ORGANISATION EUROPEENNE POUR LA RECHERCHE NUCLEAIRE
EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH

Laboratoire Européen pour la Physique des Particules
European Laboratory for Particle Physics

THE K-CHIP REFERENCE MANUAL

DRAFT

V3A.1

Author: Kloukinas Kostas, CERN EP/CME

Version: 3A.1

Date: November 10, 2001

Last Update: December 19, 2003

Notes: This document is a draft version of the K-chip reference manual.
It describes the "Kchip_V3" version of the Kchip.

Document History

Version 0.1 DRAFT: this is the first draft of the K-chip reference manual. All included information should be considered as *target specifications*.

Version 0.2 DRAFT: this version of the manual includes the design changes on the chip in order to interface with the **PACE3 chip** and the **AD41240 0.25µm ADC**. All included information should be considered as *target specifications*.

Version 0.3 DRAFT: this version of the manual is an update of the previous version. All included information should still be considered as *target specifications*.

Version 3A.1: this version of the manual is an update of the previous version. Corrections: a)the width of the Bunch Counter is 12-bit and not 16-bit on page 7 and b)reassignment of the Trigger Commands in Table2-1 on page10. This manual reflects the functionality implemented in “**Kchip_V3**” version of the Kchip. This version of the chip will be replaced by “**Kchip_V3b**”.

Design Team

Kloukinas	Kostas,	EP/CME,	e-mail:	Kostas.Kloukinas@cern.ch
Bonacini	Sandro,	EP/MIC,	e-mail:	Sandro.Bonacini@cern.ch

Table Of contents

1 Introduction

This document describes the architecture and the implementation of the K-chip ASIC.

1.1 Preshower Front End System

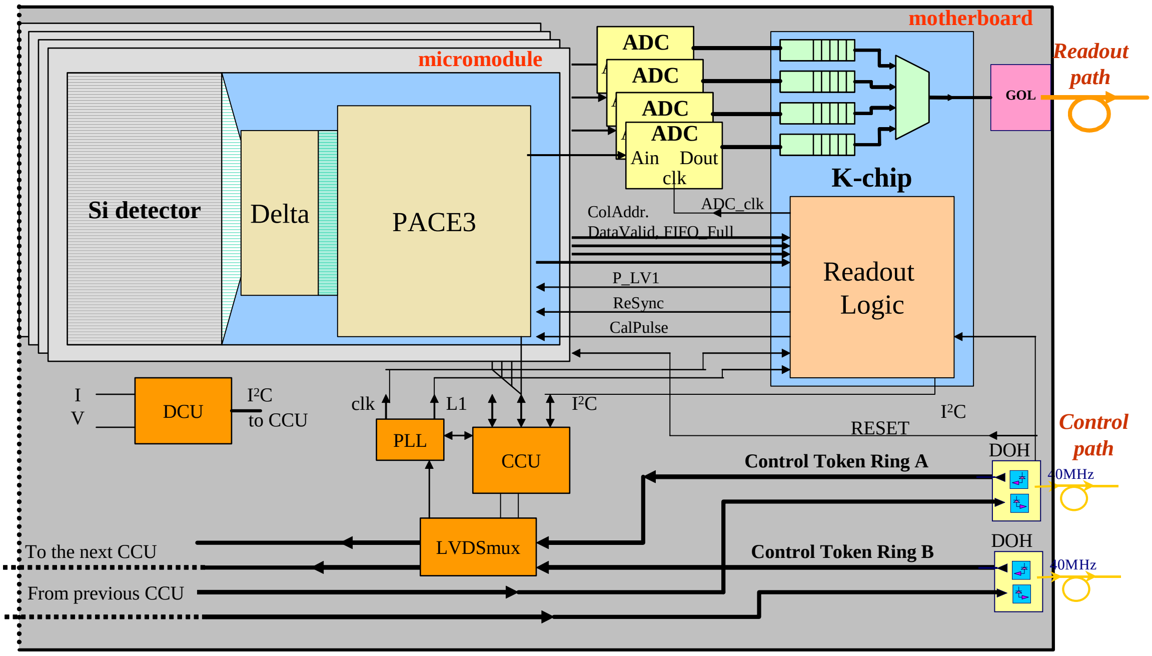


Figure 1-1 The general system architecture of the Preshower Front End Readout electronics.

2 The K-chip

2.1 Functional Description

The K-chip serves two functions:

- **Data Merging** from four PACE chips and
- PACE **Readout Control**

2.1.1 Data Merging

As a Data Merger the K-chip function is to gather data from four PACE chips and format it in a way suitable to be sent over the high-speed digital link. As no data reduction occurs in the front-end system, the K-chip has to cope with the data rates produced by the PACE front-ends; this is detailed in the following sections.

The K-chip receives data in parallel from the PACE chips, builds a packet in a format suitable for the high-speed serial link and sends this to the remote FED cards.

While preparing this data block the K-chip adds some more information to it such as:

- an event number tag
- a bunch counter tag
- CRC information
- error information, if necessary

As the input data comes in 12 bit format, the K-chip aligns the data in contiguous blocks of 16 bits as to maximally utilize the link bandwidth. The data analysis engine in the FED cards will have to unpack the data into 12 bit wide words.

The dataflow used is a simple push type architecture. All K-chips in the system are synchronous and transmit data along their link at the same time, as no data reduction is performed until the FEDs level. Event data prepared in the K-chip buffers are sent to the serial link transmitter. To simplify the system and reduce its cost this link is unidirectional and without flow-control. This means that whenever an error occurs in the transmission medium (either the serializer, the link itself or the receiver) a block of data belonging to one (or potentially even more) event(s) is irremediably lost. As another consequence, the FED has to be able to regain synchronization when one of the K-chips or links is sending data under some error condition.

2.1.2 PACE Readout Control

The K-chip generates the fast timing signals for the PACE operation. These signals are:

- the 40MHz system clock,
- the PACE 1st Level Trigger pulse (P_LV1),
- the FE readout Synchronization pulse (ReSync), and
- the Calibration Pulse (CalPulse).

The K-chip supervises the readout operation of the PACE chips that are connected to its four channel ports by monitoring the following PACE signals:

- The PACE Readout Frame qualifier signal (DataValid),
- The Column Address serial information (Col_Addr),
- The PACE Trigger FIFO Almost Full flag (FIFO_Full).

The K-chip can identify the following error conditions in the Readout Operation of the PACE chips:

- PACE out of Sync condition is signaled when the readout sequence in one or more PACE chips is not synchronous with the K-chip internal readout operation.
- PACE Trigger FIFO almost full condition.

2.2 Block Diagram

A simplified block diagram of the K-chip in the read-out system is shown in Figure 2-1.

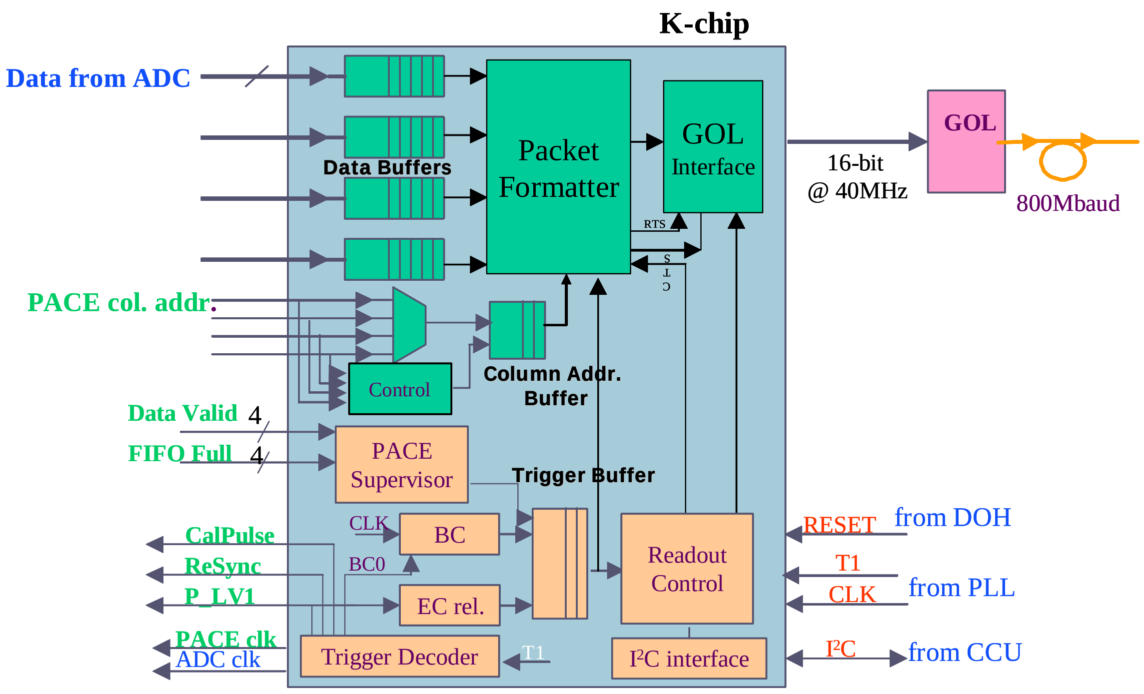


Figure 2-1 K-chip block diagram.

The K-chip consists of the following elements:

- Four Data FIFO buffers, one per input channel to store the incoming events. These are 12-bit wide FIFOs and several events deep, see below.
- A Column Address FIFO buffer, to store the column addresses of the samples of the incoming events. This is a 16-bit wide FIFO and several events deep, see below.
- A Packet Formatter used to scan the four Data FIFOs and the Column Address FIFO when a complete event has to be assembled. The Packet Formatter also aligns the 12-bit wide data to 16-bit wide format.
- A Gigabit Optical Link Interface to encapsulate the event data with the proper header and trailer information in order to transmit them through the optical link.
- A 12-bit Bunch Counter (BC) used to count continuously the incoming 40MHz system clock.
- An 8-bit Event Counter (EC) used to count the number of incoming LV1 triggers.
- A Trigger Decoder circuit in order to decode the Trigger Commands that are received through the T1 signal from the PLL chip.

- A Trigger FIFO used to store incoming triggers while the readout of a previous event is in progress
- A control logic to provide the synchronization of the entire ASIC and the supervision of the sequence of operations necessary to build an event in the output buffer.
- A PACE Readout Supervisor logic that monitors the synchronicity of the readout operation of the PACE chips that are connected to the Kchip.
- A set of user registers, accessible through the chip's slow control port (I^2C). The uses of these registers are to control and read back status information from the K-chip. They provide also the possibility to the user to write some pseudo-event data into the data FIFOs to test the functionality of the readout chain.
- An I^2C based slow control interface used to access the K-chip internal registers and data FIFOs.

The simplified protocol followed by the K-chip to assemble one event into its output buffer is the following:

- the K-chip monitors continuously the state of its trigger FIFO
- when a trigger is pending in the trigger FIFO, the Packet Formatter extracts it together with the bunch counter tag which was stored in it at the moment of the arrival of the trigger signal and stores this in the header of the outgoing data packet with the K-chip ID number
- the data blocks at the head of the four input FIFOs are read and moved to the GOL interface.
- a CRC is computed and appended to the event data packet
- the packet is streamed out through the high speed readout link.

2.3 Operation Modes

The K-chip can be initialized in two modes:

- normal read-out mode
- link test mode

In the first mode, the K-chip assembles event blocks as described above and it used in the normal data acquisition chain. In the link test mode, the K-chip can send out data which can be written into its input FIFOs via the slow control interface. This mode is used essentially as an aid to debug a malfunctioning link outside of the normal data acquisition mode

2.4 PACE Interface

The Kchip is designed to interface with the PACE3 analog memory chip.

