IProbe User's Guide

Release 09.2023

MANUAL

TRACE32 Online Help

TRACE32 Directory

TRACE32 Index

TRACE32 Documents ....

IProbe ....

IProbe User's Guide 1

Introduction 4

IProbe Features 5

Functional Overview 6

Timing Probe Features 7

Input Connector Assignment 8

Analog Probe Features 9

Input Connector Assignments 10

IProbe Input Connector Location 11

Timing Trace Setup and Configuration 12

POD threshold levels and signal display 13

Signal Names 13

General IProbe Functions 19

IProbe Trace Control 19

Basic Trace Control 20

Operation Modes 22

Automatic Trace Control 23

Using the Trigger 23

Trace Display 24

Signal Naming 24

The IProbe.List Command 25

The IProbe.Timing Command 27

Signal Processing 28

The IProbe.GET Command 29

The IProbe.View Command 30

The IProbe.DRAW Command 31

Tracking 33

Search and Compare 35

Real-Time Displays 35

Saving Trace Buffers 36

Relation of the Different Trace Load and Display Areas 37

IProbe data files used with T32-Simulator 37
VHDL and VERILOG Export 38

Simple Trigger 39
Simple Trigger for Timing Mode 39
Trigger Input 41
BusA - Trigger Input
Meaning of the Different Trigger Delay Counter 44
Trigger Out 46
Simple Trigger for Analog Mode 47
Universal Counter Signal Selection 47

Protocol Analysis 49
Timing Mode Restrictions 51

Analog Probe 52
Simple Trigger for Analog Mode 52
Voltage Measurement 52
Current Measurement 53
Power Measurement 55
Energy Analysis 57
Analog Trace Time Coverage Calculation 60

Analog Trace Setup and Configuration 16

Version 10-Oct-2023

Introduction

The IProbe timing analyzer module is a part of the **PowerTrace-II (PT-II) / PowerTrace-III (PT-III)** and the **PowerTrace Serial (PTS)** units. It is a subset of the TRACE32 timing analyzer solutions (PowerProbe and PowerIntegrator). It offers a minimum of timing capture capabilities and is not intended to replace or substitute these units. Just a simple signal qualifier can be used to trigger the analyzer. There is no external clock source and there is no complex trigger capability to qualify input signals. However, IProbe offers transient recording which allows an optimal usage of the available trace memory space.

The memory where the recorded data are stored, must be shared with the memory which normally holds code coverage data if the code coverage analysis feature is enabled. Code coverage and timing analysis can only be used **alternatively**, but not at the same time.

The IProbe offers either a 17 bit digital data trace or a 7 * 12 bit subsequent analog input trace. Depending on if a timing trace or an analog trace is required, either a timing input probe or an analog input probe must be connected to the trace port of the PT-II or PTS unit. If the required probe is attached, IProbe works as a digital or an analog trace module.

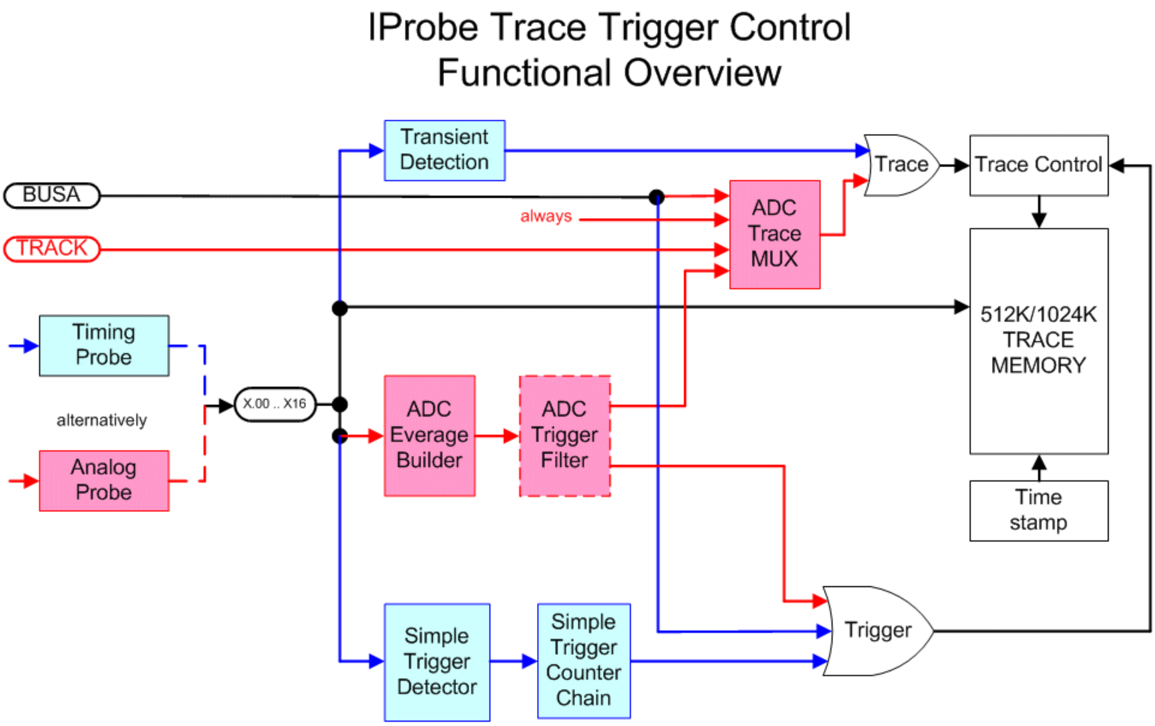
The timing probe basically consists on a high speed buffer for 17 signals, whereas the analog probe is equipped with a 7 channel AD-Converter.

There is a timestamp unit with a resolution of 5ns.

Timing or analog trace date can easily be correlated to the regular program trace.

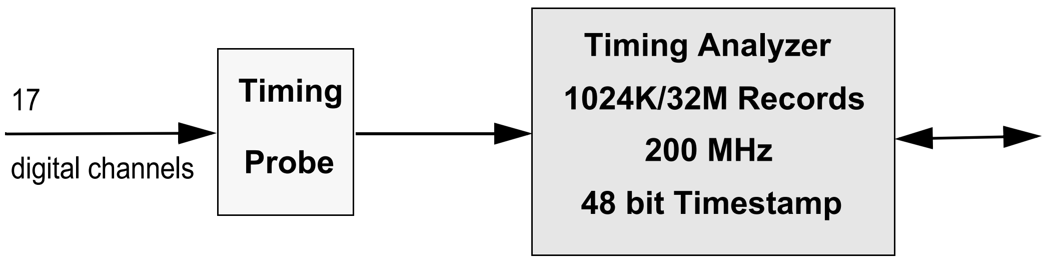
The features of the protocol analysis can also be used.

Trace	**Timing:** The timing analyzer can sample up to 17 channels with 200 MHz sample rate. Consider: Due to restrictions of the used HW, one can just reach 65...70 MHz of clock cycles to trace. Depending on the signal quality, one can get a lower or higher speed limit. **Analog:** Analog signal trace with up to 4 voltage and 3 current channels (subsequently). Alternatively there are up to 3 power channels. Max speed 625 kSPS.
Transient Recording (store data in trace memory)	**Timing:** Data of the input channels are stored in the trace buffer just in case at least one of the inputs alters its level. The total recording time depends on the occurrence of signal changes. If the traced signal changes only once in a ms, the total sampling time will be 1024 seconds for PT-II. The shortest record time is 5 ms for PT-II, which may appear if high-speed clock signals (nearly 200MHz) are recorded. **Analog:** In case of analog trace, trace entries are basically driven by the conversation time of the used Analog-Digital Converter (ADC), the number of used analog channels and the average builder of the ADC-results
Simple Triggering (stop trace)	**Timing:** The simple trigger unit uses a trigger mask, which includes level or edge detection, a trigger filter and trigger counters for generating a trigger event, which finally stops recording. **Analog:** not available



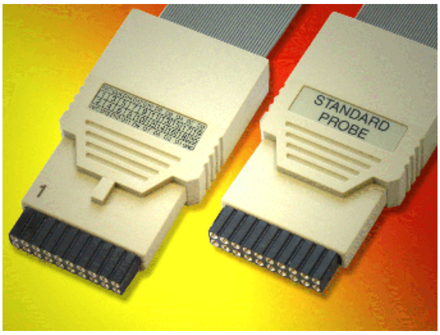
X.00 ... X16	Connector “Logic Analyzer Probe” at the PowerTrace provides 17 input channels for the timing analyzer and analog probe.
Trace Memory	The trace memory stores all data of the input channels.
Timestamp	As the trace memory samples only differences to the previous state, a timestamp and a additional timestamp memory is needed.
Transient Detection	The circuitry detects all changes of the state of the input channels.
Trace Control and ADC Trace Mux	The trace control unit generates control signals for the trace and the timestamp memory, depending on the output of transient detection circuitry and of different source of the analog channels.
Simple Trigger Detector and Coun-ter	The simple trigger system can detect one trigger pattern (combining up to 17 signals), a trigger filter and a trigger counter.
ADC Average Builder	Arithmetic average builder for analog results. 1/1, 2/1, 4/1, 8/1, 16/1, 32/1, 64/1, 128/1 and 256/1.
ADC Trigger Filter	(not yet implemented) Allows trigger if voltage or current is above or below a certain value.

Timing mode	*Timing probe required (Standard probe)!!*
Size	PowerTrace-II: 1024 KRecords PowerTrace Serial: 32768 KRecords
Record	A record consists of 72 bit and contains all trace data, control bits and timestamp.
Inputs	17 digital channels
Clock	200 MHz internal, can not be changed (timing mode only)
Recording (Trace control)	transient only (stores probe pattern just a signal alters the state. Any input can be disabled from transient recognition)
Trigger (stop recording)	Input mask, low, high, rising, falling, external (BUSA)
Trigger counter	Trigger pre-delay, Trigger width, Trigger count, Trigger delay
Trigger Output	BUSA (can be used for Program break and others)
Timestamp	48 bit, 5 ns resolution
Correlation	Program trace, other timing analyzer



D0..D15	Data 0 ... 15	- digital input for 200 MHz
CLK	Data 16	- digital input for 200 MHz (**NOT treated as external Clock** input)
GND	Signal ground	
N/C	Not connected, leave unconnected	

The timing analyzer input probe (Standard probe) has a label which describes the meaning and the position of the input pins. View is front view onto the probe.

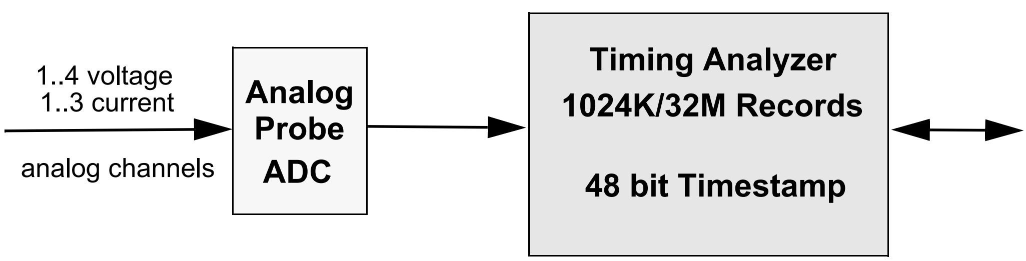


View onto timing probe.

1	3	5	7	9	11	13	15	17	19
NC	D16	D14	D12	D11	D08	D06	D04	D02	D00
NC	D15	D13	D11	D09	D07	D05	D03	D01	GND
2	4	6	8	10	12	14	16	18	20

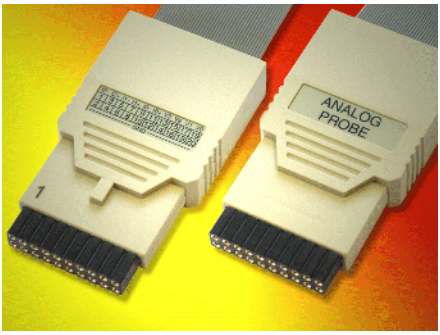
Analog Probe Features

Analog mode	*Analog probe required!!*
Size	PowerTrace-II: 1024 KRecords PowerTrace Serial: 32768 KRecords
Trace Record	A record consists of 72 bit and contains ADC value of a single channel, control bits and timestamp.
Analog Record	An analog record consists of several trace records. It contains all ADC values of all active channels, but at least one value of each channel
Analog (ADC) resolution	12 bit
Conversation time	625 kHz / number of use channels.
Input characteristic	Voltage channels : 1 MOhm, 0...5V, DC only Current channels : Umax 28Vcom.mode , Udiff+- 0.125V, 1 MOhm,DC only
Voltage channels	4 (1 at a time, shared with current channels)
Current channels	3 (1 at a time, shared with voltage channels, Shunt required)
Power measurement	automatic power calculation
Arithmetic average builder	1/1, 2/1, 4/1, 8/1, 16/1, 32/1, 64/1, 128/1, 256/1
Trace control	self-contained, program trace entry, external event (BUSA), Result filter (not yet implemented)
Trigger (stop recording)	automatic (AUTOARM), manual, Stack mode
Scan mode	managed by probe
A/D result store	SRAM, shared with code coverage (just alternatively)
Profiling	Voltage, current and power or energy
Timestamp	48 bit, 5 ns resolution
Correlation	Program trace, other Timing analyzer



V0 .. V3	4 channel analog voltage input
I0+ .. I2+	3 channel current sense input (positive port, shunt required)
I0- .. I2-	3 channel current sense input (negative port, shunt required)
GND	Analog ground

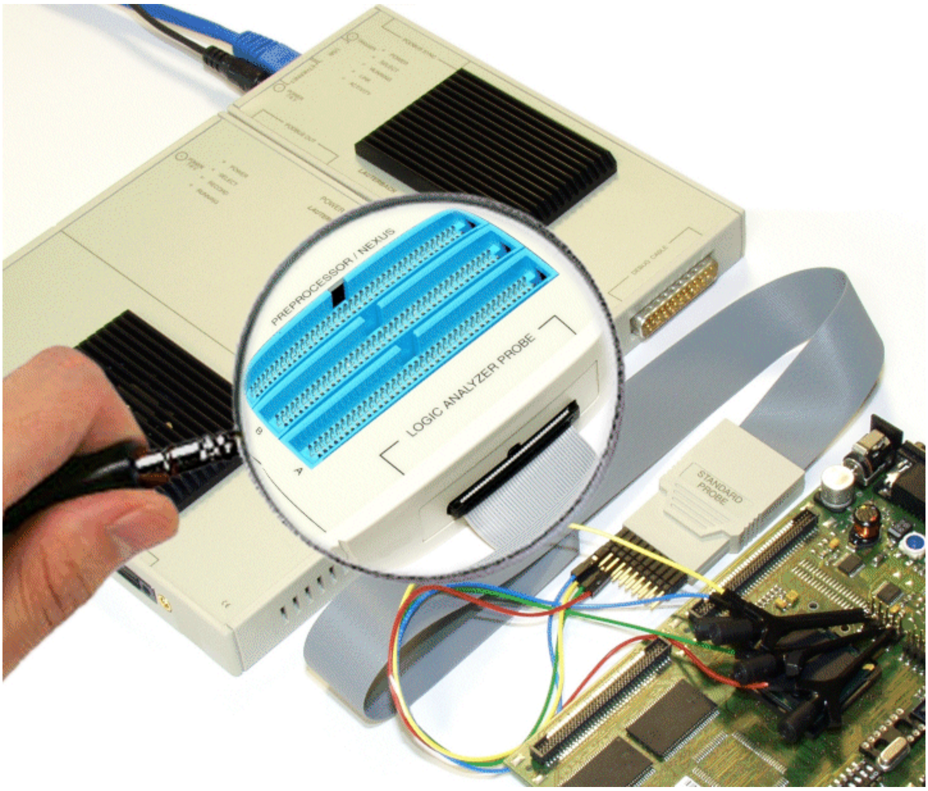
The analog input probe has a label which describes the meaning of the input pins. View is front view onto the probe.



