VME - AIO16

16 Analog Inputs and
4 Analog Outputs
with 16 (12) Bit Resolution

Software Manual

Document file:	I:\texte\Doku\MANUALS\VME\AIO16\Englisch\AIO1637S.en9
Date of print:	04.09.2002

Firmware version described:	aiosrv 07

Changes in the chapters

The changes in the manual listed below affect changes in the **hardware** as well as changes in the **description** of facts only.

Chapter	Changes compared to previous manual
-	First English edition.
-	-

Technical details are subject to change without further notice.

<u>N O T E</u>

The information in this document has been carefully checked and is believed to be entirely reliable. **esd** makes no warranty of any kind with regard to the material in this document, and assumes no responsibility for any errors that may appear in this document. **esd** reserves the right to make changes without notice to this, or any of its products, to improve reliability, performance or design.

esd assumes no responsibility for the use of any circuitry other than circuitry which is part of a product of **esd** gmbh.

esd does not convey to the purchaser of the product described herein any license under the patent rights of **esd** gmbh nor the rights of others.

esd electronic system design gmbh
Vahrenwalder Str. 207
30165 Hannover
Germany

Phone: +49-511-372 98-0
Fax: +49-511-372 98-68
E-mail: info@esd-electronics.com
Internet: www.esd-electronics.com

USA / Canada:
esd electronics Inc.
12 Elm Street
Hatfield, MA 01038-0048
USA

Phone: +1-800-732-8006
Fax: +1-800-732-8093
E-mail: us-sales@esd-electronics.com
Internet: www.esd-electronics.us

Contents

Page

Contents

Page

1. Overview

The firmware of the VME-AIO16 supports setting the D/A-converters and reading-back the A/D-converter values. Following each system boot-up the firmware carries out a self test which includes the calculation of the gain and offset errors of the input circuit. Therefore it is possible to offer the user values which have already been corrected.

In addition further parameters such as trigger conditions of the A/D-conversion, the CPU cycle frequency, the timer period or the output values of the D/A-converters can be transferred after the system was booted.

The programming occurs exclusively via the VMEbus.

Commands and parameters are transferred by setting memory cells in the shared-RAM range (parameter buffer) and triggering a commander interrupt. The transfer structure with semaphore guarantees a trouble-free access possibility via local CPU and VMEbus master.

This page has intentionally been left blank.

2. Default Parameters

The VME-AIO16 has got an I²C-EEPROM in which parameters can be stored.

The VME-AIO16 leaves the manufacturer with the default parameters for the I²C-EEPROM as will be explained the following chapters. The user can change these parameters and then store them in the I²C-EEPROM again. This guarantees the preservation of parameters after a RESET.

During each boot-up the firmware checks the I²C-EEPROM. If errors were detected, the EEPROM would be assigned with the original default parameters set by the manufacturer.

The user can also delete the I²C-EEPROM by the command 'system re-boot with default parameters' and thus restore the original parameters. Doing this all parameter changes will be lost.

The command 'system re-boot with default parameters' will be explained in more detail together with all other commands in the following chapters.

This page has intentionally been left blank.

3. Shared-Address Range

The shared-address range is the interface between the local CPU and the VMEbus. The VMEbus master and the local CPU can access these cells.

3.1 Notes on VMEbus Access

When the board leaves the manufacturer the VMEbus basis address is set to:

$xx680000

Via the **VMEbus** the shared-address range can only be accessed on every second word address. Therefore the relative VMEbus address in the following table have been listed in a separate column. The absolute address via the VMEbus is calculated as follows:

absolute_VME_Address = VME_Base_Address + VMEbus_Relative_Address

3.2 Notes on Accessing the Local CPU

The basis address for the shared-address range can be programmed in the registers of the CPU 68340. At the moment it is set to:

$0080.0000

The absolute local address is calculated as follows:

Absolute_Local_Address = Local_Base_Address + Local_Relative_Address

3.3 VMEbus-Addresses and Addresses of Local CPU

In the following table the relative addresses of the local CPU are shown in the first column and the addresses relative to the VMEbus-basis address of the VME-AIO16 are shown in the second column.

<table>
<tr><th rowspan="2">Access via VMEbus:
VMEbus base + address offset
[HEX]</th><th rowspan="2">Access via local CPU
$0080.0000 +
address offset
[HEX]</th><th colspan="2">Component/signal</th></tr>
<tr><th>Write access</th><th>Read access</th></tr>
<tr><td>7FF7C
:
00000</td><td>3FFBE
:
00000</td><td colspan="2">256 kbyte shared SRAM minus AD/DA-converter range,
16 bit data width,
accessible on VMEbus via every second word address</td></tr>
<tr><td>7FF80</td><td>3FFC0</td><td>set D/A-converter 1 without SWLDAC</td><td>D/A-converter 1: read back value from RAM-cell 1.1</td></tr>
<tr><td>7FF84</td><td>3FFC2</td><td>set D/A-converter 2 without SWLDAC</td><td>D/A-converter 2: read back value from RAM-cell 2.1</td></tr>
<tr><td>7FF88</td><td>3FFC4</td><td>set D/A-converter 3 without SWLDAC</td><td>D/A-converter 3: read back value from RAM-cell 3.1</td></tr>
<tr><td>7FF8C</td><td>3FFC6</td><td>set D/A-converter 4 without SWLDAC</td><td>D/A-converter 4: read back value from RAM-cell 4.1</td></tr>
<tr><td>7FFA0</td><td>3FFD0</td><td colspan="2">Aux_In_8</td></tr>
<tr><td>7FFA4</td><td>3FFD2</td><td colspan="2">Aux_In_7</td></tr>
<tr><td>7FFA8</td><td>3FFD4</td><td colspan="2">Aux_In_6</td></tr>
<tr><td>7FFAC</td><td>3FFD6</td><td colspan="2">Aux_In_5</td></tr>
<tr><td>7FFB0</td><td>3FFD8</td><td colspan="2">Aux_In_4</td></tr>
<tr><td>7FFB4</td><td>3FFDA</td><td colspan="2">Aux_In_3</td></tr>
<tr><td>7FFB8</td><td>3FFDC</td><td colspan="2">Aux_In_2</td></tr>
<tr><td>7FFBC</td><td>3FFDE</td><td colspan="2">Aux_In_1</td></tr>
</table>

Access via VMEbus: VMEbus base + address offset [HEX]	Access via local CPU `$0080.0000+` address offset [HEX]	Component/signal	
		Write access	Read access
7FFC0	3FFE0	activate SWLDAC	reserved
7FFC4	3FFE2	activate SWLDAC	reserved
7FFC8	3FFE4	activate SWLDAC	reserved
7FFCC	3FFE6	activate SWLDAC	reserved
7FFD0	3FFE8	activate SWLDAC	reserved
7FFD4	3FFEA	activate SWLDAC	reserved
7FFD8	3FFEC	activate SWLDAC	reserved
7FFDC	3FFEE	activate SWLDAC	reserved
7FFE0	3FFF0	SWCONV (start A/D-conversion)	reserved
7FFE4	3FFF2	activate DACDMAREQ (not yet implemented)	reserved
7FFE8	3FFF4	SWCOM (local IRQ triggered by VMEbus)	reserved
7FFEC	3FFF6	WRIVEC (write VMEbus-IRQ-vector)	reserved
7FFF0	3FFF8	not assigned	reserved
7FFF4	3FFFA	not assigned	reserved
7FFF8	3FFFC	not assigned	reserved
7FFFC	3FFFE	not assigned	reserved

Table 3.2.1: Assignment of shared areas with address of the VMEbus and local CPU

This page has intentionally been left blank.

4. Commands and Parameter Transfer

4.1 Assignment of Parameter Buffer

The parameter buffer in the shared-RAM can be divided into various functional modules. The following table gives an overview about these modules with notes on the individual chapters in which the modules will be described in detail.

Relative address range VMEbus [HEX]	Relative address range local [HEX]	Functional module	Description from page:
0000 ... 001E	0000 ... 000F	board identification	13
0020 ... 003E	0010 ... 001F	board status	14
0040 ... 004E	0020 ... 0027	command section for parameter transfer	17
00C0 ... 00DE	0060 ... 006F	preloads	27
0100 ... 011E	0080 ... 008F	A/D-buffer status structure	69
0120 ... 013E	0090 ... 009F	D/A-buffer status structure	69
0140 ... 015E	00A0 ... 00AF	control parameter	31
0160 ... 017E	00B0 ... 00BF	selective-D/A-buffer status	32
01F0 ... 01FE	00F8 ... 00FF	A/D-converter status	47
0200 ... 023C	0100 ... 011F	A/D-converter: crude data	49
0240 ... 027C	0120 ... 013F	A/D-converter: corrected data	51
0280 ... 02F8	0140 ... 017F	A/D-converter: summed-up 32-bit values	53
0300 ... 031C	0180 ... 018F	Aux_In_1-8: crude data	62
0320 ... 033C	0190 ... 019F	Aux_In_1-8: corrected data	63
0400 ... 043C	0200 ... 021F	A/D-converter: offset values	54
0440 ... 047C	0220 ... 023F	A/D-converter: .+5V-reference data	55
0480 ... 04BC	0240 ... 025F	D/A-converter: offset values (info only)	57
04C0 ... 04FC	0260 ... 027F	D/A-converter: gain values (info only)	58
0500 ... 053C	0280 ... 029F	A/D-converter: gain correction factors	60
0540 ... 05B8	02A0 ... 02DF	A/D-converter: 32-bit offset values of A/D-conversion adding	61
05C0 ... 05DC	02E0 ... 02EF	Aux_In_1-8: offset values	64
05E0 ... 05FC	02F0 ... 02FF	Aux_In_1-8: gain values	65

Relative address range VMEbus [HEX]	Relative address range local [HEX]	Functional module	Description from page:
0800 ... xxxx	0400 ... yyyy	A/D-buffer (capacity definable with restriction)	67
uuuu ... 7FDFE	vvvv ... 3FEFF	D/A-buffer (capacity definable with restriction)	67

The tables in the following chapters show the cells of the parameter buffer with their relative address in the shared RAM relevant to the user. All cells, apart from `cmmd`, `adstat`, `para` and `sema`, must only be read. They can be set indirectly via the `cmmd`-commands and its parameters.

4.2 Board Identification

In these memory cells the designation of the VME-AIO16 is stored in ASCII-code.

Relative address VMEbus [HEX]	Relative address local [HEX]	Length	Access	Cell contents	Short description
0...1E	0...F	16 bytes	read only	`ASCII-Id`	identification of VME-AIO16 via ASCII-text

As identification text

`‘esd_AIO16_LevX.Y’`

is returned as ASCII-code at the moment. ‘`X`’ and ‘`Y`’ represent the revision number of the firmware.

