 48 01 0 0 31 255                  ;byte_0 block_0 row_0 column_31
WRRC 48 02 0 0 31 1                    ;byte_1 block_0 row_0 column_31

; *****
print "Init all groups to NIL"
; *****
WRRC 48 01 0 7 0 1                     ;group_0
WRRC 48 01 0 7 2 1                     ;group_1
WRRC 48 01 0 7 4 1                     ;group_2
WRRC 48 01 0 7 6 1                     ;group_3
WRRC 48 01 0 7 8 1                     ;group_4
WRRC 48 01 0 7 10 1                    ;group_5
WRRC 48 01 0 7 12 1                    ;group_6
WRRC 48 01 0 7 14 1                    ;group_7

; *****
print "Init all groups to not used"
; *****
WRRC 48 32 0 7 0 0                     ;group_0
WRRC 48 32 0 7 2 0                     ;group_1
WRRC 48 32 0 7 4 0                     ;group_2
WRRC 48 32 0 7 6 0                     ;group_3
WRRC 48 32 0 7 8 0                     ;group_4
WRRC 48 32 0 7 10 0                    ;group_5

```

```

WRRc 48 32 0 7 12 0          ;group_6
WRRc 48 32 0 7 14 0          ;group_7

; *****
print "ACQ_INITIALIZING"
; *****
print "Packet_buffer_length = 17d"
; *****
WRITE MEGATEXT    0d1    0x00          ;reg 1    pb_length
WRITE MEGATEXT    0d2    0x17          ;reg 2    pb_length

; *****
print "Packet_buffer_start_address = Block 2 byte 5"
; *****
WRITE MEGATEXT    0d3    0x58          ;reg 3    pb_adr_2
WRITE MEGATEXT    0d4    0x10          ;reg 4    pb_adr_1
WRITE MEGATEXT    0d5    0x00          ;reg 5    pb_adr_0

; *****
print "Chap_chain_start_address = Block 1 byte 0"
; *****
WRITE MEGATEXT    0d40   0x00          ;reg 40   ffp_chap_2
WRITE MEGATEXT    0d41   0x48          ;reg 41   ffp_chap_1
WRITE MEGATEXT    0d42   0x00          ;reg 42   ffp_chap_0

; *****
print "Number_of_free_chapters 100d"
; *****
WRITE MEGATEXT    0d44   0x0           ;reg 44   nf_chap_1
WRITE MEGATEXT    0d45   0d100         ;reg 45   nf_chap_0

; *****
print "P80_chain_start_address = external ram Chapter 122 row_0"
; *****
WRITE MEGATEXT    0d64   0x83          ;reg 64   ffp_p80_2
WRITE MEGATEXT    0d65   0xd0          ;reg 65   ffp_p80_1
WRITE MEGATEXT    0d66   0x00          ;reg 66   ffp_p80_0

; *****
print "Number_of_free_p80_elements"
; *****
WRITE MEGATEXT    0d69   0d10          ;reg 69   nf_p80_0

; *****
print "P40_chain_start_address = external ram Chapter 93 row_0"
; *****
WRITE MEGATEXT    0d72   0x82          ;reg 72   ffp_p40_2
WRITE MEGATEXT    0d73   0xe8          ;reg 73   ffp_p40_1
WRITE MEGATEXT    0d74   0x00          ;reg 74   ffp_p40_0

```



