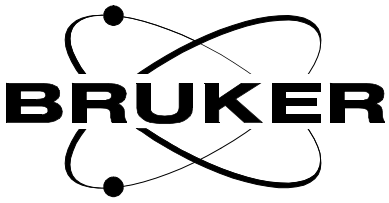


Frequency Control Unit FCU3

AQS
Technical Manual

Version 001

BRUKER

The information in this manual may be altered without notice.

This manual was written by

Thomas Eckert, Norbert Kuntz, Willy Uhrig

Rheinstetten, Germany

P/N: Z31563
DWG-Nr: 1275001

AQS FREQUENCY CONTROL UNIT, FCU3

Kuntz, Eckert, Uhrig

Contents

Figures

Tables

1. Starting with FCU3

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 FCU3

One FCU3 provides two separate Frequency Channels containing in each channel:

- 64Kx32 Bit (optional 256Kx32) Program Memory (FCU instruction list)
- 64Kx32 Bit (optional 256Kx32) SGU Data RAM used to store the amplitude, phase, frequency, gating ec. values for the SGU.
- 8Kx32 Bit dual port fifo ram to store the commands transferred via FBus
- FBus interface removes the timing bottleneck on the AQBus
- minimal execution of a FCU instruction 50ns

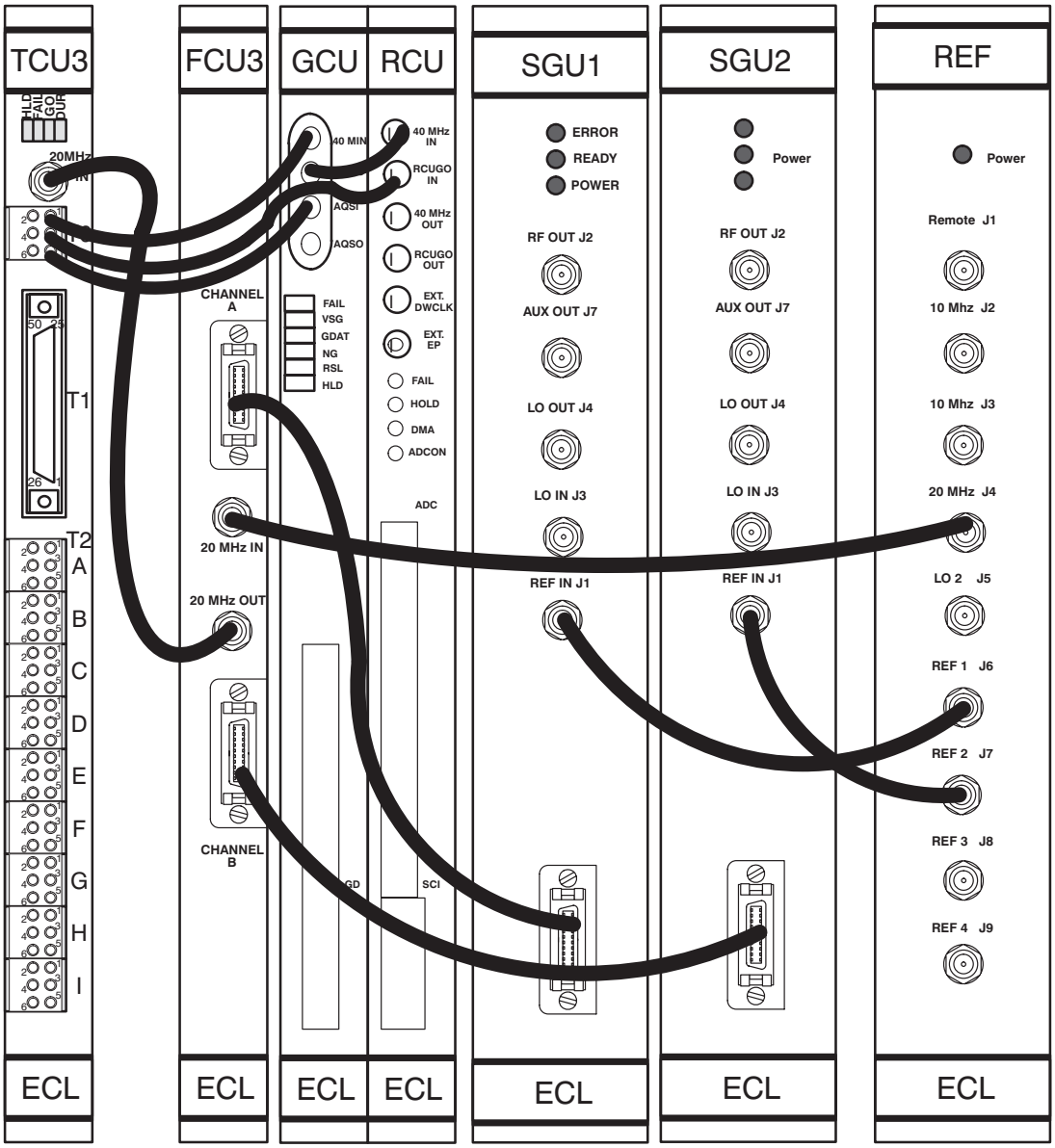
1. 2. Hardware Implementation

FCU3 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.

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.

Figure 1: TCU3,FCU3 and SGU connection



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

Four FCU3 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. FCU3 is provided with a self configuring scheme which works as follows:

Configuration Scheme

1. Each Channel A becomes the odd channel and each Channel B the even one.

2. The leftmost FCU3 (looking at the front side) configures itself to be FCU Channel 1 and 2. A FCU3 in the next position at their right side will be FCU Channel 3 and 4 and so forth.

3. Gaps or other devices between FCU3 lead to a second Channel 1/2 pair and to malfunctioning and should be avoided.

If any reason would necessitate gaps between FCU's, jumpers are provided on FCU3 (W15) which can override the configuration scheme.

1. 3. Software Implementation

XWIN NMR

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

FCU Test

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

The `fcutest` recognizes the FCU version (FCU0 or FCU3) 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 $\geq$ 01	AQX	
FCU0 / 256k	H2564	H3P1940A			
FCU0 / 64k	H2556	H3P1940A	EC $\geq$ 05	AQX	
FCU0 / 256k	H2564	H3P1940A			
TOMO FCU 64k	T5565	H3P1940A	EC $\geq$ 01	AQX	
FCU3 / 64k	H5822	H3P2260D	EC $\geq$ 00	AQS	XWIN−NMR Version 3.0
FCU3 / 256k	H9598	H3P2260D	EC $\geq$ 00	AQS	

Table 1: FCU versions

2. 2. Features

- Minimal duration 50 nsec
- Maximal duration 53 sec
- Minimal resolution 50 nsec
- Real time memory range max. 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
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
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

Name	Abbr.	Operation
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 FCU3 provides 2 channels