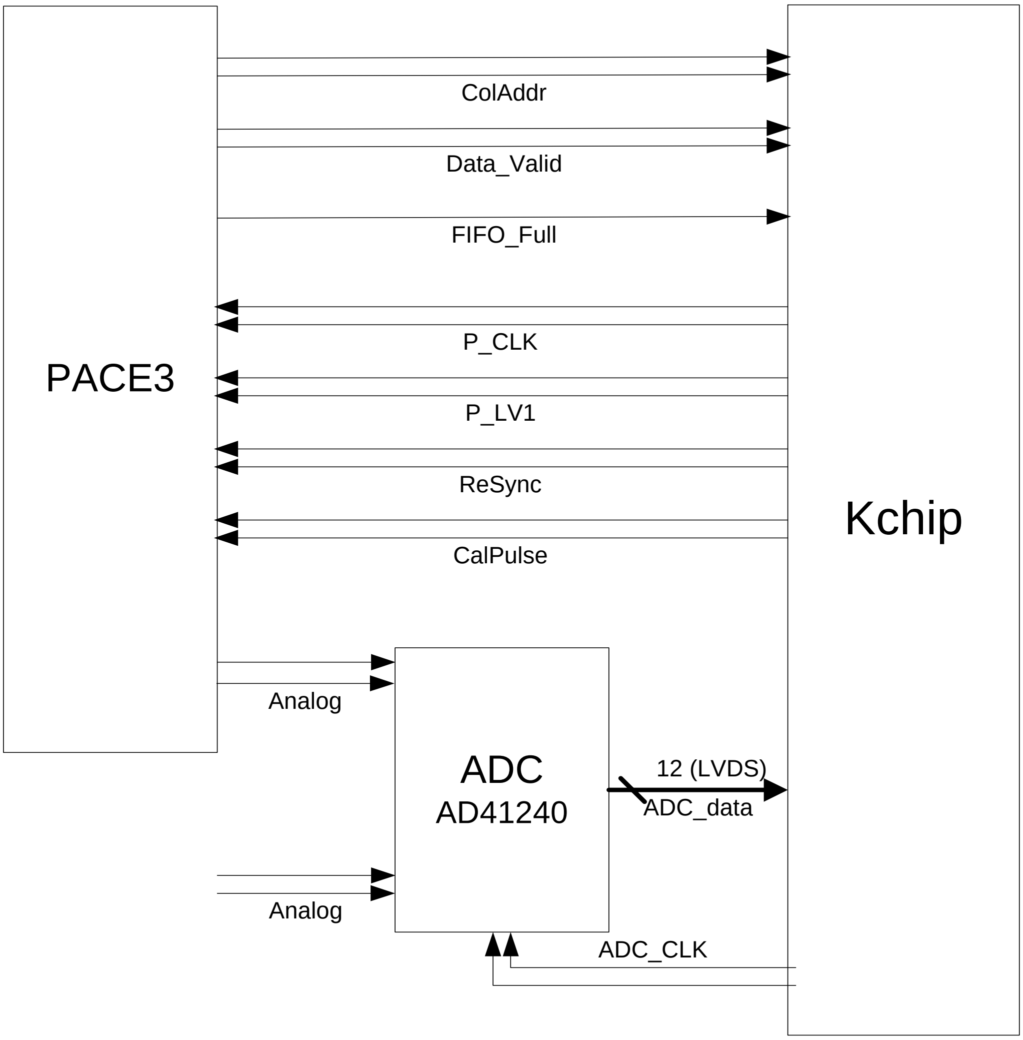
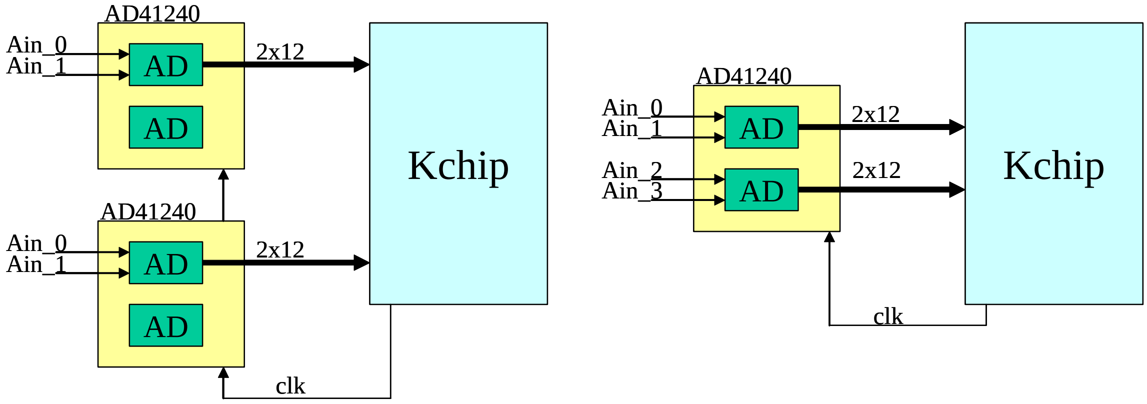


Figure 2-2 PACE3 to Kchip interface.

2.5 ADC interface

The Kchip is designed to interface with the AD41240 12bit-40Msample/s ADC, which is fabricated in the same 0.25µm CMOS technology as the Kchip. The AD41240 is a quad channel ADC with the possibility to power down individually its channels. The AD41240 has 2 12-bit data busses. On each data bus two ADC channels are multiplexed using the Double Data Rate technique.

The Kchip offers two possibilities to interface with the ADC as shown in Figure 2-3. In option (a) it is using two dual channel ADCs while in option (b) it is using a single quad channel ADC. The final choice will be made taking into account the general system aspects (motherboard design, ADC channel to channel crosstalk, noise, cost ....). The Kchip has a single LVDS clock output for the ADC.



a) 2 x dual channel ADC a) 1 x quad channel ADC

Figure 2-3 Two options for interfacing the Kchip with the AD41240 ADC chip.

2.6 Trigger Decoder

The encoded trigger uses three consecutive bits in the T1 stream to specify different conditions according to the following table:

Pattern	Command
100	LV1A (Trigger Level 1 Accept)
101	BC0 (Bunch Crossing Zero identifier)
110	ReSync (Reset FE Pipelines)
111	CalPulse (Calibration)

Table 2-1 Coding of the Trigger Commands.

The actions taken by the Kchip logic when a Trigger Command is issued are listed below.

- **LV1A:** sends an trigger pulse to the PACE3 chip and increments the EC and inserts a normal event in the trigger FIFO.
- **CalPulse:** send a calibration pulse to the PACE3 chip and starts countdown of the latency counter. Upon timeout it increments the EC and inserts a calibration event in the Trigger FIFO.
- **ReSync:** resets the EC and BC and clears the Data, Column and Trigger FIFOs. Resets PACE supervisor logic error flags.
- **BC0:** resets the EC and BC counters.

2.7 K-chip Buffers

The Kchip has three different types of buffers;

- The **Data FIFO** for storing the digitized samples of the trigger events. There are four identical Data FIFOs, one for each Kchip input channel.
- The **Column Address FIFO** for storing the column addresses that correspond to the trigger events stored in the Data FIFO. There are four identical Column Address FIFOs, one for each Kchip input channel.
- The **Trigger FIFO** for storing the time tags (Bunch counter, event Counter) and control information of the incoming triggers. There is one Trigger FIFO in the Kchip.

2.7.1 Buffer Sizes

The size of the input buffers in the K-chip determines the probability of losing an event because of a momentary congestion.

Using a software emulation model of the system we have determined the size of the three different buffer types. The simulation results are shown in Figure 2-4.

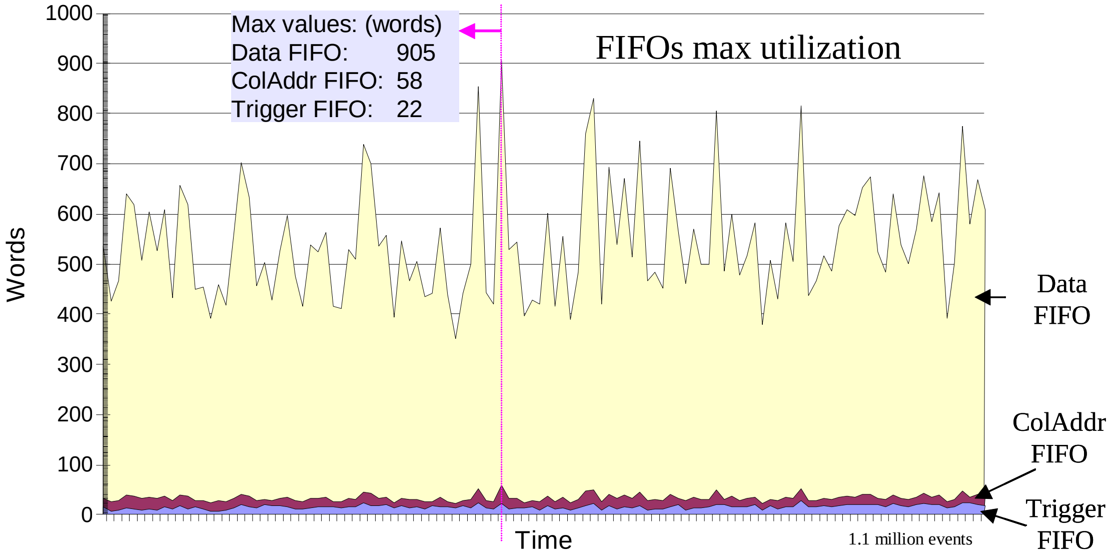


Figure 2-4 Simulation results of the Kchip at 100KHz trigger rate presenting the traffic in the Data FIFO, Column FIFO and Trigger FIFO.

The FIFO sizes implemented in the Kchip are listed in Table 2-2.

Table 2-2 FIFO sizes in the Kchip.

FIFO	Size
Data	1 Kword x 18 bits
Column Address	128 words x 27 bits
Trigger	128 words x 27 bits

2.7.2 Data FIFO

The format of the information stored in the Data FIFOs is shown in Figure 2-5.

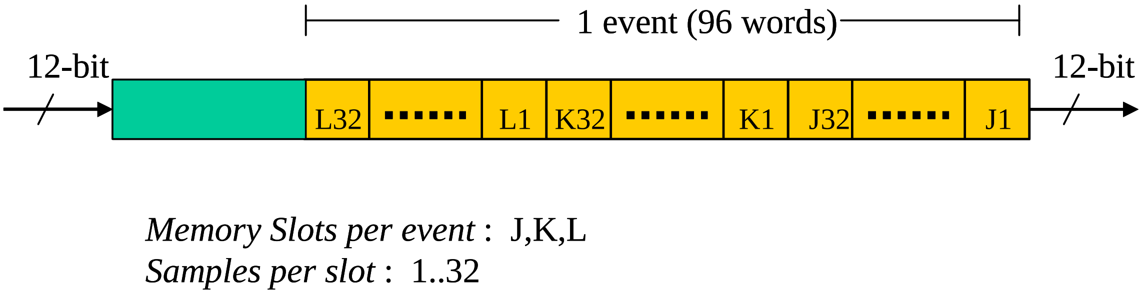


Figure 2-5 Format of the information stored in the Data FIFOs.

Having a size of 1Kword, the Data FIFO can store up to 10 events (1024/96).

2.7.3 Column Address FIFO

The format of the information stored in the Column FIFOs is shown in Figure 2-6.

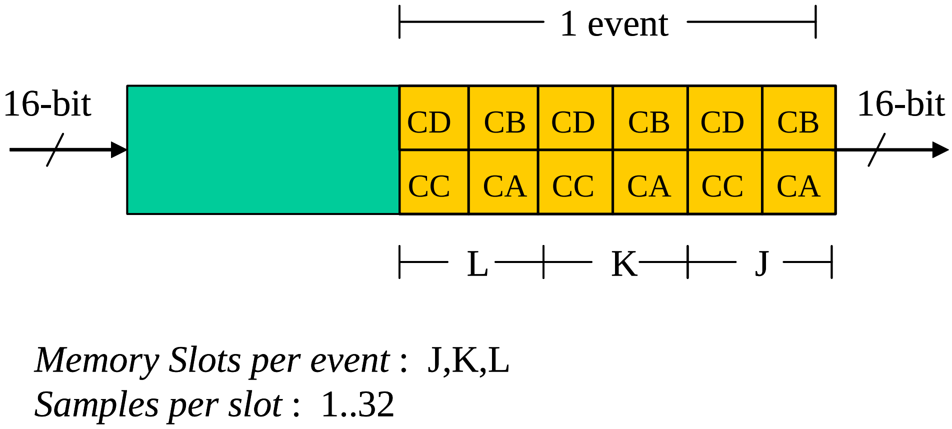


Figure 2-6 Format of the information stored in the Column FIFOs.

2.7.4 Trigger FIFO

The format of the trigger events stored in the Trigger FIFO is shown in Figure 2-7.

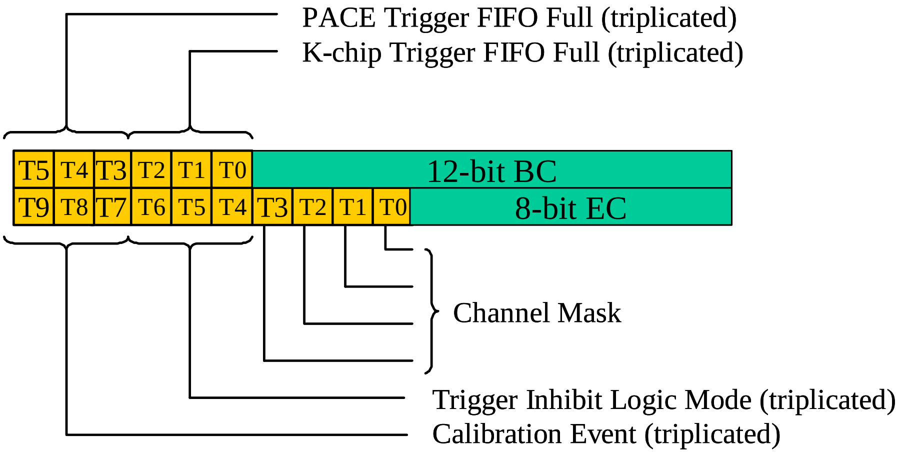


Figure 2-7 Record of an Event in the Trigger FIFO (18-bits wide).

Having a size of 128words, the Trigger FIFO can store up to 64 triggers (128/2).

The Trigger FIFO is seen by the user as a 14-bits wide FIFO. Actually is 18-bits wide but the 6 MSBs are used in a triplicated manner to code two bits, see Figure 2-7.

