Frequency Control Unit FCU4

AQS Technical Manual

Version 001

The information in this manual may be altered without notice.

This manual was written by

Kuntz, Eckert, Uhrig, Rommel

Rheinstetten, Germany

P/N: Z31651
DWG-Nr: 1357001

AQS FREQUENCY CONTROL UNIT, FCU4

Kuntz, Eckert, Uhrig

(2002/07/05)

Contents

Figures

Tables

1. Starting with FCU4

Handling Rules

- Handling under ESD safety conditions is necessary.
  Don’t touch uncovered metal of PCB and connectors before discharging yourself!
- Operating requires a backpanel version with 90–pin connector in middle position.
- Violently inserting of former devices with 30–pin connectors could damage this backpanel.
- Verifying the logic programming status is strictly recommended rather than reprogramming.
  Reprogramming of a JTAG programmable logic device is guaranteed up to about 100 times.

1. 1. Special features of FCU4 / FCU4_2

FCU4 is full software compatible to FCU3/256k. There are the two Versions of FCU4_4 and FCU4_2 respective to the assembled number of channels.

The FCU4_2 version provides two Frequeny Channels and is full compatible with the further FCU3 Controller board 256K.

The FCU4 contains in each channel:

- 256Kx32 Bit Program Memory (FCU instruction list)
- 256Kx32 Bit SGU Data RAM used to store the amplitude, phase, frequency, gating ec. values for the SGU.
- 8Kx32 Bit dual port fifo ram, for each channel, to store the commands transferred via FBus
- FBus interface avoids the timing bottlenecks on the AQ– and VME–Bus-Bus
- The minimal execution time of a FCU instruction is 50ns

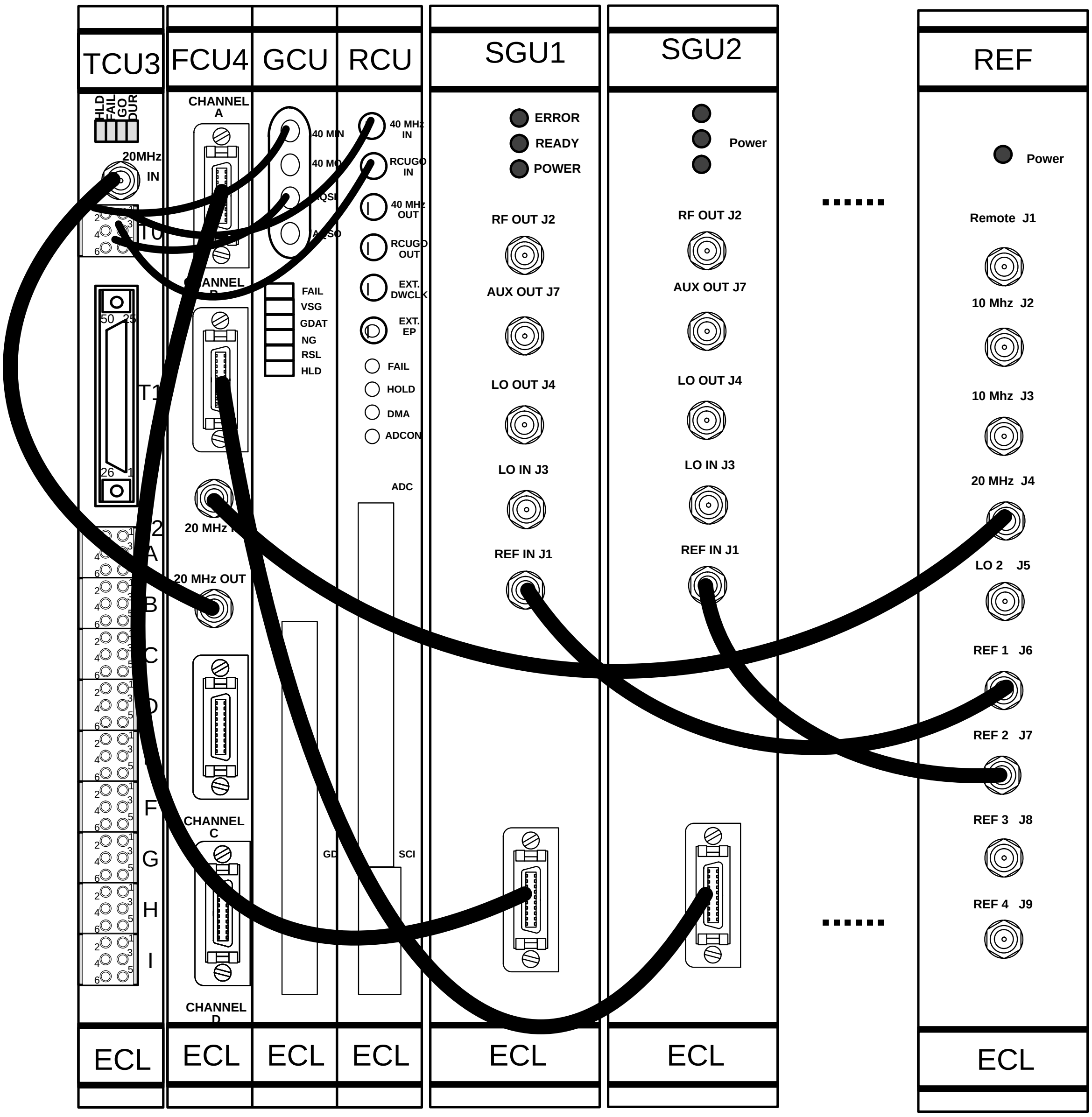
1. 2. Hardware Implementation

FCU4 can only be used in an AQS system. It needs to work with TCU3 and an advanced backpanel (90–pin connector in the middle position) providing an extended Acquisition Bus. It replaces two FCU3 boards.

Wiring

TCU3 accommodates the termination resistors of the 20 MHz reference clock line from the Reference Unit. So it has to be the last device and the Reference Unit the first one at this line.

Bild1: TCU3,FCU4 and SGU connection



1. 2. 1. Auto–Configuration of the Frequency Channels

Two FCU4_4 boards can be combined to provide 8 Frequency Channels called ”FCU Channel 1” to ”FCU Channel 8”.

Creating a special channel pair needs no jumper setting. FCU4 is provided with a self configuring scheme which works as follows:

Configuration Scheme

Slot n	Slot n+1		Slot n	Slot n+1		Slot n+1	Slot n+1
Fcu4_4	**Fcu4_4**		**Fcu4_4**	**Fcu4_2** **(FCU3)**		**Fcu4_2** **(FCU3)**	**Fcu4_4**
Cannel							
A = 1	A = 5		A = 1	A = 5		A = 1	A = 3
B = 2	B = 6		B = 2	B = 6		B = 2	B = 4
C = 3	C = 7		C = 3				C = 5
D = 4	D = 8		D = 4				D = 6

!: Gaps or other devices between FCU4 Boards lead to a second Channel 1–4 pair and to malfunctioning and should be avoided.

1. 3. Software Implementation

XWIN NMR

The `XWIN-NMR` version which is able to work with TCU3/FCU3/FCU4 is `XWIN-NMR 3.0`

FCU Test

Path `/usr/diskless/clients/spect/root/u/sys-test/fcu`

The `fcutest` recognizes the FCU version (FCU0 ,FCU3 or FCU4) on which it is requested to run and activates the correct program version.

See the AQX Test Manual for a detailed description of test programs.

2. Specifications

2. 1. FCU Versions

There are different FCU versions equipped with 64k or 256k byte memory.

Versions	Part No.	Layout No.	EC Level	Constrains	Constrains
				Hardw.	Softw.
FCU0 / 64k	H2556	H3P1940A	EC ≥ 01	AQX	
FCU0 / 256k	H2564	H3P1940A			
FCU0 / 64k	H2556	H3P1940A	EC ≥ 05	AQX	
FCU0 / 256k	H2564	H3P1940A			
TOMO FCU 64k	T5565	H3P1940A	EC ≥ 01	AQX	
FCU3 / 64k	H5822	H3P2260D	EC ≥ 00	AQS	XWIN–NMR Version 3.0
FCU3 / 256k	H9598	H3P2260D	EC ≥ 00	AQS	
FCU4_4 / 256k	H9727	H3P2570C	EC ≥ 00	AQS	
FCU4_2/ 256k	H9773	H3P2570C	EC ≥ 00	AQS	

Tabelle1: FCU versions

2. 2. Features

- Minimal duration 50 nsec
- Maximal duration 53 sec
- Minimal resolution 50 nsec
- Real time memory range 256 kWords of 64 bits
- Dual port FIFO RAM 8Kx32 Bit
- 32 sets of fast integrated pointer registers (quick pointer)
- 256 sets of memory based pointer registers (slow pointer)

2. 3. Concept and Functionality

Definition

Name	Abbr.	Operation
F–Bus		Bus from TCU3 to the FIFO's of all FCU3/FCU4
F–Bus Command	FCMD	Commands transmitted via F–Bus from TCU to FCU
FCMD FIFO	FIFO	Dual–Port RAM, managed as FIFO by the FIFO Controller
FIFO Controller		Controlled by the upper 8 bit of the FCMD, the FIFO Controller decides which FCMD out of the sequence will next be transferred to the FCON. Is able to manage Jumps, Loops and sequential outputs

Name	Abbr.	Operation
FCU Controller	FCON	Using sets of pointers carries out the FCU Program and generates the output stream of SGU Data being sent to the SGU. The usage of S– and Q–Pointer is part of the FCMD Trigger of an uninterrupted sequence is always a Real Time Command (AQCMD) sent by the TCU via the AQ–Bus
FCU Program		Lists of 64–bit program entries consisting of 32–bit SGU Data and 32–bit FCU Instructions which define the output sequence of the SGU Data
Slow Pointer	S–Pointer	256 sets of address registers located in the FCU RAM, managing these pointers requires additional RAM accesses and time.
Quick Pointer	Q–Pointer	32 sets of address registers located in the FCU Controller (”On–Chip”) updated during the output cycle of the SGU Data
FCU RAM		FCU Program Memory

Essential Differences to FCU0

1. There is no analog electronics in the digital area.

2. Each FCU4 provides 4 channels

3. Core devices are the FCU RAM and the RAM sequencer (FCU Controller) but with increased speed and advanced functionality

4. The timing of generating F–Bus Commands on TCU and the output sequence of SGU Data are made independent of each other by the FCMD FIFO

