

Timing Control Unit TCU3

**AQS
Technical Manual**

Version 001

BRUKER

The information in this manual may be altered without notice.

BRUKER accepts no responsibility for actions taken as a result of use of this manual. BRUKER accepts no liability for any mistakes contained in the manual, leading to coincidental damage, whether during installation or operation of the instrument. Unauthorised reproduction of manual contents, without written permission from the publishers, or translation into another language, either in full or in part, is forbidden.

This manual was written by

Mirek Dude, Thomas Eckert, Norbert Kuntz, Dr. Jens Rommel

© November 11, 1999: Bruker Analytik GmbH

Rheinstetten, Germany

P/N: Z31562
DWG-Nr: 1274001

AQS Timing Control Unit, TCU3

N. Kuntz/ T. Eckert / M. Dudek/ J.–M. Rommel

Contents

| | |
|---|-----------|
| 1. Starting with TCU3 | 1 |
| 1. 1. Special features of TCU3 | 1 |
| 1. 2. Hardware Implementation | 1 |
| 1. 3. Software Implementation | 2 |
| 1. 4. Modifications | 3 |
| 2. Specifications | 4 |
| 2. 1. TCU Versions | 4 |
| 2. 2. Features | 4 |
| 2. 3. Construction | 5 |
| 2. 4. Part numbers of TCU3 | 6 |
| 2. 5. Accessories | 7 |
| 2. 6. Controls and Indicators | 7 |
| 2. 7. Operational Settings | 8 |
| 2. 7. 1. Firmware | 8 |
| 2. 8. Jumper | 9 |
| 2. 9. Installing SRAM | 10 |
| 2. 10. Connectors and Signal Allocations | 11 |
| 2. 11. Power Requirements | 12 |
| 3. Service Information | 13 |
| 3. 1. New Procedures | 13 |
| 3. 2. Prototype Situation | 13 |
| 3. 3. Introduction Status | 14 |
| 3. 3. 1. Prog Pal | 14 |
| 3. 3. 2. Prog File | 14 |
| 3. 3. 3. Modifications of the introduced layout H3P2310A | 15 |
| 3. 3. 4. Modifications of the introduced layout H3P2310B | 16 |
| 3. 3. 5. Jumper Setting | 16 |
| 3. 3. 6. Firmware | 17 |
| 3. 4. History of Modifications | 17 |
| 4. Condensed Description | 18 |
| 4. 1. Architecture | 18 |
| 4. 2. Summary of Logical References to memory and I/O devices | 19 |
| 4. 3. i960 Processor Unit | 21 |
| 4. 3. 1. i960 defined address ranges | 22 |
| 4. 3. 2. Address region Configuration | 22 |
| 4. 3. 3. Firmware | 24 |
| 4. 3. 4. Processor RAM | 24 |
| 4. 3. 5. The Interrupt Scheme | 24 |
| 4. 3. 6. Y-Bus and F-Bus | 25 |
| 4. 4. VME Bus Interface | 26 |

| | |
|--|----|
| 4. 4. 1. The Arbitration Scheme | 26 |
| 4. 4. 2. Reset Control | 27 |
| 4. 4. 3. SGU Reset | 27 |
| 4. 4. 4. Go Command | 28 |
| 4. 4. 5. VME Bus Control Register | 28 |
| 4. 4. 6. VME Interrupt Request | 29 |
| 4. 4. 7. VME Interrupt Vector Register | 30 |
| 4. 4. 8. Configuration Register | 30 |
| 4. 4. 9. VME bus port | 30 |
| 4. 4. 9. 1. Master Port | 31 |
| 4. 4. 9. 2. Slave Port | 31 |
| 4. 5. The Real Time Unit | 31 |
| 4. 5. 1. Address Generation | 32 |
| 4. 5. 2. Debug Register of Duration Generator and Loop Counter .. | 33 |
| 4. 5. 3. Processor Interrupt XINT0 | 34 |
| 4. 5. 4. State of Trigger Signals | 34 |
| 4. 5. 5. Processor Interrupt XINT1 | 34 |
| 4. 5. 6. Debug Interrupt XINT7 Register | 35 |
| 4. 5. 7. Access to the Real-Time-Program (rtp) entries | 35 |
| 4. 5. 7. 1. Enable of the TCU Output Signals | 36 |
| 4. 5. 7. 2. Meaning and work of the Real-Time-Program (rtp) data bits | 36 |
| 4. 5. 7. 3. Signal and Pin assignment of word 2, 3 and 4 | 41 |
| 4. 6. Device access via external Busses | 44 |
| 4. 6. 1. JTAG access | 44 |
| 4. 6. 2. BBIS access | 44 |

Figures

| | |
|---|----|
| Figure 1: Wiring Diagram | 2 |
| Figure 2: Controls at Front Panel | 6 |
| Figure 3: Front View at LED Display | 7 |
| Figure 4: TCU3 Jumper, Prom and RAM Locations | 8 |
| Figure 5: ST2: VME J0-Connector, 90 pin Mini Metral, female ... | 11 |
| Figure 6: ST8/ST10 JTAG Connectors | 12 |
| Figure 7: Block diagram of TCU3 | 18 |
| Figure 8: Interrupt Vector Register | 30 |
| Figure 9: Address Generator | 33 |

Tables

| | |
|---|---|
| Table 1: TCU versions | 4 |
| Table 2: Table of Assembly Groups | 6 |

| | |
|--|----|
| Table 3: Part# of Accessories | 7 |
| Table 4: SRAM configuration | 11 |
| Table 5: Prog Pal, (H5815) | 14 |
| Table 6: Prog File_990113EC00 (H5816) | 14 |
| Table 7: Prog File_990122EC01 (H5816) | 15 |
| Table 8: Prog File_990122EC20 (H5816) | 15 |
| Table 9: Memory Map and Device Codes | 19 |
| Table 10: Address region assignment | 22 |
| Table 11: Address Region Configuration | 23 |
| Table 12: Binary Format of Interrupt Control Register | 35 |
| Table 13: Real-Time-Program Addresses | 36 |
| Table 14: Format of Word 1 | 36 |
| Table 15: Binary Format of the NMI control instruction in w1 | 38 |
| Table 16: Binary Format of WAIT control instruction in w1 | 39 |
| Table 17: Binary Format of LOOP control instruction in w1 | 39 |
| Table 18: Format of Word 2 | 39 |
| Table 19: Format of Word 3 | 40 |
| Table 20: Format of Word 4 | 40 |

1. Starting with TCU3

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 TCU3

The TCU3 is a faster and more comprehensive derivative version of TCU1 + TCU1 Extension.

- The TCU3 accepts 20 MHz ECL input clock from the Reference Unit and provides 40 MHz TTL output clock to RCU and GCU
- The acquisition program on TCU3 is able to access the whole VME address range
- TCU3 provides the F-Bus (FIFO-Bus) to all FCU3 to transfer the F-Bus commands in an undisturbed environment.
- TCU3 outputs 67 NMR signals, time resolution 50 nsec
- The Real Time Program unit can respond to 4 external trigger signals

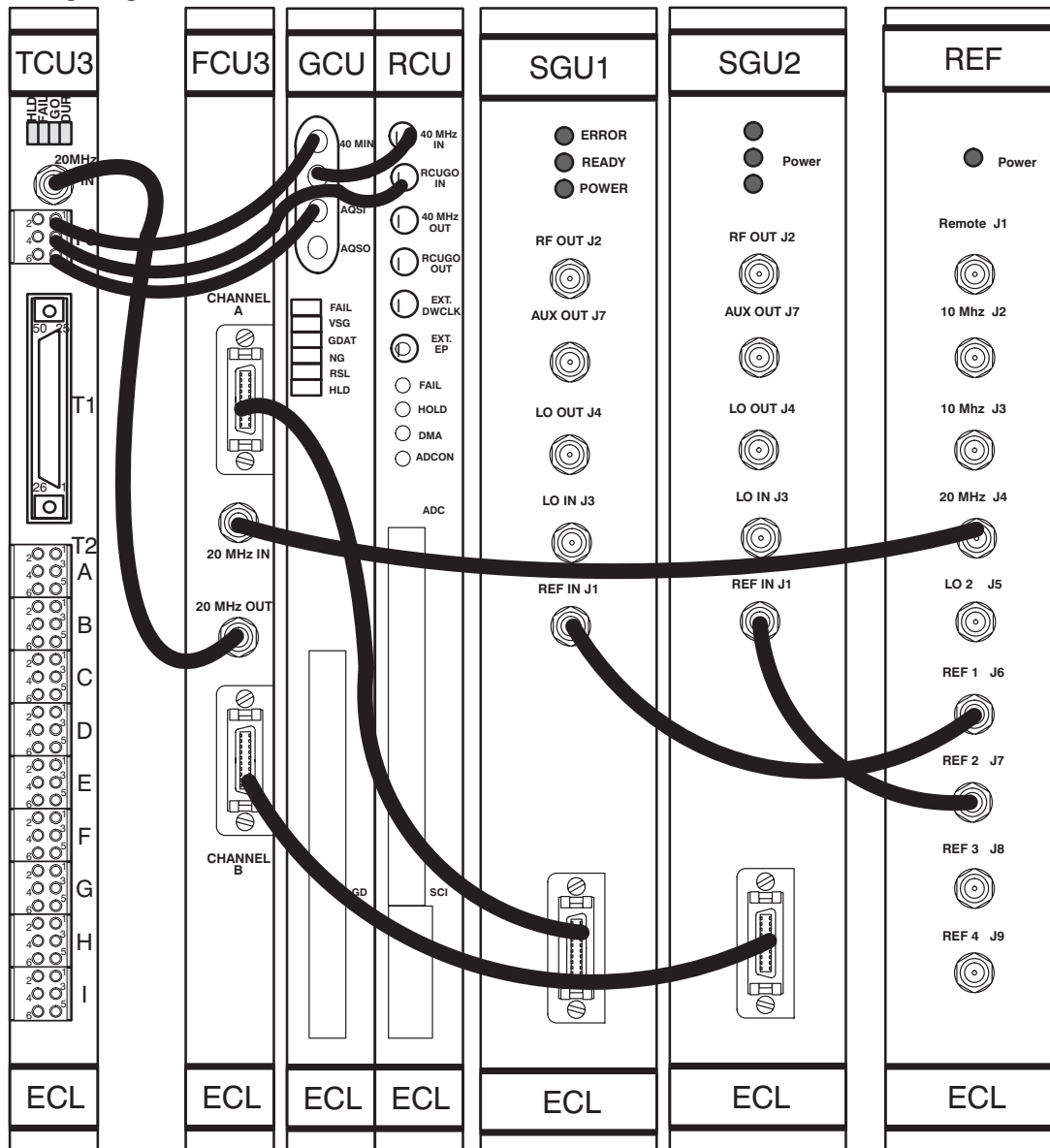
1. 2. Hardware Implementation

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

Wiring

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.

Figure 1: Wiring Diagram



1. 3. Software Implementation

Versions

| | |
|----------|---|
| XWIN-NMR | The XWIN-NMR version which is able to work with TCU3/FCU3 is expected to be called XWIN-NMR 3.0 |
| diskless | version 19990101, including TCU test 19990125 |
| TCU test | version 19990125 |

TCU Test

| | |
|------|---|
| Path | /usr/diskless/clients/spect/root/u/sys-test/tcu |
|------|---|

The `tcutest` recognizes the TCU version (TCU0/1 or TCU3) on which it is requested to run and activates the correct program version.

See the AQS Test Manual for a detailed description of test programs

1. 4. Modifications

Feature "SGU Reset"

Implementing this feature enables the software to stop all SGU activities and set the SGU in an inactive state. It needs the following modifications:

- TCU3: EC01 or higher
- Back panel "AQS VME BUS 8 Slot Rev.02" or Rev.01 with this modification: Wired connection J0 pin E3 to X3 pin 8

J0 is the connector in middle position of any Slot. E3 is its third pin from top in the right row, seen in plug in direction from front side.

X3 is the 20-pin connector to SGU-back panel. Pin 8 is the fourth pin from top of the left row, seen also from front side.

2. Specifications

2.1. TCU Versions

| Versions | Board | Part No. | Layout No. | EC Level | Constrains | Constrains |
|----------|----------|----------|------------|--------------|----------------|-------------------------|
| | | | | | Hardw. | Softw. |
| TCU0 | TCU main | H2558 | H3P1860B | $EC \geq 01$ | AQX | |
| | TCU ext | H2562 | H3P2020A | $EC < 20$ | | |
| | TCU ext | H2562 | H3P2020B | $EC \geq 20$ | | |
| TCU1 | TCU main | H5811/12 | H3P1860E | $EC \geq 00$ | AQX | XWIN–NMR Version 1.0 |
| | TCU ext | H2562 | H3P2020A | $EC < 20$ | | |
| | TCU ext | H2562 | H3P2020B | $EC \geq 20$ | | |
| TCU2 | TCU main | H5813 | H3P2200 | $EC \geq 00$ | not introduced | |
| | TCU ext | H2562 | H3P2020A | $EC < 20$ | | |
| | TCU ext | H2562 | H3P2020B | $EC \geq 20$ | | |
| TCU3 | TCU | H5813 | H3P2310A | $EC \geq 00$ | AQS | XWIN–NMR Version 3.0 |

Table 1: TCU versions

2.2. Features

- 128 KByte Real–Time–Program RAM, organized as 128 bit wide and 8k deep ring buffer, read out by the Acquisition Controller at 20 MHz and dynamically filled by the i960 processor.
- 67 real time outputs, controlled by the read out sequence of the Acquisition Controller.
They can switch in each system cycle (20 Mhz) with a common delay in each cycle of 0, 12.5, 25 or 37.5 nsec.
- That means a Resolution of 12,5 nsec
- Minimal duration is 50 nsec
- Maximal duration is 1,6777216375 sec
- There are 4 trigger inputs, edge or level sensitive
- I80960HX Microcontroller with internal 16KB instruction cache, 8KB data cache and 2KB Data Ram operating external at 25MHZ without wait states and on chip at 75 MHz.
- Fast local instruction and data RAM of 2 MByte operating with 0 wait states at read and 1 wait state at write and pipeline mode
- 32 Bit VME Bus Master/Slave Interface
- 32–bit fast command bus (F–Bus) to FCU’s and GCU

- 16-bit fast real time bus (AQ-Bus) to FCU's, RCU and GCU
- 8-bit control bus (Y-Bus) to RCU and GCU
- Bruker Identification System EEPROM, BBIS

2.3. Construction

The TCU is a VME Bus module of 4 TE with extended length. It consists of one printed circuit board.

