BaBar DIRC NOTE #88

Internal Group Note

Driving DIRC V2.1

Original versions: Gérard Oxoby

Updated by Dominique Breton in May 2002

For the DIRC Electronics, Calibration, and DAQ Groups

Date	Version	Description
9/2/97	Draft	First release of the file "DataFmt.fm5" to the DIRC Electronics Group
10/14/97	1.0	BaBar Note #88 New file name and addition of this page, many updates.
6/10/98	1.1	Major revisions of the data format
3/10/99	1.2	Updates and corrections
5/16/02	2.1	Update by Dominique Breton

Content:

This page left blank intentionally.

BaBar DIRC NOTE #88

Internal Group Note

Driving DIRC V2.1

Original versions: Gérard Oxoby[1]

Updated by Dominique Breton[1] in May 2002

For the DIRC Electronics and Calibration Groups

PURPOSE OF THIS DOCUMENT

This working document is prepared for the purpose of documenting the DIRC in view of developing the on-line software and with the ambitious intend to:

1. Update the DFB specifications [1], prepared by the Electronics Group at Orsay.
2. Examine the steps needed to acquire data from the DIRC.
3. Understand how to calibrate the system, and test the hardware.
4. Learn how to recover from errors.

Some details of the hardware need to be known to understand why things are the way they are. The background chapter should satisfy this requirement without giving too many unnecessary details.

This document repeats many details covered in [1]. It is presented here in a different format, with graphics, in the framework of developing the on-line software for the DIRC within BaBar. It may contain erroneous information since it is based on the understanding of the author[1]. Most of it was corrected by Dominique Breton in May 2002, but some might remain anyway …

1 Gerard Oxoby: gjoeb@slac.stanford.edu

Dominique Breton: breton@lal.in2p3.fr

BACKGROUND

BaBar Data Acquisition

The BaBar Data Acquisition is covered in multiple documents but for the purpose of this work the most important document to refer to is likely to be BaBar Note 281 [2]. It explains the protocol for data transmission between the detector and the Data Acquisition system located in the electronics house.

The Data Acquisition system described in this document includes parts of the Fast Control and Timing System (FCTS) [3 and 4], and what is sometimes referred to as a DAQ slave crate which is a crate specific to the detector subsystem, herein the DIRC.

For the purpose of this discussion we can simply consider the FCTS as a synchronization machine that distributes clocks, retransmits trigger information from the trigger system, and generates calibration commands towards the DAQ slave crate.

The DAQ slave crate has Read Out Modules (ROM), and an interface to the FCTS called the Fast Control Distribution Module (FCDM). The ROMs [5 and 6] are used to control the detector Front-End Electronics (FEE) and to receive data from the detector. A ROM consists of a VME processor, a Personality Card (PC) [7] that is the interface with the FEE, a Controller Card (CC) [8], that among other functions interfaces with the FCDM, and a PCI Mezzanine Card (PMC) [9] which interfaces the processor with the PC, and CC.

In run mode the FCTS may receive a signal from the trigger system. After some house keeping the signal is routed to the ROMs that send a Level 1 Trigger Accept command to the FEEs. The delay for house keeping in the trigger system, the FCTS and the ROM is called Trigger Latency. It is nominally 12μs ± 500ns. Upon receiving a L1 Trigger Accept command, the FEEs fetch the data available during that 1μs window, later called Trigger resolution, and store it into a buffer. Then the ROMs issue a Read Event command to obtain the content of the buffer on the FEEs. Since several L1 Accept triggers may occur successively, the FEEs are required to have a minimum buffer depth of four events in order to keep the detector dead time below 1%.

The Data Path

The DIRC comprises 12 sectors with nominally 896 PMTs each. Figure 1 is a very simplified diagram showing the data path for one DIRC sector. BaBar Note 324 [10] offers a detailed description of the DIRC electronics. The output signal from 64 PMTs are grouped together and connected to a 64 channel DIRC Front-end Boards (DFB). Therefore, 14 DFBs are required for each sector[2]. The 14 DFBs corresponding to a sector, and a DIRC Crate Controller board (DCC) are plugged into a VME mainframe that has a custom made backplane. The latter, called a Protocol Distribution Board (PDB), provides two uni-directional serial data links, to and from each DFB and the DCC, and facilitates a dedicated clock distribution. Each DFB receives commands from the DCC over one of the two serial links and sends data over to the DCC on the other one. The DCC has 16 outgoing command serial ports -one to each DFB slot- and 16 incom-

2 896 | 64=14

ing data serial ports -one from each DFB slot-, even if only 14 of them are actually used. The format of the commands and the protocol for data transmission are specified in [2].

The transport of data from the DFBs to the BaBar data acquisition system is performed via the DCC that receives the 16 serial data lines on the PDB. The DGB time-multiplexes the 16 lines onto one high-speed fiber-optic serial link (DLINK). This fiber-optic serial link is received by the Read Out Module's Personality Card which demultiplexes the high speed serial data stream to form 16 lower speed (60MHz) serial data streams, one for each DFB[3]. Inside the ROM, on the Personality Card, each one of the 16 data streams is converted to 32 bit parallel words that are then written into a dual port memory called the Intermediate Store (IS). The movement of data after this step is not covered herein.

A streamlined concept of this data path could ignore entirely the data transport mechanism and show the output of the DFBs directly connected to the input of 14 dual-port 1k by 32 bit memories representing the IS.