5. F–Bus between TCU3 and FCU4

6. The following digital frequency data are transferred to the SGU by a serial link:

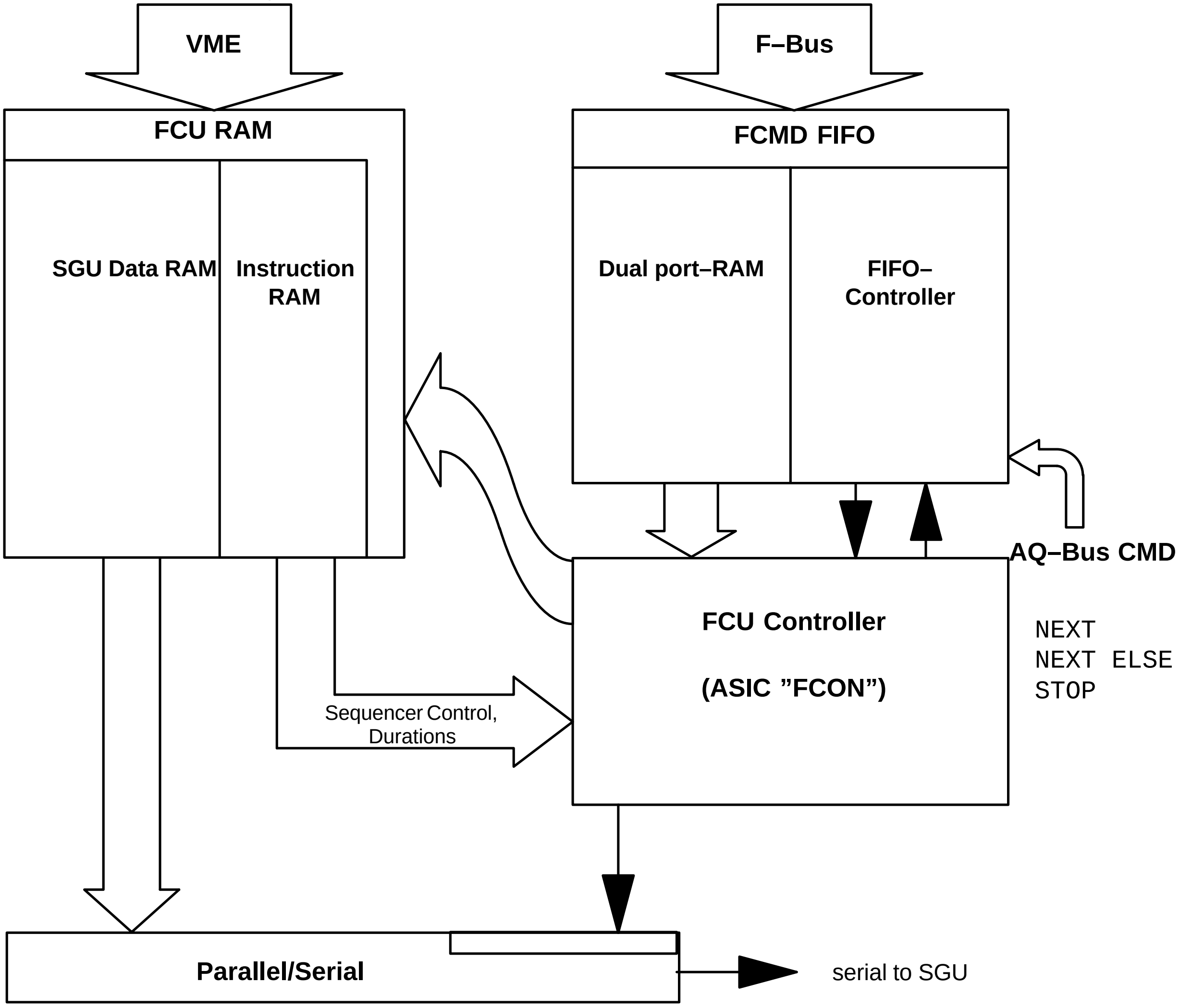
- Frequency, Phase, Amplitude, Gating, Shift, Update

Essential Differences to FCU3

1. There is no funtional differrence to FCU3

2. Each FCU4 provides 4 channels

Bild2: Functional diagram



2. 4. Construction

The FCU4 is a VME Bus module of 4 TE with an extended length. It consists of one printed circuit board containing 4 identical ”frequency channels” called ”A” , ”B”,”C”,”D”

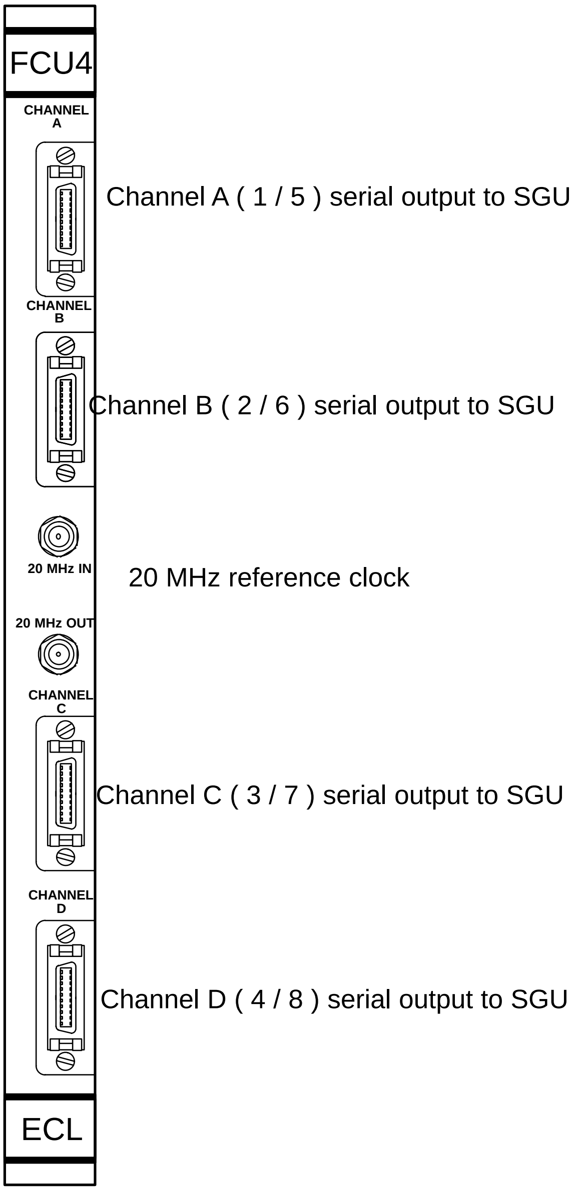
Two FCU4 modules can be combined to provide the 8 frequency channels F1 to F8. These modules have to be mounted without any gap to enable them to configure itself the channel numbers subsequently.
That means the leftmost FCU4 configures its channels A – D to be F1 – F4. Any FCU4 mounted in the next slot to the right configures its channel A –D to be F4 F8. Any other module than FCU3 or FCU4 or a gap between 2 FCU’s result to a second FCU configured to have F1 – F4.

Board Size

The real size is 233.35 mm by 280 mm . This is the so called ”Double European Standard” format with a nominal plug in depth of 280 mm.

Bild3: Front View

FCU4 Front view


FCU4
CHANNEL A
Channel A (1 / 5) serial output to SGU
CHANNEL B
Channel B (2 / 6) serial output to SGU
20 MHz IN
20 MHz reference clock
20 MHz OUT
CHANNEL C
Channel C (3 / 7) serial output to SGU
CHANNEL D
Channel D (4 / 8) serial output to SGU
ECL

2. 5. Part numbers of FCU4

Tabelle2: Table of Assembly Groups

Amount	Title	Function	Part–Nr.
1	FCU4_4	Assembled PCB with 256k words of FCU RAM	H9727
1	FCU4_2	Assembled PCB with 256k words of FCU RAM	H9773

Amount	Title	Function	Part–Nr.
1	FCU4	Layout	H3P2570
1		Plain PCB	H9728
1		Frontpanel	HZ08121

2. 6. Accessories

Tabelle3: Part# of Accessories

Part	Part Nr.
DCX Cable tree	H6694
AQS Cable Set	HCABLE
Cable coax 600mm SMA/SMA ???	HZ03805
Cable coax 250mm SMA/SMA	Hz03804/A
Cable coax 400mm SMA/SMA	Hz10105/A
SGU Interface Cable	Z13928

Connecting the 20 MHz reference clock

TCU3 includes the termination resistors of the 20 MHz reference clock line from the Reference Unit. So it has to be the last device and the Reference Unit the first one at this line.

2. 7. Operational Settings

FCU4 Clock Selection

For testing, the *20Mhz Referenz Clock* has not to be connected.

If the board detects that no clock input is connected, it swiches to a on board 20Mhz help clock to guarante all boardfunctions. This clock is not syncroniced with the TCU3 and SGU ! Thererefor, the tests which use tthe FBUS, wil not work properly.

2. 7. 1. Firmware version

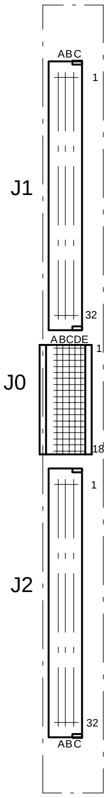
FCU4 doesn’t need any Firmware

2. 8. Connectors and Signal Allocations

2. 8. 1. Signal allocation at VME bus connectors J0 and J2

Bild4: VME bus connectors,

Plug–in direction view toward the back panel



Connector J2

The signal arrangement of the table corresponds to the contact locations shown in the figure.

	Reihe A	Reihe C
1	AQA_0	AQI_0
2	AQA_1	AQI_1
3	AQA_2	AQI_2
4	AQA_3	AQI_3
5	AQS_0	AQI_4
6	AQS_1	AQI_5
7	AQS_2	AQI_6
8	AQS_3	AQI_7
9	GND	GND
10	AQD_0	AQY_0
11	AQD_1	AQY_1
12	AQD_2	AQY_2
13	AQD_3	AQY_3
14	AQD_4	AQY_4
15	AQD_5	AQY_5
16	AQD_6	AQY_6
17	AQD_7	AQY_7
18	GND	GND
19	AQY_WR	AQY_AS
20	AQY_ACK	AQY_DS
21	GND	GND
22	AQD_8	res_1
23	AQD_9	res_2
24	AQD_10	res_3
25	AQD_11	res_4
26	AQD_12	res_5
27	AQD_13	res_6
28	AQD_14	res_7
29	AQD_15	res_8
30	GND	res_9
31	GND	GND
32	AQSTROBE	AQEXEC_J2