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

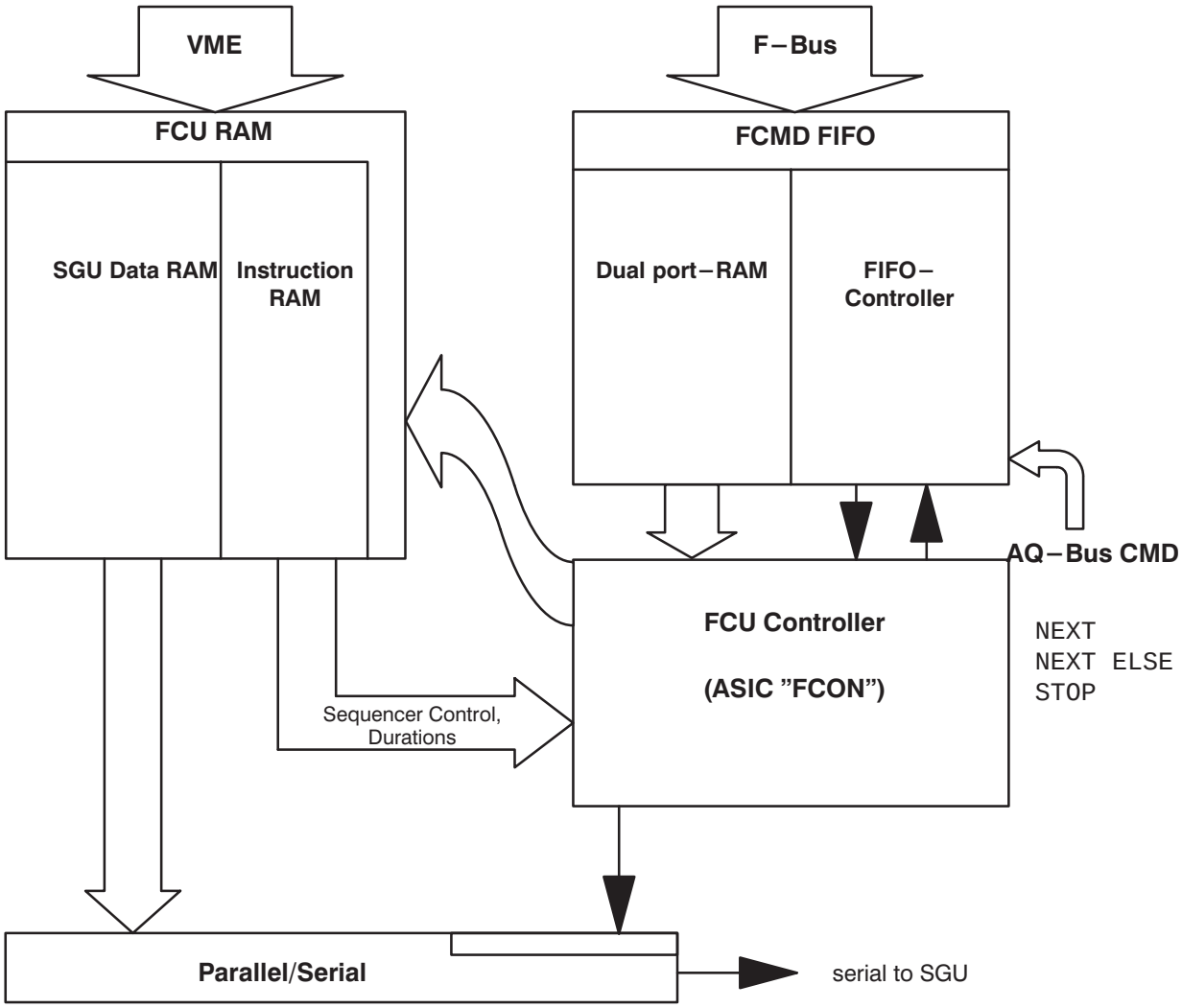
4. The timing of generating F–Bus Commands on TCU and the output sequence of SGU Data are made independent of each other by the FCMD FIFO

5. F–Bus between TCU3 and FCU3

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

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

Figure 2: Functional diagram



2. 4. Construction

The FCU3 is a VME Bus module of 4 TE with an extended length. It consists of one printed circuit board containing 2 identical "frequency channels" called "A" and "B"

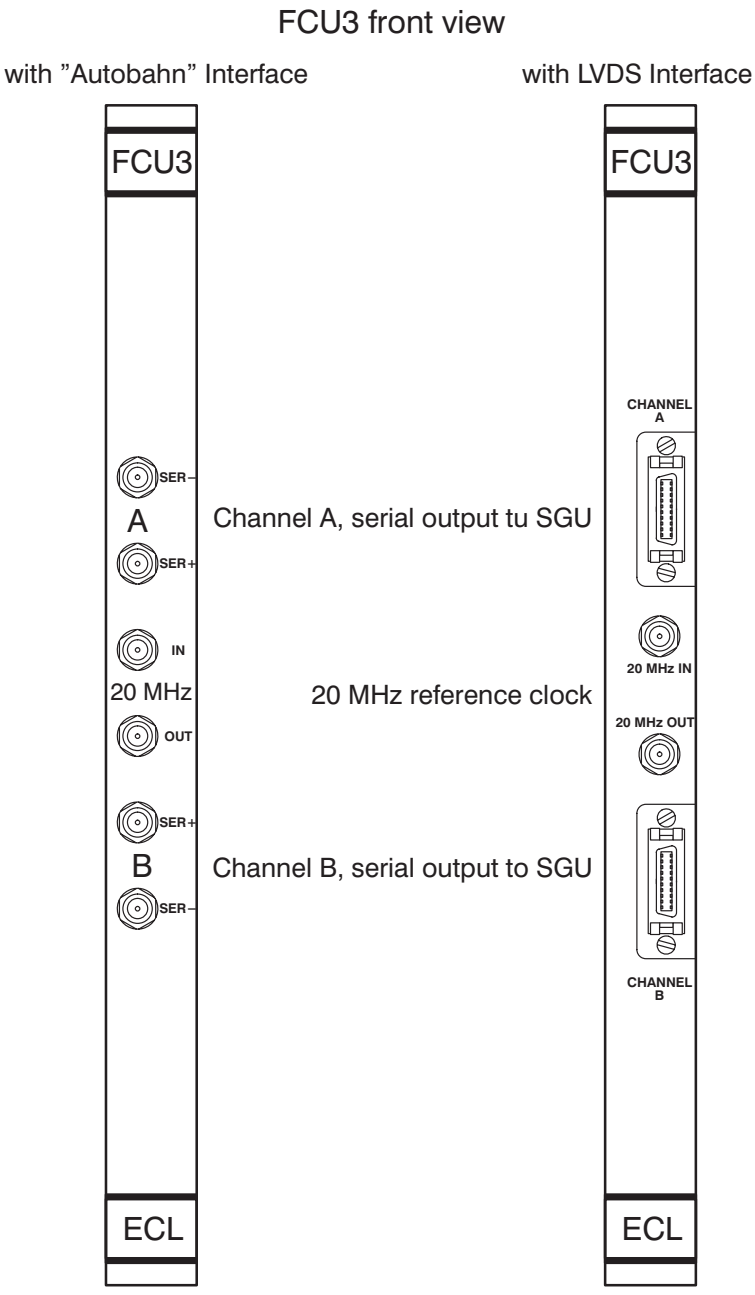
Four FCU3 modules can be combined to provide the 8 frequency channels F1 to F8. These modules have to be mounted without any gap to enable them to configure itself the channel numbers subsequently.
That means the leftmost FCU3 configures its channels A and B to be F1 and F2. Any FCU3 mounted in the next slot to the right configures its channel A and B to be F3 and F4 and so on. Any other module than FCU3 or a gap between 2 FCU3 result to a second FCU configured to have F1 and F2.