Board Size

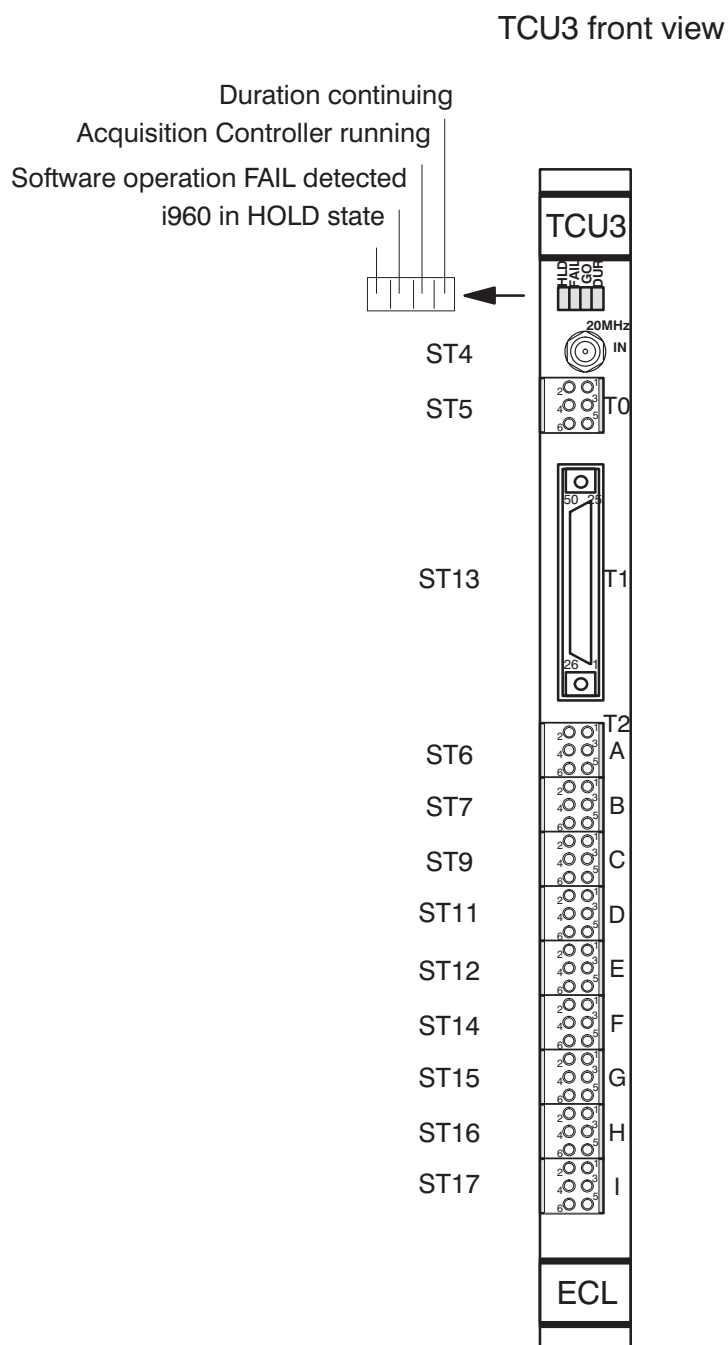
The real size is 233.35 mm by 280 mm . This is the so called "Double European Standard" format with a nominal plug in depth of 280 mm.

Extended Acquisition Bus System

It is designed for a new AQS System Rack. The Extended Acquisition Bus requires the high density minimetral 90-pin VME J0 connector. It includes the F-Bus, AQ-Bus, Y-Bus, JTAG Bus, the BBIS interface and the Slots identification scheme.

Front View

Figure 2: Controls at Front Panel



2. 4. Part numbers of TCU3

Table 2: Table of Assembly Groups

| Amount | Title | Function | Part-Nr. |
|--------|--------------|-------------------|----------|
| 1 | AQS_TCU3 | Assembled PCB | H5813 |
| 1 | PCB | Layout H3P2310A/B | H5814 |
| 1 | TCU3 PAL set | Prog PALs | H5815 |

| Amount | Title | Function | Part-Nr. |
|--------|------------------------|--------------|----------|
| 1 | TCU3 JTAG program file | Prog File | H5816 |
| 1 | Front-Panel | BICC Vero | H206063 |
| 1 | Front-Panel | Hf protected | H206235 |
| 1 | Front-Panel-Ident | | |

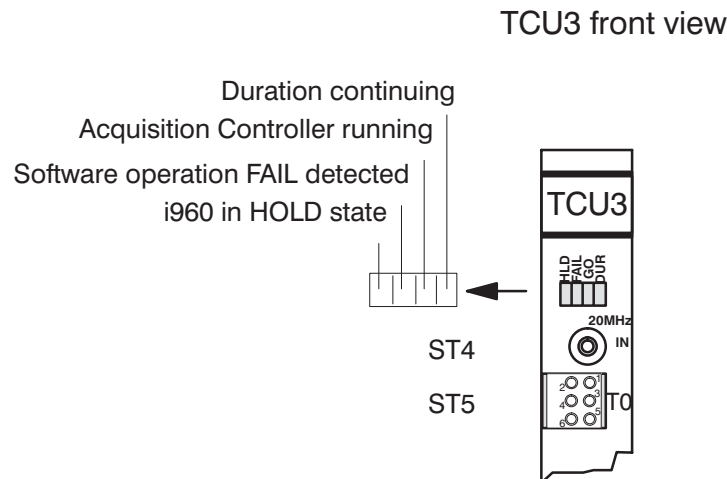
2. 5. Accessories

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 | H203804/A |
| Cable coax 400mm SMA/SMA | H210105/A |
| Cable 6P TCU T0 connector | H210104/B |
| SGU Interface Cable | Z13928 |

2. 6. Controls and Indicators

Figure 3: Front View at LED Display



Activity LED's

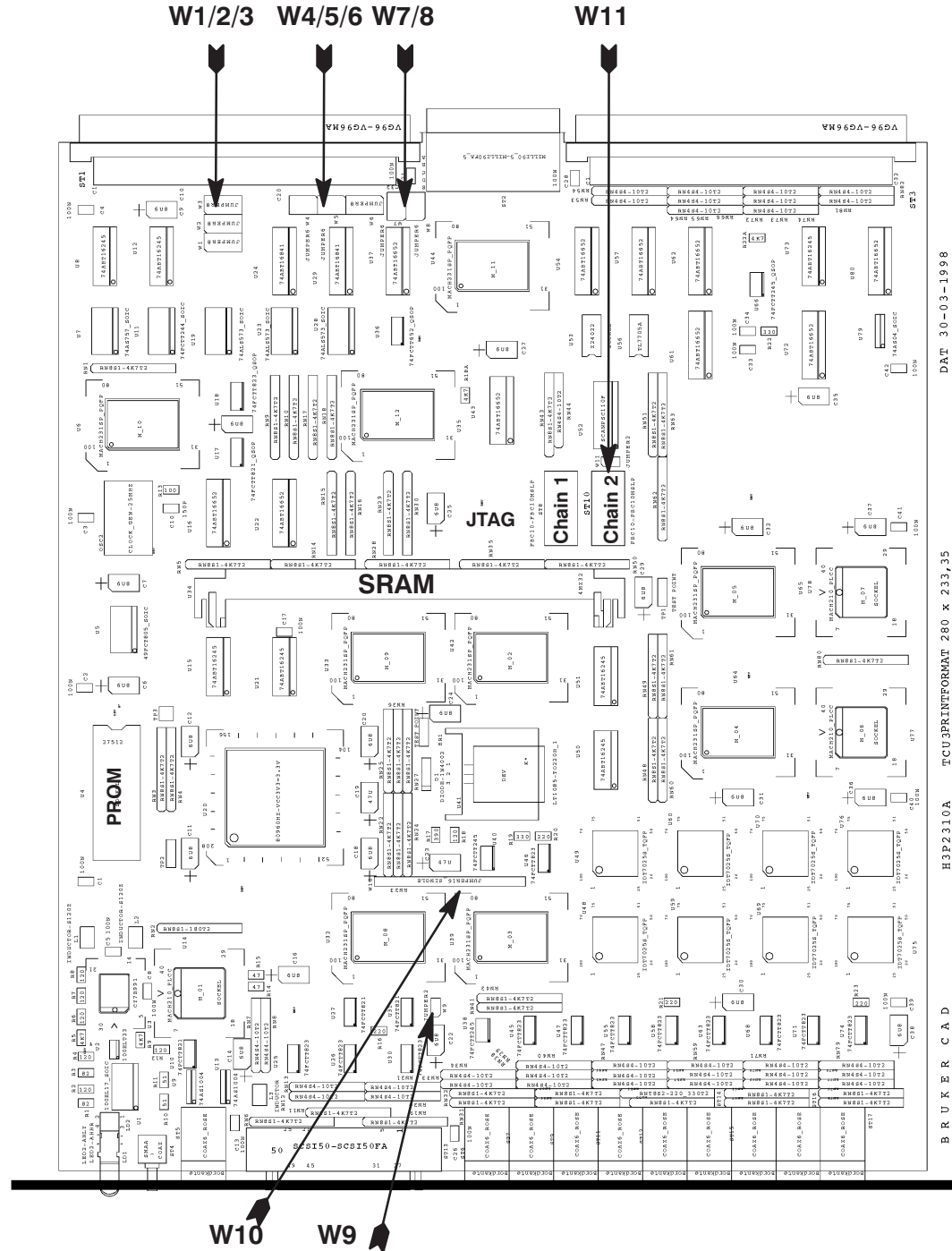
- The HOLD LED, when ON, indicates that the i960 processor has relinquished the internal bus and the i960 processor is in its idle state. The HOLD LED OFF indicates that the processor is in running state for normal operation.
- The FAIL LED, if ON, signals errors in the Self Test phase performed during initialization. The Self-Test is passed, when the FAIL LED remains OFF.
- The Acquisition Controller running LED ON indicates that the Acquisition Unit is running in loop.

- The Duration continuing LED is ON when any duration interval is in progress.

After power-up or software reset a basic procedure set the i960 processor in a HOLD state (only the HOLD LED is on and all other LED's are off).

2. 7. Operational Settings

Figure 4: TCU3 Jumper, Prom and RAM Locations



2. 7. 1. Firmware

Firmware version

- The original installed and labeled firmware EPROM is "TCU-09.06.98"


Installed type of firmware Prom's

- It is 512 KByte dual in line EPROM 8 bit wide.
- Access time is required to be 120 nsec or less.

2. 8. Jumper

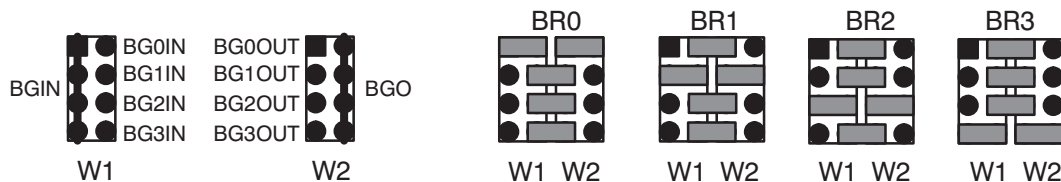
VME Bus Master Request (Jumper W3 on H5813)

The VME Bus Master request line is selected by these jumpers. Only one jumper should be set.

| W3 | | 7-8 | 5-6 | 3-4 | 1-2 | Bus Request Level |
|---|--|-----|-----|-----|-----|--------------------|
|  | | IN | IN | IN | IN | X unused not valid |
| | | OUT | OUT | OUT | IN | BR0 |
| | | OUT | OUT | IN | OUT | BR1 |
| | | OUT | IN | OUT | OUT | BR2 |
| | | IN | OUT | OUT | OUT | BR3 |
| | | OUT | OUT | OUT | OUT | X unused not valid |

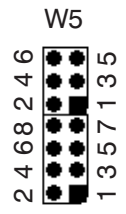
VME Bus Grant In (Jumper W1 and W2 on H5813)

This jumper is used to set the VME Bus Grant Level decoding to the Bus Request Level adjusted with W3.



