ESO - EUROPEAN SOUTHERN OBSERVATORY

E U R O P E A N S O U T H E R N O B S E R V A T O R Y
Organisation Européenne pour des Recherches Astronomiques dans l'Hémisphère Austral
Europäische Organisation für astronomische Forschung in der südlichen Hemisphäre

VERY LARGE TELESCOPE

New General Detector Controller (NGC)

USER MANUAL

Doc.-No. VLT-MAN-ESO-13660-4510

Issue 1
Date 18/01/2008
90 pages

Prepared:	M.Meyer		
	Name	Date	Signature
Approved:	D.Baade		
	Name	Date	Signature
Released:	A.Moorwood		
	Name	Date	Signature

CHANGE RECORD

Issue	Date	Affected Paragraphs(s)	Reason/Initiation/Remarks
1	05/02/2008	All	First version

TABLE OF CONTENTS

PURPOSE

This document describes operation of the New General Detector Controller REV1 hardware. It gives an understanding of the basic operation, tells about the implemented firmware and is also intended as a programmers guide for software evaluation.

Reference Documents

NGC Infrared Detector Control Software – User Manual **Document Number: VLT-MAN-ESO-13660-4085**	Jörg Stegmeier
NGC Optical DCS - User Manual **Document Number: VLT-MAN-ESO-13660-4086**	Andrea Balestra Claudio Cumani
NGC Base Software - Design Description **Doc.No. VLT-SPE-ESO-13660-3836**	Jörg Stegmeier
Next Generation detector Controller Requirements **ESO-Doc. No. VLT-SPE-ESO-13660-3207**	D. Baade

Links

FPGA	http://www.xilinx.com
ADC, DAC	http://www.analog.com
Printed Board	http://www.andus.de

List of Abbreviations/Acronyms

ADC	Analog to Digital Converter
AQ	Data Acquisition
DAC	Digital to Analog Converter
DFE	Detector Front-End Electronics
DBE	Detector Back-End Electronics
DMA	Direct Memory Access
FPGA	Field Programmable Gate Array
FIFO	First in First out Memory
IRACE	Infrared Array Control Electronics
IRQ	Interrupt request
NGC	New General Controller
RTC	Real-time Computer
Rx	Link Receiver
RxTx	Link Transceiver
Tx	Link Transmitter

1 INTRODUCTION

NGC is a modular system consisting of the Back-End module with PCI based connection to the data acquisition computer and the Front-End module(s), generating and receiving detector signals. Data and control signals between Back-End and Front-End modules are on fiber-optic link(s) with transmission rates of 2.5 GBit/s. The modularity of the system allows many combinations as multiple Back-Ends or multiple Front-Ends or combinations as desired.
Emphasis is given to low power dissipation, what is mainly important for the Front-End unit to allow operation without cooling units. The Basic Board, a complete four channel system on one board of standard VME 6U size, consumes less than 10 Watts. The AQ32 board with 32 video channels on a board of the same size has ~15 Watts of power consumption.
No processor on the Front-End side is implemented. The data acquisition computer can address all Front-End functions over the fiber link. Result is a quiet system without difficult to control processor bus activity during data acquisition.
There is no parallel video or communication data bus on the Front-End. All data and communication transfer runs over high speed serial links with transmission rates of 2.5 GBit/s. Result is minimum disturbance for the anticipated low noise operation on the low level detector signals.
All voltages for clocks and biases of the detector are remotely programmable also during readout of the detector to allow maximum comfort for evaluation and test.
Digital galvanic isolated outputs for shutter, wobbling mirror and markers are provided and the system can accept a trigger input for synchronizing the detector read-out to external events.
Monitors for video and clocks are on front-panel connectors for evaluation and maintenance.
All detector bias voltages and currents can be measured with the implemented telemetry system.

A minimum number of different components are used, glue logic is not needed due to the fact that all digital logic is implemented in high density VIRTEX-I I Pro FPGAs. All this makes maintenance easy and reliability high.

2 OVERVIEW

2.1 Block

There are two main groups of modules (Figure 1) connected by fiber duplex connection :

- The Detector Back-End Electronics.
- The Detector Front-End Electronics consists of the Basic Module(s) and if needed additional AQ modules. These are interconnected by high speed copper serial links on the backplane for command and data transfer.

The basic link configuration is the linear connection of modules. Commands are routed always from the Back-End to the first Detector Front-End Electronics module. Additional DFE modules are addressed by wormhole routing from previous modules. The same happens for answers or video data from DFE modules to DBE modules.
If more bandwidth is needed two links in parallel can be used (needs different IP on FPGA).
Additional functionality can be provided if frames of video data are routed directly out of AQ modules to additional receivers, e.g. PCI based DBE's.

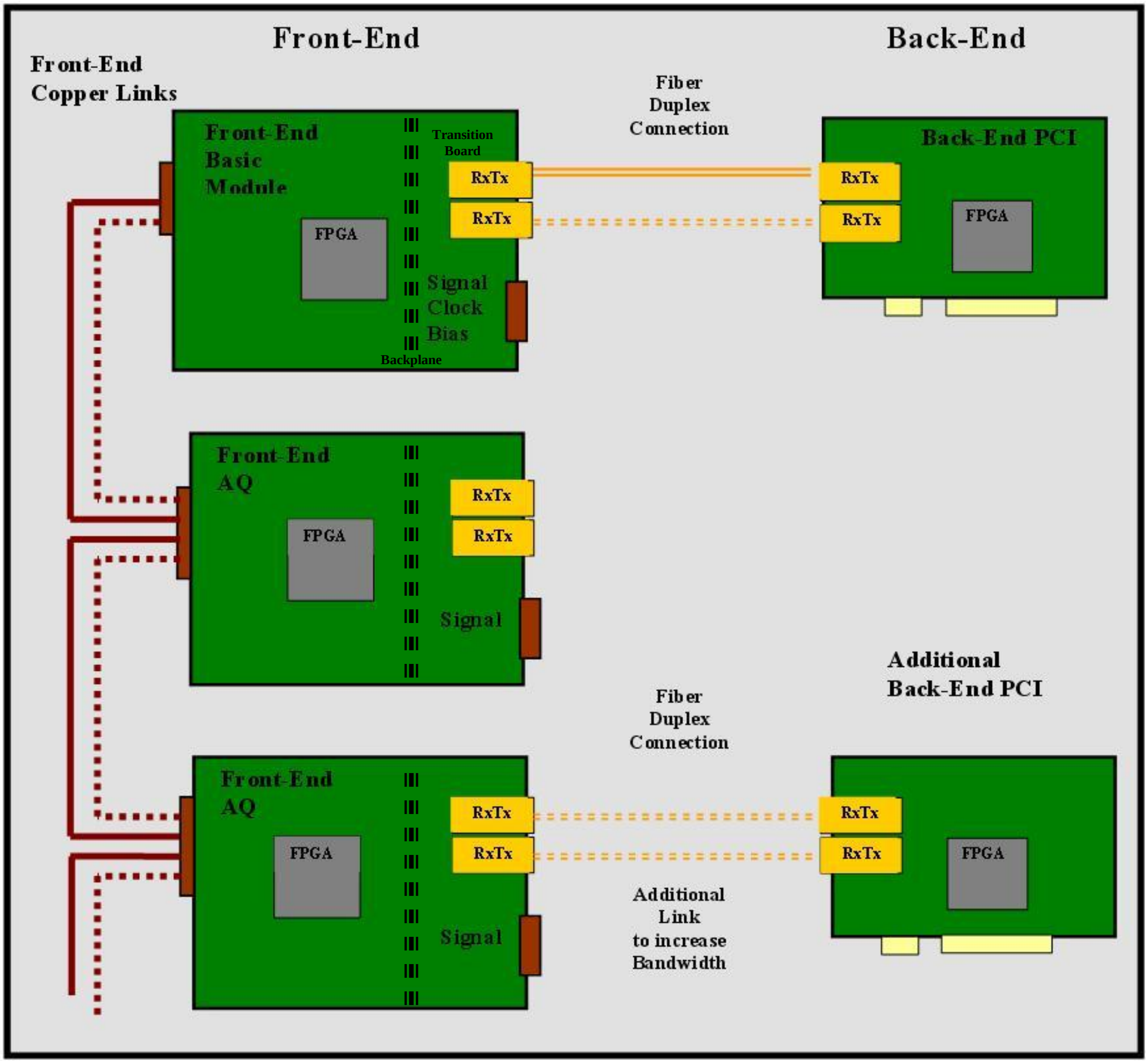


Figure 1 System Block

2.2 Minimum DFE System – Basic Board with Backplane and Transition Board

A complete DFE consists of the main board(s), the backplane and the Transition Board(s) (Figure 2).
The backplane establishes the inter board module connections.
The Transition Board sets up the connection to external functions like clocks, biases, video inputs and fiber links. In addition the Transition Board can hold special functionality not implemented in the main board.

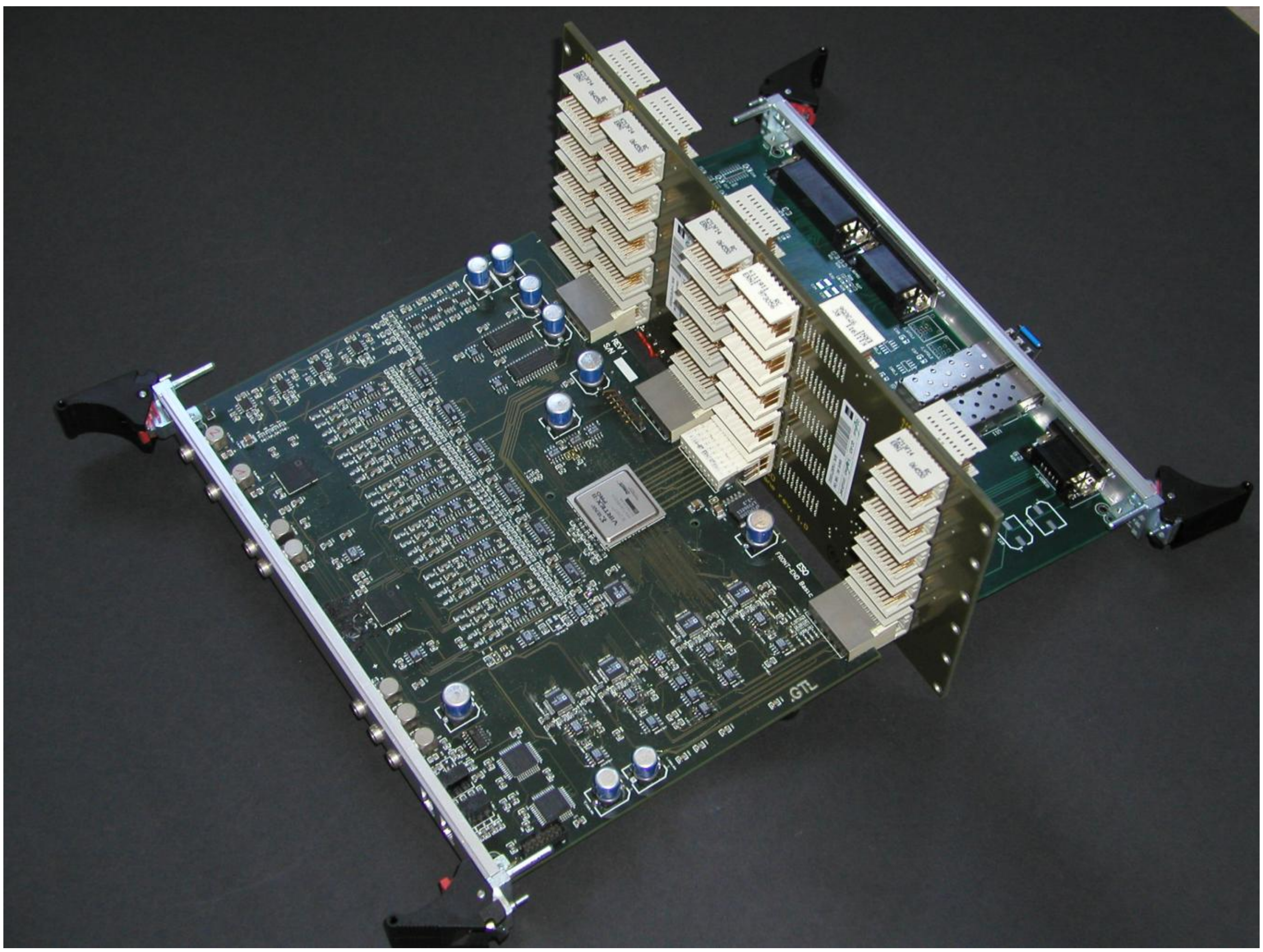


Figure 2 Minimum NGC System with Basic Board, Backplane and Transition Board

3 FUNCTIONAL DESCRIPTION

3.1 High Speed Links

Data transfer and communication work purely on serial links. The bandwidth of one link is ~200MB/s and scales well with 33MHz PCI 64 . The link architecture is determined by the connections on the backplane (Figure 3 and Figure 38), the use of the Transition Board back panel fiber optic links and the firmware in the FPGA.

Terminology : Upstream link is a link in direction towards the Back-end, Downstream link is a link in direction away from the Back-end. Sidestream link is the link to the fiber interface of the Transition Board.

On slot 1 the upstream link is routed by backplane connection always to the fiber interface on the Transition board. The Downstream link connects over the backplane to the Upstream link of the next board in the chain. Succeeding boards connect serially to each other over the backplane.
More bandwidth and computing power will be available, when on a board the Upstream link is closed, the Sidestream link is opened and the Sidestream link is routed out of the Fiber interface on the Transition Board to an additional PCI interface. This PCI interface has then full control on the chain downstream (see also 3.5.1 Front-End Configuration Register).

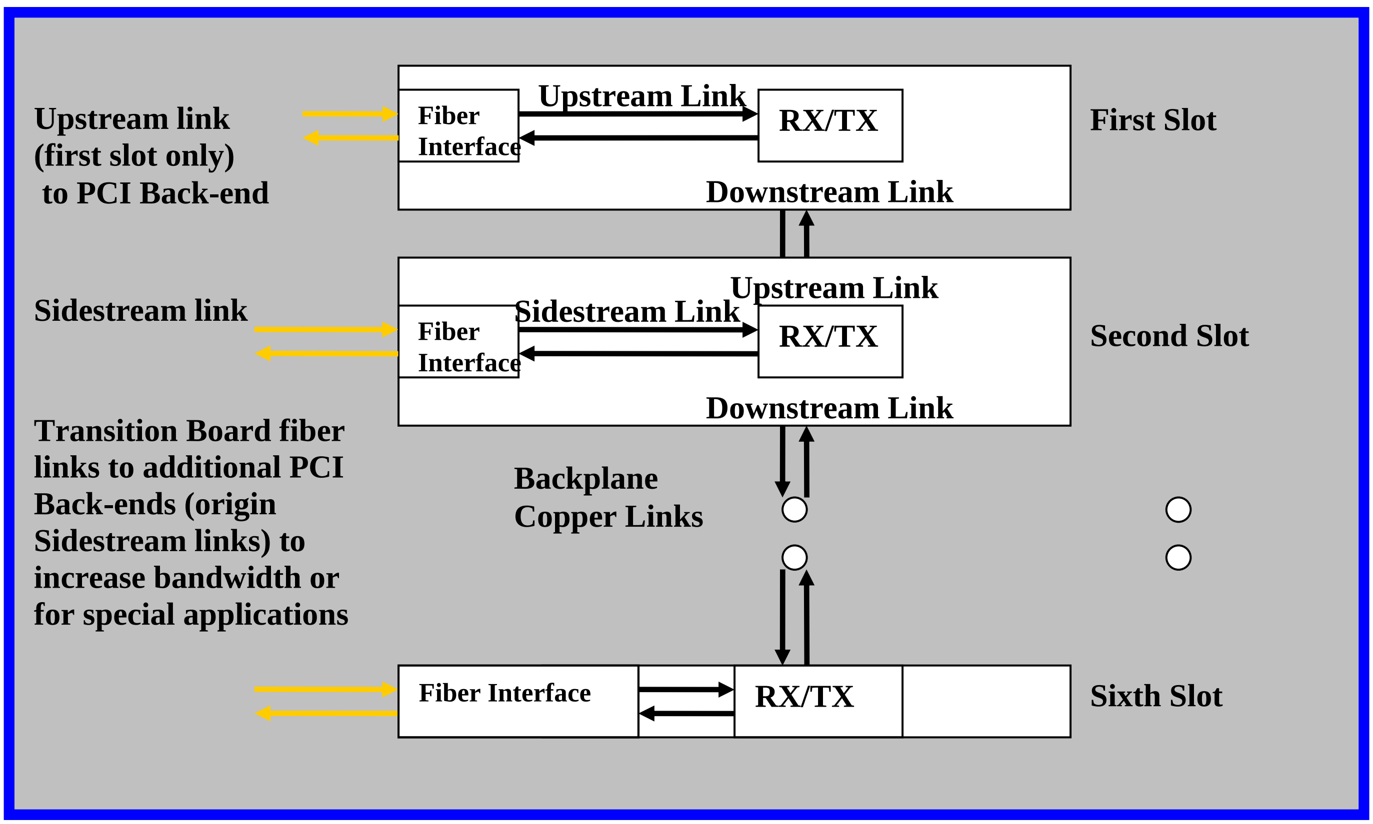


Figure 3 High Speed Links

3.2 Link Transmission and Function Addressing

The communication between all system modules is based on packet transmission over serial links. The principle of communication is the same for all modules (Figure 4). A packet structure is defined to address a function (e.g. a register or memory in a front-end module) for read or write. Upstream of module1 is the Back-End module. If from there a write to a function in module1 has to be executed, the packet header (#2) addresses first RX COM, then the address of the function (#ADDR). The next word determines that a WRITE has to be executed. The data to write are in the next word (DATA).

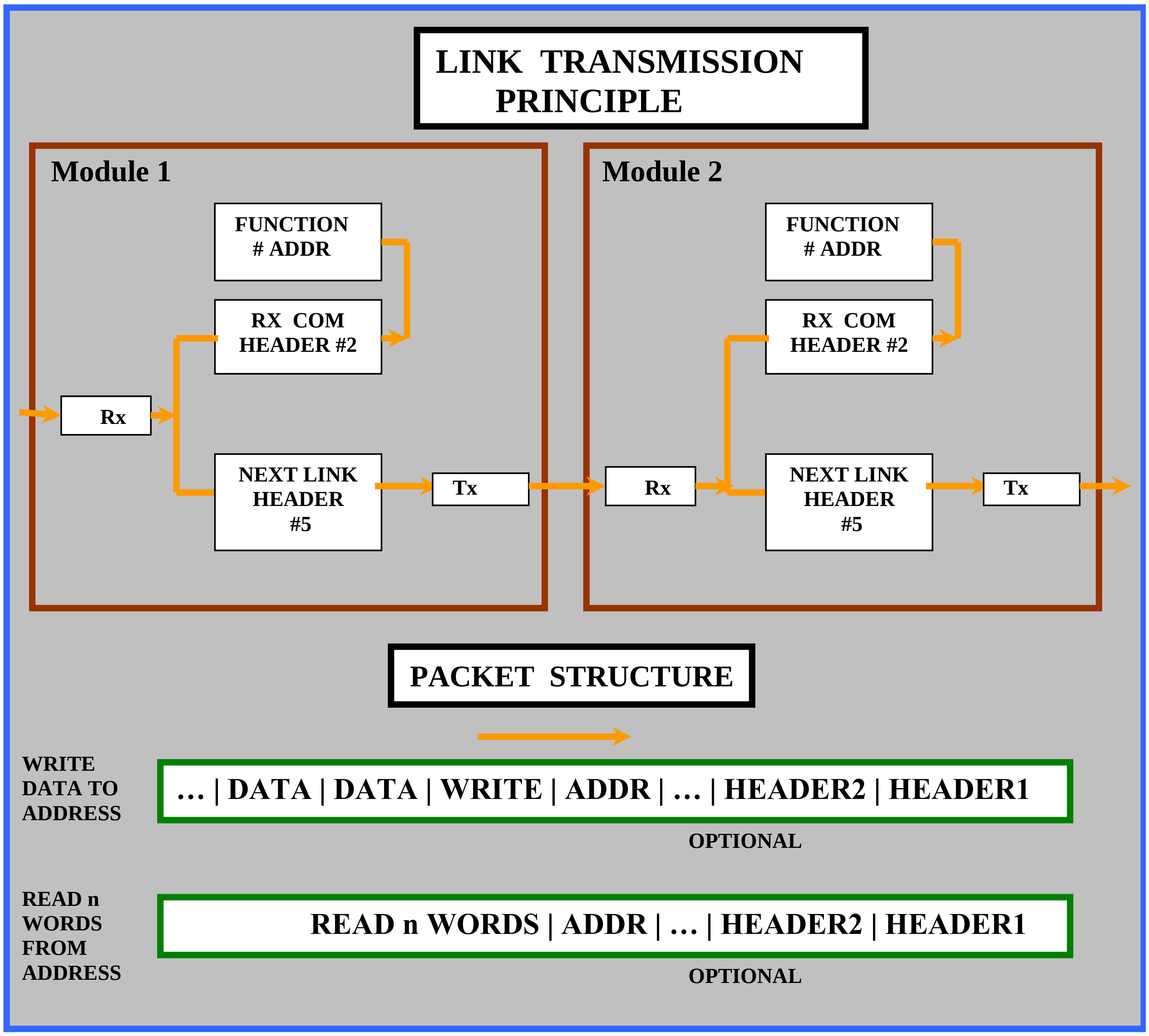


Figure 4 Link Transmission

If more data statements follow, RX COM automatically increments #ADDR, and the writes are guided to consecutive locations.

A write to #ADDR in module2 has as first word the header for NEXT LINK (#5), the next word addresses RX COM (#2), next word is WRITE, then the DATA words follow.
Any word in the packet is 32 Bits wide.

Example

Write data word content (Sequencer start = 1) to Sequencer Command register #6000 in module1 :
Packet must be filled with : #2 #6000 #0 #1

If the sequencer would be in module2 on downstream link :
Packet must be filled with : #5 #2 #6000 #0 #1

Reading data from a module has a similar structure. The function is addressed as before only the WRITE (#0) has to be replaced with a READ (#80000000) and then the number of words to read (#Number of Words). The read words are then automatically transmitted back to the receiver module (RX COM) in the Back-End.