View onto analog probe

1	3	5	7	9	11	13	15	17	19
I2-	I2+	I1-	I1+	I0-	I0+	V3	V2	V1	V0
GND	GND	GND	GND	GND	GND	GND	GND	GND	GND
2	4	6	8	10	12	14	16	18	20

IProbe Input Connector Location

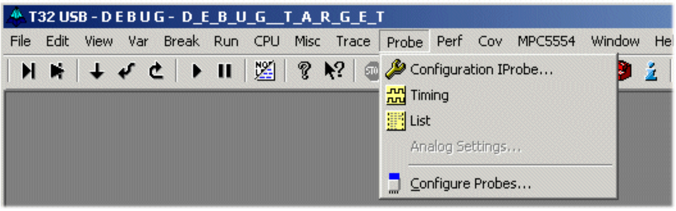


The location of the connector for the different trace probes can be found at the narrow side of the PowerTrace-II unit, close to the 3 blue trace/debug probe connectors. Take care that the probe connector polarisation fits the keyway of the PT-II box.

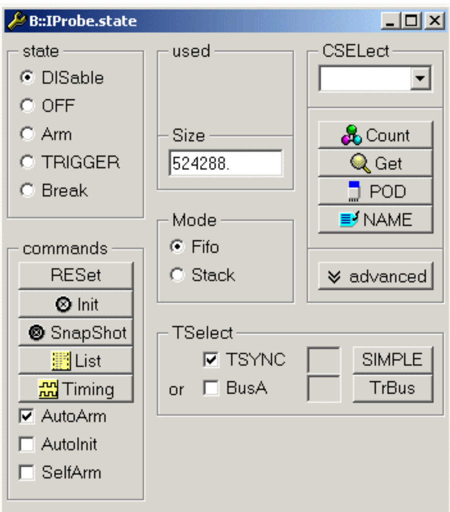
Timing Trace Setup and Configuration

The setup windows of the IProbe can either be found on top of the screen in the pull down menu area or by entering **IProbe** in the command line

If a ***timing probe*** *(Standard Probe)* is connected, a pull-down menu appears if **Probe** has been chosen.



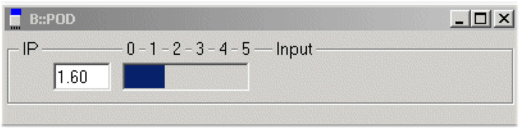
The IProbe state window appears on the screen if **Configuration IProbe** is selected.



The **advanced** button increases the **IProbe.state** window and offers additional settings for triggering.

POD threshold levels and signal display

The configuration window, called by **Configure Probes** from pull down menu, by command **POD** or by the **POD** button in IProbe.state window, allows threshold setup commonly for all digital input channels between 0.5 and 5V.

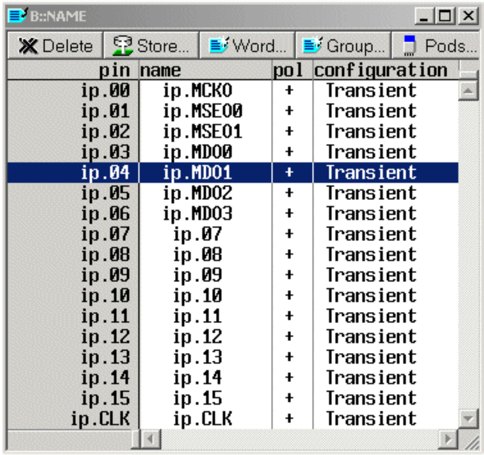


IPOD.state	Display threshold level
POD.Level	Select threshold level
POD.RESet	Set to default

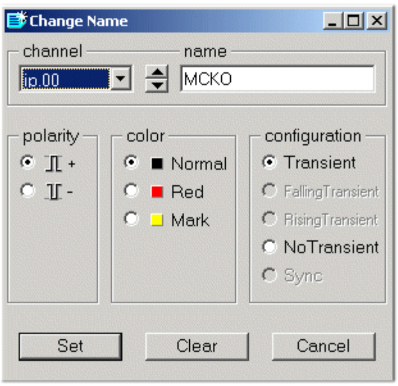
Signal Names

The **NAME** function generates logical names for input channels and additionally the polarity of the signals. For Trigger selection as well as for parameters for the various display functions, either logical definitions or physical pin names can be used.

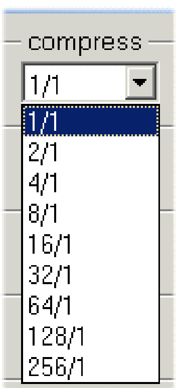
NAME.list	Display logical names
NAME.RESet	Erase logical names for input pins
NAME.Set	Define logical names for input pins
NAME.Group	Define logical names for input groups
NAME.Word	Define logical names for busses
NAME.Delete	Erase logical groups or words for input pins



A specific window allows changing parameters of a certain input. (just click to a signal name). This window allows renaming of signals as well as changing the color of the signal in the timing display window or deselecting the signal from transient detection.

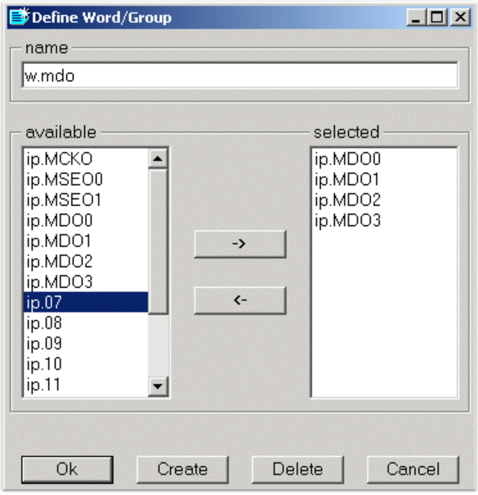


Each voltage and each current channel can be compressed to get a more smooth result. Compression is a kind of arithmetic average. Averaging will be done by hardware. There are averages between 1/1 till 256/1.



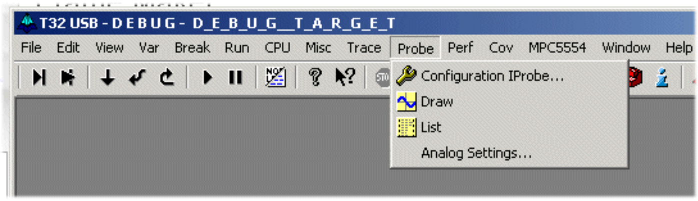
Higher compression causes longer time between two recorded ADC results.

Signals can be arranged in groups for better readability. They can also be use in display windows to display e.g. Data, Address in a decoded form

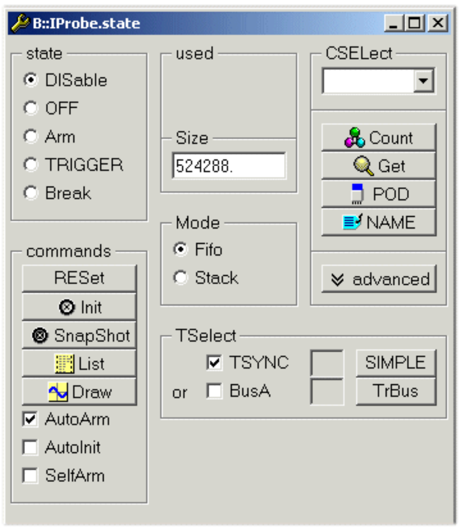


Analog Trace Setup and Configuration

If an ***analog probe*** is connected, the configuration window can be found on top of the screen. A pull-down menu appears if **Probe** has been chosen.

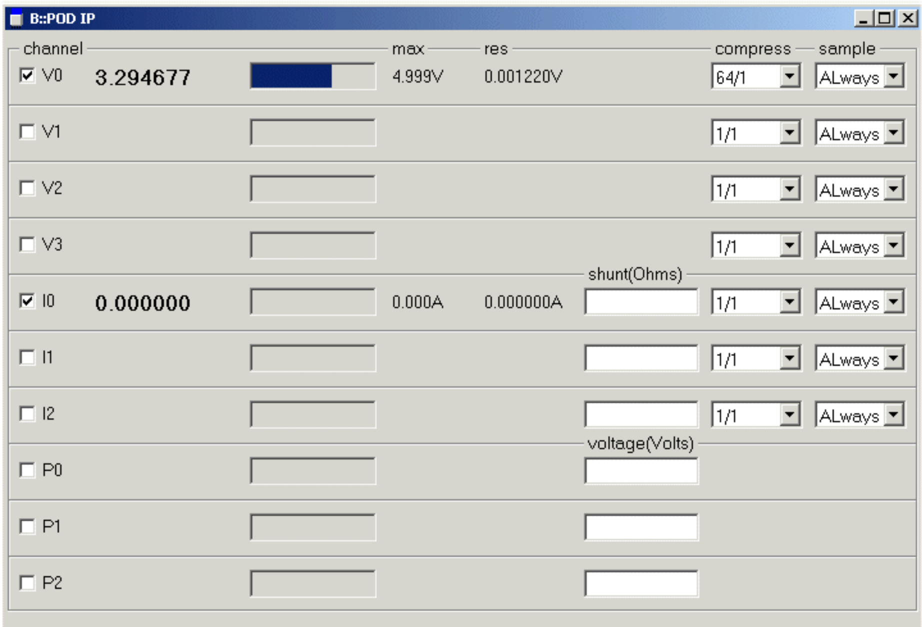


The IProbe state window appears on the screen if **Configuration IProbe** is selected.



Most IProbe commands are valid the same way as described in Timing mode. Below there is a description of the analog trace specific commands only. For all others, refer to the description above.

The configuration window, called by **Analog Settings**, allows different settings and selections for each analog channel.

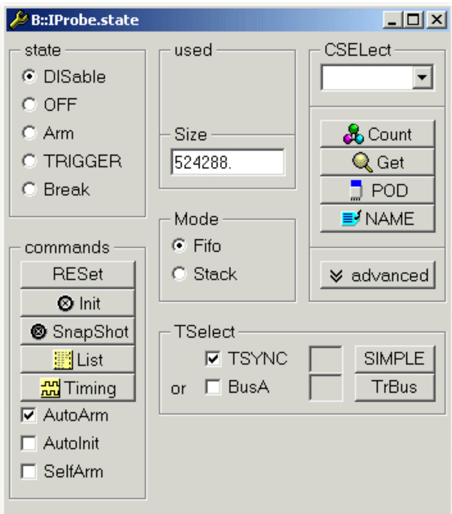


The ADC never stops conversion of the enabled channels. If there is no condition to enable an entry for the trace memory, the result of the conversion and compression will be discarded. The user can select a kind of enabler which causes an entry of a conversion result in the trace memory. This can be done by selection one of the four options in the **sample** section.

- **Always** : Causes an entry in trace every time conversion and compression is ready. (Self-contained recording)
- **Track** : Causes an entry controlled by the program trace unit. Any time there is an entry in the program trace memory, there is also an entry in the IProbe trace memory. Because the analog recording takes much more time for an entry, normally more entries of the program trace unit belong to a single analog trace entry.
- **BusA** : The PodBus trigger signal (BusA) is the source for IProbe trace entry control. This signal can be driven by a varies number of sources.
- **Filter** : (not implemented yet!) Forces an entry in the IProbe trace memory if a certain level of voltage or current has been crossed

IProbe Trace Control

Independent on if the IProbe is used as a timing or analog analyzer, trace control is done by using the **IProbe.state** window. However this window dynamically changes slightly its contents depending on if a digital or an analog probe is connected.