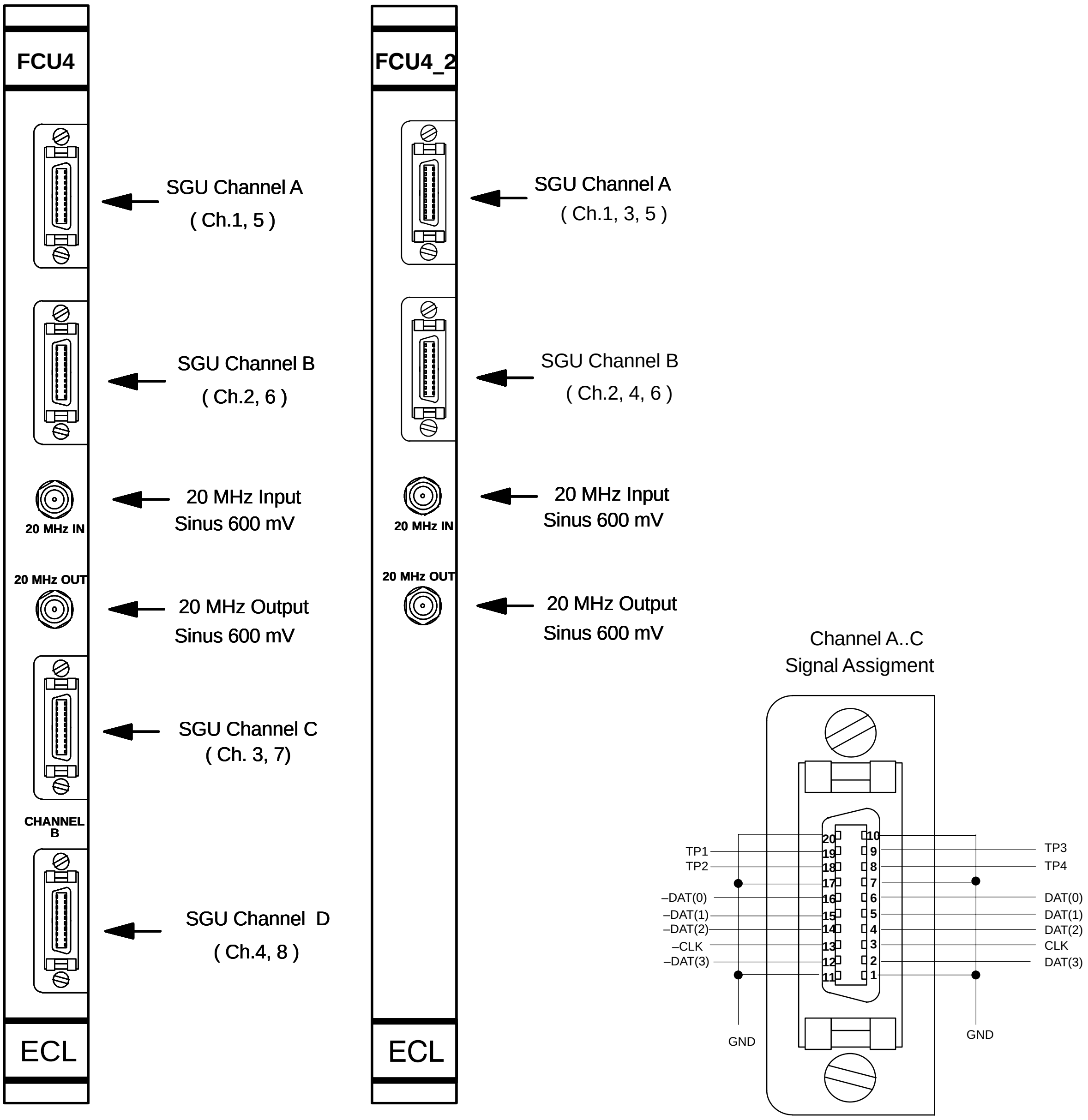
Connector J0

The signal arrangement of the table corresponds with the contact locations shown in the figure.

	Reihe A	**Reihe B**	**Reihe C**	**Reihe D**	**Reihe E**
1	Jext	TCLK	TMS	TDI	TDO
2	res SC16	res SC8	ACLK	SCLK	SDA
3	res_Ltg_4	res_Ltg_3	res_Ltg_2	res_Ltg_1	res_Ltg_0
4	GND	GND	GND	GND	GND
5	3,3V	3,3V	3,3V	3,3V	3,3V
6	res_CLK_4	res_CLK_3	res_CLK_2	res_CLK_1	res_CLK_0
7	GND	GND	GND	GND	GND
8	FD_0	FD_1	FD_2	FD_3	FD_4
9	FD_5	FD_6	FD_7	FD_8	FD_9
10	FD_10	FD_11	FD_12	FD_13	FD_14
11	FD_15	FD_16	FD_17	FD_18	FD_19
12	FD_20	FD_21	FD_22	FD_23	FD_24
13	FD_25	FD_26	FD_27	FD_28	FD_29
14	DC6_out	DC4_out	DC2_out	DC0_out	FD_30
15	DC6_in	DC4_in	DC2_in	DC0_in	FD_31
16	DC7_out	DC5_out	DC3_out	DC1_out	F_REQ
17	DC7_in	DC5_in	DC3_in	DC1_in	GND
18	SC1	SC2	SC4	AQEXEC_J0	AQSEL

2. 8. 2. Pin assignment of the SGU Interface Connector

Bild5: SGU Interface Connector



2. 9. Power Requirements

The FCU4 requires power supply of the following voltages:

	Part–No.	+5 V (3,3V intern)	+12 V	–12 V	+3,3V J0: A5, B5, C5, D5, E5
FCU4_4	H9727	2,5A	—	—	
FCU4_2	H9773	2,0A	—	—	

3. Service Information

Handling Rules

- Handling under ESD safety conditions is necessary.
  Don't touch uncovered metal of PCB and connectors before discharging yourself!
- Operating requires a backpanel version with 90–pin connector in middle position.
- Violently inserting of former devices with 30–pin connectors could damage this backpanel.
- Verifying the logic programming status is strictly recommended rather than reprogramming.
  Reprogramming of a JTAG programmable logic device is guaranteed up to about 100 times.

3. 1. New Procedures

Additionally to the Programmable Logic set ”Prog Pal” like on former FCU versions this FCU4 contains in system programmable logic devices which need a ”`Prog File`” to be programmed. These devices are on board arranged in chains and are programmed by JTAG protocol.
”Prog File” is a directory tree containing JEDEC source files, chain description files and JTAG–testfiles. The nomenclature of the Prog file is *“Px.y”* where x is the index of the Programmingversion and y is the index of the testversion

Programming can be carried out by a PC. It is planned to accomplish this in future via the CCU.

Please teach yourself in referring to the AQX Test Manual.

3. 2. Prototype Situation

FCU4 of the Introduction Status show no functional problems up to January 02.

Prior to Introduction delivered FCU4 of prototype status had been updated to the introduced one of EC00.

3. 3. Introduction Status

3. 3. 1. Prog File

The Progfile H9727A1/P1.0 has to be used for Fcu4/4.

The Progfile H9773A1/P1.0 has to be used for Fcu4/2.

3. 3. 2. Modifications of the introduced layout

There is a small wire between the two neighboring pads of the C103 and the unplaced R61.

3. 3. 3. Jumper Setting

There are no Jumpers to set.

3. 3. 3. 1. Firmware

FCU4 doesn’t need any Firmware

3. 4. History of Modifications

3. 4. 1. FCU4_4

EC No.	Date	Part Number	Description of Bugs, Changes and Modifications	Ser.No.	New EC–Level
2943	4.3.2002	9727	Introduction of FCU4, 256k words Using Prog file FCU4/4_P1.x EC00 Layout H3P2260C	0033	00

3. 4. 2. FCU4_2

EC No.	Date	Part Number	Description of Bugs, Changes and Modifications	Ser.No.	New EC–Level
2942	4.3.2002	9773	Introduction of FCU4/2, 256k words Using Prog file FCU4/2_P1.x EC00 Layout H3P2260C	0015	00

4. Condensed Description

4. 1. Construction and Configuration

One FCU4 board contains 4 separate Frequency Channels called ”Channel A”, ”Channel B”, ”Channel C” and ”Channel D”.