2.8 PACE supervisor

The Kchip provides a mechanism for checking the synchronicity of the operation of the four PACE chips that are attached to its input channels. The mechanism is using a cycle-by-cycle comparison of the PACE readout control signals with the Kchip internally generated control signals. The results of the comparison are flagged in the Event Data Packet transmitted on the Data Link and in the Kchip Status Register.

There are two readout control signals generated by the PACE chip:

- The **DataValid** signal that indicates the boundaries of the events as transmitted by the PACE chip.
- The **PACE FIFO Almost Full** signal that indicates the availability of free locations in the PACE Trigger FIFO for storing more trigger events.

These readout signals are individually compared with the Kchip internally generated readout signals as shown in Figure 2-8. The logic of the comparison is common for the two signals.

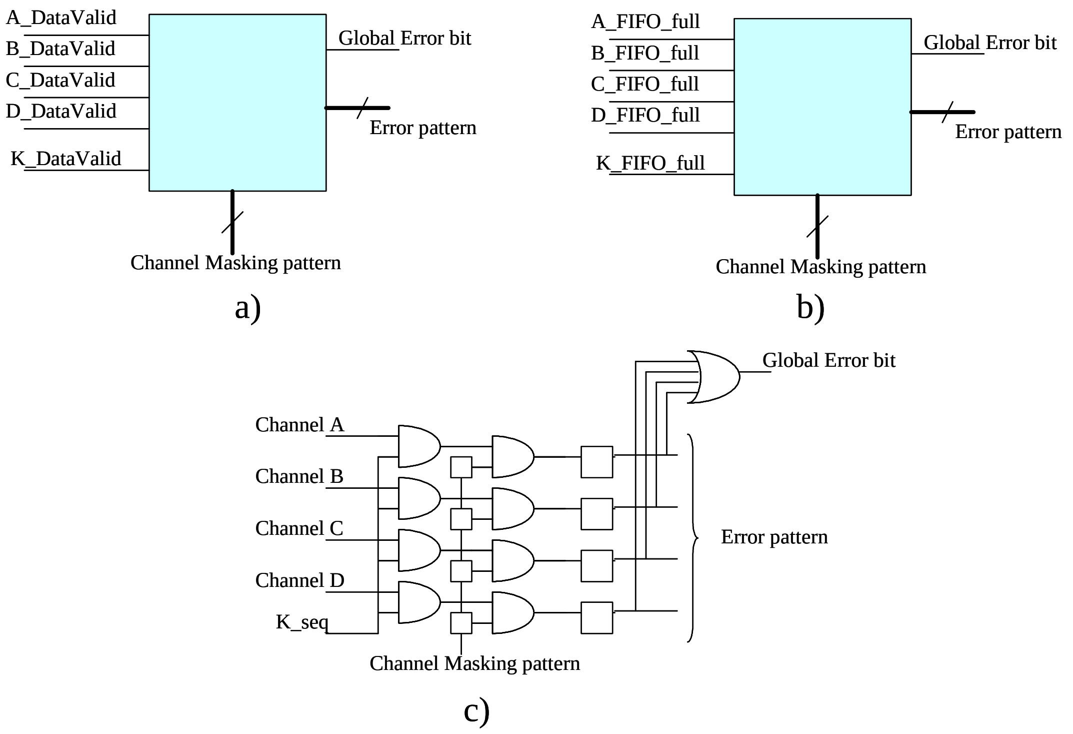


Figure 2-8 a) Comparison logic for the DataValid signal, b) comparison logic for the PACE FIFO Almost Full signal, c) schematic diagram of the comparison logic.

2.9 Kchip Data Path design

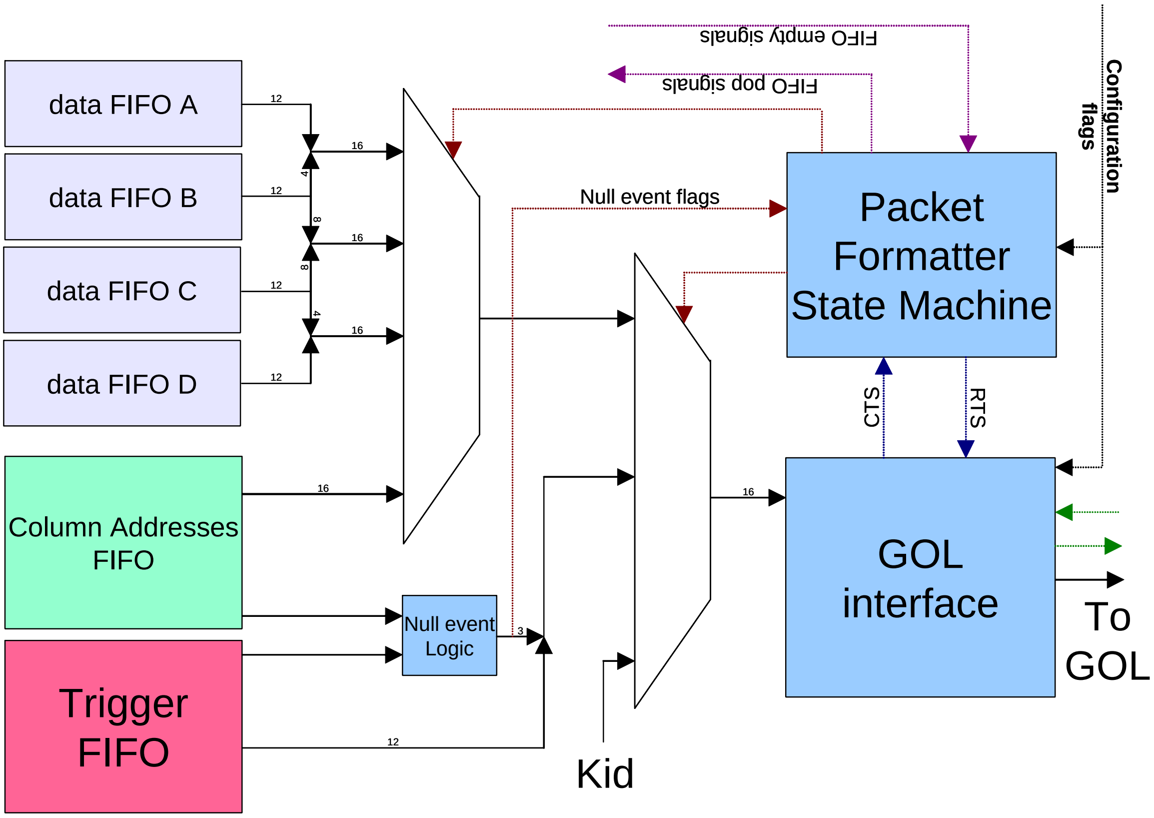


Figure 2-9 The Kchip Data Path.

2.10 Error Conditions / Error handling

2.10.1 Buffer Overflows

2.10.2 Synchronization of PACE3 chips

2.11 Packet Formatter

2.11.1 Link Data Packet Format

Figure 2-10 shows the mechanism that the K-chip employs to rearrange the incoming data in the four input channels.

The format of the data transmitted through the High Speed Link is shown in Figure 2-11 & Figure 2-12.

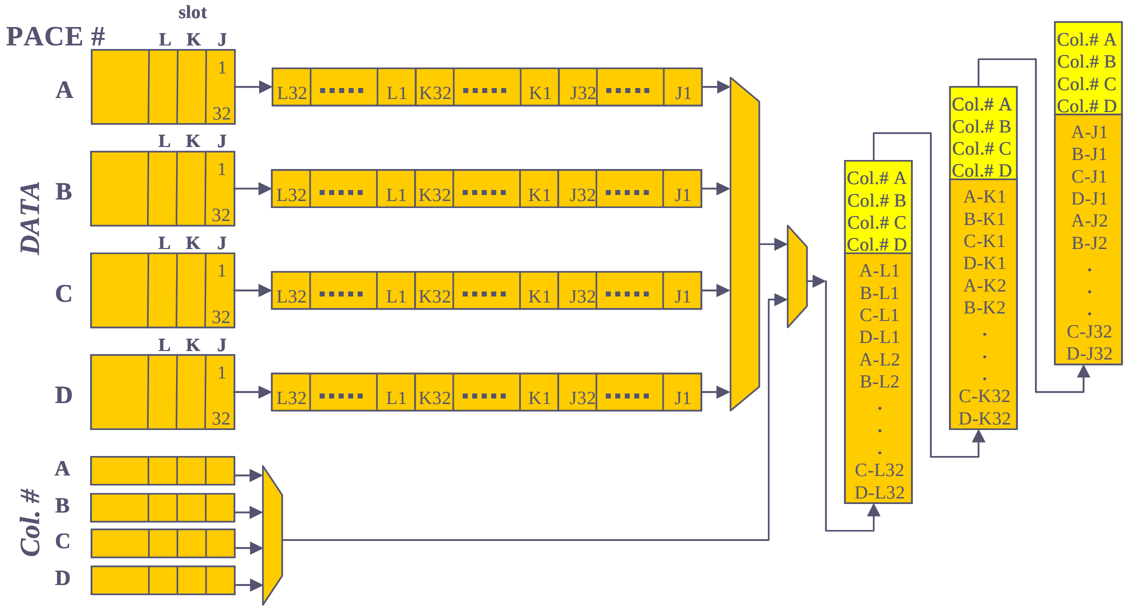


Figure 2-10 Mechanism for the Event Data Formatting.

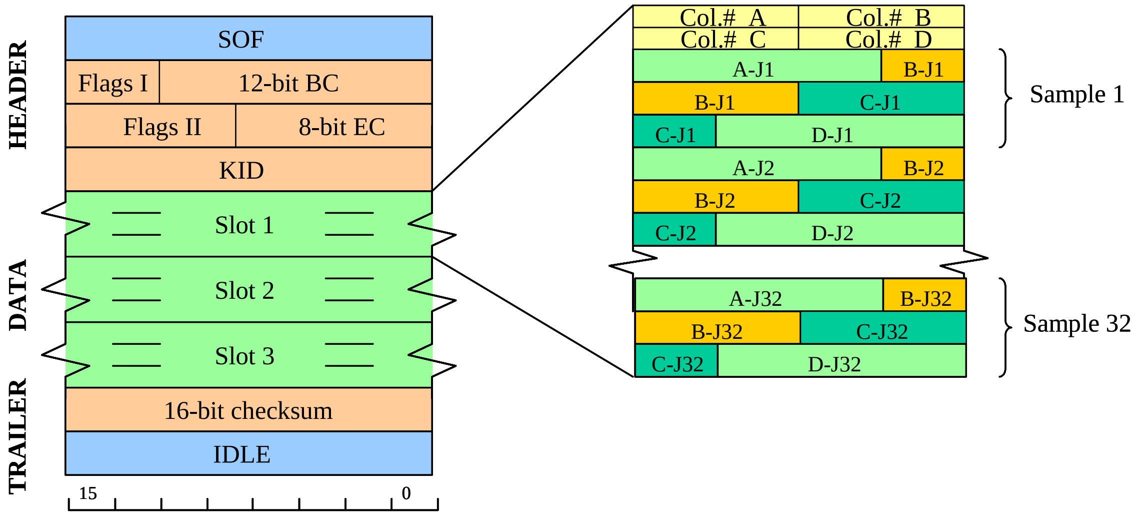


Figure 2-11 Link Data Packet Format.

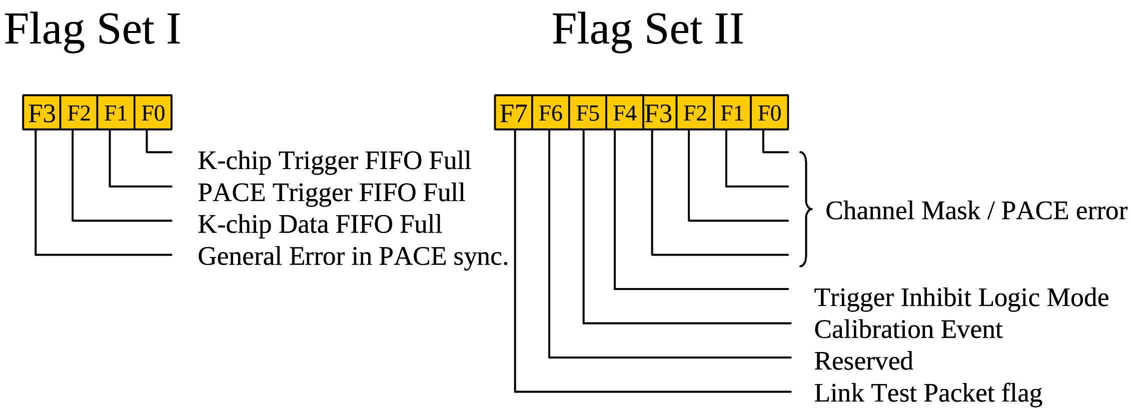


Figure 2-12 Link Data Packet Format.

Table 2-3 Explanation of the flags in the data packet.

Flag	Function
Kchip Trigger FIFO full	This flag is set only in NULL events.
PACE Trigger FIFO full	This flag is set only in NULL events.
Kchip Data FIFO full	This flag is set only in NULL events.
General Error in PACE	When this bit is set to ‘1’ it indicates that a PACE synchronization error has occurred.