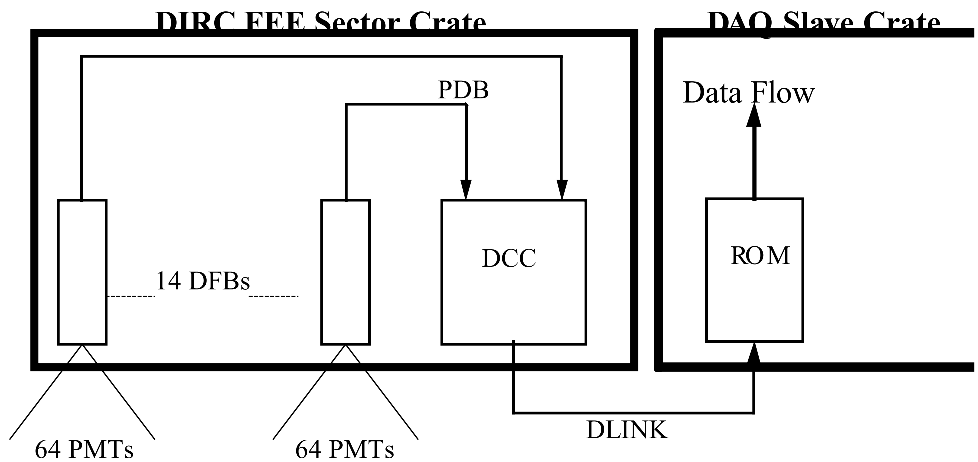

Figure 1: Simplified diagram of one DIRC Sector Data Path

The Control Path

To have access to the DIRC data some set up must be performed ahead of time. This set up is done by the software sending commands to the DFBs via the ROM and the DGB. How the software accomplishes this task may be covered later in a Data-Flow document. The bit pattern Op-codes for the commands is assembled inside the ROM, then serialized and fanned out onto 16 data streams. Each stream is ANDed with an enable signal for each DFB to provide 16 individual gated command streams[4]. In a method similar to the method used in the DCC to transmit the data to the ROM, the commands for the 16 DFB slots are time-multiplexed on the ROM and sent to the DCC over a high-speed fiber-optic link, the CLINK. The DCC that receives the CLINK demultiplexes the bit pattern and transmit the 16 commands to the DFBs via the PDB's serial command lines at 59.5MHz. A command has a five-bit Op-code field in the least significant five bits and a

3 The ROM actually receives two data streams from two DLINKs connected to two different DIRC sectors. For clarity we look only to the data from one sector since the data is handled similarly for all sectors. We should also note that the DIRC requires only 14 out of the 16 data streams the ROM is capable of handling.

five-bit information/address field in the next significant five bits. A command can be data less or followed by data.

Op-Codes

As explained in [1] and [2] the commands are divided into Global and Subsystem Specific commands. The general format of a DIRC Op-code is shown below in table 1.

.

Bit 9-5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Data	0 = Global	Hex F-0 (Run time commands)			
Address	1 = DIRC Specific	0 = read 1 = write	0 = single word 1 = block mode	Register Group (3-0)	

Table 1: DIRC Op-codes format

The Global commands or run-time commands format and their use are covered in details in [2]. They are introduced in table 2 as a reference with their respective Op-code and a short explanation of their function. Note 281 specifies the data for the L1 Trigger Accept Global command only -the trigger tag-.

.

Op-code	Command	Effect
0	NOP	No Local Operation
1	Clear Readout	Flush or Reset Event Buffers in FEEs
2	Sync	Resynchronize Timing
3	L1 Trigger Accept	Indicates that an Event Trigger occurred.
4	Read Event	ROM Request for Event Data
5	Calibration Strobe	Generate a calibration strobe
6: F	Reserved	

Table 2: BaBar Global Commands

We will now examine in more details the DIRC Specific commands. They are actually subdivided into three groups as follows:

- Group 0: Readout Setup Registers
- Group 1: Calibration Registers
- Group 2: Board Internal Tests Registers

First we will explore Group 0 commands and then look at the format of the data transferred from the FEE to the ROM. In subsequent chapters we will look at the Calibration and Tests.

4 All DFBs may receive the same command at the same time but never receive different commands at the same time. That is, one or several DFBs may be receiving a specific commands while others receive nulls (zeroes).

The DFB

The DFB design is the most influential factor driving the data format and the commands Op- codes. Antoine Ducorps's diagram "Synopsis of DIRC Front-end Board" [11] shows the components on this board. For the purpose of this exercise we only need to understand the modularity of few components on the DFB. The signals from the 64 PMTs are received by eight **eight-channel analog chips**--discriminator and shaper--each using a DAC to set its gain. The analog chips provide eight discriminator outputs and one multiplexed shaper output. This analog multiplexer address is driven by three digital inputs coming from an external priority encoder -see ANDer below. The shaper output can also be enabled or disabled by a fourth input to the analog multiplexer.

Sixteen discriminator outputs outing two analog chips are fed to an **ANDer** which, at first level, can be seen as 16 AND gates. Each gate has one input connected to a discriminator output of the analog chip and the other input connected to a channel enable bit coming from a register. The 16 outputs of an ANDer are connected to the inputs of a **sixteen-channel TDC** and to a 16-bit higher priority encoder that may control the two analog chips internal analog multiplexing of the shaper output.

The shaper output, when enabled, reproduces one of the channel's outputs. The channel selected can be either the highest order channel hit[5] -encoder mode- or a predefined channel -select mode. The shaper outputs from all eight analog chips are connected together and to the input of an eight bit **flash ADC.** Therefore only one out of the 64 channels' shaper output can be selected at one time. The 8bit output from the flash ADC is fed to the input of a RAM into which it will be written.

The TDC architecture is covered in [12]. The TDC has four internal registers that need to be loaded upon set-up.

It is now apparent that the eight analog chips and the four TDCs dictate the architecture and the modularity of the DFB. As we will see in the next chapter all set-up commands are based on that same modularity.

Other Components

At this time we are only interested in the components that affect the readout setup commands and the format of the data being read-out by the ROM.

The DGB, as a data transport agent, does not need to be covered now. However it supports the controls for the calibration LEDs and therefore will need to be covered when we reach the calibration. The ROM has been covered with the Data Acquisition and is detailed in the references aforementioned.