To set up the default setting use:

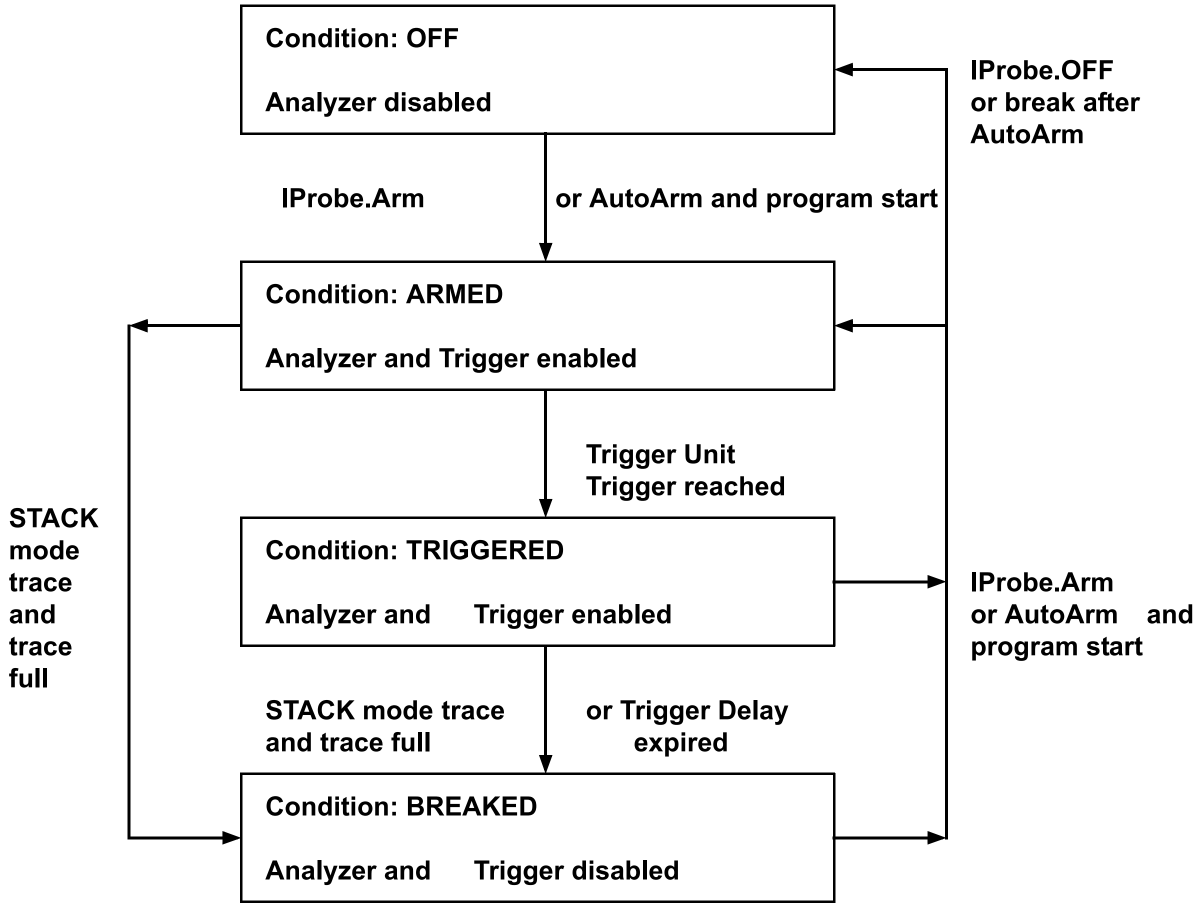
IProbe.RESet	Initialize IProbe, set up default settings

The window displays information about the actual state, the mode and the number of records in the trace buffer. It also shows information about the trigger state and trigger counters

IProbe.state	Show the IProbe configuration and state window
IProbe.DISable	Global disable for IProbe, prevents any operation
IProbe.OFF	Turn off IProbe recording
IProbe.Arm	Turn on IProbe recording
IProbe.Init	Clear the trace buffer and restart the trigger and counters.
IProbe.SnapShot	Init and ARM the IProbe, take date and display it after trace is full (Stack mode required)
IProbe.RESet	Restore all setting to the default values

For the case the IProbe is used independent on a debugger or without relation to a user program, recording of data must then be manually controlled exclusively via the **IProbe.state** window (**ARM** and **OFF**).

The trace buffer can either sample channels or display the results. In **Arm** state the input channels can be sampled. The trace contents can just be displayed in the **Off** or **Break** state.



Analyzer Operation States

TRIGGER	The analyzer is waiting for the expiration of the trigger delay.
Break	The trigger unit has stopped the recording
used	Displays the used records in the trace buffer
size	Selected Trace memory size. Can manually be modified.

Operation Modes

The behavior characteristics of the analyzer can be changed by the **IProbe.Mode** command. The basic operation mode for the trace storage can be FIFO or STACK.

IProbe.Mode Fifo	FIFO operation mode, analyzer records the last cycles before stop recording. The oldest trace entries will be overwritten.
IProbe.Mode Stack	In STACK operation mode, the analyzer stops recording, when the trace buffer is full.

Automatic Trace Control

To simplify controlling of the analyzer, different automatic control options are available. As a default, the **AutoArm** option is active. This means that the analyzer will be armed automatically when the user program is started and it will be switches to off, after stopping the real-time emulation.

IProbe.AutoArm	Arming the analyzer before any user program start, switch off after program stop/break
IProbe.AutoInit	Automatically initialization of the analyzer before AutoArm
IProbe.SelfArm	Self-contained ARM of the IProbe analyzer after OFF

The combination of **SelfArm** and **Stack** operation:

```
IProbe.Mode Stack

IProbe.SelfArm

IProbe.AutoInit

IProbe.Timing
```

The result will be a continuously updated timing list window, which shows the last sampled signals.

Using the Trigger

Refer to the **Simple Trigger** section for detailed information about trigger features.

Trace Display

Depending on the IProbe mode, there are common and individual trace display instructions.

	Timing	Analog	
IProbe.List	x	x	Displays trace in table format
IProbe.Timing	x		Displays channels as waveform graphics
IProbe.Get	x	(x)	Displays the input signal level and activity
IProbe.View	x	x	Displays just a single trace record
IProbe.DRAW		x	Display analog trace graphically

Signal Naming

The analyzer list window can display the contents of the trace memory in several formats. The displayed columns and information can be configured very flexible. By **NAME.Word** it is possible to sum-up several channels to one identifier, by **NAME.Set** a specific name for a given channel is used. The NAME commands are normally usual in timing mode.

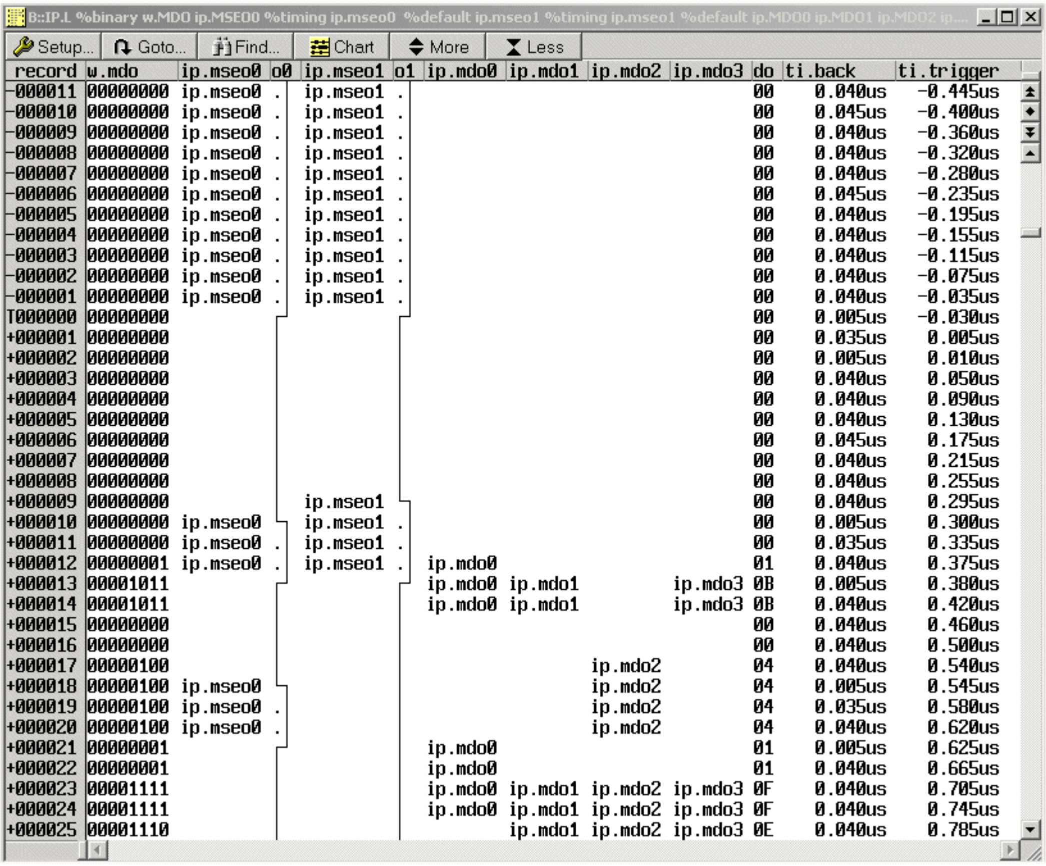
```
NAME.Set IP.01 mcko                     ; assigning suitable identifiers
NAME.Set IP.02 mseo0                    ; for each channel
NAME.Set IP.03 mseo1
NAME.Set IP.04 mdo0
NAME.Set IP.05 mdo1
NAME.Set IP.06 mdo2
NAME.Set IP.07 mdo3

; assigning word identifier "MDO" to channels 4..7
NAME.Word MDO ip.mdo4 ip.mdo5 ip.mdo6 ip.mdo7

; Display the channels of the IProbe trace memory
IProbe.List %BINary W.mdo IP.mseo0 %Timing IP.mseo0 %DEFault IP.mseo1\
%Timing IP.mseo1 %DEFault IP.mdo0 IP.mdo1 IP.mdo2 IP.mdo3 TIme.Back\
TIme.Trigger
```

To get a display of the recorded data in table form, **IProbe.List** can be used.

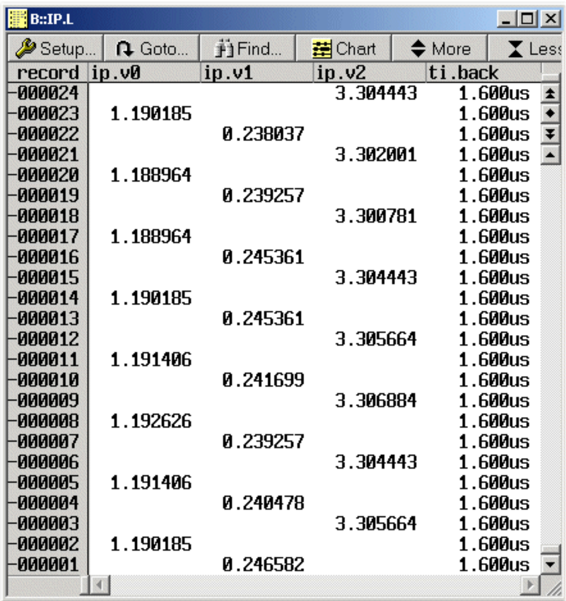
IProbe.List in timing mode.



record	w.mdo	ip.mseo0	o0	ip.mseo1	o1	ip.mdo0	ip.mdo1	ip.mdo2	ip.mdo3	do	ti.back	ti.trigger
-000011	00000000	ip.mseo0	.	ip.mseo1	.					00	0.040us	-0.445us
-000010	00000000	ip.mseo0	.	ip.mseo1	.					00	0.045us	-0.400us
-000009	00000000	ip.mseo0	.	ip.mseo1	.					00	0.040us	-0.360us
-000008	00000000	ip.mseo0	.	ip.mseo1	.					00	0.040us	-0.320us
-000007	00000000	ip.mseo0	.	ip.mseo1	.					00	0.040us	-0.280us
-000006	00000000	ip.mseo0	.	ip.mseo1	.					00	0.045us	-0.235us
-000005	00000000	ip.mseo0	.	ip.mseo1	.					00	0.040us	-0.195us
-000004	00000000	ip.mseo0	.	ip.mseo1	.					00	0.040us	-0.155us
-000003	00000000	ip.mseo0	.	ip.mseo1	.					00	0.040us	-0.115us
-000002	00000000	ip.mseo0	.	ip.mseo1	.					00	0.040us	-0.075us
-000001	00000000	ip.mseo0	.	ip.mseo1	.					00	0.040us	-0.035us
T000000	00000000									00	0.005us	-0.030us
+000001	00000000									00	0.035us	0.005us
+000002	00000000									00	0.005us	0.010us
+000003	00000000									00	0.040us	0.050us
+000004	00000000									00	0.040us	0.090us
+000005	00000000									00	0.040us	0.130us
+000006	00000000									00	0.045us	0.175us
+000007	00000000									00	0.040us	0.215us
+000008	00000000									00	0.040us	0.255us
+000009	00000000			ip.mseo1						00	0.040us	0.295us
+000010	00000000	ip.mseo0		ip.mseo1	.					00	0.005us	0.300us
+000011	00000000	ip.mseo0	.	ip.mseo1	.					00	0.035us	0.335us
+000012	00000001	ip.mseo0	.	ip.mseo1	.	ip.mdo0				01	0.040us	0.375us
+000013	00001011					ip.mdo0	ip.mdo1		ip.mdo3	0B	0.005us	0.380us
+000014	00001011					ip.mdo0	ip.mdo1		ip.mdo3	0B	0.040us	0.420us
+000015	00000000									00	0.040us	0.460us
+000016	00000000									00	0.040us	0.500us
+000017	00000100							ip.mdo2		04	0.040us	0.540us
+000018	00000100	ip.mseo0						ip.mdo2		04	0.005us	0.545us
+000019	00000100	ip.mseo0	.					ip.mdo2		04	0.035us	0.580us
+000020	00000100	ip.mseo0	.					ip.mdo2		04	0.040us	0.620us
+000021	00000001					ip.mdo0				01	0.005us	0.625us
+000022	00000001					ip.mdo0				01	0.040us	0.665us
+000023	00001111					ip.mdo0	ip.mdo1	ip.mdo2	ip.mdo3	0F	0.040us	0.705us
+000024	00001111					ip.mdo0	ip.mdo1	ip.mdo2	ip.mdo3	0F	0.040us	0.745us
+000025	00001110						ip.mdo1	ip.mdo2	ip.mdo3	0E	0.040us	0.785us

Each entry is classified by a record number. Usually the most recent entry is -1. If a trigger point is available, it is marked with a "T" and its record number is 0. After trigger event, record number are treated positive.

Probe.List in analog mode.