4.3 Board Status

This cell represents the current status of the commander. It should be read after a boot-up before the first command call in order to check, whether the self test has been successfully completed.

Relative address VMEbus [HEX]	Relative address local [HEX]	Length	Access	Parameter name	Parameter values [HEX]
20	10	WORD	read only	`card_stat`	value range: $0000...$8001 $0001 ... $000n: an error occurred during self test (esd-internal error status) $7FFF: VME-AIO16 is in self test $8001: self test has been completed successfully

4.4 Hardware Revision

This cell contains details about the hardware revision of the VME-AIO16 board.

Relative address VMEbus [HEX]	Relative address local [HEX]	Length	Access	Parameter name	Parameter value [HEX]
24	12	WORD	read only	`HWrev`	value range: $0000, $0001 $0000 old hardware $0001 new hardware (with fast SRAM)

4.5 Procedure at Command Call

The memory cells (such as A/D-values) can always be read back transparently. When setting parameters and commands, however, the following procedure has to be observed:

(1.) Only required for multi master or multi tasking applications:
Evaluation of cell `sema`
Via cell `sema` you can enquire whether the parameter buffer is free. Should this be the case, `sema` has to be set ('TAS'-command).

2. Reading cell `cmmd` (command)
If the commander is free, i.e. the last command has been executed, the value '0000' is returned at a read access (word) to the relative address $22. Only if this value has been read, the desired command can be written with its parameters into the cell of the shared RAM. Otherwise you have to wait until the commander is free.

3. Setting the command and the parameters
If the conditions under point 2 have been met, a command with the according parameters can be set in the shared RAM.

4. Triggering the commander interrupt
By write-accessing the relative word address $7FFEE8 via the VMEbus with any data an interrupt of the local CPU is triggered. The CPU then takes on the command and executes it.

5. Waiting for command processing to be completed
If the local CPU has set cell `cmmd` = 0000, the command has been completed.

6. Evaluating cell `cstat`
Here you can check, whether the command has been executed successfully (following `cmmd` = 0).
(`cstat` = 0 -> OK, `cstat` ≠ 0 -> error)

(7.) Only required for multi master or multi tasking applications:
enabling cell `sema`
If no more commands are to be executed, cell `sema` has to be enabled again.

Note: The processing period of the commands called depends mostly on the kind of command:

- for commands in the range of $0xxx : ≈100 µs
- for commands in the range of $8xxx : several milliseconds, (depending on the command)

4.6 Command Section

These cells are used for semaphore control and command transfer. Cells in the parameter buffer must only be set via the command call by `cmmd` and `para`! The only exception is cell `sema` (see below).

<table>
<tr><th>Relative address VMEbus [HEX]</th><th>Relative address local [HEX]</th><th colspan="2">Length</th><th>Access</th><th>Cell contents</th><th>Short description</th></tr>
<tr><td>40</td><td>20</td><td colspan="2">BYTE</td><td>read only</td><td><code>cstat</code></td><td>commander status</td></tr>
<tr><td>41</td><td>21</td><td colspan="2">BYTE</td><td>read/write</td><td><code>sema</code></td><td>semaphore</td></tr>
<tr><td>44</td><td>22</td><td colspan="2">WORD</td><td>read/write</td><td><code>cmmd</code></td><td>command</td></tr>
<tr><td>48</td><td>24</td><td>WORD (para1.w)</td><td rowspan="2">LWORD</td><td rowspan="3">read/write</td><td rowspan="3"><code>para</code></td><td rowspan="3">parameter of command</td></tr>
<tr><td>4C</td><td>26</td><td>WORD (para2.w)</td></tr>
<tr><td>50</td><td>28</td><td>WORD (para3.w)</td><td></td></tr>
</table>

The parameters are transferred depending on the length as shown below:

<table>
<tr><td rowspan="2">Relative addresses -> [HEX]</td><td>VMEbus</td><td>48</td><td>4C</td><td>50</td></tr>
<tr><td>local</td><td>24</td><td>26</td><td>28</td></tr>
<tr><td rowspan="2"><code>para</code> -></td><td>WORD parameter</td><td>WORD parameter 1</td><td>WORD parameter 2</td><td>WORD parameter 3</td></tr>
<tr><td>LWORD-parameter</td><td>upper WORD of LONG parameter</td><td>lower WORD of LONG parameter</td><td>-</td></tr>
</table>

4.6.1 Cells `sema` and `cstat`

Cell `sema` is not set by the local software, because it is reserved for semaphore control via the VMEbus. Only following a self test the cell is pre-set with $00.

`sema` has to be read and set by the VMEbus master before a command call is made. By this read/set sequence the user can make sure that only one master or one task accesses the commander of the VME-AIO16 simultaneously.

Following the call cell `cstat` can be evaluated in order to check whether the access has been successful.

Address [HEX]	Length	Parameter name	Parameter values [HEX]
41	Byte	`sema`	value range: $0x, $8x
			only set and evaluate bit 7: 0xxx xxxx commander is free 1xxx xxxx commander is occupied

Only bit 7 of `sema` is to be set and evaluated. Bits 6 to 0 are reserved for future applications.

The local server writes into `cstat`, whether the command transferred has been executed successfully.

Address [HEX]	Length	Parameter name	Parameter values [HEX]
40	Byte	`cstat`	value range: $00, $FF
			= 0 no error ≠ 0 error

4.6.2 General Information about Command Cell `cmmd`

In cell `cmmd` the various commands are set. If a command has been processed, the firmware sets the cell to ‘0000’. This marks the completion of the command handling.

4.6.3 Overview of Command and Configurable Parameters

Parameters are set by writing a command into cell `cmmd` and transferring the parameter in cell `para` in the parameter buffer. The write sequence has been described in the previous chapter ‘Procedure at Command Call’.

This chapter explains the individual commands with their parameters and their value ranges.

The following tables represent an overview about all previously implemented commands and their parameters.

`cmmd` [HEX]	Command designation	`para`	Status readable at address [HEX]	Short description
8000	store current parameter in I²C-EEPROM	no parameters are transferred with these ‘system commands’	-	parameters are also retained after a RESET
8001	system re-boot incl. self test (= RESET)			parameters are removed from I²C-EEPROM after a re-boot
8002	system re-boot with default parameters			standard initialisation: I²C-EEPROM is configured with default parameters again: -> all changed parameters are lost!
8003	statistic of current buffer	2x WORD	`cstat`	creating the statistic; parameter = mask for the selection of A/D-converter channels and auxiliary inputs
8004	buffer correction	3x WORD	`cstat`	executing the buffer correction; parameter = mask for selecting the offset and/or gain correction of the A/D-converter channels and the auxiliary inputs
8005	averaging process	3x WORD	`cstat`	averaging process; parameter = mask for selecting the A/D-converter channels and auxiliary inputs for averaging

cmmd [HEX]	para	Command designation	Status readable on VME or (local) address [HEX]	Name of status cell	Short description
1	WORD	set VMEbus-IRQ-level	140 (A0)	`vmelev`	interrupt of VME-AIO16 on the VMEbus
2	WORD	set VMEbus-IRQ-vector	141 (A1)	`vmevec`	vector for VMEbus-IACK cycle
3	WORD	multiplexers of A/D-inputs	144 (A2)	`muxmode`	is automatically set by firmware during adjustment
4	WORD	multiplexers of D/A-outputs	145 (A3)	`dacmode`	is automatically set by firmware during adjustment
5	WORD	select source for trigger of A/D-conversion	148 (A4)	`trigmod`	conversion can be initiated by software, timer or external trigger
6	WORD	select source for LDAC of D/A-converters	149 (A5)	`ldcmod`	LDAC is used to take on the updated D/A-values
7	WORD	selecting the A/D-data handling	14C (A6)	`vadsrv`	DMA-mode, return of corrected A/D-values, adding of A/D-values
8	WORD	select first A/D-channel for reading-back	14D (A7)	`vstart`	selection of A/D-channels whose data is to be updated by local CPU in shared RAM
9	WORD	select last A/D-channel for reading back	150 (A8)	`vend`	
A	WORD	source for VMEbus interrupt	151 (A9)	`vvtrg`	selecting interrupt source for IRQ on VMEbus
B	WORD	A/D-converter resolution	154 (AA)	`vadres`	not for programming by the user
C	WORD	selecting the D/A-data handling	155 (AB)	`vdasrv`	buffer modes of D/A-converters
D	WORD	number of A/D-values to be added	156 (AC)	`vsmcnt`	adding of A/D-values
E	2x WORD	set up A/D-buffer	100 (80)	structure `adbuf`	sets the number of frames/buffers and the number of buffers for A/D-channels
F	2x WORD	set up D/A-buffer	120 (90)	structure `dabuf`	sets the number of frames/buffers and the number of buffer for D/A-channels
10	WORD	select first D/A-channel for buffer mode	15C (AE)	`dastart`	selecting the D/A-channels to be set in buffer mode (insignificant for other D/A-operating modes)
11	WORD	select last D/A-channel for buffer mode	15E (AF)	`daend`	
12	2x WORD	selective D/A-buffer	160, 168 (B0, B4)	`vdasb` `vdamsk`	selecting the D/A-buffer and D/A-channels
13...1F	-	reserved	-	-	-

cmmd [HEX]	para	Command designation	Status readable on VME or (local) address [HEX]	Name of status cell	Short description
20	WORD	D/A-value after RESET for channel 1	D0 (68)	dac1p	output voltage which is to be put out after a RESET
21	WORD	D/A-value after RESET for channel 2	D4 (6A)	dac2p	
22	WORD	D/A-value after RESET for channel 3	D8 (6C)	dac3p	
23	WORD	D/A-value after RESET for channel 4	DC (6E)	dac4p	
24...2F	-	reserved	-	-	-
30	LONG	time of period for trigger of conversion preset by timer	C8 (64)	cnvtime	when read the actually attainable value is specified in cnvtime
31	LONG	CPU-frequency preset	C0 (60)	tifreq	in tifreq the current CPU-frequency/2 can be read

4.7 Description of System Commands

First we will explain the so-called ‘system commands’. These commands are transferred with negative values ($8xxx). They carry out superior functions and therefore require more time for their execution than the other commands. Normally they are required only once during the basic initialisation.

The above mentioned explanation will be followed by a description of commands during which parameters are transferred. The default values and the new parameters can be read in the parameter buffer.