Channel Mask / PACE error	When the “General Error in PACE” flag is set to ‘0’ these four bits indicate the active data channels in the Kchip. When the “General Error in PACE” flag is set to ‘1’ these four bits indicate the corresponding channel(s) that has the PACE chip(s) found to be out of sync.
Trigger Inhibit Logic	When this bit is set to ‘1’ it indicates that the Trigger Inhibit logic in the Kchip is ENABLED. When this bit is set to ‘0’ it indicates that the Trigger Inhibit logic in the Kchip is DISABLED.
Calibration Event	When this bit is set to ‘1’ it indicates that the current data packet being transmitted belongs to a Calibration Event.
Link Test Packet	When this bit is set to ‘1’ it indicates the transmission of a packet for link test purposes.

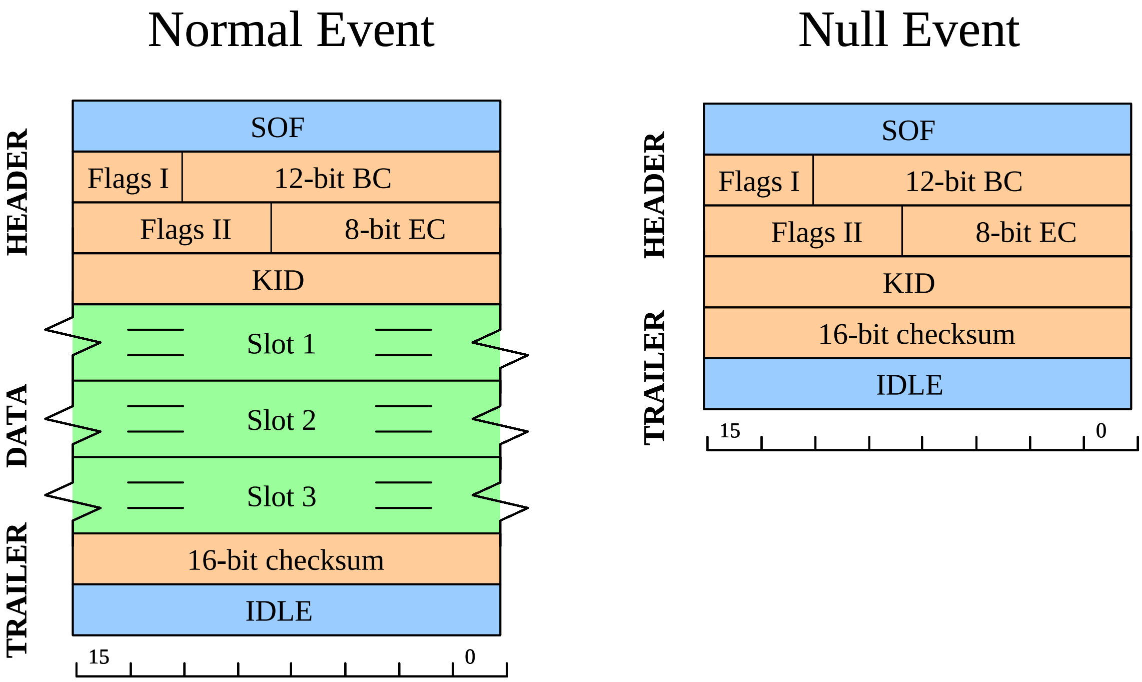


Figure 2-13 Packet Format of the Normal Event and the Null Event.

The Link Data Packet starts with a Header Field followed by the Data Payload and ends with a Trailer Field.

The Header Field consists of:

- a Start Of Frame (SOF) word which is used to synchronize the readout operation
- two sets of Flags of 12-bits in total for signaling the type of Data Packet and various Error Conditions
- an 8-bit Event Counter (EC)
- a 12-bit Bunch Counter (BC)
- a 16-bit K-chip IDentification number (KID)

The Data Payload Field consists of 3 identical data packets each containing information coming from a time slot. Each time slot data packet contains:

- the column addresses of the 4 PACE chips
- the 12-bit digitized values of the 36 data samples contained in one PACE column.
- The Trailer Field consists of:
- A 16-bit Cyclic Redundancy Checksum word as calculated over the whole information in the data packet except the SOF and EOF words. The CRC field is calculated following the CRC-16 algorithm.
- An End Of Frame (EOF) word, which is used to signal the end of the data packet.

The size of one **SLOT** in the Data Packet can be calculated as follows:

(4 PACEs x 32 samples x 12-bit)/(16-bit) + 2 words (*column address*) = 98 words of 16-bit.

The size of the **Data Payload** is then: 3 slots x 98 words = 294

The size of the **Normal Data Packet** is then calculated as:

4 words (*Header*) + 294 words + 1 words (*Trailer*) = 299 words.

2.12 Gigabit Link Interface

2.12.1 Physical Layer

The Kchip is designed to interface with the Gigabit Optical Link (GOL) chip. The Kchip – GOL interface runs at 40MHz over a 16-bit bus thus allowing a net data throughput of 640Mbps or 80Mbyte/sec.

In the case where the quality (amount of jitter) of the clock signal provided by the Tracker PLL is inadequate to operate the GOL chip the possibility of using a PLL employing a crystal oscillator (QPLL) is foreseen. Figure 2-14 shows the Kchip GOL chip interfacing.

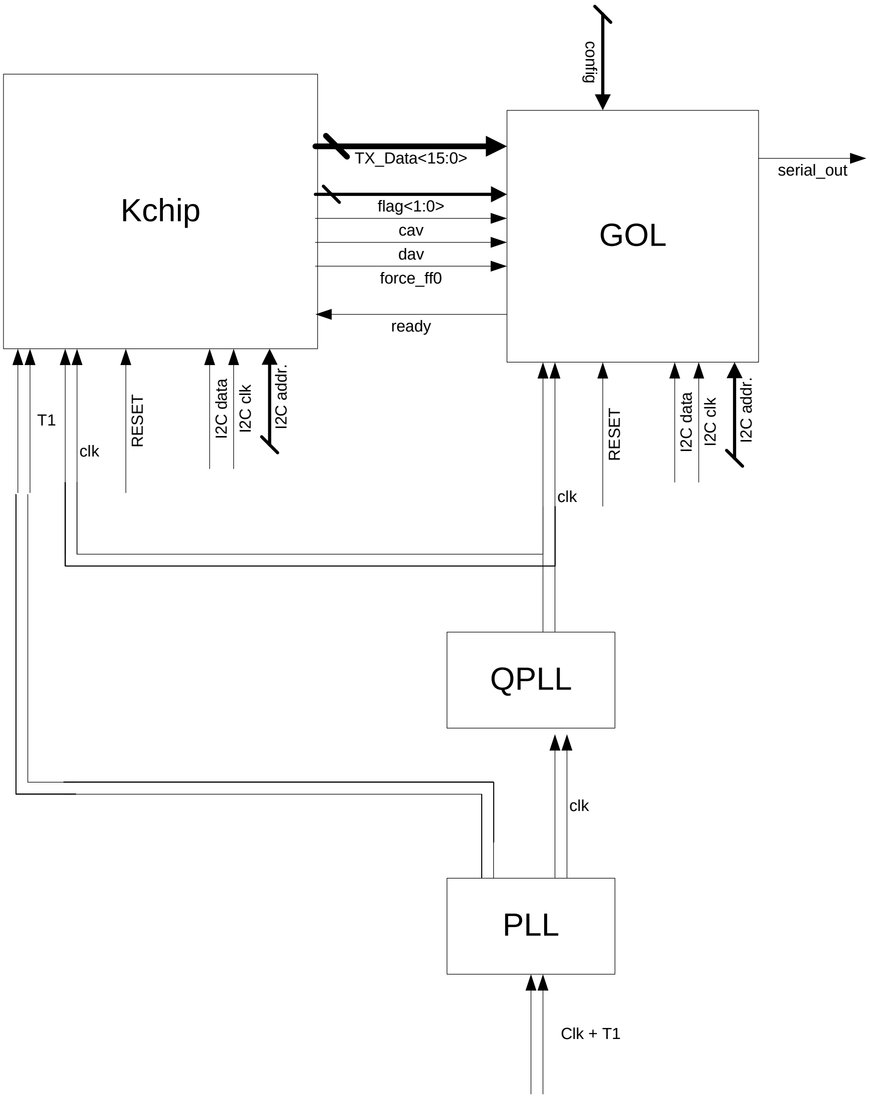


Figure 2-14 Kchip - GOL chip interface employing the QPLL for clock generation.

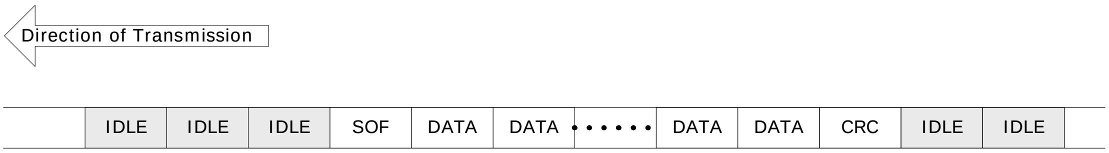
2.12.2 Link Layer

The Kchip employs a character oriented data transmission protocol. To achieve character synchronization the Kchip uses two uniquely defined transmission control characters, the IDLE and the SOH. The general data frame format is shown in Figure 2-15.

The IDLE character has two functions. Firstly, allows the receiver to obtain and maintain bit synchronization. Secondly, once bit synchronization has been acquired, allows the receiver to start to interpret the received bit stream on the correct character boundaries thus obtaining character synchronization.

The SOF character indicates the beginning of the frame and delimits the boundaries of subsequently transmitted frames (back-to-back transmission). Once the receiver has obtained character synchronization (and hence reading each character on the correct bit boundary), the receiver starts to process subsequently received character in search of the SOH character indicating the start of the frame. On receipt of the SOF character, the receiver proceeds to receive the frame contents and terminates this process.

General Frame Format



	IDLE	IDLE	IDLE	SOF	DATA	DATA	• • • • • •	DATA	DATA	CRC	IDLE	IDLE	

Figure 2-15 General Frame format.

The GOL chip supports two encoding schemes for the transmitted data, the CMIT protocol and the 8b/10b encoding protocol. The Kchip can seamlessly use both of the two transmission schemes without the need of any modification in the wiring or configuration register programming. To select the encoding scheme the user has to configure the mode selection pin of the GOL chip.

The flexibility of using both encoding schemes is realized by properly choosing the transmission control symbols (SOF, IDLE) which are supported in both encoding schemes and by implementing a link synchronization mechanism that can be applied in common for the two deserializers types used in the two encoding schemes.

The Kchip data transmission protocol employs an Error Detection mechanism to identify errors that can occur on the data links. The error detection mechanism is based on the transmission of a frame check sequence character placed at the end of the data frame. The frame check sequence character is a 16-bit CRC for the complete frame contents enclosed from the SOF character to the end of the frame. The CRC (Cyclic Redundancy Checking) generator polynomial in use is the CRC-CCITT: $x^{16} + x^{12} + x^5 + 1$.

2.12.3 Using CIMT protocol

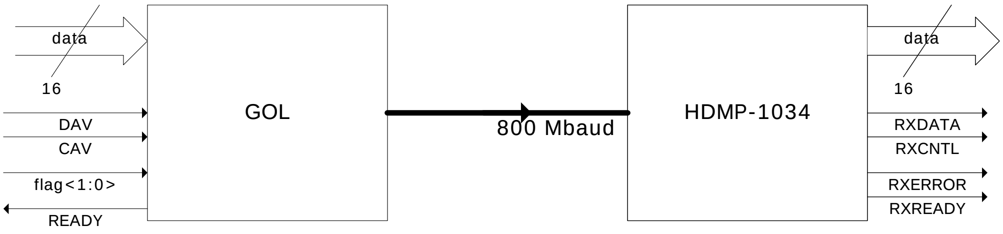


Figure 2-16 Link structure (serializer-deserializer) when using CIMT protocol.

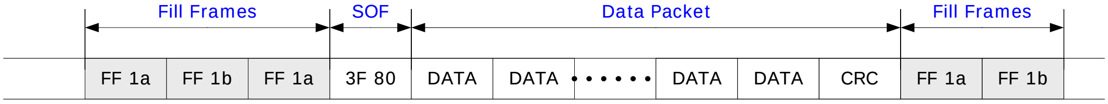


Figure 2-17 Packet format in CIMT protocol.

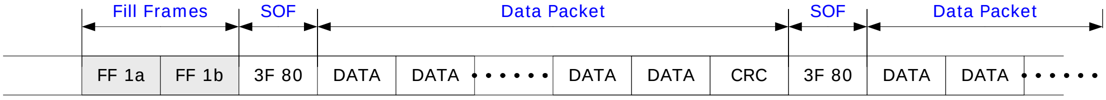


Figure 2-18 Packet format in CIMT protocol for back-to-back events.