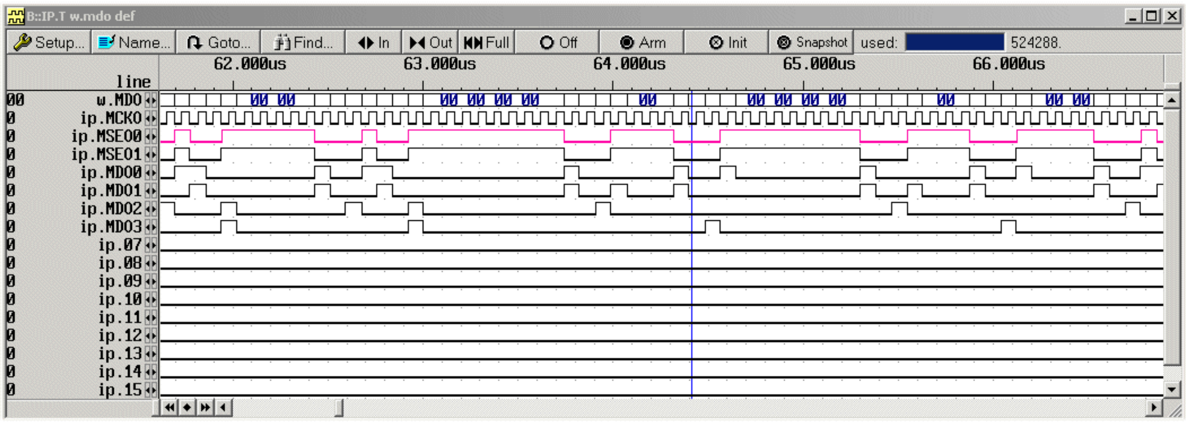
record	ip.v0	ip.v1	ip.v2	ti.back
-000024			3.304443	1.600us
-000023	1.190185			1.600us
-000022		0.238037		1.600us
-000021			3.302001	1.600us
-000020	1.188964			1.600us
-000019		0.239257		1.600us
-000018			3.300781	1.600us
-000017	1.188964			1.600us
-000016		0.245361		1.600us
-000015			3.304443	1.600us
-000014	1.190185			1.600us
-000013		0.245361		1.600us
-000012			3.305664	1.600us
-000011	1.191406			1.600us
-000010		0.241699		1.600us
-000009			3.306884	1.600us
-000008	1.192626			1.600us
-000007		0.239257		1.600us
-000006			3.304443	1.600us
-000005	1.191406			1.600us
-000004		0.240478		1.600us
-000003			3.305664	1.600us
-000002	1.190185			1.600us
-000001		0.246582		1.600us

The most used display command is the timing diagram:

```
; display content of probe trace memory as timing diagram
IProbe.Timing W.mdo DEFault
```

Timing displays can be zoomed. The left mouse bottom set the cursor. By pressing the mouse, a new zoom window can be selected. Fast zooming and de-zooming can be done by scrolling with the mouse wheel.

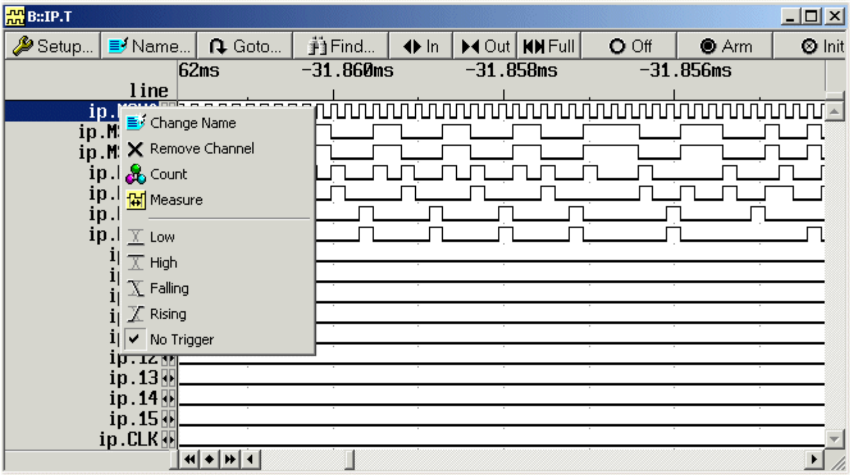
The **DEF**ault keyword selects a default set of display information.



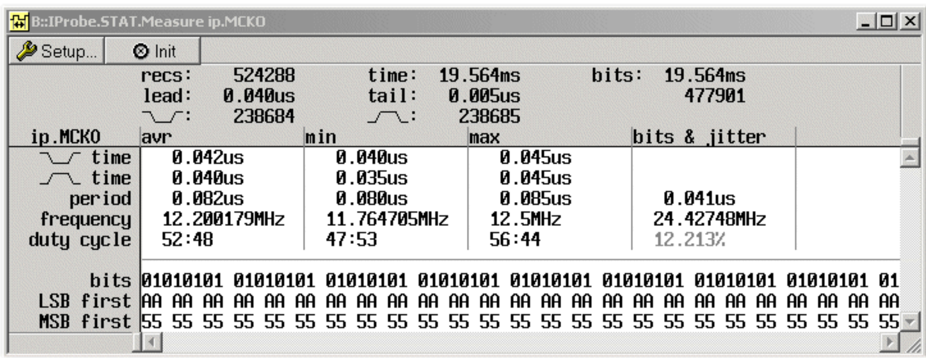
The order of the displayed signals is either given by default or it can be chosen by the user. In this case they will be displayed in the order they are written in the IProbe list or timing command. Refer also to the soft keys below the command line.

For zooming and scrolling there are a lot of instruction available on top of the window. These options allow magnification and resizing of the display area

Signals can be processed automatically by the measure option. To get this option, use right mouse click to the specific signal name and chose **Measure**.



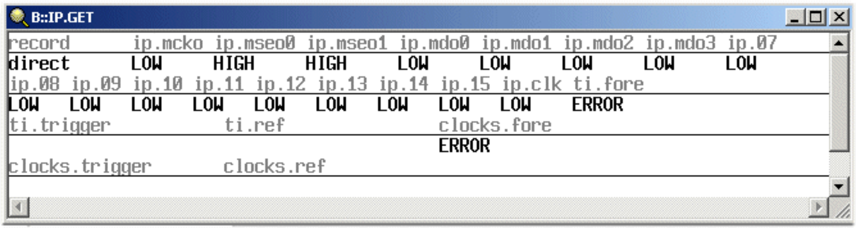
The results are different statistical information about the specific signal flow.



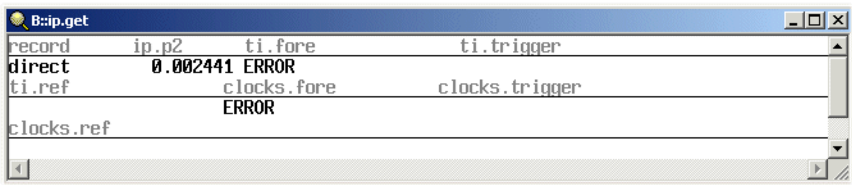
The IProbe.GET Command

The **`IProbe.Get`** command displays the actual input state and activity of the timing input channels.

HIGH	Signal stays high
LOW	Signal stays low
HILO	Signal is toggling



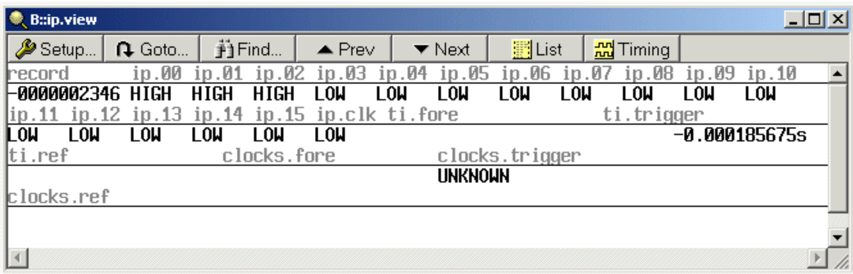
In case of analog mode, the current power and voltage of the activated channels will be displayed



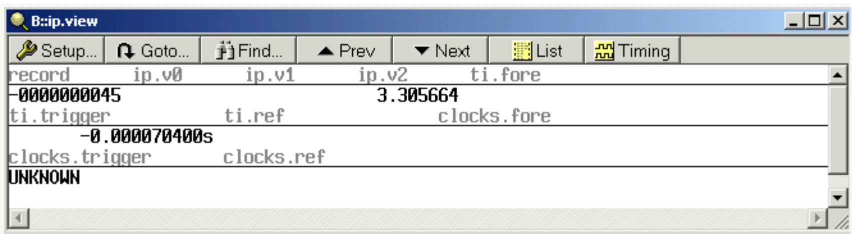
The IProbe.View Command

The **`IProbe.View`** command displays a single record of the trace memory.

Timing mode:

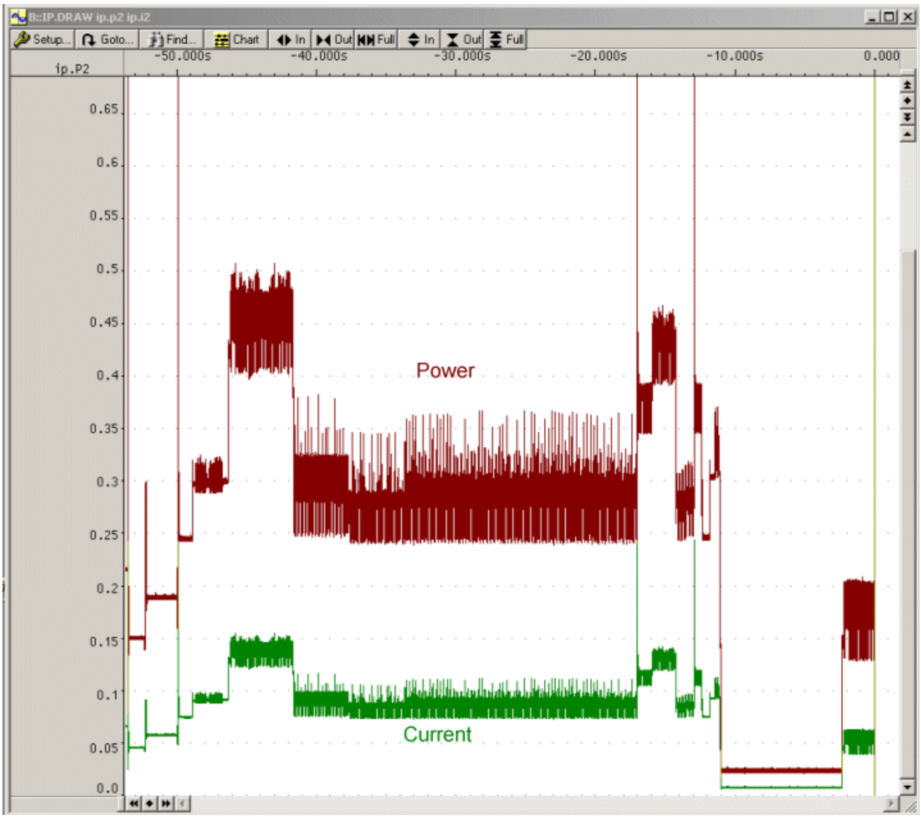


Analog Mode:

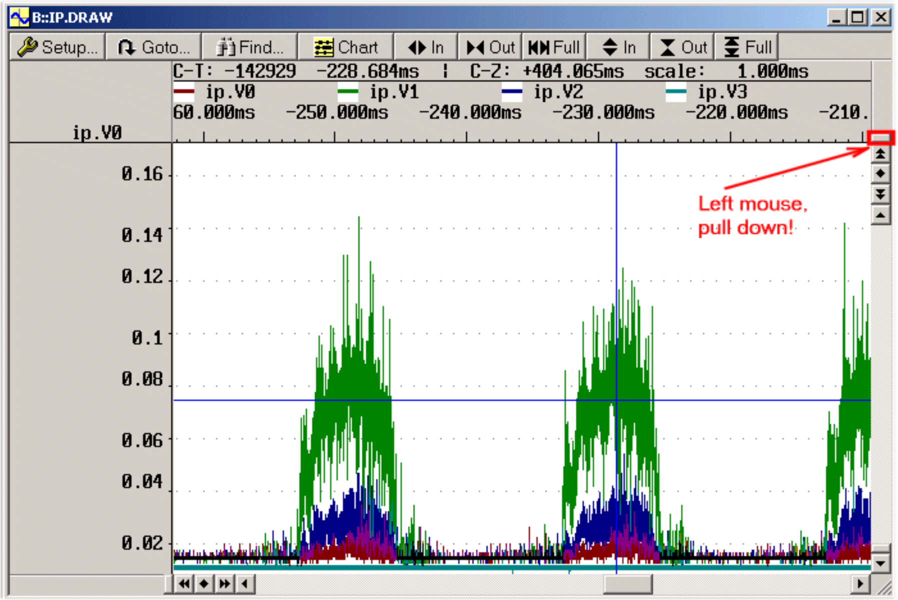


The IProbe.DRAW Command

The DRAW command displays analog trace data in a graphic waveform. Each voltage, current or power channel can be displayed separately or together in a single window.



A hidden way to display the correlation of colors and channels, is to pull down the marked area with the left mouse button in the window shown below.

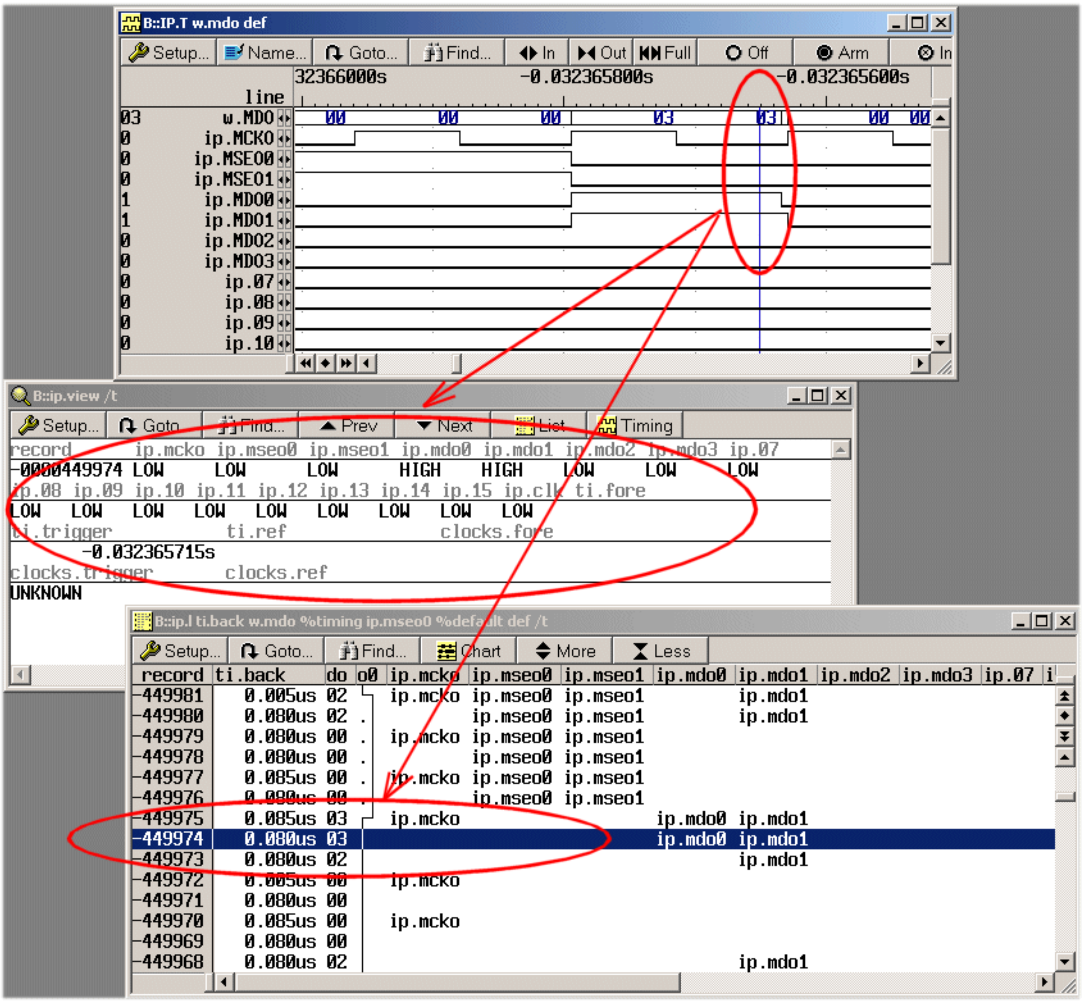


Tracking

The tracking option forces all IProbe analyzer windows and even program trace window of other trace units in the debug system which are in tracking mode (option **/Track**), to pan to the same position like the reference window. The reference window can be any analyzer display window from the state analyzer or the timing analyzer. The reference point is fixed to the absolute time. Every analyzer system has an independent, but correlated, timestamp unit. The tracking function can be used also for displaying port analyzer windows with different zoom rates. Tracking can also be done subsequently by drag-and-drop.

```
IProbe.View  /Track            ; define tracking window for port analyzer
IProbe.List /Track
IProbe.Timing W.mdo DEFault
```

The **IProbe.View** shows only one frame of the trace storage:



Analog drawings can also be tracked or can be master for tracking.