4.7.1 Store Current Parameter in the I²C-EEPROM

`cmmd` [HEX]	`para`	Status readable at address [HEX]	Name of status cell	Value range of the parameter
8000	-	-	-	no parameter transfer

If the changed parameters are to be retained following a RESET or a software re-boot, they have to be stored into the I²C-EEPROM. If the I²C-EEPROM is to be deleted again in order to reactivate the default parameters which had been standardly set, the command ‘system re-boot with default parameters’ has to be called (see following chapters).

Attention: Not all parameters can be stored in the I²C-EEPROM. The parameters of commands ‘7’, ‘C’, ‘E’ and ‘F’ for instance are not stored. Parameters which cannot be stored will be especially indicated in the command description.

4.7.2 System Re-Boot Incl. Self Test

`cmmd` [HEX]	`para`	Status readable at address [HEX]	Name of status cell	Value range of parameter
8001	-	-	-	no parameter transfer

By means of this command a ‘RESET’ can be triggered via VMEbus with software on the VME-AIO16. All parameters which have not been previously stored in the I²C-EEPROM will be lost. The current AD/DA-data in the shared RAM will also be lost. The firmware re-boots and a self test is carried out on the board. The self test includes the automatic determination of the offset and the gain of the input channels.

4.7.3 System Re-Boot with Default Parameters

cmmd [HEX]	para	Status readable at address [HEX]	Name of status cell	Value range of parameter
8002	-	-	-	no parameter transfer

The system re-boot with default parameters is similar to the re-boot described in the section above. Additionally the default parameters configured standardly are stored in the I²C-EEPROM again. All previous changes of parameters will be lost!

4.7.4 Statistic about Current Buffer

cmmd [HEX]	para	Status readable at VMEbus address [HEX]	Status readable at local address [HEX]	Name of status cell	Name of parameter	Parameter value [HEX]
8003	WORD (para1.w)	40	20	`cstat`	`auxen_stat`	value range: 0000...00FF
	WORD (para2.w)	40	20	`cstat`	`adcen_stat`	value range: 0000...FFFF

By means of this command the input channels for the statistical evaluation are selected (see 'Statistic Structure at page 75).

The success of the command transfer can be checked by evaluating `cstat`.

For this command the parameters `auxen_stat` and `adcen_stat` are set.

The parameters are assigned as follows:

Bit of `auxen_stat`	Auxiliary input
0	Enable_Aux_In_1
1	Enable_Aux_In_2
2	Enable_Aux_In_3
:	:
7	Enable_Aux_In_8
8	-
:	:
15	-

Bit of `adcen_stat`	A/D-converter channel
0	Enable_A/D_1
1	Enable_A/D_2
2	Enable_A/D_3
:	:
:	:
:	:
:	:
15	Enable_A/D_16

If a bit is set to '1', the according input channel is enabled.

4.7.5 Buffer Correction

cmmd [HEX]	para	Status readable at VMEbus address [HEX]	Status readable at local address [HEX]	Name of status cell	Name of parameter	Parameter values [HEX]
8004	WORD (para1.w)	40	20	cstat	auxen_kor	value range: 0000...00FF
	WORD (para2.w)	40	20	cstat	adcen_kor	value range: 0000...FFFF
	WORD (para3.w)	40	20	cstat	off_gain	value range: 0000...FFFF

Via this command the input channels are selected for whose values an offset or/and a gain correction is to be made.

By evaluating `cstat` you can check whether the command has been completed successfully.

For this command the parameters `auxen_kor, adcen_kor` and `off_gain` are set.
The parameters `auxen_kor` and `adcen_kor` are assigned as follows:

Bit of auxen_kor	Auxiliary input
0	Enable_Kor_Aux_In_1
1	Enable_Kor_Aux_In_2
2	Enable_Kor_Aux_In_3
:	:
7	Enable_Kor_Aux_In_8
8	-
:	:
15	-

Bit of adcen_kor	A/D-converter channel
0	Enable_Kor_A/D_1
1	Enable_Kor_A/D_2
2	Enable_Kor_A/D_3
:	:
:	:
:	:
:	:
15	Enable_Kor_A/D_16

If a bit is set to ‘1’, the according input channel is enabled.

Parameter `off_gain`:

By means of this parameter you can select, whether only an offset correction or an offset and a gain correction is to be executed for the channels executed via `auxen_kor` and `adcen_kor` :

`off_gain` = 0 : execute only offset correction
`off_gain` ≠ 0 : execute offset and gain correction

4.7.6 Averaging Process

cmmd [HEX]	para	Status readable at VMEbus address [HEX]	Status readable at local address [HEX]	Name of status cell	Name of parameter	Parameter value [HEX]
8005	WORD (para1.w)	40	20	cstat	auxen_avr	value range: 0000...00FF
	WORD (para2.w)	40	20	cstat	adcen_avr	value range: 0000...FFFF
	WORD (para3.w)	40	20	cstat	avr_cnt	value range: 0000...FFFF

Via this command the input channels are selected for which an average is to be made.

By evaluating `cstat` it can be checked, whether the command has been executed successfully.

For this command the parameters `auxen_avr, adcen_avr` and `avr_cnt` are set.

Parameters `auxen_avr` and `adcen_avr` are assigned as follows:

Bit of auxen_avr	Auxiliary input
0	Enable_Average_Aux_In_1
1	Enable_Average_Aux_In_2
2	Enable_Average_Aux_In_3
:	:
7	Enable_Average_Aux_In_8
8	-
:	:
15	-

Bit of adcen_avr	A/D-converter channel
0	Enable_Average_A/D_1
1	Enable_Average_A/D_2
2	Enable_Average_A/D_3
:	:
:	:
:	:
:	:
15	Enable_Average_A/D_16

If a bit is set to ‘1’, the according input channel is enabled.

Parameter `avr_cnt`:

By means of this parameter you can select how many values are to be used for averaging:

`avr_cnt` = 1, 3, 5, ...

4.8 Preload

In these cells the period time for the trigger of conversion via timer, the CPU-frequency and the D/A-converter values which are to apply at the outputs after a system re-boot can be specified.

4.8.1 Overview of Readable Cells of Parameter Buffer

The following table shows the memory cells of the parameter buffer in which the current preload values can be *read*.

The values must be set by command call via the cells `cmmd` and `para`!

Relative address VMEbus [HEX]	Relative address local [HEX]	Length of status cell	Access	Name of status cell	Short description
C0	60	Upper WORD	read only	`tifreq`	timer input frequency [Hz] (= CPU-frequency: 2)
C4	62	Lower WORD			
C8	64	Upper WORD	read only	`cnvtime`	current conversion time in [ns]
CC	66	Lower WORD			
D0	68	WORD	read only	`dac1p`	D/A-converter 1: value after RESET
D4	6A	WORD	read only	`dac2p`	D/A-converter 2: value after RESET
D8	6C	WORD	read only	`dac3p`	D/A-converter 3: value after RESET
DC	6E	WORD	read only	`dac4p`	D/A-converter 4: value after RESET

4.8.1.1 Reading the Timer Input Frequency (`tifreq`)

In cell `tifreq` the so-called timer input frequency is returned. This frequency corresponds to half the CPU-frequency. It is specified as 32-bit value in Hz (not like the specification of the CPU-frequency in cell `para` as 16-bit value in kHz!).

```
                     para [kHz]
tifreq [Hz] = ------------- · 1000
                        2
```

Example: In default setting of the CPU-frequency with 25.165 kHz (= $624D kHz) the value $C00000 would be returned.

The addresses of cell `tifreq` are represented in the overview table.

4.8.1.2 Reading the Current Conversion Time (`cnvtime`)

In this cell the current conversion time is returned in [ns]. After setting the conversion time via command `cmmd` = $30 the local software determines the actually attainable conversion time and writes it into this cell.

The value range of this parameter depends on the CPU-frequency selected (see chapter 'Specify Trigger Period Time via Timer (`cnvtime`)')

4.8.1.3 Reading 'D/A-Converter Values after RESET' (`dac1p, dac2p...`)

The default values of the D/A-converters which are to be taken over after a RESET are programmed via commands $20...$23. They can be read on the relative VMEbus addresses $D0...$DC (see previous table).

4.8.2 Output values of D/A-converters after a RESET (`dacxp`)

Via this command the output voltages of the D/A-converter channels are specified which are to apply to the outputs at P2 after a system re-boot.

cmmd [HEX]	para	Status readable at VMEbus address [HEX]	Status readable at local address [HEX]	Name of status cell	Parameter values [HEX]	

20
21
22
23
	WORD
	WORD
	WORD
	WORD
		D0
		D4
		D8
		DC
			68
			6A
			6C
			6E
				`dac1p`
				`dac2p`
				`dac3p`
				`dac4p`
					Value range: 8000...7FFF	Default value: 0000

					8000 U_{OUT} = -10 V : : 0000 U_{OUT} = 0 V : : 7FFF U_{OUT} = 10 V - 1 LSB	

4.8.3 Specify Trigger Period Time via Timer (`cnvtime`)

Here the desired period time for starting the conversions via he CPU-timer can be specified in nanoseconds. From this the firmware determines the actually possible conversion time and writes this value into cell `cnvtime`.

cmmd [HEX]	para	Status readable at VMEbus address [HEX]	Status readable at local address [HEX]	Name of status cell	Parameter values [HEX]	
30	LONG	C8, CC	64, 66	`cnvtime`	value range: \$0000.4E20 ... T_{MAX}	default value: \$0002.49E2
					\$0000.4E20 -> 20 µs 1*) : : \$0002.49E2 -> 149.9 µs : : \$004F.7967 -> 5.209 ms 2*)	

The maximum value T_{MAX}, which can be specified for the conversion time, depends on the CPU-frequency selected. It is determined as follows:

$$T_{MAX}\ [\text{ns}] = \frac{10^{9} \cdot 2^{17}}{\text{CPU\_frequency}\ [\text{Hz}]}$$

1*) In buffer mode the period time can be reduced down to 10 µs, if only data of an A/D-converter and a D/A-channel are processed.

2*) Here the maximum period time for the default setting of the CPU-frequency (25.165.000 Hz) is given.

4.8.4 Specify CPU-Frequency

Via this command the user can set the desired CPU-cycle frequency in kHz. From this the firmware determines an actually possible value which is as close to the desired value as possible. **This value cannot be read back. However, the value 'CPU-frequency/2' can be read in cell `tifreq`.**

The new cycle frequency can only be taken over, if the parameters are stored in the I²C-EEPROM and then a RESET is triggered!