2.12.4 Using 8b/10b encoding protocol

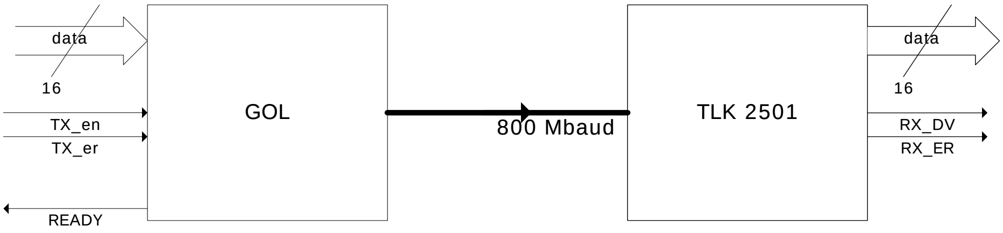


Figure 2-19 Link structure (serializer-deserializer) when using 8b/10b encoding protocol.

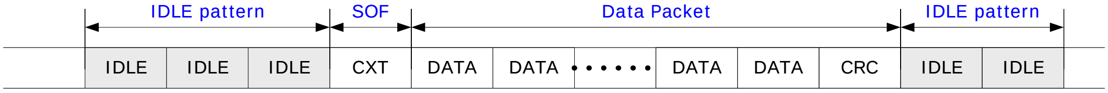
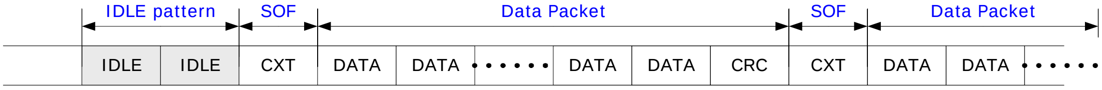


Figure 2-20 Packet format in 8b/10b protocol.



IDLE = <K28.5, D5.6> or <K28.5, D16.2> : Idle
CXT = <K23.7, K23.7> : Carrier Extend

Figure 2-21 Packet format in 8b/10b protocol for back-to-back events.

2.13 Link Synchronization Issues

The only way to synchronize the transmitter – receiver pair on the data links is to force the transmission of a stream of IDLE characters.

Transmission of IDLE characters commences:

- After hardware RESET has been issued and will last until data are present for transmission.
- After a ReSync has been issued and will last until data are present for transmission.
- When the "FORCE_IDLE" bit is asserted in the packet formatter.
- While the "GOL_READY" bit is de-asserted in the "STATUS_0" register.
- When the programmable counter (counting transmitted data packets) time outs. A sync pattern of programmable length is transmitted in programmable intervals.

2.13.1 Link Test Mode

A special "Link Test Mode" is implemented in the Kchip logic in order to facilitate the testing of the Gigabit Optical Links connecting the Kchips with the off-detector electronics.

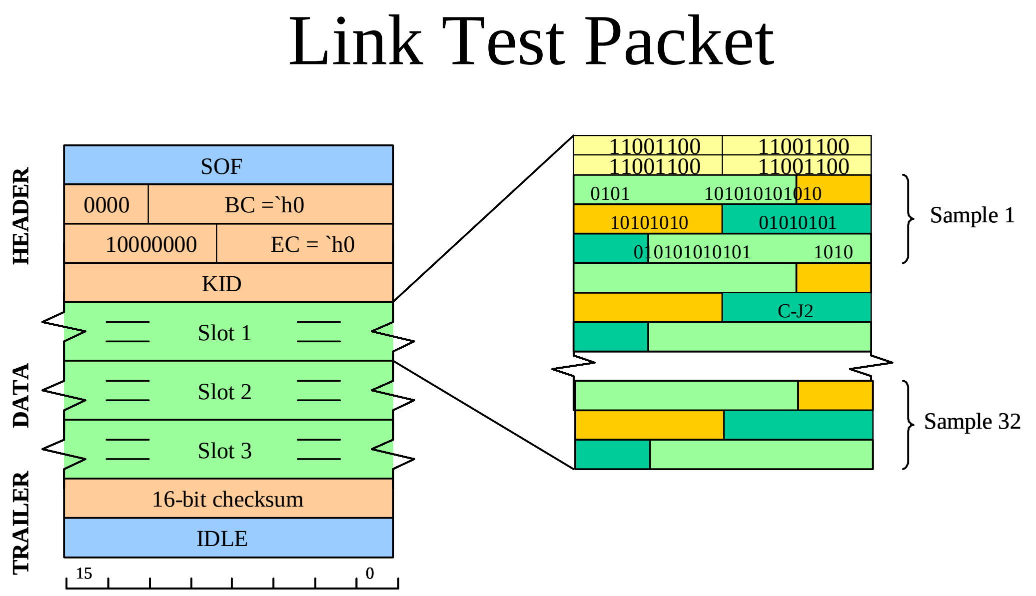


Figure 2-22 Format of the Link Test Packet.

2.14 Calibration Circuit

2.14.1 The DLL circuit

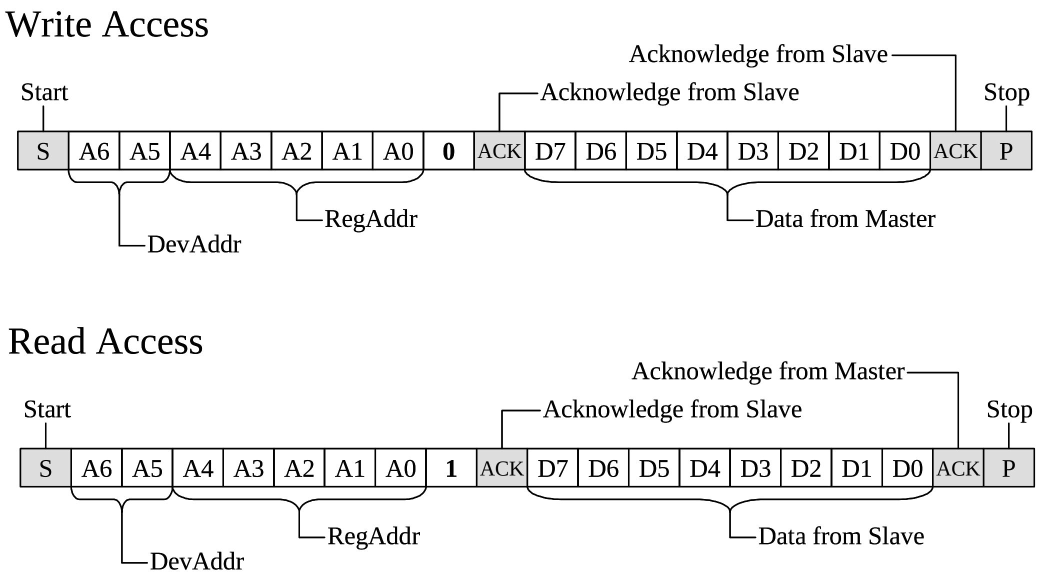
The DLL block gets reset upon ResetB signal going low ~~and upon a ReSync command.~~

2.14.2 Generating and Reading out calibration events*.*

2.15 I^{2}C Interface

The Kchip has a slow control interface following the I^{2}C-bus PHILIPS standard. This is specified completely in the PHILIPS datasheet and is not described here. The Kchip I2C interface allows for **7-bit addressing single byte transfers**. The format of the Read and Write access cycles that can be targeted to the Kchip are shown in Figure 2-23. Table 2-4 shows the bit assignment for the address field of the I^{2}C bus transactions. Incremental Read or Write accesses without retransmitting the address field are not possible.

The Kchip I^{2}C slave interface is implemented using synchronous logic clocked at 40MHz. A synchronizer circuit is employed in the SCA SDA signals to alleviate any metastability problems.



Slave is the Kchip
Master is the CCU chip

Figure 2-23 Write and Read access cycles through the Kchip I^{2}C bus.

Table 2-4 Bit assignment for the address field of the I^{2}C bus transactions.

Bits <6:0>	**Name**	**Comments**
4:0	RegAddr<4:0>	Selects the location in the Internal Register address space.
6:5	DevAddr<1:0>	Sets the Device Address on the I^2C bus.

3 SEU Tolerant Techniques

3.1 General Architecture

The Kchip will have to operate in a radiation environment where the flux of energetic particles that can cause Single Event Upsets (SEUs) is high.

The design guidelines for the Kchip are:

To protect the **Control Logic** circuitry using Triple Module Redundancy. SEUs in the Kchip control logic would cause loss of synch that can only be recovered by resetting the Kchip. To avoid the need of frequent resets it is necessary that we protect as much as possible the operation of all the Kchip State Machines and the configuration registers.

To leave the **Data Path** unprotected. The triplication of the Kchip Data Path is not foreseen since the errors created by the SEU will affect the integrity of a small amount of information being processed at the time of the SEU incidence and will not lead to a loss of sync of the Kchip operation.

3.2 State Machines

The implemented Triple Module Redundancy scheme for the Kchip state machines is shown in Figure 3-1.

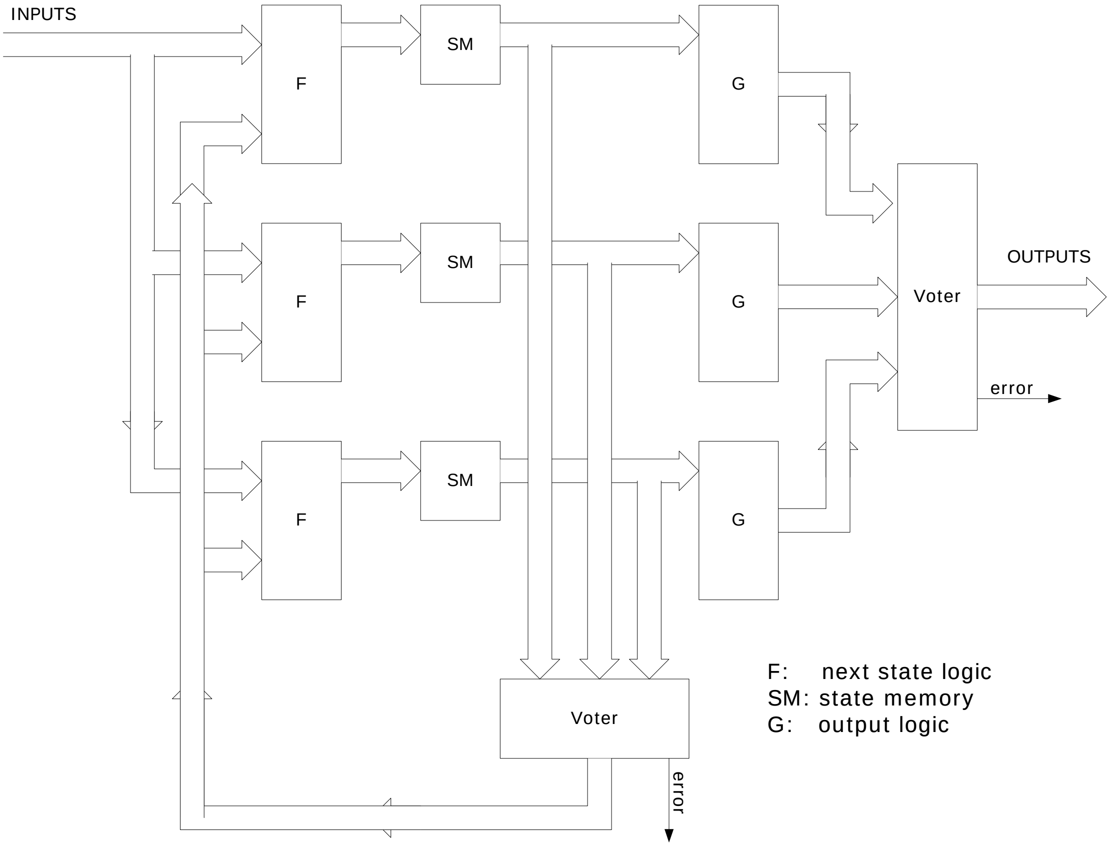


Figure 3-1 Triple Module Redundancy for the Kchip State Machines.

3.3 Configuration Registers

The implemented Triple Module Redundancy scheme for the Kchip configuration registers is shown in Figure 3-2.