```

; *****
print "Number_of_free_p40_elements = 700"
; *****
WRITE MEGATEXT 0d76 0x02 ;reg 76 nf_p40_1
WRITE MEGATEXT 0d77 0xbc ;reg 77 nf_p40_0
; *****
print "Set Slicer control = 26h"
; *****
WRITE MEGATEXT 0d81 0x26 ;reg 81 slicer_control
; *****
print "Init internal memory allocation register"
print ""
print "IAT_2 :B3/By5 B3/By4 B3/By3 B3/By2 B3/By1 B3/By0 B2/By5 B2/By4"
print "      0      0      0      0      0      0      1      1  "
print "IAT_1 :B2/By3 B2/By2 B2/By1 B2/By0 B1/By5 B1/By4 B1/By3 B1/By2"
print "      1      1      1      1      1      1      1      1  "
print "IAT_0 :B1/By1 B1/By0 B0/By5 B0/By4 B0/By3 B0/By2 B0/By1 B0/By0"
print "      1      0      1      1      1      1      1      1  "
; *****
WRITE MEGATEXT 0d88 0x03 ;reg 88 iat_2
WRITE MEGATEXT 0d89 0xff ;reg 89 iat_1
WRITE MEGATEXT 0d90 0xbf ;reg 90 iat_0
; *****
print "Init external memory allocation register"
print ""
print "XAT_2:CHAP23 CHAP22 CHAP21 CHAP20 CHAP19 CHAP18 CHAP17 CHAP16"
print "      0      0      0      0      0      0      0      0  "
print "XAT_1:CHAP15 CHAP14 CHAP13 CHAP12 CHAP11 CHAP10 CHAP09 CHAP08"
print "      0      0      0      0      0      0      0      0  "
print "XAT_0:CHAP07 CHAP06 CHAP05 CHAP04 CHAP03 CHAP02 CHAP01 CHAP00"
print "      0      0      0      0      0      0      0      0  "
; *****
WRITE MEGATEXT 0d91 0x0 ;reg 91 xat_2
WRITE MEGATEXT 0d92 0x0 ;reg 92 xat_1
WRITE MEGATEXT 0d93 0x0 ;reg 93 xat_0
; *****
print "Data entry window start line = 6"
; *****
WRITE MEGATEXT 0d96 0d6 ;reg 96 dew_start_line
; *****
print "Data entry window end line = 23"
; *****
WRITE MEGATEXT 0d97 0d23 ;reg 97 dew_end_line

```

```

; *****
print "VPS dataline = 16"
; *****
WRITE MEGATEXT    0d98    0d16                ;reg 98    single_data_line
; *****
print "TTX_framing_code_window = 62h"
; *****
WRITE MEGATEXT    0d99    0x62                ;reg 99    ttx_framing_window
; *****
print "VPS_framing_window = e0h"
; *****
WRITE MEGATEXT    0d100  0xe0                ;reg 100   extra_framing_window
; *****
print "Reception threshold = cfh"
; *****
WRITE MEGATEXT    0d105  0xcf                ;reg 105   reception_threshold
; *****
print "All pseudopackets are enabled"
; *****
WRITE MEGATEXT    0d106  0x0f                ;reg 106   pseudo_packet_enable
; *****
print "Compressed Basic_TOP_Table = block 1 byte 2"
; *****
WRITE MEGATEXT    0d120  0x01                ;reg 120   top_com_bas_2
WRITE MEGATEXT    0d121  0x08                ;reg 121   top_com_bas_1
WRITE MEGATEXT    0d122  0x00                ;reg 122   top_com_bas_0
; *****
print "Page trace = block 1 byte 1"
; *****
WRITE MEGATEXT    0d123  0x00                ;reg 123   pt_adr_2
WRITE MEGATEXT    0d124  0x88                ;reg 124   pt_adr_1
WRITE MEGATEXT    0d125  0x00                ;reg 125   pt_adr_0
; *****
print "Start of commands"
; *****
print "Create free chap chain"
; *****
WRITE MEGATEXT    0d37    0d7                ;mci command execute
; *****
print "Create free p40 chain"
; *****
WRITE MEGATEXT    0d37    0d8                ;mci command execute

```

```

; *****
print "Create free p80 chain"
; *****
WRITE MEGATEXT    0d37    0d9                ;mci command execute

; *****
print "Init group_0 with type_2 and subpage don't care"
; *****
WRITE MEGATEXT    0d11    0x2                ;type 2                mci0_2
WRITE MEGATEXT    0d12    0x0                ;group 0            mci0_1
WRITE MEGATEXT    0d16    0x07              ;do care mask        mcil_5
WRITE MEGATEXT    0d17    0xff              ;do care mask        mcil_4
WRITE MEGATEXT    0d18    0x0              ;do care mask        mcil_3
WRITE MEGATEXT    0d19    0x0              ;do care mask        mcil_2
WRITE MEGATEXT    0d20    0x0              ;do care mask        mcil_1
WRITE MEGATEXT    0d37    0d50              ;command WRITE_GROUP

; *****
print "Switch acq on with VPS, PAL, new ttx_fc and page trace = on"
; *****
WRITE MEGATEXT    0d13    0x2d    ;vps,pal,new ttx_fc,page trace    mci0_0
WRITE MEGATEXT    0d12    0xe4    ;framing code                    mci0_1
WRITE MEGATEXT    0d37    0d0      ;command ACQ_CONTROL