ZoomTrack

A very helpful options during tracking is the **ZoomTrack** (**/ZT**) option. This option causes not only chaining different windows together regarding time, it also keeps the zoom factor of the different windows in common. With other words, if the zoom factor of the master window will be modified, all other windows will change their zoom factor as well.

This option can be used in timing mode and in analog mode.

Search and Compare

Several commands allow to search for specific events, or compare the trace against a reference:

IProbe.GOTO	Track the display window to a new record
IProbe.Find	Search for records matching the pattern
IProbe.ComPare	Compare two traces or the trace buffer against a file
IProbe.REF	Set reference record for timing measurements

The **Find** command allows to search for the occurrence of a data pattern:

```
IProbe.Find , data w.test:0x55          ; search for matching data on
                                        ; probe A is 55h

IProbe.Find , x.puls1 on at -1.         ; search for rising edge of
x.nmi off                               ; the puls1 signal

IProbe.Find , w.dat  0x5--0x44 or       ; search for different data bytes
w.dat 0xff

IProbe.Find                             ; search for next occurrence
```

The **ComPare** command can compare the current trace against a reference trace saved on disk:

```
IProbe.load x1

IProbe.l /track                 ; display current trace

IProbe.l /track /file           ; display reference trace

IProbe.cp , , ip.0 ip.0         ; compare two lines against
/file                           ; file (complete trace memory)

ip.cp                           ; compare next entries
```

Real-Time Displays

The information recorded by the analyzer can be displayed, while the analyzer is sampling information.

IProbe.SelfArm	Arm the analyzer after all windows have been updated

The command **AutoTEST** can be used to make random samples and show the results continuously.

Saving Trace Buffers

The contents of the trace buffer can be saved on disk and recalled later. The recalled trace buffer can be accessed by all regular analyzer commands by adding the option **/FILE**.

IProbe.SAVE	Save the contents of the trace buffer
IProbe.EXPORT	Exports trace data to a VHDL or VERILOG file
IProbe.LOAD	Load a saved trace buffer as a reference in virtual memory 1
IProbe.FILE	Load a saved trace buffer as a reference in virtual memory 2

Saving a part of the trace buffer can be done by the following command:

```
ip.save test (-1000.)--0x0
```

The trace can be recalled and viewed again by the **File** command:

```
ip.file test

ip.l /file
```

Comparing the file against a new record is possible with the **ComPare** command:

```
ip.load test

ip.cp (-1000.)--0x0 -1000. C0 C1 /file
```

Even two loaded trace files can be displayed and processed at the same time.

```
ip.load test1

ip.t

ip.file test2

ip.t /file
```

Relation of the Different Trace Load and Display Areas

There are 3 areas which can contain trace data and which can be displayed and processed:

	TRACE MEMORY	Virtual Trace Memory 1	Virtual Trace Memory 2
Filled by:	**ip.ARM**	**ip.LOAD filename**	**ip.FILE filename**

IP.LOAD without a file name clears virtual memory 1.

IP.FILE without a file name clears virtual memory 2.

The 3 areas can be displayed in following constellations:

Display :	no *ip. load*	**ip.t**	TRACE MEMORY
	ip.load file1	**ip.t**	Virtual Trace Memory 1
	ip.file file2	**ip.t /File**	Virtual Trace Memory 2

From display window point of view, **ip.load** overwrites (temporarily) TRACE MEMORY display.

The source of the trace data will be displayed on the bottom of the window in the left corner.

IProbe data files used with T32-Simulator

After starting the T32-Simulator, IProbe commands are not allowed, because it is still unknown which type of analyzer and corresponding commands should be offered.

To be able to display IProbe data files stored by **ip.save** with the T32-Simulator, just load the file by either:

T.LOAD *<file>* or **T.FILE** *<file>*

After that, the Simulator recognizes IProbe file and allows to use IProbe commands for display. All other commands described in the previous section are also allowed, including PROTO options.

Exporting signals of the IProbe trace buffer to a **VHDL** or **VERILOG** file can be done by the following command:

```
IProbe.EXPORT test.vhdl /VHDL
```

```
              library ieee, std;
use std.textio.all;
use ieee.std_logic_1164.all;

entity testbench is
port(
IP_MCKO : out std_logic;
IP_MSEO0 : out std_logic;
IP_MSEO1 : out std_logic;
IP_MDO0 : out std_logic;
IP_MDO1 : out std_logic;
IP_MDO2 : out std_logic;
IP_MDO3 : out std_logic;
IP_07 : out std_logic;
IP_08 : out std_logic;
IP_09 : out std_logic;
IP_10 : out std_logic;
IP_11 : out std_logic;
IP_12 : out std_logic;
IP_13 : out std_logic;
IP_14 : out std_logic;
IP_15 : out std_logic;
IP_CLK : out std_logic
    );
end testbench;

architecture test of testbench is
begin

process
begin
IP_MCKO <= '0';
IP_MSEO0 <= '0';
IP_MSEO1 <= '0';
IP_MDO0 <= '0';
IP_MDO1 <= '0';
IP_MDO2 <= '0';

                 *************
```

Simple Trigger

Simple Trigger for Timing Mode

Simple Trigger functionality will be offered in the **IProbe.state** window. These trigger options allow stop of trace recording by certain events.

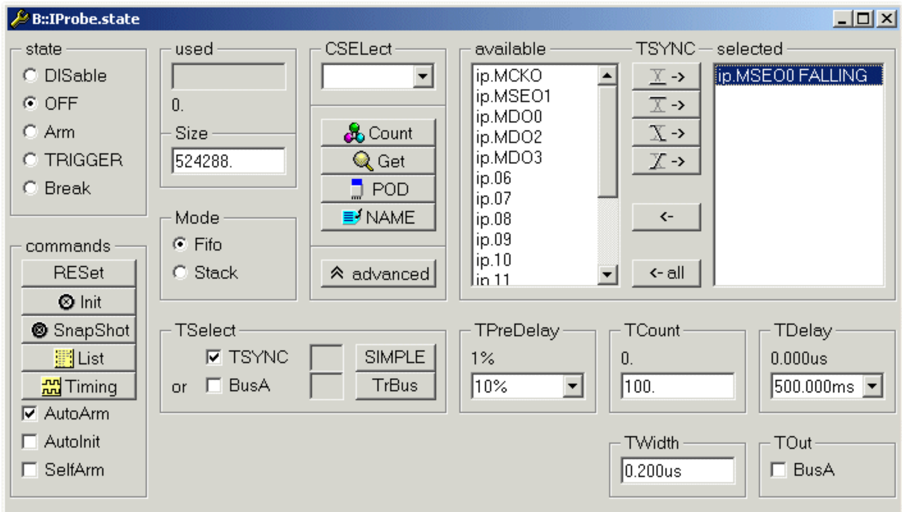
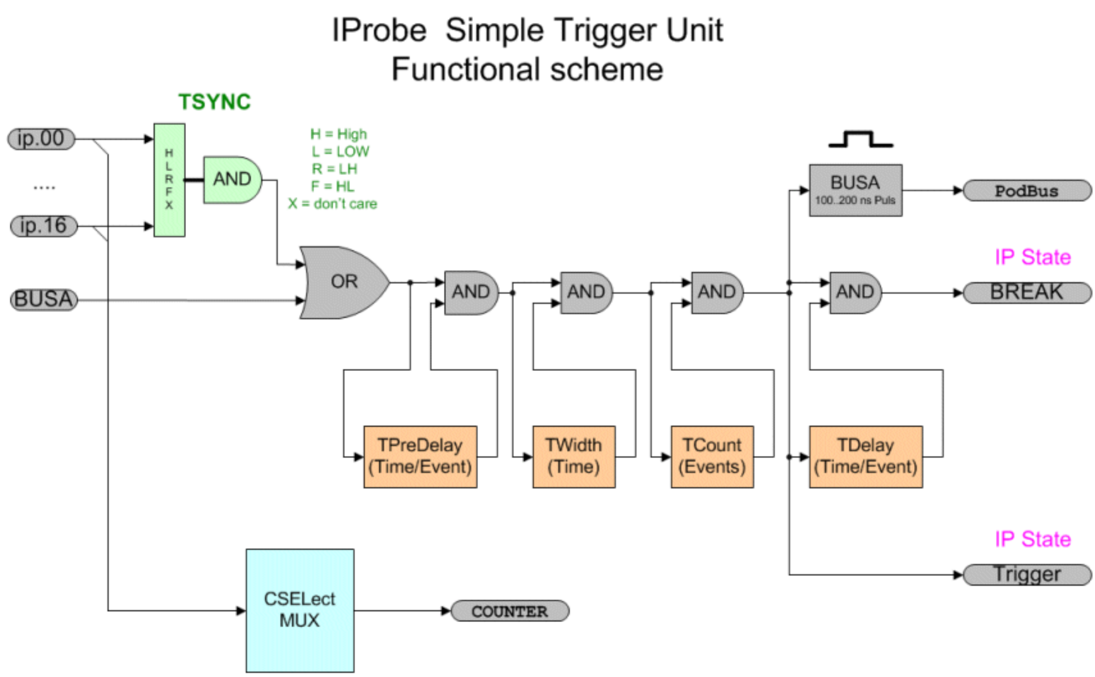


Illustration of the Simple Trigger Unit functionality

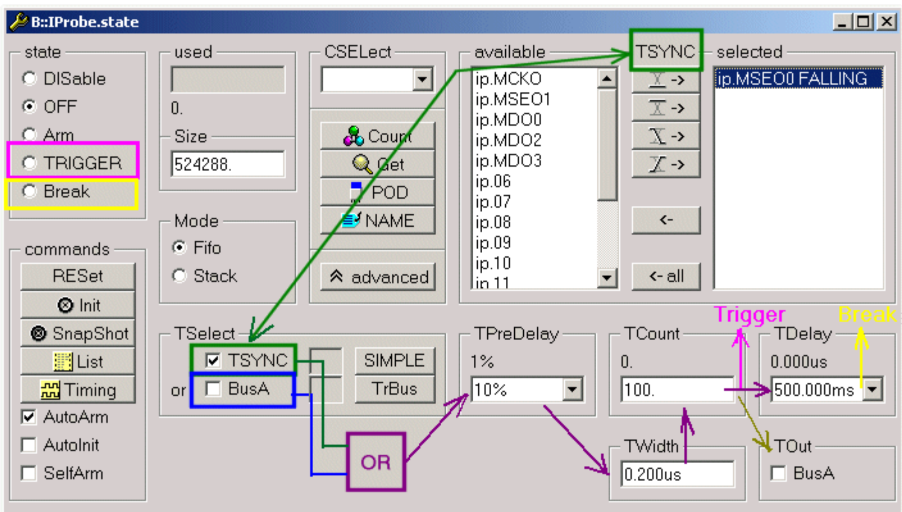


For the timing mode of the IProbe there is a Simple Trigger Unit (STU). The Simple Trigger system can be used to stop recording of pattern in the IProbe trace memory. It allows to define one or more input signal or other external events to be used as a trigger event to stop tracing immediately or to start trigger counters first and stop recording afterwards. The trigger counter allow to adjust the trigger point inside (trigger point is marked in the trace) or outside the trace memory.

The state of a signal can be LOW, HIGH or can be a rising or falling edge of the data stream. All selected trigger inputs are treated as a trigger pattern. All not selected signals are treated as don't care. If the trigger pattern (states and edges) is true, the trigger is valid and cause the next action. The next action can be either immediately stop of recording or start of the trigger counter system.

The STU will be initialized during any kind of **Init** and started as soon as the IProbe is armed. As long as no trigger event is valid, any alternation of an input signal level causes an entry in the trace memory. In FIFO mode infinitely, in STACK mode as long as the trace is not full. If a specified trigger pattern occurs, first the selected TPreDelay and then the TCounter is started. As soon as the final trigger point has been reached, the state of the IProbe changes from **ARM** to **TRIGGER**. After the **TDelay Counter** has been elapsed, the state changes to **BREAK** and simultaneously recording stops.

The picture below gives an overview of the relation of the trigger signals and the order in which the trigger counter are started.



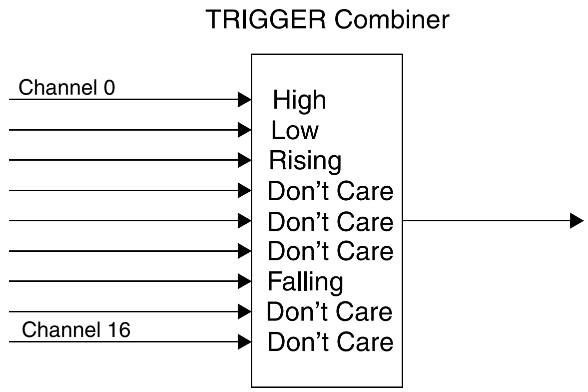
Trigger Input

There are two sources for trigger signals. If **TSYNC** is selected, one or more signals of the input probe can be used as trigger. If **BusA** is selected, the system internal ***Podbus*** trigger signal is used as trigger input. BusA can be driven by any other unit connected to the ***Podbus***. Both sources are or-ed together.