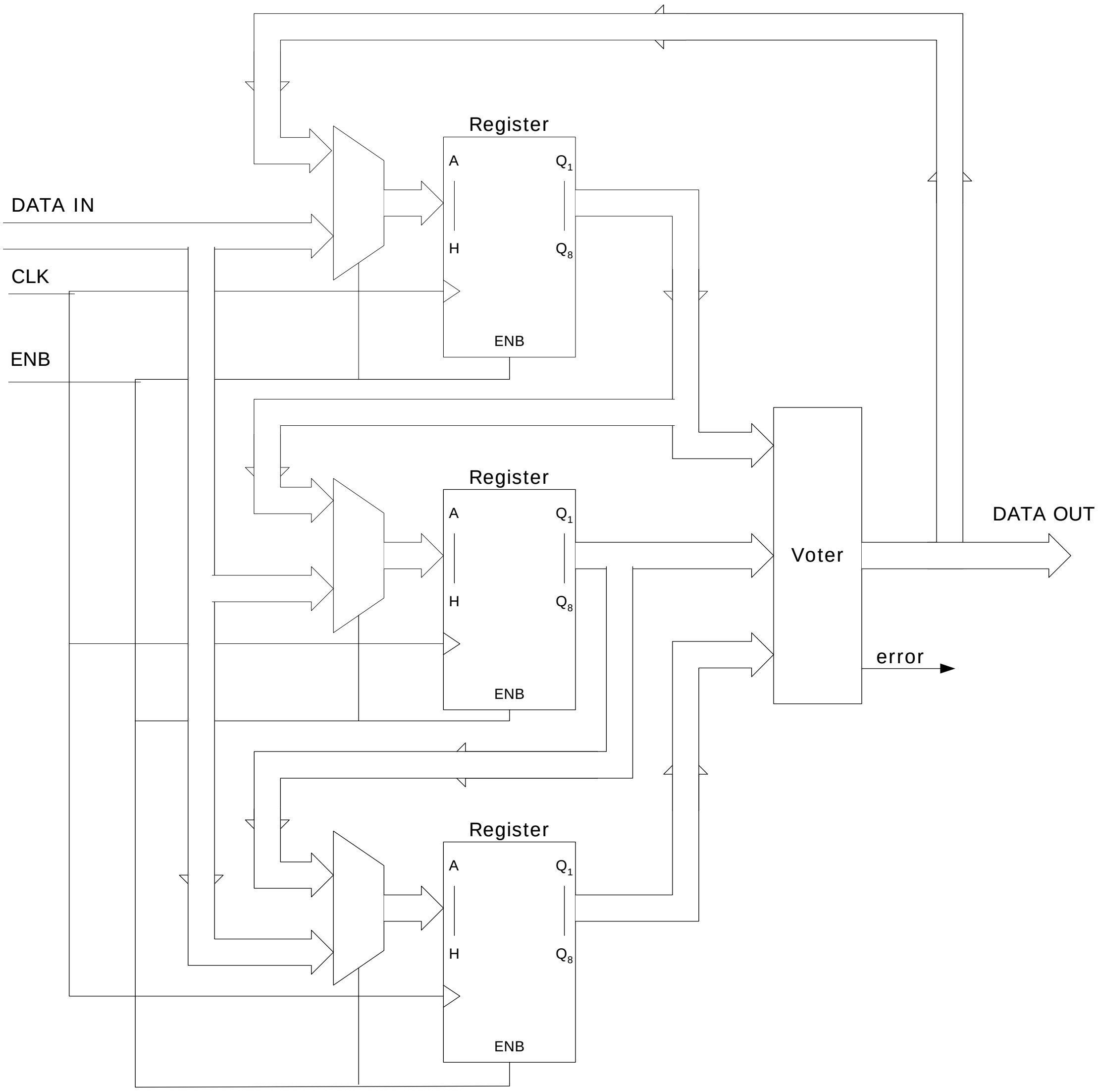


Figure 3-2 Triple Module Redundancy for the Kchip Configuration Registers.

3.4 The SEU_COUNTER

The error signals from all the Triplicated Modules are ORed together to provide a global error signal. This signal is fed to a synchronous counter that is clocked from the 40MHz system clock. The counter value will indicate the number of Single

Event Upsets encountered on all the Triplicated Modules on the Kchip since the last hardware reset. The SEU_COUNTER is accessible through the I^{2}C bus.

Optionally the SEU_COUNTER can be used as a tool for the production testing of the Kchip. The chips having a production fault can be detected by checking that the contents of the SEU_COUNTER are constantly changing.

4 Internal Registers

The Table 4-1 specifies the registers (all 8-bit wide) accessible via the I2C interface in the K-chip (double registers are tagged with a _H-L name).

Table 4-1 K-chip Internal Registers

Name	I^{2}C Address	Function	Type
CONFIG	0	This register contains various configuration and mode fields as specified below. Default value: 8`b00001111	R/W
ECONFIG	1	This extra-configuration register contains various configuration and mode fields as specified below. Default value: 2`b00	R/W
KID_H-L	3-2	K-chip ID register. Default value: 16`b00000000000000xx	R/W
MASK_T1CMD	4	This register contains a 4 bit mask pattern for the Trigger Commands. Default value: 4’b0000	R/W
LAST_T1CMD	5	A read operation from this register returns the last issued Trigger Command.	RO
LATENCY	6	This register sets the Trigger Latency value, in clock cycles, for generating a readout cycle after a Calibration command has been issued. Default value: ‘d128	R/W
EVCNT	7	A read operation from this register gives the 8 bit current content of the Event Counter in the K-chip	RO
BNCHCNT_H-L	9-8	A read operation from this register returns the 12 bit bunch counter value used to tag the last event.	RO
RESERVED	A		
GINT_BUSY	B	This register contains the timeout period for transmitting a synchronization pattern by the GOL interface. Default value: `d0	R/W
GINT_IDLE	C	This register contains the length of the synchronization pattern in the GOL interface. Default value: `d0	R/W
FIFOMAP	D	This register contains a pointer to one of the FIFOs in the chip, it is used to direct read/write operations to the corresponding FIFO	R/W

FIFODATA_H-L	E-F	When in Link Test mode, writing to this register causes data to be written into the FIFO pointed to by the FIFOMAP register; a read operation instead reads data from the corresponding FIFO. When in normal mode, read/write operations to this register are ignored.	R/W
STATUS_0	10	This register contains a number of status bits as specified below. It is reset upon a ReSync command.	RO
STATUS_1	11	This register contains a number of status bits as specified below. It is reset upon a ReSync command.	RO
SEU_COUNTER	12	This register contains the total number of single Events Upsets encountered on the chip since the last hardware RESET.	RO
CalPulse_DELAY	13	This register contains the delay period for generating a PACE calibration pulse. Default value: `d1	R/W
CalPulse_WIDTH	14	This register contains the width period of the PACE calibration pulse. Default value: `d1	R/W
RESERVED	15		
RESERVED	16		
RESERVED	17		
RESERVED	18		
RESERVED	19		
RESERVED	1A		
RESERVED	1B		
RESERVED	1C		
RESERVED	1D		
RESERVED	1E		
RESERVED	1F		

4.1.1 The *CONFIG* Register

Table 4-2 *Config* register bit assignment.

Name	Position	Function
KchipMode	7	Determines if the K-chip is in normal or test mode. A 0 in this bit corresponds to NORMAL mode A 1 in this bit corresponds to TEST mode. The bit is reset to 0 after an external reset to the Kchip.

TriggerInhibitMode	6	Determines the mode of the Trigger Inhibit logic. A 0 in this bit sets the logic into the INHIBIT mode. A 1 in this bit sets the logic into the PASSIVE mode. The bit is reset to 0 after an external reset to the Kchip.
TX_EnableB	5	Controls the data transmission through the GOL link. A 0 in this bit ENABLES the data transmission. A 1 in this bit DISABLES the data transmission. The bit is reset to 0 after an external reset to the Kchip.
Link_Test	4	Link Test Mode. A 0 in this bit enables the normal data transmission. A 1 in this bit sets the chip in the link test mode. The bit is reset to 0 after an external reset to the Kchip.
ChannelEnable_D	3	
ChannelEnable_C	2	
ChannelEnable_B	1	
ChannelEnable_A	0	

4.1.2 The *ECONFIG* Register

Table 4-3 *ECONFIG* register bit assignment.

Name	Position	Function
	7	
	6	
	5	
	4	
	3	
	2	
Mask_CalEvent	1	Controls the automatic generation of calibration events after the “latency” period following the issue of a Calibration Trigger Command. A 0 in this bit ENABLES the generation of Cal_Events. A 1 in this bit DISABLES the generation of Cal_Events.
DLL_Off	0	Enables or Disables the DLL logic. A 0 in this bit ENABLES the DLL logic. A 1 in this bit DISABLES the DLL logic.

4.1.3 The *STATUS_0* Register

Table 4-4 *STATUS_0* register pin assignment.

Name	Position	Function
KOS	7	Kchip Out of Sync. This bit sticks until the K-chip receives a ReSync command or a RESET is applied.
GOL_READY	6	This bit monitors the Gigabit Optical Link READY signal.
	5	
	4	
	3	
	2	
	1	
	0	

4.1.4 The *STATUS_1* Register

Table 4-5 *STATUS_1* register pin assignment.

Name	Position	Function
PROS_D	7	PACE Readout Out-of-Sequence This bit is set when the PACE corresponding to input channel D has delivered a DataValid signal that does not follow a correct readout cycle sequence. This bit sticks until the K-chip receives a ReSync command or a RESET is applied.
PROS_C	6	PACE Readout Out-of-Sequence This bit is set when the PACE corresponding to input channel C has delivered a DataValid signal that does not follow a correct readout cycle sequence. This bit sticks until the K-chip receives a ReSync command or a RESET is applied.
PROS_B	5	PACE Readout Out-of-Sequence This bit is set when the PACE corresponding to input channel B has delivered a DataValid signal that does not follow a correct readout cycle sequence. This bit sticks until the K-chip receives a ReSync command or a RESET is applied.

PROS_A	4	PACE Readout Out-of-Sequence This bit is set when the PACE corresponding to input channel A has delivered a DataValid signal that does not follow a correct readout cycle sequence. This bit sticks until the K-chip receives a ReSync command or a RESET is applied.
PFOS_D	3	PACE FIFO Out-of-Sequence This bit is set when the PACE corresponding to input channel D has delivered an Almost_Full FIFO signal that does not follow a correct sequence. This bit sticks until the K-chip receives a ReSync command or a RESET is applied.
PFOS_C	2	PACE FIFO Out-of-Sequence This bit is set when the PACE corresponding to input channel C has delivered an Almost_Full FIFO signal that does not follow a correct sequence. This bit sticks until the K-chip receives a ReSync command or a RESET is applied.
PFOS_B	1	PACE FIFO Out-of-Sequence This bit is set when the PACE corresponding to input channel B has delivered an Almost_Full FIFO signal that does not follow a correct sequence. This bit sticks until the K-chip receives a ReSync command or a RESET is applied.
PFOS_A	0	PACE FIFO Out-of-Sequence This bit is set when the PACE corresponding to input channel A has delivered an Almost_Full FIFO signal that does not follow a correct sequence. This bit sticks until the K-chip receives a ReSync command or a RESET is applied.

4.1.5 The *FIFOMAP* Register

Table 4-6 *FIFOMAP* bit assignment.

Bits <5:0>	**Map**
5	R/W operations to Trigger FIFO.
4	R/W operations to Column Address FIFO.
3	R/W operations to Data FIFO channel D
2	R/W operations to Data FIFO channel C
1	R/W operations to Data FIFO channel B
0	R/W operations to Data FIFO channel A

4.1.6 The *FIFODATA* Register

This 16 bit register is used to write 12 bits of data into the FIFO pointer to by the FIFOMAP register. The write operation actually occurs when the FIFODATA_L register is written. A read operation reads the last in the FIFO and decrements by one the number of words in it.

4.1.7 The *EVNCNT* Register

This Read-Only register return the value of the Event Counter associated with the last occurrence of a Level 1 trigger

4.1.8 The *BNCHCNT* Register

This Read-Only registers returns the value of the Bunch Counter associated with the last occurrence of a Level 1 trigger.

4.1.9 The *MASK_T1CMD* Register

This Read-Write register sets a mask pattern in the Trigger Decoder logic which can be used to individually enable or disable the decoding of trigger commands.

Table 4-7 *MASK_T1CMD* bit assignment.

Bits <3:0>	Map
3	A 1 in this bit masks the **BC0** command.
2	A 1 in this bit masks the **CalPulse** command.
1	A 1 in this bit masks the **ReSync** command.
0	A 1 in this bit masks the **LV1** command.

4.1.10 The *LAST_T1CMD* Register

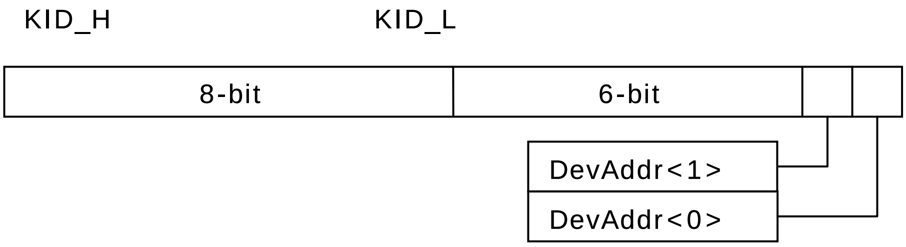
This Read-Only register returns the type of the last Trigger Command that was received and decoded by the Trigger Decoder logic.

Table 4-8 *LAST_T1CMD* bit assignment.

Bits <3:0>	Map
3	A 1 in this bit indicates a **BC0** command.
2	A 1 in this bit indicates a **CalPulse** command.
1	A 1 in this bit indicates a **ReSync** command.
0	A 1 in this bit indicates an **LV1** command.

4.1.11 The *KID* Register

This Read-Write 16-bit register sets the Kchip Identification number. The least significant part of the register (KID_L) is composed by the two hardwired pins that define the Kchip slave interface in the I2C bus and 6-bits that are accessible by the user. The format of the KID register is shown in the figure below.



The KID is transmitted with every event data packet.

4.1.12 The *LATENCY* Register

This Read-Write register sets the latency value in clock cycles for initiating the readout cycle of the calibration event.