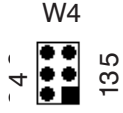
VME Interrupt Request (Jumper W6 and W5 on H5813)

The VME Interrupt request line is selected by these jumpers. Only one jumper should be set.

| W5 | | W6 | W5 | IRQ |
|---|--|-----|-----|-----|
|  | | 1-2 | --- | 1 |
| | | 3-4 | --- | 2 |
| | | 5-6 | --- | 3 |
| | | 7-8 | --- | 4 |
| | | --- | 1-2 | 5 |
| | | --- | 3-4 | 6 |
| | | --- | 5-6 | 7 |

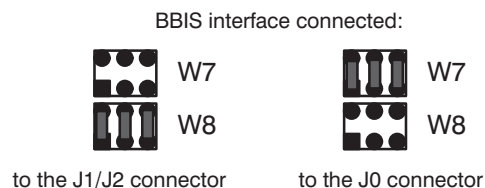
VME Interrupt Level (Jumper W4 on H5813)

This jumper is used to set the interrupt level decoding to the appropriate VME Interrupt Request Level adjusted with W6 and W5.

| W4 | 5-6 | 3-4 | 1-2 | INT. LEVEL |
|---|-----|-----|-----|----------------------------|
|  | IN | IN | IN | X unused not a valid conf. |
| | IN | IN | OUT | 1 |
| | IN | OUT | IN | 2 |
| | IN | OUT | OUT | 3 |
| | OUT | IN | IN | 4 |
| | OUT | IN | OUT | 5 |
| | OUT | OUT | IN | 6 |
| | OUT | OUT | OUT | 7 |


BBIS interface selection (Jumper W7, W8)

The BBIS interfaces can be connected by this Jumper to the VME Bus J1/J2 or to the J0 connector.



JTAG Bridge Enable (Jumper W11)

The JTAG Bridge on board can be disabled with this jumper.

| W11 | 1-2 | JTAG Bridge Enable |
|---|-----|---|
|  | OUT | JTAG Bridge disabled, programming via local connectors ST8 and ST10 |
| | IN | JTAG Bridge enabled, programming via backplane |

- JTAG Local Port ST8 is used to programming JTAG Chain 1
- JTAG Local Port ST10 is used to programming JTAG Chain 2

2. 9. Installing SRAM

- The TCU3 supports up to 2 MByte SRAM installed as Module of 20 ns access time.

- Alternatively it is possible to insert a 72-pin socket (P/N 65661) and to use a SIMM SRAM module

Table 4: SRAM configuration

| Inserted as | TCU3 | |
|-------------|-----------|--|
| | SRAM Size | Module Equipment |
| U34 | 2MByte | ZIP SRAM Modul P/N 69823 |
| U34 | 2MByte | SIMM SRAM Modul, 72-pin socket P/N 65661 |

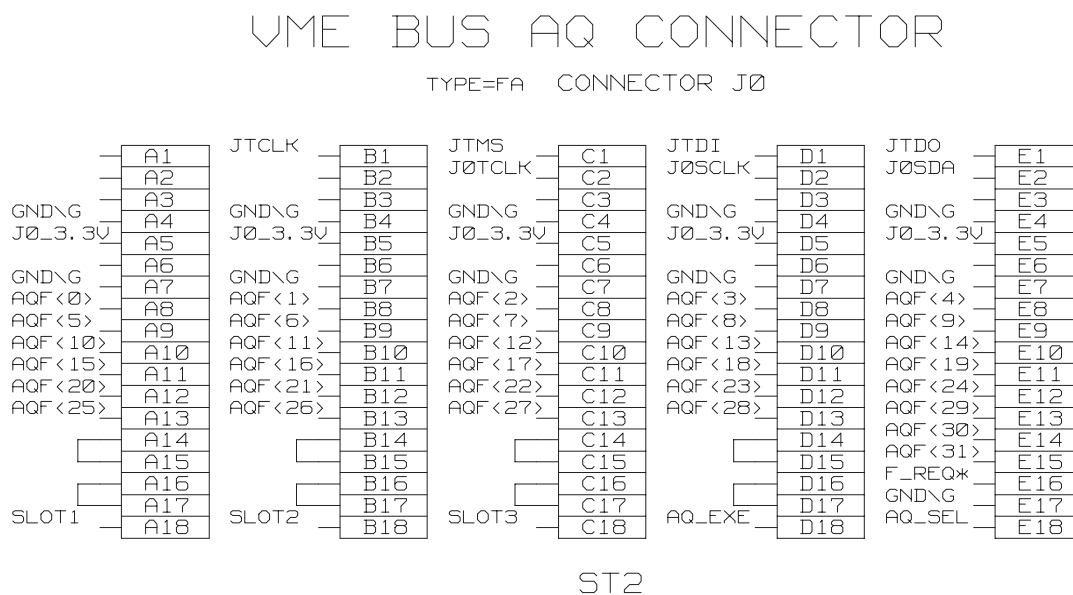
2. 10. Connectors and Signal Allocations

NMR Connectors T0, T1, T2

Refer to the section "Signal and Pin assignment of NMR words 2, 3 and 4"

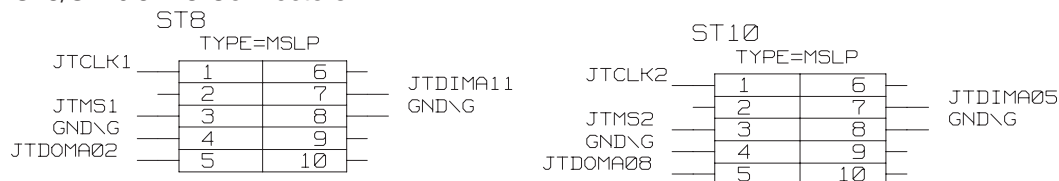
ST2 AQ Extended J0-Connector

Figure 5: ST2: VME J0-Connector, 90 pin Mini Metral, female



ST8/ST10 JTAG Connectors

Figure 6: ST8/ST10 JTAG Connectors



JTAG Chain Structure

```

Chain 1   TDI   TDO
TCU3MA11 JTDIMA11 JTDOMA11
TCU3MA12 JTDOMA11 JTDOMA12
TCU3MA10 JTDOMA12 JTDOMA10
TCU3MA09 JTDOMA10 JTDOMA09
TCU3MA02 JTDOMA09 JTDOMA02

Chain 2   TDI   TDO
TCU3MA05 JTDIMA05 JTDOMA05
TCU3MA04 JTDOMA05 JTDOMA04
TCU3MA03 JTDOMA04 JTDOMA03
TCU3MA08 JTDOMA03 JTDOMA08

```

2. 11. Power Requirements

The TCU3 requires power supply of the following voltages:

| | Part-No. | +5 V | +12 V | -12 V | +3,3V J0: A5, B5, C5, D5, E5 |
|------|----------|------|-------|-------|------------------------------------|
| TCU3 | H5813 | 4,7A | | | |

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 TCU versions this TCU3 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

TCU3 of the Introduction Status show no functional problems up to January 99.

Prior to Introduction delivered TCU3 of prototype status had been updated to the introduced one of EC00.

Feature "SGU Reset"

Implementing this feature enables the software to stop all SGU activities and set the SGU in an inactive state. It needs the following modifications:

- TCU3: EC01 or higher
- Back panel "AQS VME BUS 8 Slot Rev.02" or Rev.01 with this modification: Wired connection J0 pin E3 to X3 pin 8

J0 is the connector in middle position of any Slot. E3 is its third pin from top in the right row, seen in plug in direction from front side.

X3 is the 20-pin connector to SGU-back panel. Pin 8 is the fourth pin from top of the left row, seen also from front side.

3. 3. Introduction Status

3. 3. 1. Prog Pal

Table 5: Prog Pal, (H5815)

| TCU3 Prog_PAL H5815, EC00 | | | | | | |
|---------------------------|----------|---------------------|-------|------------|---------|-------|
| Programmed Device | | | | Device | | |
| IC# | Checksum | PAL Name | H-Nr. | Type | Package | Part# |
| U14 | 8f5b | TCU3MA01 | H9580 | MACH211-7 | PLCC 44 | 68089 |
| U77 | 6892 | TCU3MA06 | H9581 | MACH210-7 | PLCC 44 | 66401 |
| U78 | d9c9 | TCU3MA07 | H9582 | MACH210-7 | PLCC 44 | 66401 |
| U4 | | EPROM960-9 81001 | H9583 | 27C512-120 | DIL 28 | 14568 |
| U53 | | (BBIS) | H9584 | X24022 | DIL 8 | 65415 |

3. 3. 2. Prog File

Table 6: Prog File_990113EC00 (H5816)

| TCU3 Prog_File H5816 : TCU3_990113EC00 | | | | | |
|--|----------|----------|--------------|----------|-------|
| Programmed Device | | | Device | | |
| IC# | Checksum | PAL Name | Type | Package | Part# |
| U42 | 4306 | TCU3MA02 | MACH231SP-10 | PQFP 100 | 67858 |
| U39 | ee54 | TCU3MA03 | MACH231SP-10 | PQFP 100 | 67858 |
| U64 | 531d | TCU3MA04 | MACH231SP-10 | PQFP 100 | 67858 |
| U65 | adc8 | TCU3MA05 | MACH231SP-10 | PQFP 100 | 67858 |
| U32 | d684 | TCU3MA08 | MACH231SP-10 | PQFP 100 | 67858 |
| U33 | e2c8 | TCU3MA09 | MACH231SP-10 | PQFP 100 | 67858 |
| U6 | d9b9 | TCU3MA10 | MACH231SP-10 | PQFP 100 | 67858 |
| U44 | 0b4e | TCU3MA11 | MACH231SP-10 | PQFP 100 | 67858 |
| U35 | 9edf | TCU3MA12 | MACH231SP-10 | PQFP 100 | 67858 |

At introduction of the layout version H3P2310B with EC20 this Prog File will be replaced by TCU3_990122EC20. TCU3MA09 will become obsolete, as U42 TCU3MA02 will be replaced by TCU3MA22.

Table 7: Prog File_990122EC01 (H5816)

| TCU3 Prog_File H5816 : TCU3_990122EC01 | | | | | |
|--|----------|----------|--------------|----------|-------|
| Programmed Device | | | Device | | |
| IC# | Checksum | PAL Name | Type | Package | Part# |
| U42 | 4306 | TCU3MA02 | MACH231SP-10 | PQFP 100 | 67858 |
| U39 | ee54 | TCU3MA03 | MACH231SP-10 | PQFP 100 | 67858 |
| U64 | 531d | TCU3MA04 | MACH231SP-10 | PQFP 100 | 67858 |
| U65 | 7374 | TCU3MB05 | MACH231SP-10 | PQFP 100 | 67858 |
| U32 | d684 | TCU3MA08 | MACH231SP-10 | PQFP 100 | 67858 |
| U33 | e2c8 | TCU3MA09 | MACH231SP-10 | PQFP 100 | 67858 |
| U6 | d9b9 | TCU3MA10 | MACH231SP-10 | PQFP 100 | 67858 |
| U44 | 0b4e | TCU3MA11 | MACH231SP-10 | PQFP 100 | 67858 |
| U35 | 9edf | TCU3MA12 | MACH231SP-10 | PQFP 100 | 67858 |

Table 8: Prog File_990122EC20 (H5816)

| TCU3 Prog_File H5816 : TCU3_990122EC20 | | | | | |
|--|----------|----------|--------------|----------|-------|
| Programmed Device | | | Device | | |
| IC# | Checksum | PAL Name | Type | Package | Part# |
| U42 | 2c88 | TCU3MA22 | MACH231SP-10 | PQFP 100 | 67858 |
| U39 | ee54 | TCU3MA03 | MACH231SP-10 | PQFP 100 | 67858 |
| U64 | 531d | TCU3MA04 | MACH231SP-10 | PQFP 100 | 67858 |
| U65 | 7374 | TCU3MB05 | MACH231SP-10 | PQFP 100 | 67858 |
| U32 | d684 | TCU3MA08 | MACH231SP-10 | PQFP 100 | 67858 |
| | | | | | |
| U6 | d9b9 | TCU3MA10 | MACH231SP-10 | PQFP 100 | 67858 |
| U44 | 0b4e | TCU3MA11 | MACH231SP-10 | PQFP 100 | 67858 |
| U35 | 9edf | TCU3MA12 | MACH231SP-10 | PQFP 100 | 67858 |

3. 3. 3. Modifications of the introduced layout H3P2310A

On the introduced Layout version H3P2310A and EC00 the following modifications are required:

- Prog File TCU3_990113EC20EC00
- Connect U6 pin 12 to U24 pin 1 (V_TAKE* signal)
- Bend up U6 pin 38
- Insert a 4k7 resistor between C35 pin 1 and U 61 pin 3
- Connect U14 pin 39 to U57 pin 19 (AQSTRI signal)
- Connect U57 pin 1 to U57 pin 3 (LONE10 signal)
- Following components must not be mounted: CB6A, RB11, RB22
- Following resistor are placed with value of 4k7: R18A, R22A, RB35A, RB38A, RB38B
- Connect W8 pin 1 to W7 pin 2 (TCLK signal)
- Connect W8 pin 3 to W7 pin 4 (SCLK signal)
- Connect W8 pin 5 to W7 pin 6 (SDA signal)

3. 3. 4. Modifications of the introduced layout H3P2310B

SGU Reset

Implementing the feature "SGU Reset" needs this modification on layout H3P2310B

- Prog File TCU3_990122EC20
- Connect U65 pin 11 to U65 pin 38
- Connect U65 pin 38 to J0 (St2) pin E3

Implementing failed pull-up resistor on a NMR4_8 signal

- Connect ST13 pin 35 to RN11 pin 9

3. 3. 5. Jumper Setting

Normal setting of installed jumpers:

Jumper W7,W8 on H5813: not set on H3P2310A Layout version

Jumper W9 on 5813

Dual-Port Size Enable



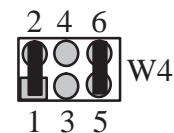
Jumper W11 on 5813

JATAG Bridge Enable



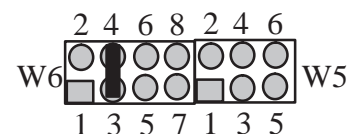
Jumper W4 on 5813

Interrupt Vector 0x2



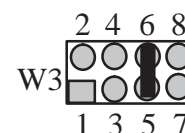
Jumper W6 and W5 on H5813

Interrupt request on IRQ2

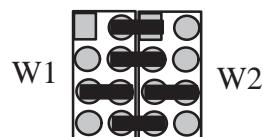


Jumper W3 on H5813

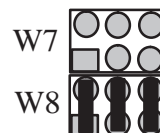
VME Bus request BRQ2



Jumper W1 and W2 on H5813
VME Bus BG2IN and BG2OUT



Jumper W7 and W8 on H5813
BBIS connected to J0 on H3P2310B only



3. 3. 6. Firmware

Firmware version

- The originally installed firmware EPROM is labeled "EPROM960-19981001"

Installed type of firmware Prom's

- 512 KByte EPROM, 8 bit wide in dual in line package.
- Access time is required to be 120 nsec or less.

