Datimizer.

Introduction:

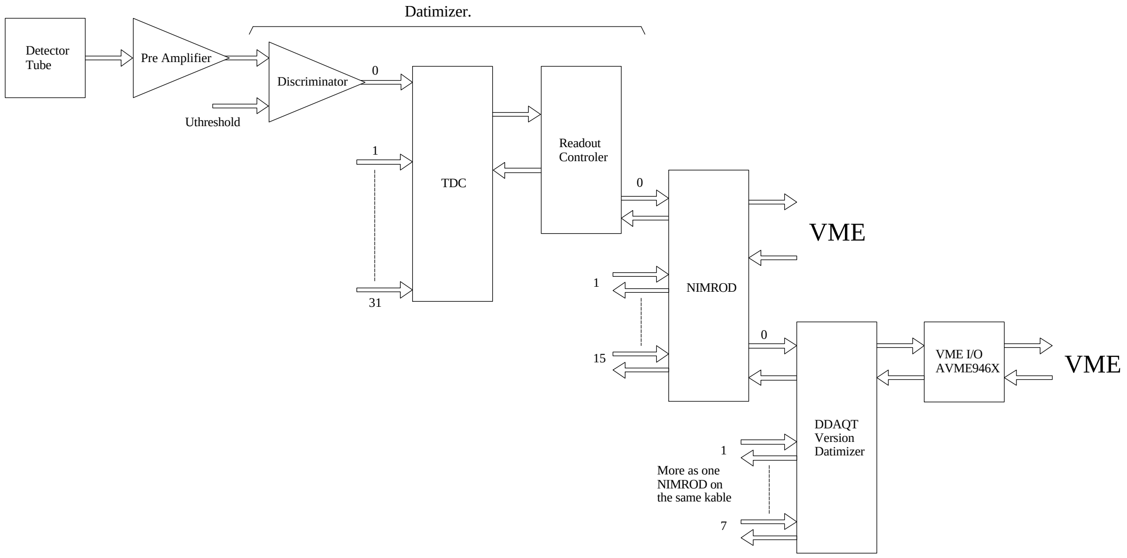


Figure 1: Schematic of the DATCHA readout.

In figure 1 the diagram of the Datcha readout system is drawn. In the Datimizer three functional block of this diagram are realised.

These blocks are: discriminator,
TDC,
Readout control.

In the following chapters these blocks will be described separately.

Chapter 1: The Discriminator.

In figure 2 the schematic of the discriminator is given.
This schematic is based on the discriminator of the Bimux card used in the DATCHA set-up now.

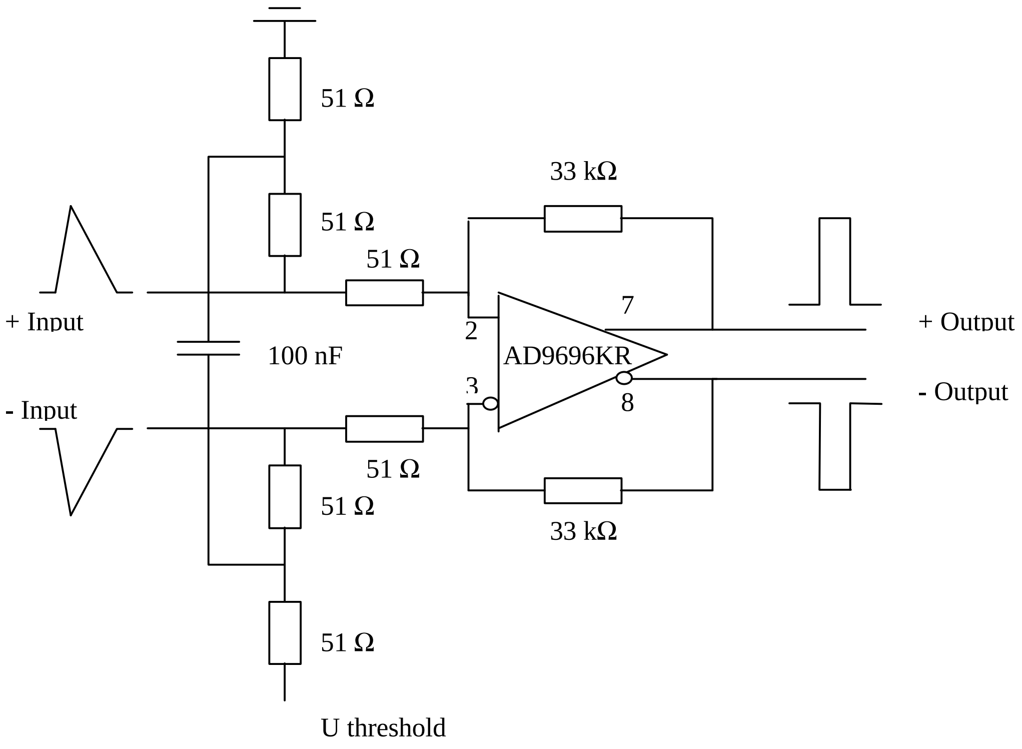


Figure 2: The schematic of the discriminator.

In compare to the original Bimux discriminator three changes are made, because the specifications of the discriminator are different. These changes are:

1	The discriminator in the Bimux works also as an oneshot. The TDC can handle much shorter pulses as the logic in the Bimux, so the need for an oneshot was no longer there.
2	The hysteresis of the discriminator is made four times bigger, to 15 mV. This change is realised by putting in resisters in series with the input of the discriminator and adding a positive feedback also from the other output. Because of this the circuit has become symmetrical.
3	Between the output of the pre amplifier and the discriminator resistors are placed. These resistors attenuate the signals from the discriminator to the pre amplifier.

After the design was made I a prototype was build. As input circuit a Fbpanic-04 pre amplifier was used.
The results of the test with this prototype are given in the chart.