4.1.13 The *GINT_BUSY* Register

This Read-Write register sets the timeout period of continuously transmitted event data without intermittent IDLE characters. The timeout period is defined in number of event data packets. For examples refer to Table 4-10.

4.1.14 The *GINT_IDLE* register

This Read-Write register sets the length of the link synchronization pattern (SYNC pattern). The length is defined in number of IDLE characters. For examples refer to Table 4-9.

Table 4-10 Possible combinations for the GINT_BUSY and GINT_IDLE registers.

GINT_BUSY	GINT_IDLE	Result
0	0	No IDLE characters are inserted.
0	1	No IDLE characters are inserted.
1	0	No IDLE characters are inserted.
1	1	One IDLE character is inserted in every DATA PACKET.
1	2	Two IDLE characters are inserted in every DATA PACKET.
2	1	One IDLE character is inserted every two DATA PACKETs.

5 Operating Conditions

5.1 Recommended Operating Conditions

	MIN	TYP	MAX	Unit
V_{DD} Supply Voltage	2.25	2.50	2.75	V
VI_H High Level Input Voltage	V_{DD}-0.25			V
V_{IL} Low Level Input Voltage			0.50	V
T_A Operating Free-Air Temperature	-10	25	75	^oC

6 Timing Characteristics

Soon to come.

7 Packaging

Soon to come.

8 Pin Assignments

Table 8-1 Pin assignments sorted by pin functionality.

PIN #	Name	Type	Description
PACE/ADC Interface Channels			
1	AB_ADC_pos<0>	LVDS input	Channel A & Channel B ADC data inputs.
2	AB_ADC_neg<0>		
3	AB_ADC_pos <1>		
4	AB_ADC_neg <1>		
5	AB_ADC_pos <2>		
6	AB_ADC_neg <2>		
7	AB_ADC_pos <3>		
8	AB_ADC_neg <3>		
9	AB_ADC_pos <4>		
10	AB_ADC_neg <4>		
11	AB_ADC_pos <5>		
12	AB_ADC_neg <5>		
13	AB_ADC_pos <6>		
14	AB_ADC_neg <6>		
15	AB_ADC_pos <7>		
16	AB_ADC_neg <7>		
17	AB_ADC_pos <8>		
18	AB_ADC_neg <8>		
19	AB_ADC_pos <9>		
20	AB_ADC_neg <9>		
21	AB_ADC_pos <10>		
22	AB_ADC_neg <10>		
23	AB_ADC_pos <11>		
24	AB_ADC_neg <11>		
25	A_ColAddr_pos	LVDS input	Channel A Serial Column Address line from PACE.
26	A_ColAddr_neg		
27	A_DataValid_pos	LVDS input	Channel A Data Valid line from PACE.
28	A_DataValid_neg		
29	A_FIFO_Full	CMOS input	Channel A FIFO Full flag from PACE.
30	A_PACE_CLK_pos	LVDS output	Channel A 40MHz PACE clock line.
31	A_PACE_CLK_neg		
32	A_LV1_pos	LVDS output	Channel A PACE First Level Trigger command line.
33	A_LV1_neg		
34	A_ReSync_pos	LVDS output	Channel A PACE Resynchronization command line.
35	A_ReSync_neg		
36	A_CalPulse_pos	LVDS output	Channel A PACE Calibration Pulse Injection command line.
37	A_CalPulse_neg		
38	B_ColAddr_pos	LVDS input	Channel B Serial Column Address line from PACE.
39	B_ColAddr_neg		
40	B_DataValid_pos	LVDS input	Channel B Data Valid line from PACE.
41	B_DataValid_neg		

42	B_FIFO_Full	CMOS input	Channel B FIFO Full flag from PACE.
43	B_PCLK_pos	LVDS output	Channel B 40MHz PACE clock line.
44	B_PCLK_neg		
45	B_LV1_pos	LVDS output	Channel B PACE First Level Trigger command line.
46	B_LV1_neg		
47	B_ReSync_pos	LVDS output	Channel B PACE Resynchronization command line.
48	B_ReSync_neg		
49	B_CalPulse_pos	LVDS output	Channel B PACE Calibration Pulse Injection command line.
50	B_CalPulse_neg		
51	CD_ADC_pos <0>	LVDS input	Channel C & Channel D ADC data inputs.
52	CD_ADC_neg <0>		
53	CD_ADC_pos <1>		
54	CD_ADC_neg <1>		
55	CD_ADC_pos <2>		
56	CD_ADC_neg <2>		
57	CD_ADC_pos <3>		
58	CD_ADC_neg <3>		
59	CD_ADC_pos <4>		
60	CD_ADC_neg <4>		
61	CD_ADC_pos <5>		
62	CD_ADC_neg <5>		
63	CD_ADC_pos <6>		
64	CD_ADC_neg <6>		
65	CD_ADC_pos <7>		
66	CD_ADC_neg <7>		
67	CD_ADC_pos <8>		
68	CD_ADC_neg <8>		
69	CD_ADC_pos <9>		
70	CD_ADC_neg <9>		
71	CD_ADC_pos <10>		
72	CD_ADC_neg <10>		
73	CD_ADC_pos <11>		
74	CD_ADC_neg <11>		
75	C_ColAddr_pos	LVDS input	Channel C Serial Column Address line from PACE.
76	C_ColAddr_neg		
77	C_DataValid_pos	LVDS input	Channel C Data Valid line from PACE.
78	C_DataValid_neg		
79	C_FIFO_Full	CMOS input	Channel C FIFO Full flag from PACE.
80	C_PACE_CLK_pos	LVDS output	Channel C 40MHz PACE clock line.
81	C_PACE_CLK_neg		
82	C_LV1_pos	LVDS output	Channel C PACE First Level Trigger command line.
83	C_LV1_neg		
84	C_ReSync_pos	LVDS output	Channel C PACE Resynchronization command line.
85	C_ReSync_neg		
86	C_CalPulse_pos	LVDS output	Channel C PACE Calibration Pulse Injection command line.
87	C_CalPulse_neg		
88	D_ColAddr_pos	LVDS input	Channel D Serial Column Address line from PACE.
89	D_ColAddr_neg		
90	D_DataValid_pos	LVDS input	Channel D Data Valid line from PACE.
91	D_DataValid_neg		

92	D_FIFO_Full	CMOS input	Channel D FIFO Full flag from PACE.
93	D_PACE_CLK_pos	LVDS output	Channel D 40MHz PACE clock line.
94	D_PACE_CLK_neg		
95	D_LV1	LVDS output	Channel D PACE First Level Trigger command line.
96	D_LV1b		
97	D_ReSync_pos	LVDS output	Channel D PACE Resynchronization command line.
98	D_ReSync_neg		
99	D_CalPulse_pos	LVDS output	Channel D PACE Calibration Pulse Injection command line.
100	D_CalPulse_neg		
101	ADC_CLK_pos	LVDS output	40MHz ADC clock line.
102	ADC_CLK_neg		
Gigabit Optical Link interface			
103	TX_data<15>	CMOS output	Gigabit Optical Link data output.
104	TX_data<14>		
105	TX_data<13>		
106	TX_data<12>		
107	TX_data<11>		
108	TX_data<10>		
109	TX_data<9>		
110	TX_data<8>		
111	TX_data<7>		
112	TX_data<6>		
113	TX_data<5>		
114	TX_data<4>		
115	TX_data<3>		
116	TX_data<2>		
117	TX_data<1>		
118	TX_data<0>		
119	CAV/TX_ER	CMOS output	Control Available / Transmit Error
120	DAV/TX_EN	CMOS output	Data Available / Transmit Enable
121	READY	CMOS input	Data Link Ready
I2C interface			
122	I2C_SCL	CMOS input	I2C interface clock line.
123	I2C_SDA	CMOS bidirectional	I2C interface data line.
124	I2C_addr<1>	CMOS input	I2C device address.
125	I2C_addr<0>		
Fast Timing input signals			
126	CLK_IN_pos	LVDS input	40Mhz LHC clock from PLL chip.
127	CLK_IN_neg		
128	T1_pos	LVDS input	LV1 trigger command input from PLL chip.
129	T1_neg		
130	RESETb	CMOS input Schmitt	Hardware reset line from optical hybrid.
131	test_se	CMOS input	Scan path enable pin. (1=enable scan)
Power and Ground			
132	VDD_CORE	2.5V	Core power.
133	VDD_CORE		
134	VDD_CORE		
135	VDD_CORE		

136	VDD_PERI		I/O pad power.
137	VDD_PERI		
138	VDD_PERI		
139	VDD_PERI		
140	GND_CORE	GND	Core power.
141	GND_CORE		
142	GND_CORE		
143	GND_CORE		
144	GND_PERI		I/O pad power.
145	GND_PERI		
146	GND_PERI		
147	GND_PERI		
148	GND_PERI		

9 Appendixes

9.1 Kchip Interfacing

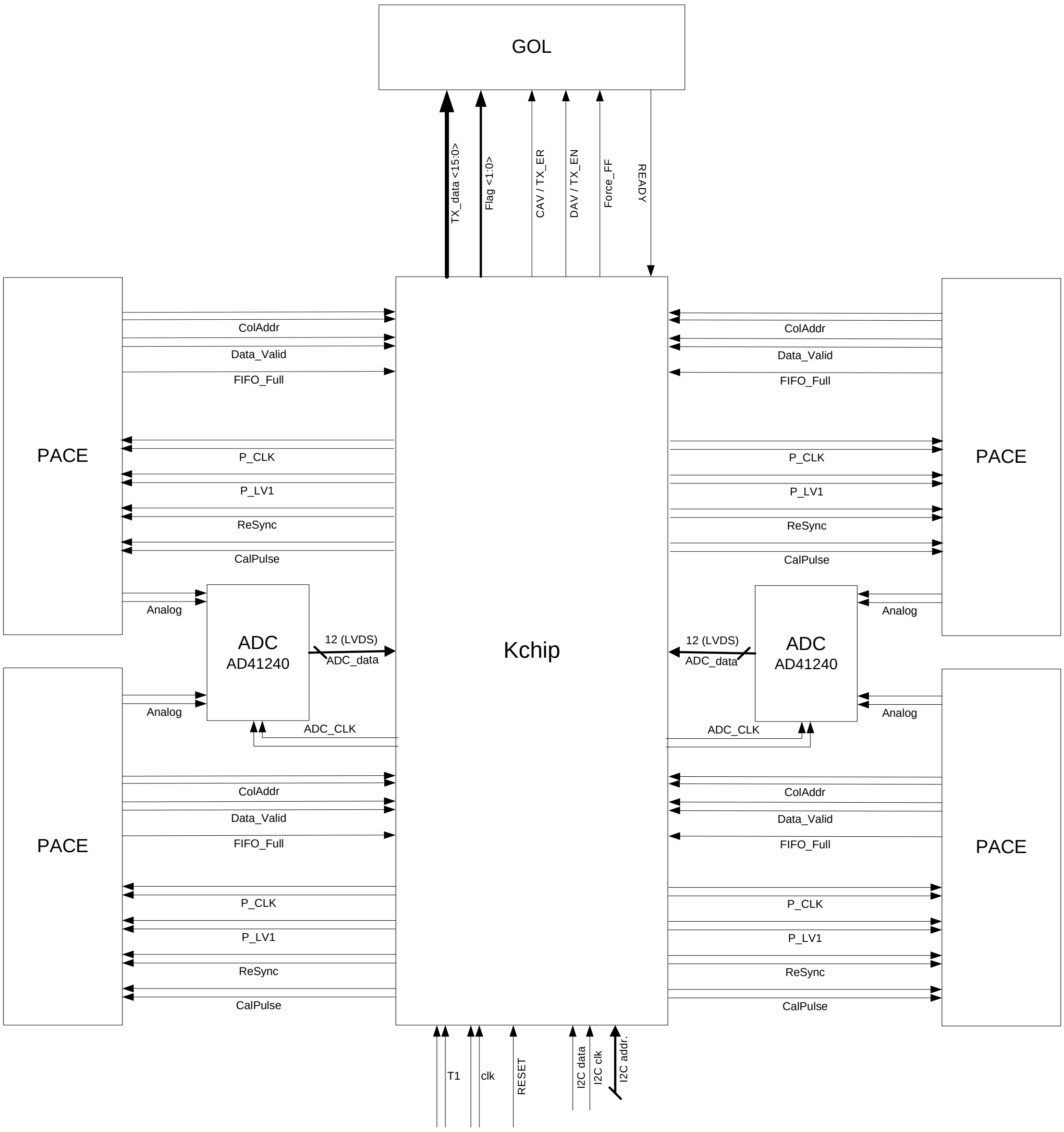


Figure 9-1 The Kchip interfaces.

9.2 PACE3 Overflow Probability simulation results.

08/08/02

The following simulation results are based on the recent digital specifications of the PACE3 chip (08/08/02). The simulation model emulates the operation of the column address FIFO. On every LV1 trigger there are 3 column addresses pushed in the FIFO. A readout logic monitors the *"Emptyb"* FIFO signal. When this signal is asserted it pops out one column address and begins a readout cycle that lasts for 19+9+(2X32) = 92 clock cycles (2.3µsec).