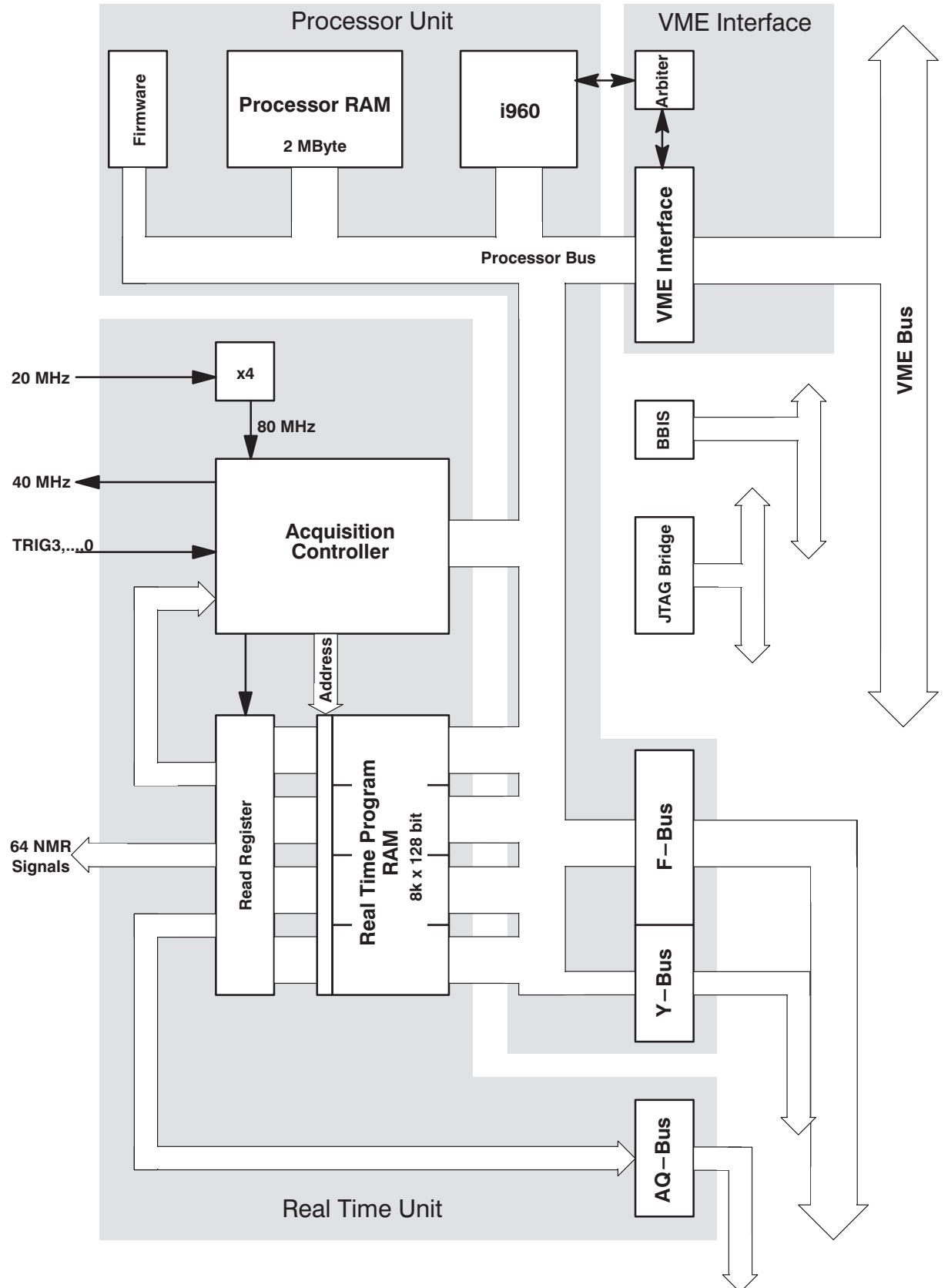
3. 4. History of Modifications

| EC No. | Date | Part Number | Description of Bugs, Changes and Modifications | Ser.No. | New EC-Level |
|--------|---------|-------------|---|--------------|--------------|
| 2539 | 11.1.99 | H5813 | Introduction, layout version A | 0010 | 00 |
| 2546 | 28.1.99 | H5813 | SGU Reset implemented; this needs also the modification described in this EC on back panel "AQS VME BUS 8 Slot Rev. 01", H9578 Pull-up on NMR4_8 signal | | 01 |
| 2547 | 28.1.99 | H5813 | Introduction, layout version B | 0045 | 20 |
| 2559 | 16.3.99 | H5813 | Inserting 2 missed Pull-up resistors at EXT_DW and EXT_RGP | 0010 0053 | 02 21 |
| 2577 | 3.5.99 | H5813 | Solve termination and DMA Problems New Prog Files TCU3_990528EC03 and TCU3_990528EC22 | 0010 0053 | 03 22 |
| 2599 | 8.7.99 | H5813 | Reset connection to JTAG controller of i960 | 0010 0053 | 04 23 |
| 2616 | 17.8.99 | H5813 | Softreset does no longer switch the TCU output signals to the inactive state | 0010 0053 | 05 24 |

4. Condensed Description

4. 1. Architecture

Figure 7: Block diagram of TCU3



The TCU can be considered to comprise the following functional units:

- Processor Unit
- Real Time Unit
- VME Bus Interface

4. 2. Summary of Logical References to memory and I/O devices

Table 9: Memory Map and Device Codes

| VME Address | Local Address | Action | Mode R/W | Oprd. Byte 3 2 1 0 | Only TCUi |
|-----------------------------|-----------------------------|--|----------|--------------------|-----------|
| 1A000000 to 1A1FFFFC | 10000000 to 101FFFFC | Instruction and Data RAM of "i960", 2M byte | R/W | b b b b | 3 |
| no access | 80000000 to 8FFFFFFC | Region 8 access of i960 to VME address range 0xVxxxxxxx; V=content of the vharg register | R/W | b b b b | 3 |
| 19200000 to 1921FFFFC | E0000000 to E001FFFFC | Real-Time-Program (RTP) RAM 8k words of 4x32 bit | R/W | b b b b | |
| 19220000 | B0000000 | Interrupt Vector of TCU on VME Bus | R/W | x x x b | |
| 19220004 | B0000004 | VME Bus Control Register | R/W | x x x b | |
| 19220008 | B0000008 | Debug and local Interrupt Register of "i960" | R/W | x x x b | |
| 19220010 | no access | Software initiated Hardware Reset of the whole board | W | x x x x | |
| 19220014 | no access | Go-Command to "i960" | W | x x x x | |
| no access | B0000018 | Access to High Address Register vharg; vharg=VME <31..28> | W | x x x b/2 | 3 |
| 19220020 | B0000020 | TCU3 Configuration Register 0 | R | x x x b | 3 |
| 19220024 | B0000024 | TCU3 Configuration Register 1 | R | x x x b | 3 |
| 19220028 | B0000028 | TCU3 Configuration Register 2 | R | x x x b | 3 |
| 1922002C | B000002C | TCU3 Configuration Register 3 | R | x x x b | 3 |
| 19220030 | B0000030 | AQ interrupt | R | x x x b | 3 |
| 19220034 | B0000034 | Software initiated Hardware SGU Reset | W | x x x x | 3 |
| 19221010 | C0000010 | Set Interrupt Control Register1 | W | x x x b | |
| 19221010 | C0000010 | Clear and disable Interrupt 1 | R | x x x x | |
| 19221020 | C0000020 | Read Debugging Signals of duration generator and loop counter | R | x x x b | |
| 19221028 | C0000028 | Read Trigger Inputs | R | x x x b | |
| 19221030 | C0000030 | Clear and disable NMI | R | x x x x | |
| 19221040 | C0000040 | Enable Interrupt 0 | W | x x x x | |
| 19221040 | C0000040 | Clear and disable Interrupt 0 | R | x x x x | |
| 19221050 | C0000050 | Clear and disable WAIT | R | x x x x | |

| VME Address | Local Address | Action | Mode R/W | Oprd. Byte 3 2 1 0 | Only TCUi |
|-------------|---------------|--|----------|--------------------|-----------|
| 19221060 | C0000060 | Set HMD1 Register | W | x x x b | 0/1 |
| 19221068 | C0000068 | Set HMD2 Register | W | x x x b | 0/1 |
| 19221080 | C0000080 | RUN Start of Clock Generator (run of Address Generator) | W | x x x x | |
| 19221084 | C0000084 | STEP Stops Clock Generator, steps to next edge, (The effect of each STEP is the next clock edge, high>low>high>low) | W | x x x x | |
| 19221088 | C0000088 | LDREG Preload of Start Register, (Clock Generator keeps running) | W | x x b b | |
| 1922108C | C000008C | STOP Stop of Clock Generator, (clock at "high") | W | x x x x | |
| 19221090 | C0000090 | START Load Address Gen. through Start Reg. with operand and Start Clock Generator | W | x x b b | |
| 19221094 | C0000094 | Step Address Generator by one if Clock Gen. off | W | x x x x | |
| 19221098 | C0000098 | LDADDR Load Address Gen. through Start Reg. with operand if Clock Generator is off, (Clock Generator stays off) | W | x x b b | |
| 1922109C | C000009C | DEVST Load Address Generator with value of Start Register (During RUN this address becomes valid after the actual duration) | W | x x x x | |
| 192210C0 | C00000C0 | RDADDR Read of Address Generator Register | W | x x b b | |
| 19221100 | C0000100 | INIT Stop of Clock Generator and Clear all registers of Address Generator | W | x x x x | |
| 19221104 | C0000104 | Read NMR1 – Preregister into NMR – Reg. | R | x x b b | 0/1 |
| 19221108 | C0000108 | Read NMR2 – Preregister into NMR – Reg. | R | x x b b | 0/1 |
| 1922110C | C000010C | Read NMR3 – Preregister into NMR – Reg. | R | x x b b | 0/1 |
| 19221110 | C0000110 | Read NMR4 – Preregister into NMR – Reg. | R | x x x b | 0/1 |
| 19221114 | C0000114 | Read NMR5 – Preregister into NMR – Reg. | R | x x b b | 0/1 |
| 19221118 | C0000118 | Read NMR6 – Preregister into NMR – Reg. | R | x x x b | 0/1 |
| 1922111C | C000011C | Read NMR7 – Preregister into NMR – Reg. | R | x x b b | 0/1 |

| VME Address | Local Address | Action | Mode R/W | Oprd. Byte 3 2 1 0 | Only TCUi |
|----------------------------|----------------------------|--|----------|--------------------|-----------|
| 19221120 | C0000120 | Read NMR8 – Preregister into NMR – Reg. | R | x x b b | 0/1 |
| 19221124 | C0000124 | Read NMR1 – Register into NMR – Reg. | R | x x b b | 0/1 |
| 19221128 | C0000128 | Read NMR2 – Register into NMR – Reg. | R | x x b b | 0/1 |
| 1922112C | C000012C | Read NMR3 – Register into NMR – Reg. | R | x x b b | 0/1 |
| 19221130 | C0000130 | Read NMR4 – Register into NMR – Reg. | R | x x x b | 0/1 |
| 19221134 | C0000134 | Read NMR5 – Register into NMR – Reg. | R | x x b b | 0/1 |
| 19221138 | C0000138 | Read NMR6 – Register into NMR – Reg. | R | x x x b | 0/1 |
| 1922113C | C000013C | Read NMR7 – Register into NMR – Reg. | R | x x b b | 0/1 |
| 19221140 | C0000140 | Read NMR8 – Register into NMR – Reg. | R | x x b b | 0/1 |
| 19221150 | C0000150 | Read BLKTRA(8..15) into NMR – Reg. B1 Read BLKTR(1,2,5,6) and BLK – GD(X,Y,Z) into NMR – Reg. Byte0 | R | x x b b | 0/1 |
| 19221154 | C0000154 | Read BLKTRA(1..8) into NMR – Reg. B1 Read BLKTRC(1..8) into NMR – Reg. Byte0 | R | x x b b | 0/1 |
| 19221158 | C0000158 | Read BLK_F(1..4) and SP_F(1..4) into NMR – Reg. Byte0 | R | x x x b | 0/1 |
| 1922115C | C000015C | Read SP_PA(1..4) and SP_FA(1..4) and OBSCH(1..4) and TUNE,RCPO into NMR. | R | x x b b | 0/1 |
| 19221200 | C0000200 | NMR Output Enable on, sets the TCU Output Signals active | W | x x x x | |
| 19221210 | C0000210 | NMR Output Enable off, sets the TCU Output Signals inactive | R | x x x x | |
| 19221300 | C0000300 | Assert Interrupt to VME – Bus, Level defined by Jumper Wx | W | x x x x | |
| 19221310 | C0000310 | Read Y – Bus Data back | R | x x x b | |
| 192213F0 | C00003F0 | Read NMR – Register into port 4 | R | b b b b | 0/1 |
| 192213F8 | C00003F8 | Read NMR – Buffer into port 4 | R | b b b b | 0/1 |
| 192213E0 | C00003E0 | Read AQ – Bus into port 4 Read SPP – Register into port 3 | R | b b b b | 0/1 |
| 19221400 to 192217FC | C0000400 to C00007FC | Read or write 8 bit data from or to Y – Bus | R | x x x b | |
| 19222400 to 192227FC | D0000400 to D00007FC | Read or write 32 bit data from or to F – Bus | R/W | b b b b | 3 |

4. 3. i960 Processor Unit

The processor unit includes:

- The i960
- Some processor dedicated parts like EPROM and Initial Boot Record

- Processor defined address mapping
- Address region configuration
- Interface to F– and Y–Bus

i960

An important part of the Timing Control Unit is the Intel i960 Microprocessor operating external at 25MHZ with 2MB fast (20ns) SRAM and 64KB EPROM. The operation mode (i.e. Bus configuration, access speed, ready timing) of the i960 bus controller is programmable. For a detailed description see chapter "Memory region configuration". The i960 has an on chip interrupt controller with 8 dedicated interrupt pins and a separate NMI input. The interrupts are detected at low level.