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.

Front View

Figure 3: Front View



2. 5. Part numbers of FCU3

Table 2: Table of Assembly Groups

Amount	Title	Function	Part–Nr.
1	FCU3	Assembled PCB with 64k words of FCU RAM	H5822
1	FCU3	Assembled PCB with 256k words of FCU RAM	H9598

Amount	Title	Function	Part−Nr.
1	FCU3	PAL set	H5824
1		Layout	H3P2260D
1		Plain PCB	H5823
1		Frontpanel	Hz06521
1		Frontpanel Assembly Set	
1		Frontpanel Ident	
1	SGU INTERFACE	Assembled PCB with LVDS Interface	Z003564
		Assembled PCB with ”Autobahn” Interface	not introduced
2	FIFO−CONTROLLER	Assembled PCB	H5820
		Layout	H4P2370A
		Plain PCB	H5821

2. 6. Accessories

Table 3: 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
Cable 6P TCU T0 connector	HZ10104/B
SGU Interface Cable	Z13928

Connecting the 20 MHz reference clock

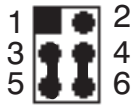
TCU3 includes the termination resistors of the 20 MHz reference clock line from the Reference Unit. So it has to be the last device and the Reference Unit the first one at this line.

2. 7. Operational Settings

2. 7. 1. Jumper Setting

Auto Configuration of Frequency Channels

FCU3 is provided with a self configuring scheme which works as follows:



Auto configuration enabled (default)

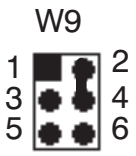
- Each Channel A becomes the odd channel and each Channel B the even one.
- The leftmost FCU3 (looking at the front side) configures itself to be FCU Channel 1 and 2. A FCU3 in the next position at their right side will be FCU Channel 3 and 4 and so forth.

Disableing the auto configuration

Gaps or other devices between FCU3 lead to a second Channel 1/2 pair and to mal–functioning and should be avoided. If any reason would necessitate gaps between FCU's, the jumper W9 is provided to configure a fixed FCU channel pairs.

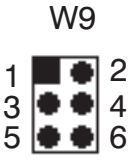
Channel 1/2

Channel 3/4

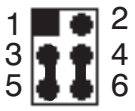


Channel 5/6

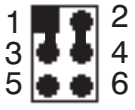
Channel 7/8



FCU3 Clock Selection



Default !!
The FCU3 Board is provided with the 20 MHz clock from the referenz Board



Only for Test !!
The FCU3 Board use the internal 20 MHz Clock.

ASIC Delayed Clock Configuration

A programmable delay line on the FCU3 is used to provide the ASIC FCON with a delayed 20MHz Clock Signal. This Jumper must not be changed because the ASIC may not properly work on any other configuration.

5–6	3–4	1–2	Clock Delay
IN	IN	IN	6 ns
IN	IN	OUT	7 ns
IN	OUT	IN	8 ns
IN	OUT	OUT	9 ns
OUT	IN	IN	**10 ns default**
OUT	IN	OUT	11 ns
OUT	OUT	IN	12 ns
OUT	OUT	OUT	13 ns

Selection of In System Programming path (ISP)

1–2	JTAG Bridge
OUT	disabled –––> ISP via ST5,ST6
IN	enabled (default) ISP via VME Extener

Figure 4: Jumper locations

2. 7. 2. Firmware version

FCU3 doesn't need any Firmware

2. 7. 3. RAM Configuration

FCU3 can be equipped with 64k words of FCU RAM, 64–bit each or 256k words of FCU RAM. Each channel needs 2 modules of 64k or 256k words of 32 bits:

Exchanging the modules doesn't need any Jumper setting.

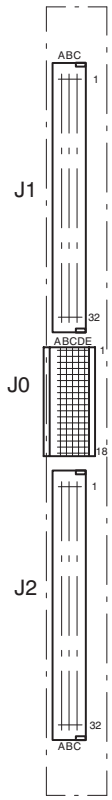
Versions	Part No.	Layout No.	SRAM Module	Part#
FCU3 / 64k	H5822	H3P2260D	MT8S6432Z–20	42001
FCU3 / 256k	H9598	H3P2260D	MT8S25632Z–20	84319