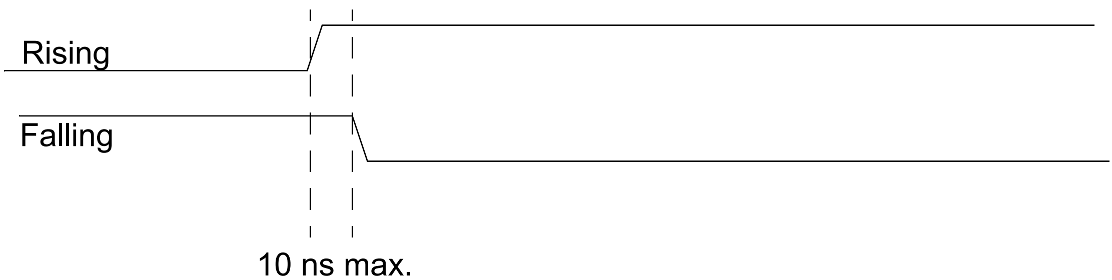
IProbe.TSYNC ON | OFF	Select one or more (max. 17) signal of the input probe as trigger pattern.
IProbe.TSELect BusA ON | OFF	Select the system internal ***Podbus*** trigger signal (BusA)

To select one or more signals of the input probe as a trigger event, the wanted signal(s) must be marked in the ***available*** area of the **IProbe.state** window and moved into the ***selected*** area. Depending on the button the state or the edge of the signal is used as a part of a valid trigger mask.

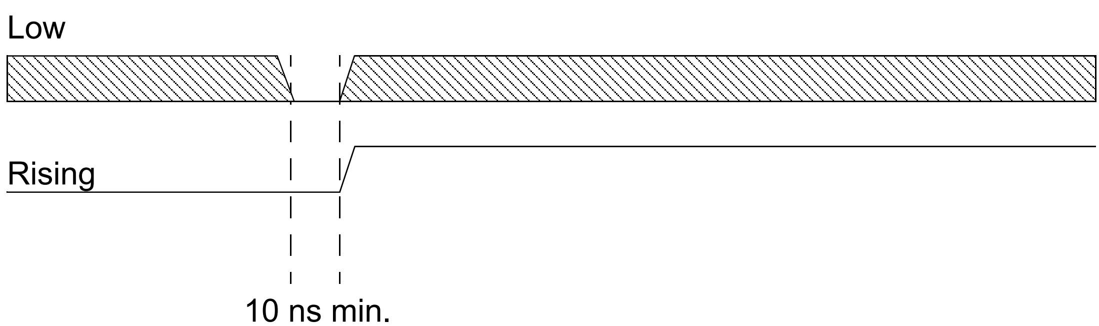
A trigger signal can be generated out of the 17 port channels. Every signal can be qualified as high, low, rising and falling edge or don’t care.



More than 1 edge can be combined to a trigger word. To detect a valid combination of edges, the edges must have a max. skew of 10 ns.

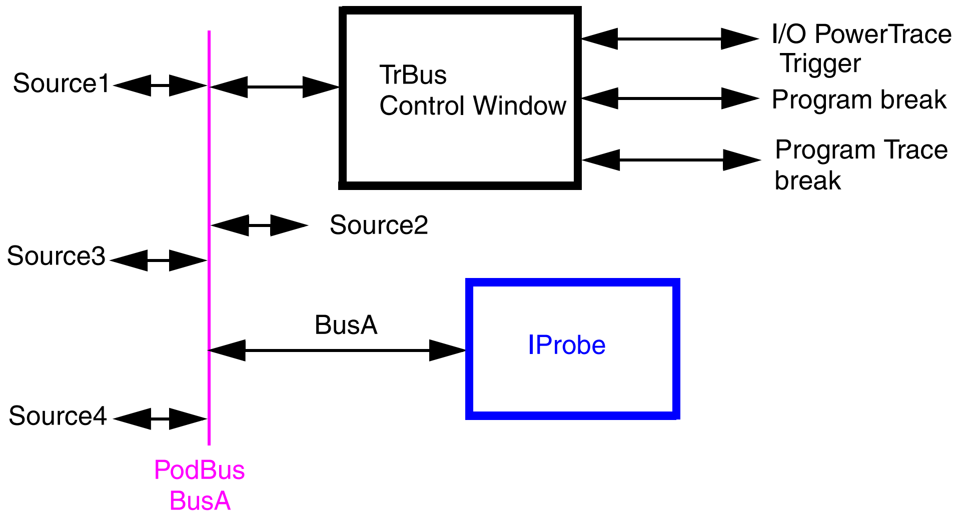


Edges and state signals can be combined. The state signal must be stable 10ns before the edge. The sampling of the state signal is guaranteed before the edge is detected.

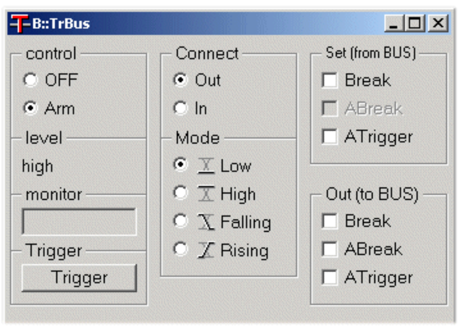


BusA - Trigger Input

As known, BusA is a system internal trigger signal of the PodBUS which could be fed by several sources. These Sources can be used to trigger the IProbe.



The button **TrBus** opens the PodBus control window.

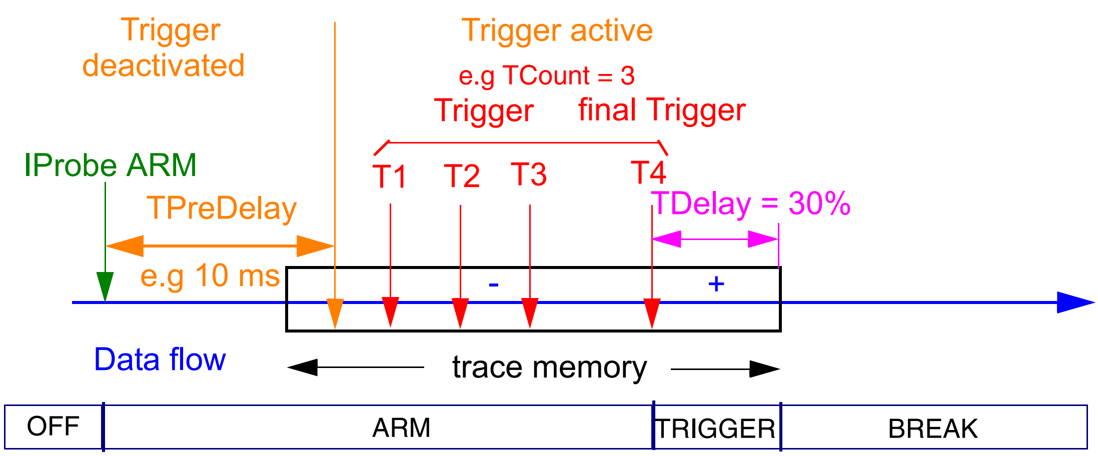


IProbe.TPreDelay	Define the trigger pre-delay counter. During TPreDelay counter is running, the trigger system is deactivated. After expiring, trigger counter or trigger delay counter will be started. The selected value can either be the percentage of trace memory size which will be sampled respectively overwritten after the occurrence of the **trigger event** released from the trigger unit or a time up to 214 seconds.
IProbe.TWidth	Define trigger pulse width. There is no trigger if selected pulse width is not exceeded. Consider that rising and falling signal edge as trigger can not be used in this case!
IProbe.TCount	Define a trigger counter. After number of triggers, trigger delay counter will be started.
IProbe.TDelay	Define a trigger delay counter. After expiring, trace will be stopped. The selected value can either be a percentage of trace memory records which will be sampled respectively overwritten after the occurrence of the **trigger event** released from the trigger unit or a time up to 214 seconds.

The actual value of **TDelay** or **TPreDelay** can be less if the recording is **switched to off manually** by the user before the maximum value is reached.

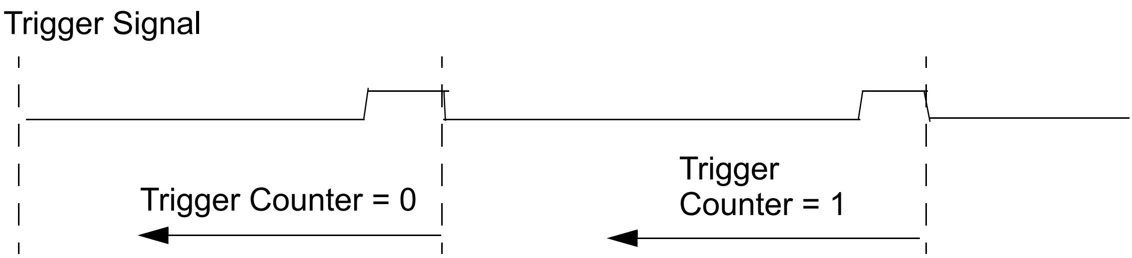
Trigger PreDelay

The PreDelay counter can be used to activate the trigger system after a certain delay time or after a certain number of trace entries.



Trigger Counter

The trigger counter delays the final trigger event on the n-th event of a valid trigger condition. The value zero means triggering after the first occurrence, one on the second occurrence of the trigger event.

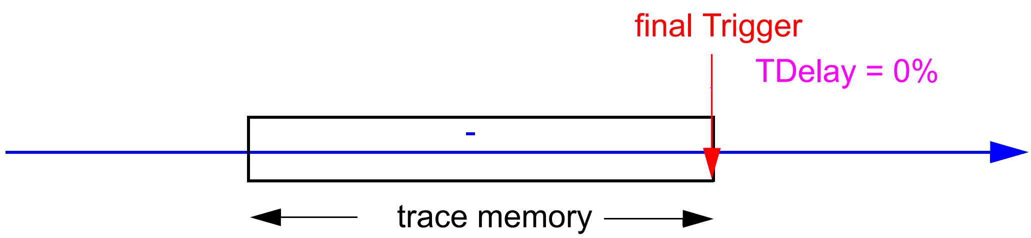


Trigger Delay

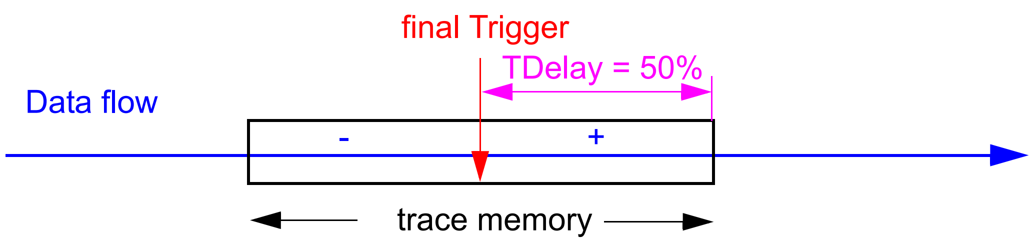
After the final trigger condition has been latched, a trigger delay is used before stopping the IProbe analyzer. The delay can be defined as an absolute time or in percentage of the trace storage. It can be used to place the trigger event in center or wherever in the visible trace window (even outside the trace memory).The default trigger delay is 0. In this case the trigger point is at the last sampled record in the trace memory.

e.g.

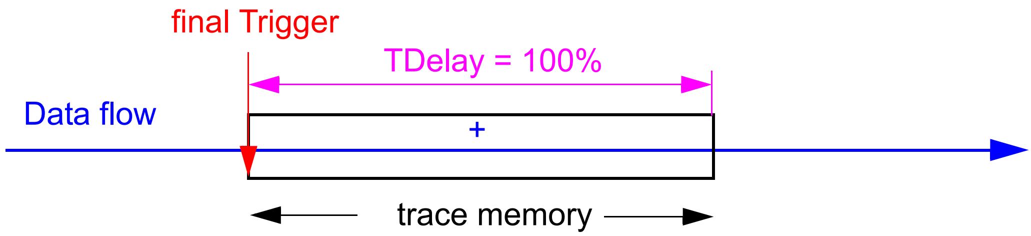
0% = recording stops immediately after trigger occurs. Trace memory contains maximum numbers of data before trigger. Data behind trigger can not be seen. Record numbers are negative.



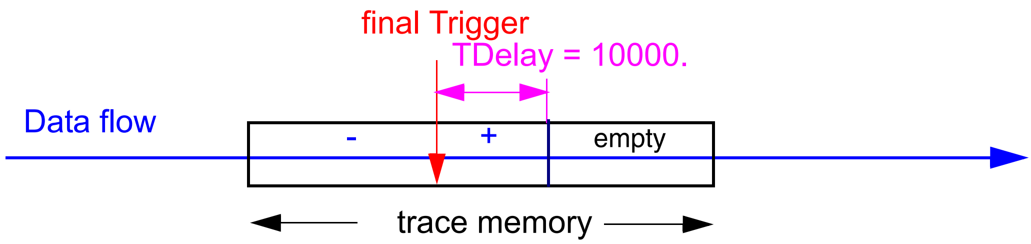
50% = up to half trace size records will be recorded. Trace memory contains data before and behind the trigger event. Record numbers before the trigger are negative, behind positive



100% = up to full trace size records will be recorded. Trace memory contains maximum numbers of data behind trigger. Data before trigger can not be seen. All record numbers are positive

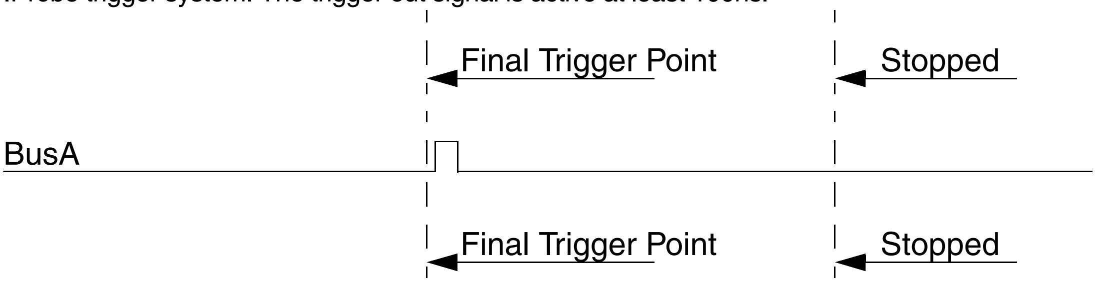


TDelay = 10000. Depending on when the trigger occurs, trace memory may not be full.



Trigger Out

When reaching the trigger state, all other Podbus units of the debugger system can be triggered by the IProbe trigger system. The trigger out signal is active at least 100ns.