The i960 Processor has an linear address space from 0 to $2^{32}-1$. Some parts of this address space are reserved or assigned to special functions. The valid operand format in device accesses depend on their region configuration.

4. 3. 1. i960 defined address ranges

| Physical Address | Location assigned to |
|-----------------------|--|
| 0000 0000 | NMI VECTOR |
| 0000 0004 – 0000 003F | INTERNAL DATA RAM (OPT. INTERRUPT VECTORS) |
| 0000 0040 – 0000 07FF | INTERNAL DATA RAM (Available for Data) |
| 0000 0800 | Optional Register Cache Frames |
| 0000 0804 – FEFF FF2F | EXTERNAL RAM Space |
| FEFF FF30 – FEFF FF5F | INITIALIZATION BOOT RECORD (EPROM) |
| FEFF FF60 – FEFF FFFF | RESERVED |
| FF00 0000 – FFFF FFFF | INTERNAL REGISTER SPACE |

4. 3. 2. Address region Configuration

The address space of the i960 can be mapped as read–write, read–only memory and memory mapped I/O. The whole memory space is divided into 16 regions each 256 MBytes in range. The upper four bits of the address (A31 – A28) indicate which of the 16 regions is currently selected.

Address region assignment

The regions 0 to 7 are reserved for the inst./data SRAM. The 2KB internal RAM of the i960 is mapped into region 0. The internal RAM of the i960 can only be accessed by loads, stores, or DMA instructions.

The regions 8,9,A are reserved for user definable memory space (i.e. VME DMA I/O). Region B is used for VME I/O devices codes i.e VME interrupt vector register, VME address modifier register, General Control register, and local interrupt register, Configuration Register and VME High Address Register VA<31..28>.

Region C is used for AQ–Bus register the duration– and address generator.

Region D is used for special F–Bus access.

Region E is used for the Real–Time–Program RAM whereas Region F is used for the EPROM boot sequence and for monitor code.

Table 10: Address region assignment

| Address Region | Region# | Assigned Location | Note |
|---------------------|----------|---|------------------------------------|
| 0000 0000–0FFF FFFF | REGION 0 | SRAM_CONF (internal 1KB data RAM) | 32 bit data bus, 25 nsec access |
| 1000 0000–1FFF FFFF | REGION 1 | EXTERNAL 2MB I/D SRAM CONF. | |
| 2000 0000–2FFF FFFF | REGION 2 | SRAM_CONF | |
| 3000 0000–3FFF FFFF | REGION 3 | SRAM_CONF | |
| 4000 0000–4FFF FFFF | REGION 4 | SRAM_CONF | |
| 5000 0000–5FFF FFFF | REGION 5 | SRAM_CONF | |
| 6000 0000–6FFF FFFF | REGION 6 | SRAM_CONF | |
| 7000 0000–7FFF FFFF | REGION 7 | SRAM_CONF | |
| 8000 0000–8FFF FFFF | REGION 8 | VME–BUS Master Access | bus back off |
| 9000 0000–9FFF FFFF | REGION 9 | RESERVED FOR VME–BUS DMA | |
| A000 0000–AFFF FFFF | REGION A | RESERVED FOR VME–BUS DMA | |
| B000 0000–BFFF FFFF | REGION B | VME Control Reg. DEVICE_CODEs | |
| C000 0000–CFFF FFFF | REGION C | ADD. and DURATION GEN., AQ–BUS, LOOP COUNTER | |
| D000 0000–DFFF FFFF | REGION D | F– BUS | |
| E000 0000–EFFF FFFF | REGION E | REAL–TIME–PROGRAM RAM. | |
| F000 0000–FFFF FFFF | REGION F | EPROM_CONF | |

Region Configuration

The purpose of configurable memory regions is to provide system hardware interface support. Each region has independent software programmable parameters which define the data-bus width, ready control, number of wait states, pipeline read mode, byte ordering, parity enable and burst mode. These parameter are stored in the region configuration table.

Because of slow external memory devices the i960 has to be able to generate wait states for any region. Five parameters define the wait-state-generator operation:

| | |
|--------|---|
| NRAD – | Number of wait cycles for Read Address-to-Data (0 to 31) |
| NRDD – | Number of wait cycles for Read Data-to-Data (0 to 3) |
| NWAD – | Number of wait cycles for Write Address-to-Data (0 to 31) |
| NWDD – | Number of wait cycles for Write Data-to-Data (0 to 3) |
| NXDA – | Number of wait cycles for X (read or write) Data-to-Address (0 to 15) |

Table 11: Address Region Configuration

| Parameter | Address Region | | | | | | |
|----------------|----------------|---------------|---------------|----------------|---------------|---------------|---------------|
| | 0,..,7 | 8 | B | C | D | E | F |
| Data–Bus Width | 32 Bit | 32 Bit | 32 Bit | 16 Bit | 32 Bit | 32 Bit | 8 Bit |
| NRAD | 0 wait states | 0 wait states | 3 wait states | 32 wait states | 0 wait states | 2 wait states | 6 wait states |

| Parameter | Address Region | | | | | | |
|------------------|----------------|---------------|---------------|----------------|---------------|---------------|---------------|
| | 0,...,7 | 8 | B | C | D | E | F |
| NRDD | 0 wait states | 0 wait states | 1 wait state | 3 wait states | 0 wait states | 2 wait states | 3 wait states |
| NWAD | 2 wait states | 0 wait states | 3 wait states | 32 wait states | 0 wait states | 2 wait states | 5 wait states |
| NWDD | 2 wait states | 0 wait states | 1 wait states | 3 wait states | 0 wait states | 2 wait states | 3 wait states |
| NXDA | 0 wait states | 0 wait states | 0 wait states | 0 wait states | 0 wait states | 0 wait states | 1 wait states |
| Pipelining | YES | NO | NO | NO | NO | NO | NO |
| External Ready | NO | YES | YES | NO | YES | NO | NO |
| Burst | YES | NO | NO | NO | NO | NO | NO |
| Parity Enable | NO | NO | NO | NO | NO | NO | NO |
| Config.Reg PMCON | 0x11808200 | 0x20800000 | 0x20804343 | 0x0040DFDF | 0x20800000 | 0x00808282 | 0x0001C5C6 |

4. 3. 3. Firmware

The 512 kByte EPROM contains the Initial Boot Record for the i960 and the GNU debugger software.

Access Characteristics

- The EPROM data bus is 8 bit wide and mapped in region 0xF.

4. 3. 4. Processor RAM

In this application the i960 operates with its on board external RAM in pipeline mode with 0 wait state in read and 2 wait state in write cycles. Fast logic is used to generate the SRAM address for pipeline operation.

Access Characteristics

- A memory address is a 32 bit value within 0 and FFFFFFFF Hex. It can be used to reference single byte , 2 bytes , 4 bytes , double word (8 bytes), triple word (12 bytes) or quad words (16 bytes) in memory, depending on the instructions being used.

| VME Address | Local Address | Action | Mode R/W | Oprd. Byte 3 2 1 0 | Only TCUI |
|----------------------------|----------------------------|--|----------|--------------------|-----------|
| 1A000000 to 1A1FFFFC | 10000000 to 101FFFFC | Instruction and Data RAM of "i960", 2M byte | R/W | b b b b | 3 |

4. 3. 5. The Interrupt Scheme

The i960 provides the 7 input pin for external interrupts XINT7,....,XINT0.

Six interrupt sources are direct connected to these inputs. The priority is programmable

by internal means but is assumed to be in reverse order with XINT0 having the highest one.

Interrupt source assignment

| i960 Input | Source |
|------------|---|
| XINT0 | EQUAL error interrupt of the Real Time Unit |
| XINT1 | Trigger input state of TRIG3,...,0 |
| XINT2 | free and pulled up |
| XINT3 | free F– and Y–Bus Interrupt AQI0 |
| XINT4 | Ored F– and Y–Bus Interrupt AQI1 of all FCU's |
| XINT5 | F– and Y–Bus Interrupt AQI2 of RCU |
| XINT6 | F– and Y–Bus Interrupt AQI7 of GCU |
| XINT7 | Software Debug Interrupt |

4. 3. 6. Y–Bus and F–Bus

Y and F–Bus use partially the same physical resources e.g. signal lines. They can only alternatively be used at the same time.

The F–Bus takes advantage of the higher pin count of the new VME J0 connector. It has a fast protocol, separate data and address lines and is 4 bytes wide. The F–Bus has been implemented to provide an unimpeded command channel between TCU and slave devices.

The Y–Bus uses multiplexed data and address lines, as low protocol and is 1 byte wide. It has not been removed to provide compatibility.

Signal assignment

| F–Bus Function | Y–Bus Function | J0 pin count | J2 pin count | Signal |
|----------------|--------------------|--------------|--------------|-------------|
| Address lines | Address/Data lines | 0 | 8 | AQY_0,...,7 |
| none | Address strobe | 0 | 1 | AQY_AS |
| none | Data strobe | 0 | 1 | AQY_DS |
| Write Enable | Write Enable | 0 | 1 | AQY_WR |
| Acknowledge | Acknowledge | 0 | 1 | AQY_ACK |
| Request | none | 1 | 0 | F_REQ |
| Data | none | 32 | 0 | FD_0,...,32 |
| Interrupt | Interrupt | 0 | 8 | AQI_0,...,7 |

Access Characteristics

The only master in Y– and F–Bus accesses is the TCU.

Address range Y–Bus: 256 locations of 1 byte
 F–Bus: 256 locations of 4 byte

i960 address region Y–Bus: Region C
 F–Bus: Region D

Access time Y-Bus: ca. 1200 nsec
 F-Bus: ca. 200 nsec

Device Codes

| VME Address | Local Address | Action | Mode R/W | Oprd. Byte 3 2 1 0 | Only TCUi |
|----------------------------|----------------------------|--|----------|--------------------|-----------|
| 19221400 to 192217FC | C0000400 to C00007FC | Read or write 8 bit data from or to Y-Bus | R | x x x b | |
| 19222400 to 192227FC | D0000400 to D00007FC | Read or write 32 bit data from or to F-Bus | R/W | b b b b | 3 |

Assigned F- and Y-Bus address ranges

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.

| 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.

4. 4. VME Bus Interface

Comprising:

- The VME master and slave ports
- VME Interrupt Vector Register
- VME Bus Control Register
- Debug and Local Interrupt Register of i960
- TCU Configuration Register

The VME Bus Interface lets access its internal registers by the i960 and the VME Bus, the VME Bus by the i960 and the processor bus of the i960 by an external VME master.

4. 4. 1. The Arbitration Scheme

Getting into or through the VME Interface by the i960 or getting into by external VME masters is managed by an arbiter. Getting access to the processor bus is managed by the HOLD logic of the i960.

The Arbiter

As it is on the VME bus the i960 accesses to region B are ready controlled too.

The READY signal to the i960 and the DTACK to the VME bus are controlled by the arbiter. An access from the VME bus is delayed until the READY signal of any pending i960 cycle goes inactive.

In the other case the i960 waits for READY to go active until a pending VME access has been executed.

Both requesters have the same priority.

HOLD Logic

The VME interface asserts a HOLD request if an external VME master wants to access a device which is connected to the processor bus. The i960 is master and does or doesn't grant its bus to the VME interface.

4. 4. 2. Reset Control

The reset logic is used to initialize the TCU circuitry and the i960 processor. Three conditions activate the reset logic:

1. Power up reset
2. VME-Bus System reset
3. Software reset

After RESET the on board control logic is ready to work but the i960 keeps staying in RESET STATE until a GO command is set off by the Host CPU. In RESET STATE the Host CPU can load the TCU instruction RAM with program code or the operation of the TCU can be tested via VME -Bus.

Soft Reset

The Host CPU can reset the TCU via this device code. The i960 processor will enter the reset state and can only be started by the Go command. During the reset state a VME-Bus Master can access the internal instruction RAM, for example to load a program or to test this RAM. In the reset state the front LEDs 'FAIL' and 'HOLD' are switched on.

The active or inactive states of the TCU Output Signals are affected by the Power-up and the VME-Bus System reset but not by the Soft Reset. Likewise doesn't the Soft Reset activate the SGU Reset.

Device Codes

| VME Address | Local Address | Action | Mode R/W | Oprd. Byte 3 2 1 0 | Only TCUI |
|-------------|---------------|--|----------|--------------------|-----------|
| 19220010 | no access | Software initiated Hardware Reset of the whole board | W | x x x x | |

4. 4. 3. SGU Reset

Access to this device code leads to a low pulse (120 nsec) at J0_E3. It is intended to be used reset the SGU into its stop state.