Example

Read 10 words from sequencer memory in module1 (Sequencer RAM #4000) :
Packet must be filled with : #2 #4000 #80000000 #A

If the sequencer would be in module2 on downstream1 link :
Packet must be filled with : #5 #2 #4000 #80000000 #A

3.3 Back-End

- Back-End PCI is a module with connection to a 64 Bit PCI bus.
- Function is based on the XILINX Virtex Pro FPGA XC2VP7 FF 672 .
- A PCI master/slave interface with scatter/gather DMA is implemented.
- The slave IF is used for communication.
- The master IF is used for video data DMA transfers to PCI.
- RocketIO transceivers (2.5 GBit each) are used for communication and data transfers

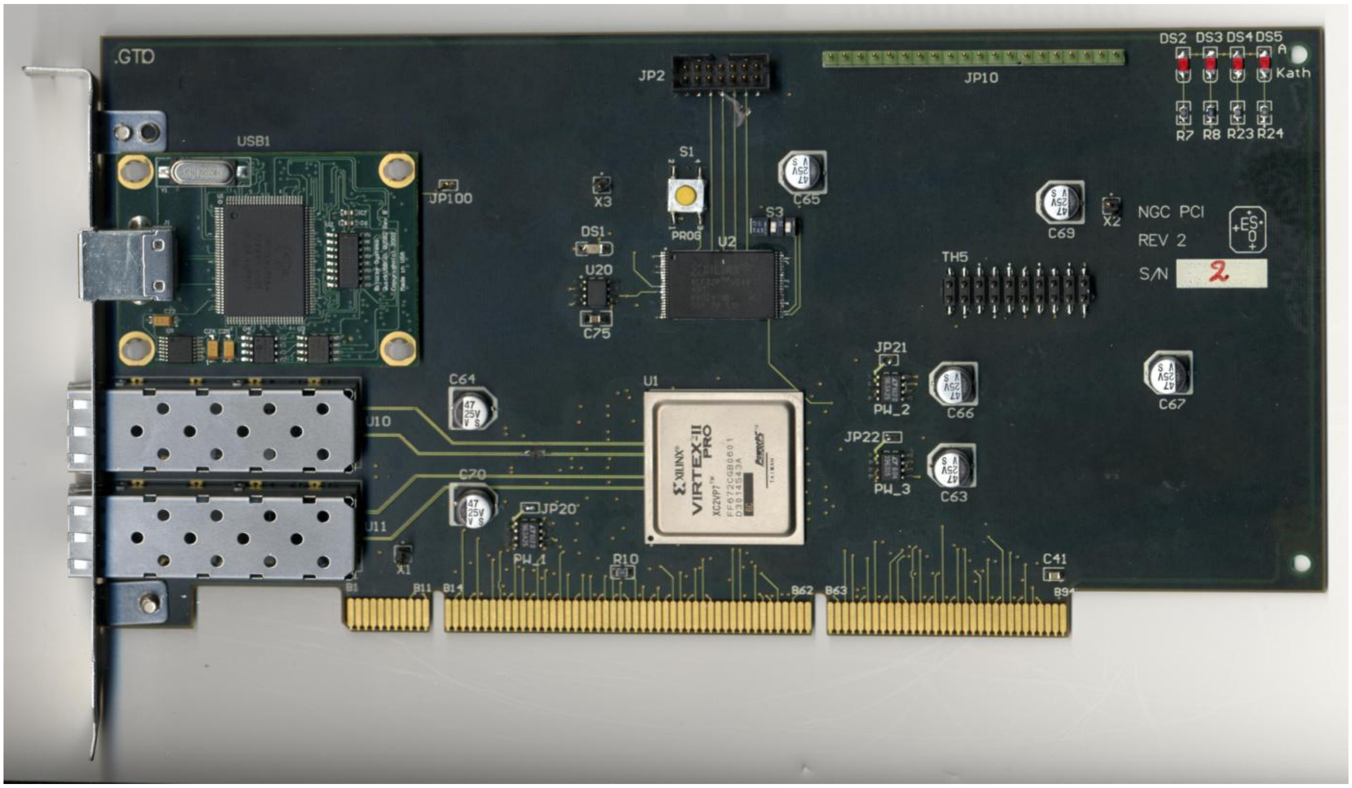


Figure 5 PCI Back-End

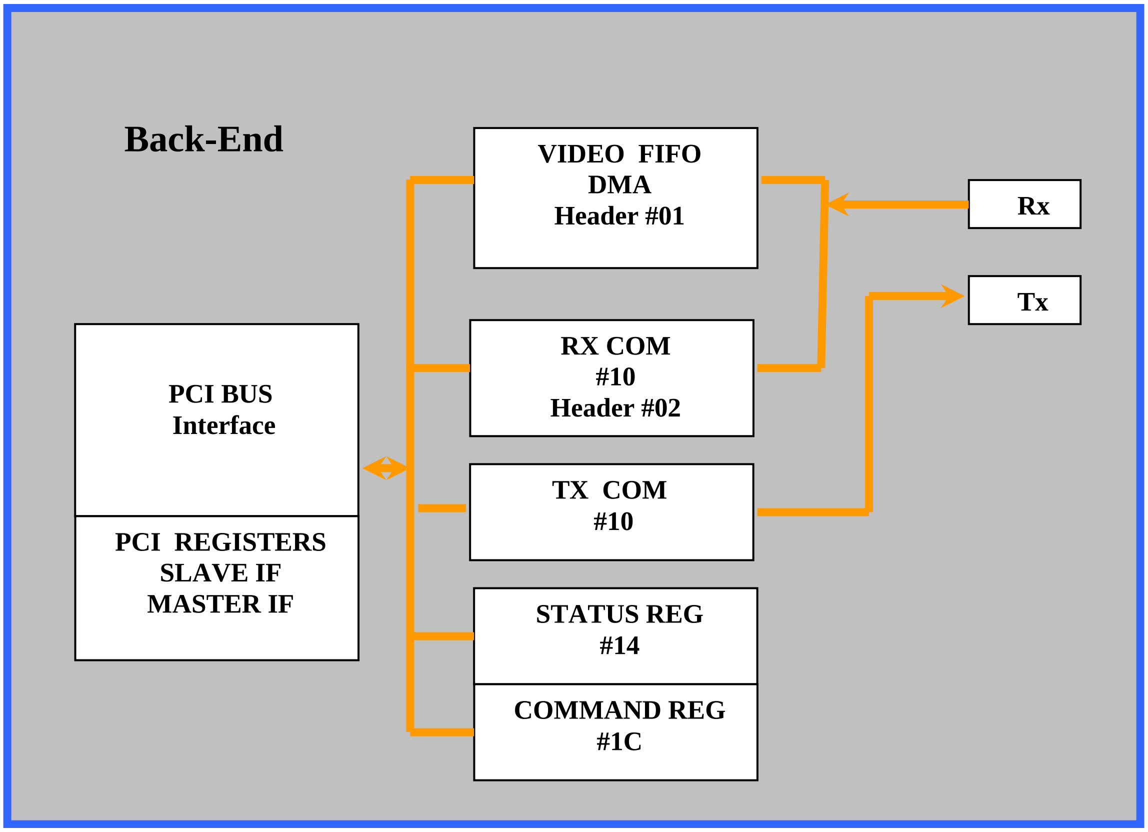


Figure 6 Back-End Structure

The Back-End module (Figure 5 and Figure 6) connects to the data acquisition computer via a 64 bit PCI bus interface based on a XILINX IP core. The PCI bus interface has master and slave capabilities. The slave forms the communication interface to the Front-End, the master is responsible for DMA transfers of the Front-End video data to the acquisition computers memory. Commands and data transfers can run concurrently. The communication between Back-End and any Front-End module is based on packet transmission over high speed serial links. Data packets for communication have to be written to TX COM, read data packets from the Rx link are routed into RX COM. The COMMAND REGISTER initiates actions like fifo clear or transmission start. The STATUS REGISTER holds status information like fifo status or the transfer acknowledge bit. Video data from the link are automatically routed to the VIDEO FIFO.
A transfer handshake protocol must always be followed for communication operations.

- **The packet has to be written from PCI to the Back-End transmitter fifo (address #10).**
- **Then the transmission has to be initiated by writing from PCI to the command register (Write h10 to address #14).**
- **Then the Acknowledge register (address #14) has to be polled till acknowledge is received (Bit 7 set).**

- **Acknowledge register Bit 0 declares a finished operation, Bit 1 successful operation on a valid address.**
- **Acknowledge register is cleared by reading the receiver fifo (address #10)**

Addresses

Table 1 Configuration Space

Relative address[Hex] Config Space		Function
C W		DMA Control Register
10 R/W		Read Rx FiFo / Write Tx Fifo
14 R		COMmunication Status Register
1C W		COMmunication Command Register
68 R/W		PCI_Descr_Pointer #90 ADI
84 R/W		PCI_Addr_Reg
8C R/W		PCI_ DMA_Counter_Reg
90 R/W		PCI_Descr_Pointer
A0 R		PCI_Board Revision and Date

Hint : Read from NGC Panel with " ior local 0x<rel addr> "

Table 2 DMA Space

Relative address[Hex] DMA Space		Function
0		------
8 R		DMA Status Register

Registers

Table 3 DMA Status Register – Read (Addr #8)

Bit 5	**Data FiFo Empty**
Bit 6	**Data FiFo Full**
Bit 7	**Data FiFo Full – Write (Cleared by Clr Data FiFo)**

Table 4 Read Rx FiFo / Write Tx FiFo (Addr #10)

Bit 0..31	**Com Data**

Table 5 Communication Command – Write (Addr #1C)

BIT 0	**Clear Com FiFos**
Bit 1	**Clear Video FiFo**
Bit 4	**Transfer Enable TX**
Bit 7	**Test Box Enable**

Table 6 Communication Status – Read (Addr #14)

BIT 0	**Ack Received**
Bit 1	**Valid Address accessed**
Bit 5	**Video FiFo Empty**
BIT 6	**Video FiFo Full**
Bit 7	**Overflow Error Flag (Write on Video FiFo Full - Cleared by Clr Video FiFo)**
Bit 8	**RX FIFO Empty**
Bit 9	**RX FIFO Full**
Bit 10	**TX FIFO Empty**
Bit 11	**TX FIFO Full**
Bit 12	**Link Channel Up**
Bit 13	**Link Hard Error**
Bit 14	**Link Soft Error**
Bit 15	**Link Framing Error**

Table 7 Interrupt_Ctr_Reg (Addr #68)

Bit 21	**Interrupt Flag**

Table 8 PCI_Addr_Reg (Addr #84)

Bit 0..31	**DMA Address**

Table 9 PCI_DMA_Counter (Addr #8C)

Bit 2..12	**DMA Count**

Table 10 PCI_Descr_Pointer (Addr #90)

Bit 4..31	**Initial DMA Descriptor**

Table 11 PCI Backend Board ID Register (Addr #A0)

PCI Backend HW Revision 2
Firmware Revision 1.1)

Bit (3 .. 0) Board Type	**Value = 1 → Basic Board** **2 → AQ32** **8 → PCI Backend**
BIT 0	**0**
Bit 1	**0**
Bit 2	**0**
BIT 3	**1**
Bit (7 .. 4) Board Sub-type	**Value = 1 33MHz Latency disabled**
Bit 4	**1**
Bit 5	**0**
Bit 6	**0**
Bit 7	**0**
Bit (11 .. 8) HW Revision	
Bit 8	**0**
Bit 9	**1**
Bit 10	**0**
Bit 11	**0**
Bit (15 .. 12) Firmware Revision	
Bit 12	**1**
Bit 13	**0**
Bit 14	**0**
Bit 15	**0**
Bit (19 .. 16) Firmware Sub-Revision	
Bit 16	**1**
Bit 17	**0**
Bit 18	**0**
Bit 19	**0**
Bit 31 .. 20	**reserved**

Table 12 DMA Command Register – Write only (Addr #A8)

BIT 1	**Start DMA**
Bit 2	**Abort DMA**
Bit 3	**Clear Interrupt**

3.4 Front-End

The Front-End is connected to the Back-End with an Upstream link. A Downstream link on the first Front-End module connects to a downstream module like AQ32 or other basic modules (more clocks and biases or multiple detectors). From there again a link connects downstream to the next module.

Before any addressing of functions on Front-End modules the link structure of the Front-End system must be defined. This happens by writing to the CONFIG register of each module.

RX_COM together with TX_COM form the communication interface on the Front-End. The upstream link send all set-up and command information for the onboard functions to RX_COM, where address and data are extracted and send to the individual on board modules - replies from the addressed modules enter TX_COM for uplink transmission.

The module structure is the same for all Front-End modules (FEB shown in Figure 8, AQ32 in Figure 34), only the set of registers differs slightly between FEB and AQ32.

3.4.1 Front-End-Basic Board (FEB)

The Front-End Basic Module (Figure 7) is based on the XILINX Virtex Pro FPGA XC2VP7 FF 672. Main functions of this module are

- **Communication**
- **Video data transfer**
- **Sequencer**
- **DAC voltage generator for clock and bias**
- **Clock drivers (18 clocks)**
- **Bias drivers (18 biases)**
- **Four data acquisition channels (can be installed with either 16 or 18 Bit ADC's)**
- **Telemetry**
- **Clock monitoring**
- **Video monitoring**

- Communication and data transfer to the Back-End is handled with the FPGA's RocketI/O transceivers.
- The sequencer is completely contained within the FPGA. The digital clock driver lines of the sequencer connect without glue logic to the clock driver switches.
- The ADC outputs of the four acquisition channels connect without glue logic to the FPGA due to the high pin count available there. Used ADC's are the AD76xx types from Analog Devices. The preamplifier input is fully differential (see Table 39 on next page).
- Connection to additional multi channel AQ modules is over the backplane by copper with the high speed links of the FPGA.
- Telemetry of biases and clocks.
- Two independent Monitors for clocks.
- Monitor for video signals.

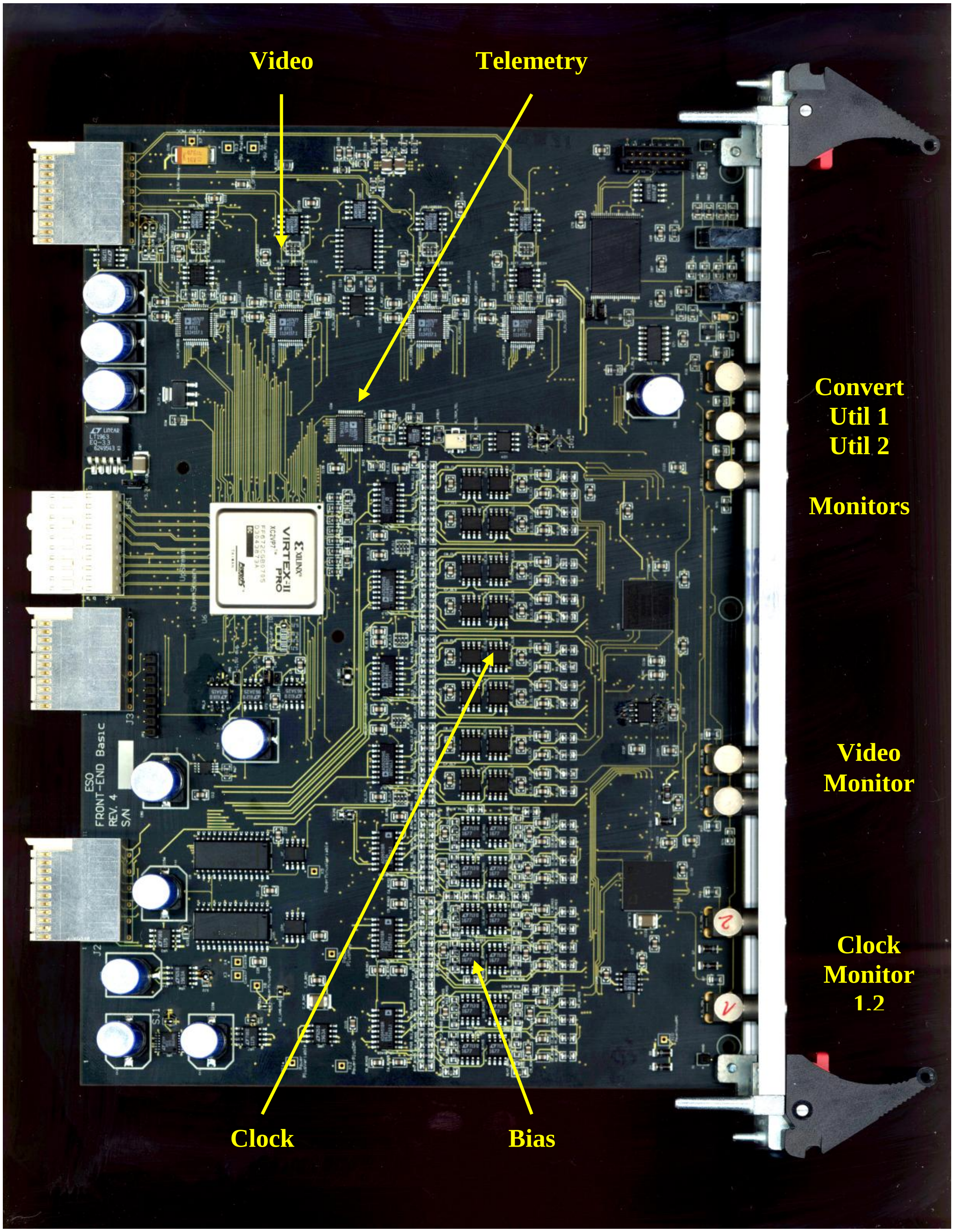


Figure 7 **Front-End Basic Module**

The module structure of the Front-End-Basic Board is shown below (Figure 8 Front-End Basic Module Structure). The corresponding module blocks are described in the following sections.

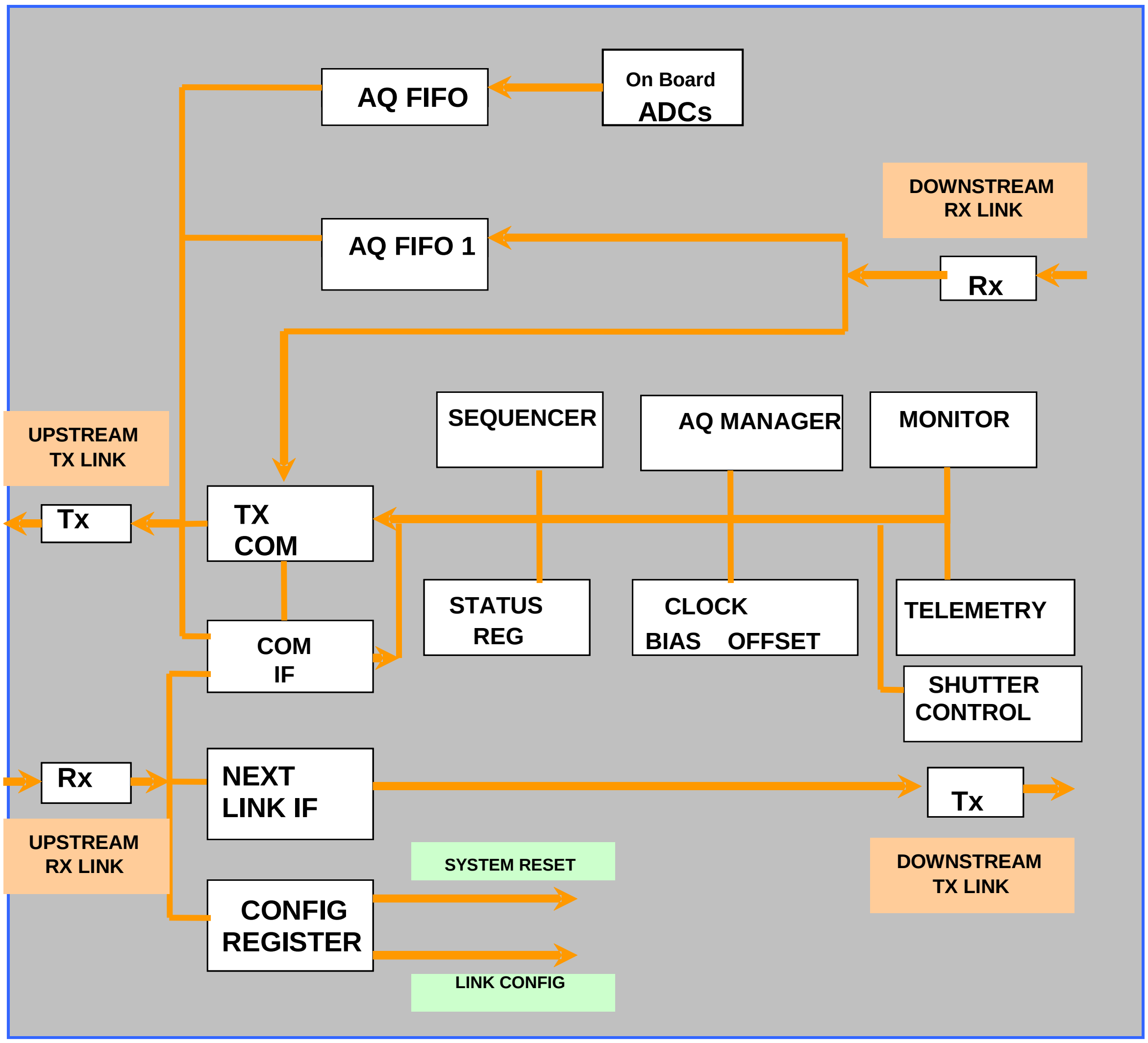


Figure 8 Front-End Basic Module Structure

3.4.1.1 Configuration Register

Each NGC Front-End board has a Configuration Register. Before any addressing of functions on Front-End modules the link structure of the Front-End system must be defined. This is accomplished in the Front-End Link Config Register(s) [Bit 3..0]. It is the only register on the Front-End where no handshake signals are generated, just because without structure definition no reply is possible. These registers (for each board) are the first ones to set in any system set-up. The registers are addressed directly by header addressing in an order that the modules next to the Back-End have to be programmed first.
A general module reset [Bit 15] similar to power up can also be executed by this register.

Remark on link direction : Uplink on a NGC Front-End board has the direction from module towards Back_End, Downlink direction correspondingly away from Back_End.

Table 13 Configuration Register **(Header 0X8)**

BIT 3..0	**Number of UpStream links till Back-End (Minimum is 1)**
BIT 5..4	**Not used**
BIT 6	**SidelinkNotUplink** **Closes the Uplink and switches to the Sidelink on the Transition Board for communication and data transfer (1)**
BIT 7	**Downlink_Powerdown** **Unpowers the Downlink of this module**
BIT 14..8	**Not used**
Bit 15	**Global Reset (pulsed)** **Resets module to power up condition and puts global reset to backplane what resets all boards connected to the backplane.** **(Not affected are sequencer memories and bias/clock voltage settings)**

(1) The sidelink option can also be set by a hardware jumper(see Figure 22). When the jumper is set, BIT 6 of the configuration register is a “don’t care”.

Example :

Module1 (e.g. Basic Module) connected to Back-End
= 1 upstream link on Module1
Module2 (e.g. first AQ32) connected to Basic module
= 2 upstream links for Module2
Module3 (e.g. second AQ32) connected to Module2
= 3 upstream links for Module3
Module4 (e.g. second Basic Module) connected Module3
= 4 upstream links for Module4

Packet Data for the Front-End Configuration Register :

0X8 0x1	**(Basic module)**
0x5 0x8 0x2	**(first AQ32 module)**
0x5 0x5 0x8 0x3	**(second AQ32 module)**
0x5 0x5 0x5 0x8 0x4	**(second Basic module)**

3.4.1.2 Status and ID Register

Front-End status contains two read only registers. The status register contains the module status information. The ID register describs module set-up.