---

5 Say channels 2, 5 and 7 have a hit. In encoder mode, the shaper output for channel 7 will appear at the chip's shaper output when the multiplexer output is enabled (see Charge Measurement Mode on page 6).

READOUT SET-UP

This chapter deals with the procedures required to read event or calibrate data from the DIRC. As mentioned earlier the DIRC electronics need to be set-up before it can perform, in particular acquire data and transmit it to the Data Acquisition system. In order to do the basic set-up, the registers of group 0[6] must be loaded with specific values. We will begin by looking at the commands which need to be generated in the ROM to access these registers and continue with the format of the data transmitted to the ROM when reading back registers and event or performing calibration.

Set-Up Registers (Group 0)

These registers are used to set-up the DFB before run time commands can be issued. The Op-code to access these registers is shown in table 3 below.

Bit 9-5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Address	1	0 = read 1 = write	0	0	0

Table 3: Registers Group 0 Op-code Format

[1] states: the internal structure of the board (DFB) is 16-bit wide; therefore, the data field has always a length multiple of 16. This means that when writing data to the DFBs (bit 3 set in the Op-code) a data word is always 16-bit wide.

There are seven readout register types to condition the components of the DFBs. They are listed below and the data associated with them is shown graphically when deemed useful.

Gain DACs:
(8 DACs, one for each analog chip)

Addresses

7-0 for analog chips 7-0 respectively

The DAC value occupies the 12 least significant bits of the 16-bit word.
The DACs range from -5V (0000[16]) to +5 Volts (0FFF[16]).

Charge Measurement Mode:
(One for each ANDer)

Addresses

08 Channels 15-0

6 Some registers in Group 1 and 2 -calibration and internal tests registers- may need to be set up before one can trustfully start acquiring data.

16 This means the value is presented here in hexadecimal code.

0A Channels 31-16
0C Channels 47-32
0E Channels 63-48

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	-	-	Mode		Select Channel Number			

Table 4 : Charge Measurement Mode Register

The Mode field is defined as follows[7]:

0 <= **Encoder mode**: the highest order non masked channel's shaper output address, out of 16 channels, will be encoded in a given ANDer. Then the same operation will be performed at the board level (64 channels). *Pay attention to the fact that the address encoded always corresponds to one among the* ***first*** *hits having reached the encoder.*

1 <= **Select mode**: the Channel selected in the Select Channel Number field will have its shaper output digitized, *as soon as any of the non-masked channels of the* ***same ANDer*** *will be fired.*

2 or 3 <= **Disable**: the shaper output of the two analog chips associated with this ANDer are disabled[8].

During normal run time, the default setting for this register is zero. For individual channel calibration, the Select Mode has to be used, *and all the channels except the one being currently calibrated have to be masked* (see below). This is to be sure only the given channel might trigger itself. In this mode, the mask actually allows you to choose the channels **of the same ANDer** who will fire the ADC conversion, while the address loaded in the Select Channel Number register allows you to choose which signal you want to measure. If you choose any firing channel out of the given ANDer, you might have no charge measurement eventually.

ANDer Channel Enable Pattern (Mask):

(One for each ANDer)

Addresses
09 Channels 15-0
0B Channels 31-16
0D Channels 47-32
0F Channels 63-48

These four 16-bit wide registers drive the enable input to the ANDers as explained on page 6.

A zero in a bit disables the corresponding channel from generating a hit in the TDC and therefore its shaper output from being presented to the input of the flash ADC when in encoder mode. A bit set to one enables the corresponding channel. *During calibration, to be able to*

7 See DFB shaper output on page 5

8 At least three of the four addresses need then to be disabled since there is only one ADC.

correlate properly the TDC hit and the charge, you should mask all channels but the one you want to calibrate.

During normal operation with all channels enabled the default setting for this register is FFFF^{16}.

TDC Trigger Window Register:

(One for each TDC)

Addresses

10 Channels 15-0
14 Channels 31-16
18 Channels 47-32
1C Channels 63-48

The last paragraph of the Data Acquisition, on page 2, explains the trigger latency -delay- and resolution -uncertainty window. This TDC register sets the width and the position of the resolution window relative to the L1 Trigger Accept command with a granularity of four system clock periods[9]. The most significant eight bits set the beginning of the window within which data will be written into an event buffer. The lower eight bit set the end of the window. This is done by writing into these two bytes the values as shown in table 5.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
$Latency - \frac{Resolution}{2} - 1$								$Latency + \frac{Resolution}{2}$							

Table 5: TDC Trigger and Resolution Register

The Latency is expressed as an eight-bit integer and the resolution as a five-bit integer. The diagram in figure 2 tries to illustrate this operation

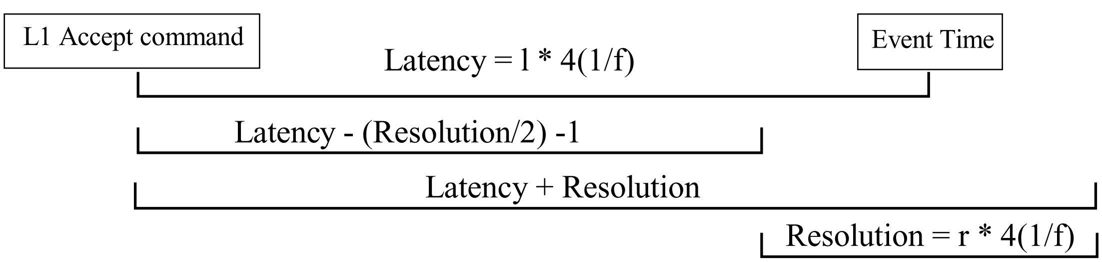

Figure 2: Illustration of the Latency and Resolution setting

TDC Selective Readout Register:

(One for each TDC)

Addresses

9 The system clock frequency is 59.5MHz, giving a clock period of 16.8ns. Therefore the granularity is 67.227ns.

11 Channels 15-0
15 Channels 31-16
19 Channels 47-32
1D Channels 63-48

This set of registers is used to generate the FIFO full status within the latency window. The functions set parameters for a loop in the TDC firmware [12].