Power-up-, VME-Bus System- and Software Reset have no effect on SGU Reset.

Device Codes

| VME Address | Local Address | Action | Mode R/W | Oprd. Byte 3 2 1 0 | Only TCUi |
|-------------|---------------|---------------------------------------|----------|--------------------|-----------|
| 19220034 | B0000034 | Software initiated Hardware SGU Reset | W | x x x x | 3 |

4. 4. 4. Go Command

This Device Code is used to release the i960 processor from the reset state. Following the i960 will execute the Initial Boot Record located in the PROM and will branch to the entry point of the initially loaded code in RAM.

Device Codes

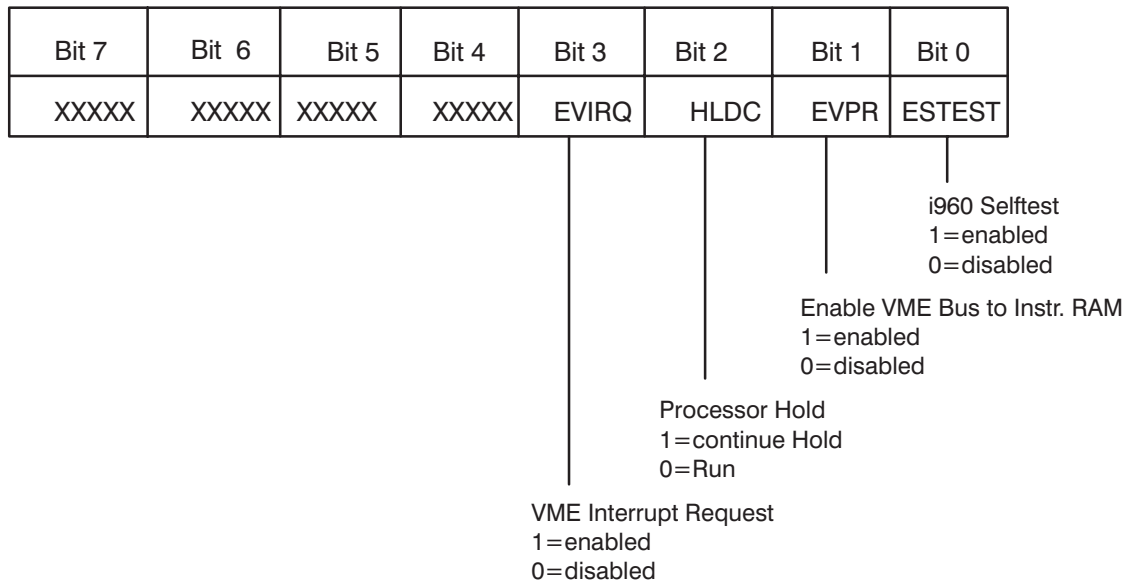
| VME Address | Local Address | Action | Mode R/W | Oprd. Byte 3 2 1 0 | Only TCUi |
|-------------|---------------|----------------------|----------|--------------------|-----------|
| 19220014 | no access | Go-Command to "i960" | W | x x x x | |

4. 4. 5. VME Bus Control Register

The VME Bus Control register is an 8 bit read/writeable register used to control basic operation modes of VME Bus and the i960.

Device code

| VME Address | Local Address | Action | Mode R/W | Oprd. Byte 3 2 1 0 | Only TCUi |
|-------------|---------------|--------------------------|----------|--------------------|-----------|
| 19220004 | B0000004 | VME Bus Control Register | R/W | x x x b | |

Format**Action**

- ESTEST :** If set (1) the internal selftest on the i960 is enabled . The i960 will execute the selftest operation after reset and test its internal register and bus structure and the external Bus for any contention. If the Test passes through without an error the front LED is switched off, otherwise the LED will be on.
- EVPR :** Setting this Bit to 1 allows a VME-Bus Master to access the i960 local instruction RAM without prior setting the i960 in the Hold or Reset State.
The i960 will only be in Hold State for the duration of the VME access cycle.
If set to 0 the local instruction RAM is not accessible by a VME-Bus Master unless the i960 is in reset or Hold State.
- HLDC :** If set to 1 the i960 will go into the continues Hold state. For example to load a new program or initializing the i960 . The i960 will leave the Hold state if this bit is set to 0.
- EVIRQ :** If set to 1 the i960 can generate an interrupt request on the VME-Bus .When set to 0 the interrupt to the VME-Bus is disabled.

4. 4. 6. VME Interrupt Request

The i960 processor or the Host CPU can write to this device code to generate an interrupt request on the VME-Bus. Prior to this action the 'EVIRQ' bit in the General Control register must be set to enable this function.

The VME bus interrupt level can be configured by Jumper.

Device Codes

| VME Address | Local Address | Action | Mode R/W | Oprd. Byte 3 2 1 0 | Only TCUI |
|-------------|---------------|---|----------|--------------------|-----------|
| 19221300 | C0000300 | Assert Interrupt to VME–Bus, Level defined by Jumper Wx | W | x x x x | |

4. 4. 7. VME Interrupt Vector Register

An 8 Bit wide register is used to store the appropriate interrupt vector for an VME–interrupt acknowledge cycle. It should be written before any interrupt action of the TCU is set off.

Figure 8: Interrupt Vector Register

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| VD7 | VD6 | VD5 | VD4 | VD3 | VD2 | VD1 | VD0 |

Device Codes

| VME Address | Local Address | Action | Mode R/W | Oprd. Byte 3 2 1 0 | Only TCUI |
|-------------|---------------|------------------------------------|----------|--------------------|-----------|
| 19220000 | B0000000 | Interrupt Vector of TCU on VME Bus | R/W | x x x b | |

4. 4. 8. Configuration Register

The Configuration Register is only readable. It consists of four 8 bit registers accessible via the VME bus and locally by the i960.

Each configuration register provides two hexadecimal digits indicating the version of two individual onboard subsystems or its features as follows:

Device codes and Format

Reading these addresses from TCU0 or 1 provides 0xFF.

| VME Address | Local Address | Action | Mode R/W | Data Bit D7,D6,D5,D4 | Data Bit D3,D2,D1,D0 | Only TCUI |
|-------------|---------------|-------------------------------|----------|---------------------------------------|---------------------------------|-----------|
| 19220020 | B0000020 | TCU3 Configuration Register 0 | R | PRZV processor version: 0x1=i960HD | TCU version: 0x3=TCU3 | 3 |
| 19220024 | B0000024 | TCU3 Configuration Register 1 | R | Instr. RAM: 0x2=2MB | Clockfrequ: 0x0=25MHz | 3 |
| 19220028 | B0000028 | TCU3 Configuration Register 2 | R | RTP RAM: 0x0=8k | Slot#: 0x0,...,7= 1,...,8 | 3 |
| 1922002C | B000002C | TCU3 Configuration Register 3 | R | x: 0xF | x: 0xF | 3 |

4. 4. 9. VME bus port

The interface of the i960 to and from the VME–Bus has 32 bit wide address and data busses. The i960 can set off interrupts and read and write accesses via the VME master

port. The local TCU resources can be accessed by an external VME master via the VME slave port.

4. 4. 9. 1. Master Port

Features

- Single word (4 bytes) write and read accesses possible, no nibble no page mode
- Address region 8 of the i960 has been assigned to the VME master port.
- Region 8 is 1 of 16 regions coded by the 4 highest address bits of the i960. That means the processor can only issue dynamically address 0,...,27 to select a VME bus address location. The VME address bit 28 to 31 are issued out of the address register VHARG.
This register can be loaded by data lines 0,...,3 and a special device code.

The VME Bus Address space is accessible in two steps:

1. Load the VME High Address Register (VHARG) with 4 high address bits <31..28> via data lines <D3..D0>
2. Access the VME Bus Address space with i960 via region "8"

Device Codes

| VME Address | Local Address | Action | Mode R/W | Oprd. Byte 3 2 1 0 | Only TCUi |
|-------------|----------------------------|---|----------|--------------------|-----------|
| no access | 80000000 to 8FFFFFFC | Region 8 access of i960 to VME address range 0xVxxxxxxx; V=content of the vharg register | R/W | b b b b | 3 |
| no access | B0000018 | Access to High Address Register vharg; vharg=VME <31..28> | W | x x x b/2 | 3 |

4. 4. 9. 2. Slave Port

The local RAM and nearly all TCU registers are read-/ writeable from the VME-Bus. This is an important test feature. All functions, which are normally controlled by the i960 can also be activated via the VME-Bus e.g. by the CCU.
Refer to the "VME Address" column of the "Memory Map and Device Code" table.

Features

- Single word (4 bytes) write and read accesses possible, no nibble no page mode

4. 5. The Real Time Unit

Comprising:

- Real-Time-Program (RTP) RAM, organized as dual port ring buffer
- Acquisition Controller with:
Address generator,
Duration generator,
Loop counter,

- Clock Generator

The sources of all TCU output signals and AQ-Bus activities are the Real-Time-Program entries which are stored in the 8k deep Dual-Port RAM and read out on a 20 MHz time base by the Acquisition Controller.

The Dual-Port RAM and also each entry is 128 bits wide.

The Dual-Port RAM is organized as ring buffer, sequentially written by the i960 and sequentially read out by the Acquisition Controller and its Address Generator.

4.5.1. Address Generation

The Address Generator determines the next address to be read. It is driven by the Clock Generator which fixes the point in time when the next address is to be read.

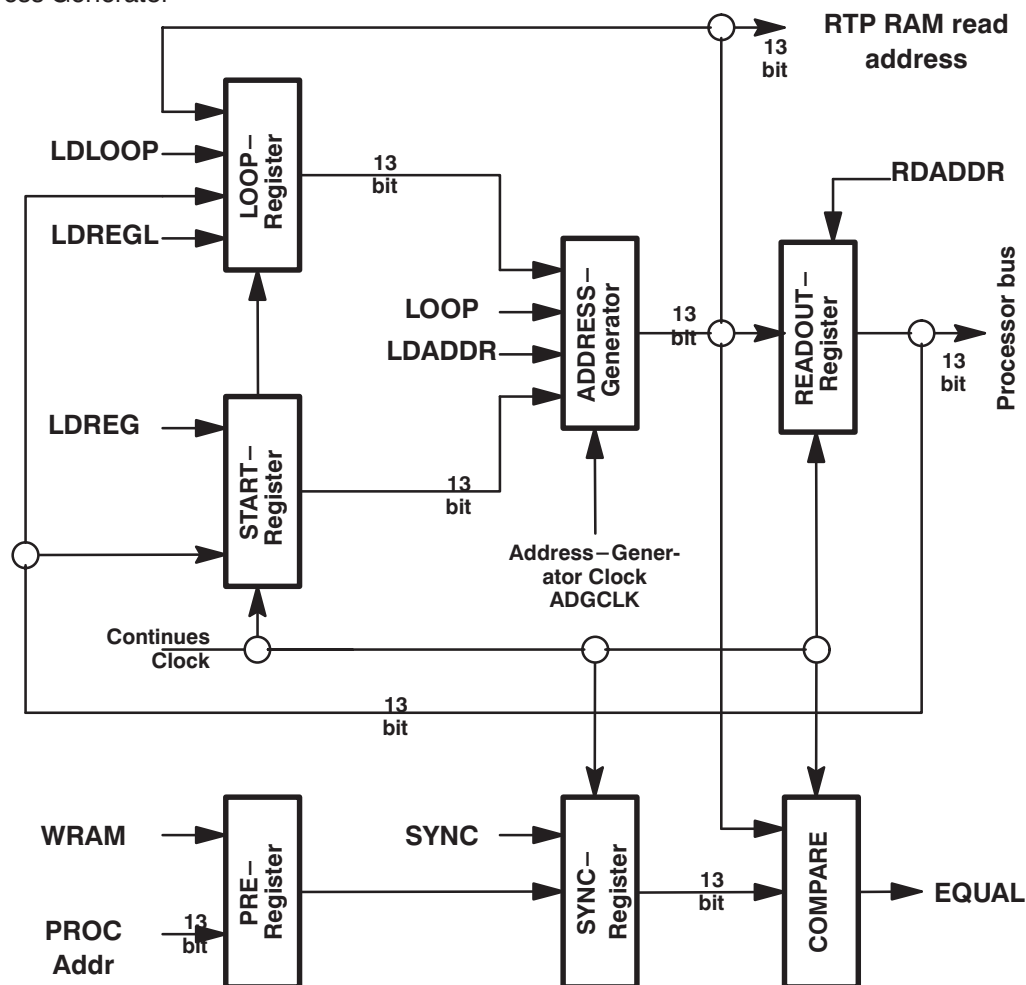
The point in time depends on any duration requested in the current RTP entry. The address is sequential in normal operation or nonsequential during loops.