IProbe.TOut on/off	Activates/deactivates Trigger Output Signal (BUSA)

Simple Trigger for Analog Mode

Simple Trigger functionality will not be offered in case of analog mode. **TSYNC** and **BusA** are not available. There are also no trigger counter. Trace can only be stopped by **Stack** Mode or by **ARM / OFF**

Universal Counter Signal Selection

Probe.CSELelect	Select counter signal

Example

```
IProbe.CSEL ip.0                    ; selects signal RD

C.Select ip.0                       ; selects signal
```

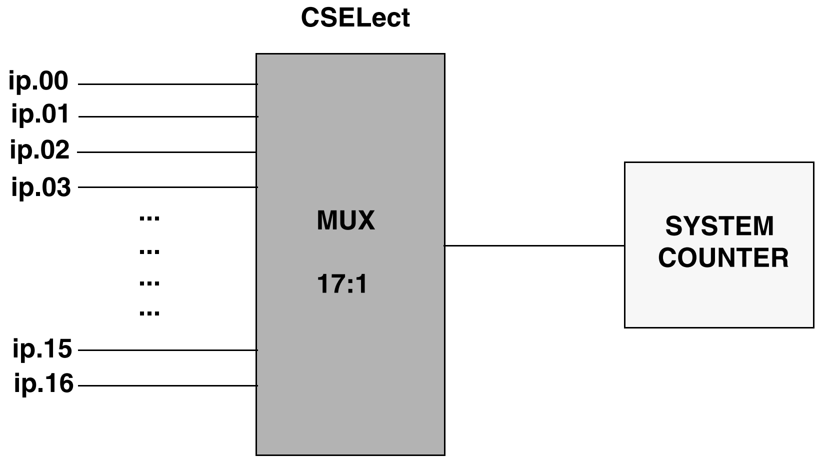
The universal counter is the logic measurement system for pulses and frequencies. An input multiplexer enables the counter to measure all important CPU signals and all external probe inputs. Therefore the counter input normally need not be hard wired to the signal.

The Counter can be displayed and selected by entering **Count** or via a pull down menu **(Misc.)**.

An IProbe input signal can also be a source and can be selected by the function **IP.CSELect**. *<input>* or via the **IP.state** window.

The function **Count.Mode** is used to change the counter mode and the **Count.Gate** function defines the gate time. Frequency and event analyzing may be qualified by the foreground running signal.

If there is no event counting, it will be possible to activate more than one count window. Every window represents a separate counter. For example it is possible to check the clock frequency and the pulse width on some probe inputs simultaneously.



The count ranges are:

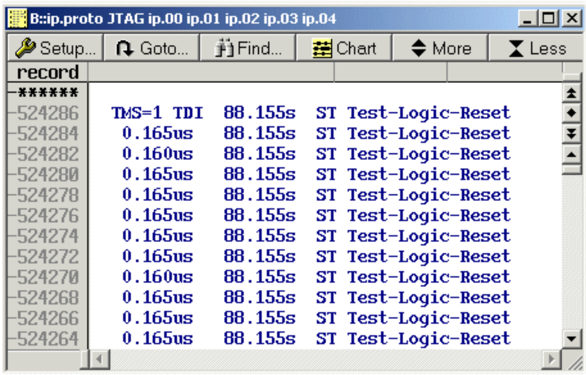
frequency:	0 … 200 MHz
Pulse width:	100 ns 300 days
Period:	100 ns 300 days
Events:	2.8 * 10E+14, max. rate 10 MHz

Protocol Analysis

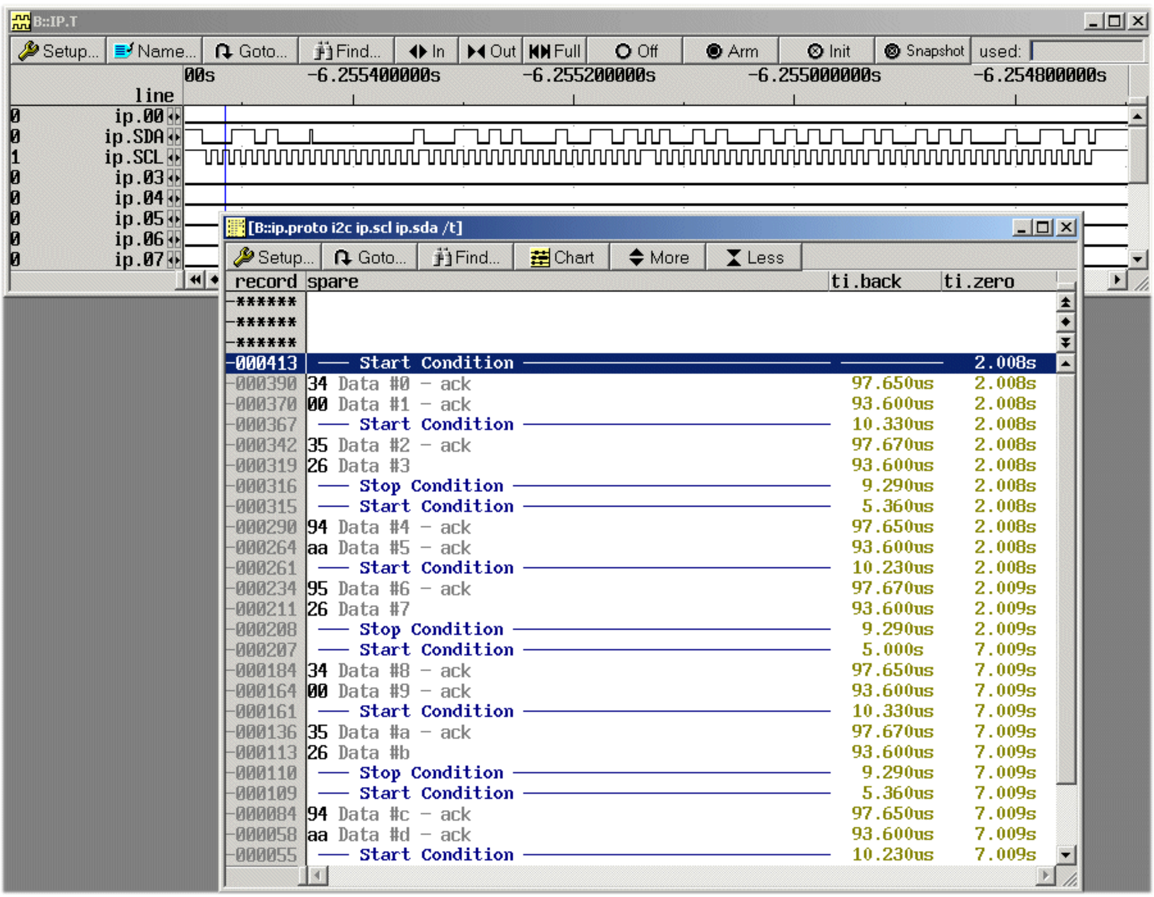
A very interesting feature of all TRACE32 timing trace tools is the protocol analysis feature. There are many built-in protocol decoders (JTAG,CAN,USB,I^2C,ASYNC) available. It is also possible to use custom protocol decoders. For more information on how to write a custom protocol decoder, see **"Protocol Analyzer Application Note"** (protocol_app.pdf).

This feature is just available in Timing Mode.

JTAG protocol example:



I2C protocol example:



Timing Mode Restrictions

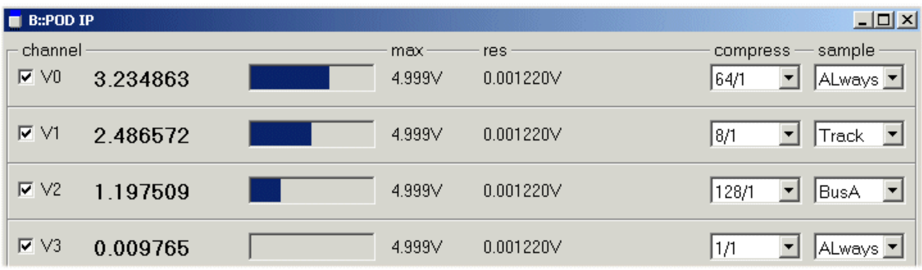
- Input signals shorter than 10ns will not be recognized as static LOW or HIGH.

Simple Trigger for Analog Mode

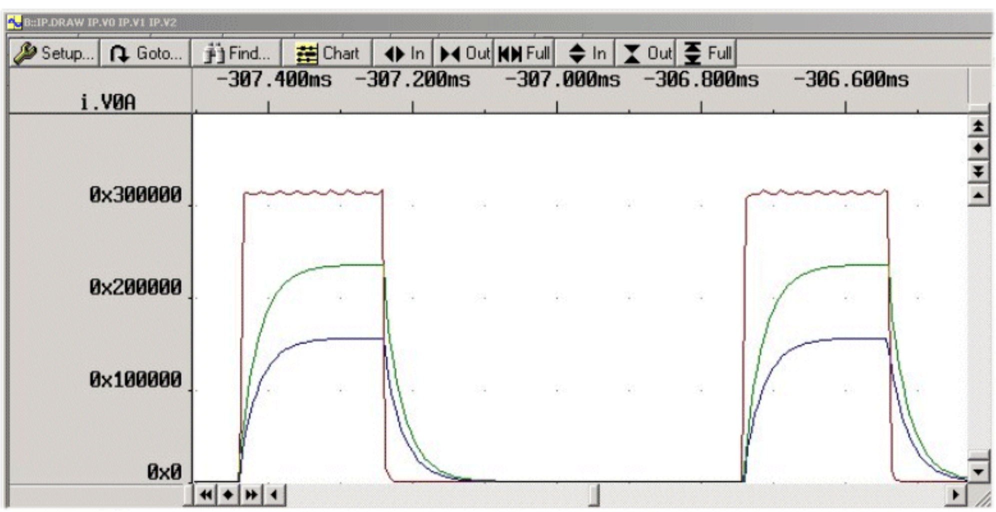
Simple Trigger functionality is not available in case of analog mode. **TSYNC** and **BusA** can not be used for trigger. There are also no trigger counter. Trace can only be stopped by **Stack** mode or by **ARM / OFF**.

Voltage Measurement

There are 4 voltage measurement channels. Each channel must be enabled and the appropriate input pin must be connected to the voltage source and to a ground pin. As long as the IProbe is enabled and not in **ARM** state, the current voltage will be displayed. Depending on the compression factor and depending on the sample source, the result of the A/D conversion will be stored in the IProbe memory.



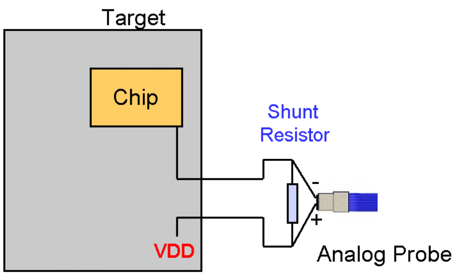
The recorded result can be displayed in several ways. An impressive way is the **IP.DRAW** instruction



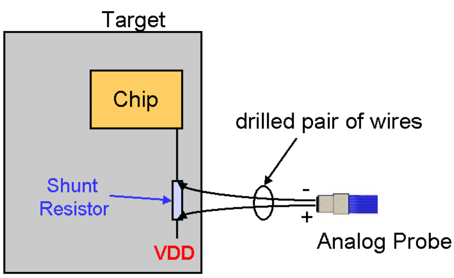
Current Measurement

There are 3 current measurement channels which are based on voltage measurement on a shunt resistor. The shunt can either be located on the target or external the target. One must take care to connect the + port of the current channel to the higher voltage and the - port to the lower voltage.

Shunt external



Shunt on the target



For shunt evaluation it is important to know that full scale voltage above the shunt must not exceed **0.125 V**.

Shunt formula: **Rs = 0.125V / Imax**

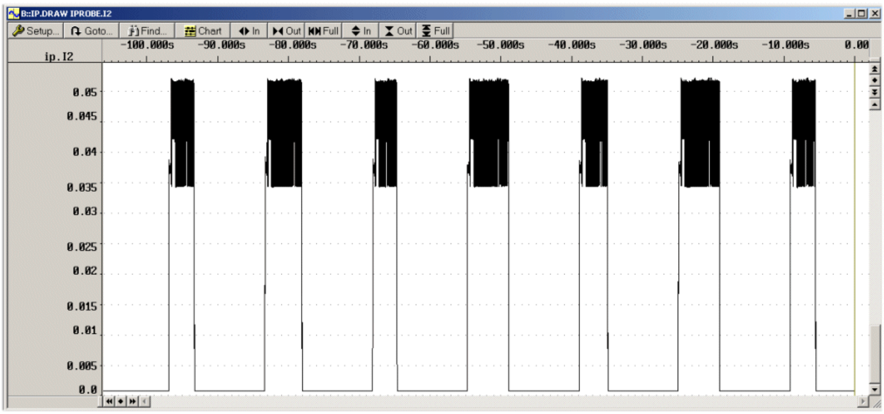
If one can live with a voltage reduction of 125 mV in case of full supply current, full range of the 12 BIT ADC can be used.

For low voltage applications, a smaller shunt value can be used, but in this case one can just expect to get a part of the available full scale range.

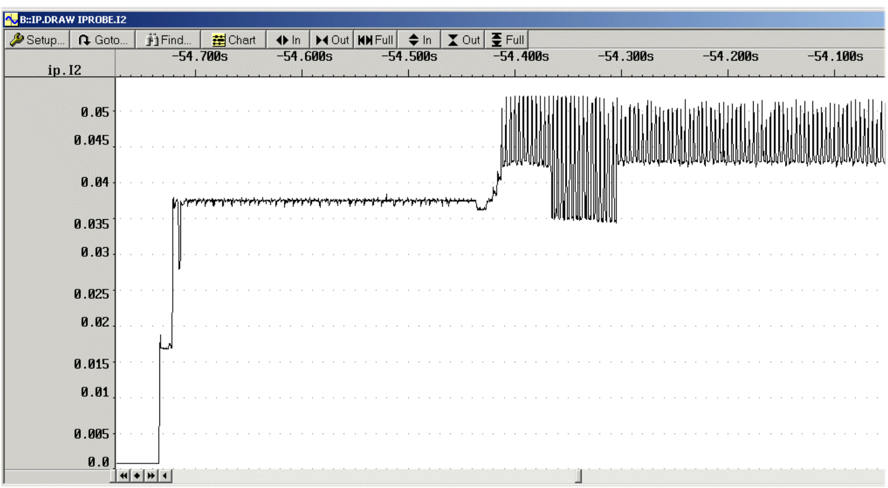
The value of the shunt must be entered in the analog settings window (in shunt area of each channel). The current resolution and full scale current will then be displayed automatically.