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
int [(L - (R/2) - 2) / R] - 1						L - (R/2) – R x {int [(L – (R/2) – 2) / R] – 1}					R – 1				

Table 6: TDC Selective Readout Register

L <= Latency in multiple of four clock periods
R <= Resolution in multiple of four clock periods

TDC Internal Mask Register:

Addresses
12 Channels 15-0
16 Channels 31-16
1A Channels 47-32
1E Channels 63-48

This TDC register is 16-bit wide. Each bit enables the corresponding channel when set to a one and disable it when set to a zero. It is different than the ANDer Channel Enable Pattern (Mask) from page 7 because this register does not affect the charge measurement. It only disables TDC channels. *Thus a given hit could be seen by the encoder and not by the TDC.*

During normal operation with all channels enabled the default setting for this register is $FFFF^{16}$.

TDC Noise Suppression Register:

Addresses
13 Channels 15-0
17 Channels 31-16
1B Channels 47-32
1F Channels 63-48

The TDC is designed with a feature that dynamically prevents a noisy channel from filling up the latency FIFO. This feature is explained in chapter four of reference [13]. A four-bit number is used to generate the width of a time window during which every channel hit is monitored. If a channel exceeds 32 hits within the window it is disabled during the following time slice. The least significant bit of that four bit nibble weights 256 system clock periods. For proper operation of the TDC it is absolutely necessary to have a non-zero time in that register. The range of the noise suppression is therefore from 4.3 to 64.5µs.

Table 7 depicts these registers.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	Time window				1	1	1	1	1	1	1	1

Table 7: Noise Suppression Register

Other Set-Up Operations

To insure proper operation several registers from the Groups 1 and 2 also need to be properly set- up. Registers from group 1 in this category are the Calibration Mode, the ADC Offset DAC and the Pattern Delay DAC if one wants to measure amplitude during data acquisition.

REGISTER READ BACK

Once data has been written into registers it is necessary to verify that the data intended is in fact correct. Therefore we can read back these registers. The ROM architecture is based to a four byte, 32 bit, data path, this being naturally accommodated when reading out from the DFB whose readout format is also based on 32 bit information units. The data transmitted from the DFB to the DAQ always contain four types of words indicated by bits 1 and 0 as follows:

Bit 1	Bit 0	
0	1	<= One DFB Header word
1	1	<= One or more data words
1	0	<= One DFB Status word
0	0	<= Trailer word, 32 zeroes interpreted as an end-of-record by the ROM

The first word sent from the DFB to the ROM is a header shown in table 8. It contains the DFB serial number, the Op-code the board is responding to -read group 0 register-, and the address of the register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1-0
Board Serial Number								-	-	-	-	-	-	-	-	-	-	-	Op-Code Received					Register Address					0	01

Table 8: DFB Read Register Header

The second word, table 9, contains the register data in the most significant 16 bits and the value contained in the Next Transfer Address (NTA)[10] register used in block transfer mode that we will examine in a subsequent chapter11.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1-0
Register Data																Current NTA									-	-	-	-	-	11

Table 9: Register Data Word

The data transfer ends with a Status word, table 10. It contains a status field that reflects the content of the Status register of table 22 on page 19, and the word count for the current transfer.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1-0
1	Board Status															--	--	--	--	--	Word count									10

Table 10: DFB Read Register Status Word

10 ***<u>The NTA on page 19 is only 8 bits, should it be bits 15-7 here?</u>***

11 Word Count and Next Transfer Address Registers are covered in the Group 2 register section on page 18.

EVENT DATA FORMAT

Once the set-up has been done the system is ready to acquire data. But before looking at the format of the data it may be useful to summarize here the TDC functionality which is detailed in reference [12]. The TDC has a free running 11-bit counter clocked at the system clock frequency of 59.5MHz. It is reset to zero when a Sync command is issued (Op-code 02, see table 1 on page 4). A five-bit vernier synchronized on the system clock provides a fine time measurement of 525.21ps. When a hit on a channel is detected, the content of the 11-bit counter and the value of the vernier are saved inside the TDC as a 16-bit hit time for that channel. Upon a L1 Trigger accept command, the TDC stores the instantaneous value of the 11-bit counter and the time-ordered hits which are within the trigger resolution window into an output FIFO.

During a physics run the TDCs continuously record PMT hits. If an event is detected by the trigger system, the ROM issues a L1 Accept command (Op-code 3). The DFBs store the TDC data within the resolution window into one of the event buffers. The DAQ then issues a Read Event command (Op-code 4). Each DFB sends data to the DAQ in the following format.

The DFB header contains: an 11 bit trigger time field which is the value of the internal 11 bit counter from of the first TDC (TDC 0) at the time the L1 Trigger Accept command is interpreted, the DFB board serial number, and the trigger tag associated with the L1 Trigger Accept command which has caused that buffer to be filled.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1-0
0	0	0	0	0	11 bit Trigger Time Generated by TDC											Board Serial Number								Trigger Tag					1	01

Table 11: DFB Read Event Header

The next word, table 12, is the header for the first TDC. It has an 11-bit Trigger time field, which must be the same as the trigger time in table 11 - if the trigger time between TDCs does not match a synchronization command, Op-code#02 will need to be issued. The TDC No. field is zero for the first TDC channels (15-0), up to three for the last TDC channels (63-48).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1-0
11 bit Trigger Time Generated by TDC											0	0	0	0	0	-	-	-	-	-	-	-	-	TDC No.		0	1	0	0	00

Table 12: TDC Header (1 per TDC)

Bits 15-8 contain the same value as bits 31-24 but are not relevant.