`cmmd` [HEX]	para	Status readable at VMEbus address [HEX]	Status readable at local address [HEX]	Name of status cell	Parameter values [HEX]	
					Value range: $0000...61A8	Default value: $5A1C
31	WORD	C0, C4	60, 62	`tifreq·2`	$0000 -> 0 **k**Hz : : $3E80 -> 16,000 **k**Hz : : $5A1C -> 23,068 **k**Hz : : $61A8 -> 25,000 **k**Hz	

4.9 Control Parameters

In this area of the parameter buffer the control parameters for the A/D- and D/A-channels and interrupt handling are stored.

4.9.1 Overview of Readable Cells of the Parameter Buffer (Control Parameter)

The following table shows the memory cells of the parameter buffer in which the values of the control parameters can be read.

The values must be set by command call via the cells `cmmd` and `para`!

Relative address VMEbus [HEX]	Relative address local [HEX]	Length of status cell	Access	Names of status cells	Short description
140	A0	BYTE	read only	`vmelev`	current VMEbus interrupt level
142	A1	BYTE	read only	`vmevec`	VMEbus interrupt vector
144	A2	BYTE	read only	`muxmode`	multiplexer of A/D-inputs
146	A3	BYTE	read only	`dacmode`	multiplexer of D/A-outputs
148	A4	BYTE	read only	`trigmod`	trigger mode for A/D-inputs
14A	A5	BYTE	read only	`ldcmod`	multiplexer for LDAC-signal
14C	A6	BYTE	read only	`vadsrv`	selecting the A/D-data handling
14E	A7	BYTE	read only	`vstart`	selecting the A/D-channels whose values are to transferred to RAM
150	A8	BYTE	read only	`vend`	
152	A9	BYTE	read only	`vvtrg`	condition for VMEbus interrupt
154	AA	BYTE	read only	`vadres`	setting A/D-converter resolution
156	AB	BYTE	read only	`vdasrv`	selecting D/A-data handling
158	AC	WORD	read only	`vsmcnt`	number of A/D-values to be added
15C	AE	BYTE	read only	`dastart`	select first D/A-channel for buffer mode
15E	AF	BYTE	read only	`daend`	select last channel for buffer mode

Relative address VMEbus [HEX]	Relative address local [HEX]	Length of status cell	Access	Names of status cells	Short description
100	80	LONG	read only	structure `adbuf`	set up A/D-buffer (readable is A/D-buffer status structure)
120	90	LONG	read only	structure `dabuf`	set up D/A-buffer (readable is D/A-buffer status structure)
160	B0	LONG	read only	`vdasb`	D/A-buffer base
168	B4	WORD	read only	`vdamsk`	D/A-enable word

1FC
1F8
1F4
1F0
	FE
	FC
	FA
	F8
		WORD	read/ write !	`adstat0`
				`adstat1`
				`adstat2`
				`adstat3`
					status of A/D-data in RAM (attention: in contrast to the other cells these are set directly!)

In the following chapters setting the cells via command calls and the value range of the parameters will be described.

4.9.2 Set VMEbus Interrupt Level (`vmelev`)

This command sets the interrupt level of the VMEbus interrupt which can be triggered by the VME-AIO16.

`cmmd` [HEX]	para	Status readable at VMEbus address [HEX]	Status readable at local address [HEX]	Name of status cell	Parameter values [HEX]	
1	WORD	140	A0	`vmelev`	Value range: $0000...$0007	Default values (status cell): $05
					$0000 no VMEbus interrupt $0001 IRQ-level 1 : : $0007 IRQ-level7	

4.9.3 Set VMEbus Interrupt Vector (`vmevec`)

This command sets the interrupt vector which is transferred to the VMEbus in IACK-cycle, if the VME-AIO16 has triggered an IRQ.

`cmmd` [HEX]	para	Status readable at VMEbus address [HEX]	Status readable at local address [HEX]	Name of status cell	Parameter value [HEX]	
2	WORD	141	A1	`vmevec`	value range: $0000...00FF	Default value (status cell): $000F
					$0000 vector $00 : : $00FF vector $FF	

4.9.4 Set Multiplexers of A/D-Inputs (`muxmode`)

This command is automatically set by the firmware during the automatic adjustment. It is not be used by the user!

`cmmd` [HEX]	para	Status readable at VMEbus address [HEX]	Status readable at local address [HEX]	Name of status cell	Parameter value [HEX]	
3	WORD	144	A2	`muxmode`	Value range: $0000...$0003	Default value (status cell): $00
					$0000 $0001 $0002 $0003	external signal at A/Ds DACs at A/D-inputs +5V-ref. at A/D-inputs GND at all A/D-inputs

4.9.5 Set Multiplexers of D/A-Converter Outputs (`dacmode`)

This command is set by the firmware during the automatic adjustment. Afterwards it is set to '00'. The command must not be set by the user!

`cmmd` [HEX]	`para`	Status readable at VMEbus address [HEX]	Status readable at local address [HEX]	Name of status cell	Parameter values [HEX]	
4	WORD	145	A3	`dacmode`	value range: $0000...$0001	default value (status cell): $00
					$0000 $0001	D/A-outputs at P2 D/A-channels switched to A/D-outputs for internal adjustment

4.9.6 Select Trigger Source to Initiate A/D-Conversion (`trigmod`)

Here you can select how to initiate the conversion of A/D-channels. The triggering via software occurs via a write access to the relative local word address $7FFE0 (see chapter 'Shared Address Range'). When triggering via the CPU-timer, this has to be set accordingly (see command 'Specify Conversion Period for Timer')

`cmmd` [HEX]	para	Status readable at VMEbus address [HEX]	Status readable at local address [HEX]	Name of status cell	Parameter values [HEX]	
5	WORD	148	A4	`trigmod`	value range: $0000...$0002	default value (status cell): $00
					$0000 software trigger $0001 external trigger $0002 trigger via timer	

4.9.7 Select Source for LDAC of D/A-Converters (`ldcmod`)

By activating the LDAC-signal the D/A-converters adopt the new value at their outputs.

`cmmd` [HEX]	`para`	Status readable at VMEbus address [HEX]	Status readable at local address [HEX]	Name of status word	Parameter values [HEX]	
6	WORD	149	A5	`ldcmod`	value range: $0000...$0001	default value (status cell): $01
					$0000	LDAC is set when the new D/A-values are loaded of by write accessing the LDAC-address
					$0001	LDAC is set when the A/D-converters are convert-started

Mode $0000:
The LDAC-signal (SWLDAC) can be set immediately when the D/A-converters are loaded or separately by write-accessing a trigger address. The latter has got the advantage that all D/A-converters adopt their new value simultaneously at the output. The according addresses have already been described in chapter ‘VMEbus-Addresses and Addresses of Local CPU’.

Mode $0001:
The LDAC-signal is set again with every new convert-start signal of the A/D-converters. This causes a defined delay at the control loops between input capacity (A/D-input) and manipulated variable (D/A-output). New D/A-converter values can be set asynchronously to this.

4.9.8 Selecting the A/D-Data Handling (`vadsrv`)

`cmmd` [HEX]	`para`	Status readable at VMEbus address [HEX]	Status readable at local address [HEX]	Name of status cell	Parameter values [HEX]	
					value range: $0000...$000B	default value (status cell): $01
7	WORD	14C	A6	`vadsrv`	$0000	transfer of A/D-values into shared RAM (only for test and diagnosis)
					$0001	transfer of A/D-values into shared RAM via DMA-transfer
					$0002	as 01, but additionally determination of corrected A/D-values and storage into shared RAM
					$0003	as 01, but additionally adding of A/D-values and offset correction
					$000A	buffer mode: ADC-one-shot (see page 67)
					$000B	buffer mode: continuous (see page 67)

In mode 'Adding of A/D-values' the A/D-data is added, the offset corrected and the determined values are stored into a new buffer. The number of data to be added can be preset via the command 'Number of A/D-values to be added'. In mode 3, as in mode 2, in addition to the values corrected you can also read the A/D-values which have not been corrected.

Modes 'ADC-one-shot' and 'ADC-buffer mode' will be described in a separate chapter in detail, because they differ fundamentally from other operating modes (see page 67).
It has to be taken into account that the buffer modes are not being stored when the parameters are stored in the I²C-EEPROM. Only modes are stored for which the most significant parameter bit is set to '0'.

The VME-AIO16 requires different handling periods for the modes described. In addition the transfer speed depends on the CPU-cycle frequency and the number of converted channels.
The following table shows exemplary the time required from initiating the conversion to setting the local 'data read' flag. This flag signalizes the reception of data in the shared RAM and triggers (if trigger status activated) the VMEbus interrupt.

The values specified are only valid, if the local SRAM-buffer is designed as 'FAST SRAM' (access time FAST-SRAM ca. 15 ns). For hardware versions succeeding '0' (see page 15) a FAST-RAM is always equipped.

CPU-cycle frequency	para	Handling time in dependence on the number of A/D-channels whose data are to transferred into the RAM in [µs]		
		16 channels	8 channels	1 channel
25 MHz	$0000	<23	15	10
	$0001	<18	12	8
	$0002	<75	43	18
	$0003 [1*]	<50	26	12

[1*] Times specified are valid for status cell `vsmcnt` = 4.

Handling speed in dependence of the number of A/D-channels and D/A-channels in buffer mode:

Number of A/D-converter channels	Number of D/A-converter channels	Number of frames [2*] per second
1	1	118 000
2	1	110 000
2	2	103 000
6	4	74 300
16	4	50 700

[2*] A frame contains the data of all previously defined A/D or D/A-channels.

For more combinations of the number of A/D and DA-channels the number of frames per second can be determined with the following equation:

```
frames/time = 1 / (number_ADC · 0.625µs + number_DAC · 0.675µs + 6.2µs + 1µs)
```

4.9.9 Select A/D-Channels (`vstart, vend`)

The commands 'select first A/D-channel for reading-back' and 'select last A/D-channel for reading-back' are used in order to relieve the local CPU, if not all 16 A/D-channels are required. By doing this the sampling rate of the remaining A/D-channels can be increased.

`cmmd` [HEX]	para	Status readable at VMEbus address [HEX]	Status readable at local address [HEX]	Name of status cell	Parameter values [HEX]	
8 9	WORD WORD	14D 150	A7 A8	`vstart` `vend`	value range of parameters: see below	default value (status cells): `vstart` = $01, `vend` = $10

Value range of parameters:

The positive number 1 to 16 are assigned to the 16 A/D-converter channels.

The negative numbers '-1' to '-8' are assigned to the eight auxiliary inputs.