The normal operation can be influenced by commands like "START", "STOP", "LOAD" and so on.

| VME Address | Local Address | Action | Mode R/W | Oprd. Byte 3 2 1 0 | Only TCUi |
|-------------|---------------|--|----------|--------------------|-----------|
| 19221080 | C0000080 | RUN Start of Clock Generator (run of Address Generator) | W | x x x x | |
| 19221084 | C0000084 | STEP Stops Clock Generator, steps to next edge, (The effect of each STEP is the next clock edge, high>low>high>low) | W | x x x x | |
| 19221088 | C0000088 | LDREG Preload of Start Register, (Clock Generator keeps running) | W | x x b b | |
| 1922108C | C000008C | STOP Stop of Clock Generator, (clock at "high") | W | x x x x | |
| 19221090 | C0000090 | START Load Address Gen. through Start Reg. with operand and Start Clock Generator | W | x x b b | |
| 19221094 | C0000094 | Step Address Generator by one if Clock Gen. off | W | x x x x | |
| 19221098 | C0000098 | LDADDR Load Address Gen. through Start Reg. with operand if Clock Generator is off, (Clock Generator stays off) | W | x x b b | |
| 1922109C | C000009C | DEVST Load Address Generator with value of Start Register (During RUN this address becomes valid after the actual duration) | W | x x x x | |

| VME Address | Local Address | Action | Mode R/W | Oprd. Byte 3 2 1 0 | Only TCUi |
|-------------|---------------|---|----------|--------------------|-----------|
| 192210C0 | C00000C0 | RDADDR Read of Address Generator Register | W | x x b b | |
| 19221100 | C0000100 | INIT Stop of Clock Generator and Clear all registers of Address Generator | W | x x x x | |

Figure 9: Address Generator



4. 5. 2. Debug Register of Duration Generator and Loop Counter

Reading this address provides the state of 2 debugging signals

Format

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------|-------|-------|-------|-------|-------|--------|--------|
| x | x | x | x | x | | DebugL | DebugD |

Loop Counter debug signal

Duration Generator debug signal

Device Codes

| VME Address | Local Address | Action | Mode R/W | Oprd. Byte 3 2 1 0 | Only TCUI |
|-------------|---------------|---|----------|--------------------|-----------|
| 19221020 | C0000020 | Read Debugging Signals of duration generator and loop counter | R | x x x b | |

4. 5. 3. Processor Interrupt XINT0

Interrupt 0 of "i960" is used as error interrupt. It is released by the run out condition in the real time program memory. That means the generated read address is getting equal to the last write position of the i960. EQUAL becomes active and triggers off interrupt 0.

Interrupt 0 can be enabled, cleared and disabled via VME-Bus or by local access.

Device Codes

| VME Address | Local Address | Action | Mode R/W | Oprd. Byte 3 2 1 0 | Only TCUI |
|-------------|---------------|-------------------------------|----------|--------------------|-----------|
| 19221040 | C0000040 | Enable Interrupt 0 | W | x x x x | |
| 19221040 | C0000040 | Clear and disable Interrupt 0 | R | x x x x | |

4. 5. 4. State of Trigger Signals

The operation of the Acquisition Controller and the i960 can be influenced by 4 external trigger signals TRIG3,...,0.

This access provides the current state of TRIG3,...,0 on Data lines D3,...,D0.

| VME Address | Local Address | Action | Mode R/W | Oprd. Byte 3 2 1 0 | Only TCUI |
|-------------|---------------|---------------------|----------|--------------------|-----------|
| 19221028 | C0000028 | Read Trigger Inputs | R | x x x b | |

4. 5. 5. Processor Interrupt XINT1

Interrupt 1 of "i960" can be initiated by one of the 4 trigger inputs TRIG0, TRIG1, TRIG2 or TRIG3.

Associated to interrupt 1 is the Interrupt-Control-Register. It can be accessed via VME-Bus or by the "i960".

Device Codes

| VME Address | Local Address | Action | Mode R/W | Oprd. Byte 3 2 1 0 | Only TCUI |
|-------------|---------------|---------------------------------|----------|--------------------|-----------|
| 19221010 | C0000010 | Set Interrupt Control Register1 | W | x x x b | |
| 19221010 | C0000010 | Clear and disable Interrupt 1 | R | x x x x | |

Table 12: Binary Format of Interrupt Control Register

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | Meaning |
|----|----|----|----|----|----|----|----|---|
| x | x | x | 0 | 0 | | | | TRIG0 |
| x | x | x | 0 | 1 | | | | TRIG1 |
| x | x | x | 1 | 0 | | | | TRIG2 |
| x | x | x | 1 | 1 | | | | TRIG3 |
| x | x | x | | | 0 | 0 | 0 | INT if trigger signal at low level |
| x | x | x | | | 0 | 0 | 1 | INT if trigger signal at high level |
| x | x | x | | | 0 | 1 | 0 | INT if trigger signal goes from high to low |
| x | x | x | | | 0 | 1 | 1 | INT if trigger signal goes from low to high |
| x | x | x | | | 1 | 0 | 0 | INT at any transition of trigger |
| x | x | x | | | 1 | 0 | 1 | not defined |
| x | x | x | | | 1 | 1 | 0 | NO INT possible, masked |
| x | x | x | | | 1 | 1 | 1 | INT set independent of any trigger signal |

The interrupt has to be cleared via VME-Bus or by local access.

Clearing the interrupt sets also the bits D2, D1 and D0 to 1.

4. 5. 6. Debug Interrupt XINT7 Register

It provides the source bits for the Loop Counter test signals TESTA, TESTB, TESTC and for the interrupt7 to the i960.

Format

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| XINT7 | x | x | x | x | TESTC | TESTB | TESTA |

i960 interrupt7
1=activated
0=cleared
Loop Counter test signals

Device Codes

| VME Address | Local Address | Action | Mode R/W | Oprd. Byte 3 2 1 0 | Only TCUi |
|-------------|---------------|--|----------|--------------------|-----------|
| 19220008 | B0000008 | Debug and local Interrupt Register of "i960" | R/W | x x x b | |

4. 5. 7. Access to the Real-Time-Program (rtp) entries

A Real-Time-Program entry is composed out of 4 words (NMR words) and has to be fed into the Dual-Port RAM by the i960 within 4 separate write cycles to addresses which differ in their low significant hex digit.

Table 13: Real–Time–Program Addresses

| Address of VME Acc. | Address of Local Acc. | Word# | Part of Real–Time–Program entry | Mode R/W | Size of Operand Byte 3 2 1 0 |
|----------------------------|----------------------------|-------|---|----------|---------------------------------|
| 19200000 to 1921FFF0 | E0000000 to E001FFF0 | 1 | rtp data bit 0,...,31 Sequencer Control Signals | w | b b b b |
| 19200004 to 1921FFF4 | E0000004 to E001FFF4 | 2 | rtp data bit 32,...,64 AQ–Bus + 3 TCU Output Signals | w | b b b b |
| 19200008 to 1921FFF8 | E0000008 to E001FFF8 | 3 | rtp data bit 65,...,95 TCU Output Signals | w | b b b b |
| 1920000C to 1921FFFC | E000000C to E001FFFC | 4 | rtp data bit 96,...,127 TCU Output Signals | w | b b b b |

4. 5. 7. 1. Enable of the TCU Output Signals

The TCU Output Signals (w2_29 to w2_31, w3_0 to w3_31 and w4_0 to w4_31) are switched to the high impedance state by the following actions:

- Power on reset
- Reading from the device code "NMR Output Enable off"

They can only be switched to the active state by writing to the device code "NMR Output Enable on".

The "Soft Reset" device code has no influence to the TCU Output Signals.

Device Codes

| VME Address | Local Address | Action | Mode R/W | Oprd. Byte 3 2 1 0 | Only TCUI |
|-------------|---------------|--|----------|-----------------------|-----------|
| 19221200 | | NMR Output Enable on, sets the TCU Output Signals active | W | x x x x | |
| 19221210 | | NMR Output Enable off, sets the TCU Output Signals inactive | R | x x x x | |

4. 5. 7. 2. Meaning and work of the Real–Time–Program (rtp) data bits

All 4 words of each Real–Time–Program entry can be read out and clocked into their output registers at the same time. The data bits of each word take their effects out of these registers.

The output register of word 3 and word 4 are only updated by their corresponding bits of the Real–Time–Program if special enable bits in word 1 (w1_0, w1_1) are set.

That means word 3 and 4 of an entry have only to be written by the i960 if they have to be updated by that entry.

The rtp data bits of each word are labeled as w1_0 to w1_31 for word 1 and so on.

Word 1 of the Real–Time–Program entry:

Table 14: Format of Word 1

| w1 data bits | | | | | | | | | | | | | | | Comment | | | | |
|--------------|----------|-----------------|----|----|------------|------|----|------------|------------|----------------|----------------|----------------|----------------|-----|---|------------------|-----------------------|-------------------------|-----------------|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | ... | 14 | 13 | ... | | 4 | 3 | 2 | 1 |
| 0 | Duration | | | | | | | | | | | | | | x | v | v | v | normal duration |
| 1 | 0 | Number of Loops | | | | | | | | Short Duration | x | v | v | v | load loop counter | | | | |
| 1 | 1 | 0 | 1 | 1 | T1/0 | C1/0 | P | don't care | | | Short Duration | x | v | v | v | conditional WAIT | | | |
| 1 | 1 | 1 | 0 | 0 | T1/0 | C1/0 | P | don't care | | | Short Duration | x | v | v | v | conditional NMI | | | |
| 1 | 1 | 1 | 1 | 0 | T1/0 | 0 | 0 | P | don't care | | | Short Duration | x | v | v | v | conditional loop back | | |
| 1 | 1 | 1 | 1 | 0 | x | x | 1 | 1 | 0 | don't care | | | Short Duration | x | v | v | v | unconditional loop back | |
| 1 | 1 | 1 | 1 | 1 | don't care | | | | | Short Duration | x | v | v | v | loop if >0 and decrement loop counter; (repeat) | | | | |

Legend

| | |
|------|---------------------------|
| v | valid in this combination |
| T1/0 | Trigger selection |
| C1/0 | Condition selection |
| P | Polarity |

Bit description

| | |
|------|--|
| w1_2 | Removes any 12.5 nsec step difference between the Real–Time–Memory read–out clock and the system clock |
| w1_1 | Enable update of w4 output register |
| w1_0 | Enable update of w3 output register |

Command description

| | |
|-------------------|--|
| Normal duration | Loads the Duration Counter with a 27–bit number of 50–nsec steps, waits up to the run out of the counter before going to the next entry. Any other instruction loads the counter with a 9–bit number which leads to a Short Duration up to going to the next entry. |
| Load loop counter | Loads the Number of Loops into the Loop Counter and the address of the next entry into the LOOP Register of the Address Generator. Goes to the next entry. |
| Conditional NMI | If the condition is true while this entry is effective or later on: Sets off a NMI to the i960 |
| Conditional WAIT | Stops and waits up to the condition is true: Goes then to and carries out the next entry. |
| Cond. loop back | If the condition is true while this entry is effective or later on: Goes to and carries out the next entry and goes then to the address in the LOOP Register. |
| Uncond. loop back | Goes to and carries out the next entry and goes then to the address in the LOOP Register. There is no escape from this loop. |

Decr. loop counter If the loop counter > 0:
Goes to and carries out the next entry, decrements the loop counter and goes then to the address in the LOOP Register.

Meaning of w1_22 to 26

- The destination to be controlled by bits w1_22 to w1_26 is selected by the setting of the bits w1_27 to w1_31
- Control destinations are:
NMI, WAIT, LOOP

NMI Conditions

The NMI of the "i960" can be initiated by one of the 4 trigger inputs TRIG0, TRIG1, TRIG2 or TRIG3.

The trigger, the condition and the polarity are selected by the bits 22 to 26.

The NMI has to be cleared via VME-Bus or by local access.

| VME Address | Local Address | Action | Mode R/W | Oprd. Byte 3 2 1 0 | Only TCUi |
|-------------|---------------|-----------------------|----------|--------------------|-----------|
| 19221030 | C0000030 | Clear and disable NMI | R | x x x x | |

Table 15: Binary Format of the NMI control instruction in w1

| 31 | 30 | 29 | 28 | 27 | 26 (T1) | 25 (T0) | 24 (C1) | 23 (C0) | 22 (P) | Meaning |
|----|----|----|----|----|---------|---------|---------|---------|--------|---|
| 1 | 1 | 1 | 0 | 0 | | | | | | Sets the NMI control register with D26,...D22 |
| | | | | | 0 | 0 | | | | TRIG0 |
| | | | | | 0 | 1 | | | | TRIG1 |
| | | | | | 1 | 0 | | | | TRIG2 |
| | | | | | 1 | 1 | | | | TRIG3 |
| | | | | | | | 0 | 0 | 0 | NMI if trigger signal at low level |
| | | | | | | | 0 | 0 | 1 | NMI if trigger signal at high level |
| | | | | | | | 0 | 1 | 0 | NMI if trigger signal goes from high to low |
| | | | | | | | 0 | 1 | 1 | NMI if trigger signal goes from low to high |
| | | | | | | | 1 | 0 | 0 | NMI at any transition of trigger |
| | | | | | | | 1 | 0 | 1 | not defined |
| | | | | | | | 1 | 1 | 0 | NO NMI possible, masked |
| | | | | | | | 1 | 1 | 1 | NMI set independent of any trigger signal |

WAIT Conditions

The WAIT logic provides the hold function of the Acquisition Controller.