The next word, table 13, is the hit data. There is one such word for every hit falling within the trigger resolution window. The most significant 16-bit field is the hit time with 525ps resolution. The charge field comes from the RAM where the ADC output is stored -see Charge Measurement Mode on page 6. This is followed by the TDC number and the channel number within the TDC.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1-0
Hit Time Generated by TDC																8 bit Charge								TDC No.		Channel No				11

.

Table 13: TDC Hits Data, if any (1 per hit in each TDC)

If the TDC receives more than one hit either in the same channel or in different channels, the data words for the subsequent hits follow each other in the order the hits occurred.

Accidentally, a zero time and charge for TDC zero and a channel lower than four might produce 32 zero bits which would be interpreted as a trailer by data flow. To prevent this, the least significant bit of the time is set to one, if ever the above condition is met.

The last word from a TDC is the status word, table 14. The most significant 16 bits correspond to the FIFO full flag for the respective 16 channels.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1-0
Input FIFO full flags: Four deep FIFOs hold 1µs window																0	0	0	0	0	0	0	1	TDC No.		1	0	0	0	00

Table 14: TDC Status (1 per TDC)

The data presented in tables 12, 13, and 14 is repeated for each TDC having received at least one hit. If a TDC got no hit, only the TDC header and status words are generated. Therefore, the minimum number of 32 bit words from a DFB without hits is 10 words and a 32-bit trailer.

The last significant word in the data set is the DFB status word. Therein we find a summary of the FIFO full flags for each TDC and the word count for the data from the DFB. This word count is only for TDC Header, Hits Data, and Status words and does not include the DFB Header, DFB Status and Trailer words.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1-0
1	--	--	--	--	--	--	--	--	--	--	--	OR of input FIFO full flags				--	--	--	--	--	Word count									10

Table 15: DFB Read Event Status Word

Finally, when all the data from a DFB has been transmitted a trailer is sent to indicate the end of record. A trailer is a word with 32 zeroes.

Data in the Intermediate Store

The data formatted as described above is received by the Personality Card of the ROM as explained on page 2. It is first stored in the Intermediate Store where each DFB is allocated a 1k by 32-bit block of memory. Table 16 is an example of how DFBs' event data may be stored in the IS.

The content of the IS will later be DMAed to the processor memory.

DFB No.	TDC No.	32 bit word	Refer to	Comments
DFB 0 1k x 32 bits Memory block		DFB Header	Table 11	
	TDC 0	TDC Header	Table1 2	DFB 0, TDC 0 has two hits
		Hit Data	Table 13	
		Hit Data	Table 13	
		TDC Status	Table 14	
	TDC 1	TDC Header	Table 12	DFB 0, TDC1 has no hit
		TDC Status	Table 14	
	TDC 2	TDC Header	Table 12	DFB 0, TDC 2 has three hits
		Hit Data	Table 13	
		Hit Data	Table 13	
		Hit Data	Table 13	
		TDC Status	Table 14	
	TDC 3	TDC Header	Table 12	DFB 0, TDC 3 has no hit
		TDC Status	Table 14	
		DFB Status Word	Table 15	DFB 0 has no more Data
		Trailer	Table 16	
	Possibly old data from a previous event			
DFB 1 1k x 32 bits Memory block		DFB Header	Table 11	
	TDC 0	TDC Header	Table 12	DFB 1, TDC1 has no hit
		TDC Status	Table 14	
	TDC 1	TDC Header	Table 12	DFB 0, TDC1

Table 16: DFB Raw Data Loaded into the Intermediate Store

CALIBRATION

The line between Calibration and Board Internal Tests, as mentioned on page 4, is rather thin. Here we keep them separated with the philosophy that the former is somewhat analog and the latter is strictly digital.

Calibration Background

In BaBar the calibration procedure for all the detector subsystems must follow a rather complicated path to insure independency between subsystems as well as overall BaBar calibration. It involves partitioning as described in the FCTS documents [3 and 4], and in the Data Flow documentation [14, 15, and 16]. In summary, for the purpose of this DIRC document, a detector subsystem or a group of detector subsystems can be selected to follow a calibration procedure independently of the rest of BaBar. Through the FCTS a calibration signal is injected and sent as a command, on the CLINK, to the FEEs -DFB or DGB in DIRC. This calibration command (Global Op-code 5) causes the DFB, or the DGB, to generate a precisely timed pulse called calibration strobe. The calibration strobe is followed by a L1 Trigger accept command (Op-code 3) exactly one latency time later and then by a Read Event command (Op-code 4). Therefore, if the calibration strobe, somehow, injects a known signal into the DFB, the electronics chain can be calibrated for the optimal detector's performance.

The DIRC Electronics Requirements [17] give a list of means used to guarantee the proper functioning of the DIRC. However it encompasses what are considered internal tests within this document. After the Preliminary Design Review, another document -Electronics Commands for the DIRC Calibration [18]- was generated to answer the reviewers' concerns. This last paper presents a fairly complete overview of the calibration that will be covered herein.

There are only few knobs available to guarantee an adequate operating range. They are the high voltage adjustments and the amplifier sensitivity

In the DIRC, calibration of the electronics chain will be as close to the detector as the PMTs. On a calibration command each DIRC sector can be illuminated by a blue flashing LED located in the electronics house as explained in BaBar note 25 [18]. The light is piped to each sector by a high quality optical fiber. The PMTs detecting this light will transmit a signal to their respective DFB the same way as would happen for a physics event. The timing and the intensity of the light pulse can be controlled to suit the best operational conditions. The timing of the pulse is controlled by a register located on the DGB. The DGB decodes the commands the same way as a DFB would; but, these particular commands are a No Op for the DFBs. This calibration will enable us to set the best operating high voltage on the PMTs and verify the timing chain.

A second type of calibration is implemented to calibrate the analog front-end of the DFBs; that is, to set the gain for each analog chip as seen on page 6. A pulse simulating a PMT hit is generated on the DFB itself and injected into the input of the analog chip. The amplitude of the pulse used to simulate the charge is controlled by a DAC.