If the IProbe is not **DISABLED** or not in **ARM** state, the result of the measurement will be instantly displayed. just during the time the IProbe is in ARM state, there is no display of the actual voltage, current or power values.

There are several ways to display the recorded analog results. The most important is the **IP.DRAW** instruction.



More details can be seen if the window is magnified using the **IN** and **OUT** button on top of the screen.



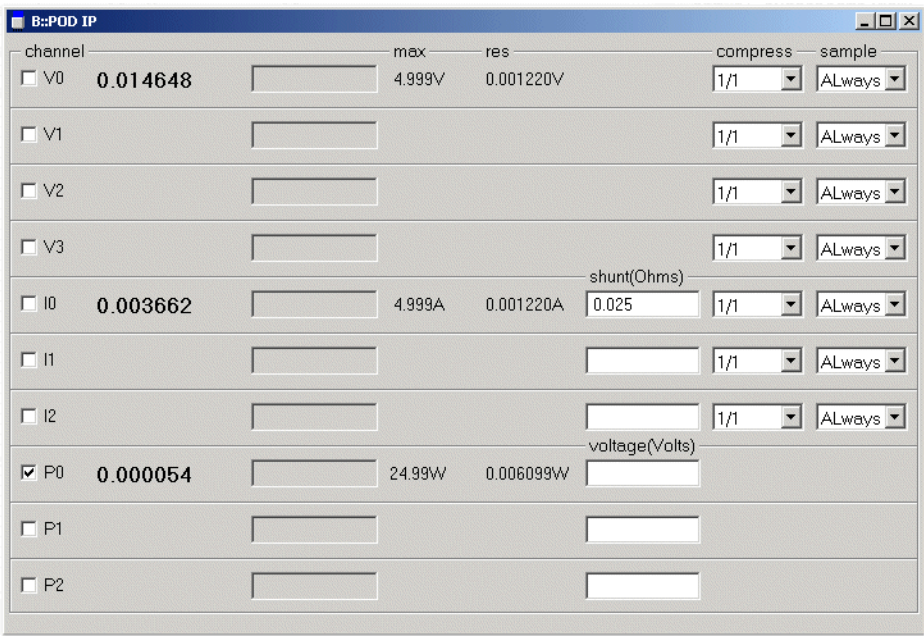
Power Measurement

Power measurement needs current and voltage. The current value is multiplied by a voltage value and results in the power value. The displayed power value is calculated and **not** generated in real time during recording.

To be able to record power consumption with the IProbe, the current path must be connected and prepared (shunt) exactly the same way as described for current measurement.

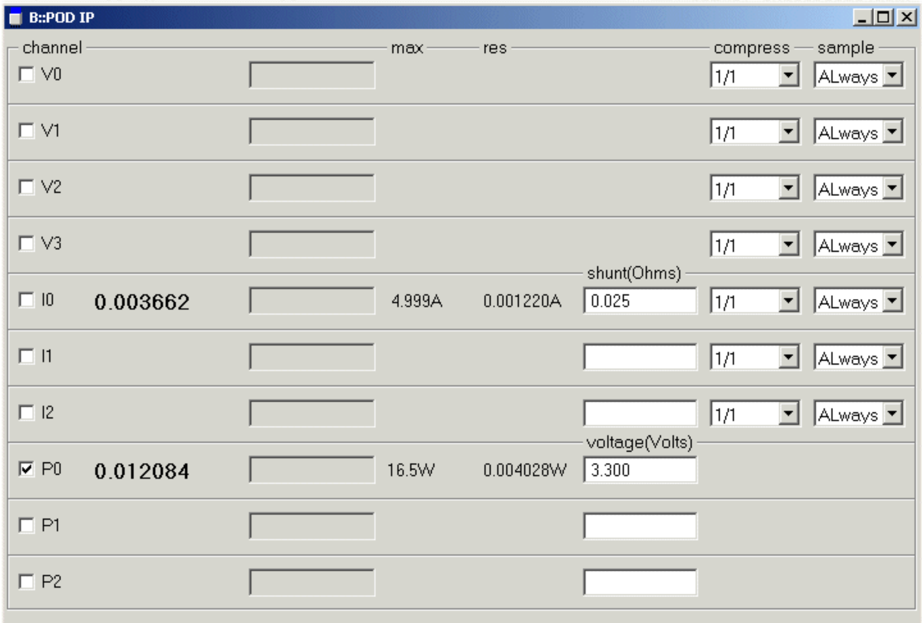
For the voltage path there are two solutions to get better flexibility.

1. The voltage value can be taken by one of the voltage input channels. If P0, P1 and/or P2 is selected and there is no entry in the **voltage(Volts)** area, IProbe assumes that one of the voltage channels is used to provide the voltage value. The appropriate voltage channel is selected automatically.




```
P0 is related to I0 and V0.
P1 is related to I1 and V1.
P2 is related to I2 and V2.
```

2. The voltage value can be predefined manually. For higher voltages than 5V it's better **not** connect the voltage path to the target. If P0, P1 and/or P2 is selected and there is an entry in the **voltage(Volts)** area, IProbe uses this manually entered value for the power calculation.



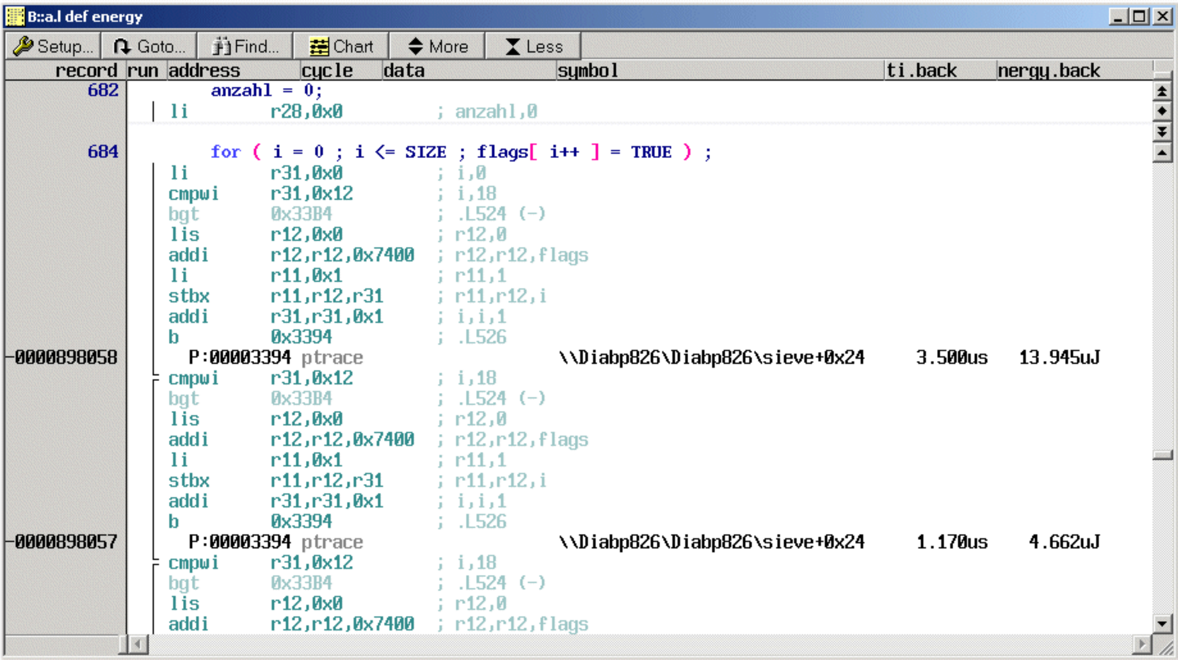

```
P0 is related to I0.
P1 is related to I1.
P2 is related to I2.
```

If the IProbe is not **DISABLED** or not in **ARM** state, the result of the power measurement will be instantly displayed.

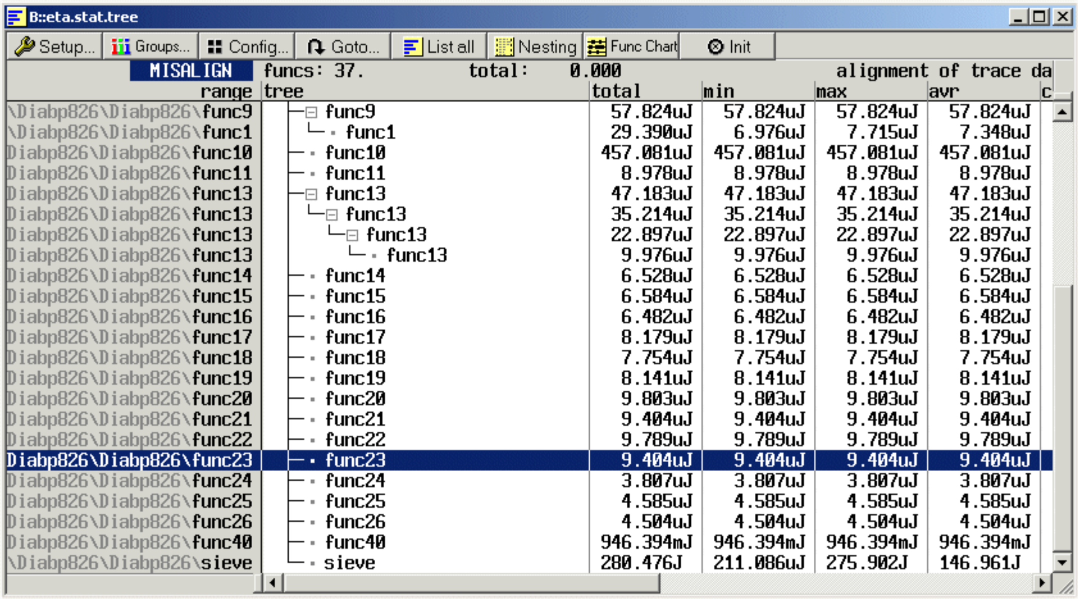
Energy is power over time.

If the consumed power of a target is known for a time, the energy consumption can be calculated. Due to the correlation of traced program flow and traced current and voltage values with timestamp, it is easy to provide Energy Statistics and a visualization of Energy Consumption. It allows code optimization regarding minimum of power consumption.

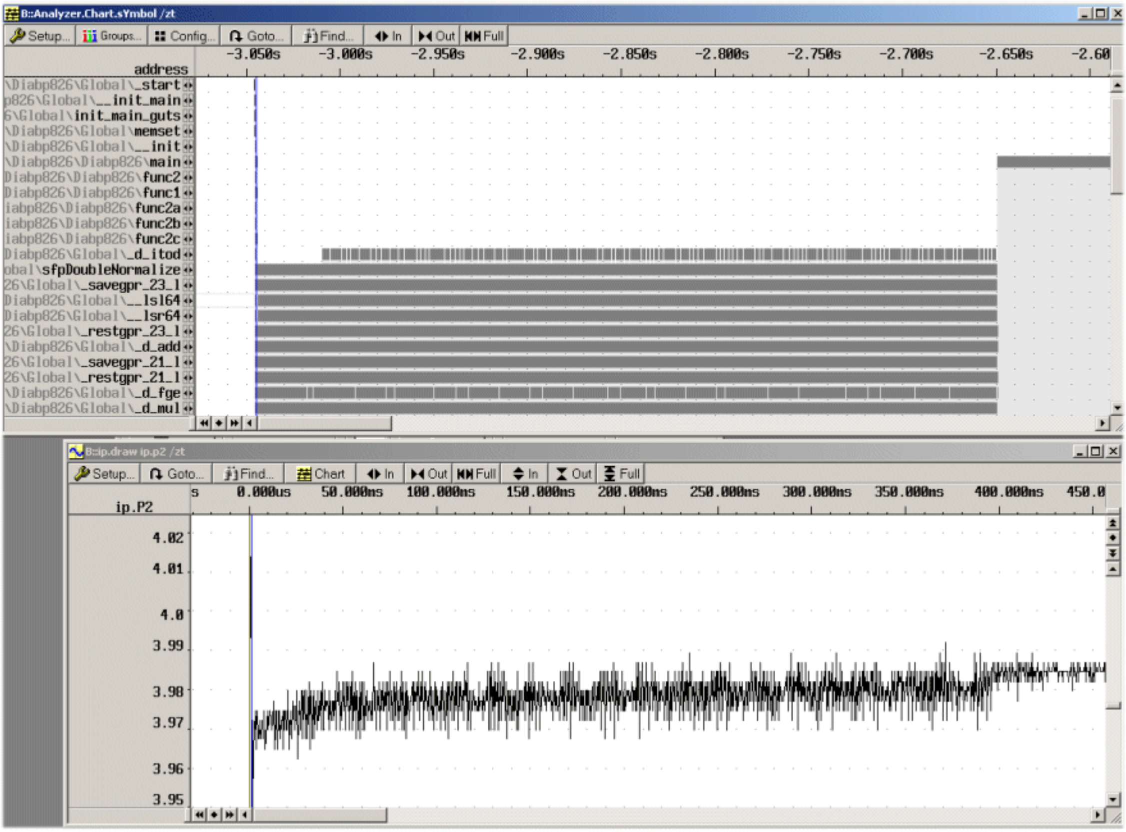
Program Flow and Energy Consumption



Energy Statistic



Power Consumption and Program Function Chart Display



Analog Trace Time Coverage Calculation

To calculate the time which can be covered by certain setup in analog mode or to calculate the settings for a particular given time frame, the formulas below can be used.

Analog-Channel

Trace memory and time consumption calculation, depending on channel numbers and average value of each channel.

1. **Time period for a channel x**

   $t_{cx} = a_{cx} * c_n * 1{,}6\ us$

2. **Time period for the slowest channel (record period)**

   $t_r = a_{cmax} * c_n * 1{,}6\ us$

3. **Number of trace entries for a record period**

   $$e_r = \frac{a_{cmax}}{a_{c0}} + \frac{a_{cmax}}{a_{c1}} + \frac{a_{cmax}}{a_{c2}} + \frac{a_{cmax}}{a_{c3}} + \frac{a_{cmax}}{a_{c4}} + \ldots$$

   Add 0 for each inactive channel !

4. **Time captured by the Analyzer**

   $$T = \frac{TM}{e_r} * t_r$$

Legend

TM : trace memory record size

record : a record contains all ADC values of all active channels, but at least one value of each channel.

t_r = time per record
(time between two entries of the channel with the highest average value)

t_{cx} = time per record x
(time between two entries of channel x with it's specific average value).

a_{cx} = average value for channel x

a_{cmax} = highest average value

c_n = number of active channels

e_r = number of trace entries of a record period

T = total amount of time until trace memory is full (with the given channel settings)