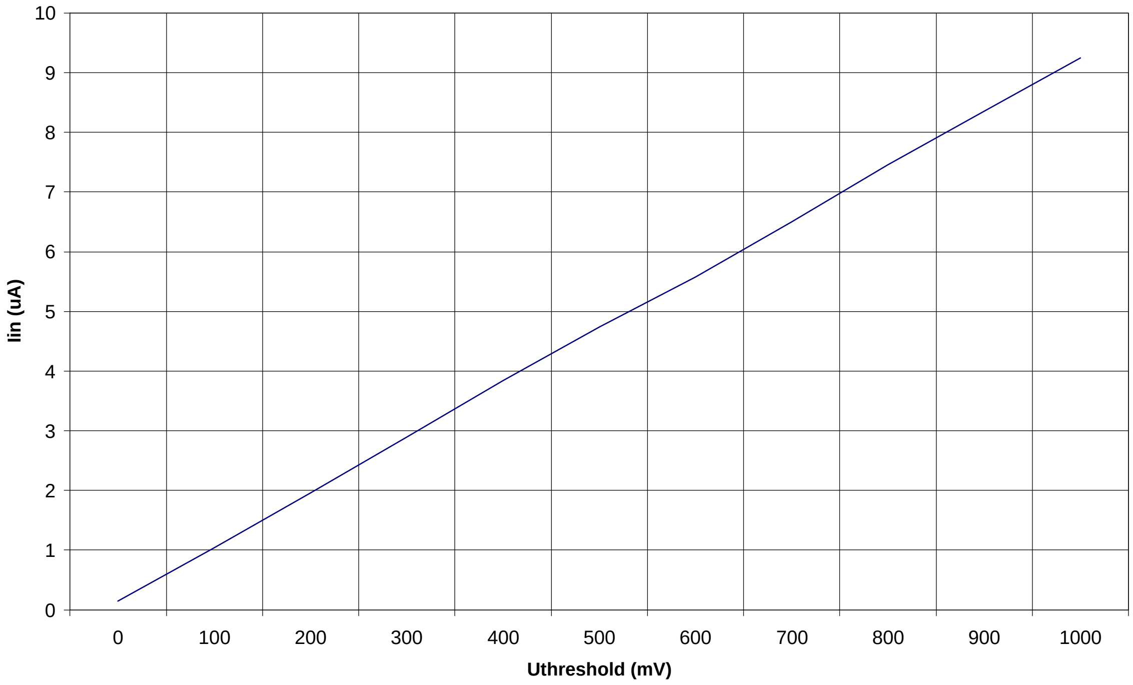


Figure 3: Sensibility of the circuit against the Threshold voltage.

As you can see is the result between the U threshold and the Input current linear. The pre amplifier should give 100 mV for every µA. In this case he did give a little greater signal. This is due to normal tolerances of the components.

The discriminator Outputs:

The outputs of the discriminators are bipolar TTL signals. The positive going output is connected to the TDC for time digitising.
Both outputs of the discriminators are wired also to two extension connectors. This opens the possibility to upgrade the system by making an extension board on the Datimizer.

Input options of the Discriminator.

In the set-up above its only possible to use the Datimizer in combination with a pre amplifier with bipolar voltage outputs. Soon after the start of the Datimizer project a second client appeared who also wants to use the Datimizer in his readout system. The problem was that his detector electronics had a bipolar current outputs, somewhat like a LVDS outputs, except that they sank current thou a resistor of 51 Ω from earth. To support this kind of outputs we added two resister of 0 Ω to the input of the discriminator. By assembling on the PCB the components marked with a * or marked with a $ we can choose between current input and discriminator.

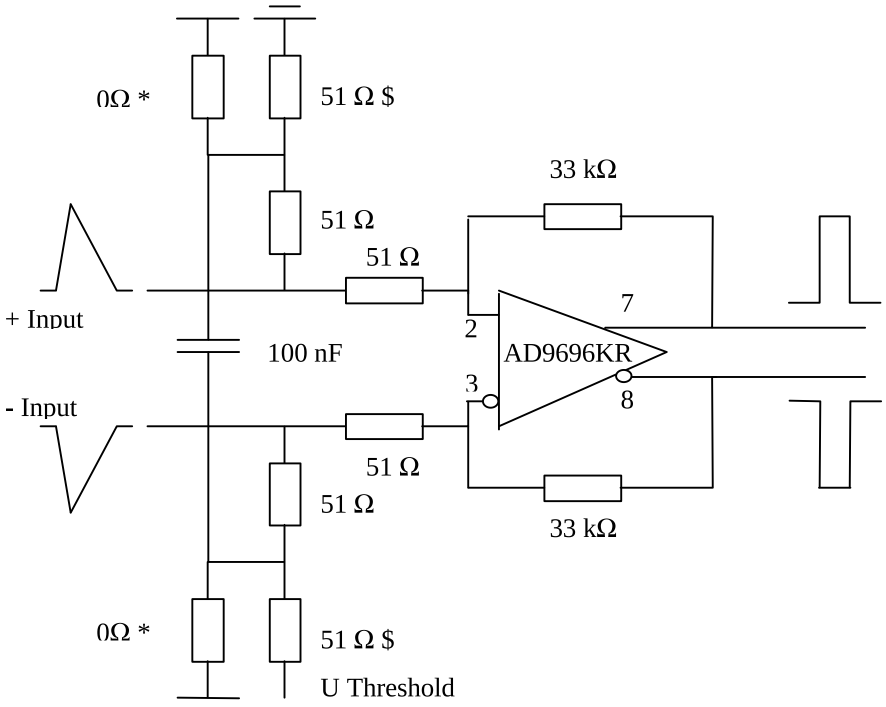
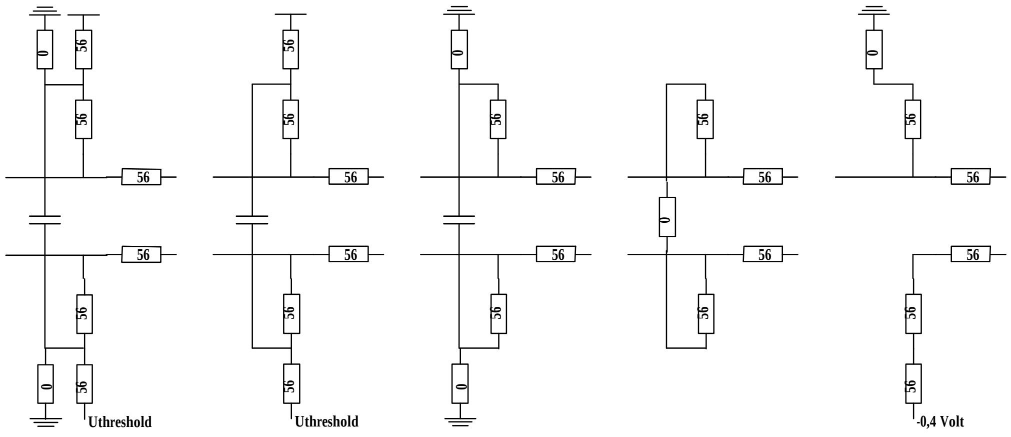