Above we looked at the DIRC electronics calibration starting as close as possible to the detector; but we will most likely start the calibration by first setting the threshold of the analog chips to a known value and then set the PMT high voltage.

Calibration Register (Group 1)

As seen earlier, on page 4, the registers to set-up the calibration belong to register group one. The Op-code to write to and read from these registers is shown in table 17.

Bit 9-5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Address	1	0 = read 1 = write	0	0	1

Table 17: Group 1 Registers Op-code

The convention used for writing to and reading from these registers follows the same rules as seen in the Group 0 Registers section. In this register group there are five different set-up register types, two of them are DGB registers, the remaining are DFB registers, among them, one type has addresses for four DACs.

Calibration Mode Register:

Address 0

Table 18 shows the format of this register.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Not Used								Erase Charge	Mask All	Auto Mask	Cal Strobe Mode			Odd	Even

Table 18: Calibration Mode Register

When **bit 7** is set, the charge RAM, where the value of the flash ADC is stored, is reset after it is read out. It is necessary by default.

When **bit 6** is set, all channels are disabled, which would be the equivalent of setting the ANDer mask registers to zero.

Bit 5: this bit is provided to mask the trailing edge of the calibration pulse that may cause parasites in the front end of the analog chip and fire some channels consequently. The mask is active 200ns after reception of the calibration strobe and lasts 1µs, in order to cover properly the necessary duration. *This bit must be set to one when performing the DFB's internal calibration, but only in this case.*

The Calibration Strobe Mode field is used to define the source and function of the calibration strobe as follows:

0 <= **Normal run mode**: the signal is generated by the PMTs, the calibration strobe is not generated.

1 <= **LED Pulsing**: the blue LEDs are pulsed by a calibration strobe generated by the DGB. The DFBs ignore this command and treat the data the same way as physics data.

2 <= **DFB internal analog calibration**: analog pulses are generated by the DFBs towards the analog chips upon reception of a calibration strobe to simulate

signals from the PMTs. As explained below, even and odd channels can be pulsed each independently of the others. ***Warning***: *the Pattern Delay DAC* ***and*** *the Calibration Timing Register must be properly set*.

3 <= **Digital Test Pattern**: this is part of what we consider here internal tests; however, since the DFB needs to generate a calibration strobe for this test, the mode needs to be included here. Upon reception of a calibration strobe, the DFB injects a series of patterns formerly loaded in the so named Pattern FIFO into the input of the TDCs. This is a powerful and deterministic way to check the TDC and the whole readout system. ***Warning***: *the Pattern Delay DAC must be properly set*.

4 <= **Pedestal mode**: in this mode, an ADC conversion is launched by the DFB upon reception of a calibration strobe regardless of the input to the ADC. No signal other than the quiescent PMTs output has to be injected into the analog chips. ***Warning***: *the Calibration Timing Register must be properly set*.

When set to one, bits 1 and 0 enable respectively odd and even channels calibration pulses. As we will see later, pulses of different amplitudes can be injected simultaneously in the even and odd channels.

The default setting for this register is bit 7 set (0080^{16}) for normal data acquisition, to which you should add bit 2 (0084^{16}) for LED calibration.

Calibration Timing Register:

Address 1

Table 19 shows the format of this register.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	-	-	ADC Conversion Delay					
										7ns			2ns		

Table 19: Calibration Timing Register

The ADC Conversion Delay field (6 bits) of this register allows the user to adjust (a priori forever) the delay between the calibration strobe and the start of the conversion by the ADC to catch the peak of the signal. Bits 5 down to 3 have a least significant weight of 7ns, and bits 2 down to 0 have a 2ns weight.

The default setting for this register is $002F^{16}$ (which corresponds to a delay of 47ns).

The following two registers, LED Pulse Timing and LED Pulse Mode, are located on the DCC and must be treated differently than the registers on the DFBs. The DFBs will not respond to the commands sent to the DCC. When addressing the DCC with the CLINK, the ROM's CLINK enable register needs to enable the CLINK for DFB zero.

When reading back the two registers located on the DCC the data format is somewhat different than what we saw on page 10. In this case there is no DFB Read Register Header and no DFB Read Register Status word. Bits 1 and 0 are set to one and are followed by the register content.

LED Pulse Timing Register:

Address 8

This register has only ten significant bits, bits 9 through 0, to set a vernier which determines the exact time the LED will flash. The calibration strobe is always in time with the system clock that has a period of 16.8ns. With this register the flashing of the LED can be set anywhere within the 1µs resolution window around the calibration strobe in 500ps steps. To accomplish this there are two verniers, as shown in table 20 that provides a range from zero to 536ns.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	System Clock Vernier (16.8ns)					500ps Vernier				

Table 20: LED Pulse Timing Register

LED Pulse Mode Register:

Address 9

This register, is also located on the DCC and is accessed in the same manner as the Pulse Timing register. It has only bit zero used to enable the pulsing of the LED when it is set to one. When reset to zero the calibration command is ignored when the calibration mode register specifies LED pulsing.

DAC Registers:

Addresses

10: **Odd Channels Input Pulse Level** delivered in DFB analog calibration mode.
11: **Even Channels Input Pulse Level** delivered in DFB analog calibration mode.
12: **Pattern Delay Vernier**, used in all calibration modes except Pedestal mode. *This register shall **never** be loaded with a value above 0500^{16} (-3.5V).*
Its default value is 0200^{16} (-4.4V). It might be changed only for an expert's Pattern Calibration.
13: **ADC Offset Adjustment** is used to set the range of the ADC to allow an optimized match between the ADC and the output offset of the analog chips.
Its default value is 0870^{16} (-2.2V).

Each one of these registers is 12-bit wide and is written within the range of bits 11 down to 0 in the 16-bit word. The DAC outputs range from -5V (0000^{16}) to 0V ($0FFF^{16}$).