Bild6: Block diagram of FCU4

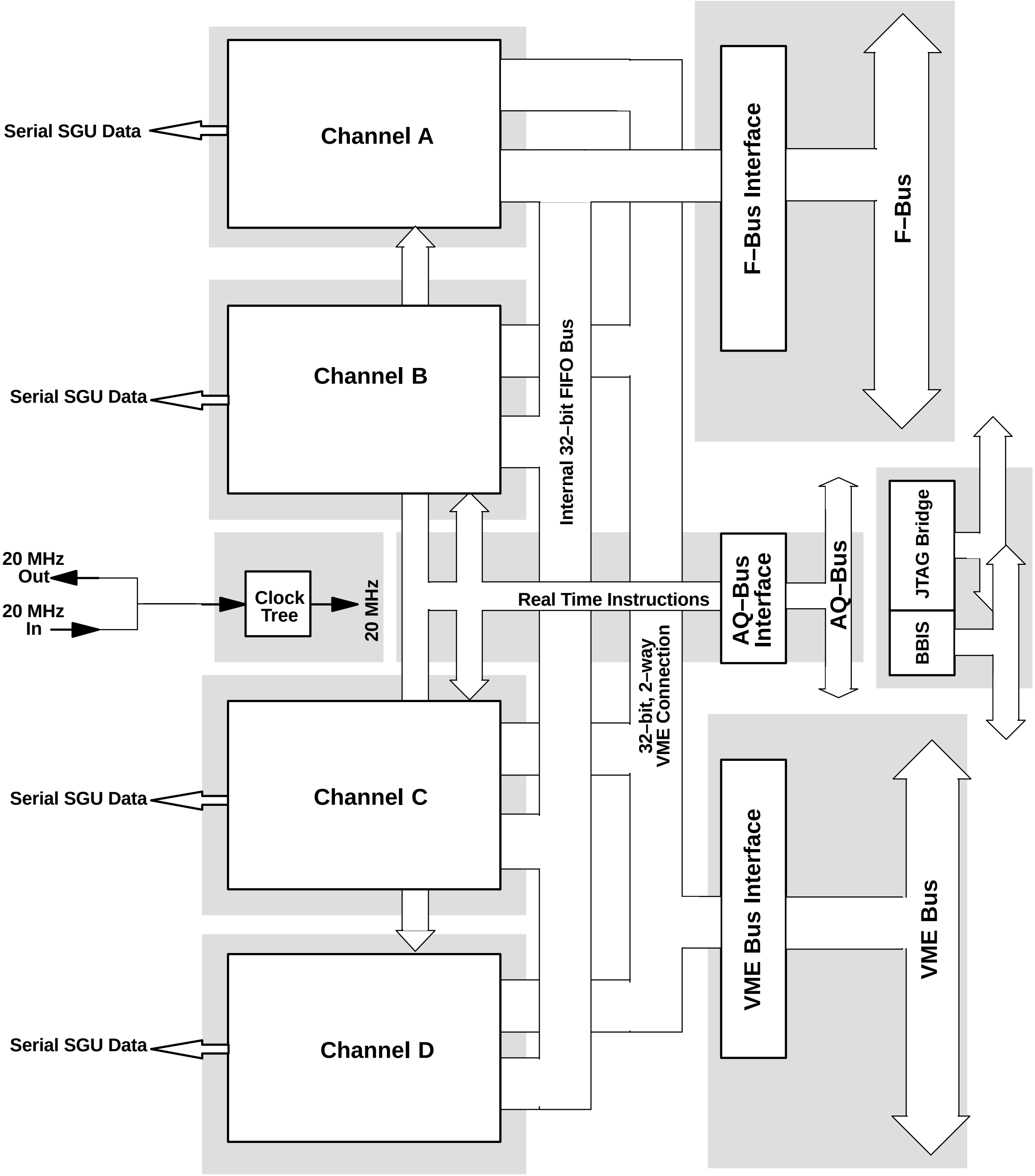
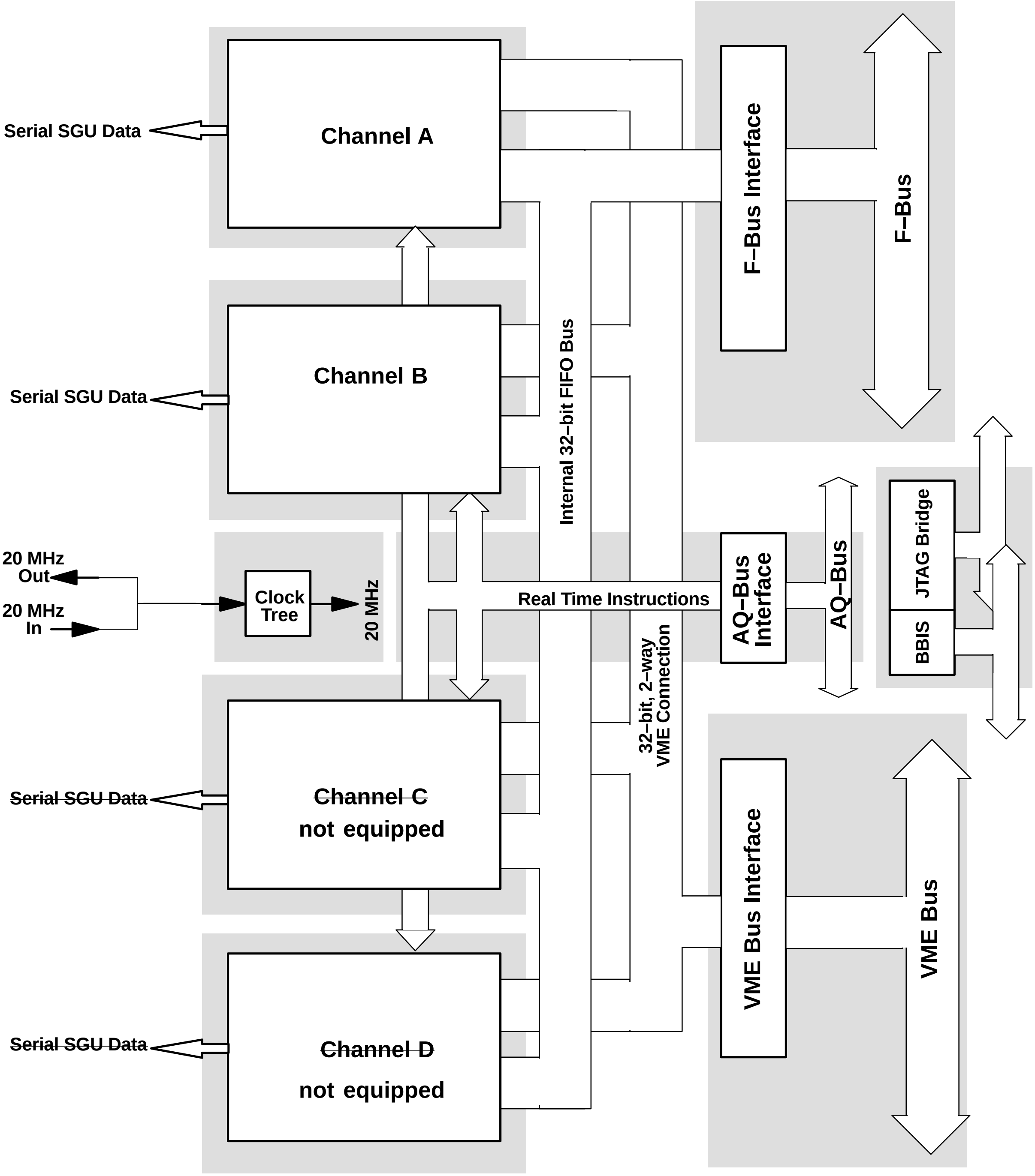


Bild7: Block diagram of FCU4/2



4. 1. 1. Auto–Configuration of the Frequency Channels

Four FCU4 boards can be combined to provide 8 Frequency Channels called ”FCU Channel 1” to ”FCU Channel 8”.

Creating a special channel group needs no jumper setting. FCU4 is provided with a self configuring scheme which works as follows:

Configuration Scheme

Slot n	**Slot n+1**		**Slot n**	**Slot n+1**		**Slot n+1**	**Slot n+1**
Fcu4_4	**Fcu4_4**		**Fcu4_4**	**Fcu4_2** **(FCU3)**		**Fcu4_2** **(FCU3)**	**Fcu4_4**
Cannel							
A = 1	A = 5		A = 1	A = 5		A = 1	A = 3
B = 2	B = 6		B = 2	B = 6		B = 2	B = 4
C = 3	C = 7		C = 3				C = 5
D = 4	D = 8		D = 4				D = 6

! Gaps or other devices between FCU's lead to a second Channel group 1–4 and to malfunctioning and should be avoided.

4. 2. Channel Architecture

As brought out in Bild8: the FCU4 can be considered to comprise these functional units:

- FCU RAM
- FCU Controller
- FIFO
- VME–Bus Interface
- F–Bus Interface
- AQ–Bus Interface
- SGU Interface

Bild8: Block diagram of one FCU4 channel

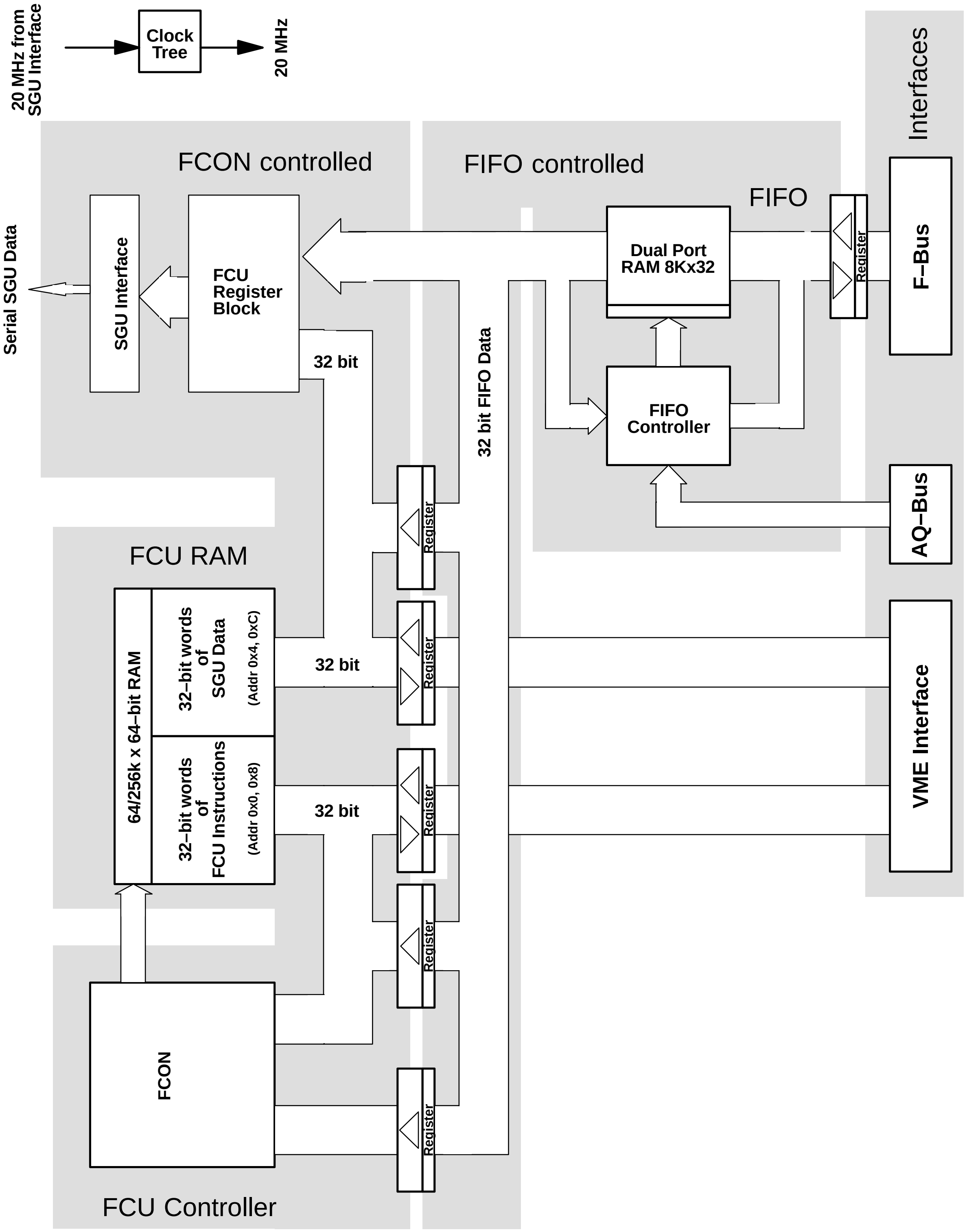
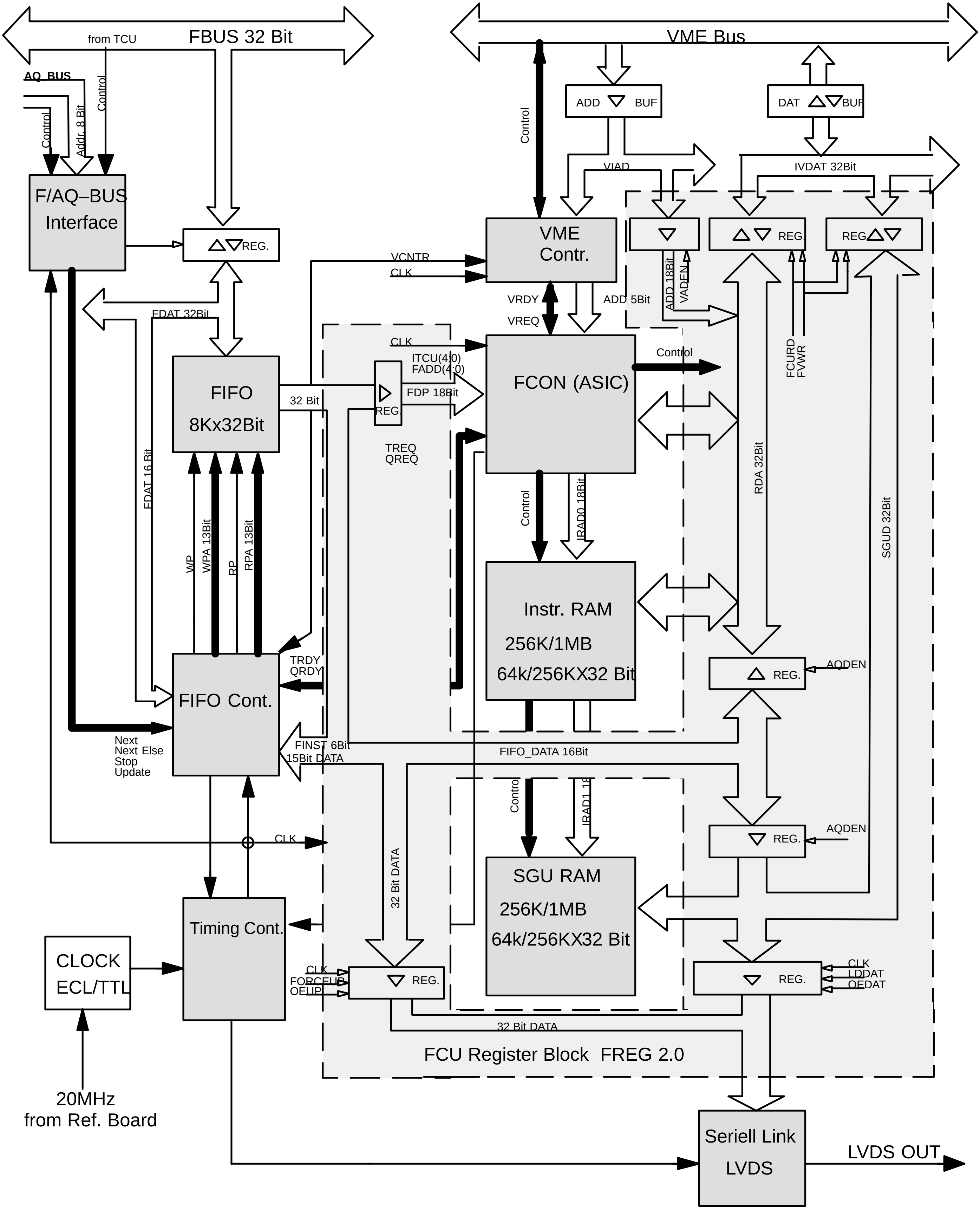