Figure 4: The schematic of the Discriminator on the Datimizer PCB.

Looking at the schematic (figure 4) of the discriminator on the PCB we realised that there are more input options possible. In figure 5 we have drawn al options.



Schematic of the datimizer discriminator.

Schematic of the datimizer discriminator. As discriminator. Input impedance 110 Ohm.

Schematic of the datimizer discriminator. With current inputs. Input impedance 110 Ohm

Schematic of the datimizer discriminator. As ECL of LVDS receiver.Input impedance 110 Ohm

Schematic of the datimizer discriminator. NIM standard inputconvertor, inverting. Input impedance 56 Ohm.

Figure 5: Input options discriminator.

Chapter 2: The TDC.

The most important component of the Datimizer is the TDC.
Chosen is the 32 Channel general purpose Time to Digital Converter developed by Jorgen Cristiansen at CERN.

The time resolution of this chip is 0.78 ns. This is realised in two steps:
1 A course time counter at 40 MHz, resolution 25 ns.
2 A 32-stage delay lock loop, resolution 0.78 ns.

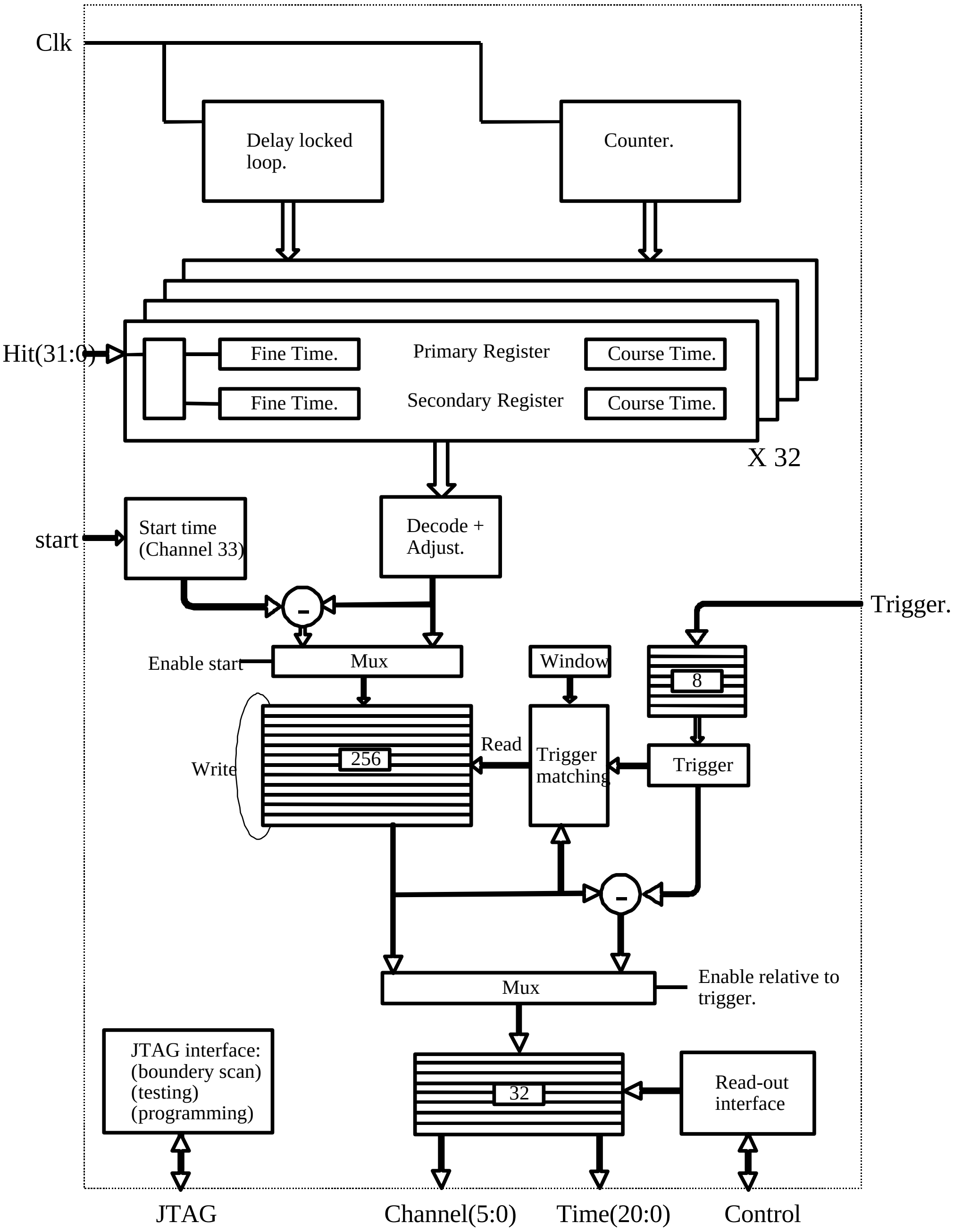


Figure 6: Functionally diagram of the TDC.