Table 14 Status Register – Read only ADDR #1000

BIT 0	**UPSTREAM_CHANNEL_UP**
Bit 1	**SIDESTREAM_CHANNEL_UP**
Bit 2	**DOWNSTREAM1_CHANNEL_UP**
BIT 3	**UPSTREAM_HARD_ERROR**
Bit 4	**SIDESTREAM _HARD_ERROR**
Bit 5	**DOWNSTREAM1_HARD_ERROR**
Bit 6	**UPSTREAM_SOFT_ERROR**
Bit 7	**SIDESTREAM _SOFT_ERROR**
Bit 8	**DOWNSTREAM1_SOFT_ERROR**
Bit 9	**UPSTREAM_FRAME_ERROR**
Bit 10	**SIDESTREAM _ FRAME_ERROR**
Bit 11	**DOWNSTREAM1_ FRAME_ERROR**
Bit 12	**Not used**
Bit 13	**Not used**
Bit 14	**Not used**
Bit 15	**Not used**
Bit 16	**Not used**
Bit 17	**Not used**
Bit 18	**Not used**
Bit 19	**Not used**
Bit 20	**RX_COM_FIFO_EMPTY**
Bit 21	**RX_COM_FIFO_FULL**
Bit 22	**TX_AQ_FIFO1_EMPTY**
Bit 23	**TX_AQ_FIFO2_EMPTY**
Bit 24	**TX_AQ_FIFO1_FULL**
Bit 25	**TX_AQ_FIFO2_FULL**
Bit 26	**TX_COM_FIFO_FULL**
Bit 27	**TX_COM_FIFO_EMPTY**
Bit 28	**NEXT_LINK_FIFO_FULL**
Bit 29	**NEXT_LINK_FIFO_EMPTY**
Bit 30	**OUTPUT ENABLED**
Bit 31	**SEQUENCER RUNNING**

Table 15 ID Register – Read only **ADDR #1002 (Below example Basic Board :**
18Bit ADC HW Revision 3
Firmware Revision 4.1)

Bit (3 .. 0) Board Type	**Value = 1 → Basic Board** **2 → AQ32**
BIT 0	**1**
Bit 1	**0**
Bit 2	**0**
BIT 3	**0**
Bit (7 .. 4) Board Sub-type	**Value = 1 to 4 → 16Bit ADC /1MHz Type 1..4** **5 to 8 → 16Bit ADC /1MHz Type 1..4** **9 to 12 → 16Bit ADC /3MHz Type 1..4** **13 to 15 → 18Bit ADC /1MHz Type 1..4**
Bit 4	**Set by external Jumpers**
Bit 5	**Set by external Jumpers**
Bit 6	**Set by external Jumpers**
Bit 7	**Set by external Jumpers**
Bit (11 .. 8) HW Revision	
Bit 8	**1**
Bit 9	**1**
Bit 10	**0**
Bit 11	**0**
Bit (15 .. 12) Firmware Revision	
Bit 12	**0**
Bit 13	**0**
Bit 14	**1**
Bit 15	**0**
Bit (19 .. 16) Firmware Sub-Revision	
Bit 16	**1**
Bit 17	**0**
Bit 18	**0**
Bit 19	**0**
Bit 31 .. 20	**reserved**

The software reads the ID register, to determine that the correct board is installed in the system. The Board Type tells about what board is installed, Basic Board, AQ32 or other TBD.
The Board Sub-type is set by external jumpers to accomplish the different board set-up modes for the specific board (see also Figure 22). The values in ID Register Bit 7..4 determine:

Table 16 ID Register - Board Sub-Type

Board Sub-Type Value Bit (7 .. 4)	**Set-up**
0	Not used
1 CCD (Optical) Application Standard Setting	16 Bit ADC/ 1MHz Clock Low/Hi Range : -10V/ 10V Bias 1- 16 Range : 0V to 28V Bias 17- 20 Range : 10V to -10V
2 CCD Application	16 Bit ADC/ 1MHz Clock Low/Hi Range : -10V/ 10V Bias 1- 8 Range : 0V to 28V Bias 9- 16 Range : -5V to 28V Bias 17- 20 Range : 10V to 10V
3 CCD Application	16 Bit ADC/ 1MHz Clock Low/Hi Range : -10V/ 10V Bias 1- 8 Range : 0V to 10V Bias 9- 16 Range : -10V to 10V Bias 17- 20 Range : -10V to 10V
4	16 Bit ADC/ 1MHz TBD
5 CMOS (IR) Application	16 Bit ADC/ 1MHz Clock Low/Hi Range : -6V / 6V Bias 1- 8 Range : 0V to 6V Bias 9- 16 Range : 0V to 6V Bias 17- 20 Range : 0V to 6V
6,7,8	16 Bit ADC/ 1MHz TBD
9 CCD (Optical) Application	16 Bit ADC/ 3MHz Clock Low/Hi Range : -10V/ 10V Bias 1- 16 Range : 0V to 28V

	Bias 17- 20 Range : 10V to -10V
10 CCD Application	16 Bit ADC/ 3MHz Clock Low/Hi Range : -10V/ 10V Bias 1- 8 Range : 0V to 28V Bias 9- 16 Range : -5V to 28V Bias 17- 20 Range : 10V to 10V
11 CCD Application	16 Bit ADC/ 3MHz Clock Low/Hi Range : -10V/ 10V Bias 1- 8 Range : 0V to 10V Bias 9- 16 Range : -10V to 10V Bias 17- 20 Range : -10V to 10V
12	16 Bit ADC/ 3MHz TBD
13 CCD Application	18 Bit ADC/ 1MHz Clock Low/Hi Range : -10V/ 10V Bias 1- 16 Range : 0V to 28V Bias 17- 20 Range : 10V to 10V
14 CCD Application	18 Bit ADC/ 1MHz Clock Low/Hi Range : -10V/ 10V Bias 1- 8 Range : 0V to 28V Bias 9- 16 Range : -5V to 28V Bias 17- 20 Range : 10V to 10V
15 CCD Application	18 Bit ADC/ 1MHz Clock Low/Hi Range : -10V/ 10V Bias 1- 8 Range : 0V to 10V Bias 9- 16 Range : -10V to 10V Bias 17- 20 Range : -10V to 10V

The HW Revision in Bit (11 .. 8) determine the board hardware revision. The Firmware Revision and the Firmware Sub-Revision are in Bit 15 to 12 and Bit 20 to16.

3.4.1.3 Monitor

The Monitor module sets the clock and the video monitors to the channels chosen in the corresponding registers and routes the signals to the front panel Lemo connectors.

There is one register for the video channel and two registers for the clocks.

Remark:
Clock 17 and 18 can only be monitored via the onboard test points

Table 17 Video Monitor Register – Write ADDR #B000

BIT 1 .. 0	**Video Channel 1 to 4**

Table 18 Clock Monitor1 Register – Write ADDR #B001

BIT 3 .. 0	**Clock Channel 1 to 16**

Table 19 Clock Monitor2 Register – Write ADDR #B002

BIT 3 .. 0	**Clock Channel 1 to 16**

3.4.1.4 Detector Bias Generation

Detector Bias Generation is responsible for programming the voltages of clocks and biases. The module contains two registers. The lower 14 Bits of the Bias Register set the bias value, the next five Bits the DAC channel number, Bit 21 selects the DAC chip.

Physically there are two 32channel DAC's for the voltages of clocks and biases on the board. DAC chip1 and channel 1 and 2 of DAC2 are for the 18 clocks , DAC chip2 channel 5 to 24 for the 20 biases.

Bit 31 programs an offset common to all DACs on the selected DAC chip (Bit 21). For the offset no DAC channel number is required.

To make three voltages independent of the offset setting a special scheme is used for them. The offset of the video amplifier is set with the difference of DAC chip2 channel 27 and DAC chip2 channel 28. Detector preamplifier bias is set with the difference of DAC chip2 channel 29 and DAC chip2 channel 30. A detector diode bias is set with the difference of DAC chip2 channel 31 and DAC chip2 channel 32.

Table 20 Bias Set-up Register – Write ADDR #8000

BIT 13 .. 0	Data Value
Bit 21 .. 16	DAC Channel Number **Clocks** Clock_1 _low = 0 Clock_1_high = 1 Clock_2 _low = 2 Clock_2_high = 3 . . Clock_16 _low = 1E Clock_16_high = 1F Clock_17 _low = 20 Clock_17_high = 21 Clock_18 _low = 22 Clock_18_high = 23 **Biases** DC_Bias_1 = 24 . . DC_Bias_7 = 2A DC_Bias_12 = 30 DC_Bias_20 = 38 Video_Offset_P = 3A Video_Offset_N = 3B Preamp Bias_P = 3C Preamp Bias_N = 3D Diode Bias P = 3E Diode Bias N = 3F
Bit 31	Offset Select

The binary value in BIT 13 .. 0 (**V_DATA_VALUE** or **V_OFFSET_VALUE**) is converted to a DAC output voltage corresponding to the formula below :

V_OUT = 0.001259 * V_DATA_VALUE – 0.001076 * V_OFFSET_VALUE + Individual OFFSET

> Individual OFFSET is the offset introduced by DAC and Buffer Amplifier (~ 100mV) on each channel individually – can be masked out by software for each channel on the board individually

The Control register enables the clock and bias outputs to the detector, bit 15 resets all Biases to Zero Volts.

Table 21 Bias Control Register – Write ADDR #8001

BIT 0	**Enable Bias and Clock Outputs**
Bit 15	**Reset all Biases to Zero Volts**

3.4.1.5 Telemetry

Telemetry reads the voltage of clock levels and biases and digitizes with 16 Bit accuracy. The user has to write the channel address, this issues automatically after a delay the conversion command and writes the telemetry adc data to a register. After the delay (~1ms) the data are ready for read. This is accomplished by a read the telemetry register.

Channel 0 to 35 read the clock levels (low, high) after a series resistor of 27 Ohms.
Channel 36 to 55 read the Bias levels after a series resistor of 100 Ohms.

Clock and Bias current measurements can be carried out by reading a voltage one time with output enable on and the other measure with enable off . Current can be calculated by dividing the voltage difference by the series resistor.

Table 22 Telemetry Register – Write ADDR #A000

BIT 5 .. 0	Channel address (Hex) Clock_1 _low = 0 Clock_1_high = 1 Clock_2 _low = 2 Clock_2_high = 3 . . Clock_16 _low = 1E Clock_16_high = 1F Clock_17 _low = 20 Clock_17_high = 21 Clock_18 _low = 22 Clock_18_high =23 DC_Bias_1 = 24 . . DC_Bias_16 = 2F . DC_Bias_20 = 30

Table 23 Telemetry Register – Read ADDR #A000

BIT 15 .. 0	**Telemetry data (305.2 uV/ADU)**

To accommodate the range of the telemetry ADC (+/- 10V on 16Bit) the bias voltages are subdivided by a factor of 3.

3.4.1.6 Sequencer

The Sequencer (see principle in Figure 9) generates the clock patterns for the readout of the detector. They are transformed to analogue clocks with the voltage settings of the Clock and Bias module.

3.4.1.6.1 Principle

The Pattern Ram (2048 x 64) is loaded with the clock patterns. Any pattern length between 1 and 2048 is possible. Pattern Ram Low holds Bit 31 downto 0, Pattern Ram High holds Bit 63 downto 32. The dwell time of a pattern word is included in the word as well as special function bits (see pattern ram description).
The Sequencer Ram (2048 x 32) holds the Seq Code Bits, the pattern address and the pattern repetition count. The Seq Code Bits (see Seq Ram description) feed the Seq Code Interpreter. He decides if a pattern address is written into the Pattern Address Fifo and with the repetition count (16 bit) how often it is executed.
The Pattern Address Fifo contains the start address of the pattern supplied by the Sequencer Ram. The time counter loaded from the Pattern Ram determines the Dwell Time in steps of 10ns (16 bit – one bit = 10ns, minimum dwell time value = 2) of a pattern. The pattern address counter is incremented each time the time count is reached. When the End of Pattern is present (bit 31) in the pattern word, the next word with the start address of the next pattern is read from the FiFo.
The sequencer starts with the Sequencer Start Command (Command Register Bit 0). The first pattern is output after the code interpreter has written eight words into the pattern address FiFo. The sequencer stops after the dwell time of a pattern when

1. the pattern address FiFo is empty (No sequence loaded or Empy Read happened)
2. a breakpoint in a pattern (bit 29 high word) is detected and the Sequencer Stop Command (Command Register Bit 1) was executed before (programmed end of sequence – option 1)
3. the Program End Bit (bit 30 high word) in a pattern is detected (programmed end of sequence – option 2)
4. Sequencer Reset Command (Command Register Bit 15) is executed (immediate stop)

In case 1,2,3 the status, counters, controls are as at stop time, so to start a new sequence or to restart the old one, a Sequencer Reset Command has to be executed before.

If in a pattern word the Wait for Trigger (bit 28 high word) is set, the sequencer stops after the dwell time of this pattern and waits for a trigger signal, which may be set from external inputs or software (Bit 7 Command Register).
See in detail Chapter 3.4.1.7 NGC Trigger and Shutter Control.

There are two trigger modes. In the direct mode the trigger signals are routed from the Transition board directly to the sequencer state machine (see chapter 3.4.1.6). In shutter mode the shutter state machine sends the trigger signals to the sequencer state machine. The trigger mode (Bit 7 of Sequencer Command Register) determines the trigger mode.

The Sequencer Status register (see Sequencer Status register description) contains status information about code interpretation and sequence termination. This information is cleared with the Sequencer Reset command. FiFo status is always available.

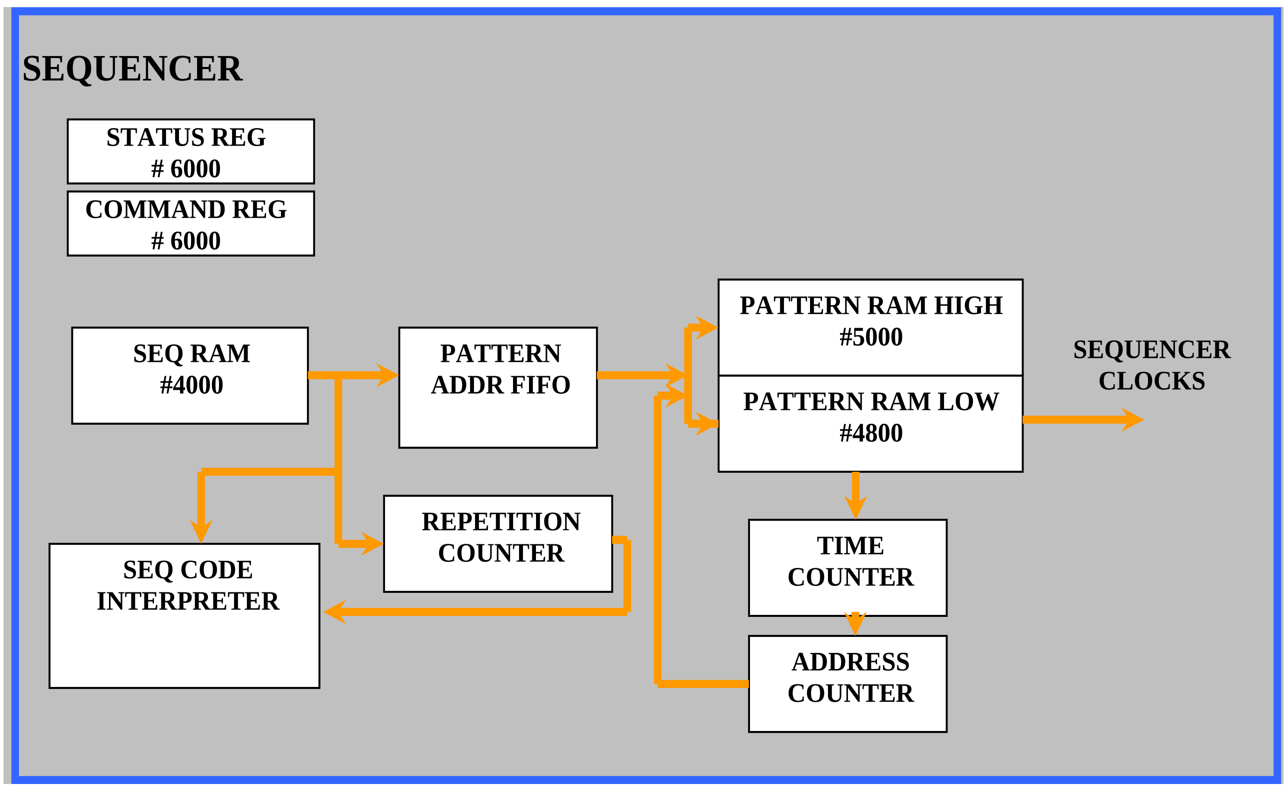


Figure 9 Sequencer Block

Table 24 Sequencer Command Register – Write ADDR = #6000

BIT 0	**Sequencer Run (Pulsed)**
BIT 1	**Sequencer Run Disable (used if more than one SEQ are installed in a backplane - only one can be active)**
BIT 2	**Reserved**
BIT 3	**Demand Exit at next Breakpoint**
BIT 6	**Enable Trigger Mode**
BIT 7	**Software Trigger (Pulsed)**
BIT 8..14	**Reserved**
BIT 15	**Sequencer Stop and Reset (Exit immediate)**
BIT 16..31	**Reserved**

Table 25 Status Register – Read ADDR = #6000

BIT 0	**Code Interpretation Running**
BIT 1	**Sequencer Running**
BIT 2	**Sequencer Waiting for Trigger**
BIT 3	**Acquisition Paused**
BIT 4	**End of Program Signal (cleared after Sequencer Run)**
BIT 5	**End at Breakpoint Signal (cleared after Status Read)**
BIT 6	**Fifo Empty**
BIT 7	**Fifo Empty Read (cleared after Sequencer Run)**
BIT 8	**Fifo Full**
BIT 9..15	**Reserved**
BIT 26..16	**Pattern Ram Address**
BIT 17..31	**Reserved**

Table 26 Seq RAM **ADDR = #4000 till #47FF**

BIT 0 .. 10	**Pattern Start Address**
BIT 11 .. 26	**Pattern Repetition Count**
BIT 27	**reserved**
BIT 28 .. 30	**Sequence Code**
BIT 31	**reserved**

Table 27 Pattern RAM Low **ADDR = #4800 till #4FFF**

BIT 0 .. 31	**Clock 1 ..32 (presently 18 Clocks used)**

Table 28 Pattern RAM High **ADDR = #5000 till #57FF**

BIT 0	**Convert 1 (Sequencer Clock 33)**
BIT 1	**Convert 2 (Sequencer Clock 34)**
BIT 2	**Sequencer Clock 35 (Output on Marker Lemo 1)**
BIT 3	**Sequencer Clock 36 (Output on Marker Lemo 2)**
BIT 4	**Utility Signal 1 (Routed to Opto Outputs)**
BIT 5	**Utility Signal 2 (Routed to Opto Outputs)**
BIT 6	**reserved**
BIT 7	**Sequencer BIT (7 .. 0) Write Disable**
BIT 8	**Sequencer BIT (15 .. 8) Write Disable**
BIT 9	**Sequencer BIT (23 .. 16) Write Disable**
BIT 10	**Sequencer BIT (31 .. 24) Write Disable**
BIT 11	**reserved**
BIT 12 .. 27	**Pattern Dwell Time (Value x 10ns)** **Minimum Dwell Time Value = 2**
BIT 28	**Wait for Trigger (causes Sequencer Wait for Trigger)**
BIT 29	**Breakpoint (causes Sequencer to stop at this pattern after SEQ STOP command - Routed to Status)**
BIT 30	**End of Program (Routed to Status Register for inspection of Sequence Termination)**
BIT 31	**End of Pattern (causes fetch of next Pattern after Dwell Time of Pattern)**

Table 24 lists the sequencer codes and the code interpretation times. Because the sequencer code is continuously interpreted and pattern start addresses are written to the pattern fifo, care must be taken that the fifo does not empties, when code interpretation time is longer than code execution times determined by the read speed vectors.

Table 29 Sequencer Codes and Interpretation Time

Code	Function	Code Interpretation Time [Clock Cycles]
000	**Stop Pattern Interpretation**	**-**
001	**EXEC Pattern (Number of Pattern, Number of Repetitions)**	**2 + 2 * number of repetitions**
010	**LOOP (Number of Repetitions)**	**6**
011	**LOOP END**	**8**
100	**LOOP INFINITE**	**6**
101	**JUMP SUBROUTINE (Address)**	**3**
110	**RETURN SUBROUTINE**	**2**
111	**Stop Pattern Interpretation**	**-**

3.4.1.7 NGC Trigger and Shutter Control

(See also NGC. Shutter and TIM Interface).

A NGC system can run in triggered mode and is also able to control a shutter. The operation for IR and optical systems (Shutter control) are slightly different, but run with the same Sequencer State Machine. To enable triggered mode Bit 6 of the *Sequencer Command Register* (Address #6000) must be set. The wait for trigger time point is determined with the WAIT FOR TRIGGER BIT (PATTERN RAM HIGH Bit 28).

3.4.1.7.1 NGC Trigger – IR Operation

Figure 10 shows the connections on NGC triggered operation in IR.

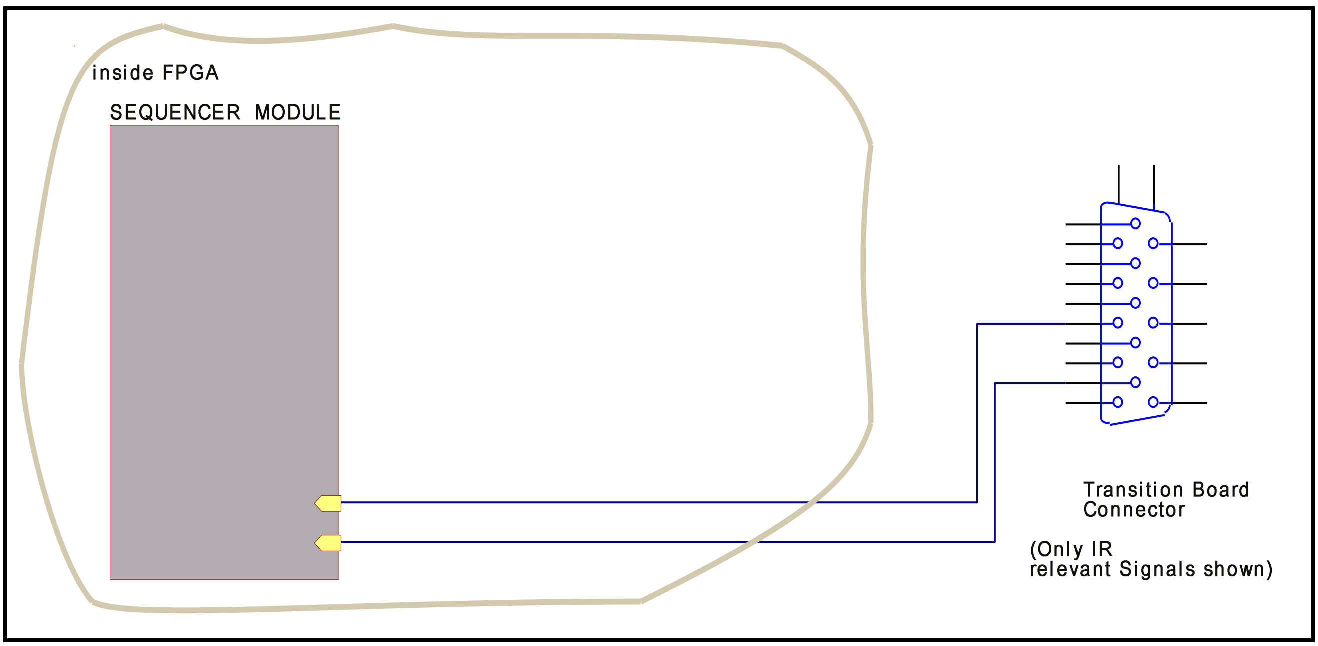


Figure 10 Connection Diagram NGC triggered operation in IR

For wobbling secondary mirror applications NGC triggers on the signals Trigger Phase-A and Trigger Phase-B. They might be derived from the standard ESO-TIM module to trigger at an absolute time. These two inputs are located on the Transition Board HD-15 connector (see chapter 3.4.2) and are routed galvanically isolated to the FPGA on the Basic Board. Figure 11 shows the typical trigger signals of Trigger Phase-A and Trigger Phase-B.