Bild9: Functional Diagram of one channel



4. 3. Summary of Logical References

VME Bus Address Codes

Tabelle4: VME Bus address ranges of FCU Channels

Channel	Address Range
FCU Channel 1	16000000 —– 161FFFFF
FCU Channel 2	16400000 —– 165FFFFF
FCU Channel 3	16800000 —– 169FFFFF
FCU Channel 4	16C00000 —– 16DFFFFF
FCU Channel 5	17000000 —– 171FFFFF
FCU Channel 6	17400000 —– 175FFFFF
FCU Channel 7	17800000 —– 179FFFFF
FCU Channel 8	17C00000 —– 17DFFFFF

Adding a value of 0x800000 to each address provides the address codes of the following 3 channel groups.

Tabelle5: VME Bus Address Map and Device Codes

Shown are the address codes of channel A, B, C and D configured as FCU Channel 1 – 4

VME Bus Address	Device	Reg. Name	Mode R/W	Oprd. Byte 3 2 1 0
16000000 to 161FEFFF	2MB Real Time Program Memory Channel 1 or A		R/W	b b b b
161FF000 to 161FF1FF	Quick Pointer Array (4 x 32 Pointer) Ch.1	qpoint	R/W	b b b b
161FF204	SGU Interface Transfer Reg., Ch.1	pitra	W	b b b b
161FF220	Read RAM Addr & Instr Status, Ch.1	rdra	R	b b b b
161FF300	Configuration Reg.0 Ch.1 – Ch.4	config0	R	x x x b
161FF304	Configuration Reg.1 Ch.1 – Ch.4	config1	R	x x x b
161FF308	Configuration Reg.2 Ch.1 – Ch.4	config2	R	x x x b
161FF30C	Configuration Reg.3 Ch.1 – Ch.4	config3	R	x x x b
161FF320	Software Reset, Ch.1	vres	W	x x x x
161FF380	Stop FCU, Ch.1	vstop	W	x x x x
161FF3C0	Read VME Address Reg., Ch.1	vadd	R	b b b b

VME Bus Address	**Device**	**Reg. Name**	**Mode R/W**	**Oprd. Byte 3 2 1 0**
16400000 to 165FEFFF	2MB Real Time Program Memory Channel 2 or B		R/W	b b b b
165FF000 to 165FF1FF	Quick Pointer Array (4 x 32 Pointer), Ch.2	qpoint	R/W	b b b b
165FF200	SGU Interface Control Reg., Ch.2	pictr	W	x x b b
165FF204	SGU Interface Transfer Reg., Ch.2	pitra	W	b b b b
165FF220	Read RAM Addr & Instr Status, Ch.2	rdra	R	b b b b
165FF300	Configuration Reg.0 Ch.1 – Ch.4	config0	R	x x x b
165FF304	Configuration Reg.1 Ch.1 – Ch.4	config1	R	x x x b
165FF308	Configuration Reg.2 Ch.1 – Ch.4	config2	R	x x x b
165FF30C	Configuration Reg.3 Ch.1 – Ch.4	config3	R	x x x b
165FF320	Software Reset, Ch.2	vres	W	x x x x
165FF380	Stop FCU, Ch.2	vstop	W	x x x x
165FF3C0	Read VME Address Reg., Ch.2	vadd	R	b b b b

16800000 to 169FEFFF	2MB Real Time Program Memory Channel 3 or C		R/W	b b b b
169FF000 to 169FF1FF	Quick Pointer Array (4 x 32 Pointer), Ch.3	qpoint	R/W	b b b b
169FF200	SGU Interface Control Reg., Ch.3	pictr	W	x x b b
169FF204	SGU Interface Transfer Reg., Ch.3	pitra	W	b b b b
169FF220	Read RAM Addr & Instr Status, Ch.3	rdra	R	b b b b
169FF300	Configuration Reg.0 Ch.1 – Ch.4	config0	R	x x x b
169FF304	Configuration Reg.1 Ch.1 – Ch.4	config1	R	x x x b
169FF308	Configuration Reg.2 Ch.1 – Ch.4	config2	R	x x x b
169FF30C	Configuration Reg.3 Ch.1 – Ch.4	config3	R	x x x b
169FF320	Software Reset, Ch.3	vres	W	x x x x
169FF380	Stop FCU, Ch.3	vstop	W	x x x x
169FF3C0	Read VME Address Reg., Ch.3	vadd	R	b b b b

16C00000 to 16DFEFFF	2MB Real Time Program Memory Channel 4 or D		R/W	b b b b
16DFF000 to 16DFF1FF	Quick Pointer Array (4 x 32 Pointer), Ch.4	qpoint	R/W	b b b b

VME Bus Address	Device	Reg. Name	Mode R/W	Oprd. Byte 3 2 1 0
16DFF204	SGU Interface Transfer Reg., Ch.4	pitra	W	b b b b
16DFF220	Read RAM Addr & Instr Status, Ch.4	rdra	R	b b b b
16DFF300	Configuration Reg.0 Ch.1 – Ch.4	config0	R	x x x b
16DFF304	Configuration Reg.1 Ch.1 – Ch.4	config1	R	x x x b
16DFF308	Configuration Reg.2 Ch.1 – Ch.4	config2	R	x x x b
16DFF30C	Configuration Reg.3 Ch.1 – Ch.4	config3	R	x x x b
16DFF320	Software Reset, Ch.4	vres	W	x x x x
16DFF380	Stop FCU, Ch.4	vstop	W	x x x x
16DFF3C0	Read VME Address Reg., Ch.4	vadd	R	b b b b

F–Bus Address Codes

The 8 address bits AQY7,..,0 are divided into a device address part AQY7,..,4 coding one of the acquisition devices and a subaddress part AQY3,..,0 coding special on board function codes.

Tabelle6: F–Bus Device and Function Codes

Device	AQY7,...,0
FCU1	00,..,0F
FCU2	10,..,1F
FCU3	20,..,2F
FCU4	30,..,3F
FCU5	40,..,4F
FCU6	50,..,5F
FCU7	60,..,6F
FCU8	70,..,7F
RCU	80,..,8F
GCU	90,..,9F
unused	Ax,..,Fx

The source of AQY7,...,0 are the VME or Local address bit A9,...,A2.

Tabelle7: FBUS Device Codes

Shown are the VME address codes of channel A configured as FCU Channel 1. The base addresses of the following channels can be get by adding 0x40 each.

VME Bus access to the F–Bus is possible via the TCU3 only.

Address of VME–Bus access	Address Codes on F–Bus AQY3,...,AQY0	Action	Destination Name	Mode R/W	Size of Oprd. Byte 3 2 1 0
19222000	0	Read number of empty FIFO lines	rddiff	R	x x b b
19222004	1	FIFO Write pointer	wpoint	R/W	x x b b
19222008	2	FIFO Read pointer	rpoint	R/W	x x b b
1922200C	3	Write Control word	wrcon	W	x x b b
19222010	4	Read and Clear Status register	rdstat	R/W	x x b b
19222014	5	Version register	rdvers	R	x x b b
19222020	8	FIFO Memory	fram	R/W	b b b b
19222024	9	Write RAM and store If–Address	wrif	W	b b b b
19222028	A	Go to end of ELSE and Write RAM	wrelse	W	b b b b
19222030	C	Execute AQIF (Test function)	fif	W	x x x x
19222034	D	Execute AQNEXT (Test function)	fnext	W	x x x x

AQ–Bus Codes

There are 8 address bits divided into a device address part AQA3,..,0 coding one of the acquisition devices and a function part AQS3,..,0 coding special on board function codes.

The source of AQA3,...,0 and AQS3,...,0 are the bit w2_23,...,w2_16 of the Real–Time Program entry on TCU.

The AQ–Bus data bit AQD15,..,0 have no meaning on FCU4.

Tabelle8: AQ–Bus Device Codes

Device	AQA3,..,0
FCU1	0
FCU2	1
FCU3	2
FCU4	3
FCU5	4
FCU6	5
FCU7	6
FCU8	7
RCU	8
GCU	9
unused	A,..,F

Tabelle9: AQ–Bus Function Codes

Function Codes AQS3,...,AQS0	Action of FIFO Controller	Destination Name	Mode R/W	Size of Oprd. Byte 3 2 1 0
0x1	AQ–STOP, stops the FCU Controller	FIFO Contr.	W	x x x x
0x2	AQ–NEXTIF, takes the 1. command of an IF sequence to FCON	FIFO Contr.	W	x x x x
0x4	AQ–NEXT, takes the next command to FCON	FIFO Contr.	W	x x x x

4. 4. Bus Structure

The bus structure of one FCU channel is functionally split up into 3 partitions

- Interfaces
- Buses controlled by the FIFO Controller
- Buses controlled by the FCU Controller

Data and instructions are exchanged between these sections via registers which are read or loaded by the master of the one section and on special request read or loaded by the master of the other section.
An interface can request the FIFO and the FCON master and the FIFO can request the FCON master but not vice versa.
This ensures an uninterrupted operation of the FCU Controller and an operation of the FIFO uninterrupted by an interface if necessary.