Functionally diagram of the TDC is given in figure 6.

On top of the diagram the Delay Locked Loop divine’s the resolution of the TDC. He divides the 25 ns of the clock (40 MHz) into 32 steps of 0.78 ns. The clock also drives the course time counter.
Both these blocks deliver time information to the channels.

Each channel (of 32) has two registers (primary and secondary) to store the time information of the incoming signal from a detector.

During the measurement the data is transferred from these registers to a 256 places deep circular buffer. This transfer is controlled by a ‘registered arbitration scheme’. In this scheme the transfer requests are latched into a register. The requests in the register are transferred from the channel registers to the buffer with the priority low numbers first. Afterwards the register is loaded again with the requests that did arrive after the last load. This system prevents the continuously pushing back of low priority channels due to arriving data in channels with a higher priority.

During the transfer the start time can be subtracted from the channel time. This option is selectable through the JTAG interface.

The data in the buffer is a gathering of all data from the inputs. Part of this data is rubbish; the rest belongs to one or more events.
The ‘Trigger matching logic’ does the search the buffer for relevant data. All parameters for the Trigger matching logic are set through the JTAG interface.
The trigger moment is stored into the trigger FIFO (just the course time). Related to this time a trigger window is divined. All data with a time stamp inside the trigger window belongs to this trigger (event).
Wen two triggers arrive shortly after each other it is possible that the same data belongs to two triggers.
This data is then stored in the readout buffer with event separators between them.
During this last action the trigger time can be subtracted from the data, to make the time relative to the trigger. This option is selectable through the JTAG interface.
During the search for data in the buffer a start search pointer is set at the first hit matching the trigger. Wen no trigger arrives during some time it is possible that the write pointer over takes the start search pointer, and pushes it forward. For the first trigger that arrives the system will need a long searching time, because the data belonging to it is standing on the other end of the buffer. To shorten this time the automatic ‘event reject function’ is build in. Wen the time of the start search pointer is older than the reject time counter, this mechanism will increment the start search pointer.

Chapter 3, The readout controller.

The readout controller is divided into two functional blocks:

1. The FE-link.
   1.1. On the output side of the Datimizer a FE-link is used. This link transports four signals: 40 MHz clock signal and trigger/reset signal to the Datimizer, data and strobe signal from the Datimizer.
      1.1.1. 40 MHz clock. This is the base clock for the Datimizer. In the TDC a Delay Locked Loop is controlled by this clock, so he must be distributed with care.
      1.1.2. Trigger/reset signal. On this line the trigger and reset signals are combined. The readout controller decodes this line to two signals and sends them to the TDC and the readout logic. The trigger pulse is one clock period long followed by two periods zero. The reset signal is three clock periods long.
      1.1.3. Data and strobe. The data received from the readout logic and the TDC is encoded to the data/strobe protocol for frequency reduction on the FE-link. In short this protocol works as follows: the data is put on one connection. A strobe is generated on the moments between two bits of the data word, were they are the same. So no strobe when there is an edge on the data line, and strobe when there isn't an edge on the data line.

2. The readout logic.
   2.1. The first task after a trigger is generating a data header
   2.2. The second task is reading al the data of the TDC into the data/strobe converter.
   2.3. The last task is generating a data trailer.

Data format header:

31. . . 28	27. . . 24	23. . . 12	11. . . 0
1010	0100	Event number	Datimizer number

Data format TDC data:

31. . . 28	27	26 . . . 24	23 . . . 18	17	16 . . . 5	4 . . . 0
0011	Error bit	TDC number	Channel	Leading or trailing edge	Coarse count (25ns)	Fine count (25ns/32)

Data format trailer:

31. . . 28	27. . . 24	23. . . 12	11. . . 0
1101	0101	Event number	Word count

Chapter 4, AJTAG.

On the datimizer adresseble JTAG is used.
The JTAG chain is drawn in figure 7. In the readout the adresses are:

IR	Discription
0000	User identity code
0001	Control register
0010	Setup register
0011	Version register
0100	Diagnostic register
0101	DAC register

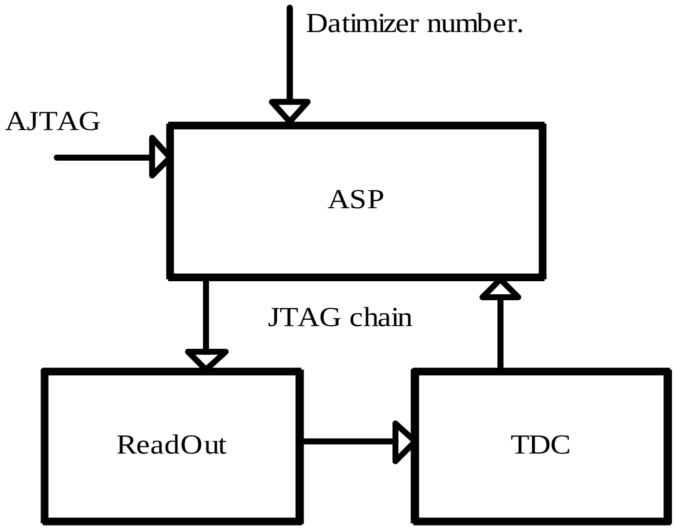


Figure 7: AJTAG chain.