The FIFO has 32 locations overall. An almost full (AFull) signal is generated by the FIFO control circuit in order to inhibit these triggers that would cause an overflow to occur. This almost full signal can be generated either when 30 positions are occupied in the FIFO or when 28 positions are occupied in the FIFO.

Simulation results for ~2E06 events follows.

Trigger Rate	FIFO AFull threshold	Overflow Probability
100KHz	30	1.9E-04
100KHz	28	3.1E-04
75KHz	30	1.7E-06
75KHz	28	5.7E-06

The emulation model assumes that there are no inefficiencies introduced by the "skip controller logic" in the operation of the FIFO.

PACE3 overflows will not cause the readout chain to go out of sync. There is no dead time introduced after a lost event. The PACE3 overflow probability, as estimated by the simulation model, should be considered as a component of the detector electronics inefficiency.

9.3 PACE3-AM DLL

Quentin Morrissey, RAL 21-Oct-2002

This document gives a very brief description of the structure and operation of calibration circuit taken from the APV25. The entire calibration circuit has been supplied, though only the delay locked loop is required for the PACE3-AM chip.

IO

Inputs:

SELDEL<7:0> - Selects delay setting, activate one bit only, active low

CDRV<7:0> - Enable signals for output lines, not required

CAL_REQ - Triggers generation of a calibrate pulse, active high

CAL_OFF - Deactivates calibrate circuit, active high

ICAL - Bias current for output stages, not required

CLK - 40MHz system clock

RES - Reset signal, active high

Outputs:

CAL_OUT<7:0> - Output signal lines

DEL1 - Testpoint at end of reference delay line

DEL2 - Testpoint at end of signal delay line

bias_level - Testpoint for charge pump output voltage

Structure

The APV25 calibration circuit breaks down into 3 main sections, these being the calibrate logic, the delay chains, and the output stages. Since the PACE system has different calibration requirements to the APV25, the output stages are redundant in this case. The remaining to sections are of interest.

Delay chain

This block forms the heart of the DLL, and consists of a charge pump and two identical delay lines. The delay lines are each constructed of 16 ‘starved’ (current controlled) buffers. The charge pump is used to control the voltages on a pair of capacitors, generating bias levels to control the buffers, and hence the delays through those buffers. One delay line is used by the calibrate logic (explained below) to measure and control the timing of signals passed through the delay lines. The other line is used for generating the delay in the calibrate signal, and it is possible to tap the delay line at any of the central 8 buffers. The system is designed to allow 2 clock cycles, 50ns, for a pulse pass from one end of the delay chain to the other. The 16 elements therefore give a timing resolution of 3.25ns, with the tap points allowing an offset of –9.75ns to +13ns from the rising edge of the calibrate request signal.

Calibrate logic

The calibrate logic has the job of timing the delay lines in the delay chain block and of controlling the charge pump. This is done using two sub-blocks, the calibrate sequencer and the phase detector. The sequencer is a small state machine that generates pulses in a regular sequencer. Every 8 clock cycles a single 50ns pulse is applied to the input of the reference delay line. Two clock cycles later another 50ns pulse is generated and passed to the phase detector. The phase detector compares the second pulse from the sequencer with the returning pulse from the reference line. Depending on which pulse arrives first, a charge or discharge signal will be sent to the charge pump which will alter the bias levels of the buffers accordingly. This process of timing the reference line will run continuously while CAL_OFF is low, and does not interfere with the generation of calibrate signals.

The calibrate logic also contains a small circuit for generating calibrate signals upon request. This passes the CAL_REQ signal into the signal delay line, and waits for the signal to return from one of the tapped points on the line. The rising edges of CAL_REQ and the delayed return signal are used to clock a toggle flip-flop and a registered output. This output is used by the output stages to switch generate a voltage step on the output lines. It should be noted that this system results in output signals which alternate polarity with every calibrate request. This was required since the APV25 was designed to readout both N-in-P and P-in-N detectors, and may not be applicable to the PACE3 system.

Operation

Operation of the circuit is quite simple, once the delay has been set. This is done by setting one bit of SELDEL<7:0> low. All other bits should remain high, as if more than one bit is low, only the least significant bit will take effect. SELDEL selects a delay relative to the rising edge of CAL_REQ. This delay ranges from –9.75ns for SELDEL<0>, up to +13.0ns for SELDEL<7>, increasing in steps of 3.25ns. For the full circuit, any number of output channels could be enabled using CDRV<7:0>, however this is not relevant to the PACE system.

Once the delay has been chosen, operation is controlled using CAL_OFF and CAL_REQ. CAL_OFF is used to shut down all operations within the calibration circuit including the delay reference line, and was used to reduce noise during data taking on the APV25. The rising edge of CAL_REQ is used to trigger a calibration event, and can be one or more clocks cycles in length. The CAL_SWITCH and CAL_SWITCHb outputs of the calibrate logic will change after a delay of 25ns + <delay> as selected with SELDEL, relative to the rising edge of CAL_REQ. As mentioned, the direction of the change to CAL_SWITCH/b will alternate with every calibrate request.

Layout

Layout of the calibration circuit is designed to give the best matching between the reference and signal delay lines. These two lines run in parallel across the lower section of the layout, with the space above containing the calibrate logic and the charge pump with it’s bias capacitors. The resistor based output stages can be found arrayed along the very bottom of the layout, and should not be difficult to remove.

Powering

It should be noted that the APV25 was a dual supply chip, with VDD = 1.25V and VSS = -1.25V. These supplies can easily be converted to 2.5V and ground, however the mid-rail ground supply was used in the APV25 to bias the gate of a device in the charge pump voltage generator (cal_v2c block). Therefore, an alternate method of biasing this device will need to be found.

9.4 Kchip Design Review minutes Date: 4/4/2002

Present: K. Kloukinas, B. Lofstedt, S. Reynaud, P. Aspell, D. Barney.

Time: between 9:30 AM to 5:30 PM

Introduction

The primary purpose of this design review was to review the functionality of the Kchip in the front-end readout chain and to point out any necessary changes. Also we wanted to ensure that the two front end chips, the PACE and the Kchip, can work together and that future changes on the PACE (PACE2b in DMILL and PACE3 in IBM) will not affect the Kchip design. A second Kchip design review will take place some time later. This second review will focus on the technical details of the circuit implementation on silicon and should be the final review before submitting the chip. For this review some people from the MIC group should be invited to participate.

The review was mainly focused on the following issues:

PACE – Kchip interface

Kostas presented the operation of the PACE-Kchip interface in the current Kchip design. The present version of the Kchip deduces the PACE 20MHz internal signal “MUXclk” from the 40MHz system clock and synchronizes its phase when a ReSync command is applied. This signal is necessary for the Kchip in order to drive properly the ADC clock line and to latch the ADC data output. Another crucial interface signal is the DataValid signal which is emitted by the PACE chip and changes state when a new event starts being readout. This interface configuration is, to some extent, sensitive to phase delays between the system 40MHz clock and the two PACE control signals MUXclk and DataValid, which can change with the radiation exposure of the PACE chip. The synchronization problem gets more complicated by the fact that there are four PACE chips connected to a single Kchip which might have differences in the propagation delays on these signals and which might degrade differently with the radiation dose.

Post irradiation measurements (after 10Mrad and ~1month of annealing period) showed a difference in the clk-to-DataValid propagation delay before and after irradiation of ~6ns maximum. Moreover, the spread of chip-to-chip clk-to-DataValid propagation delays was found to be large (2.5ns min, 6.5ns max), making their interfacing with the Kchip more prone to loss of synchronization. Putting this value into the Kchip + 4PACEs simulation test bench showed that the interface synchronicity is maintained but the ADC sampling point shifts to a region where the PACE analog output might be in a transitory state. It was clear that in order to guarantee the performance of the system it is needed to perform more rigorous irradiation measurements on the propagation delay changes of the control signals and the timing of the analog output of the PACE chip.

Another approach for interfacing the PACEs to a Kchip is to have a MUXclk output from each PACE chip delivered to the Kchip. The Kchip will have four independently running interfaces for the 4 PACE – ADC channels that are clocked by four separate MUXclk signals. The phase difference between the 40MHz clock and these signals is not bound to be the same in this implementation. The problem is now that the Kchip has to synchronize the four PACE – ADC interfaces to its internal clock.

Serge presented the implementation of the above-mentioned interfacing scheme and proposed a possible data synchronizer circuit. For more details on that have a look at the presentation made by Serge.

It was decided that we should adopt this type of interface and therefore we should take the following actions:

1. The PACE 2b and the PACE 3 chips should have an LVDS output pad for the MUXclk signal, and

2. the Kchip should be redesigned in ordered to accommodate this interfacing scheme. The Kchip redesigning is quite extensive. The changes are not confined to the operations of the 4 data FIFOs that are attached to the ADCs but also affect the way the Kchip stores and synchronizes data and column addresses from the 4 PACEs.

A possible disadvantage for this solution is that it could result in some more noise in the PACE chip. To avoid redesigning again the PACE we decided that the MUXclk output should be programmable so that it can be enabled or disabled. Of course, we are aware that if we fall back to a “ non MUXclk “ interface the Kchip might need some modifications.

Data Path design and data flow

Kostas gave an overview of the general operation of the Kchip and presented the data flow inside the chip.

It was agreed that the dataflow and the functionality implemented in the Kchip meets the readout system needs.

Buffer Overflow handling

Kostas presented the behavior of the Kchip when one of the following conditions occurs:

1. the PACE chip overflows (almost full condition),

2. the Kchip data buffers overflow and

3. the Trigger buffer overflows.

For the case number 1 the Kchip has a programmable trigger inhibit logic that filters out all LV1 commands when the PACE chips report an almost full condition. The filtered out event is flagged in the data buffers and a null event is generated in order to maintain readout synchronicity. The same policy applies for case number 2, i.e. the data buffers are protected from overflowing and losing their data and the skipped events are flagged so that a null event is inserted in their place. For case number 3 a programmable trigger inhibit logic filters out all LV1 commands that will overflow the trigger buffer. The blocked LV1 commands are properly flagged while the EC and the BC are unconditionally incremented.

GOL interface

The GOL chip speed/protocol final choice was still an open issue at the time of this review. The current version of the Kchip is not fixed to a specific mode of operation of the GOL chip. Instead a generic interface with 16bit parallel bus and a strobe (clock) line is implemented to allow interfacing with a FIFO.

Bo proposed that we should wait a few weeks before making a final decision. It is likely that by that time a decision in the ECAL community will be taken.

I2C interface

The I2C port operation was presented focusing on the Kchip addressing mode. The Kchip I2C port is a synchronous design with a synchronizer circuit on the SCL & SDA interface lines. It accepts 7-bit addressing transactions. The 4 LSB are used for internal addressing giving an address space for 16 control/status registers. Two more bits are used for Kchip device addressing and the MSB of the 7-bit address field is left unused.

The general addressing scheme for the front-end chips on the motherboard/hybrid level was also discussed during the meeting. The Kchip addressing is currenty designed to comply with the motherboard addressing scheme proposed in the "Preshower Front-End Readout & Control" manual.

Bo proposed that Serge and Kostas should review this addressing scheme in order to make sure that in the system we can individually address (identify) any motherboard in the system with a unique address while we are not restricted of having multiple motherboards with same address on the control rings. The addressing scheme on the I2C level should also be reviewed.

SEU protection

The protection of the Kchip logic against SEU induced by the high fluence of ionizing particles during operation was not addressed in this design review. It was proposed that the Kchip control logic should be protected as much as possible against errors while the data path can tolerate a certain amount of SEUs. The Kchip designer will come up with a proposal following the two above mention design guidelines on a forthcoming Preshower meeting for discussion.

Packaging

Kostas proposed to have a ceramic PGA package for testing the Kchip on the Kmotherboard or on the digital tester and an fpBGA for mounting the final Kchip on the final motherboard. The PGA package is consuming a lot of real estate on the board but allows cheap in-house bonding. After implementing the proposed changes on the chip, which affect the chip pin count, a final choice of the test package will be made.

Other Proposed changes on the current Kchip design

1. The status of the DataValid signals from the 4 PACE chips should be flagged in the control field of the Data Packet.

2. The ReSync command should unconditionally clear the events that are stored in the Data Buffer as well as the pending events in the Trigger Buffer on the Kchip. The current version of the chip has this as a programmable feature.

3. The Kchip should increment the Event Counter when a calibration event is generated. A calibration event is generated by the Kchip when it receives a CalPulse command. After a period equal to the trigger latency an LV1 pulse is issued from the Kchip to the PACEs in order to read out the calibration pulse. This LV1 should increment the Kchip Event Counter as any other normal event does. A special flag in the control field of the data packet will indicate that this is a Calibration Event.

4. The Kchip should electrically buffer the RESETb signal that it receives from the CCU reset output. We should study the necessary level of fanout needed for this signal in order to be able to drive all the ASICS on the motherboard and the hybrids. The RESEtb distribution scheme should take into account the redundancy provided by the dual DOH modules.

5. The Kchip should have four independent differential CMOS (0.0-2.5V) outputs for the ADC clock lines instead of the four single ended outputs that are implemented in the current version of the chip.

10 Reference Documents