4. 4. 1. Bus master functions

F–Bus Interface

Requester	Source of access	Destination of access	Requesting to
F–Bus	F–Bus	Exchange Register to FIFO	FIFO Controller

VME–Bus Interface

Requester	Source of access	Destination of access	Requesting to
VME–Bus	VME–Bus	Exchange Register to SGU Data RAM	FCON
		Exchange Register to FCU Instr. RAM and FCON registers	

AQ–Bus Interface

Requester	Source of access	Destination of access	Requesting to
AQ–Bus	AQ–Bus	FIFO Controller	FIFO Controller
		FCU Controller	

FIFO Controller

Requester	Source of access	Destination of access	Requesting to
F–Bus Interface	Exchange Register to FIFO	Dual Port RAM	none
AQ–Bus	Dual Port RAM	Register to SGU Interface	FCON
		Register to SGU Data RAM	FCON
		Register to SGU Instruction RAM	FCON
		Instruction register to FCON	FCON

FCU Controller (FCON)

Requester	Source of access	Destination of access	Requesting to
VME–Bus Interface	Exchange Register to SGU Data RAM	SGU Data RAM	none
	Exchange Register to SGU Instr. RAM	SGU Instruction RAM	none
FIFO Controller	Register to SGU Interface	SGU Interface	none
	Register to SGU Data RAM	SGU Data RAM	none
	Register to SGU Instruction RAM	SGU Instruction RAM	none
	Instruction register to FCON	FCON	none
Instructions	SGU Instruction RAM	FCON	none
	SGU Data RAM	SGU Interface	none

4. 5. Interfaces

4. 5. 1. VME Bus Interface

The Interface to the VME Bus is a single word (32Bit) slave interface. It has no Nipple or Burst Mode capability. The access to the resources on the FCU is controlled by the FCU Controller FCON (ASIC). There are four equivalent channels which share the common VME Bus interface. It is possible to read and write from the VME Bus the following destinations on the FCU:

- FCON(ASIC) Q–Pointer and Register
- Configuration Register
- Instr. RAM
- Data RAM
- SGU Interface

If the FCU executes an Instruction List with short durations and the CCU wants to access the SGU RAM via VMEBus the VME Bus might not be serviced. In this case a busy flag

is set in the FCU status register ’RDSTAT’(Bit 8). The CCU should read this bit after every VME access to the FCU to check the successfully operation. If this bit is set the CCU should it clear and repeat the operation once more.
For a special address line test it is possible to read the address of an prior VME write cycle.

4. 5. 1. 1. Reset Control

Soft Reset

VME Bus Address	Device	Reg. Name	Mode R/W	Oprd. Byte 3 2 1 0
161FF320	Software Reset, Ch.1	vres	W	x x x x

The FCU4 has four channels and each channel has a separate device code for reset. The VME Bus Sysreset line initializes all channels of the FCU4. The control logic on the board i.e. the FCON (ASIC), the FIFO Controller, F–Bus and the VME–Bus Controller are initialized.

The FCON (ASIC) will stop the execution of a program list without saving the actual pointer and goes into the ’START’ State. The buses (RDA,SGUD), controlled by the FCON, will be in tree state. The content of the pointer array isn’t changed.

The FIFO–Controller is also initialized and it stops the execution of a FIFO Instruction list. The following internal register will be cleared (set to 0).

- FCU Status register (rdstat) a pending interrupt will be cleared
- FCU control register (wrcon)
- Loop counter
- Read Pointer
- Write Pointer
- Write Pointer barrier Register

4. 5. 1. 2. VME Dev. Code ’vstop’

VME Bus Address	Device	Reg. Name	Mode R/W	Oprd. Byte 3 2 1 0
161FF380	Stop FCU, Ch.1	vstop	W	x x x x

A running program (FCU instr. list) being executed by the FCU Controller FCON is stopped. The FCU Controller will go in State ’START’ and the current ’slow pointer’ is saved. The FCON will continue execution a FCU instruction when triggered by the FIFO Controller on a new AQNEXT command from the AQBus.

4. 5. 1. 3. VME Dev. Code ’rdra’

This is a option of FCU3. This option does work only in singlestep mode. Because FCU4 does not support singlestep, this is no valid command for FCU4.

4. 5. 1. 4. VME address register

This register is storing the FCU RAM address of the last access from VME–Bus to any part of the FCU–RAM. It can be read for test purposes via the VME–Bus.

Device Codes

VME Bus Address	Device	Reg. Name	Mode R/W	Oprd. Byte 3 2 1 0
161FF3C0	Read VME Address Reg., Ch.1	vadd	R	b b b b
165FF3C0	Read VME Address Reg., Ch.2	vadd	R	b b b b
169FF3C0	Read VME Address Reg., Ch.3	vadd	R	b b b b
16DFF3C0	Read VME Address Reg., Ch.4	vadd	R	b b b b

4. 5. 1. 5. Configuration register

The Configuration register comprises 4 8–bit registers called ”config0,..,3”.

Device Codes

VME Bus Address	Device	Reg. Name	Mode R/W	Oprd. Byte 3 2 1 0
161FF300	Configuration Reg.0 Ch.1 to Ch.4	config0	R	x x x b
161FF304	Configuration Reg.1 Ch.1 to Ch.4	config1	R	x x x b
161FF308	Configuration Reg.2 Ch.1 to Ch.4	config2	R	x x x b
161FF30C	Configuration Reg.3 Ch.1 to Ch.4	config3	R	x x x b

Bit Allocation

config0 reserved

config1 D4....D0 contain the binary code of the slot number.
D6....D5 contain the Daisychain bits for the channel number
00 ==> Channel 1..4 are adressed on this FCU4
01 ==> Channel 2..6 are adressed on this FCU4
10 ==> Channel 5..8 are adressed on this FCU4
11 ==> Channel 7..8 are adressed on this FCU4

config2 D3....D0 contain the Layout version of the FCU

config3 D7....D0 contain the FCU version (0x4 => FCU4)

4. 5. 2. F–BUS Interface

The TCU3 can send a sustained stream of commands to all FCU channels via the F–Bus unaffected by any other data traffic

Features

- 32 data bits
- 8 address bits, (same as Y–Bus)
- TCU3 is the only master
- Faster protocol as Y–Bus
- Accessible from i960 on TCU3 and from VME–Bus via TCU4

F–Bus Address ranges

F–Bus access can be accomplished by the i960 of the TCU3 or by the VME–Bus using addresses out of the TCU3 address range via TCU3.

The 8 F–Bus address bits AQY7,..,0 are divided into a device address part AQY7,..,4 coding one of the acquisition devices and a subaddress part AQY3,..,0 coding special on board function codes.

The source of AQY7,...,0 are the VME bus or the local address bits A9,...,A2 on the TCU.

Tabelle10: F–Bus address ranges of Acquisition Devices

Devices	Address Ranges	
	F–Bus address ranges AQY7,...,0	Equivalent VME–Bus ranges
FCU Channel1	00,..,0F	19222000 – 1922203F
FCU Channel2	10,..,1F	19222040 – 1922207F
FCU Channel3	20,..,2F	19222080 – 192220BF
FCU Channel4	30,..,3F	192220C0 – 192220FF
FCU Channel5	40,..,4F	19222100 – 1922213F
FCU Channel6	50,..,5F	19222140 – 1922217F
FCU Channel7	60,..,6F	19222180 – 192221BF
FCU Channel8	70,..,7F	192221C0 – 192221FF
RCU	80,..,8F	19222200 – 1922223F
GCU	90,..,9F	19222240 – 1922227F
unused	Ax,..,Fx	19222280 – 192223FF

Tabelle11: Codes of F–Bus connected devices on FCU4

Shown are the VME address codes of channel A configured as FCU Channel 1. The base addresses of the following channels can be get by adding 0x40 each.

VME Bus access to the F–Bus is possible via the TCU3 only.

Address of VME–Bus access	Address Codes on F–Bus AQY3,...,AQY0	Action	Destination Name	Mode R/W	Size of Oprd. Byte 3 2 1 0
19222000	0	Read number of empty FIFO lines	rddiff	R	x x b b
19222004	1	FIFO Write pointer	wpoint	R/W	x x b b

Address of VME–Bus access	Address Codes on F–Bus AQY3,...,AQY0	Action	Destina-tion Name	Mode R/W	Size of Oprd. Byte 3 2 1 0
19222008	2	FIFO Read pointer	rpoint	R/W	x x b b
1922200C	3	Write Control Word Register	wrcon	W	x x b b
19222010	4	Read and Clear Status register	rdstat	R/W	x x b b
19222014	5	FIFO Controller version register	rdvers	R	x x b b
19222020	8	FIFO Memory	fram	R/W	b b b b
19222024	9	Write FIFO Memory and store If–Address	wrif	W	b b b b
19222028	A	Go to end of ELSE and Write FIFO Memory	wrelse	W	b b b b
19222030	C	Execute AQIF (Test function)	fif	W	x x x x
19222034	D	Execute AQNEXT (Test function)	fnext	W	x x x x

4. 5. 2. 1. Status Register and Control Word Register

The Status Register contains the 16 high active status bits bit–0,...,bit–15. Its information can only be read. Writing a ”1” to any of these bits will selectively clear it.
The high active state of any bit out of bit–0 to bit–12 can generate an interrupt to the TCU if this had been enabled by writing a ”1” to corresponding bit of the Control Word Register.
All bits of both registers will be cleared by Soft Reset or Sys Reset.

Meaning

Bit#	Set to ”1” by:
3,..,0	FIFO Instruction INTRPT3,..,0
4	Not defined FIFO Instruction
5	Asic FCON doesn’t answere
6	AQNEXT too early, prior FIFO Instruction not finished
7	No valid If–Address when AQ–NEXTIF on AQ–Bus occurs
8	VME Bus time out because FCON busy
9	Duration error in FCON
10	Read pointer of FIFO reaches the write pointer or vice versa
11	Stack overflow of FCON
12	If–Address was reached by the read pointer but no AQ–NEXTIF occured so far
13	not used
14	For test and debugging only
15	Or’d Interrupts enabled in the Control Word Register

4. 5. 3. AQ–BUS Interface

The AQ–Bus is a clocked real time bus with a 20 MHz time base. It transfers acquisition commandos which have exactly be timed from the TCU to all acquisition devices.

Features

- Clocked bus, 50 nsec cycle time
- 16 data bits, not used on FCU4
- 8 address bits
- TCU is the only master

4. 5. 3. 1. AQ–Bus Address ranges

The 8 address bits are divided into a device address part AQA3,..,0 coding one of the acquisition devices and a function part AQS3,..,0 coding special on board function codes.