The following examples represent the function of the parameters:

Value of parameters		Channels selected for reading-back	
`vstart`	`vend`	Auxiliary inputs	A/D-converter channels
1	16	-	ADC_1, ADC_2, ... ADC_16
3	3	-	ADC_3
-1	16	Aux_In_1	ADC_1, ADC_2, ... ADC_16
-6	8	Aux_In_6, Aux_In_5, ... Aux_In_1	ADC_1, ADC_2, ... ADC_8
-8	16	Aux_In_8, Aux_In_7, ... Aux_In_1	ADC_1, ADC_2, ... ADC_16

4.9.10 Select Source for VMEbus Interrupt (`vvtrg`)

Via this command you can select under which condition an VMEbus interrupt can be triggered by the VME-AIO16. Three possibilities are available:

1. **no interrupt**

2. **IRQ_on_Acknowledge**
   This interrupt is triggered after the A/D-data is available in the SRAM-buffer and the status words `adstat0` or `adstat1` have been set. The status words can therefore be evaluated after the interrupt.

3. **IRQ_on_DMA_done**
   This interrupt is triggered after the A/D-data is available in the SRAM-buffer. The status words will be set at a later period, however (see the following figure 4.8.1). If they are requested immediately after the IRQ, it might be that they have not yet been updated.

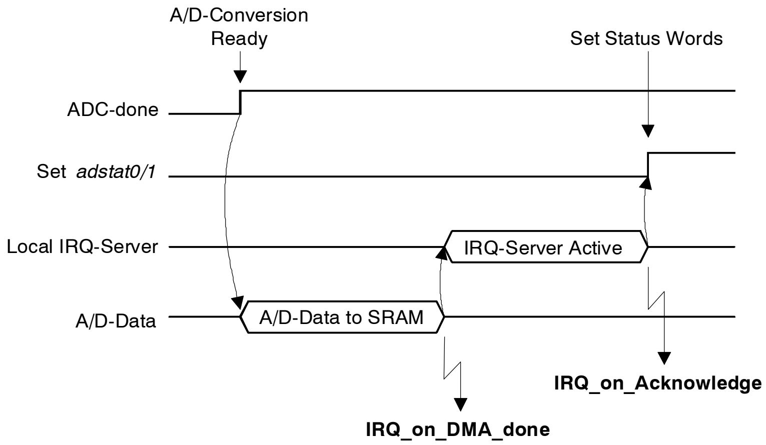

Fig. 4.8.1: Time of interrupt trigger

<table>
<tr><th>cmmd
[HEX]</th><th>para</th><th>Status readable at VMEbus address [HEX]</th><th>Status readable at local address [HEX]</th><th>Name of status cell</th><th colspan="2">Parameter values
[HEX]</th></tr>
<tr><td rowspan="4">A</td><td rowspan="4">WORD</td><td rowspan="4">151</td><td rowspan="4">A9</td><td rowspan="4">vvtrg</td><td>value range:
$0000,
$007F,
$00FF</td><td>default value (status cell):
$00</td></tr>
<tr><td>$0000</td><td>no VMEbus interrupt</td></tr>
<tr><td>$007F</td><td>IRQ_on_Acknowledge</td></tr>
<tr><td>$00FF</td><td>IRQ_on_DMA_done</td></tr>
</table>

4.9.11 Set A/D-Converter Resolution (`vadres`)

Into this cell the local firmware specifies with which A/D-converter type the VME-AIO16 has been equipped. The cell must not be changed by the user.

<table>
<tr><th>cmmd
[HEX]</th><th>para</th><th>Status readable at VMEbus address [HEX]</th><th>Status readable at local address [HEX]</th><th>Name of status cell</th><th colspan="2">Parameter values
[HEX]</th></tr>
<tr><td rowspan="2">B</td><td rowspan="2">WORD</td><td rowspan="2">154</td><td rowspan="2">AA</td><td rowspan="2">vadres</td><td>value range:
$0000...$00FF</td><td>default value (status cell):
$00</td></tr>
<tr><td colspan="2">$0000 16-bit A/D-converter equipped
≠ $0000 12-bit A/D-converter equipped</td></tr>
</table>

4.9.12 Selecting the D/A-Data Handling (`vdasrv`)

Into this cell various modes for setting the D/A-converter channels can be specified.

`cmmd` [HEX]	para	Status readable at VMEbus address [HEX]	Status readable at local address [HEX]	Name of status cell	Parameter values [HEX]	
C	WORD	155	AB	`vdasrv`	value range: $0000...$0002	default value (status cell): $00
					$0000 normal operating mode that means setting the channels via individual VMEbus write accesses (see page 66) $0001 buffer mode: DAC-one-shot (see page 67) $0002 buffer mode: DAC-wrap mode (continuous) (see page 67) $0003 buffer mode: selective DAC-buffer	

When storing the parameters in the I²C-EEPROM ‘0’ is always stored for `vadsrv`, that means the setting of the buffer mode cannot be stored permanently!

4.9.13 Number of A/D-Values to be Added (`vsmcnt`)

This command is only significant, if the mode 'adding the A/D-values' has been selected via command '`vadsrv`'.

`cmmd` [HEX]	`para`	Status readable at VMEbus address [HEX]	Status readable at local address [HEX]	Name of status cell	Parameter values [HEX]	
D	WORD	158	AC	`vsmcnt`	value range: $0004...$7FFF	Default value (WORD-status cell): $0000
					number of A/D-values to be added: (values from '4' are useful)	

The local software determines from the parameter value determined the next possible number of values to be added. When reading the parameter value the determined, actually set value is returned.

4.9.14 Set Up A/D-Buffer (`adbuf`)

Via this command a memory area is reserved in the shared RAM for the A/D-data which are to be recorded in buffer mode.

<table>
<tr><th>cmmd
[HEX]</th><th>para</th><th>VMEbus-start address of structure [HEX]</th><th>local start address of structure [HEX]</th><th>Name of structure</th><th colspan="2">Parameter values
[HEX]</th></tr>
<tr><td rowspan="2">E</td><td>Upper WORD</td><td rowspan="2">100</td><td rowspan="2">80</td><td rowspan="2">adbuf</td><td>number of frames per buffer</td><td>value range: $0000...$7FFF</td></tr>
<tr><td>Lower WORD</td><td>number of buffers</td><td>value range: $0000...$7FFF</td></tr>
</table>

The function of the buffer mode and the structure of buffers and frames will be described in a separate chapter from page 67 in greater detail.
The parameters of this command transferred cannot be stored in the I^2C-EEPROM.

4.9.15 Set Up D/A-Buffer (`dabuf`)

Via this command a memory area is reserved in the shared RAM for the D/A-data which are to be set in buffer mode.

<table>
<tr><th>cmmd
[HEX]</th><th>para</th><th>VMEbus-start address of structure [HEX]</th><th>local start address of structure [HEX]</th><th>Name of structure</th><th colspan="2">Parameter values
[HEX]</th></tr>
<tr><td rowspan="2">F</td><td>Upper WORD</td><td rowspan="2">120</td><td rowspan="2">90</td><td rowspan="2">dabuf</td><td>number of frames per buffer</td><td>value range: $0000...$FFFF</td></tr>
<tr><td>Lower WORD</td><td>number of buffers</td><td>value range: $0000...$FFFF</td></tr>
</table>

The function of the buffer mode and the structure of buffers and frames will be described in a separate chapter from page 67 in greater detail.
The parameters of this command transferred cannot be stored in the I^2C-EEPROM.

4.9.16 Select D/A-Channels (`dastart, daend`)

The commands ‘select first D/A-channel for reading-back’ and ‘select last D/A-channel for reading-back’ are used to relieve the local CPU, if not all four D/A-channels are required. By doing this the conversion rate of the other D/A-channels and the sampling rate of the A/D-channels can be increased.

<table>
<tr><th><code>cmmd</code>
[HEX]</th><th>para</th><th>Status readable at VMEbus address [HEX]</th><th>Status readable at local address [HEX]</th><th>Name of status cell</th><th colspan="2">Parameter values
[HEX]</th></tr>
<tr><td rowspan="2">10

11</td><td rowspan="2">WORD

WORD</td><td rowspan="2">15C

15E</td><td rowspan="2">AE

AF</td><td rowspan="2"><code>dastart</code>

<code>daend</code></td><td>value range of parameters:
$0001...$0004
$0001 ≤ first <code>channel</code> ≤ last channel ≤ $0004</td><td>default values (status cells):
<code>dastart</code> = $01,
<code>daend</code> = $04</td></tr>
<tr><td colspan="2">examples:
<code>dastart</code> = $02,
<code>daend</code> = $04 -> D/A-channels 2 to 4 are evaluated
<code>dastart</code> = $03,
<code>daend</code> = $03 -> only D/A-channel 3 is evaluated</td></tr>
</table>

4.9.17 Selective D/A-Buffer

cmmd [HEX]	para	Status readable at VMEbus address [HEX]	Status readable at local address [HEX]	Name of status cell	Name of parameter	Parameter values [HEX]
12	upper WORD	40	20	cstat	buffer_no	value range: 0000...FFFF
	lower WORD	40	20	cstat	dac_mask	value range: 0000...000F

Via this command the D/A-buffer and the D/A-converter channel(s) are selected. Before this command is called the D/A-buffers have to be set up via command $F (set up D/A-buffer).

buffer_no = 1, 2, ...N (N = Number_of_Buffers)

Number _of_Buffers = maximum number of D/A-buffers after calling command $F (see D/A-buffer-status structure on page 69)

The buffer start address can be read in the cell vdacsb ($B0) (see page 70).

dac_mask: This parameter determines the D/A-converter channels involved in the buffer.

dac_mask : bit 0 - DAC 1
bit 1 - DAC 2
bit 2 - DAC 3
bit 3 - DAC 4

The value of dac_mask can be read in the cell vdamsk ($B4).

4.10 A/D-Converter Status (`adstat0...adstat3`)

In these cells the current status of the A/D-conversion can be read. The cell can, for example, be requested by polling to find out about the completion of the conversion, if it is not to be followed by a VMEbus interrupt.

After the data have been read the respective cell should be reset to value ≠ FFFF, because the server does not reset it automatically. In contrast to the other parameters these cells are set directly, that means they are not set via the commands `cmmd` and `para`!

The cell `adstat0` shows that the incorrect data are available in the SRAM while the cell `adstat1` shows that the corrected values are available in the SRAM.

The cell `adstat2` is not being used at the moment. It is reserved for future applications.

Cell `adstat3` shows whether the added A/D-values in operating mode 'adding the A/D-values' have been determined.