; *****
print "ADD all pages from 100 to 106 with command ADD_USER_PAGE"
; *****
WRITE MEGATEXT    0d8      0x04                ;pblf = 1                mci0_5
WRITE MEGATEXT    0d9      0x00                ;                        mci0_4
WRITE MEGATEXT    0d12    0x00                ;group = 0              mci0_1
; *****
WRITE MEGATEXT    0d16    0x01                ;magazine 1            mcil_5
WRITE MEGATEXT    0d17    0x00                ;page_100              mcil_4
WRITE MEGATEXT    0d34    0d04                ;mci command execute

; *****
print "ADD_PAGE 100"
; *****
WRITE MEGATEXT    0d17    0x01                ;page_101              mcil_4
WRITE MEGATEXT    0d37    0d04                ;mci command execute

; *****
print "ADD_PAGE 101"
; *****
WRITE MEGATEXT    0d17    0x02                ;page_102              mcil_4
WRITE MEGATEXT    0d37    0d04                ;mci command execute

; *****
print "ADD_PAGE 103"
; *****

```

```

WRITE MEGATEXT    0d17    0x03                ;page_103          mcil_4
WRITE MEGATEXT    0d37    0d04                ;mci command execute
; *****
print "ADD_PAGE 103"
; *****
WRITE MEGATEXT    0d17    0x04                ;page_104          mcil_4
WRITE MEGATEXT    0d37    0d04                ;mci command execute
; *****
print "ADD_PAGE 104"
; *****
WRITE MEGATEXT    0d17    0x05                ;page_105          mcil_4
WRITE MEGATEXT    0d37    0d04                ;mci command execute
; *****
print "ADD_PAGE 105"
; *****
WRITE MEGATEXT    0d17    0x06                ;page_106          mcil_4
WRITE MEGATEXT    0d37    0d04                ;mci command execute
; *****
print "ADD_PAGE 106"
; *****
End

```

Problem	Solution
You want to get the physical memory address of a page or a packet to display the received page.	Use the "SEARCH" command and the address pointer will be returned.
You want to erase any data to obtain memory space for new pages.	Use the "REMOVE" command and the address pointer will be returned. Acquisition need not be switched off.
You want to request new pages.	Use the appropriate ADD command. Acquisition need not be switched off.
You want to change the memory segment limits.	If you only want to devote more memory space on pseudopacket acquisition, you will have to use the command "GET_FREE_CHAP". In this case the acquisition need not be switched off. Be sure that the chapter chain has free elements before using this command.

11 Feature Table

Feature	Flyback Mode
Request of pages with table look up (fixed do care mask)	Yes (2048 pages)
Request of pages with user do care mask (5 bytes)	Yes
Request of non-page-related data	Yes
Multimode acquisition	Yes (up to eight different groups)
Page collect mode	Yes (ref. to REQUEST_ALL_PAGES)
Automatic request of FLOF pages	Yes
Request of top tables	Yes
Compression of basic TOP tables in parallel mode	Yes
Subpage collect mode	Yes
Subpage collect mode with overwrite	Yes
Subpage select mode	Yes
Subpage don't care mode	Yes
Request of page-related pseudopackets X/25, X/26, X/27 and X/28	Yes Dynamic allocation of memory (40 bytes for each received packet)
Request of non-page-related pseudopackets X/29, 8/30	Yes Allocation of memory (80 bytes for each requested packet)
Page-selective pseudopacket enable	Yes
Packet-selective pseudopacket enable	Yes
Page trace	Yes
Subpage trace	Yes
Pseudopacket trace	Yes
NTSC/PAL switch	Yes
External memory upto 2 Mbyte	Yes
Integrated memory manager	Yes
Programmable reception threshold	Yes (1-bit framing code error, magazine number errors, row number errors)

Feature Table (cont'd)

Feature	Flyback Mode
Online checks (page-selective programmable)	Yes parity 1-byte hamming, 3-byte hamming, mixed check (TOP)
Programmable teletext framing code	Yes
Clear before first reception	Yes
Hold enable	Yes
Remove protection	Yes
End-of-page interrupt	Yes
Signal quality support	Yes
Rolling header support	Yes
FLOF support (MCI commands)	Yes
VPS support	Yes (biphase decoding included)

TOP Support (MCI Commands)

ADD_ALL_TOP_PAGES	Yes
COUNT_TOP_PAGENUMBER	Yes
WRITE_TOP_TITLE	Yes
SEARCH_TOP_PAGE	Yes

FLOF Support (MCI Commands)

EXECUTE_FLOF	Yes
--------------	-----