The source of AQA3,...,0 and AQS3,...,0 are the bit w2_23,...,w2_16 of the Real–Time Program entry on TCU3.

The AQ–Bus data bits AQD15,..,0 have no meaning on FCU3.

Tabelle12: AQ–Bus address ranges

Device	AQA3,..,0
FCU1	0
FCU2	1
FCU3	2
FCU4	3
FCU5	4
FCU6	5
FCU7	6
FCU8	7
RCU	8
GCU	9
unused	A,..,F

Tabelle13: AQ–Bus Function Codes

Function Codes AQS3,...,AQS0	Action of FIFO Controller	Destination Name	Mode R/W	Size of Oprd. Byte 3 2 1 0
0x1	AQ–STOP, stops the FCU Controller	FIFO Contr.	W	x x x x
0x2	AQ–NEXTIF, takes the 1. command of an IF sequence to FCON	FIFO Contr.	W	x x x x
0x4	AQ–NEXT, takes the next command to FCON	FIFO Contr.	W	x x x x

4. 5. 3. 2. Signal Description

An AQ–Bus command needs 1 20–MHz clock cycle to be transferred. All signals have to be valid and stable during this cycle.
Except the address and data bits there are two control signals which should be described.

AQSTROBE

The AQSTROBE indicates which 20–MHz clock cycle conveys a valid AQ–Bus command on the rest of the signal lines. To meet the real time requirements and the protocol of earlier versions of acquisition devices (RCU, GCU) the TCU3 provides 3 versions of the AQSTROBE derived from the same source bit w2_27 in word 2 of the Real Time Program entry.

AQEXEC

AQEXEC is active during AQ–Bus commands which have just to be executed and which are to get other not addressed devices to execute simultaneously their stored commands.

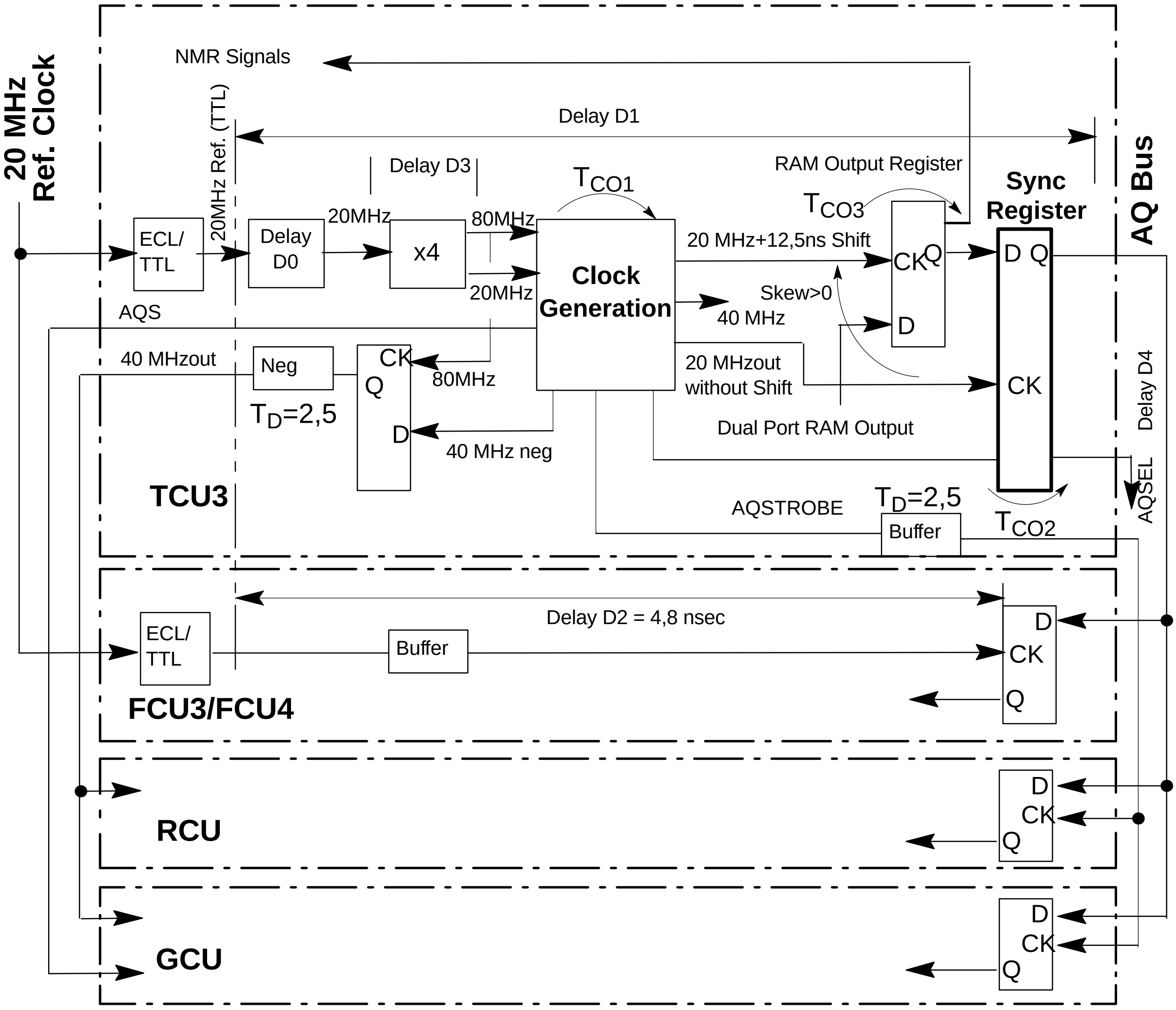
AQ–Bus devices which are addressed with a command containing AQEXEC as inactive store their command and execute it on the next command with AQEXEC active regardless of its address.

Signal Name	Description	Active Level	Location
AQSTROBE	1 edge,12,5 nsec before the end of the AQ–Bus cycle	low–high edge	J2_A32
AQSEL	active during the AQ–Bus cycle and the Duration like the address and data bits	high	J0_E18
AQS	Pulse width 25 nsec, from the middle to the end of the AQ–Bus cycle	low	Frontpanel connector
AQEXEC	active during the AQ–Bus cycle and the Duration like the address and data bits	low	J0_D18 J2_C32

4. 5. 3. 3. Timing relations on TCU3 and FCU3/FCU4

Wiring

Bild10: Wiring of Clocks and Real Time signals

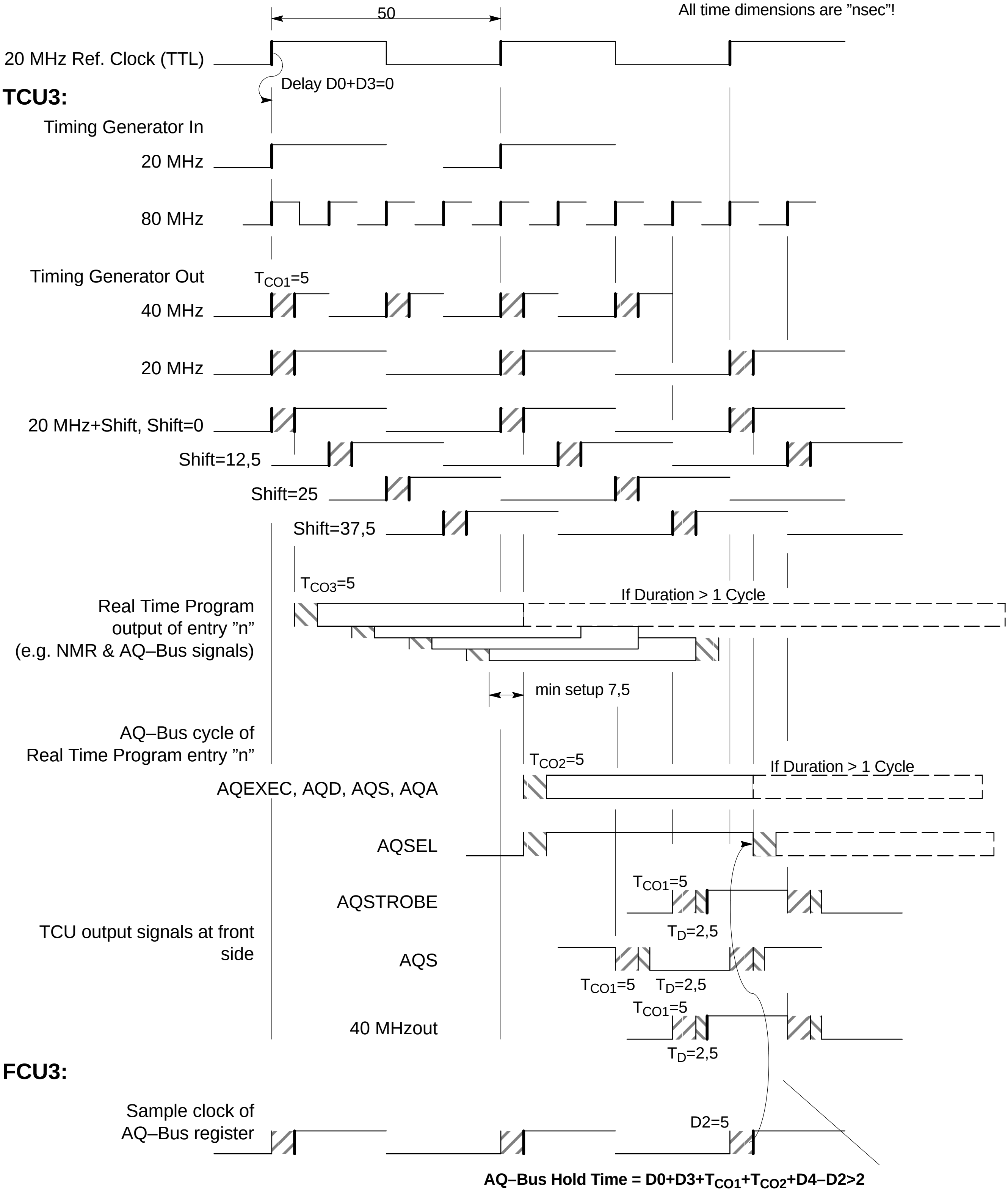


Timing

- The ECL/TTL transformers on all boards are assumed to have the same delay!
- D0 is the delay of the adjustment combination RB11A/CB6A on TCU3. So far this time constant is zero and D0 can therefor assumed to be zero too. Increasing the time constant expands the AQ–Bus Hold time on FCU4 from a minimum of T_{CO2}+D4 to a larger one.
  To avoid AQ–Bus communication errors the Hold time should be longer than 2nsec.
- D3 is the delay of the 80+20 MHz PLL and programmed to be zero.

- The following output signals of the Timing Generator are clocked out of the same device and can be assumed to have the same T_{CO1}:
  20 MHzout, 40 MHzout, 20 MHz+Shift, AQS, AQSTROBE, source of AQSEL

Bild11: Timing of Clocks and Real Time Signals between TCU3 and FCU3



4. 5. 4. The SGU Interface

This paragraph refers to the SGU Interface which uses National's "LVDS" device.