Relative address VMEbus [HEX]	Relative address local [HEX]	Length	Access	Name of status cell	Parameter values [HEX]

1FC
1F8
1F4
1F0
	FE
	FC
	FA
	F8
		WORD	read/
			write
				`adstat0`
				`adstat1`
				`adstat2`
				`adstat3`
					value range: $0000...$FFFF
					$FFFF new A/D-data have been stored in the RAM
					≠ $FFFF new data have not yet been received

This page has intentionally been left blank.

5. Reading the A/D-Measured Values and Setting the D/A-Converters

5.1 Reading the A/D-Measured Values

Below the cells of the analog measured values will be described. These cells are set by the local firmware and must only be set!

Note:	Correcting the A/D-data with user parameters: The parameters `offs01...offs16`, `offs01.summed...offs16.summed`, and `scale01...scale16` determined during self test can be changed by the user for *private correction*!

5.1.1 Crude Data of A/D-Converters - `adwert01...adwert16`

Here the A/D-values which are put out by the A/D-converters can be read. The values have not been corrected.

Relative address VMEbus [HEX]	Relative address local [HEX]	Length	Access	Cell contents	Short description	
200	100	WORD	read only	`adwert01`	crude data of A/D-converters	1
204	102			`adwert02`	:	2
208	104			`adwert03`	:	3
20C	106			`adwert04`	:	4
210	108			`adwert05`	:	5
214	10A			`adwert06`	:	6
218	10C			`adwert07`	:	7
21C	10E			`adwert08`	:	8
220	110			`adwert09`	:	9
224	112			`adwert10`	:	10
228	114			`adwert11`	:	11
22C	116			`adwert12`	:	12
230	118			`adwert13`	:	13
234	11A			`adwert14`	:	14
238	11C			`adwert15`	:	15
23C	11E			`adwert16`	crude data of A/D-converter	16

Cell contents	Parameter values [HEX]
`adwert01...adwert16`	value range: 8000...7FFF
	8000 -10 V : : 0000 0 V : 7FFF + 10 V -1 LSB

5.1.2 Corrected A/D-Converter Data - `advac01...advac16`

Here the A/D-values with corrected gain and offset error can be read after the conversion, if the conversion mode '2' (see command `vadsrv`) has been selected.

Relative address VMEbus [HEX]	Relative address local [HEX]	Length	Access	Cell contents	Short description
240	120	WORD	read only	`advac01`	corrected A/D-data channel 1
244	122			`advac02`	: 2
248	124			`advac03`	: 3
24C	126			`advac04`	: 4
250	128			`advac05`	: 5
250	12A			`advac06`	: 6
258	12C			`advac07`	: 7
25C	12E			`advac08`	: 8
260	130			`advac09`	: 9
264	132			`advac10`	: 10
268	134			`advac11`	: 11
26C	136			`advac12`	: 12
270	138			`advac13`	: 13
274	13A			`advac14`	: 14
278	13C			`advac15`	: 15
27C	13E			`advac16`	corrected A/D-data channel 16

Cell contents	Parameter values [HEX]
`advac01...advac16`	value range: 8000...7FFF
	8000 -10 V : : 0000 0 V : : 7FFF + 10 V -1 LSB

The corrected values are determined as follows:

```
                             scale
advac = (advalue - offs)(1 + ------)
                              2^16
```

$$advac = (advalue - offs)\left(1 + \frac{scale}{2^{16}}\right)$$

with

`advalue` = crude data of A/D-converter channels
`offs` = offset value of A/D-converter channels
`scale` = gain correction factor * 2^{16}
`advac` = corrected A/D-values

The correction factors will be described from page 54.

5.1.3 Added A/D-Converter Data - `adc01.summed...adc16.summed`

Here the A/D-values determined after conversion and adding can be read, if the conversion mode '3' (see command `vadsrv`) has been selected.

Relative address VMEbus [HEX]	Relative address local [HEX]	Length	Access	Cell contents	Short description
280	140	LONG	read only	`adc01.summed`	added A/D-data channel 1
288	144			`adc02.summed`	: 2
290	148			`adc03.summed`	: 3
298	14C			`adc04.summed`	: 4
2A0	150			`adc05.summed`	: 5
2A8	154			`adc06.summed`	: 6
2B0	158			`adc07.summed`	: 7
2B8	15C			`adc08.summed`	: 8
2C0	160			`adc09.summed`	: 9
2C8	164			`adc10.summed`	: 10
2D0	168			`adc11.summed`	: 11
2D8	16C			`adc12.summed`	: 12
2E0	170			`adc13.summed`	: 13
2E8	174			`adc14.summed`	: 14
2F0	178			`adc15.summed`	: 15
2F8	17C			`adc16.summed`	added A/D-data channel 16

Cell contents	Parameter values [HEX]
`adc01.summed ... adc16.summed`	value range: 80000000...7FFFFFFF
	80000000 -10 V : : 00000000 0 V : : 7FFFFFFF + 10 V -1 LSB(32)

After adding and re-formatting the A/D-values the offset is also corrected with the data `offs01.smmd`...`offs16.smmd`. The A/D-converter value determined this way is always shown as longword left-aligned. If only an evaluation of 16 bits is required, the two MSB have to be read.

5.1.4 Offset of A/D-Converter - `offs01...offs16`

These measured values have been adopted during self test. They are the measured values which result, if the A/D-converter inputs are applied to GND via multiplexers and therefore represent the offset error of the input circuit.

Relative address VMEbus [HEX]	Relative address local [HEX]	Length	Access	Cell contents	Short description	
400	200			`offs01`	offset of A/D-converter	1
404	202			`offs02`	:	2
408	204			`offs03`	:	3
40C	206			`offs04`	:	4
410	208			`offs05`	:	5
414	20A			`offs06`	:	6
418	20C			`offs07`	:	7
41C	20E	WORD	read/ (write)	`offs08`	:	8
420	210			`offs09`	:	9
424	212			`offs10`	:	10
428	214			`offs11`	:	11
42C	216			`offs12`	:	12
430	218			`offs16`	:	13
434	21A			`offs14`	:	14
438	21C			`offs15`	:	15
43C	21E			`offs16`	offset of the A/D-converter	16

Cell contents	Parameter value [HEX]
`offs01...offs16`	value range: 8000...7FFF
	8000 -10 V : : 0000 0 V : : 7FFF + 10 V -1 LSB

5.1.5 A/D-Values at Reference Voltage Applied - `ref501...ref516`

These measured value have been adopted during self test. They are measured values which result, if the A/D-converter inputs have been applied to the +5 V-reference voltage via multiplexers. The offset error has already been considered for these values.

Relative address VMEbus [HEX]	Relative address local [HEX]	Length	Access	Cell contents	Short description	
440	220	WORD	read only	ref501	A/D-value at +5V-ref channel	1
444	222			ref502	:	2
448	224			ref503	:	3
44C	226			ref504	:	4
450	228			ref505	:	5
454	22A			ref506	:	6
458	22C			ref507	:	7
45C	22E			ref508	:	8
460	230			ref509	:	9
464	232			ref510	:	10
468	234			ref511	:	11
46C	236			ref512	:	12
470	238			ref513	:	13
474	23A			ref514	:	14
478	23C			ref515	:	15
47C	23E			ref516	A/D-value at +5V-ref channel	16

Cell contents	Parameter values [HEX]
ref501...ref516	value range: 8000...7FFF
	8000 -10 V : : 0000 0 V : : 7FFF + 10 V -1 LSB

The values are determined as follows:

$$ref5 = (advalue_{at\ 5V} - offs)$$

with
$advalue_{at\ 5V}$ = crude data of A/D-converter channels with 5V-reference voltage applied
`offs` = offset value of A/D-converter channels
`ref5` = A/D-value at + 5 V with correct offset

5.1.6 A/D-Values with D/A-Voltage of 0 V Applied: D/A-Offset (`adc001...adc016`)

These measured values have been adopted during self test. They are measured values which result, if the A/D-converter inputs are connected to the D/A-converter outputs via multiplexers. The D/A-converter are set to 0 V in this case.

From the measured values the offset error of the D/A-converters can be determined.

Because of the higher number of A/D-converters each D/A-converter is measured four times. These four measured values should be similar.

Relative address VMEbus [HEX]	Relative address local [HEX]	Length	Access	Cell contents	Short description
480	240	WORD	read only	`adc001`	A/D-value at DAC1=0 V channel 1
484	242			`adc002`	: DAC2 : 2
488	244			`adc003`	: DAC3 : 3
48C	246			`adc004`	: DAC4 : 4
490	248			`adc005`	: DAC1 : 5
494	24A			`adc006`	: DAC2 : 6
498	24C			`adc007`	: DAC3 : 7
49C	24E			`adc008`	: DAC4 : 8
4A0	250			`adc009`	: DAC1 : 9
4A4	252			`adc010`	: DAC2 : 10
4A8	254			`adc011`	: DAC3 : 11
4AC	256			`adc012`	: DAC4 : 12
4B0	258			`adc013`	: DAC1 : 13
4B4	25A			`adc014`	: DAC2 : 14
4B8	25C			`adc015`	: DAC3 : 15
4BC	25E			`adc016`	A/D-value at DAC4=0 V channel 16

Cell contents	Parameter values [HEX]
`adc001...adc016`	value range: 8000...7FFF
	8000 -10 V : : 0000 0 V : : 7FFF + 10 V -1 LSB

5.1.7 A/D-Values at Applied D/A-Value $7F00: D/A-Gain (`adc101...adc116`)

These measured values have been adopted during self test. They are measured values which result, if the A/D-converter inputs are connected to the D/A-converter outputs via multiplexers. The D/A-converters have got the value $7F00 in this case. From the measured value the gain error of the D/A-converters can be determined. Because of the higher number of A/D-converters, each D/A-converter is measured four times. These four measured values should be similar.

The offset error and the gain correction of the A/D-channels have already been considered for these values. Furthermore the values have been corrected with the offset error of the D/A-channels.