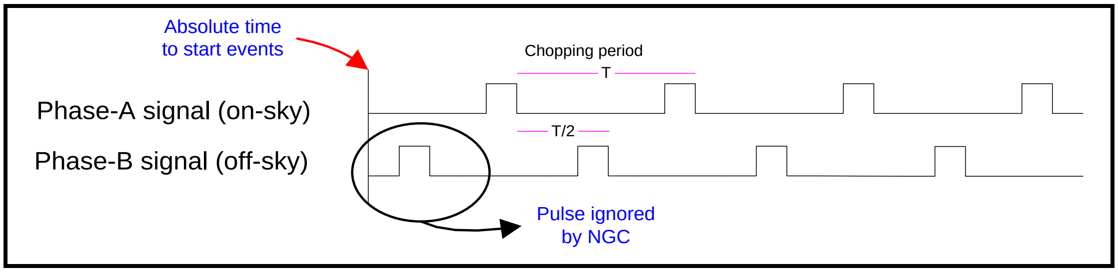


Figure 11 **Connection of NGC to TIM and TIM clock signals**

In the event that a pulse on phase-B is generated before the one on phase-A, the firmware on NGC will ignore the pulse on phase-B. Therefore, the first event executed will always be upon the reception of the pulse on phase-A.
When only one trigger signal is needed or available, the two trigger inputs might be connected together. Then the sequencer is triggered on each event.

A state machine controls the operation of the sequencer. A state diagram is shown in Figure 12. After RESET the sequencer is in STATE 1, the IDLE state.
After a RUN SEQUENCER command STATE 2 is entered and leads to the execution of the loaded sequence and running of the corresponding clock patterns. In the most simple case, free run, code is executed till a STOP CONDITION is valid and the state machine returns to IDLE.
For triggered operation the TRIGGER ENABLE [Command register, Bit 6] must be set.
The state machine enters STATE 3, and waits for a trigger, if a WAIT FOR TRIGGER [PATTERN RAM HIGH, Bit 28] is encountered . The trigger signal in STATE 3 can be a software trigger [Command Register, Bit 7] or the TRIGGER PHASE A signal from the Basic Board Utility connector on the Transition Board.
After the trigger, the machine enters STATE 4 and continues pattern execution. Code is executed till either a STOP CONDITION is valid and the state machine returns to IDLE state or a WAIT FOR TRIGGER [PATTERN RAM HIGH, Bit 28] is found. Then the state machine enters STATE 5 and waits for a trigger signal. The trigger signal in STATE 5 can be a software trigger [Command Register, Bit 7] or the TRIGGER PHASE B trigger signal from the Transition Board of the Basic Board Utility connector.

Comment : Two different triggers are needed for IR operation with wobbling secondary to start the measurement always in the same mirror position. If mirror position is not an issue the two trigger inputs can be wired together.

After the trigger the machine enters STATE 6 and continues pattern execution. Code is executed till either a STOP CONDITION is valid and the state machine returns to IDLE or a WAIT FOR TRIGGER (PATTERN RAM HIGH Bit 28) is found. Then the state machine enters STATE 3 and waits for a trigger signal. The trigger signal in STATE 3 can be a software trigger [Command Register, Bit 7] or the TRIGGER PHASE B signal from the Transition Board of the Basic Board Utility connector.

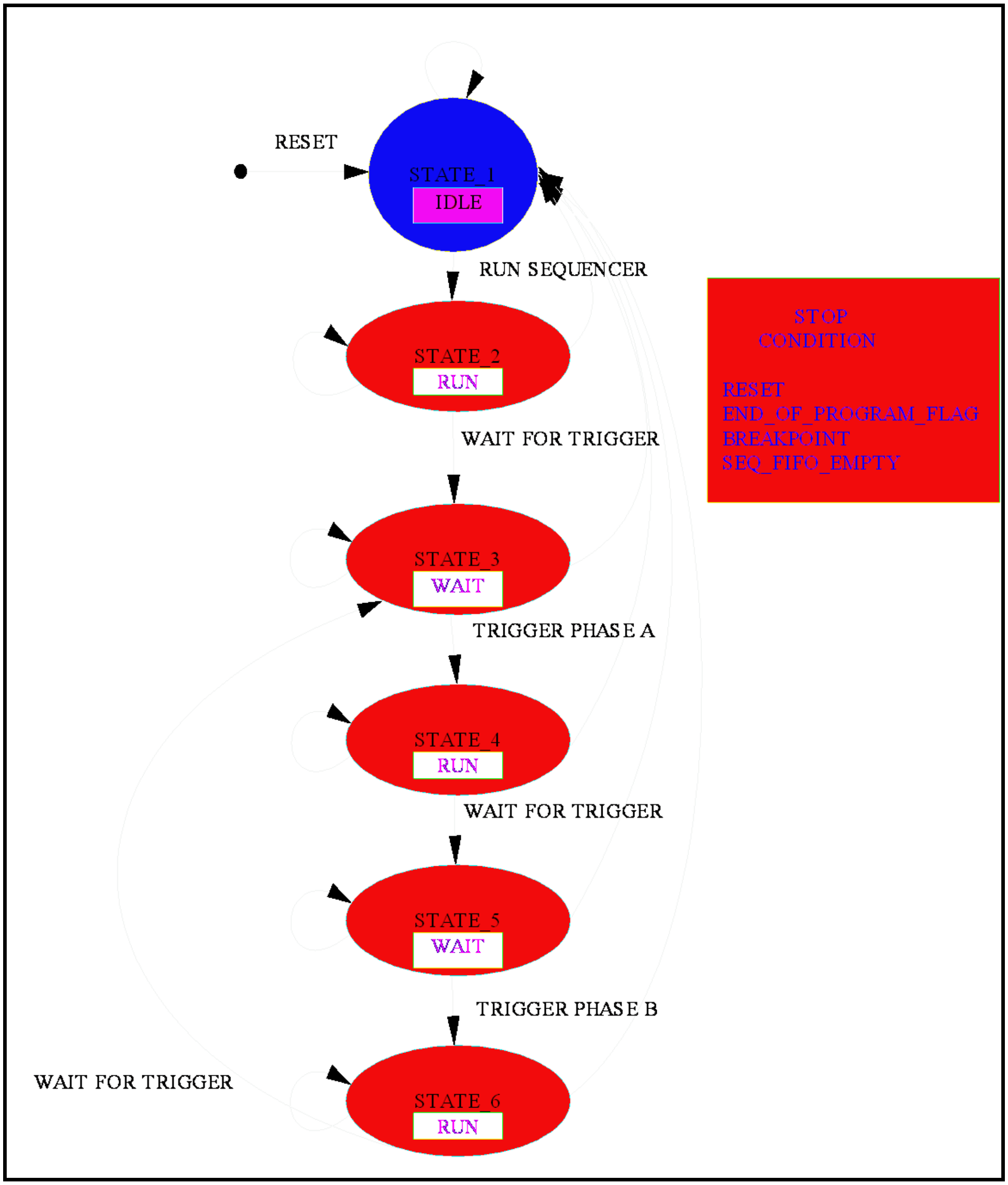


Figure 12 Sequencer State Diagram - IR

A BREAKPOINT [PATTERN RAM HIGH Bit 29] can be set anywhere in the clock patterns and returns the state machine to IDLE state, when the DEMAND EXIT AT NEXT BREAKPOINT [Command Register, Bit 7] is set.

STOP CONDITIONs

RESET (stops immediately)	
GLOBAL RESET	[Front-End Config Register (Header 0X8)]
SEQUENCER RESET	[Command Register [ADDR = #6000], Bit 7]
END OF PROGRAM FLAG	[PATTERN RAM HIGH Bit 30]
BREAKPOINT	[PATTERN RAM HIGH Bit 29]
SEQ FIFO EMPTY	[error condition – could also mean that user did not terminate his last executed pattern correctly with END OF PROGRAM FLAG]

3.4.1.7.2 NGC Trigger – Shutter Operation

The Shutter Module (Figure 13) is a companion module to the Sequencer Module, what means, that commands have to be written to both modules for Shutter Operation of NGC . The Sequencer is only used to execute the pattern sequences needed for CCD operation and trigger the different operational modes (WIPE, INTEGRATE, READOUT) of the acquisition. The shutter module is responsible for Exposure Time , Shutter Open and Shutter Close signals and the monitoring of shutter close/open status and fail condition. The exact exposure time is set in the Shutter Exposure Time Register (# 7004) and the shutter set-up is done with the Shutter Control/Status Register.

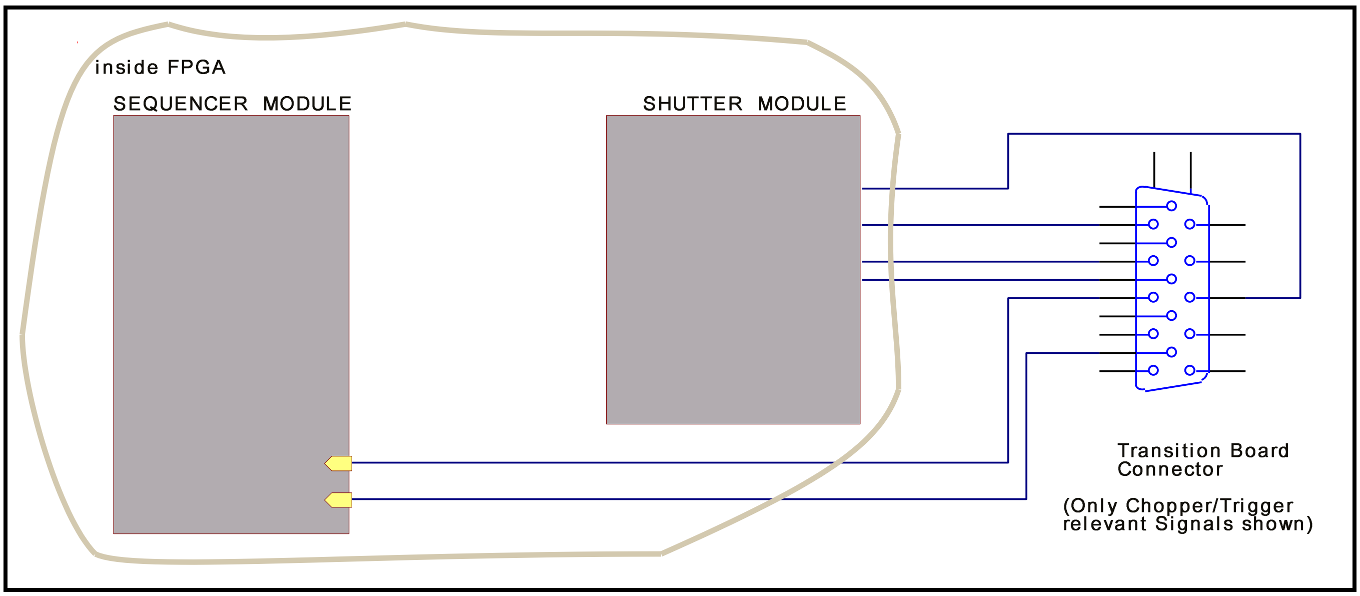


Figure 13 NGC Trigger Connections Shutter Operation

Figure 14 is the most general timing we can encounter in the interface to a shutter. This figure is fundamental to understand the firmware module. The signal `ShutterOpenCommand` is an output that commands the opening and the closing of the shutter blades. An ideal shutter remains open whenever the `ShutterOpenCommand` signal is asserted and close otherwise. The time this signal is asserted is the desired exposure time. The signal `ShutterCloseStatus` is an input to NGC and is asserted as long as the shutter is fully closed. The signal `ShutterOpenStatus` is also an input and it becomes asserted as soon as the shutter is fully open. The `ShutterFail` signal reports a failure on the shutter.

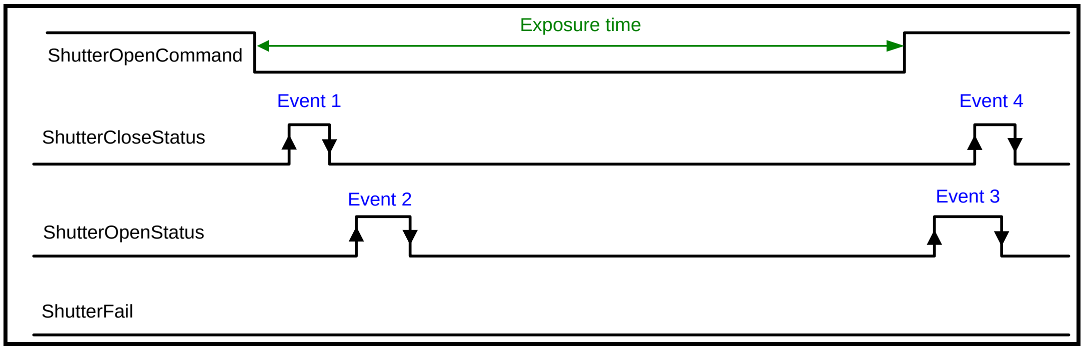


Figure 14 Shutter Signals

Due to the programmable shutter interface, different shutter configurations can be implemented. The programmability of the interface allows for different signal polarities and for the measurement of the delays of the shutter mechanics with respect to the exact time, at which a start or end of an exposure is commanded.

The exposure time is controlled by the FPGA on the NGC Basic board. The timing control is based on the board local 100.000 MHz clock and hence accurate to within +/- 10 ns. However, this fast clock is divided down to 1ms and this slow clock is then used for the exposure time counting. The resolution of the exposure time setting is therefore 1ms and the maximum exposure time is 280 minutes (4.6 hours) because the exposure time register is 24-bit.

Signal	Type	Function
/ShutterOpenCommand	Output	Shutter opens when signal is active (active low or high programmable)
/ShutterCloseStatus	Input	Pulses inactive, when shutter leaves and enters fully closed status (acc. to configuration of Fig. XXX)
/ShutterOpenStatus	Input	Pulses inactive, when shutter leaves and enters fully open status (acc. to configuration of Fig. XXX)
ShutterFail	Input	Shutter failed

Table 30 Shutter Signals

The control of the shutter is done via a set of 6 registers detailed in Table 31.

Address	Register	Access type	length
0x7000	Shutter Control/Status Register	r/w	16-bit
0x7004	Shutter Exposure Time Register	r/w	24-bit
0x7008	Event1 counter	ro	16-bit
0x700C	Event2 counter	ro	16-bit
0x7010	Event3 counter	ro	16-bit
0x7014	Event4 counter	ro	16-bit

Table 31: Shutter interface registers.

bit	write access	read access
15	StartExposure	not used
14	AbortExposure	not used
13	PauseExposure	not used
12	Continue Exposure	EndofExposure
11	DarkExposure	
10	OpenCommand Activ Hi	
9	not used	Reserved
8	not used	ShutterFail Line
7	Event4 is a rising edge	Event4 counter overrun
6	Event3 is a rising edge	Event3 counter overrun
5	Event2 is a rising edge	Event2 counter overrun
4	Event1 is a rising edge	Event1 counter overrun
3	not used	Event4 encountered
2	not used	Event3 encountered
1	Shutter mode bit 1	Event2 encountered
0	Shutter mode bit 0	Event1 encountered

Table 32: Shutter Control/Status register

A state diagram for shutter operation is shown in Figure 15 and described below.
The operation of the sequencer and associated shutter is controlled by the same sequencer state machine as for IR acquisition. The TRIGGER ENABLE [Command register, Bit 6] must be set.
After RESET the sequencer is in STATE 1, the IDLE state. Then the complete sequence and the patterns consisting of Wipe(s), Integrate and Read-Out and the voltages for Wipe are loaded. After a RUN SEQUENCER command [Command register, Bit 0] STATE 2 is entered and leads to the execution of the Wipe sequence with the corresponding clock patterns and voltages. The last pattern of Wipe contains the WAIT FOR TRIGGER Bit [PATTERN RAM HIGH, Bit 28] . The state machine enters STATE 3, a wait state. The software recognizes this state by reading the Sequencer Waiting for Trigger Bit in the Status Register (#6000,Bit 2). Now the software sets the voltages for Integrate. The trigger signal to exit STATE 3 will then be a software trigger [Command Register, Bit 7]. The software writes now a Start Exposure to the Shutter module (Shutter Control/Status Register), what opens the Shutter and starts exposure time measurement. In STATE 4 the Integrate patterns are executed. The last pattern of Integrate contains the WAIT FOR TRIGGER Bit [PATTERN RAM HIGH, Bit 28] and the machine enters STATE 5 a Wait state (The Integrate Pattern is hold, because the Wait State always holds the last pattern where the WAIT FOR

TRIGGER Bit was set). The software recognizes this state by reading the Sequencer Waiting for Trigger Bit in the Status Register (#6000,Bit 2). The software also checks the Exposure Time in the Shutter Exposure Time Register (set with Shutter module set-up) and determines finished exposure. The Shutter module itself is responsible for closing the shutter. Now the voltages for Read-Out are loaded and Read-Out starts in STATE 6 after a software trigger is issued. The last pattern of Read-Out contains the END OF PROGRAM FLAG [PATTERN RAM HIGH Bit 30] and the state machine enters STATE 1, the IDLE state .The END OF PROGRAM Bit (Sequencer Status Bit 4) is set.

An exposure can be paused by writing the to the Shutter module Shutter Control/Status Register the PAUSE EXPOSURE signal. A CONTINUE EXPOSURE command to the Shutter Control/Status Register clears the PAUSE EXPOSURE signal and the acquisition continues.

Remark : The Trigger Phase signals are in shutter mode still enabled, this opens the possibility to synchronize shutter operation with the time reference signal.

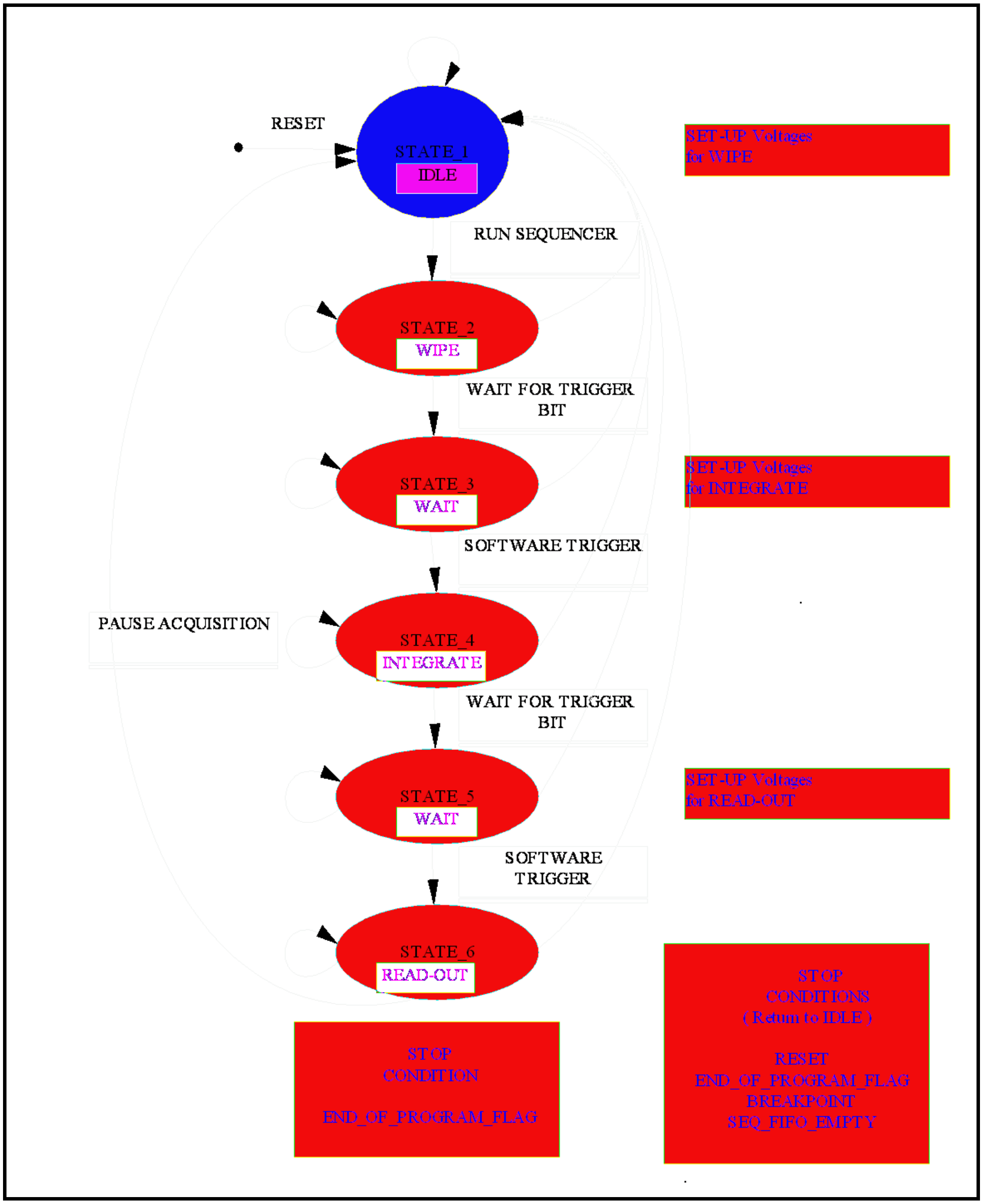


Figure 15 Sequencer State Diagram - Shutter

3.4.1.8 AQ Manager

Each board contains one AQ Manager. He is responsible for organizing the video data transfer of and through this board. The AQ Manager contains in addition a programmable video data simulator.
After conversion and packet formation as defined in the AQ Manager Command Register, the data are written automatically to the AQ FIFO's and transferred. Then AQ manager looks in the Transfer Counter if additional packets should arrive from the downstream link. If yes and arrived, these data packets are also written to the AQ FIFO's and transferred.
In simulation mode the video data are generated inside the FPGA.
In simulation mode 1 the video data are derived by a counter incremented by the conversion strobe.
In simulation mode 2 the video data reflect the video channel number.