4. 5. 4. 1. Transfer register

Values written to this register will be sent to the SGU. It can be accessed from sources as follows:

- VME Bus
- SGU Data out of the FCU RAM and controlled by the FCU Controller
- SGU Data out of the FIFO and controlled by the FIFO Controller

VME Bus Device Codes

VME Bus Address	Device	Reg. Name	Mode R/W	Oprd. Byte 3 2 1 0
161FF204	SGU Interface Transfer Reg., Ch.1	pitra	W	b b b b
165FF204	SGU Interface Transfer Reg., Ch.2	pitra	W	b b b b
169FF204	SGU Interface Transfer Reg., Ch.3	pitra	W	b b b b
16DFF204	SGU Interface Transfer Reg., Ch.4	pitra	W	b b b b

4. 6. FIFO

The so called FIFO RAM is used as an asynchronous buffer between the TCU3 and the FCU4. It is used to store the commands sent by the TCU3 which will be executed in real time, triggered by the AQ_Bus. This F–Bus ensures an undisturbed command traffic.

A 8–K Word Dual Ported RAM is used to implement this FIFO. One port is read/writeable by the F–Bus and the other port transfers the FCMD (Instr./data) to the FIFO Controller or FCON (ASIC). The FIFO RAM is totally controlled by the FIFO Controller.

Format

<table>
<tr><th colspan="15">F–Bus Command</th><th rowspan="2">Comment</th></tr>
<tr><th>31</th><th>30</th><th>29</th><th>...</th><th>24</th><th>23</th><th>...</th><th>18</th><th>17</th><th>...</th><th>8</th><th>7</th><th>...</th><th>0</th><th></th></tr>
<tr><td>SW =0</td><td>FWAIT</td><td colspan="3">FITCU</td><td colspan="3">FADD Q–pointer</td><td colspan="3">Data</td><td colspan="4">S–Pointer</td><td>Command to the FCU Controller</td></tr>
<tr><td colspan="5">Interpreted by the FIFO Controller</td><td colspan="10"></td><td></td></tr>
<tr><td colspan="2"></td><td colspan="13">Interpreted by the FCU Controller(FCON)</td><td></td></tr>
</table>

FIFO Instructions

The Read Pointer will be incremented after any execution of a FIFO Instruction.

Binär	Hex	Instr. Memo	Comment
100000	20	NOOP	No Operation
100001	21	FSTOP	Stops the execution of an FCU List (ASIC)

Binär	Hex	Instr. Memo	Comment
100010	22	LDPREG	load Q_Pointer Preregister in ASIC the value is located in Bit 17..0
100011	23	LOADLP	load FIFO–Cont. loop register the value is located in Bit 15..0
100100	24	REPEAT	repeat a loop in the fifo after loadlp is specified
100101	25	REPEAT	repeat a loop in the fifo after loadlp is specified
100111	27	CLRLP	clears the loop counter if set with load loop remove the loop barrier
101000	28	INTRPT0	generate interrupt to TCU ; FCU status register ’RDSTAT’ Bit 0 and 15 are set
101001	29	INTRPT1	generate interrupt to TCU ; FCU status register ’RDSTAT’ Bit 1 and 15 are set
101010	2A	INTRPT2	generate interrupt to TCU ; FCU status register ’RDSTAT’ Bit 2 and 15 are set
101011	2B	INTRPT3	generate interrupt to TCU ; FCU status register ’RDSTAT’ Bit 3 and 15 are set
101100	2C	INSLOW	load Exchange Reg. to Instr. RAM (low word) the value is located in Bit 15..0
101101	2D	INSLHIGH	load Exchange Reg. to Instr. RAM (high word) the value is located in Bit 15..0
101110	2E	SGULOW	load Exchange Reg. to SGU RAM (low word) the value is located in Bit 15..0
101111	2F	SGUHIGH	load Exchange Reg. to SGU RAM (high word) the value is located in Bit 15..0

4. 7. FCU RAM

4. 7. 1. RAM organization

The size of the FCU RAM is 2 Mbyte. It is organized in 8–byte locations of 2 4–byte words. Each lower 4–byte part contains an FCU instruction or is used for a Slow Pointer register. The accompanying higher 4–byte contains the SGU Data of that FCU instruction.

The Access is possible from the VME Bus to 4–byte words or from the FCU Controller (FCON) to 8–byte words. The SGU Data part is transferred to the SGU interface on access of the instruction part by FCON.

The VME and the FCON access use a different address bit notation.

Map of the FCU RAM

8–byte wide FCU RAM		VME address	FCON address
4–byte SGU Data part	4–byte FCU Instruction part	A20,...,A0	B17,...,B0
VME addr. A2,A1,A0=1xx	VME addr. A2,A1,A0=0xx		
unused	16 4–byte registers of the 1. Slow Pointer (16 words)	0x0	0x0
		0x78	0xF
	16x255 4–byte range of the following 255 Slow Pointer (4k–16 words)	0x80	0x10
		0x7FF8	0xFFF
FCU Program range of 256k–4k 8–byte words		0x8000	0x1000
		0x1FFFF8	0x3FFFF

4. 8. FCU Controller

The FCU Controller FCON is the central unit of the FCU4. All access from VME Bus to the Insttr.–.or SGU RAM and to the SGU Interface are controlled by the ASIC FCON.

A command list in the FCU Instr. RAM controls the operation of this Sequencer. In this way the FCU can send digital informations about Frequency,Phase,Gating and Amplitude setting to the SGU. Different sets of Pointers (Q–Pointer,Slow–Pointer) can be used to address different instruction List.

The Q–Pointers are located in the ASIC and the Slow Pointer are in Instr. RAM. The usage of these Pointers is controlled by the FIFO Command Word (FCMD), which sends the TCU via F–Bus to the on board FIFO–RAM. The sequencer runs with the 20MHz System clock and is triggered by an TCU AQ–Bus commands. The execution of a Q–Pointer Instruction takes one cycle (50ns) whereby a Slow Pointer Instr. can take up to 5 cycles.

Slow Pointer Instructions

Binär	Hex	Instr. Memo	Comment
000000	00	NOOP	generate a TREQ Cycle but the ASIC (FCON) execute a NOOP Instr.
000001	01	EXONE	The FCU Instr. addressed by the Slow Pointer is executed and the Slow Pointer is incremented. Data Output to Serial Link not possible.

Binär	Hex	Instr. Memo	Comment
000010	02	INCRE	The S–Pointer is incremented by one.
000011	03	EXLIST	A FCU Instr. List is started. The ASIC (FCON) will continue the instruction List until it detect a STOP Instruction.
000100	04	RELOAD	The S–Pointer is loaded from the INIT REG. of the S–Pointer Register Block
000101	05	LOADP	The addressed S–Pointer is loaded from the FCU Instr. Exchange Register
000110	06	LOADDAT	The FCU Inst. RAM addressed by the S–Pointer is loaded from the FCU Instr. Exchange Register
000111	07	EXEC	The FCU Instr. addressed by the S–Pointer is executed and the S–Pointer is incremented. Data is transferred from the SGU RAM to the Serial Link (SGUI).
001000	08	LOADSGU	The SGU Data RAM addressed by the S–Pointer is loaded from the SGU Exchange Register

4. 8. 1. Q–Pointer

Inside the sequencer FCON there are 32 Quick Pointer (Q–Pointer) with 18 Bit in length. Each Q–Pointer consist of 3 registers:

- Current Register
- Begin Register
- End Register

These registers are read /writeable via the VME Bus. The F–Bus only can write the ’Current Register. The Current Register holds the actual address of the Q–Pointer.

The Current Register is loaded from the Begin Register on a QRELOAD Instr. or on a QRINCREX, QINCR Instr, if the current pointer has reached the value of the End Register.

On QDECR Instr. the Current Register is updated with the value of the End Register if the current pointer has reached the value of the Begin Register. The Q–Pointer directly addresses the Instr./SGU RAM.

Quick Pointer Instructions

Binär	Hex	Instr. Memo	Comment
010000	10	QRELOAD	The Q–Pointer is loaded with the value of the Q–Pointer start Reg.
010001	11	QINCR	The Q–Pointer is incremented by one
010010	12	QEXINCR	The FCU Instr. addressed by the Q–Pointer is executed and the Q–Pointer is incremented
010011	13	QLOAD	The Q–Pointer is loaded from the Preregister (LDPREG) in the ASIC
010100	14	QDECR	The Q–Pointer is decremented by one
010101	15	QLDSGU	The SGU RAM (Add. by the Q–Pointer) is loaded from the SGU Exchange register

Binär	Hex	Instr. Memo	Comment
010110	16	QEXEC	The FCU Instr. addressed by the Q–Pointer is executed
010111	17	QNOOP	generate a QREQ Cycle but the ASIC (FCON) execute a NOOP Instr.
011000	18	QRELDEX	The Q–Pointer is loaded with the value of the Q–Pointer start Reg. and the FCU Instr. addressed by this Q–Pointer is executed
011001	19	QEND	The QREQ Signal is released without any further action
110101	35	QFIFOSGUA	The serial Link is loaded from the SGU Exchange Reg.
110010	32	QRINCREX	The FIFO Controller repeats until the loop counter is zero; The Q–Pointer is incremented by one and the FCU Instr. addressed by the Q–Pointer is executed.
110110	36	QREXE	The FIFO Controller repeats until the loop counter is zero; The FCU Instr. addressed by this Q–Pointer is executed

4. 9. Device access via external Busses

4. 9. 1. JTAG access

The JTAG Interfaces is designed for the programming of JTAG programmable logic devices arranged in two JTAG chain's.

It is based on the National's JTAG Bridge SCANPSC110F.

The programming procedure is described in detail in the "AQX Test Manual"

4. 9. 2. BBIS access

There is one physically BBIS Prom on the FCU4. It respond to each logical channel, adressed via BBIS–Access.

The FCU BBIS Proms respond to the following BBIS addresses.

Device	SBA binary										SBA hex
	9	8	7	6	5	4	3	2	1	0	
	Proto-col bit	Group address						EEPROM address			
	P0	A5	A4	A3	A2	A1	A0	A2	A1	A0	
FCU Channel 1	0	0	0	1	0	0	0	0	0	0	040
FCU Channel 2	0	0	0	1	0	0	1	0	0	0	048
FCU Channel 3	0	0	0	1	0	1	0	0	0	0	050
FCU Channel 4	0	0	0	1	0	1	1	0	0	0	058
FCU Channel 5	0	0	0	1	1	0	0	0	0	0	060
FCU Channel 6	0	0	0	1	1	0	1	0	0	0	068
FCU Channel 7	0	0	0	1	1	1	0	0	0	0	070
FCU Channel 8	0	0	0	1	1	1	1	0	0	0	078