Calibration Procedures

This section to be completed later

BOARD INTERNAL TESTS

This register group has been implemented to provide a means to verify proper operation and troubleshoot the DFBs. It allows to look at several statuses within the DFB, write to and read from the event and ADC buffers, and run dynamic tests on the TDCs.

Internal Test Registers (Group 2)

As with the other register groups table 21 shows the Op-code to write to and read from these locations.

Bit 9-5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Address	1	0 = read 1 = write	0 = single word 1 = block mode	1	0

Table 21: Group 2 Registers Op-code

Even though we call this group a register group, some location to be written into and read from are memories -FIFO and RAM- which have several locations. Therefore, we will introduce in this section the block transfer mode which makes use of the Next Transfer Address (NTA) and Word Count (WC) registers. These two registers are written into from the ROM prior to a block transfer to or from the DFB. The NTA register provides the memory address which will be accessed during the **next** block mode 16-bit transfer regardless of read or write and the WC register keeps track of the remaining words that need to be transferred.

The block mode transfer has a slightly different implementation when the command is a write to the DFB command -bit three set- and when it is a read from the DFB command. However, the function of the NTA and WC registers is the same in both cases. In the case of a write command, the command needs to be issued with a 16-bit data field for every word, the NTA register incrementing and the WC register decrementing after each writing. A read command causes the full data set, up to 511 words, to be transmitted to the ROM. The first word is as usual the DFB header (table 8 on page 10). It is followed by up to 511 data words (table 9) until the WC register reaches the value 0. The NTA register increments after each word transfer to address the next memory location in RAM. The DFB status word (table 10) and a trailer, 32 zeroes, are inserted to indicate an end of record.

In this section we also need to introduce the notion of Multi-Event-Buffers. Up to now we have seen that the DFB has four event buffers, and it is straightforward to understand the functionality of the system. However, the hardware implementation is somewhat different. Indeed there are four buffers to store event data, but more specifically there is a Multi-event buffer associated with each TDC chip. Each Multi-event buffer, implemented as a 512 word deep FIFO, can store up to four events from a TDC in contiguous FIFO memory.

We will now look at the details of the group 2 registers.

Status:

Address 0

This read only register is a hodge-podge of status bits shown in table 22.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	-	-	-	Buffer 3		Buffer 2		Buffer 1		Buffer 0		Test Pattern		1	1
				F	E	F	E	F	E	F	E	F	E		

Table 1: Internal Test Status Register

Since the status word is the last word transmitted before the trailer with a read register command (page 10), bit 15 must be set to one to prevent the ROM from detecting the trailer too early.

Bits 11 through 4 are the FIFO Full and Empty flags for the four Multi-event buffers, bits 3 and 2 are for the Test Pattern FIFO that we will introduce later. Bits 1 and 0 show the number of L1 Trigger Accept commands received since the last Read Event command was issued; therefore the number of event buffers occupied.

Next Transfer Address:

Address 2

The eight least significant bits of this register correspond to a RAM address. An explanation on the use of the NTA and WC registers appears in the preceding page.

Word Counter:

Address 3

The word counter is pre-loaded prior to block mode data transfer to indicate the number of 16-bit words that will be transferred.

Test Pattern FIFO:

(One for four TDCs)

Address 8

This FIFO is 16 bit wide and 512 words deep. It is used to inject a test pattern into the respective inputs of the four TDCs. It is written into one word at the time and read from in block mode as described on page 18. However, since this is a FIFO the content of the NTA and WCR are irrelevant, only the number of words necessary for a test need to be written into the FIFO. The test pattern timing is started by a Calibration Strobe pulse generated on the DFB, but since the Calibration strobe is synchronous with the 59.5MHz a delay -Pattern Delay- allows to position the pattern with a 30ps[12] precision. This delay is set by the group 1 -calibration group- DAC register at address 12 shown on page 17.

12 This is the precision in the usable range (DAC value between 0000^{16} and 0500^{16}).

Address C

A data-less Group two command at this address resets the FIFO read pointer to allow a new calibration strobe to start transmitting the present FIFO's patterns. This command should also be executed to read back the test pattern FIFO after a calibration strobe.

Multi-Event Buffers:

(One per TDC)

Address 17-10

This command allows to write into the four Multi-event buffers for each TDC which were introduced on page 18. Note that only data related to the TDCs is written into the Multi-event buffers, the ADC data is written into a separate RAM that is also read-out after a Read Event command is issued.

The write command to the Multi-event Buffers is a block mode command to a FIFO. The data written into the event buffers can subsequently be read with a fake Read Event command (Op- code 4) generated by the ROM. The format of the data sent to the ROM is the same as the one shown for event data on pages 11 and 12, with the exception that the TDC data is the data just written into the event buffers and the ADC data is whatever is in the ADC RAM at that time. Each multi-event buffer is 512 words deep and 22-bit wide and must therefore be written into using two block mode transfers.

Addresses 10 through 13 address 16-bit words that emulate the hit time in each Multi-event buffer as shown in table 13 on page 12. Addresses 14 through 17 address the remaining 6 bits associated with the TDC, namely a two bit TDC number field and a 4-bit channel address.

Address 18

Similarly to the Group two command with address C, above, a data-less Group two command with address 18 resets the read address pointer of the Multi-Event Buffer FIFO.

Flash ADC RAM:

Address 19

The output of the flash ADC that measures the charge from a PMT is written into a 64-byte RAM. This RAM can be written into and read from with this command in write and read block mode, the NTA registers specifying the memory internal address to be accessed. The data consists of the least significant eight bits of the 16-bit word being transferred.

In principle this RAM can also be read with a Read Event command. However, only the channels with a TDC hit will be read out in this mode.

Reset Board:

Address 1F

This data-less command is used to reset the DFB. It is not recommended to use this command since it destroys the content of the TDCs registers and the DACs value.