2. 8. Connectors and Signal Allocations

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

Figure 5: 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

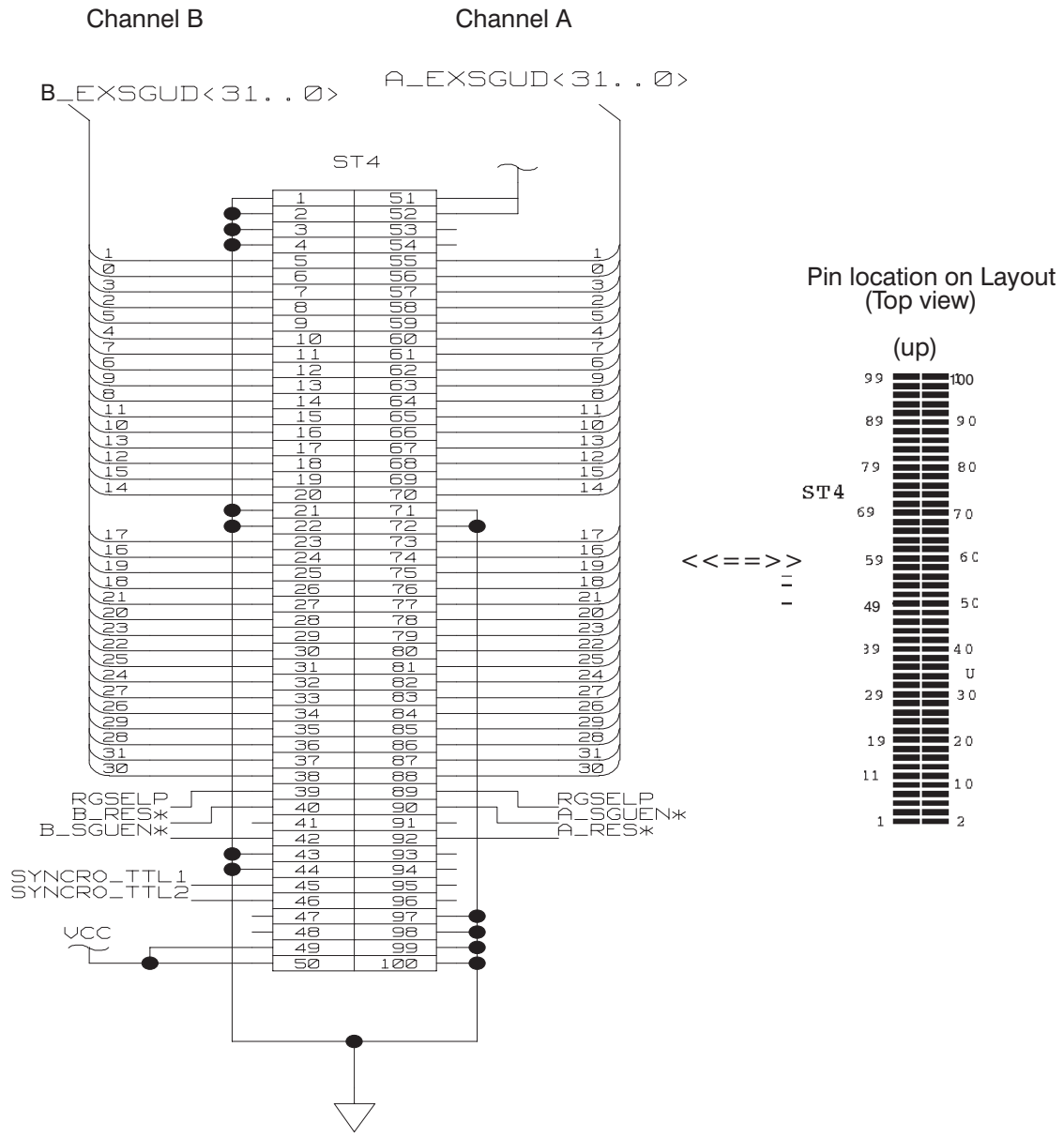
Connector J0

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

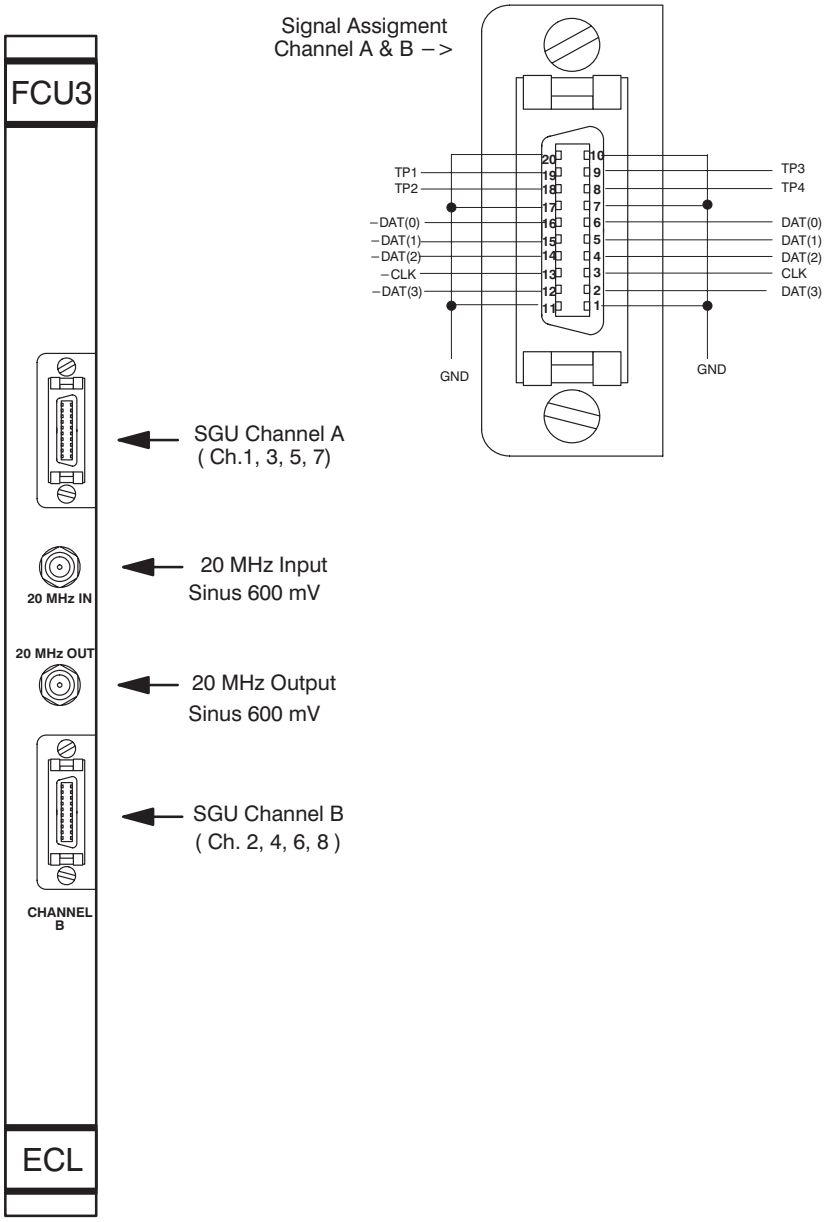
2. 8. 2. Pin assignment of the Connector ST4 between FCU and SGU Interface

Figure 6: Interface Connector ST4



2. 8. 3. Pin assignment of the SGU Interface Connector

Figure 7: SGU Interface Connector
FCU3 front view



2. 9. Power Requirements

The FCU3 requires power supply of the following voltages:

	Part–No.	+5 V	+12 V	–12 V	+3,3V J0: A5, B5, C5, D5, E5
FCU3	H5822	3.9A			

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 FCU3 contains in system programmable logic devices which need a "`Prog File_yymmddECxx`" 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 and chain description files

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

Teach yourself in referring to the AQX Test Manual.

3. 2. Prototype Situation

FCU3 of the Introduction Status show no functional problems up to January 99.

Prior to Introduction delivered FCU3 of prototype status had been updated to the introduced one of EC00.

3. 3. Introduction Status

3. 3. 1. Prog Pal

Table 4: FCU3 Layout Version H3P2260D, Prog–PAL H5824

FCU3 Prog_PAL H5824, EC00						
Programmed Device				Device		
IC#	Checksum	PAL Name	H–Nr.	Type	Package	Part#
U77	0000	FCU3AA01	H5825	MACH231–10	PLCC 84	67858
U52	935B	FCU3AB02	H5826	MACH211–7	PLCC 44	68089

FCU3 Prog_PAL H5824, EC00						
Programmed Device				Device		
IC#	Checksum	PAL Name	H–Nr.	Type	Package	Part#
U25	189A	FCU3AA03	H5827	MACH231–10	PLCC 84	67858
U51	767C	FCU3AA04	H5828	PAL22V10–5	PLCC 28	42492
U66	70E6	FCU3AA05	H5829	PAL22V10–5	PLCC 28	42492
		FCU3AA06	H5830	BIS Prom IC24022	DIL 8	65415

3. 3. 2. Prog File

Table 5: FCU3 Layout Version H3P2260D, Prog–File H9579

FCU3 Prog_File H9579 : FCU3_981016EC00					
Programmed Device			Device		
IC#	Checksum	PAL Name	Type	Package	Part#
U3	4DE3	FCU3AA07	MACH5–512/160–10	PQFP 208	69653
U1	CC09	FCU3AA08	MACH4–128/64–7	PQFP 100	69654
U2	1B31	FCU3AA09	MACH4–128/64–7	PQFP 100	69654

3. 3. 3. Modifications of the introduced layout

3. 3. 4. Jumper Setting

Introduced status of Jumper setting:

3. 3. 4. 1. Firmware

FCU3 doesn’t need any Firmware

3. 4. History of Modifications

3. 4. 1. FCU3 with 64k Memory

EC No.	Date	Part Number	Description of Bugs, Changes and Modifications	Ser.No.	New EC–Level
2536	8.1.99	5822	Introduction of FCU3, 64k words	0010	00
2575	21.4.99	5822	Grounding the TEST_SE input of ASIC FCON, connect pin 148 with pin 156	0010	01