The hold time depends on the conditions in the "Meaning" column. Reset or a special device code stop the WAIT situation .

| VME Address | Local Address | Action | Mode R/W | Oprd. Byte 3 2 1 0 | Only TCUi |
|-------------|---------------|------------------------|----------|--------------------|-----------|
| 19221050 | C0000050 | Clear and disable WAIT | R | x x x x | |

Table 16: Binary Format of WAIT control instruction in w1

| 31 | 30 | 29 | 28 | 27 | 26 (T1) | 25 (T0) | 24 (C1) | 23 (C0) | 22 (P) | Meaning |
|----|----|----|----|----|------------|------------|------------|------------|-----------|---|
| 1 | 1 | 0 | 1 | 1 | | | | | | Sets the WAIT control register with D26,...D22 |
| | | | | | 0 | 0 | | | | TRIG0 |
| | | | | | 0 | 1 | | | | TRIG1 |
| | | | | | 1 | 0 | | | | TRIG2 |
| | | | | | 1 | 1 | | | | TRIG3 |
| | | | | | | | 0 | 0 | 0 | WAIT if trigger signal at low level |
| | | | | | | | 0 | 0 | 1 | WAIT if trigger signal at high level |
| | | | | | | | 0 | 1 | 0 | WAIT until trigger signal goes from high to low |
| | | | | | | | 0 | 1 | 1 | WAIT until trigger signal goes from low to high |
| | | | | | | | 1 | 0 | 0 | WAIT until any transition of trigger |
| | | | | | | | 1 | 0 | 1 | not defined |
| | | | | | | | 1 | 1 | 0 | Unconditional WAIT |
| | | | | | | | 1 | 1 | 1 | NO WAIT |

LOOP Conditions

Table 17: Binary Format of LOOP control instruction in w1

| 31 | 30 | 29 | 28 | 27 | 26 (T1) | 25 (T0) | 24 (C1) | 23 (C0) | 22 (P) | Meaning |
|----|----|----|----|----|------------|------------|------------|------------|-----------|--|
| 1 | 1 | 1 | 1 | 0 | | | | | | Sets the LOOP control register with D26,...D22 |
| | | | | | 0 | 0 | | | | TRIG0 |
| | | | | | 0 | 1 | | | | TRIG1 |
| | | | | | 1 | 0 | | | | TRIG2 |
| | | | | | 1 | 1 | | | | TRIG3 |
| | | | | | | | 0 | 0 | 0 | LOOP if trigger signal at low level |
| | | | | | | | 0 | 0 | 1 | LOOP if trigger signal at high level |
| | | | | | | | 0 | 1 | 0 | not defined |
| | | | | | | | 0 | 1 | 1 | not defined |
| | | | | | | | 1 | 0 | 0 | not defined |
| | | | | | | | 1 | 0 | 1 | not defined |
| | | | | | | | 1 | 1 | 0 | Unconditional LOOP |
| | | | | | | | 1 | 1 | 1 | not defined |

Word 2 of the Real-Time-Program entry:

Table 18: Format of Word 2

| w2 data bits | Signal | Function |
|--------------|------------|------------------------|
| 31 | BLK_GRAD_Z | Gradient Blanking Puls |
| 30 | BLK_GRAD_Y | " |
| 29 | BLK_GRAD_X | " |

| w2 data bits | Signal | Function |
|--------------|---------------|---|
| 28 | RCU_GO | Generate RCU–GO Pulse (high active) |
| 27 | AQ–Bus ENABLE | initiates an AQ–Bus Cycle as: AQS at T0_5 (high active) AQSEL at J0_E18 (high active) AQSTROBE at J2_A32 (high edge) |
| 26 | AQEXEC | AQ–Bus EXEC Signal starts action on the FCU or GCU as: AQEXEC at J0_D18 and J2_C32 (low active) |
| 25 | | |
| 24 | | |
| 23 | AQA(3) | AQ–Bus Address Bit |
| 22 | AQA(2) | ” |
| 21 | AQA(1) | ” |
| 20 | AQA(0) | ” |
| 19 | AQS(3) | AQ–Bus sub Address |
| 18 | AQS(2) | ” |
| 17 | AQS(1) | ” |
| 16 | AQS(0) | ” |
| 15..0 | AQD(15..0) | AQ–Bus Data |

Bit description

| | |
|-----------------|--|
| w2_29,...,w2_31 | BLK_GRAD_i are connected to the front panel coax A1, A4, A6 |
| w2_28 | RCU_GO is connected to front panel coax T0_3 |
| w2_27 | is the source of AQS at the front panel T0_5, AQSEL at J0_E18 and AQSTROBE at J2_A32 |
| w2_26 | is the source of AQEXEC_J0 at J0_D18 and AQEXEC_J2 at J2_C32 |

Word 3 of the Real–Time–Program entry:

Table 19: Format of Word 3

| w3 data bits | | | | |
|---------------|-----|----|---------------|---|
| 31 | ... | 16 | 15 | 0 |
| NMR3_15,...,0 | | | NMR2_15,...,0 | |

Bit description

| | |
|----------------|---------------------------------------|
| w3_0,...,w3_31 | drive coax outputs at the front panel |
|----------------|---------------------------------------|

Word 4 of the Real–Time–Program entry:

Table 20: Format of Word 4

| w4 data bits | | | | | | | |
|--------------------|-----|----|--------------|-----|----|---------------|---|
| 31 | ... | 24 | 23 | ... | 16 | 15 | 0 |
| NMR8_15,14,9,...,4 | | | NMR6_7,...,0 | | | NMR5_15,...,0 | |

Bit description

w4_0,...,w4_31 drive coax outputs at the front panel

4. 5. 7. 3. Signal and Pin assignment of word 2, 3 and 4

| Designation at TCU0/1 | Meaning | Destination | Signal& rtp NMR word_bit | Contact | Con- nector |
|-----------------------|-------------|-------------|--------------------------|---------|-------------|
| | | REF | 20M_IN | | ST4 |
| | | RCU | 40M_OUT_1 | 1 | T0 |
| | | GCU | 40M_OUT_2 | 2 | T0 |
| | RCU_GO | RCU | RCU_GO | 3 | T0 |
| NMR8_5 | EXT DW | RCU | w4_25 | 4 | T0 |
| | AQSTROBE | RCU, GCU | AQS | 5 | T0 |
| NMR8_6 | EXT RGP | RCU | w4_26 | 6 | T0 |
| | | | | 1 | T1 |
| NMR5_0 | GAIN_0_TR1 | | w4_0 | 2 | T1 |
| NMR5_2 | C/AB_TR1 | | w4_2 | 3 | T1 |
| | | | | 4 | T1 |
| NMR5_3 | GAIN_0_TR2 | | w4_3 | 5 | T1 |
| NMR5_5 | GAIN_2_TR2 | | w4_5 | 6 | T1 |
| | | | | 7 | T1 |
| NMR5_6 | GAIN_2_TR1 | | w4_6 | 8 | T1 |
| | | | | 9 | T1 |
| NMR5_7 | RELAY_H | | w4_7 | 10 | T1 |
| | | | | 11 | T1 |
| NMR5_9 | RELAY_Y | | w4_9 | 12 | T1 |
| NMR5_11 | RCP | | w4_11 | 13 | T1 |
| NMR5_12 | RELAY_Z | | w4_12 | 14 | T1 |
| NMR5_13 | RCP_Scope | | w4_13 | 15 | T1 |
| NMR5_14 | RCP_EXT_DEV | | w4_14 | 16 | T1 |
| NMR5_15 | RCP | | w4_15 | 17 | T1 |
| NMR6_0 | STP1_DIR | HIGH POWER | w4_16 | 18 | T1 |
| NMR6_2 | DCM_STRT | HIGH POWER | w4_18 | 19 | T1 |
| NMR6_3 | STP1_CLK | HIGH POWER | w4_19 | 20 | T1 |
| NMR6_4 | STP2_CLK | HIGH POWER | w4_20 | 21 | T1 |
| NMR6_5 | RES_STP1 | HIGH POWER | w4_21 | 22 | T1 |
| NMR6_6 | DCM_RES | HIGH POWER | w4_22 | 23 | T1 |
| NMR6_7 | GO_POS | HIGH POWER | w4_23 | 24 | T1 |
| | | | | 25 | T1 |
| | | | GND | 26 | T1 |
| NMR5_1 | GAIN_1_TR1 | | w4_1 | 27 | T1 |

| Designation at TCU0/1 | Meaning | Destination | Signal& rtp NMR word_bit | Contact | Con-connector |
|-----------------------|---------------|----------------|--------------------------|---------|---------------|
| | | | GND | 28 | T1 |
| | | | GND | 29 | T1 |
| NMR5_4 | GAIN_1_TR2 | | w4_4 | 30 | T1 |
| | | | GND | 31 | T1 |
| | | | GND | 32 | T1 |
| | | | GND | 33 | T1 |
| | | | GND | 34 | T1 |
| NMR5_8 | RELAY_X | | w4_8 | 35 | T1 |
| | | | GND | 36 | T1 |
| NMR5_10 | RACK_ON/OFF | | w4_10 | 37 | T1 |
| | | | GND | 38 | T1 |
| | | | GND | 39 | T1 |
| | | | GND | 40 | T1 |
| | | | GND | 41 | T1 |
| | | | GND | 42 | T1 |
| NMR6_1 | LB_SEL | HIGH POWER | w4_17 | 43 | T1 |
| | | | GND | 44 | T1 |
| | | | GND | 45 | T1 |
| | | | GND | 46 | T1 |
| | | | GND | 47 | T1 |
| | | | GND | 48 | T1 |
| | | | GND | 49 | T1 |
| | | | GND | 50 | T1 |
| NMR8_4 | SEL_2H/DEC | 2H Lock Switch | w4_24 | A1 | T2 |
| NMR8_7 | Q_SWITCH | | w4_27 | A3 | T2 |
| NMR8_8 | SEL_!X/2H | 2H Lock Switch | w4_28 | A2 | T2 |
| NMR0_32 | BLK_GRAD_X | GRASP | w2_29 | A4 | T2 |
| NMR0_33 | BLK_GRAD_Y | GRASP | w2_30 | A5 | T2 |
| NMR0_34 | BLK_GRAD_Z | GRASP | w2_31 | A6 | T2 |
| NMR2_0 | !LOCK_HOLD | BSMS/LCB | w3_0 | B1 | T2 |
| NMR2_1 | !HOMOSPOIL | BSMS/SCBR | w3_1 | B2 | T2 |
| NMR2_4 | !!INT_A_(Z0) | BSMS/LCB | w3_4 | B3 | T2 |
| NMR2_2 | SELH_!H/F | 1H Transm. | w3_2 | B4 | T2 |
| NMR2_3 | SELX_!X/F | 1H Transm. | w3_3 | B5 | T2 |
| NMR2_5 | MIXCC | BP | w3_5 | B6 | T2 |
| NMR2_6 | res | | w3_6 | C1 | T2 |
| NMR2_7 | RCP_PA_SWITCH | HPPR | w3_7 | C2 | T2 |
| NMR2_8 | FXA | QNP | w3_8 | C3 | T2 |
| NMR2_9 | FXB | QNP | w3_9 | C4 | T2 |

| Designation at TCU0/1 | Meaning | Destination | Signal& rtp NMR word_bit | Contact | Con-connector |
|-----------------------|--------------------|----------------|--------------------------|---------|---------------|
| | Trigger 0 | BP HR MAS | TRIG0 | C5 | T2 |
| | Trigger 1 | BSMS SLCB | TRIG1 | C6 | T2 |
| NMR2_10 | res | | w3_10 | D1 | T2 |
| NMR2_11 | res | | w3_11 | D2 | T2 |
| NMR2_12 | res | | w3_12 | D3 | T2 |
| NMR2_13 | res | | w3_13 | D4 | T2 |
| NMR2_14 | res | | w3_14 | D5 | T2 |
| NMR2_15 | res | | w3_15 | D6 | T2 |
| | Trigger 2 | TRIG STRAFI | TRIG2 | E1 | T2 |
| | Trigger 3 | TRIG Solid MAS | TRIG3 | E2 | T2 |
| NMR5_7 | RELAY_H | 1H1 KW AMPL | w4_7 | E3 | T2 |
| NMR5_8 | RELAY_X | X1 KW AMPL | w4_8 | E4 | T2 |
| NMR5_9 | RELAY_Y | X1 KW AMPL | w4_9 | E5 | T2 |
| NMR5_10 | res | | w4_10 | E6 | T2 |
| NMR5_11 | RCP | | w4_11 | F1 | T2 |
| NMR5_12 | RELAY Z | X1 KW AMPL | w4_12 | F2 | T2 |
| NMR5_13 | RCP_Scope | | w4_13 | F3 | T2 |
| NMR5_14 | RCP_EXT_DEV | | w4_14 | F4 | T2 |
| NMR5_15 | RCP | | w4_15 | F5 | T2 |
| NMR8_9 | res | | w4_29 | F6 | T2 |
| NMR3_0 | res | | w3_16 | G1 | T2 |
| NMR3_1 | res | | w3_17 | G2 | T2 |
| NMR3_2 | res | | w3_18 | G3 | T2 |
| NMR3_3 | res | | w3_19 | G4 | T2 |
| NMR3_4 | res | | w3_20 | G5 | T2 |
| NMR3_5 | res | | w3_21 | G6 | T2 |
| NMR3_6 | res | | w3_22 | H1 | T2 |
| NMR3_7 | res | | w3_23 | H2 | T2 |
| NMR3_8 | ECG_START_TRIG | MED | w3_24 | H3 | T2 |
| NMR3_9 | AUT_TUNG_IN | MED | w3_25 | H4 | T2 |
| NMR3_10 | AKTIV_DEC_RES | MED | w3_26 | H5 | T2 |
| NMR3_11 | AKTIV_DEC_RES | MED | w3_27 | H6 | T2 |
| NMR3_12 | Customer specified | MED | w3_28 | I1 | T2 |
| NMR3_13 | Customer specified | MED | w3_29 | I2 | T2 |
| NMR3_14 | Customer specified | MED | w3_30 | I3 | T2 |
| NMR3_15 | Customer specified | MED | w3_31 | I4 | T2 |
| NMR8_14 | res | | w4_30 | I5 | T2 |
| NMR8_15 | res | | w4_31 | I6 | T2 |

4. 6. Device access via external Busses

4. 6. 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. 6. 2. BBIS access

The TCU BBIS Prom responds to the following BBIS address.

| 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 | |
| TCU | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 010 |