Table 33 AQ Manager Command Register– Write (Addr #3000)

BIT 5..0	**NADC = Number of ADCs to read on this board**
BIT 5	**reserved**
BIT 6	**Clamp/Filter Selection (0 → Filter) (Clamp is Sequencer Bit 35) see**
BIT 7	**Filter Time Constant 0 → .5 us 1 → 5 us see**
BIT 15..8	**PACKETSIZE =Number of data words (32Bit) per Data Packet (must be multiple of NADC)**
BIT 19..16	**Transfer Counter (Number of packets arriving from Downlink → 1 means 1 packet from Downlink)**
BIT 20	**Enable Conversion by Convert1 (Convert1 is Sequencer Bit 33)**
BIT 21	**Enable Conversion by Convert2 (Convert2 is Sequencer Bit 34)**
BIT 23..22	**reserved**
BIT 24	**Declare Module as First in Chain (module needs no transfer permission)**
BIT 27..25	**reserved**
BIT 28	**Simulation Mode/Acquisition Mode**
BIT 29	**Simulation Mode : 0 =Numbers 1 =Counter (not considered in Acquisition Mode**

BIT 31..30	**reserved**

Table 34 AQ Manager Status Register– Read (Addr #3000)

BIT 0	**AQ State " IDLE "**
BIT 1	**AQ State " Waiting for CONVERT "**
BIT 2	**AQ State " Waiting for Downlink Data "**
BIT 3	**Error Condition : Data Overrun AQ FIFO**
BIT 4	**Conversion Error Bit**
BIT 5..6	**Reserved**
BIT 7	**Bandwidth Switch 0→ High Bandwidth**
BIT 11..8	**Number of ADCs (Replica of Set-up)**
BIT 19..12	**Packet Size (Replica of Set-up)**
BIT 23..20	**Upstream_Count (Replica of Set-up)**
BIT 27..24	**Transfer_Counter (Initial) (Replica of Set-up)**
BIT 31..28	**Transfer_Counter (Actual State)**

An ADC Error register and ADC Error Bit (see Table 34) have been implemented for test and maintenance. The Error Register shows, if all ADC's are converting. A not converting ADC might hang up the data transmission (the communication would still work), a temporary missing ADC scramble the image. A read of the AQ Manager Status Register will then show the ADC Error Bit set, in the ADC Error register the not working ADC will have the bit of the corresponding ADC set. The Error signals once set, are only cleared by writing to the ADC Error register (value don't care) or after a Reset.

Table 35 ADC Error Register (Addr #3002)

BIT 0	**ADC 0 Error Bit**
BIT 1	**ADC 1 Error Bit**
BIT 2	**ADC 2 Error Bit**
BIT 3	**ADC 3 Error Bit**
BIT 31..4	**'0'**

The ADC numbering (see Figure 22) corresponds to the values in the real time display in simulation mode and read-out mode ADC.

Two conversion signals are available

Table 36 Convert Signals

Convert1	**Sequencer Clock 32 (starting with clock 0)**
Convert2	**Sequencer Clock 33 (starting with clock 0)**

The AQ Manager Delay Register delays the conversion strobe to the ADC's (in relation the sequencer generated clocks) by max 2.56 us.

Table 37 AQ Manager ADC Delay Register – Write (Addr #3001)

BIT 7..0	**Delay of Conversion Strobe = (Value * 10ns + 10ns)**

3.4.1.9 Basic Board Front Panel

The Basic Board Front Panel green LED's indicate :

- System Reset
- Convert
- Sequencer Running
- Bias and Clock output enable

The red LED's show :

- Link Lock Status of Uplink and Downlink
- Tx and Rx transfers

LEMO connectors carry :

- The buffered conversion signal (Sequencer Clock 33 or 34)
- Two digital markers (Sequencer Clock 35, 36)
- A software selectable buffered differential video channel signal
- Two independently selectable buffered detector clock signals

RESET
CONVERT
SEQ_RUN
OUTPUT_EN

RX
TX
DOWNLINK LOCKED
UPLINK LOCKED

Figure 16 Front-End Basic LED Indicators

Clock 2	Monitor 1	Video differential	Markers 2	1	Convert	LED's

Figure 17 Front-End Basic Board - Front Panel

3.4.1.10 Video Chain

The video chain consists of the video amplifier and the ADC connected to the FPGA. The gain of the video amplifier is 1. The video amplifier has a differential input, offset control (see Table 38) and selectable bandwidth (see Table 40).

Table 38 Bias Set-up Register (Extract Offset Control) – Write ADDR #8000

BIT 13 .. 0	Data Value
Bit 21 .. 16	**Video_Offset_P = 3A** **Video_Offset_N = 3B**

Different ADC types can be installed

Table 39 Possible ADC Types on Basic Board

ADC Type	Video Input Range	ADC Range
AD 7677 1MHz/16Bit	0 to 4 Volt	+/- 2.5 Volt
AD 7621 3MHz/16Bit	0 to 2.5 Volt	+/- 2.5 Volt
AD 7677 1MHz/16Bit	0 to 4 Volt	+/- 4 Volt

The digital output of the ADC's is in straight binary notation.

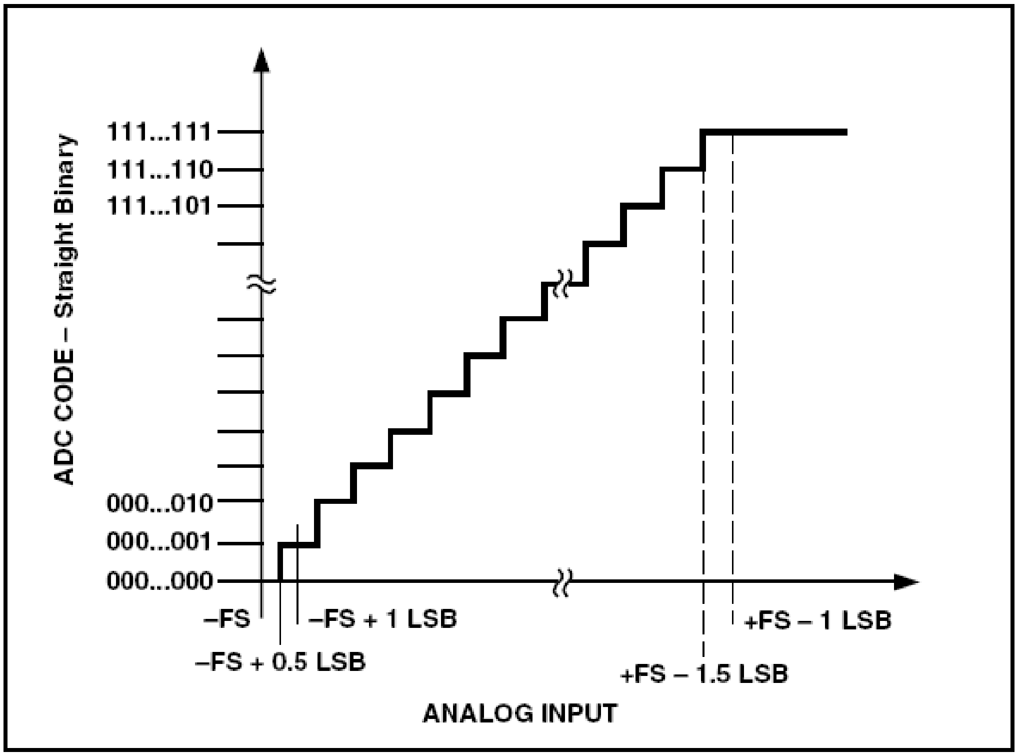


Figure 18 ADC Code against Analog Input

The video amplifier can be used in two modes selected in the AQ Manager Command Register:
The normal mode selects a switchable filter (Bit 6 =0). The time constant of the filter is chosen with Bit7.
Clamp/Sample mode requires a hardware change. The Filter resistor is exchanged to a sampling capacitor and the filter capacitor is taken out and a low value Resistor (100 Ohm) inserted instead.
Bit 6 of the AQ Manager Command Register is set to 1. Then sequencer clock 35 is the clamp/sample control. Bit 7 has no function in clamp/sample mode.

Table 40 AQ Manager Command Register (Clamp/Filter Selection Bits) – Write (Addr #3000)

BIT 6	**Clamp/Filter Selection (0 → Filter)** **(Clamp is Sequencer Bit 35) see**
BIT 7	**Filter Time Constant** **0 → .5 us** **1 → 5 us see**

3.4.1.11 Clock and Bias Rails and Settings

To accommodate different detectors, the rails for clocks and biases are jumper selectable (Figure 19). The drawing also reflects the lay-out on the Basic Board for the clock and bias modules.

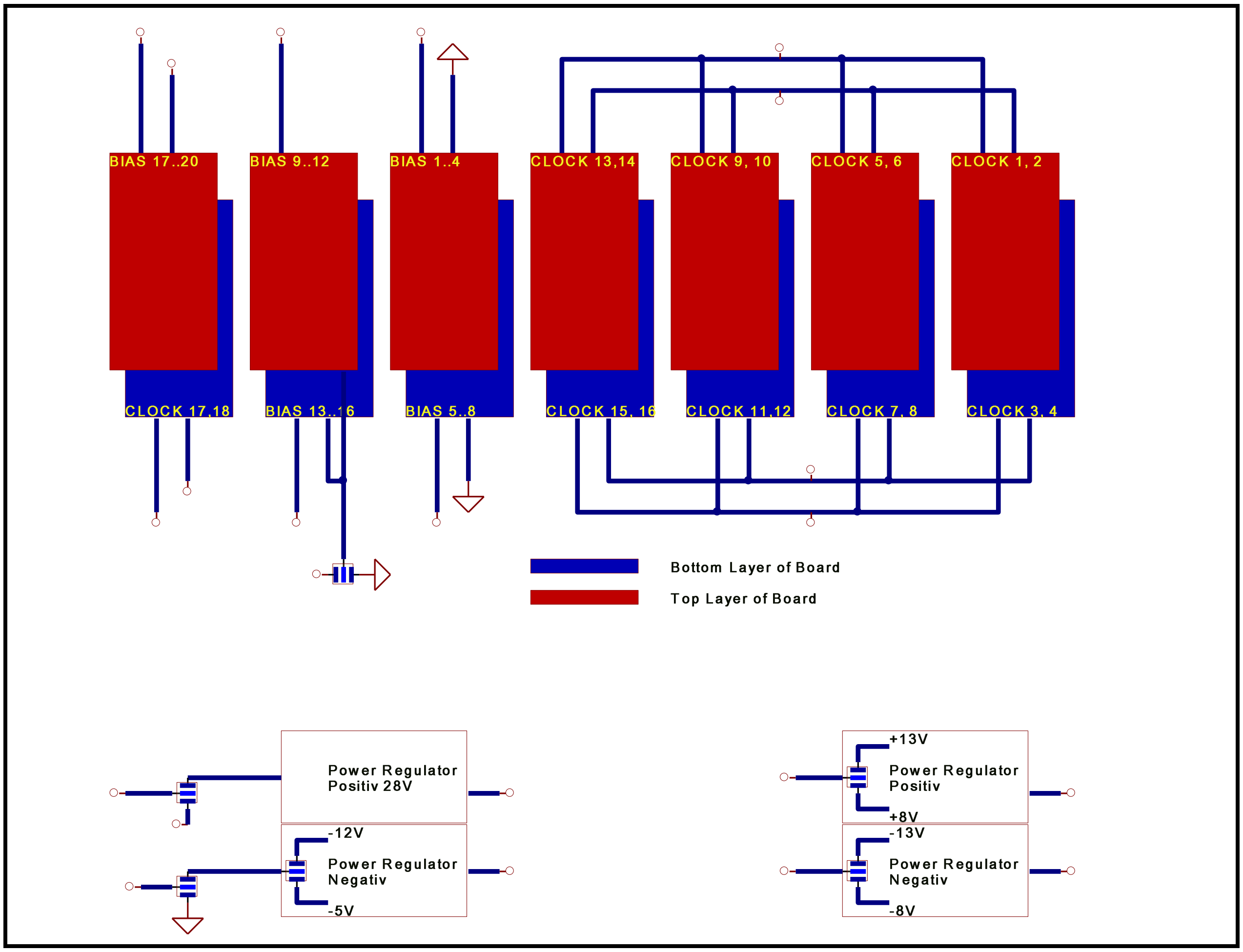


Figure 19 Clocks and Biases – Rails and Jumper Settings

3.4.1.12 Basic Board Jumpers

There are different jumper fields for the configurations of the Basic board:

- ADC Selection configures the chosen ADC and the corresponding operation mode
- Digital Base Power connects the main digital 3.3 Volt supply to the digital circuits and to the FPGA regulators (see schematic)
- ID Setting tells the ID register about the board type (see 3.4.1.2)
- FPGA Power connects the 3.3 Volt derived FPGA supplies to the FPGA
- Bias and Clock Settings selects the Bias and Clock rails (see Figure 19)

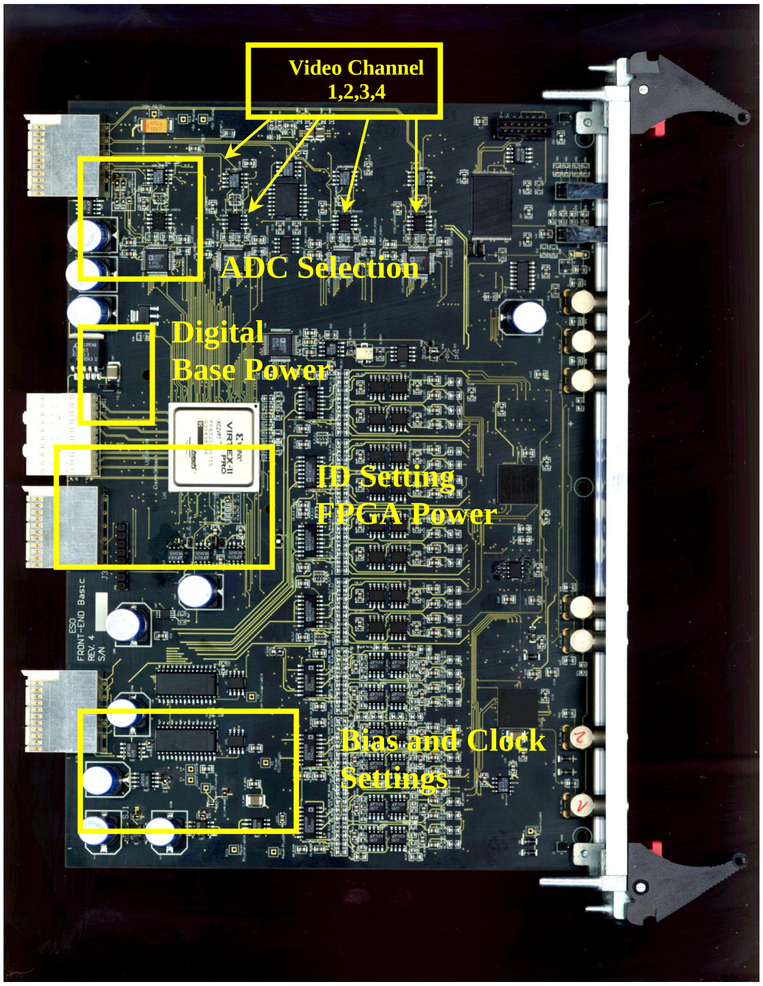


Figure 20 Basic Board Jumper Fields

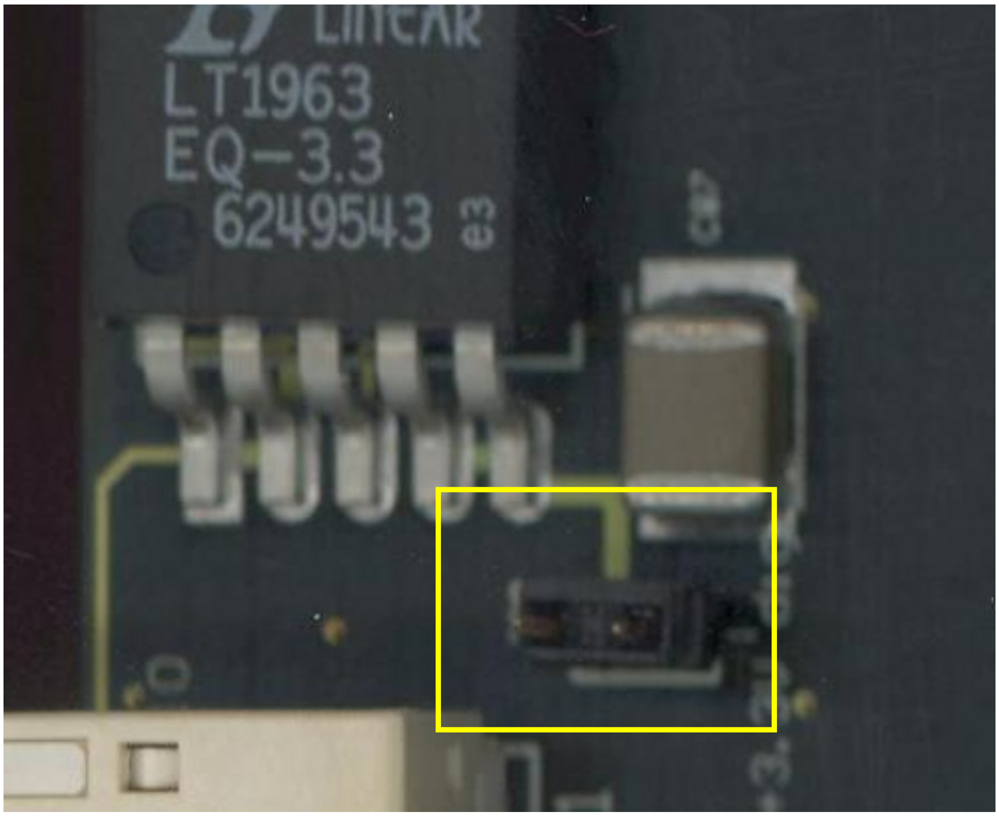


Figure 21 Digital Base Power Field

The Digital Base Power Field jumper connects the main 3.3Volt digital supply to the board.

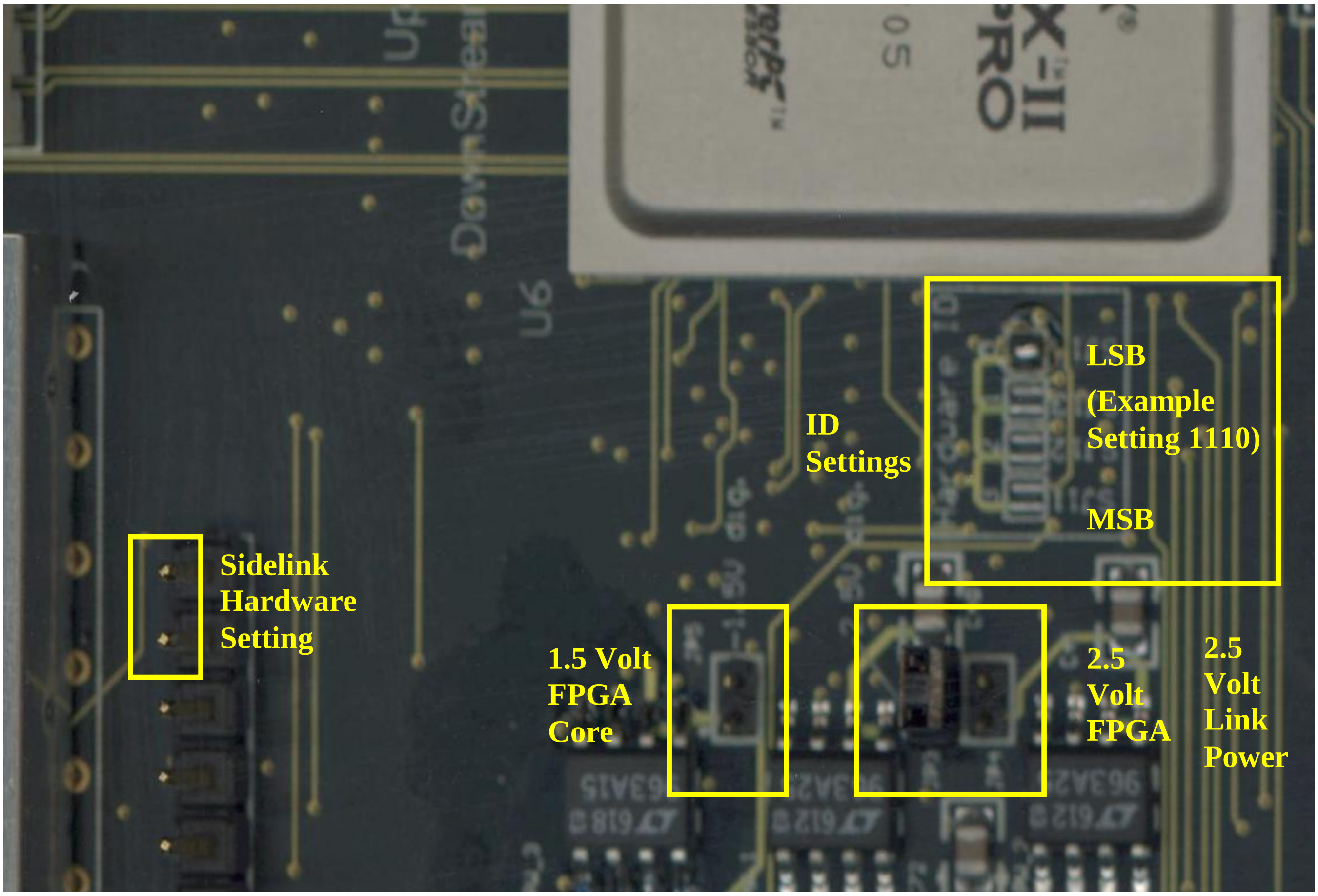


Figure 22 ID Setting, FPGA Power Field and Sidelink Selection

The FPGA Power Field jumpers separate the FPGA supplies from the FPGA.
The ID Field jumpers provide a readable board code to the ID register.
The Sidelink jumper (see chapter 3.4.1.1.) , when set, switches the uplink to the Sidelink transceiver (and then to the Transition board fiber optic transceiver) without software intervention.