User Identity Register:

This 8 bit word is used to identify the datimizer during readout through the Front End Link. This function is not jet implemented in the current version of the software.

Control Register:

This register controlles the ROC functions. The contents off this register may only been changed when the trigger is disabled.

Bit	Function	Default	Decription
0	Jreset	0	Generate Reset signal for ROC and TDC
1	Jtrigger	0	Generate Trigger signal for ROC and TDC
2	Not used	0	
3	Not used	0	
4	EnETrigger	0	Enables external Trigger and reset from test connector
5	EnART	0	Enables artificial Reset and Trigger signals generated via Control register bits 0 and 1.
6	Inhibit FE Link	1	Disable Reset/Trigger signals via FE link
7	Inhibit ROC	1	Disable ROC (Holds the reset signal for the ROC and the TDC)

Setup register:

Bit	Function	Default	Decription
0	EnTDC	1	Enable the readout from the TDC
1	Not used	0	
2	Not used	0	
3	Not used	0	
4	Not used	0	
5	EnEOG	1	Enable the End Of Group word
6	EnSUM *	0	Enable checksum word
7	EnSTAT *	0	Enable status Word

The functions market with a * are not jet implemented.

Version Register:

Shows the version number of the software in the chip.

Diagnostic register:

Bit	Function	Default	Decription
0	Active	0	The TDC or a internal register has the token and is preparing data for readout.
1	Data Ready	0	The TDC or an internal register has set the Data Ready signal
2	Token Away	0	The ROC has send out the token to the TDC
3	TdcAct	0	The TDC Active signal
4	TdcErr	0	The TDC Error signal
5	GetData	0	The ROC is asking for data from the TDC or an internal register
6	SerBusy	0	The data serializer is sending a word vie the FE Link
7	SendEvent	0	The ROC received a trigger and is sending out data belonging to this event

The last register is only mend to be used by experts for debugging purposses.

DAC:

8 bits DAC register contents Threshold voltage.

JTAG of the TDC

The second chip in the AJTAG chain is the Christiansen TDC. The information about the register in this chip are copied from the discription of the TDC [1]

JTAG Test and Programming port

The TDC is tested and programmed through a JTAG port.
The JTAG protocol is today an accepted standard for chip and module testing, and easy to use debugging tools are avaleble on the PC. Full boundery scan is supported to be capable of preforming extensive testing of TDC modules while located in the system. In addition special JTAG registers have been included in the data path of the chip to be capable of preforming effective testing of all timing registers and embedded memorystructures.
Programming the device is separated into two scan path groups. The setup group consists of setups which can not be changed while the system is active (trigger window, looking back window, enable of trigger matching, etc.) and a control group which can be changed during a run (enabling an disabe of noisy channels). To save silicon area for the large shift registers for the programming data they are not made with two sets of JTAG registers (shift and update register). This has the invonvenience that if the programming data are read via the JTAG (only during functional testing of devices) one of the programming bits will be lost and the bit sequence will be shifted by one.
An additional scan path is avaleble to read the status information fron the TDC while it is running (error flags).

The JTAG instuctions

The JTAG instruction register is 4 bits long. The JTAG controller macro from ES2 does not support being read from JTAG. When passed through the JTAG state CAPTURE-DR the instruction register gets to set to 0001 Bin (IDCODE).

IR	Register	Description
0000	EXTEST	Boundery scan for test of inter-chip connections on module.
0001	IDCODE	Scan out of chip identification code
0010	SAMPLE	Sample of all chip pins via boundery scan registers
0011	INTEST	Using boundery scan registers to test chip itself
0100..0111	Not Used	
1000	Setup	Load of setup data
1001	Control	Load of control data
1010	Status	Read of status information
1011	Coretest	Access to internal test scan registers
1100..1110	Not used	
1111	BYPASS	

[1] A 32 channel general purpose Time to Digital Converter, J. Christiansen CERN/ECP-MIC.

Boundery scan register

All signal pins of the TDC are passed through JTAG boundery scan registers. All JTAG test modes related to boundery scan registers are supported (EXTEST, INTEST, SAMPLE).

BSR#	Pin name	Description
0	Enable Data_ready	Enable of data_ready drive (not a direct pin)
1	Enable bus	Enable of read-out bus drivers (not a direct pin)
6:2	Vernier[4:0]	Read-out vernier
22:7	Coarse out[15:0]	Read-out coarse time out
28:23	Channel[5:0]	Read-out channel
29	Rising	Read-out rising edge
30	Active	TDC active in read-out
31	Event-end	Read-out event end
32	Data-ready	Read-out data ready
33	Error	Programmable error status output
49:34	Coarse_in[15:0]	Read-out time coarse in (parallel trigger in)
50	Chip select	Read-out chip select
51	Next event	Read-out next event
52	Get data	Read-out get data
53	Trigger	Trigger in
54	Reset	Reset of buffers and counters
87:55	Hit[32:0]	Hits
88	clk	Clock

ID code

The 32 bit identification code.

Bits#	Bit name	Description
0		Start bit =1
11:1	ES2 code	ES2 manufaturer code = 00001000111
27:12	TDC part code	decimal 3500 = binary 0000110110101100
31:28	Version code	First version = 0000

Set-up register

This register is used to programme the set-up of the TDC. It can not be changed during run. For testing purposes it is possible to read the status of internal counter via this path.

Bits#	Description
15:0	Yes/no trigger time tag offset [15:0]. (read back of trigger time tag counter)
16	Enable parallel trigger [*]
17	Enable sync trigger *
18	Enable serial trigger *
34:19	Trigger matching window [15:0]. (read back of active trigger)
35	Enable read-out of start measurement
36	Enable subtraction of trigger time tag
37	Enable overlapping triggers

[*] Only one of the three trigger modes can be enabled.