EC No.	Date	Part Number	Description of Bugs, Changes and Modifications	Ser.No.	New EC – Level
2598	8.7.99	5822	Modification of Clock	0141	02
2623	14.9.99	5822	Introduction of Layout H3P2260E which includes all former modifications		03

3. 4. 2. FCU3 with 256k Memory

EC No.	Date	Part Number	Description of Bugs, Changes and Modifications	Ser.No.	New EC – Level
2574 2574A	16.4..99 30.6.99	9598	Introduction of FCU3, 256k words Modification of Clock	0010	00
2623	14.9.99	9598	Introduction of Layout H3P2260E which includes all former modifications		01

3. 4. 3. FIFO Controller

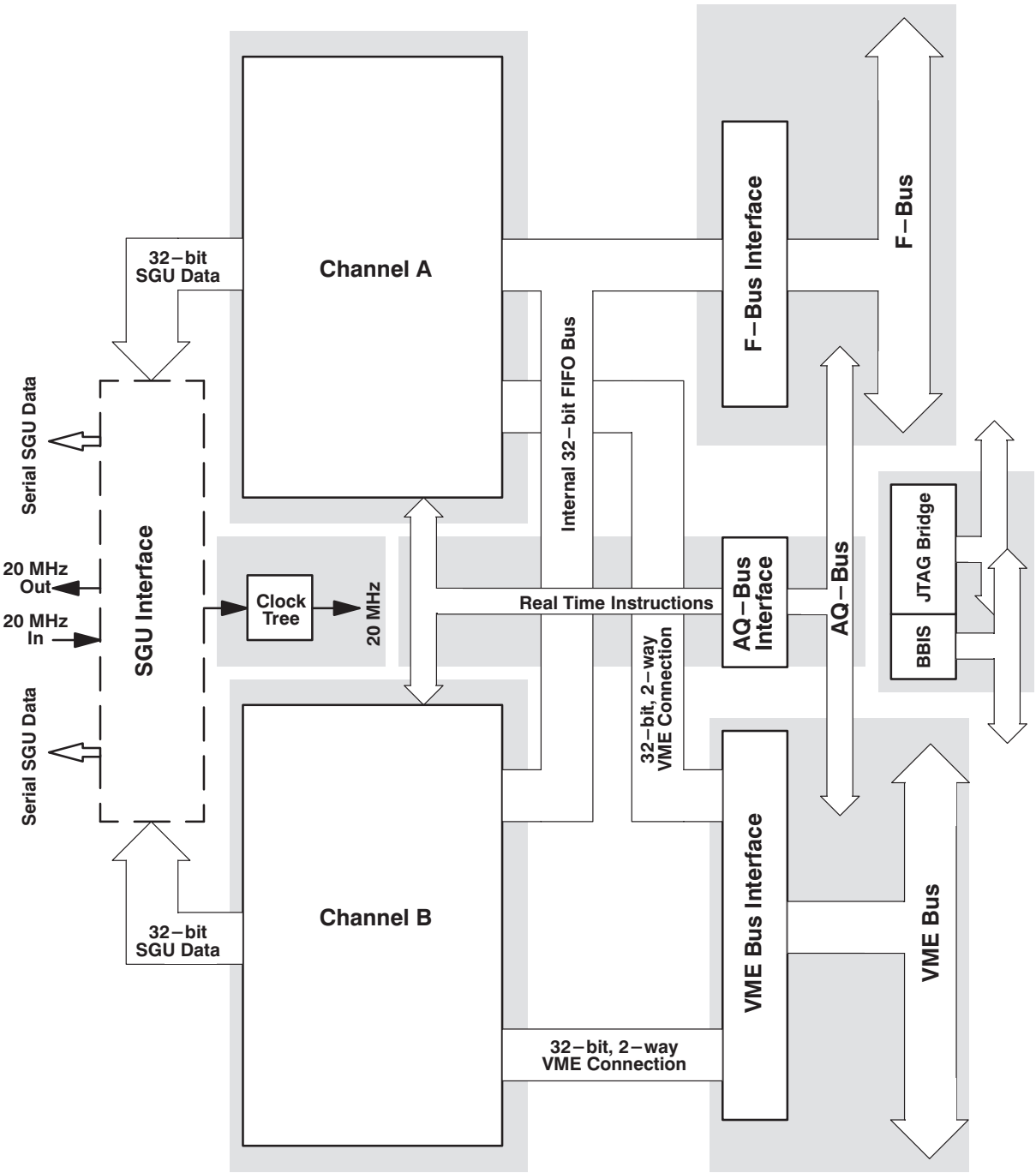
EC No.	Date	Part Number	Description of Bugs, Changes and Modifications	Ser.No.	New EC – Level
2618	30.8.99	H5820	Introduction of the FIFO Controller, layout h4p2370a Prog file FCU3_161098EC00	0010	00
2617	19.8.99	H5820	Introduction of the FIFO Controller using a manufactuerer programmed ACTEL FPGA, layout h4p2500	0350	20

4. Condensed Description

4. 1. Construction and Configuration

One FCU3 board contains 2 separate Frequency Channels called ”Channel A” and ”Channel B”.

Figure 8: Block diagram of FCU3



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

Four FCU3 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. FCU3 is provided with a self configuring scheme which works as follows:

Configuration Scheme

1. Each Channel A becomes the odd channel and each Channel B the even one.

2. The leftmost FCU3 (looking at the front side) configures itself to be FCU Channel 1 and 2. A FCU3 in the next position at their right side will be FCU Channel 3 and 4 and so forth.

3. Gaps or other devices between FCU3 lead to a second Channel 1/2 pair and to malfunctioning and should be avoided.
If any reason would necessitate gaps between FCU’s jumpers are provided on FCU3 which can override the configuration scheme.

4. 2. Channel Architecture

As brought out in Figure 9: the FCU3 can be considered to comprise these functional units:

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

Figure 9: Block diagram of one FCU3 channel

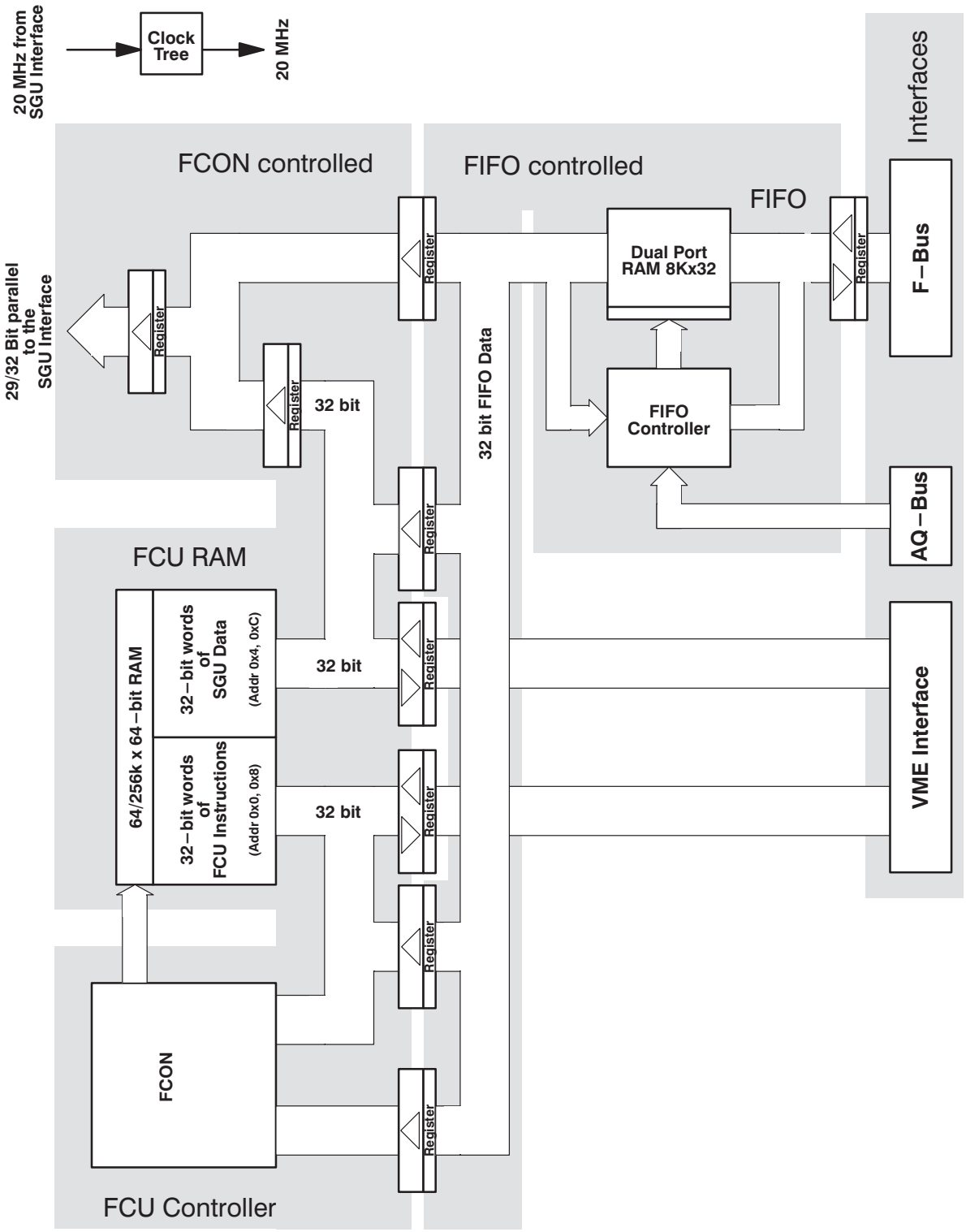
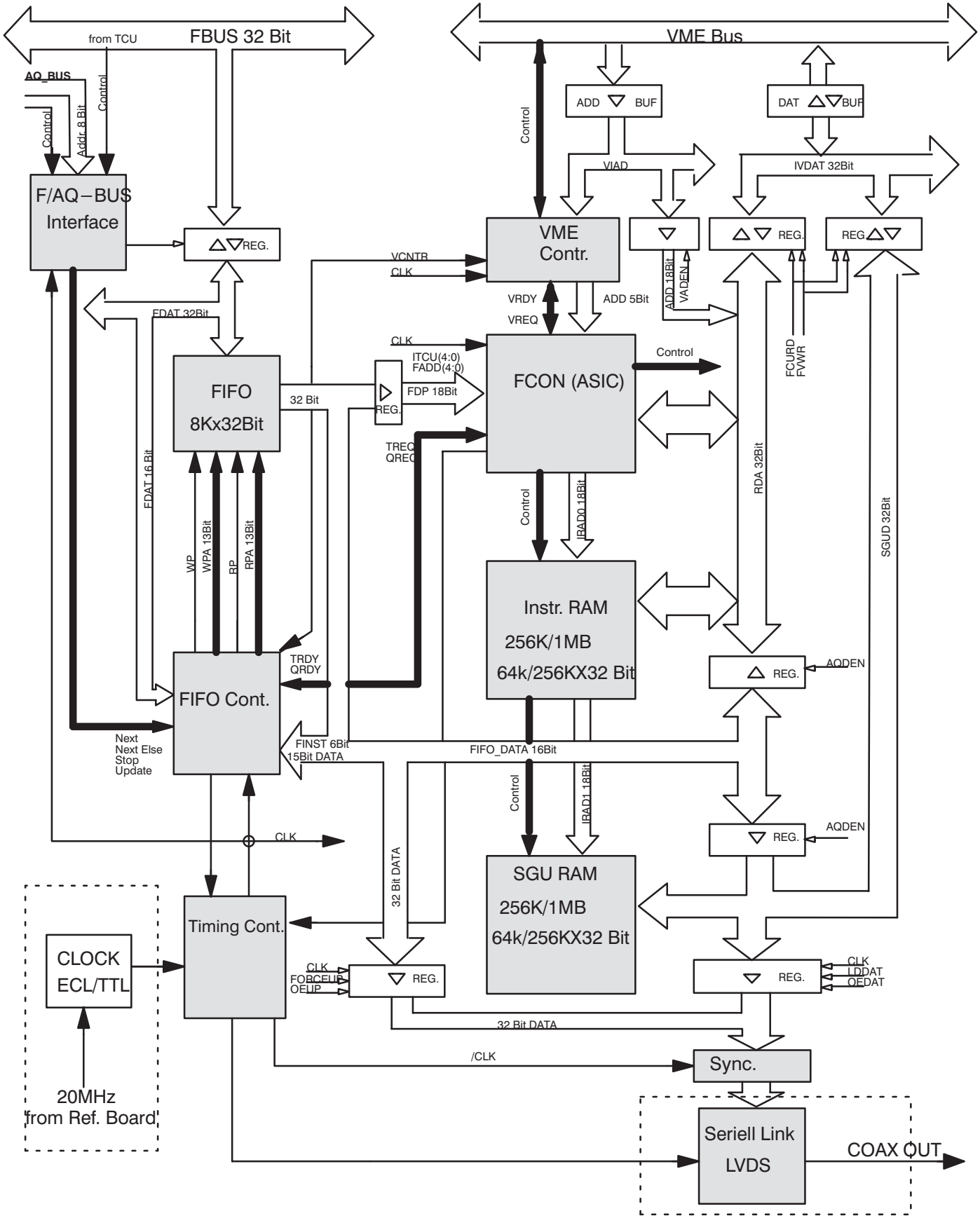


Figure 10: Functional Diagram of one channel



4. 3. Summary of Logical References

VME Bus Address Codes

Table 6: 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.

Table 7: VME Bus Address Map and Device Codes

Shown are the address codes of channel A and B configured as FCU Channel 1 and 2.