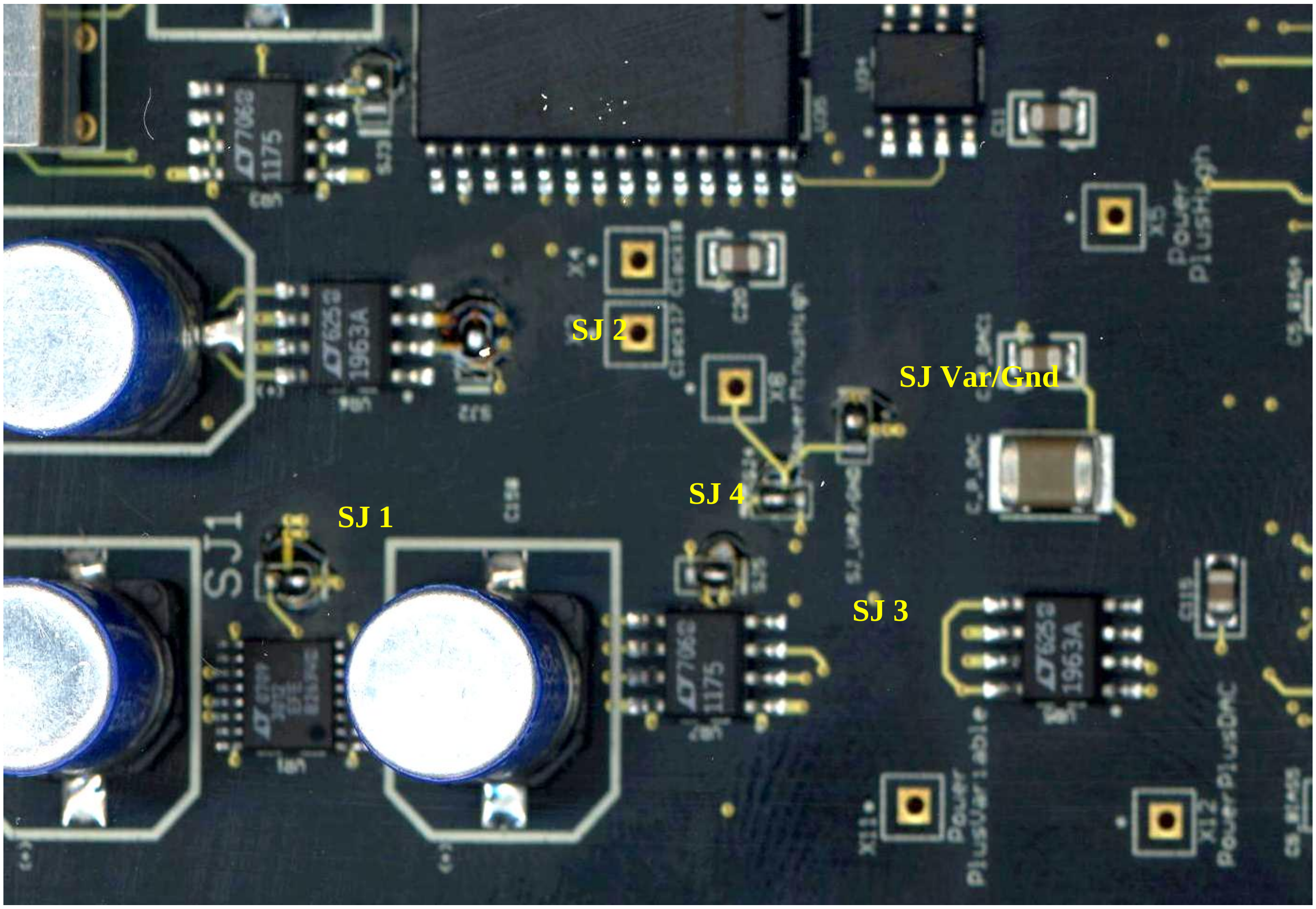


Figure 23 Bias and Clock Setting Field

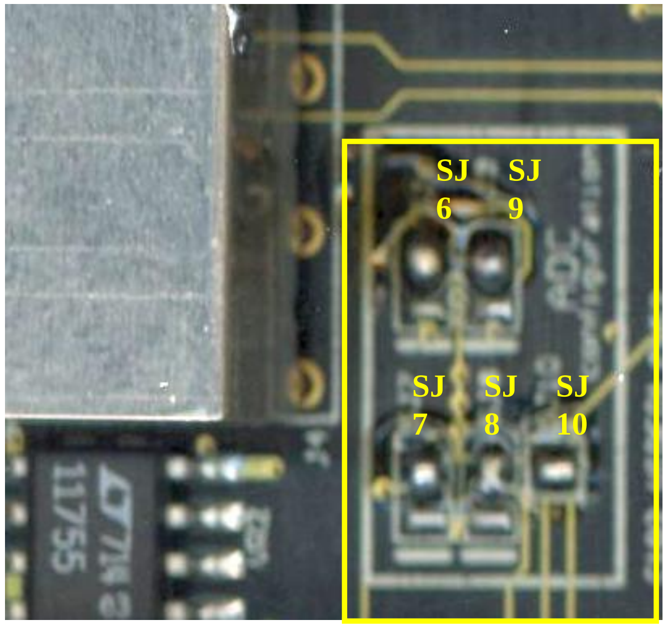


Figure 24 ADC Selection Field (Configuration AD 7677)

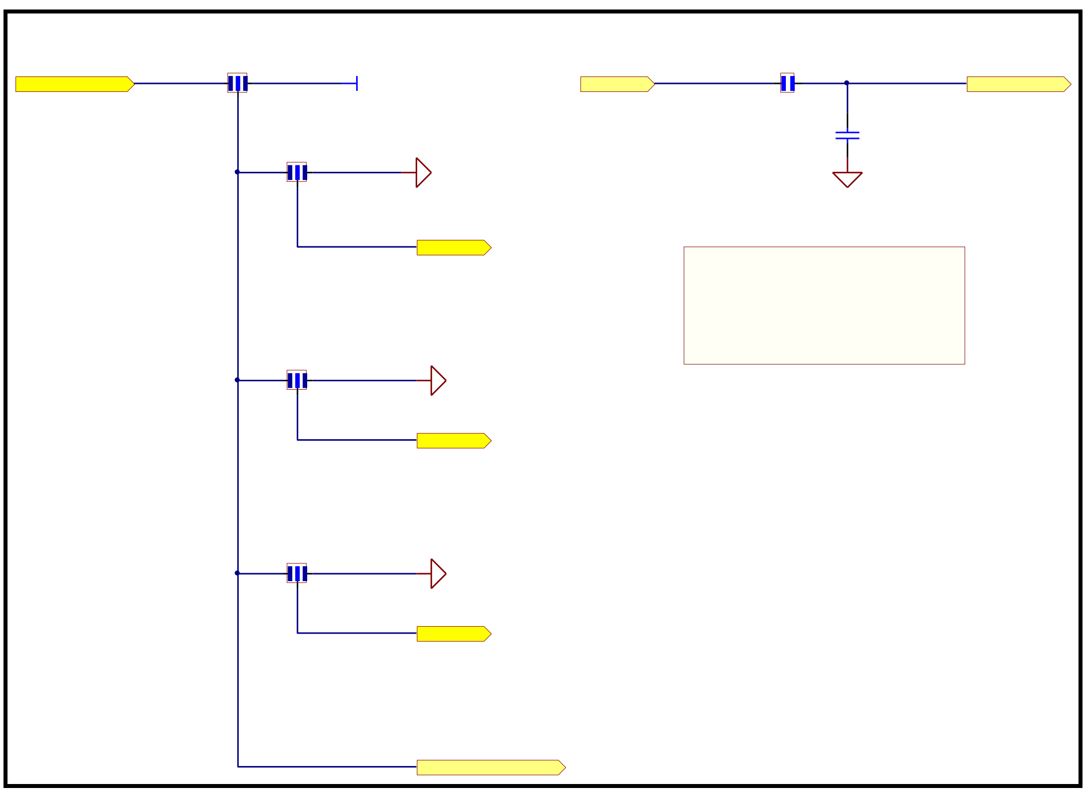


Figure 25 Jumpers for ADC Selection

3.4.2 Transition Board of Front-End Basic-Board

The Transition Board of the Front-End Basic-Board establishes the interface connections to the external world on different connectors :

- Clocks and Biases outputs
- Video signal inputs
- .Shutter Control I/O
- Fiber Optic transceivers of high speed links

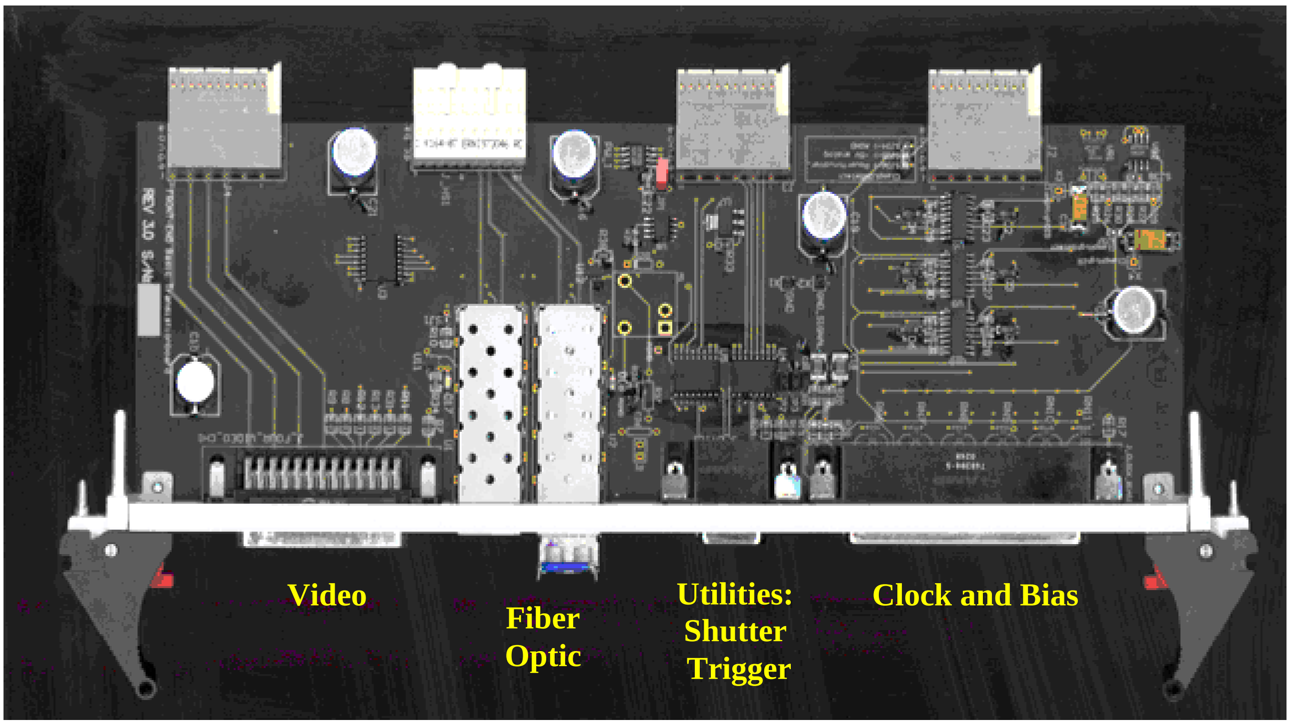


Figure 26 Transition Board of the Front-End Basic-Board

The video connector pin-out is shown in Figure 27

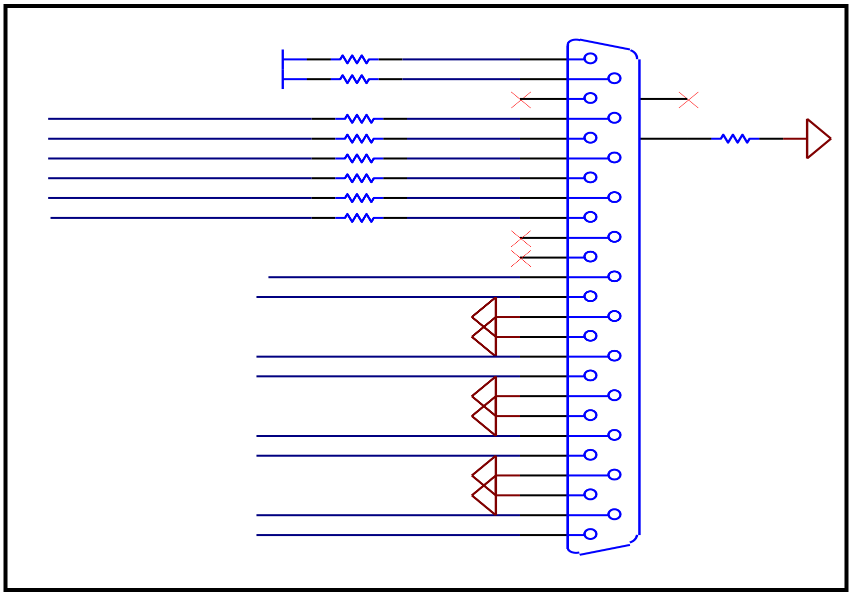


Figure 27 Video Connector

POWER_MINUS_PREAMP POWER_PLUS_PREAMP	Power Lines (+/- 6V) for Video Amplifier
PREAMP_BIAS_OUT	Analog Output (+/- 5V) – can be used for Video Amplifier Offset
DIODE_BIAS_OUT	Analog Output (+/- 5V) – can be used for Detector Bias
I2C_CLK I2C_SDA	I2C Bus
GPIO_EXT_8 ..9	General Purpose galvanically decoupled digital lines
PREAMP_BIAS_OUT	Video Amplifier Bias Voltage [+/- 5V]
DIODE_BIAS_OUT	Detector Bias Voltage [+/- 5V]
VIDEO_NEG x VIDEO_POS x	Differential Video Inputs
	Analog GND

Table 41 Video Connector

The utilities connector (Figure 28) carries signal for shutter control on optical systems and trigger inputs for IR applications.

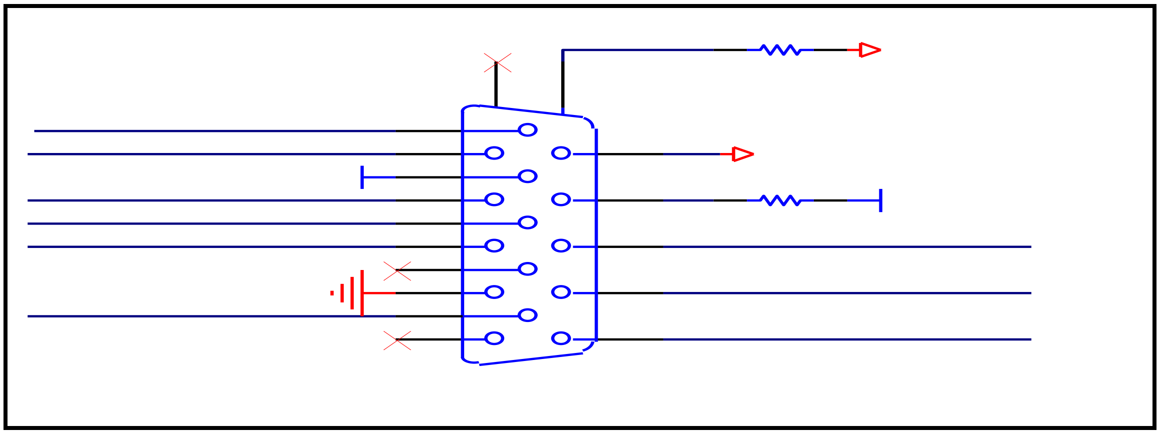


Figure 28 Utilities Connector

I2C_SDA_EXT I2C_SCL_EXT	I2C BUS
Shutter Signals	Shutter Control Signals
TIM_PHASE_A TIM_PHASE_B	Trigger Inputs
V_IN_DIG	Digital Power Supply (5V) (only used for test purpose – R24 is removed in production boards)
	Digital Ground
VCC_EXTERNAL	External Power Supply Input for Isolators
GND_EXTERNAL	External Power Supply GND

Table 42 Utilities Connector

The Clock and Bias Connector (Figure 29) provides clocks and biases to the detector and power supply pins for special applications like clock buffers for mosaics.

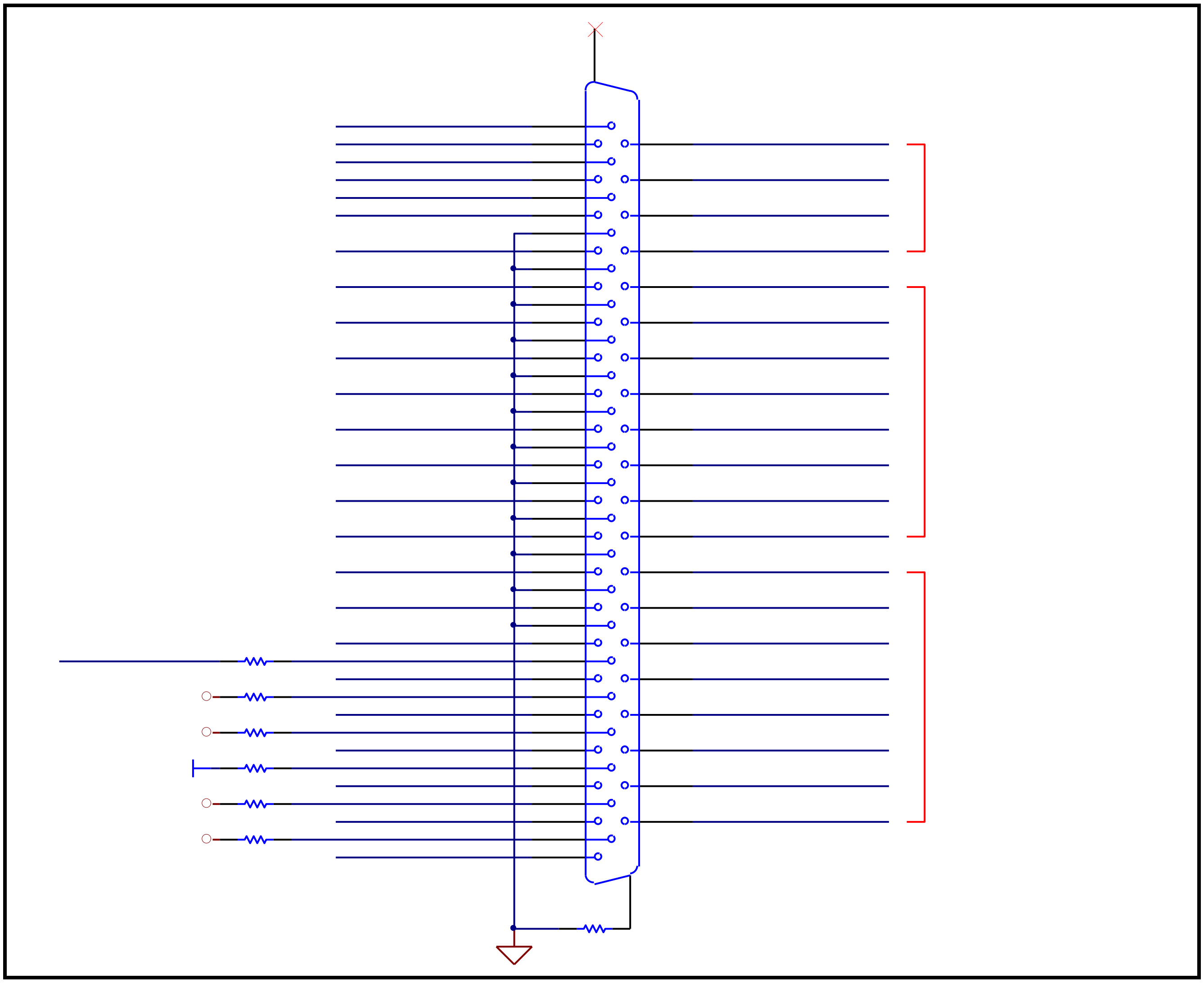


Figure 29 Clock and Bias Connector

BIAS 1..8	Unipolar Biases (0 to PowerPlusHigh)
BIAS 9..16	Configurable Biases (PowerMinusHigh to PowerPlusHigh)
BIAS 17..20	Bipolar Biases (Range of Clock Rails : PowerMinusVariable to PowerPlusVariable)
CLOCK 1.. 18	Detector Clocks (Range of Clock Rails : PowerMinusVariable to PowerPlusVariable)
CLOCK 19.. 24	Switch generated Detector Clocks (see chapter 3.4.2.1)
OUTPUT_ENABLE_EXT	Output Enable Signal for Clock and Bias
PowerMinusVariable_BP PowerPlusVariable_BP	Clock Power Supply
PowerMinusHigh_BP PowerMinusHigh_BP	Bias Power Supply
	Analog GND

Table 43 Clock and Bias Connector

3.4.2.1 Flexible Read-out of Multi-output CCDs with Switches on the Transition Board

To read out multi-output CCDs like the CCD for MUSE (e2v CCD 231) in a flexible manner (1,2,4 port readout) with a minimum number clocks the following system is implemented on the Transition Board.

See below the Chip Schematic of e2v CCD 231

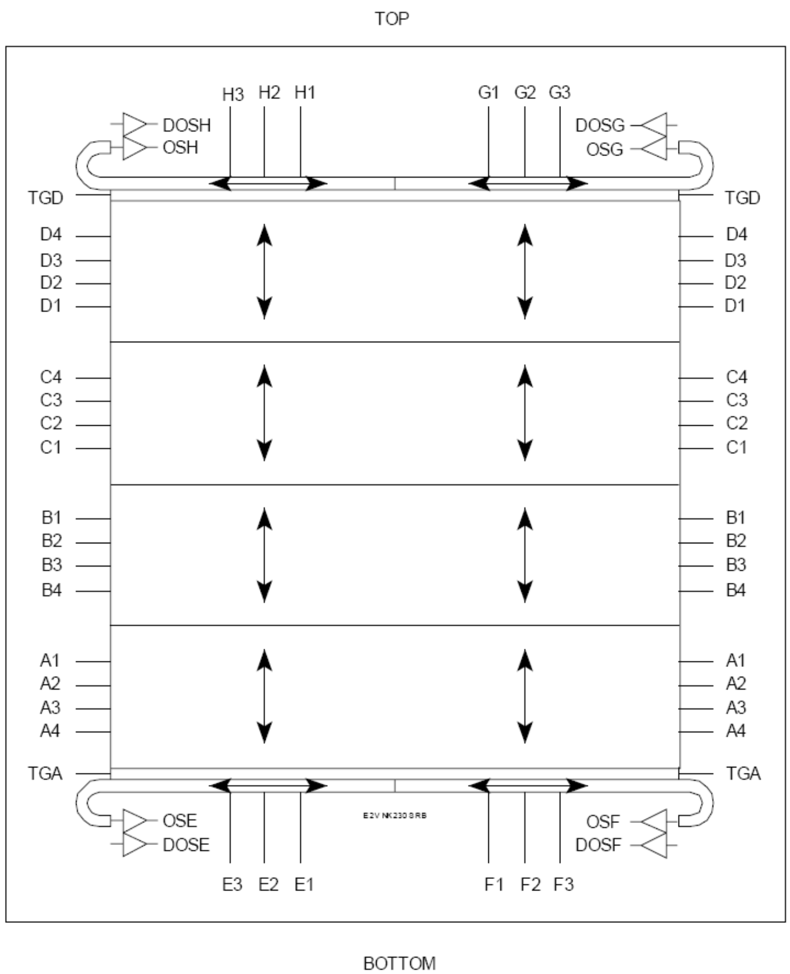


Figure 30 Chip Schematic of e2v CCD 231

Line clocks organization for Read-out through 4 outputs

	IØ1	IØ2	IØ3	IØ4	
A section transfer towards E-F register	A1	A2	A3	A4	TGA = IØ4
B section transfer towards E-F register	B1	B2	B3	B4	
C section transfer towards G-H register	C1	C2	C3	C4	
D section transfer towards G-H register	D1	D2	D3	D4	TGD = IØ1

Table 44

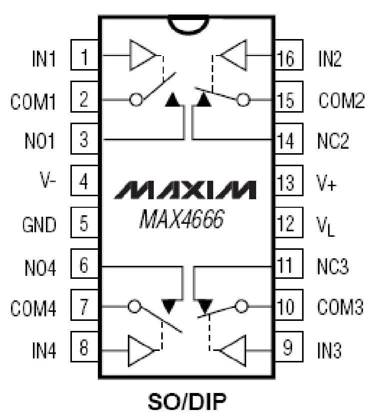
Line clocks organization for Read-out through 1 or 2 outputs

	IØ1	IØ2	IØ3	IØ4	
A section transfer towards E-F register	A1	A2	A3	A4	TGA = IØ4
B section transfer towards E-F register	B1	B2	B3	B4	
C section transfer towards E-F register	C4	C3	C2	C1	
D section transfer towards E-F register	D4	D3	D2	D1	TGD = “low”

Table 45

The difference between the readouts is the ordering of the C and D section with respect to A and B.