45:38	Looking back window
53:46	Looking ahead window
54	Enable substraction of start time measurement (channel 32)
55	Enable matching
56	Enable automatic reject
72:57	Reject offset [15:0] (read back of reject counter)
78:73	Adjust channel [5:0]
86:79	Adjust [7:0]
87	Enable individual adjust
89:88	Test mode: 00 = normal mode 01 = Coarse, Vernier, channel, falling from internal test scan path 10 = Vernier from internal test scan path 11 = Toggle between internal tes scan path and its inverted value
94:90	DLL_current_level_b [4:0], charge pump levvels inverted For normal operation = 11110 bin (minimum current levels)
95	DLL reset
96	Detect falling edge start
97	Detect falling edge odd channels
98	Detect falling edge even channels
99	Detect both edges (all channels)
100	Enable empty start
101	Must be equal to set-up bit 54
102	Enable use of double synchronisers
103	Enable double hit priority queue
104	Enable start gating
120:105	Course time offset[15:0] (read back of course count)
121	Enable token readout
122	Not locked error mask
132	Hit_error error mask
124	Event_buffer_overflow error mask
125	Trigger_buffer_overflow error mask
126	Serial_trigger_error error mask

The programming of channel adjustment constants

The programming of the channel adjustment constants is a bit more complicated than the other programmeble parameters. The required 32 eight bit constants are not direct acceesseble as a individual field of the setup scan path.
The 32 channel specific constants are stored in a memory written into via the setup scan path. The adjustment of the commen channel (start channel) is taken as the last adjustment constant is shifted into the set-up chain (not contained in channel adjustment memory). This approach saves significant silicon area, but requirers the constants to be shifted in one by one. In the case only a commen adjustment is required the adjustment memory can be left non initialized.
Loading of one channel adjustment constants onto the channel adjusment memory is preformed as follows. The channel identifier Ajust_channel[5:0] and its corresponding adjustment constant Adjust[7:0] must be loaded into the set-up scan path. When the set-up dat has been loaded, the JTAG controller will give the signal update, by passing the JTAG state Update. The adjustment will be stored therefor in the memory on the right spot. During the loading of the memory all other bits of the set-up data can be left undefined.
The commen adjustmend is not a part of the memory, but of the set-up scan path.
It must be loaded as last, togather with al the other bits. The Adjust_channel[5:0] = 10 0000 bin.

Preforming the reset of the DLL.

The DLL of the TDC is a rather slow process. Therefor it is not initiated by the reset pin of the chip. So after power-up it is not resetted automaticaly.Due to this, the reset pin may be used freely at regular interfaces to reset the rest of the TDC.
The DLL must be resetted after power-up with 2 JTAG commands. In the first command the reset signal must be one. To get it into action, the JTAG controler must pass the Update state. To start the DLL lock tracking the signal must be made zero in the same way. After this the DLL starts locking, and after obtained the not_locked status bit will be cleared.

Control register.

The JTAG control scan path is used to enable or disable channels of the TDC. This can be done during run time.

Bit#	Description
32:0	Channel enable

Status register.

Bit#	Description
0	Not_Locked
1	Hit_error
2	Event_buffer_overflow
3	Trigger_buffer_overflow
4	Serial_trigger_error

Internal test register.

Used in combination with the test bits for debugging purpose only. The internal scan path gives direct access to the interface between the hit registers and the first level buffer logic.

Bit#	Description
31:0	Vernier in not encoded form
47:32	Coarse[15:0]
53:48	Channel[5:0]
54	Falling edge
55	Selest difference
56	Select start (read only)
57	Store start (read only)
58	Store event (read only)

Chapter 5: Technical specifications.

Power Supply:

Supply	Voltage	Current
VCC	+ 6 V	1.3 A
VEE	- 6 V	0.2 A

Threshold Voltage:

Range: 0 to 255 mV.
Controlled by AJTAG.

Data output:

Data strobe protocol.
Data format see Readout.

Board overview

In figure 8 a global drawing is giver of the board to give an idea about what is on it.