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
161FF200	SGU Interface Control Reg., Ch.1	pictr	W	x x b b
161FF204	SGU Interface Transfer Reg., Ch.1	pitra	W	b b b b
161FF300	Configuration Reg.0 Ch.1 + Ch.2	config0	R/W	x x b b
161FF304	Configuration Reg.1 Ch.1 + Ch.2	config1	R	x x x b
161FF308	Configuration Reg.2 Ch.1 + Ch.2	config2	R	x x x b
161FF30C	Configuration Reg.3 Ch.1 + Ch.2	config3	R	x x x b
161FF320	Software Reset, Ch.1	vres	W	x x x x
161FF340	Run FCU, Ch.1	vrun	W	x x x x
161FF360	Step FCU, Ch.1	vstep	W	x x x x
161FF380	Stop FCU, Ch.1	vstop	W	x x x x
161FF3A0	Read RAM Addr & Instr Status, Ch.1	rdra	R	b b b b
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

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
165FF300	Configuration Reg.0 Ch.1 + Ch.2	config0	R/W	x x b b
165FF304	Configuration Reg.1 Ch.1 + Ch.2	config1	R	x x x b
165FF308	Configuration Reg.2 Ch.1 + Ch.2	config2	R	x x x b
165FF30C	Configuration Reg.3 Ch.1 + Ch.2	config3	R	x x x b
165FF320	Software Reset, Ch.2	vres	W	x x x x
165FF340	Run FCU, Ch.2	vrun	W	x x x x
165FF360	Step FCU, Ch.2	vstep	W	x x x x
165FF380	Stop FCU, Ch.2	vstop	W	x x x x

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.

Table 8: 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.

Table 9: 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 FCU3.

Table 10: 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

Table 11: 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

1. Interfaces

2. Buses controlled by the FIFO Controller

3. 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 two 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 FCU3 has two channels and each channel has a separate device code for reset. The VME Bus Sysreset line initializes both channels of the FCU3. 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 ’vstep’

VME Bus Address	Device	Reg. Name	Mode R/W	Oprd. Byte 3 2 1 0
161FF360	Step FCU, Ch.1	vstep	W	x x x x

This device code is only used for test purposes to step trough a FCU program List or to test internal functions of the FCON. The Clock of the FCU Controller FCON is stopped (no toggle) and each new vstep command will generate a low or high transition of the ASIC clock. In this way two vstep commands will generate one clock cycle of the FCON.

4. 5. 1. 4. VME Dev. Code ’vrun’

VME Bus Address	Device	Reg. Name	Mode R/W	Oprd. Byte 3 2 1 0
161FF340	Run FCU, Ch.1	vrun	W	x x x x

The command ’vrun’ cancels the ’vstep command. The 20MHz clock to the ASIC witch was stopped by the vstep command is running .

4. 5. 1. 5. VME Dev. Code ’rdra’

VME Bus Address	Device	Reg. Name	Mode R/W	Oprd. Byte 3 2 1 0
161FF3A0	Read RAM Addr & Instr Status, Ch.1	rdra	R	b b b b

This command reads the address lines and the internal status of the ASIC FCON. It is only used for debugging of the ASIC address and data path. The format of the data read is as follows:

Bit <17..0>	Address of the current used pointer
Bit <19>	FCON Sequencer RUN Bit
Bit<20>	CARRY Bit of the current slow pointer
Bit<27..23>	internal State I–lines
Bit<32..28>	internal State Y–lines

4. 5. 1. 6. 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

4. 5. 1. 7. Configuration register

The Configuration register comprises 4 7–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 + Ch.2	config0	W	x x b x
161FF300	Configuration Reg.0 Ch.1 + Ch.2	config0	R	x x x b
161FF304	Configuration Reg.1 Ch.1 + Ch.2	config1	R	x x x b
161FF308	Configuration Reg.2 Ch.1 + Ch.2	config2	R	x x x b
161FF30C	Configuration Reg.3 Ch.1 + Ch.2	config3	R	x x x b

Bit Allocation

config0	Only 3 bits are valid. Write on Bits 9,8,7 set the TestA,TestB,TestC inputs of the duration counter. These bits are connected to the FCU Controller FCON. They will be used to test the duration counter in the FCON. The value of these bits can be read back on D0..D2. Default value is 0.
config1	D6....D0 contain the binary code of the slot number.
config2	D3....D0 contain the Layout version of the FCU D6....D4 contain the FCU version (0x3 => FCU3)
config3	reserved

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 TCU3

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.

Table 12: 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

Table 13: Codes of F−Bus connected devices on FCU3

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

Protocol

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 FCU3
- 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.

Table 14: 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

Table 15: 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.

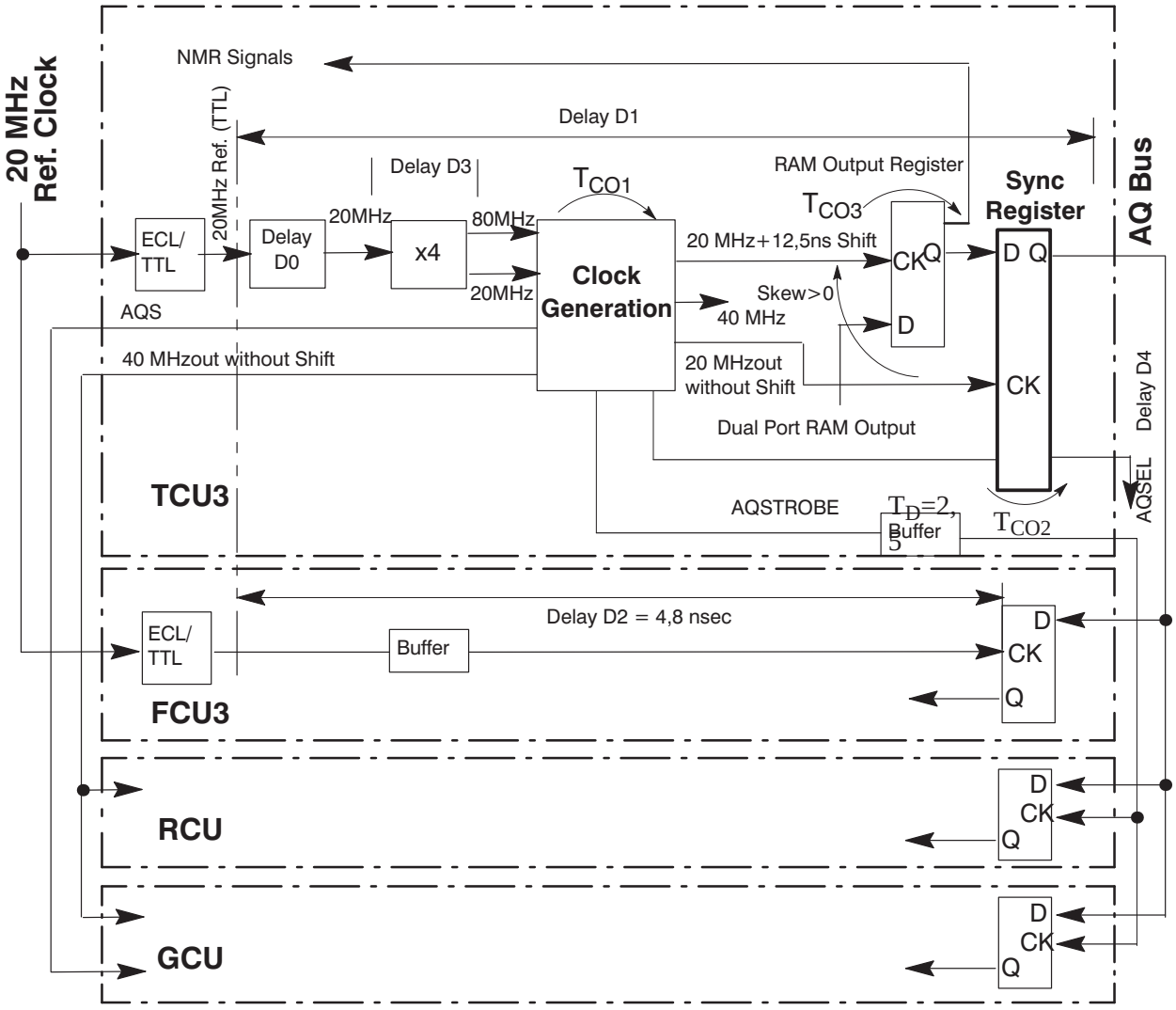
AQ–Bus devices which are addressed with a command containing AQEXEC as inactive store their command and execute it on the next command with AQEXEC active regardless of its address.