This reordering can be accomplished by analog switches
(MAX4666, SO16, Ron < 5 Ohm)



MAX4666		
LOGIC	SWITCHES 1, 4	SWITCHES 2, 3
0	OFF	ON
1	ON	OFF

Figure 31 MAX 4666 Analogue Switch

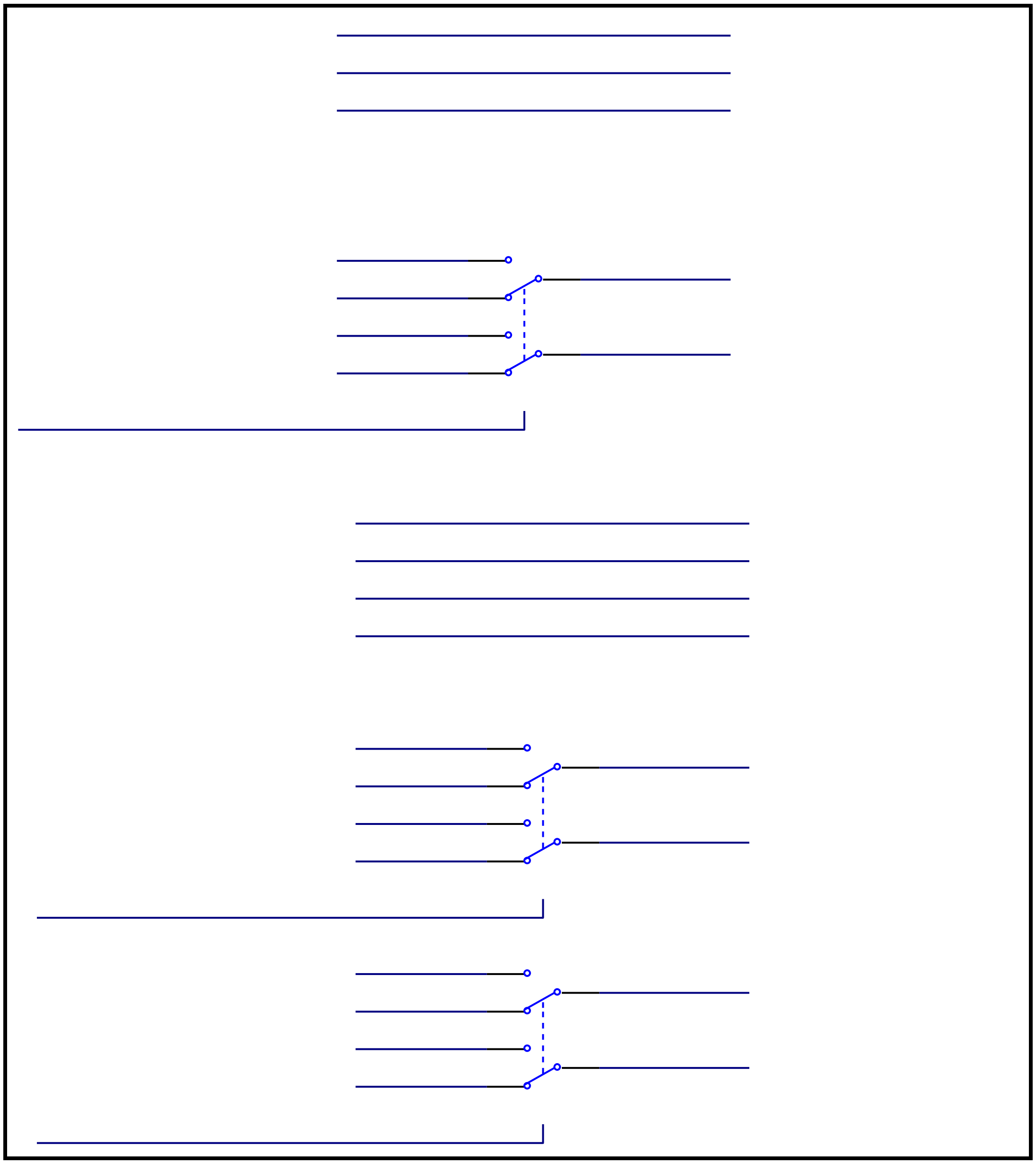


Figure 32 Switch Set-Up

Table 46 Control Clock Settings

Vertical Control Bit	Horizontal Control Bit	CCD Readout Mode
0	0	1 Channel Readout
0	1	2 Channel – Right and Left Amplifier Readout
1	0	2 Channel – Top and Down Amplifier Readout
1	1	4 Channel Readout

In this way a CCD can be read on 1 or 2 or 4 ports with only one clock pattern. This is a lot of simplification and creates symmetrical patterns automatically. Using a scheme with individual clocks would require 4 different patterns – with the possibility of errors in symmetry and elsewhere.

With two clock sub-pattern changes – can be set by two variables in the software (re-grouping : e.g. 4 to 1, 1 to 4, 2 to 3, 3 to 2 on image clocks) even the selection of the output amplifier(s) (up, down, right, left) can be accomplished. Using a scheme with individual clocks would require 9 different patterns – with the possibility of errors as stated

Archiving of only **one** clock pattern for a given CCD is needed.

The control bits are always needed to tell the software, if readout of 1,2 or 4 channels is done.

Finally the multiplexer scheme helps in archiving and creation of the CCD read out clock designs. The logical signals of the clock multiplexers can be derived either from a register or out of the sequencer clocks (must be hold throughout all patterns). The system file will contain the setting of the control bits, this information is anyway needed when the number of outputs is changed. Therefore preference is given to the register approach.

3.4.3 AQ32 Board

The AQ Module (Figure 33) is based on the XILINX Virtex Pro FPGA XC2VP20 FF1024. The addressing and function of this board is exactly as for the Basic Board (Figure 34). Main features of this module are

- **Communication**
- **Video data transfer**
- **Sequencer (used only for stand-alone tests of the board)**
- **32 data acquisition channels (can be installed with either 16 or 18 Bit ADC's)**
- **Video monitoring**

- ADC outputs of the acquisition channels connect with little glue logic on a bus structure to the FPGA.
- ADC's are the AD7677 types from Analog Devices. Optionally the board can be equipped with high speed 16Bit/3MHz ADCs type AD7621.
- The preamplifier is fully differential, input range is +/- 2.5V. There is no clamp/sample implemented in the analog chain.

The Video Channels on the board are organized in four groups of eight channels in each group. The firmware reads always four channels at the same time, video channel 1 of all groups followed by video channel 2 of all groups then video channel 3 of all groups …
The Video Connector layout is described in 3.4.4 Transition Board of the AQ32 Board.

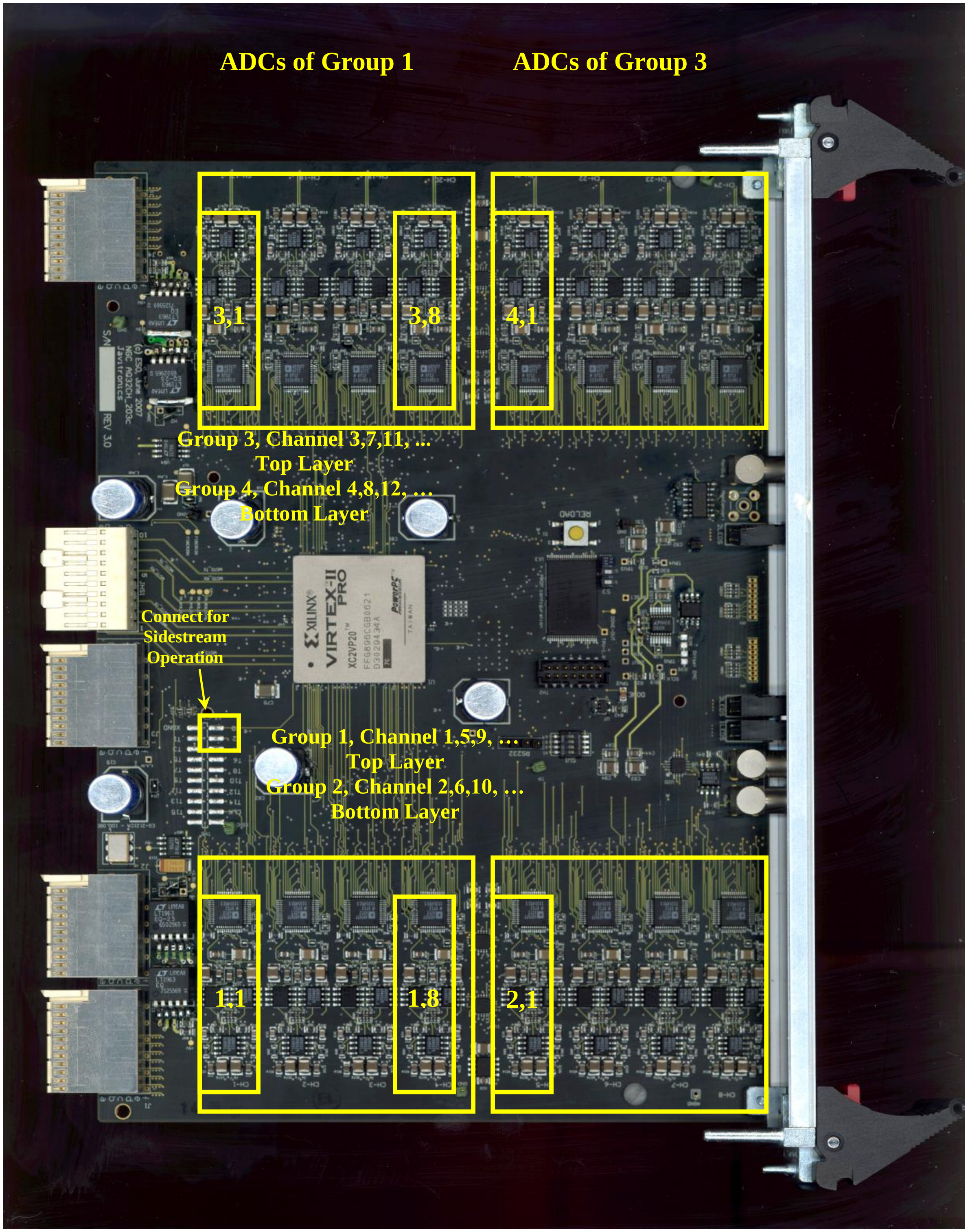


Figure 33 AQ32 Module

The addressing of functions and the structure of the AQ32 board is identical to the Basic Board – the same addresses are used (see chapter 3.4.1).

Before any addressing of functions on Front-End modules the link structure of the Front-End system must be defined. This happens by writing the CONFIG register (see also 3.4.1.1) of the module.

The status register holds system information for control and debugging. The ID register contains a unique identification of the board (see also 3.4.1.2).

The Monitor module (see also 3.4.1.3) sets the video monitors to the channels chosen and routes the signals to the front panel Lemo connectors.

ADC data from the 32 onboard video channels or the downstream links are written to the AQ fifos. The AQ Manager (see 3.4.1.8) organizes the video data transfer to the downstream link.

There is no module for Telemetry on this board.

The Bias Register has only four channels and provides the offset for the video amplifiers.

3.4.3.1 Configuration Register

Description identical to 3.4.1.1 (Configuration Register Basic Board)

3.4.3.2 Status Register

Description identical to 3.4.1.2

3.4.3.3 Monitor

The Monitor module sets the clock and the video monitors to the channels chosen in the corresponding registers and routes the signals to the front panel Lemo connectors.

Table 47 Video Monitor Register – Write ADDR #B000

BIT 2 .. 0	**Video Channel Number** **(0 corresponds to Channel 1)**
BIT 4 .. 3	**Video Channel Group Number** **(0 corresponds to Group 1)**

3.4.3.4 Conversion Error Register

At address 0x3002 there is a 32-bit register called Conversion Error Register to determine in case of a conversion error (what leads to no transmitted data), which ADC(s) did not respond to the convert signal. Each individual ADC has one bit on this register, a "1" means Conversion Error. The register is cleared after writing to the AQ32 Error register or a board reset.
The Conversion Error Bit in the AQ32 Status register (see Table 34) is cleared in the same action.

Table 48 Conversion Error Register - Read

BIT 31 .. 0 (1)	**Video Channel Number**

(1) Bit Number = (Group Number (starting with 1) * Channel Number (starting with 1)) - 1

3.4.3.5 Sequencer

The Sequencer (see function in Chapter 3.4.1.6) on the board is only used for basic tests in board stand-alone operation. Only a conversion strobe signal is generated. No clocks are fed out to the backplane.

3.4.3.6 Video Offset Register

Table 49 Video Offset Register – Write ADDR #8000

BIT 13 .. 0	Data Value
Bit 21 .. 16	DAC Channel Number

3.4.3.7 AQ Manager

Description identical to 3.4.1.8

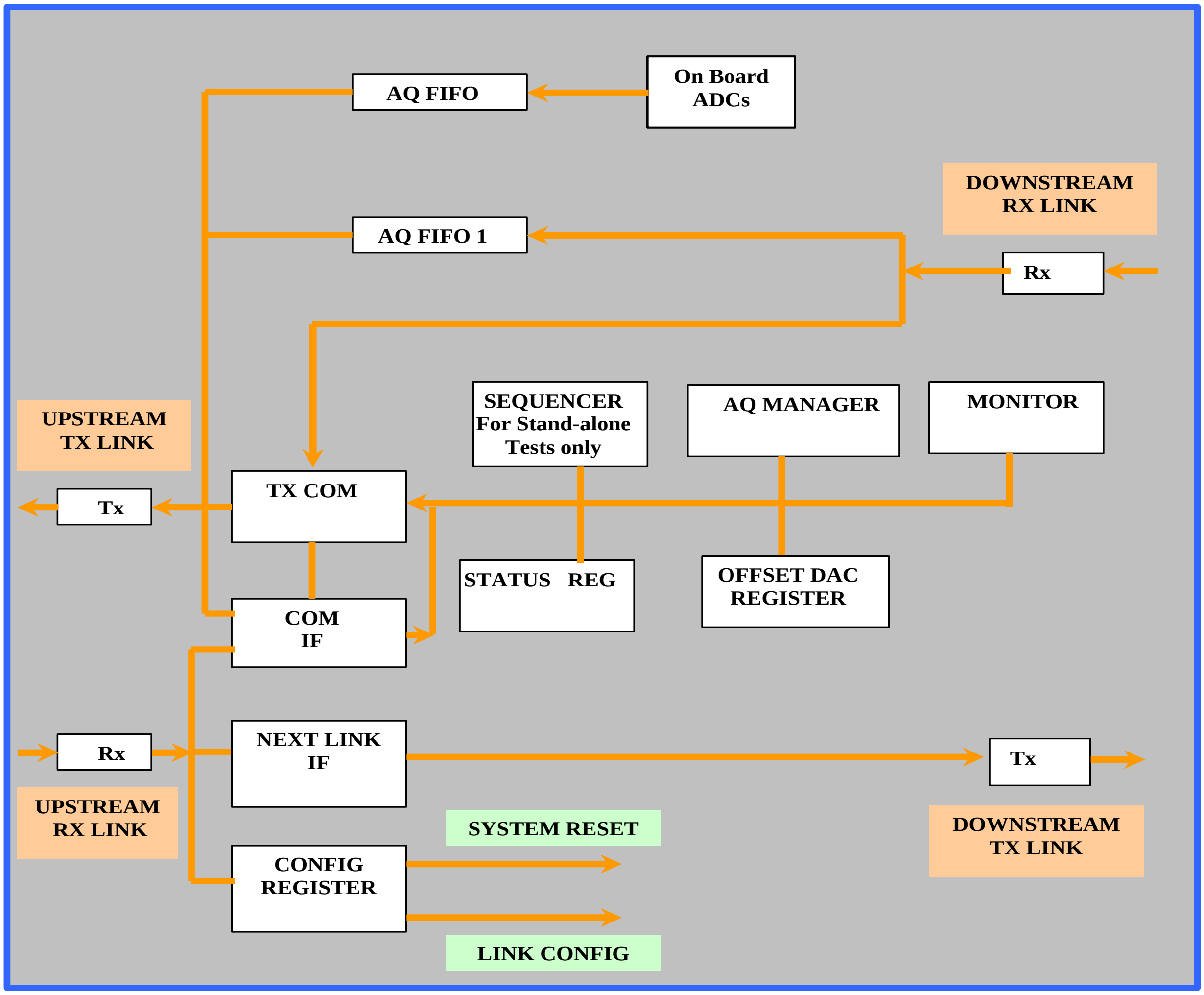


Figure 34 AQ32 Module Structure

3.4.4 Transition Board of the AQ32 Board

The Transition Board (Figure 35) of the AQ32 Board establishes the external interface connections.

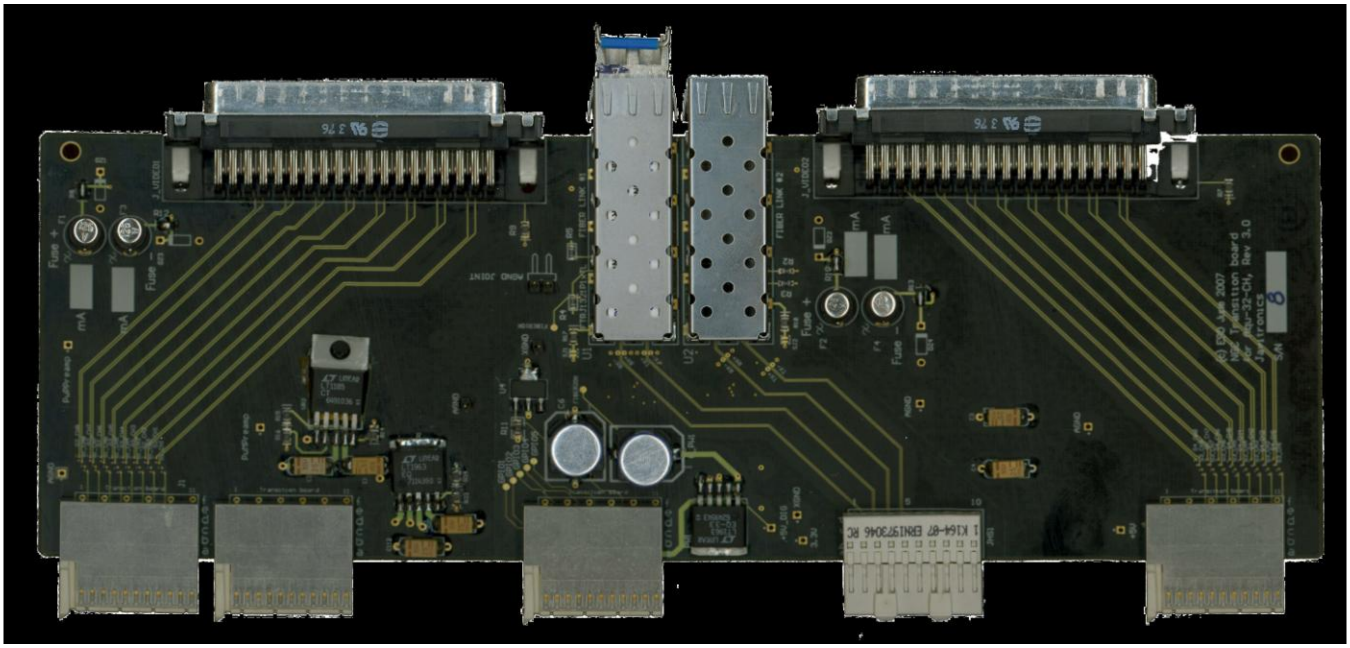


Figure 35 Transition Board of the AQ32 Board

The video signals are fed in from two connectors (Figure 37). The channels of the corresponding AQ32 Board are organized in four groups. Read-out and transmission order is channel 1 of group 1, followed by channel 1 of group 2,followed by channel 1 of group 3 …ending with channel 8 of group 4 (see

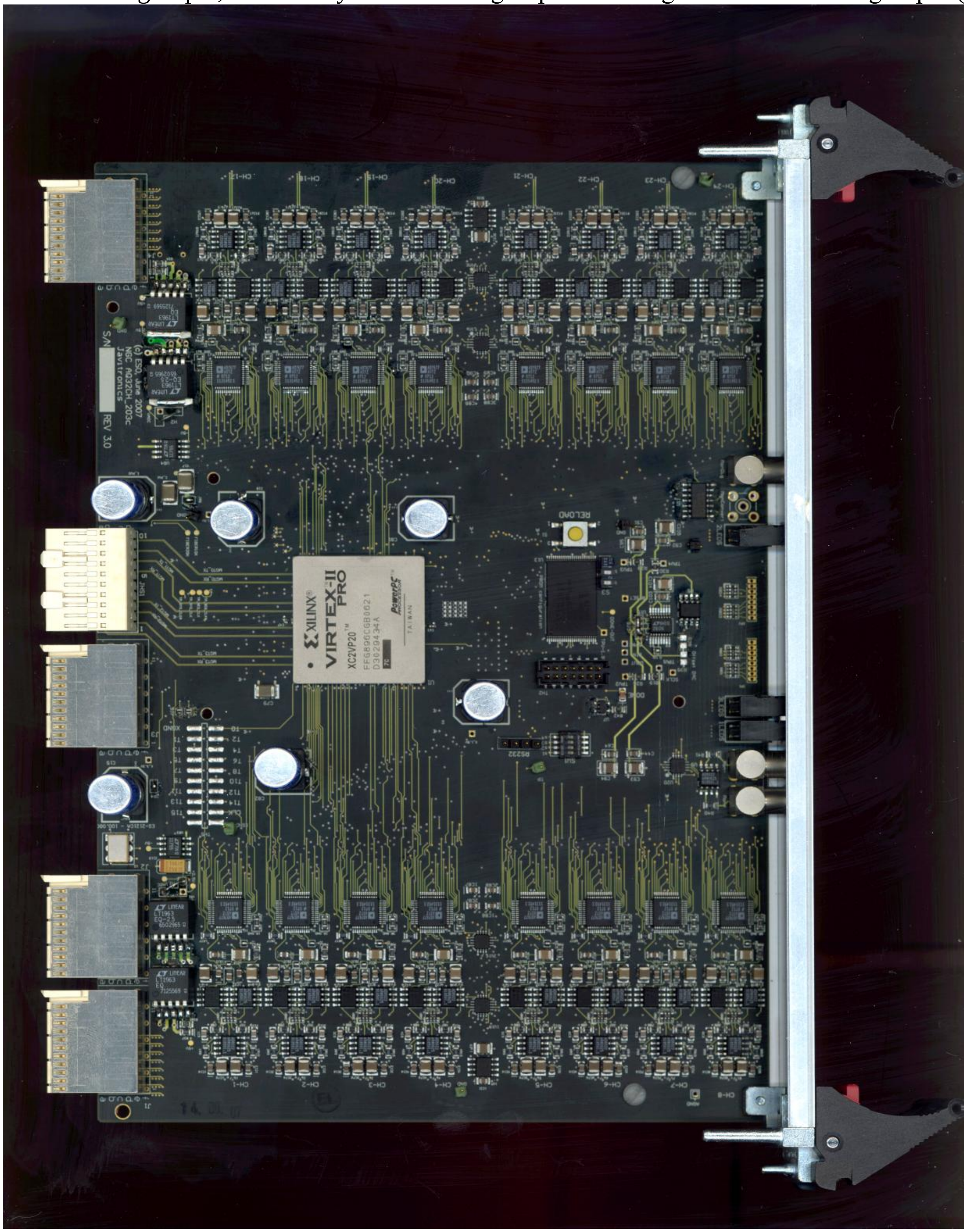


Figure 33). To simplify usage, the video connector pin description runs from Video Channel 1 to 32.
The relation between readout description and connector layout is shown in Figure 36.
The power lines for the detector preamplifier are fed out on each video connector.