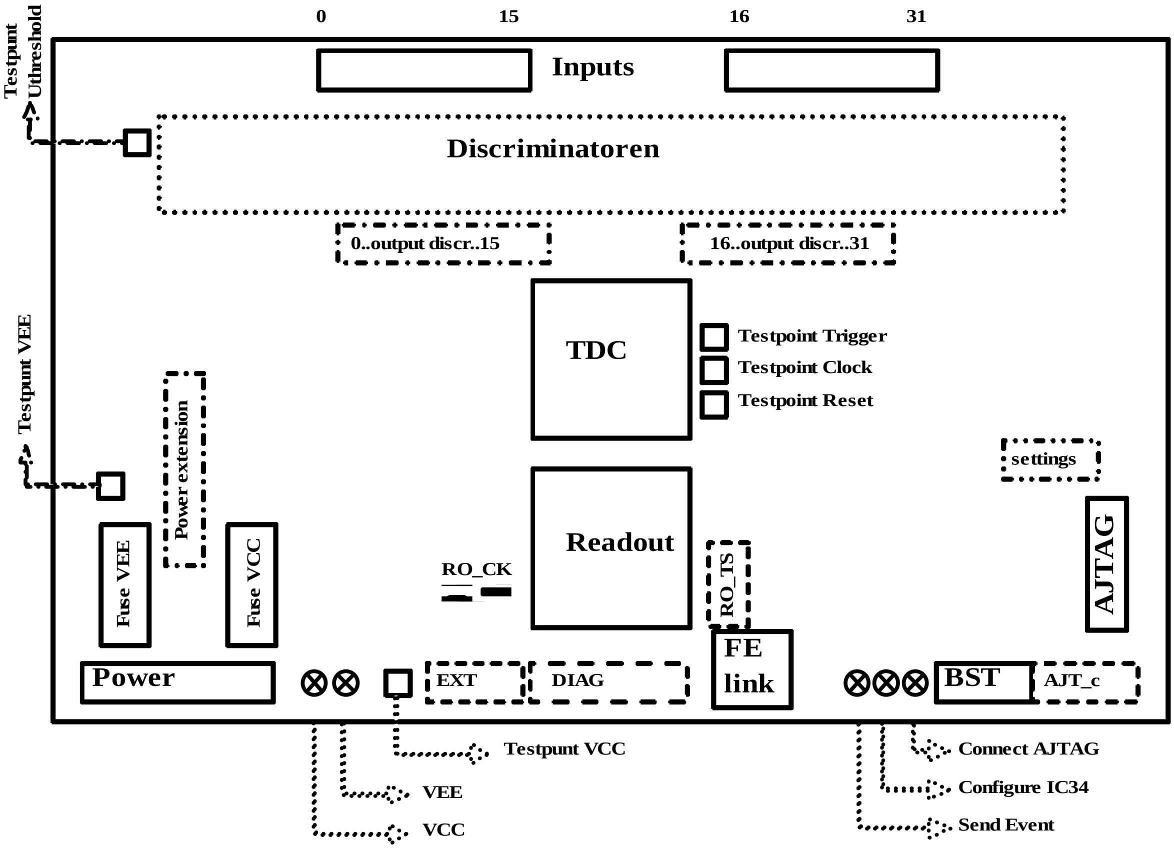


Figure 8: Overview of the PCB.

Connectors:

The board has the following connectors for normal use:

Connector	Signals	Remarks
Inputs	2 34 pole connectors for the 32 bipolar input signals.	For DATCHA mounted on the back for pre-amplifier board, else on top of the board
FE link	In coming Clock and Trigger/Reset, out going Data and Strobe	8 polige FE-link connector For DATCHA 180°, else 90°.
POWER	VCC,GND and VEE	26 pole connector, layout is symetrical. For DATCHA 180°, else 90°
AJTAG, AJT_c	TCK, TMS, TDI and TDO	AJTAG for DATCHA and else AJT_c.

And for the posseble extension card:

Connector	Signals	Remarks
Output discriminators	Output of the discriminators	Can be used as a probe pad for debugging, bipolair output.
Power extension	VCC,GND and VEE	No extension card shortcut bretween the pins 21 and 22 to close the JTAG chain.

For debugging:

Connector	Signals	Remarks
settings	TCK, TMS, TDI and TDO	Direct access
RO_TS	Read-out timing signals	No connector, only for probing.
RO_CK	Read-out clock	No connector, only for probing.
DIAG	Read-out timing signals	No connector, only for probing.
EXT	spare	No connector, only for probing.
BST	Boundery scan for IC34	Optional for reprogramming during debugging.

The first test results.

The test setup:

To test the Datimizer we build a small system, which is capable of generating al signals needed for the test and reading the data from the Datimizer.
The setup system is drawn in figure 8.

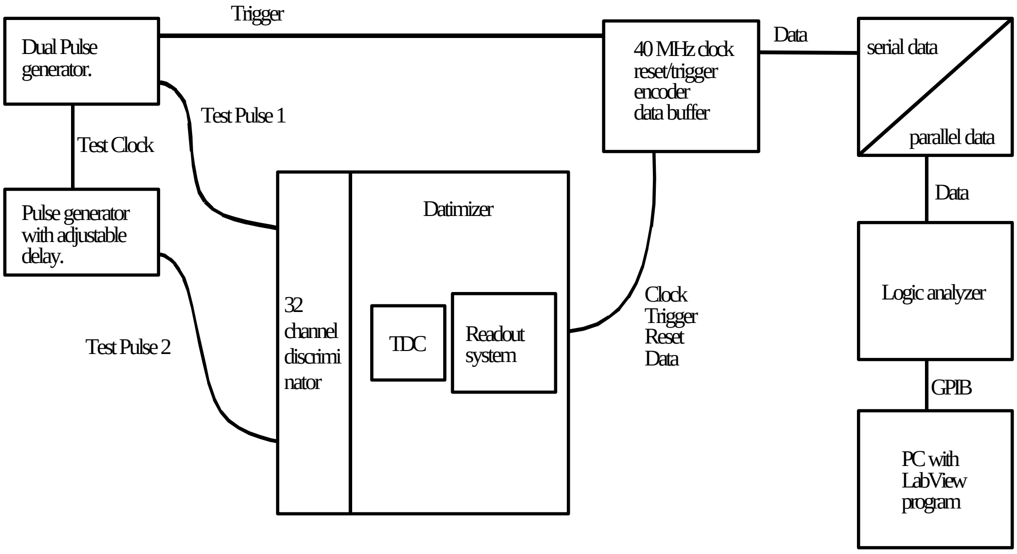


Figure 8: The test set-up.

The test set-up is build around a PC with LabView.

For this measurement the TDC is set-up to calculate the difference between channel time and trigger time. The test pulse generator is not synchronised with the 40 MHz clock. These two facts made it impossible to measure the channel resolution.
The problem was the trigger resolution. In the readout system as well as in the TDC the trigger resolution is 25 nsec. This problem is solved by using a second pulse generator with build in delay option on a second input channel. By calculating the difference between these channels the combined resolution of two channels is measured.
The test sequence is drawn in figure 9.