Summary of the DIRC Registers

Address	Register	Page
7-0	Analog Chips Gain DACs	6
08, 0A, 0C, 0E	Charge Measurement Mode Registers	6
09, 0B, 0D, 0F	ANDer Channel Enable Pattern (Mask) Registers	7
10, 14, 18, 1C	TDC Window Register	7
11, 15, 19, 1D	TDC Selective Readout Registers	8
12, 16, 1A, 1E	TDC Internal Mask Registers	9
13, 17, 1B, 1F	TDC Noise Suppression Register	9

Table 23: Readout Set-up Registers (Group 0)

Address	Register	Page
0	Calibration Mode Register	15
1	Calibration Timing Register	16
8	LED Pulse Timing Register	16
9	LED Pulse Mode Register	16
10	Odd Channels Calibration Pulse Level DAC	17
11	Even Channels Calibration Pulse Level DAC	17
12	Pattern Delay Vernier DAC	17
13	ADC Offset DAC	17

Table 24: Calibration Registers (Group 1)

Address	Register	Page
0	Status Register	19
2	Next Transfer Address Register	19
3	Word Counter	19
8	Test Pattern FIFO	19
C	Test Pattern FIFO reset read pointer	20
13-10	TDC Multi-event Buffers 16 Bits (Hit Time)	20
17-14	TDC Multi-event Buffers 6 Bits (Channel Number)	20
18	TDC Multi-event Buffers reset read pointer	20
19	Flash ADC RAM	20
1F	Reset Board	20

Table 25: Internal Tests Registers (Group 2)

REFERENCES

[1] The DIRC Subsystem Commands
Specifications for the DIRC Front-end Board, version 2.0 (64 channel DFB)
DIRC Electronics group at Orsay, March 19, 1997
http://www.slac.stanford.edu/BFROOT/doc/Electronics/Common_Electronics/fee.html
DIRC Protocol

[2] Electronics Control and Dataflow between the Read-Out Module and the Front-End Electronics Systems.
G. Oxoby and G. Haller 4/15/96
http://www.slac.stanford.edu/BFROOT/doc/Electronics/Common_Electronics/fee.html
Note 281 Description of FEE/ROM Protocol Version 1.1

[3] Conceptual Design of the BaBar Fast-Control and Timing System
G. Haller, Apr 16, 1997
http://www.slac.stanford.edu/BFROOT/doc/Electronics/Fast_Cntrl_Tim_System/index.html
Fast Control Conceptual Design Document Ver. 3.0

[4] Architecture for the BaBar Fast Control and Timing System
L. Sapozhnikov, Jun 10, 1997
http://www.slac.stanford.edu/BFROOT/doc/Electronics/Fast_Cntrl_Tim_System/index.html
Overview of Read-Out Module

[5] DAQ Readout Module Overview
G. Haller et al., Draft Nov 19, 1996
http://www.slac.stanford.edu/~leosap/fcts.html
Write-up

[6] Read-Out Module Architecture
J. Dowdell., Nov 20, 1996
http://www.slac.stanford.edu/BFROOT/doc/Electronics/ReadoutModule/rom.html

[7] Architecture for the BaBar Triggered Personality Card.
G. Haller, March 14, 1997
http://www.slac.stanford.edu/BFROOT/doc/Electronics/ReadoutModule/rom.html
Triggered Personality Card Architecture

[8] BaBar DAQ Controller Card Description
J. Dowdell, June 26, 1997
http://www.slac.stanford.edu/BFROOT/doc/Electronics/ReadoutModule/rom.html
Triggered Personality Card Architecture Rev. 5

[9] PCI Mezzanine Card
G. Oxoby, March 5, 1997
http://www.slac.stanford.edu/BFROOT/doc/Electronics/ReadoutModule/rom.html

PCI Mezzanine Card Description

[10] BaBar Note 324: DIRC Electronics
P. Bailly et al. Oct 2, 1996

[11] Synopsis of DIRC Front-end Board (Drawing)
Antoine Ducorps, L.A.L. Orsay - June 13, 1997
http://www.lpnhep.in2p3.fr/babar/ele/dfb/dfb.html
DIRC Front End Board Block Diagram

[12] Development d'un circuit integre VLSI assurant mesure de temps et lecture selective dans l'electroniquefrontale du compteur DIRC de l'experience BABAR a SLAC
THESE Presentee par Zhang Bo, Universite Pierre et Marie Curie, PARIS VI, Septembre 29, 1997

[13] DIRC Digital TDC Chip Reference
P. Bailly et al., June 12, 1997
http://www.slac.stanford.edu/~genat/fdr/digchip/digchip.html
DIRC Digital Chip Reference, June 1st, 1997

[14] DAQ Programmer's Guide
R. Hamilton et al.
http://www.slac.stanford.edu/BFROOT/doc/Electronics/DataFlow/OnLine DataFlow.html
Data Acquisition Programmer's Guide

[15] The DataFlow Platform User's Guide
R. Hamilton et al.
http://www.slac.stanford.edu/BFROOT/doc/Electronics/DataFlow/OnLine DataFlow.html
DataFlow Platform User's Guide

[16] The DataFlow Reference Manual
R. Hamilton et al.
http://www.slac.stanford.edu/BFROOT/doc/Electronics/DataFlow/OnLine DataFlow.html
DataFlow Reference Manual

[17] DIRC Electronics Requirements
The DIRC Electronics Group, Feb 20, 1996
http://www.slac.stanford.edu/~genat/fdr/fdr.html
DIRC Electronics Requirements
Also DIRC Note 29

[18] DIRC Electronics Commands for the DIRC Calibration
The DIRC Electronics and Calibration Groups, July 12, 1996
http://www.slac.stanford.edu/~genat/elec.html
DIRC Calibration document for Dresden

[19] DIRC Note 25 Report on the DIRC LED Calibration System

P. Bourgeois et al. Feb 26, 1996
http://www.slac.stanford.edu/BFROOT/internal/DIRC/list.html
Report on the DIRC LED Calibration System