Grp-ch	Video channel	Grp-ch	Video channel
G1-1	1	G3-1	3
G1-2	5	G3-2	7
G1-3	9	G3-3	11
G1-4	13	G3-4	15
G1-5	17	G3-5	19
G1-6	21	G3-6	23
G1-7	25	G3-7	27
G1-8	29	G3-8	31
G2-1	2	G4-1	4
G2-2	6	G4-2	8
G2-3	10	G4-3	12
G2-4	14	G4-4	16
G2-5	18	G4-5	20
G2-6	22	G4-6	24
G2-7	26	G4-7	28
G2-8	30	G4-8	32

Figure 36 Relation of Video Groups and Video Channel Connector Layout

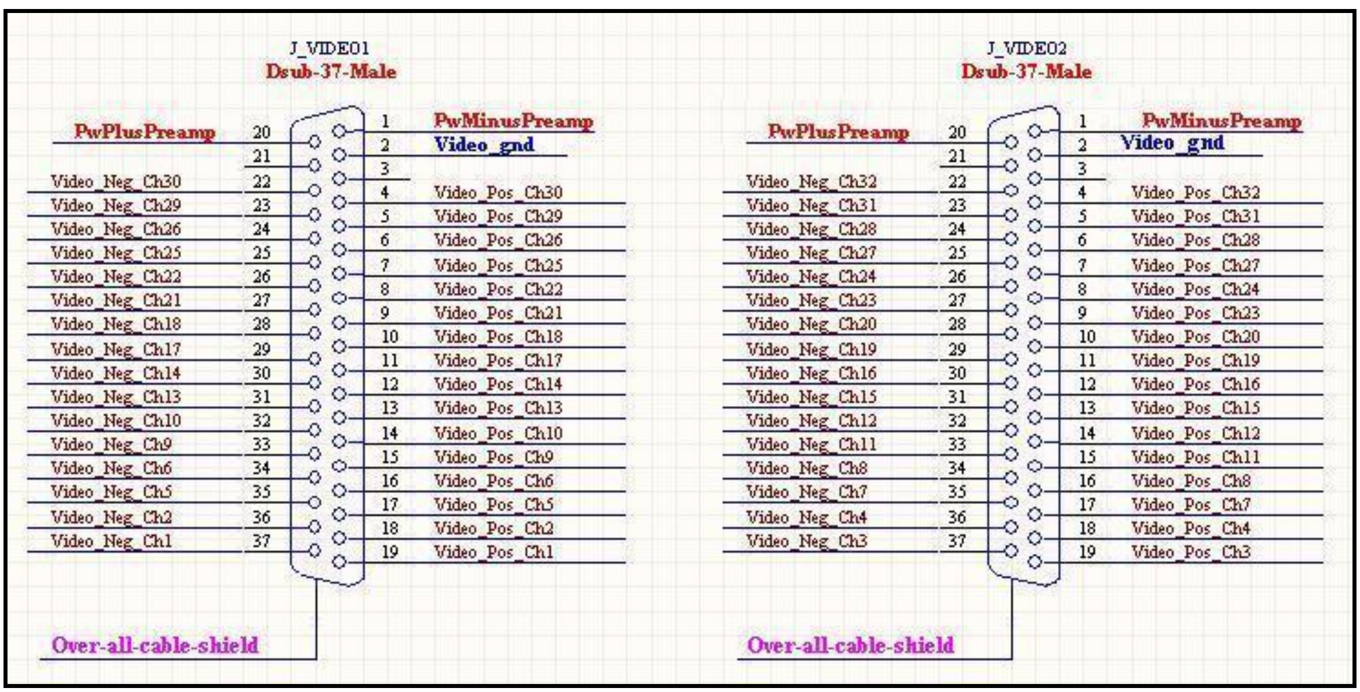


Figure 37 AQ32 Video Connectors

3.5 NGC Front-End Backplane

The NGC Front-End Backplane (Figure 38) connects the Front-end boards located in Slot 1 to 6. Power to the system is provided via the Power Connector. Separate lines out of clock multiplexers carry the clocks for the sequencer(s) and the rocket I/O of the FPGAs for each slot. The link connections to the individual slots and to the Transition Boards are point to point connections. All other connections like PowerUpReset, SEQ_RUN, CONVERTx, SEQ_BIT(x) and MODE(x) are parallel on all slots.

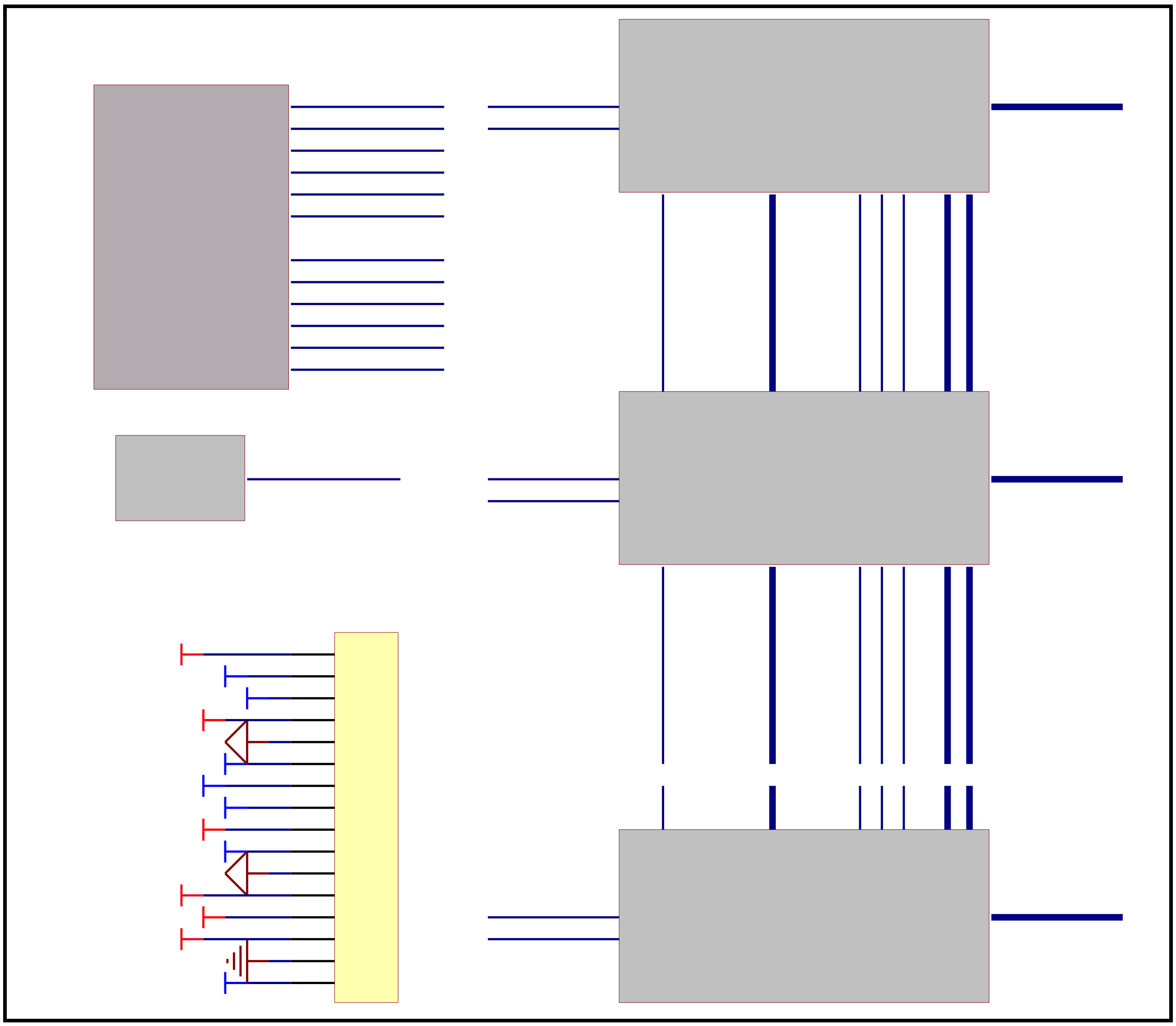


Figure 38 NGC Front-End Backplane

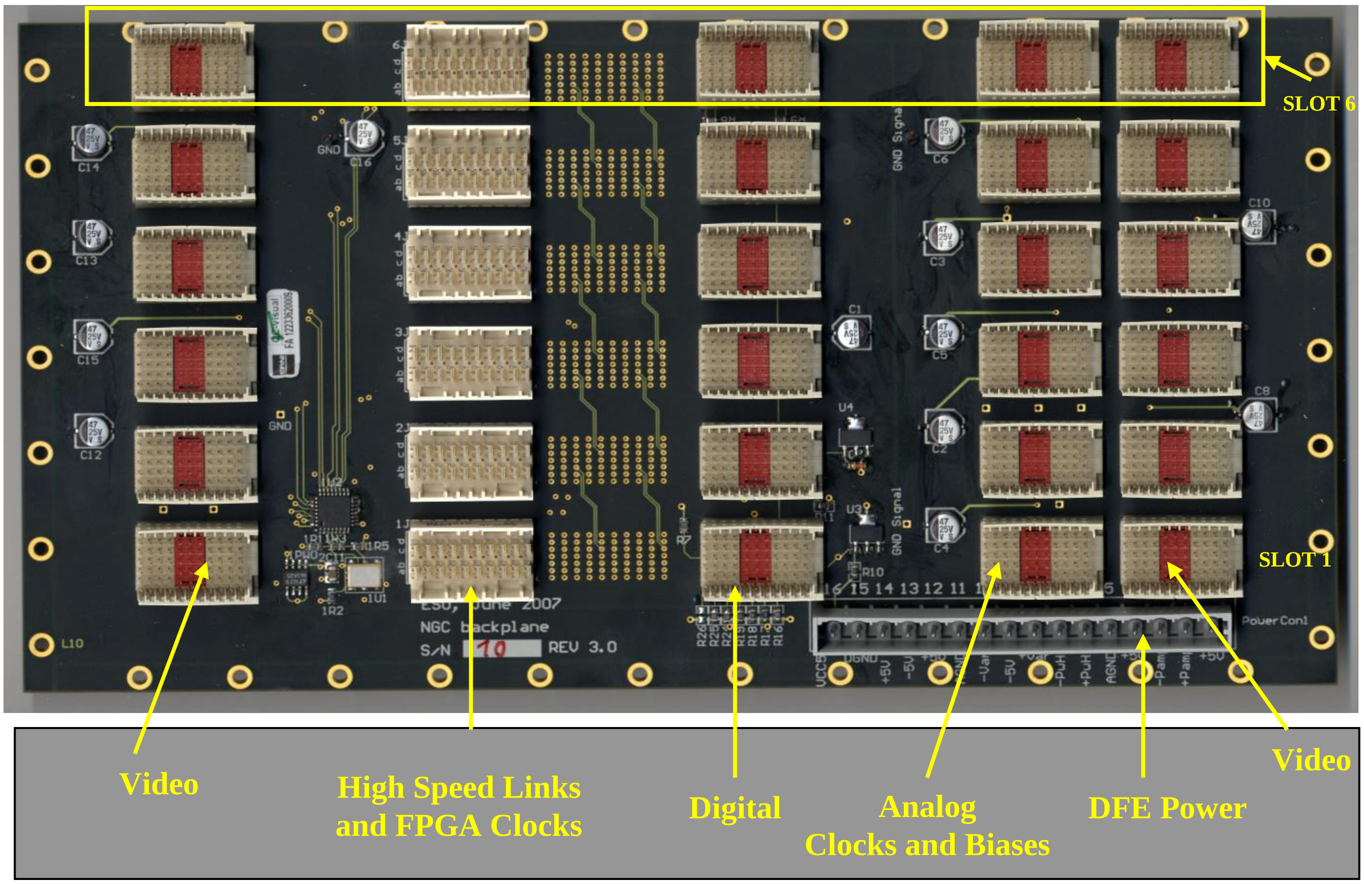


Figure 39 Backplane - Transition Board Side

3.6 PMC Based Low Latency DMA Channel

For interferometry use of NGC a DMA channel with minimal data latency is needed. The interferometry acquisition system is based on VME Motorola LCU's with a VxWorks operating system. The fastest way to transfer data with minimum latency is a PMC DMA interface on a PMC slot of the Motorola LCU. The PMC interface always strips off one header from each link packet and routes all modified packets through. In addition it recognizes video data packets from the Front-End link and writes the video data per DMA into the VME LCU.
The Front-End set-up and all communication comes as usual from the LINUX workstation PCI interface, the video data are available on the VME Motorola LCU and the LINUX workstation. Initial system tests can be carried out with all standard NGC software tools based on Linux. The PMC module does not contain a communication interface.

Hint: See the configuration of Figure 40 Example Interferometry System**:**
To write from Back-End PCI to the first Front-End module use 0x5 0x2 <Addr> <data>
In the Configuration Register (see Table 13) of the first Front-End module the *Number of UpStream links till Back-End* is 2

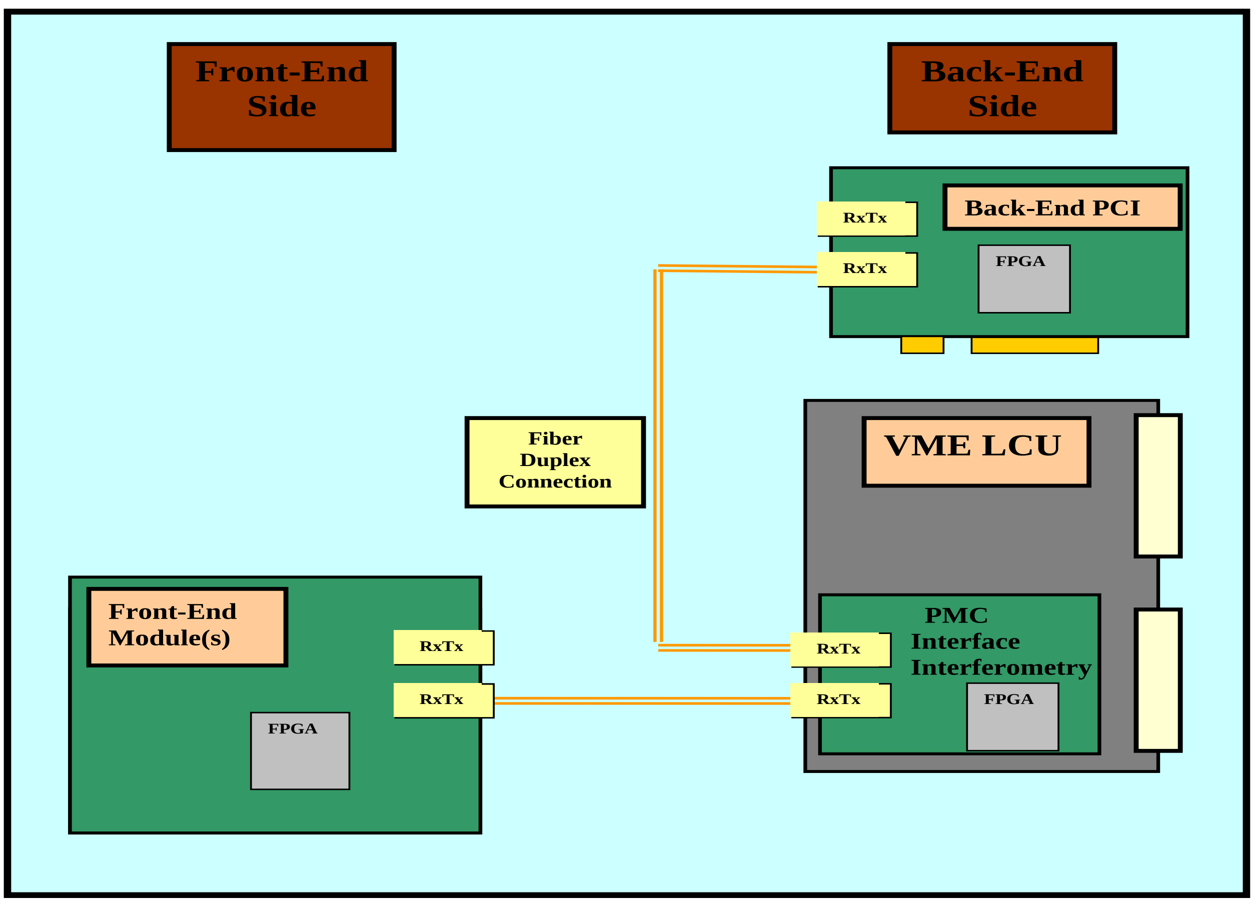


Figure 40 Example Interferometry System

3.6.1 PMC Interface

The PMC Interface is a 64 Bit DMA master interface to a PCI bus in PMC format (see Figure 41). It routes packets through and strips always one header off the packets. Video data packets from the Downstream link are recognized and written to the video fifo. From there DMA transfers to the PMC PCI bus can be executed. The register set is identical to the PCI Back-End module (see Table 50 and Table 51) with the exception that no PCI slave interface for communication is implemented.

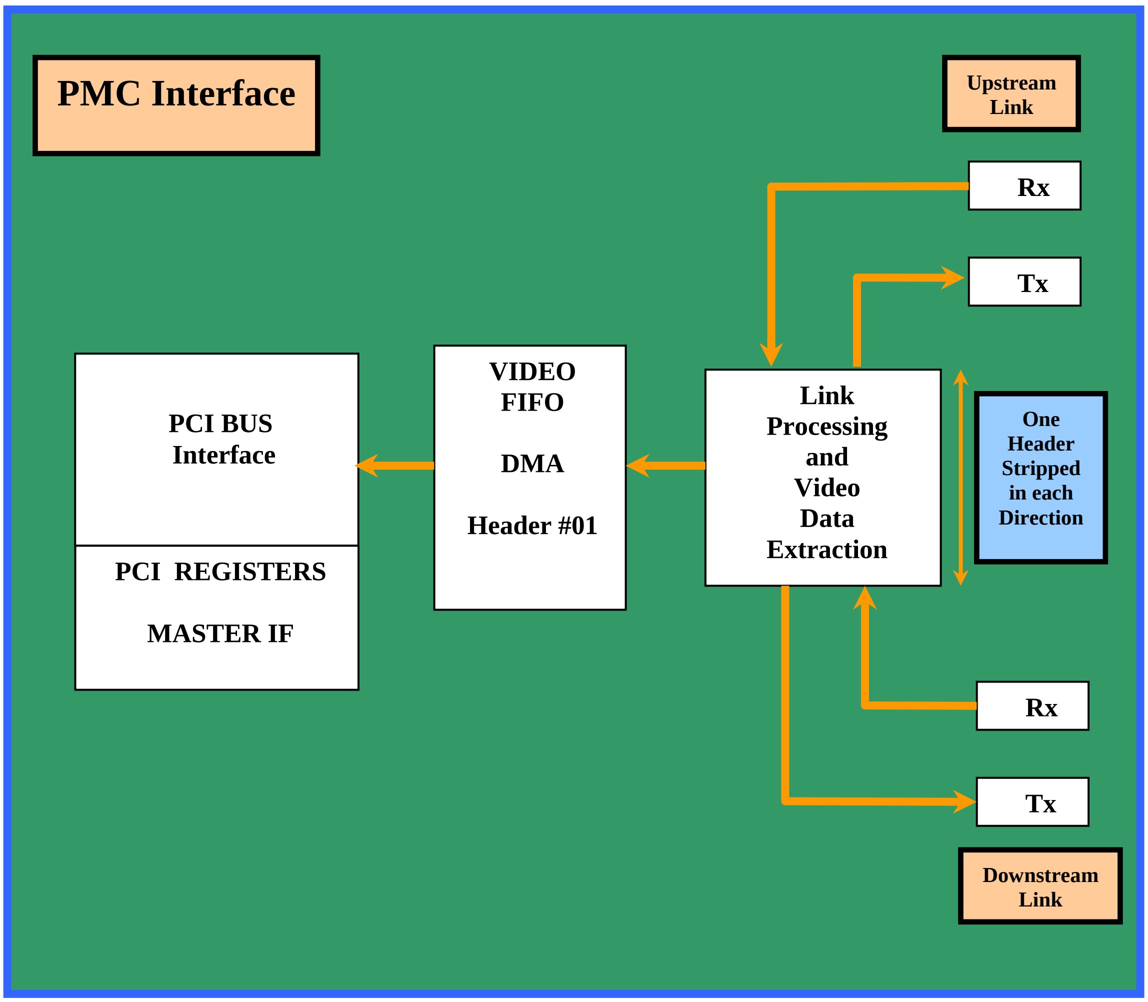


Figure 41 PMC Interface

Addresses

Table 50 Configuration Space

Relative address[Hex] Config Space		Function
C W		DMA Control Register
14 R		Communication Status Register
1C W		Communication Command Register
68 R/W		PCI_Descr_Pointer #90 ADI
84 R/W		PCI_Addr_Reg
8C R/W		PCI_ DMA_Counter_Reg
90 R/W		PCI_Descr_Pointer
A0 R		PCI_Board Revision and Date

Hint : Read from NGC Panel with “ ior local 0x<rel addr> “

Table 49 DMA Space

Relative address[Hex] DMA Space		Function
8 R		DMA Status Register

Registers

Table 51 DMA Status Register – Read (Addr #8)

Bit 5	**Data FiFo Empty**
Bit 6	**Data FiFo Full**
Bit 7	**Data FiFo Full – Write (Cleared by Clr Data FiFo)**

Table 52 Communication Command – Write (Addr #1C)

BIT 0	**Not Used**
Bit 1	**Clear Video FiFo**
Bit 4	**Not Used**
Bit 7	**Not Used**

Table 53 Communication Status – Read (Addr #14)

BIT 0	**Not used**
Bit 1	**Not used**
Bit 5	**Video FiFo Empty**
BIT 6	**Video FiFo Full**
Bit 7	**Overflow Error Flag (Write on Video FiFo Full - Cleared by Clr Video FiFo)**
Bit 8	**Upstream Link Channel Up**
Bit 9	**Upstream Link Hard Error**
Bit 10	**Upstream Link Soft Error**
Bit 11	**Upstream Link Framing Error**
Bit 12	**Downstream Link Channel Up**
Bit 13	**Downstream Link Hard Error**
Bit 14	**Downstream Link Soft Error**
Bit 15	**Downstream Link Framing Error**

Table 54 Interrupt_Ctr_Reg (Addr #68)

Bit 21	**Interrupt Flag**

Table 55 PCI_Addr_Reg (Addr #84)

Bit 0..31	**DMA Address**

Table 56 PCI_DMA_Counter (Addr #8C)

Bit 2..12	**DMA Count**

Table 57 PCI_Descr_Pointer (Addr #90)

Bit 4..31	**Initial DMA Descriptor**

Table 58 PMC Interface (Addr #A0)

PMC Interface HW Revision 1
Firmware Revision 1.1)

Bit (3 .. 0) Board Type	**Value = 1 → Basic Board** **2 → AQ32** **8 → PCI Backend** **9 → PMC Interface**
BIT 0	**1**
Bit 1	**0**
Bit 2	**0**
BIT 3	**1**
Bit (7 .. 4) Board Sub-type	
Bit 4	**1**
Bit 5	**0**
Bit 6	**0**
Bit 7	**0**
Bit (11 .. 8) HW Revision	
Bit 8	**1**
Bit 9	**0**
Bit 10	**0**
Bit 11	**0**
Bit (15 .. 12) Firmware Revision	
Bit 12	**1**
Bit 13	**0**
Bit 14	**0**
Bit 15	**0**
Bit (19 .. 16) Firmware Sub-Revision	
Bit 16	**1**
Bit 17	**0**
Bit 18	**0**
Bit 19	**0**
Bit 31 .. 20	**reserved**

Table 59 DMA Command Register – Write only (Addr #A8)

BIT 1	**Start DMA**
Bit 2	**Abort DMA**
Bit 3	**Clear Interrupt**