Relative address VMEbus [HEX]	Relative address local [HEX]	Length	Access	Cell contents	Short description
4C0	260	WORD	read only	`adc101`	A/D-value at DAC1=$7F00 channel 1
4C4	262			`adc102`	: DAC2 : 2
4C8	264			`adc103`	: DAC3 : 3
4CC	266			`adc104`	: DAC4 : 4
4D0	268			`adc105`	: DAC1 : 5
4D4	26A			`adc106`	: DAC2 : 6
4D8	26C			`adc107`	: DAC3 : 7
4DC	26E			`adc108`	: DAC4 : 8
4E0	270			`adc109`	: DAC1 : 9
4E4	272			`adc110`	: DAC2 : 10
4E8	274			`adc111`	: DAC3 : 11
4EC	256			`adc112`	: DAC4 : 12
4F0	278			`adc113`	: DAC1 : 13
4F4	27A			`adc114`	: DAC2 : 14
4F8	27C			`adc115`	: DAC3 : 15
4FC	27E			`adc116`	A/D-value at DAC4=$7F00 chan. 16

Cell contents	Parameter values [HEX]
`adc101...adc116`	value range: 8000...7FFF
	8000 -10 V : : 0000 0 V : : 7FFF + 10 V -1 LSB

The values are determined as follows:

$$adc1 = (advalue_{at\ DAC=7F00} - adc0)$$

with

$advalue_{at\ DAC=7F00}$	=	measured A/D-value, if D/A-converters are set to $7F00
`adc0`	=	offset value of D/A-converter channels
`adc1`	=	offset-corrected D/A-value at $7F00 to determine the GAIN-correction

5.1.8 A/D-Converter: Gain-Correction Factors (`scale01...scale16`)

The gain-correction factor is used to determine the corrected A/D-values. The factor is determined after measuring the A/D-values with the +5 V-reference voltage.

Relative address VMEbus [HEX]	Relative address local [HEX]	Length	Access	Cell contents	Short description
500	280	WORD	read/ (write)	`scale01`	gain factor of A/D-converter 1
504	282			`scale02`	: 2
508	284			`scale03`	: 3
50C	286			`scale04`	: 4
510	288			`scale05`	: 5
514	28A			`scale06`	: 6
518	28C			`scale07`	: 7
51C	28E			`scale08`	: 8
520	290			`scale09`	: 9
524	292			`scale10`	: 10
528	294			`scale11`	: 11
52C	296			`scale12`	: 12
530	298			`scale13`	: 13
534	29A			`scale14`	: 14
538	29C			`scale15`	: 15
53C	29E			`scale16`	gain factor of D/A-converter 16

Cell contents	Parameter values [HEX]
`scale01...scale16`	value range: 8000...0000...7FFF

$$\text{scale} = \frac{\text{addesvalue}-\text{advalue}}{\text{addesvalue}}\ 2^{16}$$

$$\text{advalue}_z = \text{advalue}_{in}-\text{offs}$$

$$\text{advalue}_{corr} = \text{advalue}_z + \frac{\text{advalue}_z * \text{scale}}{2^{16}}$$

with

`advalue` = measured A/D-value with applied +5 V-reference voltage
`addesvalue` = ideal desired value at +5V
`scale` = gain correction factor

5.1.9 Offset Values of Added 32-Bit-A/D-Values (`offs01.summed...offs16.summed`)

The values which can be read here correspond to those measured and added at 0 V in the self test phase. They can be changed via VMEbus-accesses.

Relative address VMEbus [HEX]	Relative address local [HEX]	Length	Access	Cell contents	Short description
540	2A0	LONG	read/ write	`offs01.summed`	32-bit offset A/D-channel 1
548	2A4			`offs02.summed`	: 2
550	2A8			`offs03.summed`	: 3
558	2AC			`offs04.summed`	: 4
560	2B0			`offs05.summed`	: 5
568	2B4			`offs06.summed`	: 6
570	2B8			`offs07.summed`	: 7
578	2BC			`offs08.summed`	: 8
580	2C0			`offs09.summed`	: 9
588	2C4			`offs10.summed`	: 10
590	2C8			`offs11.summed`	: 11
598	2CC			`offs12.summed`	: 12
5A0	2D0			`offs13.summed`	: 13
5A8	2D4			`offs14.summed`	: 14
5B0	2D8			`offs15.summed`	: 15
5B8	2DC			`offs16.summed`	32-bit offset A/D-channel 16

Cell contents	Parameter values [HEX]
`offs01.summed ... offs16.summed`	value range: 80000000...7FFFFFFF
	80000000 -10 V : : 00000000 0 V : : 7FFFFFFF + 10 V -1 LSB

5.2 Auxiliary Inputs

The crude data of the auxiliary inputs are local memory cells of the VME-AIO16, which can be set via the VMEbus. The data of these memory cells can be handled like the A/D-converter data by the local firmware.

5.2.1 Crude Data of Auxiliary Inputs `auxin1...auxin8`

Here the A/D-values can be read which are given by the VMEbus data interface. The values have not been corrected.

Relative address VMEbus [HEX]	Relative address local [HEX]	Length	Access	Cell contents	Short description	
300	180	WORD	read only	`auxin8`	crude data of auxiliary input	8
304	182			`auxin7`	:	7
308	184			`auxin6`	:	6
30C	186			`auxin5`	:	5
310	188			`auxin4`	:	4
314	18A			`auxin3`	:	3
318	18C			`auxin2`	:	2
31C	18E			`auxin1`	crude data of auxiliary input	1

Cell contents	Parameter values [HEX]
`auxin1...auxin8`	value range: 8000...7FFF
	8000 : 0000 : 7FFF

5.2.2 Corrected Data of Auxiliary Inputs - `auxcr1...auxcr8`

Relative address VMEbus [HEX]	Relative address local [HEX]	Length	Access	Cell contents	Short description
320	190	WORD	read only	`auxcr8`	corrected data of auxiliary input 8
324	192			`auxcr7`	: 7
328	194			`auxcr6`	: 6
32C	196			`auxcr5`	: 5
330	198			`auxcr4`	: 4
334	19A			`auxcr3`	: 3
338	19C			`auxcr2`	: 2
33C	19E			`auxcr1`	corrected data of auxiliary input 1

Cell contents	Parameter values [HEX]
`auxcr1...auxcr8`	value range: 8000...7FFF
	8000 : 0000 : 7FFF

The corrected values are determined as follows:

$$auxcr = (auxvalue - offs)\left(1 + \frac{scale}{2^{16}}\right)$$

with
`auxvalue` = crude data of auxiliary inputs
`offs` = offset value of auxiliary inputs
`scale` = gain correction factor * 2^{16}
`auxcr` = corrected auxiliary input values

The correction factors will be described from page 64.

5.2.3 Offset of Auxiliary Inputs - `ofsax1...ofsax8`

Relative address VMEbus [HEX]	Relative address local [HEX]	Length	Access	Cell contents	Short description
5C0	2E0	WORD	read only	`ofsax8`	offset of auxiliary input 8
5C4	2E2			`ofsax7`	: 7
5C8	2E4			`ofsax6`	: 6
5CC	2E6			`ofsax5`	: 5
5D0	2E8			`ofsax4`	: 4
5D4	2EA			`ofsax3`	: 3
5D8	2EC			`ofsax2`	: 2
5DC	2EE			`ofsax1`	offset of auxiliary input 1

Cell contents	Parameter values [HEX]
`ofsax1...ofsax8`	value range: 8000...7FFF
	8000 : 0000 : 7FFF

5.2.4 Gain Correction Factors of Auxiliary Inputs (`sclax1...sclax8`)

Relative address VMEbus [HEX]	Relative address local [HEX]	Length	Access	Cell contents	Short description
5E0	280	WORD	read only	`sclax8`	gain factor of auxiliary input 8
5E4	282			`sclax7`	: 7
5E8	284			`sclax6`	: 6
5EC	286			`sclax5`	: 5
5F0	288			`sclax4`	: 4
5F4	28A			`sclax3`	: 3
5F8	28C			`sclax2`	: 2
5FC	28E			`sclax1`	gain factor of auxiliary input 1

Cell contents	Parameter values [HEX]
`sclax1...sclax8`	value range: 8000...FFFF

$$auxcr = (auxvalue - ofsax)\left(1 + \frac{sclax}{2^{16}}\right)$$

5.3 Setting the D/A-Converters

The D/A-converters are set via VMEbus-write accesses to the D/A-converter addresses. The D/A-converter addresses are listed on page 8.

5.3.1 Notes on Setting the D/A-Converters

When write-accessing the D/A-converters cells of the shared RAM are assigned with the D/A-values simultaneously. A write access leads to the SRAM and the previously set value can be read-back.

5.3.2 Data Format of D/A-Converter Values

The data format for the D/A-converter values is as follows:

D/A-converter data [HEX]	Output voltage value [V]
8000	-10 V
:	:
0000	0 V
:	:
7FFF	+ 10 V -1 LSB

6. Buffer Mode

6.1 Overview

In comparison to the previously described operating modes, the buffer mode offers the possibility to store the A/D-converter measured values sequentially. The measured values are also stored in an area of the shared RAM, the so-called ‘buffer’, in the chronological order in which they were determined. The sequential storing also results in an increased conversion rate. Reference values to the conversion rates can be found on page 38.
While the analog input dates are stored in the buffer mode the statistical value, such as ‘crude data’ and ‘corrected data’ are supplied.

6.2 Buffer and Frames

The A/D-data and the D/A-data is stored in buffer mode organized into ‘buffers’ and ‘frames’. A frame contains the A/D or D/A-values of all converters enabled for processing. A frame can therefore contain (-8)...16 A/D-values for the analog inputs. The number of channels to be converted is defined, as in the other operating modes, via parameters `vstart` and `vend`. A frame with D/A-converter data can contain one to four D/A-values. The selection is made via parameters `dastart` and `daend`.

Example for the structure of an A/D-frame:

A/D-Value(`vstart`)	A/D-Value(`vstart`+1)	A/D-Value(`vstart`+2)	...	A/D-Value(`vend`)

Example for the structure of a D/A-frame:

D/A-Value(`dastart`)	D/A-Value(`dastart`+1)	D/A-Value(`dastart`+2)	...	D/A-Value(`daend`)

Several frames are combined to one buffer. For the A/D-converter data and for the D/A-converter data the number of frames per buffer can be defined. In addition the number of buffers desired can be specified.

The maximum number of buffers is limited by the memory capacity available. The local firmware has got a memory capacity of $800 (VMEbus address) to $7FDFE. If now, for example, first the memory area for the A/D-data were requested, an area from $800 rising is reserved. Incase of the D/A-data a memory area of $7FDFE falling were reserved. The limits of the actually available memory capacity is written into the A/D or D/A-buffer-status structure by the firmware .
This kind of memory capacity assignation has got the advantage that the space available can be used more for A/D or D/A-values, depending on the requirements.