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

Wiring

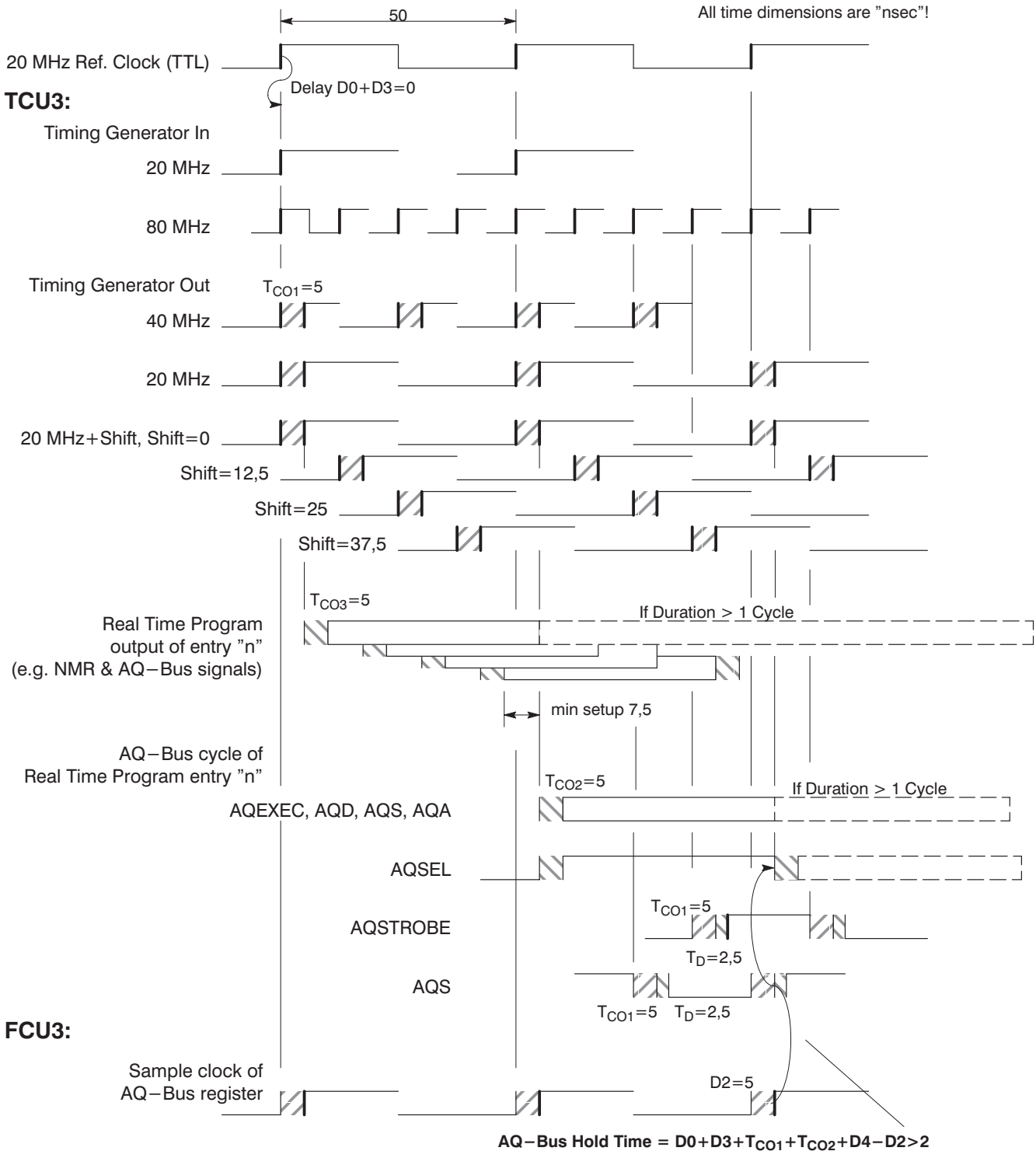
Figure 11: Wiring of Clocks and Real Time signals



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 FCU3 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

Figure 12: 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. Controllregister

The Controllregister can be accessed via VME bus only. Values written to this register define the transfer mode and can set the initial state.

VME Bus Device Codes

VME Bus Address	Device	Reg. Name	Mode R/W	Oprd. Byte 3 2 1 0
161FF200	SGU Interface Control Reg., Ch.1	pictr	W	x x b b
165FF200	SGU Interface Control Reg., Ch.2	pictr	W	x x b b

Meaning of values

0xDF	Transfer mode 32 Bit, ignore Timeout, This value has to be written after soft or hardware reset and after clearing all internal registers.
0xDE	Clear all internal registers of the AUTOBAHN device (not necessary after reset)

4. 5. 4. 2. 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

4. 6. FIFO

The so called FIFO RAM is used as an asynchronous buffer between the TCU3 and the FCU3. 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/write-able 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="12">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></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="3">S−Pointer</td><td>Command to the FCU Controller</td></tr>
<tr><td colspan="6">Interpreted by the FIFO Controller</td><td colspan="7"></td><td></td></tr>
<tr><td></td><td colspan="12">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)
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

<table>
<tr><th colspan="2">8−byte wide FCU RAM</th><th>VME address</th><th>FCON address</th></tr>
<tr><th>4−byte SGU Data part</th><th>4−byte FCU Instruction part</th><th rowspan="2">A20,...,A0</th><th rowspan="2">B17,...,B0</th></tr>
<tr><th>VME addr. A2,A1,A0=1xx</th><th>VME addr. A2,A1,A0=0xx</th></tr>
<tr><td rowspan="5">unused</td><td rowspan="2">16 4−byte registers of the 1. Slow Pointer
(16 words)</td><td>0x0</td><td>0x0</td></tr>
<tr><td></td><td></td></tr>
<tr><td>0x78</td><td>0xF</td></tr>
<tr><td rowspan="3">16x255 4−byte range of the following 255 Slow Pointer
(4k−16 words)</td><td>0x80</td><td>0x10</td></tr>
<tr><td></td><td></td></tr>
<tr><td>0x7FF8</td><td>0xFFF</td></tr>
<tr><td colspan="2" rowspan="3">FCU Program range of
256k−4k 8−byte words</td><td>0x8000</td><td>0x1000</td></tr>
<tr><td></td><td></td></tr>
<tr><td>0x1FFFF8</td><td>0x3FFFF</td></tr>
</table>

4. 8. FCU Controller

The FCU Controller FCON is the central unit of the FCU3. 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

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	
	Protocol 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