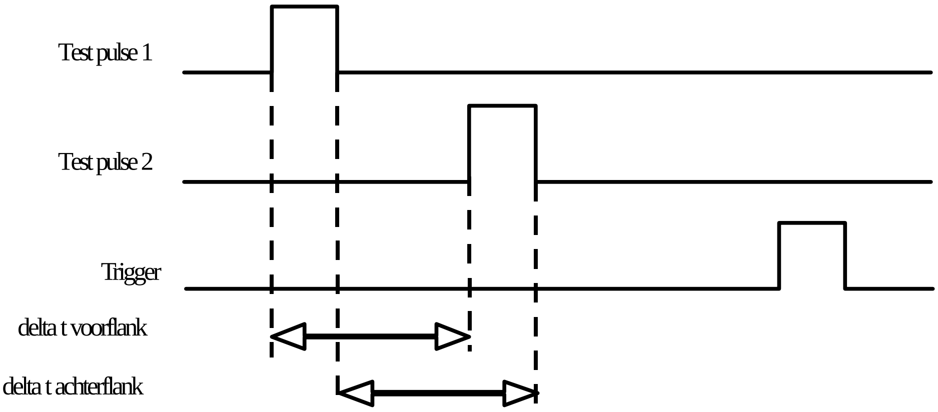


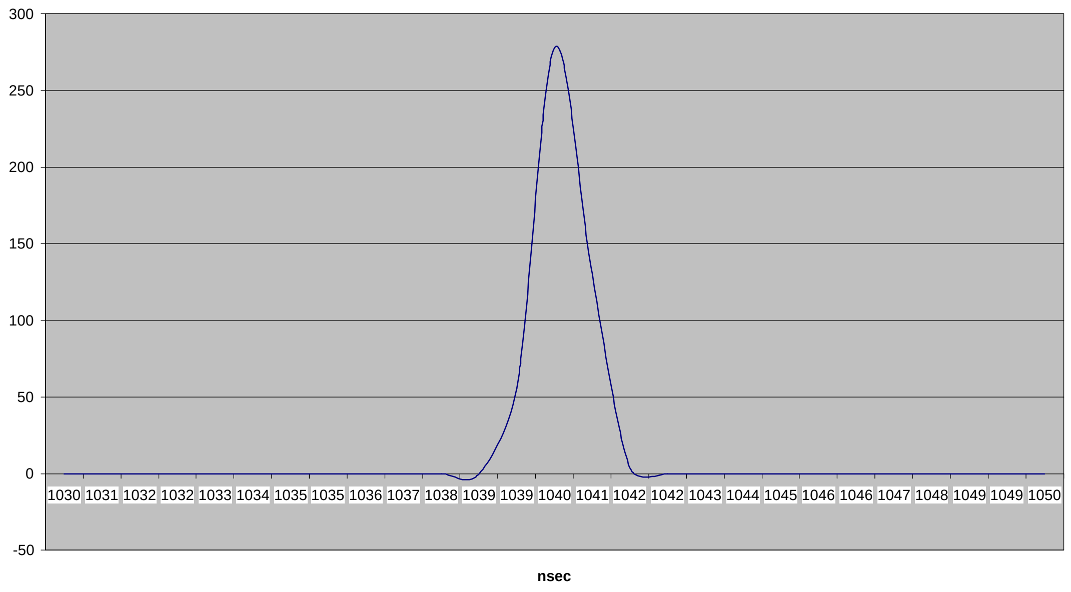
Figure 9: Test pulse sequence.

Test results:

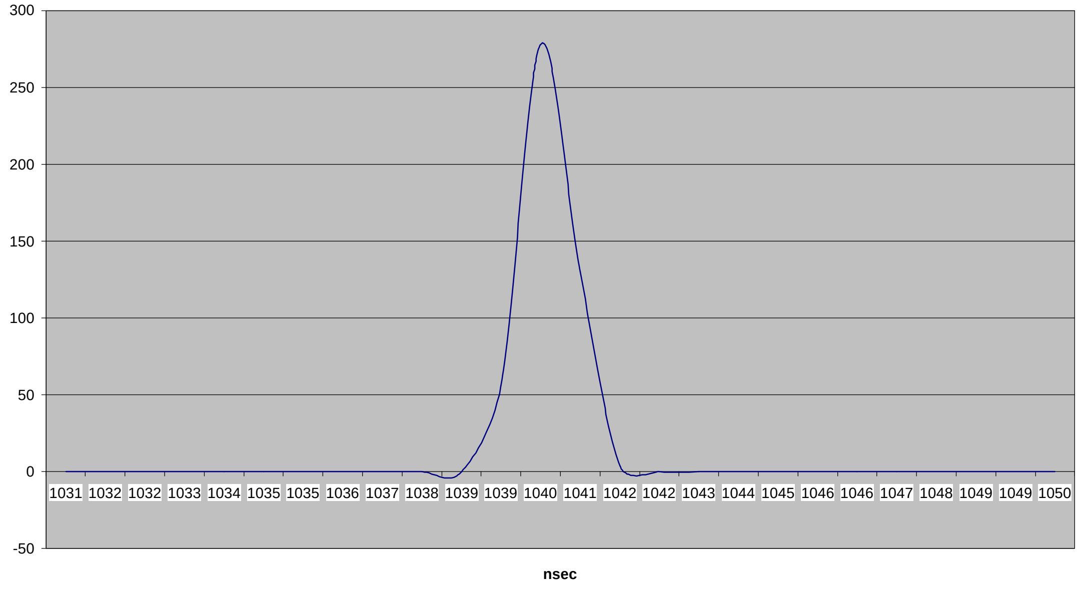
	rising edge	**falling edge**
Datimizer number	521	521
first channel	0	0
second channel	16	16
mean	1039,981	1007,642
standard deviation	0,495	0,415

In the table all information about the Datimizer, channels, mean value and the standard deviation is given.
On the next page two histograms are drawn of these measurements.

Delta t rising edge



Delta t falling edge


nsec