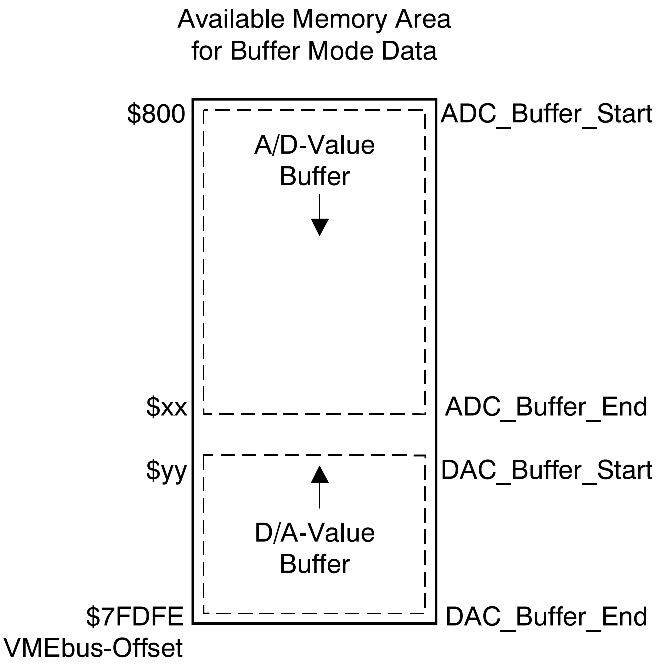

Fig. 6.2.1: Assignment of available memory space with data of the A/D-buffer and the D/A-buffer

6.3 A/D and D/A-Buffer Status Structures

The A/D-buffer status structure and the D/A-buffer status structure are memory areas in which information about the respective buffers are stored. Here the user can read where the memory area of the A/D and D/A-buffers made available actually starts and ends.

6.3.1 A/D-Buffer Status Structure

Address offset VME (local) [HEX]	Length	Cell contents	Default values [HEX]	Comment
100 (80)	LONG	ADC_Buffer_Start_Offset	0000 0800	the offset is returned within the memory for VMEbus accesses
108 (84)	LONG	ADC_Buffer_End_Offset	0000 0800	
110 (88)	WORD	Number_of_ADCs_per_Frame	0	value range: 1...16 [1*)]
114 (8A)	WORD	Frames_per_Buffer	0	e.g. $400 for 1024
118 (8C)	WORD	Number_of_Buffers	0	1...N
11C (8E)	WORD	Buffer_Number_in_Work	0	1...N

[1*)] Number_of_ADCs_per_Frame = `vend - vstart` +1

6.3.2 D/A-Buffer Status Structure

Address Offset VME (local) [HEX]	Length	Cell contents	Default values [HEX]	Comment
120 (90)	LONG	D/A_Buffer_Start_Offset	0007 FE00	the offset is returned within the memory for VMEbus accesses
128 (94)	LONG	D/A_Buffer_End_Offset	0007 FE00	
130 (98)	WORD	Number_of_D/As_per_Frame	0	value range: 1...4 [2*)]
134 (9A)	WORD	Frames_per_Buffer	0	e.g. $400 for 1024
138 (9C)	WORD	Number_of_Buffers	0	1...N
13C (9E)	WORD	Buffer_Number_in_Work	0	1...N

[2*)] Number_of_D/As_per_Frame = `daend - dastart` +1

VME-AIO16**6.3.3 Selective D/A-Buffer-Structure**

Address Offset VME (local) [HEX]	Length	Cell contents	Comment VME-AIO16
160 (B0)	LONG	Pointer_to_Buffer (`vdasb`)	D/A-buffer base address
168 (B4)	WORD	Mask (`vdamsk`)	D/A-enable word

6.4 Chronological Course

The buffer mode is active immediately following the respective command ($7 or $C)! The following points, however, must apply:
- `vend`, `vstart`, `daend` and `dastart` must have been set and
- the memory area for the buffers must have been designed (`adbuf`, `dabuf`).

6.4.1 Example for *Synchronous* Start

In order to synchronously start the A/D-buffer mode and the D/A-buffer mode the following steps have to be carried out:

No.	Command	`cmmd` [HEX]	`para` [DEC]
1	select source for trigger of A/D-conversion	5	`trigmod` = 00 (software trigger)
2	select first an last A/D-channel for conversion	8, 9	`vstart` = ... `vend` = ...
3	select first and last D/A-channel for conversion	10, 11	`dastart` = ... `daend` = ...
4	set up A/D-buffer [1*)]	E	`adbuf` = ...
5	set up D/A-buffer [1*)]	F	`dabuf` = ...
6	select A/D-data processing (activate buffer mode)	7	`vadsrv` = 10 or 11
7	select D/A-data processing (activate buffer mode)	C	`vdasrv` = 1 or 2
8	select source for LDAC of D/A-conversion	6	`ldcmod` = 1
9	specify period time for trigger of conversion by timer	30	`cnvtime` = yx
10	select source for trigger of A/D-conversion	5	`trigmod` = 02 (trigger by timer)

[1*)] Once after RESET or changing of `vstart/vend` or `dastart/daend`.

6.4.2 Timing Diagram of an A/D-D/A-Transfer

The timing of the D/A-conversion is linked to the one of the A/D-conversion, in order to enable a synchronous I/O-transfer. To make sure that the D/A-converter has got a defined initial value, it is preset with the first value of the D/A-buffer even before the synchronisation starts.

At a synchronous start of A/D-conversion and D/A-conversion the physical D/A-converter output signal will only be valid after two sample intervals. This is because

1. `ldcmod` is set to ‘1’, i.e. that the ‘conv_start’-signal sets the LDAC-signal and
2. after the A/D-transfer has been completed the following D/A-value is written into the ‘D/A-hold’-register.

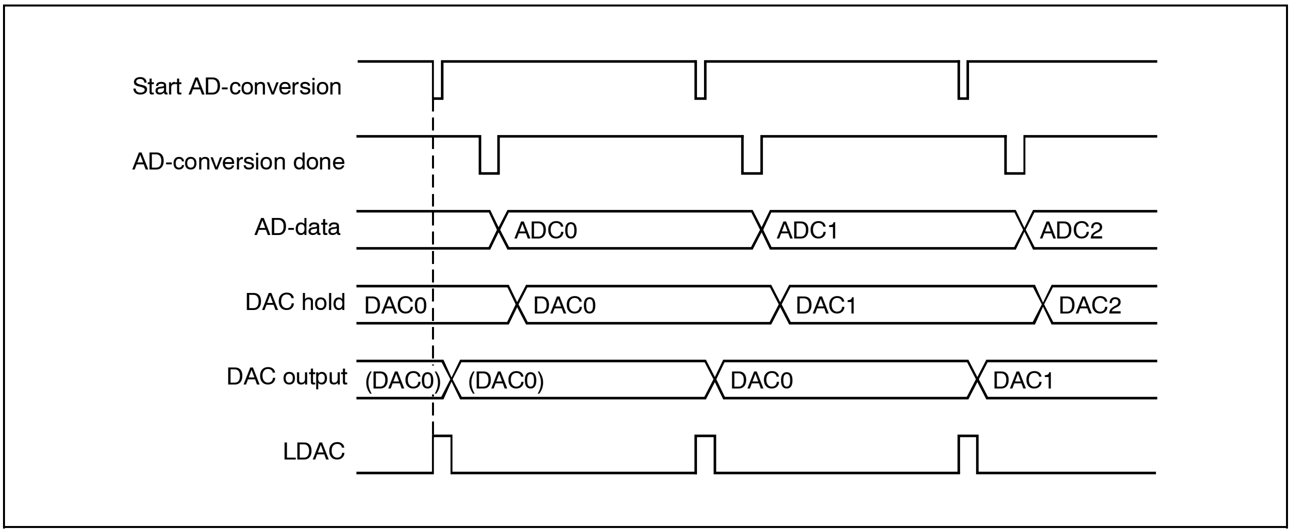

Fig. 6.4.1: Connection between A/D-converter timing and D/A-converter timing

6.5 Buffer Interrupts

When an ADC/DAC-buffer is completed, a VMEbus interrupt is ALWAYS triggered (independently from `vvtrg`)! At this moment the specification in the buffer-status structure 'Buffer_ Number_in_Work' is valid.

Example: ADC-transfer, 16 frames/buffers, 4 buffers, continuous transfers

Set `vadsrv` = 11_{DEZ}	-> Buffer_Number_in_Work = 1	
16 A/D-frames transferred	-> Buffer_Number_in_Work = 2	-> VMEbus interrupt
16 A/D-frames transferred	-> Buffer_Number_in_Work = 3	-> VMEbus interrupt
16 A/D-frames transferred	-> Buffer_Number_in_Work = 4	-> VMEbus interrupt
16 A/D-frames transferred	-> Buffer_Number_in_Work = 1	-> VMEbus interrupt

This page has intentionally been left blank.

7. Buffer Statistic

7.1 Design of the Statistic-Structure

In this structure statistical values can be stored for each A/D-converter channels and each auxiliary input. For each channels 8 bytes are available. They are assigned as follows:

Address offset (local):	0	2	4	6
contents:	`Average`	`Max_Wert`	`Min_Wert`	reserved

7.2 Addresses of the Structure Cells

Relative address range VMEbus [HEX]	Relative address range local [HEX]	Channel
0600 ... 060E	0300 ... 0307	Stat. Aux_In_8
0610 ... 061E	0308 ... 030F	Stat. Aux_In_7
0620 ... 062E	0310 ... 0317	Stat. Aux_In_6
0630 ... 063E	0318 ... 031F	Stat. Aux_In_5
0640 ... 064E	0320 ... 0327	Stat. Aux_In_4
0650 ... 065E	0328 ... 032F	Stat. Aux_In_3
0660 ... 066E	0330 ... 0337	Stat. Aux_In_2
0670 ... 067E	0338 ... 033F	Stat. Aux_In_1
0680 ... 068E	0340 ... 0347	Stat. ADC_1
0690 ... 069E	0348 ... 034F	Stat. ADC_2
06A0 ... 06AE	0350 ... 0357	Stat. ADC_3
06B0 ... 06BE	0358 ... 035F	Stat. ADC_4
06C0 ... 06CE	0360 ... 0367	Stat. ADC_5
06D0 ... 06DE	0368 ... 036F	Stat. ADC_6
06E0 ... 06EE	0370 ... 0377	Stat. ADC_7
06F0 ... 06FE	0378 ... 037F	Stat. ADC_8
0700 ... 070E	0380 ... 0387	Stat. ADC_9
0710 ... 071E	0388 ... 038F	Stat. ADC_10
0720 ... 072E	0390 ... 0397	Stat. ADC_11
0730 ... 073E	0398 ... 039F	Stat. ADC_12
0740 ... 074E	03A0 ... 03A7	Stat. ADC_13
0750 ... 075E	03A8 ... 03AF	Stat. ADC_14
0760 ... 076E	03B0 ... 03B7	Stat. ADC_15
0770 ... 077E	03B8 ... 03BF	Stat. ADC_16